Draft for ctd-presentation

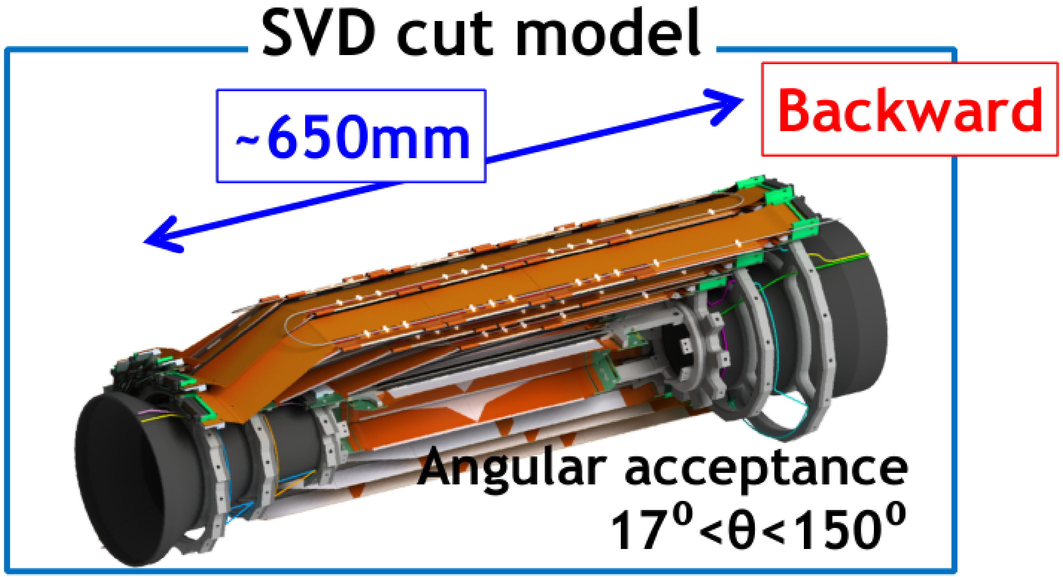
Jakob Lettenbichler
for the
Tracking Group of Belle II

No eyecandy yet – will be added later

Overview

- Belle II VXD Detector
- The VXDTF
  - SectorMap
  - Current performance
- VXDTF-refactoring
  - General approach
  - SpacePoints
  - SectorMap II
  - SegmentNetworkProducer
- Next steps

Belle II VXD

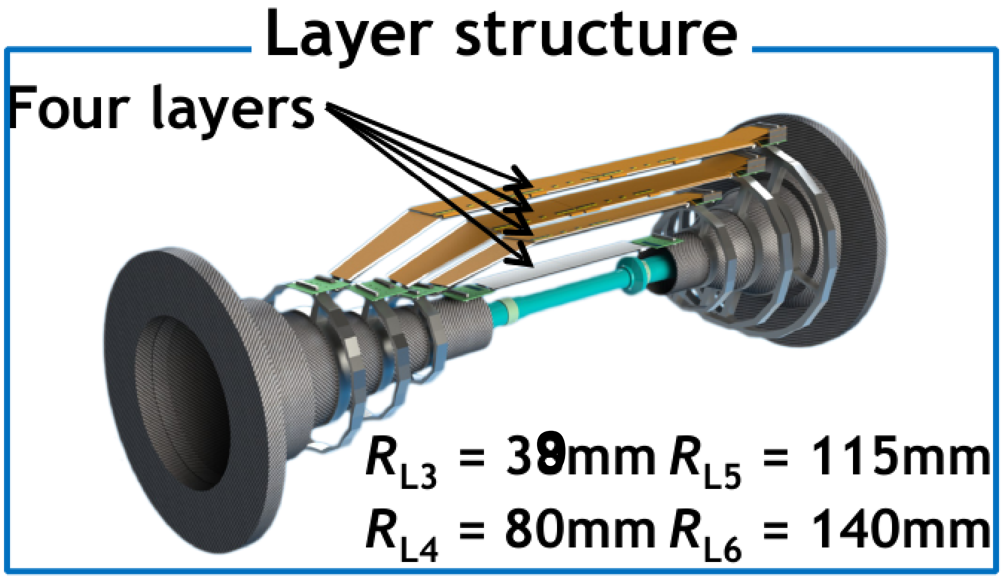


SVD cut model

~650mm

Backward

Angular acceptance $17^0<\theta<150^0$

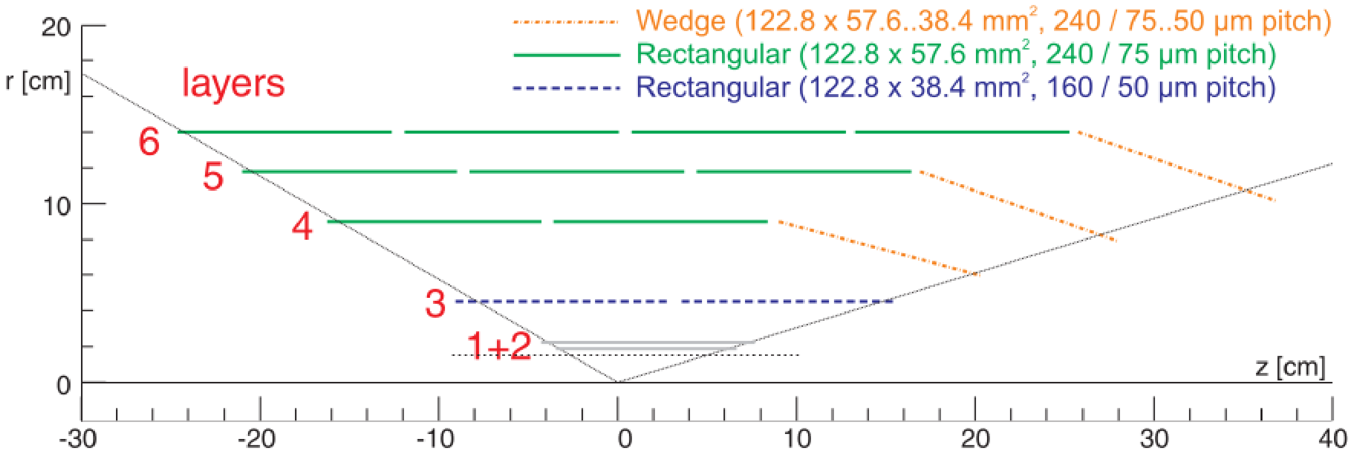


Layer structure

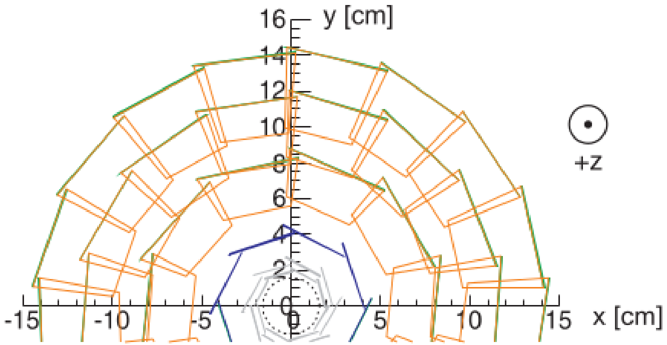
Four layers

R_{L3} = 38mm R_{L5} = 115mm

R_{L4} = 80mm R_{L6} = 140mm



Wedge (122.8 x 57.6..38.4 mm², 240 / 75..50 µm pitch)
Rectangular (122.8 x 57.6 mm², 240 / 75 µm pitch)
Rectangular (122.8 x 38.4 mm², 160 / 50 µm pitch)
layers
6
5
4
3
1+2
r [cm]
z [cm]
20
10
0
-30
-20
-10
0
10
20
30
40



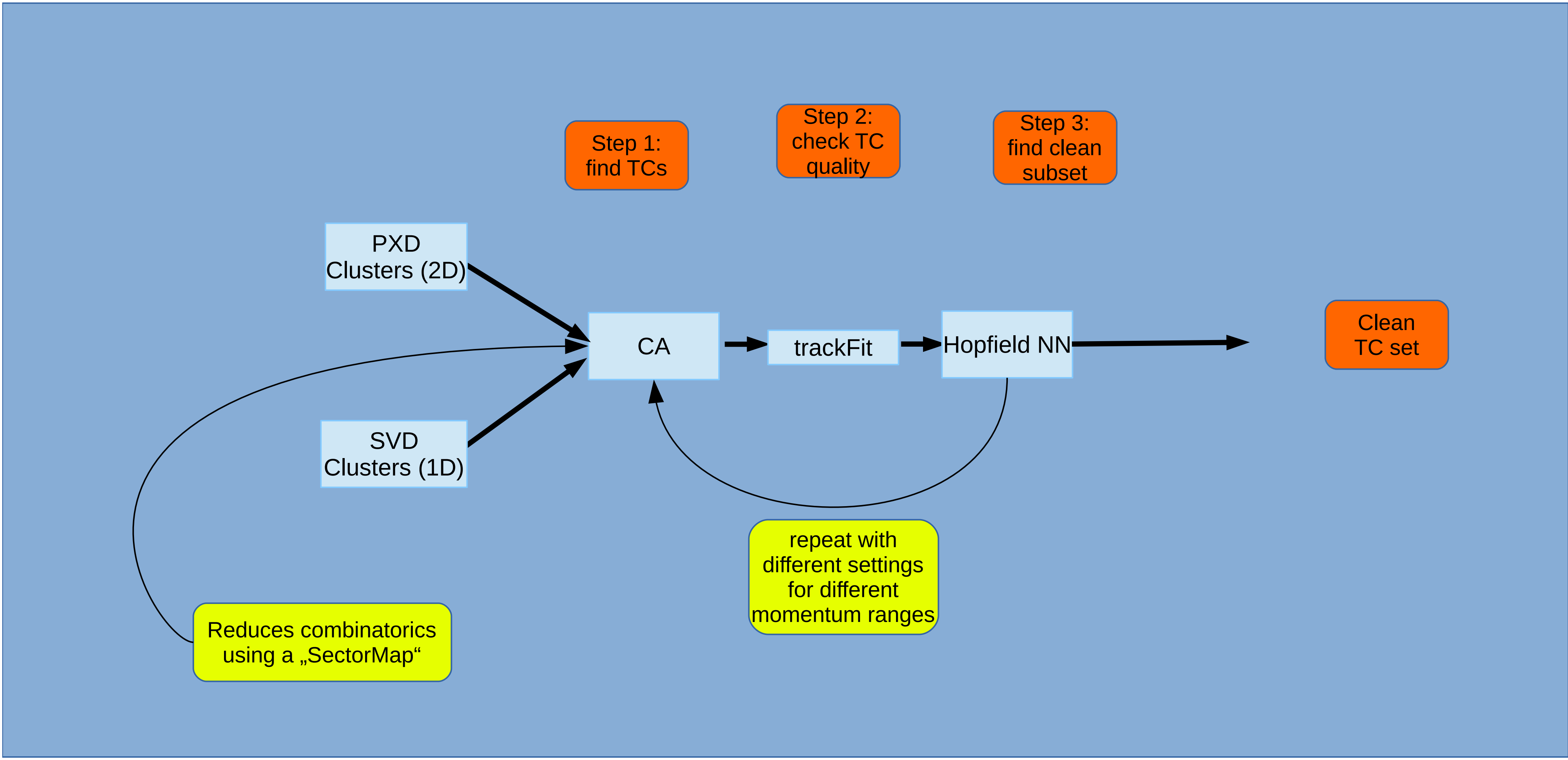
Vertex Detector (VXD) consists of:

- 2 layers of DEPFET Pixels (PXD), @ radii: 1.4, 2.2 cm
- 4 layers of double sided silicon strip (DSSD) sensors (SVD), @ radii: 3.9, 8, 11.5, 14 cm

VXD Tracking has to deal with..

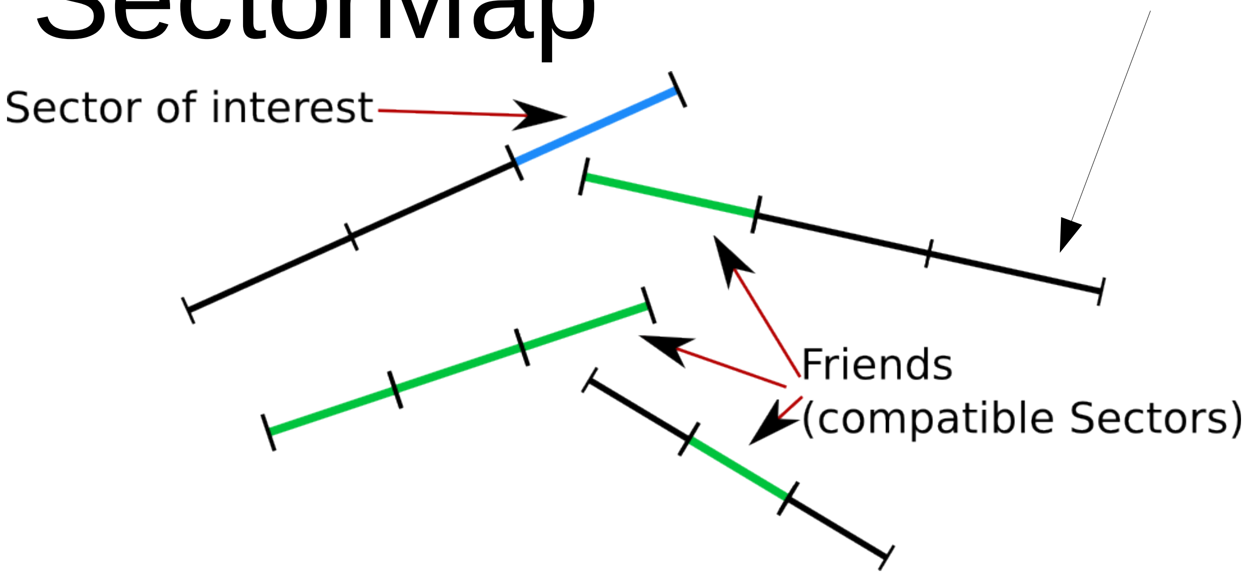
- 1.5 T magnetic field
- Windmill design with overlaps & slanted sensor
- HLT: 4 layer SVD tracking (on-line)
- Fast reco: 6 layer (SVD+PXD) tracking with predefined ROIs for the PXD (off-line)
- Goal: reco down to p_T = 50 MeV/c
- ghost hits (SVD)
- high energy deposit for low momenta (p_T< 100 MeV/c)
- Loopers/Curlers for tracks with p_T< 500 MeV/c

The basic VXD TrackFinder (VXDTF) approach



- TC: Track Candidate
- CA: Cellular Automaton
- TrackFit: e.g. CircleFit or Kalman Filter (KF)
- Hopfield NN: a neural network of Hopfield type

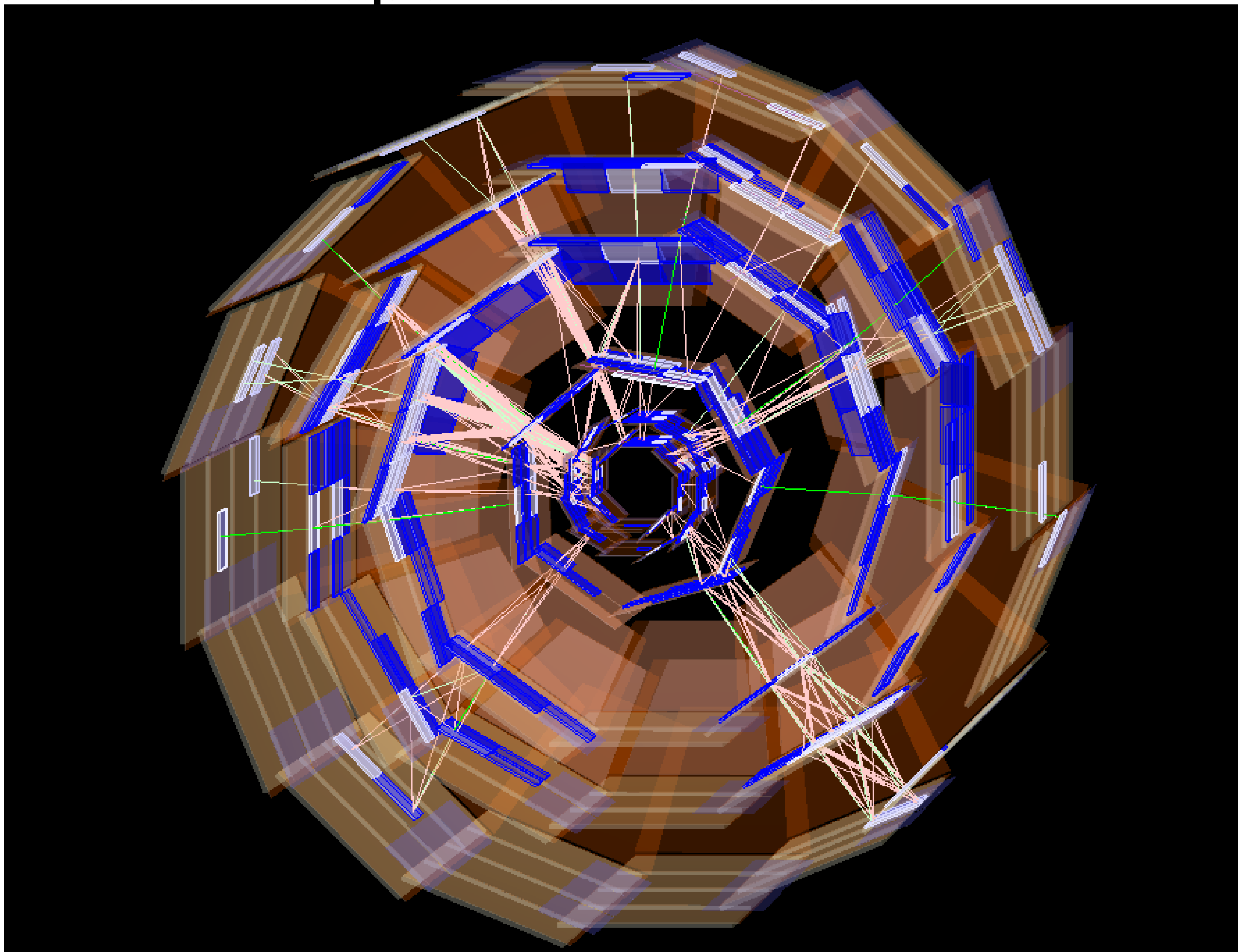
The SectorMap



- Sensors are sub-divided into *Sectors*
- Each sector knows its *friend* sectors
- Sectors are *friends* if a track from the vertex can pass through both of them
- SpacePoints are sorted into sectors
- Only SpacePoints in *friend* sectors can be combined

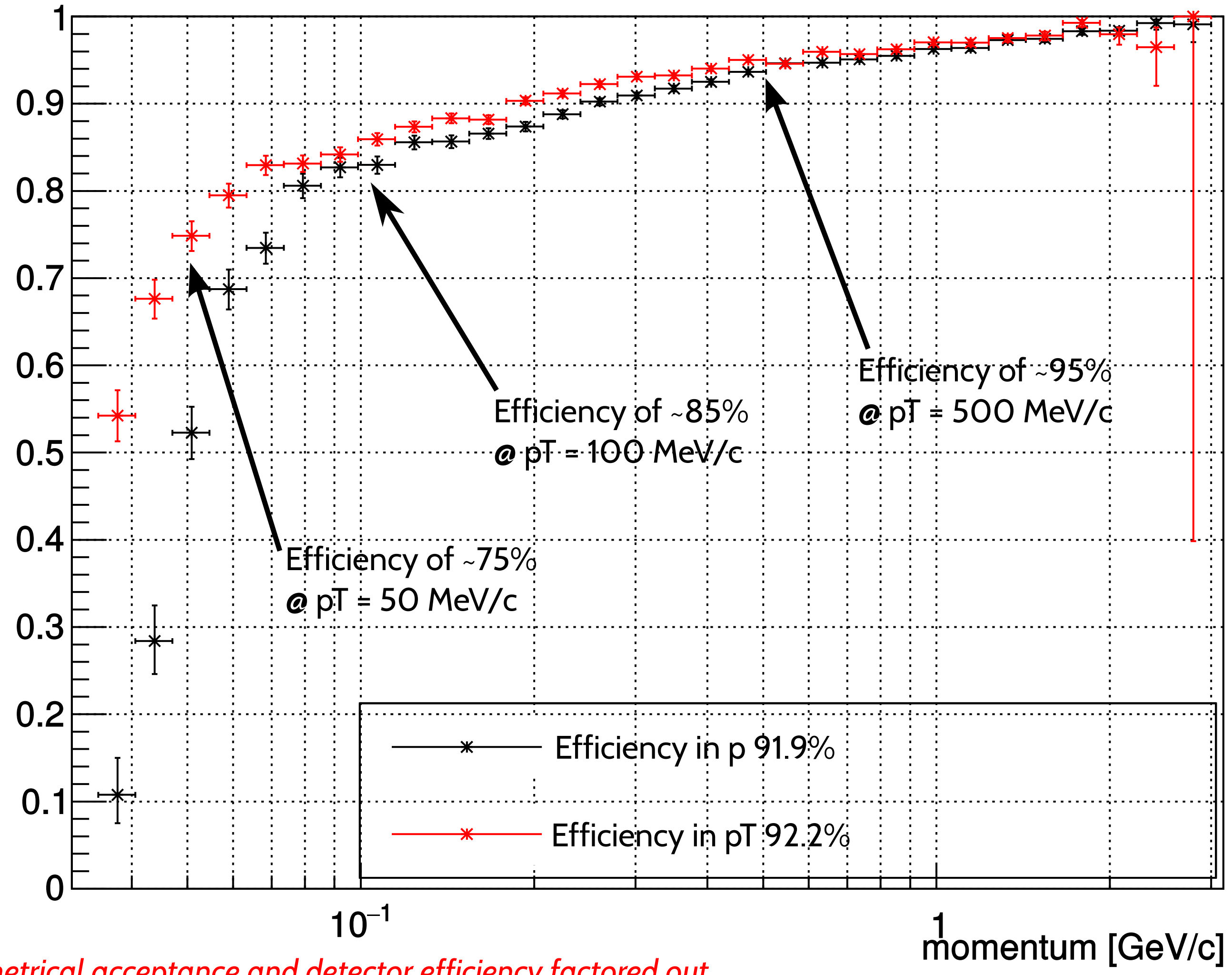
The SectorMap - II

- *Friend* sectors are combined via training on MC-data
- Sector-combinations store filter-cuts used for reducing combinatorics of SpacePoints
- 2-,3- & 4-SpacePoint-cuts are independent for each sector-combination
- Different sectorMaps (with their independent cutoff-sets) for different momentum ranges
- About 10 sectors per sensor are used



VXDTF - current performance

Efficiency vs momentum



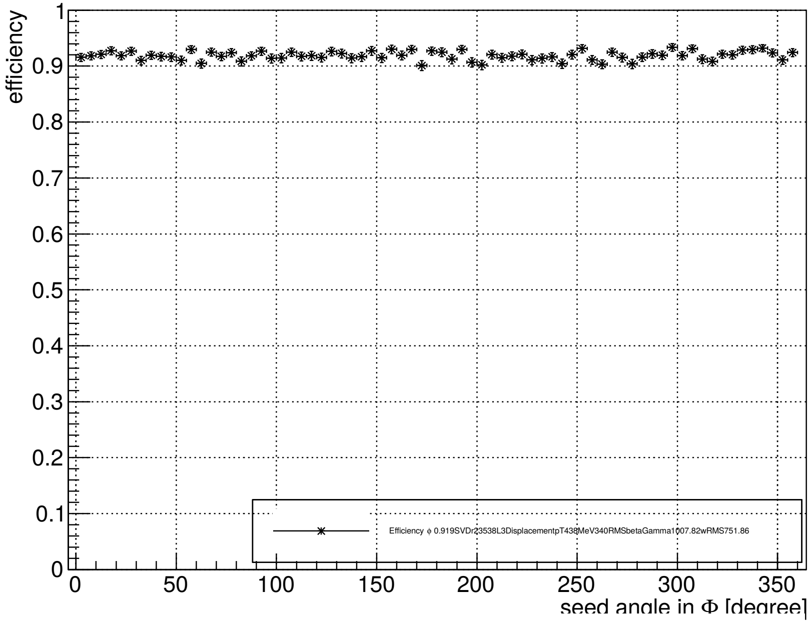
Efficiency of ~75%
@ pT = 50 MeV/c

Efficiency of ~85%
@ pT = 100 MeV/c

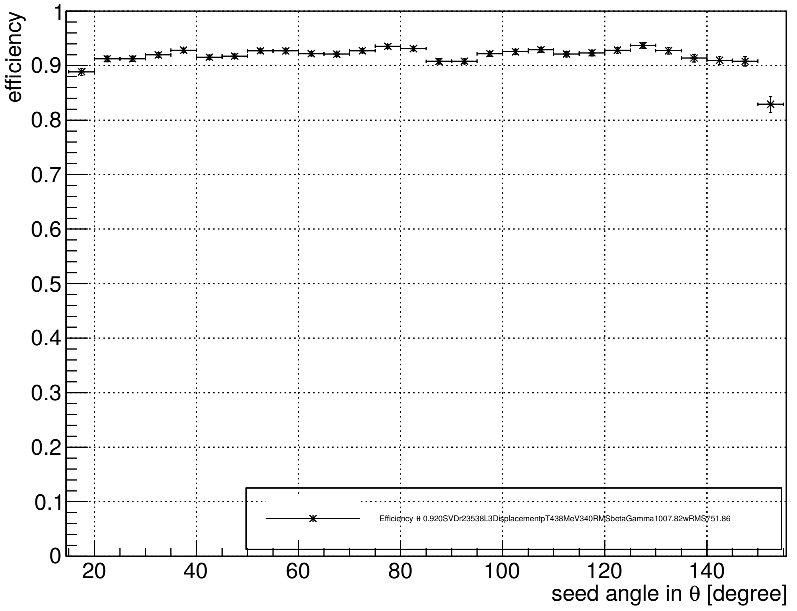
Efficiency of ~95%
@ pT = 500 MeV/c

etrical acceptance and detector efficiency factored out

Efficiency of Φ



Efficiency of Φ



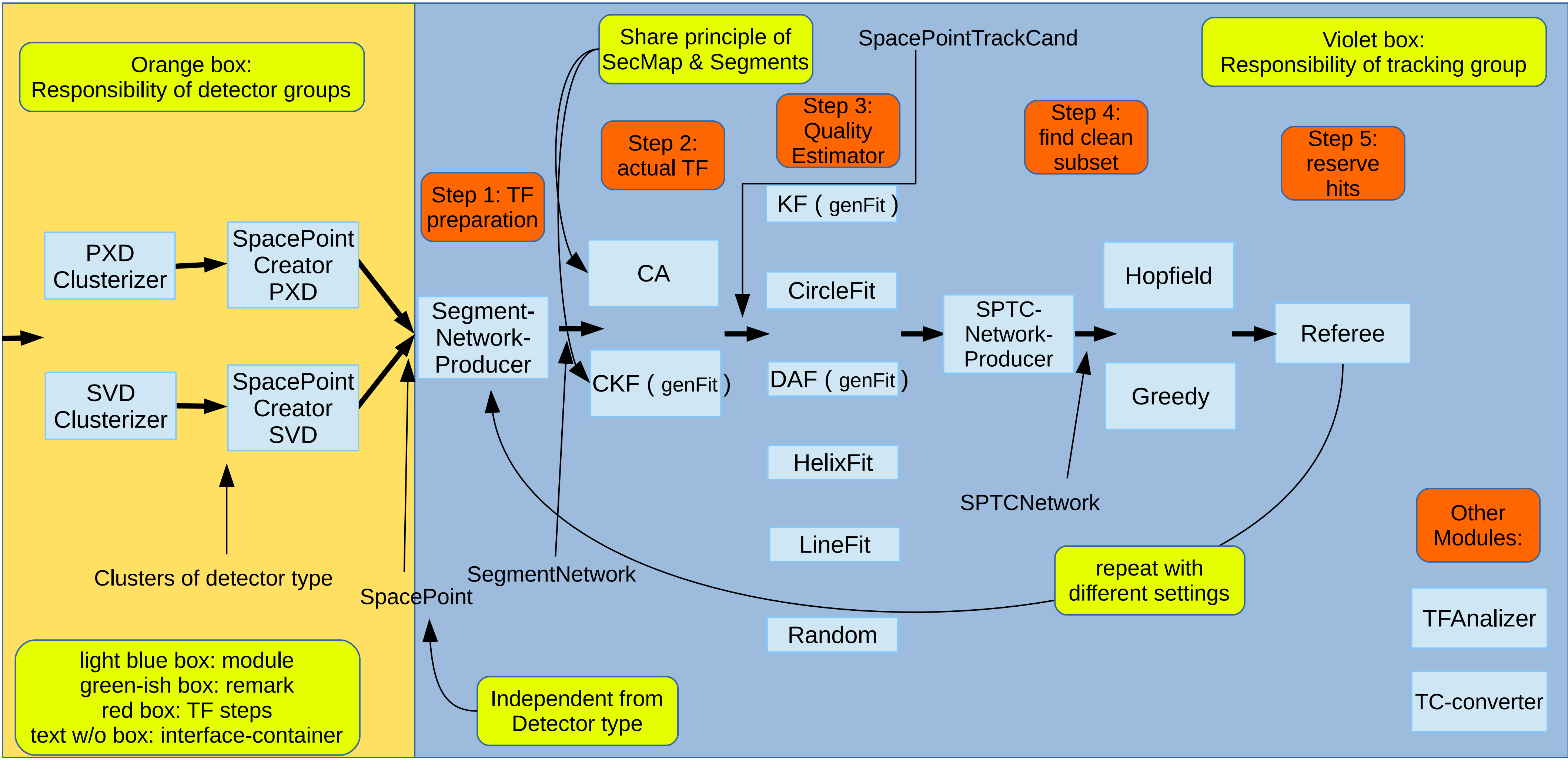
Y(4s) results with typically 10 tracks per event

TODO: Belle I & BaBar Data

VXDTF – refactoring goals

- About 2 years to go, so we are fine? → No we refactor the code, because:
- We want to Simplify debugging:
  - High coverage of unit- and integration tests
  - High flexibility using modular design (CA, CKF, DAF, other filters replaceable)
  - Allow shared workload on several developers
- We want to have extended training capabilities:
  - Allow bigger sample sizes for sectorMap-training
  - More tools for finding issues (loops in sectorMap, bad cuts, automatized cut-selection)

Planned modules for the VXDTF (event-part)



- CA: Cellular Automaton
- KF: Kalman Filter
- DAF: Deterministic Annealing Filter
- Hopfield: a neural network of Hopfield type
- SPTC: SpacePointTrackCandidate

SpacePoints

- Detector-independent – „just a point in 3D-Space“
  - Used for PXD and SVD
  - Hides detector specific treatment from TF
  - Quality indicator carrying extra info
- SVD (Double sided strip detector):
  - Combination of 1D-Clusters shall be done using:
    - Use of Energy deposit correlations
    - Hit time correlations (time resolution up to ~2 ns), curler detection
    - Further things to be investigated
- PXD
  - Energy deposit possible
  - (Bad) Cluster shape

SegmentNetworkProducer I

Some thoughts:

- SectorMap is actually a ***directed graph without loops*** (like the CA)
  - Each event a different subGraph of that SectorMap is needed → sectors having SpacePoints in that event
  - The sectors containing SpacePoints in an event are called *ActiveSectors* and form that subGraph
- But SpacePoints and track segments can form such graphs too!
- This means that there are a lot of graphs – or networks – to be formed within an event → lets unify this a bit: *DirectedNodeNetworks*

SegmentNetworkProducer II

Basic principle of DirectedNodeNetwork (DNN):

- Objects are related forming a network, where objects treated as Nodes and Links/Edges indicate their compatibility
  - Nodes can carry anything (sectors, hits, segments, integers, ..) as „node-entries“
  - Only following requirements to node-entries:
    - Minimal requirements needed for std::vector (e.g. Public constructor without arguments)
    - '==' operator must be defined
  - Cell-features or other „meta info“ can be attached via template parameter → CA could be applied to any network without modifying the Node-Entries
  - Links/Edges carry no extra info to minimize overhead.

SegmentNetworkProducer III

In Action:

- SpacePoints are matched to their sectors → ActiveSectors store event-dependent info
- ActiveSectorNetwork: built from Sectors which have got hits in that event
  - only compatible (Active-)sectors are linked
  - '0/1'-hit-filter: only physically relevant hits can form a sufficiently long chain of ActiveSectors
  - Serves as input for the *SpacePointNetwork*
- SpacePointNetwork: built from SpacePoints which are in ActiveSectors of ActiveSectorNetwork
  - Two hits get linked, when 2-hit-tests are passed (e.g. dist3D)
  - '2'-hit-filter
  - Serves as input for the *SegmentNetwork*

VXDTF2 – SegNetProducer III

- SegmentNetwork: build from SpacePoint-combinations in SpacePointNetwork
  - Two hits-pairs (→ segments) get linked, when 3-hit-tests are passed (e.g. angle3D, FastBDT (see Thomas Madleners talk)
    - *„A linked pair of nodes of one network becomes the node of the next one“*
  - '3'-hit-filter
  - Serves as input for the CA or the CKF

VXDTF2

- Some preliminary results (if I am successful in creating some)

VXDTF2 – next steps

- Combined beam test 2016
- Finishing VXDTF2 draft stage and implementing proof-of-concepts for CKF and DAF
- Tons of studies

This is it

- Many thanks to all members of the Tracking group, especially:
  - Rudolf Frühwirth
  - Eugenio Paoloni
  - Martin Heck
  - Martin Ritter and Christian Pulvermacher
  - Thomas Madlener
  - Tobias Schlüter
  - Giulia Casarosa

Your suggestions?

- What was missing
- What was too detailed
- What should be removed
- What should be changed