SLAC NATIONAL ACCELERATOR LABORATORY

Uncertainty Propagation & Estimation

in LArTPC Reconstruction Models

D. Douglas, A. Mishra,
K. Terao, D. Ratner

NPML 2024 – ETH Zurich

U.S. DEPARTMENT OF ENERGY | Stanford University

Overview

A Brief Introduction to Uncertainty Quantification

Reconstruction Study I:

In Which Energy Regression Uncertainty is Driven by Modelling Error

Reconstruction Study II:

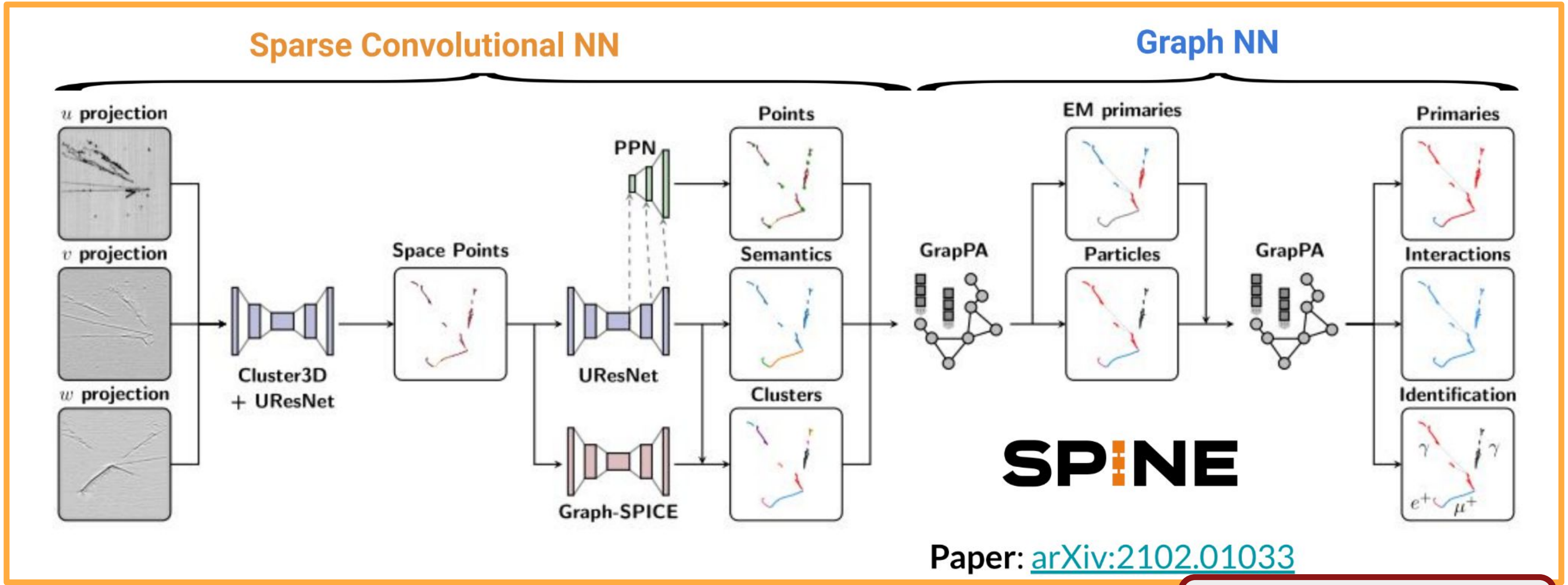
In Which Shower Fragment Association is Improved by Uncertainty Propagation

UQ & YOU

Sequential Models and Making Mistakes

D.H. Koh:

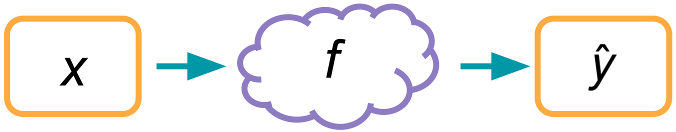
“We separate our model into a chain with distinct tasks and well defined intermediate representations so we can recognize cases more susceptible to error”

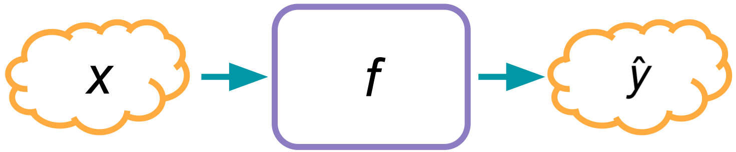


F. Drielsma *et al.*

Types of Uncertainty in Modelling

- **Aleatoric/Statistical** – arise from peculiarities of an individual input
  - Can be *homoscedastic* or *heteroscedastic*
  - Irreducible!





- **Epistemic/Systematic** - arise from the model itself – is the model capable of expressing the underlying process (do the *perfect* weights correspond to the "true" model)? How susceptible is the model to less-than-ideal training (Do the *actual weights* come close the the *perfect weights*)?

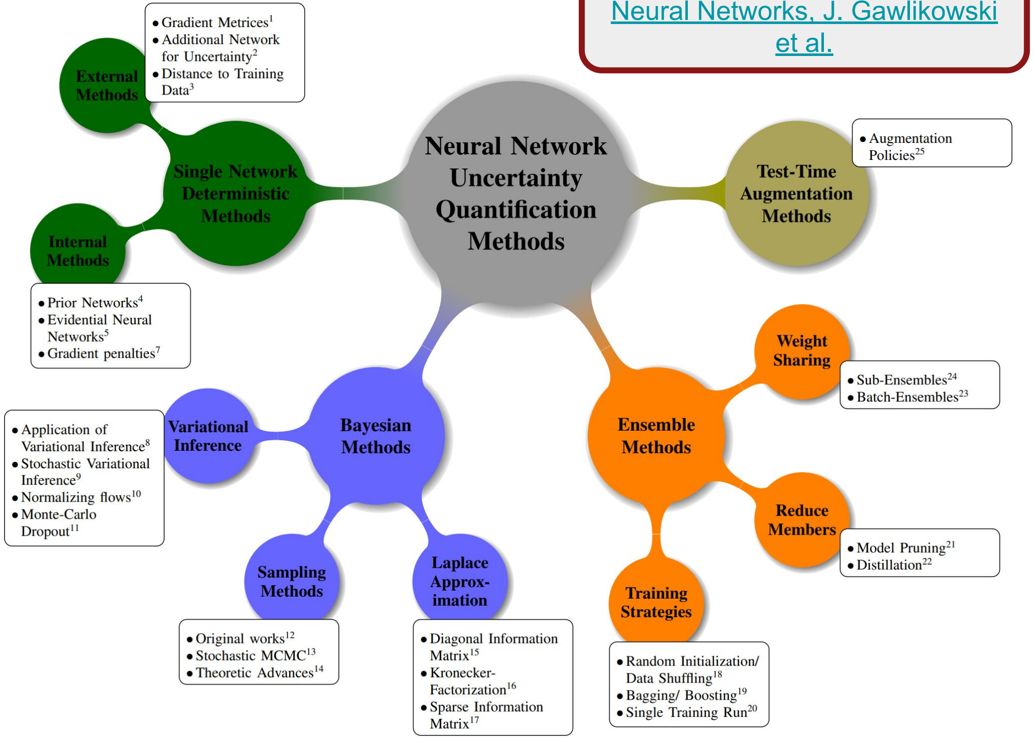
Uncertainty-Enabled Models

This is really about moving from predicting scalar values from scalar inputs to predicting distributions from distributions.

There is a **menagerie of methodologies** available to achieve this, though some techniques are better suited for some tasks than others.

Here, I'll discuss a few simple ones

[A Survey of Uncertainty in Deep Neural Networks, J. Gawlikowski et al.]

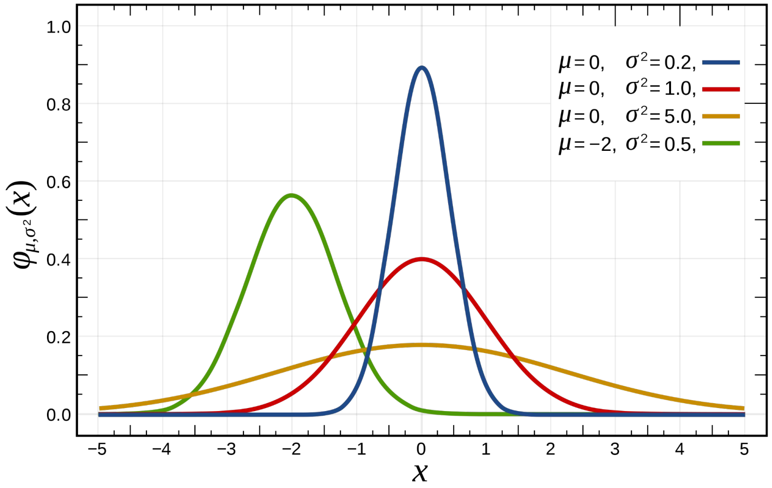


Probabilistic Predictions

The simplest method to implement is sometimes enough

Make your model predict parameters of a distribution and use a likelihood-based loss.

Hint: if you're using MSE, you're already maximizing a likelihood with a homoscedastic model!



predicted

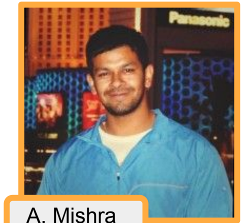
$$\log \mathfrak{L} = \sum_{i \in \text{voxels}} -\frac{1}{2}\frac{(\boxed{\hat{E}_i} - E_i)^2}{\boxed{\hat{\sigma}_i^2}} - \log(\boxed{\hat{\sigma}_i})$$

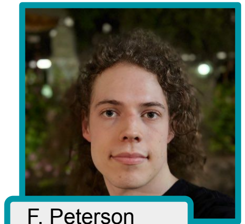
Stable Distribution Propagation

A scalable way to propagate input uncertainties with large models – works for Gaussian & Cauchy PDFs

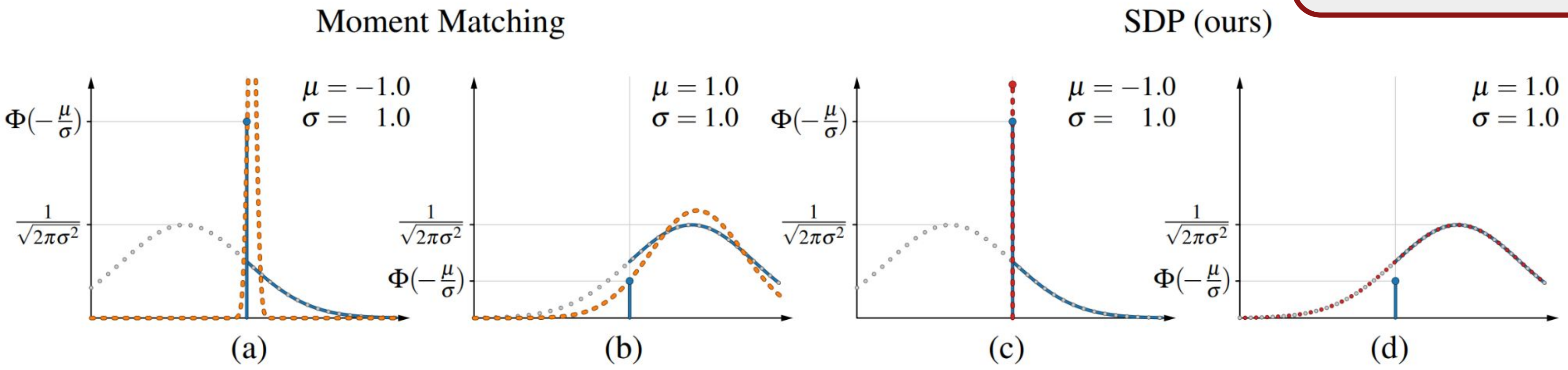
[distprop github]

$$\Sigma_y = J(f) \cdot \Sigma \cdot J(f)^\top$$


A. Mishra


F. Peterson

[Uncertainty Quantification via Stable Distribution Propagation, F. Peterson, A. Mishra, et al.]

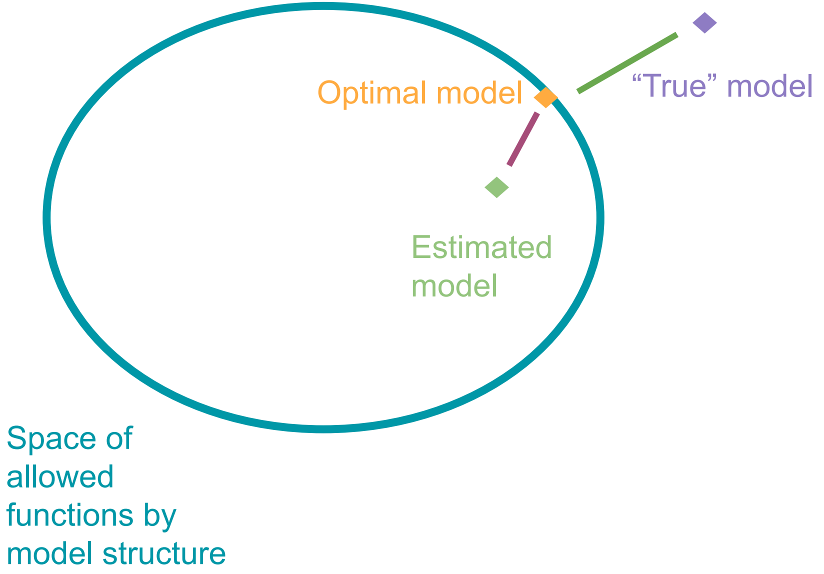


Model Uncertainty

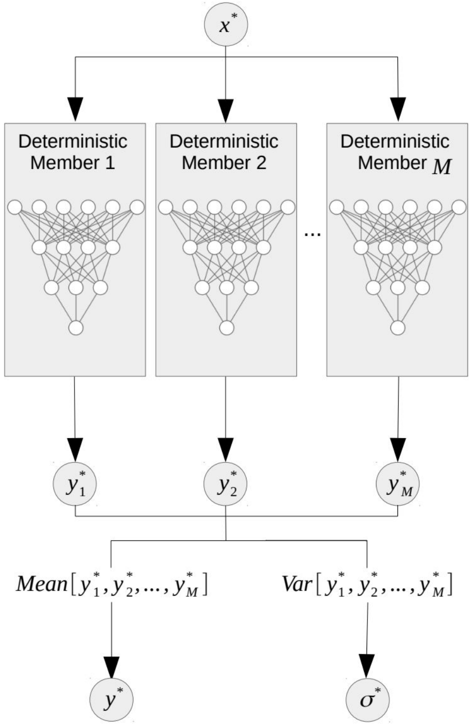
Consider that the family of functions defined by our model may be quite far away from the “true” model

Model error arises from inability for chosen model to express the “true” model (**approximation error**)

More model error arises from error in finding the optimal model (**estimation error**). This stems from algorithmic errors, sampling error within training data, etc.



Ensembling Methods



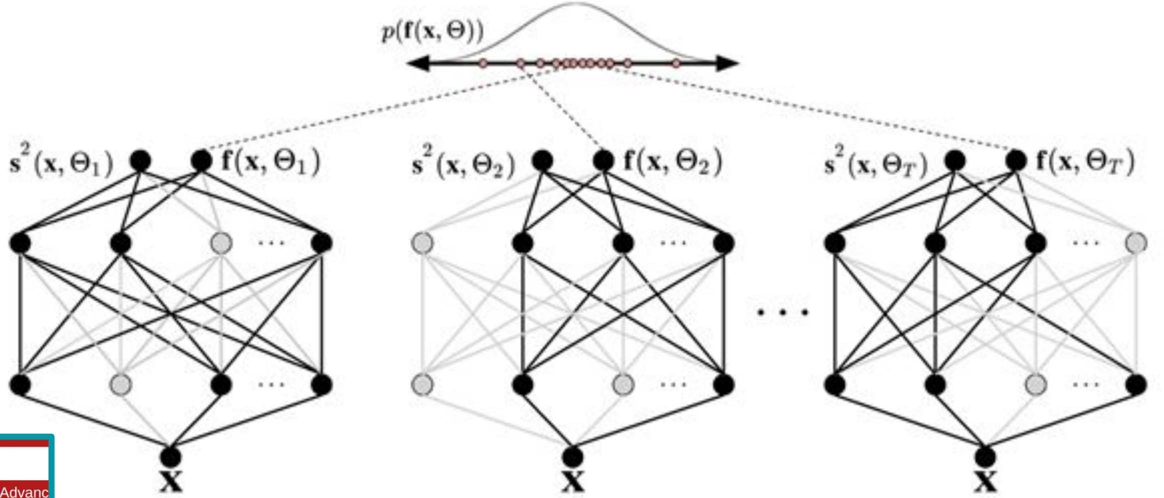
To quantify **estimation error**, we can train the same model with independently estimated weights, and use the distribution of predictions produced by these models.

For a regression problem, we can simply use the mean and variance of the model mean predictions.

For a probabilistic network, the ensemble inference can be treated as a gaussian mixture model.

MC Dropout: the Lazy Person's Ensemble

Monte Carlo Dropout is a method for stochastically changing your model in order to approximate a posterior distribution of a model's prediction




arXiv > stat > arXiv:1506.02142

Search...

Help | Advanc

Statistics > Machine Learning

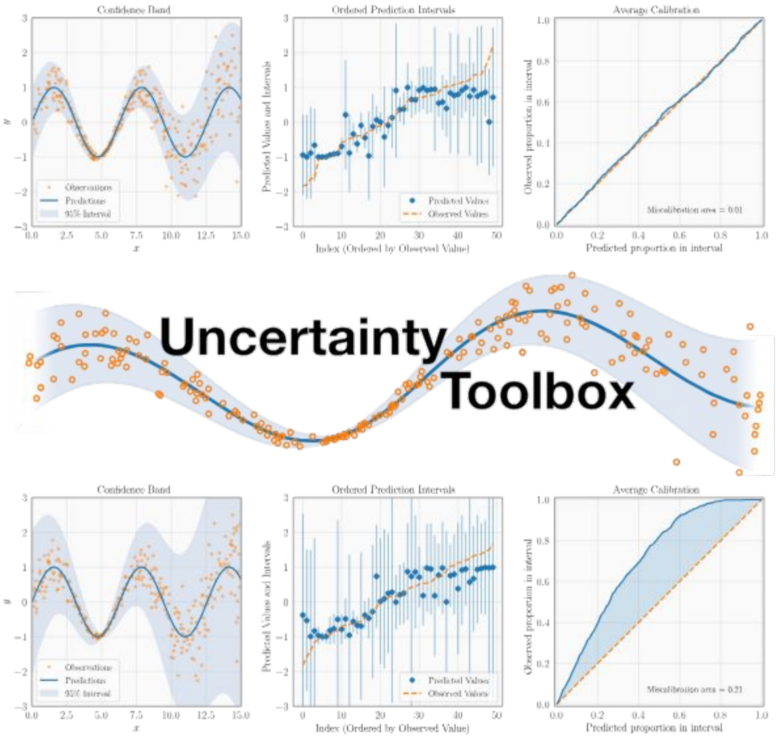
[Submitted on 6 Jun 2015 (v1), last revised 4 Oct 2016 (this version, v6)]

Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

Yarin Gal, Zoubin Ghahramani

[Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning, Y. Gal, Z. Ghahramani]

UQ Tools!



Public – and hackable! – tools are available in Uncertainty Toolbox

Contains useful plots, metrics, and methods for doing UQ

Adversarial mis-calibration assessment, re-calibration and many other tools

Uncertainty Toolbox - github

W. Neiswanger

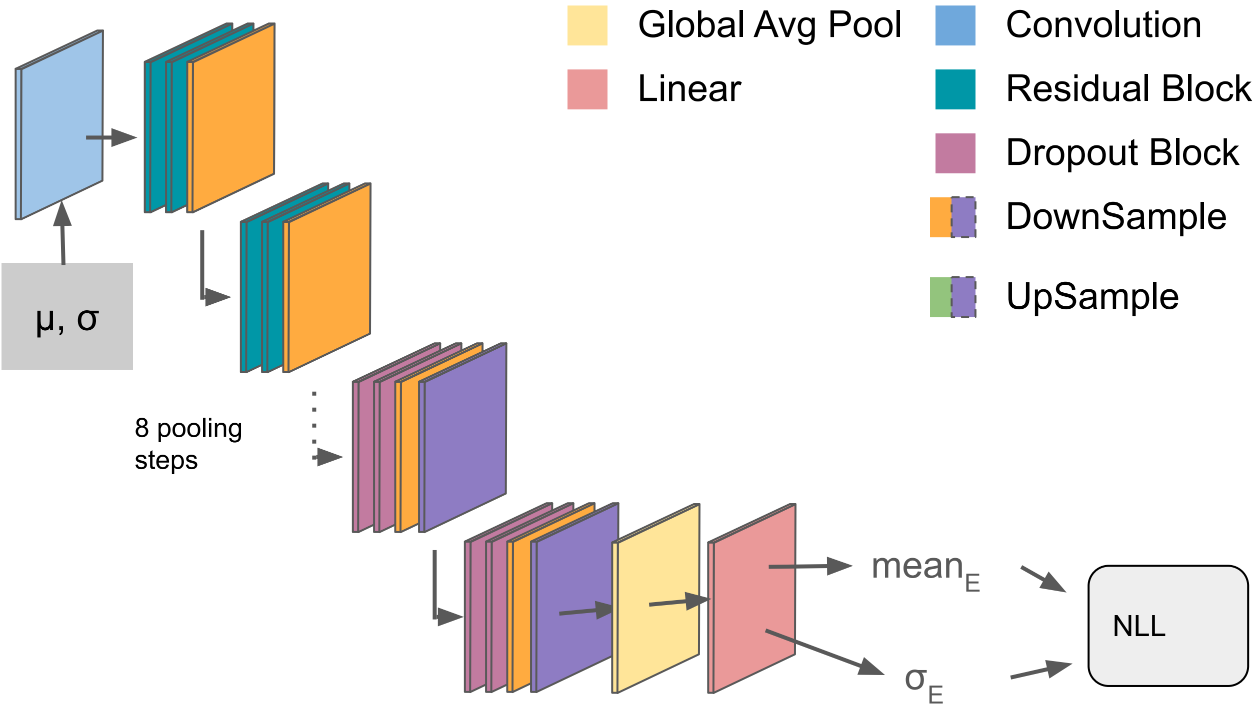
Case Study I:

Point Cloud Image
Shower Energy Regression

Whole Image Primary Energy Regression

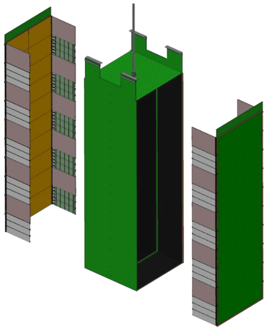
Simple ResNet encoder with dropout layers

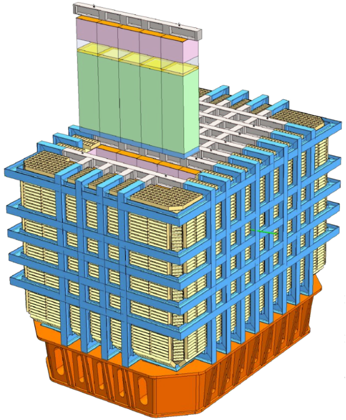
Deterministic (no sigma output) and Probabilistic versions

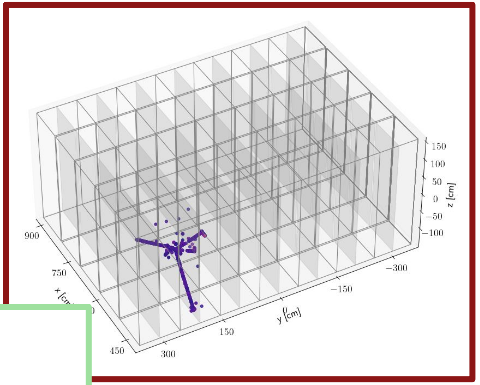


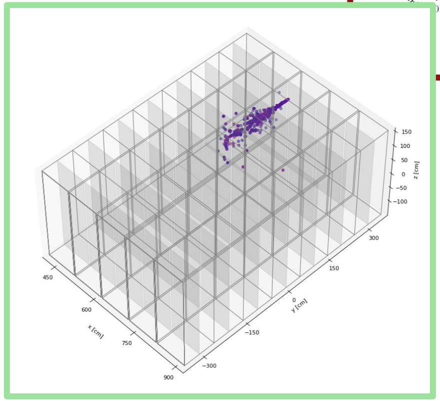
Training Set - DUNE ND LArTPC Simulation

For this task, we use electron and gamma-initiated showers (0-1 GeV) simulated in a DUNE ND-LAr-like detector





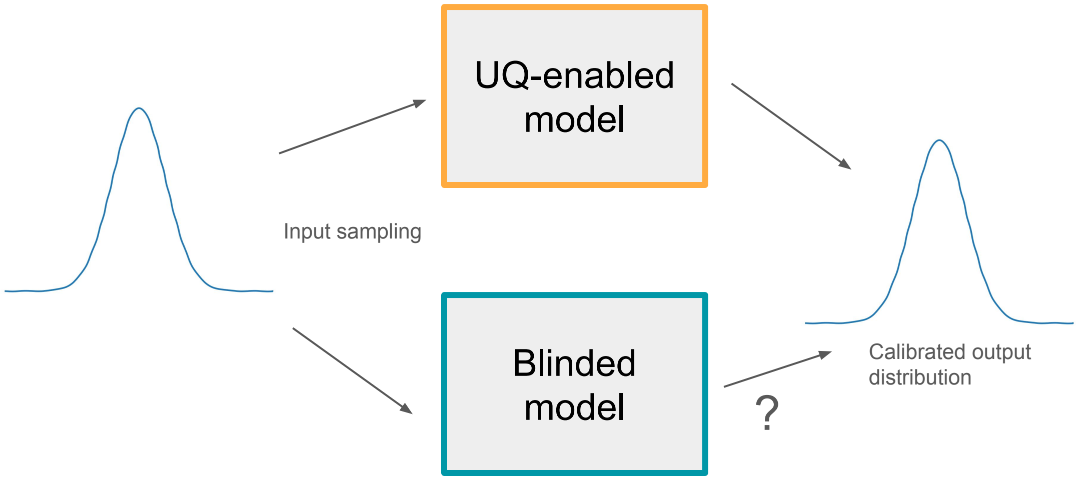




A Minimally-viable Aleatoric/Input UQ Problem

Question: Can a model *without* explicit uncertainty information learn the underlying uncertainty distribution from variance seen in training input?

This helps to understand a model's sensitivity to input-by-input uncertainty vs. the systematic error of the model



Adding Artificial Input Noise

$$\sigma_i = \alpha_i f_i$$

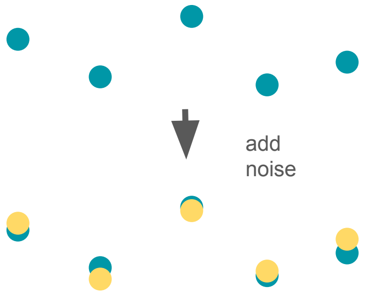
$$\alpha_i \sim U[0.02, 0.05]$$

$$\tilde{f}_i = f_i + \beta_i \sigma_i$$

$$\beta_i \sim N[0, 1]$$

Now, add a new feature for every existing feature which is 2-5% of the feature's true value.

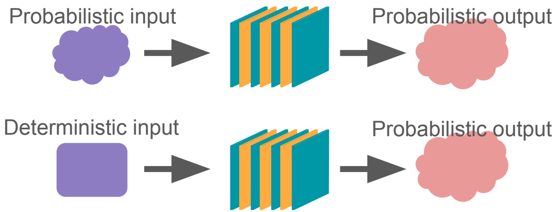
The true value is then smeared by a gaussian with this width, giving a noisy feature and a calibrated input uncertainty



Performance

Both models perform with similar accuracy

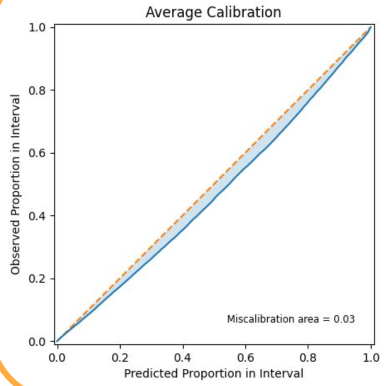
UQ-enabled model is slightly less accurate, but produces better uncertainty estimates

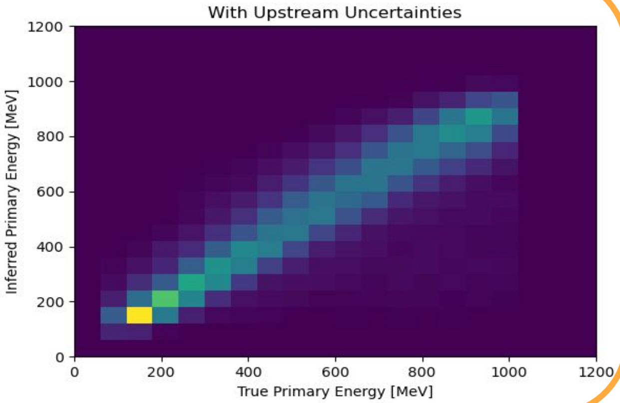


MCA - 0.034

MAE - 14.795

NLL - 4.206

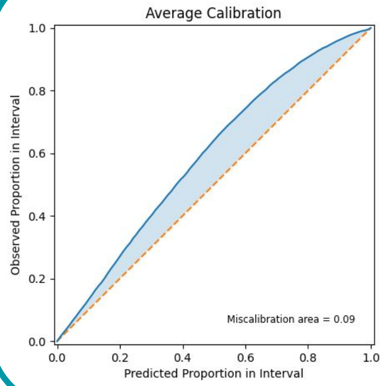


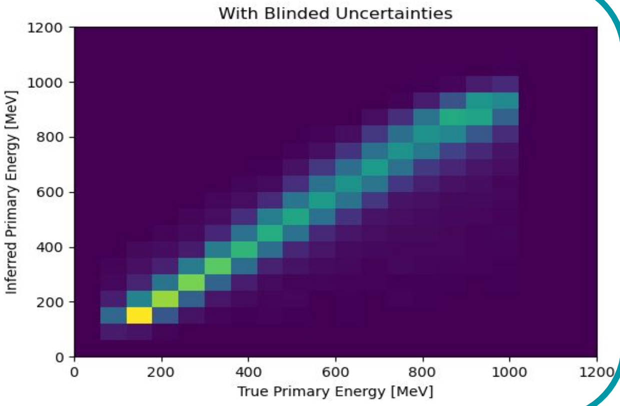


MCA - 0.092

MAE - 13.053

NLL - 4.129





Case Study II:

Shower Fragment Association

Track Cluster Association Task

Data for this task is taken from an SSI challenge

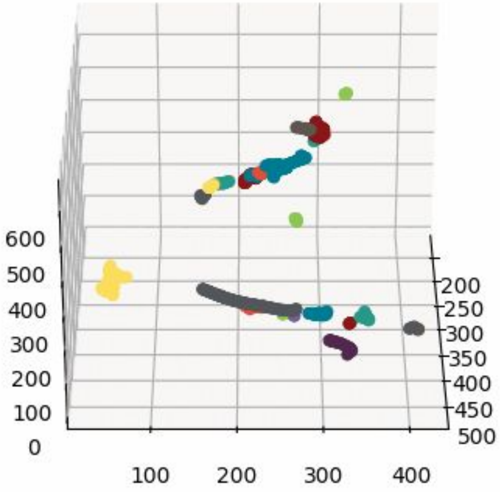
Inputs are nodes defined from disconnected fragments of showers in LArTPC simulation

There are multiple showers within each image

Each node is labelled by whether it is upstream or downstream

Edges are labelled by the shower association

Shower fragments



Track Cluster Association Task

- Primary fragment
- Secondary fragment

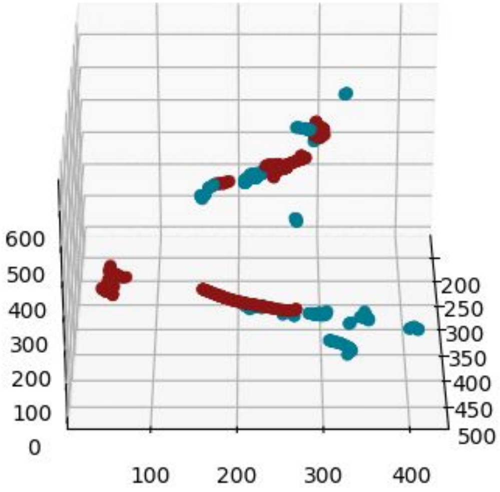
Data for this task is taken from an SSI challenge

Inputs are nodes defined from disconnected fragments of showers in LArTPC simulation

There are multiple showers within each image

Each node is labelled by whether it is upstream or downstream

Edges are labelled by the shower association



Track Cluster Association Task

Data for this task is taken from an SSI challenge

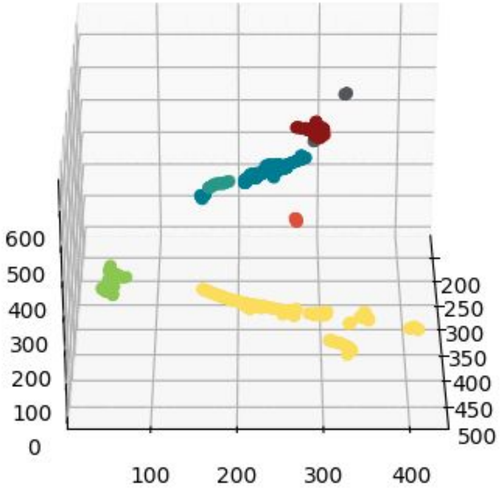
Inputs are nodes defined from disconnected fragments of showers in LArTPC simulation

There are multiple showers within each image

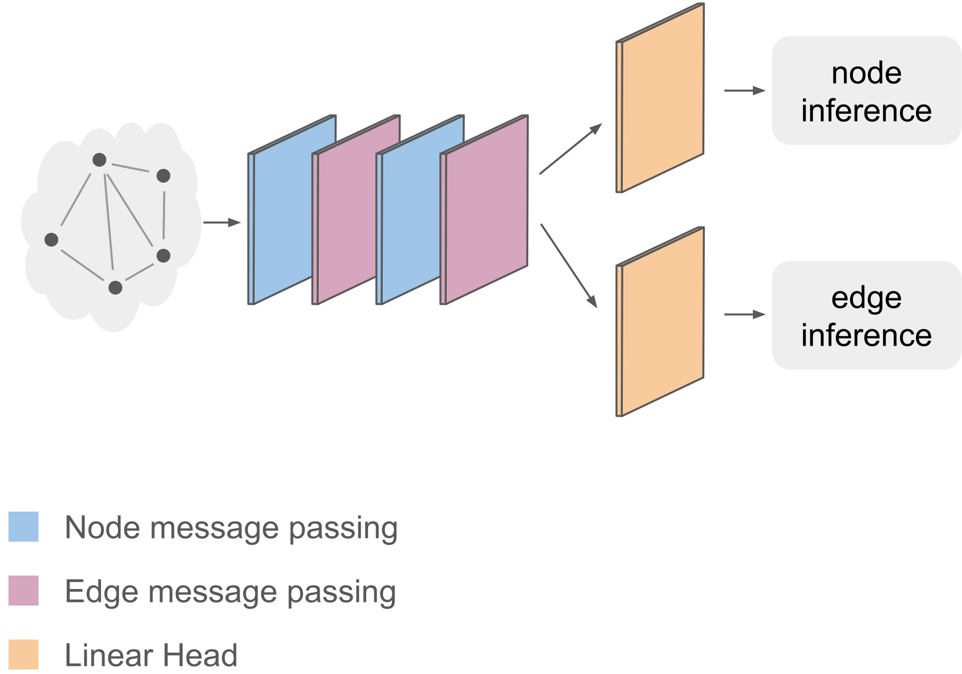
Each node is labelled by whether it is upstream or downstream

Edges are labelled by the shower association

Complete showers



Network Architecture



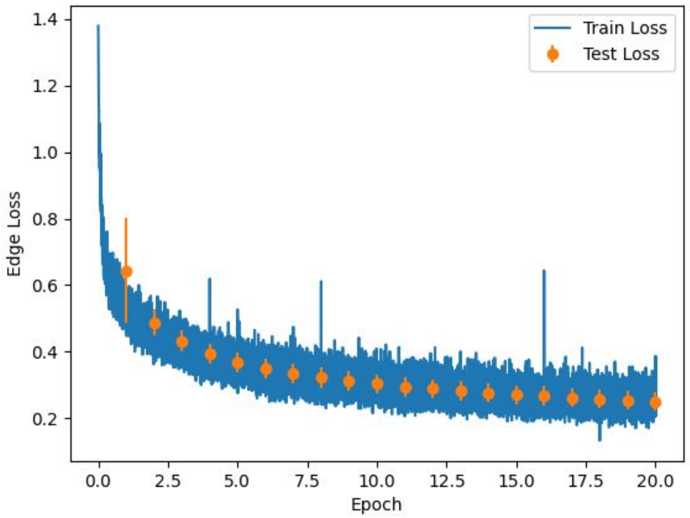
Simple EdgeConv network with two MLP heads for node and edge classification

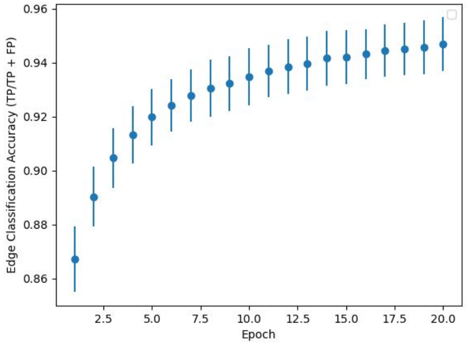
Node Classification for upstream/downstream identification

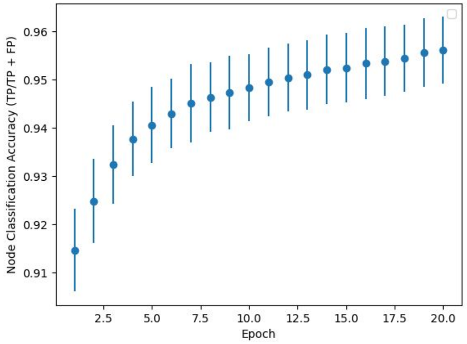
Edge Classification for association of disconnected fragments

Full graph partitioning isn't considered in this task (yet!), only edge-by-edge classification

Loss – Training without Noise





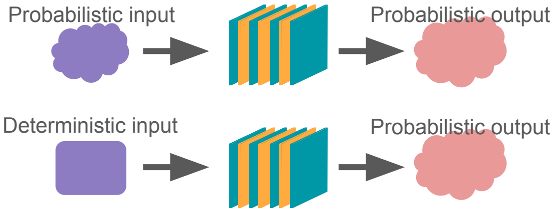


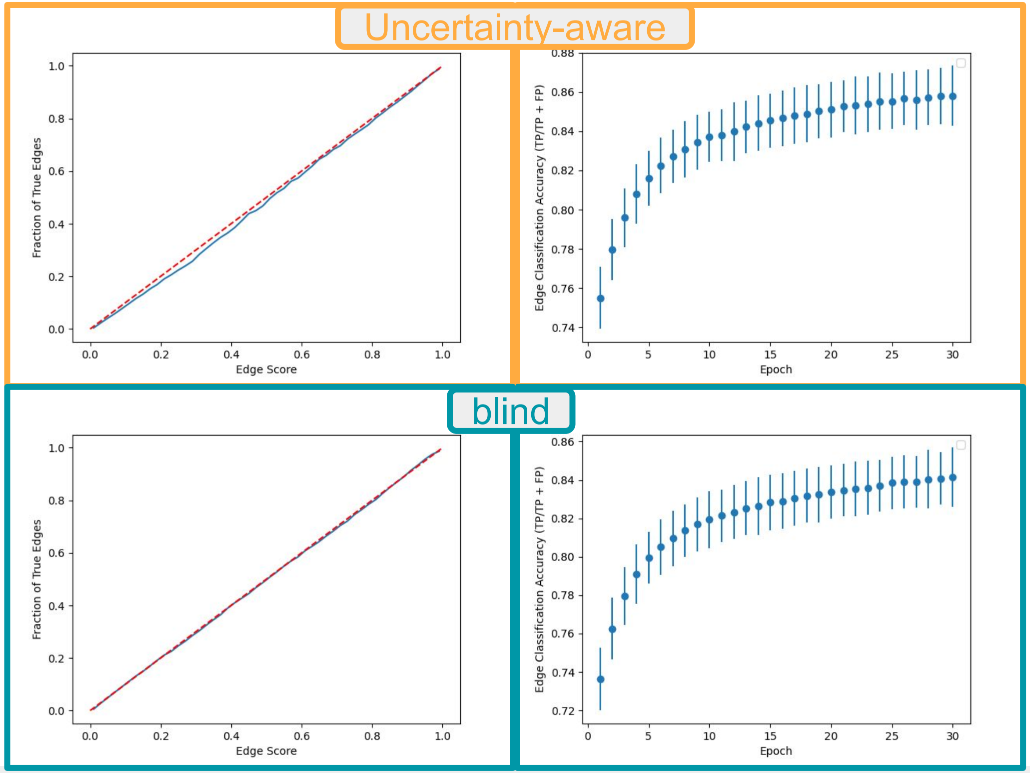
Reaches about 95% accuracy at both edge and node classification tasks

Performance – Uncertainty-Enabled and Blinded

Both models are exceptionally well-calibrated out of the box

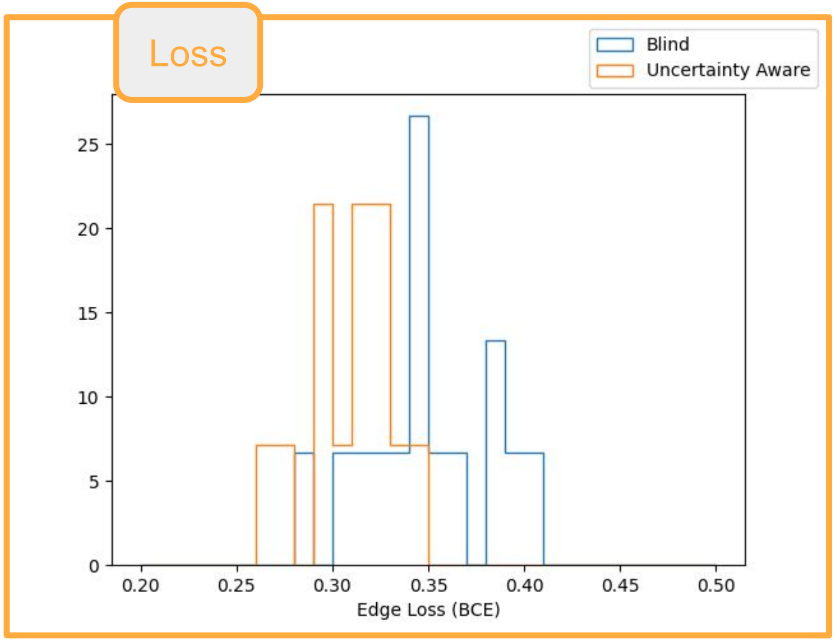
The uncertainty-aware model produces ~3% more accurate inferences

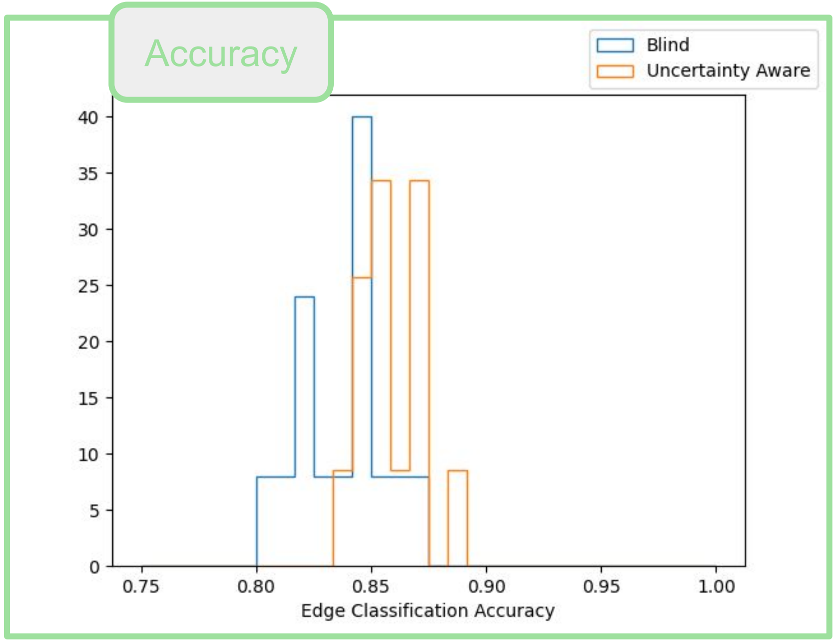




Ensemble Metrics

See a consistent improvement across a (relatively small) ensemble of models!

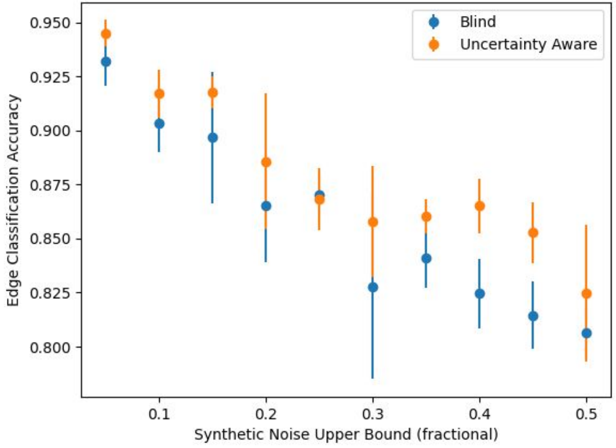




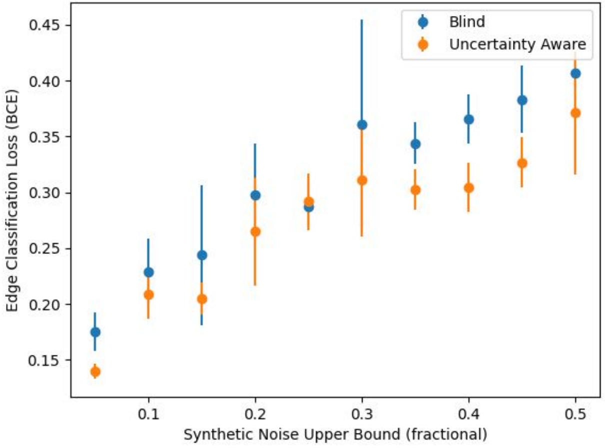
Performance as a Function of Noise Magnitude

The uncertainty-aware model performs consistently better at edge classification (~3-5% more accurate) across a wide range of input uncertainty!

Accuracy



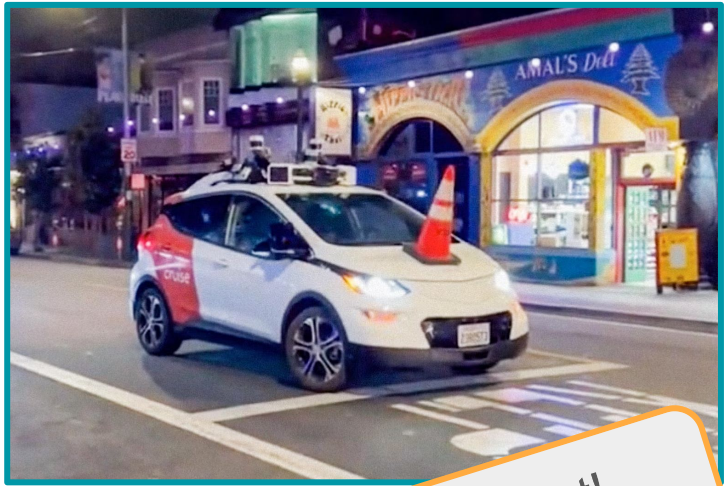
BCE



$$\sigma_i = \alpha_i f_i \qquad \alpha_i \sim U[0.05, x \quad]$$

Conclusions

- A short survey of two very different reconstruction tasks in LArTPC reconstruction and how to quantify per-inference uncertainty
- Some tasks are much less sensitive to input uncertainty, instead they are dominated by model error
  - This will inform a strategy towards full end-to-end propagation in a large model like SPINE
- Uncertainty Quantification is important! An "accurate" deterministic model may inspire overconfidence and blindness to anomalous inferences!



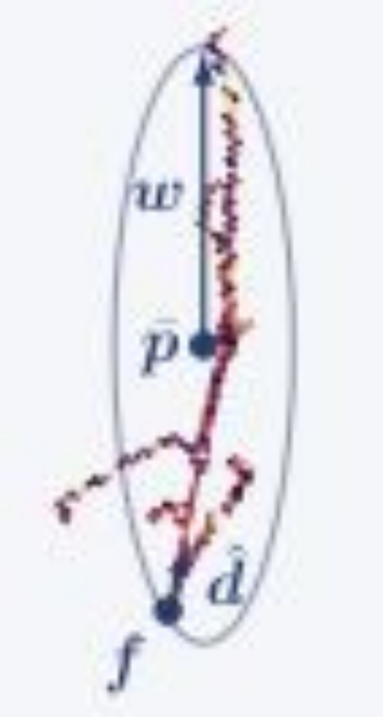
Be confident!
Not too confident…

BACKUP

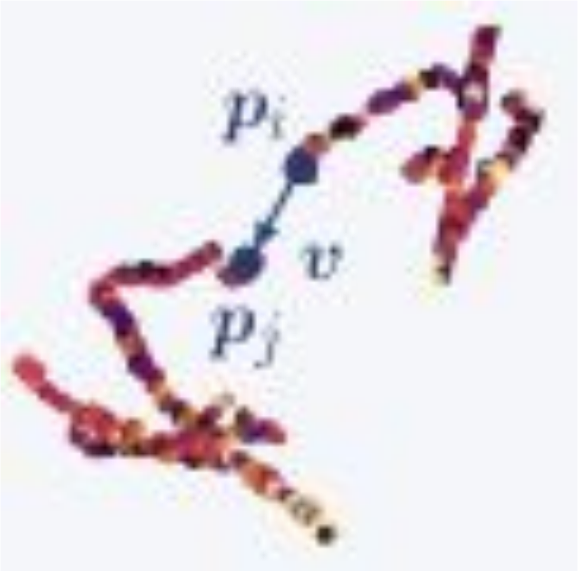
Feature Definitions

Node features:

- Fragment voxel centroid (3)
- Fragment voxel covariance matrix (9)
- Fragment voxel principal axis (3)
- Number of voxels in the fragment (1)



Feature Definitions



Edge features:

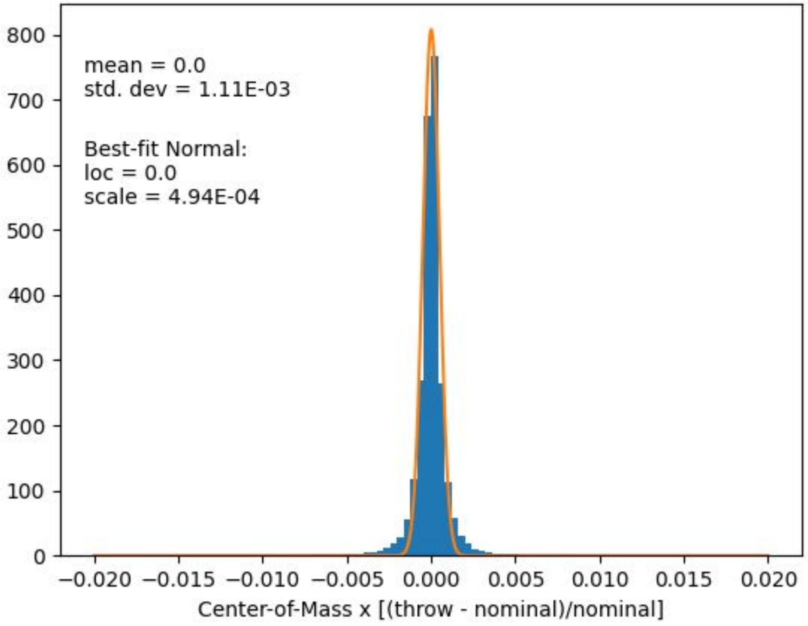
- Closest points of approach (6)
- Displacement vector between closest points of approach (3)
- Outer product of displacement (9)
- Norm of displacement (1)

80% Dropout Feature Distribution

Node Features

Not quite gaussian

Characterize distribution width as a function of dropout efficiency

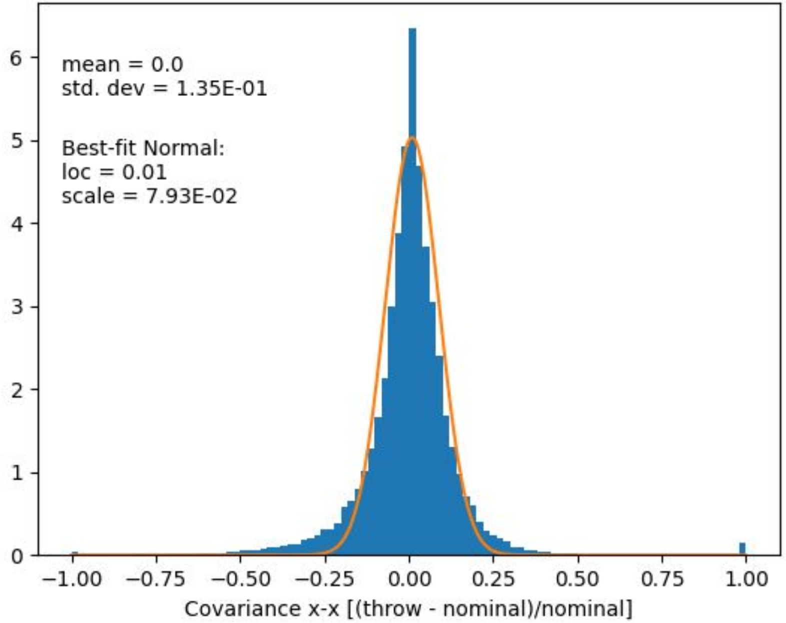


80% Dropout Feature Distribution

Node Features

Not quite gaussian

Characterize distribution width as a function of dropout efficiency

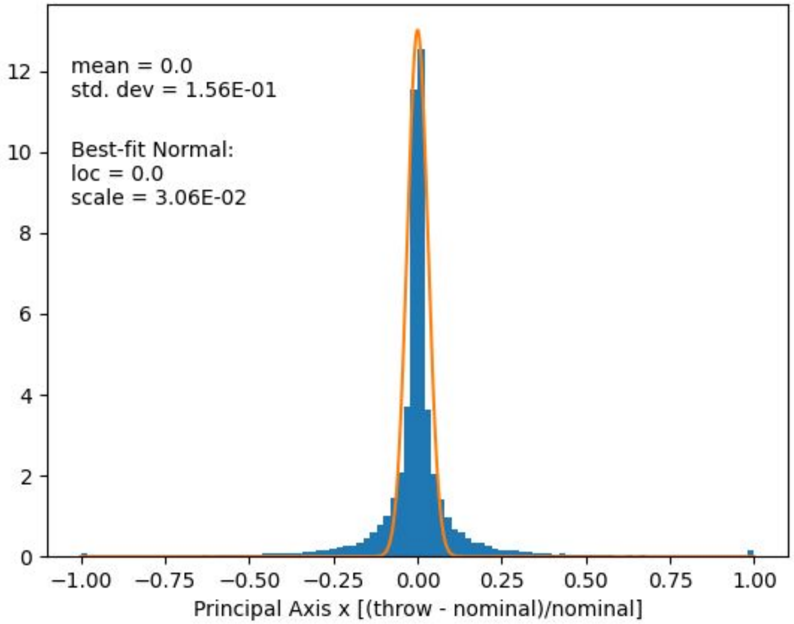


80% Dropout Feature Distribution

Node Features

Not quite gaussian

Characterize distribution width as a function of dropout efficiency

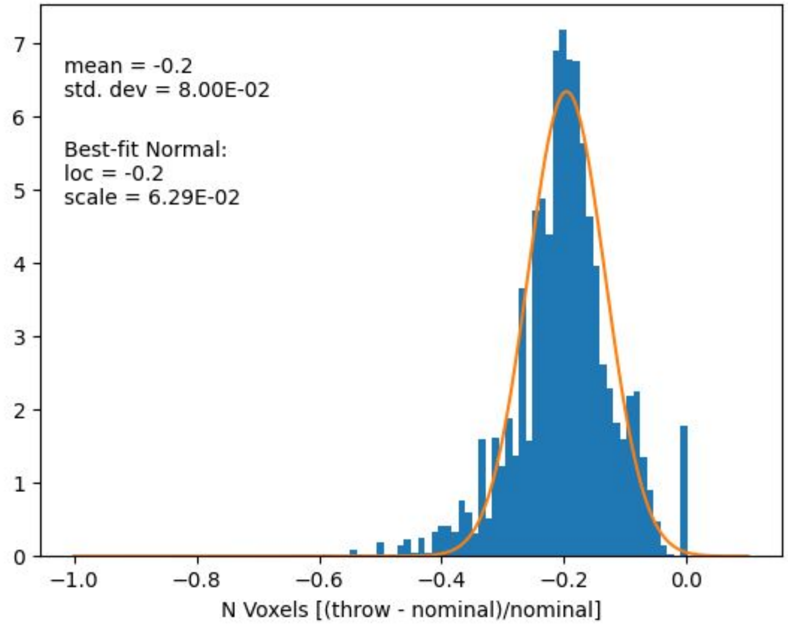


80% Dropout Feature Distribution

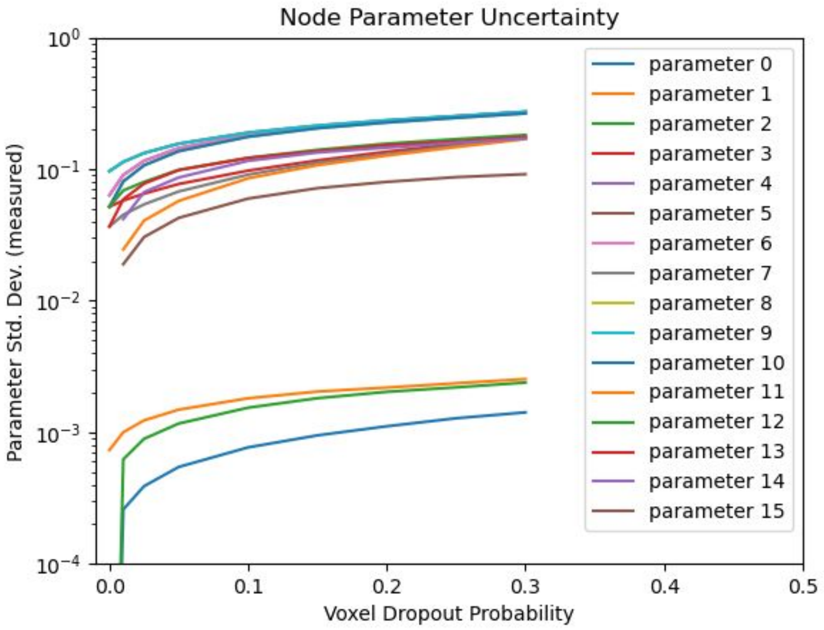
Node Features

Not quite gaussian

Characterize distribution width as a function of dropout efficiency



Node Parameter Uncertainty Scaling



Some features have as much as 25% uncertainty with high dropout, but most are in the <10% region

For the spatial parameters, fractional uncertainty throws are not the best approach