

# Uncertainty in Recurrent Decision Tree Classifiers

Stefan Wezel

Explainable Machine Learning

October 23, 2020

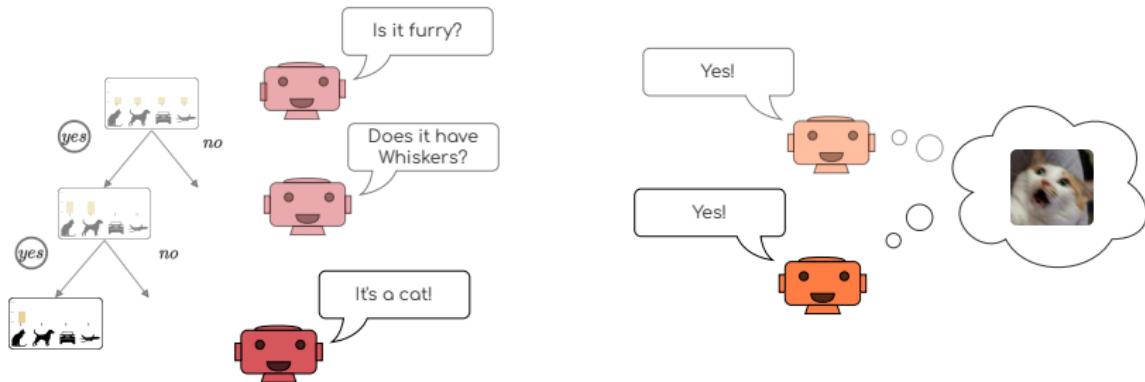
# What?

## Setting

- Many powerful architectures for image classification
- Prominent example: ResNet
- Popular models only yield classification
- No reasoning behind classification

# What?

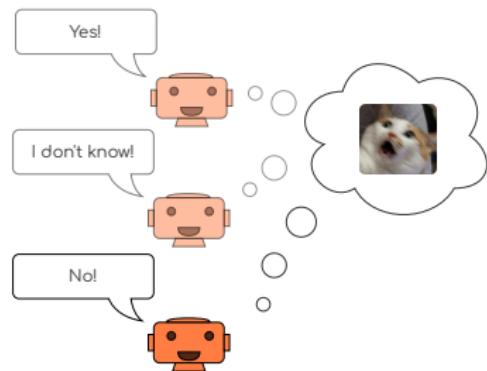
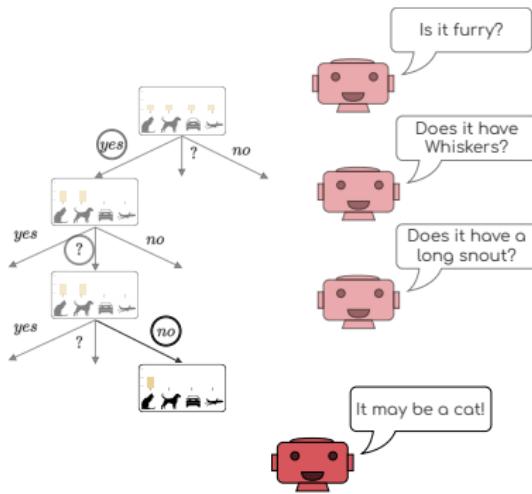
What is a Recurrent Decision Tree Classifier?



- Alaniz and Akata [1] propose RDTC
- Two communicating agents
- Left: ask questions — right: look at data and answer them
- Unfolding tree reveals reasoning behind classification

# What?

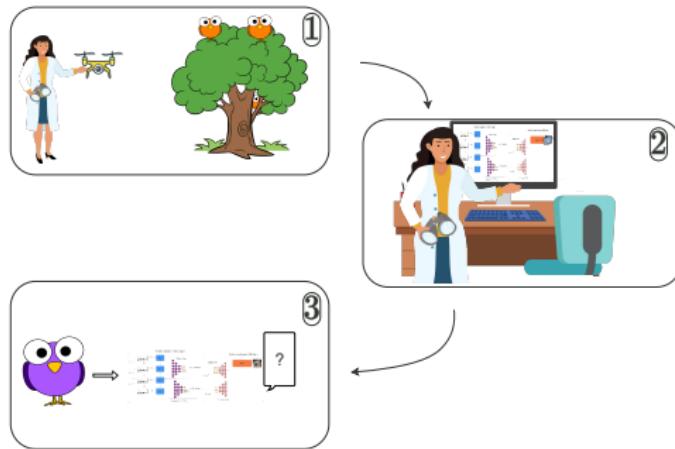
## Introducing Uncertainty to a RDTC



- Right agent is aware of uncertainties
- Communicates this to left agent

# Why do we need uncertainty?

A Practical Example...

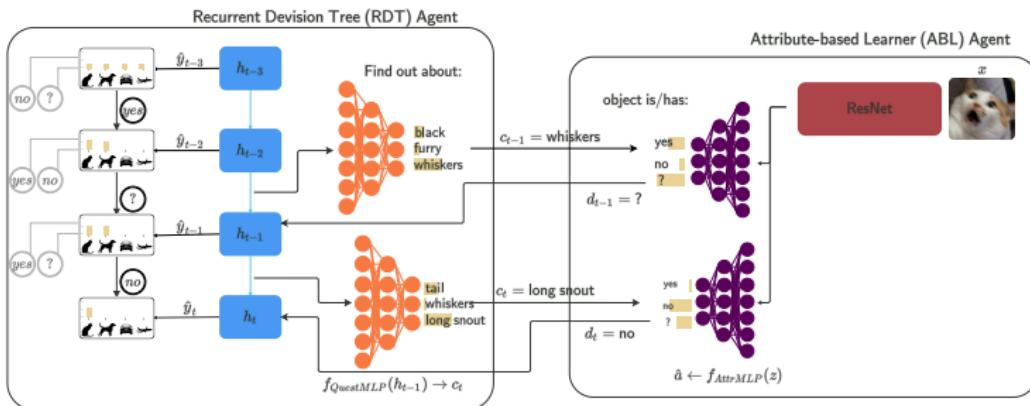


- Ornithologist surveys area using drone and CV software
- Classification is automated with our model
- Bird species unknown to model yield high uncertainty
- Those can be classified manually



# How?

## Architecture



- RDT can not see the images and only ask if attribute is present
- The AbL can see the image and answer RDT's questions

# Attribute-based Learner

## Answering questions

- Extract features from image using ResNet
- MLP maps features to 'yes-no' answers
- those indicate absence/presence of attributes
- Discrete answer from AbL though applying

$$\text{TempSoftmax}(\log \pi) = \frac{\exp((\log \pi_i)/\tau)}{\sum_{j=1}^K \exp((\log \pi_j)/\tau)} = d_t \text{ on } \log \pi$$

which are logit values for either 'Yes', or 'No' per attribute.

- TempSoftmax as differentiable approximation to argmax

# Recurrent Decision Tree

## Building a decision tree

- LSTM
  - Hidden state based on previous hidden states and new answers
- Explicit Memory
  - Stores all questions and corresponding answers
  - Its content is the decision tree
- $f_{ClassMLP}$ 
  - Make classification based on LSTM's hidden state
- $f_{QuestMLP}$ 
  - Find next question to ask based on LSTM's hidden state
  - Next question is the index the  $f_{QuestMLP}$  poses
  - To turn logits into a discrete value, Gumbel softmax is used
  - This allows us to sample an index from the logits
  -

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

# Training

## Joint Objective

- We optimize for class (and attribute accuracy)

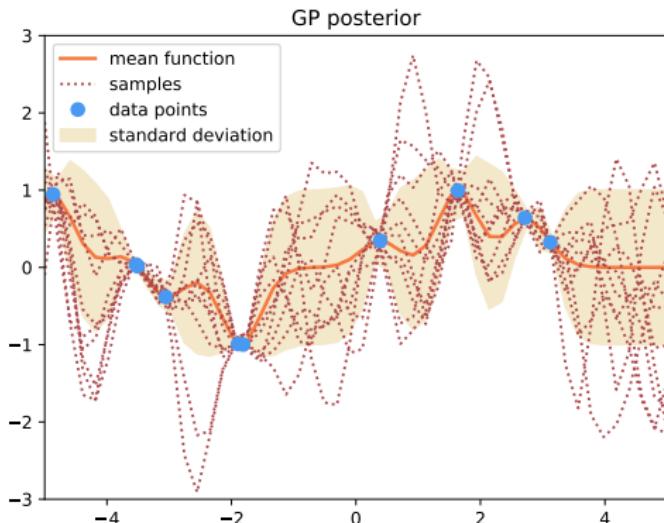
$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^T [(1 - \lambda) \mathcal{L}_{CE}(y, \hat{y}_t) + \lambda \mathcal{L}_{CE}(\alpha_{y,c_t}, \hat{\alpha}_{c_t})]$$

- $\lambda$  can be used to balance the two loss terms
- For all of our experiments, we use  $\lambda = 0.2$



# Background

## A small excursion to Gaussian Processes (GP)



- Data can be described by (infinitely) many functions
- A GP is a PDF over these functions
- Intuition: → GP yields probability for function values
- Parameterized by mean function and covariance function
- Variance corresponds to model uncertainty

# Background

## Dropout Uncertainty Estimation

- Gal and Ghahramani [2] proof that variance from dropout corresponds to model uncertainty
- We use their proof as foundation for our uncertainty estimate
- Neural net is set of weighted linear functions, activated by non-linearity
- Putting PDF over each weight creates finite GP

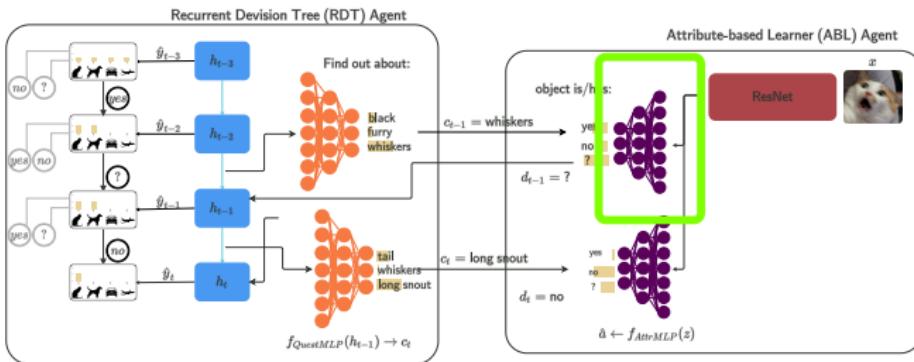
# Background

## Dropout Uncertainty Estimation

- We can view neural net as approximation to GP
- Posterior over functions requires computing integrals
- Intractable integrals require methods of variational inference
  - GP objective → minimization objective
  - For covariance function → Monte-Carlo integration
- Variational GP's objective can be rewritten as dropout net
- Variance arising from dropout can be interpreted as model uncertainty

# How?

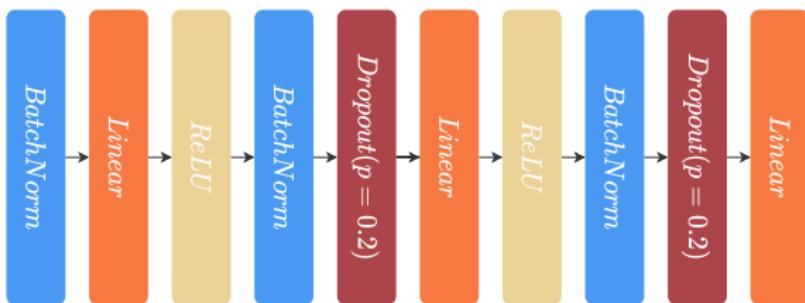
How do we get uncertainty information?



- We use Gal and Ghahramani's [2] proof to estimate uncertainty
- We want AbL to say '?' in the case of high uncertainty
- Make  $f_{AttrMLP}$  a dropout MLP
- After extracting features → compute  $Var(n \text{ forward passes})$
- According to proof, this corresponds to model uncertainty

# Getting Uncertainty Information

## Estimating Uncertainty in the AbL



- We include dropout layers in our  $f_{AttrMLP}$
- We tested different configurations and a combination of batchnorm and dropout worked best

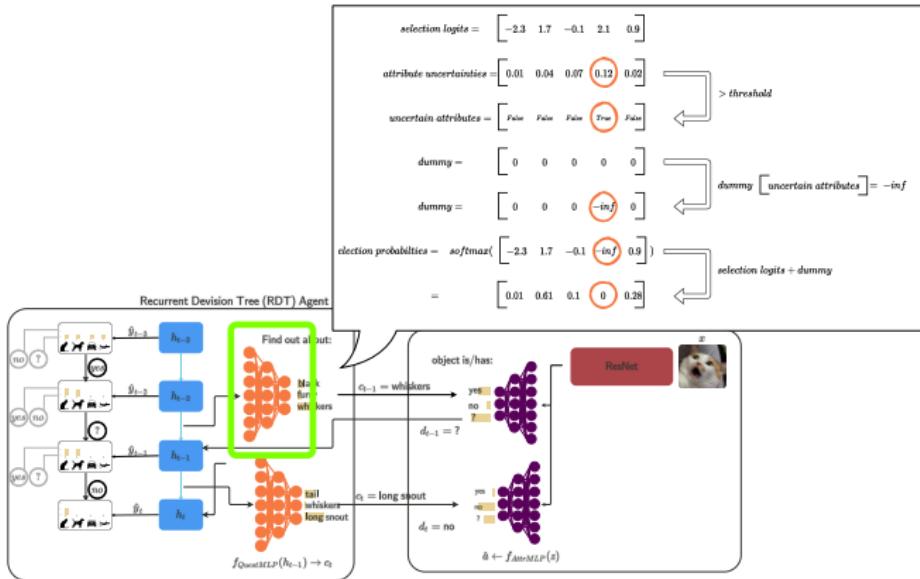
# Using Uncertainty Information

## Uncertainty Information as Inductive Bias

- Now, we have an uncertainty estimate
- But now, we need a way to use it
- we use two different strategies
  - We prevent the model from asking questions regarding uncertain attributes → remRDTC
  - We give the model the ability to answer with 'I don't know' as extended vocabulary → extRDTC
- For both strategies, we need to make sure the model does not use any gradients coming from uncertain attributes
- This ensures that the uncertainty information only serves as an inductive bias and the model does not misuse it

# Using Uncertainty Information

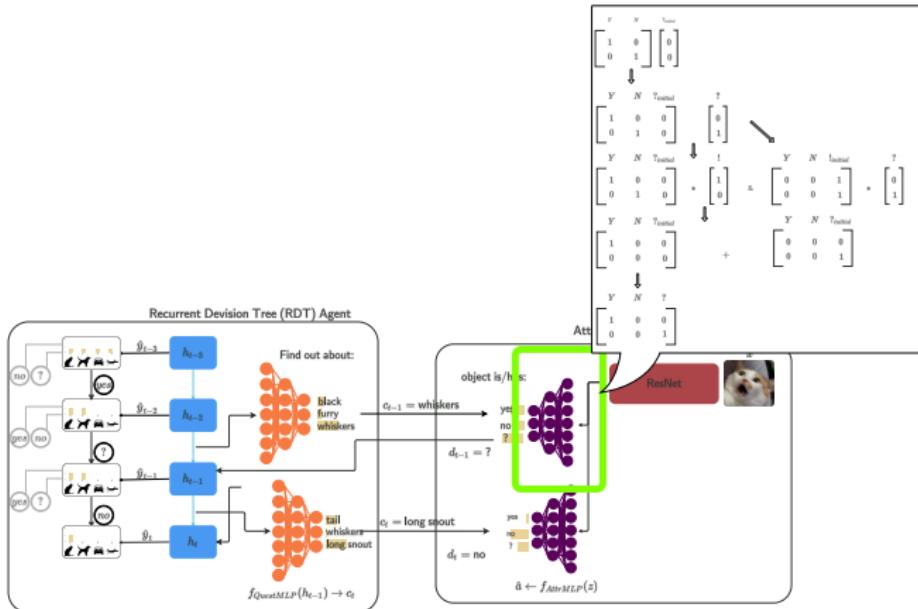
## Removing uncertain attributes (remRDTc)



- The output from  $f_{QuestMLP}$  is an index that indicates the attribute in question
- In case, an attribute is deemed uncertain by the AbL, we replace selection logits at those indices with  $-\infty$
- This prevents the Gumbel softmax from picking uncertain

# Using Uncertainty Information

Extending the vocabulary (extRDTc)



- We create a binary uncertainty vector with 1s where uncertainty is above a given threshold
- This is appended to the initial answer and can be used by the RDT



# Experiments

- Now, we can make our model aware of, and express its uncertainties
- We use this in our experiments to:
  - Investigate uncertainty and its relationship to other variables
  - Test our model on OOD data
  - Test the model's performance on benchmark datasets

# Experiments

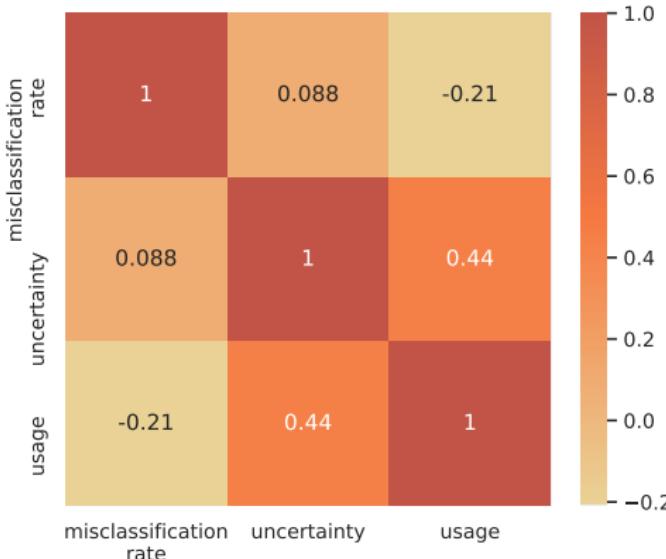
## Datasets

- Animals with Attributes 2 (AWA2)
  - medium size, coarse grained, image classification
- aPY
  - small size, coarse grained, image classification
- CUB
  - large size, fine grained, image classification

# Experiments

## Investigating Uncertainties

- We investigate covariance of misclassification rate, uncertainty, and usage of attributes in CUB

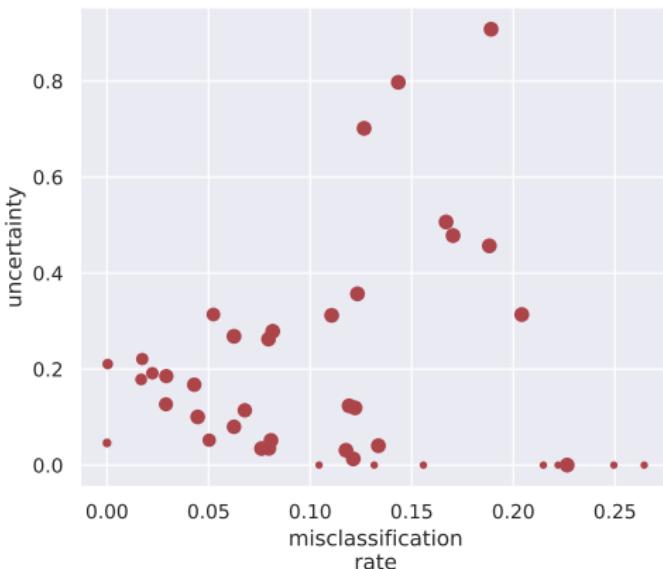


- Negative correlation between usage and misclassification rate
- Almost no correlation between uncertainty and

# Experiments

## Investigating Uncertainties

- Let's have a look at the actual values
- Uncertainty and misclassification rate (usage is represented through size)

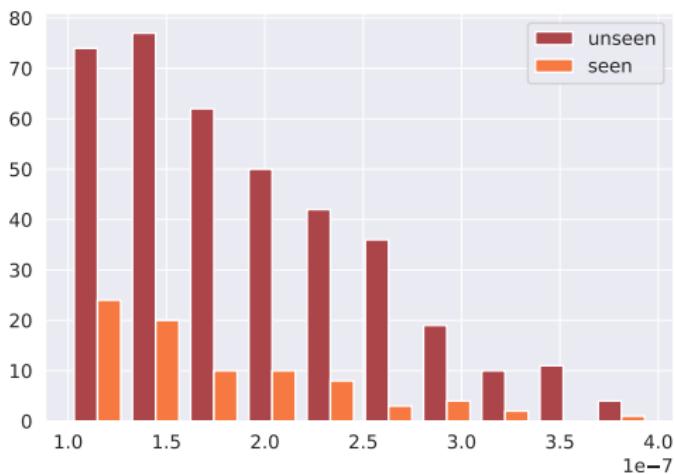


- Uncertain attribute seem to be misclassified more often

# Experiments

## OOD Detection

- We test extRDTc in a zero shot setting (using CUB)
- A quarter of all classes were never seen during training
- We compare uncertainty values of seen and unseen classes



- Unseen classes have higher uncertainty values

# Experiments

## Comparison to other models

- Decision Tree (DT)
  - Use extracted ResNet features
  - Try to split until every leaf node corresponds to one class
- Explainable Decision Tree (XDT)
  - Same as DT, but using learned attribute representations instead of features
  - Attribute representations are created using  $f_{AttrMLP}$  as head for ResNet
- dNDF [4]
  - Every node in the tree is a parametric differentiable function
  - Every route through the tree leads to a leaf node representing a class distribution
  - Objective is to learn the optimal route through the tree for each example
- aRDTC
  - RDTC with  $\lambda > 0$
- ResNet
  - Not explainable
  - Trained on ImageNet and then fine-tuned for specific datasets

# Experiments

## Results on Benchmark Datasets

	AWA2	aPY	CUB
ResNet [3]	98.2± 0.0	85.1± 0.6	79.0± 0.2
DT	78.0± 0.4	64.3± 0.6	19.3± 0.3
dNDF[4]	97.6± 0.2	85.0± 0.6	73.8± 0.3
RDTC[1]	98.0± 0.1	85.7± 0.7	78.1± 0.2
XDT	73.9± 0.9	59.9± 1.5	4.9± 1.3
aRDTC[1]	98.6	86.1	77.9± 0.6
remRDTC(ours)	<b>98.7</b>	<b>86.4</b>	77.7
extRDTC(ours)	<b>98.7</b>	85.4	77.8

# Experiments

## Results on Benchmark Datasets

	aRDTC [1]	Random Baseline	remRDTC	extRDTC
<b>AWA2</b>				
Class	<b>98.6</b>	98.5	98.7	98.7
Attribute	80.4	84.6	<b>87.5</b>	82.31
<b>aPY</b>				
Class	86.1	<b>86.5</b>	86.4	85.4
Attribute	86.4	86.2	<b>87.6</b>	87.1
<b>CUB</b>				
Class	<b>77.9</b>	76.8	77.7	77.8
Attribute	68.6	70.0	77.4	<b>82.6</b>



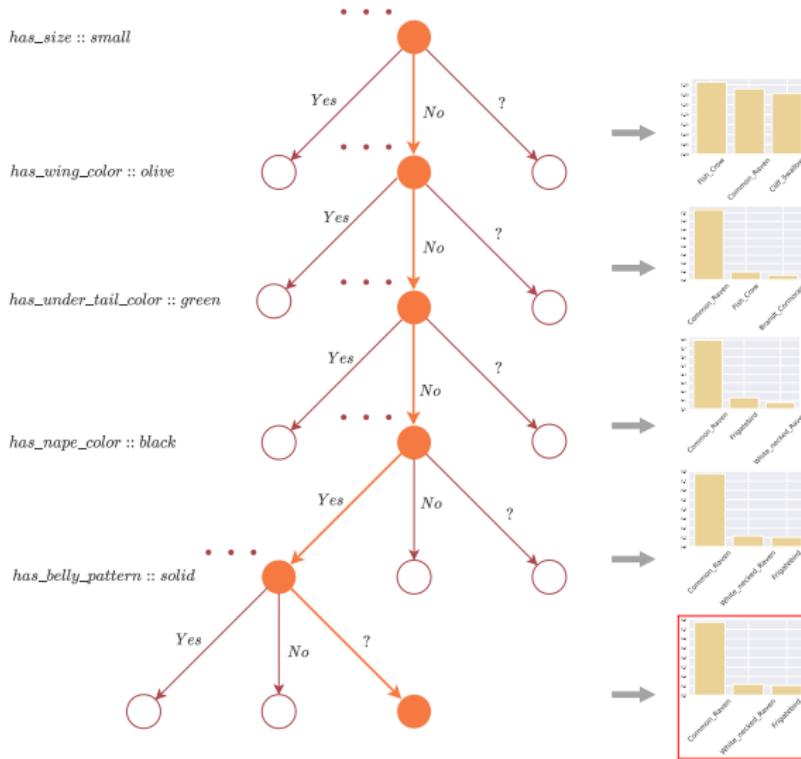
# Discussion

Looking back...

- The right kind of uncertainty?
  - We only consider model uncertainty (epistemic uncertainty)
  - Uncertainty arising from noise or occlusions in the data is not considered
- Other methods besides dropout uncertainty estimation
  - Unfortunately, they are often computationally expensive
  - Uncertainty estimation in neural networks is challenging in itself
- Beyond attribute uncertainty
  - Estimate uncertainty in other parts of the model (i.e.  $f_{ClassMLP}$ )

# Conclusions

## A qualitative Example





## References

- [1] S. Alaniz and Z. Akata. Explainable observer-classifier for explainable binary decisions. arXiv preprint arXiv:1902.01780, 2019.
- [2] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In international conference on machine learning, pages 1050–1059, 2016.
- [3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
- [4] P. Kotschieder, M. Fiterau, A. Criminisi, and S. Rota Bulo. Deep neural decision forests. In Proceedings of the IEEE international conference on computer vision, pages 1467–1475, 2015.