Study of Tracking and Flavor Tagging with FPCCD Vertex Detector

Tatsuya Mori, Daisuke Kamai, Akiya Miyamoto[A], Yasuhiro Sugimoto[A], Akimasa Ishikawa, Taikan Suehara[B], Eriko Kato, Hitoshi Yamamoto

Tohoku University, KEK[A], Kyushu University[B]

Outline

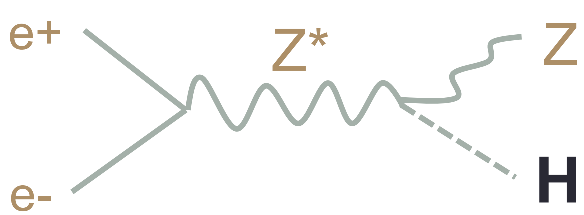
1. Introduction to FPCCD
2. Development of FPCCD Track Finder
3. Tracking Efficiency
   - with and without pair BGs
4. Flavor Tagging
5. Summary

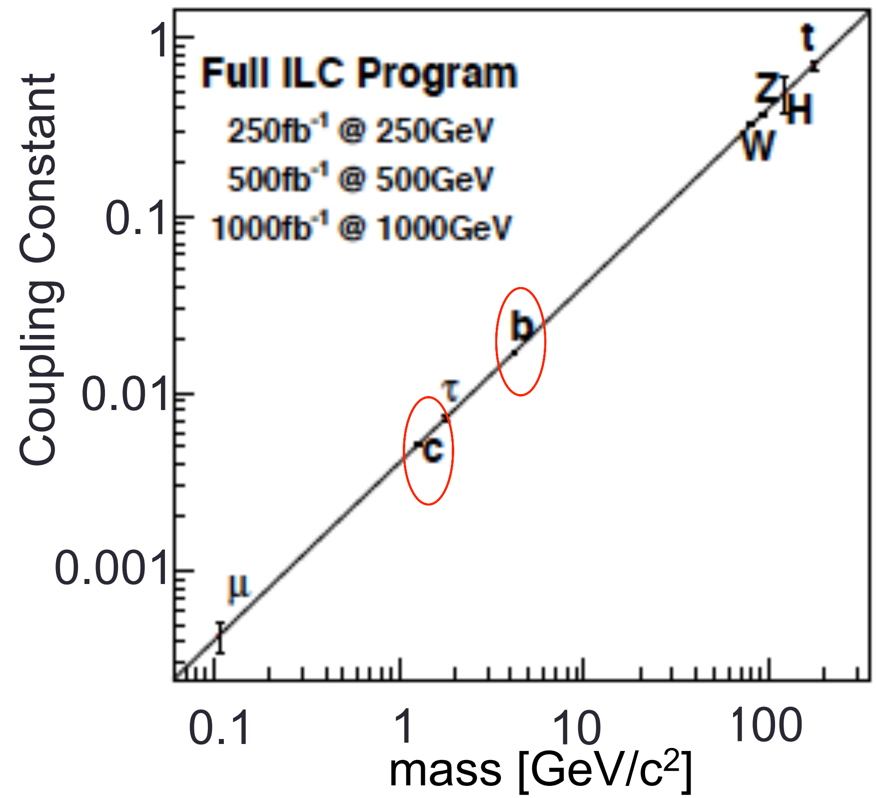
Introduction to FPCCD

Role of Vertex Detector

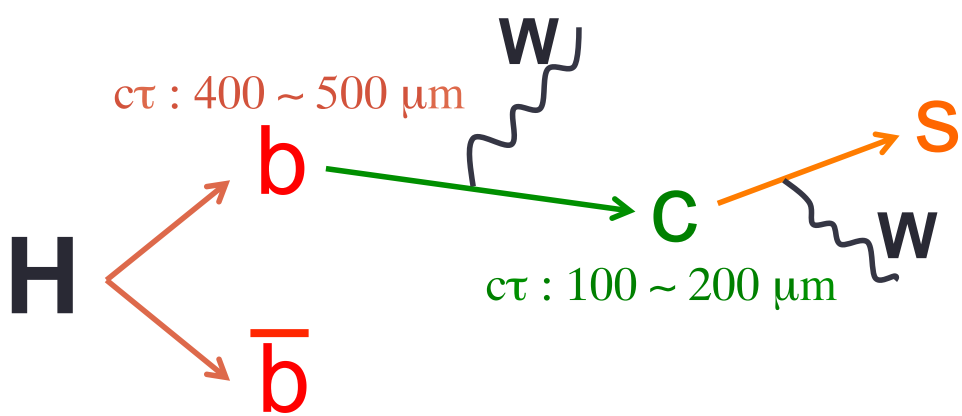
one of the ILC physics goals :

Precise measurement of Higgs coupling constant to “c, b-quark, gluon”





Precise identification of $H \rightarrow b\bar{b}$, $c\bar{c}$, gg is required



We need VXD with high performance

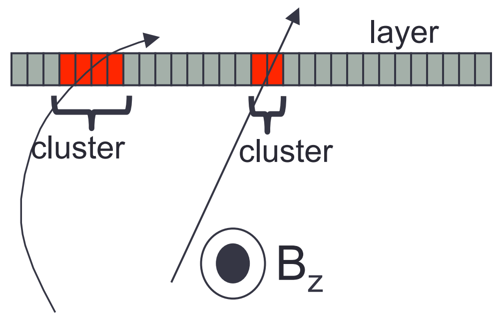
FPCCD Vertex Detector

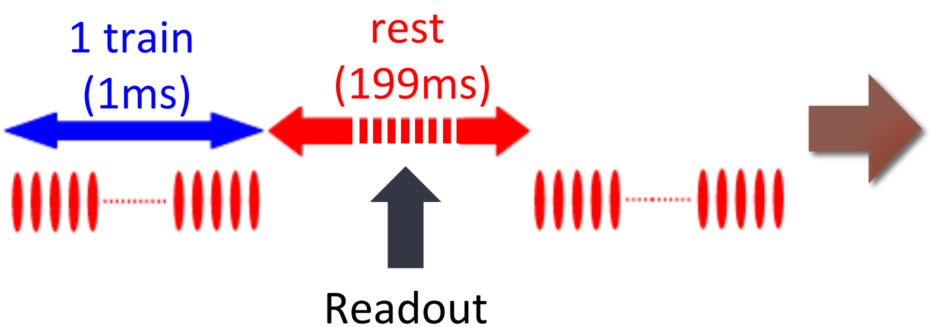
FPCCD (Fine Pixel CCD) Features

- Small pixels: 5-10 μm (see right)
- Sensitive / Total thickness: 15 / 50 μm
- # of pixels : ~ 0.4 x 10^9
- Possible to see cluster shape for:
  - ✓ Extrapolation of tracks
  - ✓ Improvement of position resolution
  - ✓ Discrimination : BG cluster & signal cluster
- Readout between trains :
  - All bunches in a train are accumulated

Geometry

layer	distance from IP (mm)	pixel size (μm²)
0, 1	16 , 18	5 × 5
2, 3	37 , 39	10 × 10
4, 5	58 , 60	10 × 10



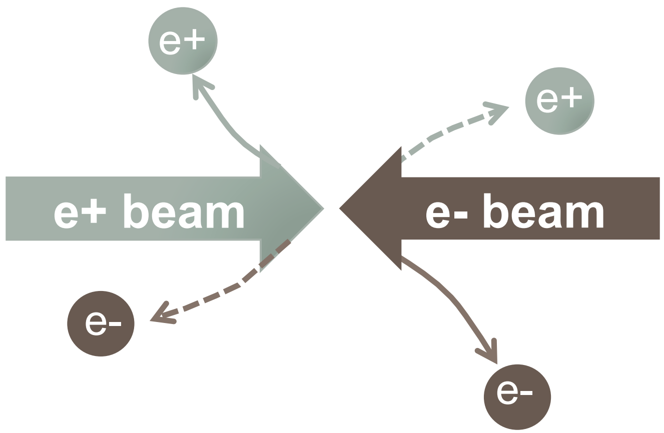


pro :
Noise from Electromagnetic Interference (EMI) can be ignored

con :
Tracking is challenging due to so many hits

Occupancy and Impact Parameter Resolution

- Dominant BG : **e+e- pair BG**



(reported in ECFA 2013)

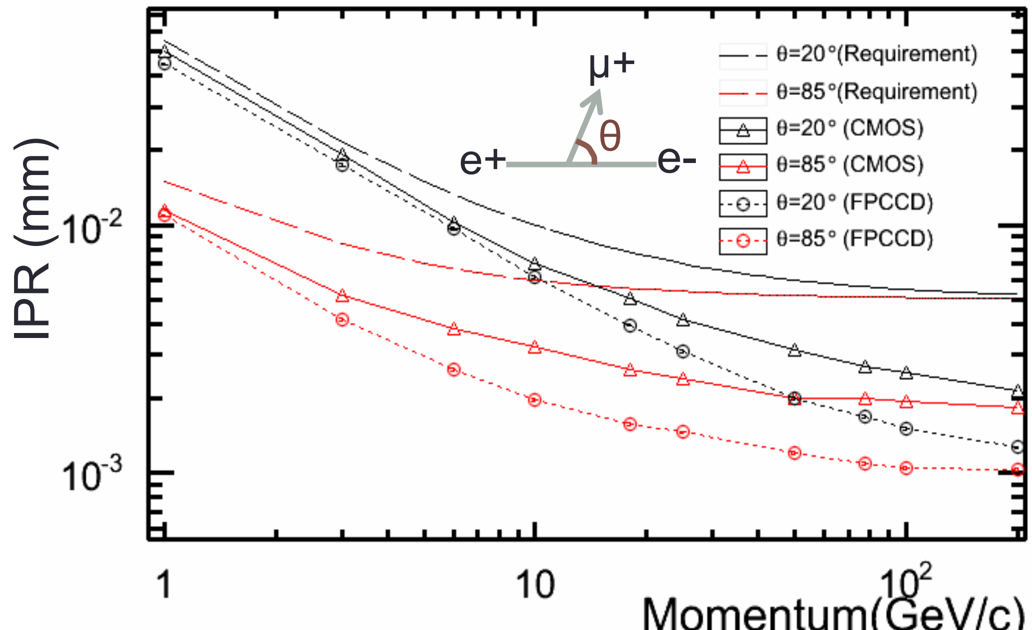
E_{CM} (GeV)	occupancy in 0th layer (%)
250	0.8
350	0.9
500	2.8
1000	19.6

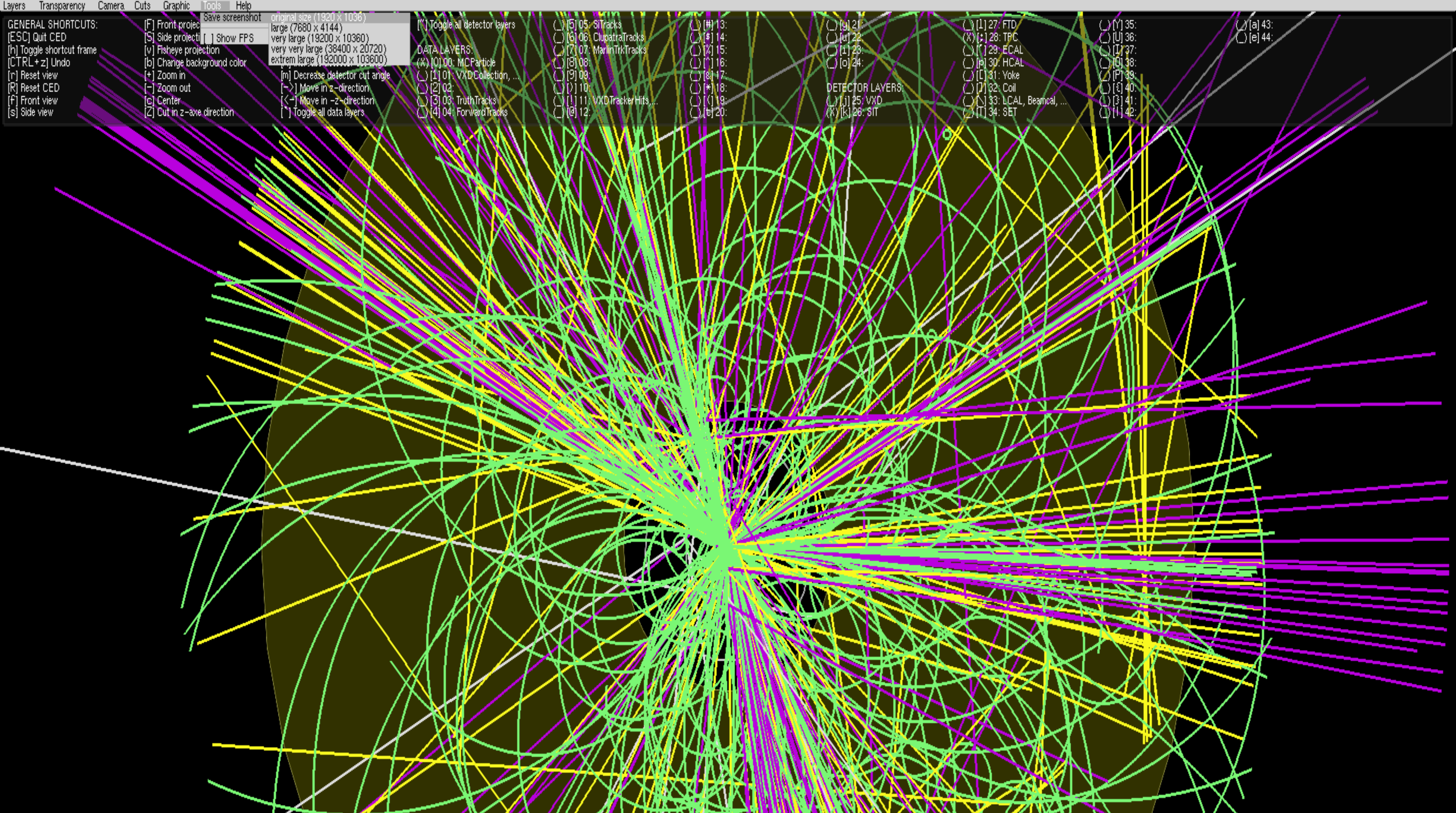
- Performance goal of Impact Parameter Resolution (IPR)

$$\sigma_{r\phi} = 5\mu m \oplus \frac{10 \mathrm{GeV/c}}{p \cdot \sin^{3/2}\theta} \mu m$$

→**Satisfied** and
IPR ~ **1 μm** in high P region

(reported in ECFA 2013)





Development of FPCCD Track Finder

ILD Tracking Algorithm for DBD study

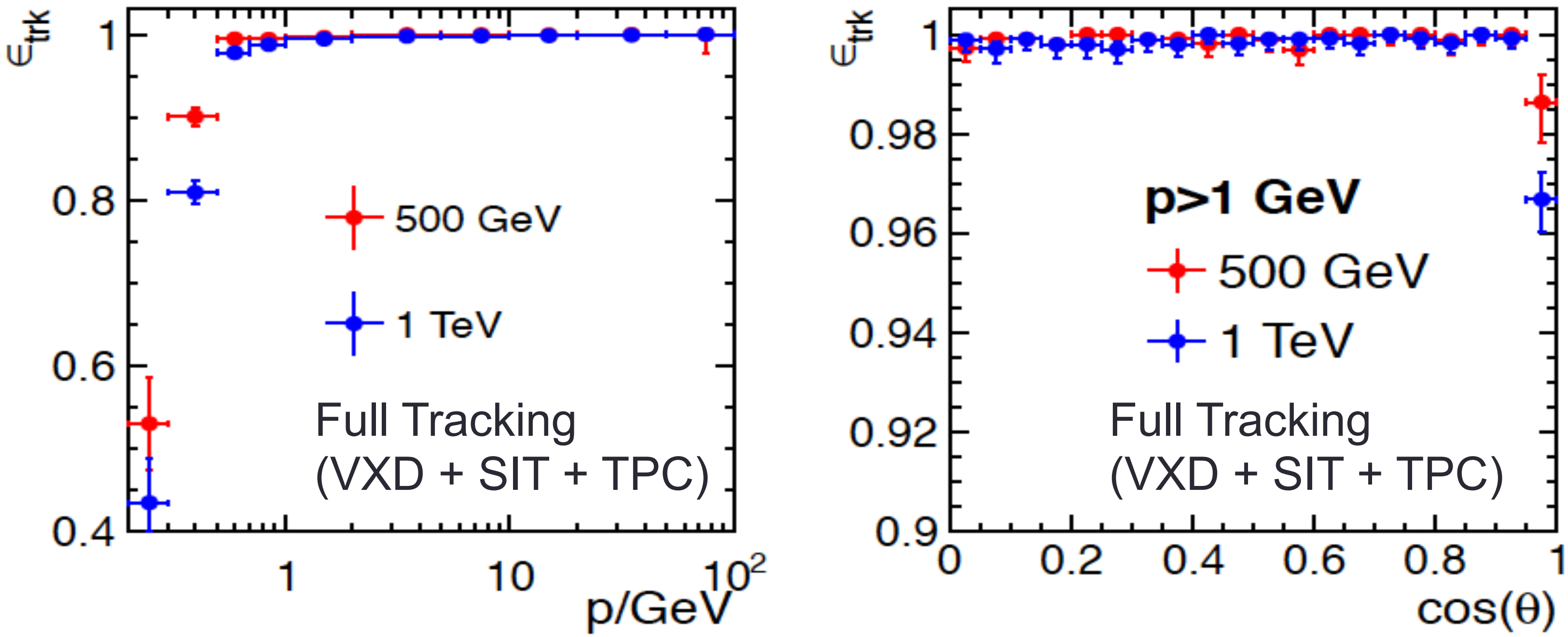
stand-alone Silicon Tracking:

- VXD + SIT
- Outside-in tracking algorithm
- Track seeding with 24 layer-combinations

Current ILD VXD Configuration for DBD (**current VXD sim.**) **FPCCD**

layer	distance from IP (mm)	position resolution (μm) (current VXD sim.)	position resolution (μm) (FPCCD)
0, 1	16 , 18	2.8 / 6.0	1.4 / 1.4
2, 3	37 , 39	4.0 / 4.0	2.8 / 2.8
4, 5	58 , 60	4.0 / 4.0	2.8 / 2.8

DBD Study on Tracking Efficiency



Some of the counted tracks have imprecise impact parameter resolution because those tracks may not have enough VXD hits due to the requirement

Having VXD hits is crucial for flavor tagging

Current ILD Tracking with FPCCD

Tracking Efficiency : η ≡

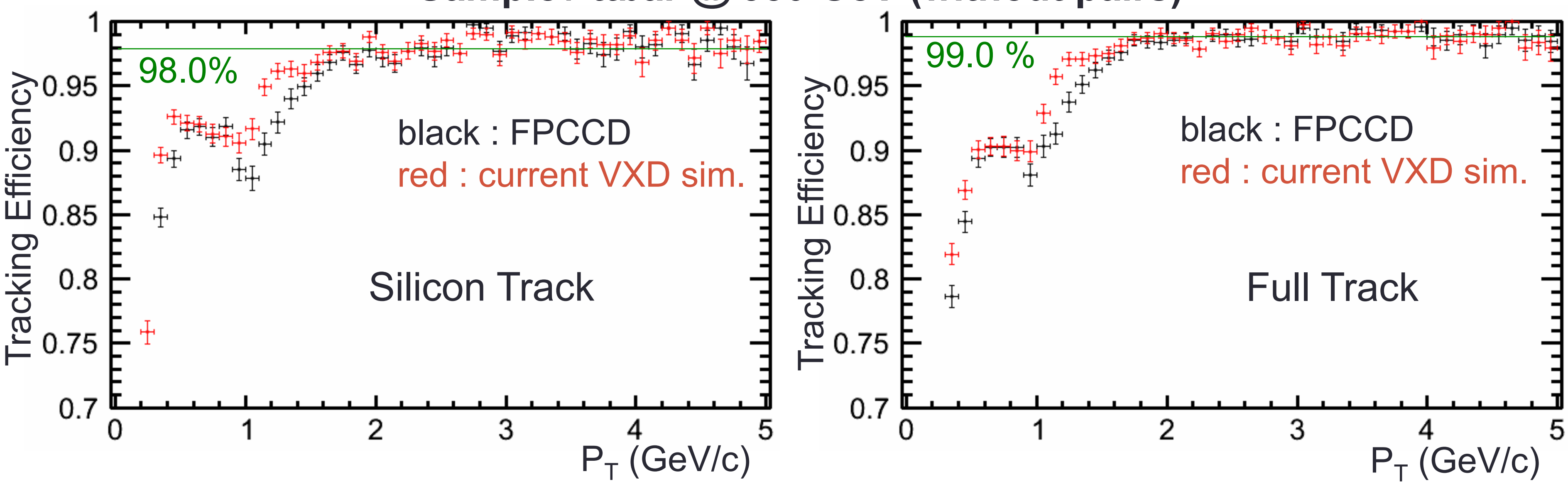
$$\eta \equiv \frac{\text{\# of tracks with VXD hits} >= 5 \ \&\& \ \text{track purity} > 75\%}{\text{\# of MCParticles creating VXD sim-hits} >= 6 \ \&\& \ \text{SIT sim-hits} >= 4}$$

Note: P_{Tmin} to reach TPC
R_{in} : 0.4 GeV/c
R_{out} : 1.8 GeV/c

track purity:

$$\frac{\text{(\# of the MCP's hits of track)}}{\text{(\# of all hits of track)}}$$

Sample : ttbar @ 350 GeV (without pairs)



Efficiency : degraded @ P_T < 1.7GeV/c

→ **Improvement of Silicon Tracking are needed**

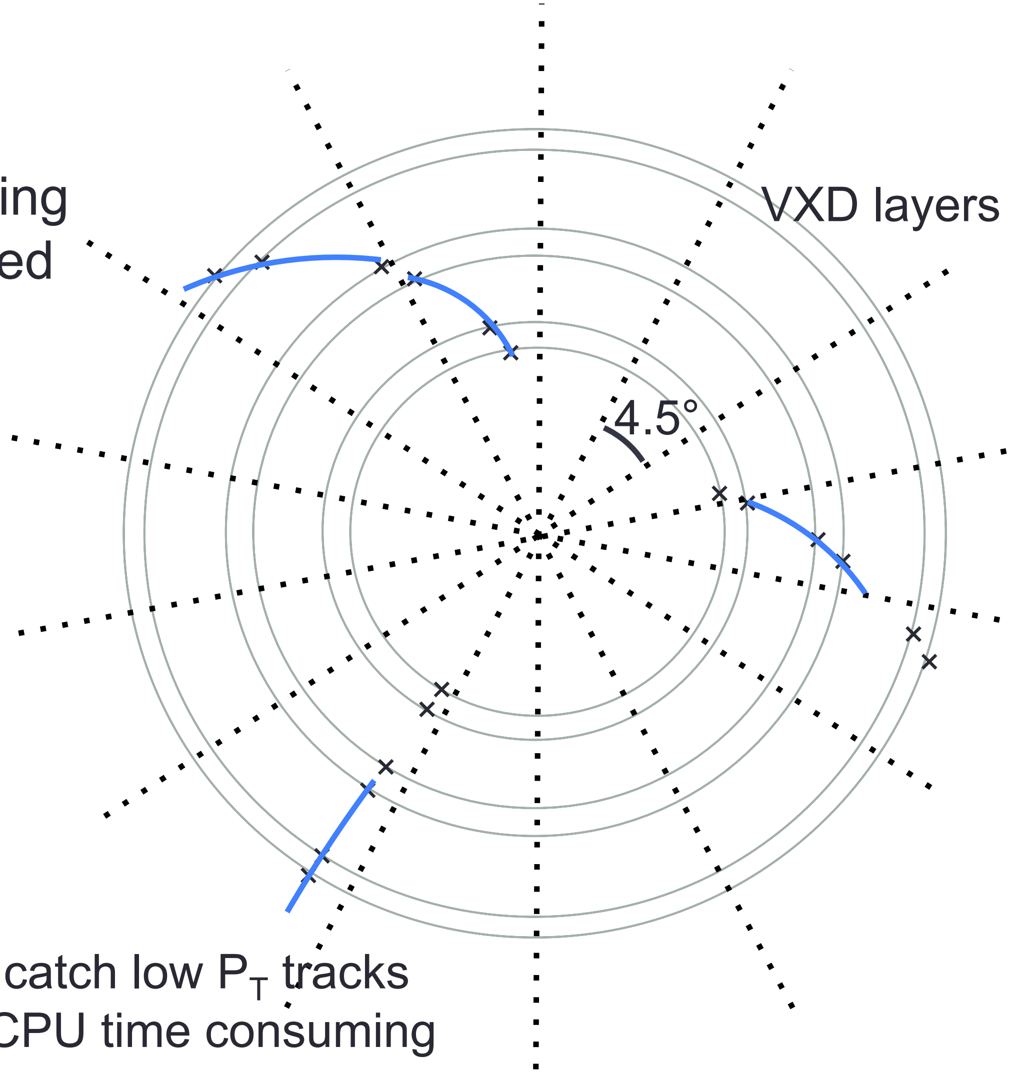
Problems in Silicon Tracking (Track Seed)

Track Seed

Track seeds are generated by combining 3 hits on the 3 layers in Φ sector divided into 80 (4.5°)

- the 3 layers (SIT: 8, 6 VXD: 5~0)

8 6 5	8 6 4	8 6 3	8 6 2	8 5 3	8 5 2
8 4 3	8 4 2	6 5 3	6 5 2	6 4 3	6 4 2
6 3 1	6 3 0	6 2 1	6 2 0	5 3 1	5 3 0
5 2 1	5 2 0	4 3 1	4 3 0	4 2 1	4 2 0



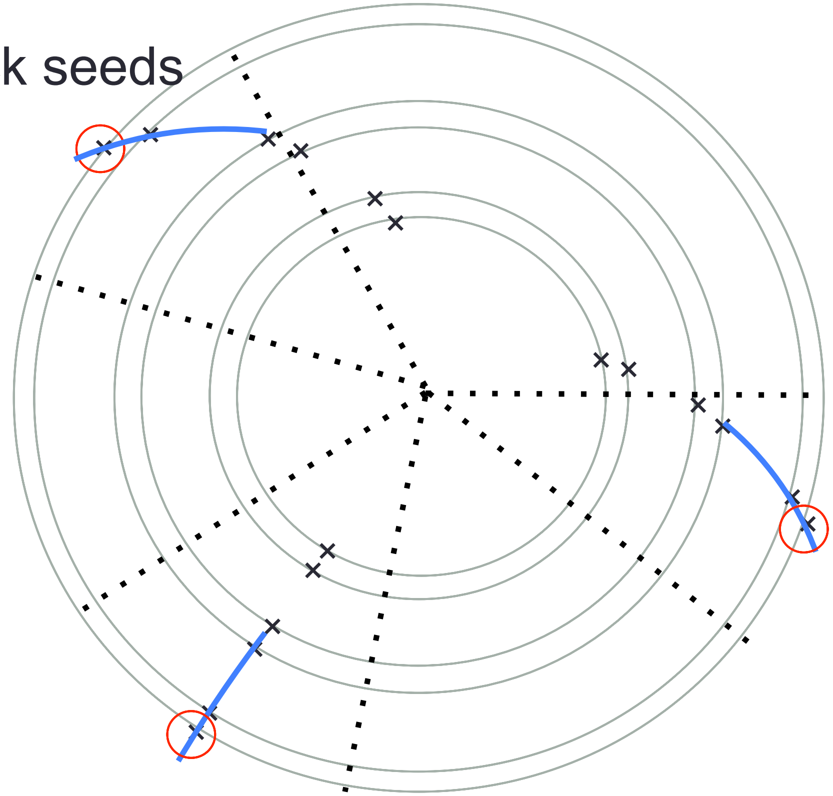
Problems in Track Seed

- 4.5° search windows are too narrow to catch low P_T tracks
  - wider? → larger ghost seeds and CPU time consuming
- # of seeds is too many, especially using inner-most doublet for FPCCD
  → larger ghost seeds and CPU time consuming

Solutions for Track Seed

Solutions:

- Search window enough wide to cover track seeds generated with P_T > 0.18 GeV/c tracks is calculated from a hit on the outer layer



- Combinations of 3 layers are reduced as

  Old:

  8 6 5 8 6 4 8 6 3 8 6 2 8 5 3 8 5 2
  8 4 3 8 4 2 6 5 3 6 5 2 6 4 3 6 4 2
  6 3 1 6 3 0 6 2 1 6 2 0 5 3 1 5 3 0
  5 2 1 5 2 0 4 3 1 4 3 0 4 2 1 4 2 0

  New:

  8 6 5 8 6 4 8 5 4 6 5 4 5 4 3

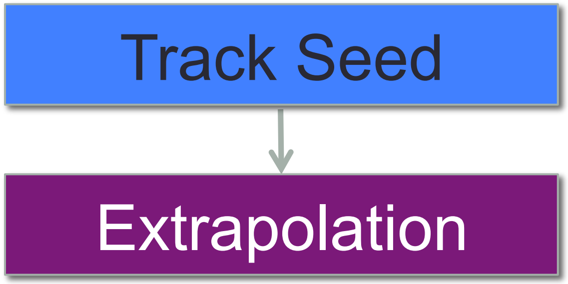
  Inner-most doublet is not used to reduce ghost seeds

Implemented in FPCCD Track Finder

Results:

We can reduce both CPU time and ghost tracks, and catch low P_T tracks

Problems in Silicon Tracking (Extrapolation)

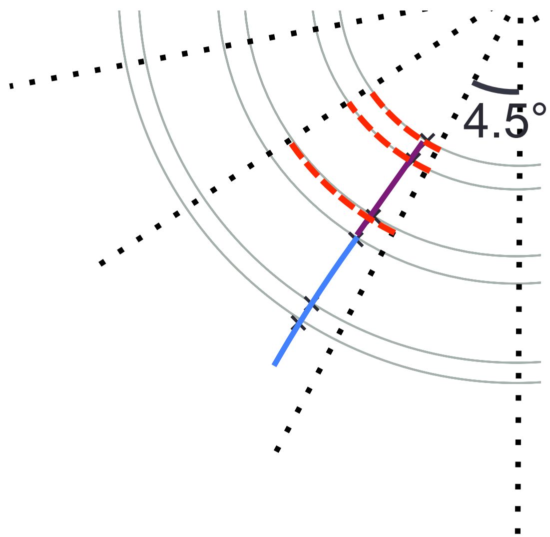


Window for extrapolation :
divided by 4.5° in the direction of Φ

Fitter :
Simple Helix Fit

Problems in Extrapolation

- Tracks are not extrapolated to neighboring Φ sector
  → Some true hits are ignored
- Φ window is fixed
  → Many false hits are considered
- Fitter is Simple Helix Fit
  → Chi2 of some low P_T tracks is too high due to not considering multiple-scattering → rejected

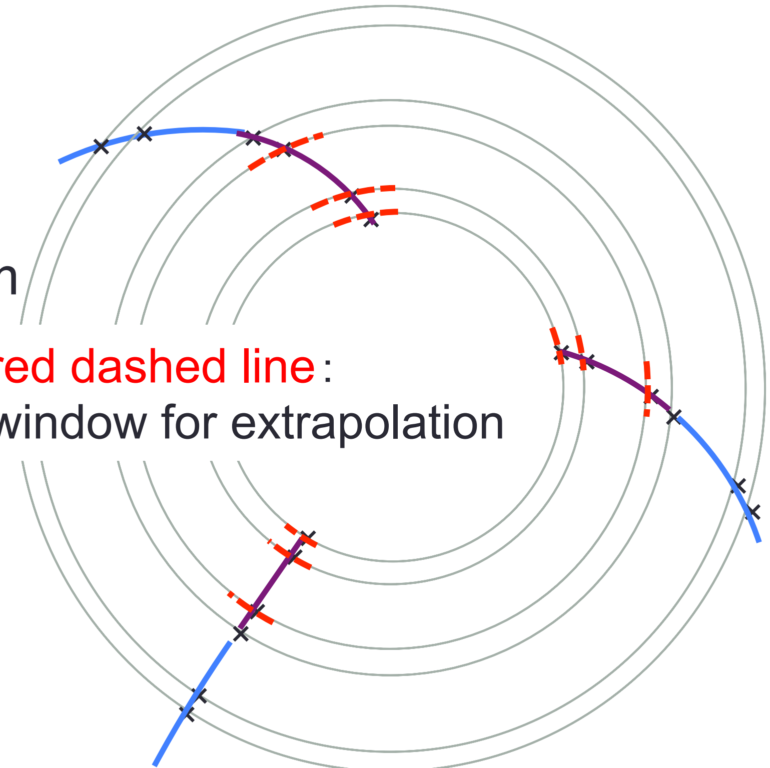


red dashed line : window for extrapolation

Solutions for Extrapolation

Solutions:

- Kalman Filter is used as Fitter
- Window for extrapolation is determined from track parameters calculated by the fitter



Implemented in FPCCD Track Finder

Results:

- Flexible window for extrapolation can catch true hits and avoid taking most of false hits

- Chi2 of low P_T tracks is calculated more properly
  → low P_T tracks can survive

Tracking Efficiency

FPCCD Track Finder VS Current Tracking with FPCCD (P_T)

Tracking Efficiency : η ≡

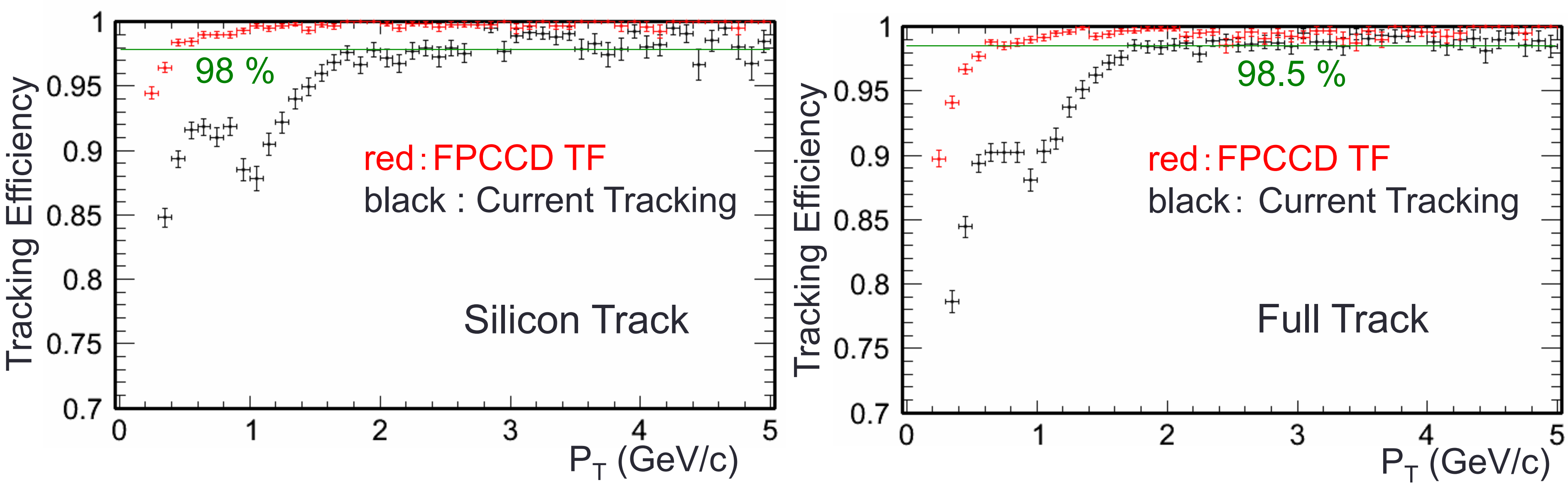
$$\eta \equiv \frac{\text{\# of tracks with VXD hits >= 5 \&\& track purity > 75\%}}{\text{\# of MCParticles creating VXD sim-hits >= 6 \&\& SIT sim-hits >= 4}}$$

Note: P_{Tmin} to reach TPC
R_{in} : 0.4 GeV/c
R_{out} : 1.8 GeV/c

track purity:

$$\frac{\text{(\# of the MCP's hits of track)}}{\text{(\# of all hits of track)}}$$

Sample : ttbar @ 350 GeV (without pairs)



Efficiency : ~ 99 % @ P_T > 0.6 GeV/c

FPCCD Track Finder VS Current Tracking with FPCCD (cosθ)

Tracking Efficiency : η ≡

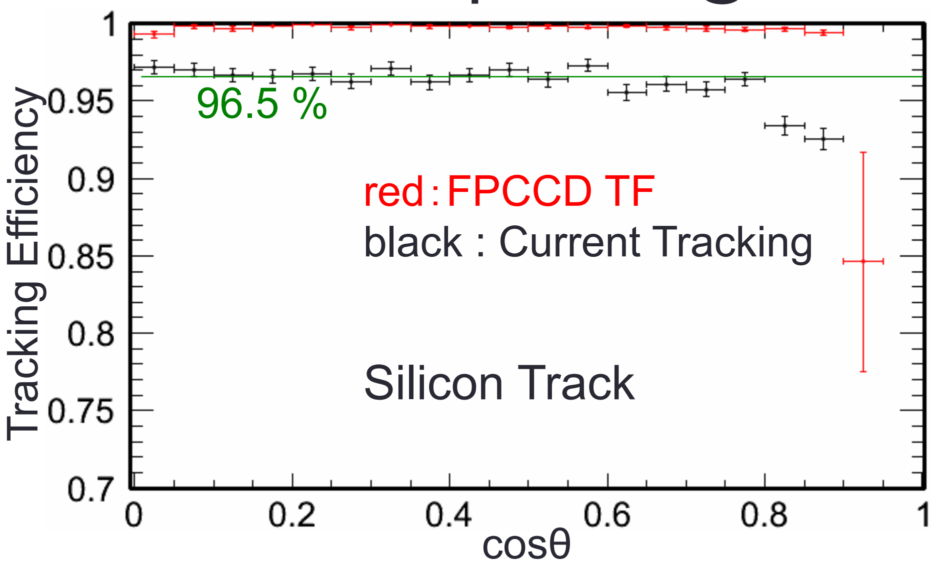
$$\eta \equiv \frac{\text{\# of tracks with VXD hits >= 5 \&\& track purity > 75\%}}{\text{\# of MCParticles creating VXD sim-hits >= 6 \&\& SIT sim-hits >= 4}}$$

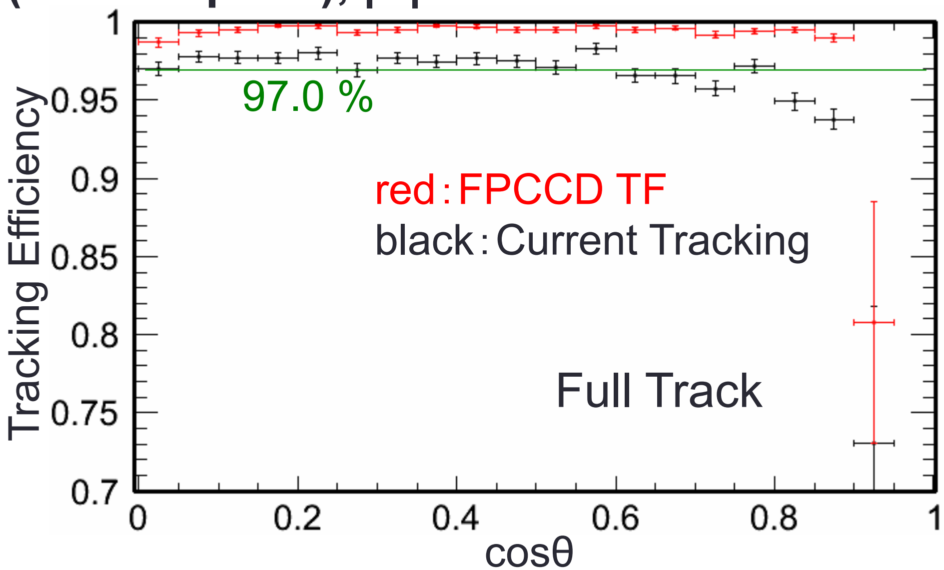
Note: SIT coverage
cosθ < 0.9

track purity:

$$\frac{\text{(\# of the MCP's hits of track)}}{\text{(\# of all hits of track)}}$$

Sample: ttbar @ 350 GeV (without pairs), |P| > 1 GeV/c





Efficiency : ~ 99 % @ cosθ < 0.9

FPCCD Track Finder: without / with pairs from 1 train (P_T)

Tracking Efficiency : η ≡

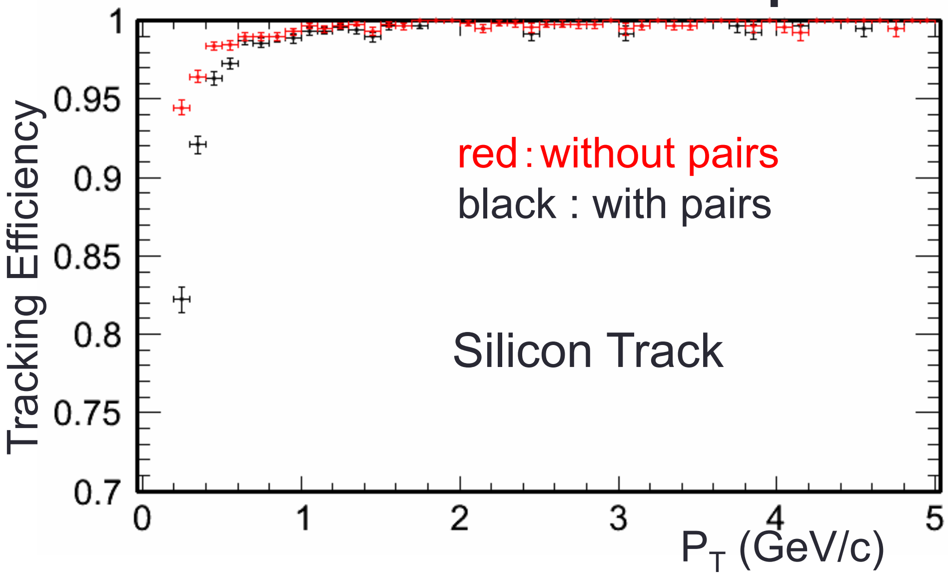
$$\eta \equiv \frac{\text{\# of tracks with VXD hits >= 5 \&\& track purity > 75\%}}{\text{\# of MCParticles creating VXD sim-hits >= 6 \&\& SIT sim-hits >= 4}}$$

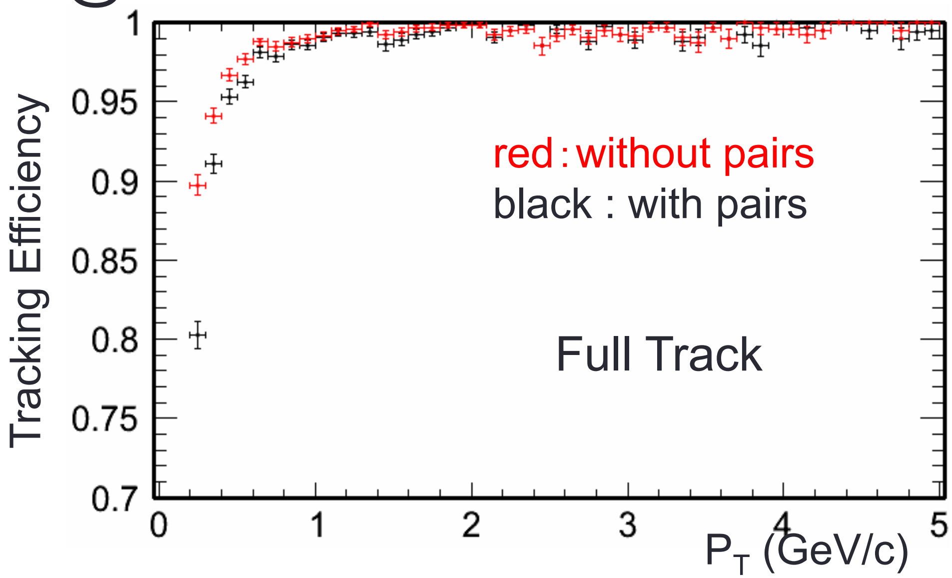
Note: P_{Tmin} to reach TPC
R_{in} : 0.4 GeV/c
R_{out} : 1.8 GeV/c

track purity:

$$\frac{\text{(\# of the MCP's hits of track)}}{\text{(\# of all hits of track)}}$$

Sample: ttbar @ 350 GeV





Slightly degraded with pairs @ P_T < 0.6 GeV/c

FPCCD Track Finder: without / with pairs from 1 train (cosθ)

Tracking Efficiency : η ≡

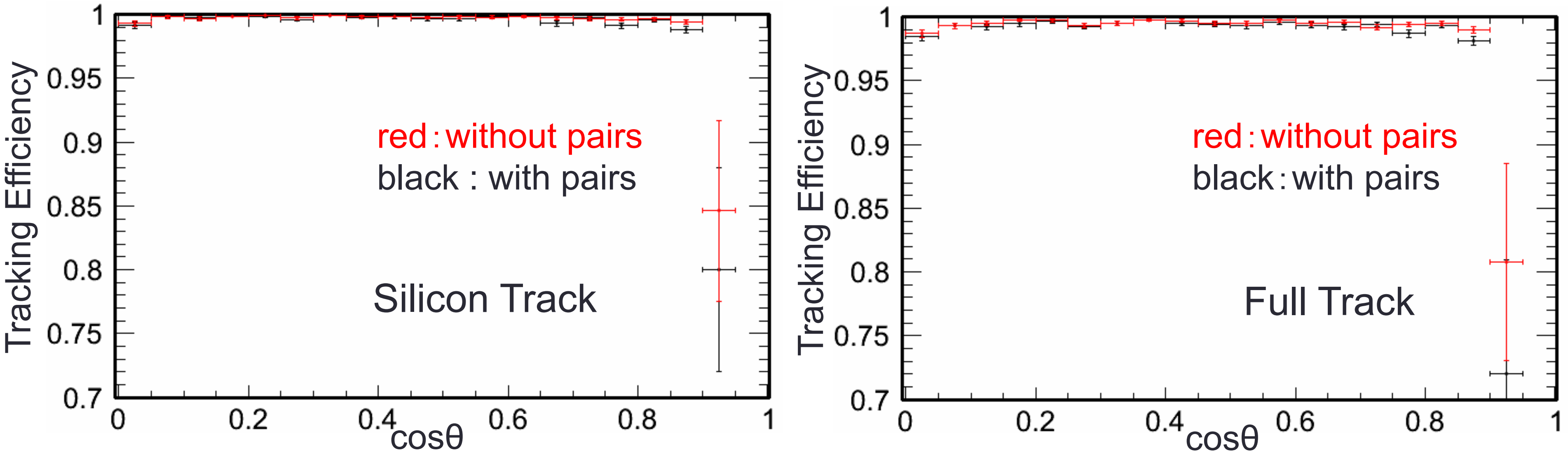
$$\eta \equiv \frac{\text{\# of tracks with VXD hits >= 5 \&\& track purity > 75\%}}{\text{\# of MCParticles creating VXD sim-hits >= 6 \&\& SIT sim-hits >= 4}}$$

Note: SIT coverage
cosθ < 0.9

track purity:

$$\frac{\text{(\# of the MCP's hits of track)}}{\text{(\# of all hits of track)}}$$

Sample: ttbar @ 350 GeV, |P| > 1 GeV/c



Slightly degraded with pairs

CPU time and memory usage of FPCCD Track Finder

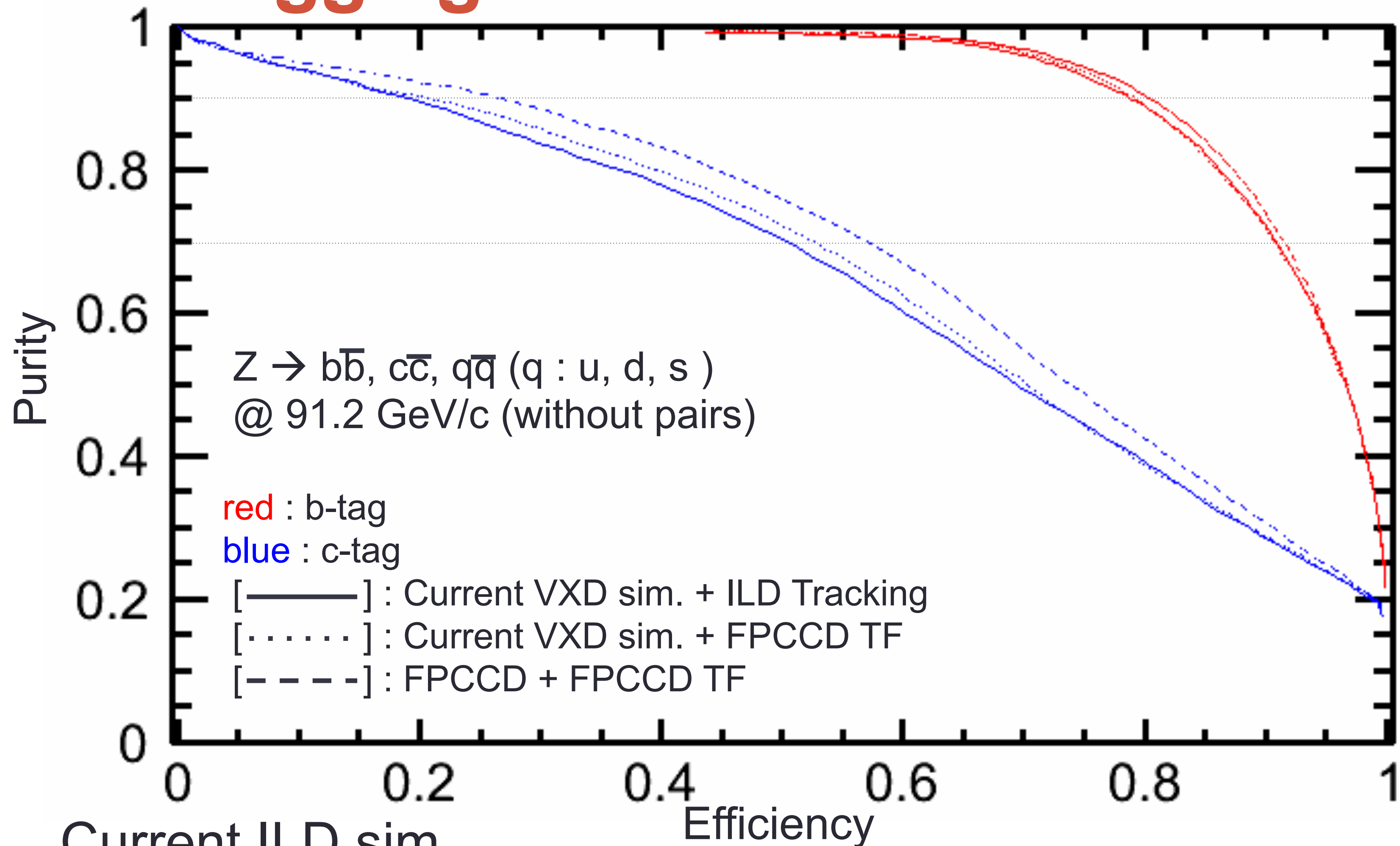
Sample: ttbar 350 GeV/c

- CPU Time
  - without pairs → almost same as ILD tracking
  - with pairs from 1 train → ~ 3 hours / evt
    - Process of track seed consumes CPU time dominantly
      Track seed : Extrapolation = 5 : 1

- Memory
  - with pairs from 1 train → ~ 3.5 GB / evt
  - note: ttbar @ 1 TeV + pairs from 1 train + current ILD Tracking + FPCCD
    → ~ 50 GB / evt
    I didn't check in the case of FPCCD Track Finder,
    but the situation would be similar

Flavor Tagging

Flavor Tagging



Current ILD sim.
+ FPCCDTF
↓
FPCCD + FPCCDTF

b-tag : efficiency 2% Up @ purity 90%

c-tag : efficiency 4% Up @ purity 70%

by using FPCCD we can improve efficiency for ZHH → b$\bar{b}$b$\bar{b}$b$\bar{b}$

Summary and Plan

◆ Summary

- FPCCD Track Finder has been developed
  - Tracking Efficiency is **~ 99 %** @ P_T > 0.6 GeV/c & cosθ < 0.9
  - The First success of tracking with pair background
    - Efficiency slightly degrades by pairs
  - FPCCD Track Finder improves flavor tagging performance
    - c-tag efficiency increases by 2.5 % @ purity 70 %
  - Using FPCCD gives us better flavor tagging performance than using current VXD in simulator

◆ Plan

- Further development of FPCCD Track Finder
- Evaluation of flavor tagging in the presence of pairs
- Check physics performance

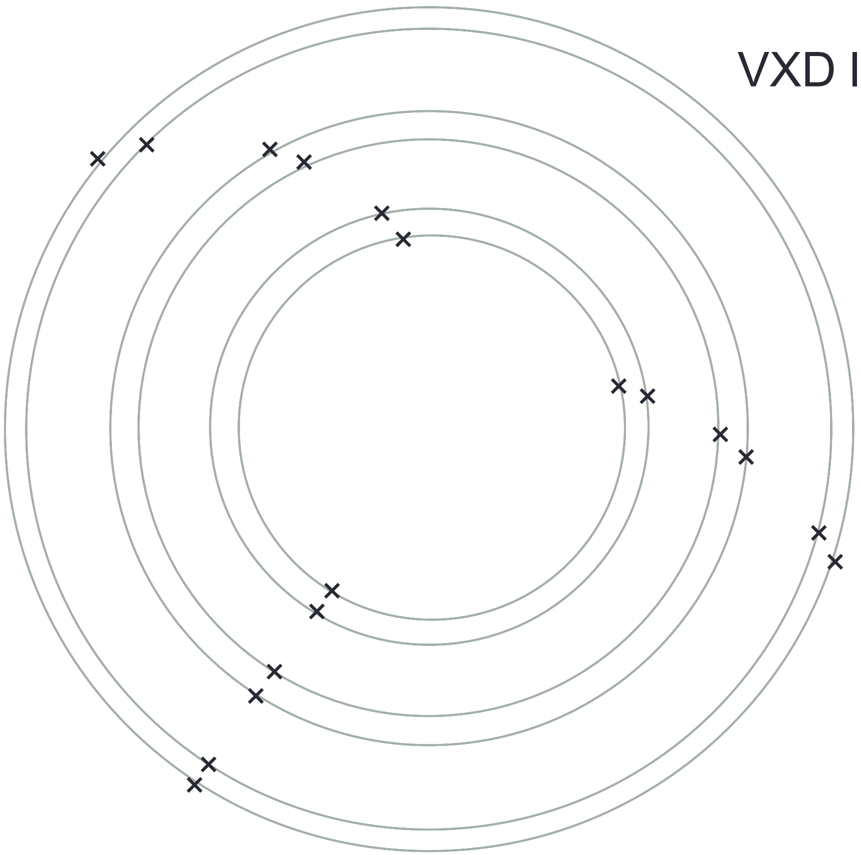
Backup

About Digitizing Hits

- In this study, digitizer for FPCCD is used (FPCCDDigitizer, FPCCDClustering)
- Pixel hits are created by the digitizer which takes into account Landau distribution, threshold, path length, noise
- Pair background hits is also digitized by the digitizer

Current ILD Silicon Tracking

For ease,
We don't consider SIT and FTD

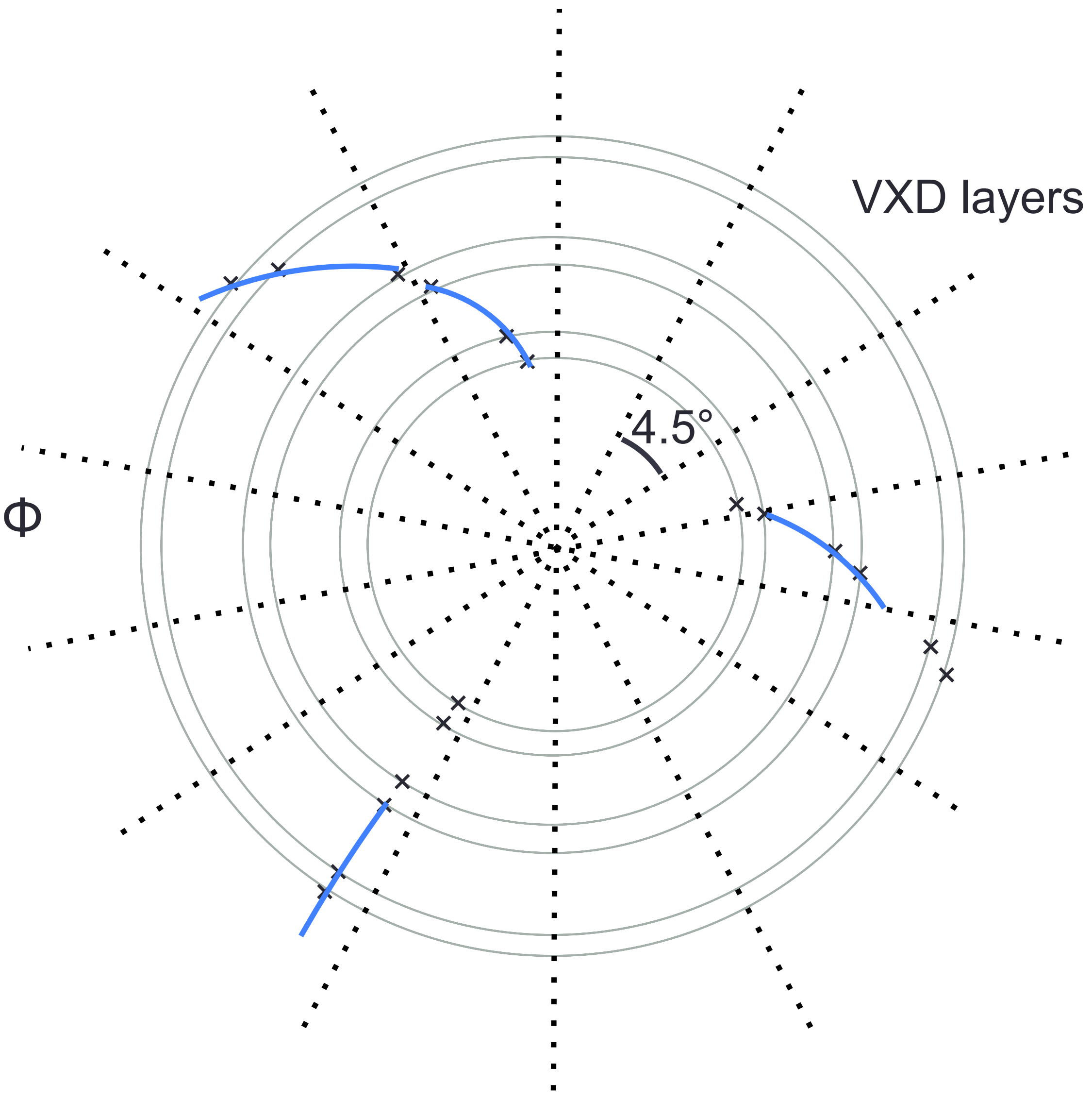


We approximate VXD shape by cylinder

DBD Silicon Tracking

Track Seed

Track seeds are generated
by combining 3 hits on each of
the 3 layers in each area
divided by 4.5° in the direction of Φ

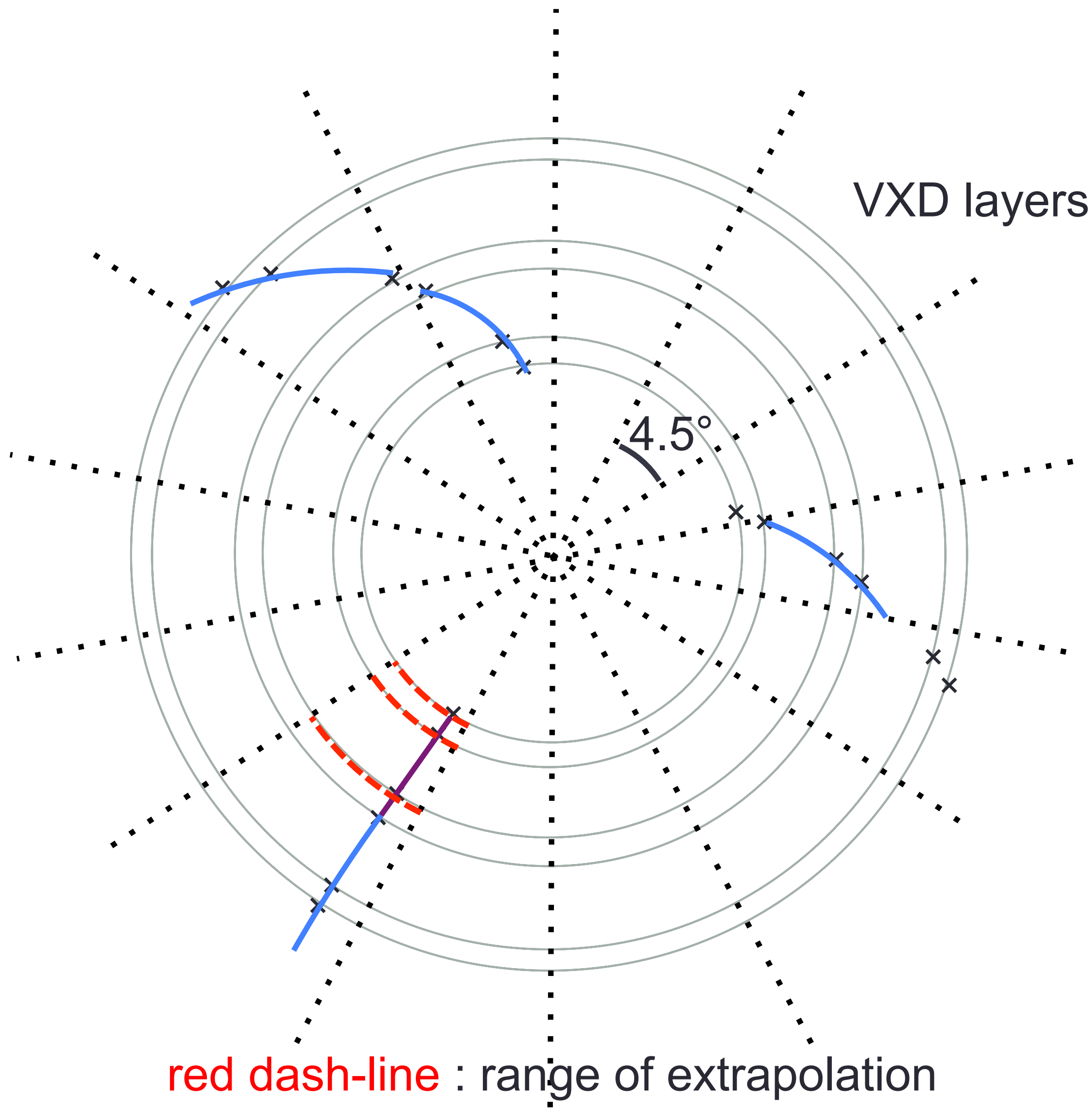


DBD Silicon Tracking

Track Seed

↓

Extrapolation

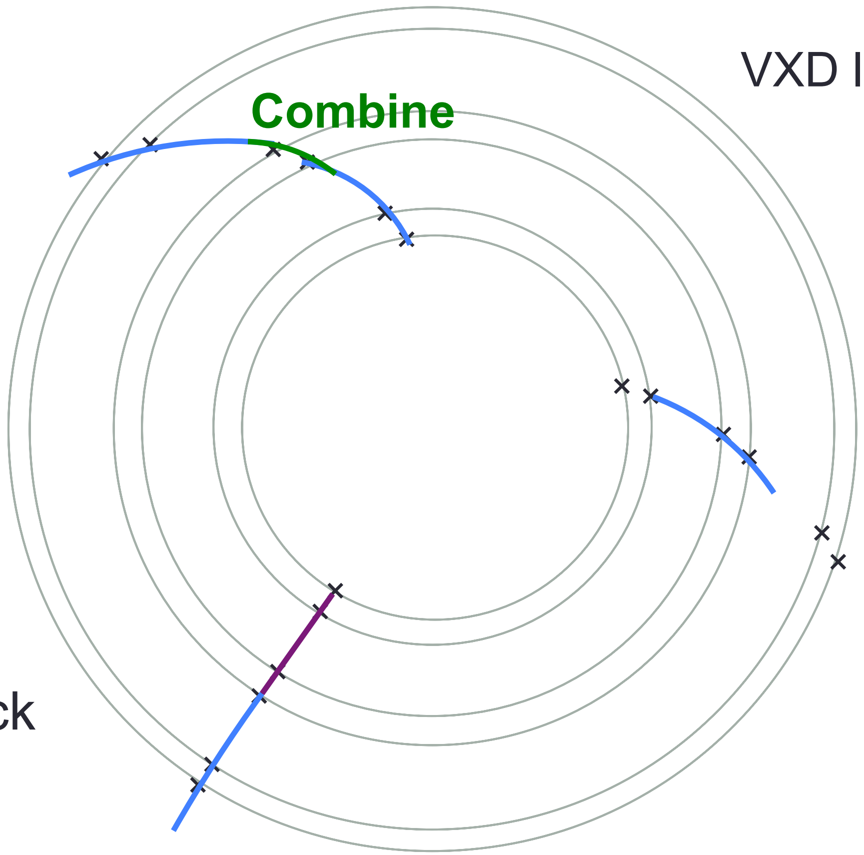


Area for extrapolation :
divided by 4.5°
in the direction of Φ

Fitter :
Simple Helix Fit

DBD Silicon Tracking

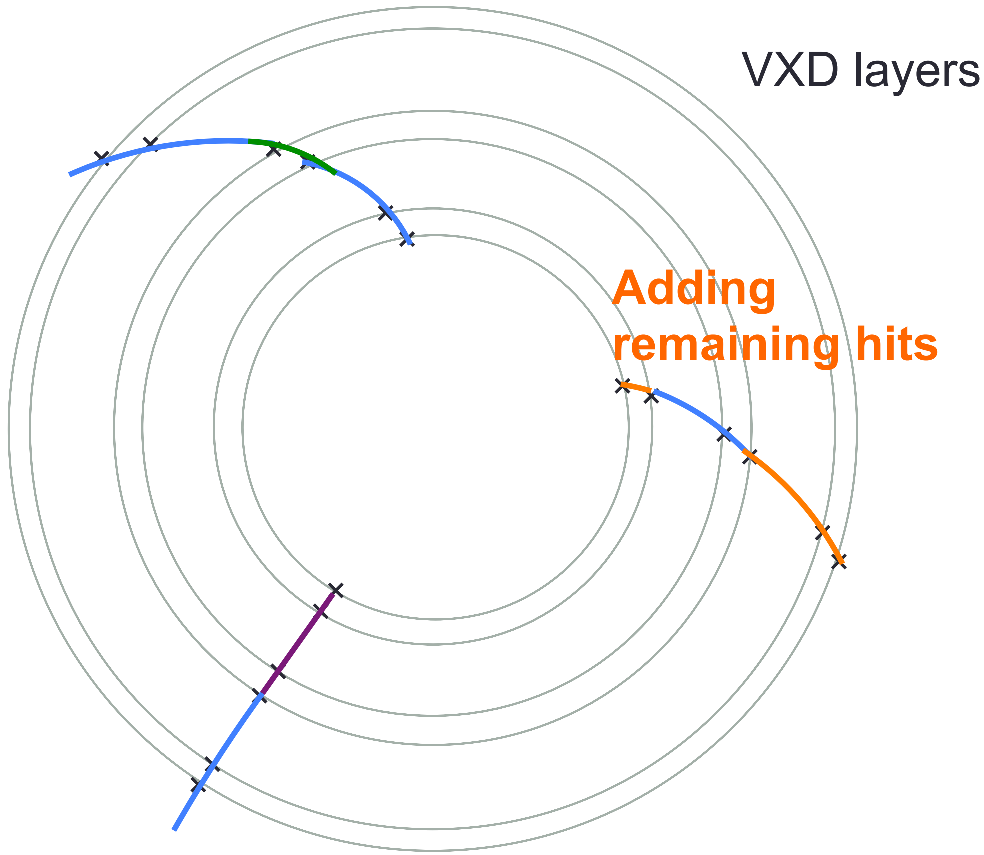
Track Seed
↓
Extrapolation
↓
Combining tracks



If possible,
we combine a track and another track

DBD Silicon Tracking

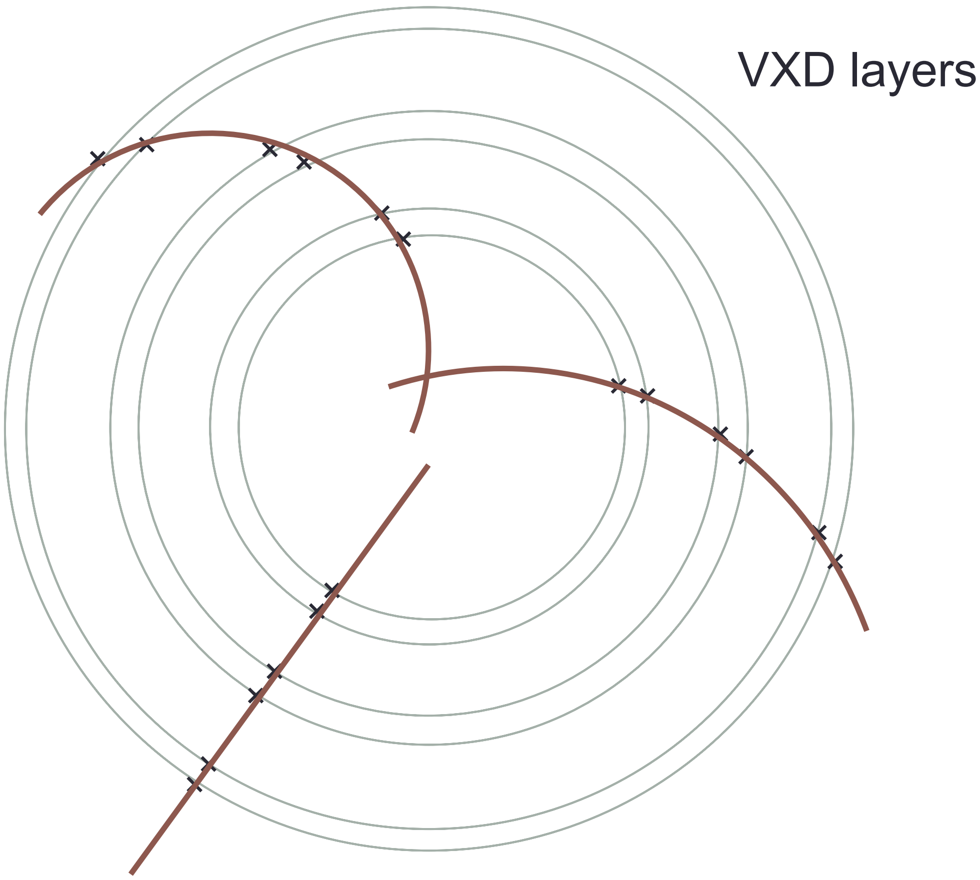
Track Seed
↓
Extrapolation
↓
Combining tracks
↓
Adding hits



If possible,
we add remaining hits to tracks

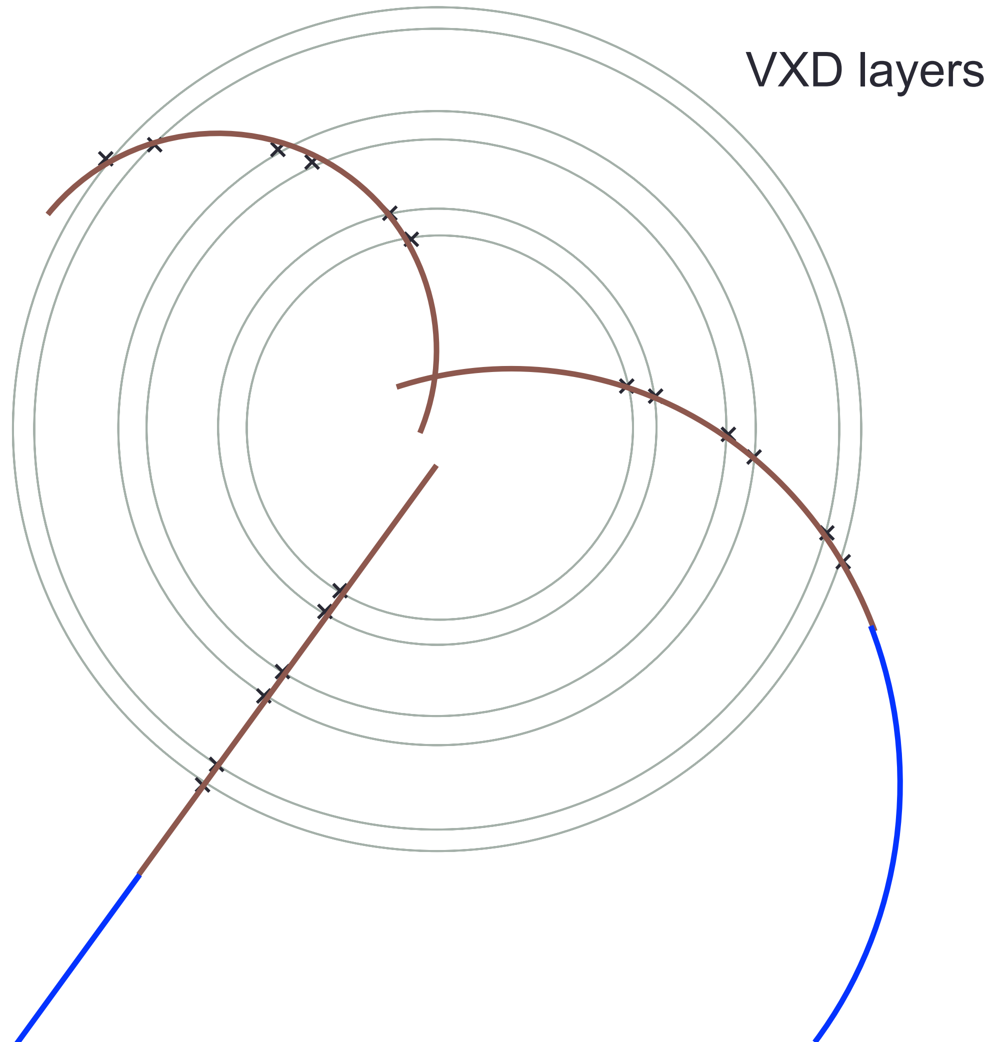
DBD Silicon Tracking

1. Track Seed
2. Extrapolation
3. Combining tracks
4. Adding hits
5. Refit with Kalman Filter

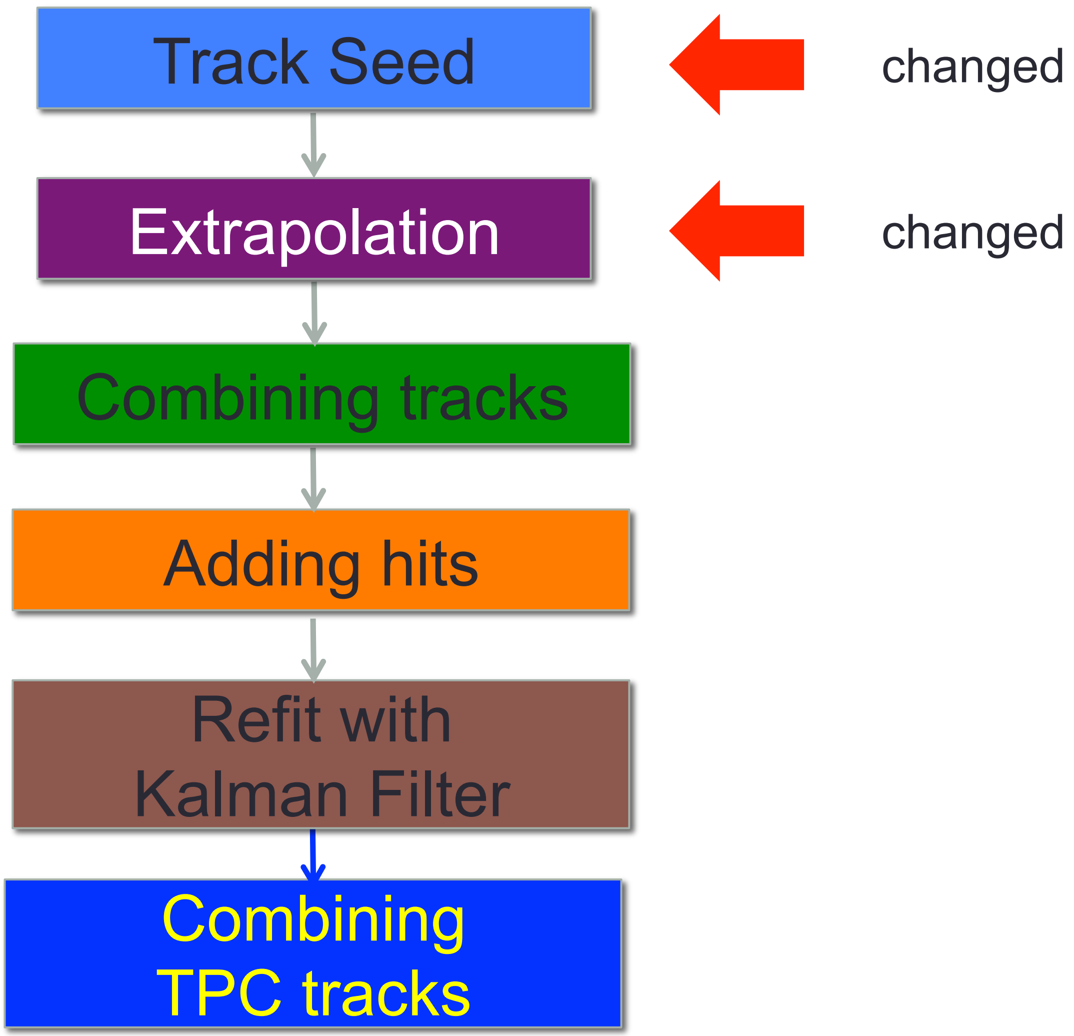


Full Track

If possible, we combine TPC tracks with silicon tracks, and then refit tracks with Kalman Filter



Differences between DBD ver. and FPCCD ver.



FPCCD Track Finder

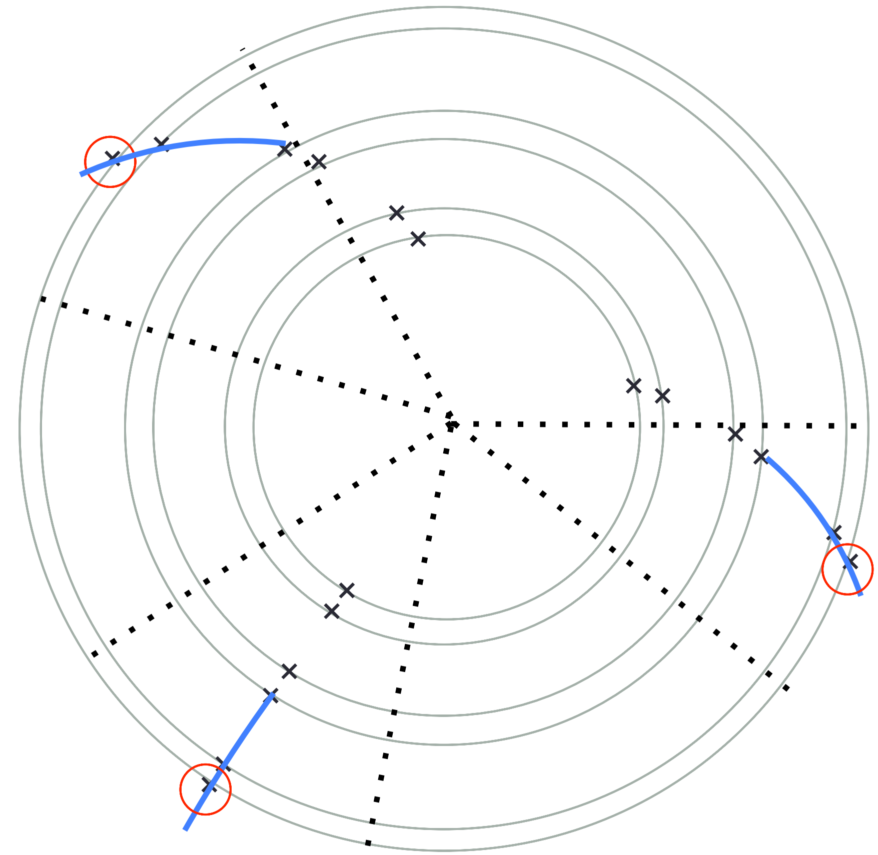
Track Seed

↓

Extrapolation

↓

the same



- 3 layers for search

(SIT: 8, 6 VXD: 5~0)

DBD version:

8 6 5	8 6 4	8 6 3	8 6 2
8 5 3	8 5 2	8 4 3	8 4 2
6 5 3	6 5 2	6 4 3	6 4 2
6 3 1	6 3 0	6 2 1	6 2 0
5 3 1	5 3 0	5 2 1	5 2 0
4 3 1	4 3 0	4 2 1	4 2 0

FPCCD version:

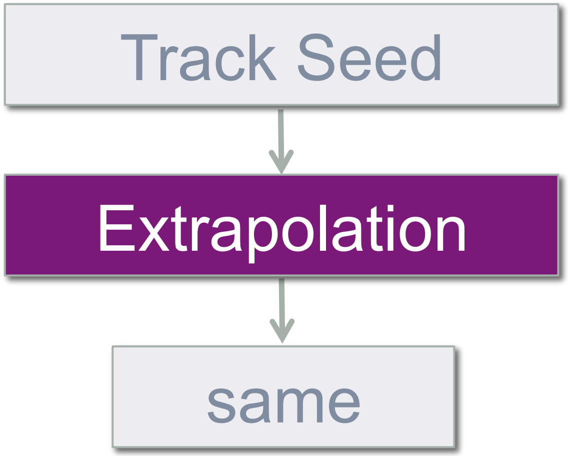
8 6 5	8 6 4	8 5 4	6 5 4
5 4 3			

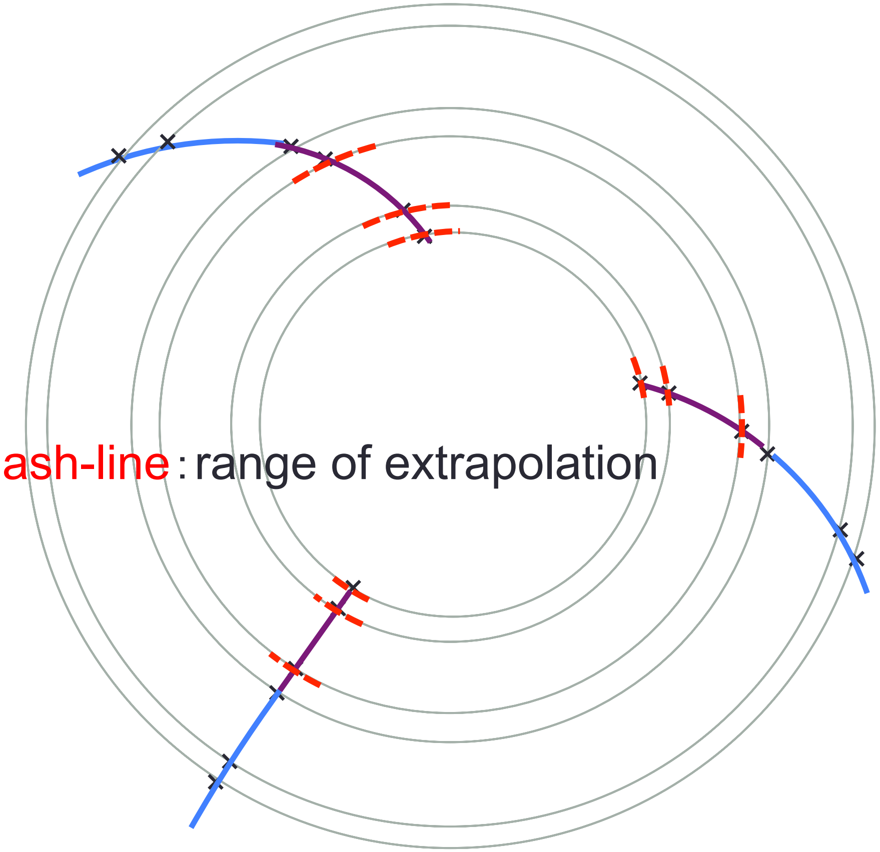
(FPCCD version)
We calculate Φ width enough to generate track seeds with P_T > 0.18 GeV/c on the basis of a hit on the outer layer

↓

We generate a track seed from 3 hits within the calculated Φ width on each of the 3 layers

FPCCD Track Finder





(FPCCD version)

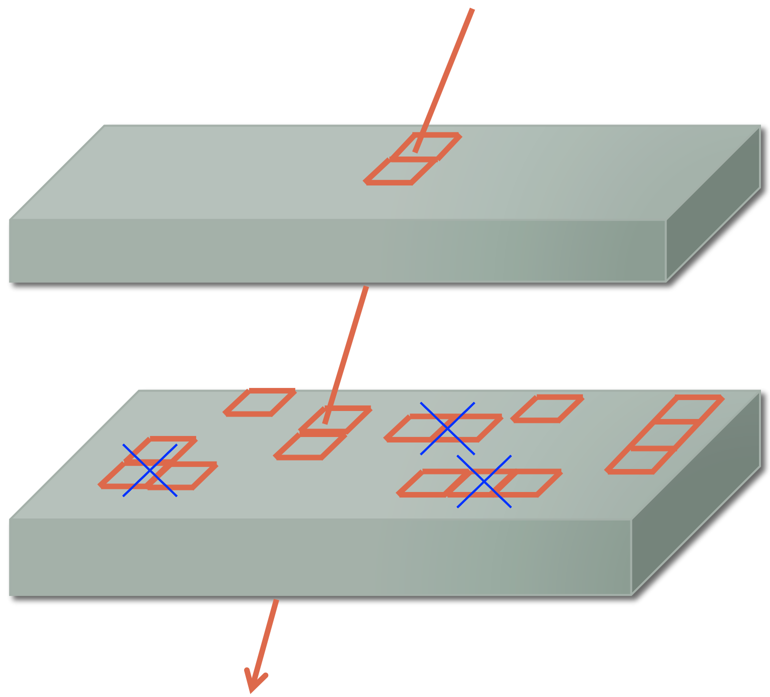
Fitter : **Kalman Filter**

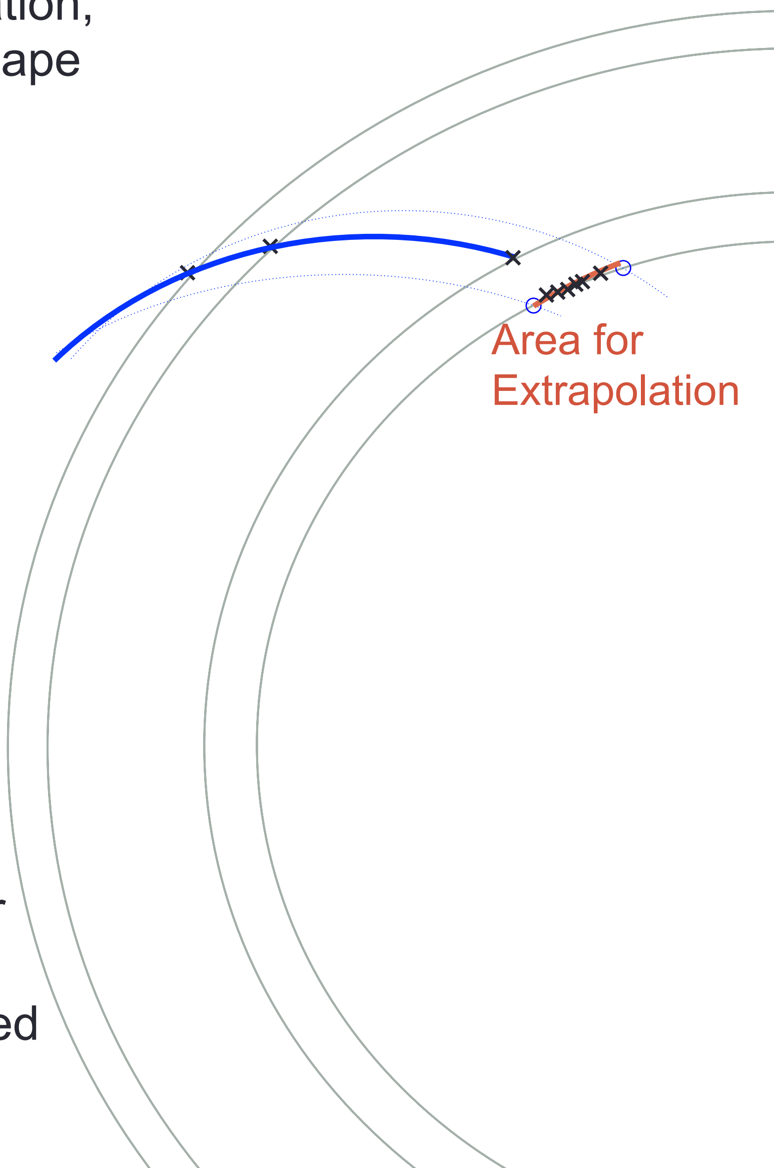
Φ width for extrapolation : determined from track parameters from the fitter

Algorithm for matching hit clusters : **optionally available : purity ↑**

Algorithm for matching hit clusters

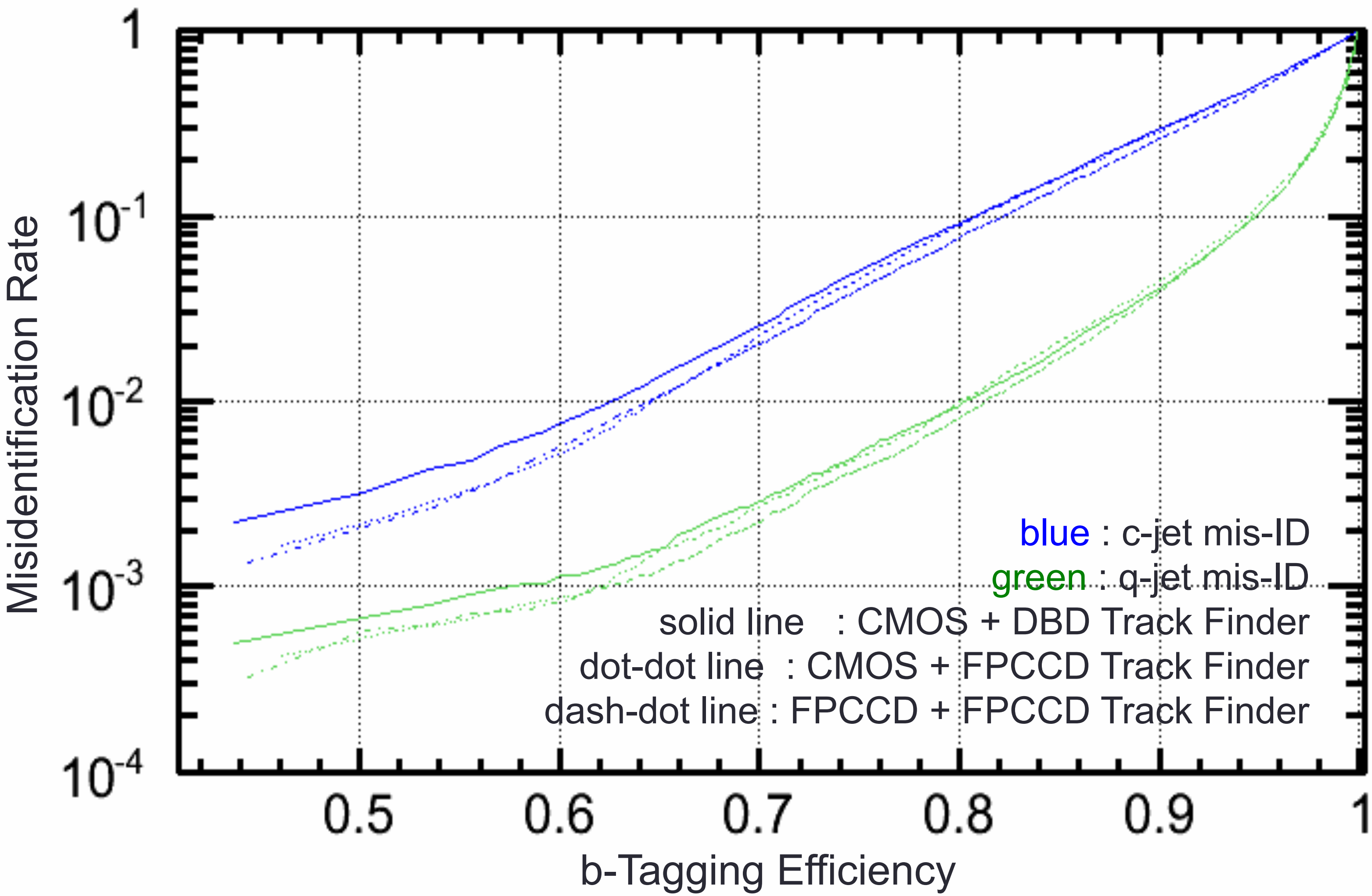
If there are many cluster hits in an area for extrapolation, we can reduce misextrapolations by using cluster shape





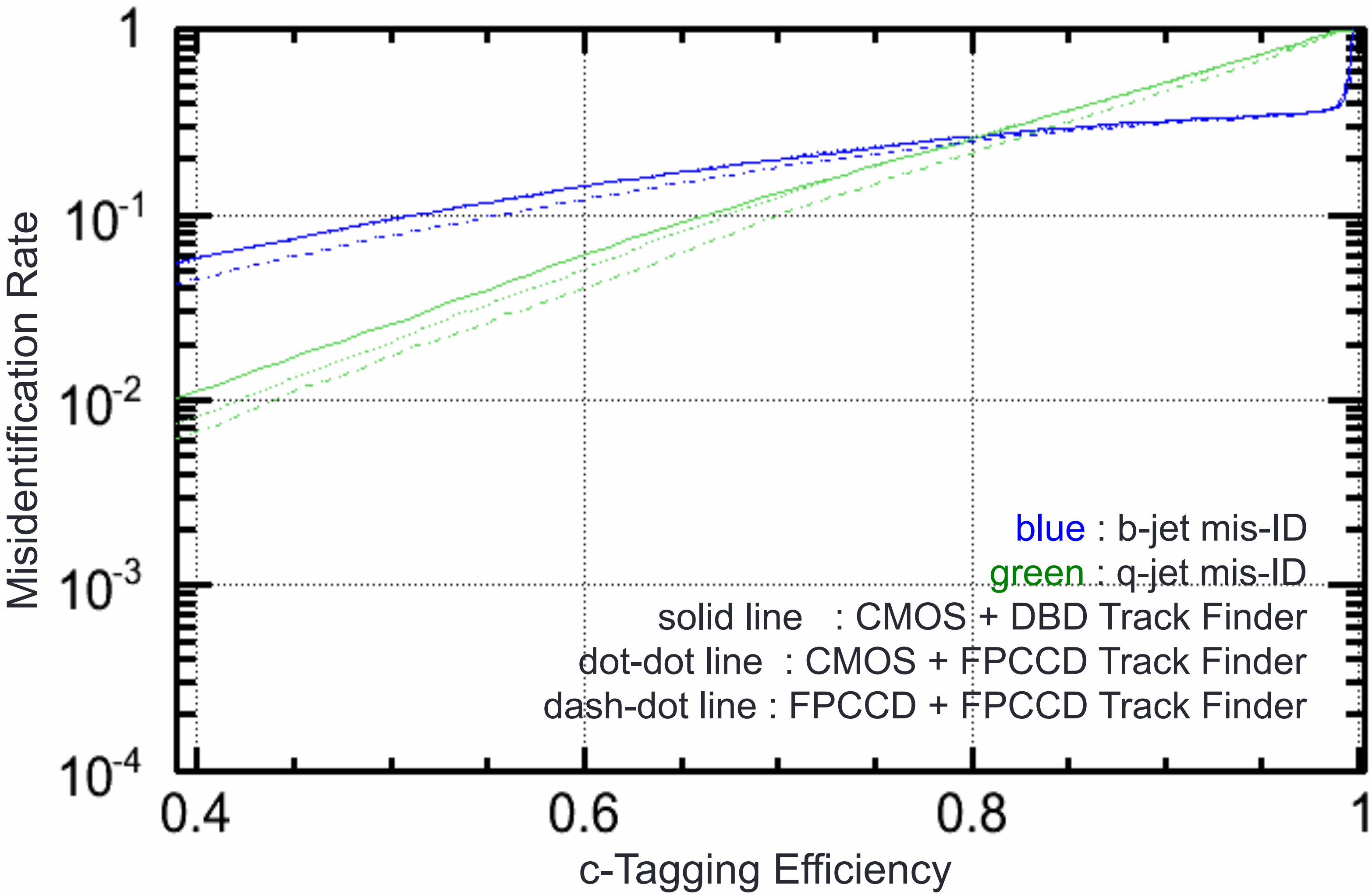
1. We calculate inner dot between candidate cluster and a cluster on the neighbor layer
2. If the dot is < 0.4, the candidate cluster is excluded from the candidates

Flavor Tagging (b-tag Misidentification Rate)



b-tag misidentification rate : slightly improved

Flavor Tagging (c-tag Misidentification Rate)



c-tag misidentification rate : improved