# XAI with Machine Teaching when Humans Are (Not) Informed about the Irrelevant Features*

Brigt Arve Toppe Håvardstun✉[1], Cèsar Ferri[2], Jose Hernández-Orallo[2], Pekka Parviainen[1], and Jan Arne Telle[1]

[1] Department of Informatics, University of Bergen, Norway.
[2] VRAIN, Universitat Politècnica de València, Spain.
Brigt.Havardstun@uib.no, cferri@dsic.upv.es, jorallo@upv.es, Pekka.Parviainen@uib.no,Jan.Arne.Telle@uib.no

**Abstract.** Exemplar-based explainable artificial intelligence (XAI) aims at creating human understanding about the behaviour of an AI system, usually a machine learning model, through examples. The advantage of this approach is that the human creates their own explanation in their own internal language. However, what examples should be chosen? Existing frameworks fall short in capturing all the elements that contribute to this process. In this paper, we propose a comprehensive XAI framework based on machine teaching. The traditional trade-off between the fidelity and the complexity of the explanation is transformed here into a trade-off between the complexity of the examples and the fidelity the human achieves about the behaviour of the ML system to be explained. We analyse a concept class of Boolean functions that is learned by a convolutional neural network classifier over a dataset of images of possibly rotated and resized letters. We assume the human learner has a strong prior (Karnaugh maps over Boolean functions). Our explanation procedure then behaves like a machine teaching session optimising the trade-off between examples and fidelity. We include an experimental evaluation and several human studies where we analyse the capacity of teaching humans these Boolean function by means of the explanatory examples generated by our framework. We explore the effect of telling the essential features to the human and the priors, and see that the identification is more successful than by randomly sampling the examples.

## 1 Introduction

In the field of eXplainable AI (XAI), there are multiple ways to explain humans how an AI system works, one of them being example-based XAI [16, 21, 24], where the XAI system aims to find examples showing how the machine learning system acts in different situations. Machine teaching is the research area of actively selecting an optimal (e.g., minimal) set of examples so that a learner can identify a given concept or model [27]. The goal is for the teacher to find

* A preliminary version of this work was presented as a poster at AAIP@IJCLR2022. Supported by the Norwegian Research Council, project Machine Teaching for XAI.

the smallest training set —known as the *teaching* or *witness* set— such that, a learning algorithm, when given the teaching set as an input, produces a target concept. In this work, we propose a framework based on machine teaching techniques where the XAI system (the teacher) provides explanatory examples to humans (the learners). The target concept is (a part of) the black-box AI system that needs explanation. The machine teaching algorithm must find a small set of labelled examples that will allow the human to build their own model of the AI system and thereby arrive at an explanation of the target concept [19, 17]. We demonstrate the validity of our proposal by including some results of an experimental evaluation where we evaluate the results of teaching a black-box model to humans. Specifically, the black box to explain is an artificial neural network learned from images generated by Boolean expressions. We choose Boolean functions because the notions of prototype, centroid, anchors or boundary examples are more elusive in discrete concept classes like this, but we also use neural networks that might use some other features. We analyse the effect of giving humans information about how the examples were chosen and indications about the relevant features. The results show that our framework can generate explanatory examples useful to teach humans Boolean functions, better than sampling examples at random.

The paper is structured as follows. In Section 2 we review part of the literature related to XAI and machine teaching. Section 3 describes the framework we developed to generate explanatory witness sets. We instantiate that method for explaining neural network classifiers of images representing Boolean concepts in Section 4. Section 5 describes the experiments and human studies, and discusses the results. Finally, Section 6 closes the paper with conclusions and future work.

## 2 Machine Teaching for XAI

Explainable AI (XAI) is an active research field aiming at explaining the decisions of AI systems [16]. Machine learning is a key component of many AI systems, and therefore XAI usually focuses on explaining machine learning models [7, 22].

Explainable AI must usually face several trade-offs, such as the tension between fidelity (level of coincidence between the predicted or understood behaviour of the system and the actual behaviour of the model) and comprehensibility (how much effort it takes for the human to understand) [5]. In general, making useful explanations among these tensions requires a great deal of abstraction, additionally modelling machine behaviour [20] in a way that is comprehensible to humans.

XAI approaches are divided into two families. In the first one, the goal is to extract an abstract representation of the AI system to serve as an explanation to a human. An example of this approach is extracting comprehensible rules from models [3]. In the second family, the goal is to use examples such that humans can infer their explanation themselves, known as exemplar-based explanations. An example of this approach is using anchors or partial examples [21].

*Machine teaching* [26] is a research field that is sometimes considered as an inverse problem to machine learning. In machine teaching the examples are chosen wisely by a teacher to teach a concept to the learner. Figure 1 shows a situation where the teacher has the concept of reversing a list. The teacher could try to explain the concept, but the languages employed by the learner and teacher might not be the same. In this situation, as happens with humans frequently, a few examples may be more effective. In the image, the teacher sends a couple of input-output pairs to the learner, thinking that this would be useful for the learner to build and identify the concept.

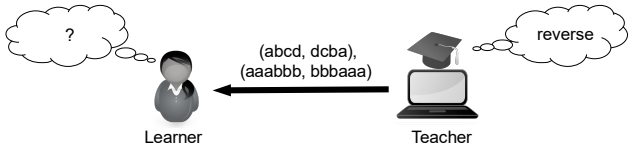


**Fig. 1.** Machine teaching example. The teacher tries to teach the concept of the *reverse* of a string. The teacher selects two examples carefully and shows them to the learner: the input string *abcd* being mapped into *dcba*, and the input string *aaabbb* being mapped into *bbbaaa*. The learner must infer the concept from only these two examples.

Mainly, machine teaching has been used to comprehend and depict how humans teach. An example is the analysis conducted by [12], which examines the teaching of 1D concepts (intervals) to machines, comparing a machine teaching environment with a curriculum learning environment. In both instances, the question is whether humans provide examples at the boundaries to assist the learner in replicating these boundaries or if they provide examples in clear areas so that the user can interpolate, as outlined by [1].

Our focus lies in machine teaching for the purpose of explaining concepts to humans [8]. In certain models, the teacher can interact with the learner by posing questions (e.g., [15]). On the other hand, some methods have attempted to expand the machine teaching framework by using examples to achieve explainable AI. A few proposals stray from the traditional machine teaching approach and instead utilize well-selected demonstrations in inverse reinforcement learning [9], or in the Cooperative Inverse Reinforcement Learning (CIRL) framework [6].

Yang et al. [25] evaluated the effectiveness of example-based explanations for AI using Bayesian Teaching, with a focus on high sensitivity and high specificity, and we will compare our findings to theirs. Another approach to teaching for XAI is the decomposition of the learner's hypothesis into an attention function and a decision function, as proposed by Chen et al. [2]. Ouyang [18] presents an algorithm for the Bayesian inference of regular expressions using examples. The teaching paradigm proposed is also linked to how humans communicate and how the speaker chooses the appropriate word based on their listener.

## 3 A MT framework to generate explanatory teaching sets

In machine teaching, the teacher $T$ is viewed as a function from concepts to sets of labelled examples, with $T(\theta) = S$ denoting the labelled examples $S$ the teacher employs to teach concept $\theta$. Likewise, the learner $L$ is viewed as a function from sets of labelled examples to concepts, and we require that the concept guessed by the learner is compatible with the given examples $S$, denoted $L(S) \models S$. Correct teaching is achieved if $L(T(\theta)) = \theta$, i.e. the guessed concept is indeed the one the teacher had in mind. To achieve an efficient teaching protocol we employ simplicity $\beta$ on concepts and $\delta$ on example sets (Occam's razor), as in [23]. $\beta$ is shared by learner and teacher, and $\delta$ is used to prioritise simple witness set. When applying this to XAI the concept $\theta_{AI}$ can be the entire AI model to be explained or some particular substructure. To build our XAI system we employ i) an machine learning algorithm $L_M$ modelling the human learner $L_H$ with its simplicity prior $\beta$ on guessed concepts, ii) a simplicity prior $\delta$ on example sets, and iii) a loss function $\lambda$ giving a penalty for deviations of the guess $\theta_M$ from the intended $\theta_{AI}$.

We propose a parameterised framework to generate explanatory examples from a black-box model $\theta_{AI}$. In the framework, we explore the trade-off between fidelity (squared error of the guessed model compared to the black-box model) and teaching complexity (measured as the complexity of the set of labelled examples used as a teaching set) [14, 24]. The framework is defined as:

$$T(\theta_{AI}) = \underset{S:\theta_{AI} \models S}{\operatorname{argmin}} \{\delta(S) + \mu \cdot \lambda(\theta_{AI}, \theta_M) : L_M(S) = \theta_M\} \tag{1}$$

$$L_M(S) = \underset{\theta_M : \theta_M \models S}{\operatorname{argmin}} \{\beta(\theta_M)\}$$

In these equations $T$ is a teacher, aiming to teach a concept $\theta_{AI}$ to a human learner $L_H$, by finding a teaching set $S$ such that $L_H(S) = \theta_{AI}$. To achieve automation and increase iteration speed a model $L_M$ of $L_H$ is used, and the teacher will therefore aim for $T(\theta_{AI}) = S$ **s.t.** $L_M(S) = \theta_{AI}$. The fidelity function becomes $1 - \lambda$ and it measures how closely the guessed concept $\theta_M$ matches the concept $\theta_{AI}$, while the factor $\mu$ allows us to balance the influence of complexity ($\delta$) and fidelity ($1 - \lambda$). In this work, we present an implementation[3] of Equation 1 tested on a machine learning model trained on images generated by basic Boolean functions.

## 4 Obtaining explanatory examples from a neural network

In this section we discuss how the framework presented in the previous section is applied to a black-box model represented by a neural network learned from images generated by basic Boolean functions.

[3] `https://github.com/BrigtHaavardstun/ExplainableAI`

### 4.1 The black-box model $\theta_{AI}$

For the experimental setting, we implemented our own $\theta_{AI}$, with the task of learning a Boolean function on four variables, $\phi(A, B, C, D)$. Determining the subjective difficulty of learning Boolean functions has been addressed in the literature, see e.g. [4]. The input to $\theta_{AI}$ will be a bitmap containing a subset of letters from the alphabet $\Sigma = \{A, B, C, D\}$, with the letters present being the variables set to True. The bitmaps thus represent an example, with letters being rotated and scaled and placed randomly. This gives us the possibility of extensive training data for our AI. The output space of $\theta_{AI}$ is {0, 1}. For instance, with the concept $\phi = (A \wedge B) \vee (C \wedge D)$, we label an example 1 if $\phi$ evaluates to True, and 0 if $\phi$ evaluates to False.

We chose a Convolutional Neural Network (CNN)[13], as a common technique for images, while at the same time not interpretable by themselves, making them a good choice for generating our $\theta_{AI}$. We implemented a CNN with 8 layers in Python using Keras and TensorFlow.

### 4.2 The model of the human $L_M$

For simplicity, our model $L_M$ of the human learner will not be given bitmaps as examples. Instead, it takes as input the letters present in each image. We thus hypothesise that the human will pay attention to the letters present in the image and disregard other information such as rotation, size and position.

The hypothesis class of $L_M$ will consist of all Boolean functions over the 4-letter alphabet. Then, given a teaching set like $S = \{(AC, 0), (AD, 0), (BD, 0), (AB, 1), (BC, 1), (CD, 1)\}$, we must decide how $L_M$ will act. We assume a human constructs something like a partial truth table, in this case with 3 rows out of $2^4 = 16$ rows total filled with True, 3 rows filled with False, and 10 rows filled with Don't-Cares (x). Applying Occam's razor, we need to define the function $\beta$, to choose the Boolean function that is most simple and adheres to these constraints. A commonly accepted answer is the use of Karnaugh maps[11].

We use disjunctive normal form (DNF) which mimics human reasoning. To verify a positive instance you need only to confirm one clause, whereas to confirm a negative instance you always need to check all clauses. The resource-heavy task of confirming a negative compared to a positive is somewhat similar to how humans are poor at negations [10]. For each teaching set the Karnaugh map technique can find many possible DNFs, and in the spirit of K-map minimization we use the following scheme to pick the simplest. The DNFs are sorted in order by fewest clauses, and to break ties we compare clauses starting from the simplest one, using the criteria 1) fewest variables, 2) fewest negations, 3) lexicographic order. This defines $\beta$ and gives us a unique Boolean formula in DNF form for each teaching set.

### 4.3 The fidelity function $1 - \lambda$

When we want to compare $\theta_{AI}$ and $\theta_M$, we need to view the former as an approximation to some Boolean function, but also being affected by the location,

rotation, etc., of the letter. Consequently, for each subset of letters (logical example), we estimate the percentage of images containing exactly these letters that $\theta_{AI}$ evaluates to True on new images, based on the full training set. We get values like the top row in Table 1. We observe that $\theta_{AI}$ predicts some letter groups the same and is more undecided on other letter combinations.

**Table 1.** Top row shows the percentage of bitmaps on letters for that column for which $\theta_{AI}$ evaluates to True. Bottom row shows the truth table of $\theta_M = (A \wedge B) \vee (C \wedge \neg A)$, and $\lambda(\theta_{AI}, \theta_M) = \frac{0.2222}{16} \approx 0.0139$ is the MSE of the difference of all 16 columns, giving fidelity 0.9861.

| Symbol | $\emptyset$ | A | B | C | D | AB | AC | AD | BC | BD | CD | ABC | ABD | ACD | BCD | ABCD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\theta_{AI}$ predicts | 0.00 | 0.00 | 0.00 | 0.95 | 0.00 | 0.99 | 0.02 | 0.00 | 0.63 | 0.02 | 0.91 | 1.00 | 1.00 | 0.04 | 0.74 | 1.00 |
| $\theta_M$ evaluates | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |

To evaluate how well the $\theta_M$, returned by the learner as the Boolean formula minimizing $\beta$, matches $\theta_{AI}$, we use its truth table as in the bottom row of Table 1. We then compare the two rows (for $\theta_{AI}$ and $\theta_M$) using Mean Square Error (MSE) to get $\lambda$ and fidelity $1 - \lambda$.

We have experimented with various definitions for the complexity function, to punish large and complicated teaching sets $S$. The chosen $\delta$ is a simple squared sum of the number of variables present in each example, plus 0.1 for the empty set (corresponding to setting no variable to True). We thus keep low the total number of variables in all examples while simultaneously putting a high cost on a single large example. Note that the $\delta$ values are typically much higher than the $\lambda$ values, so in our first set of experiments we set the multiplicative factor $\mu = 800$ when computing the aggregated score $\delta(S) + \mu \cdot \lambda(\theta_{AI}, \theta_M)$.

### 4.4 The teacher $T$

The goal of the teacher is to find a teaching set explaining $\theta_{AI}$, by iterating over potential teaching sets. For each teaching set $S$, we compute $L_M(S) = \theta_M$ as described earlier, and the aggregate score $\delta(S) + \mu \cdot \lambda(\theta_{AI}, \theta_M)$. During the iteration we retain the best aggregate score. For these experiments the iteration is an exhaustive search.

## 5 Experimental evaluation

Given the previous setting we performed a set of experiments using different concepts and parameters to analyse the effect of several elements in the machine teaching process on explaining the behaviour of various AI models. In particular, we played with AI models trained on different sized training sets, which approximate the original Boolean function to different levels of accuracy. Depending on how well the AI is approximated by a Boolean function the trade-off parameter $\mu$ between fidelity $(1 - \lambda)$ and teaching complexity $(\delta)$ has different effects.

### 5.1 Generation of teaching sets

**5.1.1 Fixed $\mu$ for varying models** We trained nine different $\Theta_{AI}$ models with differently sized data sets. In this first experiment, all models are trained with the ground truth $\phi = (A \wedge B) \vee C$ and the alphabet $\Sigma = \{A, B, C\}$. The data set sizes used in the experiment are: {10, 50, 100, 500, 1000, 2000, 5000, 10000, 50000}. Accordingly, we denote the different models: {$AI_{10}$, $AI_{50}$, $AI_{100}$, $AI_{500}$, $AI_{1000}$, $AI_{2000}$, $AI_{5000}$, $AI_{10000}$, $AI_{50000}$}.

In Table 2 we show several results. In the first row we see the expected result that the accuracy of the models wrt the original concept $\phi$ increases as more training examples were given to the neural network. In the next rows we show the Boolean expression that best approximates the model, with its associated highest possible fidelity $(1 - \lambda)$ over all Boolean functions. We see that the language of Boolean functions obtains a perfect match for the case of $AI_{10}$ (because the underlying concept is very simple, always predicting True, which is a Boolean function) and almost perfect for $AI_{10000}$ and $AI_{50000}$ (because the number of training examples leads to a concept that is very close to $\phi$). Note that also in other cases the most accurate Boolean function is $\phi$ (from $AI_{2000}$ and up).

**Table 2.** Several $\Theta_{AI}$ models $AI_t$ trained for size $t$ of training examples for $\phi = AB + C = (A \wedge B) \vee C$. We first show accuracy with respect to $\phi$. The next two rows show the closest Boolean expression ($AB+C$ from $AI_{2000}$ and on) and its fidelity value $1 - \lambda$. Then we do teaching with $\mu = 800$, and show the Boolean concept taught by the system, the teaching set and its complexity, the fidelity and aggregate score.

| AIs | $AI_{10}$ | $AI_{50}$ | $AI_{100}$ | $AI_{500}$ | $AI_{1000}$ | $AI_{2000}$ | $AI_{5000}$ | $AI_{10000}$ | $AI_{50000}$ |
|---|---|---|---|---|---|---|---|---|---|
| Accuracy AB+C | 62.50 | 72.85 | 78.38 | 81.01 | 88.47 | 91.42 | 94.74 | 98.72 | 99.36 |
| Boolean with highest $1 - \lambda$ | Always True | A+B+C | A+B+C | AB+ AC+BC | AB+ AC+BC | AB+C | AB+C | AB+C | AB+C |
| Highest $1 - \lambda$ | 1 | 0.927 | 0.9578 | 0.9226 | 0.9843 | 0.9752 | 0.9936 | 0.9994 | 0.9999 |
| Model taught $\theta_M$ | Always True | A+B+C | A+B+C | A+C | AB+ AC+BC | AB+C | AB+C | AB+C | AB+C |
| Teaching Set $S$ | {(∅,1)} | {(∅,0), (A,1), (B,1), (C,1)} | {(∅,0), (A,1), (B,1), (C,1)} | {(∅,0), (A,1), (C,1)} | {(A,0), (AB,1), (AC,1), (B,0), (BC,1), (C,0)} | {(A,0), (AB,1), (B,0), (C,1)} | {(A,0), (AB,1), (B,0), (C,1)} | {(A,0), (AB,1), (B,0), (C,1)} | {(A,0), (AB,1), (B,0), (C,1)} |
| $\delta(S)$ | 0.1 | 3.1 | 3.1 | 2.1 | 15 | 7 | 7 | 7 | 7 |
| $1 - \lambda(AI_x, \theta_M)$ | 1 | 0.927 | 0.9578 | 0.9179 | 0.9843 | 0.9752 | 0.9936 | 0.9994 | 0.9999 |
| $\delta + 800\lambda$ | 0.1 | 61.52 | 36.71 | 67.82 | 27.63 | 26.84 | 12.17 | 7.49 | 7.09 |

Now let us look at the next few rows showing results for the teaching framework when run with the chosen parameter $\mu = 800$. First we show the Boolean concept $\theta_M$ that is actually taught by the system and note that it is almost always equal to the Boolean concept with highest fidelity $(1 - \lambda)$ value in the 2nd row. The only exception is $AI_{500}$ where the trade-off between $\delta$ and $\lambda$ favours the Boolean concept $A \vee C$ instead of $(A \wedge B) \vee (A \wedge C) \vee (B \wedge C)$ because the teaching set for the former is much simpler ($\delta = 2.1$) than the teaching set for the latter ($\delta = 15$ as can be seen under $AI_{1000}$). The next rows show the teaching set employed, its $\delta$ value, the fidelity value and the aggregate score.

There are three clear cases in the table ($AI_{10}$, $AI_{10000}$ and $AI_{50000}$) where a simple teaching set allows the teacher to convey a concept to the learner that very closely captures the model. But there are other cases, such as $AI_{1000}$ and $AI_{2000}$, where the situation is less clear. For $AI_{1000}$ the fidelity is not bad ($1 - \lambda = 1 - 0.0157 = 0.9843$) but the complexity of teaching becomes high ($\delta = 15$) so even if a sufficiently accurate concept can be taught this is at the cost of a higher effort from the learner. For $AI_{2000}$ we see that this cost is reduced but the fidelity is worse ($1 - \lambda = 1 - 0.0248 = 0.9752$).

**5.1.2 Varying $\mu$ for a single model** In a second experiment we trained a $\Theta_{AI}$ model on a data set of size 350 for $\phi = (A \wedge B) \vee (C \wedge D) = AB + CD$ on 4 variables/letters. The accuracy was 78.25% and the closest Boolean function, with a fidelity value $1 - \lambda$ of $1 - 0.06 = 0.94$, turned out to be $ABC + ABD + ACD + BCD$, which can be interpreted as "True if and only if at least 3 letters present". To investigate the trade-off between fidelity and complexity, teaching was done with varying values of $\mu$, see left column in Table 3. We see that as $\mu$ increases more emphasis is put on fidelity at expense of complexity. Note that at $\mu = 3200$ the fidelity is as good as possible (i.e. highest possible $1 - \lambda$) since the teaching set at $\mu = 3200$ is optimal for that optimal $\theta_M$ so increasing $\mu$ will have no effect. Of course, this comes at the expense of a high complexity. An option worth exploring is to take the characteristics of the human user into account when deciding on the fidelity vs complexity trade-off, e.g., having a high value of $\mu$ for an expert and a low value for a non-expert.

**Table 3.** Results for a single AI model where the closest Boolean function turned out to be $ABC+ABD+ACD+BCD$. Teaching was done with varying values of $\mu$, see left column. As $\mu$ increases more emphasis is put on lower fidelity $1 - \lambda = 1 - \lambda(\theta_{AI}, \theta_M)$ at expense of higher teaching complexity $\delta$.

| Range $\mu$ | Model taught $\theta_M$ | $1 - \lambda$ | $\delta$ | Teaching set |
|---|---|---|---|---|
| 16 | A | 0.812 | 1.1 | {(∅, 0),(A,1)} |
| 160-960 | AC+BD | 0.9119 | 12 | {(A,0),(B,0),(C,0),(D,0),(AC,1),(BD,1)} |
| 1120-1840 | AC+BCD+AD | 0.9282 | 30 | {(A,0),(AC,1),(AD,1),(BC,0),(BD,0),(CD,0)} |
| 1920-2400 | AC+ABD+BCD | 0.9344 | 42 | {(AC,1),(AB,0),(AD,0),(BC,0),(BD,0),(CD,0),(ABD,1),(BCD,1)} |
| 3200 - $\infty$ | ABC+ABD+ACD+BCD | 0.94 | 60 | {(AB,0),(AC,0),(AD,0),(BC,0),(BD,0),(CD,0),(ABC,1),(ABD,1),(ACD,1),(BCD,1)} |

This second experiment also shows that it is not difficult to determine when the language used for the explanation leads to low fidelity and/or complex explanations. Actually, in this case, since the function captured by the AI model does not have a clean Boolean concept, we can detect that teaching will either lead to low fidelity or complex explanation (or both). In sum, the use of the complexity of the teaching set in the trade-off is not only the right choice when

doing example-based XAI but it also leads to the same insights as when the complexity of the concept is taken into account.

### 5.2 Different hypothesis spaces

This section will examine the effects of different hypothesis spaces (representation languages) between the AI model, the ground truth, the model of the learner $L_M$ and the actual human learner $L_H$. In our exemplar-based explanation system, we added letter rotations, letter resizing and letter location as cognitive noise and extra features, so that we have more confounders, and motivated by these spurious variations the neural network will have error with respect to the ground truth. This makes things more realistic, with the neural network creating patterns that are not fully captured by $L_M$. A neural network trained on images can in principle model functions over all possible images creating an enormous hypothesis space $H_{pix}$. On the other hand, the ground truth labelling function is on a small set of features, i.e. the presence or absence of $k$ letters, giving the small hypothesis space $H_{p^k}$, with $p = \{0, 1\}$ indicating the two possibilities of present or absent. When the actual human learner $L_H$ is given a set of labelled images from the example space $H_{pix}$, they will create a rule based on the features of the images they consider relevant.

Our exemplar-based explanation system has a focus on the simplicity of examples, and so far this has been with respect to the $\delta$ function. But simplicity also comes into play when generating images with certain letters present. The simplest images contain letters that all have the same size, with no rotation and with uniform placement, and these can be used as the simplified examples most compatible with the smaller example space $H_{p^k}$. The research question we want to address with the following experiment is whether using such simplified examples helps align the hypothesis spaces.

The experiment will compare the options for aligning the hypothesis spaces, by three groups that are given the teaching sets in different formats

- Group I: Use original images.
- Group II: Use simplified images, without irrelevant features.
- Group III: Use original images, but alert learners to essential features

We created a 2AFC (two-alternative forced choice) survey. Participants were shown a teaching set of carefully selected images and the (binary) classification of these images into Box 1 or Box 2 (True/False). They were then shown a test set of unclassified images and tasked to classify each image into one either Box 1 or Box 2. In Figure 2, we display how these tasks were presented to the participants.

The teaching sets were selected according to the system presented in the earlier sections. The test sets were randomly selected with the restriction that exactly one image should contain the same letter combination as one in the teaching set.

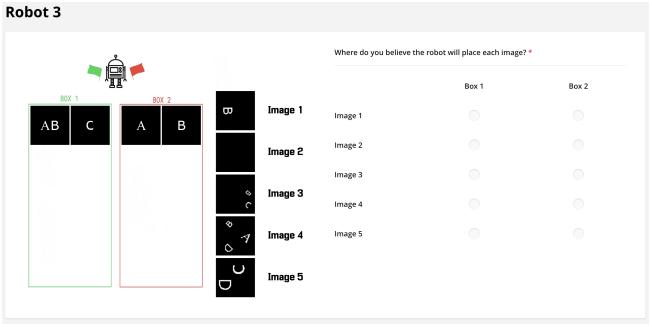


**Fig. 2.** Participants in group II were presented with this screen when tested on the formula $\phi_3$ (C or both A and B). Note the teaching set for Box 1 (True) and Box 2 (False) have images without the noise (rotation, resizing and relocation) found in the five images of the Test Set. For Groups I and III also the teaching set had noise. For group III the following text was displayed prominently: "NB: Size, placement and rotation do not matter. Focus on present/absent letters."

In total, we trained six AI models (called 'robots' in the survey) to be tested. All of them were trained to a high degree of accuracy. Each robot was trained on a different Boolean expression $\phi$.

We asked each participant to classify five test instances for each Boolean expression $\phi$. The participant's answer to the $i$th test is denoted $p(\phi, i)$. The correct answer to each test is given by $\phi(i)$. To calculate a participant's score for a single Boolean expression, we use the following formula: $p(\phi) = \frac{1}{6}\sum_{i=1}^{6}[p(\phi, i) = \phi(i)]$ where: $p(\phi, i) = \phi(i)$ is 1 if the participant's answer is correct and 0 otherwise. We then calculate the score of the $j$th participant across all Boolean expressions as follows: $p_j = \frac{1}{6}\sum_{i=1}^{6} p(\phi_i)$. Here, $p(\phi_i)$ represents the participant's score for the $i$th Boolean expression.

In total, we had 42 voluntary participants, who were master students, doctoral students or faculty in informatics, none of whom received compensation. The participants were presented with the survey and freely choose to participate. The participants were randomly assigned to the groups, with 12 participants in group I, 17 in group II, and 13 in group III.

The average scores for Groups I, II, and III were 0.564, 0.664, and 0.732, respectively. Furthermore, we observed that group III had the highest average score on 5 out 6 test instances. Though the results are not conclusive due to the small sample size[4], we decided to move on with option III, as we know that the teacher should do something to align the hypothesis spaces of $L_M$ and $L_H$ to achieve efficient learning. Next, we look at the quality of the teaching sets, by comparing our teacher to a teacher randomly selecting teaching sets of similar complexity, with both presenting the teaching sets as group III.

[4] We conducted t-tests for all pairs of groups to test whether means differ statistically significantly and got p-values 0.0297, 0.0013, and 0.0747 for pairs $(I, II)$, $(I, III)$ and $(II, III)$, respectively.

### 5.3 Compare to teaching sets chosen randomly

In this section, we discuss a second survey where we compare the teaching sets given by our exemplar-based explanation system (the smart teacher) to a system where the teaching sets are chosen randomly but correctly labelled and without repetitions (the random teacher).

To make the comparison of the random teacher and smart teacher fair, both will present their teaching sets as in group III. For a given Boolean formula, we first find the teaching set $S_S$ used by the smart teacher, and then we ensure that the random teaching set $S_R$ has a complexity ($\delta^*$-value[5]) close to $S_S$ (i.e. within some $\pm\epsilon$ additive difference) by choosing $S_R$ as follows, while making sure that no letter combination is repeated in $S_R$:

1. while $\delta(S_R) < \delta(S_S) - \epsilon \rightarrow$ add a random new image to $S_R$
2. if $\delta(S_S) - \epsilon \leq \delta(S_R) \leq \delta(S_S) + \epsilon \rightarrow$ use $S_R$
3. if $\delta(S_S) + \epsilon < \delta(S_R)$ then set $S_R = \emptyset$ and restart from 1.

To avoid bias from the previous survey, we changed most Boolean expressions for the new survey. They were (again) chosen with a variation in terms of expected difficulty, see Table 4. The formulas used can be found in Table 4.

**Table 4.** Boolean expressions used in the second survey.

| Nr | Prior $L_H$ | Short description | Boolean expression |
|---|---|---|---|
| $\phi_1$ | High | Less Than Two Letters | $(\neg A \wedge \neg B \wedge \neg D) \vee (\neg B \wedge \neg C \wedge \neg D) \vee$ $(\neg A \wedge \neg C \wedge \neg D) \vee (\neg A \wedge \neg B \wedge \neg C)$ |
| $\phi_2$ | Medium | A or both B and D | $(B \wedge D) \vee A$ |
| $\phi_3$ | Medium | B or D | $B \vee D$ |
| $\phi_4$ | High | Exactly One Letter | $(A \wedge \neg B \wedge \neg C \wedge \neg D) \vee (\neg A \wedge B \wedge \neg C \wedge \neg D) \vee$ $(\neg A \wedge \neg B \wedge C \wedge \neg D) \vee (\neg A \wedge \neg B \wedge \neg C \wedge D)$ |
| $\phi_5$ | Medium | No D | $\neg D$ |

Both groups $G_S$ given smart teaching sets and $G_R$ given random teaching sets will be shown the same test sets, and we now discuss how to generate the testing sets. When generating testing sets, we want them to be fair with regards to both teaching sets $S_S$ and $S_R$ so that none of them get an unfair advantage. Define $l(S)$ to be the set of letter combinations in the teaching set $S$, and define $X$ to be the set of all images. We generate test sets for $S_S$ and $S_R$ by choosing images from $X$ as follows, while ensuring that each letter combination appears at most once:

1. As long as there are new letter combinations in $l(X)/(l(S_R) \cup l(S_S))$, choose such an image at random.
2. Otherwise, fill the test set with images from the set of letter combinations in $l(S_R) \cap l(S_S)$, chosen randomly

[5] We use $\delta^* =$ *Number of present letters*

We select a test set of size five by the above protocol, for each Boolean expression, see Table 6. We will thus be able to make a fair comparison between the random and smart teaching sets.

**Table 5.** Teaching sets used for $G_S$ on left and $G_R$ on right (B1=Box 1, B2=Box 2).

| Nr | Teaching set group $G_S$ | $\delta^*(G_S)$ | Teaching set group $G_R$ |
|---|---|---|---|
| $\phi_1$ | B1:{ A , B , C , D }<br>B2:{ AB , AC , AD , BC , BD , CD } | 16 | B1:{ ∅ , A , B , C }<br>B2:{ AC , ACD , AD , BCD , CD } |
| $\phi_2$ | B1:{ A , BD } B2:{ B , D } | 5 | B1:{ A , AC } B2:{ B } |
| $\phi_3$ | B1:{ B , D } B2:{ ∅ } | 2.1 | B1:{ D } B2:{ C } |
| $\phi_4$ | B1:{ A , B , C , D }<br>B2:{ ∅ , AB , AC , AD , BC , BD , CD } | 16.1 | B1:{ A , B , C }<br>B2:{ AB , ABC , ABD , AD , BCD } |
| $\phi_5$ | B1:{ ∅ } B2:{ D } | 1.1 | B1:{ ∅ } B2:{ BD } |

**Table 6.** Test sets used for both $G_S$ and $G_R$.

| Nr | Test set |
|---|---|
| $\phi_1$ | { ABC , ABCD , ABD , C , AD } |
| $\phi_2$ | { ABD , ACD , C , ∅ , AD } |
| $\phi_3$ | { ABC , BD , ABCD , AB , ACD } |
| $\phi_4$ | { ACD , ABCD , B , AD , C } |
| $\phi_5$ | { AC , CD , A , ABC , BC } |

### 5.4 Overall results

We will now discuss the results of the second survey. In total, we had 56 participants, none of them overlapping with the previous test. The participants were students in a university-level informatics course. The participants were randomly assigned into two groups, with 22 participants in group $G_S$ and 34 in group $G_R$.

We start by looking at each group's average score for each $\phi_i$. The results are shown in Figure 3. Our initial observation is that the group $G_S$ has average accuracy over all 5 Boolean expressions of 0.809 versus 0.699 for the group $G_R$, suggesting that the smart teaching sets have an advantage. This difference is statistically significant ($p = 0.00016$ from t-test). There are two cases where $G_R$ exhibits slightly higher average accuracy than $G_S$, namely for $\phi_1$ and $\phi_4$. Notice these concepts are the ones we classified to have high prior for $L_H$ in Table 4 and if we look at Table 5 we see these are also the concepts where our automatic system generates teaching sets with large size ($\delta$-value). When the system is

| Task | Group $G_S$ | Group $G_R$ | $G_S + G_R$ |
|---|---|---|---|
| $\phi_1$ | 0.882 | 0.924 | 0.908 |
| $\phi_2$ | 0.954 | 0.759 | 0.836 |
| $\phi_3$ | 0.827 | 0.547 | 0.657 |
| $\phi_4$ | 0.873 | 0.888 | 0.882 |
| $\phi_5$ | 0.509 | 0.376 | 0.428 |
| Total | 0.809 | 0.699 | 0.742 |

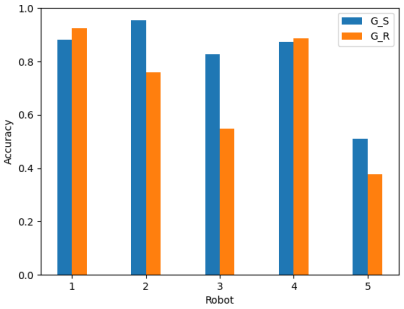


**Fig. 3.** The table shows average accuracy in Groups $G_S, G_R$ and $G_S + G_R$, for each $\phi_i$ and Total. The bar plot shows average accuracy in $G_S, G_R$ for each $\phi_i$.

aligned[6], as with $\phi_2$ and $\phi_3$, our system achieves substantially higher accuracy than the random teacher.

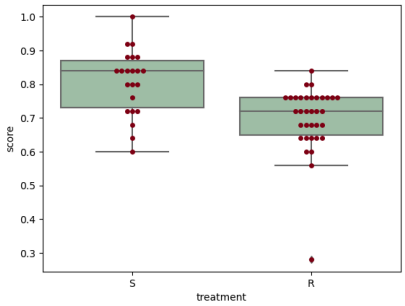


**Fig. 4.** Boxplot showing results of the two survey groups, to the left teaching sets with the exemplar-based explanation system, and to the right the random teaching sets. There is a clear difference between the groups.

Table 7 gives information on the most common answers for each robot. The most common answer vector of group $G_S$ is the correct one for 4 of the 5 $\phi_i$s, while for $G_R$ it is the correct one for 3 of the 5.

**5.4.1 Detailed discussion of $\phi_4$ (Exactly One Letter in Box 1)** Note in Table 7 that for $\phi_4$ a full 85 % of the participants in $G_R$ had all answers correct, whereas this drops to 64 % for group $G_S$. We believe this is because

[6] We say that the system is aligned when the prior of $L_M$ is similar to the prior of $L_H$.

**Table 7.** The most common answer for both groups. We show how correct the answer is, with 1.0 being all 5 tests correct, and we also show how common it is.

| Concept | Most common answer $G_S$ | Score [0..1] | Fraction of participants $G_S$ | Most common answer $G_R$ | Score [0..1] | Fraction of participants $G_R$ |
|---|---|---|---|---|---|---|
| $\phi_1$ | [2,2,2,1,2] | 1.0 | 59% | [2,2,2,1,2] | 1.0 | 76% |
| $\phi_2$ | [1,1,2,2,1] | 1.0 | 82% | [1,1,2,2,1] | 1.0 | 44% |
| $\phi_3$ | [1,1,1,1,1] | 1.0 | 68% | [2,1,1,2,1] | 0.6 | 29% |
| $\phi_4$ | [2,2,1,2,1] | 1.0 | 64% | [2,2,1,2,1] | 1.0 | 85% |
| $\phi_5$ | [2,2,2,2,2] | 0.2 | 41% | [2,2,2,2,2] | 0.2 | 53% |

the smart teaching set happens to be compatible with the (wrong) concept 'Odd Number of Letters'. Thus when shown the test set containing 'ACD' almost a third (7/22) of those thought with smart teaching set made a wrong choice, while less than a tenth (3/34) of those taught with the random teaching set selected the wrong box. The teaching sets are in Table 5. This is why we believe the random teaching set is slightly better (accuracy 0.888 vs 0.873, see Figure 3) for formula $\phi_4$ where the procedure built on Karnaugh map used in our system generates a very large smart teaching set.

We also asked participants for how they themselves would explain what they thought each robot was doing. This information is useful to elucidate why the smart teaching set does worse than the random teaching set on $\phi_4$.

In the group $G_S$ (the smart teaching set), 10 of the 22 subjects did not write any explanation while 12 subjects had an explanation. 7 people answered wrong for test 'ACD' and 3 of these had no explanation, whereas the other 4 confirm our suspicion that they are focusing on odd/even numbers of letters.

In the group $G_R$ (the random teaching set) 27 subjects had an explanation. 3 people answered wrong for test 'ACD' and 2 of these had no explanation, whereas the 3rd had an explanation that actually should have led the subject to classify 'ACD' correctly.

## 6 Conclusions

The results of the paper are indeed promising and have the potential to advance the field of explainable AI. Our proposed framework based on machine teaching can effectively teach complex functions to humans using explanatory examples, with a clear advantage over choosing the examples randomly. These findings demonstrate that machine teaching is a valid approach for exemplar-based explainable AI, but also that the expectations on the features and the priors of the humans is critical to get effective explanations from as few examples as possible. As future work, we propose the study of $L_M$ models better aligned with humans. Also, we are considering the use of teaching examples generated by recent language models.

## References


1. Basu, S., Christensen, J.: Teaching classification boundaries to humans. In: Twenty-Seventh AAAI Conference on Artificial Intelligence (2013)
2. Chen, Y., Mac Aodha, O., Su, S., Perona, P., Yue, Y.: Near-optimal machine teaching via explanatory teaching sets. In: International Conference on Artificial Intelligence and Statistics. pp. 1970–1978 (2018)
3. Domingos, P.: Knowledge discovery via multiple models. Intelligent Data Analysis **2**(1-4), 187–202 (1998)
4. Feldman, J.: Minimization of boolean complexity in human concept learning. Nature **407**(4), 630–633 (2000)
5. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Computing Surveys **51**(5), 93 (2018)
6. Hadfield-Menell, D., Russell, S.J., Abbeel, P., Dragan, A.: Cooperative inverse reinforcement learning. In: NIPS. pp. 3909–3917 (2016)
7. Hernández-Orallo, J.: Gazing into clever hans machines. Nature Machine Intelligence **1**(4), 172 (2019)
8. Hernández-Orallo, J., Ferri, C.: Teaching and explanations: aligning priors between machines and humans. Human-Like Machine Intelligence pp. 171–198 (2021)
9. Ho, M.K., Littman, M., MacGlashan, J., Cushman, F., Austerweil, J.L.: Showing versus doing: Teaching by demonstration. In: NIPS, pp. 3027–3035. Curran (2016), `http://papers.nips.cc/paper/6413-showing-versus-doing-teaching-by-demonstration.pdf`
10. Hoosain, R.: The processing of negation. Journal of Verbal Learning and Verbal Behavior **12**(6), 618–626 (Dec 1973). https://doi.org/10.1016/S0022-5371(73)80041-6, `https://www.sciencedirect.com/science/article/pii/S0022537173800416`
11. Karnaugh, M.: The map method for synthesis of combinational logic circuits. Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics **72**(5), 593–599 (Nov 1953). https://doi.org/10.1109/TCE.1953.6371932, conference Name: Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics
12. Khan, F., Mutlu, B., Zhu, J.: How do humans teach: On curriculum learning and teaching dimension. In: NIPS. pp. 1449–1457 (2011)
13. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
14. Lipton, P.: Contrastive explanation. Royal Institute of Philosophy Supplements **27**, 247–266 (1990)
15. Liu, W., Dai, B., Li, X., Liu, Z., Rehg, J.M., Song, L.: Towards black-box iterative machine teaching. arXiv preprint arXiv:1710.07742 (2017)
16. Molnar, C.: Interpretable machine learning. Lulu. com (2020)
17. Ortega, A., Fierrez, J., Morales, A., Wang, Z., Ribeiro, T.: Symbolic AI for XAI: Evaluating LFIT inductive programming for fair and explainable automatic recruitment. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 78–87 (2021)
18. Ouyang, L.: Bayesian inference of regular expressions from human-generated example strings. arXiv:1805.08427 (2018)

19. Pisano, G., Ciatto, G., Calegari, R., Omicini, A.: Neuro-symbolic computation for xai: Towards a unified model. In: WOA. vol. 1613, p. 101 (2020)
20. Rahwan, I., et al.: Machine behaviour. Nature **568**(7753), 477 (2019)
21. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-precision model-agnostic explanations. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
22. Samek, W., Müller, K.R.: Towards explainable artificial intelligence. In: Samek, W. (ed.) Explainable AI, pp. 5–22. Springer (2019)
23. Telle, J.A., Hernández-Orallo, J., Ferri, C.: The teaching size: computable teachers and learners for universal languages. Machine Learning (2019), https://doi.org/10.1007/s10994-019-05821-2
24. van der Waa, J., Nieuwburg, E., Cremers, A., Neerincx, M.: Evaluating XAI: A comparison of rule-based and example-based explanations. Artificial Intelligence **291**, 103404 (2021)
25. Yang, S.C.H., Vong, W.K., Sojitra, R.B., Folke, T., Shafto, P.: Mitigating belief projection in explainable artificial intelligence via Bayesian teaching. Scientific Reports **11**(1), 9863 (Dec 2021). https://doi.org/10.1038/s41598-021-89267-4, http://www.nature.com/articles/s41598-021-89267-4
26. Zhu, X.: Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In: AAAI. pp. 4083–4087 (2015)
27. Zhu, X., Singla, A., Zilles, S., Rafferty, A.N.: An overview of machine teaching (2018), http://arxiv.org/abs/1801.05927