# Connecting The Dots 2016

22-24 February 2016 HEPHY Vienna Europe/Vienna timezone

# Tracking in the Belle II Vertex Detector

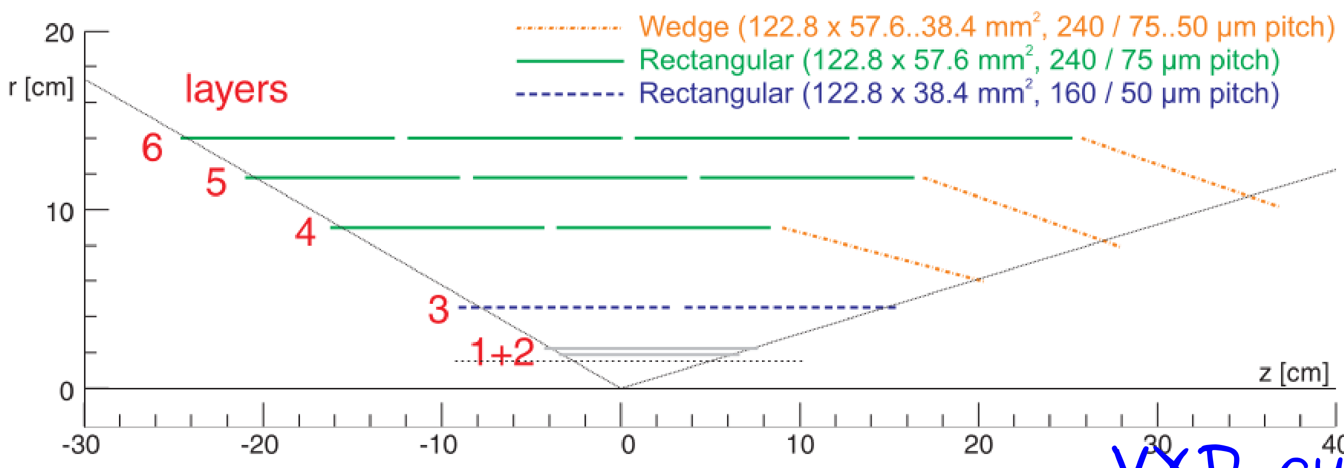

Jakob Lettenbichler
for the
Tracking Group of Belle II

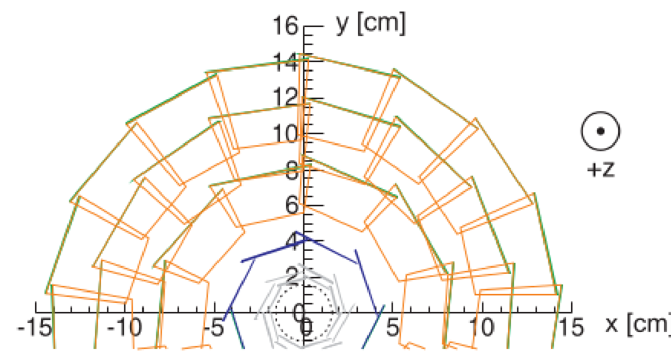
# Overview

- Belle II VXD Detector
- The VXD-TrackFinder (VXDTF)
  - SectorMap
  - Current performance
- VXDTF-refactoring
  - General approach
  - SpacePoints
  - SectorMap II
  - SegmentNetworkProducer
  - CA
  - Preliminary performance
- Next steps

# Belle II VXD

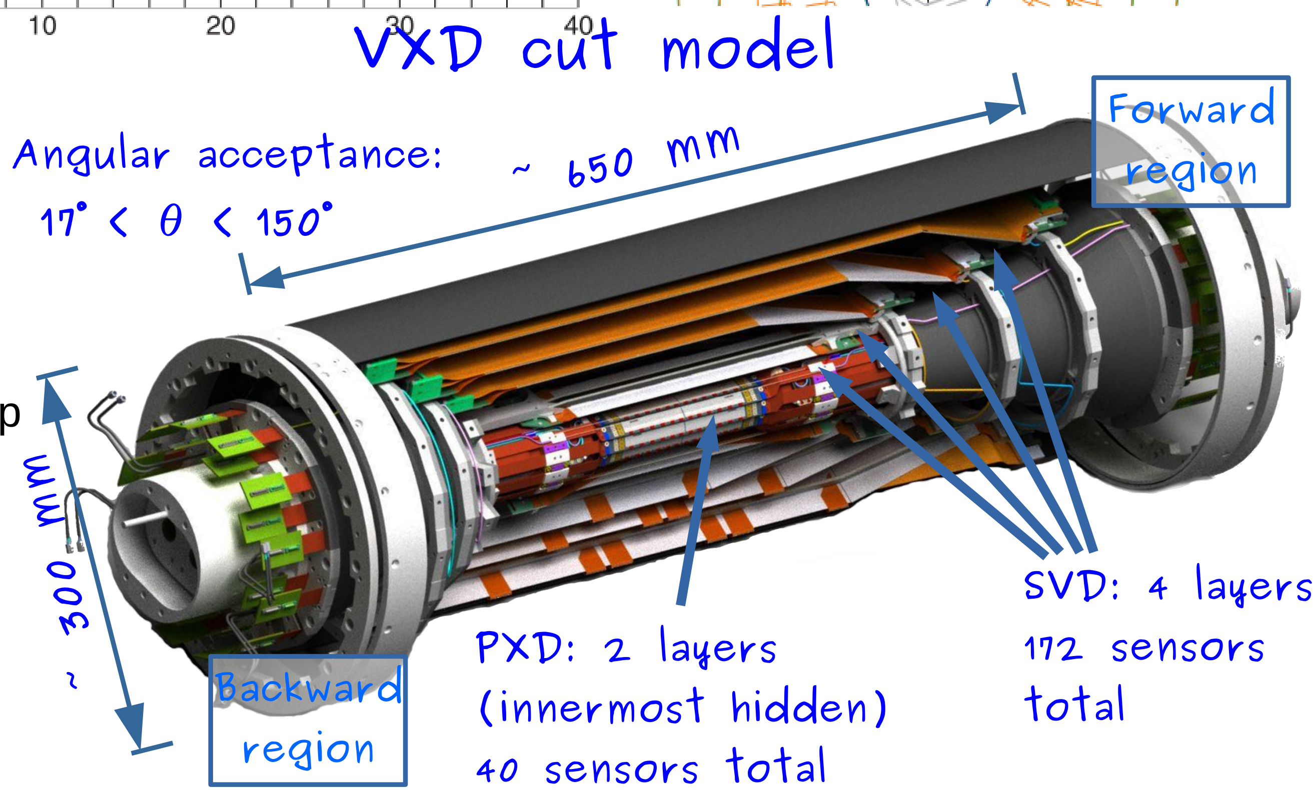





Vertex Detector (VXD) consists of:

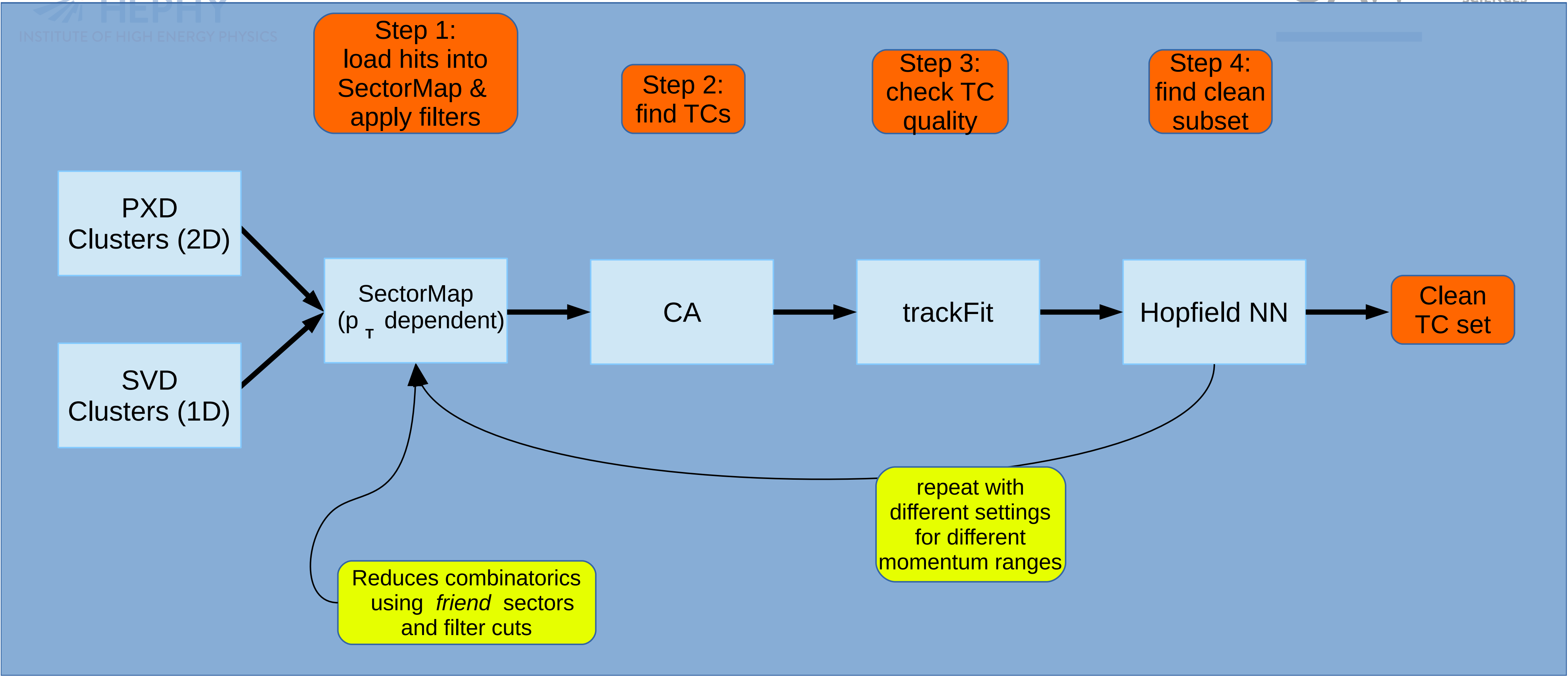
- 2 layers of DEPFET Pixels (PXD), @ radii: 1.4, 2.2 cm
- 4 layers of double sided silicon strip (DSSD) sensors (SVD), @ radii: 3.9, 8, 11.5, 13.5 cm, low material budget: **X/X0: ~ 0.55% / Layer**



# VXD Tracking has to deal with...

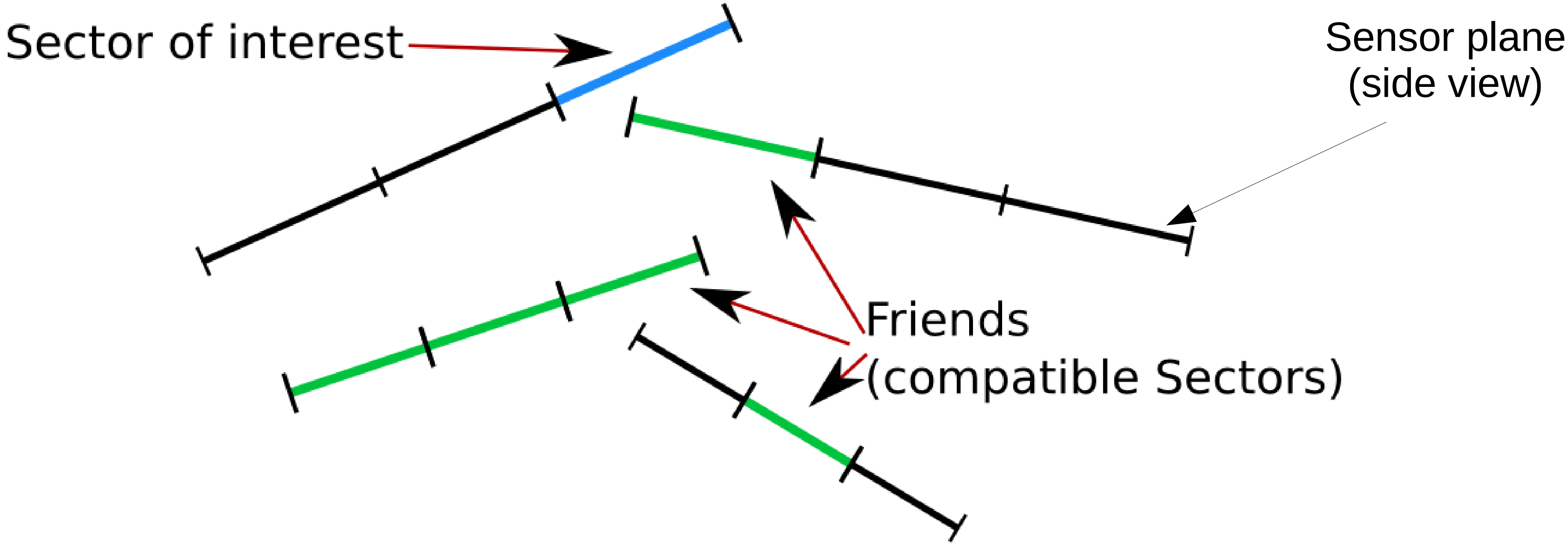
- 1.5 T magnetic field
- Windmill design with overlaps, slanted-, rectangular- and trapezoidal sensors
- 2 use-cases:
  - HLT: 4 layer SVD tracking
  - Fast reconstruction: 6 layer (SVD+PXD) tracking with predefined ROIs for the PXD
- Goal: reconstruction down to $p_T$ = 50 MeV/c
- Ghost hits (SVD)
- 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) structure




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

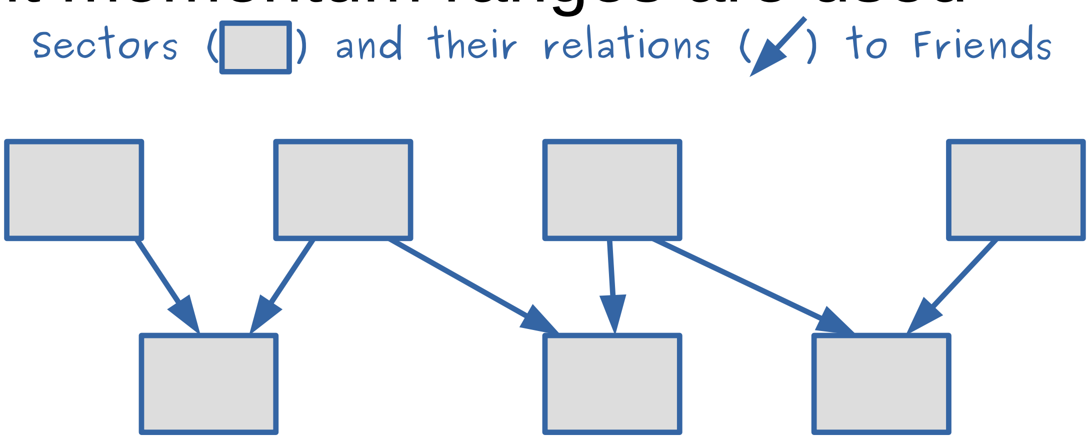
# The SectorMap



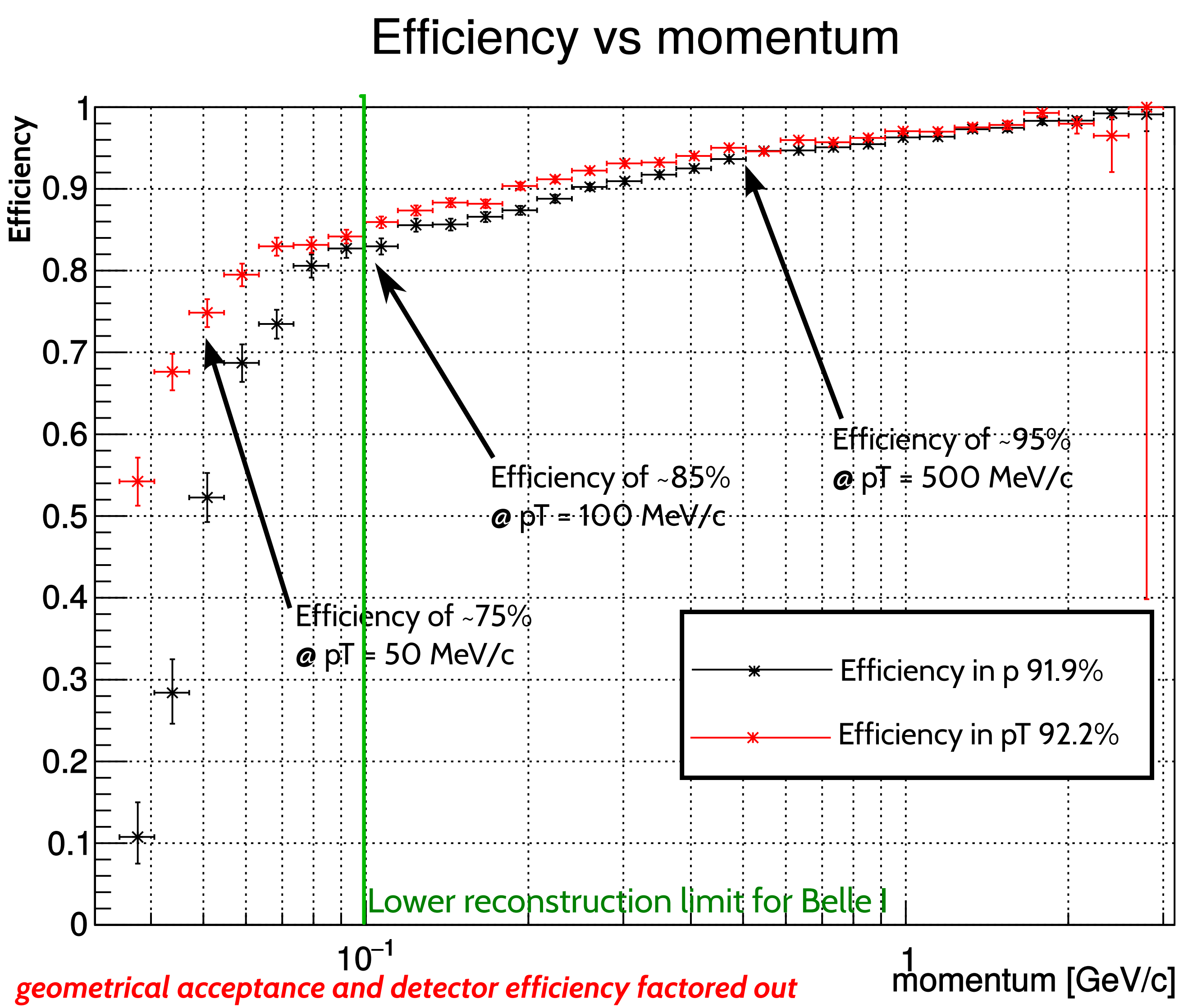

- Sensors are sub-divided into *Sectors (About 10 sectors per sensor are used)*
- Each Sector knows its F*riend* Sectors (directed graph)
- Sectors are Friends (a Sector→Friend-relation) if a track from the vertex can pass through both of them
- SpacePoints of an event are sorted into Sectors
- Only SpacePoints in Friend-*S*ectors can be combined

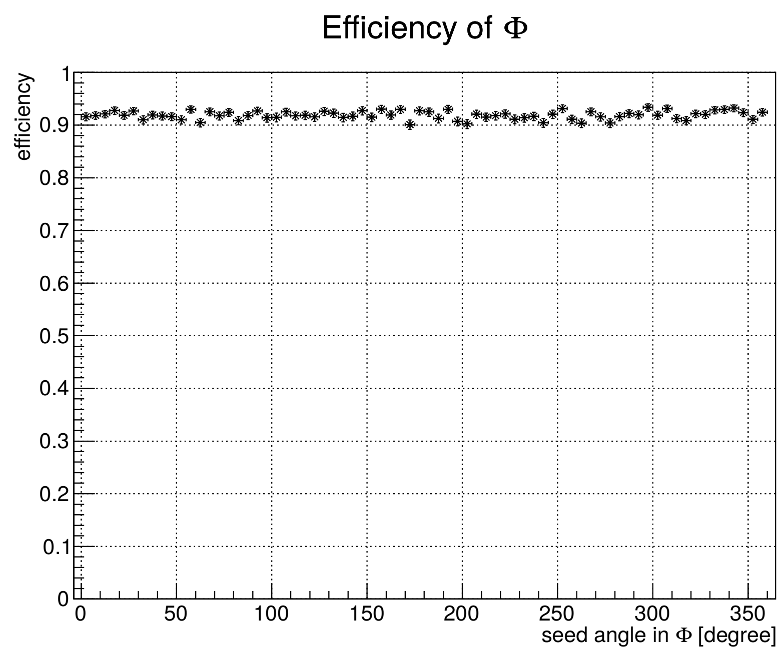
# The SectorMap - II

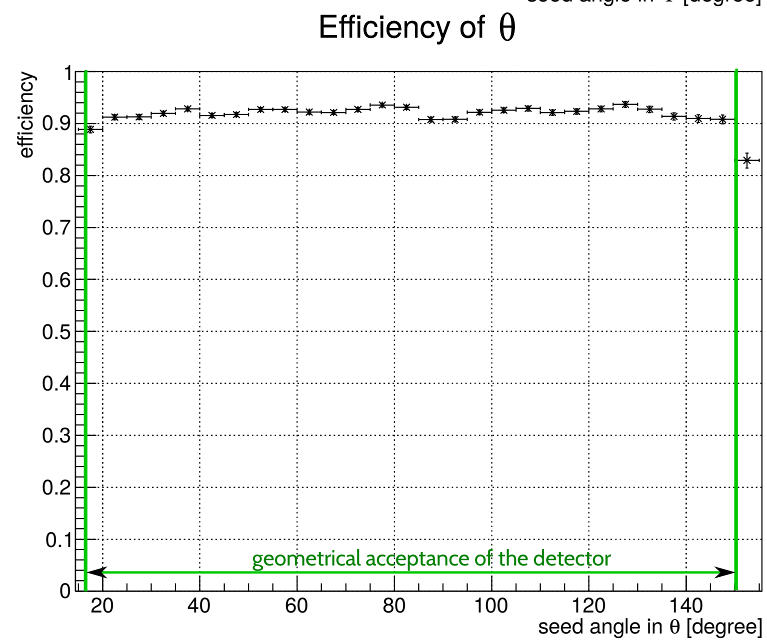
- *Friendsectors* are found via training on MC-data
- Each Sector→Friend-relation stores independent cuts for their filters
- Sector→Friend-relations can form chains of 2-4 sectors, which store the cuts for 2-4-SpacePoints-filters, respectively
- Cuts, filters to be used and Sector→Friend-relations are stored in a single sectorMap
- Different sectorMaps (with their independent cutoff-sets) for different momentum ranges are used



# VXDTF - current performance



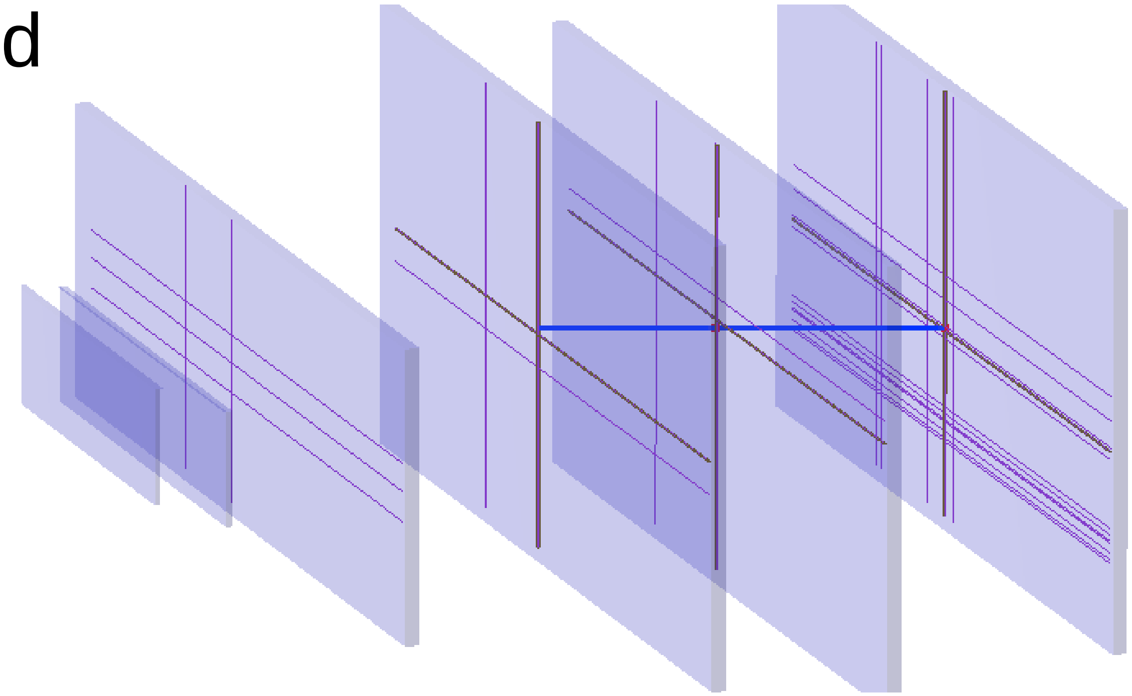




*geometrical acceptance and detector efficiency factored out*

Fake rate: 4.84% - hit efficiency: 91.7% - clone rate: 0.16%

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

# VXDTF – current performance



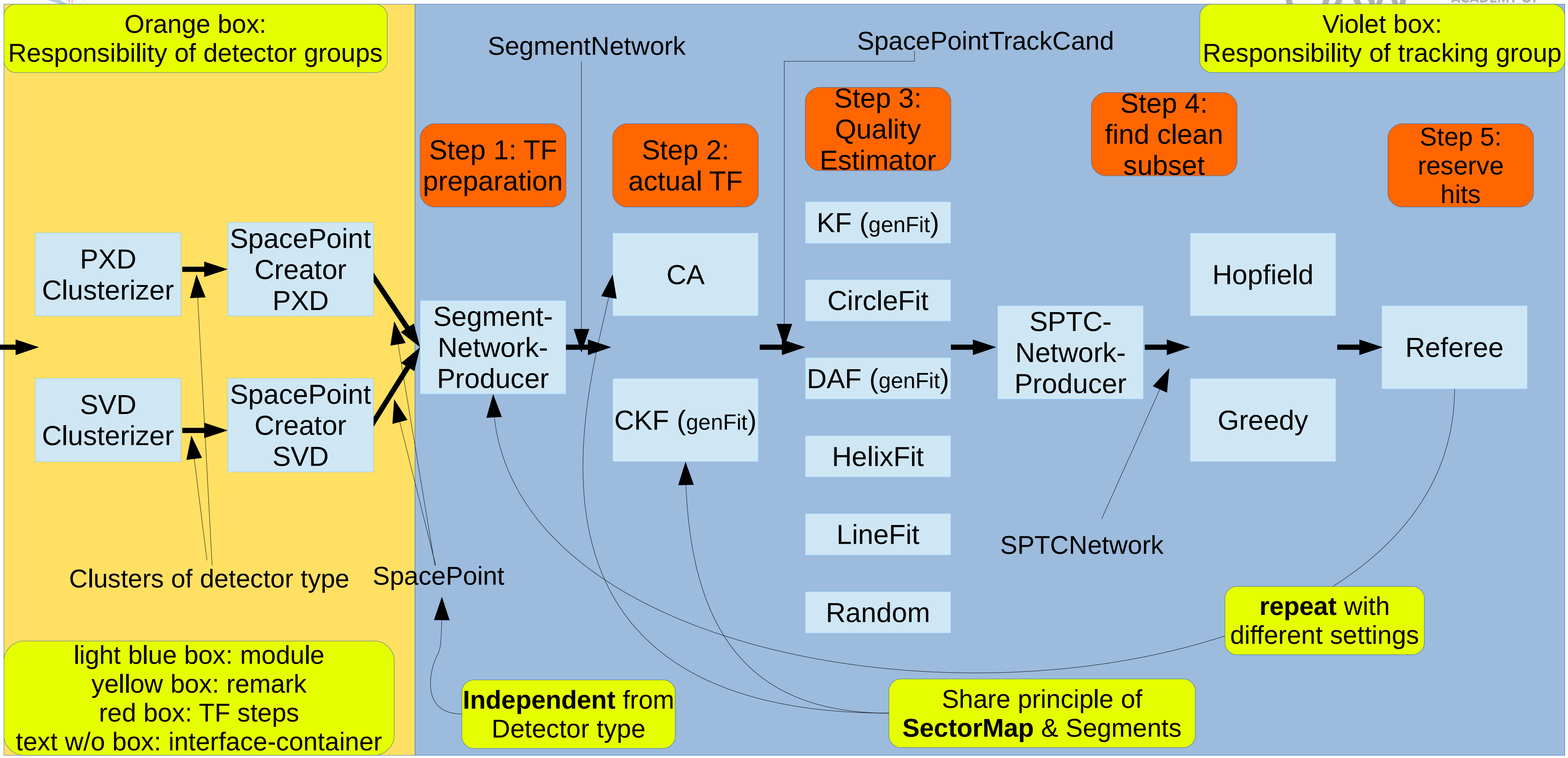
- Implemented in standard reconstruction chain (used for MC and other studies) since 2013
- Successfully used in combined (VXD, DAQ and HLT) beam-test in 2014 (see picture and proceedings) (T.Bilka *et al.*, ``Demonstrator of the Belle II Online Tracking and Pixel Data Reduction on the High Level Trigger System,'' IEEE Trans. Nucl. Sci. **62** (2015) 3, 1155 [arXiv:1406.4955])
- About 2 years to go, so we are fine?

→ No, we refactor the code now

# VXDTF – refactoring goals

- We want to simplify debugging:
  - High coverage of unit- and integration tests
  - High flexibility using modular design (CA, CKF, DAF, other filters replaceable)
  - Allow workload to be shared by 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)

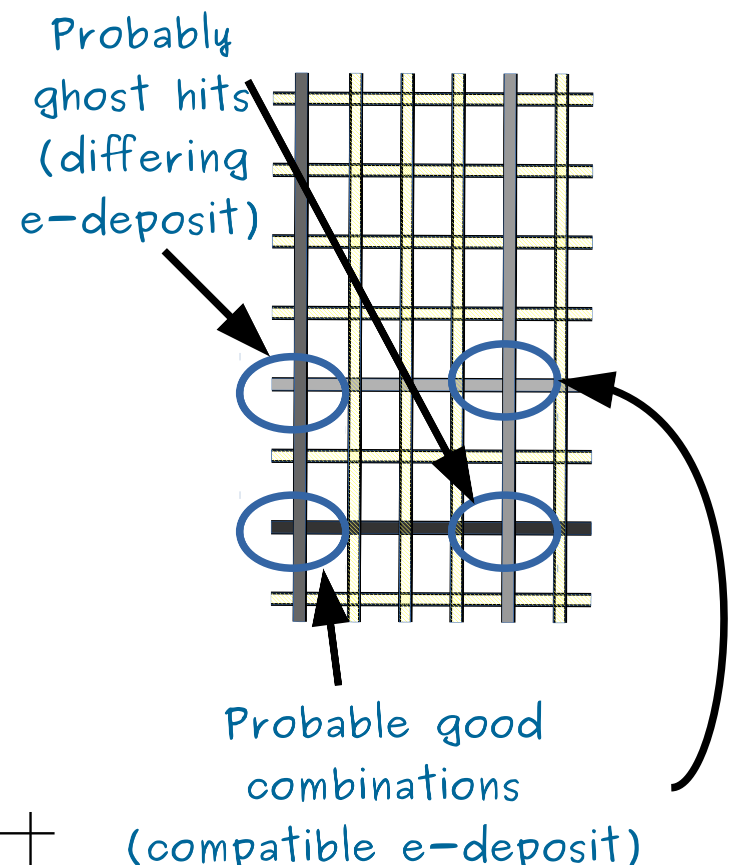
# Planned structure for the VXDTF (event-part)

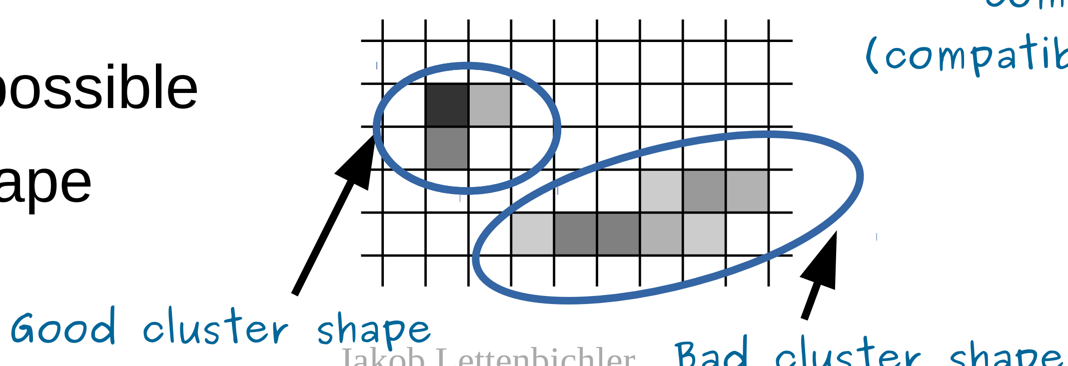



- CA: Cellular Automaton
- KF: Kalman Filter
- CKF: Combinatorial KF
- DAF: Deterministic Annealing Filter
- Hopfield: a neural network of Hopfield type
- SPTC: SpacePointTrackCandidate

# 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 down 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 needs a different subGraph of that SectorMap
    → Sectors having SpacePoints in that 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):

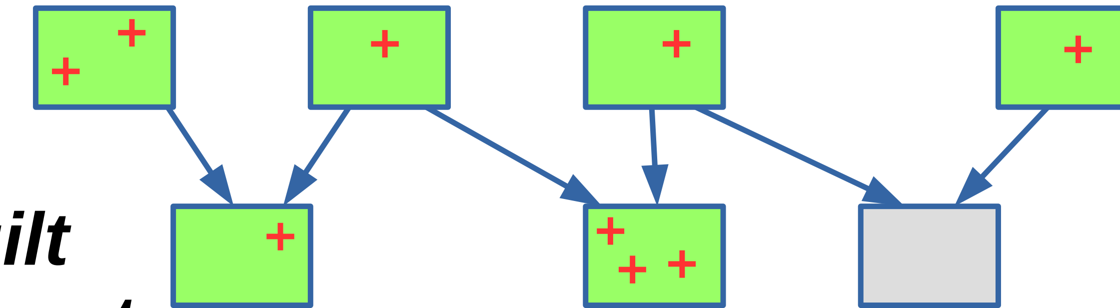
- Objects are related forming a network, where objects are treated as Nodes and Links/Edges indicate their compatibility
  - Nodes can carry anything (Sectors, SpacePoints, segments, basic types, ..) as „node-entries“
  - Only following requirements to node-entries:
    - Storabe in a std::vector
    - '==' 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 carry no extra info to minimize overhead

# SegmentNetworkProducer III

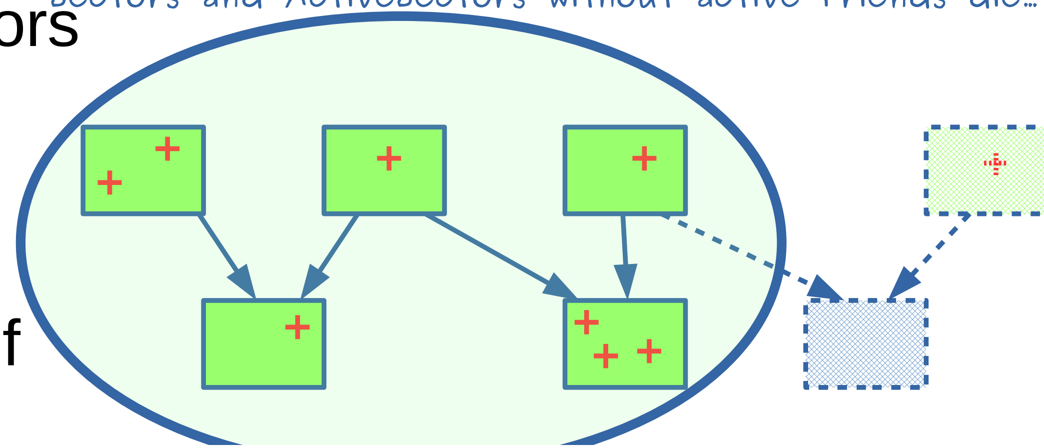
In Action (for each event):

- SpacePoints are matched to their Sectors → ActiveSectors store event-dependent information
- ***ActiveSectorNetwork: built from Sectors which have got hits in that event***
  - only compatible (Active-)Sectors are linked
  - '1'-SpacePoint-filter: only physically relevant hits can form a sufficiently long chain of ActiveSectors
  - Serves as input for the *SpacePointNetwork*

SpacePoints (+) sorted in Sectors (▭), which become ActiveSectors (▭)

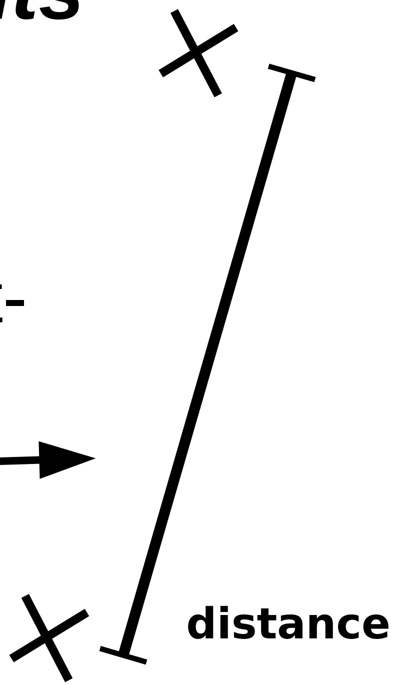


Only ActiveSectors come into the ActiveSectorNetwork. Sectors and ActiveSectors without active Friends die…

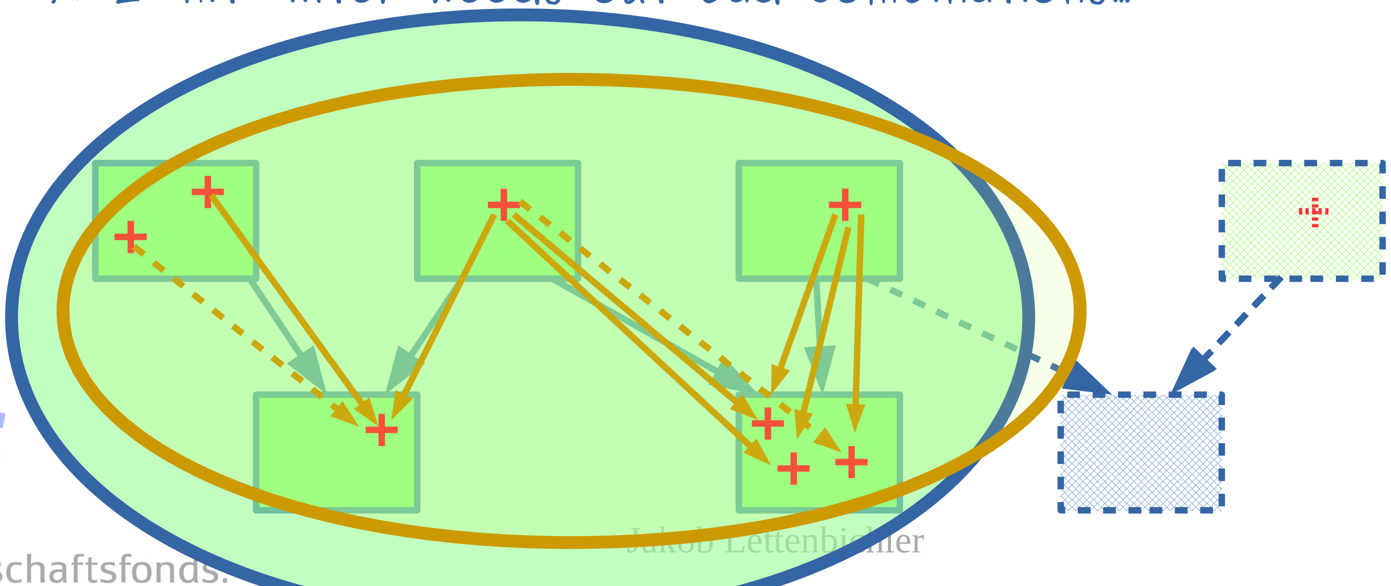


# SegmentNetworkProducer IV

- ***SpacePointNetwork: built from SpacePoints retrieved from ActiveSectors of the ActiveSectorNetwork***
  - Two SpacePoints get linked, when 2-SpacePoint-tests are passed (e.g. distance3D)
  - '2'-SpacePoint-filter
  - Serves as input for the *SegmentNetwork*

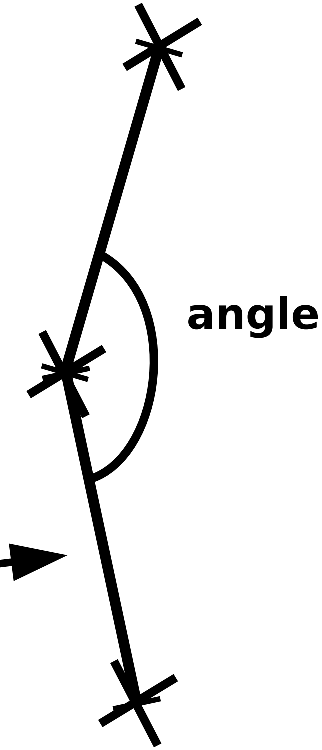



Only SpacePoints of FriendSectors are tested.
A 2-hit-filter weeds out bad combinations...



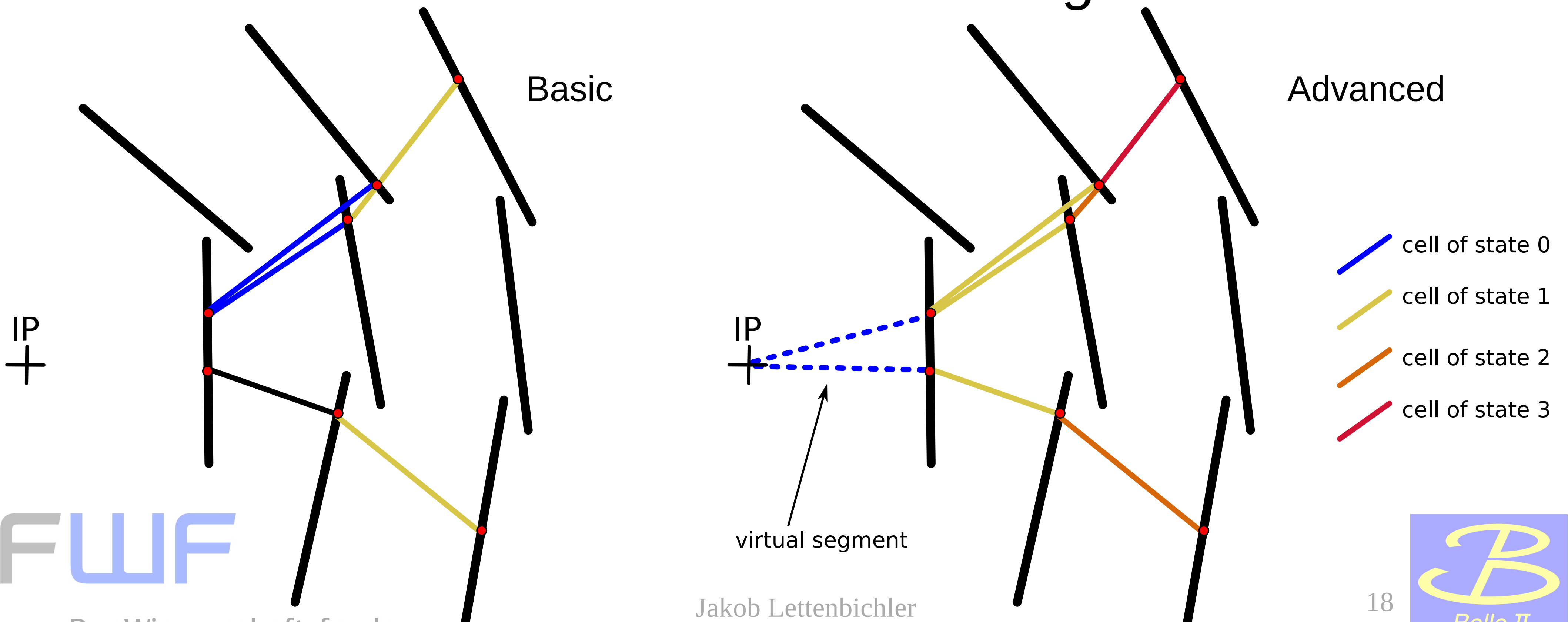
# SegmentNetworkProducer V

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



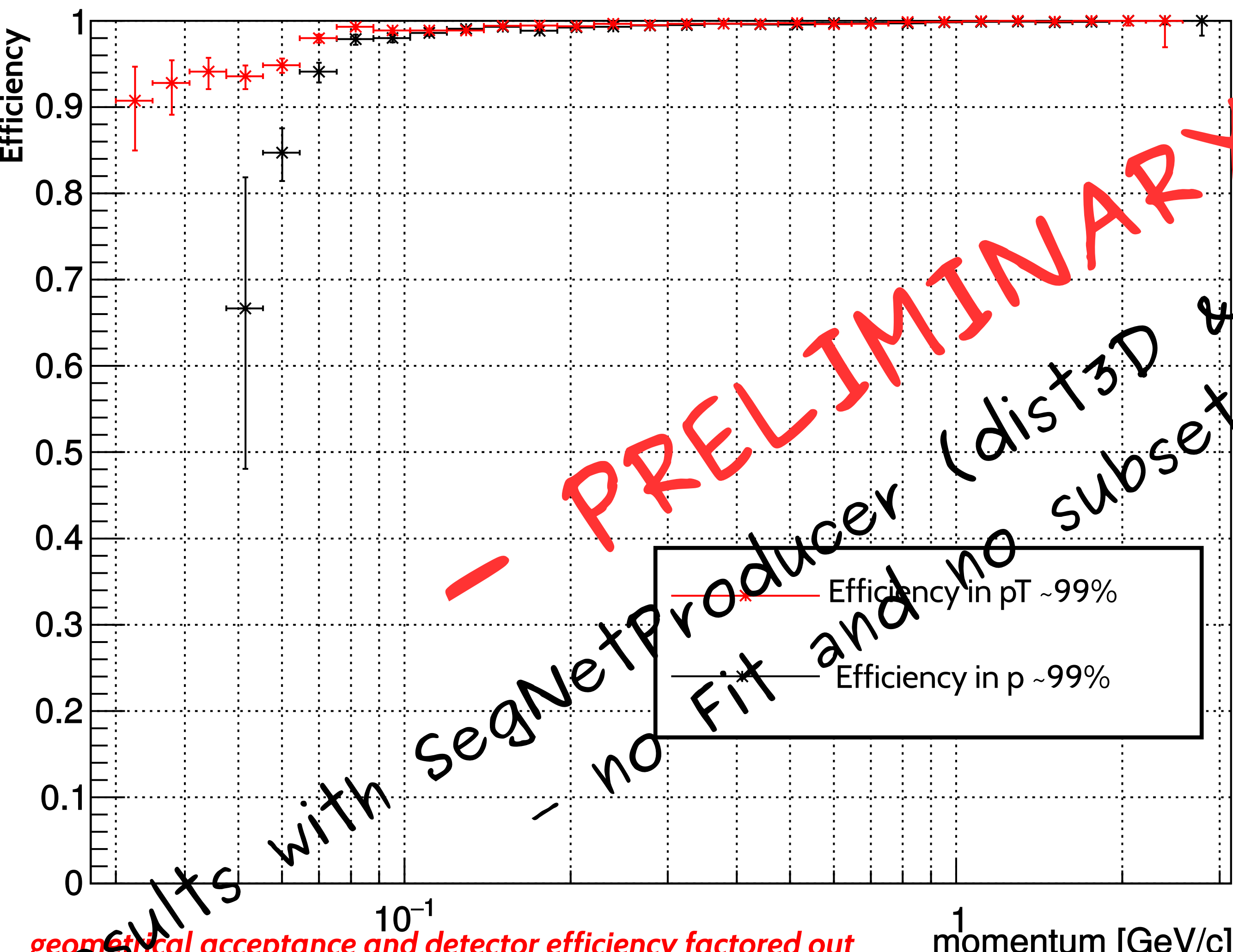
# CA – the technical details

- First the neighbourhood is defined (SegmentNetworkProducer), then the CA is applied
- Segments (of the SegmentNetwork) are used as Cells
- Virtual interaction point connected to innermost hits used as innermost Cells → *virtual segment*

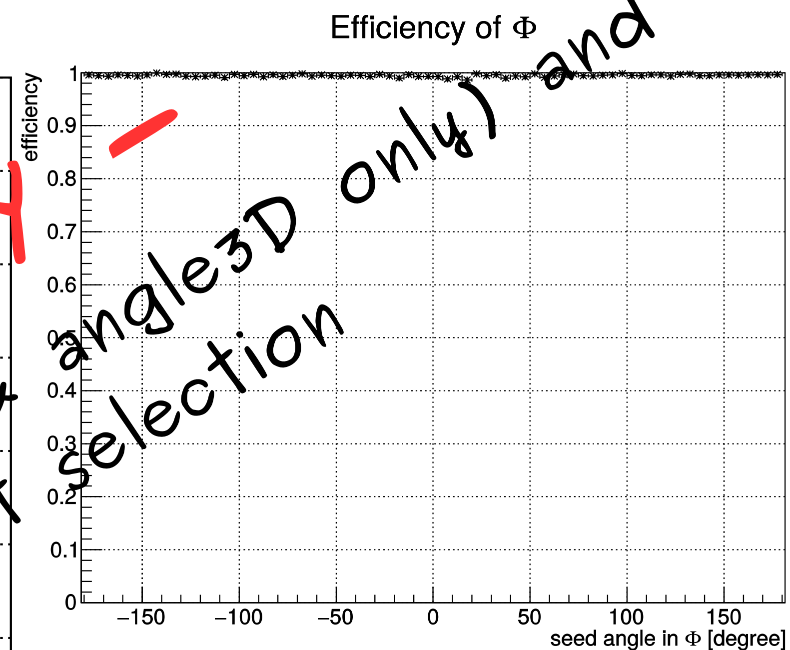


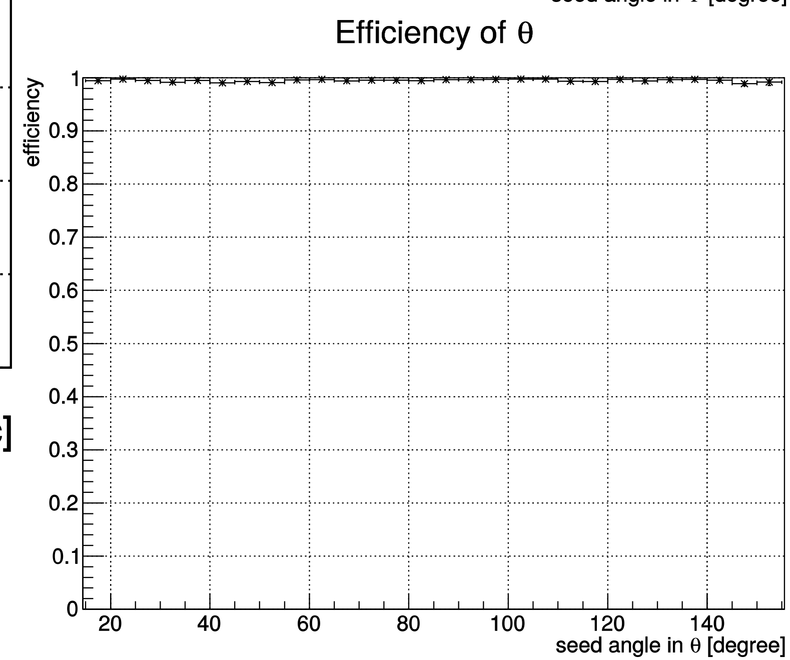
# VXDTF2 – current performance

## Efficiency vs momentum




*geometrical acceptance and detector efficiency factored out*




– PRELIMINARY –

Results with SegNetProducer (dist3D & angle3D only) and CA

– no Fit and no subset selection

Fake rate: **2948.04%** - clone rate: **844.57%**

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

# VXDTF2 – next steps

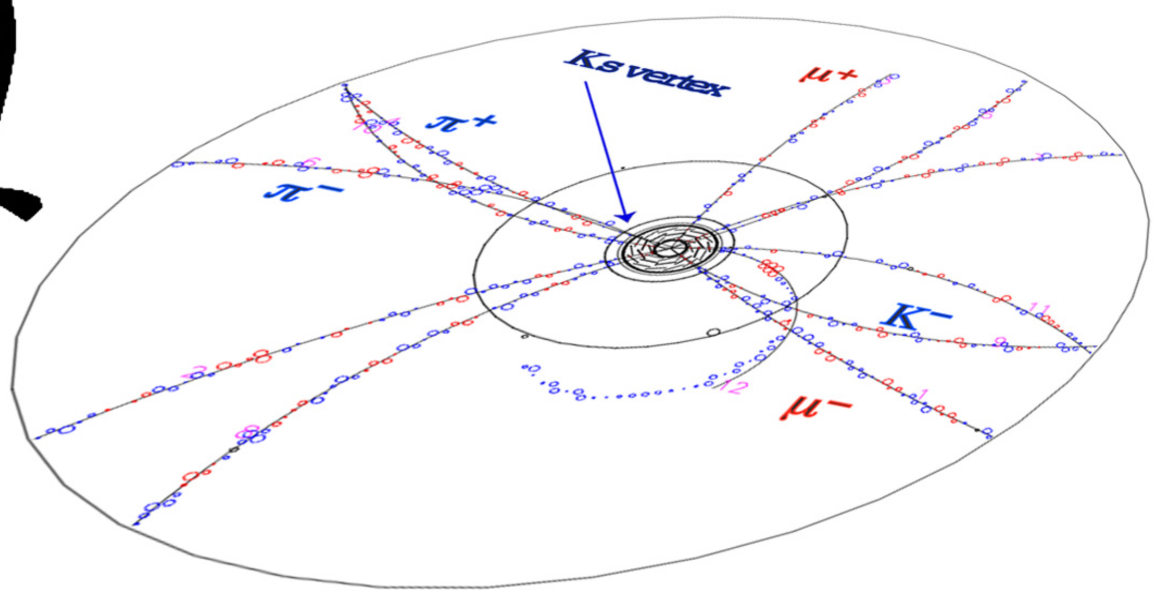


- Combined beam test 2016
- Finishing VXDTF2 draft stage and implementing proof-of-concepts for CKF and DAF
- Tons of studies

# Thanks for listening!

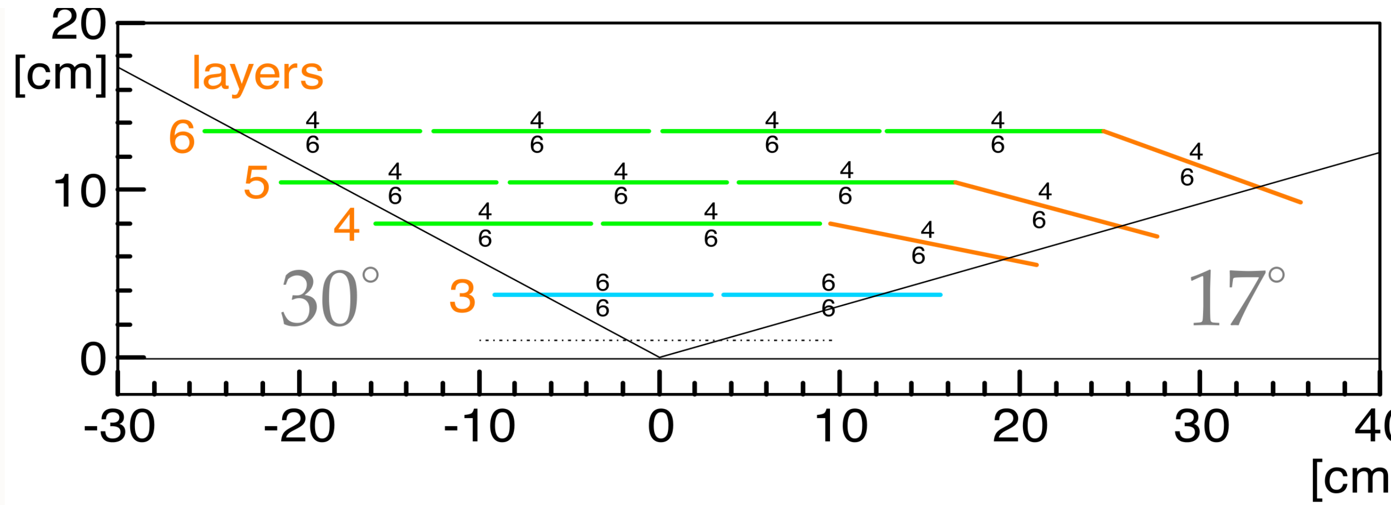
- Many thanks to all members of the Tracking group, especially:
  - Rudolf Frühwirth
  - Eugenio Paoloni
  - Martin Heck
  - Martin Ritter and Christian Pulvermacher
  - Thomas Madlener
  - Tobias Schlüter
  - Giulia Casarosa

That's all folks!



# The SVD in detail

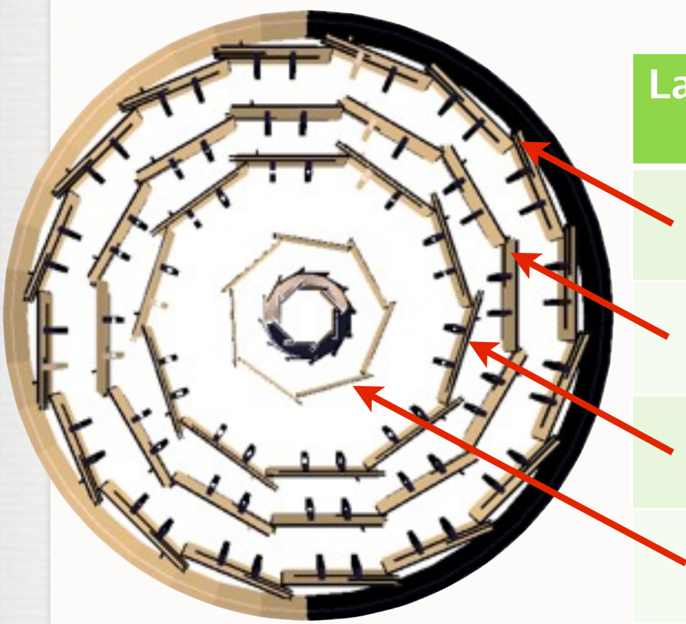
- 4 layers of double sided silicon strip detectors
- Lampshade geometry for layers 4, 5 & 6
- Individual readout on each silicon sensor
- Very light mechanical structure




z APVs / rphi APVs — Rect (122.8 x 38.4 mm$^2$, 160 / 50 um pitch)

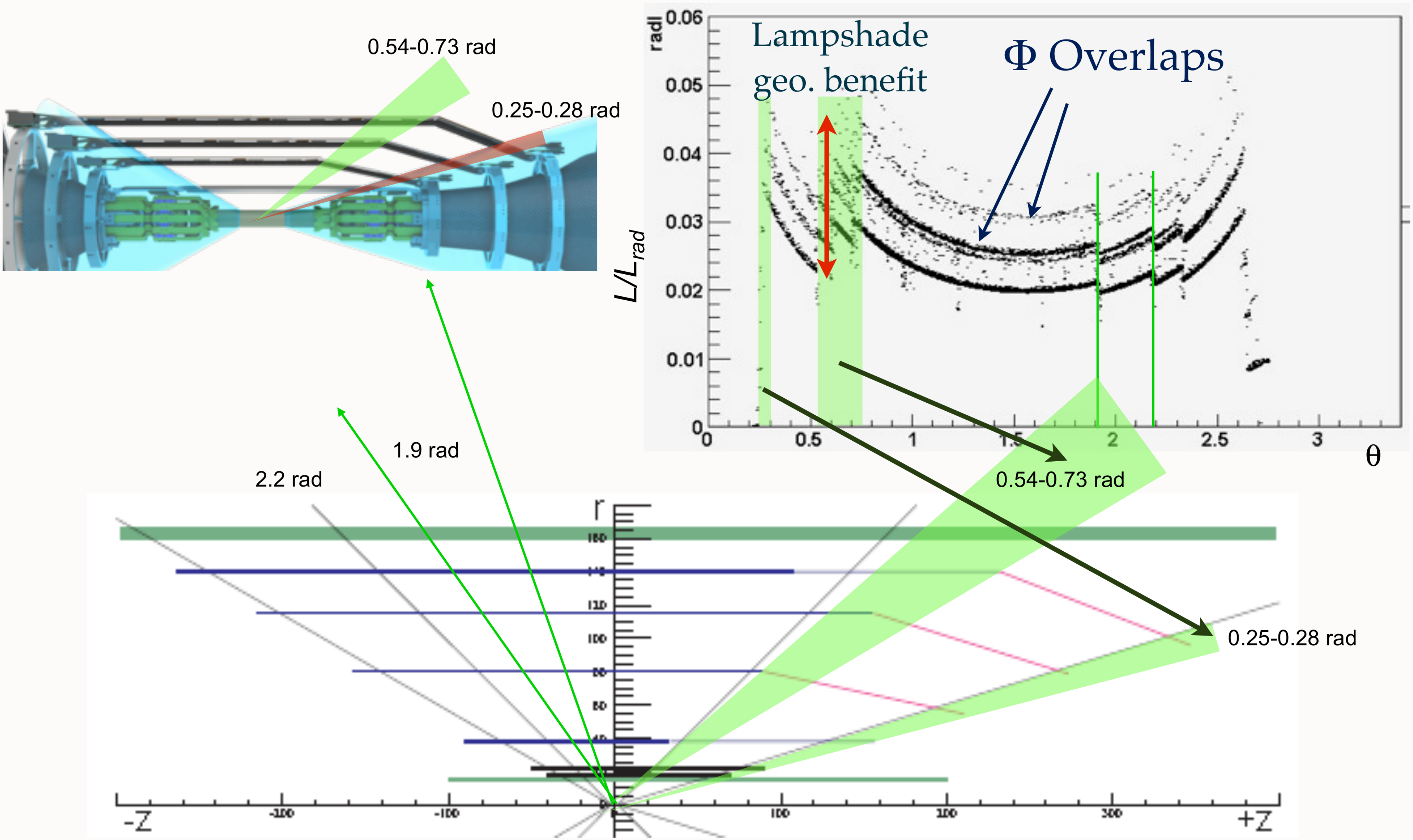
z APVs / rphi APVs — Rect (122.8 x 57.6 mm$^2$, 240 / 75 um pitch)

z APVs / rphi APVs — Wedge (122.8 x 57.6-38.4 mm$^2$, 240 / 75..50 um pitch)



| Layer | Avg.Radius (mm) | Ladders | Sensors / Ladder | Slanted? | Windmill angle [°] | Overlap [%] |
|---|---|---|---|---|---|---|
| 6 | 135 | 16 | 5 | ✓ | 7 | 10.8 |
| 5 | 105 | 12 | 4 | ✓ | 5 | 5.1 |
| 4 | 80 | 10 | 3 | ✓ | 6 | 17.6 |
| 3 | 39 | 7 | 2 | x | 6 | 5.9 |

# Backup: X/X0 in detail



# Material effects in Detail

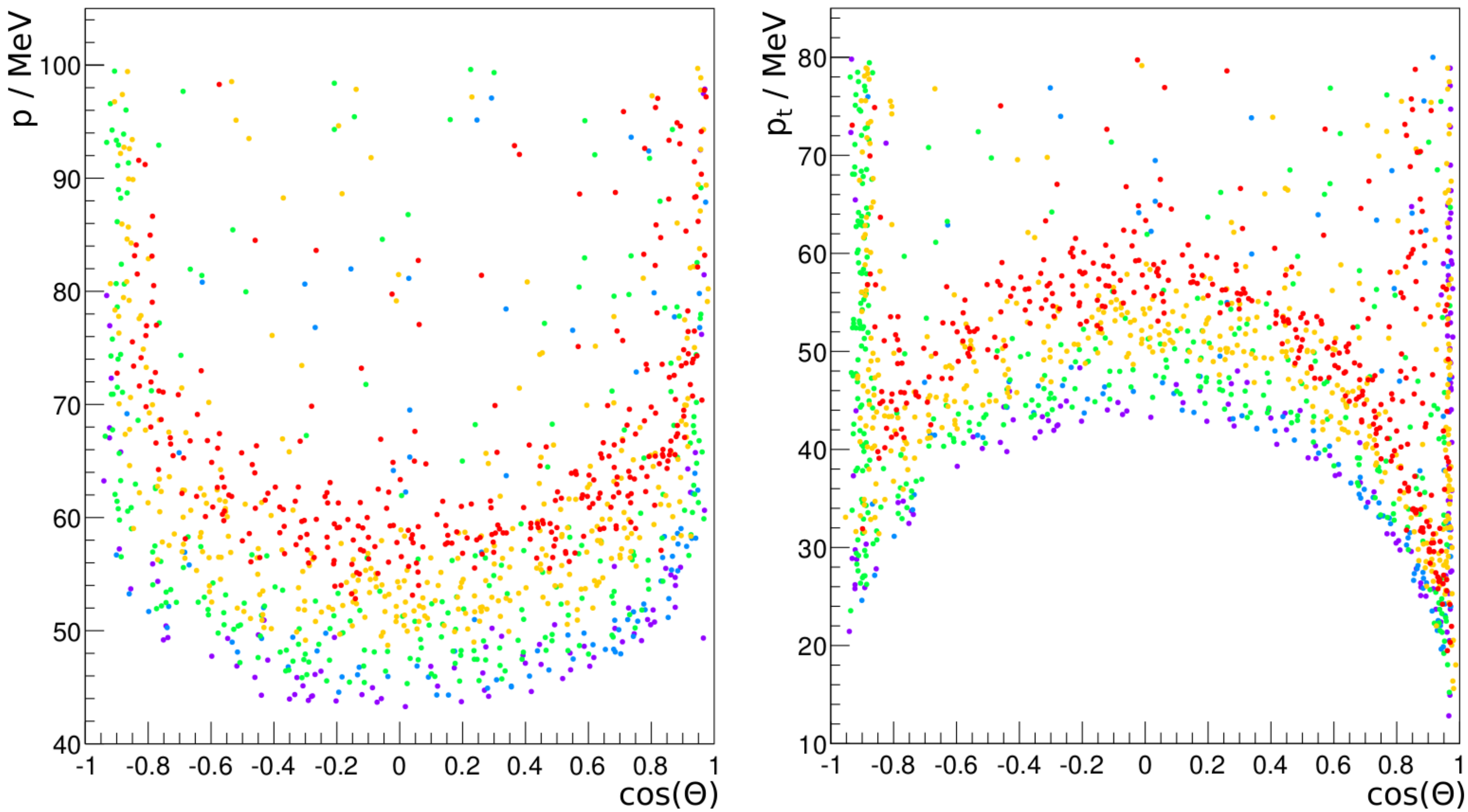



Figure 5.4.: Total and transverse momentum of simulated pions that do **not** reach the last SVD layer, over $\cos\theta = p_z/p$. Colours indicate the outermost layer reached: violet and blue for PXD layers 1 and 2; green, yellow and red for SVD layers 1, 2, and 3, respectively.

# Backup: Distributions of tracks