Machine Learning workflows in KM3NeT

Stefan Reck

IWAPP workshop

2021-03-10

KM3NeT

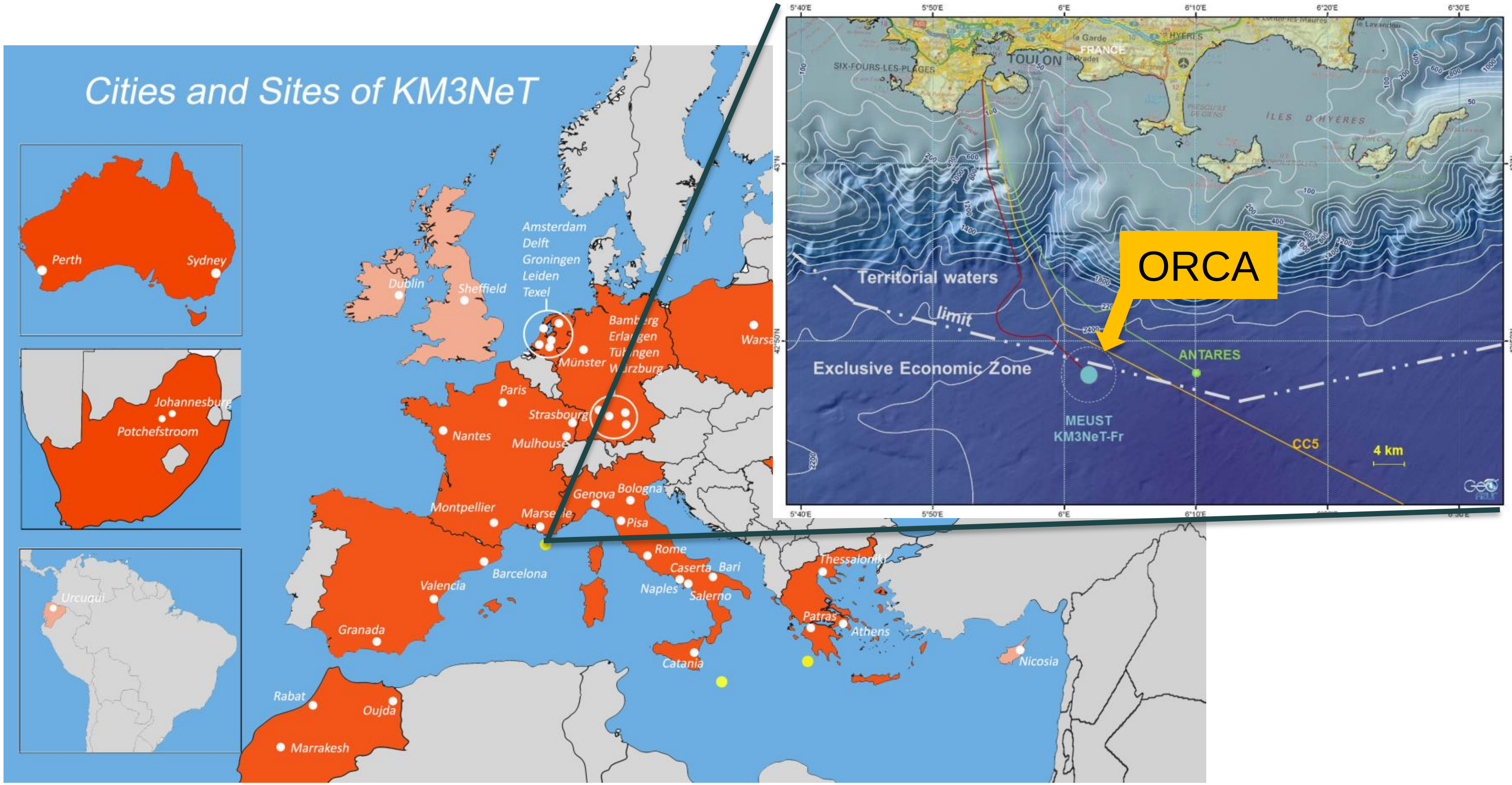
ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

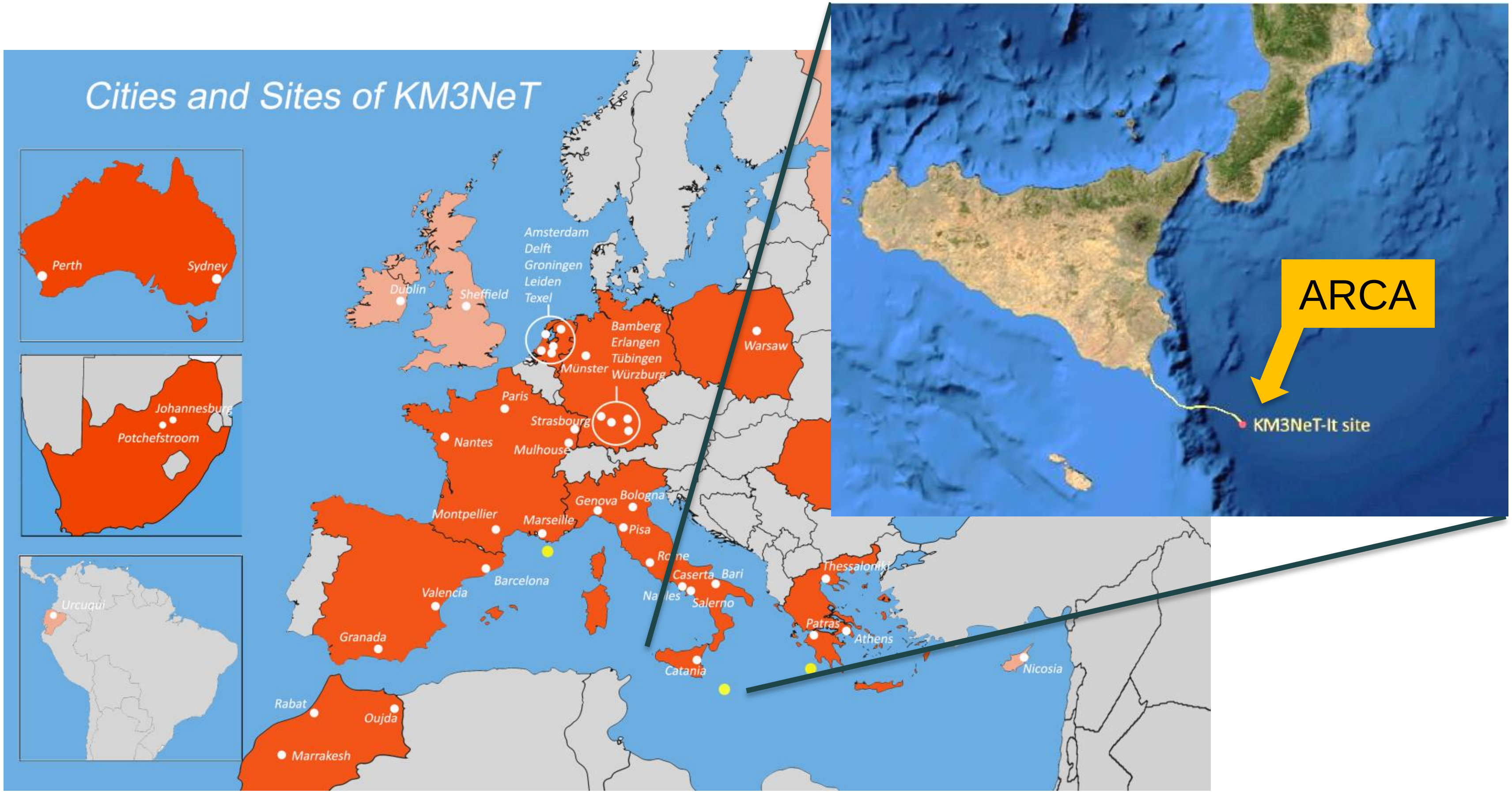
ORCA and ARCA

same technology – different sites

KM3NeT: neutrino detector network

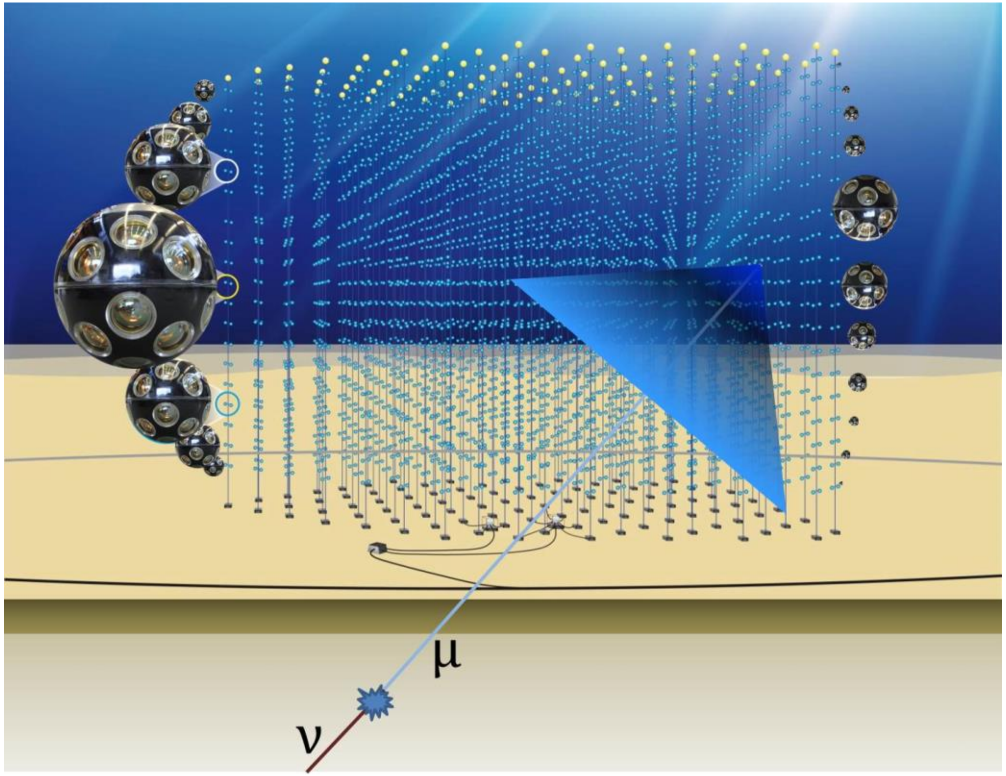


KM3NeT: neutrino detector network



KM3NeT: neutrino detector network

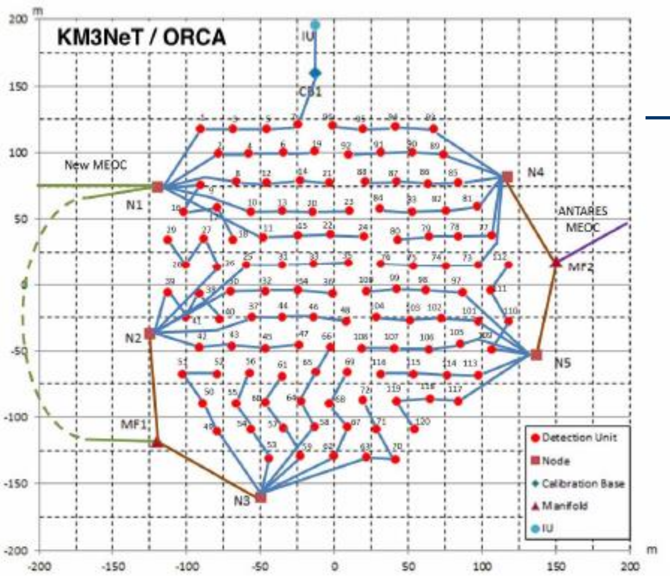
→ detection principle: measure Cherenkov radiation of charged particles



DOM	31 PMTs
String	18 DOMs
Total (planned)	3 blocks of 115 strings each
Currently	6 strings (ORCA) 1 string (ARCA)

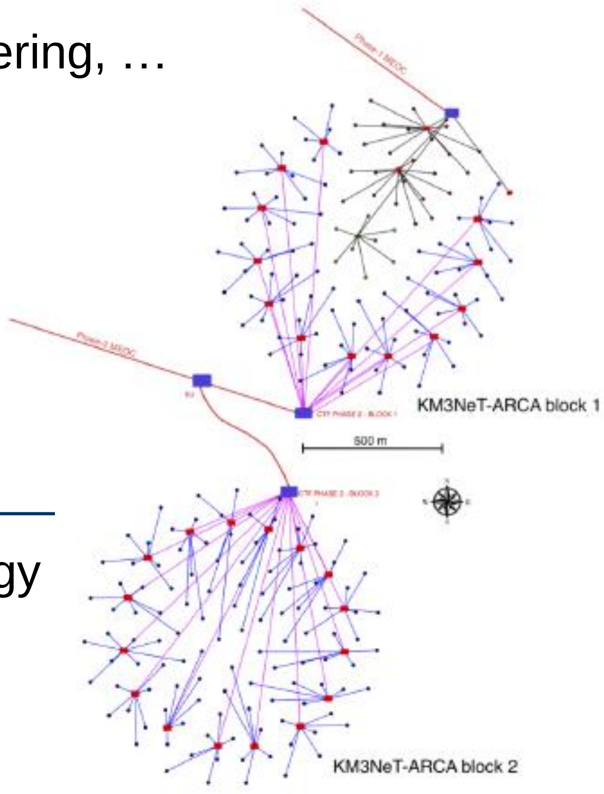
KM3NeT: neutrino detector network

https://arxiv.org/abs/1601.07459



KM3NeT ORCA

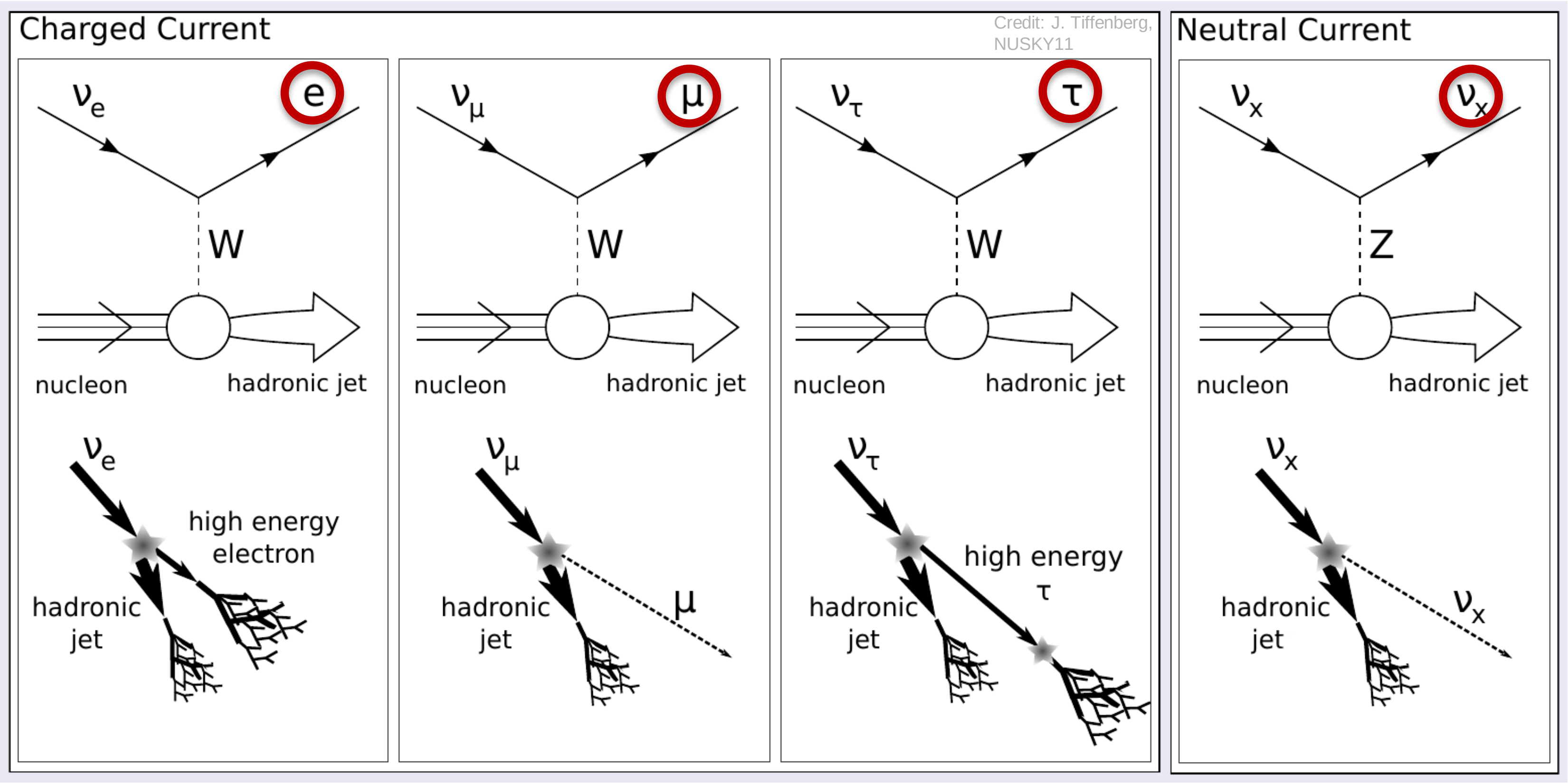
- for neutrinos with GeV energy
- neutrino oscillations, mass ordering, …
- height: 150m
- line spacing: about 20 - 23m



KM3NeT ARCA

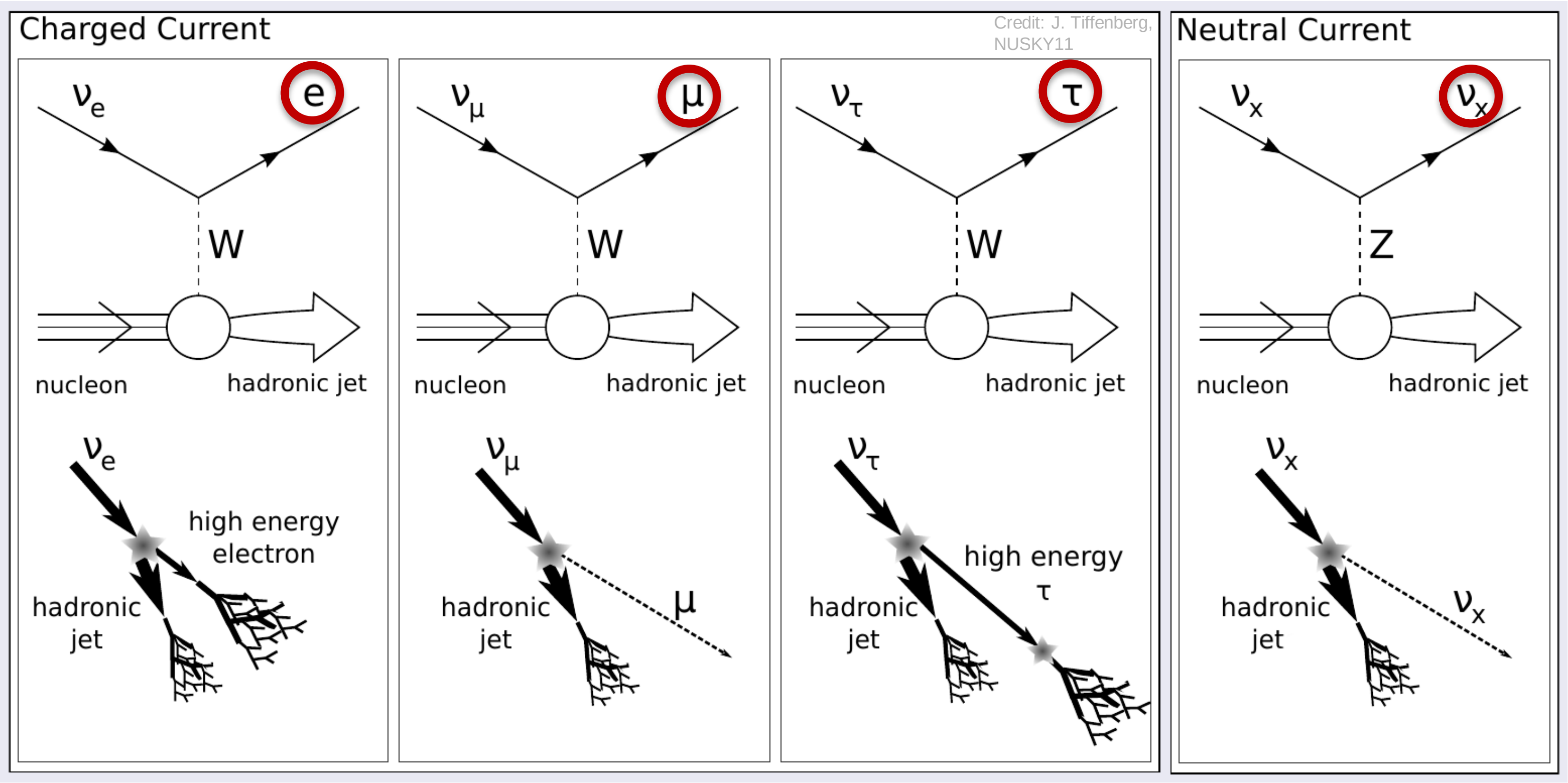
- for neutrinos with TeV energy
- cosmic neutrinos, …
- height: 600m
- line spacing: about 90m

Neutrino interactions



$\nu_e - \mathrm{CC}$ $\nu_\mu - \mathrm{CC}$ $\nu_\tau - \mathrm{CC}$ $\nu - \mathrm{NC}$

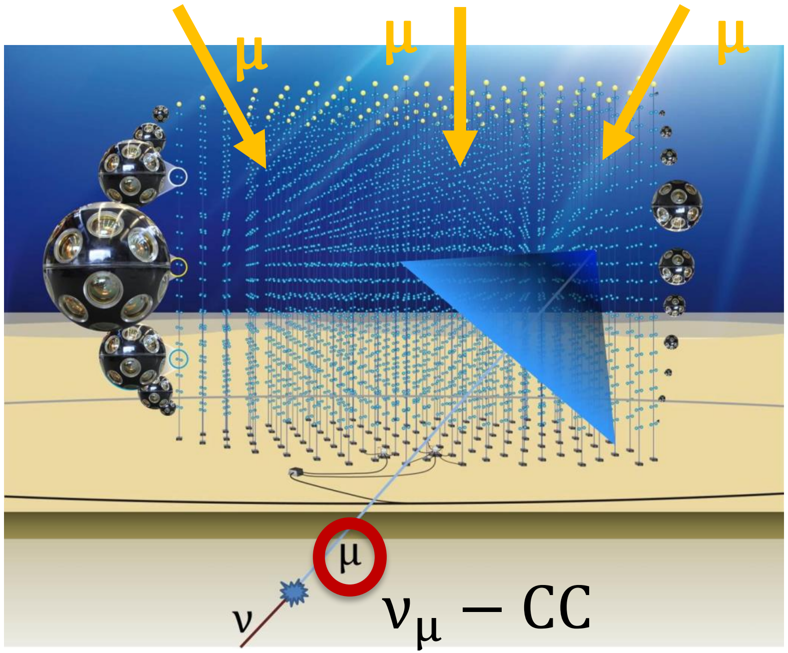
Detector signatures



shower-like	track-like	shower-like (83%)	shower-like
		track-like (17%)	

KM3NeT backgrounds

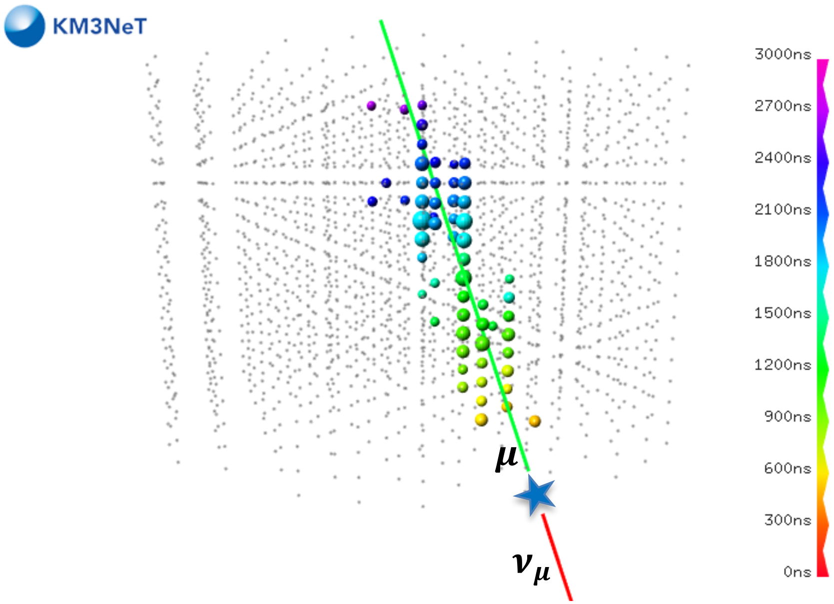
- Two types of background producing photons in the deep sea:
  1. **Atmospheric muons** passing the detector from above
  2. **Random noise**, by K-40 beta decays and bioluminescent organisms



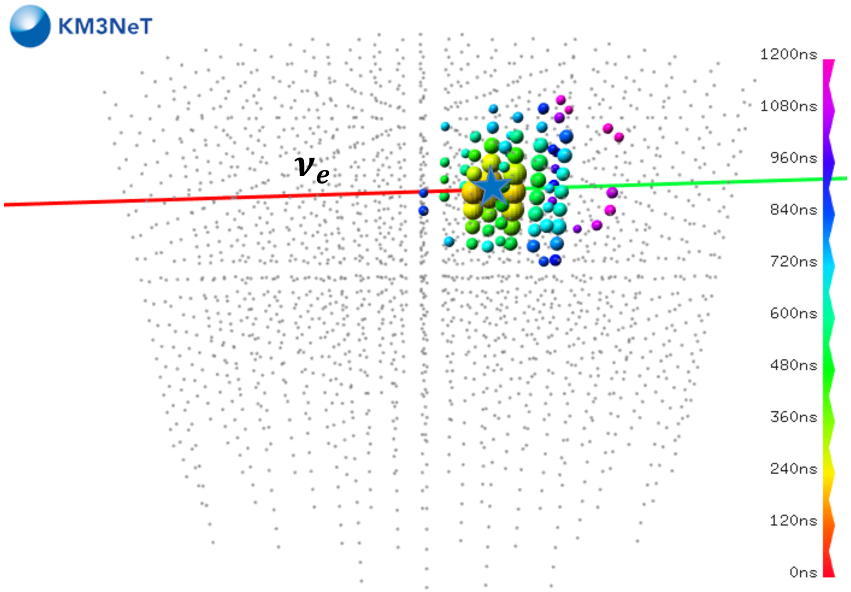
Event topologies

https://arxiv.org/abs/1601.07459

Up-going ν_μ – CC **track**-like event

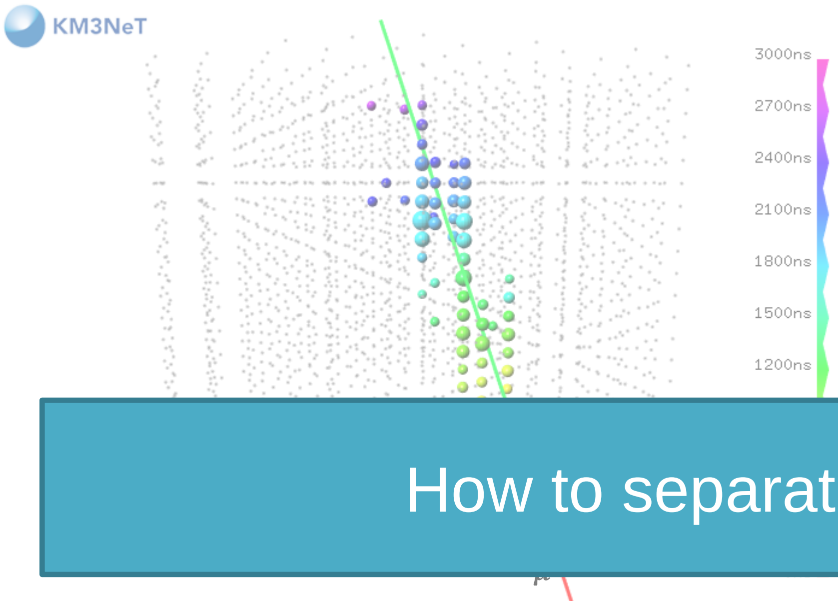


ν_e – CC **shower**-like event

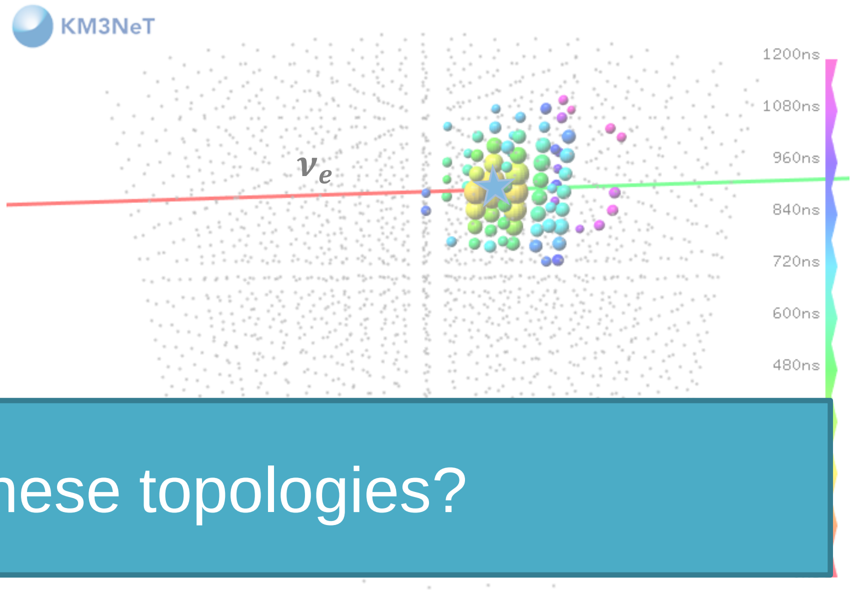


Event topologies

Up-going ν_μ – CC **track**-like event



ν_e – CC **shower**-like event



How to separate these topologies?

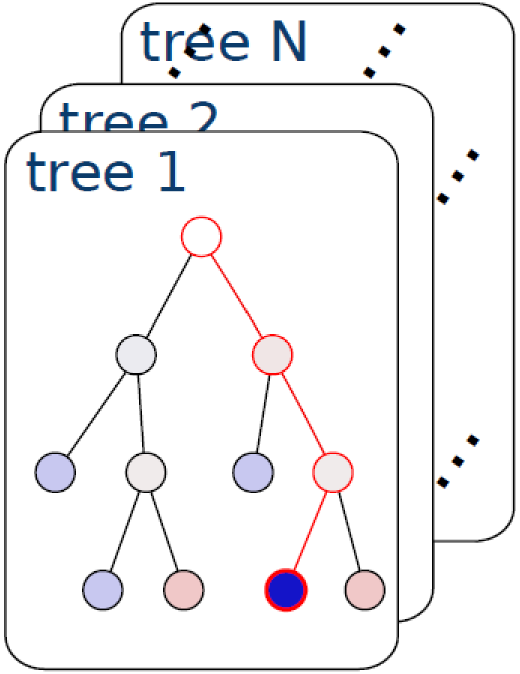
Random Decision Forests

KM3NeT

ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Random decision forest



- ensemble of decision trees
- input: "hand-crafted" features

  → these need to be manually designed

  in total: ~150 features!

- each tree is trained on random subset of features
- each tree outputs either "track" or "shower"

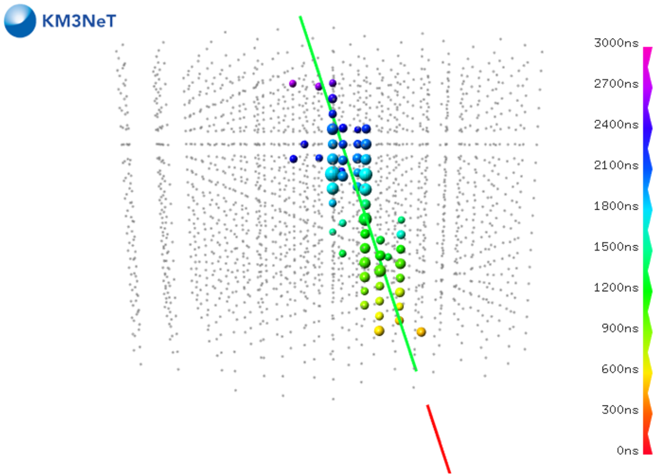
→ final score:

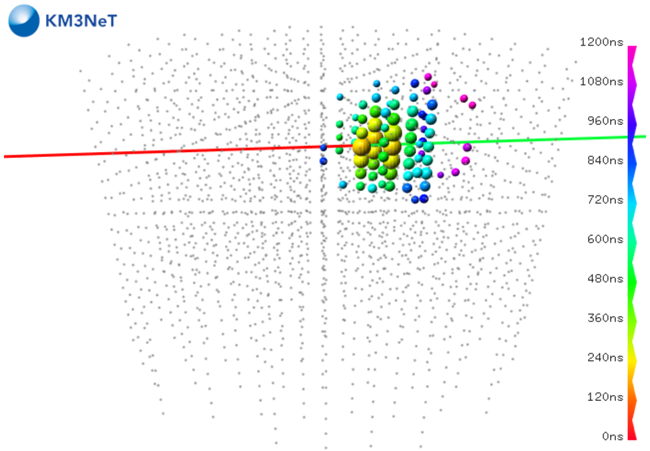
$$S = \frac{N(\text{trees voting for target class})}{N(\text{total trees})}$$

Random decision forest

what are the input features?
→ e.g. fit quality

- use maximum-likelihood-based reconstruction of observables
- compare the reco quality of track and shower hypothesis

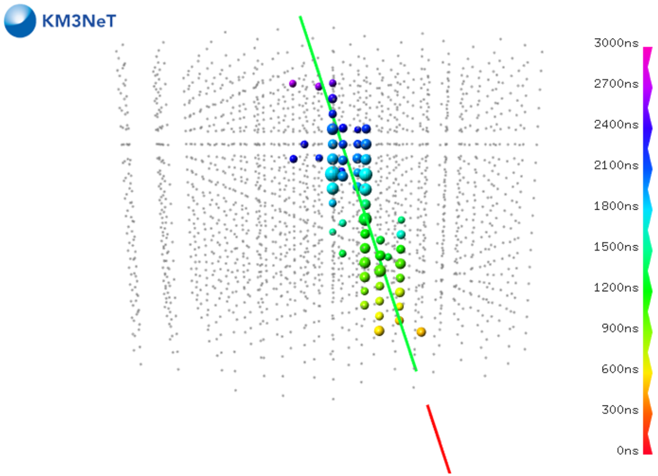


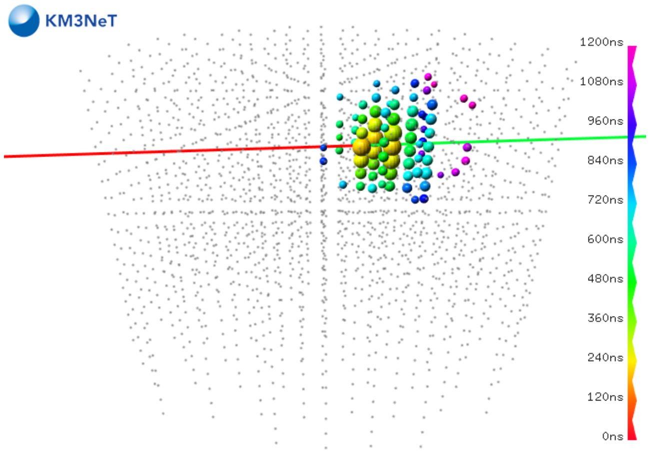


Random decision forest

what are the input features?
→ e.g. hit distribution

- compare distribution of hits to expectation in simulations
- shower events are more spherical



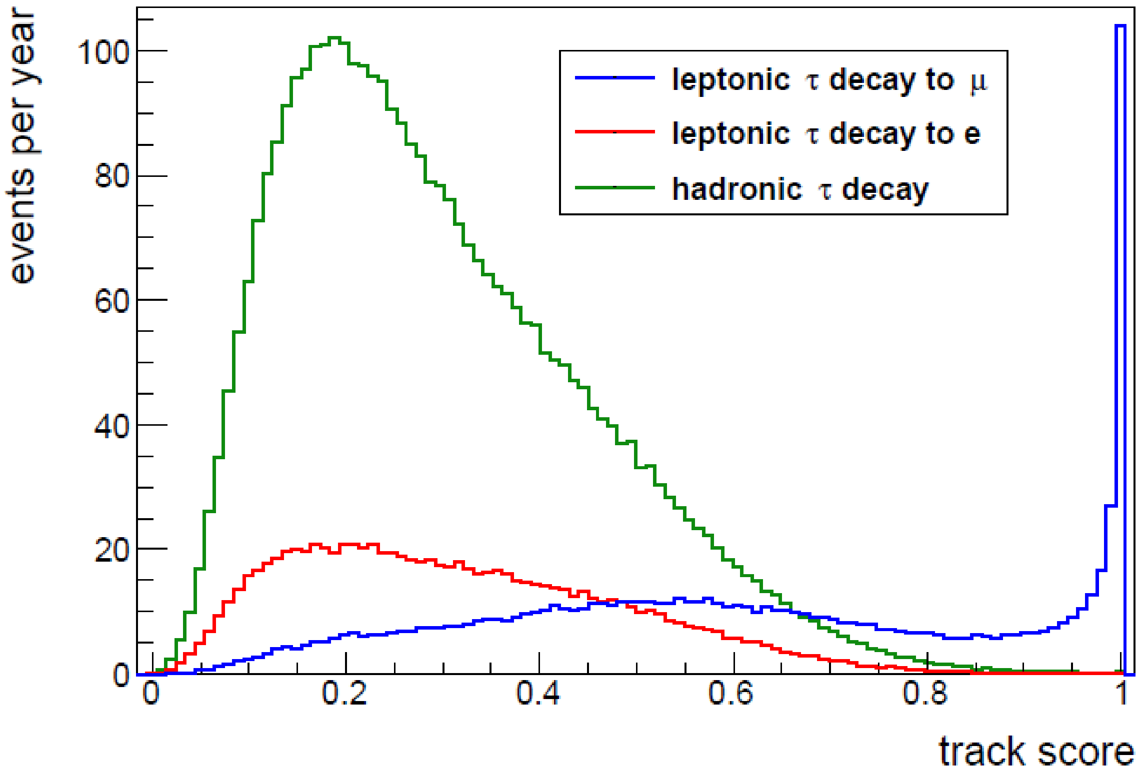


Random decision forest

Result:

- good separation between track-like and shower-like events
- define separation power *S*: quantifies the overlap between the distributions

PhD thesis Steffen Hallmann



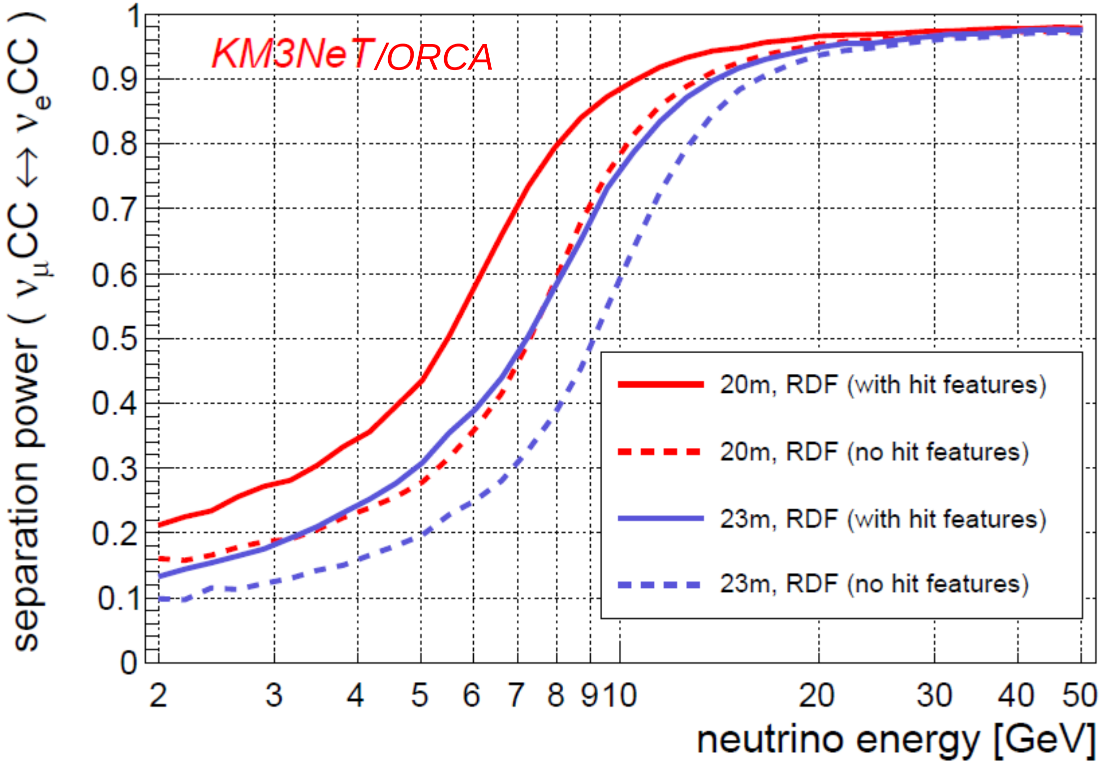
Random decision forest

Result:

- good separation between track-like and shower-like events
- define separation power *S*: quantifies the overlap between the distributions

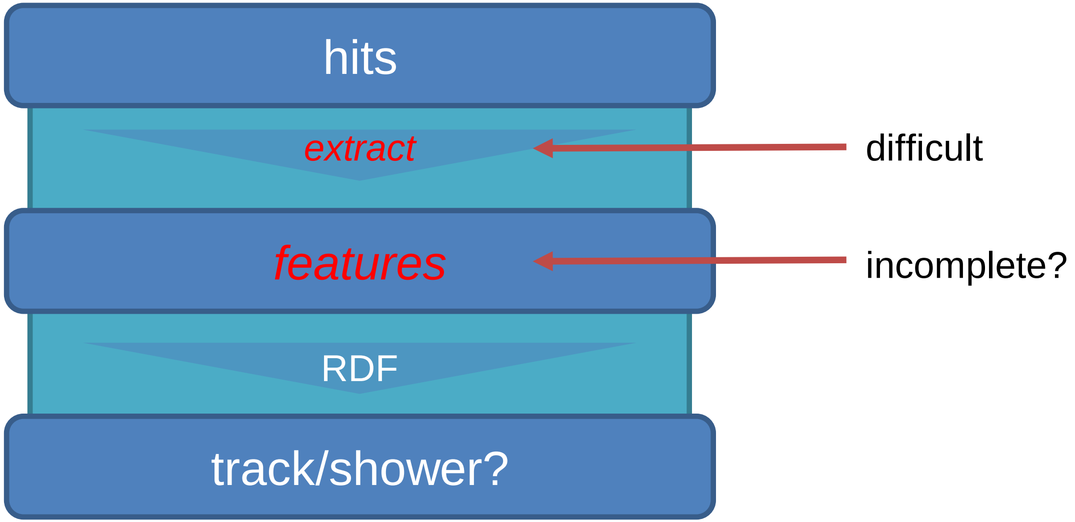
RDFs are also used for separating neutrinos from background

PhD thesis Steffen Hallmann



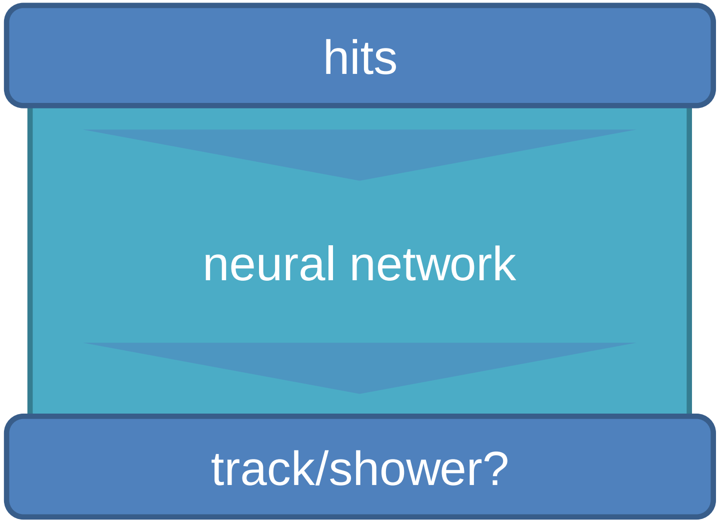
Random decision forest

- Problem: **feature design is not easy** and maybe we missed some good features?



Random decision forest

- Problem: **feature design is not easy** and maybe we missed some good features?
- idea: let an algorithm learn the features directly on low-level simulations



convolutional neural networks

KM3NeT

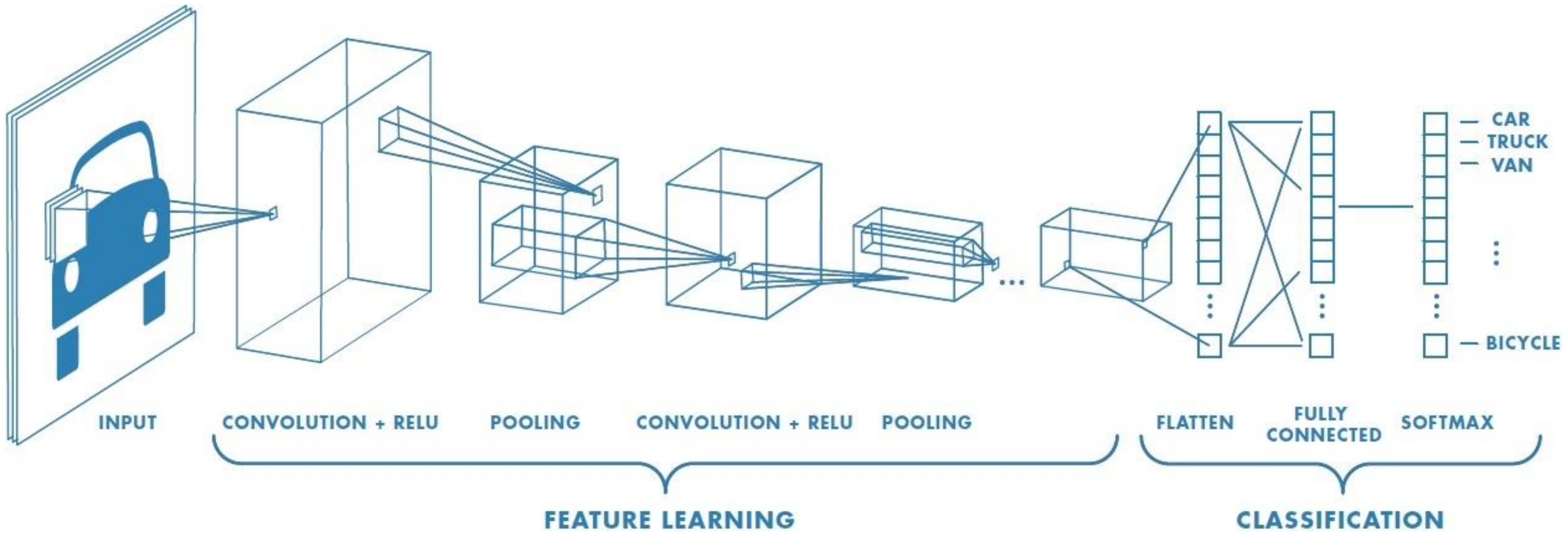
ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Convolutional networks

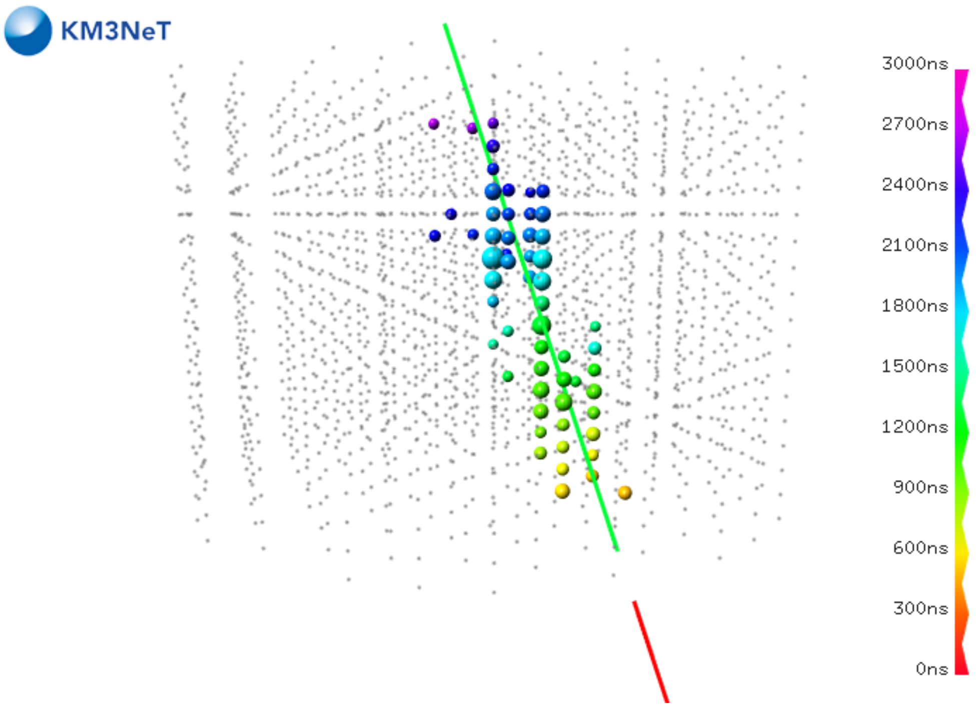
Successful model architecture in image recognition:

Convolutional neural networks (CNNs)



Simplified working principle of a CNN

Our data



How does our data look like?

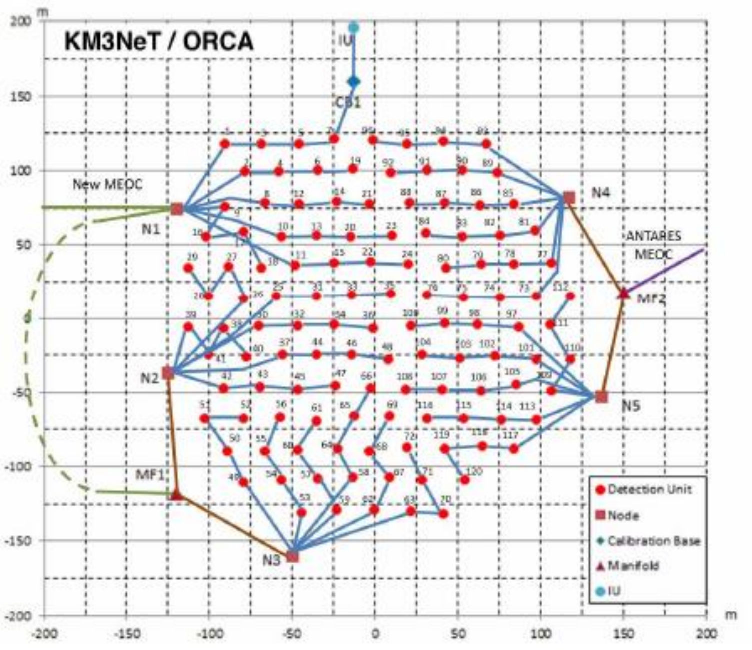
→ spatial: 3D detector
→ temporal
→ pmt direction: 31 orientations per DOM

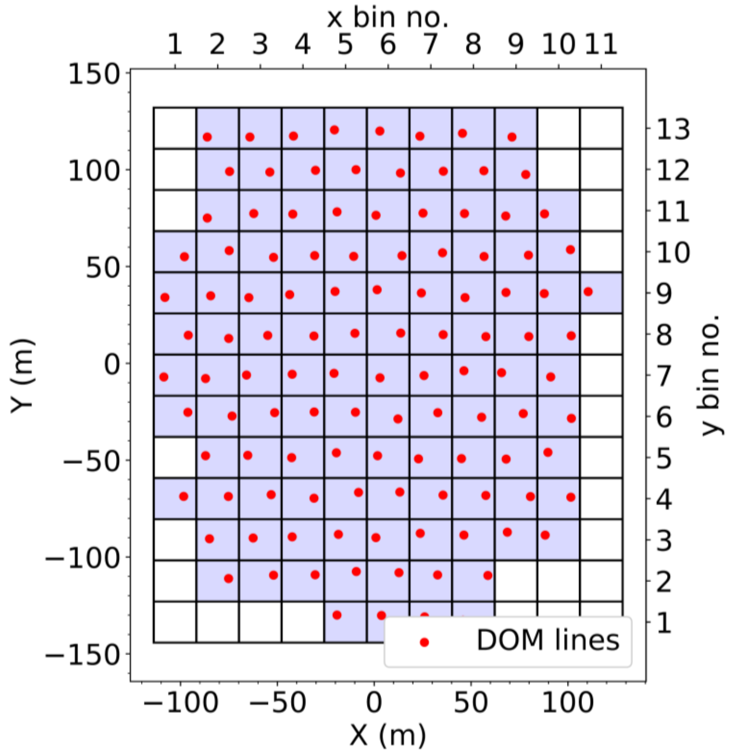
Our data: X-Y-plane

JINST 15 P10005 (2020)

- line anchors in x-y plane are not on rectangular grid
- apply grid with <1 anchor per bin (assuming static detector)

→ 11 bins in X, 13 bins in Y (ORCA115)

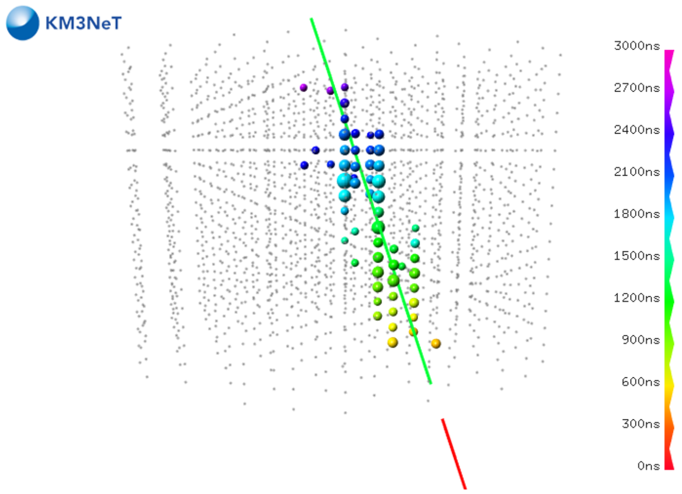




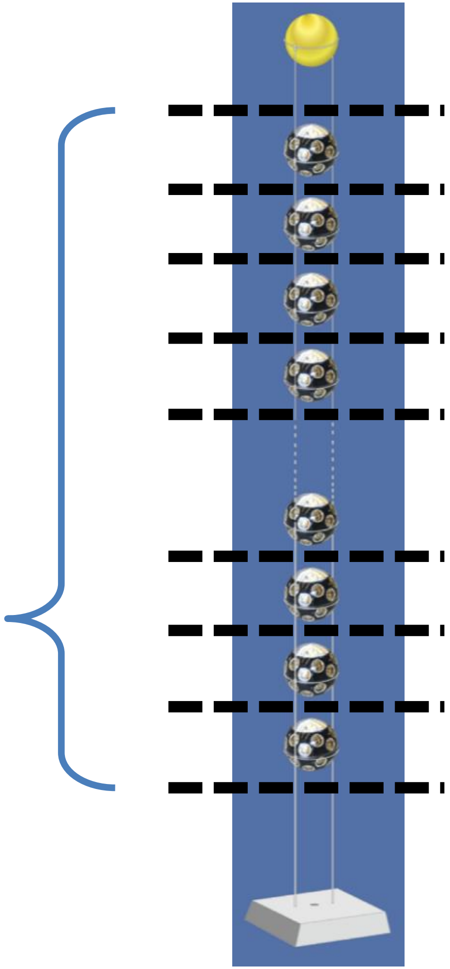
Our data: Z-plane

- each line has 18 DOMs

→ similar heights for all lines
→ can be easily binned (assuming static detector)



18 bins



Our data: time

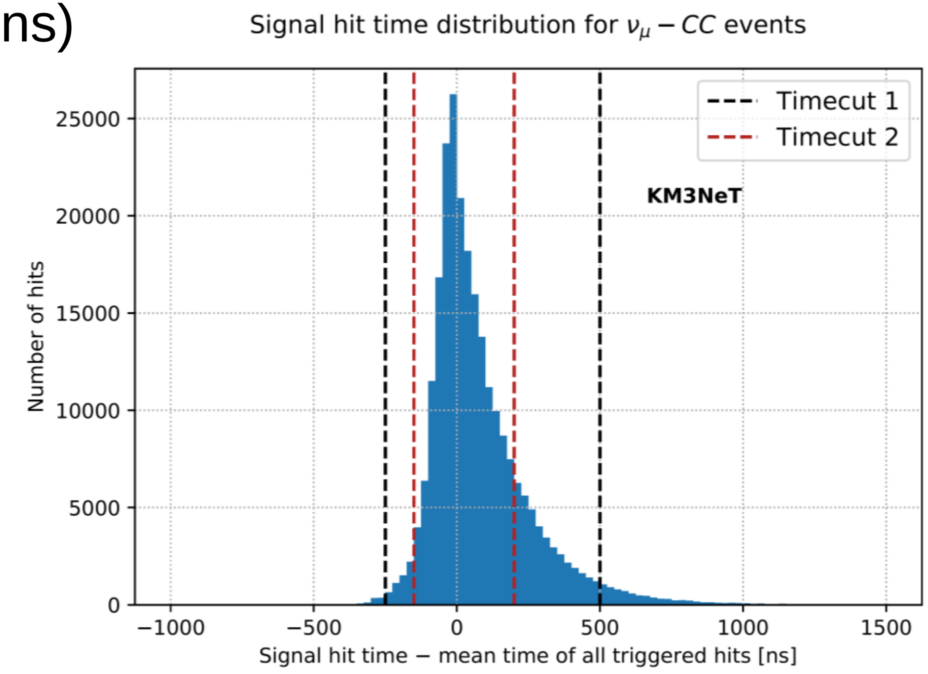
JINST 15 P10005 (2020)

- time coordinate is unbounded and continuous

→ only use hits in a time window (e.g. 750 ns)

→ choose time resolution (e.g. 7.5 ns/bin)

- time resolution limited by hardware
- choose e.g. 100 time bins



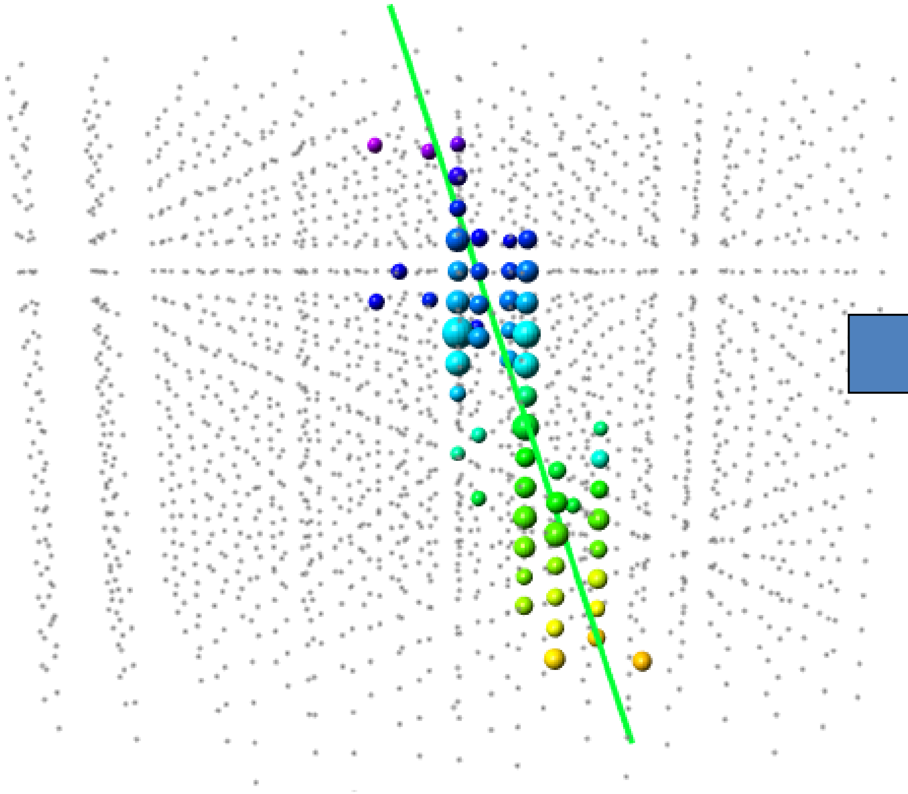
Our data: pmt direction

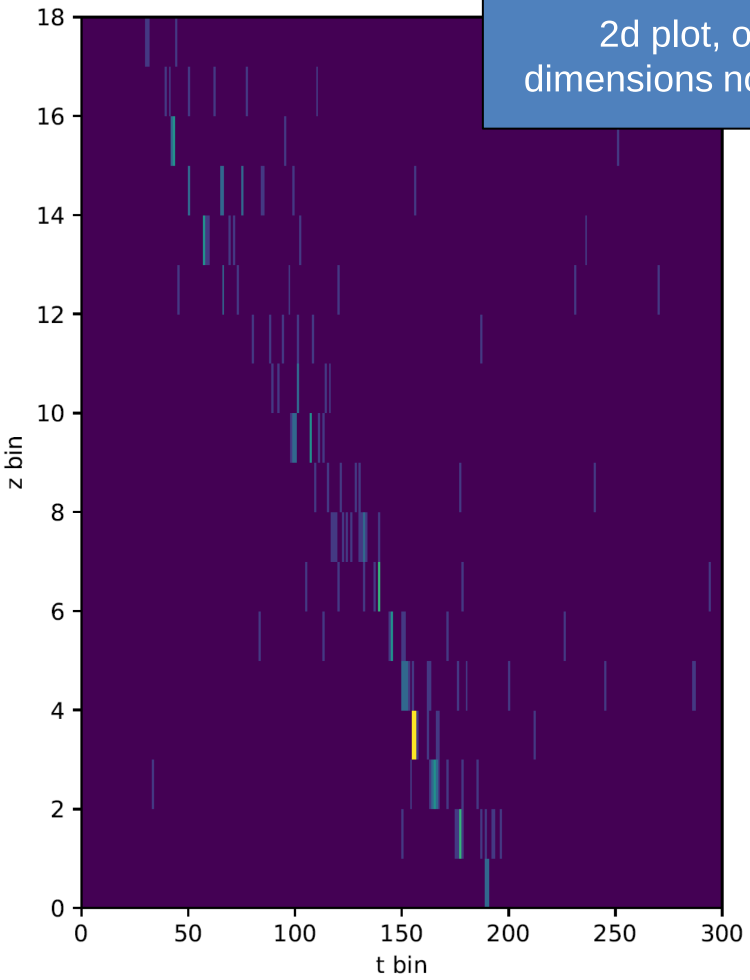
31 pmts arranged on a 2 sphere

→ no spherical convolution in tensorflow, so we use the color channel of convolutions

→ instead of multiple colors, we supply multiple pmt directions!

Input for convolutional networks





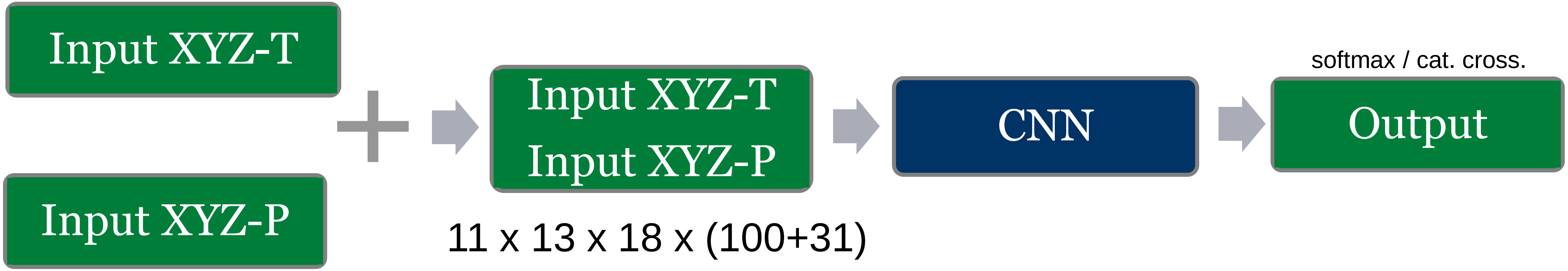
2d plot, other dimensions not shown

Input for convolutional networks

- In total, we end up with 5d data (x, y, z, t, pmt)
- But tensorflow only supports up to 4D input to convolutions!

→ Solution:

  - Stack two projections of the event: xyz-pmt and xyz-t
  - use color channel of convolution for stacked dimension

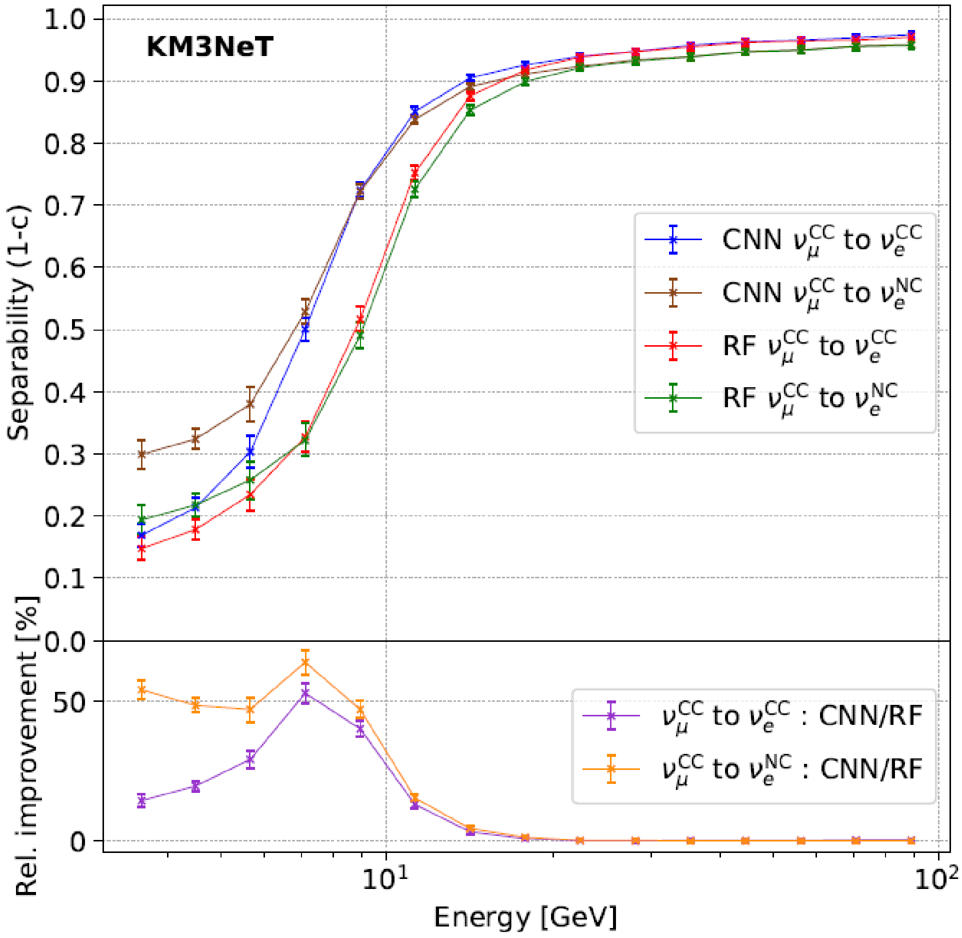


Event topology classification

JINST 15 P10005 (2020)

- Separability between track and shower

Neural network
vs
random forest

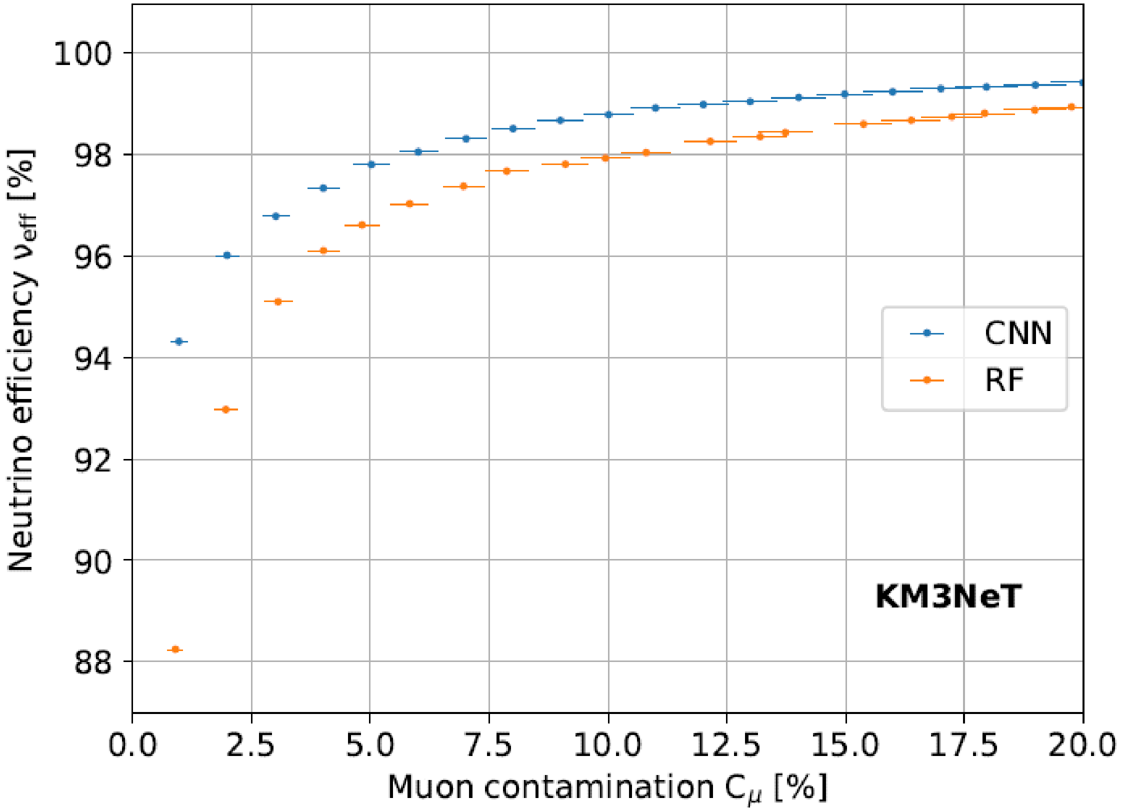


Background classification

JINST 15 P10005 (2020)

- separate atmospheric muons and neutrinos

Neutral network
vs
random forest



Graph networks

- Convolutional networks on our data have various issues:
  - no 5D convolution
  - xyz positions need to be binned (problem for non-static detector)
  - fixed time window with limited resolution

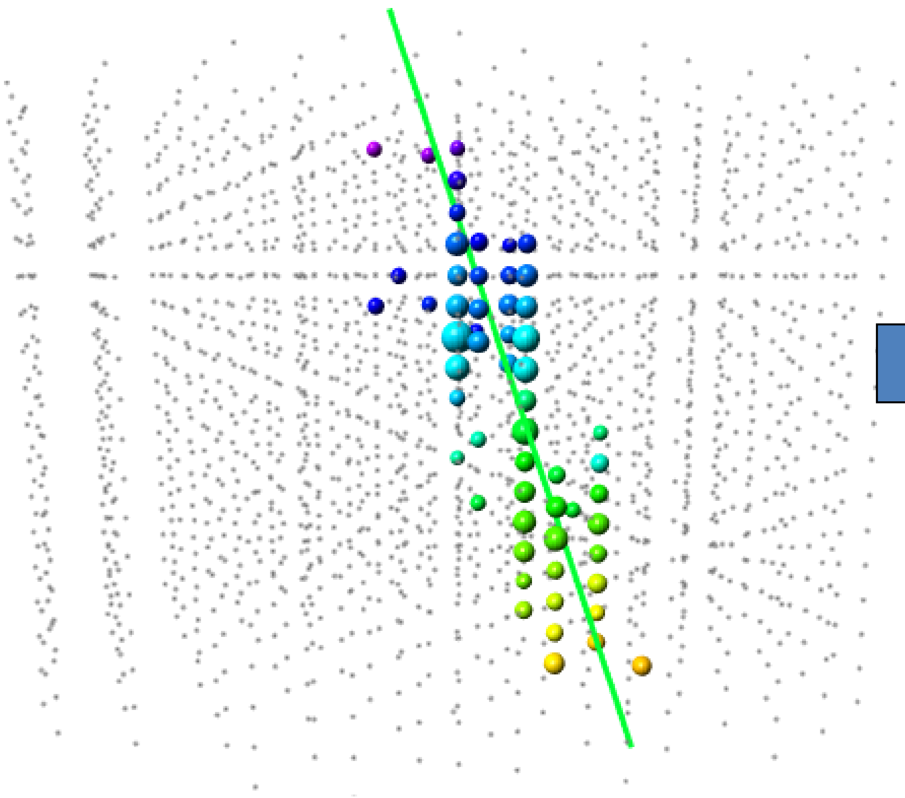
Idea: Use a network architecture that operates on graphs

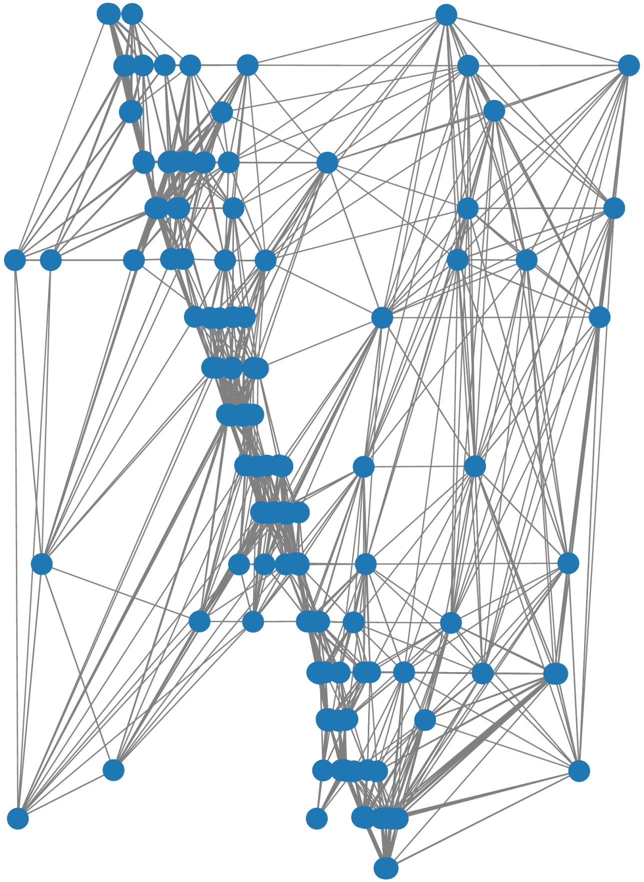
graph convolutional neural networks

ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

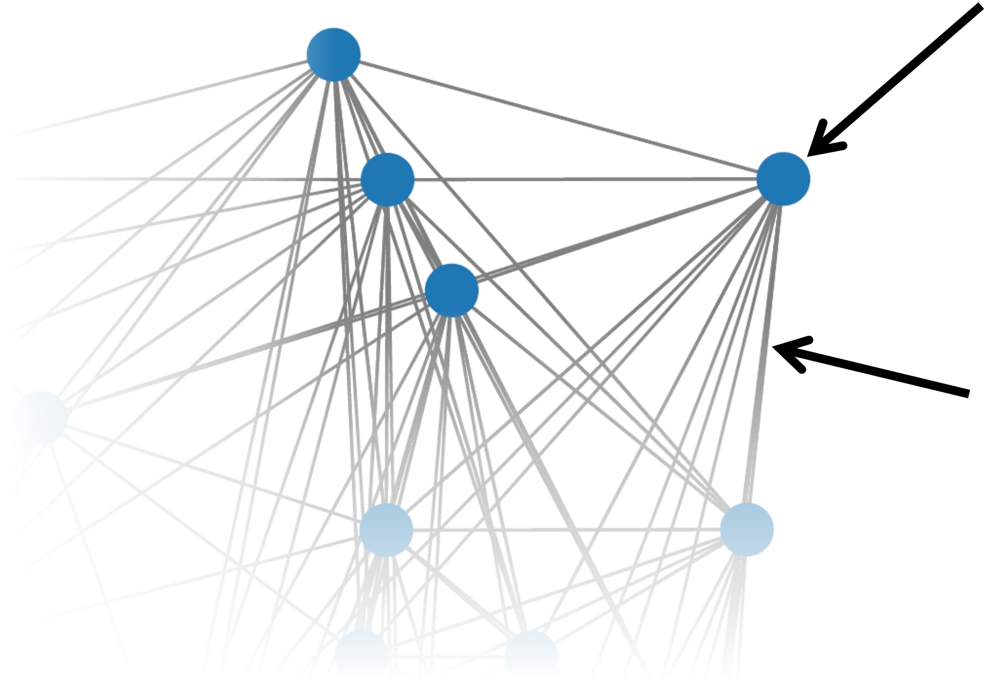
Input for graph networks





Edge convolution

https://arxiv.org/abs/1902.08570



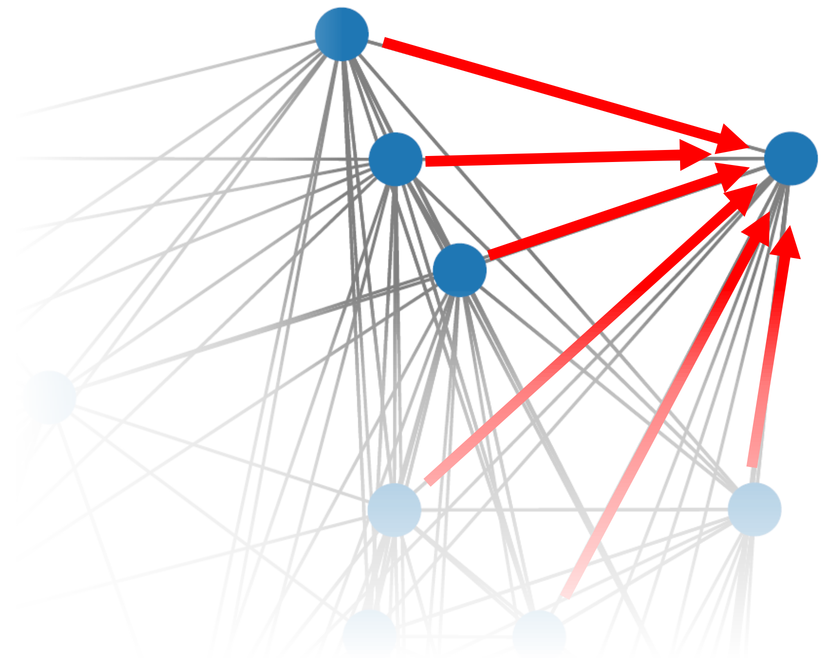
each hit is a **node** $\boldsymbol{\vec{x}_i}$

= $(x, y, z, t, \overrightarrow{pmt})$ of the hit

2 nodes are connected with **edge** $\boldsymbol{\vec{e}_{ij}}$

= $(\boldsymbol{\vec{x}_i}, \boldsymbol{\vec{x}_i} - \boldsymbol{\vec{x}_j})$ of the hits i,j

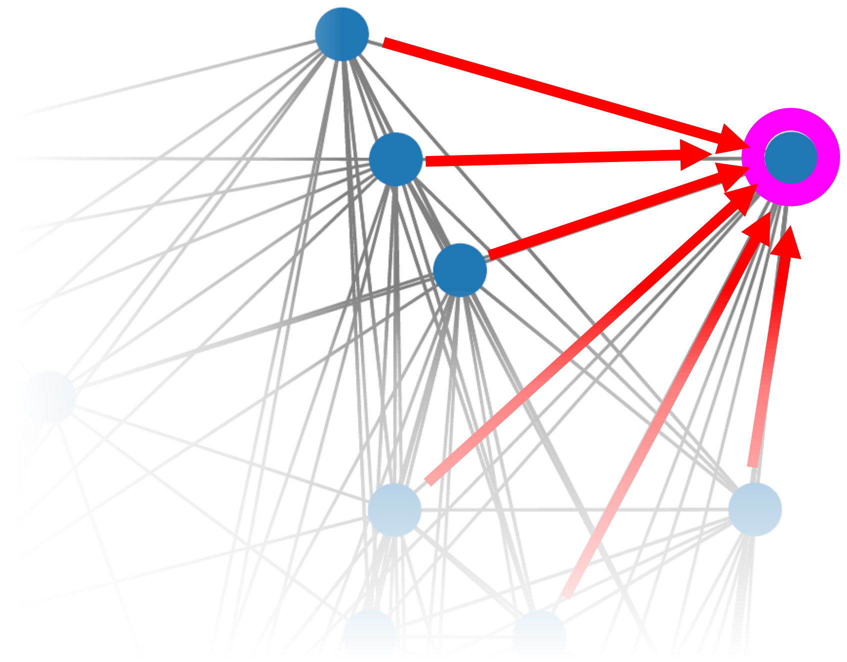
Edge convolution



Then:

- define a multi-layer perceptron and convolve over all edges
  - ➢ produces an **update** $\overrightarrow{u_{ij}}$ from each edge

Edge convolution



Then:

- define a multi-layer perceptron and convolve over all edges
  - ➢ produces an **update** $\overrightarrow{u_{ij}}$ from each edge
- Update central node $\overrightarrow{x_i}$ with averaged updates from k nearest neighbours:

$$\overrightarrow{x_i} \rightarrow \overrightarrow{x_i} + \left\langle \overrightarrow{u_{ij}} \right\rangle_j$$

Graph networks

- Convolutional networks on our data have various issues:
  - no 5D convolution **Fixed, can use n-D convolution**
  - xyz positions need to be binned (problem for non-static detector) **Fixed, no spatial binning necessary**
  - fixed time window with limited resolution **Fixed, unlimited resolution/time window**

Idea: Use a network architecture that operates on graphs

Graph networks

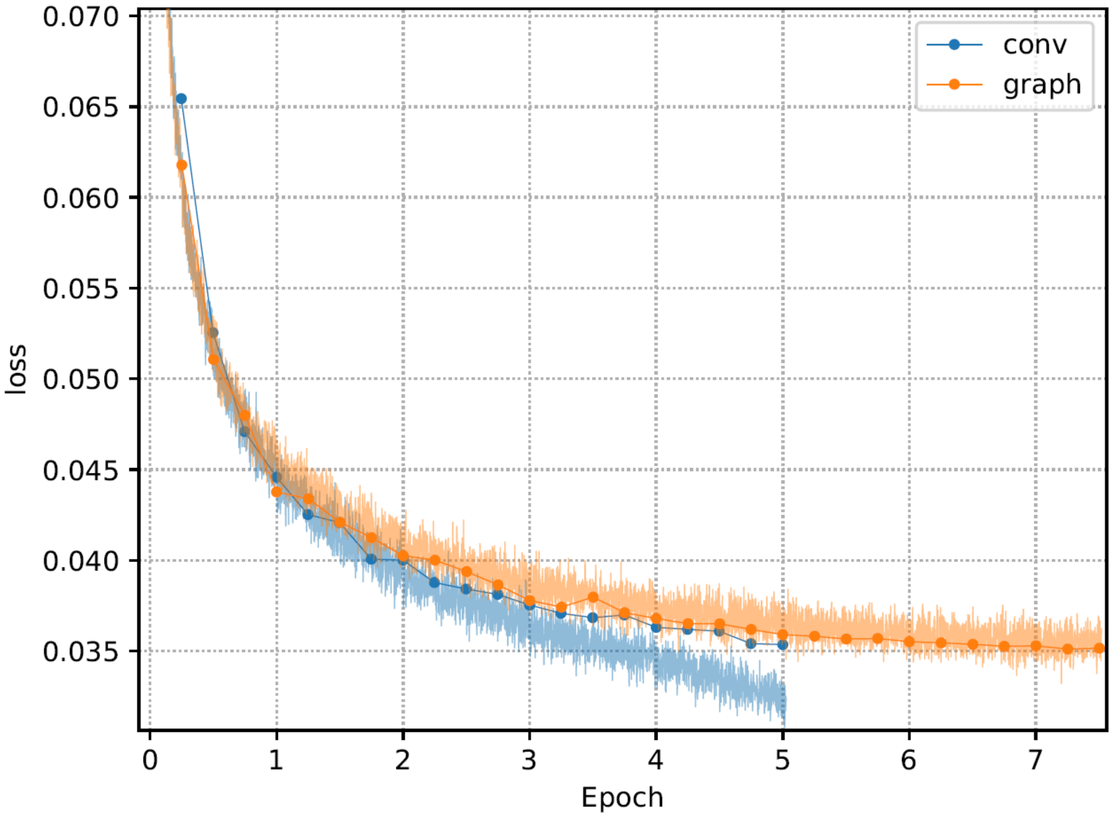
How good is the EdgeConv compared to convolutions?

	Convolution	Graph
train time / epoch	8.3h	**2.0h**
free parameters	8.4m	**370k**

→ faster, fewer parameters (→ less overfitting)

Graph networks: direction

- goal: reconstruct direction of atmospheric muons



Best validation loss

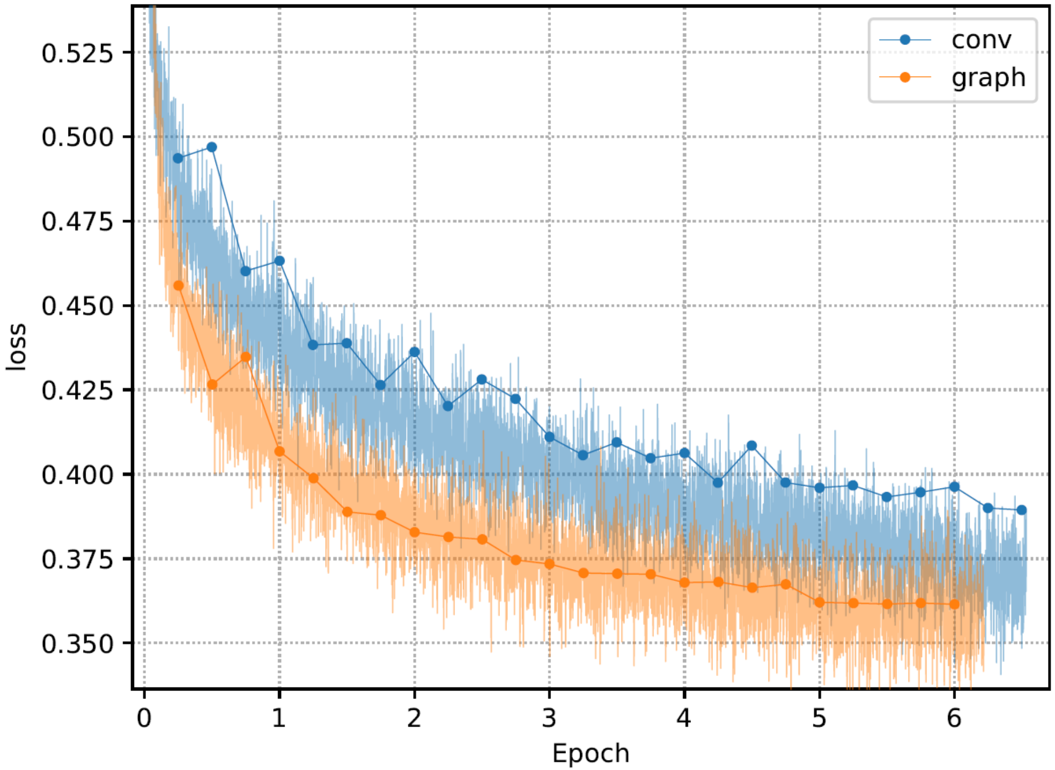
Convolution	Graph
0.0354	**0.0349**

(mean absolute error)

loss (here: mean absolute error) is used to judge performance of the reconstruction – the lower the better!

Graph networks: multiplicity

- goal: reconstruct number of atmospheric muons in an event



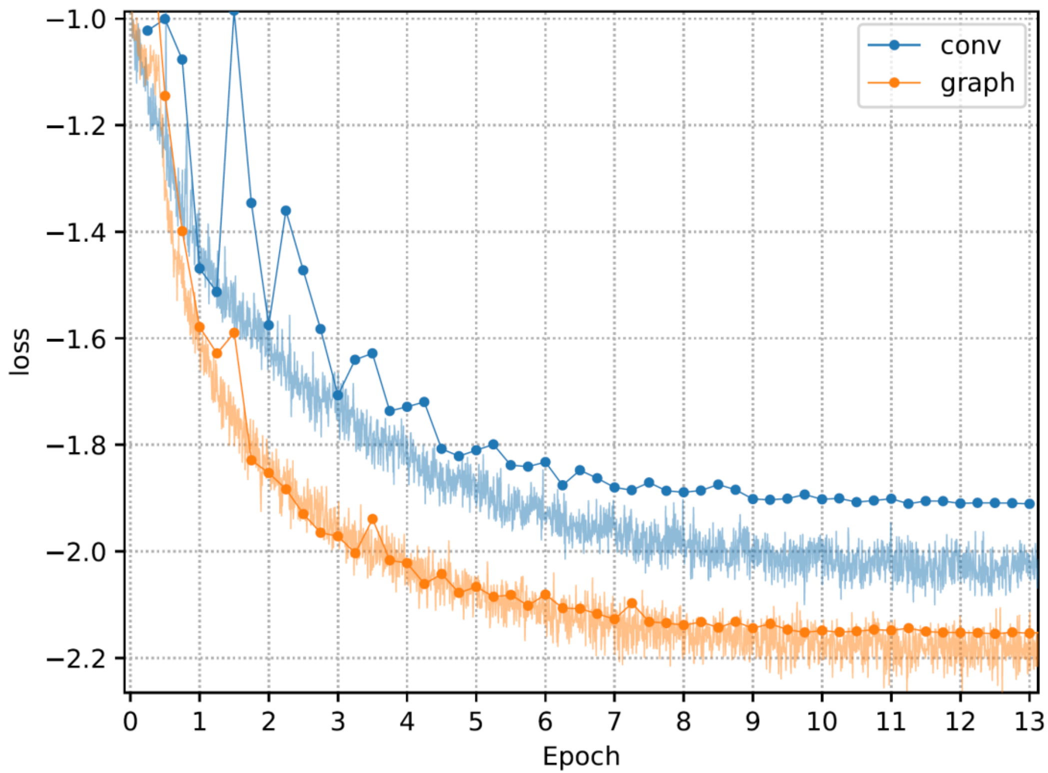
Best validation loss:

Convolution	Graph
0.389	**0.361**

(categorical cross-entropy)

Graph networks: muon distance

- goal: reconstruct distance between atmospheric muons



Best validation loss:

Convolution	Graph
-1.911	**-2.156**

(negative log-likelihood)

Summary

- Machine Learning is an important tool for event reconstruction in KM3NeT
- allows to solve otherwise difficult to tackle problems
- workflows are improved continuously and adapted to our data