Graph Convolution Networks for FTS

Panda Collaboration Meeting I/20

10.03.20 I Waleed Esmail

Institut Für Kernphysik (IKP), Forschungszentrum Jülich

Current Approach (local Method):

I. **Create Track Segments** by using a simple feed-forward network

to accept **hit-pairs**, and decides weather

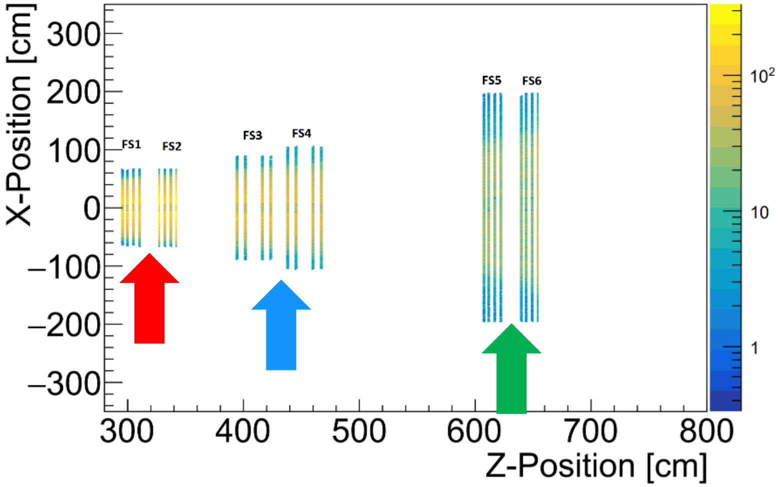
or not they are on the **same track or not**

Track segments in

(FS1+FS2)

(FS3+FS4)

(FS5+FS6)



II. **Interpolate Track Segments** by using recurrent neural network.

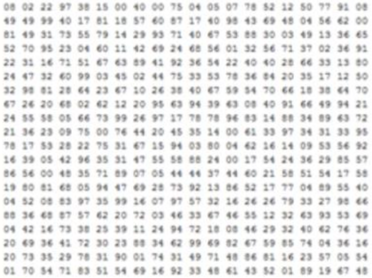
	TrackSeg 1	TrackSeg 2	TrackSeg 3
TrackSeg 1	🟦	🟥	🟥
TrackSeg 2	🟥	🟦	🟥
TrackSeg 3	🟥	🟥	🟦

Convolution Neural Network (CNN):

- The convolutional neural network architecture is central to computer vision (Image classification, Object detection).

- The input is an image (2D array of pixels).

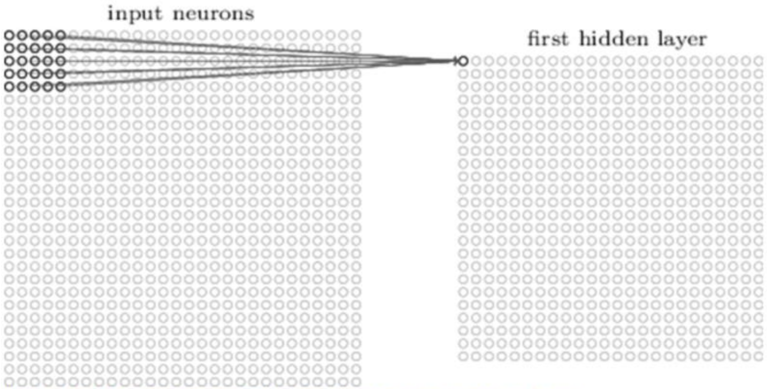
What We See



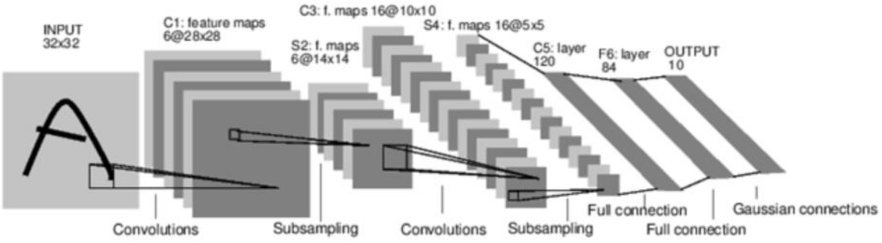
```
08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48
```

What Computers See

- The first layer in a CNN is always a **Convolutional Layer**.
- Learning weights here are called **filters.**



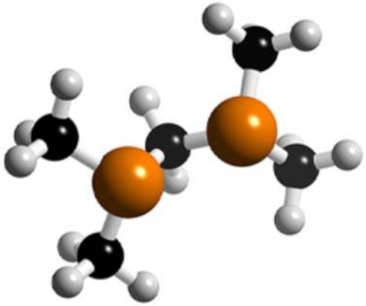
Visualization of 5 x 5 filter convolving around an input volume and producing an activation map



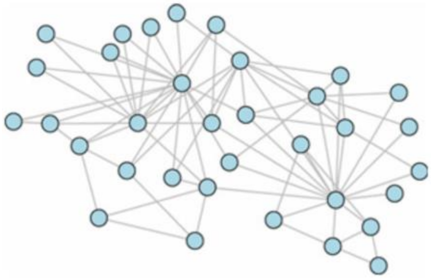
Graph Neural Networks (GNN):

- The vast majority of deep learning is performed on **Euclidean data.**

- There are also **Non-euclidean** (e.g. social networks) data can represent more complex items and concepts with more accuracy than 1D or 2D representation.



Molecules



Networks

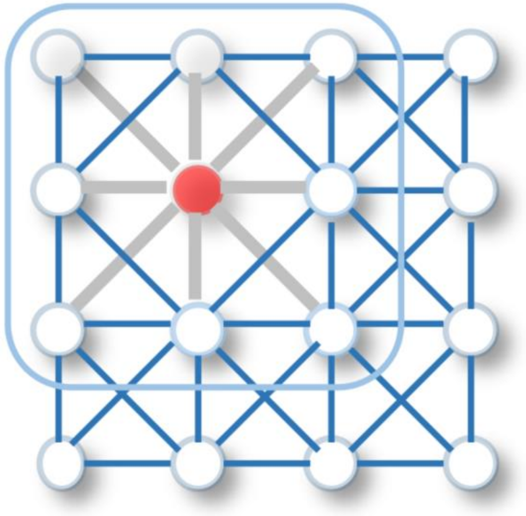
- The prime example of a **non-euclidean** datatype is a **graph**.

- **Graphs** are a type of data structure that consists of **nodes** (entities) that are connected with **edges** (relationships).

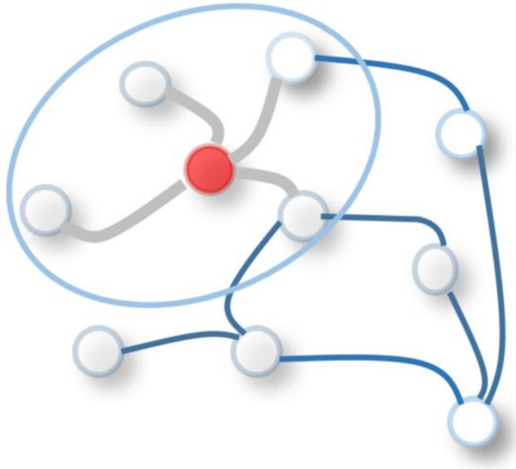
- Building on this intuition, **Graph Neural Network (GNN)** aims to learn from non-euclidean data (**graphs**).

Graph Convolution Networks (GCN)

.



The 2D convolution takes the weighted average of pixel values of the red node along with its neighbours (determined by the filter)



Graph Convolution is an operation to take the average value of the node features of the red node along with its neighbours (but nodes/pixels) are unordered and variable in size)

Message Passing Neural Networks (MPNN) aim to learn node representations with recurrent neural architectures (**graph iterations**)

It has two phases: **message passing** phase, **read-out** phase.

A Comprehensive Survey on Graph Neural Networks (arXiv 1901.00596)

GCN applied to FTS (Global Method):

- GCN is used as a **binary classifier** (hit-pairs classification or **edge classification**).
- **Input** is a **graph** (FTS hits of one event only vertical layers).
- Two main components: **edge network** and **node network.**
- Edge network uses the **node features** to compute edge weights.
- Node network **aggregates node features** with the edge weights and **updates node features.**
- With each **graph iteration**, the model **propagates information** through the graph, strengthens important connections, and weakens useless ones.

node features = [x, z, isochrone, layer_id, phi].

graph iterations = 5

Graph Construction:

- ·Box Generator:

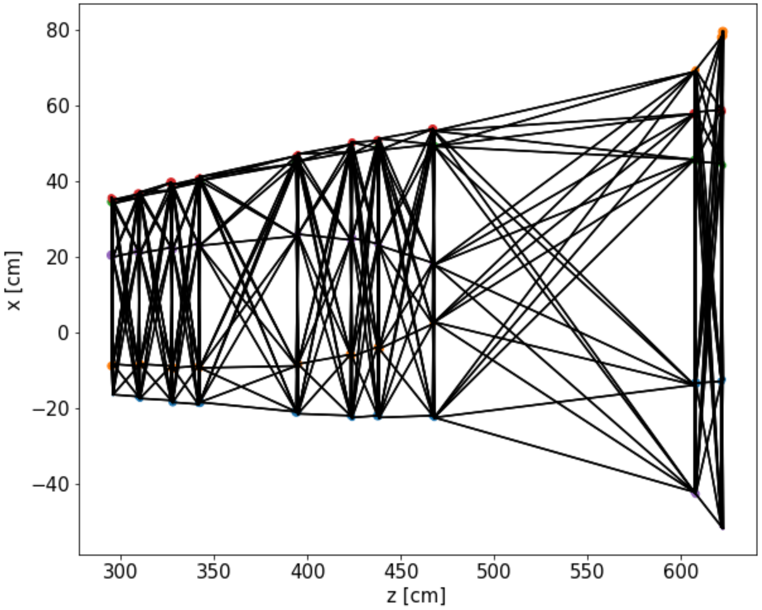
  Momentum 0.1 – 6 GeV/c

  Polar angle 0 – 10º.

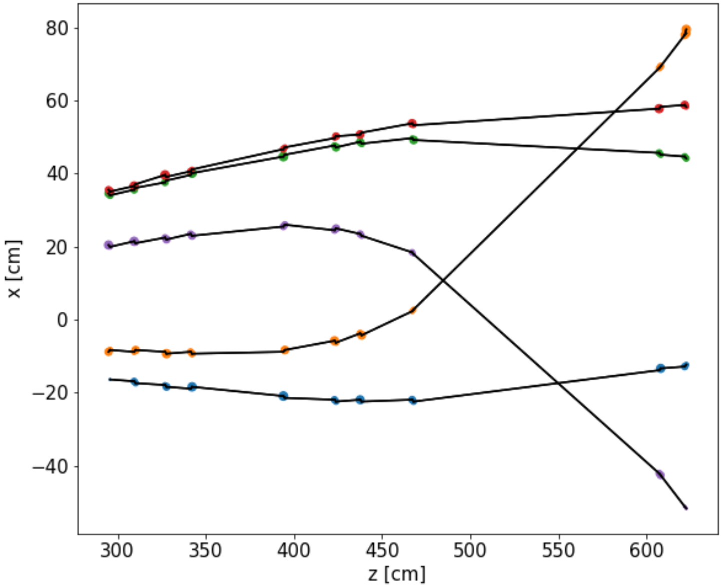
  6 tracks per event $\mu^{\pm}$.

- Trained on 10k events (Classification Accuracy 98.5%).

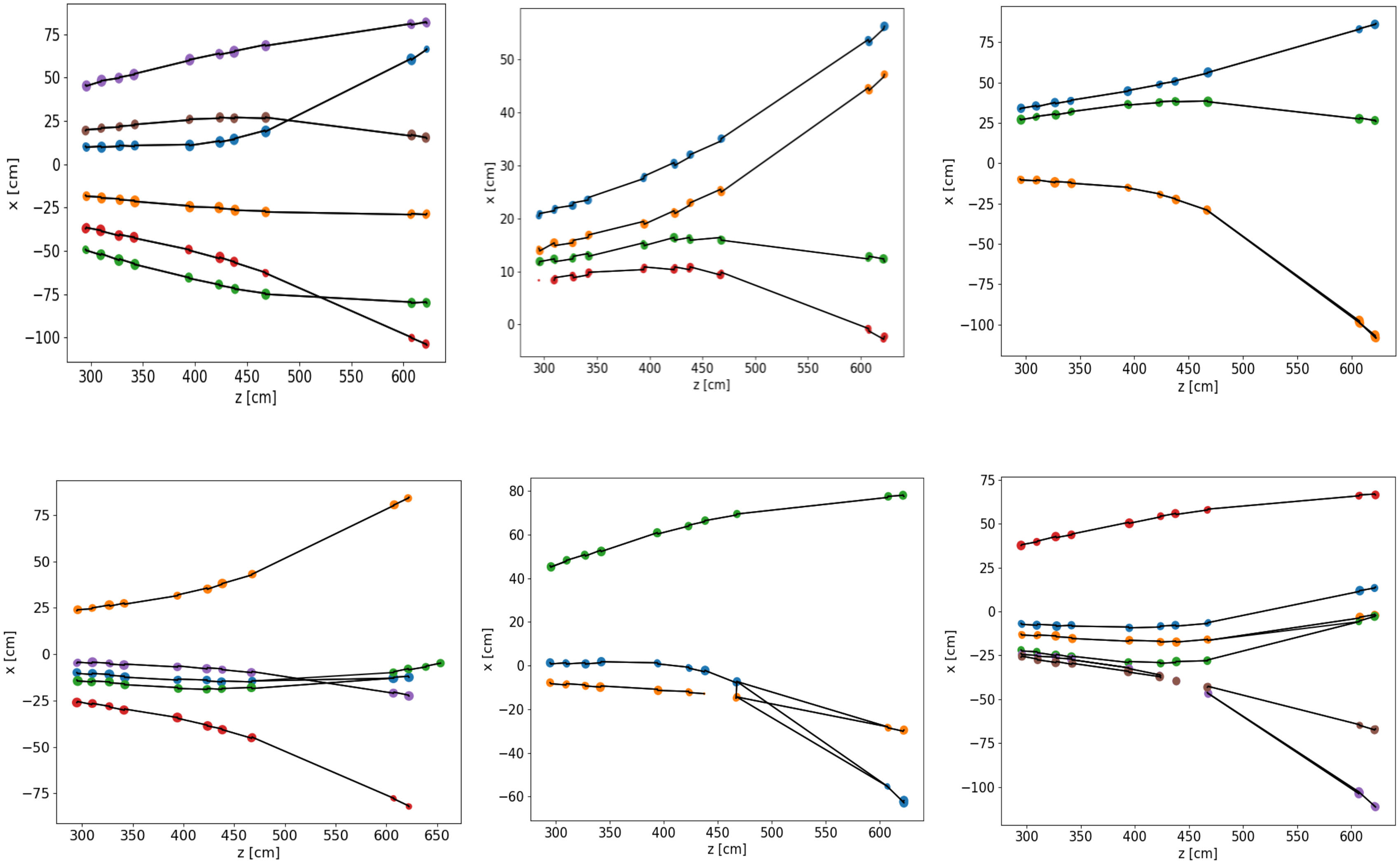
Whole event graph



Ideal output

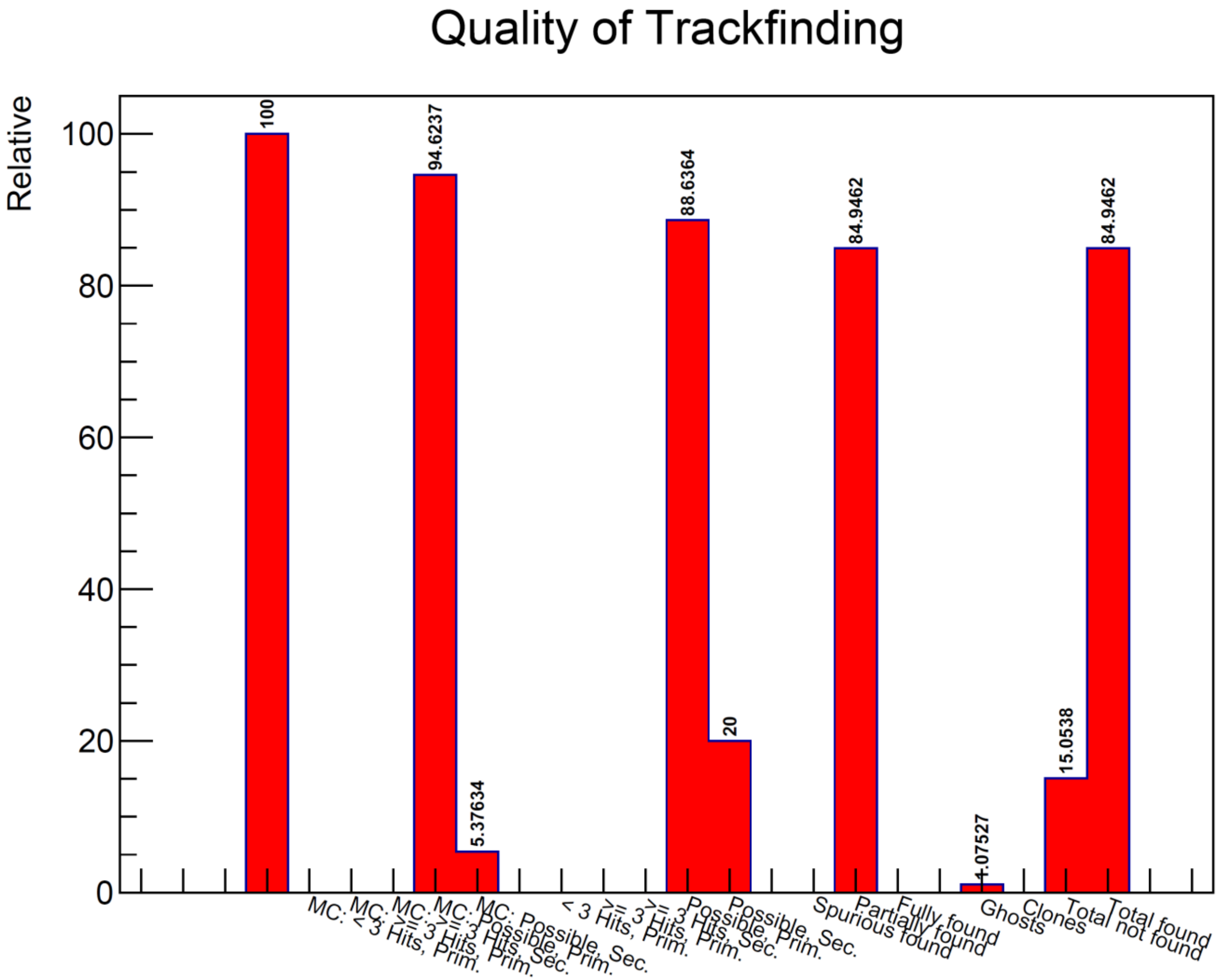


Example Output Graphs:



Network Performance:

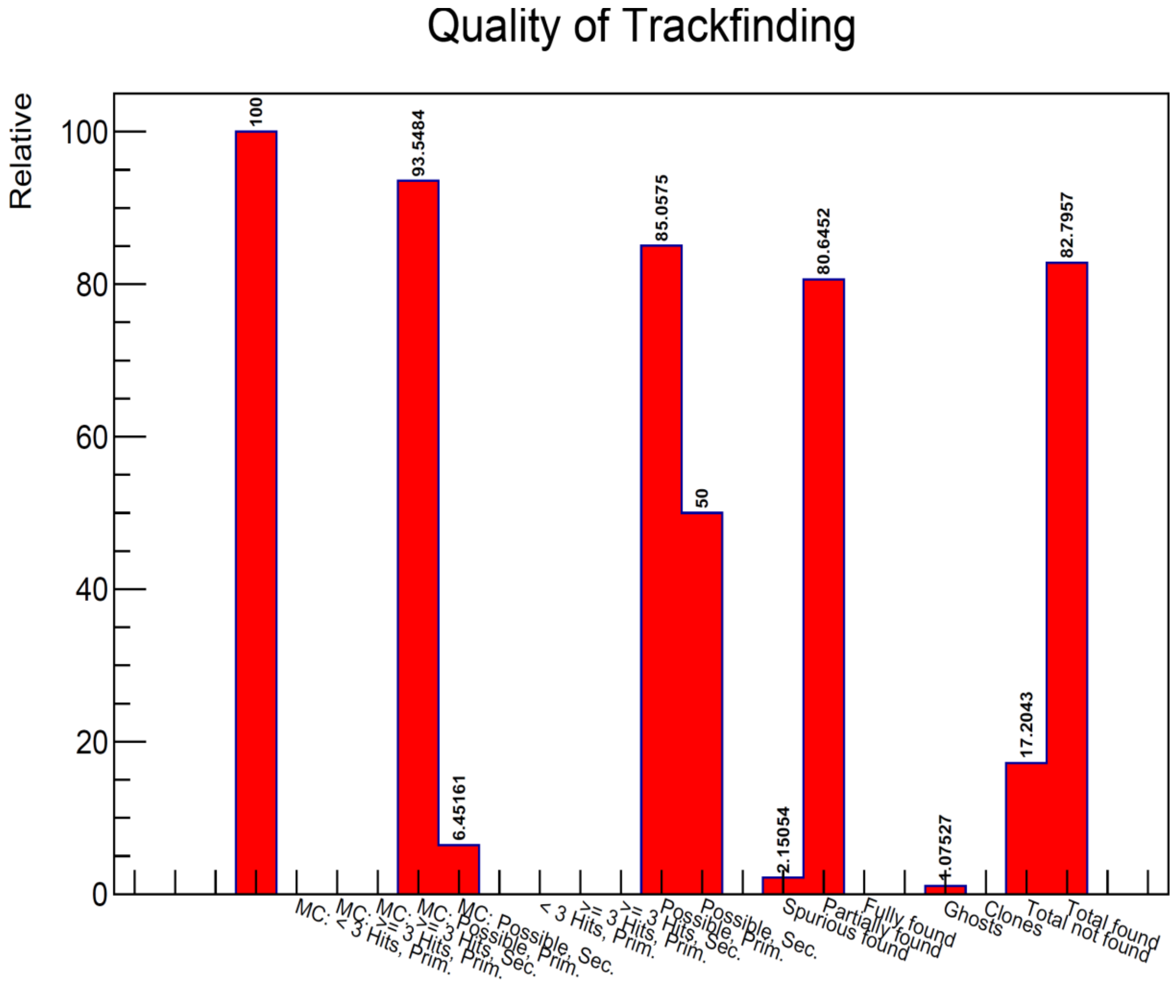
Box Generator **1-track/event**, **momentum 1 GeV/c**



JÜLICH
Forschungszentrum

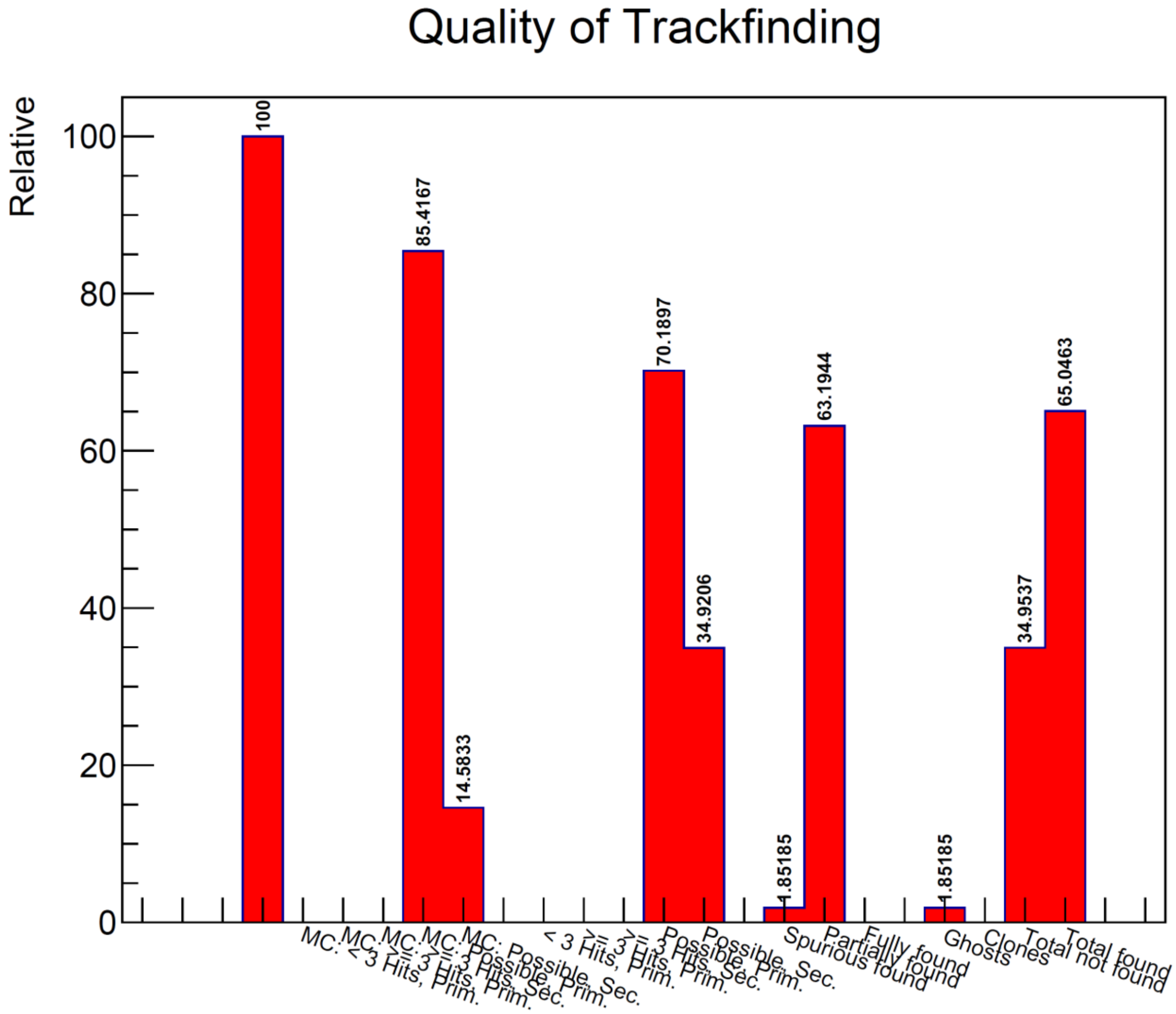
Network Performance:

Box Generator **1-track/event**, **momentum 3 GeV/c**



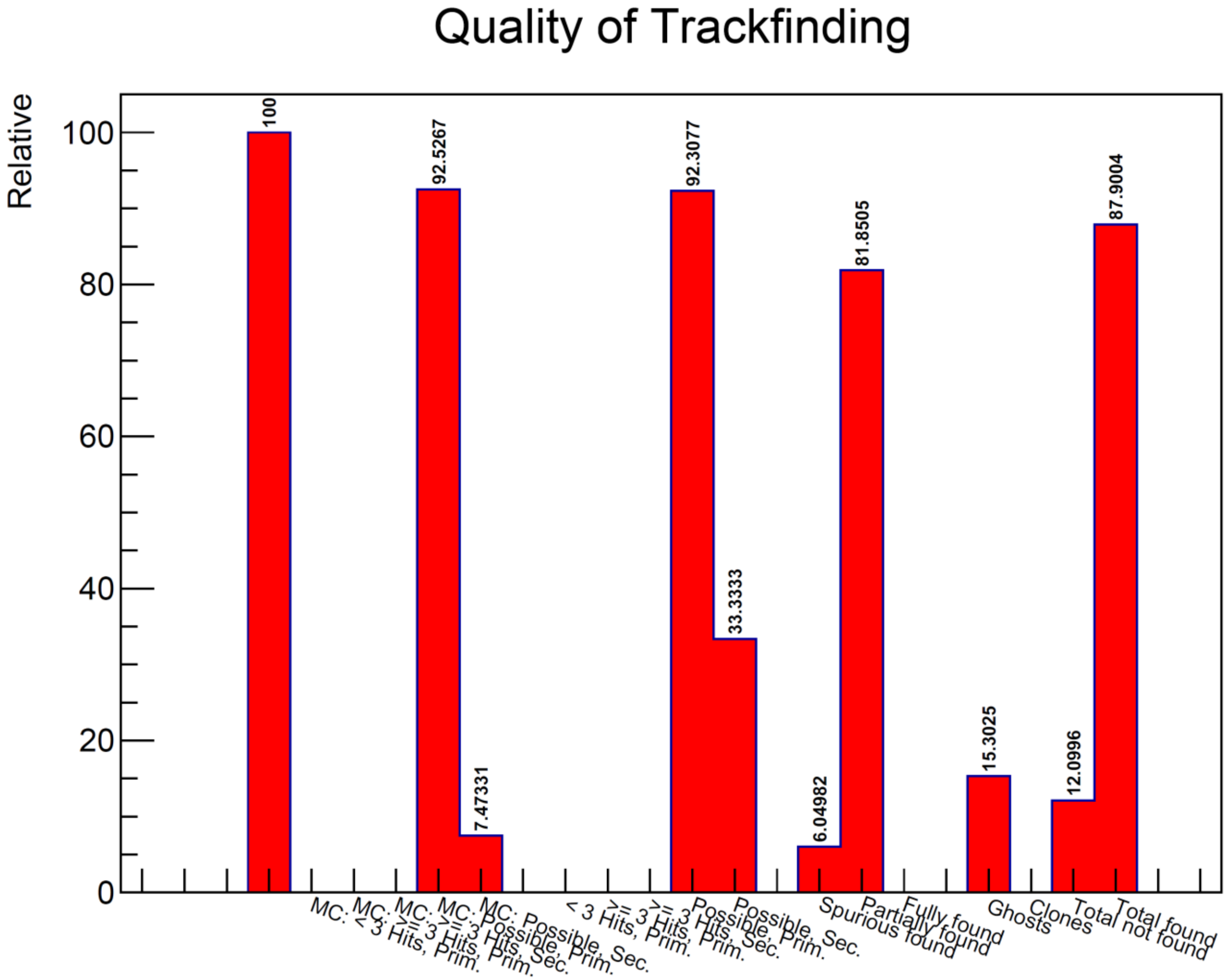
Network Performance:

Box Generator **1-track/event**, **momentum 6 GeV/c**



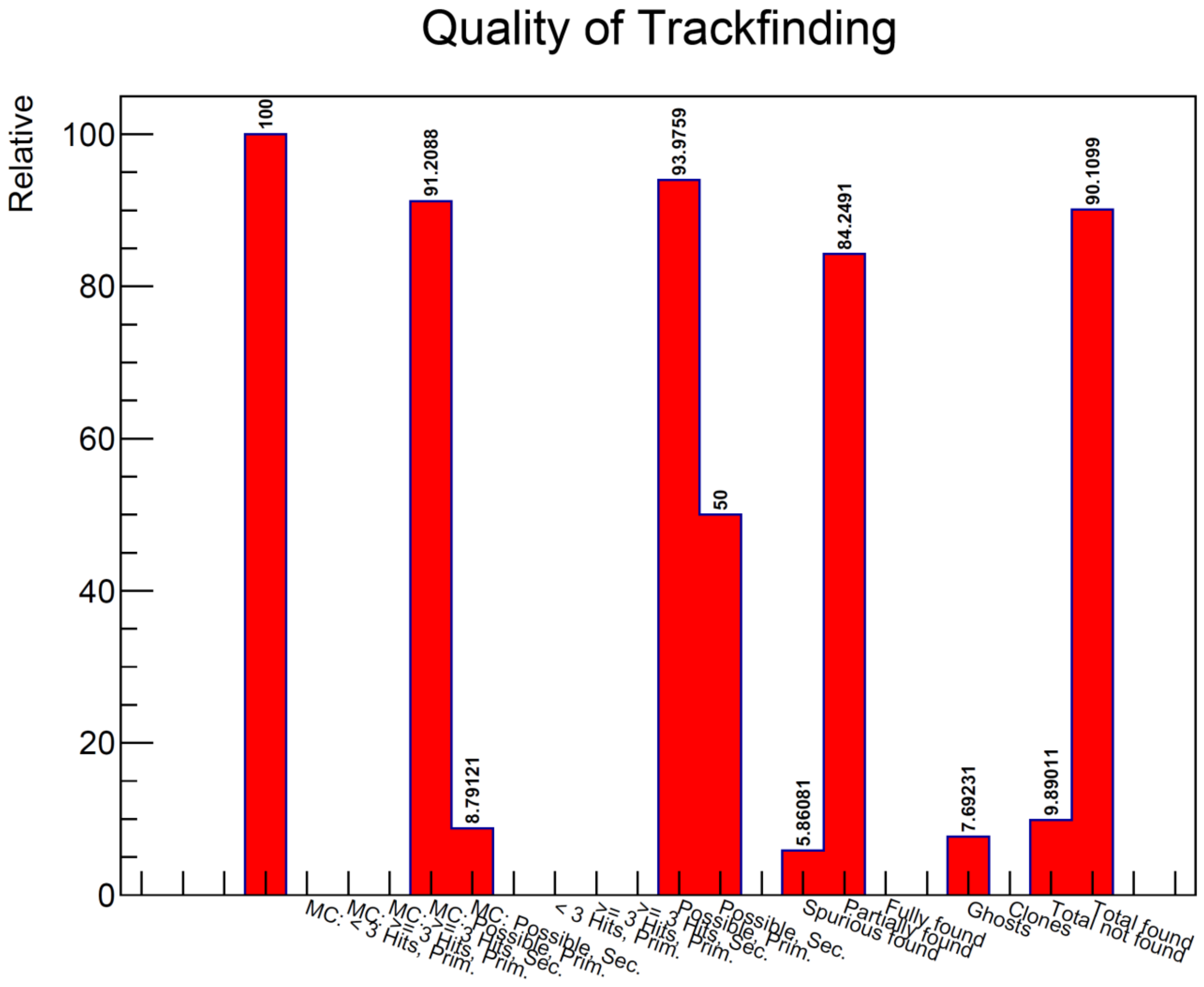
Network Performance:

Box Generator **3-track/event**, **momentum 1 GeV/c**



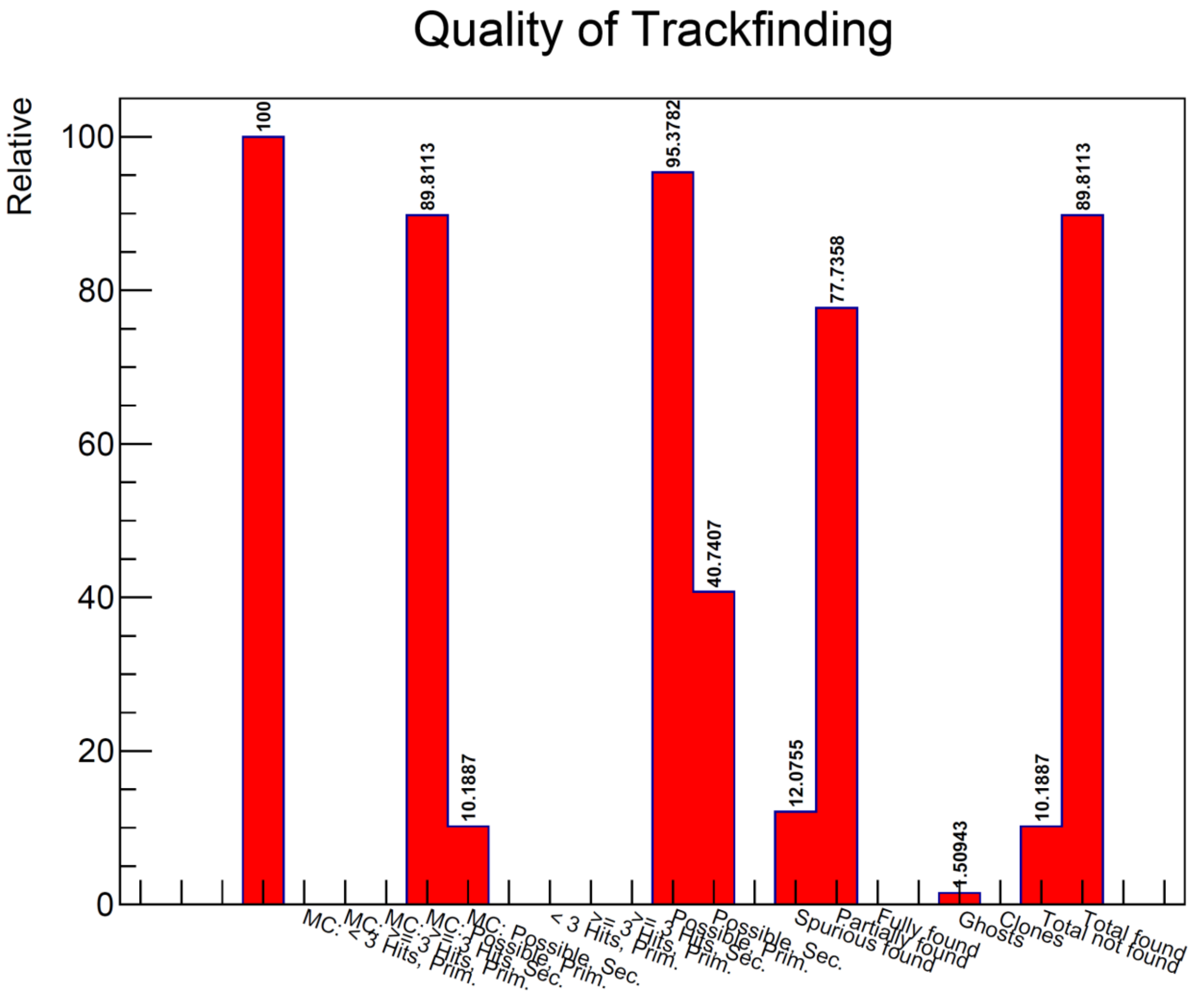
Network Performance:

Box Generator **3-track/event**, **momentum 3 GeV/c**



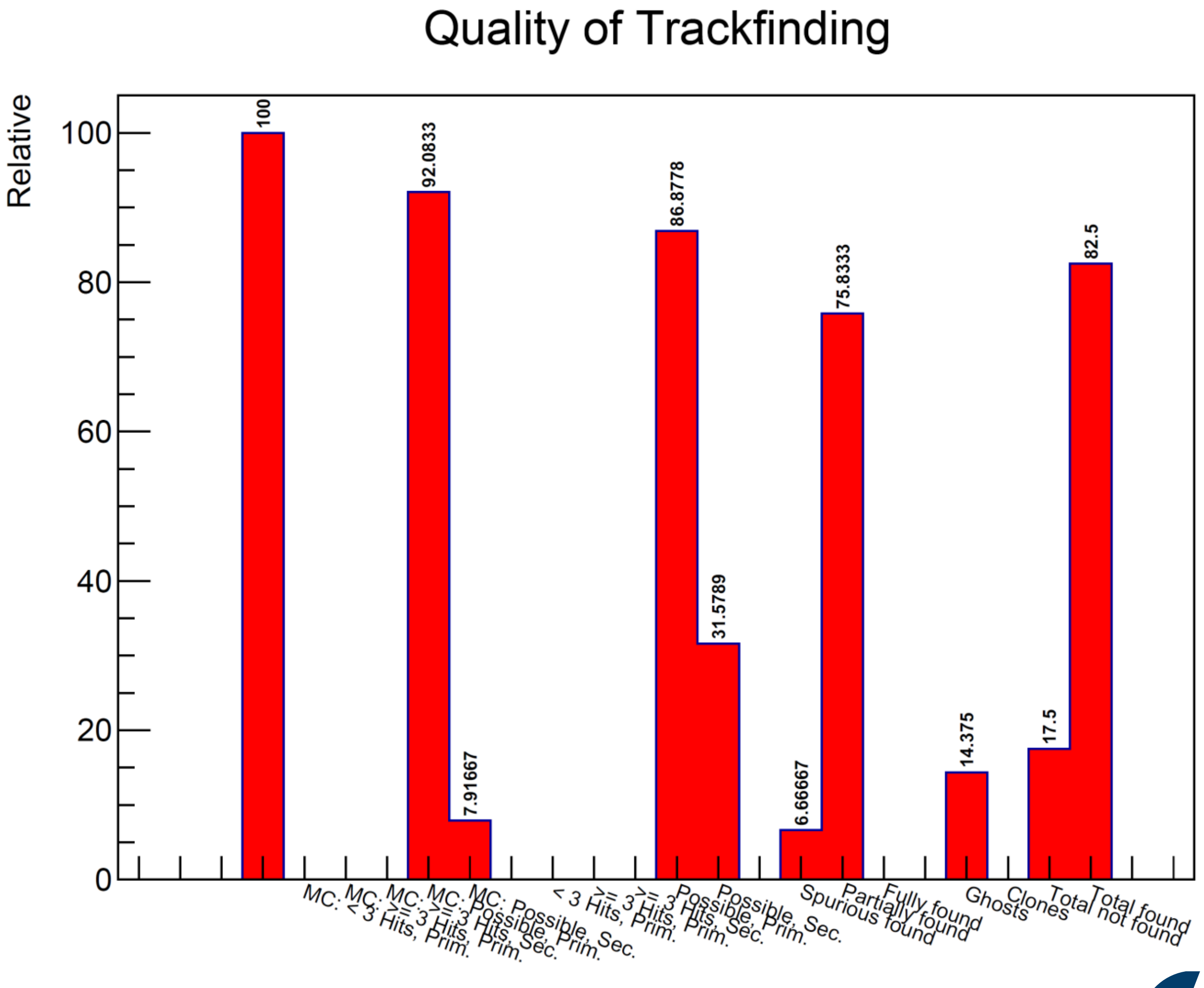
Network Performance:

Box Generator **3-track/event**, **momentum 6 GeV/c**



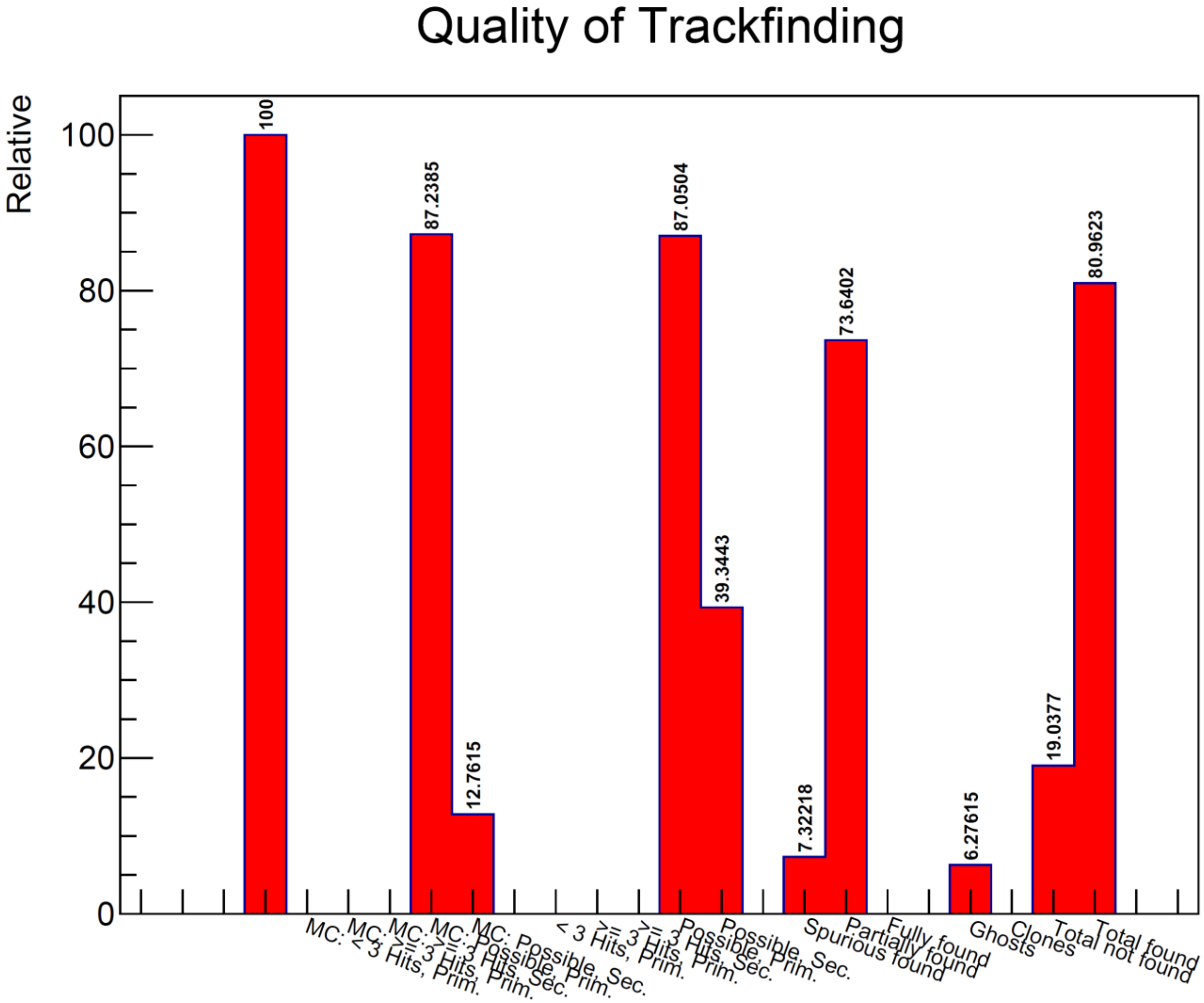
Network Performance:

Box Generator **5-track/event**, **momentum 1 GeV/c**



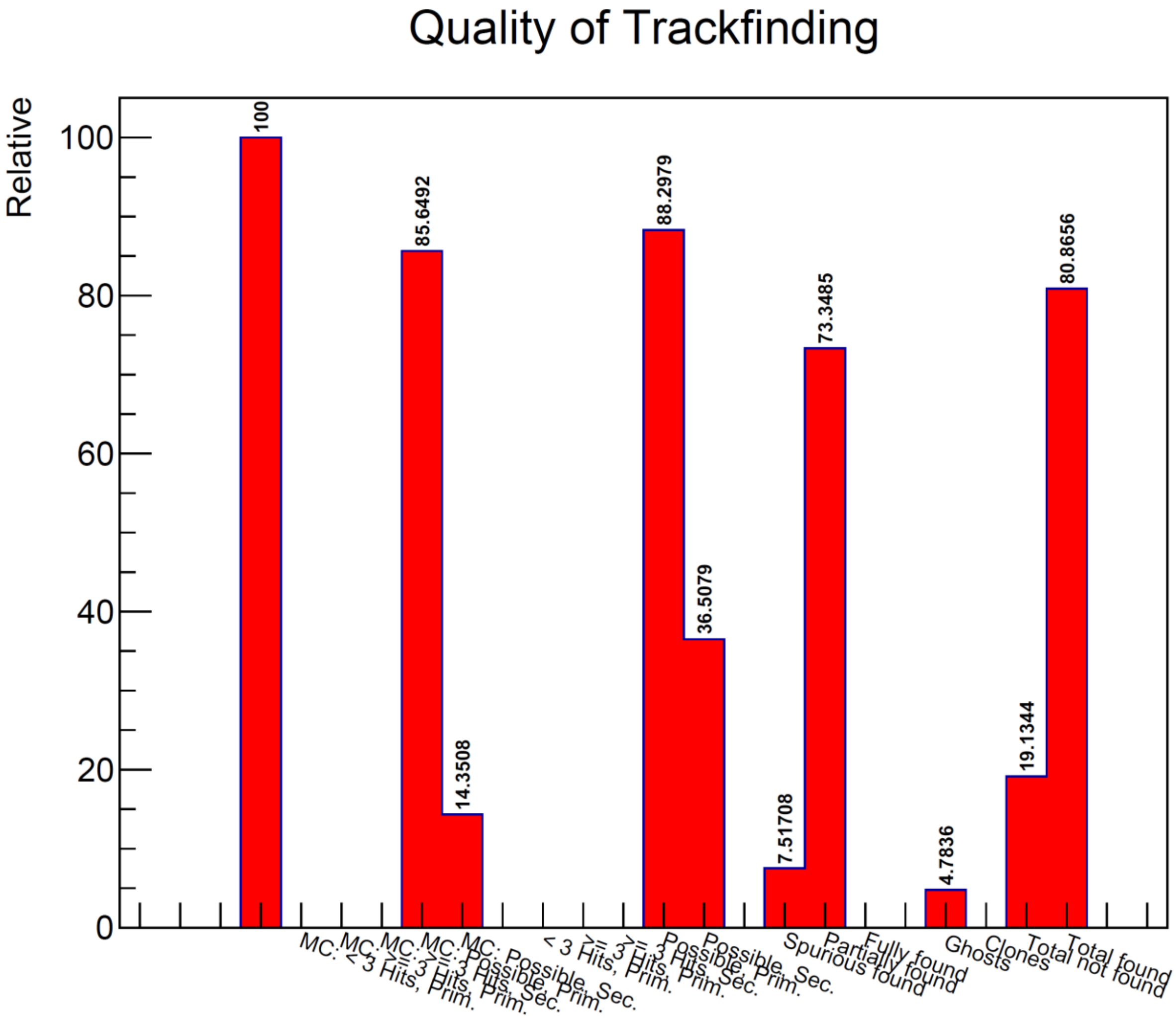
Network Performance:

Box Generator **5-track/event**, **momentum 3 GeV/c**



Network Performance:

Box Generator **5-track/event**, **momentum 6 GeV/c**



Conclusion and outlook:

- Another deep learning approach (global method)
- Accuracy ~99%
- Clustering by just network output is promosing.

- Same method for STT (Adeel Akram)
- Implementation on FPGA (Weijia Wang - KIT)
- Try different architectures (for node embedding)

THANK YOU