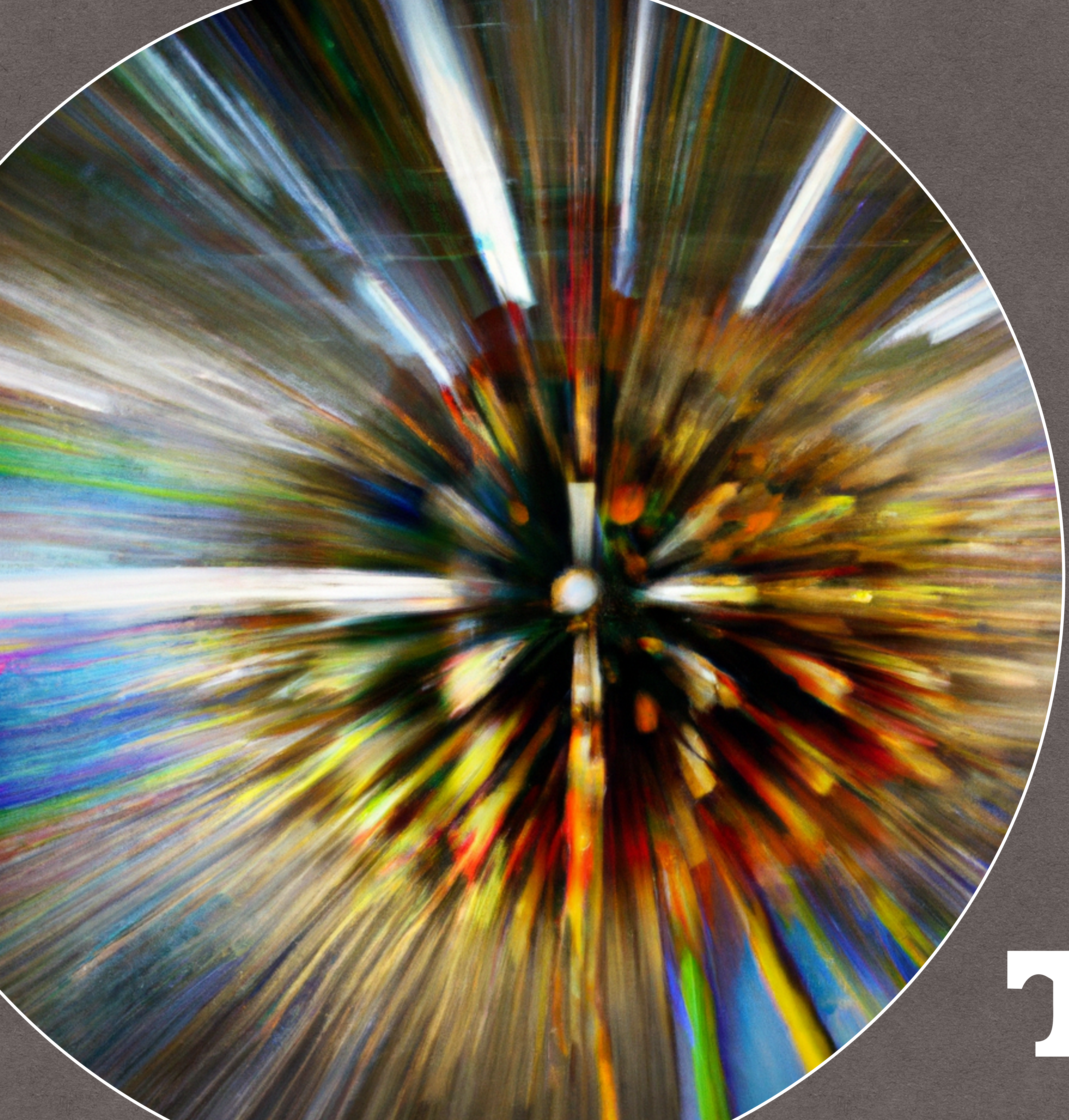


U.S. DEPARTMENT OF ENERGY | Office of Science

10 TeV MuCol

ECAL Calibration

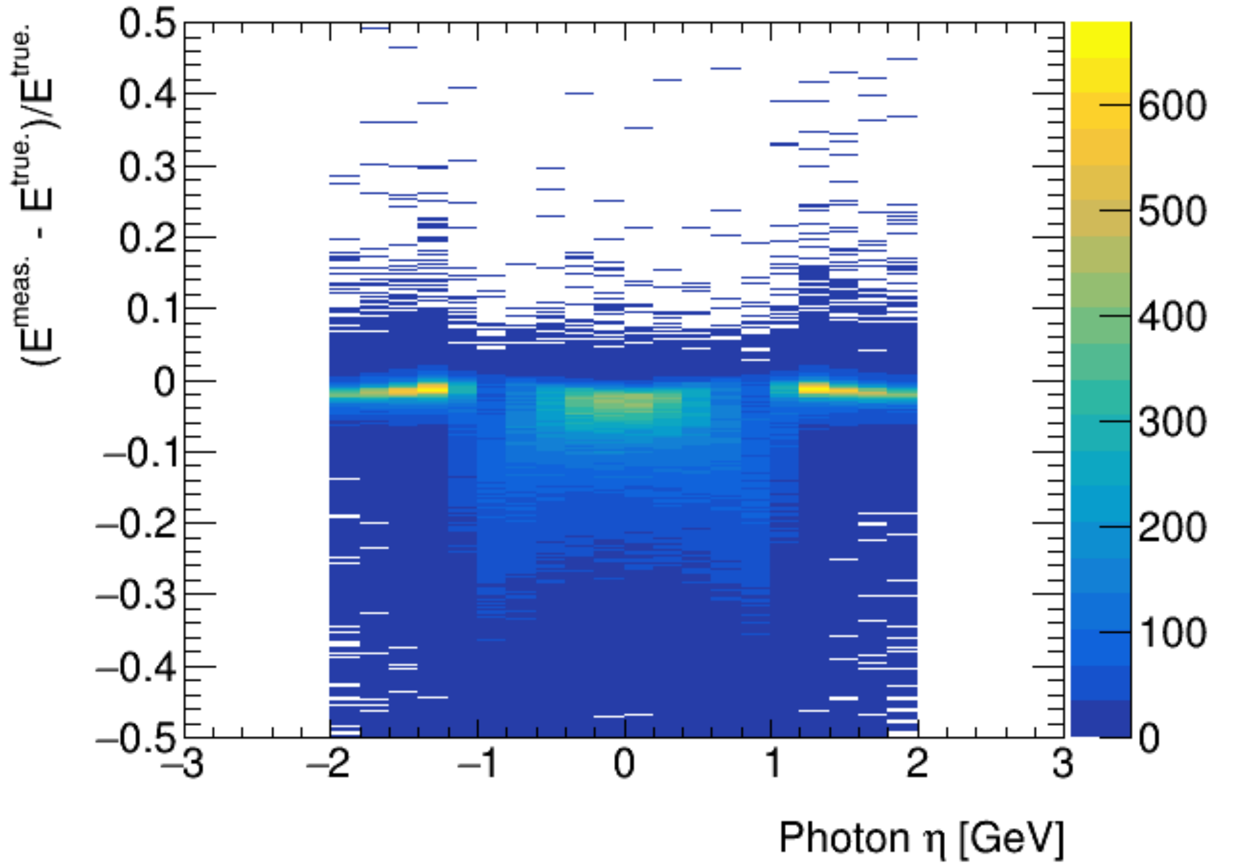
TOVA HOLMES, U. OF TENNESSEE
10 TEV MUCOL STUDIES
APRIL 17, 2024

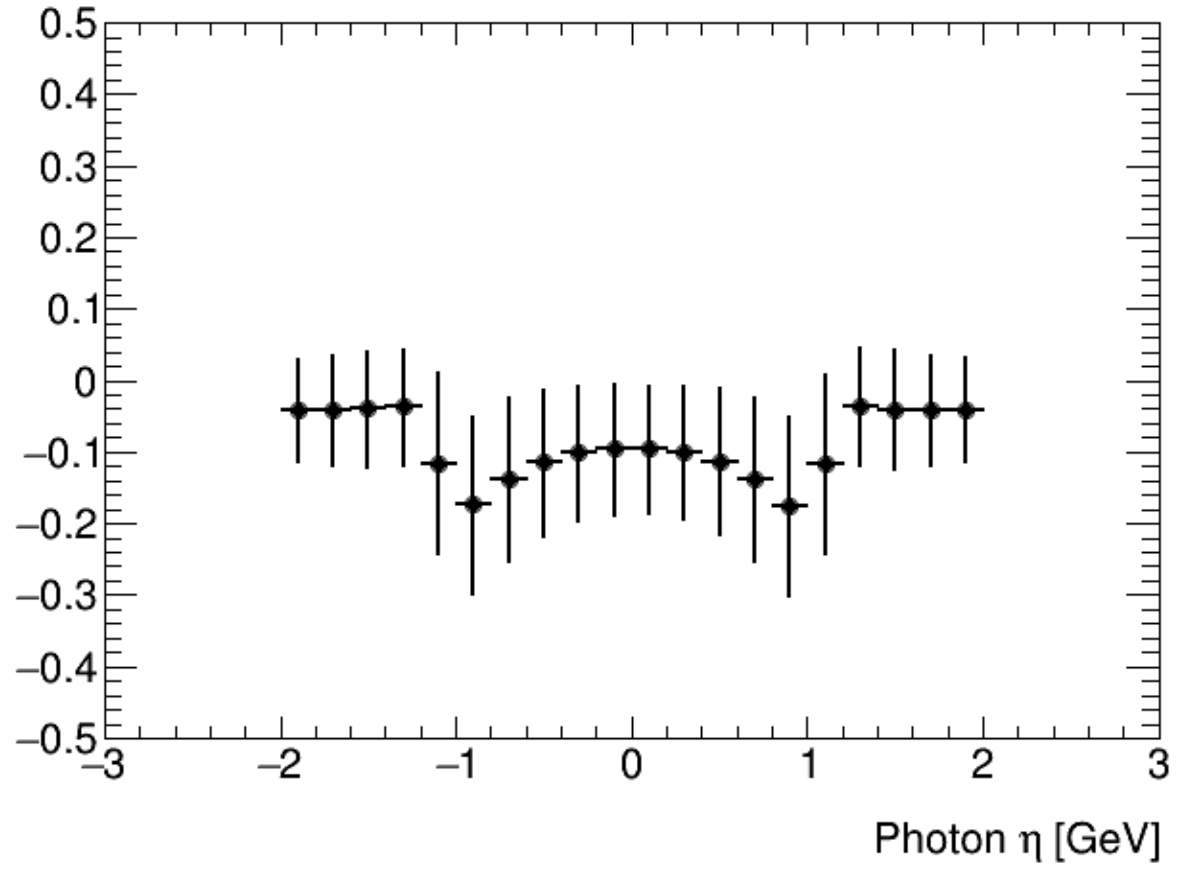
How does calibration work?

- What calibration stages are there?
  - First we have a flat sim->digi scaling factor that is currently the same everywhere
  - Then at the PFO cluster stage, we re-calibrate using the E_reco dependent response curves that Fede and I made
- Really need both:
  - First stage can correct for geometric dependence within a cluster, and can create consistent ratios between ECAL and HCAL once both are applied, which helps make consistent clusters from the combo
  - Second stage can correct for energy dependent effects, like loss in the solenoid
  - Could additionally later apply different calibrations for different objects

This week's work

- Last week found strong eta-dependence in ECAL energy response, and realized that this was due to increase (more accurate) material added to the barrel layers
- To properly handle this, need z/theta/eta-dependent sim-digi calibration factors
    - Can really only calculate these via full sum of ECAL energy compared to true photon, because we don't have a way of knowing how much energy "should" be deposited in each cell or layer
    - Making plots vs. theta (and eta, for fun). Making vs. z doesn't really make sense in this context, since an mcp has no fixed z
    - Using only the highest E slice (250-1000) so that we're less affected by energy loss before reaching the ECAL – will need to account for this at the cluster-level calibration
- Once this is in place, can run a similar procedure on HCAL, subtracting off calibrated ECAL energy

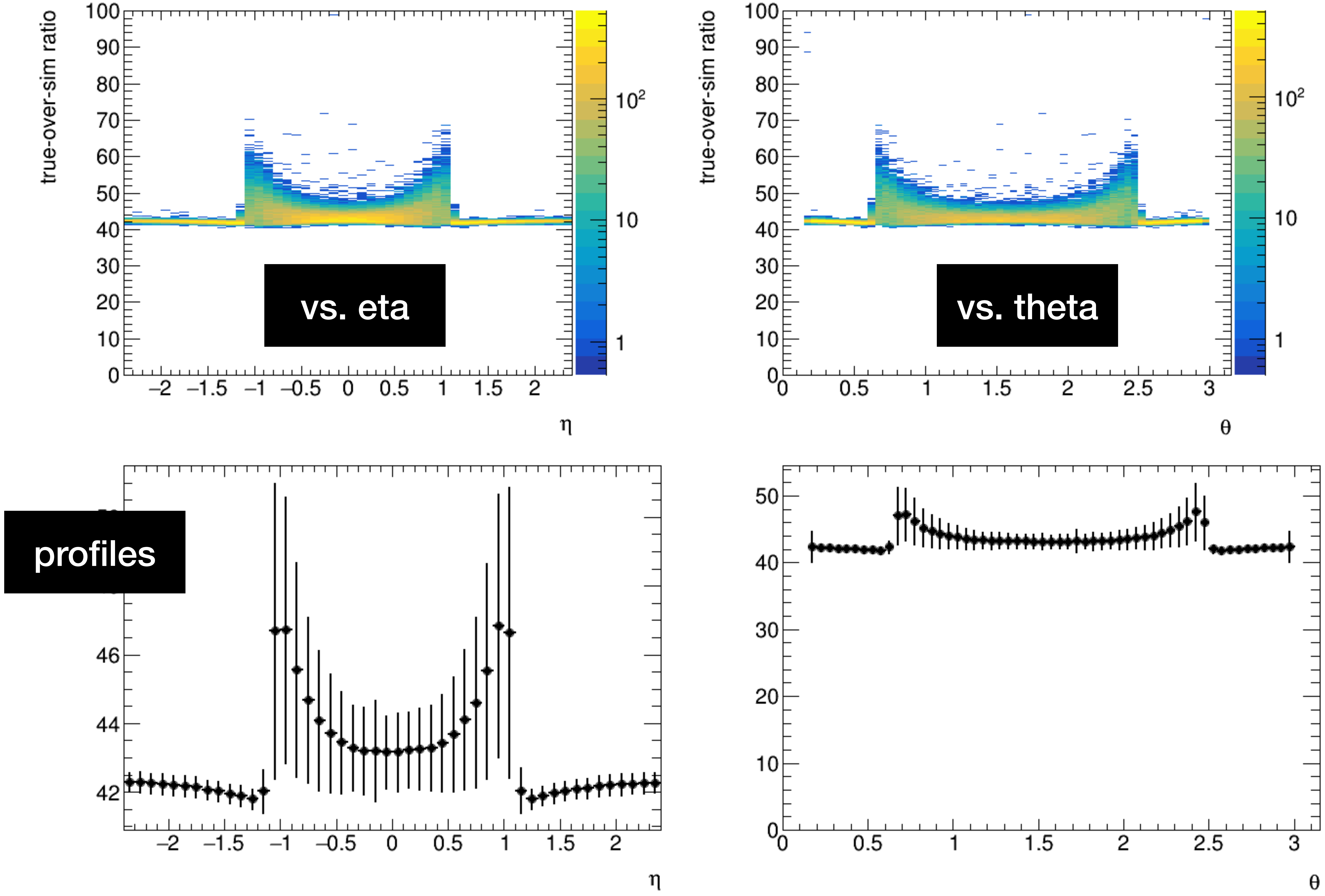




Technical details

- Using scripts found here: https://github.com/trholmes/mucolstudies
  - New script for this task: getSimDigiCalibration.py
- Only using these files: /data/fmeloni/DataMuC_MuColl10_v0A/reco/photonGun_E_250_1000
  - (Accessed on 4/15/2024)

Resulting distributions



- Can clearly see that we'll have worse energy resolution in the barrel still, even with a correction (material, solenoid)
- Made plots as a function of eta and theta – I think the binning in **theta** make more sense, and propose using those

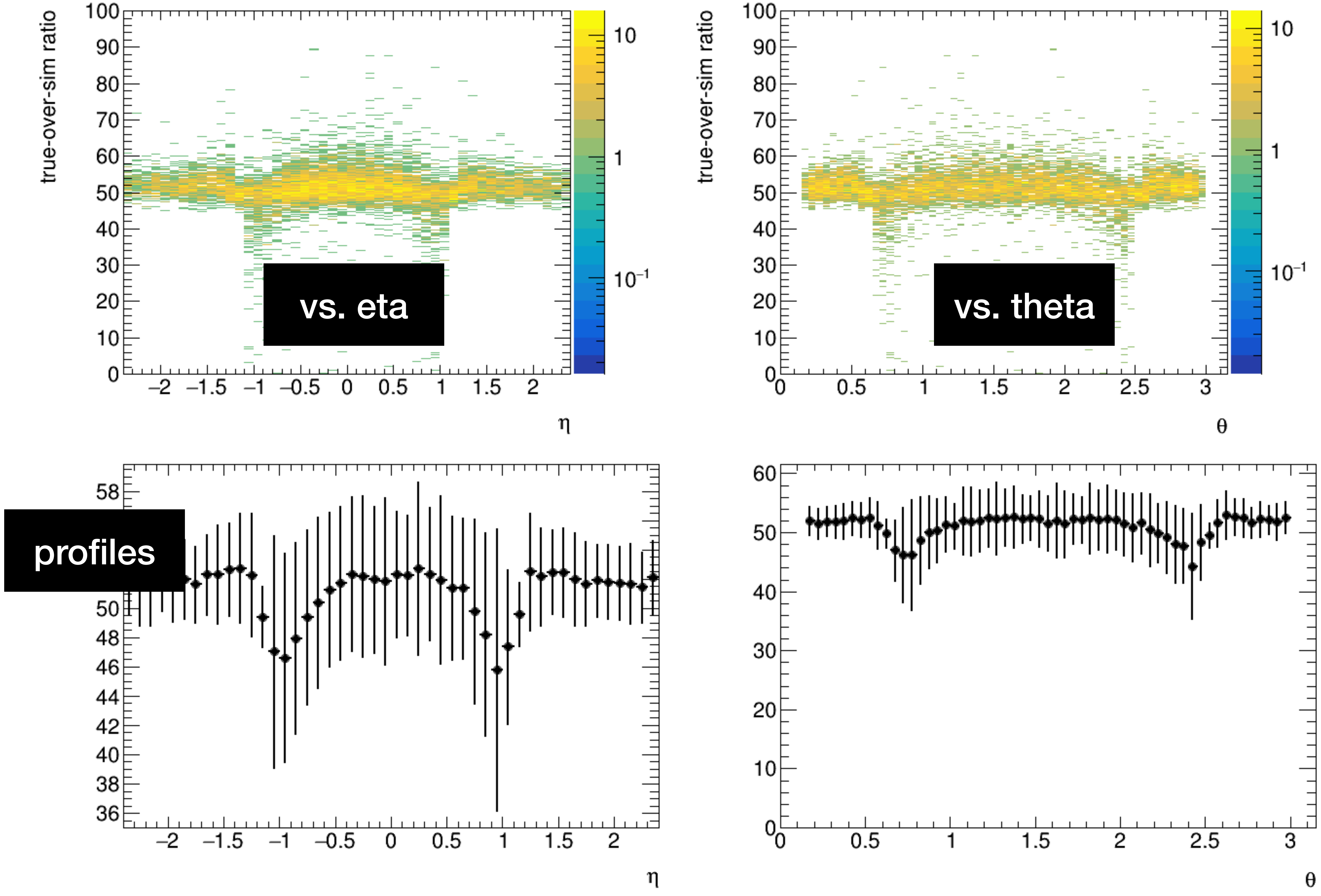
What comes next?

- Now need to apply these theta-dependent corrections at the sim level, then we can:
  - Re-calibrate at the ECAL cluster level using photons
  - Subtract off calibrated ECAL sim energy to do this same procedure on the HCAL
  - Re-calibrate at the HCAL cluster level using [pions? jets? first pions then jets?]
- Alternate first pass:
  - Can also try do the 2nd bullet point above before we re-run to save us some computing power

Did the first-pass HCAL calibration

- Made script getHCALSimDigiCalibration.py
  - Loads in ECAL and HCAL sim hits
  - Applies calibration to ECAL hits based on previous step and theta of MCP
    - This isn't exactly what we would do in our central software – there we'd apply it based on theta of hit! But should be OK for a first pass. Could consider iterating later.
  - Subtracts calibrated energy from ECAL hits from MCP, and uses remaining energy to define an HCAL sim scaling
- Used the equivalent slice (albeit pT now not E)
  - /data/fmeloni/DataMuC_MuColl10_v0A/reco/pionGun_pT_250_1000
  - This has lower stats (does pion have 10 events/file while photon has 100?)

Resulting distributions



vs. eta

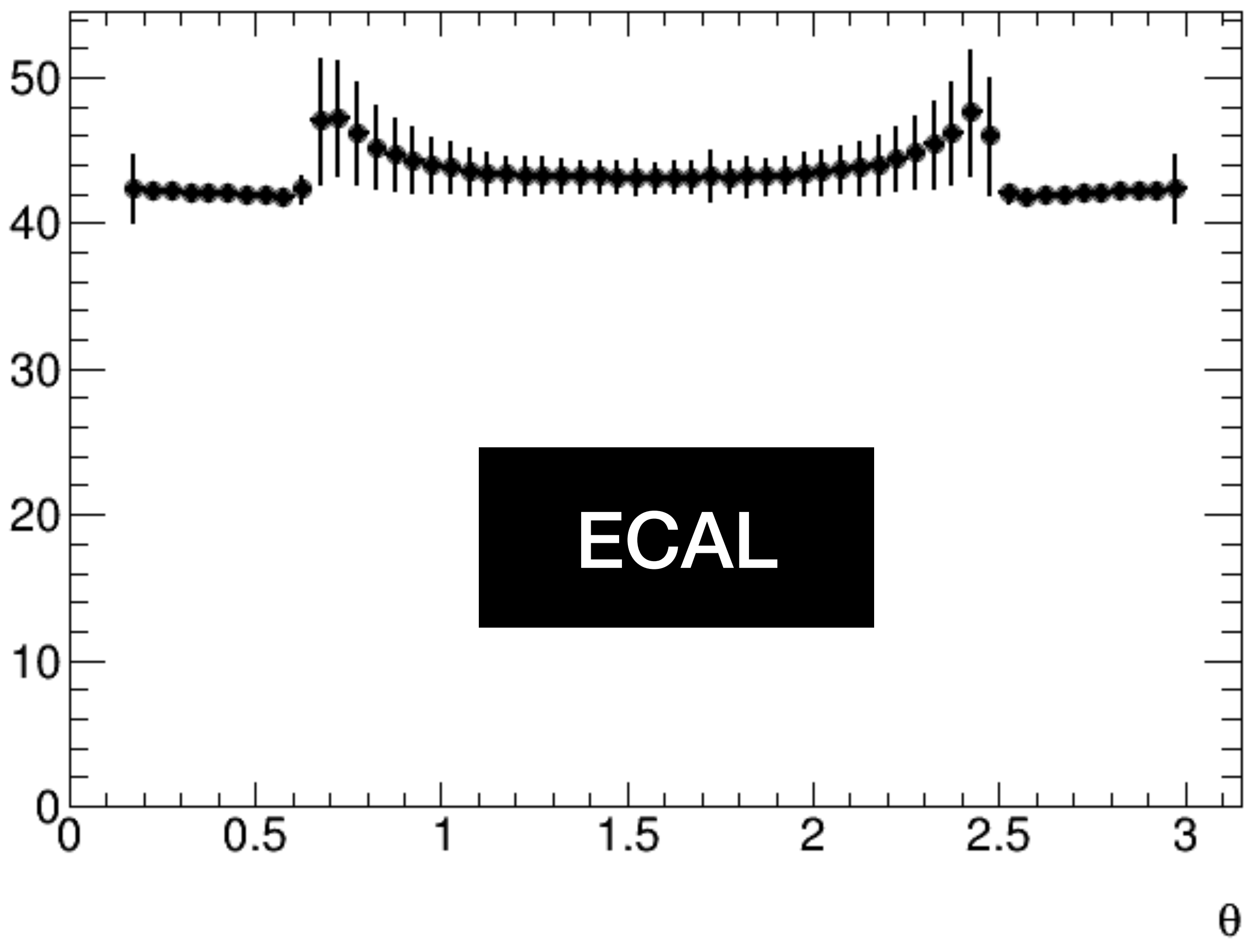
vs. theta

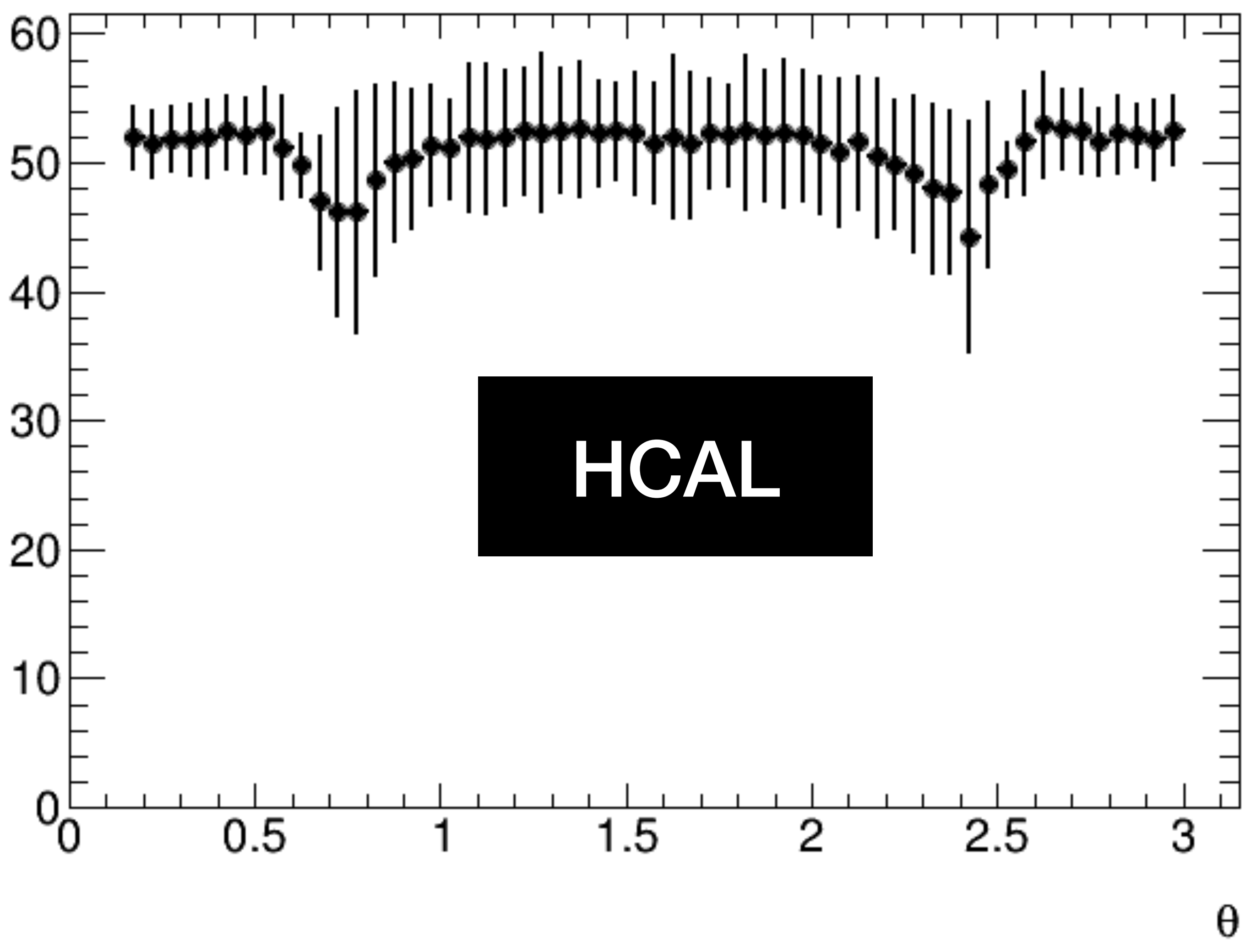
profiles

- Less eta dependence here, mostly issues in the transition region.
- (I'm not sure of the status of realism of materials in the barrel in HCAL.)

Next steps

- Now have a first pass set of calibration values for both ECAL and HCAL
  - ○ Need to make a system to input them into our workflow (Thomas?)
  - ○ Then can re-make cluster-level calibrations for:
    - Photons, charged/neutral pions, jets, etc.
  - ○ Can also then re-visit this HCAL calibration using the properly applied ECAL sim calibration, and see if it changes anything enough to edit it.


ECAL
θ


HCAL
θ