BESIII Drift Chamber Tracking with Machine Learning

24th International Conference on Computing in High Energy and Nuclear Physics in Adelaide

Yao Zhang*[1], Ye Yuan[1], Ting Zhou[2], Karimi Khozani Hosein[1], Liqing Qin[2] , Qiumei Ma[1],

* zhangyao@ihep.ac.cn

1. Institute of High Energy Physics, CAS, Beijing, China
2. Guangxi Normal University, Guilin, China

BESIII experiment

The Beijing Spectrometer III (BESIII, Fig.1) has been running at the Beijing Electron Positron Collider II (BEPCII) for Tau-Charm physics since 2008.

The tracking detector of BESIII is a Multilayer Drift Chamber (MDC). The tracking efficiency for high transverse momentum (p_T) is high for charged particles but lower when (p_T) < 120MeV/c as shown in Fig. 2.

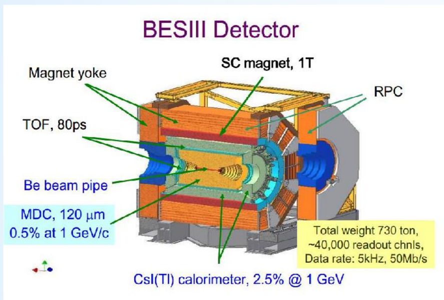

Fig.1 BESIII detector

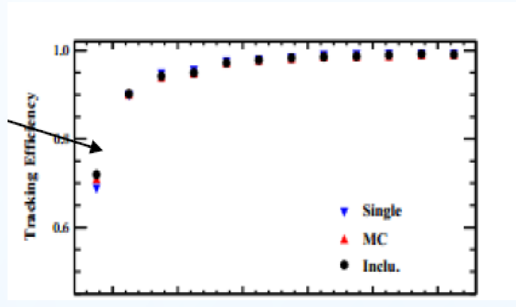

Fig.2 Tracking efficiency of π^- vs p_T

Multi-turn track finding

Some of the curling tracks with multi turn ($|\cos\theta|$<0.2) have bad tracking efficiency and quality as Fig. 3 shows.

For multi-turn tracks, the hits from different turns will close to each other and even overlap as Fig. 4 shows. The tracking efficiency is low due to effect of non-first turn's hits.

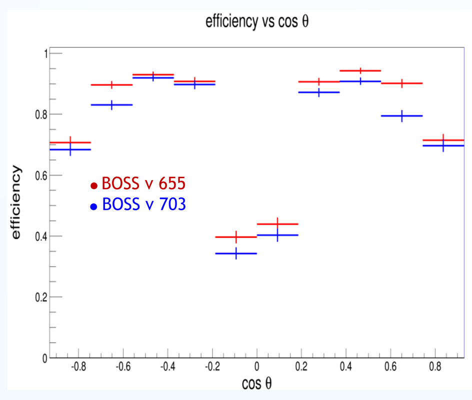

Fig.3 Curling track efficiency vs cosθ

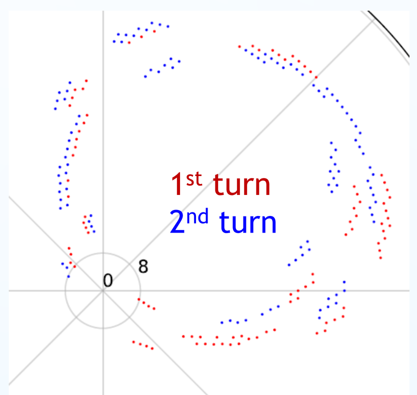

Fig.4 A multi-turn track in MDC

The current work to remove non-first turn hits is to apply a cut on the distance between track and wire as shown in Fig. 5.

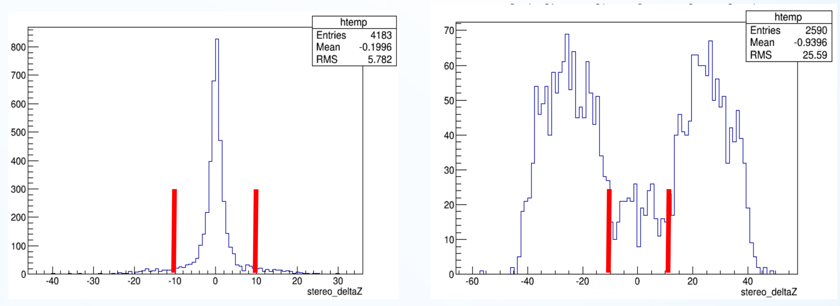

Fig.5 Distance between 1st turn's track and hits from 1st turn(left) and non-1st turn(right)

In this work we use the technology of the supervised semantic segmentation of deep learning to do the hit level classification for the single track. The aim is to separate 1st turn's hit from hits of other turns.

Data sample

To learn the behavior of multi-turn tracks, the training sample should have multi-turn curling tracks which are at small dip angle and various transverse momentum.

The particle used for this work are single μ^- and μ^+ . The track with transverse momentum less than 120MeV/c will curling in the MDC. Only track of more than one turn have been selected for the training and validation and testing.

The momentum, angle and turn number distribution of data set show in the Fig. 6.

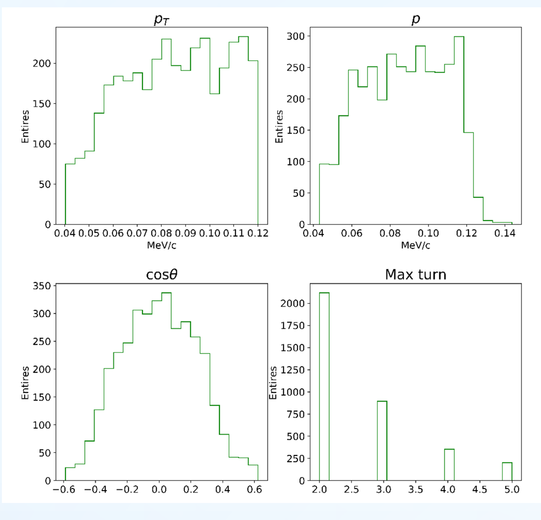

Fig.6 Distributions of data sample

Prepare for the input data

The BESIII drift chamber has 43 layers and the cell number of each layers range from 40 to 288. To make the shape fit between input and output, the rows and columns of the matrix are set to be the power of 2. The drift chamber cells of each layers were top centered mapped into the corresponding row of a 64 x 256 matrix.

The feature set to be the raw time of each hit after correction with the event start time. The cell without hits were set with a default time of 2000ns.

The truth turn number from Monte-Carlo simulation used as target. The hits from the first turn were marked as 1. The hits from other turns and cells without hit were set to be the default value of 0.

Dummy elements in the matrix were set to be the default values.

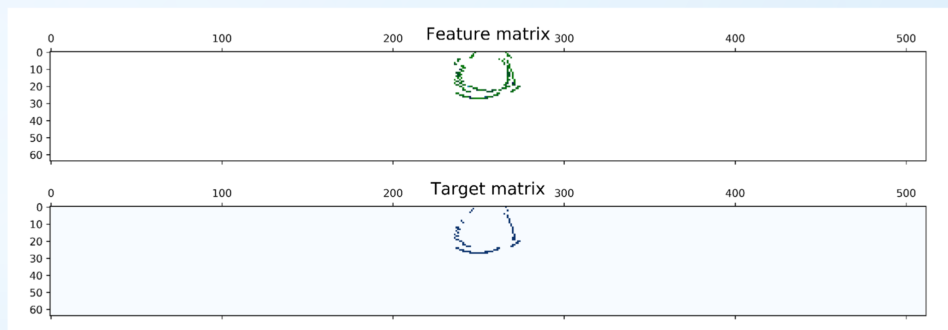

Fig.7 The feature and target matrixes of one track

Network model

U-Net is a convolutional networks for pixel level semantic segmentation. (https://arxiv.org/abs/1505.04597)
Following is the U-Net model used in this work.

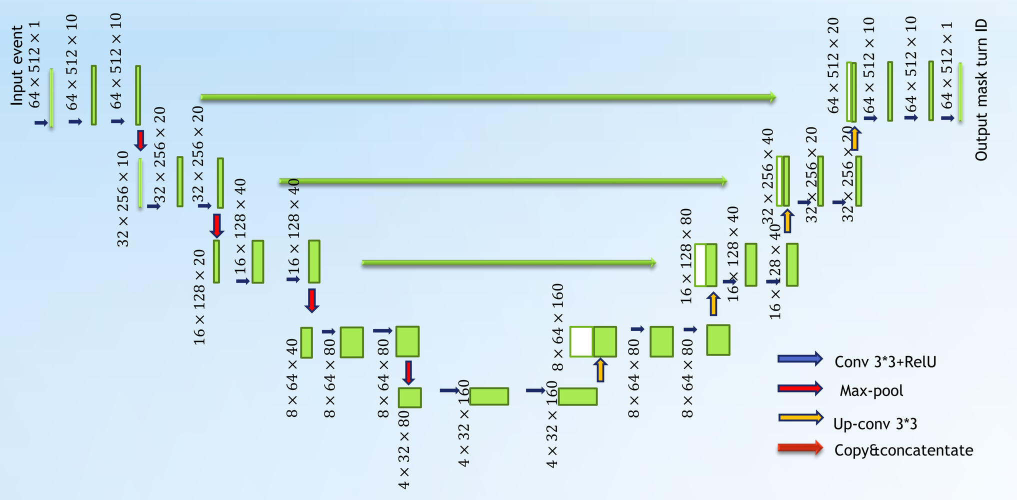

Fig.8 U-Net model used in this work

Loss function

The loss function used in this work is a binary cross entropy(BCE) plus a dice coefficient loss.
Where $BCE = -[Y\log f(X) + (1-Y)\log(1-f(X))]$.
Dice coefficient loss as a deep learning loss function for highly unbalanced segmentations. It was chosen because of the unbalanced number of element between 1st turn and non-1st turn.
(https://arxiv.org/abs/1707.03237)

$$Dice\ coefficient = \frac{2|X \cap Y|}{|X|+|Y|},\quad loss=1-(Dice\ coefficient)+BCE.$$

Where, X is the target matrix and Y is the prediction matrix.
So in this work, $|X|$ and $|Y|$ are approximately equal with the number of true first turn's hits and predicted first turn's hits respectively, $|X| \cap |Y|$ is approximately equal with the number of hits marked as first turn hits and predicted as first turn hits.

Training

A sample of 10 thousands track have been used for training. The data set was separated into 7500 tracks for training and 2500 tracks for validation. The batch size was 64 and the epoch number was 50. Following are the training loss and model accuracy v.s. epoch number. The training has converged after about 15 epochs for validation data set.

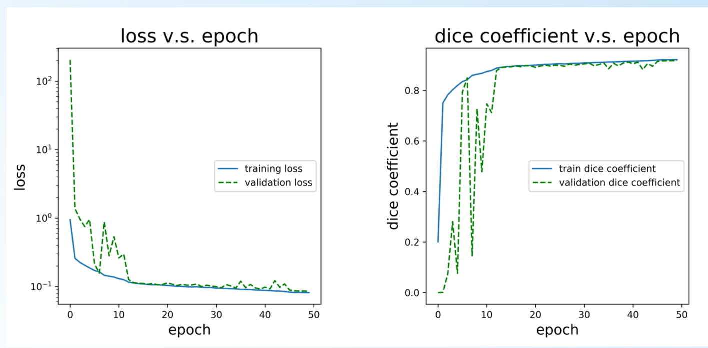

Fig.9 Loss and dice coefficient v.s. epoch

Performance

The efficiency and purity have been evaluated for single μ^- track with validation sample. The definition of efficiency and purity show as following: Efficiency $=\frac{TP}{P}$, Purity$=\frac{TP}{P'}$.

Where TP is the number of predicted first turn and marked as first by Monte-Carlo truth, P is the number of true first turn hits and P' is the number of hits predicted as first turn.

The ROC curve of the average hit efficiency and purity (Fig. 10) shows that at the threshold of about 0.85 the prediction gives the efficiency and purity are about 91%.

Fig.11 shows more than half of the events have efficiency and purity greater than 95%.
From the event display (Fig.12) we can see that the efficiency of the hits from inner layers could be improved.

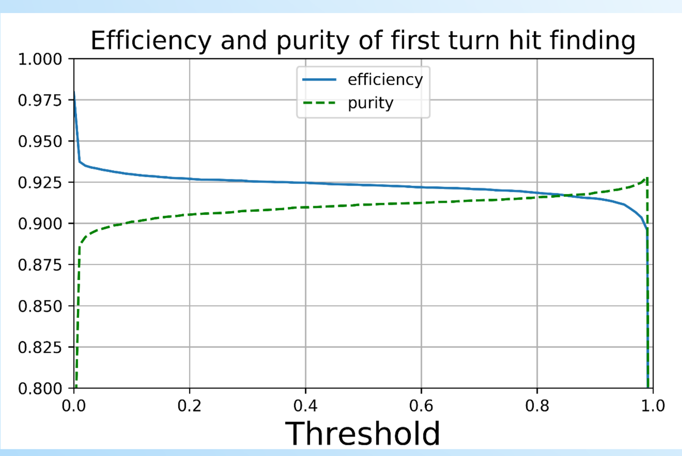

Fig.10 ROC of average efficiency and purity

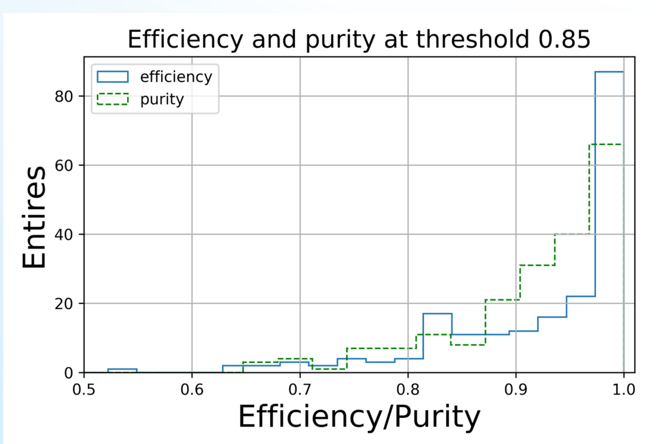

Fig.11 Distribution of efficiency and purity at threshold 0.85

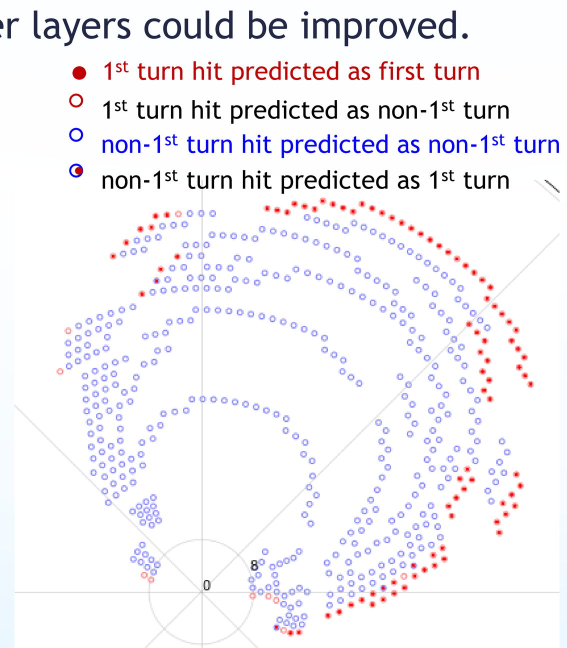

Fig.12 An event display of prediction

Future work

Train the network with other types of particle sample.
Train the network with the event with background and noise.
Study and optimize the performance.
Add detector geometry and track model as priori knowledge to the network.
Apply under BESIII offline software framework.

Conclusion and prospection

A deep learning effort has been performed to solve the problem of multi-turn curling track finding for BESIII drift chamber.

The preliminary performance is satisfied and promising.

When threshold is at 0.85, the average efficiency and purity for first turn's hits is about 91%.

We intend to investigate the separation of the hits for each track object using instance segmentation in deep learning in the future.