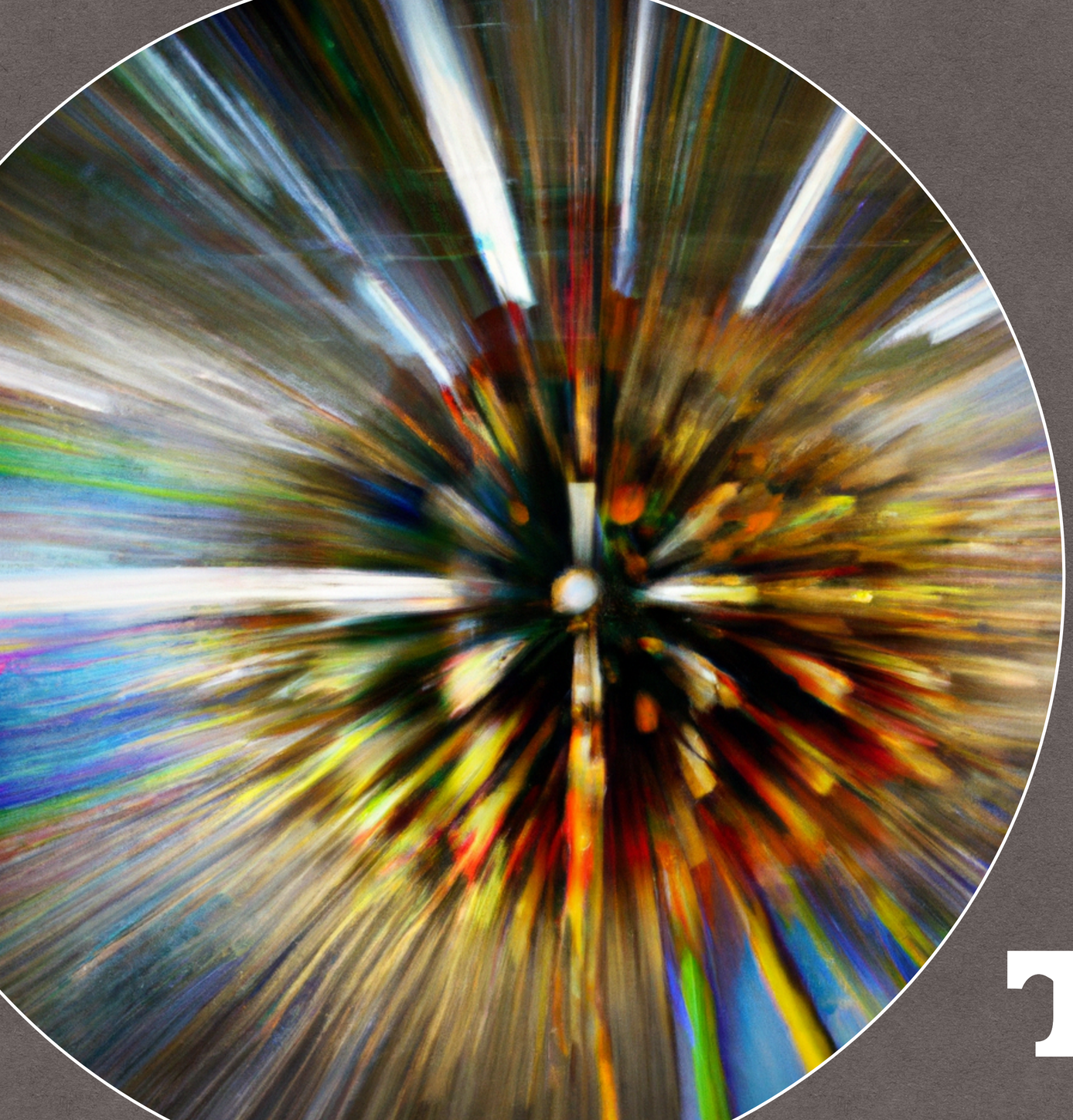


U.S. DEPARTMENT OF ENERGY | Office of Science

10 TeV MuCol

MEMORY CONSUMPTION ISSUES + electrons

TOVA HOLMES, U. OF TENNESSEE

10 TEV MUCOL STUDIES

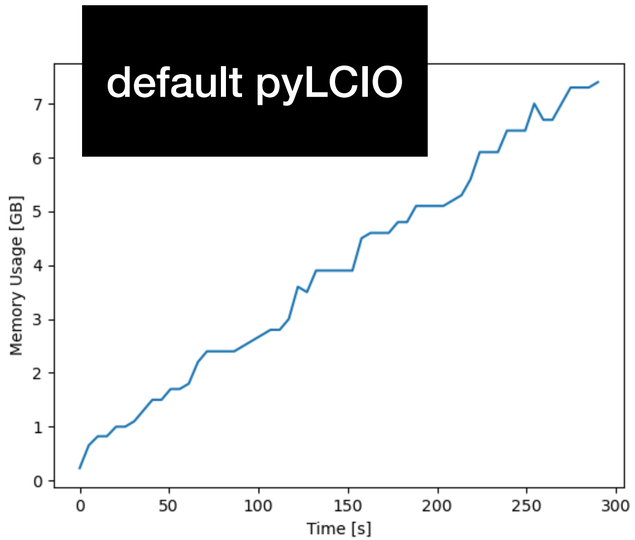
OCTOBER 7, 2023

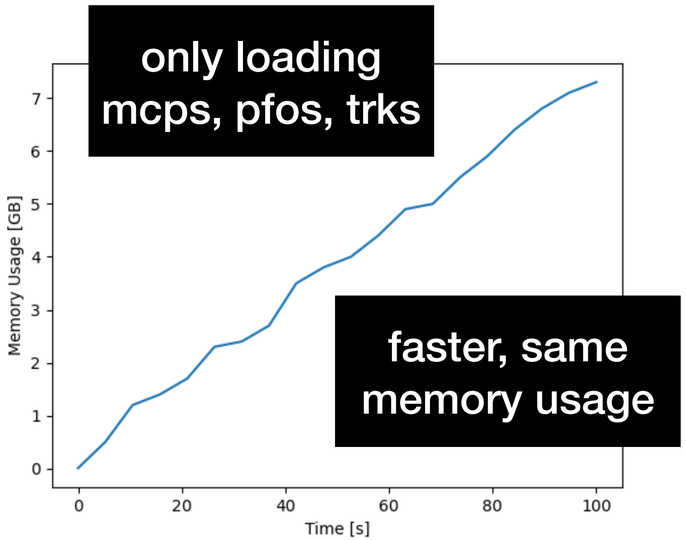
Memory consumption issues

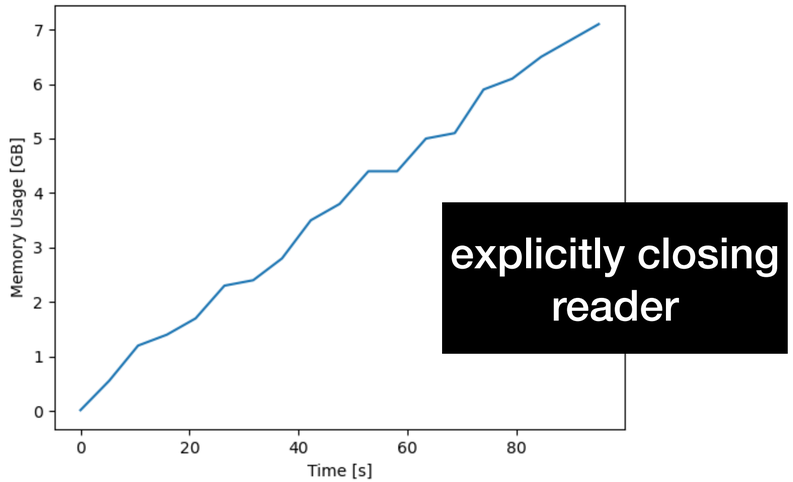
- Trying to run over all the electron samples w/o BIB
  - Tried both key4hep and slcio and in both had memory issues
  - Tried a simple example with nothing in the loop, just opening files in slcio

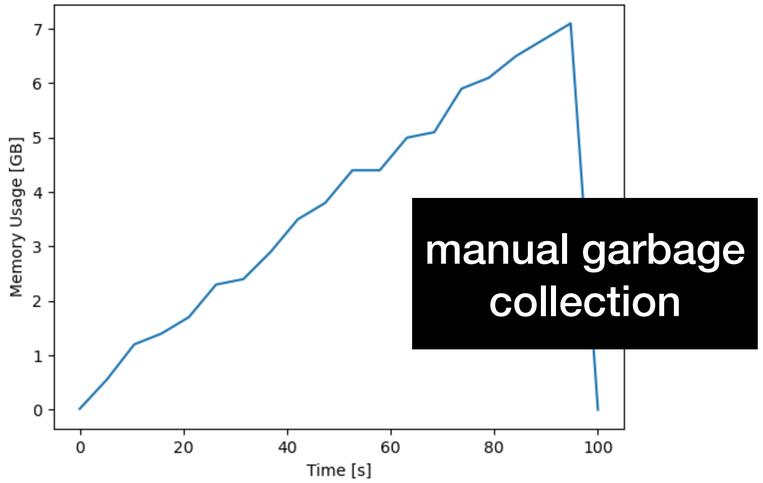
```
 + test.py
import glob
import gc
max_events = 1000

# pylcio simplified setup
import pyLCIO
files = glob.glob("/data/fmeloni/DataMuC_MuColl10_v0A/reco/electronGun_pT_250_1000/*.slcio")
i = 0
for f in files:
    if i >= max_events: break
    reader = pyLCIO.IOIMPL.LCFactory.getInstance().createLCReader()
    reader.setReadCollectionNames(["MCParticle", "PandoraPFOs", "SiTracks_Refitted"])
    reader.open(f)
    for event in reader:
        if i >= max_events: break
        if i%100 == 0: print("Processing event ", i)
        i += 1
    reader.close()
    gc.collect()

print("Done.")
~
~
~
~
~
~
```

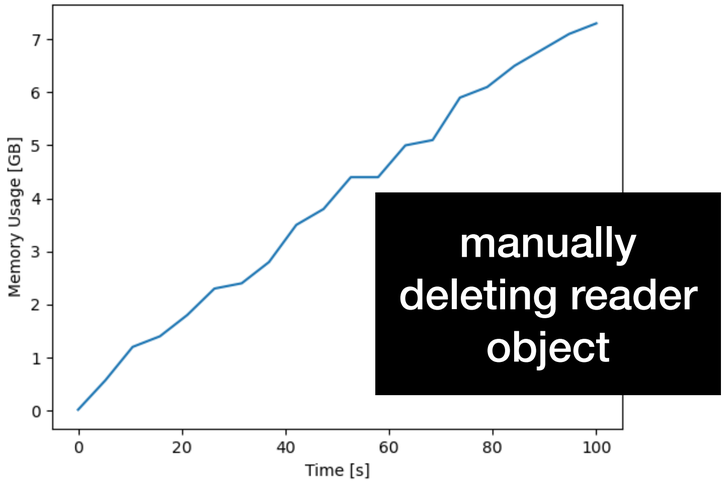
To run 1000 events, nothing done inside loop, pyLCIO

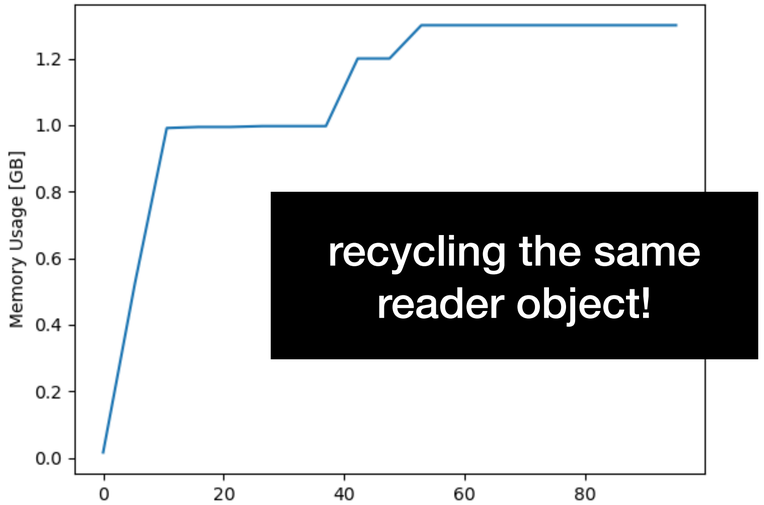






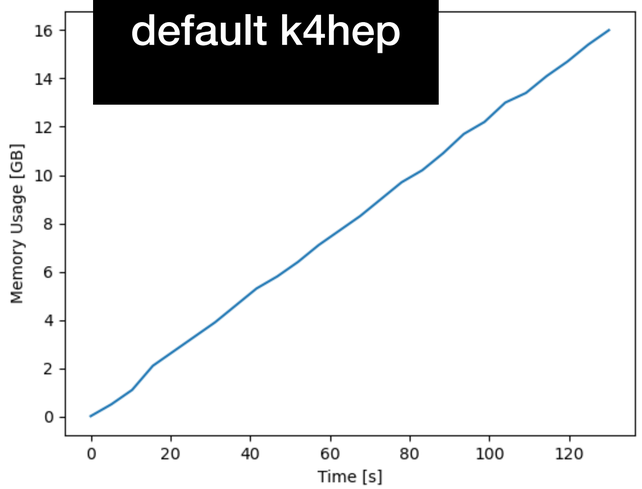




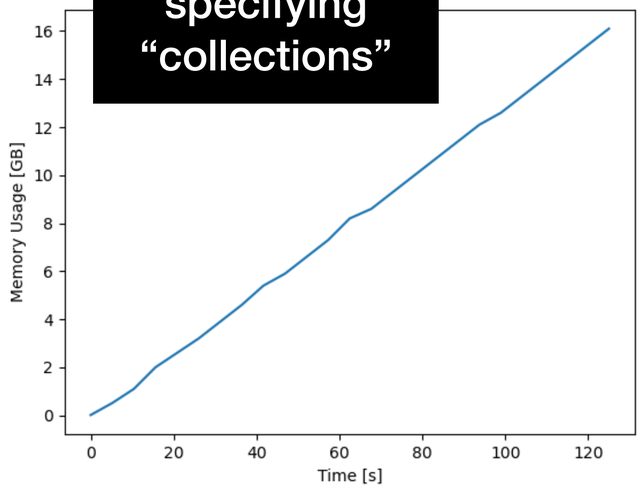


To run 1000 events, nothing done inside loop, k4hep

default k4hep



specifying "collections"



- Default is using the podio.root_io.Reader(filename) function
  - No ability to load files separately from initializing reader (I think?) so cannot do the same fix
  - Can specify collections
    - reader.collections = ["MCParticle", "PandoraPFOs", …]
    - Doesn't do anything (maybe needs to be called before opening a file, but can't because initialization is the only way to pass a filename AFAIK)
      - Can still run code that accesses un-specified collections

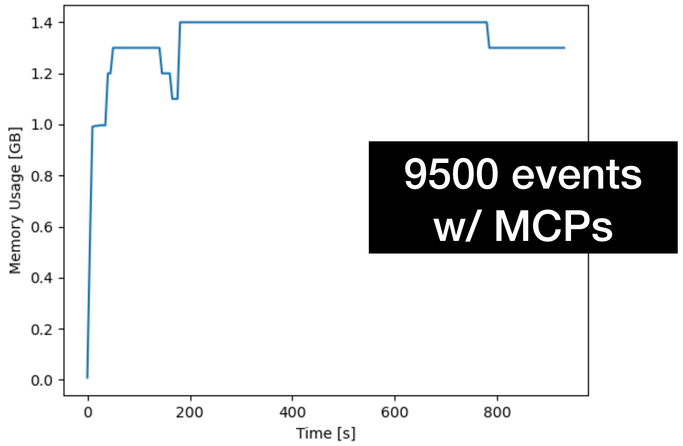
Without fix, runs faster but consumes more memory compared to pyLCIO

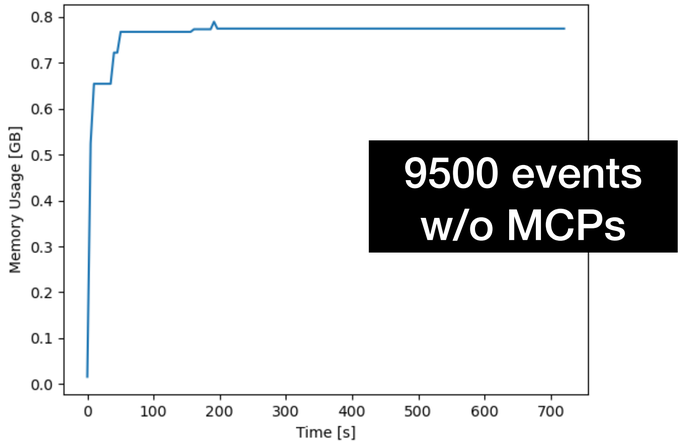
Possible solution is to work with the ROOTFrameReader() directly and have more tunability

Memory consumption issues

- In the end, was able to get OK memory performance with pyLCIO
  - Didn't have access to the same knobs in key4hep
  - Still very slow!! (around 10 events/s)
  - Not loading in MCPs halves memory but not much speed change (around 13 events/s)
  - Merging files may help with this

```
 + test.py
import glob
#import gc
max_events = 10000

# pylcio simplified setup
import pyLCIO
files = glob.glob("/data/fmeloni/DataMuC_MuColl10_v0A/reco/electronGun_pT_250_1000/*.slcio")
i = 0

reader = pyLCIO.IOIMPL.LCFactory.getInstance().createLCReader()
#reader.setReadCollectionNames(["MCParticle", "PandoraPFOs", "SiTracks_Refitted"])
reader.setReadCollectionNames(["PandoraPFOs", "SiTracks_Refitted"])

for f in files:
    if i >= max_events: break
    reader.open(f)
    for event in reader:
        if i >= max_events: break
        if i%100 == 0: print("Processing event ", i)
        i += 1
    reader.close()
    #del(reader)
    #gc.collect()

print("Done.")
```





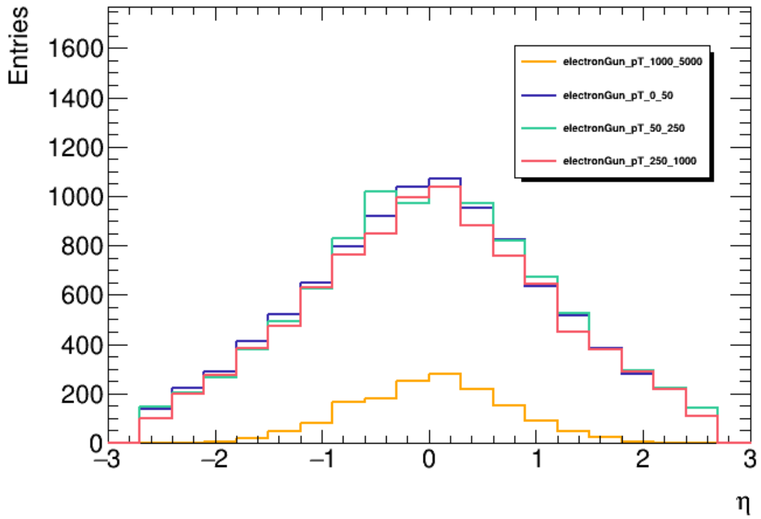
Technical details

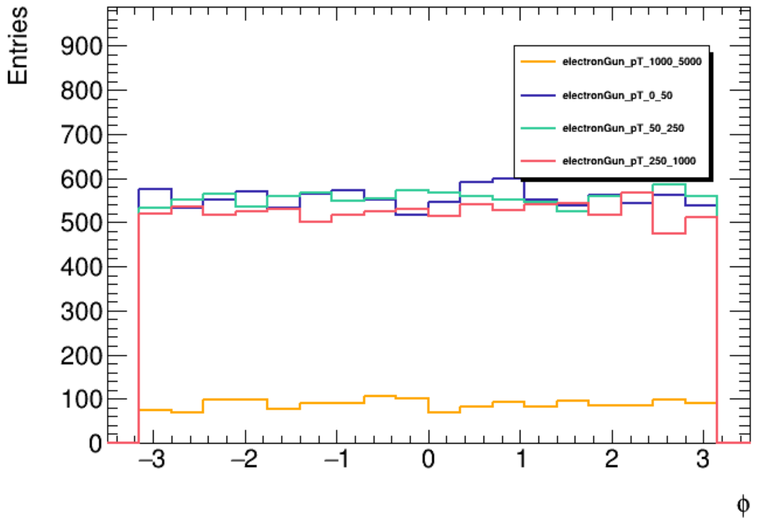
- Plotting code here:
  - https://github.com/trholmes/mucolstudies/blob/main/electronStudies.py
- Running on input:
  - /data/fmeloni/DataMuC_MuColl10_v0A/k4reco/electronGun*.root (10 TeV, no BIB)
  - Using k4hep files – confused about how to do LCRelations when the collections do exist
  - 4 slices:
    - 0-50, 50-250, 250-1000, 100-5000 GeV
    - All plotted separately without weighting

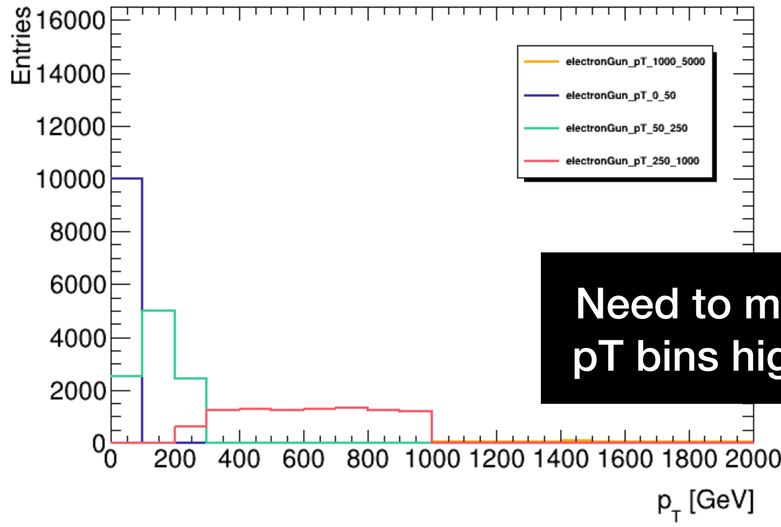
MCP electron plots

Everything looks normal

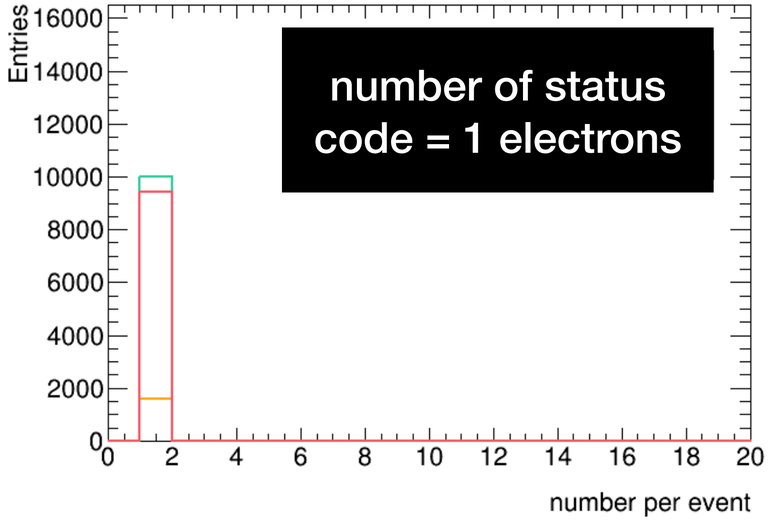
No BIB



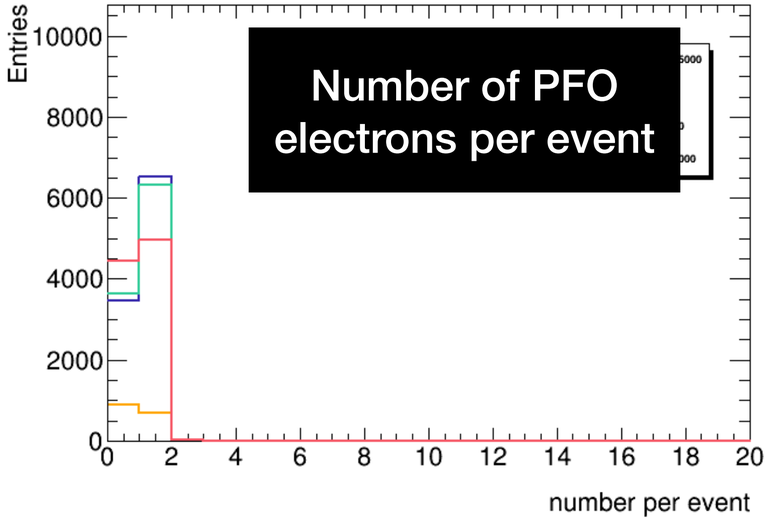




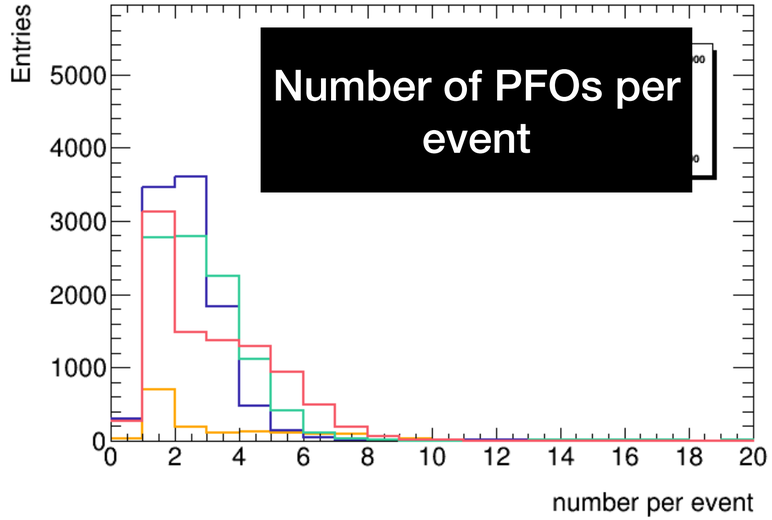
Need to make pT bins higher



number of status code = 1 electrons



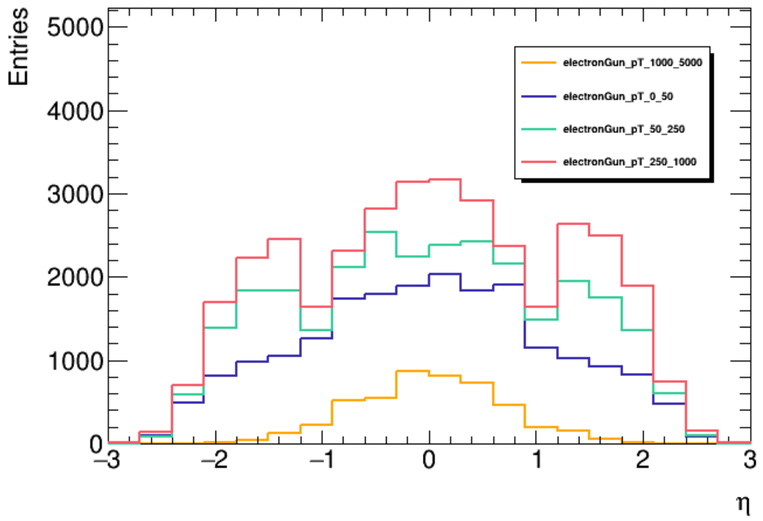
Number of PFO electrons per event

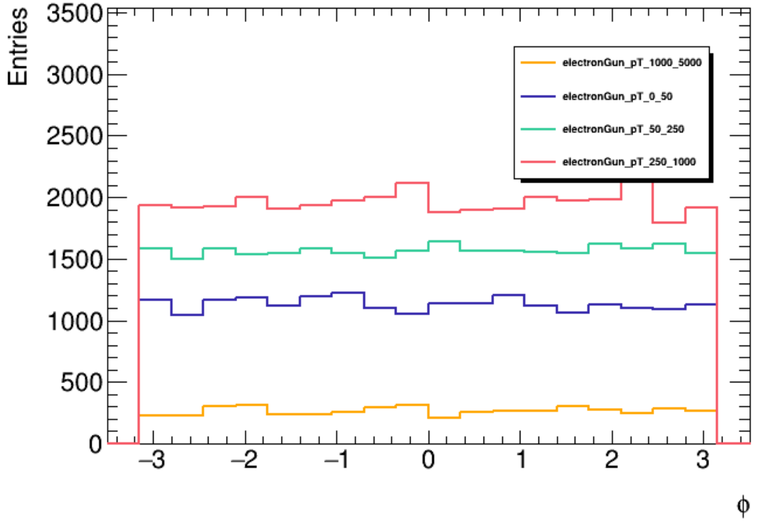


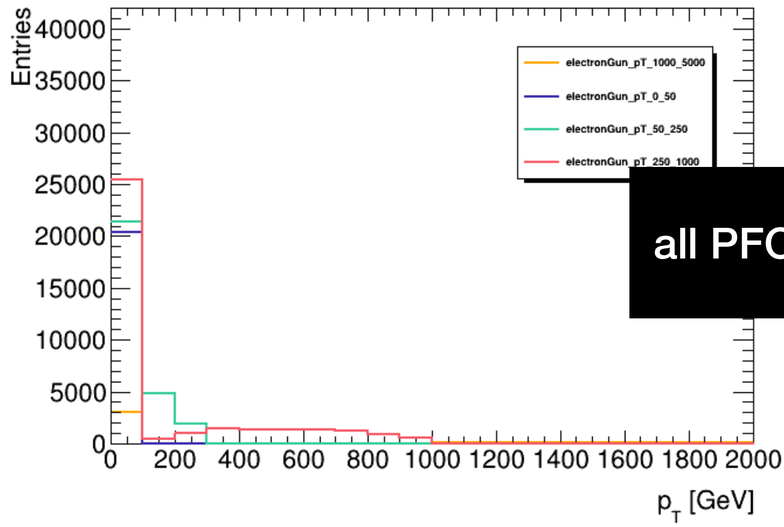
Number of PFOs per event

PFO plots

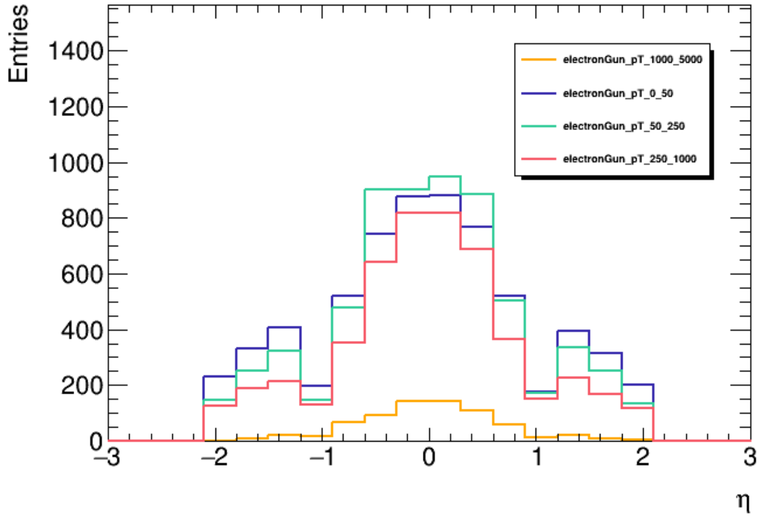
No BIB

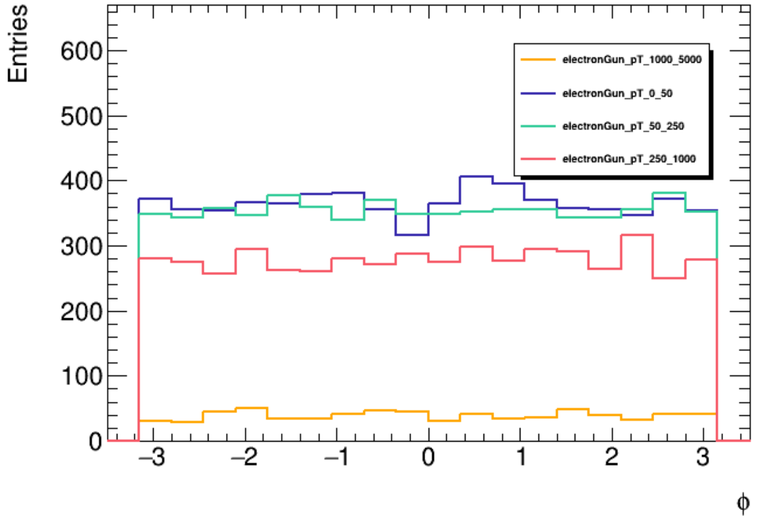


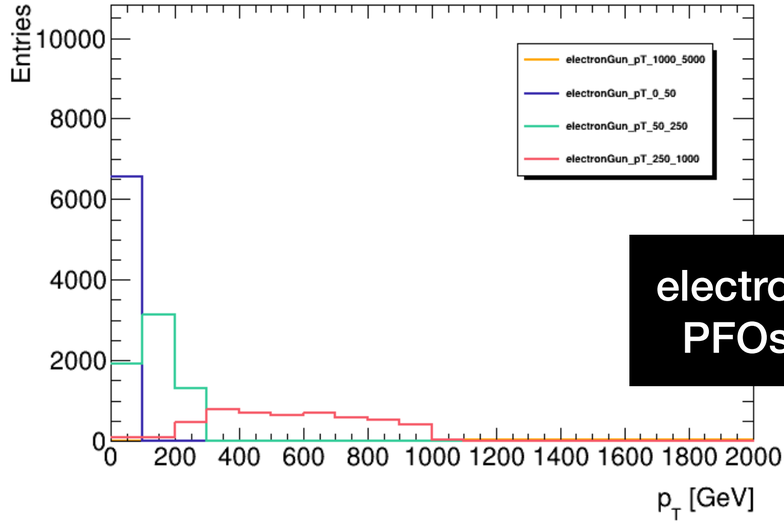




all PFOs



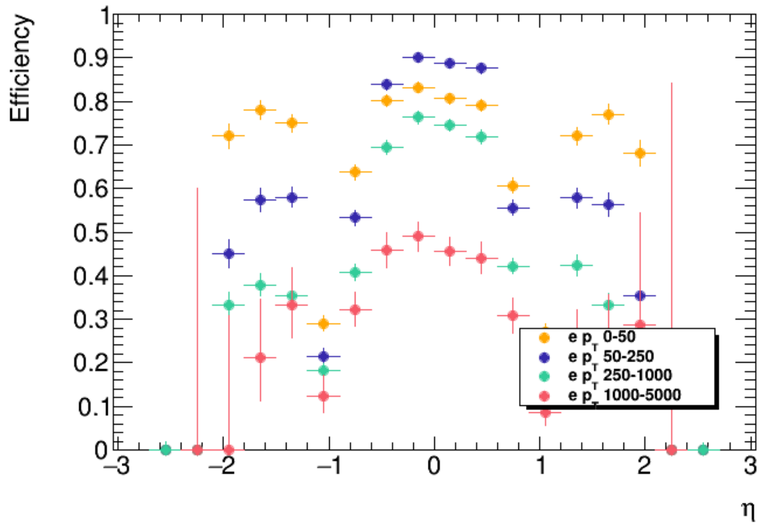


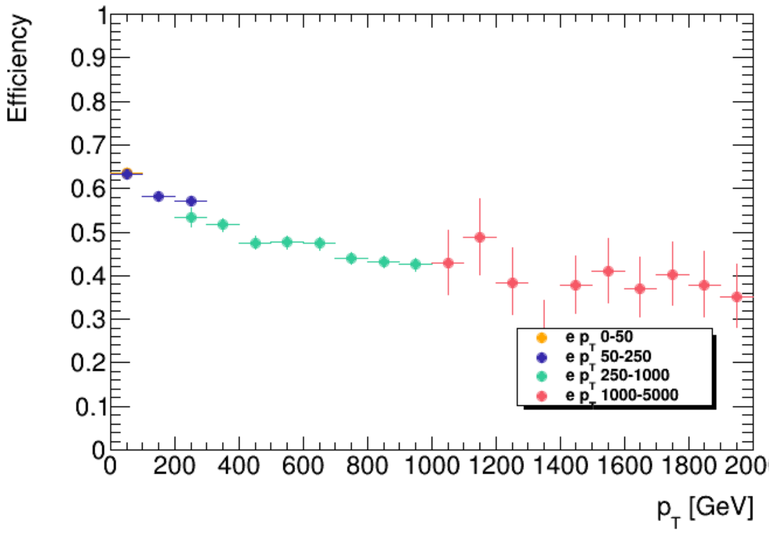


electron PFOs

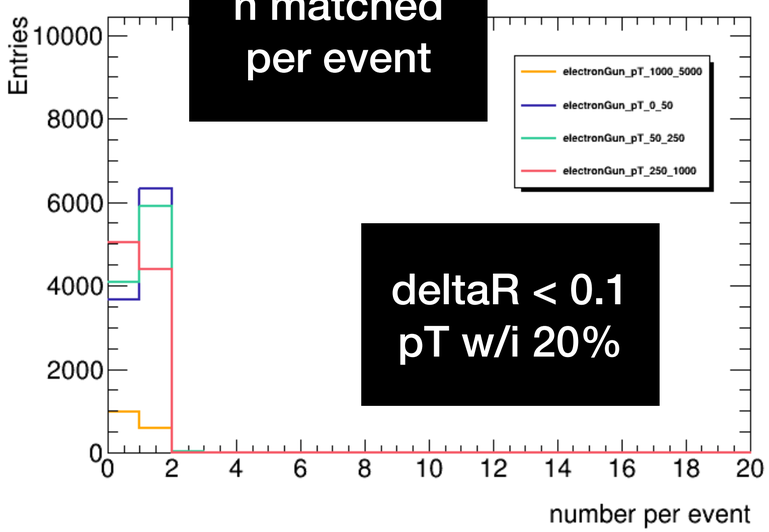
Efficiencies for matched electrons

No BIB

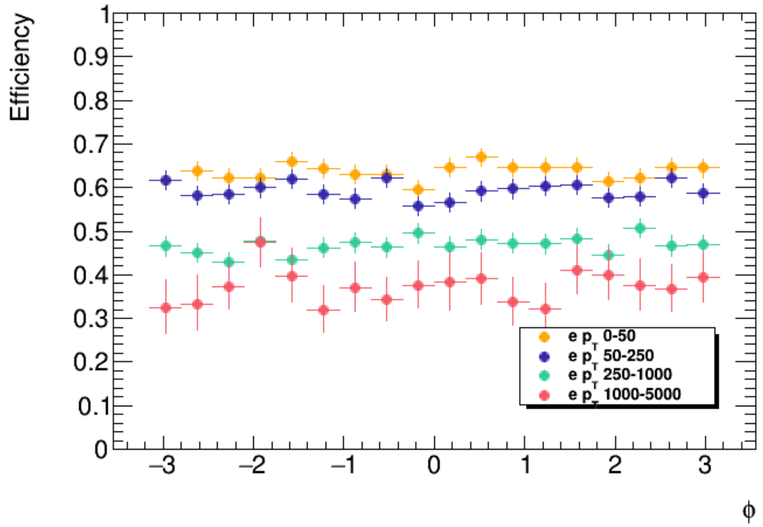




n matched per event



deltaR < 0.1
pT w/i 20%



Low pT objects relatively flat in eta, but high pT have low efficiency in endcaps.

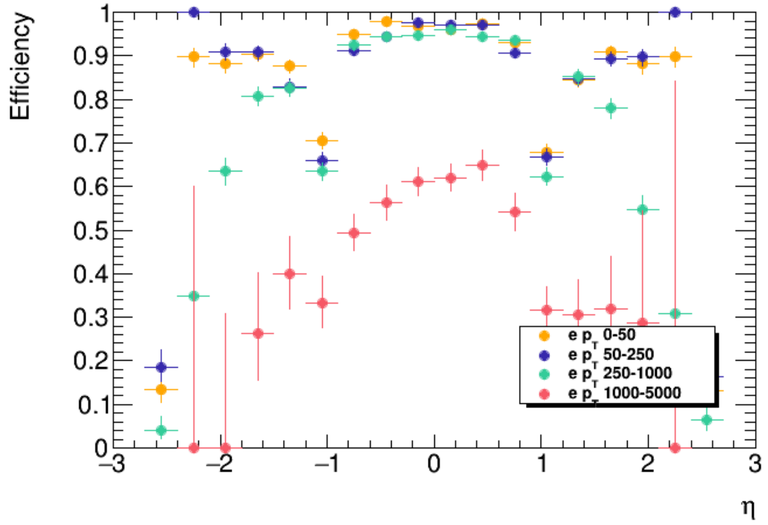
No distinct phi features.

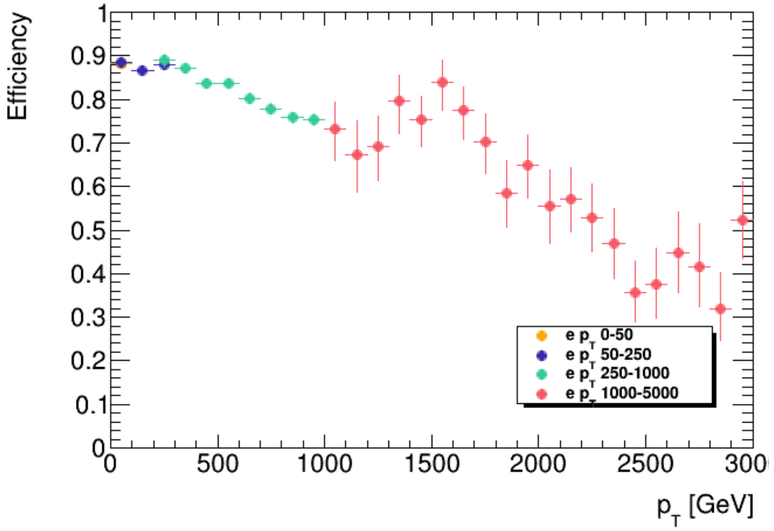
Transition region 😢

note: small error in these plots because I very occasionally match more than one electron — need to update spatial matching

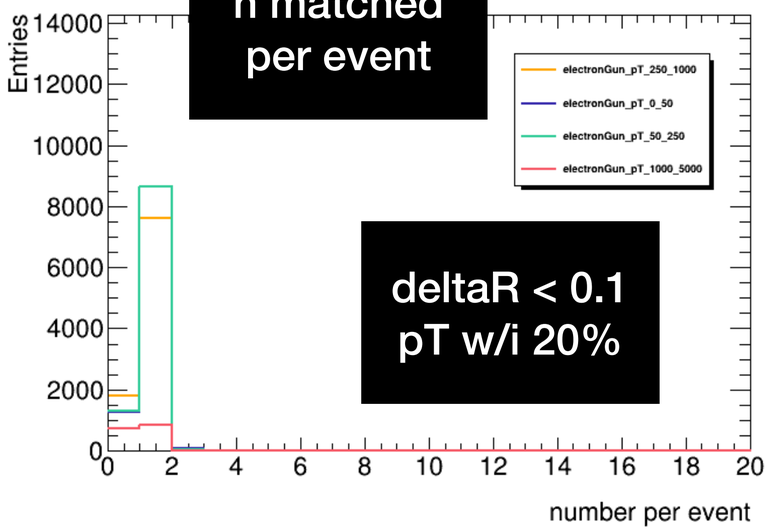
Efficiencies for matched PFOs (no el requirement)

No BIB

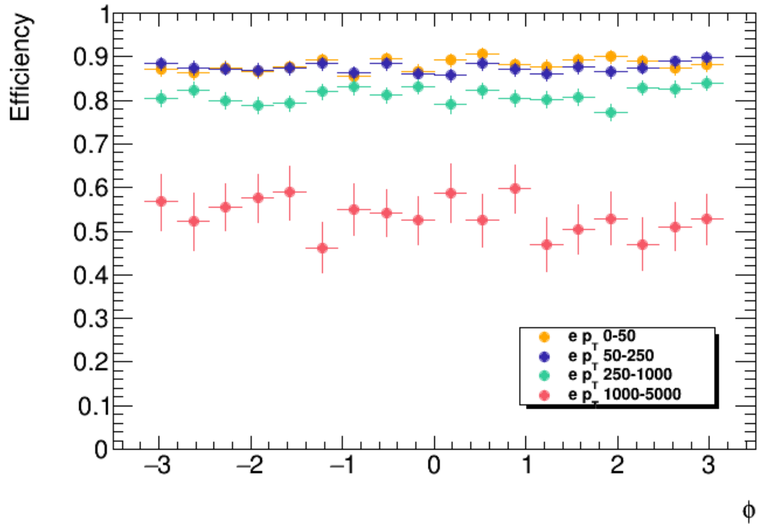




n matched per event



deltaR < 0.1
pT w/i 20%

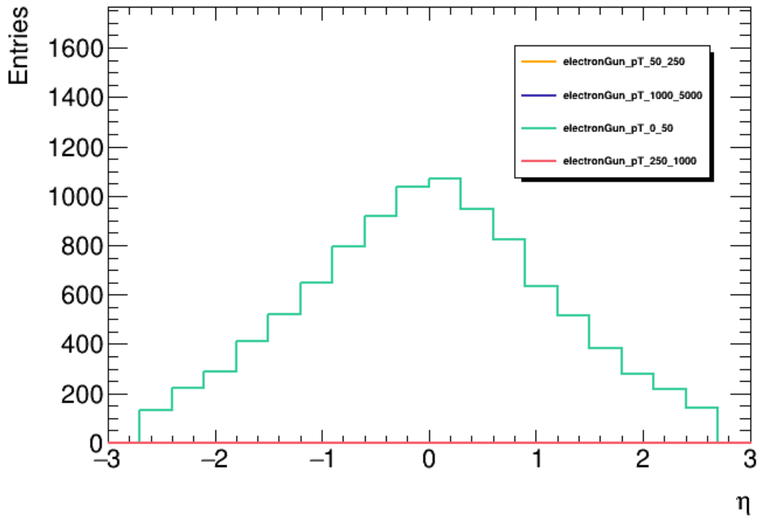


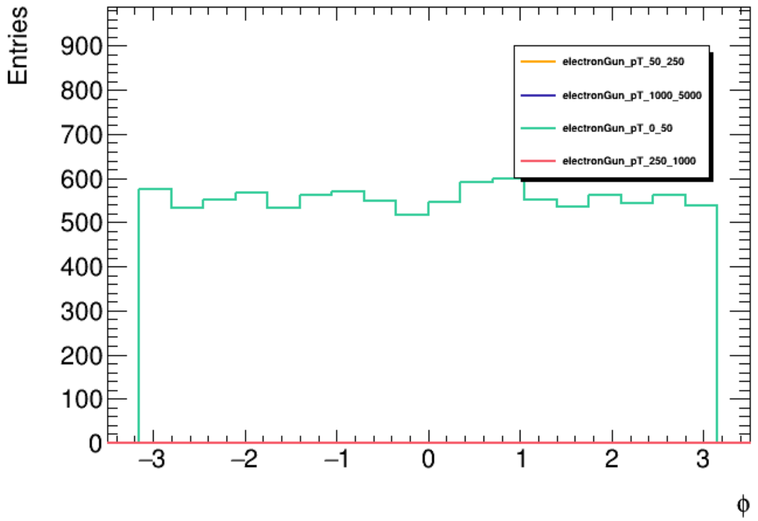
Now do well up to 250 GeV pT
Interesting shapes, maybe depend on my pT criterion – more studies needed!

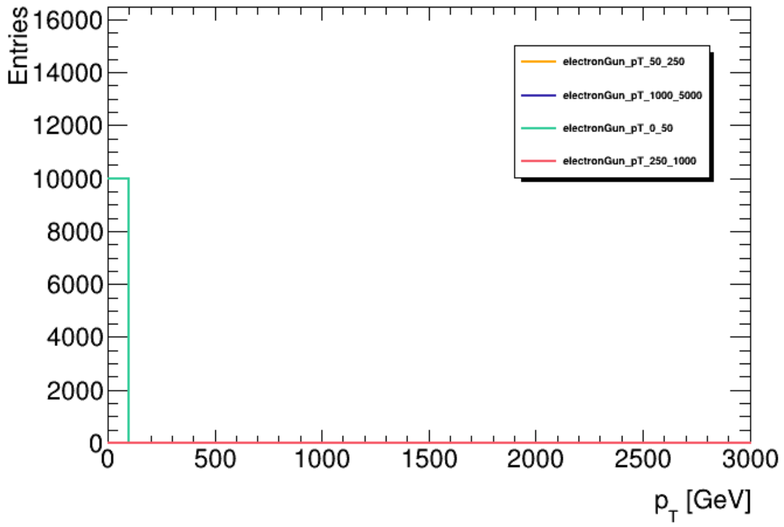
note: small error in these plots because I very occasionally match more than one electron — need to update spatial matching

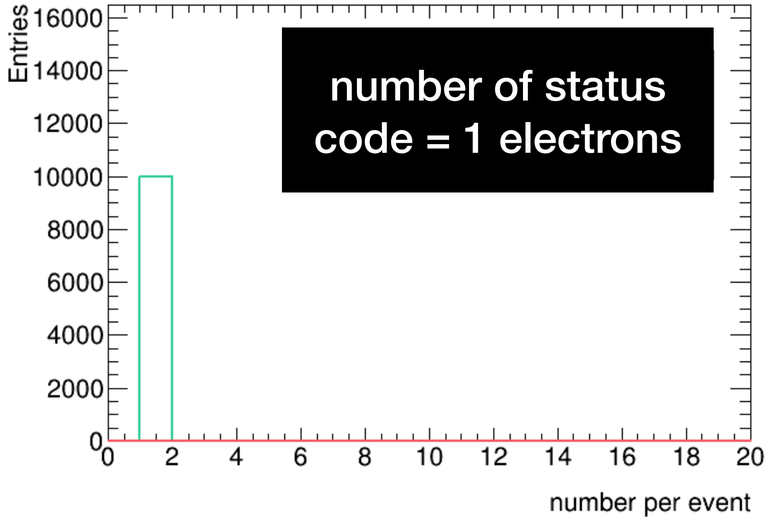
MCP electron plots

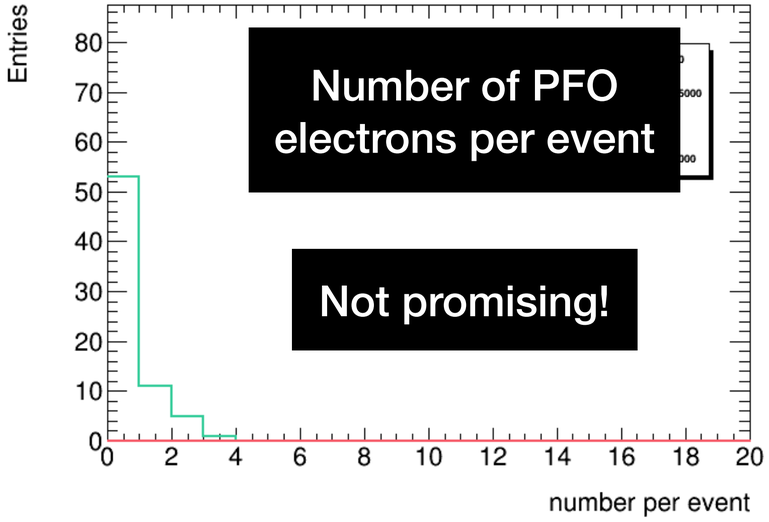
with BIB







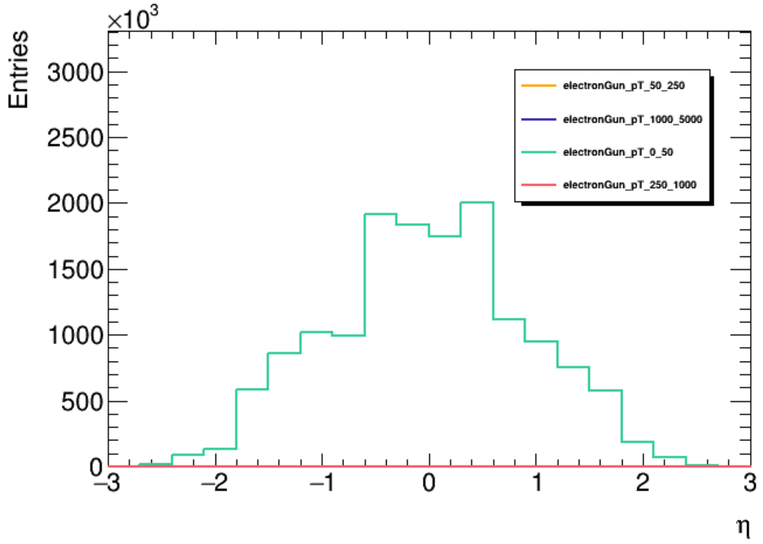


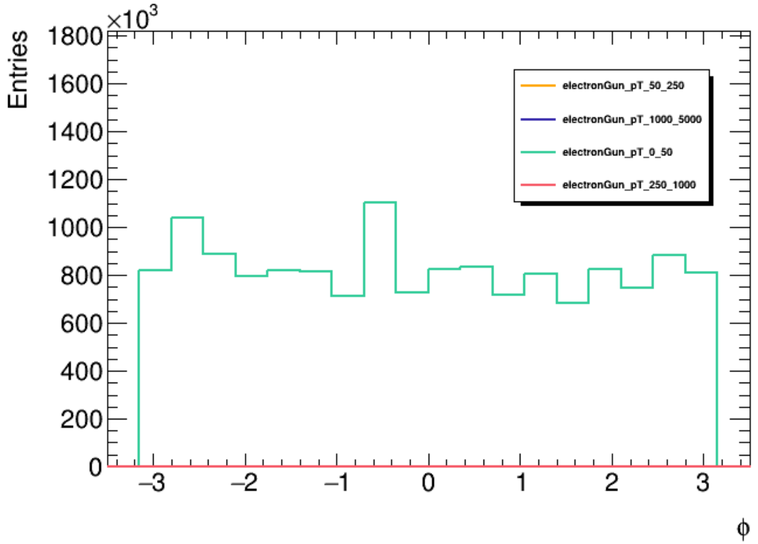


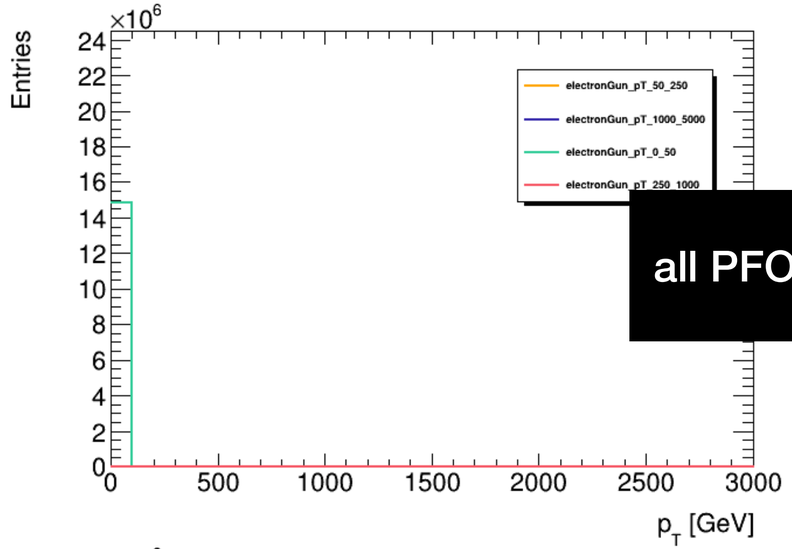
Only lowest pT sample exists

PFO plots

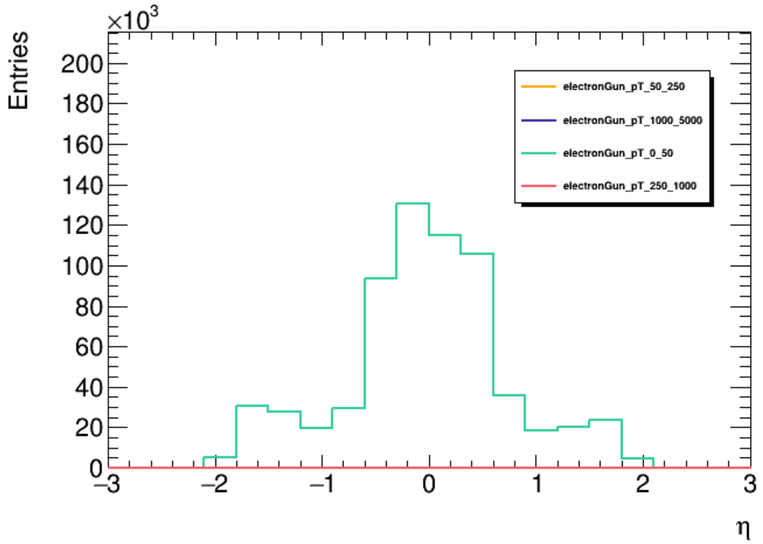
with BIB

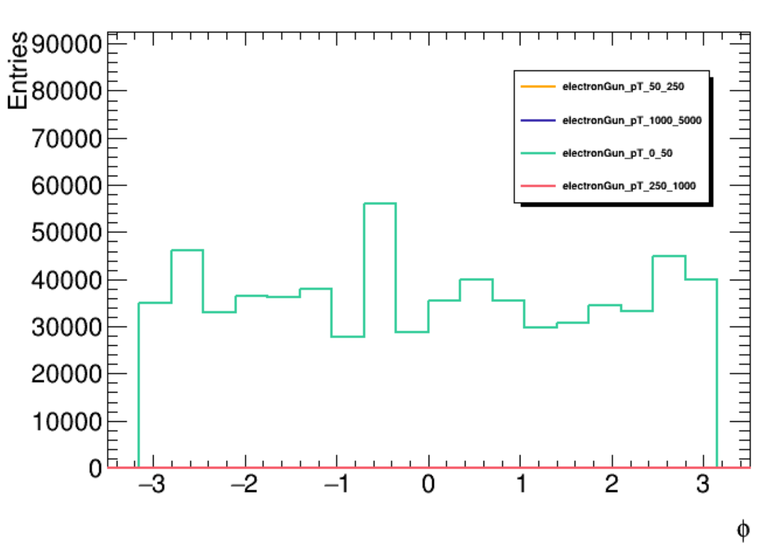


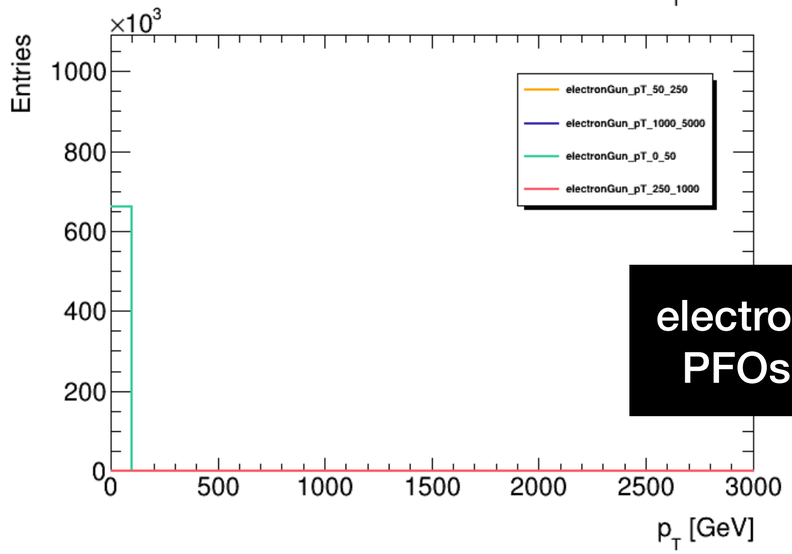




all PFOs



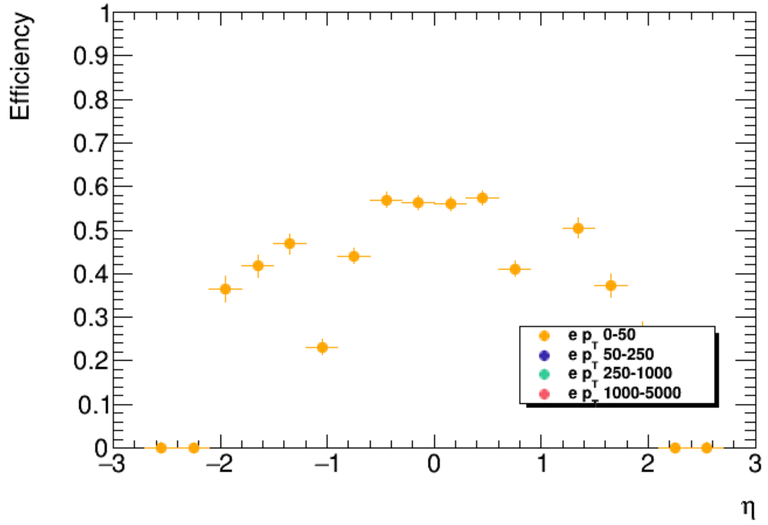


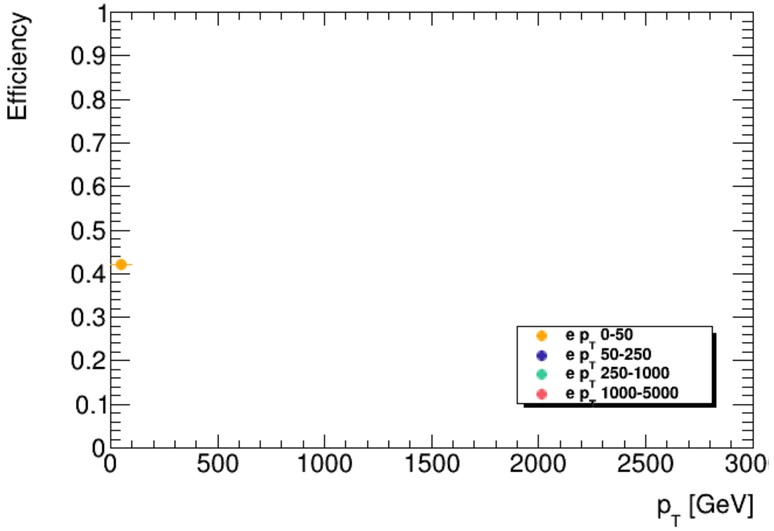


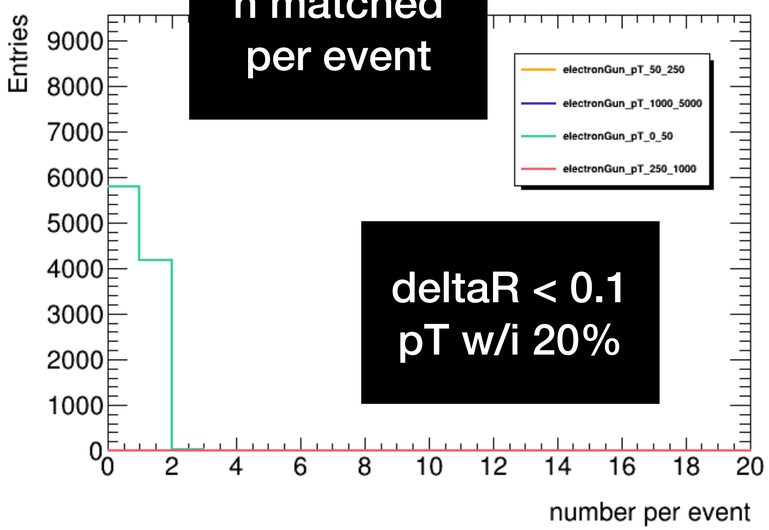
electron PFOs

Efficiencies for matched electrons

with BIB

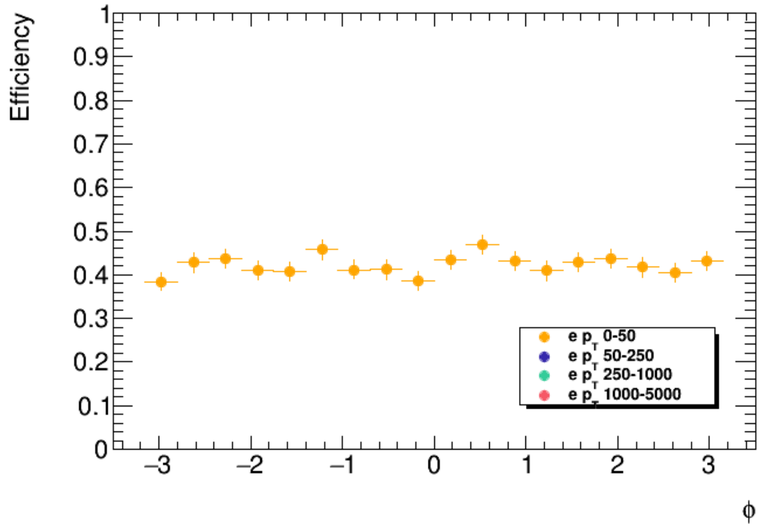






n matched per event

deltaR < 0.1
pT w/i 20%



Total efficiency drops from ~60% to ~40%

matching seems to work pretty well here