Pileup Suppression Techniques: Looking to the Future

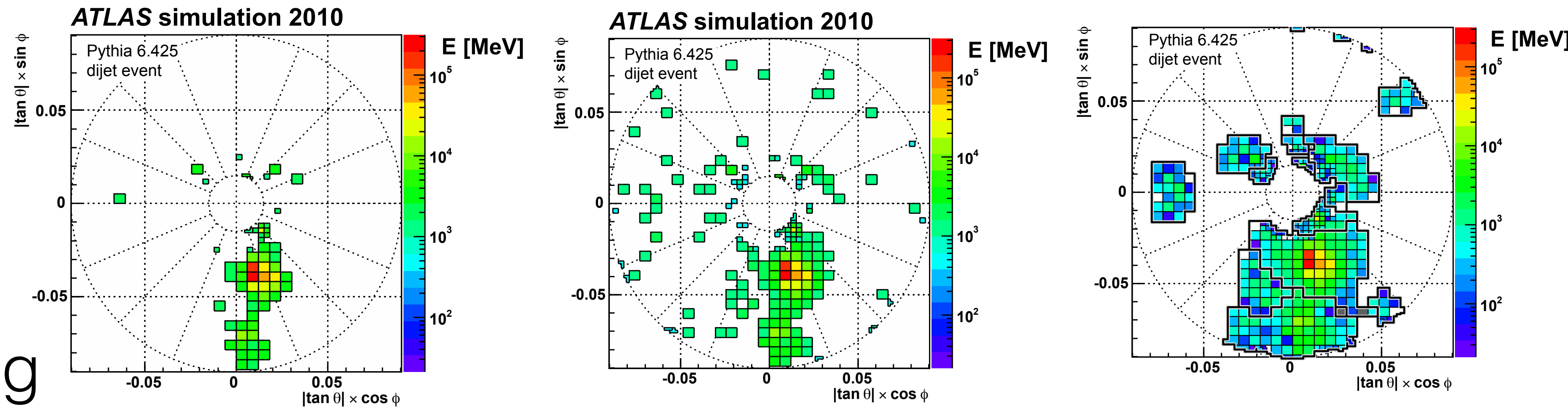
Jennifer Roloff
October 24, 2018

Pileup: Back to Basics

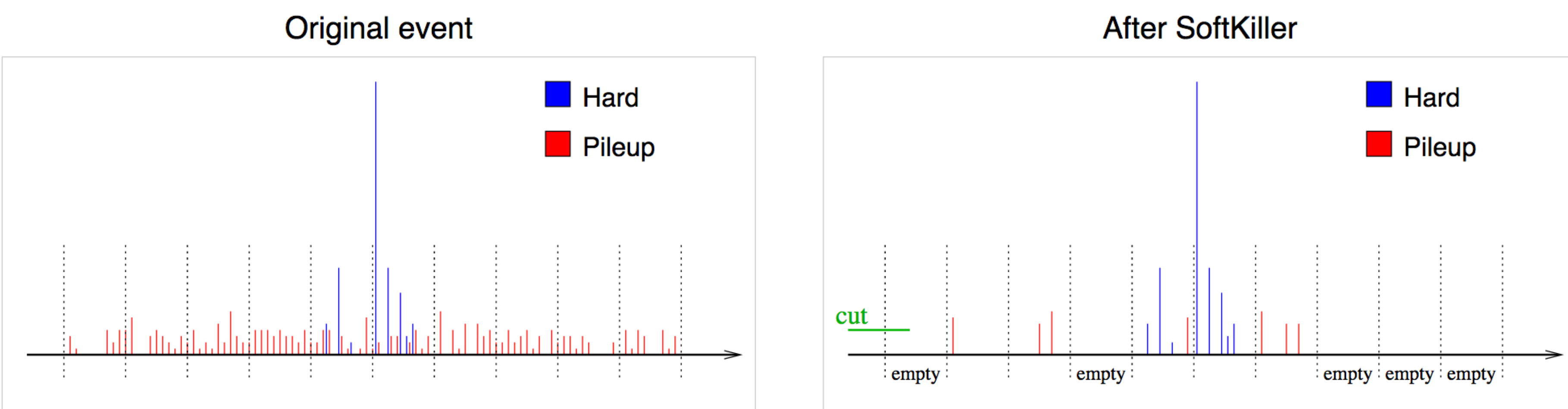
- Multiple collisions per bunch interaction at the LHC allows us to take huge amounts of data
  - These soft collisions contaminate events with hard collisions
    - Extra radiation included inside jets → worse resolution for all jet observables
    - More reconstructed objects → extra pileup jets in the event
- *Characteristics of Pileup*
  - Fairly soft collisions → low p_T particles
  - On average, fairly uniformly distributed in ϕ, and some slight dependence on η
  - For a given event, the energy density distribution is approximately Gaussian, with an average ρ, and a spread σ

How do we mitigate pileup?

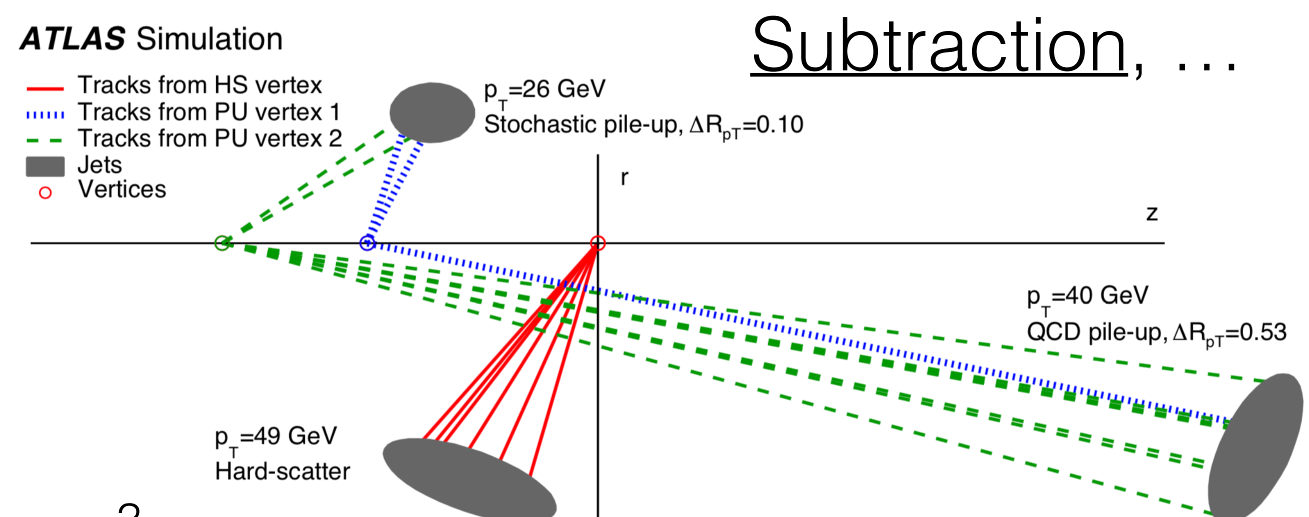
- ***Detector-level suppression***
    - e.g. Noise thresholds, timing information



- ***Constituent-level mitigation***
    - e.g. CHS, PUPPI, SoftKiller, Constituent Subtraction, Voronoi Subtraction, …



- ***Jet-level information***
    - e.g. Jet area subtraction, JVT , fJVT , Pileup Jet ID, jet cleansing



Pileup Mitigation

- What do we want in a pileup mitigation technique?
  - *Stability* — Same parameter choice works for a variety of observables over a wide range of pileup conditions
  - *Average correction* — Correction should produce the same result as zero pileup conditions on average
  - *Resolution* — Technique should result in as little worsening of resolution as possible
  - *Simplicity* — Easy to optimize parameter choices
- *To understand the future of pileup mitigation, we need to understand why current techniques work*
  - Have both event-level and local observables which can improve

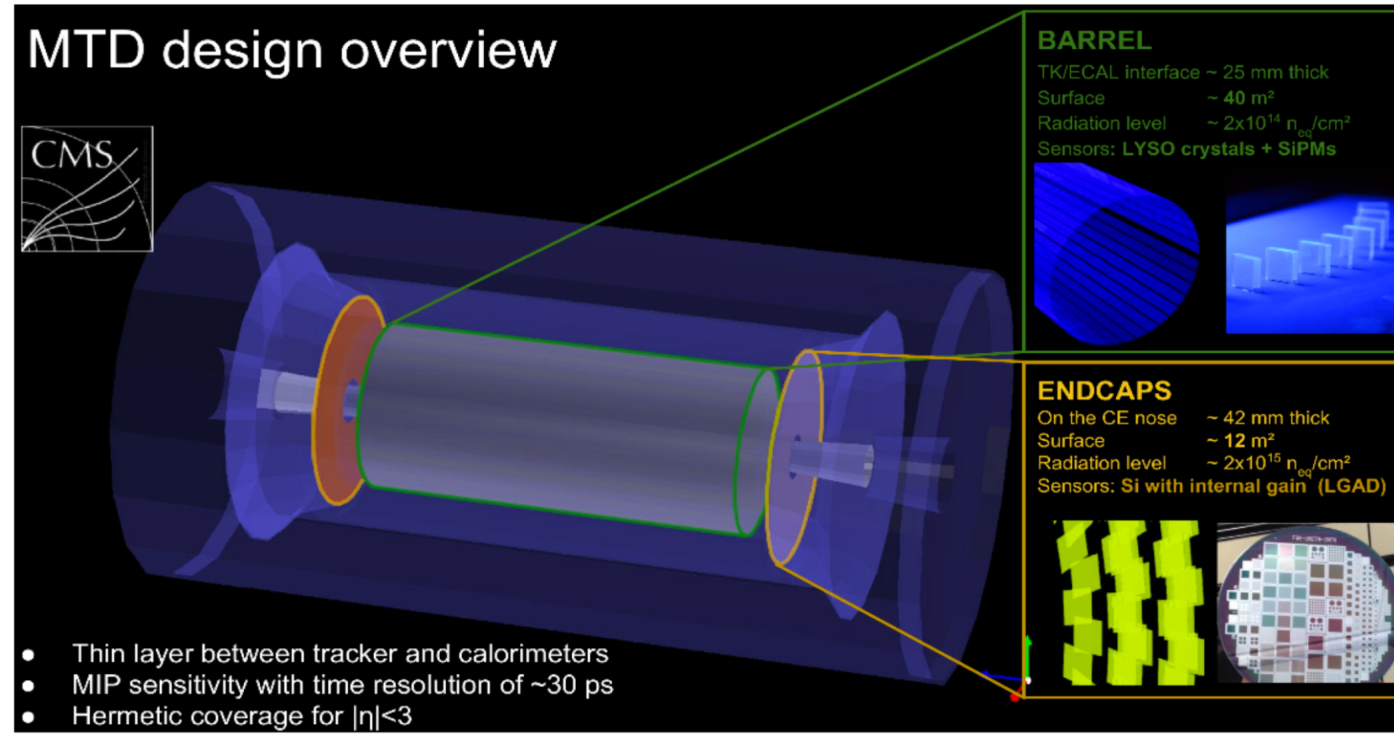
Event-Level Observables

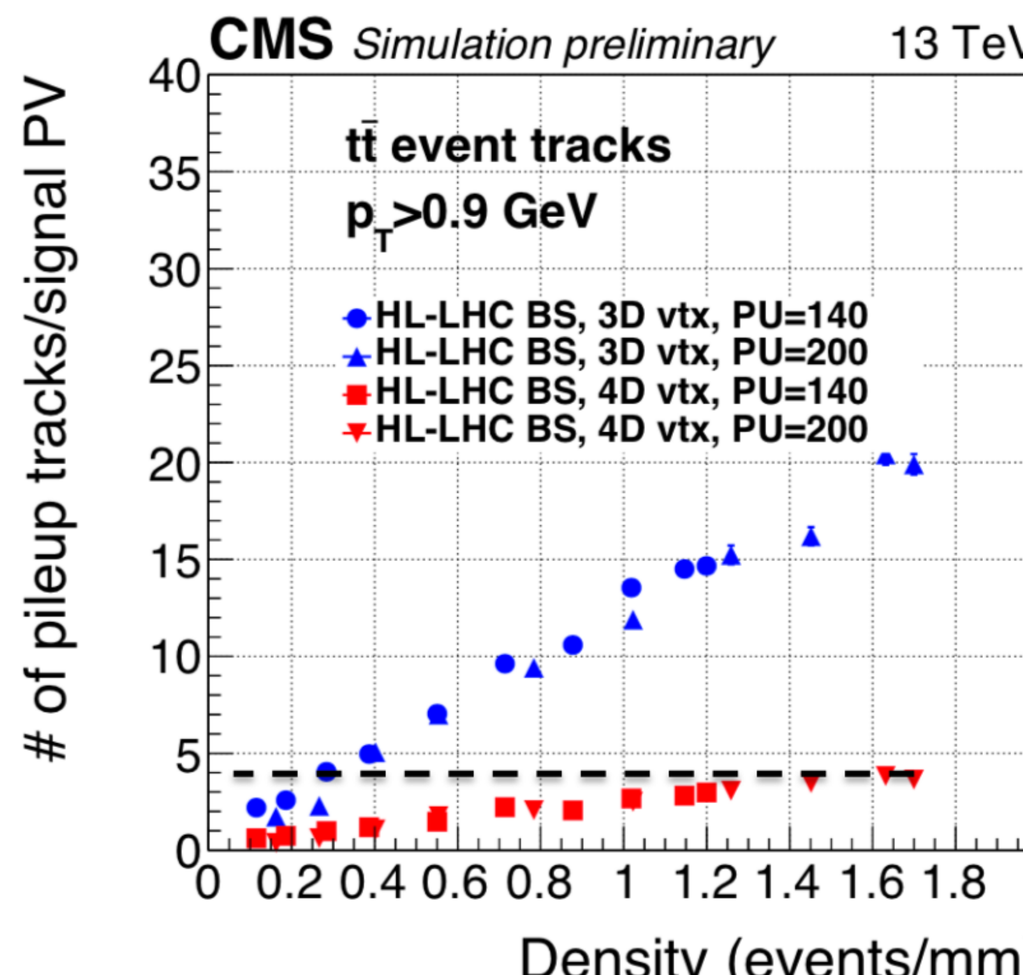
Detector Level

ATLAS HGTD

CMS MTD

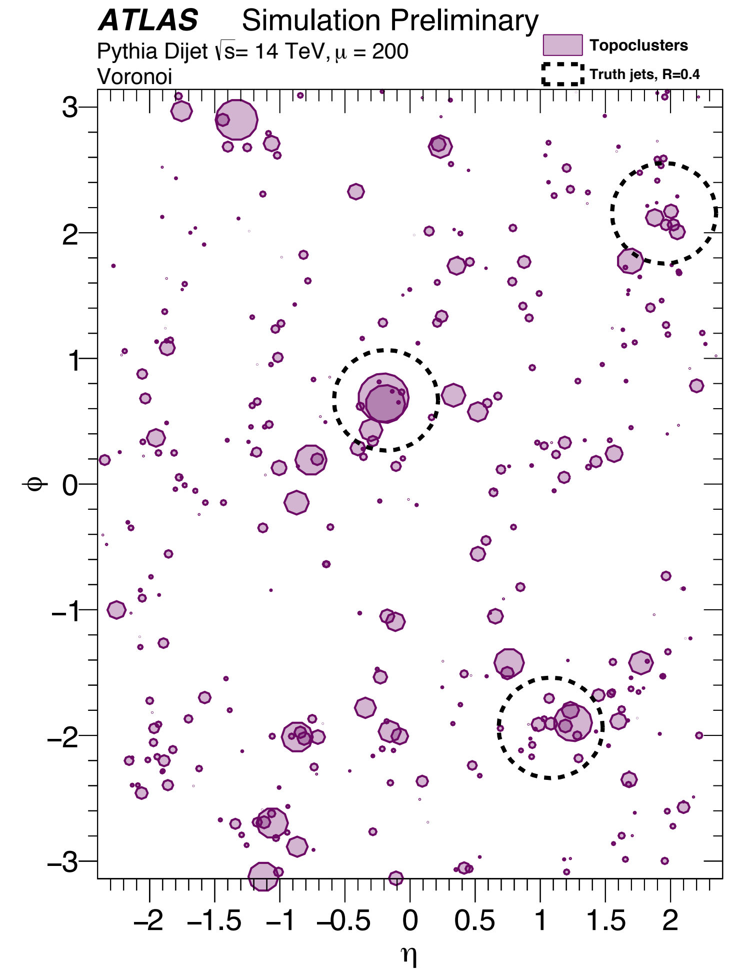
- Both ATLAS and CMS will be upgraded with timing detectors
  - Will cover different η regions
- Can help deal with vertex merging without requiring better vertex ID
- See yesterday's talk by Andrea for more information about this work on CMS

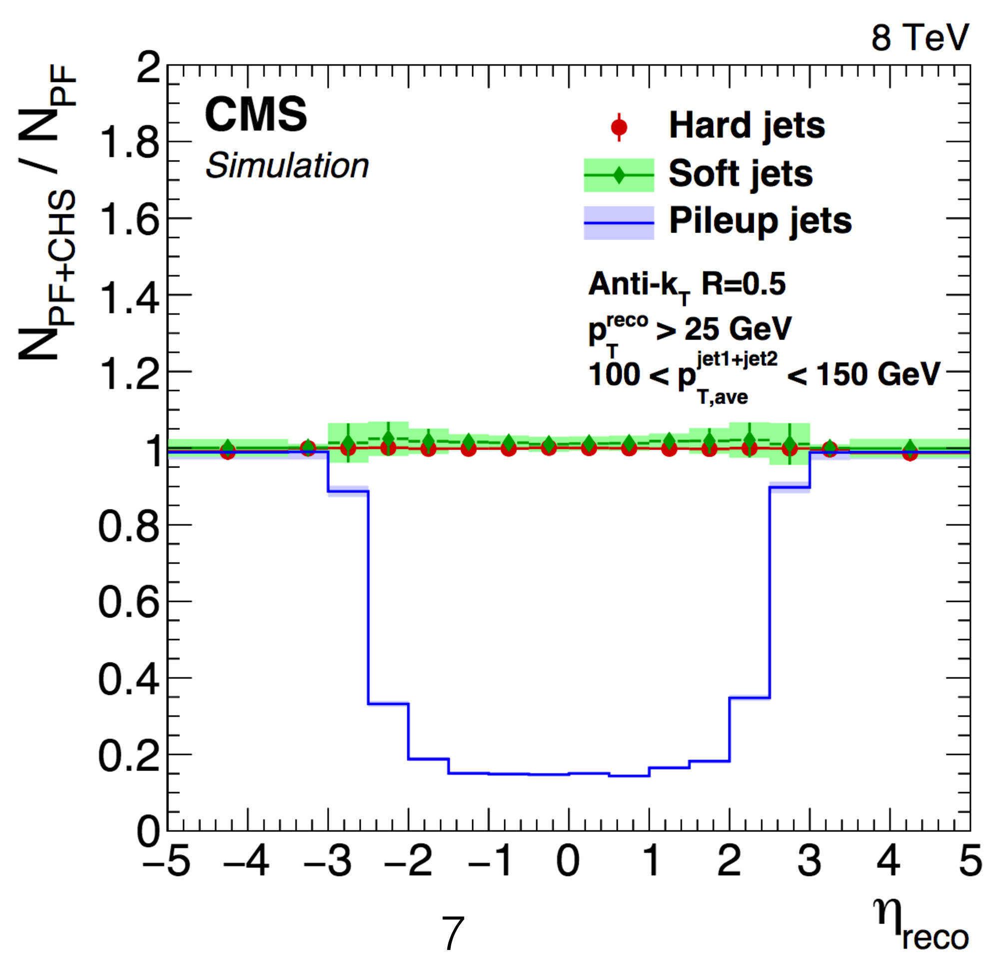


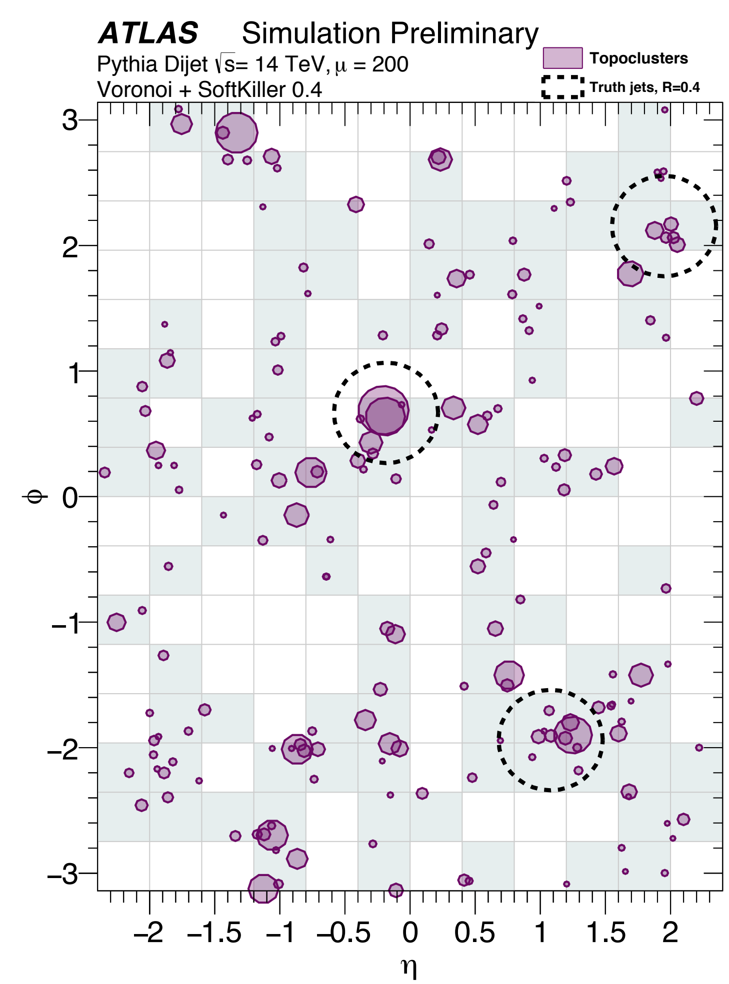


Object Level

- Event-level information can tell you how much pileup is present
  - NPV, mu, and ρ are several of the most common observables
- Median energy density **ρ** used for a many pileup mitigation schemes
  - The spread of ρ in an event is also of interest, though not often used
- Distribution of constituent properties for the event
- Vertex association for tracks

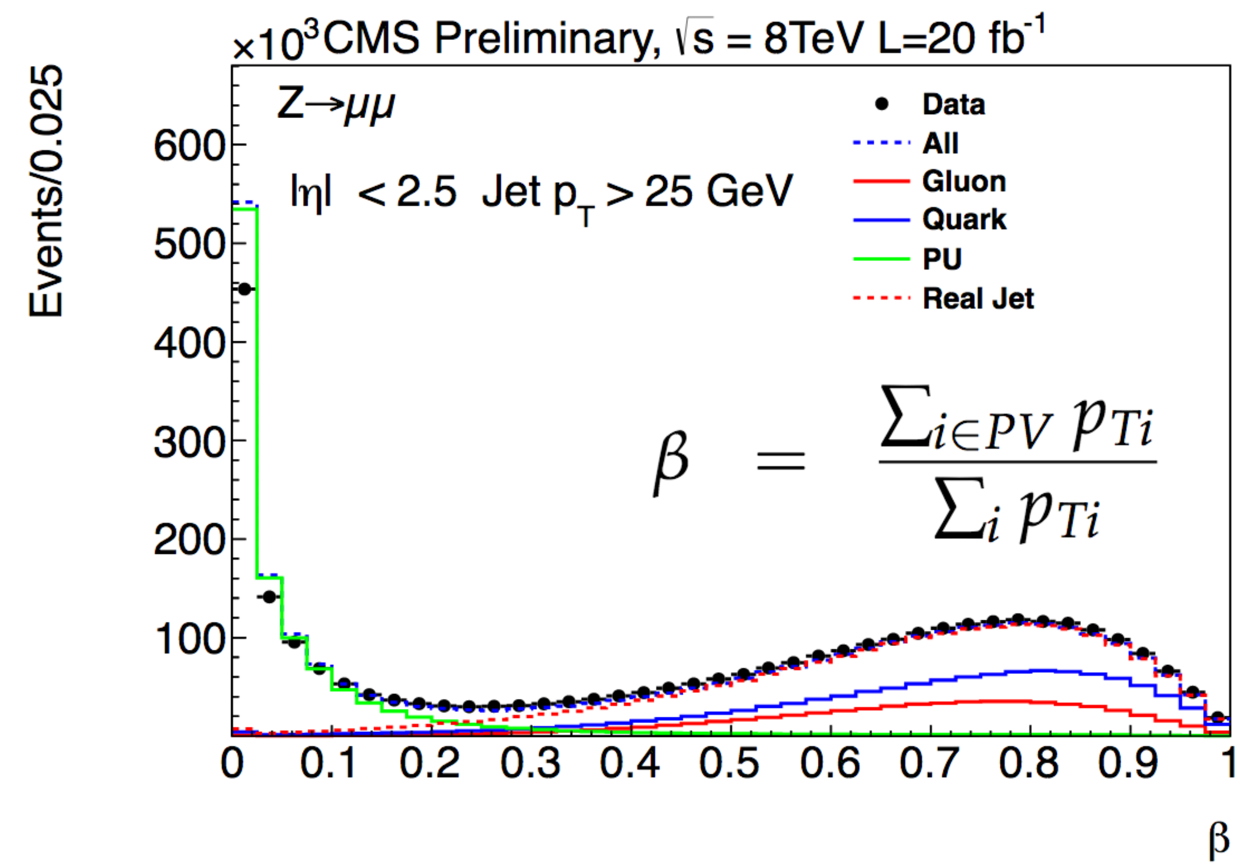


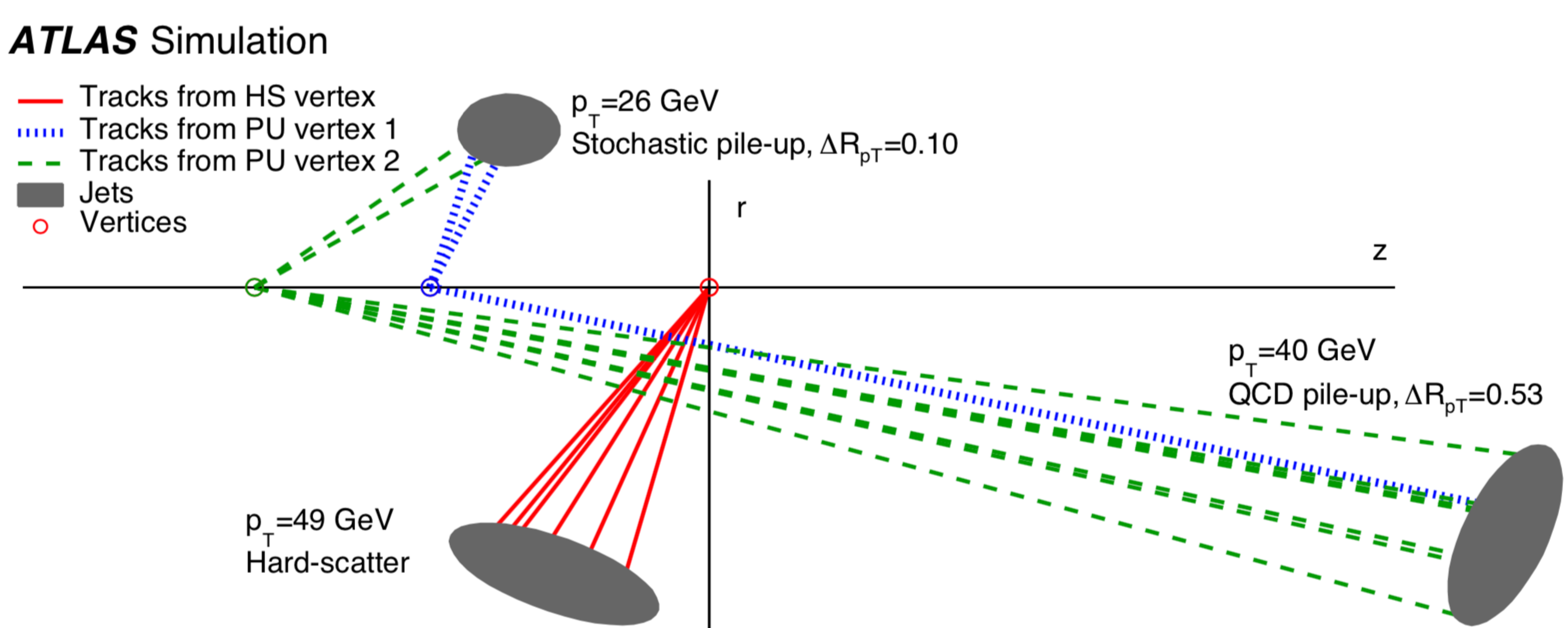




Jet Level

- Median energy density
  - Jet area subtraction
- Other jets in the event
  - Used by fJVT
- Vertex association of tracks in the jet

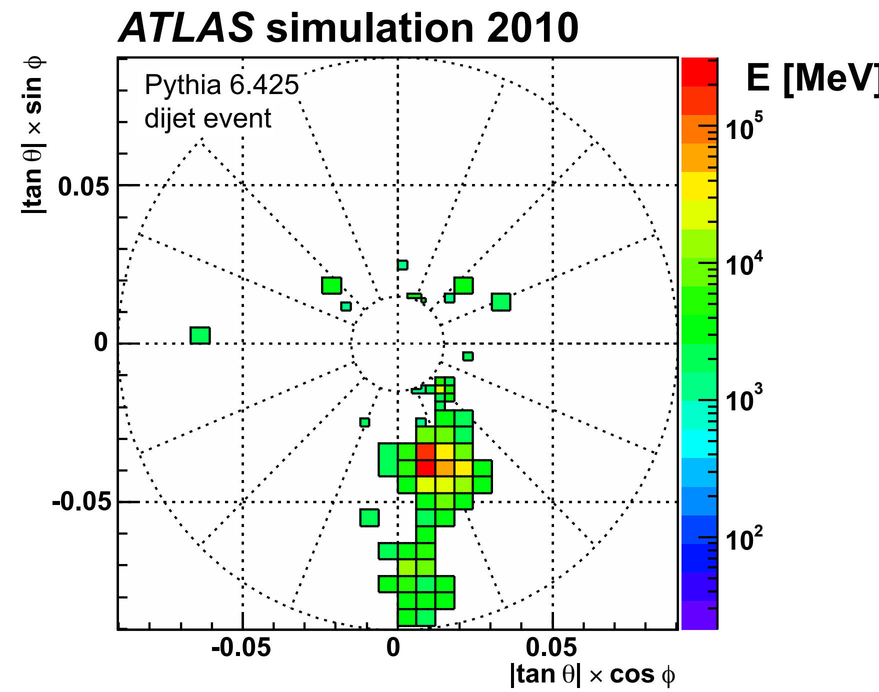


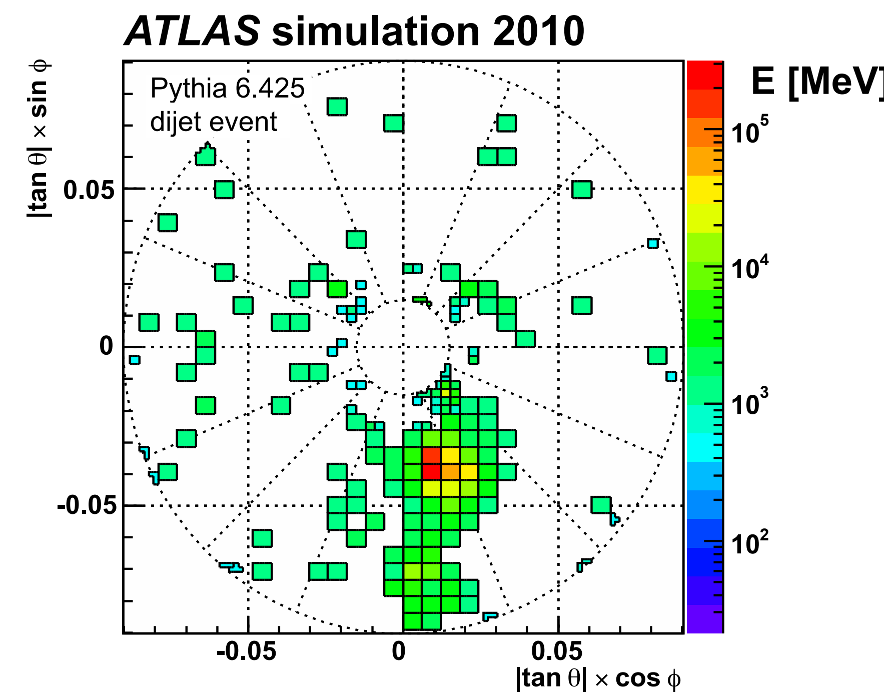


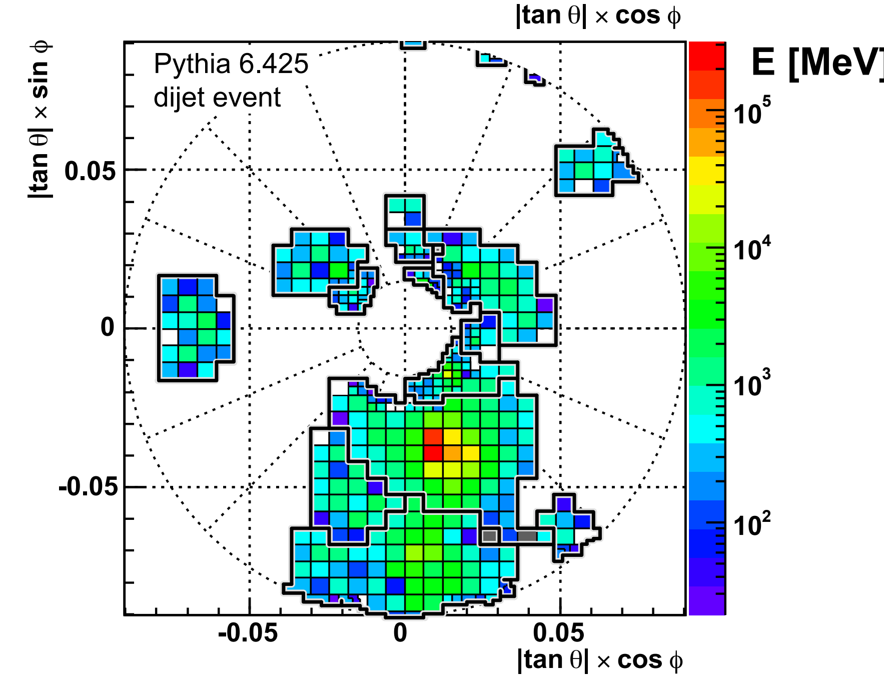
Local Observables

Detector Level

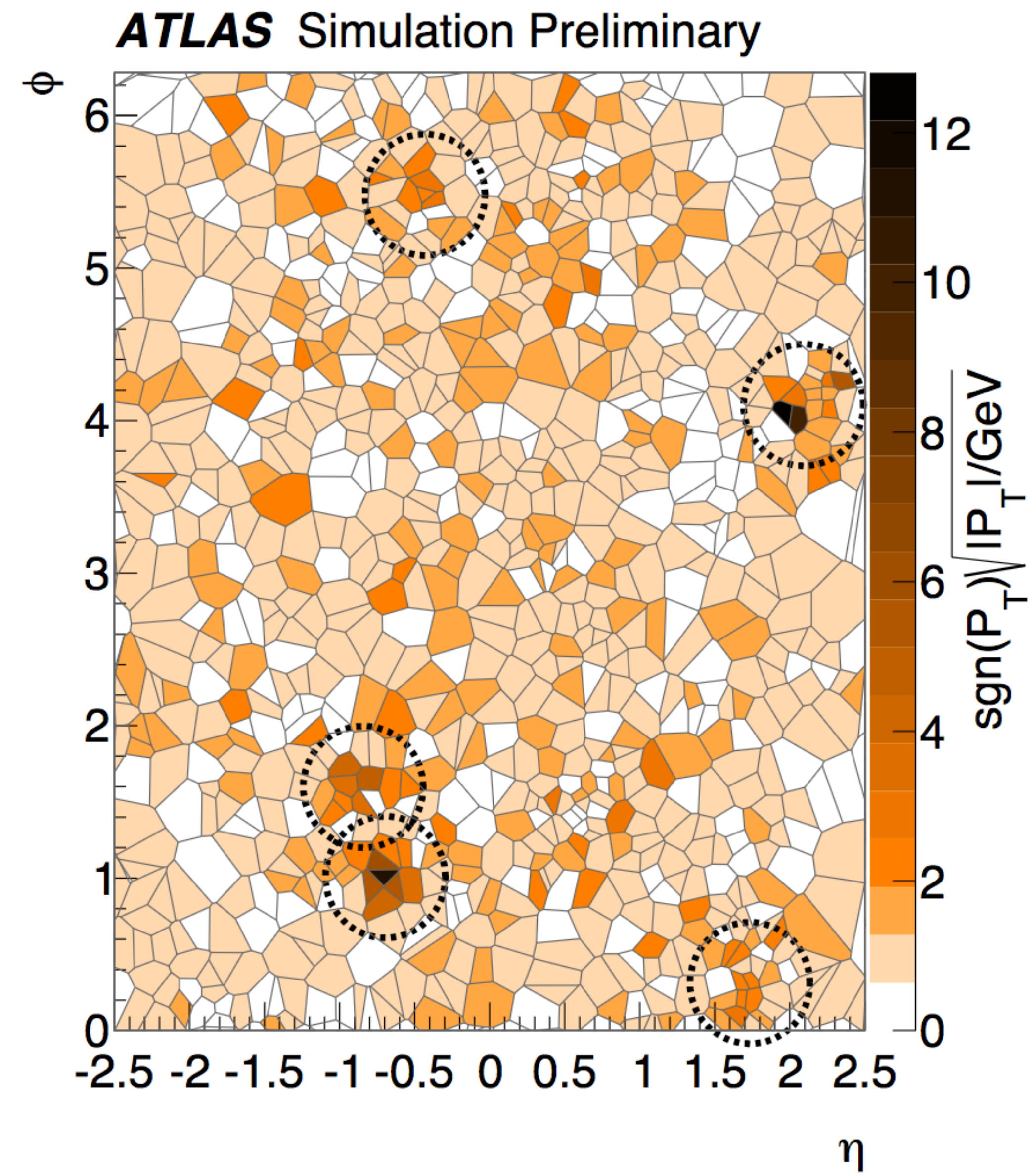
- ***Amount of energy deposited***
  - Used to seed clusters for both ATLAS and CMS
  - Helps eliminate low-energy deposits
- ***Nearby activity in the detector***
  - Used to include soft energy
- ***Timing information***
  - Can be used to eliminate out-of-time pileup





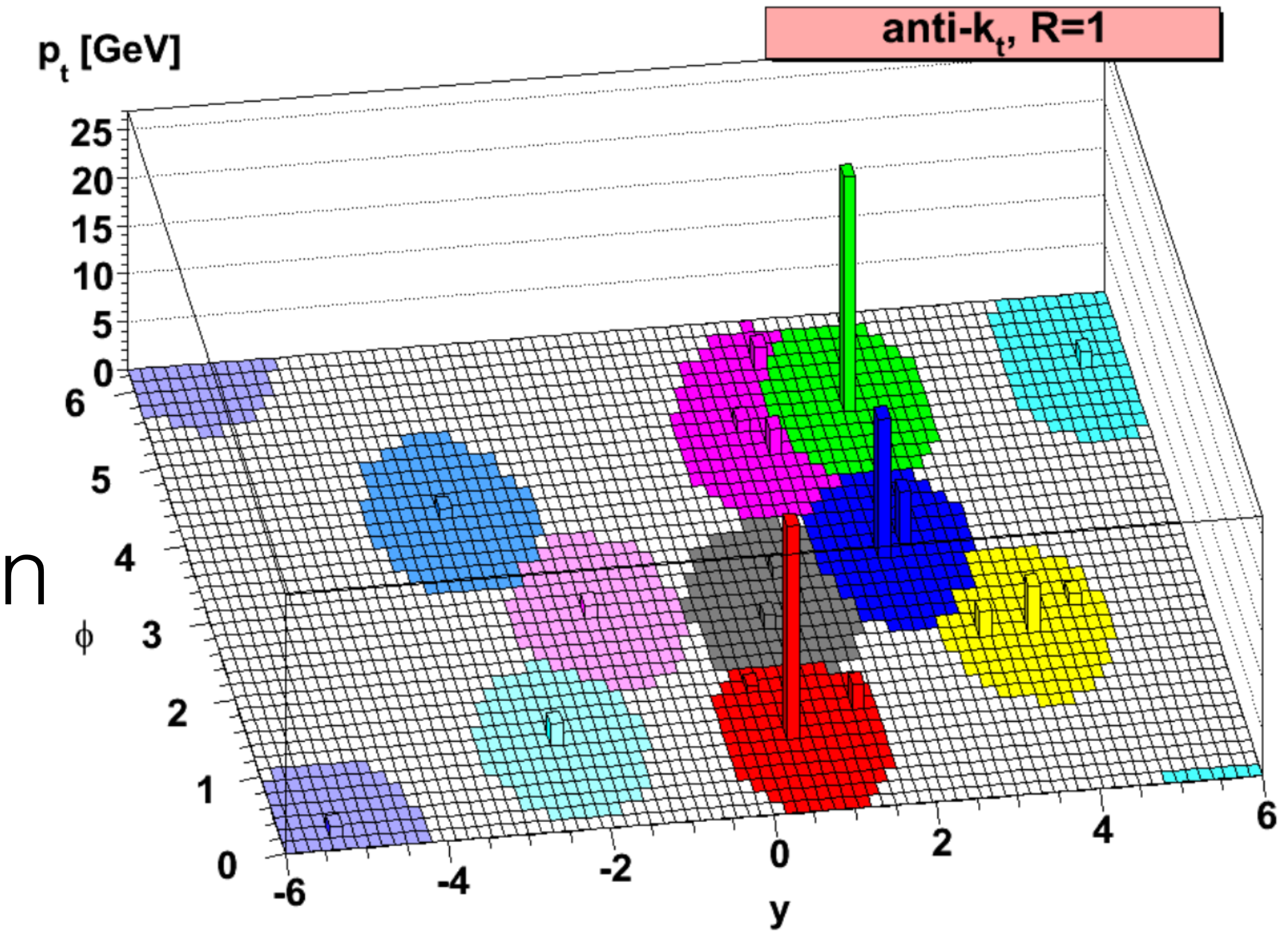


Object Level



- ***Distribution of nearby constituents***
  - Used by Voronoi subtraction to determine what area to assign to each constituent
  - Used by PUPPI to determine α
  - Used by Constituent Subtraction to determine ghost association
- ***p_T of constituents***
  - Used by PUPPI to determine α
  - Indirectly used by Constituent Subtraction to determine ghost subtraction
- ***Local energy density***

Jet Level



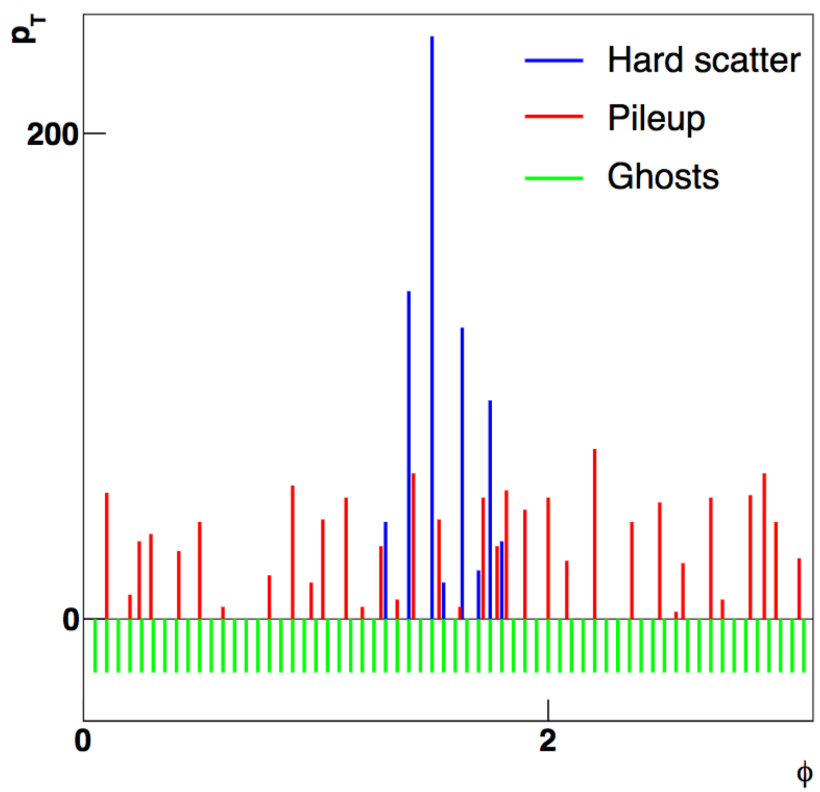
- ***Area of jet***
  - Used in jet area subtraction
- ***Jet width***
  - Used by fJVT
- ***Jet timing***
- Lots of different jet characteristics which can be used to discriminate between HS and PU

A Couple Examples

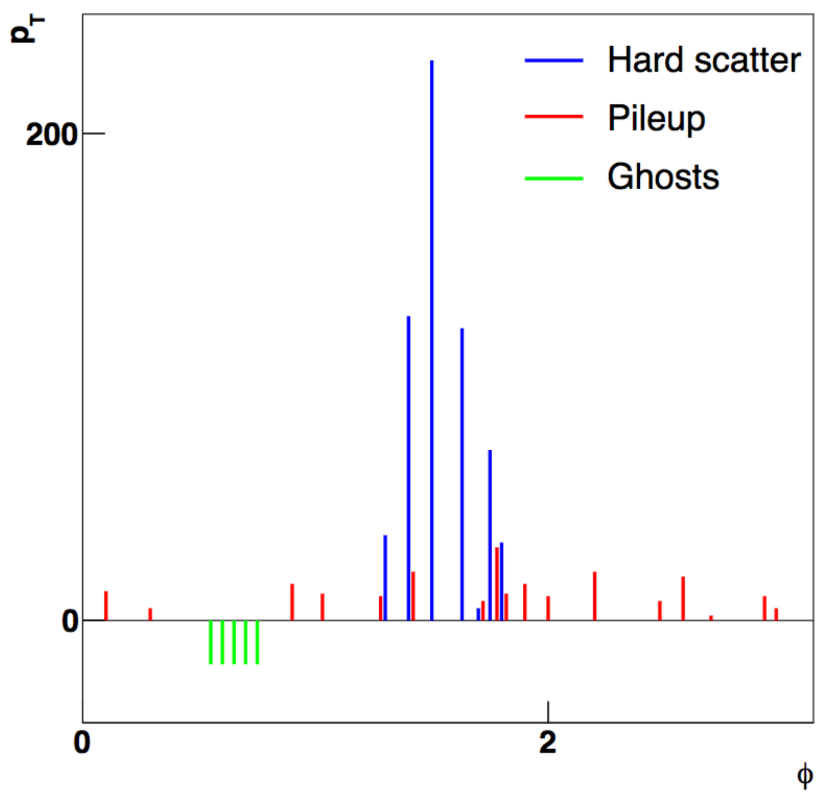
Case Study: Constituent Subtraction

- Calculate the median energy density ρ
- Add low-p_T ghosts to the event such that the energy density of ghosts is the same as the median energy density
- Cluster ghosts and constituents together using ΔR matching
  - Only match up to some maximum ΔR
  - Subtract off the ghost p_T from the matched constituent
- Once a constituent has zero p_T, it won't be matched to more ghosts

Constituent Subtraction



Whole event before correction

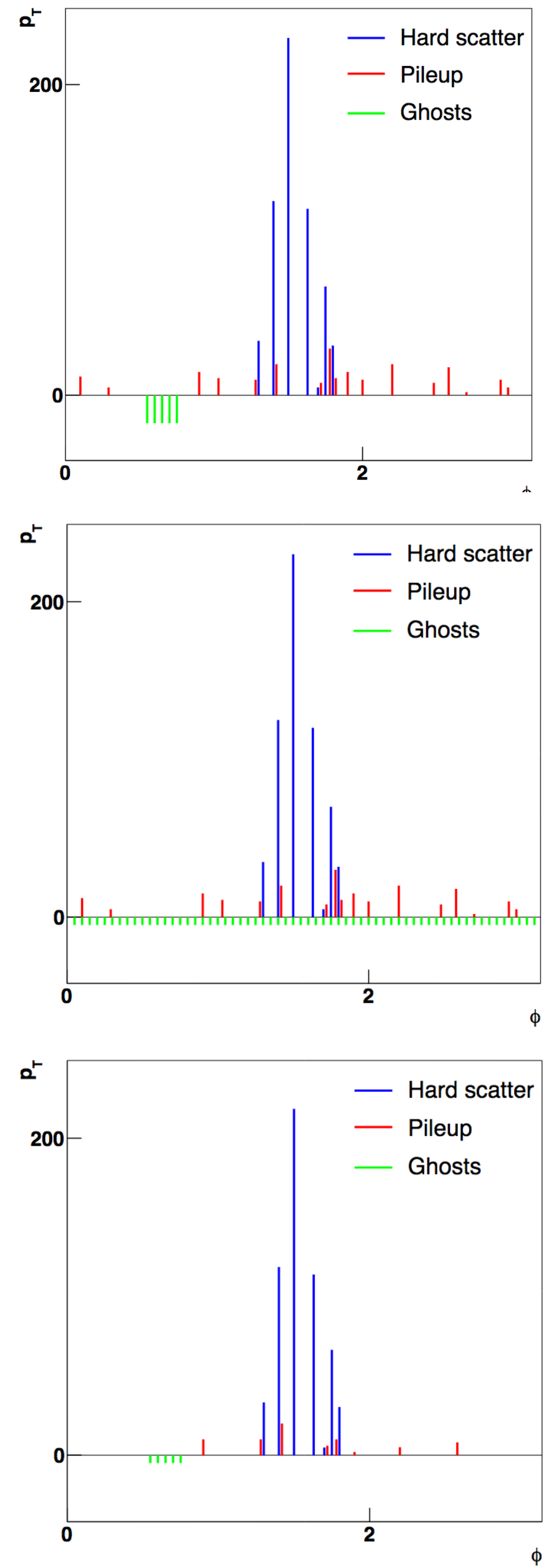


Whole event after correction

P. Berta @ BOOST2018

Case Study: Constituent Subtraction

- *ρ changes as a function of rapidity*
  - Introduced rapidity dependence in ghost momentum
- *ρ fluctuates across the event*
  - This means that we are under-subtracting in regions which fluctuate up
  - Could change radius parameter ΔR_{max}, but this leads to over-subtraction in jets
  - Iterative CS redistributes the remaining p_T to a set of new ghosts
    - Allows for additional subtraction in regions with energy density above ρ without overly biasing jets



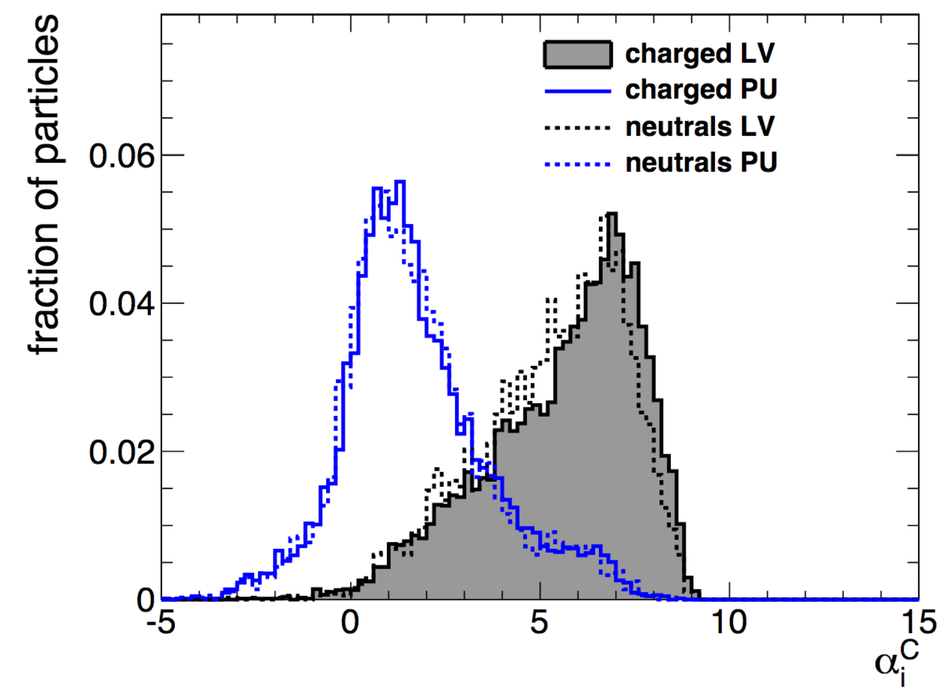
Case Study: PUPPI

- PUPPI uses a local variable α to signify how HS- or PU-like a constituent is

$$\alpha_i = \log \sum_{j \in \text{event}} \xi_{ij} \times \Theta(R_{\text{min}} \leq \Delta R_{ij} \leq R_0)$$

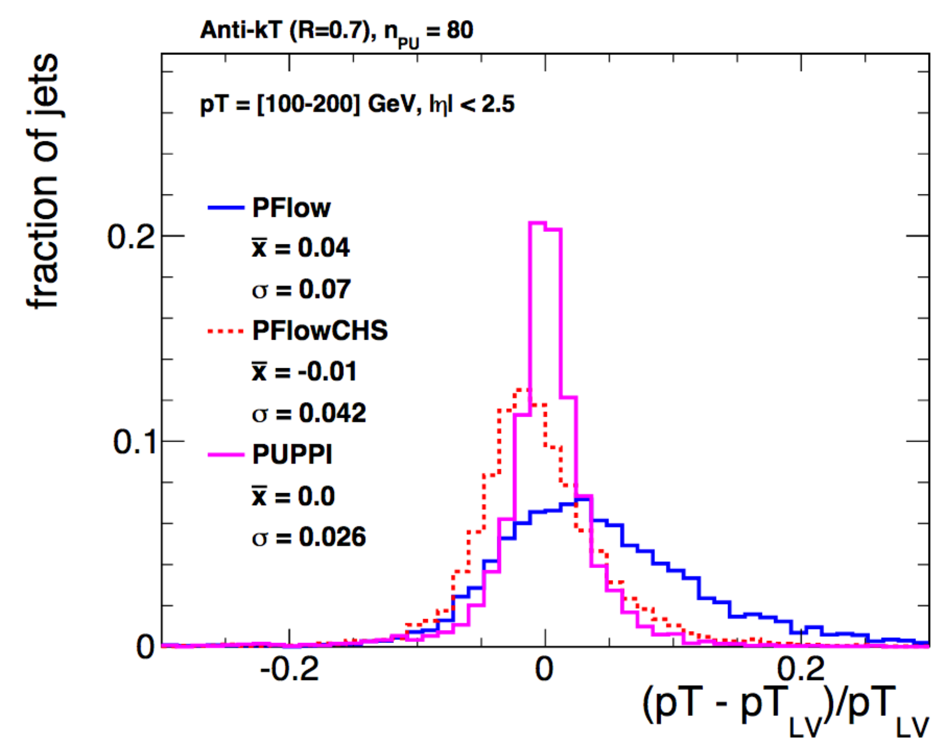
$$\text{where } \xi_{ij} = \frac{p_{Tj}}{\Delta R_{ij}}$$

  - Quantifies at how close it is to hard PU or HS activity



- Uses the distribution of α for charged PU in that event to determine the weighting for the neutral constituents in the event

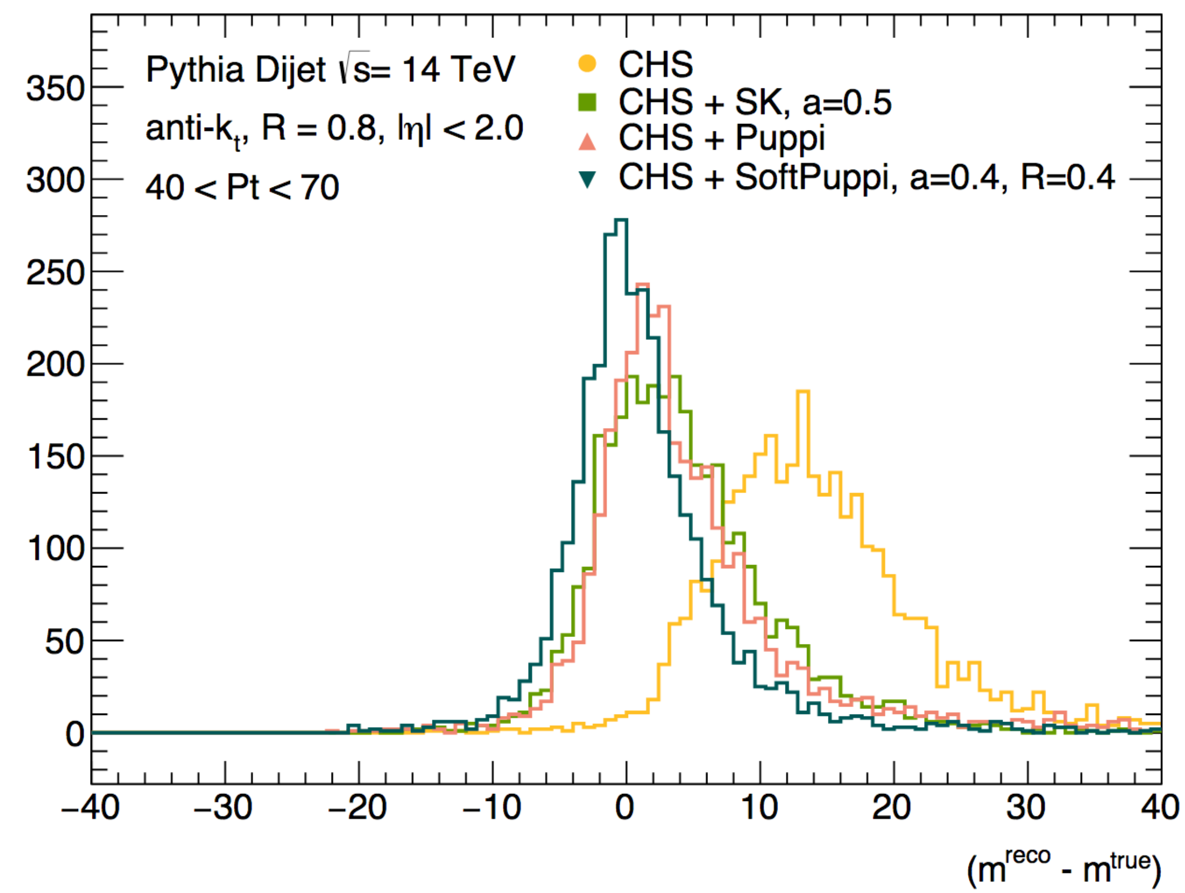
- Apply an NPV-dependent p_T cut to the constituents
  - Different treatment needed in regions with or without tracking
  - In total, have somewhere around 6+ parameters to optimize



Case Study: PUPPI

SoftPuppi

- PUPPI has lots of tunable parameters
  - Can use SoftKiller instead to determine p_T cut → cuts down parameters significantly
- Many other possible improvements to the α metric as well
  - Currently only considers relationship to HS constituents
    - Why not also include information about PU vertices?
  - Could also incorporate other pileup information into the metric



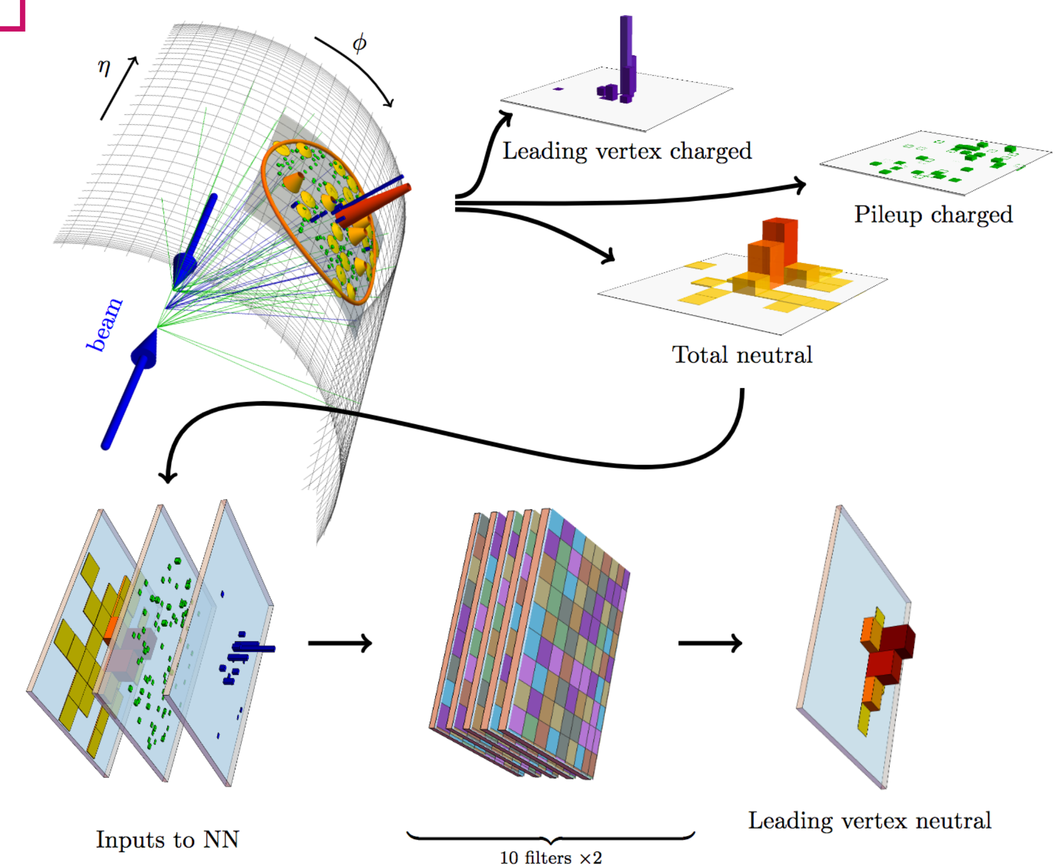
NPU = 100

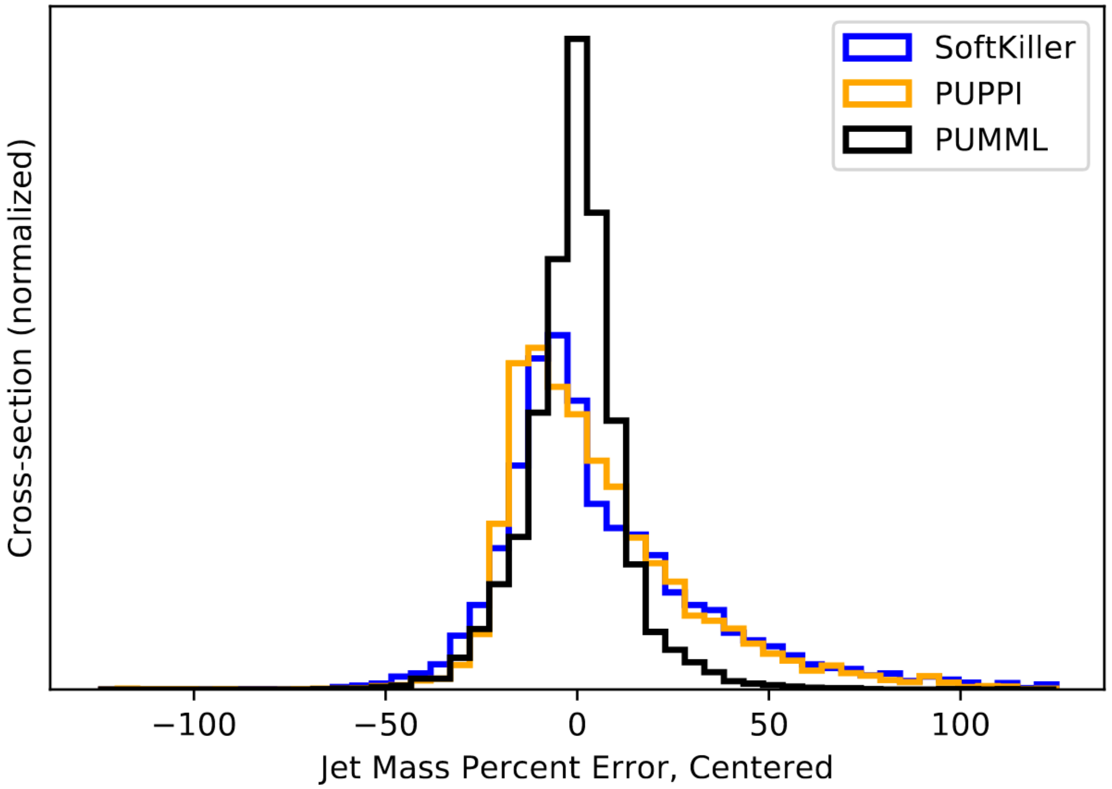
Bringing it all together: Machine Learning

- Machine learning techniques can help tie all of these ideas together
  - Need to give relevant information to deep learning algorithms in order to make use of them
- Challenges: what is the best way to use all relevant information?

Case Study: PUMML

- PUMML: **P**ile**U**p **M**itigation with **M**achine **L**earning
  - Uses a convolutional neural network with a jet image to determine what to subtract
- Jet images indirectly contain information about several pileup-related observables
  - Encodes charged HS, charged PU, and neutral activity separately
  - p_T of constituents → can eliminate low p_T pileup constituents
  - Density of constituents → can reduce non-collinear emissions
- Still more information that could be included
  - Event-level information





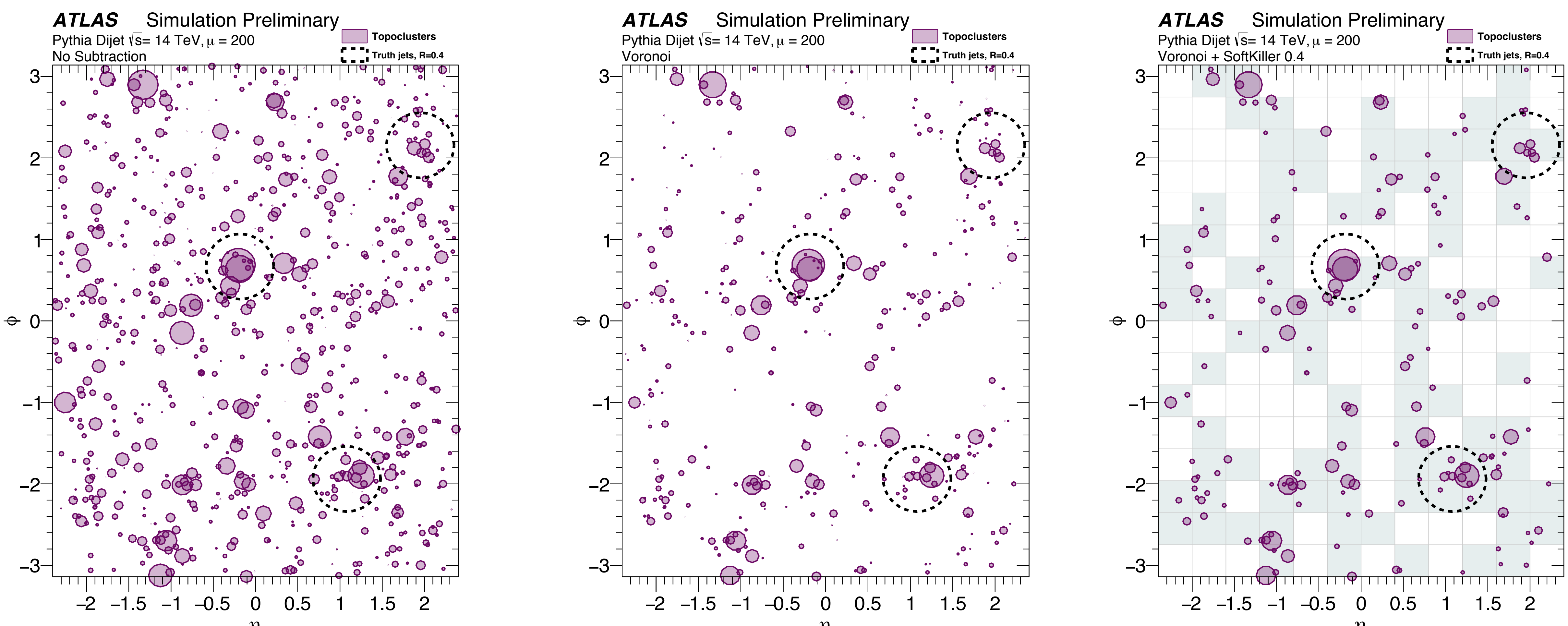
Summary

- ATLAS and CMS are both using a variety of techniques to deal with pileup
- Current techniques can be improved in a few different ways
  - *Encoding information differently* — what are the best, most concise ways of representing the information we have?
  - Including *more pileup-sensitive observables* in current algorithms
  - Creating *new observables* which are sensitive to pileup
- *Machine learning* can help bring together information from a variety of sources
  - Still a variety of things to understand about how to do this best
  - How do we best represent our jets and events?

Backup

Pileup Mitigation: SoftKiller

- Determines an event-by-event p_T cut for constituents
  - Should apply either Voronoi Subtraction or Constituent Subtraction first
- Makes a grid, finds p_T cut where half of grid cells are empty afterwards
  - Makes the median energy density approximately zero



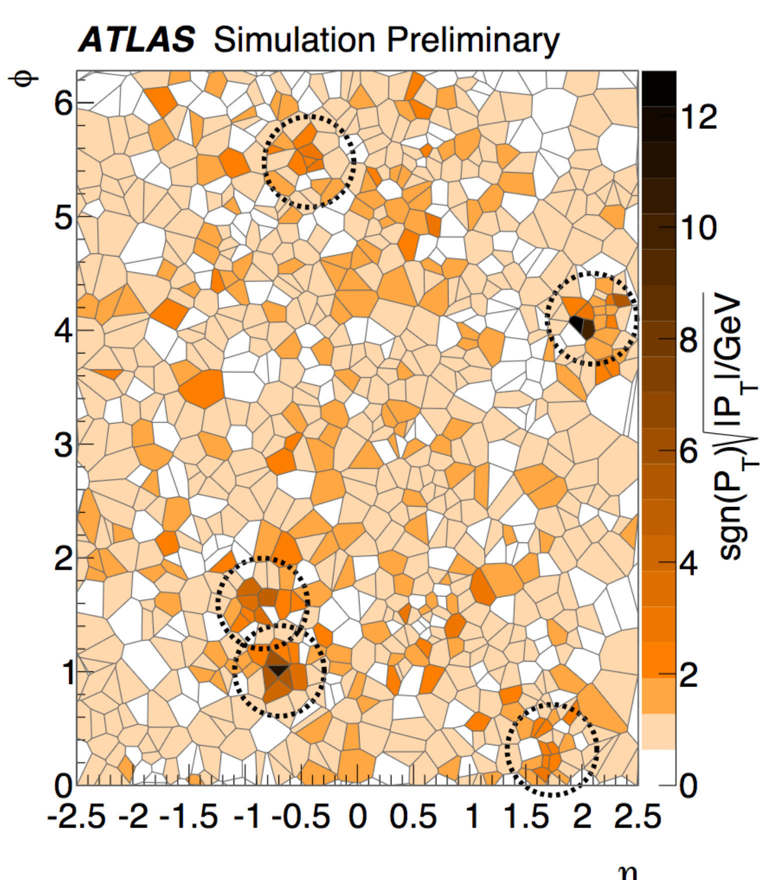
SoftKiller Paper

CONF note on pileup mitigation

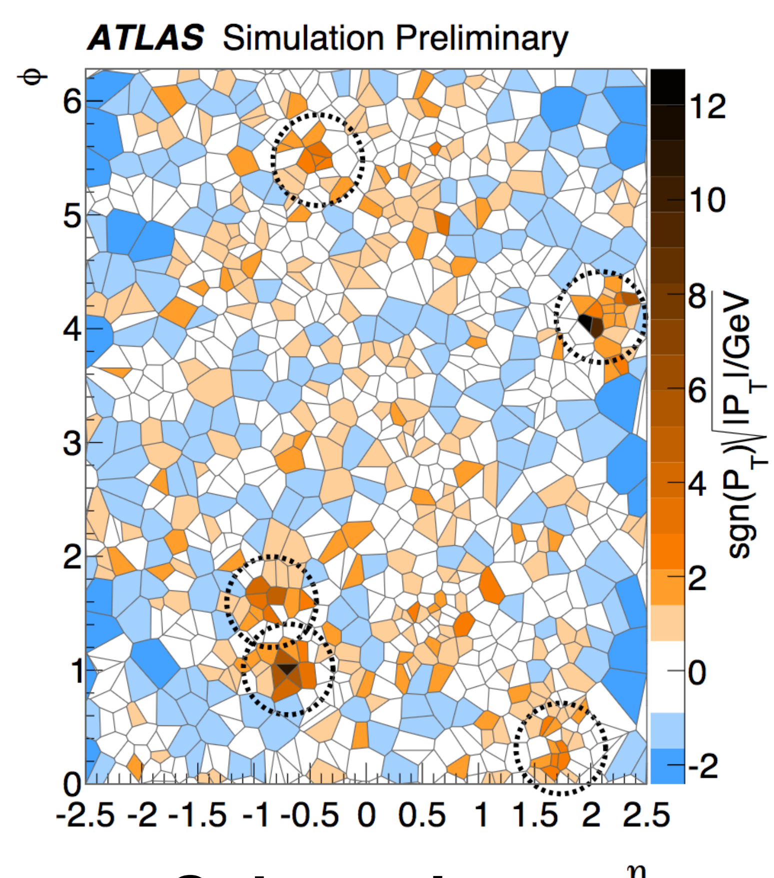
Voronoi Subtraction

CONF note on pileup subtraction

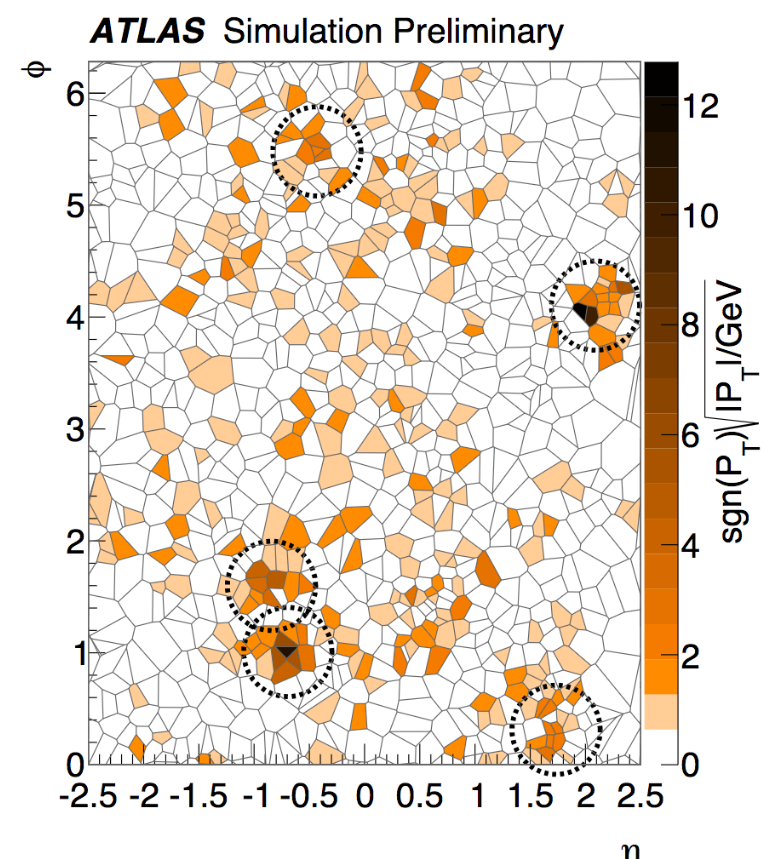
- Voronoi subtraction is a type of constituent-level pileup mitigation which uses the **median energy density** (rho) and the **Voronoi area** to reweight constituents
  - Voronoi area is the area of points in η-ϕ space which are closer to a constituent than any other
- Voronoi subtraction will leave some constituents with negative pT — use **Voronoi suppression,** which discards any constituents with negative pT



Uncorrected



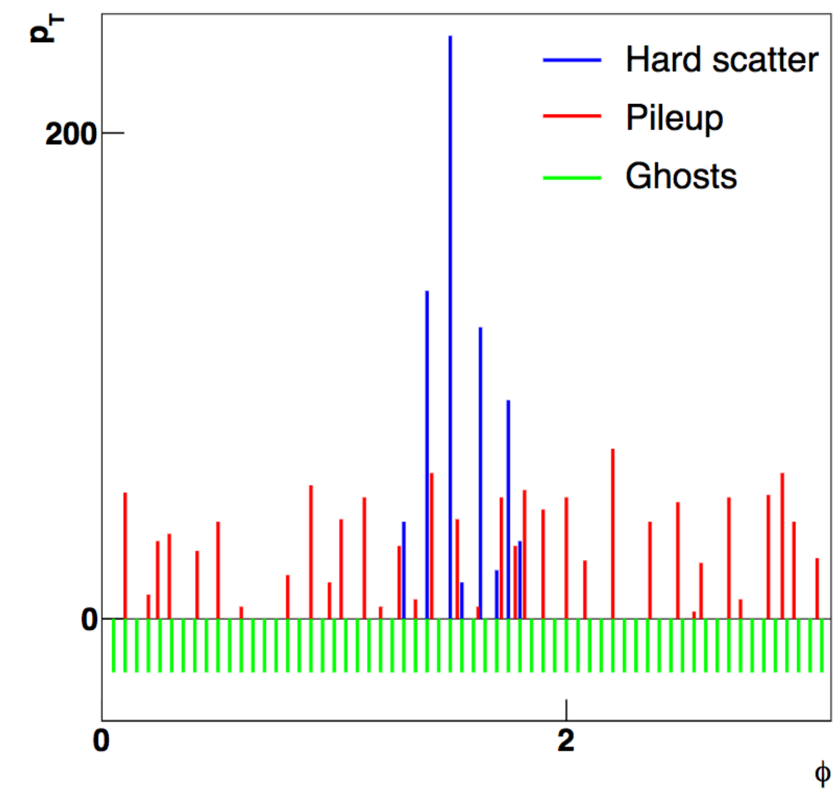
Subtraction



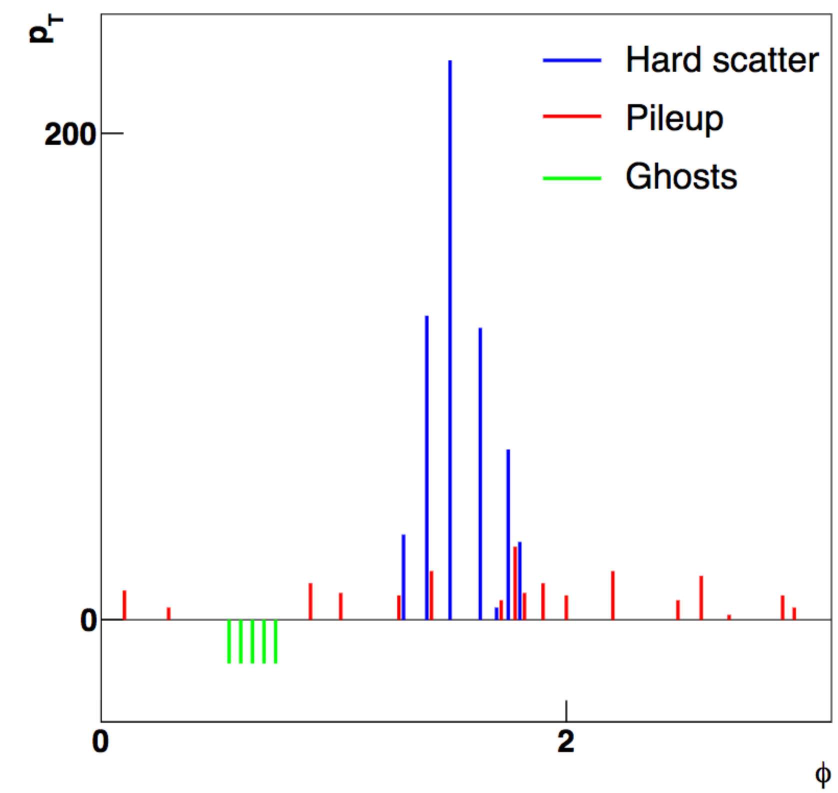
Suppression

Pileup Mitigation: Constituent Subtraction

- Constituent-level pileup mitigation technique which *rescales* the constituent 4-momentum
- Adds ghosts evenly throughout an event with p_T density equal to the median energy density ρ
- Ghosts matched to constituents, and the ghost p_T is subtracted off
  - Only matched within some maximum ΔR of the constituent
- After subtraction, the median energy density should be approximately zero



Whole event before correction



Whole event after correction

Constituent Subtraction Paper

CONF note on pileup mitigation