

# DyTER - Dynamic Track and Event Reconstruction

Michael Papenbrock<sup>1,\*</sup> on behalf of the PANDA collaboration

<sup>1</sup>Uppsala University, Department for Physics and Astronomy

**Abstract.** The future PANDA experiment at FAIR experiment aims to cover a wide range of processes in antiproton-proton collisions at event rates of up to 20 MHz. Coupled with a purely software-based event filter, this puts a high demand on the involved reconstruction algorithms. Investigating complex event topologies with displaced vertices increases the difficulty even further. Here we present two attempts to meet these future challenges: a more traditional algorithm for track reconstruction based on pattern matching with pre-determined look-up tables, and as a continuation of this approach a system of neural networks for identifying specific particle track candidates and predicting their momentum.

## 1 Track and Event Reconstruction at PANDA

PANDA (Antiproton Annihilation at Darmstadt) [1] is a multi-purpose detector (see figure 1) currently under construction at the future Facility for Antiproton and Ion Research (FAIR) in Darmstadt, Germany. The fixed-target experiment aims to investigate a wide range of antiproton induced reactions with an angular coverage of nearly  $4\pi$ . The High Energy Storage Ring (HESR) will deliver high intensity antiproton beams with momenta of up to 15 GeV/c with a nearly continuous beam structure. The event rates are expected to reach up to 20 MHz. Coupled with an event filtering that will be implemented purely in software, this puts high demands on the online reconstruction algorithms.

Amongst the many processes PANDA will study, there are those with challenging topologies involving secondary vertices. Considering that in traditional algorithms the interaction point often serves as a powerful constraint, reconstructing tracks that originate several centimetres or even metres away from the primary vertex can drastically increase the complexity of the algorithms. Combinatorics may increase dramatically and so do the computing requirements. Hence, flexible algorithms are needed for track and event reconstruction.

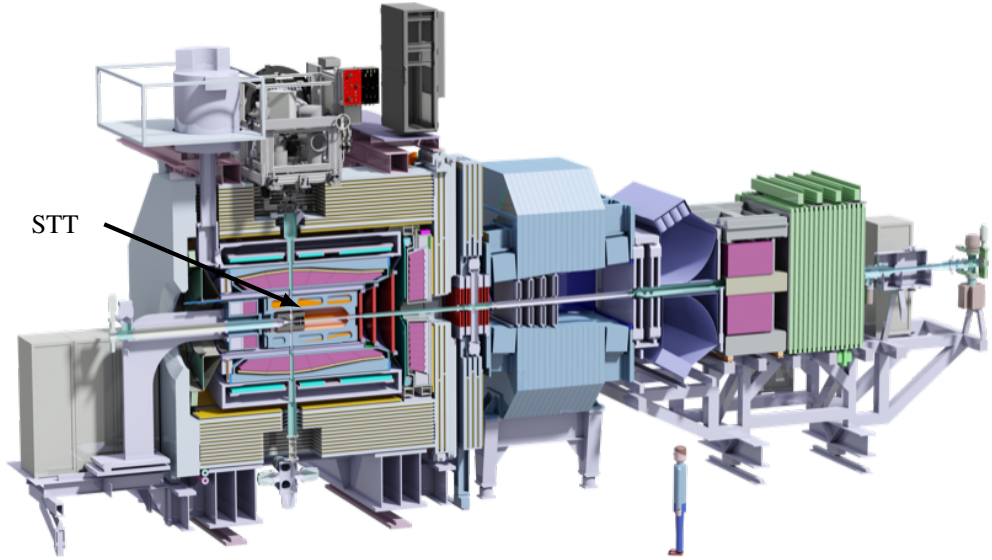
Here, hyperon reactions prove to be good benchmark channels due to their complex topologies. These include displaced vertices, intersecting tracks, and the final state particles reach all of PANDA's subdetectors for charged particle detection. Consequently, the following studies put a particular focus on the  $\bar{p}p \rightarrow \bar{\Lambda}\Lambda$  reaction.

## 2 Pattern Matching

A more traditional approach to track reconstruction can be found in the form of pattern matching. The basic concept is to simulate virtually every track that can occur in the detector in a

---

\*e-mail: [μmichael.papenbrock@physics.uu.se](mailto:μmichael.papenbrock@physics.uu.se)

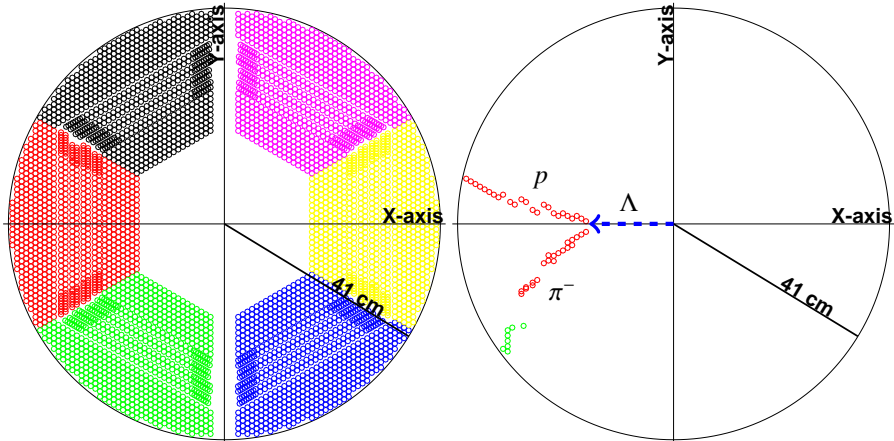


**Figure 1.** The PANDA detector. Indicated is the Straw Tube Tracker (STT).

real measurement and store its distinct combination of activated detector elements, the pattern, in some form of look-up table. Once the investigation of realistic data commences, the incoming detector signatures are compared with the recorded patterns and the best matches are identified as possible track candidates. This approach is appealing as it has proven to be rather reliable and robust over the years (*e.g.* [2]). Furthermore, it lends itself well to implementation on hardware accelerators such as FPGAs, which is of particular importance when considering it for potential online use, *i.e.* while the experiment is running and acquiring data.

In order to create the look-up table for the pattern matching algorithm, the segmentation of the Straw Tube Tracker of PANDA into six individual sectors corresponding to the readout electronics is taken into account [3], illustrated in figure 2. Training data are generated with *PandaRoot*, from which patterns are extracted. First, events are created with an appropriate event generator. The resulting tracks are then propagated through the detector using Geant4 [4]. After a digitisation stage which creates realistic hit data, track candidates are identified by an ideal track finder using Monte Carlo truth information. These steps yield all the information needed to create a pattern database. A pattern is composed as follows: The actual hit pattern, the momentum of the particle creating that pattern, the time stamps of the hits, the sector ID as well as the number of occurrences with the same tube IDs.

Considering only trajectories from delayed decays, as is the case in hyperon production reactions, the range of possible combinations of corresponding activated straw tubes becomes rather large. In order to estimate the feasibility of the pattern matching approach for such reactions, the amount of needed training data and eventual size of the pattern database have to be considered. For a first estimate, we generated one million events of the  $\bar{p}p \rightarrow \bar{\Lambda}\Lambda$  reaction, yielding  $\approx 2.2$  million patterns. Afterwards, patterns with an identical set of tubeIDs were merged. While a subset of patterns showed a high repetition rate, up to  $O(1000)$  occurrences per pattern, a very large set of generated tracks would be necessary for convergence, as illustrated in figure 3 (black dots). However, only considering exact matches is a steep requirement. Requiring a share of 90% of matching tubes for patterns in order to merge, con-



**Figure 2.** **Left:** A sketch of the PANDA Straw Tube Tracker divided into six sectors. **Right:** The signature of a  $\bar{p}p \rightarrow \bar{\Lambda}\Lambda$  event. The  $\Lambda$  traverses into the STT and decays into a  $p$  and a  $\pi^-$ , creating hit patterns across multiple sectors.

vergence is achieved much more quickly. These are also shown in figure 3 (blue triangles). They converge rather quickly and saturate at  $\lesssim 100000$ , which corresponds to a database size of less than 100 MB. Further investigation will aim to find a good balance between accuracy and computing requirements.

While this can be further developed into a standalone track finder suitable to be run on FPGAs, we also explored the possibility to the generated database as an input for machine learning algorithms.

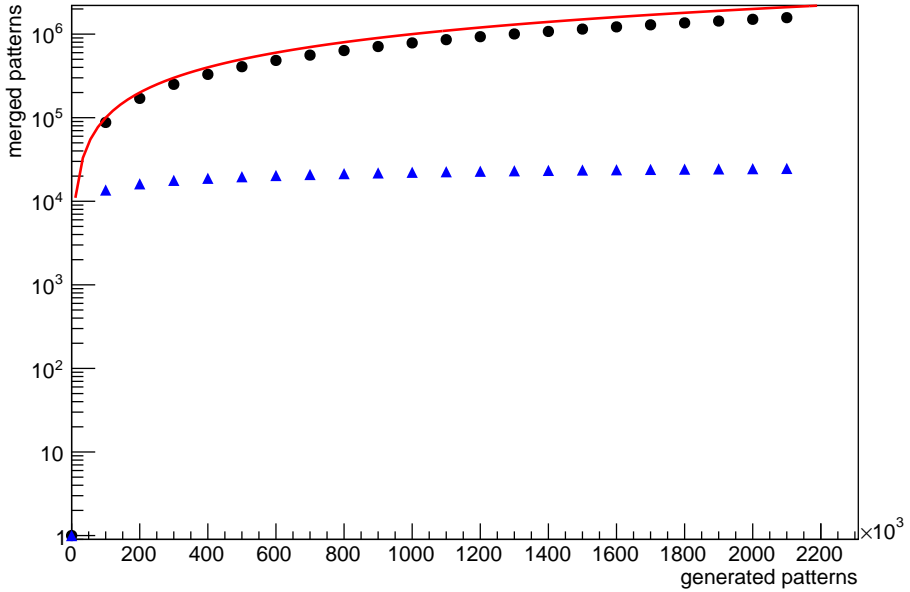
### 3 Neural networks

In recent years, the field of machine learning and artificial intelligence has seen an incredible rise in popularity. The applications and implementations cover a wide range, with image processing using convolutional neural networks (CNN) among the more common ones. While neural networks had a presence in high energy physics for many decades, this recent surge has not gone unnoticed in this realm either. In fact, it has lead to considering machine learning algorithms for applications ranging from track reconstruction to high level analysis [5]. The situation seems more appealing than ever. For example, neural networks can be efficiently mapped onto the hardware of an FPGA. Furthermore, the manufacturers like nVidia have acknowledged the trends and are including tensor processing cores designed for machine learning algorithms on their current GPUs. This allows these techniques to achieve performances that would be otherwise difficult to reach with more traditional algorithms.

As an extension of the developments outlined in section 2 we explored the possibilities of employing a neural network for the purpose of particle tracking with the PANDA detector. As before, we focused on the Straw Tube Tracker. A first prototype was developed [6] with a simplified problem description:

- Can we identify one specific particle track within a  $\bar{\Lambda}\Lambda$  event topology?
- Can we extract physical observables, *i.e.* the transversal momentum components?

To answer these questions, two fully connected neural networks were implemented, one for pattern recognition, the other for momentum regression. The implementation was done in



**Figure 3.** The merged patterns plotted versus the generated ones for 1000000  $\bar{p}p \rightarrow \bar{\Lambda}\Lambda$  events. The red line illustrates the case when no merging is done at all. The black points represents the number of patterns in the database after those with identical sets of tubeIDs have been merged. The blue triangles show the number of patterns after those with 90% matching tubeIDs have been merged.

MATLAB and the networks were trained with data from the aforementioned pattern matching algorithm.

The purpose of the pattern recognition was defined to identify a proton track candidate in the events of a  $\bar{p}p \rightarrow \bar{\Lambda}\Lambda$  test sample. The input layer consisted of the activated tubeIDs of all raw Straw Tube Tracker hits in an event. The output layer would then return the set of tubeIDs for the specific particle track. In between, there were four hidden layers. The features of the network are given in table 1.

Layer	Number of neurons	Activation function
Input layer	4542	-
Hidden layer 1	300	ReLU
Hidden layer 2	200	ReLU
Hidden layer 3	200	ReLU
Hidden layer 4	300	ReLU
Output layer	4542	Sigmoid

**Table 1.** Properties of the neural network for pattern recognition. The used activation functions are the rectified linear unit activation function (ReLU) with  $f(x) = \max(0, x)$  in  $[0, \infty)$  and the Sigmoid function with  $f(x) = \frac{1}{1+e^{-x}}$  in  $(0, 1)$ . Taken from [6].

The output of the pattern recognition network then served as input for the momentum regression network with the goal to obtain an estimate for the track's  $p_x$  and  $p_y$  momentum components, which would constitute the output layer. Six hidden layers were used. More features for this network can be found in table 2.

Layer	Number of neurons	Activation function
Input layer	4542	-
Hidden layer 1	400	ReLU
Hidden layer 2	300	ReLU
Hidden layer 3	200	ReLU
Hidden layer 4	100	ReLU
Hidden layer 5	50	ReLU
Hidden layer 6	30	ReLU
Output layer	2	Identity

**Table 2.** Properties of the neural network for momentum regression. The used activation functions are the rectified linear unit activation function (ReLU) with  $f(x) = \max(0, x)$  in  $[0, \infty$  and the Identity function with  $f(x) = x$  in  $(-\infty, \infty)$ . Taken from [6].

A data set containing  $\approx 750000 \bar{p}p \rightarrow \bar{\Lambda}\Lambda$  events was simulated using *PandaRoot*. The networks were then trained with over 2000 batches from this training set with one batch containing 1000 events. A separate set containing 10000 events was used for validation.

The workflow of the neural networks are visualised in figure 4. A state vector derived from the raw hits of the Straw Tube Tracker is used as an input layer. Then, the networks determines a probability for each tube in the detector for a proton having been present, retaining only those tubes which yielded a proton hit probability larger than 0.99. This output is correlated with the actual input, keeping only those tubes which gave a physical hit signal. The resulting state vector is passed as input to the momentum regression network which gives an estimate for  $p_x$  and  $p_y$ . This is compared here with the true Monte Carlo momentum that was originally generated.

The left part of figure 5 shows the loss and prediction accuracy of the pattern recognition network. It reaches a high accuracy at approximately 95%. Furthermore, with the performance on the test data being very similar to that on the training data, signs of overfitting are rather small. The same plot for the momentum regression network can be seen in the right hand part of figure 5. A prediction was defined as correct if the momentum vector falls within  $5^\circ$  of the Monte Carlo truth momentum vector and within 10% of its magnitude. Here, the accuracy is notably lower and larger batch numbers, signs of overfitting increase.

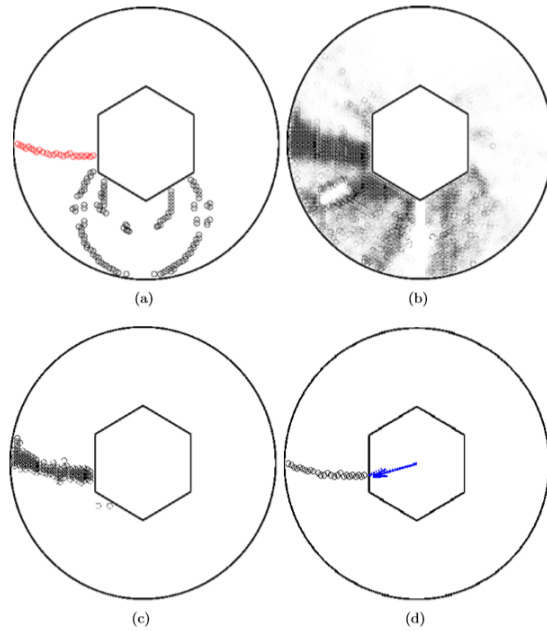
In order to evaluate the performance of this prototype from the perspective of a track finding algorithm, we define the following quantities

$$\text{prediction accuracy} = \frac{TH + TM}{TH + TM + FH + FM}, \quad (1)$$

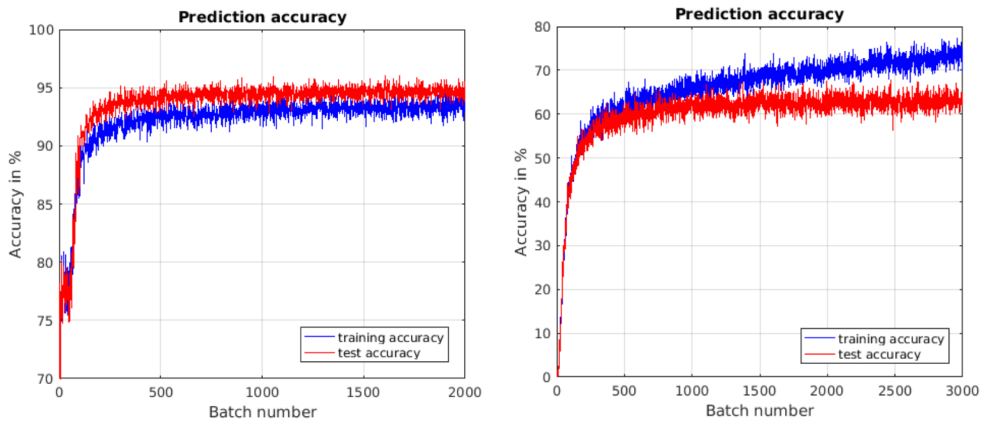
$$\text{efficiency} = \frac{TH}{TH + FM}, \quad (2)$$

$$\text{purity} = \frac{TH}{TH + FH}, \quad (3)$$

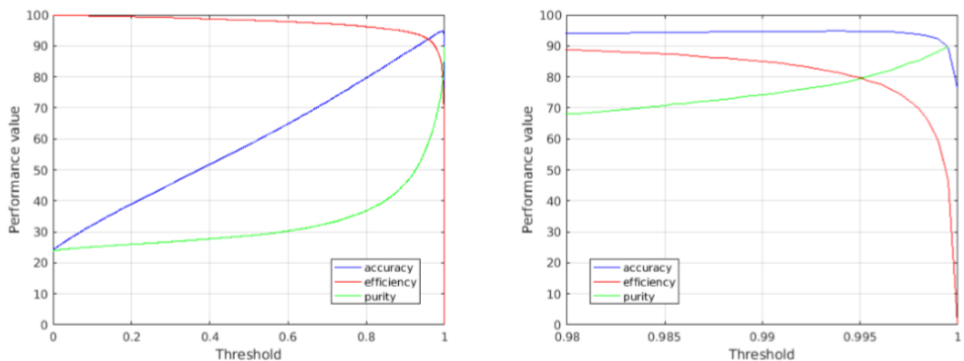
with  $TH$  being true hits,  $TM$  true misses,  $FH$  false hits, and  $FM$  false misses. These are plotted in figure 6 against the threshold value. These give insight how the threshold value can be optimised to find a suitable compromise between accuracy, efficiency, and purity, in particular in the range between 0.98 and 1.0.



**Figure 4.** Visualisation of the neural networks. (a) The input for the pattern recognition network. The proton track is highlighted in red for the reader. (b) The raw output of the network. The gradient corresponds to the probability that a proton was present. (c) The output of the network after a threshold cut was applied for the probability. Only probabilities  $> 0.99$  were retained. (d) The filtered hits after correlation with the initial input. Also shown are the predicted momentum (solid arrow) together with the true momentum (dotted arrow).



**Figure 5.** The prediction accuracy of the pattern recognition network (left) and the momentum regression network (right) as a function of the number of batches used in training. Taken from [6].



**Figure 6.** The prediction accuracy, efficiency, and purity for the pattern recognition network as a function of the threshold value for the full range (left) and the final 2% (right). Taken from [6].

## 4 Conclusions

Developing track and event reconstruction for the PANDA experiment is a challenging task. The high expected event rates coupled with a purely software-based triggering system puts high demands on the reconstruction algorithms, in particular when being used online. A more traditional pattern matching approach has been studied, which uses pre-determined look-up tables in order to find expected track signatures in the incoming data stream.

A natural extension of this concept is the exploration of machine learning techniques for track and event reconstruction, in particular with neural networks. A first prototype was developed showing promising results in the areas of specific track identification and momentum regression. These networks tend to be highly parallelisable, making them a good match for GPUs or FPGAs. Hence, they may present a good candidate to meet the performance requirements in high event rate scenarios.

With our future efforts we will explore the applicability of both the pattern matching and neural network algorithms to be run on FPGAs. Furthermore, the neural network solutions are currently being ported to Python/TensorFlow and being extended to all types of tracks and other tracking detectors in PANDA.

The authors are grateful to the Knut and Alice Wallenberg Foundation for the support.

## References

- [1] PANDA Collaboration, arXiv:0903.3905 (2009)
- [2] N. Ilic, JINST **12**, (2017)
- [3] PANDA Collaboration, Eur.Phys.J. **A49**, 25 (2013)
- [4] S. Spataro, JoP: Conf. Series **119**, 032035 (2008)
- [5] K. Albertsson *et al.*, arXiv:1807.02876, (2018)
- [6] A. Jonnarth and A. Hedkvist, project report, (2018), [http://www.it.uu.se/edu/course/homepage/projektTDB/ht17/project01/Project01b\\_report.pdf](http://www.it.uu.se/edu/course/homepage/projektTDB/ht17/project01/Project01b_report.pdf)