Machine Learning For Track Finding at PANDA FTS

Waleed Esmail, Tobias Stockmanns, and James Ritman

On Behalf of the PANDA Collaboration

Institut für Kernphysik (IKP)

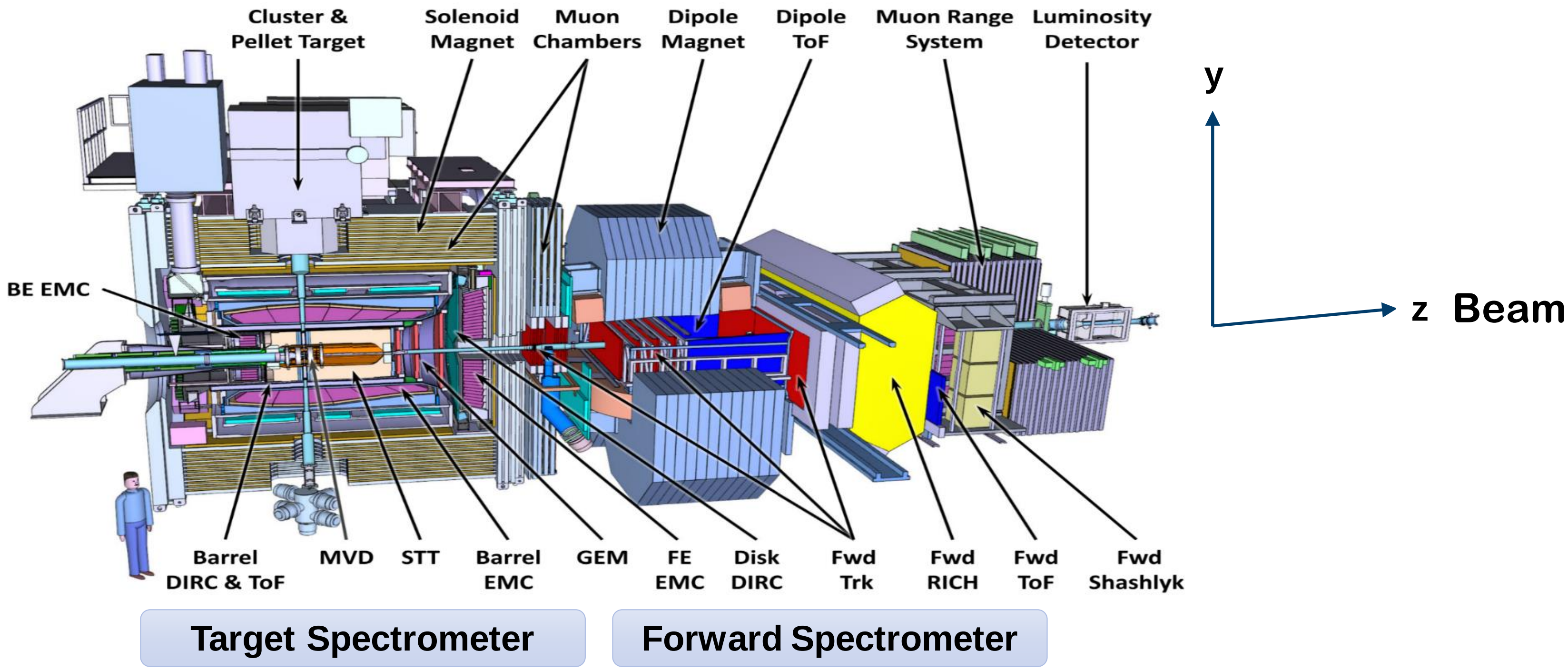
Forschungszentrum Jülich

Outlines:

- PANDA Detector.
- Forward Tracking System.
- Tracking Model.
- Artificial Neural Networks.
- Addition of Skewed Layers.
- Recurrent Neural Networks.
- Conclusion and outlook.

PANDA Detector:

antiProton ANnihillation at DArmstadt



- Hadron Spectroscopy.
- Hadron Structure
- Hypernuclei

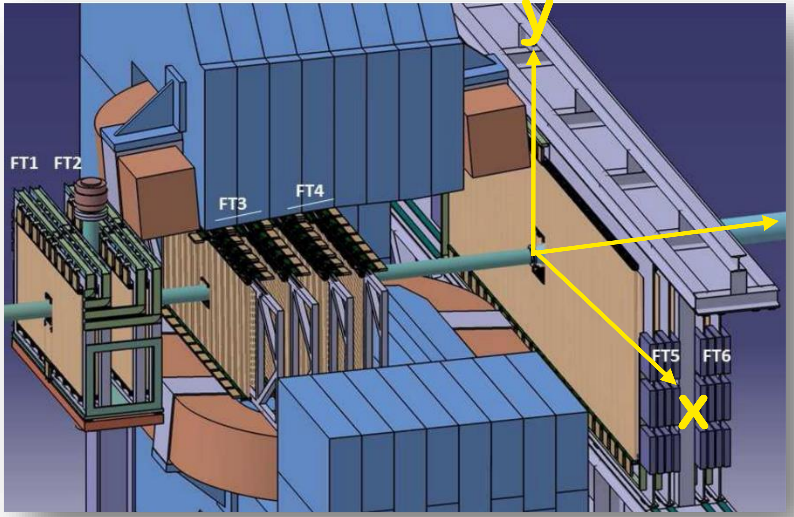
- 4π Acceptance
- Good Tracking
- Good PID

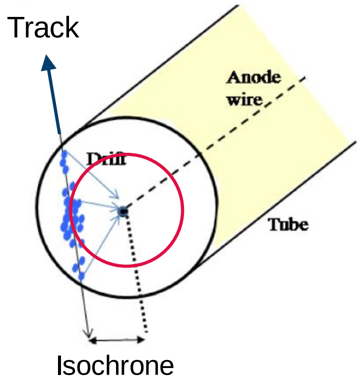
Beam Momentum:
1.5 GeV/c to 15 GeV/c

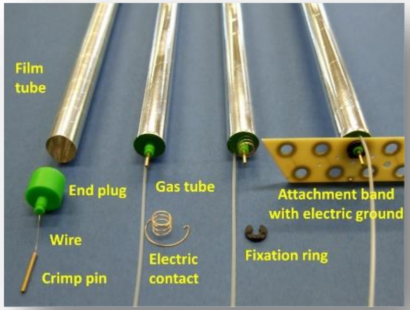
PANDA Detector:

Forward Tracking System (FTS):

- ✓ Straw tubes, same as in the barrel, vertically arranged in double layers
- ✓ 3 stations with 2 chambers each
  - FTS1&2 : Before the Field
  - FTS3&4 : In the Field (2Tm)
  - FTS5&6 : After the Field
- ✓ 8 double layers per chamber.
- ✓ Orientations 0°/+5°/-5°/0° per chamber
- ✓ Tracks are defined by distance of closest approach to the wire (Isochrones)







JÜLICH Forschungszentrum

Tracking Model:

The current approach

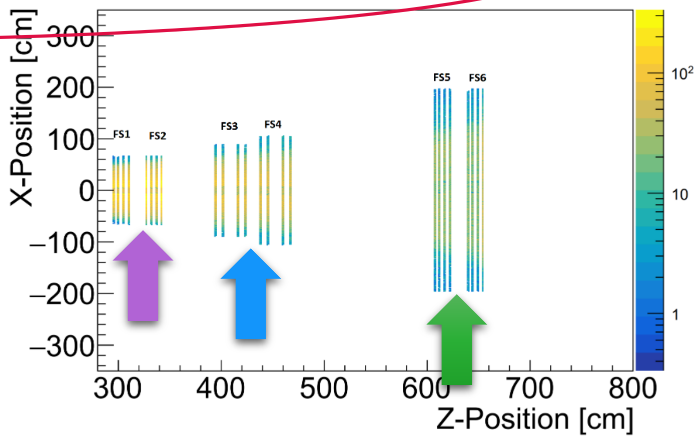
I. **Create Track Segments** by using k-menas.

I. **Create Track Segments** by using a deep neural network.

(FST1+FST2)

(FST3+FST4)

(FST5+FST6)



III. **Interpolate Track Segments** by using a recurrent neural network

	TrackSeg 1	TrackSeg 2	TrackSeg 3
TrackSeg 1			
TrackSeg 2			
TrackSeg 3			

Create Track Segments by using k-menas.

✓ *k-means* algorithm works with unlabeled multidimensional dataset.

✓ *k-means* to group the samples based on their *feature similarities*.

✓ The Algorithm:

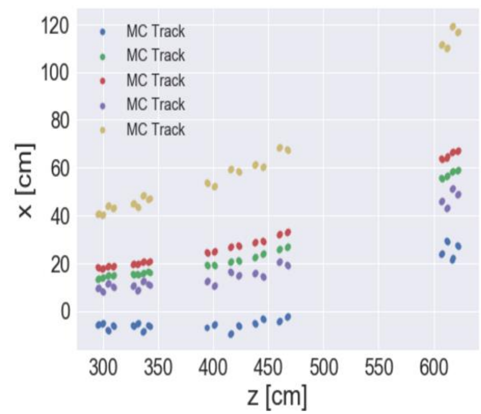
1. Randomly pick *k* clusters (*centroids*) from the sample points as initial cluster centres.
2. Assign each sample to the *nearest* centroid. μ_j: $j \in \{1, \ldots, k\}$.
3. Move the centroids to the centre of the samples that were assigned to it.
4. *Iterations,* Repeat steps 2 and 3 until the cluster assignments do not change or maximum number of iterations is reached.

✓ how do we measure *similarity* between samples?!.

➢ $d(x,y)^2 = \sum_{j=1}^{m} (x_j - y_j)^2$.

➢ Minimization of *cluster inertia*:

Inertia $= \sum_{i=1}^{n} \sum_{j=1}^{k} w^{i,j} \left\| x^i - \mu^j \right\|^2$



How to specify the k centroids:

1. ***Elbow*** method is performance metric for clustering algorithms

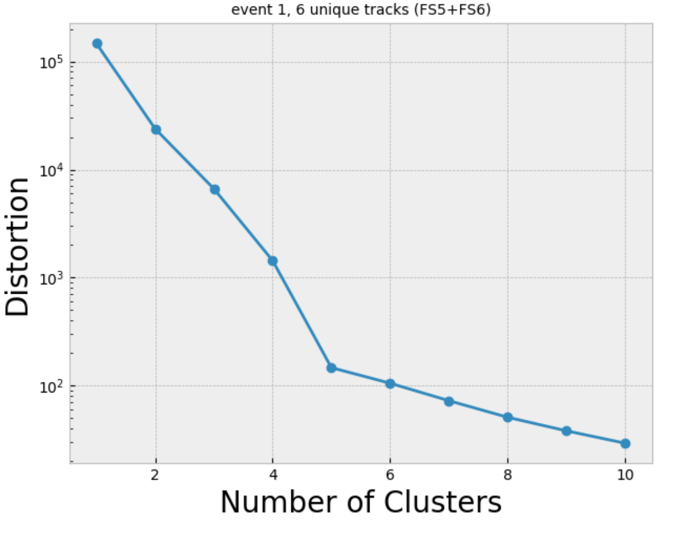
✓ The *idea* behind the elbow method is to identify the value of ***k*** where the *distortion* begins to decrease most rapidly.

2. ***Silhouette analysis*** is a measure of how tightly grouped the samples in the clusters are. This can be done by defining *silhouette coefficient*:

$$s^{(i)} = \frac{b^{(i)} - a^{(i)}}{\max\left\{b^{(i)}, a^{(i)}\right\}}$$

✓ ***a*** is *cluster cohesion*.

✓ and ***b*** is *cluster separation*.



How to specify the k centroids:

1. ***Elbow*** method is performance metric for clust… …ms

✓ The *idea* behind the elbow method is to ide… …of ***k*** where the *distortion* begins to decrease most rapi…

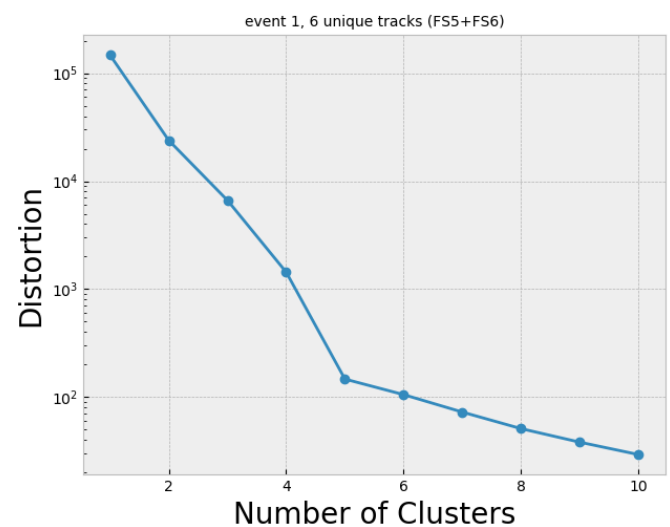
2. ***Silhouette analysis*** is a measure of… …ouped the samples in the clusters are. This can be done by d… …*tte coefficient*:

$$\ldots \{b^{(i)}, a^{(i)}\}$$

✓ ***a*** is *cluster cohes*…

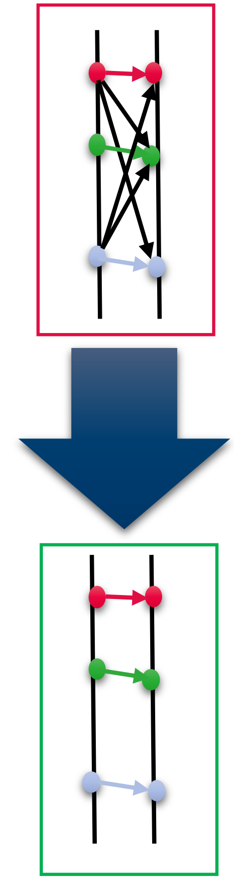
✓ and ***b*** is *clus*… .

Fails in case of 1 track per event



Artificial Neural Networks:

Application to the FTS:



- ✓ Create all possible combinations of hit pairs (adjacent layers).
- ✓ Train the network to predict if hit pairs are on the same track or not.

- ✓ **Input observables:**
  1) Hit pair positions in x-z projection (vertical layers).
  2) Drift radii (Isochrones).
  3) Distance between hits.
- ✓ **Output:**
  1) Probability that hit pair are on the same track.

- ✓ Connect hits that pass the probability cut (threshold).

  e.g. probability(h_1-h_2)> threshold, and
  probability(h_2-h_3)> threshold, so
  h_1, h_2, h_3 are on the *same track.*

Artificial Neural Networks:

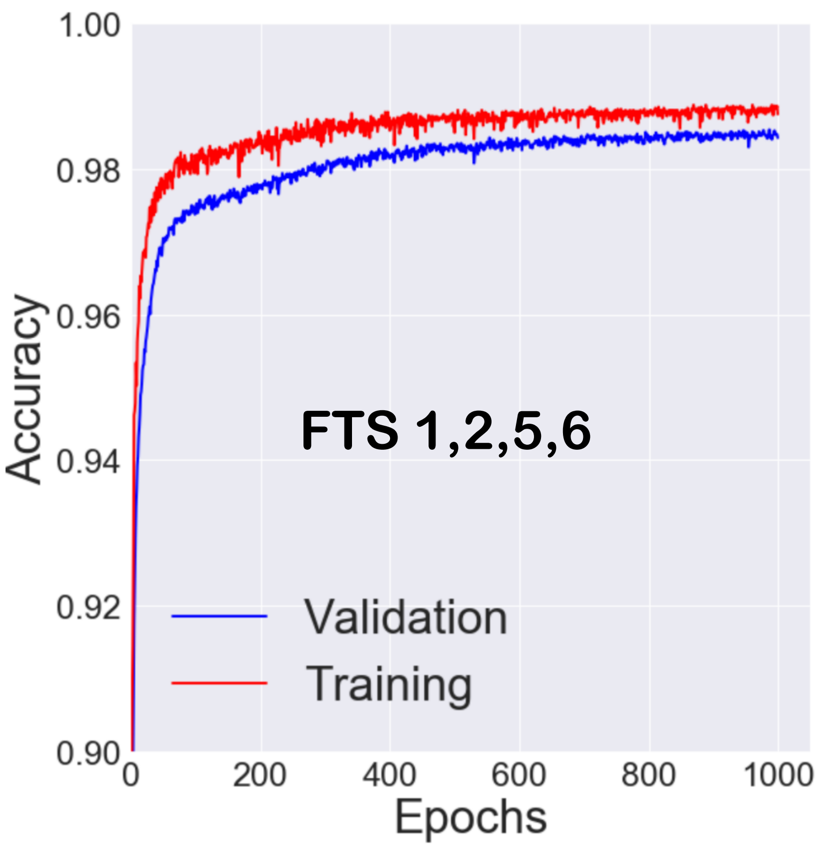
Application to the FTS:

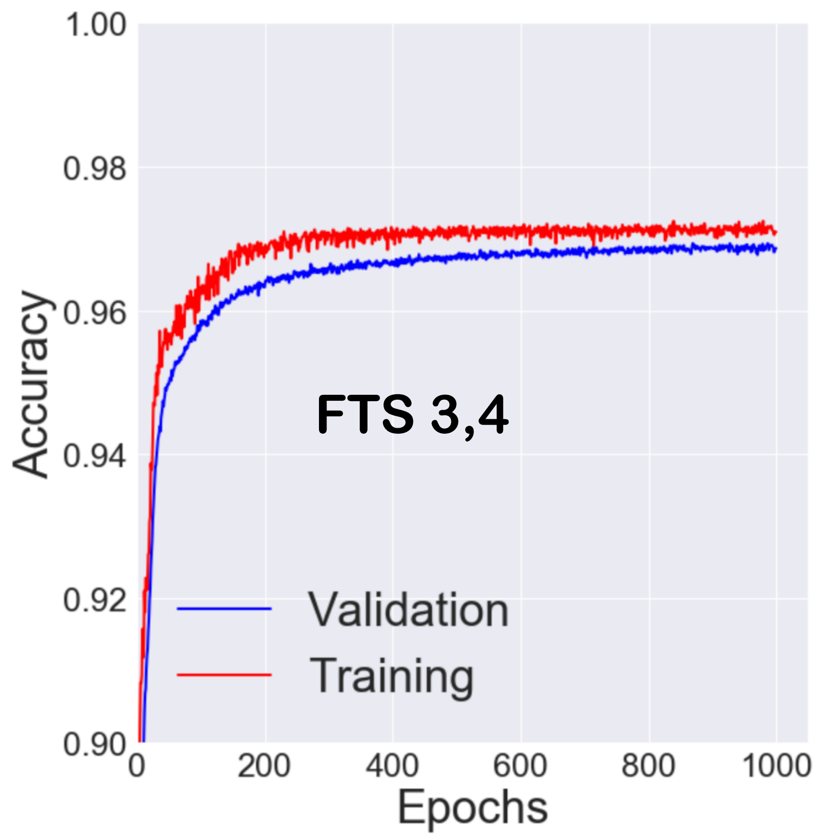
✓ **Network Architecture**:
- 5 hidden layers (400, 300, 200, 100, 50)
- Drop-out layers with 50%
- Tanh activation
- Last layer "Sigmoid" activation

✓ **Training data**: Particle gun
- Momentum Range 1 - 10 GeV/c
- Polar Angle $0.5^{o} – 10^{o}$
  - 5μ per event

Artificial Neural Networks:

Network Accuracy:

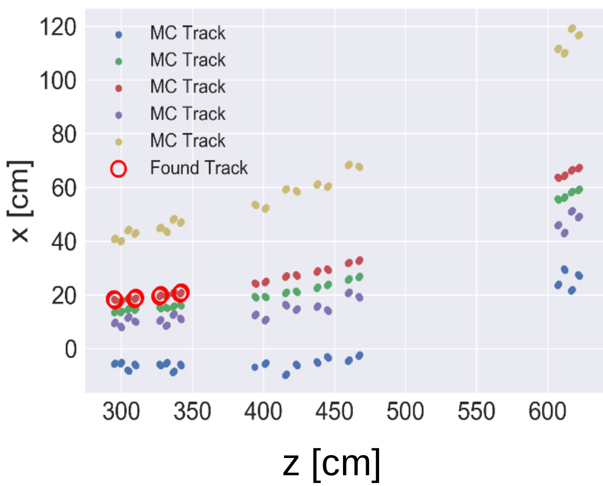




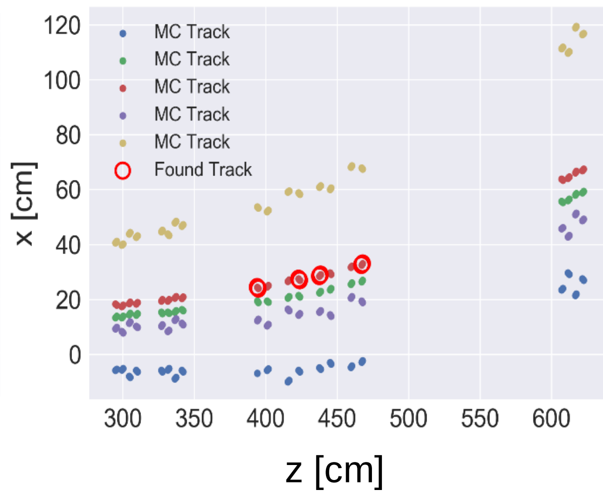
Artificial Neural Networks:

Pictorial Representation (example found track):

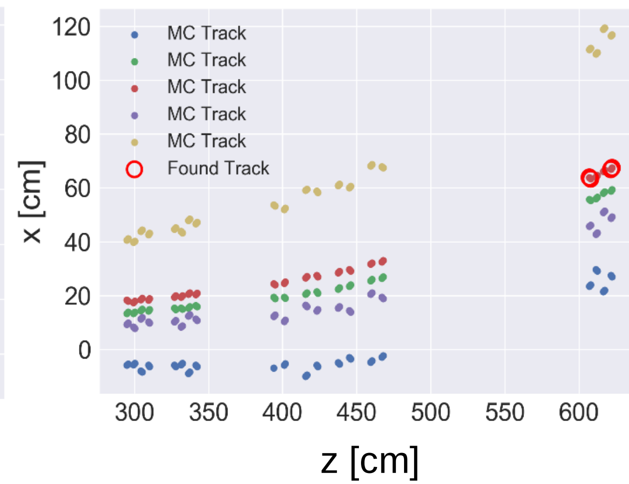
FTS 1,2



FTS 3,4



FTS 5,6



JÜLICH Forschungszentrum

Artificial Neural Networks:

Some Results:

- ✓ Data are generated by particle gun
- ✓ 1000 events (5 muons per event).
  - ✓ Criteria:

1. If found track has less than 4 hits, do not count the track.
2. Find the matching particle (the matching particle is the one to which the majority of the tracks points belong).
3. Calculate purity: $(n_{correct}/n_{all})$ if purity>0.8 count reconstructed track.

	FTS 1 & FTS 2	FTS 3 & FTS 4	FTS 5 & FTS 6
Purity	0.99	0.99	0.99
Efficiency	0.96	0.95	0.96

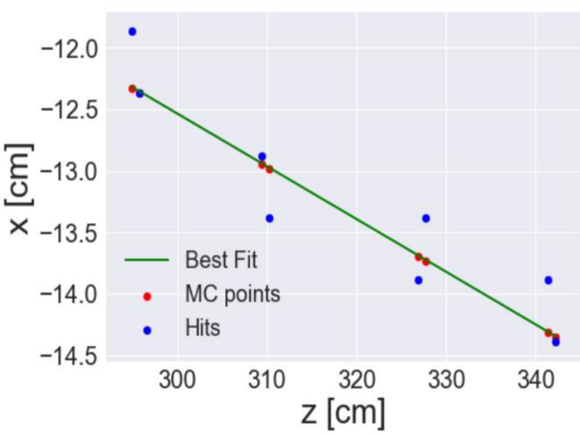
Addition of Skewed Layers:

✓ **The y-z track motion is extracted from the skewed layers.**

✓ **Thus x-z projection candidates are used as "seed" for such task.**

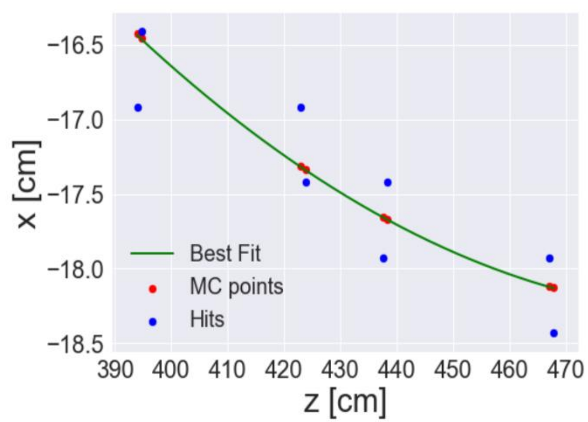
1) Fitting (Orthogonal Distance Regression).

FTS 1,2



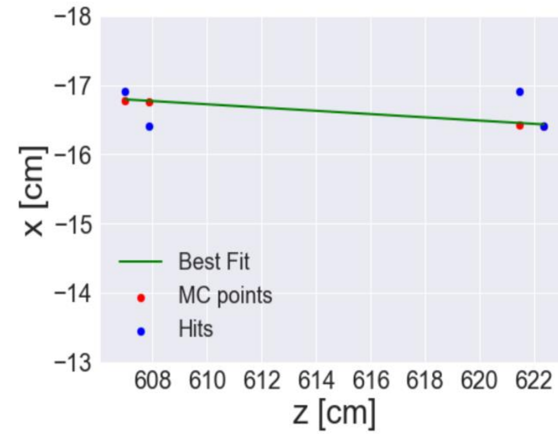
Linear

FTS 3,4



Parabolic

FTS 5,6



Linear

Addition of Skewed Layers:

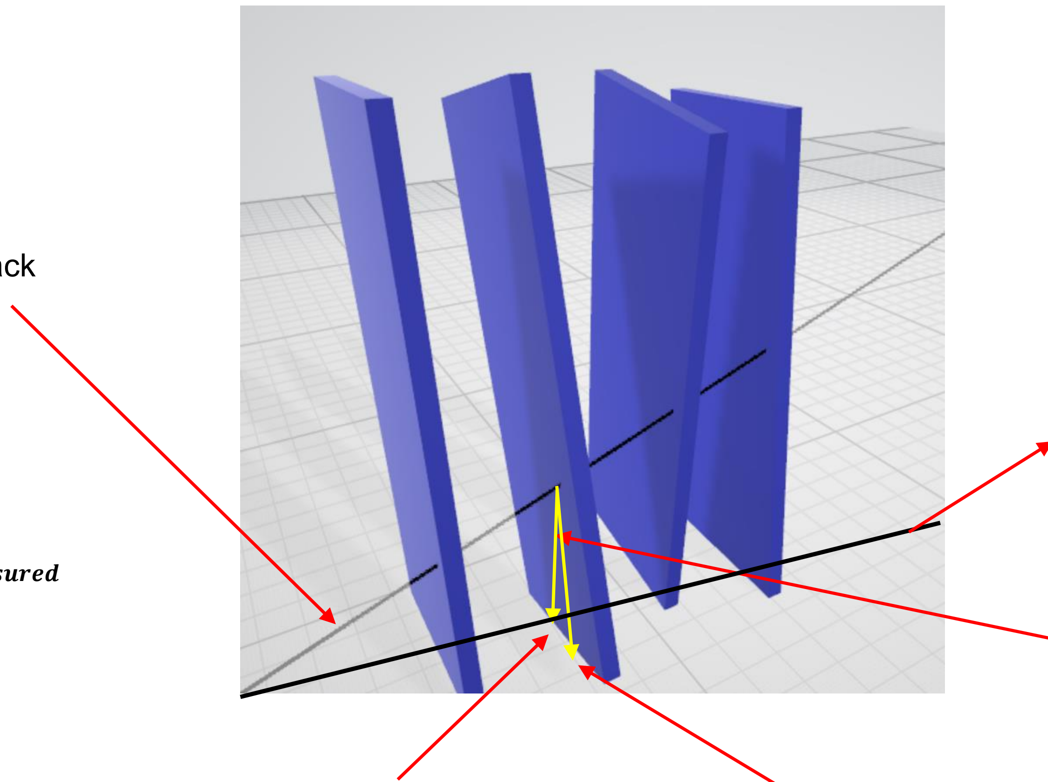
2) Using the fit **predict** the **true x position** of the **skewed layers**.

3) The **distance** between the skewed layers measurements and the predicted x position allows to identify a **y measurement**.

4) Collect all hits that have the *same* slope (y/z).

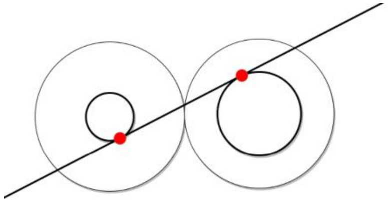
$$\Delta y = \frac{\Delta x}{\tan\theta}$$

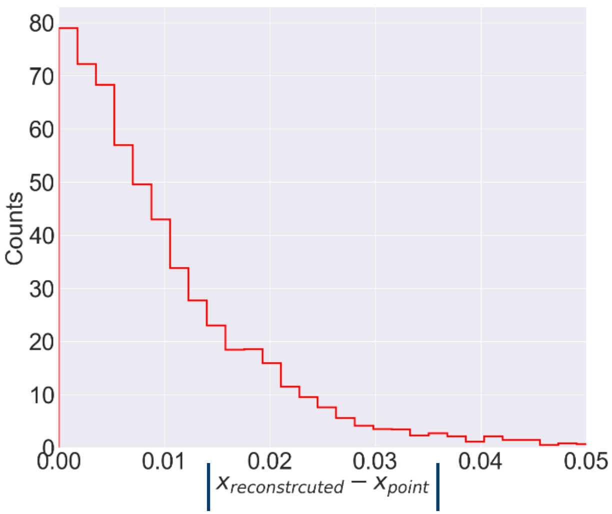
$$\Delta x = x_{pred} - x_{measured}$$

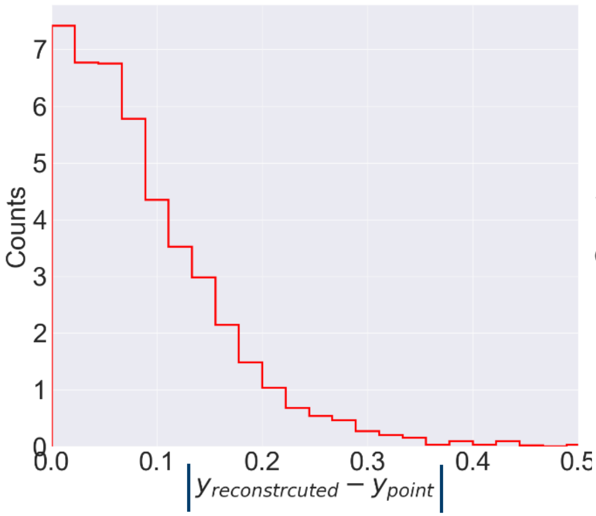


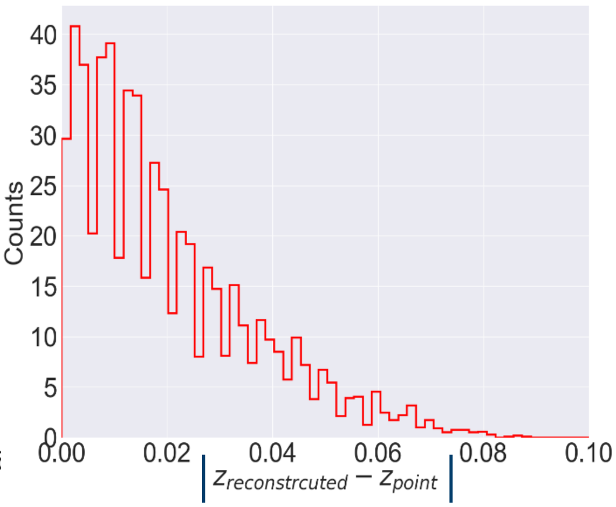
Addition of Skewed Layers:

- ✓ The fitting provides the correct hit positions (tangent to isochrones).
- ✓ Linear fitting in y-z plane.





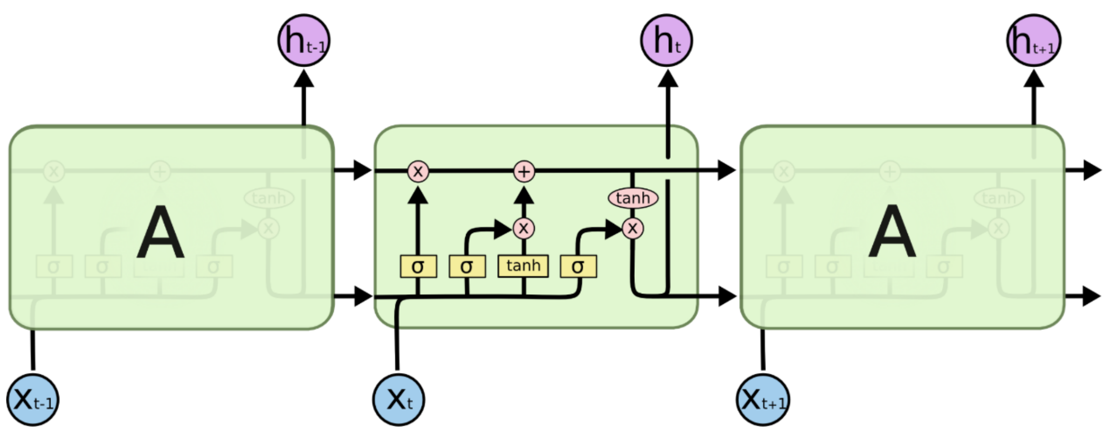




Recurrent Neural Networks (RNN):

Long Short-Term Memory (LSTM):

- ✓ **LSTMs are a special kind of RNN, capable of learning long-term dependencies.**
- ✓ **LSTMs also have the same chain like structure as simple RNN, but the repeating module has a different structure. Instead of a simple neuron, it is a cell.**
- ✓ **The key to LSTMs is the cell state.**



Credit: http://colah.github.io/posts/2015-08-Understanding-LSTMs

Recurrent Neural Networks (RNN):

Long Short-Term Memory (LSTM):

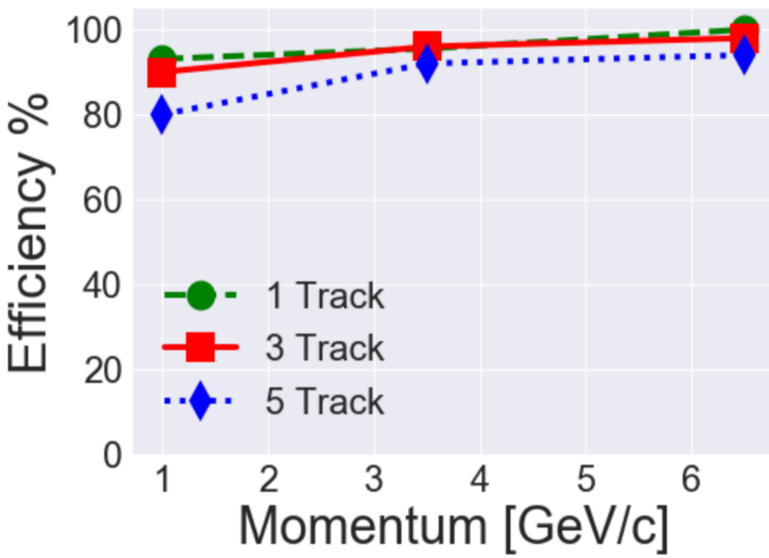
✓ All possible combinations of track segments in FTS(1,2), FTS(3,4), FTS(5,6).

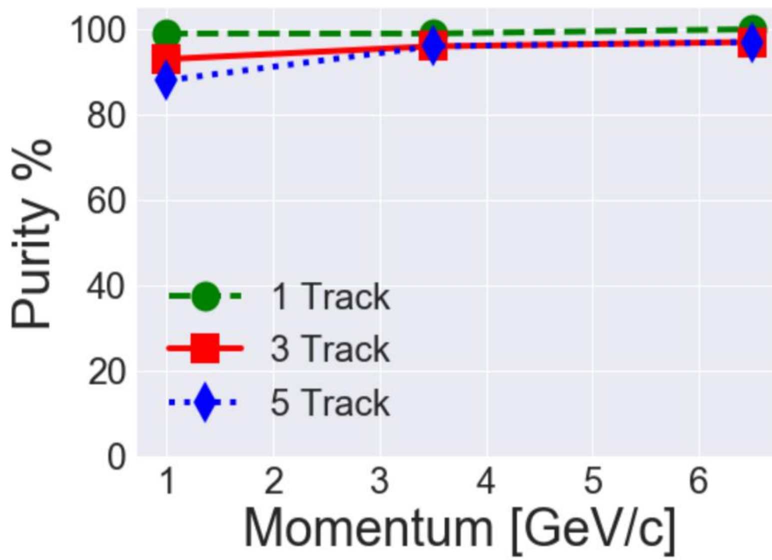
✓ **Input observables:**
1) Corrected hit positions (x,y,z).

✓ **Network Architecture**:
- 3 hidden layers (Bidirectional LSTM) (300, 200,100)
- Drop-out layer with 50%.
- Last layer “Sigmoid” activation.

Recurrent Neural Networks (RNN):

Results:





Conclusion and outlook:

✓Proof of concept implementation.

✓Results from ANN as well as RNN are promising.

✓Deep learning methods are in principle applicable to tracking finding problem, at least in PANDA FTS.

✓Test on real data.

✓Track Fitting (RNN?!).

✓Move the implementation to PandaRoot (C++).

Thank you for your Attention