Explainable Observer-Classifier for Explainable Binary Decisions

Stephan Alaniz
Max Planck Institute for Informatics
salaniz@mpi-inf.mpg.de

Zeynep Akata
University of Tübingen
zeynep.akata@uni-tuebingen.de

Abstract

Explanations help develop a better understanding of the rationale behind the predictions of a deep neural network and improve trust. We propose an explainable observer-classifier framework that exposes the steps taken through the decision-making process in a transparent manner. Instead of assigning a label to an image in a single step, our model makes iterative binary sub-decisions, and as a byproduct reveals a decision tree in the form of an introspective explanation. In addition, our model creates rationalizations as it assigns each binary decision a semantic meaning in the form of attributes imitating human-annotations. On six benchmark datasets with increasing size and granularity, our model outperforms classical decision-trees and generates easy-to-understand binary decision sequences explaining the network's predictions.

1. Introduction

Often the classification decision of a deep neural network (DNN) is hard to interpret, hindering its practical employment in critical applications such as health-care. 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. The former is generated by a second neural network that justifies the output of the decision maker, e.g., a language model explaining the classification decision of a vision model by mentioning discriminative features of the object [11, 12]. Post-hoc rationales help the end user to understand the model by generating explanations in a way similar to how humans justify behaviors/decisions of each other, e.g., "The car is accelerating because there are no other car in its lane" [16]. On the other hand, introspections may reveal the internal thought process of the decision maker to help a machine learning practicioner to understand of the model, e.g., by visualizing features [36, 43, 34, 37], saliency maps [35, 29], interpretable features [1], and modular networks [3]. In this work, we combine both of these traits in a single model.

Figure 1: Our explainable decision tree model is composed of two agents. The *decision-tree classifier* (blue) asks questions that expect a yes/no answer from the *binary-attribute observer* (red). At every step, the classifier uses the binary response to update its class prediction. The thought process resembles a binary decision tree.

We propose to formulate the classification task as a communication protocol between two agents. Our Explainable Observer-Classifier (XOC) framework exposes a decision path in the form of an explainable decision tree by breaking down the decision process into many small decisions (see Figure 1). Our setup consists of a decision-tree classifier (blue) agent and a binary-attribute observer (red) agent. The blue agent does not have access to any image information, but instead has to infer the image class by asking binary questions about the image. The red agent tries to answer these questions with yes/no by looking at the image. This process is repeated until the classifier reaches a decision about the image class. For instance in Figure 1, by asking if the object is furry and receiving the answer "yes", the classifier learns that the object is an animal rather than a vehicle. The second question – "does it have whiskers?" – then allows the classifier to discriminate between the remaining two classes (cat and dog) to conclude that the object is a dog. The introspective explanations correspond to the binary subdecisions that form an interpretable decision tree whereas the post-hoc rationales are obtained by conditioning the binary answers to a set of human-interpretable class-attributes, giving the communication a semantic meaning.

Our contributions are: 1) We propose to integrate in-

trospection and post-hoc explanations into a single framework where two agents collaboratively solve an image-classification task via explainable binary decisions. 2) We showcase on six datasets that our model outperforms classic decision trees and qualitatively demonstrate that our model learns attributes that lead to explainable decision chains. 3) We propose zero-shot learning as a testbed for interpretability and show that our learned attributes outperform semantic embeddings extracted from Wikipedia.

2. Related Work

We review prior work on deep learning with decision trees, multi-agent communication, and interpretable machine learning related to ours.

Decision Trees with Neural Networks. Adaptive Neural Trees [38] 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 by using a recurrent network in the classifier that 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 (dNDF) [19], 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. Our method differs in that we focus on explainability by explicitly only considering a hard binary decision at each node while dNDF uses soft decisions, making a large portion of the tree responsible for the predictions, and, thus, it is harder to interpret.

Multi-Agent Communication. Learning to communicate in a multi-agent setting has gained interest with the emergence of deep reinforcement learning [9, 10, 22, 4, 15, 7, 6]. For instance, image reference games are used to study the emergence of language [22] and how agents can learn to communicate more effectively even when concepts are being misunderstood [6]. Most related to our work, [9] propose the use of an agent that composes a message of categorical symbols 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, the focus is not on explainability, i.e., it does not allow the fine-grained introspection.

Explainability. The importance of explanations for an end-user has been studied from the psychological perspective [24, 27], 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 about the goal in question.

Explainability has been recently growing as a field in computer vision and machine learning [11, 28, 3, 44]. Textual explanations are explored by [11] where the task is to generate sentences that realize class specificity and image relevance. [3] 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, [44] propose to apply a prediction-difference analysis to a specific input. [28] utilize a visual-attention module that justifies the predictions of deep networks. Grad-CAM [34] uses the gradients of any target concept to produce a localization map highlighting the important regions in the image that lead to the prediction based on an intermediate network layer. FullGrad [37] combines importance scores of both the intermediate feature maps and the input to capture both local and global relevance with respect to the network's output. Interpretable CNNs [41] modify the convolutional layer, such that each filter map corresponds to an object part in the image, and a follow-up work [42] uses a classical decision tree to explain the predictions based on the learned object-part filters. [5] establishes the connection between causality and network attribution and [33] propose a model agnostic way of training a causal explanation model that interprets a given predictive model.

Following the convention of [28], our model combines introspective explanations where a deep network as the decision maker is trained to explain its own decision with post-hoc rationalizations which is useful in increasing trust for the end user in a single framework.

3. Explainable Observer-Classifier (XOC)

Our XOC model is set up as a sequential interaction between two agents to solve an image-classification task through communication. Having two different agents insures the two types of explanations that our framework provides. The classifier's decision tree allows introspection while the observer's predicted attributes provide rationales to make the communication human-understandable. In the following, we first explain how the observer-classifier communication is achieved and then detail the two agents in our XOC framework, i.e., the decision tree classifier and the binary attribute observer.

3.1. Observer-Classifier Communication

For any single image, our classifier starts with no prior knowledge. It sends a query message c_t to the observer requesting information about the image x or pre-extracted image features z. The observer answers the query c_t in regard to the image x with a binary response $d_t \in \{0, 1\}$. The classifier uses d_t to update its belief on the class y that the image belongs to. This constitutes one iteration t 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 objective of the two agents is to jointly learn to communicate the most im-
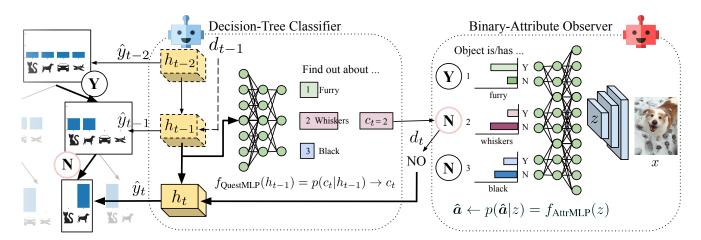
Figure 2: XOC Model. A single communication step is shown. The *decision-tree classifier* uses the hidden state h_{t-1} of its LSTM (yellow) to choose a single attribute a_{c_t} through its f_{QuestMLP}. The classifier requests from the observer whether the attribute is present or absent in the image. The *binary-attribute observer* uses its f_{AttrMLP} to decide about the presence/absence of attributes in the image using CNN features z. As a binary response the observer returns its prediction $d_t = \hat{a}_{c_t}$. Finally, the classifier updates its state h_t with the binary response to improve its classification prediction $\hat{y}_t$.

portant bit of information about the image at each time step, such that the classifier's label prediction improves.

Communication Protocol. Each word in the vocabulary corresponds to a binary attribute $a \in A$ that the observer predicts about a given image. The observer can learn to attach a human-understandable meaning to the words when annotated attribute data is available. This vocabulary is usually different for each dataset, but otherwise remains fixed.

At each communication step t, the classifier chooses one attribute a_{c_t} from the vocabulary, identified by its index c_t, and requests the presence or absence of this attribute in the image. The observer then provides this binary information d_t based on its own prediction. We deliberately limit the observer's messages to be binary as for humans, clear yes/no answers are easier to interpret than probability values.

Discrete Messages. Both the classifier and the observer communicate with discrete messages. The classifier indicates the attribute whose presence/absence is desired via the attribute index c_t and the observer produces the binary response d_t. We need to ensure that c_t and d_t are discrete signals, while at the same time being differentiable to train the model end-to-end. The Gumbel-softmax estimator [14, 25] allows to sample from a discrete categorical distribution via the reparameterization trick [18, 31] to obtain the gradients of this sampling process. To get a discrete sample with the Gumbel-softmax estimator, we sample g_i from a Gumbel distribution and then compute a continuous relaxation of the categorical distribution

$$\text{GumbelSoftmax}(\log \boldsymbol{\pi})_i = \frac{\exp((\log \pi_i + g_i)/\tau)}{\sum_{j=1}^{K} \exp((\log \pi_j + g_j)/\tau)} \quad (1)$$

where $\log \pi$ are the unnormalized log-probabilities of the categorical distribution 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 as it guarantees the communication signals to always be discrete during training.

3.2. Decision-Tree Classifier

The main output of the classifier is the classification decision. The decision tree is a byproduct obtained by restricting the observer's response to be binary. 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 [13] and a question-decoder module, *Question MLP* (see Figure 2). The LSTM contains a hidden state h_t that encodes both the information the classifier gathers about the image and the question the classifier wants to pose to the observer.

To decide which attribute information to request, the classifier uses f_{QuestMLP} decoding the last hidden state h_{t-1} into a categorical distribution

$$\log p(c_t|h_{t-1}) = f_{\text{QuestMLP}}(h_{t-1}) \quad (2)$$

where $p(c_t|h_{t-1})$ indicates the likelihood of requesting a particular attribute from the observer. We denote the at-

tribute index $c_t \in \{1, \dots, |A|\}$ as a sample from $p(c_t|h_{t-1})$ obtained by applying the Gumbel-softmax estimator

$$c_t = \text{GumbelSoftmax}(f_{\text{QuestMLP}}(h_{t-1})). \qquad (3)$$

Hence, c_t becomes a scalar index communicated to the observer, which, in return, responds whether the attribute with that index is present or absent in the image.

After each iteration of the communication loop, the classifier updates its hidden state with the new information from the observer's binary response d_t to update its internal state

$$h_t = \text{LSTM}(h_{t-1}, d_t). \qquad (4)$$

Apart from generating the question, the hidden state h_t is also used to predict the class label

$$\hat{y}_t = \text{softmax}(W h_t + b) \qquad (5)$$

optimized with parameters W and b.

Since the primary objective of 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:

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_{CE}(y, \hat{y}_t) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{i} y_i \log \hat{y}_{t,i}. \qquad (6)$$

By averaging the cross-entropy loss over all T time steps, we encourage the model to predict the correct class in as few communication steps as possible.

3.3. Binary Attribute Observer

The observer converts the binary responses of the classifier to human interpretable attributes via an attribute-prediction module, *Attribute MLP* (see Figure 2) by predicting a set of learned binary attributes $\hat{a}$ about the image, independent of the classifier's query. To do this, the observer feeds its CNN image features z to f_{AttrMLP} that models a probability distribution over a set of learned binary attributes

$$\log p(\hat{a}|z) = f_{\text{AttrMLP}}(z). \qquad (7)$$

By applying the Gumbel softmax estimator, we obtain binary attributes $\hat{a} \in \{0, 1\}^{|A|}$, an instantiation of $p(\hat{a}|z)$:

$$\hat{a} = \text{GumbelSoftmax}(f_{\text{AttrMLP}}(z)). \qquad (8)$$

The predicted attributes $\hat{a}$ are either discovered end-to-end with a classification loss as formulated in Equation 6 or they correspond to human-interpretable concepts formalized by the attribute loss as in Equation 9.

Whenever the classifier requests a particular attribute with its query c_t, the observer simply returns the binarized attribute of the specified index to the observer. We denote this selection operation as $d_t = \hat{a}_{c_t}$.

Attribute Loss. Minimizing the classification 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 perception CNN.

We propose to learn attributes $\hat{a}$ that align with class-level human-annotated attributes α, and thus, making them interpretable. A second cross-entropy term encourages this correspondence:

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\left[\mathcal{L}_{CE}(y, \hat{y}_t) + \lambda \mathcal{L}_{CE}(\alpha_{y,c_t}, \hat{a}_{c_t})\right] \qquad (9)$$

weighted by a hyperparameter λ. Here, α_{y,c_t} corresponds to the ground-truth attribute label of class y that matches the observer's response $\hat{a}_{c_t}$. The final loss ($\mathcal{L}$) 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. If an attribute is deemed to not be useful, e.g., if an attribute is weak or hard to predict, the classifier can learn to ignore requesting that attribute. In that case, the attribute loss is not applied on that attribute, focusing the observer's training signal on predicting informative attributes.

When $\lambda > 0$, our model learns to use ground-truth attributes in order to give the binary splits a semantic meaning. In this case, we can translate the communication into natural language. For instance, the classifier's question for attribute with index c_t can be interpreted as "does it have whiskers?" when a_{c_t} corresponds to attribute "has whiskers". When $\lambda = 0$, our model does not use any human-annotated attributes and automatically discovers attributes with no additional supervision. Either of these settings may be desirable given the application as we empirically show in the following section.

4. Experiments

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

4.1. Experimental Setup

Datasets and attributes. We experiment on six datasets (see Table 1 for a summary of dataset statistics). The small-scale MNIST [23] consists of 60K/10K training/test examples from 10 handwritten digits and CIFAR-10 [20] contains 50K/10K training/test examples from 10 classes

	AWA2	CUB	aPY	MNIST	CIFAR-10	ImageNet
Attributes	available			not available		
# of images	37K	11K	15K	70K	60K	1.2M
# of classes	50	200	24	10	10	1K
Dataset size	medium			small		large
Difficulty	coarse	fine		coarse		fine

Table 1: A summary of the datasets in terms of the availability of attributes, number of images/classes, dataset size (medium, small, and large-scale) and difficulty (coarse-grained and fine-grained)

whereas the large-scale ImageNet [32] contains 1.2 million high-resolution images from 1000 categories. We validate our model on classification accuracy on MNIST, CIFAR-10, and ImageNet as human-annotated attributes are not available. AWA2 [21], CUB [39], aPY [8] are three benchmark attribute datasets proposed by the computer vision community. 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.

Our XOC with the attribute loss requires ground-truth attributes. The attributes are collected manually by asking the relevance of an attribute for each class to experts. Since our model does not consider splits on soft probabilities but rather on hard binary decisions, it is beneficial to have binary attribute data. We binarize the attributes on all datasets with a threshold at 0.5, i.e., an attribute is present if more than 50% of the annotations agree.

Experimental setting. Unless it is stated explicitly, for all experiments across the datasets we randomly assign 20% of each class as test data for image classification when an official classification test set is not provided. We randomly split 10% of the training data as a validation set to tune hyperparameters. The MLPs consist of two layers with a ReLU nonlinearity in between. It is beneficial to learn the temperature hyper-parameter τ of the Gumbel-softmax estimator jointly with the network so we only choose an initial value for τ. During training, we always roll out the decision sequence to a maximum number of steps and during testing we stop as soon as the classifier reaches a confidence level specified by a *threshold* parameter (or once the maximum number of decisions is reached, whatever happens first). We run all experiments 5 times and report the mean and standard deviation of the performance measures, e.g., classification accuracy. Values for all hyperparameters can be found in the supplementary material.

4.2. Baseline and Ablation Study

In this section, we compare our model XOC and its ablation IOC with two classical decision tree baselines XDT and DT as explained below. Note that the image features are extracted from 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 AWA2, aPY, CUB and ImageNet. The resulting softmax classifier serves as non-explainable deep neural network that corresponds to the upper bound [1]. After this pretraining, we fix the weights of the perception module and extract the same z for our explainable and introspective observer-classifier, i.e., XOC and IOC, as well as the explainable and non-explainable decision tree, i.e., XDT and DT.

Our model and ablations. Our Explainable Observer-Classifier (XOC) model uses the attribute loss to incorporate explainable binary decisions. Hence, this model provides both introspection and rationalization. On the other hand, our Introspective Observer-Classifier (IOC) does not use an attribute loss, and therefore purely optimizes classification performance. Although it does not associate the binary decision with human-interpretable attributes, i.e., no rationalization, this model still provides a tree of the decision process, i.e., introspection.

Baselines. Our first baseline is the classical decision tree (DT) on top of the same image features z from the perceptual module. At each time step, the dataset is split using a single dimension of z until a leaf node only contains samples of the same class or a regularization strategy leads to early stopping. In this case, no semantic meaning is attached to a split in the tree so this model served as a baseline for IOC, only introspection and no rationalization.

To be directly comparable with XOC, we incorporate rationalizations into the decision tree baseline and call it the Explainable Decision Tree (XDT) as our second baseline. We give each split a semantic meaning by training a tree on top of predicted attributes. First, we train a MLP on top of the image features z to predict the binarized class attributes for each image using a binary cross-entropy loss analogously to the attribute loss of our XOC model. Secondly, we fit a decision tree on these predicted attributes for each image to determine the class label. Equivalently to our XOC model, the explainable decision-tree baseline splits on whether an attribute is present or not.

For these decision tree baselines, we use the Gini impurity index as splitting criterion because it has a slight computational advantage over entropy-based methods [30], such as information gain. We report the outcome with the best validation results after randomized hyperparameter

---

[1]Note that, the non-explainable state-of-the art for AWA2, aPY, CUB, MNIST, CIFAR-10 and ImageNet is 96.04%, 88.78%, 80.12%, 99.28%, 93.02%, 73.01% respectively.

Model	AWA2	aPY	CUB	MNIST	CIFAR-10	ImageNet
DT (baseline)	78.03 ± 0.39	64.29 ± 0.64	19.33 ± 0.29	93.08 ± 0.22	92.46 ± 0.18	55.21 ± 1.03
IOC (ours)	$\mathbf{95.34} \pm 0.19$	$\mathbf{82.59} \pm 0.81$	$\mathbf{70.55} \pm 0.38$	$\mathbf{99.06} \pm 0.10$	$\mathbf{93.12} \pm 0.32$	$\mathbf{60.77} \pm 3.58$
XDT (baseline)	73.92 ± 0.93	59.90 ± 1.45	04.87 ± 1.31	N/A	N/A	N/A
XOC (ours)	$\mathbf{89.78} \pm 2.50$	$\mathbf{76.69} \pm 3.95$	$\mathbf{44.68} \pm 6.15$	N/A	N/A	N/A

Table 2: Comparing our model XOC (with attributes) and its ablation IOC (without attributes) to the decision-tree baselines XDT and DT respectively (for XOC, $\lambda = 0.25$). MNIST, CIFAR-10 and ImageNet do not have attributes, i.e., XOC and XDT are not applicable. We report the standard deviation over 5 runs. The rationalizations are not available (N/A) for MNIST, CIFAR-10 and ImageNet as these datasets do not have human-annotated attributes.

search on regularization parameters, i.e., minimum sample size for splits, minimum reduction in impurity per split.

Results. From the results in Table 2 we observe that our model variants consistently outperform the decision-tree variants across all datasets. On CUB specifically, the classification accuracy of IOC is 3.5 times higher than DT (70.55% vs 19.33%) and XOC is 11 times higher than XDT (44.68% vs 4.87%). The decisions on fine-grained datasets, e.g. CUB, are extremely challenging to explain because they rely on nuances. As it is hard for non-experts to judge the correctness of the predictions, explanations in this domain are particularly important.

ImageNet poses an extreme challenge for being a large-scale dataset that requires significantly bigger trees than the other datasets. As increasing the tree depth simply translates to increasing the number of binary decisions, i.e., time steps of the Observer-Classifier communication, 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. Hence, in addition to outperforming the decision-tree baseline in terms of accuracy (60.77% vs 55.21%), our IOC model scales better with increasing tree size.

We also observe a tradeoff between classification accuracy and explainability across all datasets, e.g., classification accuracy decreases with improved explainability. In other words, introspective observer classifier (IOC) without the attribute loss achieves higher accuracy than explainable observer-classifier (XOC) with the attribute loss. Although for the challenging fine-grained dataset CUB the gap is larger, for the coarse-grained attribute datasets such as AWA2 and aPY, the gap reduces as expected. The CUB dataset is more challenging due to its fine grained nature as distinguishing between closely related classes requires a large number of class attributes, which leads to sparse attribute vectors and an imbalanced decision tree.

Compared to the Decision Tree baselines of their kind, i.e., IOC vs DT and XOC vs XDT, our model variants achieve significantly higher accuracy across all datasets. By

Images	AWA2	aPY	CUB
Correct	97.22 ± 0.71	89.55 ± 1.67	75.27 ± 0.28
Incorrect	83.95 ± 1.57	74.39 ± 3.70	71.52 ± 2.35
All	96.16 ± 0.92	84.45 ± 0.82	72.94 ± 0.59

Table 3: Agreement of XOC attributes with human-labeled attributes. We report the agreement in percent on correctly / incorrectly classified images and their average.

jointly optimizing the classification loss and the attribute loss, our XOC model can choose to ignore attributes that are not suitable for good decision-tree splits since predicting these attribute would only contribute 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.

As a side-note, although our introspective model (IOC) works with constrained single-bit communications to improve explainability, it succeeds in maintaining the accuracy of the non-explainable state-of-the-art on the medium or small-scale and coarse-grained datasets such as AWA2 (95.34% vs 96.04%), MNIST (99.06% vs 99.28%), and CIFAR-10 (93.12% vs 93.02%) which is already quite encouraging.

4.3. Agreement with Human-Judgement

The attribute datasets contain the information that for every image a certain attribute is present or absent judged by an expert annotator. Using these expert judgements, we can evaluate our XOC model on how accurate its explainable decisions are. We extract the binary decisions of XOC (with $\lambda = 0.25$) on the test set of AWA2, aPY, and CUB and calculate the percentage of how often a predicted attribute of XOC matched the ground-truth human label on a per-image basis. Thus, the agreement for a single image is calculated as $\frac{1}{T} \sum_t \mathbb{1}_{\alpha_{y,c_t} = \hat{a}_{c_t}}$.

We report the attribute agreement averaged by the correctly or incorrectly classified images and all images in Table 3. Across datasets, the agreement drops significantly
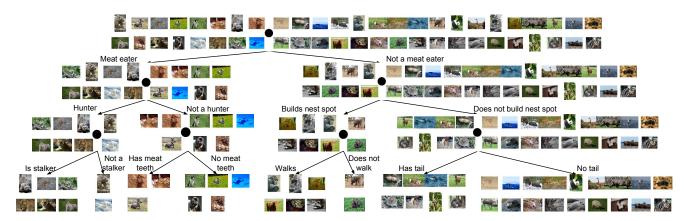
Figure 3: Learned explainable decisions on AWA2 by our XOC model. We show the decision tree of the most likely path for each class, i.e., introspection, and give each decision a human-understandable meaning, i.e., rationalization. The tree exposes the thought process of our model, e.g., it decides to separate meat-eating animals from all other animals in the first step.

whenever an image is incorrectly classified. For instance while the agreement between human annotators and our learned attributes for correctly classified images is 97.22% on AWA2, it reduces to 83.95% for incorrectly classified images. Note that on an average our learned attributes are in line with the human judgement, e.g., 96.16% on AWA2. This shows that our model is faithful to its explanations as judged by experts.

The practical implication of this experiment is as follows. Since humans understand the individual interpretable concepts, a user can spot inconsistencies in the explanation and, based on that, identify wrong predictions during test time because the explanation is flawed more often whenever the model is making a mistake. This is a step towards one of the goals of explainable AI, i.e., to allow users to make judgement calls whether or not to trust the system based on its explanations.

4.4. Qualitative Results

The main premise of our explainable observer-classifier (XOC) model is that the decision making process is explained by pointing to the tree branch, into which a certain image falls, and by the attribute being chosen at each node. By visualizing the tree, the user can get an explainable overview of the internal decision process of the whole classifier.

We inspect the learned structure of the decision tree by illustrating the splits from our model on AWA2 in Figure 3, where the left and right sub-tree indicates that the attribute is present or absent respectively. The first decision deals with identifying meat-eating animals, separating 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 de-
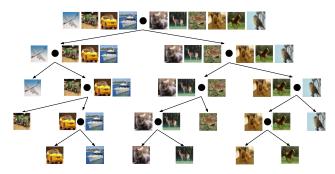

Figure 4: Learned decision tree on CIFAR-10 by our IOC ablation model which does not use attribute data, i.e., decisions cannot be named. We show the decision tree of the most likely path for each class, i.e., we can introspect that our model decides to separate animals from vehicles first.

termines the vocabulary of the explanations, it is worth considering different types of attributes depending on the use case, e.g., when one desires to only use visual attributes.

On the other hand, our introspective observer-classifier (IOC) ablation model does not use the attribute loss, exposing the decision tree structure without assigning a semantic meaning to the decisions. Although IOC fails to assign an attribute to each branch, it still provides introspection into the model's intermediate class splits and can be applied on datasets without attribute information. Figure 4 shows the decision tree of IOC when applied on CIFAR-10, where we observe that the model separates the animal classes from the vehicles in the first binary decision.

Zooming into the decision process on AWA2, we can investigate how our model treats counterfactuals, i.e., negative decisions. This is useful for the user as explanations are often contrastive [12]. In Figure 5, when an image is misclassified, we inspect the point in the tree where the error
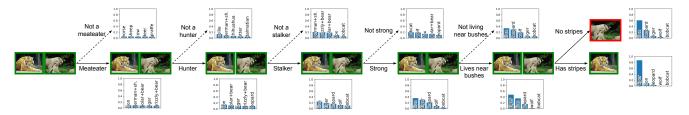
Figure 5: Decision process for two tiger images in AWA2 along with the current label prediction at each step. The lower (upper) path is taken when the attribute is present (absent) for a given class. Both images follow the same path except for the last decision of "stripes". For the white tiger as there are no stripes and it gets classified incorrectly as a lion.

side information		AWA2 (d)	aPY (d)	CUB (d)
w2v	learned	41.0 (400)	34.6 (400)	25.9 (400)
IOC att	learned	54.1 (40)	36.0 (30)	43.6 (100)
att	expert	66.2 (85)	38.0 (64)	47.6 (312)

Table 4: Zero-shot learning with human-annotated attributes (att), attributes learned by our IOC (IOC att, i.e., $\lambda = 0$), by Word2Vec (w2v). (d = dimension)

occurred, exposing detailed information of when the image is mistaken to be from another class. 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, the error occurs in the sixth decision. For the white tiger, our model decides "no stripes" and incorrectly classifies it as a lion. In addition, our XOC model depicts its current belief of the correct class at any time during the process, i.e., probability plots at every branch. This also reveals some critical binary decisions, when the predicted class changes drastically, such as the "has stripes" decision. This way, a user inspecting the individual rationals can make a more informed decision on the value of the model's predictions.

4.5. Zero-Shot Learning as a Testbed

Explanations are useful when they enable solving an independent task [24]. We argue that zero-shot learning (ZSL) is suitable for this purpose because solving this task requires using interpretable features as side information. Hence, the attribute quality directly affects the ZSL performance and evaluating our model on this task shows the effectiveness of our learned attributes.

In ZSL since the training and test classes are disjoint, to predict the unseen class for a query image the model needs to transfer information using some form of side information, e.g., expert-annotated attributes. For a fair comparison, i.e., in order not to use any expert annotation, we set $\lambda = 0$ as in our IOC model. 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 images of the dataset. We use the same image features and proposed train-test split as in [40]. For zero-shot prediction, we use the SJE [2] technique and compare ours with another learned embedding, Word2Vec [26].

Our results in Table 4 show that the attributes learned by our IOC model achieve 54.1% accuracy, significantly outperforming Word2Vec with 41.0%. This behavior generalizes to other datasets. In fact, on aPY our learned attributes come close to the performance of human-annotated attributes (36.0% vs 38.0%). Moreover on the CUB dataset, our model outperforms Word2Vec by a large margin (43.6% vs 25.9%). 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 extracted from Wikipedia while being much lower dimensional, i.e., 40 vs 400 on AWA2, 30 vs 400 on aPy and 100 vs 400 on CUB. These results suggest that the hierarchical clustering from our decision tree carries an interpretable meaning without requiring any expert annotation.

5. Conclusion

In this work, we presented a two-agent framework that tackles the image-classification task using human-interpretable binary decisions as a hierarchical decision process. Trained end-to-end, our model achieves competitive accuracy to the state-of-the-art non-explainable models by revealing its internal thought process, i.e., introspection, and by relating its binary decisions to human-understandable concepts, i.e., rationalization. As indicated by our results that compare human-annotated attributes with our learned attributes, the hierarchical clustering and explainable binary decisions allow the user to better understand the iterative predictions of the network as well as help identify failure cases at test time. Proposing zero-shot learning as a testbed to evaluate explanations, we show promising results validating that our model indeed learns transferable and discriminative binary features across classes.

References

[1] Tameem Adel, Zoubin Ghahramani, and Adrian Weller. Discovering interpretable representations for both deep generative and discriminative models. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018. 1

[2] Zeynep Akata, Scott E. Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 8

[3] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2

[4] Kris Cao, Angeliki Lazaridou, Marc Lanctot, Joel Z. Leibo, Karl Tuyls, and Stephen Clark. Emergent communication through negotiation. In *International Conference on Learning Representations ICLR*, 2018. 2

[5] Aditya Chattopadhyay, Piyushi Manupriya, Anirban Sarkar, and Vineeth N. Balasubramanian. Neural network attributions: A causal perspective. In *Proceedings of the 36th International Conference on Machine Learning ICML*, 2019. 2

[6] Rodolfo Corona, Stephan Alaniz, and Zeynep Akata. Modeling conceptual understanding in image reference games. In *Advances in Neural Information Processing Systems*, 2019. 2

[7] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, 2019. 2

[8] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 5

[9] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2016. 2

[10] Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Advances in Neural Information Processing Systems*, 2017. 2

[11] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *ECCV*, 2016. 1, 2

[12] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *ECCV*, 2018. 1, 8

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8), 1997. 3

[14] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. 3

[15] Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. In *Neural Information Processing Systems, NeurIPS*, 2018. 2

[16] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self driving vehicles. In *European Conference of Computer Vision (ECCV)*, 2018. 1

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*, 2015. 12

[18] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 3

[19] Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulò. Deep neural decision forests. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI*, 2016. 2

[20] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 5

[21] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3), 2014. 5

[22] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In *International Conference on Learning Representations ICLR*, 2018. 2

[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998. 5

[24] T. Lombrozo. *Explanation and abductive inference.* The Oxford handbook of thinking and reasoning, 2012. 2, 8

[25] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017. 3

[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 2013. 8

[27] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019. 2

[28] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[29] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: randomized input sampling for explanation of black-box models. In *British Machine Vision Conference BMVC*, 2018. 1

[30] Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1), 2004. 6

[31] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference

in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML*, 2014. 3

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 2015. 5

[33] Patrick Schwab and Walter Karlen. Cxplain: Causal explanations for model interpretation under uncertainty. In *Advances in Neural Information Processing Systems*, 2019. 2

[34] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision, ICCV*, 2017. 1, 2

[35] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013. 1

[36] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014. 1

[37] Suraj Srinivas and François Fleuret. Full-gradient representation for neural network visualization. In *Advances in Neural Information Processing Systems*, 2019. 1, 2

[38] Ryutaro Tanno, Kai Arulkumaran, Daniel C. Alexander, Antonio Criminisi, and Aditya V. Nori. Adaptive neural trees. *CoRR*, abs/1807.06699, 2018. 2

[39] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5

[40] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning - the good, the bad and the ugly. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2017. 8

[41] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[42] Quanshi Zhang, Yu Yang, Ying Nian Wu, and Song-Chun Zhu. Interpreting cnns via decision trees. *CoRR*, abs/1802.00121, 2018. 2

[43] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2016. 1

[44] Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. In *ICLR*, 2017. 2

Supplementary Material for
Explainable Observer-Classifier 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 6, we show the explainable decision tree learned by our XOC model. The left/right path of each node indicates the presence/absence of the human-interpretable attribute used to make the decision. In Figure 7, 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.
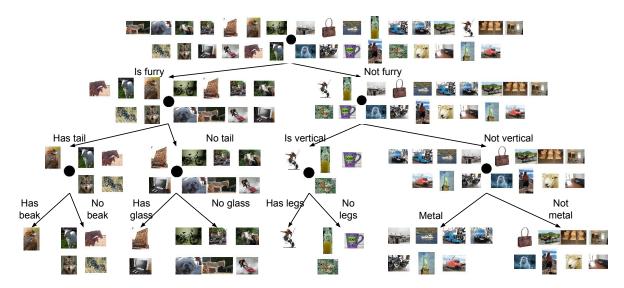

Figure 6: Learned Explainable Decision Tree on aPY using our XOC model and the attribute loss ($\lambda = 0.25$). We show the first decisions of the most likely path for each class and give each decision a human-understandable meaning based on the class attribute that was used at each node.
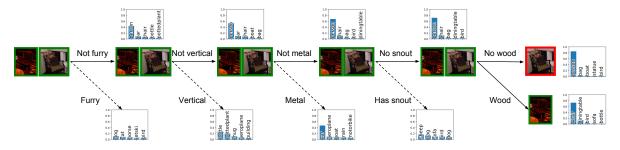

Figure 7: Qualitative Explanation of Classification of Two Chairs in APY. We show the binary decisions made for two images of a chair along with the current label prediction at each step. The upper path corresponds to when the attribute is not present for a given class. Both images follow the same path 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.

Table 5: Shared hyperparameters across datasets.

Hyperparameter	Value
Optimizer	Adam[17]
λ	0.25
Initial τ	5
Initial learning rate	0.001
Learning rate schedule factor	0.1
Batch size	100
Stopping threshold	0.95

Table 6: Varying hyperparameters per datasets.

Hyperparameter	AWA2	aPY	CUB	MNIST	CIFAR10	ImageNet
Epochs	150	200	500	50	50	200
Maximum binary decisions	30	20	32	4	6	34
Learning rate schedule step size	40	40	150	30	30	40
Learned attribute size for IOC	40	30	100	-	-	-

B. Code and Hyperparameters

Table 5 & 6 report all hyperparameters that are tuned for our XOC and IOC models along their final configuration. The hyperparameters are chosen based on the validation set performance.