

XOC: Explainable Observer-Classifier for Explainable Binary Decisions

Stephan Alaniz¹ Zeynep Akata¹

Abstract

When deep neural networks optimize highly complex functions, it is not always obvious how they reach the final decision. Providing explanations would make this decision process more transparent and improve a user’s trust towards the machine as they help develop a better understanding of the rationale behind the network’s predictions. Here, we present an explainable observer-classifier framework that exposes the steps taken through the model’s decision-making process. Instead of assigning a label to an image in a single step, our model makes iterative binary sub-decisions, which reveal a decision tree as a thought process. In addition, our model allows to hierarchically cluster the data and give each binary decision a semantic meaning. The sequence of binary decisions learned by our model imitates human-annotated attributes. On six benchmark datasets with increasing size and granularity, our model outperforms the decision-tree baseline and generates easy-to-understand binary decision sequences explaining the network’s predictions.

1. Introduction

A classification decision made by a deep neural network (DNN) is hard to interpret. However, to spread adoption and create a widespread acceptance by the user, the predictions of a neural network need to be explainable. An explanation can help an end-user to establish trust or help a machine-learning practitioner to understand or debug deep models. We distinguish between two types of explanations: post-hoc rationalizations and introspections. Post-hoc rationales are generated by a second neural network that justify the output of the first network, i.e., the decision maker. For instance, a language model explaining the classification decision of a vision model by talking about discriminative features of the object (Hendricks et al., 2016; 2018) falls into this category. Post-hoc rationales, however, do not guarantee

¹University of Amsterdam. Correspondence to: Stephan Alaniz <s.alaniz@uva.nl>.

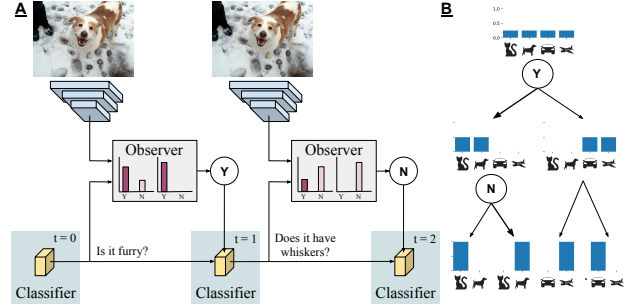


Figure 1. Our Observer-Classifier Framework. **A.** For classifying an image into cat, dog, car, plane categories, the *observer* combines image features with a query message of the *classifier* at each time step and creates a binary response. The classifier, i.e. an LSTM, uses its hidden state as a query message. At every step, the state of the LSTM is updated with the binary response to improve classification accuracy. **B.** The corresponding decision making process as the underlying decision tree.

that the explanation reflects the decision-maker’s internal thought process. We focus on introspective explanations revealing which pathway is taken to determine the final decision. Work on introspective explanations includes, e.g., visualizing features (Springenberg et al., 2014; Zhou et al., 2016; Selvaraju et al., 2017), saliency maps (Simonyan et al., 2013), interpretable features (Adel et al., 2018), and modular networks (Andreas et al., 2016). Among those, Andreas et al. (2016) expose a transparent network structure and Adel et al. (2018) give network features a semantic meaning. We combine both of these traits in a single model.

Our Observer-Classifier (OC) framework exposes a decision path in the form of an explainable decision tree. It breaks down the decision process into many small decisions via a two-agent setup (see Figure 1). The first agent, the classifier, does not have access to any visual information about the input. It formulates a question to the second agent, the observer, who uses a perception unit to respond to this question with a yes/no answer. By repeating these questions and their binary answers several times, the classifier obtains information about the input. The communication between the agents is learned end-to-end that allows us to condition the binary answers to be from a set of human-interpretable class-attributes, i.e., Explainable Observer-Classifier (XOC). When our XOC model solves a classification task, we extract explainable binary decisions that lead to the final prediction.

We emphasize that our OC framework uses hard yes/no decisions instead of binary distribution probabilities for the classification procedure. Because of the strict use of binary decisions, the decision trees as well as the resulting hierarchical clustering of the data become easy to interpret.

Furthermore, an additional loss on our binary decisions encourages to mimic attributes that attach a semantic meaning to the binary answers. We propose zero-shot learning as a natural test bed for evaluating the interpretability of the attributes learned by our model. Zero-shot learning relies on transferable side information in order to generalize from classes seen during training time to new classes only seen at test time. We show that our model learns discriminative class-level attributes that perform well in zero-shot learning.

Our contributions are as follows: 1) We propose a two-agent Observer-Classifer framework that learns to make iterative binary decisions to collaboratively solve an image-classification task. 2) We showcase on six datasets that our model outperforms classic decision trees and qualitatively demonstrate that our model learns attributes about the decision process that lead to explainable decision chains. 3) We propose zero-shot learning as an evaluation metric for interpretability and show that our learned attributes outperform Word2Vec-based semantic embeddings.

2. Related Work

Our work is related to the combination of deep learning and decision trees, multi-agent communication, and interpretable machine learning.

Decision Trees with Neural Networks. Adaptive Neural Trees (Tanno et al., 2018) directly model the neural network as a decision tree, where each node and edge correspond to one or more modules of the network. Our model is self-adapting through the use of a recurrent network in the classifier that makes a prediction at every node and can be easily rolled out to a greater depth without changing the architecture or number of weights. The prior work closest to ours is the Deep Neural Decision Forest (Kontschieder et al., 2016), which first uses a CNN to determine the routing probabilities on each node and then combines nodes to an ensemble of decision trees that jointly make the prediction. Similarly, in our Explainable Observer, we compute the binary decisions, i.e., router nodes of the tree, once before using them ad hoc. Our method differs in that we focus on explainability by explicitly only considering a hard binary decision at each node while other methods use soft decisions, making a large portion of the tree responsible for the predictions, and, thus, are harder to interpret.

Multi-Agent Communication. Learning to communicate in a multi-agent setting has recently gained interest in the research community mostly due to the emergence of deep

reinforcement learning (Foerster et al., 2016; Havrylov & Titov, 2017). Most related to our work, Foerster et al. (2016) propose the use of an agent that composes a message of categorical symbols at once and another agent that uses the information in these messages to solve a referential game. For discrete symbols, they also rely on the Gumbel-softmax estimator, but, in contrast to our model, their focus is not on explainability and their communication is not iterative, i.e., it concludes after one message, which does not allow the fine-grained introspection as in our model.

Explainability. The importance of explanations for an end-user has been studied from the psychological perspective (Lombrozo, 2012), showing that humans use explanations as a guidance for learning and understanding by building inferences and seeking propositions or judgments that enrich their prior knowledge. They usually seek explanations to fill the requested gap depending on prior knowledge and the goal in question.

In support of this trend, explainability has been recently growing as a field in computer vision and machine learning (Hendricks et al., 2016; Park et al., 2018; Andreas et al., 2016; Zintgraf et al., 2017). Following the convention of Park et al. (2018), our focus is on introspective explanations, where a deep network as the decision maker is trained to explain its own decision which is useful in increasing trust for the end user and provides a means to detect errors in the predictions of the model.

Textual explanations are explored by Hendricks et al. (2016) who propose a loss function based on sampling and reinforcement learning that learns to generate sentences that realize a global sentence property, such as class specificity. Andreas et al. (2016) compose collections of jointly trained neural modules into deep networks for question answering by decomposing the questions into their linguistic substructures and using these structures to dynamically instantiate modular networks with reusable components.

As for visual explanations, Zintgraf et al. (2017) propose to apply a prediction-difference analysis to a specific input. Park et al. (2018) utilize a visual-attention module that justifies the predictions of deep networks for visual question answering and activity recognition. Grad-CAM (Selvaraju et al., 2017) uses the gradients of any target concept, e.g., a predicted action, flowing into a convolutional layer to produce a localization map highlighting the important regions in the image that lead to predicting the target concept. Interpretable CNNs (Zhang et al., 2018a) modify the convolutional layer, such that each filter map corresponds to an object part in the image, and a follow-up work (Zhang et al., 2018b) uses a classical decision tree to explain the predictions based on the learned object-part filters.

3. Observer-Classifier Framework

Our framework is set up as a sequential interaction between two agents to solve an image-classification task by communicating. The first agent, the observer, holds information about the image $x \in \mathbb{R}^D$, by having access to the actual image or pre-extracted image features $z \in \mathbb{R}^Z$. The second agent, the classifier, predicts the associated ground-truth class $y \in \mathcal{Y}$ using only the messages broadcast by the observer, without having direct access to the image.

For a single image, the classifier sets its initial state to its prior belief of the class distribution $\hat{y}_0 \in \mathbb{R}^C$. In a balanced dataset, this would correspond to a uniform distribution over all the class labels. It then creates a query message $h_t \in \mathbb{R}^H$ sent to the observer requesting information about the image. The observer processes the query together with the input image to construct a binary response $d_t \in \{0, 1\}$. The classifier uses this one bit of information to update its state minimizing the classification error. This constitutes one iteration of the observer-classifier communication. The interaction repeats until a maximum number of steps is reached or until the classifier is confident in its decision.

The classification loss is minimized when the two agents jointly learn to communicate the most important bit of information about the image at each time step, such that the classifier’s label prediction improves. We deliberately limit the observer’s messages to be binary as clear yes/no answers are easier for humans to interpret than probability values.

3.1. Classifier

Main output of the classifier is the classification decision. The decision tree is a byproduct obtained from restricting the inputs of the classifier to be binary. Hence, the binary response sequence of the observer corresponds to one particular path along the decision tree. Since the classifier only takes discrete inputs, we can map out all possible binary sequence paths up to a desired length at test time, which provides us with a binary tree structure. We construct the classifier as an LSTM (Hochreiter & Schmidhuber, 1997), making each node of the decision tree correspond to a hidden state of the LSTM and the classifier’s prediction distribution.

We use the hidden state $h_t \in \mathbb{R}^H$ of the LSTM as a query message for the observer. Each iteration the classifier receives the observer’s binary decision together with the message from the previous time step $[d_{t-1}, h_{t-1}]$ to update its state and, thus, produce h_t . The message h_t is also used to make the class prediction using an affine transformation followed by a softmax:

$$\hat{y}_t = \text{softmax}(Wh_t + b) \quad (1)$$

with $W \in \mathbb{R}^{|\mathcal{Y}| \times H}$, $b \in \mathbb{R}^{|\mathcal{Y}|}$. Since the primary objective of

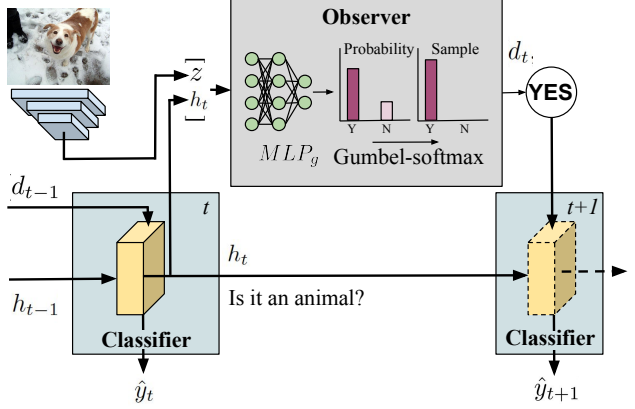


Figure 2. General Observer-Classifier (OC) Model: The classifier sends a query message h_t to the observer that the observer combines with image features z to construct a binary response d_t using its MLP and the Gumbel-softmax estimator. The response is fed into the classifier in the next time step, such that the predicted class distribution \hat{y} better fits the true class label.

the classifier is to maximize the classification performance, we minimize the cross-entropy loss of the predicted class probabilities \hat{y}_t and the true class probabilities y :

$$\mathcal{L}_{CE}(y, \hat{y}_t) = - \sum_i y_i \log \hat{y}_{t,i}. \quad (2)$$

While decision trees usually employ a classification loss on the leaf nodes, we use the cross-entropy loss also after every binary decision. This results in the loss function:

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^T \mathcal{L}_{CE}(y, \hat{y}_t) \quad (3)$$

where we enforce a loss on the current classification distribution after each iteration of the observer-classifier communication loop. This encourages the network to predict the correct class in as few communication steps as possible.

3.2. Observer: Binary Decisions

The observer model can have different architectures depending on whether or not certain capabilities are required. We present the General Observer, which optimizes for classification accuracy without using attribute data, as well as the Explainable Observer that introduces an additional attribute loss for human interpretability.

General Observer. Our General Observer architecture is outlined in Figure 2. In this setting, the observer consists of a multi-layer perceptron (MLP) that takes as inputs both the image features and the message from the classifier $[z, h_t]$ and produces the binary decision d_t with $MLP_g : \mathbb{R}^{Z+H} \rightarrow \{0, 1\}$. Since we require the output of MLP_g to be a discrete binary signal, its construction needs

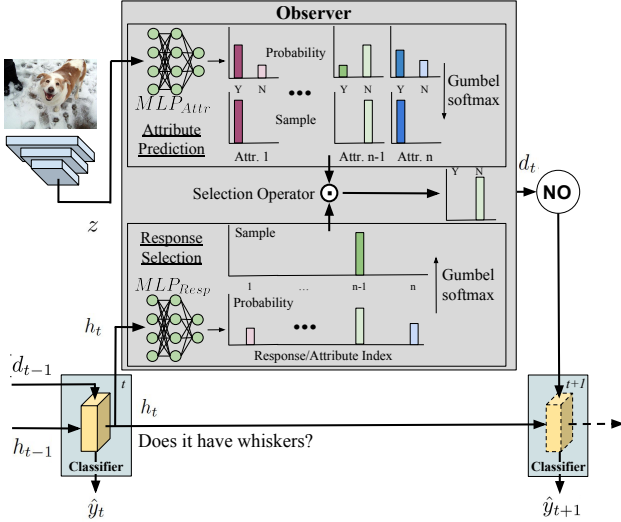


Figure 3. Explainable Observer-Classifier (XOC) Model: Our explainable observer uses two MLPs to process z and h_t separately. From image features z_t , it predicts binary attributes, while from the query message h_t , it learns to pick an attribute as a response to the classifier. We impose a loss on our learned attributes to match them with human-annotated attributes to make them understandable for the user.

to be differentiable in order to train our model end-to-end. The Gumbel-softmax estimator (Jang et al., 2017; Maddison et al., 2017) allows us to sample from a discrete categorical distribution and use the reparameterization trick to obtain the gradients of this sampling process. To obtain a binary sample with the Gumbel-softmax estimator, we sample g_i from a Gumbel distribution and then calculate a continuous relaxation of the categorical distribution

$$d_i = \frac{\exp((\log o_i + g_i)/\tau)}{\sum_{j=1}^K \exp((\log o_j + g_j)/\tau)} \quad (4)$$

where $\log o$ is the unnormalized output of our MLP and τ is the temperature that parameterizes the discrete approximation. When τ approaches 0, the output becomes a one-hot vector (binary when $K = 2$) and otherwise, it is a continuous signal. Popular training strategies include annealing the parameter τ over time or augmenting the Gumbel-softmax with an $\arg \max$ function that discretizes the activation in the forward pass and a straight-through identity function in the backward pass. We resort to the second strategy since doing so guarantees the response d_t from the observer to be always binary during training, which empirically leads to better results.

Explainable Observer. In addition, we develop our Explainable Observer (see Figure 3), where the observer needs to first predict a set of binary attributes for the image independent of the query message of the classifier. The query message h_t is then used to decide on the attribute as a re-

sponse, which could be seen as the classifier requesting a particular attribute at each communication time step. The perception module is extended with an *Attribute MLP*, i.e., MLP_{Attr} that outputs a set of binary attributes $\hat{a} \in \{0, 1\}^A$ with the dimensionality A chosen in advance. Each of these attributes uses the Gumbel-softmax estimator with $K = 2$ to sample and discretize them as binary features. Once the binary attribute vector is obtained, a *Response MLP*, i.e., MLP_{Resp} takes the query message from the classifier as input and produces a probability distribution $\pi_t \in \mathbb{R}^A$ over the binary attributes to indicate which one serves as the response to the classifier. The choice distribution π_t over the attributes is again discretized using the Gumbel-softmax estimator resulting in a one-hot vector c_t of size A . The final binary decision $d_t = \hat{a} \odot c_t$ is computed by selecting the position encoded by c_t from \hat{a} (denoted by \odot).

Attribute Loss for Explainable Observer. The classification loss allows to learn both the observer and the classifier in conjunction. Minimizing this loss at each time step is equivalent to finding the binary split of the data that reduces the class-distribution entropy the most. In this regard, it is similar to what is usually referred to as information gain in classical decision trees. However, a split that best separates the data is not always easy to interpret, especially when the features used to do this split result from a non-linear transformation such as, in our case, with the CNN.

We, therefore, propose to align the binary decisions with class-level attributes corresponding to human-interpretable features. We consider attributes to be assigned a binary value, e.g., statements that are either true or false about a particular entity in question, such as “the object is red”. Since this information is only needed on the class level, minimal additional supervision is required to obtain this labeled data. As the Explainable Observer predicts a set of A binary attributes, we add a second cross-entropy term for these learned attributes to be consistent with the human-labeled attributes:

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^T \left[\mathcal{L}_{CE}(y, \hat{y}_t) + \lambda \mathcal{L}_{CE}(\alpha_{y, c_t}, \hat{a}_{c_t}) \right] \quad (5)$$

weighted by a hyperparameter λ . α_{y, c_t} corresponds to the true attribute label of class y that was chosen by the observer at time step t and \hat{a}_{c_t} is the matching inferred attribute of the same choice c_t . This loss encourages the network to learn attributes that agree with human-annotated attributes while optimizing for classification accuracy. Note that the attribute loss is only imposed on those attributes employed by the model. When $\lambda > 0$, our model uses ground-truth attributes in order to give the split a semantic meaning. In this case, we set the number of learned attributes A to the number of ground-truth attributes. However, our model learns to use only a subset of the ground-truth attributes, focusing on

	AWA2	CUB	aPY	MNIST	CIFAR-10	ImageNet
attributes	available			not available		
# image	37K	11K	15K	70K	60K	1.2M
# class	50	200	24	10	10	1K
scale	medium			small		large
difficulty	coarse	fine		coarse		fine

Table 1. A summary of the datasets that we use in terms of the availability of attributes, number of images, and number of classes. As our model tackles image classification, we group the datasets based on scale in terms of classes (medium, small, and large-scale) and based on difficulty (coarse-grained and fine-grained)

the ones that minimize the cumulative attribute loss. When $\lambda = 0$, i.e., our model does not use any human-annotated attributes, it automatically discovers attributes to be used as side information for zero-shot learning.

4. Experiments

In this section, we describe our experimental setup, provide quantitative and qualitative results demonstrating the performance of our model and evaluate the attributes learned by our model in zero-shot learning.

4.1. Experimental Setup

Here we detail our setup in terms of datasets, side information and experimental settings.

Datasets. We experiment on six datasets (see Table 1). MNIST (Lecun et al., 1998) and CIFAR-10 (Krizhevsky, 2009) are small-scale datasets in terms of the number of classes, i.e., MNIST consists of 60K/10K training/test examples from 10 handwritten digits, CIFAR-10 contains 50K/10K training/test examples from 10 classes. ImageNet (Russakovsky et al., 2015) is a large-scale dataset with 1.2 million high-resolution images from 1000 categories. We validate our model on classification accuracy on MNIST, CIFAR-10, and ImageNet as attributes are not available. AWA2 (Lampert et al., 2014), CUB (Wah et al., 2011), aPY (Farhadi et al., 2009) are three benchmark zero-shot attribute datasets. AWA2 comprises 37,322 images from 50 animal classes annotated with 85 attributes, e.g., furry, red, etc., while CUB contains 11,788 images from 200 different bird species with 312 attributes, and aPY contains 15,339 images from 32 classes with 64 attributes.

Side Information. The attributes are collected manually from experts by asking the relevance of an attribute for each class. Since our model does not consider splits on soft probabilities but rather on hard binary decisions, it is beneficial to have binary attribute data. When we train our explainable model with attribute loss, we binarize all the

attributes with a threshold at 0.5, i.e., an attribute is present if more than 50% of the annotations agree.

The attribute quality directly affects the zero-shot-learning performance. Hence, evaluating our model on this task shows the effectiveness of our XOC model in attribute prediction. We compare the learned attributes by our model with Word2Vec feature vectors (Mikolov et al., 2013) extracted from Wikipedia articles (Akata et al., 2015).

Setting. For fully supervised learning, we randomly assign 20% of each class as test data for image classification when an official test set is not provided. In all the experiments across the datasets, we randomly separate 10% of the training data as a validation set. The MLPs consist of two layers with a ReLU non-linearity in between. It is beneficial to learn the temperature hyper-parameter τ of the Gumbel-softmax estimator jointly with the network.

4.2. Decision-Tree Baseline

We use the classical decision tree as a baseline for evaluating our OC framework. A decision tree is fitted on top of the image features extracted from ResNet (He et al., 2016) using one dimension of the features at a time to make a split, e.g., $z_{42} < 0.23$, until a leaf node only contains samples of the same class or a regularization strategy leads to early stopping. Since no semantic meaning is attached to a split in the tree, this model serves as the decision-tree baseline optimized for classification accuracy.

For the explainable decision-tree baseline, we first train a linear layer on top of the image features to predict the binarized class attributes (instead of the class labels) for each image using a binary cross-entropy loss as in our XOC model. Secondly, we predict attributes for each image and fit a decision tree to determine the class label. We emphasize that the network will predict probabilities per attribute which we binarize before fitting the decision tree. This is important to be comparable with our proposed XOC model that also uses binary attributes and to be in line with the yes/no questions for improved interpretability. If we did not enforce this, the decision tree could, for instance, learn two subsequent splits, such as $p(\text{"swims"}) < 0.8$ and $p(\text{"swims"}) > 0.6$, to isolate samples that cover a certain range of probabilities for an attribute, which weakens the explainability. Equivalently to our XOC model, the explainable decision-tree baseline splits on whether an attribute is present or not.

We use the Gini impurity index as splitting criterion because it has a slight computational advantage over entropy-based methods (Raileanu & Stoffel, 2004), such as information gain. We report the outcome with the best validation results after randomized hyperparameter search on regularization parameters, i.e., minimum sample size for splits, minimum reduction in impurity per split.

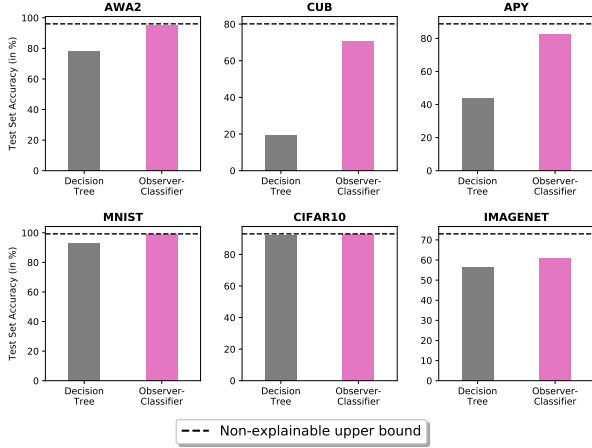


Figure 4. Ablation Study: Our OC model (General Observer without any attribute loss) is compared to the decision-tree baseline as well as the non-explainable upper bound (softmax on image features) for image classification. In all cases, image features come from a state-of-the-art CNN pre-trained on ImageNet (except for MNIST) and fine-tuned on the respective datasets.

4.3. Image Classification with OC

We compare the classifier of our OC model with the softmax classifier as the non-explainable upper bound and with the classical decision tree as the baseline. We use a simple CNN for MNIST, ResNet-18 for CIFAR-10, and ResNet-152 pre-trained on ImageNet and fine-tuned on each of the datasets for the remaining datasets.

Figure 4 shows the classification accuracy of our OC model compared with the decision-tree baseline as well as the upper bound for each dataset. During training of the upper-bound model, the softmax classifier can update the weights of the CNN perception module. We fix the CNN’s weights from the upper-bound model and use it to extract image features for the decision tree and our model. While our model works with constrained single-bit communications to improve explainability, it succeeds in maintaining the same classification accuracy as the upper-bound model on the medium/small-scale and coarse-grained datasets (AWA2, MNIST, CIFAR-10). Furthermore, our model exposes a binary decision-tree structure and hierarchical clustering of the data as additional outcome that improves interpretability of the predictions. For instance, on CIFAR-10, we observe that our OC model separates the animal classes from the vehicles in the first binary decision. While this helps in getting a better understanding on the data, it can also be used to debug failure cases by analyzing at which part in the decision-tree path a mistake occurred.

We also observe that our OC model consistently outperforms the decision-tree baseline across all datasets. On CUB and aPY specifically, the classification accuracy of our model is

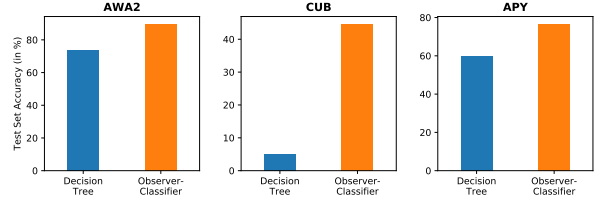


Figure 5. Our XOC model, where each decision node has a human-interpretable meaning, is compared to the decision-tree baseline. In all cases, image features come from a state-of-the-art CNN pre-trained on ImageNet (except for MNIST) and fine-tuned on the respective datasets. The attribute loss is weighted with $\lambda = 0.25$.

3.5 and 2 times higher, respectively. Fine-grained datasets, such as CUB, are challenging to explain because their classification relies on nuances. Moreover, this makes it hard for non-experts to judge the correctness of the predictions, making explanations particularly important. Thus, achieving good results on CUB helps in addressing this issue. ImageNet poses an extreme challenge for both decision-tree approaches due to being a large-scale dataset (1,000 classes and 1.2 million images) that requires significantly bigger trees than the other datasets. In addition to outperforming the decision-tree baseline in terms of accuracy, our OC model scales better with increasing tree size in large-scale datasets. This is due to increasing the tree depth simply translates to incrementing the number of binary decisions, i.e., time steps of the Observer-Classifier communication. Therefore, our model scales linearly with the depth of the tree while the number of weights stays constant as opposed to classical decision trees that grow exponentially with the depth of the tree.

4.4. Explainable Image Classification with XOC

We present results of our Explainable Observer-Classifier (XOC) model, where we include the attribute loss to incorporate explainable binary decisions. We set λ to 0.25 across all experiments, as it leads to the best results based on grid search on the validation set.

Comparing our explainable model with the explainable decision-tree baseline in Figure 5, we observe that on AWA2 and aPY, the classification accuracy comes close to the performance of the model without attribute loss. This shows that constraining the model to use attributes as binary decisions does not harm classification accuracy and a discriminative decision tree can be learned either way. On the other hand, the CUB dataset is more challenging. Since the dataset is fine grained, distinguishing between closely related classes require a large number of class attributes, which leads to sparse attribute vectors and an imbalanced decision tree. Nonetheless, our XOC achieves nine-times-higher accuracy than the decision tree on CUB. It also sig-

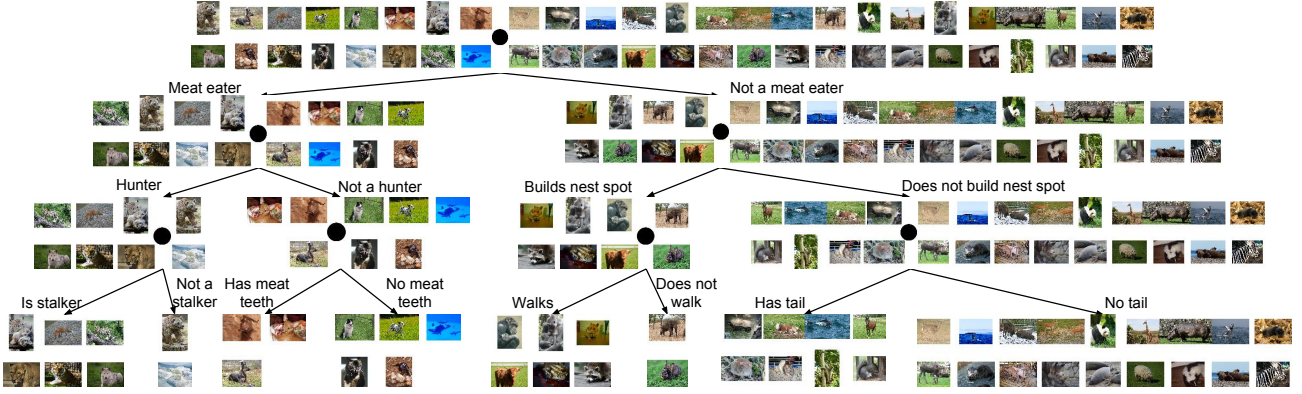


Figure 6. Learned Explainable Decision Tree on AWA2 using our Explainable Observer-Classifier model and the attribute loss. We show the first decisions of the most likely path for each class and give each decision a human-understandable meaning based on the class-attribute that was used at each node. The tree exposes a transparent overview of how our model comes to its classification conclusions, e.g., it decides to separate meat-eating animals from all other animals in the first step.

nificantly outperforms the decision tree on AWA2 and aPY. These results show the benefit of our XOC model from the joint optimization of the classification loss as well as the attribute loss. The decision-tree baseline first optimizes for the best attribute prediction and then builds the tree on top of these features. By doing this process collectively, our model can choose to ignore attributes that are not suitable for good decision-tree splits since predicting these attributes would only constitute to a larger penalty in the attribute loss. We specifically design the attribute loss, such that it only acts on attributes our XOC model uses. Fine-grained datasets, such as CUB, show the difficulty in predicting these attributes, where the decision-tree baseline performs particularly poorly.

4.5. Introspective Explanations: Visualizing Decisions

The main benefit of the XOC model is that classification predictions can now be explained with a human-understandable thought process, which we demonstrate qualitatively. Our model reveals its decision-making process by pointing to the tree branch, into which a certain image falls, and the attribute being chosen at each node. We also inspect the learned structure of the decision tree by illustrating the splits from our model on AWA2. For more examples, we refer to the supplementary material.

In Figure 6, we observe the separation of classes in three decision steps and each decision is associated with a human-interpretable attribute. The left/right sub-tree indicates that the attribute is present/absent respectively. By visualizing the tree, we can get an explainable overview of the internal decision process of the whole classifier. For instance, the first decision deals with identifying meat-eating animals and those who do not eat meat. With this attribute split,

our model effectively separates dogs, bears, cats, big cats, and foxes from all the other animals. These categories get further refined with each binary split building a hierarchical clustering that defines the XOC model’s decision structure. Since the pool of attributes determines the vocabulary of the explanations, it is worth considering different sets of attributes depending on the use case, e.g., if only attributes that refer to visual features are desired.

Explanations are often contrastive (Hendricks et al., 2018). In addition to justifying a positive decision, our model can reason about negative decisions. When images are misclassified, we inspect the point in the tree where the error occurred, exposing detailed information of when the features are mistaken to be from another class.

As an example, in Figure 7, we show the first six decisions for two tiger images. The lower path corresponds to when the model thinks the attribute is present for a given class. Both images follow the same path for five decisions that indicate the animal is a meat-eater, a hunter, a stalker, is strong, and lives near bushes. The last shown decision of whether the animal has stripes is different for the two images. For one of the tigers, our model decides it does not have stripes. Therefore it is ultimately incorrectly classified as a lion, while the other image is correctly classified as a tiger. In addition, our XOC model depicts its current belief of the correct class at any time during the process. This also reveals some critical binary decisions, such as the one shown in the example of whether the animal has stripes or not. In fact, we observe a high probability of “tiger” if the answer is “yes” and a high probability of “not tiger” if the answer is “no”. This way, a user inspecting the individual rationals can make a more informed decision on the value of the model’s predictions.

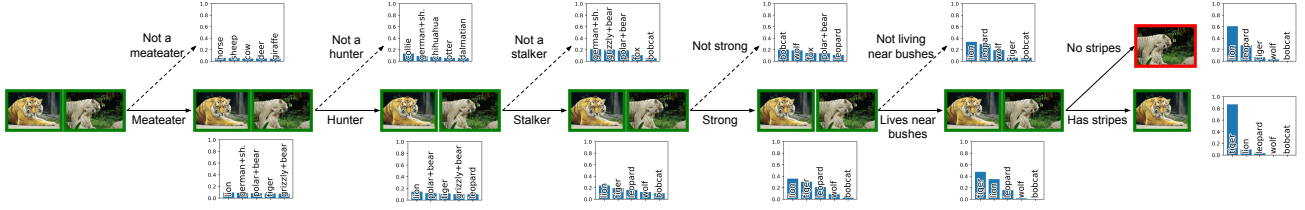


Figure 7. Qualitative Explanation of Classification of Two Tigers in AWA2. We show the binary decisions made for two images of a tiger along with the current label prediction at each step. The lower path corresponds to when the attribute is present for a given class. Both images follow the same path except for the last shown decision of whether the animal has stripes. The one tiger, for which our model decides there are no stripes, becomes ultimately classified incorrectly as a lion.

side info	superv.	AWA2 (d)	CUB (d)	aPY (d)
att	human	66.2 (85)	47.6 (312)	38.0 (64)
att (ours)	learned	54.1 (40)	43.6 (100)	36.0 (30)
w2v	learned	41.0 (400)	25.9 (400)	34.6 (400)

Table 2. Zero-Shot Learning on AWA2, CUB, aPY using SJE. We compare side information (d = features’ dimensionality): human-annotated attributes (att), learned attributes by the XOC model (att (ours), $\lambda = 0$), learned attributes by Word2Vec (w2v)

4.6. Explanations help other tasks: Zero-Shot Learning

Explanations are useful when they enable solving an independent task (Lombrozo, 2012). We argue that zero-shot learning is suitable for this purpose because solving this task requires using interpretable features as side information.

In zero-shot learning, the training and test classes are disjoint. In order to predict the unseen class for a query image, the model needs to do information transfer. Side information, often in a complementary modality, is used as a means of such a transfer as it associates seen and unseen classes. The most widely used side information comprises human-annotated attributes. In our XOC model, in order not to use any expert annotation we set $\lambda = 0$. After obtaining the probabilities of each learned binary attribute via softmax, we stack the attributes in a per-class attribute vector and scale the attribute values to be between -1 and 1. For a particular class, the attributes are averaged over all training set images. We compare against Word2Vec (Mikolov et al., 2013) trained on Wikipedia articles to make a fair comparison with a technique that does not use human annotations as side information for zero-shot learning. Our experiments follow the recommendations of Xian et al. (2017), i.e., we use the same image features and their proposed train-test split. As the zero-shot learning model, we use Structured Joint Embedding (SJE) (Akata et al., 2015), the best zero-shot-learning performance on AWA2 according to Xian et al. (2017).

Our results in Table 2 show that while expert-annotated attributes reach 66.2% accuracy on AWA2, the attributes learned by our XOC model achieve 54.1% accuracy, significantly outperforming Word2Vec with 41.0%. This behavior generalizes to other datasets. On aPY, we observe that our learned attributes (36.0%) come close to the performance of human-annotated attributes (38.0%) and on the CUB dataset, our model (43.6%) outperforms Word2Vec (25.9%) by a large margin. This result is encouraging as it demonstrates that our learned attributes lead to discriminative and interpretable representations that are useful for tackling the challenging task of zero-shot learning. It also shows that our model learns representations more discriminative than the ones Word2Vec extracted from Wikipedia articles.

Another interesting observation is that while the dimensionality of Word2Vec equals 400, our learned attributes reach a significantly better accuracy using considerably fewer attributes: 40 on AWA2, 30 on aPY, and 100 on CUB. The number of attributes learned for CUB is required to be larger as CUB is a fine-grained dataset. These results suggest that the hierarchical clustering from our decision tree carries an interpretable meaning that generalizes to unseen classes.

5. Conclusion

In this work, we presented a two-agent framework that tackles the image-classification task using human-interpretable binary decisions in the form of a hierarchical process. Trained end-to-end, our model achieves marginally close accuracy as the state-of-the-art non-explainable models while revealing its internal thought process. In addition, our XOC model relates its binary decisions to human-understandable concepts. The hierarchical clustering and explainable binary decisions allow to better understand the iterative predictions of the network as well as help identify failure cases at test time. We propose zero-shot learning as an evaluation metric for evaluating explanations and show promising results validating that our model indeed learns transferable and discriminative binary features across classes.

References

- Adel, T., Ghahramani, Z., and Weller, A. Discovering interpretable representations for both deep generative and discriminative models. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.
- Akata, Z., Reed, S. E., Walter, D., Lee, H., and Schiele, B. Evaluation of output embeddings for fine-grained image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. Neural module networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- Foerster, J. N., Assael, Y. M., de Freitas, N., and Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2016.
- Havrylov, S. and Titov, I. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Advances in Neural Information Processing Systems*, 2017.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, 2016.
- Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., and Darrell, T. Generating visual explanations. In *ECCV*, 2016.
- Hendricks, L. A., Hu, R., Darrell, T., and Akata, Z. Grounding visual explanations. In *ECCV*, 2018.
- Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8), 1997.
- Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017.
- Kontschieder, P., Fiterau, M., Criminisi, A., and Bulò, S. R. Deep neural decision forests. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI*, 2016.
- Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.
- Lampert, C. H., Nickisch, H., and Harmeling, S. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3), 2014.
- Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.
- Lombrozo, T. *Explanation and abductive inference*. The Oxford handbook of thinking and reasoning, 2012.
- Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 2013.
- Park, D. H., Hendricks, L. A., Akata, Z., Rohrbach, A., Schiele, B., Darrell, T., and Rohrbach, M. Multimodal explanations: Justifying decisions and pointing to the evidence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- Raileanu, L. E. and Stoffel, K. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1), 2004.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 2015.
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision, ICCV*, 2017.
- Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.
- Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. A. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014.
- Tanno, R., Arulkumaran, K., Alexander, D. C., Criminisi, A., and Nori, A. V. Adaptive neural trees. *CoRR*, abs/1807.06699, 2018.
- Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

- Xian, Y., Schiele, B., and Akata, Z. Zero-shot learning - the good, the bad and the ugly. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2017.
- Zhang, Q., Nian Wu, Y., and Zhu, S.-C. Interpretable convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018a.
- Zhang, Q., Yang, Y., Wu, Y. N., and Zhu, S. Interpreting cnns via decision trees. *CoRR*, abs/1802.00121, 2018b.
- Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., and Torralba, A. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2016.
- Zintgraf, L. M., Cohen, T. S., Adel, T., and Welling, M. Visualizing deep neural network decisions: Prediction difference analysis. In *ICLR*, 2017.

Supplementary Material for: XOC: Explainable Observer-Classifer for Explainable Binary Decisions

A. Qualitative Results on aPY

We trained our XOC model on the aPY dataset equivalently as on AWA2 reported in the paper. In Figure 8, we show the explainable decision tree learned by our XOC model. The left/right path of each node indicate the presence/absence of the human-interpretable attribute used to make the decision. In Figure 9, we illustrate a qualitative example of the classification of two images of chairs made by our model. Both follow the upper branch that indicates the chairs are not furry, not a vertical object, not from metal, and do not have a snout. In the last shown step that decides whether or not it is made of wood, the decision is different for the chairs, and the one not made of wood according to our model is ultimately classified incorrectly. This serves as another example of introspection that our model allows to make a more informed decision about the value of the network’s prediction.

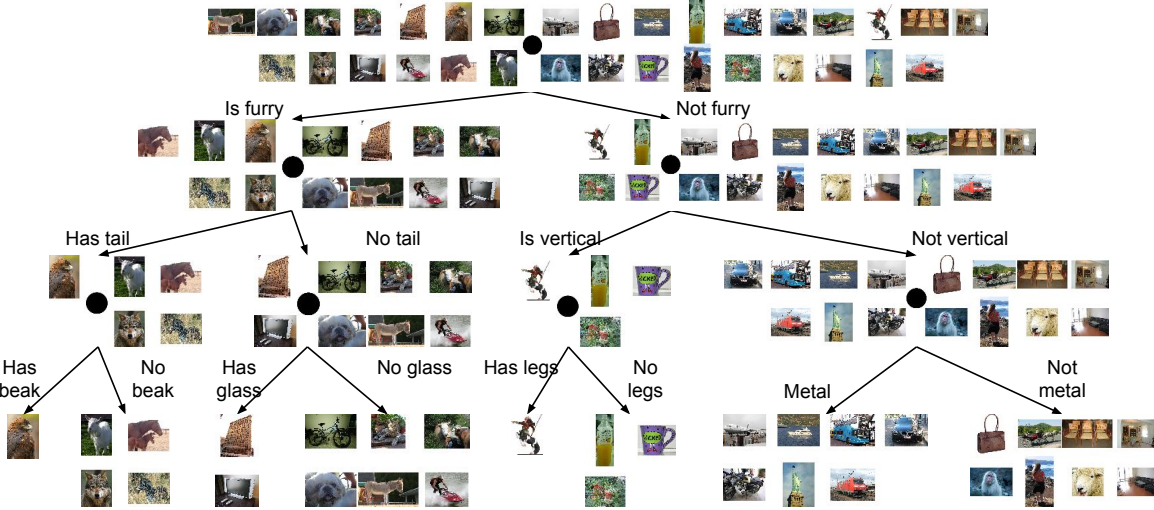


Figure 8. Learned Explainable Decision Tree on aPY using our XOC model and the attribute loss ($\lambda = 0.25$). We show the first decisions of the most likely path for each class and give each decision a human-understandable meaning based on the class attribute that was used at each node.

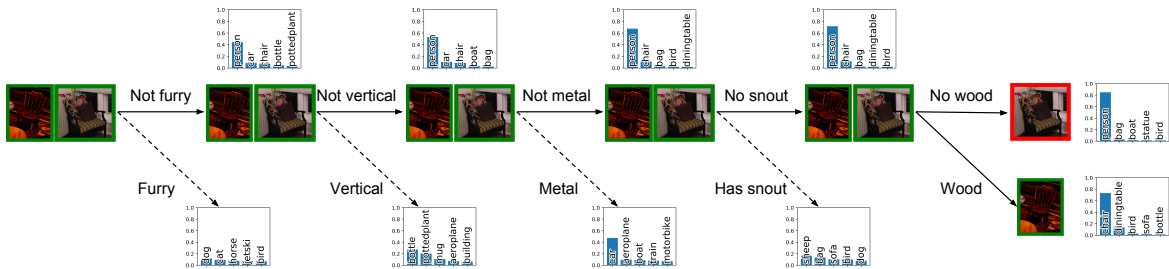


Figure 9. Qualitative Explanation of Classification of Two Chairs in APY. We show the binary decisions made for two images of a chair along with the current label prediction at each step. The upper path corresponds to when the attribute is not present for a given class. Both images follow the same path except for the last shown decision of whether the object is made of wood. The one chair for which our model decides it is not made of wood is ultimately incorrectly classified.