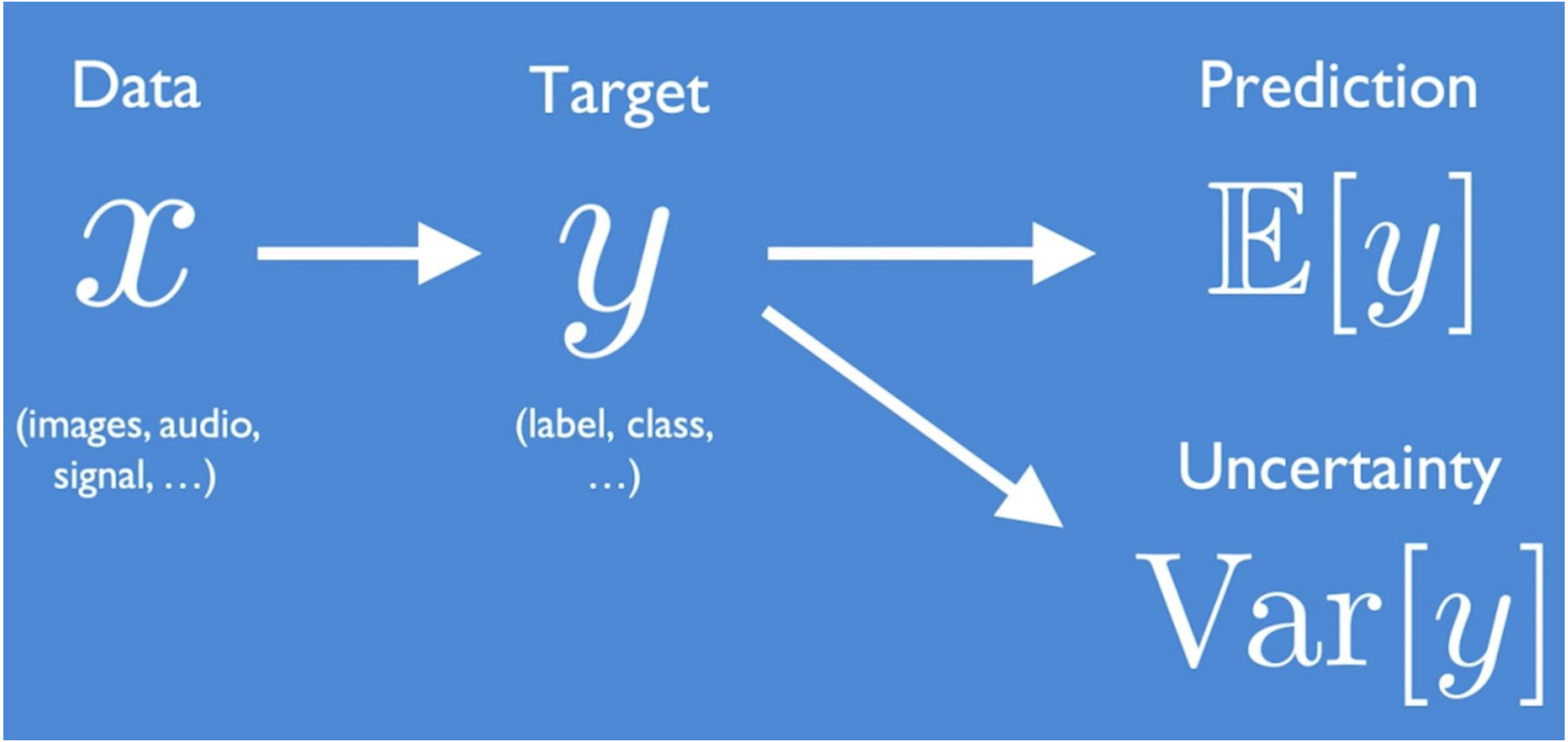
Uncertainty Quantifiction and Anomaly Detection with Evidential Deep Learning

Mark Neubauer

University of Illinois at Urbana-Champaign

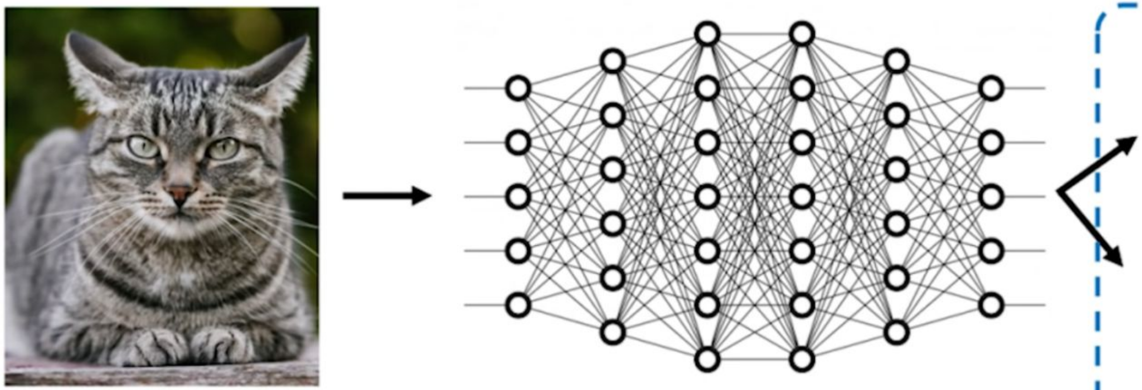
Artificial Intelligence and the Uncertainty Challenge in Fundamental Physics Workshop

Probabilistic Learning



http://introtodeeplearning.com/2021/slides/6S191_MIT_DeepLearning_L7.pdf

Learning Probabilistic Outputs



x

$$p(y = \text{“}cat\text{”} \mid x)$$

$$p(y = \text{“}dog\text{”} \mid x)$$

Probability distribution over
discrete class categories

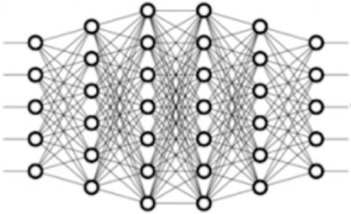
http://introtodeeplearning.com/2021/slides/6S191_MIT_DeepLearning_L7.pdf

Learning Discrete Class Targets

Classification

x

→



$$p(y = \text{“cat”} \,|\, x)$$

$$p(y = \text{“dog”} \,|\, x)$$

Activation: softmax(z) → $\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$

Loss: Neg. Log Likelihood (Cross Entropy) → $-\sum_{i=1}^{K} y_i \log p_i$

Why?

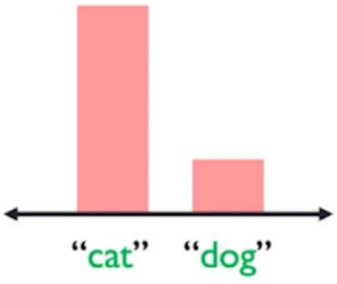
$$\boldsymbol{y} \sim \text{Categorical}(\boldsymbol{p})$$

Class Labels

Likelihood function

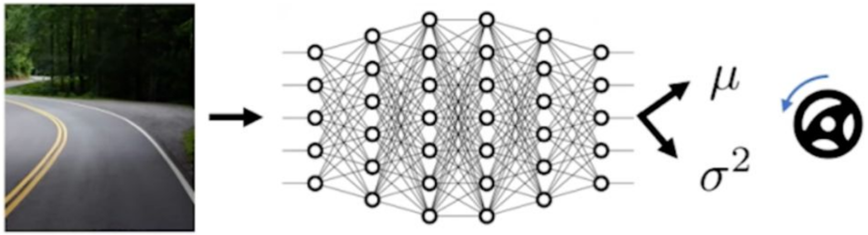
Distribution parameters (probabilities)

$$f(y = y_i \,|\, \boldsymbol{p}) = p_i$$



Learning Continuous Class Targets

Regression



x

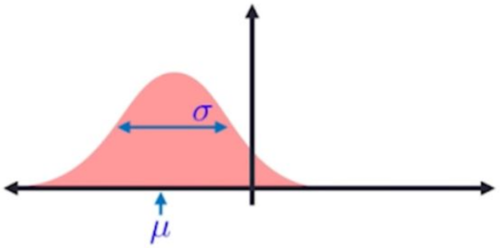
Activation: $\mu \in \mathbb{R}$, $\sigma > 0$ → $\mu = z_\mu$, $\sigma = \exp(z_\sigma)$

Loss: Neg. Log Likelihood → $-\log\left(\mathcal{N}(y|\mu,\sigma^2)\right)$

Why?

$$\underbrace{y}_{\text{Target Labels}} \sim \underbrace{\mathrm{Normal}}_{\text{Likelihood function}}\underbrace{(\mu,\sigma^2)}_{\text{Distribution parameters}}$$

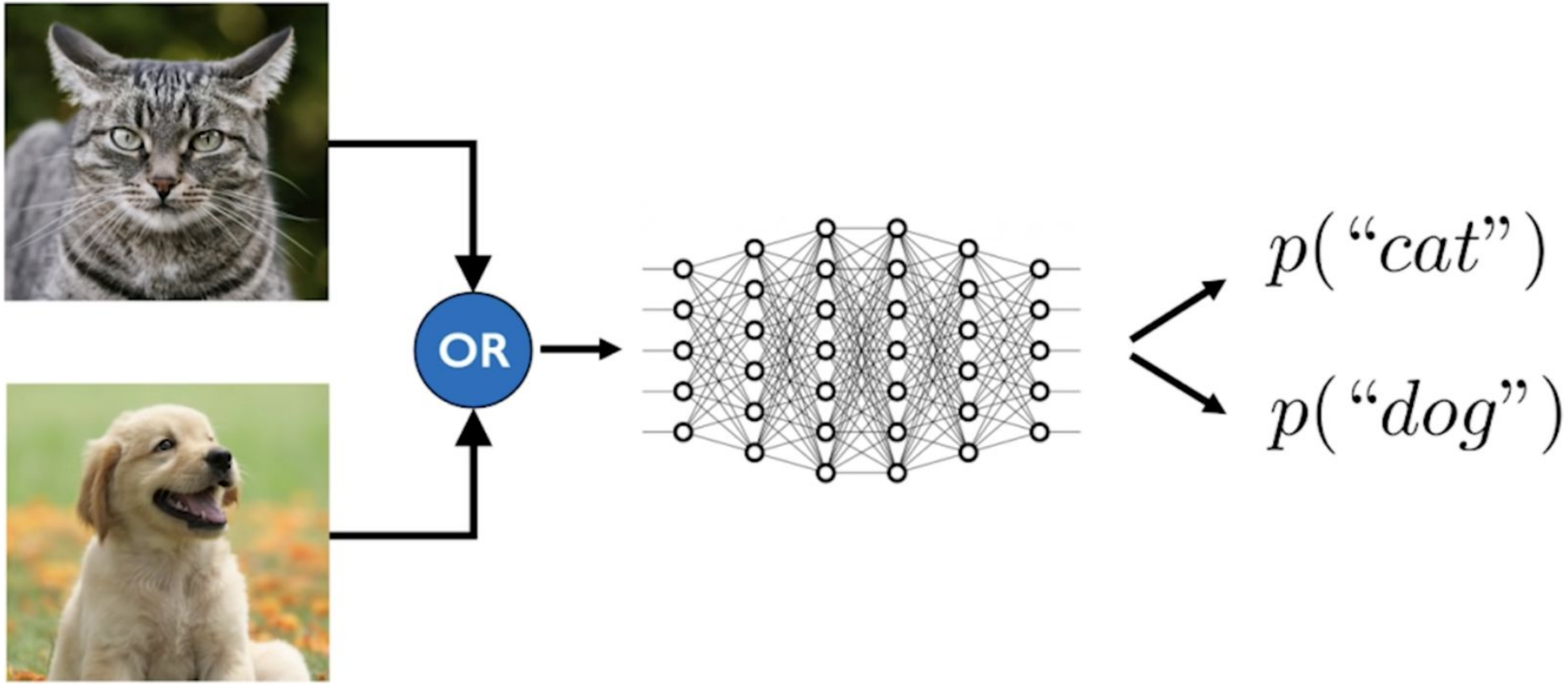
$$f(y\,|\,\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$$



http://introtodeeplearning.com/2021/slides/6S191_MIT_DeepLearning_L7.pdf

Likelihood vs Confidence

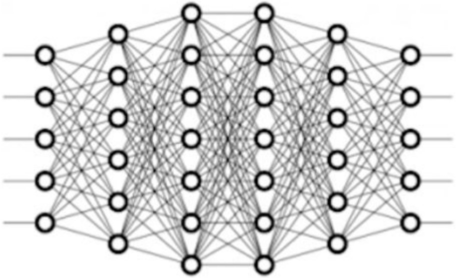
Do not mistake likelihood (probability) for model confidence!



http://introtodeeplearning.com/2021/slides/6S191_MIT_DeepLearning_L7.pdf

Likelihood vs Confidence

Do not mistake likelihood (probability) for model confidence!

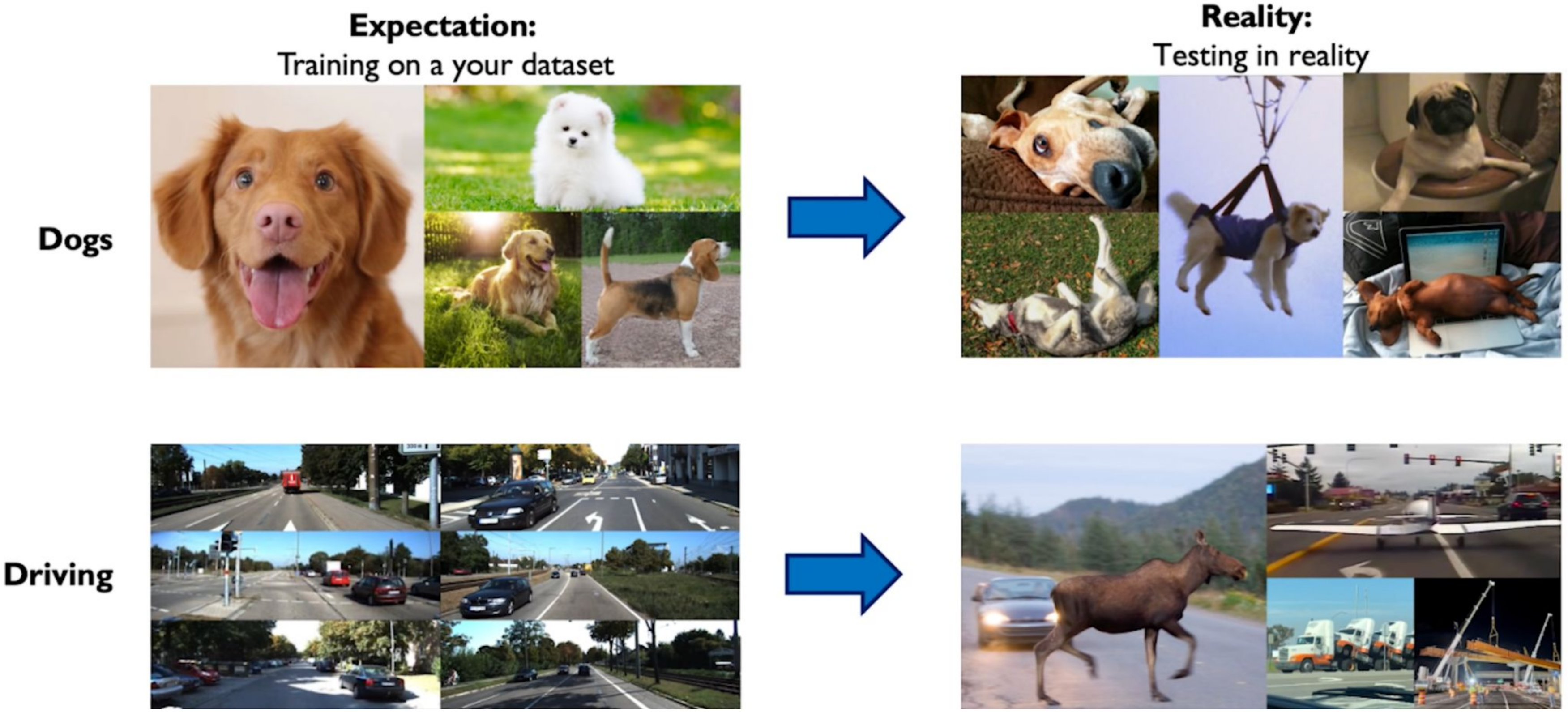


$$p(\text{“}cat\text{”}) = 0.5$$

$$p(\text{“}dog\text{”}) = 0.5$$

Likelihood vs Confidence

Do not mistake likelihood (probability) for model confidence!

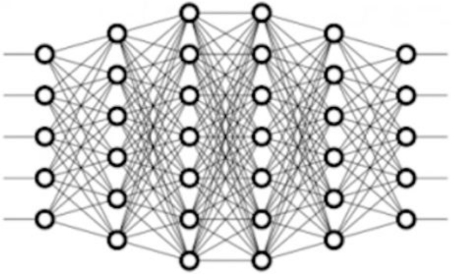


http://introtodeeplearning.com/2021/slides/6S191_MIT_DeepLearning_L7.pdf

Likelihood vs Confidence

Do not mistake likelihood (probability) for model confidence!

The output likelihoods will be unreliable if the input is ***unlike anything during training***



$$p(\text{“}cat\text{”}) = 0.15$$

$$p(\text{“}dog\text{”}) = 0.85$$

$$p(\text{“}cat\text{”}) + p(\text{“}dog\text{”}) = 1$$

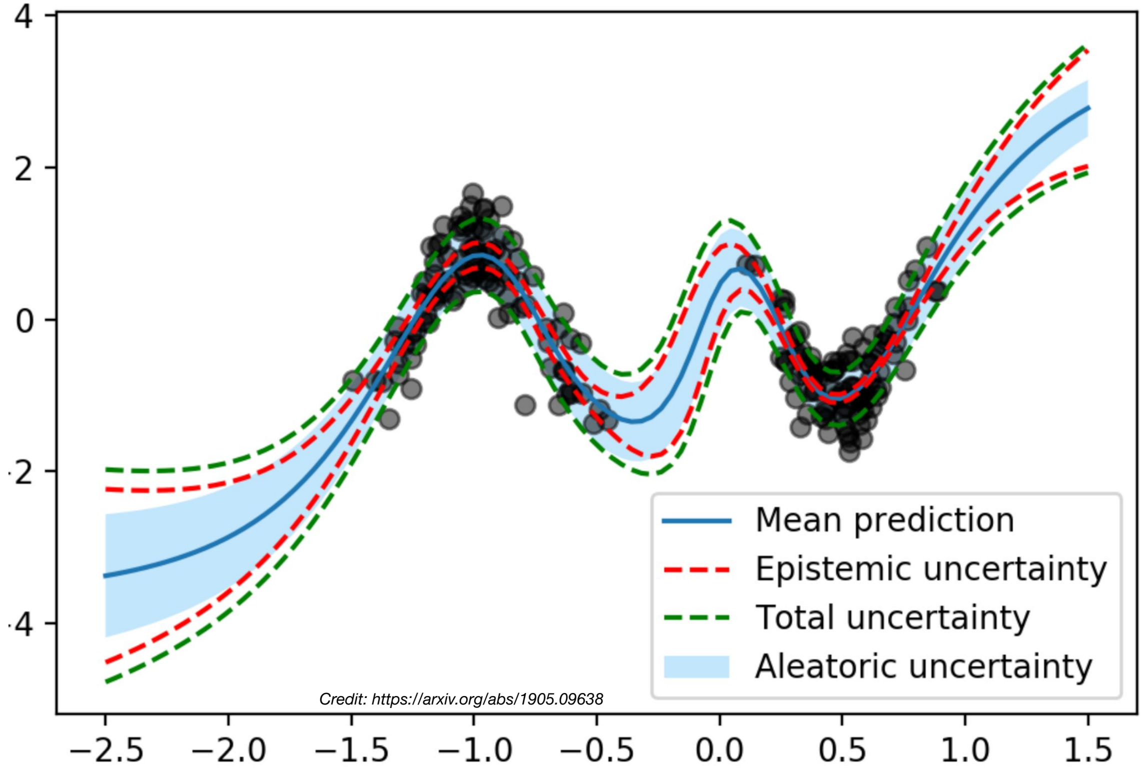
Types of Uncertainty

Aleatoric Uncertainty

- Describes the confidence in the input data
- Large when input data is noisy
- Cannot be reduced by simply adding more data

Epistemic Uncertainty

- Describes the confidence in the prediction
- Large when insufficient training data
- Can be reduced by adding more data



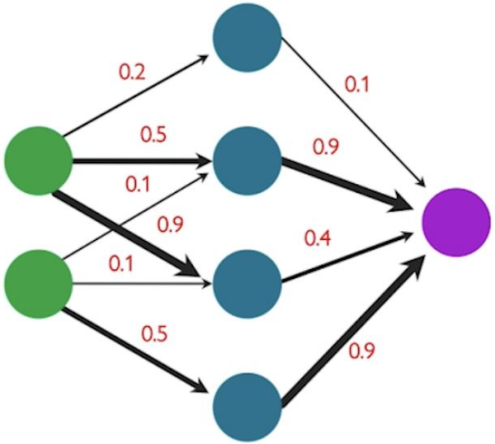
Credit: https://arxiv.org/abs/1905.09638

Estimating epistemic uncertainty

- Aleatoric uncertainty can be learned directly using NNs
- Epistemic uncertainty is much more challenging to estimate

Q: How can a model understand when it doesn't know the answer?

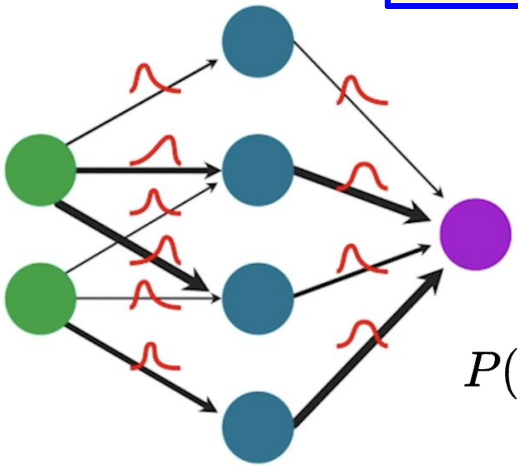
Deterministic NN



Learn fixed set of weights ***W***

One solution…

Bayesian NN



Learn a posterior over weights $P(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{Y})$

$$P(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{Y}) = \frac{P(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{W})P(\boldsymbol{W})}{P(\boldsymbol{Y}|\boldsymbol{X})}$$

Problem: Intractable! Needs approximations…

Gal+ ICML 2016

Approximations through Sampling

Evaluate T stochastic forward passes using different samples of weights $\{\boldsymbol{W}_t\}_{t=1}^{T}$

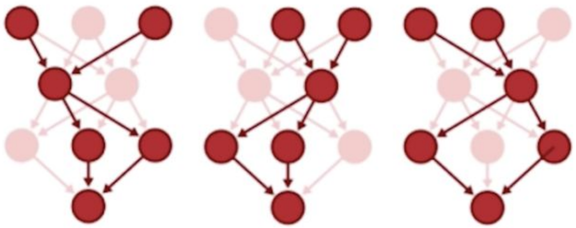
- Dropout as a form of stochastic sampling

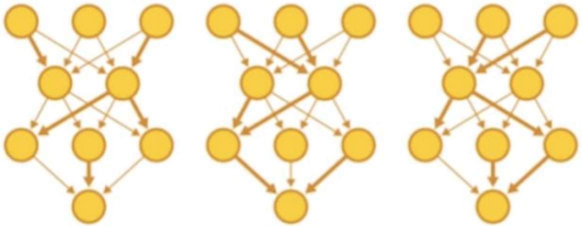
$$z_{w,t} \sim Bernoulli(p) \quad \forall\, w \in \boldsymbol{W}$$

- Ensemble of T independently trained models, each learning a unique

$$\boldsymbol{W}_t = train(f; \boldsymbol{X}, \boldsymbol{Y})$$

Monte Carlo Dropout





Model Ensembles

Epistemic uncertainty:

$$\boxed{Var(\hat{\boldsymbol{Y}}|\boldsymbol{X}) = \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{X})^2 - \mathbb{E}(\hat{\boldsymbol{Y}}|\boldsymbol{X})^2}$$

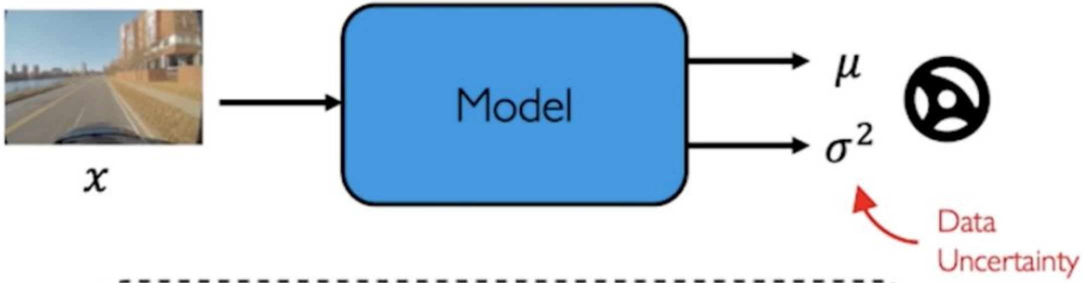
where $\mathbb{E}(\hat{\boldsymbol{Y}}|\boldsymbol{X}) = \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{X}|\boldsymbol{W}_t)$

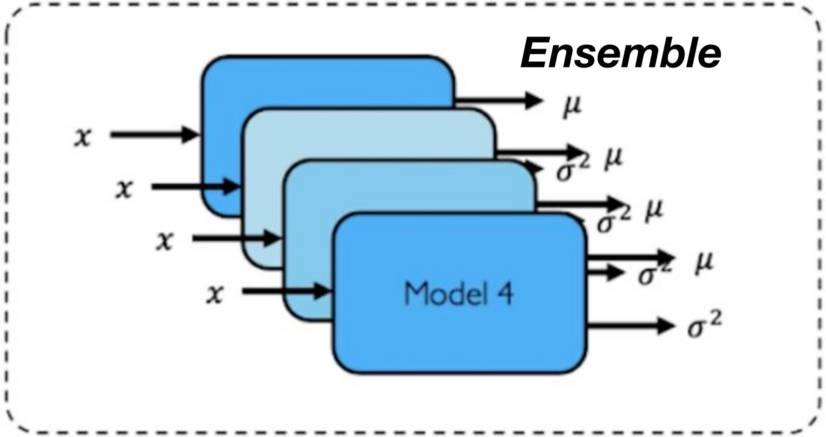
Downsides of Bayesian Deep Learning

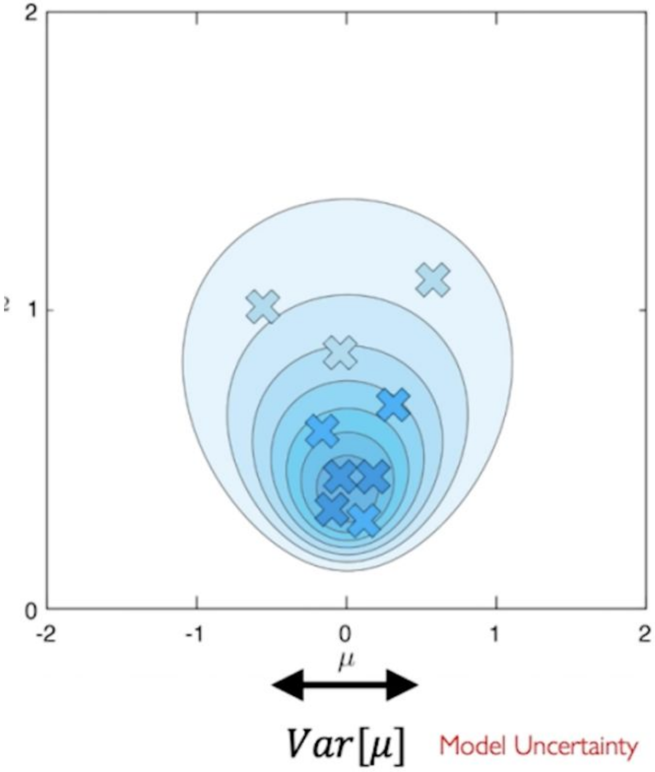
- ***Slow***: Requires running network T times for each input
- ***Memory***: Stores T copies of the network in parallel
- ***Efficiency***: Sampling hinders real-time on edge devices
- ***Calibration***: Sensitive to prior and often over-confident

Gal+ CIML 2016. Lakshminarayan+ NIPS 2017

Uncertainty Estimation: Sampling







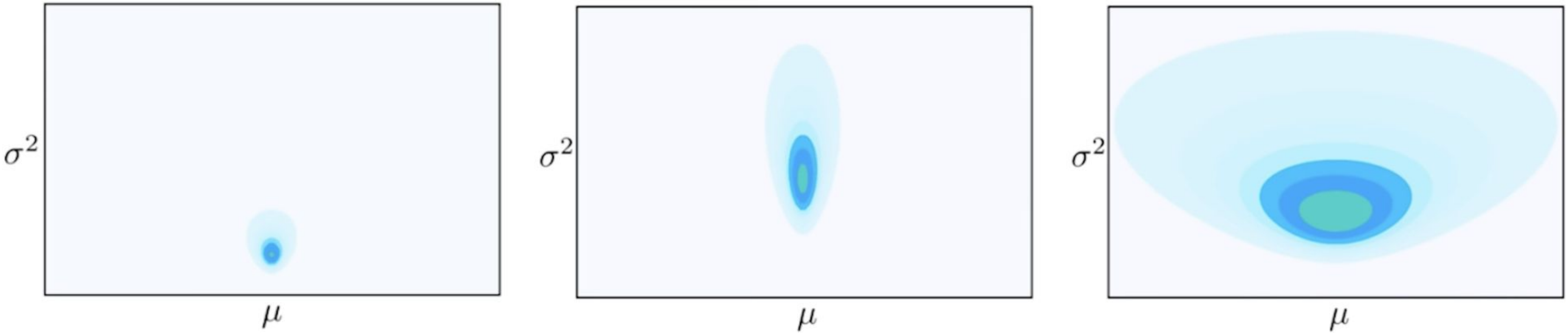
Q: Can we *directly* learn the parameters defining this likelihood distribution?

Lakshminarayan+ NIPS 2017

Evidential Deep Learning (EDL)

Treat learning and an evidence acquisition process

- More evidence → Increased predictive confidence



Low uncertainties → High confidence

High aleatoric (data) uncertainty

High epistemic (model) uncertainty

Goal: *train a neural network to learn these type of evidential distributions*

EDL for Regression

Key point to remember: **Sampling from an evidential distribution yields individual new distributions over the data**

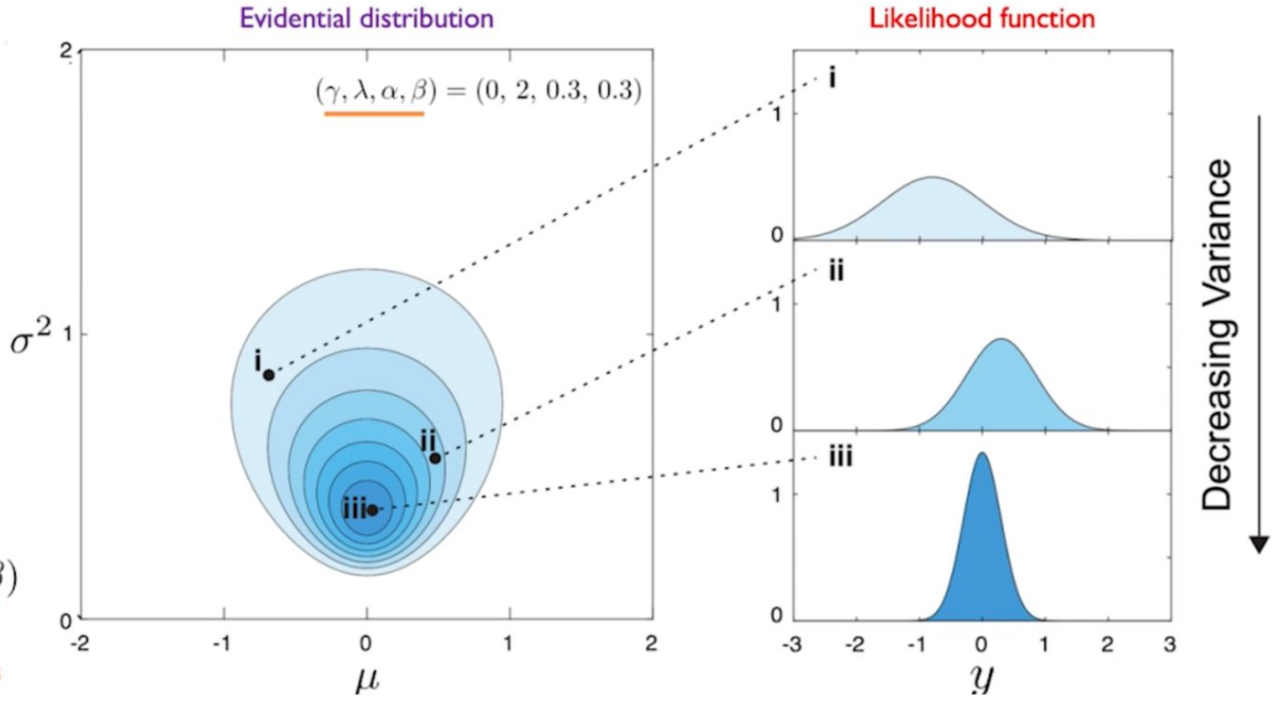
$$\underbrace{y}_{\text{Target Labels}} \sim \underbrace{\text{Normal}}_{\text{Likelihood function}}\underbrace{(\mu, \sigma^2)}_{\text{Distribution parameters}}$$

Assume the distribution parameters are not known, place priors over each and probabilistically estimate!

$$\mu \sim \text{Normal}(\gamma,\ \sigma^2 \upsilon^{-1})$$

$$\sigma^2 \sim \Gamma^{-1}(\alpha, \beta)$$

$$\underbrace{\mu, \sigma^2}_{\text{Distribution parameters}} \sim \underbrace{\text{NormalInvGamma}}_{\text{Evidential Prior}}\underbrace{(\gamma, \upsilon, \alpha, \beta)}_{\text{Model parameters}}$$



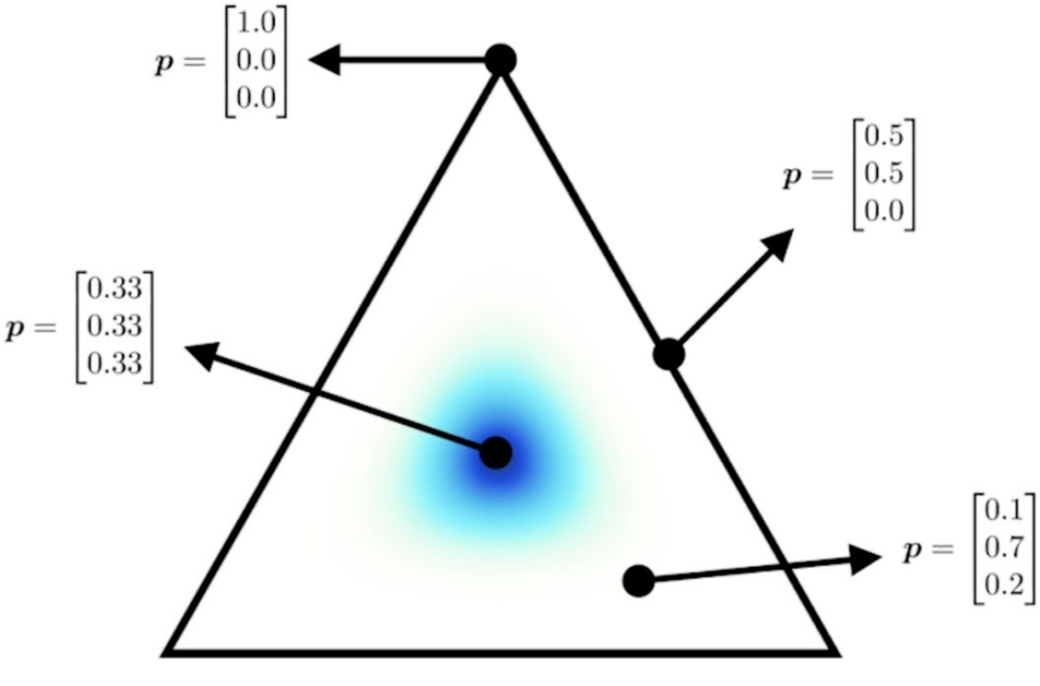
EDL for Classification

Sampling from an evidential distribution yields individual new distributions over the data

$$y \in \{1, \cdots, K\}$$

$$\underbrace{y}_{\text{Class Labels}} \sim \underbrace{\text{Categorical}}_{\text{Likelihood function}}(\underbrace{\boldsymbol{p}}_{\text{Distribution parameters (probabilities)}})$$

$$\underbrace{\boldsymbol{p}}_{\text{Distribution parameters}} \sim \underbrace{\text{Dirichlet}}_{\text{Evidential Prior}}(\underbrace{\boldsymbol{\alpha}}_{\text{Model parameters}})$$



$$K = 3; \quad \boldsymbol{\alpha} = (5, 5, 5)$$

Sensoy+ NeurIPS 2018

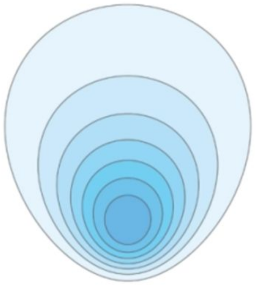
EDL Distributions

Regression

$y \in \mathbb{R}$

$y \sim \mathrm{Normal}(\mu, \sigma^2)$

$\mu, \sigma^2 \sim \mathrm{NormalInvGamma}(\gamma, \upsilon, \alpha, \beta)$

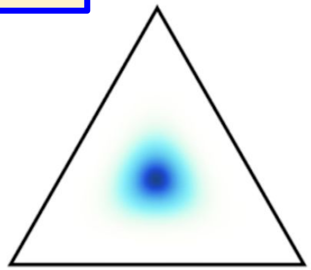


Classification

$y \in \{1, \cdots, K\}$

$y \sim \mathrm{Categorical}(\boldsymbol{p})$

$\boldsymbol{p} \sim \mathrm{Dirichlet}(\boldsymbol{\alpha})$



Note that the choice of evidential distributions is closely related to conjugate priors in the context of Bayesian inference

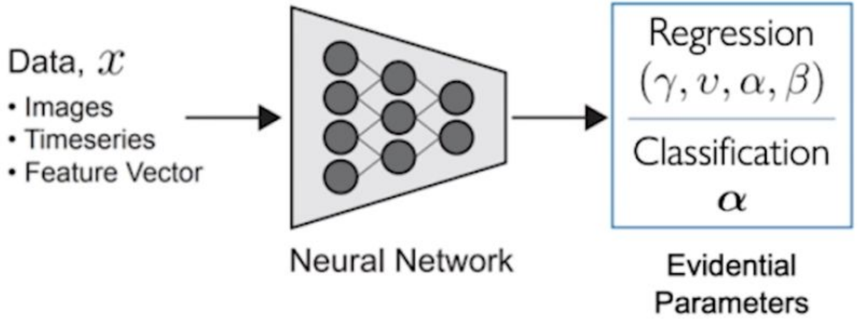
- It is often easiest for computations to pick your evidential distribution to be a conjugate prior of your likelihood:

$$p(\theta|y) = \frac{p(y|\theta)\, p(\theta)}{\int_{\theta'} p(y|\theta')\, p(\theta')\, d\theta'}$$

EDL Model and Training

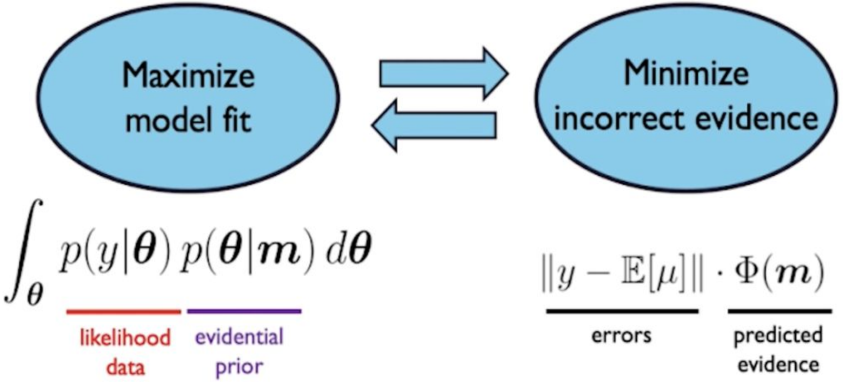
Model

Train the network to output the parameters of an evidential distribution:



Optmization

Perform multi-objective training:



For classification:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{N} \mathcal{L}_i(\Theta) + \lambda_t \sum_{i=1}^{N} KL[D(\mathbf{p}_i|\tilde{\boldsymbol{\alpha}}_i) \,||\, D(\mathbf{p}_i|\mathbf{1})]$$

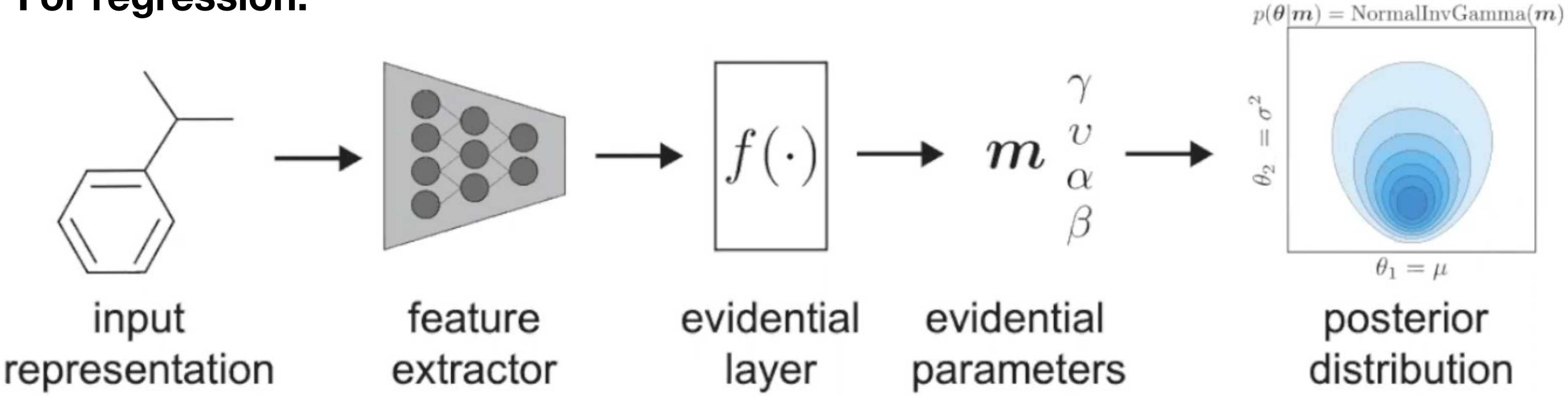
Reconstruction Loss

Penalty for assigning large confidence to uncertain samples

Amini+ NeurIPS 2020, Sensoy+ NeurIPS 2018

Forming EDL Predictions

For regression:



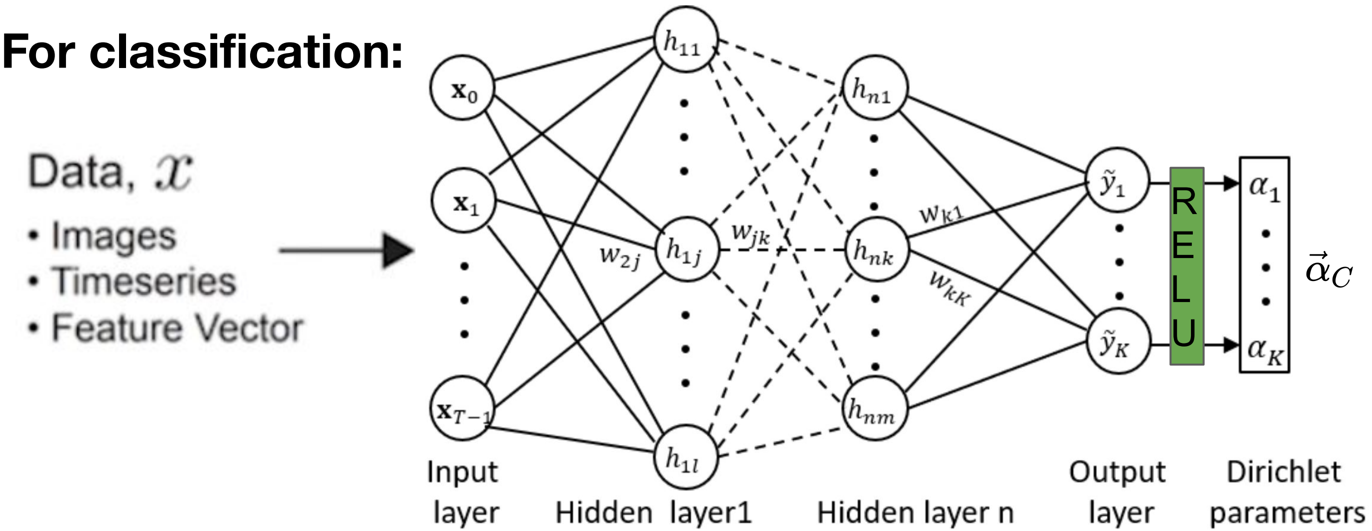
$$\underbrace{\mathbb{E}[\mu] = \gamma}_{\text{prediction}} \qquad \underbrace{\mathbb{E}[\sigma^2] = \frac{\beta}{\alpha - 1}}_{\text{aleatoric}} \qquad \underbrace{\text{Var}[\mu] = \frac{\beta}{\upsilon(\alpha - 1)}}_{\text{epistemic}}$$

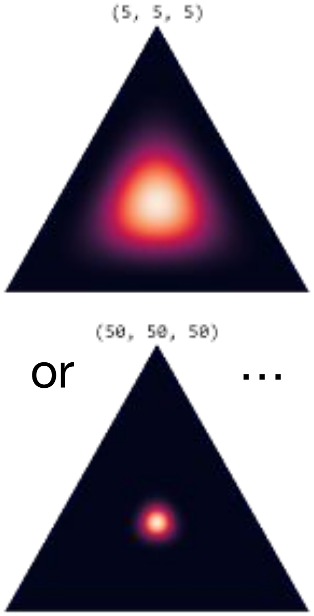
→ ***Evidential Uncertainty can be easily integrated with 4x parameters and a new loss***

Amini+ NeurIPS 2020, Sensoy+ NeurIPS 2018

Forming EDL Predictions

For classification:





Once the network learns the parameters **α** , its mean, can be taken as an estimate of the *K* class probabilities

$$\boxed{\tilde{p}_c = \alpha_c / S}$$

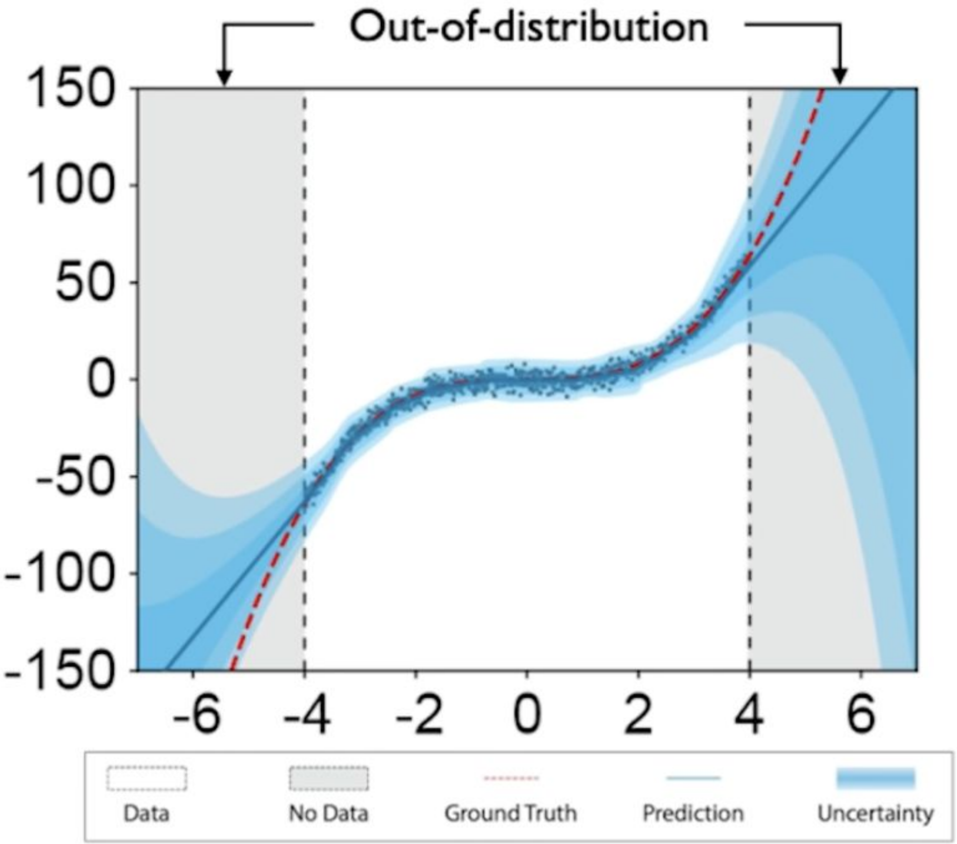
The epistemic uncertainty *u* on the prediction is computed as the inverse of total evidence or Dirichlet strength *S*

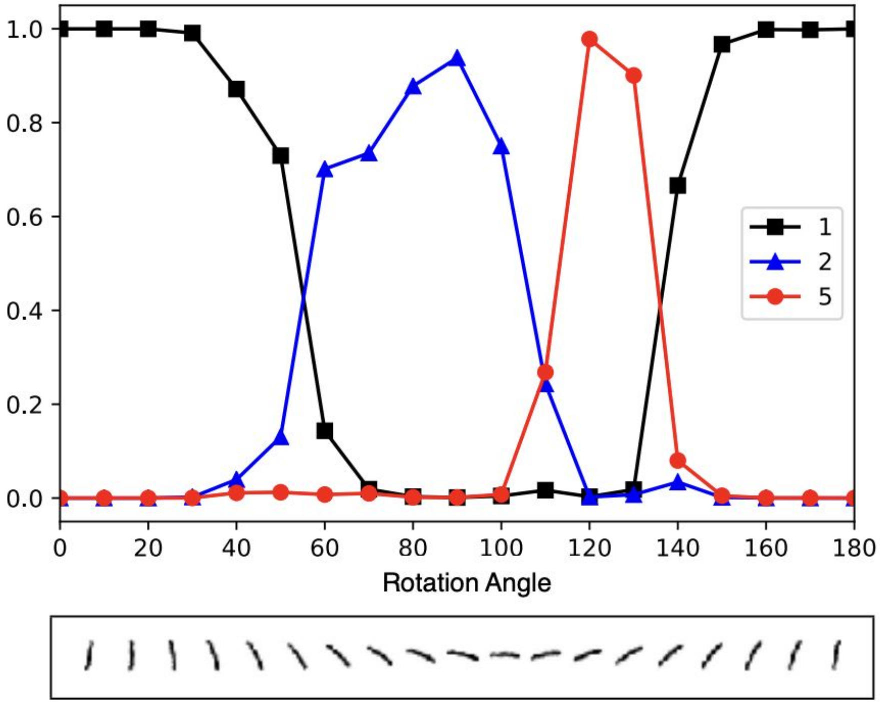
$$\boxed{u = K/S}$$ where $$S = \sum_{c=1}^{K} \alpha_c$$

→ ***Evidential Uncertainty can be easily integrated with × K parameters and a new loss***

Amini+ NeurIPS 2020, Sensoy+ NeurIPS 2018

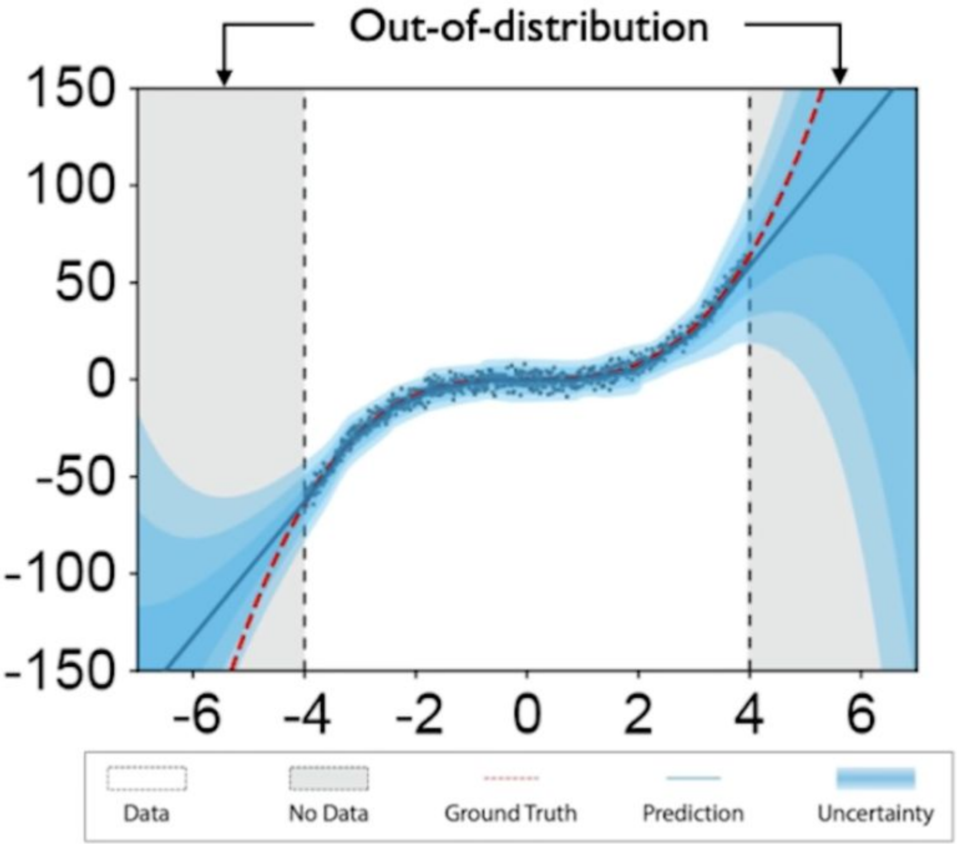
EDL Toy Learning Problems

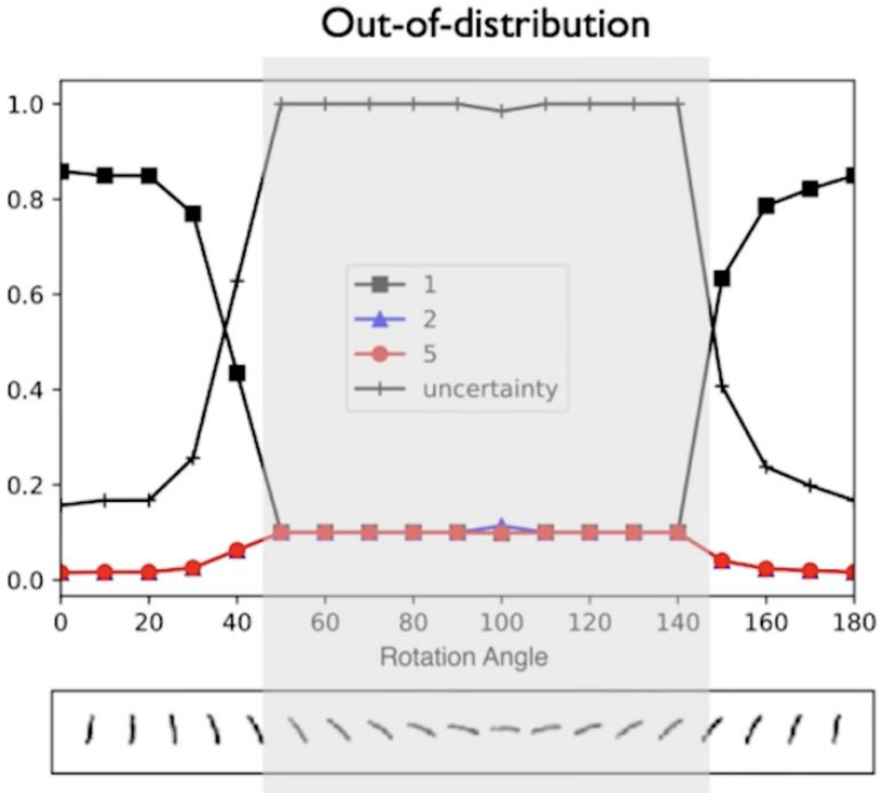




Amini+ NeurIPS 2020, Sensoy+ NeurIPS 2018

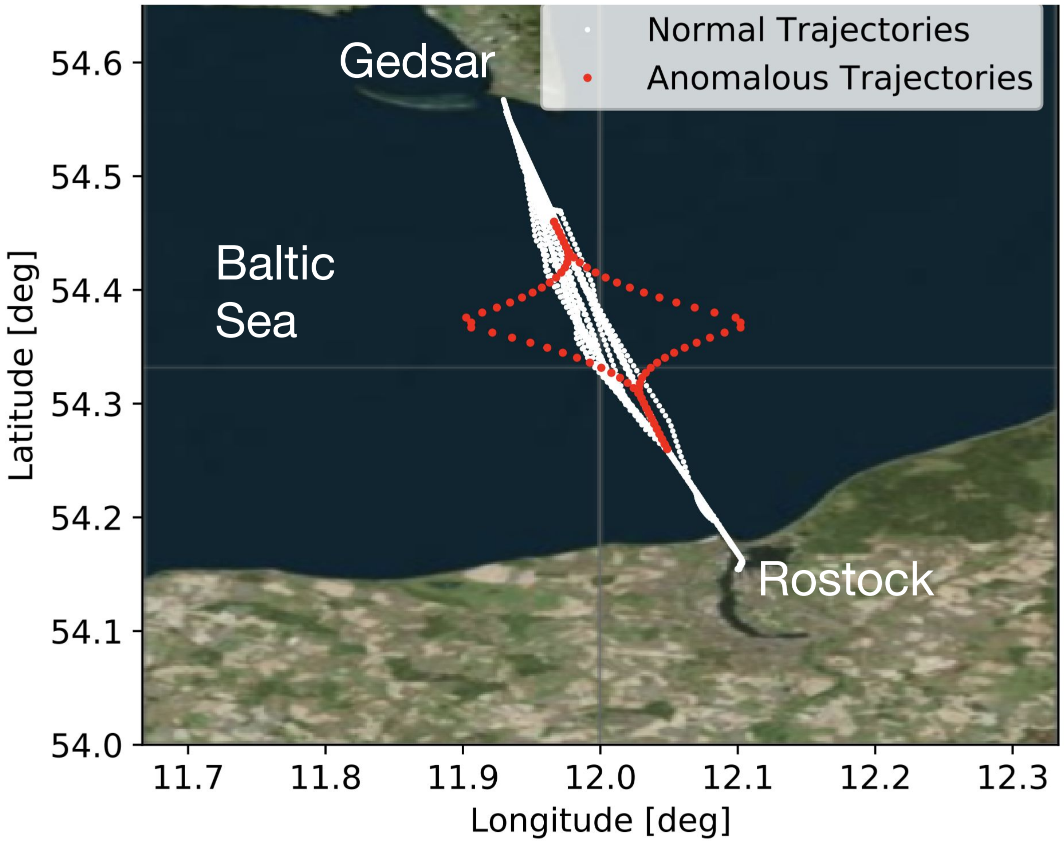
EDL Toy Learning Problems





Amini+ NeurIPS 2020, Sensoy+ NeurIPS 2018

EDL Applied to Anomaly Detection



Maritime Anomaly Detection

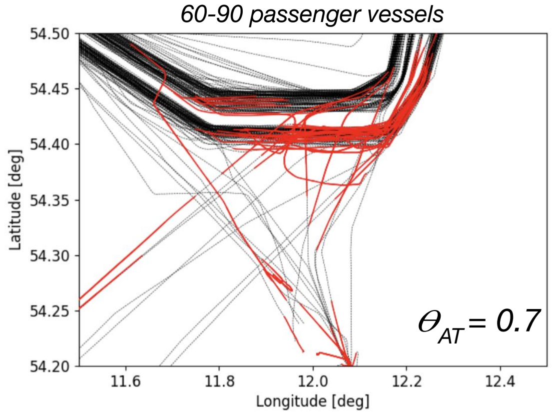
Most ships are equipped with automatic identification system (AIS) transponders to provide their static and dynamic information

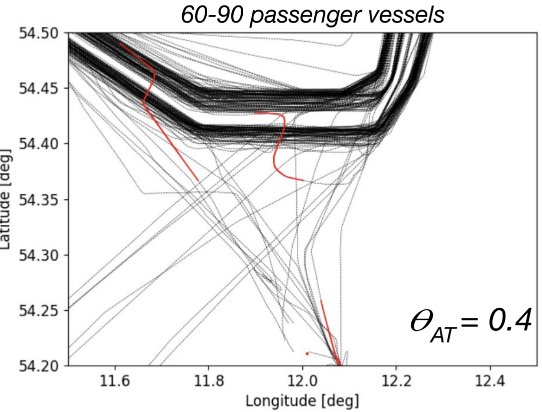
Vessels' location, navigational status, and voyage-related information can be used for

- *collision-avoidance mechanisms*
- *vessel tracking*
- detection of *loss of AIS signal* and *anomalous trajectories*

https://arxiv.org/pdf/2107.01557v1.pdf

Maritime Anomaly Detection





EDL for Anomalous Trajectory Detection

High epistemic uncertainty may represent anomalous trajectory. However, different output features are predicted with different uncertainties, so comparing segments with a set uncertainty threshold might not be a good idea

Thus, a trajectory segment is defined as anomalous if the predicted sequences of the segment have an abrupt transition in their epistemic uncertainties

$$\min_d \left[\frac{\min_j(\mathrm{var}[\mu_j^d])}{\max_j(\mathrm{var}[\mu_j^d])} \right] < \Theta_{AT}$$

This selects the feature *d* and output sequence *j* with the minimum normalized epistemic uncertainties. If this value is below Θ_{AT} , then the segment is considered as anomalous

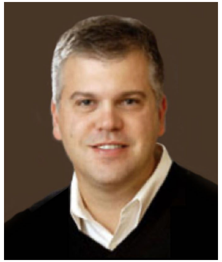
A vessel's trajectory is termed as anomalous if it contains one or more anomalous segments

https://arxiv.org/pdf/2107.01557v1.pdf

Builds upon XAI results from [arxiv: 2210.04371](https://arxiv.org/abs/2210.04371)
Published in 2023 *Mach. Learn.: Sci. Technol.* 4 035003


Ayush Khot

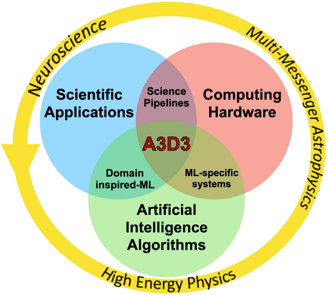

Avik Roy


Mark Neubauer


Dewen Zhong


Xiwei Wang

This work was supported by the FAIR Data program of the DOE ASCR under contract number DE-SC0021258, DOE OHEP under contract number DE-SC0023365, and NSF subaward from award MPS/PHY-2117997



U.S. DEPARTMENT OF ENERGY | Office of Science

FAIR4HEP

A Detailed Study of Interpretability of Deep Neural Network based Top Taggers

Ayush Khot, Mark S. Neubauer, and Avik Roy[1]

Department of Physics & National Center for Supercomputing Applications (NCSA)
University of Illinois at Urbana-Champaign

E-mail: akhot2@illinois.edu, msn@illinois.edu, avroy@illinois.edu

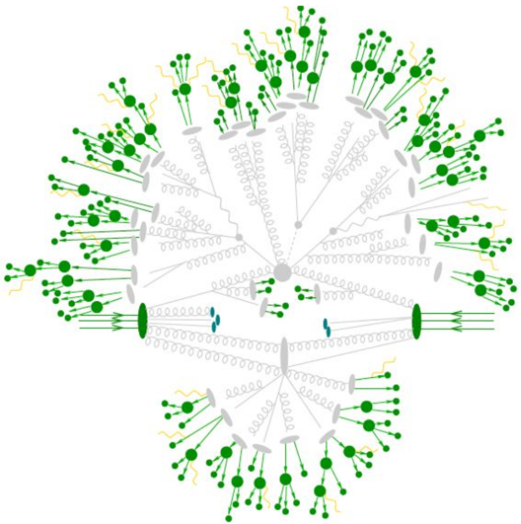
ABSTRACT: Recent developments in the methods of explainable AI (XAI) allow researchers to explore the inner workings of deep neural networks (DNNs), revealing crucial information about input-output relationships and realizing how data connects with machine learning models. In this paper we explore interpretability of DNN models designed to identify jets coming from top quark decay in high energy proton-proton collisions at the Large Hadron Collider (LHC). We review a subset of existing top tagger models and explore different quantitative methods to identify which features play the most important roles in identifying the top jets. We also investigate how and why feature importance varies across different XAI metrics, how correlations among features impact their explainability, and how latent space representations encode information as well as correlate with physically meaningful quantities. Our studies uncover some major pitfalls of existing XAI methods and illustrate how they can be overcome to obtain consistent and meaningful interpretation of these models. We additionally illustrate the activity of hidden layers as Neural Activation Pattern (NAP) diagrams and demonstrate how they can be used to understand how DNNs relay information across the layers and how this understanding can help to make such models significantly simpler by allowing effective model reoptimization and hyperparameter tuning. These studies not only facilitate a methodological approach to interpreting models but also unveil new insights about what these models learn. Incorporating these observations into augmented model design, we propose the Particle Flow Interaction Network (PFIN) model and demonstrate how interpretability-inspired model augmentation can improve top tagging performance.

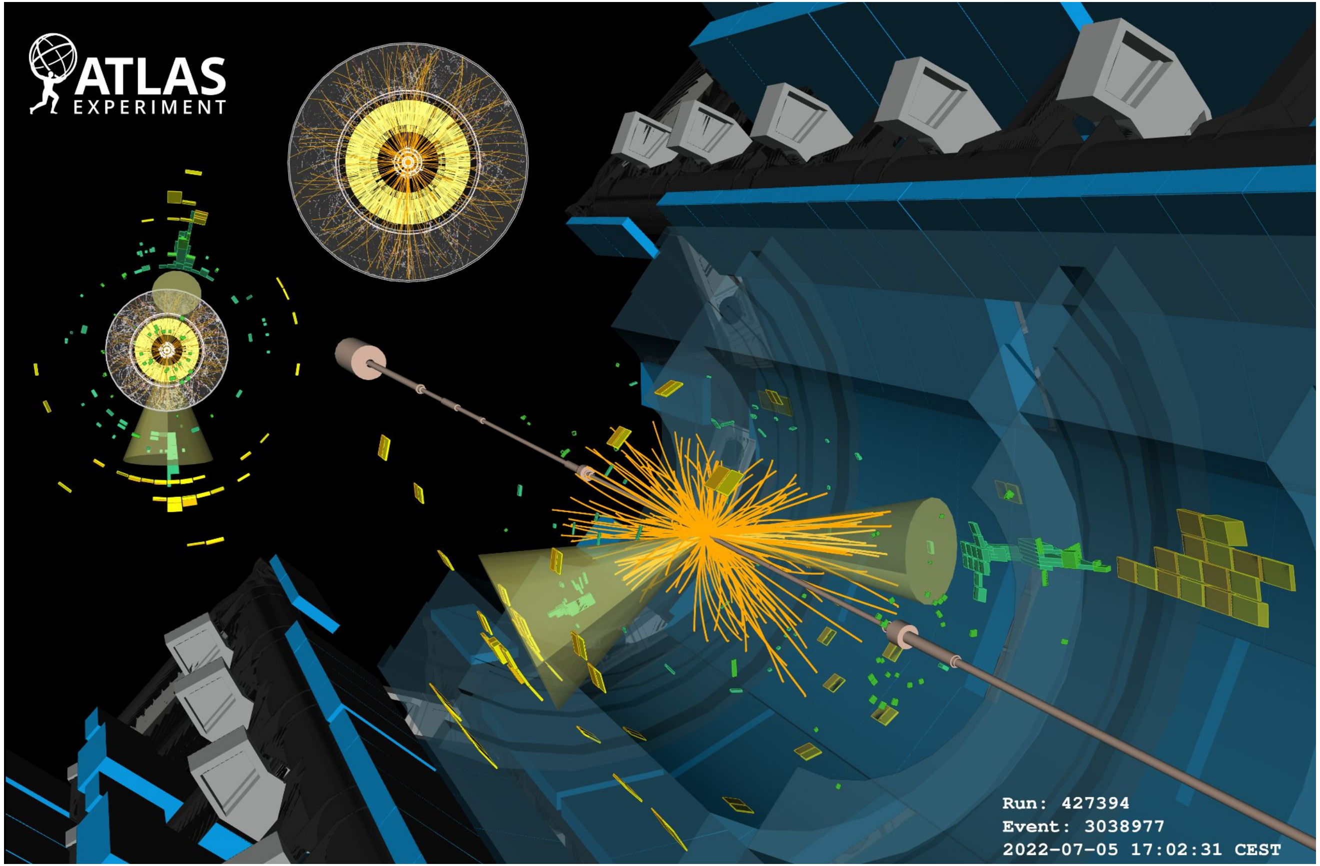
see [XAI talk in the Tuesday session]

Jets at the Large Hadron Collider

Colliding protons at the LHC produce collections of particles called “jets”

- Observed as clusters of energy and tracks





Jet Tagging: Classification in HEP

- Jets can emerge from many processes, and we want to identify the "type" of the process that gives us the jet
- Classic example from HEP: QCD and top jet classification
- Includes information about momenta (p_x, p_y, p_z) and energy (e) of up to 200 particles that make up the jet
- The total energy (momentum) of the jet is obtained by a scalar (vector) summation of the particle-level

Transverse momentum

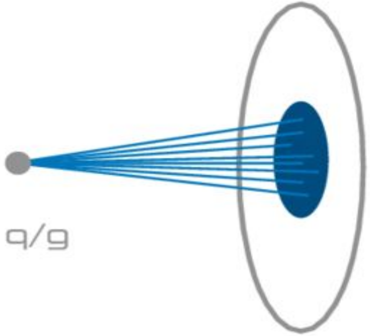
$$p_t = \sqrt{p_x^2 + p_y^2}$$

Azimuthal angle

$$\phi = tan^{-1}\left(\frac{p_y}{p_x}\right)$$

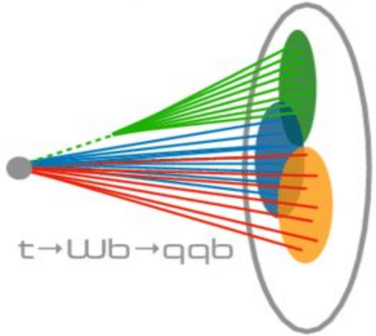
pseudorapidity

$$\eta = \frac{1}{2}\ln\left(\frac{e+p_z}{e-p_z}\right)$$



Jets from light quarks/gluons: QCD jets

Simulated dataset with 2M jets available at: zenodo: 2603256



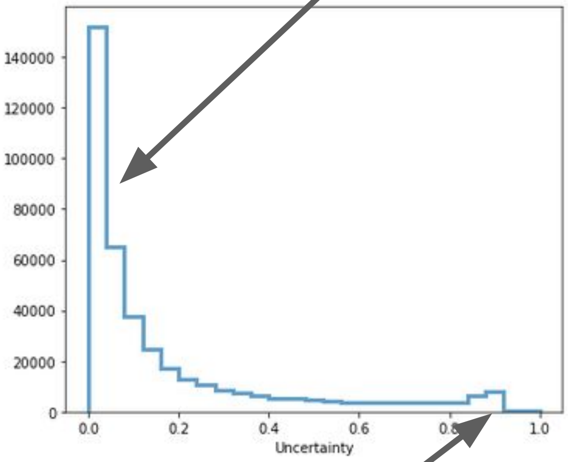
Jets from top quarks: top jets

Uncertainties in Jet Tagging - I

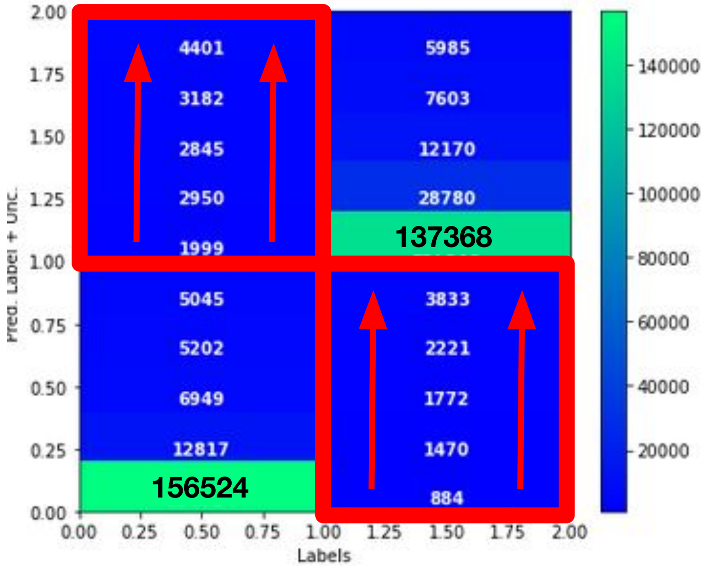
- **Q**: To what extent can a jet tagging model be confident in its predictions?

- Using the XAI-Inspired Interaction based Graph NN called [Particle Flow Interaction Network]() (PFIN) for top tagging
- Binary classification goal: distinguish signal ***top-quark jets*** (*label*=1) from the background **QCD jets** (*label*=0)

Model shows high confidence for most jets



Large uncertainties dominated by misclassified jets!

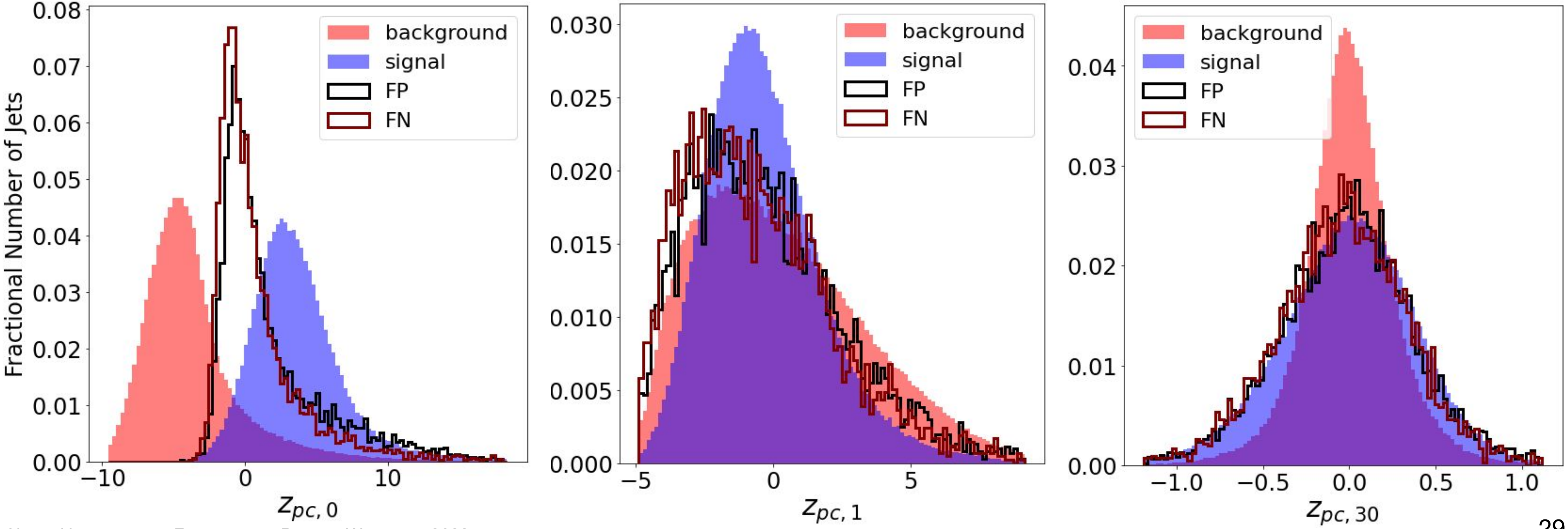


Increasing uncertainty for misclassified jets!

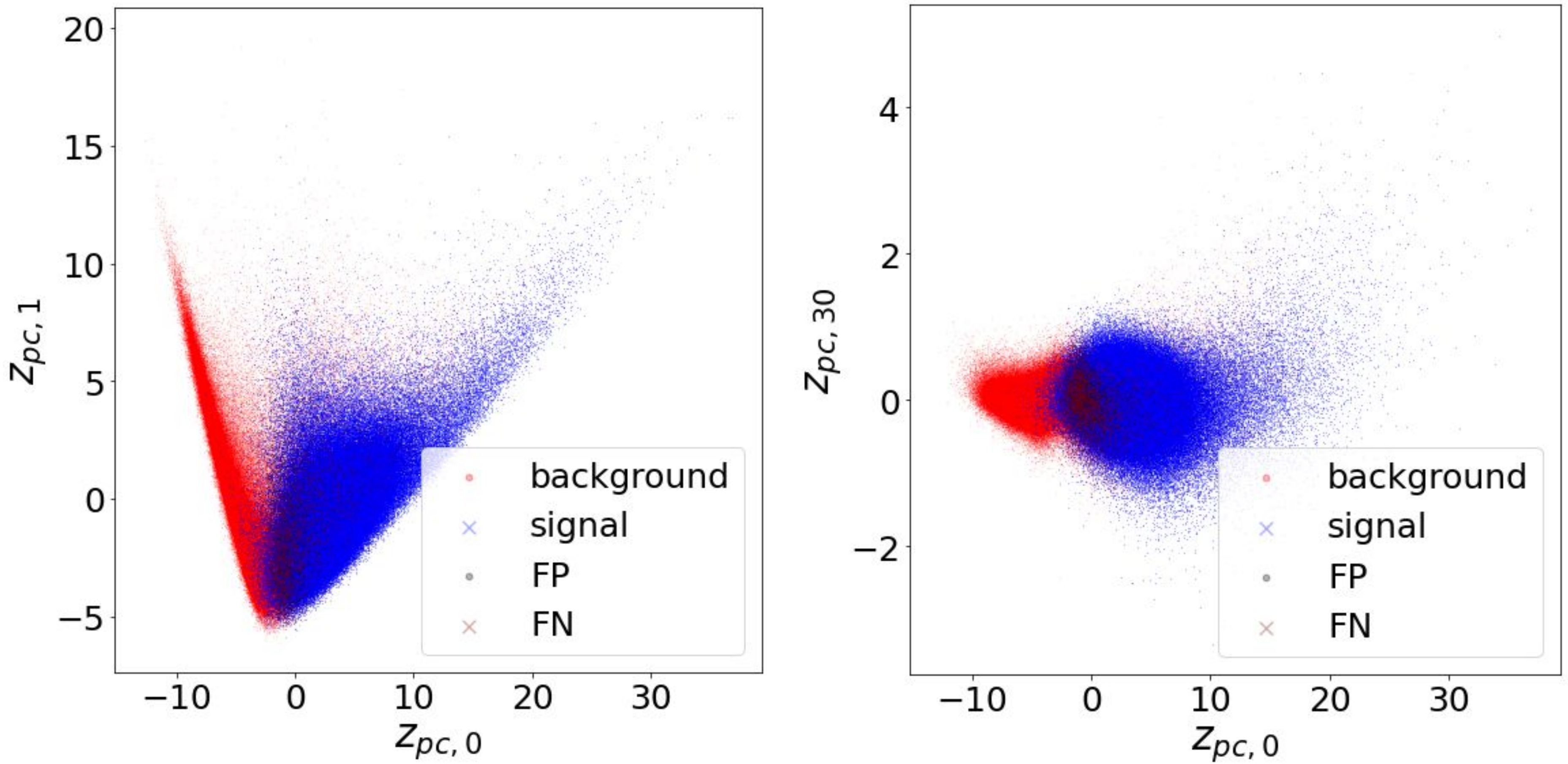
Who Gets Largest Uncertainties?

[Studies on XAI](#) using Principal Component Analysis on the jet classifier model latent spaces show expressive discrimination (see also the [XAI talk in the Tuesday session](#))

And we see that samples with large EDL-based uncertainty (> 0.8) lie in the overlap region, where discrimination is the hardest (expected "I don't know" from the model!)

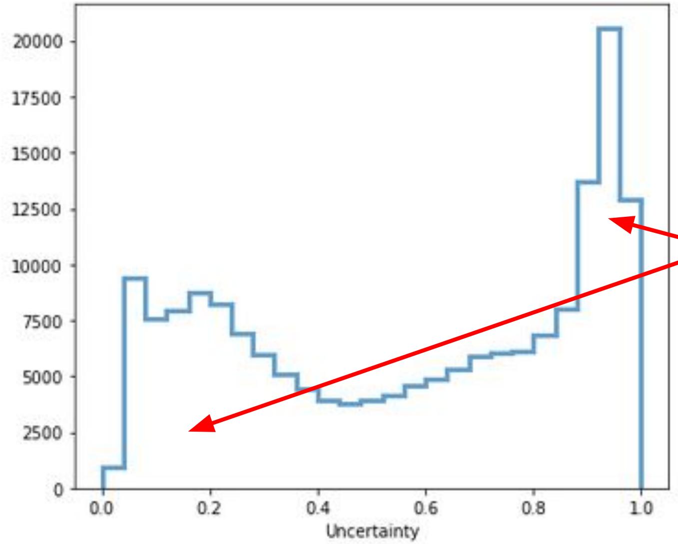


Who Gets Largest Uncertainties? *(cont.)*



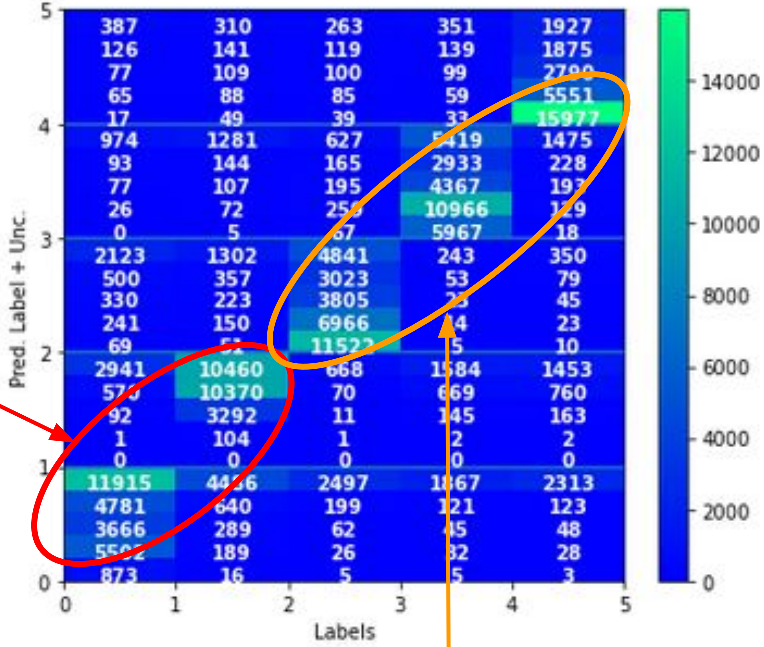
Uncertainties in Jet Tagging - II

- PFIN model applied to a multi-class problem with JetNet Dataset: distinguishing jets coming from: *light quarks* (0), *gluons* (1), *top quarks* (2), *W bosons* (3), *Z bosons* (4)



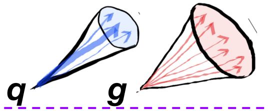
Bimodal distribution with a large peak at large uncertainties dominated by correctly classified quark and gluon jets

These jets have similar physical characteristics, and are hard** to tell apart



Heavier jets tend to have lower uncertainties

**but not impossible

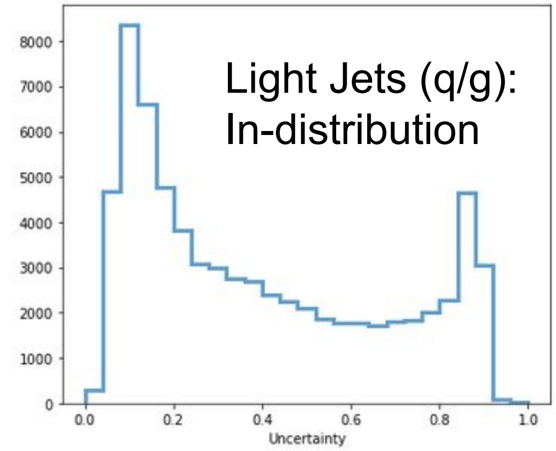


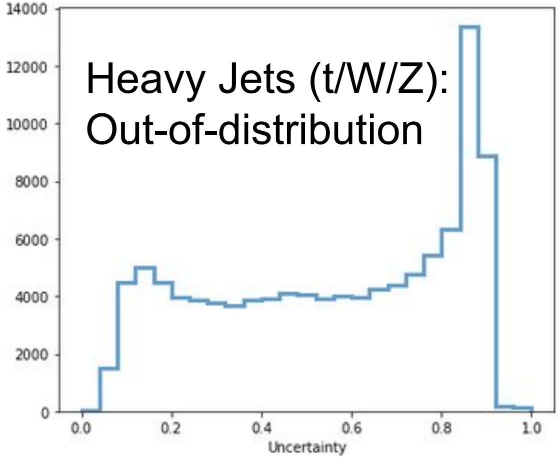
More constituents w/ more uniform energy fragmentation and wider

Studies on Anomaly Detection

Q: What happens if the models encounter jets that they have not "seen" before (i.e. trained on)?

- **Anomaly detection** with EDL can be tested by withdrawing some jet classes from training dataset
  - **In-Distribution** (ID): jets the model is trained on
  - **Out of Distribution** (OOD): jets withdrawn from training
- Models trained with EDL tend to assign a large "uncertainty" score to anomalous (OOD) classes
  - Model saying "hmmm…I don't know"
- One major challenge remains: how do we distinguish "hard-to-tell" jets from "anomalous jets" using a single uncertainty metric?

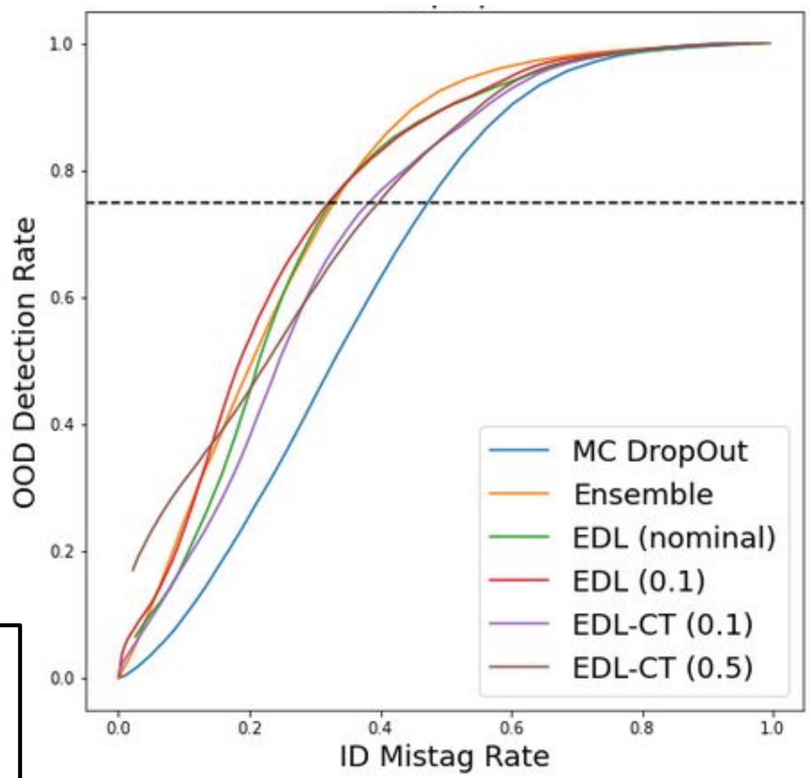




Comparing with Ensemble Methods

- Comparison can be done using ROC
  - A larger AUC would indicate a better performing model
- Key metrics:
  - ***OOD Detection Rate***: what fraction of OOD samples are correctly identified
  - ***ID Mistag Rate***: what fraction of ID samples are incorrectly identified

EDL shows equivalent performance to ensemble methods and better than MC Dropout



EDL-CT is a “Confidence Tuned” variant of the EDL method where the model is first allowed to converge w/o any annealing and then the parameters are tuned by retraining the model with annealing

Lessons Learned and Future Work

Evidential Deep Learning (EDL) involves training a deterministic NN to place uncertainty priors over the predictive distribution, requiring only a single forward pass to estimate uncertainty

The EDL approach to uncertainty estimation proved to be well calibrated on the Top tagger and JetNet datasets and was capable of detecting OOD samples

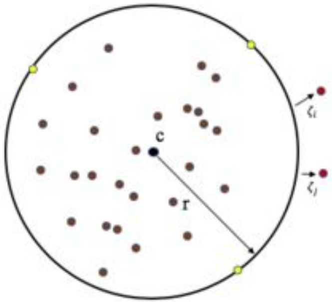
EDL shows equivalent performance to ensemble methods and better than MC Dropout

Challenge: No clear metric to differentiate between “it’s hard to tell” and “I don’t know”

Next steps:

- Try these methods on the Jet Class dataset
- Bind in together with **One Class Classifier Methods** (OCC), as the current approach only works when at least two training classes exist
- Differentiate between uncertain ID samples and anomalous samples
- Explore the XAI aspects: exploring latent spaces is a good place to start

OCC



OCC trained to project in-distribution events within a hypersphere of radius *c*