Low-Energy Neutrino Reconstructions using Recurrent Neural Networks

Bjørn H. Mølvig
Supervisors: Troels Petersen and Oswin Krause

KØBENHAVNS UNIVERSITET

Outline

- Neutrinos 101
  - History
  - The Standard Model
  - IceCube Neutrino Observatory
- Reconstruction Algorithms
  - Current state-of-the-art
  - Recurrent Neural Networks
- Performance
  - Classification and Regression

Neutrinos 101

History of the Neutrino

- Expected $p_1 \to p_2 + p_3$ → discrete energy spectrum
- A fourth particle was postulated by Pauli: a *neutron**, the "neutral one"
- *Neutrino* discovered in 1956, Nobel Prize in 1995

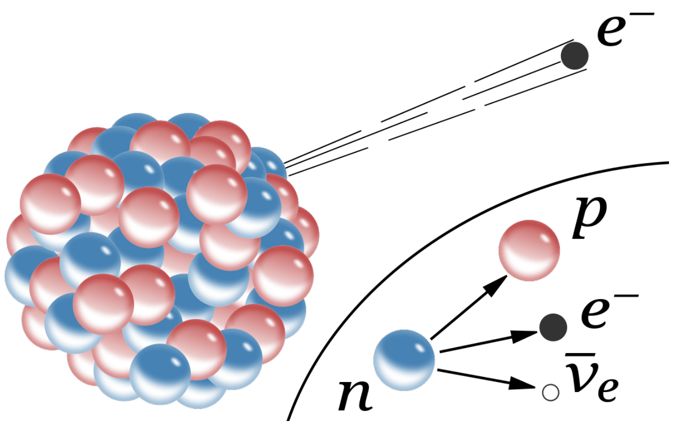


Figure from https://en.wikipedia.org/wiki/Beta_decay

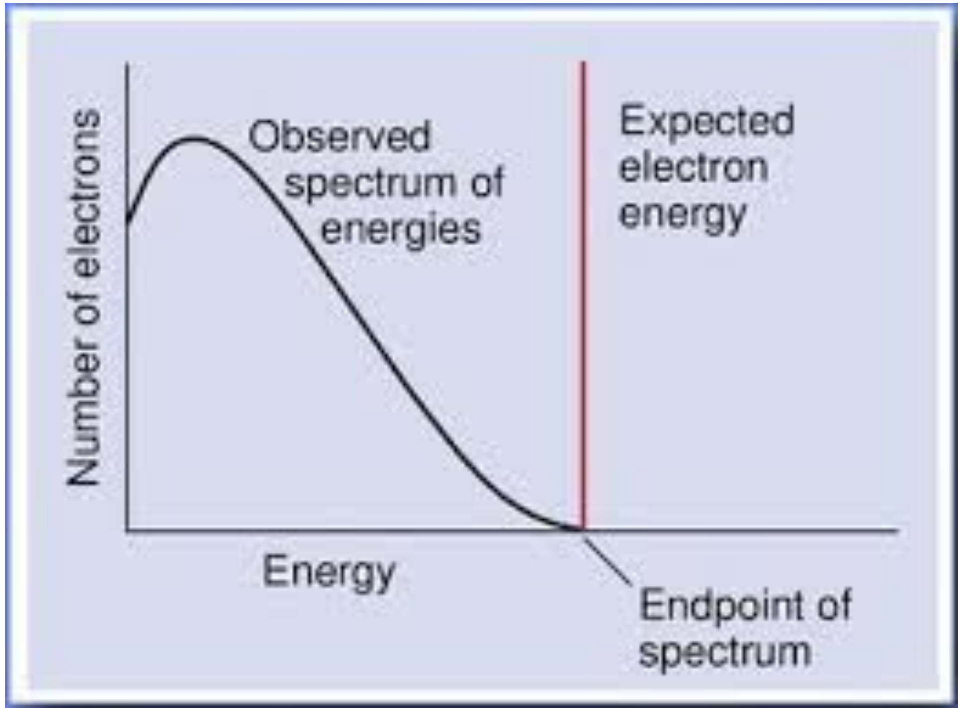


Figure from https://nl.pinterest.com/pin/555490935266054221/

* *What we today know as the neutron was discovered in 1932*

The Standard Model

… But what about

- Matter-antimatter asymmetry?
- Dark matter?

Maybe neutrinos play an important role!

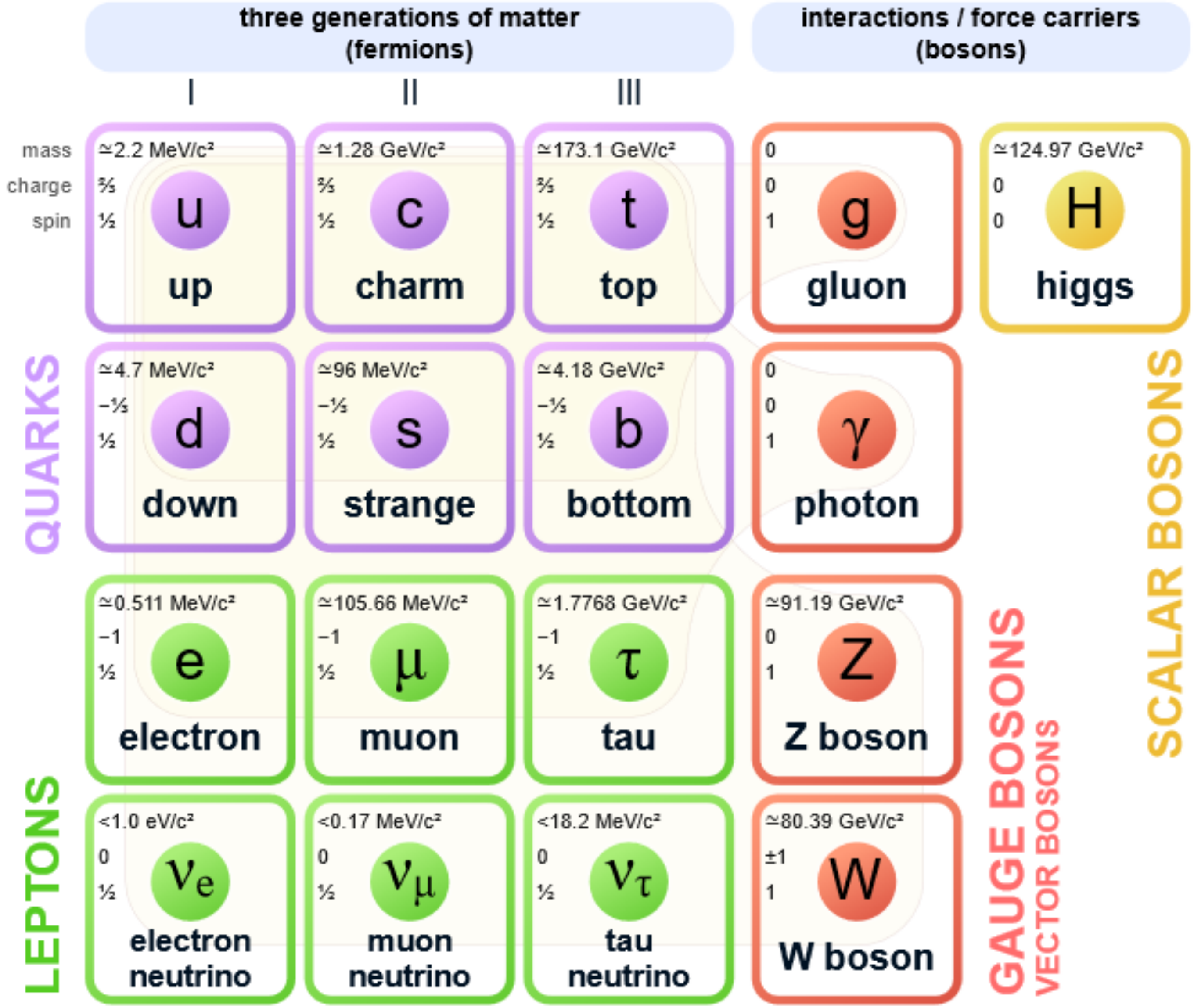
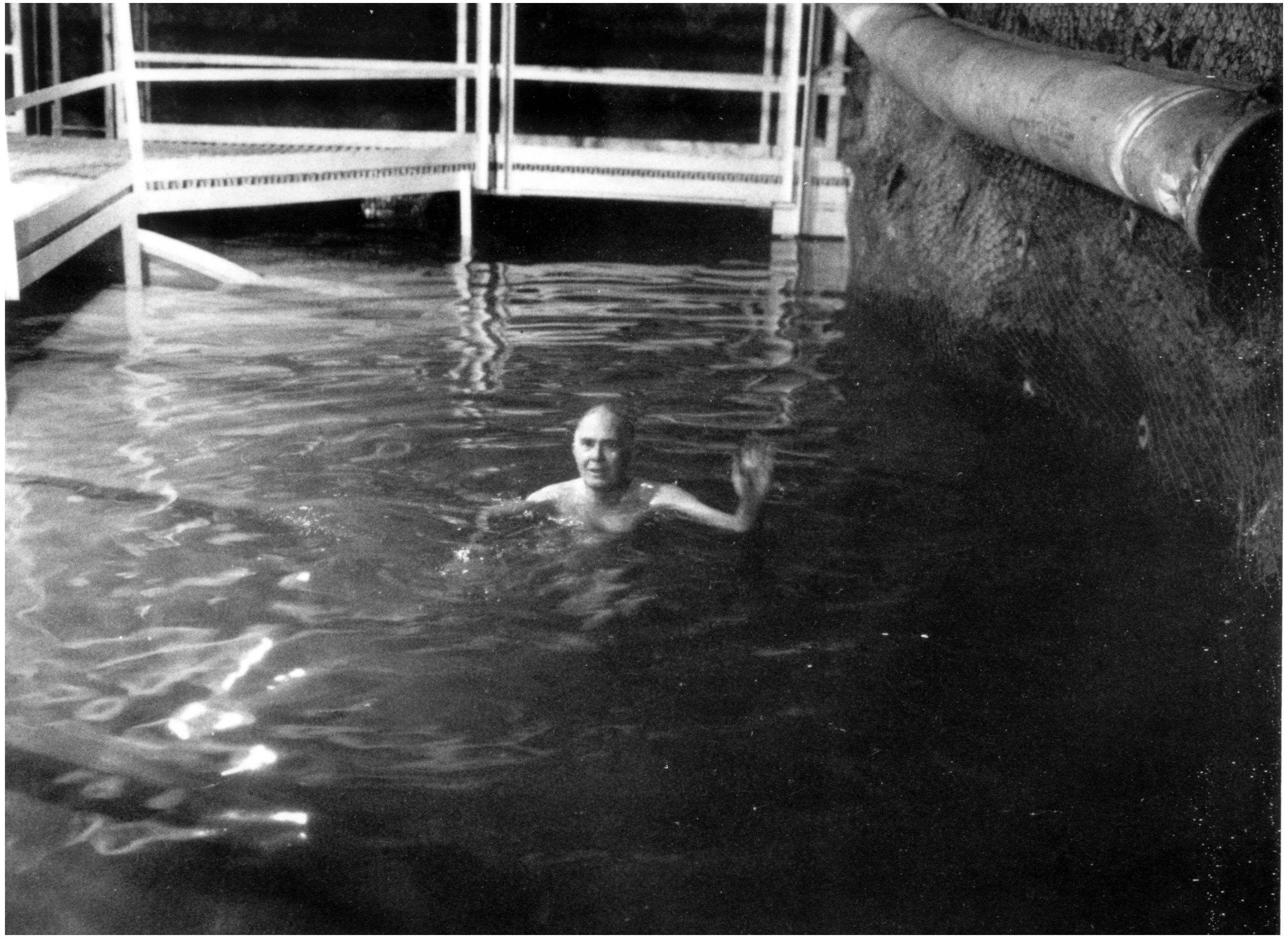


Figure from https://en.wikipedia.org/wiki/Standard_Model

Neutrino Oscillations

- Interest began in the 1950s: The Solar Neutrino Problem.
- Solution: Weak eigenstates are related to mass eigenstates by

$$\begin{bmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{bmatrix} = \begin{bmatrix} U_{e1} & U_{e2} & U_{e3} \\ U_{\mu1} & U_{\mu2} & U_{\mu3} \\ U_{\tau1} & U_{\tau2} & U_{\tau3} \end{bmatrix} \begin{bmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{bmatrix}$$



Nobel Laureate Raymond Davis in the Homestake Mine
Picture from https://www.bnl.gov/bnlweb/raydavis/pictures.htm

Neutrino Oscillations

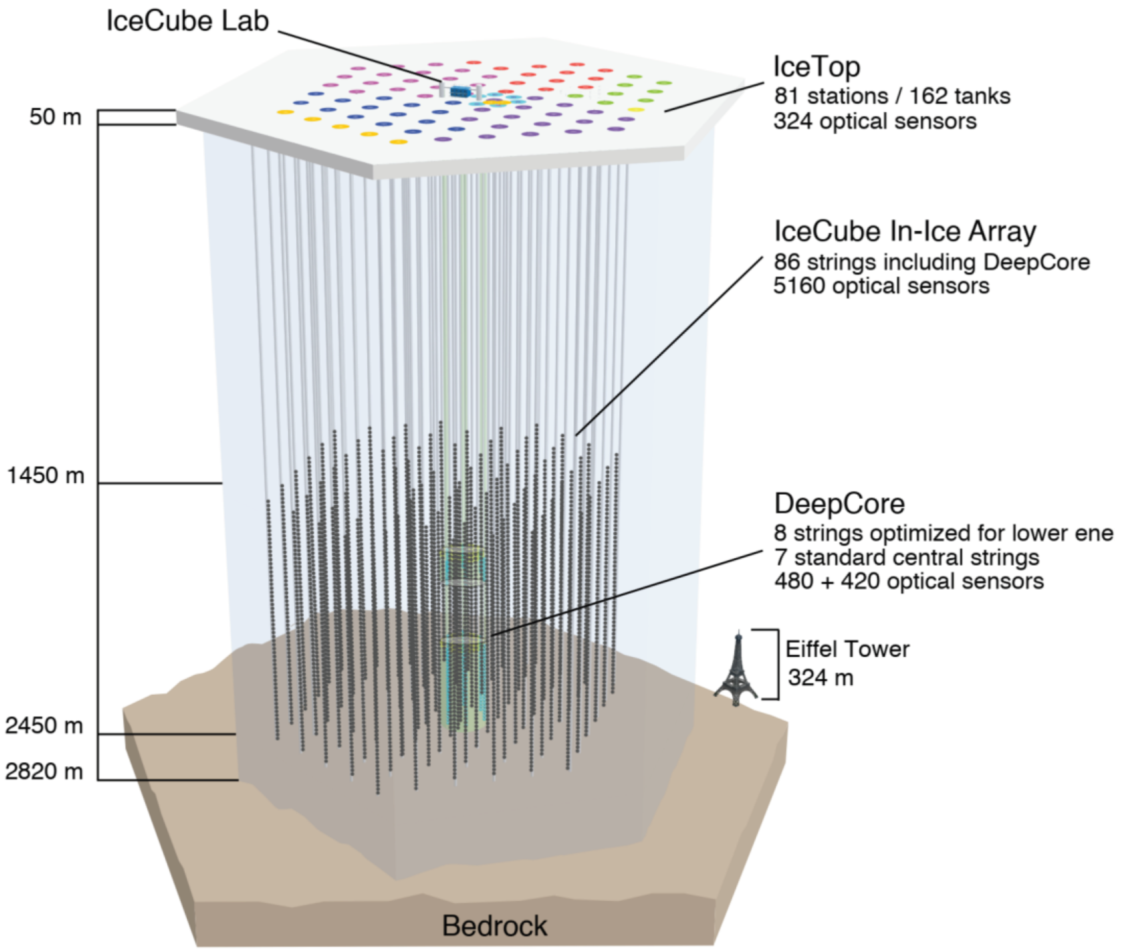
$$\Delta m_{ij}^2 = m_i^2 - m_j^2$$

- .. A bunch of algebra later:

$$P(\nu_\alpha \to \nu_\beta) = \delta_{\alpha\beta} - 2\sum_{i,j=1}^{3} \mathrm{Re}\left(U_{\alpha i}U_{\beta i}^* U_{\alpha j}^* U_{\beta j}\right)\sin^2\left(\frac{\Delta m_{ij}^2 L}{4E}\right) + \sum_{i,j=1}^{3} \mathrm{Im}\left(U_{\alpha i}U_{\beta i}^* U_{\alpha j}^* U_{\beta j}\right)\sin\left(\frac{\Delta m_{ji}^2 L}{2E}\right)$$

- The *combined* neutrino flux from the sun checks out!

The IceCube Neutrino Observatory



Method of Detection

- Detect the *Cerenkov radiation* of decay products emitted at an angle

$$\cos\theta_C = \frac{1}{n\beta}$$

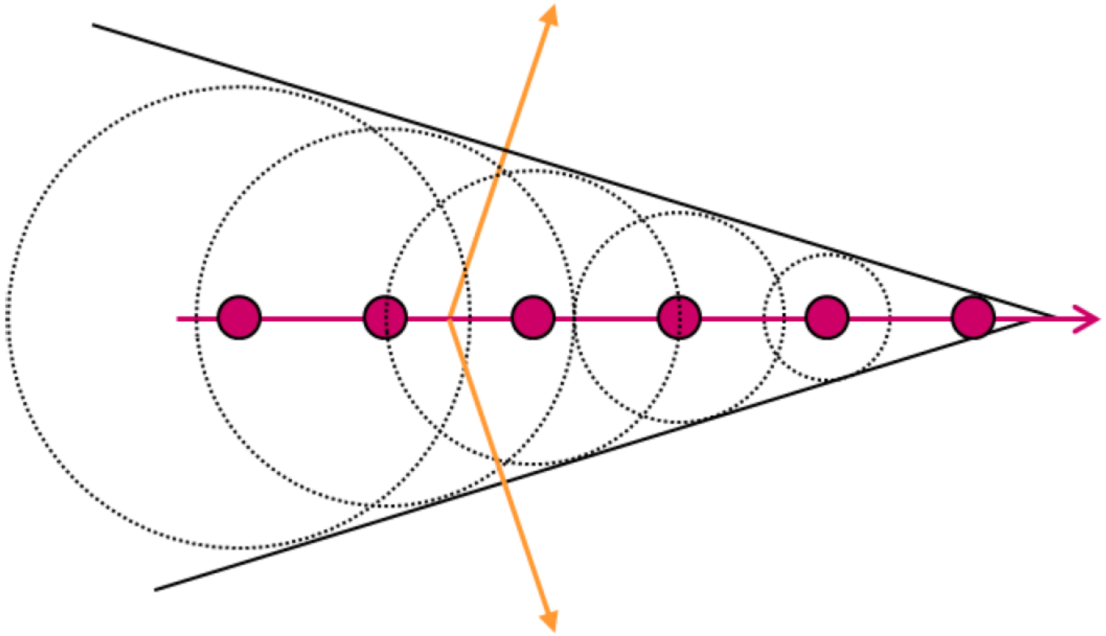


Figure from http://large.stanford.edu/courses/2014/ph241/alaeian2/

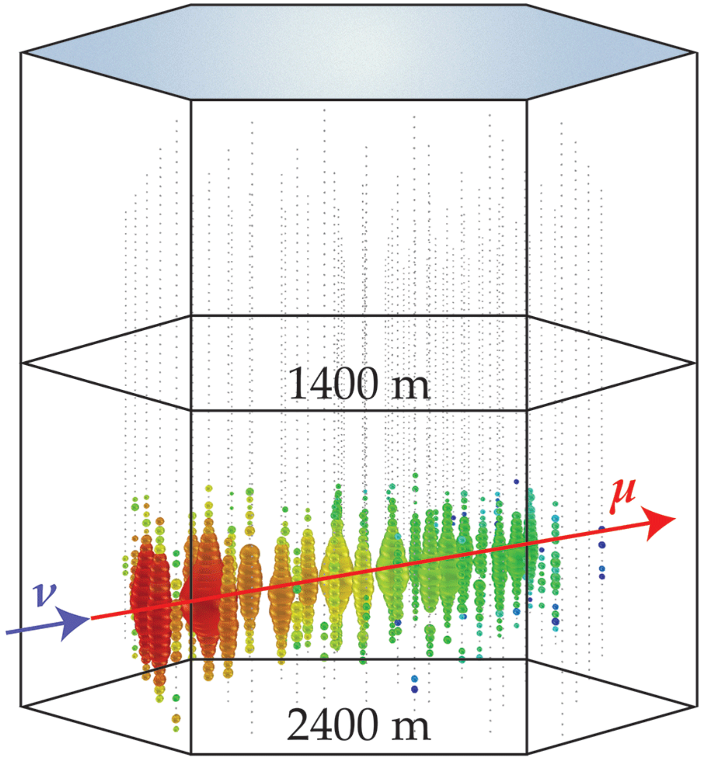


Figure from https://physics.aps.org/articles/v7/88

Atmospheric Neutrinos

- Generation in atmosphere
  → travel through Earth → Detection

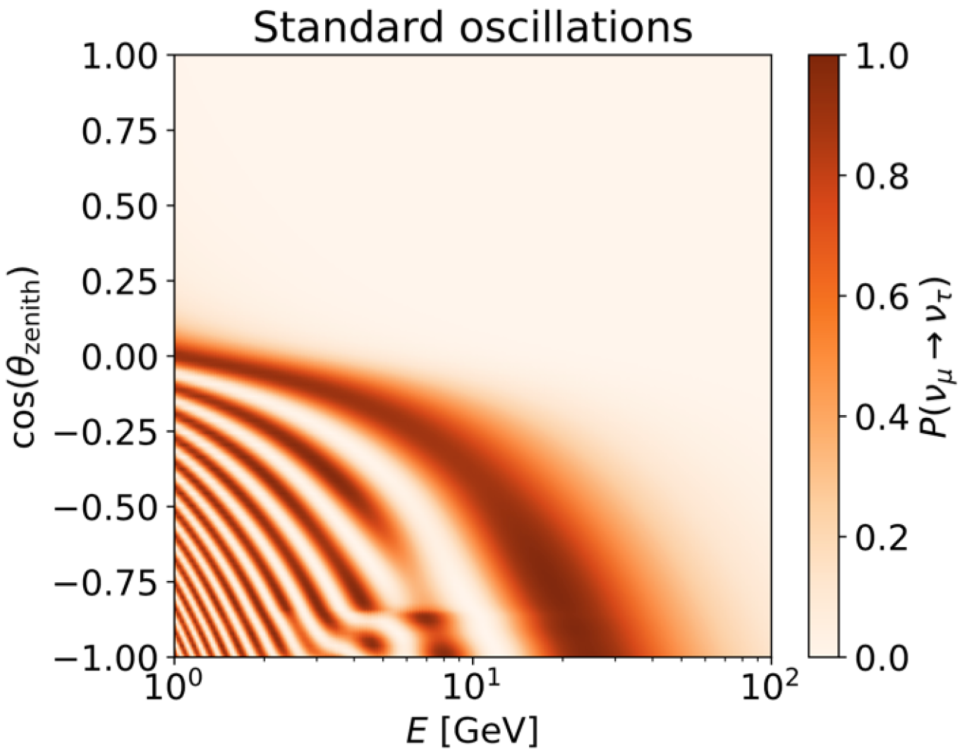


Figure courtesy of Tom Stuttard

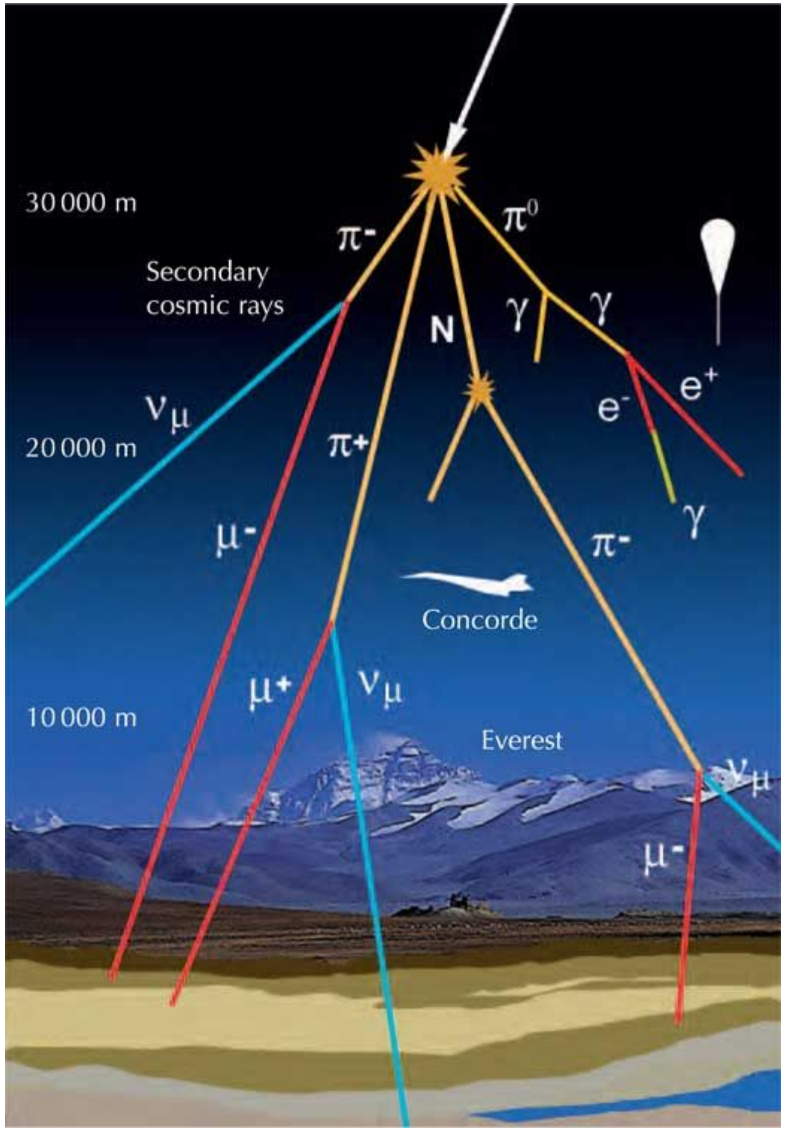
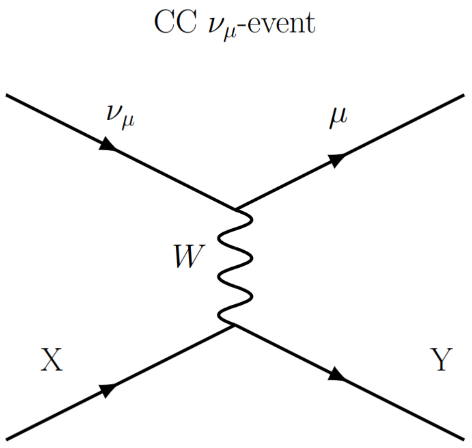
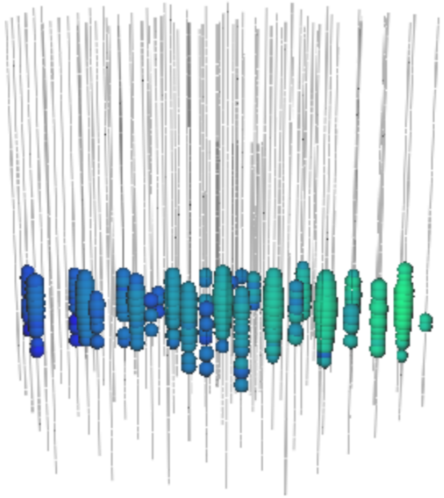


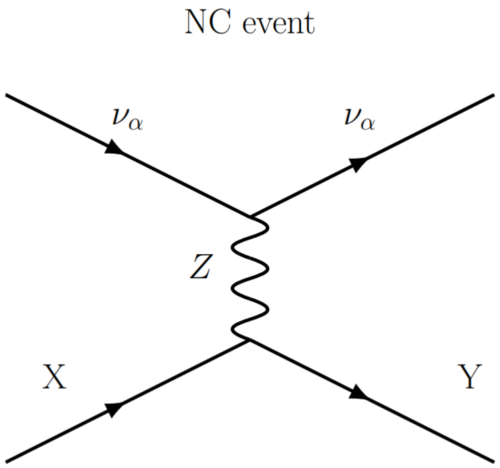
Figure from https://www.pinterest.co.uk/pin/250090585534637268/

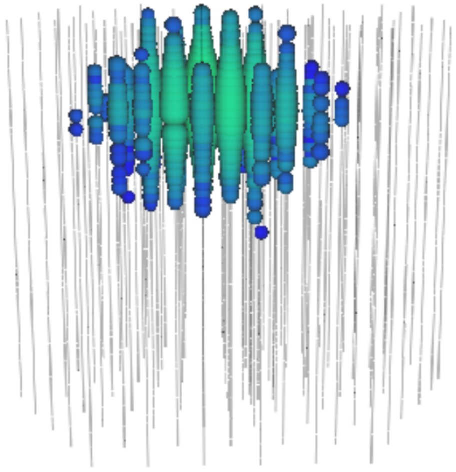
Event signatures

- 2 different signatures at low energy: *Tracks* (above) and *cascades* (below).









Figures "*Origin of IceCube's Astrophysical Neutrinos: Autocorrelation, Multi-Point-Source and Time-Structured Searches*"

In reality…

- 30 GeV Tau- (a), muon- (b) and electron neutrino (c) events

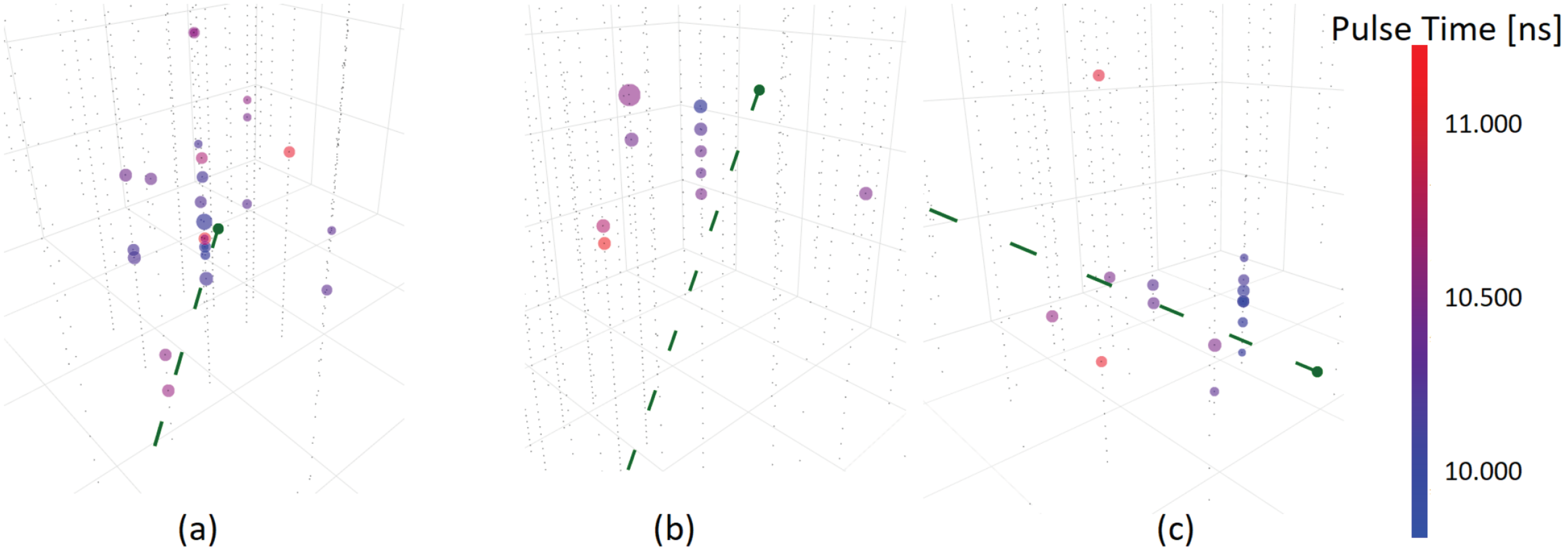


Figure courtesy of Mads Ehrhorn

Neutrino Reconstruction

The Retro Reconstruction Algorithm

- Current and best low-energy reconstructor
- Table-based maximum likelihood estimator
- Disadvantages
  - Storing tables requires ~1 TB memory
  - A single reconstruction can take minutes

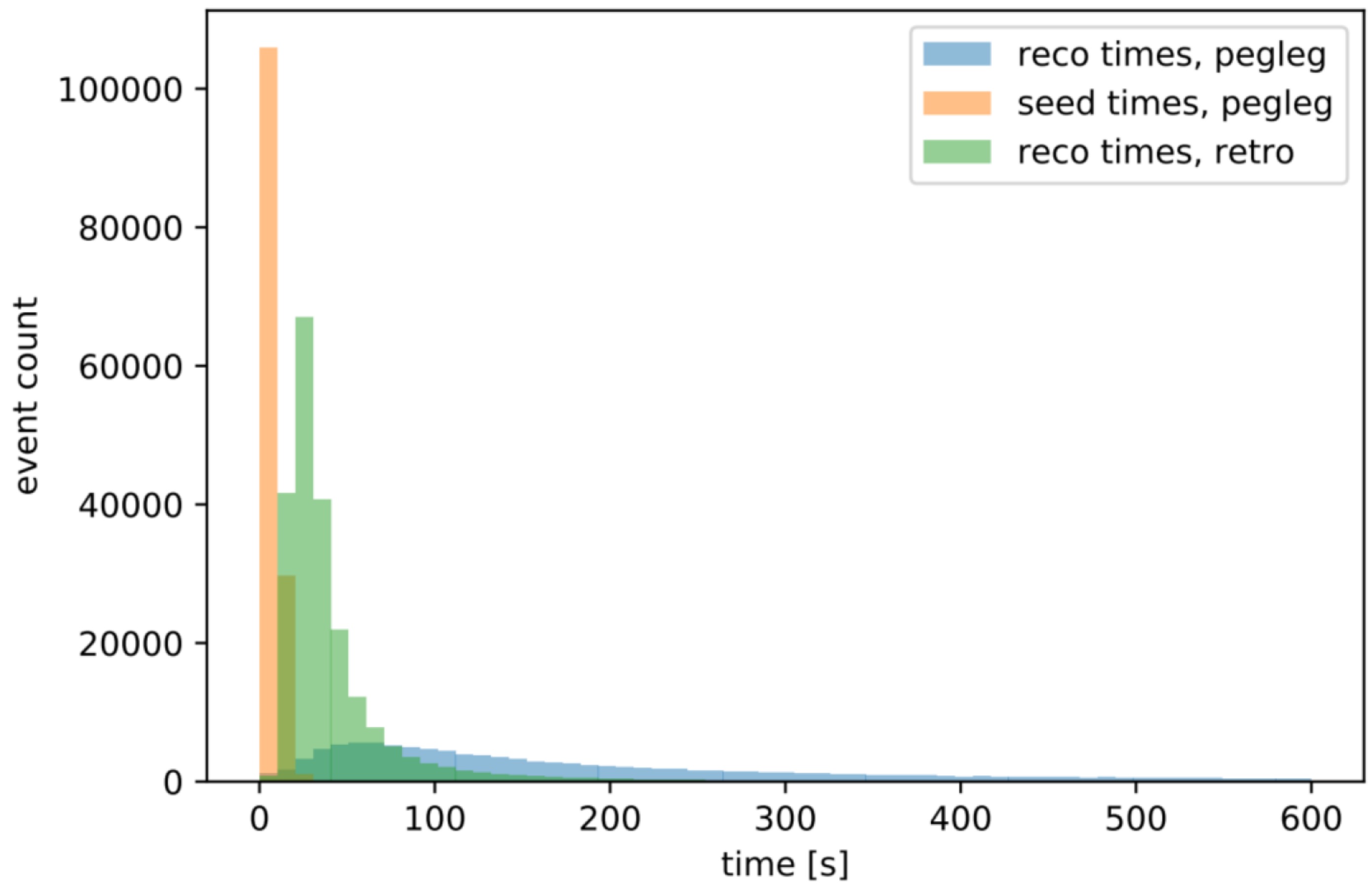


Figure courtesy of Tom Stuttard

Machine Learning – Overview

Given ***labelled*** events $\{(x_1, y_1), .., (x_N, y_N)\}$ ***train*** a model to minimize the empirical ***loss***

$$\hat{L}(h) = \frac{1}{N} \sum_{i=0}^{N} l(h(x_i), y_i)$$

i.e. find

$$h^* = \arg\min_{h \in \mathcal{H}} \hat{L}(h)$$

capable of reconstructing ***unseen*** events

Image from https://xkcd.com/1838/

Machine Learning – Overview

- ***Labelled Events:*** From simulation
- ***Training***: Using Stochastic Gradient Descent

$$w_{t+1} = w_t - \eta \nabla_w \hat{L}(w_t),\ \eta > 0$$

- ***Loss***: Our measure of what is good

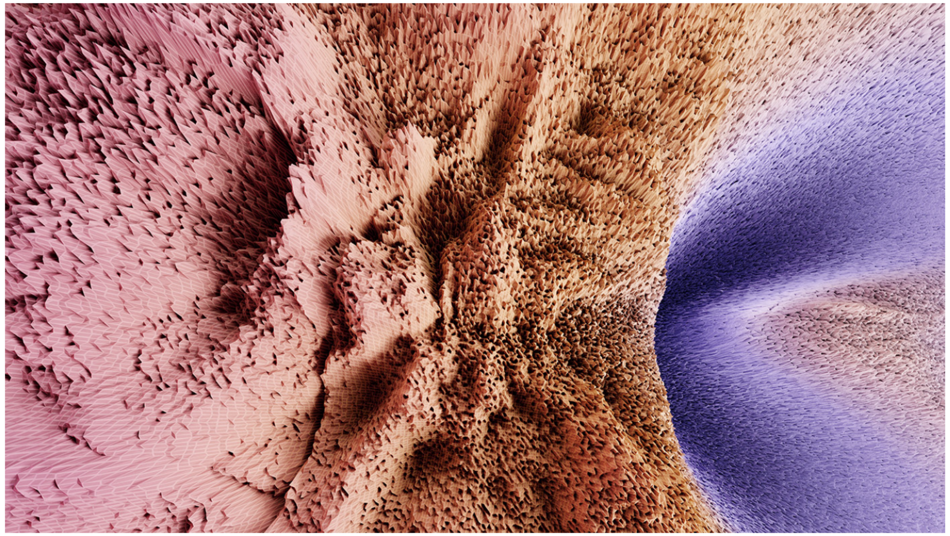
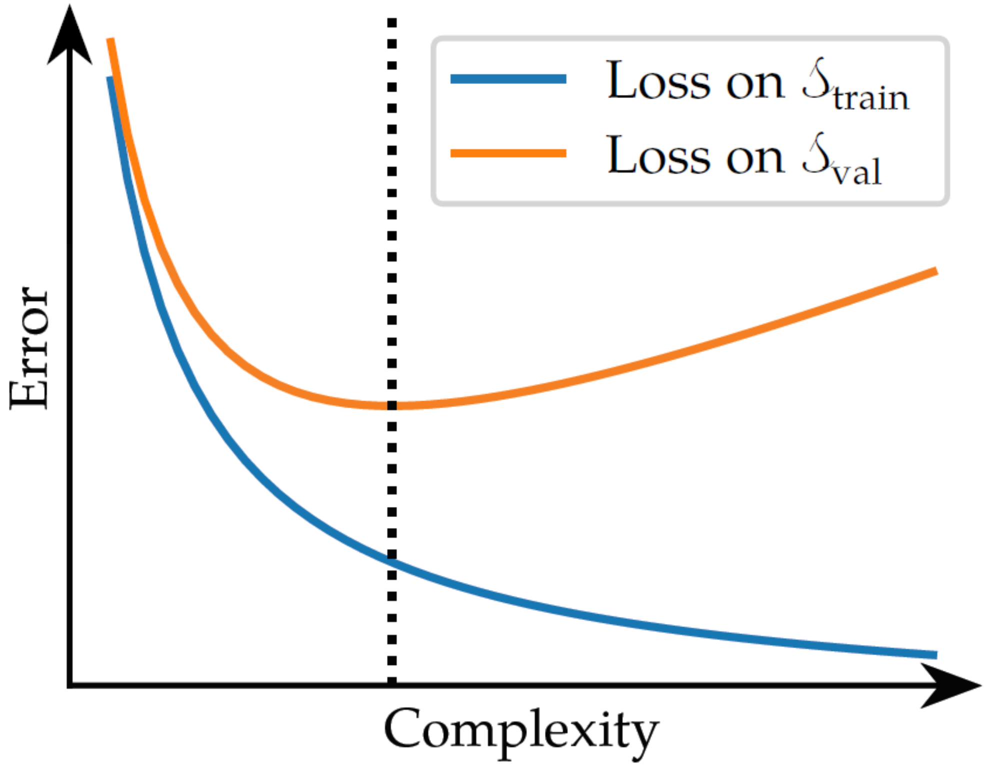


Image from https://losslandscape.com/



Setup

- Feed a Recurrent Neural Network (RNN) time-ordered DOM sequence.

DOM_x: The x-coordinate of the DOM.

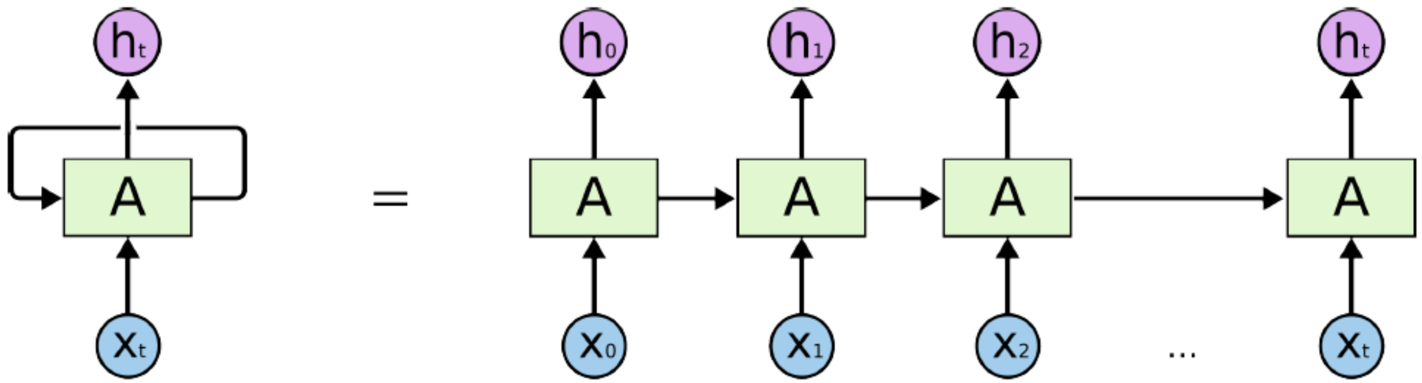
DOM_y: The y-coordinate of the DOM.

DOM_z: The z-coordinate of the DOM.

DOM_q: The charge extracted from the raw waveform.

DOM_t: The time (in ns) w.r.t. the triggertime at which the pulse was detected.

DOM_ATWD: A Boolean. 1 if a ATWD-digitizer recorded the waveform or 0 if a fADC-digitizer recorded the waveform.

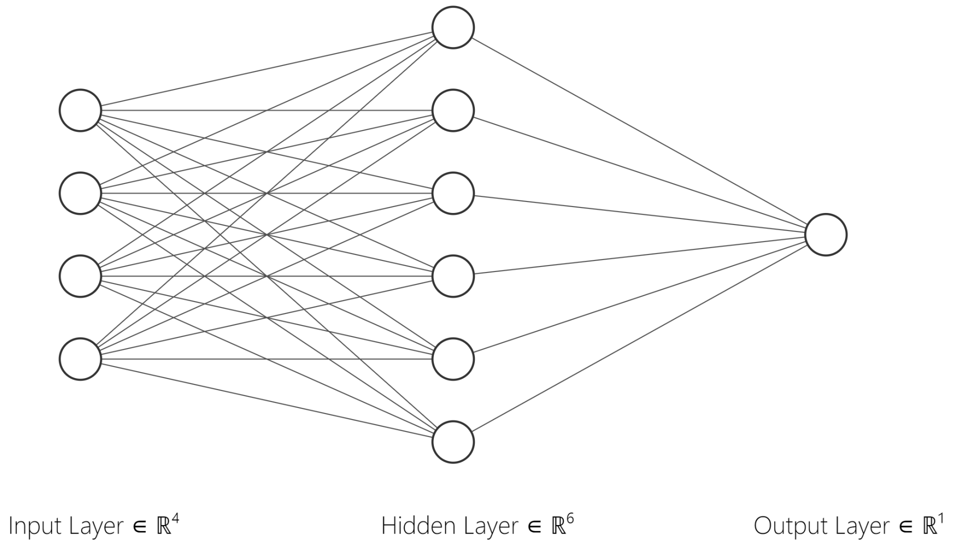


Figures https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Recurrent Neural Networks

For each neuron

- Input feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{h}_{i-1}$
- Apply nonlinear function
- Output $\boldsymbol{h}_i$



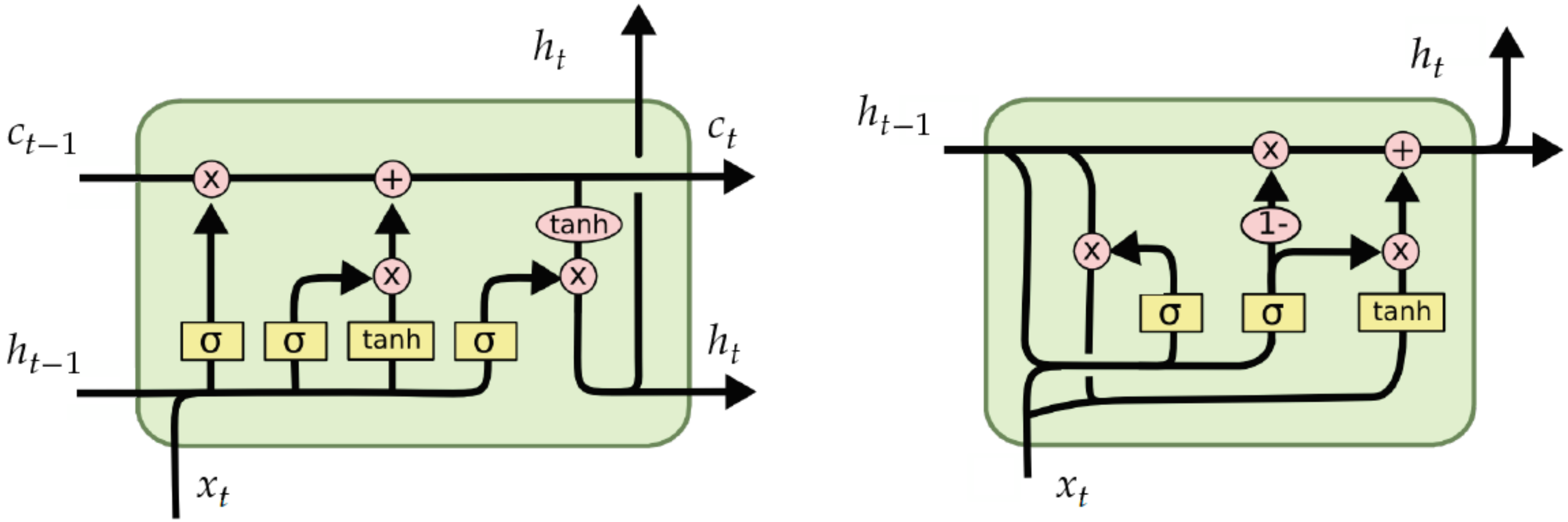
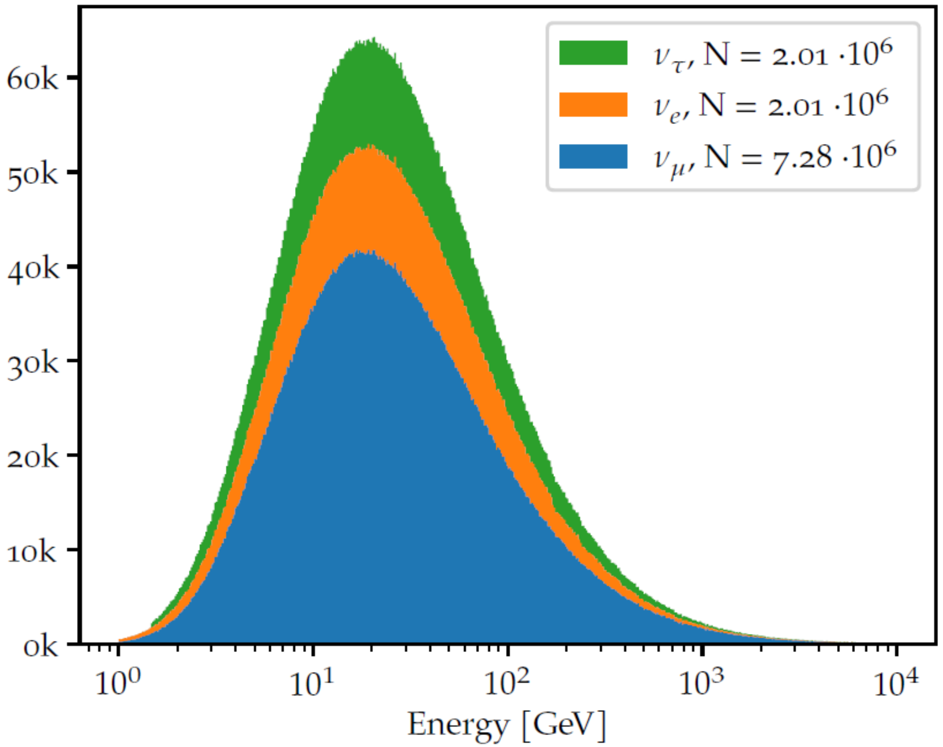


Figure https://colah.github.io/posts/2015-08-Understanding-LSTMs/

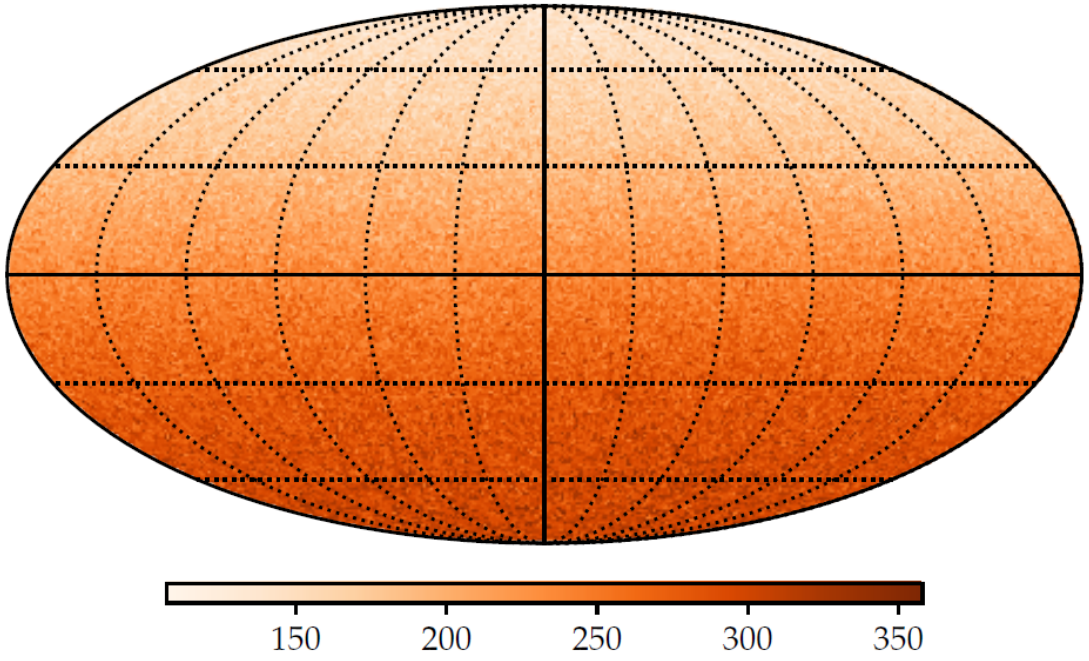
Data

	All Data			After selection		
	Train	Val.	Test	Train	Val.	Test
ν_e	1.61 M	0.20 M	0.20 M	1.59 M	0.19 M	0.19 M
ν_μ	5.82 M	0.73 M	0.73 M	5.75 M	0.72 M	0.72 M
ν_τ	1.61 M	0.20 M	0.20 M	1.59 M	0.19 M	0.19 M

OscNext Lvl5



OscNext Lvl5



Targets

Regression

- Neutrino energy
- *x*-, *y*-, *z*- and *t*-components of interaction vertex
- *x*-, *y*- and *z*-components of direction unit vector

Classification

- Type

Corresponding losses

$$\hat{L} = \frac{1}{N}\sum_{i=1}^{N} \operatorname{logcosh}\left(\log\left(\frac{E_{\text{reco}}}{E_{\text{true}}}\right)_i\right) + \operatorname{logcosh}(|\boldsymbol{r}_{\text{reco}} - \boldsymbol{r}_{\text{true}}|_i) + \operatorname{logcosh}(|\boldsymbol{p}_{reco} - \boldsymbol{p}_{true}|_i) + \operatorname{logcosh}(t_{reco} - t_{true})_i$$

$$\hat{L} = -\mathbb{E}_{p_{\text{true}}}[\log(p_{\text{predicted}})]$$

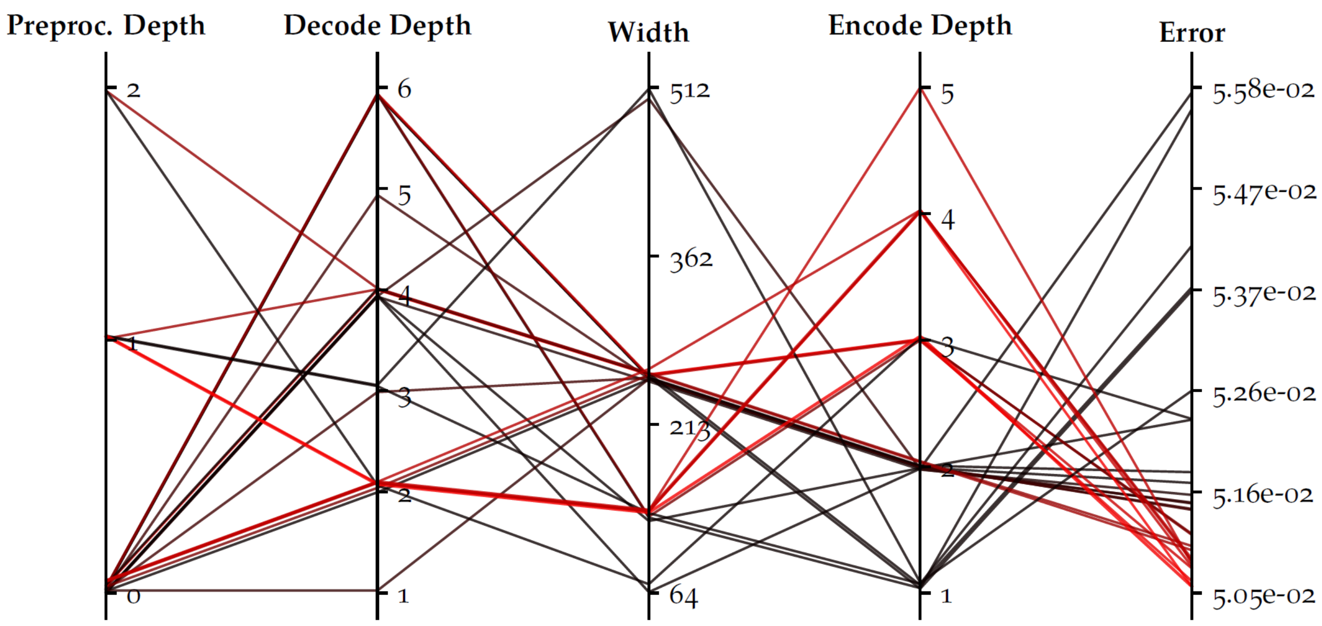
Best Architecture

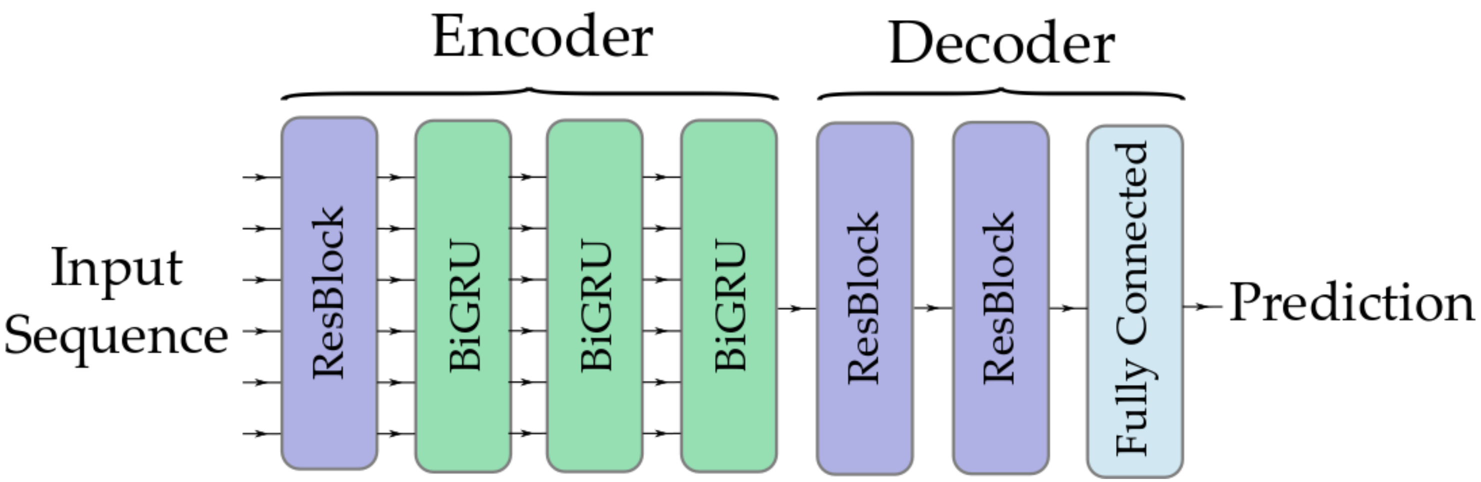
Preprocessing layer

→ Sequence to sequence

→ Many-to-one

→ Decoding layers





Reconstruction Performance

Classification Performance – Track vs Cascade

Performance captured by the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC)

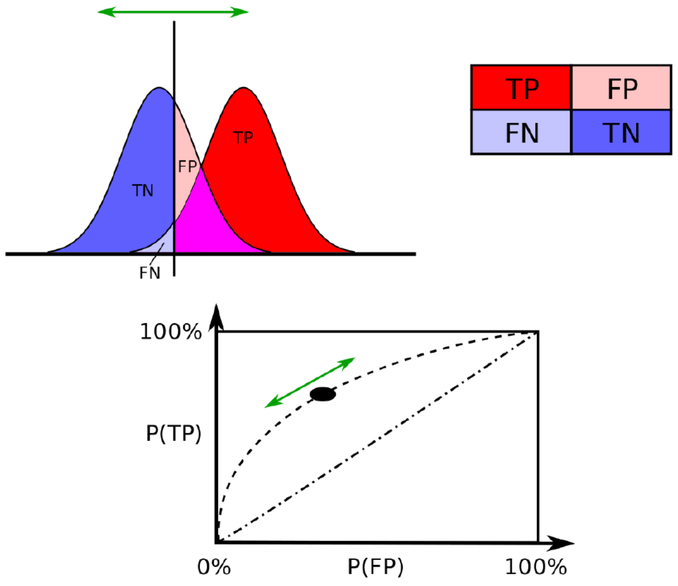
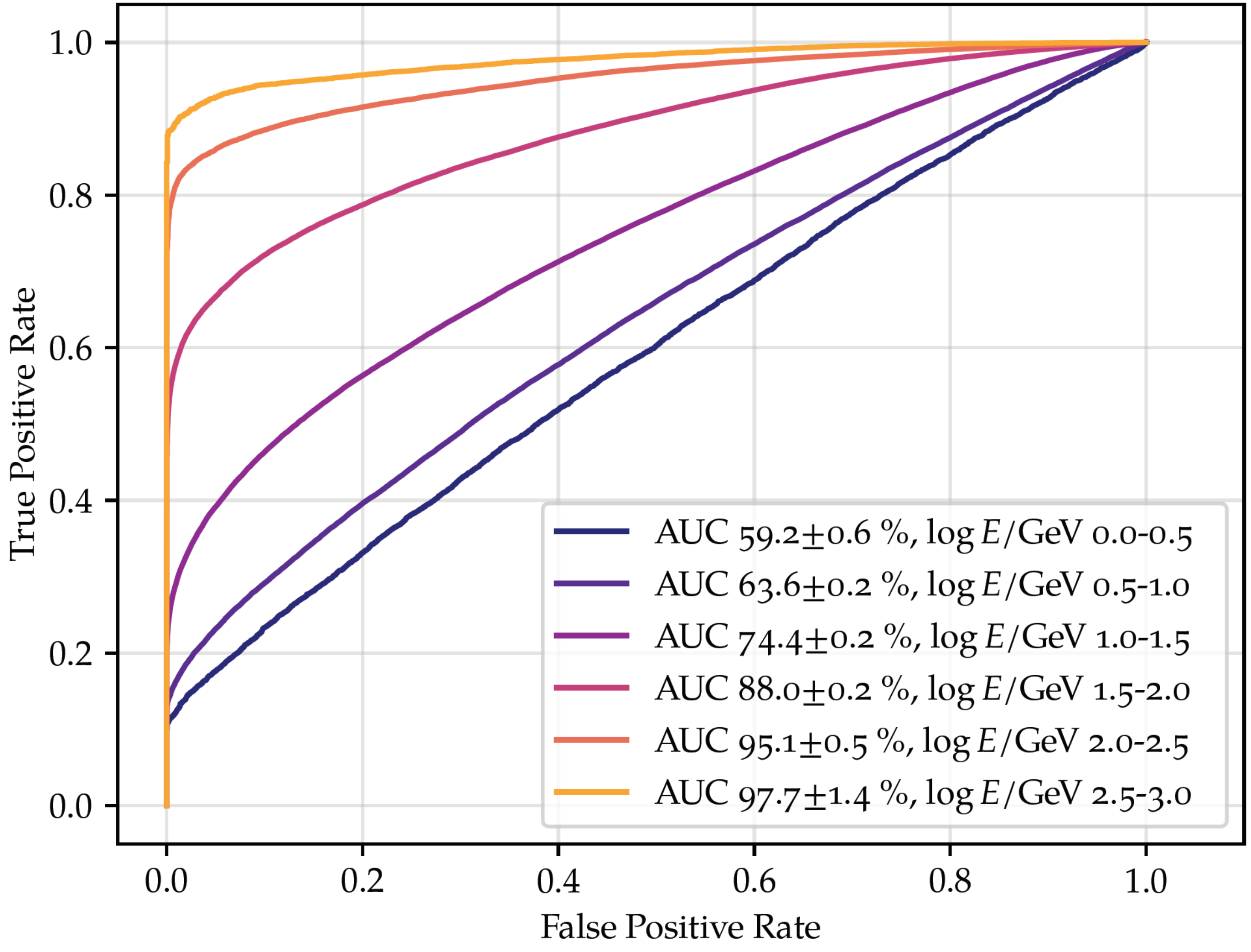


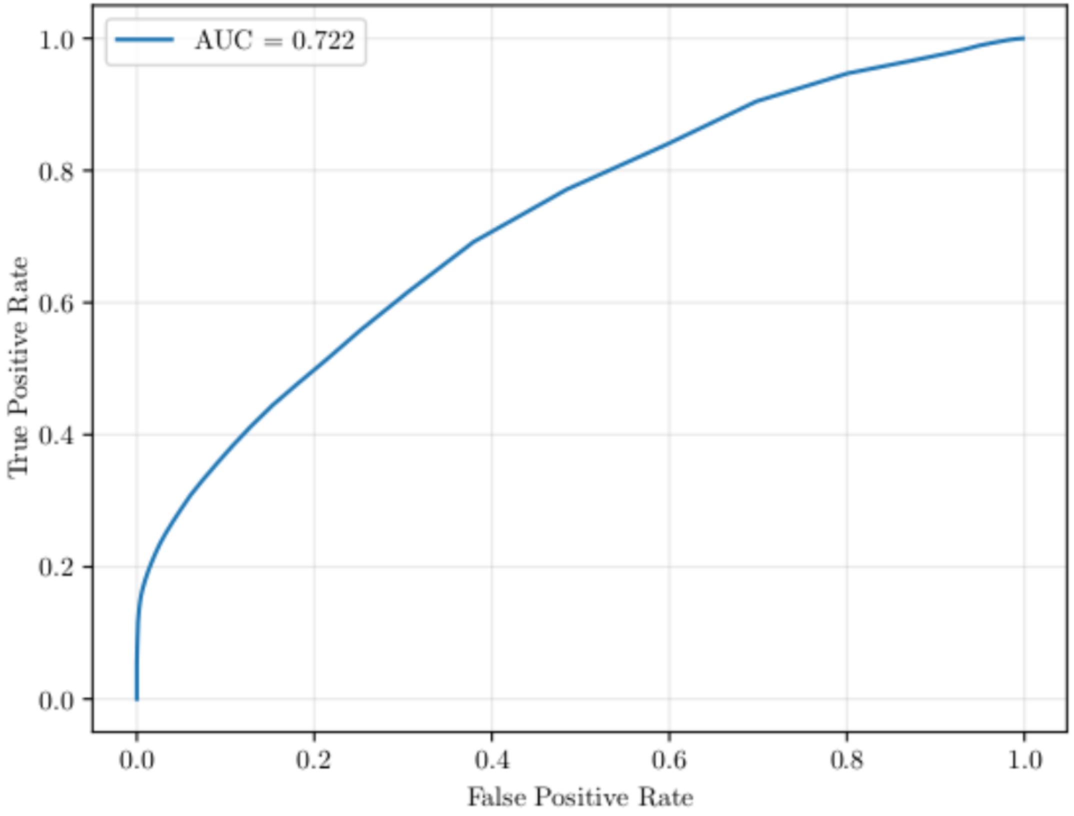
Figure from https://en.wikipedia.org/wiki/Receiver_operating_characteristic



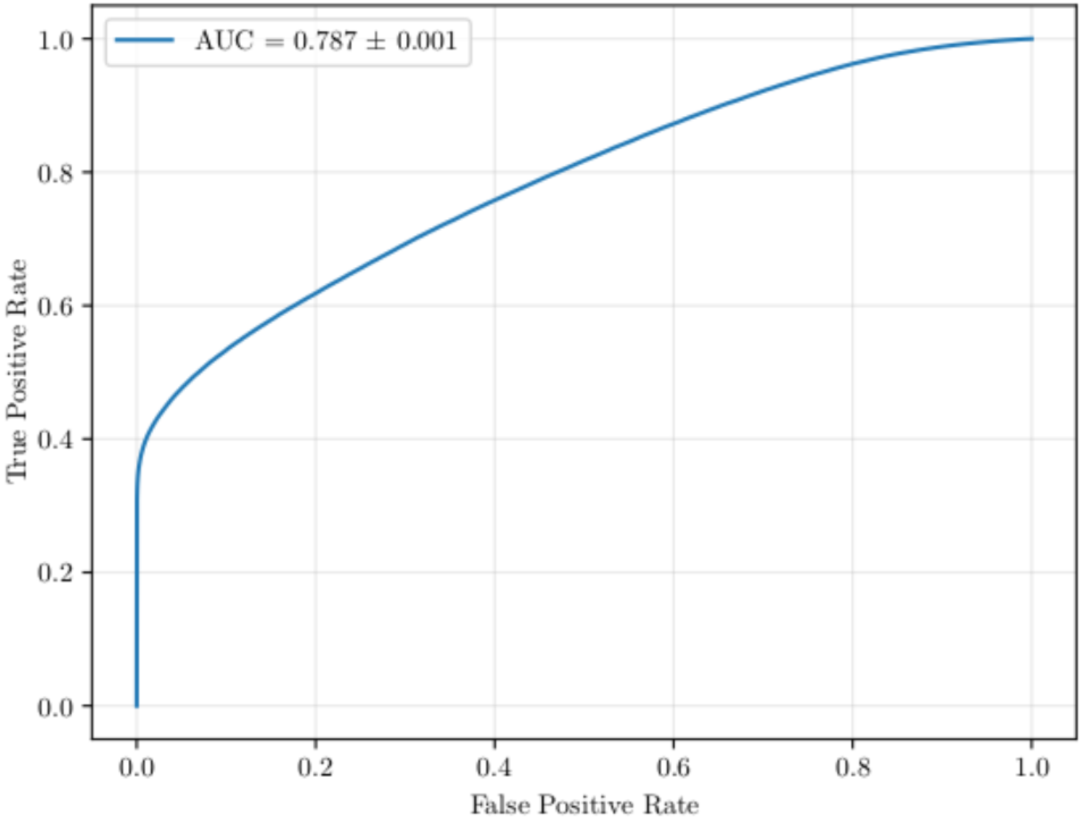
Classification Performance – Track vs Cascade

WARNING: IceCube performance "by eye" and potentially Different datasets

IceCube



RNN



Regression Performance

Performance: Width of the binned error distributions

$$W(e) = \frac{IQR(e)}{1.349}$$

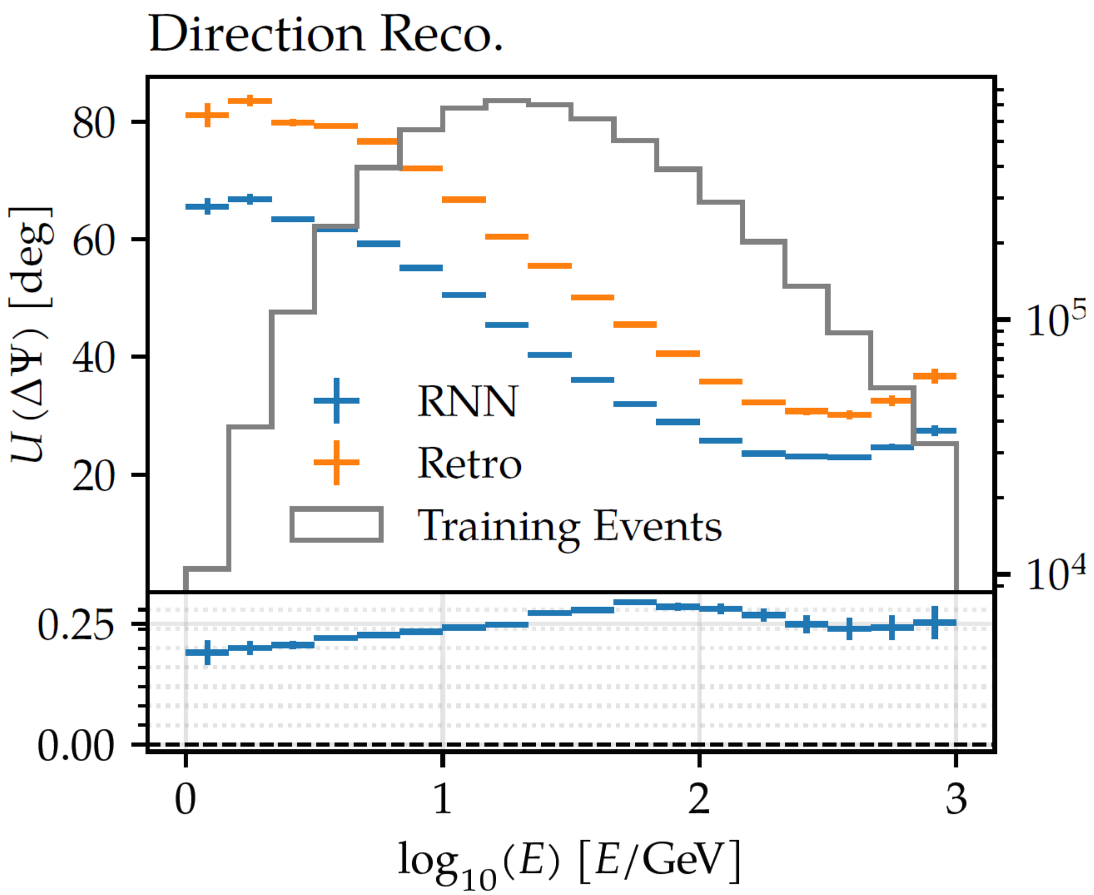
for unbounded error distributions

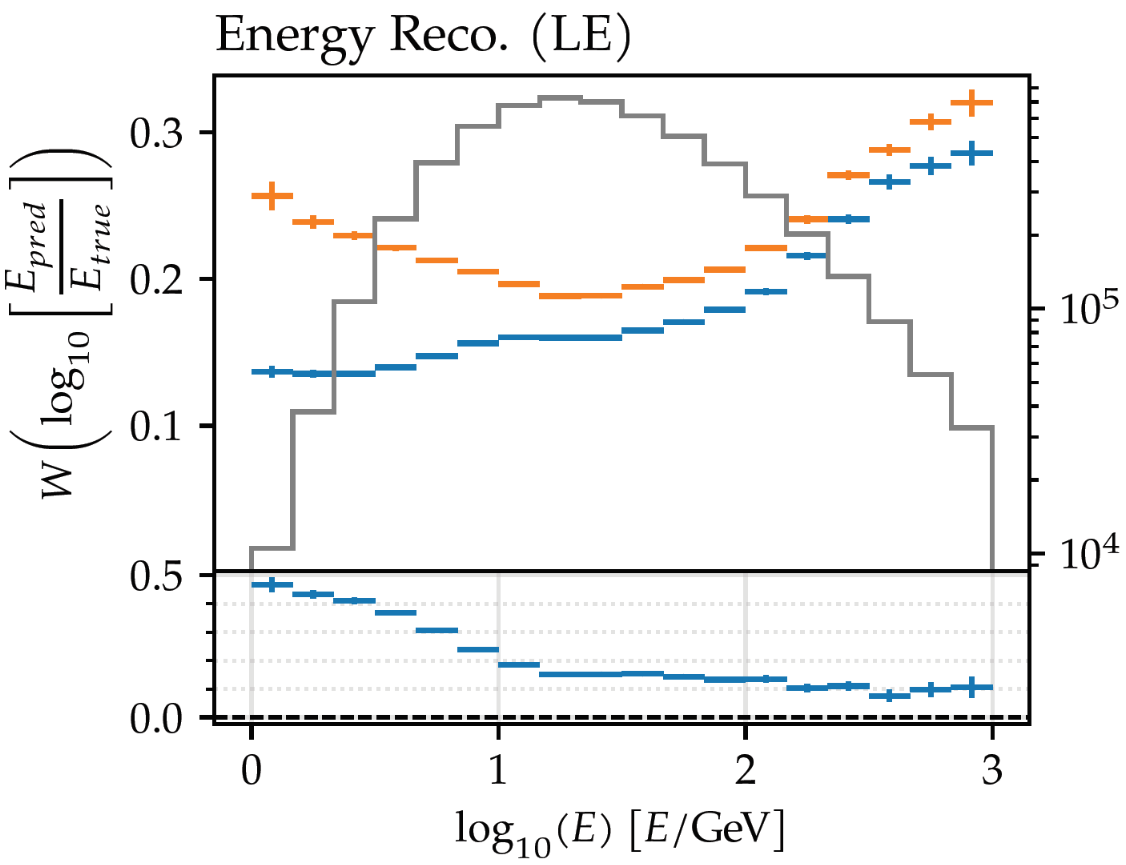
… And 68th percentile of the binned error distributions

$$U(e) = e_{0.68}$$

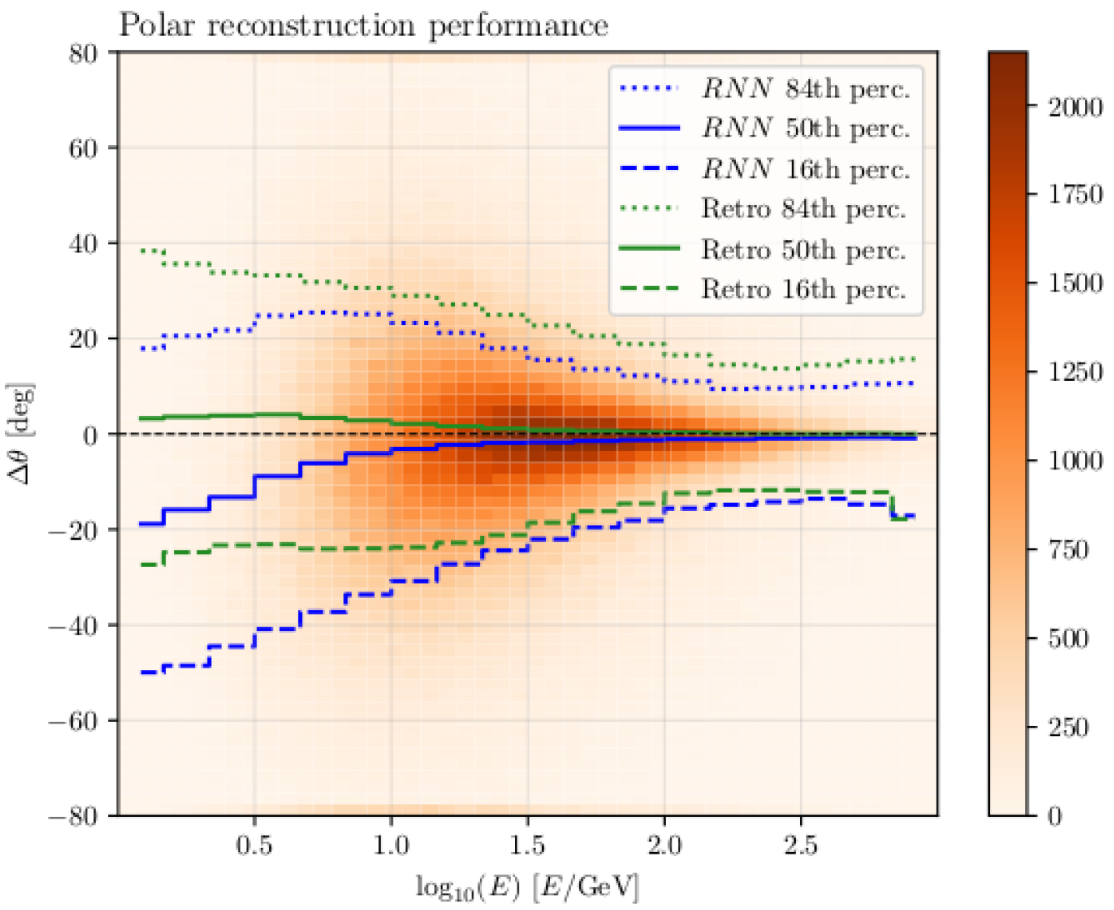
For bounded error distributions

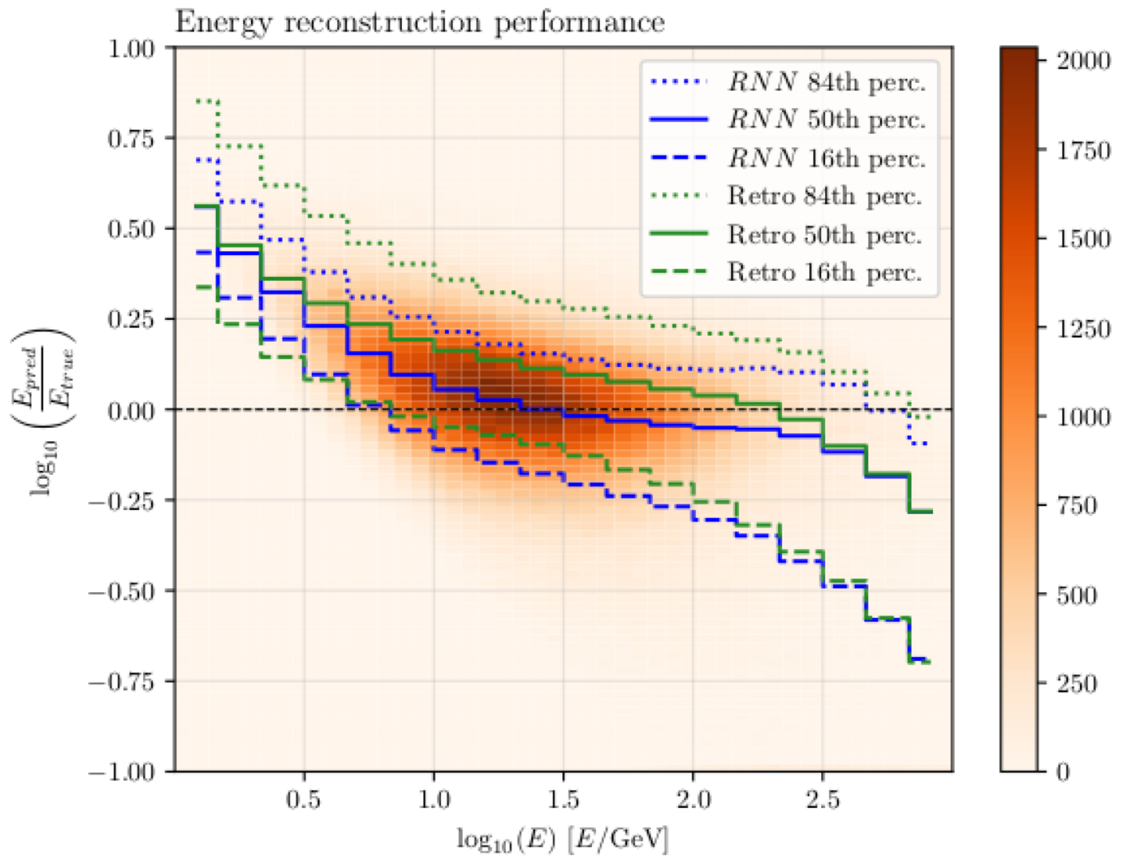
Muon Neutrino Reconstruction





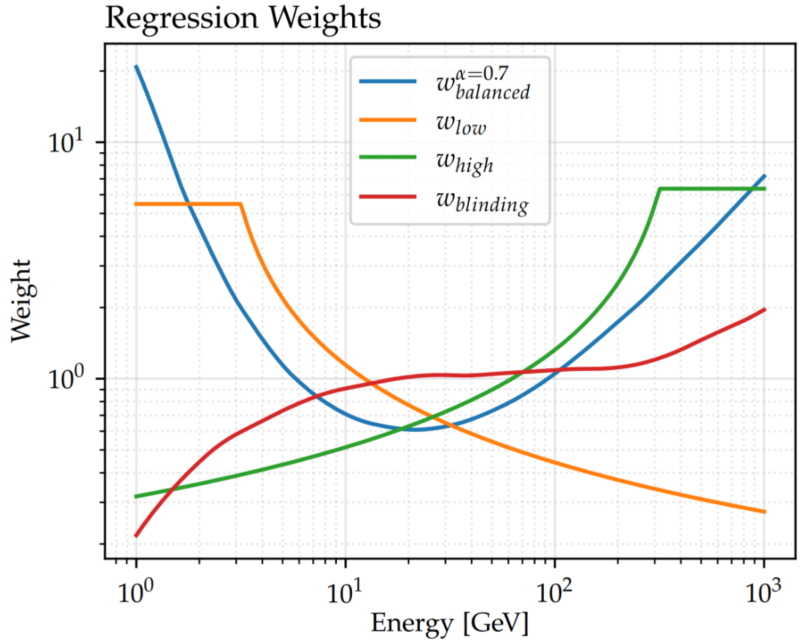
Muon Neutrino Reconstruction

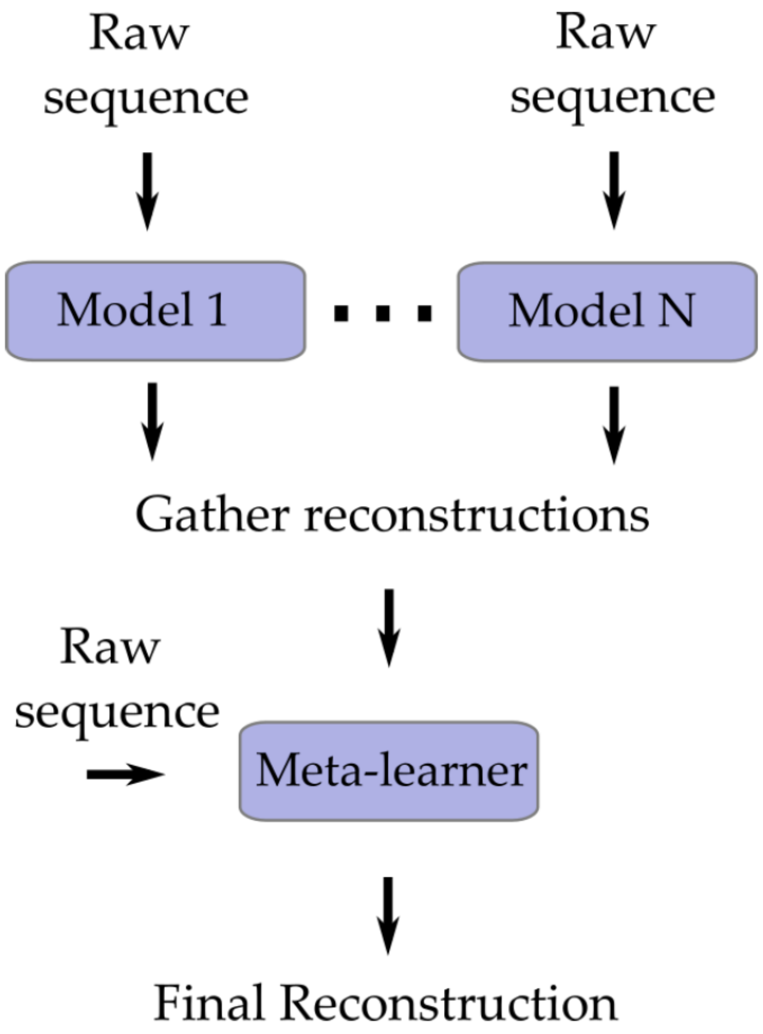




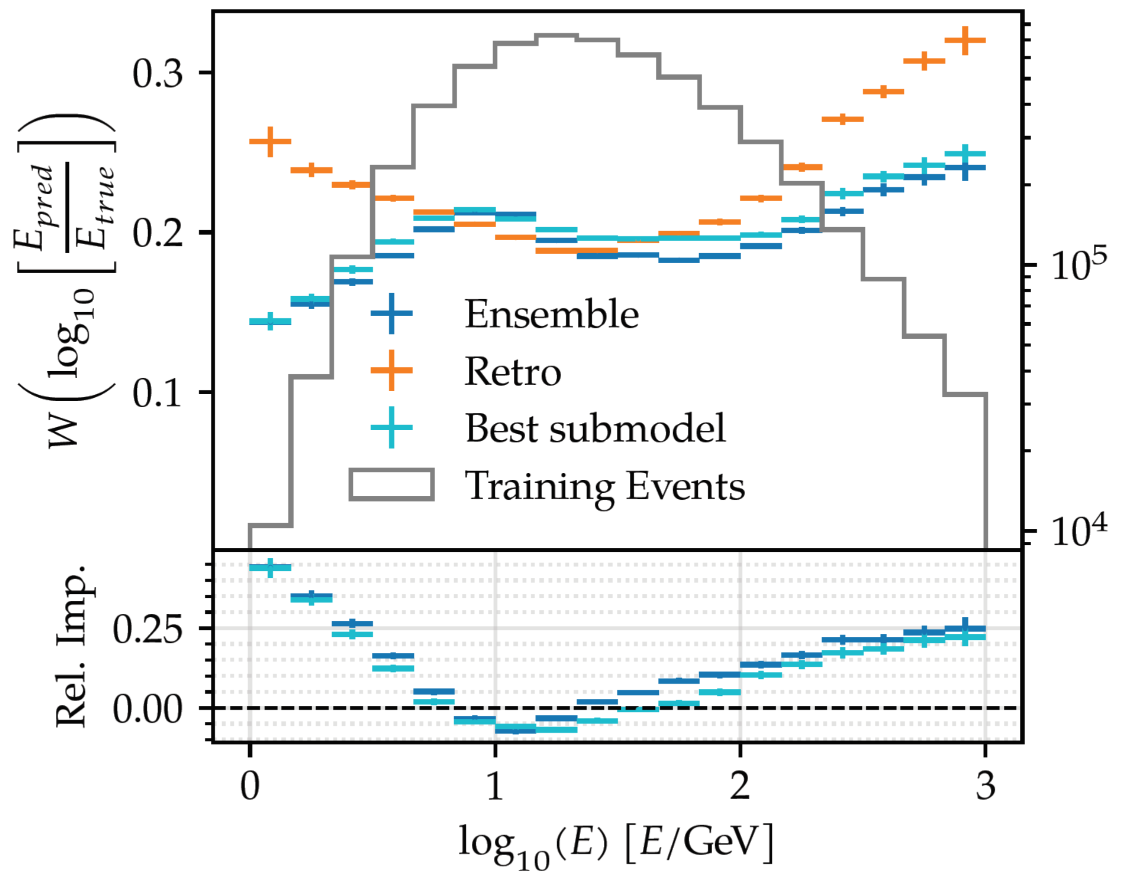
Ensemble Model

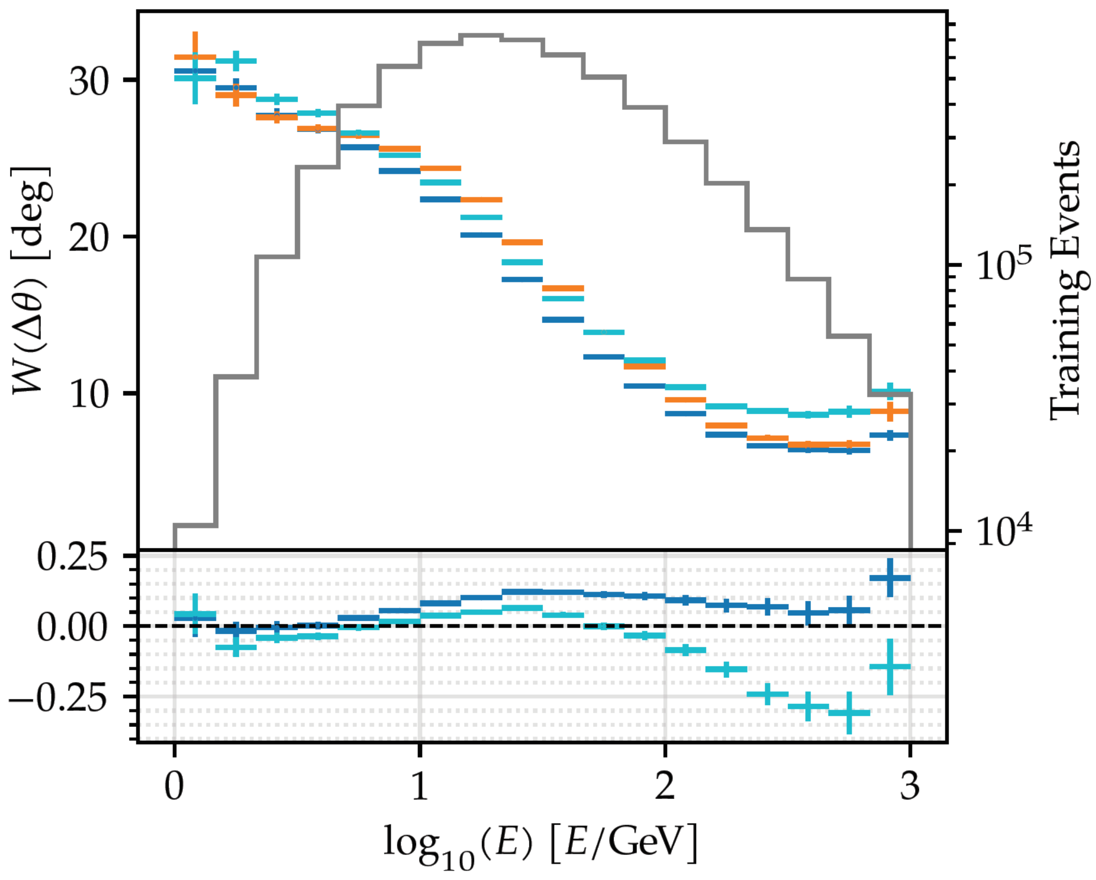
- Combine outputs of several models
- Control performance with weights





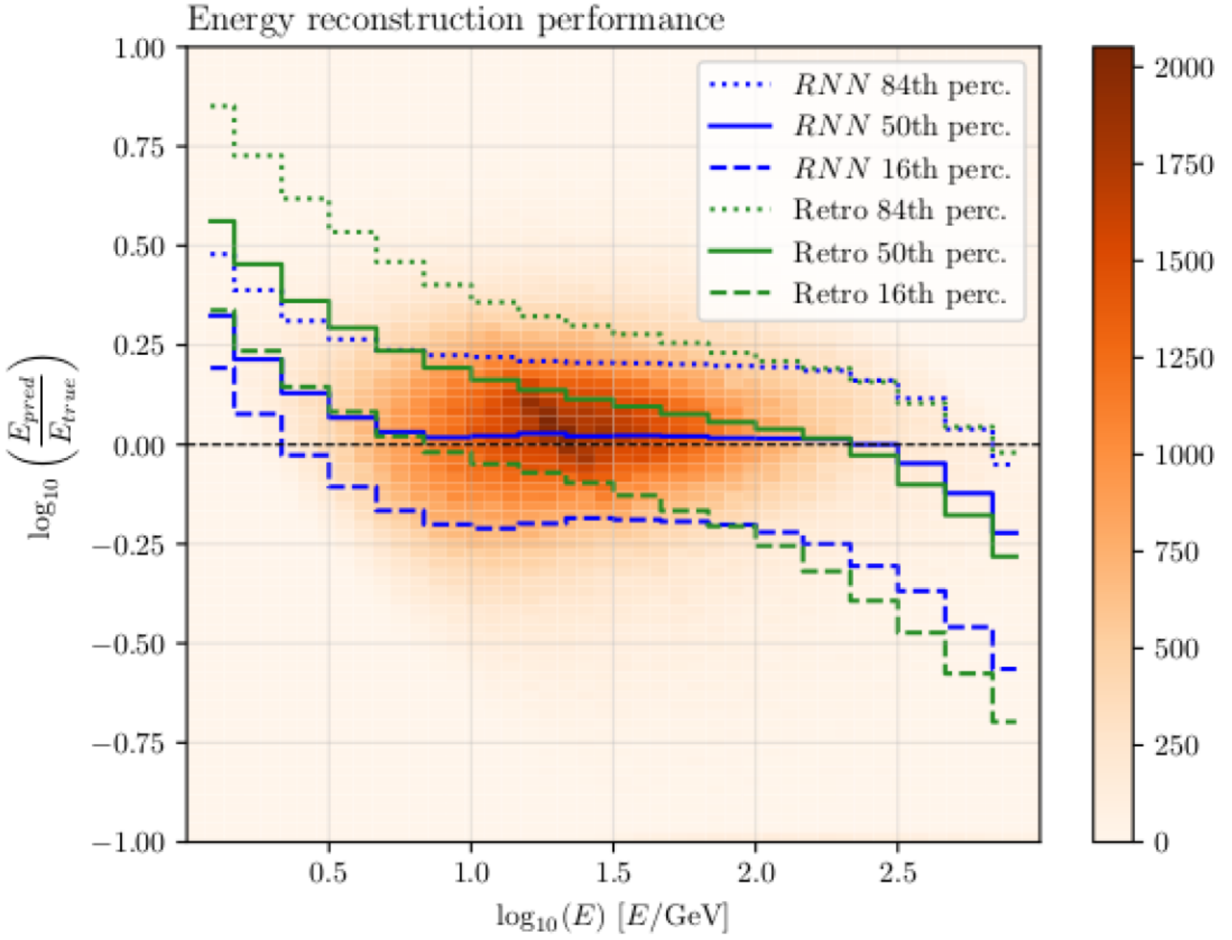
Ensemble Model – 8 submodels





Ensemble Model – bias (almost) gone

- Bias removed for energy reconstruction through reweighting
- Requires more data to remove bias in direction reconstruction
- Inference speed: 5000 Reconstructions per second



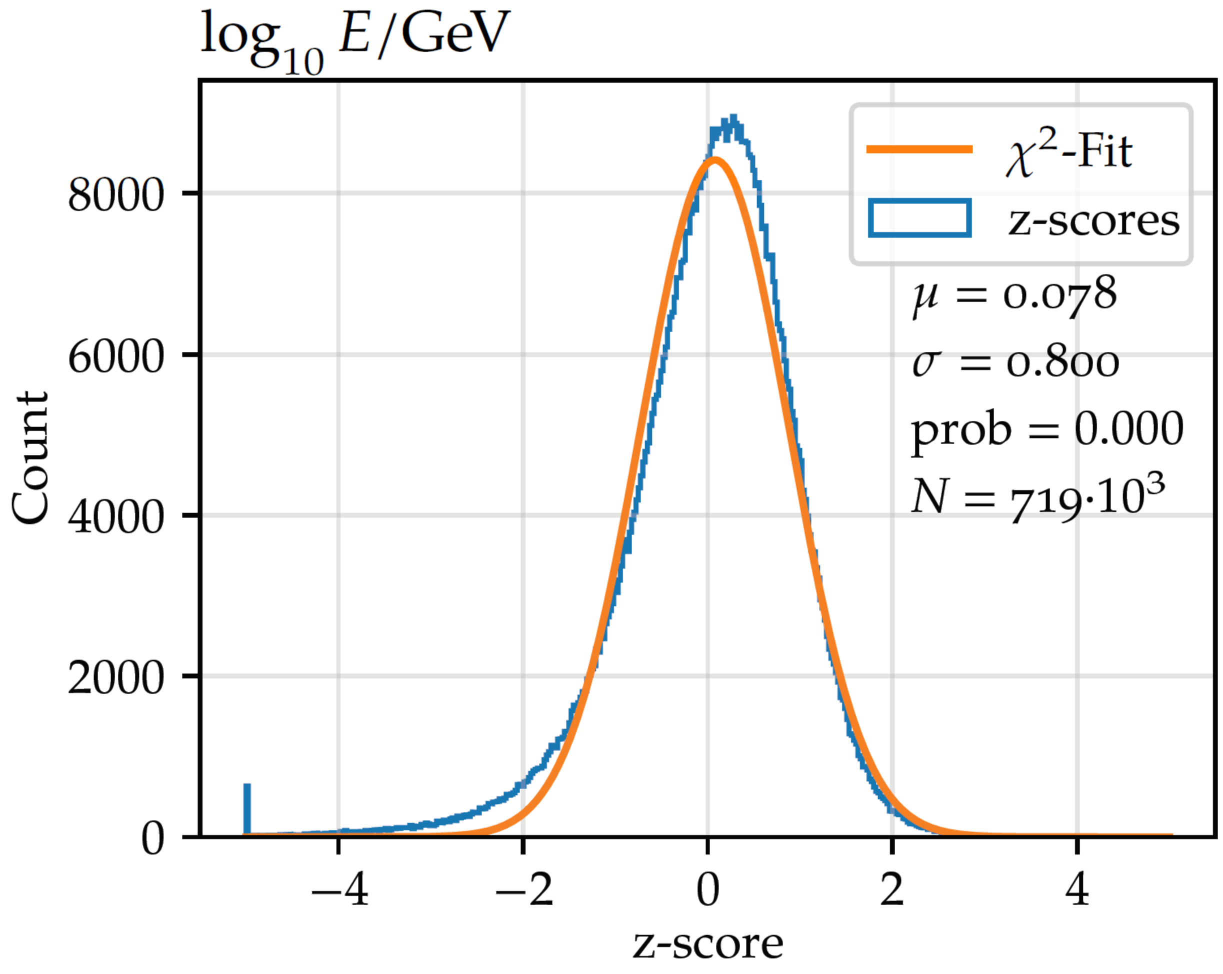
Ensemble Model – Error Estimates

- Train meta-learner with Gaussian LLH-loss

$$l(x, \mu, \sigma^2) = \log \sigma + \frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2$$

- z-score unit Gauss expected, where

$$z = \frac{X-\mu}{\sigma}$$



Outlook

- Inference speed: 5000 Reconstructions per second
- More data = better performance!
- Train noise + BG rejector

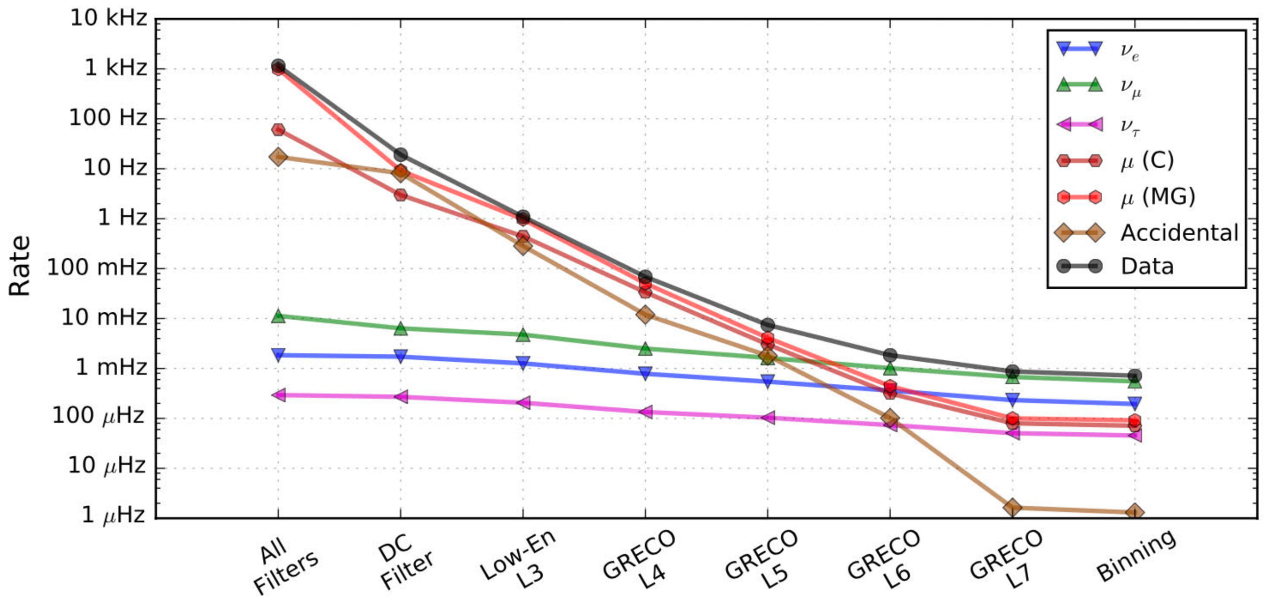
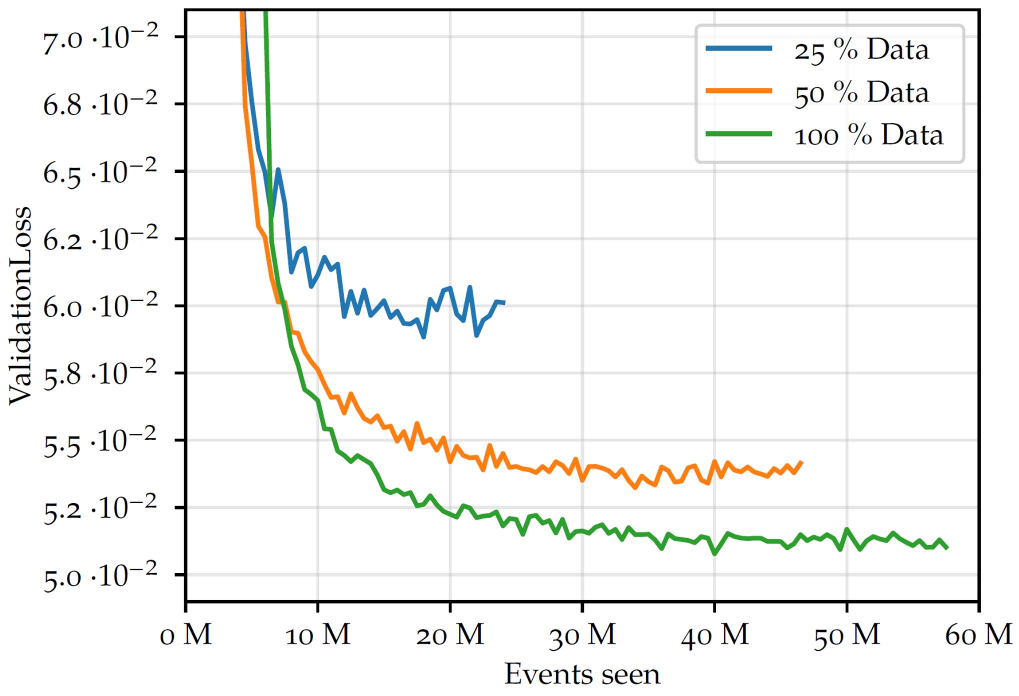


Figure courtesy of Michael Larson

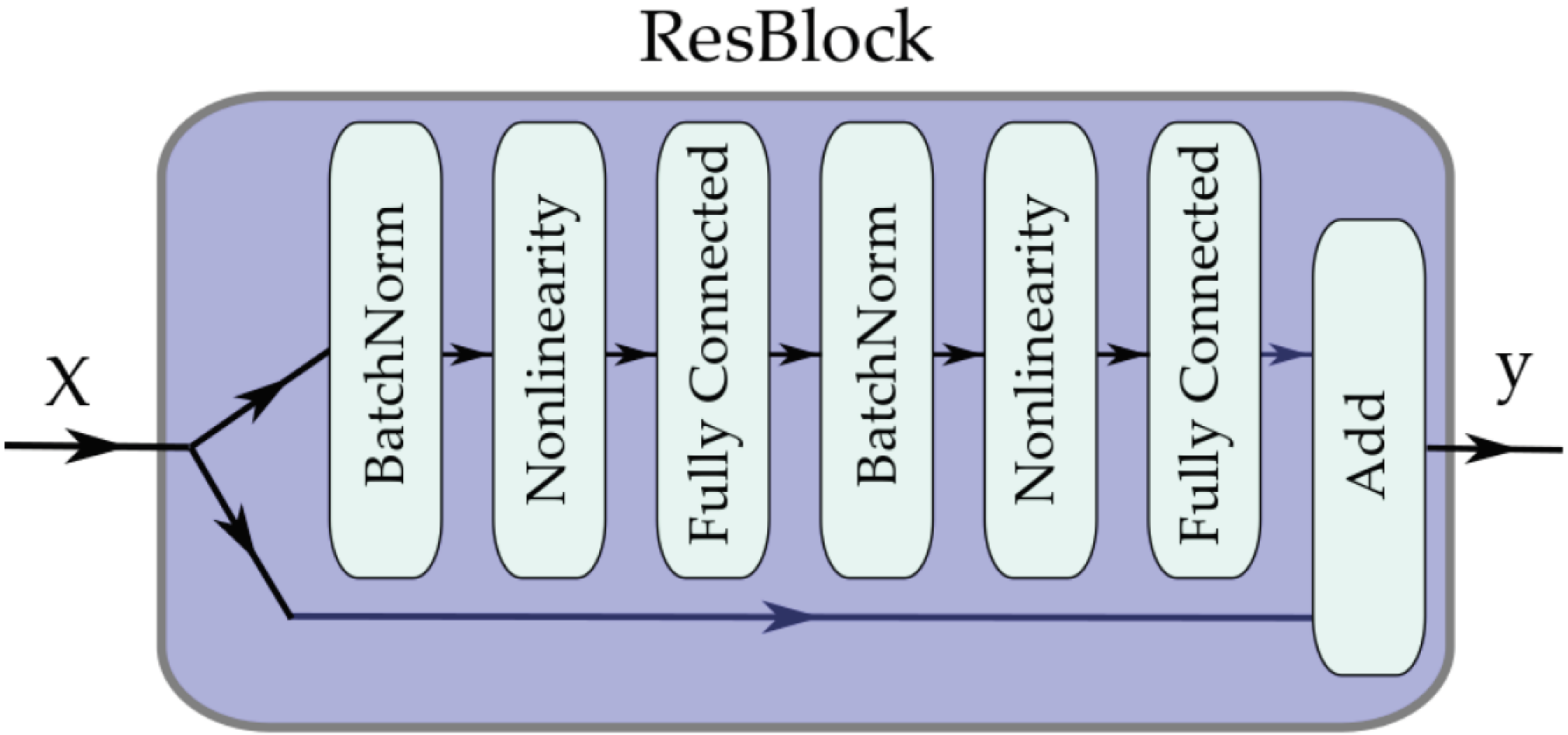


Thanks for listening!

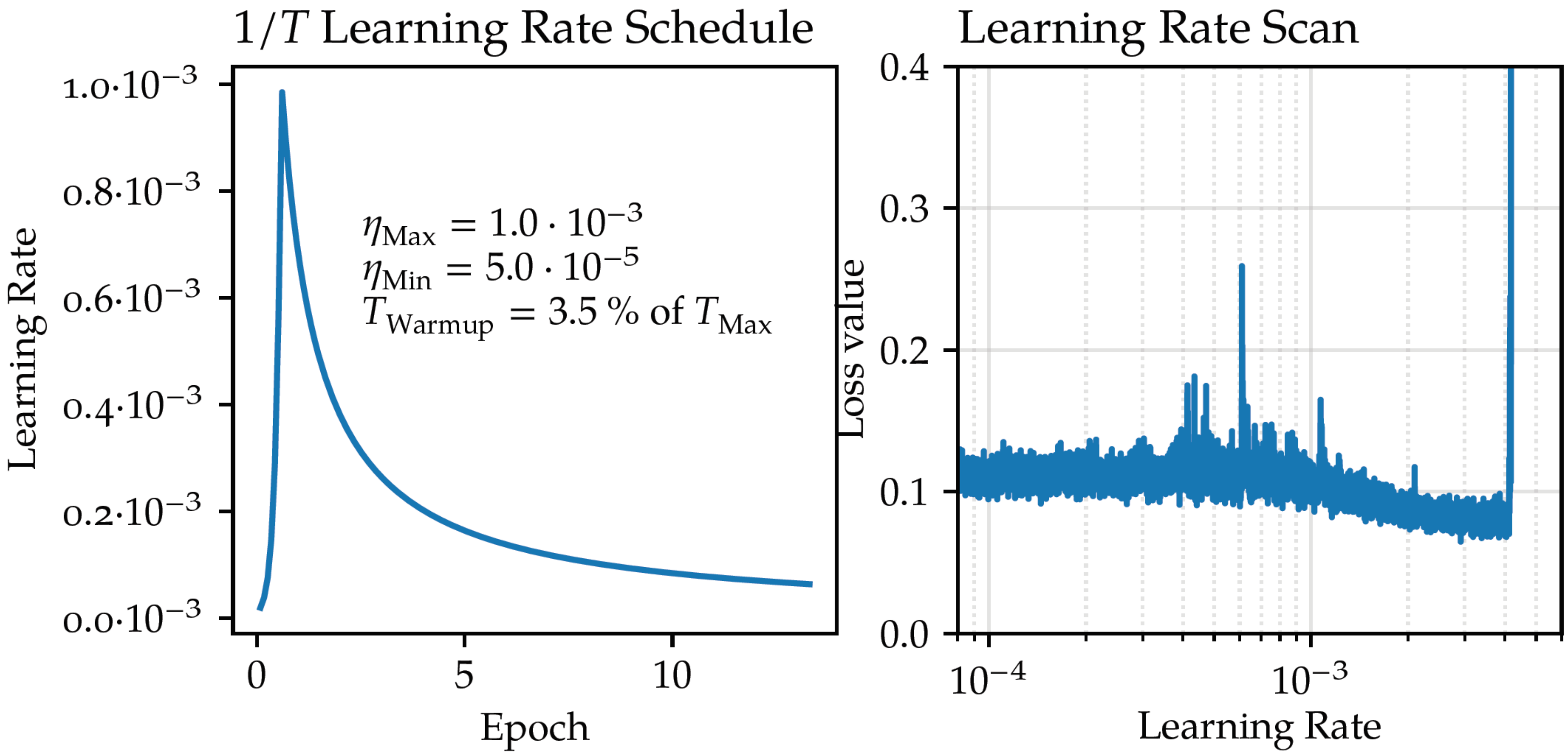
GRU Update rules

$$
\begin{aligned}
r_t &= \sigma\left(\boldsymbol{W}_{rx}\boldsymbol{x}_t + \boldsymbol{b}_{rx} + \boldsymbol{W}_{rh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{rh}\right),\\
z_t &= \sigma\left(\boldsymbol{W}_{zx}\boldsymbol{x}_t + \boldsymbol{b}_{zx} + \boldsymbol{W}_{zh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{zh}\right),\\
n_t &= \tanh\left(\boldsymbol{W}_{nx}\boldsymbol{x}_t + \boldsymbol{b}_{nx} + r_t * \left(\boldsymbol{W}_{nh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{gh}\right)\right),\\
h_t &= (1 - z_t) * n_t + z_t * h_{t-1},
\end{aligned}
$$

ResBlock


ResBlock
X
BatchNorm
Nonlinearity
Fully Connected
BatchNorm
Nonlinearity
Fully Connected
Add
y

Learning Rate Schedule

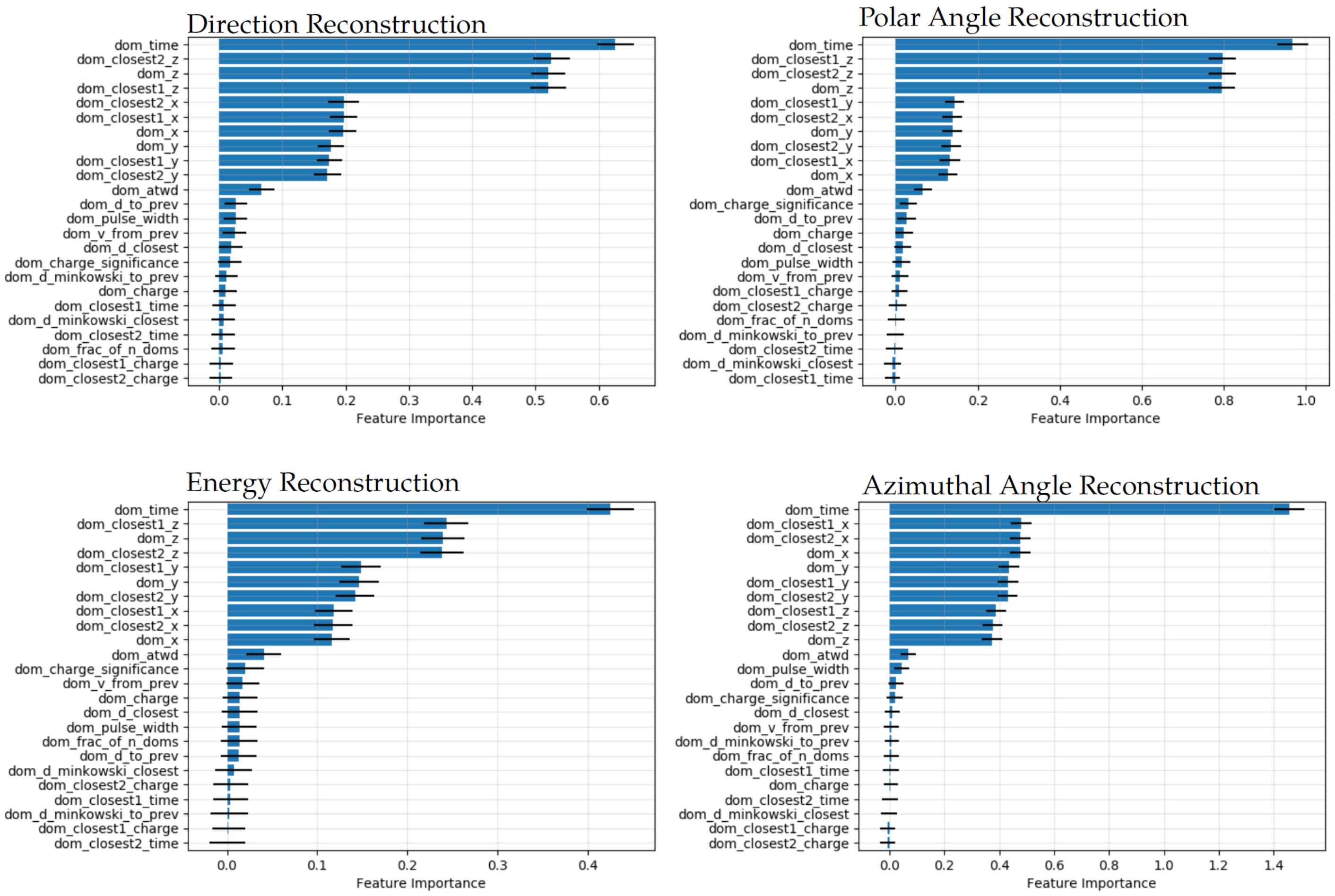


Hyperparameter Searchspace

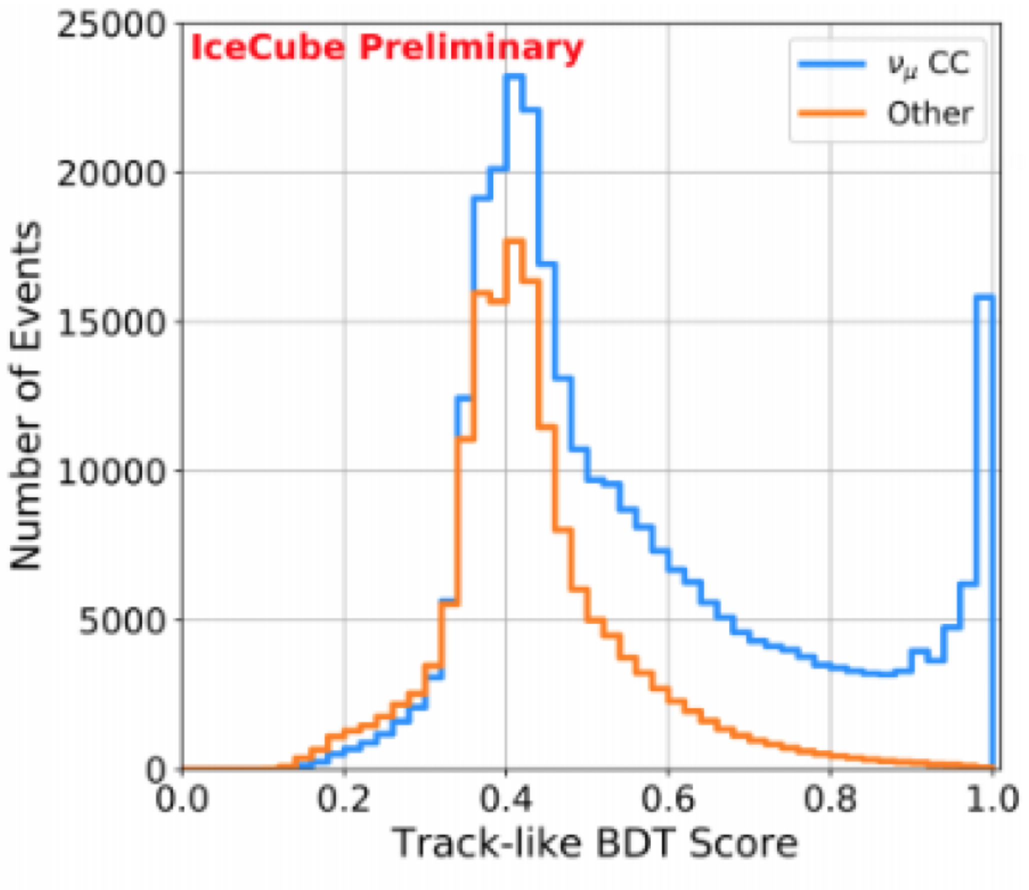
Hyperparameter	Searchspace
Batchsize	32, 64, 128, 256, 512
Optimizer	SGD, Adam, NAG
LR schedule	Inverse decay w. warmup
Layer Widths	64, 128, 256, 512, 1028
Decoding ResBlocks	0, 1, 2, 3, 4, 5, 6
Encoding Att. Blocks	0, 1, 2, 3, 4, 5, 6, 7
Encoding RNN layers	0, 1, 2, 3, 4
Encoding RNN type	Vanilla, GRU, LSTM, BiGRU, BiLSTM
Nonlinearity	LeakyReLU, Mish
Encoding norm.	None, LayerNorm
Decoding norm.	None, BatchNorm
Regularization	None, Dropout($p \in [30\%, 50\%, 80\%]$)
Regression loss	L1, L2, logcosh
Classification loss	CrossEntropy
Many-to-One	MaxPool, AvePool, KeepLast
Weight init.	Kaiming

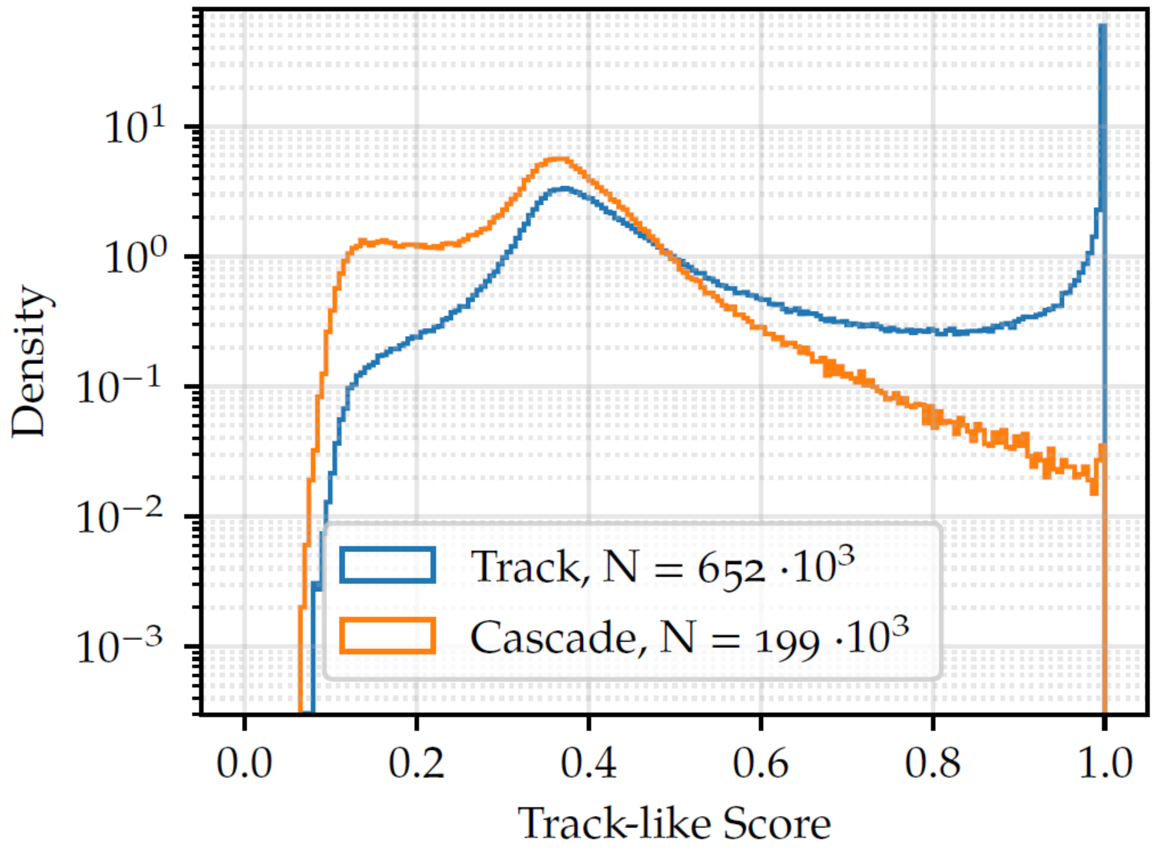
Table 4: Summary of the hyperparameter searchspace. The regression loss functions are not complete; regression-specific alterations are introduced later.

Permutation Feature Importance

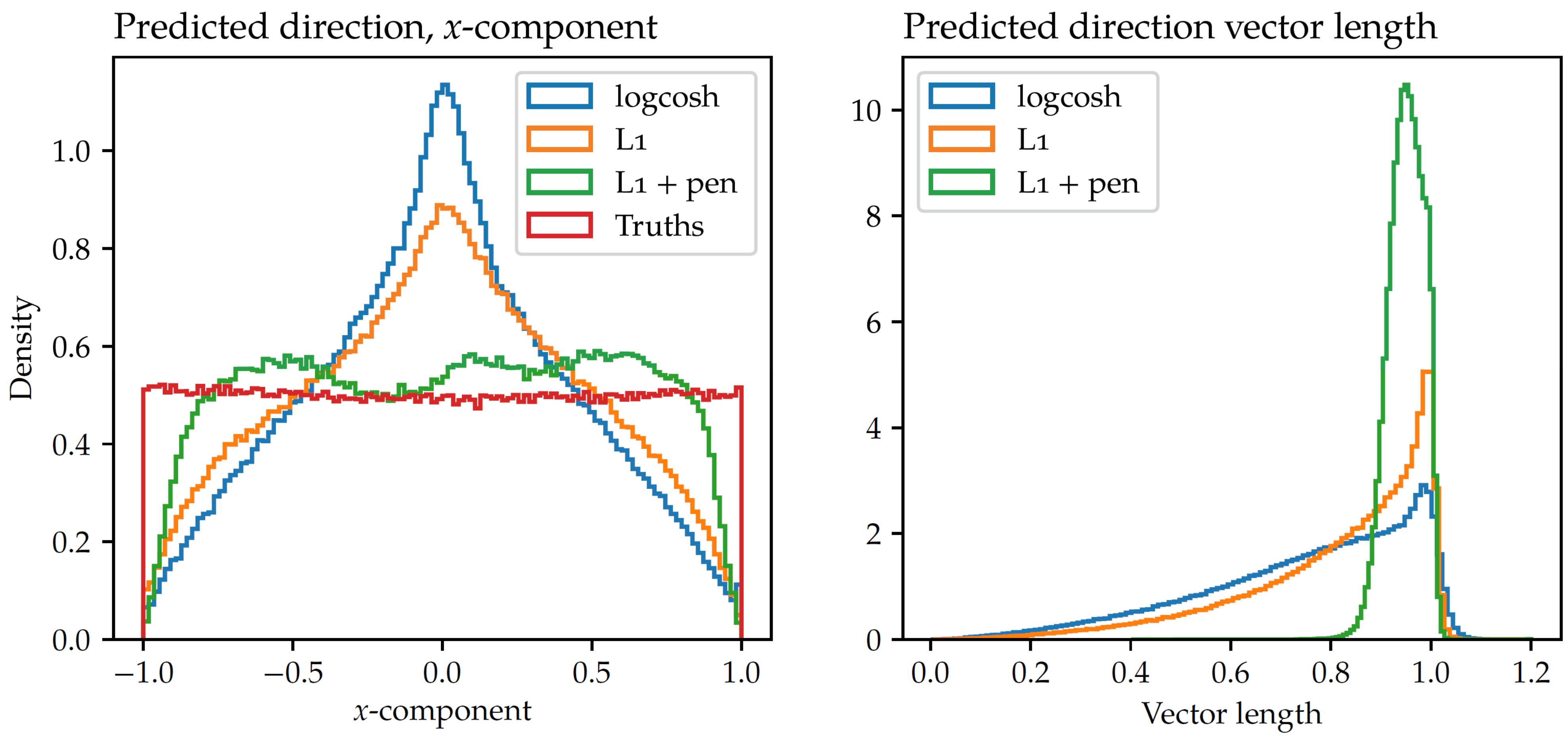


Classification - comparison





Direction unit vector distributions



Logarithmic vs Relative error