SAGE: Task-Environment Platform for Autonomy and Generality Evaluation

Leonard M Eberding[1,2], Arash Sheikhlar[1], and Kristinn R. Thórisson[1,3]

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

Abstract. While several tools for training and evaluating narrow machine learning (ML) algorithms exist, 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 if any specific tools exist. In this paper we introduce a new methodology for evaluating the *autonomy* and *generality* of artificial intelligence (AI) and ML architectures, and a new tool that builds on this methodology. The tool and methodology platform are called *SAGE* (Simulator for Autonomy & Generality Evaluation), which works for training and evaluation of both kinds of systems as well as for detailed comparison between narrow and general ML and AI. It provides a variety of task construction and tuning options, allowing isolation of single parameters of different complexity dimensions. SAGE is aimed at helping AI researchers map out – and compare – strengths and weaknesses between divergent AI and ML approaches. Our hope is that it can help deepen understanding of the various tasks we want AI systems to do, including the relationship between their composition, complexity, and difficulty for various AI systems, as well as contribute to building a clearer road map for the field. We discuss the reasons why we think both narrow and general AI systems are in equal need of better tools and evaluation methodologies, describe the requirements that lead to the platform's creation and give examples of use.

Keywords: Evaluation · Generality · Autonomy · Task-environments

1 Introduction

Many good reasons exist for wanting proper evaluation methods for machines capable of complex tasks [3], 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 depend on the evaluation's purpose, whether it's the ability to learn a particular task or many, to learn quickly, reliably, to learn complex things, causal relations, or to handle novelty. Most current proposals for evaluating artificial intelligence (AI) systems focus on subsets of the spectrum of abilities that may be necessary to evaluate in general machine intelligence (GMI), or are too narrowly focused on particular tasks or domains.

In mathematics something is considered "general" which can be applied to a multitude of problems, whilst in physics a model of an observed entity is more general than another if it explains the entity and its behaviour better. In the second sense of "generality", a GMI is a system that can autonomously identify relationships and descriptive parameters of many aspects of its environment and use them in order to achieve goals. This means identifying and using causal relations between observed variables, because without a valid model of causation the achievement of goals devolves to trial and error and cannot scale to complex task-environments.A system exhibits autonomy when it can use descriptive parameters of its environment to achieve a goal without outside help. From the combination of the two we define *autonomous generality* as a system's ability to identify cause-and-effect chains of its environment and exploit them in order to reach a goal under the assumption of insufficient knowledge and resources [20]. Only a system which shows autonomous generality can be considered to be generally intelligent.

Good measuring tools and methodologies are key ingredients to assess progress towards autonomous generality. Especially if it allows comparison of different systems, not to mention of different kinds. The vast majority of evaluation methods proposed to date take a single measurement where a series of measurements could possibly much better separate between autonomous, general systems and narrow machine intelligence (NMI), giving much deeper insight into the nature of the chosen methodology. Here we introduce a new platform for intelligence evaluation that attempts to bridge the gap between low- and high-level intelligence and provide some methods for analyzing and constructing evaluation tasks in a granular way. SAGE is a task-environment simulation tool based around the idea of breaking tasks, and the environments they are performed in, into variables (observable, unobservable, manipulable, and non-manipulable) and transition functions that control their changes ([16,17]). Task-environments in SAGE may be constructed with a variety of characteristics and different levels of complexity, including causal and statistical relations, deterministic and non-deterministic behaviors, hidden or partially observable variables, distracting variables, noise, and much more. On this basis the gap between narrow and general intelligence may be bridged by varying the complexity and abstraction of tasks, increasing either relatively smoothly along one or more dimensions.

The design of SAGE is based on a new MVC-A (model-view-controller-agents) paradigm that enables an evaluator to change individual variables in the task-environment model, the controller, and the agent itself, independently of the others. The controller simulates the environment making adjustments to the internal workings of the simulation possible. The model provides further adjustability options for the observables and actions of the task. Lastly, the agent can be any machine learning system which can process the observable data of the model. By dividing these parts into different processes allows not only independent adjustments to be made, but the whole processes can be physically divided between processors or computers, connected via network protocols.

The paper is organized as follows: In section 2 we shortly discuss current task-theories with respect to their applicability in AI evaluation methodologies, past proposals for AI evaluation, and necessary properties of task-environments for generality and autonomy evaluation. Section 3 puts a focus on task-environment creation in SAGE and describes its implementation. In section 4 we show early results of evaluating generality and autonomy of two narrow AI systems. Section 5 discusses current status of the future possibilities offered by SAGE.

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 ([9]), Lovelace Test 2.0 [12], and the Toy Box Problem [6]. Others are too domain-specific, e.g. general game-playing ([14]), or highly dependent on knowledge of human social conventions or human experience and skills, e.g. Wozniak's Coffee Test and the Turing Test [10]. What is needed, as many have argued [1, 3, 5, 16], 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 11 dimensions that ideally should be controllable by a creator of a task-environment for measuring intelligent behaviour [16] ; Russell & Norvig (2016) present a somewhat comparable subset of seven dimensions [13]. The environment can be categorised along different dimensions, namely determinism (See [2] regarding the importance of noise control), staticism, observability, agency, knowledge, episodicity, and discreteness. TE properties include next to the seven environment properties ergodicity, asynchronicity, controllability, number of parallel causal chains, and periodicity [13, 16].

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 IRT it was shown that the difference of performance score between different ML techniques does not necessarily correlate with the systems level of abilities. Thus a simple performance rating like achieved game score cannot describe the progress of AI by itself [5]. 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.

In accordance with many working definitions of AI [20], an evaluation platform should provide the possibility to introduce novelty at any moment of the evaluation process. Such novelties could include for example unknown states, unknown transition functions, or unknown mechanics influencing the outcome of known state-action combinations. Without these novelties a proper evaluation of the learning – and thus a system's autonomous generality – cannot be performed [3]. Such changes in the task-environment can provide evidence of a system's pragmatic understanding of the causal relations between factors in the environment, arguably an important aspect of any GMI ([17]).

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 Task-Environment Creation in SAGE

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 papers ([3, 15, 16]) and is perhaps closest in spirit to Thorarensen's FraMoTEC environment (sans an emphasis on autonomy and generality).

In SAGE, assessing an AI system's ability to address novel things can be done by introducing new 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 learned variables. The response of a system to variable changes leads to conclusions about its ability to extract causal relations and its autonomy in exploiting them to achieve or hold the goal conditions.

3.1 Requirements

To apply to both narrow AI systems and GMIs, the following requirements were integrated in the SAGE evaluation platform:

1. The possibility to evaluate both NMIs and GMIs and make them comparable.
2. Easy generation of task-environments.
3. Possibility to include tasks ranging from low to high complexity.

4. Full 3D simulation of real-world scenarios (e.g. robotics) including sensors and actuators for high complexity tasks.
5. Inclusion and adjustability of complexity dimensions ([16]) before and during training.
6. Additional for GMI aspiring systems: physical division of evaluation platform and learner for evaluation of resource management.

If an AI system can provide evidence of being able to solve such environments one can further evaluate its capability to solve complex tasks by introducing more complex task-environment:

7. *Observability* - By changing the observability (even during run time) the systems adaptability to different, novel observations can be evaluated.
8. *Episodicness* - By introducing complex causal chains a sequential environment can be generated to evaluate the systems capabilities to extract causal relationships between past actions and current states.
9. *Number of causal chains* - Not only the episodicness is of interest, but also the degree of complexity in sequential environments. Parallel causal chains test for the ability to differentiate between correlation and causation.
10. *Agency* - Lastly by introducing a number of agents to the same task-environment the ability to interact with other agents and developing a goal oriented strategy for multi-agent systems can be evaluated. Other agents introduce the highest degree of unknown environments by changing their own strategies constantly while learning a task and its environment.

3.2 Task-Environment Properties

SAGE currently includes most of Thórisson's et al. 11 proposed controllable task-environment dimensions desired for evaluating AI systems [16]: Determinism, staticism, observability, episodicity, and discreteness can be adjusted both beforehand and during the training/learning/evaluation processes. It also provides reproducibility by using randomization seeds. Stochasticity can be adjusted in the observable variables, agent actions, and in environment dynamics. Staticism can be changed by either introducing different tasks, or changing environmental variables at runtime. Which variables can be observed by the learner/agent can be changed, as well as the variables available for manipulation. Lastly, the discreteness of observation and/ or action can be changed.

These adjustments make an evaluation of a learner's capability to cope with sensor noise, actuator impreciseness, and noise on hidden variables (e.g. wind forces) possible. Training on a variety of sensors before removing causally redundant ones tests the learner's capacity for knowledge generalization and causal relation extraction. The same holds for modifying controllability with which the system could exploit causal relations by applying previously unavailable actions to causally linked variables. Lastly, by changing the parameters during simulation without human interference, the autonomy of the agent can be evaluated.

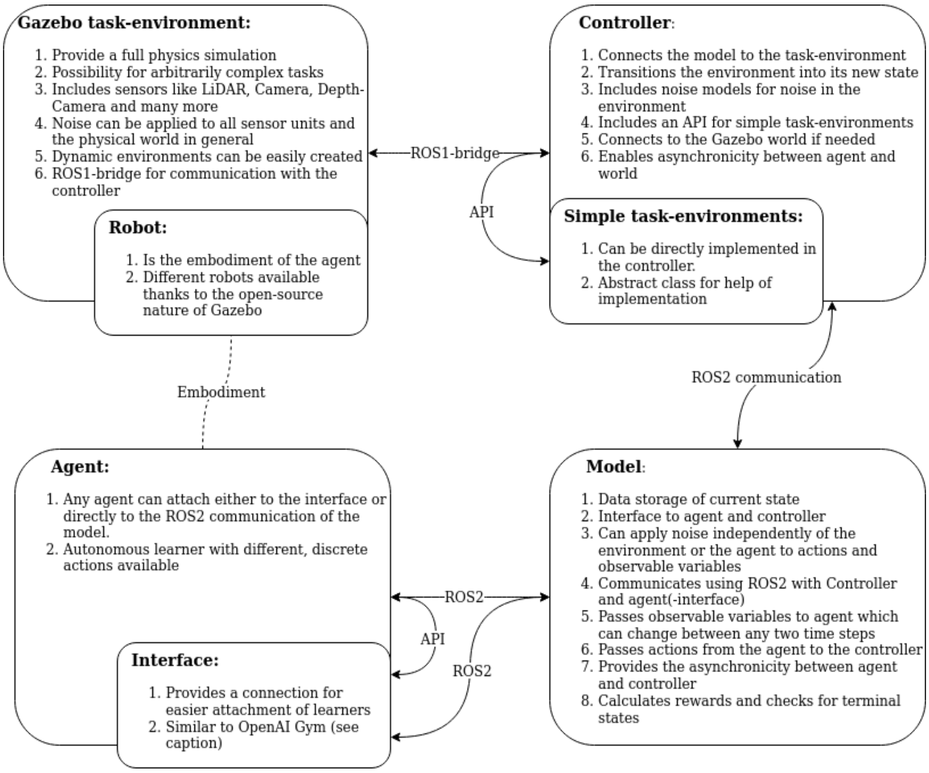

Fig. 1. Flowchart for illustration of the different components and their interactions. The controller provides the simulation of the task-environment either itself or by connecting to a Gazebo [7] world. The model node provides the system with a data storage and an environment independent noise system. It passes the actions from the agent to the controller and the current state from the controller to the agent. The agent is the learner to be tested and the interface gives the possibility to easily connect existing learners by providing a similar interface as OpenAI Gym [4].

3.3 MCV-A Approach

SAGE works using a Model-View-Controller-Agent (MCV-A) approach (see Fig. 1). Each part of the MVC-A architecture is implemented as a ROS2-node [11][1] using ROS2 for platform-independent inter-process communication. The current state of the environment including all observables, non-observables, manipulables, time, and energy is stored in the model. The model exposes all as observable defined variables via network communication to any attached agent through an interface module. The same module receives manipulables from the agent and passes them on to the model. In the interface noise and discretization models can be applied to the data independently from the rest of the simulation. The model

[1] https://index.ros.org/doc/ros2/ accessed on 26th of February 2020

communicates with the controller using network connection as well passing the current state of the environment including manipulables for further processing. The controller provides the simulation of the task-environment. Simple tasks can be easily added to the system as task modules, while the controller itself provides an interface to a Gazebo [7][2] simulation of a 3D world including a variety of robots, sensors, sensor-noise models etc. ROS2 as middle-ware between agent and evaluation platform makes the learners interface independent from the task-environment and therefore provides easy attachment of any MI 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 variable monitoring can be done using ROS2's internal rqt-graphs. The connection to the rqt-graph is again established using network communication enabling remote monitoring during evaluation.

This approach brings many advantages. First the previously described physical separation of agent and environment makes resource management evaluation possible. Further by dividing agent, model and controller into different processes real-time processing and asynchronous calculations can be easily added when needed. These assessment possibilities are especially important when GMIs are evaluated to fulfil the assumption of limited time and resources in the task environment ([19]).

Lastly, using the MVC-A approach gives an easy opportunity to increase the agency of the simulation. By using the network connection for communication any number of agents can communicate with the model interface simultaneously.

All adjustable parameters are wrapped in YAML-files making adjustments during run-time possible, while tasks are compiled and can be changed during evaluation.

4 Proof of Concept

Evaluating an actor-critic (AC) [8] as well as a double-deep-Q (DDQ) [18] learner on the cart-pole task ([4]) implemented in SAGE shows promising results regarding generality and autonomy evaluation (see Fig. 2).

1. The evaluation of influence of noise on the system for example shows the differences between environment noise (noise on dynamics of the inverted-pendulum) in comparison to noise on the observations or actions received/ given by the agent. Environment noise simulates noise outside the agent, observation noise simulates sensor noise and action noise simulates actuator imprecision, respectively.
2. Further the DDQ-learners capability to cope with hidden and random variables was evaluated. For this the velocity measurement was either turned of as an observable or was randomized with a standard deviation of 24 m/s (10x the usually occurring velocity values). This test shows the influence of random variables into the learning of both AC- and DDQ-learner.

[2] http://gazebosim.org/ accessed on 26th of February 2020

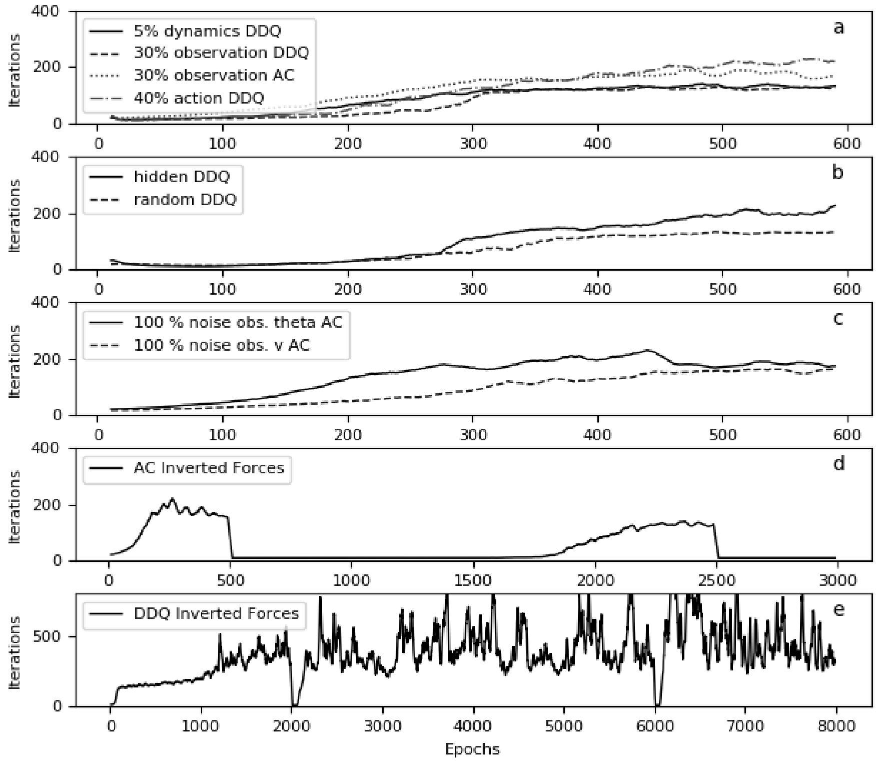

Fig. 2. Evaluation of an Actor-Critc (AC) and a Double-Deep-Q (DDQ) learner. All results are the average over 40 trials plotted with a running mean with window-size 10. **a**: different applications of noise on the two learners. Noise on environment dynamics (3%), noise on the observation (30%), and 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 onlz on single variables of the observation. Percentage definition see **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.

3. The influence of noise on only one of two different variables was evaluated to assess the importance of the correctness of this value.
4. Lastly we inverted the action direction after training of the learners to evaluate their generality of knowledge. 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.

These tests provide the research community with new insights into the methodologies of both the learners and current evaluation strategies. While noise changes in the observation or the actions make an assessment of learning with noisy data possible changes like inversion or hiding of variables make a generality and autonomy evaluation possible. An agent which cannot reach a similar performance with a randomised variable, as with this variable hidden from observation cannot be regarded as a general agent. When generalizing knowledge any random variable should be excluded from future decision making in order to create an expectable 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) shows that not cause-effect-chains were extracted but rather a simple state to action mapping took place. Otherwise this general knowledge could be used in order to find similarities between the source and the target task.

5 Conclusions & Future Work

SAGE shows high potential for evaluating generality and autonomy of AI architectures. By bridging the gap between general and narrow AI SAGE offers a platform to continuously assess the progress of AI research towards GMI. First results of NMI evaluation shows the possibilities of this platform. Adjustment of variables in tasks can not only lead to a better understanding of the AI system, its flaws and its advantages, but also to a better understanding of the task-environment itself. Thus the evaluation results can help identifying research gaps in both AI and task-theory.
General learners have not been evaluated, yet, due to the problem, that a) only few exist, and b) the setting up of the learner to being able to attach to the platform is a non-trivial, time consuming problem. However, we are confident that due to the various possibilities of TE adjustments an evaluation of a general learner and a multidimensional comparison with current learners is possible.
After implementing automated tests suits to collect a range of performance data, the creation of a multidimensional generality and autonomy collection of different NMI and GMI systems is planned. This makes the identification of important elements for GMIs possible and gives the AI research community a better insight into the complexity and therefore difficulty describing parameters of tasks.

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 learningenvironment: An evaluation platform for general agents(extended abstract). In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI). p. 4148–4152 (2015)
3. Bieger, J., Thórisson, K.R., Steunebrink, B.R., Thorarensen, T., Sigurardóttir, J.S.: Evaluation of general-purpose artificial intelligence: why, what & how. Evaluating General-Purpose AI (2016)
4. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: Openai gym. arXiv preprint arXiv:1606.01540 (2016)
5. 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)
6. Johnston, B.: The toy box problem (and a preliminary solution). In: Conference on Artificial General Intelligence. Atlantis Press (2010)
7. 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)
8. Konda, V.R., Tsitsiklis, J.N.: Actor-critic algorithms. In: Advances in neural information processing systems. pp. 1008–1014 (2000)
9. Levesque, H., Davis, E., Morgenstern, L.: The winograd schema challenge. In: Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning (2012)
10. Oppy, G., Dowe, D.: The turing test (2003)
11. 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)
12. Riedl, M.O.: The lovelace 2.0 test of artificial creativity and intelligence. arXiv preprint arXiv:1410.6142 (2014)
13. Russell, S.J., Norvig, P.: Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited, (2016)
14. Świechowski, M., Park, H., Mańdziuk, J., Kim, K.J.: Recent advances in general game playing. The Scientific World Journal **2015** (2015)
15. Thorarensen, .: FraMoTEC: A Framework for Modular Task-Environment Construction for Evaluating Adaptive Control Systems. Ph.D. thesis (2016)
16. 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)
17. Thórisson, K.R., Bieger, J., Thorarensen, T., Sigurardóttir, J.S., Steunebrink, B.R.: Why artificial intelligence needs a task theory. In: International Conference on Artificial General Intelligence. pp. 118–128. Springer (2016)
18. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: Thirtieth AAAI conference on artificial intelligence (2016)
19. Wang, P.: The logic of intelligence. In: Artificial general intelligence, pp. 31–62. Springer (2007)
20. Wang, P.: On defining artificial intelligence. Journal of Artificial General Intelligence **10**(2), 1–37 (2019)