The LOOT Model for Primary Vertex Finding

Tracking and vertexing

Event reconstruction

Event reconstruction plays a key role in data processing for High Energy Physics experiments. It consists of two stages: recognizing the tracks and then finding the vertices.

The classical HEP algorithms for track and vertex reconstruction are based on the Kalman Filter (**KF**) method, since KF makes it easy to take into account the inhomogeneity of the magnetic field, multiple scattering and energy losses.

KF is used sequentially station by station for track recognizing and fitting and then also sequentially for vertex finding.

However, in order to start KF needs a very time consuming preliminary search of the initial set of parameters (so called seeding).

Besides, KF suffers from exponentially growing computational complexity and lack scalability while increasing the event multiplicity

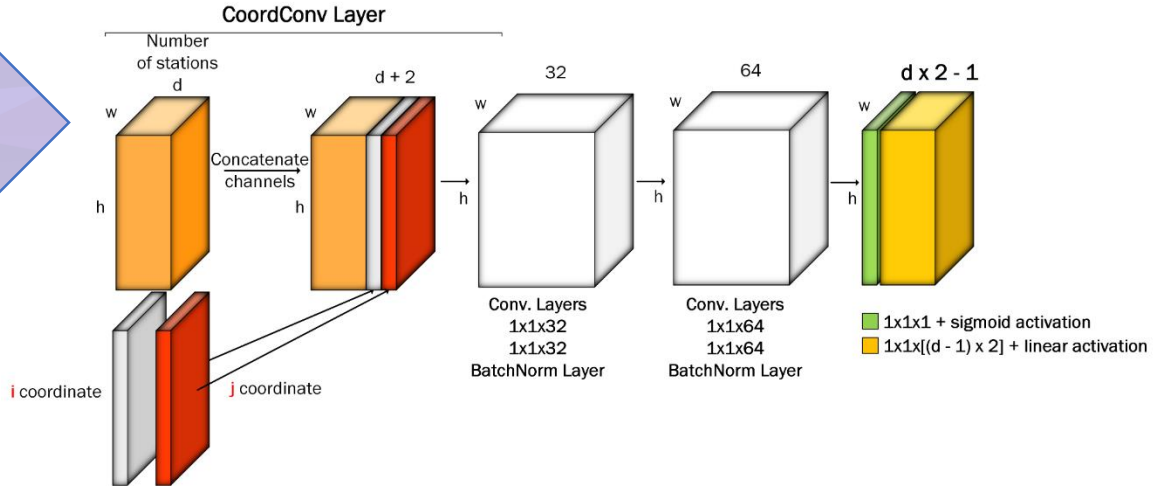
Despite the KF success and many tricks to reduce the seeding time, this method still has several disadvantages caused just by its locality, when tracks are reconstructed one by one. Local approaches have an obvious drawback: they do not allow access to the global picture of an event and see the dependence between individual tracks or groups of tracks.

At the same time, there is another global approach, in which the recognition of the entire event including all tracks and vertex itself among noises is performed immediately across the whole picture of this event.

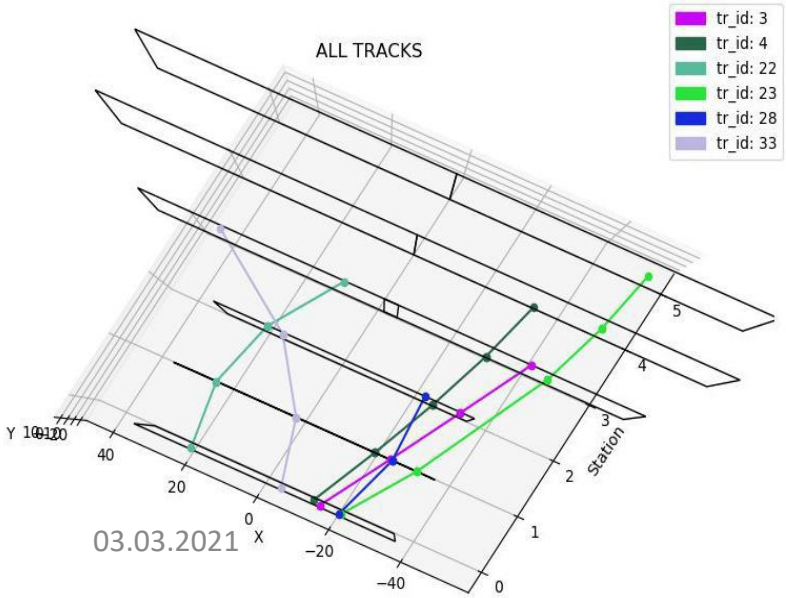
How to extend a convolutional neural network to represent a physical event

Using Look Once On Tracks (LOOT) model

Our main idea is to use OZ dimension instead of RGB channels – it's a radically new approach. Height and Width are the sizes of the largest station (most often the last).



See Goncharov et al http://ceur-ws.org/Vol-2507/130-134-paper-22.pdf



Event - image. Stations - color

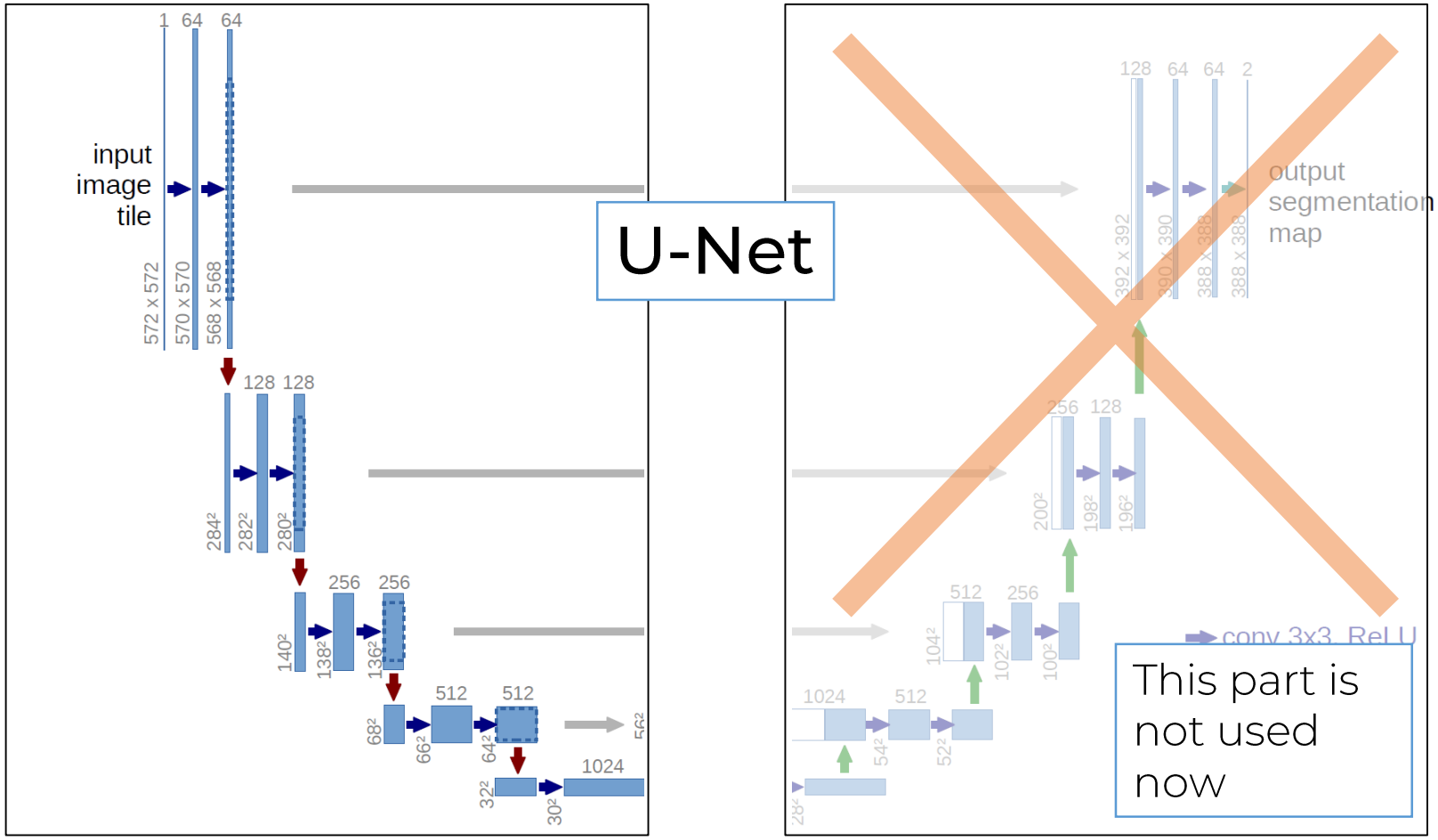
Images have 3D format: Height+Width+RGB

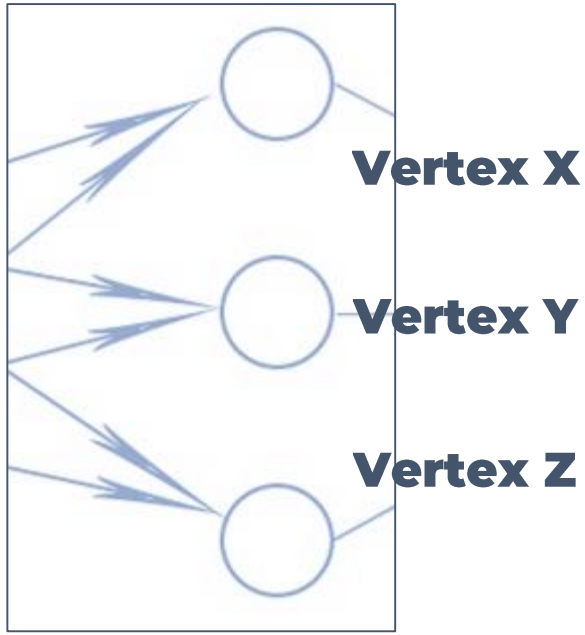
- ✓ Data from each station is a **sparse matrix of zeros and ones**, where ones indicate hits appearance
- ✓ Events have 3D format too: Height+Width+Stations

LOOT + U-net architecture for vertex prediction

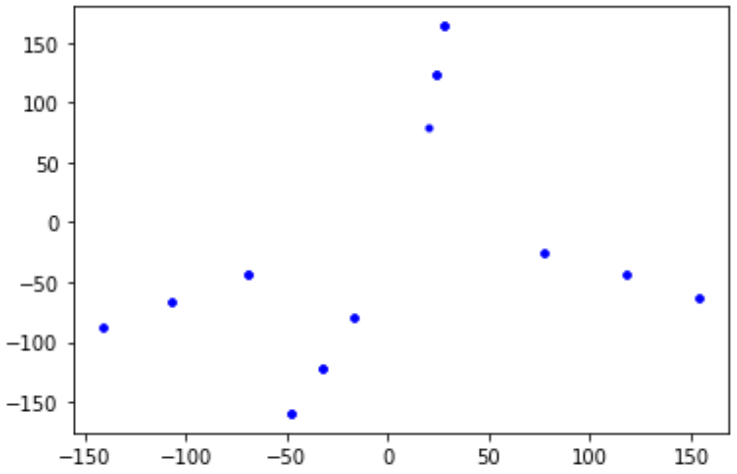
U-Net is a convolutional neural network that was developed for biomedical image segmentation.

Network consists of a contracting path and an expansive path, which gives it the u-shaped architecture.



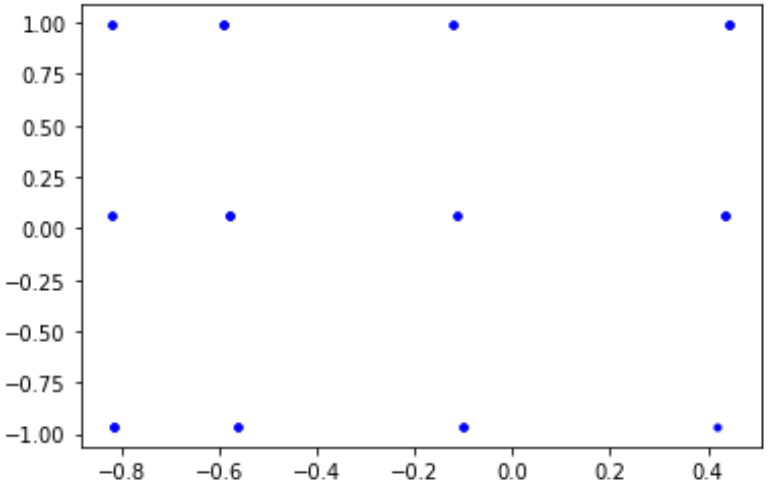


Data preprocessing



Translation from rectangular to cylindrical coordinates

$$Z = \sqrt{x^2 + y^2},$$
$$X = arctg\left(\frac{y}{x}\right)$$
$$Y = z$$



Data normalization

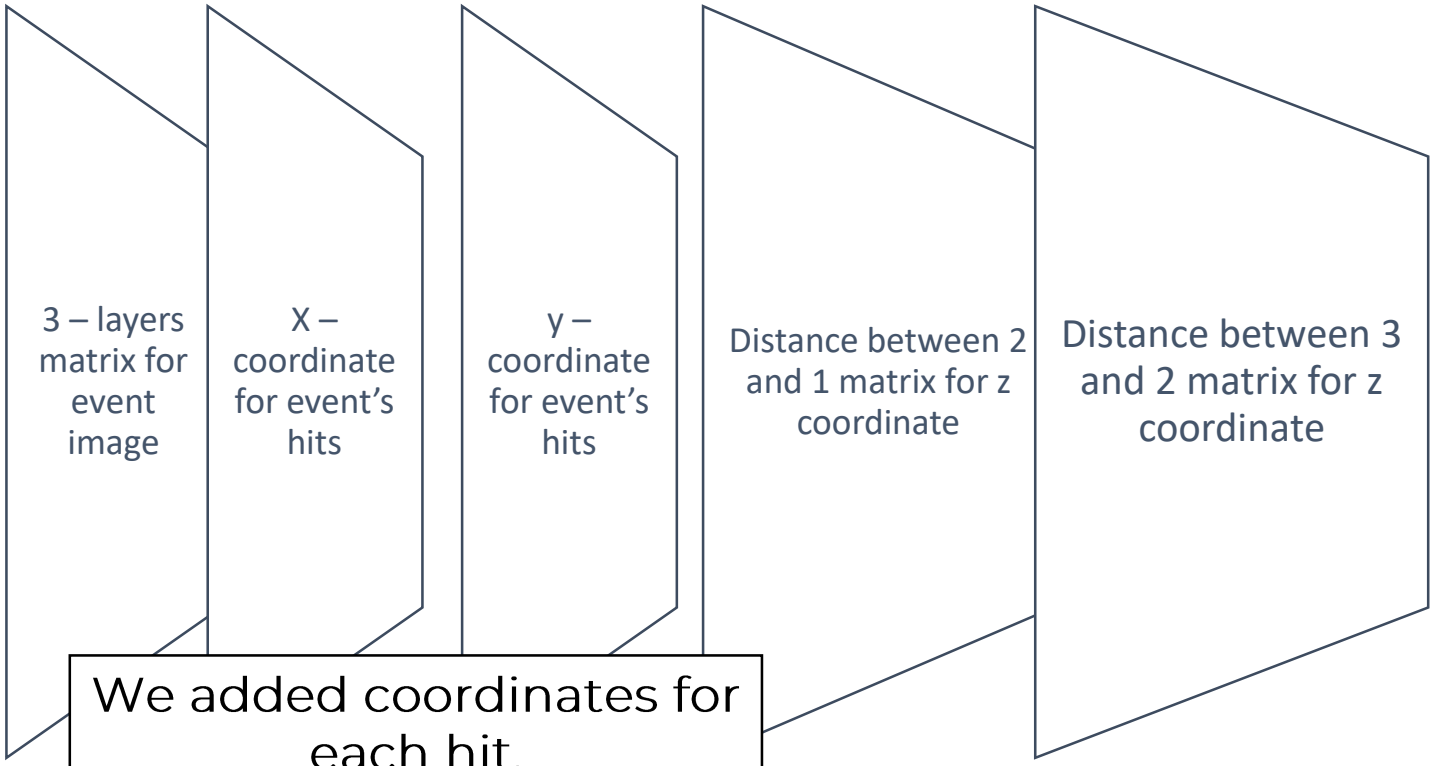
0	1	1	0	1	0
1	0	1	0	0	0
1	1	0	1	0	0
0	0	1	0	0	
1	0	0	0	0	

256

256

Events to binary matrix

Data preprocessing



We added coordinates for each hit.
For both X and Y, the matrix is 256x256 representing the spatial map of the coordinates grid.

For the Z coordinate, there must be a two-dimensional tensor, where the third dimension is two matrices with the size of 256x256.

Training model

Model inputs

3 – layers matrix for event image

X – coordinate for event's hits

y – coordinate for event's hits

Distance between 2 and 1 matrix for z coordinate

Distance between 3 and 2 matrix for z coordinate

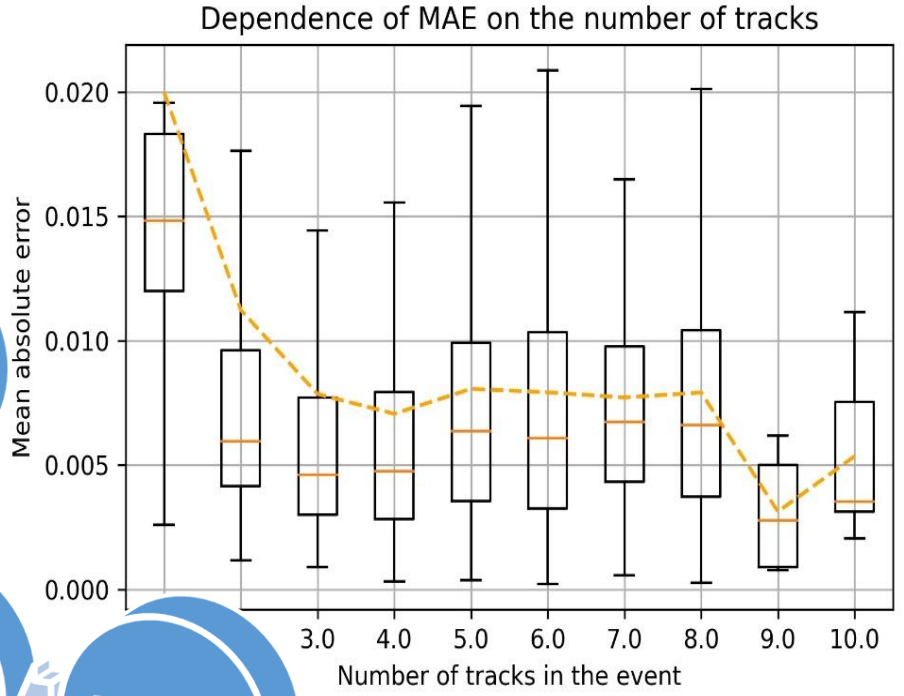
20000 – events for training

Optimizer Adam

Model

Mean absolute error for loss-function

5000 – events for validation



Predicted vertices with mean absolute error = 0,009

Ariadne: PyTorch Library for Particle Track Reconstruction Using Deep Learning

Ariadne – the first library for deep learning tracking on Python:

- ✓ any type of event data including collider and fixed-target experiments
- ✓ metrics logging, multiprocessing for data preparation, multi-GPU training
- ✓ open source and fully deterministic (https://github.com/t3hseus/ariadne)

2 models of neural networks for tracking have already been successfully trained by Ariadne.
Loot will also be added to this library.