

Artificial Open World for Evaluating AGI: a Conceptual Design

Bowen Xu^{1,2}[0000–0002–9475–9434] and Quansheng Ren^{1,2}[0000–0001–8698–9603]

¹ School of Electronics, Peking University, Beijing, 100871, China

² {xubowen, qsren}@pku.edu.cn

Abstract. How to evaluate Artificial General Intelligence (AGI) is a critical problem that is discussed and unsolved for a long period. In the research of narrow AI, this seems not a severe problem, since researchers in that field focus on some specific problems as well as one or some aspects of cognition, and the criteria for evaluation are explicitly defined. By contrast, an AGI agent should solve problems that are never-encountered by both agents and developers. However, once a developer tests and debugs the agent with a problem, the never-encountered problem becomes the encountered problem, as a result, the problem is solved by the developers to some extent, exploiting their experience, rather than the agents. This conflict, as we call *the trap of developers’ experience*, leads to that this kind of problems is probably hard to become an acknowledged criterion. In this paper, we propose an evaluation method named Artificial Open World, aiming to jump out of the trap. The intuition is that most of the experience in the actual world should not be necessary to be applied to the artificial world, and the world should be open in some sense, such that developers are unable to perceive the world and solve problems by themselves before testing, though after that they are allowed to check all the data. The world is generated in a similar way as the actual world, and a general form of problems is proposed. A metric is proposed aiming to quantify the progress of research. This paper describes the conceptual design of the Artificial Open World, though the formalization and the implementation are left to the future.

Keywords: Evaluation · Artificial Open World · Artificial General Intelligence

1 Introduction

In AGI research, how to evaluate AGI is a critical problem. In “narrow AI [8,20]”, evaluation seems not a severe problem, since in that field the criteria are explicit, for example, in the field of Image Recognition, researchers aim to rise up the accuracy of classification and use any tricks to solve that problem. Few may deny that datasets, as problems for evaluation, play an important role in the rapid progress of narrow AI. However, in AGI research, it is quite a different story on evaluation. Despite different definitions, goals, and pathways of AGI [20], under the perspectives of intelligence in Sec. 2, we hold that an AGI agent should

solve problems that are unknown to both agents and developers. However, once a developer tests and debugs the agent with a problem, the unknown problem becomes a known problem, as a result, that problem is no longer suitable for evaluating AGI agents – the developers are able to construct a problem-specific system that could not be applied to other situations, and the performance of a system in this problem does not reflect the progress on AGI. We call this trouble *the trap of developers’ experience*. To deal with this trouble, an alternative is to design new problems constantly [5], though we adopt a different path to jump out of the trap in this paper, *i.e.*, designing an artificial world. The Artificial Open World is generated in a similar way as the actual world, currently based on a classical world-outlook. The world should be open, in the sense that the causations in the world are time-varying on some abstract level, and problems to be solved are continuously changing. Implicitly infinite instances of the world can be generated so that for any of the instances, developers are possibly unable to perceive the world and solve problems by themselves based on their experience of the actual world. Nevertheless, after testing, developers are allowed to check all the data and analyze the activities of agents, and then perceive the instance of the world. The developers’ knowledge of one instance of the world is not necessary to be applied to another instance, such that facing a new instance, an agent has to solve problems by exploiting its own intelligence. The world should be generated in a similar way as the actual world, so that the knowledge of the generation is allowed to be known by agents in advance, because the agent with the knowledge would be still able to adapt to the actual world, without being disturbed by problem-specific knowledge from developers. To quantify the progress of AGI research, a metric is also proposed. We consider three aspects of performance, *i.e.*, the speediness of adaptation, the goodness of adaptation, and the goodness of generalization (see Sec. 3.3), and they should be merged together into one value, as the measure of intelligence. It should be noted that the value is a lower-bound of intelligence, and complicated situations partially stem from the competition between different agents in the world.

2 What Intelligence is

Before proposing the evaluation method, in this section, we should first figure out what *that thing* which is called *intelligence* is. We are not trying to propose a definition of intelligence within a brief sentence, but we are trying to describe our perspectives on that thing which is called intelligence.

Different perspectives on intelligence lead to different work. If one regards intelligence as the ability to solve complex problems, he or she would specify a sufficiently complex problem to be solved by a machine [2, 14]. If one treats intelligence as a set of cognitive functions, he or she would model human cognition with a cognitive architecture [9] or would let machines have capabilities that are presented in human beings, such as image recognition, natural language processing, *etc.* However, an agent, which possesses that thing which is called intelligence, should not merely solve several specified problems, no matter how

complex they are, and should not have only parts of the capabilities of human cognition. Therefore, to distinguish the goal of creating a general-purpose system and the specific methods of solving specified problems, the term *AGI (Artificial General Intelligence)* is invented [8]. An AGI agent should own that thing which is called intelligence. What is that thing after all?

The definitions of intelligence is discussed by a lot of predecessors (*e.g.*, [6, 11, 19]). Among the definitions, Pei Wang’s grasped some essential aspects of intelligence. In Wang’s definition, “*Intelligence is the capacity of an information-processing system to adapt to its environment while operating with insufficient knowledge and resources*” [16, 19], where *insufficient knowledge and resources* means being *finite*, being *open*, and working in *real-time*. Being *finite* means a system has insufficient spatial resources to store information and insufficient time to process information. As an intuition, an algorithm which searches exhaustively an answer, which is stored in an infinite memory, is not of intelligence. In this sense, insufficiency is critical. Being *open*, in Wang’s theory, means the content of tasks should not be specified before the system has been developed. Working in *real-time* means multiple tasks may occur in the same time, and one task may interrupt another. Adaptation in the definition refers to “the mechanism for a system to summarize its past experience to predict the future situations accordingly, and to allocate its bounded resources to meet the unbounded demands”. In Pei Wang’s theory [18], the constraints of insufficient knowledge and resources have been placed at the forefront, though they are obvious in human beings’ and machines’ lives.

François Chollet proposed the “generalization spectrum” – *absence of generalization, local generalization, broad generalization, and extreme generalization* – and use the word intelligence to refer to the *extreme generalization* [3]. An agent, *e.g.* a sorting algorithm, with absence of generalization can only handle those situations with no uncertainty. An agent, *e.g.* current machine learning systems, with local generalization, should handle a single task or a few tasks, which are well scoped by developers. An agent, with broad generalization, should generalize to unknown unknowns across a broad category of related tasks, for example, an image classifier could recognize dog while it is trained with cat images. An agent, with intelligence, as Chollet considered, should generalize to unknown unknowns across an unknown range of tasks and domains. Chollet may presuppose implicitly that unknowns and knowns have similarity on some abstract level, and an agent who is able to identify that kind of similarity is of intelligence. We generally agree Chollet’s view that an agent with intelligence should adapt to “unknown unknowns across an unknown range of tasks and domains”, though the meaning of “adapt” here may not be the same as that in Chollet’s definition, the meaning of which we approve is closer to that in Wang’s.

As our position, we hold that intelligence is a unity, which implies that it is a whole which can be described from different points of view. From one perspective, intelligence is a property with which an agent is able to deal with tasks in an open environment with limited resources. From another perspective, intelligence is an object which involves principles of representation-interaction. Informally,

an environment is open, which means that causations in the environment is time-varying to some extent.

As a further illustration, facing with the open environment, on one hand, an agent with intelligence should generalize to unknowns scenes, which means that, facing problems which are not encountered before, the agent should take reasonable solutions based on its past experience. The agent have to use the similarity on some abstract level to deal with the unknowns. On the other hand, after encountering a series of similar problems, which are expected to be solved by the agent, it should adapt to the problems as quickly as possible and performs as well as possible. Further more, as an explicit claim, the agent should be able to match a special-purpose system designed for specific tasks without losing the ability to adapt to new problems. Intelligence is the thing which facilitates an agent to meet the requirements mentioned above.

3 Evaluation

It is merely impossible to exhaustively review plenty of proposals and work on evaluating intelligence in this paper. Nevertheless, we briefly review some pieces of work and then propose our solution. A typical sort of evaluation is similar to I-athlon (Olympic Decathlon of Intelligence) [1]: a series of cognitive tasks are defined to test different capabilities. Broadly speaking, that evaluation method seems to assume that the more tasks an agent can fulfill and the better the agent performs in a task, the more intelligent the agent is. Some work focuses on the difficulty of problems and designs some puzzles to be solved by agents, *e.g.*, the “Bongard problem” [10]. To evaluate cognitive architecture, some metrics are proposed, *e.g.* [22], and we agree with some of them, especially the metric “taskability”, which is the ability to adapt to new tasks. To evaluate human-level AGI, Goertzel and Bugaj proposed to build a school environment and educate agents in it, and whether an agent has some skills, *e.g.*, logical-math, music, story understanding, *etc.*, determines the extent of intelligence [7]. Regardless of the feasibility in practice, there is a more severe problem: as Wang pointed out,

Though such activities do stimulate interesting research, it still has the danger of leading the research to problem-specific solutions, no matter how carefully the problems are selected — after all, this was why problems like theorem proving and game playing were selected in the early days of AI, and the resulting techniques have not been generalized to other fields very well. [17]

3.1 The Trap of Developer’s Experience

Those of AGI evaluation also encountered the same trouble as those work on evaluating narrow AI, *e.g.* datasets such as ImageNet [4], games such as Chess [2] and Go [14], *etc.*: developers may solve the problem and exploit their problem-specific knowledge, using any tricks, to program an agent. The problems for

evaluation are hard to avoid this kind of cheating, such that a problem-specific method performs better than a general system, which makes the problems unsuitable to evaluate a general system.

Even if at first a problem is not permitted to be seen by developers, after testing an agent, the problem should be presented to the developers for further analysis, otherwise, this kind of problems is almost not suitable for advancing the research. As thus, the unknown problem becomes a known problem, and developers’ experience on the specific problems would inevitably impact their designing the model of intelligence.

The agent with intelligence is necessary to adapt to an open environment, as we claim in Sec. 2. The environment could be complex or simple, actual or artificial, however, openness plays a critical role. The environment human faces is an actual, complex, and open one. The environment AlphaGo [14] faces is an artificial, simple, and closed one. The environment of ImageNet [4] is an actual, complex, and closed one. If the environment is closed, which means that the problems can be one by one solved by human developers, it is almost inevitable for developers to introduce their problem-specific experience to the machine. Eventually, it is not a machine but a human who solves problems.

This trouble, which is the reason why the traditional problems have the danger of leading the research to problem-specific solutions, is what we call *the trap of developer’s experience*.

What we need is an admitted criterion, which could be used to compare different AGI agents within a relatively long period. To jump out of *the trap of developers’ experience*, we first consider some overall principles of designing the evaluation method and then give some more detailed description of our proposal, the *Artificial Open World*.

3.2 Overall Principles

An AGI agent is required to be adaptive when faced with various problem in an open environment, and to find reasonable solutions without adaptation facing with new circumstances; simultaneously, for a specific problem, the agent should perform well with sufficient training, while it is still able to adapt to other problems and environments. Therefore, we should test how fast and how well an agent adapt to new environments and how well the agent generalize to new environments which are similar to the past.

Further, we suggest several criteria, for designing the Artificial Open World, that an AGI test should follow:

- (1) **Independence.** The test should be abstract and independent of the actual world, which means that developers’ experience of the actual world is not necessary to be applied to the artificial world. When solving problems in such a world, there are no problem-specific priors of developers, because the developers and the agents live in two worlds independent of each other – for example, knowledge of vision in the actual world does not have to be true in the artificial world.

- (2) **Similarity.** The artificial world is similar with the actual world in the process of generation, *i.e.*, the actual world is similar to one instance of the artificial world. If an agent performs well in the artificial world, it will be adaptive not only to our humans' actual world, but also to any worlds which has common natures in some sense. The knowledge about the generation is permitted to be priori knowledge of developers, since even though a developer convert this knowledge into a skill of an agent, the agent is still able to adapt to the actual world.
- (3) **Openness.** The world should be open in the sense that causations in the world are time-varying to some extent, and *new* problems can be generated continuously.
- (4) **Asymmetry.** To generate the world is easy, but to conjecture directly the parameters or structures of the world inversely should be hard or even impossible, so that developers cannot use the artificial-world-specific algorithm to acquire knowledge, which is only applied to one instance of the world.

3.3 Conceptual Design of Artificial Open World

Generation. There are three steps to generating the world. The first step is *differentiation*. As shown in Fig. 1a, two different kinds of *entities* are generated: one is positive, and another is negative. A number of entities are generated in the world, and the basic property of an entity is its spatial position. The second step is generating *causations*, as shown in Fig. 1b and Fig. 1c. Every two entities interact with each other, and several entities combine together as a whole, the

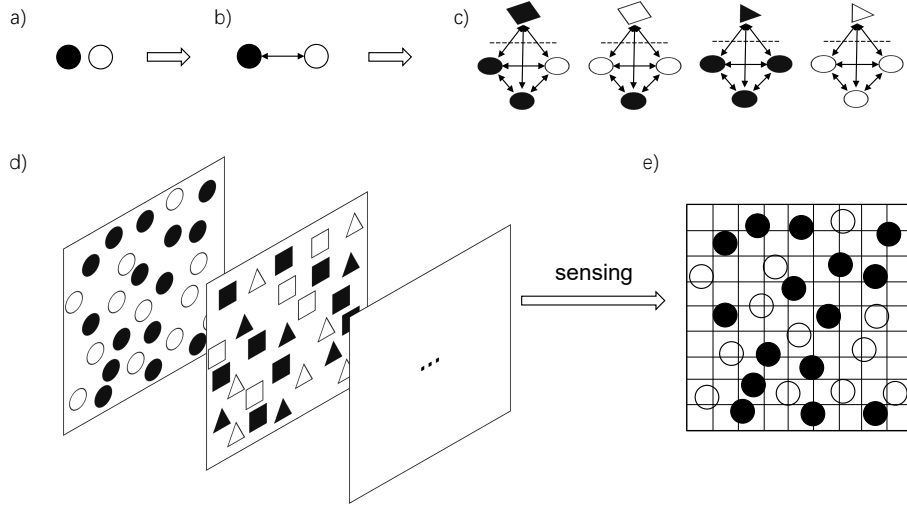


Fig. 1: The generation process of the Artificial Open World.

whole as an entity interacts with others. A relation of the interaction is called a *causation*. Through the combination, the world is hierarchical, as shown in Fig. 1d. The third step is to import the mind. The entities with a mind constitute an agent, and the entities themselves constitute the body of the mind. There is also a set of causations, as an interface, between the mind and the body. The world without the mind is mechanical, rigid, and inanimate, however, the mind makes the world complex and vibrant – just as in a board of Go, players’ mind leads to various complex situations.

The causations should be generated in some way. For example, the causation between two entities were a second-order differential equation, and the coefficients in the equation were randomly generated; further, the equation were not necessary to be a second-order differential equation, and the form of the equation were randomly generated. The causations do not have to be the same as those in the actual world so that the developers’ experience of the actual world is almost unsuitable to the generated world.

Some of the causations are stable, while some of the causations are time-varying. For example, in the hierarchical structures shown in Fig. 1d, the causations in lower levels are fixed, and the ones in higher levels are continuously changing. This is similar to the actual world: the dynamics of microscopic particles are stable, while the weather of an area is changing.

Furthermore, it should be noted that the complexity of the world partially stems from the intelligent agents, though there are some basic rules of the world. For example, in the game of Go, the two agents, as players, lead to complex situations, though the basic rules of Go are simple. As the development of the Artificial Open World with agents, the environments would evolve more and more complex; some of the evolved environments could be used as benchmarks, so that agents could adapt to them and be evaluated.

Mind-Body Interface. Here, we use the term *mind* to refer to *intelligence*, though they are not completely equivalent in some sense [23]. To enable an agent with the mind act and solve problems in the world, there are two kinds of mind-body interfaces, *i.e.*, sensor and motor. The former is a causation where the cause is the entities outside the mind, and the latter is a causation where the cause is the mind. Through sensor, the mind can sense the basic entities (in Fig. 1a), however, due to the limitation of resources, the data sensed is a projection of the entities, with a certain resolution, as shown in Fig. 1e. For example, the retina cannot sense every atom accurately but can sense the environment with a certain resolution. Through motor, the mind can affect spatial positions of the body, which is a set of entities, and through further interactions between entities, the mind can affect a broader range of the world.

At the current stage, there are two considerations on the body. One is that the mind has a fixed body, which would not be destroyed by the environment so that the mind can survive in the world to solve problems for further evaluation. The other is that the body is evolved and could be destroyed, and one goal of the mind is to maintain the existence of the body.

Problems. To measure the intelligence, we should define the problems to be solved in the world. However, once a specific problem is defined, developers would solve the problem and put the skills into a machine, as a result, it is not machines but developers who solve problems. To avoid the *trap of the developers' experience*, we consider a general form of the problems.

The objective status of the entities at time t is denoted as s_t , and the target status at time t is denoted as s'_t . A problem is defined as the pair (s_t, s'_t) . To solve the problem is defined as to find a series of actions, which is denoted as a , so that s_t is evolved to s'_t . An agent is informed of the problem in a certain way and gave a score for solving the problem. The considerations for calculating the score are illustrated in the next sub-section *Metric*.

As thus, the implicitly countless problems could be generated, even if a developer debugs the program and checks how an agent solves a problem, the future encountered problems are not solved by the developer, and the developer's experience is not necessarily suitable for those cases.

Metric. To evaluate the adaptability of an agent, an intuition is that the agent should solve the problems with fewer observations and attempts, simultaneously, for a problem which are similar, on some abstract level, to those solved ones, an agent should solve the problem, to some extent, without attempts, *i.e.*, the agent should generalize its experience to the new problem.

Based on these considerations, there are some indicators to be measured. We denote the number of observations for an agent to solve a problem as O and the duration consumed as D . The indicators O and D are objective in the sense that they are independent of the implementation of agents. We denote the memory resources consumed for an agent to solve a problem as M and the calculation resources consumed as C . The indicators M and C are subjective in the sense that they depend on the implementation, *e.g.*, programming language, hardware, theoretical model, *etc.* Whenever an agent solves a problem, it obtains a score, denoted as S . The score S should be negatively related to O and D . The score S can be normalized by M and C so that different AGI models can be Relatively fairly compared.

Given the scores which varies with time, the time derivative of S , dS/dt , is calculated, and the typical curves are drawn in Fig. 2a. The derivative dS/dt reflects the performance of adaptation to some extent. The faster a curve rises up, the faster the agent adapts to a new circumstance, *i.e.* it reflects the speediness of adaptation; the higher a curve reach, the better the agent adapts to a particular circumstance, *i.e.* it reflects the goodness of adaptation. After the causations are changed, the performance of the agent to obtain the scores would draw-down, and the extent of it reflects the goodness of generalization. In some way, indicators α , which denotes the speediness of adaptation, β , which denotes the goodness of adaptation, and γ , which denotes the goodness of generalization, are calculated all based on dS/dt . Finally, there should be an overall metric of intelligence, $I = M(\alpha, \beta, \gamma)$, where M is a function to merge the three indicators into one value I .

We argue that the metric I is a lower-bound of the measure: an agent is voluntary in some sense, which means that it may choose to do nothing at all in the world, without presenting its wisdom. Nonetheless, in a test, to increase the lower-bound, developers are allowed to modify some parameters of their models, so that agents are proactive in solving problems. In this sense, the metric I provides evidence that an agent is of intelligence.

We argue that there would be two stages of evaluating AGI. At the first stage, an agent is tested in the artificial world without other agents participating in; thus, at this stage, the metric I is an absolute one, which only reflects the ability of understanding the world. At the second stage, multiple agents lives in the same world, and more complex phenomena would emerge. Agents would compete and cooperate with each other, and communicate with each other, when game behaviors and language might emerge; thus, at this stage, the metric I is a relative one, and those agents who is better at game, or has the capability of language, might obtain relatively higher I .

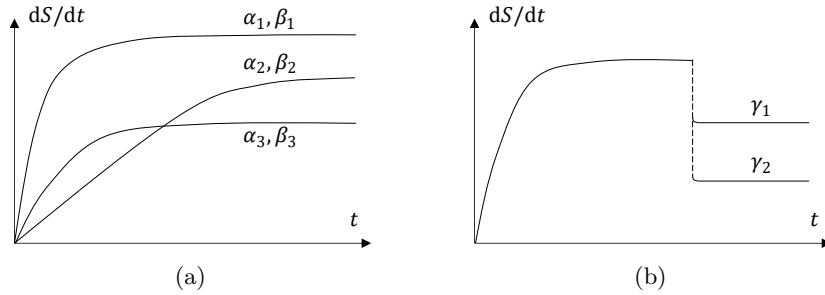


Fig. 2: Curves for evaluation. (a) Comparison on adaptation, where α indicates the speediness of adaptation, and β indicates the goodness of adaptation. In this figure, $\alpha_1 > \alpha_3 > \alpha_2$ and $\beta_1 > \beta_2 > \beta_3$. (b) Comparison on generalization, where γ indicates the goodness of generalization. In this figure, $\gamma_1 > \gamma_2$.

Future Work. We will formalize the description in Sec. 3.3 and implement the Artificial Open World in the future so that researchers can easily install the environment, test their agents, and compare their models with others' practically.

There are still some theoretical troubles of *Metric* in Sec. 3.3. For example, how to quantify the subjective indicators M and C in practice, and how to adjust the indicators O and D according to the difficulty of s_t reaching s'_t without agents' efforts.

The previous problems, including Game of Go, theorem proving, image recognition, natural language understanding, *etc.*, should be special cases of the problems in the artificial world, however, this deserves further justification. The issue of causation, which is an important concept in our design, is discussed for a long

time in philosophy [13], as well as in AGI [21], and the term causation should be further clarified.

It should be clarified that the current Artificial Open World is generated under a Classical world view. However, with the development of quantum computing, a world with quantum effects might be considered.

4 Discussion

The trap of developers’ experience is not a novel and fantastic idea, which may be realized by many researchers before, however, we put it in the first place when designing the evaluation method. It might be a misunderstanding that the trap could be an another name of “developers training in a test-dataset”. In fact, even if a developer has never seen the test-dataset, his experience on the training data can lead to falling into the trap. For example, when a developer is creating an agent to answer questions according to given images, the developer is possible to assign the format of the input and output data, as well as the loss function of a neural network, which are the developers’ experience on the problem, so that the agent cannot adapt to other situations, such as mastering the game of Go, without modifying any codes; If developers find that objects always locate in the center of images, which is another type of developers’ experience, they could make the agent always focus on the center by clipping the images, and this agent are possibly not able to adapt to those images where objects locate at arbitrary positions.

The Artificial Open World might be similar to an environment for multi-agent reinforcement learning at first glance, however, it is essentially different from traditional environments. Traditional reinforcement learning environments, such as *Melting Pot* [12], are faced with the trap of developers’ experience, since developers are possible to solve the predefined problems in the environments by themselves. Another similar evaluation environment is *XLand* [15], where the environment is generated somehow, and plenty of tasks are generated temporarily in *XLand*, however, the trouble of the trap above is still not avoided, as a developer is possible to research the patterns in each of the tasks. These problems designed for evaluation, just as those such as datasets of image recognition, are valuable in terms of guiding a researcher in creating his own model of intelligence if he keeps away from importing problem-specific experience cautiously, while they are not suitable to be treated as benchmarks or criteria around the community, whereas cheating is unavoidable to some extent. By contrast, the Artificial Open World aims to ensure that it is the agents, rather than their developers, who solves problems by exploiting their own intelligence.

An interesting issue is the logic in the Artificial Open World. The logic can be adaptive, which means that the logic rules and their truth functions are acquired through interactions with the world, but are not designed and fixed. The following illustration is under the context of Non-Axiomatic Logic [18]. For example, an acquired relation is represented as $\langle (*, T_1, T_2) \rightarrow R \rangle$, where R is the

relation term; a syllogistic rule can be

$$\begin{aligned} & \langle (\wedge, \langle (*, \$M, \$P) \rightarrow inheritance \rangle, \langle (*, \$S, \$M) \rightarrow inheritance \rangle) \\ & \Rightarrow \langle (*, \$S, \$P) \rightarrow inheritance \rangle \end{aligned} \quad (1)$$

Suppose that the truth value of the two premises $\langle (*, \$M, \$P) \rightarrow inheritance \rangle$ and $\langle (*, \$S, \$M) \rightarrow inheritance \rangle$ are (f_1, c_1) and (f_2, c_2) , and that the truth value of the conclusion $\langle (*, \$S, \$P) \rightarrow inheritance \rangle$ is (f, c) . The truth value (f, c) is determined by (f_1, c_1) and (f_2, c_2) through a function $F(f_1, c_1, f_2, c_2)$. The function $F(\cdot)$ is acquired via experience, rather than identified in advance. Further, the syllogistic rule (Eq. 1) does not have to be true and can be acquired via experience. The intriguing questions occur: will the agent in the artificial open world follow the same logic which is discovered in the actual world? Will the logics, which are learned by agents in different configurations of the world, be the same to some extent? Will the logics emerged be appropriate for the agent in the actual world? If the answers are “yes”, it will be quite strange that the logic seems a universal existence. If the answers are “no”, then the artificial open world puts forward a higher demand for researchers to design an adaptive logic.

5 Contributions&Acknowledgements

Bowen Xu proposes the main idea and writes this paper; Quansheng Ren, who reviews and modifies the paper, points out the key idea that the complexity of the world stems from agents’ behaviors. We thank Pei Wang for sharing some pieces of literature on evaluating AGI. We thank those who review this paper.

References

1. Adams, S.S., Banavar, G., Campbell, M.: I-athlon: Towards a multidimensional turing test. *AI Magazine* **37**(1), 78–84 (2016)
2. Campbell, M., Hoane Jr, A.J., Hsu, F.h.: Deep blue. *Artificial intelligence* **134**(1-2), 57–83 (2002)
3. Chollet, F.: On the measure of intelligence. arXiv preprint arXiv:1911.01547 (2019)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
5. Genesereth, M., Björnsson, Y.: The international general game playing competition. *AI Magazine* **34**(2), 107–107 (2013)
6. Goertzel, B.: Artificial general intelligence: concept, state of the art, and future prospects. *Journal of Artificial General Intelligence* **5**(1), 1 (2014)
7. Goertzel, B., Bugaj, S.V.: Agi preschool: a framework for evaluating early-stage human-like agis. In: *Proceedings of AGI*. vol. 9, pp. 31–36 (2009)
8. Goertzel, B., Pennachin, C.: *Artificial general intelligence*, vol. 2. Springer (2007)
9. Hart, D., Goertzel, B.: Opencog: A software framework for integrative artificial general intelligence. In: *AGI*. pp. 468–472 (2008)

10. Hofstadter, D.R.: Gödel, Escher, Bach: An Eternal Golden Braid. Basic Books, 20th anniversary edition edn. (1999)
11. Legg, S., Hutter, M., et al.: A collection of definitions of intelligence. *Frontiers in Artificial Intelligence and applications* **157**, 17 (2007)
12. Leibo, J.Z., Dueñez-Guzman, E.A., Vezhnevets, A., Agapiou, J.P., Sunehag, P., Koster, R., Matyas, J., Beattie, C., Mordatch, I., Graepel, T.: Scalable evaluation of multi-agent reinforcement learning with melting pot. In: *International Conference on Machine Learning*. pp. 6187–6199. PMLR (2021)
13. Schrenk, M.: *Metaphysics of science: A systematic and historical introduction*. Routledge (2016)
14. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al.: Mastering the game of go without human knowledge. *nature* **550**(7676), 354–359 (2017)
15. Team, O.E.L., Stooke, A., Mahajan, A., Barros, C., Deck, C., Bauer, J., Sygnowski, J., Trebacz, M., Jaderberg, M., Mathieu, M., et al.: Open-ended learning leads to generally capable agents. *arXiv preprint arXiv:2107.12808* (2021)
16. Wang, P.: *Non-axiomatic reasoning system: Exploring the essence of intelligence*. Ph.D. thesis, Indiana University (1995)
17. Wang, P.: The evaluation of agi systems. In: *Proceedings of the Third Conference on Artificial General Intelligence*. vol. 11, pp. 164–169. Citeseer (2010)
18. Wang, P.: *Non-axiomatic logic: A model of intelligent reasoning*. World Scientific (2013)
19. Wang, P.: On defining artificial intelligence. *Journal of Artificial General Intelligence* **11**(2), 73–86 (2020)
20. Wang, P., Goertzel, B.: *Theoretical foundations of artificial general intelligence*, vol. 4. Springer (2012)
21. Wang, P., Hammer, P.: Issues in temporal and causal inference. In: *International Conference on Artificial General Intelligence*. pp. 208–217. Springer (2015)
22. Wray, R., Lebiere, C.: Metrics for cognitive architecture evaluation. In: *Proceedings of the AAAI-07 Workshop on Evaluating Architectures for Intelligence*. pp. 60–66 (2007)
23. Xu, B., Zhan, X., Ren, Q.: The gap between intelligence and mind. In: *International Conference on Artificial General Intelligence*. pp. 292–305. Springer (2021)