TRUTH ENERGY CONSERVATION

A Note on How to Relate MCTruth, MCNeutrino, MCParticle, MCShower, MCTrack Energies in LArSoft

Kaleko 042815

INPUT DATA

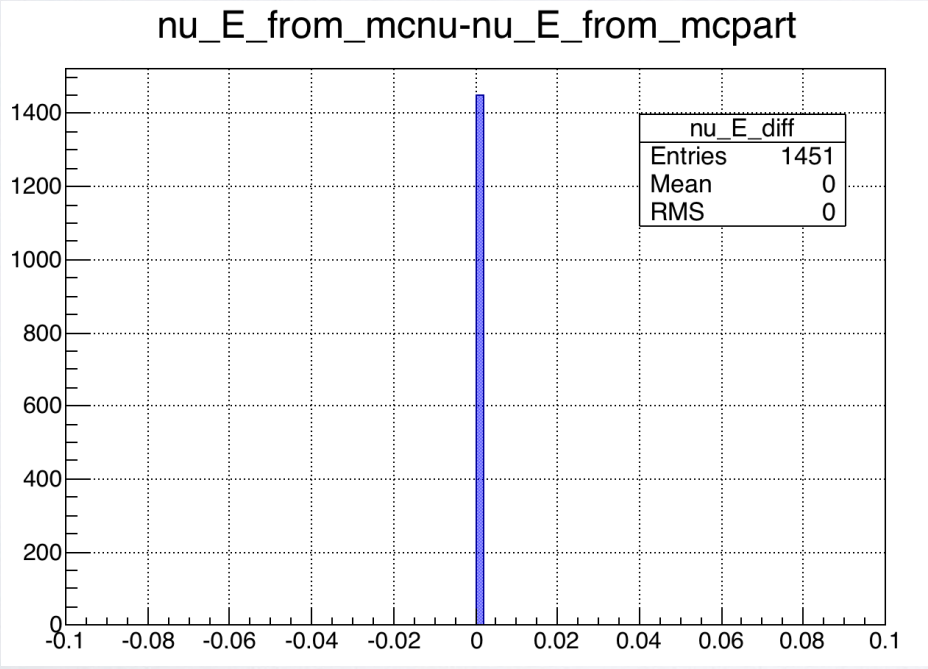
- Running on MCC5 bnb_nue files.
- Using CC1E filter (**only events that are charged current, 1 electron in the final state**)
    - This filter isn't actually necessary for this study, but these are events that I personally am interested in.
- MCTruth, MCParticle, MCNeutrino information all comes directly from GENIE.
- MCShower and MCTrack come from energy depositions in GEANT4.

MCNU VS. MCPART

- Neutrino energy from MCNu matches exactly
  Neutrino energy from MCPart

```
nu_E_from_mcnu = ev_tru->at(0).GetNeutrino().Nu().Trajectory().at(0).E()
```

```
for (auto const mcp : ev_tru->at(0).GetParticles())
  if (abs(mcp.PdgCode()) == 12)
    nu_E_from_mcpart = 1000.*mcp.Trajectory().at(0).E();
```

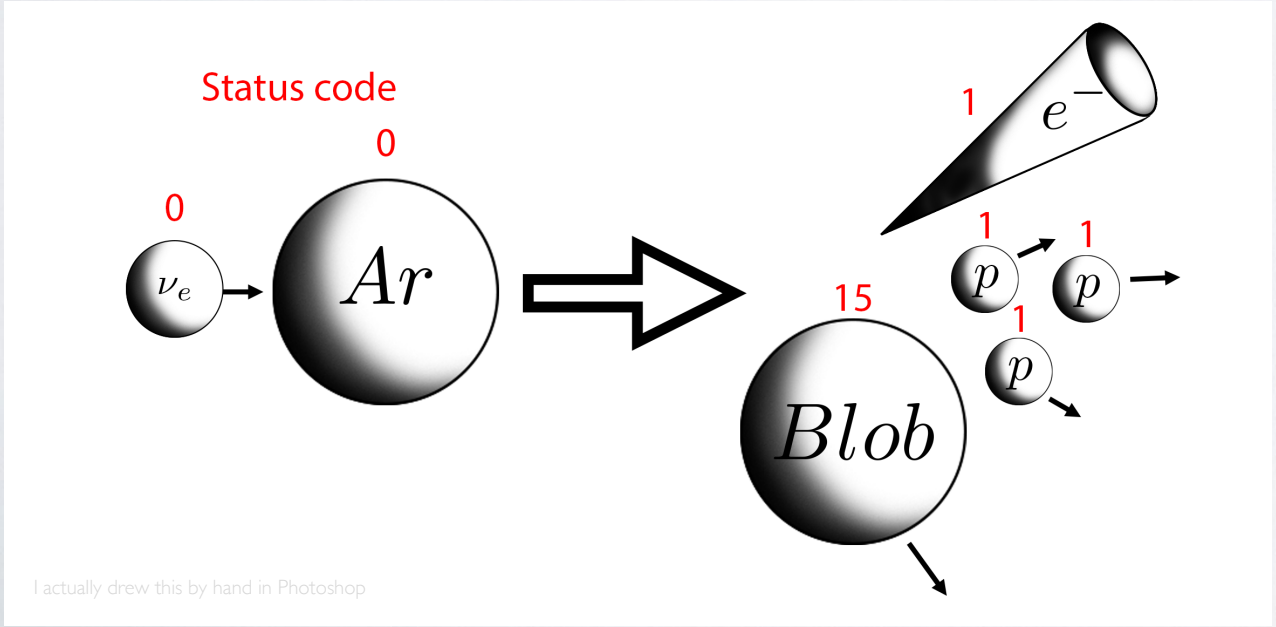


Note: the vertex from each matches exactly as well.

Note: Code snippets shown in these slides are actually written in LArLite, but the data products in LArSoft are identical.

MCPARTICLE "STATUS CODE"

```
Status code definitions:
-1  => Undefined
0   => Initial State (generator level)
1   => Stable Final State (generator level, particles to be
                           tracked by detector-level MC)
2   => Intermediate State
3   => Decayed State
10  => Correlated Nucleon
11  => Nucleon Target
12  => DIS Prefragmented Hadronic State
13  => Predecay Resonant State
14  => Hadron in the Nucleus (marked for hadron transport modules to act on)
15  => Final State Nuclear Remnant
       (low energy nuclear fragments entering the record
        collectively as a 'hadronic blob' pseudo-particle)
16  => Nucleon Cluster Target
```



I actually drew this by hand in Photoshop

Note: If you print out all of the MCParticles, you will find some with status codes other than 0, 1, and 15. Ignore these.

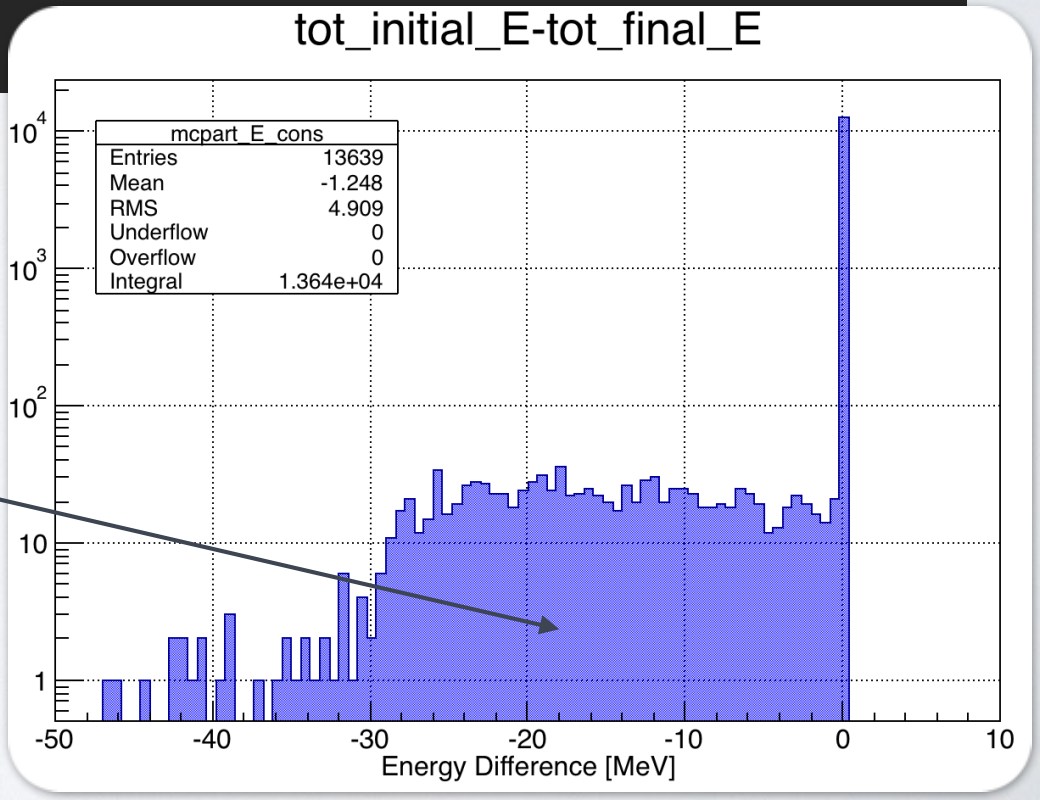
Note: Never deal with Kinetic energies, always total energies.

MCPARTICLE ENERGY CONSERVATION

```
double NuEnergyReco::ComputeTotalInitialE(const std::vector<larlite::mcpart> & myparts){

  double tmp_E = 0.;
  for (auto const mcp : myparts){
    if (mcp.StatusCode() == 0)
      tmp_E += 1000.*mcp.Trajectory().at(0).E();
  }

  return tmp_E;
}

double NuEnergyReco::ComputeTotalFinalE(const std::vector<larlite::mcpart> & myparts){

  double tmp_E = 0.;
  for (auto const mcp : myparts){
    if (mcp.StatusCode() == 1 || mcp.StatusCode() == 15)
      tmp_E += 1000.*mcp.Trajectory().at(0).E();
  }
  return tmp_E;
}
```

Note, the "1000.*" is just to put energies in MEV everywhere (personal preference)

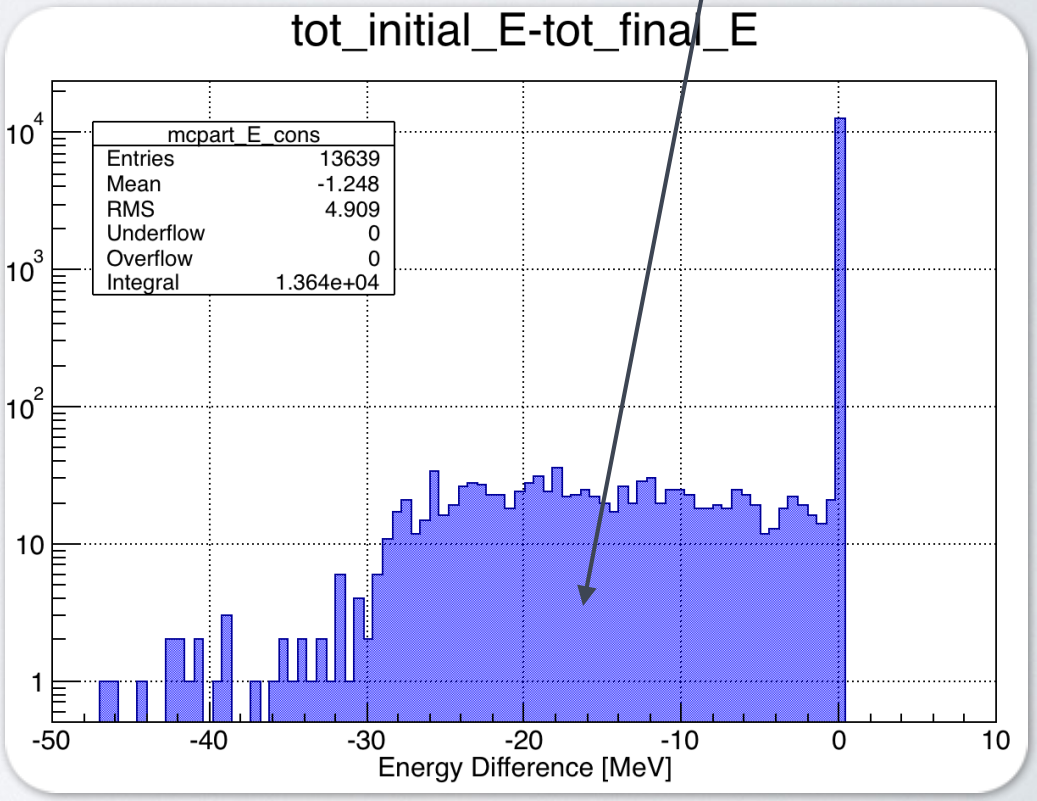
This extra tens of MeV energy is not yet fully understood



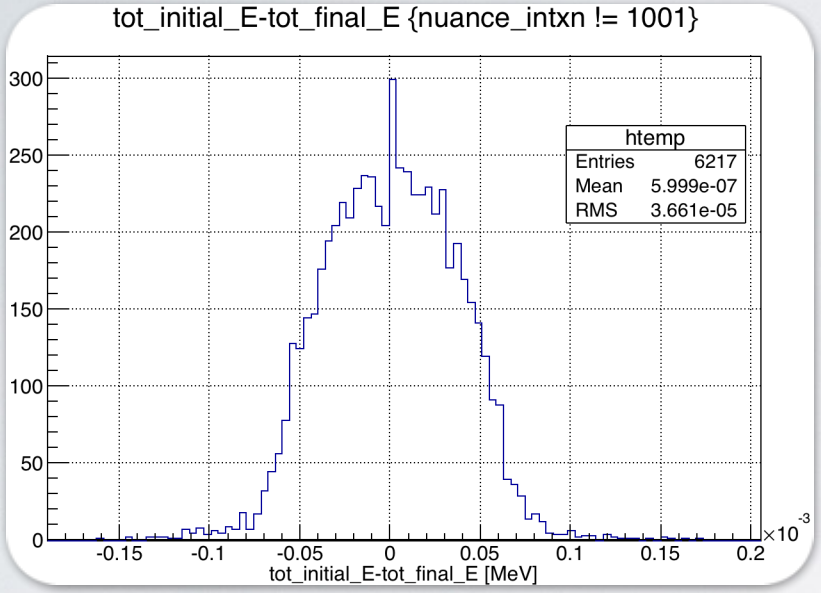
ASIDE: GENIE QUESTION

- What is this extra tens of MeV?
- Interesting tidbit: **if I require that the NUANCE interaction channel is not 1001 (so, non CCQE events), the above plot only has the peak at zero (the grass disappears)**
  - Non CCQE has 6217 entries, all at 0 +/- a few eV.
  - CCQE has 7422 entries, 1082 (14.5%) are in the grass.
- I am already taking into account the 29.5 MeV binding energy that pops out sometimes with PDG matching "genie::kPdgBindino"
  - Including when this pops out more than once in an event.

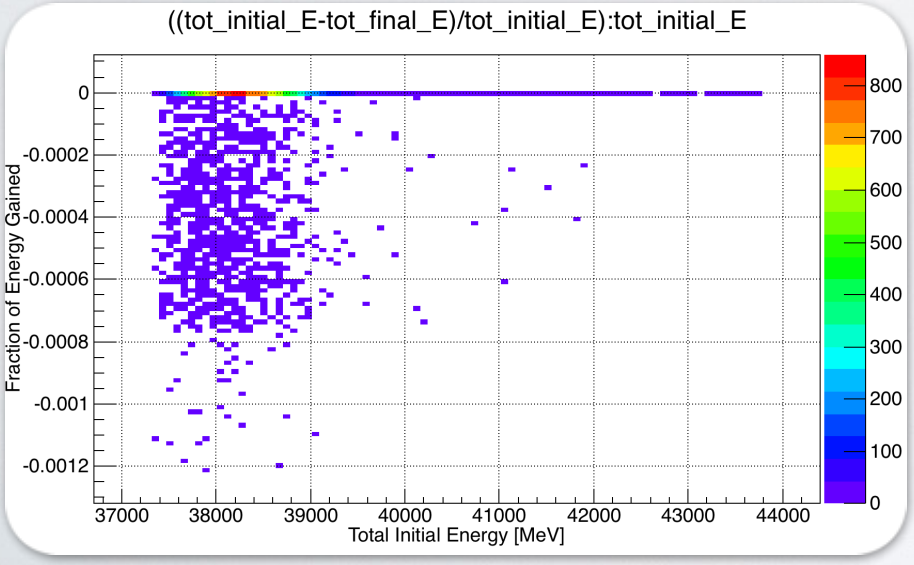
What is this extra tens of MeV energy?



ASIDE: GENIE QUESTION



Non-CCQE energy difference. Note the x-axis scale: this peak has RMS of 36 eV.



Correlation between fraction of energy difference and total energy.

GENIE QUESTION ANSWER

This is from e-mail correspondence with Steve Dytman (thanks Steve!)
Long story short: don't fret over these ~10s of MeV.

GENIE cares a lot about cons of energy and momentum, but it's not simple.
A MIT student found a disagreement and it was discussed in Cross Section group.
Flavio did a study which I append that says there is 'no' problem.

However, I acknowledge that there is difficulty in computing cons of energy for GENIE events. The binding energy is taken out in a way that creates problems. It is also taken out with a high energy approximation that won't be valid for lower energies of Microboone.

One goal is make the BE correction in GENIE more straightforward, only (my) time is needed.
A better goal is to have the spectral function as a standard model, that is happening.
v2.9.0 has an alternate QE model (effective spectral function of Bodek, Christy, and Coopersmith) which does BE in a different way. Benhar spectral function model will be in next release and that should do it right.

MCSHOWER + MCTRACK ENERGY CONSERVATION

```
for (auto const& mcshower : *ev_mcs){
  //Avoid double counting
  if(mcshower.AncestorTrackID() != mcshower.TrackID()) continue;
  //Not using detprofile here. This energy includes mass.
  tot_mcs_E += mcshower.Start().E();
}

for (auto const& mctrack : *ev_mct){
  //Avoid double counting
  if(mctrack.AncestorTrackID() != mctrack.TrackID()) continue;
  //This energy includes mass. Not counting only deposited energy.
  tot_mct_E += mctrack.Start().Momentum().E();
}
```

Total MCShower Energy
(Note: not using the usual "DetProfile" for deposited energy)
(Note: asking the shower is its own ancestor is to avoid double-counting)

Total MCTrack Energy
(Note: not using the usual deposited energy)
(Note: asking the track is its own ancestor is to avoid double-counting)

```
auto mcpart_list = ev_tru->at(0).GetParticles();
for (auto const mcp : mcpart_list){

  if (mcp.StatusCode() == 1)
    //These are all the PDGs that are made into mcshowers or mctracks (by definition)
    //pi0 removed because its two gammas (22) are what matter. these are removed
    if( mcp.PdgCode() != 22 &&
        mcp.PdgCode() != 11 &&
        mcp.PdgCode() != -11 &&
        mcp.PdgCode() != 211 &&
        mcp.PdgCode() != -211 &&
        //mcp.PdgCode() != 111 &&
        mcp.PdgCode() != 13 &&
        mcp.PdgCode() != -13 &&
        mcp.PdgCode() != 2212 &&
        mcp.PdgCode() != 321 &&
        mcp.PdgCode() != -321  )
      non_mcs_mct_E += 1000*mcp.Trajectory().at(0).E();

  if (mcp.StatusCode() == 15)
    remnant_E = 1000*(mcp.Trajectory().at(0).E());
}
```

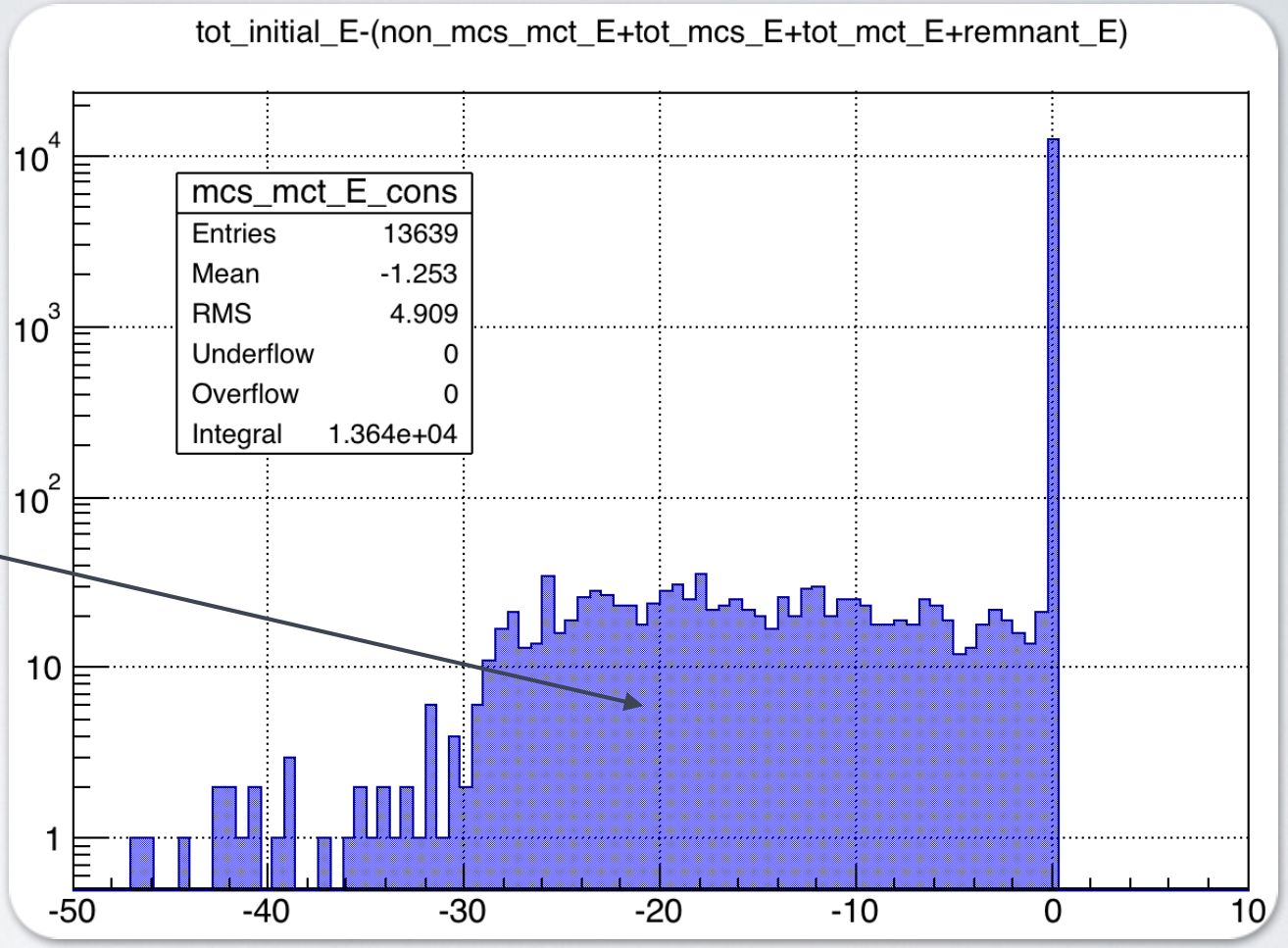
Here I sum the energy from all MCParticles from which MCShowers and MCTracks are not created.

This is the total energy of the remnant "blob" nucleus that has status code == 15.

Total final energy = all energy from MCShowers + all energy from MCTracks + all (MCParticle) energy from particles from which MCShowers and MCTracks are not created + remnant nucleus energy.

MCSHOWER + MCTRACK ENERGY CONSERVATION

This extra tens of MeV energy is not yet fully understood, but is *not* an feature of MCShower +MCTrack



Total final energy = all energy from MCShowers + all energy from MCTracks + all (MCParticle) energy from particles from which MCShowers and MCTracks are not created + remnant nucleus energy.

SUMMARY

- Shown how to demonstrate conservation of energy with GENIE input <—> GENIE output.
  - Don't yet understand the ~10s of MeV discrepancy.
- Shown how to demonstrate conservation of energy with GENIE output <—> GEANT4 energy depositions via MCShower and MCTrack objects.
- Hopefully these slides will save some people a few hours of histogramming to solve similar points of confusion.