Upgrade of the Neural Network Track Trigger for Belle II

Simon Hiesl Christian Kiesling Kai Unger Sebastian Skambraks Timo Forsthofer

Master student LMU

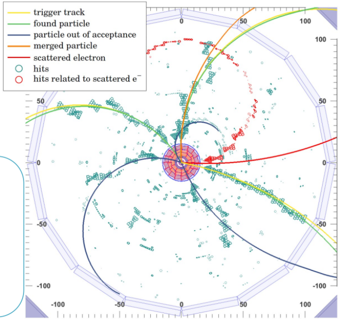
01.10.2024

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

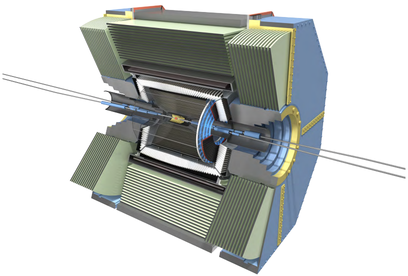


Focus of the MPI/LMU/TUM group:

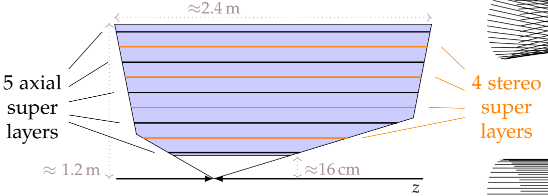
Track Triggers

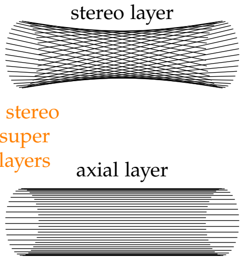
The Central Drift Chamber (CDC) of Belle II

The Belle II Detector



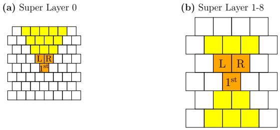
The CDC



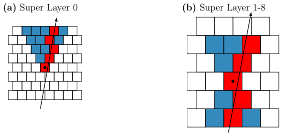


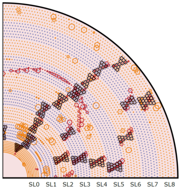
- TS = Wire pattern compatible with a crossing track → 2336 TS in 9 Super Layer (SL)

Track Segment (TS)



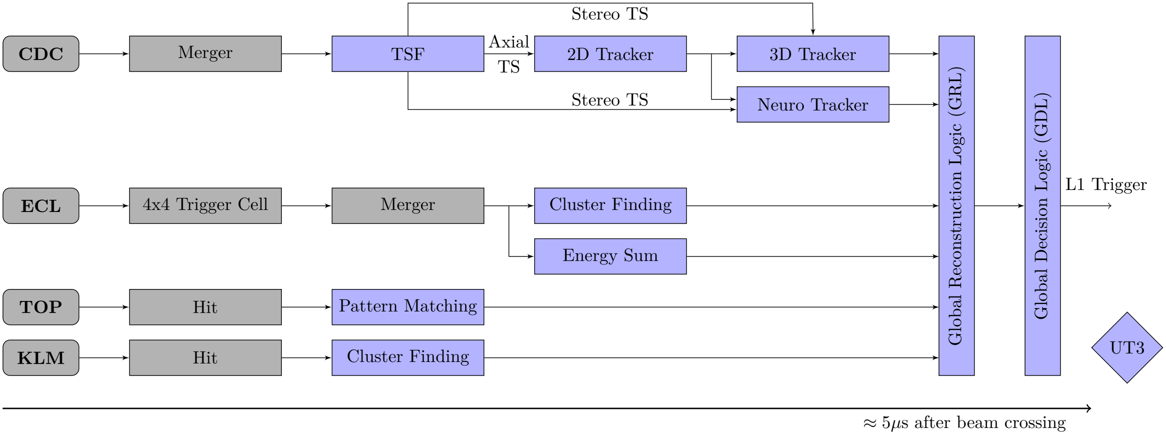
Track Segment Hit





The Old L1 Trigger Pipeline

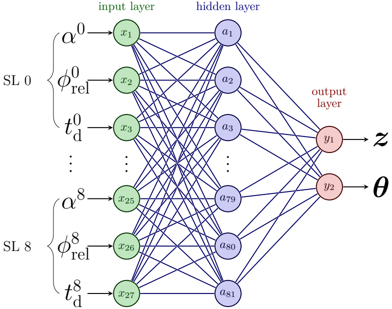
- Present implementation → 2DFinder and Neuro Trigger on separate FPGA boards
- The available latency for the Neuro Tracker is just **300ns**

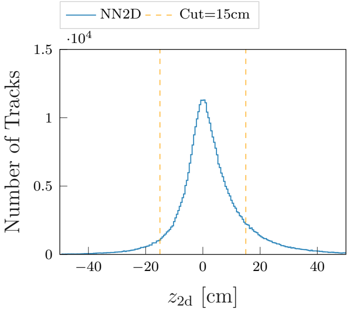


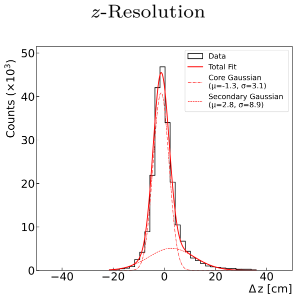
The Neural Network

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

2D track + Stereo TS $\implies$ $z + \theta$ prediction (Current Network: One hidden layer with 81 nodes)







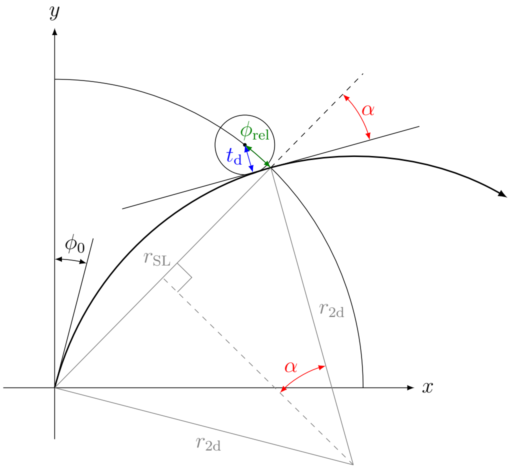
$\implies$ z-cut of $\pm 15\,\text{cm}$ used

Single-Track-Trigger: Low multiplicity trigger activated if the momentum is above $0.7\,\text{GeV}$ (S. Bähr et al., arXiv:2402.14962)

$$p[\,\text{GeV}] = \frac{1}{\omega[1/\text{m}]\sin(\theta)\, 0.3\, B[\text{T}]} \geq 0.7\,\text{GeV}$$

Preprocessing for the L1 Neural Network Trigger

From the track finding (Hough transformation) we get: $\boxed{\omega = \pm 1/r_{\mathrm{2d}} \text{ and } \phi_0}$



With the TS information

$$\phi_{\mathrm{wire}},\ n_{\mathrm{wire}},\ r_{\mathrm{SL}},\ \sigma_{\mathrm{LR}},\ t_{\mathrm{d,wire}}$$

we can calculate:

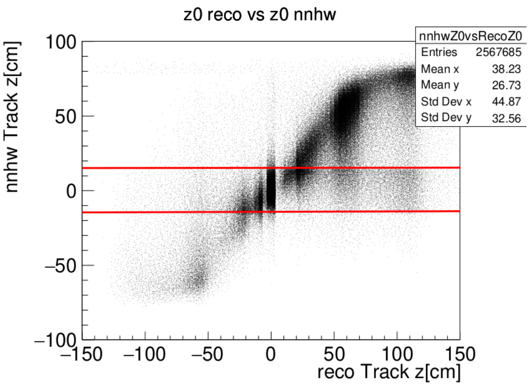
$$\alpha = \arcsin\left(\frac{1}{2}\frac{r_{\mathrm{SL}}}{r_{\mathrm{2d}}}\right)$$

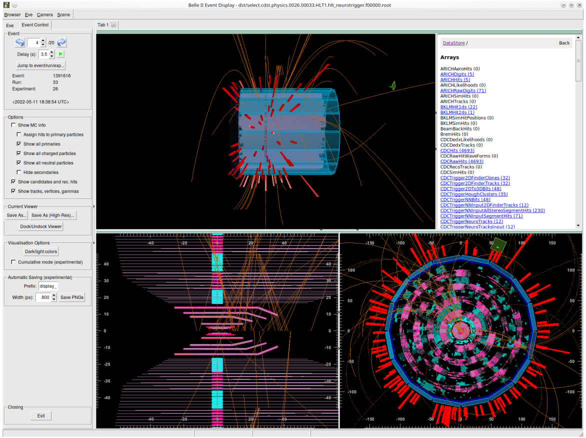
$$\phi_{\mathrm{rel}} = \phi_{\mathrm{wire}} - n_{\mathrm{wire}} \cdot \left(\frac{\phi_0 - \alpha}{2\pi}\right)$$

$$t_{\mathrm{d}} = \sigma_{\mathrm{LR}} \cdot (t_{\mathrm{d,wire}} - t_{\mathrm{d,min}})$$

Problems with the L1 Neural Network Trigger

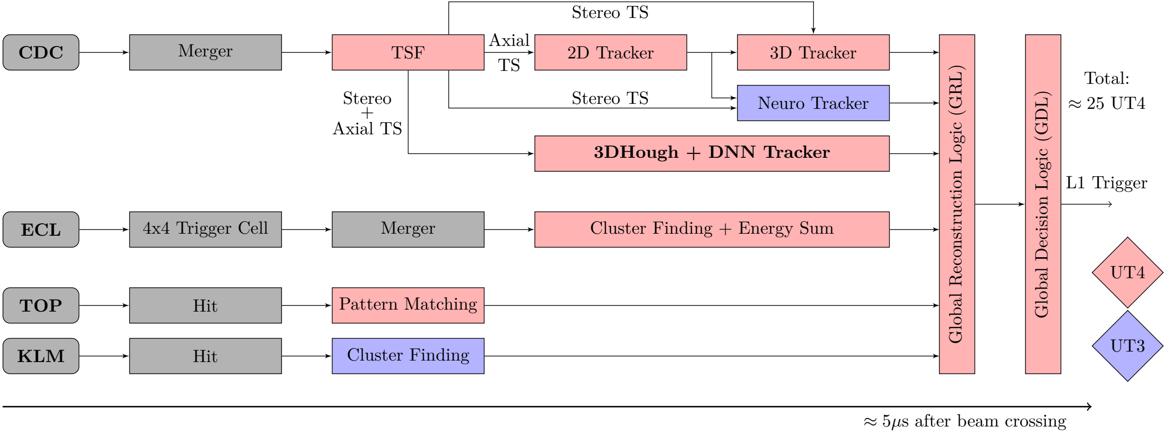
- "Feed-Down" effect: Background tracks → Vertex tracks
- High number of 2D track candidates from the 2DTracker when the background is high
  $\Longrightarrow$ Many Fake-Tracks using stereo background hits





The New L1 Trigger Pipeline

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

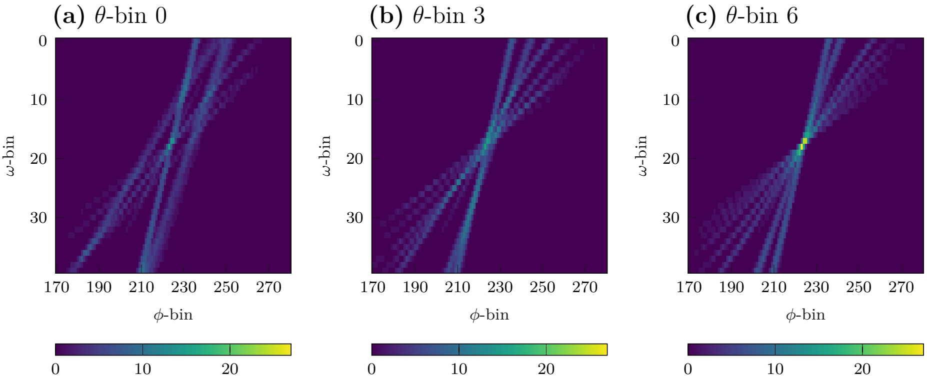


Extension to 3D: The 3DFinder

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 $\omega \propto q \cdot p_T^{-1}$, $p_T \in [0.25, 10]$ 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

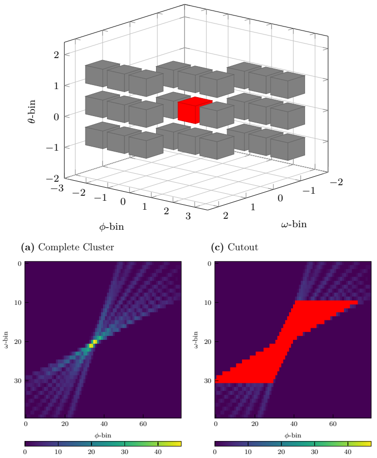
Original algorithm (Sebastian Skambraks): DBSCAN
Impossible to implement on an FPGA
(non-deterministic length $\implies$ latency not fixed)

Update: A new clustering option implemented in basf2:
Fixed Volume Clustering

Three steps, repeated **`iterations`** times:

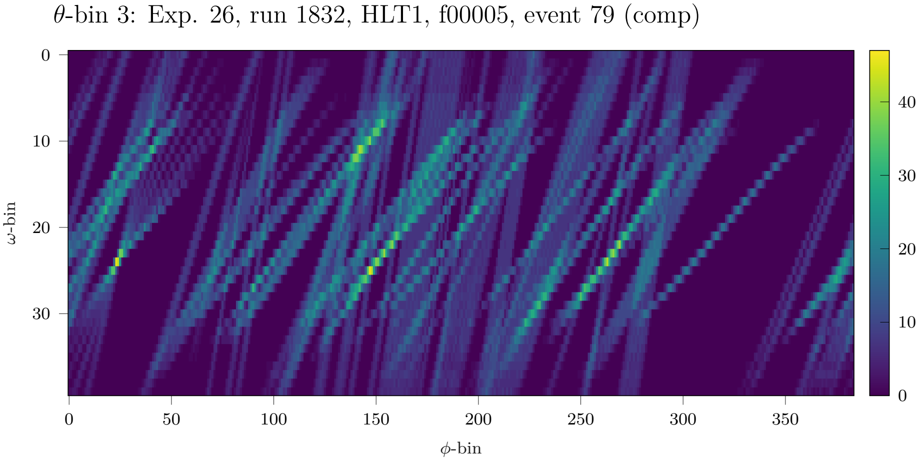
- Step 1: Global maximum search on Hough space
- Step 2: A fixed shape is put around the maximum
  - The weights in this shape are added up (total weight)
  - If total weight $\geq$ **`mintotalweight`** and peak weight $\geq$ **`minpeakweight`** the cluster is saved
  - All hits (TS) are extracted and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

Fixed shape:



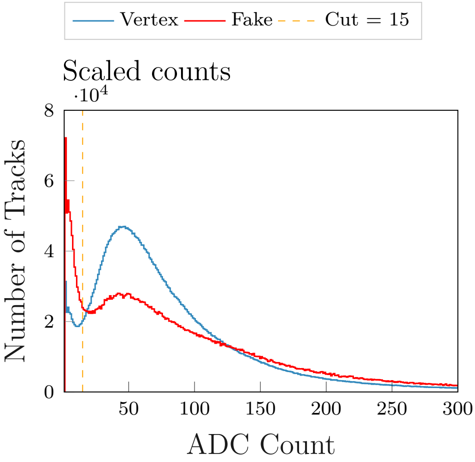
Real Data Analysis: Experiment 26 Single Track Events

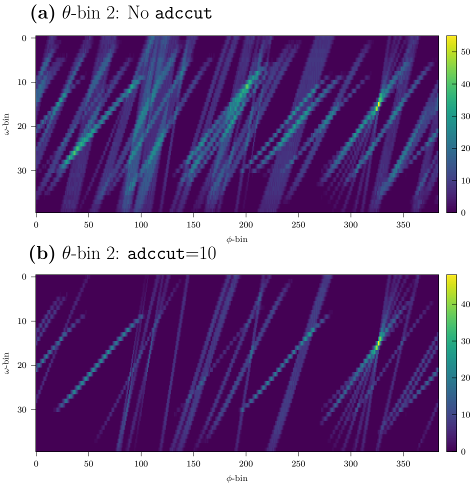
- Very high backgrounds were observed shortly before the long shutdown
- The Hough spaces contain a lot of fake track segments



Introducing a Cut on the ADC Count

- A cut on the ADC count of the wires has been made possible by the new UT4 boards

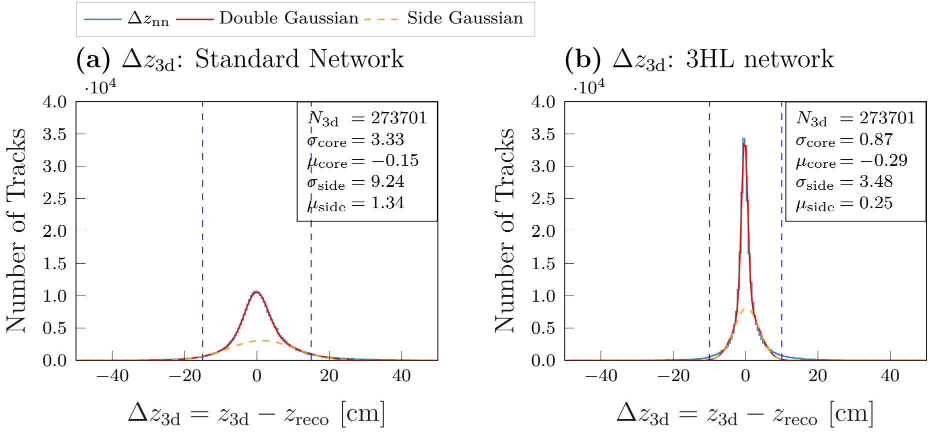




$\Longrightarrow$ Reduction of noise using a cut on the ADC count

FPGA Implementation

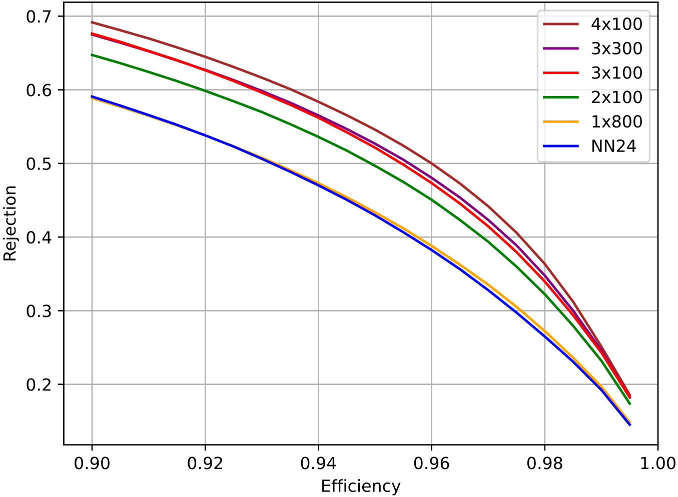
Timo Forsthofer (master's thesis, next presentation):



$\Longrightarrow$ Cut reduction from ± 15 cm to less than ± 10 cm possible due to better resolution

Rejection vs. Efficiency of Different Network Architectures

Timo Forsthofer (master's thesis, next presentation):



- Deeper neural networks are considerably better than wide neural networks
- The performance is restricted by the limited input (Only 9 TS at the most)

Efficiency on Real Single Track Events

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

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

Fake-Rate for all found tracks:

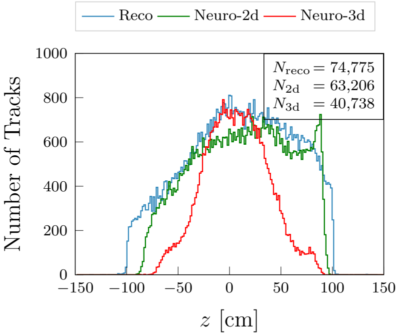
`adccut`	Fake-Rate 3D	Fake-Rate 2D
No Count	13.1%	31.6%
10 Counts	5.8%	13.5%

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



Currently:

- 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

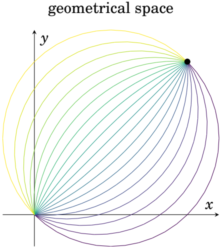
Backup

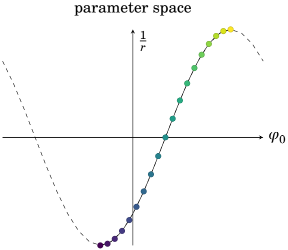
Preprocessing of the Network Input: Track Finding

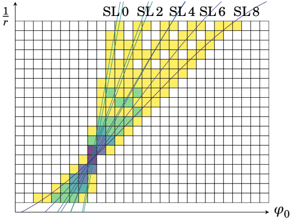
Which TS belong to a real track?

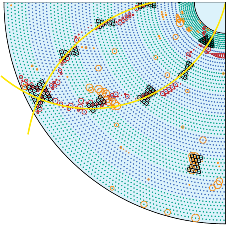
TS selection using a two-dimenisonal Hough transformation:

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







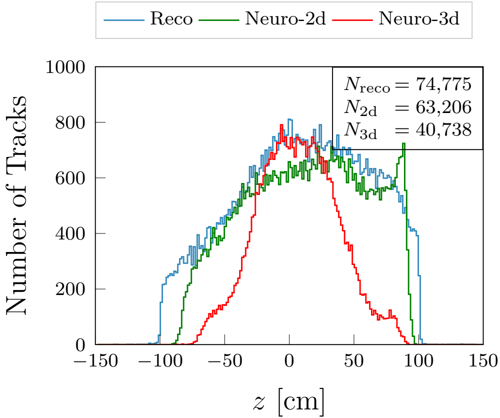


$\Longrightarrow$ 2D track candidate

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

Background Suppression of Monte Carlo

- Promising results from **Particle Gun** single tracks ($z_{\text{reco}} \in [-100, 100]$ cm)
- The number of found tracks falls quickly with large $|z|$



Parameter optimization did not sufficiently solve the following problems:

- Resource heavy on the hardware, non-deterministic length, difficult to implement
- Clusters can get very large $\implies$ Very bad resolutions for some tracks
- Very high fake rates when using nominal phase-3 background (3 fakes for 1 signal)

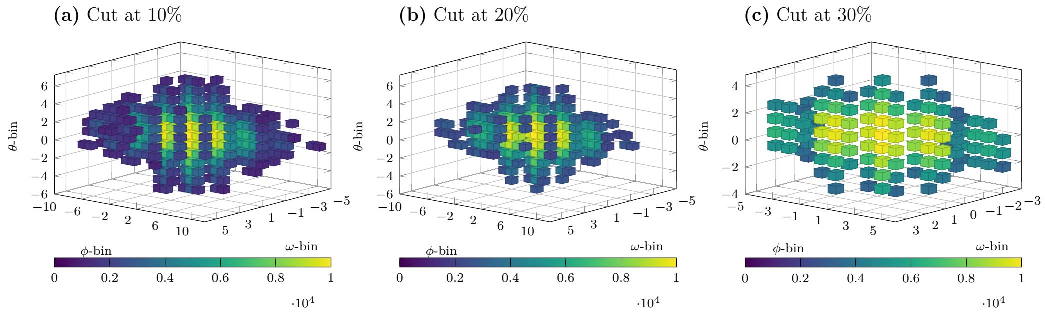


Figure: The cut percentage defines how often (on average) a cell was present in the clusters.

Cluster Statistics: Average cell weights

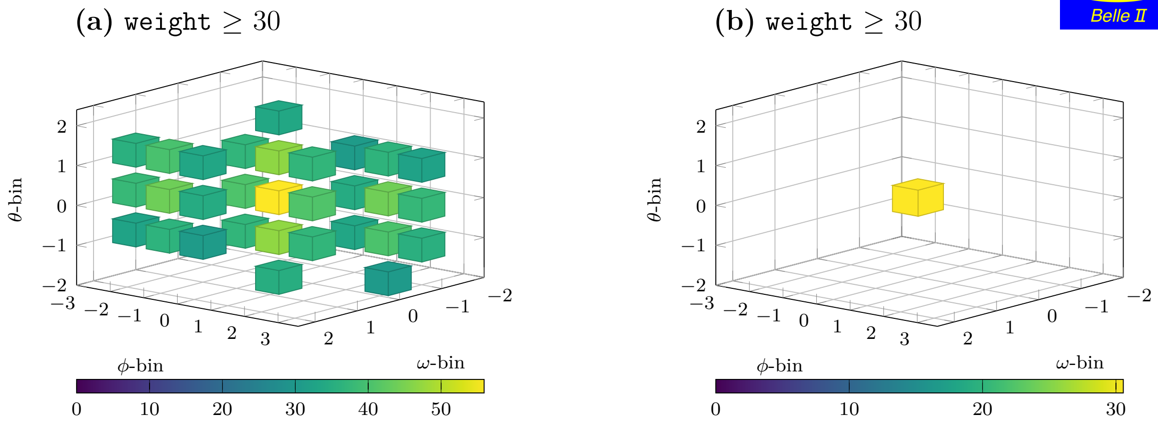
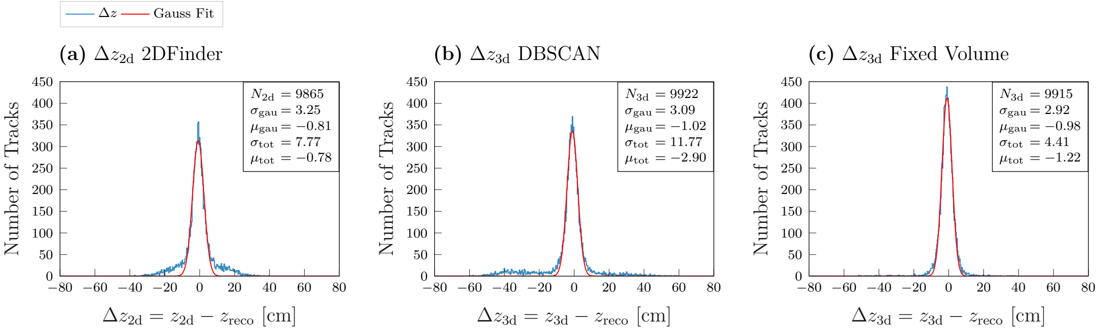


Figure: The average cluster weights (10000 clusters) above a weight of 30: **(a)** Real tracks with nominal phase-3 background, **(b)** Fake tracks from only nominal phase-3 background.

New idea: Fixed cluster shape clustering

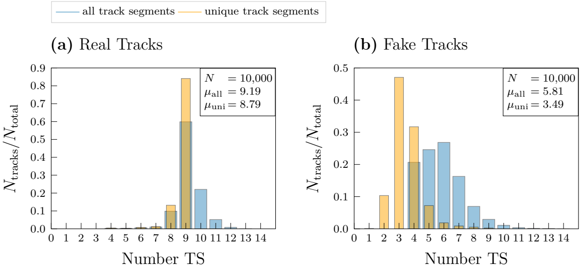
Monte Carlo Single-IP Tracks

- 10 000 single IP tracks
- The efficiency and resolution are high with the fixed volume clustering (plot **(c)**):



Nominal Phase-3 Background Studies

- Very high fake rates are observed $\implies$ Solution: Cut on the number of hit super layers



Background rate per event (10 000 neutrinos):

Clustering	`minhits`	`minsuper`	N_{3d}^{all}
DBSCAN	4	0	29 424
DBSCAN	6	0	11 350
Fixed Volume	5	5	783

- Default DBSCAN: **290%**
- Fixed Volume with `minsuper` = 5 cut: **6%**