Dual Indicators to Analyse AI Benchmarks: Difficulty, Discrimination, Ability and Generality

Fernando Martínez-Plumed and José Hernández-Orallo

***Abstract*—With the purpose of better analysing the result of AI benchmarks, we present two indicators on the side of the AI problems, *difficulty* and *discrimination*, and two indicators on the side of the AI systems, *ability* and *generality*. The first three are adapted from psychometric models in item response theory (IRT), whereas generality is defined as a new metric that evaluates whether an agent is consistently good at easy problems and bad at difficult ones. We illustrate how these key indicators give us more insight on the results of two popular benchmarks in AI, the Arcade Learning Environment (Atari 2600 games) and the General Video Game AI competition, and we include some guidelines to estimate and interpret these indicators for other AI benchmarks and competitions.**

***Index Terms*—AI evaluation, AI benchmarks, item response theory, generality.**

I. Introduction

Systems in all areas in AI, however well designed or implemented, are ultimately evaluated on a set of problems or applications. In order to prevent systems from specialising to these benchmarks, these try to include as many problems and as diverse as possible. Also, because of the maturity of some subfields in AI, many different techniques are available. This leads to a situation where many techniques are usually confronted with many problems. This is what we usually find in the experimental section of many technical papers and, especially, on running competitions.

We see this change from the breakthroughs of one system on one single problem, to the evaluation of many systems on many problems. Examples of the former are draughts (checkers) in the 1950s [1], chess in the 2000s with *Deep Blue* against the human chess champion Garry Kasparov [2], the 2010s IBM's program Watson winning the *Jeopardy!* TV quiz [3], or, recently, the game of Go [4]. Examples of the latter are the UCI machine learning repository [5], the ICAPS planning and scheduling competitions [6] or the ImageNet challenges [7] (see [8] for a more complete list of benchmarks and competitions).

Furthermore, AI is nowadays used to generate much more general-purpose, adaptive and successful behaviours primarily in (video) games. For instance, deep reinforcement learning and other approaches are now able to perform extremely well in board games (e.g., [4]) and relatively well in many arcade games (e.g., [9]). This brings the potential to use them as *non-player characters* (NPC) or non-human opponents for more complex games in the future [10], where the same architecture can be retrained for different games, without the effort of designing specific NPCs for each game. However, in these more generic scenarios, we do not know how to analyse their behaviour beyond specific performance, especially when we want to compare different approaches for a range of games. In particular, apart from specialised agents, it would be useful to have generic algorithms that can produce relatively good NPCs for games that look of easy or medium difficulty, instead of those that are very good, or specialised, at some hard games but very poor at many easy others.

AI is also now paying attention to systems that solve several tasks at a time [8], [11]. Indeed, a popular setting for general-purpose evaluation today is a collection of games under an interactive scenario, where agents can perceive and act, and are rewarded when they make good choices. Many different platforms have recently appeared in this regard [12], laying special emphasis on the use of 2D/3D videogames for AI evaluation [13] and attracting mainstream attention [14].

Two representative examples are the *Arcade Learning Environment* (ALE) [15], a collection of Atari 2600 games; and the *General Video Game AI* (GVGAI) competition [16], a benchmark that comprises a large number of real-time 2D grid games. Both ALE and GVGAI are remarkable benchmarks that allow us to observe the performance of AI agents on a multiplicity of problems. They have both received plenty of interest and have become a reference for AI experimentation and evaluation in the past few years. The popularity of these AI benchmarks have produced a good number of results that can now be analysed in hindsight and used to better understand not only these benchmarks, but also general-purpose AI overall. This analysis can be understood from the viewpoint of the AI systems (and how to improve AI techniques) but also from the viewpoint of the problems (and how to improve the benchmarks). In this paper we argue that using performance is insufficient to get a proper insight of what the systems are able to do (and how they achieve it) and what the problems in the benchmark are evaluating.

In this paper we present two pairs of key indicators that can help us understand the results in AI benchmarks in a more informative way. On the one hand, we claim difficulty and discrimination as key indicators for AI problems. On the other hand, we postulate ability and generality as key indicators for AI systems. Namely,

- Looking at the problems, inferring a difficulty indicator helps us control whether we are evaluating a proper range of difficulties, and clarifies that we expect a general systems to perform well for almost all easy problems before we can direct our progress towards areas of higher

F. Martínez-Plumed is with Universitat Politècnica de València, Spain e-mail: fmartinez@dsic.upv.es.

J. Hernández-Orallo is with Universitat Politècnica de València, Spain e-mail: jorallo@dsic.upv.es.

Manuscript received X, 20XX; revised X, 20XX.

difficulty. Relatedly, it is also important to detect whether difficult problems are only solved by able systems, and not by chance or specialisation by very poor systems. This is the notion of discrimination, which will spot that some problems may be useless, or even detrimental, for an efficient and robust evaluation. Considering the increasing computing demands of training and evaluation for recent algorithms, any understanding of what the key tasks are can imply an important contribution for AI researchers.

- Looking at the systems, ability gives us a different perspective from performance, as it considers the difficulty of the problems, instead of a simple average. The most novel insight comes from seeing whether a system –motivated by increasing performance– focuses on a big pocket of problems while neglecting some other smaller pockets. It is of key importance, however, and widely overlooked in AI, that *we must understand generality in the context of difficulty* [11]. In other words, if a system covers some of the low-hanging fruits but excludes others, we may suspect there is some specialisation. On the contrary, if a system covers all low-hanging fruits and almost none of the hard problems, we can usually infer some kind of systematic generality in the behaviour of the system. Ultimately, it is crucial for AI researchers to know whether they are progressing through generality or through the exploitation of subfamilies of problems.

The analysis under these key indicators represents a novel way of understanding not only benchmark results in AI [12], but also video game competitions (e.g., *Super Mario Bros* [17] or *Angry Birds* [18] or StarCraft [19] AI competitions) as well as the existing architectures for multi-purpose game agents and bots addressing them [20]. This kind of assessment may have a huge impact on how players and competitions are designed and how the results of the AI systems (and humans) are interpreted. We are not claiming that these two pairs of indicators are necessarily giving us the most complete information (this is ultimately given by the whole data), but they are a good trade-off between monolithic indicators (and limited insight) and too many indicators (and strong overlap).

In this paper, we obtain these indicators in different ways. Some of them (difficulty, discrimination and ability) are estimated through simple models inherited from Item Response Theory (IRT), a powerful technique from psychometrics [21]. Generality, as newly defined in this paper, is derived from the dispersion statistics, but taking difficulty into account. In all cases –and this is important to note–, the indicators are populational, i.e., they depend on a set of AI systems and a set of AI problems. Consequently, when we change the population, the obtained indicators may change as well.

The rest of the paper is organised as follows. Section II presents the indicators we propose for analysing the results of AI benchmarks. Section III describes the result data we will explore: the ALE and GVGAI problems and the AI techniques used to solve them. The estimation of the indicators for these benchmarks and how they can be used to understand the behaviour of problems and systems, is seen in sections IV and V respectively. Section VI summarises the main findings and contributions, describes some limitations and guidelines for evaluation, and discusses the future work.

II. Key indicators for AI benchmark results

Whenever a benchmark is built, its creators usually consider a set of problems or tasks that are representative of the kinds of applications we want to progress on. One way of looking at this progress is in terms of overall performance. If the set of systems or agents j is Π, with $|\Pi| = m$, and the set of problems, tasks or items i is M, with $|M| = n$, we can have a measure of the result (the *response*) of each system j for each problem i, as $R_{j,i}$, making up an $m \times n$ result matrix[1]. Then, for a single system j, we can calculate its (weighted) average performance $\sum_j w_i R_{j,i}$, where w_i are weights given to problems. The first thing this assumes is that performances are commensurate, with the weights giving more or less relevance to some problems depending on their importance. In this case, we would say that a system A is better than a system B if the (weighted) average performance is higher. In some other cases, if the performances are not commensurate, one can be satisfied with a binary comparison by how many wins/ties/losses there are in their performances. If we want to compare more than two systems at the same time, we would use rankings instead, which can be produced from the aggregated performance, from a count of pairwise comparisons or in other ways. There are many other variations, especially when we want to apply some statistical tests on the results, but all of them are based on some notion of aggregated or comparative performance.

However, there is another way of looking at this. One can consider that not all problems have the same difficulty. That does not mean that difficult problems should count more than easy problems, but that difficulty should be taken into account in any notion of quality of a system, and most importantly, in any measure of progress. Indeed, a system behaving well on difficult items but poorly on easy items would certainly be a strange specimen. In a way, we not only expect a positive monotonicity between the quality of a system and the probability of a correct response, but also some kind of positive monotonicity between the quality of a system and the difficulty of the problems it can likely solve. This observation suggests a completely different way of analysing results, which led to IRT, as we introduce next.

A. IRT in AI: ability, difficulty and discrimination

Item response theory (IRT) [21] has mainly been used in educational testing and psychometric evaluation in which examinees' ability is measured using a test with several questions (i.e., items). In essence, IRT is a set of mathematical models that describe the relationship between a latent trait of interest and the respondents' answers to individual items, where the probability of a response for an item is a function of the examinee's ability and some item's parameters. There are models developed in IRT for different kinds of response, but we will focus on the dichotomous models where responses can be either correct or incorrect. Multiple choice items (more

[1] We follow the usual convention in IRT with items being columns referred to by index i.

than two options) can also be considered dichotomous since they can still be scored as correct/incorrect.

In this context, let $R_{j,i}$ be the binary result of a respondent j to item i, with $R_{j,i} = 1$ for a correct response and $R_{j,i} = 0$ otherwise. Let θ_j be the ability or proficiency of j, and let us imagine for a moment that this value is known. Now, assuming that the result only depends on the ability of the respondent, and we assume a particular value for respondent j, we can express the result as a function of i alone, i.e. R_i. For the basic 3-parameter (3PL) IRT model, the probability of a correct response on an item given the examinee's ability is modelled as a logistic function:

$$P(R_i = 1|\theta_j) = c_i + \frac{1 - c_i}{1 + e^{-a_i(\theta_j - b_i)}} \tag{1}$$

The above model provides an *Item Characteristic Curve (ICC)* (see Fig. 1 (Left)) with three parameters:

- *Difficulty* (b_i): it is the location parameter of the logistic function and can be seen as a measure of item difficulty. When $c_i = 0$, then $P(R_i = 1|b_i) = 0.5$.
- *Discrimination* (a_i): it indicates the steepness of the function at the location point. Alternatively we can use the slope at location point, computed as $a_i(1 - c_i)/4$ to measure the discrimination value of the instance.
- *Guessing* (c_i): it represents the probability of a correct response by a respondent with very low ability ($P(R_i = 1| -\infty) = c_i$). This is usually associated to a result given by chance.

The basic IRT models can be simplified to two parameters (e.g., assuming that $c_i = 0$), or just one parameter (assuming $c_i = 0$ and a fixed value of a_i, e.g., $a_i = 1$).

In our adaptation of IRT, an item in IRT can be identified with a problem or task in AI (e.g., an ALE or GVGAI game), and an individual, subject or respondent can be identified with an AI method, technique or system [22]. While a guessing parameter might be meaningful in some AI problems (e.g., classification [23], [24]), it is not appropriate when a random agent is expected to score poorly (e.g., in a videogame). On the contrary, the discrimination parameter is very informative about whether a particular instance is aligned with ability (i.e., to detect a negative monotonicity between the quality of a system and the probability of a correct response).

For IRT, the difficulty and discrimination of the items are considered latent traits that may be unknown, and they have to be estimated from the result matrix. Similarly, the ability of an individual is considered a latent trait that can also be estimated based on her responses to discriminating items with different levels of difficulty. Respondents who tend to correctly answer the most difficult items will be assigned to high values of ability. Difficult items in turn are those correctly answered only by the most proficient respondents. Notice that ability and difficulty appear subtracted in the exponent of the logistic model in Eq. 1, so they are on the same scale, which gives these two parameters a dual character (e.g., an agent of ability 4 has 0.5 probability of being correct for an item of difficulty 4). Also many models assume that both parameters follow a normal distribution.

Straightforward methods based on maximum-likelihood estimation (MLE) can be used to estimate either the item parameters (when respondent abilities are known) or the abilities (when item parameters are known). A more difficult, but common, situation is the estimation when both the item parameters and respondent abilities are unknown. In this situation, an iterative two-step procedure, the Birnbaum's method [25], can be adopted for dichotomous items:

- (1) Start with initial values for abilities θ_j (e.g., random values or the number of correct responses).
- (2) Estimate the model parameters, assuming the abilities of the previous step.
- (3) Estimate the abilities θ_j, assuming the model parameters in the previous step.
- (4) Until stop condition, go to (2).

Some implementations iterate the above process a fixed number of times (1,000 in `ltm` R package[2]) and/or stop when the model's goodness of fit reaches a given threshold. In Birnbaum's method, the fit of the model is based on approximate marginal maximum likelihood, using the Gauss-Hermite quadrature rule for the approximation of the required integrals. Item parameters and respondent abilities are alternatively estimated in this iterative process, and overall they are derived only based on a set of observed responses to items, with no previous knowledge about the true ability of the respondents.

The key assumption, in any case, apart from the particular model family and the parameter scales and distributions, is monotonicity. Fig. 1 (Left) shows an item characteristic curve, where the probability of correct response grows monotonically as a function of the ability of a classifier.

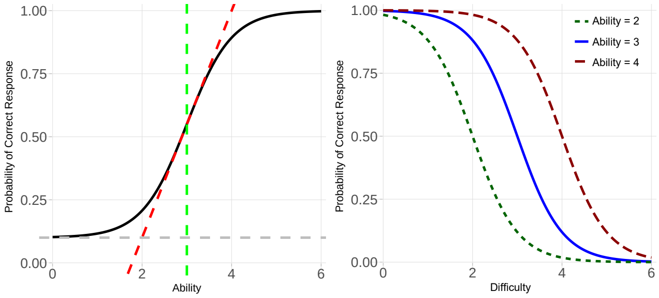

Fig. 1: Left: Example of a 3PL IRT ICC curve , with slope $a = 2$, location parameter $b = 3$ and guessing parameter $c = 0.1$. Right: Example of PCC curves with different abilities.

A positive (i.e., increasing) monotonicity is captured by a positive discrimination parameter. We actually expect more able systems to perform better than less able systems for that item. If this is not the case, when discrimination is negative, we have an unusual problem (and abstruse item, in the IRT terminology). If the problems are well selected (or well filtered) we should not have negative discriminations. When comparing positive discriminations, higher positive values (steeper curves) indicate that the item is very informative around its level of difficulty (very discriminative in the region where a high slope takes probability from low to high values).

[2] https://cran.r-project.org/web/packages/ltm/

On the contrary, low positive discrimination (flatter curves) means that the item is informative in a wider range of abilities, but less crisp near the difficulty parameter.

In a dual way, for a given ability we can plot the probability of correct response against difficulty. Fig. 1 (Right) shows three person characteristic curves (PCC) for three agents with different abilities. For AI we will call them *agent characteristic curves*. We expect these curves to be decreasingly monotonic, with very able systems being good at easy problems and decaying later than less able systems.

B. Precluding specialisation in AI: generality

The dichotomy between narrow and general AI has caused a long and controversial debate we will not reproduce here. However, even when we restrict to particular domains, there is usually the very good argument that we cannot expect a general system to be good at all possible problems. For some classes of problems, it is possible to build a system that is valid for all the problems in the class, but the situation becomes trickier for some other classes. This is especially the case for very open classes of problems such as video games. The idea of thinking of a *general* system that excels at all of them seems infeasible for many areas and benchmarks. This means that systems will fail at some tasks.

The subset of problems a system fails at can be completely random or can follow some pattern. If this pattern represents some particular characteristics these problems do (not) have, we can infer that the system has somewhat specialised in favour of (or against) that pattern. But if the pattern is related to the resources or the difficulty of the problem, we tend to consider this as a general adjustment between possibilities and resources. For instance, a calculator that could only multiply even numbers is not considered very general, whereas an ordinary pocket calculator is still considered general for multiplication, even if it fails for all the large numbers that go beyond its digit capacity. This suggests that the crux of the question about generality is capacity, or *difficulty*. One *can* actually be good at all (or almost all) problems up to certain difficulty, even for very broad problem classes.

This is related to some fundamental questions such as whether it is possible to build a system that excels at all possible problems. The no-free-lunch theorems [26], [27] argue that this is not possible, if one assumes block-uniformity for all possible problems. But this assumption will not hold if we order problems by some metric of difficulty (making very difficult problems less likely) and hence we look for agents that are good up to a certain level of difficulty. Actually, it is not a surprise that one finds free lunches, with systems that work generally well, when problems are built in such a way that resources (and difficulty) is not a completely unbounded and random variable, as happens with benchmarks in AI (e.g., [28]), or other games that were originally conceived for humans, which are resource-bounded systems. As we will see, only when the notion of difficulty is introduced (as we have done with IRT in the previous subsection), the analysis of generality becomes really meaningful.

To make the point even clearer, let us start with a notion that ignores difficulty, and we will introduce a version that does consider difficulty afterwards. Ignoring difficulty or any other parameter of the problems, one can simply introduce a measure of dispersion. Let us denote by σ_j^2 the populational variance of results for system j:

$$\sigma_j^2 = \frac{\sum_i (R_{j,i} - \bar{R}_j)^2}{n} \tag{2}$$

where $\bar{R}_j$ is the average result for system j. Considering this variance as an extra, informative, parameter, we could simply define a measure of *regularity* as the inverse of the variance. In an AI benchmark, such as ALE or GVGAI, one system would be regular if their results have low variance. If a system has very good results on some problems but very bad results on others, even if the overall quality is good, the regularity would be low.

For binary responses, we have a Bernouilli distribution, which means that the variance is reduced to just $\bar{R}_j \cdot (1 - \bar{R}_j)$. Consequently, we would not need an extra parameter for the dispersion of results for an agent, as variance, and hence regularity, would just be a function of the average performance $\bar{R}_j$ of the agent. This is one reason why an extra fourth parameter is not usually considered in the binary models in IRT. However, if the models are not binary, things are different (further information can be found in [29]).

In an AI benchmark, such as ALE or GVGA, one system would be regular if their results have low variance. If a system has very good results on some problems but very bad results on others, even if the overall quality is good, the regularity would be low. But is this variance caused by failing at more difficult problems (as expected) or failing at some subfamilies (pockets) of problems? This is when difficulty comes in: any generality metric (and associated plot) has to be analysed in terms of difficulty. We now introduce a new metric of generality that does take difficulty into account:

$$\gamma_j = \frac{1}{\sum_{h=1}^{h_{max}} (\sigma_j^{[h]})^2} \tag{3}$$

where $\sigma_j^{[h]}$ is the standard deviation of agent j on all problems of difficulty h. Continuous difficulties could be handled by using an integral instead of a sum. However, as we will estimate generality from a sample, we assume difficulties are discrete (or discretised by bins). Note that for small samples the number of bins is important. For instance, if there is only one single bin for all items then generality becomes equal to regularity. The higher the number of bins the better, approaching a continuous notion of difficulty. However, in estimation, for a finite number of instances, binning has to be done with at least a minimum number of examples per bin, to avoid undefined or very unstable variances.

The estimation can be done up to a maximum difficulty h_{max}. Nevertheless, if we assume that systems have zero performance once a certain difficulty is reached, then there is no need to set a limit of difficulties on the sum.

Given Equation 3, how can we get maximum generality? This is actually achieved when the slope of the agent characteristic curve is $-\infty$, i.e., the agent is perfect up to a given difficulty and hopeless from that moment on. In this case all variances are 0 and generality is infinity.

This relation of generality to the slope of the agent characteristic curve finally completes the circle and the duality between agents and problems, since generality can be seen as dual to discrimination (the slope of the item characteristic curve). For binary answers, there is no extra degree of freedom, and this extra parameter is confounded by all the others. But for continuous values (scores or probabilities), the slope has this extra degree of freedom.

In the same way that two agents with the same average performance (or ability parameter) cannot be distinguished as more or less general, we can have two systems with the same generality value with very different behaviour. For instance, we can have an agent A that is perfect up to difficulty $h = 5$ and an agent B that is perfect up to a difficulty $h = 2$. Looking at generality, both would have $\gamma = \infty$ and would then be indistinguishable with generality. But, clearly, agent A is more capable than agent B. Actually, in this case, we have a very interesting way of looking at their relation. If two agents are perfectly general and one is more capable than the other, we have a dominance relation. In our example, A dominates B. We can assert that whatever is solved by B is solved by A. This is closely related to the intuition of the transitivity of performance, already explored in previous papers of the GVGAI competition, which has raised doubts about the generality of the participants [30], [31].

In the following sections, we will better analyse the behaviour and interpretability of this new notion of generality. But let us first introduce the two benchmarks we will work with.

III. Benchmarks: ALE and GVGAI

In this section we will describe the benchmarks (ALE and GVGAI) we will use for the experiments in the next section[3].

A. The Arcade Learning Environment

The Arcade Learning Environment (ALE) was introduced by [15], after compiling a good number of games for Atari 2600, a popular console of the late 1970s and most of the 1980s. The simplicity of the games from today's perspective and the use of a visual input of 210 × 160 RGB pixels at 60Hz makes the benchmark sufficiently rich (but still simple) for the AI algorithms of today. After [9] achieved superhuman performance for many of the ALE games, the benchmark became very popular in AI. There are so many platforms, techniques and papers using ALE today that the results on this benchmark are usually analysed when talking about breakthroughs[4] and progress[5] in AI.

We have performed a bibliographical search to find all the papers that include experiments with a wide range of ALE games. We first discarded those techniques that use look-ahead access to a simulator (this is common in search-based approaches [32], [33], [34], but not comparable to humans). This is due to the real-world situation of human players that have to perform with no access to the game other than the screen, and this is the standard for comparison. Look-ahead techniques could have been studied separately (the ones we use in the next section with GVGAI), but what should not be done is to combine results with different rules.

Hence, we will use the results obtained with truly learning approaches (most, but not necessarily all, using reinforcement techniques, usually in conjunction with deep learning). In this category, we are flexible about whether the results include human demonstrations or not ("noop" and "humanstarts" settings). Overall, we integrated about 40 techniques from about a dozen papers covering classical deep reinforcement learning techniques (DQN) [9], [35], as well as specific adaptations to the DQN such as those using *duelling* architectures [36], those with prioritised experience replay [37], or those reducing inherent estimation errors of learning [38]. We also analysed more recent approaches which improve the stability, convergence and runtime of DQN [39], [40], [41], as well as some distributed/parallel versions [42], [43]. Evolution strategies [44], such as a scalable alternative to DQN, were also included. We discarded some papers because they did not include the results for all the 49 games that are most common in many papers. As some results (especially DQN) are reported repeatedly for some papers, we removed all results with a correlation higher than 0.99. In other cases, the results for the same technique with different parameters were kept. We also removed repeated results. Note that some techniques, such as DQN, are used repeatedly, but with different conditions (parameters). Only exact equal results were removed.

B. The General Video Game Playing Competition

The General Video Game AI (GVGAI) competition [16] was one of the first AI competitions featuring a significant number of unseen games within a relatively large problem space (after [45]). As in ALE, this competition focuses on video games, in particular two-dimensional games including classic arcade, puzzles, shooters and many more. The games can also differ in the way players are able to interact with the environment (actions), the scoring systems, the objects that are part of a game or the conditions to end the game. Unlike ALE, GVGAI was created to avoid participants tailoring their submissions to a few well-known games. Instead, participants are pitted against a number of unseen games. Another difference is that controllers are able to access an abstract representation of the game state (so complex perception is not needed) as well as a simulator so that (look-ahead) tree search algorithms and other planning approaches can be used.

Because of this access to the simulator, those controllers based on Monte Carlo Tree Search (MCTS) [46], Rolling Horizon Evolutionary Algorithms (RHEA) [47] as well as hybrids with popular tree search methods have been successful on this benchmark. However, as shown in [30], performance is non-transitive since different controllers play different games best and, thus, no algorithm dominates all the others.

[3] For the sake of reproducibility and transparency, all the code and data is on Github (https://github.com/nandomp/AI_benchmark_analysis).

[4] http://cdn.aiindex.org/2017-report.pdf and https://www.eff.org/ai/metrics

[5] http://www.milesbrundage.com/blog-posts/my-ai-forecasts-past-present-and-future-main-post

Regarding the data, we will work with the scores of 49 games and the 23 controllers (agents) that were submitted to the 2015 GVGAI competition[6] [31]. Each game has 5 levels, and each level was attempted 5 times. This makes a total of $23 \times 49 \times 5 \times 5 = 28175$ trials. For each trial the data includes the win/loss achieved by the controller.

C. Normalisation

In the case of ALE, we have point scores (usual in videogames), which are clearly not commensurate (10,000 points in a game may be low while 50 in another may be high). It is then common to normalise them by human scores (where 0 equals random, and 100 equals human level), usually putting human level as a target for a "successful" or "acceptable" result. On the other hand, for GVGAI we do not have human results as a reference, but a notion of success is given by the "win/loss" values (1:win, 0:loss), which indicate whether the agent 'beat' the game. Each game was attempted 5 times, so win/loss values can be averaged by the number of trials to obtain scores. When analysing both benchmarks, we have linearly scaled their results to z-scores (or standard scores) for both benchmarks so that we can compare the results in a more meaningful way. Then we apply the error function, so $R_{j,i}$ always ranges from 0 to 1.

IV. Task analysis: difficulty and discrimination

Even if we normalise or scale the scores, this does not give us any idea of the difficulty of the tasks or their discriminating power. In these games it is hard to derive a specific notion of difficulty in terms of the features of the games. The number of NPCs, the size of the game or other metrics would not help much to a notion of difficulty[7]. Other general definitions of theoretical difficulty and discrimination [11] would be computationally very expensive. As a result, a populational approach, as represented by IRT, seems the most straightforward approach to derive the difficulty and the discrimination parameters. In order to apply binary IRT, we consider a 'success' as explained in the previous section (above or equal human performance for ALE, and equal or more wins than loses for GVGAI games).

Some tasks for ALE were always below or always above human performance for all techniques (i.e, constant results after normalisation). This implies that the IRT models cannot be fitted for these tasks, so these were excluded for the rest of the analysis. Similarly, some tasks for GVGAI were removed as they have the same result (0 or 1) for all agents.

Once the data is ready, a 2-parameter IRT logistic model (2PL) is learned for each ALE and GVGAI game. We adopt MLE to estimate all the model parameters for all instances and the classifier abilities simultaneously, as usual in IRT. In particular, for generating the IRT models, we used the `ltm` R package[2], using Birnbaum's method, as explained in section II-A. The package `ltm` (as many other IRT libraries) outputs indicators about the goodness of fit, which can be used to quantify the discrepancy between the values observed in the data (items) and the values expected under the statistical IRT model. Item-fit statistics may be used to test the hypothesis of whether the fitted model could truly be the data-generating model or, conversely, we expect the item parameter estimates to be biased. In practice, an IRT model may be rejected on the basis of bad item-fit statistics, as we would not be reasonably confident about the validity of the inferences drawn from it [48]. Apart from the goodness of fit, in order to double-check the results, we recommend re-estimating the parameters with different initial values (seeds) for every model you fit in order to check whether the estimates are consistent. In the present case, none of the estimated models were discarded because of bad item-fit statistics or inconsistency in their results.

Regarding the results, for the ALE games, difficulties range from -10.51 to 8.22, while discriminations range from -0.64 to 58.27. For the GVGAI games, difficulties range from -29.58 to 84.53, while discriminations range from -0.19 to 123.22.

The item parameter that is easiest to understand is difficulty. Because of the MLE estimation method, the value is not equal but well correlated with the percentage of AI techniques that are successful for the game. Intuitively, easy games are solved by almost all techniques, and difficult games are those that are only solved by very able techniques. Figs. 2 (left) and 3 (left) show, respectively, the ICCs of those three most (and least) difficult ALE and GVGAI games with positive discrimination. From those games, the most difficult ALE game seems to be *H.E.R.O*, and *iceandfire.1* for GVGAI. However, we see cases such as *Tennis* (ALE), which has the highest difficulty (8.22) but negative discrimination (-0.13, Fig. 2 right). According to [15], it is a challenging game that requires fairly elaborate behaviour before observing any positive reward, but simple behaviour can avoid high negative rewards by not ever serving, which possibly explains the negative discrimination. Something similar happens with the GVGAI games, where *factorymanager.3* is the third most difficult one (20.9), but its discrimination is negative (-0.15, Fig. 3 right).

The discrimination parameter measures the capability of a game to differentiate between techniques. When applying IRT to evaluate techniques, the slope of an instance can distinguish between strong or weak techniques. Figs. 2 (middle) and 3 (middle) show, respectively, the ICCs of the most discriminating ALE and GVGAI games. From the 41 ALE games analysed, 37 had positive discrimination. Regarding the 154 GVGAI games of different modes analysed, 148 had positive discrimination. For all these the probability of correct responses is positively related to the estimated ability of the techniques. However, negative discriminations were observed for 4 ALE (Fig. 2 right) and 6 GVGAI games (Fig. 3 right).

These "abstruse" cases (most frequently solved by the weakest techniques) are anomalous in IRT, and should be considered with extreme care for the analysis of new AI algorithms. Are these games particularly difficult or are they just useless for evaluation since most able techniques do worse

[6] Results courtesy of Julian Togelius.

[7] The ALE games include the notion of *mode*, which alters the games (e.g, changing the game dynamics, actions, etc.), but it is not linked to difficulty. ALE supports non-commensurate difficulties since its version 0.6 (Sept 2017), but no new systematic results are available with these.

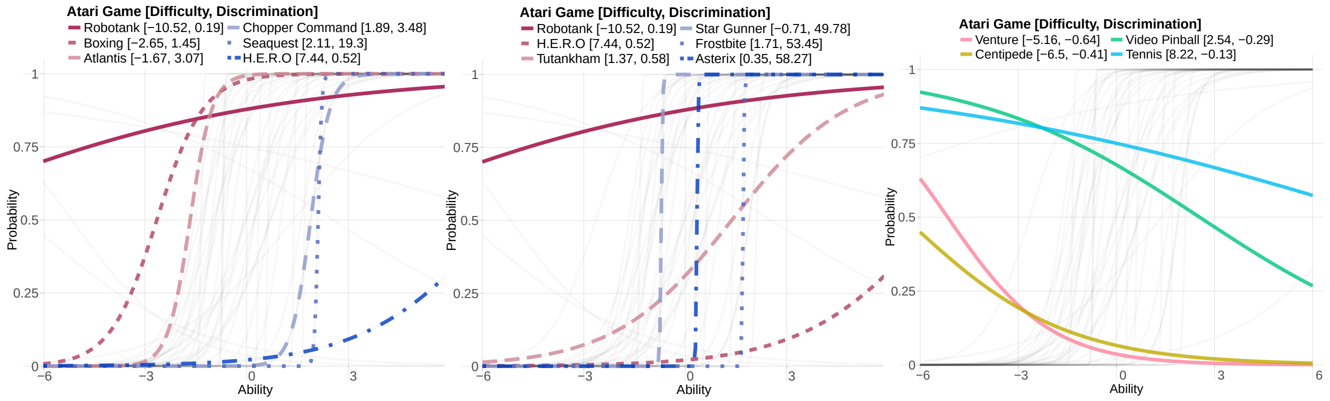

Fig. 2: ICCs of the 3 most (bluish colours) and least (reddish colours) difficult (left) and discriminating (middle) ALE games. In both cases the negative discrimination instances have been filtered out (which are shown on the right plot). All ICC plots from the benchmark are shown in grey.

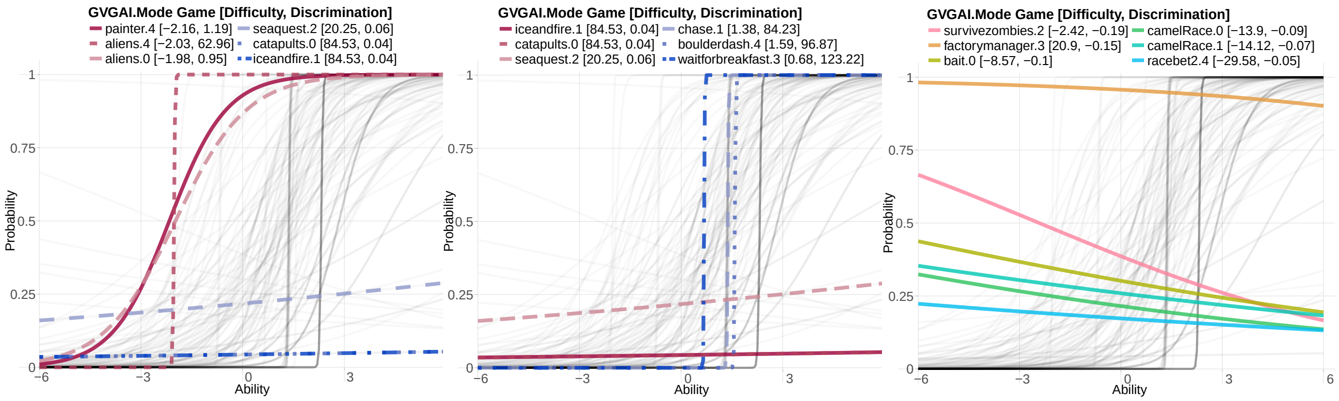

Fig. 3: ICCs of the 3 most (bluish colours) and least (reddish colours) difficult (left) and discriminating (middle) ALE games. In both cases the negative discrimination instances have been filtered out (which are shown on the right plot). All ICC plots from the benchmark are shown in grey.

than those less able ones? Should we restrict our benchmarks to those items to positive (preferably high) discrimination? That depends on the purpose and resources of the evaluation, but what is clear is that, to determine which games are most informative for the analysis of new AI algorithms, difficulty alone is insufficient: we also need to look at discrimination.

In a nutshell, the discrimination parameter provides us an extra dimension to characterise a game. If a game has positive discrimination, it is actually well aligned with ability, and only the good AI techniques obtain good scores. In case a game has a discrimination close to 0, it has a high failure ratio (but happens with good and bad classifiers equally). Finally, if a game has negative discrimination, it is not aligned with ability (with more good AI techniques failing to obtain good scores than bad classifiers). These three cases explain the role of the discrimination parameter when evaluating AI techniques.

V. Technique analysis: ability and generality

As we mentioned in Section II-A, IRT has a dual character: we get information about the items (games) but also about the respondents (AI techniques). IRT estimates a value of ability θ for each AI technique. Unlike average scores, ability takes difficulty into account and is normalised. For instance, if an AI system scores well for difficult games but fails for some easy ones, IRT can give it more value than the opposite situation, depending on their item parameters. Also, IRT penalises those AI techniques that score well in games with negative discriminations.

If there are not many items with negative discrimination, as in our case, ability will be similar to an aggregation of results. Fig. 4 shows scores on the x-axis and ability as graded colour from red to blue, with almost perfectly aligned rankings.

Ability assumes that agents are better at easy instances than they are at difficult instances, but this implies an *uneven* treatment on subpopulations of problems. As discussed in previous sections, this is reasonable. The question is what pattern the "unevenness" (or dispersion) has. If it is actually unrelated to difficulty, we may have pockets of good performance (and pockets of poor performance) with different problem patterns, and the technique would not be very general.

Let us start with the global variance as an indication of regularity, not taking difficulty into account. In Fig. 4, we see that for the same score and ability value, regularity may vary significantly. For instance, for ALE, `PRIOR DQN` and `PRIOR.DUEL` (DQN-based methods from [36], labelled in Fig. 4, top) have similar score and ability, but the former seems

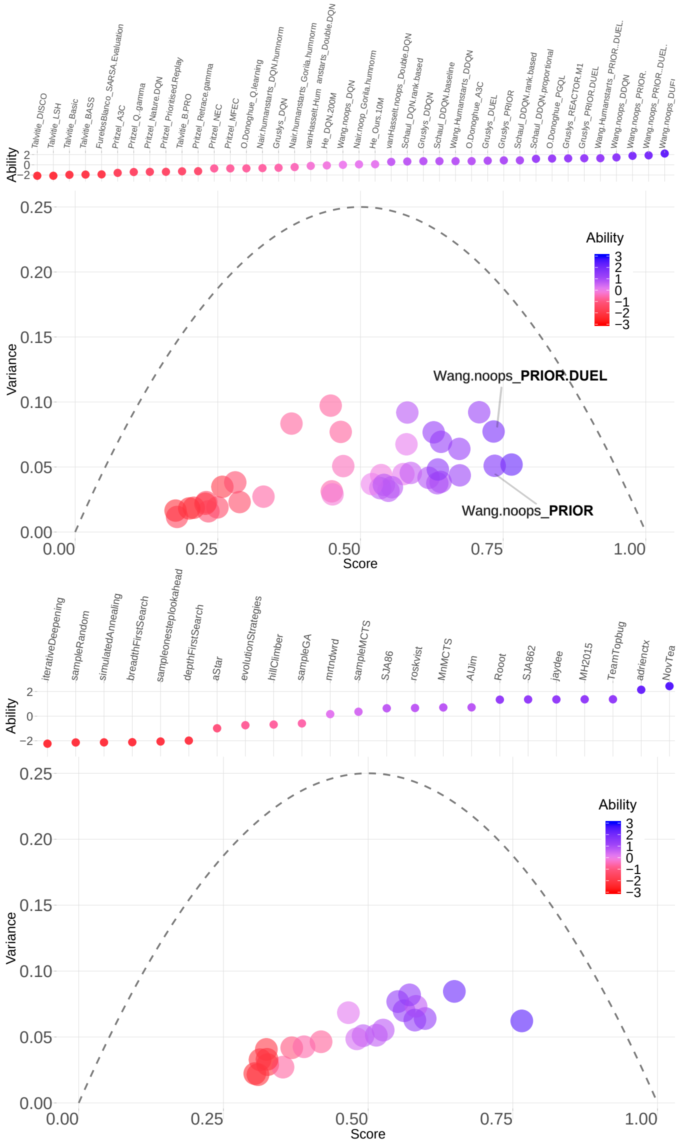

Fig. 4: Variance vs. average normalised score for the AI techniques included in this study (top: ALE, bottom: GVGAI). The IRT abilities are shown with graded colours from red to blue. The dashed grey curve is the variance of a Bernoulli distribution (the worst case).

more regular (with a variance of only 0.05).

But is this actually a measure of generality? Is it so different from the definition of generality in equation 3 that does take difficulty into account? Let us find out. Fig. 5 (top) shows the theoretical agent characteristic curves (ACCs). The IRT models are logistic, with just one parameter varying for the agents (the position, i.e., the ability). If we fix the difficulty, and assume the discriminations a_h are similar for all difficulties h, we have that all slopes are the same. That means that for each technique j, using equations 1 and 3, and applying the variance of a Bernoulli distribution, we have:

$$\gamma_j = \frac{1}{\sum_{h=1}^{h_{max}} \frac{1}{1+e^{-a_h(\theta_j - h)}}(1 - \frac{1}{1+e^{-a_h(\theta_j - h)}})}$$

$$= \frac{1}{\sum_{h=1}^{h_{max}} \frac{e^{-a_h(\theta_j - h)}}{(1+e^{-a_h(\theta_j - h)})^2}}$$

We would have that if θ_j is sufficiently large (so that the variance approaches zero for $h = 1$) and not sufficiently close to h_{max} (so that the variance approaches zero for $h = h_{\max}$), then we would have the same γ_j for each j, as the only thing that changes is location. This is what we see in Fig. 5 (top).

However, the empirical curves, as shown in Fig. 5 (bottom) give us a different view. Some techniques have different slopes, which, together with the discontinuities and monotonicities, give us different generalities, as shown with the areas of the semitransparent ribbons (the smaller the area inside the ribbon at the bin points the higher the generality). See, for instance, the methods `DDQN Rank` (ALE, in olive green colour) or `AIJim` (GVGAI, in pink colour) in Fig. 5 (bottom). Both have high values of generality but low regularity. The reverse is also the case. `NovTea` (GVGAI, in green marine colour) has high regularity but medium-low generality value. All this can be seen more clearly in Fig. 6, where we use the slope of the empirical curves at mean score 0.5 as a proxy for generality (the higher the slope the higher the generality).

These observations are somehow confirmed by the nature of the methods used. If we focus on the GVGAI methods because of its wider variety of algorithms, we see that `AIJim`, a variant of *MCTS* that performs well in related domains (see [46]), has high generality. As for the low generality of `NovTea`, it is an *Iterated Width*-based approach [49], originally a planning technique, which tries to outperform MCTS in GVGAI with *specific* tuning (pruning using novelty test [31]).

Fig. 6 shows there is a correlation between regularity and generality (0.62 and 0.4 for ALE and GVGAI, respectively), but they are different concepts. To see this more clearly, we find a negative correlation between ability and regularity (−0.68 and −0.84 for ALE and GVGAI): most able techniques are those that have higher variance (as we saw in Fig. 4). However, there is no clear correlation between ability and generality (−0.16 and −0.03 for ALE and GVGAI). This gives us the reassuring insight that the progress in these two benchmarks is not significantly due to a loss of generality.

As generality and capability could be increased (or sacrificed) independently, we can ask the question of how generality should be used for competitions, or for encouraging further progress in AI. Several options exist, such as setting a limit of generality in order to qualify for the competition, or integrate generality and ability in some compound metric. It is important to see how generality behaves for all the participants, as if many of them are general then, because of the duality of the parameters, we will have very discriminating items. Actually, maximum generality for all agents implies maximum discrimination for all items and vice versa. Consequently, there is a risk of trying to eliminate items with low discrimination to increase the overall generality (and hence transitivity). It is also important to determine whether low or negative discriminations are caused by issues of a problem or game (e.g., it depends too much on random effects, or strong discontinuities in the scores in terms of the effort needed to solve them, etc.), so that removing it will strengthen the evaluation, or it

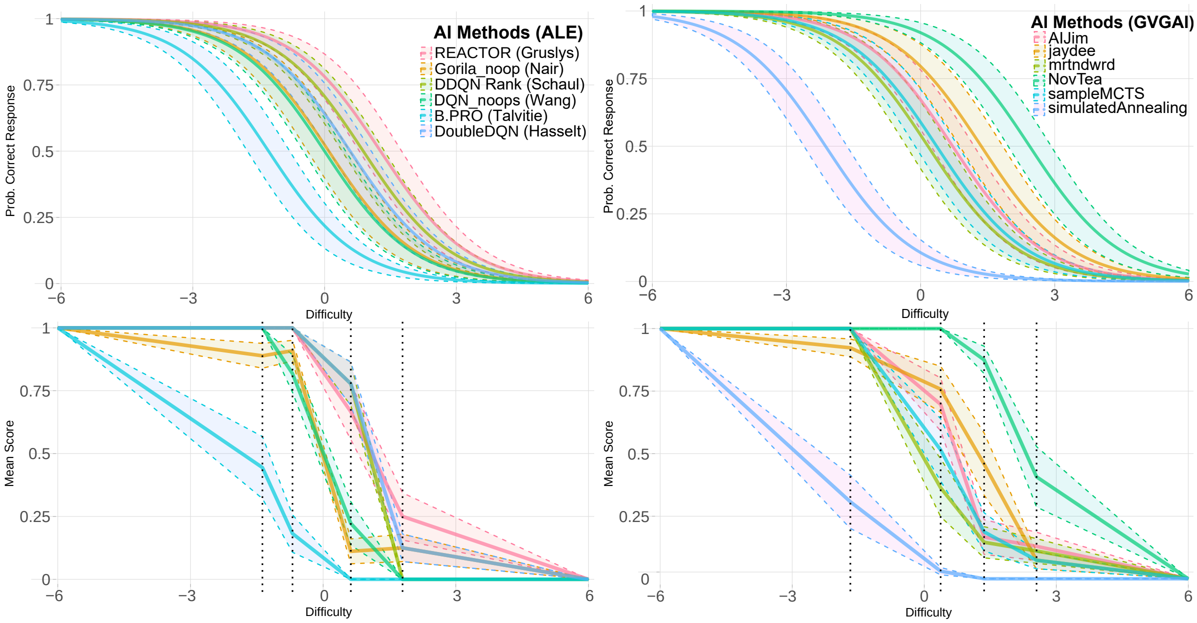

Fig. 5: (top) Theoretical ACCs, i.e., probability of a correct response as a function of the difficulty parameter for an illustrative subset of techniques for ALE (left) and GVGAI (right). $(bottom)$ Empirical ACCs (across bins on the difficulty parameter) for the same subset of techniques. In order to see some progression in the curves (sufficient detail) but still some robustness without spurious peaks, the bins had to contain a minimum number of instances in each interval. Consequently, we set a minimum number of 4 bins with at least 10 examples per bin. Dashed black vertical lines represent the average difficulty values for the instances in each bin. Variance is also represented for each technique (semitransparent ribbon in $\pm({}^{1}\!/{}_{2})\sigma^2$). Negative discrimination instances filtered out.

is because lack of generality of the population of agents, which may be solved by having more general agents in subsequent competitions. Negative discrimination can also suggest that the problem is actually an outlier, very different from the rest, and hence it may be useful to include new problems of similar characteristics to make a benchmark more general.

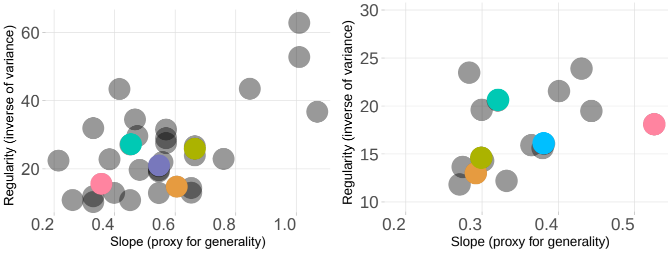

Fig. 6: Slope (proxy for generality) against regularity (inverse of the variance) for all the analysed techniques. Left: ALE, Right: GVGAI. Coloured points represent those systems in Fig. 5.

VI. Discussion

The previous sections have identified four indicators to analyse the results from sets of AI problems (games) and AI systems (players). Difficulty and discrimination have been shown useful for any populational analysis of results in other areas, and they can also play an important role in AI and games. When looking at an AI problem, we see that its difficulty can be caused by several reasons (difficult underlying state representations, varying speeds and types of enemies or goals, etc.). It is however when we analyse the discrimination parameter that we at least can see whether a problem is difficult due to different reasons: *(1)* it is difficult because only the good techniques are able to score well at it, or *(2)* it is difficult because no technique gets it right (having a flat slope). As for ability, while usually related to average performance, it is a normalised parameter, which also takes difficulty into account. Indeed, when the discrimination of a problem is flat or negative, we cannot expect a positive monotonicity between the ability of a method and the probability of a correct response for the item. This would make many poor agents (below the difficulty of the item) getting it right and many good agents (above the item difficulty) getting it wrong.

The techniques in this paper also have some limitations. IRT needs to estimate many parameters, and it can only be applied once we have a good number of results of the respondents (controllers/algorithms) over the items (games). This is why we have chosen ALE and GVGAI to illustrate their use, as we have been able to get a relatively large results table. In the case of ALE, this can be done when the benchmark has a sufficient large number of problems and has attracted sufficient attention to get many different techniques being evaluated on it. For competitions such as GVGAI, once the participants of the competition have submitted their controllers to the game sets, the results can be used o obtain both the final rankings and the IRT parameters.

But, *once the parameters are estimated*, one can obtain the ability and generality for a single new agent, especially in the context of adaptive testing, without a re-estimation of all the item parameters. The obtained ability is defined on a normal scale, which is more informative and illustrative than a ranking (we can see if the winner is much better than the runner-up, for instance, in the context of the population). In general, if two editions of the same competition (or two rounds of the same competition) use the same items (or we just analysed the common items), the parameters of the first can be used to evaluate the results of the second without a re-estimation of the item parameters. Indeed, this is the recommendation when a competition is held for several rounds or editions: calculate the item parameters, and use them for the evaluation of new techniques. From time to time, the parameters can be re-evaluated when the agent population has changed significantly.

Another feature of IRT and the four parameters introduced is that they are sample dependent. Of course we expect that things will change when we change the items (so the benchmark is measuring a different thing), but it is hard to understand that in single-player games the ability of an agent changes if some other new agents are included in the pool, like in adversarial games such as chess and Go (although this is not always seen as a negative thing, see e.g., [50]). The reason why the parameters of one agent are affected by other agents is that the notion of discrimination/difficulty in IRT is populational, so when we change the population (e.g., improving an agent), the obtained parameters will differ. Similarly, in this paper, IRT is also used to estimate the ability of a new technique, so it is also populational. As the measure of generality uses difficulty, it also becomes populational. In these conditions, the way in which agents progress can make an impact on several metrics. For instance, if agents improve on the easy items, they will become more general. The generality of good agents will help contribute making discrimination positive for more items, since discrimination is negatively affected by good agents performing poorly on easy items.

One possibility to make ability and generality less prone to changes in the population is to calibrate some parameters with the use of agents for which we have theoretical expectations, especially if we can change some of their configurations or we can tune them by some of their hyper-parameters. For instance, in [23] the IRT models were derived for a random forest classifier, whose ability was gauged by the number of trees that were used in the multiclassifier.

A more principled alternative to the sample dependence issue is to make difficulty non-populational, using some notion of difficulty that derives from the tasks themselves. With this we would have a metric of generality (and capability) that would not depend on the other agents, and would not change whenever the agent population changes.

Overall, it is important to give some methodological take-aways for those general situations where we have to estimate the four parameters. First, we should wait to have a wide range of problems or games that are representative of what we want to evaluate or set as a benchmark. Adding many small variations of the same problem will affect the metrics of ability but most especially generality, as they can create clusters for which the agents can specialise. Second, we will have to wait until an important number of techniques have been applied to the benchmark, either through a literature meta-review (as we have done here for ALE) or from the results of a competition (as we have done here for GVGAI). Third, publishing the estimated parameters for problems and agents publicly is key for other researchers to use them in the evaluation of new agents or the definition of new benchmarks.

The most novel contribution in this paper is the introduction of the generality indicator, which becomes meaningful precisely because of the use of difficulty: different levels of difficulty imply different distribution of results for a method. Generality should not be understood in terms of the global variance. This paves the way for a better understanding of the G in AGI (Artificial General Intelligence), and other domains in AI that are aiming at more general-purpose AI systems.

The four indicators, which can be obtained easily with the code we provide, can also be particularly meaningful from the viewpoint of AI benchmarks [12] and (videogame) competitions [51], [19], [17], [18], as they provide a proper insight of what the games (and other tasks) are evaluating, and even whether they can be considered useless for a good evaluation in the benchmark. Also, taking into account the long training and evaluation times of recent computing-demanding techniques, any understanding of what the key games are (in order to reduce the size of the benchmark, specially in the hyperparameter search) can imply an important contribution for AI researchers. With the generality metric, we also have an extra parameter that can give us more information about whether the negative discriminations are caused by some pathologies of the item or because the agent population has insufficient generality.

On the other hand, we can also obtain further insight of those AI systems addressing these games (beyond their aggregated performance). It is important to determine whether the new techniques, especially those that rely on long training stages with a game, are coping well generally, and not only for a pocket of problems, but failing in some situations. This is relevant for both the AI and video game communities.

Of course, other models are possible, with more or less parameters, and estimated in different ways. For instance, we want to derive more sophisticated 4-parameter IRT models using continuous inputs. In general, the specific number of parameters will depend on whether some of the variables are given or not (for instance, if we have a theoretical notion of difficulty). Overall, the most important insight is this dual view between AI tasks and AI systems.

Acknowledgments

This material is supportedby the U.S. Air Force Office of Scientific Research under award number FA9550-17-1-0287, the EU (FEDER) and the Spanish MINECO under grant TIN 2015-69175-C4-1-R, the Generalitat Valenciana PROMETEOII/2015/013. F. Martínez-Plumed was also supported by INCIBE (Ayudas para la excelencia de los equipos de investigación avanzada en ciberseguridad). J. Hernández-Orallo also received a Salvador de Madariaga grant (PRX17/00467) from the Spanish MECD and a BEST grant (BEST/2017/045) from the GVA for research stays at the CFI, and the FLI grant RFP2-152.

References

[1] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959. 1

[2] M. Campbell, A. J. Hoane Jr, and F.-h. Hsu, “Deep Blue,” *Artificial Intelligence*, vol. 134, no. 1-2, pp. 57–83, 2002. 1

[3] D. Ferrucci, A. Levas, S. Bagchi, D. Gondek, and E. T. Mueller, “Watson: Beyond jeopardy!” *Artificial Intelligence*, vol. 199, pp. 93–105, 2013. 1

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, 2016. 1

[5] M. Lichman, “UCI machine learning repository,” 2013. [Online]. Available: http://archive.ics.uci.edu/ml 1

[6] M. Vallati, L. Chrpa, M. Grześ, T. L. McCluskey, M. Roberts, S. Sanner *et al.*, “The 2014 international planning competition: Progress and trends,” *AI Magazine*, vol. 36, no. 3, pp. 90–98, 2015. 1

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015. 1

[8] J. Hernández-Orallo, “Evaluation in artificial intelligence: From task-oriented to ability-oriented measurement,” *Artificial Intelligence Review*, vol. 48, no. 3, pp. 397–447, 2017. 1

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015. 1, 5

[10] G. N. Yannakakis and J. Togelius, “A panorama of artificial and computational intelligence in games,” *IEEE Trans. on Computational Intelligence and AI in Games*, vol. 7, no. 4, pp. 317–335, 2015. 1

[11] J. Hernández-Orallo, *The Measure of All Minds: Evaluating Natural and Artificial Intelligence*. Cambridge University Press, 2017. 1, 2, 6

[12] J. Hernández-Orallo, M. Baroni, J. Bieger, N. Chmait, D. L. Dowe, K. Hofmann, F. Martínez-Plumed, C. Strannegård, and K. R. Thórisson, “A new AI evaluation cosmos: Ready to play the game?” *AI Magazine*, vol. 38, no. 3, 2017. 1, 2, 10

[13] J. Hernández-Orallo and D. L. Dowe, “Measuring universal intelligence: Towards an anytime intelligence test,” *Artificial Intelligence*, vol. 174, no. 18, pp. 1508–1539, 2010. 1

[14] D. Castelvecchi, “Tech giants open virtual worlds to bevy of ai programs,” *Nature News*, vol. 540, no. 7633, p. 323, 2016. 1

[15] M. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, “The arcade learning environment: An evaluation platform for general agents,” in *24th Int. Joint Conf. on Artificial Intelligence*, 2015. 1, 5, 6

[16] D. Perez-Liebana, S. Samothrakis, J. Togelius, T. Schaul, S. M. Lucas, A. Couëtoux, J. Lee, C.-U. Lim, and T. Thompson, “The 2014 general video game playing competition,” *IEEE Trans. on Computational Intelligence and AI in Games*, vol. 8, no. 3, pp. 229–243, 2016. 1, 5

[17] J. Togelius, N. Shaker, S. Karakovskiy, and G. N. Yannakakis, “The mario ai championship 2009-2012,” *AI Magazine*, vol. 34, no. 3, pp. 89–92, 2013. 2, 10

[18] J. Renz *et al.*, “Aibirds: The angry birds artificial intelligence competition.” in *AAAI*, 2015, pp. 4326–4327. 2, 10

[19] S. Ontanón, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss, “A survey of real-time strategy game ai research and competition in starcraft,” *IEEE Trans. on Computational Intelligence and AI in games*, vol. 5, no. 4, pp. 293–311, 2013. 2, 10

[20] A. Khalifa, M. Preuss, and J. Togelius, “Multi-objective adaptation of a parameterized GVGAI agent towards several games,” in *Int. Conf. on Evolutionary Multi-Criterion Optimization*, 2017, pp. 359–374. 2

[21] S. E. Embretson and S. P. Reise, *Item response theory for psychologists*. L. Erlbaum, 2000. 2

[22] F. Martínez-Plumed and J. Hernández-Orallo, “Ai results for the Atari 2600 games: difficulty and discrimination using IRT,” *2nd Int. WS Evaluating General-Purpose AI, Melbourne, Australia*, 2017. 3

[23] R. B. Prudêncio, J. Hernández-Orallo, and A. Martínez-Usó, “Analysis of instance hardness in machine learning using item response theory,” in *2nd Int. WS on Learning over Multiple Contexts. Porto*, 2015. 3, 10

[24] F. Martínez-Plumed, R. B. Prudêncio, A. Martínez-Usó, and J. Hernández-Orallo, “Making sense of item response theory in machine learning,” in *European Conference on Artificial Intelligence, ECAI*, 2016, pp. 1140–1148. 3

[25] A. Birnbaum, *Statistical Theories of Mental Test Scores*. Reading, MA.: Addison-Wesley, 1968, ch. Some Latent Trait Models and Their Use in Inferring an Examinees Ability. 3

[26] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997. 4

[27] D. H. Wolpert, “What the no free lunch theorems really mean; how to improve search algorithms,” in *Santa Fe Institute*, 2012. 4

[28] D. Ashlock, D. Perez-Liebana, and A. Saunders, “General video game playing escapes the no free lunch theorem,” in *IEEE Conf. on Computational Intelligence and Games*. IEEE, 2017, pp. 17–24. 4

[29] F. Martínez-Plumed and J. Hernández-Orallo, “Analysing results from ai benchmarks: Key indicators and how to obtain them,” *arXiv preprint arXiv:1811.08186*, 2018. 4

[30] T. S. Nielsen, G. A. Barros, J. Togelius, and M. J. Nelson, “Towards generating arcade game rules with vgdl,” in *IEEE Conf. Computational Intelligence and Games*, 2015, pp. 185–192. 5

[31] P. Bontrager, A. Khalifa, A. Mendes, and J. Togelius, “Matching games and algorithms for general video game playing,” in *12th AI and Interactive Digital Entertainment Conf.*, 2016, pp. 122–128. 5, 6, 8

[32] Y. Naddaf, “Game-independent ai agents for playing atari 2600 console games,” *MSc, Dep. of Comp. Science, University of Alberta*, 2010. 5

[33] N. Lipovetzky, M. Ramirez, and H. Geffner, “Classical planning with simulators: results on the atari video games,” in *Heuristics and Search for Domain-independent Planning*, 2015. 5

[34] A. Shleyfman, A. Tuisov, and C. Domshlak, “Blind search for atari-like online planning revisited,” *Heuristics and Search for Domain-independent Planning*, p. 85, 2016. 5

[35] D. Furelos-Blanco, *Learning and Generalization in Atari Games*. Bachelor Thesis, Universitat Pompeu Fabra, 2015. 5

[36] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, “Dueling network architectures for deep reinforcement learning,” *arXiv preprint arXiv:1511.06581*, 2015. 5, 7

[37] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, “Prioritized experience replay,” *arXiv preprint arXiv:1511.05952*, 2015. 5

[38] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning.” in *AAAI*, 2016, pp. 2094–2100. 5

[39] F. S. He, Y. Liu, A. G. Schwing, and J. Peng, “Learning to play in a day: Faster deep reinforcement learning by optimality tightening,” *arXiv preprint arXiv:1611.01606*, 2016. 5

[40] A. Pritzel, B. Uria, S. Srinivasan, A. Puigdomènech, O. Vinyals, D. Hassabis, D. Wierstra, and C. Blundell, “Neural episodic control,” *arXiv preprint arXiv:1703.01988*, 2017. 5

[41] E. Talvitie and M. Bowling, “Pairwise relative offset features for atari 2600 games,” in *WS on Learning for General Competency in Video Games*, 2015. 5

[42] A. Gruslys, M. G. Azar, M. G. Bellemare, and R. Munos, “The reactor: A sample-efficient actor-critic architecture,” *arXiv preprint arXiv:1704.04651*, 2017. 5

[43] A. Nair, P. Srinivasan, S. Blackwell, C. Alcicek, R. Fearon, A. De Maria, V. Panneershelvam, M. Suleyman, C. Beattie, S. Petersen *et al.*, “Massively parallel methods for deep reinforcement learning,” *arXiv preprint arXiv:1507.04296*, 2015. 5

[44] T. Salimans, J. Ho, X. Chen, and I. Sutskever, “Evolution strategies as a scalable alternative to reinforcement learning,” *arXiv preprint arXiv:1703.03864*, 2017. 5

[45] M. Genesereth, N. Love, and B. Pell, “General game playing: Overview of the AAAI competition,” *AI magazine*, vol. 26, no. 2, p. 62, 2005. 5

[46] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez-Liebana, S. Samothrakis, and S. Colton, “A survey of monte carlo tree search methods,” *IEEE Trans. on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012. 5, 8

[47] D. Perez-Liebana, S. Samothrakis, S. Lucas, and P. Rohlfshagen, “Rolling horizon evolution versus tree search for navigation in single-player real-time games,” in *15th Conf. on Genetic and evolutionary computation*. ACM, 2013, pp. 351–358. 5

[48] A. Maydeu-Olivares, “Goodness-of-fit assessment of item response theory models,” *Measurement: Interdisciplinary Research and Perspectives*, vol. 11, no. 3, pp. 71–101, 2013. 6

[49] N. Lipovetzky and H. Geffner, “Width and serialization of classical planning problems,” in *Proceedings of the 20th European Conference on Artificial Intelligence*. IOS Press, 2012, pp. 540–545. 8

[50] D. Balduzzi, K. Tuyls, J. Perolat, and T. Graepel, “Re-evaluating evaluation,” *arXiv preprint arXiv:1806.02643*, 2018. 10

[51] P. Hingston, *Believable Bots: Can Computers Play Like People?* Springer, 2012. 10