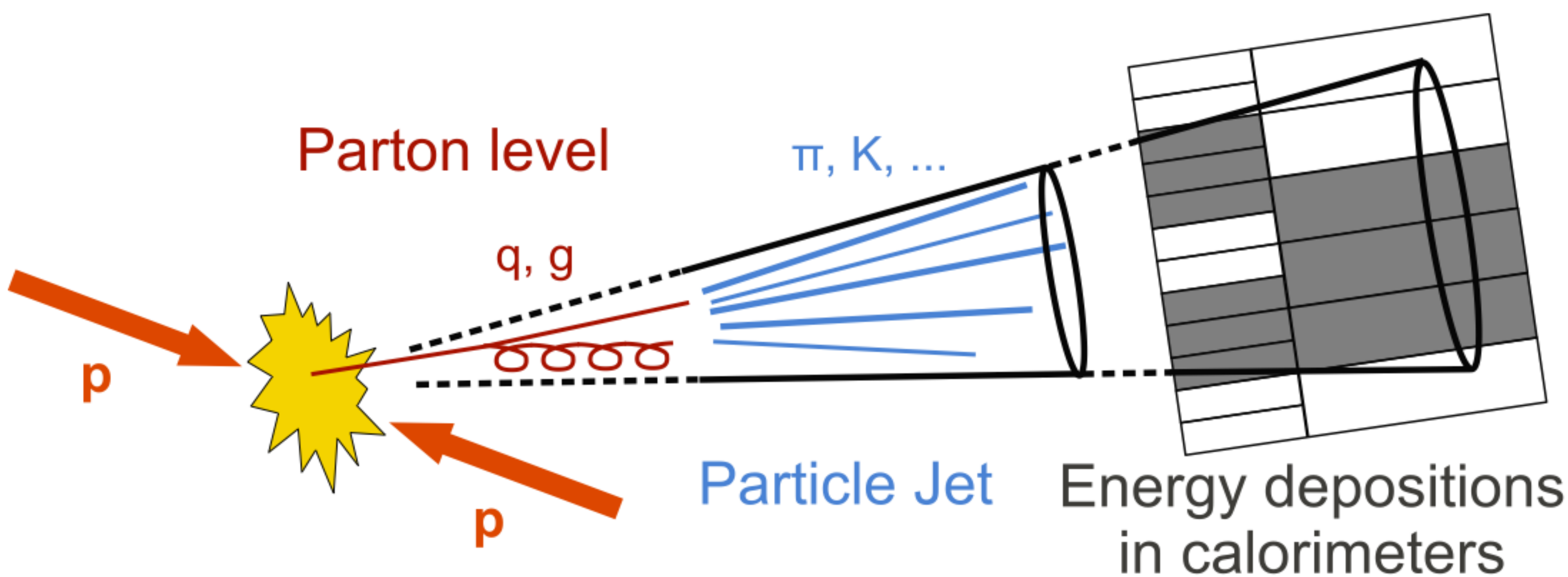
Jet Physics and Machine Learning: Lecture 2 :
Jet reconstruction & measurements

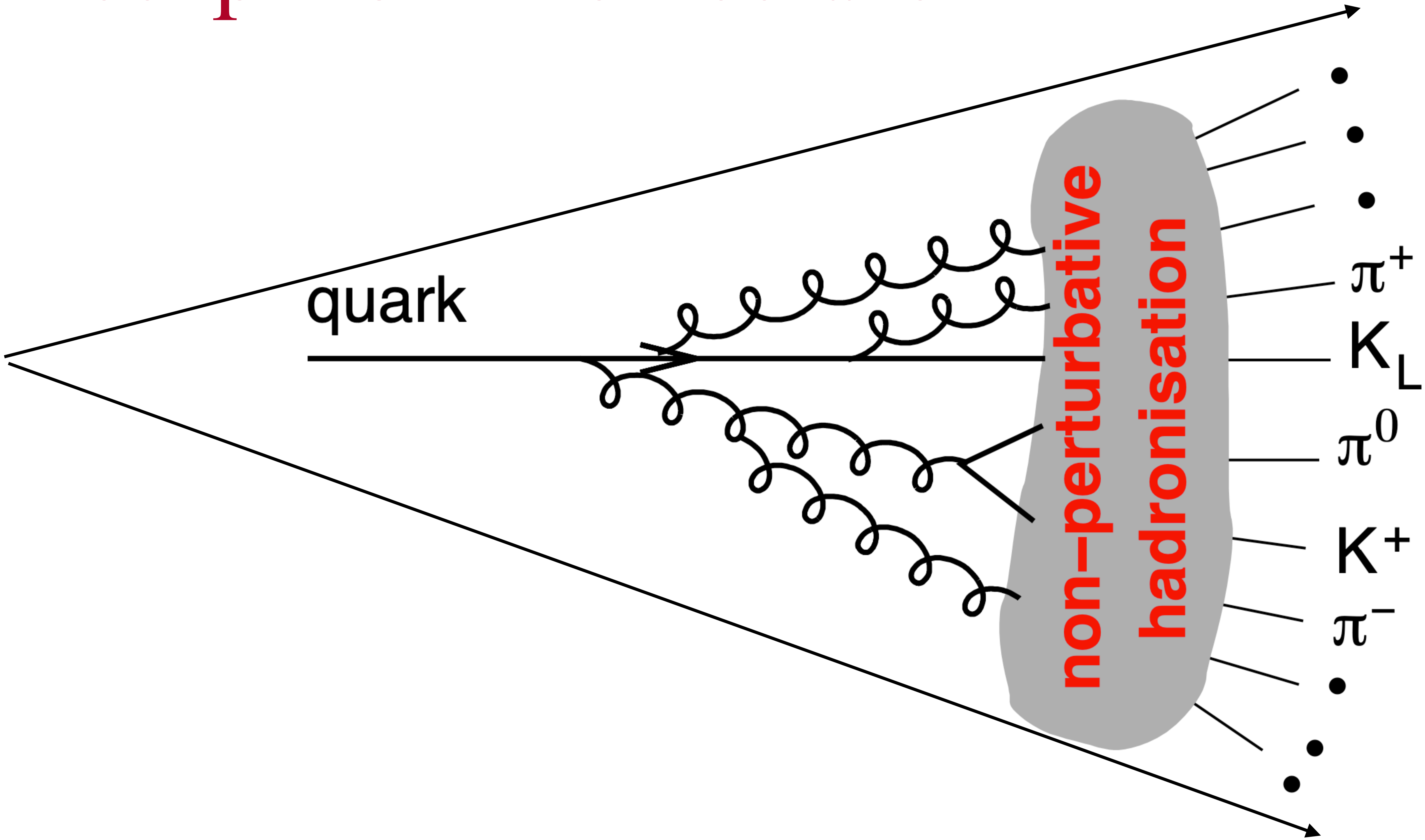
Sanmay Ganguly
IIT-Kanpur
sanmay@iitk.ac.in

25/02/2025

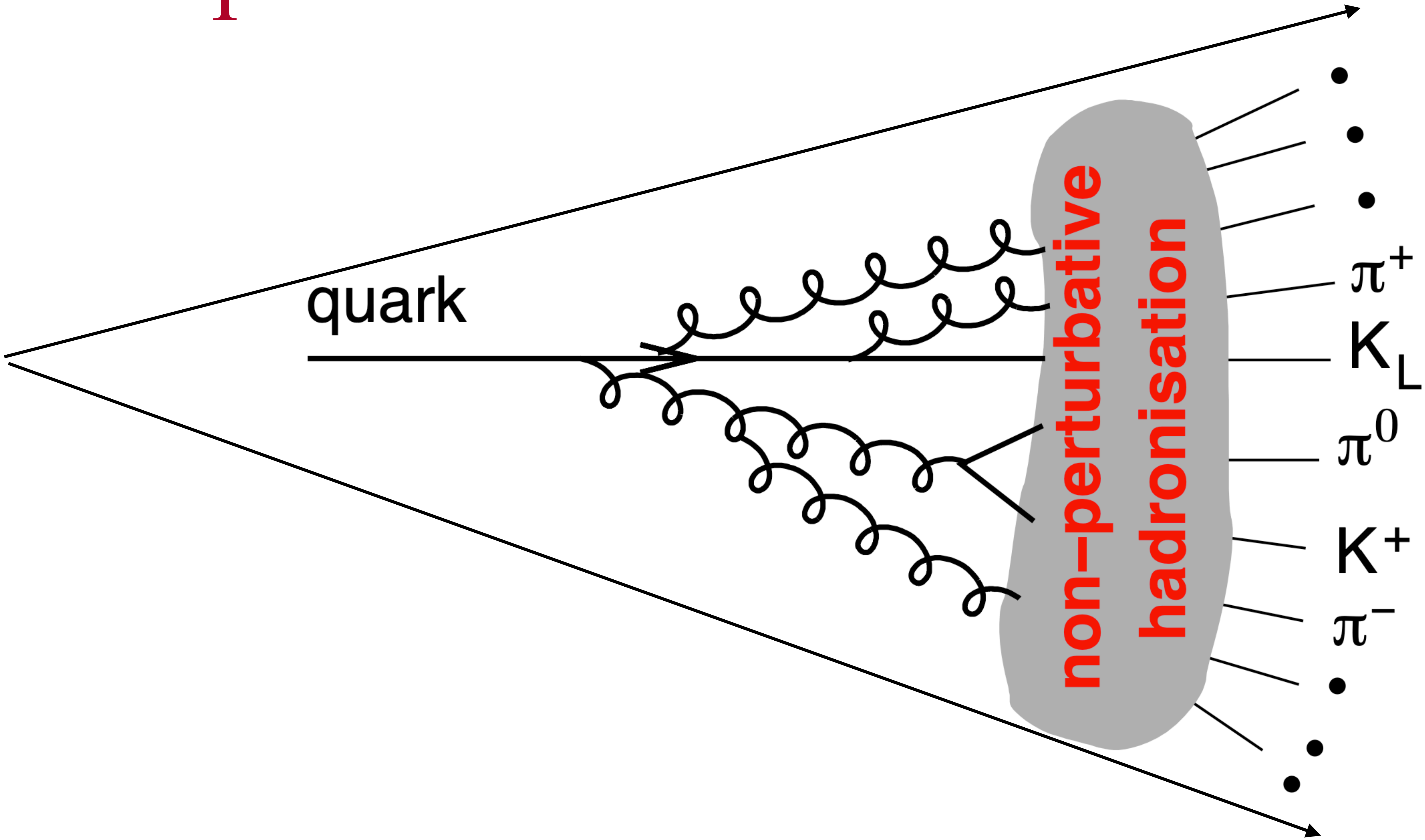
IMSc Spring School on High Energy Physics - 2025



Recap from last lecture

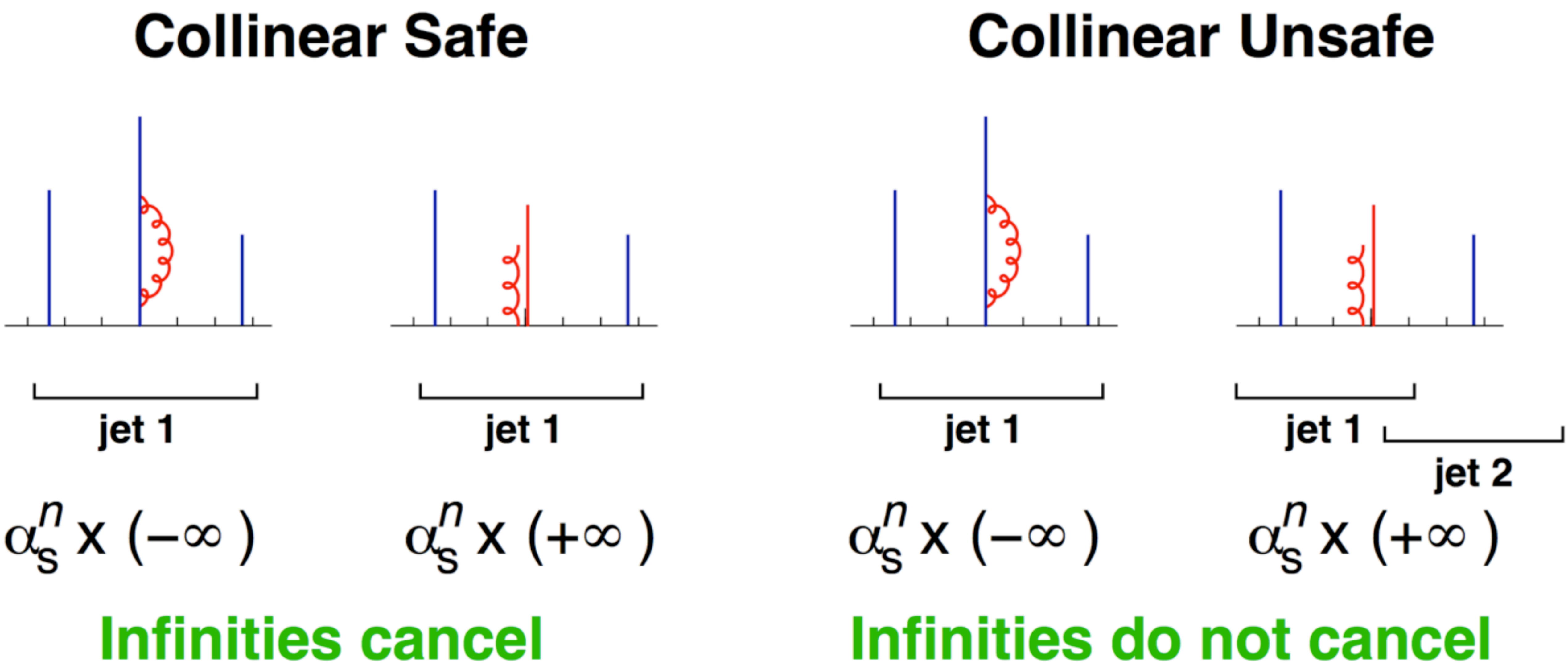


Recap from last lecture



A spray of particles gets produced along the line of flight of the original parton, governed by the factor : $dS = \approx \frac{2\alpha_S\, C_{F/A}}{\pi} \frac{dE}{E} \frac{d\theta}{\theta}$.

Jet defs : which one is legal?



Perturbative calculations of jet observable will only be possible with collinear (and infrared) safe jet definitions

IRC safety

An observable is **infrared and collinear safe** if, in the limit of a **collinear splitting**, or the **emission of an infinitely soft** particle, the observable remains **unchanged:**

$$O(X; p_1, \ldots, p_n, p_{n+1} \to 0) \to O(X; p_1, \ldots, p_n)$$

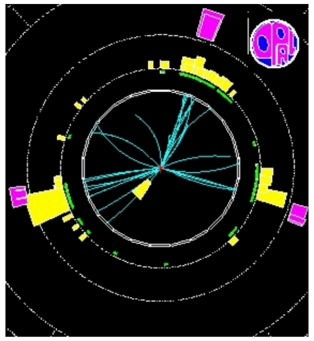
$$O(X; p_1, \ldots, p_n \parallel p_{n+1}) \to O(X; p_1, \ldots, p_n + p_{n+1})$$

This property ensures cancellation of **real** and **virtual** divergences in higher order calculations

If we wish to be able to calculate a jet rate in perturbative QCD the jet algorithm that we use must be IRC safe:
soft emissions and collinear splittings must not change the hard jets

Main approaches of jet clustering

1. Find regions where a lot of energy flows



or

2. Decide which particles are "close", aggregate them

In HEP these are usually called **cone** and **sequential recombination** algorithms respectively

(in other fields they are often called partitional-type clustering and agglomerative hierarchical clustering)

Several important properties that should be met by a jet definition are:

1. Simple to implement in an experimental analysis;
2. Simple to implement in the theoretical calculation;
3. Defined at any order of perturbation theory;
4. Yields finite cross sections at any order of perturbation theory;
5. Yields a cross section that is relatively insensitive to hadronisation.

Two main classes of algorithms

- ## Sequential recombination algorithms

Bottom-up approach: combine particles starting from **closest ones**

How? Choose a **distance measure**, iterate recombination until few objects left, call them jets

Works because of mapping closeness ⇔ QCD divergence

Examples: Jade, k_t, Cambridge/Aachen, anti-k_t, …..

Usually trivially made IRC safe, but their algorithmic complexity scales like N^3

- ## Cone algorithms

Top-down approach: find coarse regions of energy flow.

How? Find **stable cones** (i.e. their axis coincides with sum of momenta of particles in it)

Works because QCD only modifies energy flow on small scales

Examples: JetClu, MidPoint, ATLAS cone, CMS cone, SISCone…...

Can be programmed to be fairly fast, at the price of being complex and IRC unsafe

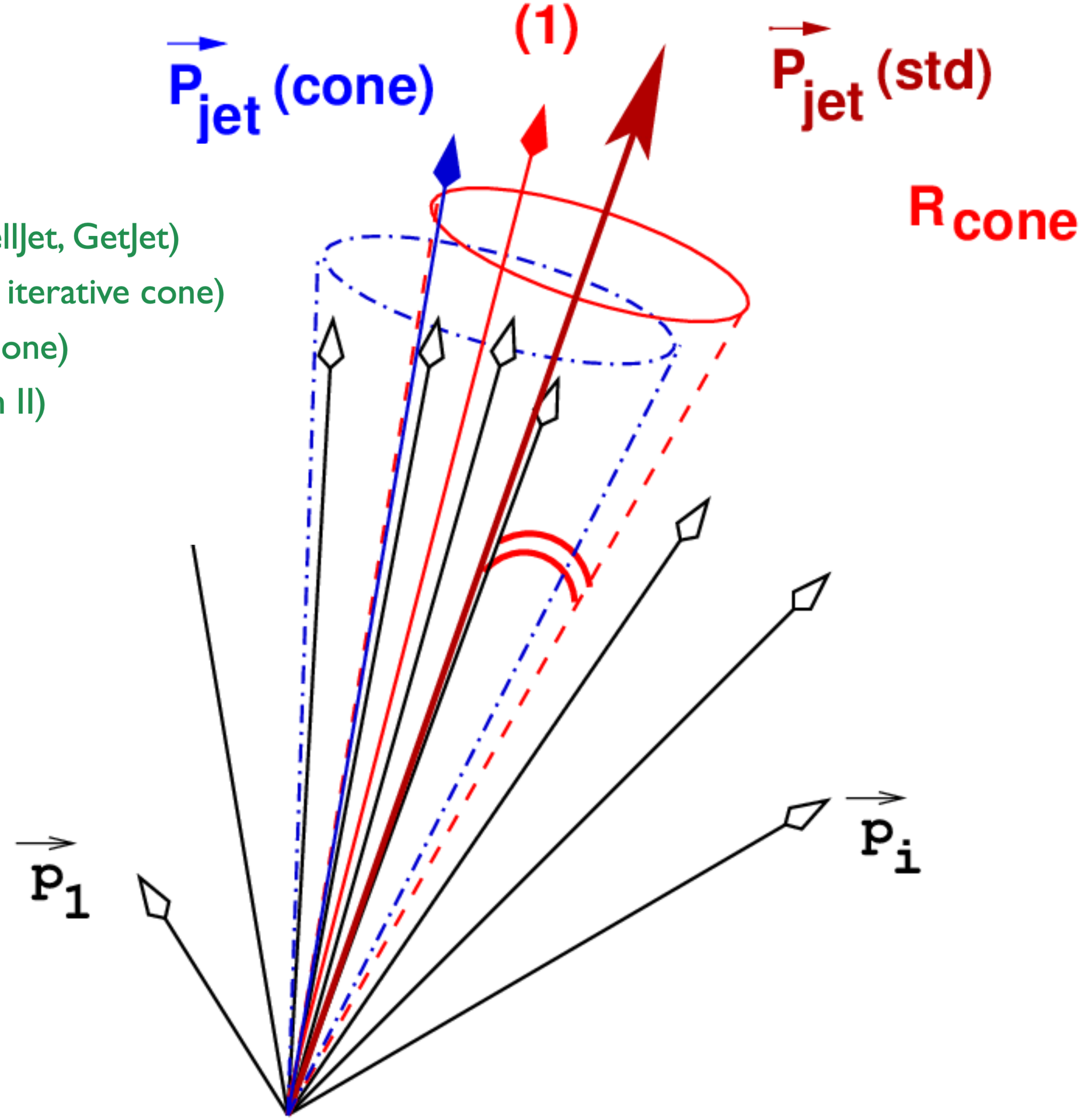
A word or two about cone algorithms

In partitional-type algorithms (i.e. cones), one wishes to find the **stable configurations**:
axis of cones coincides with sum of 4-momenta of the particles it contains

The 'safe' way of doing so is to test **all possible combinations** of N objects

The main sub-categories of cone algorithms are:

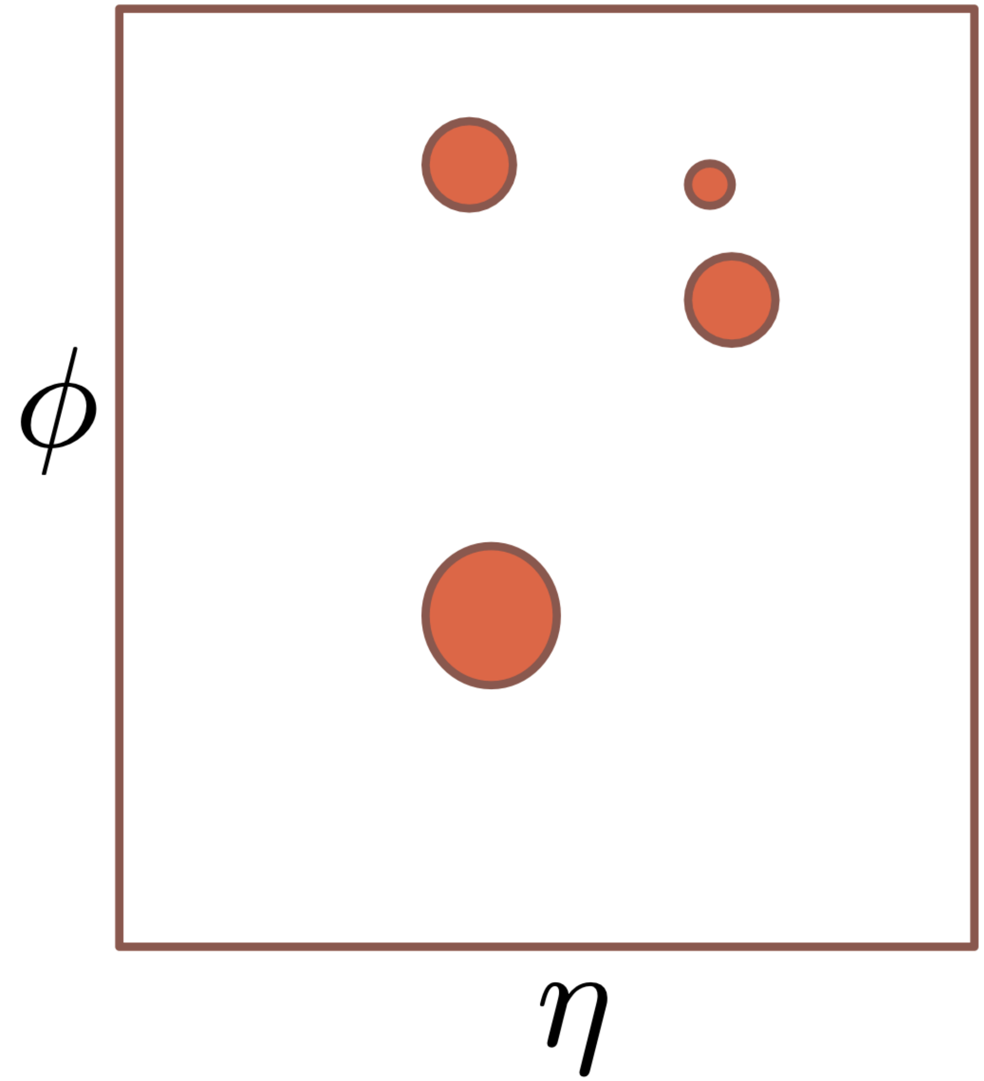
- **Fixed** cone with **progressive removal** (FC-PR) (PyJet, CellJet, GetJet)
- **Iterative** cone with **progressive removal** (IC-PR) (CMS iterative cone)
- **Iterative** cone with **split-merge** (IC-SM) (JetClu, ATLAS cone)
- **IC-SM** with **mid-points** (IC_{mp}-SM) (CDF MidPoint, D0 Run II)
- **IC_{mp}** with **split-drop** (IC_{mp}-SD) (PxCone)
- **Seedless** cone with **split-merge** (SC-SM) (SISCone)



Recombination algorithm : basic intuitions

Start with a set of 4-vectors p^i

Represent the particles in η , ϕ plane



$$\eta = -\frac{1}{2} ln\left(tan \ \frac{\theta}{2}\right)$$

$$\phi = arctan \ \left(\frac{p_y}{p_x}\right)$$

Recombination algorithm : basic intuitions

Start with a set of 4-vectors p^i

Represent the particles in η , ϕ plane

$$\eta = -\frac{1}{2} ln\left(tan \ \frac{\theta}{2} \right)$$

$$\phi = arctan \ \left(\frac{p_y}{p_x} \right)$$

Recombination algorithm : basic intuitions

Start with a set of 4-vectors p^i

Represent the particles in η , ϕ plane

Calculate the pairwise distance :

$$d_{ij} = \sqrt{(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2}$$

$$\boxed{\begin{aligned} \eta &= -\frac{1}{2} ln\left(tan\ \frac{\theta}{2}\right) \\ \phi &= arctan\ \left(\frac{p_y}{p_x}\right) \end{aligned}}$$

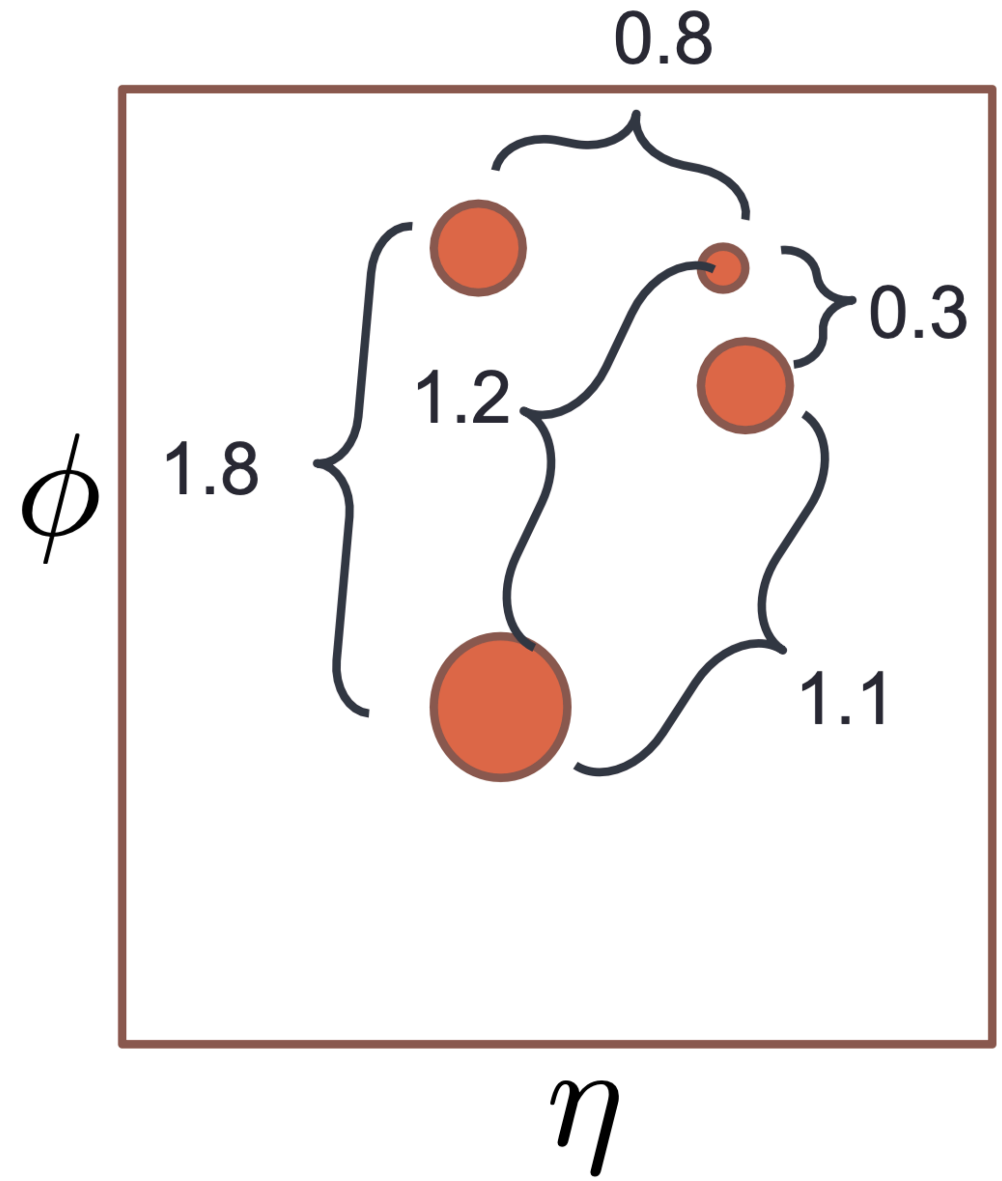
Recombination algorithm : basic intuitions

Start with a set of 4-vectors p^i

Represent the particles in η , ϕ plane

Calculate the pairwise distance :

$$d_{ij} = \sqrt{(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2}$$



$$\eta = -\frac{1}{2} ln\left(tan\ \frac{\theta}{2}\right)$$

$$\phi = arctan\ \left(\frac{p_y}{p_x}\right)$$

Recombination algorithm : basic intuitions

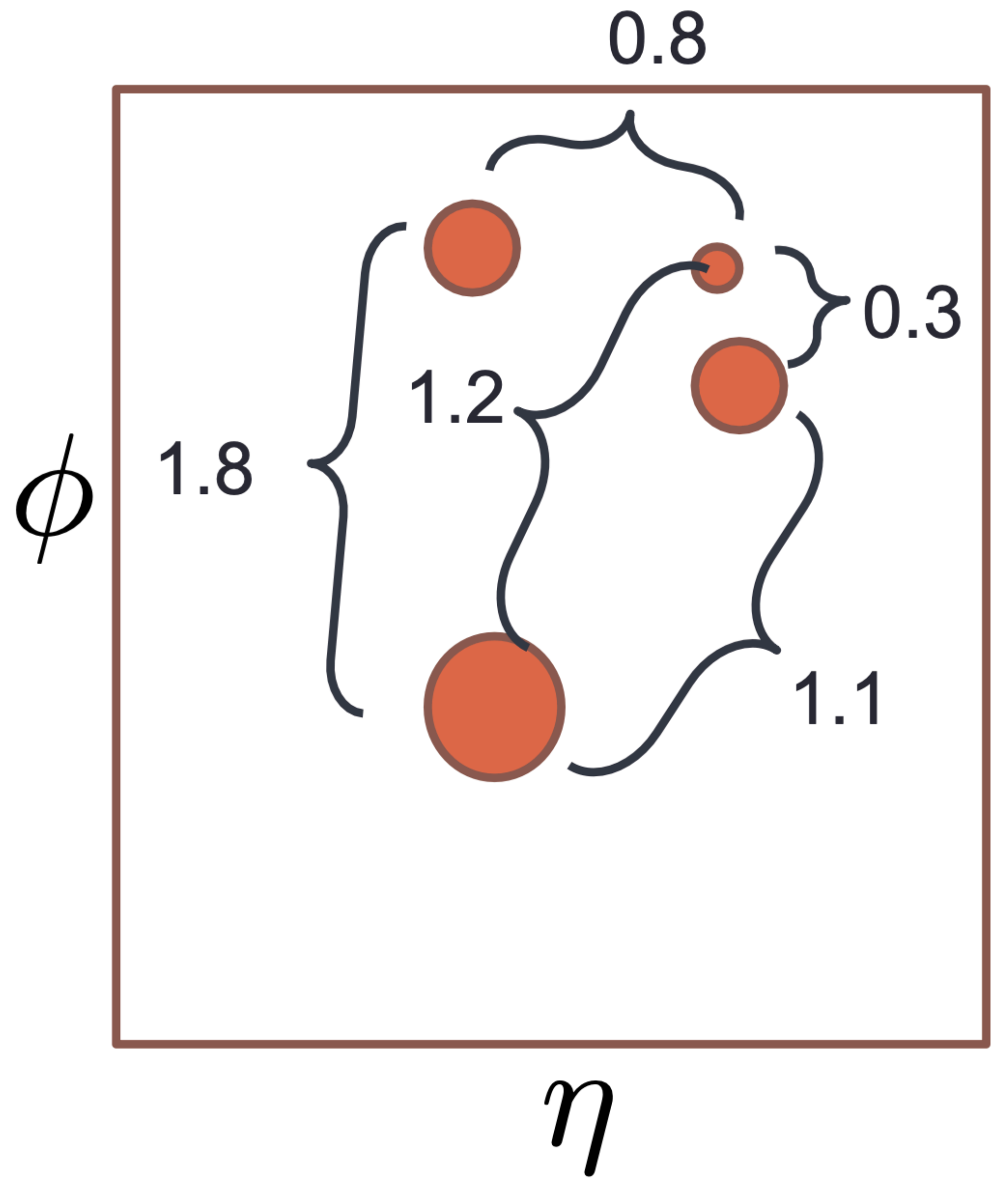
Start with a set of 4-vectors p^i

Represent the particles in η , ϕ plane

Calculate the pairwise distance :

$$d_{ij} = \sqrt{(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2}$$

Merge the two closest particle.
i.e. replace the pair by a combined representation.



$$\eta = -\frac{1}{2} ln\left(tan\ \frac{\theta}{2}\right)$$

$$\phi = arctan\ \left(\frac{p_y}{p_x}\right)$$

Recombination algorithm : basic intuitions

Start with a set of 4-vectors p^i

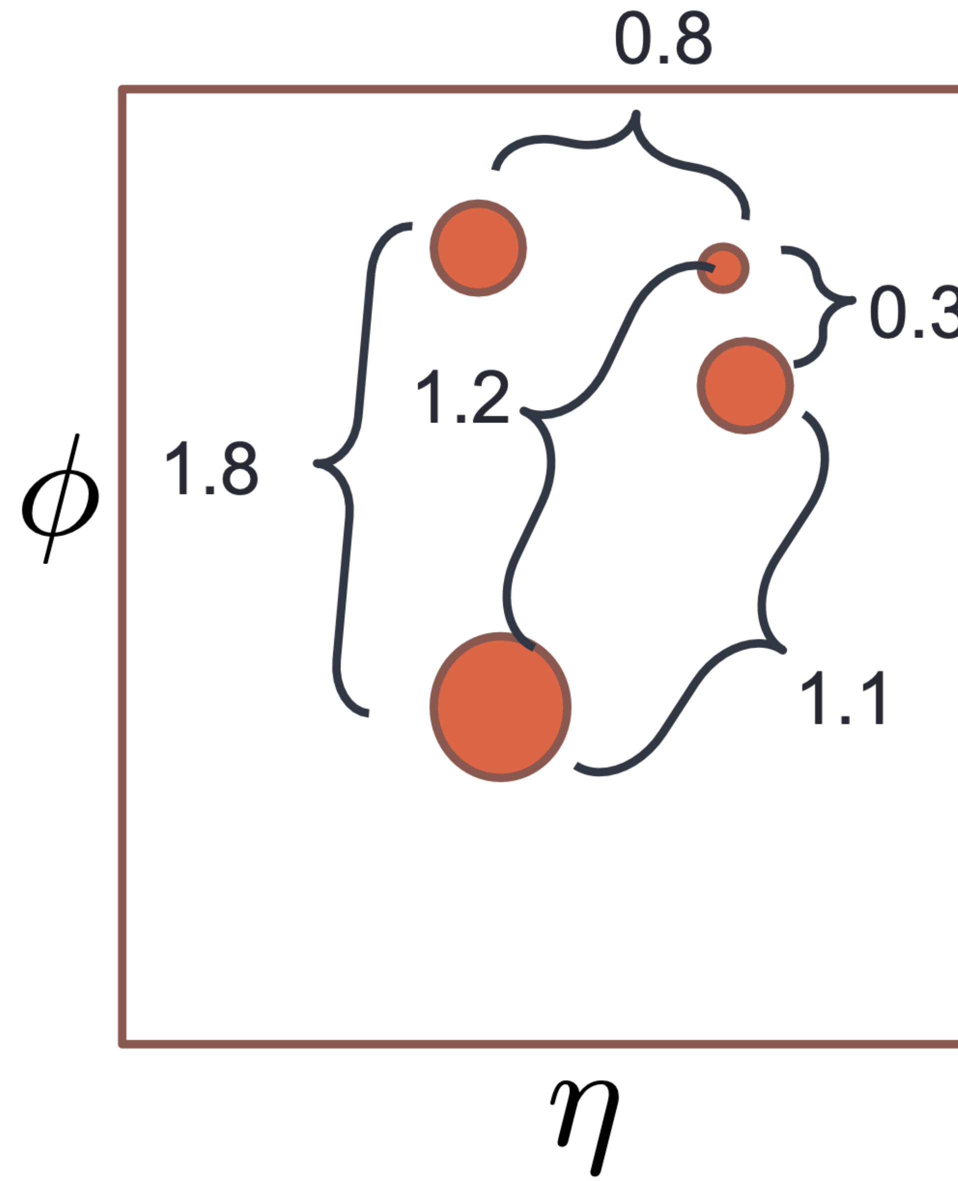
Represent the particles in η , ϕ plane

Calculate the pairwise distance :

$$d_{ij} = \sqrt{(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2}$$

Merge the two closest particle.
i.e. replace the pair by a combined representation.

Continue until no two particles are closer than R



$$\eta = -\frac{1}{2} ln\left(tan\ \frac{\theta}{2}\right)$$

$$\phi = arctan\ \left(\frac{p_y}{p_x}\right)$$

e^+e^- k_T algorithm

Durham: $$y_{ij} = \frac{2\min(E_i^2, E_j^2)\left(1-\cos\theta_{ij}\right)}{Q^2},$$

Geneva: $$y_{ij} = \frac{8}{9} \cdot \frac{2E_iE_j\left(1-\cos\theta_{ij}\right)}{\left(E_i+E_j\right)^2},$$

Jade-E0: $$y_{ij} = \frac{2E_iE_j\left(1-\cos\theta_{ij}\right)}{Q^2},$$

https://arxiv.org/pdf/1011.6247

e^+e^- k_T algorithm

Durham: $$y_{ij} = \frac{2\min(E_i^2, E_j^2)\,(1-\cos\theta_{ij})}{Q^2},$$

Geneva: $$y_{ij} = \frac{8}{9} \cdot \frac{2E_iE_j\,(1-\cos\theta_{ij})}{(E_i+E_j)^2},$$

Jade-E0: $$y_{ij} = \frac{2E_iE_j\,(1-\cos\theta_{ij})}{Q^2},$$

https://arxiv.org/pdf/1011.6247

1. Define a resolution parameter y_{cut}
2. For every pair (p_k, p_l) of final-state particles compute the corresponding resolution variable y_{kl}.
3. If y_{ij} is the smallest value of y_{kl} computed above and $y_{ij} < y_{cut}$ then combine (p_i, p_j) into a single jet ('pseudo-particle') with momentum p_{ij} according to a recombination prescription.
4. Repeat until all pairs of objects (particles and/or pseudo-particles) have $y_{kl} > y_{cut}$.
5. Objects with $E \geq E_{min}$ are called jets.

Generalized kT (recombination) algorithm

Generalized kT (recombination) algorithm

1. Take the particles in the event as our initial list of objects.
2. From the list of objects, build two sets of distances: an *inter-particle distance*

$$d_{ij} = \min(p_{t,i}^{2p}, p_{t,j}^{2p})\Delta R_{ij}^2, \tag{3.1}$$

Generalized kT (recombination) algorithm

1. Take the particles in the event as our initial list of objects.
2. From the list of objects, build two sets of distances: an *inter-particle distance*

$$d_{ij} = \min(p_{t,i}^{2p}, p_{t,j}^{2p})\Delta R_{ij}^2, \tag{3.1}$$

where p is a free parameter and ΔR_{ij} is the geometric distance in the rapidity-azimuthal angle plane (Eq. (2.34)), and a *beam distance*

$$d_{iB} = p_{t,i}^{2p} R^2, \tag{3.2}$$

with R a free parameter usually called the *jet radius*.

Generalized kT (recombination) algorithm

1. Take the particles in the event as our initial list of objects.
2. From the list of objects, build two sets of distances: an *inter-particle distance*

$$d_{ij} = \min(p_{t,i}^{2p}, p_{t,j}^{2p})\Delta R_{ij}^2, \tag{3.1}$$

where p is a free parameter and ΔR_{ij} is the geometric distance in the rapidity-azimuthal angle plane (Eq. (2.34)), and a *beam distance*

$$d_{iB} = p_{t,i}^{2p} R^2, \tag{3.2}$$

with R a free parameter usually called the *jet radius*.

3. Iteratively find the smallest distance among all the d_{ij} and d_{iB}
   - If the smallest distance is a d_{ij} then objects i and j are removed from the list and recombined into a new object k (using the recombination scheme) which is itself added to the list.
   - If the smallest is a d_{iB}, object i is called a *jet* and removed from the list.

Generalized kT (recombination) algorithm

1. Take the particles in the event as our initial list of objects.
2. From the list of objects, build two sets of distances: an *inter-particle distance*

$$d_{ij} = \min(p_{t,i}^{2p}, p_{t,j}^{2p})\Delta R_{ij}^2, \tag{3.1}$$

where p is a free parameter and ΔR_{ij} is the geometric distance in the rapidity-azimuthal angle plane (Eq. (2.34)), and a *beam distance*

$$d_{iB} = p_{t,i}^{2p} R^2, \tag{3.2}$$

with R a free parameter usually called the *jet radius*.

3. Iteratively find the smallest distance among all the d_{ij} and d_{iB}
   - If the smallest distance is a d_{ij} then objects i and j are removed from the list and recombined into a new object k (using the recombination scheme) which is itself added to the list.
   - If the smallest is a d_{iB}, object i is called a *jet* and removed from the list.

Go back to step-2

Generalized kT (recombination) algorithm

$$d_{ij} = \min(p_{t,i}^{2p}, p_{t,j}^{2p})\Delta R_{ij}^2$$

Generalized kT (recombination) algorithm

$$d_{ij} = \min(p_{t,i}^{2p}, p_{t,j}^{2p})\Delta R_{ij}^2$$

p = 1 k_t algorithm

S. Catani, Y. Dokshitzer, M. Seymour and B. Webber, Nucl. Phys. B406 (1993) 187
S.D. Ellis and D.E. Soper, Phys. Rev. D48 (1993) 3160

p = 0 Cambridge/Aachen algorithm

Y. Dokshitzer, G. Leder, S.Moretti and B. Webber, JHEP 08 (1997) 001
M.Wobisch and T.Wengler, hep-ph/9907280

p = -1 **anti-k_t algorithm**

MC, G. Salam and G. Soyez, arXiv:0802.1189

NB: in anti-kt pairs with a **hard** particle will cluster first: if no other hard particles are close by, the algorithm will give **perfect cones**

Generalized kT (recombination) algorithm

$$d_{ij} = \min(p_{t,i}^{2p}, p_{t,j}^{2p}) \Delta R_{ij}^2$$

Generalized kT (recombination) algorithm

$$d_{ij} = \min(p_{t,i}^{2p}, p_{t,j}^{2p})\Delta R_{ij}^2$$

p > 0

New **soft** particle ($p_t \to 0$) means that d → 0 ⇒ clustered first, no effect on jets

New **collinear** particle ($\Delta y^2 + \Delta \Phi^2 \to 0$) means that d → 0 ⇒ clustered first, no effect on jets

p = 0

New **soft** particle ($p_t \to 0$) can be new jet of zero momentum ⇒ no effect on hard jets

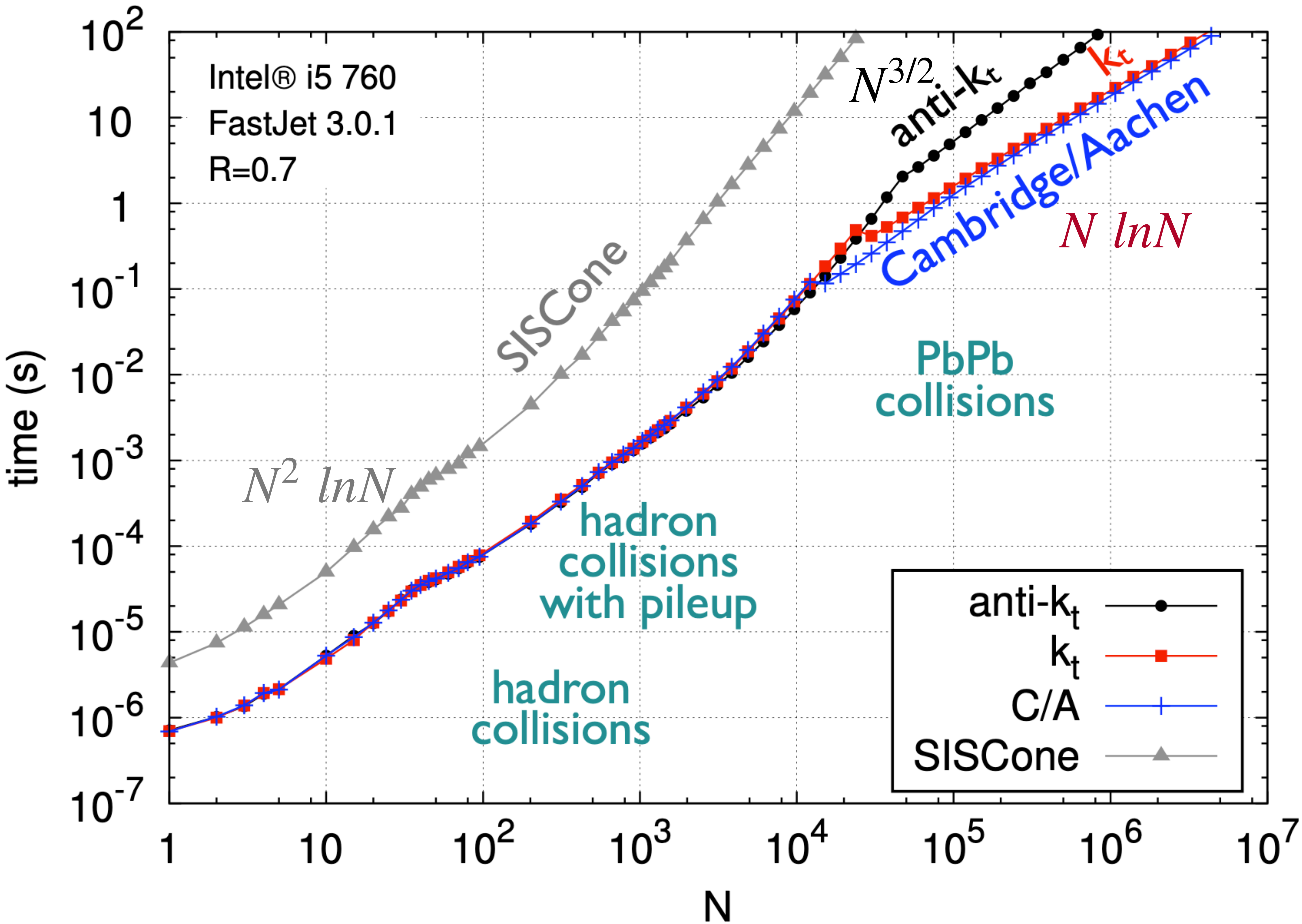
New **collinear** particle ($\Delta y^2 + \Delta \Phi^2 \to 0$) means that d → 0 ⇒ clustered first, no effect on jets

p < 0

New **soft** particle ($p_t \to 0$) means d → ∞ ⇒ clustered last or new zero-jet, no effect on hard jets

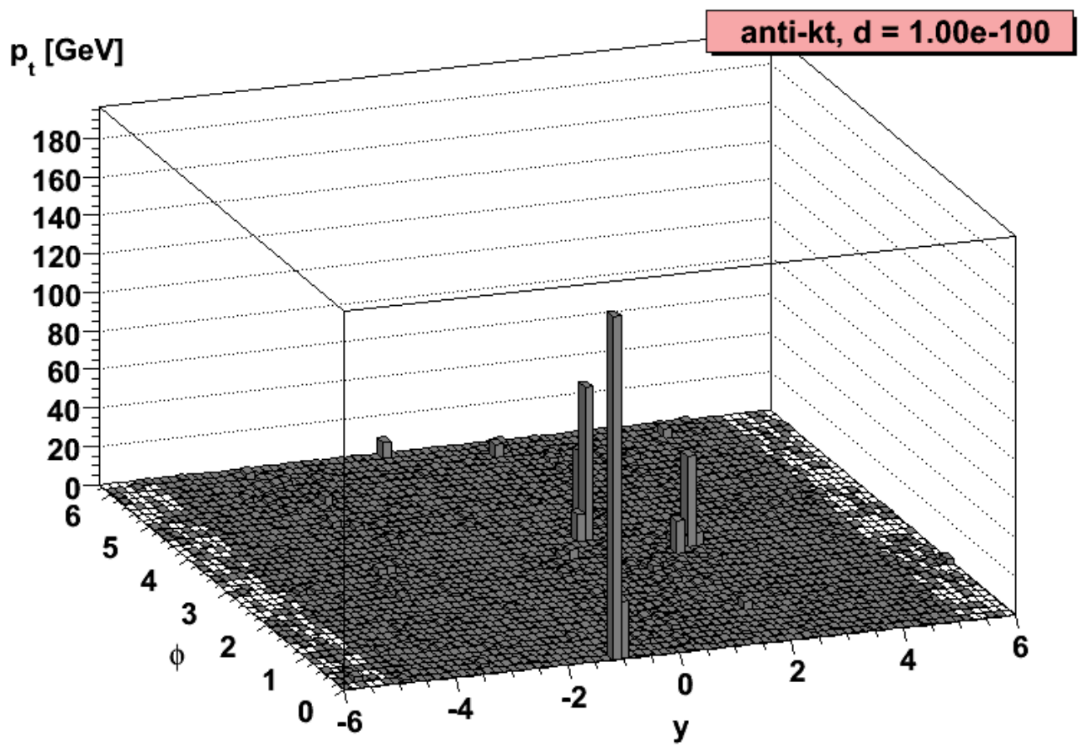
New **collinear** particle ($\Delta y^2 + \Delta \Phi^2 \to 0$) means that d → 0 ⇒ clustered first, no effect on jets

How about reconstruction time?

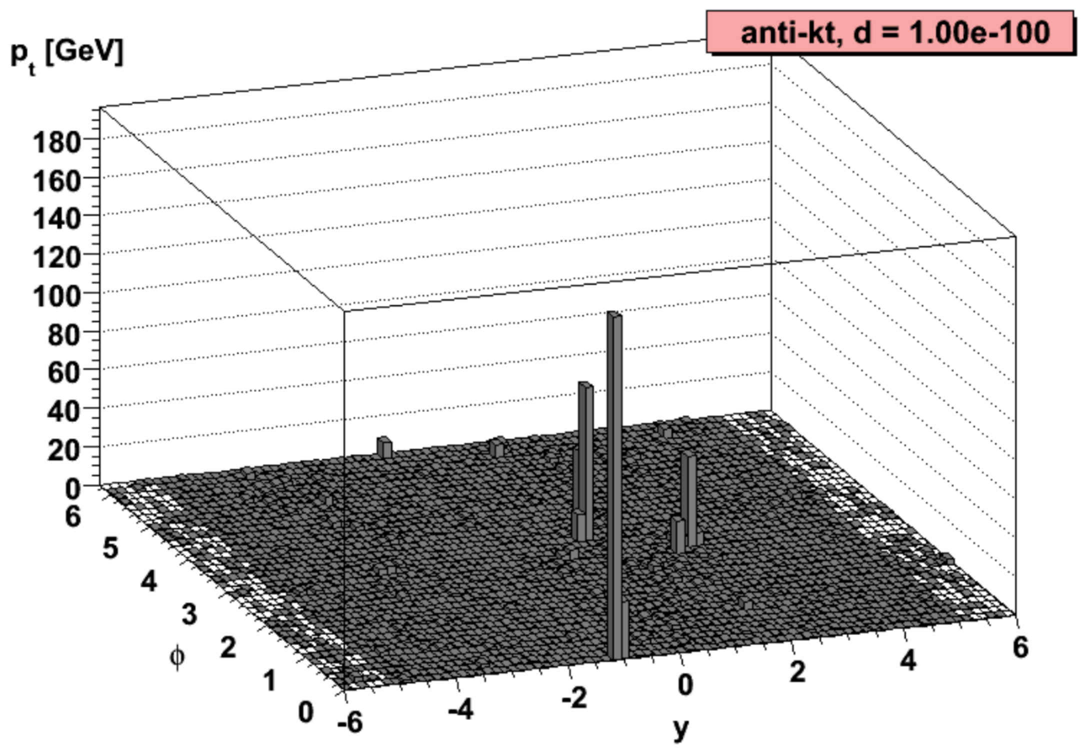


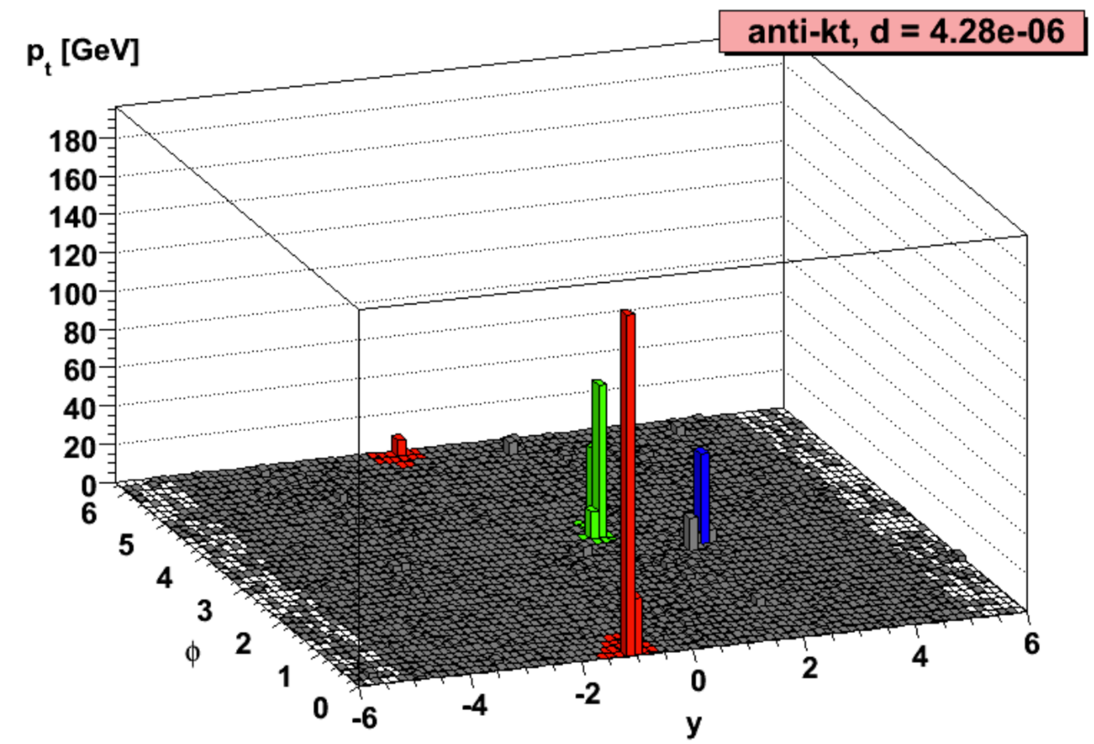
Let’s see how the clustering grows

Let’s see how the clustering grows

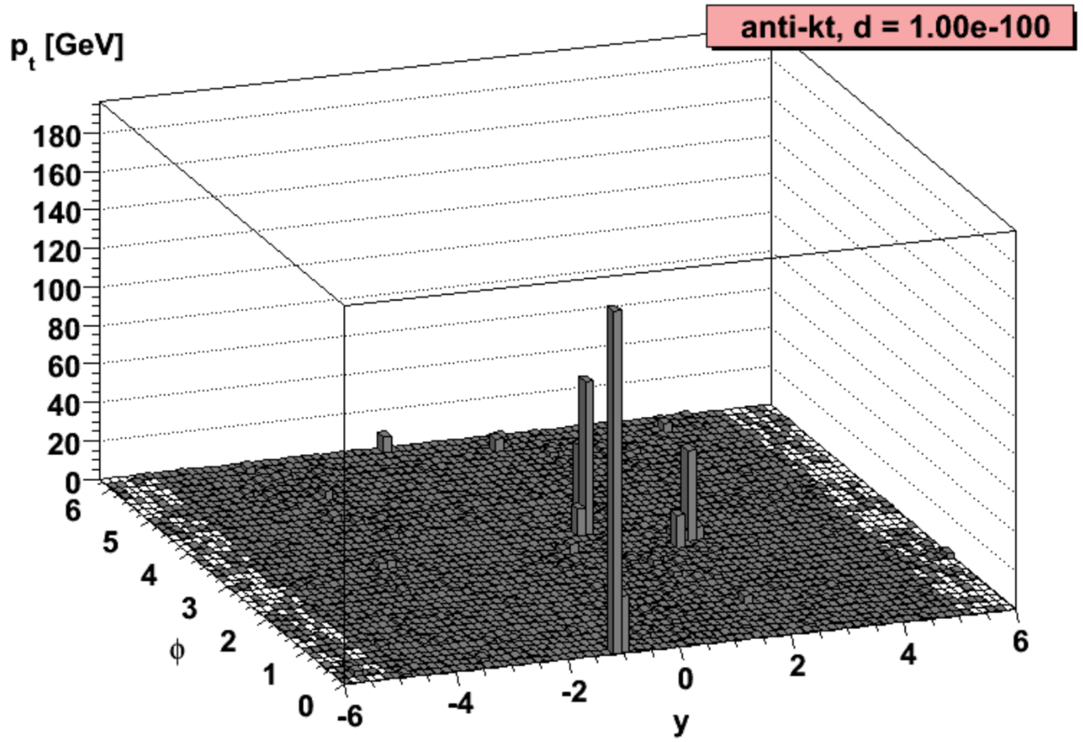


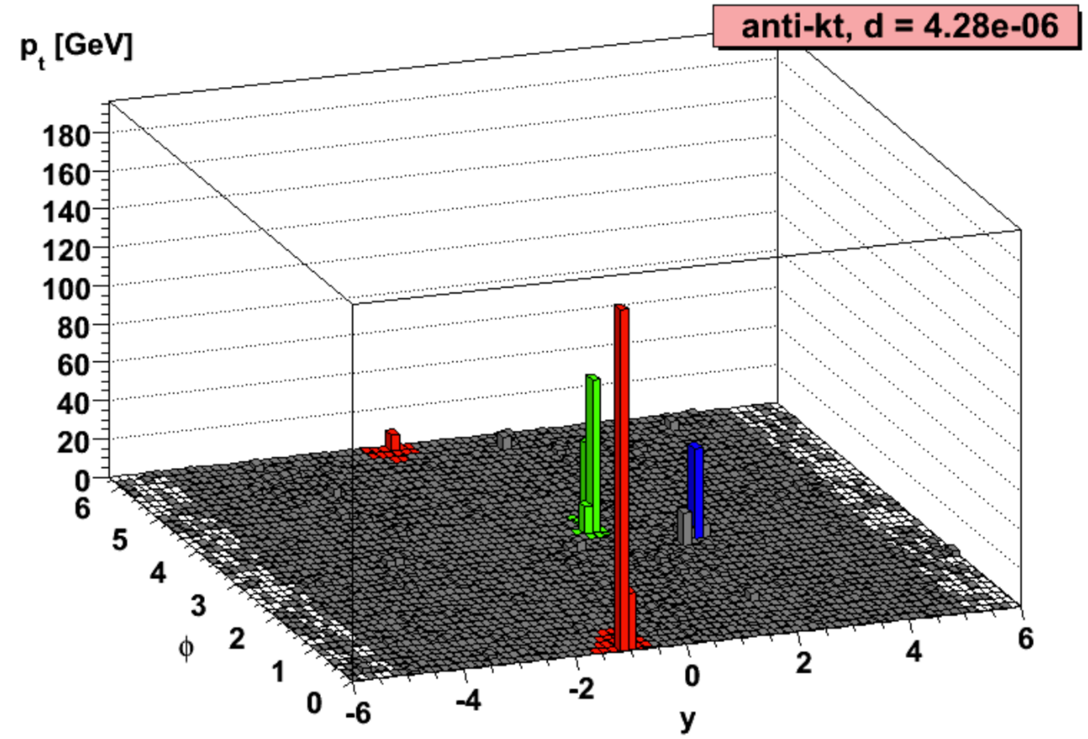
Let's see how the clustering grows

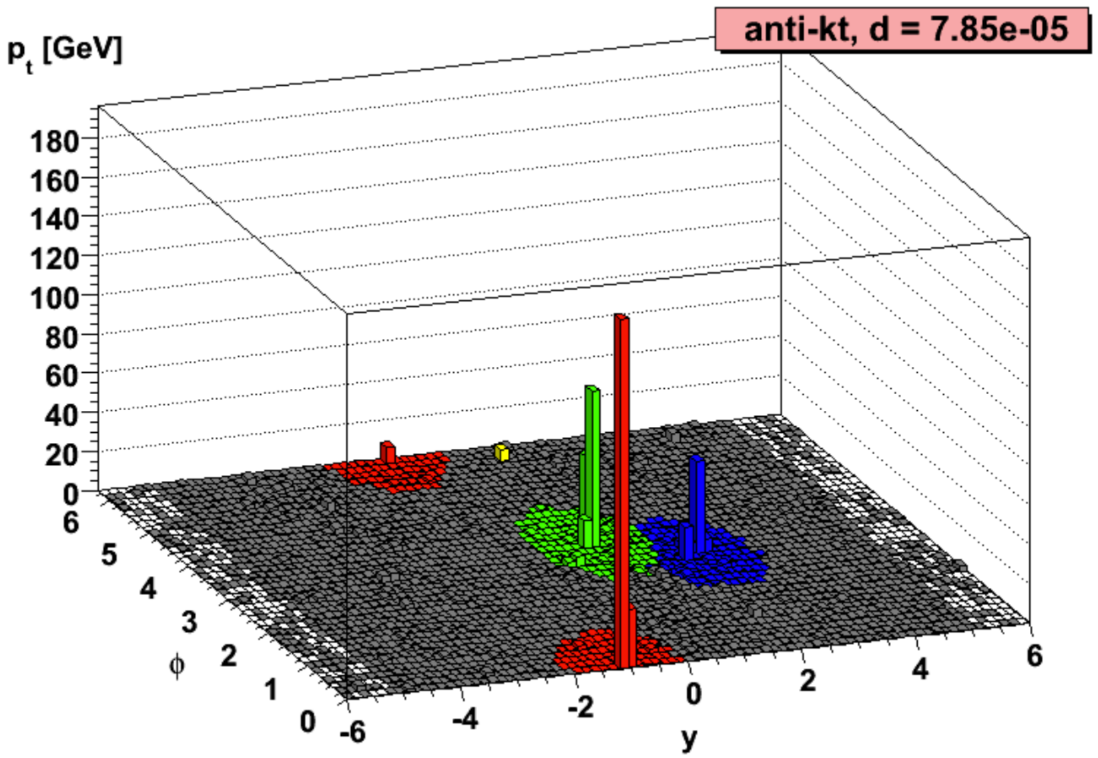




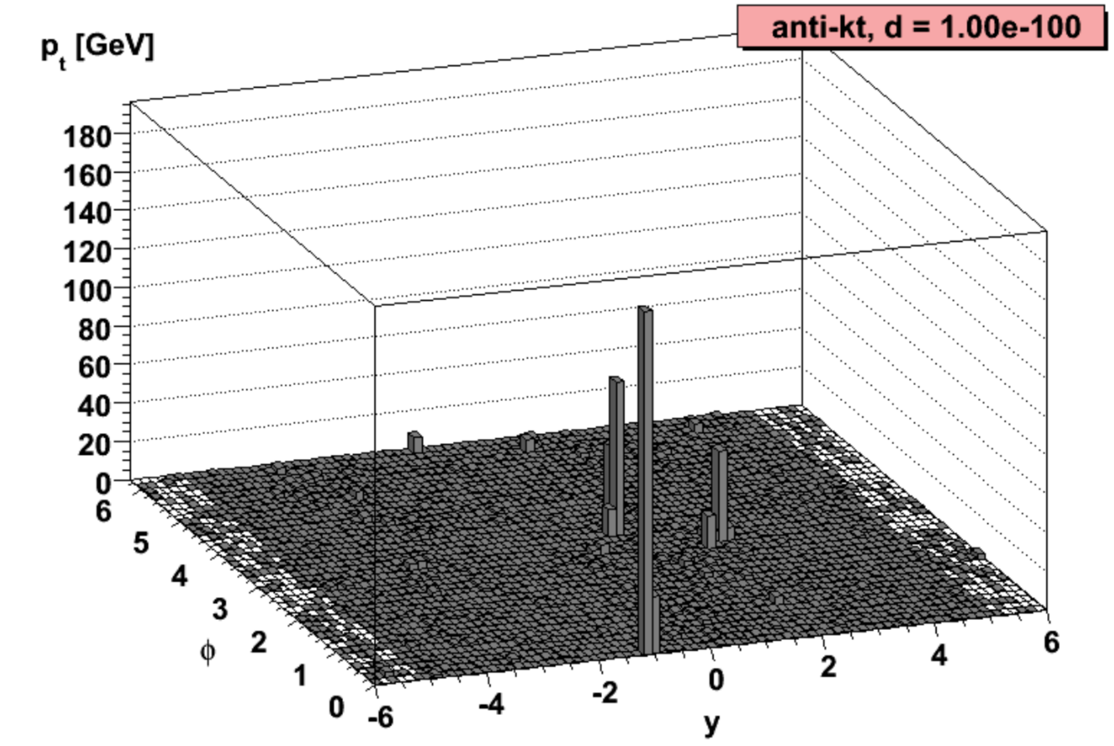
Let's see how the clustering grows

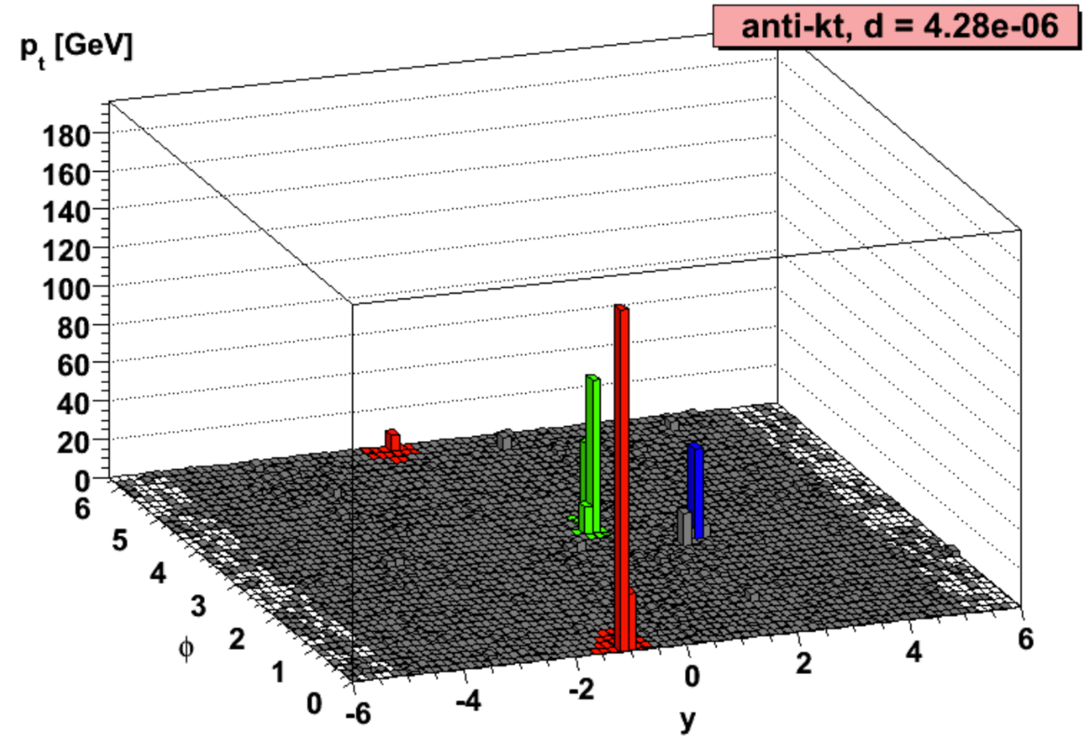


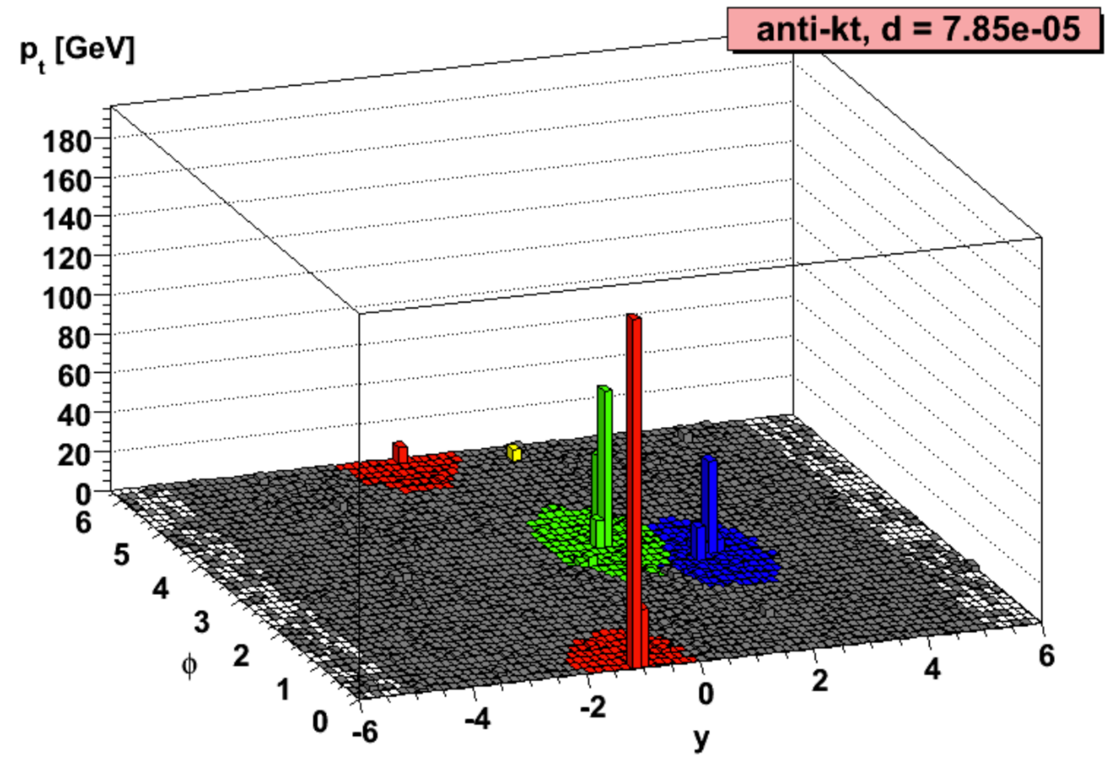


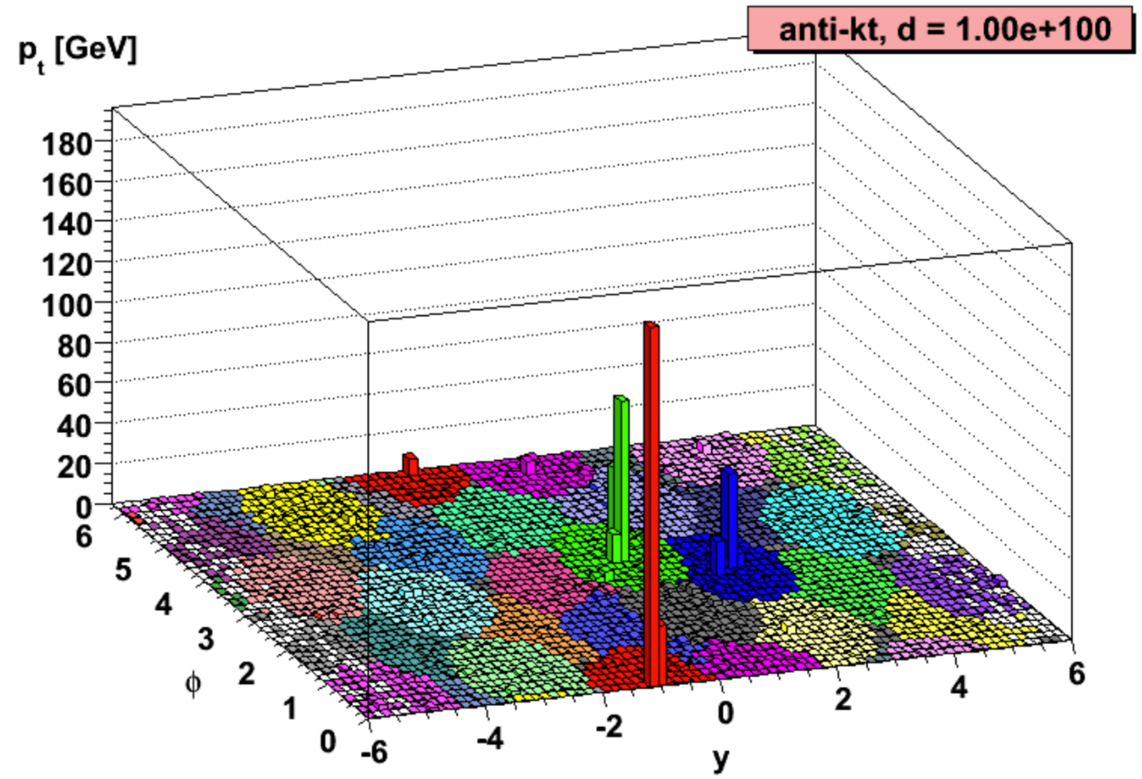


Let's see how the clustering grows

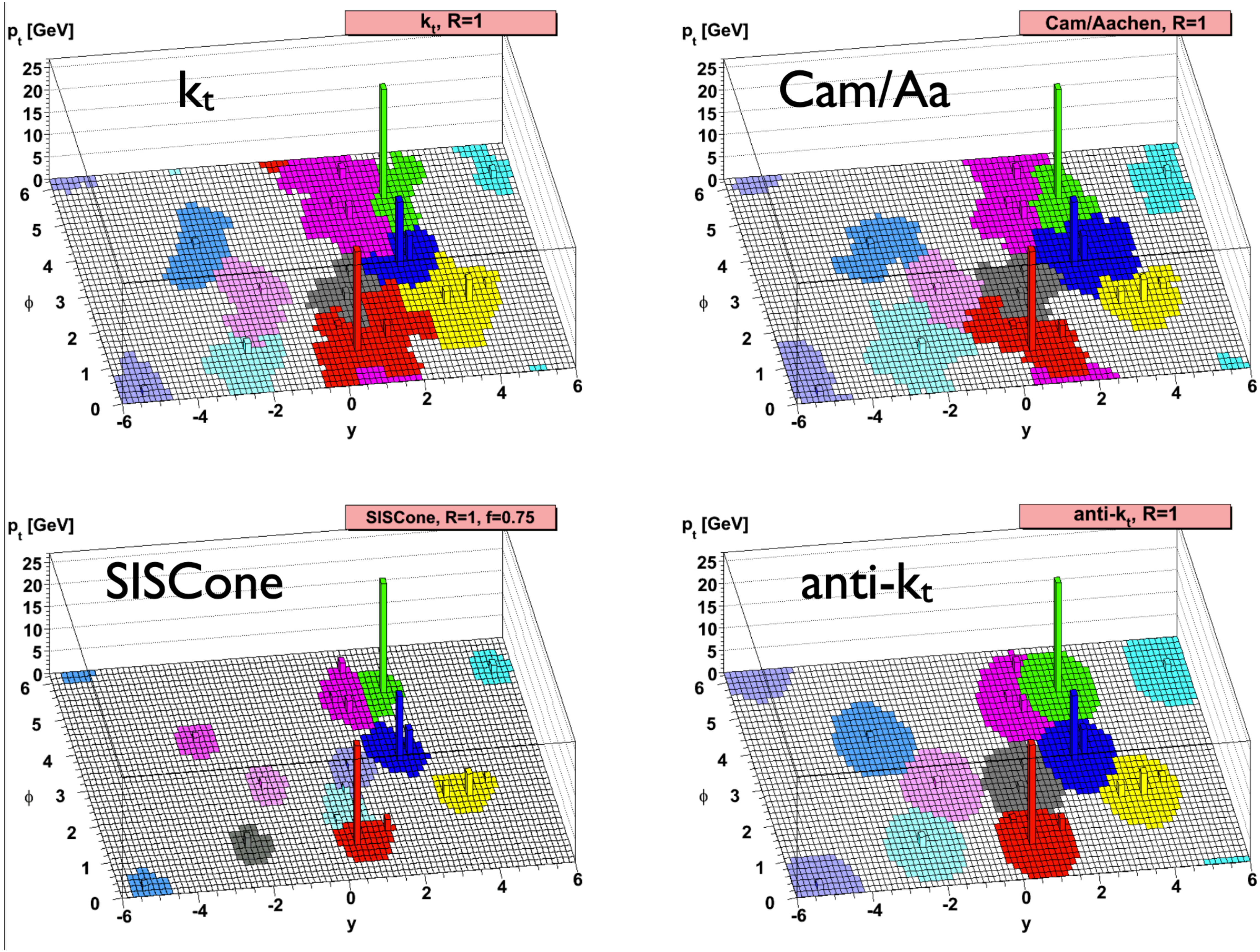








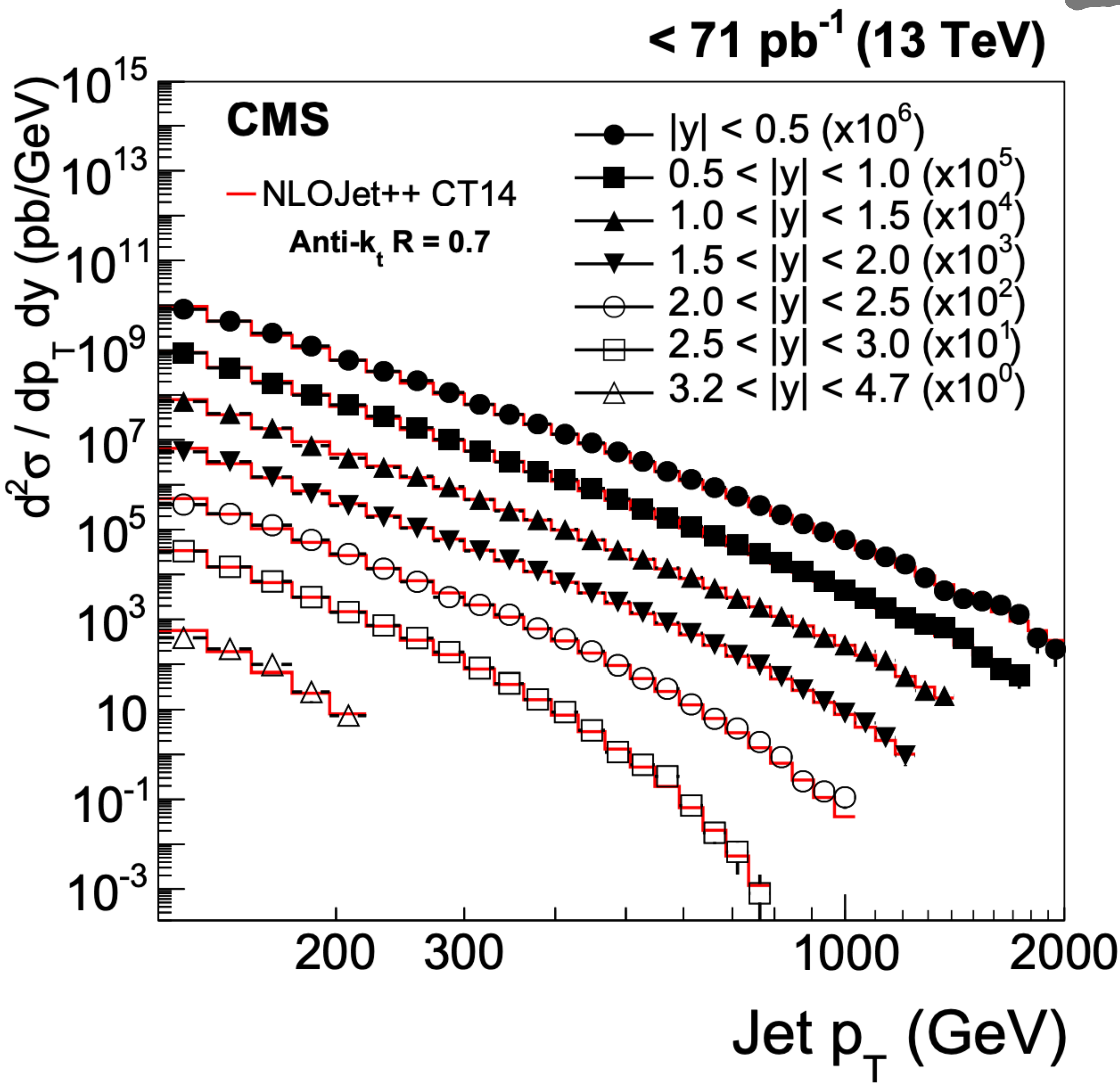
For a same set of particles, different algorithm

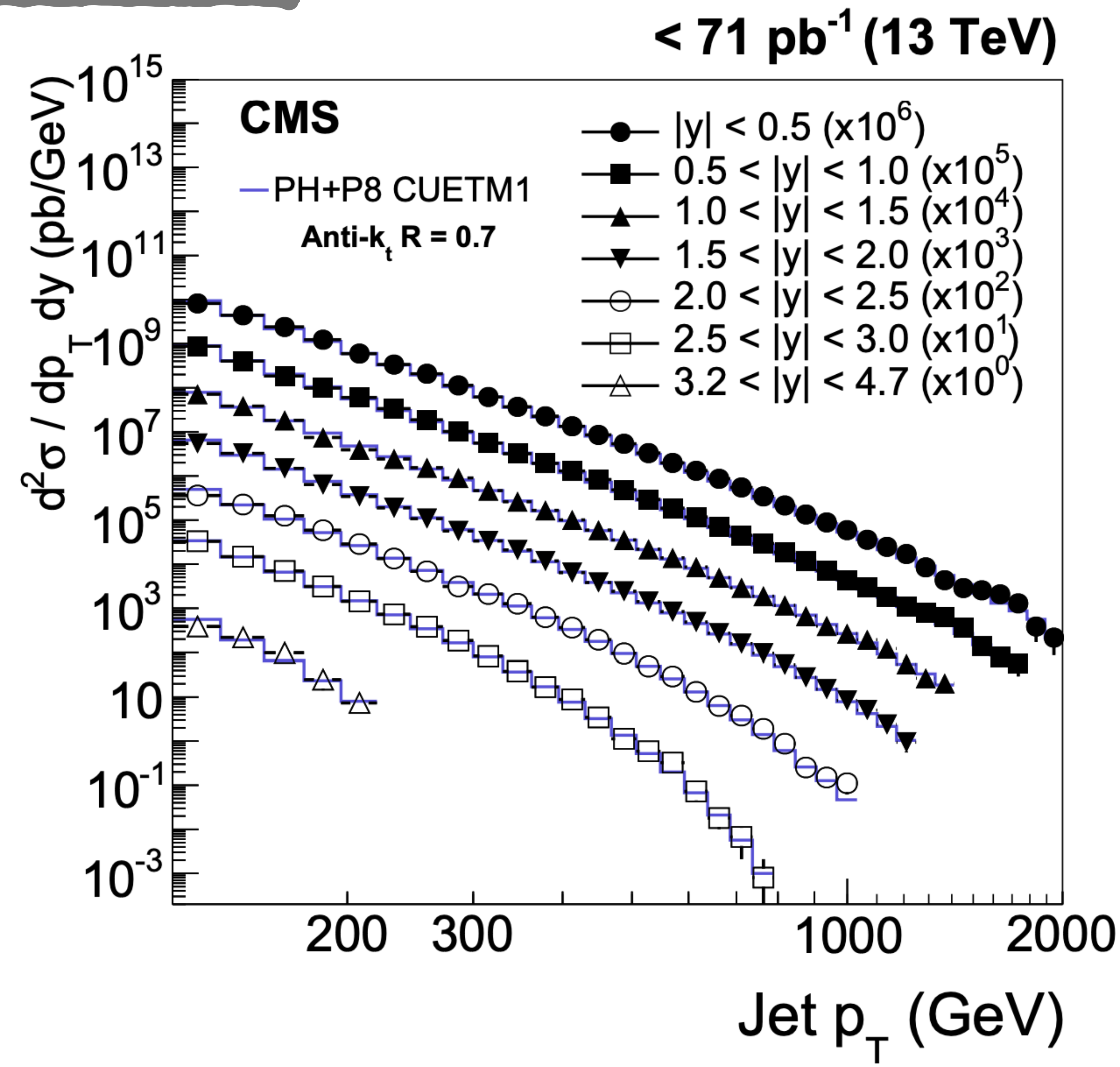


So what we can measure?

$$\boxed{\frac{d^2\sigma}{dp_T\,dy} = \frac{1}{\mathcal{L}}\frac{N_{\rm jets}^{\rm eff}}{\Delta p_T\,\Delta y}}$$

arXiv: 1605.04436

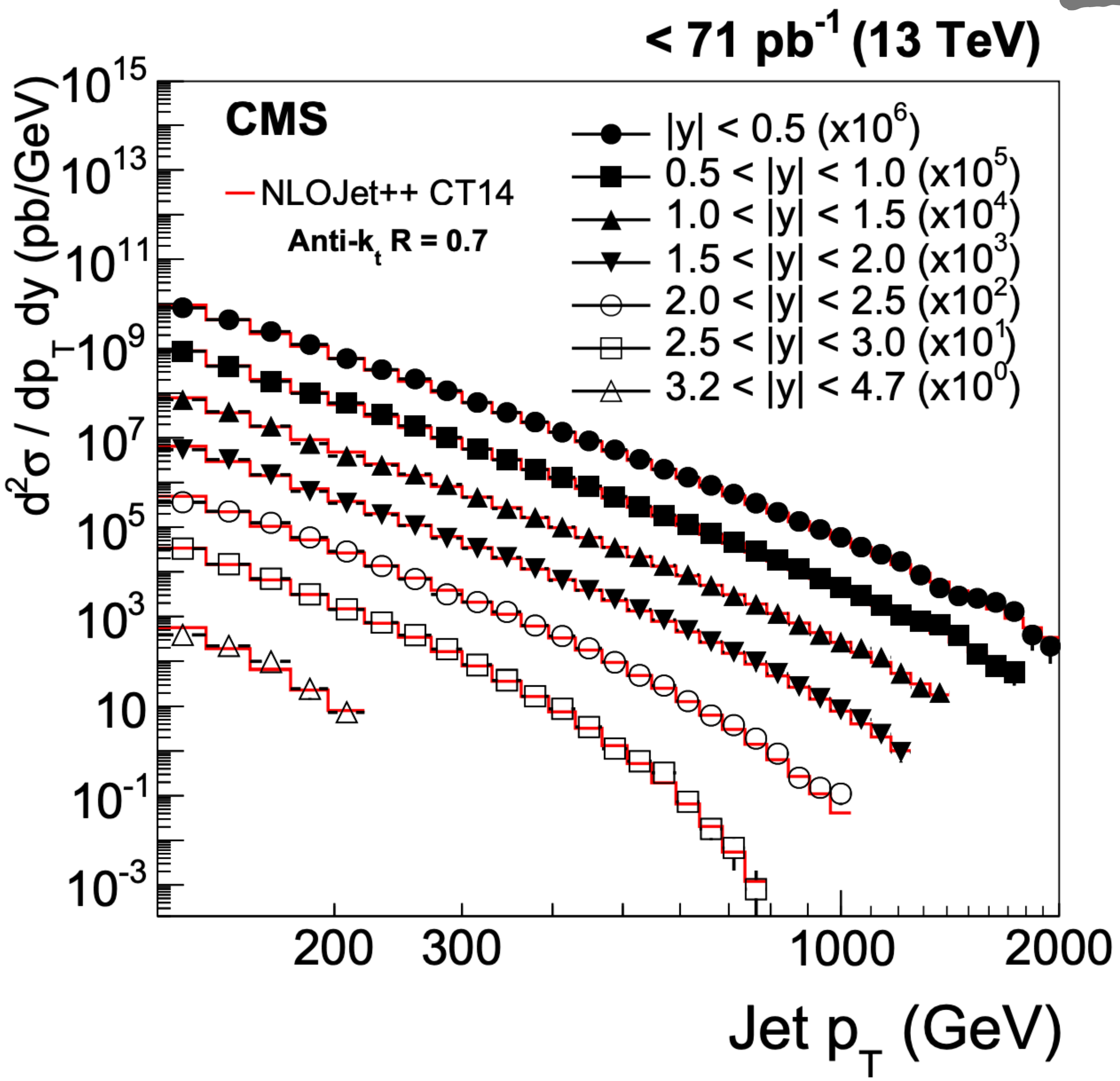


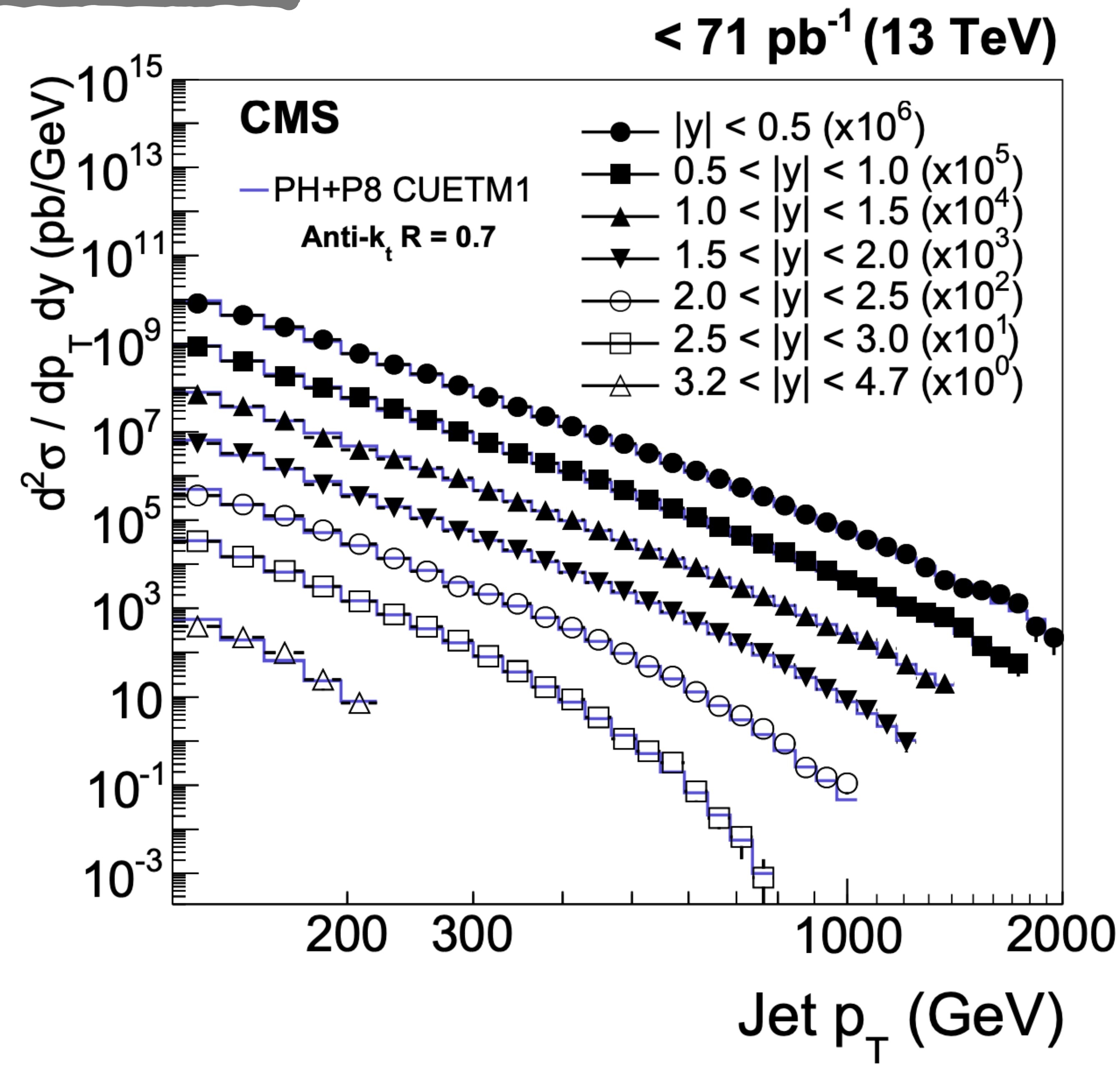


So what we can measure?

$$\frac{d^2\sigma}{dp_T\,dy} = \frac{1}{\mathcal{L}}\frac{N_{\text{jets}}^{\text{eff}}}{\Delta p_T\,\Delta y}$$

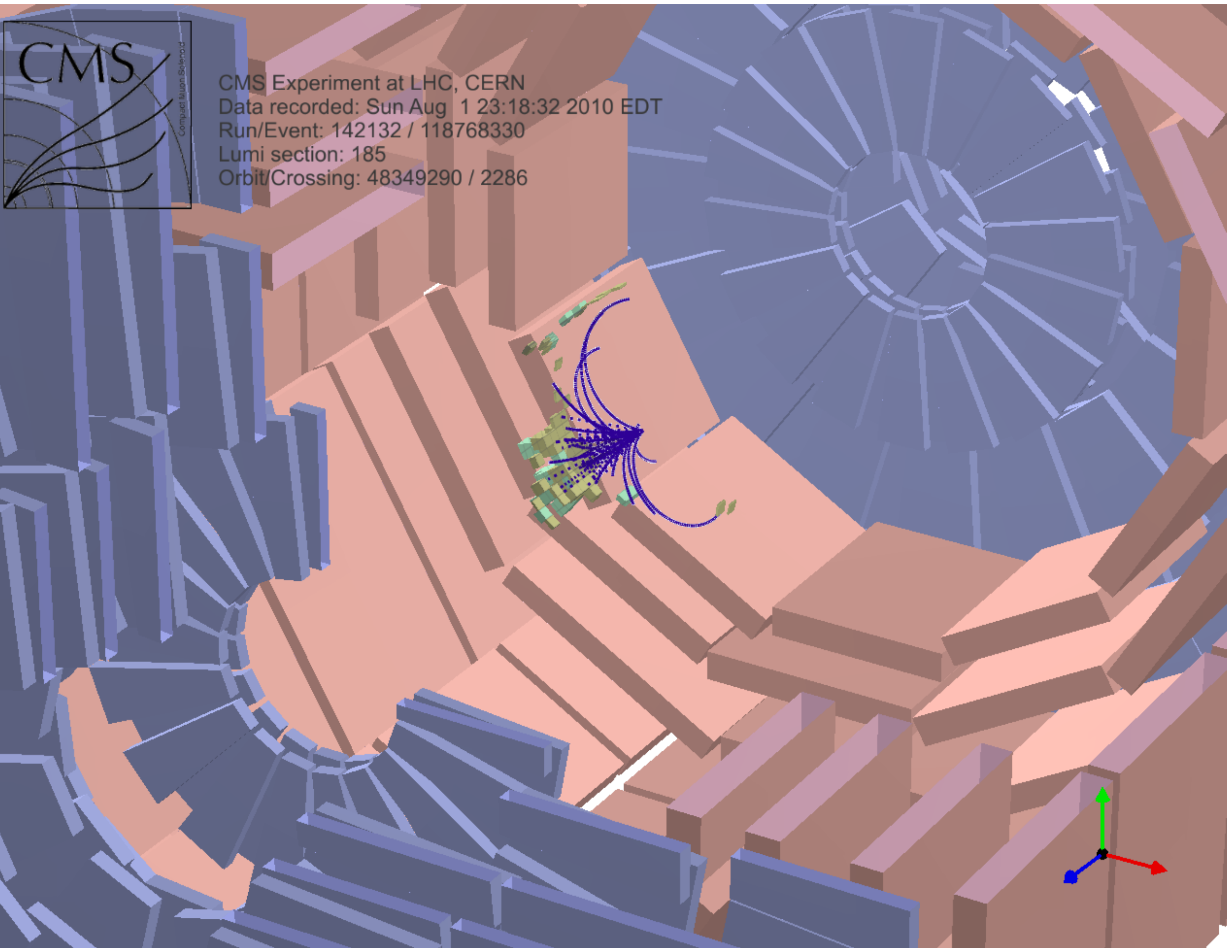
arXiv: 1605.04436

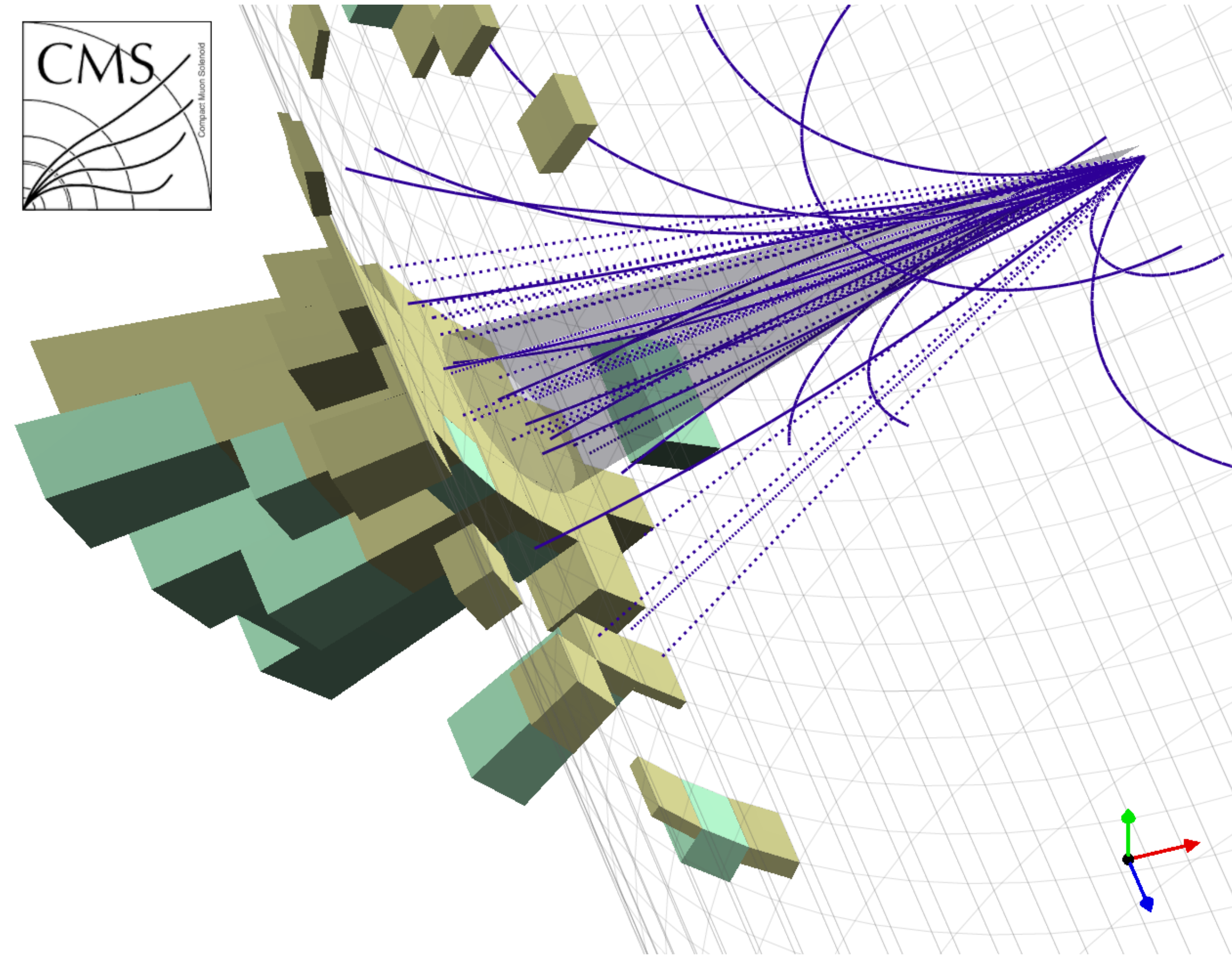


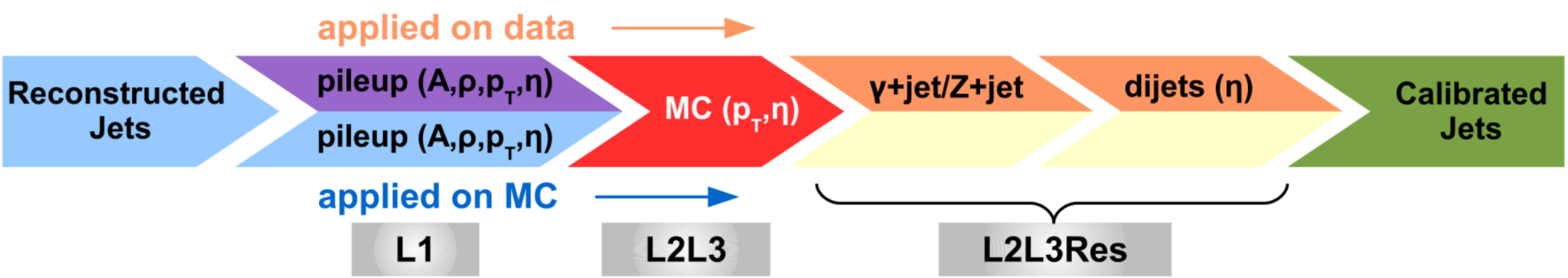


What does it to take to perform the simplest of measurements ?

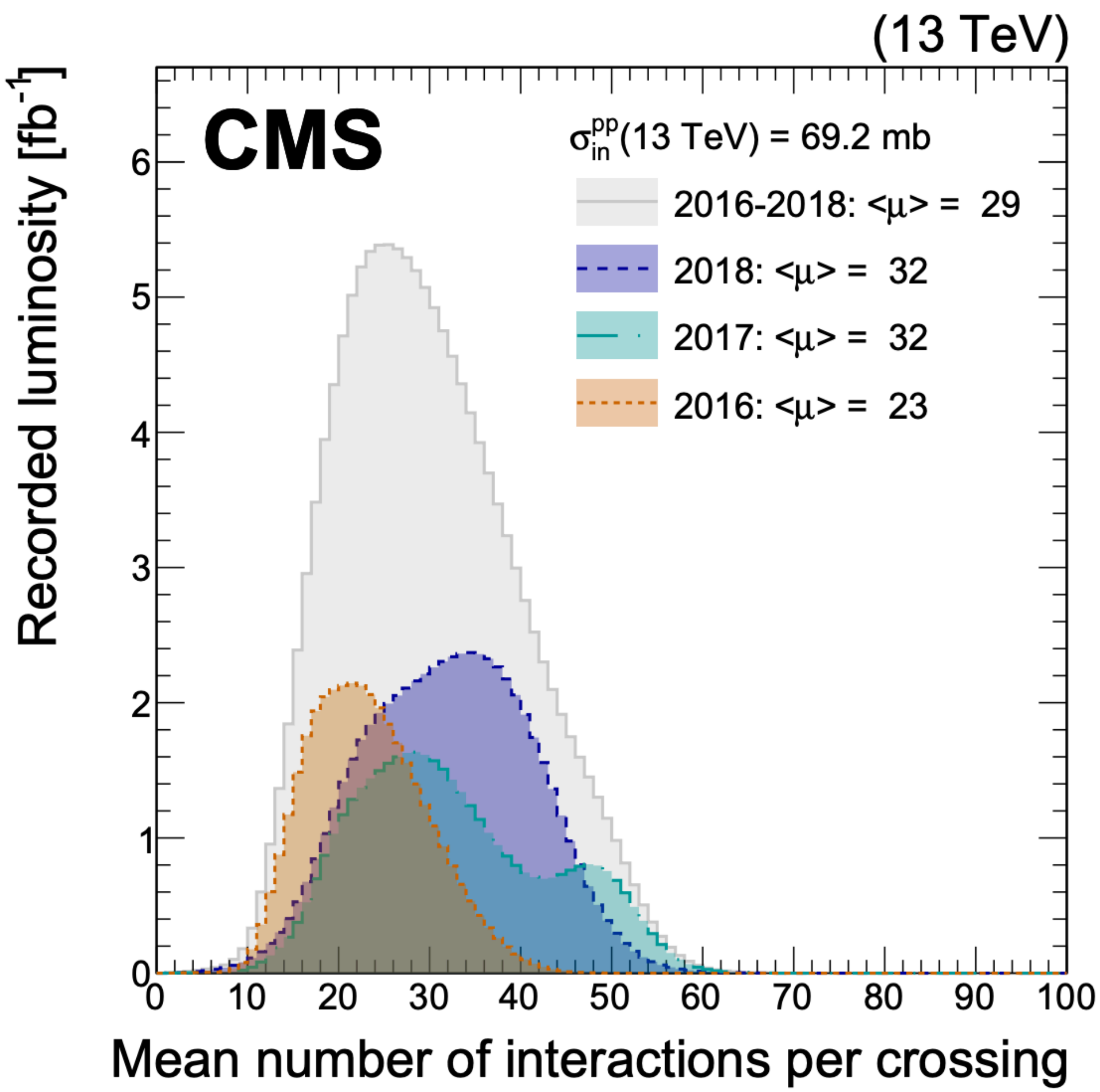
Where we start from

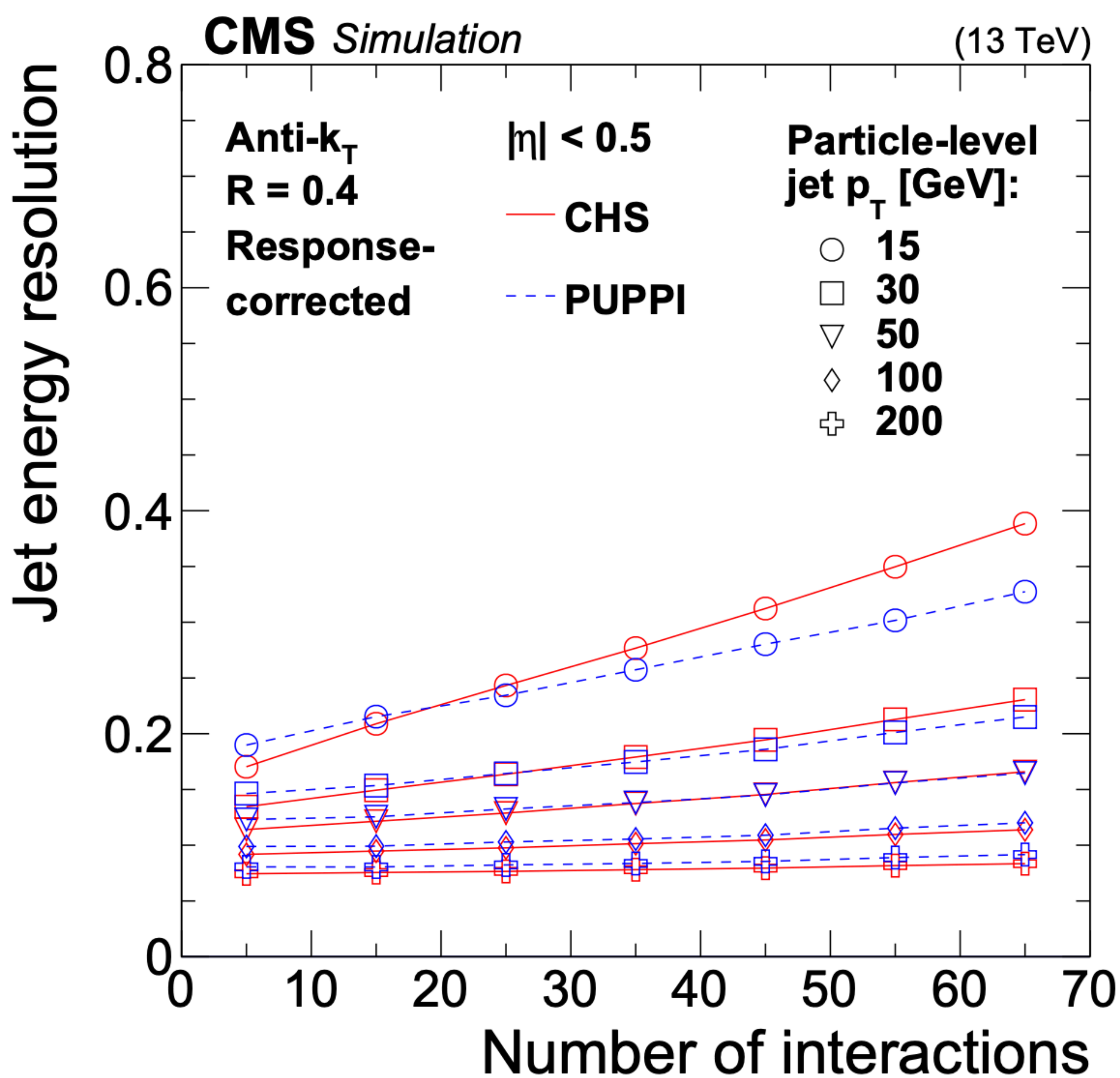






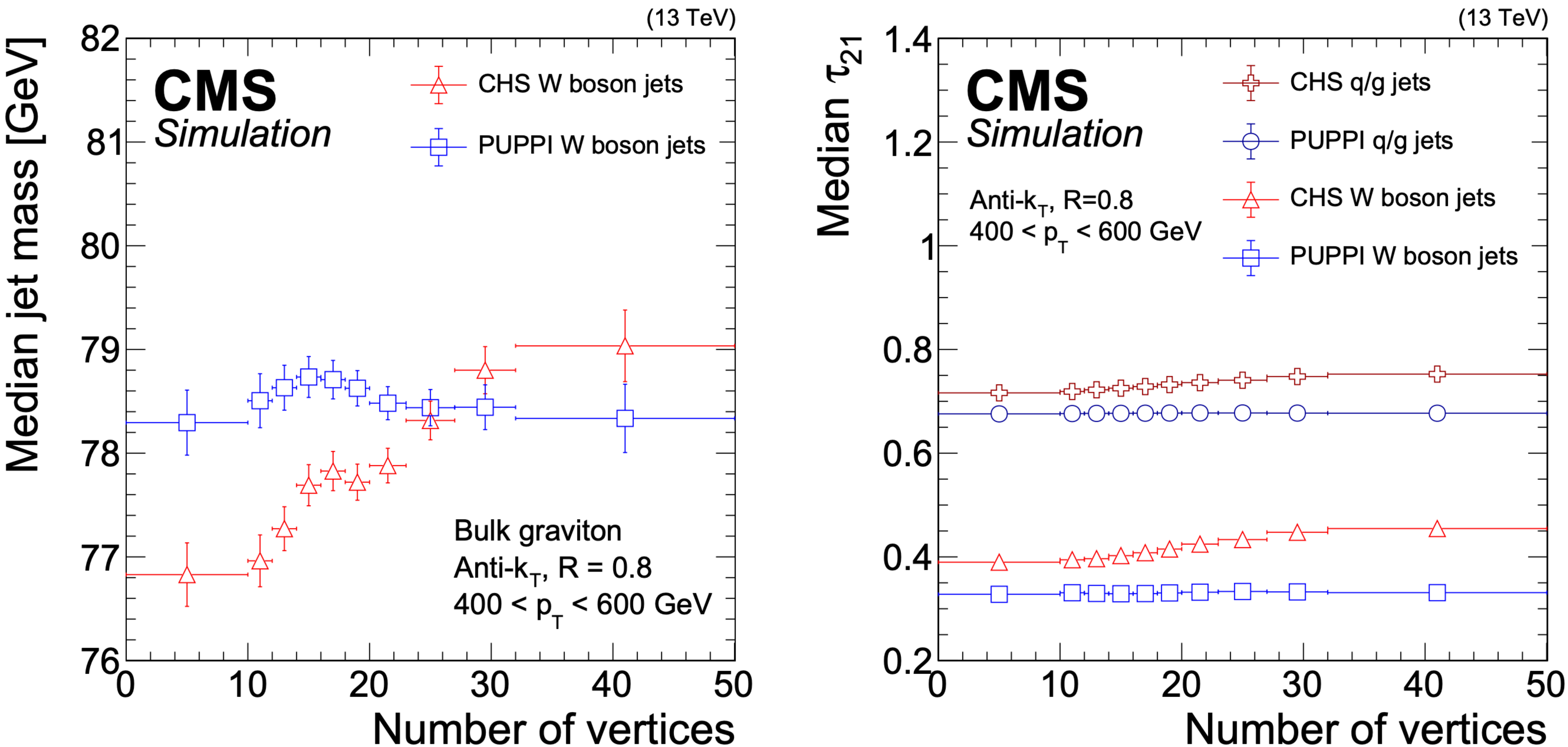
Pileup correction





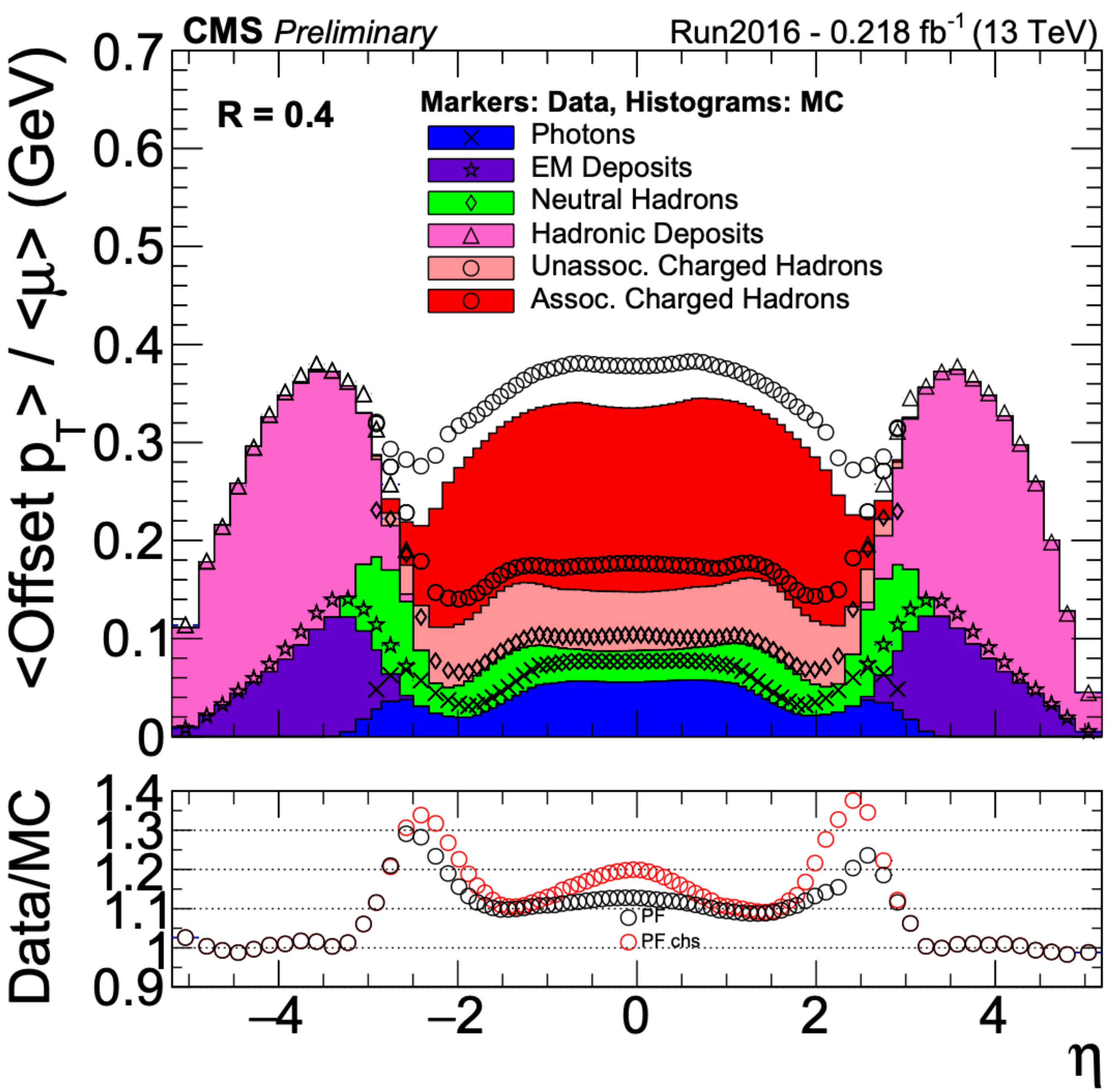
https://arxiv.org/pdf/2003.00503

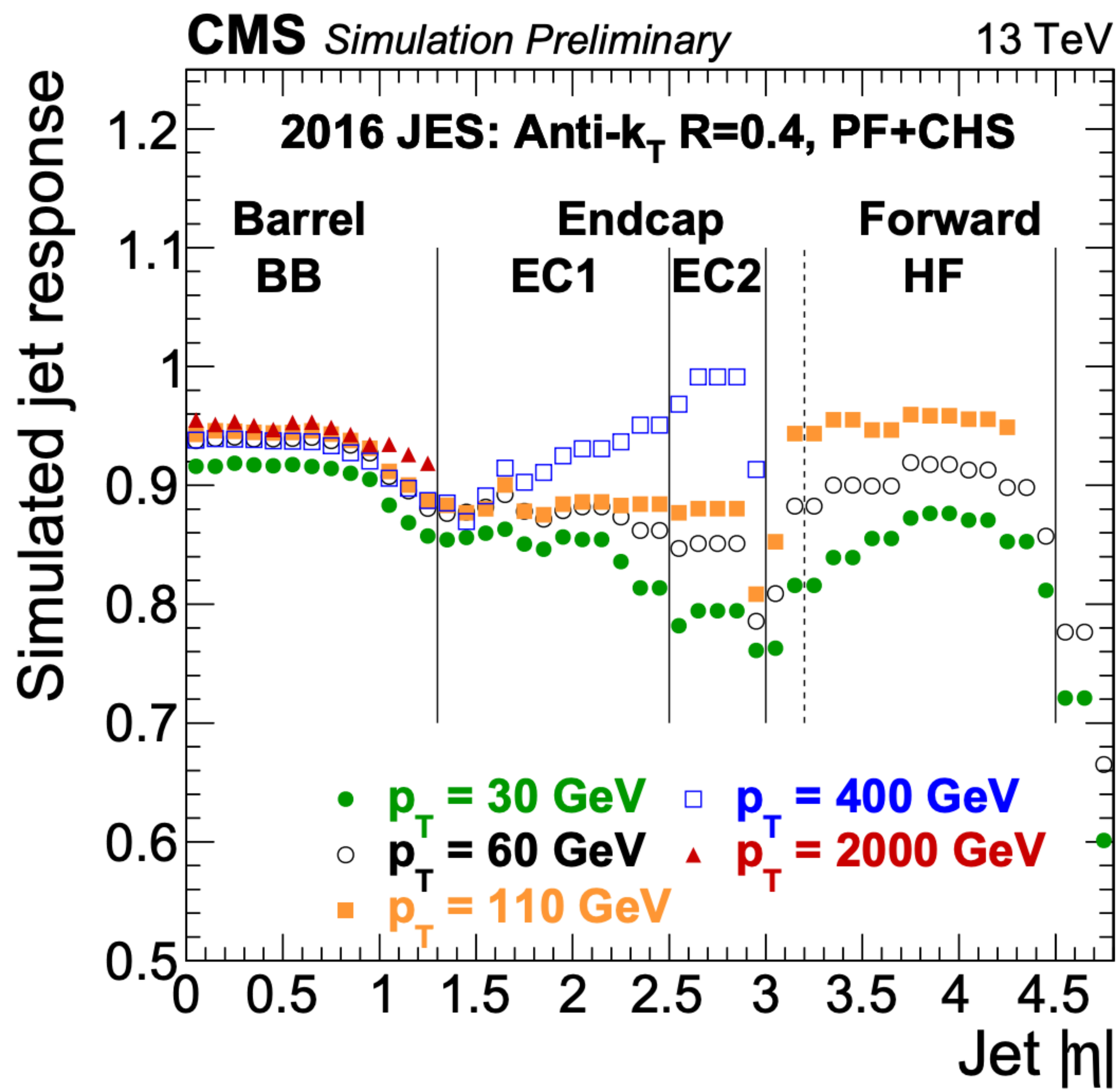
Pileup correction



https://arxiv.org/pdf/2003.00503

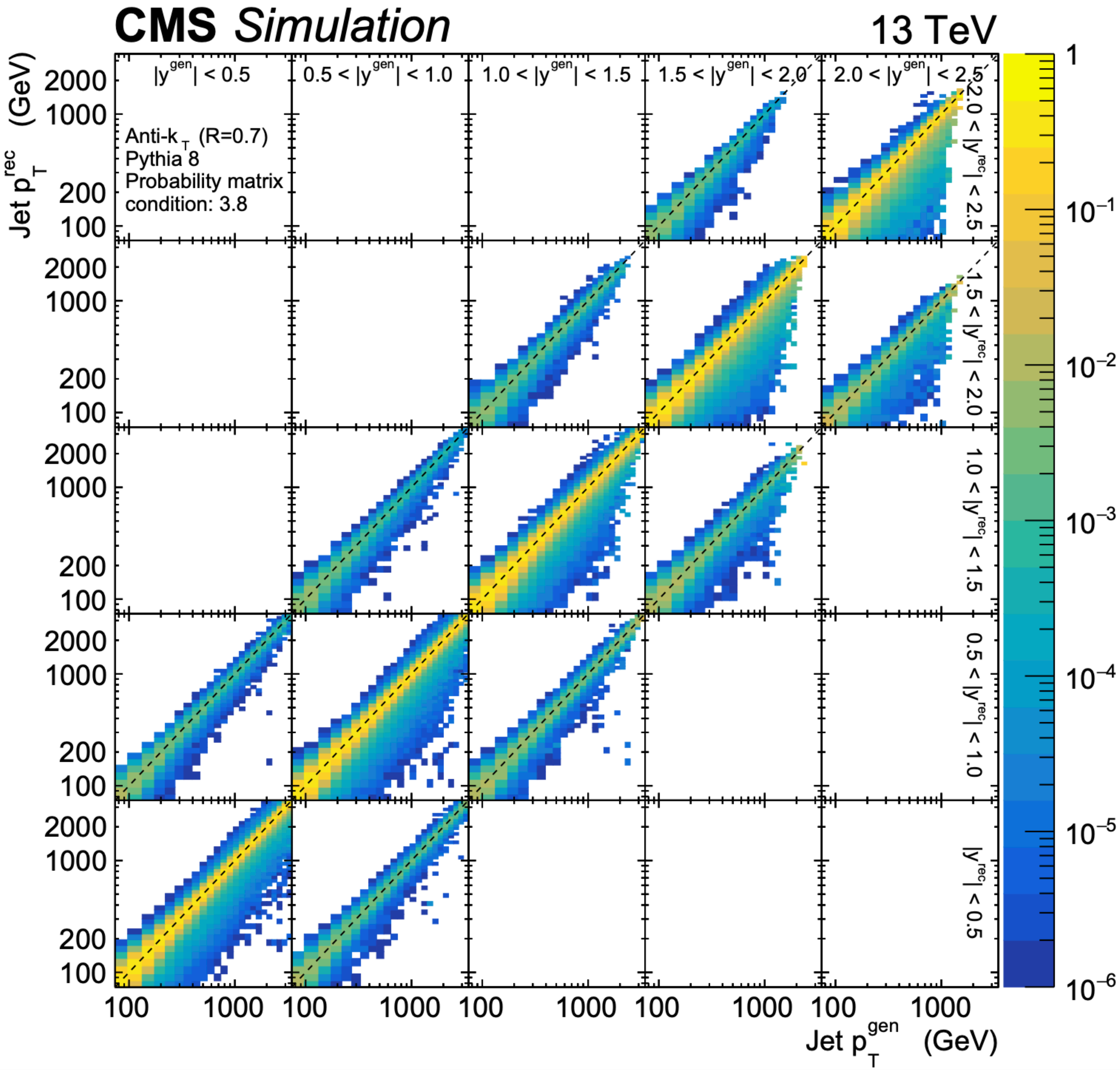
pT/eta dependent JEC



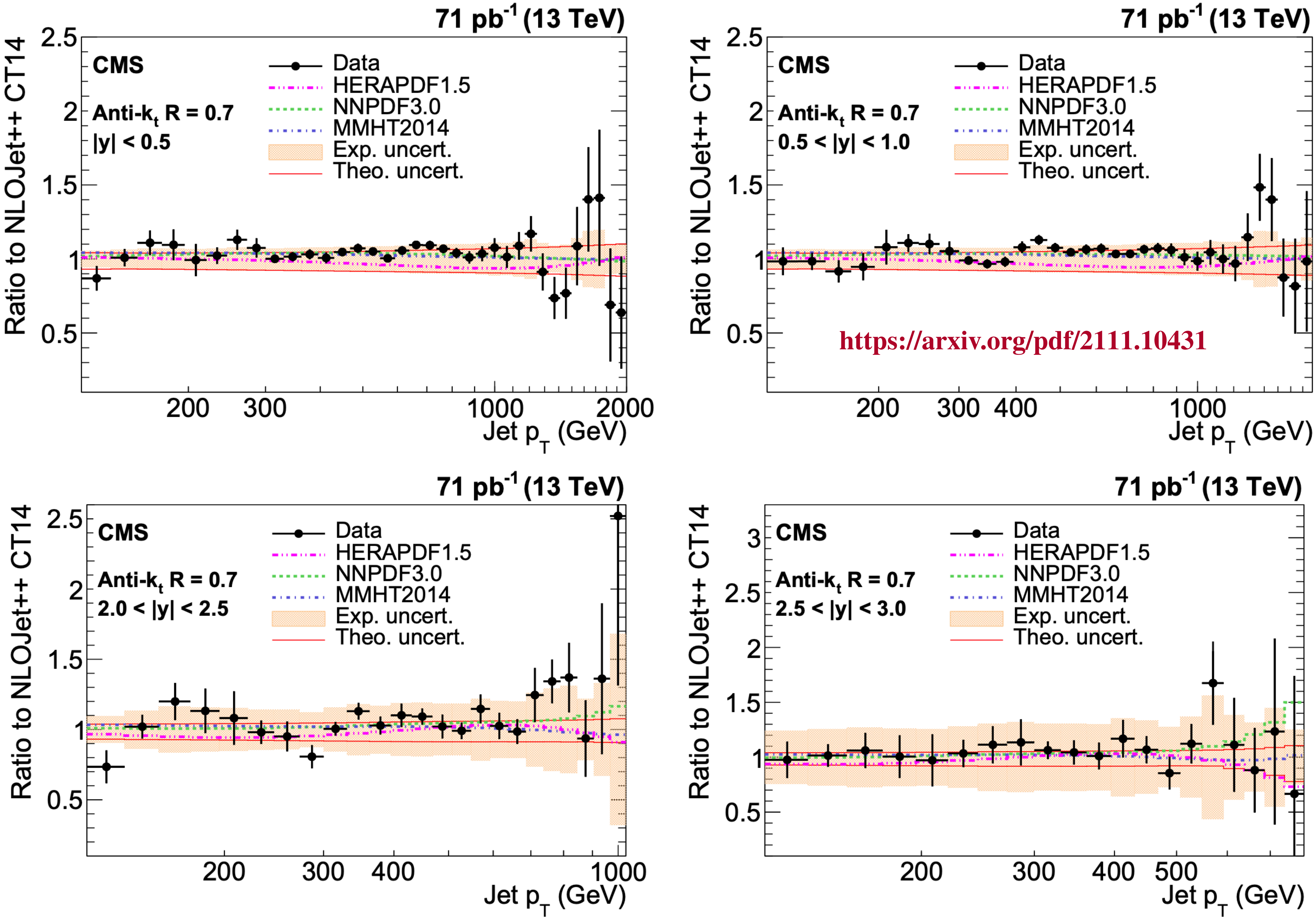


https://inspirehep.net/files/0ac1e2e157965e3491f289c6b86f18ce

Unfolding a pT spectra

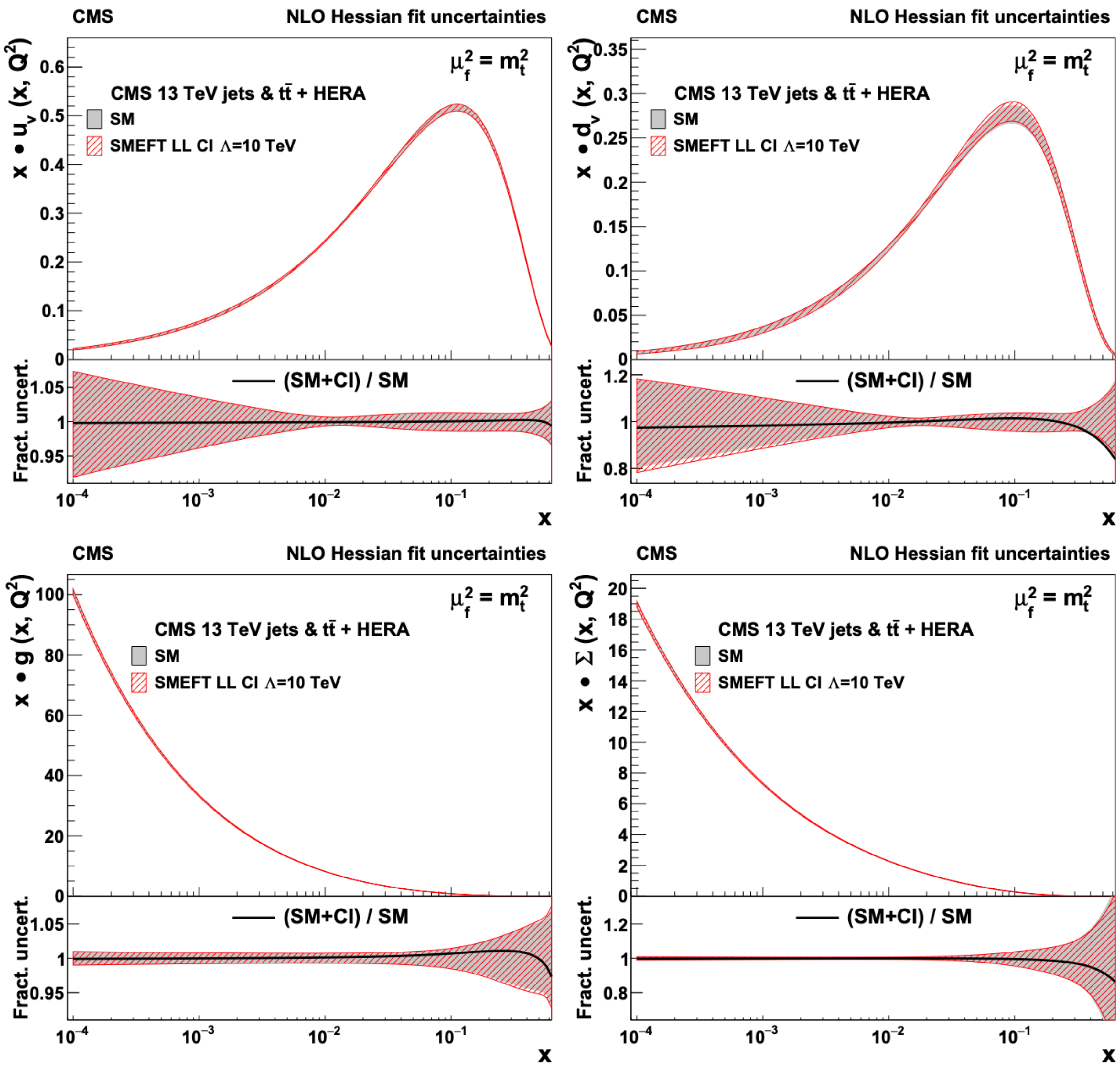


How reliable are these measurements?



PDF extraction from the measurement

https://arxiv.org/pdf/2111.10431



α_S extraction from the measurement

https://twiki.cern.ch/twiki/bin/view/CMSPublic/PhysicsResultsCombined https://arxiv.org/pdf/2111.10431