BESIII track reconstruction algorithm based on machine learning

Xiaoqian Jia[1], *Xiaoshuai* Qin[1*], *Teng* Li[1**], *Xingtao* Huang[1***], *Xueyao* Zhang[1], *Na* Yin[1], *Yao* Zhang[2], and *Ye* Yuan[2]

[1]Shandong University, 226237, Qingdao, China
[2]Institute of High Energy Physics, Chinese Academy of Sciences, 100049, Beijing, China

Abstract. Track reconstruction is one of the most important and challenging tasks in the offline data processing of collider experiments. For the BESIII detector working in the tau-charm energy region, plenty of efforts were made previously to improve the tracking performance with traditional methods, such as template matching and Hough transform etc. However, for difficult tracking tasks, such as the tracking of low momentum tracks, tracks from secondary vertices and tracks with high noise level, there is still large room for improvement. In this contribution, we demonstrate a novel tracking algorithm based on machine learning method. In this method, a hit pattern map representing the connectivity between drift cells is established using an enormous MC sample, based on which we design an optimal method of graph construction, then an edge-classifying Graph Neural Network is trained to distinguish the hit-on-track from noise hits. Finally, a clustering method based on DBSCAN and RANSAC is developed to cluster hits from multiple tracks. Track fitting algorithm based on GENFIT2 is also studied to obtain the track parameters, where deterministic annealing filter are implemented to deal with ambiguities and potential noises. The preliminary results on BESIII MC sample presents promising performance, showing potential to apply this method to other trackers based on drift chamber as well, such as the CEPC and STCF detectors under pre-study.

1 Introduction

Beijing Spectrometer III (BESIII) [1] is a magnetic spectrometer detector operating at BEPCII with $\sqrt{s} = 2 \sim 4.9\ GeV/c^2$. BESIII's primary physics goals are to study the eletroweak and strong interactions and to search for new physics at the tau-charm energy region. At the boundary between the perturbative and nonperturbative regimes of QCD, BESIII offers vast and diverse physics opportunities.Plenty results related to hadron physics and τ-charm physics play an important role in the understanding of particle's inner structure and interactions [2].

The Main Drift Chamber (MDC) is a crucial component of the BESIII detector, which is mainly responsible for the precise detection of the trajectory of charged particles. MDC is a cylindrical detector consisting of 43 layers of drift cells filled with a gas mixture. When

*e-mail: qinxs@ihep.ac.cn
**e-mail: tengli@sdu.edu.cn
***e-mail: huangxt@sdu.edu.cn

a charged particle passes through the MDC, it ionizes the gas atoms, creating electron-ion pairs. The electric field within the drift cells causes the electrons to drift towards the anode wires, while the ions drift towards the cathode plates. By measuring the arrival times of the electrons at the anode wires, the MDC is able to detect the hits of the particle along its trajectory.

Over the last decades, traditional tracking strategy for the BESIII MDC system has been successfully working as the official reconstruction algorithms, including the tracking finding algorithms via PATTSF [3, 4] and HOUGH [5], track parameters estimation based on Kalman Fitter and Rugge Kutta methods. This strategy works well, but there is still a decent room for improvement, especially for tracks with low momentum, high noise level as well as tracks from long lived particles (secondary vertex).

Inspired by the work on TrackML particle tracking challenge, which is aimed at solving tracking problems of the LHC based on machine learning methods, we stared to explore machine learning methods on tracking of the BESIII as well. In this work, we propose a comprehensive approach for track finding and kinematic information retrieval. Firstly, we employ a graph neural network (GNN) to effectively filter out noise from hit-on-track. Next, we utilize Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [6] in conjunction with Random Sample Consensus (RANSAC) [7] to perform track finding. Finally, we employ a Generic Track-Fitting Toolkit (GenFit2) [8] to retrieve the kinematic information of the identified tracks.

Section 2, 3 and 4 describe the design and implementation of a new track reconstruction algorithm for BESIII based on machine learning. The preliminary performance of the algorithm is shown in Section 5. Section 6 gives the summary and outlook on the further development of the algorithm. The data used in this work is based on the Monte Carlo (MC) data which is produced with the BESIII offline software [9].

2 Filtering noise via GNN

To achieve high-precision tracking, it is crucial to accurately associate the hits detected by the drift chamber with the corresponding particle tracks (hit-on-track). However, this task could be challenging due to the presence of noise hits, which are caused by various sources such as the electronic noise, the beam background and cosmic rays etc. Distinguishing between hit-on-track and noise hits is essential for reliable particle tracking and the subsequent processing.

In order to address this challenge, a noise filter algorithm is developed based on the edge-classifying GNN approach and innovatively establish a Pattern Map for graph construction. By training the network on a large dataset of simulated events, we aim to develop a robust and efficient algorithm for hit classification.

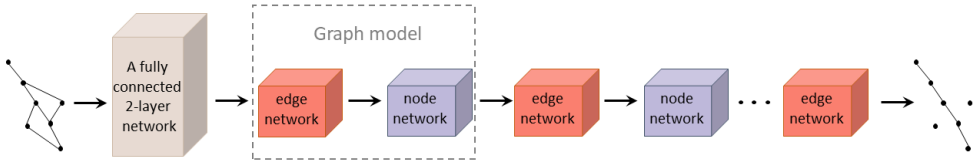


Figure 1: Network structure for noise filtering

2.1 Graph construction

In the GNN model, the graph is constructed so that the nodes are the hits detected by the detector and the edges are the valid connections of the hits between sense wires, determined by the Pattern Map. The Pattern Map defines the neighbors for each sense wire in adjacent layers by counting the occurrences of consecutive sense wire combinations that a track pass through. To accomplish this, we generate two million single-track events for five types of charged particles (including $e^{\pm}$, $K^{\pm}$, $\mu^{\pm}$, $p^{\pm}$, $\pi^{\pm}$) with a momentum ranging from 0.05 to 3 GeV/c. We calculated the probability of neighbor occurrences, and in order to further reduce the complexity of the graph, we set a small value on the probability of the neighbor combination, which is verified to effectively speed up the graph construction while keep almost the same performance of filtering noise.

During the graph construction workflow, we assign edges for the neighbors based on the Pattern Map. We also assign edges between adjacent sense wires in the same layer. Additionally, if the adjacent layer is empty, we connect it to the neighbors of neighbors on the intermediate layer to ensure uninterrupted track continuity.

2.2 Hit classification

The GNN takes in node features as input, which consist of the raw drift time and the two cylindrical coordinates (r, ϕ). These features are initially transformed into latent representations through a fully connected two-layer neural network. These representations are then fed into the GNN model during training.

The GNN model is composed of an edge network and a node network. The edge network updates the edge weights, while the node network calculates new node features. Both networks are Multi-Layer Perceptrons (MLPs) with four layers. The GNN architecture is illustrated in Figure 1 . During the information transmission, important connections are reinforced while irrelevant or spurious ones are weakened. Our research has demonstrated that the best performance is achieved when the graph model is iterated for eight epochs. The final edge weights serve as the output of the entire model, representing the probability of combination. If the output score of an edge exceeds the threshold, the two nodes connected by it are considered as hits on the track, while the remaining ones are regarded as noise hits. In the networks, we employ the Adaptive Moment Estimation (ADAM) optimizer [10] and the Binary Cross-Entropy Loss Function for optimization.

3 Clustering of tracks based on DBSCAN and RANSAC

The clustering algorithm handles the hits that are retained after GNN noise filtering. The position of the hits corresponds to the 2D coordinates of the signal wires. Therefore, our algorithm is based on a 2D clustering approach utilizing both DBSCAN and RANSAC, where hits in one cluster are considered to be on the same track. The algorithm prioritizes the use of DBSCAN for clustering. However, if a particular cluster in the clustering result contains a number of hits that exceeds a predefined threshold, it is considered that DBSCAN has mistakenly grouped multiple tracks into a single track. This situation mostly occurs when the tracks have a compact distribution. In such case, the algorithm triggers the RANSAC algorithm to perform re-clustering, optimizing the accuracy of the clustering process.

3.1 Hit clustering based on DBSCAN

DBSCAN is a density-based clustering algorithm, and we applied a transformation to the original space before applying it to make the hits on the same track more densely distributed

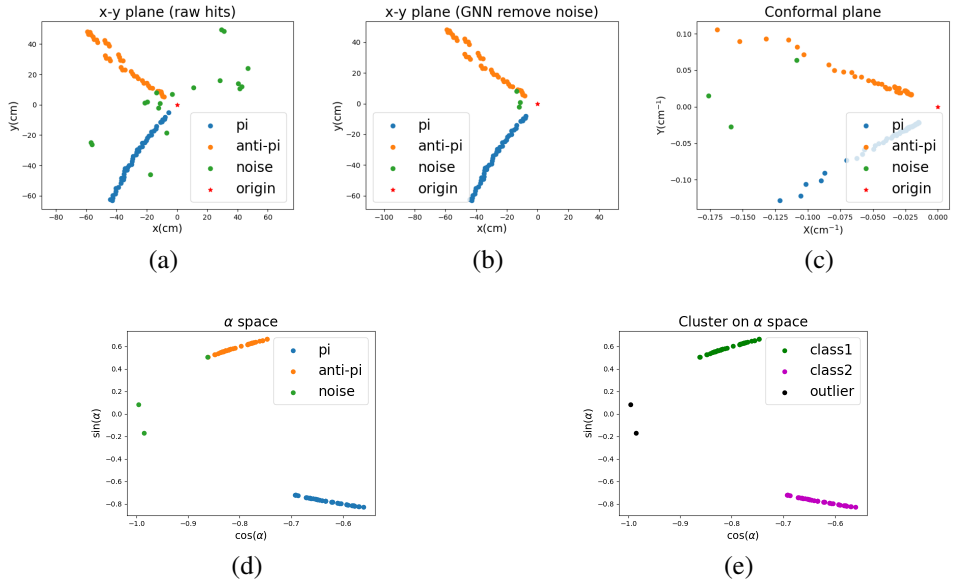


Figure 2: (a) Hits in the original space (xy space). (b) Hits in the original space after GNN filtering. (c) Hits on the conformal plane. (d) Hits on the parameter space. (e) Cluster results.

while keeping the hits from different tracks more separated. The specific transformation process is as follows:

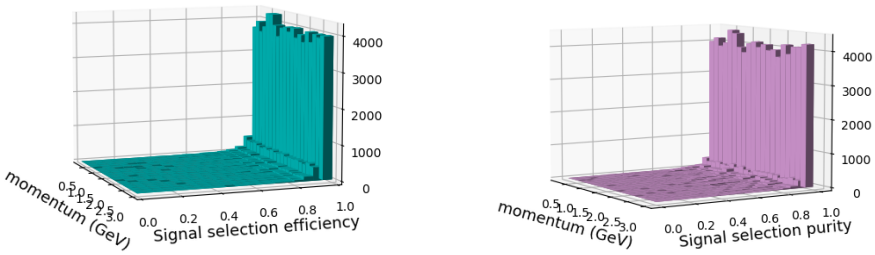
Figure 2(a) and 2(b) show the distribution of hits in the original space (xy space) before and after GNN filtering, respectively. After applying a common transformation (X=$\frac{2x}{x^2+y^2}$, Y=$\frac{2y}{x^2+y^2}$), a circle passing the origin can transform into a straight line, as shown in Figure 2(c). Then the angle α between the line where the hit and the origin are located and the positive half axis of X in the common space is obtained. Figure 2(d) represents the parameter space as cosα and sinα, and Figure 2(e) shows the clustering results.

By transforming the coordinates, we can effectively adjust the density distribution of the hits. This helps to improve the performance of the DBSCAN algorithm in identifying clusters and separating different tracks. DBSCAN is sensitive to the input parameters *eps* and *MinPts*, here *eps* represents the maximum distance between two samples for one to be considered as in the neighborhood of the other and *MinPts* represents the minimum number of samples in a neighborhood for a point to be considered as a core point. Currently, we use an automatic parameter tuning algorithm to select appropriate parameters based on the hit distribution of each event.

3.2 Hit clustering based on RANSAC

The RANSAC algorithm is capable of finding a set of data points that conform to a specific model distribution. It starts by randomly selecting a subset of data points and fitting a model to them. RANSAC then evaluates the fit of the model to the remaining data points and determines which points are considered inliers. The process is repeated for a certain number of iterations or until a stopping condition is met.

In this study, we leverage the power of RANSAC to correct clusters that have been incorrectly grouped together by DBSCAN. RANSAC is able to identify the best-fit model that optimally explains the inliers while disregarding the outliers. This makes RANSAC particularly suitable for finding a single track from multiple tracks, even when the trajectories overlap significantly. When RANSAC identifies the first track, it is then used again on the remaining hits to find subsequent tracks. This iterative process continues until the stopping condition is satisfied, which could be based on the number of remaining points. But if there is still a cluster that contains a significant number of hits, indicating a potential error in the clustering, the algorithm applies RANSAC again to that specific cluster. Currently, we have only used a simple linear model for feasibility studies. This model is not very suitable for representing tracks, so further optimization is needed in the future.



(a) Efficiency (b) Purity

Figure 3: Signal selection performance

4 Track fitting based on GenFit2

GenFit2 is an experiment-independent framework for track reconstruction for particle and nuclear physics, which is already used in $\bar{P}$ANDA, Belle II, FOPI and other experiments [11]. The deterministic annealing filter (DAF) is implemented in GenFit2 and perfectly suited to resolving the left-right ambiguities of wire measurements, which is crucial for drift chamber based tracking system such as BESIII.

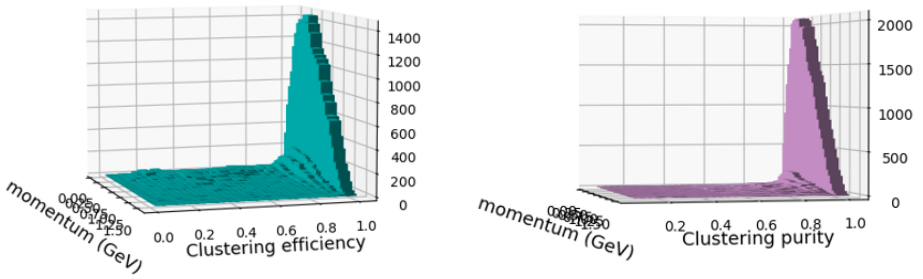
Therefore, a new toolkit based on GenFit2 is developed as an alternative track fitting algorithm with regard to traditional tracking procedure.

The input for track fitting is the selected hits on tracks after the previous track clustering.At the beginning, a rough drift distance information is provided for each hit where the left-right ambiguity is not solved yet. First attempt of track fitting will update information such as left-right ambiguity, entrance angle of tracks with respect to cell elements in MDC, which can then be used for further track fitting. The kinematic information of tracks can be highly improved and finally settled after some additional iterations, which can be used for reconstruction of the whole detector or preliminary physics analysis.

5 Preliminary results

Single-particle ($e^{\pm}$, $K^{\pm}$, $\mu^{\pm}$, $p^{\pm}$, $\pi^{\pm}$) MC sample, which mixed with BESIII random trigger data as background (~45% hits) are used to test hit classification performance. Efficiency ($N_{signal}^{predicted}$ / N_{signal}^{real}) and purity ($N_{signal}^{predicted}$ / $N_{all}^{predicted}$) of signal selection are shown in Figure 3, most events are distributed in the last bin (97% - 100%).

Figure 4 shows the performance of track clustering after filtering noise. Test data is generated by the decay channel $J/\Psi \to \rho^0\pi^0 \to \gamma\gamma\pi^+\pi^-$ simulated by MC, and the transverse momentum of the final charged particles range from 0.2 GeV to 1.4 GeV. The track clustering algorithm exhibits high purity and reasonably good efficiency, indicating that a significant proportion of the true tracks are successfully clustered.



(a) Efficiency (b) Purity

Figure 4: Clustering performance

Figure 5 presents a comparison of the track reconstruction efficiency between our method and the BESIII tracking method, PATTSF + HOUGH, using the same data set as described in the preceding paragraph. As a preliminary result of this method, the track reconstruction efficiency is slightly inferior to the traditional results but remains almost comparable. This indicates its promising potential with further improvements in the future.

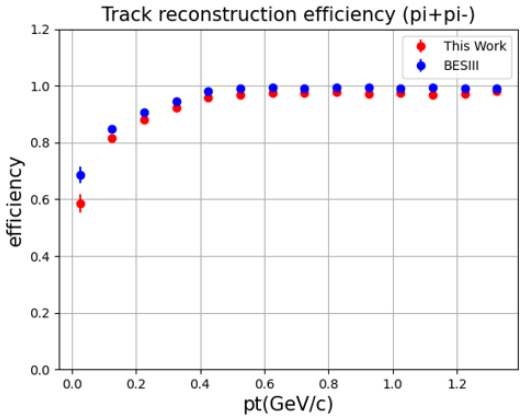


Figure 5: Track reconstruction efficiency

6 Conclusions

In this work, we demonstrate a novel tracking algorithm based on machine learning method, including a GNN to distinguish the hit-on-track from noise hits, the clustering methods based

on DBSCAN and RANSAC to cluster hits from multiple tracks, and the track parameter estimation based on GenFit2 for track fitting. The preliminary results demonstrate promising performance, with a very similar performance to traditional approaches based on PATTSF and HOUGH transformation. Further optimization of our network model and methods, especially the clustering algorithm, will result in even better performance.

7 Acknowledgments

This work is supported partly by National Key R&D Program of China under Contract No. 2020YFA0406304 and National Natural Science Foundation of China (NSFC) under Contracts Nos.12025502, 12105158, 12175124, 12188102.

References

[1] M. Ablikim, Z. An, J. Bai, N. Berger, J. Bian, X. Cai, G. Cao, X. Cao, J. Chang, C. Chen et al., Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **614**, 345 (2010)

[2] D. Asner, I. Bigi, J. Charles, J. Chen, H. Cheng, S. Descotes-Genon, K. He, H. Li, J. Liu, H. Liu et al., International Journal of Modern Physics A **24**, 499 (2009)

[3] Z. Yao, Z. Xue-Yao, L. Wei-Dong, M. Ze-Pu, M. Qiu-Mei, M. Xiang, W. Liang-Liang, W. Ji-Ke, D. Zi-Yan, Y. Zheng-Yun et al., Chinese Physics C **31**, 570 (2007)

[4] Q. Liu, S. Zang, W. Li, Z. Mao, J. Bian, G. Cao, X. Cao, S. Chen, Z. Deng, C. Fu et al., Chinese Physics C **32**, 565 (2008)

[5] J. Zhang, Y. Zhang, H.M. Liu, Y. Yuan, X.Y. Zhang, L.Y. Dong, Z. Huang, X.B. Ji, H.B. Li, W.G. Li et al., Radiation Detection Technology and Methods **2**, 1 (2018)

[6] M. Ester, H.P. Kriegel, J. Sander, X. Xu et al., *A density-based algorithm for discovering clusters in large spatial databases with noise*, in *kdd* (1996), Vol. 96, pp. 226–231

[7] M.A. Fischler, R.C. Bolles, Commun. ACM **24**, 381–395 (1981)

[8] Github, *Homepage* (https://github.com/GenFit/GenFit)

[9] W.D. Li, H.M. Liu, Z. Deng, K. He, M. He, X. Ji, L. Jiang, H. Li, C. Liu, Q. Ma et al., *The offline software for the BESIII experiment*, in *Proceeding of CHEP* (2006), Vol. 27

[10] D.P. Kingma, J. Ba, arXiv preprint arXiv:1412.6980 (2014)

[11] T. Bilka, N. Braun, T. Hauth, T. Kuhr, L. Lavezzi, F. Metzner, S. Paul, E. Prencipe, M. Prim, J. Rauch et al., arXiv preprint arXiv:1902.04405 (2019)