Evaluation of General-Purpose Artificial Intelligence: Why, What & How

Jordi Bieger[1] **Kristinn R. Thórisson**[1,2] **Bas R. Steunebrink**[3]
Thröstur Thorarensen[1] and **Jóna S. Sigurðardóttir**[2]

Abstract. System evaluation allows an observer to obtain information about a system's behavior, and as such is a crucial aspect of any system research and design process. Evaluation in the field of artificial intelligence (AI) is mostly done by measuring a system's performance on a specialized task. This is appropriate for systems targeted at narrow tasks and domains, but not for evaluating general-purpose AI, which must be able to accomplish a wide range of tasks, including those not foreseen by the system's designers. Dealing with such novel situations requires general-purpose systems to be *adaptive*, learn and change over time, which evaluation based on quite different principles. The unique challenges this brings remain largely unaddressed to date, as most evaluation methods either focus on the binary assessment of whether some level of intelligence (e.g. human) has been reached, or performance on a test battery at a particular point in time. In this paper we describe a wide range of questions which we would like to see new evaluation methods for. We take look at various purposes for evaluation from the perspectives of different stakeholders (the *why*), consider the properties of adaptive systems that are to be measured (the *what*), and discuss some of the challenges for obtaining the desired information in practice (the *how*). While these questions largely still lack good answers, we nevertheless attempt to illustrate some issues that we believe are necessary (but perhaps not sufficient) to provide a strong foundation for evaluating general-purpose AI, and propose some ideas for directions in which such work could develop.

1 INTRODUCTION

Evaluation is the empirical means through which an observing system—an *evaluator*—obtains information about another system-under-test, by systematically observing its behavior. Evaluating general-purpose artificial intelligence (AI) is a challenge due to the combinatorial state explosion inherent in any system-environment interaction where both system and environment are complex. Furthermore, systems exhibiting some form of general intelligence must necessarily be highly adaptive and continuously learning (i.e. changing) in order to deal with new situations that may not have been foreseen during the system's design or implementation. Defining performance specifications for such systems is very different than doing so for systems whose behavior is not expected to change over time. Since the inception of the field of artificial intelligence (AI) the question of how to evaluate intelligence has puzzled researchers [27, 12, 10, 13, 23]. Many evaluation proposals to date have tried to transfer ideas from human testing [27, 7, 13], but this approach has severe limitations for artificial intelligence [2], where no single reference- or abstract system model can be assumed.

In this paper we discuss several important topics related to creating a solid foundation for evaluating (artificial) adaptive systems, organized around the three main topics of *why* we need special methods for evaluating adaptive systems, *what* should be measured when evaluating such systems, and *how* one might go about taking these measurements. For each one we highlight one or more topics that we see as critical yet unaddressed in the research literature so far. While having answers to many of the important questions raised in this paper would be desirable, we acknowledge that solutions remain out of reach to us, as much as our forerunners. For some we outline promising ways to address them, but for others we can only start by summarizing key issues and questions that must be answered in coming years (and decades).

Why might we want to evaluate adaptive systems? Numerous reasons could be cited, many of which will be shared by evaluation of other non-adaptive systems. Rather than try to be comprehensive in this respect we turn our attention here to three reasons for evaluating adaptive artificial systems that we feel are likely to lead to methods different from those developed for other kinds of systems: (a) testing whether performance levels in a particular range of areas are expected to be sufficient, (b) finding a system's strong and weak properties, and (c) establishing trust in a system by finding ways to predict its behavior. Different evaluators may wish to consider these aspects for various purposes: the system's designer may wish to find areas to improve, a teacher may wish to gauge training progress, a user may wish to deploy the system in situations where it will perform well, and potential adversaries may wish to exploit possible weaknesses. Understanding the relationship between task, environment, system, and evaluation methods is of critical importance as this will determine the appropriateness, efficiency, and meaningfulness with which any such measurements can be done.

What should be measured? Given that we are focusing on evaluation of general-purpose—and therefore adaptive—systems, it is somewhat surprising how research on this topic has tended to ignore its very central issue: *the adaptation process itself*. What adaptive systems have beyond other systems is that they change. Any proper test of adaptive systems must include a way to measure such *adaptivity*, including learning rate, knowledge retention, knowledge transfer, and sensitivity to interference, among other things. Yet most ideas on how to evaluate intelligent systems, starting with the Turing test and

[1] Center for Analysis and Design of Intelligent Agents,
School of Computer Science, Reykjavik University, Iceland.
email: {jordi13,thorisson,throstur11}@ru.is
[2] Icelandic Institute for Intelligent Machines, Reykjavik, Iceland.
email: jona@iiim.is
[3] The Swiss AI Lab IDSIA, USI & SUPSI, Manno-Lugano, Switzerland.
email: bas@idsia.ch

not changed much in character over the decades, has limited its scope to a measurement at a single point in time (see e.g. [13]).[4]

For the development of general—and beneficial—artificial intelligence, merely measuring current performance on a range of task-environments does not suffice. We must ascertain ourselves of the fact that our artificial adaptive system will be able to learn to deal with *novel* situations in a safe, beneficial and expedient way. Novelty calls for a kind of generality which to date remains to be successfully implemented in an artificial system. An ability that humans (and some animals) have and which seems central to general intelligence, and in particular important for novel environments and tasks, is *understanding*. Even within the field of artificial general intelligence (AGI) this special mechanism for adaptation seems to not have gotten the attention it deserves. It seems obvious that any proper evaluation method for intelligent systems must address understanding. In addition to performance under a variety of conditions, we must evaluate a system's robustness, learning/adaptation ability, and understanding of fundamental values. Unlike more specialized systems, where reliability in the specified range of situations suffices, we need to know that a general AI would adapt its behavior but not the core of its values in new situations.

So how can such things be measured? No consensus exists on what features adaptive systems should or must have, or what their purpose should be (nor can there be, since applications of such systems are countless): Looking for a single test, or even a standard battery of tests, for evaluating such systems is futile. Instead we argue that what is called for are a *task theory* [24] and a *test theory* [20], that would specify how construction of a *variety* of evaluation tests and methods can be done, as called for by the nature of the system to be evaluated and the aims of their developers.

In the remainder of this paper we will first discuss some background knowledge in section 2. Section 3, section 4 and section 5 will discuss the why, what and how of AI evaluation. Here we will consider the various purposes for which we might want to evaluate a system, identify various important fundamental and emergent properties of adaptive systems, and look at how we could obtain information about them. In section 6 we conclude the paper with a call for increased focus on the discussed areas of AI evaluation that have so far not received sufficient attention.

2 BACKGROUND

When we talk about intelligent systems under test, we can refer to either agents or controllers.[5] An agent consists of a (physical or virtual) body, containing its sensors and actuators, and a controller that acts as the "mind" of the system. In artificial intelligence research we are usually concerned with building ever more sophisticated controllers, while in robotics or applied AI we may also design the system's body. When we use the words "system", "actor" or "entity", we refer to whatever thing is being tested, whether that includes a body or not.

Intelligent systems interact with *task-environments*, which are tuples of a *task* and an *environment*. An *environment* contains objects that a system-under-test can interact with—which may form larger complex systems such as other intelligent agents—and rules that describes their behavior, interaction and affordances. A *state* is a configuration of these objects at some point or range in time. Tasks specify criteria for judging the desirability of states and whether or not they signify the successful or unsuccessful end of a task. The manner in which the task is communicated to the system-under-test is left open, and depends on the system and desired results of the evaluation. For instance, in (classical) AI planning the task is usually communicated to the system as a goal state at the start, while most reinforcement learners only get sporadic hints about what the task is through valuations of the current state.

The ultimate goal of evaluation is to obtain information about an intelligent system and its properties. This is done by observing its performance (behavior) as it interacts with a task-environment and/or the state that the task-environment is left in. For instance, we could evaluate a system just by the final score of a tennis match (of which evidence is left in the environment), or we could carefully analyze its behavior. Another example might be a multiple-choice exam, where we only look at the filled-out form at the end and don't consider the system's behavior over time. In a more elaborate written test, we may try to reproduce the system's thought process from the end result. Looking at final results is much easier, but also potentially much less informative as it throws out a lot of information.

Black-box evaluation methods look only at the input-output behavior of the system under test and its consequences, while white-box testing can also look at a system's internals. For fair and objective comparisons between different systems (e.g. humans and machines), black-box testing is typically desirable. Nevertheless, looking at gathered and utilized knowledge, or considering the performance of different modules separately can be quite informative—e.g.when debugging, finding weak points, or assessing understanding.

To define various properties of artificial systems to be measured, we must first have a decent understanding of the task-environments in which they are measured—preferably in the form of a *task theory* [24]. Task-environments—like intelligent systems—have both fundamental and emergent properties. For instance, the number of dimensions of a task-environment is an explicit (fundamental) part of its definition, whereas complexity emerges implicitly, and factors like observability and difficulty emerge in interaction with an intelligent system. We define the set of all task-environments to be $\mathbb{TE}$ and the set of properties to be $\mathbb{P}_{\mathbb{TE}} = \mathbb{L}_{\mathbb{TE}} \cup \mathbb{N}_{\mathbb{TE}}$, where $\mathbb{N}_{\mathbb{TE}}$ is for quantitative properties and $\mathbb{L}_{\mathbb{TE}}$ for qualitative ones[6]. A quantitative property $N \in \mathbb{N}_{\mathbb{TE}}$ is defined as a function from the set of all AI systems $\mathbb{A}$ and a collection of task environments to a real number: $N \in \mathbb{N}_{\mathbb{TE}} : \mathbb{A} \times \mathbb{TE}^n \to \mathbb{R}$. Collections of quantitative properties similarly map to a vector of real values: $\mathcal{N} \subset \mathbb{N}_{\mathbb{TE}} : \mathbb{A} \times \mathbb{TE}^n \to \mathbb{R}^n$. We define a distance metric $\mathcal{D} : \mathbb{TE} \times \mathbb{TE} \to [0, \infty)$, and $\mathcal{D}_{\mathcal{N} \subset \mathbb{N}_{\mathbb{TE}}}(X, Y) = f(\mathcal{N}(X), \mathcal{N}(Y))$, where $f : \mathbb{R}^n \times \mathbb{R}^n \to [0, \infty)$ can be any metric on $\mathbb{R}^n$; e.g. absolute/manhattan distance $f(\vec{x}, \vec{y}) = \sum_{i=0}^{|x|} |x_i - y_i|$

[4] Exceptions do of course exist, but given the importance of the subject, one would have expected the exact inverse ratio. See our earlier work on requirements for an evaluation framework for AI for a more in-depth discussion [23].

[5] As in control theory, we use the term "controller" to refer to control mechanisms in the broadest sense, irrespective of the methods they employ to achieve the control. An intelligent system's "controller" *includes anything that changes* during adaptation, such as memories, knowledge, know-how, reasoning processes, insight, foresight, etc., as well as the primary mechanisms instigating, managing and maintaining those changes. Any part of a system designated as belonging to its controller defines thus the *boundary* between *that which is being controlled* (e.g. a robot's body) and *that which does the controlling* (i.e. its "mind").

[6] Some examples of qualitative properties are the type of environment (e.g. is it a grid-based environment?), the nature of another agent (e.g. is it a friend, teacher, rival, etc.?), or the presence of particular phenomena (e.g. does it involve arithmetic?). In this paper we focus on quantitative aspects of evaluation however.

or Euclidean distance $\sqrt{\sum_{i=0}^{|x|}(x_i - y_i)^2}$. We similarly define the properties of adaptive systems $\mathbb{P}_{\mathbb{A}} = \mathbb{L}_{\mathbb{A}} \cup \mathbb{N}_{\mathbb{A}}$.

One defining aspect of AGI-aspiring systems is that they must *adapt* or *learn*: their knowledge and the behavior that follows from it change over time to better handle previously unknown situations. Here we take a very broad definition of knowledge that includes declarative knowledge (beliefs), procedural knowledge (skills), and structural knowledge (priorities). While the line between the core of a system and its (more fluid) knowledge can be blurry, it is occasionally useful to consider them separately. We define $\mathcal{K}(A)$ to be the knowledge of adaptive system $A \in \mathbb{A}$, and $\mathcal{K}(\mathcal{TE}, A, K_0, t_0 : t_n)$ to be the knowledge that system A with starting knowledge K_0 acquires/acquired in task-environments $\mathcal{TE} \subset \mathbb{TE}$ between times t_0 and t_n. An equivalent but alternative view is that K contains all of A's cognitive aspects that can change over time while A (also) contains its constants (e.g. its identity).

3 THE *WHY:* PURPOSES OF EVALUATION

Evaluation at its core is about obtaining information about the intelligent system-under-test. There are a number of reasons for why one might like to evaluate such a system. Evaluation can also be done by entities with a wide variety of relations to the system. Its developers may wish to improve its design, users may want to know what it can do (in which situations), teachers are interested in current knowledge levels and supported learning methods [4], and potential rivals may wish to size up the opposition. These parties have varying levels of control over the system under test and the evaluation process itself, and will need to take those limitations into account. Evaluations can be done in the lab, where the evaluators have full control over the task-environment and the system under test and can reset and tweak it at will, or they can be performed in the wild: e.g. by other agents who wish to interact with the system-under-test in some way.

3.1 Task-specific Performance

An often asked question—by consumers and creators alike—is whether a certain device is capable of performing a particular task that they need done, and if so, how well. Many AI systems are developed for a single, specific purpose and can often be evaluated in a relatively straightforward fashion. Performance is defined by the task at hand, and task-specific knowledge can be used to devise a model of the task-environment and/or select a collection of representative situations in which the system is to be tested.

Such evaluations are suitable in cases where the variation in the task-environment is well-known or can be controlled to a sufficient degree, and no real adaptation is required. This is typically not the situation in AGI research, where intelligent systems must be able to handle a wide range of tasks, both known and unknown at system design (and test) time. Nevertheless, even an AGI system may on occasion want to learn a particular task (in addition to the other tasks it already knows), in which case evaluation of task-specific performance could be appropriate for e.g. measuring training progress.

3.2 Strengths and Weaknesses

General cognitive abilities—and most generally *intelligence*—are used across a wide variety of tasks. Examples include the ability to reason by analogy, learn from examples, perform induction/abduction/deduction, respond in real-time, remember recent or long-ago events, understand causal chains, ignore distractions, etc. Knowledge of the levels of various cognitive abilities provides information about the system's strengths and weaknesses. This is useful for a variety of reasons:

- It points the system's developers to areas that need improvement.
- It can help users determine whether the system is suitable for (a range of) tasks or environments.
- It can help a teacher or friend find methods for education/interaction that the system will respond well to.
- It can help a potential adversary select strategies that avoid strengths and exploit weaknesses.

Depending on the evaluator's role, there are various amounts of control that they can exert over the system and the evaluation process.

In AI research we are mainly interested in *cognitive* capabilities, but generally speaking evaluation can also be used to test more physical capabilities.

3.3 Trust and Predictability

AGI systems are built to be deployed in situations that are not yet fully known when the system is designed and tested. Nevertheless, we would like to ensure that it behaves acceptably. To know this, we need to evaluate the range of situations in which it will behave according to specification. We can try to limit the system's exposure to situations that fit these parameters. When this is not possible or desirable, we want to know that the system degrades gracefully in difficult and/or novel situations and ideally that it will adapt to them and learn to perform well again over time. We also want to ensure that the system understands what "perform well" and "good outcome" mean, even in completely new situations.

For the development of general, beneficial artificial intelligence merely measuring performance on a range of task-environments does not suffice [20]. We must ascertain ourselves of the fact that it will be able to learn to deal with novel situations in a safe and beneficial way. To do so, we must evaluate robustness, learning/adaptation ability and understanding of fundamental values, as well as performance under various conditions. Unlike more specialized systems, where reliability in the specified range of situations suffices, we need to know that a general AI would adapt its *behavior* but not the core of its *values* in new situations [5]. To ensure good outcomes in an unpredictable, large, and complex (real) world, we need to look at a system's robustness, adaptivity and understanding.

4 THE *WHAT:* PROPERTIES OF INTELLIGENT SYSTEMS

After having identified different purposes of evaluation we can now turn to the various properties of artificial systems about which we may desire information. Some properties—such as the nature of the system's learning algorithms or its motivational system—are inherent in the system's design and may be amenable to inspection of its implementation, while other properties—such as the performance on a certain task or the amount of knowledge necessary to learn something—are emergent from the interaction with the world and are more amenable to evaluation. Although tests for qualitative aspects of a completely black-box system could in principle be designed, we focus here on the evaluation of quantitatively measurable properties.

4.1 Performance

For virtually all quantitative evaluations, some kind of *performance* measure is used as the main dependent variable. Too often however, all focus is placed on *precision*, *accuracy* or *correctness*, while measures of *efficiency* are relegated to secondary importance or ignored altogether. This can include the *speed* with which a task is performed (i.e. time efficiency), but also the reliance on other resources such as energy and knowledge (amount, diversity and content). The (independent) variable is often (training) time, if it is measured at all. These are important factors, and there are many others that can—and probably should—be considered as well.

We define *performance level* $\mathcal{P} \in \mathbb{N}_{\mathbb{A}} : \mathbb{A} \to [-1, 1]$ to be the main dependent variable for our evaluation purposes, where $\mathcal{P}$ can be some combination of accuracy/correctness $\mathcal{A}$, speed/time-efficiency $\mathcal{T}$, energy-efficiency $\mathcal{E}$, etc. The performance level of system A with knowledge K on task X can be written as $\mathcal{P}(X, A, K)$.[7] Efficiency/resource properties can also be defined with respect to a certain level of performance. For instance, $\mathcal{T}^{\mathcal{P}=0.9}$ could be the amount of training time required to reach a score of 0.9 on performance measure $\mathcal{P}$.

4.2 Adaptivity

To measure the adaptivity of a system, it is not only important to look at the rate at which a new task is learned, but also how much new knowledge is required.[8] The capacity for lifelong [26, 19] and transfer learning [22, 16, 11] depends not just on time, but on the content of old and new knowledge, as existing knowledge determines in part the speed of acquisition of new knowledge—both with respect to prerequisite knowledge already acquired (e.g. recognizing letters helps with learning to read) and to how related knowledge may apply to a new task, also called transfer learning (e.g. knowledge of driving one kind of motorized vehicle can help speed up learning how to drive others).

The most important measures of learning are probably the ones that relate the needed time, knowledge and other resources to a desired level of performance. Performance can for instance be measured as a function of training time, by varying t_n in $\mathcal{P}(X, A, \mathcal{K}(X, A, K_0, t_0 : t_n))$, which is the performance on task X after the system has trained on it between times t_0 and t_n ($\mathcal{K}$ is the knowledge system A with starting knowledge K_0 obtained in task-environment X between times t_0 and t_n). This can show how efficiently a certain task is learned. A more general measure of learning efficiency within a class of task-environments can be obtained by taking a weighted average of the performance on those other task-environments. However, this will only work if 1) the performance measures for various tasks are normalized (e.g. between -1 and 1), and 2) if corrections are made for the complexity and size of individual tasks. The goal here would be that if we encounter a new task with broadly the same properties as the measured class of task-environments, we can use the learner's general *learn rate* property to predict with some accuracy what would be needed to learn the new task (depending on known details such as its size and complexity).

[7] As before, we can think of K as the cognitive aspects of the actor A which can change over time, while A (also) contains constant aspects such as the system's identity.

[8] Knowledge acquisition takes time, so there is a correlation, but it is not perfect. Many algorithms spend much time processing the same data over and over, and the intelligence with which a space is explored can greatly influence how much new knowledge is gathered in a given time span (cf. active learning [18]).

Transfer learning ability $\mathcal{TL} : \mathbb{A} \times \mathbb{TE} \times \mathbb{TE} \to \mathbb{R}$ of a system from one task (collection) X to another Y can be defined in a number of complementary ways. For instance, we can look at how training for some time on X affects performance on Y in several ways. In each case we compare performance from training just on the target task(s) Y to performance of training on X first and then on Y.

- Raw performance transfer is the performance on Y after having trained on X for a given amount of time (dependent variable) or until a given level of performance: $\mathcal{P}(Y, A, \mathcal{K}(X, A, K_0, t_0 : t_n))$. This can also be considered as *generalization* to a different set of tasks. A special case would be if $Y \subset X$, in which case the test corresponds to a spot check of a larger amount of knowledge.
- The performance "loss" of training on the wrong task-environments can be calculated as the difference between $\mathcal{P}(X, A, \mathcal{K}(X, A, K_0, t_0 : t_n))$ and $\mathcal{P}(Y, A, \mathcal{K}(X, A, K_0, t_0 : t_n))$.
- A performance "gain" can be calculated by comparing how much "extra" training in another task-environment helps (or hinders): $\mathcal{P}(Y, A, \mathcal{K}(Y, A, \mathcal{K}(X, A, K_0, t_k : t_0), t_0 : t_n))$. Note that in this case more total time is used for training.
- Alternatively, we could look at whether it might help to spend part of a fixed time budget on task X before moving on to Y: $\mathcal{P}(Y, A, \mathcal{K}(Y, A, \mathcal{K}(X, A, K_0, t_0 : t_m), t_m : t_n))$.

Probably the most straightforward way to apply these measures is to take performance transfer as a function of training time t_n (and t_k/t_m). However, we can also take an extra step and analyze performance transfer as a function of attained performance level on X or the amount of knowledge that was acquired. Additionally, we can look at other things than performance, such as:

- Raw knowledge transfer is defined as the minimum amount of knowledge for reaching performance level y on Y that needs to be added to the system's knowledge after having trained on X: $|\mathcal{K}^{\mathcal{P}=y}(Y, A, \mathcal{K}(X, A, K_0, t_0 : t_n))|$.
- Training time transfer is defined as the difference between the amount of time to reach performance level y on Y from the current situation, and the amount of time needed to reach that level after having trained in X first: $\mathcal{T}^{\mathcal{P}=y}(Y, A, \mathcal{K}(X, A, K_0, t_0 : t_n))$.
- Composite time transfer is training time transfer plus the amount of time spent to train on X: $\mathcal{T}^{\mathcal{P}=y}(Y, A, \mathcal{K}(X, A, K_0, t_0 : t_n)) + t_n - t_0$.

Composite transfer measures can be used in teaching scenarios to judge whether it is worthwhile to decompose a task into component parts that are learned separately before a full task is presented [4].

In each case the transfer can be positive or negative. It is possible that knowledge is acquired in X that contradicts knowledge necessary to succeed in Y, possibly through no real fault of the system (e.g. it could have a bad teacher). Nevertheless, in many cases an intelligent adaptive system should be able to make use of its previously acquired knowledge when learning something new. It is therefore important that these systems retain some plasticity, even when they acquire more and more knowledge.

While it is most intuitive to consider transfer from previous tasks to newly learned ones, there can also be transfer (or interference) the other way around. Ideally, learning new tasks (e.g. a new language) should make one better at older tasks as well (e.g. other languages), but often the reverse is true. Catastrophic interference or forgetting plagues many machine learning systems: the ability to perform old

tasks is lost when new tasks are learned [9, 3]. Interference and forgetting can be measured in similar methods as above.

So far we have assumed that one task (collection) is learned after another, but often tasks are learned and performed in parallel (cf. multitask learning [8, 17]). Again, similar measures can be defined for this scenario. In this case we also delve into the realm of distractions and robustness.

4.3 Robustness

Robustness is another important aspect of AI systems. The two main things to consider are *when* (or *if*) the system "breaks down" and *how* it does so. Furthermore, even in adverse or novel conditions we would like the system to eventually adapt so that it can properly function even in the new situation. Ideally we want a system that never breaks down, but this is likely not a realistic goal if we can not anticipate all the situations the system will find itself in. A more realistic goal may be to require that the system degrades gracefully, notices when things go awry and takes appropriate action—such as asking for help, moving back to a safer situation or gathering more information to start the adaptation process.

A general AI system may encounter various kinds of (internal and external) noise, distractions from extraneous input/output (dimensions) or parallel tasks, situations that differ on various dimensions from what it is used to, strain on its subsystems, or outright breakage of components. It is important to know that as these factors move further away from the ideal situation, the system will continue to function appropriately, detect the problem and/or degrade gracefully. Robustness can be measured with performance as the dependent variable and one or more kinds of interference as the independent variables. However, it can also be combined with other dependent variables such as training time or the ability to transfer knowledge between tasks. The standard form of measuring robustness is similar to knowledge transfer: $\mathcal{P}(Y, A, \mathcal{K}(X, A, K_0, t_0 : t_n))$. However, in this case we are more interested in the relation between task-environments X and Y than the training time and efficiency, and the difference between training on X first vs. training on Y directly. A good task-theory can help tremendously in the measurement of robustness. Most notably, we would want a task-environment generator or modifier that can create variants of the training environments X that differ in various desired ways.

One of the simplest notions of robustness is sensitivity to noise. Even a relatively primitive task-theory should make it possible to add noise, distortions or latency to the system's sensors and/or actuators. We could then draw a graph of how performance deteriorates as noise increases, which provides a nice quantitative picture of robustness in the face of a particular kind of interference. Other relatively easy-to-generate variations are to add irrelevant distractions to the environment (e.g. extra sensors, actuators or objects) or parallel tasks, to create a scarcity of resources (e.g. time, energy, knowledge).

If we look at $\mathcal{P}(Y, A, \mathcal{K}(X, A, K_0, t_0 : t_n))$ as a function of the distance between Y and X (measured along desired dimensions through some yet-to-be-invented task theory), we get a measure of *generalization*. Generalization ability of an adaptive system is among its most important properties, but we can additionally use these methods to judge the representativeness of certain training environments (X), which could then be used to more efficiently teach the system to perform well in a wide range of situations.

Aside from external sources of interference, we can also look at internal sources: what happens if faults occur inside of the system itself? If a (physical or "cognitive") component breaks down, will the system "die" or go "crazy", or will it just deteriorate performance slightly and perhaps even prompt the system to adjust and fix the problem? Some types of adversity that need not be catastrophic include memory deterioration or corruption, system/CPU strain/overload, synchronization errors, dropped messages, latency, noise, and failure of individual components (in a modular or distributed system).

In addition to looking at quantitative measures of dips in performance, it is also important to consider qualitative factors: if performance suddenly drops to zero we must ask what it means. Did the system just go crazy, or did it sensibly decide that the situation has deteriorated to the point where shutting down, warning a human or pursuing more fruitful endeavors is more appropriate? Answering such questions may require analyzing the system's behavior, motivations and/or reasoning in more detail.

4.4 Understanding

We typically want to see a certain continuity in our systems' behavior, even as they encounter new situations. For learning systems however, we also want them to adapt so that they may improve their performance even in these unforeseen situations. There is a delicate balance between change to a system's parameters that is desirable, and change that isn't. Importantly, we want performance to improve (or not degrade) from *our* perspective. Subjective improvement from the *system*'s perspective might be achieved by changing the way success is measured internally, but this is typically not something that we want. Most contemporary AI systems lack this capability, but more powerful and general systems of the future may possess the ability to recursively self-improve. While there are reasons to believe that a sufficiently intelligent system would attempt to protect the integrity of its goals [15, 6], we still need to ensure that these attempts are indeed successfully made.

Predictability results not just from vigorous quantitative tests, but also from more qualitative tests of a system's *understanding*. Some recent examples show that high performance on a task does not guarantee that the system performing it understands that task [21]. Deep neural networks have been trained to recognize images rather adequately—in some cases rivaling human-level performance—but are easily fooled with complete nonsense images or some slight deviations from the training data. When the stakes are higher, it is important to know that such weaknesses don't exist when situations differ slightly from the training scenario. By testing a system's understanding and examining its argumentation (cf. argument-based ML [14]), we can assure ourselves of the kind of reasoning that will be used even when novel situations are encountered. Perhaps even more importantly a system's ability to *grow* its understanding should be assessed to strengthen the foundation on which a system's level of adaptivity and intelligence is estimated, and the level of trust that we place in it.

Assessing a system's understanding of one or more phenomena[9] seems critical for generalizing a system's performance with respect to unfamiliar and novel tasks and environments. In prior work we have proposed a definition of understanding, based on the idea of models M of phenomena [25]. The closer the models describe important aspects of a phenomenon's properties and relations to other

[9] We define a phenomenon Φ (a process, state of affairs, thing, or occurrence) as $\Phi \subset W$ where W is the world in which the phenomenon exists and Φ is made up of a set of elements (discernible "sub-parts" of Φ $\{\varphi_1 \ldots \varphi_n \in \Phi\}$) of various kinds including relations $\Re_\Phi$ (causal, mereological, etc.) that couple elements of Φ with each other, and with those of other phenomena. See Thórisson et al. [25] for further details.

things the more general their utility, and the more deeply the system can be said to *understand* the phenomenon. Among other things, good models allow for making good predictions. Importantly, the theory goes further than this, however, requiring in addition that for proper assessment of a system's understanding its ability to explain, achieve goals with respect to, and (re-)create[10] a phenomenon must also be assessed.

In short, the theory states that, given any phenomenon Φ, model M_Φ contains information structures that together can be used to *explain* Φ, *predict* Φ, produce effective plans for *achieving goals* with respect to Φ, and *(re)create* Φ. For any set of models M, the closer the information structures $m_i \in M$ represent elements (sub-parts) $\varphi \in \Phi$, at any level of detail, including their internal and external relations/couplings $\Re_\Phi$, the greater the *accuracy* of M with respect to Φ. An adaptive system A's *understanding* of phenomenon Φ depends on the quality, that is *accuracy* and *completeness* of A's *models* M of Φ, which enable prediction, action upon, explanation, and (re)creation of Φ. The better such models describe Φ, the better any of these will be. Understanding thus has a (multidimensional) *gradient* from low to high levels [25].

Prediction is one form of evidence for understanding. Some prediction can be done based on correlations, as prediction does not require representation of the direction of causation yet captures co-occurrence of events. Prediction of a particular turn of events requires (a) setting up initial variables correctly, and (b) simulating the implications of (computing deductions from) this initial setup.

A number of different questions can be asked regarding the prediction of a phenomenon, for instance:

- From a particular (partial) start state, what is the time (range) in which the phenomenon is expected to occur (if at all)?
- What will be the state of the phenomenon at a future time, given some starting conditions?
- For some phenomenon $\Phi = \{\phi_1 \dots \phi_n\}$, given values for some subset $\Psi \subset \Phi$, predict the values for the remaining $\phi_i \in \Phi$.
- Predict the state or occurrence of related phenomena $\Omega \subset \Re_\Phi$ given the state of Φ.

Picking an appropriate set of such questions is at the heart of properly evaluating a system's ability to predict a phenomenon.

Goal Achievement. Correlation is not sufficient, however, to inform how one achieves goals with respect to some phenomenon Φ. For this one needs causal relations. Achieving goals means that some variables in Φ can be manipulated directly or indirectly (via intermediate variables). Achieving goals with respect to a phenomenon Φ does not just require understanding the individual components of Φ itself, but also how these relate to variables that are *under the system's control*. In short: the system needs models for interaction with the environment as well as the phenomenon. For a robotic agent driving a regular automobile, to take one example, the system must possess models of its own sensors and manipulators and how these relate to the automobile's controls (steering wheel, brakes, accelerator, etc.). Such interfaces tend to be rather task-specific, however, and are thus undesirable as a required part of an evaluation scheme for understanding. Instead, we call for an ability to *produce effective plans* for achieving goals with respect to Φ. An effective plan is one that can be proven useful, efficient, effective, and correct, through implementation.[11]

Goal achievement with respect to some phenomenon Φ can be defined by looking at the system's performance (cf. section 4.1) in task-environments and/or interactions that feature Φ. The phenomenon can play a number of different roles, depending on its type (e.g. event, process, tool, obstacle, etc.). Events can be caused, prevented or changed (usually within a certain time range). Objects can have their state configured in a desired manner. When an object is a tool or obstacle, we can compare the performance in environments with and without Φ. Processes can have several effects on the environment (possibly depending on the manner of their execution), and we can set a task-environment's goal to be accomplished by some of these effects and negated by others to see if the system can flexibly execute the process. If the system's performance in task-environments and/or interactions that include Φ are consistently better than when Φ is absent, this can indicate a higher level of understanding of Φ.

Explanation is an even stronger requirement for demonstrating understanding, testing a system's ability to use its models for abductive reasoning. Correlation does not suffice for producing a (true) explanation for an event or a phenomenon's behavior, as correlation does not imply causation. One may even have a predictive model of a phenomenon that nevertheless represent incorrectly its parts and their relations (to each other and parts of other phenomena). This is why scientific models and theories must be both predictive *and* explanatory—together constituting a litmus test for complete and accurate capturing of causal relations.

(Re)creating a phenomenon is perhaps the strongest kind of evidence for understanding. It is also a prerequisite for correctly building new knowledge that relies on it, which in turn is the key to growing one's understanding of the world. By "creating" we mean, as in the case of noted physicist Richard Feynman, the ability to produce a model of the phenomenon in sufficient detail to replicate its necessary and sufficient features. Note that this is not limited to (re)creation *by the system* using its own I/O capabilities, but involves an understanding of how the phenomenon can be created *in general* by the system, by others, by the environment itself, or even by some hypothetical entity with (partially) imagined capabilities. Requiring understanders to produce such models exposes the completeness of their understanding.

It is important to emphasize here that understanding, in this formulation, is not reductionist: Neither does it equate the ability to understand with the ability to behave in certain ways toward a phenomenon (e.g. achieve goals), nor the ability to predict it, nor the ability to explain it, nor the ability to (re)create it. While any of these may provides hints of a system's understanding of a phenomenon, it cannot guarantee it. In our theory *all are really required* (to some minimum extent) to (properly) assess a system's understanding; any assessment method that does not include these four in some form runs a significantly higher risk of failure.

5 THE *HOW:* CONDUCTING TESTS AND ANALYZING RESPONSES

All evaluations are contextual: i.e. they are done with respect to a task-environment, or collection of task-environments. We should examine how the measurements depend on the chosen collection of task-environments, and strive towards using as large a range as possible. We will need to say something about the range of task-environments that we think our results generalize to as well. Se-

[10] We mean this in the same sense as when we say that a chef's recipe demonstrates her understanding of baking, or a physicists' simulation of the universe demonstrates their understanding of how the universe works.

[11] Producing plans, while not being as specific as requiring intimate familiarity with some I/O devices to every Φ, requires nevertheless knowledge of some language for producing said plans, but it is somewhat more general and thus probably a better choice.

lection and/or creation of task-environments for the optimal measurement of desired system properties that generalize to other task-environments requires a *task theory* [24].

5.1 Construction and Selection of Task-Environments

We need a way to relate the things we want to test for to the information that can be obtained from (aspects of) task-environments. Due to the differences in AI systems, the purposes for which they are built and the properties we want to test for, it is impossible to construct a single test, or test battery, that measures everything for all systems. Rather, we need a task theory that allows us to analyze and construct tasks to specification, in combination with knowledge about the properties and behavior of intelligent, adaptive systems.

Given an intelligent system and a question about one or more of its properties, we should be able to a) construct a task-environment, b) adapt a given task-environment, or c) select a task-environment from some choices, in order to optimally answer that question about the system. Given a task-environment we would like to be able to predict reasonable behaviors for a certain system or class of systems with certain properties. An informative task-environment would afford multiple behaviors that are distinctive with respect to the property that we want to measure. It is likely most informative to test around the edges/limits of the system's capabilities. A task theory that allows for scaling the scope or difficulty of environments would therefore be tremendously useful [23].

In some cases it may be possible to construct batteries of tasks for answering a certain question about a set of systems—e.g. a standardized exam. In other cases the evaluation may be more interactive and explorative. Another important consideration is how much control we have over the system (e.g. can we look at its source code or memory?) and its task-environment (e.g. is it virtual and owned by us?).

Motivation and Incentive Somehow we need to get the system to actually perform the envisioned task, which may be difficult without full control. Simply placing a system in a task-environment doesn't guarantee that it will perform (or even understand) the task that we want. If you place a child in a room with a multiple-choice IQ test, will it fill it out as you want? Or will it check the boxes in an aesthetically pleasing manner? Or just ignore the test? In general we can never be sure, but we can try to incentivize good performance on the test. Alternatively, we can look at the behavior and try to derive the task the system was trying to perform (cf. inverse reinforcement learning [1]; although this tends to assume a certain level of competency on the part of the system).

5.2 Judging Behavior

Rather than just looking at end results (e.g. score on an exam or tennis match), we can also look at performance/behavior during the test (i.e. the sequence of actions in response to stimuli). This should hopefully shed some light on inner workings and allow us to construct a model that is predictive in more situations.

In situations where we know a good solution to a task, we can compare that solution (or those solutions) to the observed behavior of the system. Assuming the system has the appropriate goals, we can then see where it deviates and consider what gap in knowledge or leap in reasoning led it to do so. Alternatively, under the assumption that the system is reasonably competent, we can try to find its motivations and goals through inverse reinforcement learning [1].

Deconstruction/decomposition of tasks into multiple smaller parts can be extremely useful for this purpose. In that case, we can use easier-to-perform performance evaluations on a much more granular scale.

5.3 Evaluating Understanding

To test for evidence of understanding a phenomenon Φ (a process, state of affairs, thing, or occurrence) in a particular task environment, we may probe (at least) four capabilities of the system (a) to *predict* Φ, (b) to *achieve goals* with respect to Φ, (c) to *explain* Φ, and (d) to *(re)create* Φ [25]. All can have a value in $[0, 1]$ where 0 is no understanding and 1 is perfect understanding.[12] For a thorough evaluation of understanding all four should be applied.

The major challenge that remains is how to perform this assessment. Goal achievement can be measured in a reasonably straightforward fashion, although we do require a way to construct goals and tasks that incorporate the phenomenon for which understanding is to be tested. Similarly, it should be possible to define a task that involves the desired phenomena's recreation. Testing for high-level predictions seems more challenging if the system doesn't automatically communicate the predictions that it makes. Somewhat imperfect tests for predictions can be constructed by presenting the system with situations where correct predictions would likely prompt it to show different behavior than incorrect predictions. Alternatively, it may be possible to access the system's internals, in which case a trace of its operation may show which events and observations were expected.

Measuring explanations may be the most important and difficult challenge in AI evaluation though. Most systems are not explicitly built to provide human-understandable explanations for their actions, but from this we cannot conclude that they are not adequately modeling the causal chains of the environment and justifying their behavior to themselves in some way. If a system doesn't explicitly try to explain itself, then it seems that we can only access explanations by inspecting the system's inner workings. Subsymbolic systems are notoriously difficult to understand for humans, but even symbolic systems could present difficulties; either because their symbols are unlabeled and grounded in a different ways than ours, or because the amount of involved models and considerations in each decision are overwhelming. Overcoming these issues is an open problem, but given the importance of modeling the world's causal chains and making justifiable decisions, we suggest that AGI systems ought to be built with a faculty for explanation and summarization in mind, which should help us evaluate their understanding.

6 CONCLUSION & FUTURE WORK

Evaluation of intelligent adaptive systems is important for a wide variety of reasons. Progress in AI depends on our ability to evaluate it: to find the strengths and weaknesses of our programs and improve them where necessary. Looking at performance alone is not enough, since we need our more general systems to operate beneficially even

[12] More complex measurements could of course be used for a more thorough or faithful representation of understanding; projecting it down to a single dimension may lose some (important) information. This simplification is however immaterial to the present purposes.

in situations that we did not fully foresee. We must therefore consider these systems' robustness to changing and possibly deteriorating conditions and acquire confidence that they will adapt in ways that allow them to continue to be beneficial to their human owners.

Focus in AI evaluation has been mostly on testing for performance—often in specialized (and limited) domains—by measuring some final result that was attained on a task at a single point in time. Not only do we need to consider other factors like adaptivity and robustness: we must also look beyond the final impact that is made on the system's environment. Moment-to-moment behavior can be a rich source of information that sheds much light on how or why a certain level of performance was attained. Even more importantly, we must attempt to measure levels of understanding. An explanation is more than a single data point: it is a model that can be applied in many situations. If we know that a system understands certain concepts—most notably our values—we can be relatively confident that it will make the right considerations, even in unforeseen situations.

Measuring system properties beyond performance as well as the analysis of behavior and understanding are very challenging, and it is not obvious how to do it. It is however clear that better theories for testing, understanding and task-environments are a part of the solution. Future work must investigate these avenues of research that are necessary if we are to move forward in our quest for general-purpose adaptive AI.

ACKNOWLEDGEMENTS

The authors would like to thank Magnús Pálsson at IIIM for comments on an earlier version of this paper. This work was sponsored by the School of Computer Science at Reykjavik University, and a Centers of Excellence Grant (IIIM) from the Science & Technology Policy Council of Iceland.

REFERENCES

[1] Pieter Abbeel and Andrew Y. Ng, 'Apprenticeship learning via inverse reinforcement learning', in *Proceedings of the twenty-first international conference on Machine learning*, p. 1. ACM, (2004).

[2] Tarek Besold, José Hernández-Orallo, and Ute Schmid, 'Can Machine Intelligence be Measured in the Same Way as Human intelligence?', *KI - Künstliche Intelligenz*, 1–7, (April 2015).

[3] J. Bieger, I. G. Sprinkhuizen-Kuyper, and I. J. E. I. van Rooij, 'Meaningful Representations Prevent Catastrophic Interference', in *Proceedings of the 21st Benelux Conference on Artificial Intelligence*, pp. 19–26, Eindhoven, The Netherlands, (2009).

[4] Jordi Bieger, Kristinn R. Thórisson, and Deon Garrett, 'Raising AI: Tutoring Matters', in *Proceedings of AGI-14*, pp. 1–10, Quebec City, Canada, (2014). Springer.

[5] Jordi Bieger, Kristinn R. Thórisson, and Pei Wang, 'Safe Baby AGI', in *Proceedings of AGI-15*, pp. 46–49, Berlin, (2015). Springer-Verlag.

[6] Nick Bostrom, 'The superintelligent will: Motivation and instrumental rationality in advanced artificial agents', *Minds and Machines*, **22**(2), 71–85, (2012).

[7] Selmer Bringsjord and Bettina Schimanski, 'What is artificial intelligence? Psychometric AI as an answer', in *IJCAI*, pp. 887–893. Citeseer, (2003).

[8] Richard A. Caruana, *Multitask Learning Thesis*, PhD, Carnegie Mellon University, Pittsburgh, PA, September 1997.

[9] Robert M. French, 'Catastrophic interference in connectionist networks', in *Encyclopedia of Cognitive Science*, ed., Lynn Nadel, volume 1, 431–435, Nature Publishing Group, London, (2003).

[10] José Hernández-Orallo, 'AI Evaluation: past, present and future', *CoRR*, **abs/1408.6908**, (2014).

[11] Alessandro Lazaric, 'Transfer in Reinforcement Learning: A Framework and a Survey', in *Reinforcement Learning*, 143–173, Springer, (2012).

[12] Shane Legg and Marcus Hutter, 'Tests of Machine Intelligence', *CoRR*, **abs/0712.3825**, (2007). arXiv: 0712.3825.

[13] *Beyond the Turing Test*, eds., Gary Marcus, Francesca Rossi, and Manuela Veloso, volume 37 of *AI Magazine*, AAAI, 1 edn., 2016.

[14] Martin Možina, Jure Žabkar, and Ivan Bratko, 'Argument based machine learning', *Artificial Intelligence*, **171**, 922–937, (2007).

[15] Stephen M. Omohundro, 'The basic AI drives', *Frontiers in Artificial Intelligence and applications*, **171**, 483, (2008).

[16] Sinno Jialin Pan and Qiang Yang, 'A Survey on Transfer Learning', *IEEE Transactions on Knowledge and Data Engineering*, **22**(10), 1345–1359, (October 2010).

[17] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov, 'Actor-Mimic: Deep Multitask and Transfer Reinforcement Learning', *arXiv:1511.06342 [cs]*, (November 2015). arXiv: 1511.06342.

[18] Burr Settles, 'Active learning', *Synthesis Lectures on Artificial Intelligence and Machine Learning*, **6**(1), 1–114, (2012).

[19] Daniel L. Silver, Qiang Yang, and Lianghao Li, 'Lifelong Machine Learning Systems: Beyond Learning Algorithms.', in *AAAI Spring Symposium: Lifelong Machine Learning*, (2013).

[20] Bas R. Steunebrink, Kristinn R. Thórisson, and Jürgen Schmidhuber, 'Growing recursive self-improvers', *To be published in B. R. Steunebrink et al. (eds.), Proceedings of Artificial General Intelligence 2016*, (2016).

[21] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, 'Intriguing properties of neural networks', *arXiv:1312.6199 [cs]*, (December 2013). arXiv: 1312.6199.

[22] Matthew E. Taylor and Peter Stone, 'Transfer learning for reinforcement learning domains: A survey', *The Journal of Machine Learning Research*, **10**, 1633–1685, (2009).

[23] Kristinn R. Thórisson, Jordi Bieger, Stephan Schiffel, and Deon Garrett, 'Towards Flexible Task Environments for Comprehensive Evaluation of Artificial Intelligent Systems & Automatic Learners', in *Proceedings of AGI-15*, pp. 187–196, Berlin, (2015). Springer-Verlag.

[24] Kristinn R. Thórisson, Jordi Bieger, Thröstur Thorarensen, Jóna S. Sigurðardóttir, and Bas R. Steunebrink, 'Why Artificial Intelligence Needs a Task Theory — And What it Might Look Like', *To be published in B. R. Steunebrink et al. (eds.), Proceedings of Artificial General Intelligence 2016*, (2016). arXiv: 1604.04660.

[25] Kristinn R. Thórisson, David Kremelberg, and Bas R. Steunebrink, 'About understanding', *To be published in B. R. Steunebrink et al. (eds.), Proceedings of Artificial General Intelligence 2016*, **6**, (2016).

[26] Sebastian Thrun, 'Lifelong Learning: A Case Study', Technical CMU-CS-95-208, Carnegie Mellon University, Pittsburgh, PA, (November 1995).

[27] Alan M. Turing, 'Computing machinery and intelligence', *Mind*, **59**(236), 433–460, (1950).