Level-3 Trigger Update

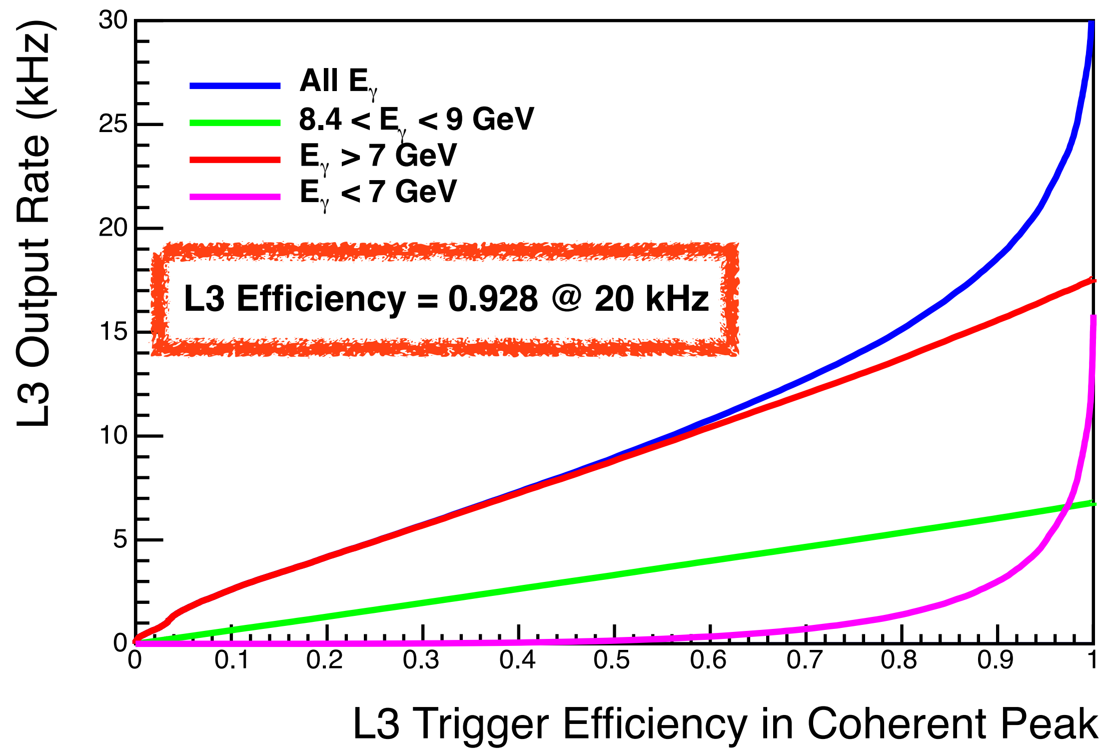
Justin Stevens

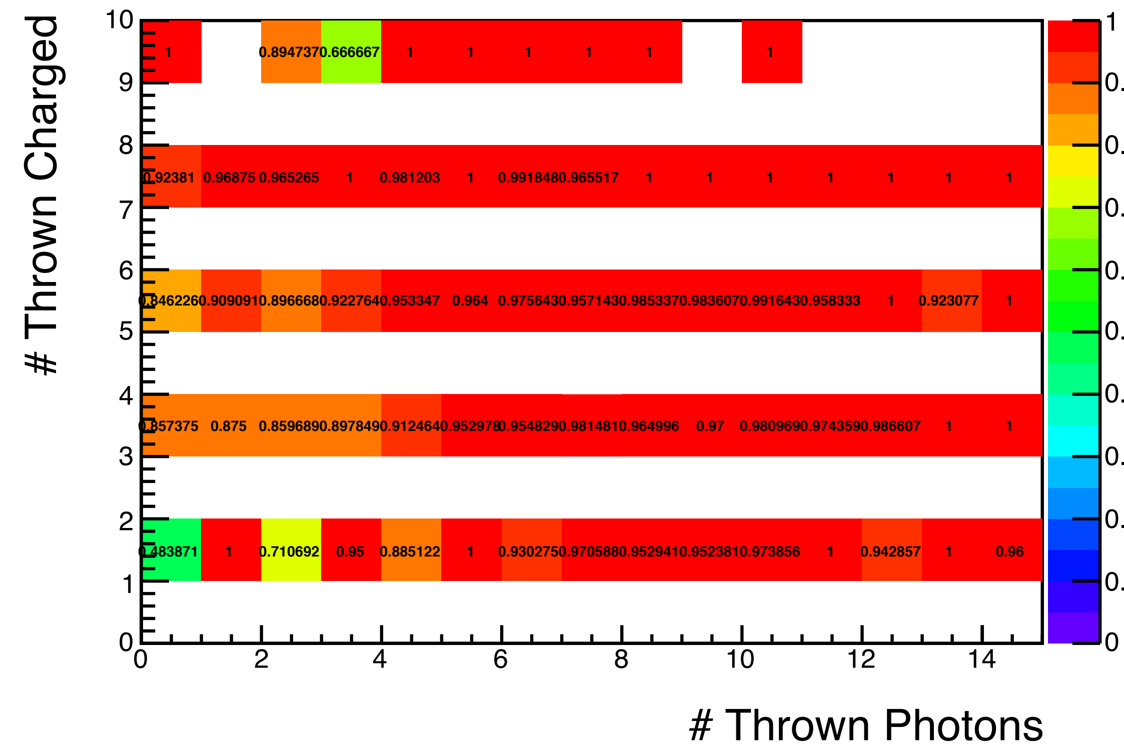
Trigger Meeting 1.21.14

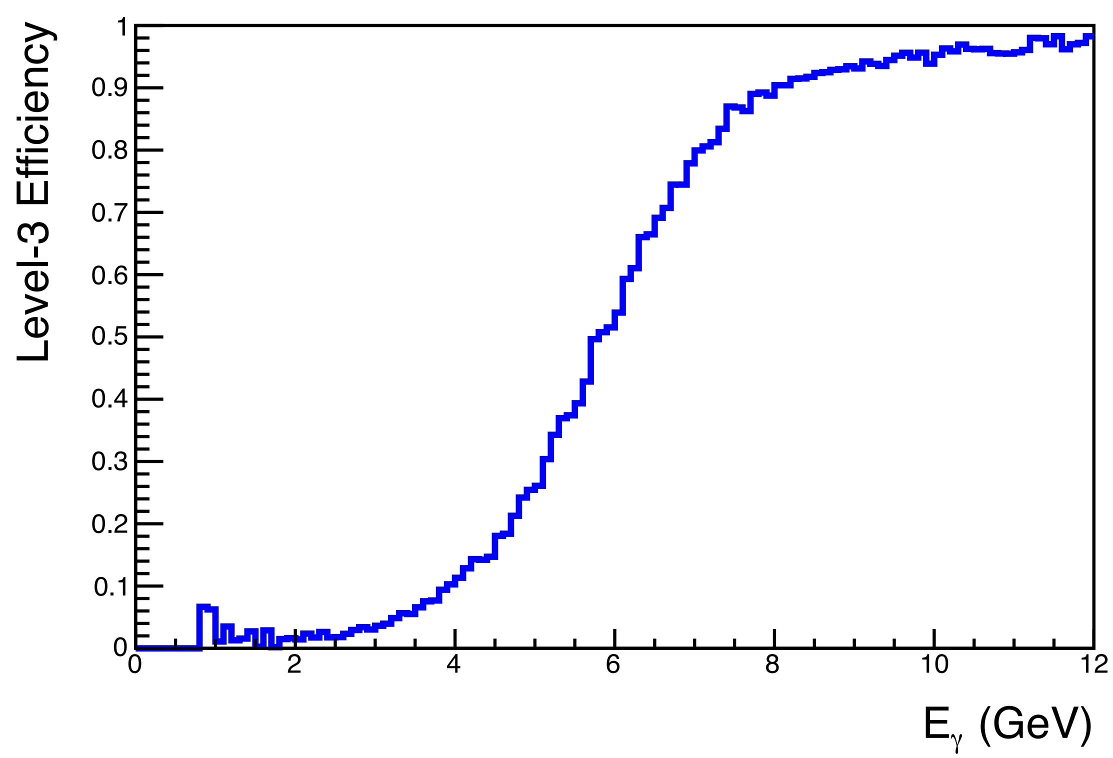
Reminder from last meeting

- Found that tracking momentum resolution was the limiting factor in Level-3 efficiency
- Goal was to study what may be possible with wire-based tracking to improve resolution
- Focus on a subset of tracks to reduce CPU cost

Level-3 Evaluation (**w/o** EM pileup)

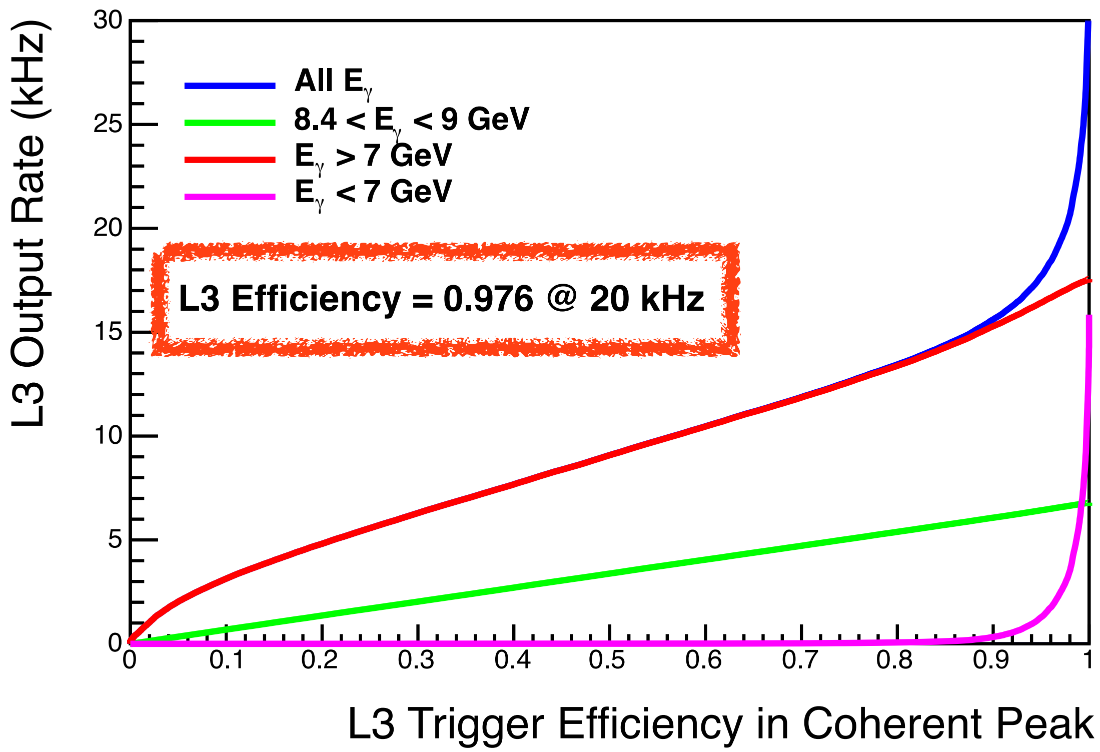


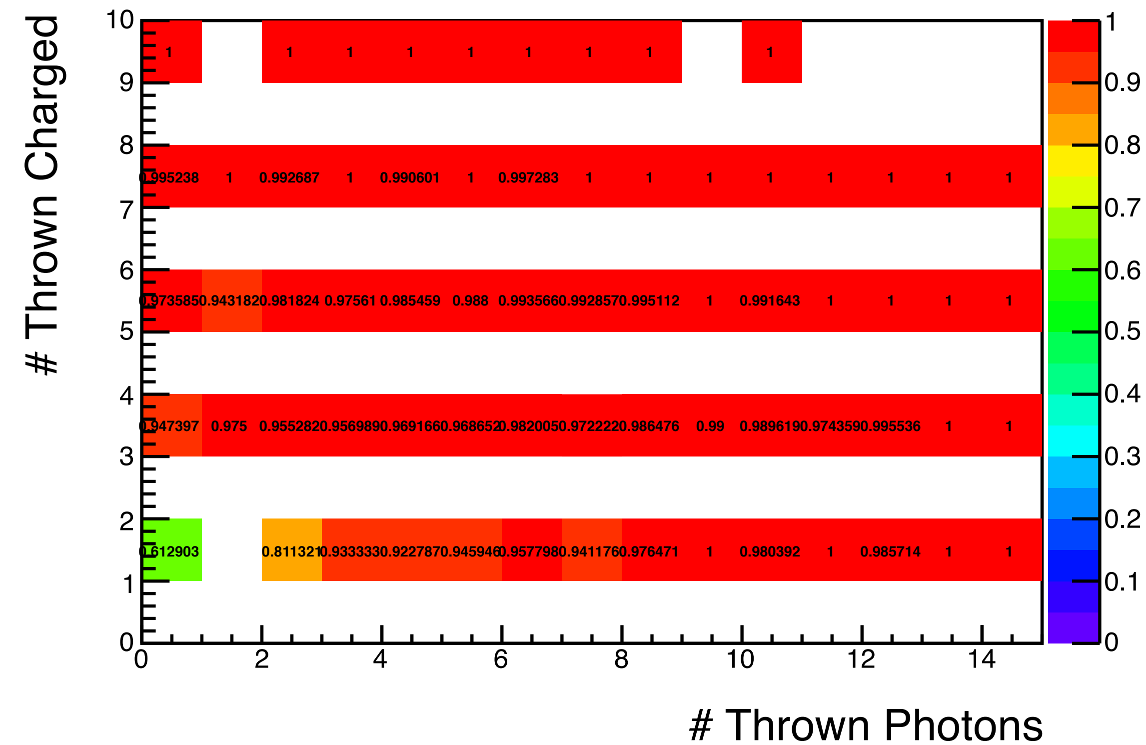


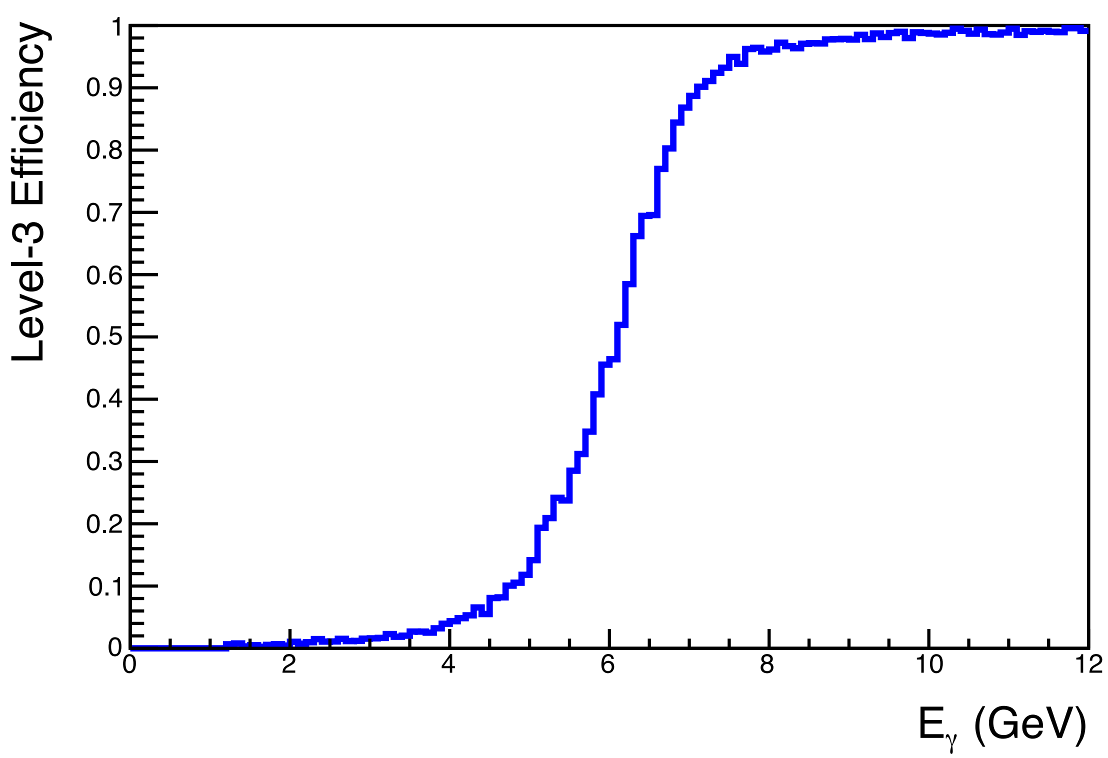


- bggen only sample (no pileup)
- Using DTrackCandidates (no wire-based tracking)
- For a rate of 20 kHz, achieve ~93% L3 average efficiency in the coherent peak
- Events with less photons have lower efficiency (~85% for zero photons)

Thrown π±/K±/p(p̄)/γ momentum sum

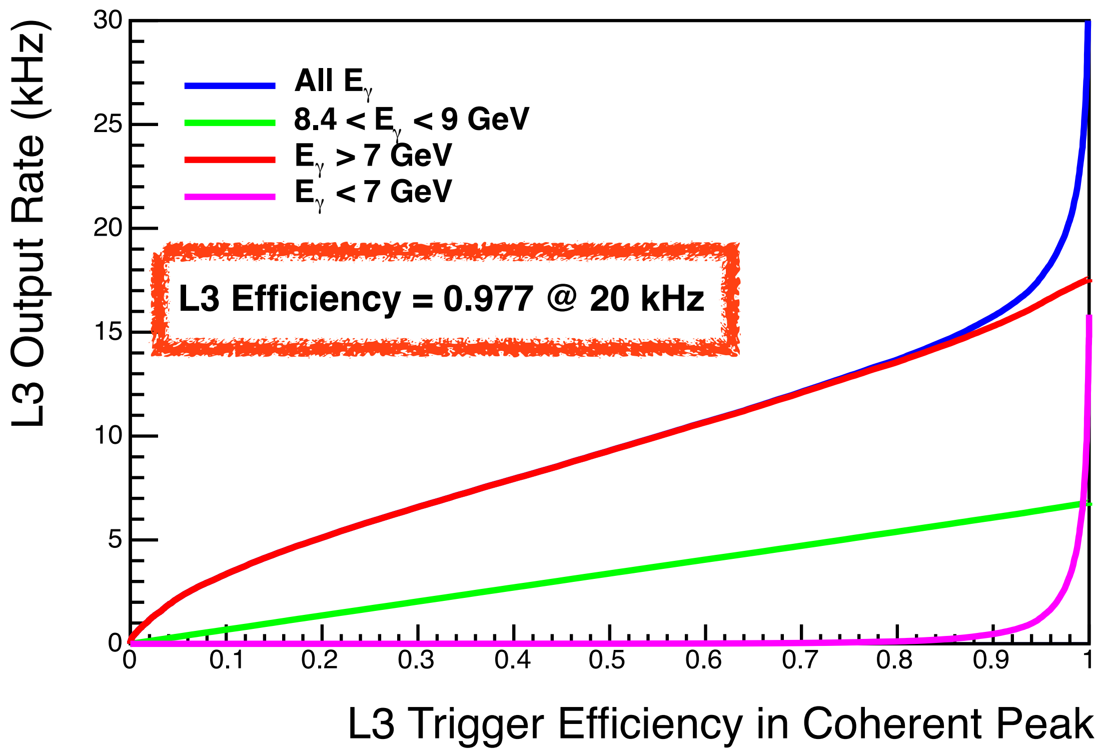


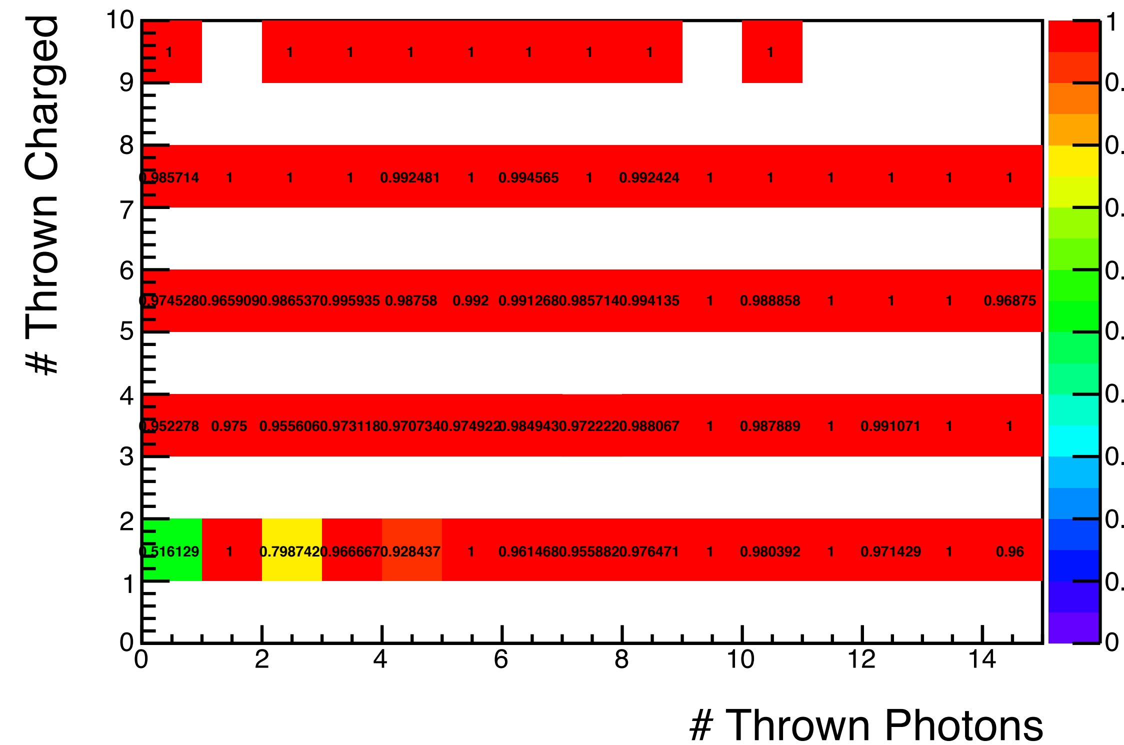


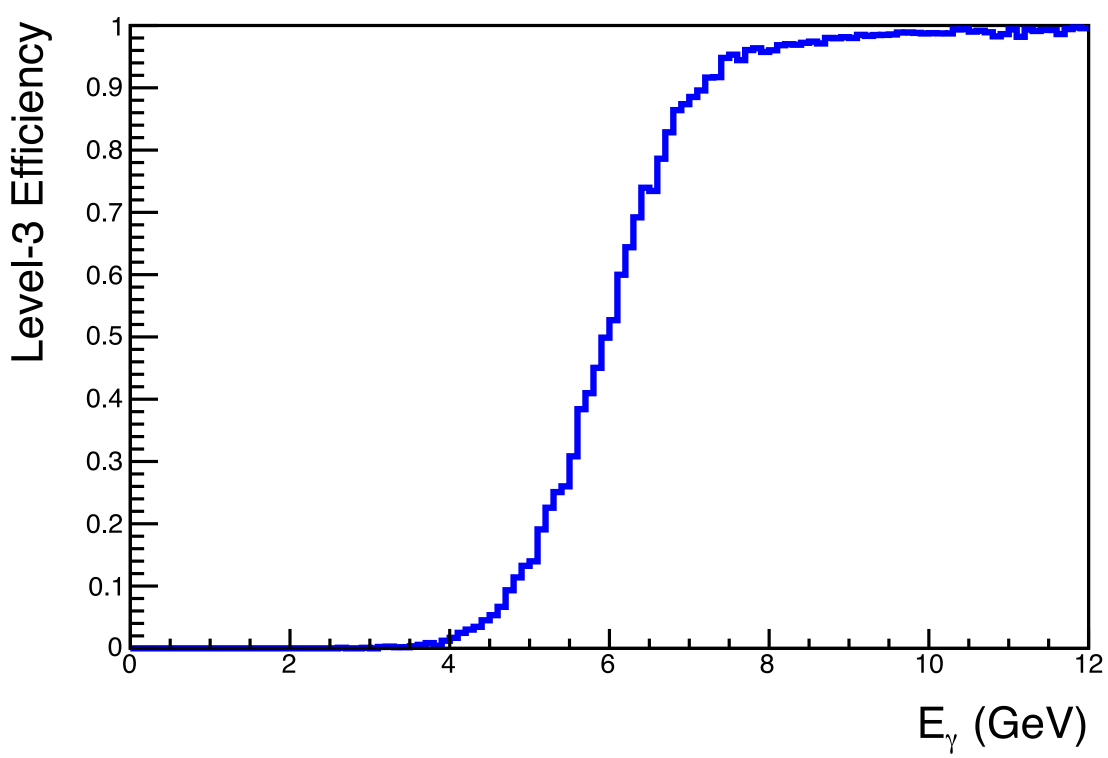


- Use only thrown particle information for π±/K±/p(p) and photons
- Very high efficiency as expected, with deviation from 100% coming from lack of neutron information
- Try using "pieces" of thrown information to see where the current weaknesses are

Sum of thrown charged particle momentum (instead of reconstructed track momentum sum)



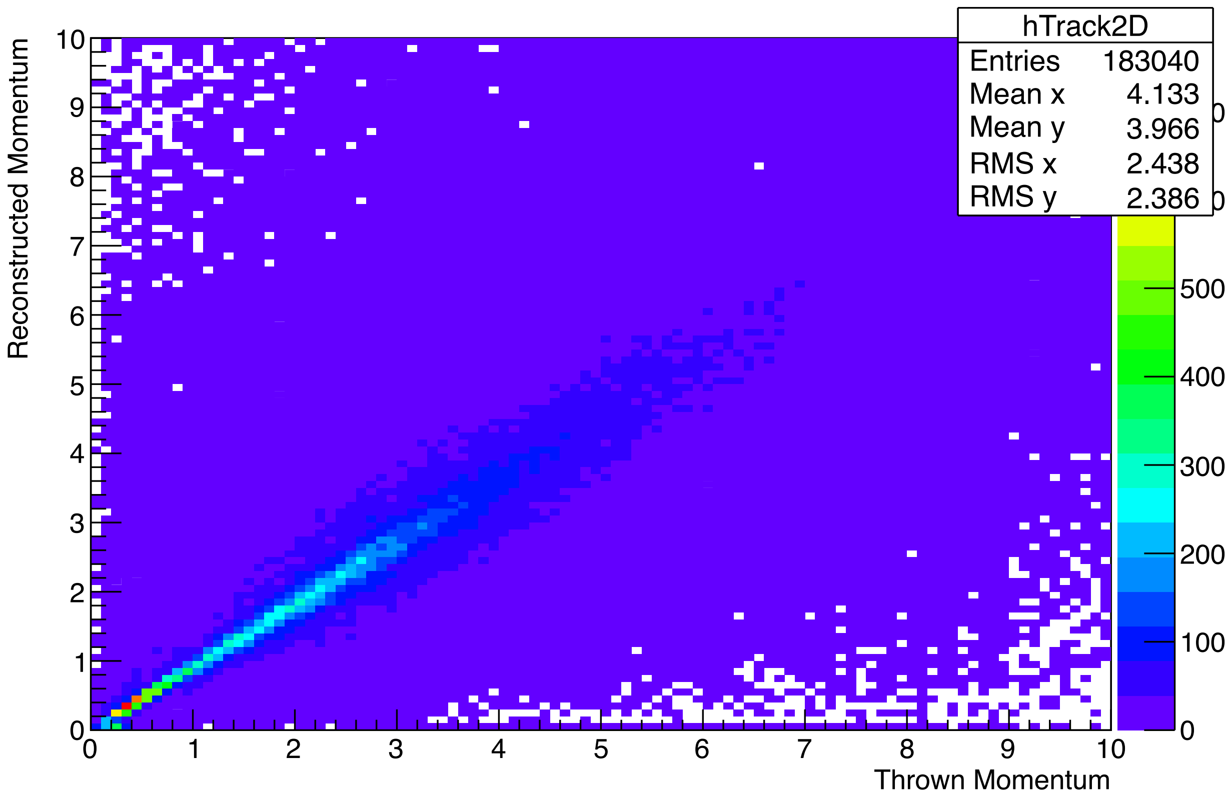




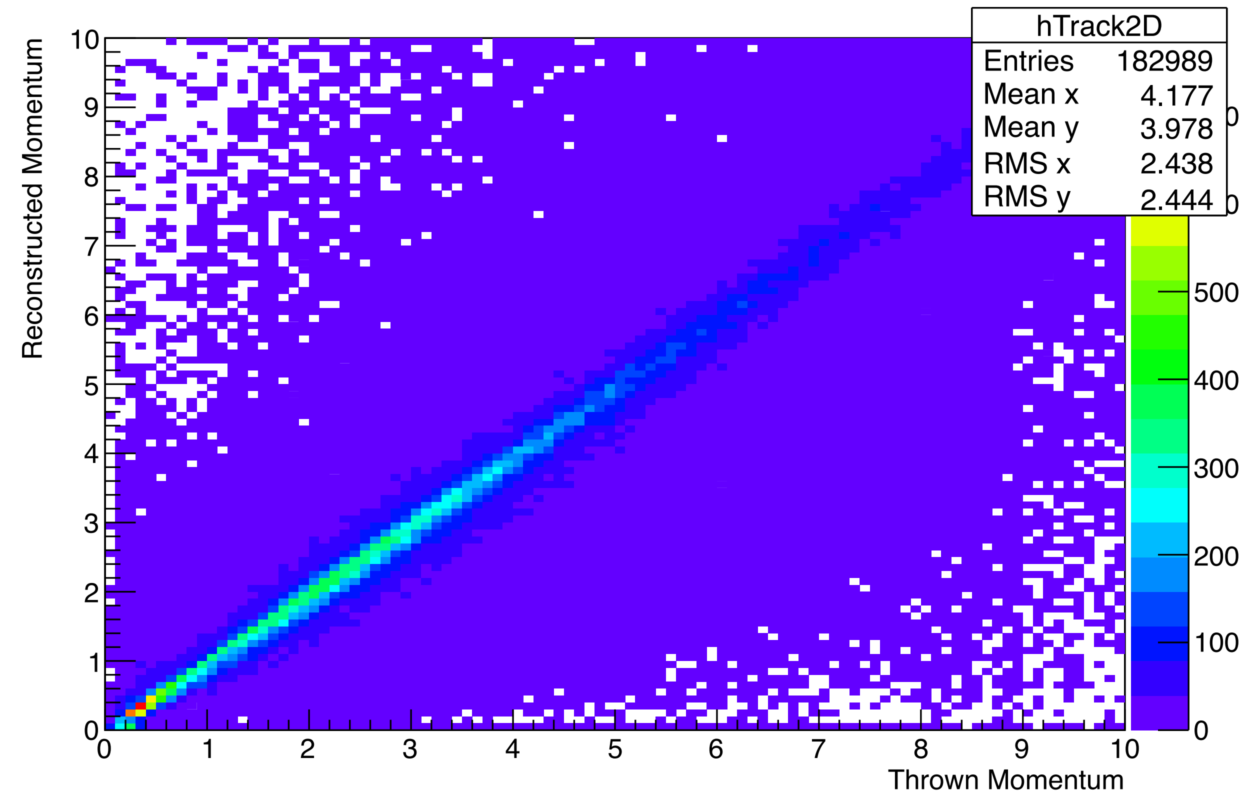
- Use **thrown** charged particle momentum sum, but **reconstructed** FCAL+BCAL energy
- For a rate of 20 kHz, achieve ~98% L3 average efficiency in the coherent peak
- Much improved performance, especially for zero photon events!
- Conclusion: track momentum sum resolution **is** the limiting factor in the current algo

Tracking: momentum resolution

DTrackCandidate

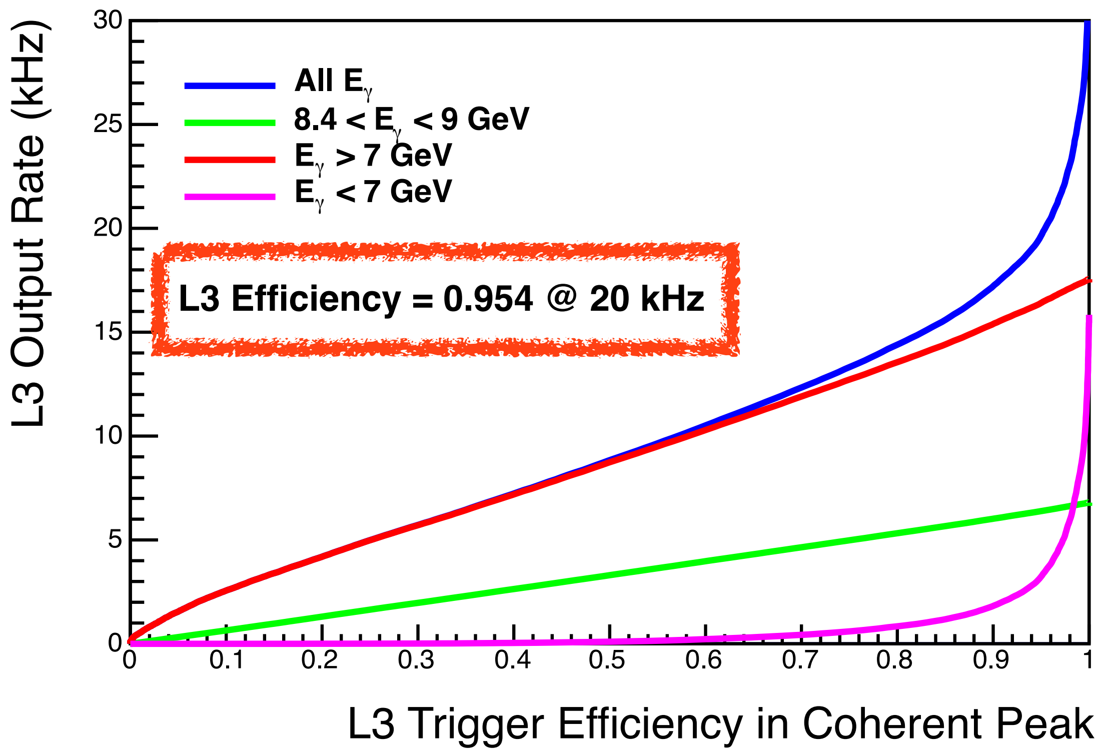


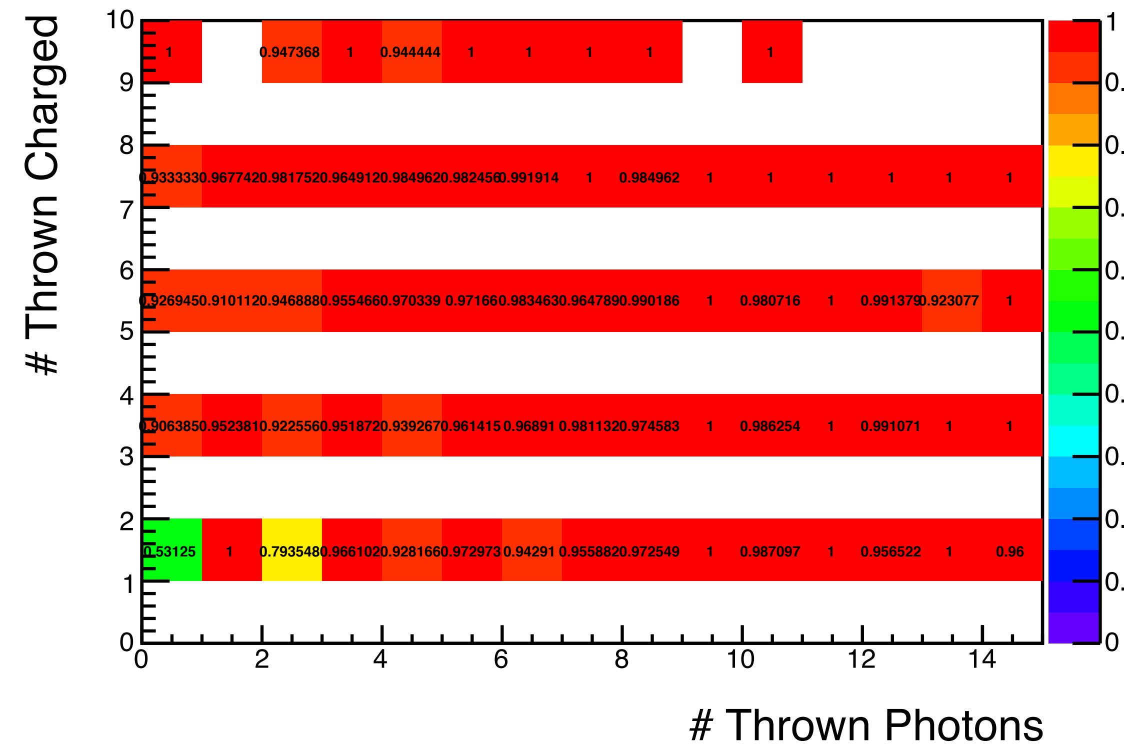
DTrackWireBased

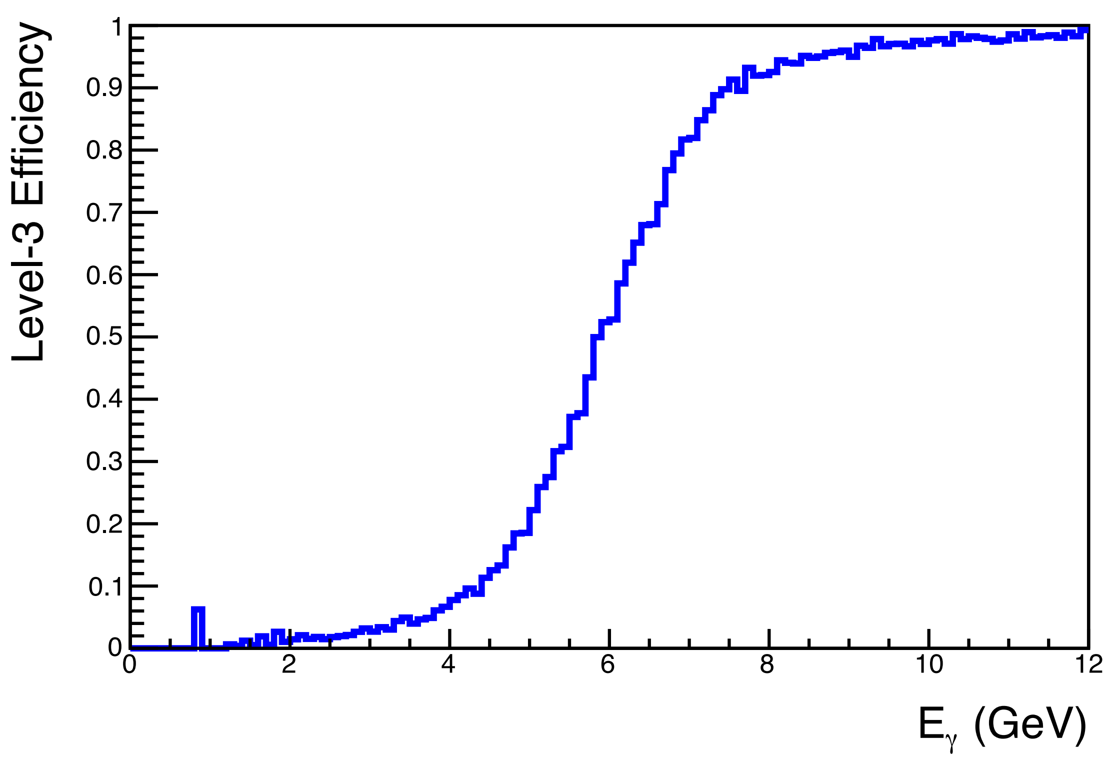


- Momentum resolution appears to be the limiting factor when the BDT is trained with DTrackCandidate
- How much does DTrackWireBased help?

Wire-based tracking (**w/o** EM pileup)

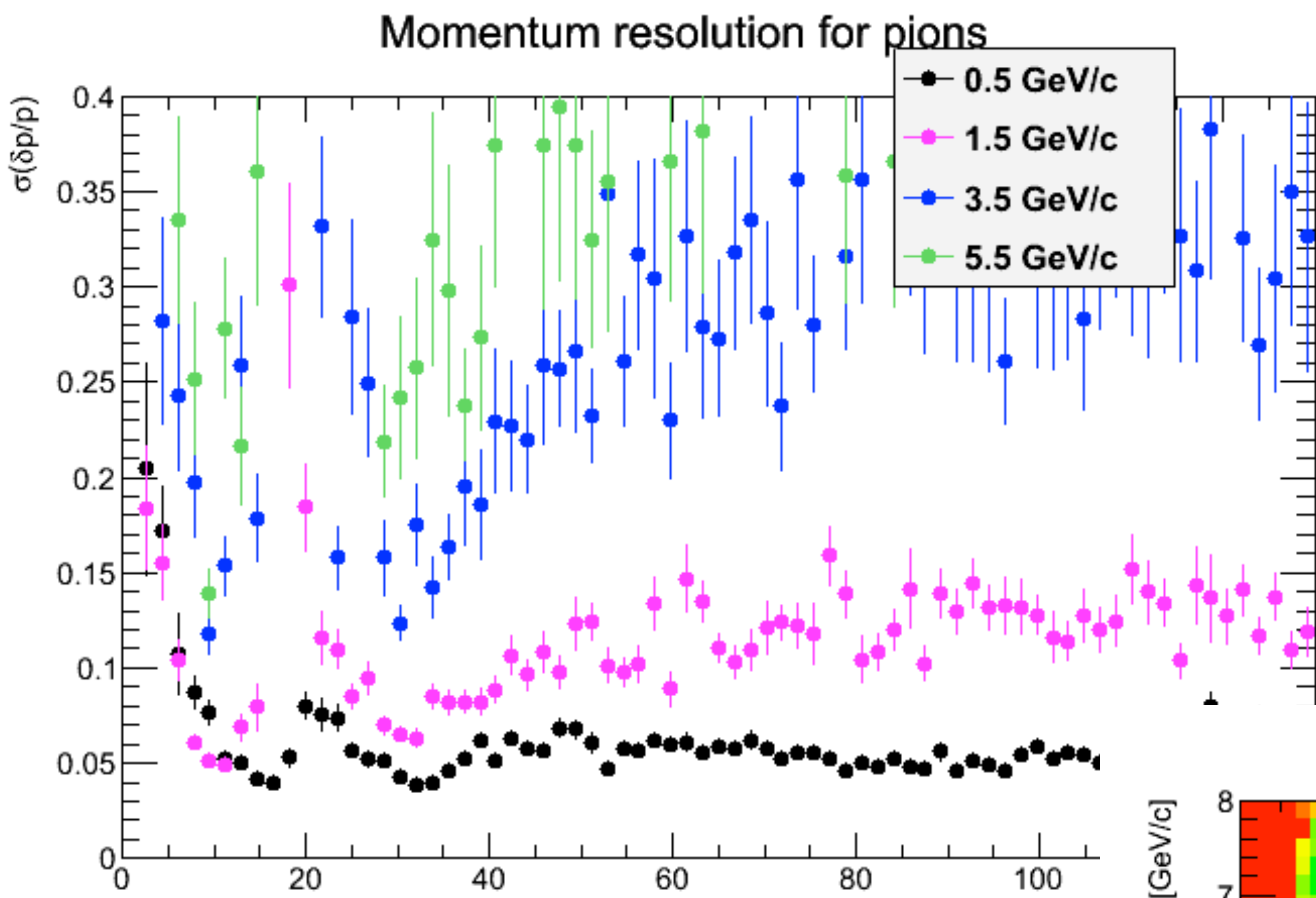




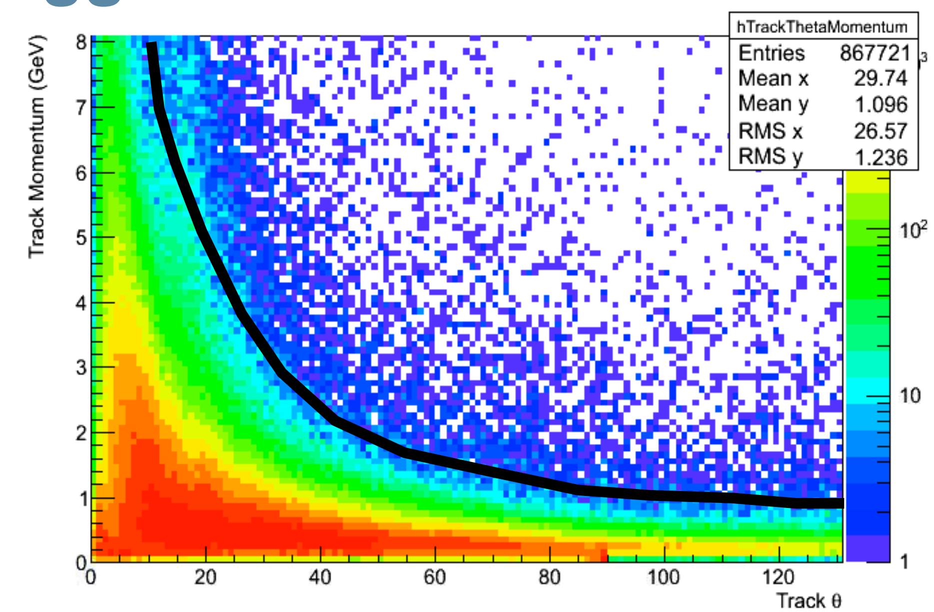


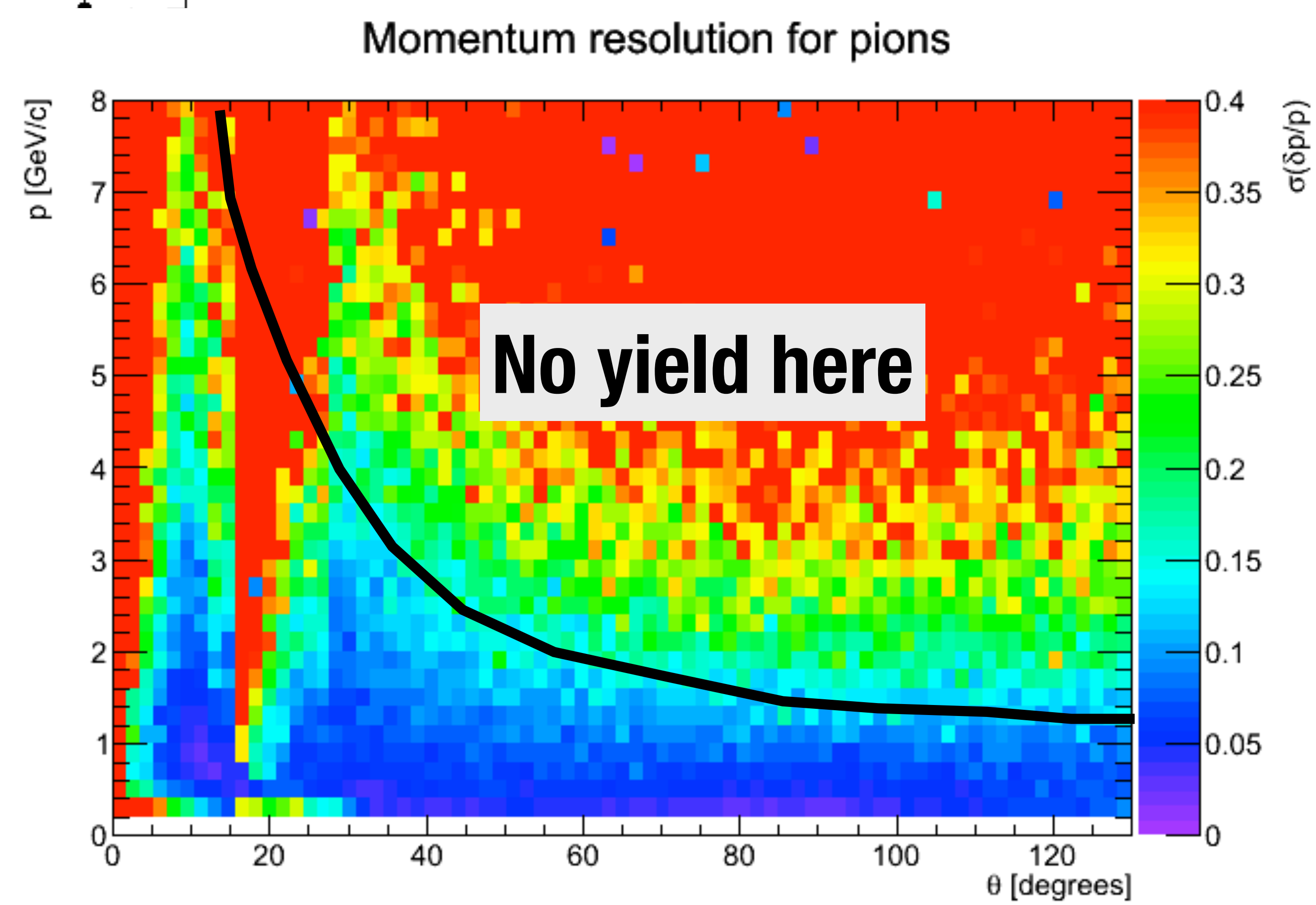
- Now use only reconstructed variables, but use **wire-based tracking**
- For a rate of 20 kHz, achieve ~95% L3 average efficiency in the coherent peak
- For # neutrons = 0, have ~97% effic
- Performance in between thrown track momentum sum and using sum of DTrackCandidate's momentum

Single Track Resolution: Candidates



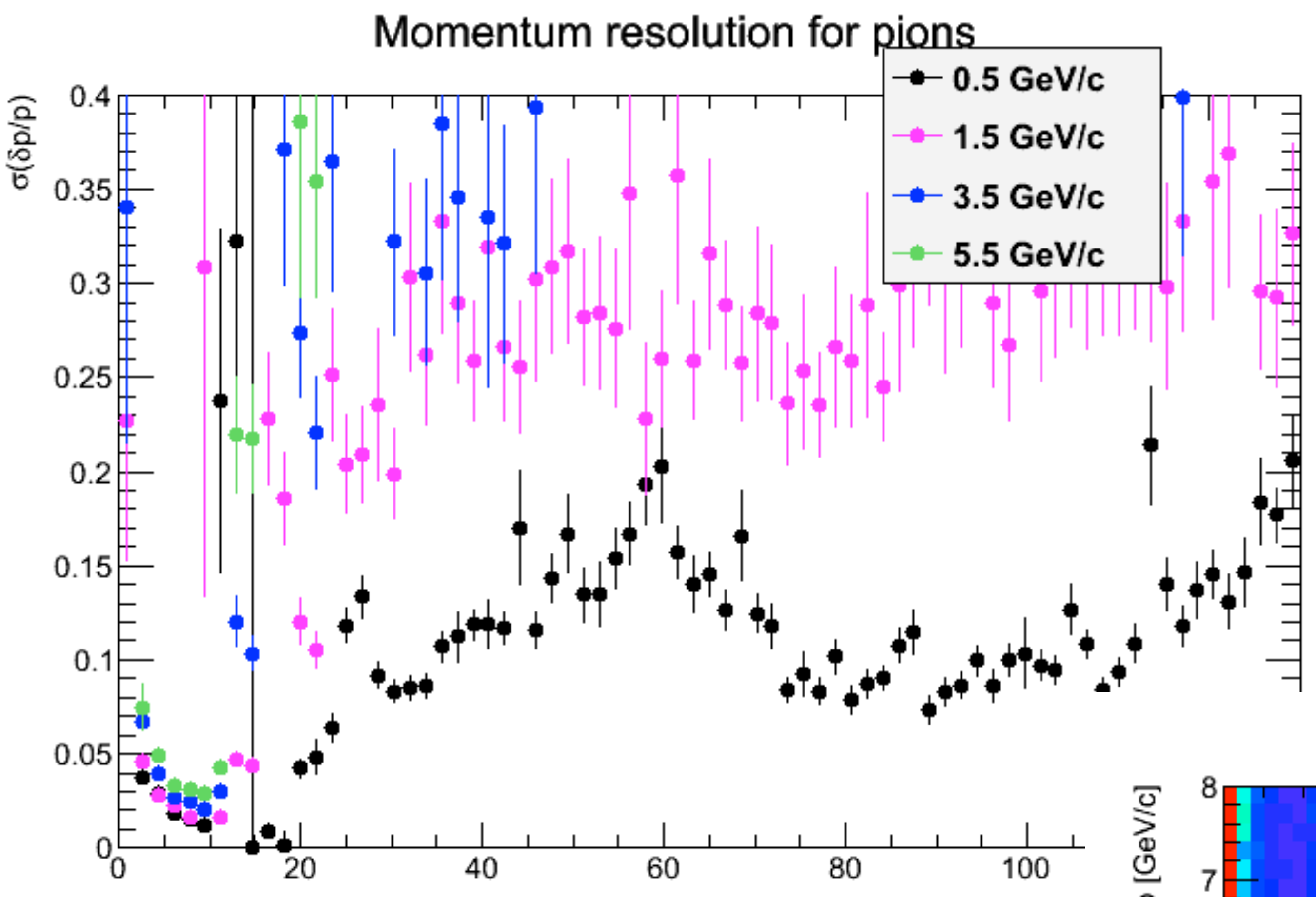
bggen reconstructed tracks

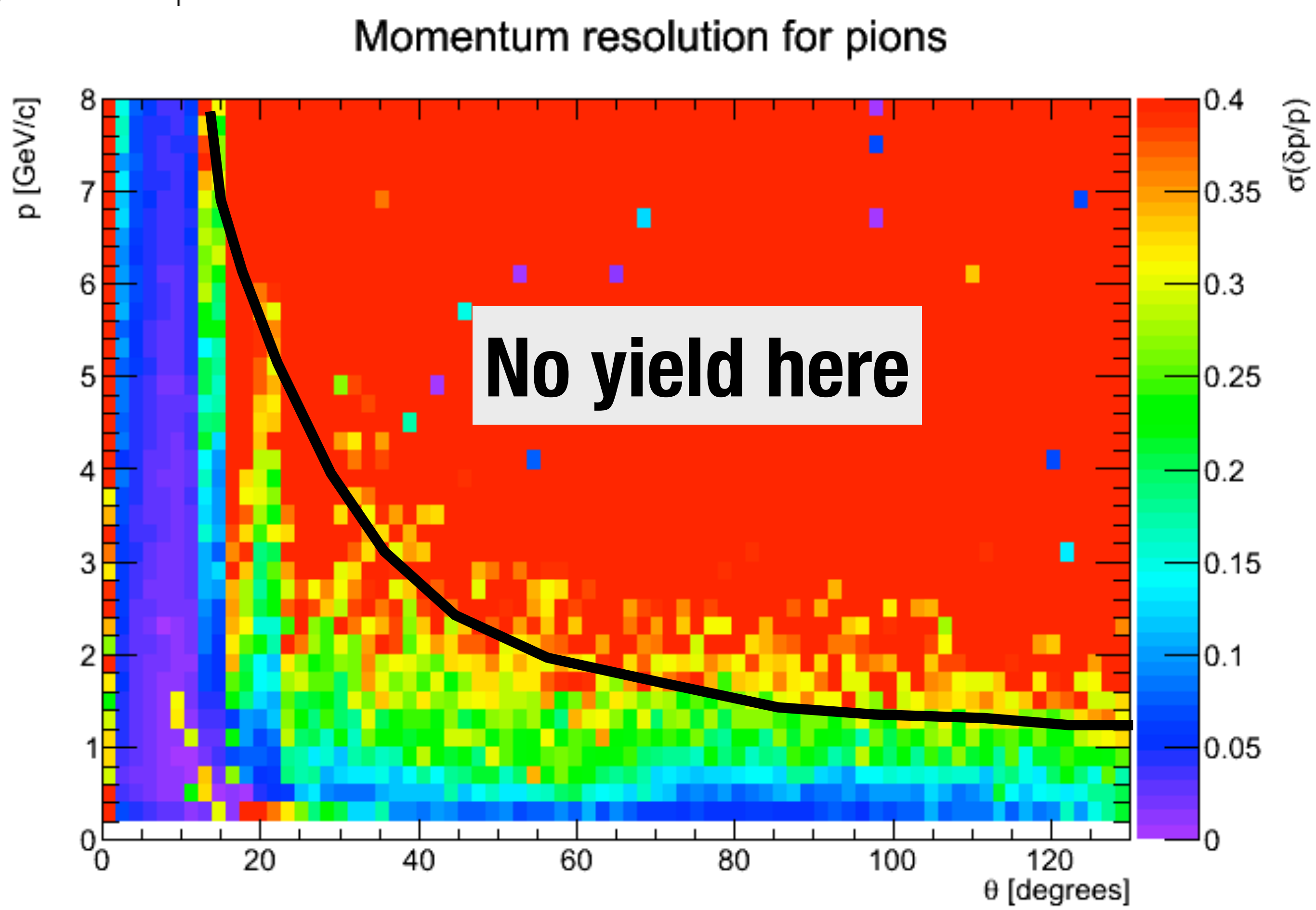




- Momentum resolution poorest at forward angles and boundary between CDC/FDC
- Try wire-based to see where most improvement is

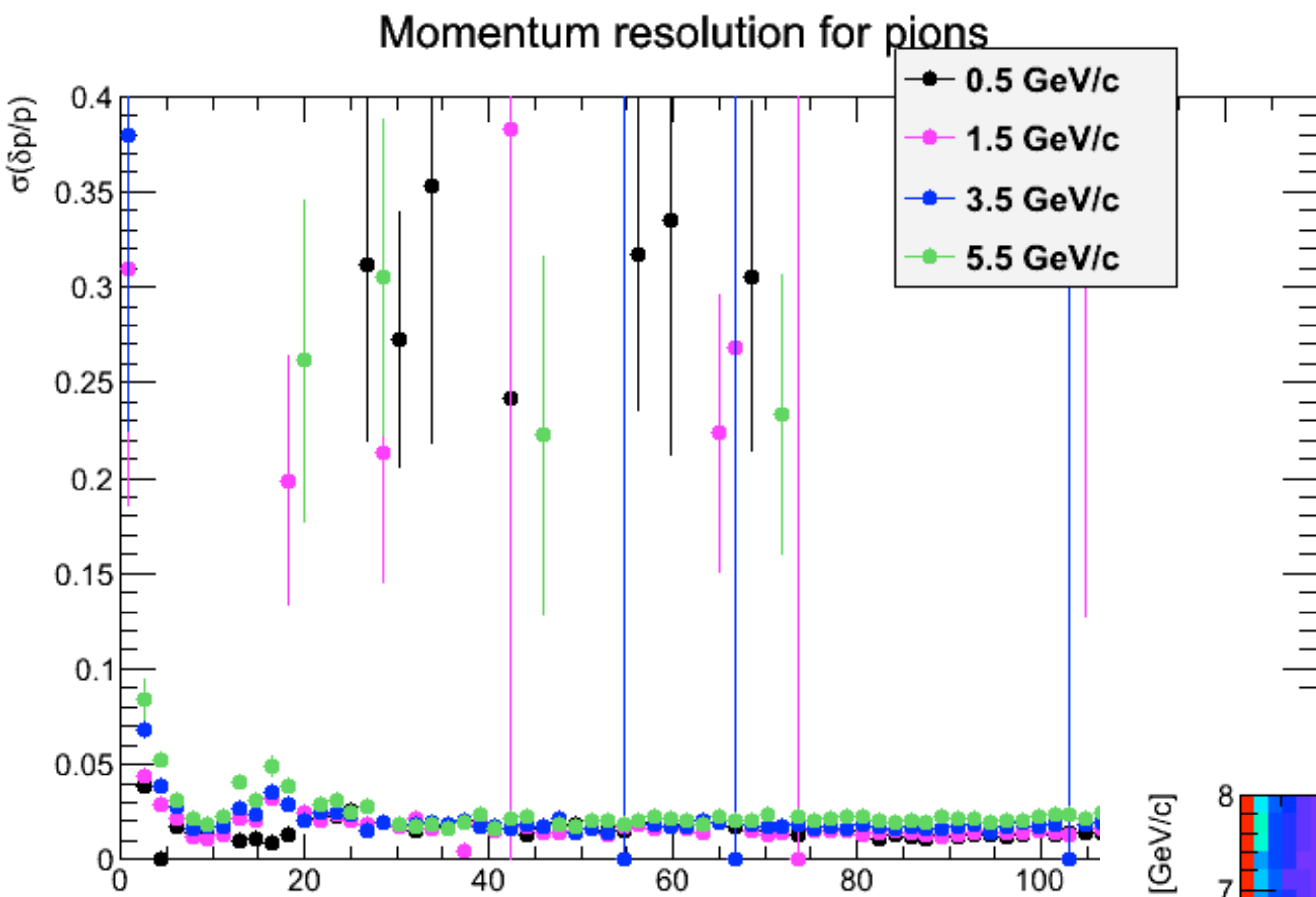
Single Track Resolution: WireBased



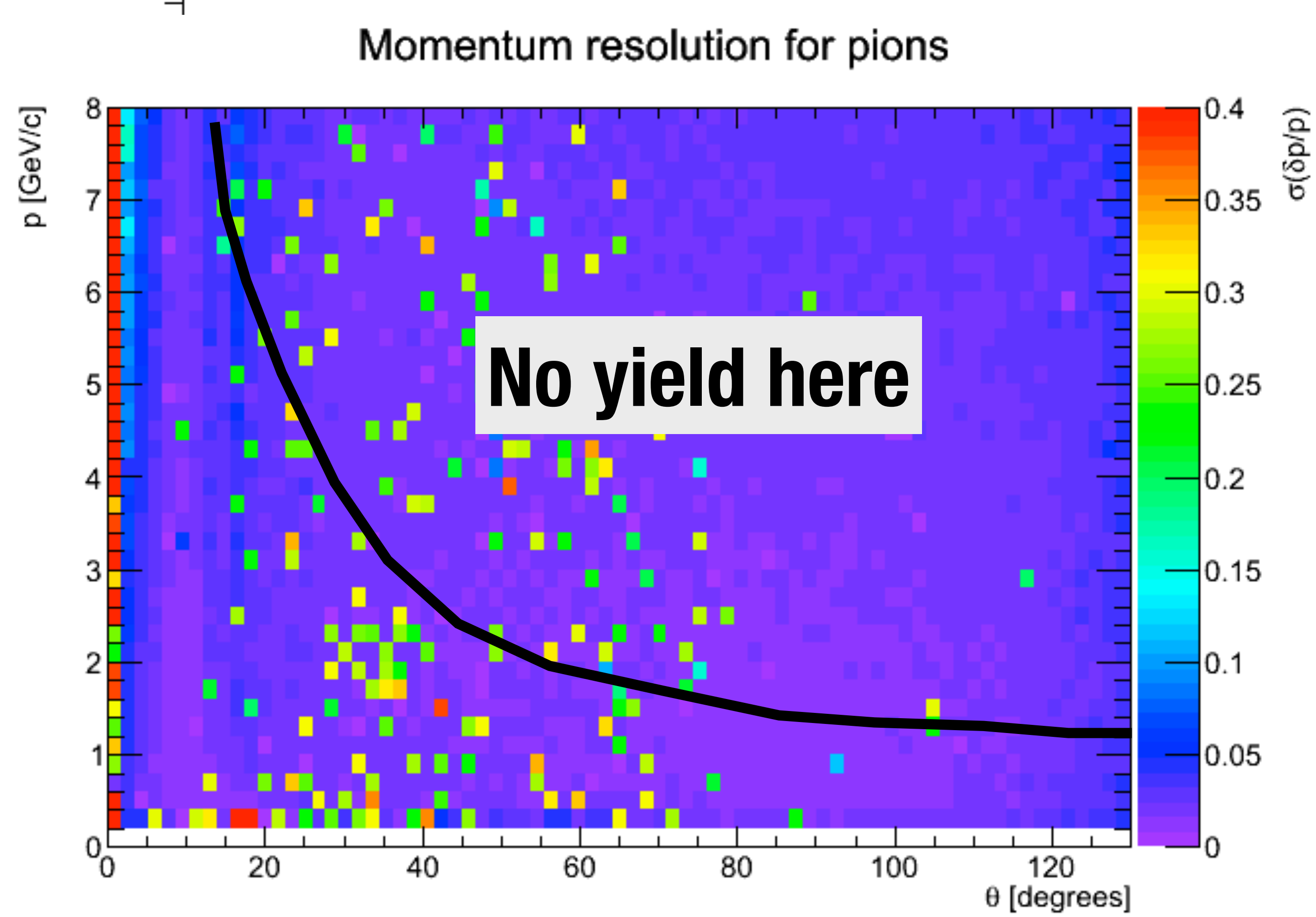


- Forward angle momentum resolution very good
- Resolution is worse for tracks in the CDC than for candidates?

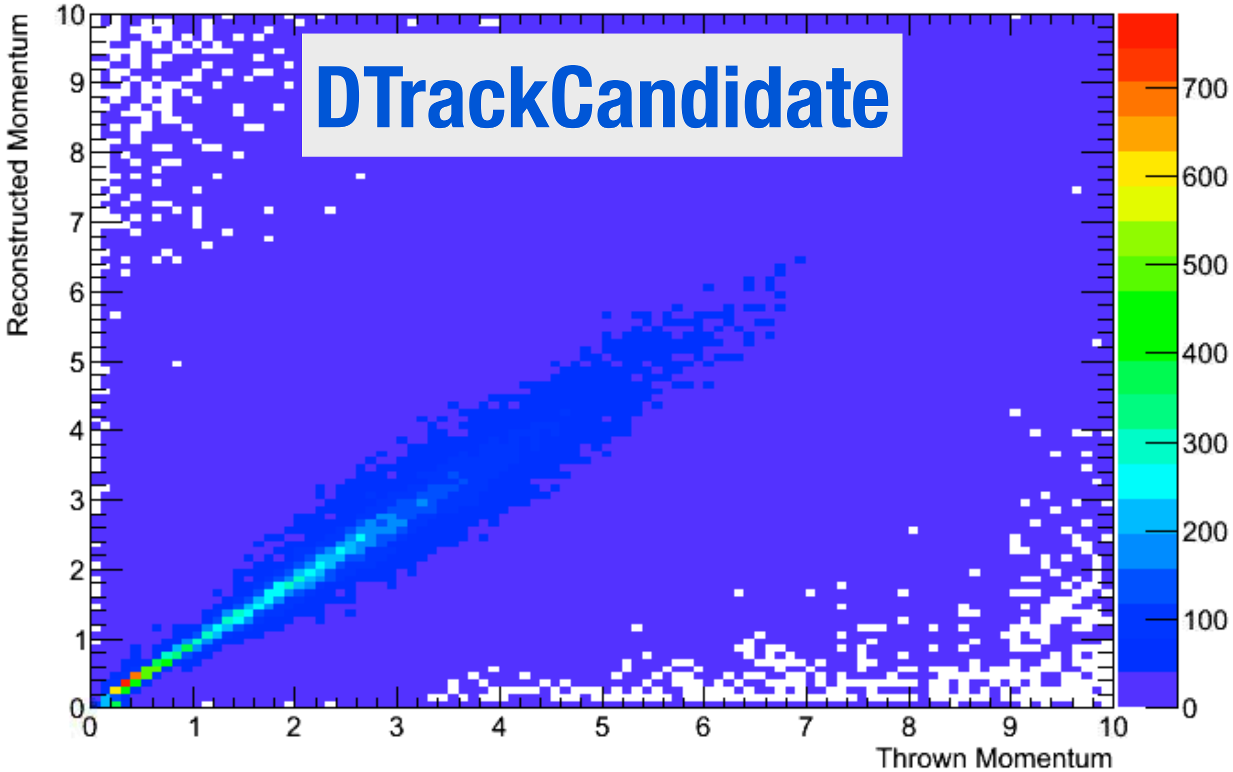
Single Track Resolution: TimeBased

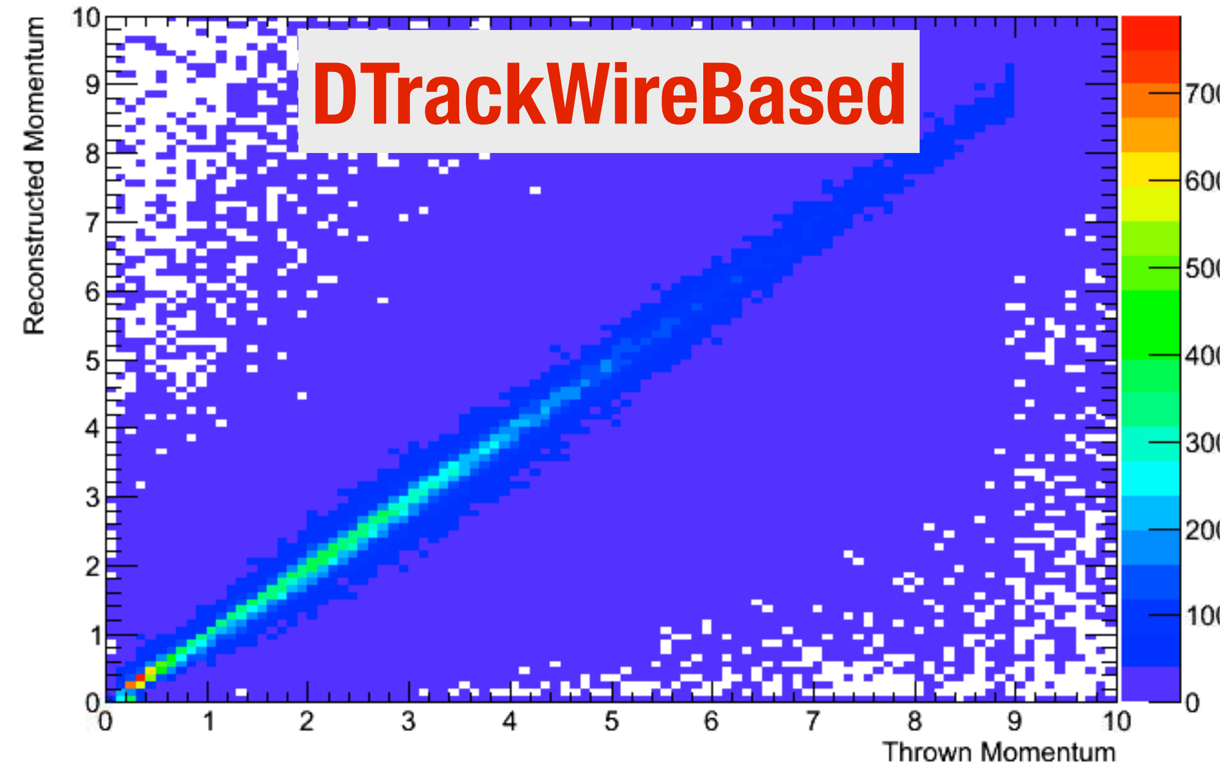


- Including drift times in the fit for time-based tracks looks good everywhere as expected



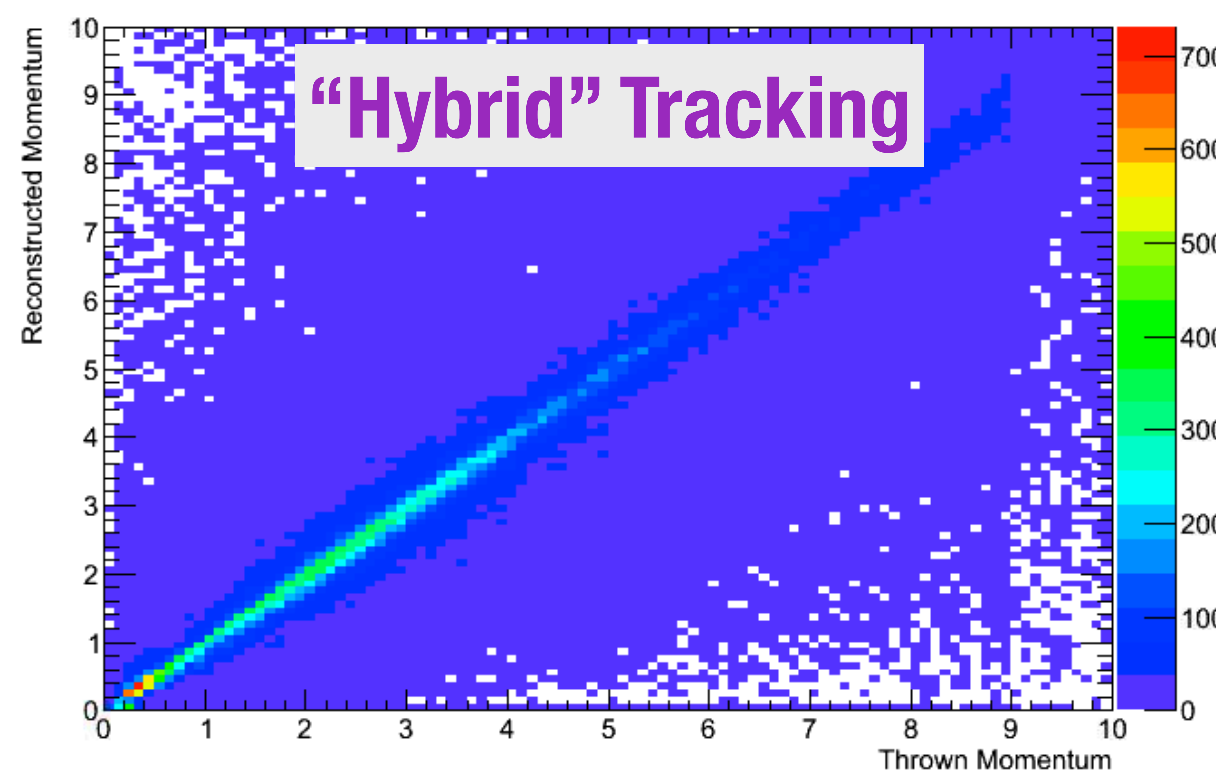
Track momentum sum resolution



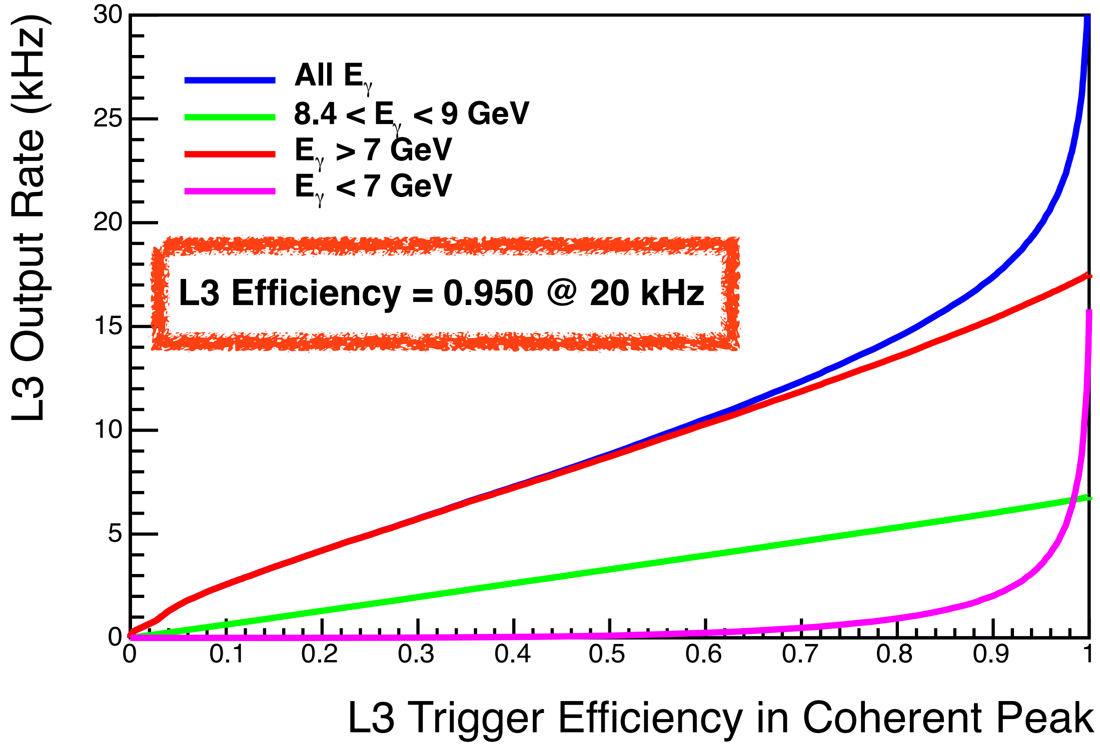


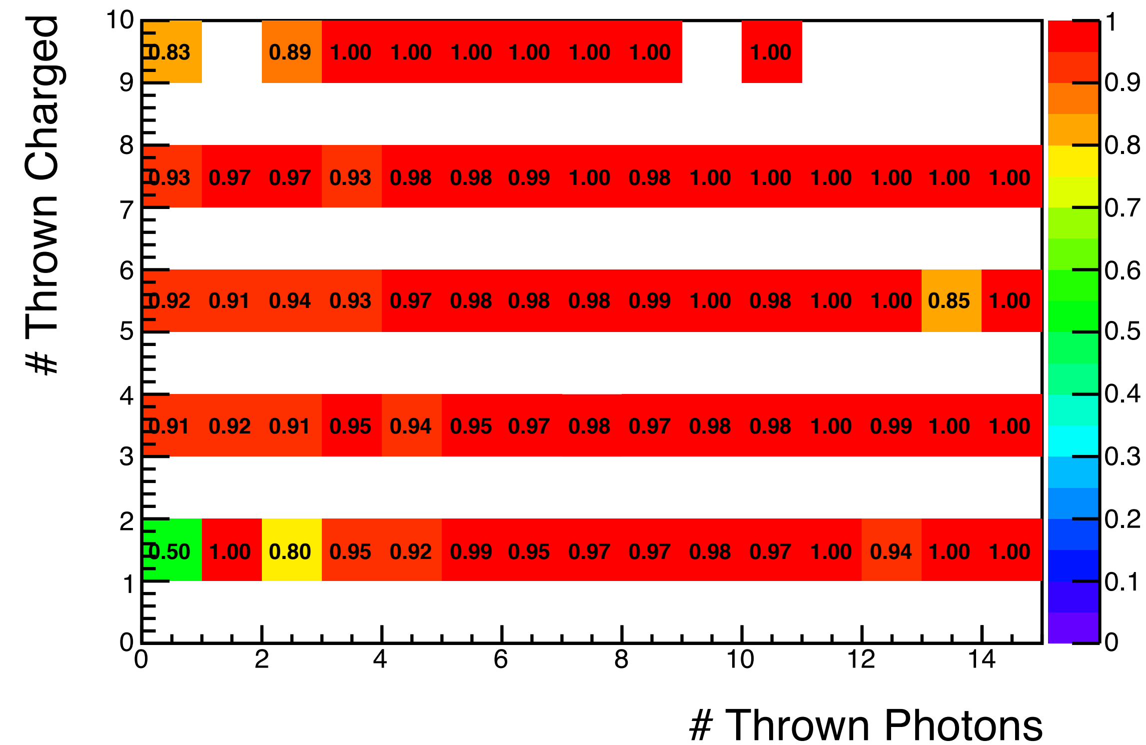
- Resolution only improved for wire-based with θ < 15

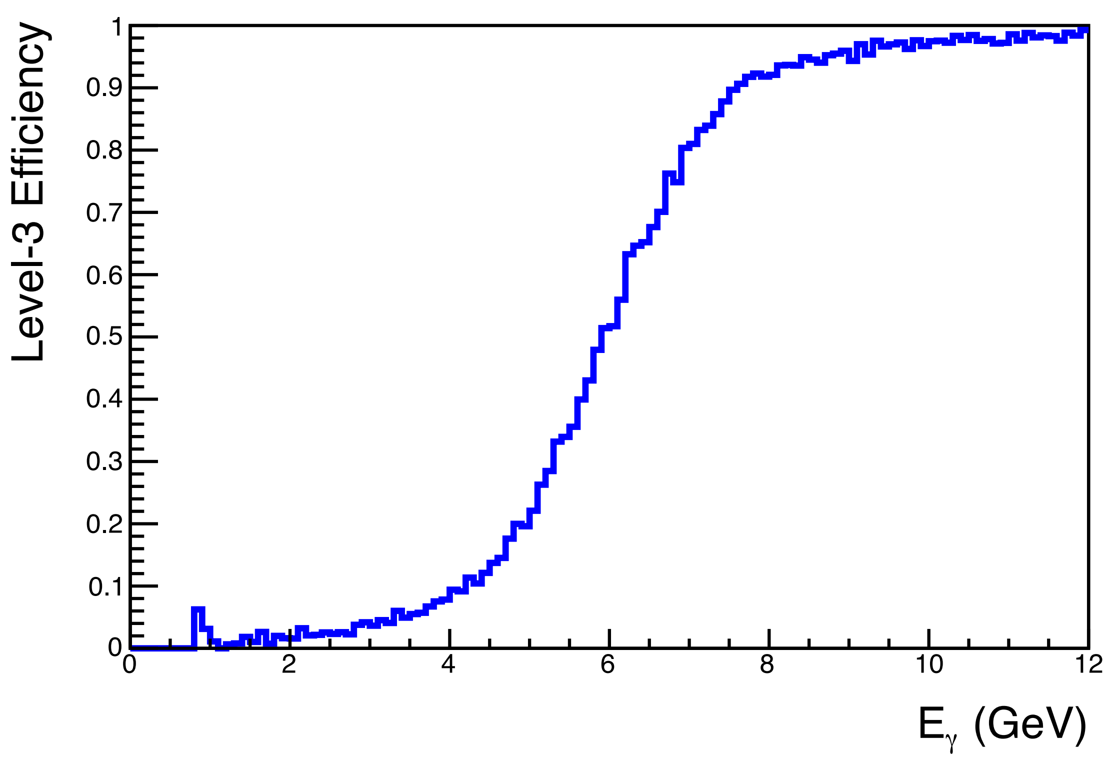
- **Take "hybrid" approach**: for DTrackCandidate's with θ < 15 (40% of candidates) and do wire-based fit and use that in track momentum sum



Hybrid tracking (**w/o** EM pileup)

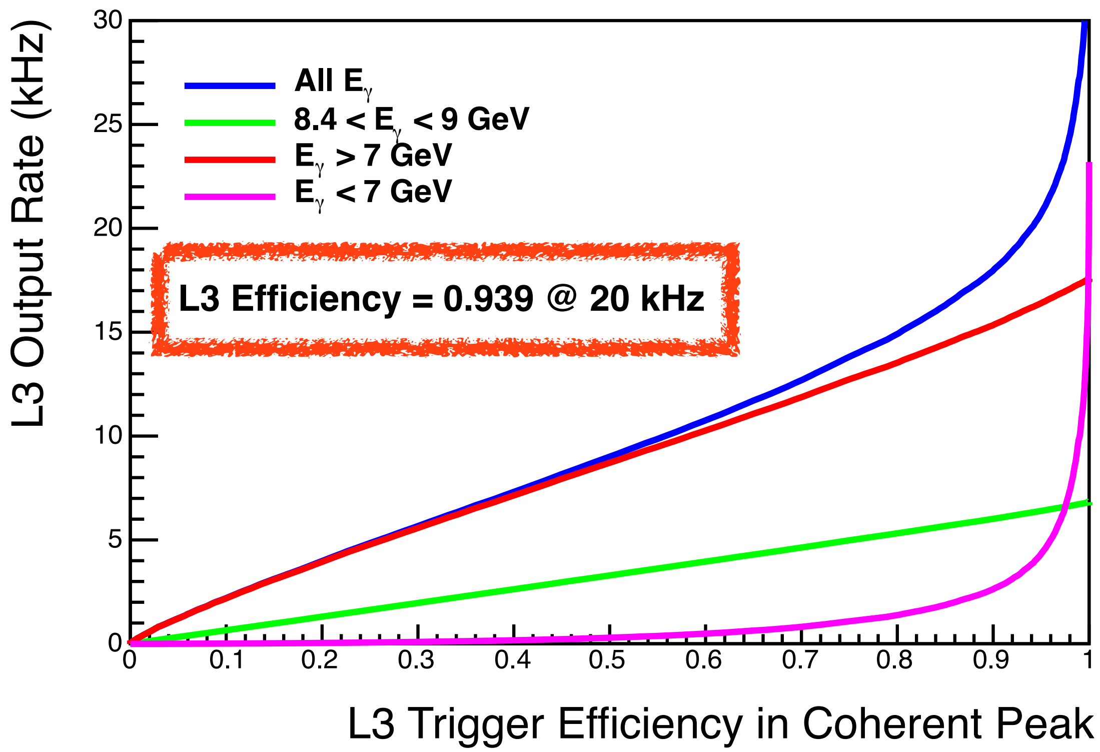


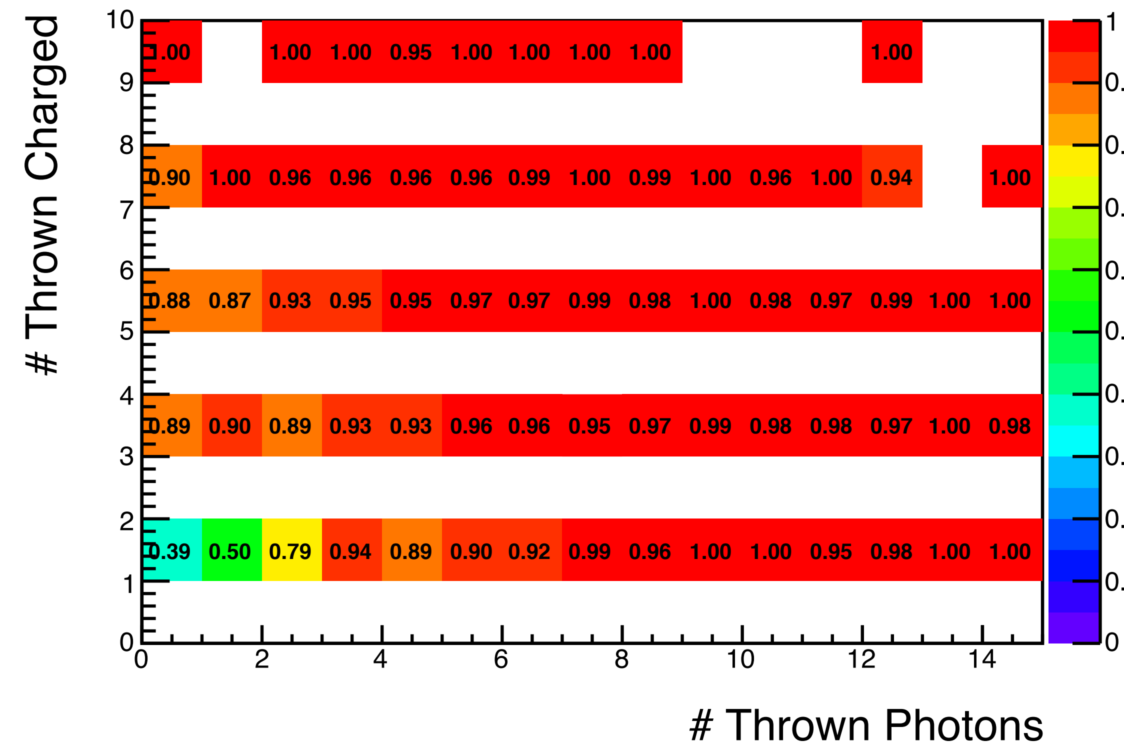


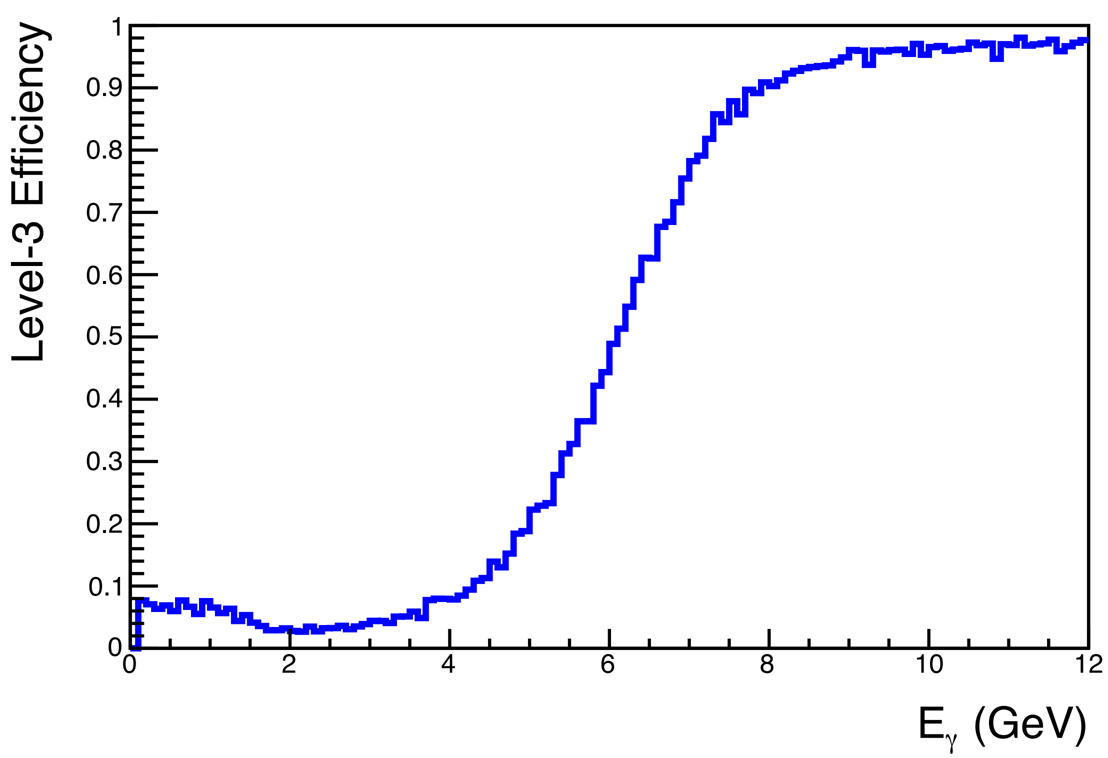


- Now use only reconstructed variables, with **hybrid tracking**
- For a rate of 20 kHz, achieve ~95% L3 average efficiency in the coherent peak
- For # neutrons = 0, have ~96% effic
- 92% efficiency for zero photon events
- Performance is similar to doing full wire-based tracking (with less CPU)

Hybrid tracking (**w/** EM pileup)







- Now use only reconstructed variables, with **hybrid tracking**
- For a rate of 20 kHz, achieve ~94% L3 average efficiency in the coherent peak
- For # neutrons = 0, have ~95% effic
- 89% efficiency for zero photon events
- Performance is similar to doing full wire-based tracking (with less CPU)

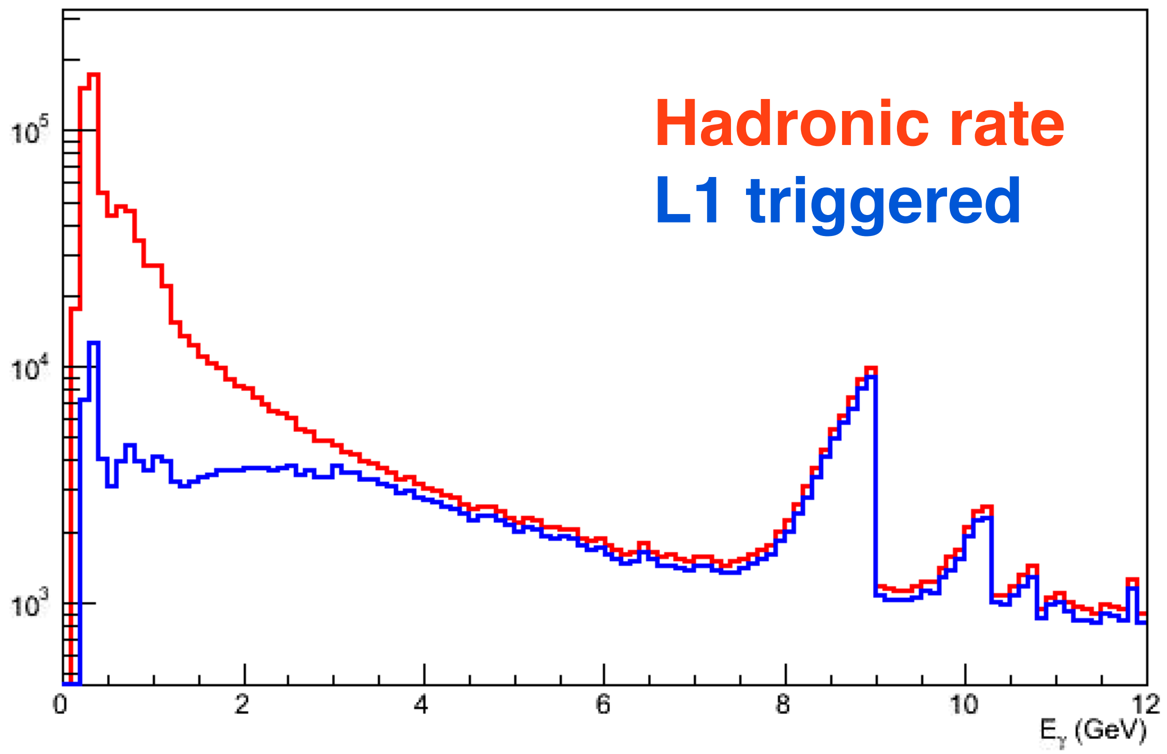
To Do List

- Test algorithm on online nodes
  - Better estimate of CPU requirements for different tracking options, and staging of BDT selection
  - May run into L3 crash from last ODC...
- Study more samples with current algorithm:
  - EM only background events
  - Some reactions of interest (eg. n3pi, b1pi, ...)

Backup

Level-1 Trigger

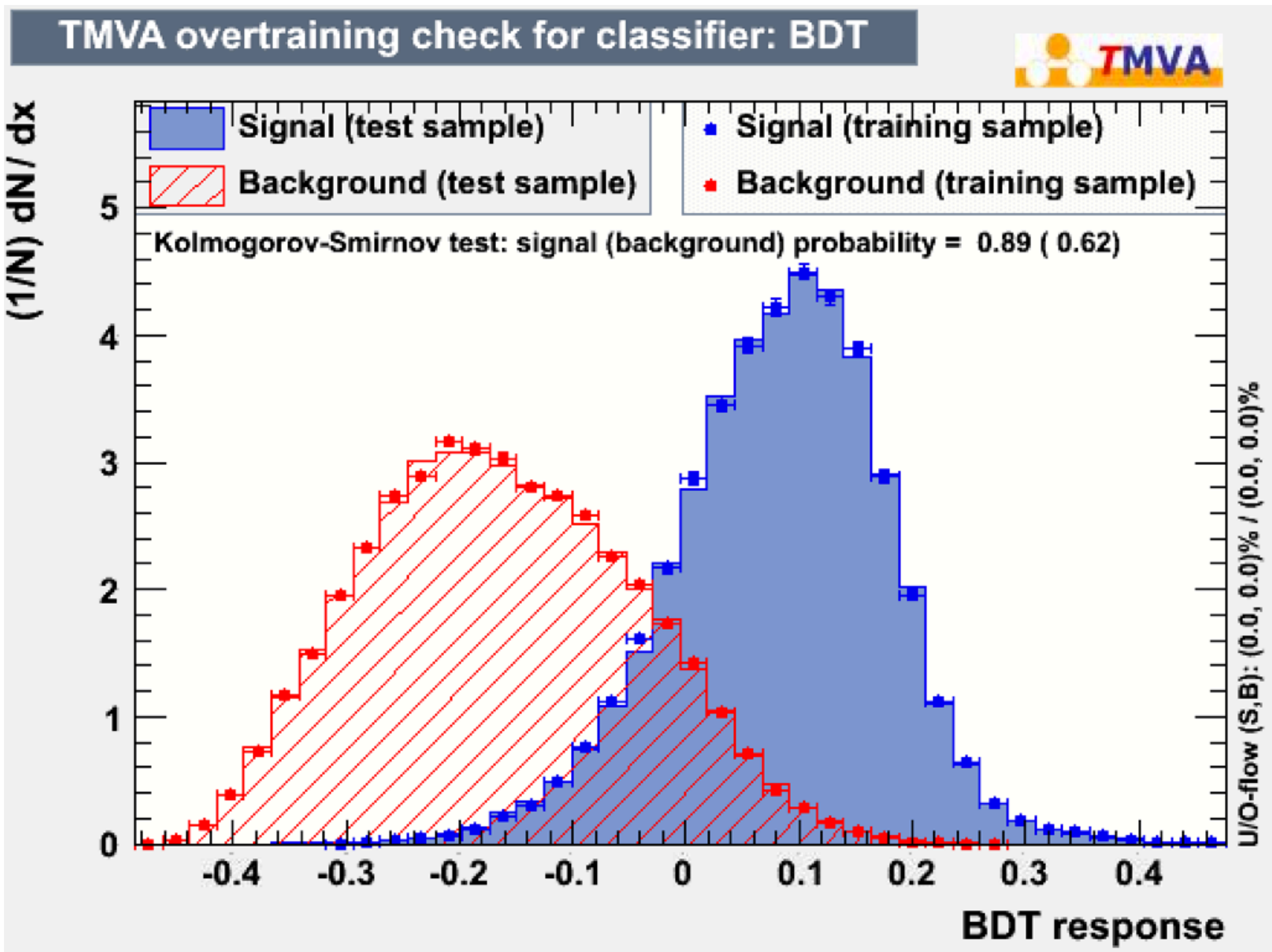
GlueX-doc-1043: Implemented in DMCTrigger



- Sample of bggen events with high-luminosity EM pileup
- Define "signal" as $E_\gamma > 7$ GeV and "background" $E_\gamma < 7$ GeV
- Accept events which fire L1a or L1b emulated trigger
- Reject ~77% of background with signal efficiency of 92%
- So far haven't considered EM only background rate

```
bool sum_cut = (Ebcal + 4.0*Efcal)>=2.0;
trig->L1a_fired = sum_cut && Ebcal>0.200 && Efcal>0.030;
trig->L1b_fired = sum_cut && Ebcal>0.030 && Efcal>0.030 && Nschits>0;
```

Level-3 Training




```
--- Factory    : Ranking input variables (method specific)...
--- BDT        : Ranking result (top variable is best ranked)
--- BDT        : ---------------------------------------------------
--- BDT        : Rank : Variable              : Variable Importance
--- BDT        : ---------------------------------------------------
--- BDT        :    1 : EfcalClusters         : 1.636e-01
--- BDT        :    2 : Ptot_tracks_cut       : 1.366e-01
--- BDT        :    3 : EbcalPoints           : 1.245e-01
--- BDT        :    4 : EbcalClusters         : 1.110e-01
--- BDT        :    5 : Ntrack_candidates_cut : 9.658e-02
--- BDT        :    6 : Ntof                  : 8.451e-02
--- BDT        :    7 : Nfcal_clusters        : 7.634e-02
--- BDT        :    8 : Nstart_counter        : 7.554e-02
--- BDT        :    9 : Nbcal_points          : 7.453e-02
--- BDT        :   10 : Nbcal_clusters        : 5.663e-02
--- BDT        : ---------------------------------------------------
```

- Only use tracking variables expected to be “stable” and able to simulate offline: Sum of track momentum and # of tracks
- This version of the algo was attempted to be used in the online data challenge (more in David’s talk)

Multiple Stages of Level-3

- Reconstruction of some input variables are more "expensive" than others

	SC	TOF	FCAL	BCAL	Tracking
Reco time (ms)	0.02	0.25	0.19	0.30	13.5

- Train BDT in stages adding more expensive variables at each stage to mainly reduce CPU from tracking

- For example, start with hadronic rate of ~50 kHz out of Level-1

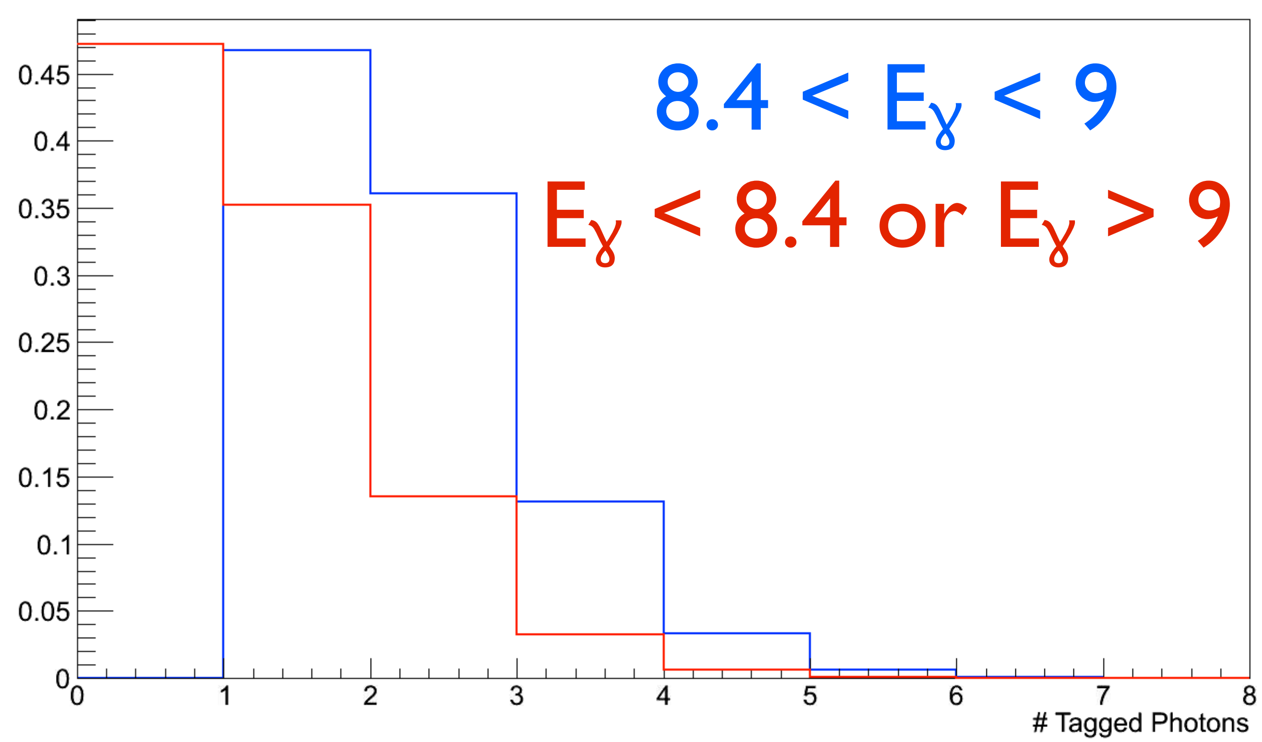
  - Train BDT at each stage with a subset of variables and make cut at $\epsilon = 0.99$

Stage	BDT Variables	Output Rate (kHz)
1	SC+FCAL	40.0
2	SC+FCAL+TOF	32.3
3	SC+FCAL+TOF+BCAL	30.1

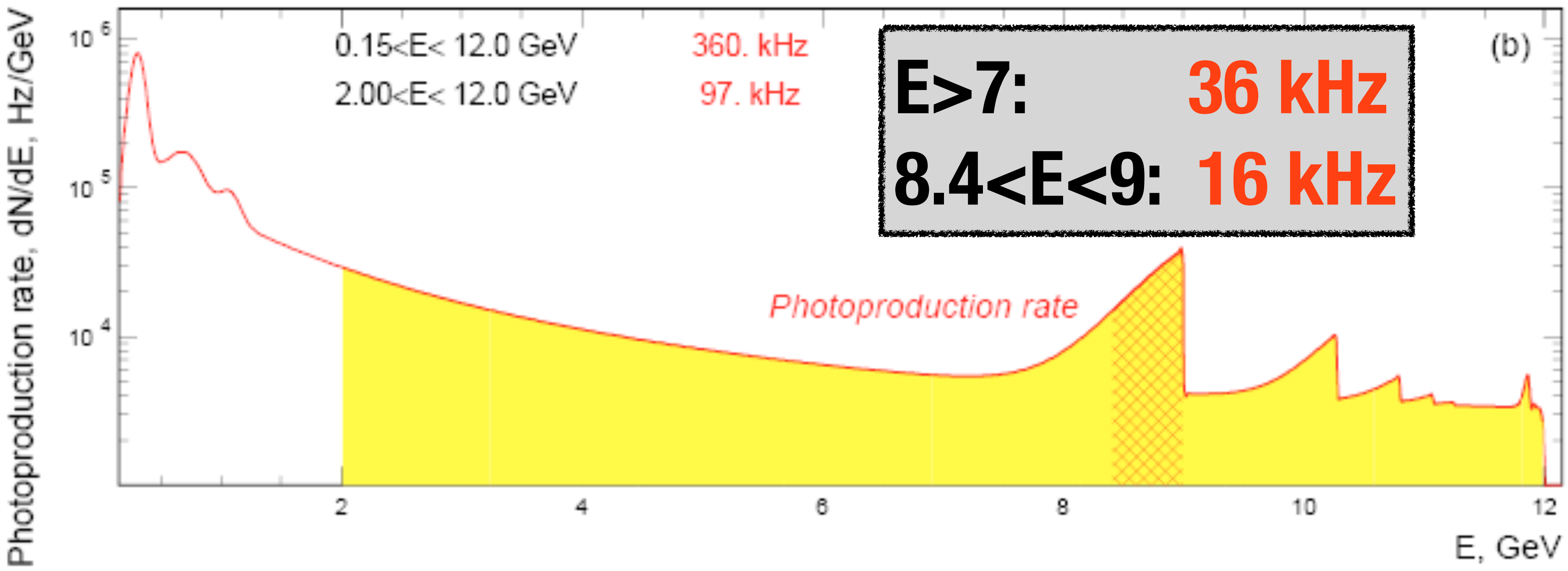
- Save roughly a factor of two in CPU time by staging

Tagger in L3

- Accidental tagged photon rate not currently in the simulation, but can model it with some numbers from Richard:
  - At 5×10^7 running, expect 0.25 accidental tags per beam bucket
  - Beam pulses every 2 ns, and tagger window of ±3 ns
- Summary: Use simple poisson statistics for (on average) 1 true + 0.75 accidental tagged photons for coherent peak events, and 0.75 accidentals for non-coherent peak events.
- Either cut on # of tagged photons or include in BDT



High Intensity: 10^8



- Can't take all $E_\gamma > 7$ GeV in 20 kHz since $E_\gamma > 9$ GeV not easily separated from coherent peak
- Need to make choices about physics priorities, some options:
  - Identify lower interest channels (with huge statistics from earlier lower intensity running) to ID and prescale
  - Identify characteristics of interesting channels (eg. strangeness: displaced vertex, CKOV upgrade, etc) to select events