Belle II Tracking and Optimisation of Hit Finding

Adeel Akram

Uppsala University
adeel.akram@physics.uu.se

15th Nordic Meeting on Nuclear Physics
Visby, Gotland, Sweden
(5 – 8 May 2026)

08-05-2026

Outline

- Belle II Experiment at KEK
- Particle Tracking
- Tracking and Optimisation of Hit Finding
- Conclusions

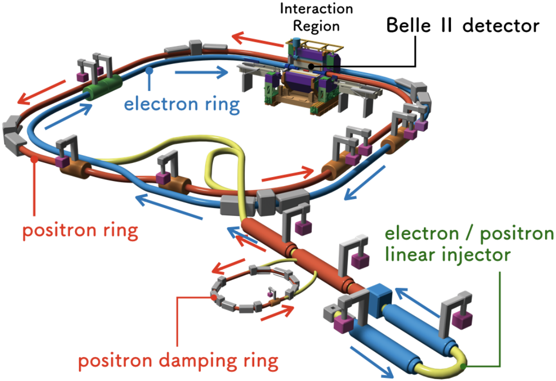
Belle II Experiment at KEK

SuperKEKB Collider:

- Asymmetric collider: 7 GeV e^-, 4 GeV e^+
- $E_{cm} = 10.58$ GeV ($\sim$ mass of $\Upsilon(4S)$) resonance), thus making it a $B-$factory
- Aiming for integrated $\mathscr{L} = 50\, ab^{-1}$, and instantaneous $\mathscr{L}_i = 6 \cdot 10^{35}$ cm^{-2}s^{-1}

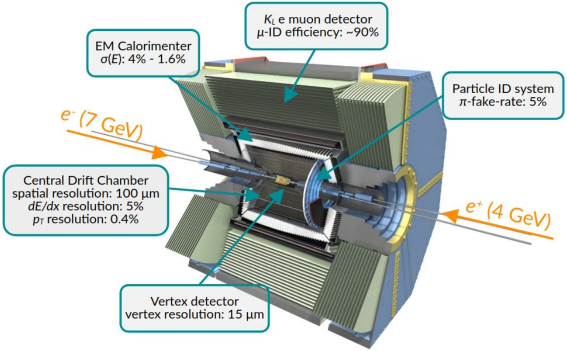
Belle II Experiment:

- Searches beyond SM at the intensity frontier
- Requires precise determination of $B-$meson decay vertices and low-momentum tracks



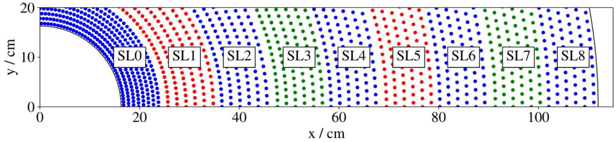
Belle II Detector

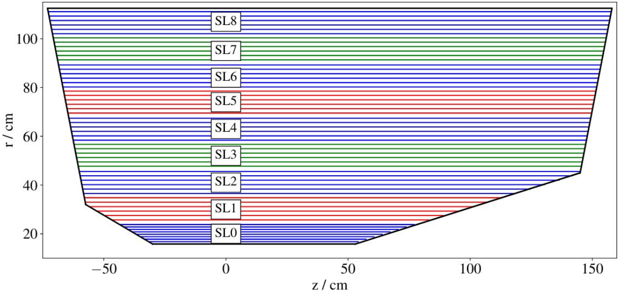
- Tracking detectors
  - Central Drift Chamber (CDC)
  - Vertex Detector (VXD)
- PID detectors
  - ECL (plus γ detector), ARICH, TOP, KLM (K_L–Muon detector)
- Solenoid Magnet (1.5 T), Hardware & software trigger, and DAQ



Central Drift Chamber (CDC)

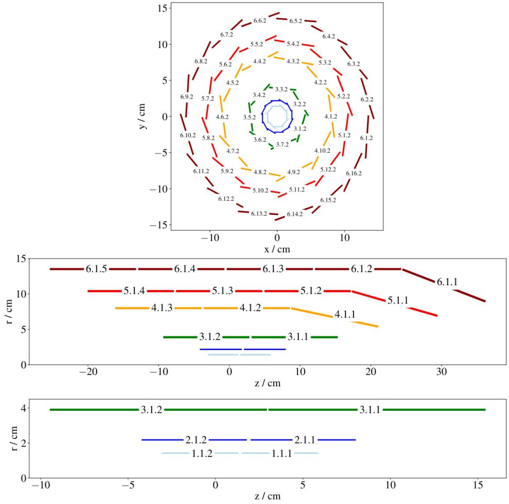
- Large multi-wire drift chamber
- 56 layers grouped into 9 superlayers
- Superlayers divided into 5 axial + 4 stereo (2U + 2V) *w.r.t* $z-$axis
  - Stereo layers allow 3D tracking: from (x, y) to (x, y, z)
- Gives resolution of $\sim 100\ \mu m$ (spatial), $\sim 0.4\%$ (p_T), and $\sim 5\%$ (dE/dx)





Vertex Detector (VXD)

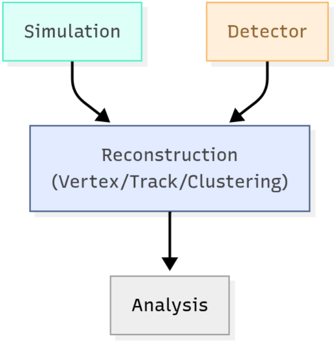
- VXD consists of PiXel Detector (PXD) and Silicon Vertex Detector (SVD)
- PXD has 2 layers of pixel sensors, and SVD has 4 layers of double-sided silicon strip sensors
- Rectangular (barrel and backward), and trapezoidal (forward) sensors



Particle Tracking

Finding particle trajectories inside detectors

- Track reconstruction is a crucial stage in the data analysis chain
- Reconstructs meaningful objects (track-level) from the raw signal (hit-level)
- Common stage for data coming from a simulation or a real detector

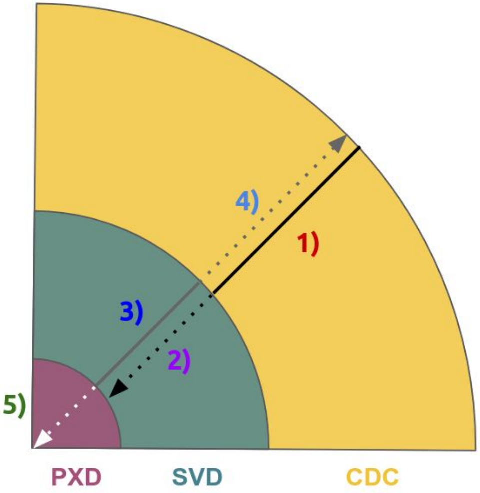


Belle II Tracking

In Belle II Software Analysis Framework (*basf2*), the tracking consists of several steps:

1. CDC tracking
2. Extrapolate to SVD with CKF (`ToSVDCKF`)
3. SVD tracking
4. Extrapolate to CDC with CKF (`ToCDCCKF`)
5. Combine and attach PXD hits

± CKF stands for Combinatorial Kalman Filter

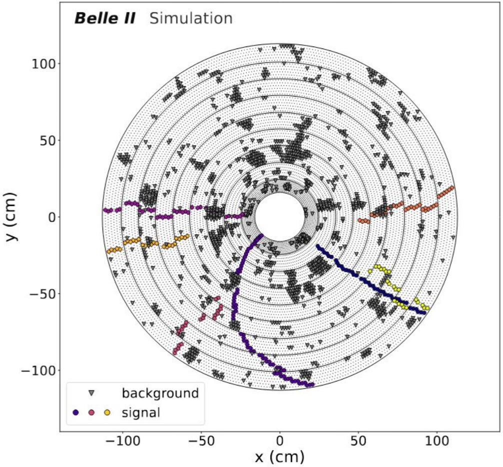


Problem: High Beam Background

Higher the $\mathscr{L} \rightarrow$ higher beam background

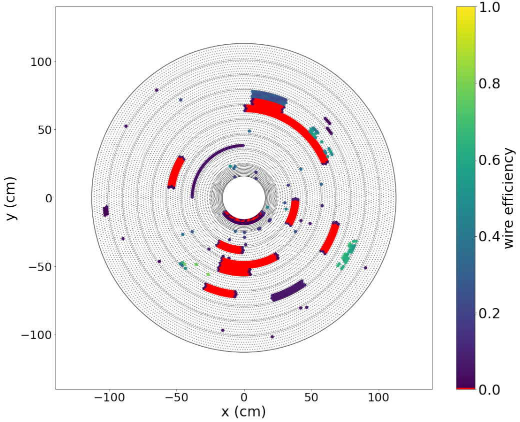
- Signal is dominated by background
- Track reconstruction becomes challenging and computationally expensive

Even existing performant algorithms need tuning



Problem: Detector Degradation

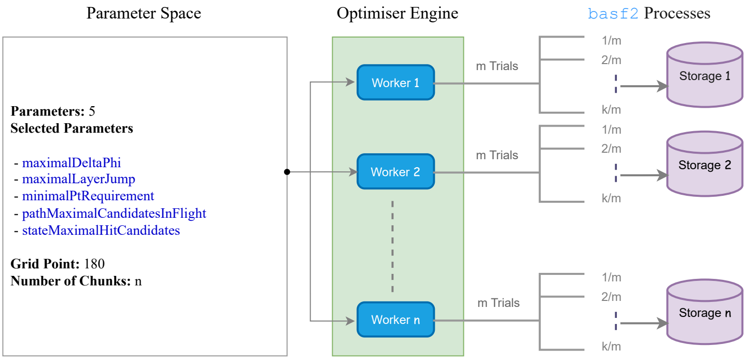
- Detector inefficiencies caused by dead readout boards → detector holes
- Extrapolation from SVD to CDC is affected
  - `ToCDCCKF` extrapolation stops/deteriorates at dead regions
- Tracking in SVD (`Tracks`), and extrapolation to CDC (`Hits`)
  - `ToCDCCKF` has ∼ 12 tunable parameters
- Optimal parameters maximising **hit efficiency** (ε) & **hit purity** (ρ) → trade-off



Simulated CDC geometry

ToCDCCKF Optimiser

Single objective optimisation, maximise F1-score $(\equiv 2 \cdot \frac{\varepsilon_{\text{hit}} \times \rho_{\text{hit}}}{\varepsilon_{\text{hit}} + \rho_{\text{hit}}})$

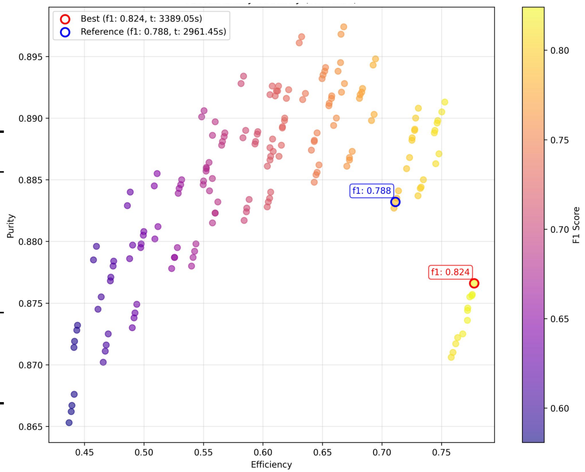


Designed for HPC deployment, written in native Python.

ToCDCCKF Optimiser

Scanned 180 different combinations, Grid Points:

Parameter	Reference	Optimal
maximalDeltaPhi	0.4	0.2
maximalLayerJump	4	4
minimalPtRequirement	0.0	0.1
pathMaximalCandidatesInFlight	3	4
stateMaximalHitCandidates	4	5
F1 Score (%)	78.80	82.41
Hit Efficiency (%)	71.13	77.75
Hit Purity (%)	88.32	87.66



Full Tracking

Running full track reconstruction, with optimal `ToCDCCKF` parameters, we get **track-level** metrics:

Metric	Reference		Optimal		Difference
	value (%)	error (%)	value (%)	error (%)	value (%)
Track Finding Efficiency	91.68	0.31	91.82	0.31	0.14
Fake Rate ($\equiv 1 -$ Purity)	3.06	0.16	3.09	0.16	0.03

Small discrepancy points toward adding other parameters in the search space.

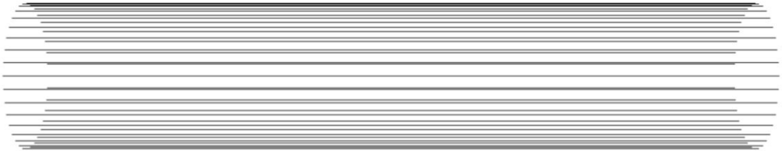
Conclusion

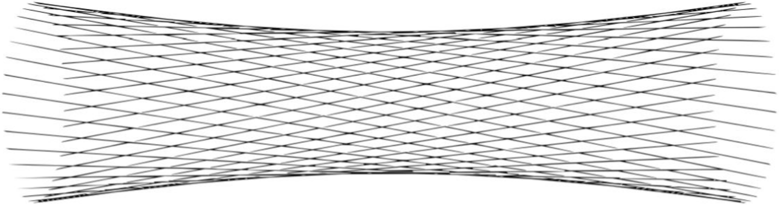
- The optimizer works as intended
- It's modular, scalable, and parallelizable (multiprocessing or HPC clusters)
- Ready to be deployed for extensive parameter search

END

Backup

CDC Layers





Hit-level Metrics

Hit efficiency (ε_{hit}) is the fraction of true hits from a Monte Carlo (MC) particle correctly used in its reconstructed track.

$$\varepsilon_{\text{hit}} = \frac{N_{\text{matched hits}}}{N_{\text{true hits}}}$$

Hit purity (ρ_{hit}) is the fraction of hits on a reconstructed track that truly belong to the same MC particle.

$$\rho_{\text{hit}} = \frac{N_{\text{matched hits}}}{N_{\text{reconstructed hits}}}$$

Tracking Metrics

The track finding efficiency (ϵ_{track}) is the fraction of particles ($N_{\text{particles}}$) that match at least one reconstructed track:

$$\epsilon_{\text{track}} = \frac{N_{\text{particles}}(\text{selected, matched})}{N_{\text{particles}}(\text{selected})}$$

The track purity (ρ_{track}) is defined as the fraction of reconstructed tracks (N_{tracks}) that match a selected particle:

$$\rho_{track} = \frac{N_{\text{tracks}}(\text{selected, matched})}{N_{\text{tracks}}(\text{selected})}$$