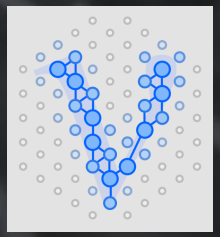
Ideas for usage of GraphNet in IceCube^{++}

Troels C. Petersen
Niels Bohr Institute (Univ. of Copenhagen)



Bornholm, 3rd GraphNet workshop, 3. May 2023

Overview

Tasks using Monte Carlo only
Large scale neutrino selection in data
An AtmosphericEvent tagger (trained in data?)
Real-time analysis/alerts
Algorithm development
Explaining / visualising GNN output
Thoughts on where to begin

Task using Monte Carlo only

Tasks using MC only

1. **High Energy neutrino classification and reconstruction**
   Somewhat repeating what we've done for low energy neutrinos, without the systematic variations, but with algorithm re-optimisation.
   Potential dataset: SnowStorm. Obvious target for a paper.

2. **GNN pulse cleaning**
   This has been worked on before, but might be revisited with GraphNet.

3. **Upgrade reconstruction**
   Again worked on before, but still important for detector optimisation.
   Status of MC from Tom? Target of a paper?

4. **Elasticity regression (for Nu vs. NuBar)**
   This should not be done for all muon neutrinos, but only those clearly identified by the Track-Cascade and with significant energy (100+ GeV?).
   Spencer might have a student on this? Also, Retro provides estimates?

Tasks using MC only

1. **High Energy neutrino classification and recon**
   Somewhat repeating what we've done for low ener… …thout the systematic variations, but with algorithm… …. Potential dataset: SnowStorm. Obvious… …per.

   Worked on (Rasmus, Andreas, etc.)
   Status: SplineMPE still better

2. **GNN pulse cleaning**
   This has been worked on before, but might be revisited with GraphNet.

3. **Upgrade reconstruction**
   Again worked on before, but still important for detector optimisation.
   Status of MC from Tom? Target of a paper?

4. **Elasticity regression (for Nu vs. NuBar)**
   This should not be done for all muon neutrinos, but only those clearly identified by the Track-Cascade and with significant energy (100+ GeV?).
   Spencer might have a student on this? Also, Retro provides estimates?

Tasks using MC only

1. **High Energy neutrino classification and recon**
   Somewhat repeating what we've done for low ener… …thout the systematic variations, but with algorithm… …. Potential dataset: SnowStorm. Obvious… …per.

   Worked on (Rasmus, Andreas, etc.)
   Status: SplineMPE still better

2. **GNN pulse cleaning**
   This has been worked on before, but might be revi… …h GraphNet.

   V1 Done (Thomas, Rasmus)

3. **Upgrade reconstruction**
   Again worked on before, but still important for detector optimisation.
   Status of MC from Tom? Target of a paper?

4. **Elasticity regression (for Nu vs. NuBar)**
   This should not be done for all muon neutrinos, but only those clearly identified by the Track-Cascade and with significant energy (100+ GeV?).
   Spencer might have a student on this? Also, Retro provides estimates?

Tasks using MC only

1. **High Energy neutrino classification and recon**
   Somewhat repeating what we've done for low ener… …thout the systematic variations, but with algorithm… …. Potential dataset: SnowStorm. Obvious… …per.

2. **GNN pulse cleaning**
   This has been worked on before, but might be revi… …h GraphNet

3. **Upgrade reconstruction**
   Again worked on before, but still important for de… …timisation. Status of MC from Tom? Target of a paper?

4. **Elasticity regression (for Nu vs. NuBar)**
   This should not be done for all muon neutrinos, but only those clearly identified by the Track-Cascade and with significant energy (100+ GeV?). Spencer might have a student on this? Also, Retro provides estimates?

Worked on (Rasmus, Andreas, etc.)
Status: SplineMPE still better

V1 Done (Thomas, Rasmus)

V1 Done (Kaare, Moust, Rasmus)

Tasks using MC only

1. **High Energy neutrino classification and recon**
   Somewhat repeating what we've done for low ener… …thout the systematic variations, but with algorithm… …
   Potential dataset: SnowStorm. Obvious… …per.

2. **GNN pulse cleaning**
   This has been worked on before, but might be revi… …n GraphNet

3. **Upgrade reconstruction**
   Again worked on before, but still important for de… …timisation.
   Status of MC from Tom? Target of a paper?

4. **Elasticity regression (for Nu vs. NuBar)**
   This should not be done for all muon neutrinos, but o… …clearly identified by the Track-Cascade and with signifi… …rgy (100+ GeV?).
   Spencer might have a student on this? Al… …provides estimates?

Worked on (Rasmus, Andreas, etc.)
Status: SplineMPE still better

V1 Done (Thomas, Rasmus)

V1 Done (Kaare, Moust, Rasmus)

Work in progress (Moust)… hard!

Tasks using MC only

1. **High Energy neutrino classification and recon**
   Somewhat repeating what we've done for low ener… …thout the systematic variations, but with algorithm… …
   Potential dataset: SnowStorm. Obvious… …per.

2. **GNN pulse cleaning**
   This has been worked on before, but might be revi… …n GraphNet

3. **Upgrade reconstruction**
   Again worked on before, but still important for de… …timisation.
   Status of MC from Tom? Target of a paper?

4. **Elasticity regression (for Nu vs. NuBar)**
   This should not be done for all muon neutrinos, but o… …clearly identified by the Track-Cascade and with signifi… …rgy (100+ GeV?).
   Spencer might have a student on this? Al… …provides estimates?

Certainly to be revised using the GNN-Transformer architecture.

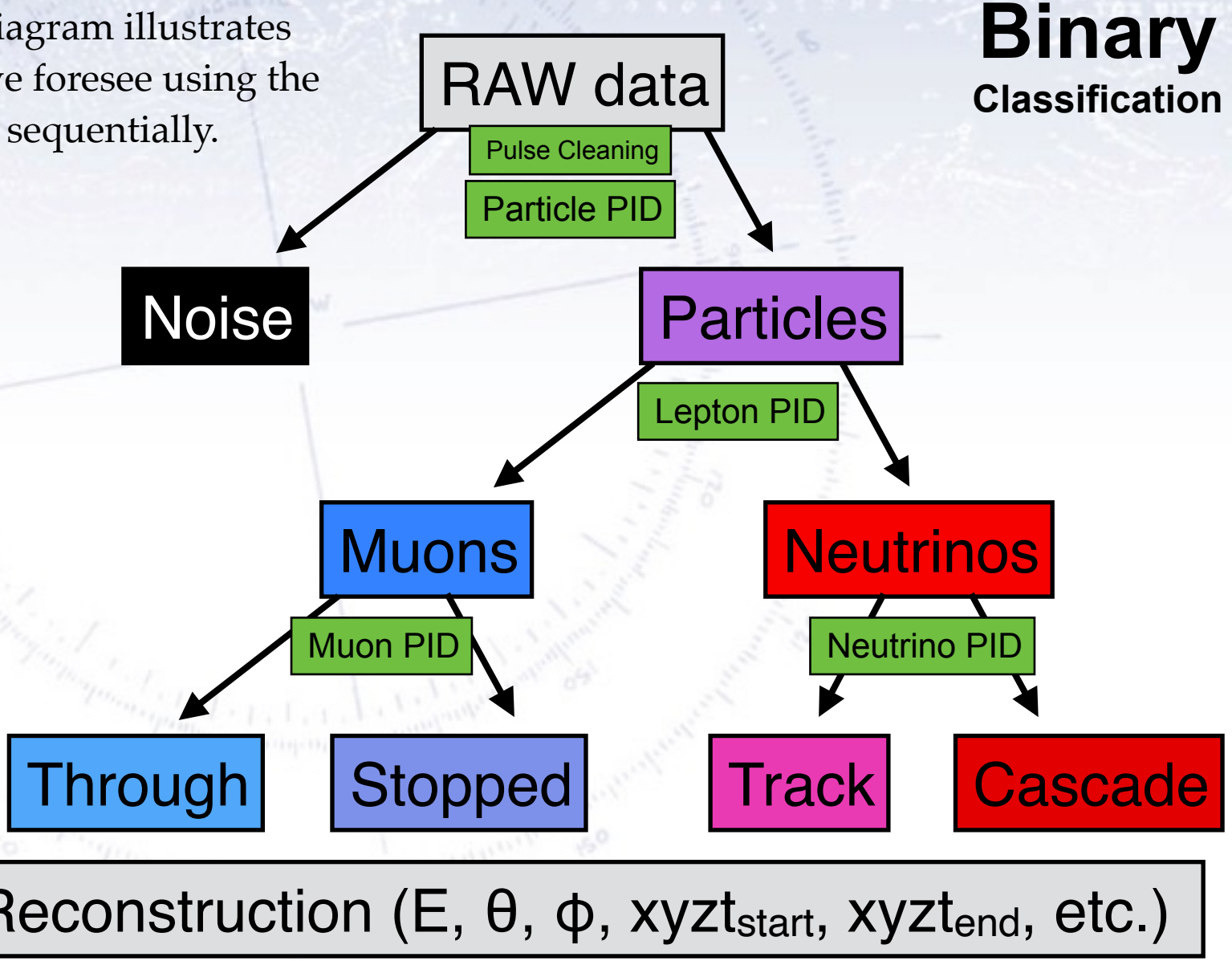
V1 Done (Thomas, Rasmus)

V1 Done (Kaare, Moust, Rasmus)

Work in progress (Moust)… hard!

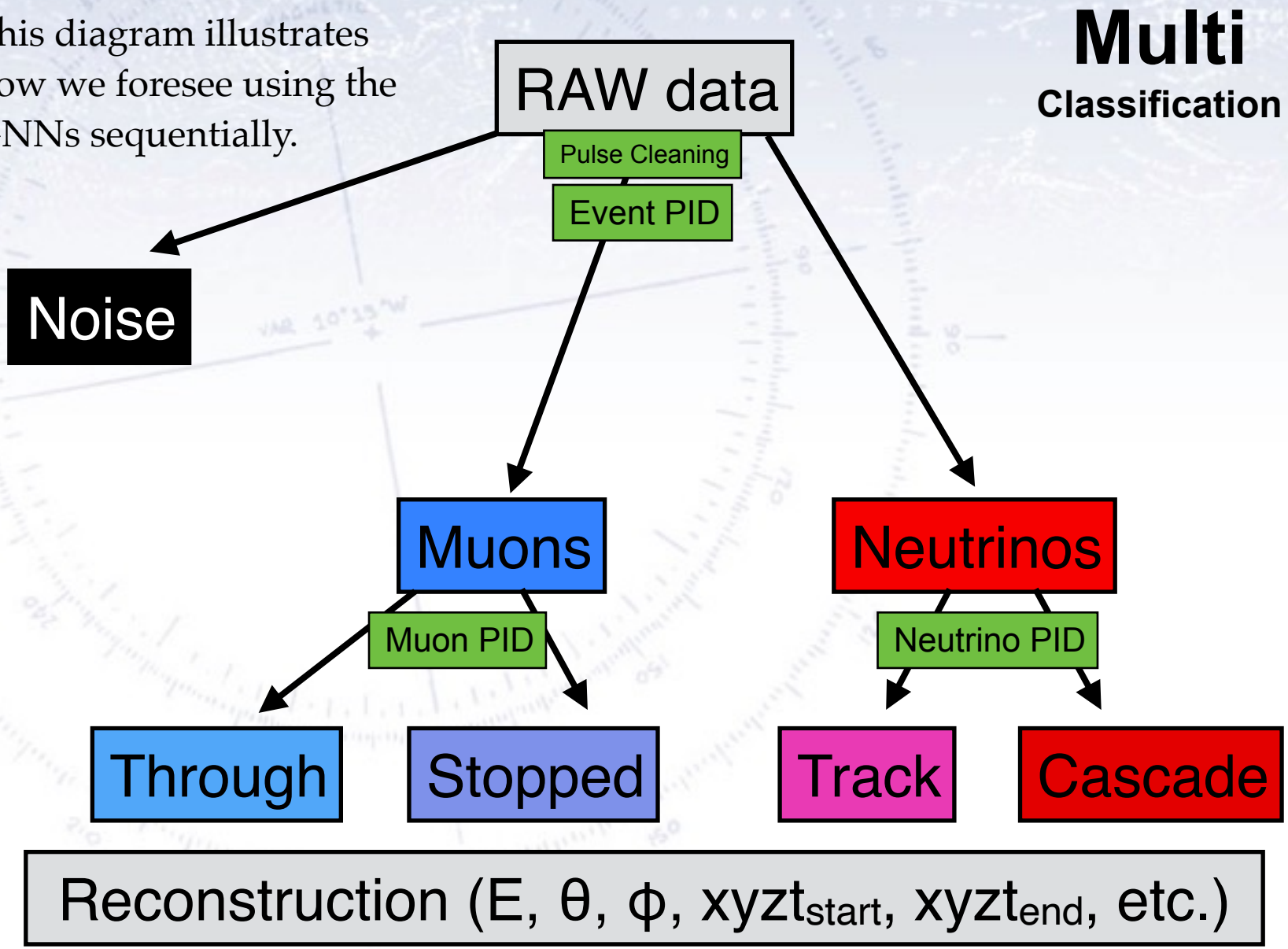
GNN classification overview

This diagram illustrates how we foresee using the GNNs sequentially.



GNN classification overview

This diagram illustrates how we foresee using the GNNs sequentially.



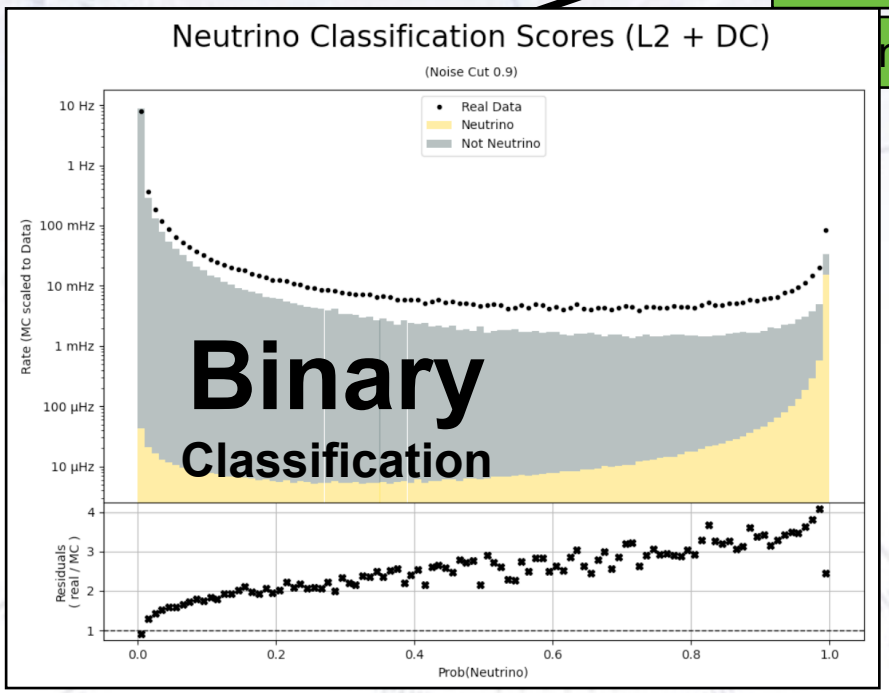
GNN classification overview

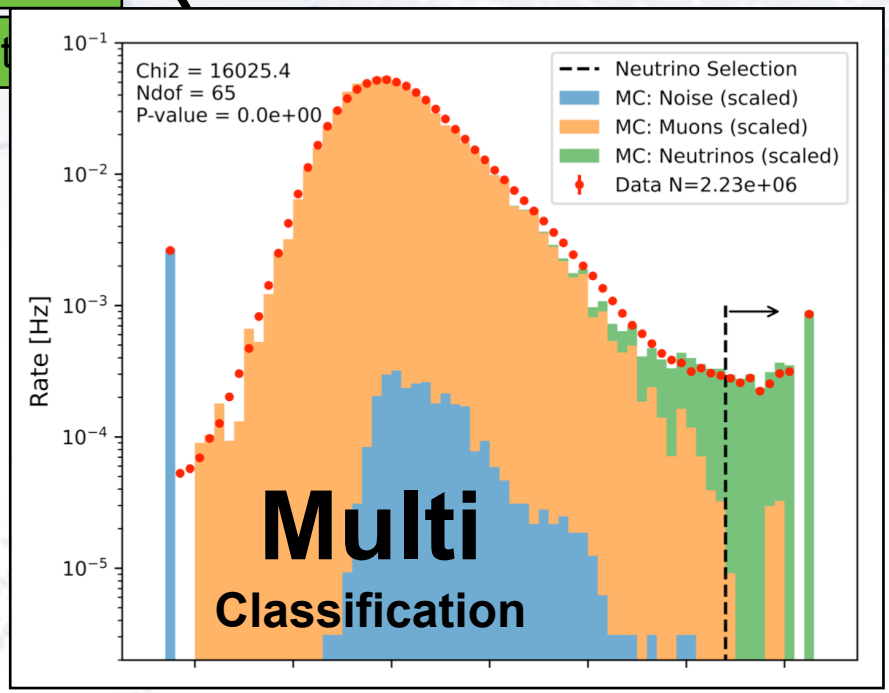
This diagram illustrates how we foresee using the GNNs sequentially.

RAW data

Pulse Cleaning

Multi
Classification





Through | Stopped | Track | Cascade

Reconstruction (E, θ, ϕ, $xyzt_{start}$, $xyzt_{end}$, etc.)

A full GNN ν data selection

A “real data campaign”?

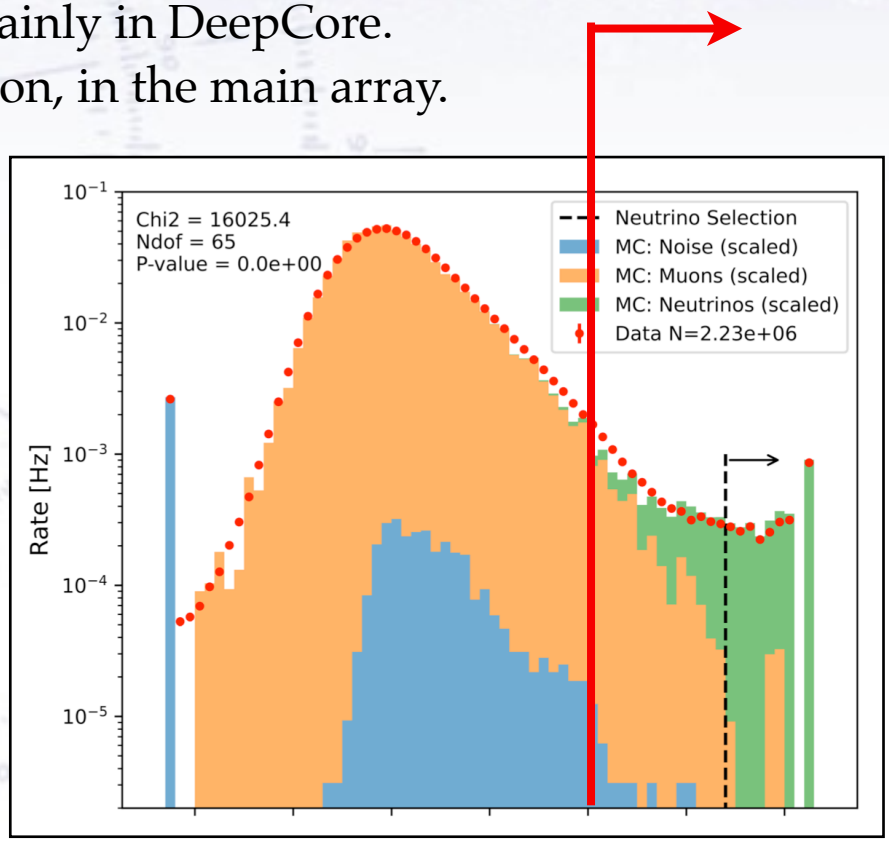
Given a good selection of neutrinos in data at Lvl2, I propose that we make “the high efficiency GNN neutrino selection”:

The two GNN ingredients would be:

- A low energy (1-1000 GeV) selection, mainly in DeepCore.
- A high energy (100-infinity GeV) selection, in the main array.

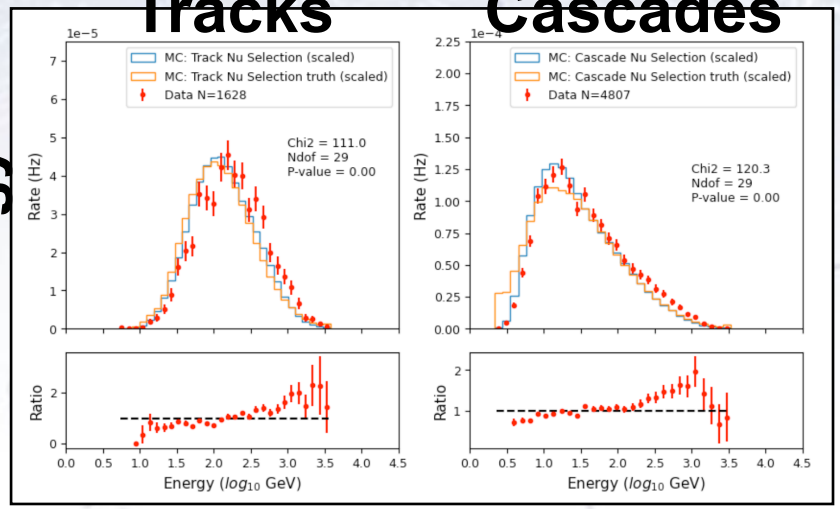
If any of the two selection algorithms go above a certain threshold, the event is selected.
Choosing VERY LOOSE thresholds will yield ~20 million events with ~2 million neutrinos in.

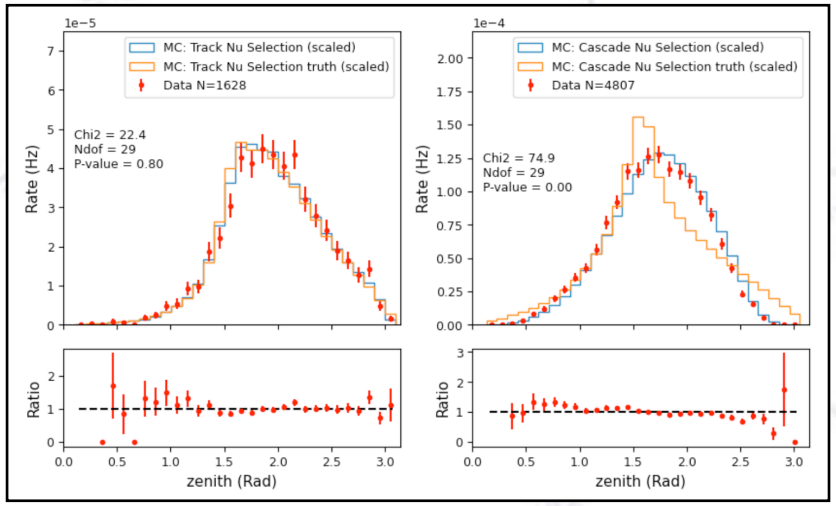
But given such a sample, one can run **improved further selections on this sample overnight on a single GPU!**

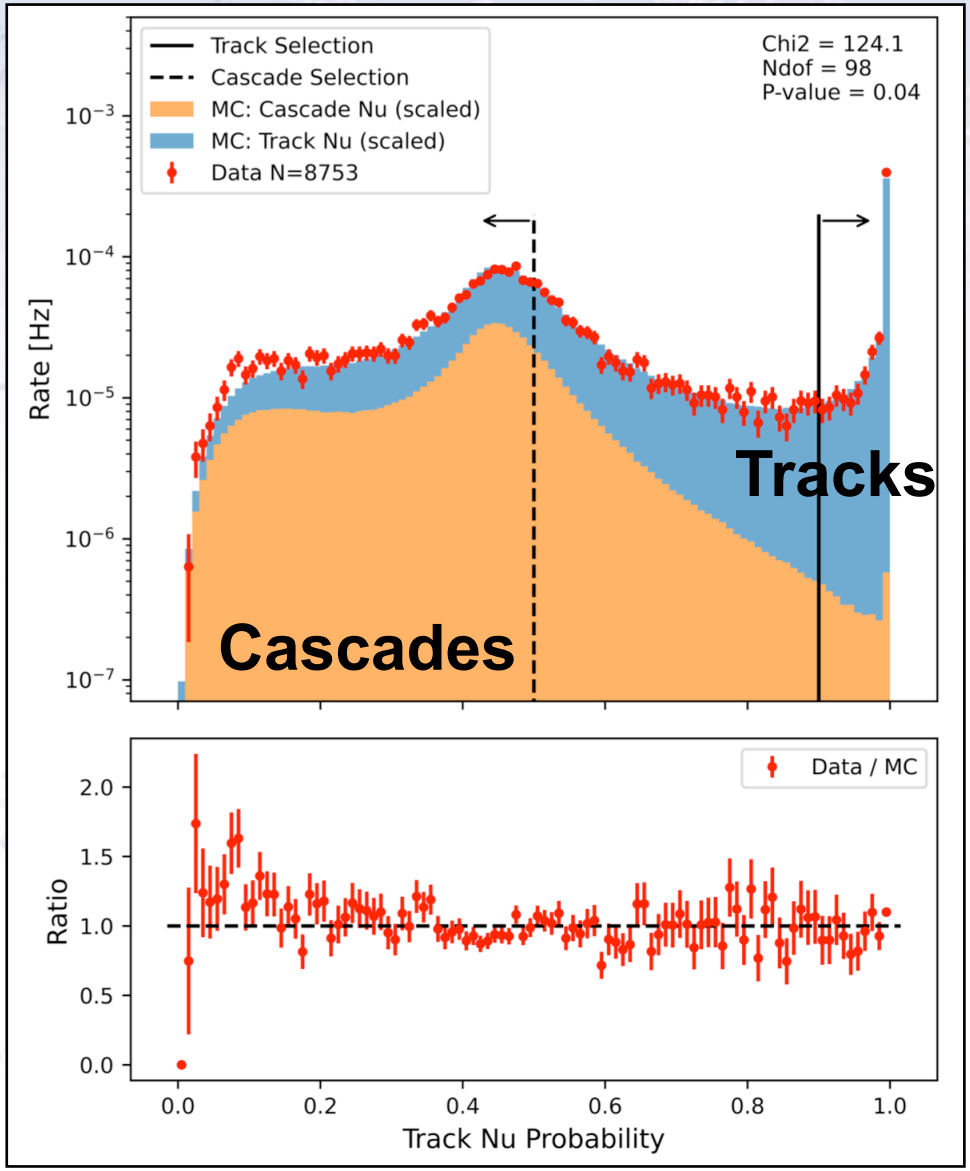


Low Energy Selection/Reco status

Things generally look "good"… :-)



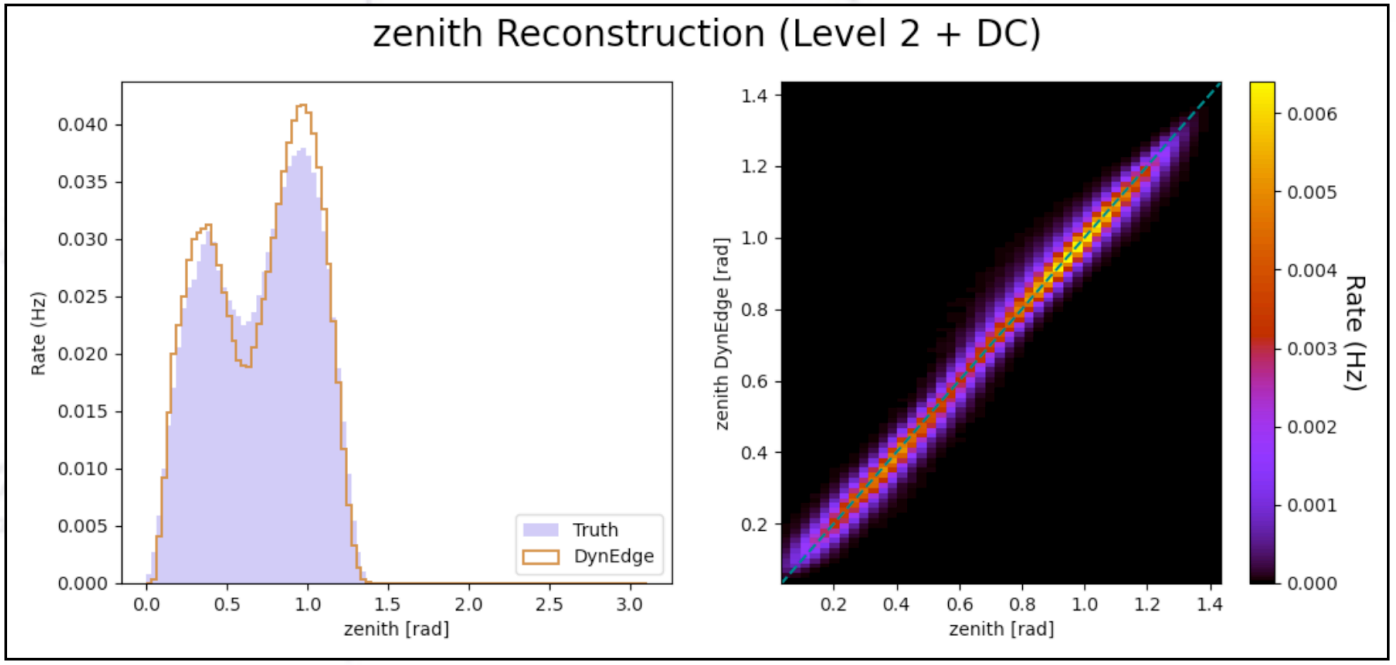




Using (stopped?) Muons

Larger scale muon selection in data

1. **Stopped Muons data-MC: Efficiencies / Systematics / Corrections**
   This is IMHO the obvious data-MC calibration sample, from which one can extract differential efficiency corrections and related systematics. This also follows Leon's work.



Larger scale muon selection in data

1. **Stopped Muons data-MC: Efficiencies / Systematics / Corrections**
   This is IMHO the obvious data-MC calibration sample, from which one can extract differential efficiency corrections and related systematics. This also follows Leon's work.

2. **Moon pointing analysis with muons**
   The moon pointing analysis is the only way to convince ourselves and others, that the GNN angular resolution is very good and measure it.

3. **Muon decay detection (for Nu vs. NuBar)**
   Given a large sample of stopped muons, one can in data search or signals from the muon decay. They will probably rarely produce enough light for a signal to be detected, but if they do, it would be a way to identify muons with high certainty, and statistically separate into neutrinos/antineutrinos.

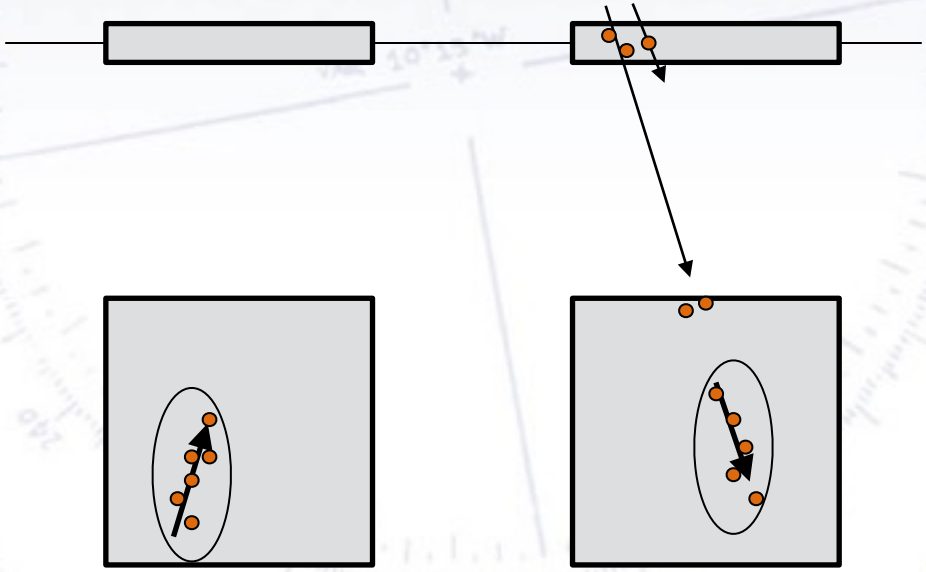
An AtmosphericEvent tagger?

Atmospheric Tagger

Idea: Use all other event activity (including IceTop) to detect if neutrino origin is atmospheric:

Up-going neutrino:
No other activity

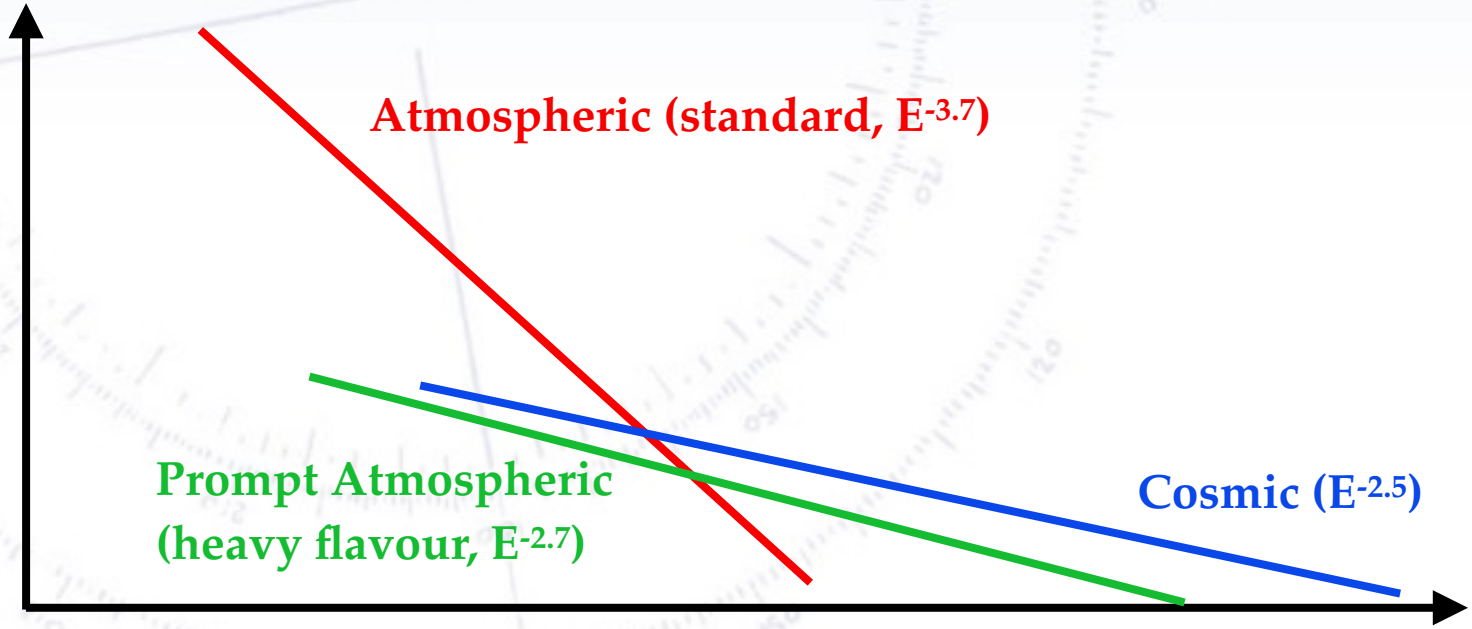
Down-going neutrino
Potential other activity



By considering neutrinos in DATA, and dividing them according to their direction (up- or down-going), and removing the pulses thought to be from the neutrinos, one obtains a sample to train an "atmospheric tagger" on.

Atmospheric Tagger

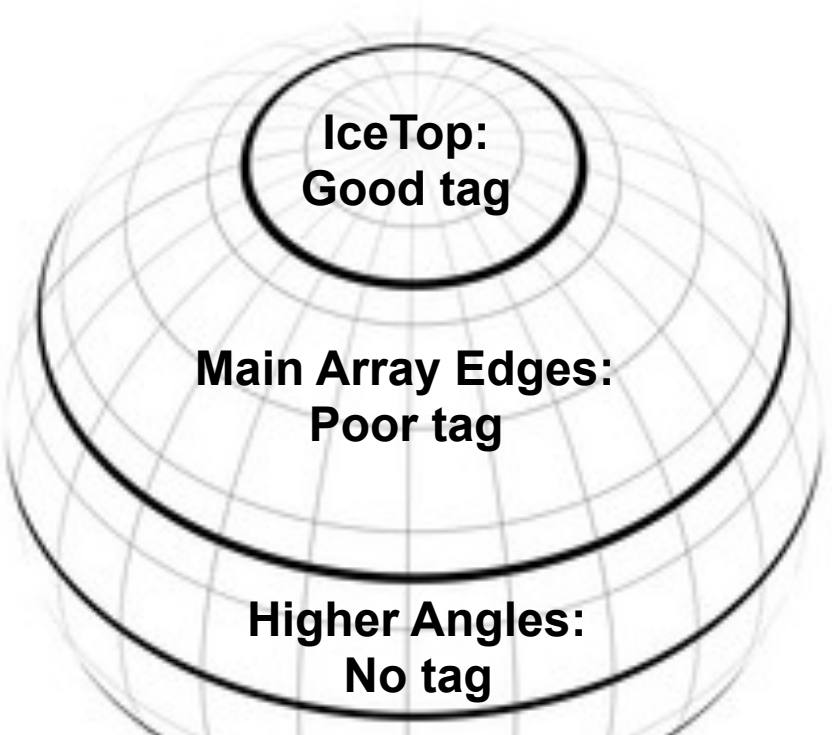
1. **For measuring hard atmospheric component**
   The heavy quarks produced by cosmic rays provide a neutrino spectrum, that falls less sharply, thus becoming the dominant atmospheric neutrinos at higher energies. But here the cosmic neutrinos dominate!
   However, by requiring that neutrinos have an atmospheric "tag", the cosmic component should disappear (idea from discussions with Mirco).



Atmospheric Tagger

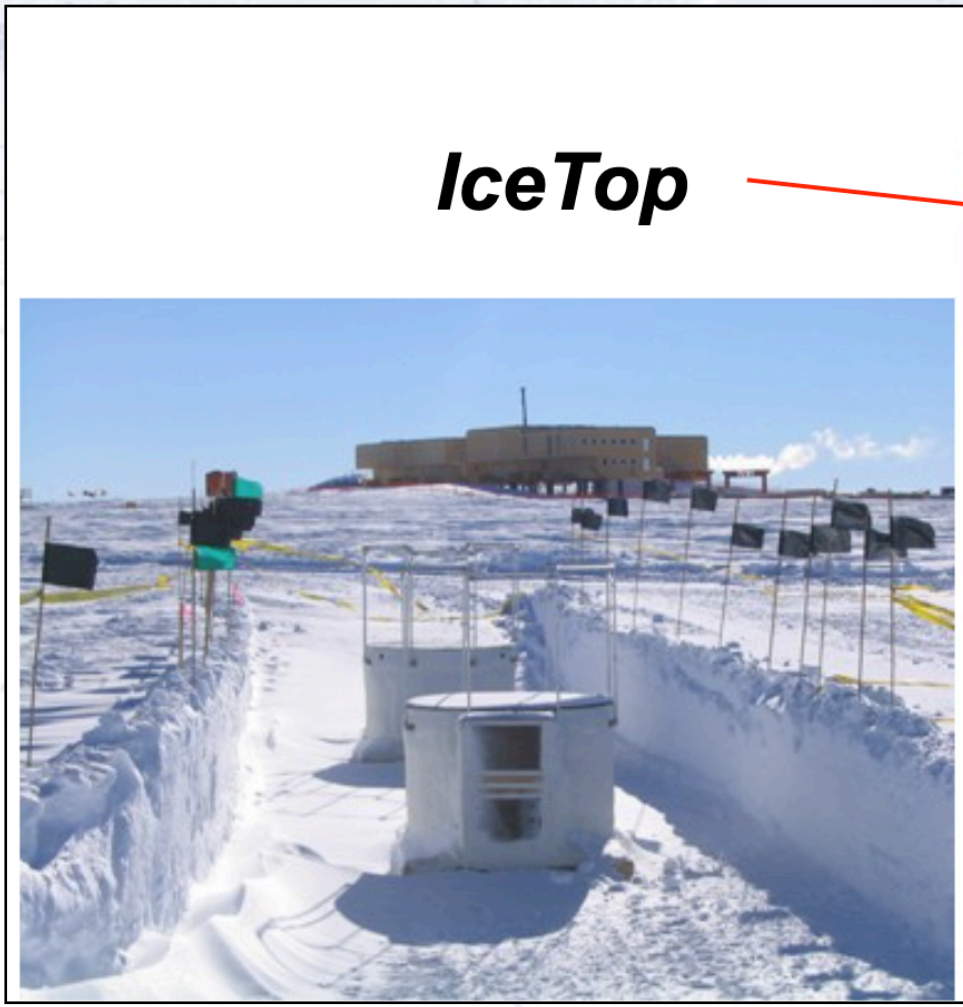
1. **For measuring hard atmospheric component**
   The heavy quarks produced by cosmic rays provide a neutrino spectrum, that falls less sharply, thus becoming the dominant atmospheric neutrinos at higher energies. But here the cosmic neutrinos dominate!
   However, by requiring that neutrinos have an atmospheric "tag", the cosmic component should disappear (idea from discussions with Mirco).

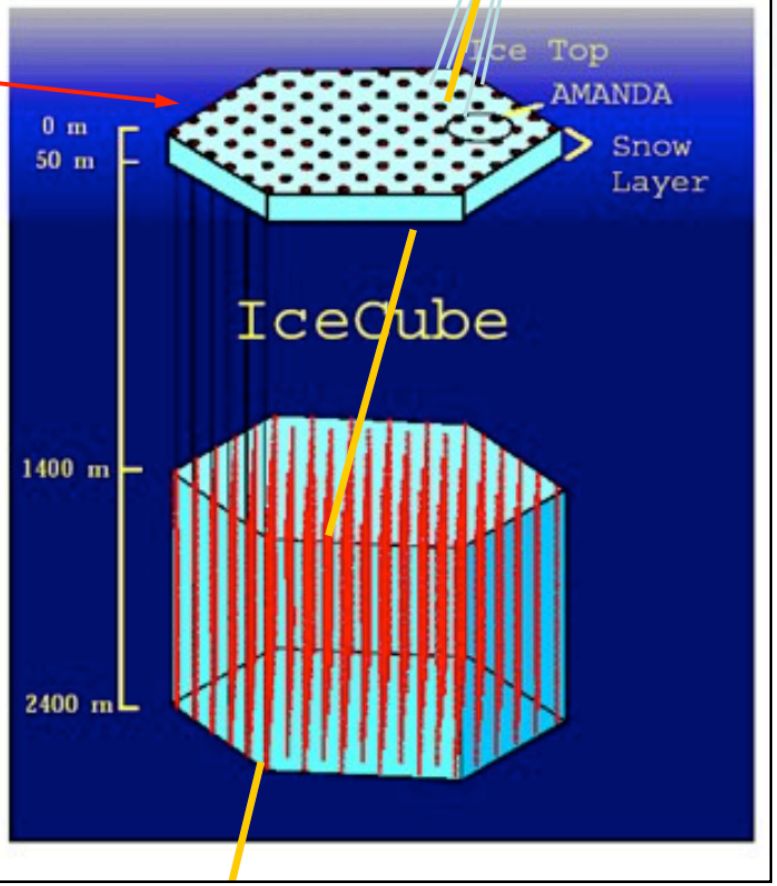
2. **For improving (low energy) alerts**
   Conversely, removing neutrino candidates with an atmospheric "tag" will yield a more pure cosmic sample, especially at low energy. However, this only works for down-going neutrinos.



Using IceTop

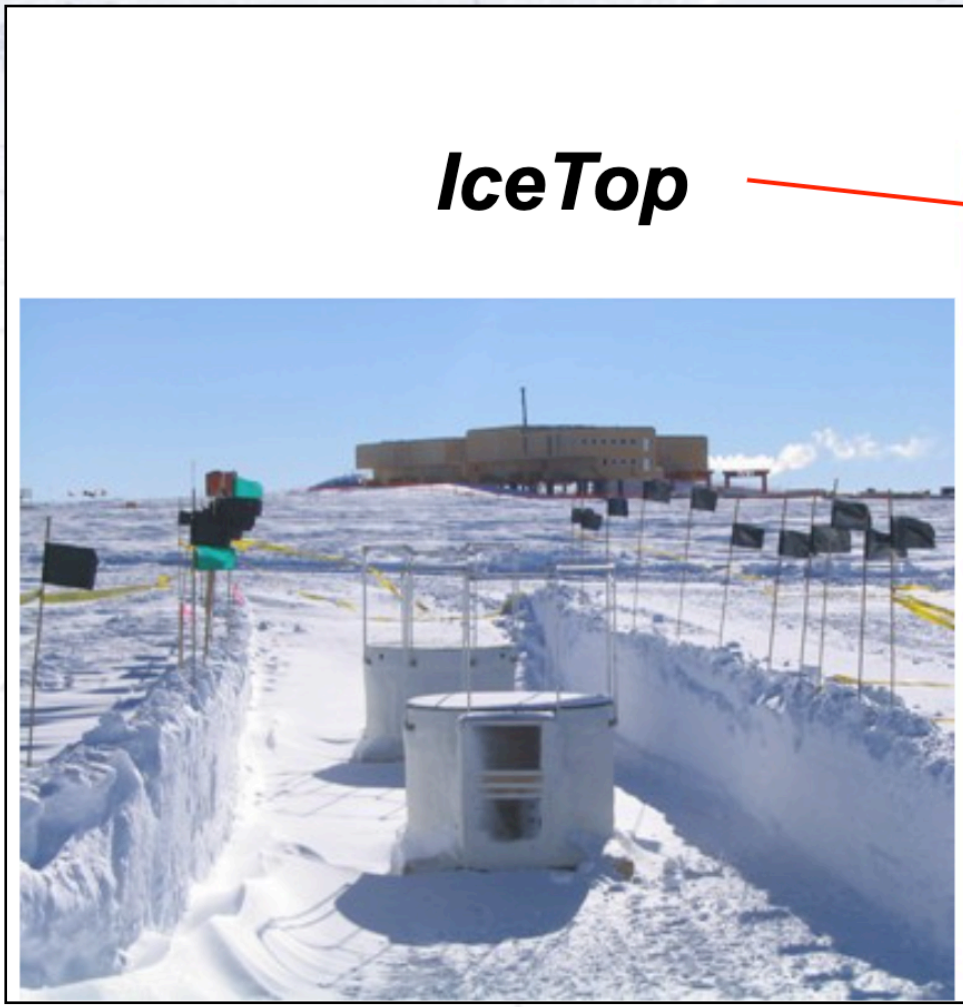
IceTop has good angular resolution of atmospheric shower:
This can be used for training neutrino angular reconstruction IN DATA!

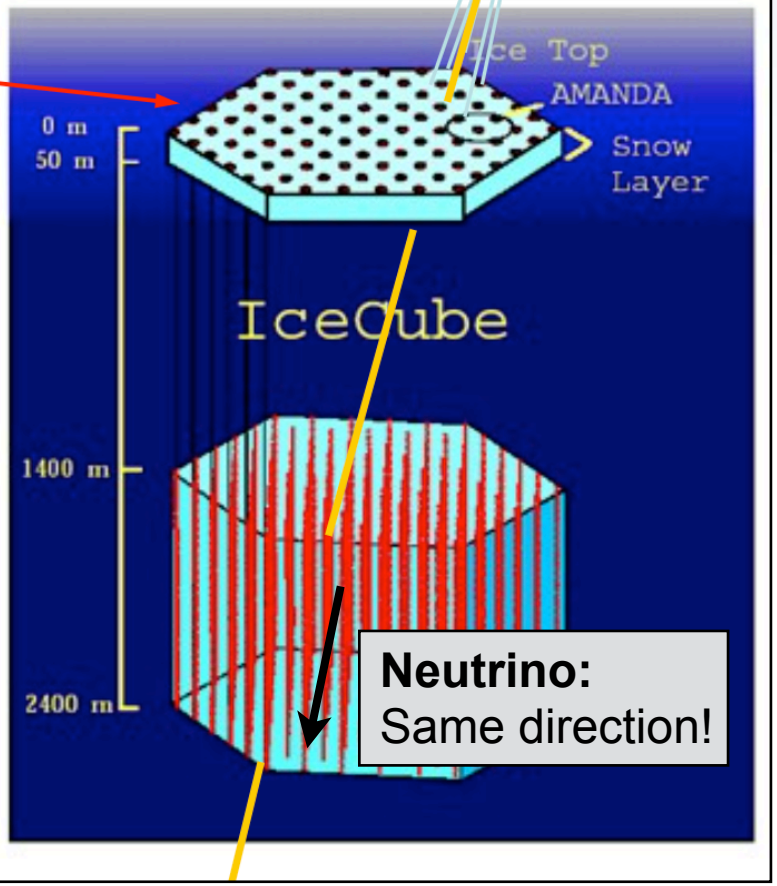




Using IceTop

IceTop has good angular resolution of atmospheric shower:
This can be used for training neutrino angular reconstruction IN DATA!

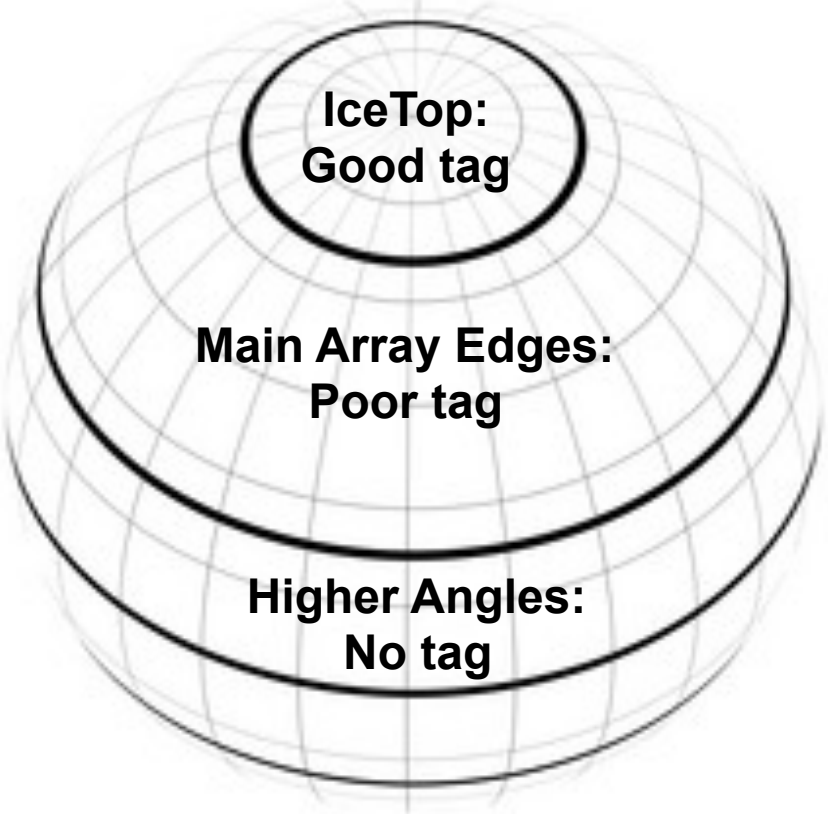


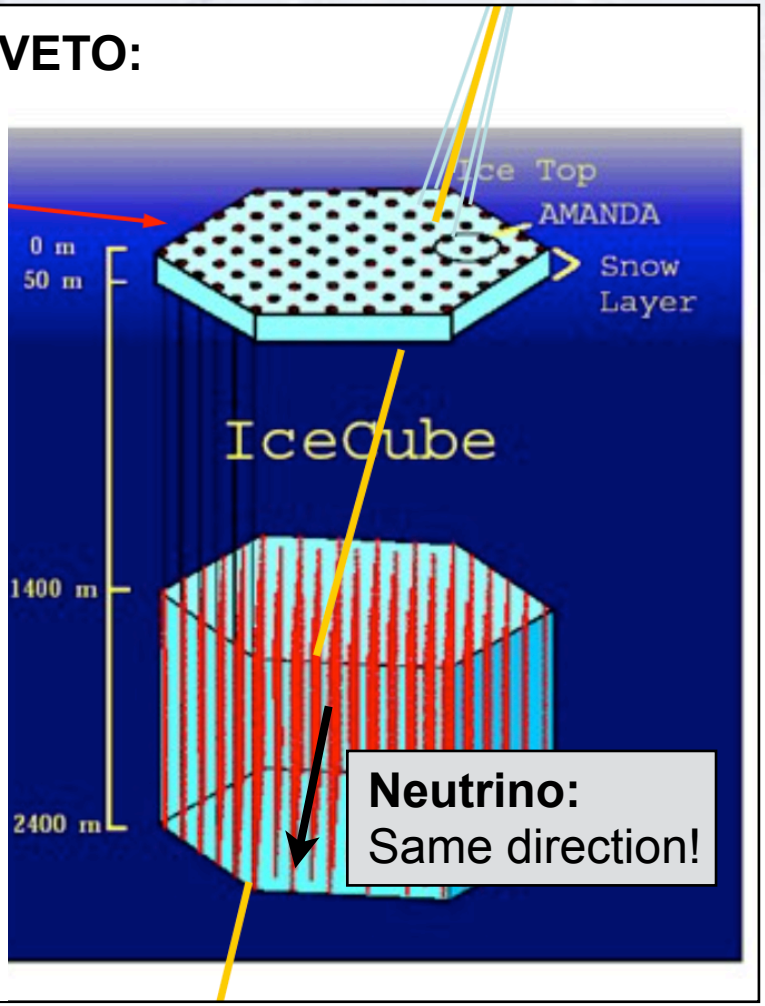


Using IceTop

IceTop has good angular resolution of atmospheric shower:
This can be used for training neutrino angular reconstruction IN DATA!

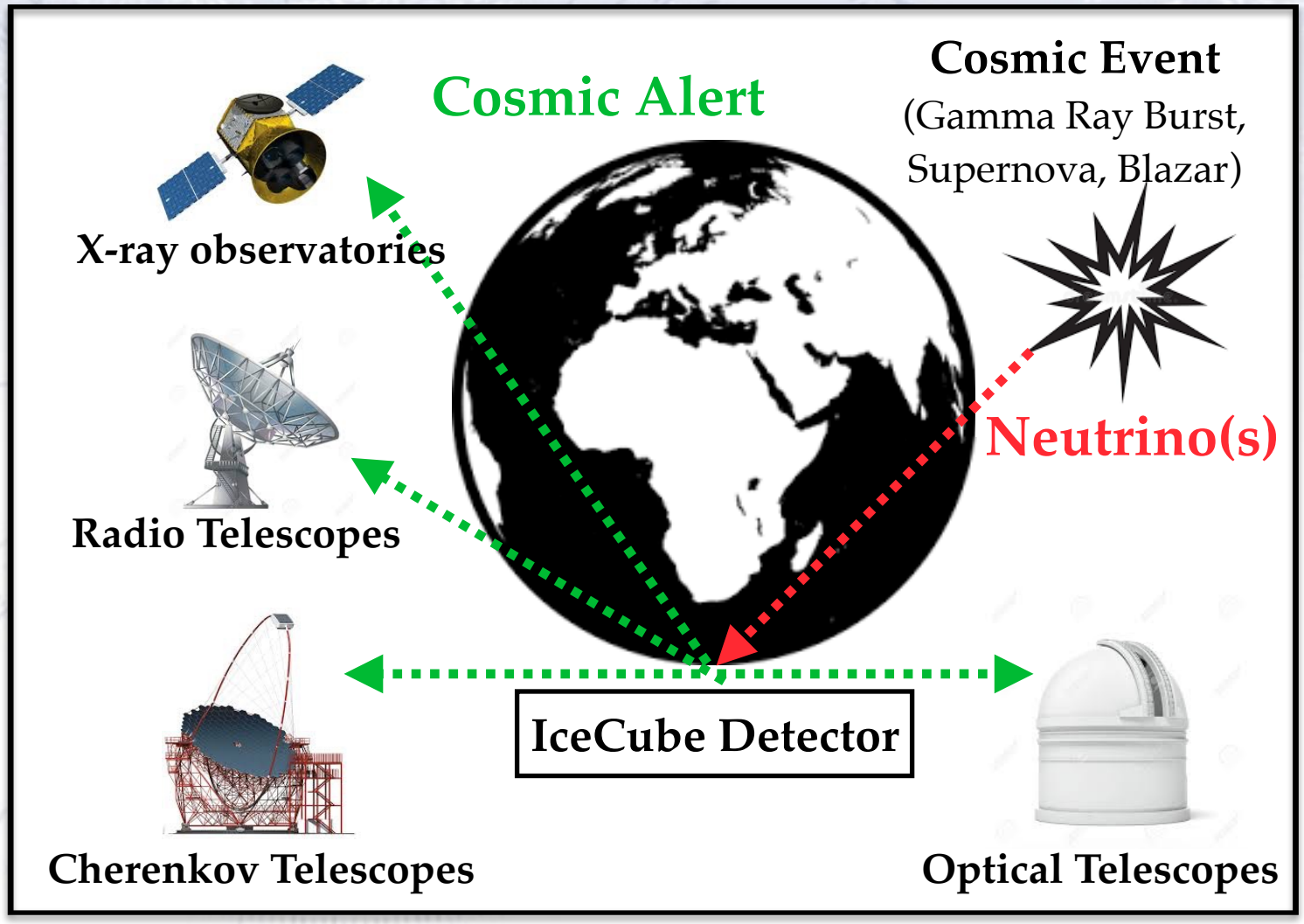
This can also be used as an atmospheric VETO:





Real-time Alerts

Seeing the Universe in ν light



Real-time analysis/alerts

The current IceCube alerts are listed below. They all require high energy (charge > 6000-7000 thus not muons) deposit.

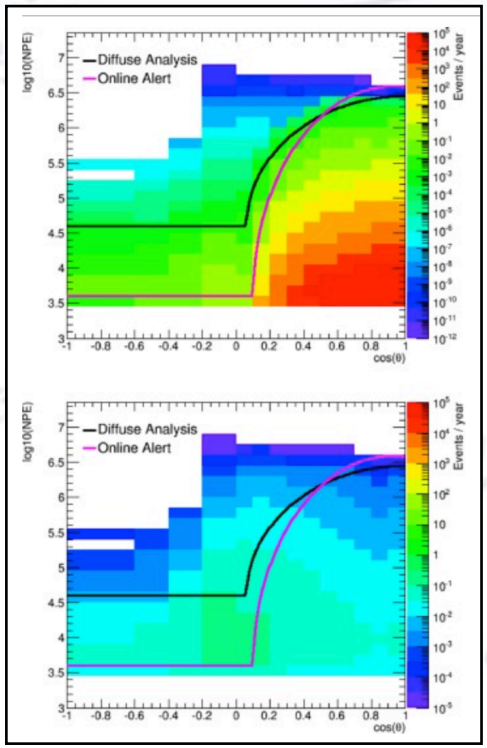
IceCube Realtime Alerts v2

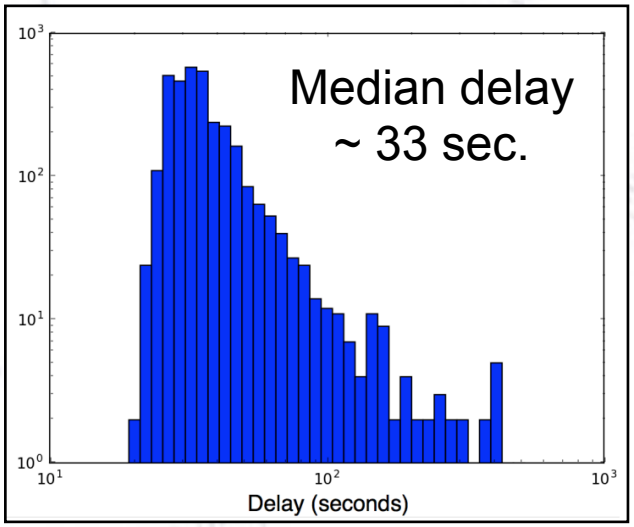
Name/Link	Brief description	Event Topology selected	Selection location/ Latency	Operational status	Triggers sent to	Expected alert rate Gold (Bronze)	Expert POC
Realtime GFU High Energy Tracks	Contribute to Gold and Bronze alerts: High Energy through going tracks from the GFU selection. Targeting detection of sources of astrophysical neutrinos.	All sky, high-energy thru-going tracks	Pole - < 3 min	Operating for V2	Public GCN alerts via AMON	~9/yr (~24/yr)	Thomas Kintscher <thomas.kintscher@desy.de>
EHE Online Alerts Alert Content Info	Contribute to Gold alerts: A very-high energy track search searching for single, high energy tracks of likely astrophysical origin. Targeting detecting of sources of astrophysical neutrinos, GZK neutrinos.	All sky high energy thru-going tracks	Pole - < 3 min	Operating for V2	Public GCN alerts via AMON	~4-6/yr (No Bronze)	Lu Lu <lu.lu@icecube.wisc.edu>
Real Time HESE	High Energy Starting Event search running in Realtime, selecting >6000pe HESE track events for alerts. Targeting detection of sources of astrophysical neutrinos.	All sky, high-energy starting events (tracks)	Pole - < 3 min	Operating for V2	Public GCN alerts via AMON	~1/yr (~2/yr)	Chris Tung <christung616@gmail.com>
Real Time HESE Cascades	High Energy Starting Event Cascade search running in Realtime, selecting >6000pe HESE shower events for alerts. Classification, direction and energy provided by DNN.	All sky, high-energy starting events (showers)	Pole - < 3 min	Operating	Public GCN alerts via AMON	~8/yr	Timothée Grégoire <timothee.gregoire@icecube.wisc.edu>
Optical Follow Up	An online multiplet neutrino search, targeting 2 or more neutrinos from a single location on the sky in 100 seconds. Targeting SN/GRB trasnsients	Northern hemisphere tracks	Event selection: pole; multiplets: North; < 3 min	Operating	PTF, Swift XRT	3/yr (Swift XRT), ~10/yr (PTF)	Anna Franckowiak <anna.franckowiak@desy.de> Alexander Stasik <alexander.stasik@desy.de>
Gamma Follow Up	An online multiplet search targeting significant bursts of neutrinos from known high-energy gamma-ray sources over periods up to 3 weeks. Targeting known flaring AGNs.	All Sky tracks	Event selection: pole; multiplet search: North;	Operating	Magic, Veritas	~2/yr	Manuela Mallamaci <manuela.mallamaci@gmail.com> Elisa Bernardini <elisa.bernardini@desy.de> Konstancja Satalecka <konstancja.satalecka@desy.de>

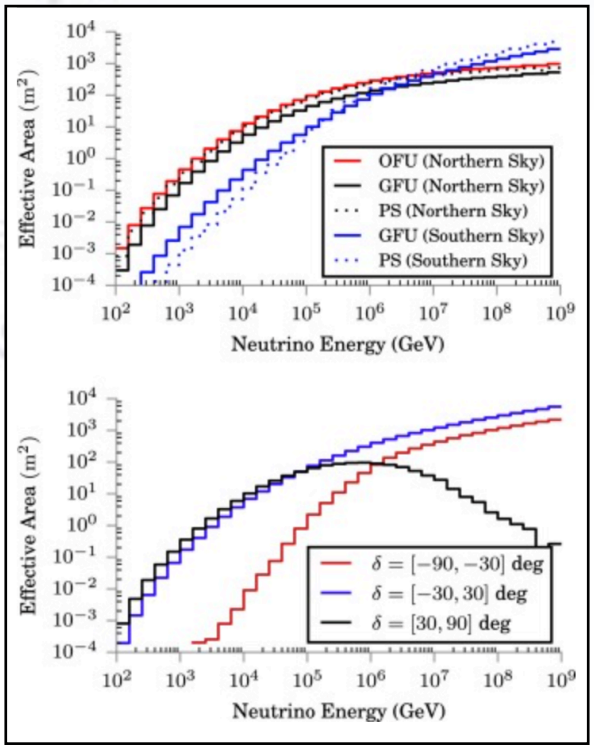
Real-time analysis/alerts

1. **Producing low energy (< 10 TeV) neutrino alert(s)**
   No such alerts currently exists, leaving four orders of magnitude in energy uncovered. Candidate events are Blazars, Tidal disruption events, AGNs, …

2. **Improving high energy (> 10 TeV) neutrino alert(s)**
   The current alerts are (as far as I can read) not as effective as they could be.







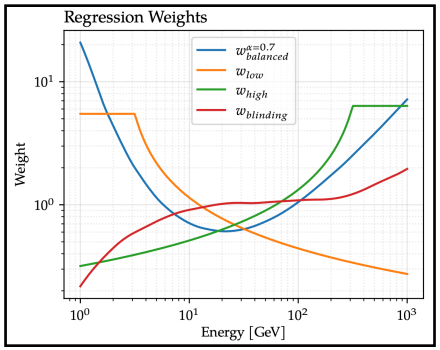
Algorithmic Developments

Algorithmic Developments

1. **GNN autoencoder (for general search)**
   If an AE learned to encode general IceCube events (say noise, muons, and neutrinos from simulation), it could be used to detect other objects.

2. **Hierarchical Graph Pooling**
   Would this work better? Or better only at high energies? Or…?

3. **GNN optimisation (architecture, training sample and reweighting, ensemble combination)**
   Much has been done, but we should push it to the limit :-)
   Perhaps put in 3D-PCA transformation to begin with?



4. **Using "non-signal" DOMs as input**
   Do these contain information to be incorporated, or not?

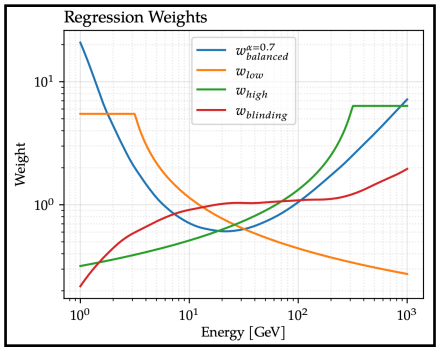
5. **Adversarial training for better performance in data**
   Do common data+MC training (Andreas' idea). But how?

Algorithmic Developments

1. **GNN autoencoder (for general search)**
   If an AE learned to encode general IceCube events (say noise, muons, and neutrinos from simulation), it could be used to detect other objects.

2. **Hierarchical Graph Pooling**
   Would this work better? Or better only at high energies? Or…?

3. **GNN optimisation (architecture, training sample and reweighting, ensemble combination)**
   Much has been done, but we should push it to the limit :-)
   Perhaps put in 3D-PCA transformation to begin with?



4. **Using "non-signal" DOMs as inp**
   Do these contain information to be in not?

Tested (by Rasmus)
No gain!

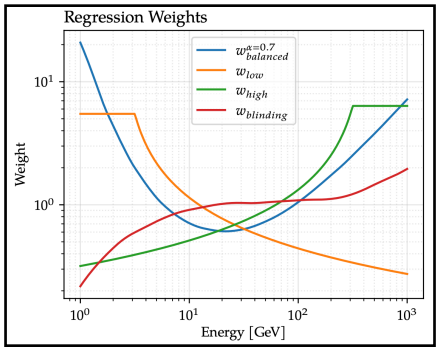
5. **Adversarial training for better performance in data**
   Do common data+MC training (Andreas' idea). But how?

Algorithmic Developments

1. **GNN autoencoder (for general search)**
   If an AE learned to encode general IceCube events (say noise, muons, and neutrinos from simulation), it could be used to detect other objects.

2. **Hierarchical Graph Pooling**
   Would this work better? Or better only at high energies? Or…?

3. **GNN optimisation (architecture, training sample and reweighting, ensemble combination)**
   Much has been done, but we should push it to the limit :-)
   Perhaps put in 3D-PCA transformation to begin with?



4. **Using "non-signal" DOMs as inp**
   Do these contain information to be in hot?

Tested (by Rasmus)
No gain!

5. **Adversarial training for better perfe in data**
   Do common data+MC training (Andreas'). But how?

Mentioned again here!

Scale Energy to ratio

Inspiration from ATLAS:

Inspired by ATLAS, the energy regression is not done to the energy! Why? Because it spans several orders of magnitude (even more so in IceCube).

Scale Energy to ratio

Inspiration from ATLAS:

Inspired by ATLAS, the energy regression is not done to the energy! Why? Because it spans several orders of magnitude (even more so in IceCube).

The solution (in ATLAS) is the regress towards a ratio:

$$\text{Train: } r_{\text{Target (ATLAS)}} = E_{\text{TRUE}} / E_{\text{RAW}} \implies$$
$$\text{Predict: } E_{\text{ML RECO}} = r_{\text{ML RECO}} \cdot E_{\text{RAW}}$$

Scale Energy to ratio

Inspiration from ATLAS:

Inspired by ATLAS, the energy regression is not done to the energy! Why? Because it spans several orders of magnitude (even more so in IceCube).

The solution (in ATLAS) is the regress towards a ratio:

$$\text{Train: } r_{\text{Target (ATLAS)}} = E_{\text{TRUE}} / E_{\text{RAW}} \implies$$
$$\text{Predict: } E_{\text{ML RECO}} = r_{\text{ML RECO}} \cdot E_{\text{RAW}}$$

A similar approach could be done in IceCube, as:

$$\text{Train: } r_{\text{Target (IceCube)}} = E_{\text{TRUE}} / E_{\text{f}(\sum \text{q})} \implies$$
$$\text{Predict: } E_{\text{ML RECO}} = r_{\text{ML RECO}} \cdot E_{\text{f}(\sum \text{q})}$$

From low to high energy…

…efficiently

Aggregation of pulses

I think that the current "standard" is provided by Mirco et al. (2101.11589):
It takes a (potentially large) number of pulses, and output 9 summary features.

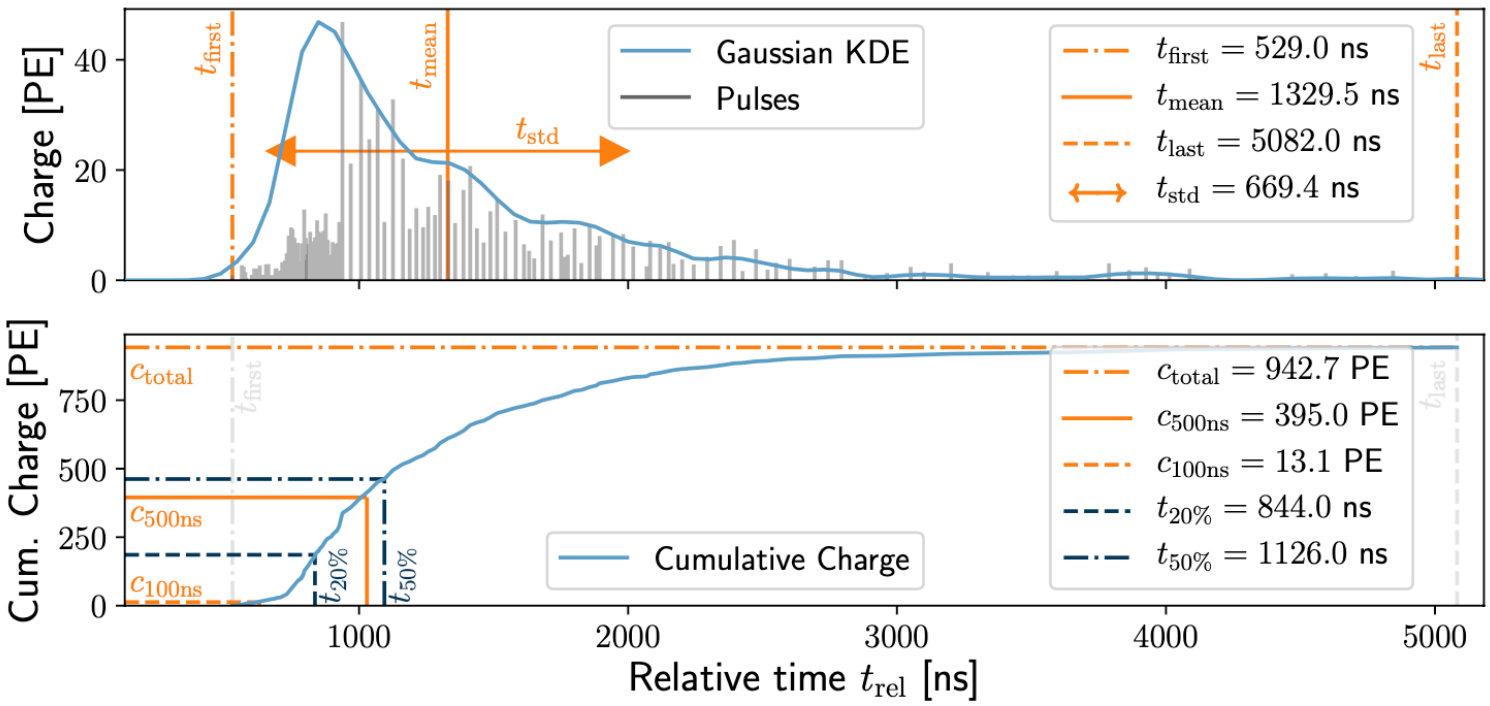
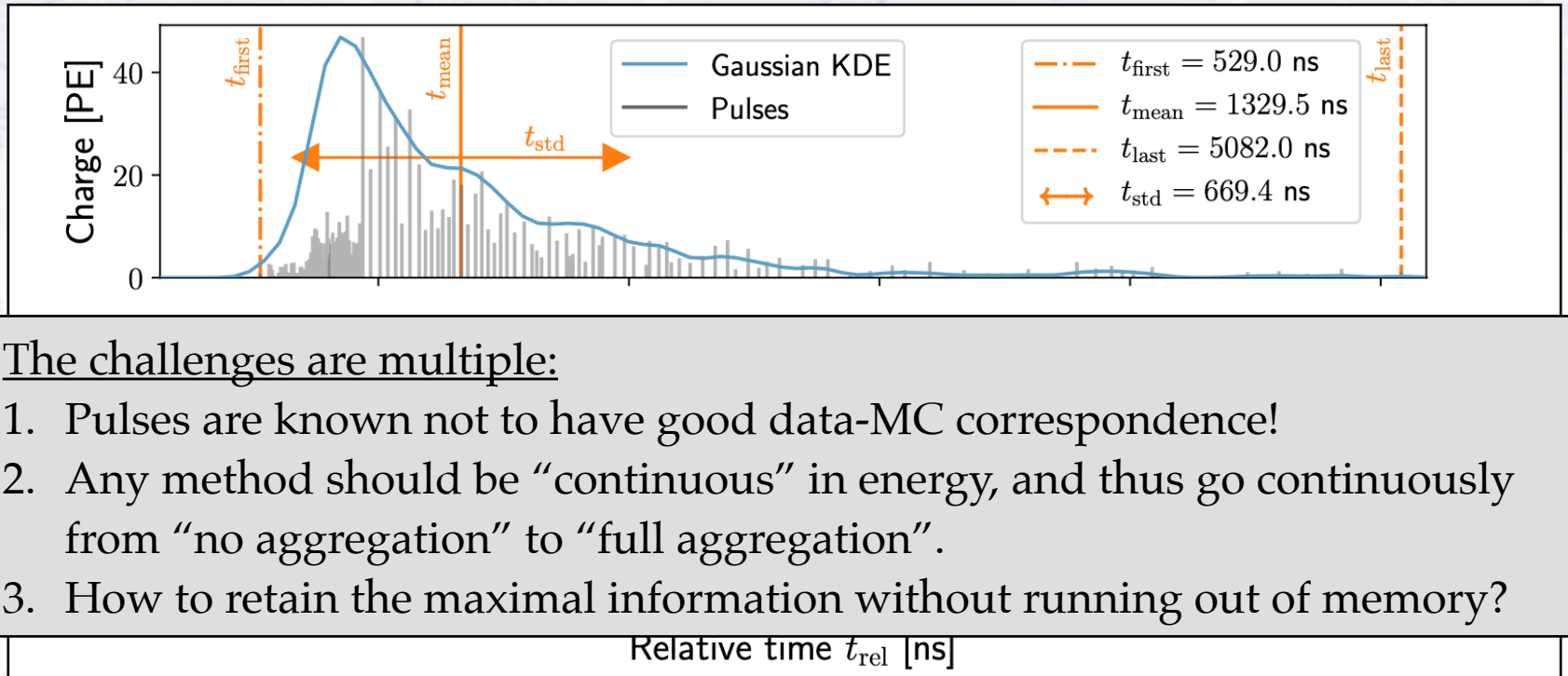


Figure 4: An example pulse series and corresponding input data for a single DOM is shown. The measured pulse series cannot directly be utilized by a CNN due to its varying length. The pulses are therefore reduced to nine input parameters (c_{total}, $c_{500\text{ns}}$, $c_{100\text{ns}}$, t_{first}, t_{last}, $t_{20\%}$, $t_{50\%}$, t_{mean}, t_{std}) which aim to summarize the pulse distribution.

Aggregation of pulses

I think that the current "standard" is provided by Mirco et al. (2101.11589):
It takes a (potentially large) number of pulses, and output 9 summary features.



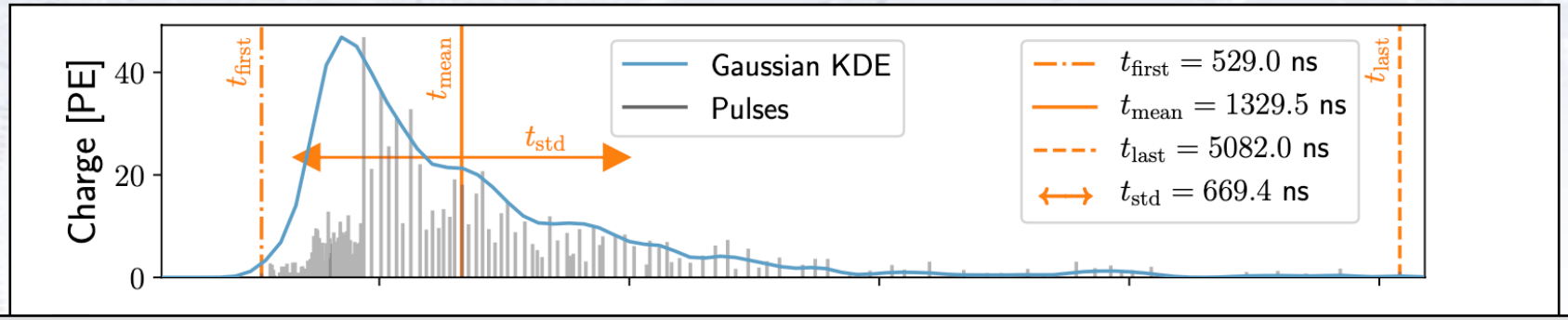
The challenges are multiple:

1. Pulses are known not to have good data-MC correspondence!
2. Any method should be "continuous" in energy, and thus go continuously from "no aggregation" to "full aggregation".
3. How to retain the maximal information without running out of memory?

Figure 4: An example pulse series and corresponding input data for a single DOM is shown. The measured pulse series cannot directly be utilized by a CNN due to its varying length. The pulses are therefore reduced to nine input parameters (c_{total}, $c_{500\text{ns}}$, $c_{100\text{ns}}$, t_{first}, t_{last}, $t_{20\%}$, $t_{50\%}$, t_{mean}, t_{std}) which aim to summarize the pulse distribution.

Aggregation of pulses

I think that the current "standard" is provided by Mirco et al. (2101.11589):
It takes a (potentially large) number of pulses, and output 9 summary features.



<u>The challenges are multiple:</u>

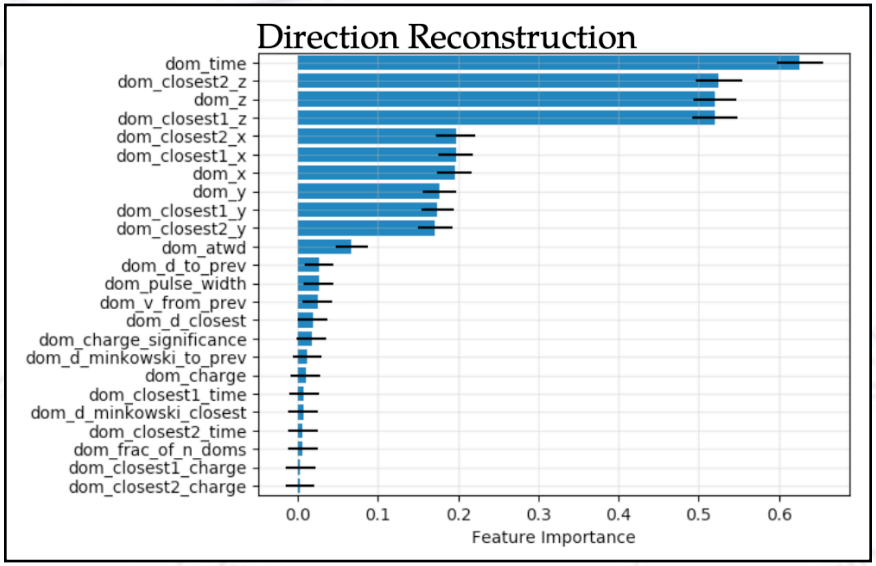
1. Pulses are known not to have good data-MC correspondence!
   Pulse calibration can be done on stopped muons and non-signal neutrinos.
2. Any method should be "continuous" in energy, and thus go continuously from "no aggregation" to "full aggregation".
   Suggestion: Retain the first 2-3 pulses and the Mirco summary data. This enlarges single puls DOM info from 6 (xyztqQE) to 16-18 (xyzt1q1t2q2 3xt 5xQ QE) variables.
3. How to retain the maximal information without running out of memory?
   Only testing will be able to tell us that - how many ways can we envision?

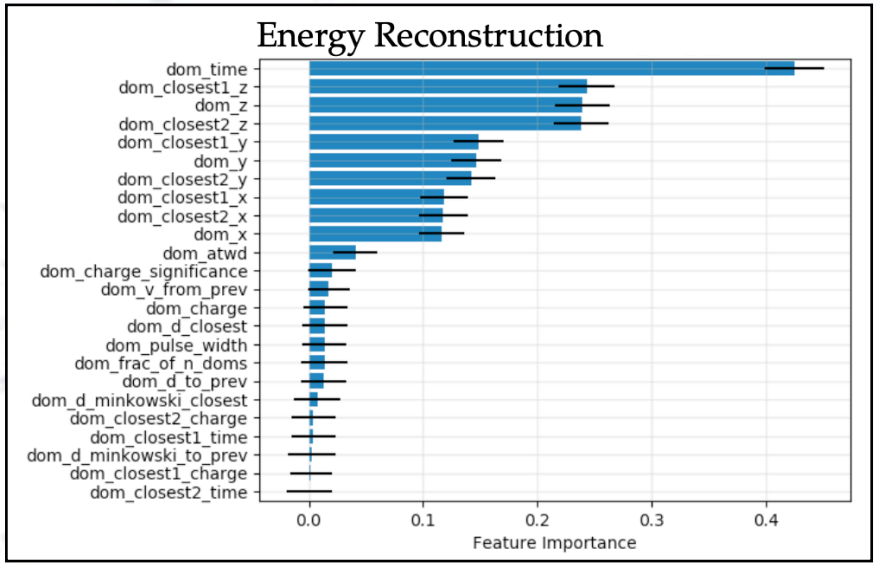
Explaining/Visualising the GNN answers

Explaining / Visualising GNN output

1. **Overall feature ranking for each task**

   Which are the important features for energy regression and directional regression? Are they the same or do some cases stand out?





Additional Ideas

A non-exhaustive list:

- Pre-train models in data… and them MC:
  Can one obtain better training (or data-MC agreement) by unsupervised training in data first?(Andreas et al.)

- Segmentation with ML:
  Can one recognise cosmic muons on top to ordinary (good) neutrino events?

- Put muons on top of event in a small (10%) fraction of neutrino events.
  This can be done directly in the pulse maps (cleaning the muon event, and putting it in at a random time). In this way the neutrino is (perhaps) saved!

- …more time… more workshops… more ideas :-)

Thoughts on where to begin

Thoughts on where to begin

I think the lowest hanging fruits most ripe for GNN reconstruction are:

1. **Producing the "Loose GNN neutrino sample",**
   Making such a sample would be a large contribution to IceCube at a whole.

2. **High energy neutrino** ($\nu_e + \nu_\mu + \nu_\tau$) classification and reconstruction.
   We need to determine if/how much we can improve on angular resolution.
   If it is significantly (> 10%) better, then I think we have an obvious paper.
   We have "everything in place" from low energy and Kaggle.

3. **Think about Low Energy neutrino alerts.**
   Leveraging the GNN speed, this suddenly becomes possible - a first!

4. **Moon pointing analysis,** possibly as a function of energy.
   This will - IMHO - be the **only way** to convince ourselves and others, that the GNN angular precision is (also) very good in real data (if it is?).

5. **Stopped Muon and non-Osc neutrino Data-MC comparison.**
   There are many things that such an analysis allows us to do, e.g. transform pulse distributions of MC to match those of data.

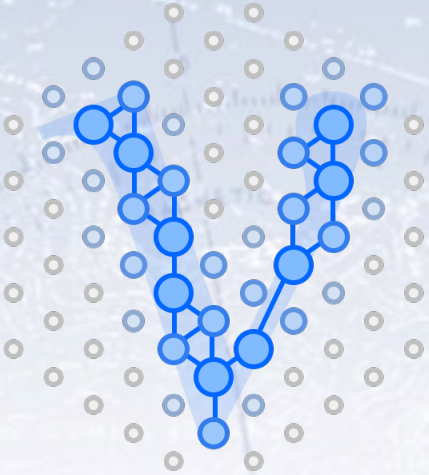
GraphNet Workshop IV

Germany/Denmark/???

Xth to Yth of Z 2023?

Bonus slides


Troels C. Petersen
SCIENCE
Formidlingspris 2015



GraphNeT

Graph Neural Networks for Neutrino Telescope Event Reconstruction

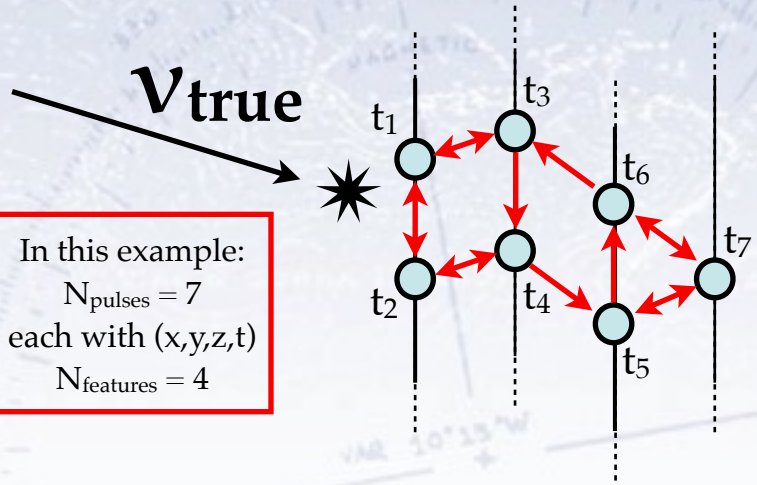
GraphNet is our attempt at putting GNN models for IceCube (and others) using the "DynEdge" architecture build in PyTorch Geometric into an easily available software package.

https://github.com/icecube/graphnet/

In the following, I will try to convince you, that GNNs "match" the IceCube data really well, and perhaps whet your appetite for using it.

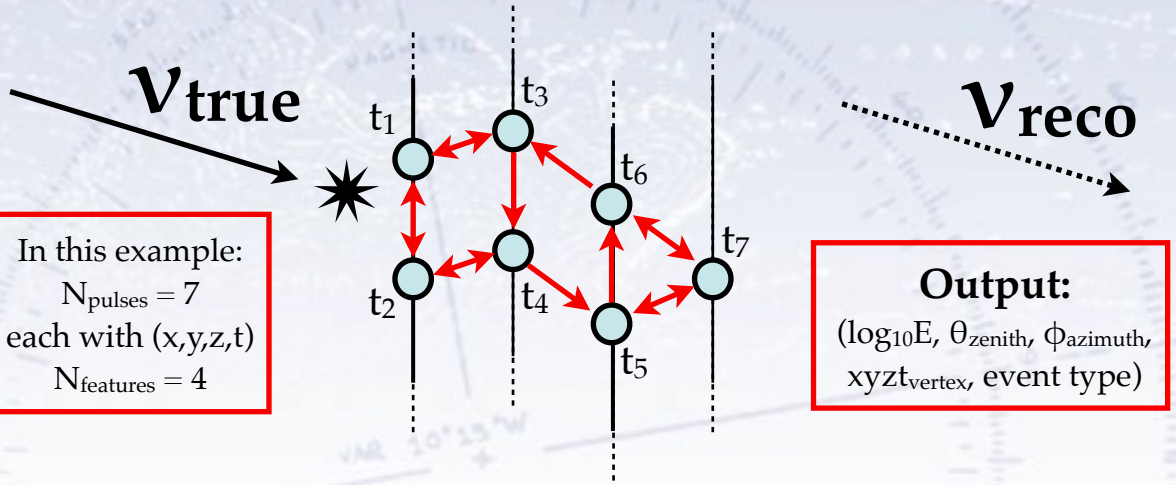
Our results can be found in an IceCube paper, submitted to JINST in September.

Details of GNN reconstruction



In this example:
$N_{pulses} = 7$
each with (x,y,z,t)
$N_{features} = 4$

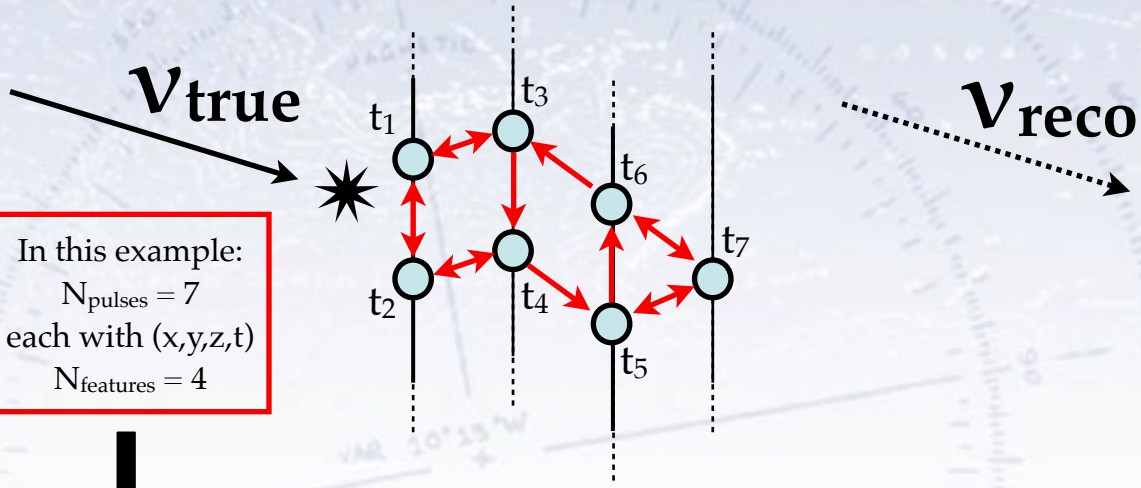
Details of GNN reconstruction



In this example:
$N_{pulses} = 7$
each with (x,y,z,t)
$N_{features} = 4$

Output:
($\log_{10}E$, θ_{zenith}, $\phi_{azimuth}$, $xyzt_{vertex}$, event type)

Details of GNN reconstruction



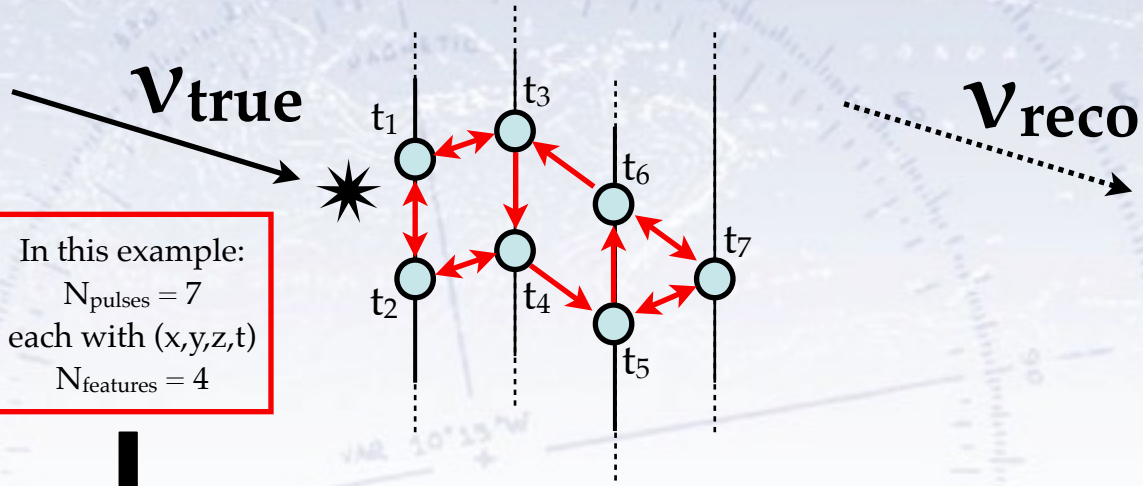
In this example:
$N_{pulses} = 7$
each with (x,y,z,t)
$N_{features} = 4$

$$\vec{v}_1 = [x_1\ y_1\ z_1\ t_1]$$
$$\vec{v}_2 = [x_2\ y_2\ z_2\ t_2]$$
$$\vdots$$
$$\vec{v}_7 = [x_7\ y_7\ z_7\ t_7]$$

Input:
$N = N_{pulses} \times N_{features}$

The input features of a node are combined with that of N (=2) nearby nodes

Details of GNN reconstruction



In this example:
$N_{pulses} = 7$
each with (x,y,z,t)
$N_{features} = 4$

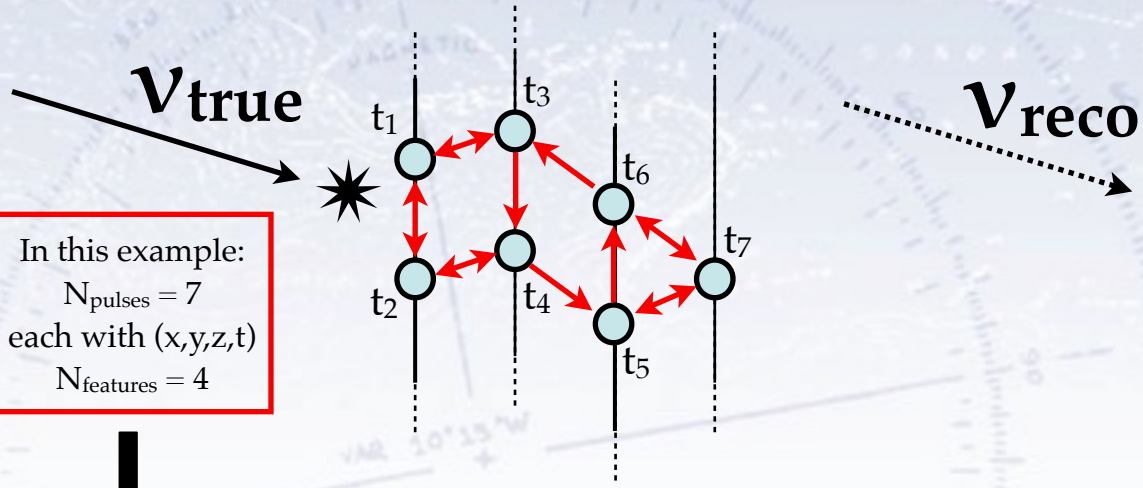
$$\vec{v}_1 = [x_1\ y_1\ z_1\ t_1] \xrightarrow{EC(\vec{v}_1,\vec{v}_2,\vec{v}_3)} [g_{11} \dots g_{1N_1}]$$
$$\vec{v}_2 = [x_2\ y_2\ z_2\ t_2] \qquad\qquad [g_{21} \dots g_{2N_1}]$$
$$\vdots \xrightarrow{EC(\vec{v}_4,\vec{v}_5,\vec{v}_6)} \vdots$$
$$\vec{v}_7 = [x_7\ y_7\ z_7\ t_7] \qquad\qquad [g_{71} \dots g_{7N_1}]$$

Input:
$N = N_{pulses} \times N_{features}$

Convolution(s):
$N = N_{pulses} \times N_1$

The input features of a node are combined with that of N (=2) nearby nodes through an NN (MLP0) function, yielding an (abstract) vector for each node. This can be repeated (not shown).

Details of GNN reconstruction



In this example:
$N_{pulses} = 7$
each with (x,y,z,t)
$N_{features} = 4$

$N_{all} = N_{features} + N_1$

$$\vec{v}_1 = [x_1\ y_1\ z_1\ t_1] \xrightarrow{EC(\vec{v}_1,\vec{v}_2,\vec{v}_3)} [g_{11} \dots g_{1N_1}] \quad [x_1\ y_1\ z_1\ t_1\ g_{11} \dots g_{1N_1}]$$
$$\vec{v}_2 = [x_2\ y_2\ z_2\ t_2] \quad [g_{21} \dots g_{2N_1}] \quad [x_2\ y_2\ z_2\ t_2\ g_{21} \dots g_{2N_1}]$$
$$\vdots \xrightarrow{EC(\vec{v}_4,\vec{v}_5,\vec{v}_6)} \vdots \quad \vdots$$
$$\vec{v}_7 = [x_7\ y_7\ z_7\ t_7] \quad [g_{71} \dots g_{7N_1}] \quad [x_7\ y_7\ z_7\ t_7\ g_{71} \dots g_{7N_1}]$$

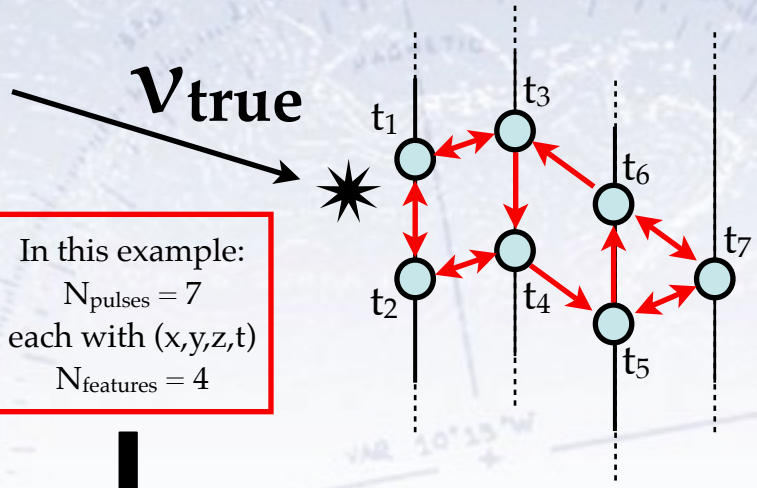
Input:
$N = N_{pulses} \times N_{features}$

Convolution(s):
$N = N_{pulses} \times N_1$

Concatenation:
$N = N_{pulses} \times N_{all}$

The input features of a node are combined with that of N (=2) nearby nodes through an NN (MLP0) function, yielding an (abstract) vector for each node. This can be repeated (not shown). All the features are then combined (concatenated) into long vectors,

Details of GNN reconstruction



In this example:
$N_{pulses} = 7$
each with (x,y,z,t)
$N_{features} = 4$

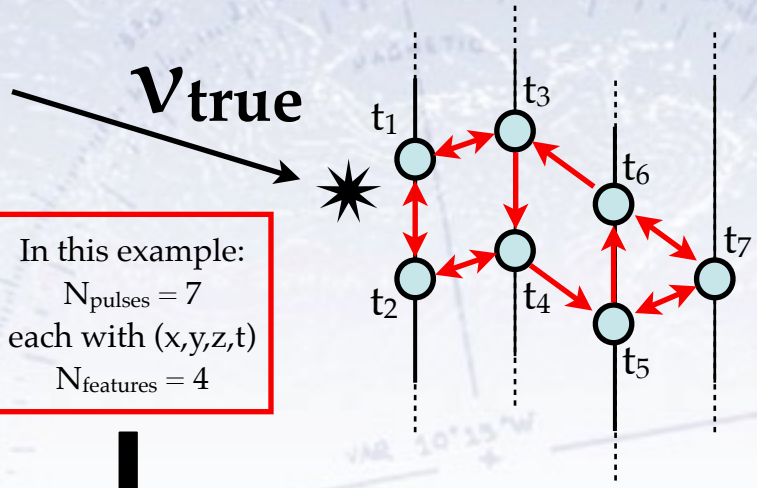
$N_{all} = N_{features} + N_1$

$$\vec{v}_1 = [x_1\ y_1\ z_1\ t_1] \xrightarrow{EC(\vec{v}_1, \vec{v}_2, \vec{v}_3)} [g_{11} \dots g_{1N_1}] \quad [x_1\ y_1\ z_1\ t_1\ g_{11} \dots g_{1N_1}] \rightarrow [h_{11} \dots h_{1N_{all}}]$$
$$\vec{v}_2 = [x_2\ y_2\ z_2\ t_2] \qquad [g_{21} \dots g_{2N_1}] \quad [x_2\ y_2\ z_2\ t_2\ g_{21} \dots g_{2N_1}] \rightarrow [h_{21} \dots h_{2N_{all}}]$$
$$\vdots \xrightarrow{EC(\vec{v}_4, \vec{v}_5, \vec{v}_6)} \vdots \qquad \vdots \qquad \vdots$$
$$\vec{v}_7 = [x_7\ y_7\ z_7\ t_7] \qquad [g_{71} \dots g_{7N_1}] \quad [x_7\ y_7\ z_7\ t_7\ g_{71} \dots g_{7N_1}] \rightarrow [h_{71} \dots h_{7N_{all}}]$$

Input:	**Convolution(s):**	**Concatenation:**	**MLP1:**
$N = N_{pulses} \times N_{features}$	$N = N_{pulses} \times N_1$	$N = N_{pulses} \times N_{all}$	$N_{all} \rightarrow N_{many} \rightarrow N_{all}$

The input features of a node are combined with that of N (=2) nearby nodes through an NN (MLP0) function, yielding an (abstract) vector for each node. This can be repeated (not shown). All the features are then combined (concatenated) into long vectors, which are again put through an NN (MLP1) function with a large hidden layer.

Details of GNN reconstruction



In this example:
$N_{pulses} = 7$
each with (x,y,z,t)
$N_{features} = 4$

$$[\Sigma_1 \dots \Sigma_{N_{all}}]$$
$$[\mu_1 \dots \mu_{N_{all}}]$$
$$[\wedge_1 \dots \wedge_{N_{all}}]$$
$$[\vee_1 \dots \vee_{N_{all}}]$$

$\text{sum}(\Sigma)$, $\text{mean}(\mu)$ $\quad$ $\max(\wedge)$, $\min(\vee)$

$N_{all} = N_{features} + N_1$

$$\vec{v}_1 = [x_1\ y_1\ z_1\ t_1] \xrightarrow{EC(\vec{v}_1,\vec{v}_2,\vec{v}_3)} [g_{11} \dots g_{1N_1}]$$
$$\vec{v}_2 = [x_2\ y_2\ z_2\ t_2] \qquad\quad [g_{21} \dots g_{2N_1}]$$
$$\vdots \xrightarrow{EC(\vec{v}_4,\vec{v}_5,\vec{v}_6)} \vdots$$
$$\vec{v}_7 = [x_7\ y_7\ z_7\ t_7] \qquad\quad [g_{71} \dots g_{7N_1}]$$

$$[x_1\ y_1\ z_1\ t_1\ g_{11} \dots g_{1N_1}] \rightarrow [h_{11} \dots h_{1N_{all}}]$$
$$[x_2\ y_2\ z_2\ t_2\ g_{21} \dots g_{2N_1}] \rightarrow [h_{21} \dots h_{2N_{all}}]$$
$$\vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots$$
$$[x_7\ y_7\ z_7\ t_7\ g_{71} \dots g_{7N_1}] \rightarrow [h_{71} \dots h_{7N_{all}}]$$

Input: $N = N_{pulses} \times N_{features}$

Convolution(s): $N = N_{pulses} \times N_1$

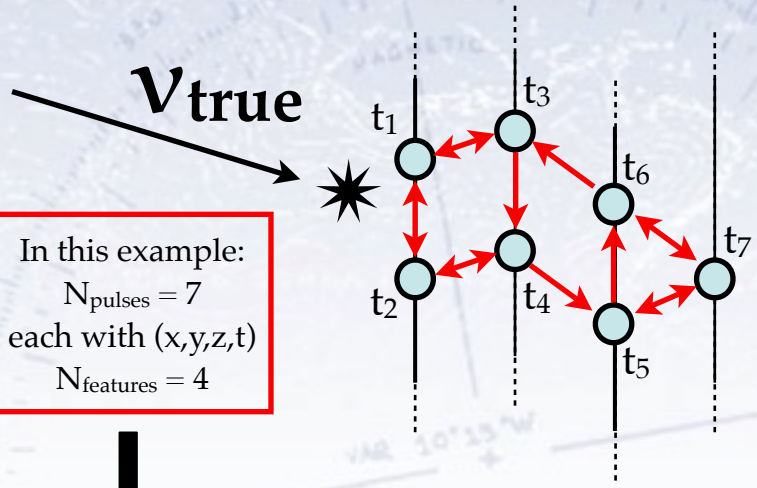
Concatenation: $N = N_{pulses} \times N_{all}$

MLP1: $N_{all} \rightarrow N_{many} \rightarrow N_{all}$

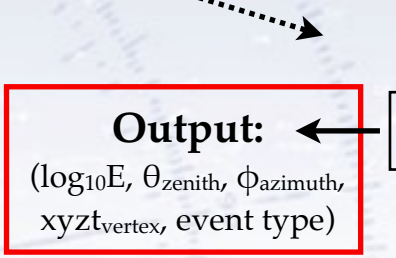
Aggregation: $N = 4 \times N_{all}$

The input features of a node are combined with that of N (=2) nearby nodes through an NN (MLP0) function, yielding an (abstract) vector for each node. This can be repeated (not shown). All the features are then combined (concatenated) into long vectors, which are again put through an NN (MLP1) function with a large hidden layer. The outputs are aggregated in four ways: Min, Max, Sum & Mean, breaking the variation with number of nodes.

Details of GNN reconstruction



In this example:
$N_{pulses} = 7$
each with (x,y,z,t)
$N_{features} = 4$



Output:
($\log_{10}$E, θ_{zenith}, $\phi_{azimuth}$, $xyzt_{vertex}$, event type)

MLP2

$[\Sigma_1 \dots \Sigma_{N_{all}}]$
$[\mu_1 \dots \mu_{N_{all}}]$
$[\wedge_1 \dots \wedge_{N_{all}}]$
$[\vee_1 \dots \vee_{N_{all}}]$

sum(Σ), mean(μ), max($\wedge$), min($\vee$)

$N_{all} = N_{features} + N_1$

$\vec{v}_1 = [x_1\ y_1\ z_1\ t_1]$ $\xrightarrow{EC(\vec{v}_1, \vec{v}_2, \vec{v}_3)}$ $[g_{11} \dots g_{1N_1}]$
$\vec{v}_2 = [x_2\ y_2\ z_2\ t_2]$ $[g_{21} \dots g_{2N_1}]$
$\vec{v}_7 = [x_7\ y_7\ z_7\ t_7]$ $\xrightarrow{EC(\vec{v}_4, \vec{v}_5, \vec{v}_6)}$ $[g_{71} \dots g_{7N_1}]$

$[x_1\ y_1\ z_1\ t_1\ g_{11} \dots g_{1N_1}]$
$[x_2\ y_2\ z_2\ t_2\ g_{21} \dots g_{2N_1}]$
$[x_7\ y_7\ z_7\ t_7\ g_{71} \dots g_{7N_1}]$

$[h_{11} \dots h_{1N_{all}}]$
$[h_{21} \dots h_{2N_{all}}]$
$[h_{71} \dots h_{7N_{all}}]$

Input:
$N = N_{pulses} \times N_{features}$

Convolution(s):
$N = N_{pulses} \times N_1$

Concatenation:
$N = N_{pulses} \times N_{all}$

MLP1:
$N_{all} \rightarrow N_{many} \rightarrow N_{all}$

Aggregation:
$N = 4 \times N_{all}$

The input features of a node are combined with that of N (=2) nearby nodes through an NN (MLP0) function, yielding an (abstract) vector for each node. This can be repeated (not shown). All the features from all the convolutions are then combined (concatenated) into long vectors, which are again put through an NN (MLP1) function with a large hidden layer. The outputs are aggregated in four ways: Min, Max, Sum & Mean, breaking the variation with number of nodes. These are then fed into a final NN (MLP2), which outputs the estimated type(s) and parameters of the event.

Further specifics of DynEdge

In DynEdge, there are several “enlargements” compared to the previous illustration of the GNN architecture. These are essentially:

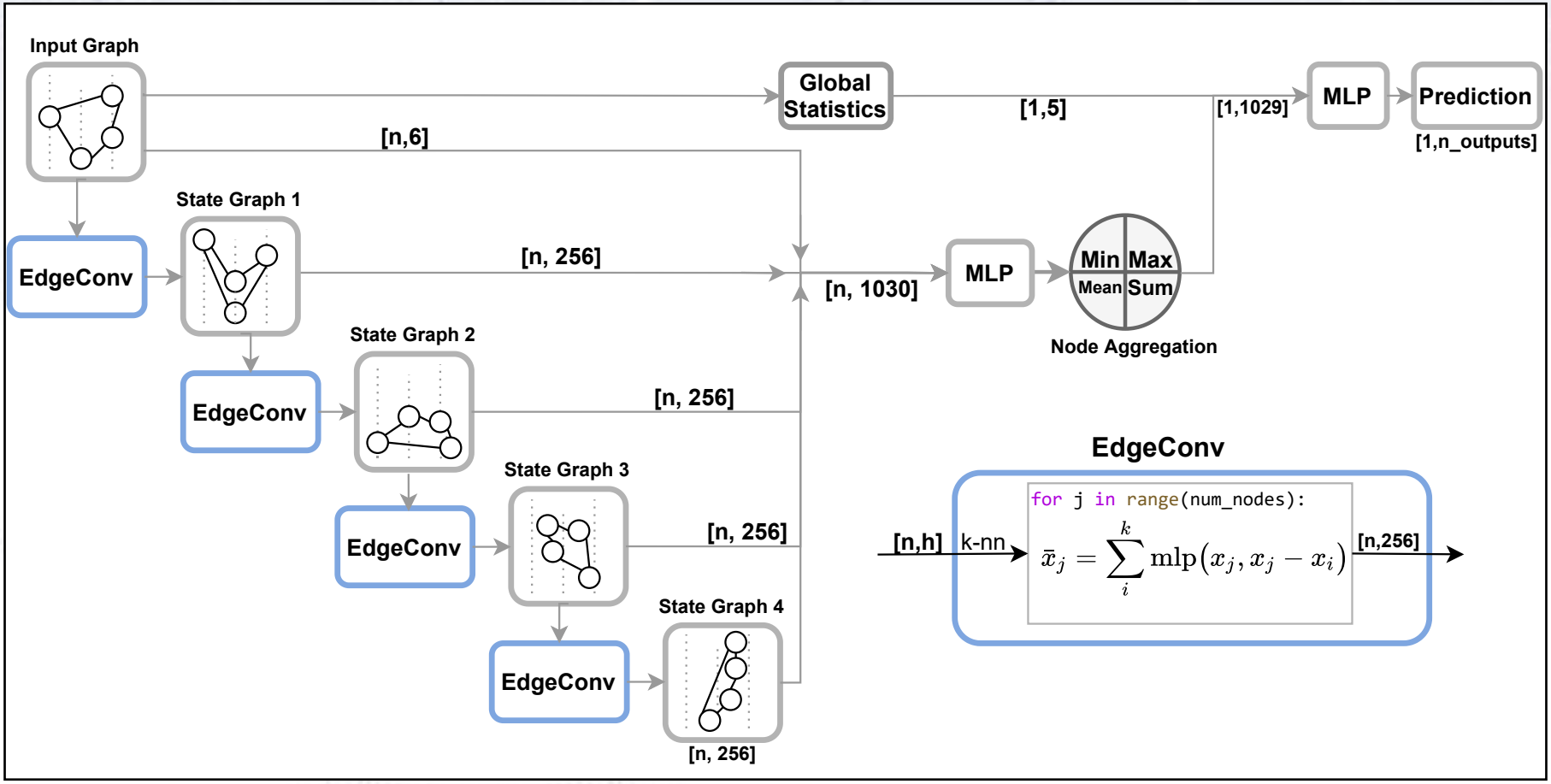
- We use 6 input features: x, y, z, t, charge, and Quantum Efficiency.
- We convolute each node with the nearest 8 nodes (not two).
- We do 4 (not 1) convolutions, each with 192 inputs and outputs.
- The concatenation is of all convolution layers and the original input.
- In the results to be shown, we have trained separate GNNs for each output.

The repeated convolutions allows all signal parts to be connected.
The EdgeConv convolution operator ensures permutation invariance.

The number of model parameters is about 750.000 for the angular regressions, while the energy only requires 150.000. In principle one can go down to 70.000 parameters, but there is no reason for this - it is already a “small” ML model.

GraphNet

The GNN model is outlined more simply below, which is also the (current) figure for the GNN paper in process.



Efficiencies of GNN (Not-even-preliminary)

Neutrino efficiencies on MC

Reproducing Etienne's fantastic rate plot, we've gotten to this (no weights)…