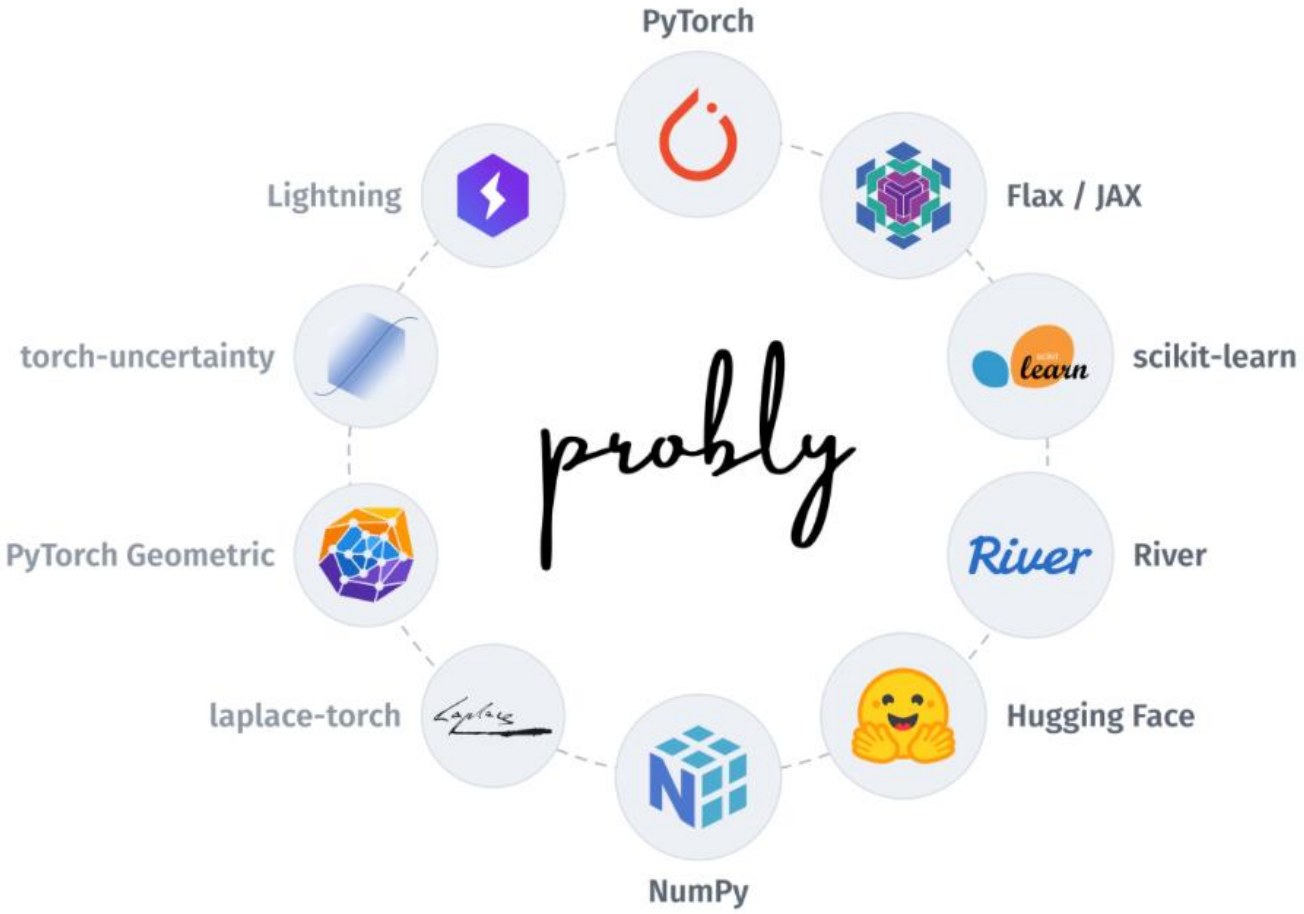


A hands-on session by Clemens Damke, Maximilian Muschalik, and Timo Löhr

probly Hi, we are Clemens, Max and Timo

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

MCML Munich Center for Machine Learning

probly Practical Uncertainty Quantification

README Contributing License

probly: Uncertainty Representation and Quantification for Machine Learning

pypi package 0.9.0 | python 3.13 | status pre-alpha | downloads 6k | License MIT

Pull Request Checks passing | codecov 92% | docs latest | contributions welcome | last commit yesterday

Turn any model into one that knows what it doesn't know.

`probly` is a **library-agnostic** toolkit for **uncertainty representation and quantification** in machine learning. Make any PyTorch, Flax/JAX, scikit-learn, River, or Hugging Face model uncertainty-aware in a single line, then **represent**, **quantify**, and **decompose** its predictive uncertainty into **aleatoric** and **epistemic** components. It ships 40+ methods ranging from Bayesian nets and deep ensembles to evidential, credal, and conformal prediction, all accessible via the same unified API.

🛠️ Install

`probly` is intended to work with **Python 3.13 and above**. Installation can be done via `pip` or `uv`:

```
pip install probly
```

```
uv add probly
```

About

Uncertainty Representation and Quantification for Machine Learning 🤔

pwhofman.github.io/probly/

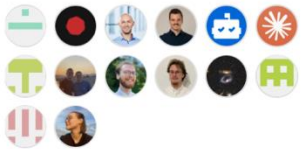
Readme
License
Contributing
Activity
57 stars
0 watching
18 forks
Report repository

Releases 4

v0.9.0 Latest
2 months ago

+ 3 releases

Contributors 55



+ 41 contributors

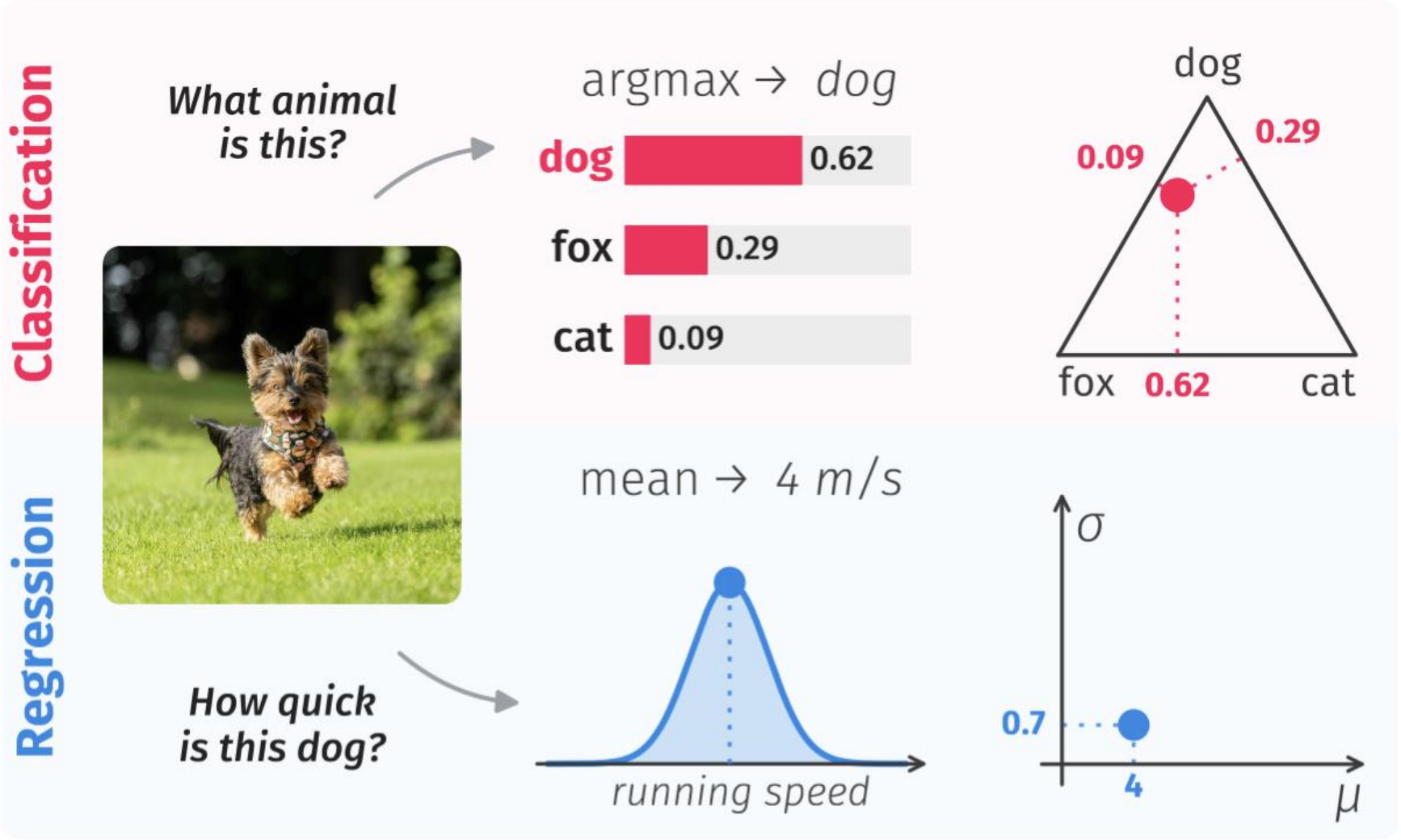
Languages

Python 100%

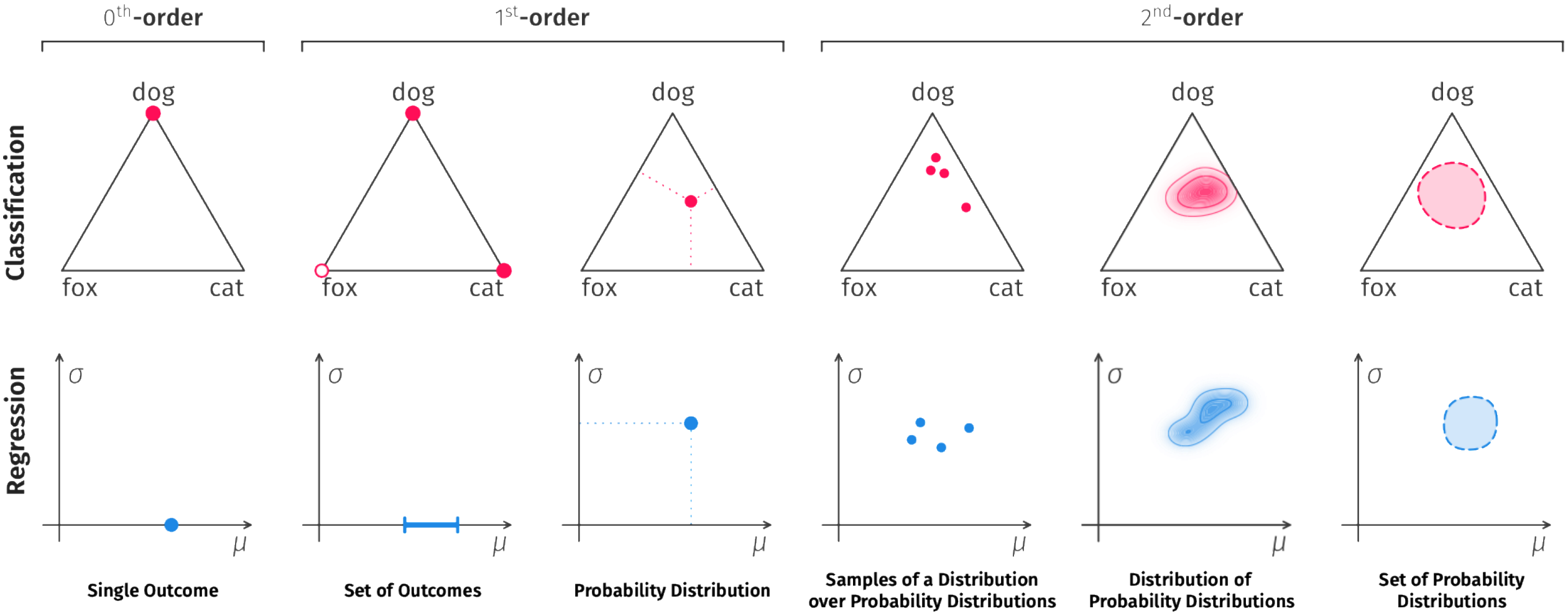
…but why?

- probly puts today's theory into practice
- probly speeds up your uncertainty experiments
- probly unifies uncertainty in ML for you

probly Uncertainty Quantification in Machine Learning



probly Uncertainty Quantification in Machine Learning



probly Uncertainty Quantification in Machine Learning

```python
import probly.transformation

# any model -> uncertainty
model = credal_ensembling(model)
train(model, data) # your logic
```

```python
import probly.representer

# build representations
rep = representer(model)
out = rep.represent(data)
```

```python
from probly.quantification

# TU / EU / AU
d = decompose(out)
d['TU'] == d['EU'] + d['AU']
```

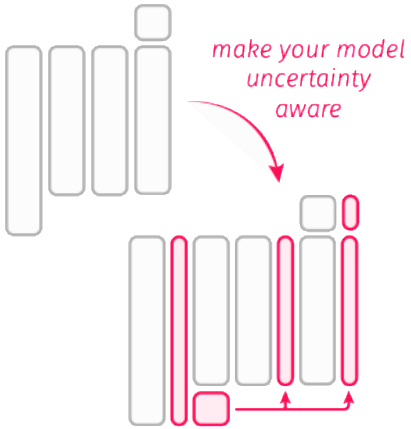
```python
import probly.decider

# bayes-optimal label
y = categorical_from_mean(out)
y = categorical_from_maximin(out)
```

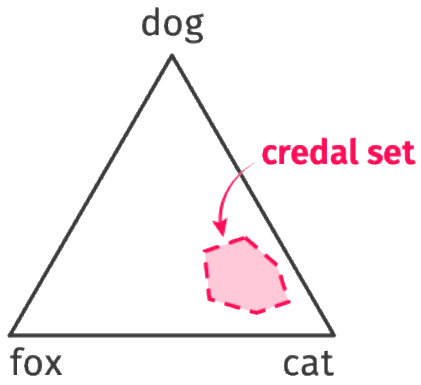
```python
import probly.datasets

# easily get first-order data
data = CIFAR10H()
data = ImageNetReaL()
```

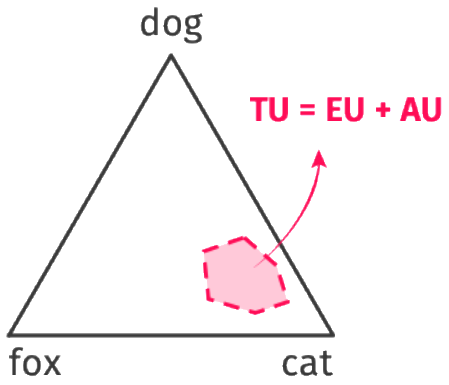
1 **transform** → 2 **represent** → 3 **quantify** **decide** → 4 **evaluate**



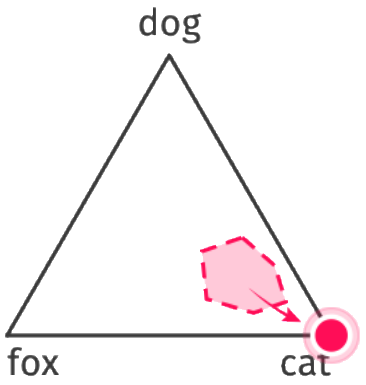
library-agnostic:
works natively with *torch*, *flax*, *huggingface*, *torch-uncertainty*, *sklearn*, *river*, ...



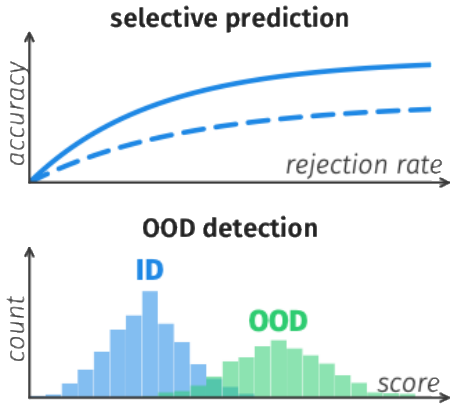
composable representations:
utilize *DirichletDistribution*, *ConvexCredalSet*, *Sample*, *OneHotConformalSet*, ...



common decompositions:
quantify the uncertainty and decompose it into *total*, *epistemic*, and *aleatoric*



canonical decisions:
use representations to make decisions or use your *own* logic



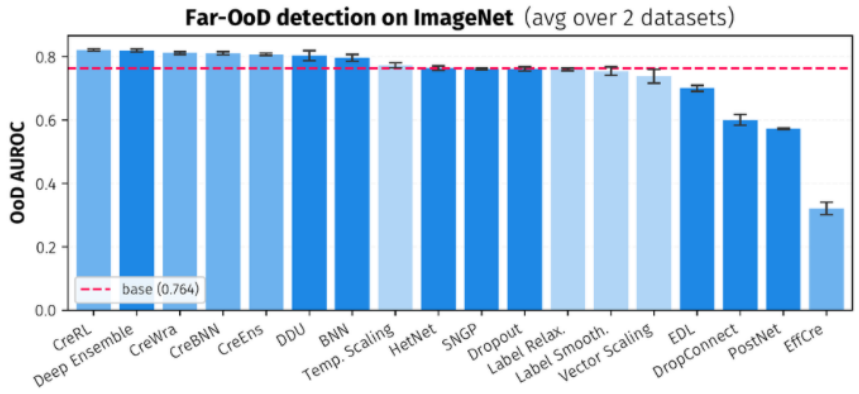
evaluation pipelines:
run experiments on *selective prediction*, *active learning*, *1st order data*, ...

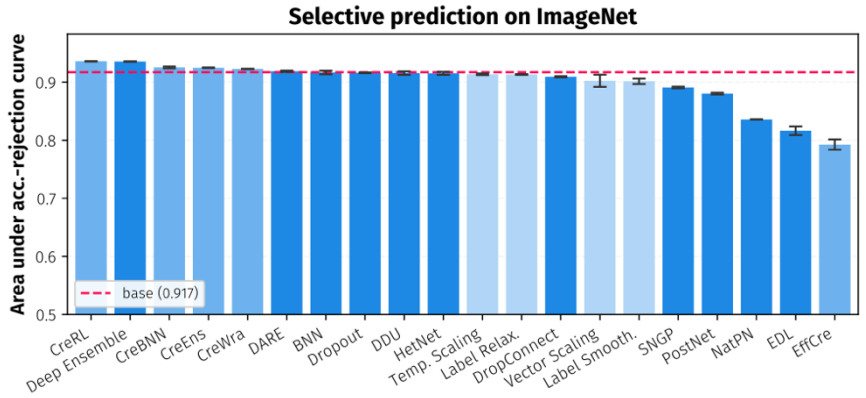
probly Uncertainty Quantification in Machine Learning

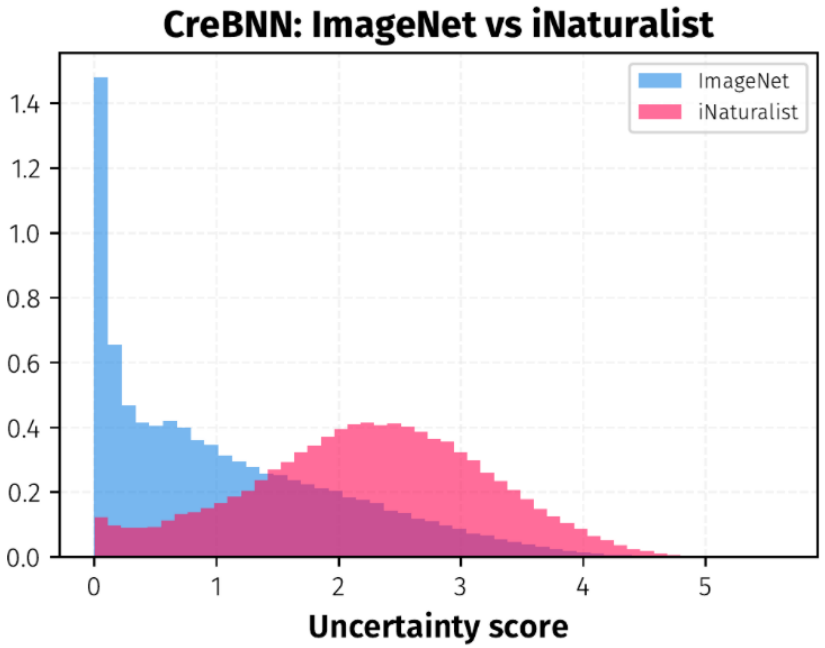
Method	How to use	probly [ours]	TU [92]	LUQ [100]	UT [119]
Second-order Distributions					
[55] MC Dropout	probly.method.dropout	✓	✓	✓	✓
[163] DropConnect	probly.method.dropconnect	✓	✗	✗	✗
[21] Mean-field BNN	probly.method.bayesian	✓	✓	✓	✗
[37] Laplace approximation	probly.method.laplace	✓	✓	✓	✓
[94] Deep Ensemble	probly.method.ensemble	✓	✓	✓	✓
[169] BatchEnsemble	probly.method.batchensemble	✓	✓	✗	✗
[154] Sub-ensemble	probly.method.subensemble	✓	✗	✗	✓
[40] DARE	probly.method.dare	✓	✗	✗	✗
[155] DUQ	probly.method.duq	✓	✗	✗	✓
[120] DDU	probly.method.ddu	✓	✗	✗	✓
[93] DEUP	probly.method.deup	✓	✗	✗	✓
[103] SNGP	probly.method.sngp	✓	✗	✓	✓
[145] Evidential classification	probly.method.evidential_classification	✓	✓	✗	✓
[26] Posterior Network	probly.method.posterior_network	✓	✗	✗	✓
[27] Natural Posterior Network	probly.method.natural_posterior_network	✓	✗	✗	✗
[111] Prior Network	probly.method.prior_network	✓	✗	✗	✗
[6] Evidential regression	probly.method.evidential_regression	✓	✓	✓	✗
[30] HetNet (heteroscedastic)	probly.method.het_net	✓	✓	✓	✓
Credal Sets					
[164] Credal Wrapper	probly.method.credal_wrapper	✓	✗	✗	✗
[125] Credal Ensembling	probly.method.credal_ensembling	✓	✗	✗	✗
[23] Credal BNN	probly.method.credal_bnn	✓	✗	✗	✗
[165] Credal Net	probly.method.credal_net	✓	✗	✗	✗
[105] Credal Relative Likelihood	probly.method.credal_relative_likelihood	✓	✗	✗	✗
[69] Efficient Credal Prediction	probly.method.efficient_credal_prediction	✓	✗	✗	✗
[79] Conformal credal set	probly.method.conformal_credal_set	✓	✗	✗	✗
[79] ↳ Conformal credal (inner product)	probly.method.conformal_inner_product	✓	✗	✗	✗
[79] ↳ Conformal credal (KL)	probly.method.conformal_kullback_leibler	✓	✗	✗	✗
[79] ↳ Conformal credal (TV)	probly.method.conformal_total_variation	✓	✗	✗	✗
[79] ↳ Conformal credal (Wasserstein)	probly.method.conformal_wasserstein_distance	✓	✗	✗	✗
[79] ↳ Conformal credal (Dirichlet RL)	probly.method.conformal_dirichlet_relative_likelihood	✓	✗	✗	✗

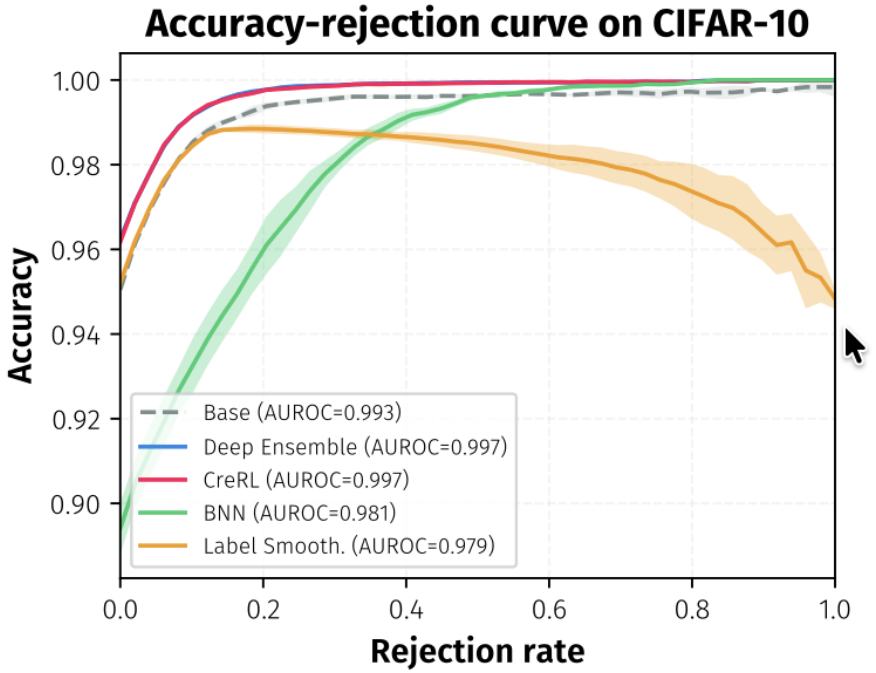
Method	How to use	probly [ours]	TU [92]	LUQ [100]	UT [119]
Conformal Prediction					
[136] LAC	probly.method.conformal.conformal_lac	✓	✓	✗	✗
[134] APS	probly.method.conformal.conformal_aps	✓	✓	✗	✗
[75] SAPS	probly.method.conformal.conformal_saps	✓	✗	✗	✗
[8] RAPS	probly.method.conformal.conformal_raps	✓	✓	✓	✗
[159] Absolute-error CP	probly.method.conformal.conformal_absolute_error	✓	✗	✗	✗
[133] CQR	probly.method.conformal.conformal_cqr	✓	✗	✓	✗
[133] CQR-r	probly.method.conformal.conformal_cqr_r	✓	✗	✗	✗
[135] UACQR	probly.method.conformal.conformal_uacqr	✓	✗	✗	✗
Calibration					
[62] Temperature scaling	probly.method.calibration.temperature_scaling	✓	✓	✓	✓
[131] Platt scaling	probly.method.calibration.platt_scaling	✓	✗	✗	✗
[62] Vector scaling	probly.method.calibration.vector_scaling	✓	✓	✗	✗
[25] Isotonic regression	probly.method.calibration.isotonic_regression	✓	✓	✗	✗
Losses					
[150] Label smoothing	probly.train.LabelSmoothingLoss	✓	✓	✗	✗
[101] Label relaxation	probly.train.LabelRelaxationLoss	✓	✗	✗	✗
[102] Focal loss	probly.train.FocalLoss	✓	✓	✗	✗
Uncertainty Quantification					
[42] Shannon entropy	probly.quantification.measure.distribution	✓	✓	✓	✓
[77] Upper and Lower Entropy	probly.quantification.measure.distribution	✓	✗	✗	✗
[145] Variance based	probly.quantification.measure.sample	✓	✗	✗	✗
[82] Dempster Shafer	probly.quantification.measure.distribution	✓	✗	✗	✓
[27] Vacuity	probly.quantification.measure.distribution	✓	✗	✗	✓
[48] Semantic entropy	probly.quantification.measure.entropy	✓	✗	✗	✗
[161] Spectral entropy	probly.quantification.spectral_entropy	✓	✗	✗	✗
[72] Clarification pipeline	probly.representer.HFQuestionClarifier	✓	✗	✗	✗

probly Uncertainty Quantification in Machine Learning





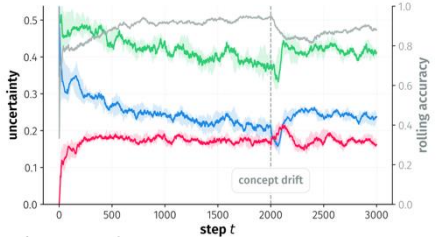




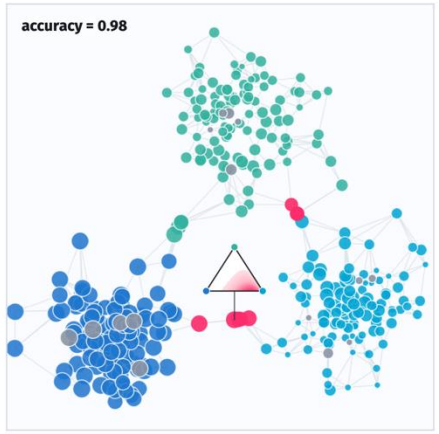
probly Uncertainty Quantification in Machine Learning

You can apply probly to…

Data Streams



Graph Analysis



LLMs, & more

Find the Intruders — a probly Challenge

1000 images came out of a digit-scanning pipeline, exactly 30 of them do not belong. Find them.

Your task

- Team up in groups up to 4
- Train your own model on MNIST
- Quantify uncertainty with probly; how is up to you
- Submit your image IDs on the challenge website
- Score = number of true intruders

Rules

- Detection must be uncertainty-based, with probly, on models you trained yourselves
- Matching against public datasets or picking images by eye is not a method
- Starter notebook available on Google Colab

Schedule

Ca. 14:30 challenge starts

15:30 -16:00 coffee break

16:30-17:00 award ceremony & solution presentation

Prizes: probly caps & LMU mugs
The winning team presents their solution

Challenge website

https://sipta2026.cortys.de