Measuring Primary Vertex Resolution in Data*

Kevin Burkett, Yanyan Gao* (Fermilab)

06 November 2009

Turf-Surf Working Meeting

Outline

- Introduction to primary vertex fitting
- Measure vertex resolution with two-vertices method
  - Algorithm description
  - Resolution and pull definitions
  - Data samples
- Resolution and Pulls
  - Compare the MC method and two-vertices method
  - Compare TTbar, QCD_Pt_30 and QCD_Pt_80 results
- Conclusion and on-going activies

Vertex Fitting Algorithm in CMS (I)

- Vertex Fitter: Adaptive filter Fitter(trks, vertices)

  Iterative re-weighted least-squares fit which down-weight tracks by their std χ^2 distance to the vertex

- At RECO, two vertices are reconstructed:
  - offlinePrimaryVertices, no BS constraint in vtx fit
  - offlinePrimaryVerticesWithBS, with BS constraint in vtx fit

- Primary Vertex Fitting in CMSSW
  - Vertex Reconstruction: RecoVertex/PrimaryVertexProducer/
  - DataFormat: DataFormats/VertexReco/

- Document: CMS Note 2006/032

Vertex Fitting Algorithm in CMS (II)

- Vertex fitting work flow

1. Select tracks based on TrkFfilter:

   - D_0 signficance > 5 - nPixelHits$>=2$ - nHit$>= 7$ - $\chi^2/\mathrm{ndof} <= 5$

2. Cluster the tracks in z

   separation between clusters < 1 mm

3. Run pvtx fitter on each cluster with and without BS constraint

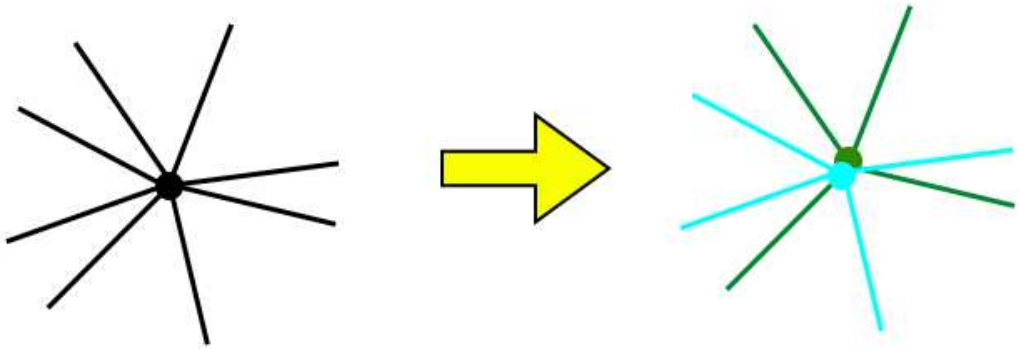
4. Select reconstructed vertices based on quality cut:

   -maxDistanceToBeam $<$ 500 μm (200 μm with BS) - minVertexFitProb $<=$ 0.01

5. Order reconstructed pvtxes in descending order of $\sum_{i=0}^{ntrks} p_T^2$

- Vertex resolution depend highly on track multiplicity and p_T

Resolution with Two-Vertices Method



1. Split tracks that pass the track filter into two independent sets
   - Tracks are filted by the same trkfilter as "offlinePrimaryVertices"

     D_0 signficance > 5 nPixelHits>=2 nHit>= 7 $\chi^2/\mathrm{ndof} <= 5$
   - Tracks are ordered in descending order of p_T
   - For each pair, randomly select which track goes to which trackset
2. Run primaryvertex fit on each track set
   - Use the fitter without BS constraint "offlinePrimaryVertices"
3. Compare the two fitted vertex positions and calculate
   - Resolution: σ of the gaussian fit to $\frac{x_1 - x_2}{\sqrt{2}}$
   - Pull: σ of the gaussian fit to $\frac{x_1 - x_2}{\sqrt{\sigma x_1^2 + \sigma x_2^2}}$

Resolution and Pulls Calculations

1. Calculate residuals of reconstructed vertex positions
   - MC Method: $x_{rec} - x_{sim}$
   - Two-Vertices Method: $\frac{x_{rec1} - x_{rec2}}{\sqrt{2}}$

2. Calculate pulls of reconstructed vertex positions
   - MC Method: $\frac{x_{rec} - x_{sim}}{\sigma x_{rec}}$
   - Two-Vertices Method: $\frac{x_{rec1} - x_{rec2}}{\sqrt{\sigma x_{rec1}^2 + \sigma x_{rec2}^2}}$

3. Fit residuals in a given nTrack bin with Gaussion
   - Resolution is taken as the fit σ
   - Statistics in each nTrack bin affects the fit performance

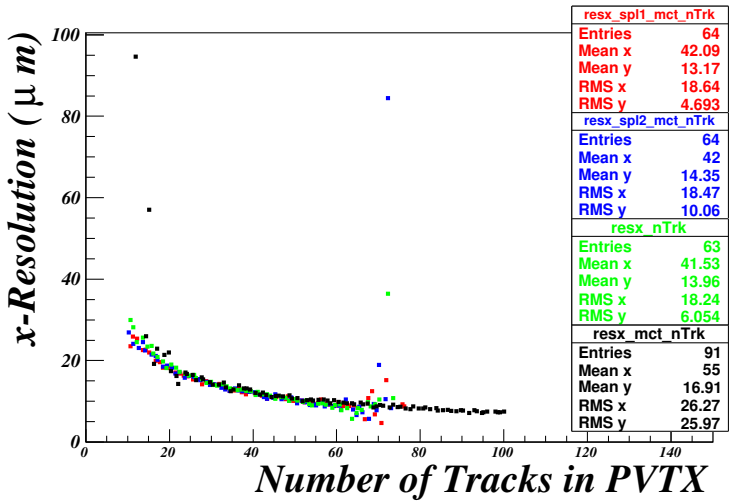
4. Fit pulls in a given nTrack bin with Gaussion
   - Check the fit σ consistency with 1.0
   - Statistics in each nTrack bin affects the fit performance

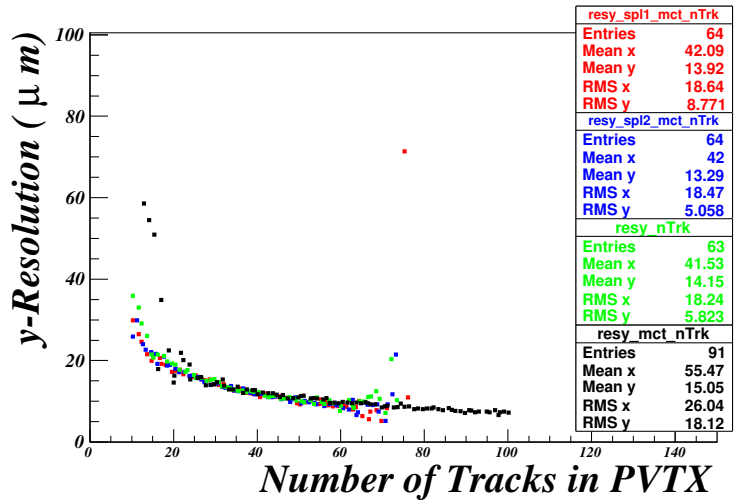
Data Samples

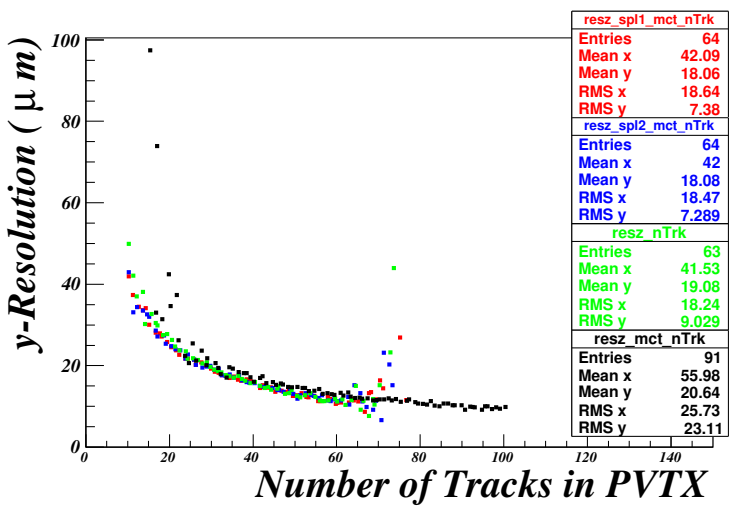
- TTBar
  - /store/mc/Summer09/TTbar/GEN-SIM-RECO/MC_31X_V3-v1
  - 100K events
- QCD_Pt_30
  - /store/mc/Summer09/QCD_Pt30/GEN-SIM-RECO/MC_31X_V3-v1/
  - 100K events
- QCD_Pt_80
  - /store/mc/Summer09/QCD_Pt80/GEN-SIM-RECO/MC_31X_V3-v1/
  - 100K events

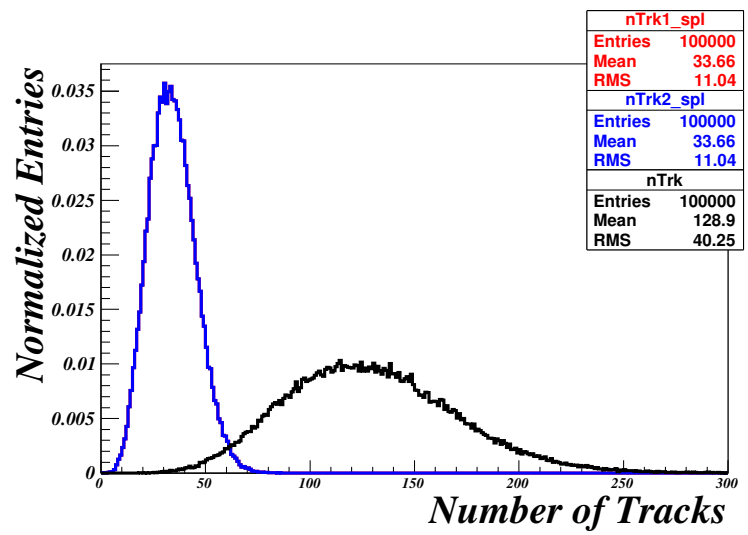
Resolutions vs Number of Tracks - TTbar

- MC methods Gaussian fit σ of $(x_{rec} - x_{sim})$
  - Black: PVTX using all tracks – PVTX using splitTracks1 – PVTX using splitTracks2
- Two-Vertices method: (– Green) Gaussian fit σ of $(\frac{x_{rec1} - x_{rec2}}{\sqrt{2}})$





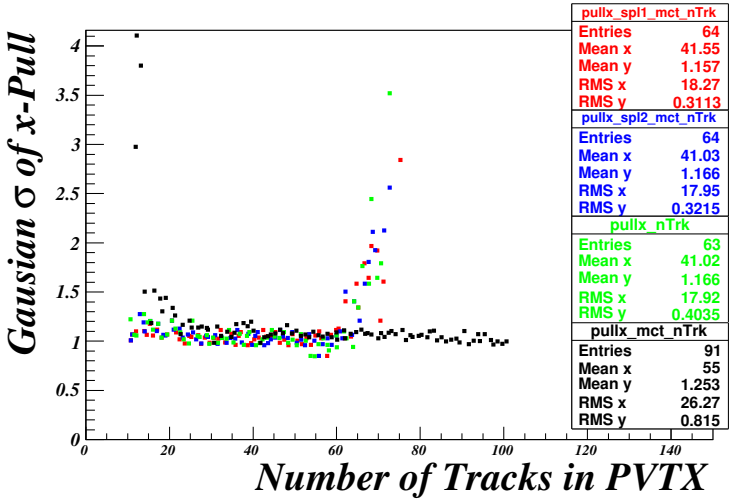


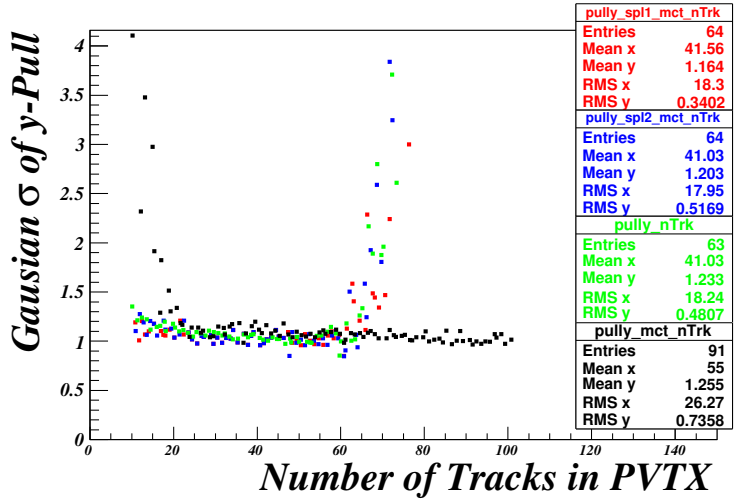


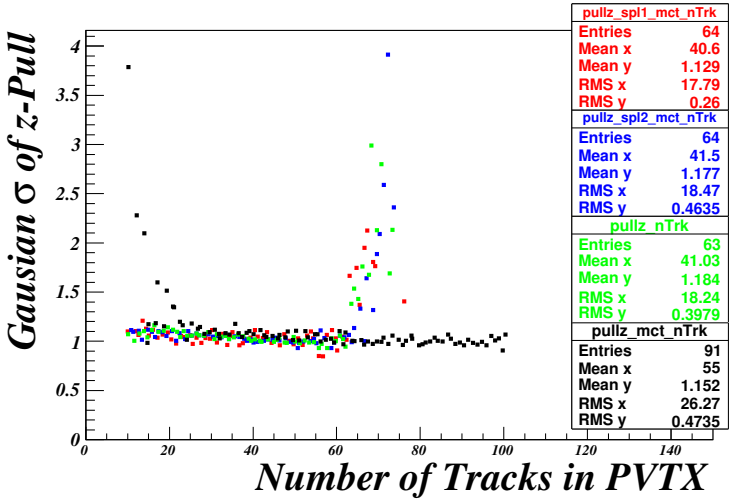
- Resolutions in all methods are consistent

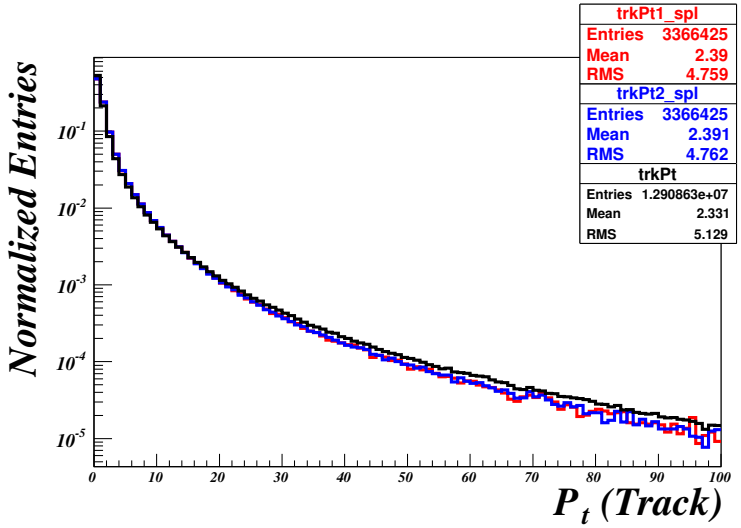
Pull vs Number of Tracks - TTbar

- MC methods: Gaussian fit σ of $\left(\frac{x_{rec}-x_{sim}}{\sigma x_{rec}}\right)$
  - Black: PVTX using all tracks – PVTX using splitTracks1 – PVTX using splitTracks2
- Two-Vertices Method: (– Green) Gaussian fit σ of $\left(\frac{x_{rec1}-x_{rec2}}{\sqrt{\sigma x_{rec1}^2+\sigma x_{rec2}^2}}\right)$





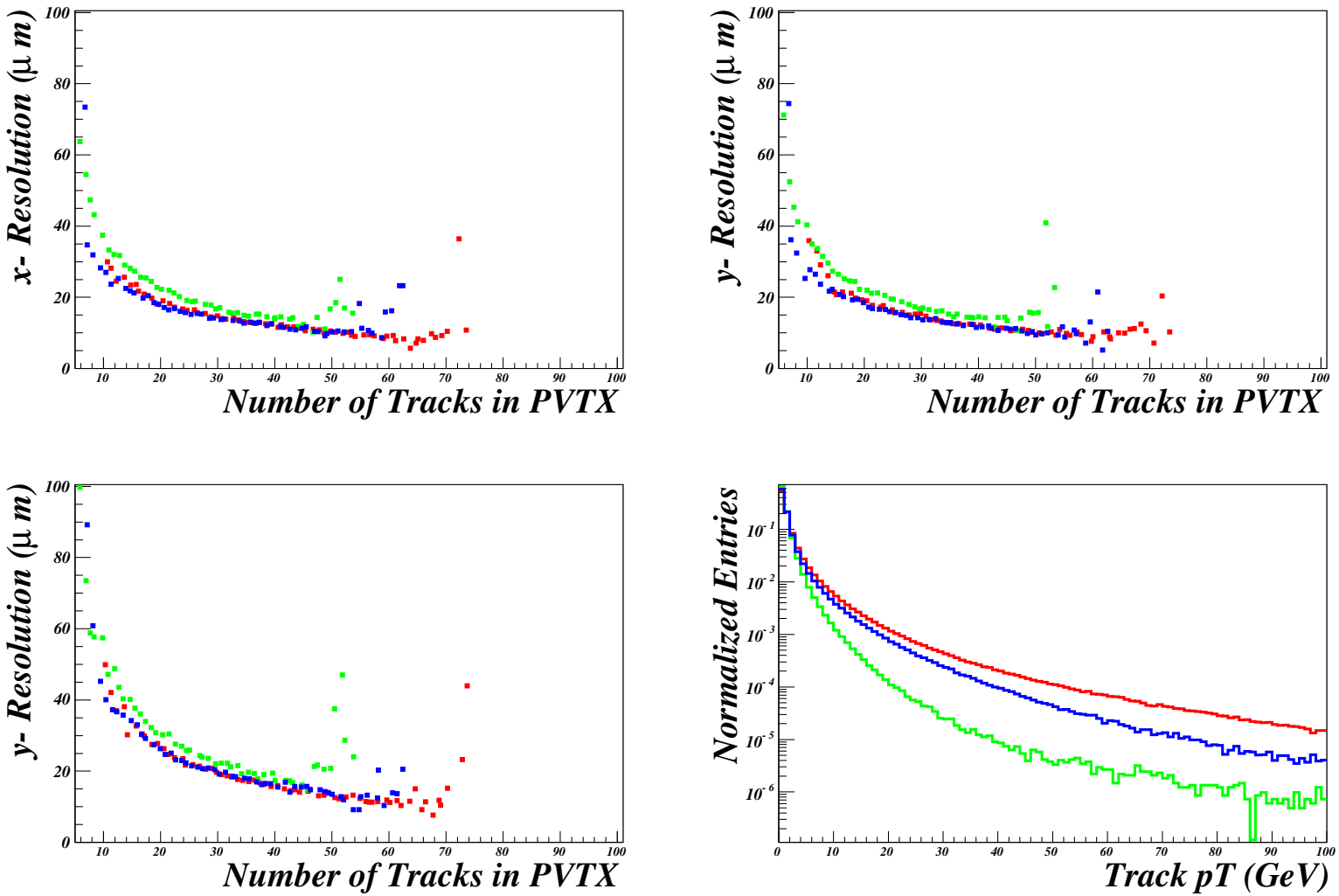




- Pulls in all methods are consistent

Compare Resolutions in Different Samples

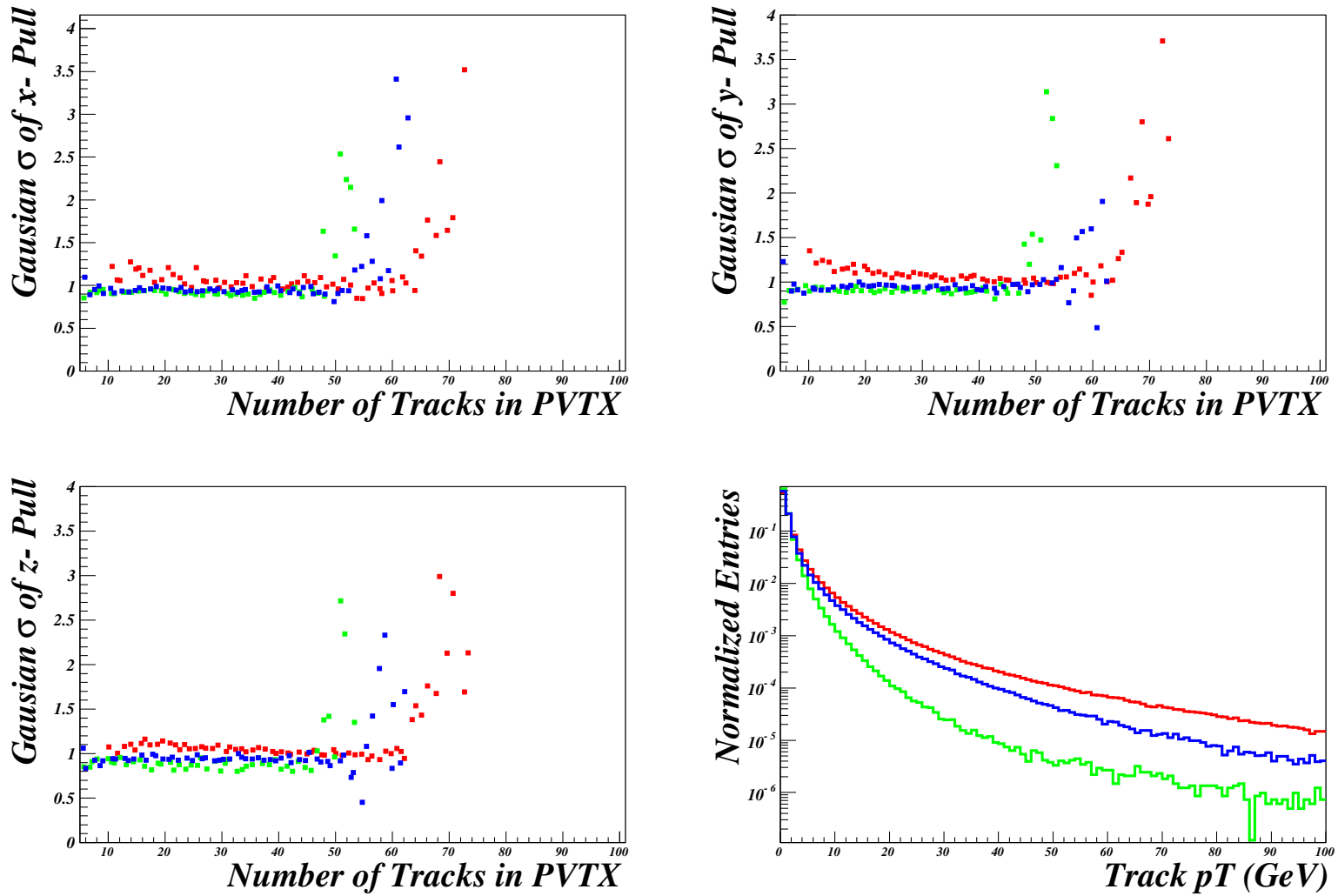
- Two-Vertices Method: Gaussian fit σ of $(\frac{x_{rec1}-x_{rec2}}{\sqrt{2}})$
  - TTbar – QCD_Pt_30 – QCD_Pt_80



- nTracks> 15: $\sigma x/\sigma y < 20\mu m$, $\sigma z < 30\mu m$
- QCD_Pt30 resolution is slightly higher due to the lower p_T

Compare Pulls in Different Samples

- Two-Vertices Method: Gaussian fit σ of $\left(\frac{x_{rec1}-x_{rec2}}{\sqrt{2}}\right)$
  - TTbar – QCD_Pt_30 – QCD_Pt_80



- Gaussian fit $\sigma \approx 1.0$

Summary

- CMS uses Adaptive filter vertex fitter for primary vertex fitting
- Implemented ''two-vertices'' method to measure the pvtx resolution
  - Work flow: splittracks−run pvtxFitter twice−Compare recVertex
  - Validation done with no PU MC TTbar, QCD_Pt_30/_80 samples
  - Resolution with two-vertices method is consistent with MC method

    nTracks> 15: $\sigma x/\sigma y < 20\mu m$, $\sigma z < 30\mu m$
  - Pull with two-vertices method is consistent with MC method

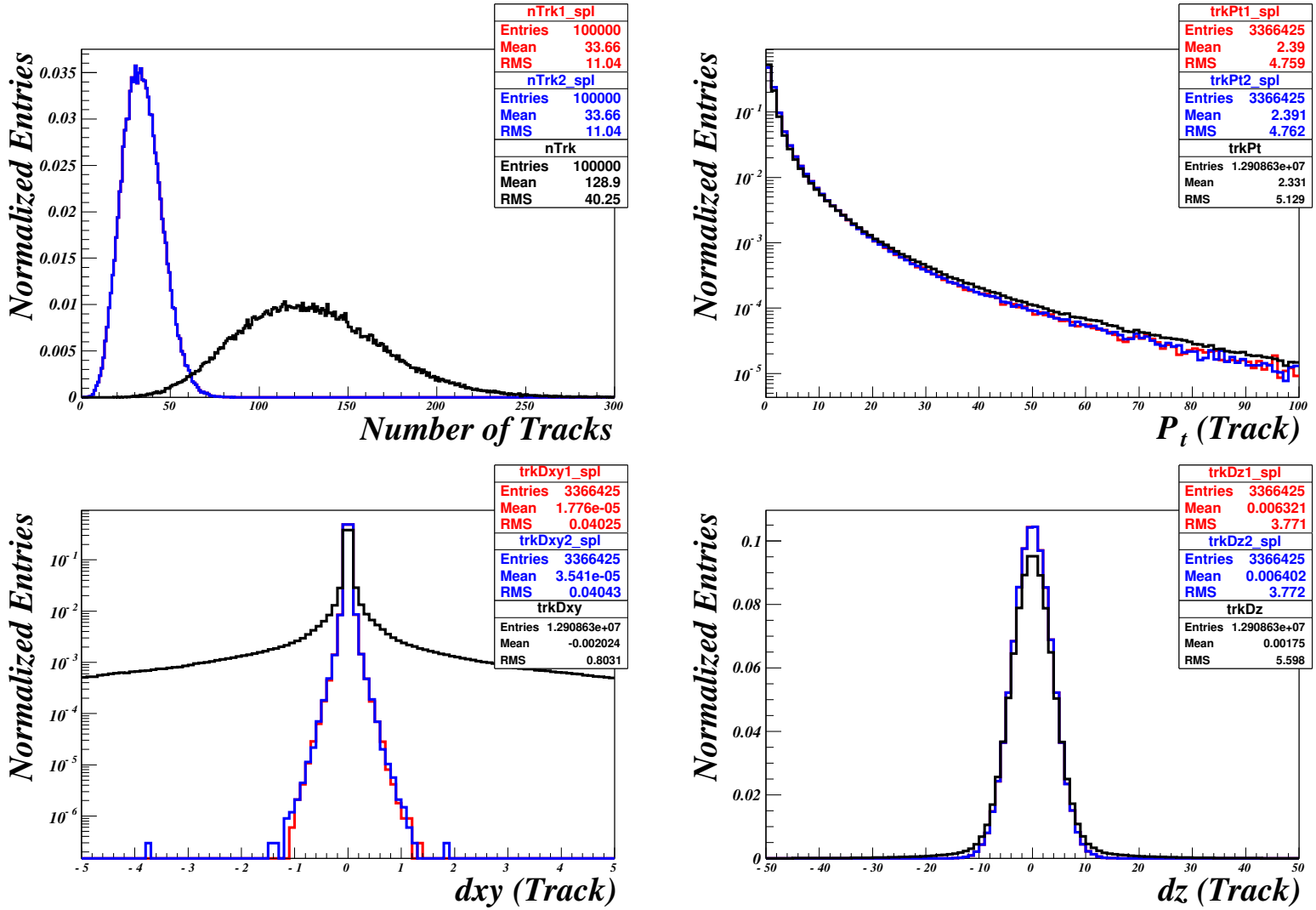
    Gaussian fit $\sigma \approx 1.0$
- On-Going Activies
  - Improve the fitting to the low statistics nTrk bins
  - Study track-splitting for data with PU
  - Test the PrimaryVertexFitter with BS effect on the results

Backup Slides

- Consistency check between the splitted samples
  - Comparing track parameters after splitting
  - Comparing reconstructed vertices
  - Comparing the tracks in reconstructed vertices

Compare Input Tracks to PrimaryVertexFitter

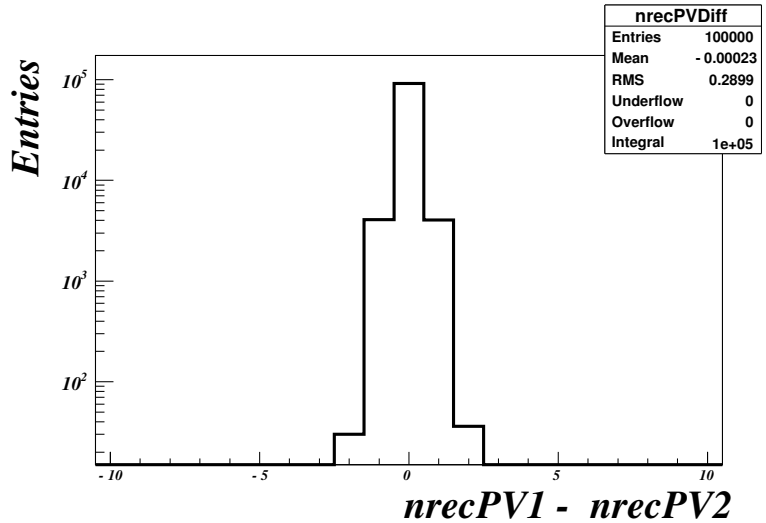
- TTbar Sample:
  - Black: generalTracks, prior to the TrkFilter in pvtx fitter
  - Splittracks 1/– Splittracks 2: after TrkFilter in pvtx fitter



- Track parameters in the split tracksets are consistent

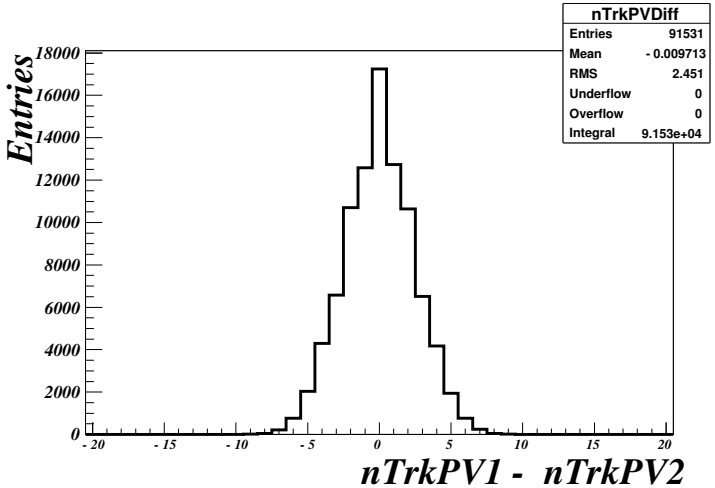
Compare Reconstructed Vertices: TTbar

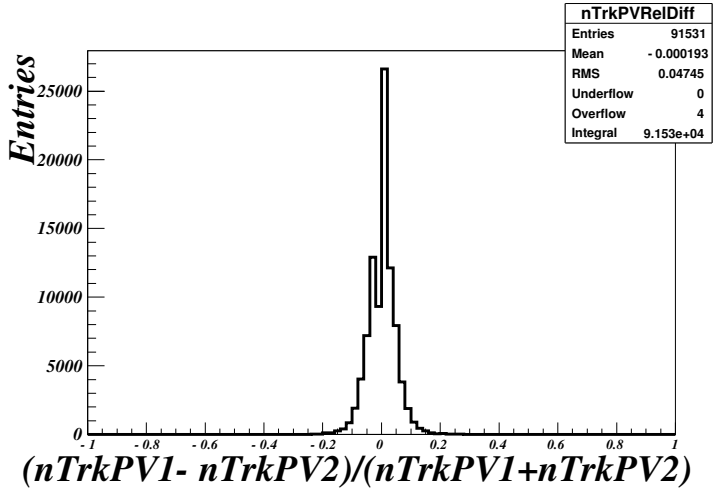
- Number of reconstructed vertices difference: nRecPV1-nRecPV2



- Number of tracks difference in the pvtx

nTrkPV1-nTrkPV2 (Left) $\frac{nTrkPV1-nTrkPV2}{nTrkPV1+nTrkPV2}$ (right)

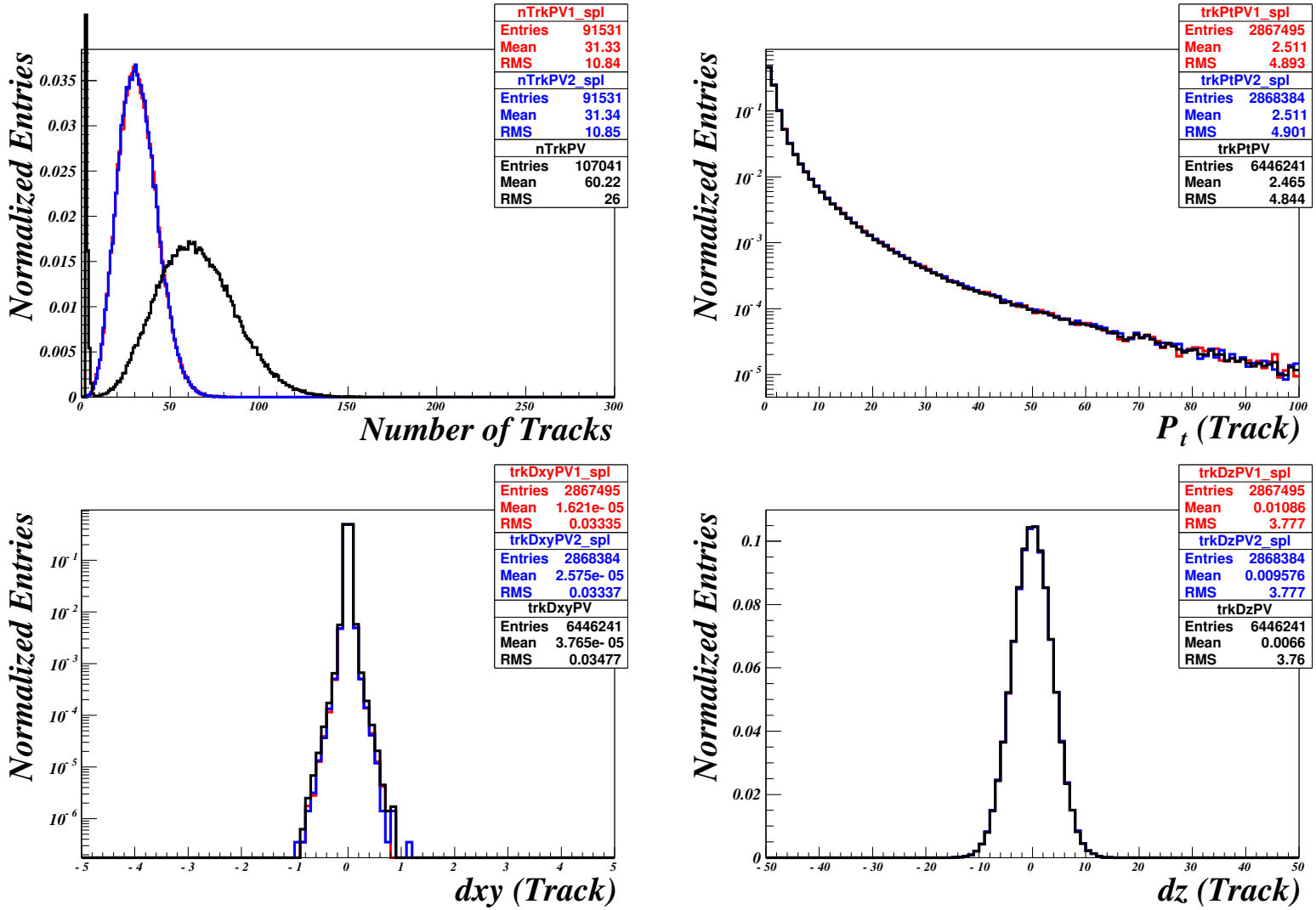




- For the resolution and pulls, select $\left|\frac{nTrkPV1-nTrkPV2}{nTrkPV1+nTrkPV2}\right| < 10\%$

Compare Tracks Used in PrimaryVertex

- TTbar: – Black: PVTX using all tracks – PVTX using splitTracks1 – PVTX using splitTracks2



- Track parameters used in vertices are consistent