Further Details on Predicting IRT Difficulty

Fernando Martínez-Plumed[1,2], **David Castellano-Falcón**[2], **Carlos Monserrat**[2], **José Hernández-Orallo** [2,3]

[1] European Commission, Joint Research Centre
[2] Valencian Research Institute for Artificial Intelligence (VRAIN), Universitat Politècnica de València
[3] Leverhulme Centre for the Future of Intelligence, University of Cambridge
fernando.martinez-plumed@ec.europa.eu, dacasfal@upv.es, cmonserr@dsic.upv.es, jorallo@upv.es

This supplementary material serves as technical appendix of the paper *When AI Difficulty is Easy: The Explanatory Power of Predicting IRT Difficulty* (Martínez-Plumed et al. 2022), published in *The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)*. The following sections give detailed information about 1) data gathering for benchmarks; 2) IRT properties and methodology followed; 3) learning models configuration and hyperparameter setting; 4) differences between difficulty prediction and class prediction; 5) the deployment and results of alternative approaches for difficulty estimation; 6) specifics and results using a *generic* difficulty metric in different applications and 7) extended IRT applications.

Benchmarks

As an exception to the instance-wise result problem, we find platforms such as OpenML (Vanschoren et al. 2014), a repository in which AI researchers and practitioners can share data sets and results in as much detail as possible. The platform also provides several sets of curated, reference datasets. OpenML-CC18 is one of those reference sets that meet several requirements in order to compile a carefully curated selection from the thousands of datasets on OpenML. From this set in OpenML, we extracted the benchmarks that had instance-wise results of a good number of models, leading to the first 14 rows in Table 1.

For other AI domains analysed in this work (automated reasoning and NLP), we relied on data scraping techniques and personal communications in order to gather the data. Concretely, for the tptp benchmark, we obtained the data from The CADE ATP System Competition website[2], which evaluates the performance classical logic order ATP systems. The evaluation is in terms of the average runtime for problems solved and the data we extracted included the number of formulae, atoms, connectives, predicates, functors and variables, and maximum depth for formulas and terms. For its part, for the sat benchmark, we extracted the data from the Pseudo-Boolean Competition website[3], also via data scraping techniques. The data extracted include syntactic elements (number of variables, number of constraints, clauses, coefficients, etc.), whether the problem is satisfiable or not and the category to which it belongs. Finally, for the NLP bechmarks SST2 and IMDB), we got the data by contacting the authors in (Mishra and Arunkumar 2021), who kindly provided instance level results for the test sets of the previous benchmarks.

IRT for difficulty estimation

In general, IRT has certain properties appropriate to our objectives:

- Consistent and independent ability estimation. IRT assumes that the ability of respondents does not change while taking a test, and each problem is independent of other problems in the same test (De Ayala 2013).
- Great explanatory capabilities. IRT parameters allow to describe both the difficulty (and discrimination) of problems and a latent parameter of the respondents (ability).
- Independency from the problems/respondents. IRT works well regardless of the difficulty of problems and the ability of respondents.

At the same time, IRT overcomes certain limitations of other approaches for difficulty estimation such as Classical Test Theory (Magno 2009) (e.g., inconsistency across samples of items and less stability), pretesting and expert judgement (Attali et al. 2014) (e.g, subjectivity, inefficiency and item exposure), the proportion of correct responses (e.g., very sensitive to abstruse respondents and the sample choice) or the Elo rating system (Elo 1978) (e.g., useful for system pairs, matches, but not for problem-system characterisation).

Note that the estimation of item difficulty can be derived *intrinsically* from the properties of an instance (e.g., size, number of components, noise, distortions, etc.) or the resources that are expected to solve it (e.g., working memory, primitives, etc.); or it can also be derived *extrinsically* from —and so being dependent on— the results of one or more systems. When only one system is used, then difficulty is just the probability of failure for that system on the instance. When several systems are used —and IRT follows this approach—, easy instances are those that are solved by most of the systems in a population. That still allows a particular system to fail on a pocket of easy instances, or succeed on a pocket of difficult instances. These insightful situations would not be possible if difficulty were only derived from the system we want to analyse. The IRT-based approach followed in this paper to infer difficulty depends on the other items and systems. The more diverse the data

[2] http://www.tptp.org/CASC/

[3] http://www.cril.univ-artois.fr/PB10/

and systems are used in the analysis the more generic the difficulty will be, and more independent of any new system we would like to analyse with that difficulty metric.

Item difficulty and confidence measures It is also worth paying attention to the relationship between item difficulty and the confidence of a learning system. Confidence should usually be high for very easy and very difficult items, and low for items of intermediate difficulty. This is actually exploited by IRT and its applications, such as adaptive testing, which focuses on the examples of intermediate difficulty. But the relation may be less straightforward in practice and may deserve a deeper analysis, especially if we also want to analyse how the reliability of the difficulty estimator relates to the confidence of the system. Both confidence and unknown unknowns can be analysed under the umbrella of aleatoric uncertainty (i.e., uncertainty due to the natural stochasticity or noise of observations) and epistemic uncertainty (i.e., uncertainty due to limited data or knowledge). Unknown unknowns are an extreme case of epistemic uncertainty, and they affect the systems themselves from which IRT difficulties (1) are derived, and also the difficulty estimator (2) that we build from this data. We may use use this distinction between aleatoric and epistemic uncertainty to clarify the connection between difficulty (at moments 1 and 2), confidence and unknown unknowns. For instance, we may gauge the ability of a model to predict difficulty. This is what (2) is about and depends more on epistemic uncertainty.

Specific IRT methodology

In practice, for generating the IRT models, we used the `MIRT` R package (Chalmers 2012), using Birnbaum's method (Birnbaum 1968). The package `MIRT` (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. 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 (Maydeu-Olivares 2013). In the present case, none of the estimated models were discarded because of bad item-fit statistics or inconsistency in their results.

Following the recommendations from (Martínez-Plumed et al. 2019) for the sake of result variability, apart from the original AI systems in each benchmark, we also introduced some artificial systems: (1) an always-wrong model, (2) an always-right model and, in classification problems, (3) a model that predicts a random class using the class prior. These synthetic systems, for which we know their abilities, are very useful as indicators (e.g., to see how calibrated difficulty and ability are).

Also, it should be noted that the inference in IRT does not scale well for many instances and/or respondents (i.e., the algorithm may not finish or estimates may not be accurate). While variational inference-based IRT approaches (Wu et al. 2020) have proven useful in scenarios dealing with tens or hundreds of thousands of respondents (such as in the PISA international assessments), this is not our case. On the contrary, we are dealing with benchmarks up to tens of thousands of items (and up to a few thousands of AI systems as respondents). In this regard, IRT assumes that the ability of respondents and the difficulty of problems are invariant to respondents and problems being used in estimation (De Ayala 2013). That means we may iterate over subsets of items because the population of AI systems does not change and thus we can estimate the item parameters more efficiently. We actually do this for all the folds, so that we cover the whole benchmark dataset.

Difficulty estimation model configurations

Feature-value becnhmarks We trained five different classical and state-of-the-art regression techniques using default hyperparameters (Fernández-Delgado et al. 2014). Namely, we used stepwise regression as a mere baseline (Derksen and Keselman 1992), elastic nets (*elasticNet*) (Zou and Hastie 2005), gradient boosting machines (*gbm*) (Friedman 2002), k-nearest neighbours (*knn*) (Fix and Hodges 1989) and random forests (*rf*) (Breiman 2001). We used the `caret` R package (Kuhn 2008) to streamline the process for learning and evaluating the above predictive models.

Image-based benchmarks We tried several modern CNN architectures, finetuning and making custom adjustments for each model and benchmark (since each model architecture is different, there is no boilerplate finetuning code that will work in all scenarios). We finally selected a subset of CNN architectures pretrained on the 1000-class Imagenet dataset (Deng et al. 2009) such as *VGG16* (Simonyan and Zisserman 2015), *ResNet-50* (He et al. 2016) and *Densenet121* (Huang et al. 2018) because of their performance. As we are dealing with a regression task, apart from updating all of the model's parameters for the new tasks at hand, we changed also the output layer by a 2-layer Multilayer perceptron (MLP) as projection head (with an intermediate dimension of 64 neurons) as non-linear function approximator for regression. This last part improves and adds stability to the predictions (Chen et al. 2020). We used *smooth L1* as loss function, optimised using the *Adam* algorithm (Kingma and Ba 2014) with learning rates between 0.001, and 0.1 depending of the benchmark and model used. We used a batch size of 1024 and 50 epochs. Furthermore, we start with a linear warm up procedure until reaching the maximum learning rate in the the first 5 epochs. Afterwards, we apply a cosine decay schedule without restarts until convergence (Loshchilov and Hutter 2017). For each image-based benchmark, we followed the same preprocessing for the input data. We normalised the data either using the mean and the standard deviation per image when using 1-channel images. For 3-channel images, the normalisation process is computed per image channel (red, green and blue) using the whole input dataset (Imagenet in this case). We did not use any data augmentation approach since this would modify the estimation of the difficulty values.

Text-based benchmarks We used the following pretrained transformer models (Vaswani et al. 2017): *T5 (small)* (Raffel et al. 2019), an encoder-decoder model pretrained on a multi-task mixture of unsupervised and supervised (text-to-text format-transformed) tasks; *BERT (sent)* (Devlin et al. 2018), a BERT-based multilingual uncased model finetuned for sentiment analysis on product reviews in different languages; and *BERT (base)* (Devlin et al. 2018), a pretrained model on English language (uncased) using a masked language modelling (MLM) (Sinha et al. 2021) objective. All these models were used following the idea of using their encoder architecture to learn a semantically meaningful latent space which will be then used by an MLP for obtaining the difficulty estimation. In this case, the learning rate we used was fixed to 5×10^{-5} and we followed the same warm-up procedure as before. We set the batch size to 20. Finally, for each model we tokenised the input text by using their own default tokeniser approach.

Original task vs. difficulty estimation

In order to understand better the difference between solving the original problem and solving the difficulty estimation problem, we are going to analyse how the features are used. Let us choose the well-known letter benchmark (Frey and Slate 1991), which has medium average **1PL** difficulty according to Fig. 2 for which the model is very successful (low NRMSE and high correlation in Table 2a). We also choose a feature vector representation because we can use exactly the same technique for the difficulty estimation and the original task, as well as the ease of analysis of the original features (for images, we have that attention maps for regression are different and more convoluted than those for classification, see e.g., Gupta et al. 2021).

The goal of this benchmark is to identify the 26 capital letters in the English alphabet from B/W rectangular pixel displays, but images are converted into a feature-vector representation with 16 numerical features. Since the best **PL1** difficulty estimator according to Table 2a is *rf*, we use it as well for generating a classifier for the problem, with the same by-default hyperparameters. We use the Gini Importance or Mean Decrease in Impurity (MDI) (Breiman 2001) to compute each feature importance.

If we look at most important features of the difficulty estimator in Fig. 3 (left) (in the main paper), we see that the variable `y.ege` (i.e., the mean edge count left to right) is more than twice as important than `x.ege` or `onpix` (i.e., total ”on” pixels in the character image), and four times more important than `x2bar` or `y2bar` (i.e., mean x/y variance) when predicting difficulty. This means that the difficulty of a character image heavily depends on the mean number of edges (an “on” pixel immediately to the right of either an “off” pixel or the image boundary) encountered when making systematic scans from left to right at all vertical positions within the box.

On the contrary, when analysing the variable importance of the *rf* classifier on the original task in Fig. 3 (right), the `y.ege` is no longer the most important variable but the second one after `x.ege`. This time, the sum of the vertical positions of edges (`y.ege`) is the most important one for the classification of character images. Also, it is noticeable that the differences between the first five variables by importance are not as prominent as in the previous case.

Similar discrepancies can be found in machine learning benchmarks, which may partly explain why in some cases the original task is easy but estimated difficulty is hard and vice versa. It is also important to note that for many existing datasets in the literature, the existing features have been created or selected to be predictive for the task, but not necessarily predictive for our new purpose of estimating difficulty. For instance, the level of blur in an image may be a very distinctive feature for estimating difficulty but possibly useless for classifying images. This suggests that datasets should be enhanced in the future with difficulty-relevant features for the sake of better evaluation.

Alternative approaches for difficulty estimation

Here we study the fits of alternative approaches to the One-Parameter Logistic (**1PL**) model: the two-parameter logistic (**2PL**) item response theory models, which estimates parameters for both the difficulty and discrimination of dichotomous items; and, a straightforward alternative for estimating item difficulty, the mean error per instance (**AvE**).

Difficulty prediction In Fig. 7 we show the estimated difficulty distribution per benchmarks using **2PL** models. As explained in the experimental setting, we crop those abnormal difficulty values ($> |6|$), removing, on average, 3% of instance per benchmark which is higher than the 0.5% removed when using **1PL** models due an overall worse goodness-of-fit. For its part, in Fig. 8 we show the difficulties computed through **AvE** per instance and benchmark. In this case, instances where the majority of AI systems addressing them fail (errors close to 1) are considered more difficult than those where only a few systems get them wrong (errors close to 0).

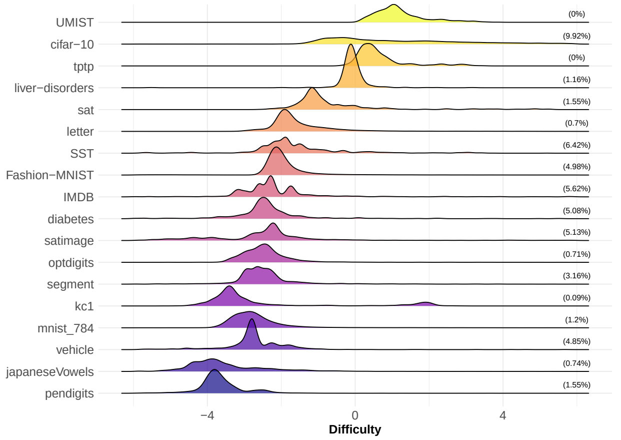

Figure 7: **2PL** difficulty distribution per dataset (percentage of difficulties outside the $[-6, 6]$ range indicated in the plots). Benchmarks sorted by average difficulty.

In the first analysis we performed we used the estimated difficulties per instance by these two approaches and, using

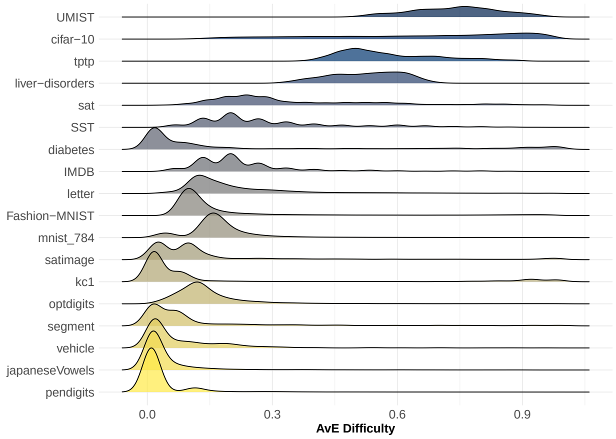

Figure 8: **AvE** difficulty distribution per instance and dataset. Benchmarks sorted by average accuracy.

the original observable features from the benchmarks, we traiedn five different classical and state-of-the-art regression techniques using default parameters. We followed the same methodology explained in the experimental setting section. We limited our analysis to those benchmarks in Table 1 following a feature vector representation for efficiency reasons. In Tables 5 and 6 we show the NRMSE results for the validation and test set for these benchmarks. We also compute pairwise comparisons in the validation set to identify significant differences between models (Wilcoxon test (Cuzick 1985)). As in the previous experiments using a **1PL** model for difficulty estimation (Table 2a), *rf* is again better than the rest of models in most cases. However, the error obtained by the models trained with difficulty values from **1PL** models are, overall, lower compared to those models trained with difficulty values from 2PL models. Note that NRSME values between **AvE** and IRT models are not comparable. We will analyse the results of the different approaches together further on, looking at the correlations.

Dataset	elasticNet		gbm		knn		lm		rf	
	NRMSE	Corr	NRMSE	Corr	NRMSE	Corr	NRMSE	Corr	NRMSE	Corr
diabetes	0.97±0.04	0.60	0.94±0.04	0.64	1±0.06	0.48	0.97±0.04	0.59	0.92±0.06	0.67
kc1	0.90±0.03	0.69	0.90±0.03	0.69	0.96±0.03	0.61	1.00±0.28	0.68	0.87±0.03	0.70
liver-disorders	1.01±0.05	0.64	1.01±0.05	0.68	1.10±0.07	0.77	1.00±0.04	0.64	1.01±0.05	0.88
japaneseVowels	0.90±0.05	0.71	0.87±0.05	0.73	0.96±0.03	0.56	0.90±0.05	0.71	**0.67±0.05**	0.85
letter	0.86±0.02	0.57	0.86±0.02	0.58	**0.64±0.02**	0.89	0.86±0.02	0.57	0.70±0.02	0.90
optdigits	0.94±0.07	0.00	0.96±0.07	0.09	0.86±0.06	0.04	0.94±0.07	0.00	0.85±0.07	0.55
pendigits	0.78±0.08	0.47	0.77±0.08	0.42	0.55±0.08	0.76	0.78±0.07	0.47	0.53±0.08	0.81
satimage	0.95±0.03	0.47	0.88±0.03	0.47	0.82±0.04	0.67	0.95±0.03	0.46	**0.77±0.03**	0.69
segment	0.92±0.04	0.70	0.90±0.04	0.67	0.83±0.05	0.68	0.92±0.04	0.69	**0.63±0.03**	0.82
vehicle	0.78±0.07	0.58	0.78±0.09	0.72	0.81±0.07	0.90	0.79±0.07	0.57	**0.70±0.09**	0.88
tptp	1.13±0.38	0.25	0.91±0.05	0.33	0.92±0.08	0.42	1.10±0.33	0.18	**0.85±0.05**	0.59
sat	0.78±0.11	0.75	0.75±0.08	0.75	0.72±0.08	0.72	0.85±0.24	0.75	**0.58±0.08**	0.84

Table 5: NRMSE results and Spearman correlations for those benchmarks in Table 1. Interpretation as in Table 2. Difficulty values obtained following a two-parameter IRT model (**2PL**).

Discrimination analysis By taking advantage of the discrimination parameter estimated by the *2PL* models, we may also analyse its possible impact on the learned difficulty functions. The discrimination parameter (slope) is a measure of the capacity of an item to differentiate between individuals (AI systems). Therefore, when applying IRT to evaluate AI systems, the slope of an instance can be used to indicate if the instance is useful to distinguish between strong or weak classifiers for a problem. From the benchmarks analysed in this work, $6.8\% \pm 13.8$ instances have negative discrimination values (negative slope) when using **2PL** models. In these cases, the probability of correct responses is negatively related to the estimated ability of the classifiers. This means that these instances are most frequently succesfully addressed by the weakest AI systems. These cases are anomalous in IRT (usually referred to as "abstruse" or "idiosyncratic" items). But in the context of AI, these are precisely the instances that may be most useful to identify particular situations. For example, if two instances 1 and 2 in a binary classification problem have exactly the same features but belong to different classes, then $P(U_{1j} = 1|\Theta_j) = 1 - P(U_{2j} = 1|\Theta_j)$. In this situation, one of the instances may have been wrongly labelled, which can result in a negative-slope ICC.

Dataset	elasticNet		gbm		knn		lm		rf	
	NRMSE	Corr	NRMSE	Corr	NRMSE	Corr	NRMSE	Corr	NRMSE	Corr
diabetes	1.06±0.04	0.52	1.03±0.06	0.56	1.11±0.05	0.40	1.07±0.04	0.50	1.01±0.05	0.60
kc1	1.00±0.05	0.62	1.00±0.05	0.62	1.05±0.06	0.54	1.01±0.05	0.61	0.96±0.05	0.63
liver-disorders	1.10±0.04	0.57	1.11±0.05	0.62	1.20±0.07	0.69	1.11±0.04	0.57	1.12±0.05	0.82
japaneseVowels	1.00±0.04	0.65	0.98±0.04	0.66	1.06±0.05	0.49	1.00±0.04	0.64	**0.73±0.04**	0.79
letter	0.97±0.01	0.50	0.96±0.01	0.51	0.75±0.01	0.82	0.97±0.01	0.50	0.74±0.01	0.83
optdigits	1.04±0.05	-0.02	1.06±0.06	0.02	0.91±0.06	-0.04	1.04±0.05	-0.11	0.91±0.06	-0.02
pendigits	0.88±0.06	0.40	0.87±0.06	0.34	0.55±0.06	0.69	0.88±0.05	0.40	0.60±0.07	0.74
satimage	1.05±0.02	0.41	1.00±0.02	0.42	0.87±0.02	0.59	1.06±0.02	0.41	0.86±0.02	0.62
segment	1.02±0.05	0.65	0.99±0.05	0.61	0.92±0.07	0.62	1.02±0.05	0.64	**0.71±0.05**	0.75
vehicle	0.90±0.12	0.42	0.88±0.15	0.57	0.94±0.11	0.77	0.91±0.11	0.41	**0.75±0.17**	0.75
tptp	1.23±0.38	0.18	1.01±0.05	0.26	1.02±0.08	0.35	1.20±0.33	0.11	**0.92±0.05**	0.44
sat	0.85±0.10	0.67	0.84±0.06	0.67	0.8±0.07	0.65	0.86±0.08	0.66	**0.64±0.07**	0.79

Table 6: NRMSE results and Spearman correlations those benchmarks in Table 1. Interpretation as in Table 2. Difficulty values obtained following the **AvE** approach.

Dataset	elasticNet		gbm		knn		lm		rf	
	NRMSE	Corr	NRMSE	Corr	NRMSE	Corr	NRMSE	Corr	NRMSE	Corr
diabetes	1.02±0.12	0.69	0.99±0.13	0.69	0.93±0.08	0.65	1.02±0.13	0.69	**0.89±0.10**	0.72
kc1	1.01±0.05	0.62	1.01±0.05	0.62	1.06±0.06	0.54	1.08±0.16	0.61	0.98±0.05	0.63
liver-disorders	1.05±0.15	0.58	0.99±0.12	0.63	1.01±0.09	0.72	1.06±0.14	0.58	**0.94±0.10**	0.82
japaneseVowels	1.01±0.08	0.64	0.98±0.08	0.66	1.07±0.07	0.49	1.01±0.08	0.64	**0.74±0.08**	0.80
letter	0.97±0.02	0.50	0.97±0.02	0.51	0.75±0.02	0.82	0.97±0.02	0.50	0.74±0.01	0.83
optdigits	1.04±0.05	0.25	1.07±0.05	0.42	0.89±0.05	0.40	1.05±0.05	0.22	0.90±0.05	0.53
pendigits	0.87±0.04	0.41	0.86±0.04	0.35	0.49±0.05	0.69	0.87±0.04	0.40	**0.54±0.05**	0.74
satimage	1.05±0.04	0.39	0.98±0.05	0.40	0.78±0.05	0.59	1.05±0.04	0.39	0.79±0.05	0.62
segment	1.03±0.07	0.63	1.00±0.07	0.60	0.91±0.07	0.61	1.03±0.07	0.62	**0.71±0.07**	0.75
vehicle	0.86±0.13	0.55	0.87±0.13	0.66	0.92±0.07	0.84	0.85±0.12	0.54	**0.71±0.14**	0.82
tptp	1.24±0.38	0.18	1.02±0.05	0.26	1.03±0.08	0.35	1.21±0.33	0.11	0.93±0.05	0.44
sat	0.89±0.11	0.67	0.86±0.08	0.67	0.83±0.08	0.66	0.96±0.24	0.67	**0.67±0.08**	0.80

Table 7: NRMSE results and Spearman correlations for those benchmarks in Table 1. Interpretation as in Table 2. Difficulty values obtained following a **2PL** IRT model. Training instances with a negative discrimination parameter have been removed from the training sets (**2PL (¬abs)**).

A common practice in IRT is to remove the items with low or negative discrimination, leaving only the items that are useful to evaluate respondents for exams and tests. In order to check whether this procedure may result in improved accuracy of the learned difficulty functions, we have removed all instances with negative discrimination from the validation set (but not in the test set) before training the different regression models. Table 7 shows that there is no improve-

Dataset	elasticNet		gbm		knn		lm		rf	
	NRMSE	Corr	NRMSE	Corr	NRMSE	Corr	NRMSE	Corr	NRMSE	Corr
diabetes	0.99±0.06	0.49	0.93±0.06	0.53	1.01±0.05	0.37	0.97±0.04	0.47	0.91±0.05	0.57
kc1	0.90±0.05	0.59	0.90±0.05	0.59	0.95±0.06	0.51	0.91±0.05	0.58	0.86±0.05	0.60
liver-disorders	1.00±0.04	0.54	1.01±0.05	0.59	1.10±0.07	0.66	1.01±0.04	0.54	1.02±0.05	0.79
japaneseVowels	0.90±0.04	0.62	0.88±0.04	0.63	0.96±0.05	0.46	0.90±0.04	0.61	0.63±0.04	0.76
letter	0.87±0.01	0.47	0.86±0.01	0.48	0.65±0.01	0.79	0.87±0.01	0.47	0.64±0.01	0.80
optdigits	0.94±0.05	-0.05	0.96±0.06	-0.01	0.81±0.06	-0.07	0.94±0.05	-0.14	0.81±0.06	-0.05
pendigits	0.78±0.06	0.37	0.77±0.06	0.31	0.45±0.06	0.66	0.78±0.05	0.37	0.50±0.07	0.71
satimage	0.95±0.02	0.38	0.90±0.02	0.39	0.77±0.02	0.56	0.96±0.02	0.38	0.76±0.02	0.59
segment	0.92±0.05	0.62	0.89±0.05	0.58	0.82±0.07	0.59	0.92±0.05	0.61	0.61±0.05	0.72
vehicle	0.80±0.12	0.39	0.78±0.15	0.54	0.84±0.11	0.74	0.81±0.11	0.38	0.65±0.17	0.72
tptp	1.13±0.38	0.15	0.91±0.05	0.23	0.92±0.08	0.32	1.10±0.33	0.08	0.82±0.05	0.41
sat	0.75±0.10	0.64	0.74±0.06	0.64	0.70±0.07	0.62	0.76±0.08	0.63	0.54±0.07	0.76

Table 8: NRMSE results and Spearman correlations for those benchmarks in Table 1. Interpretation as in Table 2. Difficulty values obtained following the **AvE** approach. Training instances with a negative discrimination parameter have been removed set (**AvE (¬abs)**).

ment (where, again, *rf* obtain the best results overall): the results are less robust compared to the setting using the 2PL approach that does not eliminate abstruse examples (Table 5) and the setting **1PL** approach (Table 2a). We have also performed the same procedure but using the **AvE** approach (removing the same examples as with the **2PL** approach from the validation set), also obtaining poorer results (see Table 8) than in the original version of the experiment (Table 6).

Applications

Explainable AI As we explained in the main text, the use of a *generic* difficulty metric $\hbar$ is very useful to understand where and how a system fails, and can be applied to any area in AI. Although it is important that the metric is *system-independent* (i.e., we analyse the problem instances, not a particular system), the use of a *attribute-based* $\hat{\hbar}$ increases the applications. First, there is no need to manually extract what makes instances hard, we can inspect what attributes make it hard, as we did with Fig. 3 (left). In that figure we see that onpix is very relevant, which is somewhat related to the density of the image, how much information or clutter it has.

While the range of applicability in Explainable AI is huge, in terms of understanding a problem and whether a system is conformant to this difficulty, the difficulty estimator can be applied to individual decisions or solutions. Many of the applications included in the main text are related to cases where we get a successful result for a hard instance or an unsuccessful result for an easy instance. This situation just triggers the analysis, but the difficulty estimator (and using XAI techniques on it, such as determining why a particular instance is difficult) can be very enlightening.

Robust evaluation and deployment For producing the SCC, we divide the instances in bins of the same length according to difficulty. For each bin, we plot on the x-axis the average difficulty of the instances in the bin and on the y-axis we plot the frequency of correct responses of the classifier (accuracy).

Figure 5 (left) shows the SCC obtained with the 70% of the letter benchmark using $\hbar$. We created 10 bins based on quantiles (but similar curves may be obtained varying this number or using same-length bins). Also, in Figure 5 (right) we use a test dataset (30% of unseen examples) to simulate a situation where we use the models in order to show whether the previous SCC can be used to select the (set of) best classifier(s) according to the difficulty ranges of the instances. Since we do not know the difficulty values of these unseen examples, we predict them using the best difficulty estimator $\hat{\hbar}$ obtained in the above experiments (see Table 2a). Then, we classify these instances (Table 9) by using the previous set of classifiers and plot the results in a new SCC according to the estimated difficulties. This way, we can decide which classifiers are preferable for a particular instance according to the estimated difficulties.

Table 9 shows the classifiers of interest and their accuracy for the letter benchmark (Frey and Slate 1991). Classifiers have been trained using their default hyperparameters.

ID	Classifier	Train	Test
rf	Random forest	1.00	0.98
c50	C5.0 decision tree	1.00	0.97
knn	k-nearest neighbours	0.98	0.96
jrip	Prop. rule learner	0.94	0.87
svm	Support vector machine	0.90	0.89
nb	Naïve Bayes	0.74	0.72
fda	Flexible discrim. analysis	0.68	0.68
nnet	Neural network	0.28	0.27
ada	Adaptive Boosting	0.29	0.28
rpart	Rec. part. and reg. tree	0.24	0.24

Table 9: Classifiers (using default parameters) and their accuracy for the letter benchmark (Frey and Slate 1991).

Analysing AI progress Figure 6 presents the SCCs for a subset of well-known CNN architectures designed to recognise visual patterns directly from pixel images (*AlexNet* (Krizhevsky, Sutskever, and Hinton 2012), *GoogLeNet* (Szegedy et al. 2015), *VGG* (Simonyan and Zisserman 2014), *ResNet* (He et al. 2016) *DenseNet* (Huang et al. 2017)) and *EfficientNet* (Tan and Le 2019). The architectures range from 2012 to 2019 and have been applied to the *CIFAR-10* benchmark (Krizhevsky, Hinton et al. 2009).

We have chosen CIFAR-10 because despite having an estimator with poor Spearman correlation, in this application we do not use the estimated difficulties but the original IRT ones, as an example of a use case of difficulty metrics for which we do not even need a (good) difficulty estimator.

Distributional and perturbational phenomena Table 10 shows the results for the study of problem shift: how does the difficulty estimator changes when we apply it to instances that come from a different problem (but share the features). We analyse this with 1000 random instances from MNIST compared to 1000 random instances from Fashion-MNIST. The estimator for MNIST, when applying to Fashion-MNIST gives higher difficulty (from -3.10 to -2.75). Note that -3.30 and -2.27 were the original average difficulties for MNIST and Fashion-MNIST respectively, but this is not necessarily calibrated for different problems. The comparison of the same estimator for two problems indicates that if these new 1000 instances were to be labelled with the original MNIST labels (independently of whether this makes sense), they would be more difficult than the original ones.

	S_{orig}	S_{shft}
$\hbar$ (mean)	-3.30	-2.27
$\hat{\hbar}$ (mean)	-3.10	-2.75

Table 10: Problem shift, between S_{orig} (1000 examples from MNIST) and S_{shft} (1000 examples from Fashion-MNIST). The first row shows the original difficulties using MNIST and Fashion-MNIST. The bottom row shows the mean difficulty using the MNIST difficulty estimator.

For the second batch of experiments where we study several kinds of perturbations, we implement the following procedures for the generation of the different samples:

- S_{advl}: we generate adversarial examples following the elastic-net regularized optimization (EAD) approach from (Chen et al. 2018). We focus on the L_2 distortion metric for the total variation for creating the adversarial examples. See Fig. 10 for a visual illustration of adversarial examples crafted by EAD.
- S_{hans}: we introduce simple watermarks (following a binary system) at the corners of each input image that help the classifier to predict the real class, emulating a Clever Hans phenomenon. Fig. 11 shows the watermarking system followed for each class.
- S^{low}_{blur}, S^{med}_{blur} and S^{high}_{blur}: we introduce increasing degrees of distortion to the input images using a Gaussian Blur filter, varying the kernel size (5, 9 and 13) and the variance in the intervals [0.1,2], [2,10] and [10,20] for low, mid and high distortion blur respectively. Fig. 12 shows the different levels of blur on the same image from MNIST.

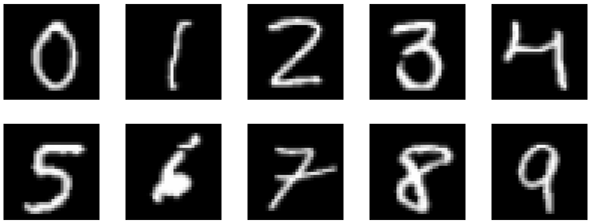

Figure 9: Selection of illustrative examples for each class in MNIST benchmark.

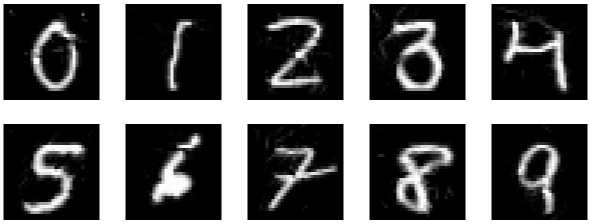

Figure 10: Visual illustration of adversarial examples crafted by EAD. Original examples in Fig. 9.

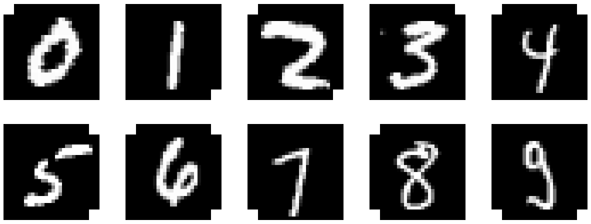

Figure 11: Visual illustration of images from MNIST with watermarks (binary symbols at the corners) for testing the Clever Hans phenomena.

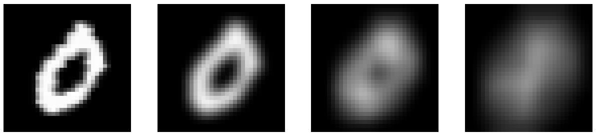

Figure 12: Example from MNIST with low, mid and high distortion (blur) levels applied. Original image on the left.

As we have seen in Table 4, the adversarial attack and Clever Hans phenomena have no effect on the difficulty estimator. However, performance is very different, going from almost total success for in S_{orig} to 0% in S_{advl}, or the the classifier succeeding for all of them in S_{hans}. It is only when we apply different levels of blur that we have an effect on the estimated difficulties and the error. More blur makes images more difficult (from -3.05 to -2.45), as expected, and the classifier has higher error (from 0.80% to 76.5%). In general, if the difficulties change in the new samples as does the performance, we can calculate whether the performance corresponds to the ability of the model. In order to do this, we derive the ability by inverting the logistic model given the average difficulty and performance (correct response percentage) in S_{orig} and then we apply the logistic model forward given the ability and the average difficulty in S_{advl} to get the expected performance (correct response percentage). If this deviates from the observed performance significantly, then we fire the alarm.

IRT in AI

One of the first approaches in which IRT is applied to machine learning and artificial intelligence can be found in (Martínez-Plumed et al. 2016; Martınez-Plumed and Hernández-Orallo 2016; Martínez-Plumed et al. 2019). In this paper, authors explain IRT and how can it be applied to evaluate machine learning algorithms performance. Authors evaluate different families of machine learning models (respondants) on different classification problems (items) showing how IRT works in that context and what their parameters mean when it is applied to machine learning evaluation.

The application of IRT to evaluate ML algorithms is mainly used in classification problems. However, some advances have been applied in IRT computation to allow applying this approximation not limited to logistic curves

(Chen et al. 2019). In this way, in (Moraes et al. 2020), authors apply IRT directly for regression. In this case, a new parameterisation is proposed based on the normalised errors produced by the respondents.

We can also find researches that use IRT to predict the minimum performance of AI agents (Chmait et al. 2017). In this approximation, authors propose the use of the tasks' difficulty measure and the agents' ability to guarantee the minimum accuracy that these agents can achieve in each task. But not only IRT has been applied recently in predicting performance of AI algorithms, it is also applied in explainable machine learning. In this subject, Kline et al (Kline et al. 2020) successfully apply IRT and the discriminant capacity of features to explain the feature importance in ML models to predict the mortality in intensive care units.

In (Lalor 2020), author demonstrates that exists certain correlation of the IRT latent parameters between human and DNNs (Lalor et al. 2018). However, the most interesting results are related to curriculum learning of DNN based on IRT measures. In this thesis, the author demonstrates that progressive learning accelerates and increases the efficiency of DNN learning process by increasing progressively the difficulty of the examples depending on the acquired abilities of the DNN being trained. Both latent parameters, difficulty and ability, are extracted from IRT approximation. This approach works in a similar way to Competency introduced by Platanios et al (Platanios et al. 2019). Nevertheless, unlike Competency approximation, in (Lalor 2020) the skill of the DNN model is dynamically computed using IRT as the model learns. In this way, the examples provided during learning are adapted to the abilities of the model, improving the speed and efficiency of learning. However, the difficulties of the examples are computed a priori using human responses or ”artificial crowds” (a set of DNNs with different skill levels).

One of the greatest handicaps that must face IRT is its scalability. Therefore, high dimensional and large datasets reduces the speed and accuracy of algorithms that tries to fit IRT models. In case of high dimensional dataset (high number of items), the problem can be solved by batching the IRT computation. However, in case of large datasets, the solution is no so easy. In (Wu et al. 2020), authors proposes the use of variational Bayesian inference to allow obtaining IRT parameters for large datasets (high number of respondants). In this case, variational inference demonstrates its power to perform fast inference for complex Bayesian models like IRT. Authors demonstrate its applicability in real scenarios and large datasets like PISA, DuoLingo and Gradescope.

Finally, it is worth to mention a couple of papers that propose an approximation similar to the one deployed in this paper but with a completely different scope. In this case, they do not aim to evaluate Machine Learning algorithms but to predict students performance in a test. In the first paper (Yeung 2019), authors propose the use of a deep learning approximation based on recursive neural network (RNN) to predict the probability that a student give a correct answer based on past interactions. The RNN is used to predict the ability of the student and the difficulty of the item. Then, they use these predicted values to calculate the probability that the student gives a correct answer using IRT. In the second paper (Cheng et al. 2019), authors use a proficiency vector (representing the degree a student masters each concept knowledge) of one student, the text of the exercise and the concepts included in this text as input to their approximation. With that information, they apply two DNNs to predict the ability of the student and the discriminatory capacity of the item (e.g. a math exercise) and an LSTM to calculate the difficulty of the item. The ability, discrimination and difficulty obtained are used in the IRT formula to predict the probability that the mentioned student gives a correct answer to that item.

Finally, With the purpose of better analysing the result of AI benchmarks, (Martínez-Plumed and Hernández-Orallo 2020) extends IRT to one further indicator on the side of the AI systems: generality. While difficulty, discrimination and ability latent parameters are adapted from psychometric models in IRT, generality is defined as a new metric that evaluates whether an agent is consistently good at easy problems and bad at difficult ones. Generality is thus useful to determine whether the new AI techniques, especially those that rely on long training stages, are coping well a wide range of problems (with different difficulties), and not only for a pocket of problems, but failing in some situations.

References

Attali, Y.; Saldivia, L.; Jackson, C.; Schuppan, F.; and Wanamaker, W. 2014. Estimating item difficulty with comparative judgments. *ETS Research Report Series*, 2014(2): 1–8.

Birnbaum, A. 1968. *Statistical Theories of Mental Test Scores*, chapter Some Latent Trait Models and Their Use in Inferring an Examinee's Ability. Reading, MA.: Addison-Wesley.

Breiman, L. 2001. Random forests. *Machine learning*, 45(1): 5–32.

Chalmers, R. P. 2012. mirt: A multidimensional item response theory package for the R environment. *Journal of statistical Software*, 48(1): 1–29.

Chen, P.-Y.; Sharma, Y.; Zhang, H.; Yi, J.; and Hsieh, C.-J. 2018. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Thirty-second AAAI conference on artificial intelligence*.

Chen, T.; Kornblith, S.; Swersky, K.; Norouzi, M.; and Hinton, G. 2020. Big Self-Supervised Models are Strong Semi-Supervised Learners. *arXiv preprint arXiv:2006.10029*.

Chen, Y.; Filho, T. S.; Prudencio, R. B.; Diethe, T.; and Flach, P. 2019. β^3-IRT: A New Item Response Model and its Applications. In Chaudhuri, K.; and Sugiyama, M., eds., *Proc. 22nd International Conference on Artificial Intelligence and Statistics*, volume 89, 1013–1021. PMLR.

Chen, Z.; and Ahn, H. 2020. Item response theory based ensemble in machine learning. *International Journal of Automation and Computing*, 17(IJAC-2020-02-029): 621.

Cheng, S.; Liu, Q.; Chen, E.; Huang, Z.; Huang, Z.; Chen, Y.; Ma, H.; and Hu, G. 2019. DIRT: Deep Learning Enhanced Item Response Theory for Cognitive Diagnosis. In Zhu, W.; Tao, D.; Cheng, X.; Cui, P.; Rundensteiner, E. A.;

Carmel, D.; He, Q.; and Yu, J. X., eds., *Conference on Information and Knowledge Management, CIKM*, 2397–2400. ACM.

Chmait, N.; Dowe, D.; Li, Y.-F.; and Green, D. 2017. An information-theoretic predictive model for the accuracy of AI agents adapted from psychometrics. In Everitt, T.; Potapov, A.; and Goertzel, B., eds., *Artificial General Intelligence*, 225–236. Springer.

Corbière, C.; Thome, N.; Bar-Hen, A.; Cord, M.; and Pérez, P. 2019. Addressing failure prediction by learning model confidence. *arXiv preprint arXiv:1910.04851*.

Cuzick, J. 1985. A Wilcoxon-type test for trend. *Statistics in medicine*, 4(1): 87–90.

De Ayala, R. J. 2013. *The theory and practice of item response theory*. Guilford Publications.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Derksen, S.; and Keselman, H. J. 1992. Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology*, 45(2): 265–282.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.

Elo, A. E. 1978. *The rating of chessplayers, past and present*. Arco Pub.

Embretson, S. E.; and Reise, S. P. 2000. *Item response theory for psychologists*. L. Erlbaum.

Fernández-Delgado, M.; Cernadas, E.; Barro, S.; and Amorim, D. 2014. Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 15(1): 3133–3181.

Fix, E.; and Hodges, J. L. 1989. Discriminatory analysis. Nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3): 238–247.

Frey, P. W.; and Slate, D. J. 1991. Letter recognition using Holland-style adaptive classifiers. *Machine learning*, 6(2): 161–182.

Friedman, J. H. 2002. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4): 367–378.

Gupta, K.; Pesquet-Popescu, B.; Kaakai, F.; Pesquet, J.-C.; and Malliaros, F. D. 2021. An Adversarial Attacker for Neural Networks in Regression Problems. In *IJCAI Workshop on Artificial Intelligence Safety (AI Safety)*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, X.; Zhao, K.; and Chu, X. 2021. AutoML: A Survey of the State-of-the-Art. *Knowledge-Based Systems*, 212: 106622.

Hendrycks, D.; and Gimpel, K. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Hoover, D. L. 2003. Another perspective on vocabulary richness. *Computers and the Humanities*, 37(2): 151–178.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2018. Densely Connected Convolutional Networks. arXiv:1608.06993.

Ionescu, R. T.; Alexe, B.; Leordeanu, M.; Popescu, M.; Papadopoulos, D. P.; and Ferrari, V. 2016. How hard can it be? Estimating the difficulty of visual search in an image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2157–2166.

Jiang, H.; Kim, B.; Guan, M. Y.; and Gupta, M. 2018. To trust or not to trust a classifier. *arXiv preprint arXiv:1805.11783*.

Kincaid, J. P.; Fishburne Jr, R. P.; Rogers, R. L.; and Chissom, B. S. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kline, A.; Kline, T.; Shakeri Hossein Abad, Z.; and Lee, J. 2020. Using Item Response Theory for Explainable Machine Learning in Predicting Mortality in the Intensive Care Unit: Case-Based Approach. *J Med Internet Res*, 22(9): e20268.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105.

Kuhn, M. 2008. Building predictive models in R using the caret package. *Journal of statistical software*, 28(1): 1–26.

Lalor, J. P. 2020. *Learning Latent Characteristics of Data and Models using Item Response Theory*. Ph.D. thesis, Doctoral Dissertations, 1842.

Lalor, J. P.; Wu, H.; Munkhdalai, T.; and Yu, H. 2018. Understanding Deep Learning Performance through an Examination of Test Set Difficulty: A Psychometric Case Study. In *Empirical Methods in Natural Language Processing*, 4711–4716. Association for Computational Linguistics.

Liu, D.; Xiong, Y.; Pulli, K.; and Shapiro, L. 2011. Estimating image segmentation difficulty. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, 484–495. Springer.

Loshchilov, I.; and Hutter, F. 2017. SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv:1608.03983.

Magno, C. 2009. Demonstrating the difference between classical test theory and item response theory using derived test data. *The international Journal of Educational and Psychological assessment*, 1(1): 1–11.

Martınez-Plumed, F.; and Hernández-Orallo, J. 2016. AI results for the Atari 2600 games: difficulty and discrimination using IRT. *EGPAI, Evaluating General-Purpose Artificial Intelligence*, 33.

Martínez-Plumed, F.; and Hernández-Orallo, J. 2020. Dual Indicators to Analyse AI Benchmarks: Difficulty, Discrimination, Ability and Generality. *IEEE Transactions on Games*, 12(2): 121–131.

Martínez-Plumed, F.; Prudêncio, R. B.; Martínez-Usó, A.; and Hernández-Orallo, J. 2019. Item response theory in AI: Analysing machine learning classifiers at the instance level. *Artificial Intelligence*, 271: 18–42.

Martínez-Plumed, F.; Prudêncio, R. B. C.; Martínez-Usó, A.; and Hernández-Orallo, J. 2016. Making Sense of Item Response Theory in Machine Learning. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, Best Paper Award*, 1140–1148.

Martínez-Plumed, F.; Castellano-Falcón, D.; Monserrat, C.; and Hernández-Orallo, J. 2022. When AI Difficulty is Easy: The Explanatory Power of Predicting IRT Difficulty. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Maydeu-Olivares, A. 2013. Goodness-of-fit assessment of item response theory models. *Measurement: Interdisciplinary Research and Perspectives*, 11(3): 71–101.

Mishra, S.; and Arunkumar, A. 2021. How Robust are Model Rankings: A Leaderboard Customization Approach for Equitable Evaluation. *arXiv preprint arXiv:2106.05532*.

Moraes, J. V. C.; Reinaldo, J. T. S.; Prudencio, R. B. C.; and Silva Filho, T. M. 2020. Item Response Theory for Evaluating Regression Algorithms. In *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Platanios, E. A.; Stretcu, O.; Neubig, G.; Poczos, B.; and Mitchell, T. M. 2019. Competence-based Curriculum Learning for Neural Machine Translation. arXiv:1903.09848.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Richards, B. 1987. Type/token ratios: What do they really tell us? *Journal of child language*, 14(2): 201–209.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.

Sinha, K.; Jia, R.; Hupkes, D.; Pineau, J.; Williams, A.; and Kiela, D. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. *arXiv preprint arXiv:2104.06644*.

Smith, M. R.; Martinez, T.; and Giraud-Carrier, C. 2014. An Instance Level Analysis of Data Complexity. *Mach. Learn.*, 95(2): 225–256.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *IEEE conference on computer vision and pattern recognition*, 1–9.

Tan, M.; and Le, Q. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 6105–6114. PMLR.

Vanschoren, J.; Van Rijn, J. N.; Bischl, B.; and Torgo, L. 2014. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2): 49–60.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Vijayanarasimhan, S.; and Grauman, K. 2009. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *2009 IEEE conference on computer vision and pattern recognition*, 2262–2269. IEEE.

Wright, B. D.; and Stone, M. H. 1979. Best test design.

Wu, M.; Davis, R. L.; Domingue, B. W.; Piech, C.; and Goodman, N. 2020. Variational Item Response Theory: Fast, Accurate, and Expressive. arXiv:2002.00276.

Yeung, C.-K. 2019. Deep-IRT: Make Deep Learning Based Knowledge Tracing Explainable Using Item Response Theory. arXiv:1904.11738.

Zou, H.; and Hastie, T. 2005. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2): 301–320.