DyTER - Dynamic Track and Event Reconstruction

Michael Papenbrock

Department for Physics and Astronomy
Uppsala University

for the PANDA collaboration

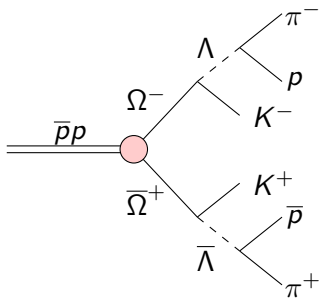
July 11th, 2018
CHEP 2018
Sofia, Bulgaria

Motivation

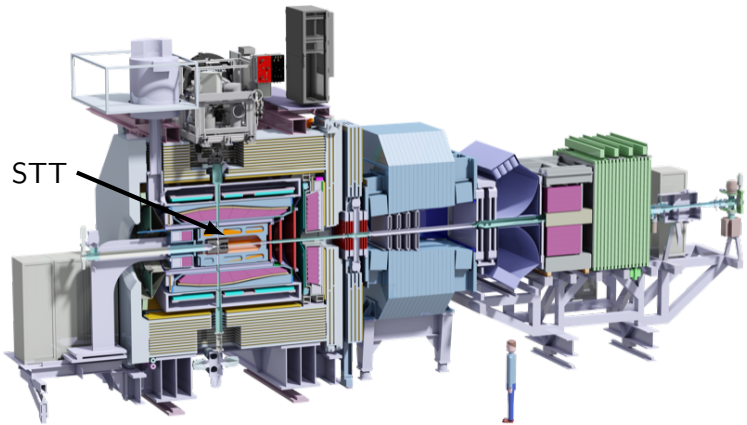
- Tracking at PANDA demanding task
- Challenging topologies with secondary vertices (e.g. D mesons)
- Flexible algorithms for track and event reconstruction needed
- Good benchmark: hyperons

Why hyperons?

- Complex topology
- Displaced vertices
- Intersecting tracks
- Different subdetectors for different tracks

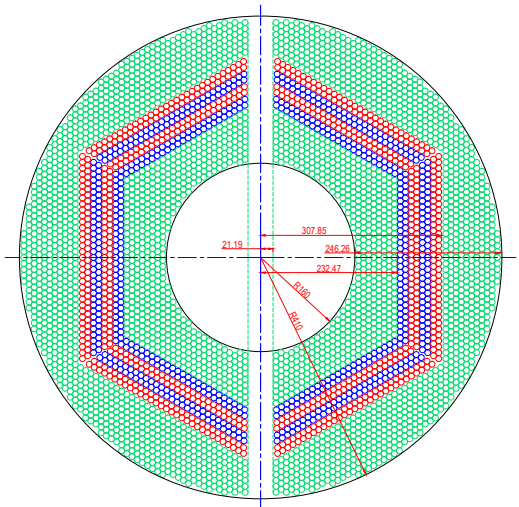


The PANDA detector

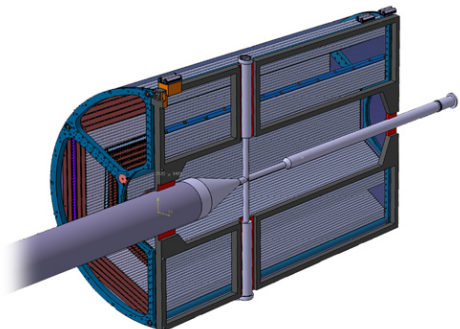


- Nearly 4π coverage
- Event rates up to 20 MHz
- Continuous $\overline{p}$ beam
- Online reconstruction
- Software-based event filtering

Initial focus: Straw Tube Tracker (STT)



- 4224 straws
- 19 axial layers (green)
- 8 stereo layers ($\pm 3^\circ$ blue/red) for z-reconstruction



DyTER - Dynamic Track and Event Reconstruction

Idea

- Focus on displaced vertices (hyperons)
- Break away from traditional event-based reconstruction
- Generate tracks and events dynamically from continuous data stream
- Use track and vertex information in event building

→ Track reconstruction and event building as an interdependent process

- Write highly modularised code

Pattern matching

- Pre-clustering (procedure suitable for FPGAs)
- Complement cellular automaton track finder
- Stand-alone track finder using machine learning

Pattern matching

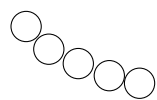
Pattern Matcher: Concept

Data preparation

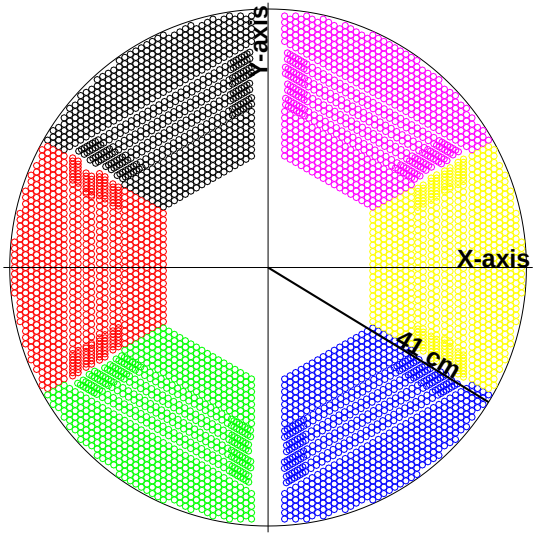
- Divide STT into 6 sectors
- Simulate training data
- Extract hit information
- Momenta from generated tracks
- Merge similar patterns
- Start matching

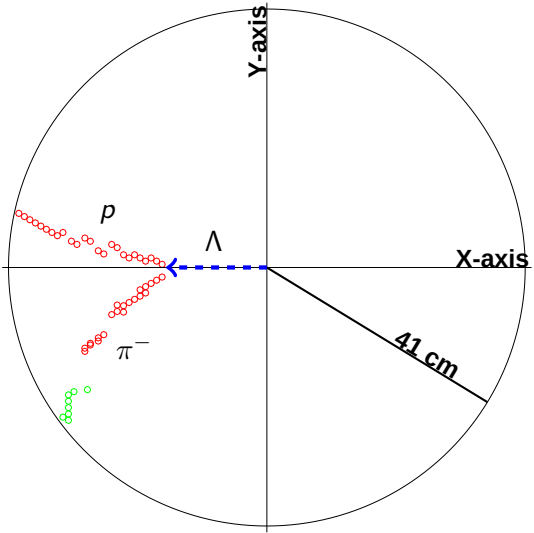
Closer look: Pattern

- tubeIDs
- momenta
- timeStamps
- sectorID
- count

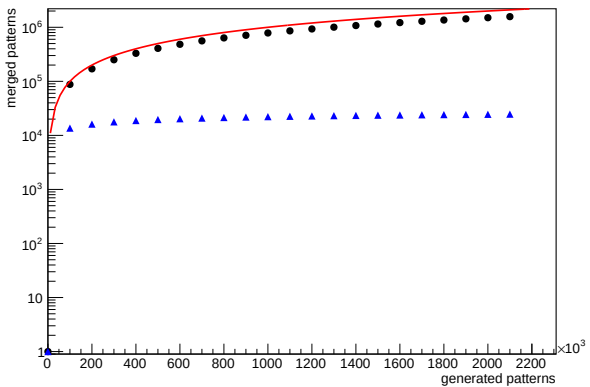


Pattern Matcher: Concept





Pattern Matcher: Merge similar patterns



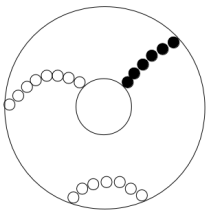
- black: merged identical patterns only
- blue: merged 90% similar patterns
- similar patterns saturate $\lesssim 100000$ (< 100 MB)

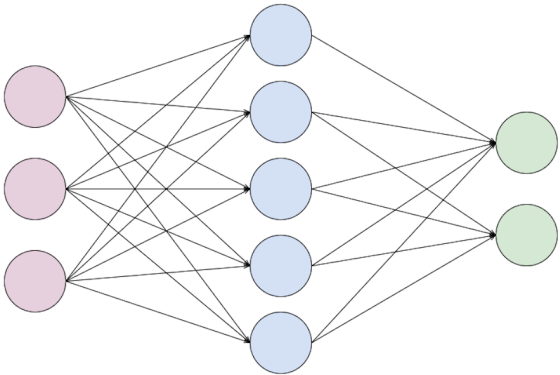
Track reconstruction with neural networks

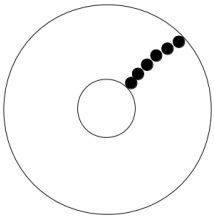
Adam Hedkvist
Arvi Jonnarth

Problem description

- Simplified concept for first prototype
- Identify 1 specific particle track in $\overline{\Lambda}\Lambda$ event topology
- Extract physical observable







Method

- Two neural networks
  - Pattern recognition
  - Momentum regression
- Trained with Monte Carlo simulations
- Implemented in Matlab

Pattern recognition network

- Identify proton track candidates
- Input: Tube IDs of raw STT hits
- Output: Specific particle track
- 4 hidden layers

Momentum regression network

- Determine p_x, p_y of track candidates
- Input: Results of pattern recognition network
- Output: 2 momentum components
- 6 hidden layers

Visualisation

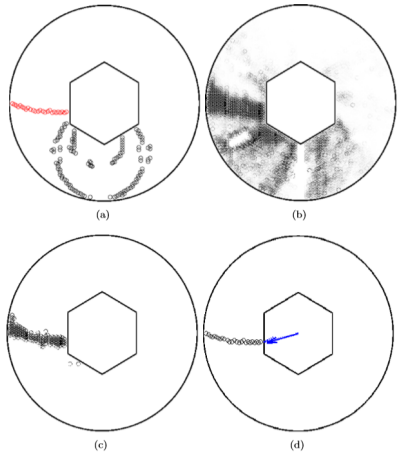
a Input to pattern recognizer

b Raw output of network

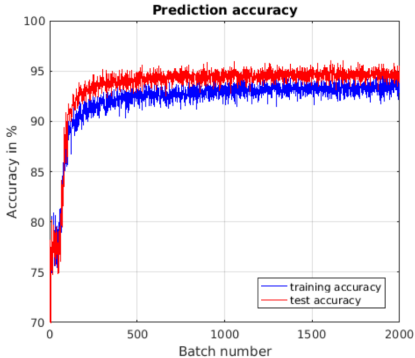
c Thresholded output (> 0.95)

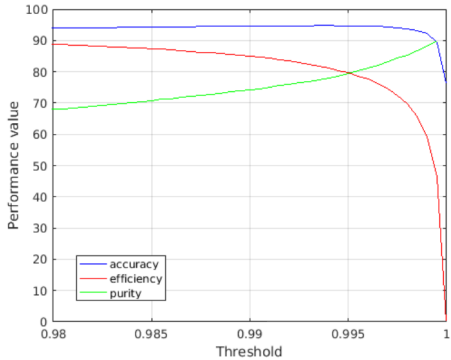
d Thresholded and filtered output

- Figure (d) also contains predicted momentum (solid arrow) and true momentum (dotted arrow)



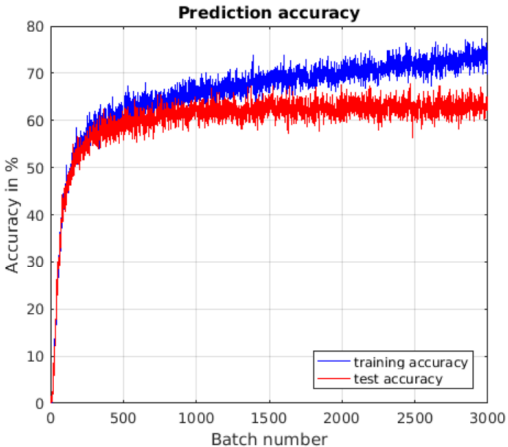
Prediction accuracy





- High accuracy
- Few signs of overfitting

Momentum regression: Prediction accuracy



- Condition: within 5% of original momentum
- Good accuracy
- Little overfitting

Summary

- Track and event reconstruction at PANDA challenging
- Development of pattern matching prototype
- Promising first results with neural networks
  - Pattern recognition
  - Momentum regression

Outlook

- Explore FPGA application for pattern matching
- Port neural networks to Python / TensorFlow
- Extend to all types of tracks
- Expand to all tracking detectors

Thank you for your attention!

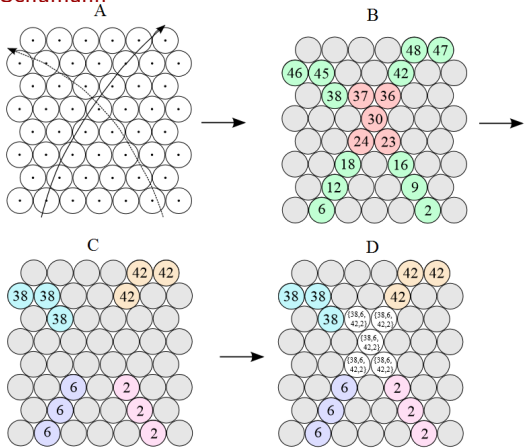
Backup slides

Neural network features

- Pattern recognition
- Momentum regression
- Evaluation time: 1.5ms per event for pattern recognition, 0.35ms per event for momentum regression
- Memory footprint $\sim$ 40 MB
- 64-bit floating point precision

SttCellTrackFinder

J. Schumann



- Mark cell as active if it corresponds to hit
- Assign unique ID to unambiguous cells (i.e. ≤ 2 neighbours)
- Set ID of cells to minimum of itself and neighbours
- Ambiguous cells: Include all IDs of neighbours