VXDTF2 – release 01: training, Sector Map, performance

FACE TO FACE TRACKING MEETING

5 DECEMBER 2017, PISA

Valerio Bertacchi,

Eugenio Paoloni,

Thomas Lück

Outline

- Training Sample (TS)
  - New training sample
  - Selection procedure – NoKick Cuts
  - Performance of the Selection
- SVD-Only tracking performances
- Fake Rate: definition issues
- How does the code work?

The new Training Sample for the VXDTF2

The importance of being trained

- The TS must teach to the Sector Map (and so the the VXDTF2) the geometrical features of the tracks, like:
  - Spatial distribution
  - Curvature (momentum)
  - Interaction with matter (deviation from helix)
- Natural choice→large sample of $\Upsilon(4S)$

New release, new training

- From the performance of previous releases:
  - low statistic of $\Upsilon(4S)$ spectrum in high pt-region→ hight pt inefficiency of VXDTF2
  - The time and the memory consumption are ruled by low momenum tracks
- New <u>default TS</u>: $10^6\Upsilon(4S)$ events $+\ 10^6\ 10\mu^{\pm}$ events $p_{\text{flat}} \in [0.8\text{GeV/c}, 10\text{GeV/c}]$
- <u>NoKick TS</u>: applied on defualt TS a selection procedure to select only the «trackable» tracks

Strategy of the Selection: from default to NoKick

Motivation of the the selection

With the default TS all the prossible events are reproduced → Included in TS rare events →

- Wider Sector Map filters→ fakes
- Reconstructed not-usable patterns → lower fitting efficiency, waste of time

Strategy of the Selection: from default to NoKick

Motivation of the the selection

With the default TS all the prossible events are reproduced → Included in TS rare events →

- Wider Sector Map filters→ fakes
- Reconstructed not-usable patterns → lower fitting efficiency, waste of time

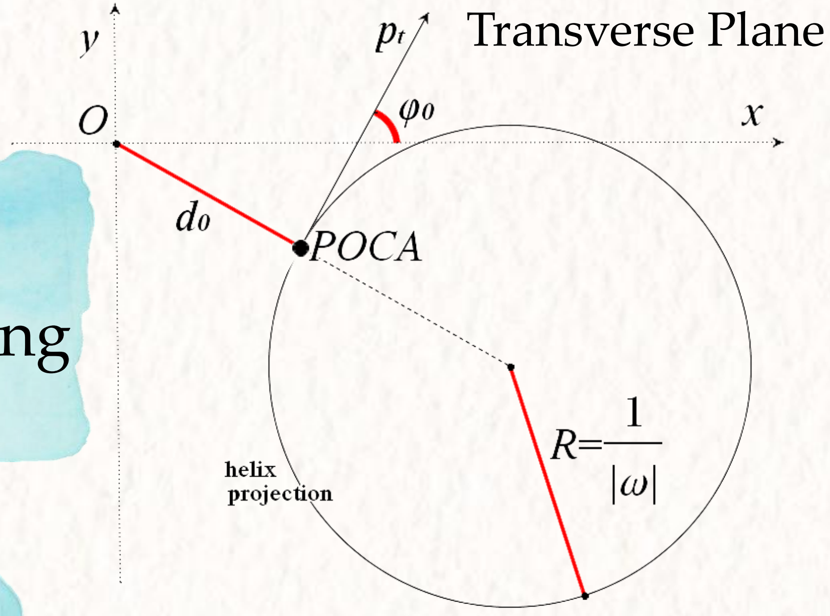
Purpose

Remove from TS rare events with non elicoidal tracks due to the material effects (multiple scatterng) a.k.a. catastrophic tracks

Strategy

- Phase A: define the list of cuts to select the TS
  - Automatic definition: teach to the VXDTF2 how to recognize the catastrophic tracks with a simulated sample
- Phase B: filter the default TS using the cuts defined in Phase A

Phase A, Step 1 – choice of variables



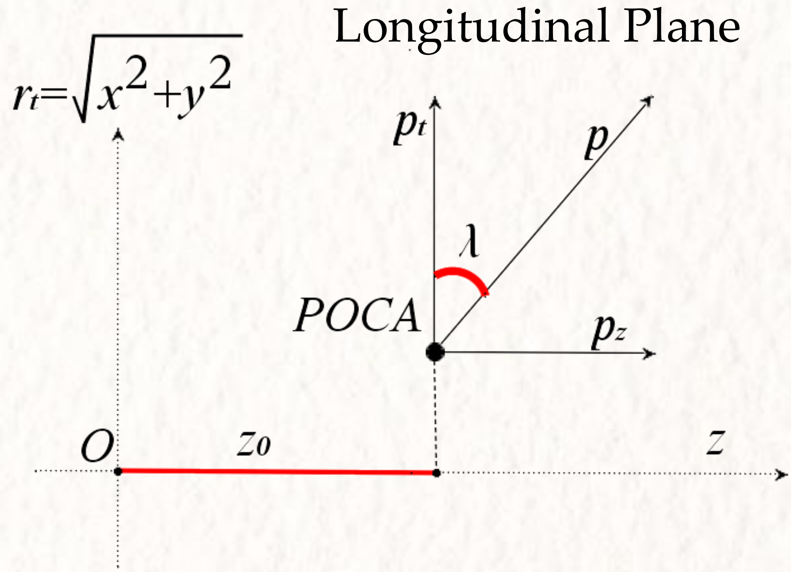
Parametrization of elices

- Parameters: $\omega, d_0, \varphi_0, z_0 \tan\lambda$ in function of **x,p**
- For each MC-hit **x,p** → tracks parameters hit-by-hit along the track

Identify Catastrophic Tracks

- Track-parameters for ideal helices constant along the track
- Strong puntual variation along the tracks → tag the anomalous interaction with the material
- ΔX = difference of X (track parameter) in VXD-layer crossing

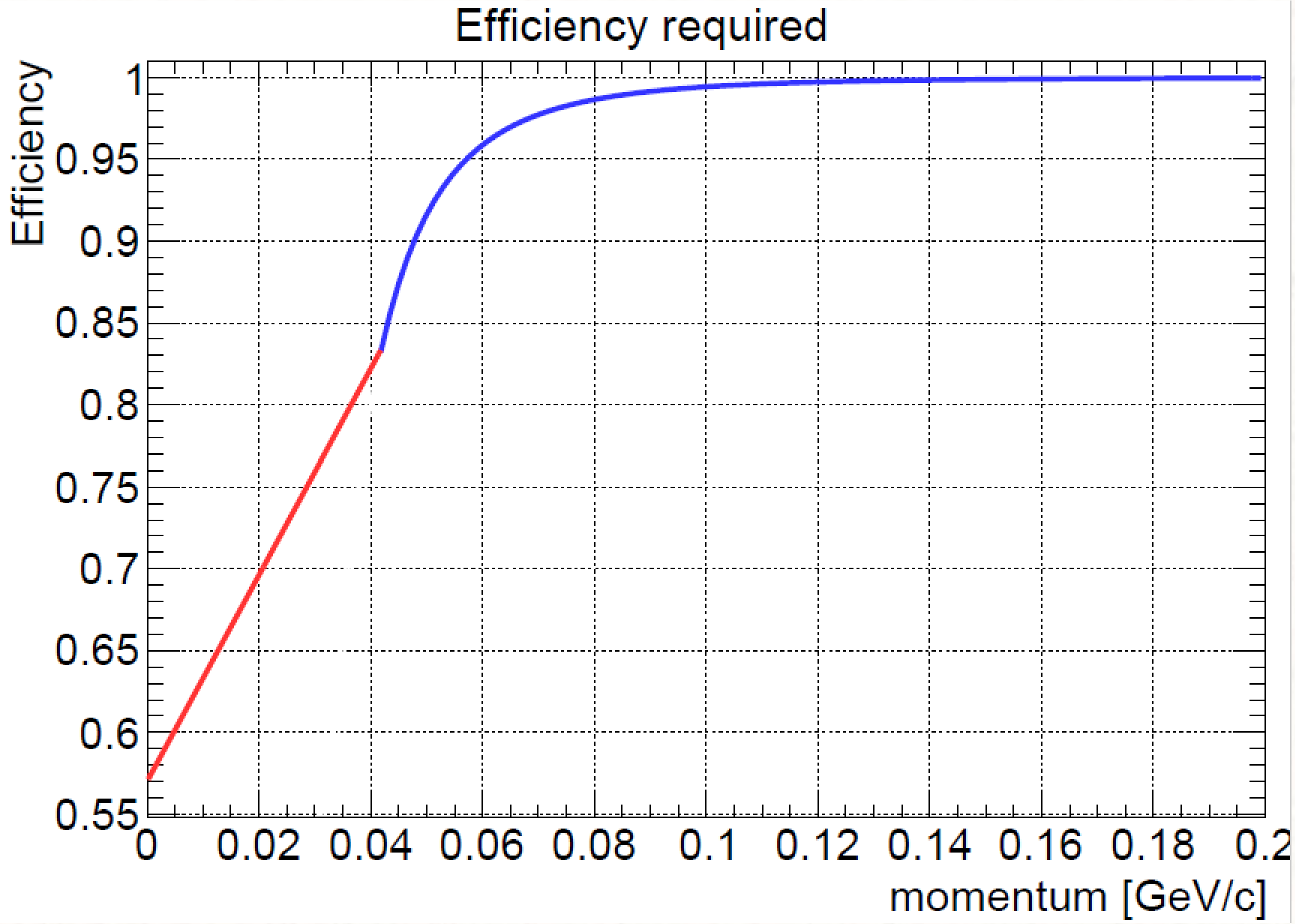
$$\Delta X(|\mathbf{p}|, \theta, \text{first layer}, \text{second layer})$$



Phase A, Step 2 – cut function

Required defined efficiency on single-cut:

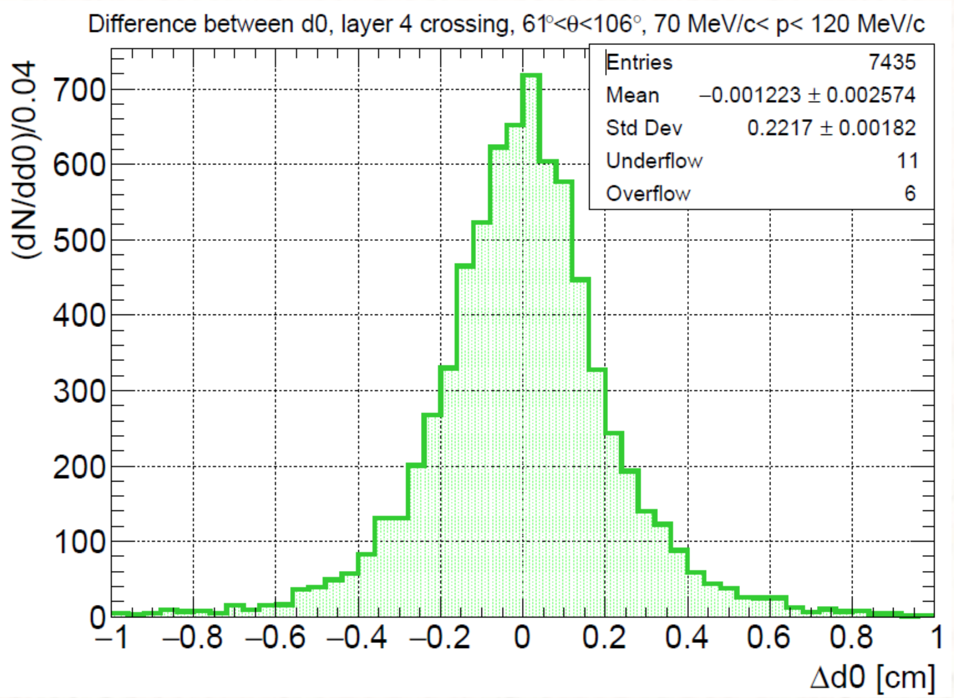
$$\varepsilon_{\mathrm{req}}(p) = -\frac{7.5 \cdot 10^{-7}}{p^{3.88}} + 1 \qquad 0.041 < p < 1\ \mathrm{GeV/c}$$



$\mathbf{p}$ [GeV/c]	ε	N_σ
0.2	0.999	3.3
0.1	0.995	2.8
0.07	0.98	2.3
0.05	0.9	1.6
0.04	0.84	1.4
0.03	0.75	1.1
0.02	0.7	1.0

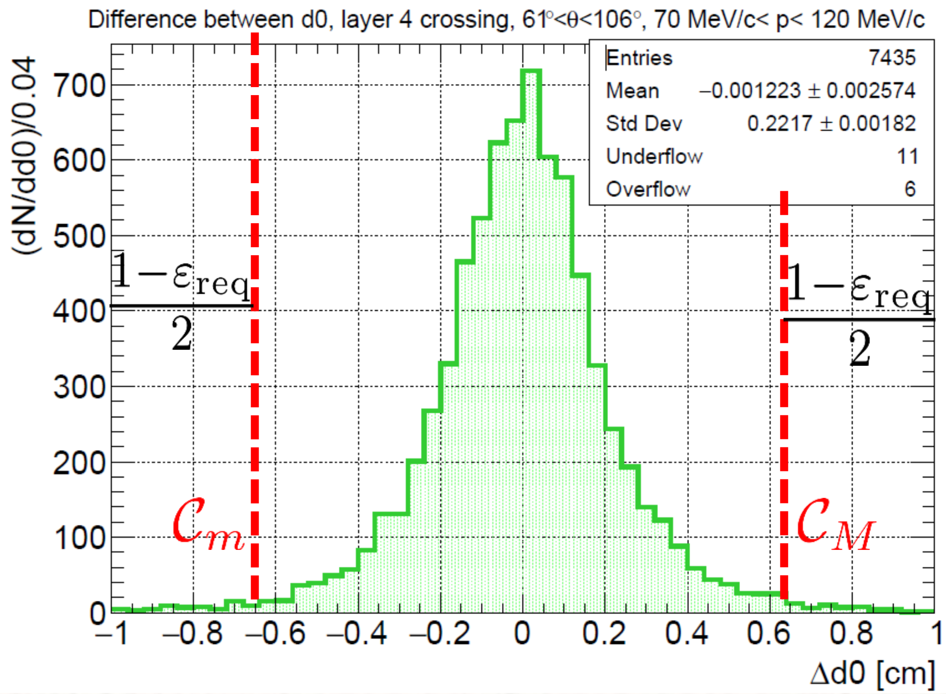
Phase A, Step 3 – cuts definition

1. Simulate a default TS-like sample ($\Upsilon(4S)$ muon enriched)
1. Required for each track at least 3 hit on 3 different SVD layer → used trackable tracks only
2. Evaluate distribution $\Delta X(|\mathbf{p}|, \theta, l_1, l_2)$



Phase A, Step 3 – cuts definition

1. Simulate a default TS-like sample $(\Upsilon(4S)$ muon enriched)
1. Required for each track at least 3 hit on 3 different SVD layer → used trackable tracks only
2. Evaluated distribution $\Delta X(|\mathbf{p}|, \theta, l_1, l_2)$
3. Defined for each ΔX distribuionthe threshold Max, min $\mathcal{C}_m, \mathcal{C}_M$

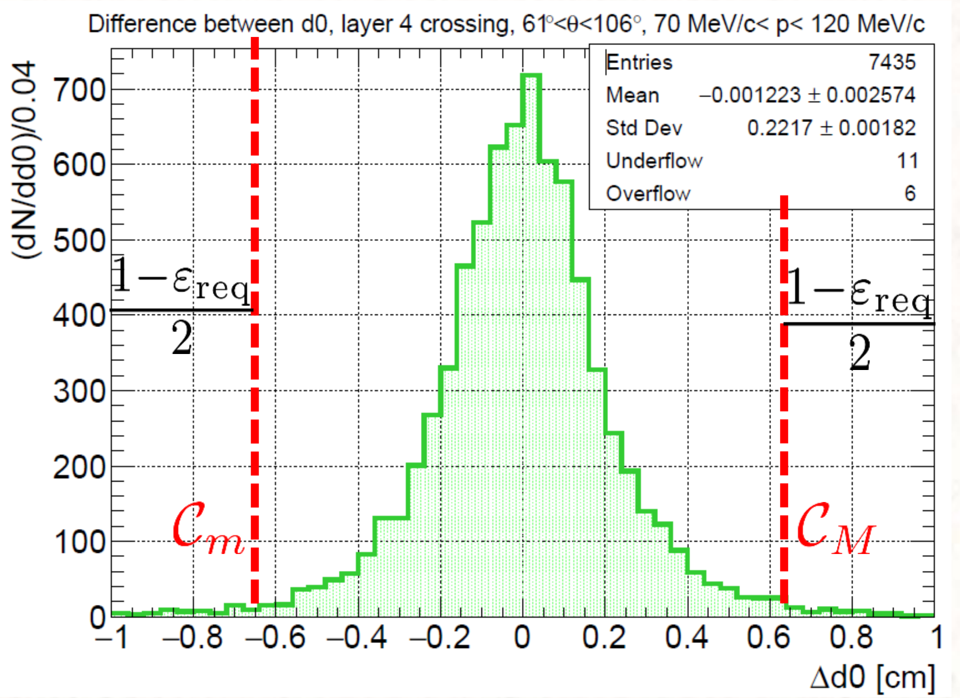


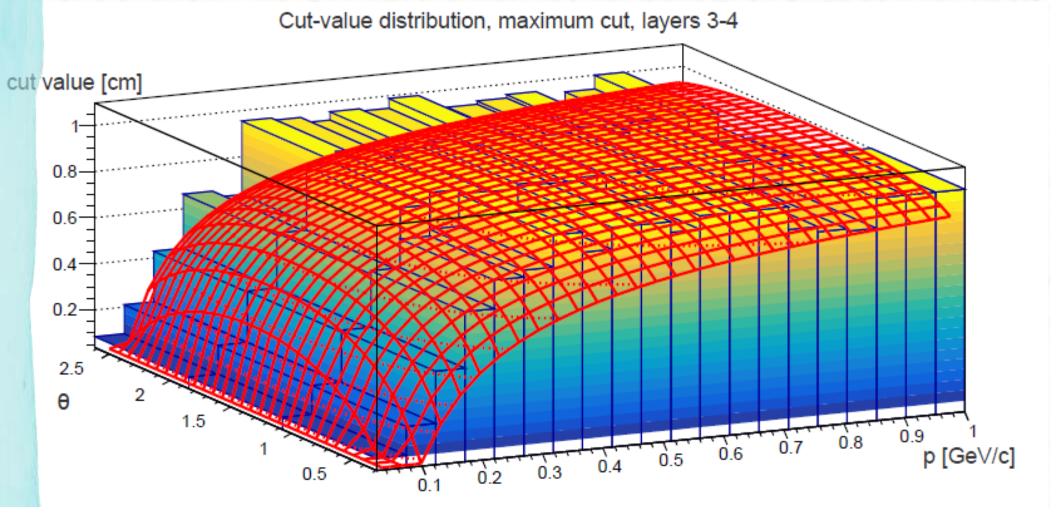
Phase A, Step 3 – cuts definition

1. Simulate a default TS-like sample ($\Upsilon(4S)$ muon enriched)
1. Required for each track at least 3 hit on 3 different SVD layer → used trackable tracks only
2. Evaluate distribution $\Delta X(|\mathbf{p}|, \theta, l_1, l_2)$
3. Defined for each ΔX distribuionthe threshold Max, min $\mathcal{C}_m, \mathcal{C}_M$
4. Build histogram $|\mathbf{p}| \times \theta$ with the values $\mathcal{C}_m, \mathcal{C}_M$ for each ΔX distribution
   - histograms limits cause controlled saturation effects: $d_0, z_0 : 1 \text{ cm}, \quad \phi_0, \tan\lambda : 0.3, \quad \omega : 0.4 \text{ cm}^{-1}$
5. Fit the $|\mathbf{p}| \times \theta$ with:

$$\Delta X(p, \theta, l_1, l_2) \sim \frac{A(X, l_1, l_2)}{(p)^{B(X, l_1, l_2)} \sqrt{\sin\theta}} + C(X, l_1, l_2)$$

6. Memorized the A,B,C parameters for pair of layer and each X parameter





Phase B– Cuts application

1. Each track of TS passed to a filter integrated in the training phase
2. Required for each track at least 3 hit on 3 different SVD layer
3. Evaluated $\Delta X(|\mathbf{p}|, \theta, l_1, l_2)$ for each pair of hit of each track (5 for each pair)
4. Starting from memorized A,B,C evaluated the actual cut:

$$\Delta X_{cut} = \frac{A}{(p)^B \sqrt{\sin\theta}} + C$$

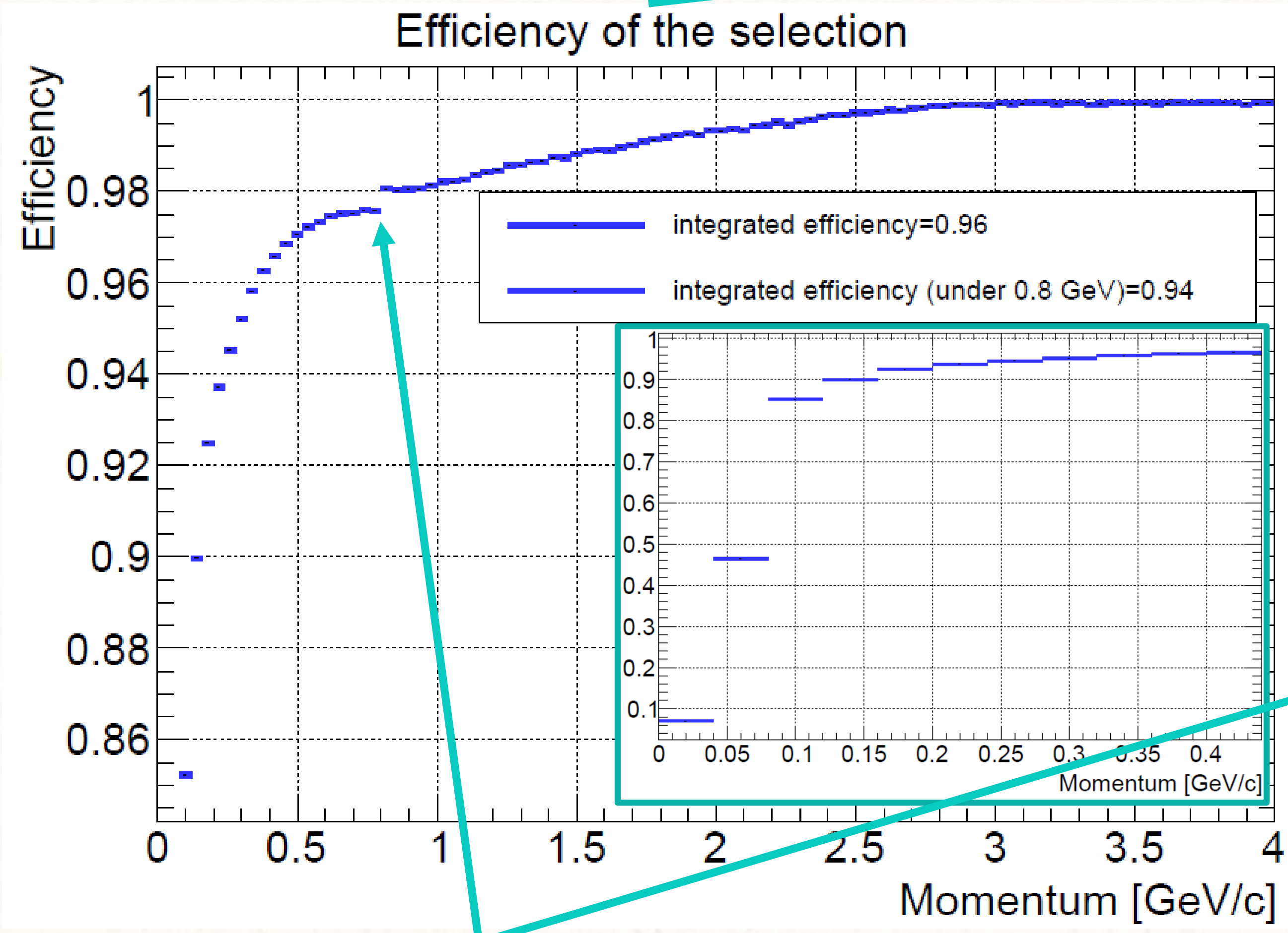
5. For each hit pair required $|\Delta X| < \Delta X_{cut}$
6. If all the requirement are overcomed the tracks is included in the NoKick TS

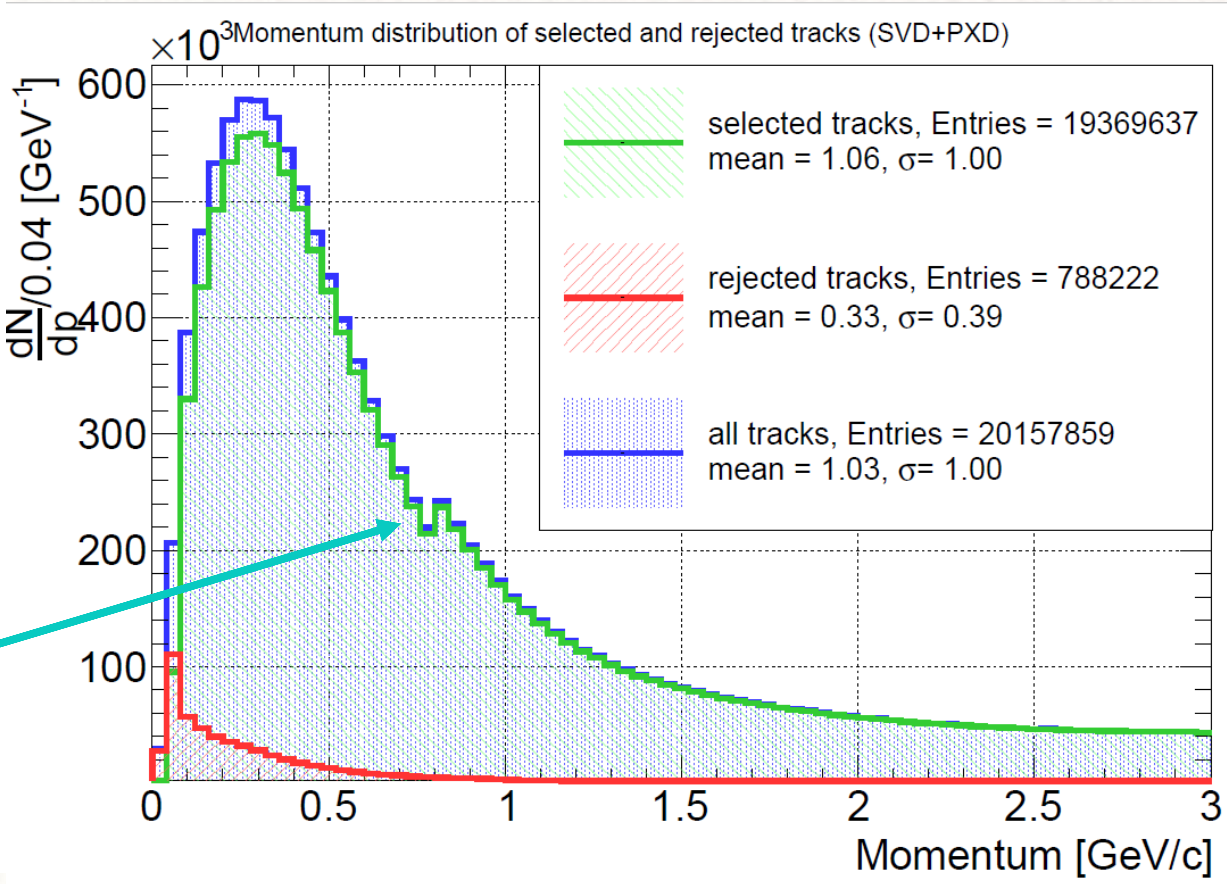
Differences with respect to release 00-09

- Removed global requirement (d0<1cm, z0<1cm)
  - New standard time-filter removed these tracks anyway
  - The time, memory & PR-burden is sustainable: these cuts can be applied offline (analysis level)
- Extended the range of the cuts to 10 GeV because of the new muon-enriched sample

Performance of the selection – TS level

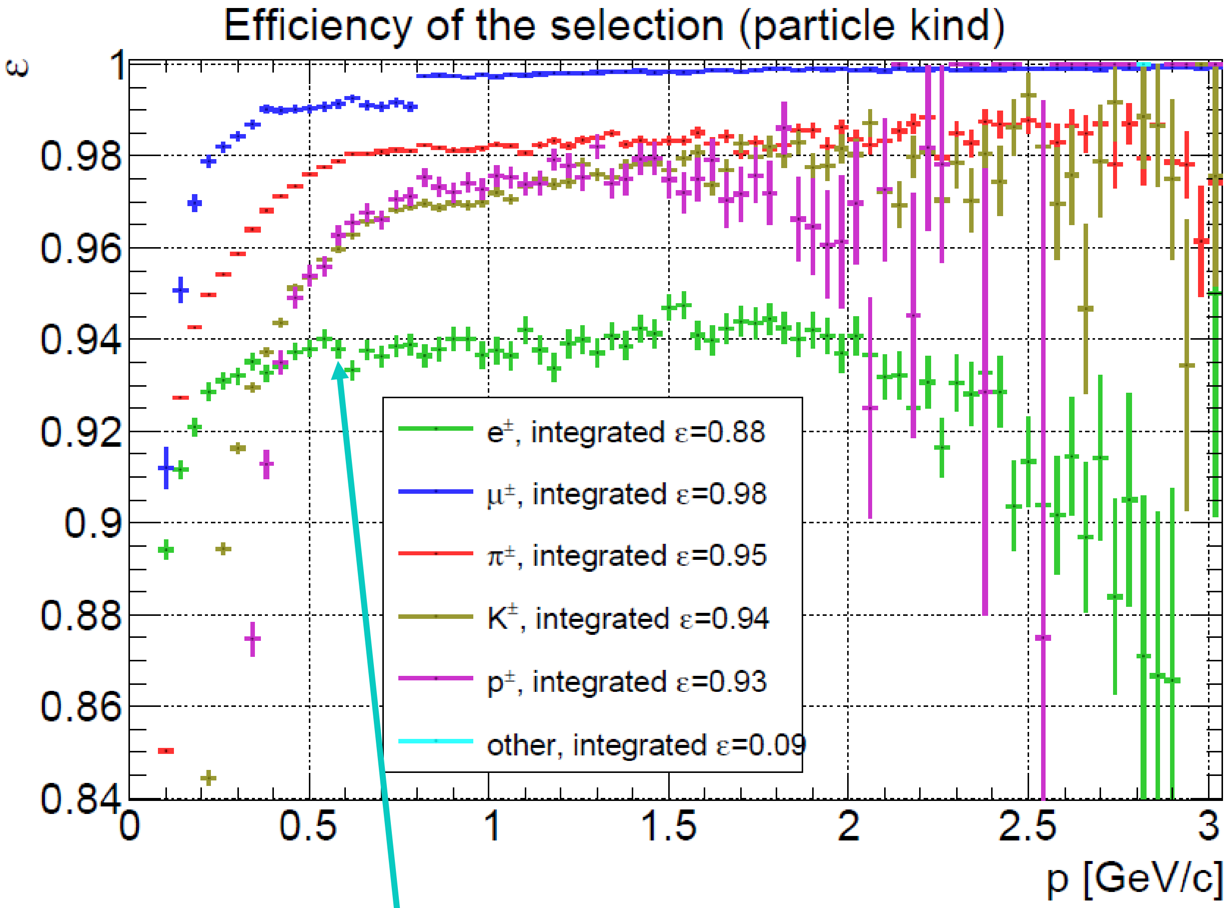
$$\varepsilon(p) = \frac{N_{\text{track pass}}(p)}{N_{\text{track tot}}(p)}$$

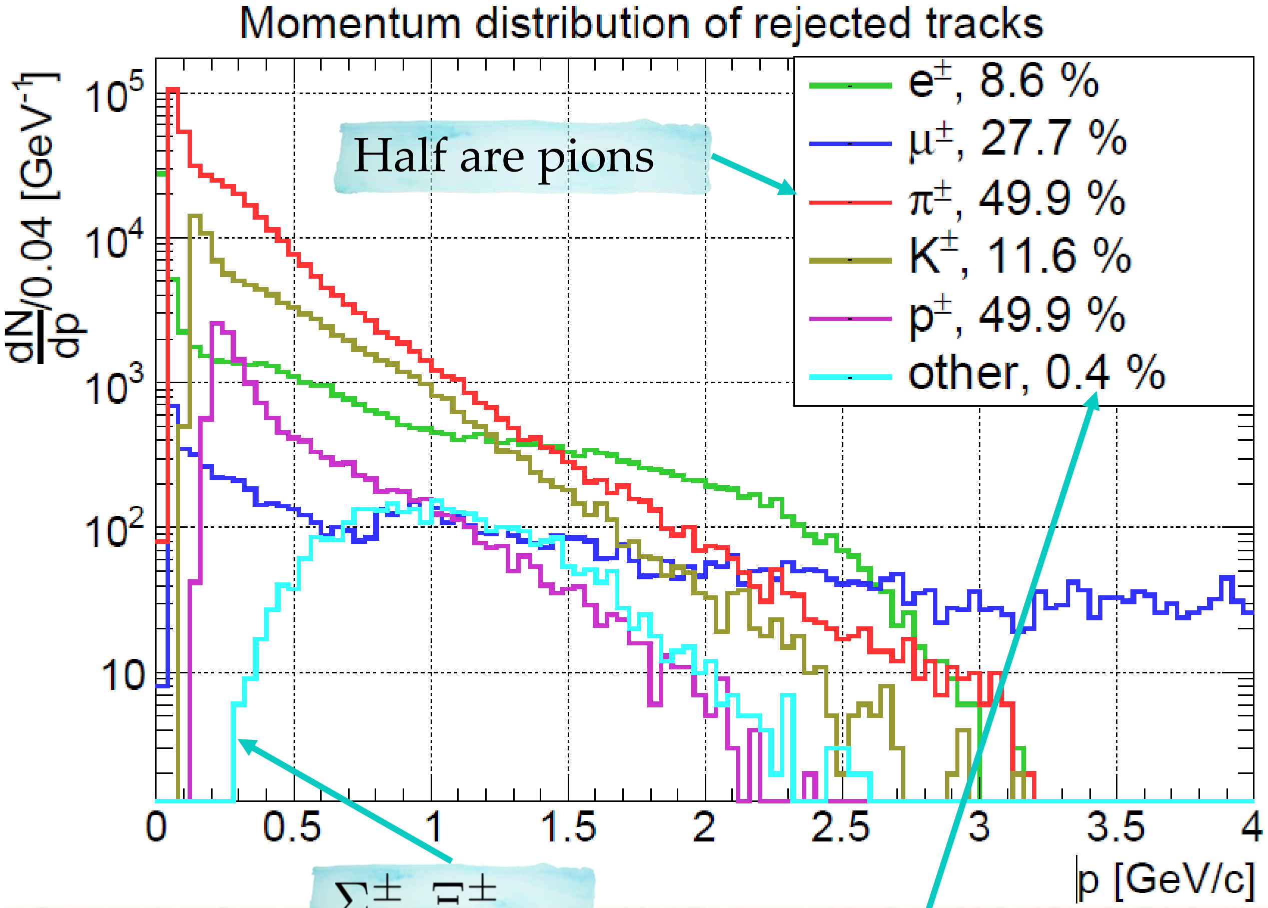




muon enrichment

Performance of the selection – TS level (2)





Low electron efficiency: maybe high-p bremsstrahlung?

% = fraction of the total number of rejected tracks

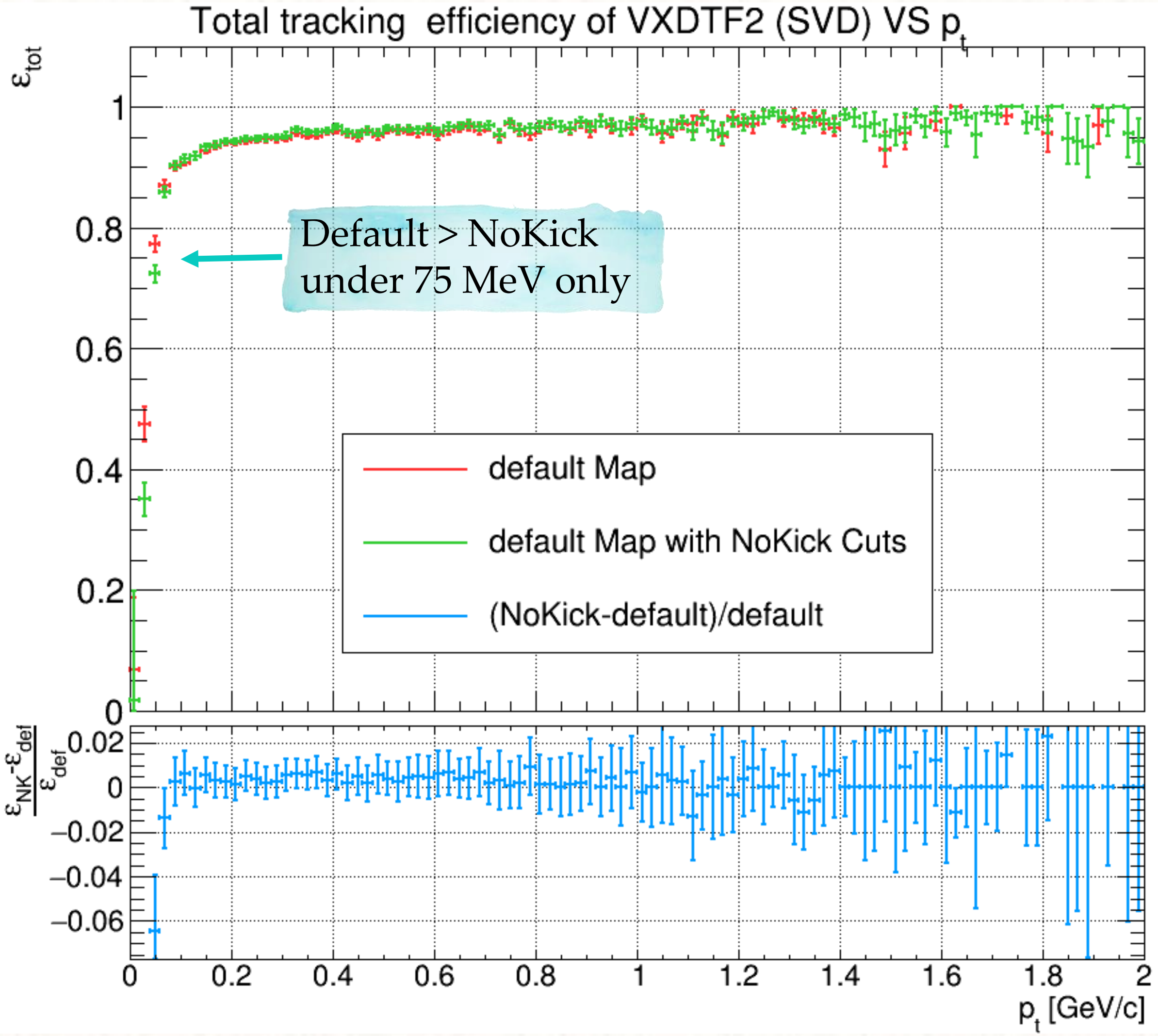
Tracking performances - efficiency

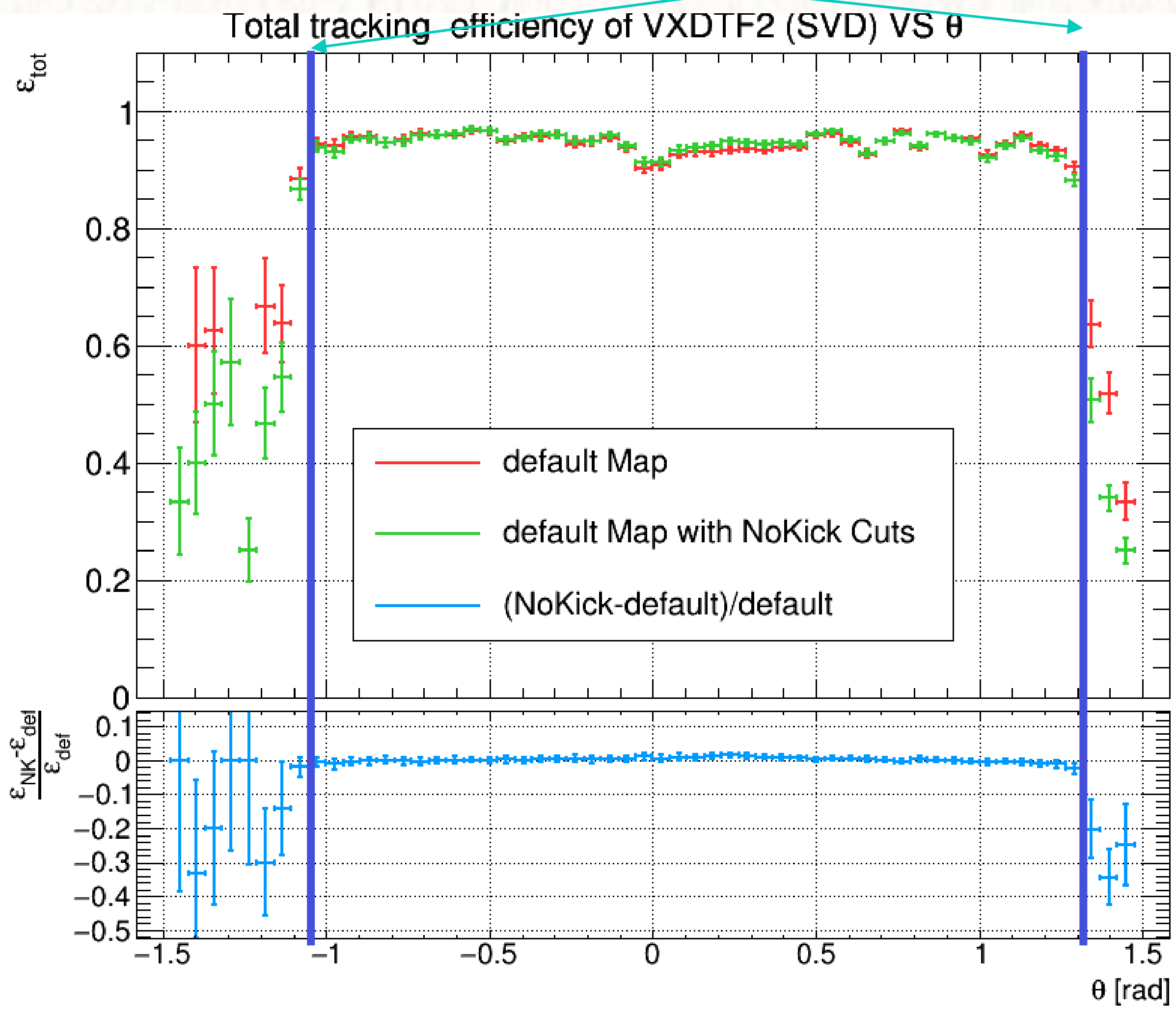
$$\varepsilon_{PR} = \frac{N_{\text{reco good}}}{N_{\text{tot}}} \qquad \varepsilon_{fit} = \frac{N_{\text{reco good + fit converged}}}{N_{\text{reco good}}} \qquad \varepsilon_{tot} = \varepsilon_{PR} \cdot \varepsilon_{fit}$$

- *good* = reco. with >2/3 hit on single MCParticle
- *tot* = in the accepance

Average NoKick = 95.2%   Average default = 95.1%

Acceptance limits SVD





Tracking performances – time & memory consumption

Validation Sample: 10k$\Upsilon(4S)$ + nominal bakground

	Pattern Recognition		Fitting	
	Time [ms/evt]	Memory [MB]	Time [ms/evt]	Memory [MB]
	SVD-only	SVD-only	SVD-only	SVD-only
Default Map (err)	7 (62)	97	151 (112)	58
NoKick Map (err)	5 (21)	6	149 (102)	75

DISCLAIMER

Fluctuation very high (100-1000%)!

Larger validation sample needed to perform accurate comparison!

Tracking performances – time & memory consumption

Validation Sample: 10k$\Upsilon(4S)$ + nominal bakground

	Pattern Recognition		Fitting	
	Time [ms/ev]	**Memory [MB]**	**Time [ms/ev]**	**Memory [MB]**
	SVD-only	SVD-only	SVD-only	SVD-only
Default Map (err)	**7** (62)	**97**	**151** (112)	**58**
NoKick Map (err)	**5** (21)	**6**	**149** (102)	**75**

Most relevant module:
TrackFinderVXDCellOMat
50% time
>90% memory

Most relevant Module:
TrackCreator
75% time
>60% memory

Tracking performances – time & memory consumption

Validation Sample: 10k $\Upsilon(4S)$ + nominal bakground

	Pattern Recognition		Fitting	
	Time [ms/ev]	**Memory [MB]**	**Time [ms/ev]**	**Memory [MB]**
	SVD-only	SVD-only	SVD-only	SVD-only
Default Map (err)	**7** (62)	**97**	**151** (112)	**58**
NoKick Map (err)	**5** (21)	**6**	**149** (102)	**75**

Without bkg

- PR time reduced of factor 2, but 100% error only
- Same for Fitting Time, but 50% error only
- Memory reduced (too high fluctuation)

Release 00-09 comparison

- In release 01 PR and Fitting time reduced of a factor 2-3
- Reduction with and without bkg

Fake rate: definition and performance

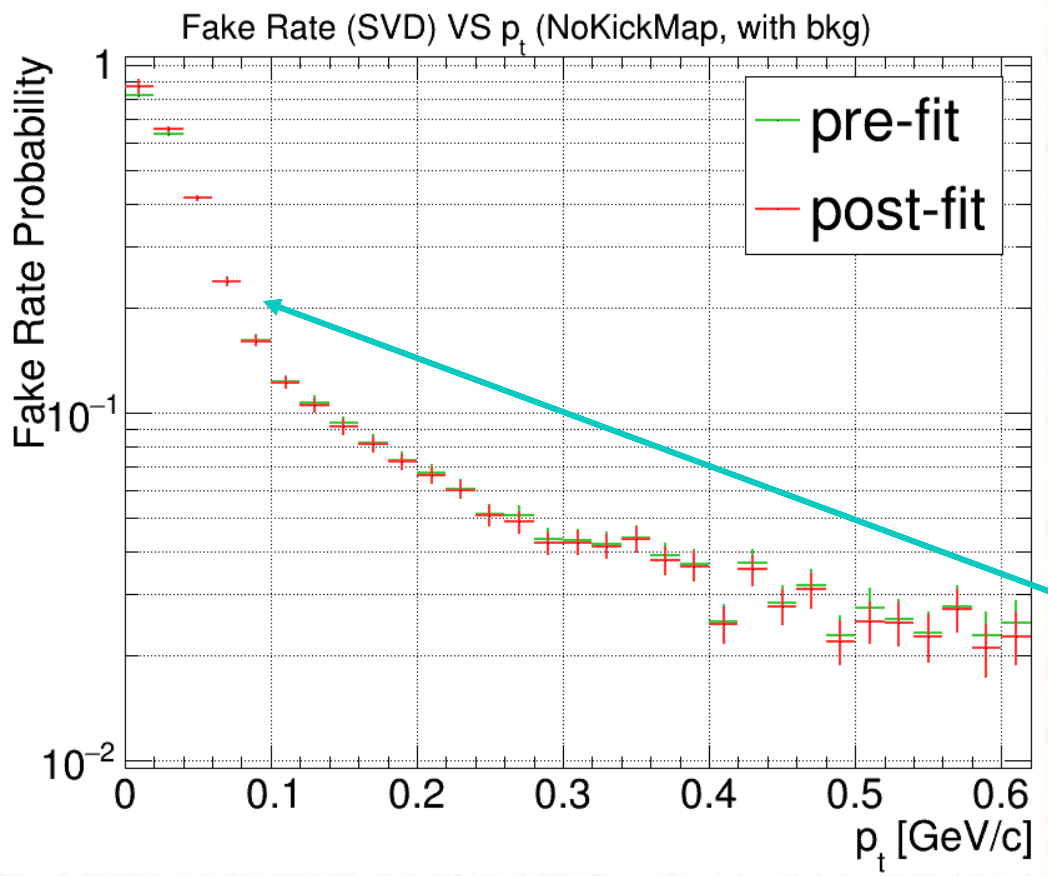
MC-Matching: best efficiency of the best purity track, over certain threshold

Purity>2/3
Efficiency>5%

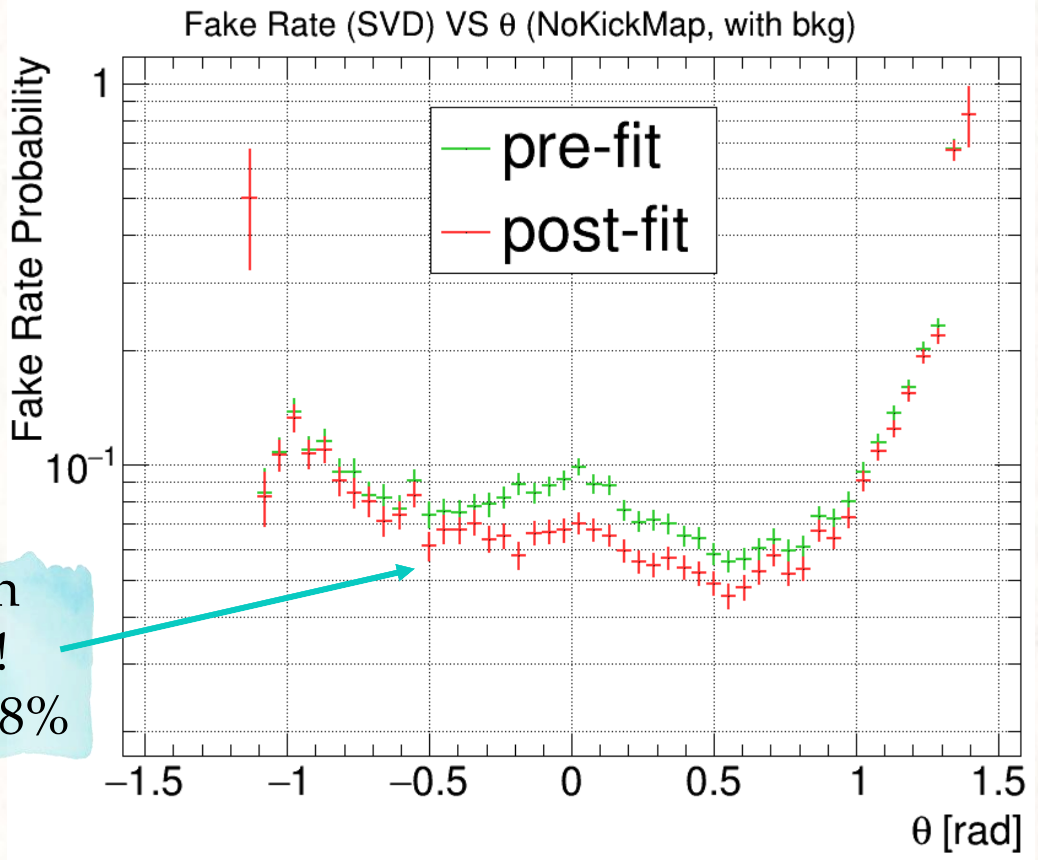
$$\frac{N_{\text{hit found}}}{N_{\text{hit MC}}}$$

$$\frac{N_{\text{hit on the same MCparticle}}}{N_{\text{hit found}}}$$

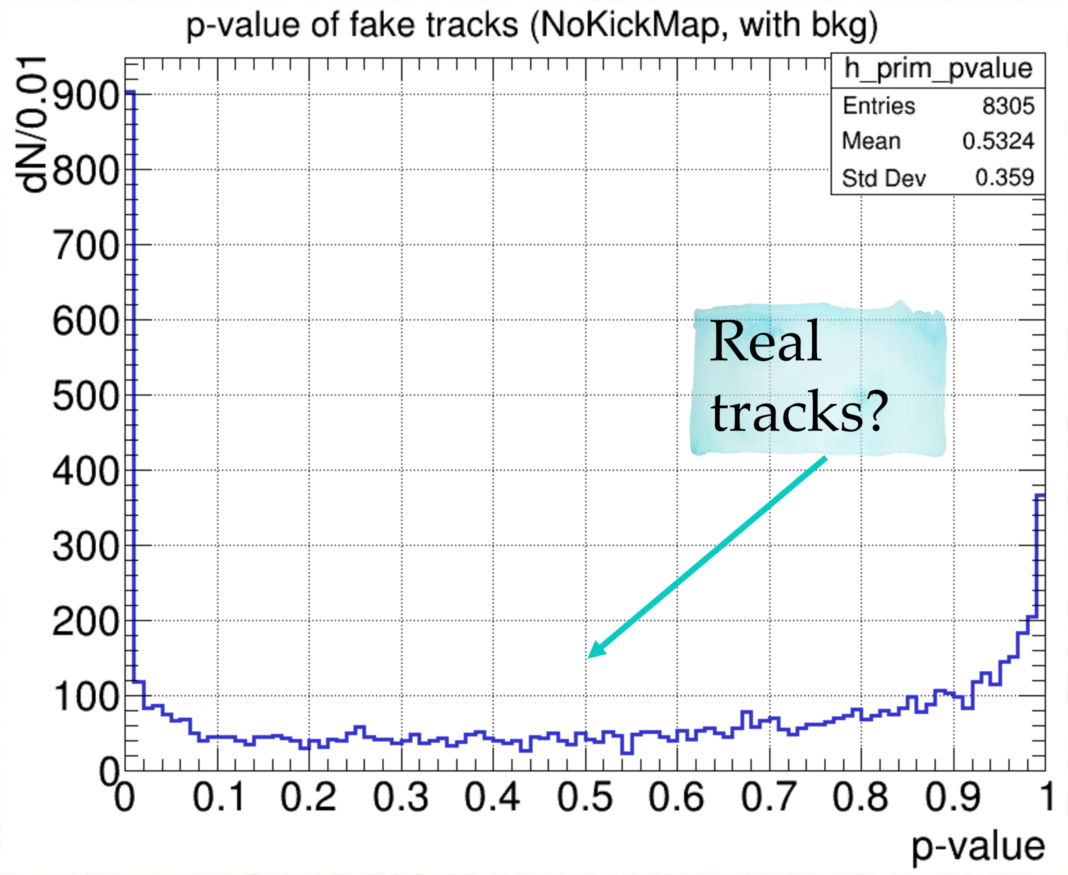
Fake: not-matched and not-cloned



Too high after fit!
Average ≈8%



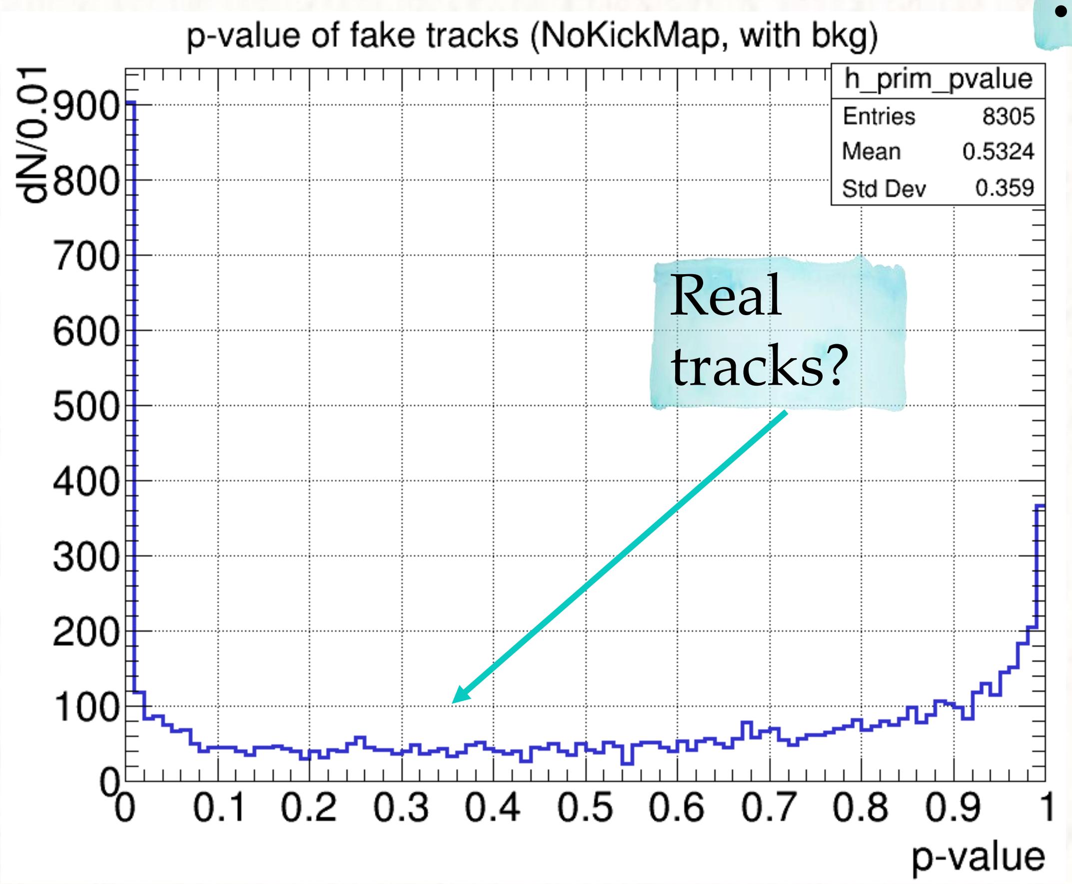
Fake rate: a definition problem?



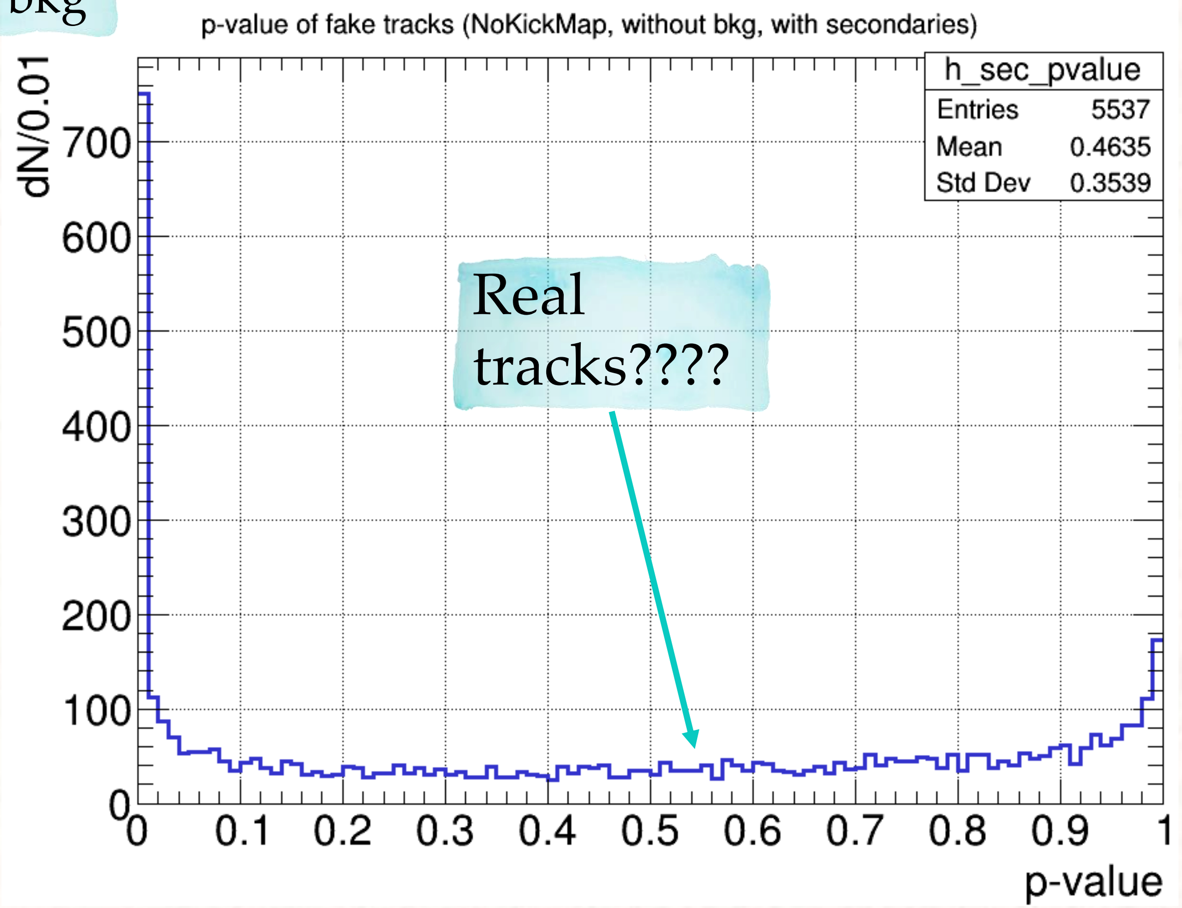
Fake rate: a definition problem?

VXDTF2 always find secondaries, we always need to include secondaries in normalization

- Included secondaries
- Removed bkg



Real tracks?

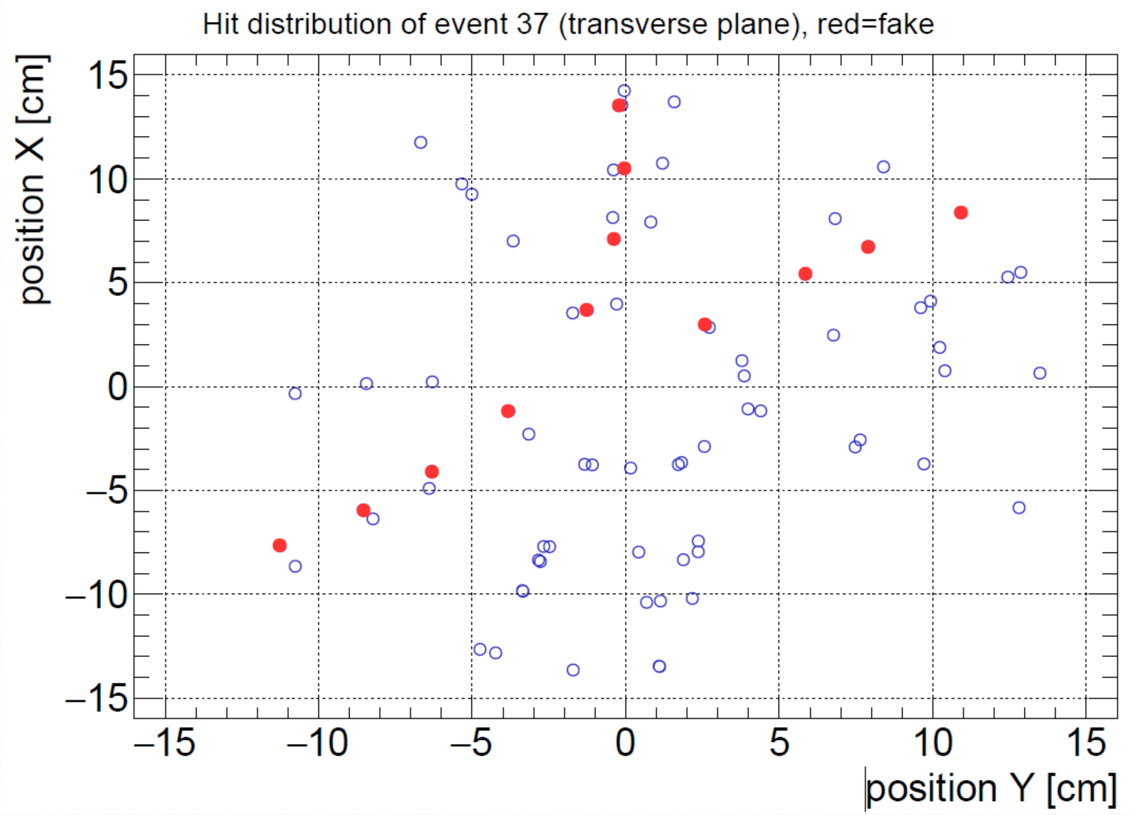


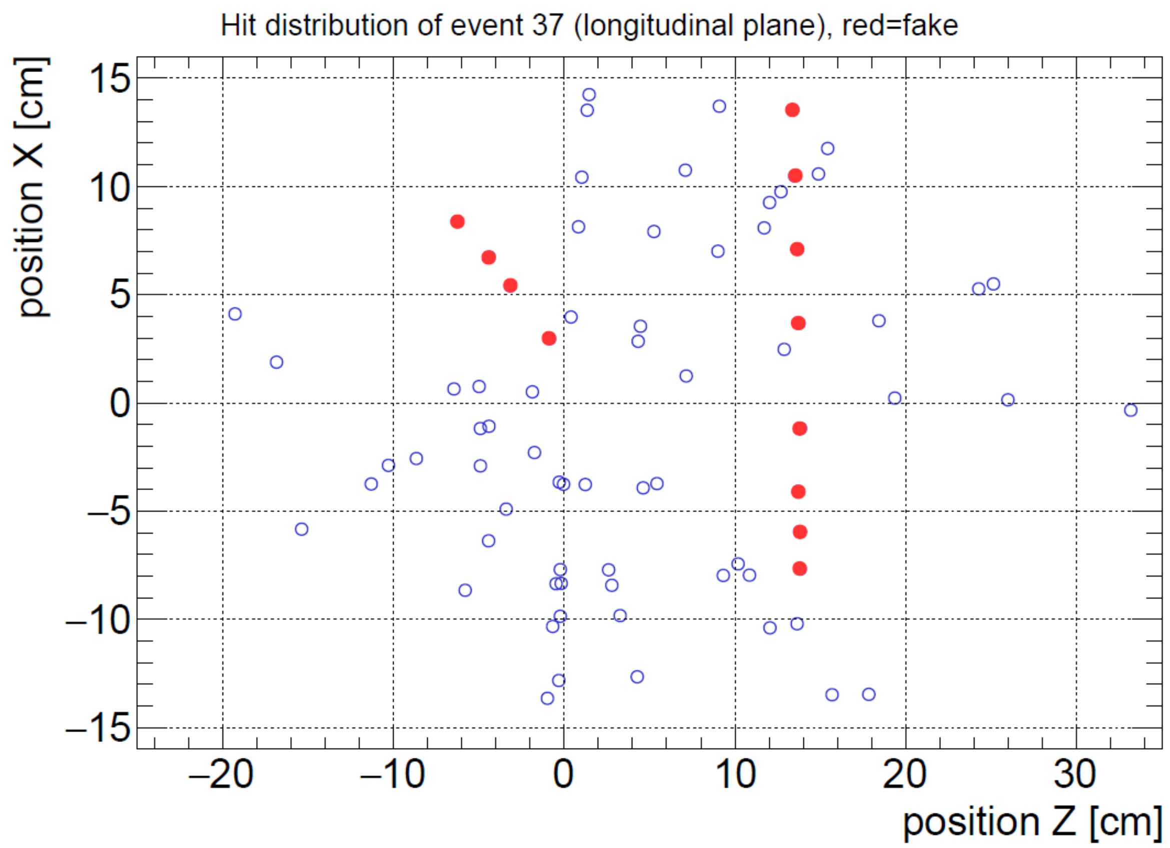
Real tracks????

Fake rate: are the loopers?

- Found an event with 11 matched tracks and 3 fake tracks
- 2 fakes came from the same MCParticle: a looper (80 MeV pion)

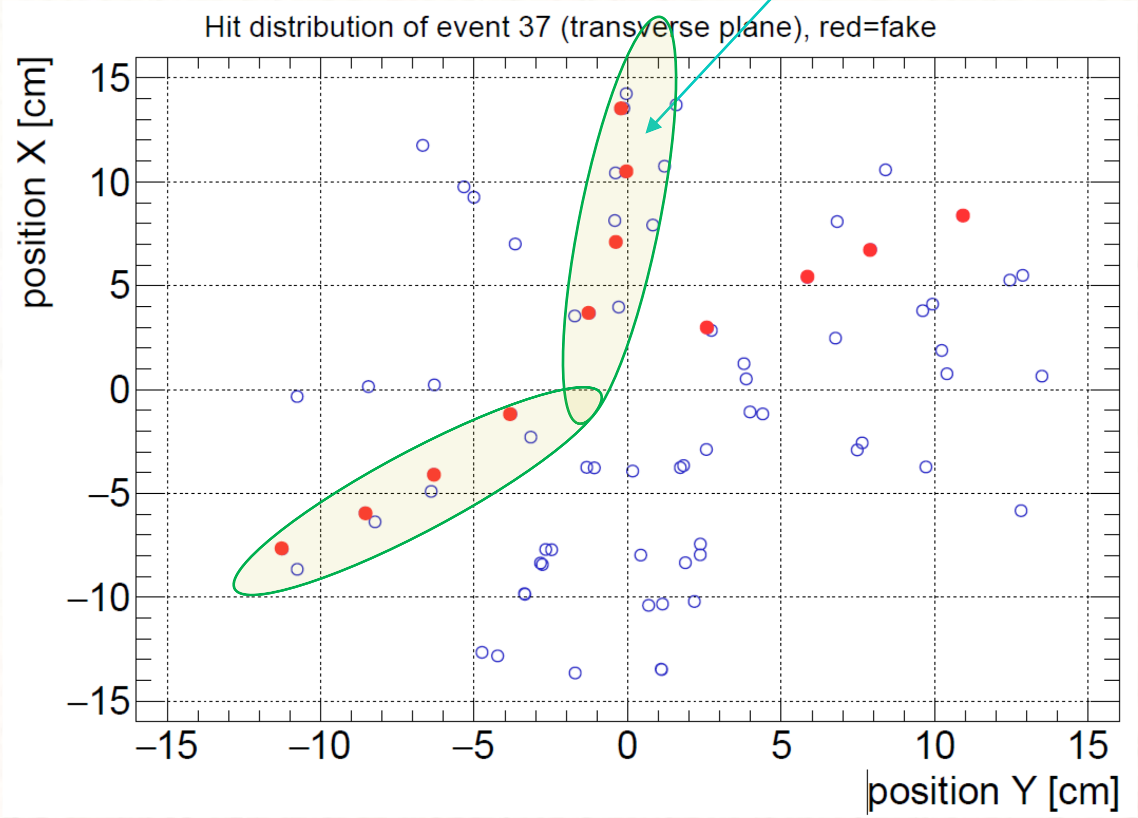
DISCLAIMER
Not sure about TrueHit - RecoTrack Matching (naive and ambiguous method!)

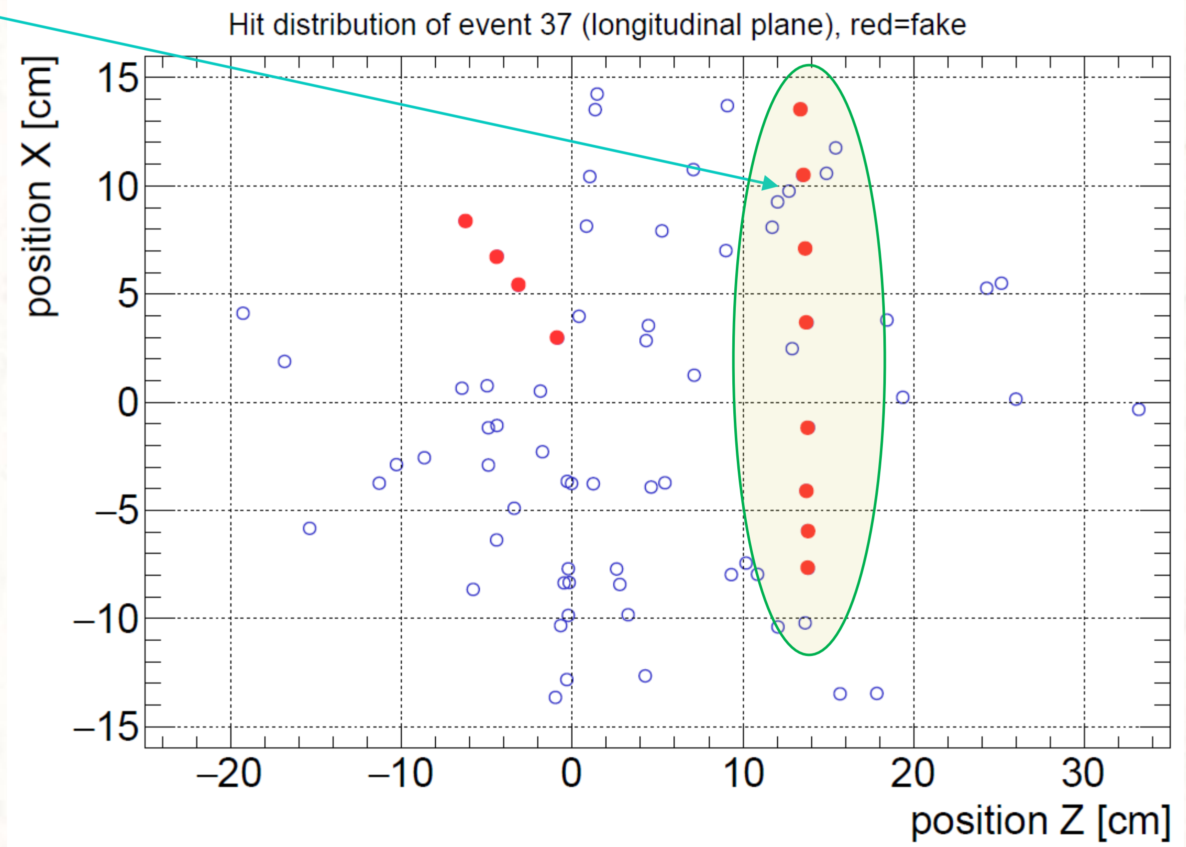




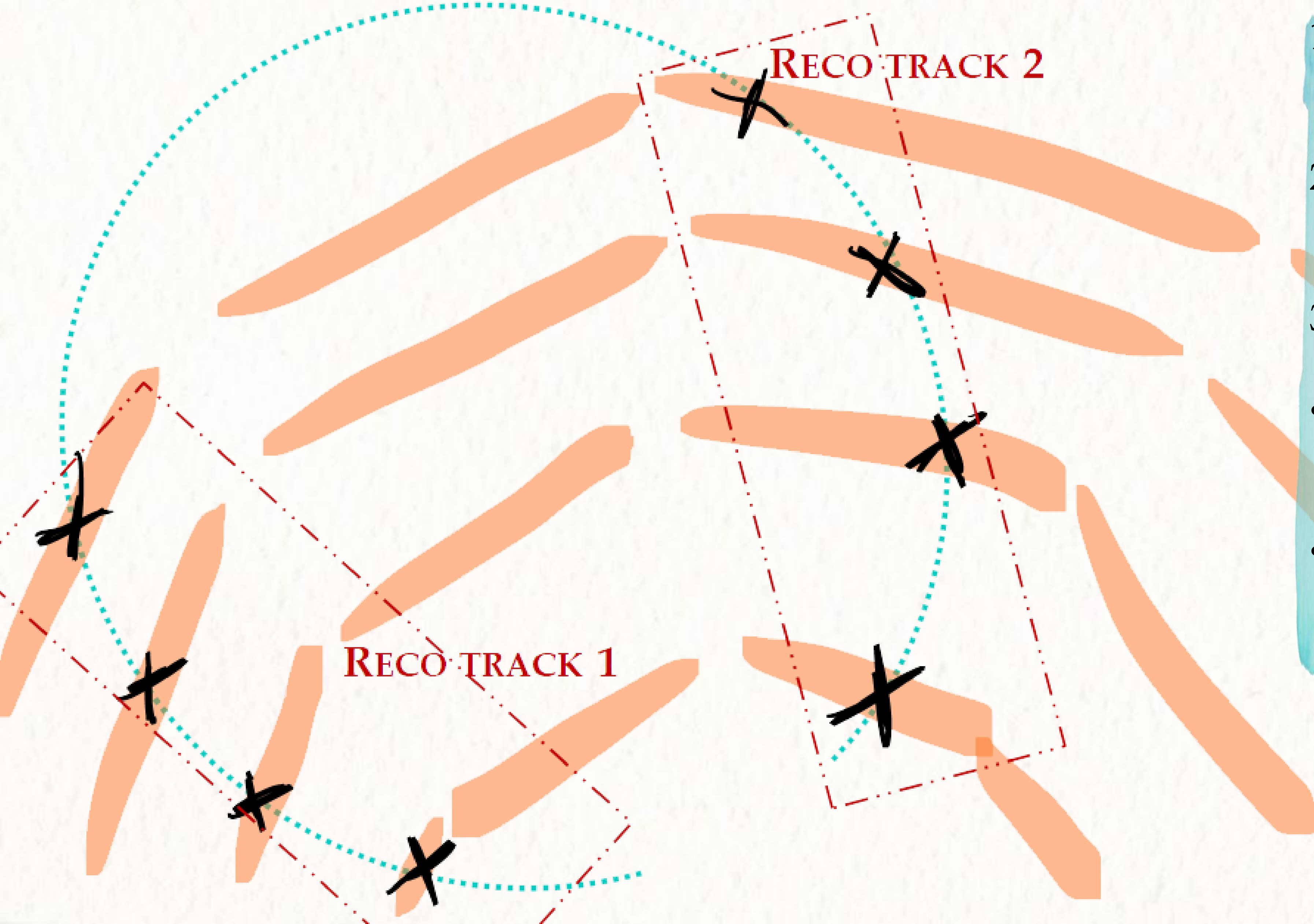
Fake rate: are the loopers?

- Found an event with 11 matched tracks and 3 fake tracks
- 2 fakes came from the same MCParticle: a looper (80 MeV pion)





Fake rate: possible explanation and proposal



1. Reconstructed 2 different tracks (one with wrong direction)
2. Two Reco Tracks do not overcome the matching requirements separately
3. Double counted as 2 fakes!

- Should the fakes be normalized to MCParticle, and not MCTracks?
- Can the CDC merging match 2 Reco Tracks and help to reduce the fakes?

Code structure – steering level (1)

How can I produce my own Sector Map?

Produce the list of cuts

- Usage: `basf2 NoKickCuts_evaluation.py -i inputfile.root -- (--useValidation) (--useFitMethod)`
- Output: `NoKickCuts.root`, which contains the list of A,B,C parameter needed to evaluate the cuts during the training.
- Options
  - The inputfile should the training sample(not mandatory, but the efficiency is optimized on it), or an equivalent input produced by `eventSimulation.py`.
  - `--useValidation` attaches to the output rootflie a detailed list of validation plots
  - `--useFitMethod` uses an alternative method to evaluate the cuts, fitting the ΔX distributions and extracting the width. It works, but it is not optimized.
- Location: `tracking/examples/VXDTF_Redesign/`
- Can I skip this step? Yes, there is a default `NoKickCuts.root` file in `tracking/data/`

Code structure – steering level (2)

Prepare the Training Sample

- Usage: `basf2 trainingPreparation.py -i inputfile.root – (--enable_selection)`
- Output: `SVDOnlyDefault_Belle2_SVDOnly.root`, or `SVDPXDDefault_Belle2_SVDOnly.root`
- Options
  - The inputfile must be produced with `eventSimulation.py`.
  - `--enableSelection` apply the TS selection using the file found at `tracking/data/NoKickCuts.root`. An additional output file (`TrackSelection_NoKick.root`) is also produced, it contains some validation plots of the selection.
  - `usePXD=True` (hardcoded option) enable the 6-layer tracking, and produce the second output
- Location: `tracking/examples/VXDTF_Redesign/`
- Can I skip this step? No, a default pre-prepared sample it is not provided

Code structure – steering level (3)

Train and produce the Sector Map

- Usage: `basf2 trainSecMap --train_sample inputfile.root --secmap outputfile.root`
- Output: `outputfile.root`
- Options
  - `--train_sample inputfile.root` must the output of TrainingPreparation.py
  - `--secmap outputfile.root` is the name of the output
- Location: `tracking/examples/VXDTF_Redesign/`
- Can I skip this step? Of course not, If you want to produce your own Sector Map

Code structure – developer level

How can I optimize and improve the TS selection procedure?

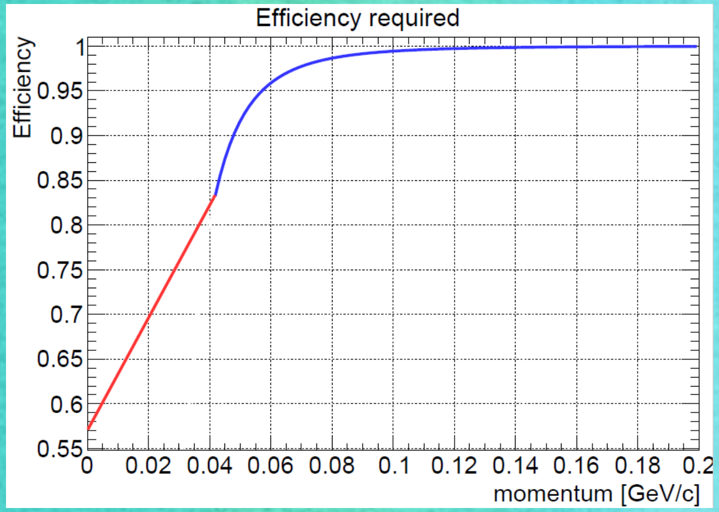
- Too complicated and technical to explain here
- I'm producing a Belle II – internal note where all the Training Sample selection procedure is described in detail
- A chapter is dedicated to the code structure in term of modules and classes involved
- **Laura Zani** (out of the tracking group…for now) will be the tester of the note, to understand if somebody else can continue easily my work

Conclusions

- The upgrades of release 01 improve the VXDTF2 performance in tem of
  - efficiency
  - time consumption
  - memory footprint
- The Selection of the TS improves the time & memory performances

NoKick Map has been chosen as default map for SVDOnly reconstruction

- The fake rate is probably overstimated because of definition problems



Next steps

- The NoKick selection has rooms of improvement
  - The cut-function can be optimized on a figure of merit
  - The fit-method can be optimized to avoid the use of the cut function
- The SVD+PXD map must be trained with NoKick selection
- The Phase 2 SVDOnly and SVD+PXD maps must be trained
- A strategy to reconstruct the loopers (both at MCTF level and at VXDTF2 level) may help to reduce the fake rate and gain efficiency
- The integration with CDC is beginning to be fundamental to improve the tracking performances

Thank you for attenction

BACKUP SLIDES

Track Parameters

$$\omega \to \frac{B_3 q}{\sqrt{P_1^2+P_2^2}} \qquad (1)$$

$$\tan\lambda \to \frac{P_3}{\sqrt{P_1^2+P_2^2}} \qquad (2)$$

$$d_0 \to \mathrm{sgn}\,(B_3 q)\left(\sqrt{\left(\frac{P_2}{B_3 q}+X_1\right)^2+\left(X_2-\frac{P_1}{B_3 q}\right)^2}-\sqrt{\frac{P_1^2+P_2^2}{B_3^2 q^2}}\right) \qquad (3)$$

$$\chi \to \tan^{-1}\left(\mathrm{sgn}\,(B_3 q)\left(\frac{P_1^2+P_2^2}{B_3 q}+P_2 X_1-P_1 X_2\right), (-P_1 X_1-P_2 X_2)\,\mathrm{sgn}\,(B_3 q)\right)$$

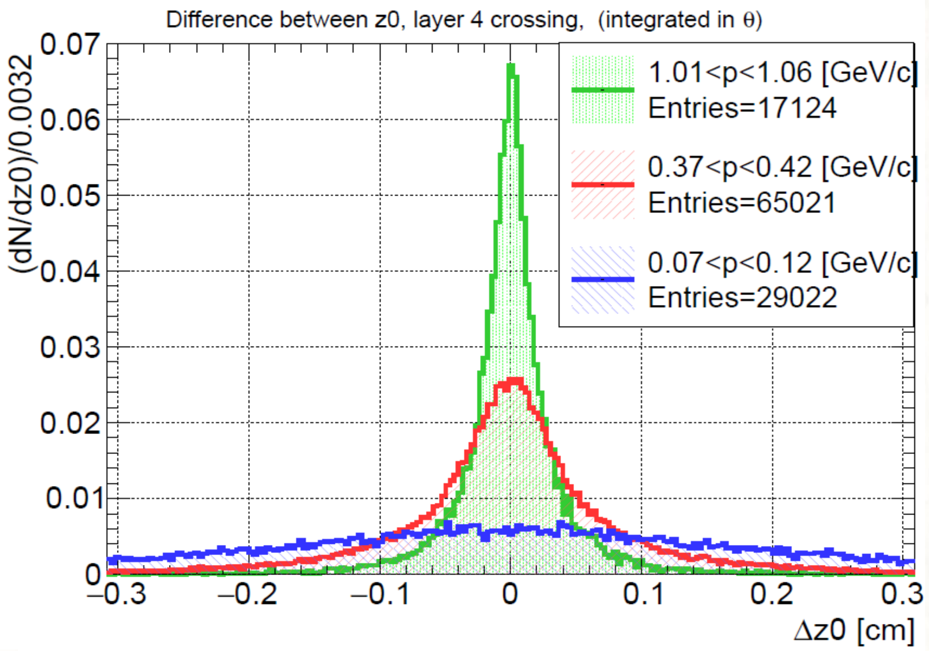
$$\varphi_0 \to \tan^{-1}(P_1, P_2)-\chi \qquad (4)$$

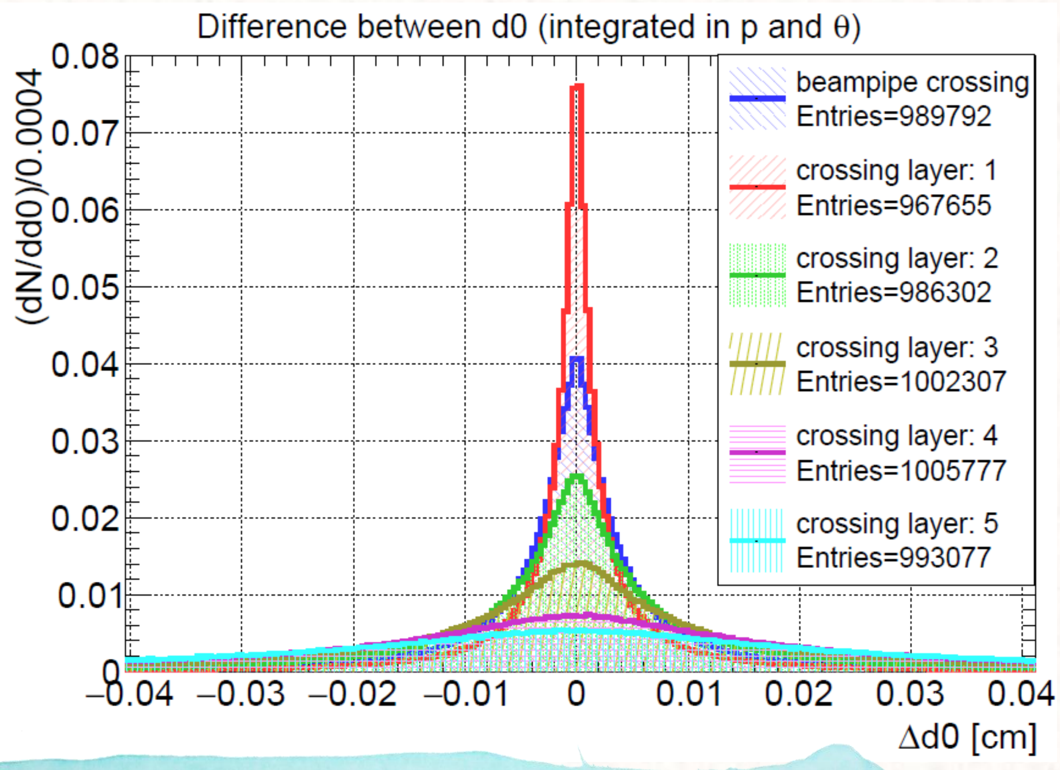
$$z_0 \to \frac{P_3 \chi}{B_3 q}+X_3 \qquad (5)$$

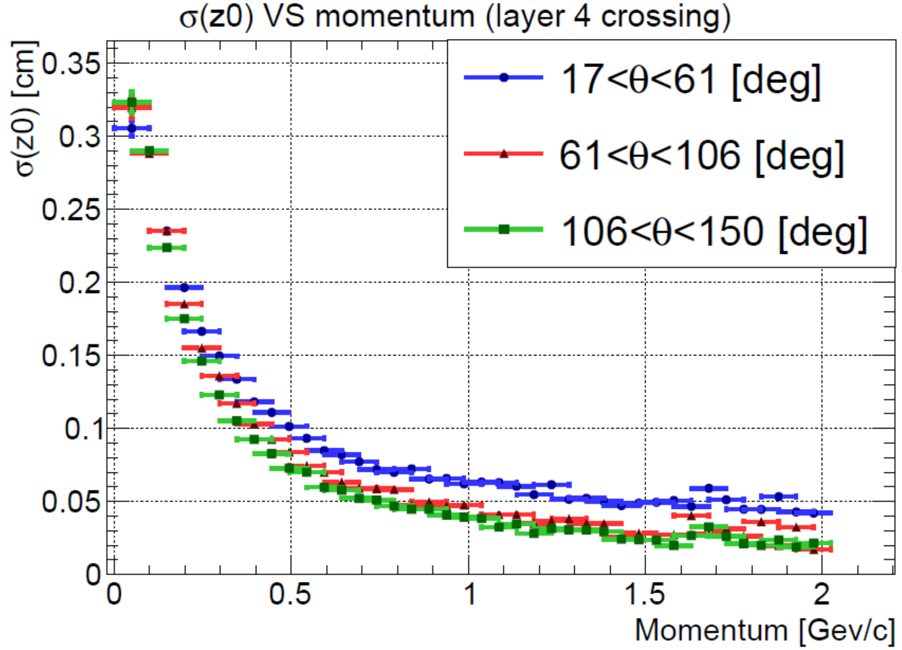
$$s \to -\frac{\sqrt{P_1^2+P_2^2}\,\chi}{B_3 q}$$

[Eugenio, Oliver, Tobi,
helices:the nitty-gritty of their Parametrization,
B2GM 2015]

Features of ΔX distributions



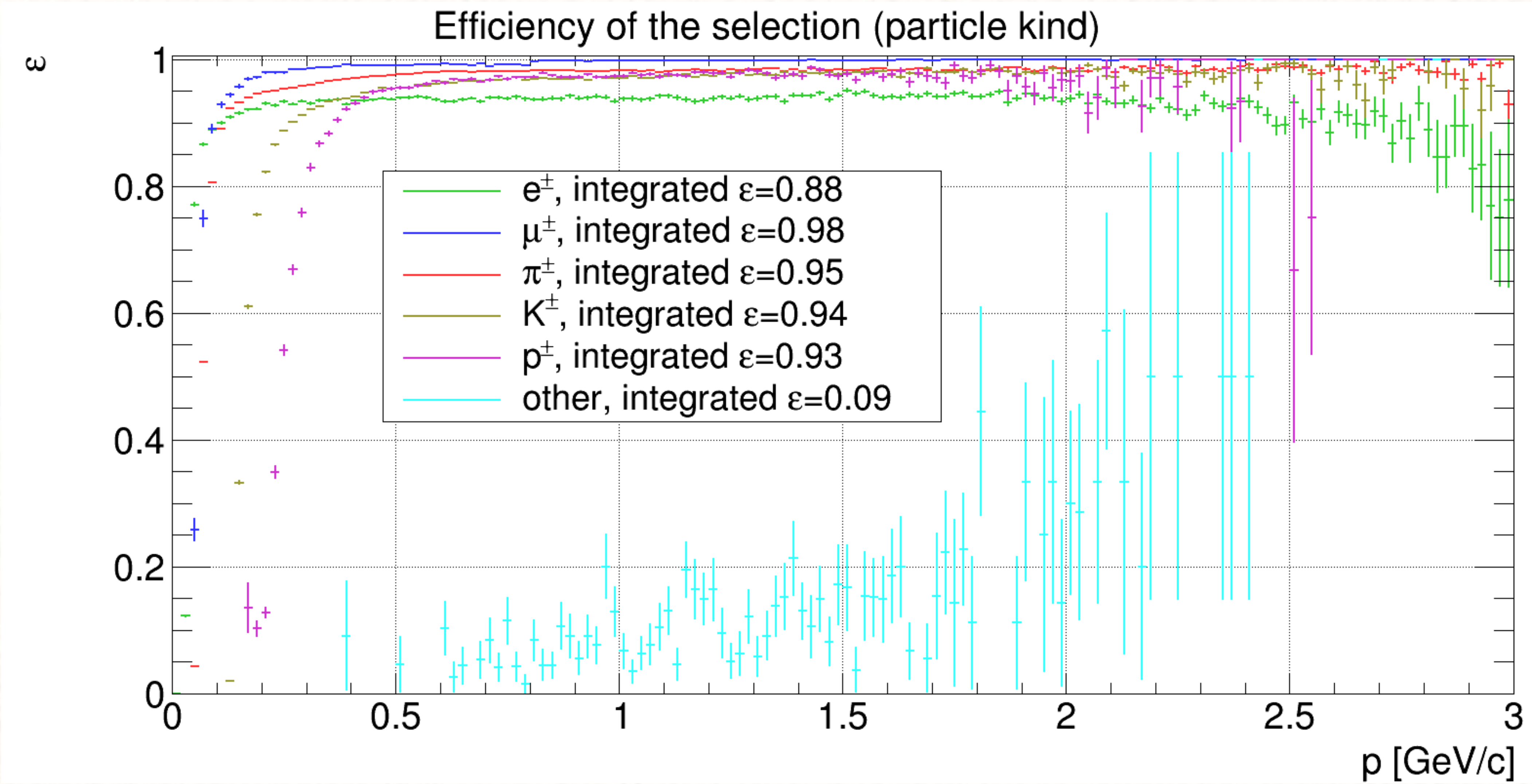




- $\Delta X(|\mathbf{p}|)$: enlarge at low p
- $\Delta X(\theta)$: about constant
- $\Delta X(l_1, l_2)$: proportional to the material, despite parameter-arm

coherent with multiple scattering

Performance of the selection – TS level



Performance of the selection – Sector Map Level

[OUTDATED]

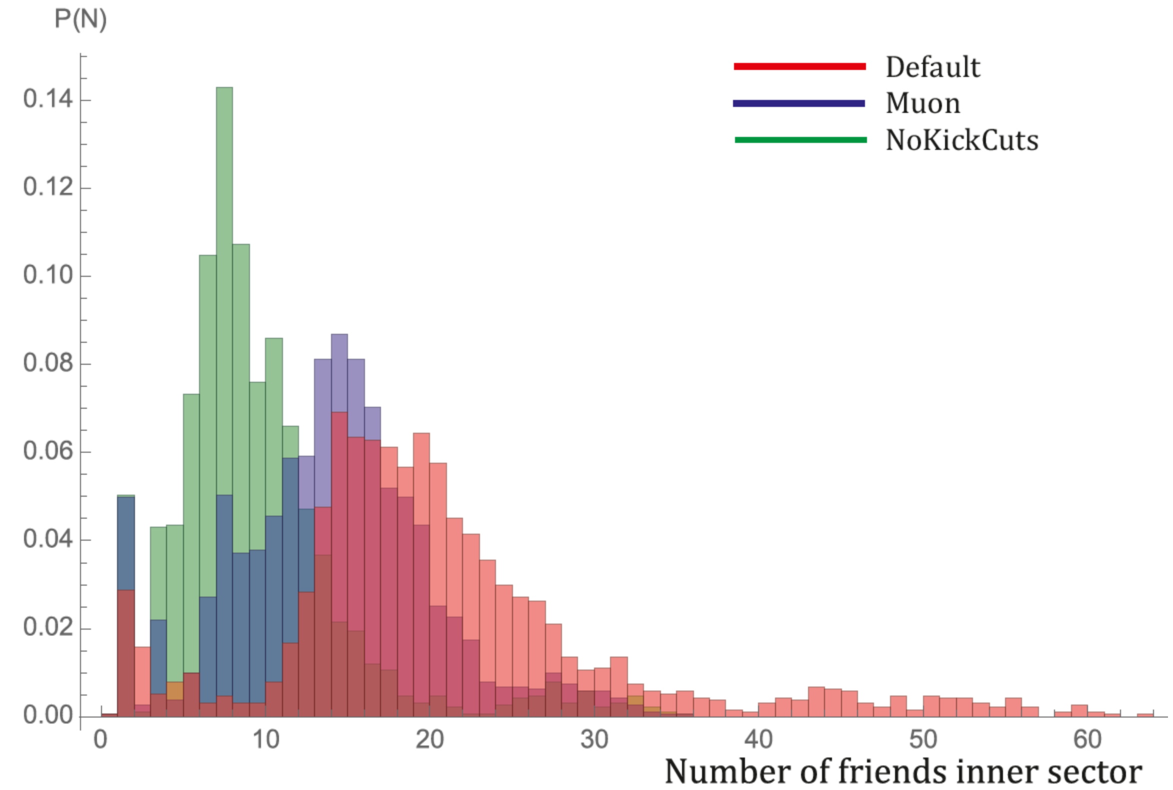
Complexity

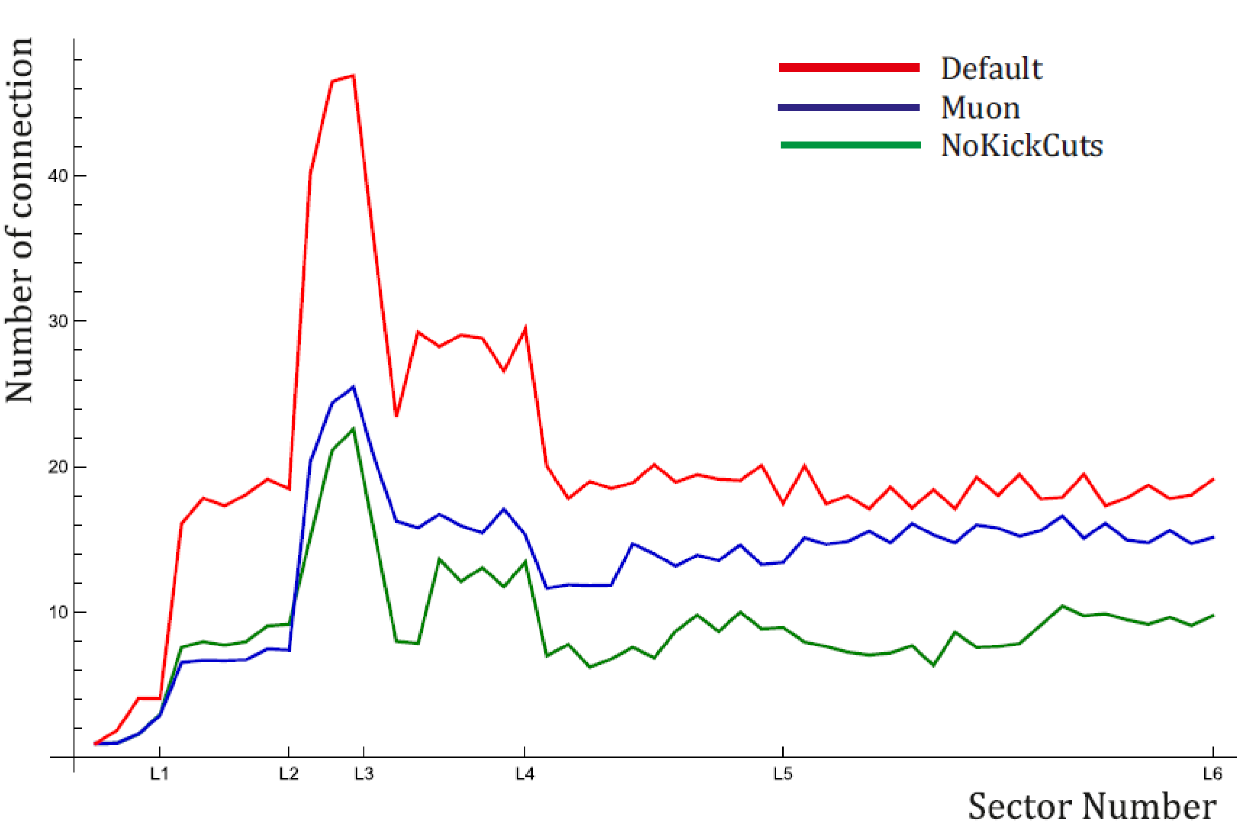
- Numero di relazioni d'amicizia:

$$N_F^{\mathrm{default}} = 38689,$$
$$N_F^{\mathrm{NoKick}} = 17545,$$
$$N_F^{\mu} = 25785.$$

Sector Map dimension

- Default (Fast/Full): 12.3 / 21.0 MB
- NoKick (Fast/Full): 5.1 / 8.3 MB
- Muon (Fast/Full): 6.8 / 14.0 MB





Time consumption and memory footprint (pattern recognition)

	default SECTOR MAP			NoKick SECTOR MAP		
DEFAULT SVD BKG PR	mean (ms)	error (ms)	MEMORY (MB)	mean (ms)	error (ms)	MEMORY (MB)
SVDSpacePointCreator	0.71	0.46		0.74	0.47	
SectorMapBootstrap	0	0		0	0	
SegmentNetworkProducer	2.19	20.47	1	1.76	10.58	1
TrackFinderVXDCellOMat	1.98	39.46	96	1.04	8.14	5
AddVXDTrackCandidateSubSets	0.07	0.03		0.08	0.03	
QualityEstimatorVXD	0.25	0.27		0.24	0.25	
BestVXDTrackCandidatesSelector	0	0.01		0	0.01	
SPTCvirtualIPRemover	0.02	0.01		0.02	0.01	
SVDOverlapResolver	1.14	1		1.1	0.92	
SPTCmomentumSeedRetriever	0.11	0.03		0.11	0.03	
SPTC2RTConverter	0.35	0.1		0.38	0.1	
TOTAL	6.82	61.84	97	5.47	20.54	6

Time consumption and memory footprint (fit)

default SECTOR MAP

DEFAULT SVD BKG FIT	mean (ms)	error (ms)	MEMORY (MB)
SetupGenfitExtrapolation	0	0	
DAFRecoFitter	36.13	39.3	16
TrackCreator	115.62	73.14	42
TOTAL	151.75	112.44	58

NoKick
SECTOR MAP

mean (ms)	error (ms)	MEMORY (MB)
0	0	
35.15	33.63	22
114.64	69.09	53
149.79	102.72	75

Also memory
fluctuates a lot!

Time consumption and memory footprint (rel9 Sector Map)

OLDDB MAP BKG PR	mean (ms)	error (ms)	MEMORY (MB)
SectorMapBootstrap	0	0	
SegmentNetworkProducer	3.62	18.19	
SegmentNetworkProducer	3.62	38.19	
TrackFinderVXDCellOMat	3.46	65.52	5
AddVXDTrackCandidateSubSets	0.12	0.04	180
QualityEstimatorVXD	0.3	0.31	
BestVXDTrackCandidatesSelector	0	0.02	
SPTCvirtualIPRemover	0.02	0.02	
SVDOverlapResolver	1.6	1.49	
SPTCmomentumSeedRetriever	0.13	0.04	
SPTC2RTConverter	0.45	0.13	
TOTAL	13.32	123.95	185

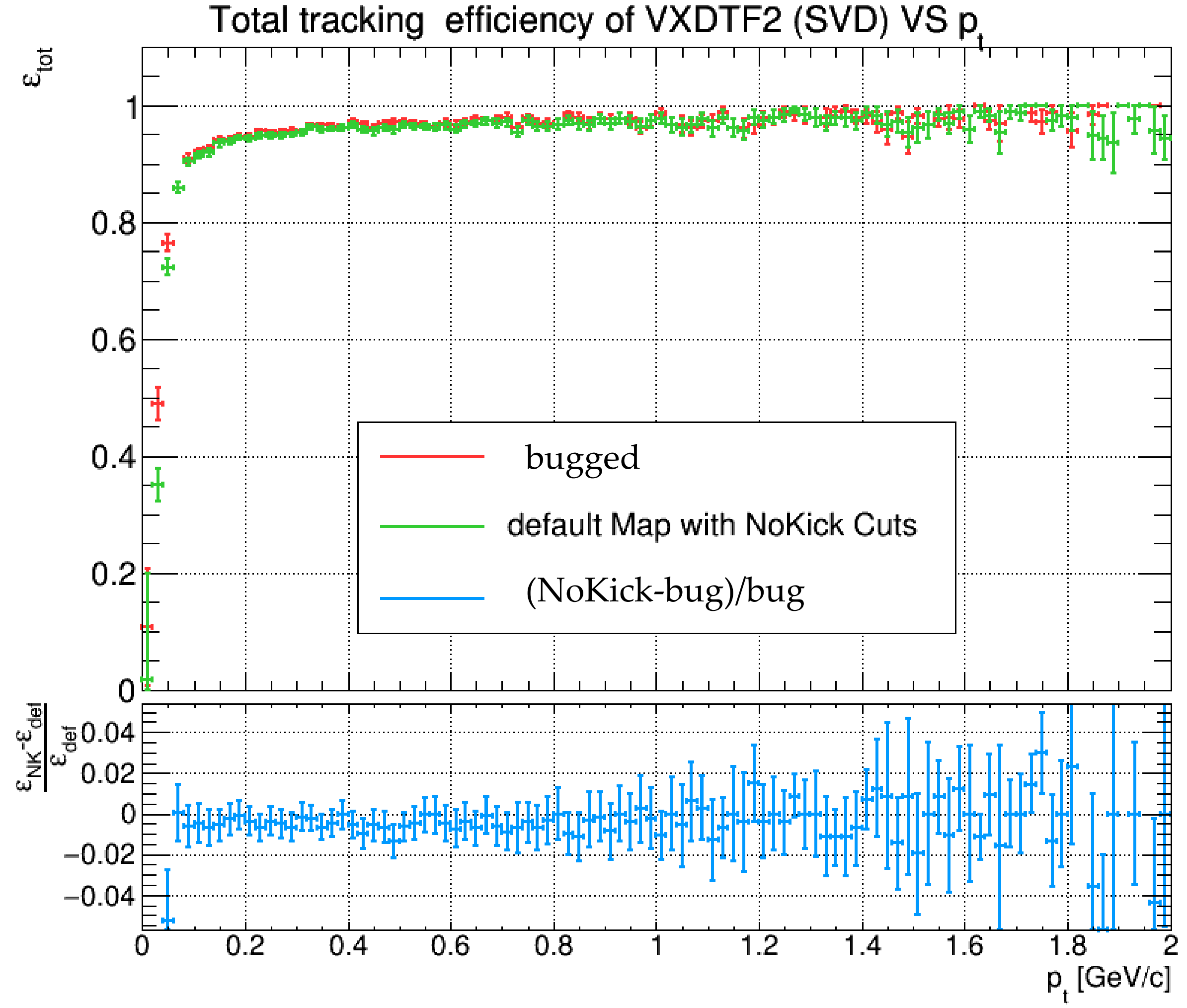
OLDDB MAP BKG FIT	mean (ms)	error (ms)	MEMORY (MB)
SetupGenfitExtrapolation	0	0	
DAFRecoFitter	40.74	88.59	12
TrackCreator	124.94	74.59	30
TOTAL	165.68	163.18	42

Tracking performances – bugged Sector Map

NB: VXDTF2 @ Relese 01-00 with bugged Sector Map

BUG
Limits of filters set to 0 if negative

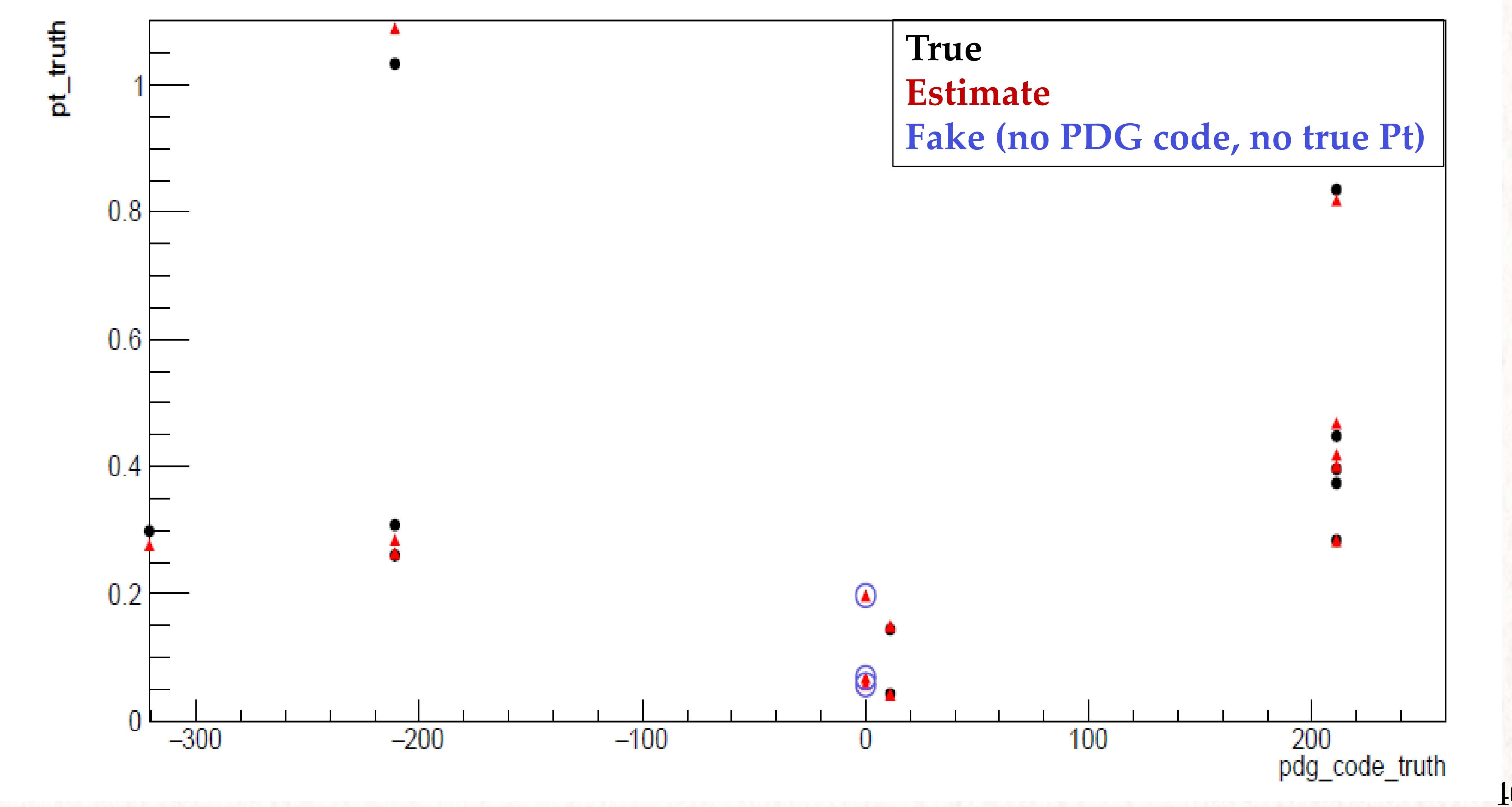
Average =96.0%



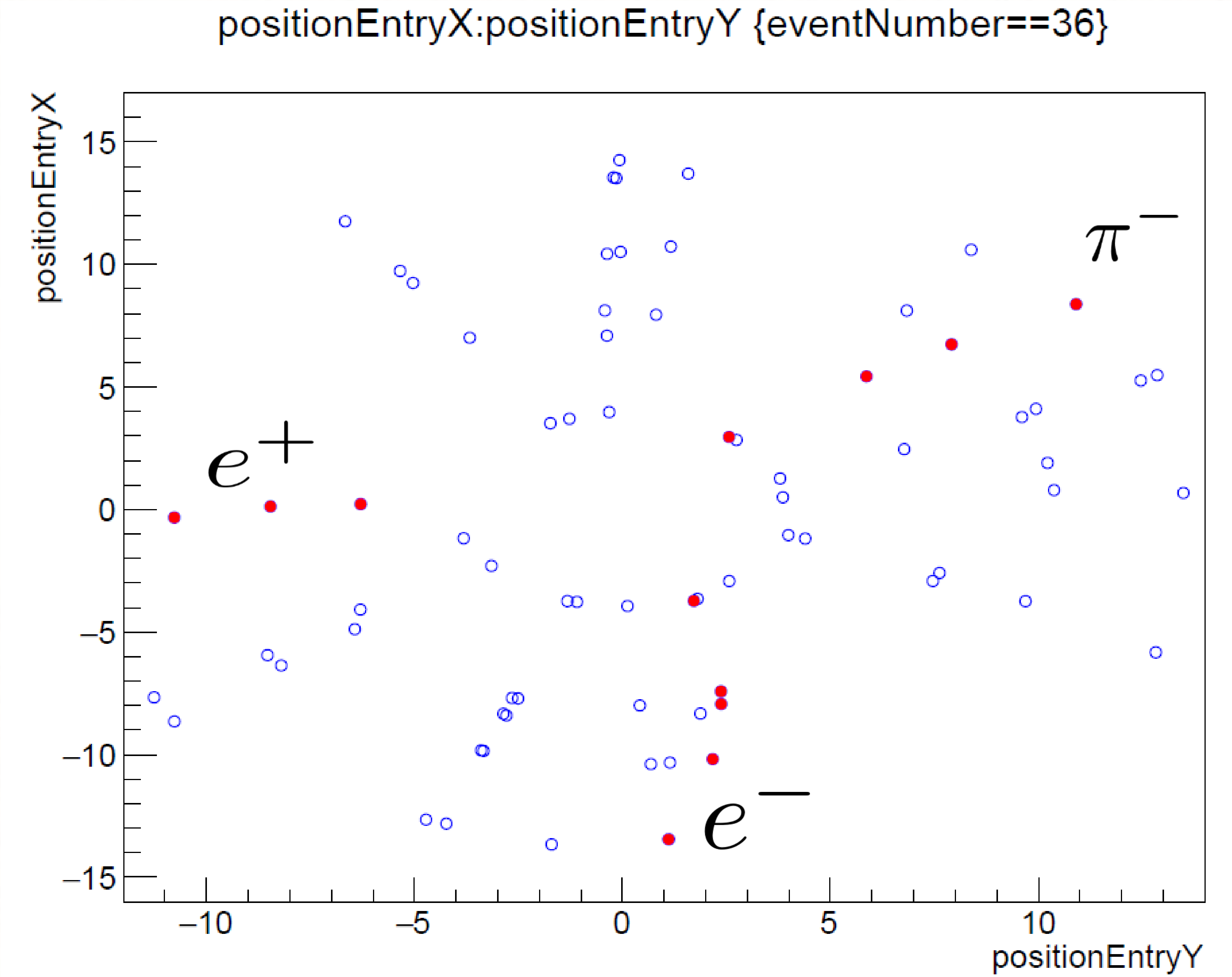
Time & Memory

- Same Fitting Time & Memory
- Doubled PR time
- 185 MB of PR-Memory

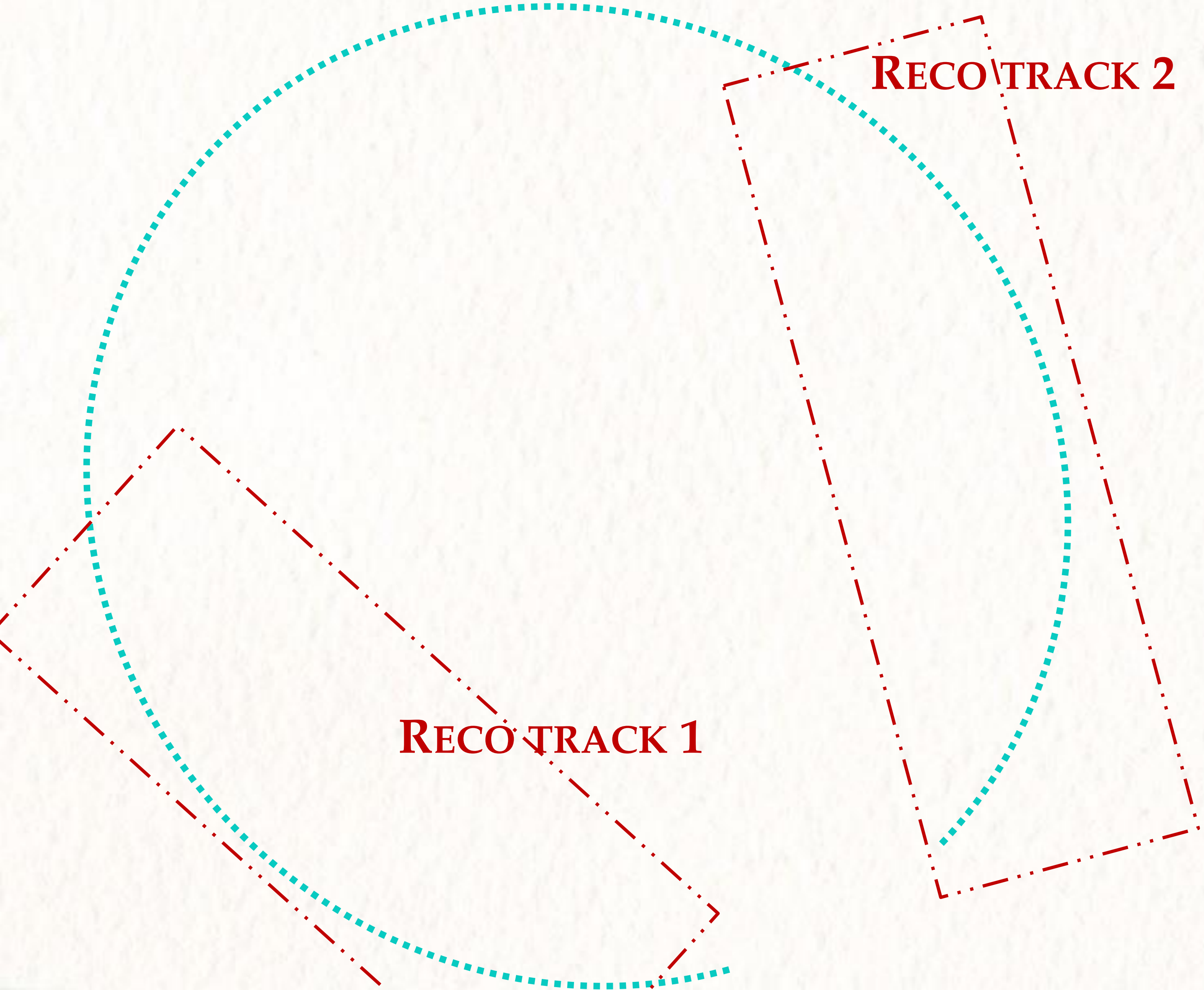
Fake Rate – track matching



Fake Rate – another track matching



Fake rate: possible explanation and proposal



1. Reconstructed 2 different tracks (one with wrong direction)
2. Two Reco Tracks do not overcome the matching requirements separately
3. Double counted as 2 fakes!

- The fake should be normalized to MCParticle, not MCTracks
- Can the CDC merging match 2 Reco Tracks and help to reduce the fakes?