PAT-driven Efficiency Measurements and Calculations

Jeffrey Berryhill

Kalanand Mishra

Fermilab

Amandeep Singh

Anil Pratap Singh

LPC & Panjab University

Overview

Since many Ewk and other electron analyses use PAT, it is imperative that we:

1.) provide a tool in the *TagAndProbe* machinery to derive efficiency tables for PAT electron reconstruction sequence.
2.) perform a comparison study of the PAT and Reco electron efficiencies.
3.) compare basic distributions of the PAT and Reco electrons in order to understand any observed difference in their efficiencies.

The PAT layer1 electron is just a wrapper around the reco::gsfElectron plus some cleaning ⇒ we expect their efficiencies to be equal or nearly equal bin-by-bin.

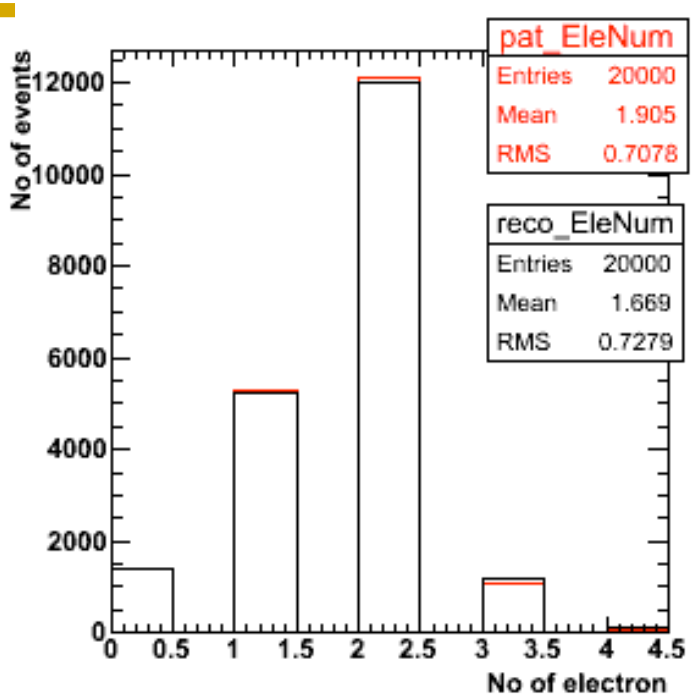
In the following slides, I will:

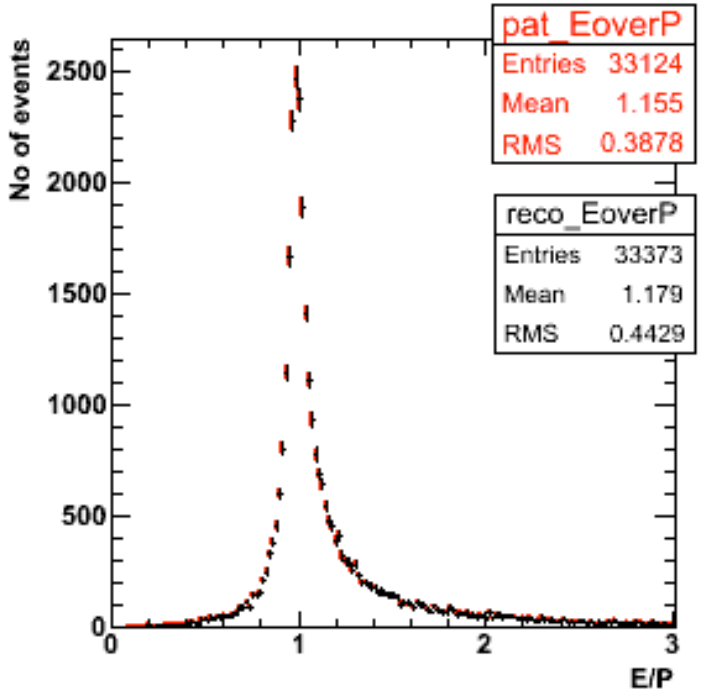
★ compare the basic reconstruction quantities for the PAT and Reco electrons
★ compare their efficiencies in Summer08 and in Relval_2_2_1 samples
★ give you some update on the TagAndProbe machinery in CMSSW_3_1_X.

Comparison of PAT & Reco electron quantities

Electron multiplicity and E/p

Summer08 sample, 20 k events

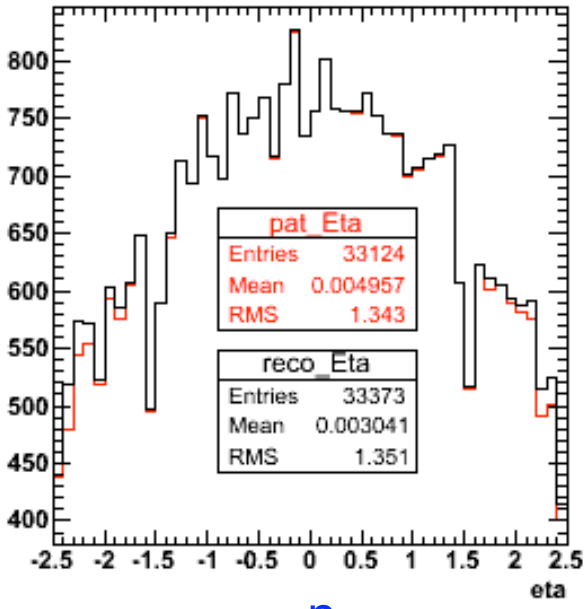




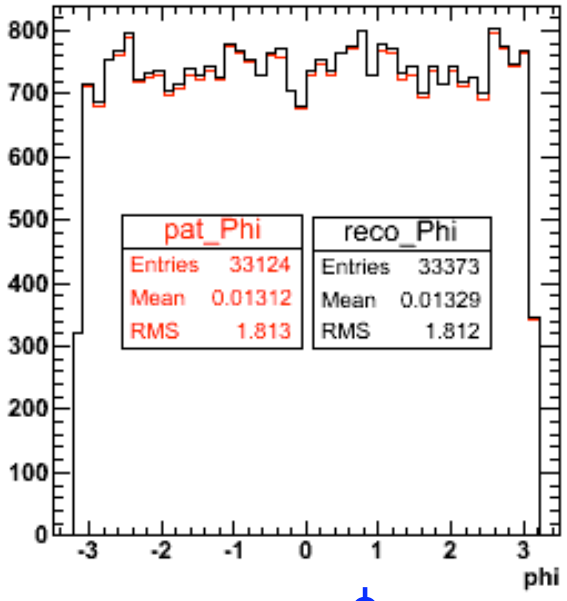
There are roughly 0.75% more reco::*gsfElectron* than PAT electron. This is because the *gsfElectron* collection sometimes contains duplicates (*e.g.*, when a super cluster is matched to more than one tracks). The PAT layer1 electrons are stored after removing duplicates. For the Reco electron sequence also we calculate efficiency after removing the duplicates.

η, ϕ, and p_T

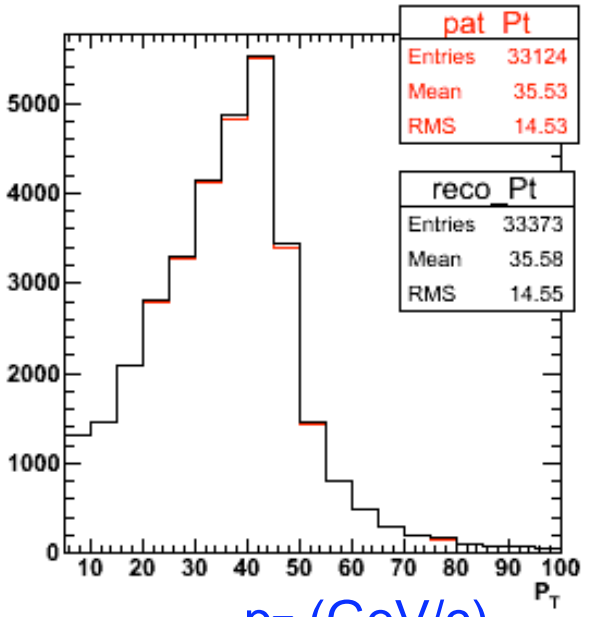
Summer08 sample, 20 k events



η



ϕ

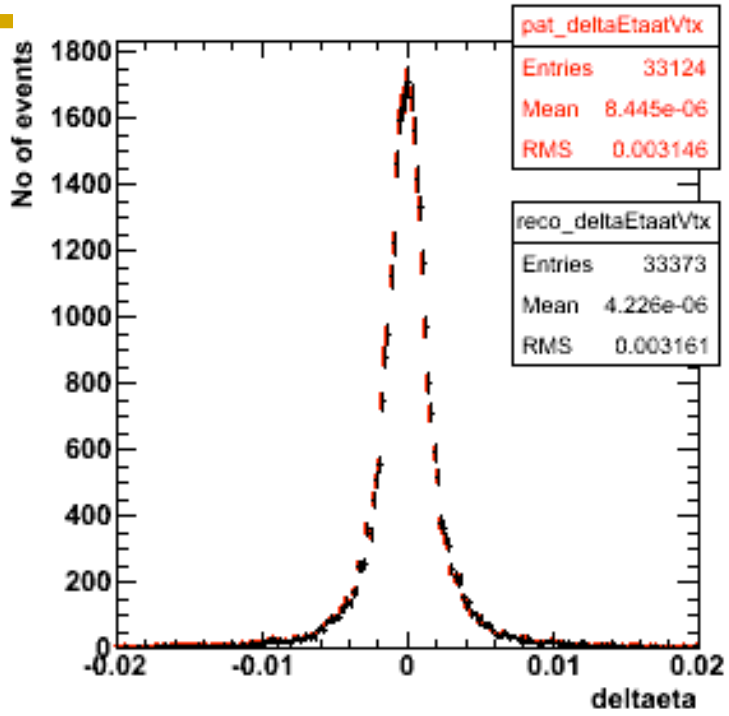


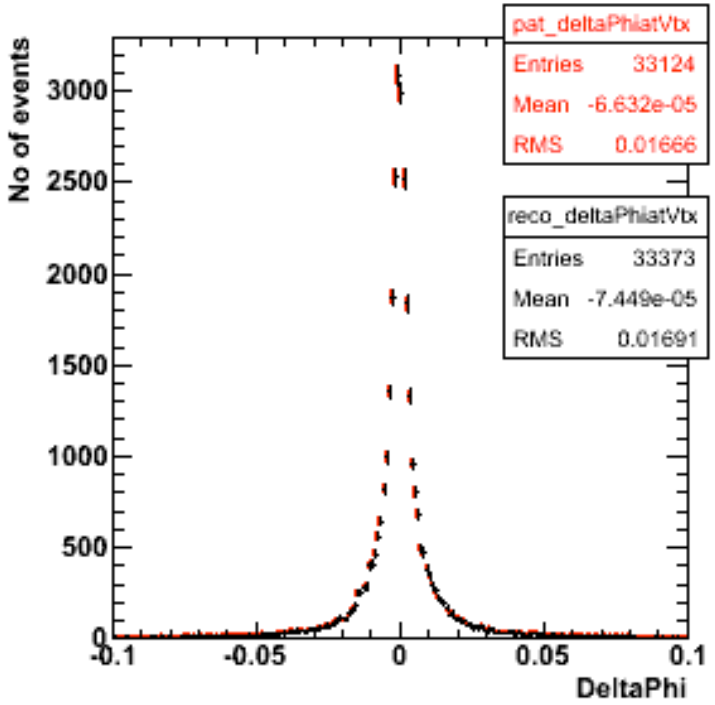
p_T (GeV/c)

Basic kinematic distributions are essentially the same for the Reco and PAT electrons. It also seems that most of the duplicates in the Reco collection are in the endcaps.

Δη and Δϕ

Summer08 sample, 20 k events





No difference in Δη or Δϕ between the Reco and PAT electrons.

Comparison of PAT vs Reco electron efficiencies in Summer08 (2_1_8) sample (scaled for 10 pb^{-1})

A quick recap of the tag & probe selection

Tag Selection

- $|\eta| < 1.4442$ || $1.56 < |\eta| < 2.5$
- $p_T > 20$ GeV/c.
- Track isolation:
  - $\Sigma\ p_t^{tracks} / p_t^{el} < 0.2$
- Electron ID: Robust Loose
- Trigger: HLT_LooseIsoEle15_LW_L1R

Probe Selection

- $|\eta| < 1.4442$ || $1.56 < |\eta| < 2.5$
- $p_T > 20$ GeV/c.
- Fit the tag-probe invariant mass to get the number of signal events.

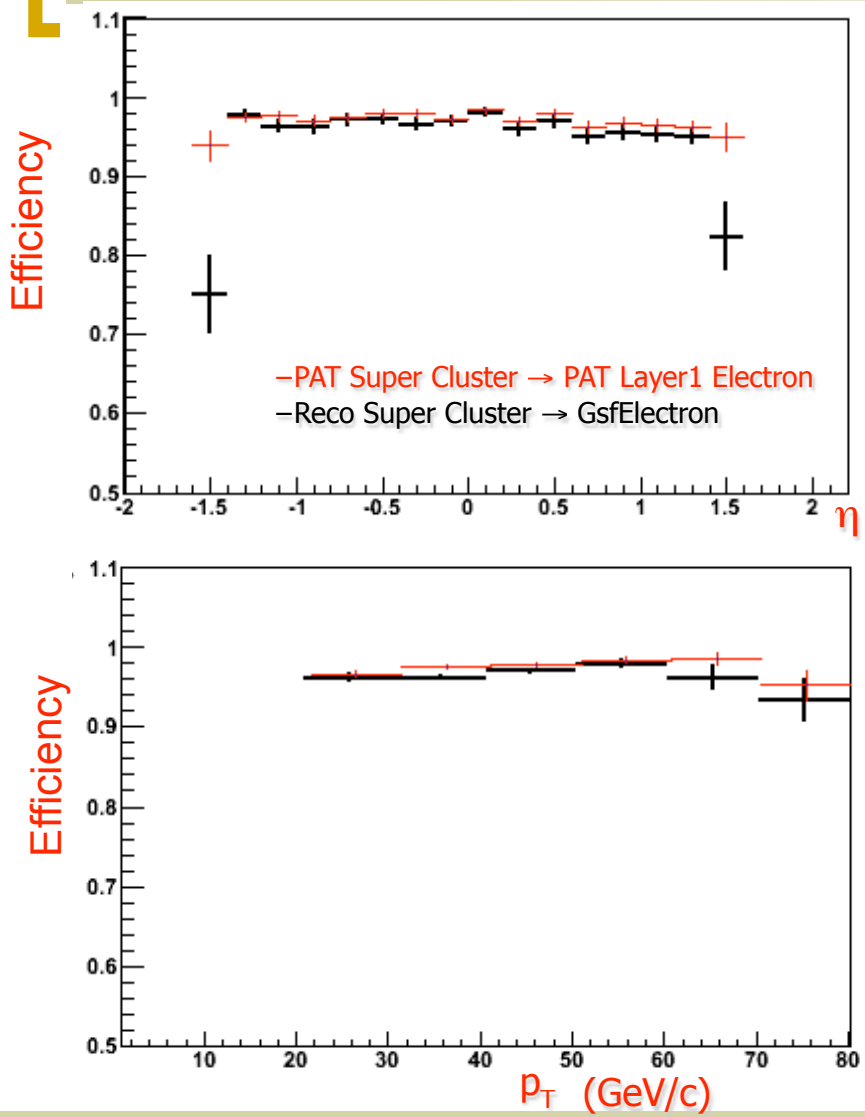
Obtain factorized efficiencies for passing probes:

SuperCluster → GsfElectron → Isolation → ID → HLT

➡ PAT electron: *PatLayer1Electrons*
➡ Reco electron: *gsfElectrons*

The *PatLayer1Electron* is just a wrapper around the *Reco::gsfElectron* along with the removal of duplicates. Therefore, we expect the PAT and Reco electron efficiencies to be almost identical.

Efficiency in Summer08: super cluster → electron

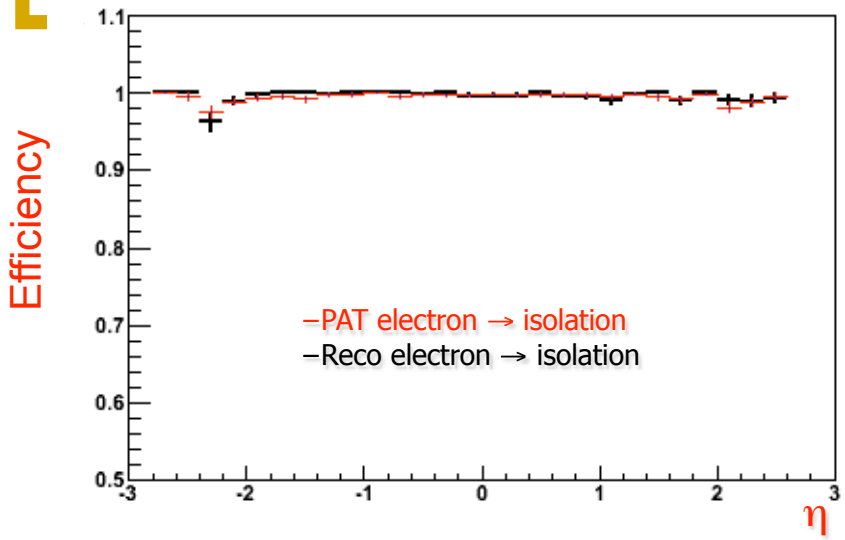


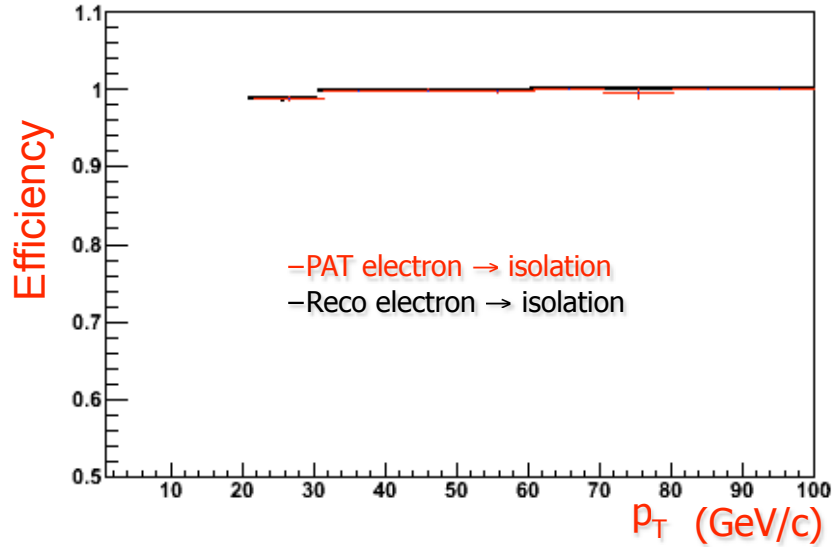
★PAT and Reco efficiencies look similar to each other.

★We will look into possible cause of small differences in a later slide.

★At the moment, can reconstruct PAT super cluster only in the barrel. Working on including the endcap super cluster also.

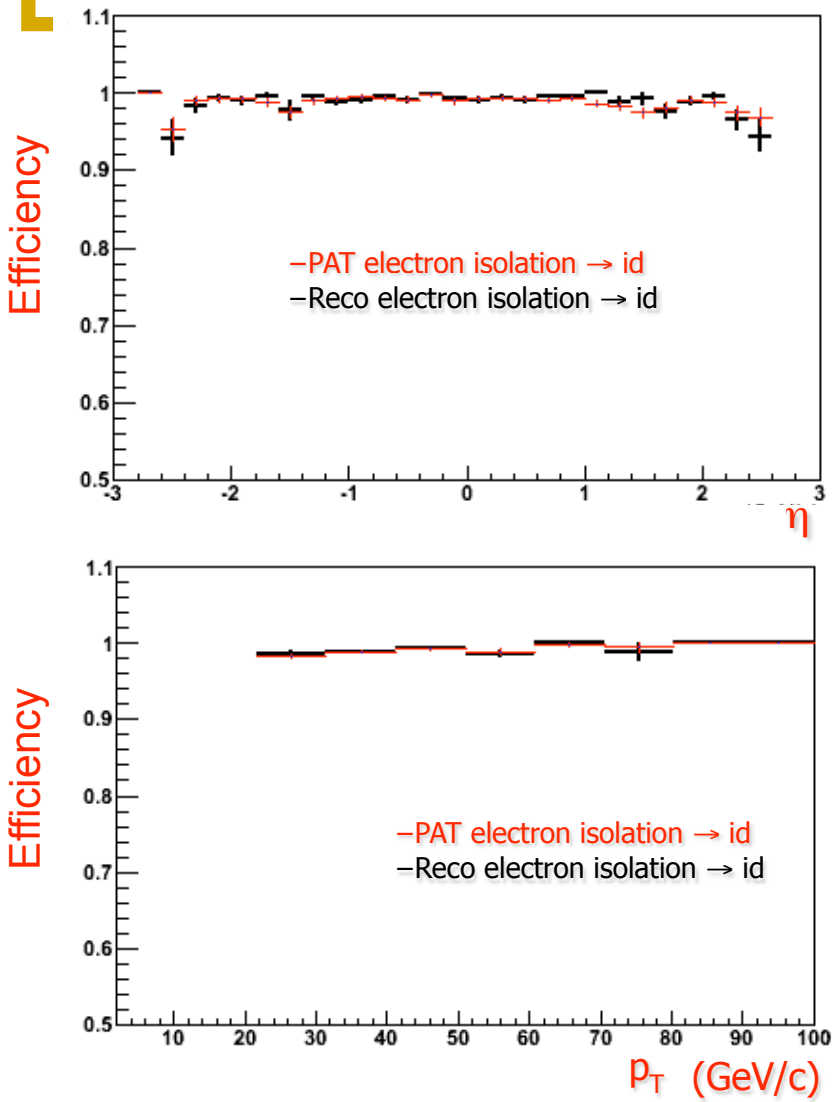
Efficiency in Summer08: electron → isolation





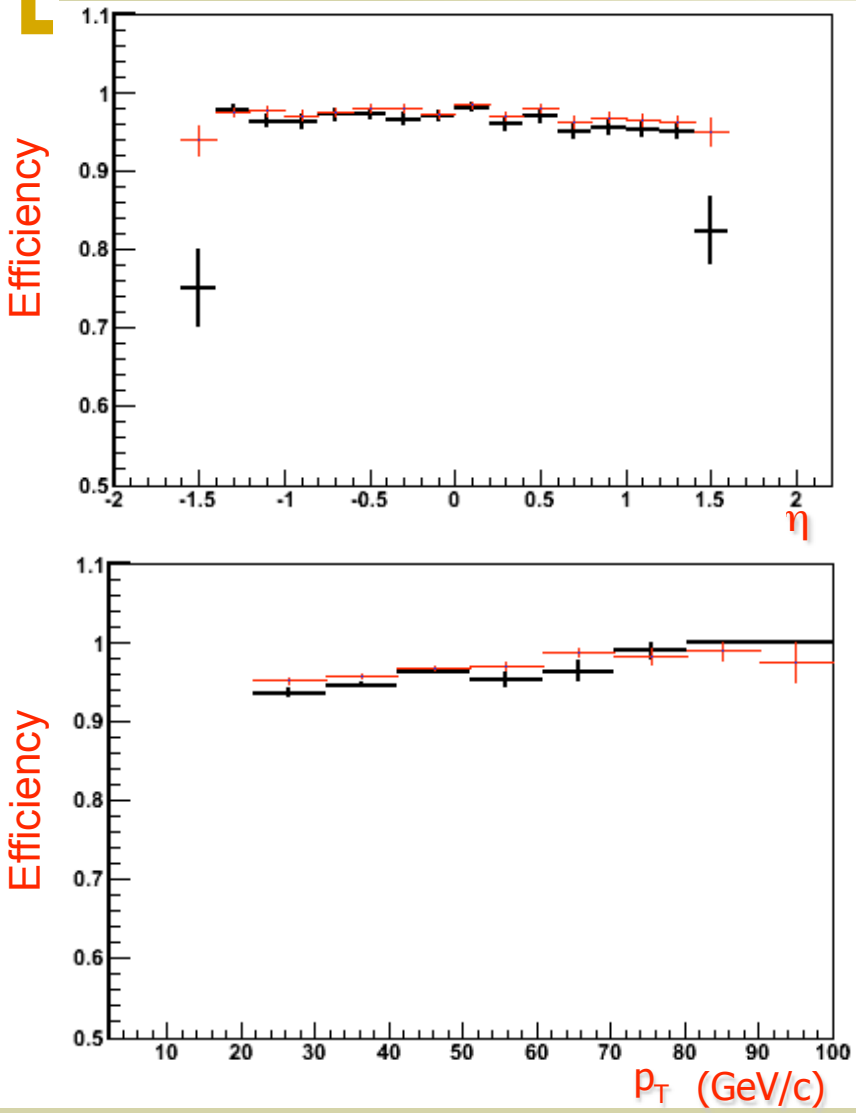
★PAT and Reco efficiencies look very similar to each other.

Efficiency in Summer08: isolation → loose id



★PAT and Reco efficiencies look very similar to each other.

Efficiency in summer08: loose id → HLT



★PAT electron efficiency is slightly higher than the Reco electron efficiency – more so in the endcaps and at low p_T.

★Reconstruction and trigger-matching steps are done exactly the same way for both (i.e., the same trigger path, filter module, and matching criteria used for both).

★The only difference is in the removal of duplicates (see next slide).

Electron duplicate removal

- In case of Reco, “duplicate removal” done by checking the match for *super cluster*. In PAT, a check is done for matching *super cluster* or matching *gsfTrack*.
- In case of Reco, “duplicate removal” is done after acceptance cut. In PAT it is before.
- In both cases, only the best duplicate is kept (the one which has E/p closer to one).

Reco electron

```
bool duplicate = false;
reco::GsfElectron* BestDuplicate = elec1;
 if( elec1->superCluster() == elec2->superCluster()) {
       duplicate = true;
       if( fabs(BestDuplicate->eSuperClusterOverP()-1.0) >=  fabs(elec2->eSuperClusterOverP()-1.0) )
          BestDuplicate = elec2;
 }
```

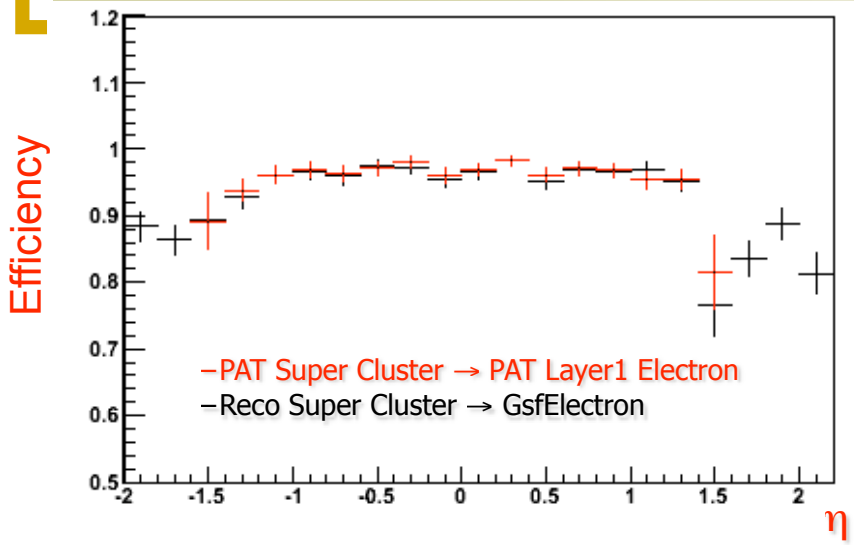
PAT electron

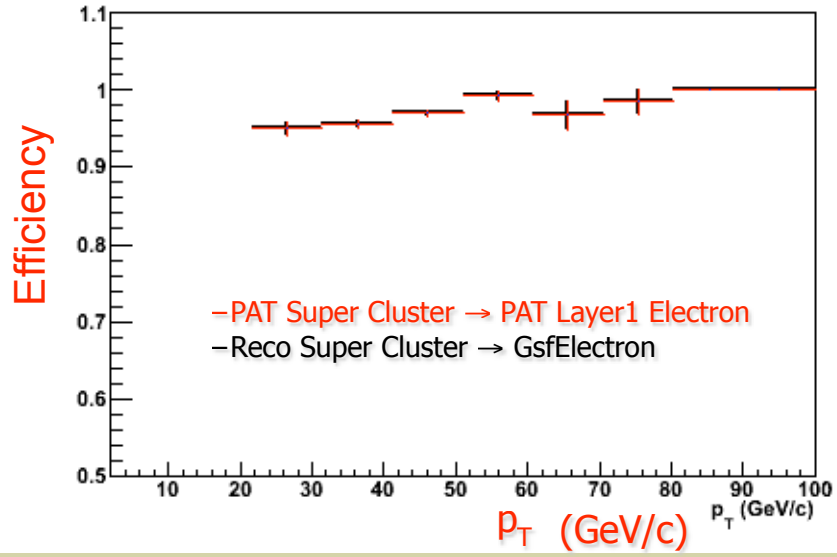
```
void PATElectronCleaner::removeDuplicates() {
class DuplicatedElectronRemover {
        public:
           struct SameSuperclusterOrTrack {
              template<typename T1, typename T2>
              bool operator()(const T1 &t1, const T2 &t2) const {
                 return ((t1.superCluster() == t2.superCluster()) ||   (t1.gsfTrack() == t2.gsfTrack()));
              }
           }; // struct
}                                  // best candidate selection on the basis of  | E/p − 1 |, same as above.
```

Not sure if these differences explain the observed small difference in efficiency.

Comparison of PAT vs Reco electron efficiencies in RelVal 2_2_1 sample

Efficiency in RelVal 2_2_1: super cluster → electron

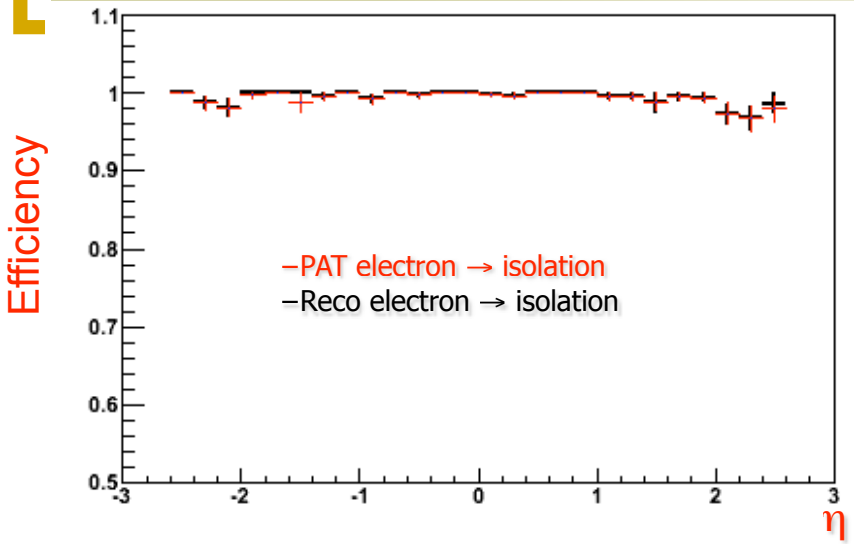


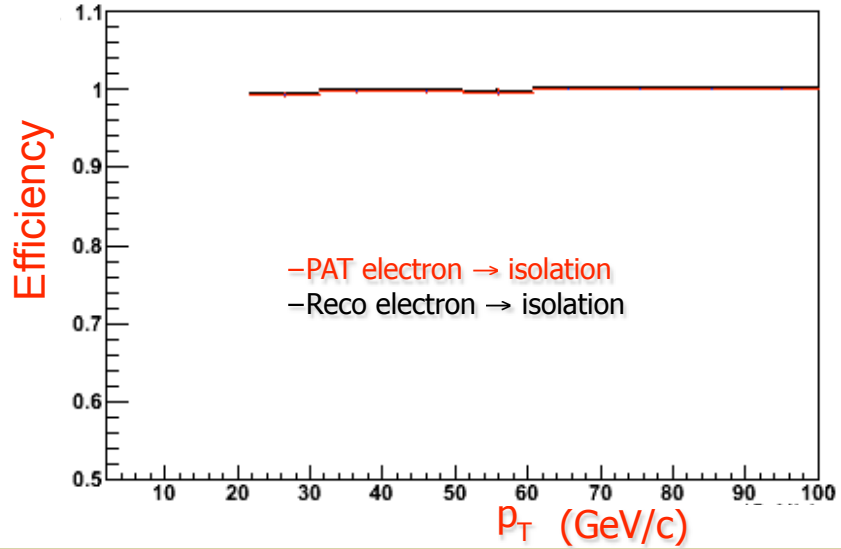


★Again, the PAT and Reco efficiencies look similar to each other.

★PAT super cluster only in the barrel for now.

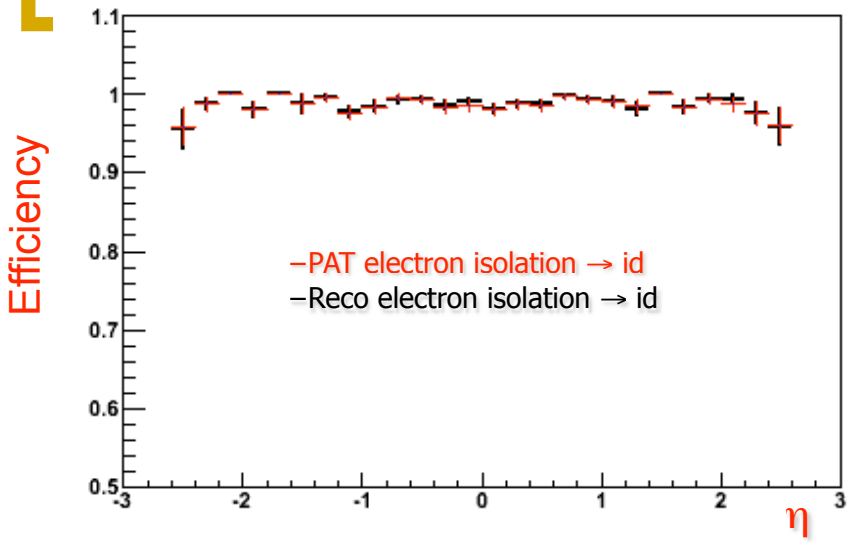
Efficiency in RelVal 2_2_1: electron → isolation

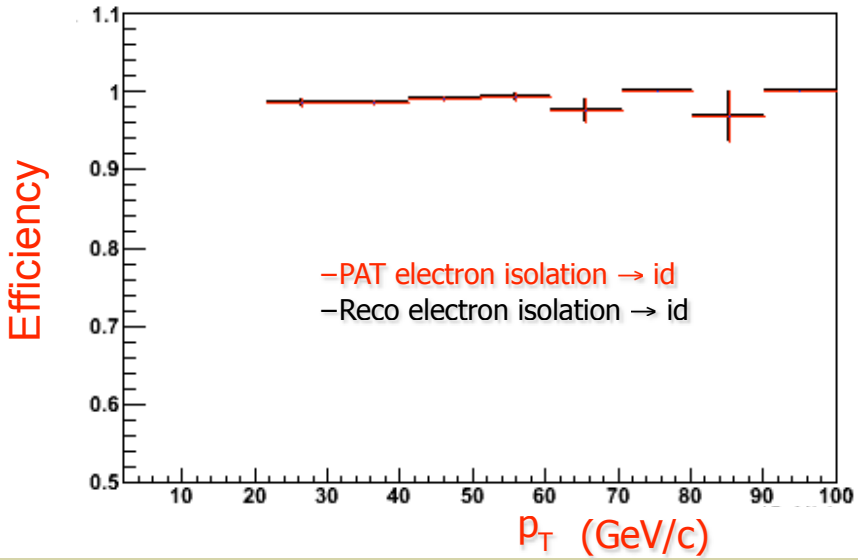




★PAT and Reco efficiencies look very similar to each other.

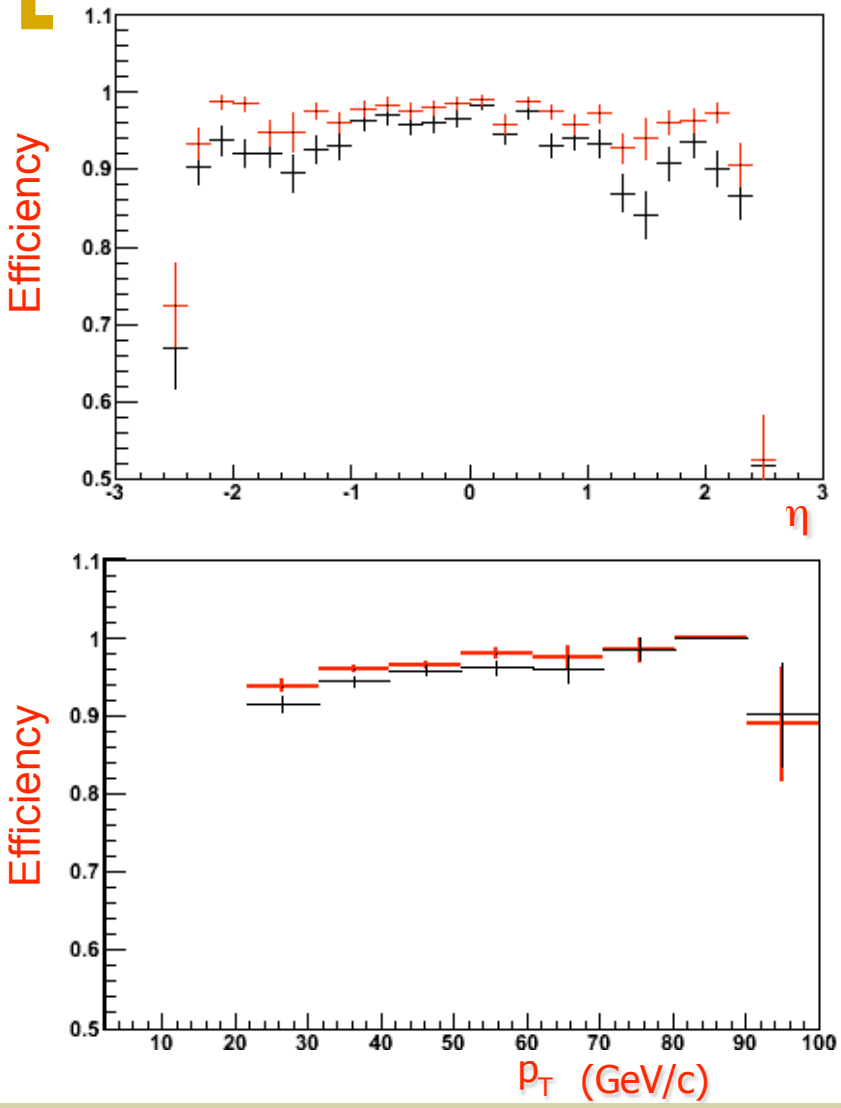
Efficiency in RelVal 2_2_1: isolation → loose id





★PAT and Reco efficiencies look very similar to each other.

Efficiency in RelVal 2_2_1: loose id → HLT



★PAT electron efficiency is slightly higher than the Reco electron efficiency – just like in case of Summer08 sample.

Status of *TagAndProbe* in CMSSW 3_1_X

TagAndProbe will come officially shipped with 3_1_X

- From CMSSW_3_1_X onwards, the *TagAndProbe* will come shipped with official release of the CMSSW by default.
- Then there will be no need to check out the package separately (except for bug-fixes and feature updates).
- Though the t & p machinery doesn't have dependence on data/MC production sequence and only a weak dependence on the *DataFormat*, the package will be useful as an analysis and validation tool inside the CMSSW / FWLite framework.
- Several functionalities in the package are not dependent on CMSSW and can be used standalone in a bare ROOT/RooFit environment. We are working to decouple these two types of functionalities.
- The current version of the package compatible with 3_X pre-releases is:

  PhysicsTools/TagAndProbe V01-06-02

- At the moment, it needs to be checked out from CVS and can be compiled in the usual way.

New developments in TagAndProbe for 3_1_X

- A machinery in place to store the factorized electron reconstruction efficiencies (and uncertainties therein) and fake rates as text tables for both RECO and PAT objects.
- The efficiencies can then be read back at analysis level.This machinery will streamline the study of systematic uncertainties in physics analysis coming from electron reconstruction.
- An improved error calculation method (using *TGraphErrors*) implemented for efficiency obtained with sideband subtraction method.
- Work underway to decouple the main fitting and calculation engine from the edm framework ⇒ this will facilitate integration with FWLite and higher-level analysis software.
- This decoupling will also enable users to plugin the *TagAndProbe* machinery with their own event reconstruction code.

Status & Summary

- ✓Performed a detailed comparison of efficiencies of PAT & Reco electron sequences using the TagAndProbe machinery.
- ✓The tag-and-probe software has been upgraded to calculate efficiencies for PAT electron reconstruction sequence. Now one can calculate PAT efficiency just like Reco electron efficiency.
- ✓TagAndProbe package will ship with CMSSW_3_1_X onwards.
- ✓Work in progress to
  - ★decouple main t & p analysis guts from edm dependency
  - ★integration with FWLite
  - ★support for customized user-defined electron selection
  - ★do a better documentation.