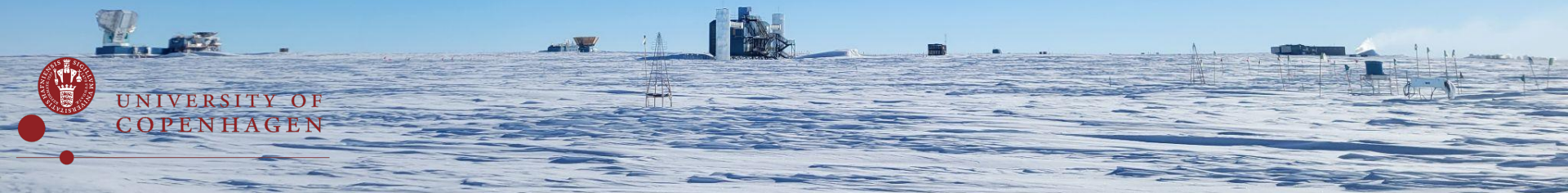
Neutrino/antineutrino separation for the IceCube Upgrade detector using graph neural networks

Applied Machine Learning 2025 final project

Johann Nikolaides & Marc Jacquart

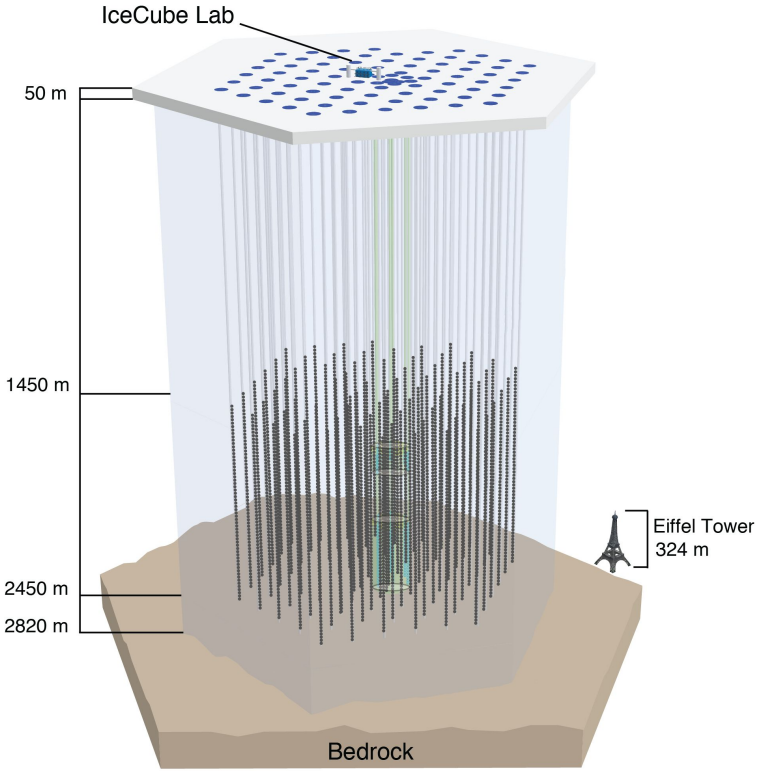


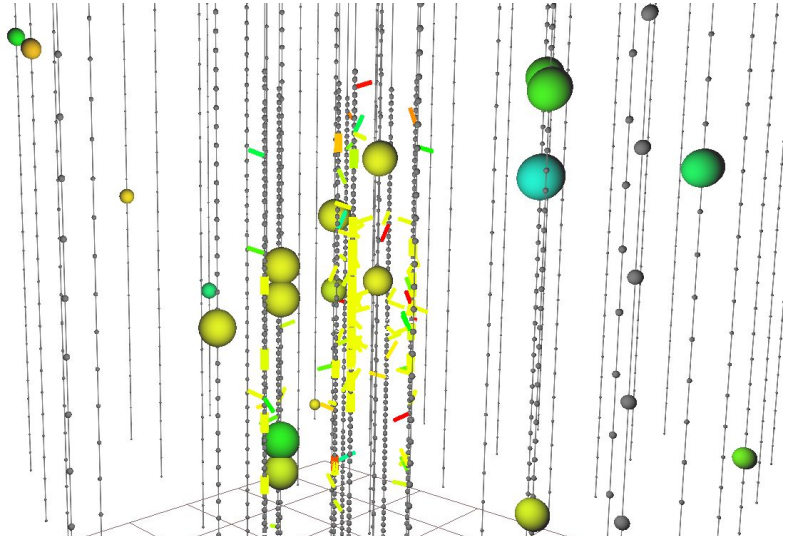
UNIVERSITY OF COPENHAGEN

Disclaimer:

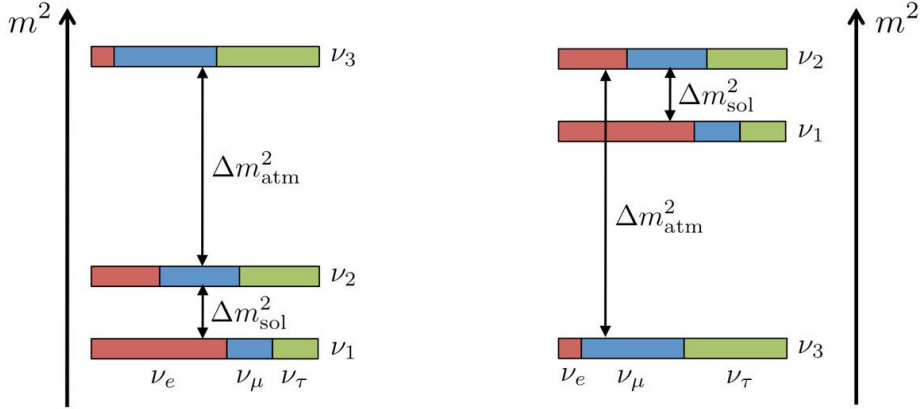
Because of approximations and time constraints tied to this final project, these results are **not** representative of any actual capability of the IceCube detector.

Motivation





Normal ordering (NO) Inverted ordering (IO)

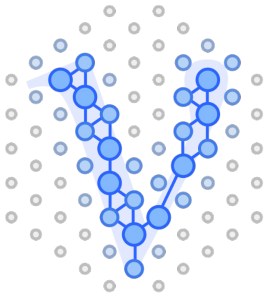


Dataset & tools

IceCube Upgrade MC simulations: 2.7M ν, 1.3M $\overline{\nu}$

GraphNeT: Pytorch-based framework for neutrino telescopes

~24h training time on 1 GPU (NVIDIA GeForce RTX 3070)



GraphNeT

Deep Learning for Neutrino Telescopes

PyTorch

Pre-processing

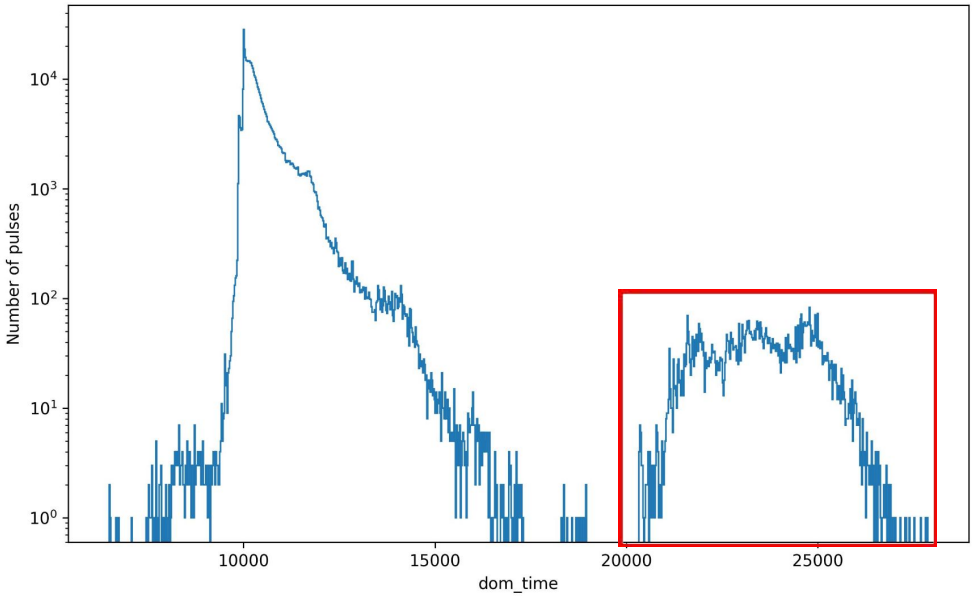
Features: IceCube Module hit (x,y,z,t)

Outlier time values

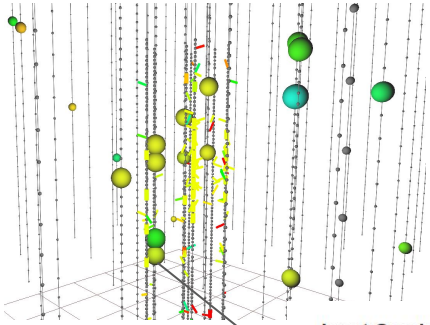
→Subtract arbitrary time offset

Rough normalization:

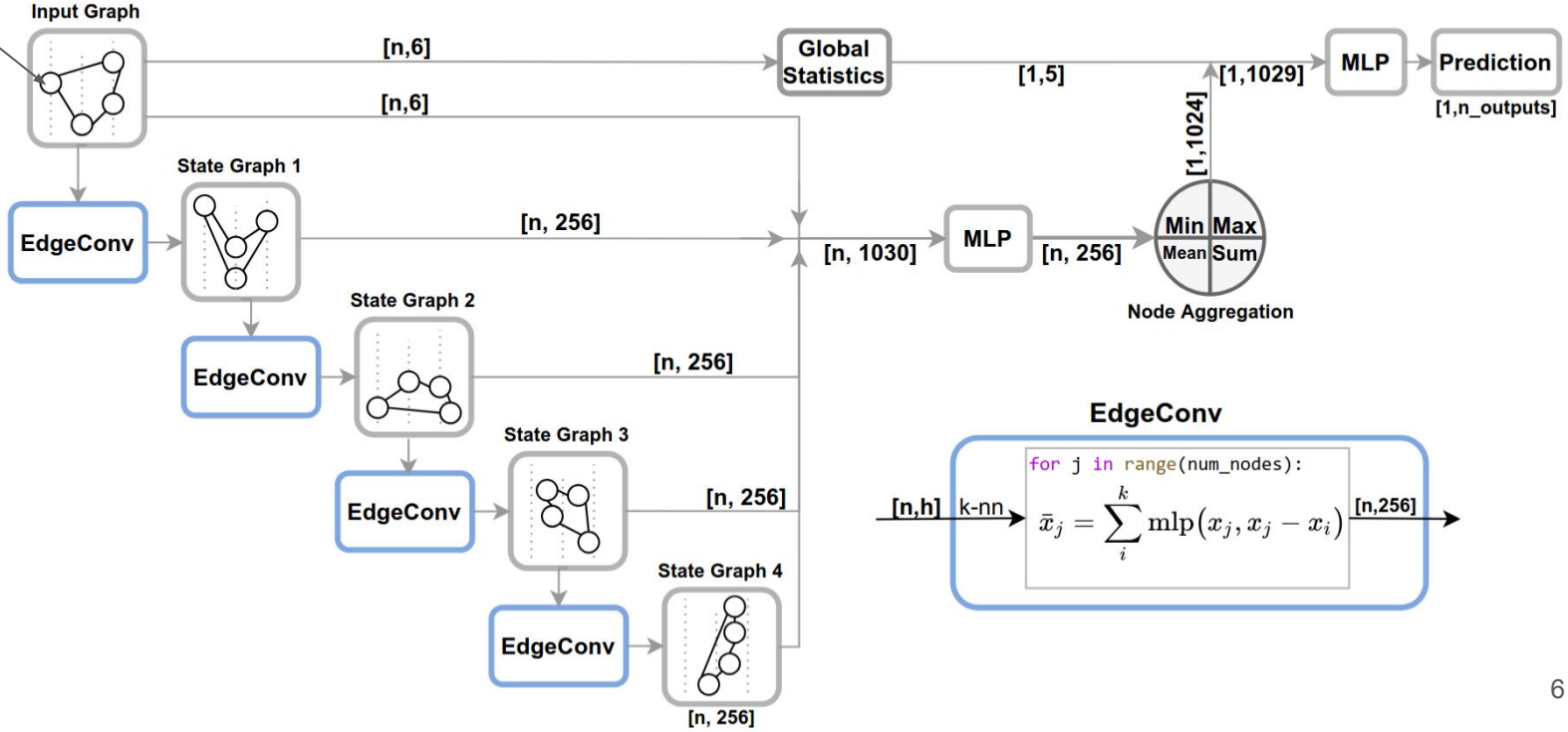
x,y,z/500 t/3000



DynEdge model



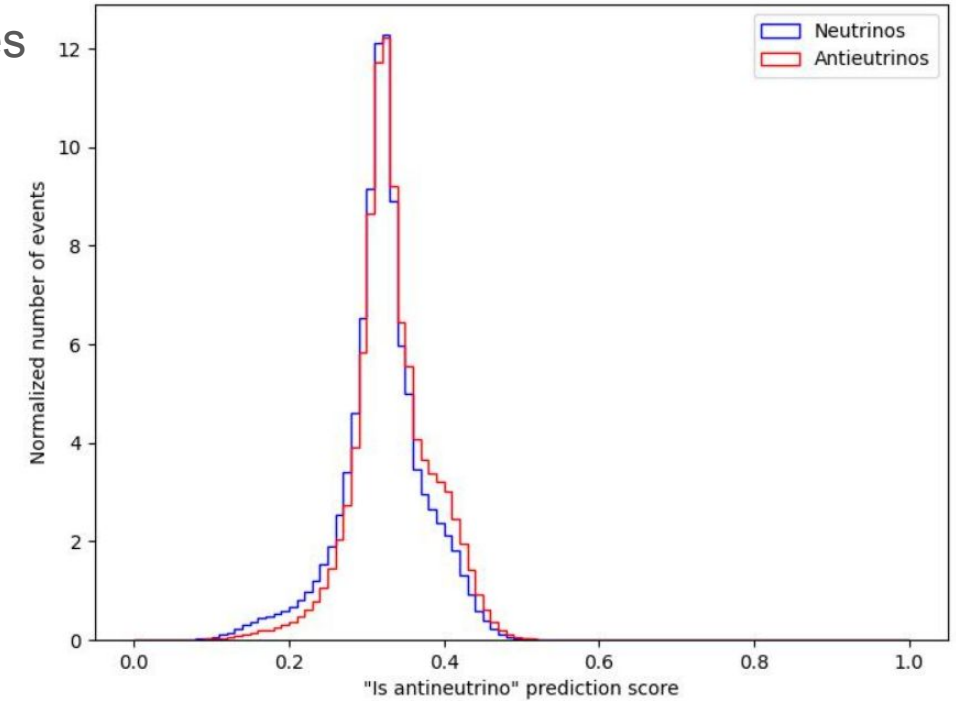
8 nearest neighbors



Binary classification

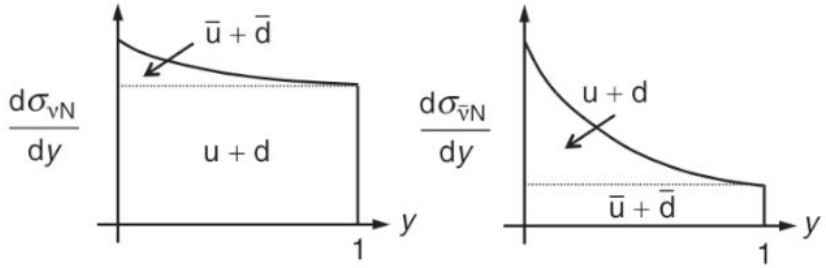
Binary cross-entropy score: $\mathrm{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right]$

Weighted events to balance $\nu/\overline{\nu}$ classes



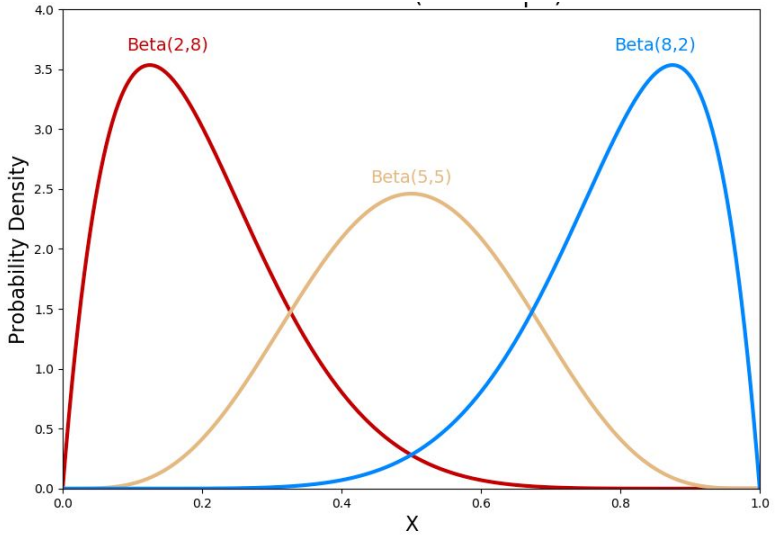
Modified data: not actual IceCube results

Inelasticity regression



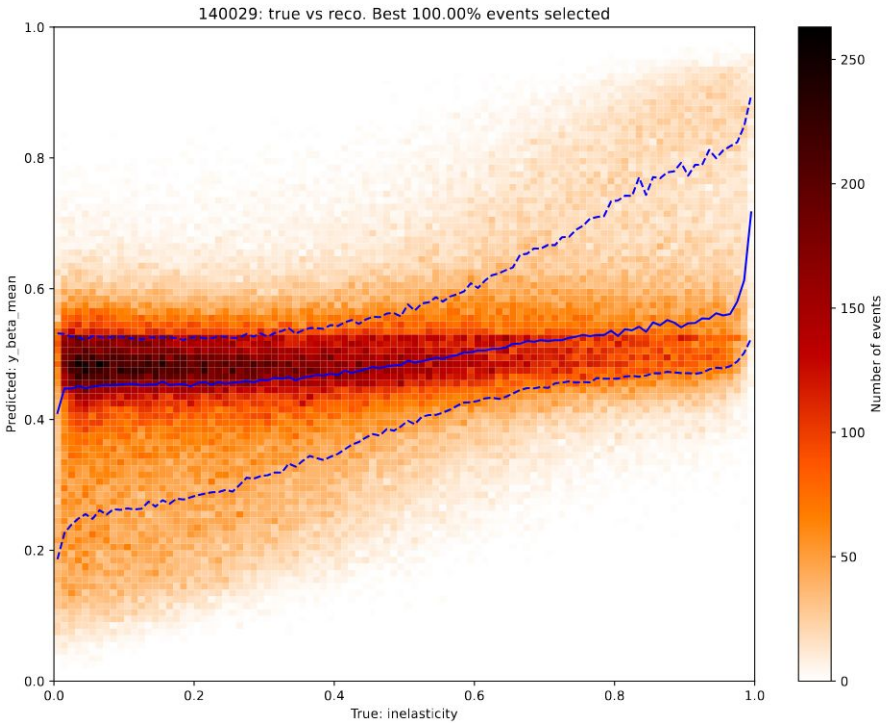
Statistical $\nu/\bar{\nu}$ separation

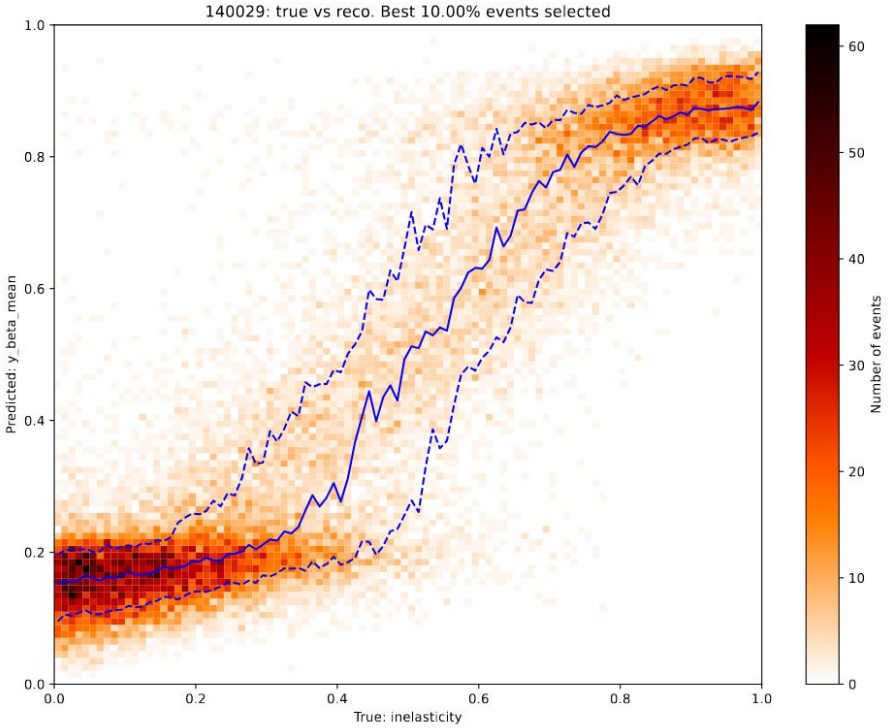
$$\text{Beta loss} = \frac{1}{N}\sum_{i=1}^{N} -w_i \, ln(\text{Beta}(y_{\text{true,i}}; \alpha_{\text{pred},i}, \beta_{\text{pred},i}))$$



Beta distribution likelihood for uncertainties predictions

Inelasticity reconstruction results





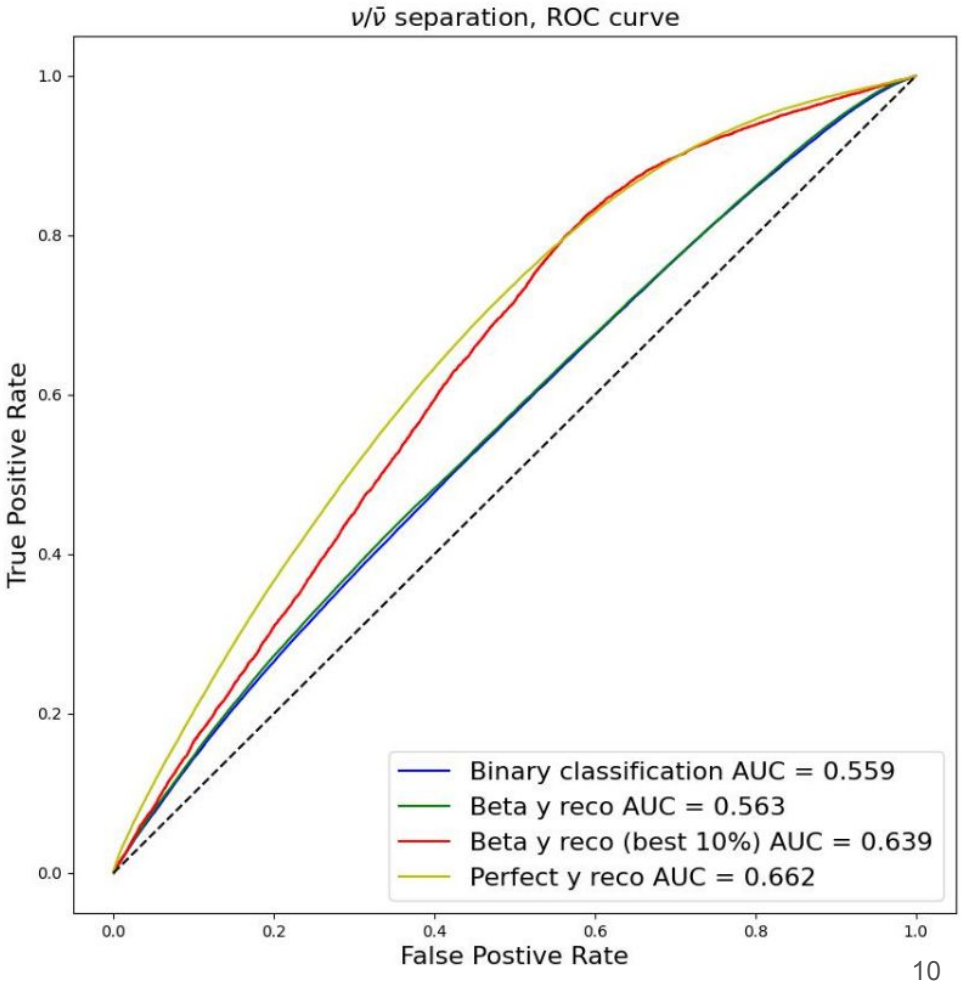
Modified data: not actual IceCube results

Results

- Binary classification ≈ inelasticity
- Model uncertainty prediction works well
- Trade-off statistics vs reco precision

Even a small $\nu/\overline{\nu}$ separation can have a significant impact for the NMO measurement!

Lesson learned:
Silent code failures are hard to track.
Make the code fail fast (and loud)!



Modified data: not actual IceCube results

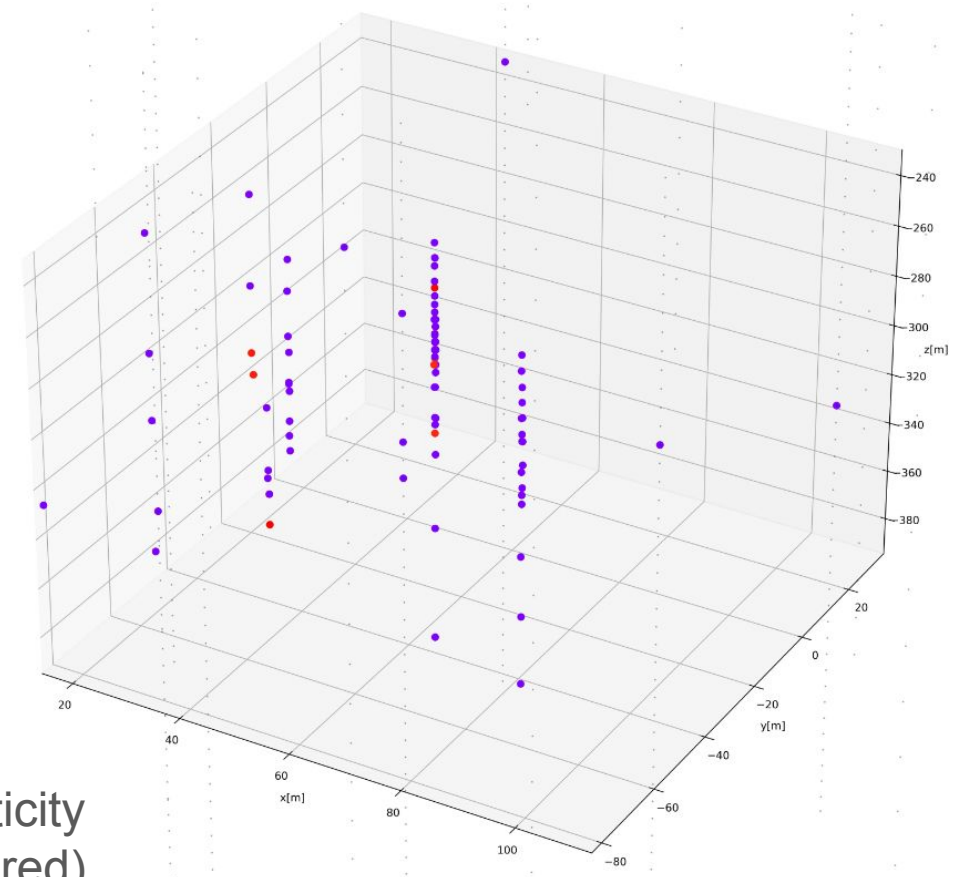
Feature & node importance

GNNExplainer package

Event topology varies a lot

Hard to interpret “per event”

→Future work



10 most important hits for inelasticity prediction (in red)

Thank you! Any questions?

Backup slides

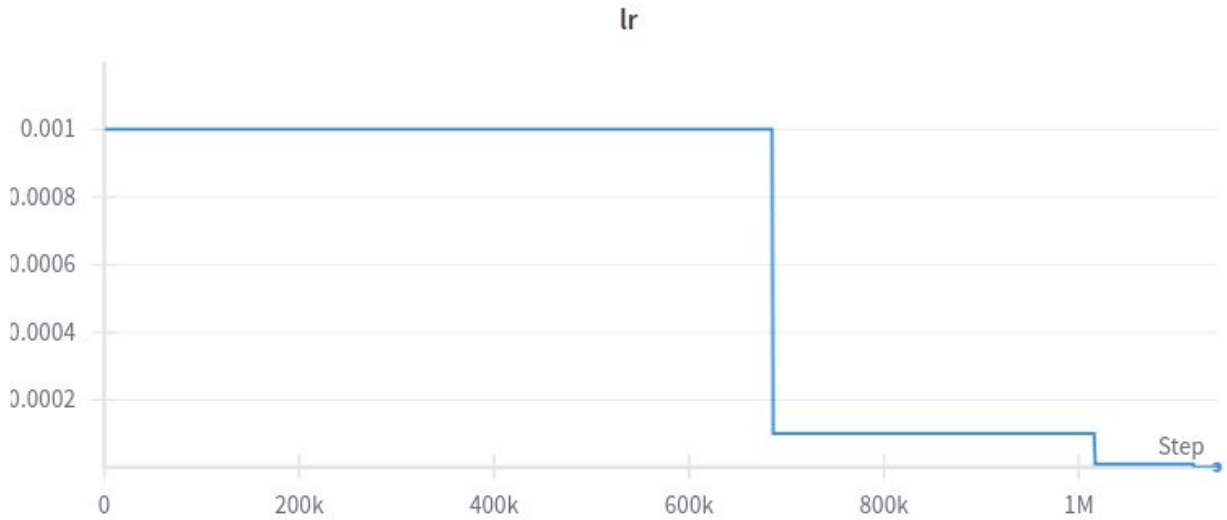
Hyperparameter optimisation

Because of the long (24h) training time and high number of hyperparameters, a rough manual search for hyperparameter optimisation has been done:

- Low energies or containment cuts improves inelasticity reconstruction
- Directionality of PMTs, quantum efficiency, DOM type feature have no impact
- DynEdge with default parameters seems well optimised for inelasticity reco

Learning rate scheduler

Start lr=0.001, divide by 10 after 3 epochs without improvement



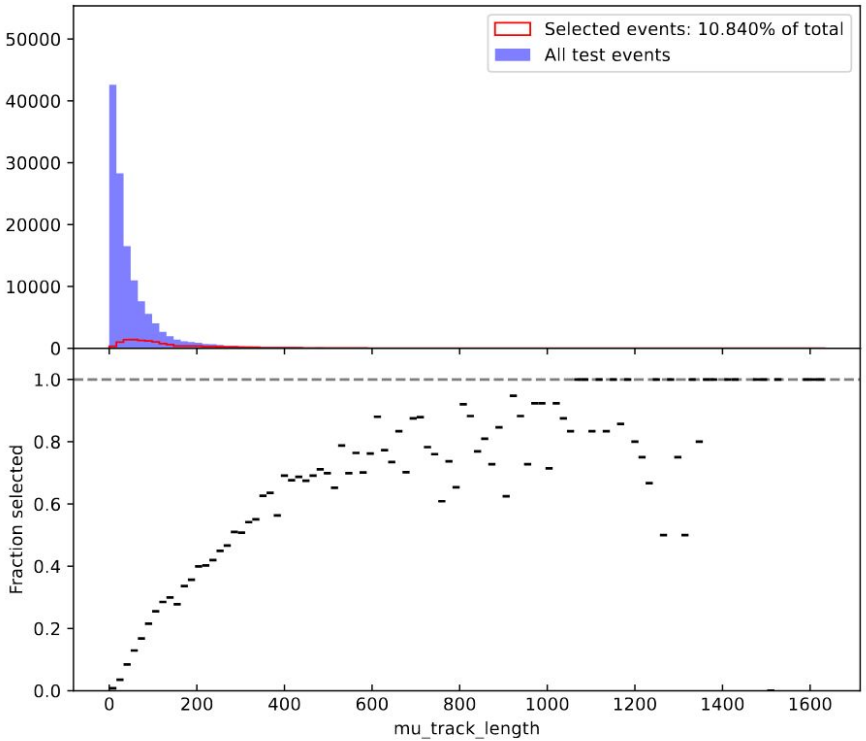
Problems encountered

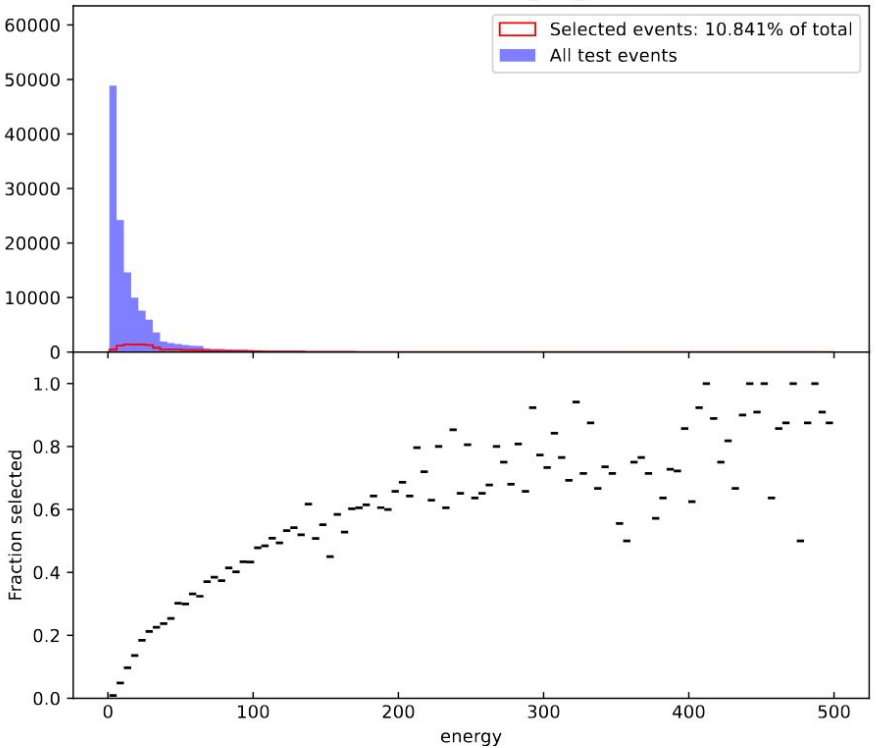
Weighting torch tensor size issue (solved by modifying loss output format)

Multi-GPUs training with slurm silently failing in GraphNeT

SQL database different variable type in truth table not supported

Model's choice of "good events"

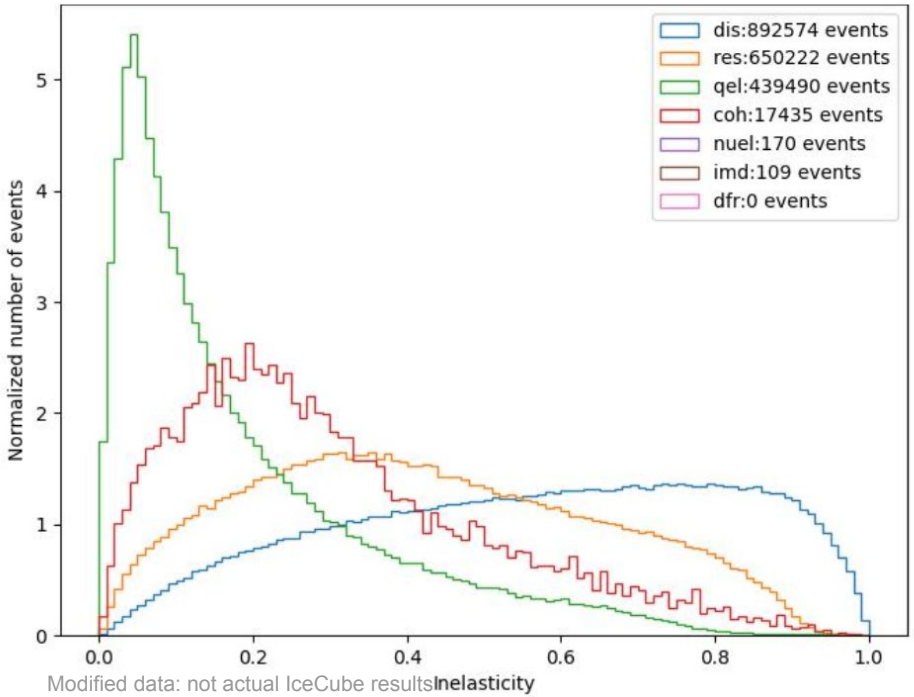




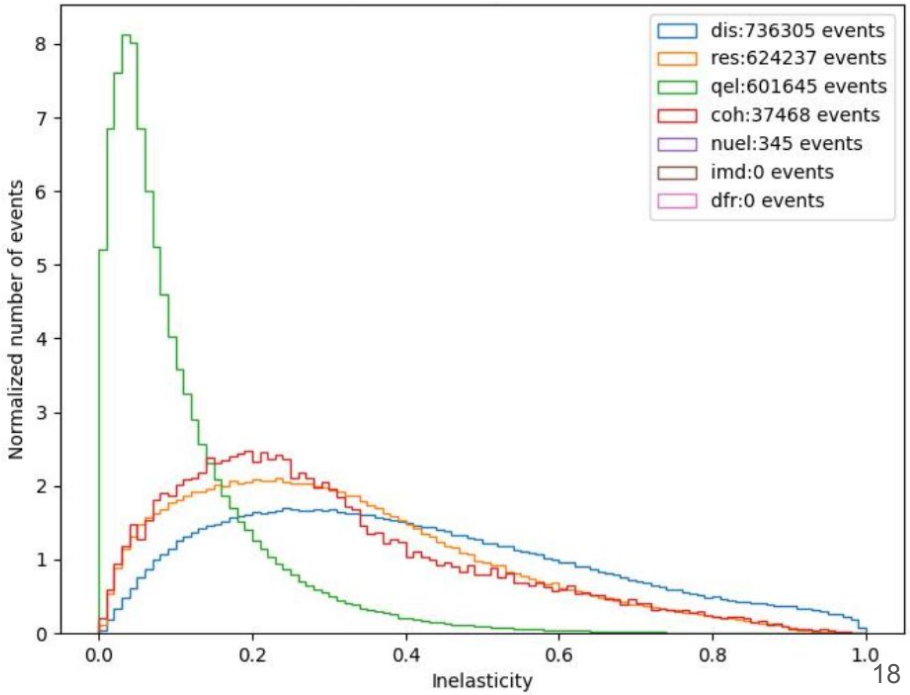
Modified data: not actual IceCube results

“True” initial (MC step 1) inelasticity distribution

ν



$\bar{\nu}$



Modified data: not actual IceCube results

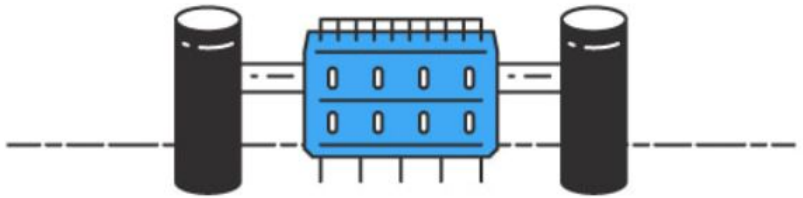
Conclusion

Binary classification has equal separation performances to inelasticity reco

Inelasticity reconstruction promising approach

Model can learn which events it can reconstruct inelasticity best

More work needed for model explanation



External pictures credits

Slide 2: freepik.com

Slide 3 & 18: IceCube collaboration

Slide 6: GraphNeT team

Slide 8: Modern Particle Physics: Mark Thomson

Wikipedia, beta distribution