Vertexing and co

Clement Helsens, CERN-EP

In collaboration with Donal Hill and Emmanuel Perez

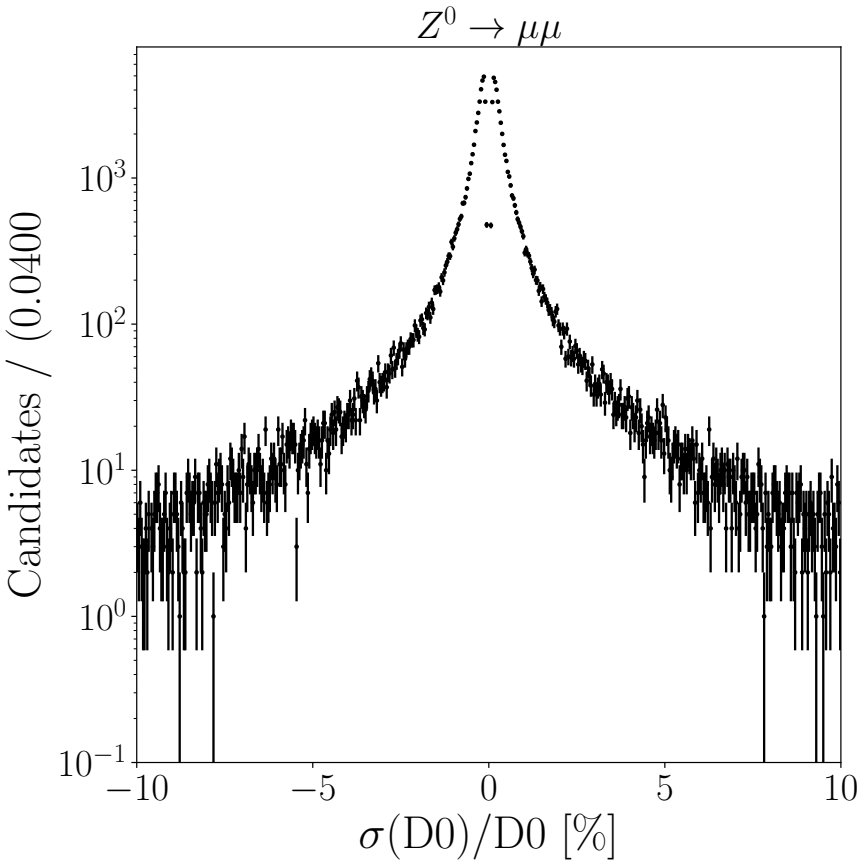
Thanks to Emilia Leogrande and Andre Sailer for the help with LC tools

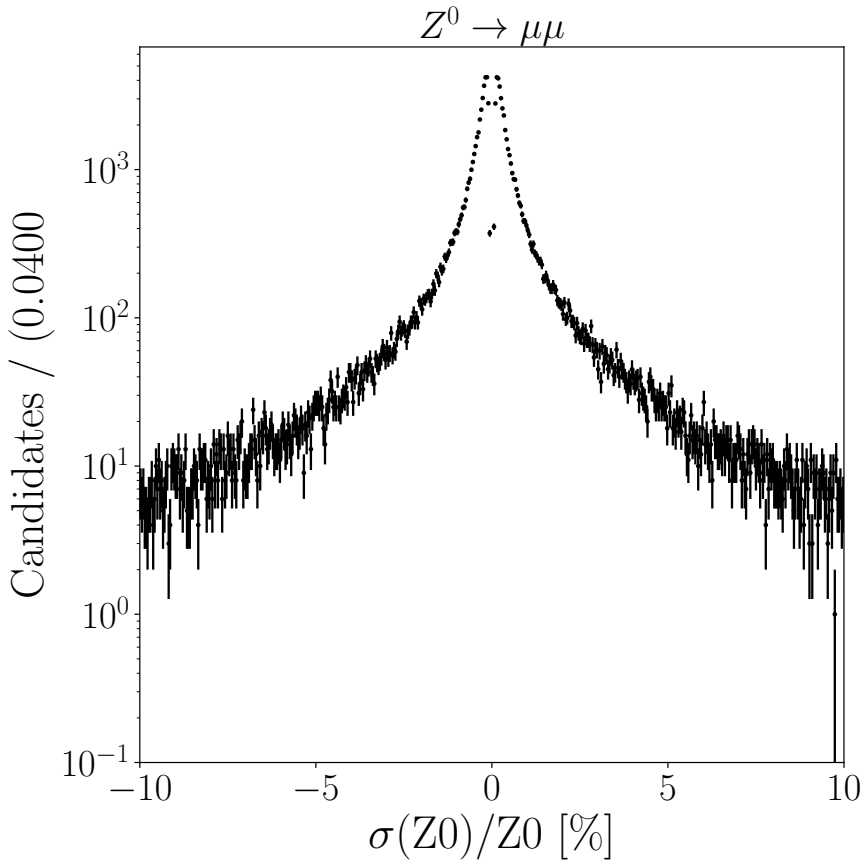
Introduction

- First look at vertexing and flavour tagging
  - Produce EDM4Hep events with Delphes IDEA trackCov
  - Convert the EDM4Hep to LCIO
  - Run primary and displaced vertexing with LCFIPlus
  - Training Flavour tagging with LCFIPlus

Sanity plots

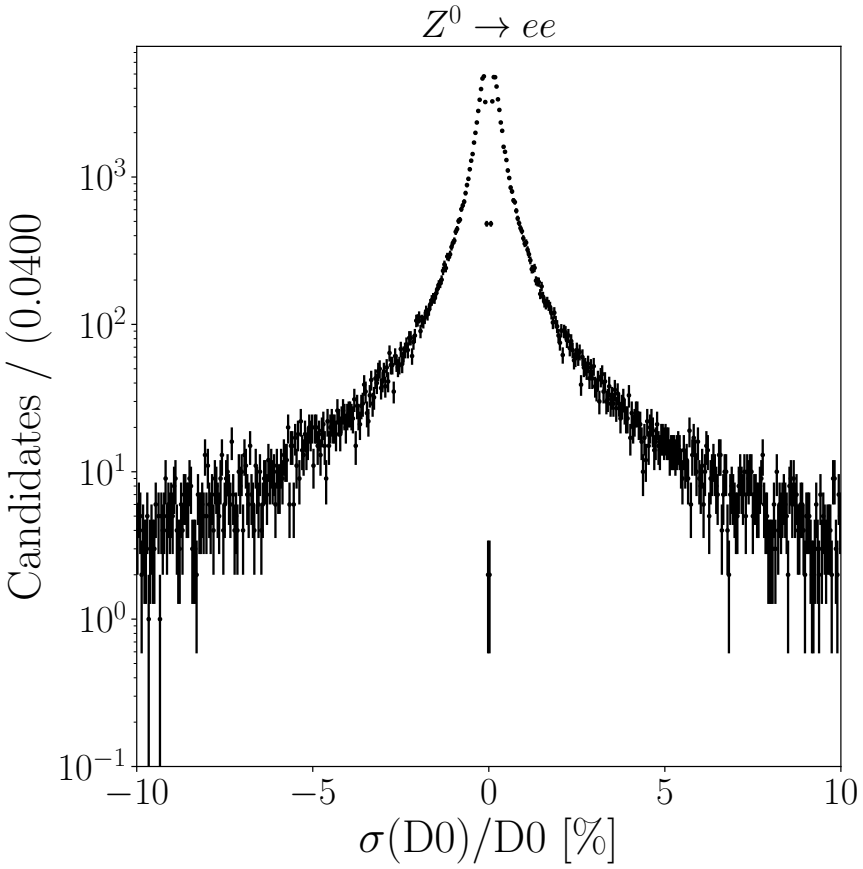
From track covariance

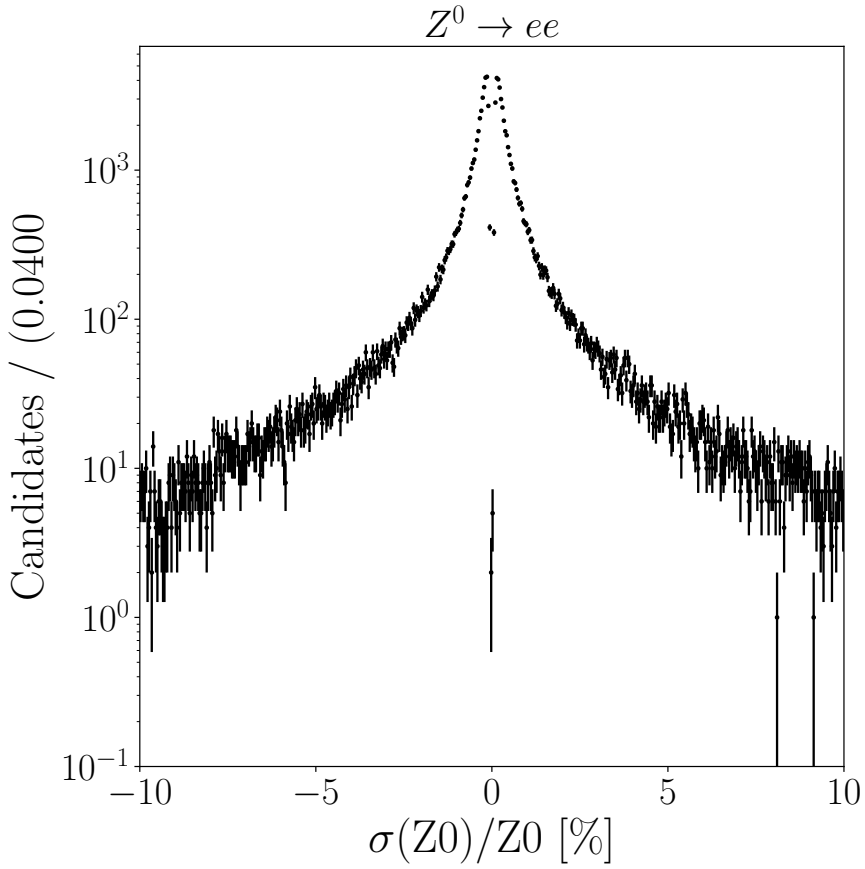




Sanity plots

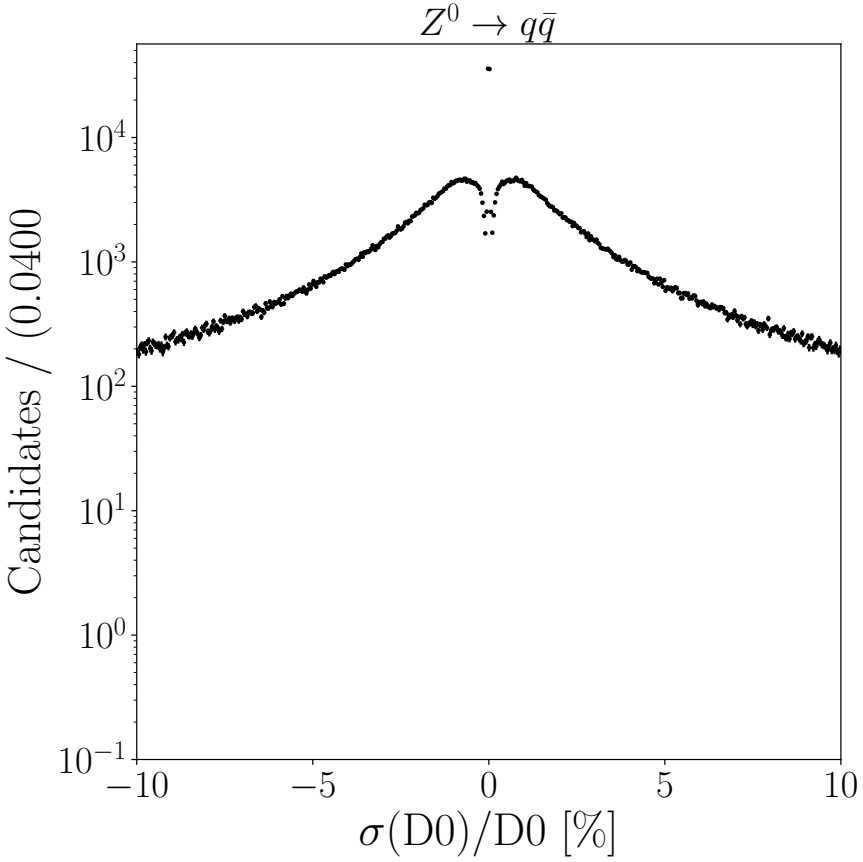
From track covariance

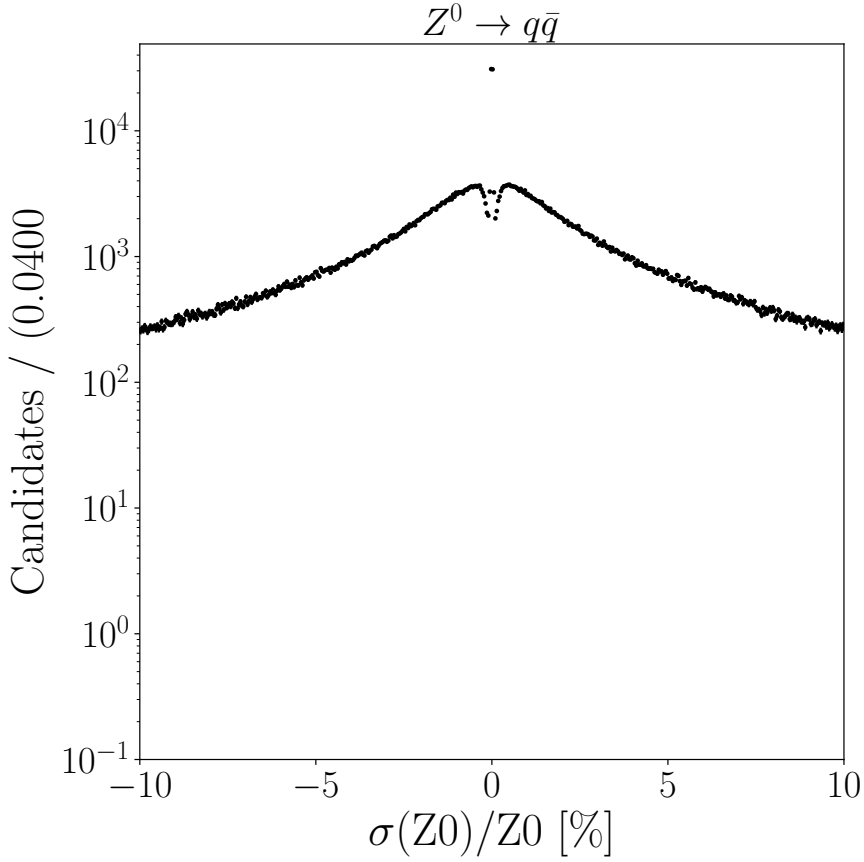




Sanity plots

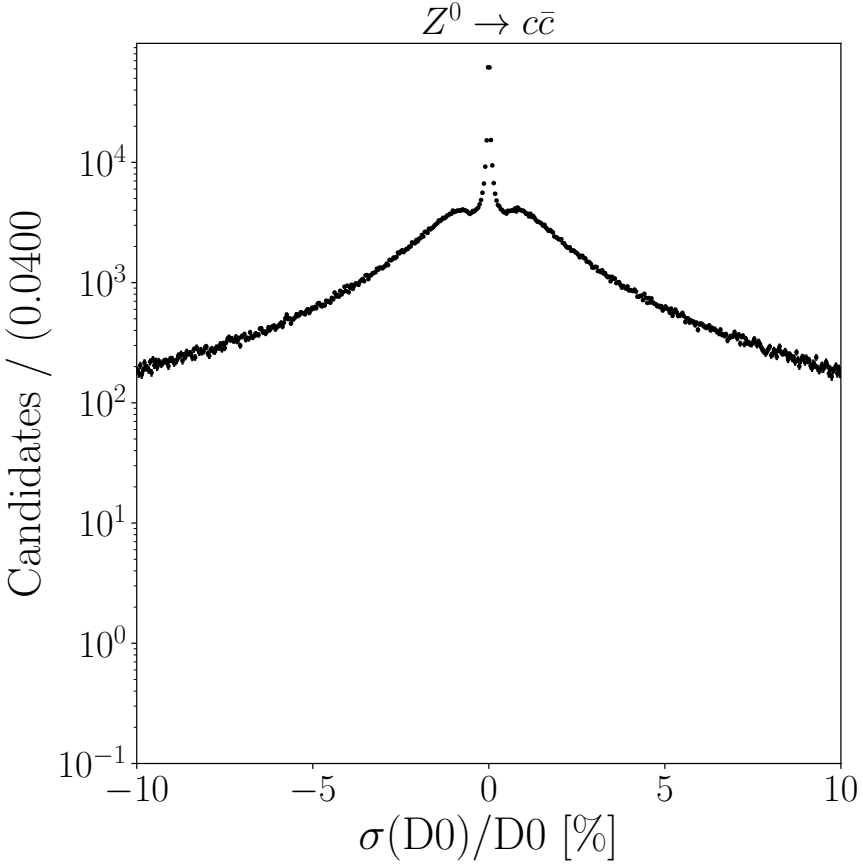
From track covariance

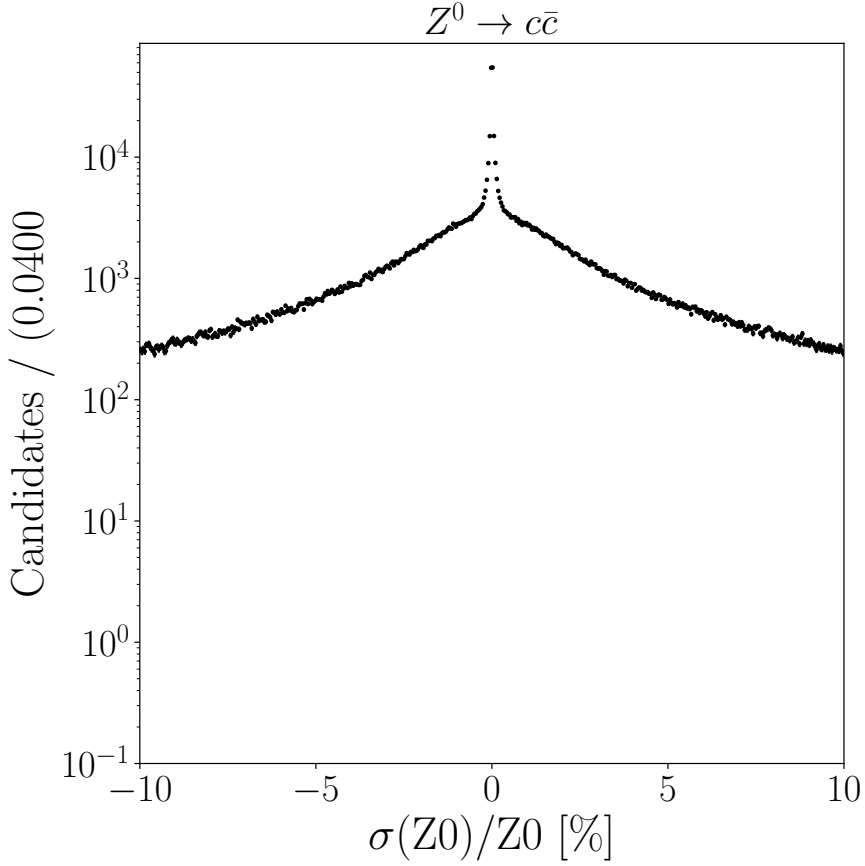




Sanity plots

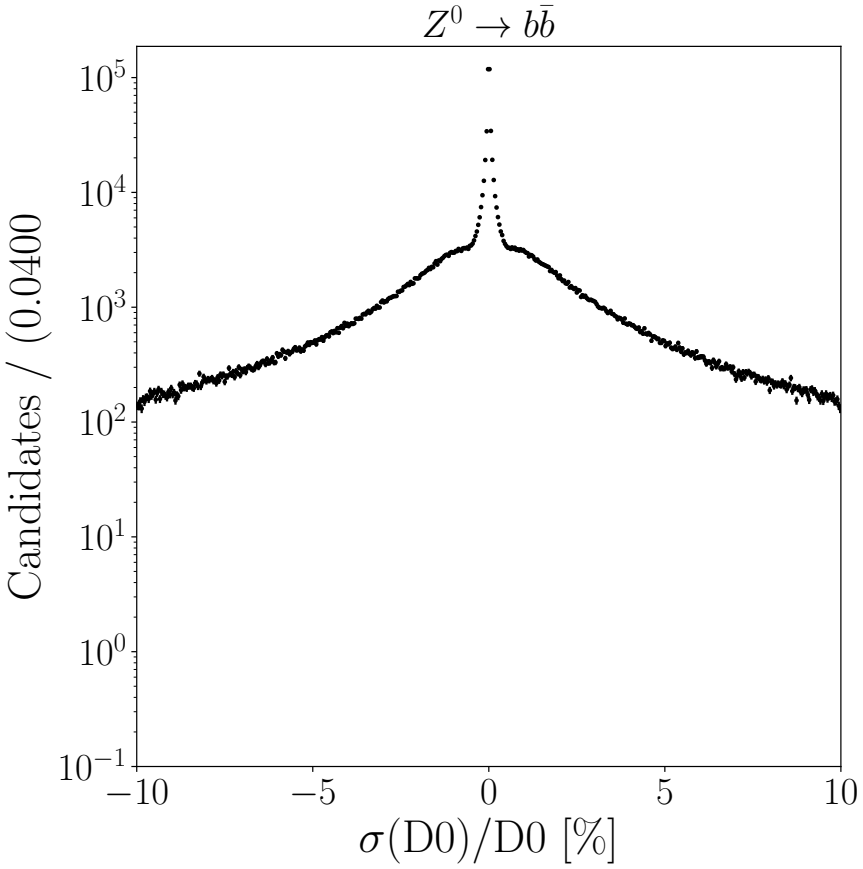
From track covariance

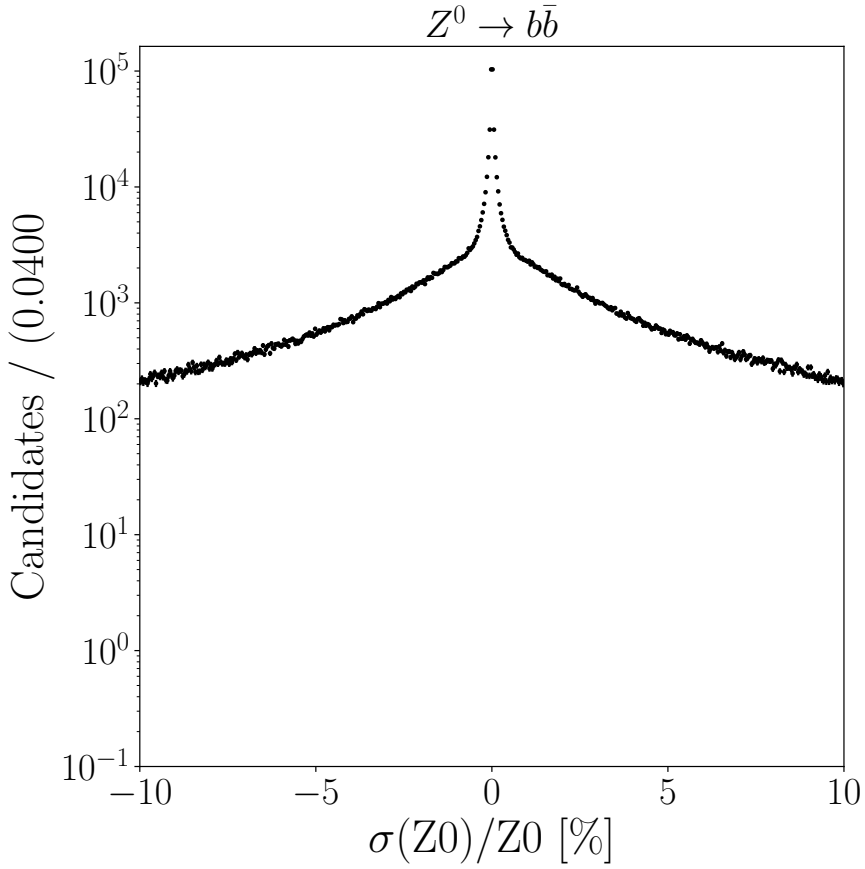




Sanity plots

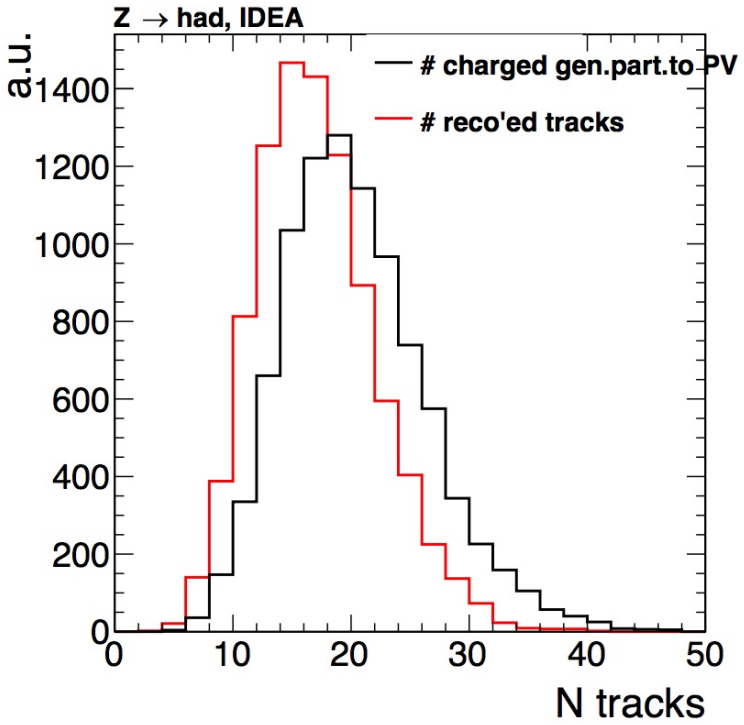
From track covariance

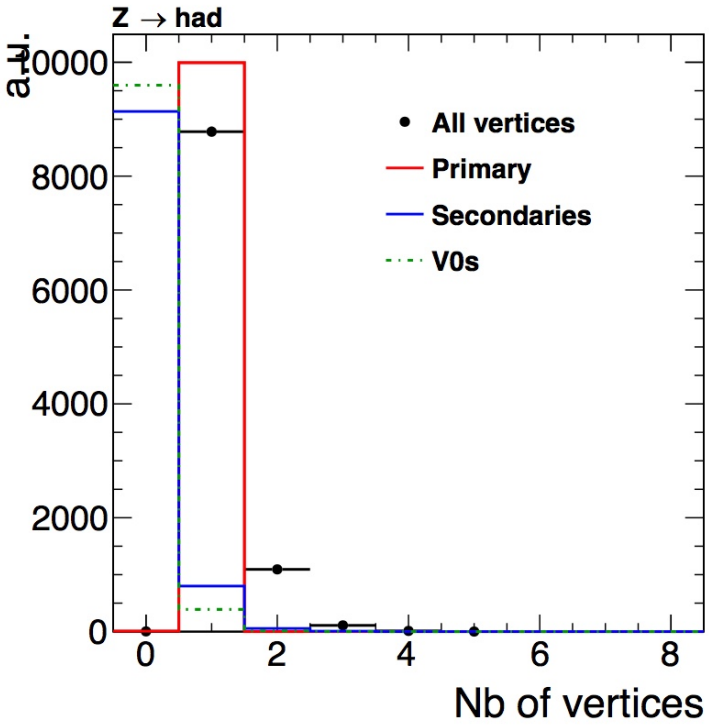




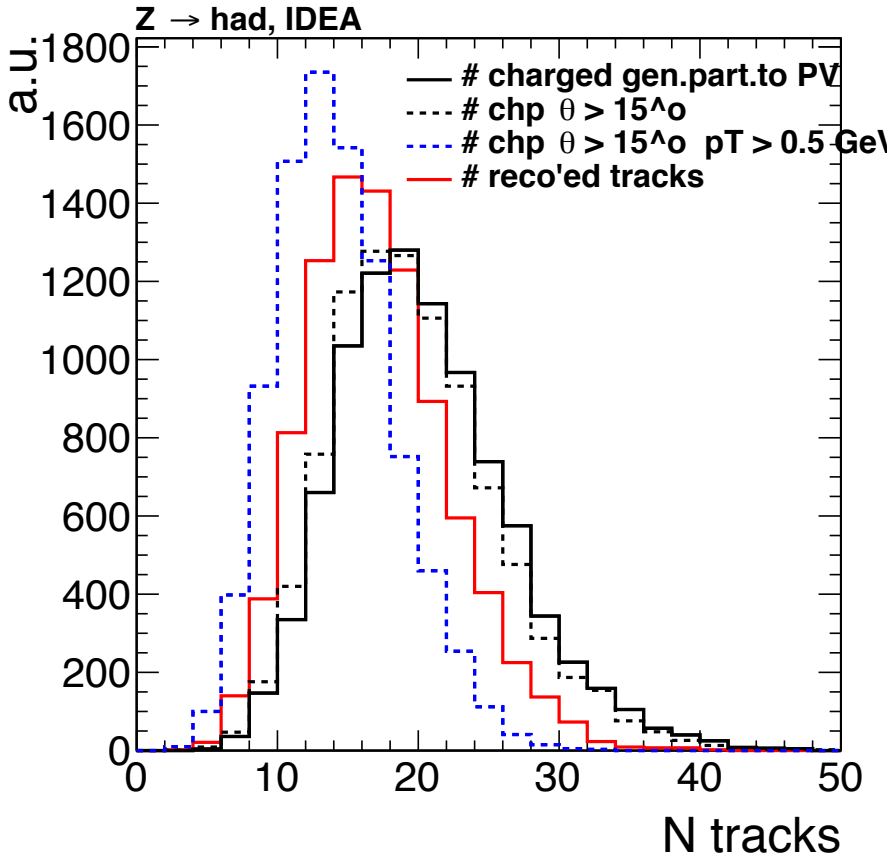
Z->qq(q=uds) θ>20°

No vertex smearing





Z->qq(q=uds)

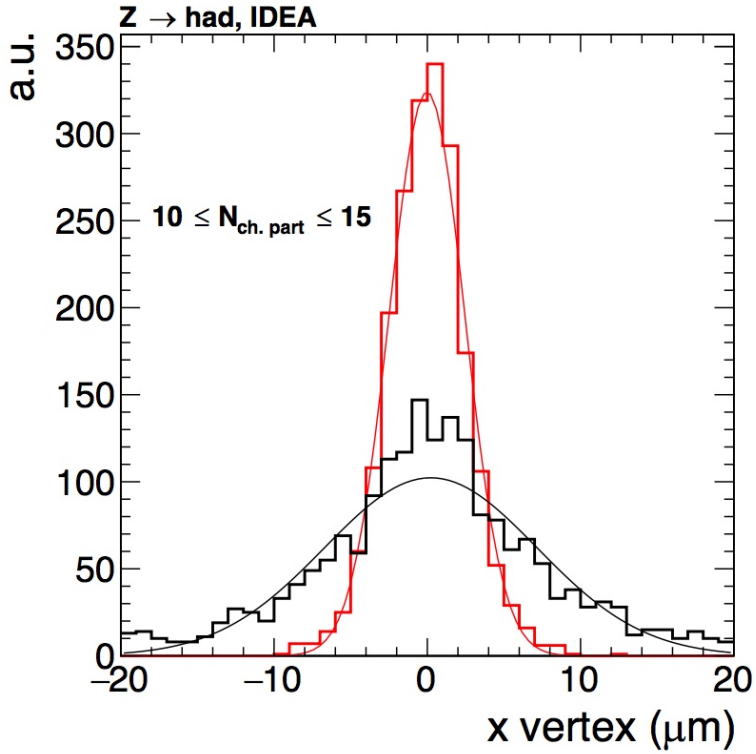


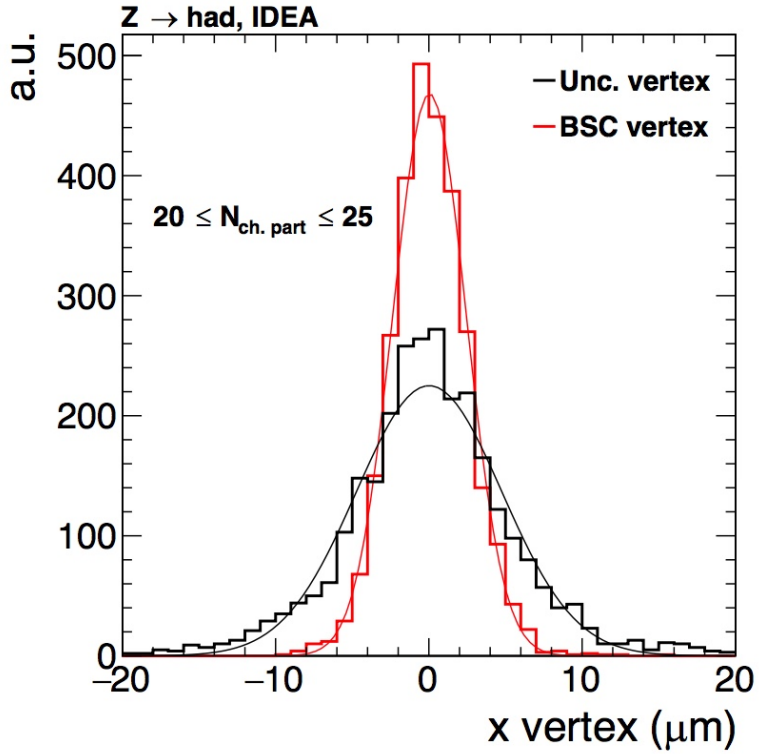
Comparing generated particles with different cuts and reconstructed ones

Z->qq(q=uds) θ>20°

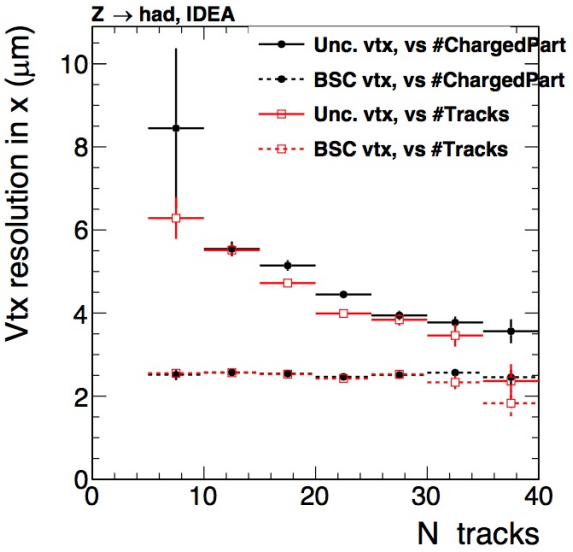
Unconstrained vertex: Less Gaussian resolution

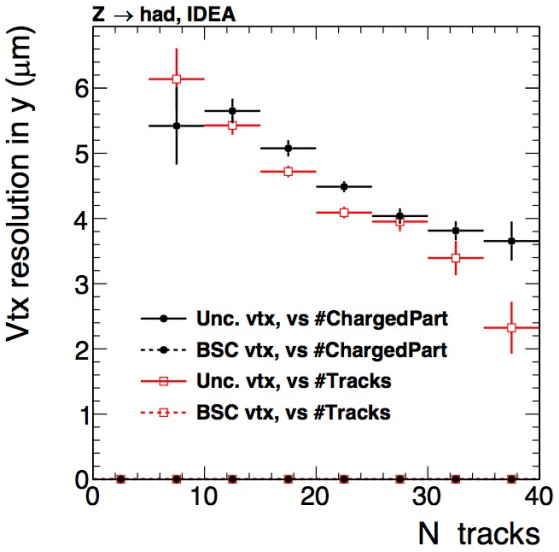
Next slide: resolution = Gaussian mean ±10 μm

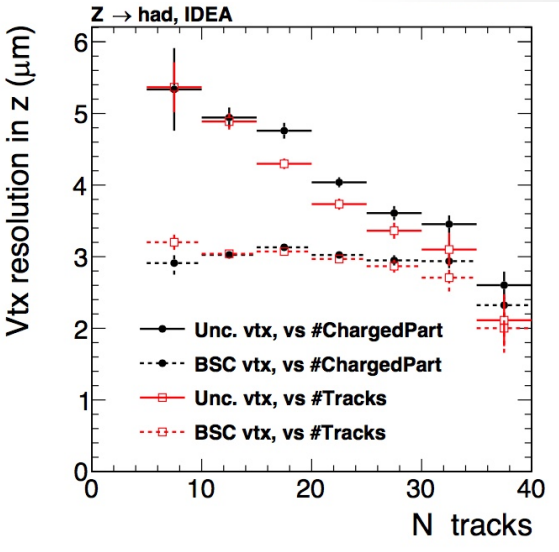




Z->qq(q=uds) θ>20°







Beam Spot Constraint:

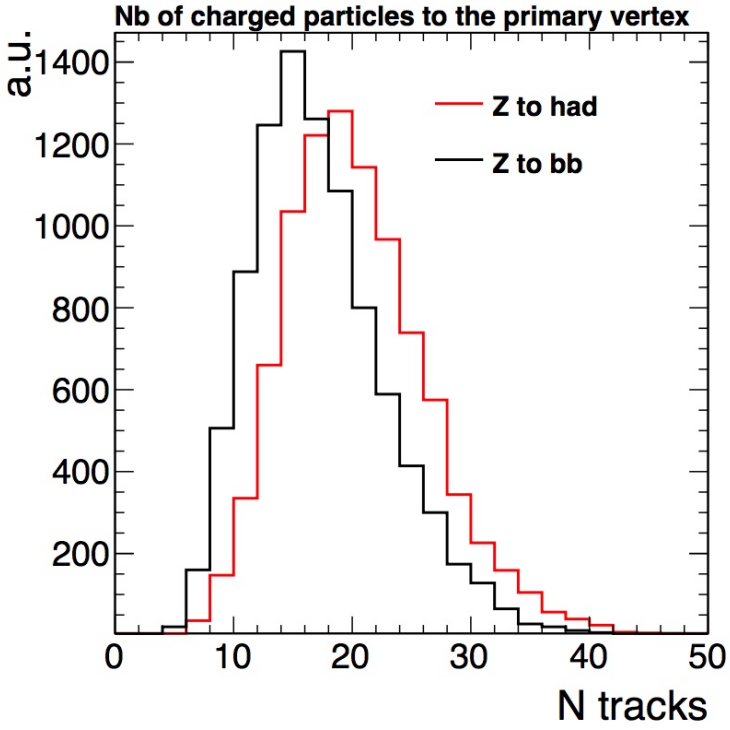
- Resolution ~ independent of N_{tracks}
- tiny resolution in y, about 0.2 nm
  - Events generated @ (0,0,0)
    since exp, resolution is >> beam size in y
- σ(x) ≈ 2 μm and σ(z) ≈ 3 μm

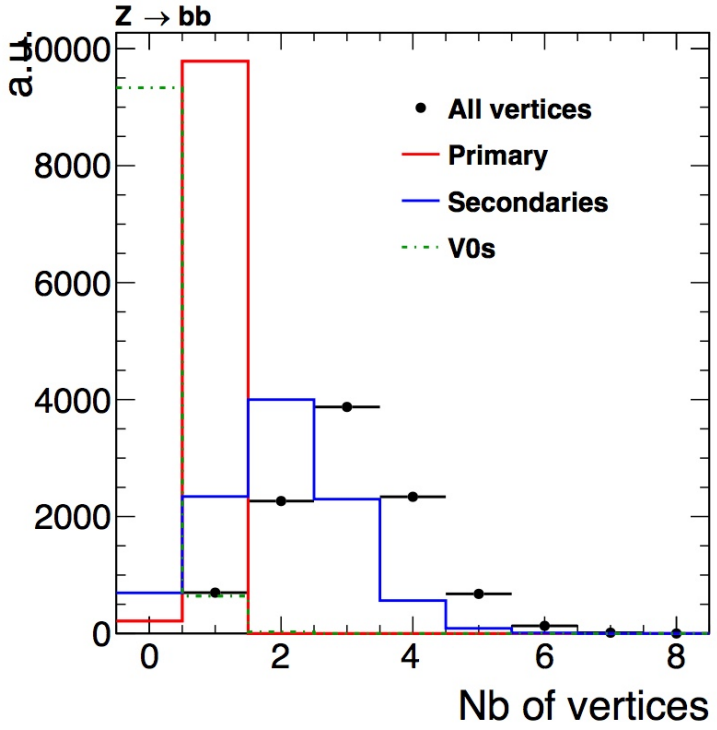
Unconstrained:

- Resolution in x and in y are similar
- σ(x, y) ≈ 4 - 5 μm and σ(z) ≈ 4 μm

Z->qq(q=uds, b) θ>20°

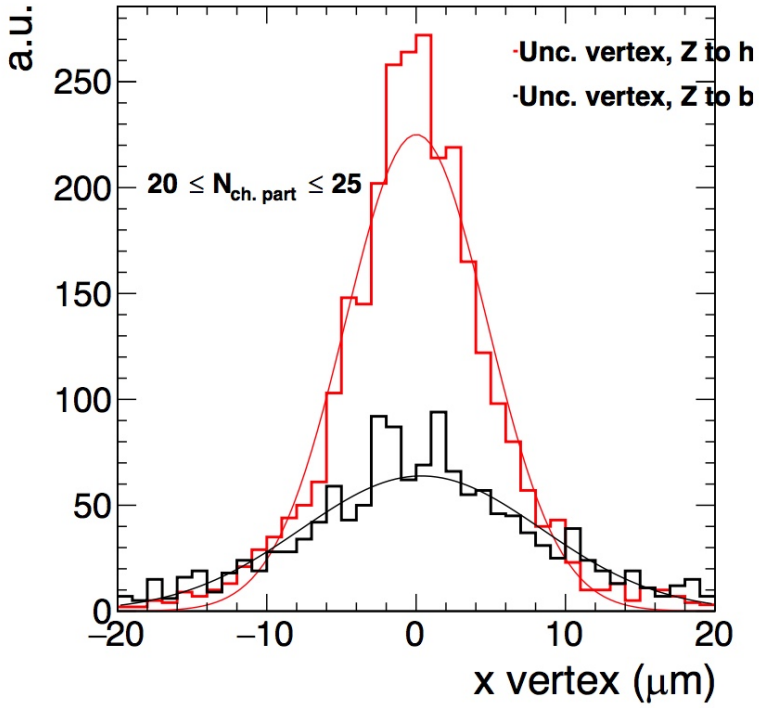
About 4-5 ”primary tracks” less compared to Z to had (remove from the charged, stable gen particles those that are produced by the B meson/hadrons formed by the b’s from the Z)

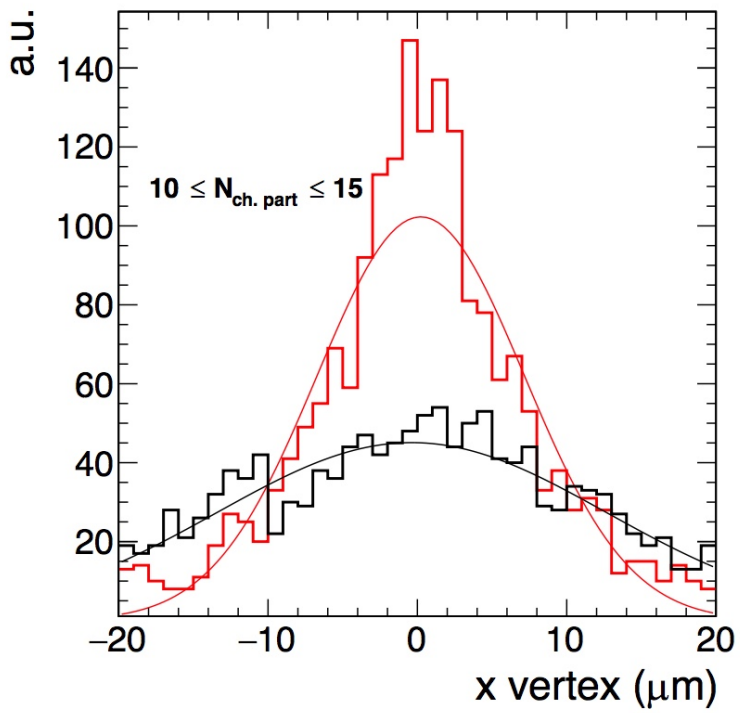




Z->qq(q=uds, b) θ>20°

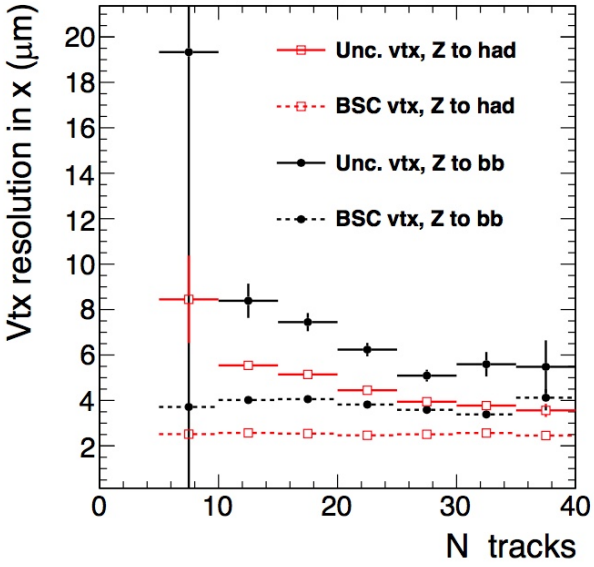
Worse resolution for Z->bb even when binned in the number of primary tracks.

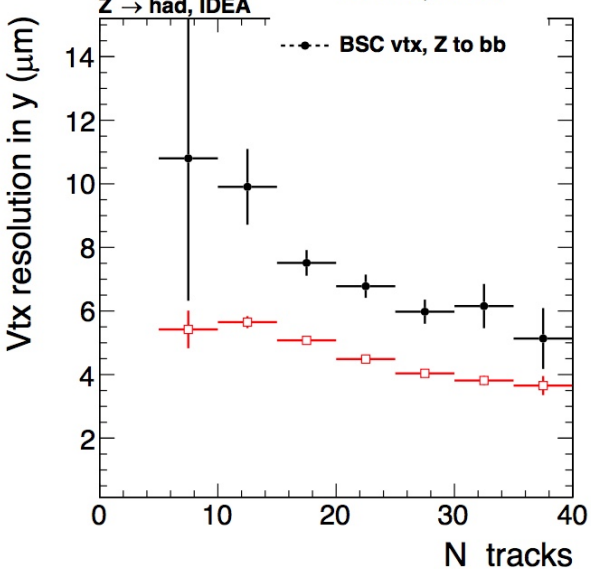


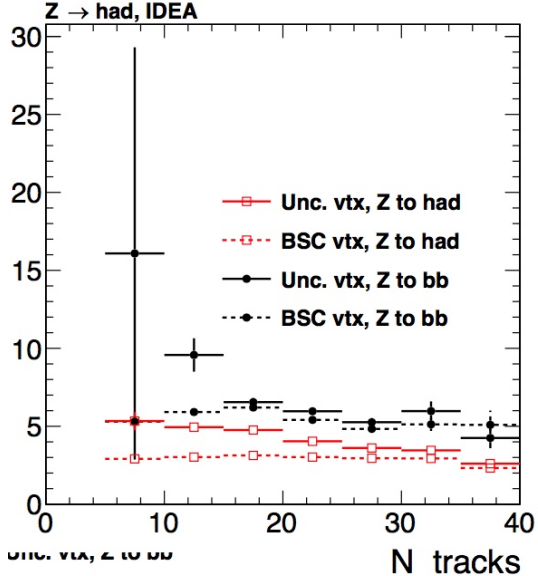


Z->qq(q=uds, b) θ>20°

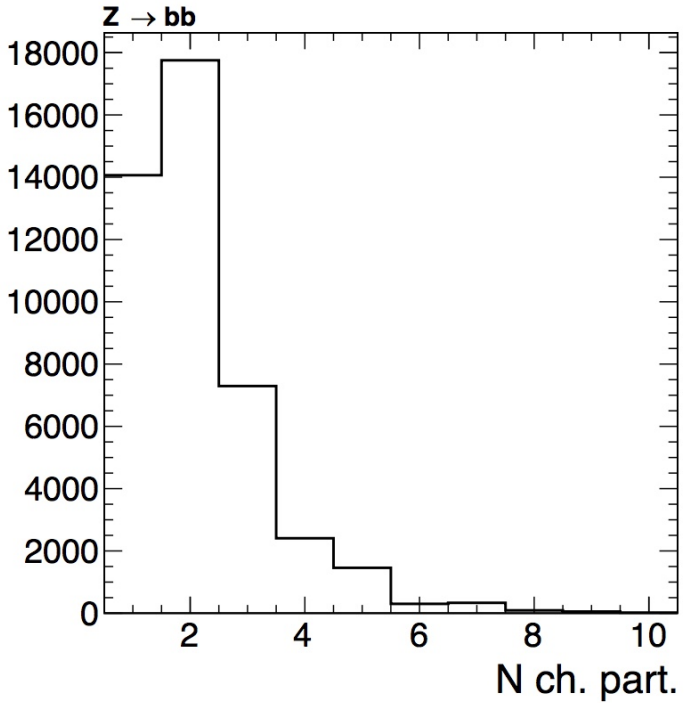
Resolution in z : about 5 μm in z for Z->bb (compared to 3-4 for Z->uds)
Resolution in (x,y) : about 6 μm for Z->bb (compared to ~ 4 μm for Z->uds)

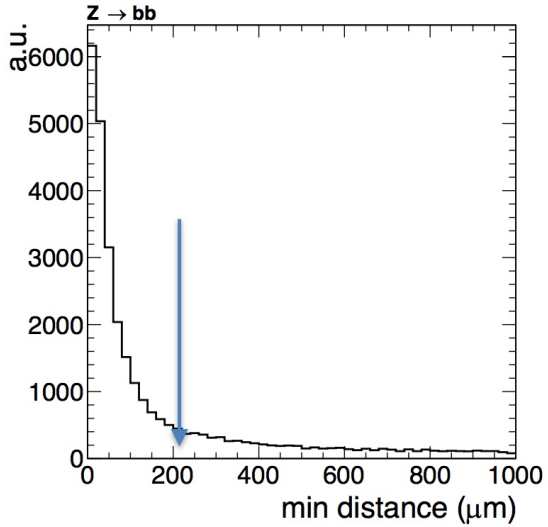






Z->bb secondary vertices

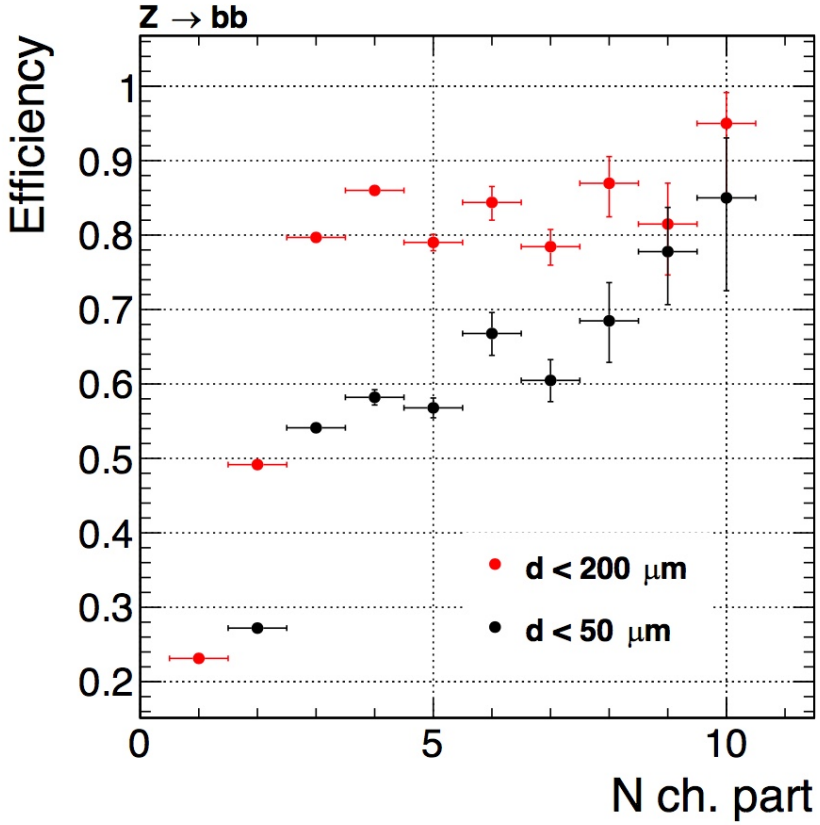




Gen level : All displaced vertices in B decay chains
Plot number of charged stable particles out of each displaced vertex

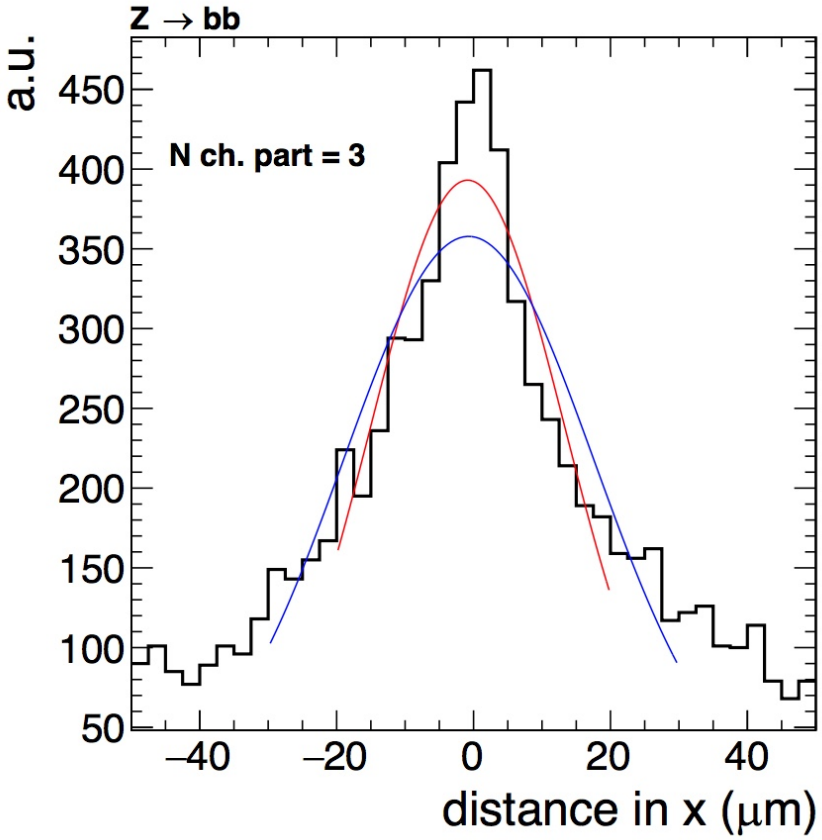
Match gen-level vertices with the reconstructed one (closest 3D distance).
Resolution plots made for vertices with d < 200 μm

Z->bb secondary vertices



Efficiency that a gen-level displaced vertex (i.e. secondary/tertiary/etc from the B's) is matched with a reconstructed displaced vertex, within 200 μm or 50 μm

Z->bb secondary vertices



Example: decays with 3 tracks (charged particles)
The distance in x to the closest reconstructed displaced vertex is shown

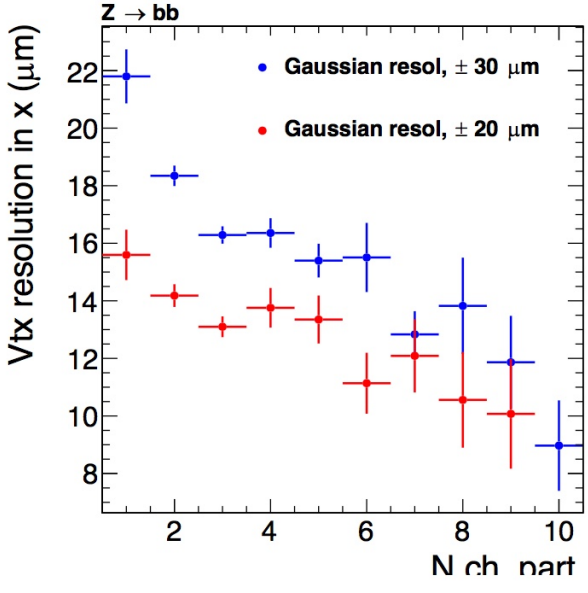
Define nevertheless a “resolution” from a Gaussian fit for simplicity.

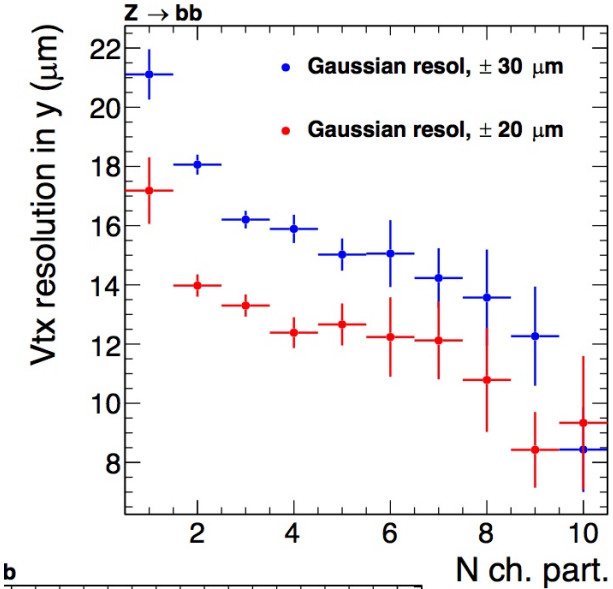
Vertex quality cuts may reduce the tails..

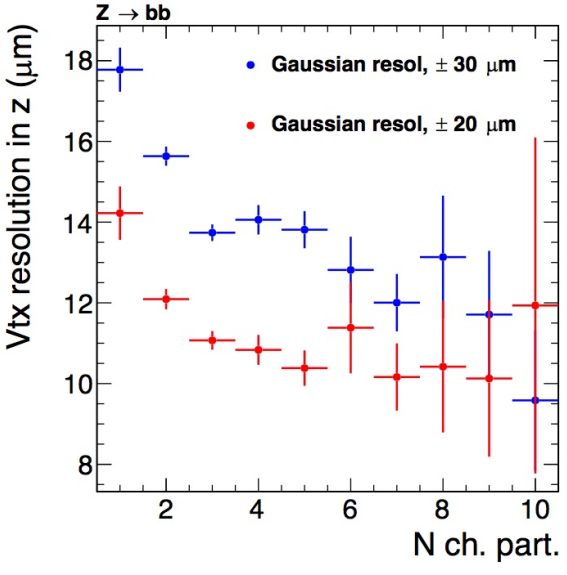
Z->bb secondary vertices

Resolutions

- ~ 15 µm in x,y
- ~ 12 µm in z





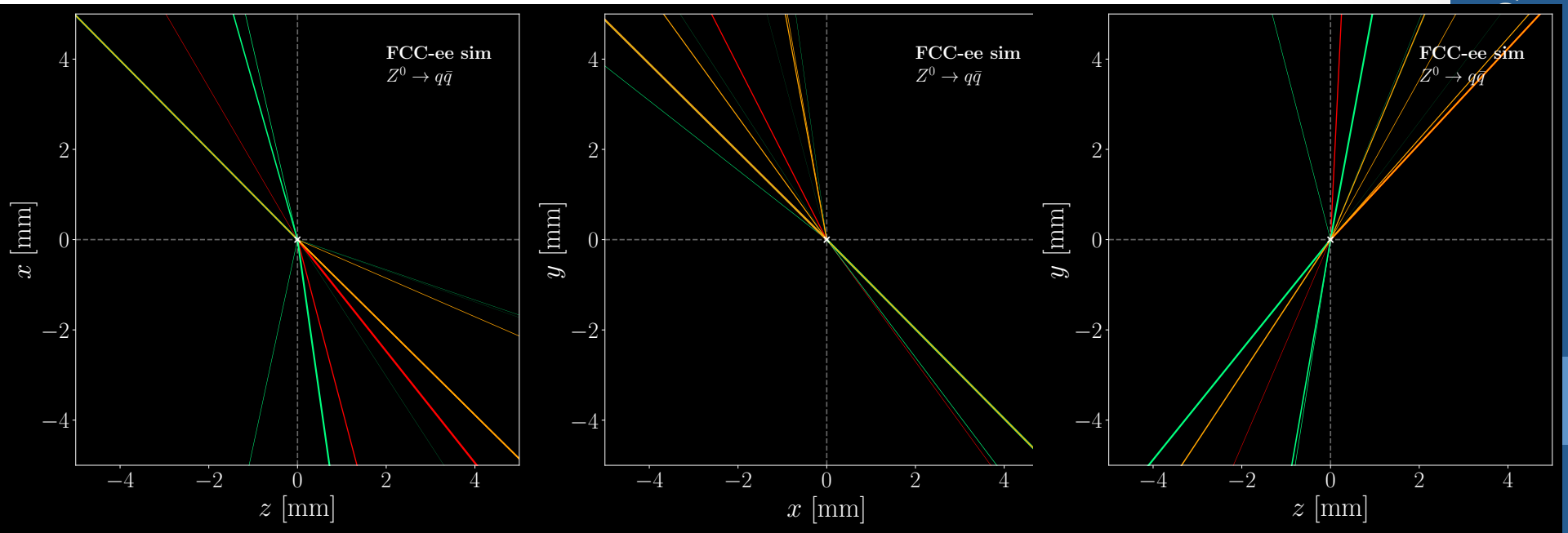


Simple Event displays

- Inputs:
  - Generated particles
  - Reconstructed vertices
- Color code:
  - Orange (photon)
  - Green (pion)
  - Red (kaon)
  - Pink (proton)
  - Blue (muon)
  - Turquoise (electron)
- Particles are drawn with a line width proportional to lop(p)

Z->qq

Everything well at the PV! (0,0,0)
no vertex smearing

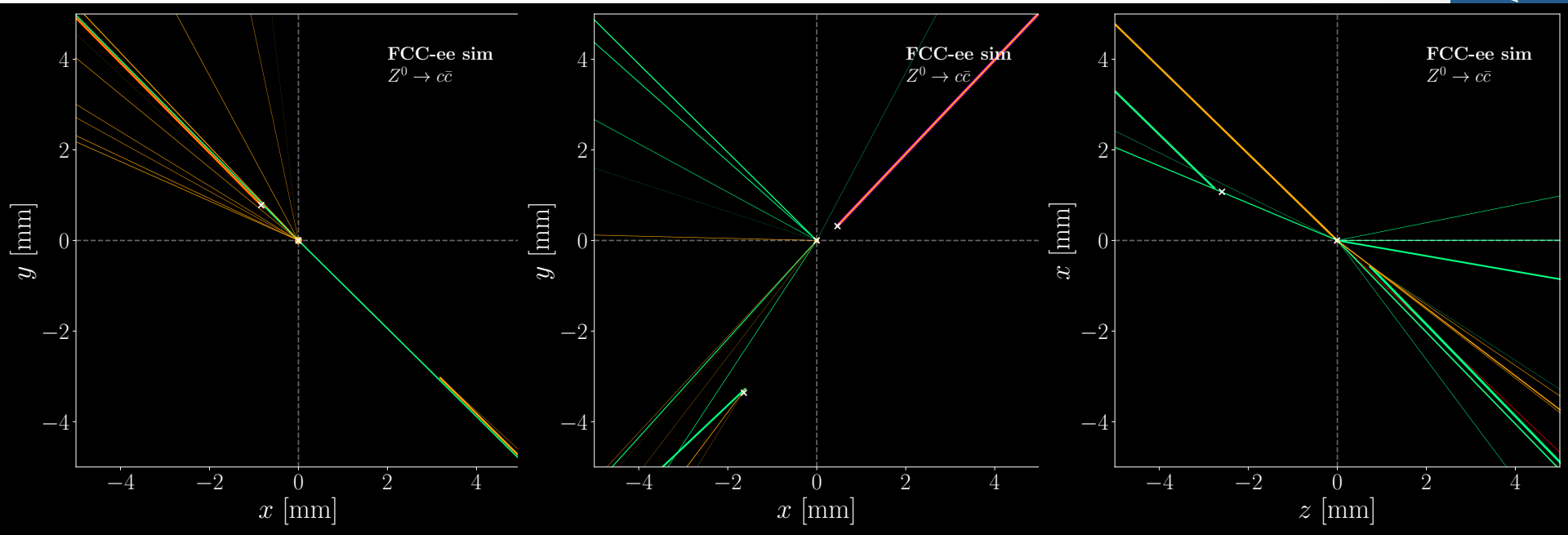


Z->cc

D0->K K candidate with well associated vertex

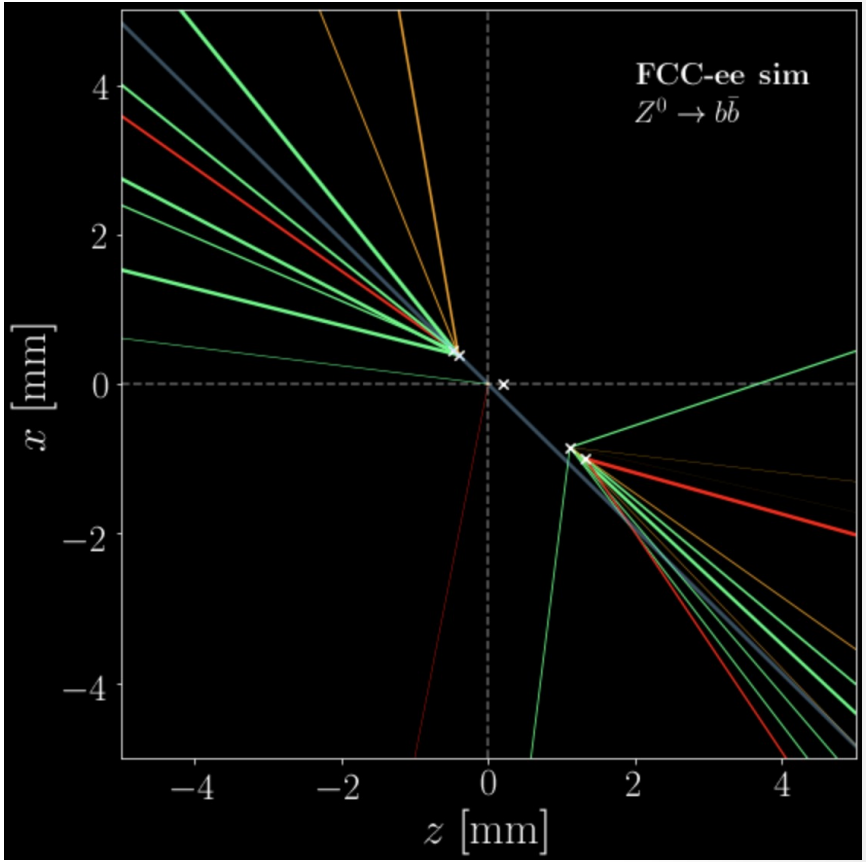
Two DV one with (ρKπγ) which could be a Λc
The other (ππ) probably from a D_0 or ρ0 produced in a D decay
Both vertices well associated

Contains a D0 (Kπ) most likely
Vertex not perfectly reconstructed



Z->bb

- Secondary and tertiary vertex from both B's
- By eye (not mine but Donal's)
  - Bottom b decay is B0->D0 pi pi, D0->K K
  - Top one can't really say, vertices are too close
- Faint blue line is event thrust axis based on the reco particles

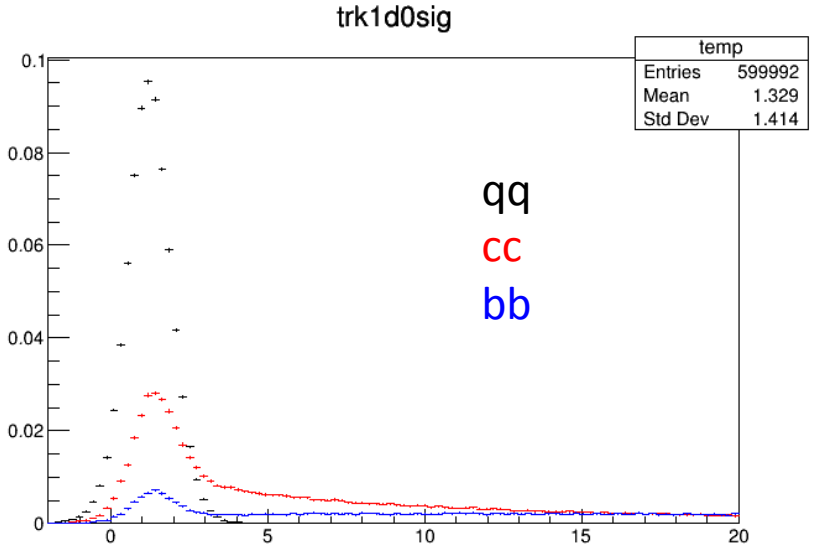


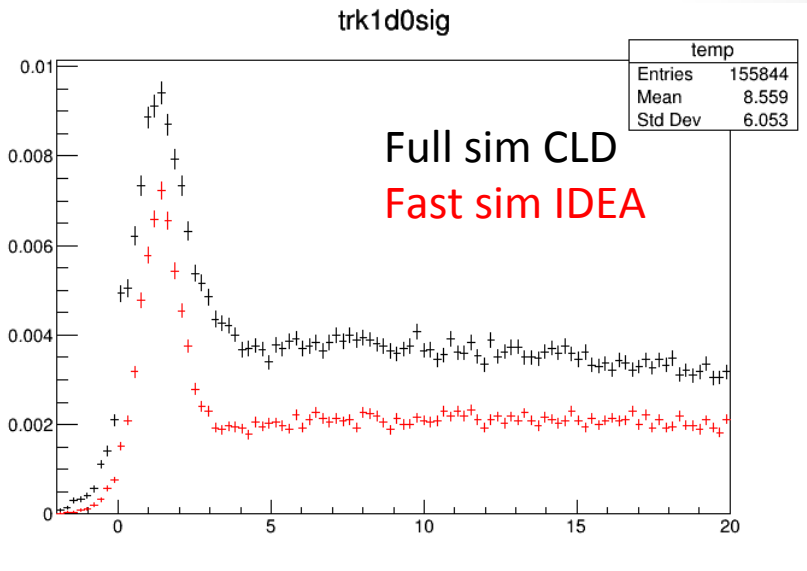
Flavour tagging: context

- Trying to use CLIC flavour tagging but using Delphes IDEA
  - https://github.com/iLCSoft/CLICPerformance/tree/master/Flavour_tagging
- Lots of preliminary results
- Had to take a lot of short cuts
- Framework integration to come later
- Goal is to compare CLIC Ftag and DL Ftag (Loukas, Michele)
  - In Delphes sim
  - In Full sim

Flavour tagging: input example

Delphes sim

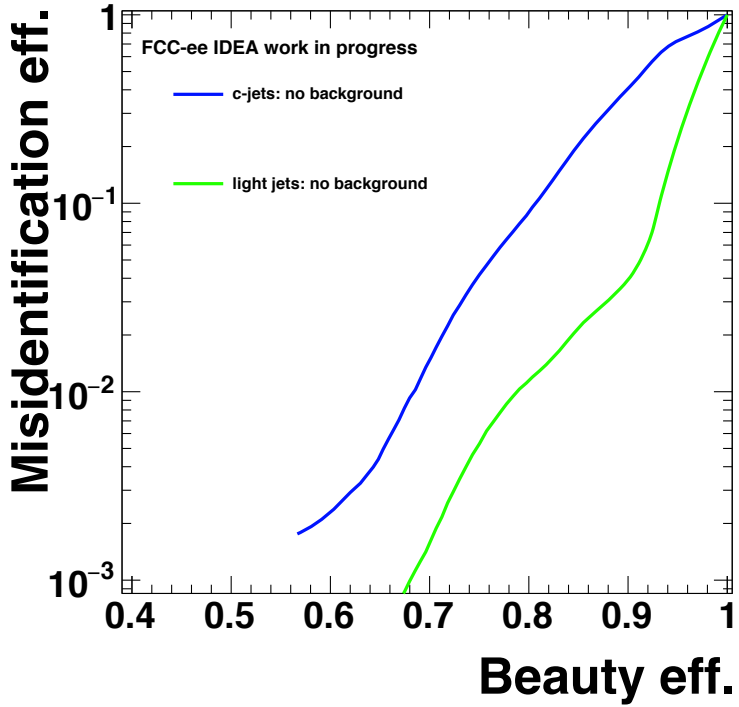


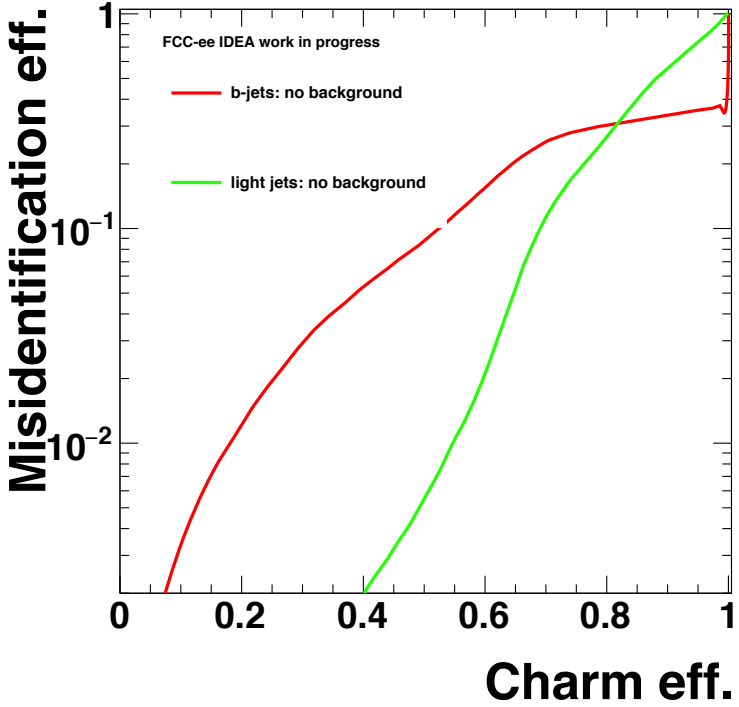


Just had time 5mins before the talk to plot one variable
CS team will have to do more detailed comparisons

Flavour tagging: results

Results out of the box, to be understood
θ=50°





Summary

- First look at vertexing and flavor tagging
- Procedure to run LCFIPlus from Delphes EDM4Hep in place
  - Framework integration will take longer
- The CS H->cc can now take over
  - All the events CLIC used for Ftag training have been produced in EDM4Hep
  - /eos/experiment/fcc/ee/tmp/FTag/Ntup/