SAGE: Task-Environment Platform for Evaluating a Broad Range of AI Learners

Leonard M Eberding[1,2] Kristinn R Thórisson[1,3]
Arash Sheikhlar[1] & Sindri P Andrason[1]

[1] Center for Analysis and Design of Intelligent Agents, Reykjavik University, Iceland
http://cadia.ru.is thorisson@ru.is, arashs@ru.is, sindri@andrason.com
[2] Institute of Photogrammetry and GeoInformation, Leibniz U., Hannover, Germany
https://www.ipi.uni-hannover.de l.eberding@stud.uni-hannover.de
[3] Icelandic Institute for Intelligent Machines, Reykjavik, Iceland

Abstract. While several tools exist for training and evaluating narrow machine learning (ML) algorithms, their design generally does not follow a particular or explicit evaluation methodology or theory. Inversely so for more *general* learners, where many evaluation methodologies and frameworks have been suggested, but few specific tools exist. In this paper we introduce a new framework for broad evaluation of artificial intelligence (AI) learners, and a new tool that builds on this methodology. The platform, called SAGE (Simulator for Autonomy & Generality Evaluation), works for training and evaluation of a broad range of systems and allows detailed comparison between narrow and general ML and AI. It provides a variety of tuning and task construction options, allowing isolation of single parameters across complexity dimensions. SAGE is aimed at helping AI researchers map out and compare strengths and weaknesses of divergent approaches. Our hope is that it can help deepen understanding of the various tasks we want AI systems to do and the relationship between their composition, complexity, and difficulty for various AI systems, as well as contribute to building a clearer research road map for the field. This paper provides an overview of the framework and presents results of an early use case.

Keywords: Evaluation · Generality · Autonomy · Task-Environments · Evaluation Framework · Machine Intelligence

1 Introduction

Many good reasons exist for wanting proper evaluation methods for machines capable of complex tasks [4], including: (a) To gauge research progress—measuring difference in performance between two or more versions of the same system can elucidate limitations and potential of various additions, modifications and extensions of the same architecture; (b) to compare the performance and potential of one or more AI systems across a set of tasks; and (c) to compare different AI systems on the same or a variety of tasks. The dependent variables in such evaluation will be conditional on the evaluation's purpose, whether it's learning

a single task or many, to learn quickly, reliably, autonomously, to learn complex things, causal relations, to handle novelty, or some combination of these—or even more. Most proposals for evaluating artificial intelligence (AI) systems focus on subsets of the possible spectrum of dependent variables relevant to general machine intelligence (GMI), or are narrowly focused on particular tasks or domains.

Good measuring tools and methodologies are necessary to assess progress in any scientific domain. They should allow comparison of systems of numerous kinds. The vast majority of evaluation methods proposed to date rely on a single measurement, where a series of multiple measurements could possibly much better separate between autonomous, general systems and narrow machine intelligence (NMI). Furthermore, many current evaluation strategies focus on evaluation of (single) tasks especially chosen to evaluate a particular (narrow) machine learning algorithm. GMI-aspiring work cannot limit itself to one or a small set of tasks, especially if they lack a) any sort of real-time or continuous settings, b) complex causal chains, c) a multiple goals, or at least d) variable feedback (reinforcement), including its absence (except in the form of a top-level goal). These features (or a subset of them) can be found in most human tasks.

While GMI-aspiring systems should ultimately be able to tackle tasks of those kinds, most evaluation platforms do not provide any functionality for creating them. This makes an evaluation of our progress on generality more difficult, since the same task environments well-suited for testing NMI do not address such matters; while platforms like OpenAI Gym [5] or the Arcade Learning Environment (ALE) [2] all provide functionality to test narrow agents, they fail to offer easy construction of tasks of greater complexity.

The SAGE task-environment simulation platform proposes to bridge the gap between evaluation of low- and high-level intelligence by providing methods for constructing and analyzing performance on tasks in a fine-grained manner. SAGE is based on breaking tasks, and the environments they are performed in, into variables (observable, unobservable, manipulable, and non-manipulable) and transition functions that control their changes over time [19, 20]. Task-environments in SAGE may be constructed with a variety of characteristics and levels of complexity, including causal and statistical relations, determinism and non-determinism, hidden and partially-observable variables, distracting variables, noise, and much more.

Puzzle boxes, to take an example of a human-level task, may lie at the far end of a complexity spectrum, yet are regularly solved by human intelligence. Such boxes invariably present features that include: a) not giving evidence for whether a chosen action was "good", or "bad", at least not by an easily observable score; b) containing highly complex, non-observable causal chains which need to be hypothesized and understood, to some extent, to solve the puzzle, and even c) acting independently from outside action, through timers. SAGE makes the setup of such tasks easier for an evaluator by providing an architecture that supports continuous changes in task variables and rewards, even with an external clock. A puzzle box task could be divided into a variety of sub-tasks, each with increasing complexity. If narrow agents are being evaluated on a subset of such

a task, the environment can be set up to give direct feedback (reward) about the value of any chosen action and affected variables possible directly observable. For GMI-aspiring systems such feedback and observablity can be reduced or removed, making a task reach human-level complexity.

The architecture of SAGE is based on a new MVC-A (Model-View-Controller-Agents) paradigm in the ROS2 framework [14], enabling the whole system to be physically run on separate processors and computers to reduce interference of processor loads on simulation integrity. Dividing a simulation logically into these parts also makes for easier adjustments of each part, independent of the others, allowing an evaluator to more easily change individual parameters and task design, up front and at runtime.

The paper is organized as follows: Section 2 covers related work, including the requirements proposed for such evaluation platforms; section 3 describes how SAGE has met these requirements; section 4 presents early results of using the framework, and section 5 draws conclusions.

2 Related Work

To date, methods for evaluating general intelligence tend to either exclusively target humans, such as IQ tests, or to exclusively target very general ("human-level") intelligence—examples include Winograd's Schema Challenge [10], Lovelace Test 2.0 [15], and the Toy Box Problem [7]. Others are too domain-specific, e.g. general game-playing (cf. [17]), or highly dependent on knowledge of human social conventions or human experience and skills, e.g. Wozniak's Coffee Test and the Turing Test [13]. What is needed, as many have argued [1, 4, 6, 19], is a flexible tool that allows construction of appropriate task-environments (TE), along with a proper task theory that enables comparison of a variety of tasks and environments. Thórisson et al. (2015) list eleven dimensions that ideally should be controllable by a creator of a task-environment for measuring intelligent behaviour [19]; Russell & Norvig (2016) present a somewhat comparable subset of seven dimensions [16]. The environment can be categorised along different dimensions, namely determinism (see [3] regarding the importance of noise control), staticism, observability, agency, knowledge, episodicity, and discreteness. TE properties include, in addition, ergodicity, asynchronicity, controllability, number of parallel causal chains, and periodicity [16, 19].

Lately, evaluation methods have focused on (general) game playing using the ability to play games as an indicator for the systems sophistication. Using psychometric evaluation like item response theory (IRT) it was shown that the difference of performance score between different ML techniques does not necessarily correlate with the systems level of abilities [12]. Thus a simple performance rating like achieved game score cannot describe the progress of AI by itself [6]. By evaluating the ability to handle TE property changes over different learners a conclusion can be drawn on the abilities of the learner in regards to autonomous generality. Such conclusions should be accompanied by evaluation strategies like IRT to show the significance of the progress. By isolating and adjusting single

parameters of the TE and testing on different learners it is furthermore possible to describe task difficulties in regards to the properties of the TE.

We have taken the evaluation of NMI and GMI further than current platforms by (a) providing the possibility to create tasks for NMI and GMI, (b) introducing changeable complexity dimensions in the generated task-environments, (c) making novelty introduction possible in any dimension (novel task, novel transitions, novel state observation, novel controllability), and (d) by making those changes during runtime without human interference in order to test the systems autonomy in coping with (b) and (c).

3 SAGE: Overview of Structure & Use

SAGE (Simulator for Autonomy & Generality Evaluation) is built to enable flexible construction of task-environments for evaluating artificial intelligence systems. One of its key requirements is that it can be used to evaluate both narrow AI systems and GMI-aspiring ones. It follows a tradition already laid out in prior work (cf. [4,18,19]) and is perhaps closest in spirit to Thorarensen's FraMoTEC [18]. In SAGE, assessing an AI system's ability to address novelty can be done by introducing new undefined variables, possibly with unknown transition functions, and unknown relations to other variables, either of which may or may not be similar to the behavior of priorly observed ones. The response of a learner to variable changes leads to conclusions about its ability to extract causal relations and its autonomy in exploiting them to achieve goals.

3.1 Requirements

The requirements for SAGE follow closely the eleven desired features listed by Thórisson et al. [19] that a task-environment platform for evaluating AI systems should contain. Any platform that meets these requirements should in theory be useful to evaluate not only GMI systems but in fact any learner.

While SAGE is still under development, it already meets all of those eleven requirements, in some way: *Determinism*, *dynamism*, *observability*, *episodicity*, and *discreteness* can be adjusted both beforehand and during the training / learning / evaluation processes, automatically without human intervention. *Stochasticity* can be adjusted in the observable variables, agent actions, and in environment dynamics, with *reproducibility* being supported through stored randomization seeds. Dynamism and episodicity can be changed by either run-time introduction of different tasks, or changing environmental variables. *Observability* and *manipulatability* of variables can be made at run-time, supporting *ergodicity*. Same goes for discreteness of observation and/or action, providing *controllable continuity*. These features make evaluation of the effects of sensor noise on learning, actuator impreciseness, and noise in hidden variables (e.g. wind forces) possible. *Causal chains* are constructed by chains of variable dependencies. Training on a variety of sensors before removing causally redundant ones may test a learner's capacity for knowledge generalization and extraction

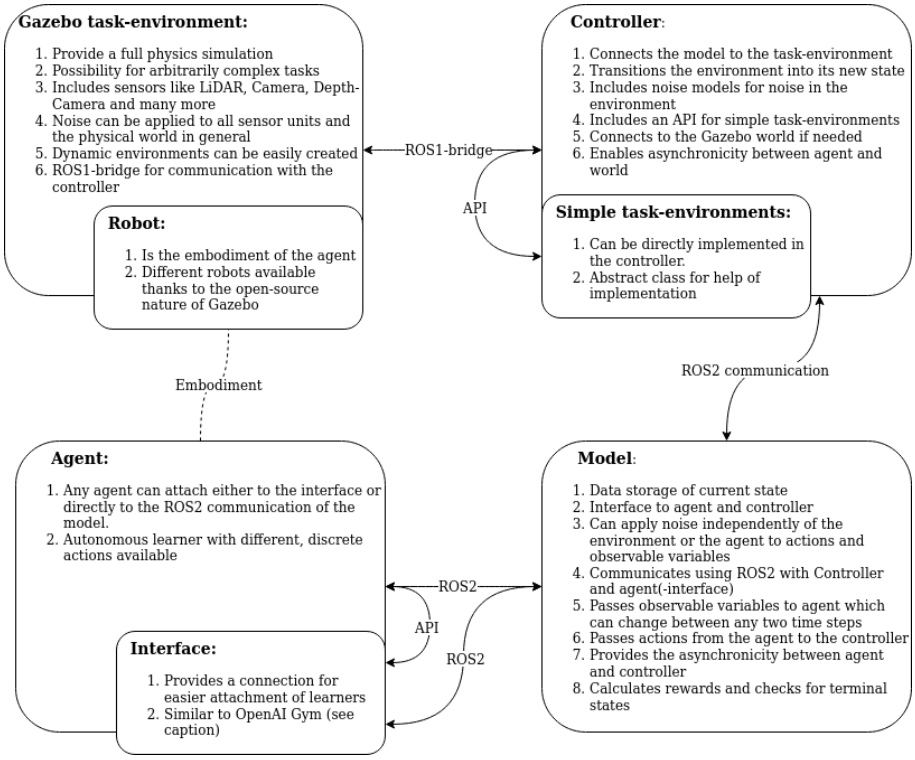

Fig. 1. Flowchart showing the main SAGE components and their interactions with each other, following the MVC paradigm, extending it with an *Agent* component that enables connecting one or more agents (similar interface as OpenAI Gym [5]). Visualization is via Gazebo [8] 3D rendering, using its standard API. In accordance with MVC, the Model node handles data storage, and includes an environment-independent noise generator for simulating stochasticity.

of causal relations. The same holds for modifying controllability with which a learner could exploit causal relations by applying previously unavailable actions to causally linked variables.

SAGE is implemented in ROS2 [14][1], which provides for a flexible framework that allows running a setup on multiple computers. Visualization of any parameters can be via Gazebo [8][2], as well as ROS2's internal `rqt-graph` function. All adjustable parameters in SAGE are wrapped in YAML-files, making adjustments straight forward, by compiling before sessions or changes at run-time.

[1] https://index.ros.org/doc/ros2/ – accessed Feb. 26^{th} 2020.
[2] http://gazebosim.org/ – accessed Feb. 26^{th} 2020.

3.2 Architecture: Model-View-Controller-Agents (MVC-A)

The architecture of SAGE follows the model-view-controller paradigm, extending it with an agent component that allows one or more learners and teachers to connect dynamically to a task-environment. Each part of our MVC-A architecture is implemented as a ROS2-node [14], using ROS2 for platform-independent inter-process communication (see Fig. 1). The current task-environment state is stored in the Model node, including all observables, non-observables, manipulables, time, and energy. The Model exposes all observable variables via network communication to any attached Agent through an interface. The same interface receives actions chosen by the agent, processes them into manipulables, if needed, and passes them to the Model node. Noise and discretization can be applied to any data independently from the rest of the simulation. The Controller manages the simulation through a network connection.

Simple tasks can be easily added to the system as task modules, while the controller itself provides an interface to a Gazebo [8] simulation of a 3D world including a variety of robots, sensors, sensor-noise models, etc. ROS2 as middleware between Agent and evaluation platform makes the learners interface independent from the task-environment and therefore provides easy attachment of any learner to the evaluation platform. For communication, either an implemented Python module can be used or the agent can be directly attached to the ROS2 message system. The View is either provided by Gazebo itself or rqt-graphs via a standard network connection, but can be served by any external node that can make use of ROS's API. The connection to rqt-graph is also established using network communication enabling remote monitoring during evaluation.

The MVC-A approach provides a straightforward way to introduce more than one simultaneous learners in the simulation, as any number of agents can communicate with the world simultaneously through the model interface.

This approach brings many advantages. To name two, the logical separation of agent and environment makes evaluation of a learner's resource management possible, and by dividing Agent, Model and Controller into separate processes, real-time processing and asynchronous calculations can be added as needed. These features are especially important when GMIs are evaluated to fulfil the assumption of limited time and resources in the task environment [22].

4 Proof of Concept

As a proof of concept we tested three learners, an actor-critic (AC) [9], a double-deep-Q (DDQ) [21] learner, and Open-NARS for Applications (ONA)[3], on the cart-pole task (cf. [5]). While this task is well known in the narrow-AI ML arena [11], few if any examples of how GMI-aspiring systems do on this task exist. The experience of attaching ONA to SAGE demonstrates the usefulness of many of SAGE's features. Figure 2 shows the performance of each leaner.

[3] https://github.com/opennars/OpenNARS-for-Applications – accessed May 10^{th} 2020.

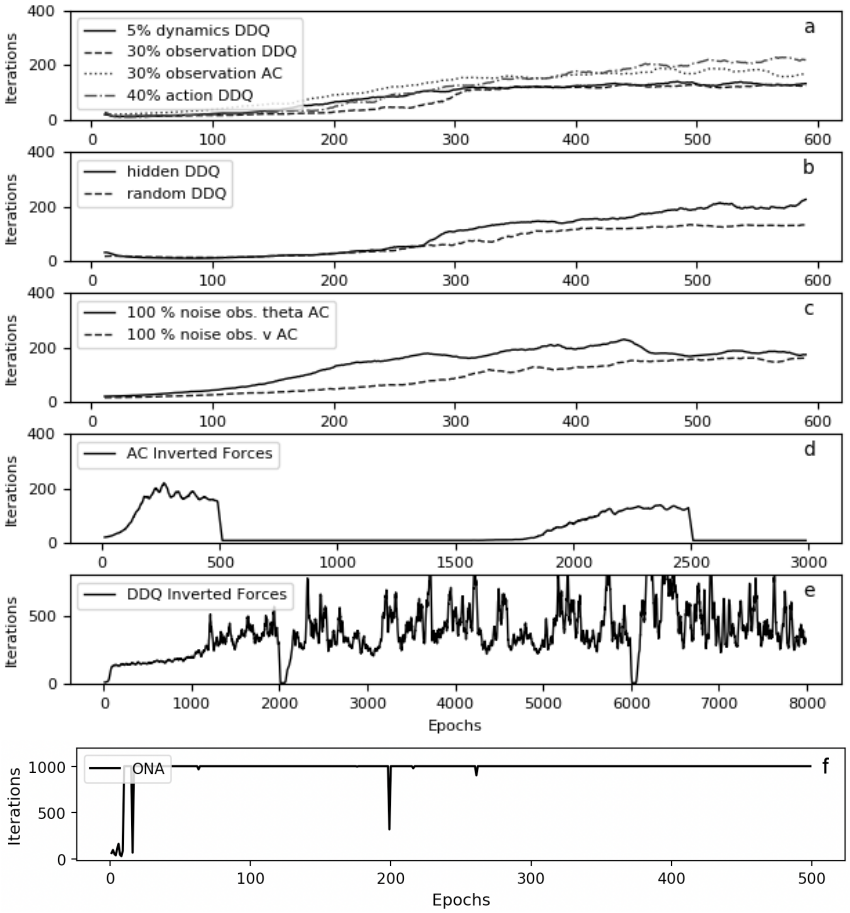

Fig. 2. Evaluation of an Actor-Critc (AC) and a Double-Deep-Q (DDQ) learner. All results are the average over 40 trials plotted with a running mean with window-size 10. **a**: Different applications of noise on the two learners. Noise on environment dynamics (3%), noise on the observation (30%), and noise on the actions (40%). Percentage in percent of the goal state ($\theta = \pm 12°$, $x = \pm 2.4m$) or commonly occurring min and max values ($v = \pm 2.4m/s$, $\omega = \pm 2.3°/s$) **b**: Test with velocity hidden from the agent and with velocity randomized ($\mu = v, \sigma = 24.00\frac{m}{s}$. **c**: Noise only on single variables of the observation. Percentage definition as in **a**. **d**: Inverted forces after 500 episodes of training AC, 2000 episodes of retraining then inverting back. **e**: Inverted forces after 2000 episodes of training DDQ, 4000 episodes of retraining then inverting back. **f**: Performance of the ONA (OpenNARS for Applications - see footnote 3) is outstanding.

1. **Three different learners on a common task**: Although the cart-pole task has only a few parameters, and may seem too simplistic for GMI-aspiring learners, for the purpose of cross-learner comparison it is a reasonable one, in our opinion. The results were surprising on two accounts.

Firstly, we were surprised that the DDQ learner did better than expected on a doubly-inverted version of it (testing for transfer learning by 180-degree reversal of the control dimension). Secondly, we were surprised by ONA's sensitivity to the format of the data (tuned by the discretization features in SAGE). In both cases the SAGE framework proved its value by allowing systematic modifications and testing automation.

2. **The influence of noise**: The first few graphs shows the differences in learning between environmental noise (noise on dynamics of the inverted-pendulum) and noise in the observations and actions received/given by the agent. Environmental noise simulates noise outside the agent, observation noise simulates sensor noise and action noise simulates actuator imprecision, respectively, for DDQ and AC. The results show that observation noise has less of an impact on performance than the dynamics, and noise on actions has no effect on learning performance at all.
3. **Coping with hidden random variables**: The DDQ-learners capability to cope with unreliable variables was tested by turning off one observable (velocity) or randomizing it with a standard deviation of 24 m/s (10x the usually occurring values). The data shows that an extremely randomized variable has a higher negative impact on learning, than hiding this variable completely resulting in the conclusion, that the DDQ learner cannot identify unreliable variables and exclude them from decision making.
4. **Influence of noise on a single variable**: To assess the importance of the correctness of the values of observables, noise was applied to a single variable. Results show, that against expectation the correctness of the velocity is of higher importance, than the correctness of the angle theta, even though velocity is not part of the failure constraint.
5. **Inversion / transfer learning**: As a test of the generality of their acquired knowledge, after training on the cart-pole we inverted the action direction (making left rigth and right left)—how would they adapt to a doubly-inverted pendulum task? The results show, that it takes almost four times as long as during the initial training to retrain the AC learner on the novel circumstances. Inverting it back after 2000 episodes of inverted training shows, that the original policy was mostly forgotten during re-training. The DDQ-learner on the other hand shows almost immediate return to previous performance, showing, that its generalization is better than that of the AC.
6. **Evaluating a GMI-aspring system**: We ran the GMI-aspiring ONA system to demonstrate SAGE's usefulness when comparing narrow and general AI systems. ONA learns the task faster than the others and handles transfer of learning much better.

These tests provide new insights into the methodologies of the three learners and current evaluation strategies. Modulation with noise of the observation and/or action variables assesses learning with noisy data; testing knowledge transfer via inversion, or hiding of variables, makes evaluating the generality and autonomy evaluation of the learners possible. When generalizing knowledge, any random variable should be excluded from future decision making to generate an expected behaviour. Further, the generality of a learner can be assessed

by changing the task-environments nature. While it is expected that inverting the forces applicable by the learner leads to an immediate performance loss, the time it takes to learn this new task (4 times the training time) in the narrow-AI systems shows that cause-effect-chains were not extracted; rather, a simple state-to-action mapping took place. The GMI-aspiring system ONA clearly outperforms the others; we are excited to see future results with varying levels of noise and inverted forces. Given the results in Figure 2 one also wonders how a human would compare, something that could be tested via visualization via Gazebo and keyboard or mouse input; other things staying exactly the same in this setup of SAGE.

5 Conclusions & Future Work

SAGE shows potential for evaluating AI architectures that follow various methodologies, bridging the gap between general and narrow AI. Our own interest in SAGE is the need to assess the progress of AI research towards general machine intelligence (GMI), however, as the examples presented here show, other uses are entirely justified. First evaluation results demonstrate some of the possibilities of this platform. A comparison of GMI-aspiring systems to narrow-AI ones not only helps highlight differences in performance and the nature of the learning of such systems, it also helps isolate their points of divergence related to deeper methodological issues, background assumptions and theoretical underpinnings.

The performance results from the learners presented here are preliminary; a future publication will present extensive tests and discuss the differences between the learners, all using SAGE of course. Future work will also include evaluating a more extensive set of learners, improve the automatic running of sets of training and evaluation sessions, and implement a library of tasks. Then we plan on making the source code available online.

Acknowledgements The authors would like to thank Hjörleifur Henriksson for help with computer setup and data collection, and Patrick Hammer for help with ONA. This work was in part supported by grants from Reykjavik University, the Icelandic Institute for Intelligent Machines and Cisco Systems, Inc.

References

1. Adams, S., Arel, I., Bach, J., Coop, R., Furlan, R., Goertzel, B., Hall, J.S., Samsonovich, A., Scheutz, M., Schlesinger, M., et al.: Mapping the landscape of human-level artificial general intelligence. AI magazine **33**(1), 25–42 (2012)
2. Bellemare, M.G., Naddaf, Y., Veness, J., Bowling, M.: The Arcade Learning Environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research **47**, 253–279 (2013)
3. Bellemare, M.G., Naddaf, Y., Veness, J., Bowling, M.: The Arcade Learning Environment: An evaluation platform for general agents (extended abstract). In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI). p. 4148–4152 (2015)

4. Bieger, J., Thórisson, K.R., Steunebrink, B.R., Thorarensen, T., Sigurdardóttir, J.S.: Evaluation of general-purpose artificial intelligence: Why, what & how. EGPAI 2016 - Evaluating General-Purpose A.I., Workshop held in conjuction with the European Conference on Artificial Intelligence (2016)
5. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: OpenAI Gym. ArXiv preprint ArXiv:1606.01540 (2016)
6. Hernández-Orallo, J., Baroni, M., Bieger, J., Chmait, N., Dowe, D.L., Hofmann, K., Martínez-Plumed, F., Strannegård, C., Thórisson, K.R.: A new AI evaluation cosmos: Ready to play the game? AI Magazine **38**(3), 66–69 (2017)
7. Johnston, B.: The toy box problem (and a preliminary solution). In: Conference on Artificial General Intelligence. Atlantis Press (2010)
8. Koenig, N., Howard, A.: Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566). vol. 3, pp. 2149–2154. IEEE (2004)
9. Konda, V.R., Tsitsiklis, J.N.: Actor-Critic algorithms. In: Advances in neural information processing systems. pp. 1008–1014 (2000)
10. Levesque, H., Davis, E., Morgenstern, L.: The Winograd schema challenge. In: Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning (2012)
11. Li, Y.: Deep reinforcement learning: An overview. ArXiv preprint ArXiv:1701.07274 (2017)
12. Martınez-Plumed, F., Hernández-Orallo, J.: Ai results for the atari 2600 games: difficulty and discrimination using irt. EGPAI, Workshop on Evaluating General-Purpose Artificial Intelligence **33** (2016)
13. Oppy, G., Dowe, D.: The turing test. In: Stanford Encyclopedia of Philosophy, pp. 519–539 (2003)
14. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source Robot Operating System. In: ICRA workshop on open source software. vol. 3, p. 5. Kobe, Japan (2009)
15. Riedl, M.O.: The Lovelace 2.0 test of artificial creativity and intelligence. ArXiv preprint ArXiv:1410.6142 (2014)
16. Russell, S.J., Norvig, P.: Artificial intelligence: A modern approach. Malaysia; Pearson Education Limited, (2016)
17. Świechowski, M., Park, H., Mańdziuk, J., Kim, K.J.: Recent advances in general game playing. The Scientific World Journal **2015** (2015)
18. Thorarensen, T.: FraMoTEC: A Framework for Modular Task-Environment Construction for Evaluating Adaptive Control Systems. M.Sc. thesis, Department of Computer Science, Reykjavik University (2016)
19. Thórisson, K.R., Bieger, J., Schiffel, S., Garrett, D.: Towards flexible task environments for comprehensive evaluation of artificial intelligent systems and automatic learners. In: International Conference on Artificial General Intelligence. pp. 187–196. Springer (2015)
20. Thórisson, K.R., Bieger, J., Thorarensen, T., Sigurdardóttir, J.S., Steunebrink, B.R.: Why artificial intelligence needs a task theory—and what it might look like. In: International Conference on Artificial General Intelligence. pp. 118–128 (2016)
21. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double Q-learning. In: Thirtieth AAAI conference on artificial intelligence (2016)
22. Wang, P.: Rigid flexibility: The logic of intelligence. Springer Science & Business Media (2006)