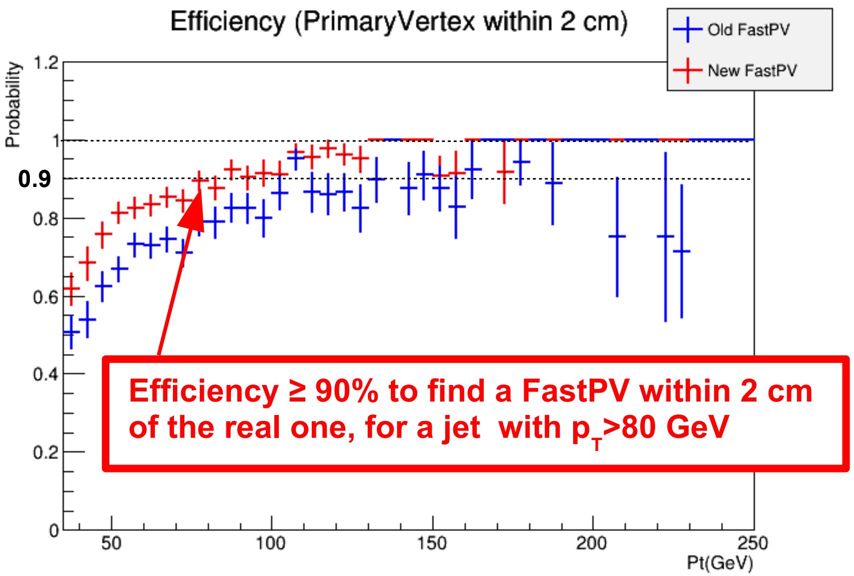
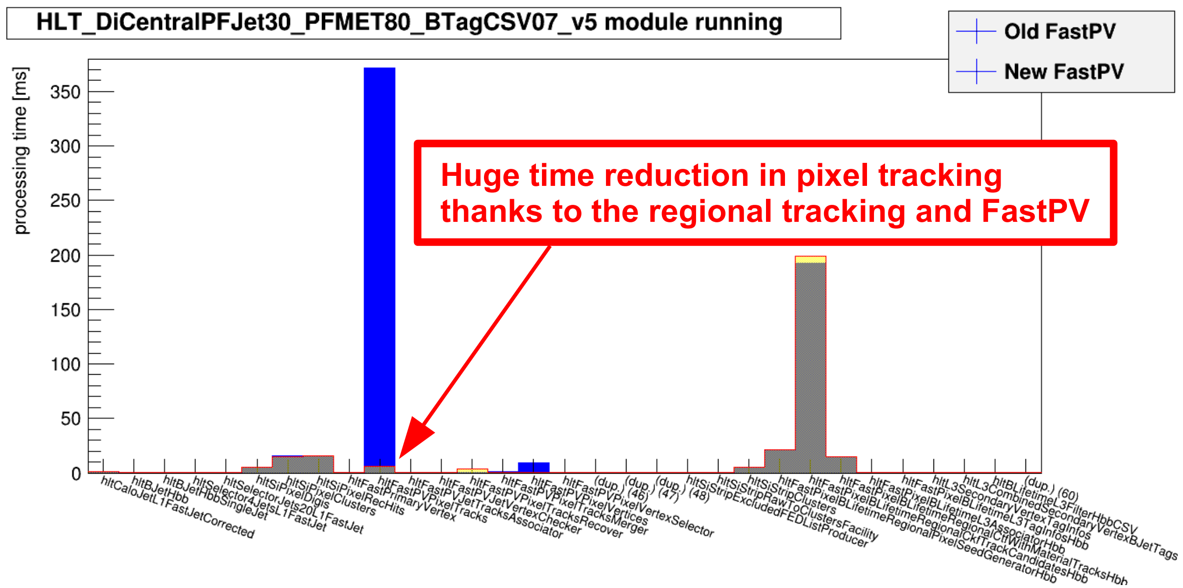
Fast vertexing and tracking @ HLT in multijet and MET trigger paths

Andrea Rizzi, Silvio Donato
(University, INFN and Scuola Normale Superiore di Pisa)

Previous results

- ✔ The last results we presented in TSG meeting are (https://indico.cern.ch/getFile.py/access?contribId=2&resId=0&materialId=slides&confId=259619):
  - ✔ an improvement of the FastPixelVertex algorithm → **more efficiency to find the FastPV**;
  - ✔ an useful exploitation of the pixel regional tracking with FastPV constraint → **very fast tracking without lose efficiency**.



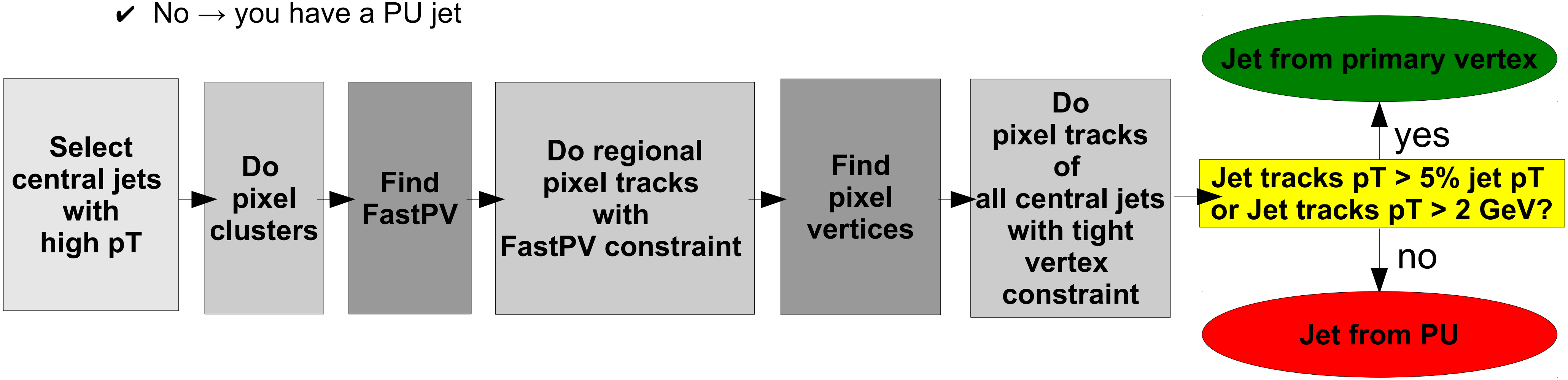


Some applications

- ✔ In Z(vv)H(bb) sample with <PU>~60 **we can find the primary vertex within 50 µm in ~87.1% times, in less then 50 ms** (running time, after CaloJets modules).
- ✔ So we propose to use the pixel regional tracking also to define a **JetNoPU tagging** and use it to reduce the rate in **multi-jet trigger**.
- ✔ We can exploit the JetNoPU tagging also to calculate a kind of "**MHTnoPU**" for **MET trigger** to improve the performances.

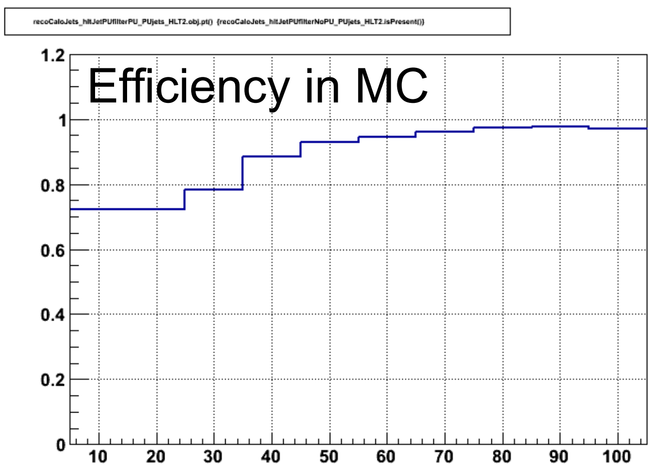
JetNoPU tagging

- In order to tag PU jets we run the following sequence:
  - **select the jets with higher Pt** (eg. Pt>40) and central (|eta|<2.4);
  - do the pixel clusters;
  - find the **FastPrimaryVertex** (with a resolution of ~1.5 cm);
  - do the **regional pixel tracks** around the jets and with the FastPV constraint (1.5 cm);
  - find the **pixel vertices** (with a resolution of ~50 μm);
  - do the **regional pixel tracks of all jets** with the pixel vertex constraint (0.5 cm);
  - **select tracks with a distance from jets direction smaller then 450 μm**;
  - the **sum of jet tracks momentum is greater then** 5% of jet momentum OR then 2 GeV?
    - Yes → you have a no PU jet.
    - No → you have a PU jet

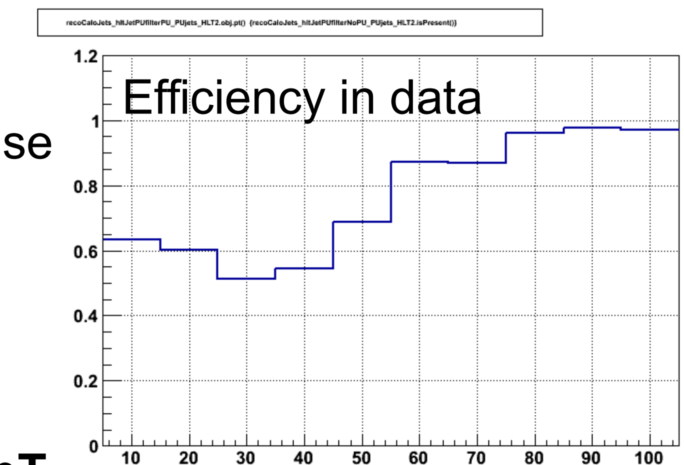


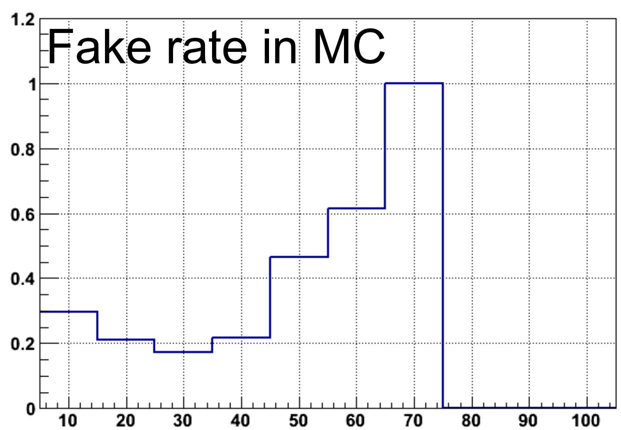
JetNoPU efficiency

- ✔ We tested the JetNoPU in data (L1QuadCJet40) and MC (Z(vv)H(bb)), both with <PU>~60.
- ✔ We use as signal PFnoPU jets in data and GenJets in MC



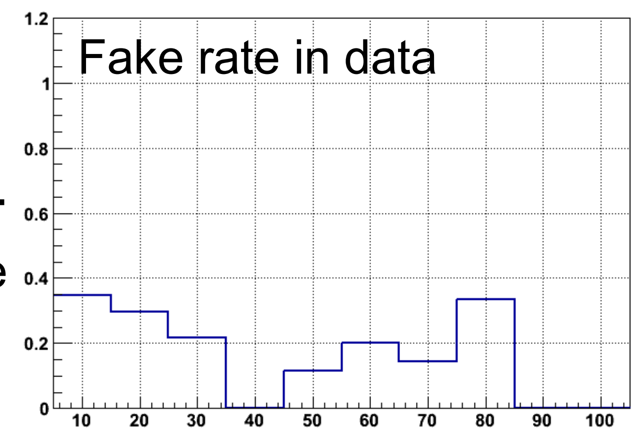
High efficiency in the **high pT region**, because we have more tracks.





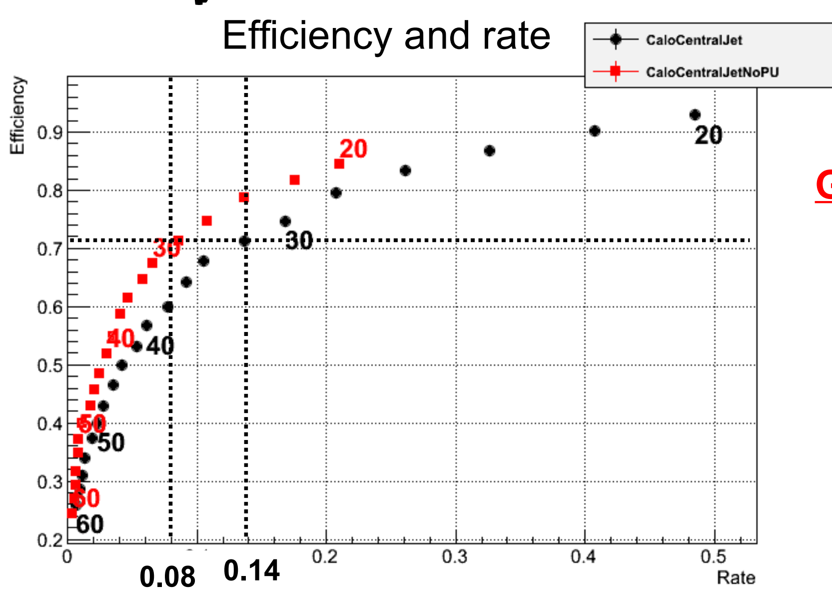
More fake rate in **low pT region,** because we cut with looser threshold (jet tracks pT > 5% jet pT)

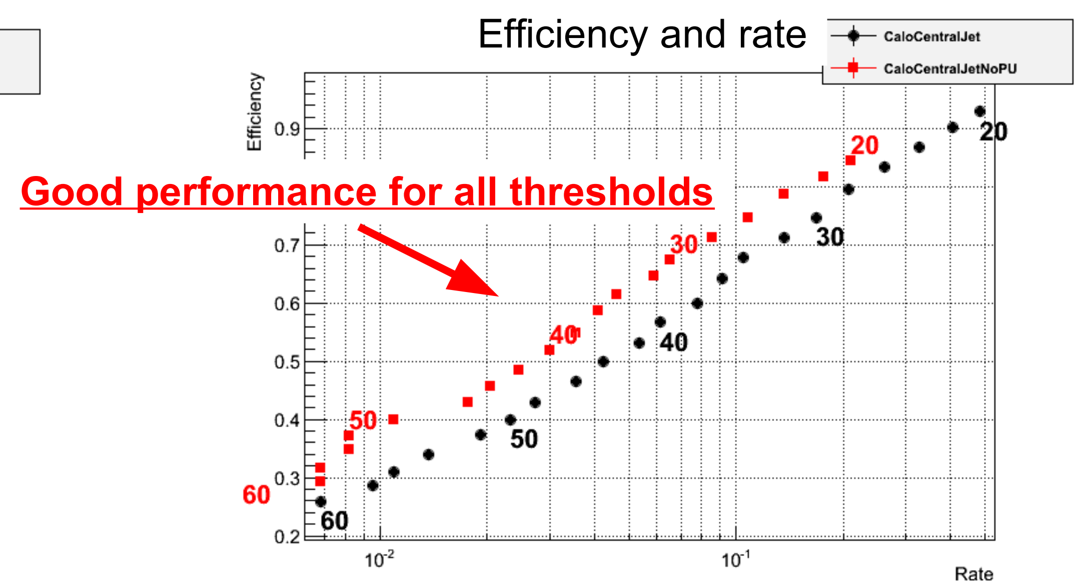
More fake rate in **high pT region** simply because we consider the vertex of the leading jet as primary.



MultiJet trigger

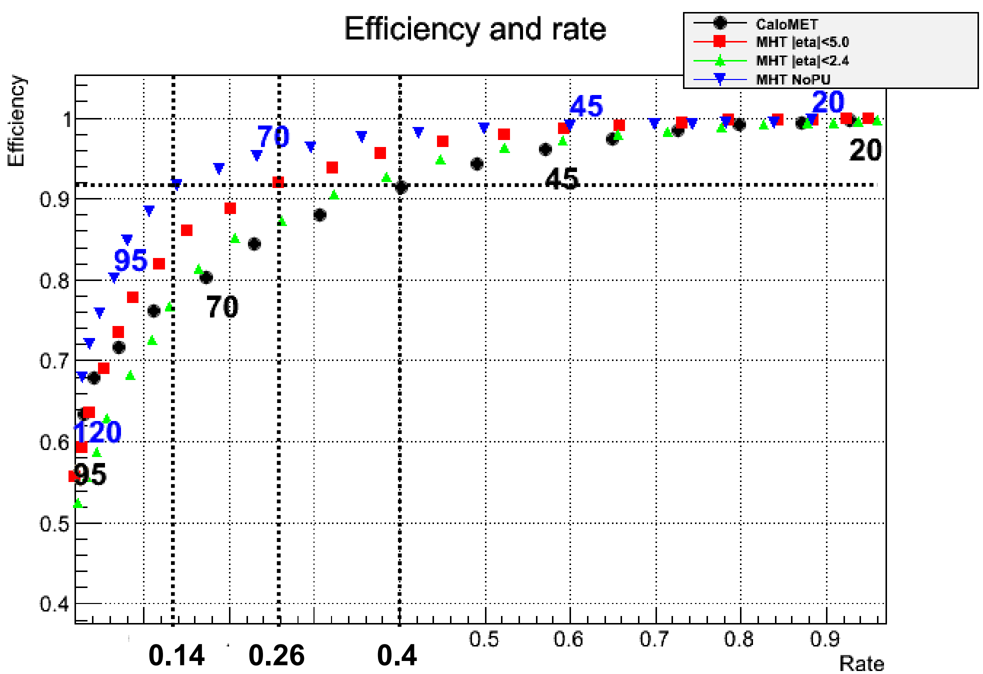
- ✔ Let's see how it works in a multi-jet trigger.
- ✔ We use MinBias data to measure the rate and RadionToHH_4b_M-450_TuneZ2star_8TeV as signal to measure the efficiency.
- ✔ Both have been preselected with L1DoubleJetC64 and requiring two central CaloJet with pT>64GeV. The mean PU is 60 for both data and MC.
- ✔ The plots show the efficiency and the rate, normalized after the pre-selection, as a function of the threshold of QuadCentralCaloJet and QuadCentralCaloJetNoPU cuts.
- ✔ **Thanks to JetNoPU we can reduce the rate up to a factor two, keeping the same efficiency!**

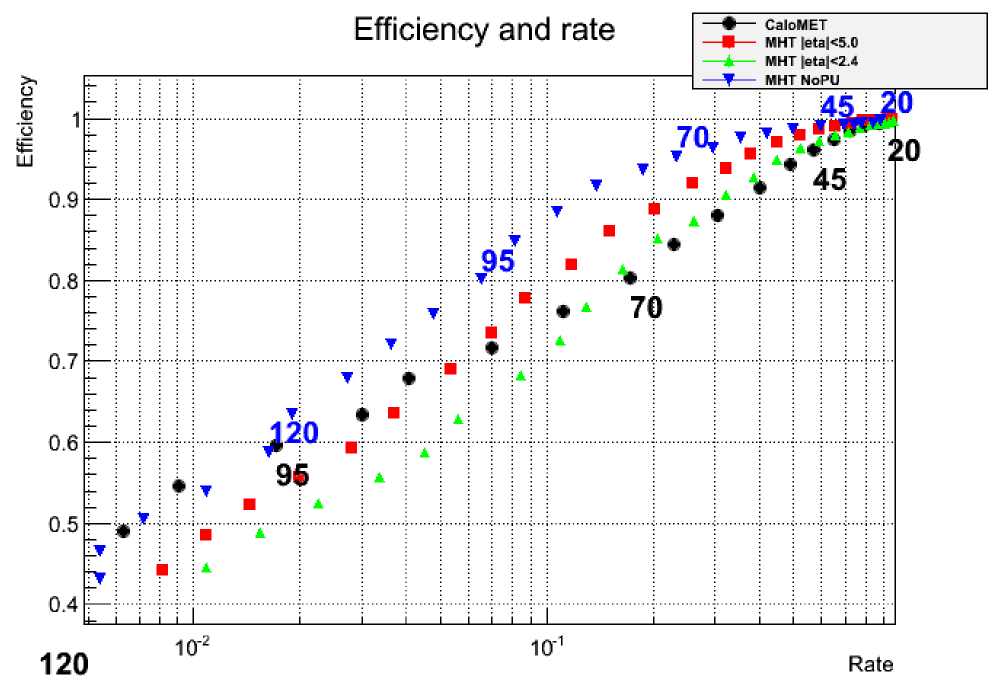




MET trigger

- ✔ For the MET trigger test we use MinBias data to measure the rate and Z(vv)H(bb), with GenMET>80 GeV, as signal to measure the efficiency.
- ✔ Both have been preselected with L1ETM40 and requiring two central CaloJet with pT1>40GeV and pT2>20GeV. The mean PU is 60 for both data and MC.
- ✔ The plots show the efficiency and the rate, normalized after the pre-selection, as a function of the threshold of MET, MHT (|eta|<2.4), MHT (|eta|<5) and MHTNoPU cuts.
- ✔ Here, MHT is defined as the module of the vector sum of the jet momentum with pT>20 and in the eta range. MHTnoPU is calculated using JetNoPU with pT>20 in |eta|<2.4 and all jets with pT>40 in 2.4<|eta|<5
- ✔ **Thanks to jetNoPU we can reduce the rate, keeping the same efficiency, up to a factor of two, compared to CaloMHT, or a factor of four, compared to CaloMET**





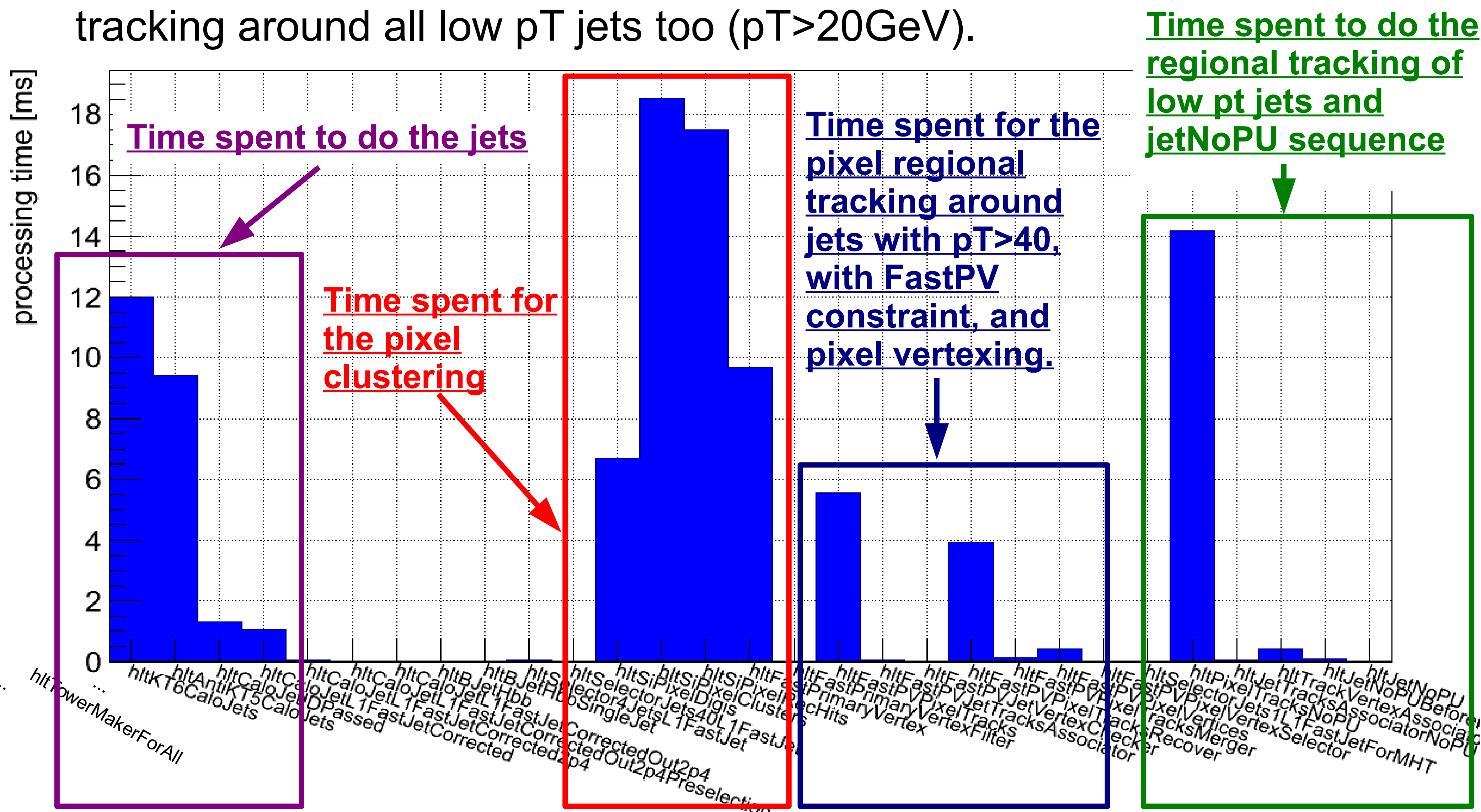
Working points

- ✔ Here we report some working point in order to summarize the results that we can achieve thanks to JetNoPu @ HLT.
- ✔ We compare the rate of the old vs new cut with, with the same efficiency.

Trigger	Old cut	New cut	Old rate	New rate
Multijet	QuadCentJet>32	QuadCentJetNoPU>28	0.14	**0.08**
MET	MET>80 MHT>55	MHTnoPU>80	0.40 0.26	**0.14**

Timing

- ✔ The plot shows the **running time** that we need to evaluate MHTnoPU.
- ✔ This is the worst scenario because we need to do the regional pixel tracking around all low pT jets too (pT>20GeV).



Use case: HLT DiCentralPFJet30 PFMET80 BTagCSV07

- ✔ As use case we propose an update of HLT DiCentralPFJet30 PFMET80 BTagCSV07 trigger path.
- ✔ We use 968 events of Z(vv)H(bb) signal with offline PFMET>100 GeV, 2 b-jet in |eta|<2.4 and pass L1_ETM40.
- ✔ We use 3682 events MinBias data passed L1_ETM40.
- ✔ Both samples have <PU>~60.

Old trigger

✔ The old trigger selection for minbias data and signal is:

Filter	Data	Signal
L1_ETM40	36872	968
MET65	8918	829
DoubleJetC20	6023	800
FastPVSelector	5027	800
Online CSV>0.7	159	566
DoublePFJetC30	120	515
PFMET80	56	482
Ratio(HLT/L1)	0.15%	50%

rate(L1_ETM40) ~ 105 kHz !
if <PU>~60 and bunch spacing 50ns

New trigger

- ✔ The new trigger selection for MinBias data and signal is:

We'll add the DoubleJetC20 filter → (SingleJetC40)

Filter	Data	Signal
L1_ETM40	36872	968
MET40	24920	829
SingleJetC40	7514	829
MHT60	4014	829
FastPVSelector	3016	829
MHTnoPU80	568	800
Online CSV>0.7	43	566
Ratio(HLT/L1)	0.11%	58%

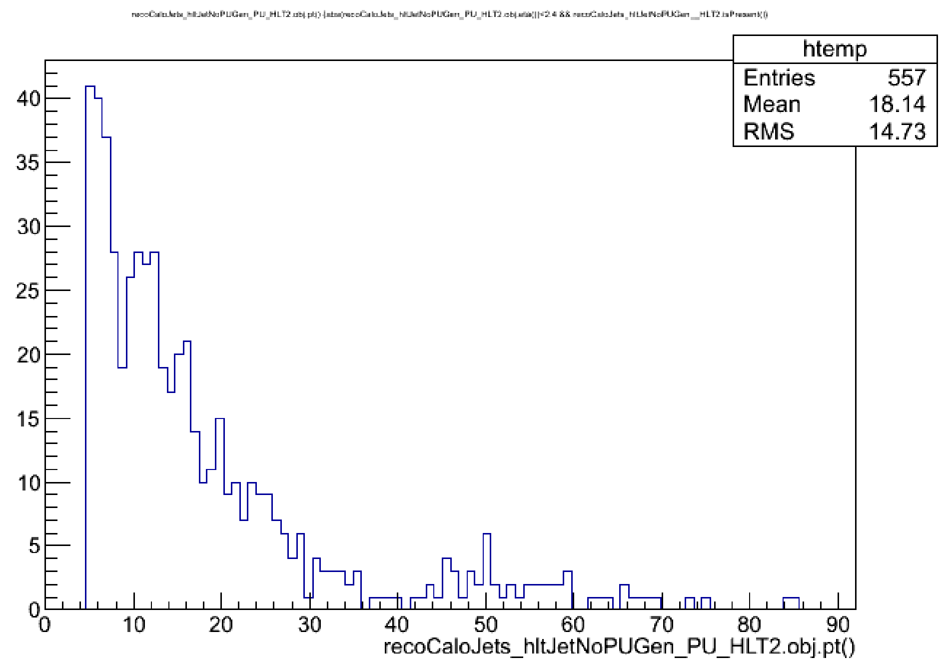
Comparison

- ✔ The new HLT path has:
  - ✔ an **efficiency** on signal of **58%** (instead of 50%);
  - ✔ a **rate reduction** to **0.11%** (instead of 0.15%).
- ✔ The mean time needed to the new trigger path is only **61 ms** on L1_ETM40 preselected data (instead of 153 ms):
  - ✔ Note: to calculate the real HLT timing we should divide it by rate(L1)/rate(L1_ETM40) ~ 17.
- ✔ We get this time reduction thanks to:
  - ✔ a better FastPV efficiency to find the right PV;
  - ✔ the regional pixel tracking;
  - ✔ (removing the PF sequence)

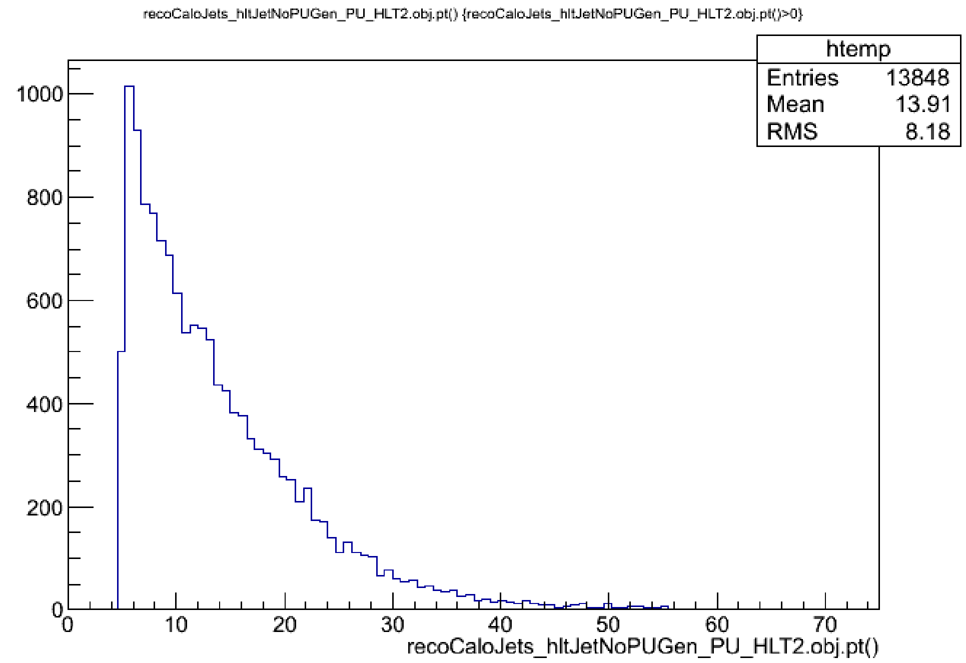
Conclusions

- ✔ FastPV and regional pixel tracking are very fast → we can find the primary vertex in 50 ms with a resolution of ~50μs.
  - ✔ Note: we could further reduce this time using a regional pixel clustering!
- ✔ So we can use them to recognize jet from PU.
- ✔ Using JetNoPU tagging in trigger paths, with the same efficiency, we can reduce the rate:
  - ✔ up to a factor x2 in a QuadCentralJet trigger;
  - ✔ up to a factor x4 in a CaloMET trigger (x2 compared with CaloMHT).
- ✔ An update of HLT DiCentralPFJet30 PFMET80 BTagCSV07 has been studied has use case:
  - ✔ the result is a faster trigger (factor x2),
  - ✔ with an high efficiency (50% → 58%),
  - ✔ and better rate reduction (0.15% → 0.11%)!

Pt distribution of PU jets



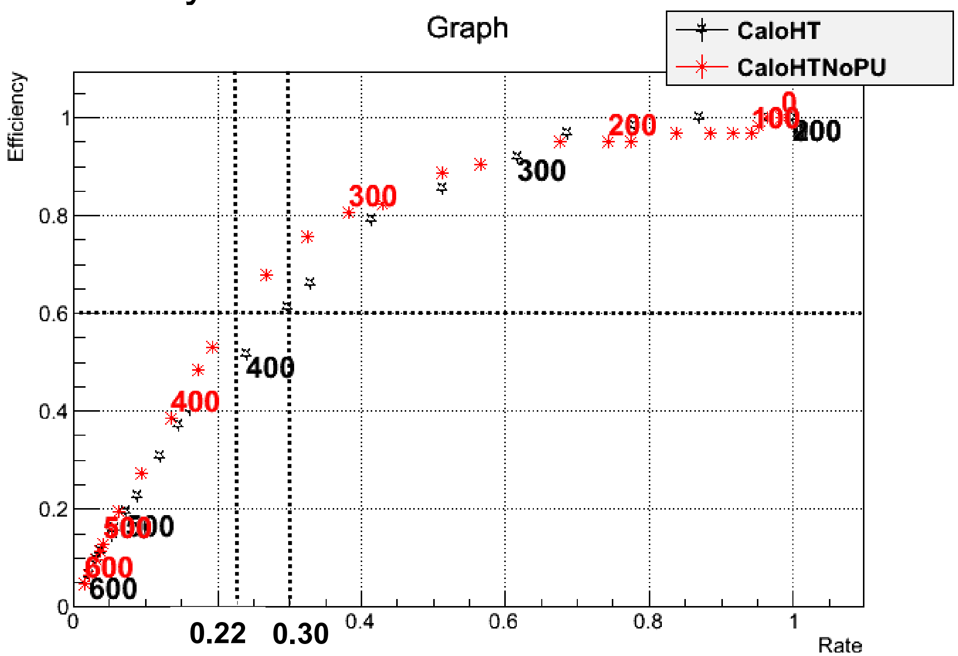
Using 155 events of
Z(vv)H(bb) with <PU> ~ 60

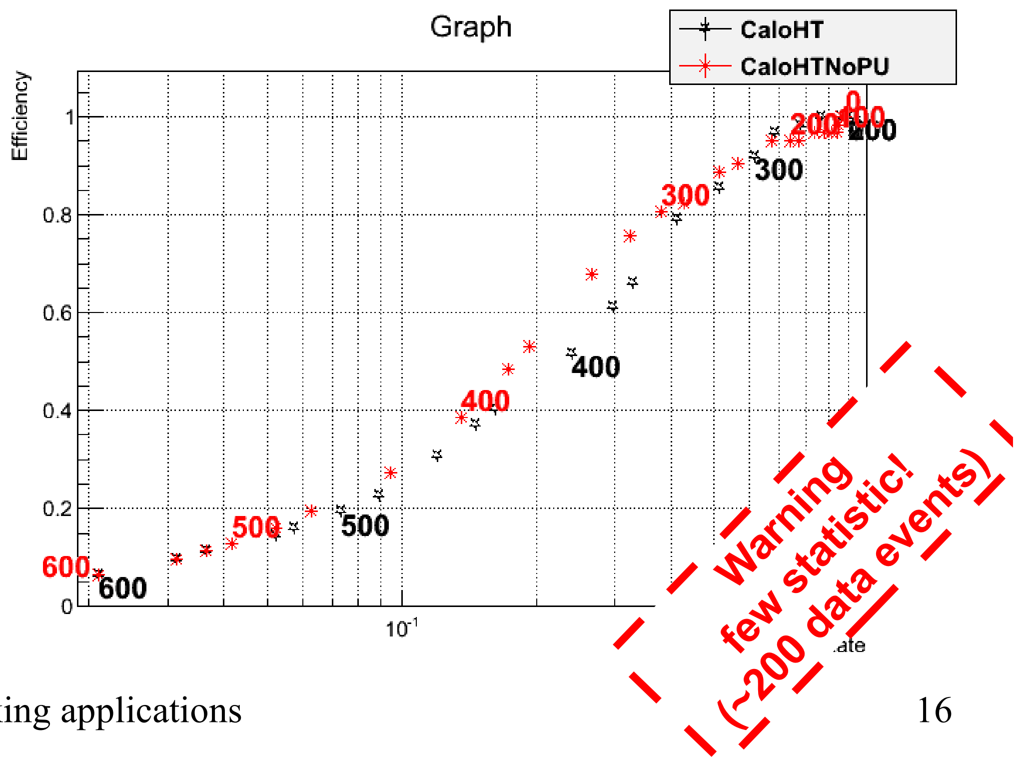


Using 2410 events of
Z(vv)H(bb) with <PU> ~ 60

HT trigger

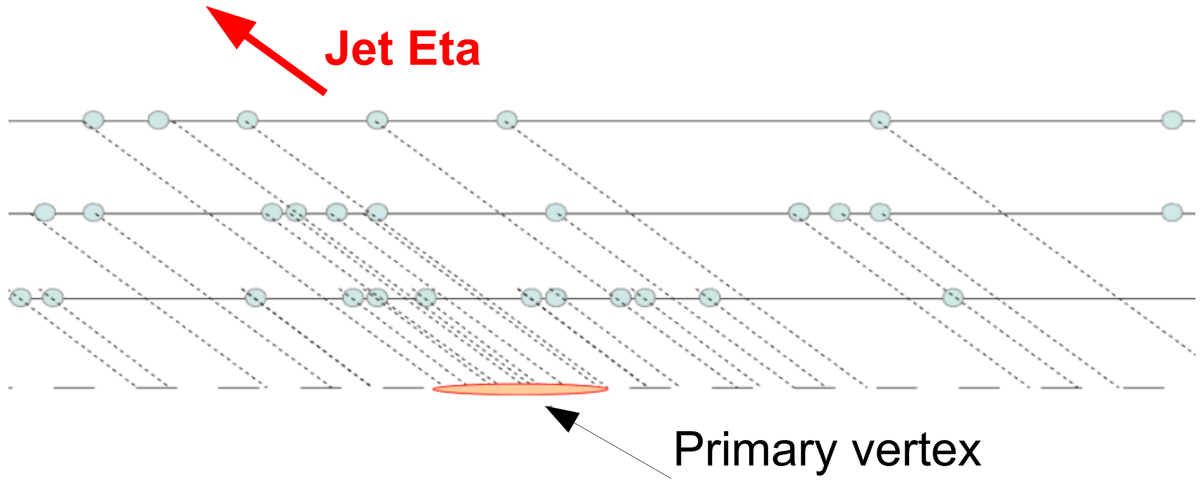
- ✔ Here we use MinBias data, to measure the rate, and MinBias data with offline PFnoPU HT>300 GeV preselected as signal to measure the efficiency.
- ✔ Both samples have <PU>~60 and they are preselected with L1QuadJetC40.
- ✔ Here HT is defined as the scalar sum of the momentum of the jet with |eta|<2.4 and pt>30GeV
- ✔ The plot shows the efficiency and rate, normalized after the pre-selection, as a function of the threshold of CaloHT and CaloHTnoPU.
- ✔ Here we get a small improvement, up to about 30% of rate reduction maintaining the same efficiency.

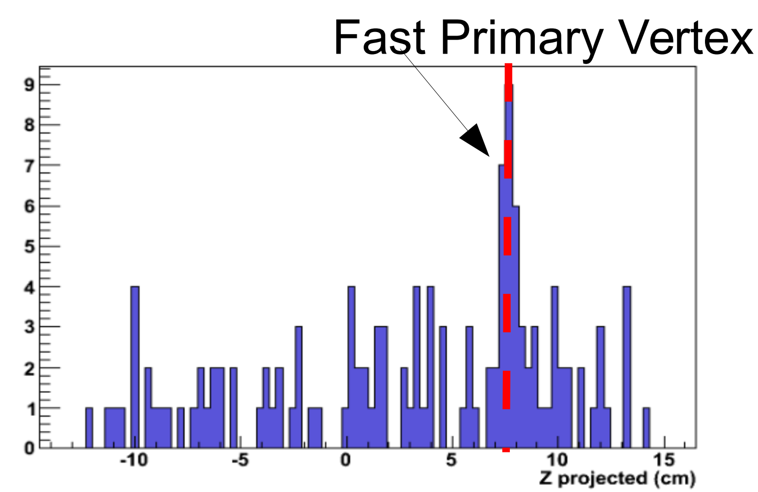




Fast Pixel Primary Vertex

- The Fast Pixel Primary Vertex is an algorithm used to find the Primary Vertex before tracking.
- Given a jet (p_T>40 and |eta|<1.6), the compatible pixel clusters are selected requiring that:
  - the clusters in pixel layers and the jet must to have the same phi coordinate;
  - the cluster sizes along Y have to be compatible with the eta of the jet (jets with high eta have clusters with a long size Y);
  - the cluster sizes X have to be small in order to select only high p_T tracks.
- These clusters are projected along z using the jet eta direction.
- The “peak” in the z projections distribution is the Fast Primary Vertex.

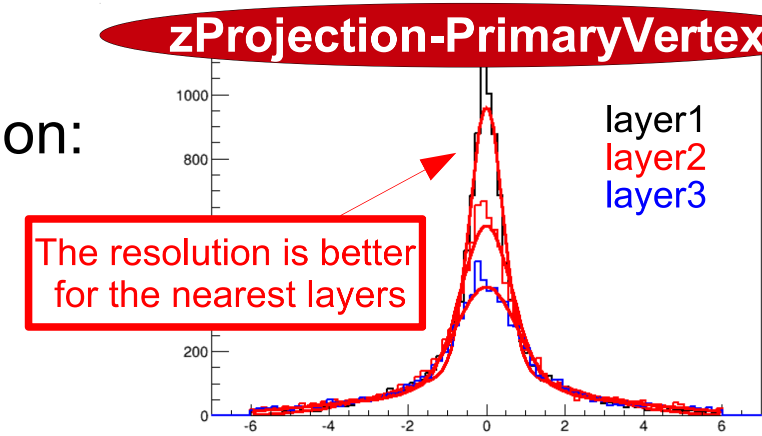




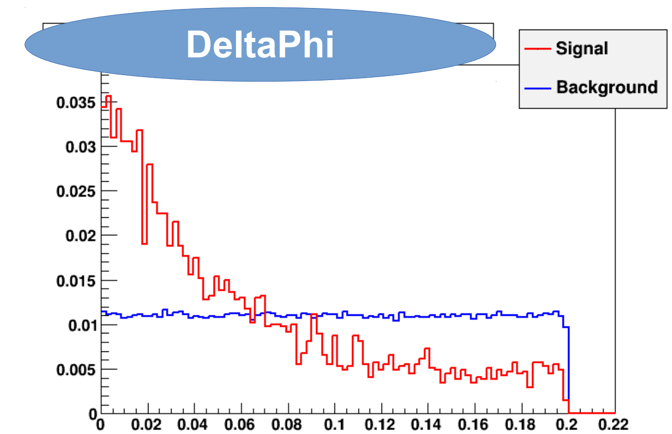
Updates and tests

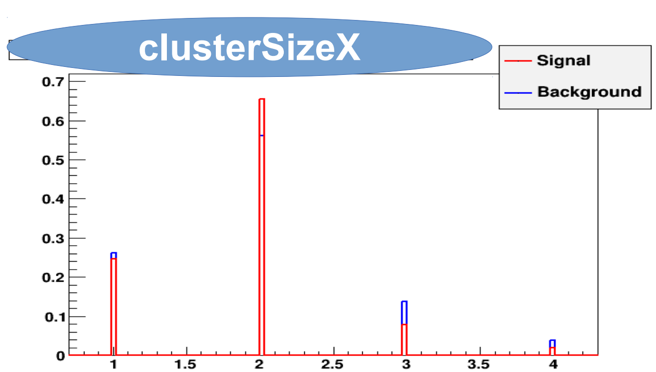
- ✔ We'll show two kinds of updates:
  - ✔ An improved version of the algorithm used to find the FastPV. We propose to:
    - ✔ weight the clusters in order to get better performances;
    - ✔ extend the jet acceptance to |eta|<2.4 , exploiting also the clusters from pixel EndCaps.
  - ✔ A different way to find the Primary Vertex using the Pixel tracks, exploiting the regional tracking and FastPV.

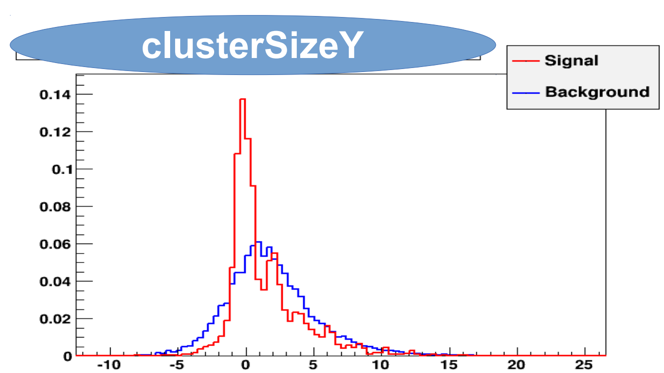
Cluster weight

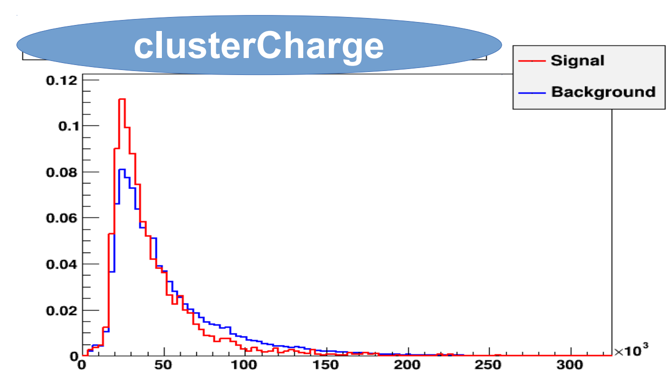


- We propose to weight the clusters depending on:
  - DeltaPhi between cluster and jet;
  - Difference between the cluster size Y and the one expect by jet eta;
  - Cluster rho coordinate;
  - Cluster charge;
  - Cluster size X.
- The distributions of these variables are shown for clusters with a z-projection near (<0.2cm) or far (>2cm) the real primary vertex.







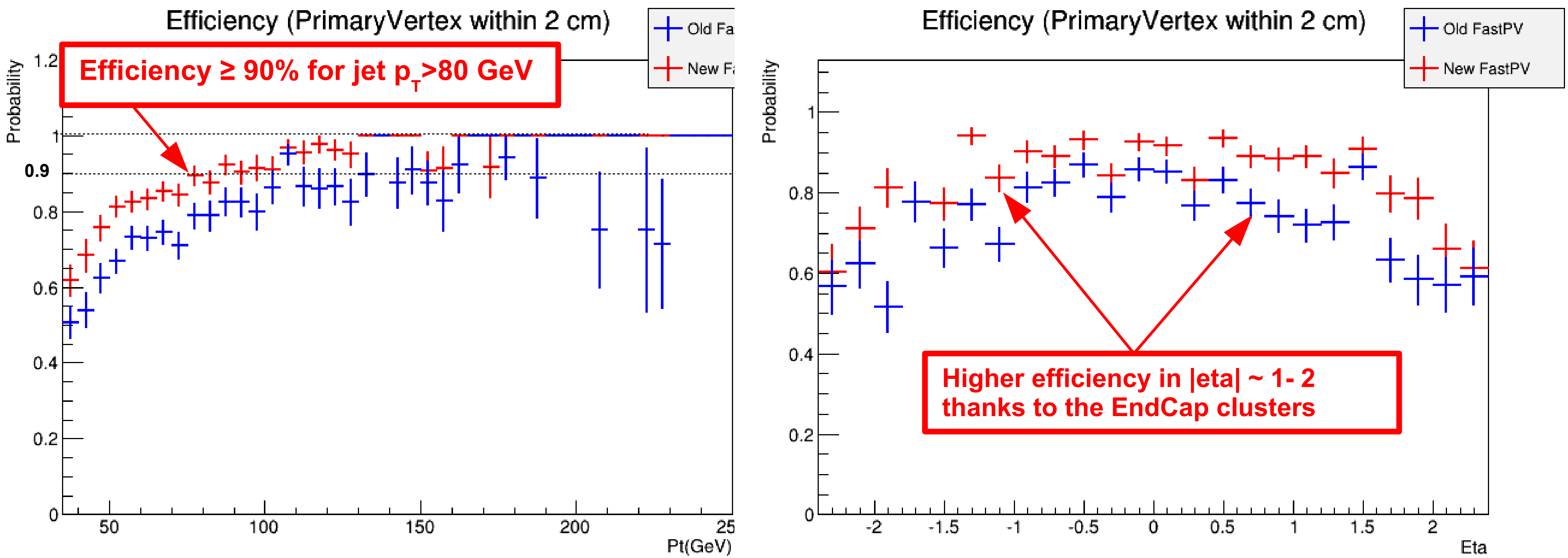


Performance of the new FPV

- ✔ We've measured the performance of the new FastPrimaryVertex using:

  ZH_ZtoNuNu_HtoBB_M-125_8TeV-powheg-herwigpp/Summer12-START50_V13-v3/GEN-SIM (RAW producted using **<PU> ~ 60**)

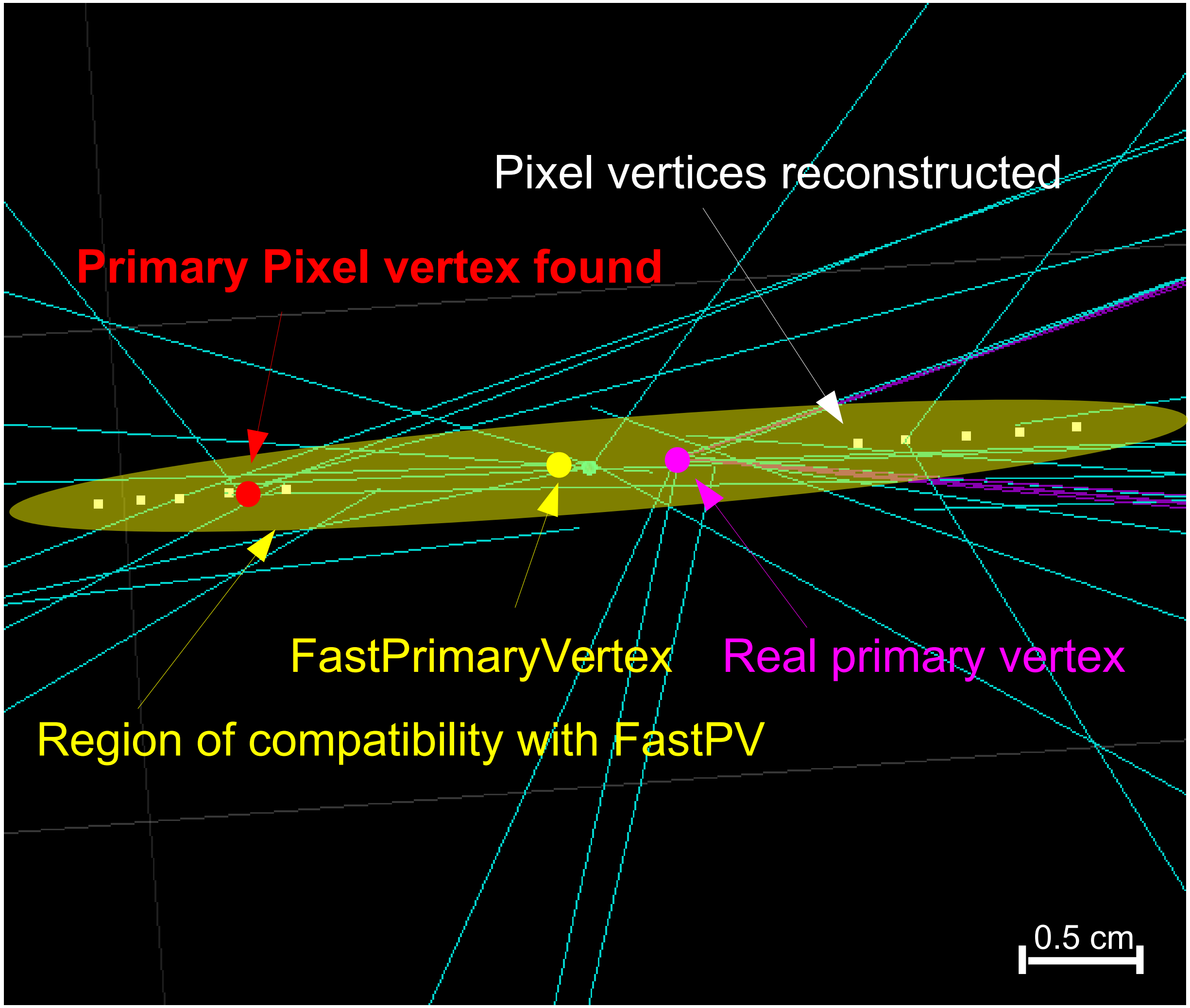
- ✔ Here, we've measured the efficiency to find the PV within an error of 2 cm, using a single jet, as a function of jet pt and jet eta.



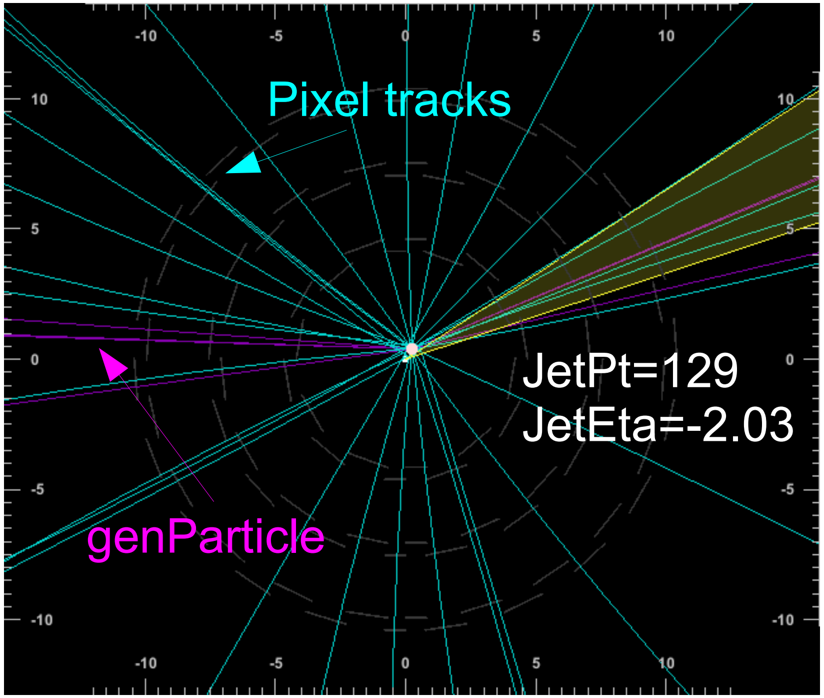
Pixel tracking

- ✔ In the present FastPV sequence we do a total pixel tracking with the FastPixelVertex constraint (1.5 cm).
- ✔ Instead, we propose to do only a regional pixel tracking around selected jets still with the FastPV constraint.
- ✔ The effects are:
  - ✔ A faster tracking! (as expected)
  - ✔ Higher efficiency to find the real Primary Vertex with Pixel tracks. (not so trivial!)
- ✔ Let's see an example.

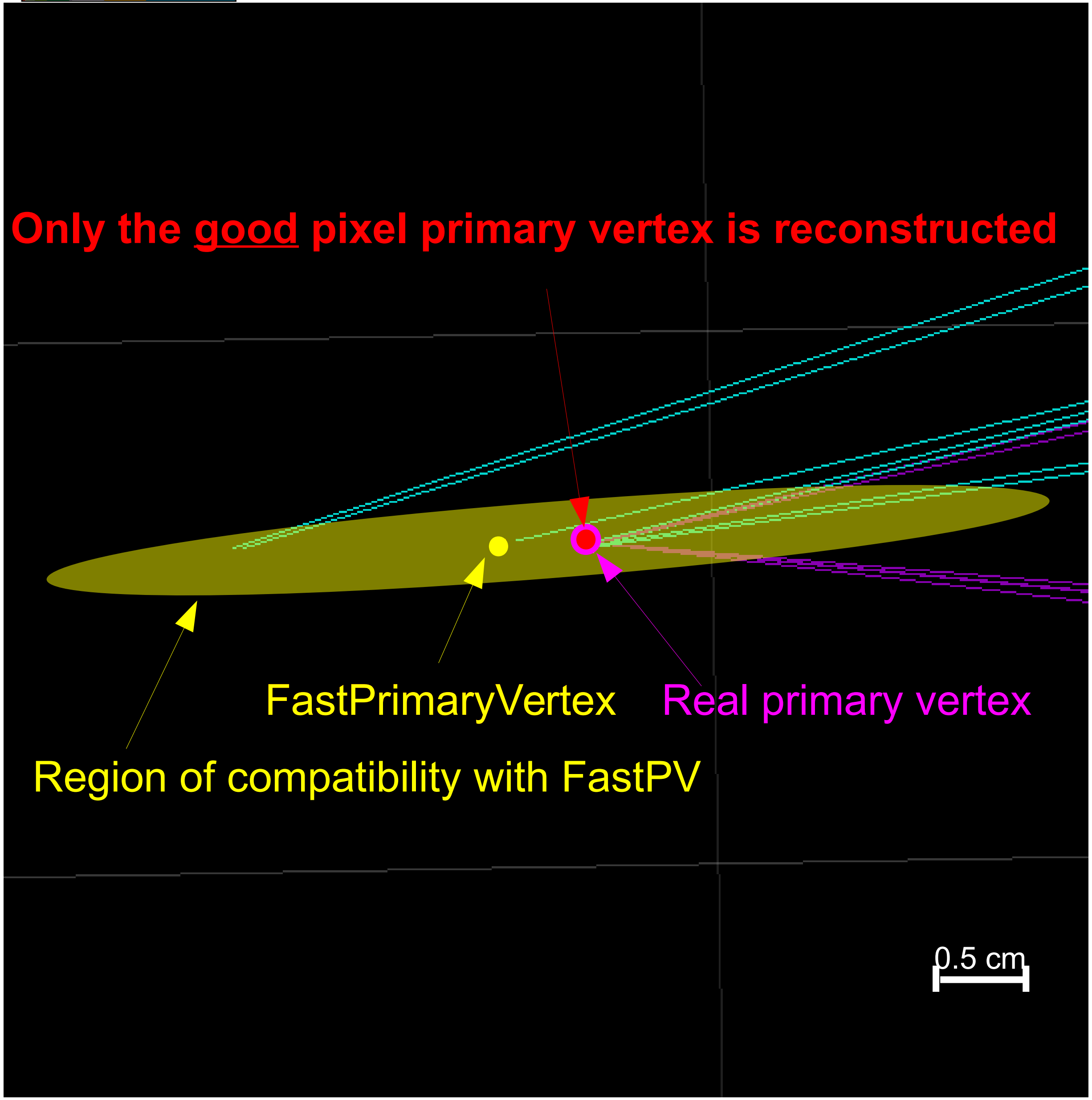
Example: old pixel tracks



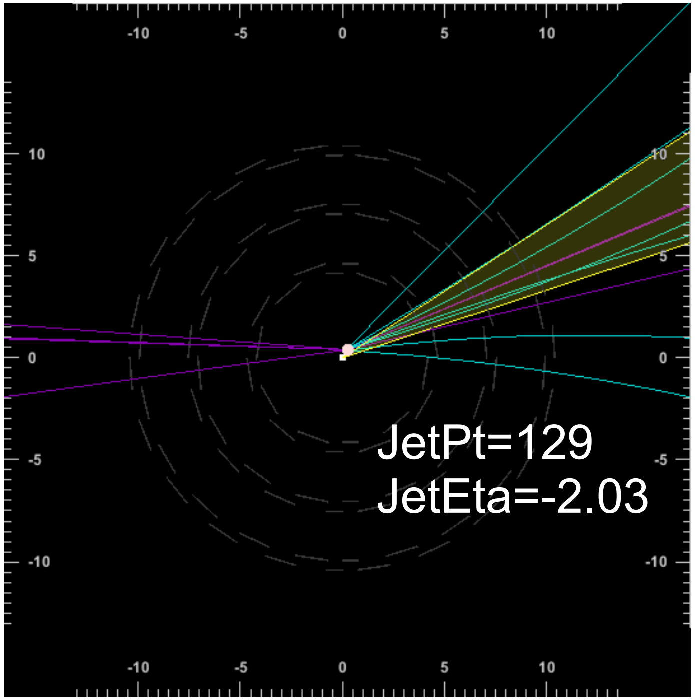
- ✔ All tracks compatible with the FastPV within 1.5 cm are reconstructed.
- ✔ We get the wrong PixelVertex!



Example: new pixel tracks



- ✔ Here, only regional tracks are used (with the FastPV contraint).
- ✔ We get the right PixelVertex!



Test in a HLT path

- ✔ Finally we tested the updates in HLT_DiCentralPFJet30_PFMET80_BTagCSV07 path using:
  - ✔ Data: MinimumBias special run 190782 (<PU> ~ 60)
    - ✔ 36.8k events after L1_ETM40 preselection;
    - ✔ 2.2k events after other calo cuts (MET>80 & jet Pt1>40 & jet Pt2>20).
  - ✔ MC: /ZH_ZtoNuNu_HtoBB_M-125_8TeV-powheg-herwigpp/Summer12-START50_V13-v3 (with <PU> ~ 60)
    - ✔ 1.8k events after L1_ETM40 preselection;
    - ✔ 1.1k events after other calo cuts (MET>80 & jet Pt1>40 & jet Pt2>20).
- ✔ Let's see the results...

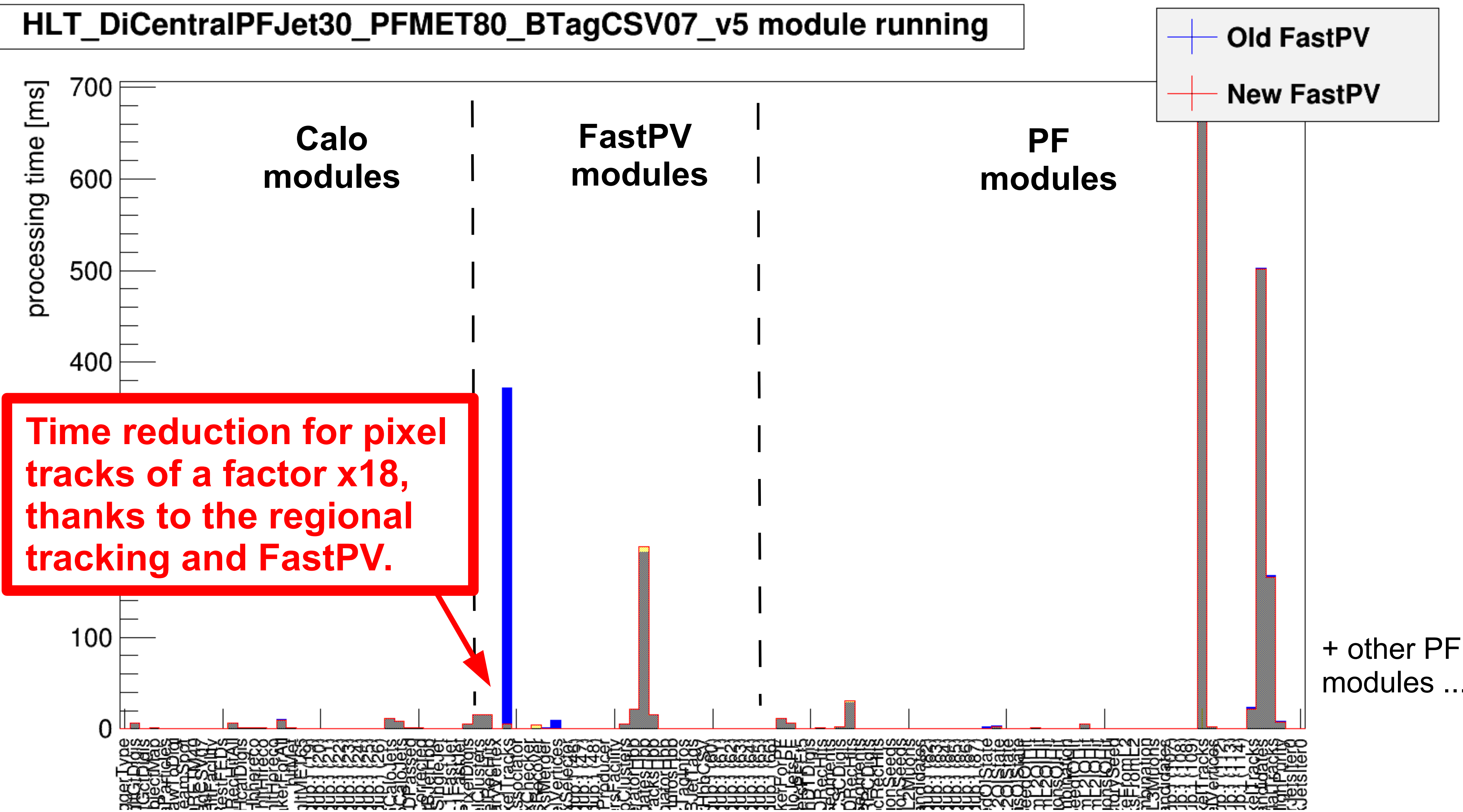
Test on signal

- For signal, the new algorithm finds:
  - Always a FastPV (instead of ~87% of the old one);
  - a good FastPV (<1.5 cm) in **~90% times (instead of ~74%)**;
  - a bad FastPV (>1.5 cm) only in ~10% times (instead of ~13%).
- We can reject events without FastPV without lose efficiency!
- **After tracking, the new algorithm finds the right primary vertex within 50 µm in ~87.1% times (it was ~83.2%).**
- The efficiency after the b-tag filter is ~67.3% (it was ~65.6%).

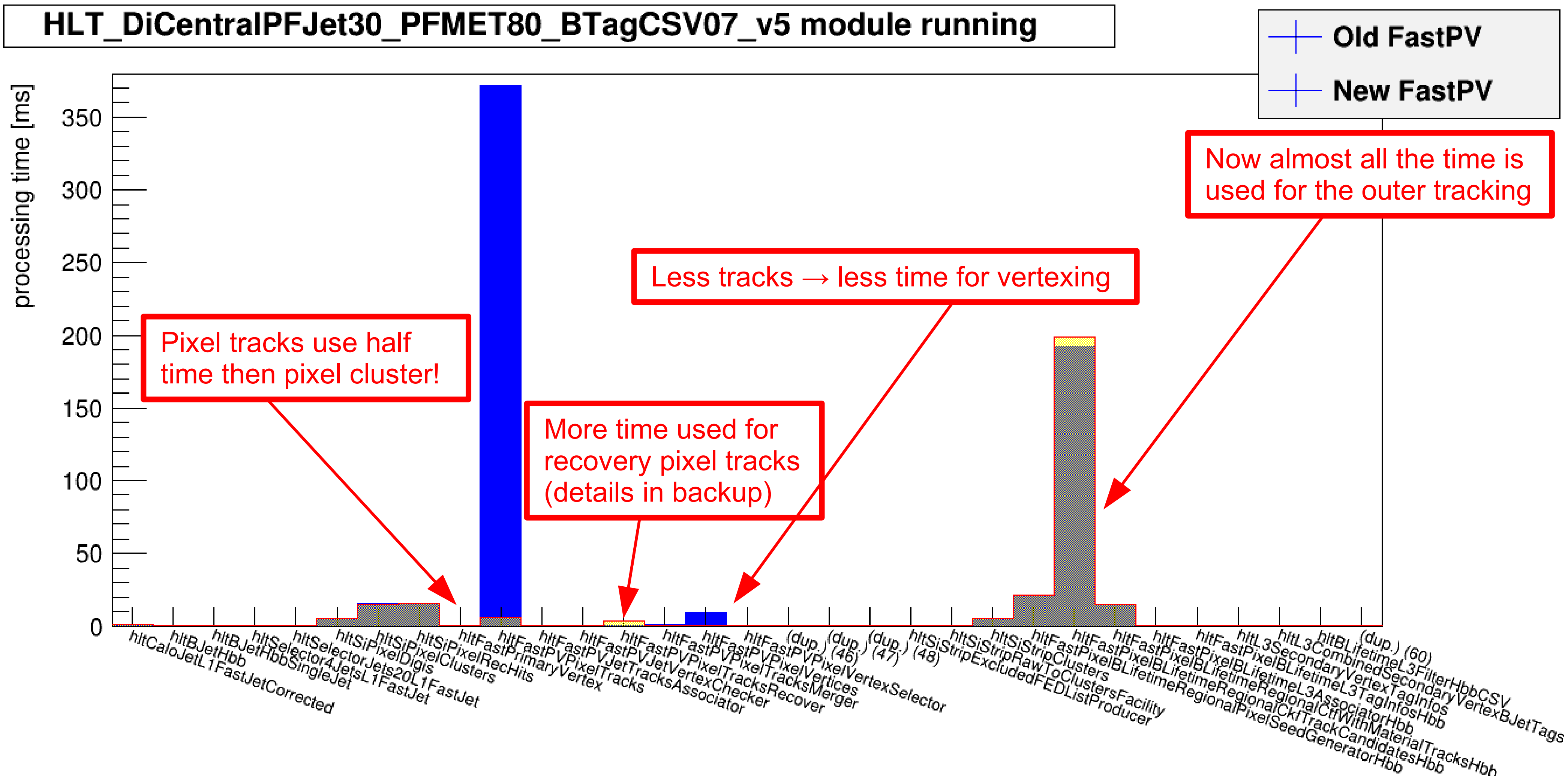
Test on data

- In MinimumBias events, the new FastPV find a primary vertex in about ~82% of events (it was ~77%).
  - So we reject the 18% of events without FastPV.
- The final efficiency for MinimumBias events, after the b-tag filter, is ~3.55% for the new algorithm and ~3.46% for the old algorithm.

Time module running



Time module running zoom



Timing

- ✔ We've used L1_ETM40 preselected events. To estimate the mean time used for a L1-pass events we divide it for rate(L1)/rate(L1_ETM40) ~ 17

- ✔ So the estimated mean time used by the trigger path is:
  - ✔ Calo: 2.1 ms (new Fast PV) vs 2.1ms (old FastPV)
  - ✔ Calo+Btag: 2.8 ms vs 4.3 ms
  - ✔ Calo+Btag+PF: 3.3 ms vs 4.9 ms

We save ~ 1.5 ms!

Conclusions

- ✔ The performance of the FastPrimaryVertex algorithm has been improved using the cluster weightings and exploiting the pixel clusters from EndCaps.
- ✔ The new algorithm has an higher efficiency to find the right primary vertex.
- ✔ The regional pixel tracking give us a faster trigger with more efficiency to find the right Primary Vertex!
- ✔ The new FastPV has been tested in HLT DiCentralPFJet30 PFMET80 BTagCSV07 trigger path using high PU data and simulated signal.
- ✔ **With the new setting we have a small increase of efficiency and rate, and a faster trigger!**

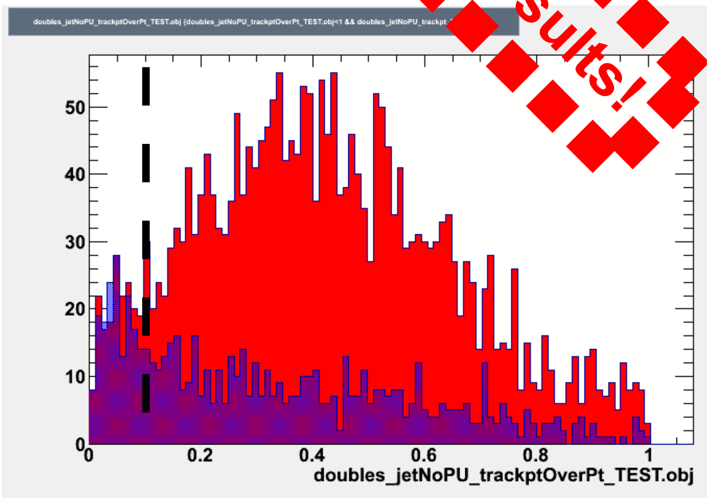
Outlook

- ✔ Recover some inefficiencies due to bad pixel vertexing, after FastPV.
- ✔ Check the tracking in the outer tracker to improve the b-tag efficiency.
- ✔ Do regional pixel cluster instead of the (slow) full pixel clustering.
- ✔ **Exploit FastPV and regional pixel tracking in multijet trigger in order to reject PU jets.**
- ✔ Study of other possible use cases (eg. find hadronic taus, calculate tracking isolation).

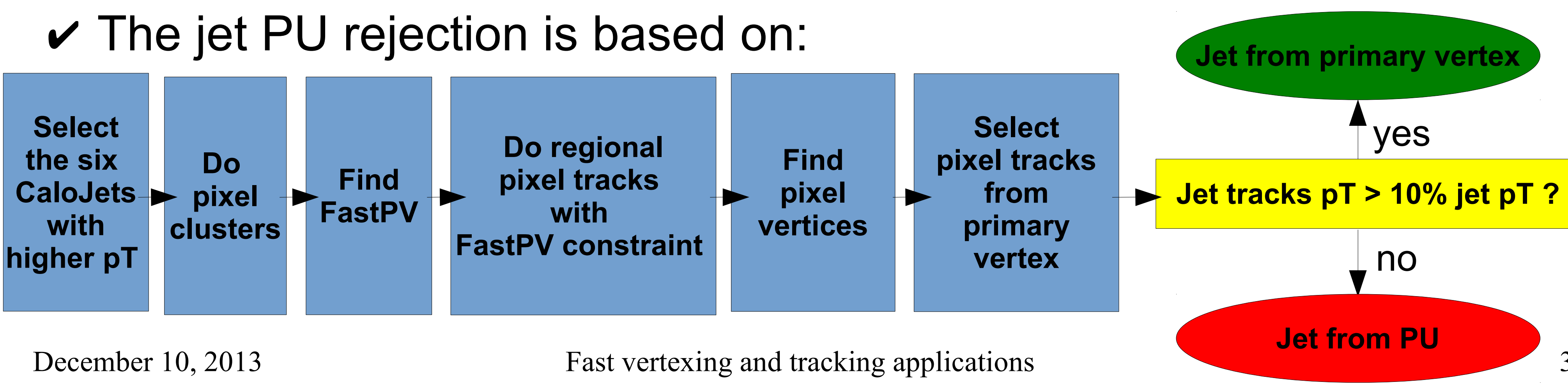
Multijet trigger

Very preliminary results!

- ✔ We tested a first implementation of a trigger with a jet PU rejection based on FastPV and regional pixel tracking.
- ✔ We used RadionToHH_4b_M-450_TuneZ2star_8TeV-Madgraph_pythia6 as signal with **<PU> ~60.**
- ✔ After the preselection of HLT_QuadJet45 (parked), we get a reduction of the rate to the 37% and an efficiency on signal of 83%.



- ✔ The jet PU rejection is based on:

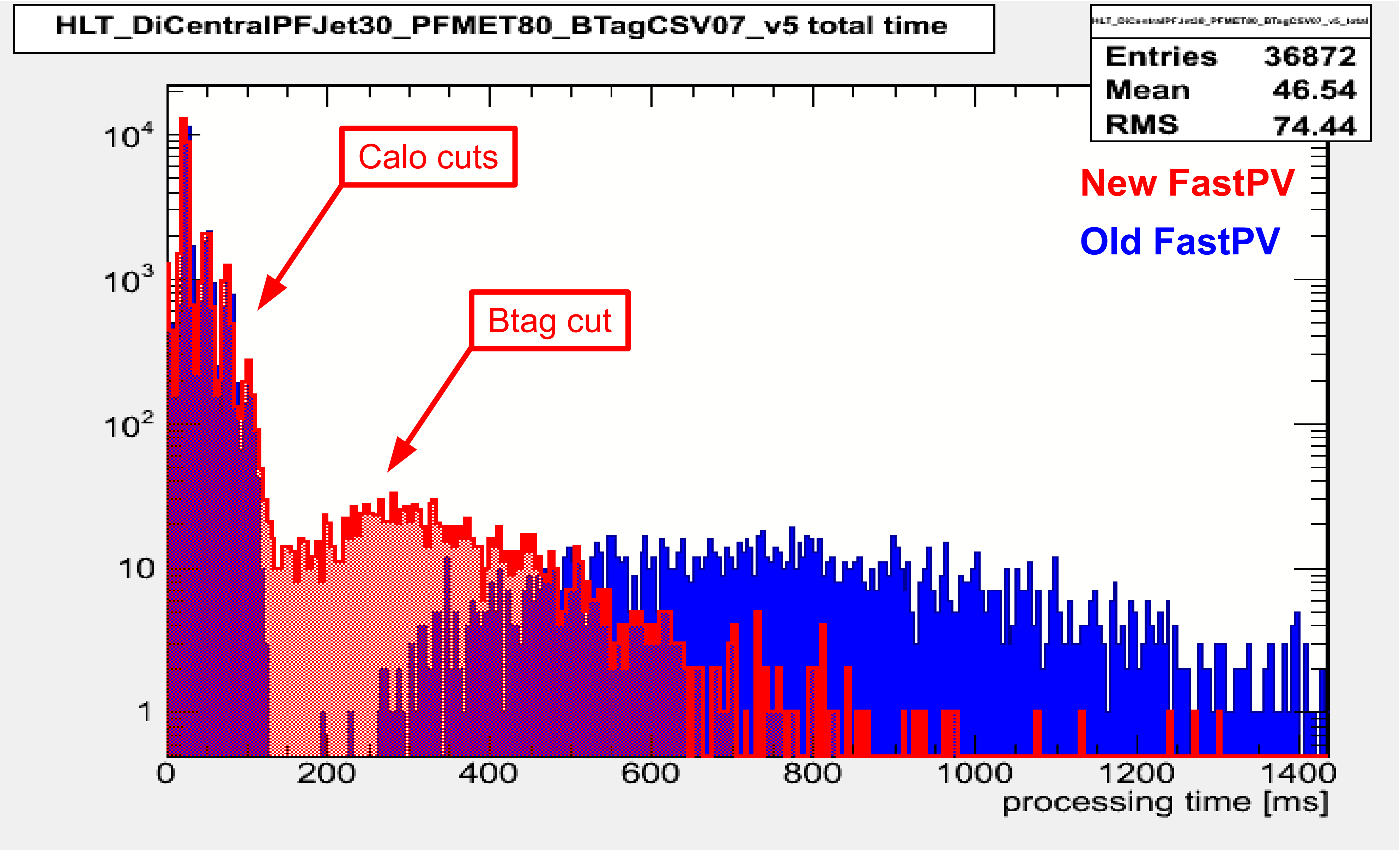


Backup

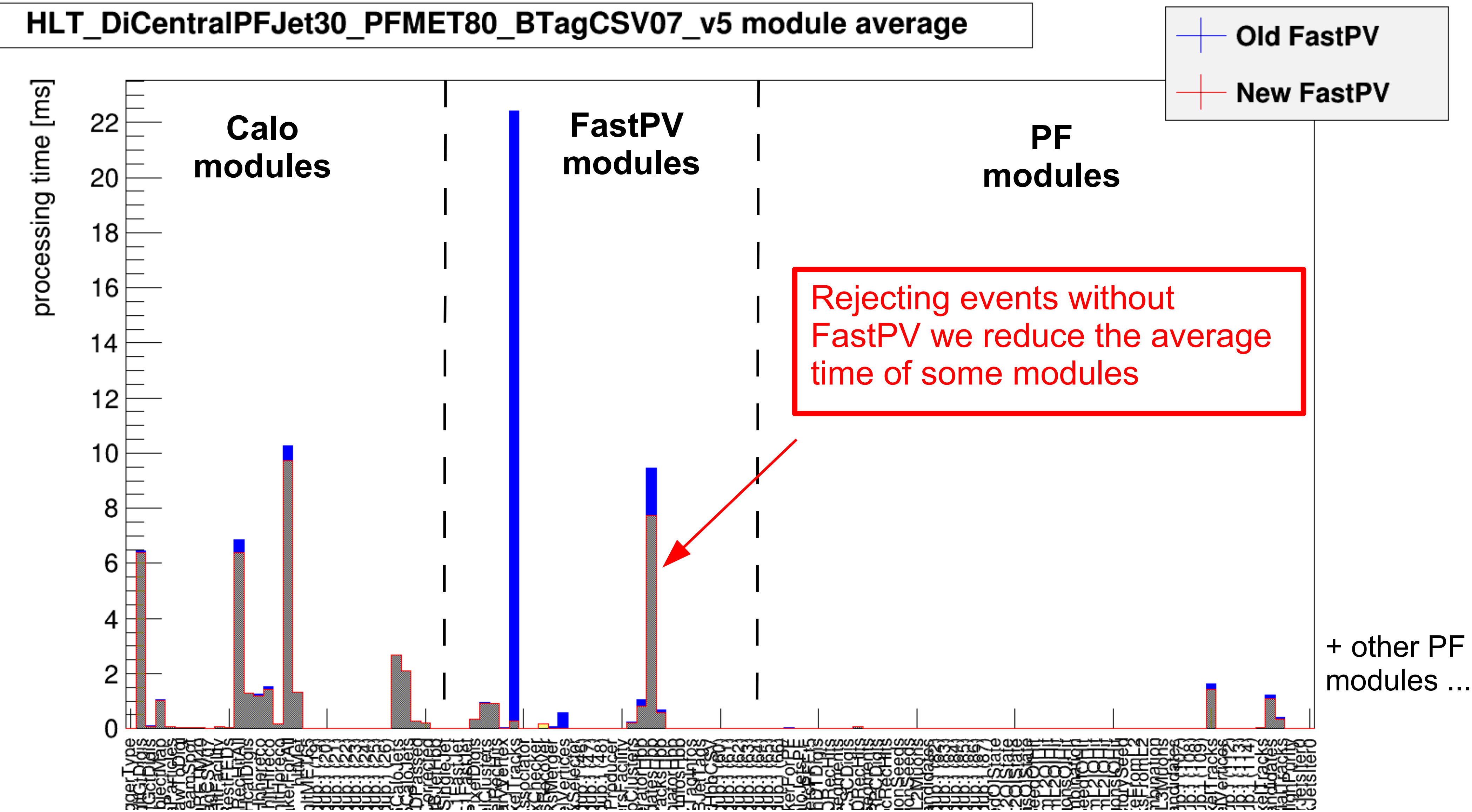
Vertex checker

- ✔ To recover the events with a wrong FastPV, we use the "vertex checker".
- ✔ For each jet, it check that at least 5% of the energy of the jet is seen in pixel tracks:
  - ✔ If a jet don't pass the checker we redo the regional tracking around the jet, but without the FastPV contraint.
- ✔ With the new algorithm we have more time so we had increased the threshold to pass to 10% of the jet energy.
  - ✔ So now we spend more time in the vertex checker module.

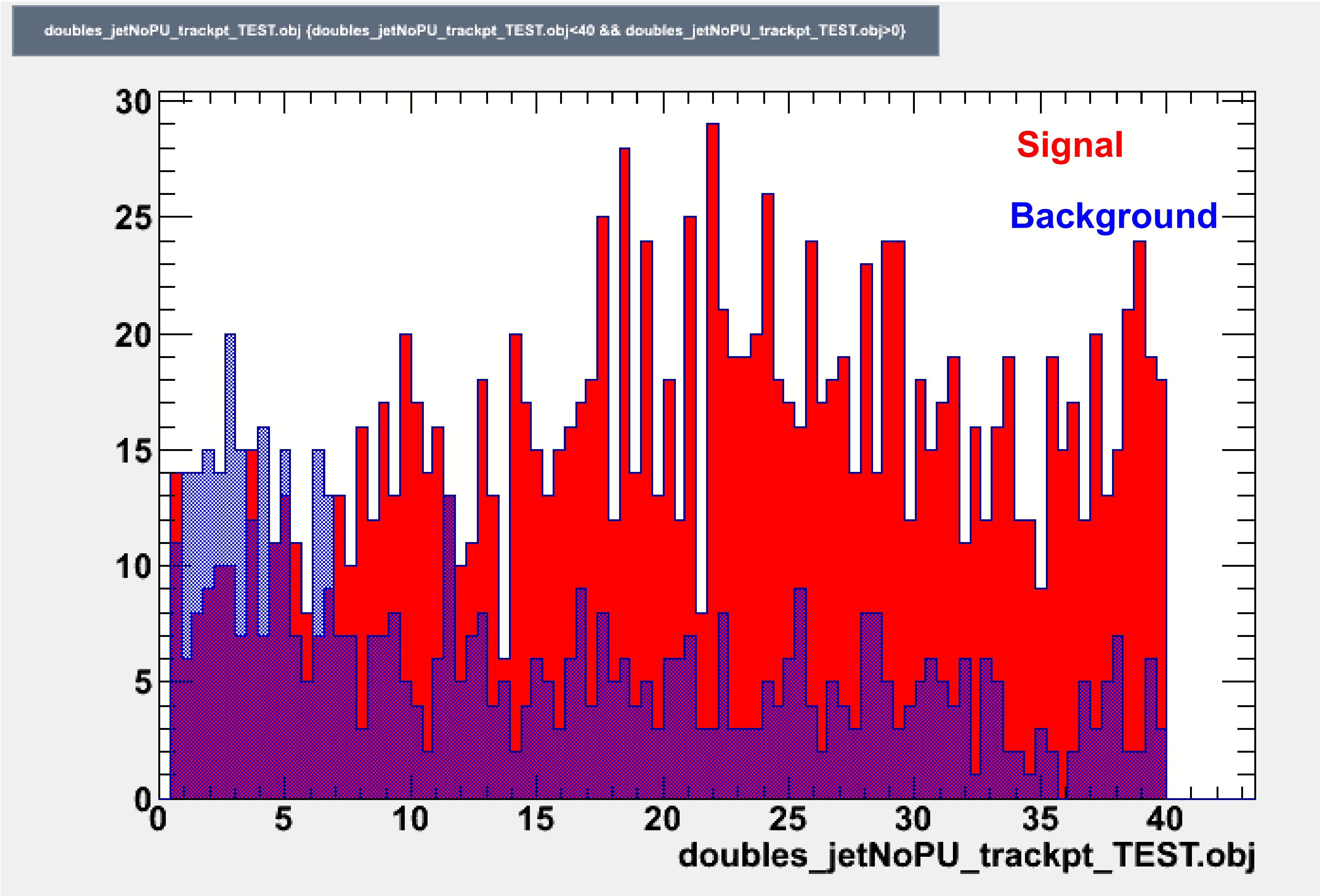
Total time (without PF)



Time module average

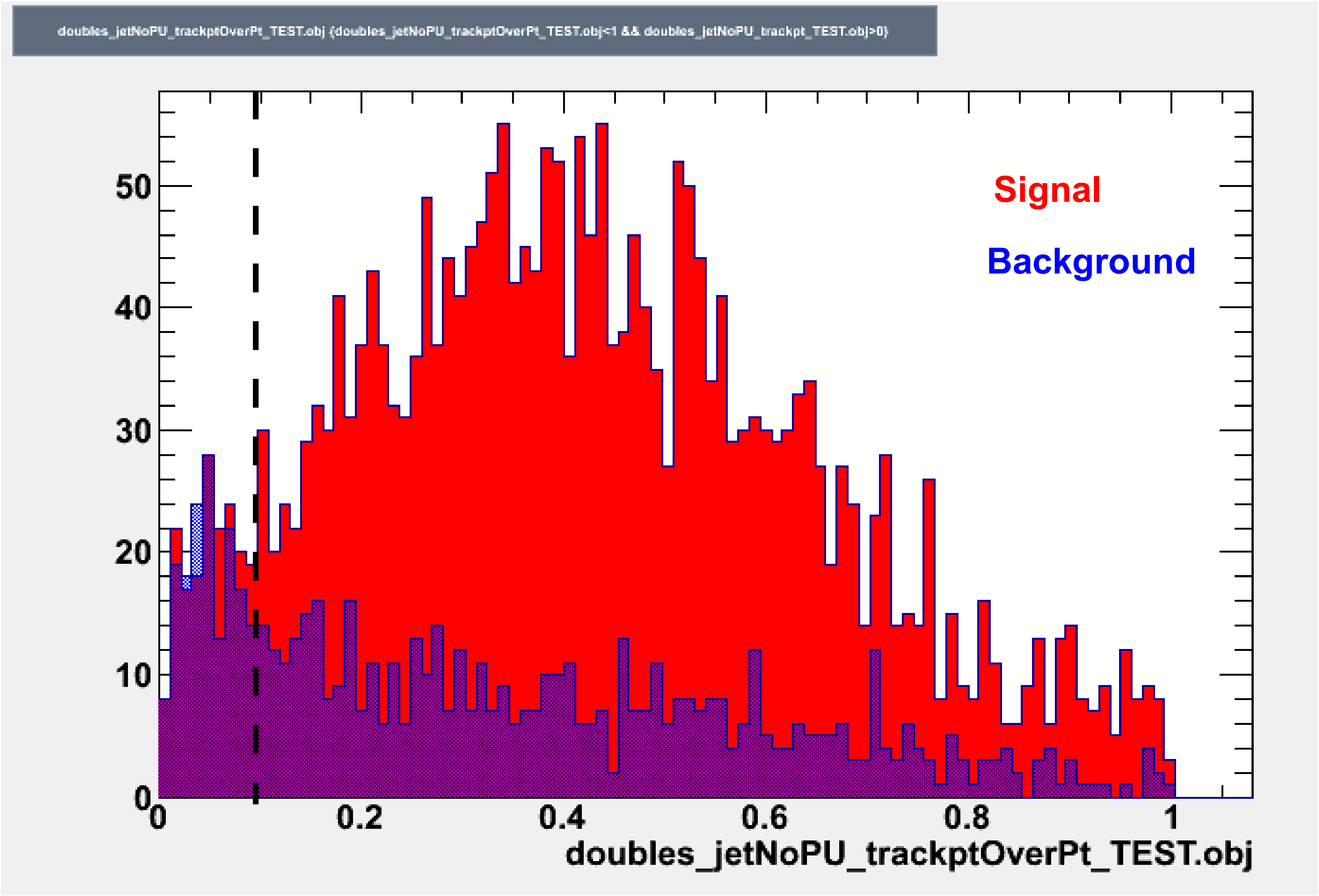


TrackPt



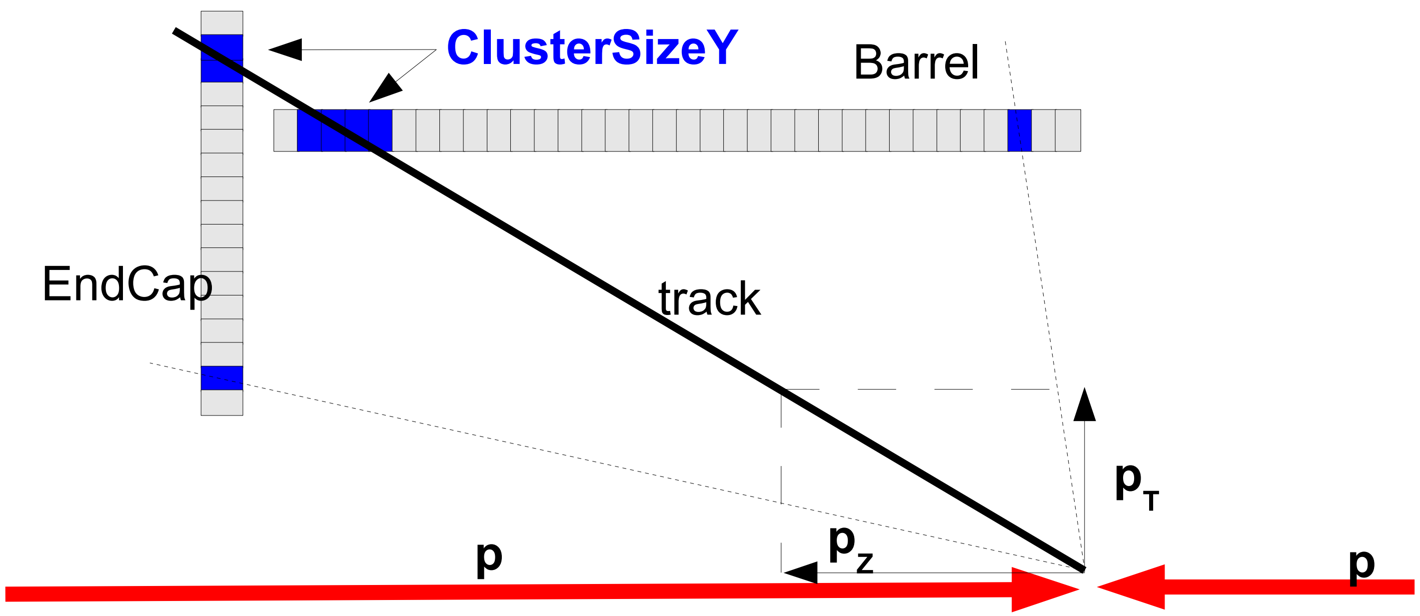
doubles_jetNoPU_trackpt_TEST.obj {doubles_jetNoPU_trackpt_TEST.obj<40 && doubles_jetNoPU_trackpt_TEST.obj>0}
Signal
Background
0
5
10
15
20
25
30
35
40
doubles_jetNoPU_trackpt_TEST.obj

TrackPt/JetPt

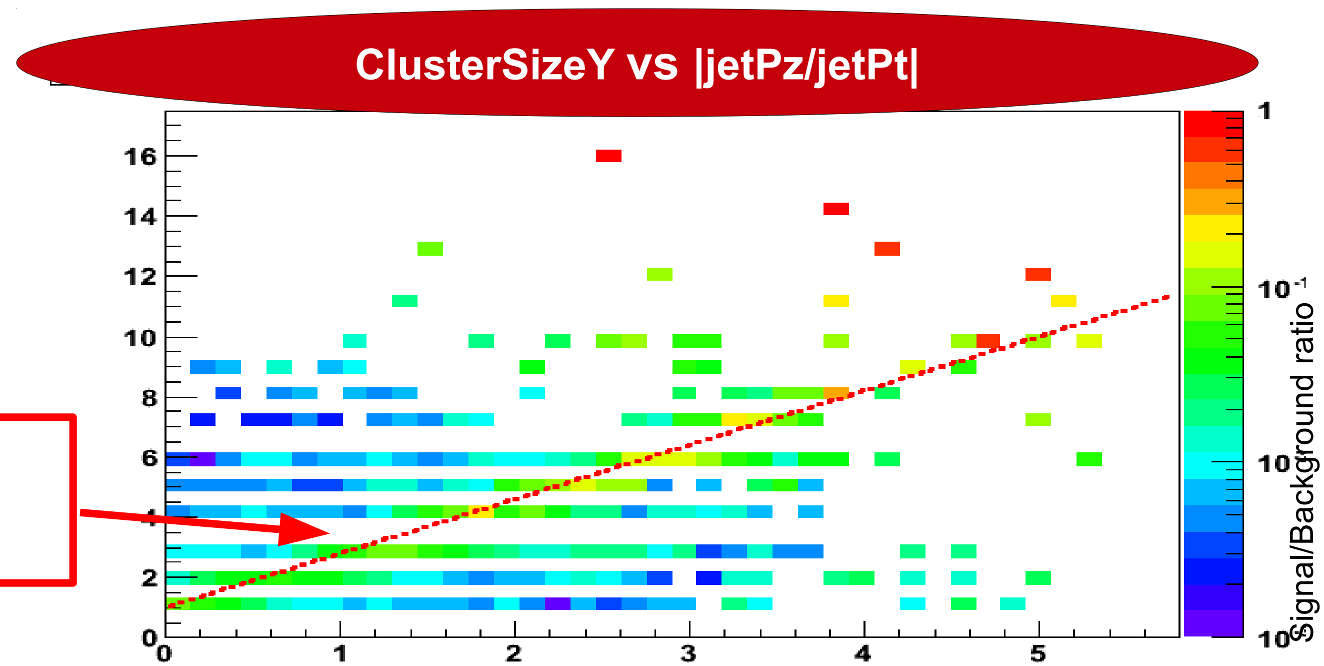


Clusters from EndCaps

- ✔ The projection of an EndCap cluster is similar.
- ✔ The main difference is the different expected clusterSizeY.
  - ✔ For Barrel we expect a clusterSizeY near to 1.8 * | jetPz / jetPt |.
  - ✔ For EndCap we expect a clusterSizeY near to 1.8 * **| jetPt / jetPz |.**



Cluster Size Y vs jet Eta



Better agreement using <u>1.8</u> instead of 2.0

PileUp distribution