Belle II Germany 2025

Upgrade of the Belle II First-Level Neural Track Trigger

Simon Hiesl Christian Kiesling Kai Unger Timo Forsthofer

PhD student LMU

09.09.2025

- New Track Finding by 3D-Hough Transforms
- Deep Neural Networks on FPGAs for Track Parameter Estimation

Members of the Belle II Trigger Group (Germany)

Belle II

KIT ITIV

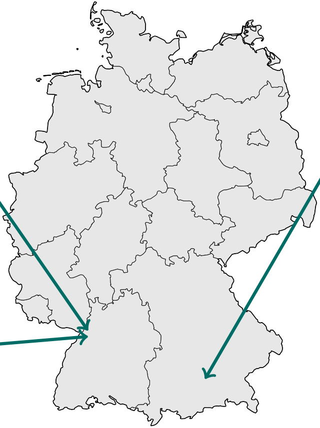
iTiV Institut für Technik der Informationsverarbeitung

- **Jürgen Becker**
- Marc Neu
- Kai Unger
- Timo Justinger
- Valdrin Dajaku
- Felix Mältzer
- Till Rädler

KIT ETP

ETP Institut für Experimentelle Teilchenphysik

- **Torben Ferber**
- Lea Reuter
- Greta Heine
- Isabel Haide



LMU/MPI

- **Christian Kiesling**
- Simon Hiesl
- Timo Forsthofer

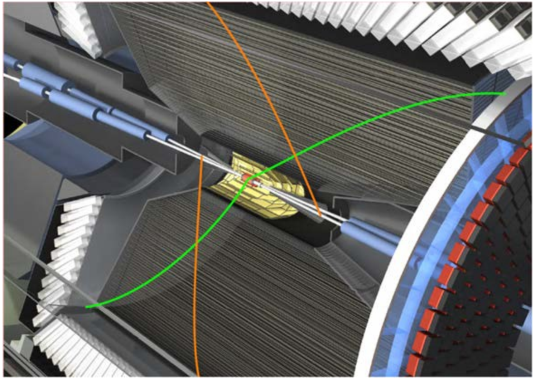
AI Trigger Projects:

- Calorimeter Trigger
- Track Trigger

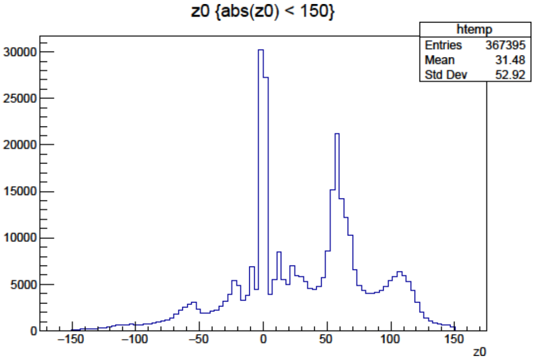
Munich: Focus on Track Triggers

Physical Motivation: z-Vertex Trigger

IP (green) and displaced (orange) tracks:



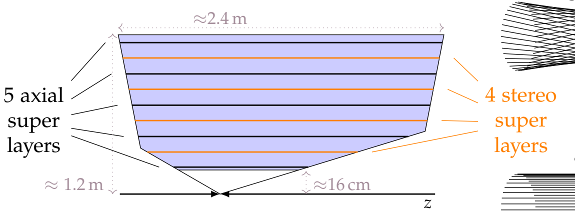
Offline z-vertex reconstruction:

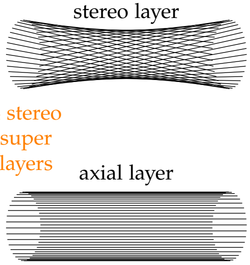


Data from 2019: **Before** z-trigger.

Only around 10-15% IP tracks! $\Longrightarrow$ **z-vertex trigger mandatory**

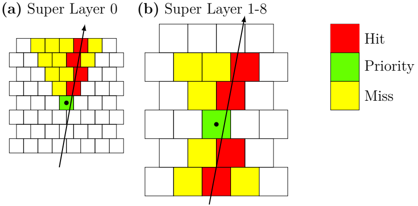
The Central Drift Chamber (CDC) and Track Segment Finder (TSF)



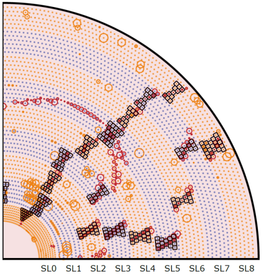


- 14336 sense wire in 56 layers
- TS = Wire pattern from track crossing
- TSF: Finds TS from CDC hits
- 2336 TS in 9 SL

Crossing Track:

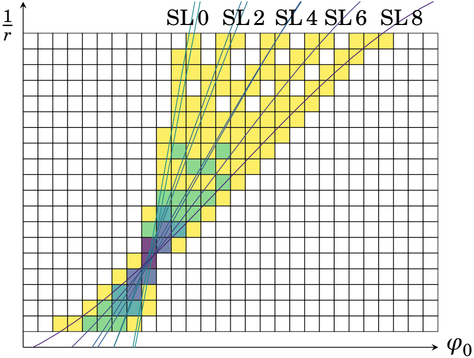


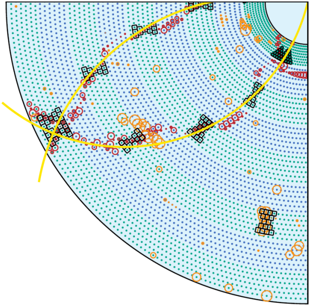
Example CDC Quadrant:



Current Level-1 Track Trigger: Track Finding Method

- Total latency $\approx 5\mu s \implies$ 3D Fitting *not possible*, requires:
  - Current track finding method: **2D Hough Transformation**
  - Current classification method: **Neural Network**





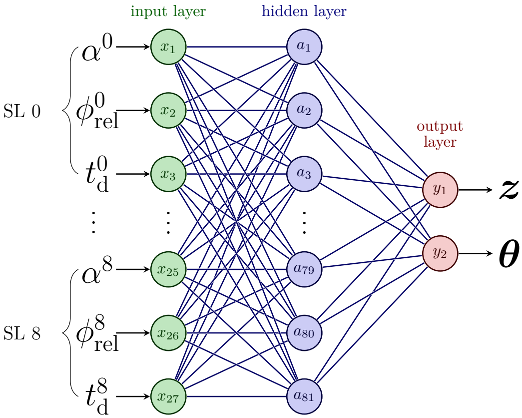
Track parameters: ϕ and $q/r = \omega$ from 2DHough track

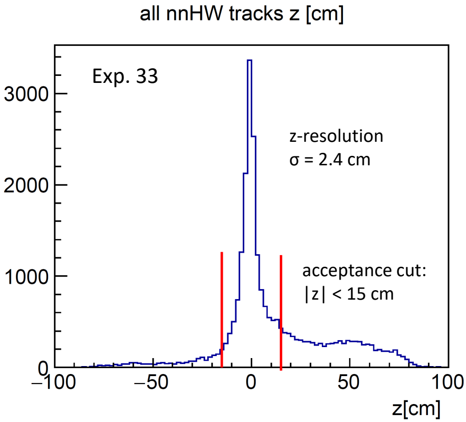
2D Track candidate: Axials only $\implies$ **Only 5 SLs** for track finding, no z-information

- Requires **4 out of 5** axial track segments (ATS)
- Requires **3 out of 4** stereo track segments (STS), have to be found **separately** (by statistical method through ϕ and drift time)

Current Level-1 Track Trigger: The Neural Network

- **z-Vertex and polar emission angle prediction with a neural network**
- 2D track + Stereo TS $\Longrightarrow$ z and theta estimation
- Current Network: One hidden layer, 81 nodes

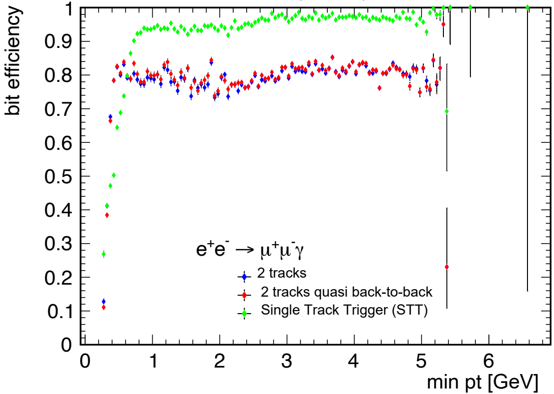




(Data from 2024)

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

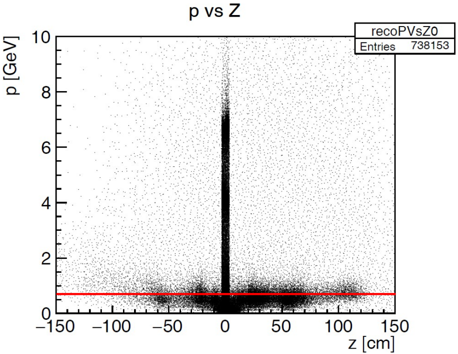
The Single-Track-Trigger (STT)



- The STT clearly outperforms the ≥2 track trigger
- Two trigger bits provided:
  - At least one track within z-cut
  - At least one track within z-cut and $p \geq 0.7\,\mathrm{GeV}$
  - (Published in "Nuclear Instruments and Methods in Physics Research A 1073 (2025) 170279", arXiv:2402.14962)

Single-Track-Trigger:
Activated if the track momentum is above 0.7 GeV

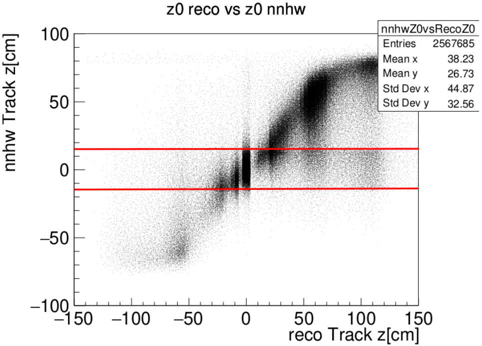
$$p[\,\mathrm{GeV}] = \frac{1}{\omega[1/\mathrm{m}]\,\sin(\theta)\,0.3\,B[\mathrm{T}]} \geq 0.7\,\mathrm{GeV}$$

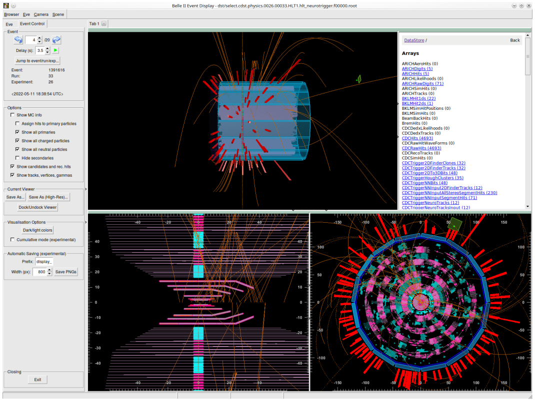


$\Longrightarrow$ Irreducible background from IP:
$e^+e^- \rightarrow e^+e^-e^+e^-$

Problems with the L1 Neural Network Trigger under High Backgrounds

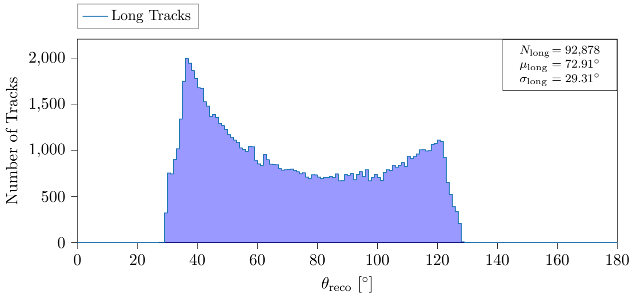
- “Feed-Down” effect: Background tracks $\rightarrow$ Vertex tracks
- High number of 2D track candidates from the 2DTracker when the background is high $\Longrightarrow$ Many Fake-Tracks using stereo background hits





Physical Motivation for New Trigger: Use all 9 Super Layers for Track Model

- Track Model: Axials **and** Stereos $\Longrightarrow$ 9 SL instead of 5
- Hence, more fake and background resistant
- Requires 3-dimensional Hough space, restricts vertex to $(0, 0, 0)$



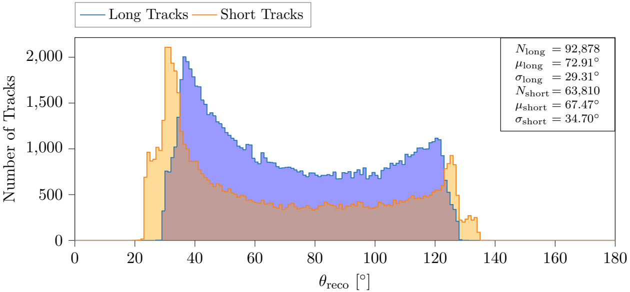
(Found single IP reconstructed tracks)

- Current polar acceptance:
  - At least **4/5** axial SLs
  - At least **3/4** stereo SLs

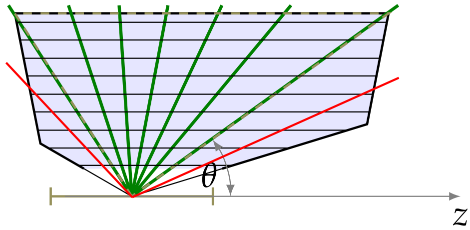
$\Longrightarrow$ Minimum 7/9 SLs

Tracks with Shallow Polar Angles: Increased Polar Acceptance

- **Extended (short track)** super layer topology → Minimum 5/9 SLs
  - At least **3/5** axial SLs
  - At least **2/4** stereo SLs



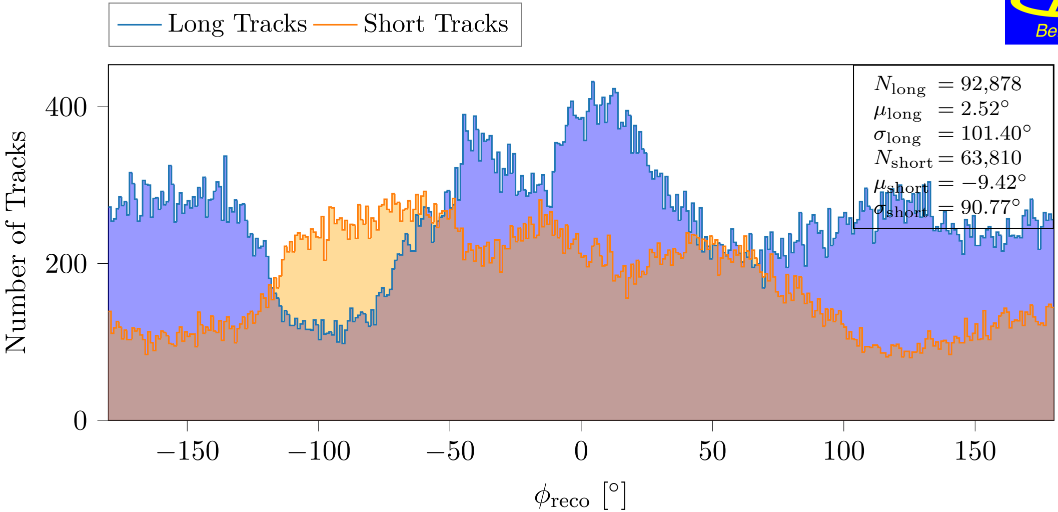
Full CDC polar acceptance:



SL number *decreases* with shallower θ

⟹ **Substantial gain** in efficiency, especially for shallow tracks

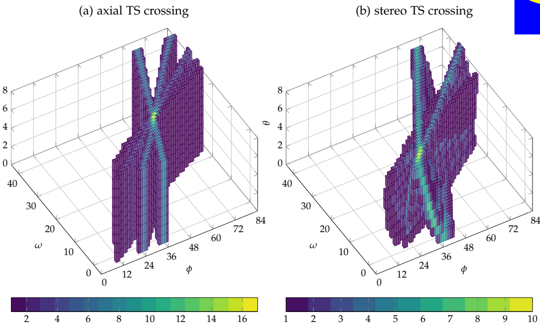
Short Track Acceptance: Compensating for CDC Inefficiencies

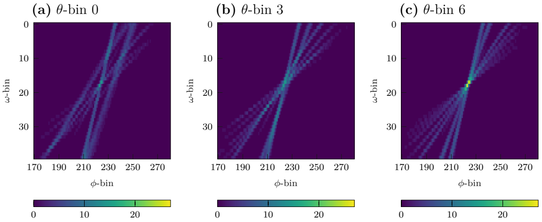


- Inefficient CDC region (around $\phi \approx -100°$) overrepresented in the short tracks

The 3DHough Space

- **Third** track parameter: Polar scattering angle θ
- In addition to: $\omega = \frac{q}{r} \propto \frac{1}{p_\mathrm{T}}$ and ϕ
- Track vertex **constraint**: $(x, y, z) = (0, 0, 0)$
- One TS $\widehat{=}$ 3D plane
- Selection of the stereo track segments from (helix) track model
- For geometrical reasons: $\boldsymbol{\cot(\theta)}$ instead of θ dimension
- Dimensions:
  - 9 bins in $\boldsymbol{\cot(\theta)}$
  - 384 bins in ϕ
  - 40 bins in ω

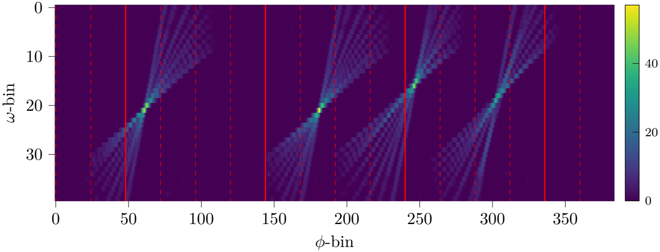




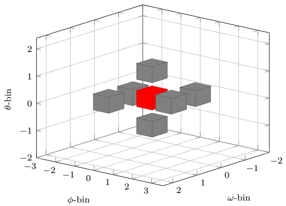
The 3DHough Finder Algorithm

1. Get TSs from TSF
2. Load the hit representations into the Hough space (**numeric** representations)
3. Scan the Hough space for maxima (global sector maximum)
4. Put an optimized fixed cluster shape around each maximum
5. Process each cluster: Center of Gravity (ω, ϕ, θ) and TS extraction
6. Apply only two cuts (on SLs): `min(Number ATS)` and `min(Number STS)`

Maximum finding (for hardware): 4 CDC Quadrants with 4 Sectors



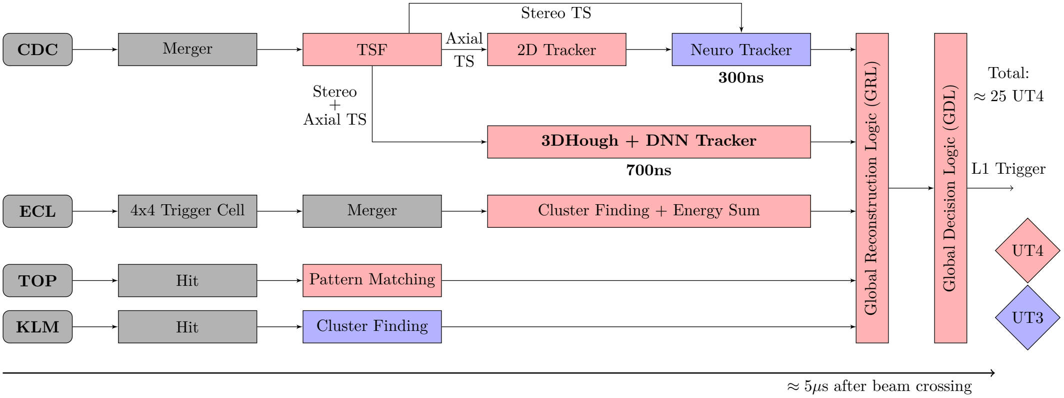
Fixed Cluster Shape



Red lines = 4 CDC quadrants with 4 sectors each: **Simultaneous** track finding on FPGA.

The L1 Trigger Pipeline

- New implementation → 3DFinder and Neuro Trigger on the **same** FPGA board (UT4)
- The available latency for the Neuro Tracker is increased from **300ns** to **700ns** (no in-between transmission)
- Deep neural networks with four hidden layers are possible

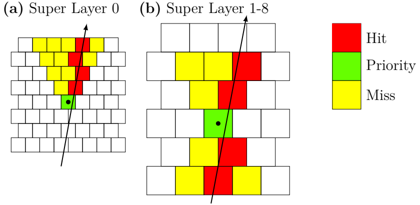


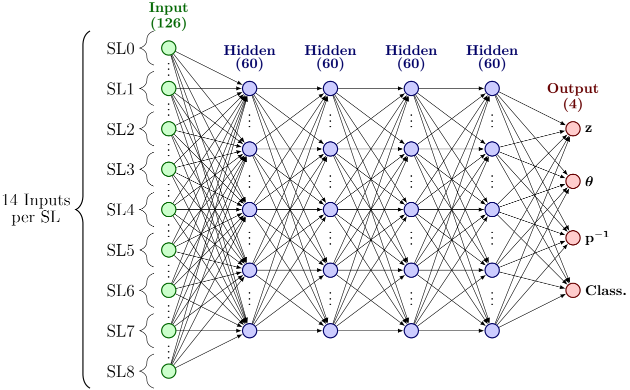
Deep Neural Network Architecture

→ Possible because of new trigger boards with Virtex Ultrascale 160/190 FPGAs

1. DNN: 4 hidden layers, 60 nodes each
2. Extended input (27 → 126)
3. Classification: IP/Non-IP
4. $1/p$ output node for STT
5. Trained on 3DFinder Tracks

Extended input:
Every wire passing ADC cut in TS
(1 hit, 0 no-hit)

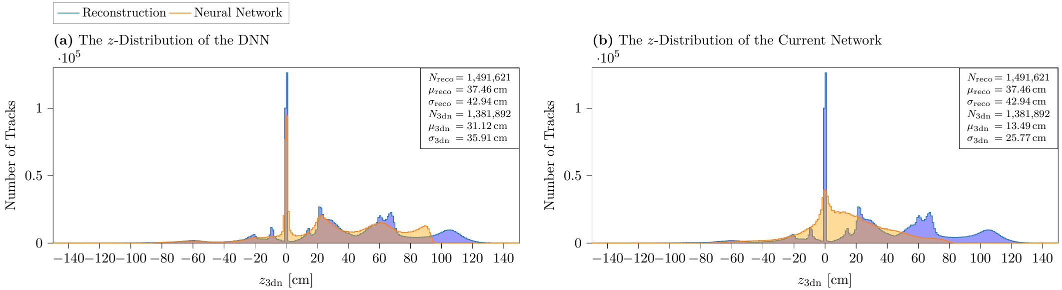




- 4 output nodes: z, θ, $1/p$, and classification
- Classification $\in [-1, 1]$, $+1 \equiv$ IP, $-1 \equiv$ Background

The Neural Network z Distributions

- Left: DNN (orange) and Reconstruction (blue)
- Right: Current Network (orange) and Reconstruction (blue)



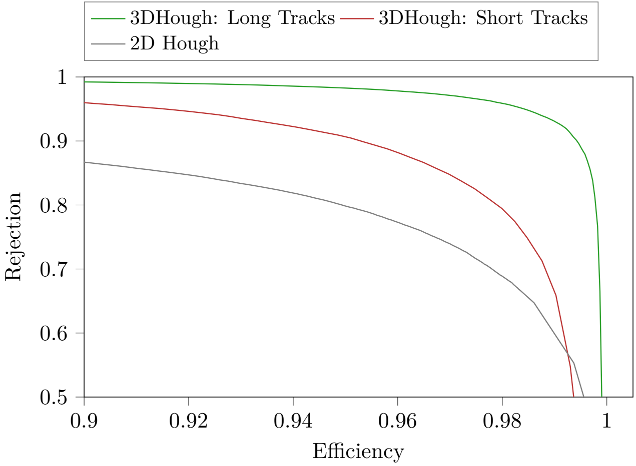
(a) The z-Distribution of the DNN

(b) The z-Distribution of the Current Network

- 3DHough + DNN: 96.7% efficiency (cut at 0.5 classification → 19.2% incorrect)
- 3DHough + Current Network: 89.0% efficiency (cut at $\pm 15\,\text{cm}$ → 69.8% incorrect)

Track Based Efficiency and Rejection

- Simulation of the new algorithms on real data from 2024 (high background)
- Long vs. short track efficiency of the 3DHough approach



- Long Track:
  $\texttt{n(ATS)} \geq 4$ and $\texttt{n(STS)} \geq 3$
- Short Track:
  $\texttt{n(ATS)} \geq 3$ and $\texttt{n(STS)} \geq 2$
  (but not long)
- Classification of 3DNeuro tracks:
  Cut on classification output node
- Classification of 2DNeuro tracks:
  Cut on z with $\pm 15\,\text{cm}$

Event Based Efficiency and Rejection: Recent High Background

- Select real data events with only a single reconstructed track (exp. 35 `neuroskim`)
- Differentiate the events in short and long track events
- Select background (all reconstructed $|z| > 1\,\text{cm}$) and fake track events (no reconstructed track in event)

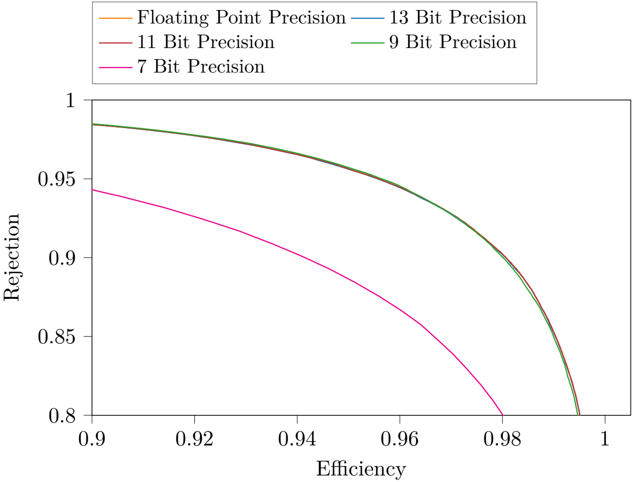
Data Subset	Total Events	2DN-Track	3DN-Track	2D-Class.	3D-Class.
Long Single Track Events	92 878	**94.11%**	**99.63%**	93.19%	98.23%
Short Single Track Events	63 810	**36.81%**	**96.34%**	35.97%	92.05%
Background/Fake Events	3 959 983	42.86%	69.62%	**12.01%**	**1.82%**

Considerably better track finding performance on short tracks for the 3DHough Neuro Trigger

- Classification 2DN: $|z| < 15$ cut
- Classification 3DN: Classification cut 0.0

Integerized Network Simulation: Fixed Bit Precision for FPGA

- For FPGA implementation:
  No floating point arithmetic, only integers
- Includes all calculations and inputs, weights, and tanh LUT.
- No difference for 13 and 11 bit.
- Only considerably worse performance at 7 bit.



Summary

New 3DHough method for track finding:

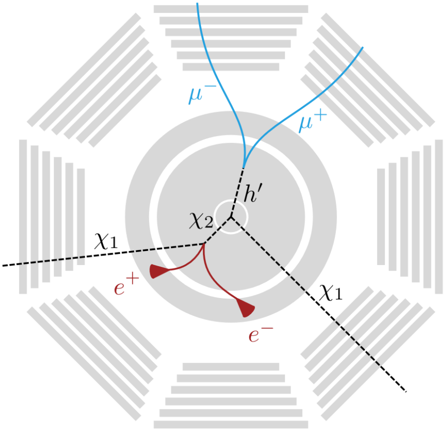
- More robust against fake tracks (9/9 TS used compared to 5/9 in 2D method)
- Increase of polar acceptance possible (3 axial + 2 stereo TS)
- DNN + additional output node shows superior performance

Currently:

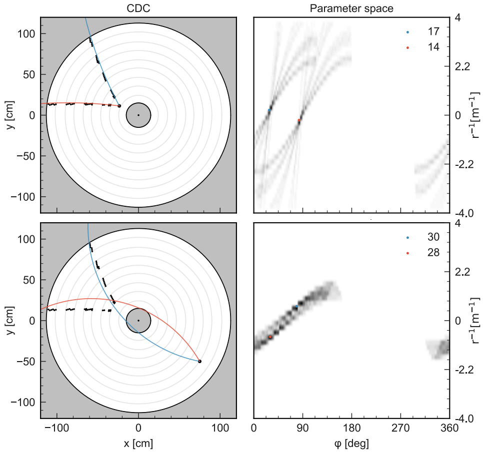
- Implementation of the 3DHough + DNN algorithm onto FPGAs (KIT)
- New displaced vertex trigger (DVT) for displaced tracks in development

Outlook: Displaced Vertex Trigger (DVT)

- Current L1 Trigger: Only capturing IP tracks
- New physics: Displaced vertices?



"Hot" physics from feebly interacting particles

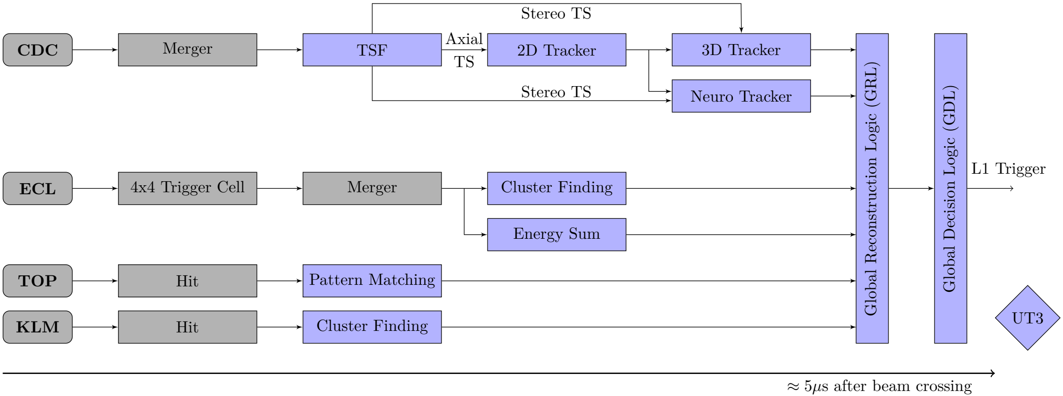


Hough track finding

Backup

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**

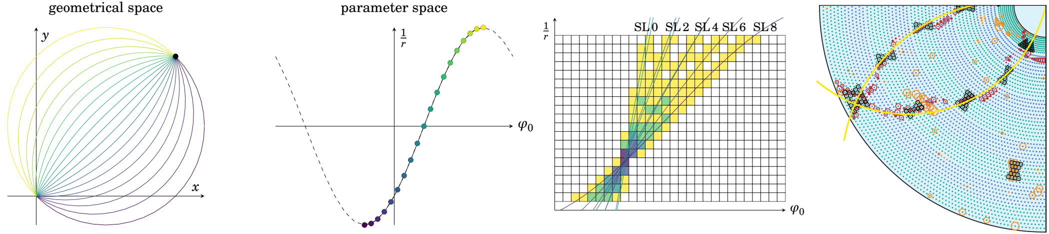


Preprocessing of the Network Input: Track Finding

Which TS belong to a real track?

TS selection using a two-dimensional Hough transformation:

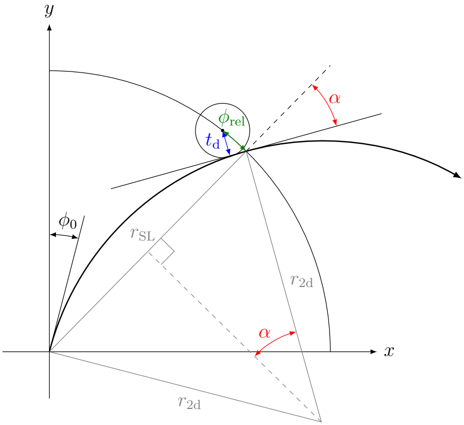
- Axial hit in CDC (TS) gets transformed to a curve in parameter (Hough) space
- Intersection point yields the track parameters ϕ and $r_{\mathrm{2d}} \propto p_{\mathrm{T}}$



$\Longrightarrow$ 2D track candidate

The Neuro Trigger has been running since January 2021 years with <u>remarkable success</u>.

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



With the TS information

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

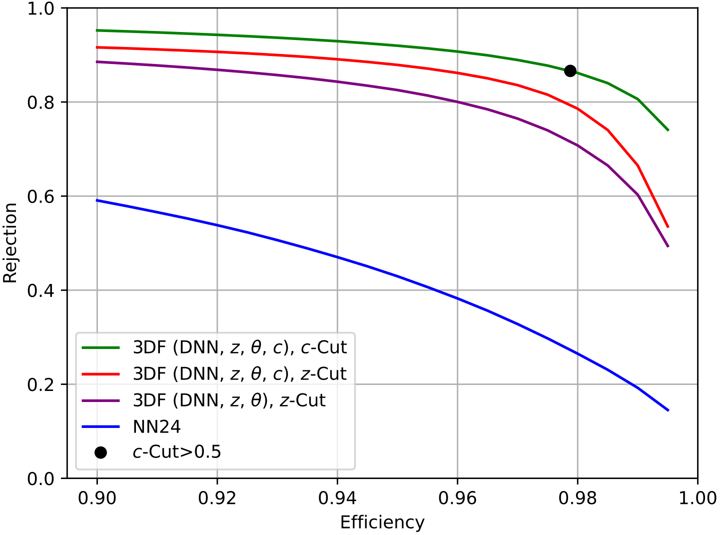
we can calculate:

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

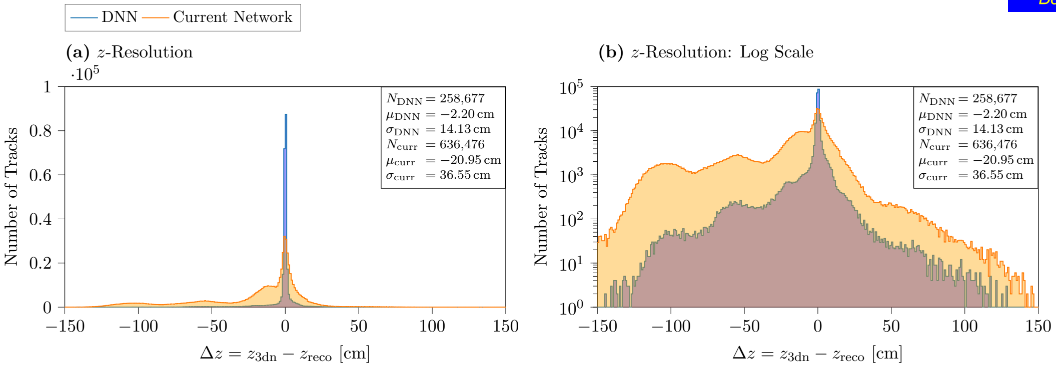
$$\phi_{\mathrm{rel}} = \phi_{\mathrm{wire}} - n_{\mathrm{wire}} \cdot \left(\frac{\phi_0 - \alpha}{2\pi}\right)$$

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

Performance Gain Through Classification



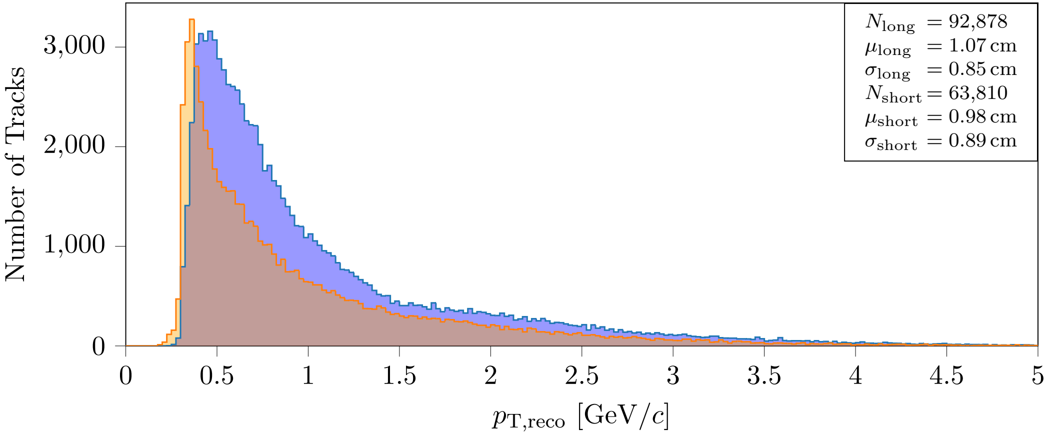
The z-Resolution after Classification



- DNN: 96.7% efficiency (cut at 0.5 classification → 19.2% incorrect)
- Current Network: 89.0% efficiency (cut at ± 15 cm → 69.8% incorrect)

The Reconstructed Single Track p_{T}-Distribution

(c) The p_{T}-Distribution of the Reconstruction



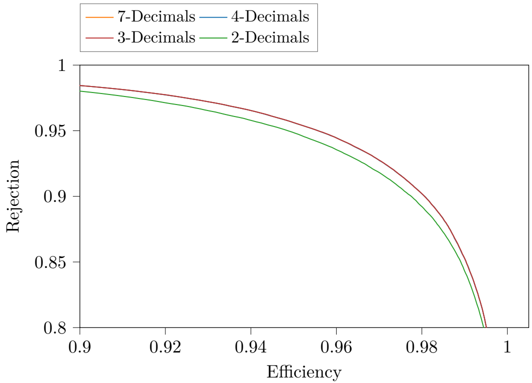
- Low momentum tracks overrepresented in the short tracks

Optimal Training Parameters

Parameter	**Value**
Epochs	1000
Patience	45
Batch Size	2048
Optimizer	Adam
Loss	MSE (target normalized to $[-1, 1]$)
Learning rate	10^{-3}
LR Scheduling Factor	0.5
LR Scheduling Patience	20
Weight decay	10^{-6}
Activation Function	tanh()
Number of Training Tracks	$\approx 10 \cdot 10^6$
Number of Validation Tracks	$\approx 1.5 \cdot 10^6$
Input Nodes	126
Hidden Layers	4 (60 nodes each)
Output Nodes	4 (z, θ, $1/p$, class.)

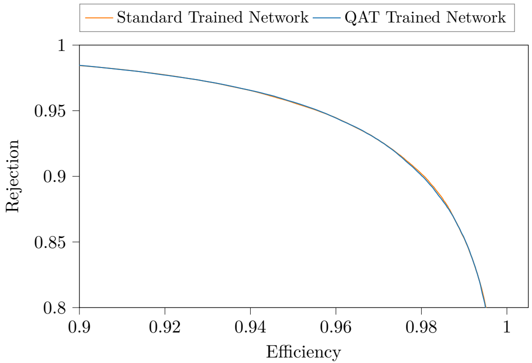
Rounded Weights and QAT

Comparison To Rounded Weights



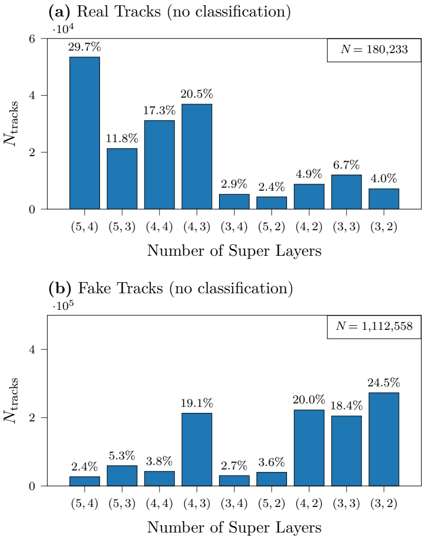
Weights and biases rounded to 3 decimals still as good as 32-bit float representation:

Quantization Aware Training



For floating point precision no improvement: Only necessary when forced to use a precision below 9 bits!

Differentiated Classification Cuts

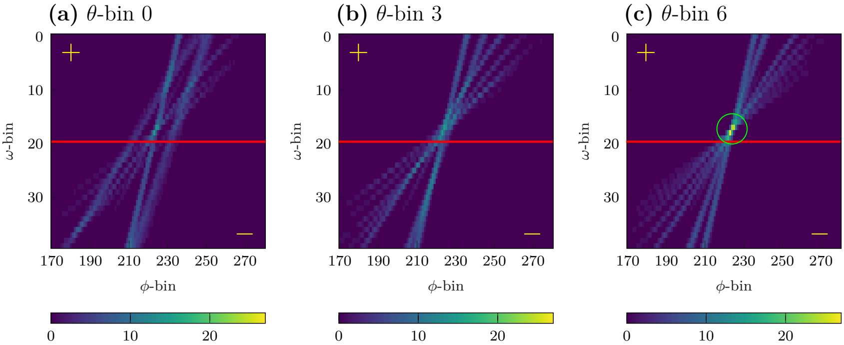


Example cut array:

- (5, 4): `0.33`
- (5, 3): `0.38`
- (4, 4): `0.33`
- (4, 3): `0.48`
- (3, 4): `0.62`
- (5, 2): `0.47`
- (4, 2): `0.66`
- (3, 3): `0.71`
- (3, 2): `0.80`

- Instead of a single across the board cut
- Optimization possible by weighting efficiency or rejection

Simulated Single Track:



Positive (negative) tracks in upper (lower) ω half plane

$\implies$ Intersection point yields ω, ϕ and $\cot(\theta)$

What is a Hit Representation?

Given any (priority) wire

$$\vec{s}(\lambda) = \begin{pmatrix} w_x \\ w_y \\ w_z \end{pmatrix} + \lambda \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix}, \qquad \lambda \in [0, 1]$$

a hit representations describes **all** possible tracks (helices) in the Hough space that **could activate** the wire.

As the Hough space is binned, this corresponds to a **non-zero integer weight** in the bin that describes one of these helices.

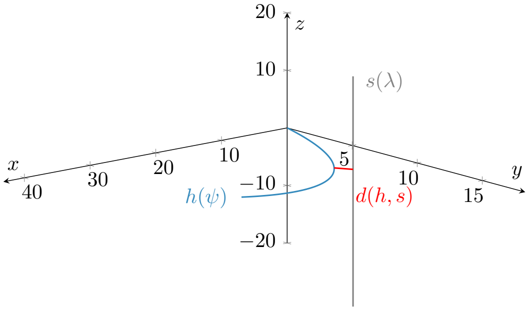
A helix can be described as:

$$\vec{h}(\psi) = \omega^{-1} \begin{pmatrix} \sin(\psi - \phi) + \sin(\phi) \\ \cos(\psi - \phi) - \cos(\phi) \\ \psi \cot(\theta) \end{pmatrix}, \qquad \psi \in [0, 2\pi]$$

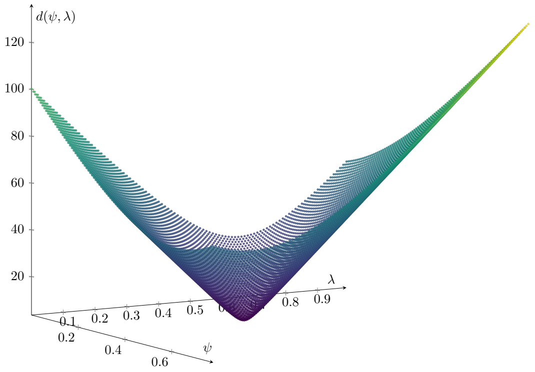
The Numeric Hit Representations

Numeric optimization of wire and helix distance:

$$d(\vec{h}(\psi), \vec{s}(\lambda)) = \sqrt{\sum_{i=1}^{3} (h_i(\psi) - s_i(\lambda))^2} \qquad \text{for} \qquad \lambda \in [0,1], \psi \in [0, 2\pi]$$



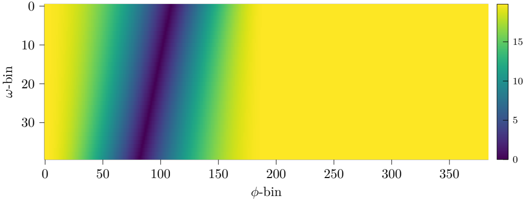
Euclidean distance between wire and helix



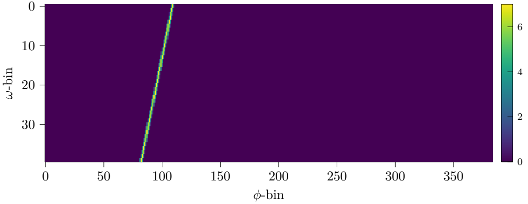
Distance minimization → Hyperplane

Creation Process

1. Take one (priority) wire
2. Consider **each** Hough space bin **individually**
3. Double Optimization (nested):
   - Optimize over all possible track parameters (helices) within this bin
   - Optimize for the minimum distance between the helix and the wire
4. Save the minimum distance and track parameter point of each bin
5. Weight each bin according to the distance to wire **and** distance to bin center

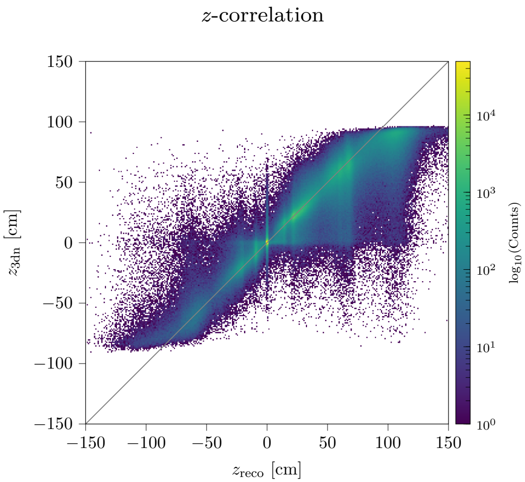


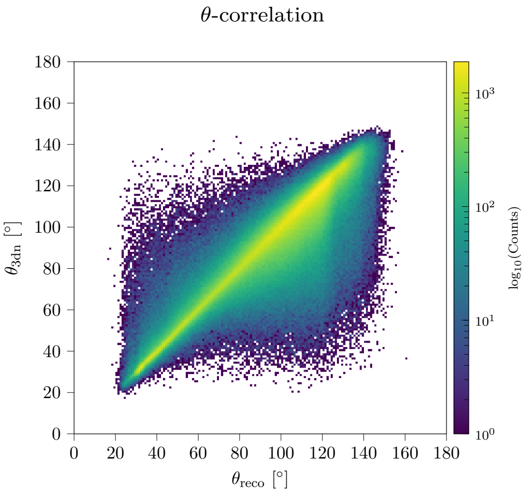
Distance Hough space [cm]



Weight Hough space [int]

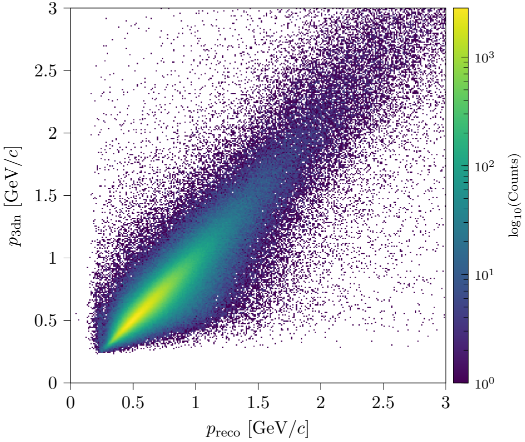
Parameter Correlations: Reconstruction vs. Neural Network



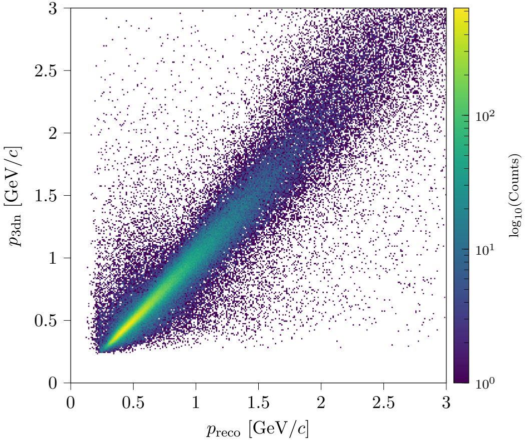


Parameter Correlations: Reconstruction vs. Neural Network

p-correlation without classification cut



p-correlation with classification cut



Some Perspective: Fake Track Segments in One Event

Real single track event from high background run: