Uncertainty in Recurrent Decision Tree Classifiers

Stefan Wezel

Explainable Machine Learning

October 28, 2020

What?

Setting

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

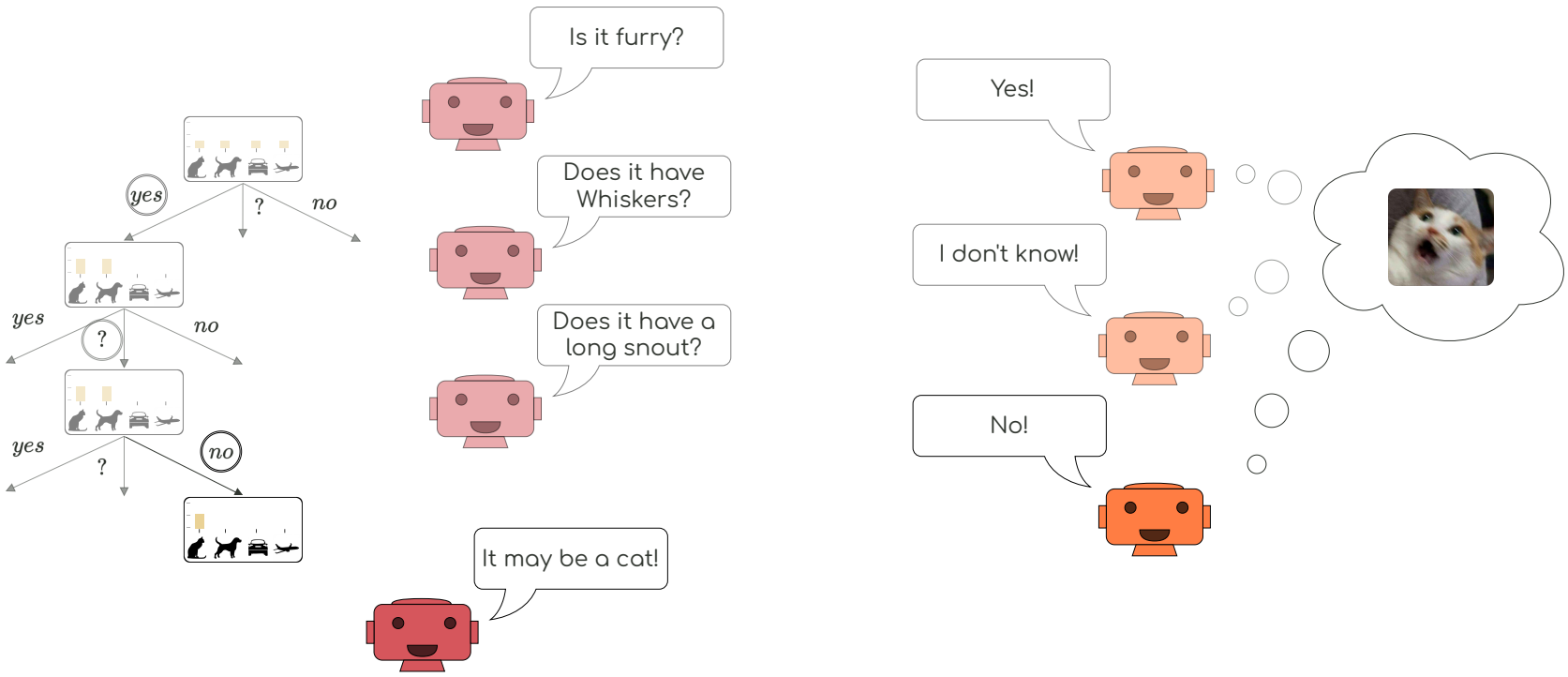
What?

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?

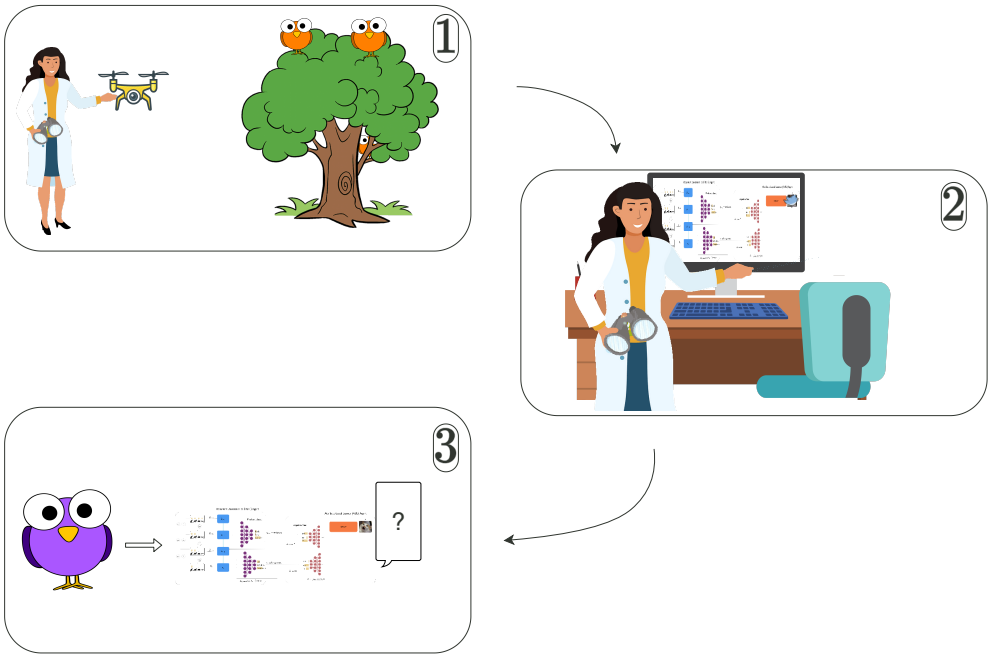
Introducing Uncertainty to a RDTC



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

Why do we need uncertainty?

A Practical Example...



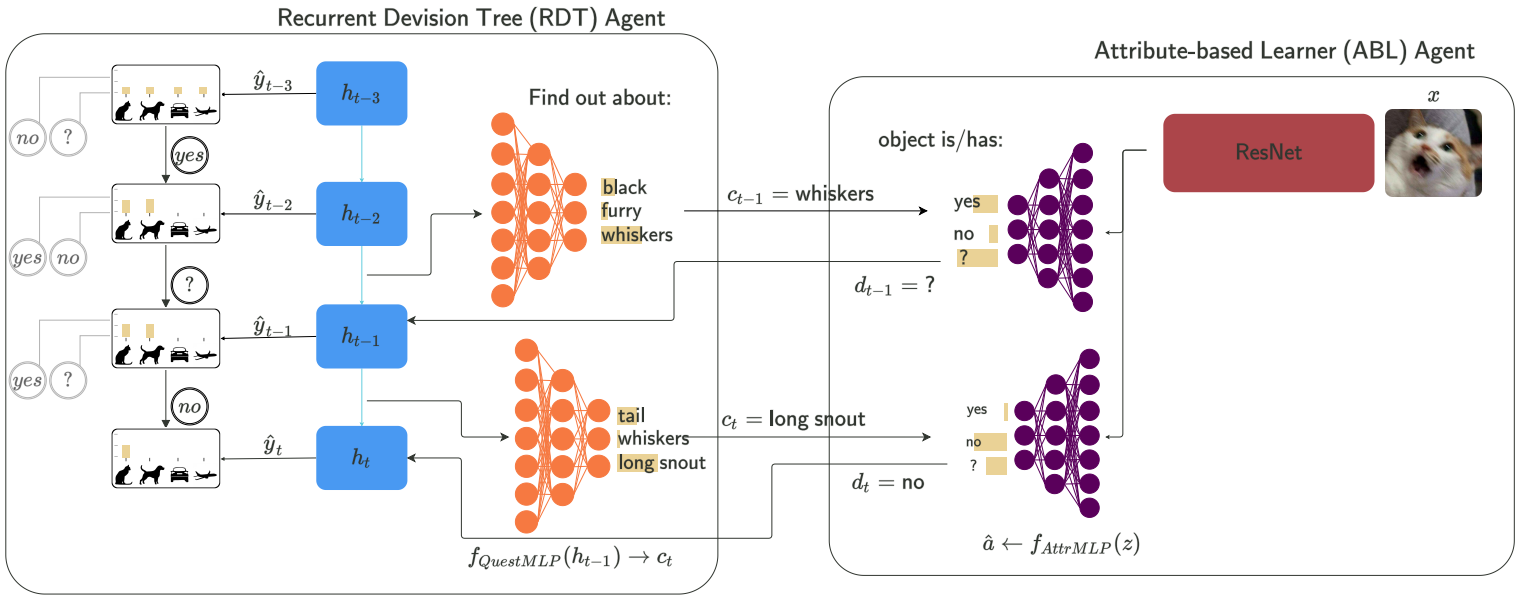
- Ornithologist surveys area using drone and CV software
- Classification is automated with our model
- Bird species unknown to model yield high uncertainty
- Those can be classified manually

Introduction

- RDTC by Alaniz and Akata [1] yields decision tree
- Decision tree is build through communication of agents
- Our aim is to introduce uncertainty to RDTC
- Uncertainty allows model to i.e. detect OOD examples

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

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

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

- TempSoftmax as differentiable approximation to argmax

Recurrent Decision Tree

Building a decision tree

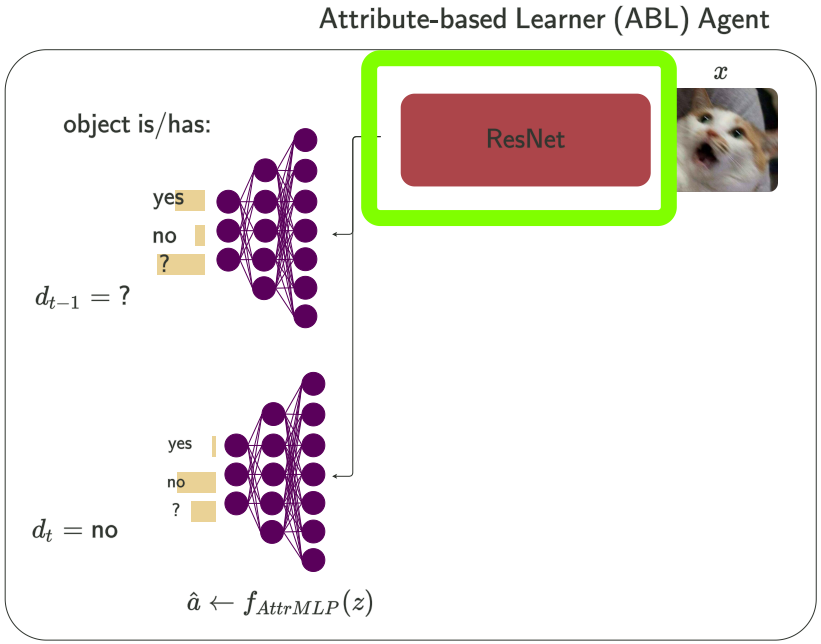
- LSTM
  - Hidden state based on previous hidden states and new answers
- Explicit Memory
  - Stores all questions and corresponding answers
  - Its content is the decision tree
- $f_{ClassMLP}$
  - Make classification based on LSTM's hidden state
- $f_{QuestMLP}$
  - Find next question to ask based 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
  -

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

Attribute-based Learner

Extracting features

- Extract features using ResNet
- Then pass to $f_{AttrMLP}$



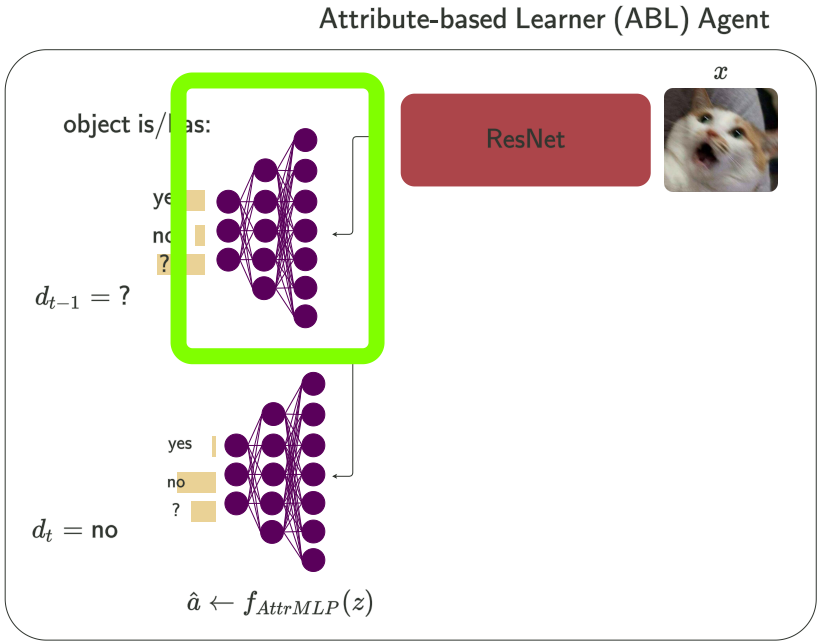
Attribute-based Learner

Mapping features to attributes

- Map features to answers
- Yes, No, ? for each attribute
- Discrete answers with TempSoftmax

$$\frac{exp((log\ \pi_i)/\tau)}{\sum_{j=1}^{K} exp((log\ \pi_j)/\tau)}$$

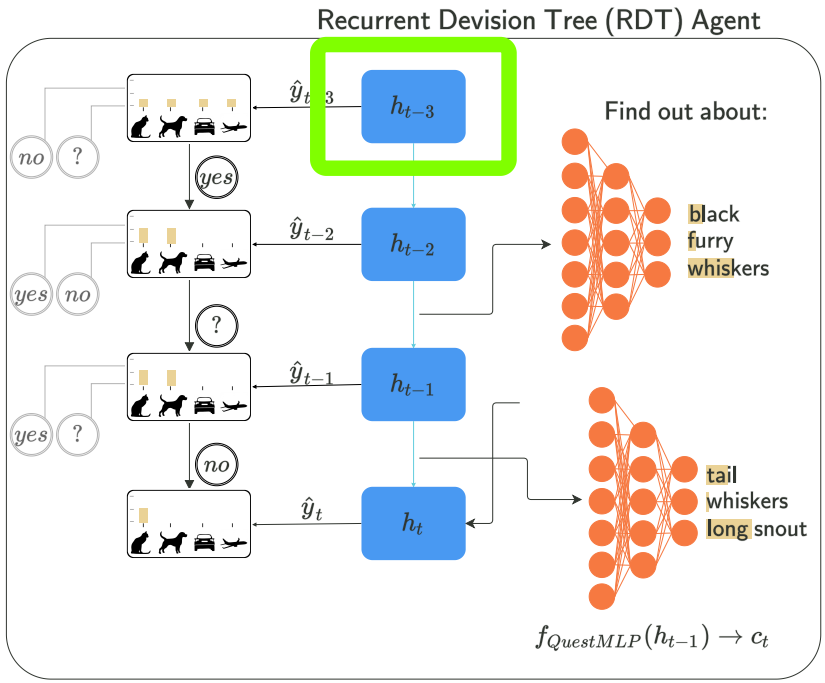
$$= \hat{a}$$



Recurrent Decision Tree

LSTM

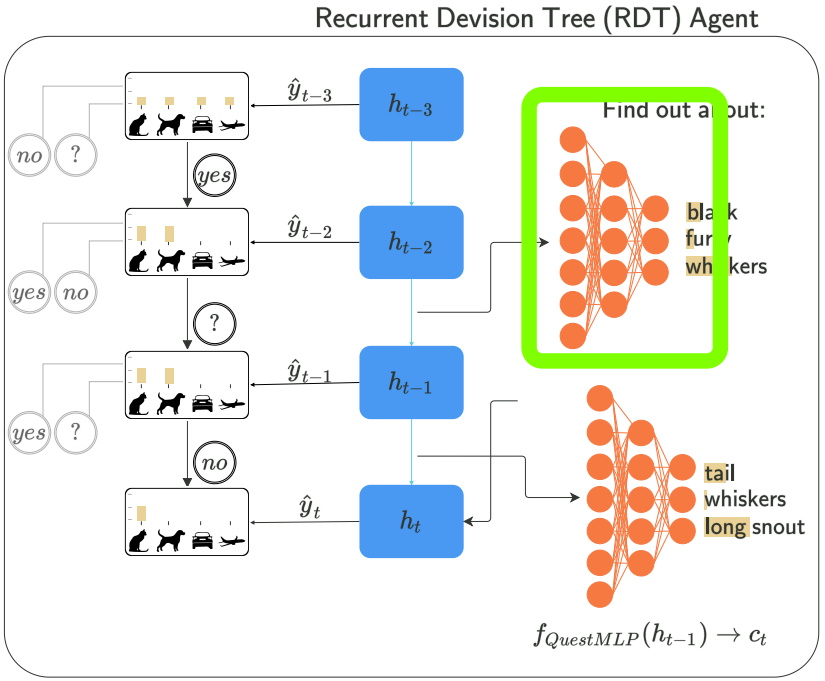
- Hidden state based on answers
- and explicit memory
- Basis for next question
- Basis for classification



Recurrent Decision Tree

Choosing questions

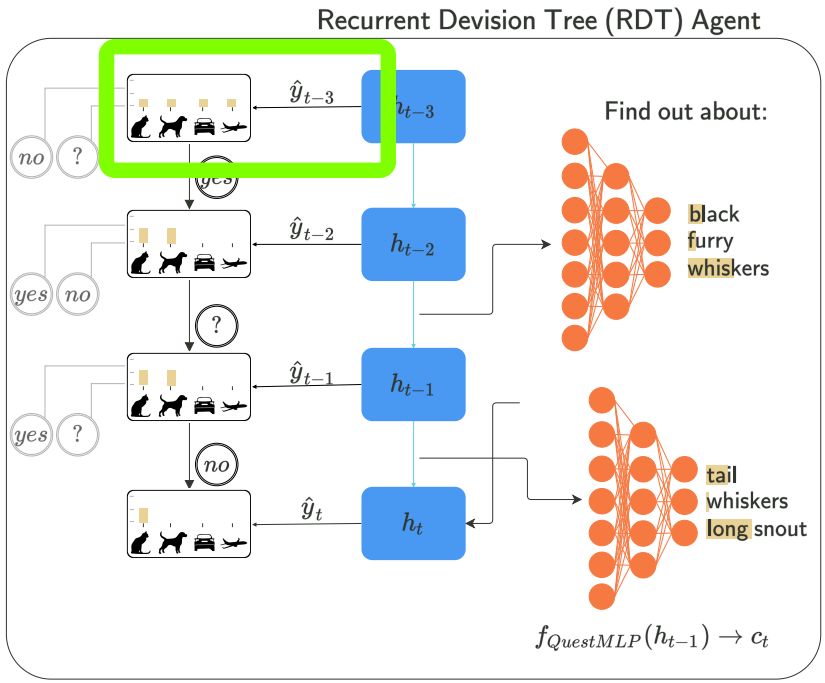
- Pose new question
- based on LSTM
- $c_t \sim p(c_t)$
- GumbelSoftmax
- Sample from categorical distribution
- $d_t = \hat{a}\,[c_t]$



Recurrent Decision Tree

Making a classification

- Classification in each communication step
- For classification loss



- Optimize for class (and attribute accuracy)

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

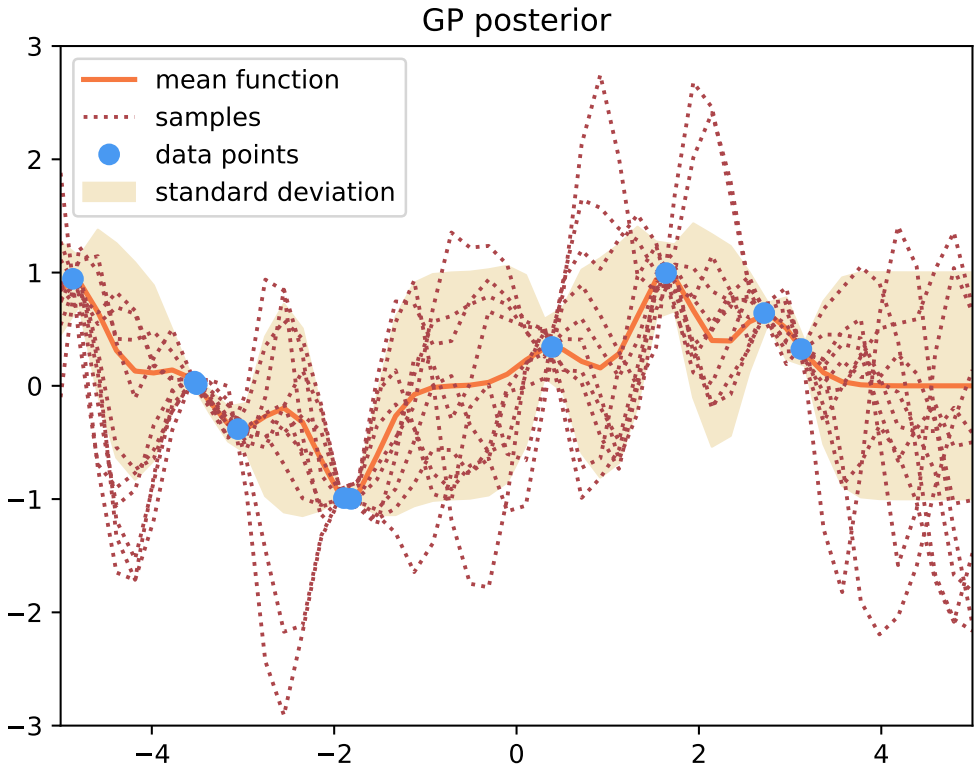
- λ can be used to balance the two loss terms
- For all of our experiments, we use $\lambda = 0.2$
- Discourage deep trees

The RDTC Model [1]

- AbL can see data, and answer all questions in advance
- RDT can choose attributes to question
- Get answer through indexing
- Joint objective of maximizing class and attribute accuracy

Background

A small excursion to Gaussian Processes (GP)



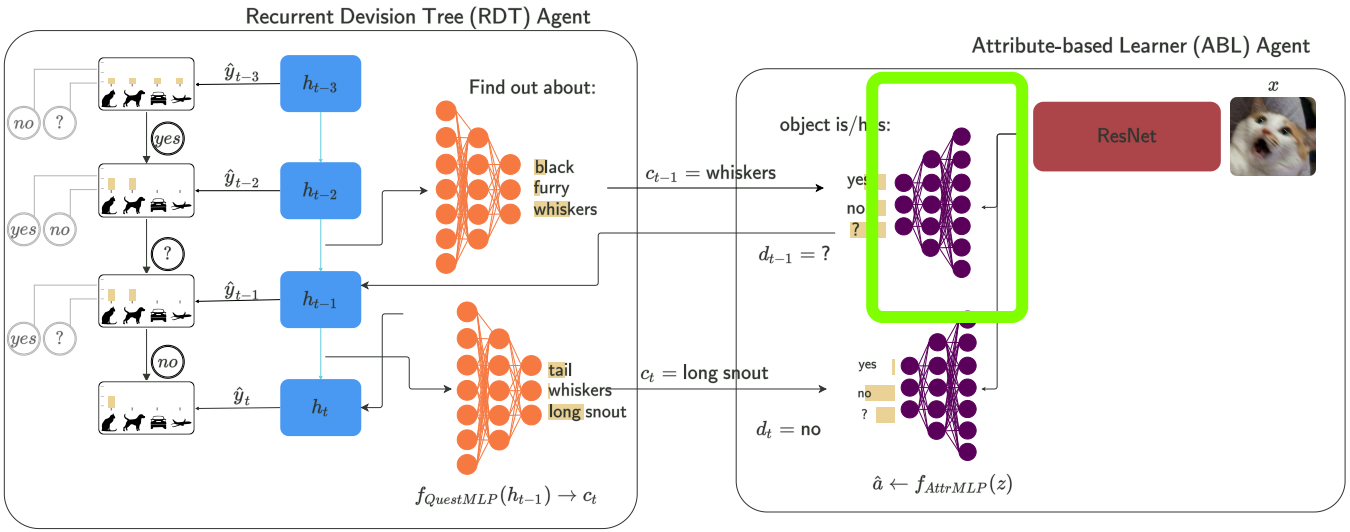
- Data can be described by (infinitely) many functions
- A GP is a PDF over these functions
- Intuition: → GP yields probability for function values
- Parameterized by mean function and covariance function
- Variance corresponds to model uncertainty

- 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

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

How?

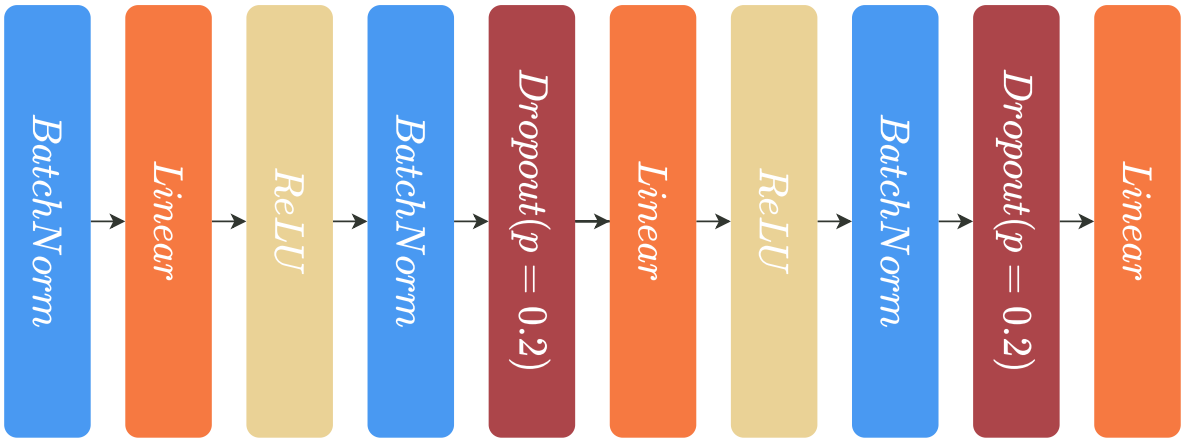
How do we get uncertainty information?



- 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 $\rightarrow$ compute $Var(n\ forward\ passes)$
- According to proof, this corresponds to model uncertainty

Getting Uncertainty Information

Estimating Uncertainty in the AbL

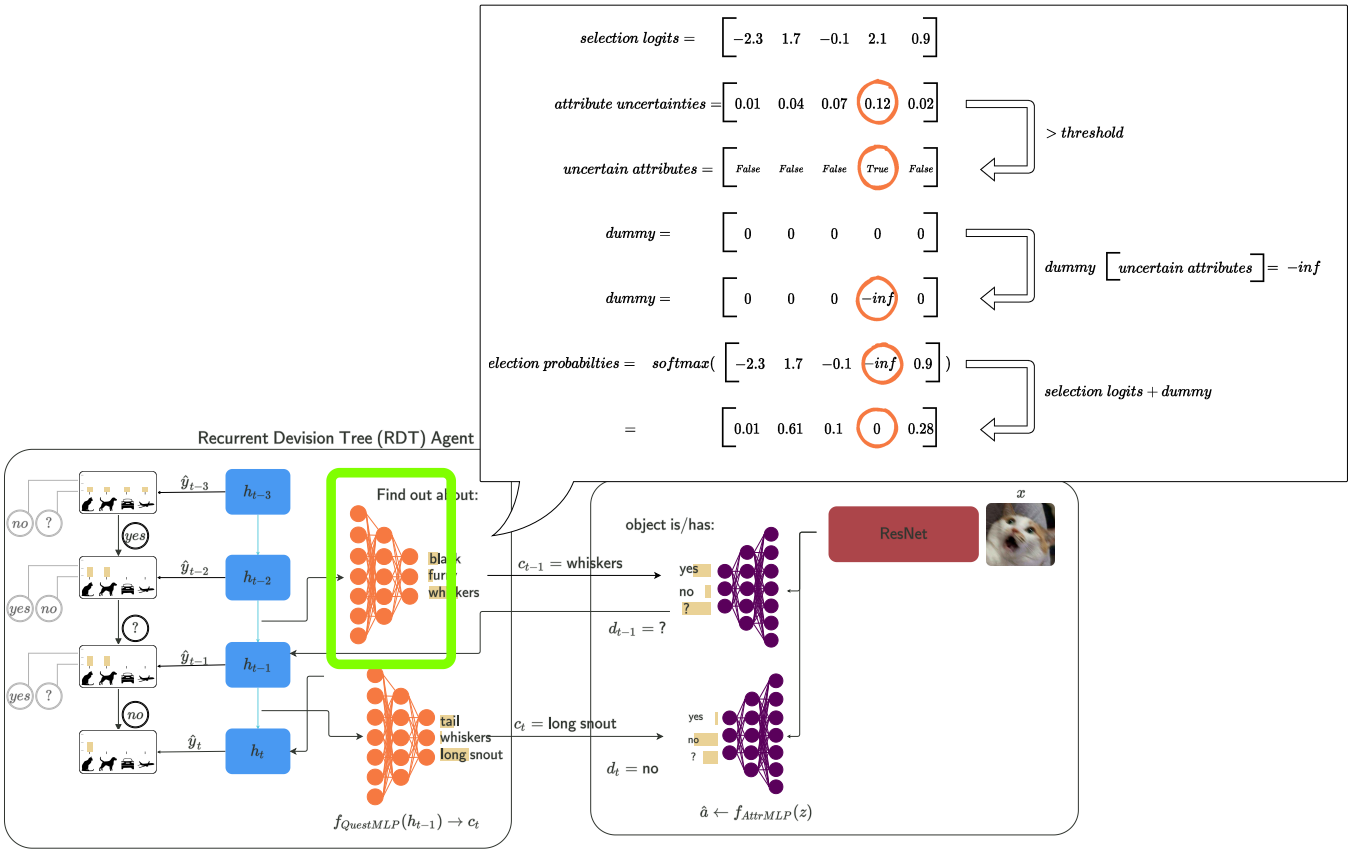


- We include dropout layers in $f_{AttrMLP}$
- We tested different configurations
- Combination of batchnorm and dropout worked best

- We use uncertainty in two different strategies
  - Prevent model from asking about uncertain attributes
    → remRDTC
  - We give the model the ability to answer with 'I don't know'
    → extRDTC
- Don't allow the model to use gradients from uncertain attributes
- Uncertainty information remains inductive bias

Using Uncertainty Information

Removing uncertain attributes (remRDTC)

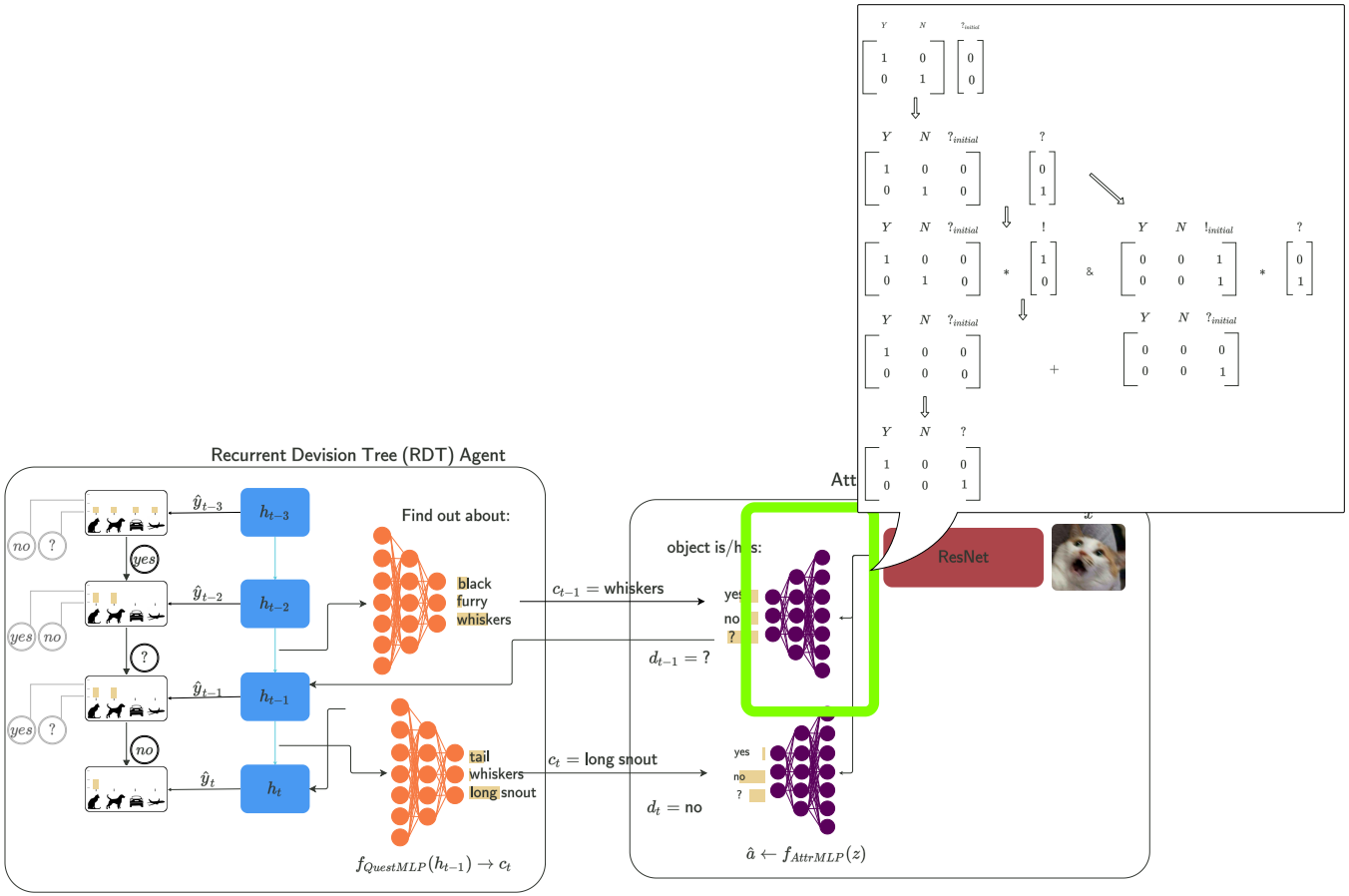


- Reminder: output from $f_{QuestMLP}$ is attribute index
- If attribute is deemed uncertain by AbL, replace selection logits at those indices with $-\infty$

  $\rightarrow$ Gumbel softmax cannot pick these attributes

Using Uncertainty Information

Extending the vocabulary (extRDTC)



- Binary vector with 1s where uncertainty is above threshold
- Append to initial answer
- Prevent conflicting answers

Introducing Uncertainty

- Gal and Ghahramani's [2] proof allows dropout uncertainty estimation
- We use their proof to introduce uncertainty to RDTC by Alaniz and Akata [1]
- Uncertainty information is used in two strategies
  - remRDTC
  - extRDTC

Experiments

- RDTC is now aware of, and can 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

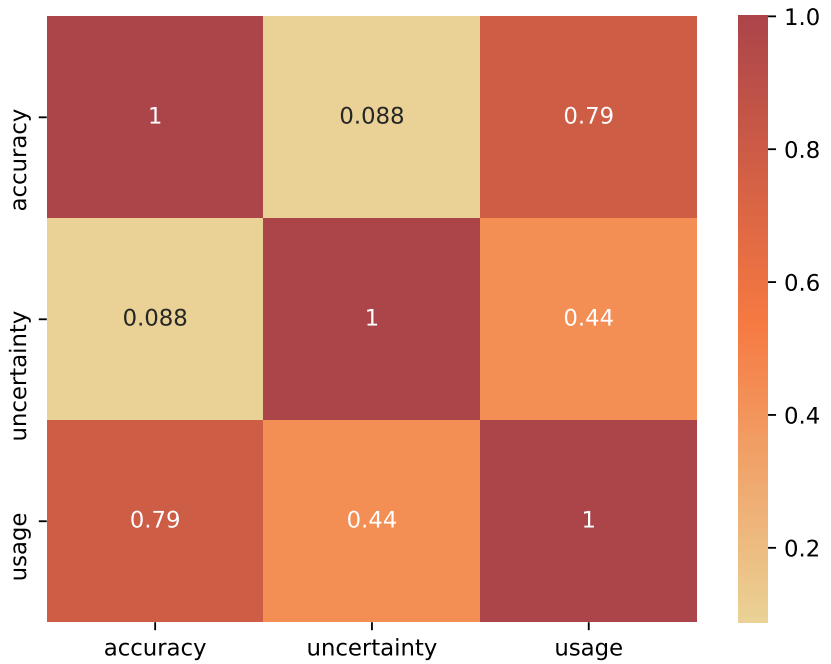
Datasets

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

Experiments

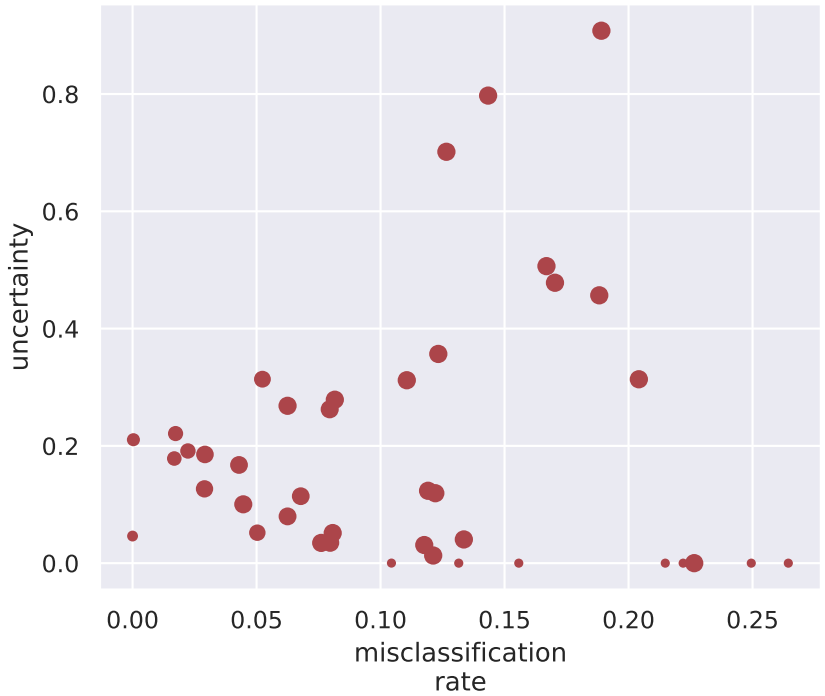
Investigating Uncertainties in CUB

- Misclassification rate, uncertainty, and usage of attributes



- Positive correlation between usage and accuracy
- Almost no correlation between uncertainty and accuracy

- Uncertainty, misclassification rate, usage (size $\propto$ usage)

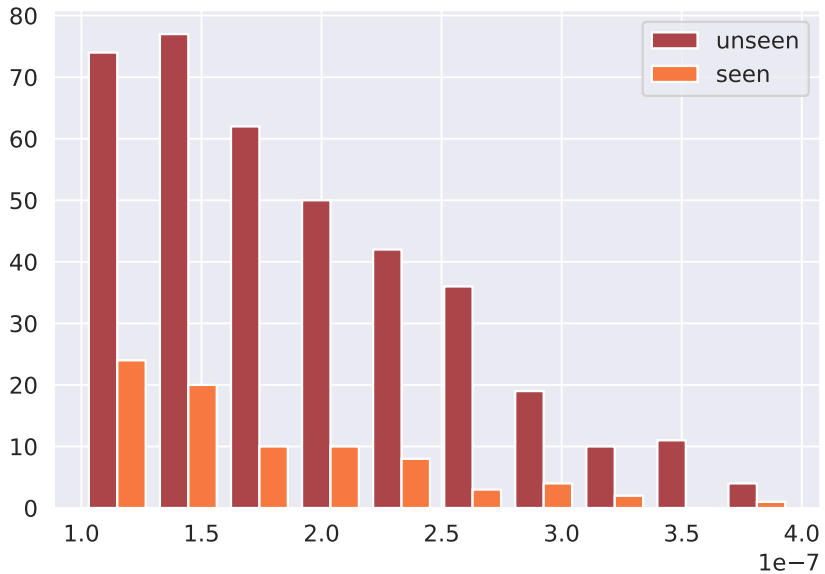


- Uncertain attribute seem to be misclassified more often

Experiments

OOD Detection

- We test extRDTC in zero shot setting (using CUB)
- $\frac{1}{4}$ of classes not seen in training
- Histogram of 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)	**98.7**	**86.4**	77.7
extRDTC(ours)	**98.7**	85.4	77.8

Experiments

Results on Benchmark Datasets

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

Experiments

- RDTC prefers attributes that it rarely misclassifies
- Uncertain attributes seem to be misclassified more often
- Unseen classes yield higher uncertainty
- Both remRDTC and extRDTC perform well on benchmark tasks

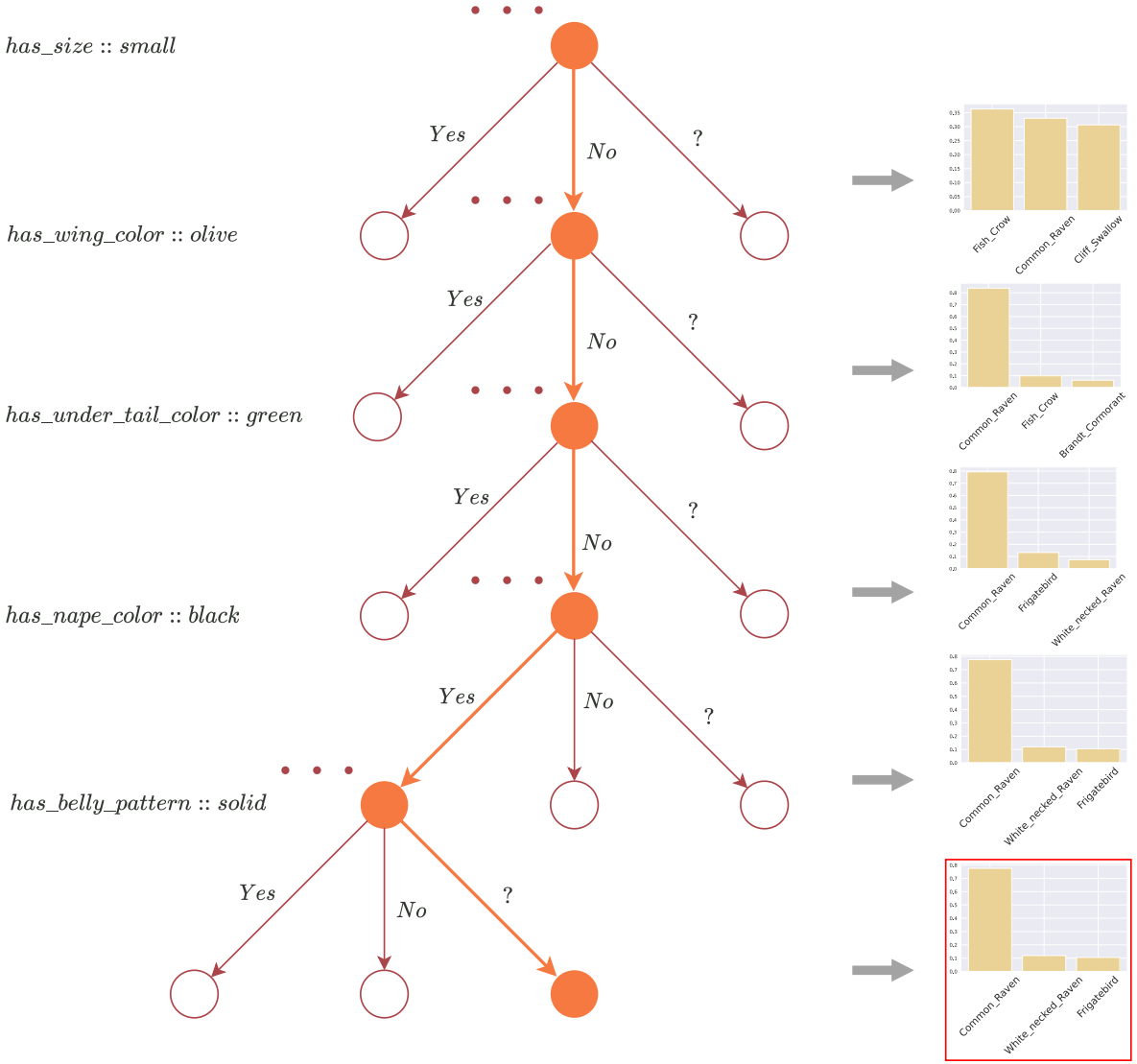
Discussion

Looking back...

- The right kind of uncertainty?
  - Uncertainty arising from noise or occlusions is not considered
  - Dropout Uncertainty Estimation lacks empirical proof
- Other methods of uncertainty estimation
  - Often computationally expensive
  - Open area of research
- Beyond attribute uncertainty
  - Estimate uncertainty in other parts of the model (i.e. $f_{ClassMLP}$)

Conclusions

A qualitative Example



Conclusions

- Based on proof by Gal and Ghahramani [2] estimate uncertainty
- Uncertainty is used in RDTC by Alaniz and Akata [1]
- Two strategies: remRDTC and extRDTC
- We investigate uncertainty and its relationship to other variables
- We show that OOD examples yield high uncertainty
- Uncertain RDTC's achieve state of the art accuracy

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. Kontschieder, 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.