Improvements of the LOOT model for primary vertex finding based on the analysis of development results

Государственный университет «Дубна»

Ekaterina Rezvaya, Pavel Goncharov, Yury Nefedov,
Gennady Ososkov, Alexey Zhemchugov

ОИЯИ

Distributed Computing and Grid-technologies in
Science and Education
July 06, 2021

Tracking and vertexing

The classical HEP algorithms for track and vertex reconstruction are based on the Kalman Filter (KF) method.

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.

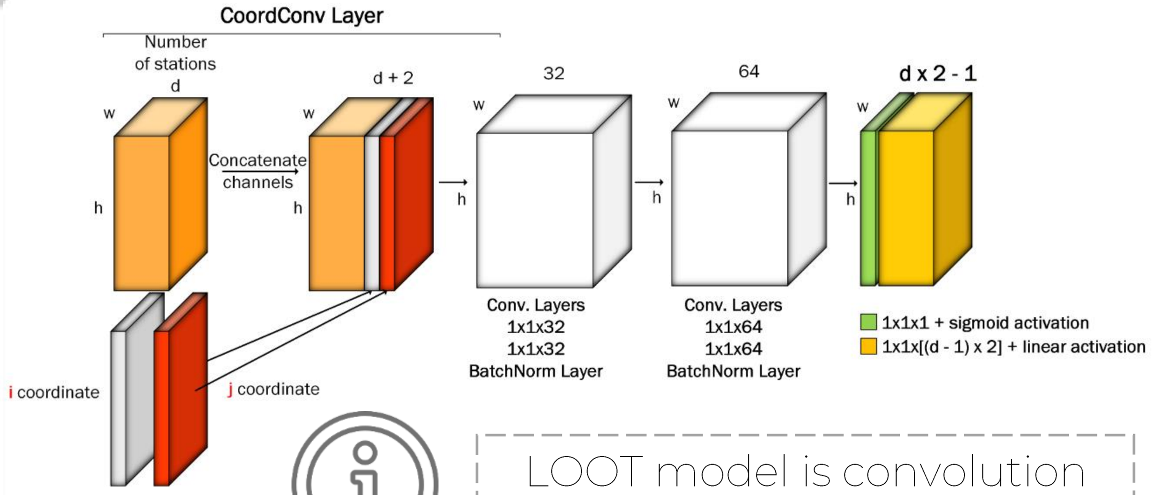
At the same time, there is another global approach, in which the recognition of the entire event including all tracks and vertex itself among noises is performed immediately across the whole picture of this event.

Knowledge about primary vertex

Leads to the significant reduction of the algorithmic complexity during the track-candidate search – from O(n^2) to O(n), and can improve the overall track reconstruction efficiency.

Helps to improve the precision in determining the particle momentum.

Look Once On Tracks (LOOT)

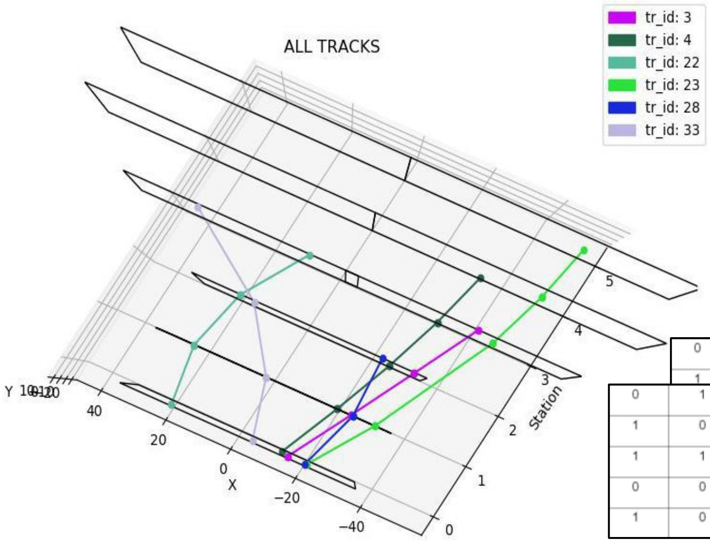


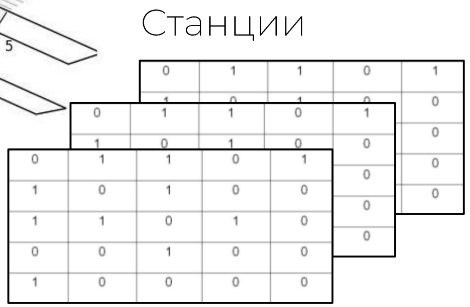
LOOT model is convolution neural network (CNN)

How to use CNN to present a physical event?

The main idea of the LOOT is to use the OZ dimension instead of RGB channels

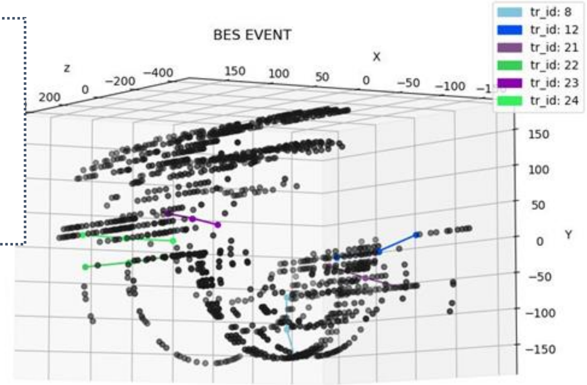
=





First results and problems

We used 200 000 Monte Carlo simulated events for the BesIII experiment.
Most of the hits are fake.



Initial data

- Cartesian coordinate system
- Not normalized

Transform to cylindrical and normalize

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

Train model

MAE = 0.009

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |x_i - y_i|^2$$

problematic to get the distance in the required units

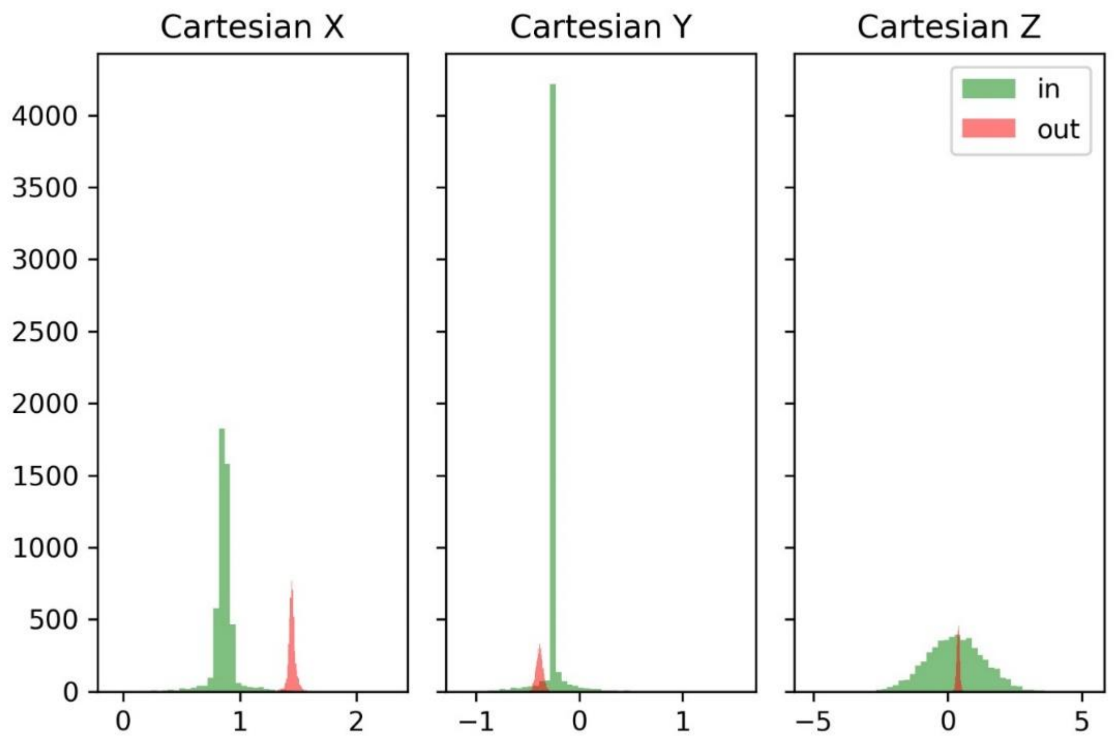
Criteria for evaluating results:
Mean Absolute Error = MAE
Mean Square Error = MSE

The problem is that the error was calculated for normalized data, and it turned out to be quite problematic to get the distance in the required units.

1 step: metrics and inverse transformation

To get the value of the metric in initial units, it is necessary that the metric accepts data in the same units as input.

Therefore, to calculate MAE in cm, an inverse transformation was added when calculating metrics.

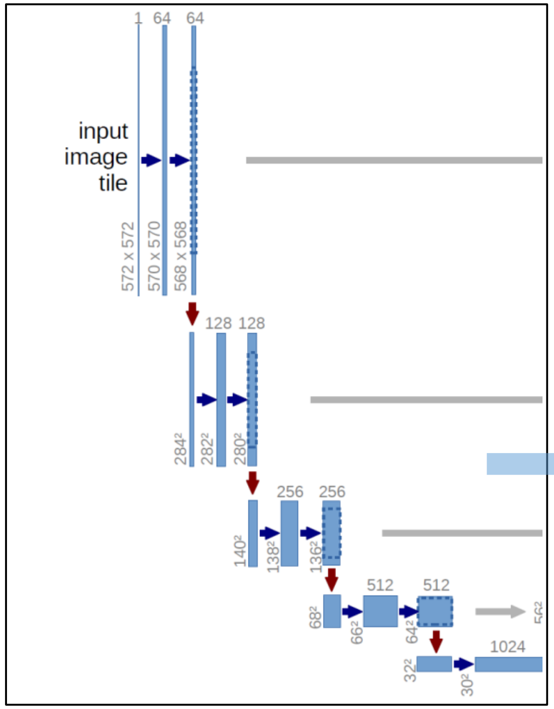


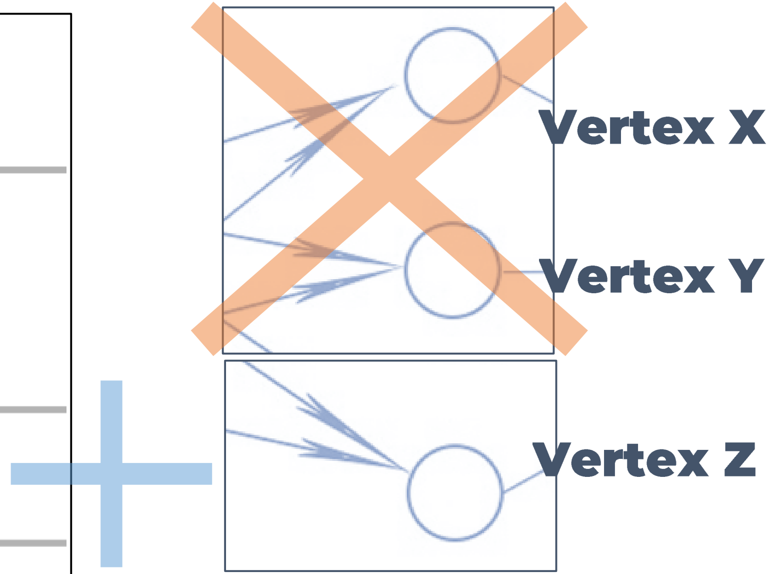
The MAE after recalculating the metric is 0.4 cm

This graph shows that the trained model is not working correctly!

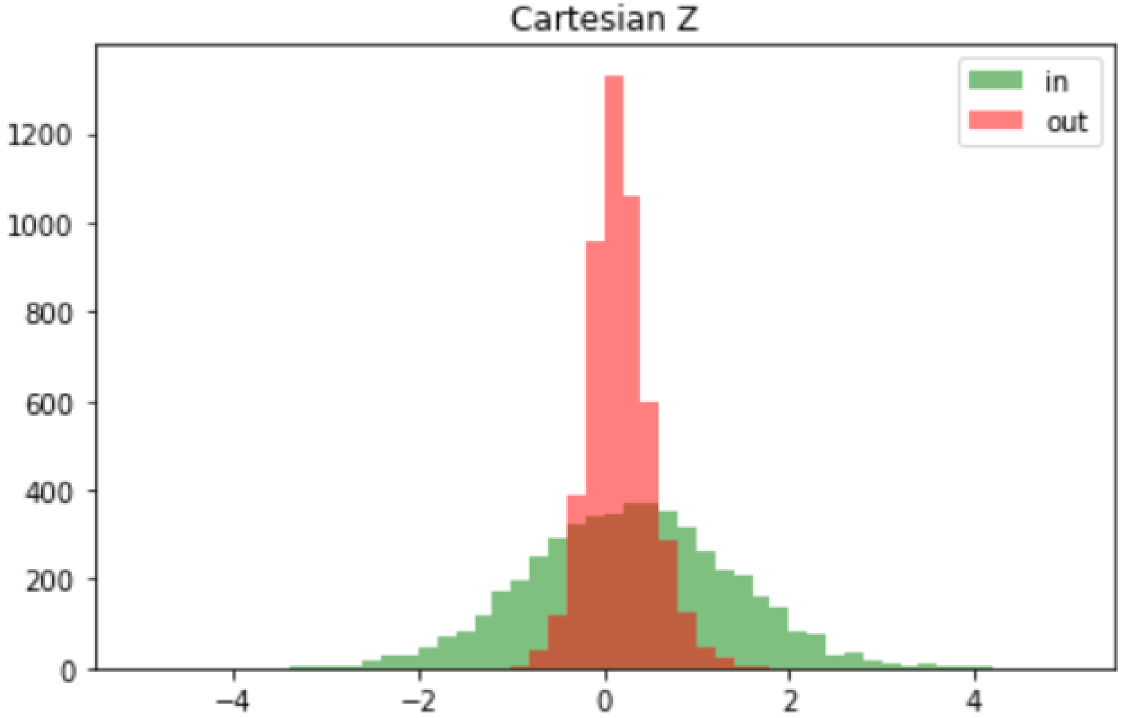
2 step: fixed X and Y coordinates value

It is worth noting that due to the structural features of the BESIII collider, X and Y coordinates are well known.
Therefore, they can be fixed at the average value and make sure that the network finds Z more accurately.





LOOT model can predicted only Z coordinate



However, shortening the coordinates for prediction didn't affect the quality of the result.

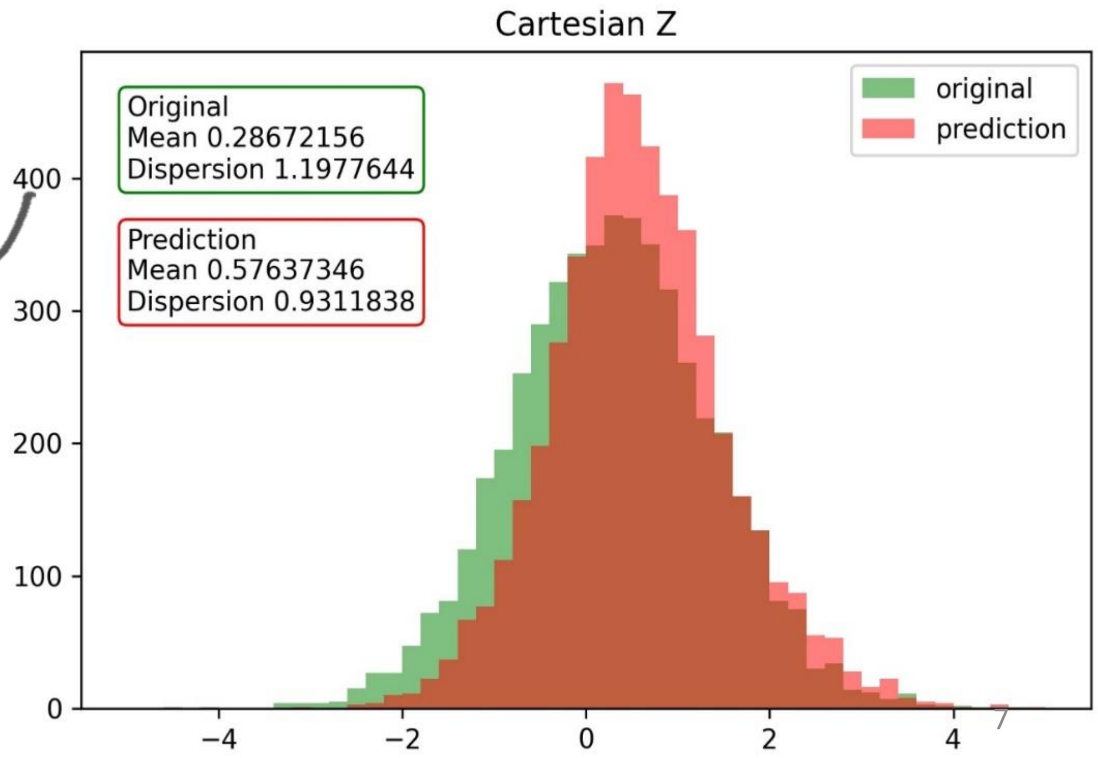
3 step: changing loss-function

$$loss(x,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2$$

Loss function takes two parameters as input - true and predicted values. The function shows how well the model performs in prediction. The higher its value, the worse the model works.

All normalized coordinate values are close to zero. Squaring such small values in the loss function gives the function value close to zero.

LOOT falls into the local minimum during the optimization process!



To solve this problem, it is necessary to give unnormalized values to the input of the loss function.

Ariadne: PyTorch Library for Particle Track Reconstruction Using Deep Learning

ARIADNE

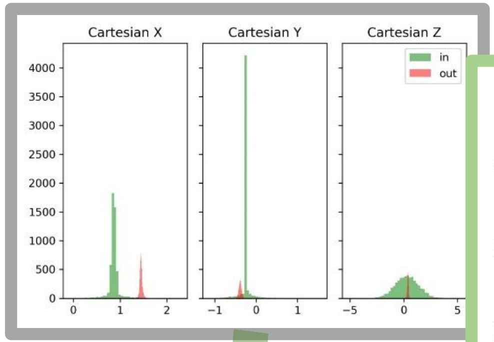
Ariadne – the first library for deep learning tracking on Python:

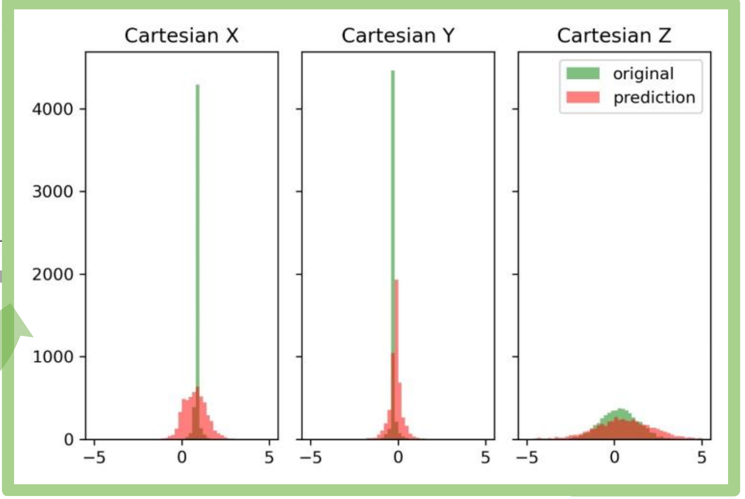
- ✓ any type of event data including collider and fixed-target experiments
- ✓ metrics logging, multiprocessing for data preparation, multi-GPU training
- ✓ open source and fully deterministic (https://github.com/t3hseus/ariadne)

Results, Conclusions and Outlook

1

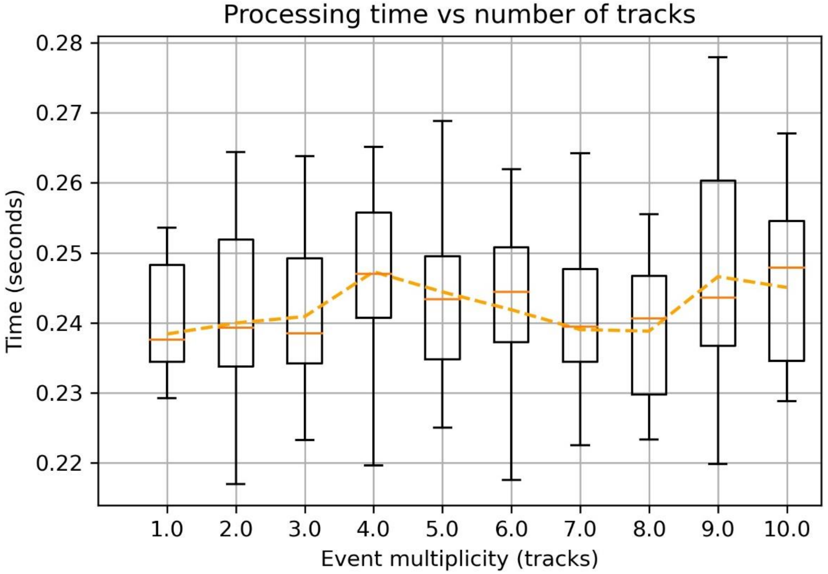
The distribution of coordinates has become closer to the true one. The result will be improved by training the model on more events.





2

Running time of the trained model does not depend on the multiplicity of the event.



3

The program code is included in the Ariadne library.

ARIADNE

Improvements of the LOOT model for primary vertex finding based on the analysis of development results

Ekaterina Rezvaya, Pavel Goncharov, Yury Nefedov, Gennady Ososkov, Alexey Zhemchugov

Государственный университет «Дубна»

ОИЯИ

Distributed Computing and Grid-technologies in Science and Education
July 06, 2021