

国際リニアコライダーのための高精細CCDを用いた崩壊点検出器のシミュレーション研究

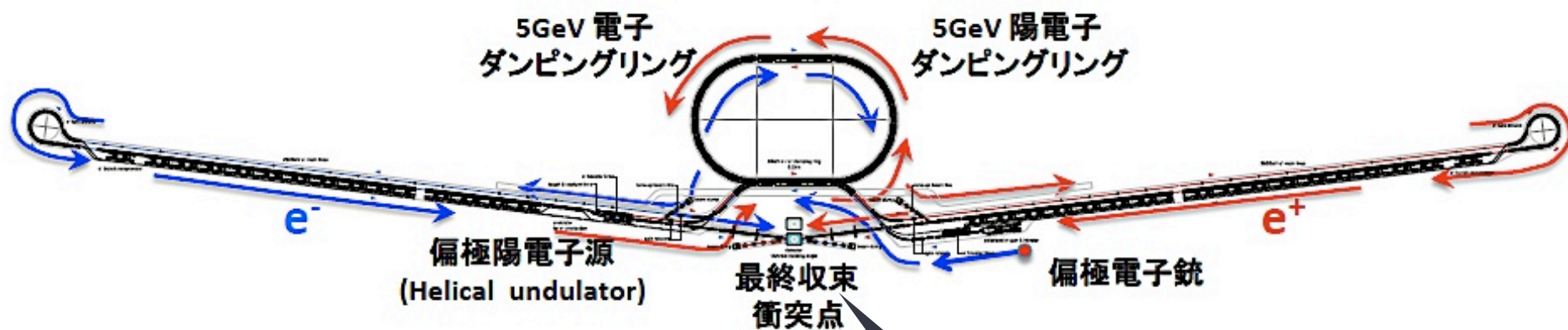
森 達哉

目次

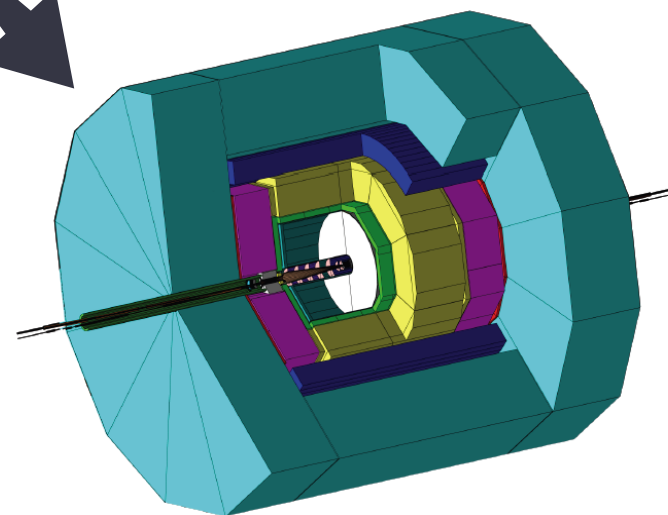
1. 国際リニアコライダー(ILC) の紹介
2. FPCCD崩壊点検出器の紹介
3. 本研究について
 1. 占有率の評価とペアBGクラスター除外アルゴリズム
 2. トラッキングの性能評価とFPCCDTrackFinder
 3. フレーバータグの性能評価
4. まとめ

国際リニアコライダー(ILC)の紹介

国際リニアコライダー(ILC)の紹介



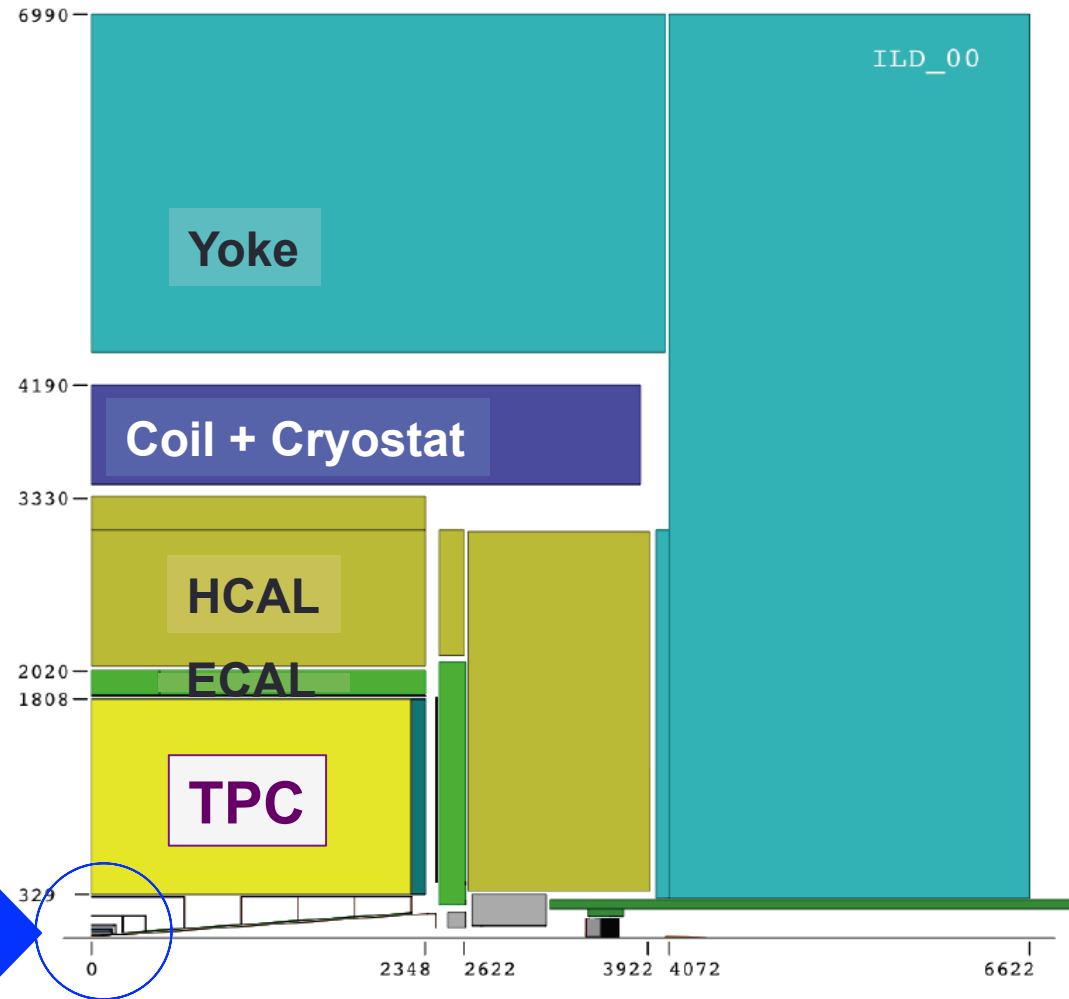
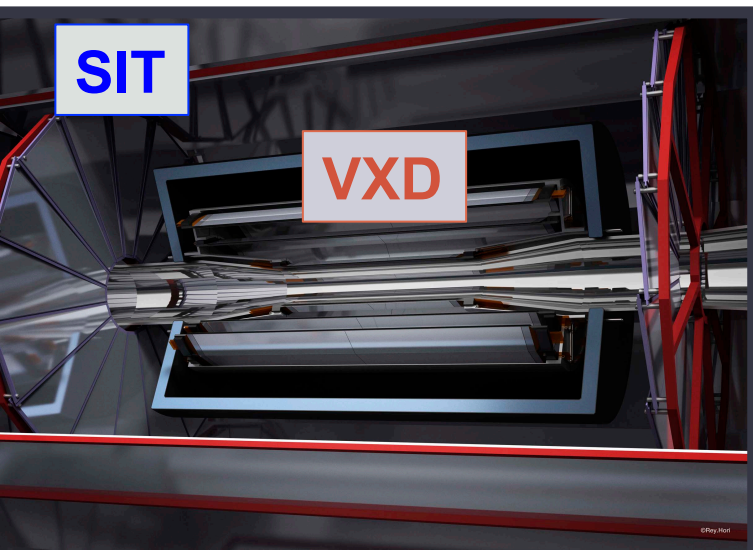
- e^+e^- 衝突型線形加速器
- $E_{CM} = 250 \sim 500 \text{ GeV}$
(アップグレード: 1 TeV)
- 全長 31 km :
(1 TeVアップグレード: 50 km)
- $\mathcal{L} = 2 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1} @ 1 \text{ TeV}$
- 202X年に運転開始が期待される



ILC の検出器ILDの紹介

トラッカーは内側から

1. 崩壊点検出器(VXD)
(シリコンピクセル型検出器)
2. SIT
(シリコンストリップ型検出器)
3. TPC
(ガスチェンバー型検出器)

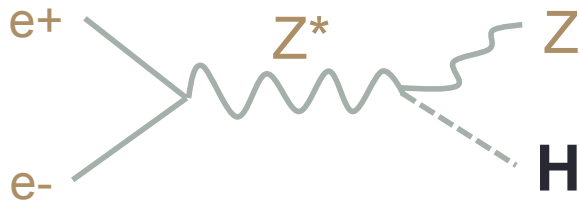


3.5 T の一様な磁場が
ビームに平行な向きに印加

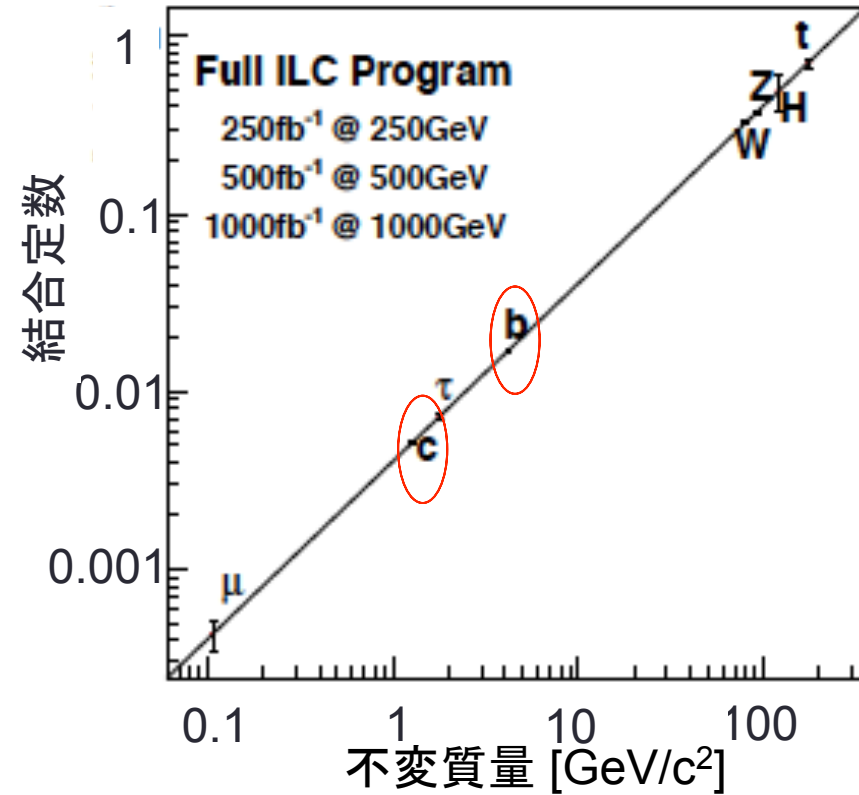
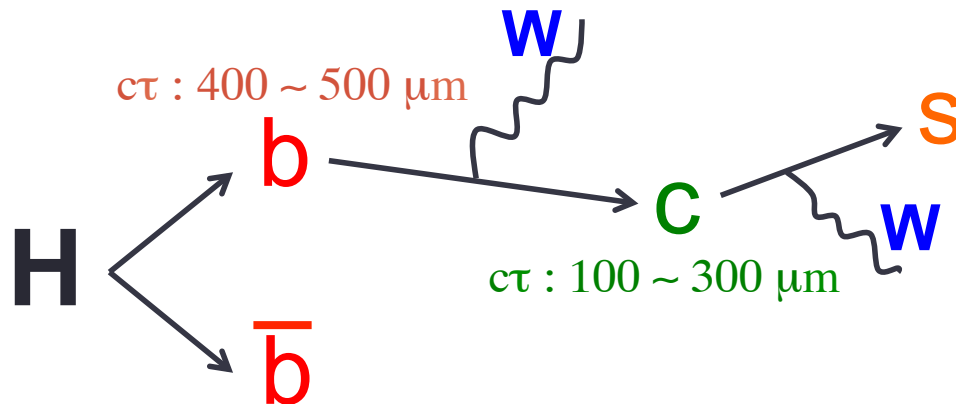
ヒッグスと崩壊点検出器

ILCの主な目的の一つ：

「c、bクォーク」と「Higgs」との
結合定数の精密測定



$H \rightarrow b\bar{b}, c\bar{c}, gg$ を正確に区別したい



高性能な崩壊点検出器が必要

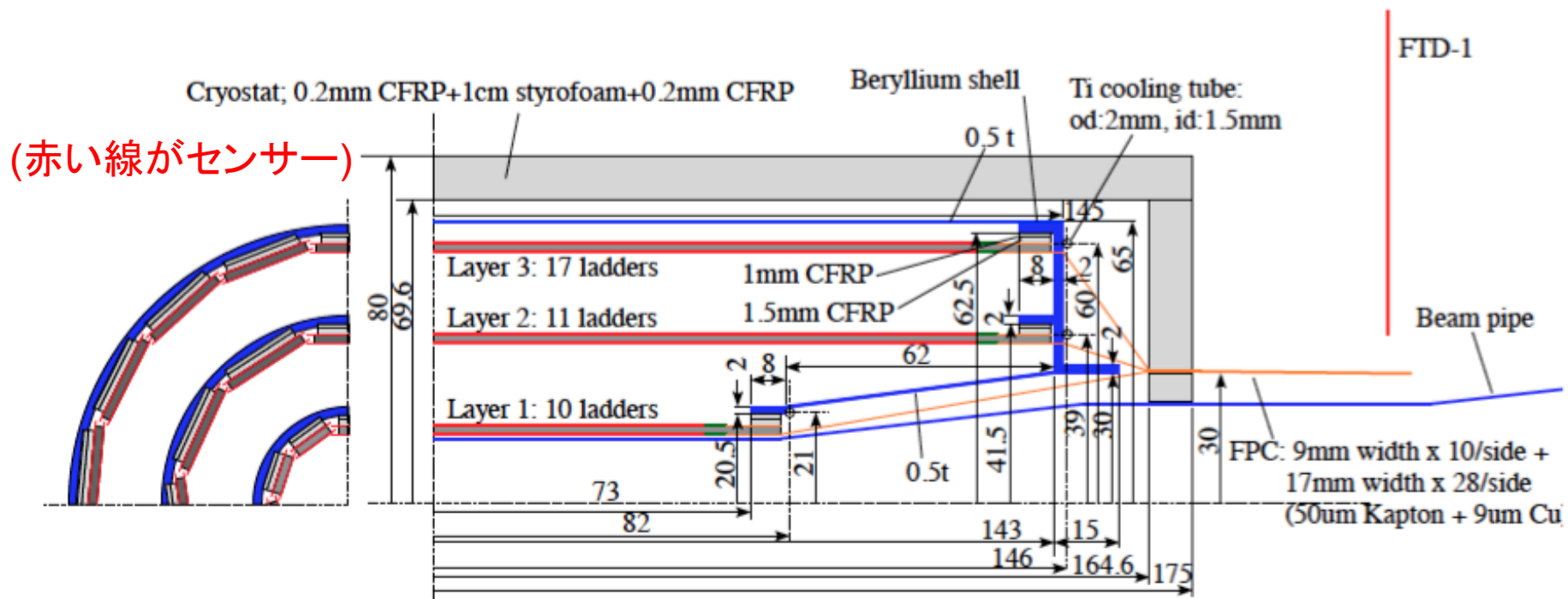
FPCCD崩壊点検出器の紹介

FPCCD崩壊点検出器1

FPCCD (Fine Pixel CCD) の特徴

- 小さなピクセル: $5\text{-}10\ \mu\text{m}$ (右表)
- 厚み: $50\ \mu\text{m}$ (その内 $15\ \mu\text{m}$ は有感層)
- ピクセル数: $\sim 0.4 \times 10^9$

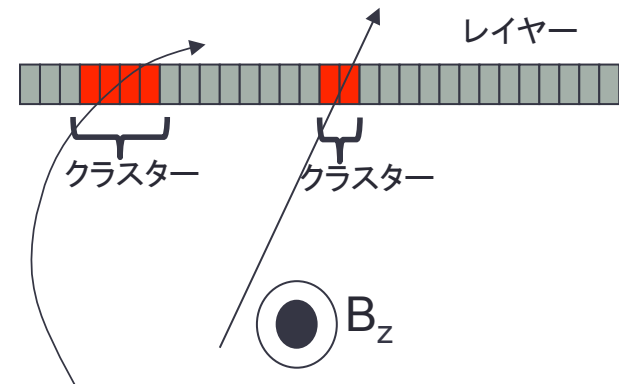
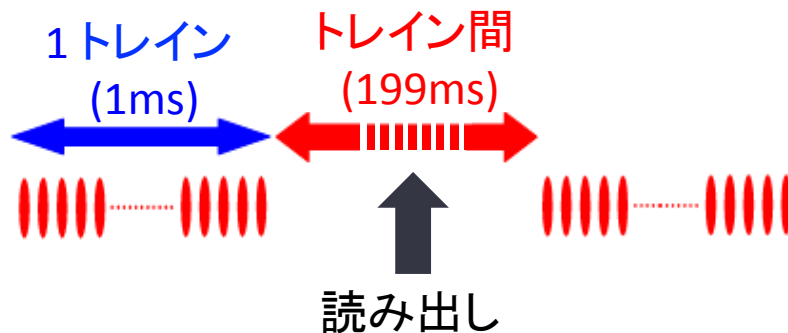
レイヤー	衝突点からの距離(mm)	ピクセルサイズ(μm^2)
0	16	5×5
1	18	5×5
2	37	10×10
3	39	10×10
4	58	10×10
5	60	10×10



FPCCD崩壊点検出器2

FPCCD (Fine Pixel CCD) の特徴

- ピクセルヒットの塊(クラスター)ができる:
 - ✓ トラック外挿に役立つ
 - ✓ 位置分解能が向上する
 - ✓ BGヒットとシグナルヒットを区別できる
- 読み出し: 1312バンチ(1トレイン)毎



メリット:

ビーム由来の高周波ノイズは無視できる

デメリット:

ヒット点が多くなるので
トラッキングが難しい

本研究について

本研究について

◆目的

- ILC の物理パフォーマンス(特にHiggs物理)の向上のため、
トラッキング効率の改善、フレーバータグの改善を目指す

◆手法

- FPCCD 崩壊点検出器の性能をシミュレーションにより評価する
- 新トラッキングアルゴリズムを開発してFPCCDの性能を上げる

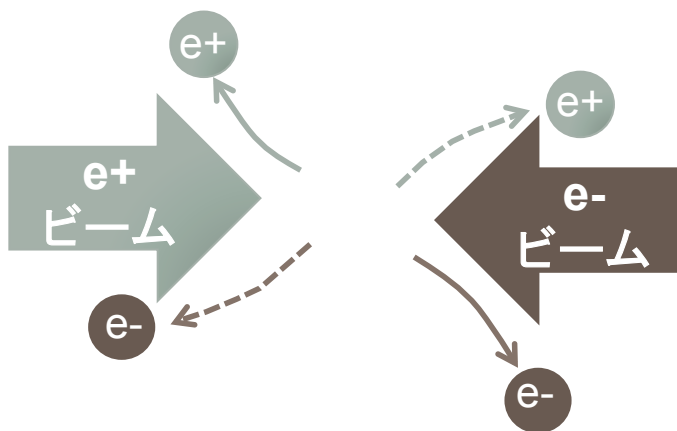
◆本研究でしたこと

- ペアBGの占有率の評価
- インパクトパラメータ分解能の評価
- トラッキング効率の評価
→FPCCDTrackFinderの開発
- フレーバータグ性能の評価

ペアBGの占有率の評価

ピクセル占有率の評価

- 崩壊点検出器における主要なBG : e^+e^- ペアBG
 - e^+e^- ビームから放射される γ が対生成して e^+e^- のペアが大量発生
 - 低い横運動量を持つ



評価方法:

ペアBGにより生成されるピクセルヒット数から各レイヤー毎の占有率を計算

サンプル: ペアBGイベント @ 250, 350, 500, 1000 GeV

サンプル数: 1 トレイン

(ただし1000 GeV は1000バンチ衝突(以下BX)のみ。これを1トレインに換算)

評価結果

E_{CM} (GeV)	最内層の占有率 (%)
250	0.56
350	0.70
500	1.24
1000	12.75

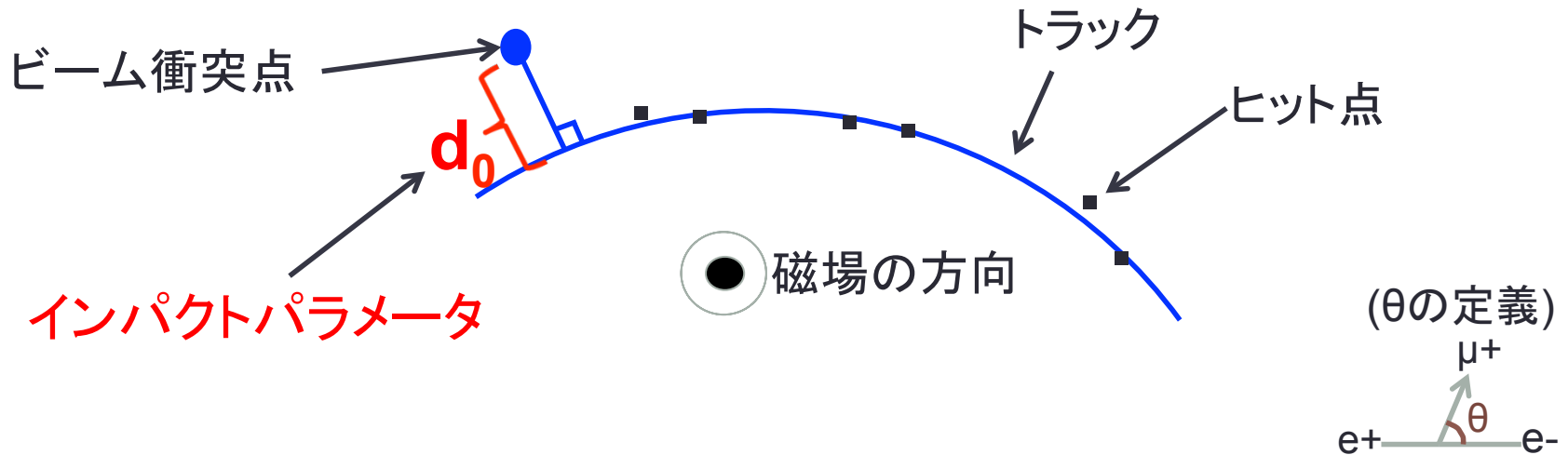
1 TeV のビームランでは問題有り

<解決策>

- レイヤーを衝突点から遠ざける
- ピクセルを小さくする
- クラスター形状からペアBGクラスターを除外する
→ (時間の都合によりカット)

インパクトパラメータ分解能の評価

インパクトパラメータ分解能とは



精度の高い崩壊点を組むためには
優秀なインパクトパラメータ分解能が必要

<要求される分解能>

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

検出器固有の分解能

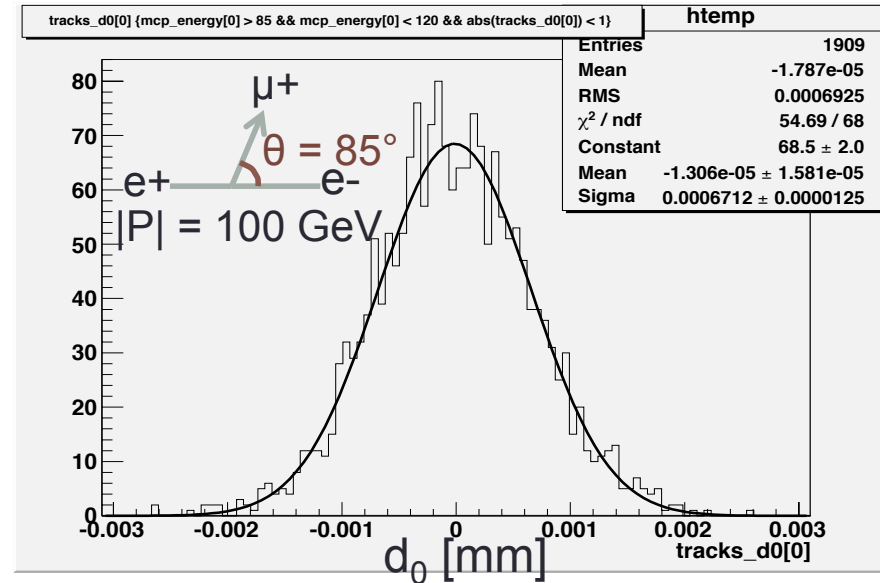
多重クーロン散乱による
分解能の悪化を考慮

セットアップ

評価方法:

1. single μ^+ イベントをトラッキング
2. トラックのインパクトパラメータをガウシアンでフィット
3. フィット結果の σ をインパクトパラメータ分解能とする

- FPCCDの比較対象としてCMOS(現行のVXDシミュレータ)も同様に評価

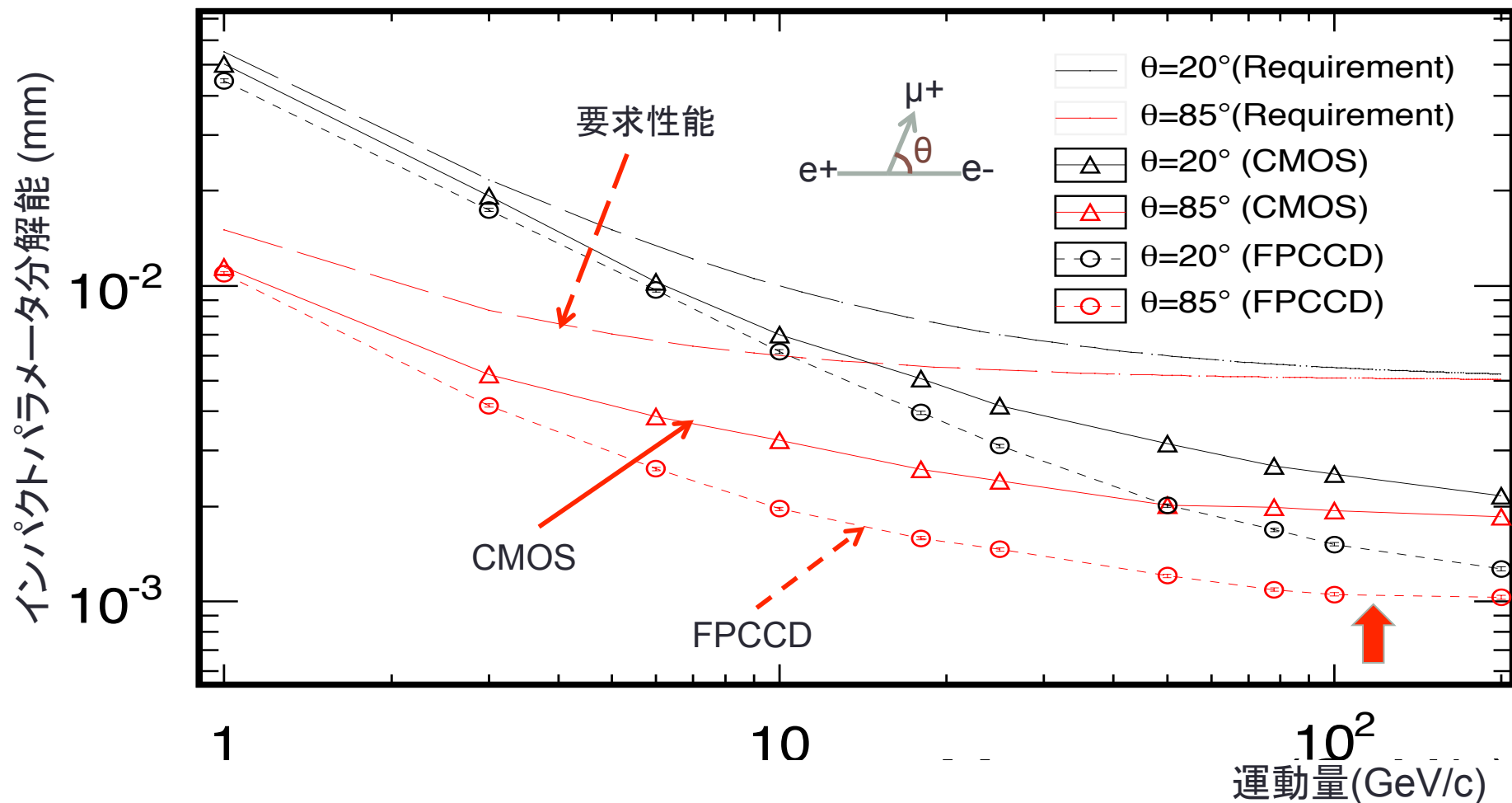


CMOS(現行のVXDシミュレータ)

FPCCD

レイヤー	位置分解能 (μm)	一度の読み出しに寄与するBX数	位置分解能 (μm)	一度の読み出しに寄与するBX数
0	2.8	90	1.4	1312
1	6.0	18	1.4	1312
2	4.0	180	2.8	1312
3	4.0	180	2.8	1312
4	4.0	180	2.8	1312
5	4.0	180	2.8	1312

インパクトパラメータ分解能の評価結果



要求性能を満たし、
高運動量領域では $\sim 1 \mu\text{m}$ の優秀な分解能を持つ

トラッキング効率の評価

現行のトラッキングアルゴリズム

第一段階:

Silicon Tracking

TPC Tracking

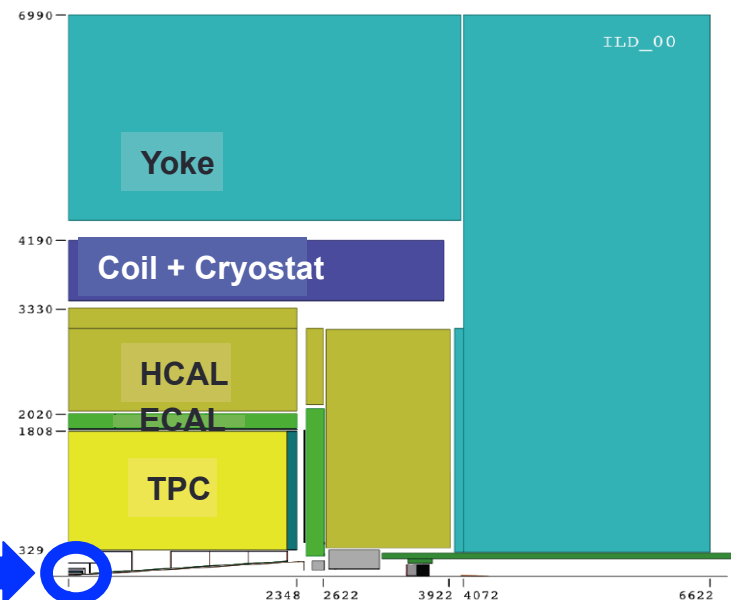
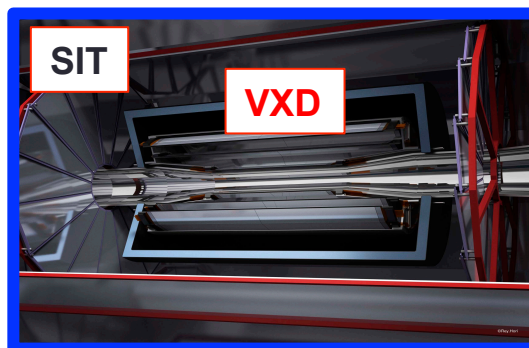
第二段階:

Silicon Track + TPC Track

Full Track

Silicon Trackingの特徴:

- VXD, SIT のヒットを使用
- 外側から内側に向かってトラックを再構成



現行のトラッキングとトラッキング効率

トラッキング効率 : $\eta \equiv$

VXDヒット ≥ 5 個 && track purity $> 75\%$ を持つトラックの数

VXDヒット ≥ 6 個 && SITヒット ≥ 4 個 作る粒子の数

参考: TPCの内径、外径に届くのに必要な P_T

内径: 0.4 GeV/c

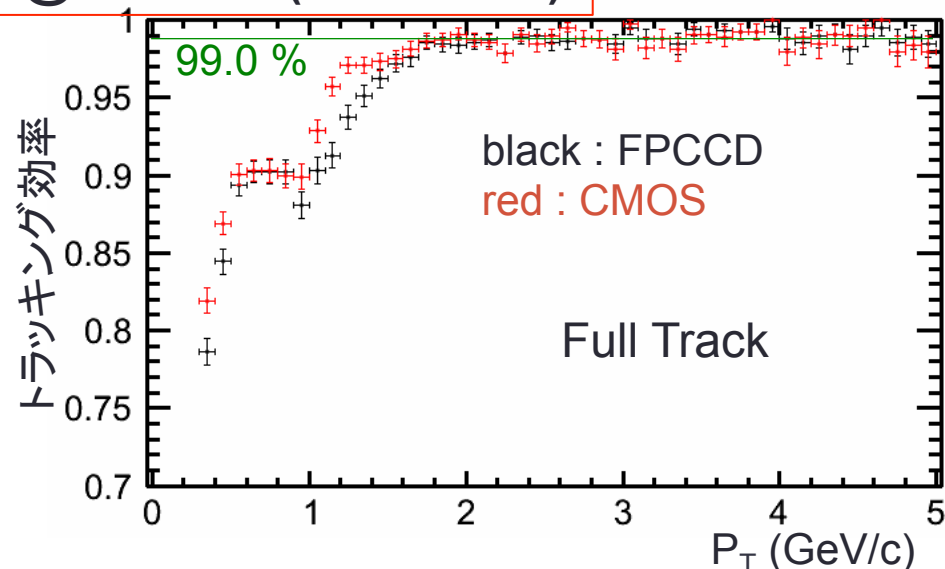
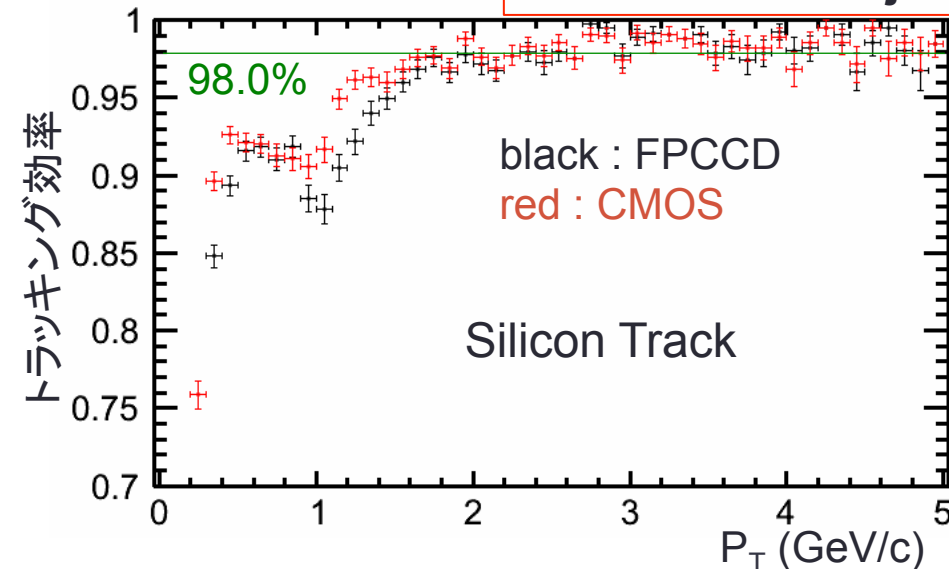
外径: 1.8 GeV/c

track purity:

(トラックの持つ正しいヒットの数)

(トラックの持つヒットの数)

サンプル: $t\bar{t} \rightarrow 6 \text{ jets @ } 350 \text{ GeV}$ (ペアBG無し)



トラッキング効率が悪化する @ $P_T < 1.7 \text{ GeV/c}$

→ Silicon Tracking を改善する必要がある

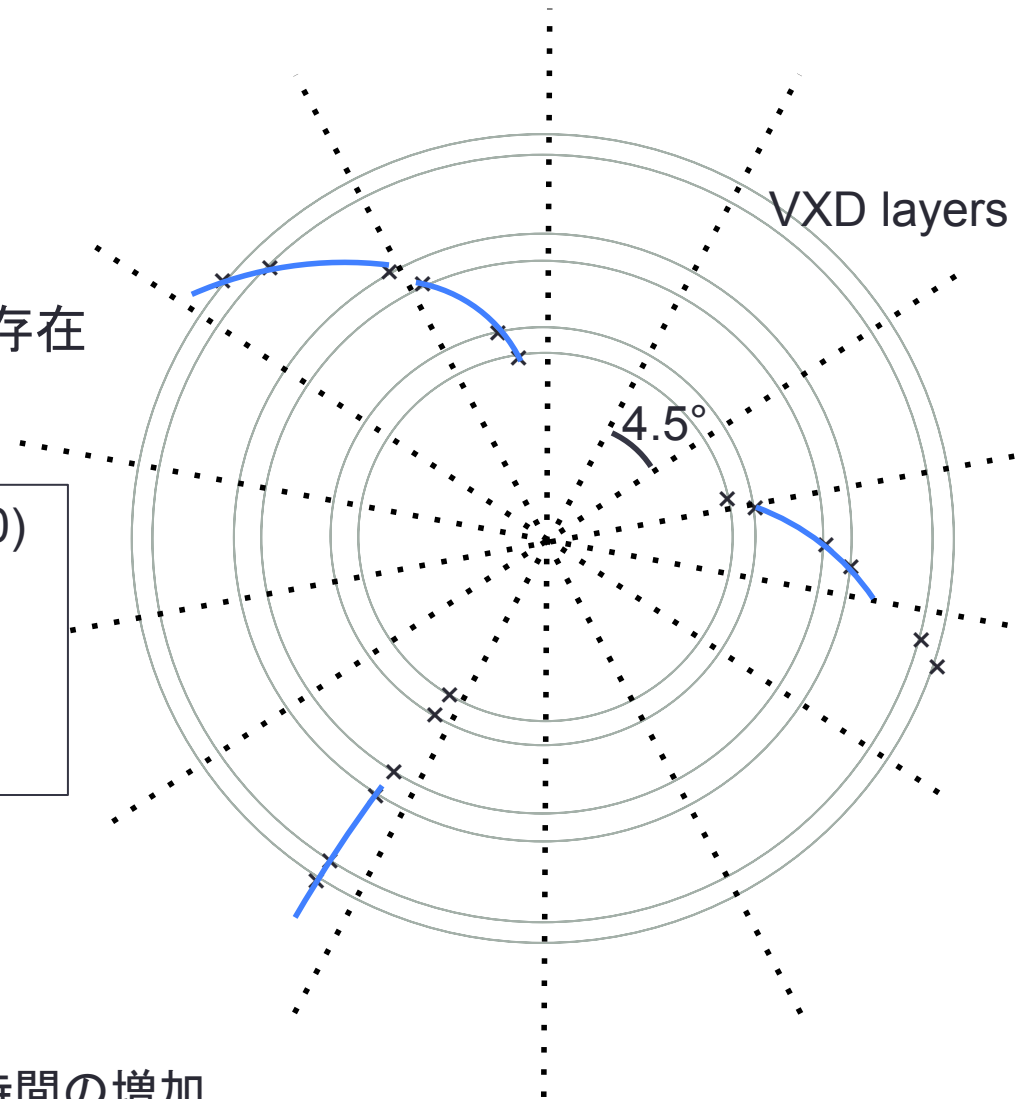
Silicon Trackingの問題1: トラックシード

トラックシード生成処理

1. ϕ 方向に80分割する(一区画 4.5°)
2. 区画内で、ある決められた
3つのレイヤー上に各々ヒット点が存在
3. フィットしてトラックシードを生成

- 3つのレイヤーの組 (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



トラックシードの問題

- 4.5° の区画が狭くて低横運動量のトラックシードが生成されない
 - 広くする? → ゴーストシードとCPU時間の増加
- 3つのレイヤーの組が多いためゴーストシードが多く、長いCPU時間が必要

FPCCDTrackFinderのトラックシード処理

解決策:

1. 区画の分割をやめ、3つのレイヤーの中で外側のレイヤーにあるヒットを基準に処理する幅を決める($P_T > 0.18 \text{ GeV}/c$ のトラックが拾えるような幅)

→低横運動量のシードが作成可

2. 3つのレイヤーの組の数を減らす

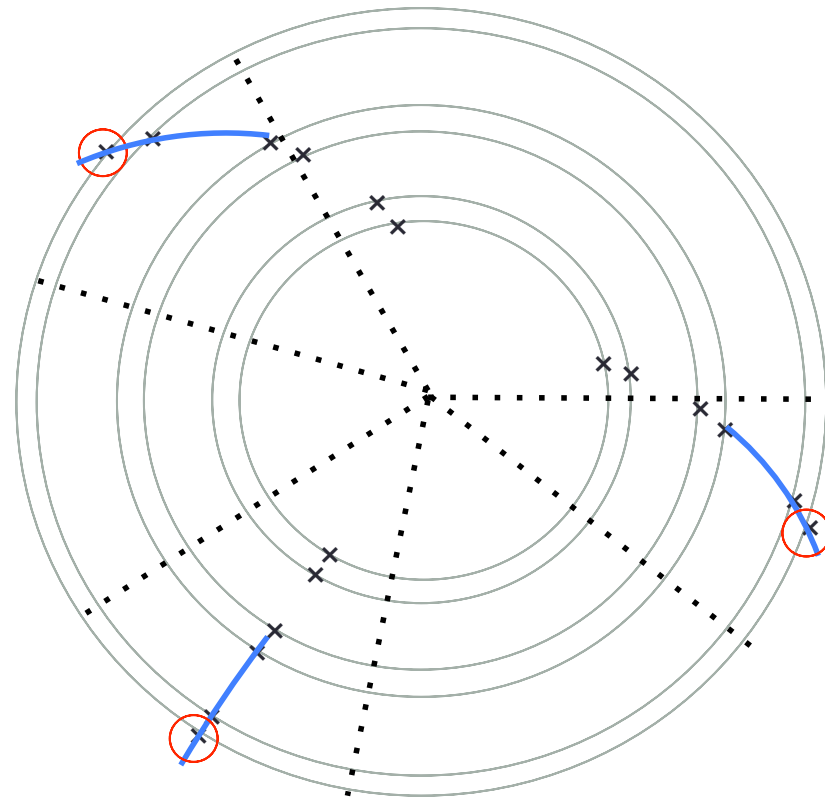
Old: (SIT: 8, 6 VXD: 5~0)

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

New:

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

→CPU時間、ゴーストシードの削減



Silicon Trackingの問題2: 外挿処理

トラックシード生成処理

外挿処理

外挿処理の範囲:

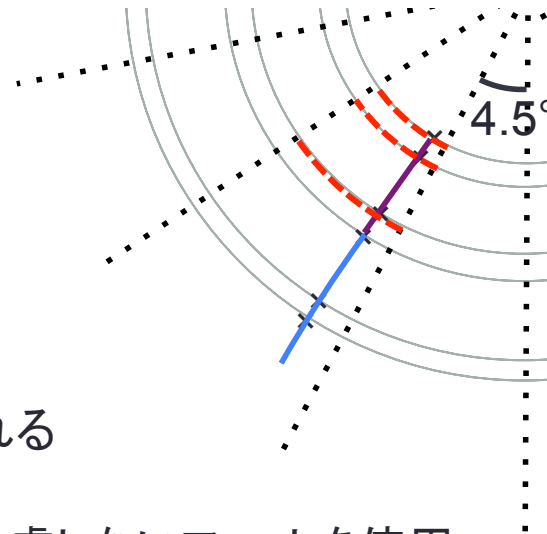
Φ 方向に80分割された領域の一區画

フィッター:

シンプル・ヘリックスフィット

外挿処理の問題:

- 隣の区画へは外挿されない
→ 幾つかの正しいヒットは無視される
- 区画の幅が一定
→ 多くの別のヒットに対しても処理される
- 多重クーロン散乱、エネルギー損失を考慮しないフィットを使用
→ 低横運動量のトラックの χ^2/ndf が高めに出やすく、
誤った外挿とみなされやすい



赤線：
外挿処理の範囲

FPCCDTrackFinderの外挿処理

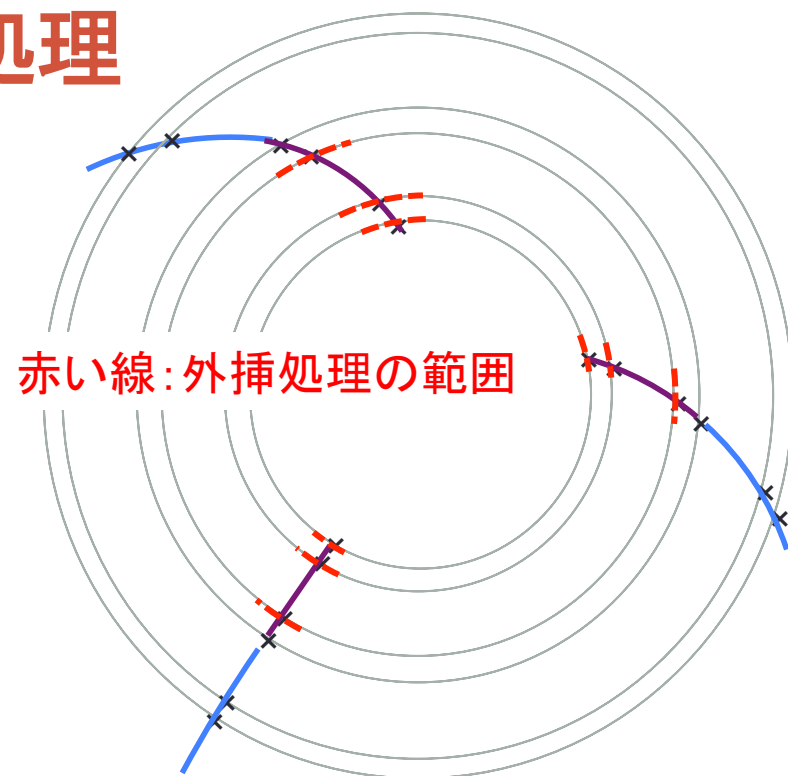
解決策:

1. フィッターに多重クーロン散乱、エネルギー損失を考慮するカルマンフィルターを使用

→ 計算量は増えるがより正當に
フィットが行われ、**低 P_T トラックが
生き残りやすい**

2. 外挿範囲をフィッターから得られるトラック
パラメターの誤差から決定

→ **効率的な外挿処理の範囲**が決まる



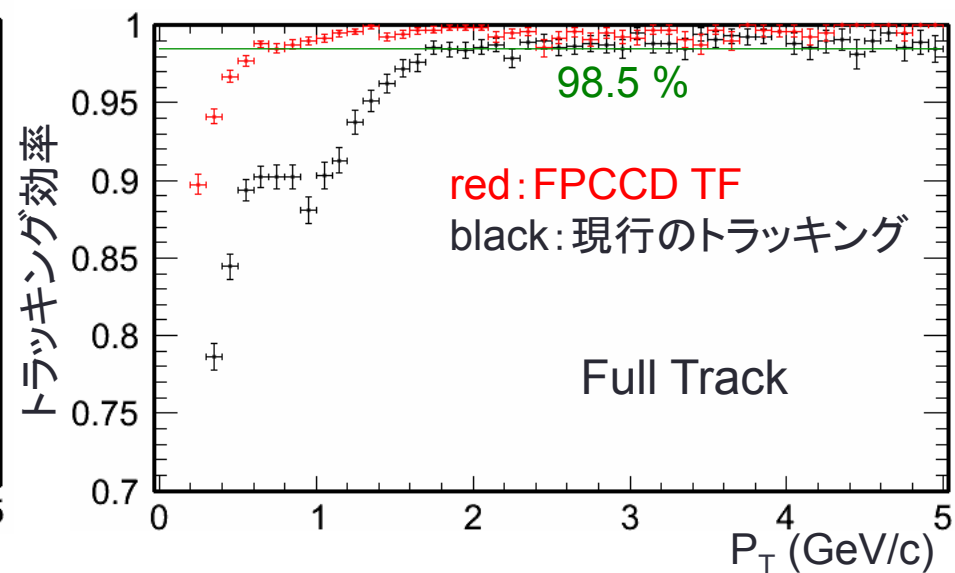
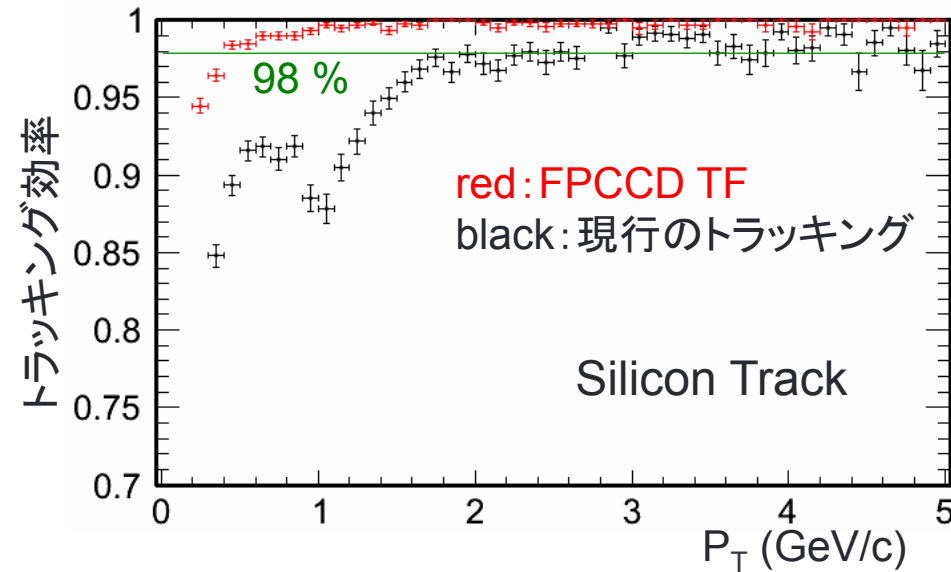
FPCCDTrackFinder VS 現行のトラッキング with FPCCD (P_T)

参考: TPCの内径、外径に届くのに必要な P_T

内径: 0.4 GeV/c

外径: 1.8 GeV/c

サンプル: $t\bar{t} \rightarrow 6 \text{ jets @ } 350 \text{ GeV}$ (ペアBG無し)



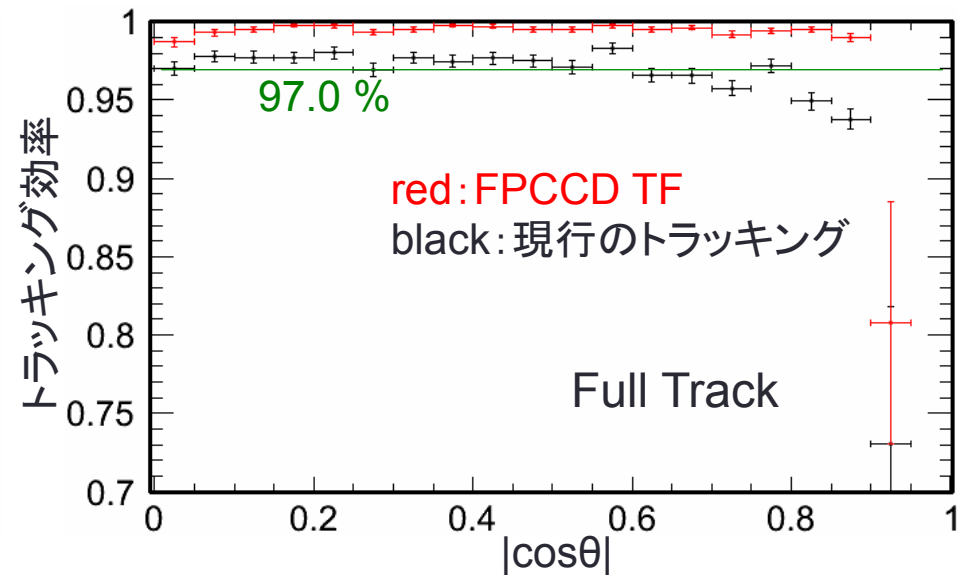
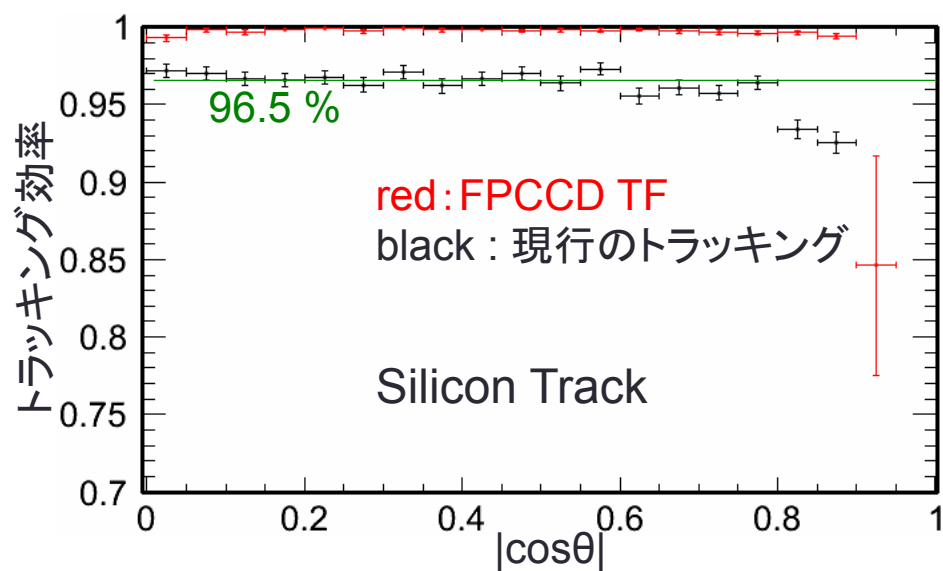
効率が ~ 99 % に改善 @ $P_T > 0.6 \text{ GeV/c}$

FPCCDTrackFinder VS 現行のトラッキング with FPCCD ($\cos\theta$)

参考: SITのアクセプタンス

$|\cos\theta| < 0.9$

サンプル: $t\bar{t} \rightarrow 6 \text{ jets @ } 350 \text{ GeV, } |P| > 1 \text{ GeV}/c$ (ペアBG無し)



効率が $\sim 99\%$ に改善 @ $|\cos\theta| < 0.9$

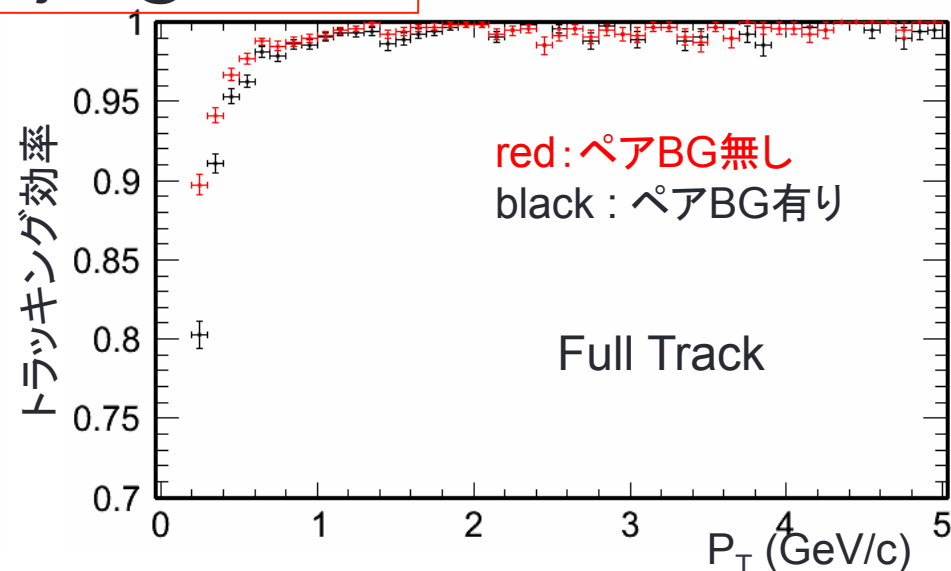
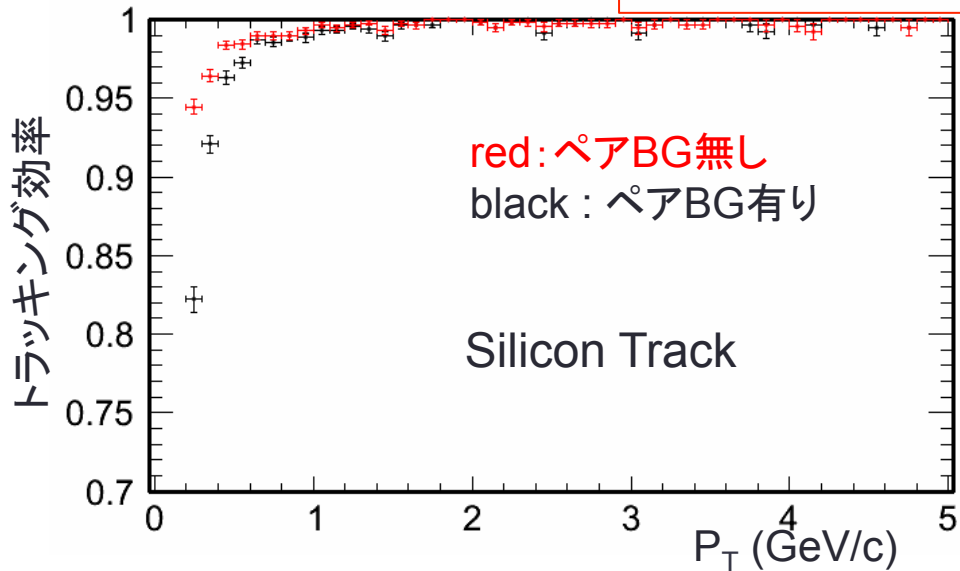
ペアBGが有る時のFPCCDTrackFinderの性能 (P_T)

参考: TPCの内径、外径に届くのに必要な P_T

内径: 0.4 GeV/c

外径: 1.8 GeV/c

サンプル: $tt \rightarrow 6 \text{ jets @ } 350 \text{ GeV}$



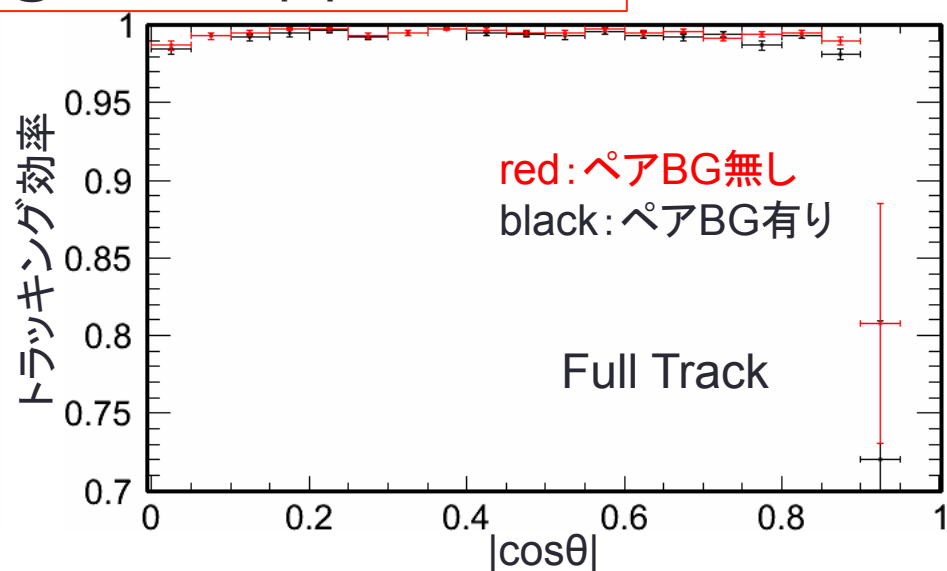
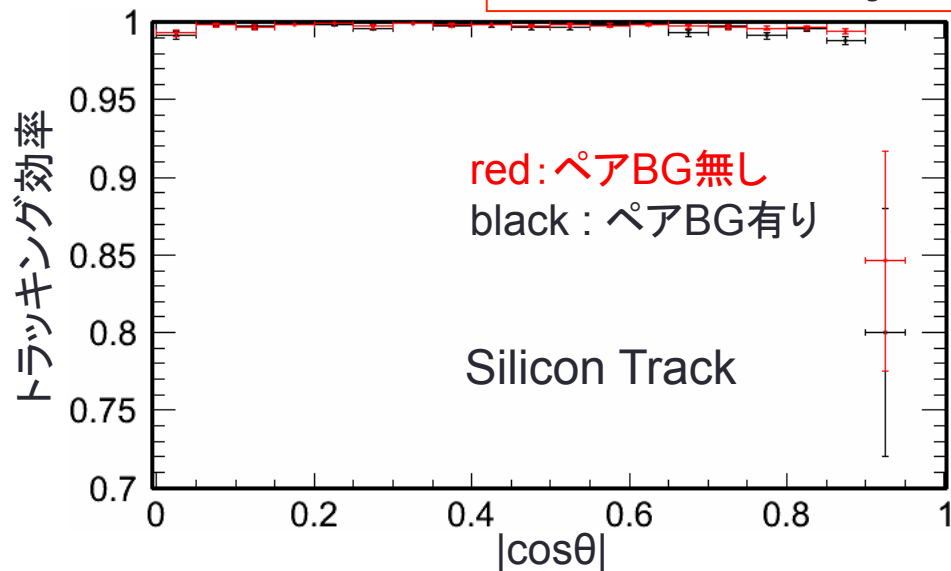
ペアBGを考慮しても $\sim 99\%$ を維持 @ $P_T > 0.6 \text{ GeV/c}$

ペアBGが有る時のFPCCDTrackFinderの性能 ($\cos\theta$)

参考: SITのアクセプタンス

$|\cos\theta| < 0.9$

サンプル: $t\bar{t} \rightarrow 6 \text{ jets @ } 350 \text{ GeV, } |P| > 1 \text{ GeV}/c$



ペアBGを考慮しても $\sim 99\%$ を維持 @ $|\cos\theta| < 0.9$

CPU時間とメモリの比較

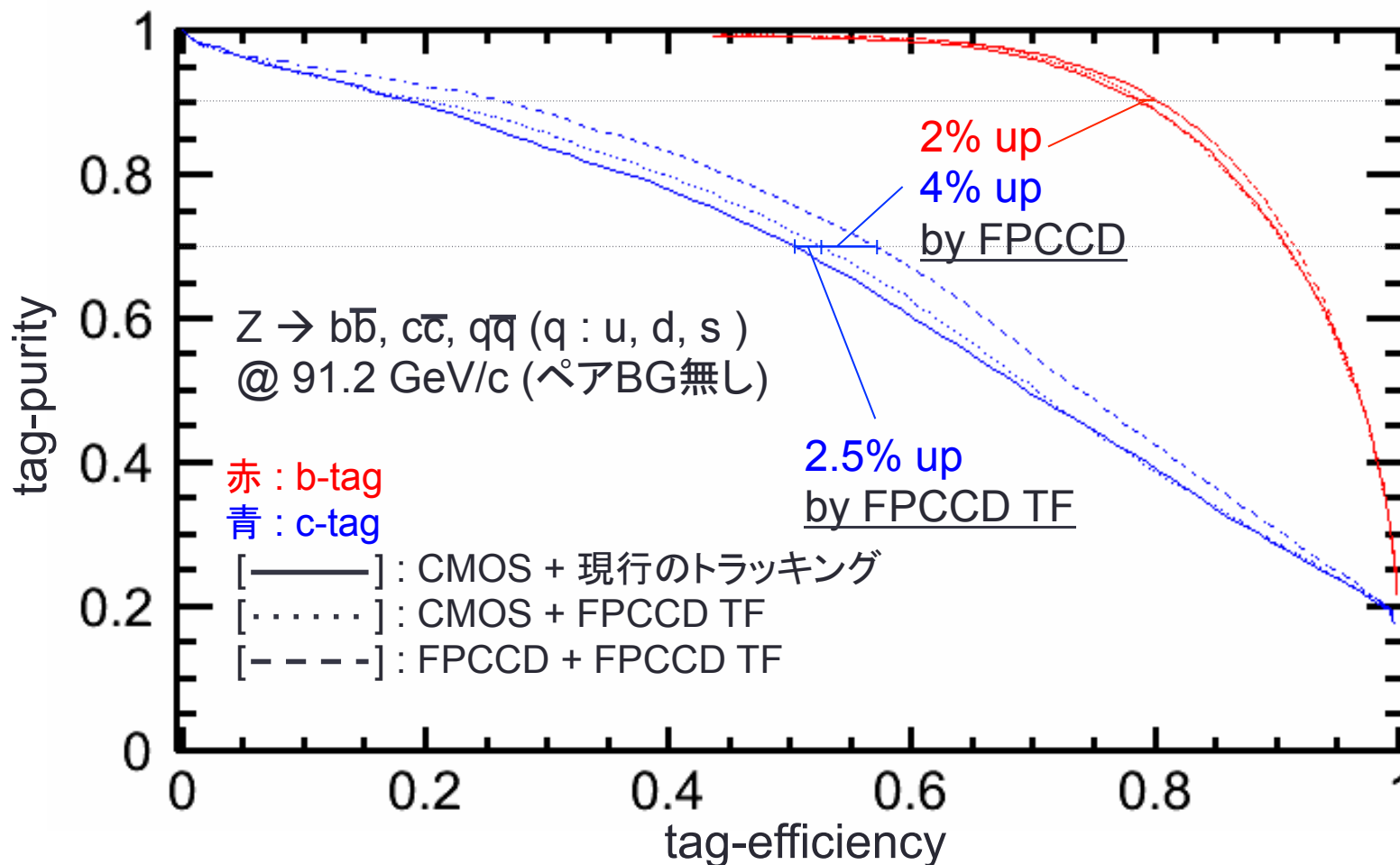
各値は $Z^* \rightarrow b\bar{b}$ @ 250 GeV のイベントサンプル 2000個についての平均値

VXD	tracking	ペアBG	CPU時間 [sec/evt]		最大メモリ [MB/evt]
			SiliconTracking	FullTracking	
CMOS	現行版	×	0.2	1.1	408.7
CMOS	現行版	○	342.0	6.8	561.5
CMOS	FPCCDTF	×	7.2	1.0	619.5
CMOS	FPCCDTF	○	34.0	3.0	709.6
FPCCD	FPCCDTF	×	5.6	1.0	623.0
FPCCD	FPCCDTF	○	407.6	27.7	2276.0

- CMOSの場合、ペアBG有りの時にFPCCDTrackFinderを使えば **CPU時間 ~1/10**
- FPCCDの場合、CPU時間が長いので、CPU時間削減が今後の課題

フレーバータグの性能評価

ペアBGが無い時のフレーバータグの性能評価

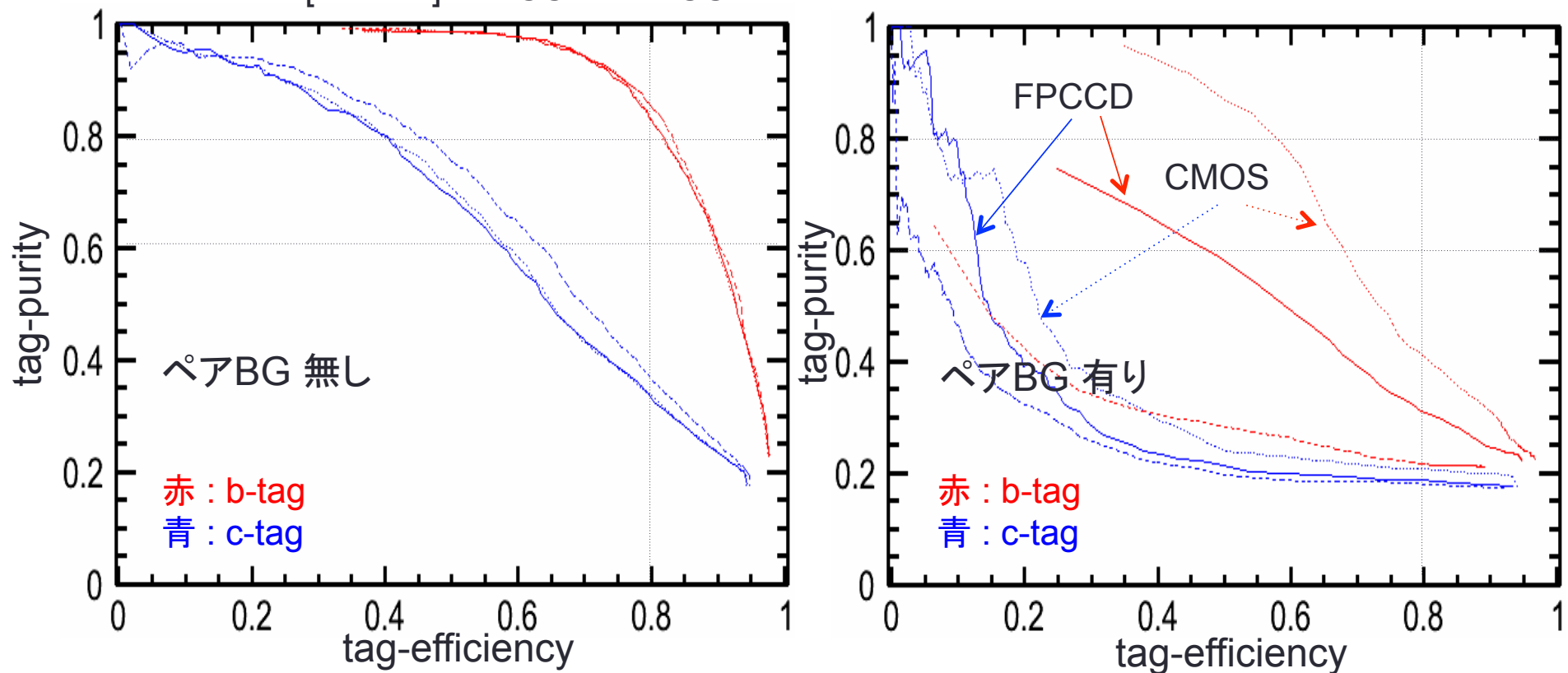


- FPCCDTrackFinderによりpurity 70% のc-tag efficiency が 2.5% 改善
- FPCCD崩壊点検出器を使用することで
 1. purity 90% のb-tag efficiency を 2% 改善
 2. purity 70% のc-tag efficiency を 4% 改善

ペアBGが有る時のフレーバータグの性能評価1

$$Z^* \rightarrow b\bar{b}, c\bar{c}, q\bar{q} \ (q : u, d, s) \ @ \ 250 \text{ GeV}$$

[——] : CMOS + 現行のトラッキング
 [.....] : CMOS + FPCCD TF
 [- - -] : FPCCD + FPCCD TF



- ペアBGによってフレーバータグの性能は低下する
- ペアBGが有る時も 現行のトラッキング より FPCCD TF の方が優勢
- ペアBGが有る時は FPCCD より CMOS の方が優勢

ペアBGによる性能の悪化

VXD	tracking	ペアBG	b-tag purity [%] @ eff. 80 %	c-tag purity [%] @ eff. 60 %
CMOS	現行版	×	52% ↓ 82.8	36% ↓ 56.4
CMOS	現行版	○	down ↓ 30.4	down ↓ 20.0
CMOS	FPCCDTF	×	42% ↓ 83.0	35% ↓ 58.1
CMOS	FPCCDTF	○	down ↓ 40.8	down ↓ 22.8
FPCCD	FPCCDTF	×	64% ↓ 85.5	45% ↓ 63.9
FPCCD	FPCCDTF	○	down ↓ 21.5	down ↓ 18.7

- ペアBGが有る時も現行のトラッキングよりFPCCD TFの方が
 - efficiency 80 % の b-tag の purity で 10% 優勢
 - efficiency 60 % の c-tag の purity で 3% 優勢
- ペアBGがある時は CMOS より FPCCD の方が
 - efficiency 80 % の b-tag の purity で 19 % 劣勢
 - efficiency 60 % の c-tag の purity で 3% 劣勢

再構成されたb-jet 中にあるトラックの P_T 分布

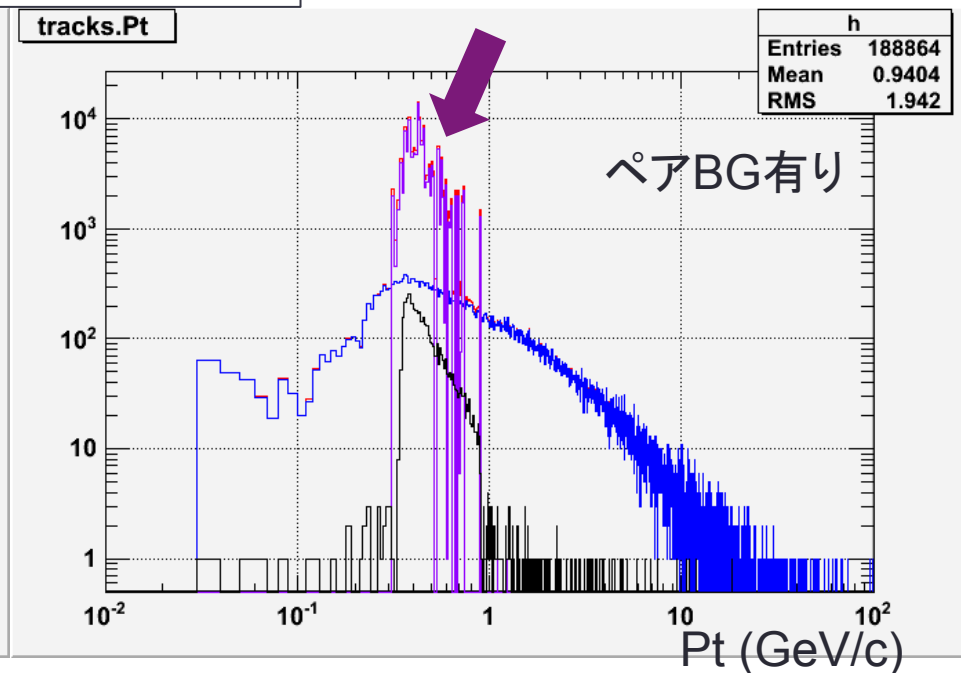
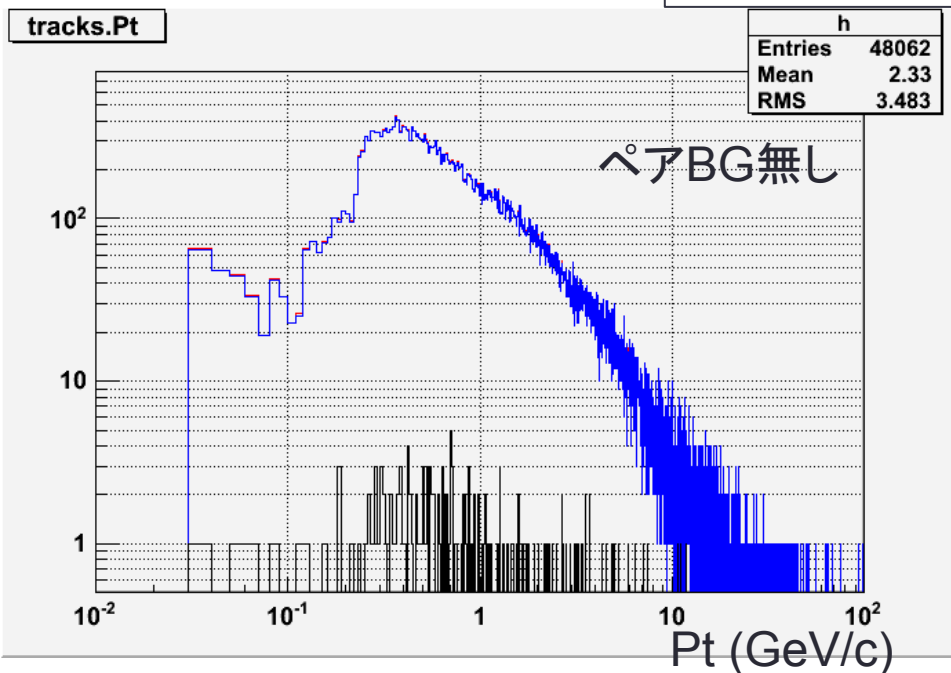
(FPCCD + FPCCDTFの場合)

red: 全トラック

blue: purity > 0.75 のトラック

black: purity < 0.75 のトラック

purple: ペアBG のトラック



- フレーバータグの性能悪化は、
大量のペアBGがb-jetの再構成に誤使用されるため

ペアBGTラックを落とすためのトラックの要求

トラックの要求: **SIT hit ≥ 1** || **TPC hit ≥ 10** || **$|\cos\theta| > 0.9$**

← 多くのペアBGTラックは SIT、TPCのヒットを持たない

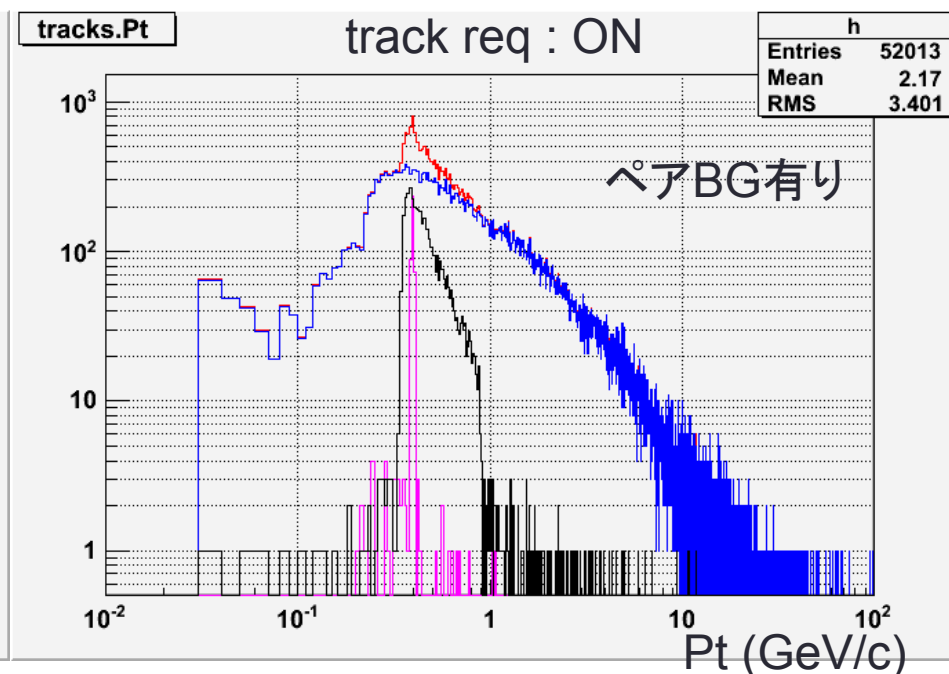
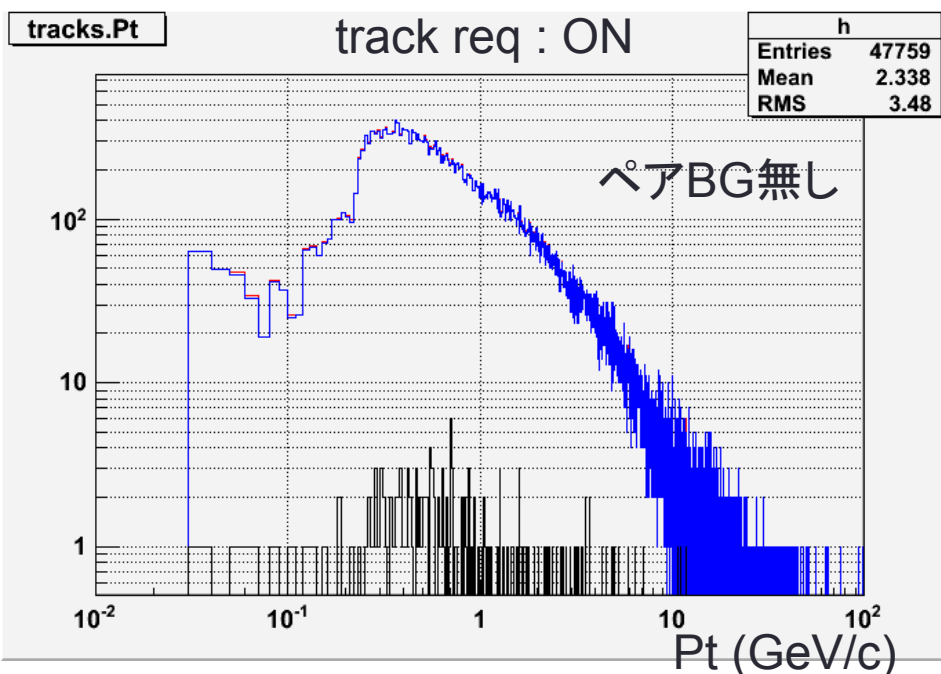
← $|\cos\theta| > 0.9$ のトラックはSIT、TPCのアクセプタンスを考慮して
SIT、TPCのヒットを要求しない

red: 全トラック,

blue: purity > 0.75 のトラック,

black: purity < 0.75 のトラック,

purple: ペアBG のトラック

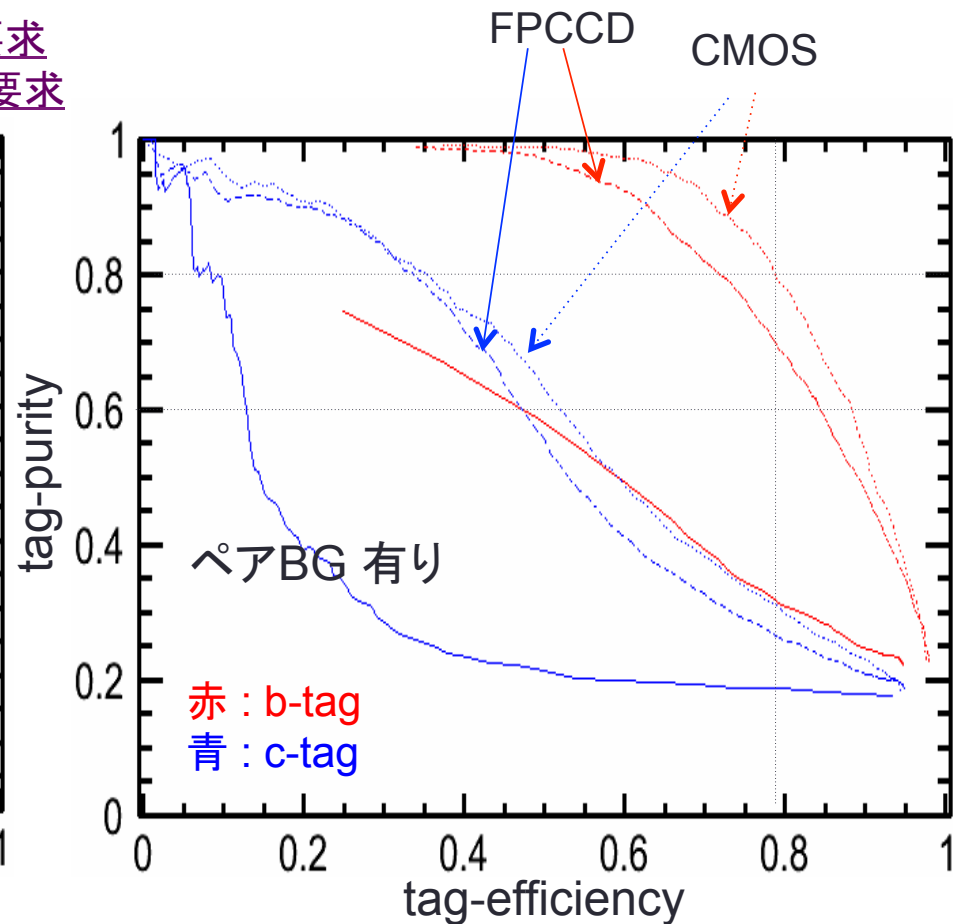
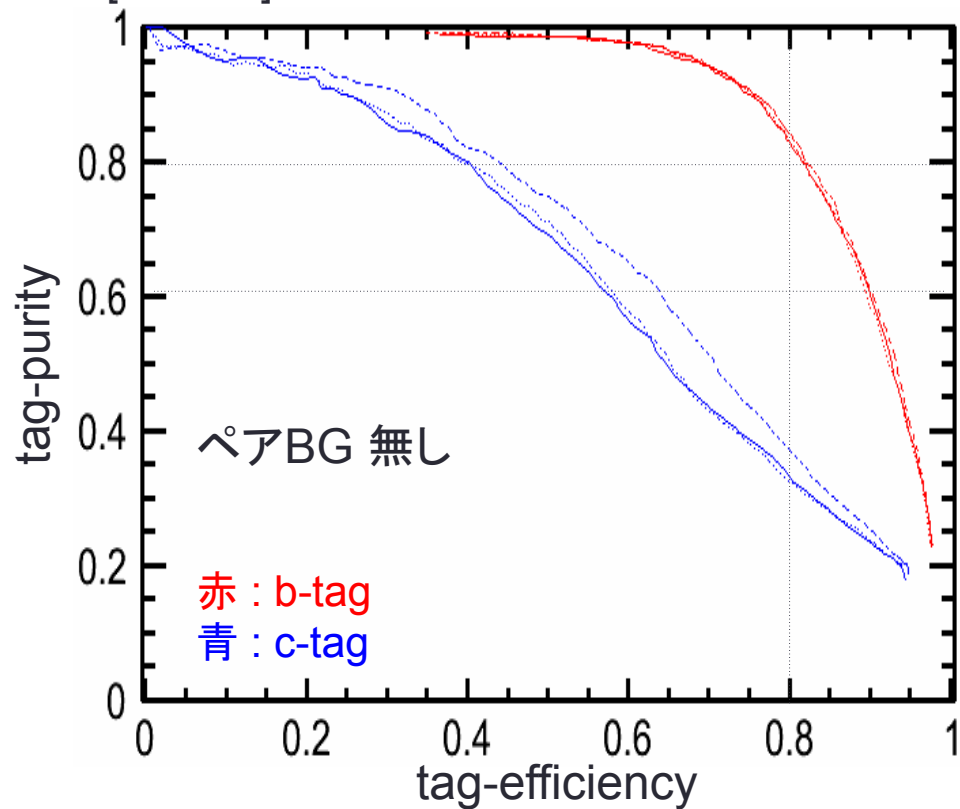


トラックの要求によりペアBGTラックが大幅に減少した

ペアBGが有る時のフレーバータグの性能評価2

$Z^* \rightarrow b\bar{b}, c\bar{c}, q\bar{q}$ (q : u, d, s) @ 250 GeV

[——] : CMOS + 現行のトラッキング
 [.....] : CMOS + FPCCD TF + トラックの要求
 [- - -] : FPCCD + FPCCD TF + トラックの要求



- トラックの要求により悪化がある程度抑えられる
- 依然としてペアBGが有る場合はFPCCDはCMOSより劣勢

トラックの要求による性能の改善

VXD	tracking	ペアBG	Track Req.	b-tag purity [%] @ eff. 80 %	c-tag purity [%] @ eff. 60 %
CMOS	std	×	×	82.8	56.4
CMOS	std	○	×	30.4	20.0
CMOS	FPCCDTF	×	×	83.0	58.1
CMOS	FPCCDTF	×	○	82.9	57.4
CMOS	FPCCDTF	○	×	37% ↓ 40.8	27% ↓ 22.8
CMOS	FPCCDTF	○	○	recover ↓ 77.6	recover ↓ 49.4
FPCCD	FPCCDTF	×	×	85.5	63.9
FPCCD	FPCCDTF	×	○	84.1	65.5
FPCCD	FPCCDTF	○	×	46% ↓ 21.5	23% ↓ 18.7
FPCCD	FPCCDTF	○	○	recover ↓ 67.8	recover ↓ 41.6

- トラックの要求による改善後もCMOSに比べFPCCDは劣勢
 - efficiency 80 % の b-tag の purity が 10 % 劣勢
 - efficiency 60 % の c-tag の purity が 8 % 劣勢

まとめ

本研究ではFPCCD崩壊点検出器のシミュレーションによる性能評価と新トラッキングアルゴリズムを開発した

◆ 占有率の性能評価

- 1 TeVのビームランでは最内層の占有率が12.75 %と問題がある

◆ インパクトパラメータ分解能の性能評価

- 要求性能を満たし、高運動量領域では $\sim 1 \mu\text{m}$ の分解能を持つ

◆ FPCCDTrackFinderの開発

- トラッキング効率が $P_T > 0.6 \text{ GeV}/c$ 、 $|\cos\theta| < 0.9$ で $\sim 99\%$ まで改善
- ペアBGを考慮してもトラッキング効率が $P_T > 0.6 \text{ GeV}/c$ 、 $|\cos\theta| < 0.9$ では $\sim 99\%$ を維持

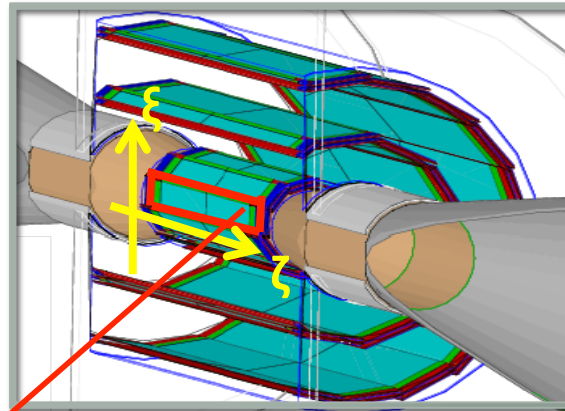
◆ フレーバータグの性能評価

- FPCCDTrackFinderにより purity 70 % の c-tag efficiency が 2.5 % 改善
- FPCCD崩壊点検出器を使用することによって、purity 90% のb-tag efficiency を2 %、purity 70% のc-tag efficiency を4 % 改善
- ペアBGによってフレーバータグの性能が低下するが、ペアBGトラックを除外するトラックの要求をつけるとフレーバータグの性能がある程度回復することが分かった

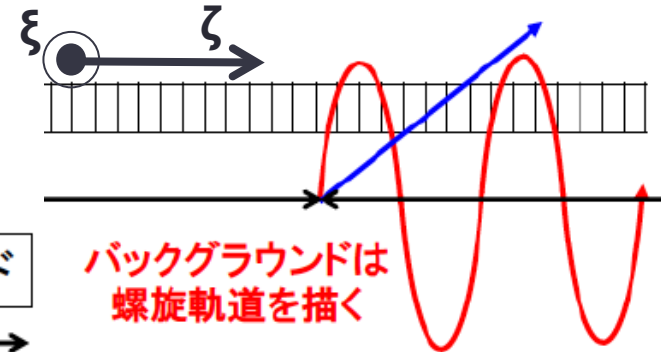
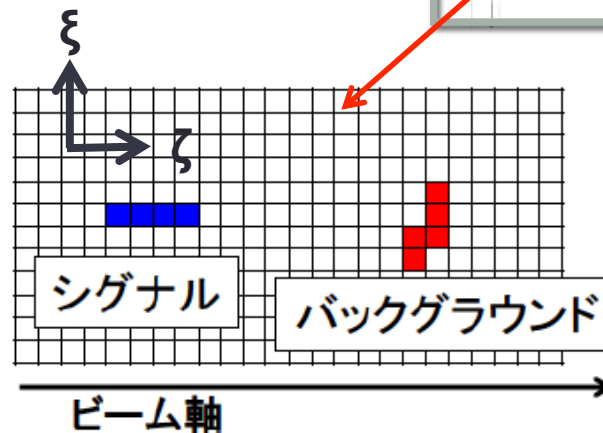
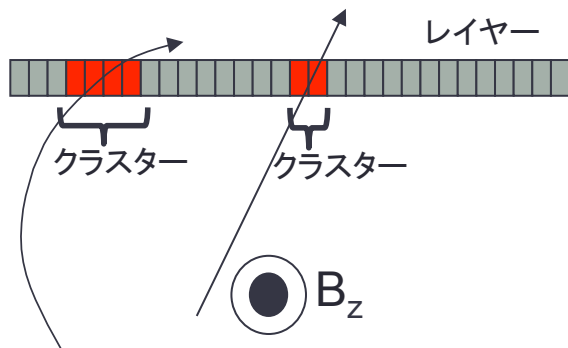
Backup

ペアBGクラスター除外アルゴリズムの開発

クラスター形状によるカット



シグナル粒子 と ペアBG の違い:



第一種クラスターカット:

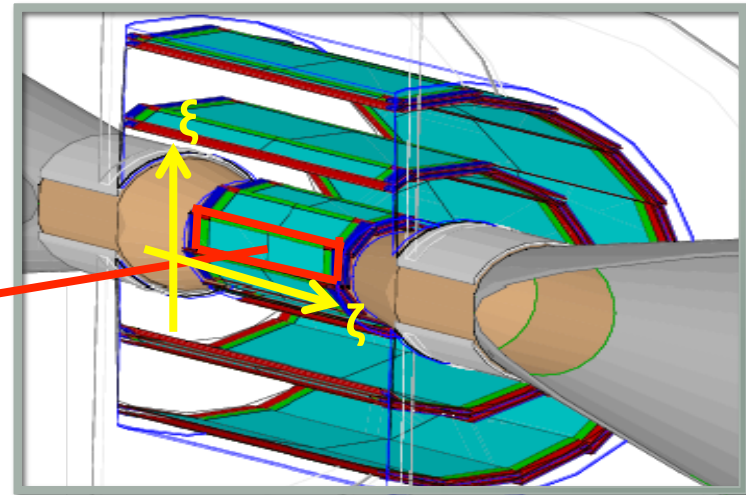
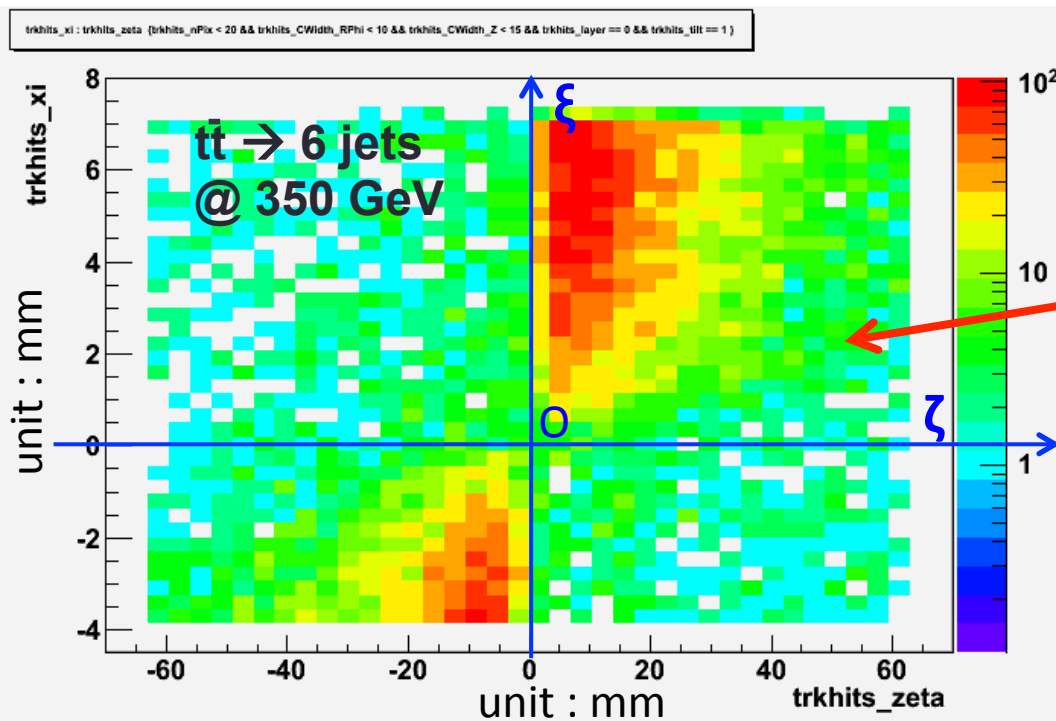
- ペアBGクラスターは比較的長いことを利用する

<サンプル>

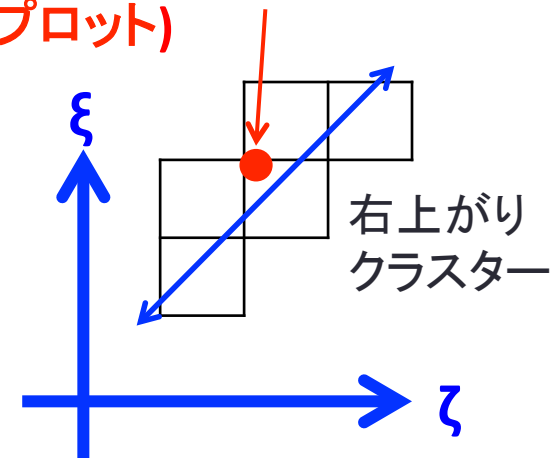
$t\bar{t} \rightarrow 6 \text{ jets @ } 350 \text{ GeV}$

レイヤー (ピクセル幅)	ピクセル数	ζ 方向の クラスター幅 [ピクセル幅]	ξ 方向の クラスター幅 [ピクセル幅]	シグナル クラスター 残存率 [%]	ペアBG クラスター 残存率 [%]
0 (5 μm)	< 20	< 15	< 10	99.28	92.85
2 (10 μm)	< 15	< 8	< 6	99.21	91.81

クラスタの座標と傾きの相関を利用したカット

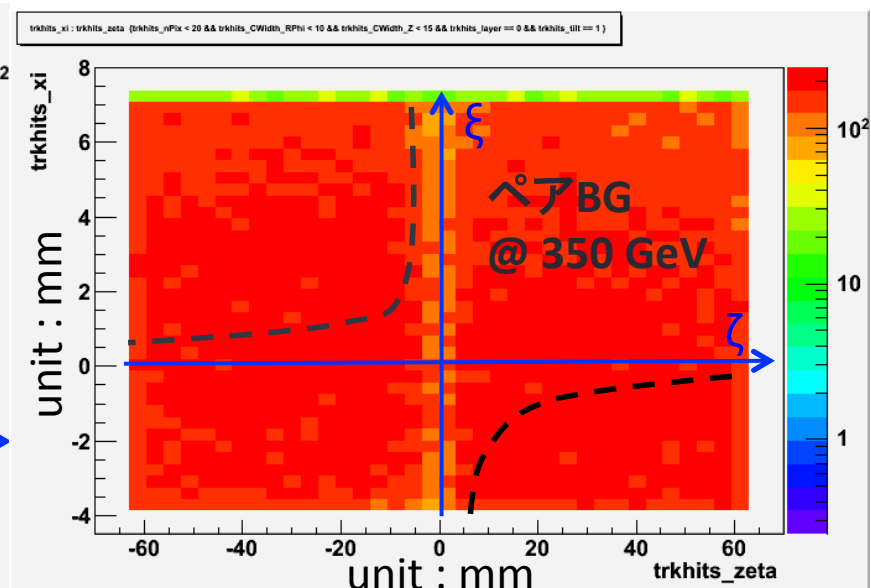
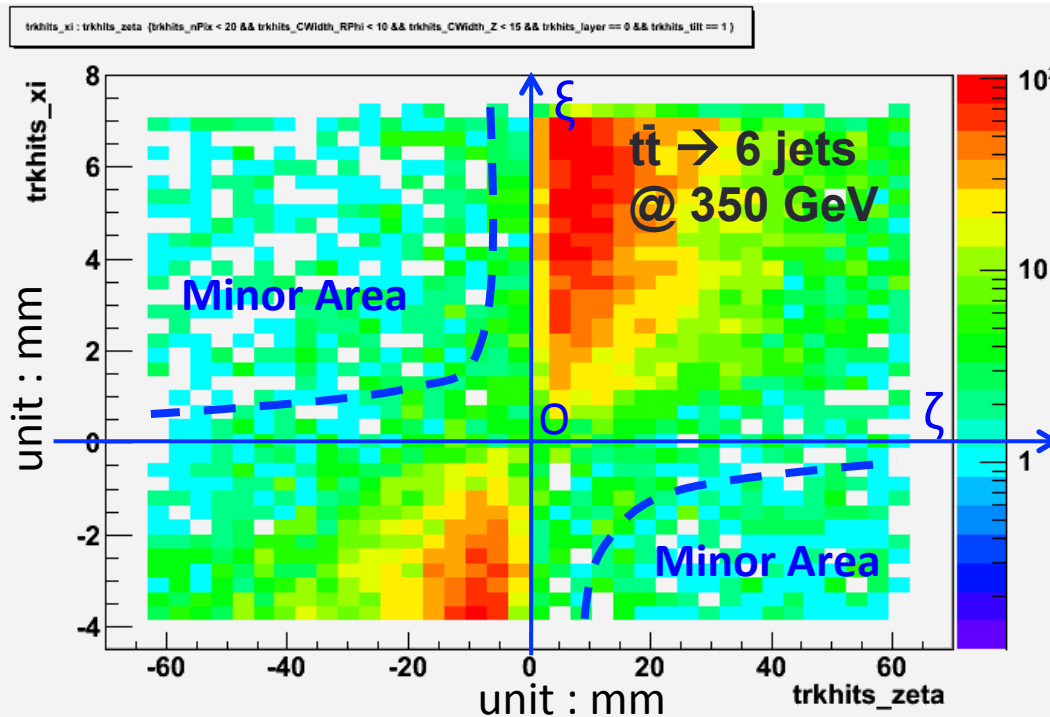


クラスタ形状、Energy deposit
から計算された**ヒットの位置**
(これをプロット)



上のヒストは右図のような
(- ξ , - ζ)方向 から (+ ξ , + ζ)方向 に伸びる
“右上がりのクラスタ”から計算された
ヒットの位置をプロットしてる

クラスターの座標と傾きの相関を利用したカット



右上がりのペアBGクラスターは
一様に分布しているのが分かる

傾きカット:

- Minor Area $\equiv \xi \times \zeta < Z_{\text{par}} : (Z_{\text{par}} > 0)$ をカット (左上がりについても行う)

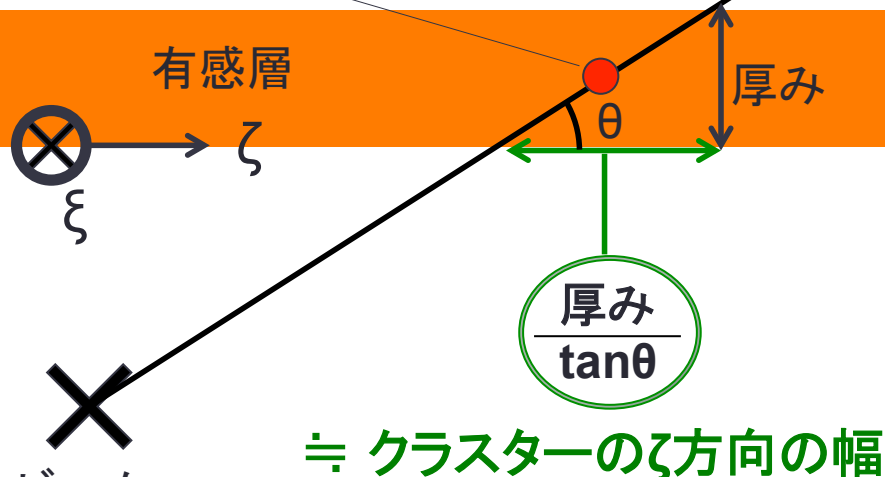
<第一種クラスターカット + 傾きカット>

<サンプル>
tt \rightarrow 6 jets
@ 350 GeV

レイヤー (ピクセル幅)	Z_{par} [mm ²]	シグナルクラスター 残存率 [%]	ペアBGクラスター 残存率 [%]
0 (5 μ m)	90	98.33	81.41
2 (10 μ m)	280	98.24	81.91

クラスタのz座標とζ方向の幅の相関を利用したカット

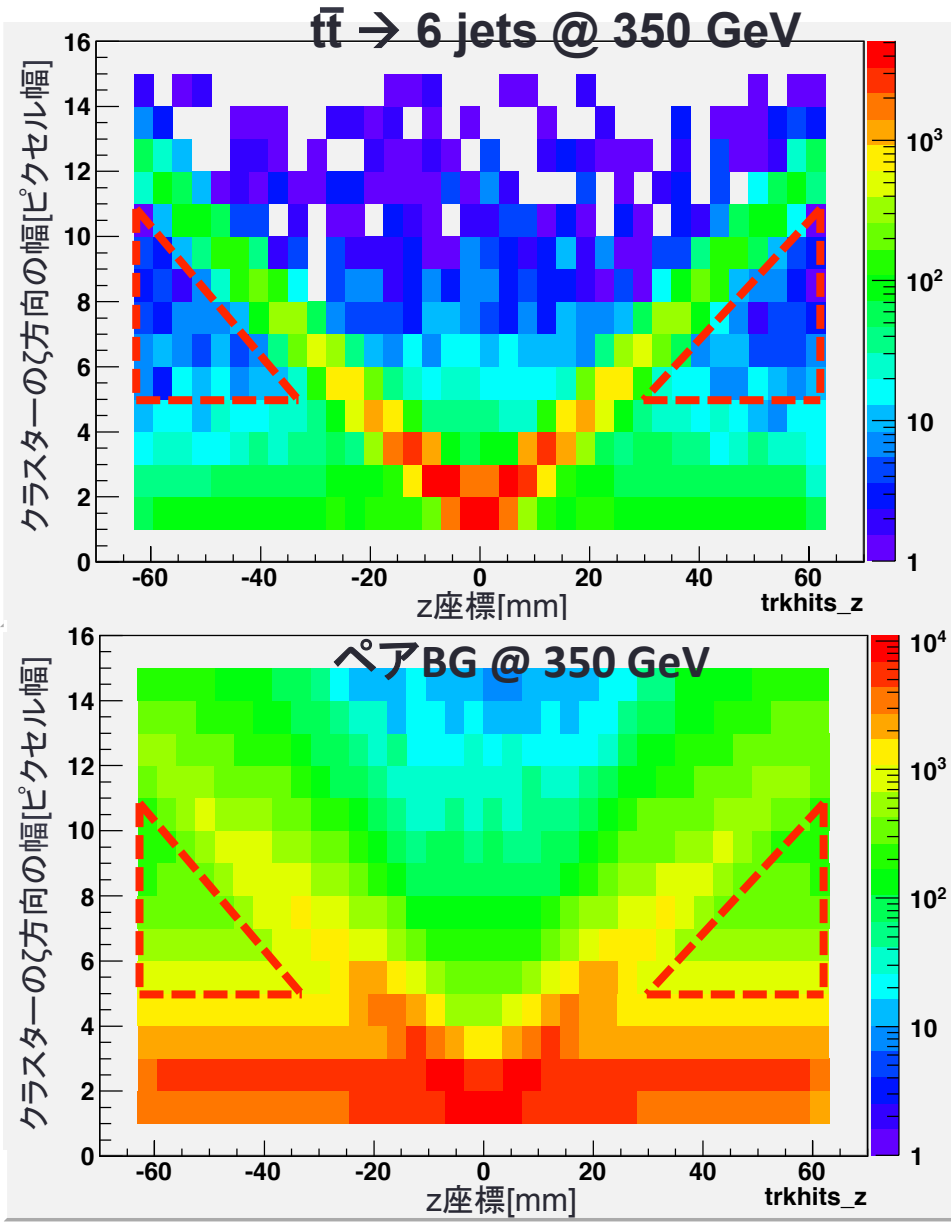
クラスタから計算される
粒子の通過点



ビーム
衝突点

ペアBGは衝突点から比較的離れたところから発生するため、z座標とζ方向の幅の相関は弱い(右図)

赤い三角形:
 $y_1 = \text{厚み} / |\tan\theta| / \text{ピクセル幅} - B_{\text{par}}$
 $y_2 = C_{\text{par}}$
 の範囲をカットする(ζ幅カット)



第一種クラスターカット + 傾きカット + 幅カット

<サンプル>

$t\bar{t} \rightarrow 6 \text{ jets @ } 350 \text{ GeV}$

(括弧内はピクセルヒット残存率)

レイヤー (ピクセル幅)	Z_{par} [mm ²]	B_{par} [ピクセル幅]	C_{par} [ピクセル幅]	シグナル クラスター 残存率 [%]	ペアBG クラスター 残存率 [%]
0 (5 μm)	90	2	4	98.00 (93.42)	79.06 (58.43)
1 (5 μm)	90	2	4	98.18 (93.51)	79.79 (60.34)
2 (10 μm)	280	1	2	98.13 (92.16)	80.85 (60.41)
3 (10 μm)	280	1	2	98.13 (92.73)	81.16 (61.42)
4 (10 μm)	600	1	2	98.73 (94.08)	88.02 (69.18)
5 (10 μm)	600	1	2	98.63 (94.04)	87.56 (68.91)

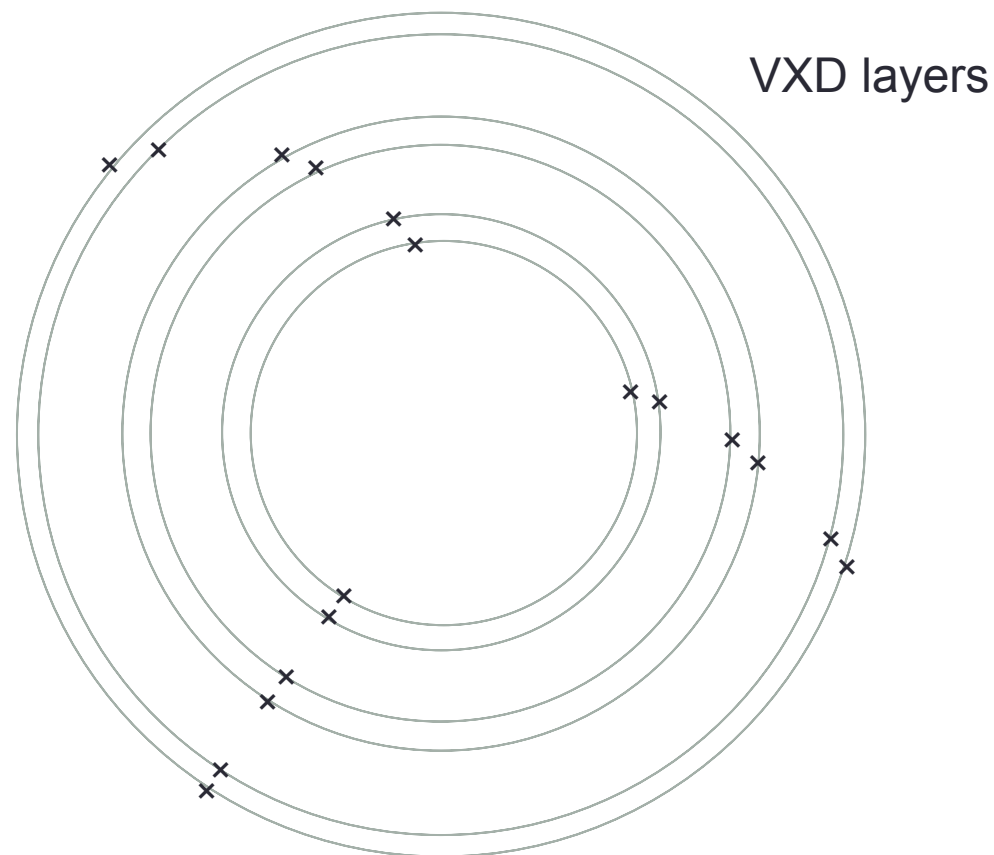
トラッキング効率、フレーバータグの性能評価
では上記のクラスターカットを適用して評価している

ヒットのデジタル化について

- 本研究において、FPCCD用のディジタイザーはFPCCDDigitizer, FPCCDClusteringを使用
- ピクセルヒットはFPCCDDigitizer, FPCCDClusteringによりランダウ分布、閾値、通過距離を考慮して生成される
 - 閾値: 0.725 keV
 - 1ADC カウントあたり0.091 keV
 - ADC カウントは7bit 使用

現行のシリコントラッキング

本当はSITも関係するが
簡単のためVXDだけで考える

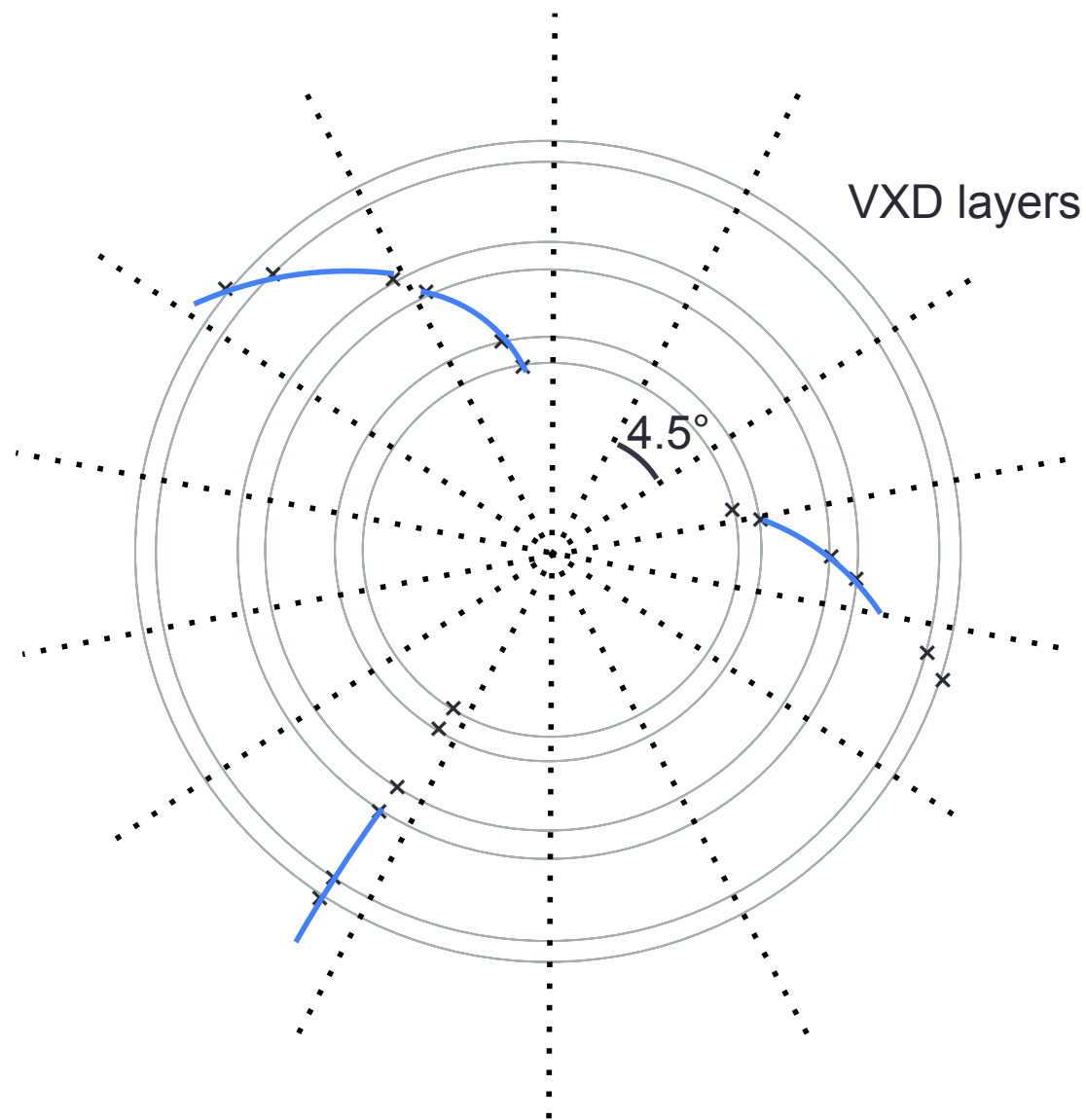


簡単のためVXD layersを円筒型で近似

現行のシリコントラックリング

トラックシード生成

φ方向に4.5°ずつ区切られた
各エリア内にある3層上の
各ヒットからトラックシードを生成



現行のシリコントラッキング

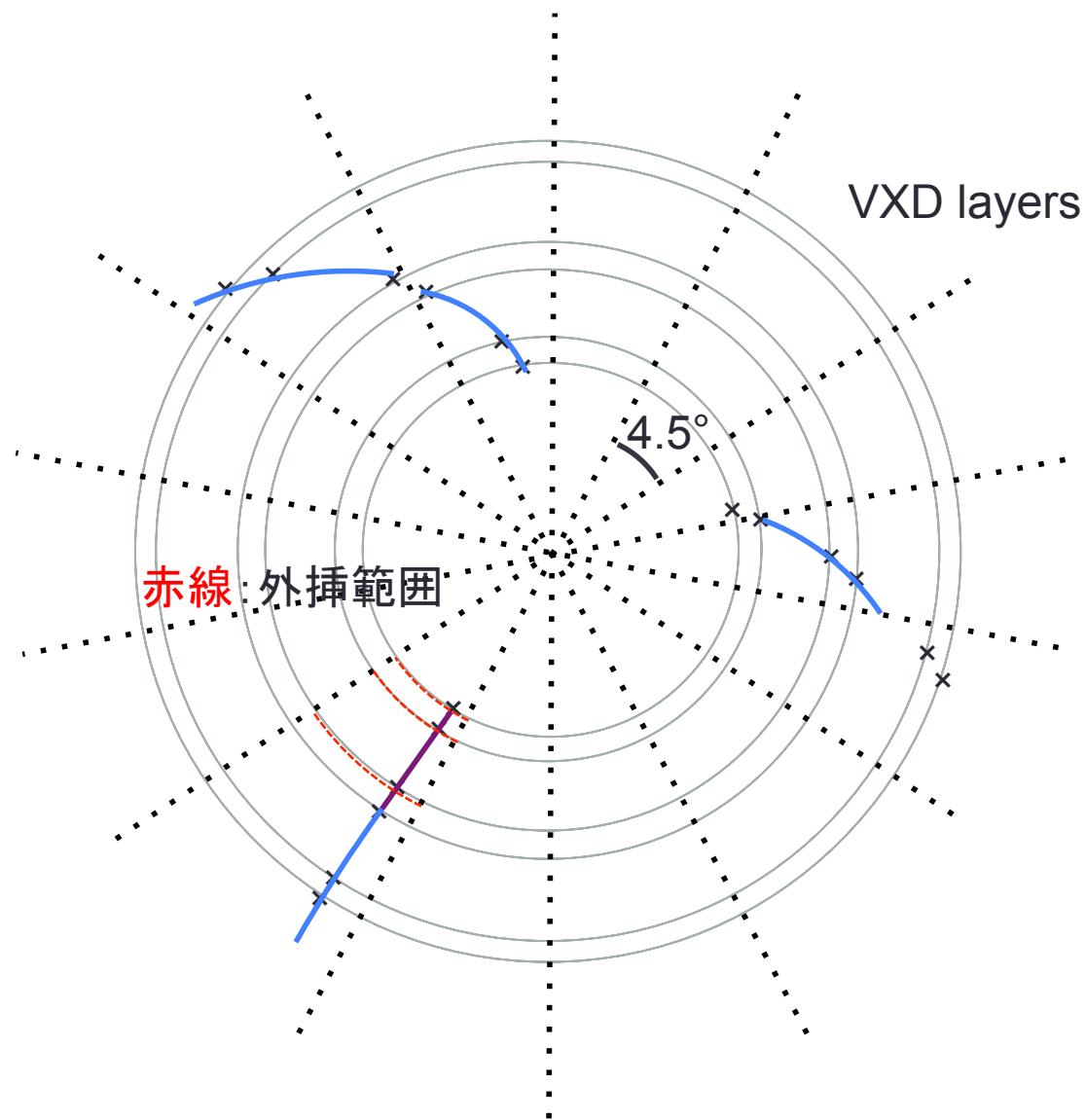
トラックシード生成



外挿処理

外挿するエリア:
同様にφ方向に
区切られたエリア内

使用フィッター:
シンプル・ヘリックスフィット



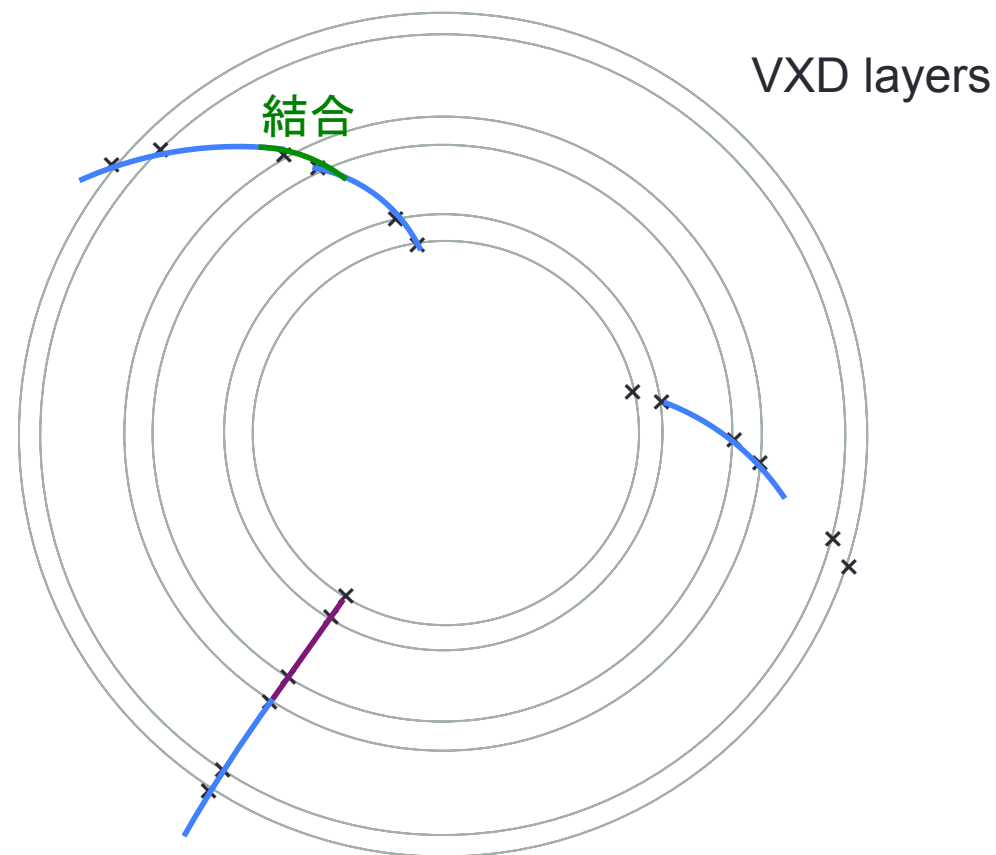
現行のシリコントラッキング

トラックシード生成

外挿処理

細切れトラックの結合

可能ならトラックと
トラックを結合



現行のシリコントラッキング

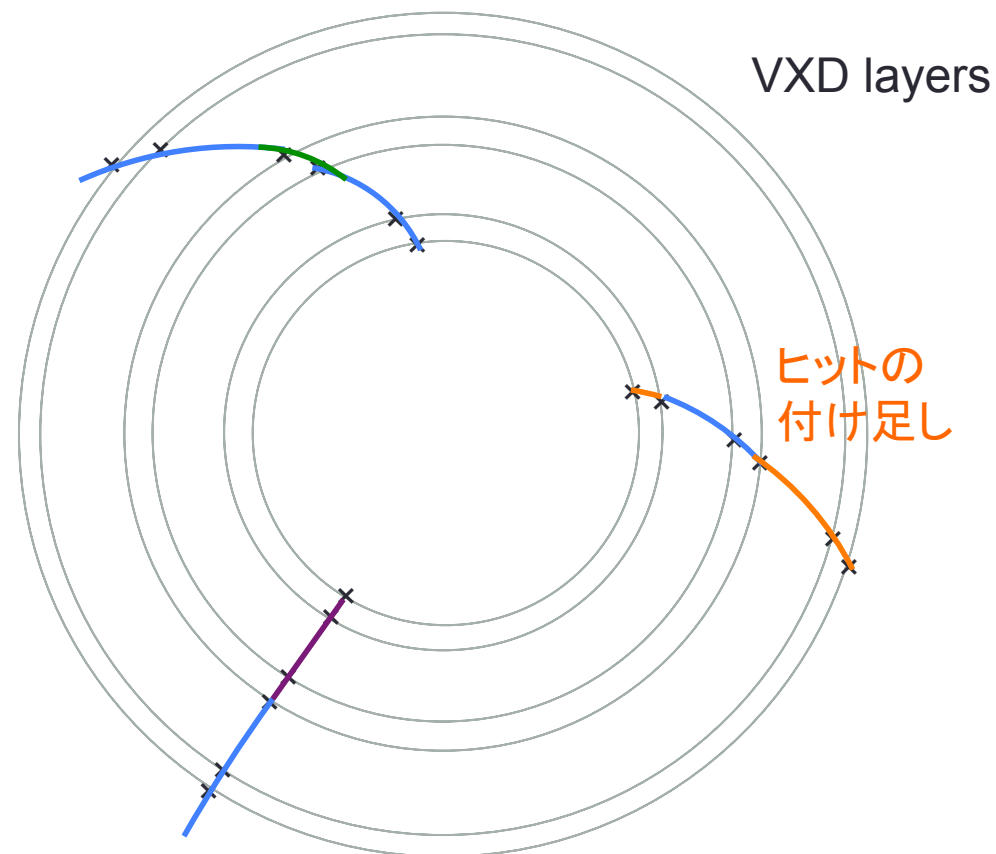
トラックシード生成

外挿処理

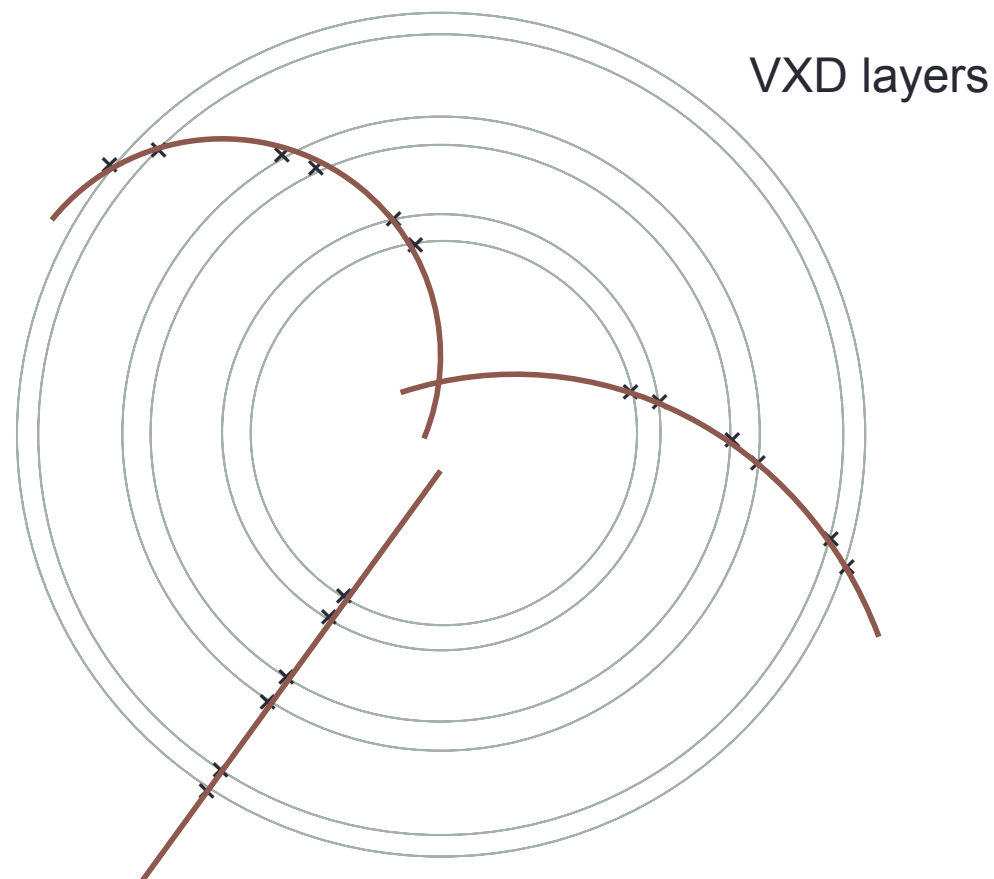
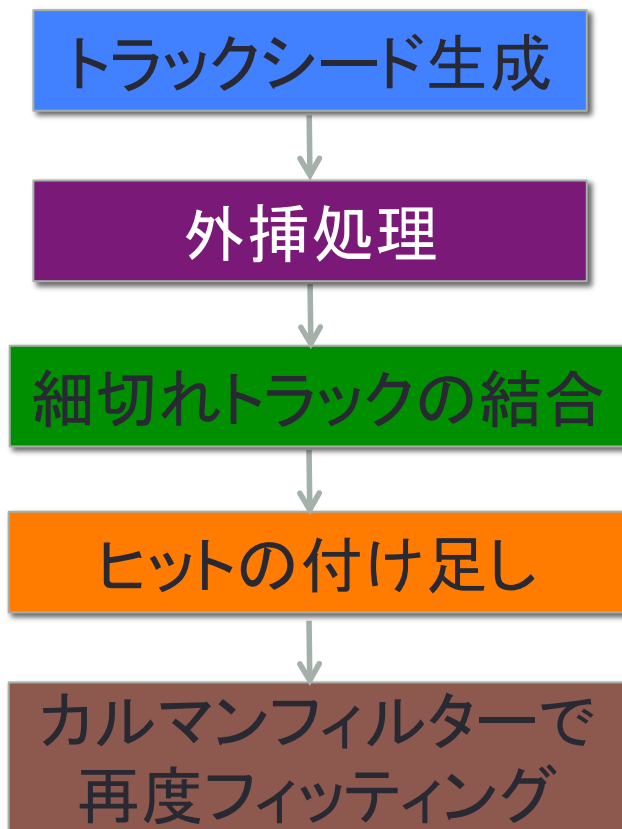
細切れトラックの結合

ヒットの付け足し

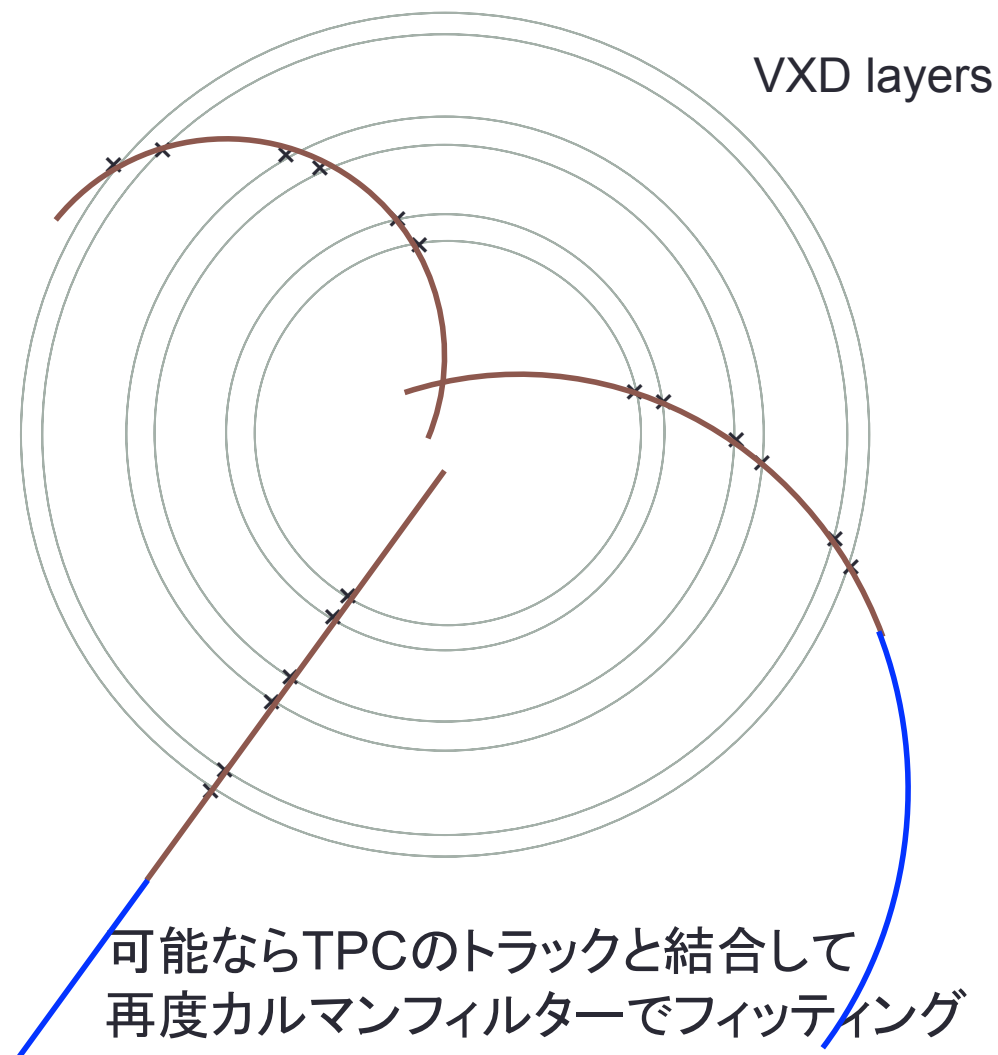
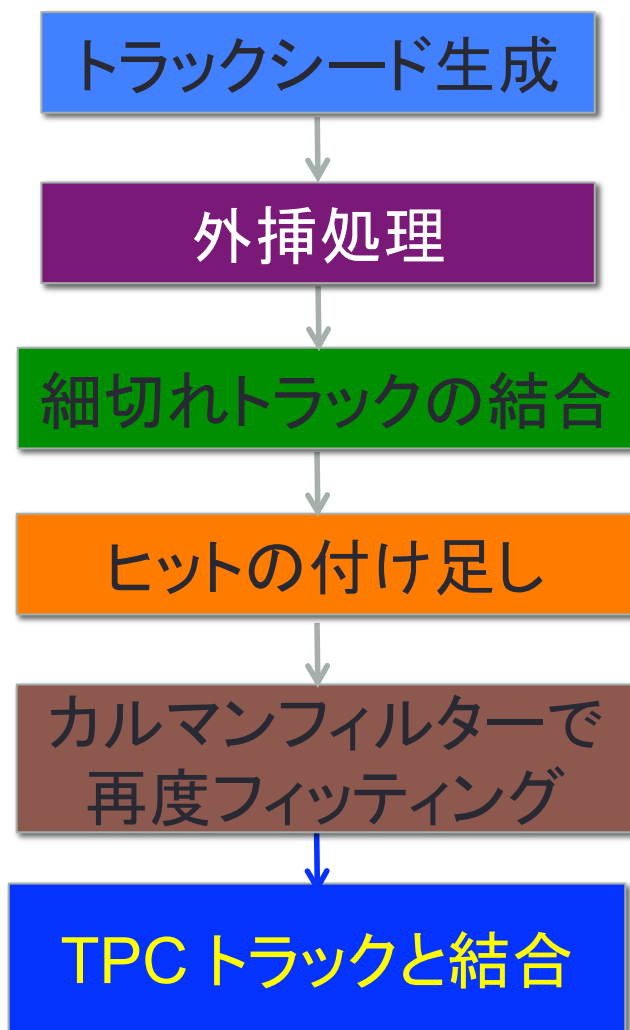
残っているトラックに
可能な限りヒットを付け足す



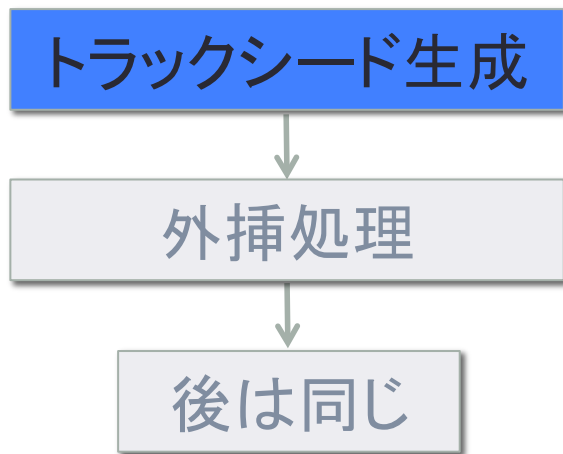
現行のシリコントラッキング



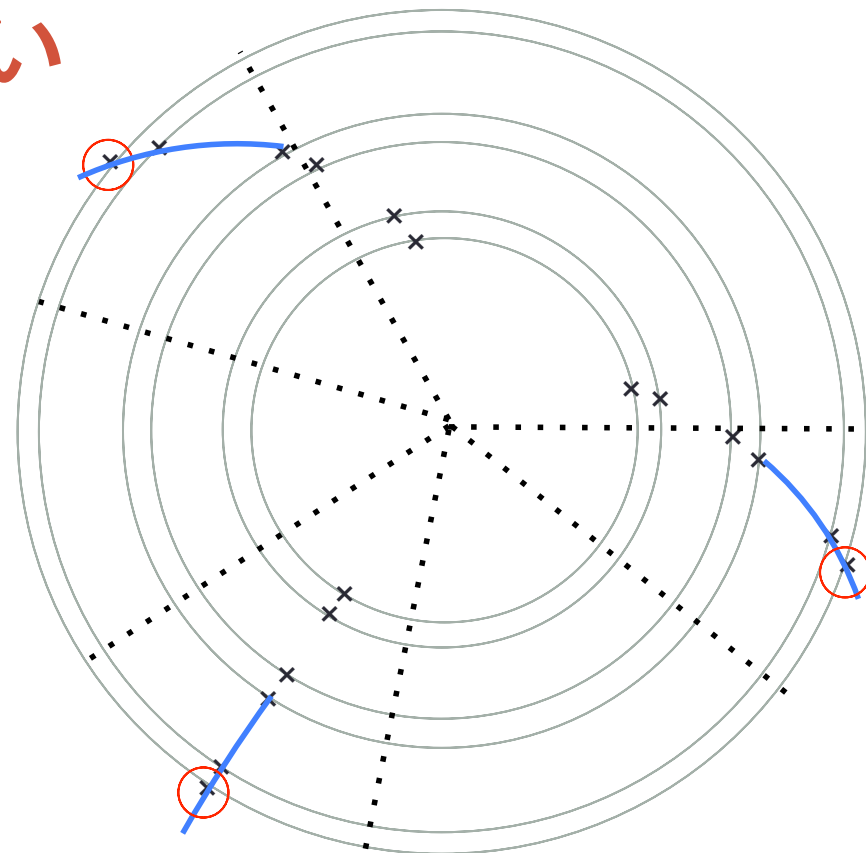
Full トラック



現行版とFPCCD専用の違い



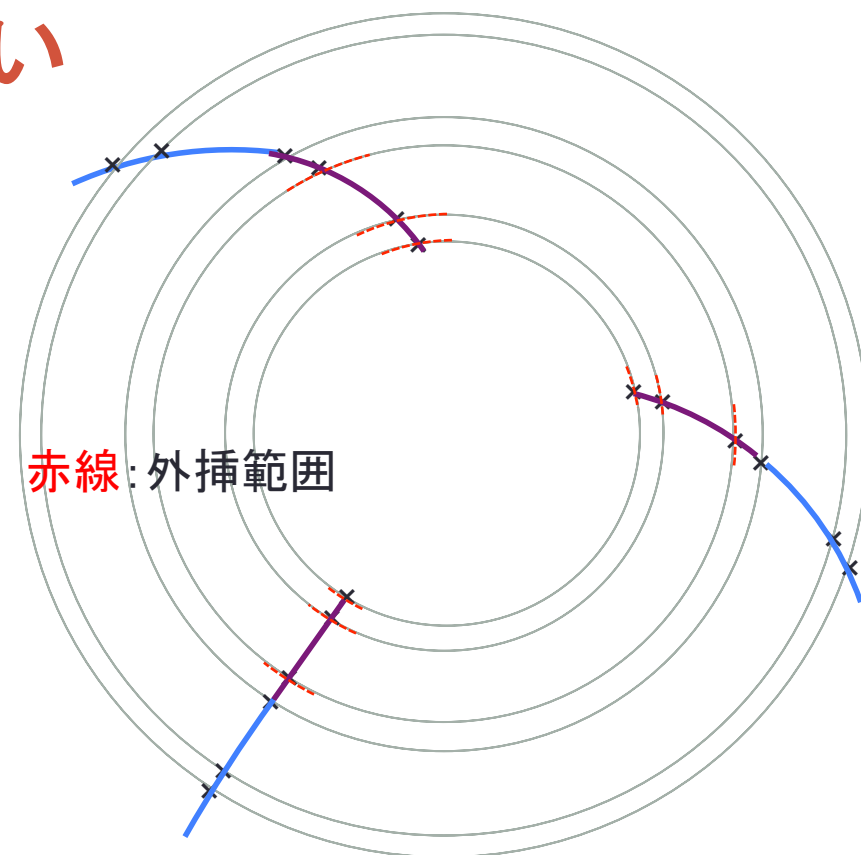
(現行版)
 ϕ 方向に 4.5° ずつ区切られた
 各エリア内にある3層上の
 各ヒットからトラックシードを生成



(FPCCD版)
 最外層のヒットを基準に $P_T > 0.18 \text{ GeV}/c$
 のトラックを拾える ϕ 幅を計算

そのエリア内にある3層上の
 各ヒットからトラックシードを生成
 (外側3層のみ使用)

現行版とFPCCD専用の違い



(現行版)

外挿するエリア:

Φ 方向に

区切られたエリア内

使用フィッター:

シンプル・ヘリックスフィット

(FPCCD版)

外挿するエリア:

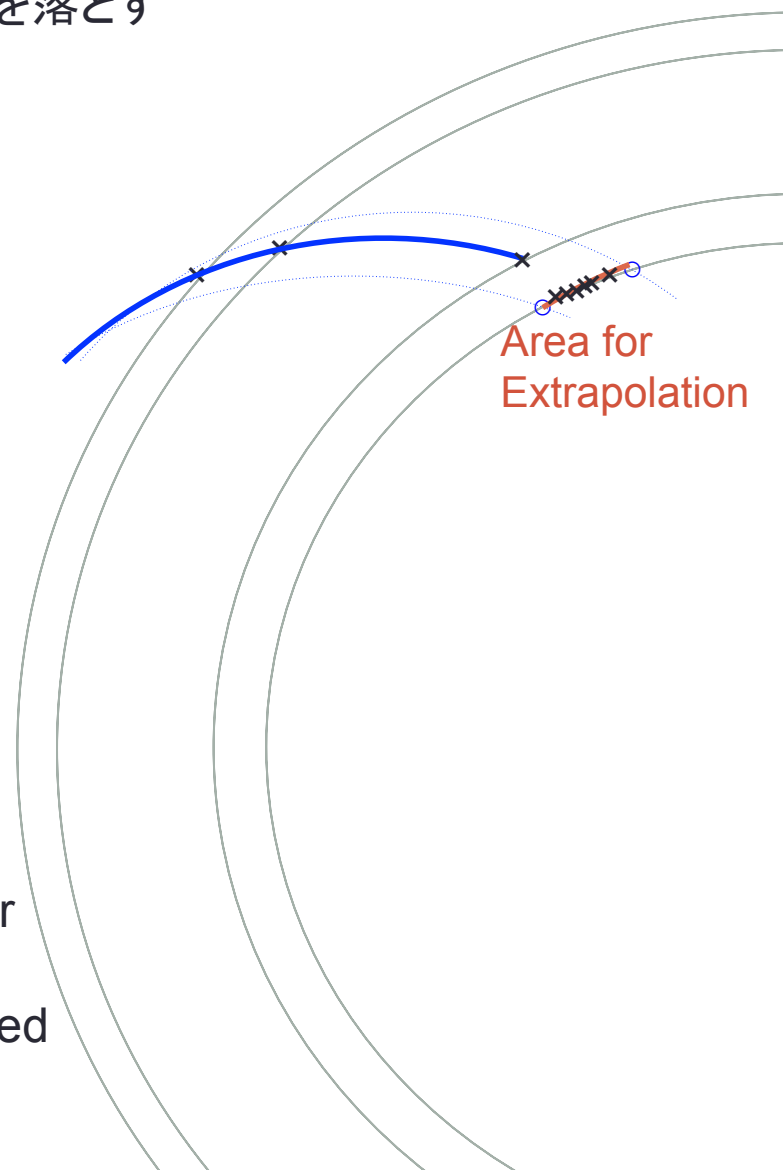
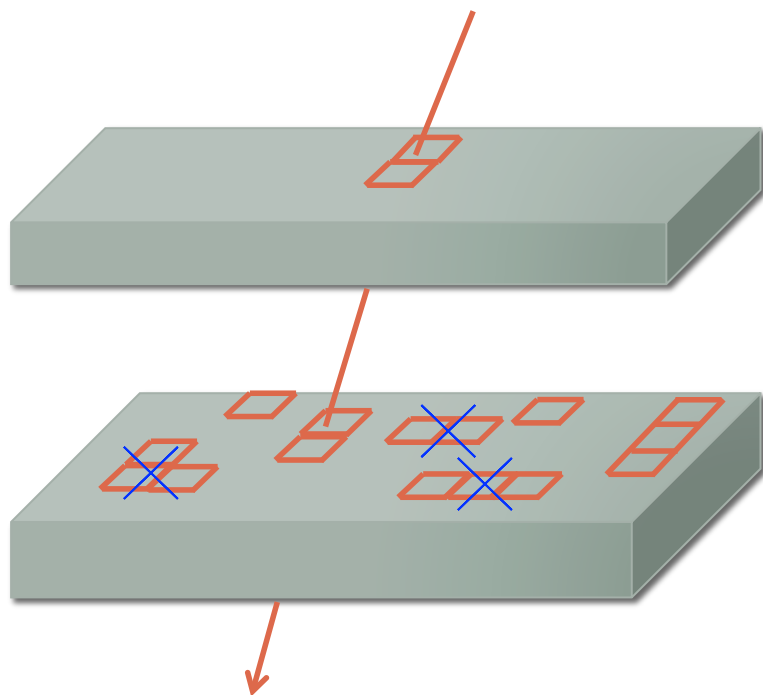
フィッターから得られるトラックパラメターから決定

使用フィッター: カルマンフィルター

クラスターの情報: 使用→外挿ミス削減

クラスターを用いた外挿処理

クラスターの形状を見ることで外挿ミスが起きる可能性を落とす



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

ttbar @ 350 GeV の トラッキングのCPU時間とメモリ

サンプル: ttbar 350 GeV/c + ペアBG

FPCCDTrackFinder を使用する

- CPU時間
 - ~ 3 時間 / event
 - トラックシード生成処理が多くのCPU時間を必要とする
Track seed : Extrapolation = 5 : 1
- メモリ
 - ~ 3.5 GB / event