Upgrade of the Neural Network Track Trigger for Belle II

Simon Hiesl Christian Kiesling Kai Unger Sebastian Skambraks Timo Forsthofer

Master student LMU

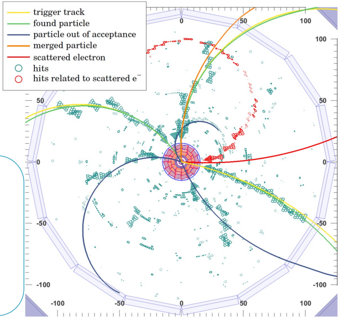
07.03.2024

Members of the Belle II Trigger Group

KIT ITIV
- Marc Neu
- Kai Unger
- Jürgen Becker

KIT ETP
- Lea Reuter
- Greta Heine
- Slavomira Stefkova
- Torben Ferber

MPI & LMU & TUM
- Christian Kiesling
- Felix Meggendorfer
- Simon Hiesl
- Timo Forsthofer
- Alois Knoll

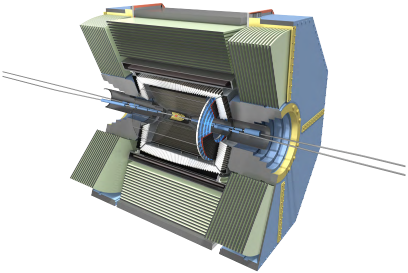


Focus of the MPI/LMU/TUM group:

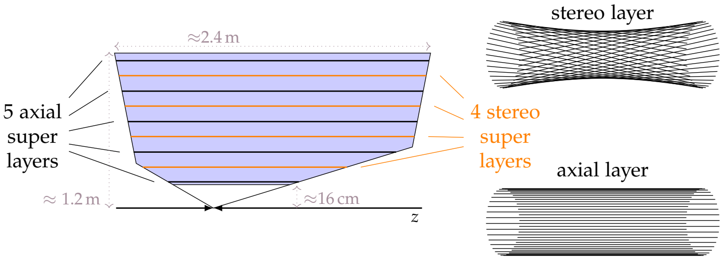
Track Triggers

The Central Drift Chamber (CDC) of Belle II

The Belle II Detector

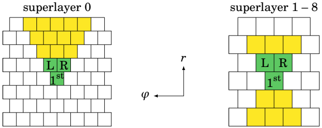


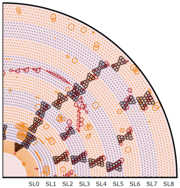
The CDC

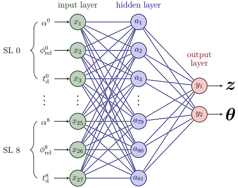


- TS = Wire pattern compatible with a crossing track → 2336 TS in 9 Super Layer (SL)

Track Segment (TS)



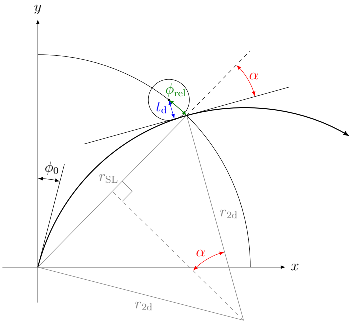


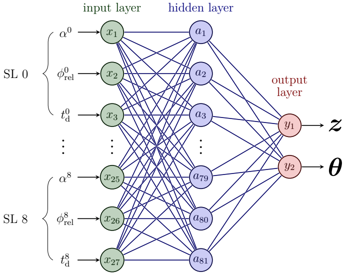


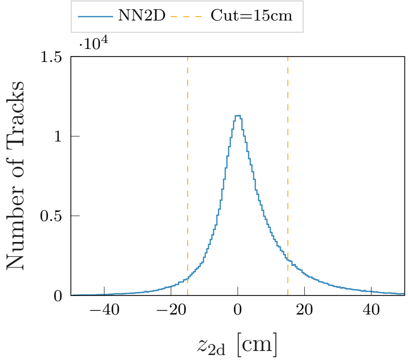
The L1 Neural Network Trigger

z-Vertex and polar emission angle prediction with neural network:

- 2D track + Stereo TS $\Longrightarrow$ $z + \theta$ prediction
- One hidden layer with 81 nodes







$\Longrightarrow$ z-cut of $\pm 15\,\mathrm{cm}$ used

Latency budget of only 5μs for the complete L1 trigger $\rightarrow$ Only **300ns** for the neural computation

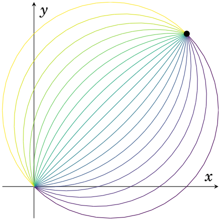
Preprocessing of the Network Input: Track Finding

Which TS belong to a real track?

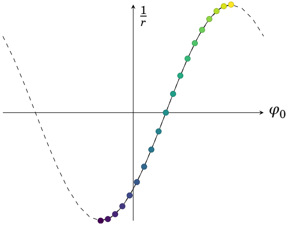
TS selection using a two-dimenisonal Hough transformation:

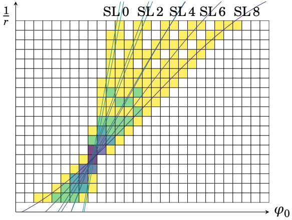
- Axial hit in CDC (TS) gets transformed to a curve in parameter (Hough) space
- Intersection point yields the track parameters ϕ and $r_{\mathrm{2d}} \propto p_{\mathrm{T}}$

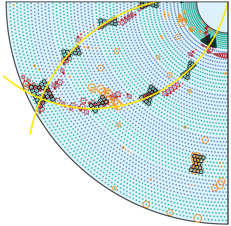
geometrical space



parameter space





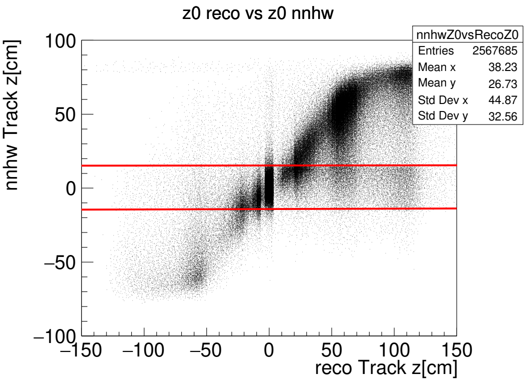


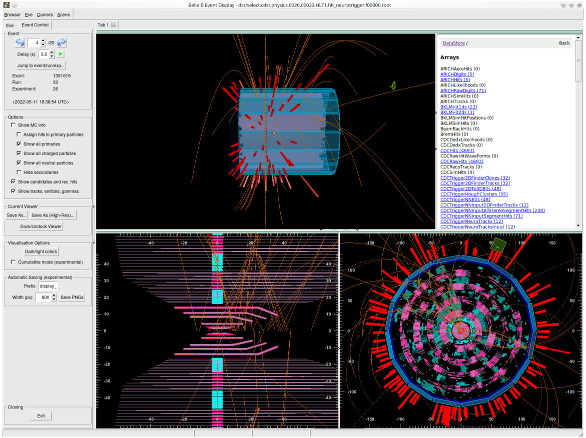
$\Longrightarrow$ 2D track candidate

The Neuro Trigger has been running since January 2021 years with remarkable success.

Problems with the L1 Neural Network Trigger

- "Feed-Down" effect: Background tracks → Vertex tracks
- Many Fake-Tracks with high Background



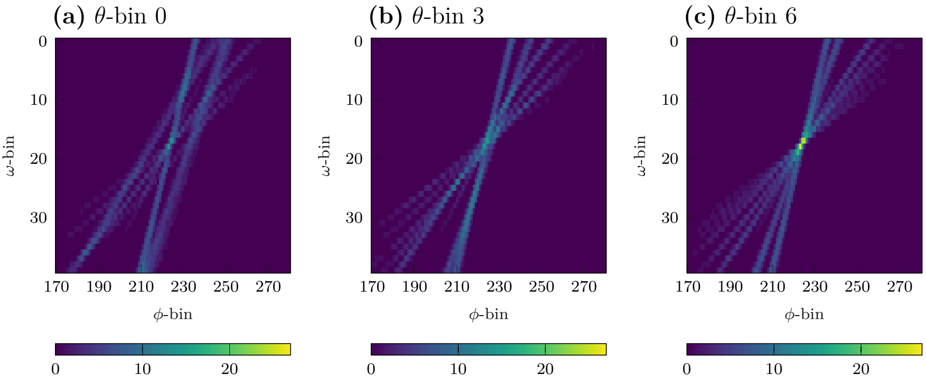


Extension to 3D: The 3DFinder

New curve parameter: Polar angle $\theta \implies$ 3D-Hough space

- 9 bins in $\theta \in [19, 140]^\circ$, 384 bins in $\phi \in [0, 360]^\circ$, 40 bins in $\omega \propto q \cdot p_\mathrm{T}^{-1}$, $p_\mathrm{T} \in [0.25, 10]\,\mathrm{GeV}/c$

Vertex assumption: The track originates from $(x, y, z) = (0, 0, 0)$ (IP)



$\implies$ Intersection point yields ω, ϕ and θ

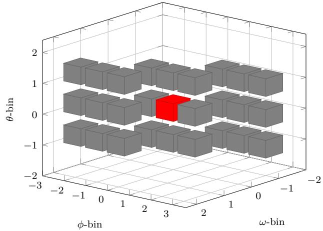
Clustering Algorithm in 3 Dimensions

Original algorithm: DBSCAN $\rightarrow$ Difficult to implement on an FPGA (non-deterministic length $\implies$ latency not fixed)

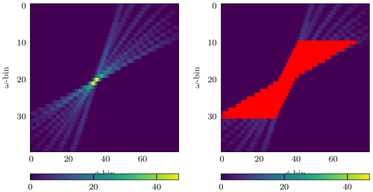
Update: **Fixed Clustering**
Three steps, repeated `iterations` times:

- Step 1: Global maximum search on Hough space
- Step 2: A fixed shape is put around the maximum
  - The weights in this shape are added up (total weight)
  - If total weight $\geq$ `mintotalweight` and peak weight $\geq$ `minpeakweight` the cluster is saved
  - All hits (TS) are extracted and have to pass two TS cuts
- Step 3: Cells around the global maximum are set to zero ("Butterfly-Shape" cutout)

Fixed shape:

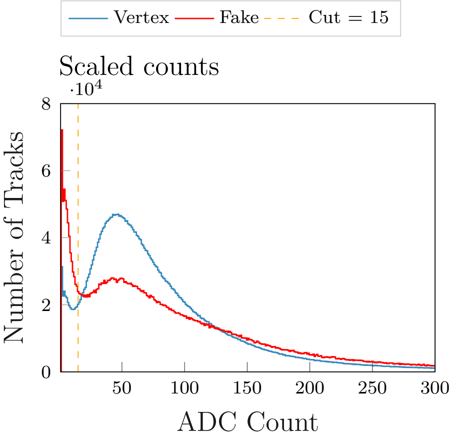


(a) Complete Cluster **(b)** Cutout

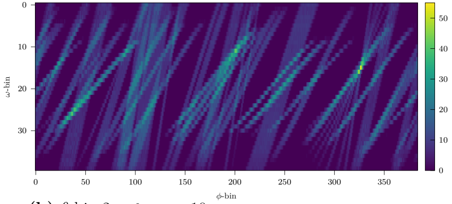


Real Data Analysis

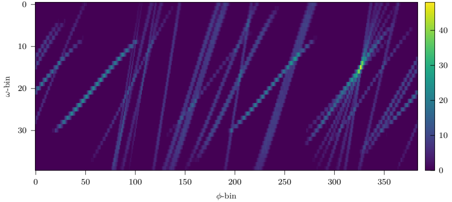
- Very high backgrounds were observed in the last experiment (due to high luminosity)
- The Hough spaces contain a lot of background track segments



(a) θ-bin 2: No `adccut`



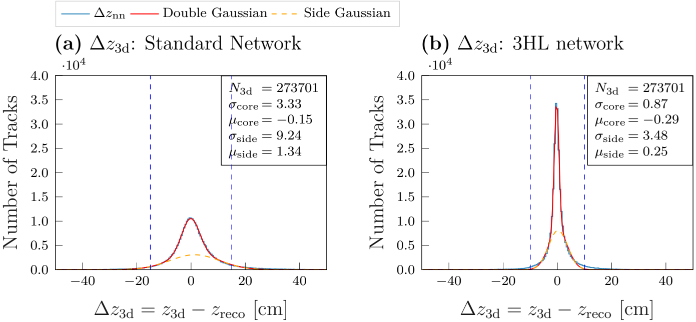
(b) θ-bin 2: `adccut`=10



$\Longrightarrow$ Reduction of noise using a cut on the ADC count

FPGA Implementation

- Present implementation → 2DFinder and Neuro Trigger on separate FPGA boards (2 UT3)
- New implementation → 3DFinder and Neuro Trigger on the same (new) FPGA board (1 UT4)
- The available latency is increased to **700ns**
- Neural networks with three or four hidden layers are possible



$\Longrightarrow$ Cut reduction from $\pm 15\,\mathrm{cm}$ to less than $\pm 10\,\mathrm{cm}$ possible due to better resolution

(see presentation by Timo Forsthofer)

Efficiency on Real Single Track Events

- Hit to cluster relation:
  - All hits in a cluster are considered
  - The largest weight distribution for each SL is used
- Cut on the number of axial and stereo SL hits (for background reduction)

Efficiency for single track events: Cut at $\pm 10\,\mathrm{cm}$

`adccut`	Efficiency 3D	Efficiency 2D
No Count	94.1%	94.0%
10 Counts	96.3%	95.3%

Fake-Rate for all found tracks:

`adccut`	Fake-Rate 3D	Fake-Rate 2D
No Count	13.1%	31.6%
10 Counts	5.8%	13.5%

But: Neural network not trained for 3D candidates at the moment (see presentation by Timo Forsthofer)

Conclusions and Next Steps

Using the 3DFinder has multiple advantages over the present 2D model with additional stereo TS selection:

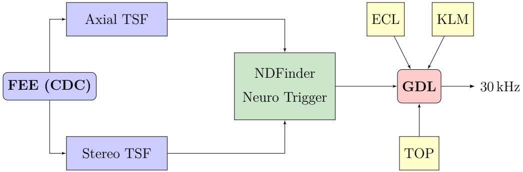
- Automatic suppression of tracks outside the interaction region (candidates implicitly originate from the IP)
- Better track segment selection $\implies$ Better resolution
- Implementation of track finding and network computation on the same FPGA board $\implies$ Deep neural networks
- Smaller Fake-Rate
- Higher efficiency

The next steps are:

- Implementation of the 3D Hough method on UT4 FPGA boards (Kai Unger)
- Improved neural network architecture (Timo Forsthofer)
- Retraining with unbiased data from the new data taking, which just has started

Backup

(a) New Pipeline



(b) Current Pipeline