Learning Decision Trees Recurrently Through Communication
-
Supplementary Material

Stephan Alaniz[1,2] Diego Marcos[3] Bernt Schiele[2] Zeynep Akata[1,2,4]
[1]University of Tübingen [2]MPI for Informatics [3]Wageningen University [4]MPI for Intelligent Systems

A. Ablating Loss and Maximum Steps T

When training our RDTC model with a cross-entropy loss for image classification, we found that training is more efficient and yields better results when applying the loss term at every step t in the communication loop up to the maximum step T.

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

One natural alternative to this approach is to apply the loss only at step T, the leaf node, essentially removing the sum Equation 1. In Figure 1, we show the difference of applying the loss at every time step (full loss) or only at the end (leaf loss) on CUB and AWA2. The final performance of the decision tree is the same, however, applying the loss at every time step produces a tree that has a better performance when evaluated at intermediate steps and results in a smaller tree after pruning, i.e., fewer tree nodes are used for the final tree to obtain best performance.

Moreover, we found that when hyperparameter T is chosen sufficiently high, we are able to reach this maximum performance while our tree distillation process ensures that the tree size does not increase past the point where the classifier achieves the highest accuracy. Figure 2 shows classification accuracy with increasing tree depth. Accuracy does not decrease past some value for T where the model performs best, and choosing any value bigger results in an equally explainable tree after pruning.

B. Decision Trees and Explanations of CUB and AWA2

Illustrating the decision making process helps the user get an explainable overview of the internal decision process of the whole classifier. We point to the tree branch into which a certain class (indicated by an example image from this class) falls along with the attribute associated with that branch. We inspect the learned structure of the decision tree by illustrating the splits from our aRDTC model

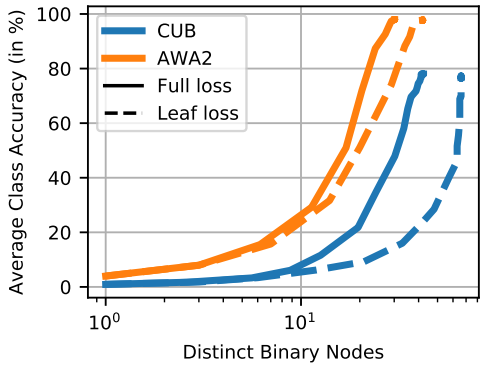

Figure 1: Accuracy and number of distinct nodes in RDTC on AWA2, CUB comparing our full loss at each time step (solid line) with a loss only applied at leaf nodes (dashed line). The full loss uses fewer nodes, i.e., a smaller tree, to achieve the same accuracy.

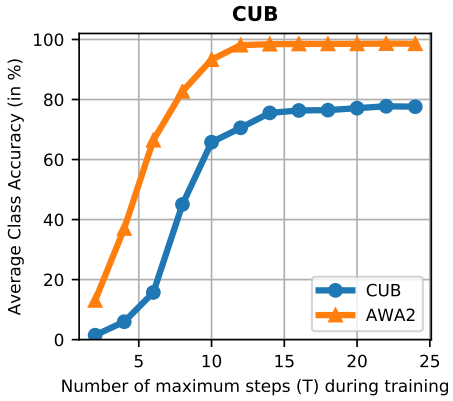

Figure 2: Training RDTC while varying hyperparameter T. As T increases, the model achieve a better accuracy up to a value of T where a plateau is reached. When increasing T further, the final tree size of RDTC does not increase due to pruning during tree distillation.

on CUB in Figure 4, and on AWA2 in Figure 6. Here, the left and right sub-tree indicates that the attribute is present or absent respectively. For instance, on CUB, the first decision deals with identifying bird with white underparts, separating these from birds with any other color. These categories get further refined with each binary split via a hierarchical clustering that reveals the decision tree structure of our aRDTC framework. These serve as additional examples of introspection, showing that our model allows to make a more informed decision about the trustworthiness of the network's prediction.

In Figure 5, we illustrate a qualitative example of the classification of two images of Scarlet Tanagers made by our model trained on CUB. Both images follow the same path for the first decisions, before diverging when it comes to the decision whether the bird has black wings. The top bird actually does not have black wings and, thus, is classified as a Summer Tanager, a bird species with the same appearance as Scarlet Tanager except for having red instead of black wings.

Equivalently in Figure 7, we illustrate a qualitative example of the classification of two images of tigers made by our model trained on AWA2. Again, both images follow the same decision until, for the white tiger, our model wrongly predicts "no stripes" and incorrectly classifies it as a lion. Together with the full decision trees, these explanations allow for detailed introspections into the global decision process our aRDTC model.

C. Explanations without Attributes

When working with datasets that do not provide annotated attributes, we can train our RDTC with $\lambda = 0$, which still exposes the decision tree structure. This allows introspection into the intermediate class splits of the model revealing a hierarchy that can reveal semantics. When applies on CIFAR-10, our RDTC model not only retains ResNet performance (93.1% vs. 93.3%), it also semantically clusters the data even though there is no attribute guidance. Figure 3 shows the resulting decision tree of RDTC on CIFAR-10. In the first binary split, we observe that RDTC separates the animal classes from the vehicles. Subsequently, vehicles are clustered into motor vehicles (car, truck) and the rest (airplane, ship). For animals, our model also finds reasonable clusters such as grouping cat and dog, as well as grouping horse and deer. ImageNet is a more challenging dataset, where we observe similar behaviour. In Figure 8, we show the decision tree of the first decisions on ImageNet with a randomly selected subset of classes, each represented by one representative image. Our model separates animals from inanimate objects in the first tree split following the data semantics. In the later decisions of the tree, there are clusters of dogs/cats, birds, monkeys on one side of the tree and clusters of furniture and electrical appliances on the other. These example show that, even when no additional attribute information is given, tree splits often follow semantics that are exposed by the decision tree learned by our RDTC.

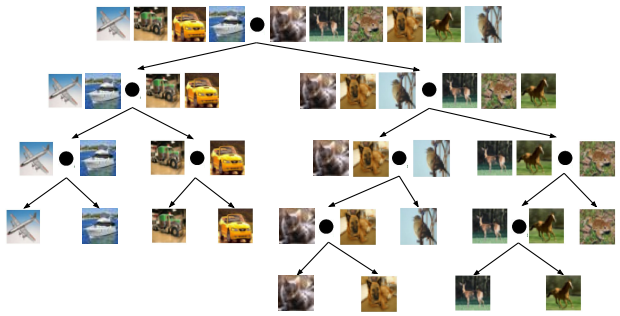

Figure 3: Our RDTC learns the decision tree on CIFAR10 without attribute data. While decision nodes do not have ground truth attributes, we can still interpret the decision, e.g., the first node separates animals from vehicles.

Algorithm 1 RDTC training

Input: Image x, label y
Max # of decisions T, Attribute data α
Output: Predicted label $\hat{y}$
Binary decision sequence $d^{(1)}, \dots, d^{(T)}$

$z = \text{CNN}(x)$
$\hat{\boldsymbol{a}} = \text{TempSoftmax}(f_{\text{AttrMLP}}(z))$
init $\mathcal{M}^0, h_0$
$\mathcal{L} = 0$
for $t = 1$ to T **do**
  $c^{(t)} = \text{GumbelSoftmax}(f_{\text{QuestMLP}}(h^{(t-1)}))$
  $d^{(t)} = \hat{a}_{c^{(t)}}$
  $\mathcal{M}^{(t)} = \mathcal{M}^{(t-1)} \oplus (c^{(t)}, d^{(t)})$
  $h^{(t)} = \text{LSTM}(h^{(t-1)}, \mathcal{M}^{(t)}, c^{(t)}, d^{(t)})$
  $\hat{y}^{(t)} = f_{\text{ClassMLP}}(\mathcal{M}^{(t)})$
  $\mathcal{L}^{(t)} = \frac{1}{T}\left[(1-\lambda)\mathcal{L}_{CE}(y, \hat{y}^{(t)}) + \lambda \mathcal{L}_{CE}(\alpha_{y,c^{(t)}}, \hat{a}_{c^{(t)}})\right]$
  $\mathcal{L} = \mathcal{L} + \mathcal{L}^{(t)}$
end for
gradient update with $\mathcal{L}$
return $\hat{y}^{(T)}; d^{(1)}, \dots, d^{(T)}$

D. RDTC Training Algorithm

For a concise representation of the RDTC training algotithm, we present a summary in Algorithm 1 including both components, RDT and AbL, iterative loss calculation and gradient updates using the terminology of the main paper.

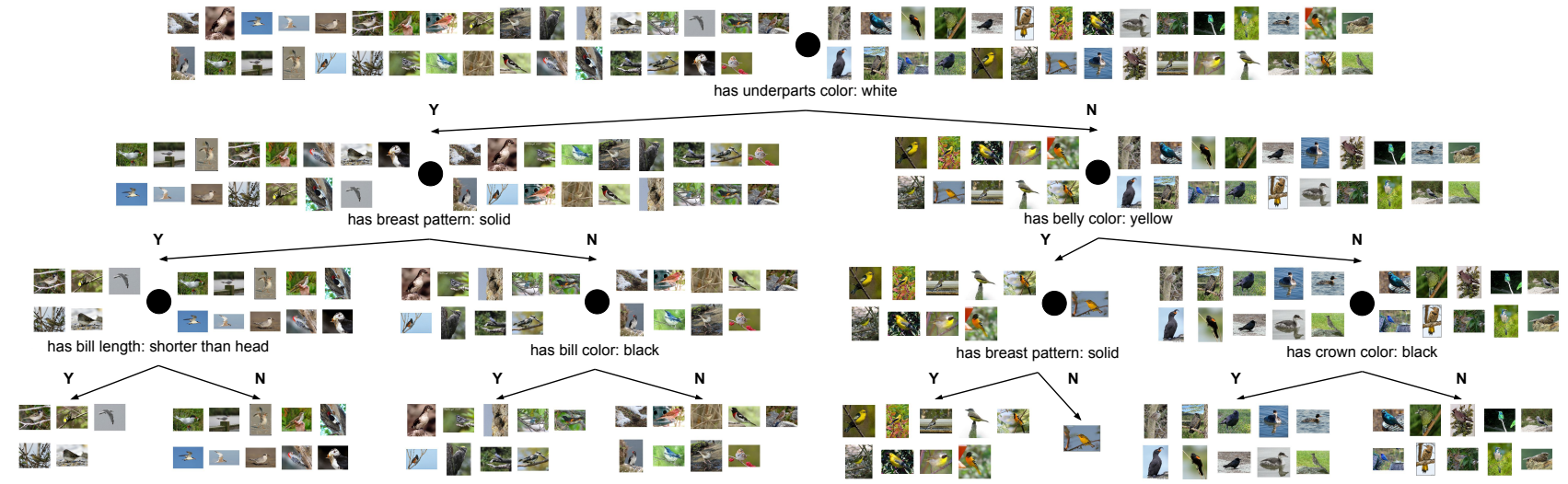

Figure 4: Our aRDTC learns explainable decisions via the decision tree showing the path for each class and it also gives each decision a human-understandable meaning. Here we show the first three decisions for a subset of the 200 classes of birds in CUB where a randomly selected image from a class represents each class.

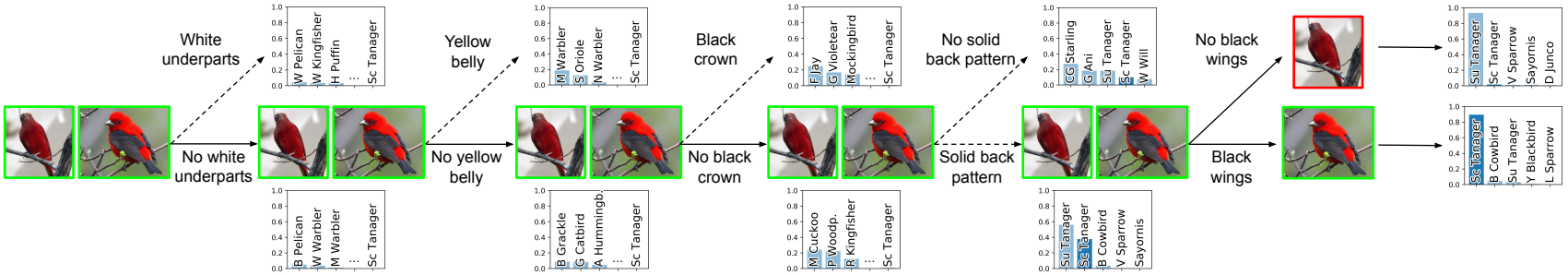

Figure 5: Our aRDTC points to the reasoning behind a wrong decision. Here we illustrate two images from the "Green Kingfisher" class. The lower path lead to a correct classification. Both images follow the same path except for the decision of "black wings". The flying bird gets classified as a "Belted Kingfisher" incorrectly because the black wings are not visible.

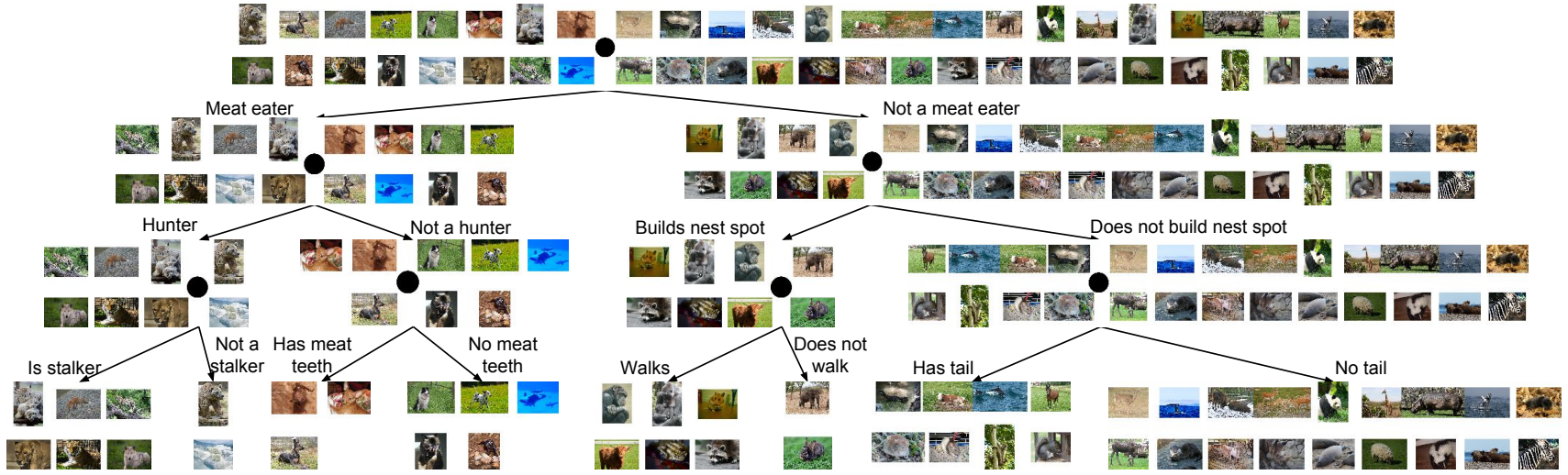

Figure 6: Learned explainable decisions on AWA2 by our aRDTC model. We show the decision tree of the most likely path for each class, *i.e.*, introspection, and give each decision a human-understandable meaning, *i.e.*, rationalization. The tree exposes the thought process of our model, *e.g.*, it decides to separate meat-eating animals from all other animals in the first step.

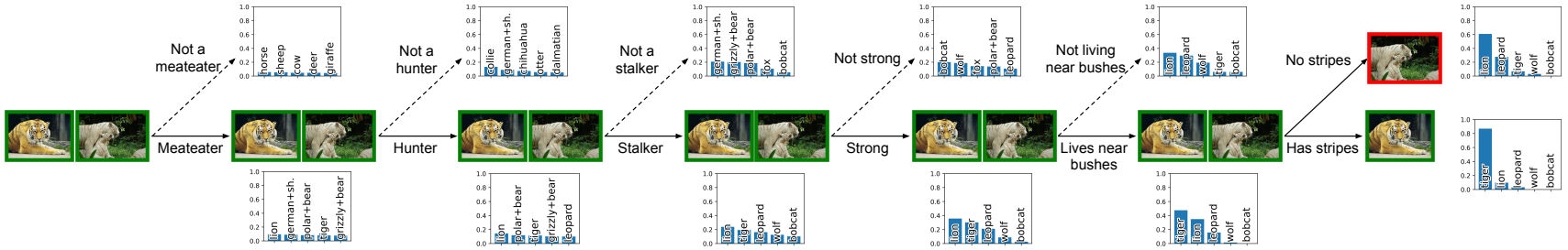

Figure 7: Decision process for two tiger images in AWA2 along with the current label prediction at each step. The lower (upper) path is taken when the attribute is present (absent) for a given class. Both images follow the same path except for the last decision, "has stripes". Since these are absent in the white tiger, it gets classified incorrectly as a lion.

Figure 8: Learned binary decisions on ImageNet by our RDTC model. A subset of randomly chosen classes are shown by one representative image of each class. Our tree reveals a clustering as decision splits narrow down towards a specific subset of classes.