KM3NeT Outing

Brían Ó Fearraigh, Schoorl 2019

ORCA
KM3NeT

Trigger Algorithms for GPUs

Trigger Algorithms for GPUs

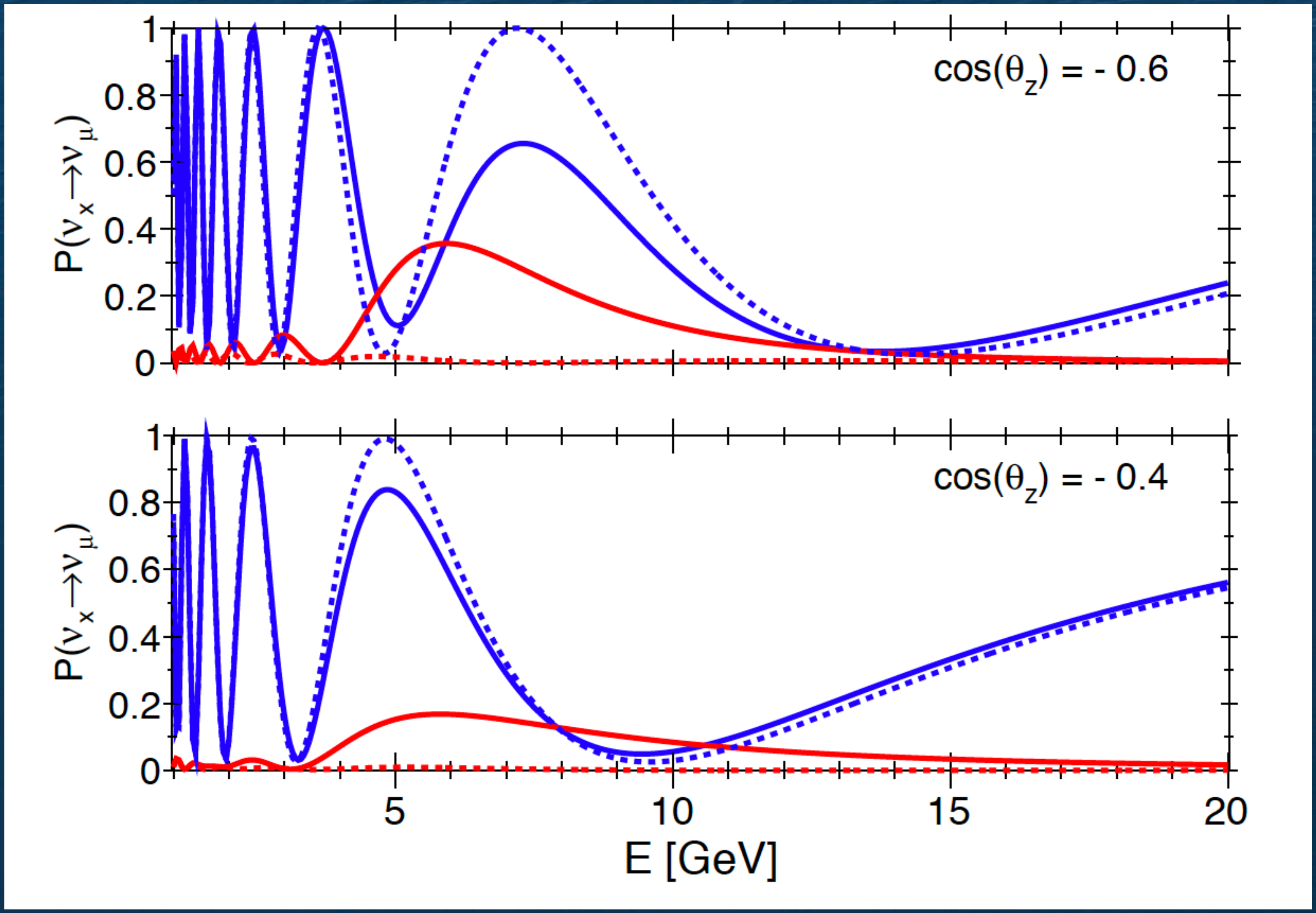
- Trigger algorithms are used to reduce data output.
- In our detector, an L0 hit is any raw PMT hit caused by a photon.
- Hits in coincidence on a DOM within a given time window (~ 10 ns) allow to go from L0 hits -> L1 hits.
- L1s are then processed by multiple trigger algorithms based on selection of causally connected hits
- We currently do not trigger on pure L0 data.

Trigger Algorithms for GPUs

- Trigger algorithms are used to reduce data output.
- In our detector, an L0 hit is any raw PMT hit caused by a photon.
- Hits in coincidence on a DOM within a given time window (~ 10 ns) allow to go from L0 hits -> L1 hits.
- L1s are then processed by multiple trigger algorithms based on selection of causally connected hits
- We currently do not trigger on pure L0 data.

Trigger Algorithms for GPUs

- For ORCA, low energy regions are of supreme interest!
  -> balance amount of noise we get with sexy neutrino events

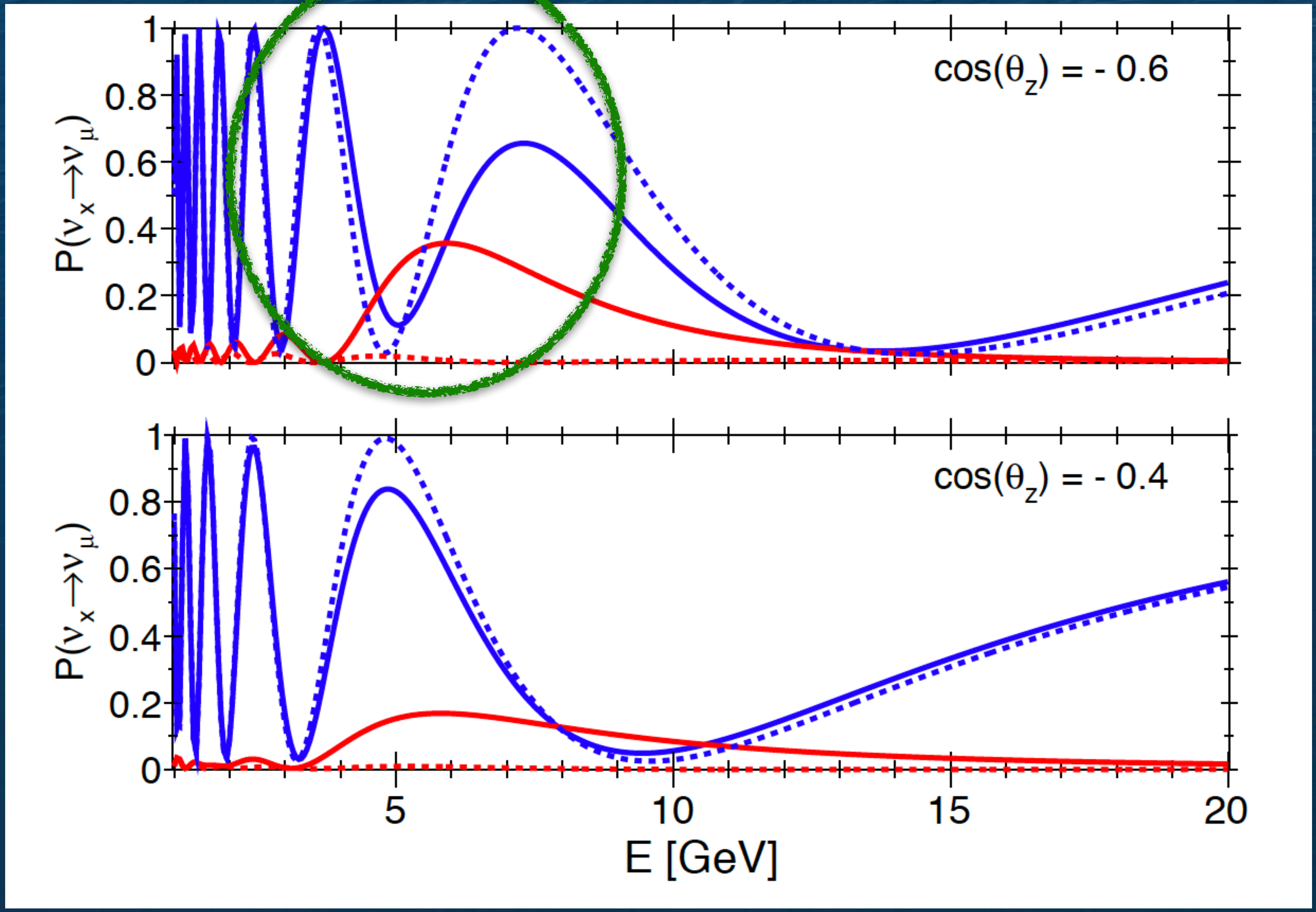


KM3NeT LOI

Trigger Algorithms for GPUs

- For ORCA, low energy regions are of supreme interest!
  -> balance amount of noise we get with sexy neutrino events

E ~ 4 - 8 GeV



Trigger Algorithms for GPUs

- Collaboration with eScience centre in Science Park to try to improve trigger algorithms and parallelise them for GPU usage.
- Ben Van Werkhoven, Stijn Heldens, Konrad Karas
- Roel Aaij from PDP in Nikhef also involved.

Trigger Algorithms for GPUs

- Random background generator provided by Roel (Python wrapper around the JPP K40 generator)

- Can generate random background events ‘on the fly’ in the form of a 4 x N ndarray of the rates, PMT ID, DOM ID and ‘ToT’

Trigger Algorithms for GPUs

```
import k40gen
gens = k40gen.Generators(21341, 1245, [7000., 700., 70., 0.])
generated_array = k40gen.generate_k40(0, int(1e8), gens, 'orca',False)
generated_array
ay([[  645700,   773000,   826492, ..., 95592595, 95592596, 96623739],
    [       1,        1,        1, ...,     2070,     2070,     2070],
    [       0,        0,        0, ...,       28,        5,        2],
    [      27,       26,       26, ...,       31,       26,       27]],
  dtype=int64)
generated_array[4]
ceback (most recent call last):
ile "<stdin>", line 1, in <module>
exError: index 4 is out of bounds for axis 0 with size 4
generated_array[:4]
ay([[  645700,   773000,   826492, ..., 95592595, 95592596, 96623739],
    [       1,        1,        1, ...,     2070,     2070,     2070],
    [       0,        0,        0, ...,       28,        5,        2],
    [      27,       26,       26, ...,       31,       26,       27]],
  dtype=int64)
```

Python K40 simulation: https://github.com/nlesc-km3net/k40gen

Trigger Algorithms for GPUs

- Using km3pipe these background events can be 'calibrated': assigned a position and direction from the appropriate detector file
- Couple with simulated events which can also be calibrated. These events go through the PMT simulation of JTriggerEfficiency where hits are merged.

Trigger Algorithms for GPUs

- On the eScience side: set up pipeline to run trigger algorithms on the GPUs (memory allocation).
- On our side: provide events pre-triggering which are understood, evaluate output of GPU algorithms, implement in the framework we use in KM3NeT.

Trigger Algorithms for GPUs

- Louvain algorithm used for detecting communities in networks. See https://arxiv.org/pdf/0803.0476.pdf .
- Communities are groups of nodes (hits) within a network that are more densely connected to one another than to other nodes.
- Essentially a means of recognising hits in the detector and determining whether these are physics or background.

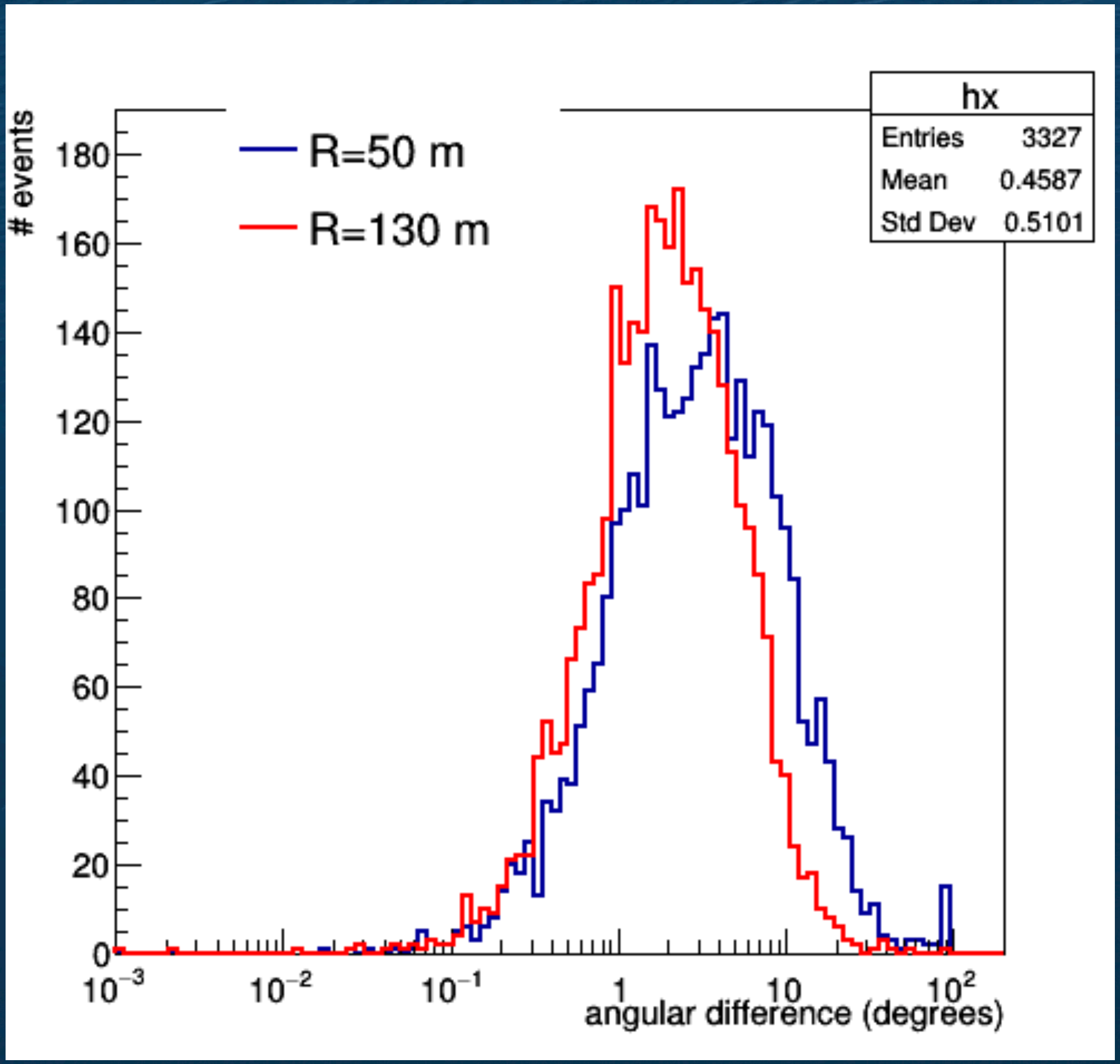
Optimising Current Track Reco

Optimising Current Track Reco

Optimising Current Track Reco

- JMCEvt ->JGandalf gives a much improved angular resolution with increased roadwidth parameter.

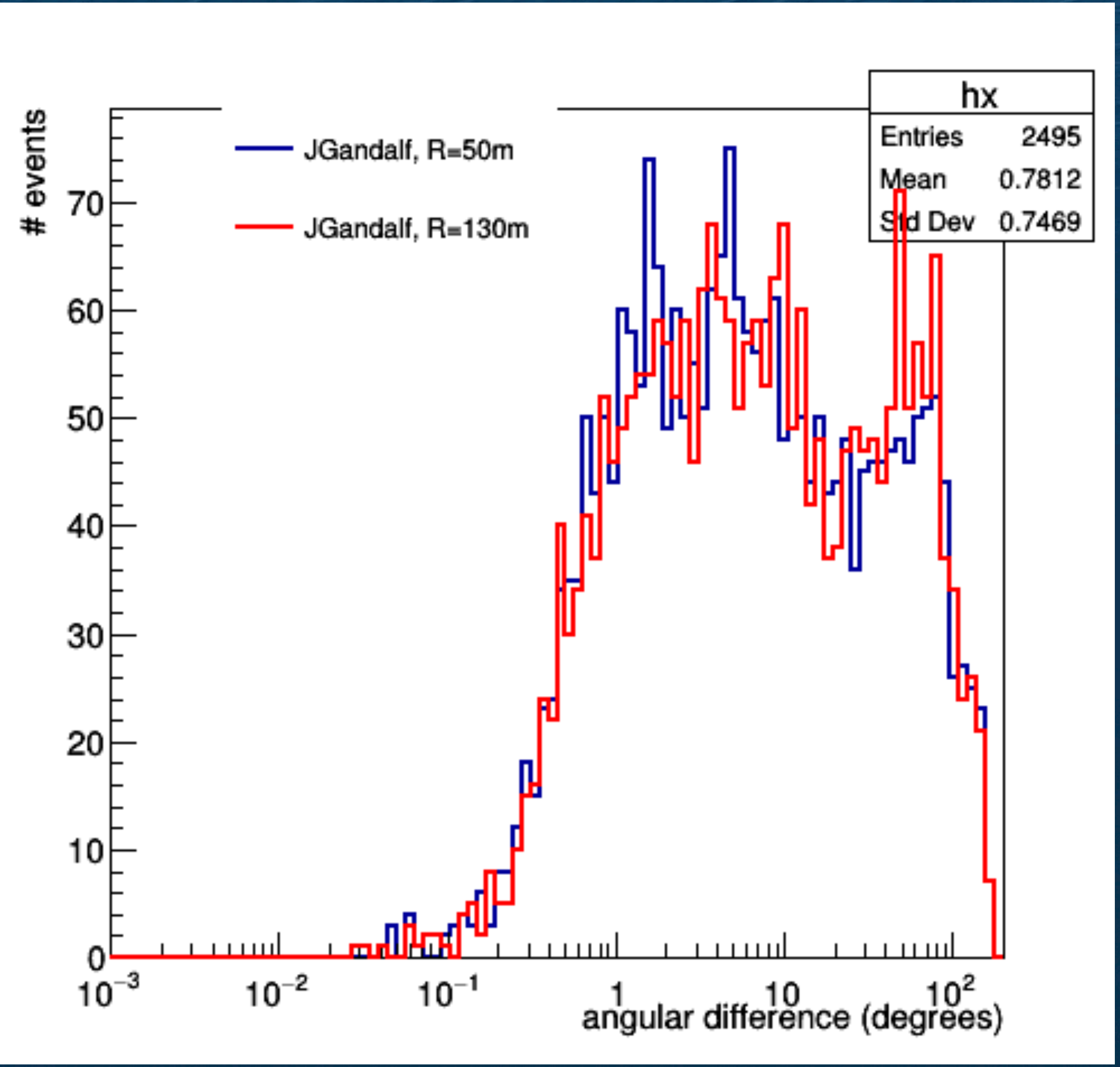
- This plot (although using MC information) is something to aim for.



Optimising Current Track Reco

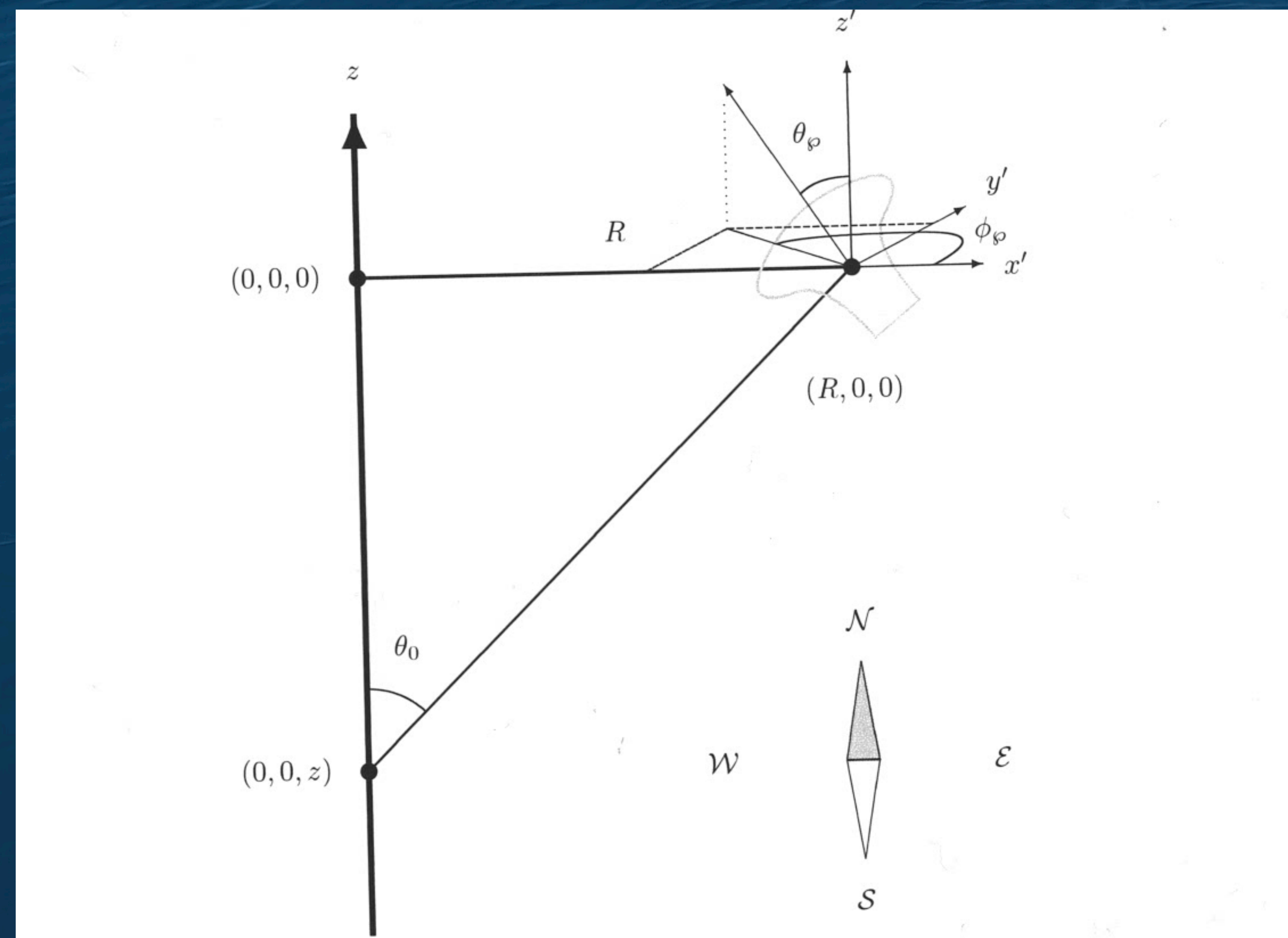
- JPrefit ->JGandalf shows no difference in angular resolution.

- NB: plot is misleading. Only one Gandalf fit with highest quality parameter is shown. Does not mean a terrible angular resolution, no energy weight.



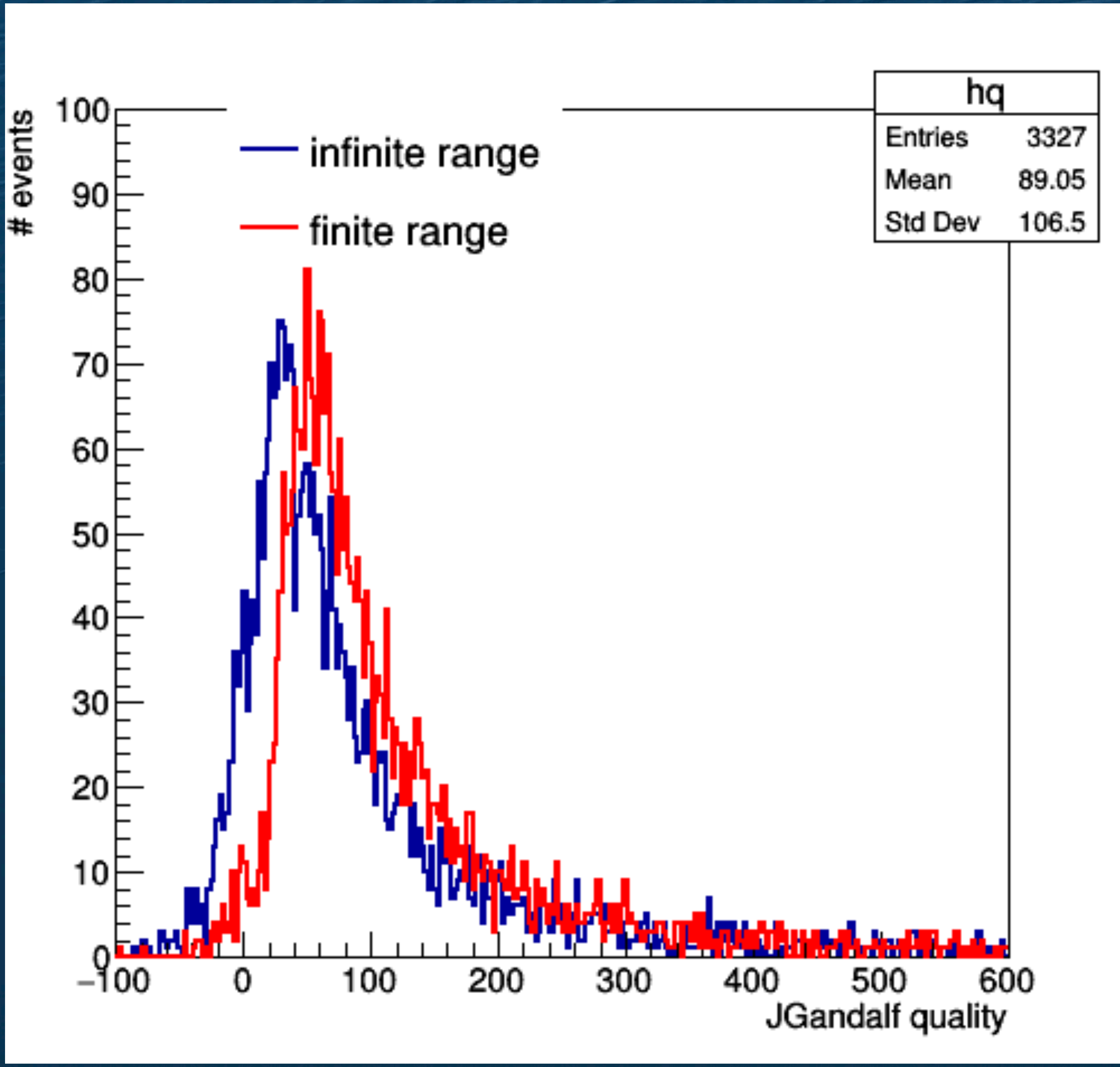
Optimising Current Track Reco

- Lead me to implementing muon range in JGandalf for ORCA .
- Currently giving a higher quality but a poorer resolution.



Optimising Current Track Reco

- Lead me to implementing muon range in JGandalf for ORCA .
- Currently giving a higher quality but a poorer resolution.



Questions, for me

- How can I fix this? Am I stuck with a worse angular resolution with a muon range included in JGandalf? Is Gandalf this reliant on background?
- (Is this worth showing in Nantes? Roadwidth used in JGandalf should be increased.)
- Will I ever run out of pictures of Gandalf to show you?

Questions, for us

- How can I fix this? Am I stuck with a worse angular resolution with a muon range included in JGandalf? Is Gandalf this reliant on background?
- (Is this worth showing in Nantes? Roadwidth used in JGandalf should be increased.)
- Will I ever run out of pictures of Gandalf to show you?
- How can we improve track reconstruction in ORCA?
- How can we improve energy reconstruction in ORCA?

Anyway, I'm sure this will happen when we are up and running properly

Speaking of black and white things..

Thanks for listening!