XX Frascati Summer School "Bruno Touschek"
in Nuclear, Subnuclear and Astroparticle Physics
July 2022, Frascati, Italy

QCD and Jets at Colliders

Matteo Cacciari
LPTHE Paris and Université Paris Cité

Lecture 1: QCD and perturbative calculations and tools

Lecture 2: Jet algorithms and substructure

[Includes material from
Gavin Salam and Grégory Soyez]

Outline

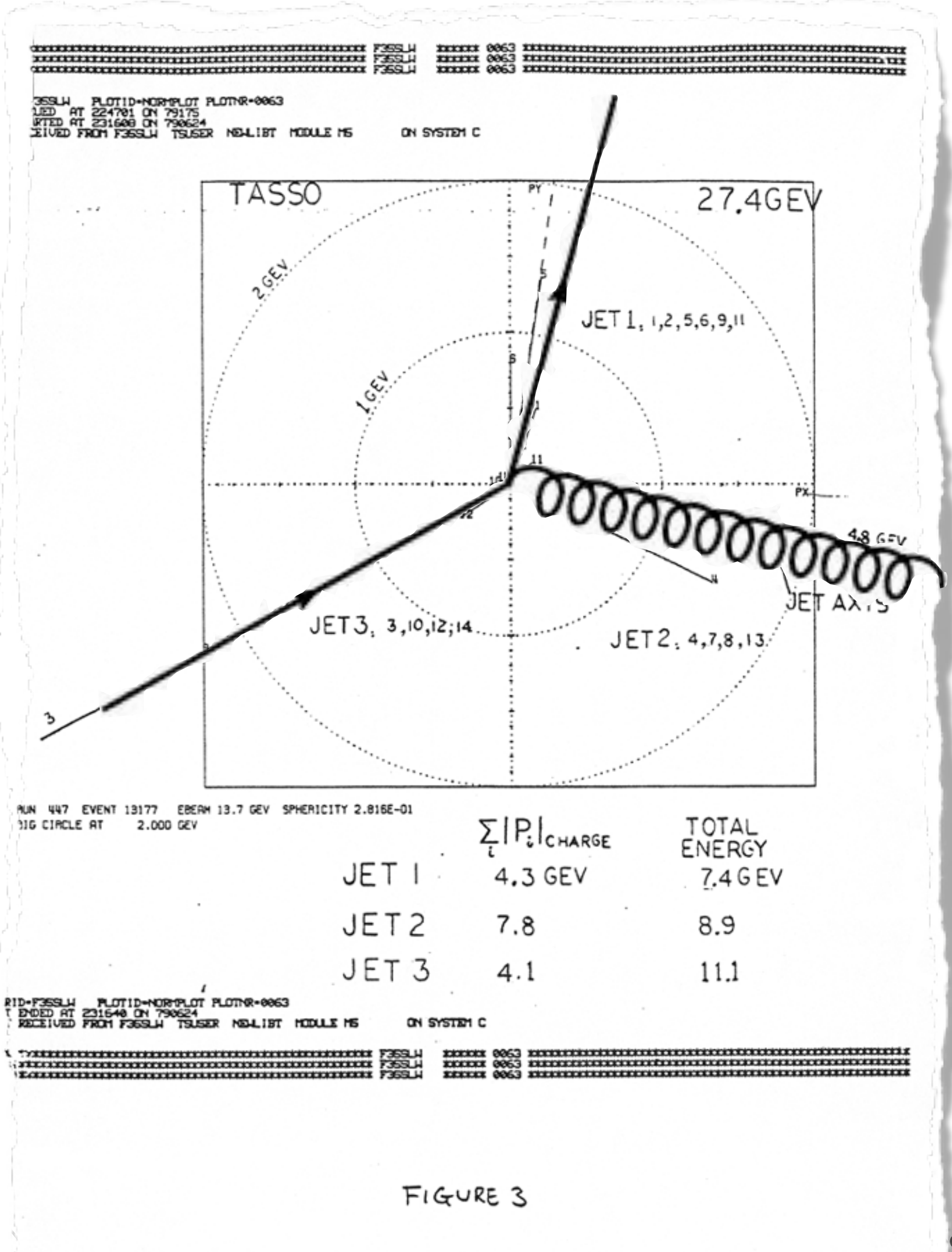
- Jet algorithms
  - How are jets made
- Jet substructure
  - What's inside them, and how to use it

What is a jet?

No, not this....

A **jet** is something that happens
in high energy events:
a collimated bunch of hadrons flying roughly in the same direction

Gluon 'discovery'

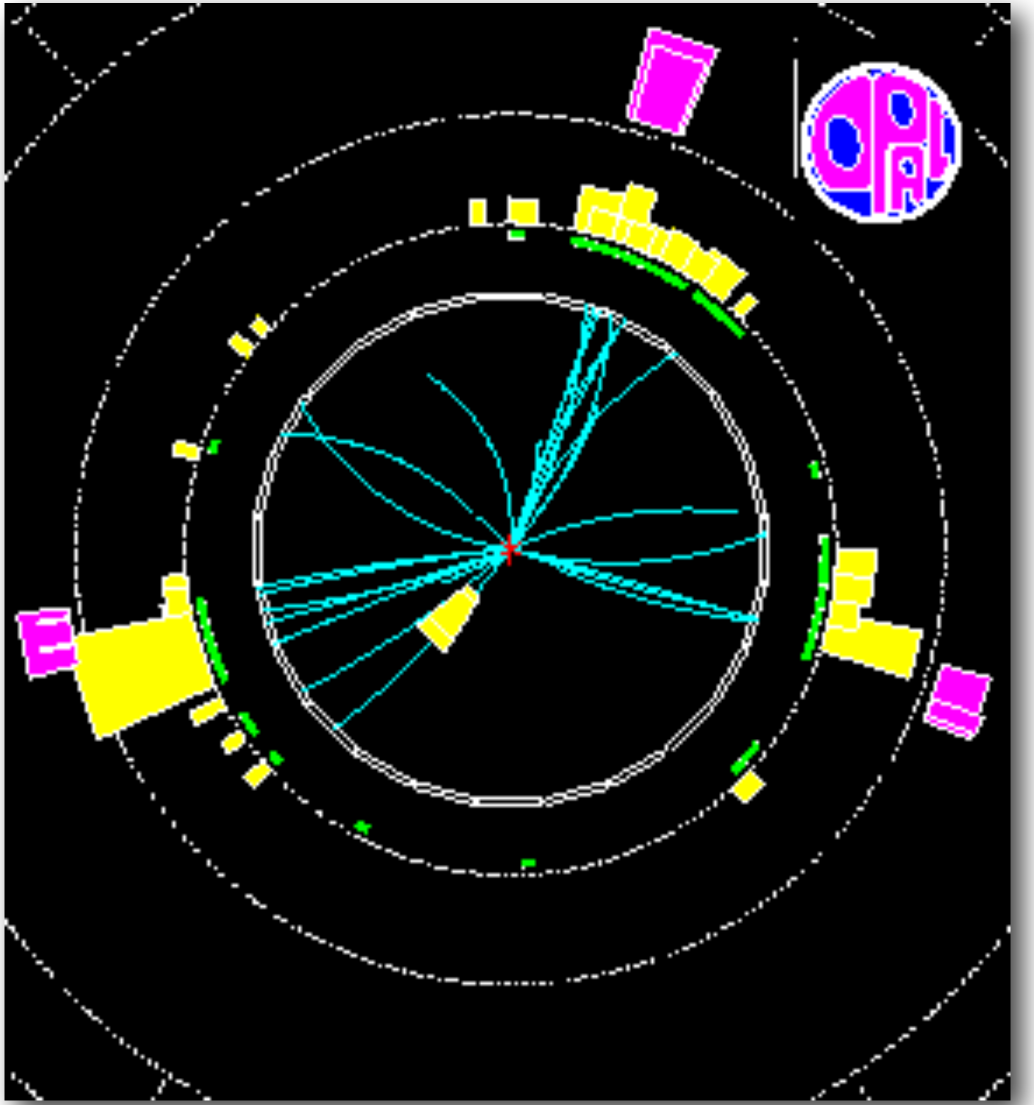


1979:

Three-jet events observed by TASSO, JADE, MARK J and PLUTO at PETRA in e^+e^- collisions at 27.4 GeV

Interpretation:
large angle emission of a hard gluon

Jets viewed as a proxy to the initial partons

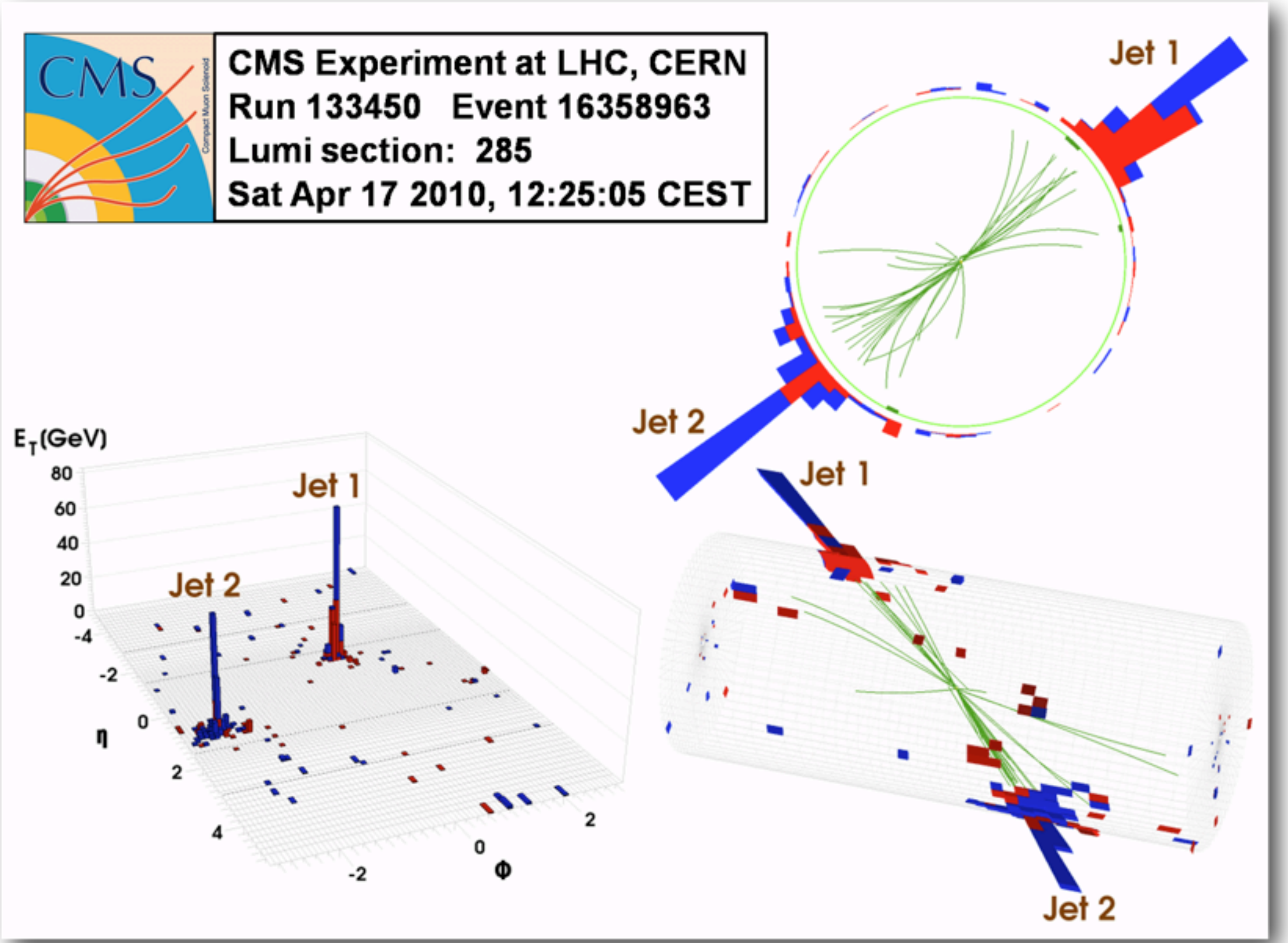


From PETRA to LEP

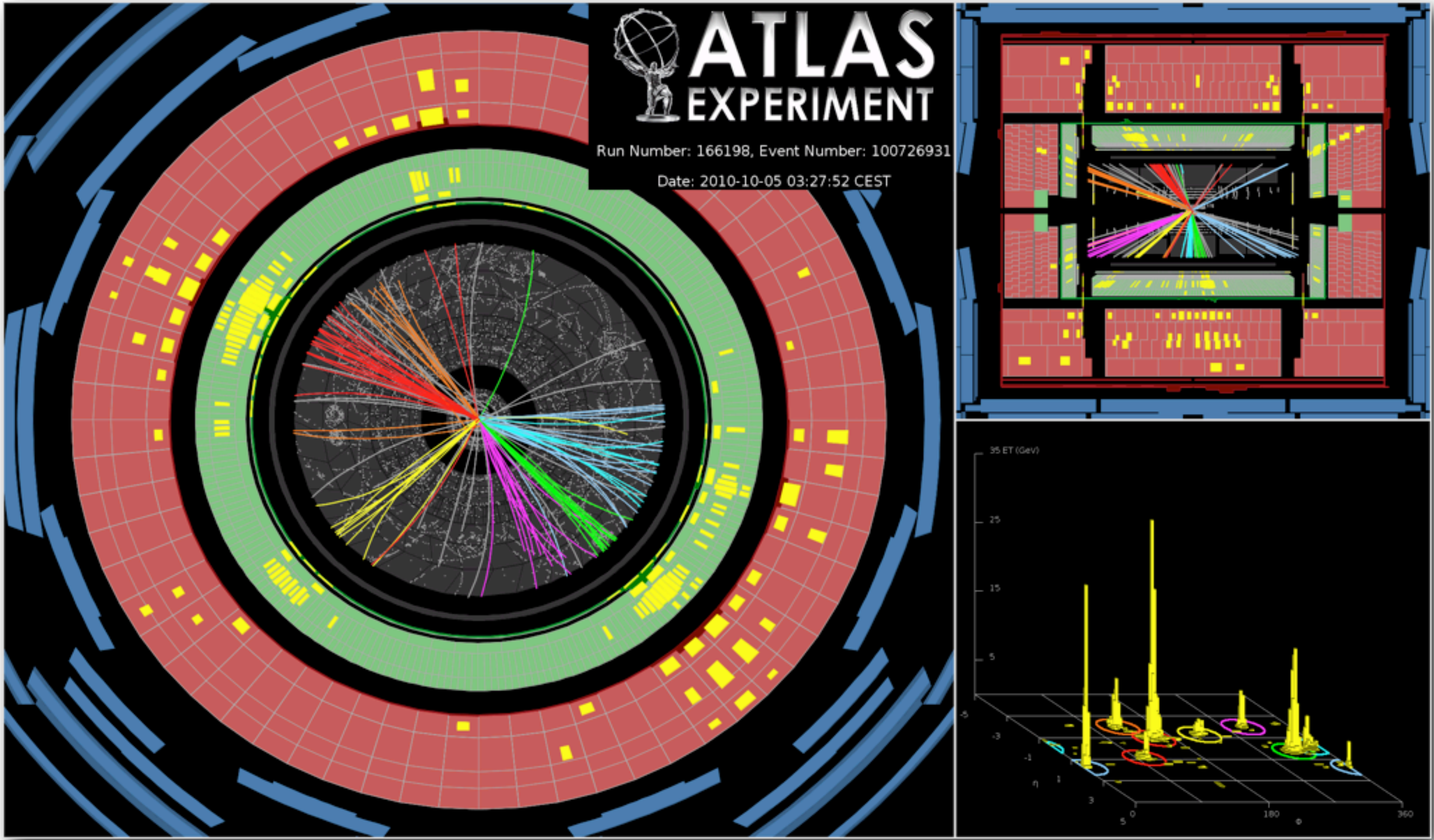
We could eyeball the collimated bunches, but it becomes impractical with millions of events

The classification of particles into jets is best done using a **clustering algorithm**

A few decades after PETRA and LEP, the event displays got prettier, but jets are still pretty much the same

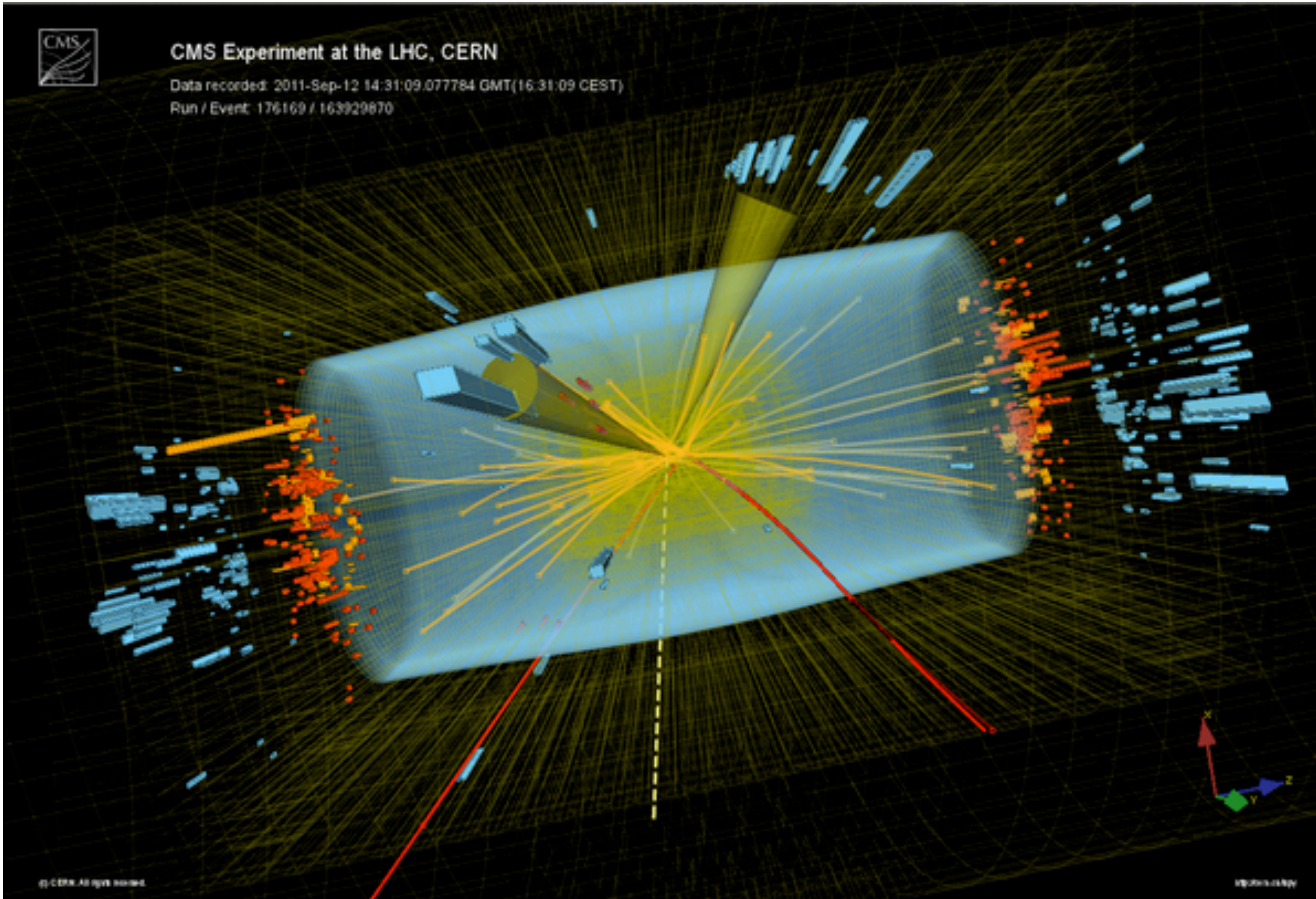


Dijet event from CMS

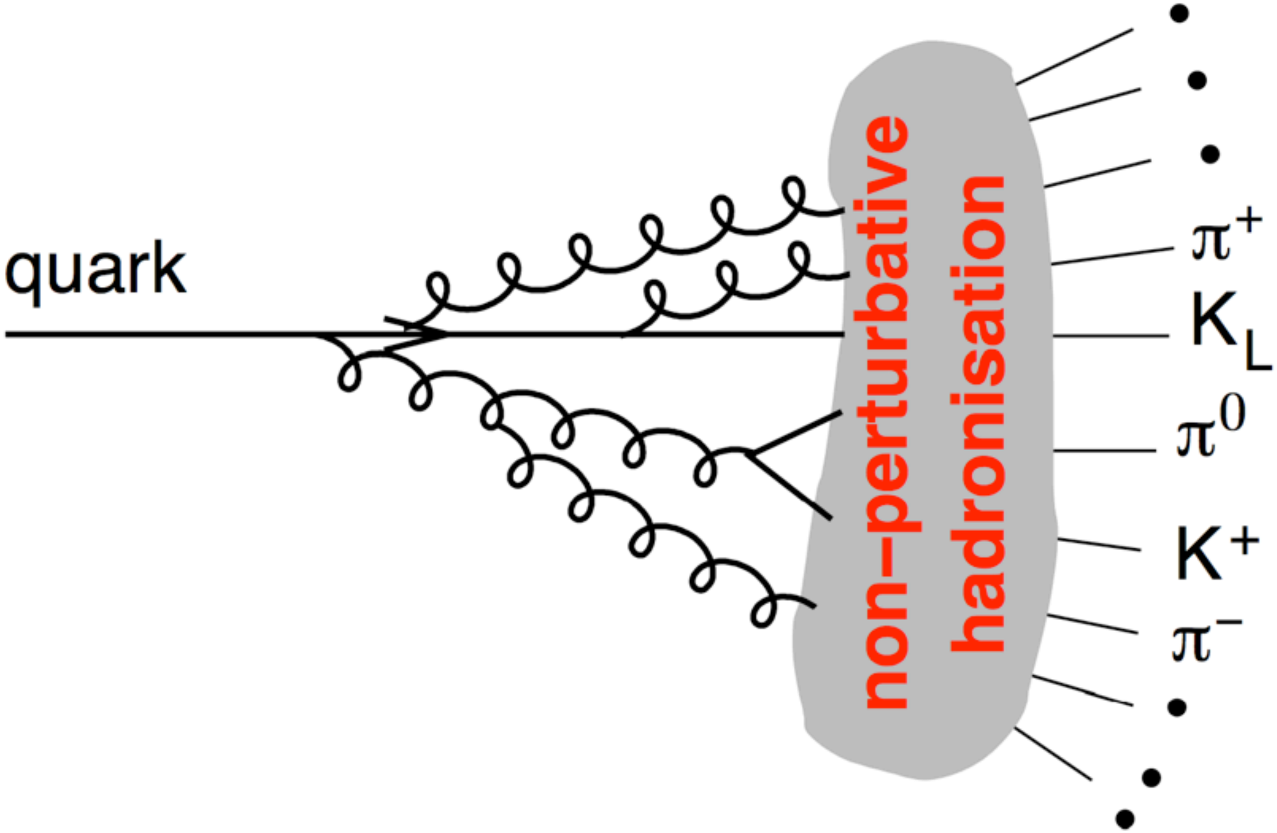


8(!) jets event from ATLAS

Jets @ LHC


CMS
CMS Experiment at the LHC, CERN
Data recorded: 2011-Sep-12 14:31:09.077784 GMT(16:31:09 CEST)
Run / Event: 176169 / 163929870

Why do jets happen?



Gluon emission

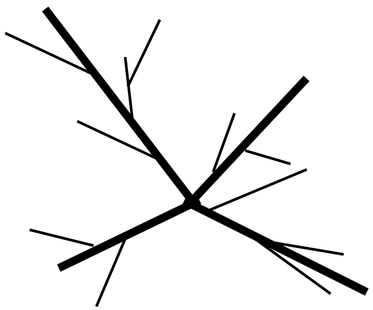
$$\int \alpha_s \frac{dE}{E} \frac{d\theta}{\theta} \gg 1$$

Non-perturbative physics

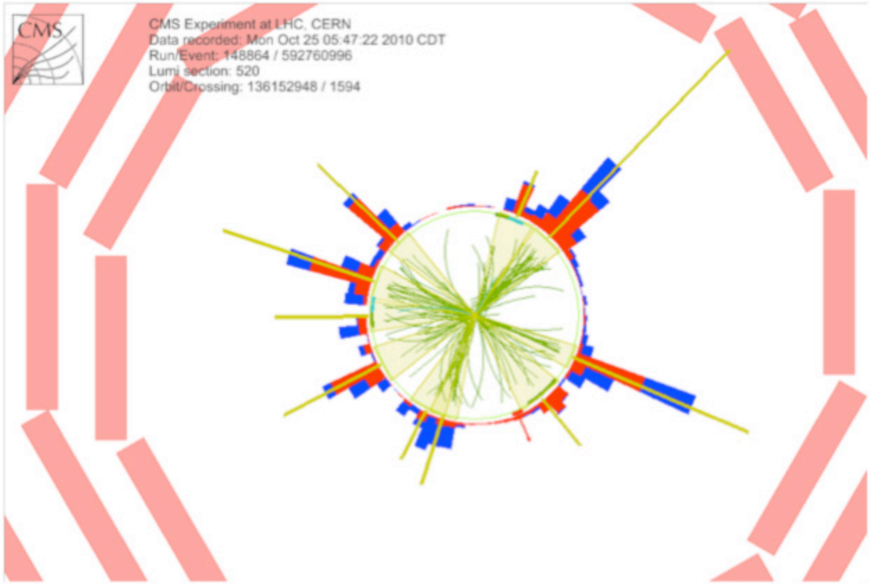
$$\alpha_s \sim 1$$

Taming reality

Multileg + PS



??



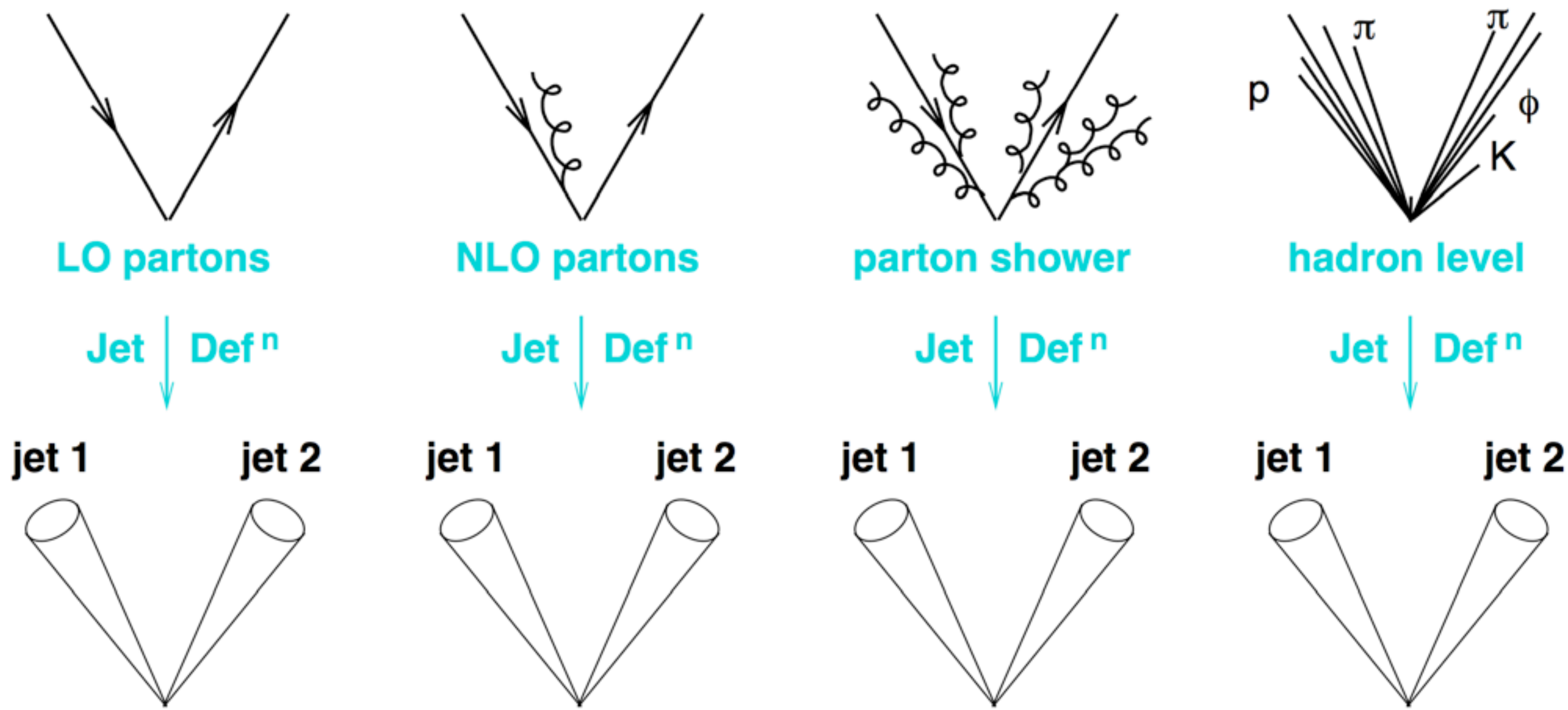
QCD predictions

Real data

Jets

One purpose of a 'jet clustering' algorithm is to **reduce the complexity** of the final state, simplifying many hadrons to **simpler objects** that one can hope to **calculate**

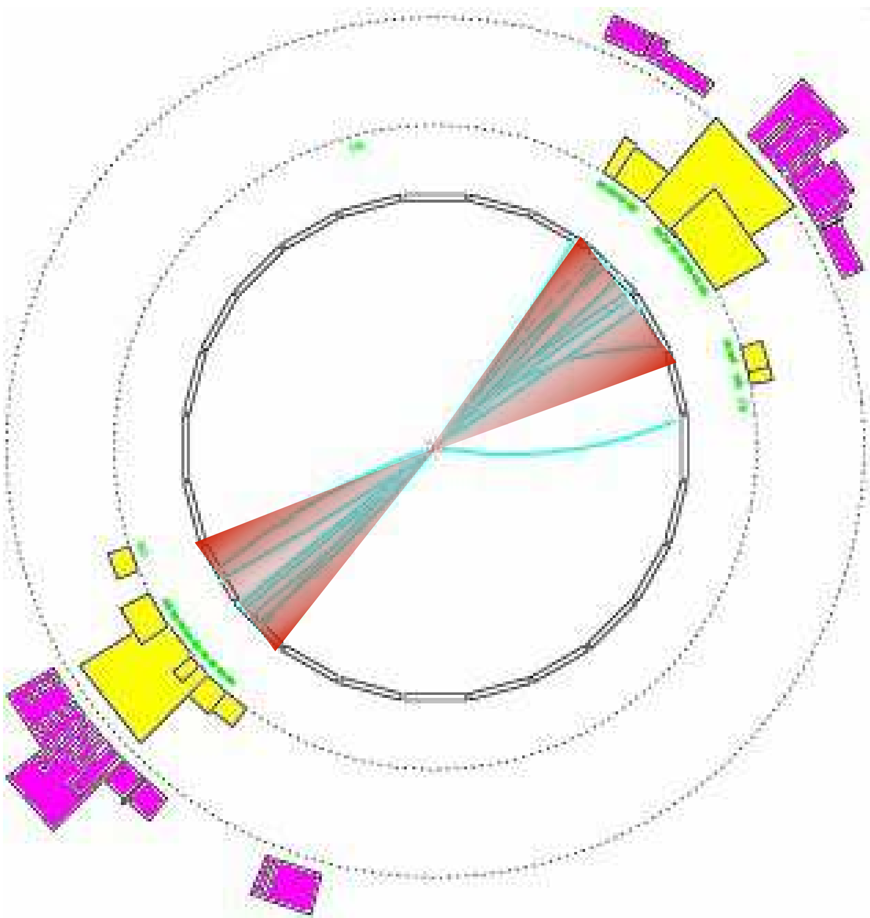
Jet definitions as projections



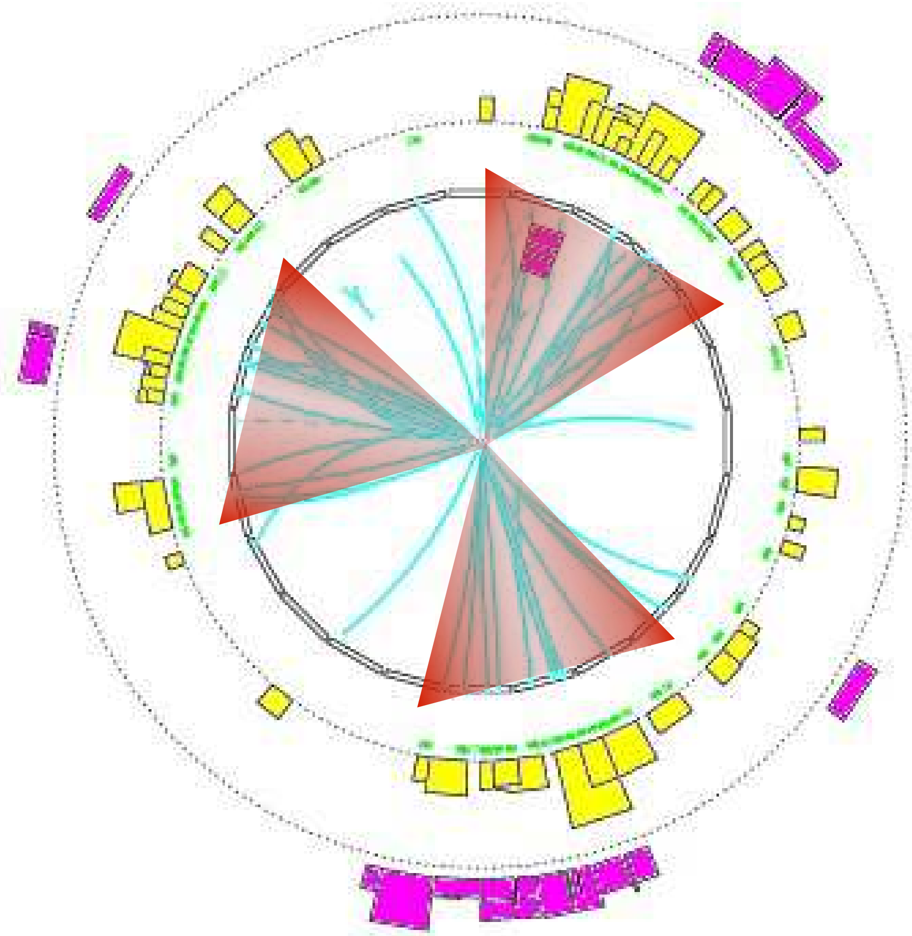
Projection to jets should be resilient to QCD effects

NB: projections are NOT unique:
a jet is NOT EQUIVALENT to a parton

Reconstructing jets is an ambiguous task

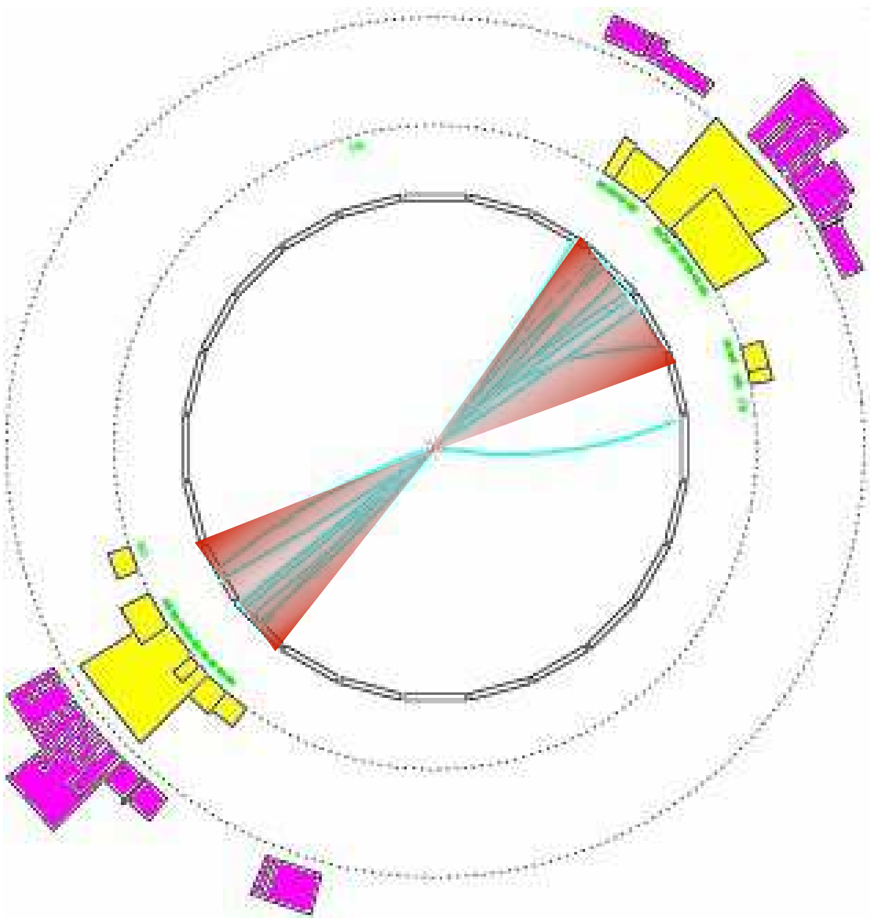


2 clear jets

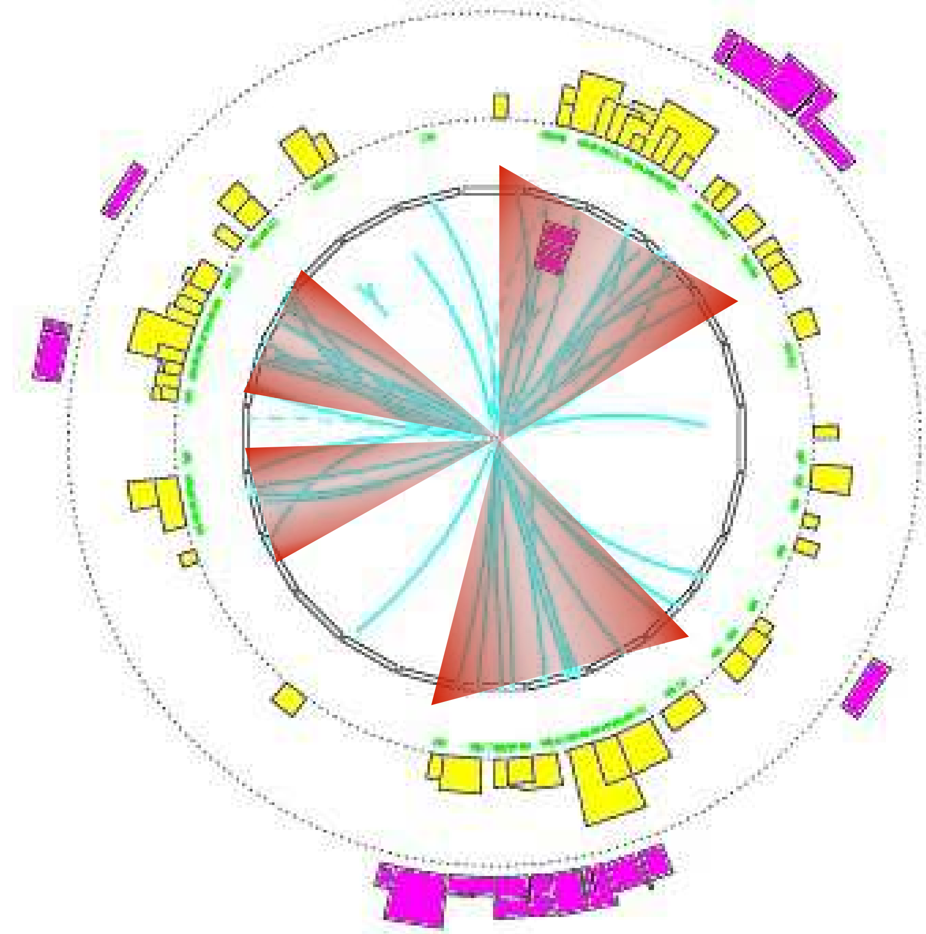


3 jets?

Reconstructing jets is an ambiguous task



2 clear jets



3 jets?
or 4 jets?

Jet clustering algorithm

A **jet algorithm** maps the momenta of the final state particles into the momenta of a certain number of jets:

$$\{p_i\} \xrightarrow{\text{jet algorithm}} \{j_k\}$$

particles,
4-momenta,
calorimeter towers, ...

jets

Most algorithms contain a resolution parameter, **R**, which controls the extension of the jet

"Jet [definitions] are legal contracts between theorists and experimentalists"
-- MJ Tannenbaum

Jets can serve two purposes

- They can be ***observables***, that one can measure and calculate
- They can be ***tools***, that one can employ to extract specific properties of the final state

Different clustering algorithms have different properties and characteristics that can make them more or less appropriate for each of these tasks

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:**

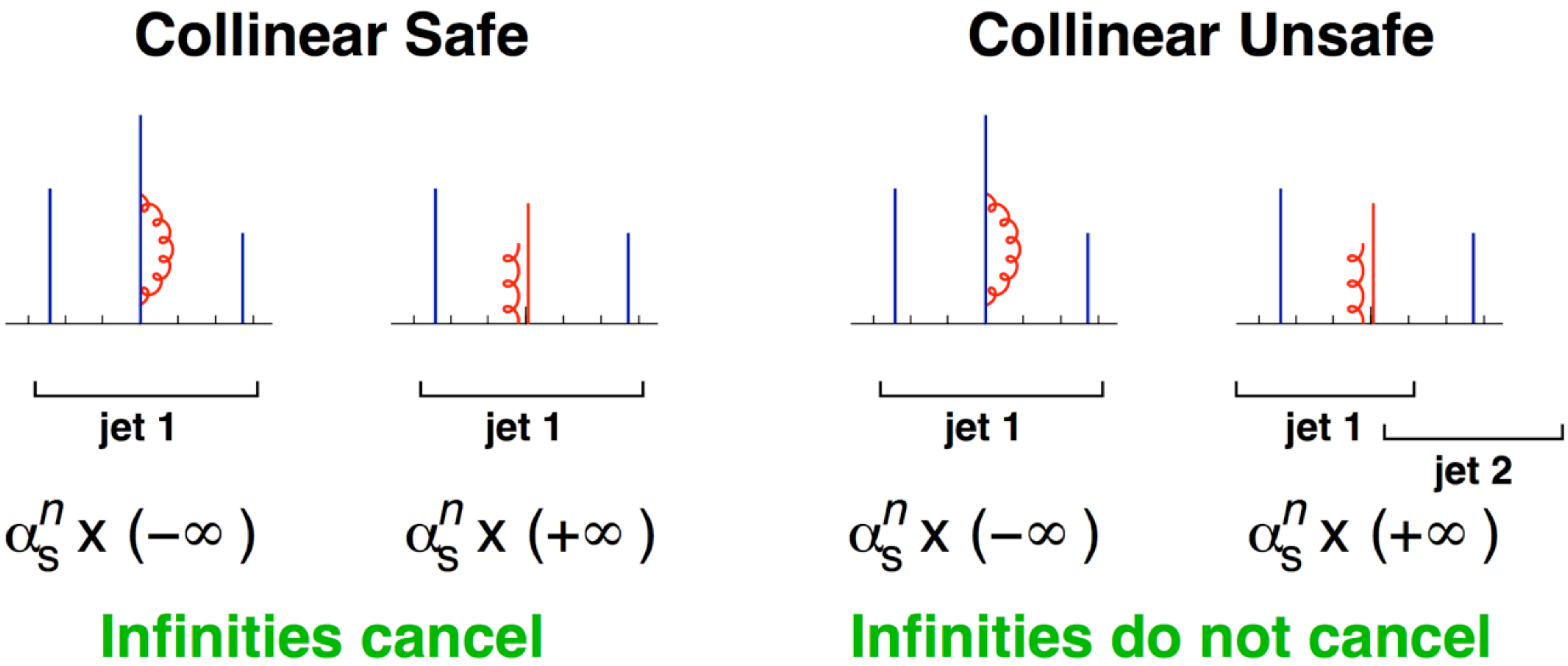
$$O(X; p_1, \ldots, p_n, p_{n+1} \to 0) \to O(X; p_1, \ldots, p_n)$$

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

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

Reconstructing jets must respect rules



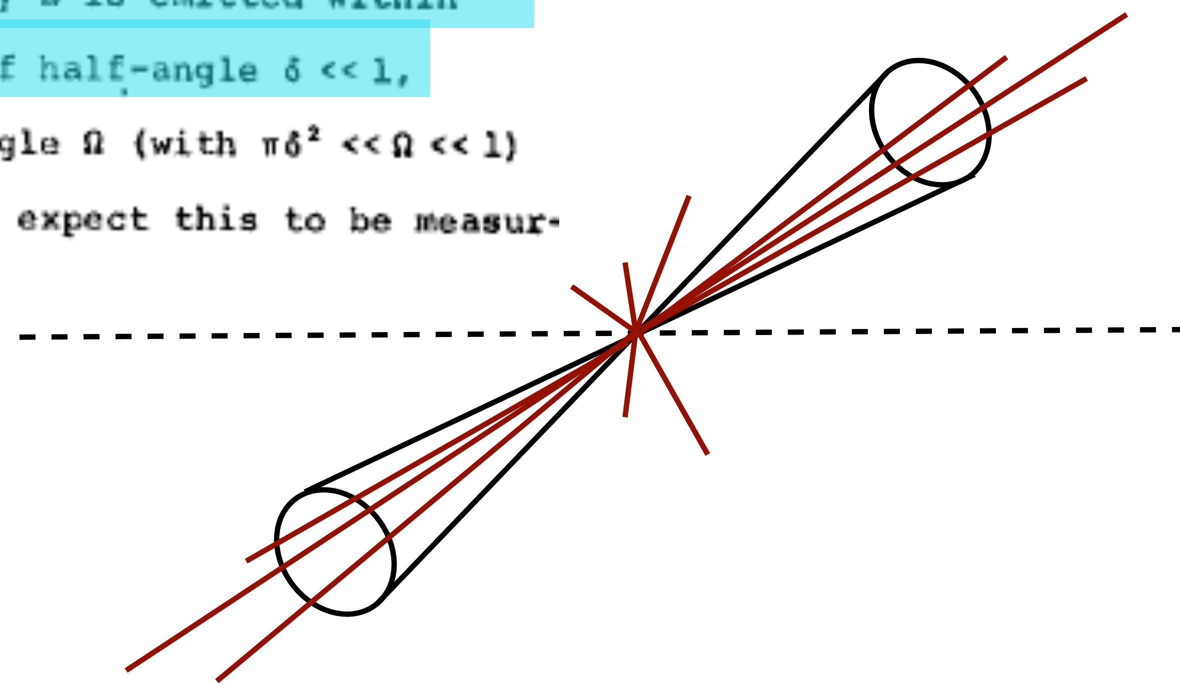
Perturbative calculations of jet observable will only be possible with collinear (and infrared) safe jet definitions

Cone algorithms

The first rigorous definition of cone jets in QCD is due to Sterman and Weinberg

Phys. Rev. Lett. **39**, 1436 (1977)

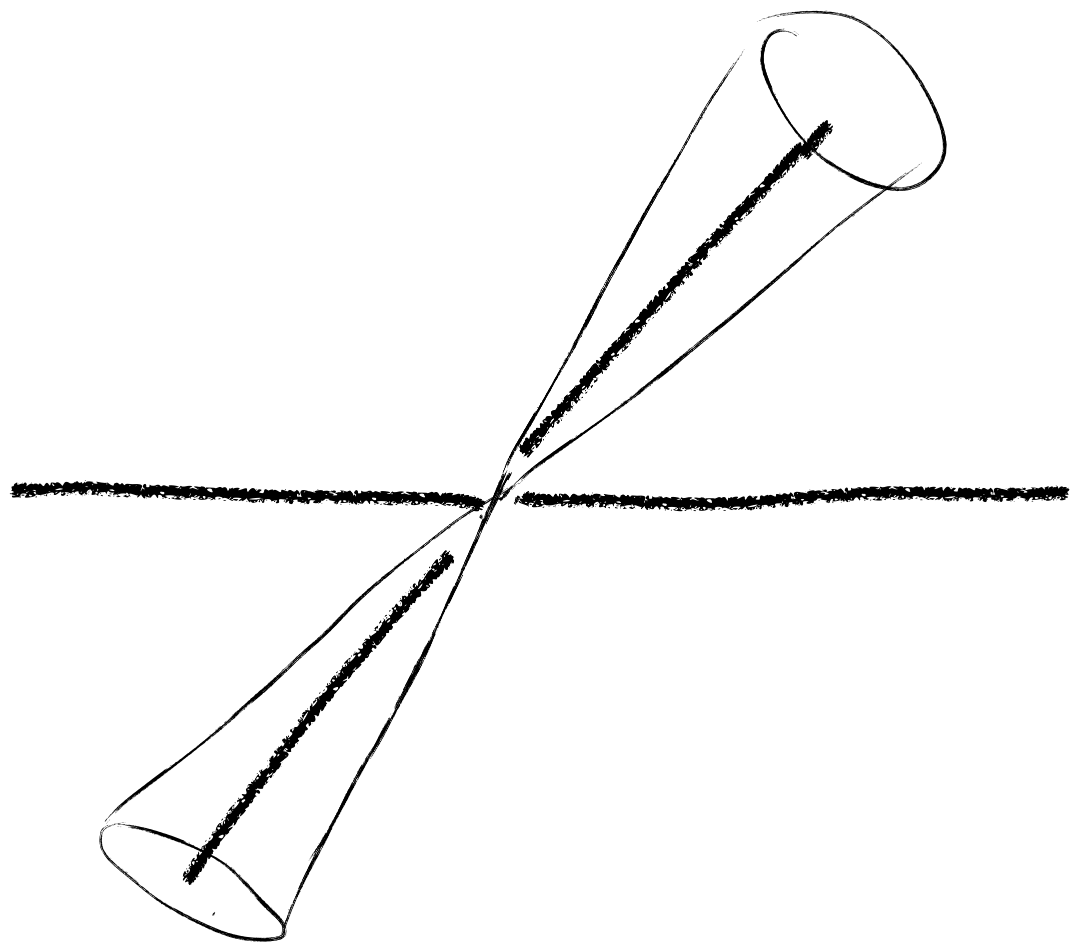
To study jets, we consider the partial cross section $\sigma(E,\theta,\Omega,\epsilon,\delta)$ for e^+e^- hadron production events, in which all but a fraction $\epsilon \ll 1$ of the total e^+e^- energy E is emitted within some pair of oppositely directed cones of half-angle $\delta \ll 1$, lying within two fixed cones of solid angle Ω (with $\pi\delta^2 \ll \Omega \ll 1$) at an angle θ to the e^+e^- beam line. We expect this to be measur-



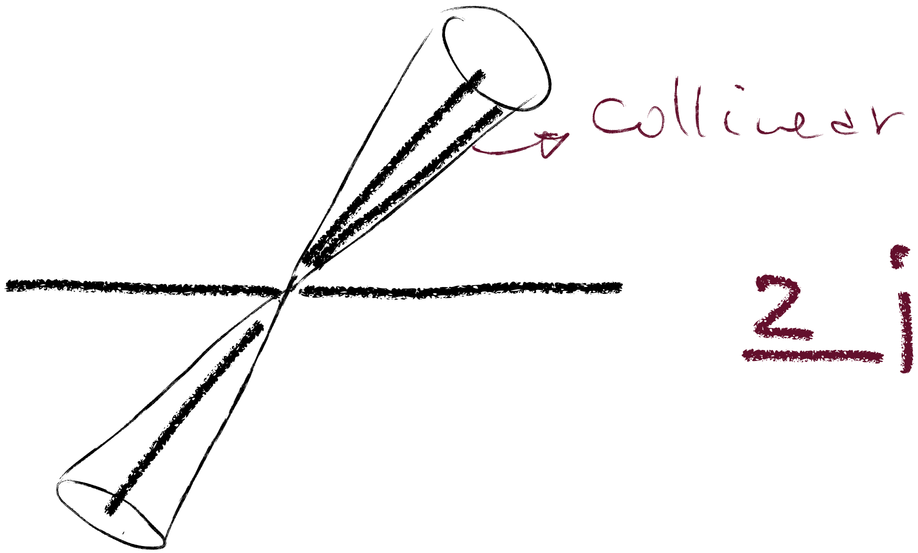
Two-jet rate:

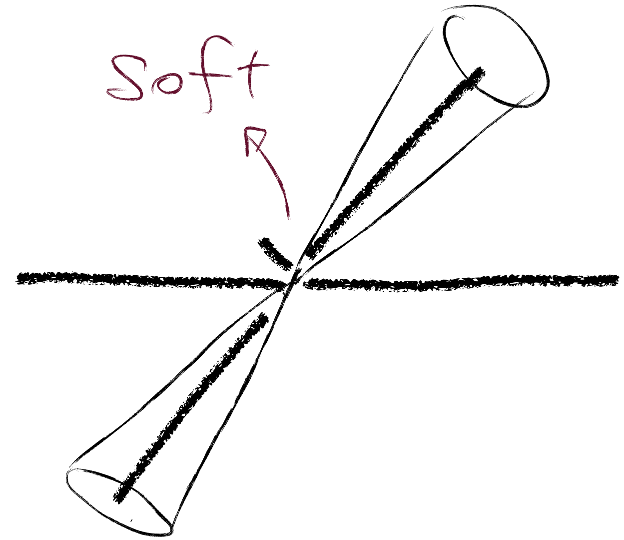
$$\sigma(E,\theta,\Omega,\epsilon,\delta) = (d\sigma/d\Omega)_0\,\Omega\left[1 - (g_E^2/3\pi^2)\left\{3\ln\delta + 4\ln\delta\,\ln 2\epsilon + \frac{\pi^3}{3} - \frac{5}{2}\right\}\right]$$

2 particles = 2 jets

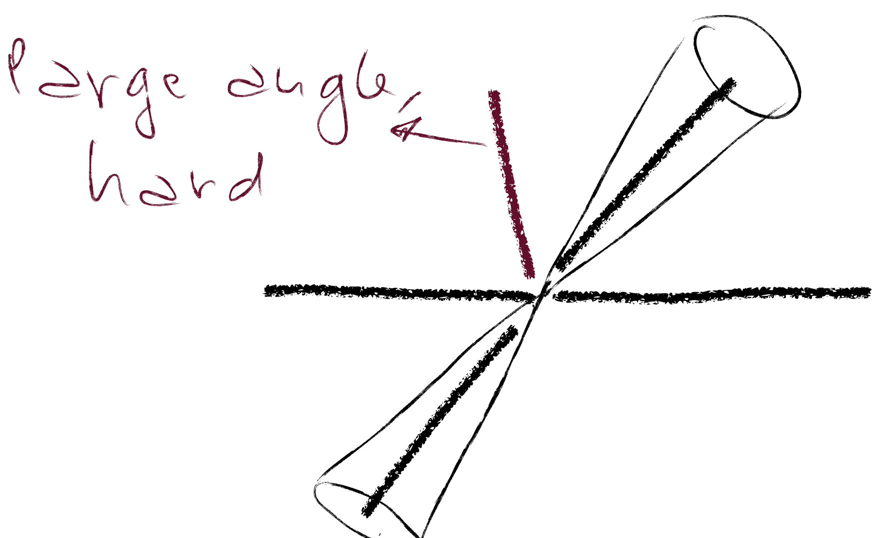


3 particles → ?





2 jets



3 jets

The Sterman-Weinberg definition is "inclusive enough" for IRC safety

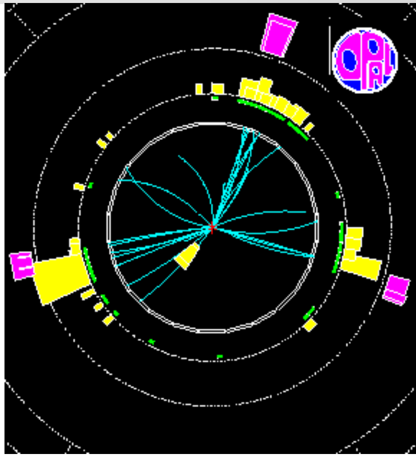
Good for 2 jets and e^+e^- collisions

What happens in a more general case, where more than two jets are likely to exist?

Where do we place the cones? **How many**?

Iterative jet algorithms

Two main approaches to jet clustering



1. Find regions where a lot of energy flows

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)

Two main classes of jet 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

Recombination algorithms

- First introduced in e^+e^- collisions in the ’80s
- Typically they work by calculating a **‘distance’** between particles, and then recombine them pairwise according to a given order, until some condition is met (e.g. no particles are left, or the distance crosses a given threshold)

IRC safety can usually be seen to be trivially guaranteed

JADE algorithm

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

- Find the minimum y_{min} of all y_{ij}
- If y_{min} is below some jet resolution threshold y_{cut}, recombine i and j into a single new particle ('pseudojet'), and repeat
- If no $y_{min} < y_{cut}$ are left, all remaining particles are jets

Problem of this particular algorithm:
two **soft** particles emitted at **large angle** get easily recombined into a single jet: counterintuitive and perturbatively troublesome

e^+e^- k_t (Durham) algorithm

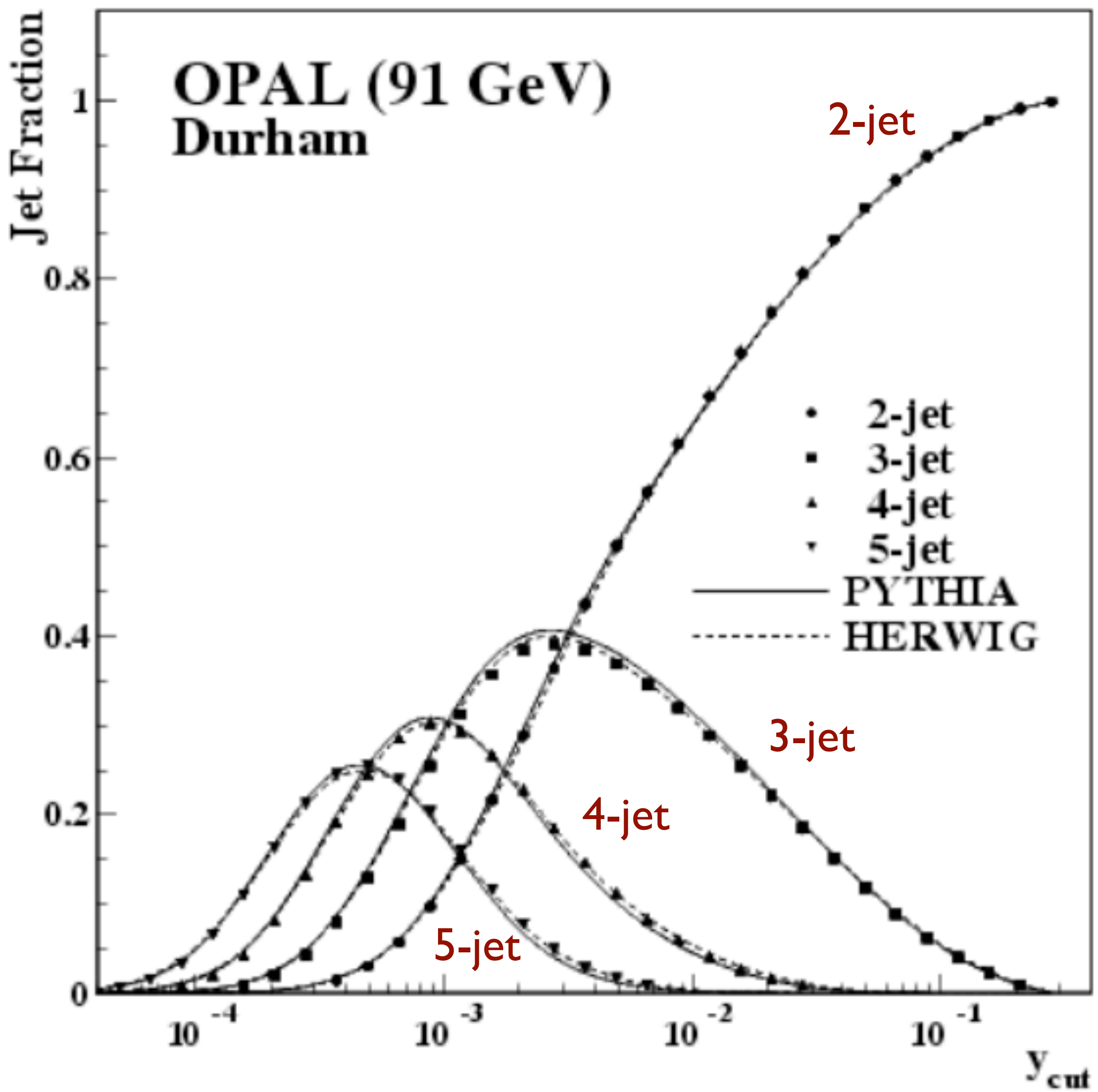
[Catani, Dokshitzer, Olsson, Turnock, Webber '91]

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

In the collinear limit, the numerator reduces to the **relative transverse momentum** (squared) of the two particles, hence the name of the algorithm

- Find the minimum y_{min} of all y_{ij}
- If y_{min} is below some jet resolution threshold y_{cut}, recombine i and j into a single new particle ('pseudojet'), and repeat
- If no $y_{min} < y_{cut}$ are left, all remaining particles are jets

e^+e^- k_t (Durham) algorithm in action



Characterise events in terms of number of jets (as a function of y_{cut})

Resummed calculations for distributions of y_{cut} doable with the k_t algorithm

e^+e^- k_t (Durham) algorithm v. QCD

k_t is a sequential recombination type algorithm

One key feature of the k_t algorithm is its relation to the structure of QCD divergences:

$$\frac{dP_{k\to ij}}{dE_i d\theta_{ij}} \sim \frac{\alpha_s}{\min(E_i, E_j)\theta_{ij}}$$

The y_{ij} distance is the inverse of the emission probability

- The k_t algorithm roughly inverts the QCD branching sequence (the pair which is recombined first is the one with the largest probability to have branched)
- The history of successive clusterings has physical meaning

k_t algorithm in hadron collisions

(Inclusive and longitudinally invariant version)

$$d_{ij} = \min(p_{ti}^2, p_{tj}^2)\frac{\Delta R_{ij}^2}{R^2} \qquad d_{iB} = p_{ti}^2$$

- Calculate the distances between the particles: $\mathbf{d_{ij}}$
- Calculate the beam distances: $\mathbf{d_{iB}}$
- Combine particles with **smallest distance** d_{ij} or, if d_{iB} is smallest, call it a jet
- Find again smallest distance and repeat procedure until no particles are left (this stopping criterion leads to the *inclusive* version of the k_t algorithm)
- Only use jets with $p_t > p_{t,min}$

The k_t algorithm and its siblings

$$d_{ij} = \min(p_{ti}^{2p}, p_{tj}^{2p}) \frac{\Delta y^2 + \Delta\phi^2}{R^2} \qquad d_{iB} = p_{ti}^{2p}$$

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**

Quite ironically, a sequential recombination algorithm is the 'perfect' cone algorithm

IRC safety of generalised-k_t algorithms

$$d_{ij} = \min(p_{ti}^{2p}, p_{tj}^{2p}) \frac{\Delta y^2 + \Delta\phi^2}{R^2} \qquad d_{iB} = p_{ti}^{2p}$$

p > 0

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

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

p < 0

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

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

IRC safe algorithms

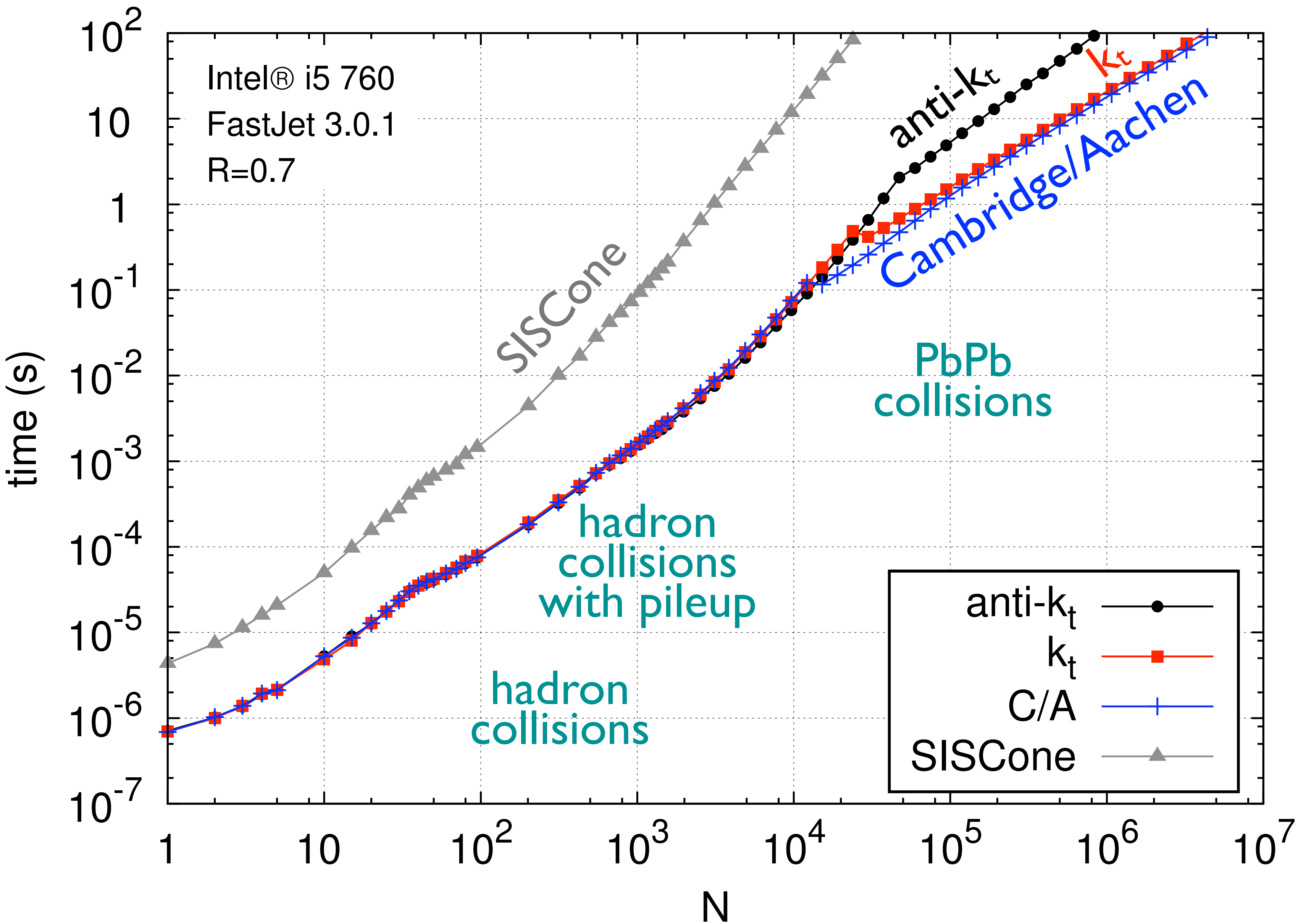
k_t	SR $d_{ij} = \min(p_{ti}^2, p_{tj}^2)\Delta R_{ij}^2/R^2$ hierarchical in rel p_t	Catani et al '91 Ellis, Soper '93	NlnN
Cambridge/Aachen	SR $d_{ij} = \Delta R_{ij}^2/R^2$ hierarchical in angle	Dokshitzer et al '97 Wengler, Wobish '98	NlnN
anti-k_t	SR $d_{ij} = \min(p_{ti}^{-2}, p_{tj}^{-2})\Delta R_{ij}^2/R^2$ gives perfectly conical hard jets	MC, Salam, Soyez '08 (Delsart, Loch)	$N^{3/2}$
SISCone	Seedless iterative cone with split-merge gives 'economical' jets	Salam, Soyez '07	$N^2 \ln N$

'second-generation' algorithms

All are available in FastJet, http://fastjet.fr

(As well as many IRC unsafe ones)

Time needed to cluster an event with N particles



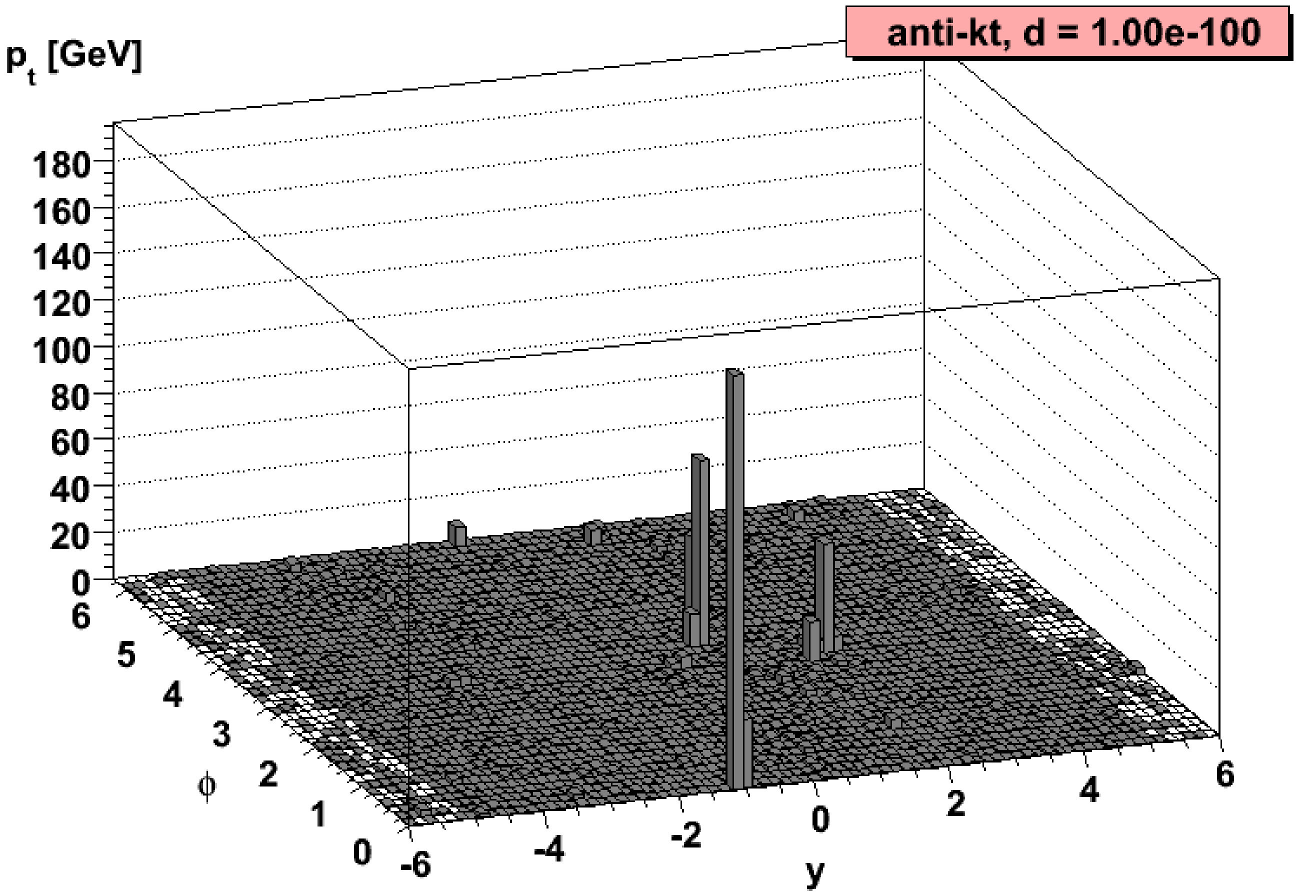
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$

Anti-k_t in action

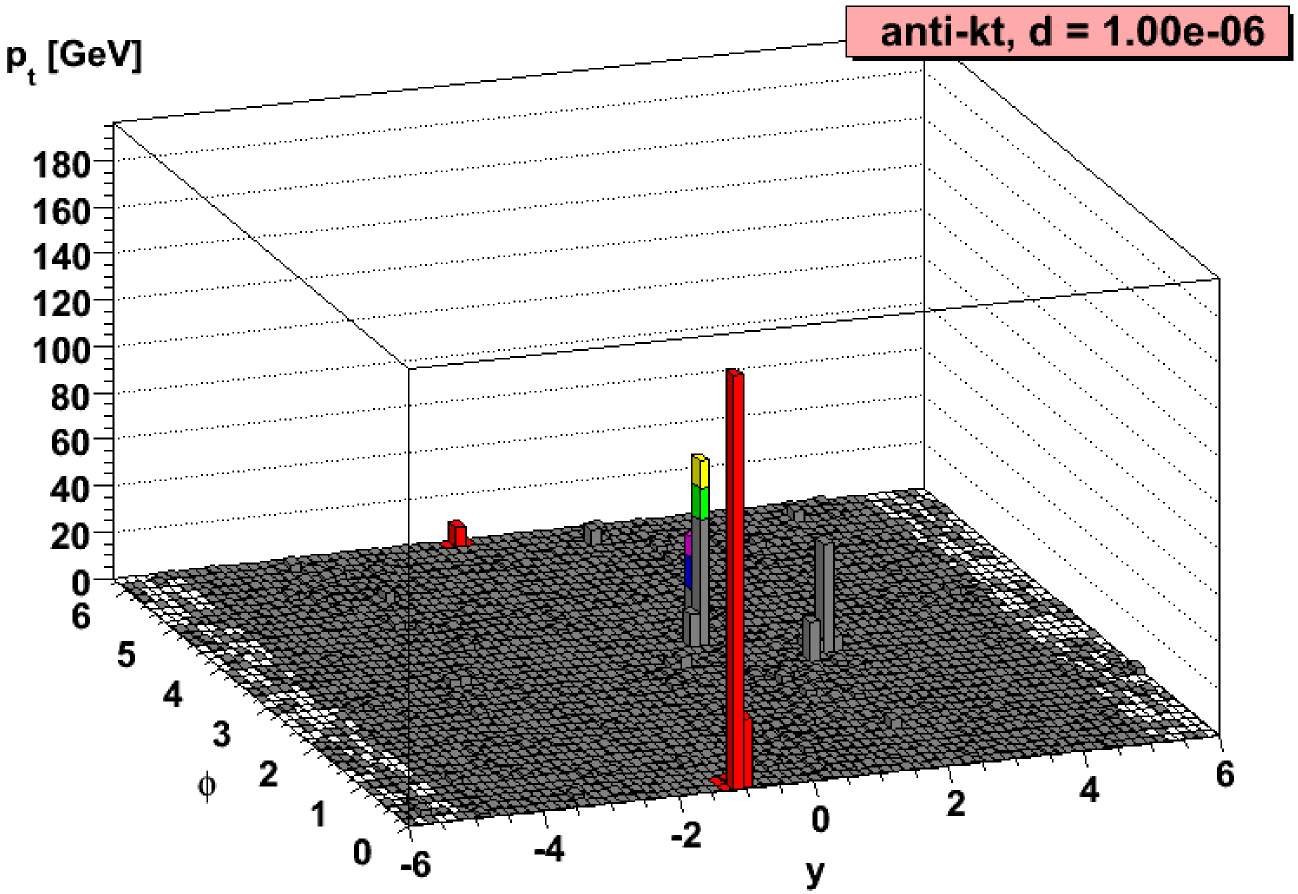
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Clustering grows around hard cores

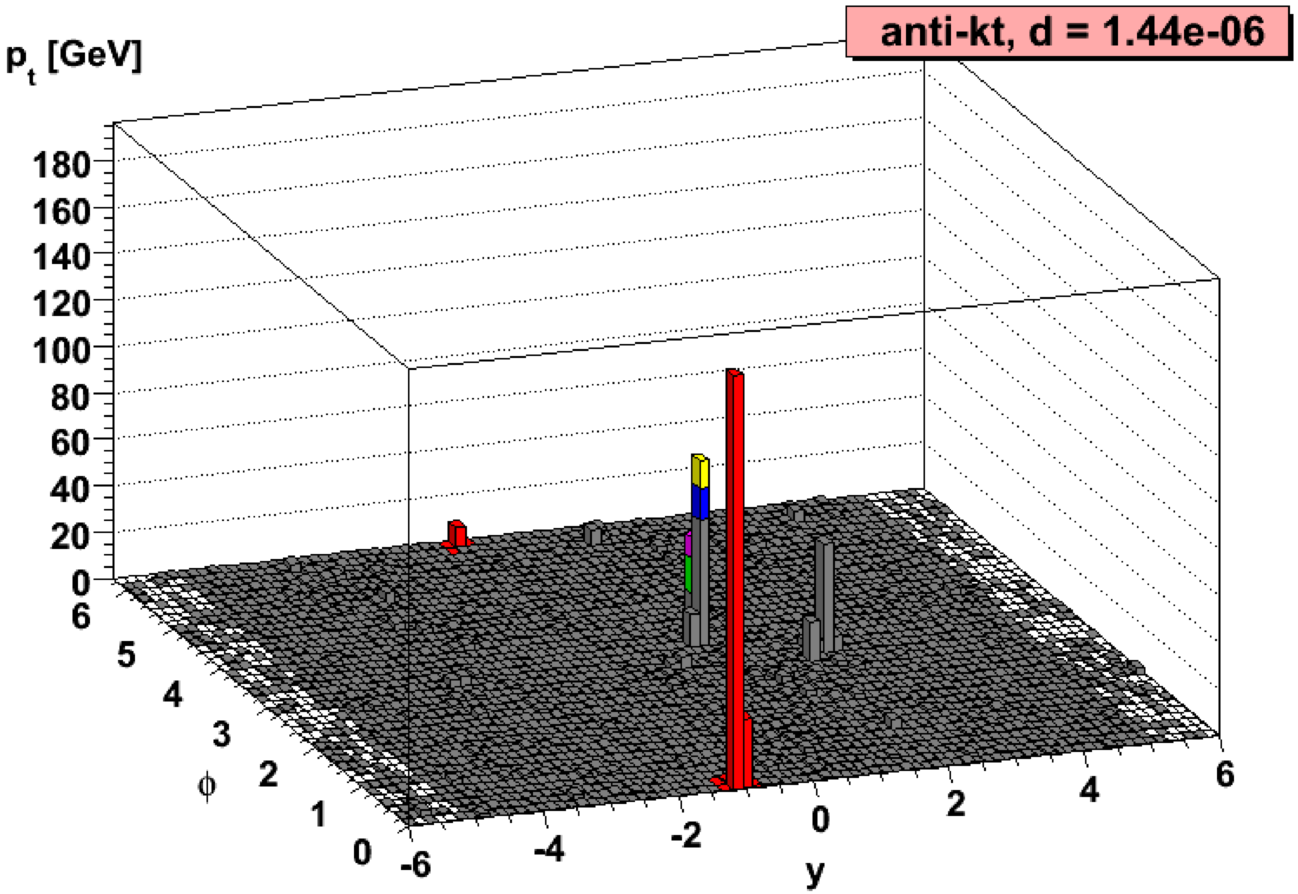
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

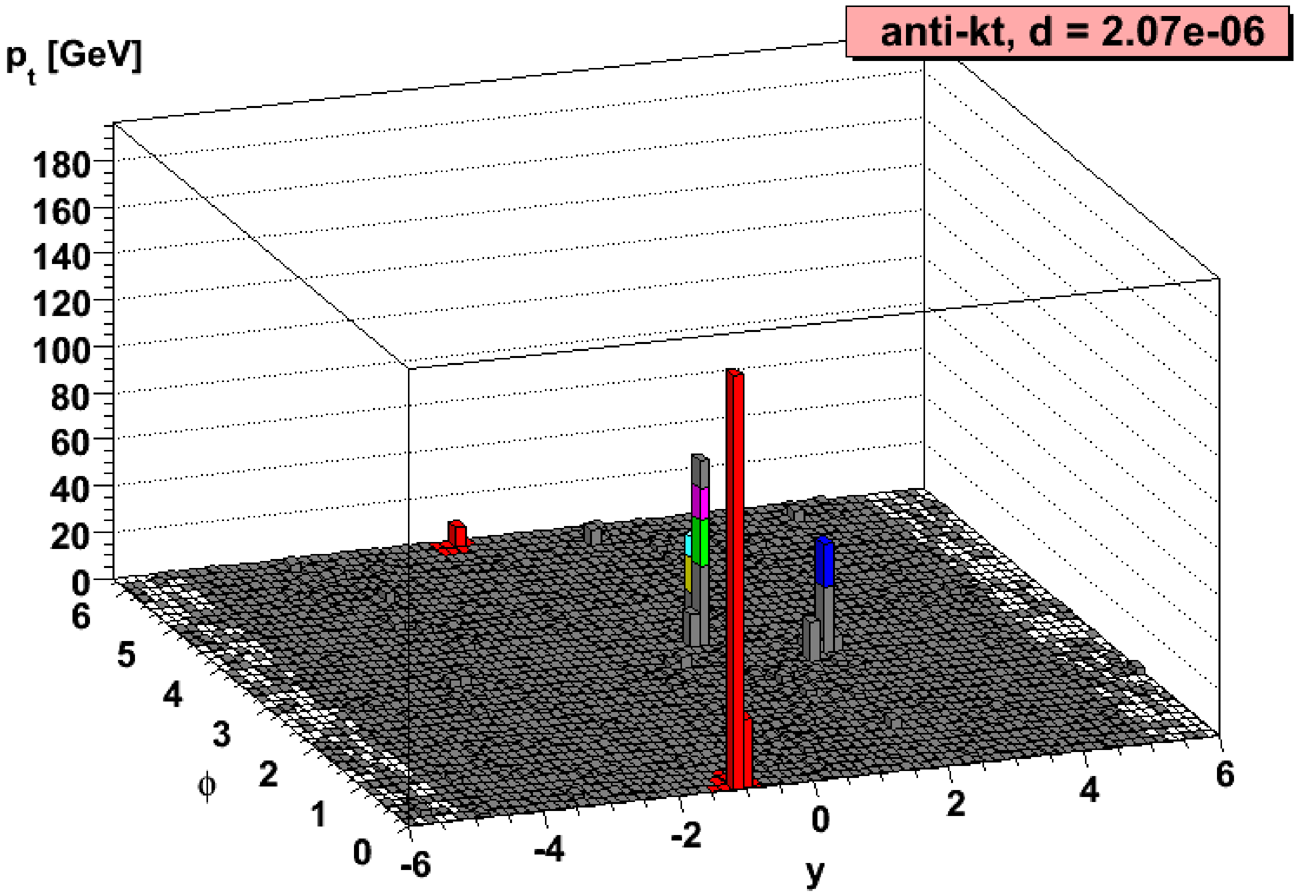
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



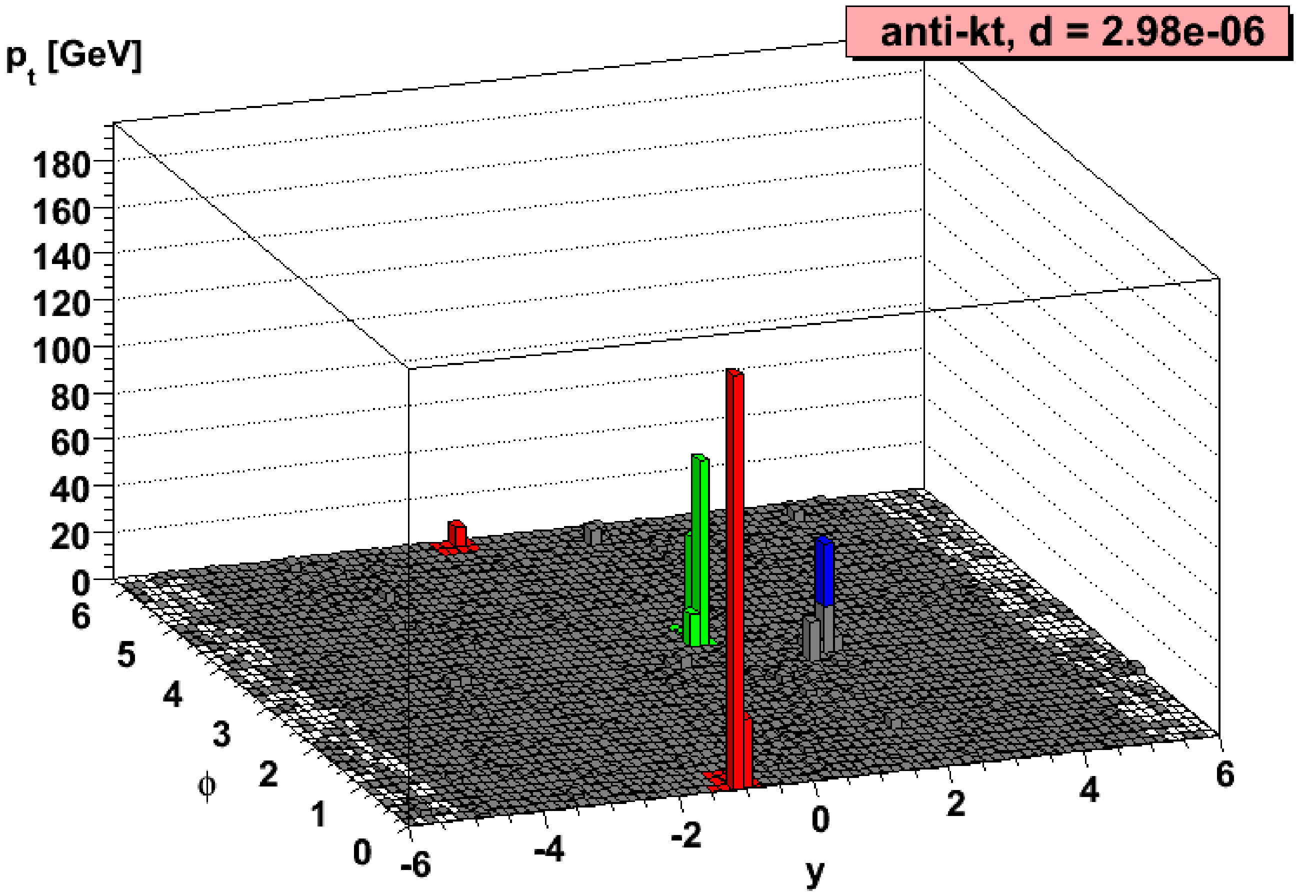
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



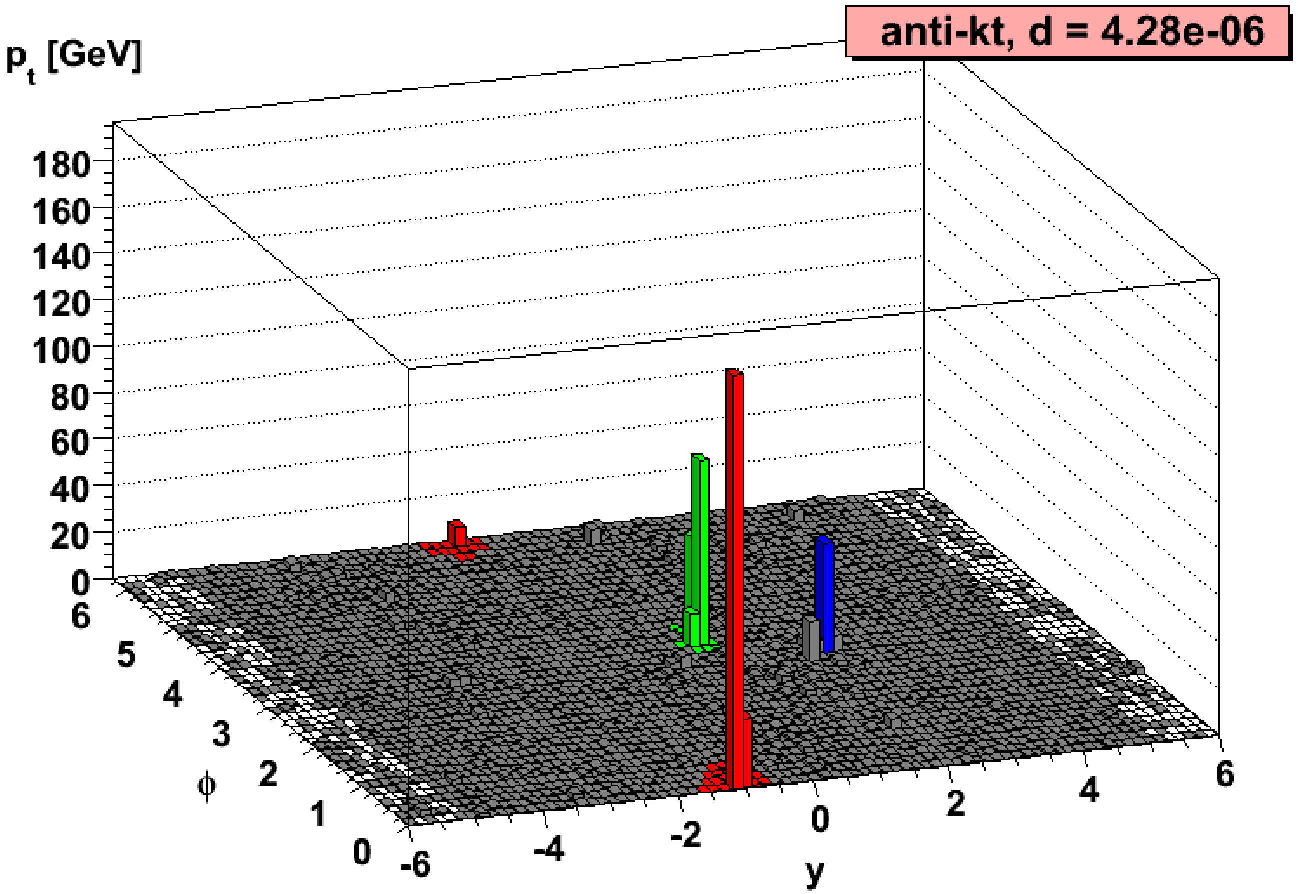
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



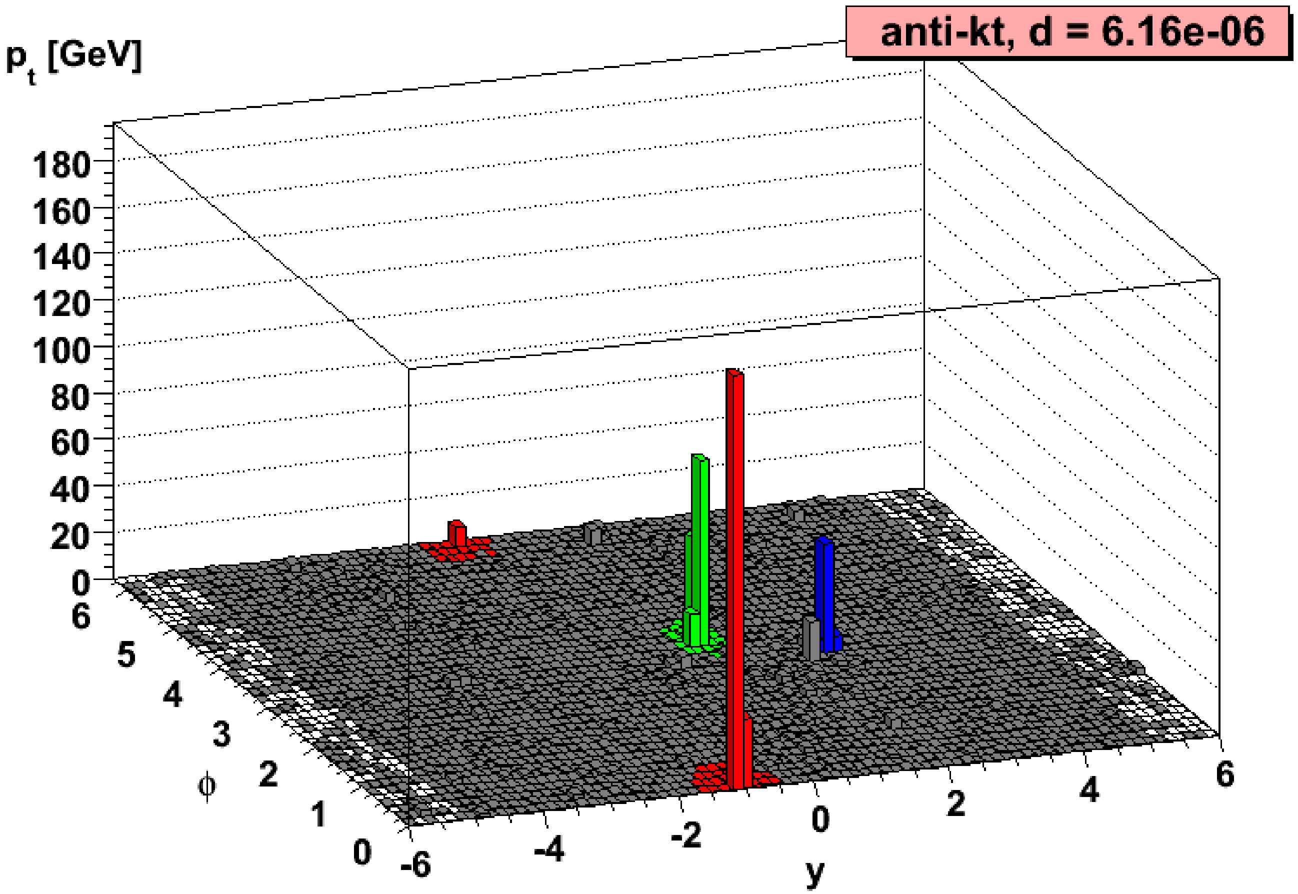
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Clustering grows around hard cores

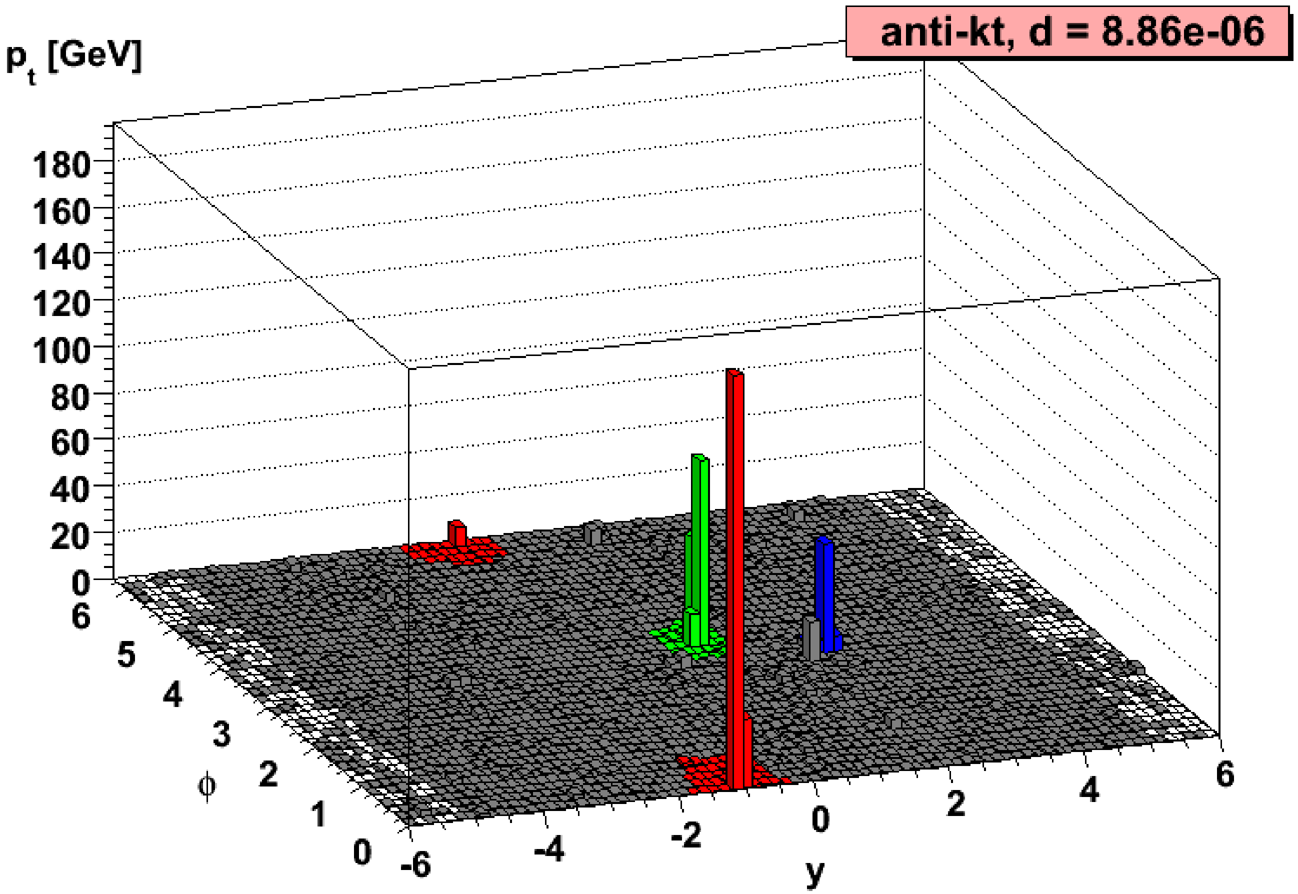
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

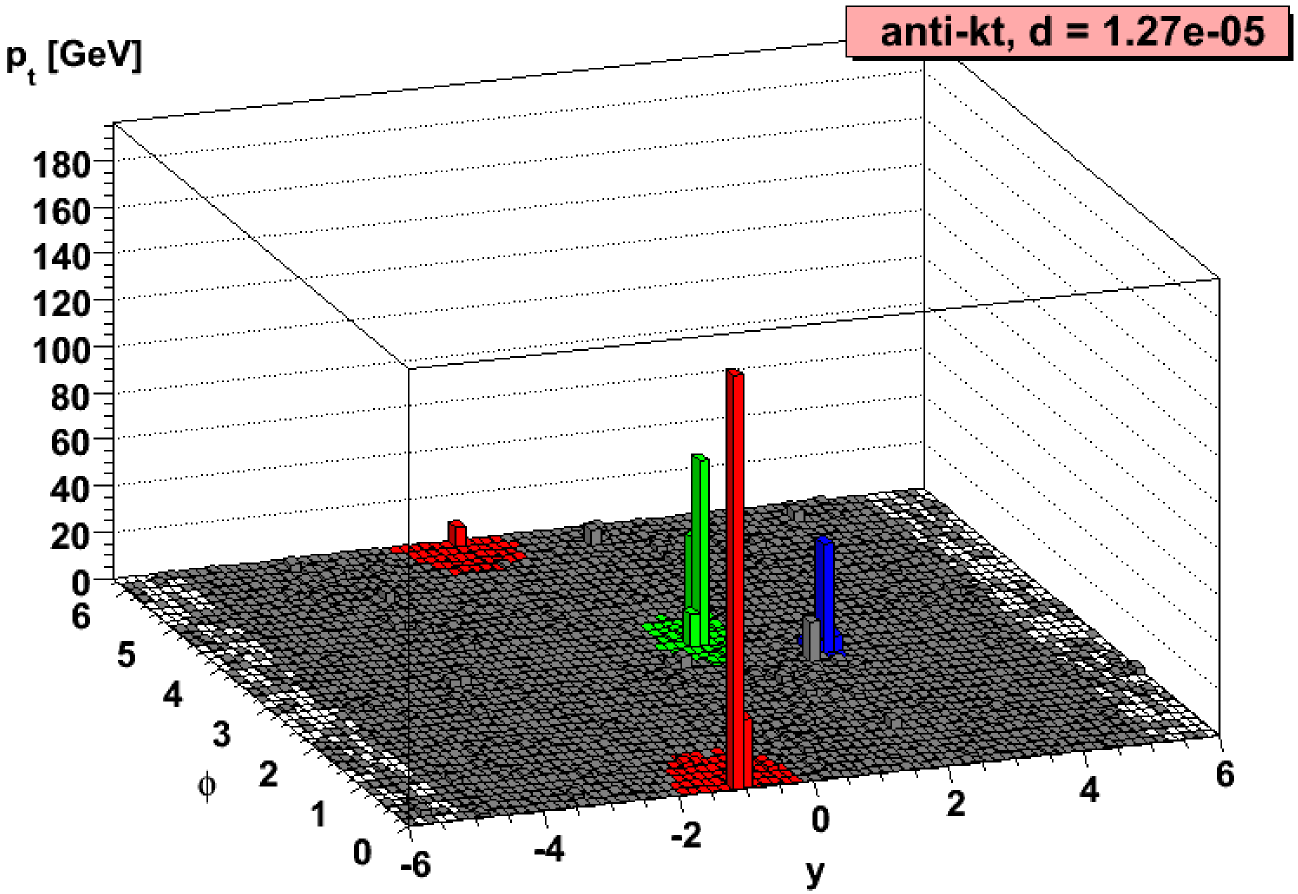
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



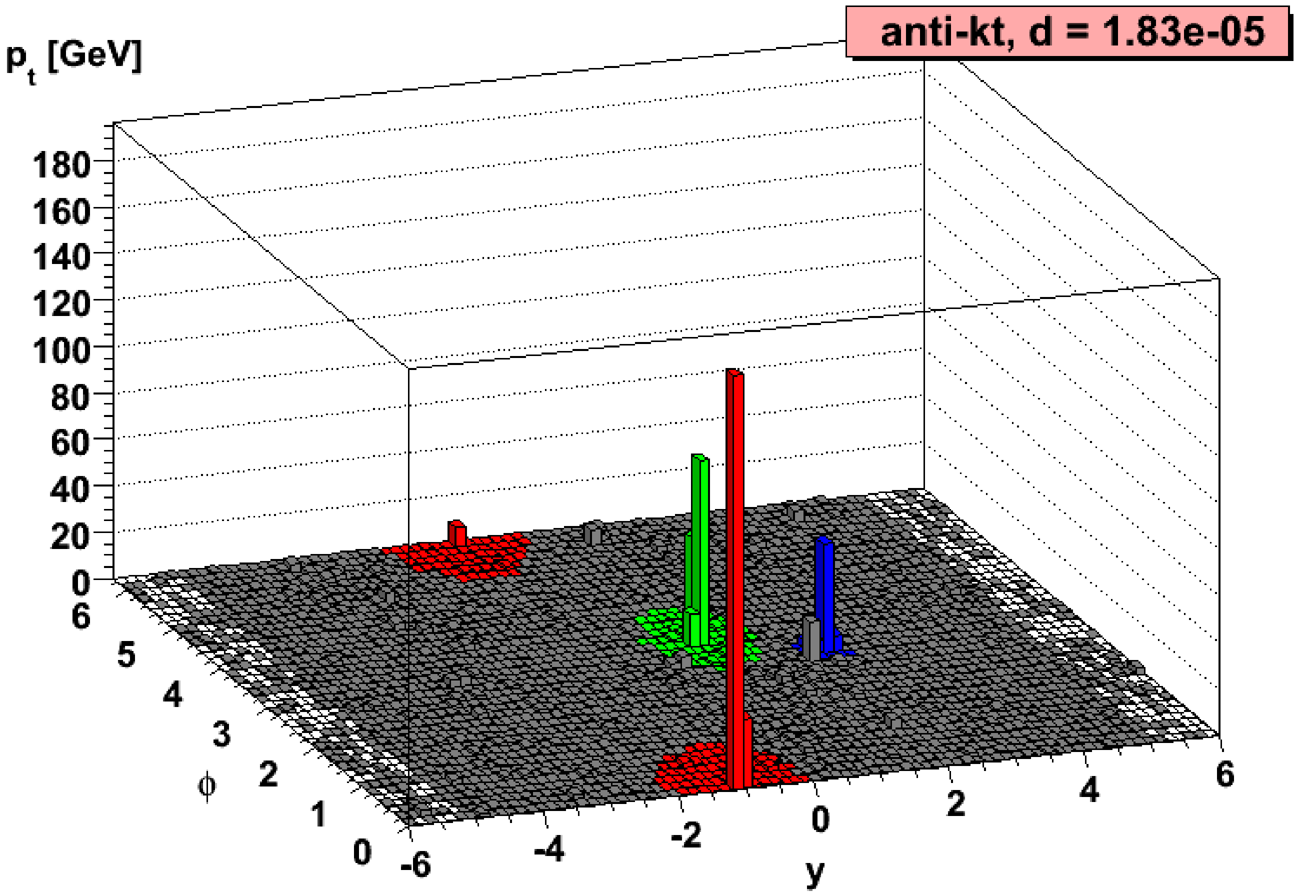
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Clustering grows around hard cores

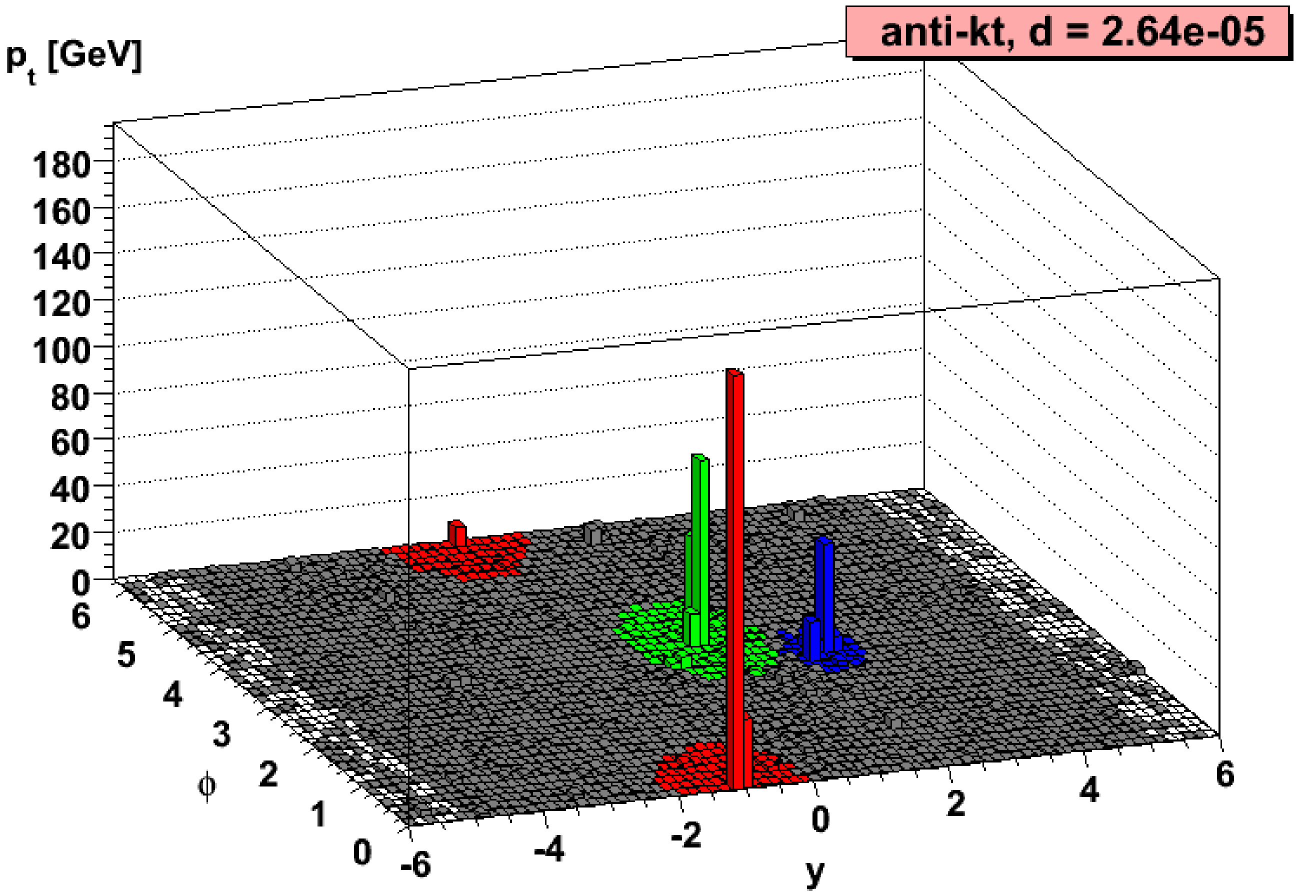
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

Clustering grows around hard cores

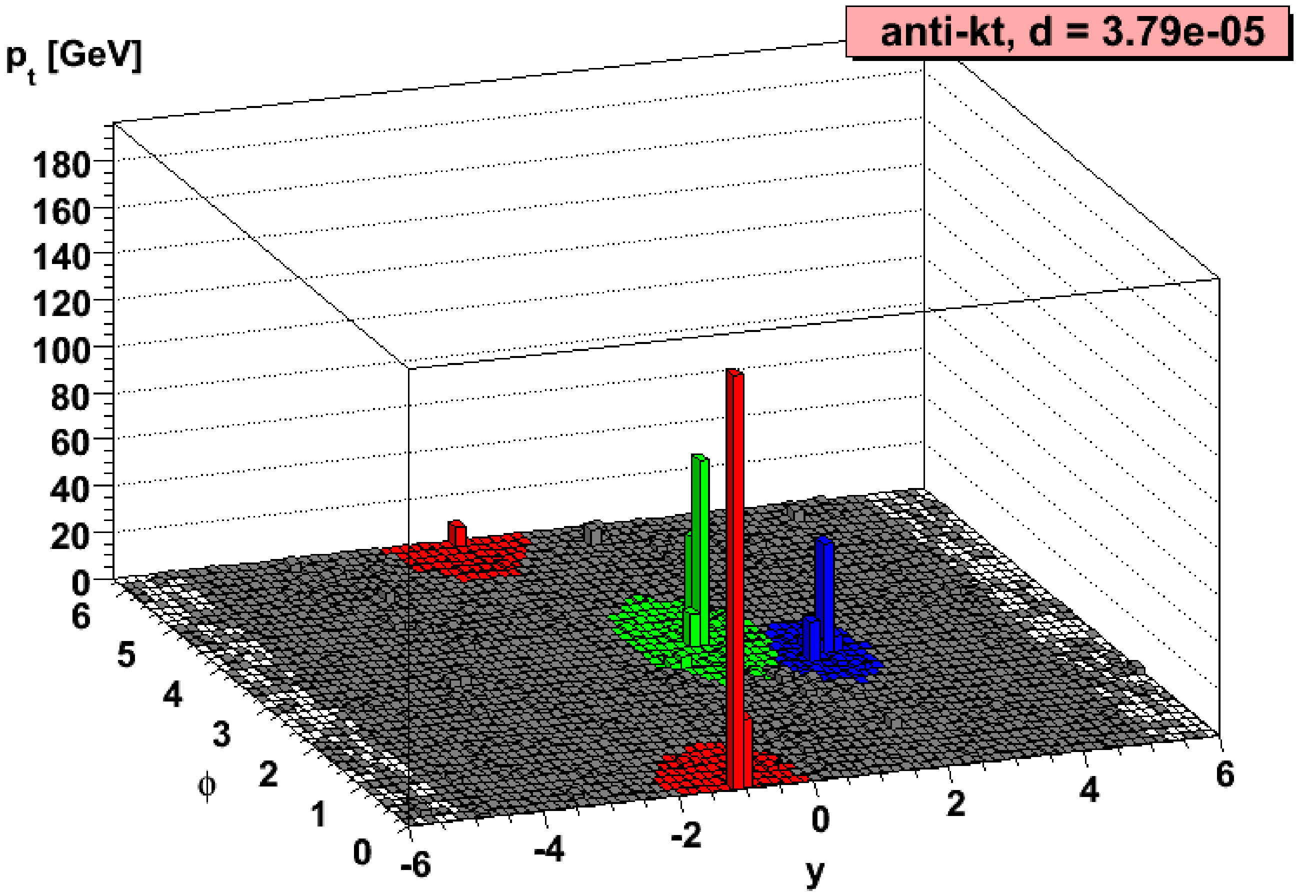
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

Clustering grows around hard cores

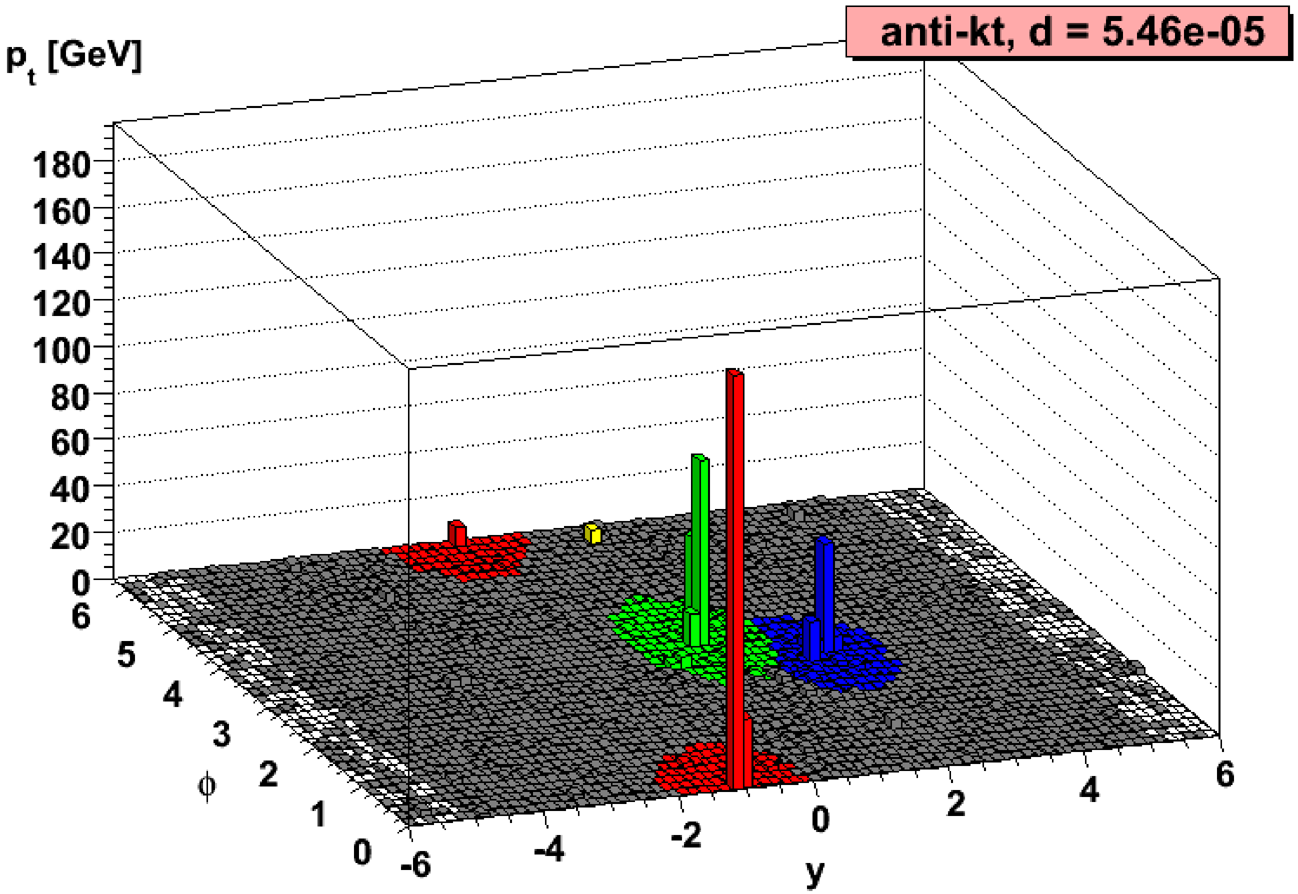
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

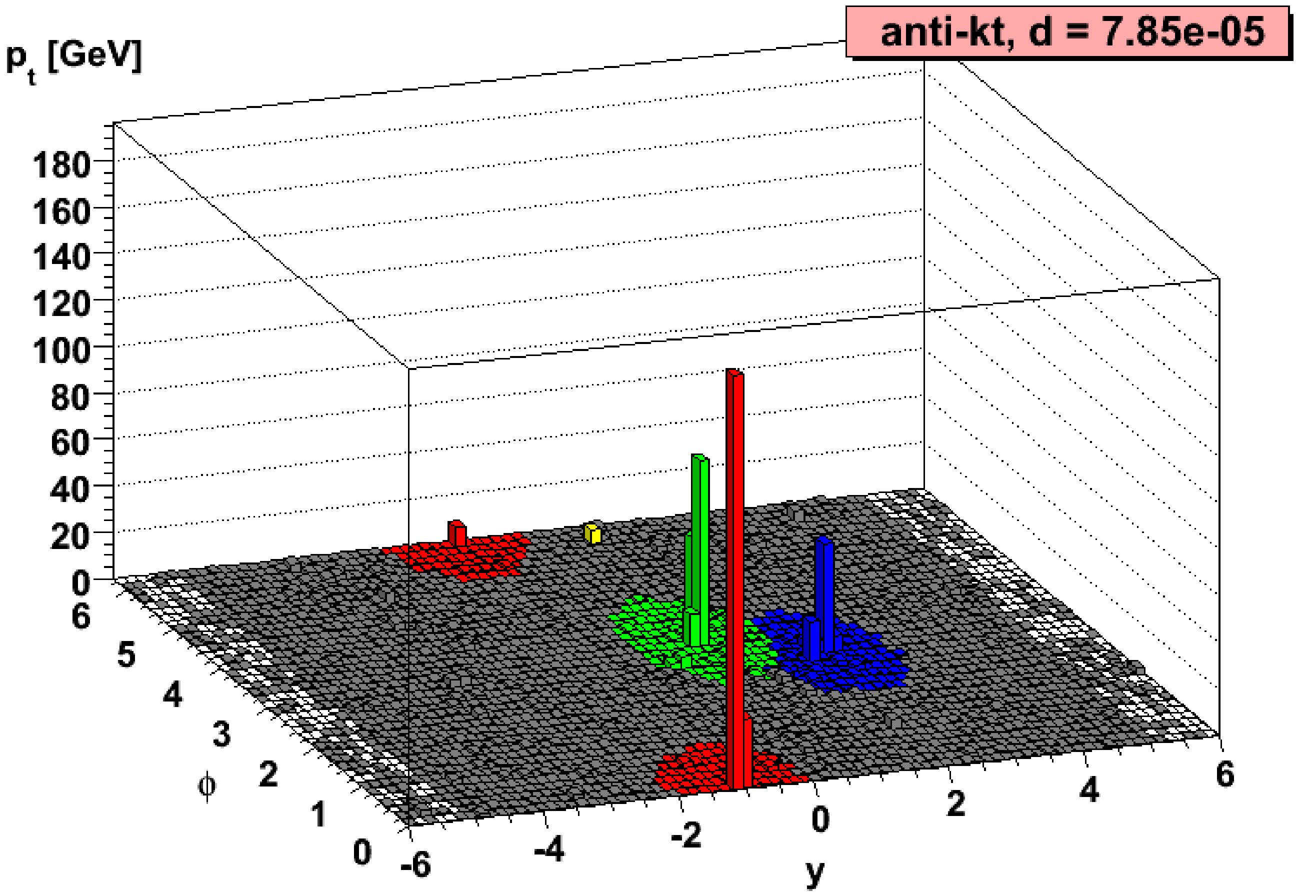
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Clustering grows around hard cores

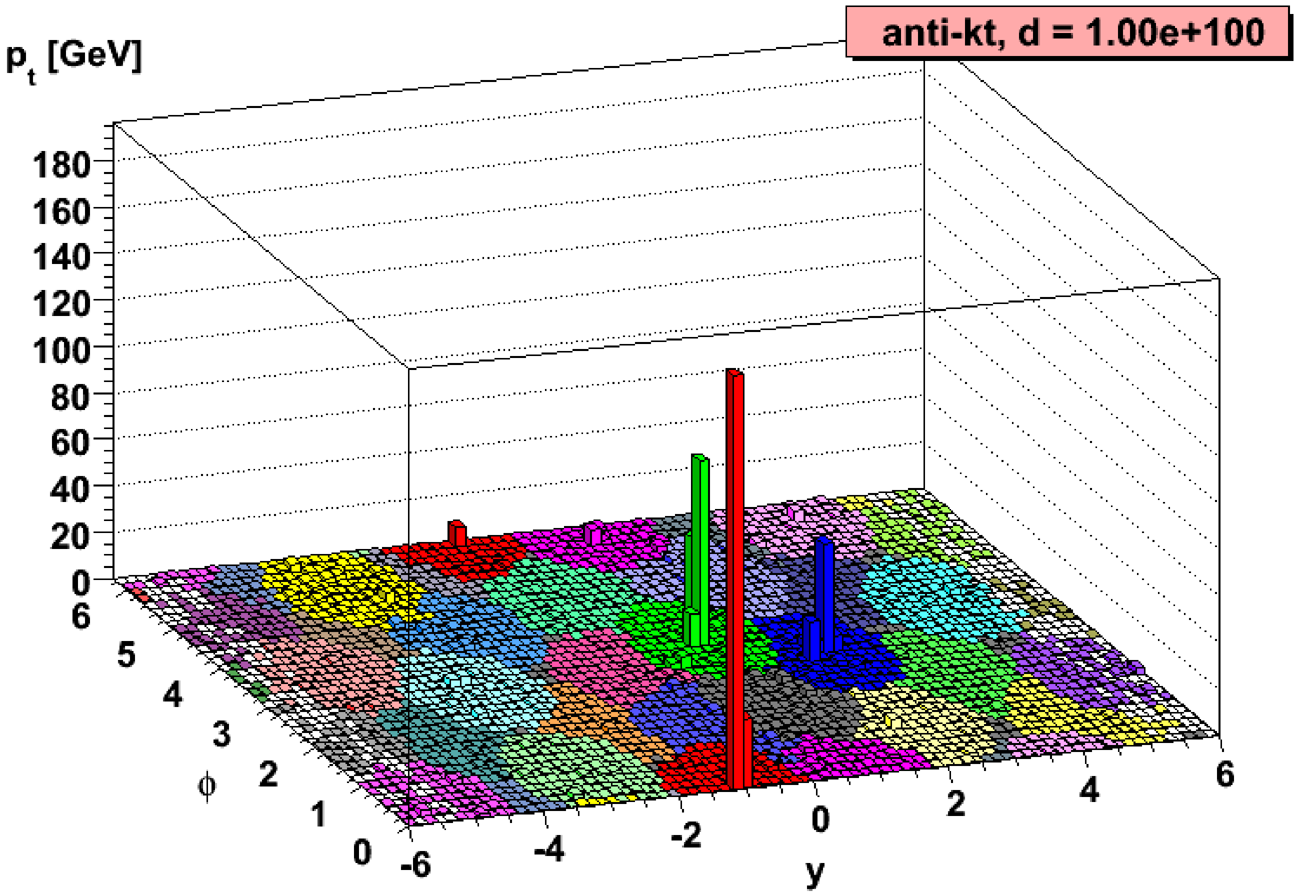
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



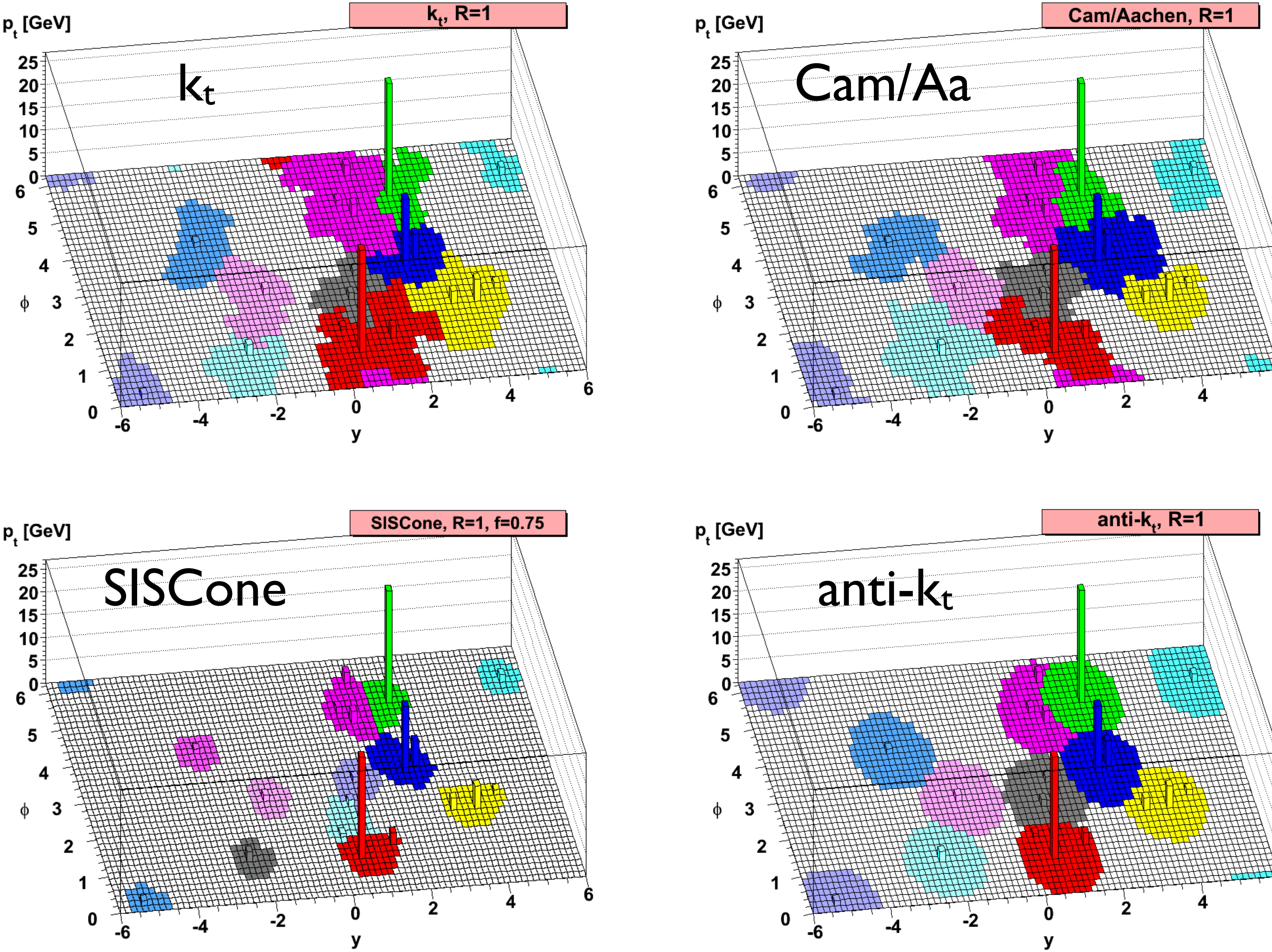
Anti-k_t in action

Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$

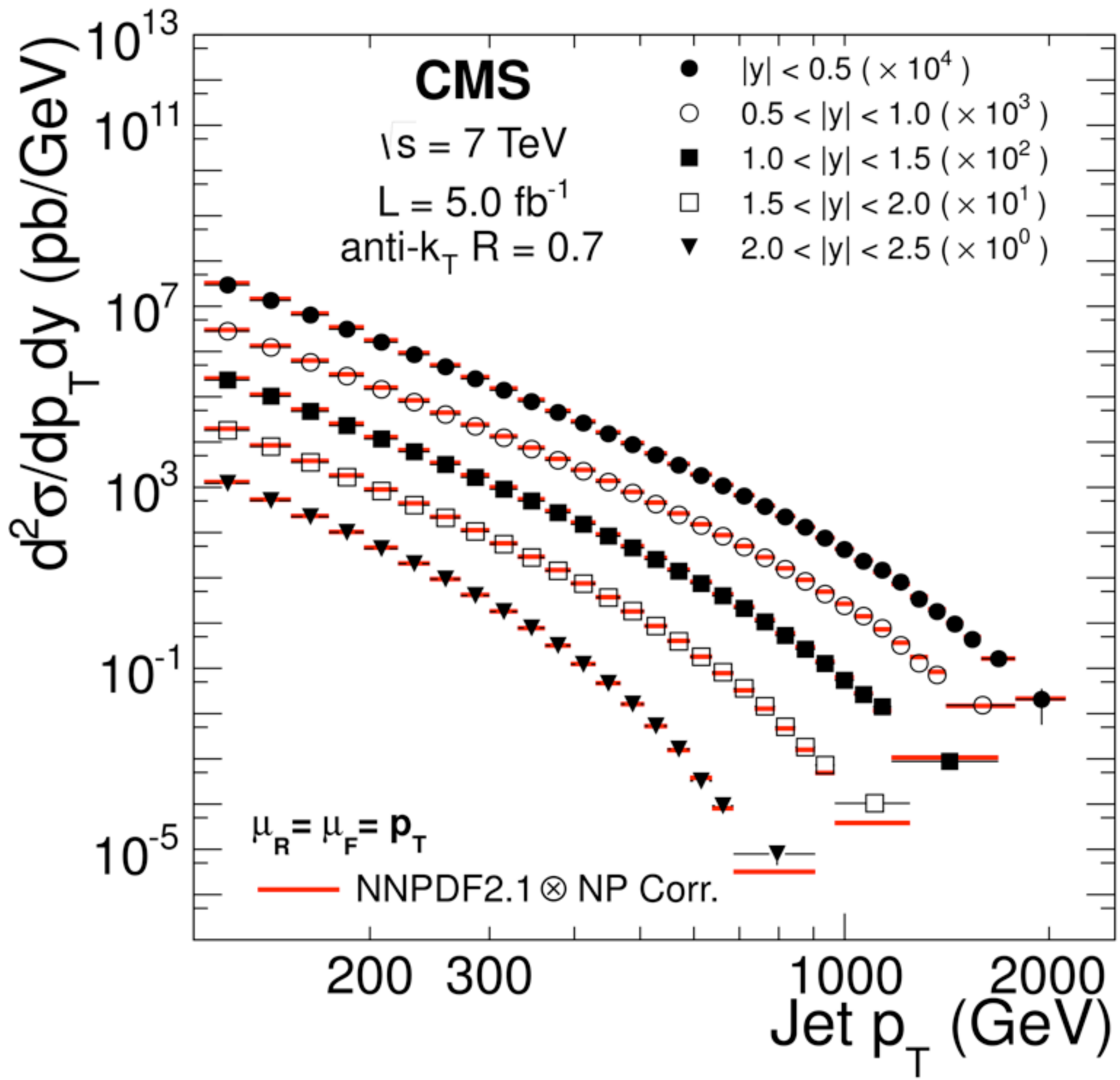


Anti-k_t gives circular jets ("cone-like") in a way that's infrared safe



kt, R=1
kt
Cam/Aachen, R=1
Cam/Aa
SISCone, R=1, f=0.75
SISCone
anti-kt, R=1
anti-kt
pt [GeV]
φ
y

Example of jet observable



Inclusive jet cross section

Excellent theory-data agreement over many orders of magnitude

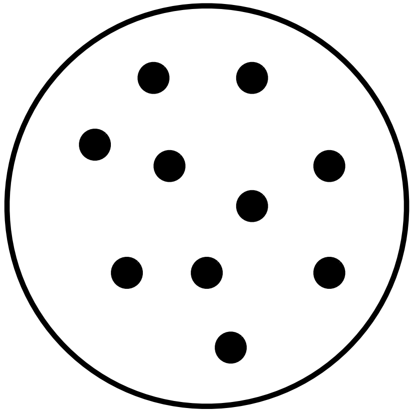
Take home points

- A vast zoology of jet algorithms has been reduced in the past few years to **4 infrared and collinear safe algorithms**
  - All are implemented in an efficient and fast way
  - Of these, **anti-k_t** is used by all the LHC collaborations as their main algorithm for "finding" jets and measuring inclusive cross sections
- The four algorithms have quite different characteristics, which makes them non easily swappable when specific properties are needed for specific tasks. On the other hand, chances are that one can chose the algorithm which is most appropriate for a specific job

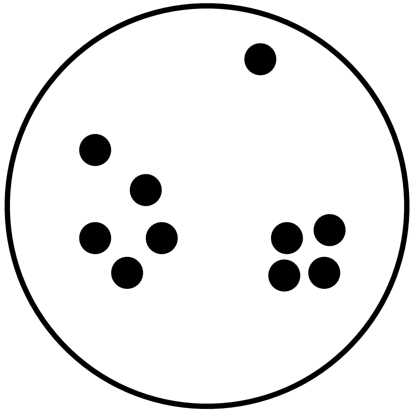
At the end of a jet finding (i.e. clustering) procedure, a jet is a **collection of constituents** to which we assign a 4-momentum

(related to the sum of the 4-momenta of the constituents)

What is the **arrangement** of the constituents inside the jet?



$\neq$

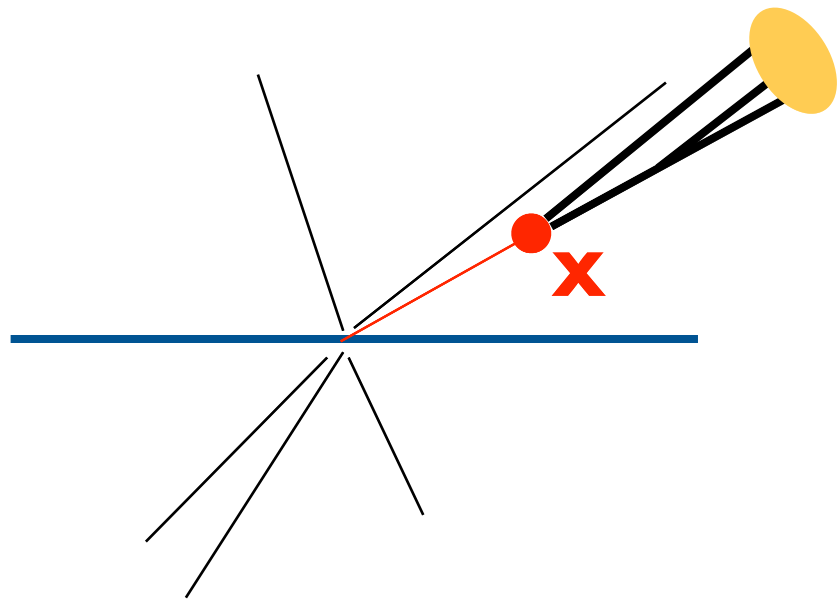


Jet substructure

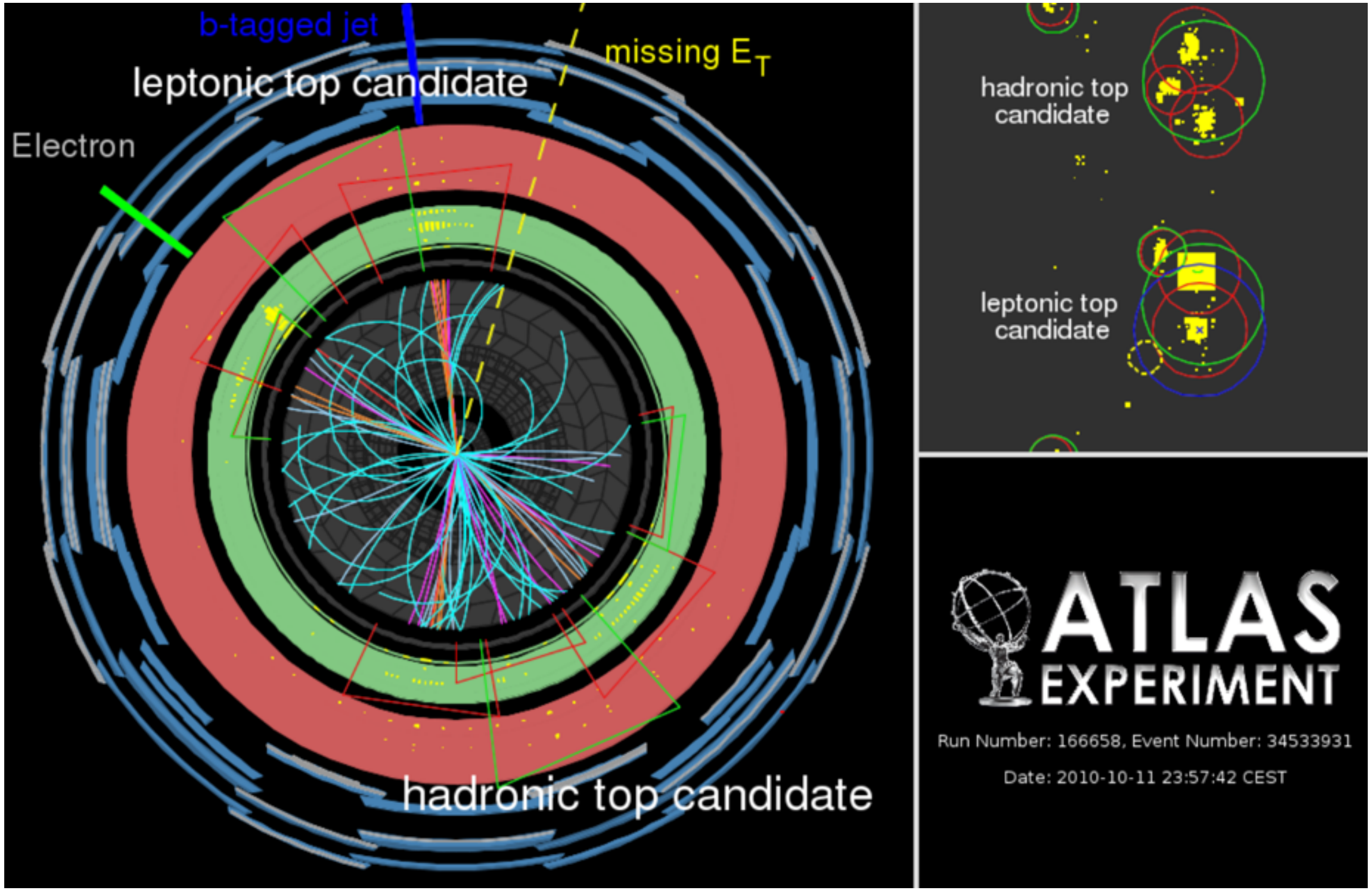
First studied by Mike Seymour in the early '90s to distinguish W jets from QCD jets

Topic revived about fifteen years ago in order to study boosted objects

[Butterworth, Davison, Rubin, Salam, 0802.2470]

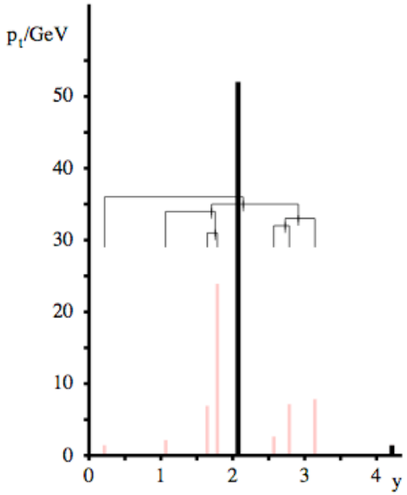


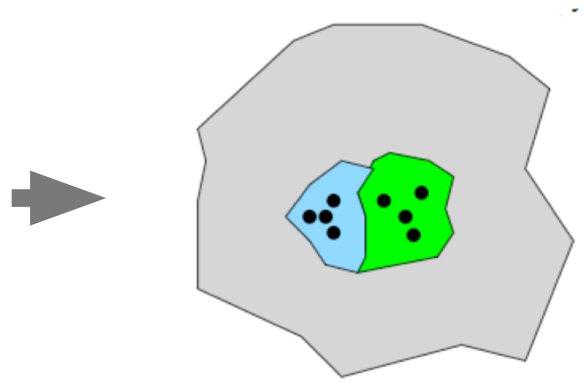
Jet substructure



The past fifteen years have seen en explosion in jet substructure studies, i.e. **how radiation is arranged within jets**, and what it can tell us

Jet substructure

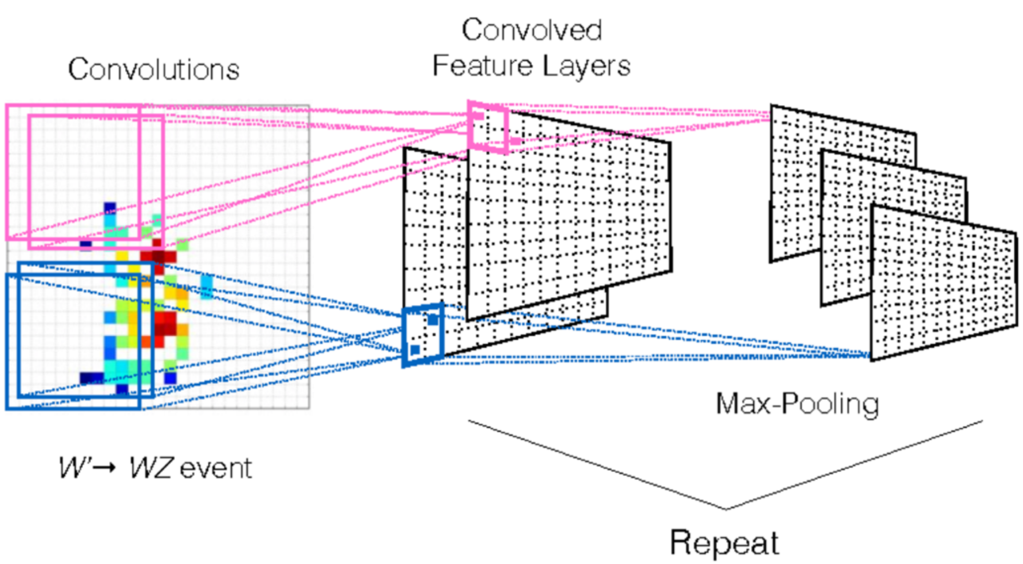




Jet declustering

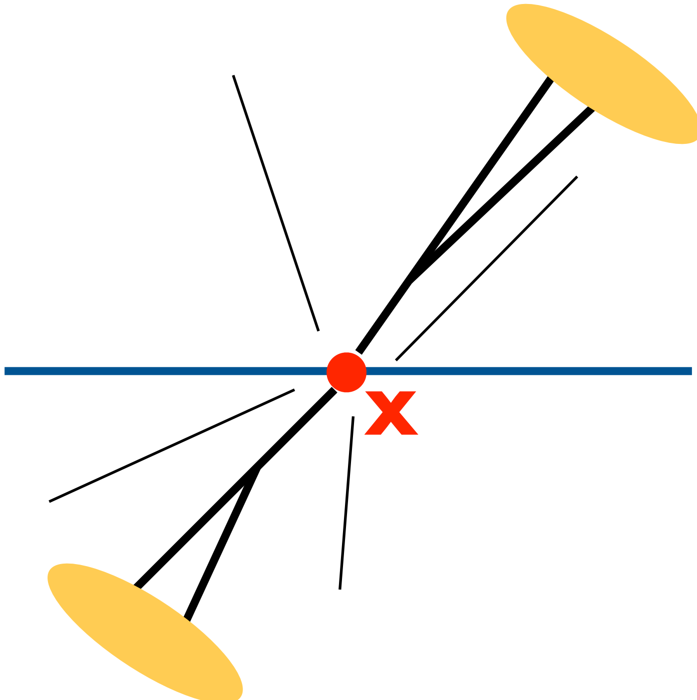
Jet shapes

(calculate a function from radiation distribution

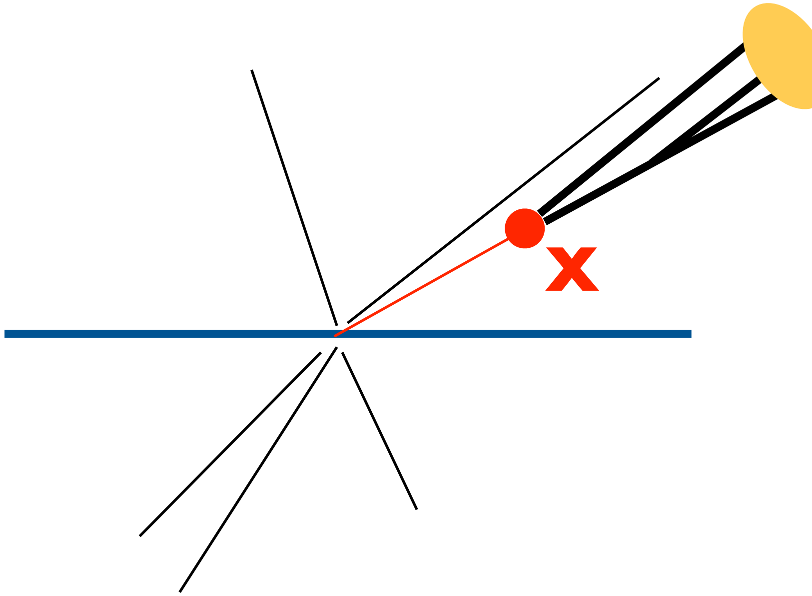


Machine learning

Why boosted objects



Heavy particle X at **rest**



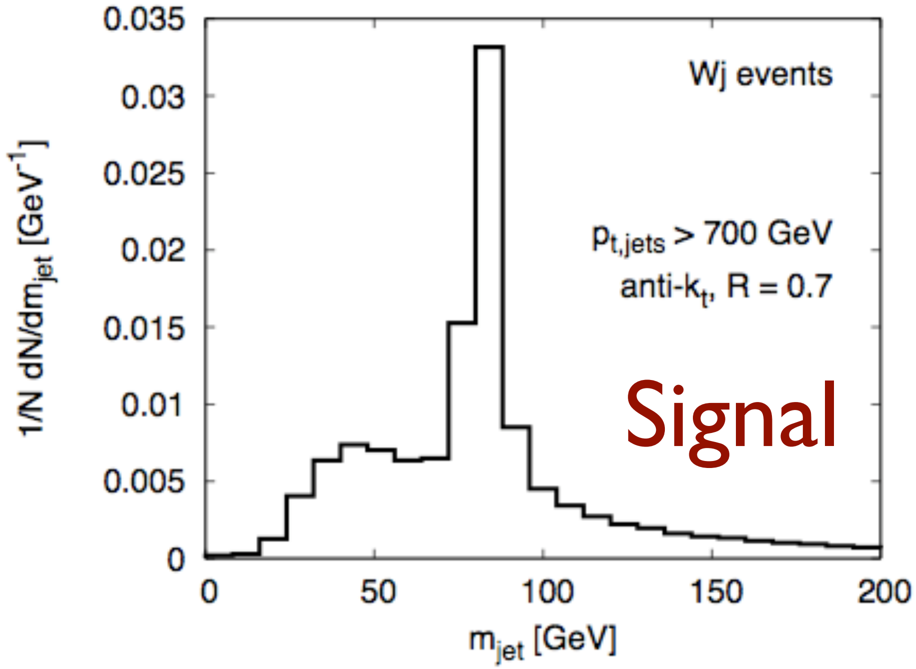
Boosted heavy particle X

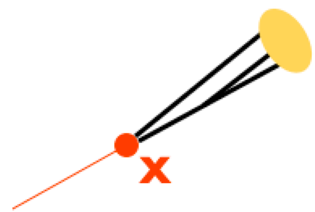
Easy to resolve jets and calculate invariant mass, but signal very likely swamped by background (eg $H \to bb$ v. $tt \to WbWb$)

Cross section very much reduced, but acceptance better and some backgrounds smaller/ reducible

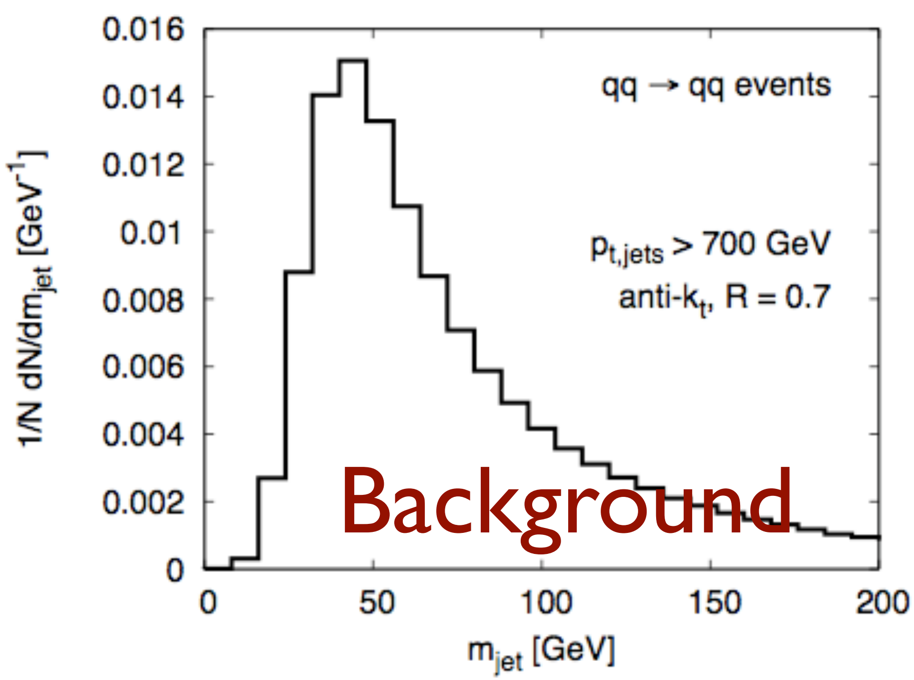
Mass of a single jet

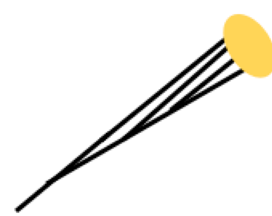
G. Salam





A heavy object decaying into a single jet naturally gives it a mass...



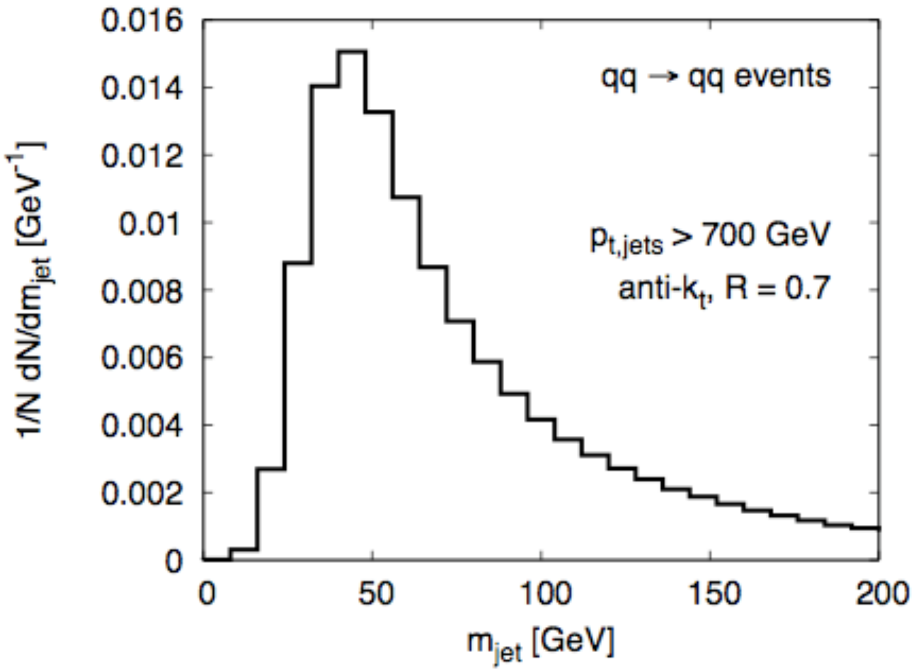


... but pure QCD jets can be massive too:

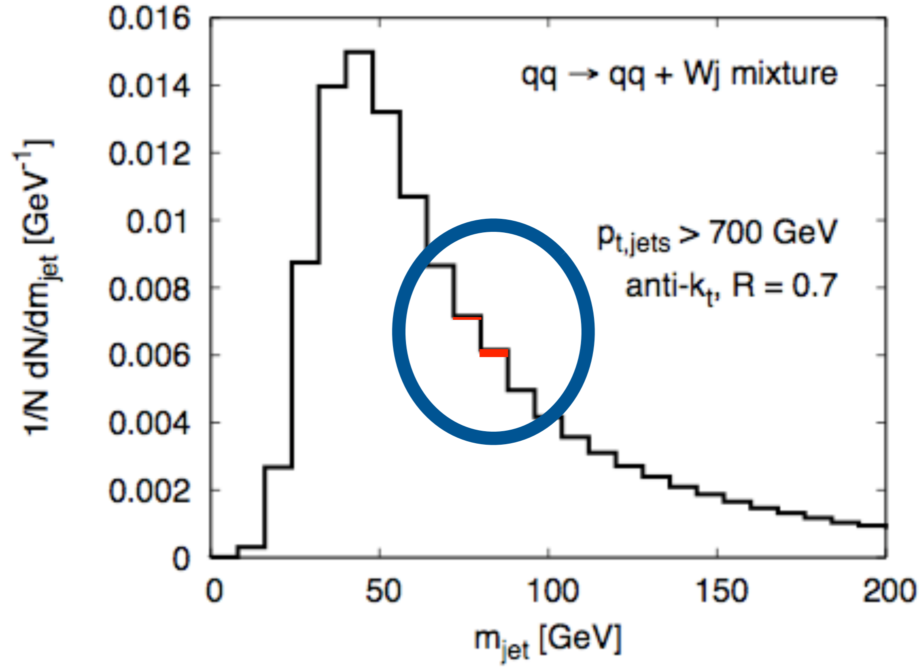
$$\frac{dN}{d\ln m} \sim \alpha_s \ln \frac{p_t R}{m} \times \text{Sudakov}$$

Mass of a single jet

Summing 'signal' and 'background' (with appropriate cross sections) shows how much the background dominates



Background only

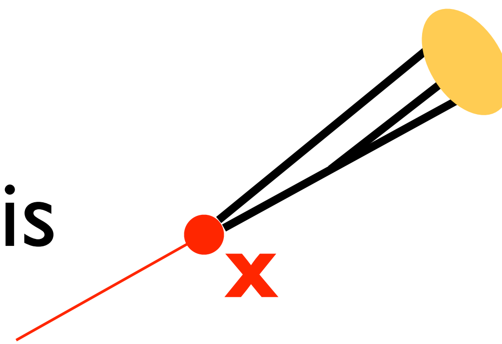


Signal + background

Practically identical

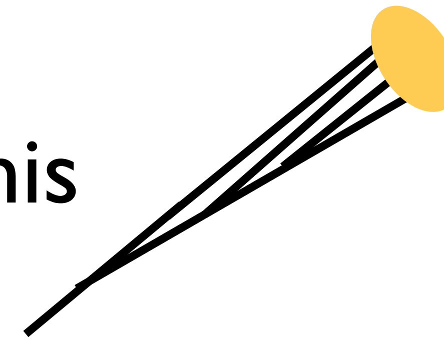
This means that one can't rely on the invariant mass only.
An appropriate strategy must be found to reduce the background and enhance the signal

How to tell this



Decay of a heavy (boosted) object

from this



Light parton fragmentation

?

Tagging and Grooming

- The substructure of a jet can be exploited to
  - ***tag*** *a particular structure inside the jet, i.e. a massive particle*
    - *First examples: Higgs (2-prong decay), top (3-prong decay)*
  - *remove background contamination from the jet or its components, while keeping the bulk of the perturbative radiation (often generically denoted as* ***grooming****)*
    - *First examples: filtering, trimming, pruning*

Nomenclature

Groomer

- procedure that always returns an output jet
  (i.e. it only subtracts uncorrelated 'UE/pileup' radiation from it. This is used to "clean" the jets from radiation largely unrelated to the fragmentation of the particle of interest)

Tagger

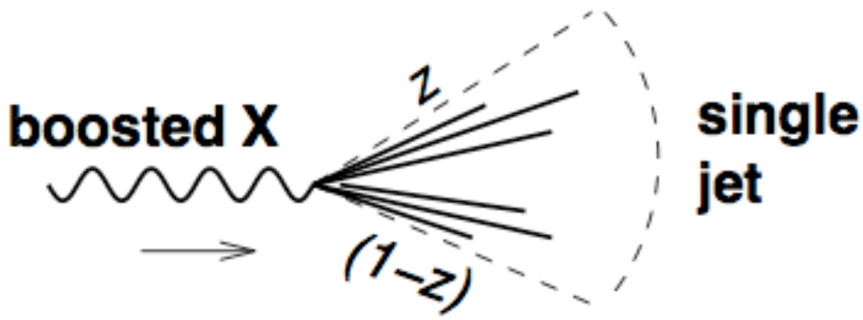
- procedure that might not return an output jet
  (i.e. it either tags a heavy particle originating the jet or returns zero. This is used to identify a specific particle originating the jet.)

In practice, this classification is not always followed.
In some cases it also denoted a 'tagger' a procedure that rejects background jets more often than signal jets

Why substructure

Scales: $m \sim 100$ GeV, $p_t \sim 500$ GeV

(e.g. electroweak particle from decay of ~ 1TeV BSM particle)



$$R \gtrsim \frac{m}{p_t} \frac{1}{\sqrt{z(1-z)}}$$

- need small R ($< 2m/p_t \sim 0.4$) to resolve two prongs
- need large R ($>\sim 3m/p_t \sim 0.6$) to cluster into a single jet

Possible strategies

- Use large R, get a single jet : background large
- Use small R, resolve the jets : what is the right scale?
  - Also: small jets lead to huge combinatorial issues

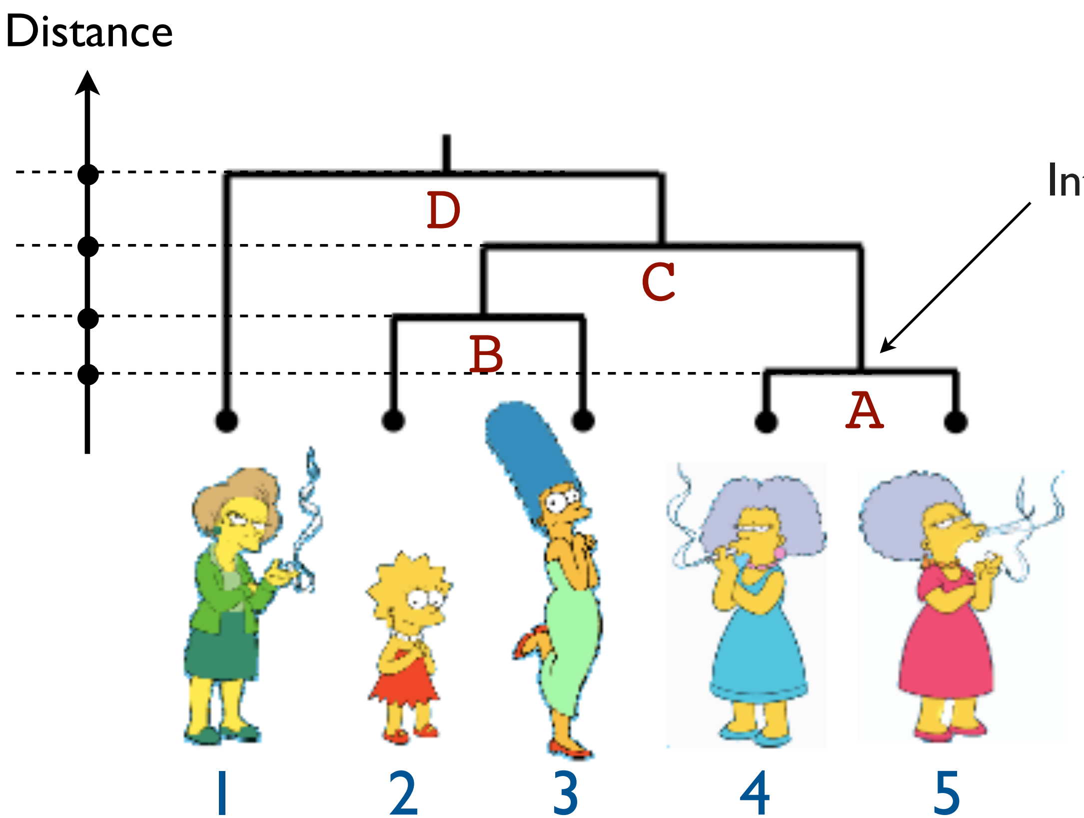
Let an algorithm find the 'right' substructure

What jets to use for substructure?

Different jet algorithms will give different ‘pictures’ of what’s inside a jet

Dendrogram

Used to represent graphically the sequence of clustering steps in a sequential recombination algorithm



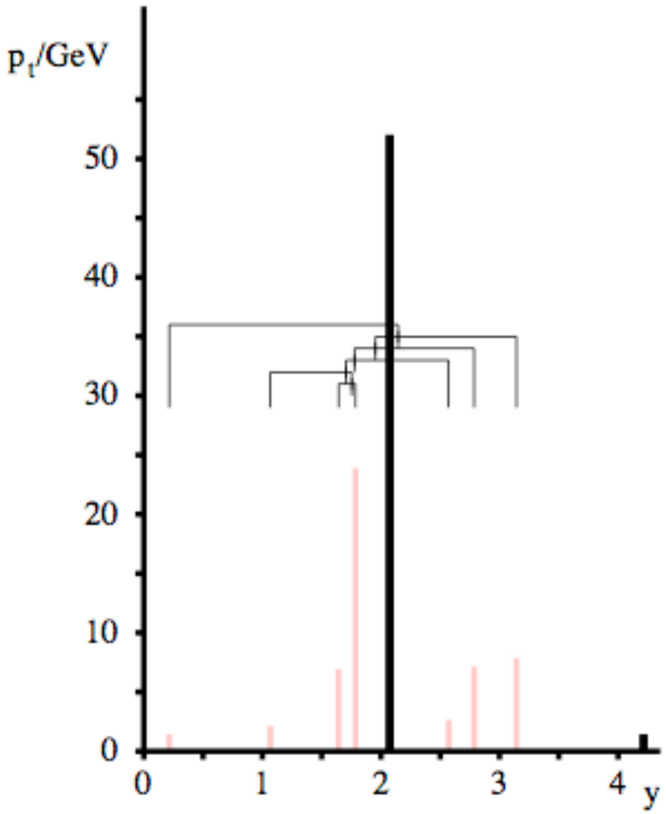
Distance between two objects is given by the **height** of the lowest internal node that they share.

Order of clustering here is A, B, C, D

The **clustering sequence** is 4-5 (A), 2-3 (B), 23-45 (C), 1-2345 (D) anti-kt

Hierarchical substructure

anti-k_t algorithm

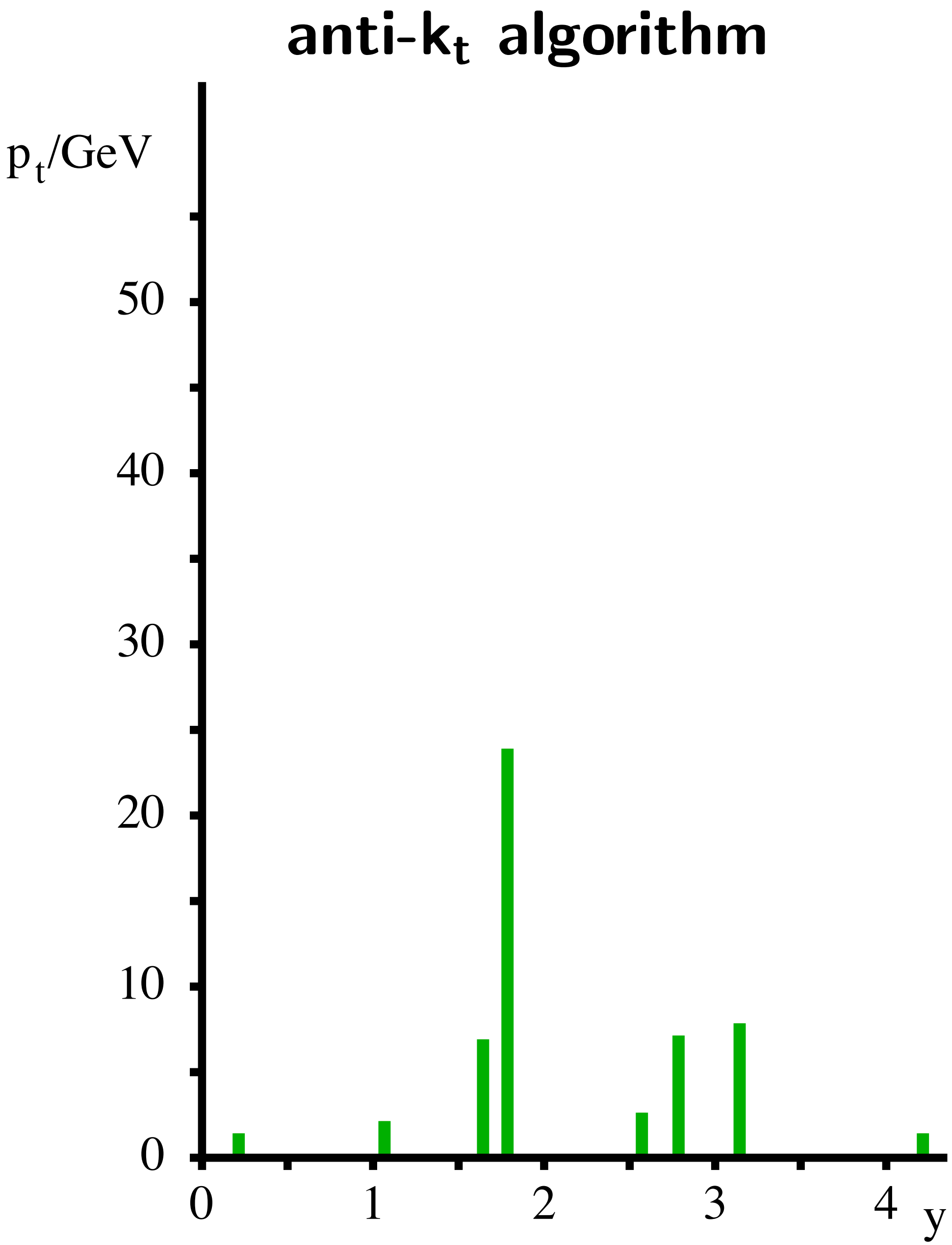


Anti-k_t distance measure

$$d_{ij} = \min\left(\frac{1}{p_{ti}^2}, \frac{1}{p_{tj}^2}\right)\frac{\Delta y^2 + \Delta\phi^2}{R^2}$$

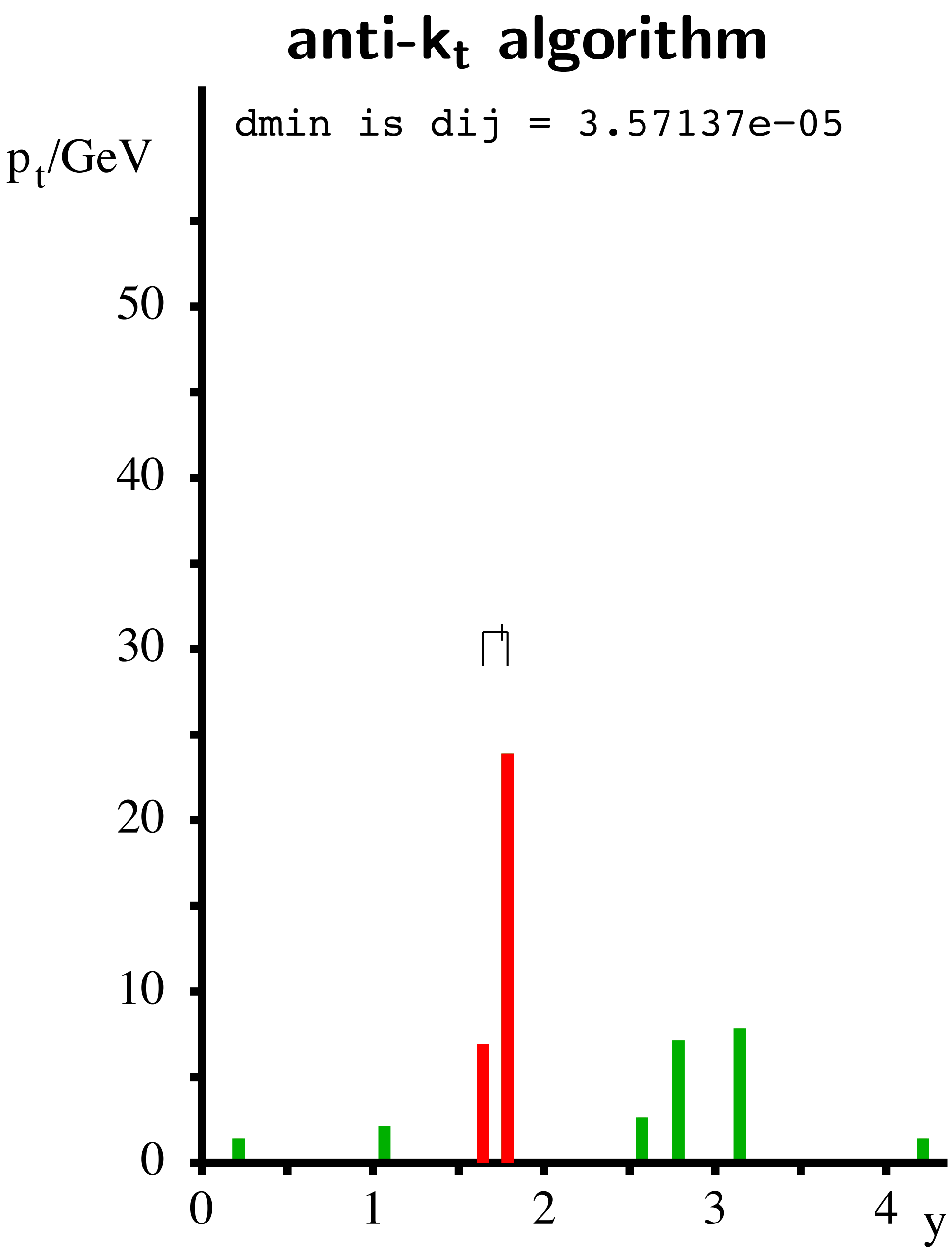
Cluster by merging
to the **hardest/closest** particle

Identifying jet substructure: try out anti-k_t



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

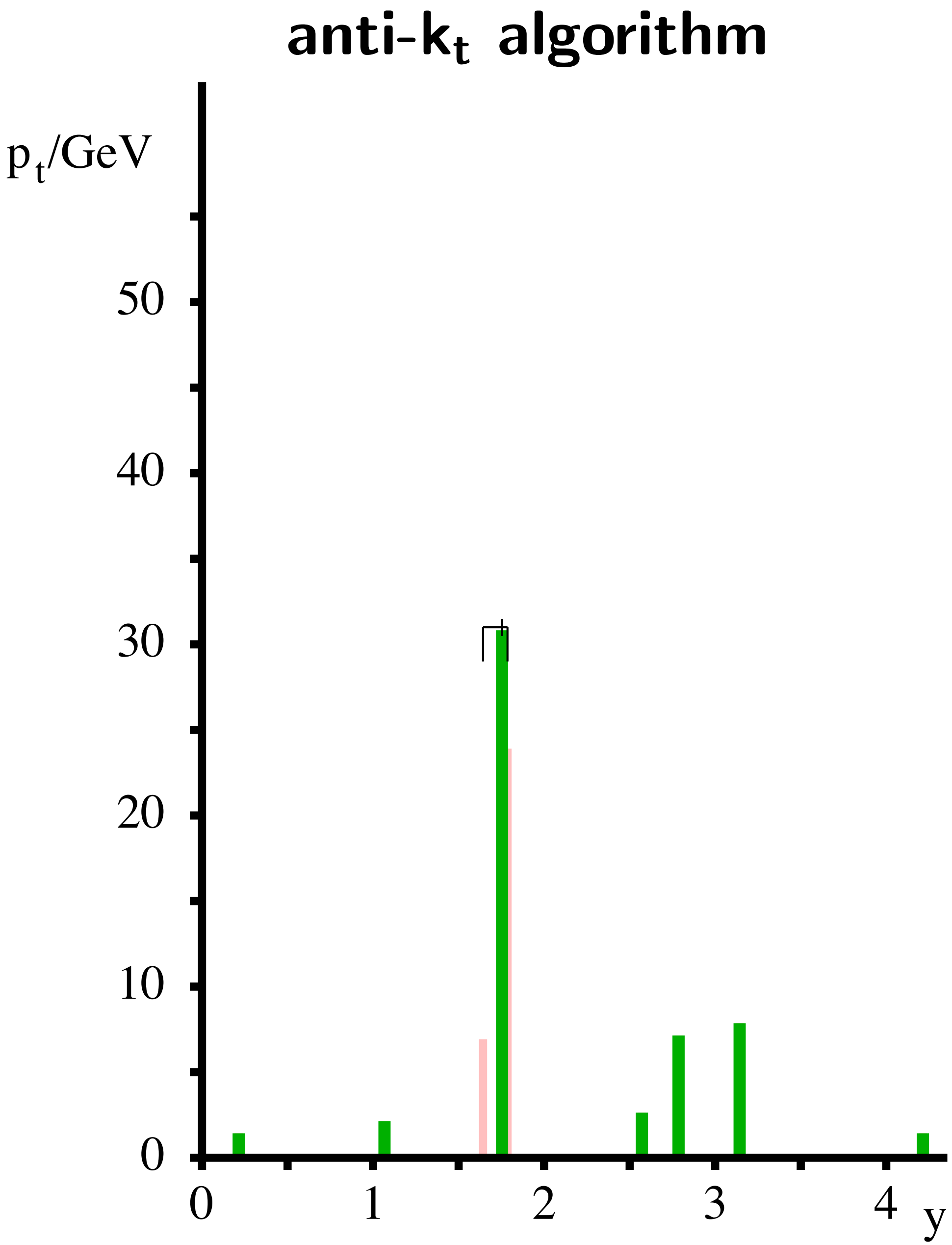
This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

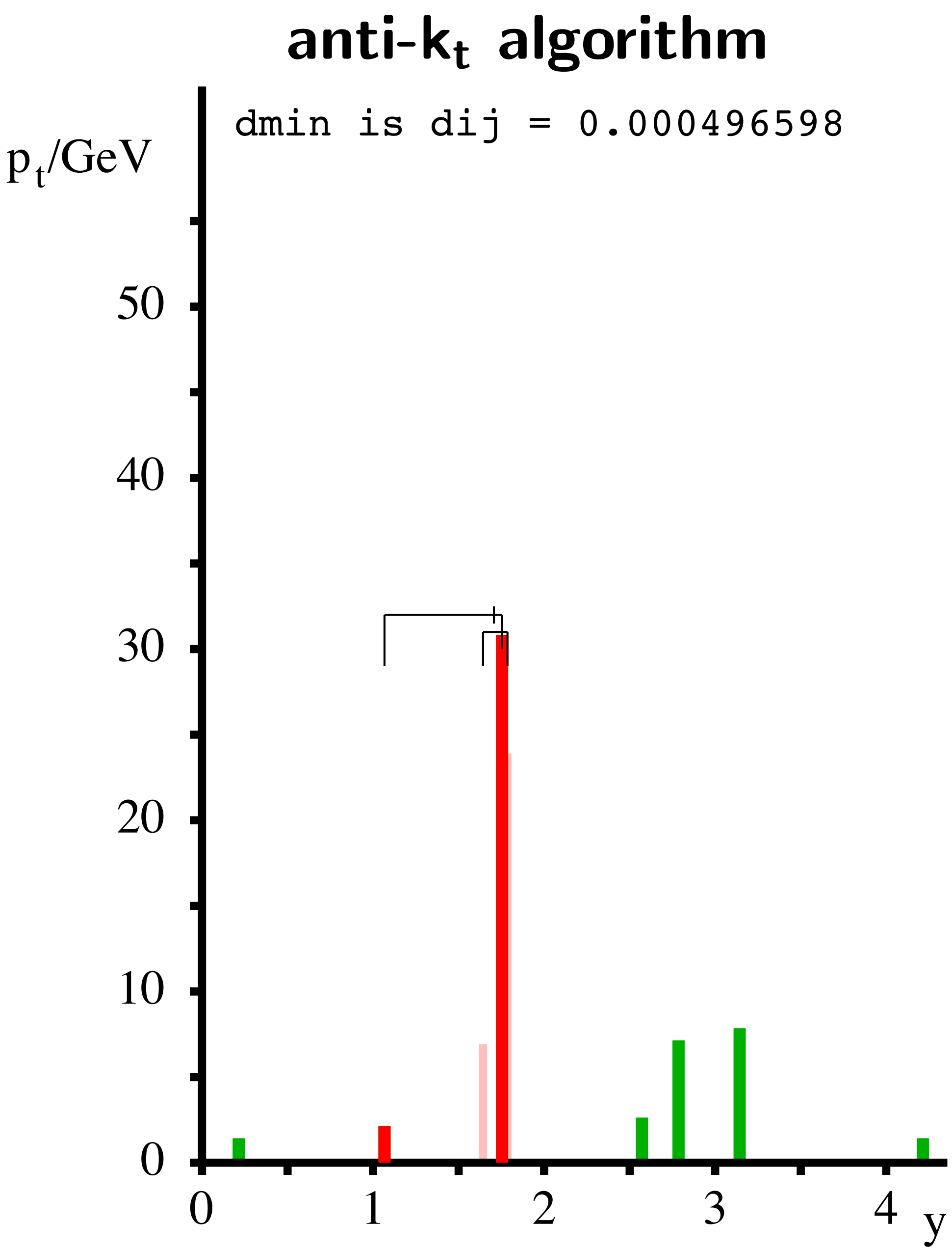
Identifying jet substructure: try out anti-k_t



How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

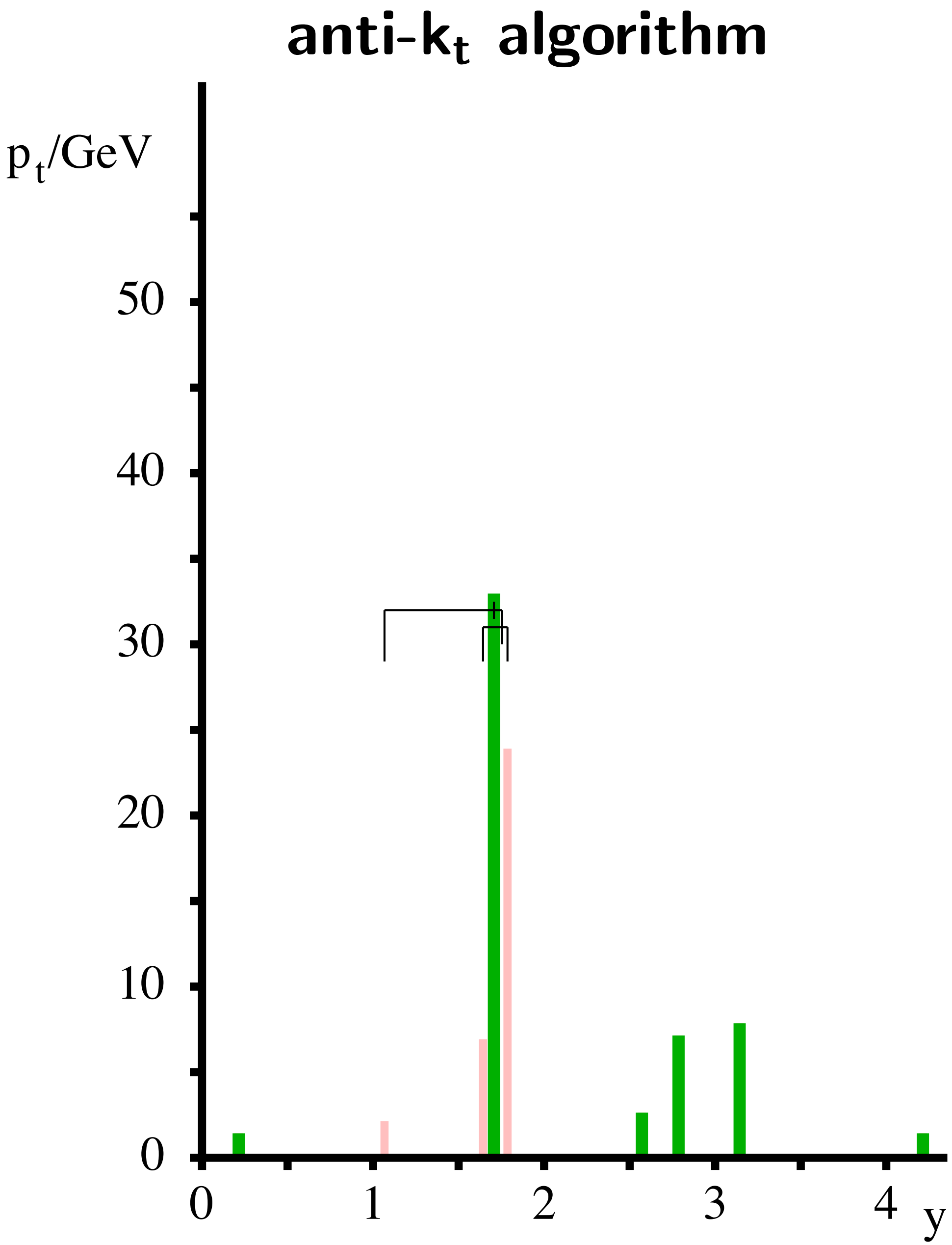
Identifying jet substructure: try out anti-k_t



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

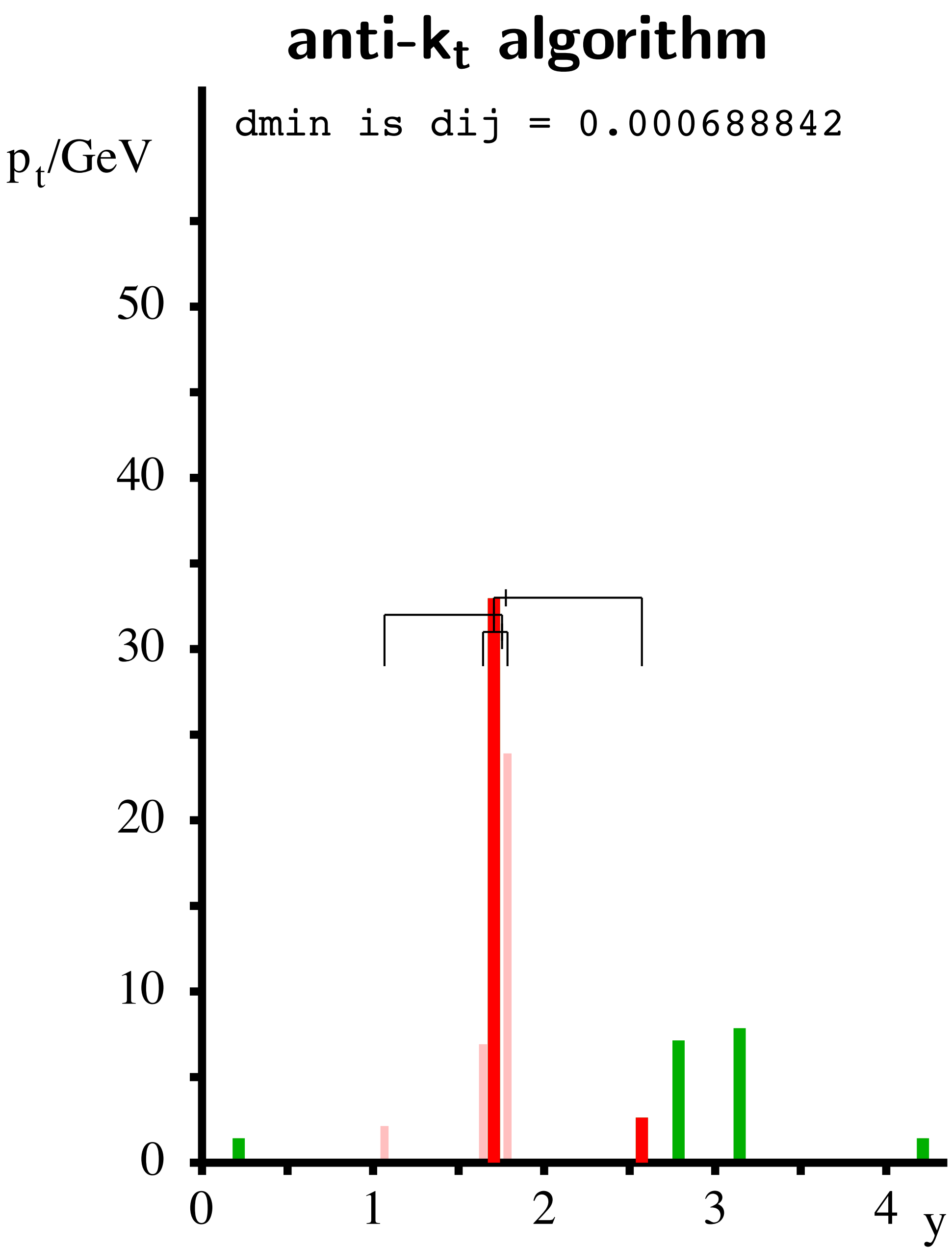
This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Identifying jet substructure: try out anti-k_t



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

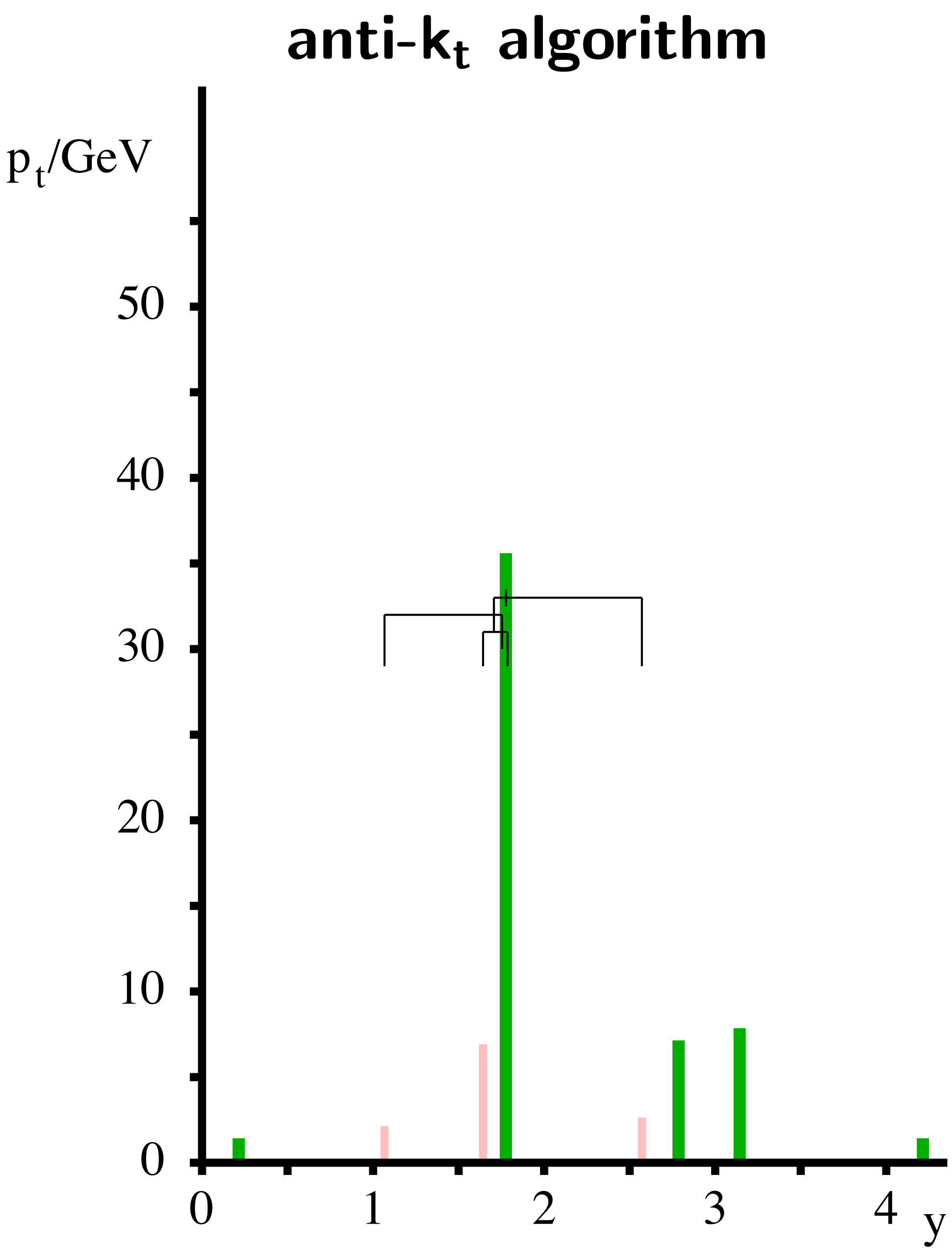
This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

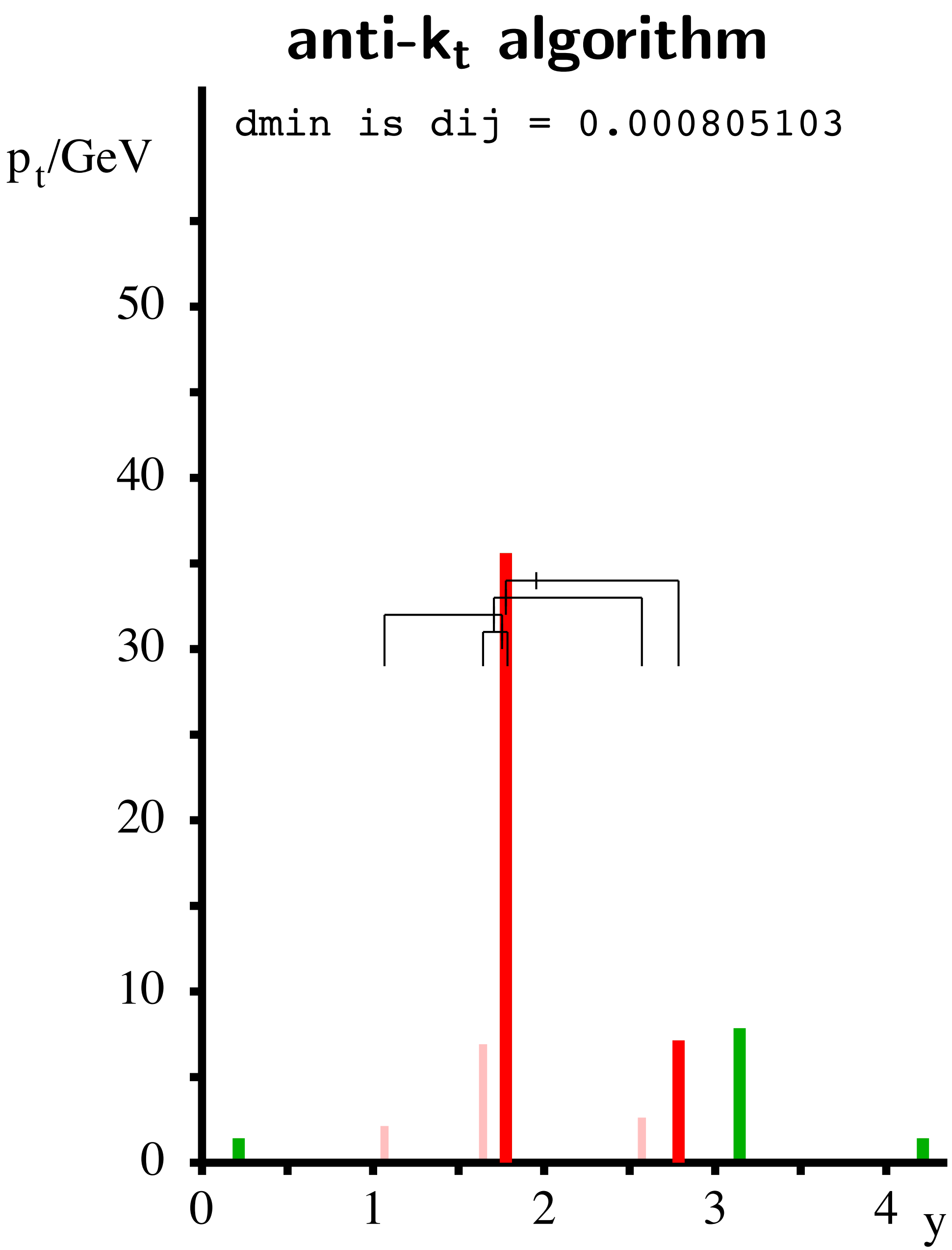
Identifying jet substructure: try out anti-k_t



How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Identifying jet substructure: try out anti-k_t

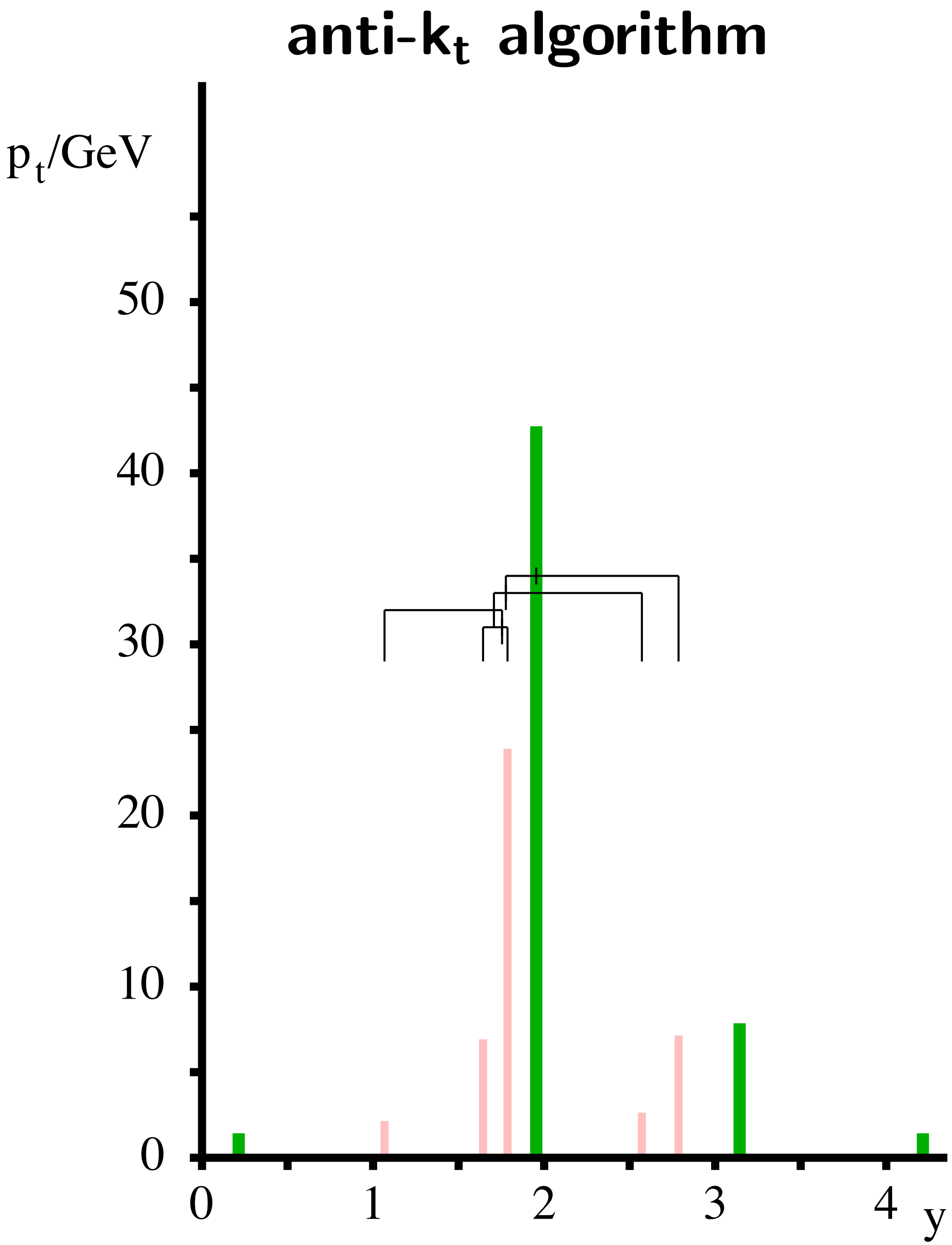


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Anti-k_t gradually makes its way through the secondary blob → no clear identification of substructure associated with 2nd parton.

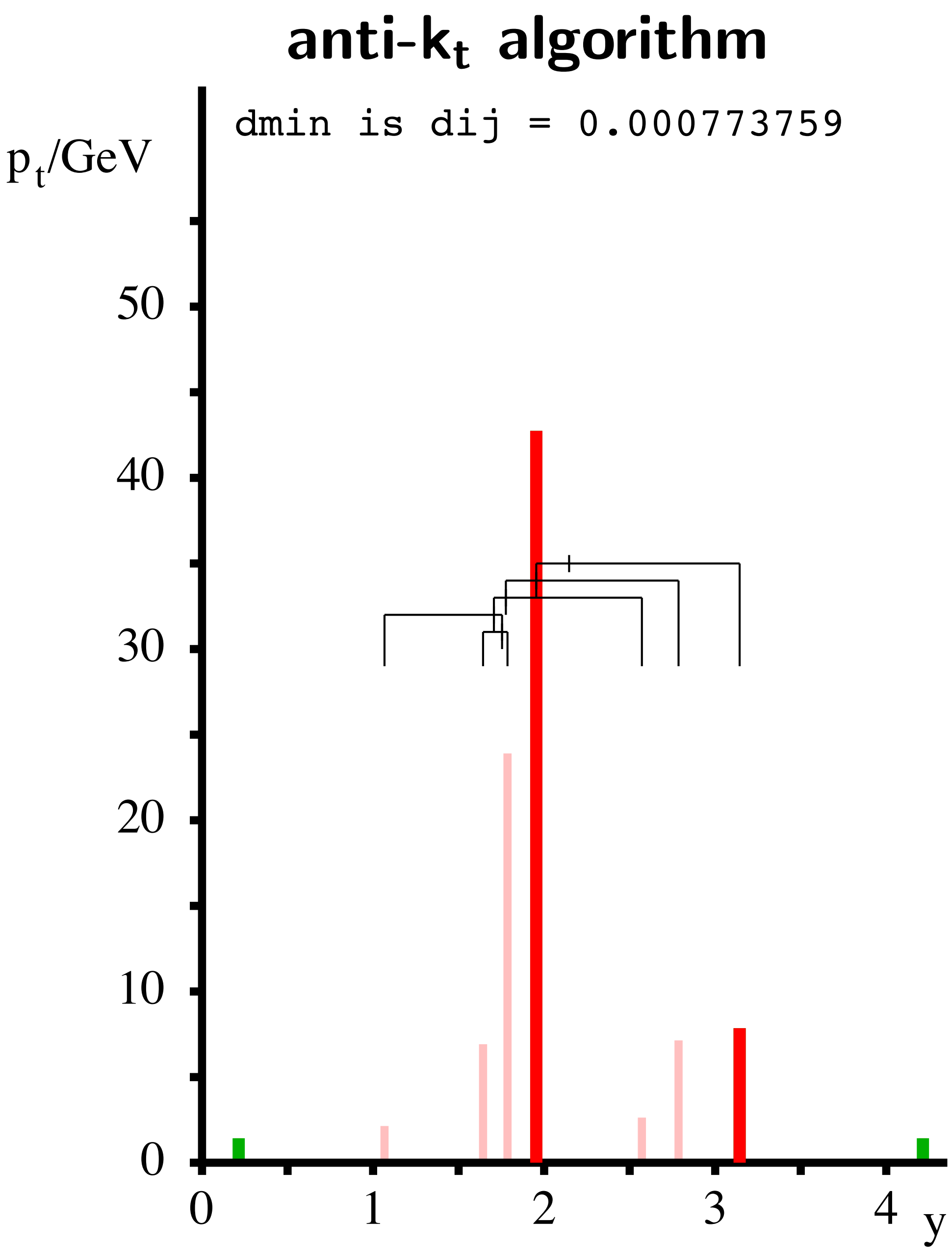
Identifying jet substructure: try out anti-k_t



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

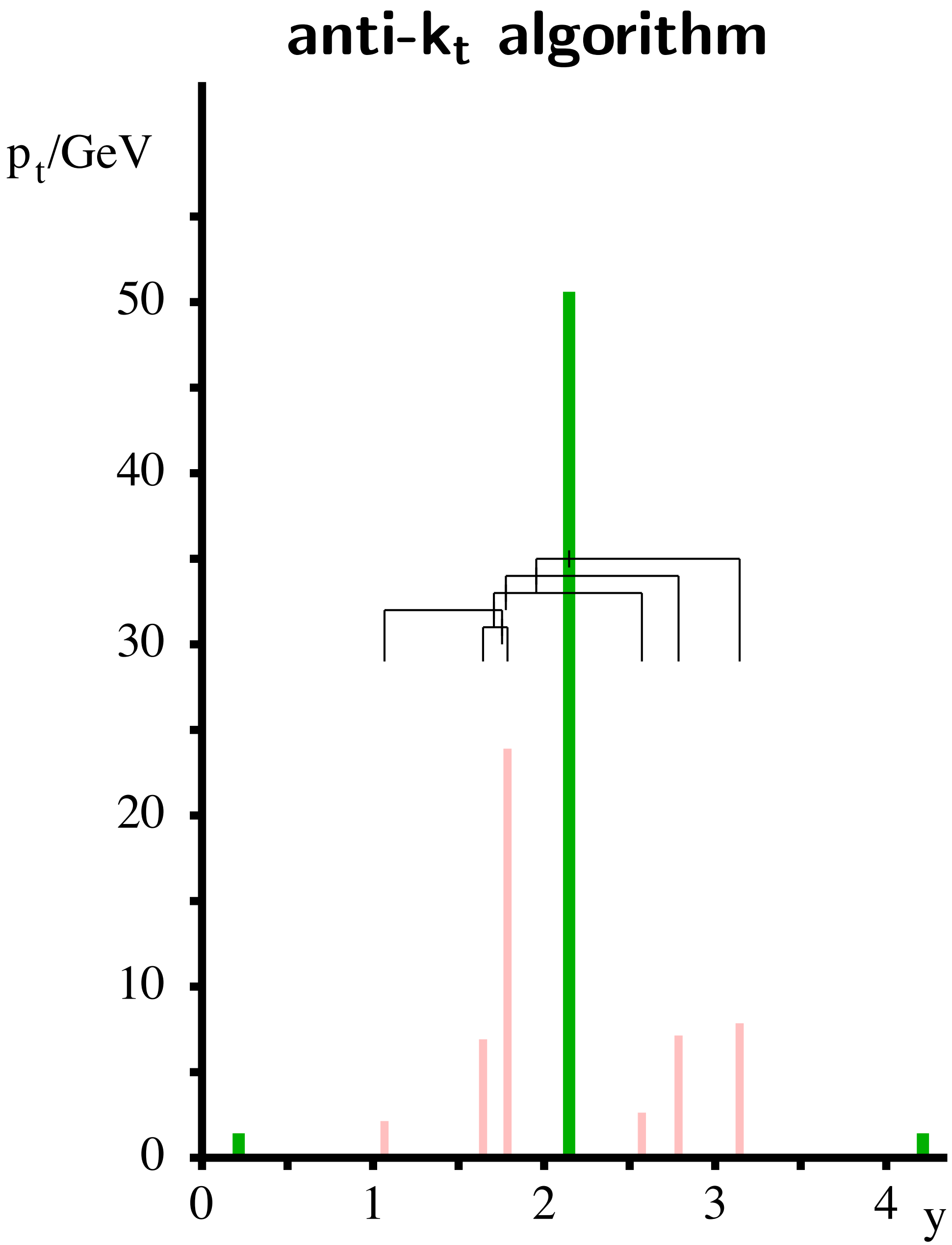
Anti-k_t gradually makes its way through the secondary blob $\rightarrow$ no clear identification of substructure associated with 2nd parton.



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

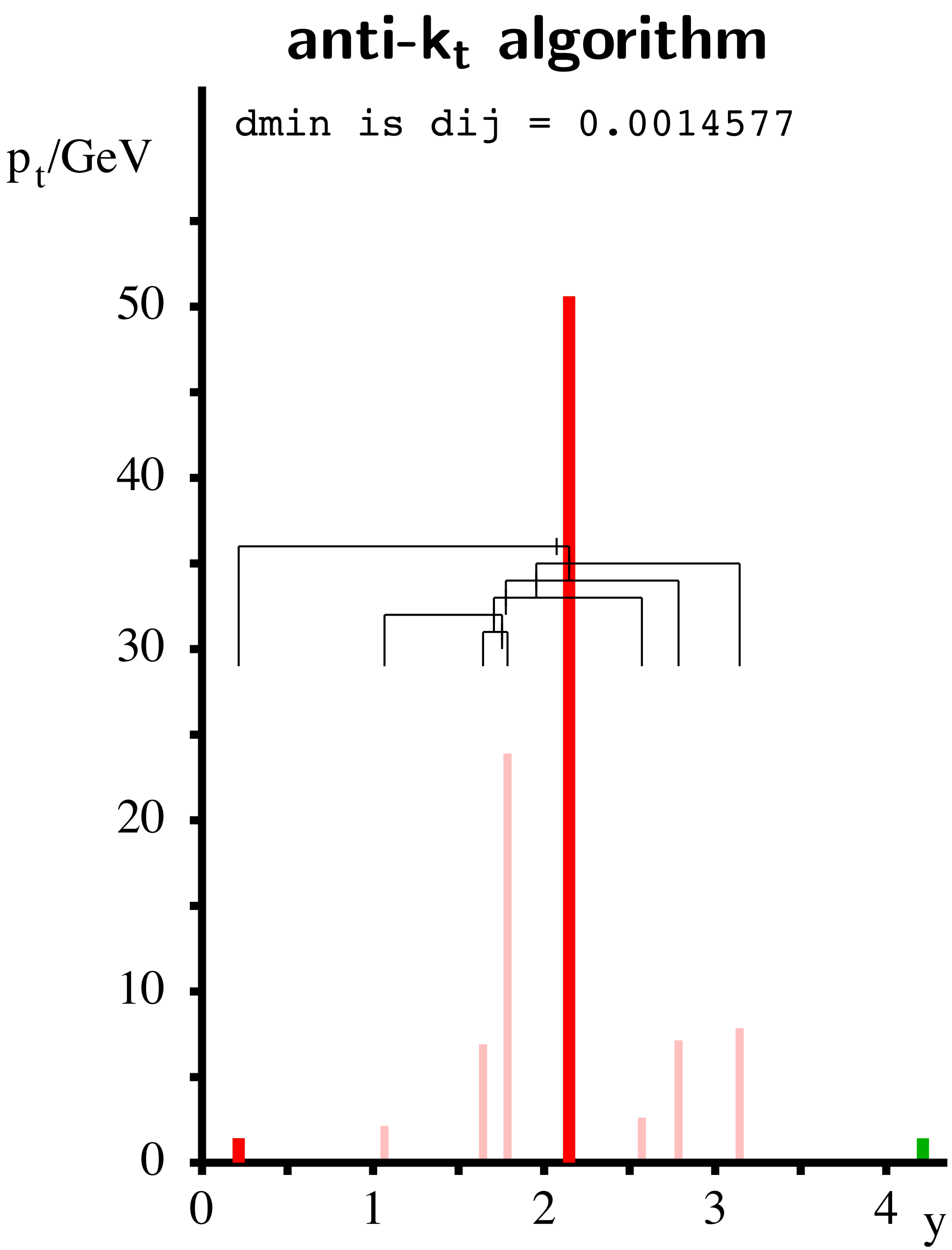
Anti-k_t gradually makes its way through the secondary blob → no clear identification of substructure associated with 2nd parton.



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Anti-k_t gradually makes its way through the secondary blob → no clear identification of substructure associated with 2nd parton.

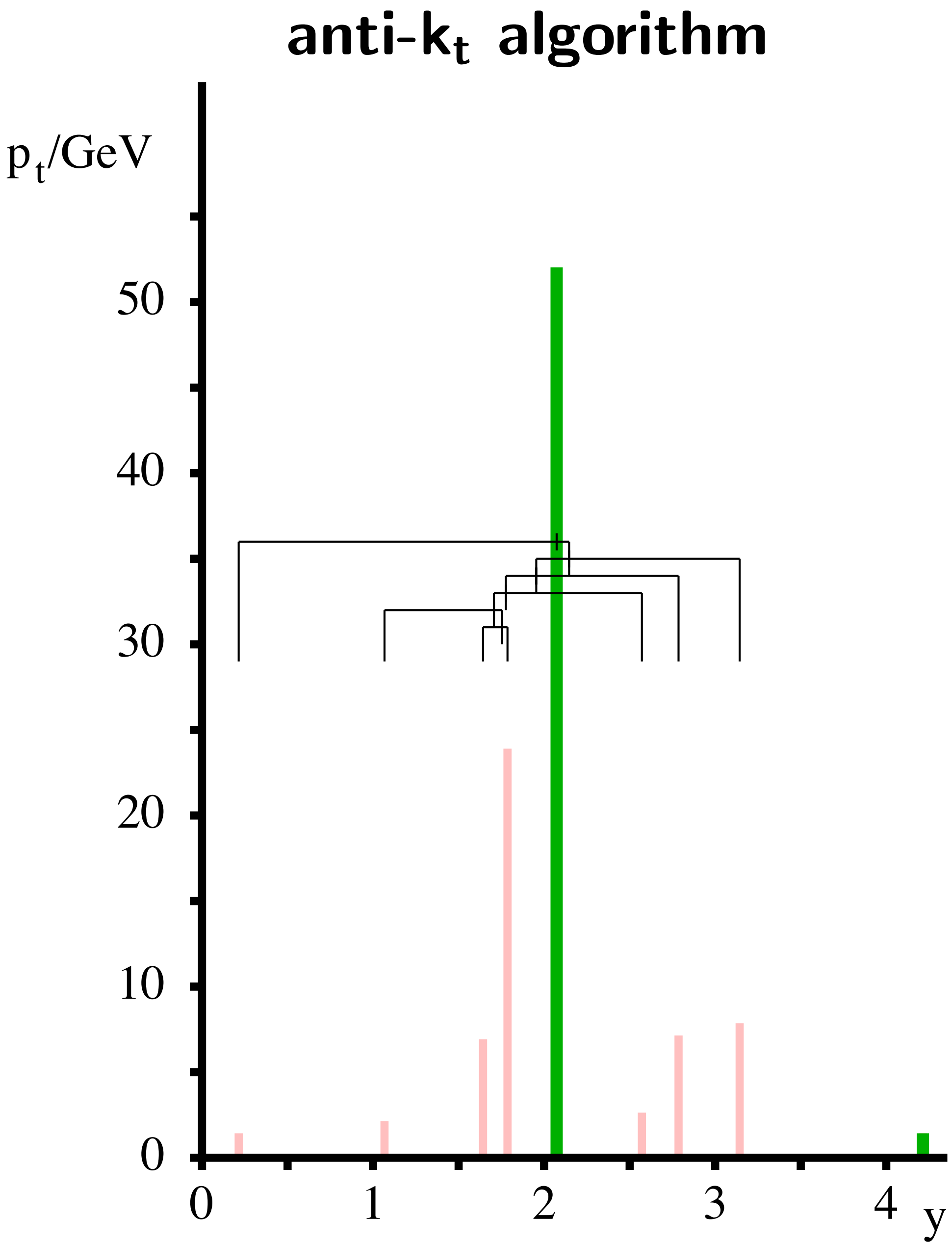


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Anti-k_t gradually makes its way through the secondary blob → no clear identification of substructure associated with 2nd parton.

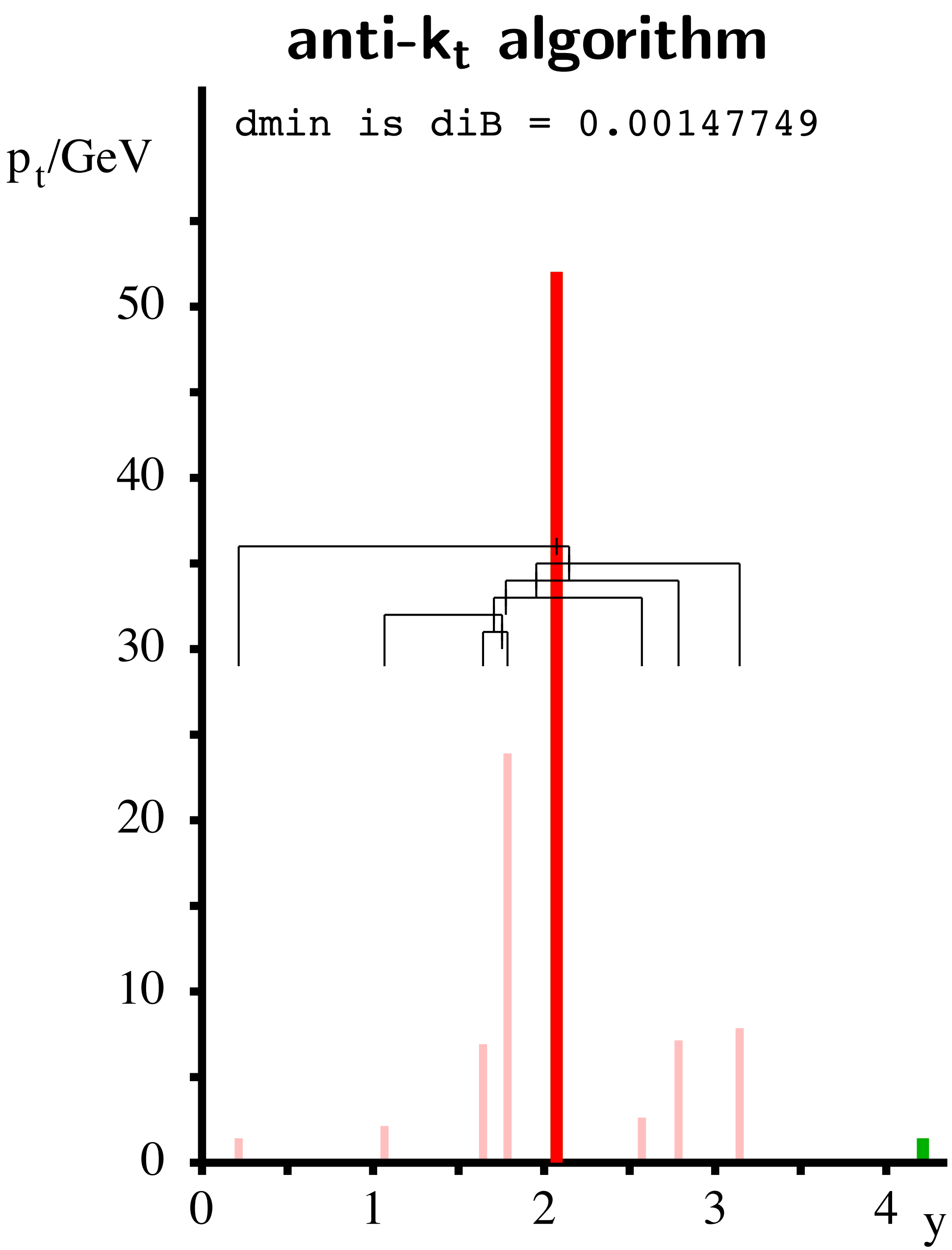
Identifying jet substructure: try out anti-k_t



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Anti-k_t gradually makes its way through the secondary blob → no clear identification of substructure associated with 2nd parton.

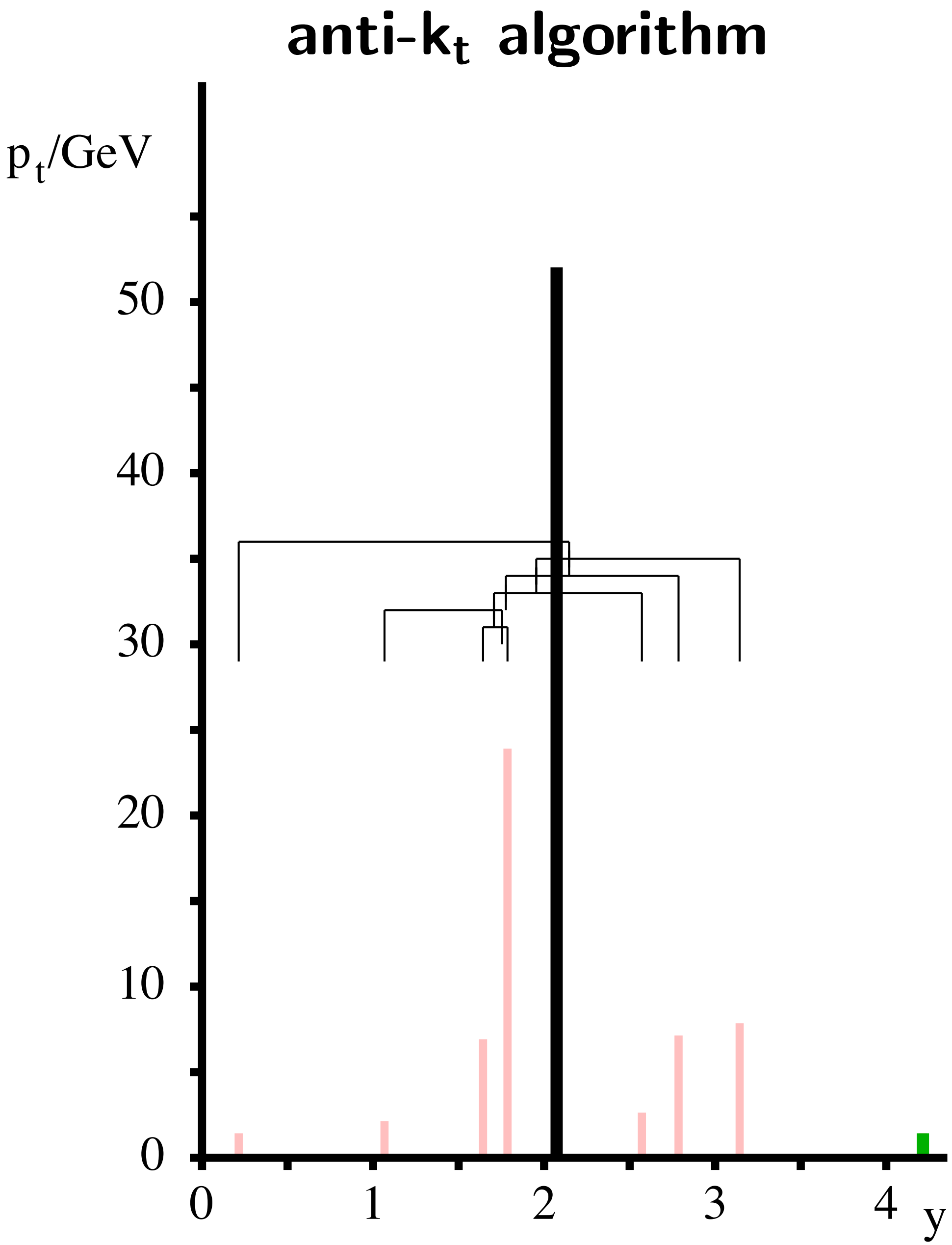


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Anti-k_t gradually makes its way through the secondary blob → no clear identification of substructure associated with 2nd parton.

Identifying jet substructure: try out anti-k_t

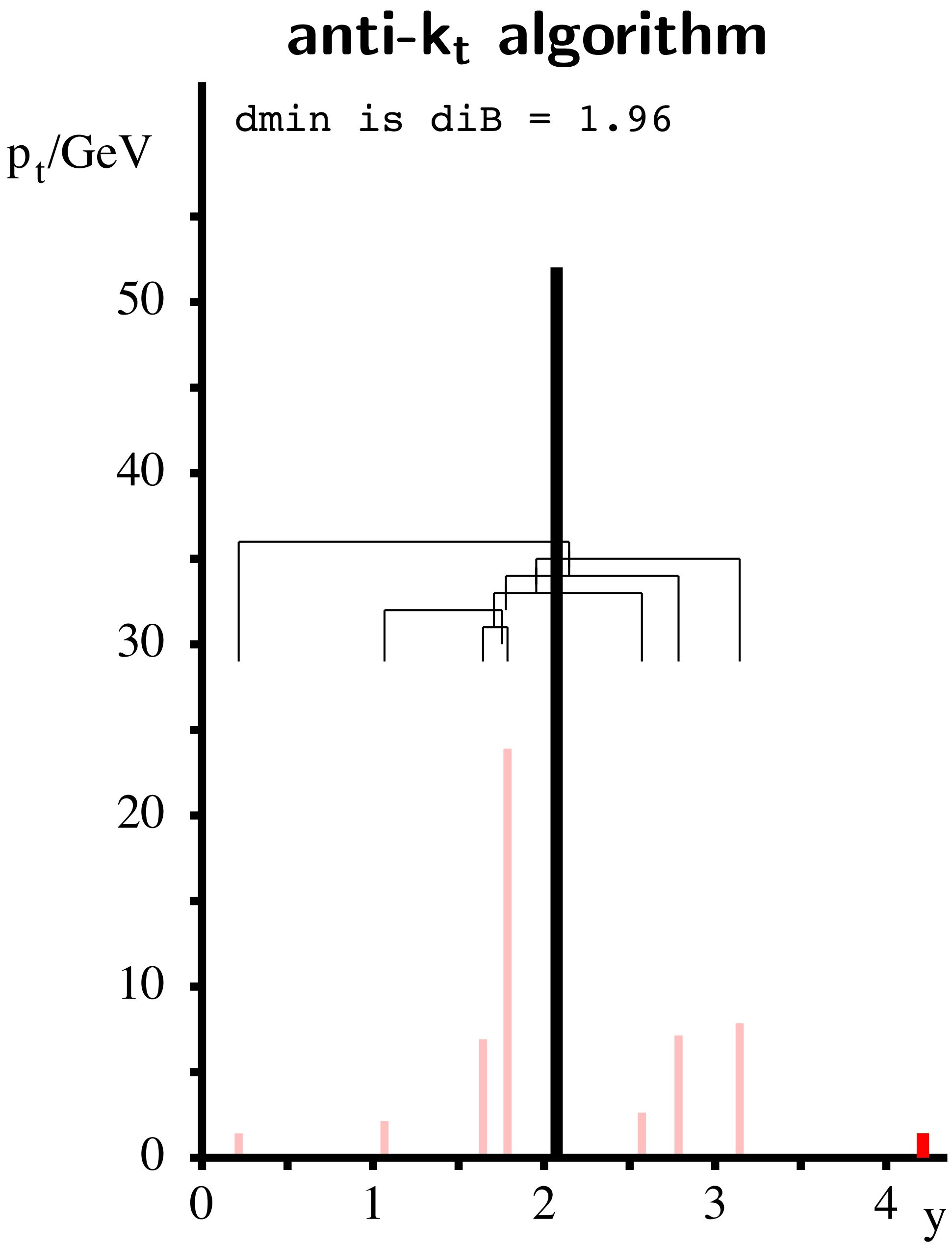


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Anti-k_t gradually makes its way through the secondary blob → no clear identification of substructure associated with 2nd parton.

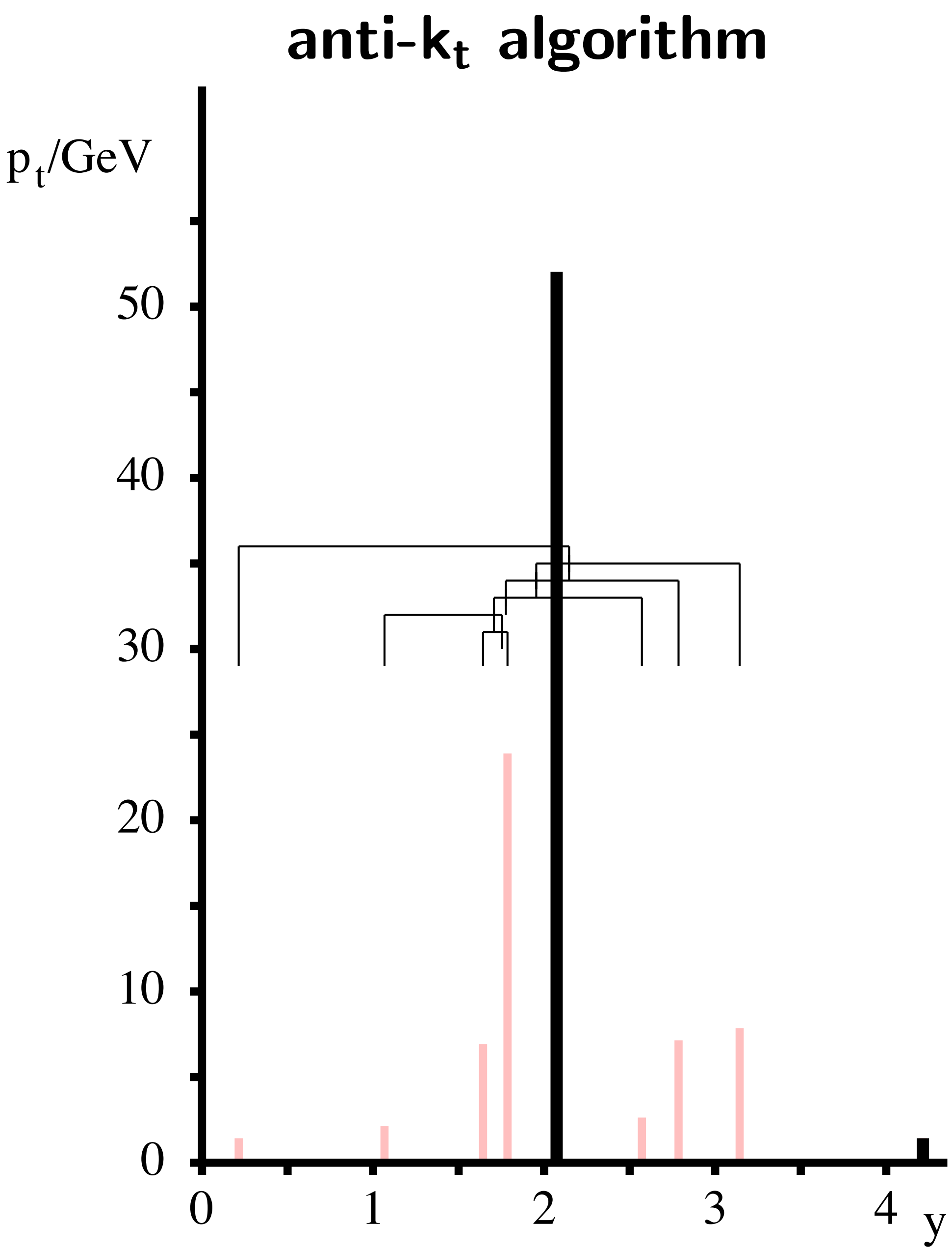
Identifying jet substructure: try out anti-k_t



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Anti-k_t gradually makes its way through the secondary blob $\rightarrow$ no clear identification of substructure associated with 2nd parton.



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

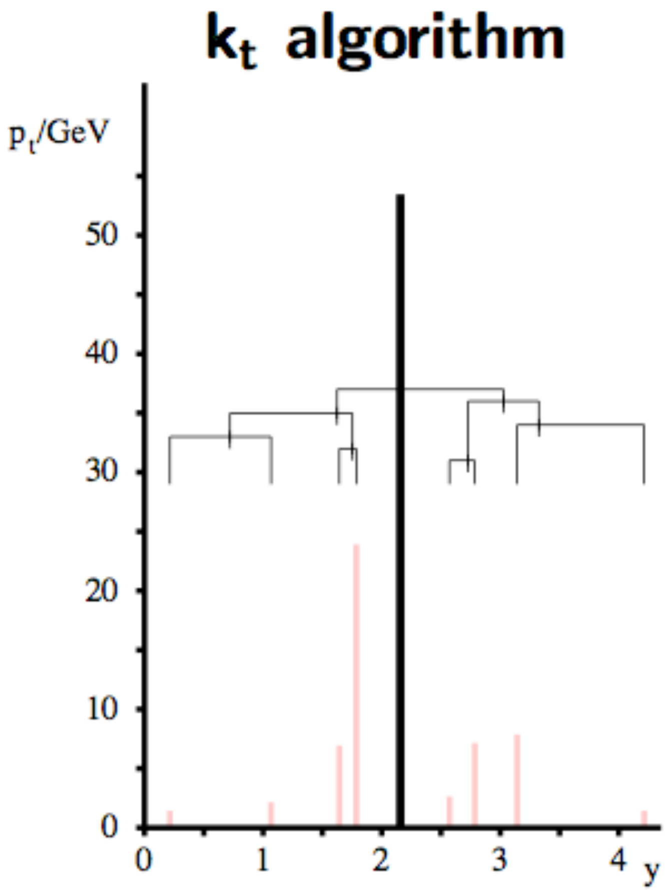
This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Anti-k_t gradually makes its way through the secondary blob $\rightarrow$ no clear identification of substructure associated with 2nd parton.

Second try

kt

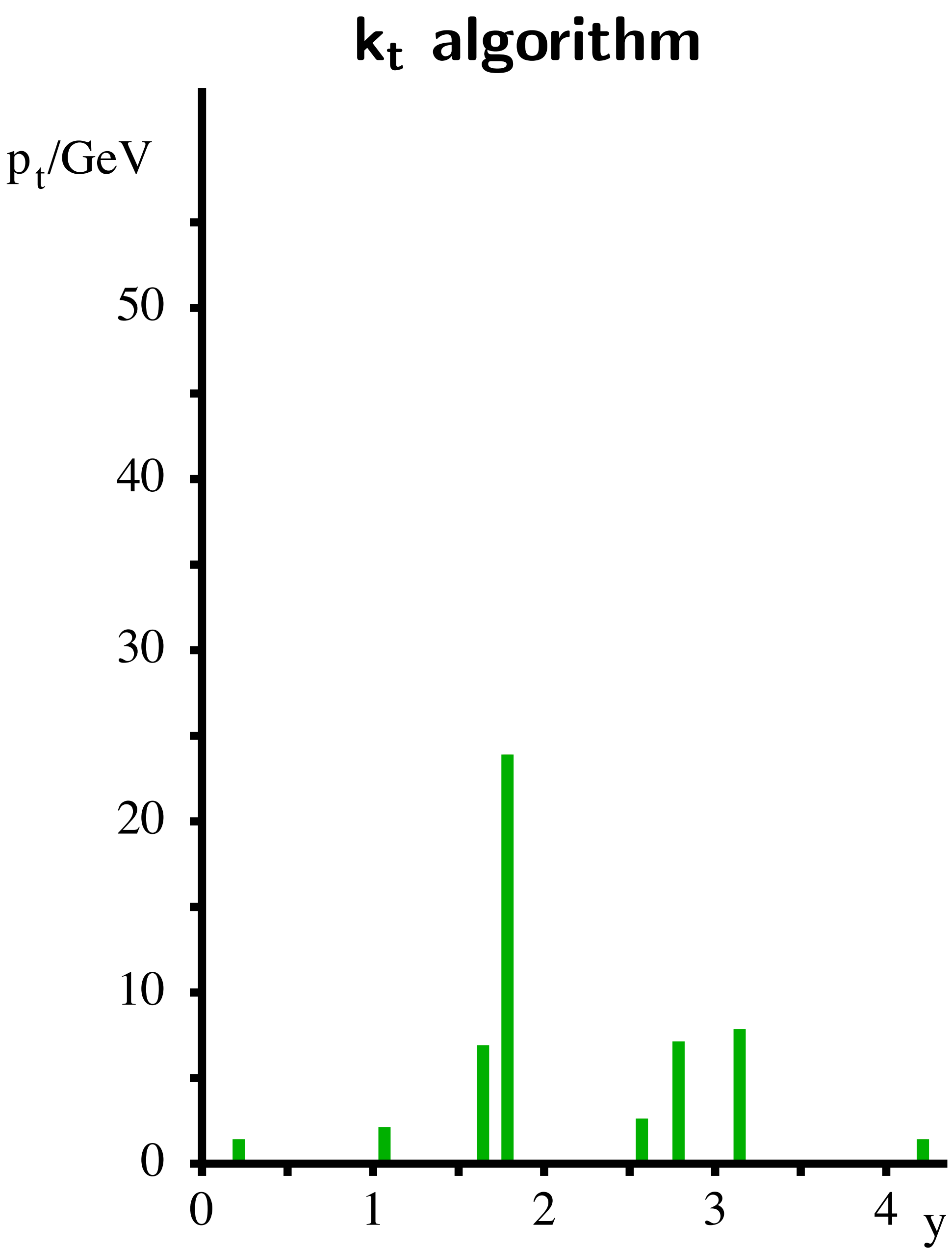
Hierarchical substructure



k_t distance measure

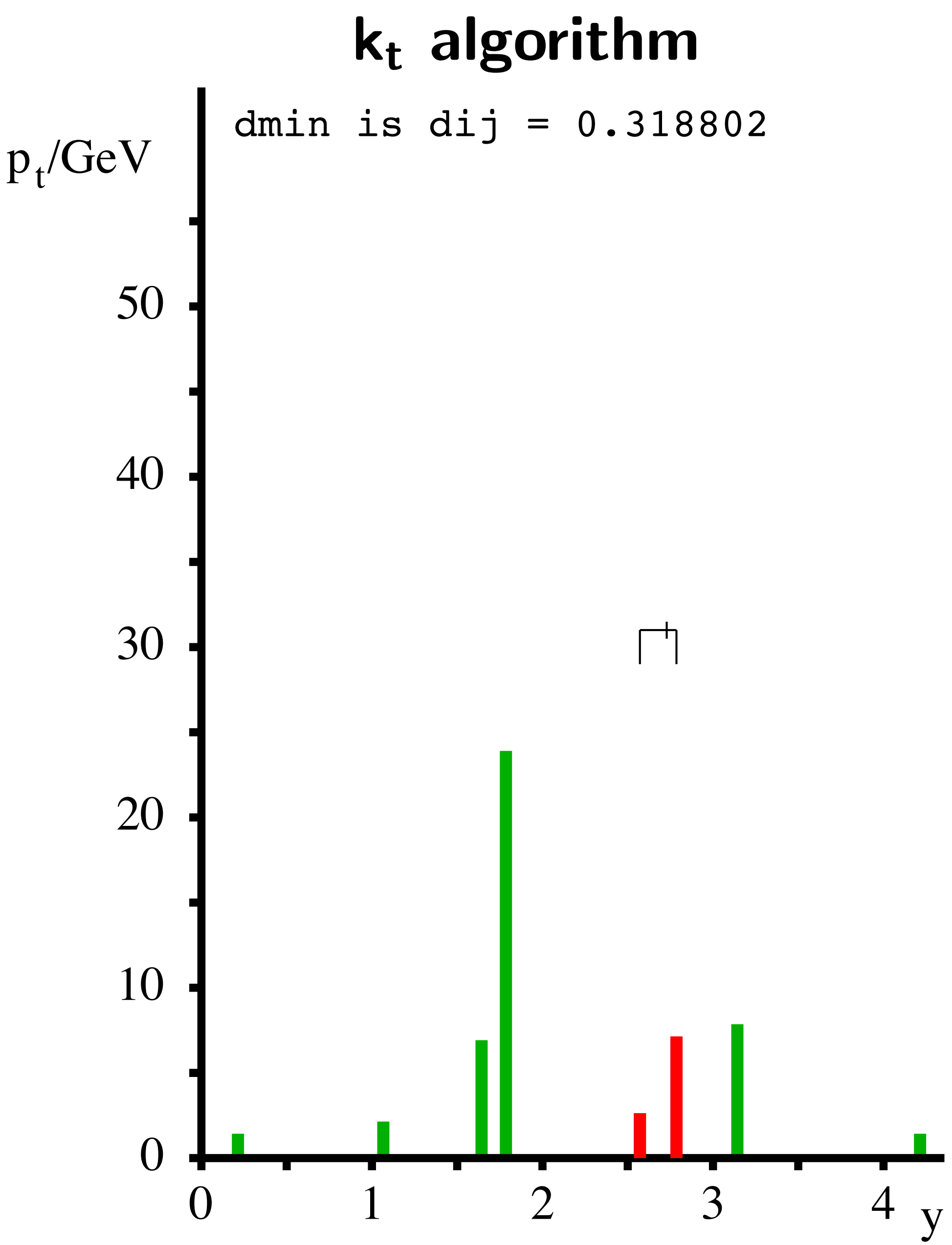
$$d_{ij} = \min(p_{ti}^2, p_{tj}^2) \frac{\Delta y^2 + \Delta \phi^2}{R^2}$$

Cluster by merging
the **softest/closest** particles



How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

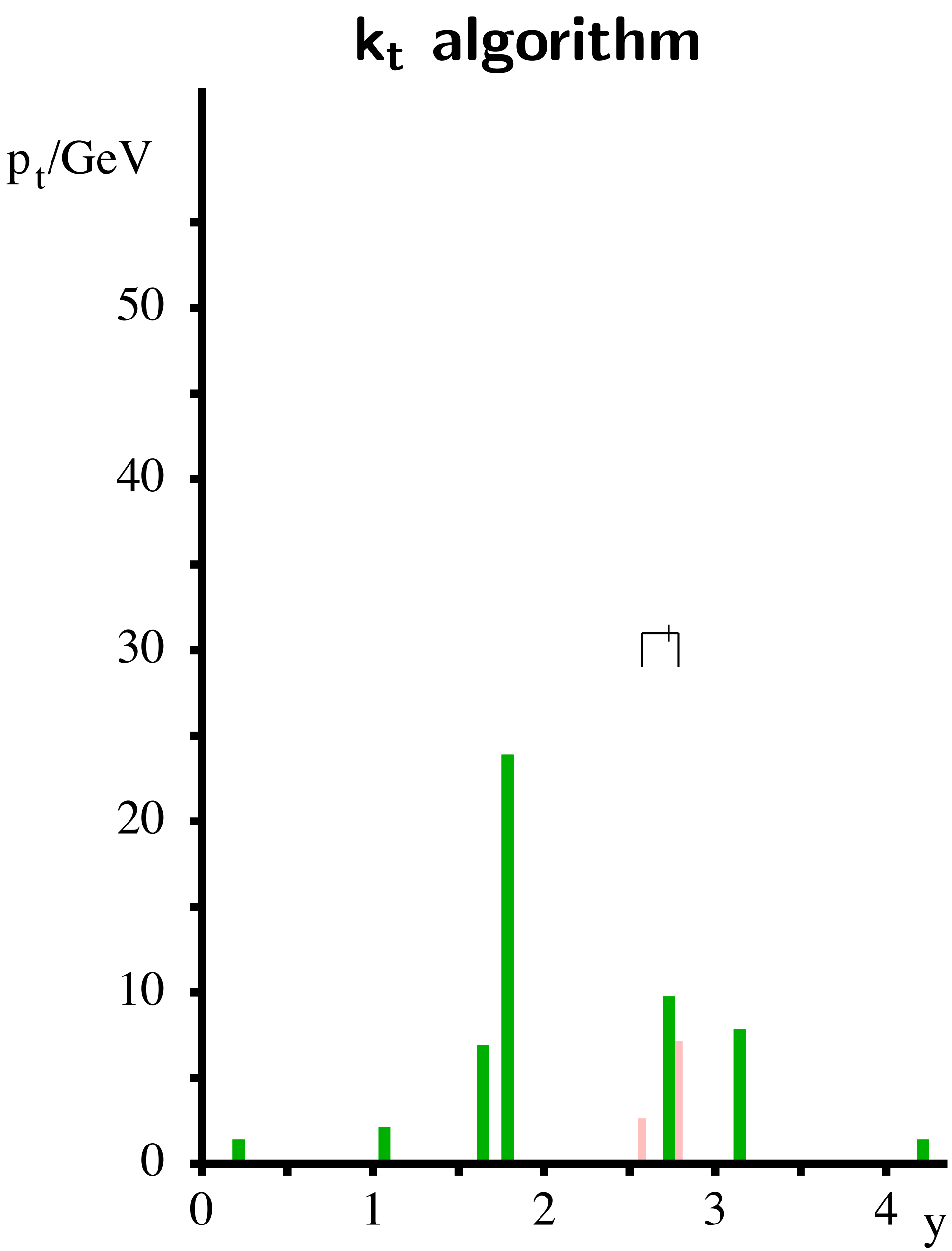
This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

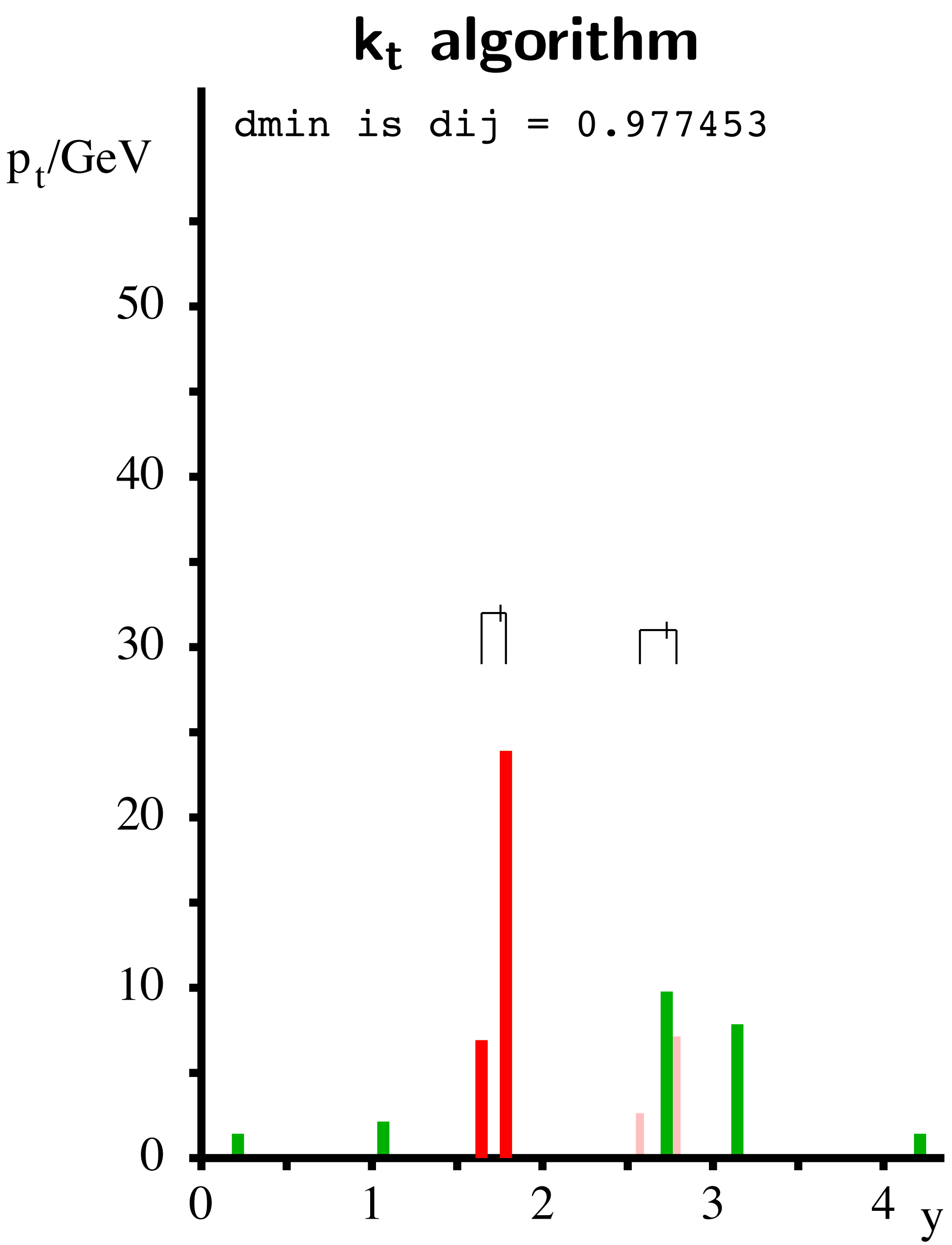
Identifying jet substructure: try out k_t



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

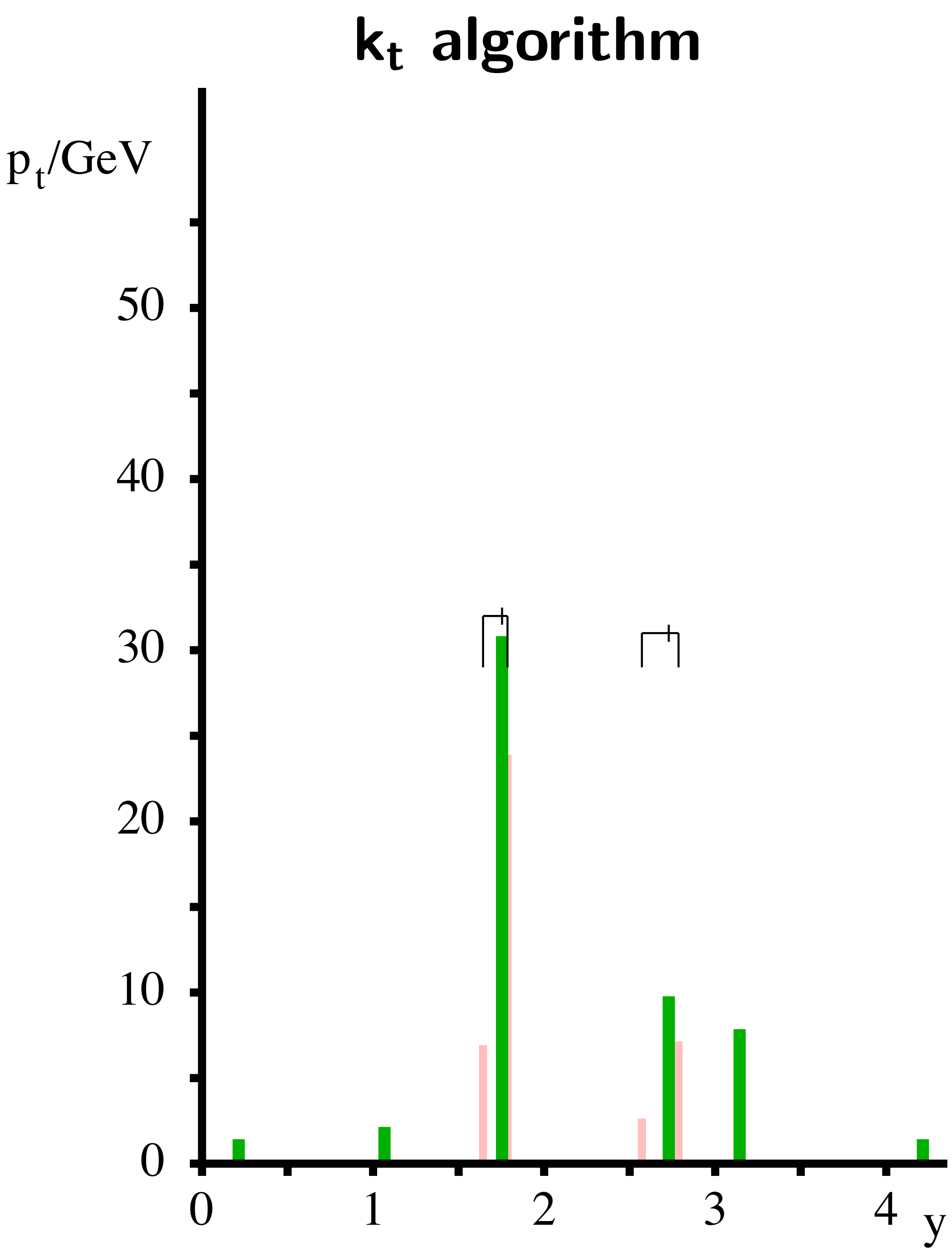
Identifying jet substructure: try out k_t



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

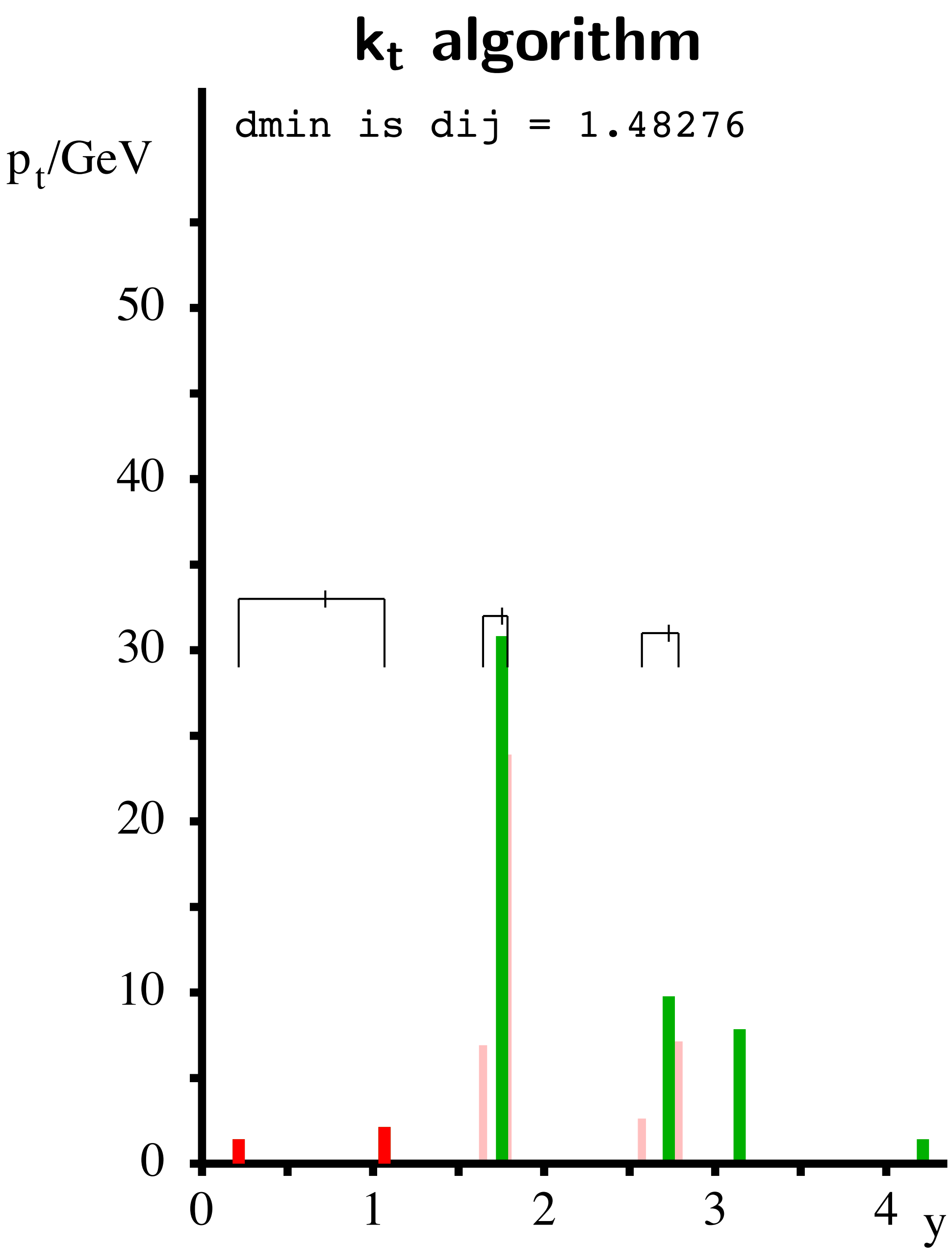
This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Identifying jet substructure: try out k_t



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

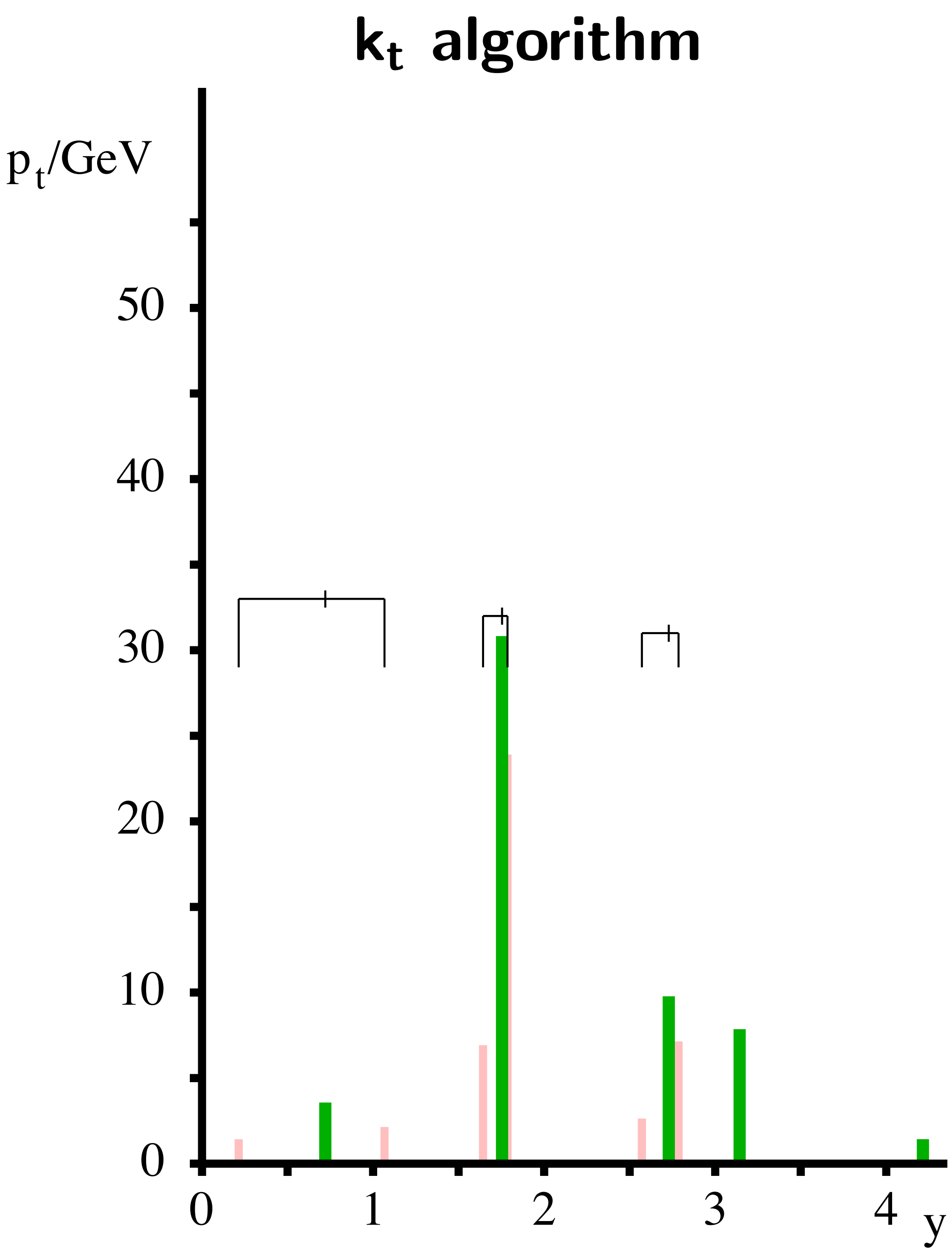
This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

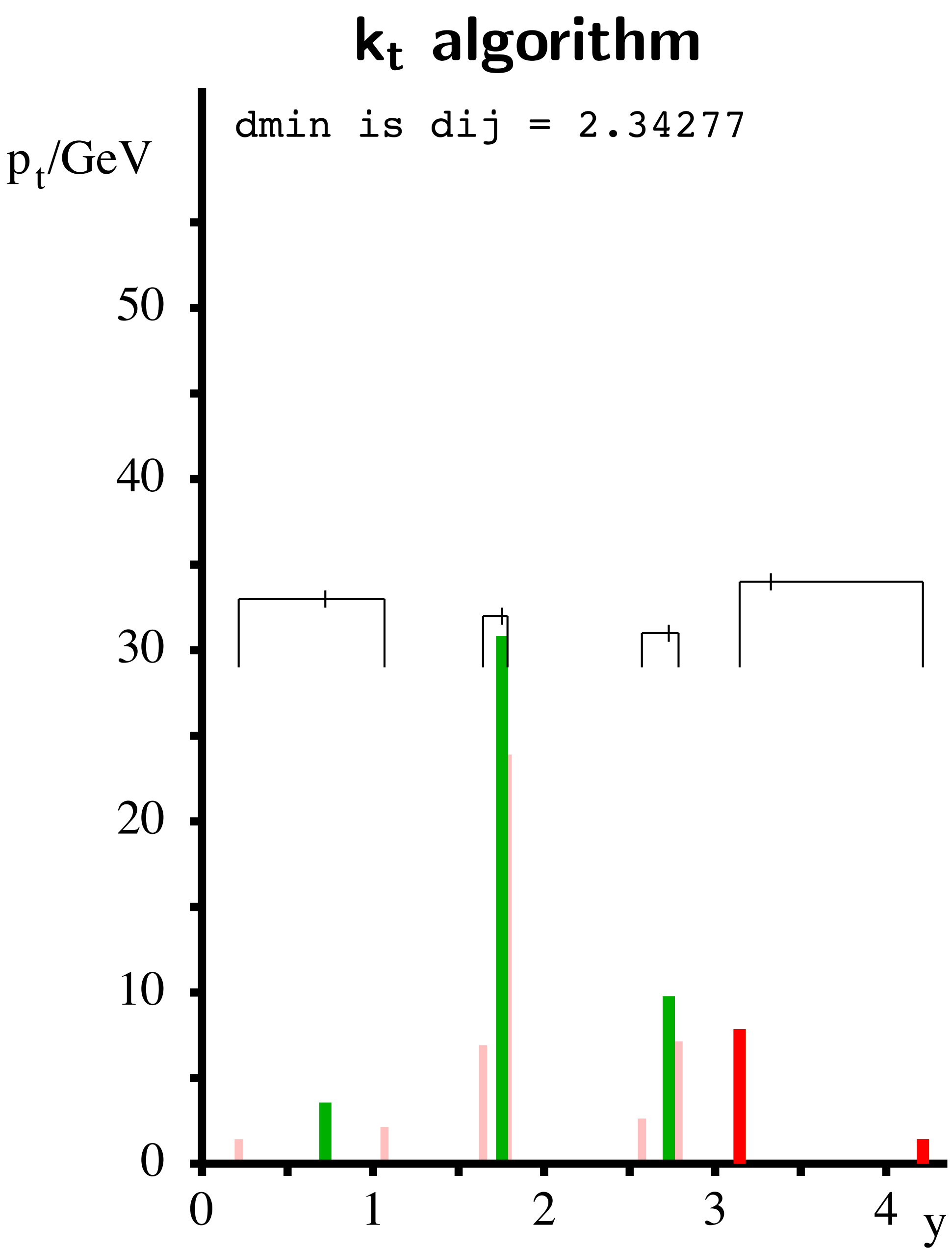
k_t clusters soft "junk" early on in the clustering



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

k_t clusters soft "junk" early on in the clustering

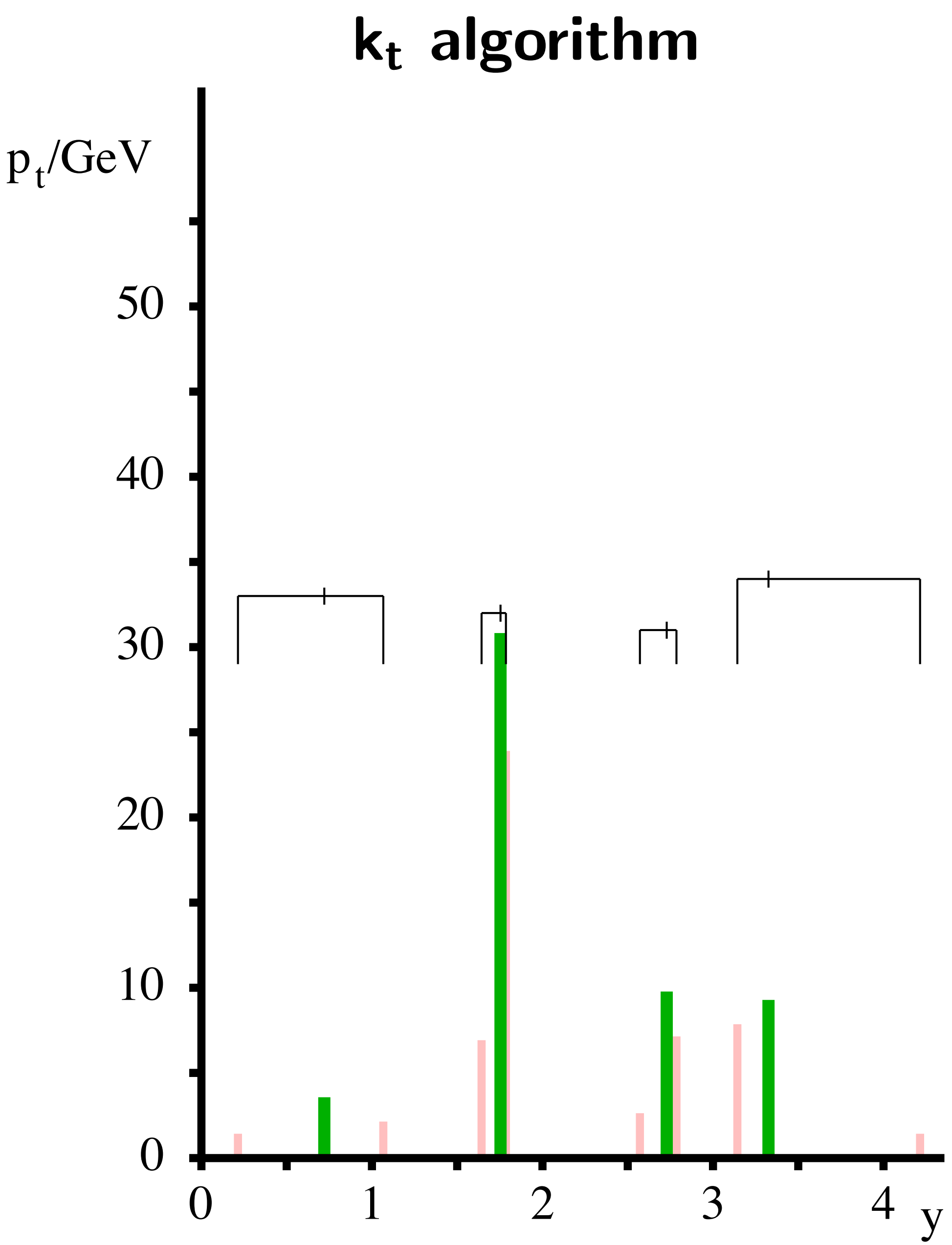


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

k_t clusters soft "junk" early on in the clustering

Identifying jet substructure: try out k_t

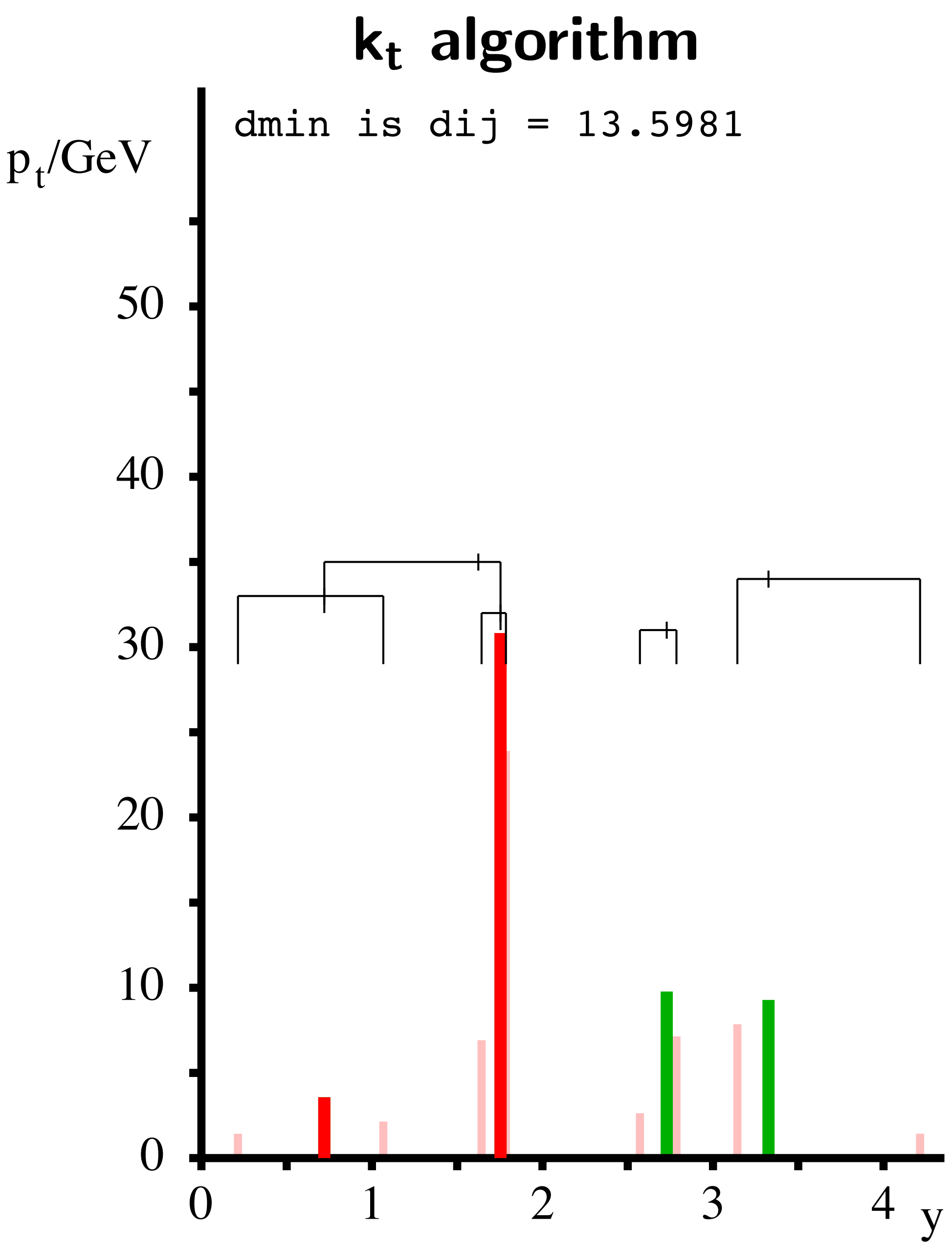


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

k_t clusters soft "junk" early on in the clustering

Identifying jet substructure: try out k_t

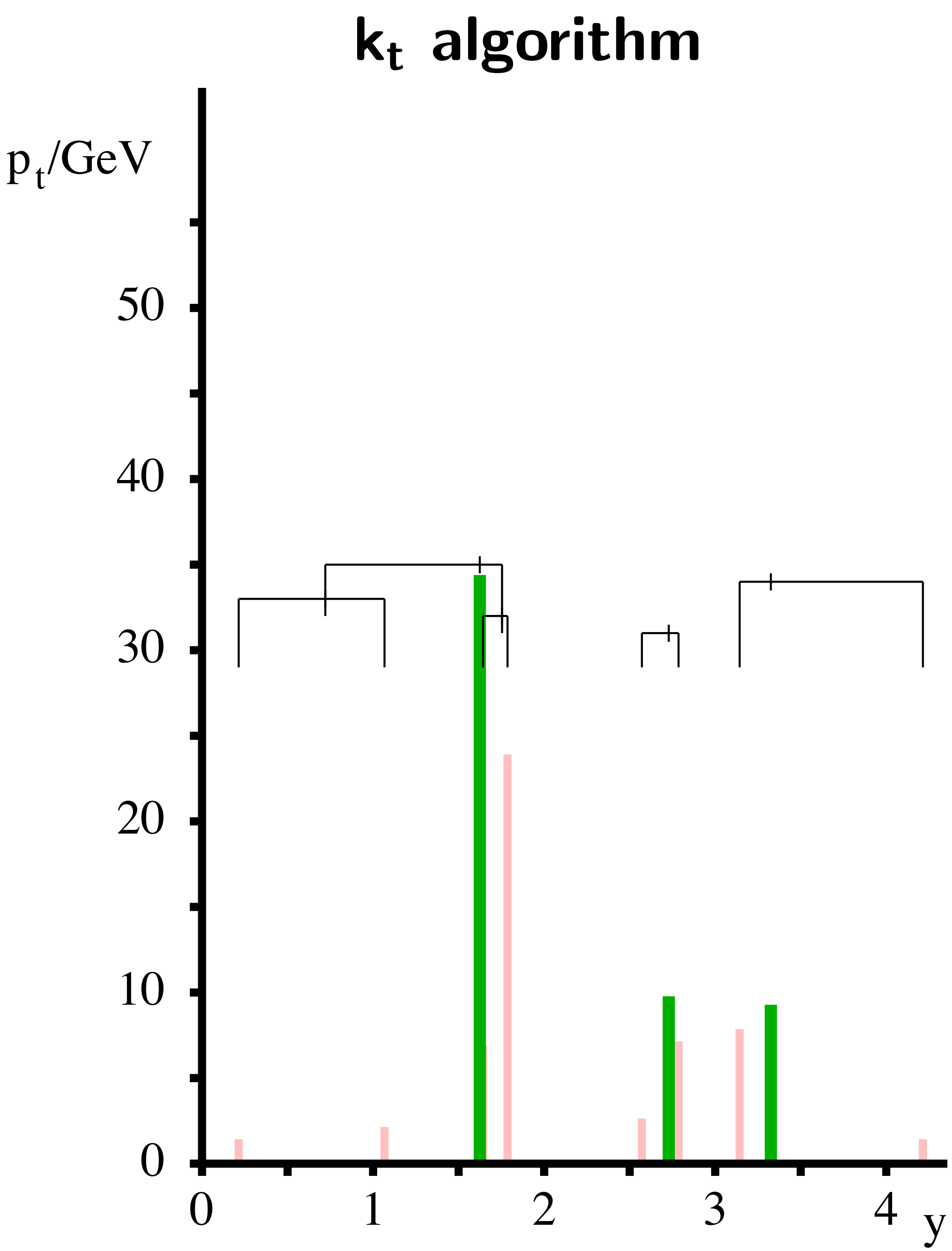


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

k_t clusters soft "junk" early on in the clustering

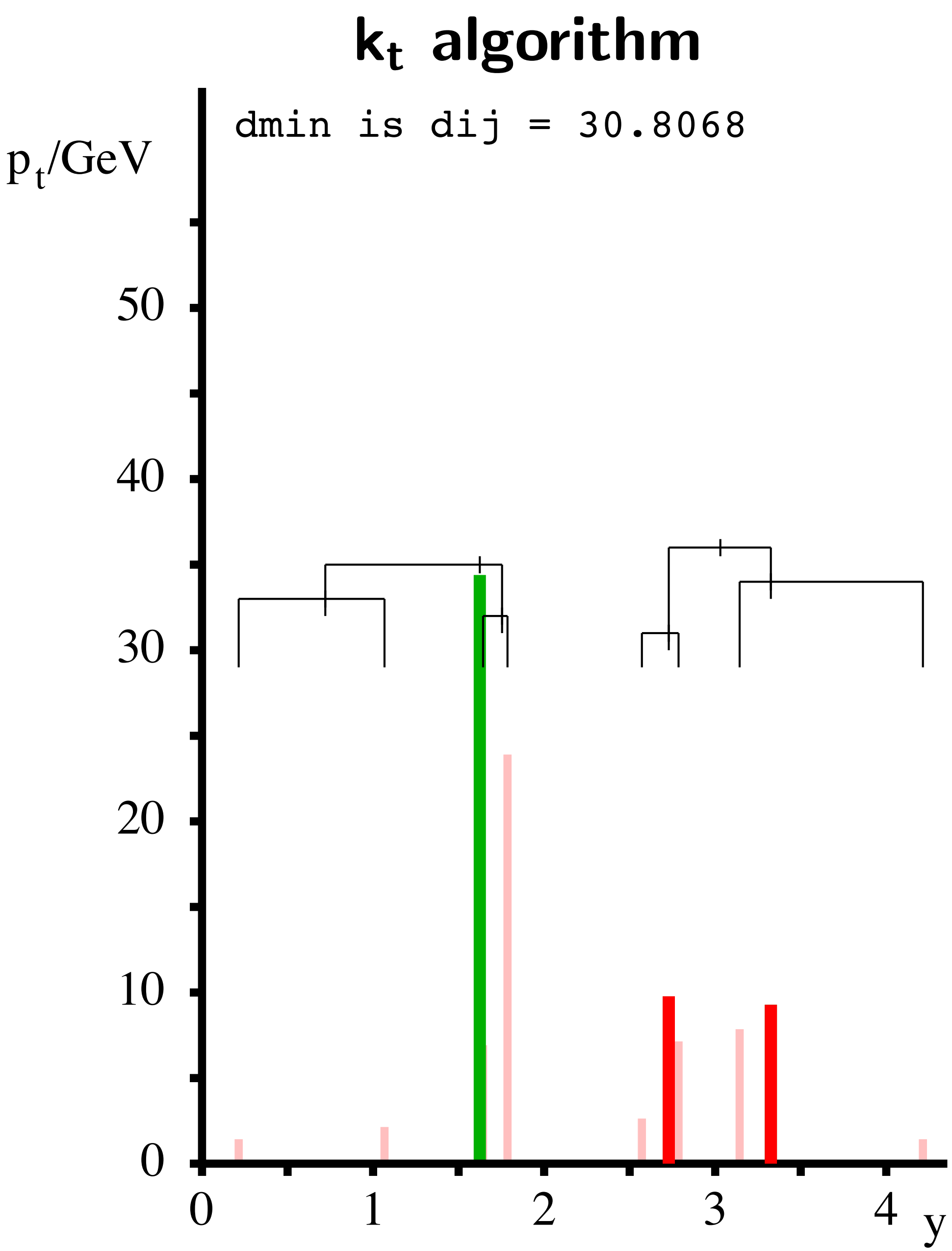
Identifying jet substructure: try out k_t



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

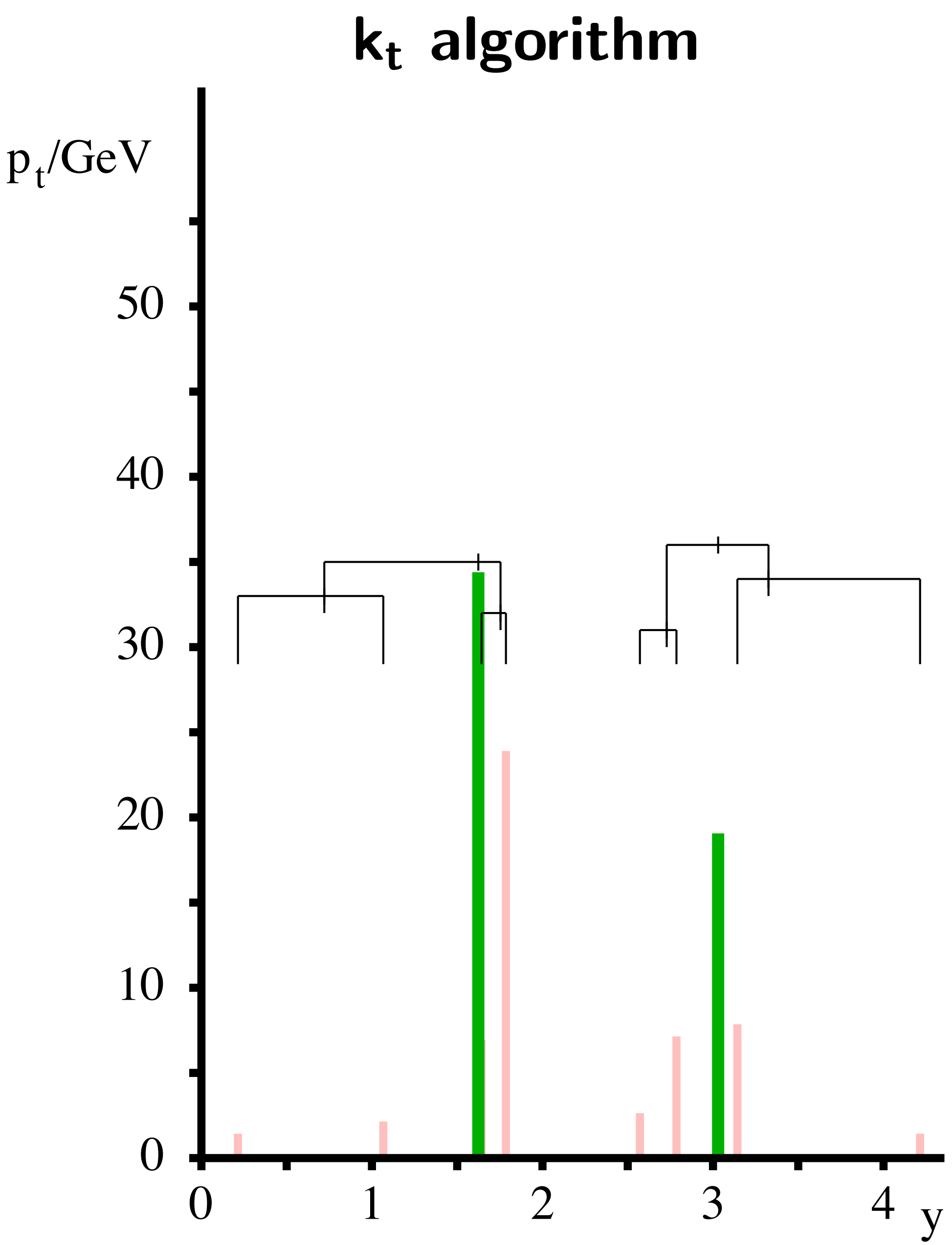
k_t clusters soft "junk" early on in the clustering



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

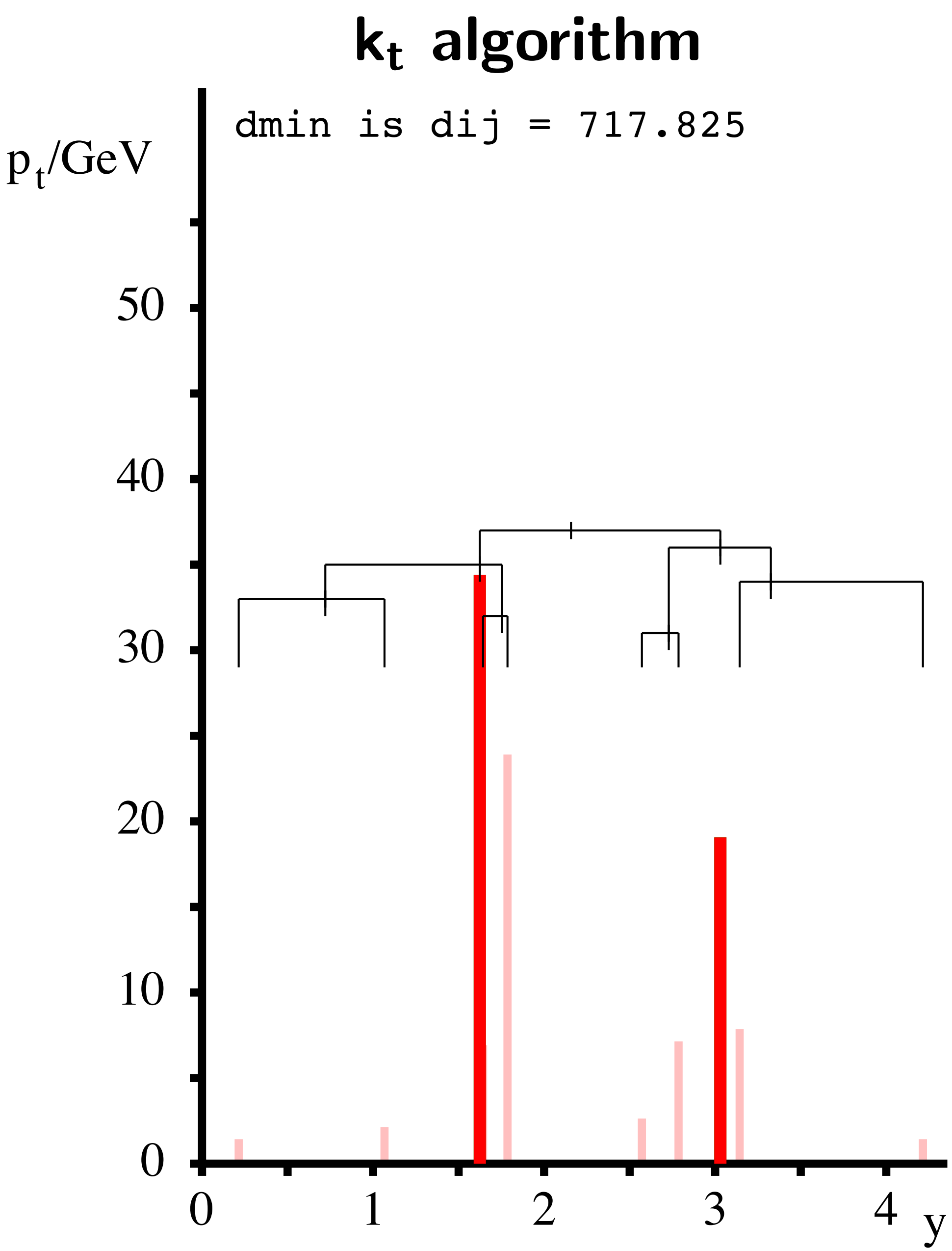
k_t clusters soft "junk" early on in the clustering



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

k_t clusters soft "junk" early on in the clustering

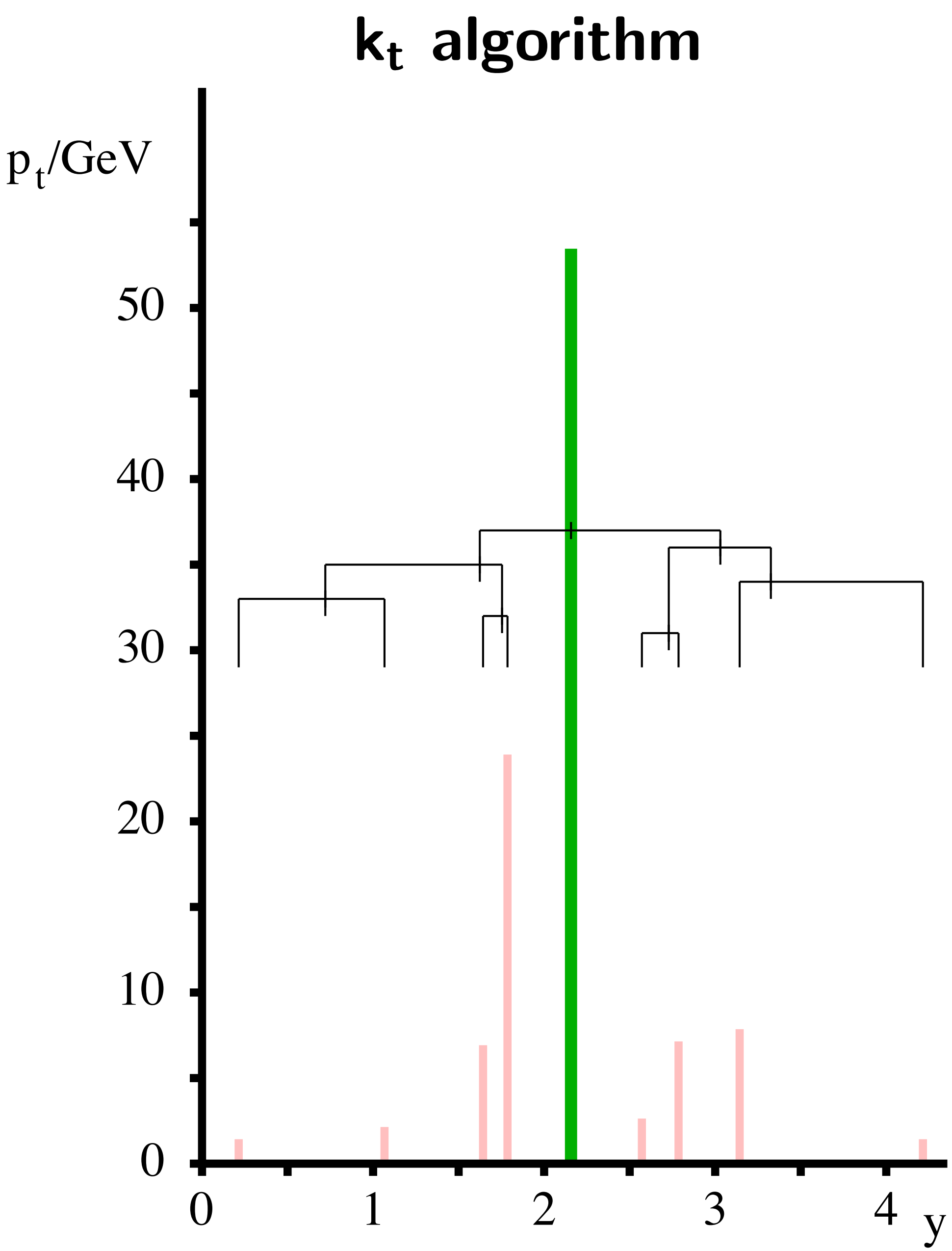


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

k_t clusters soft "junk" early on in the clustering

Its last step is to merge two hard pieces. Easily undone to identify underlying kinematics



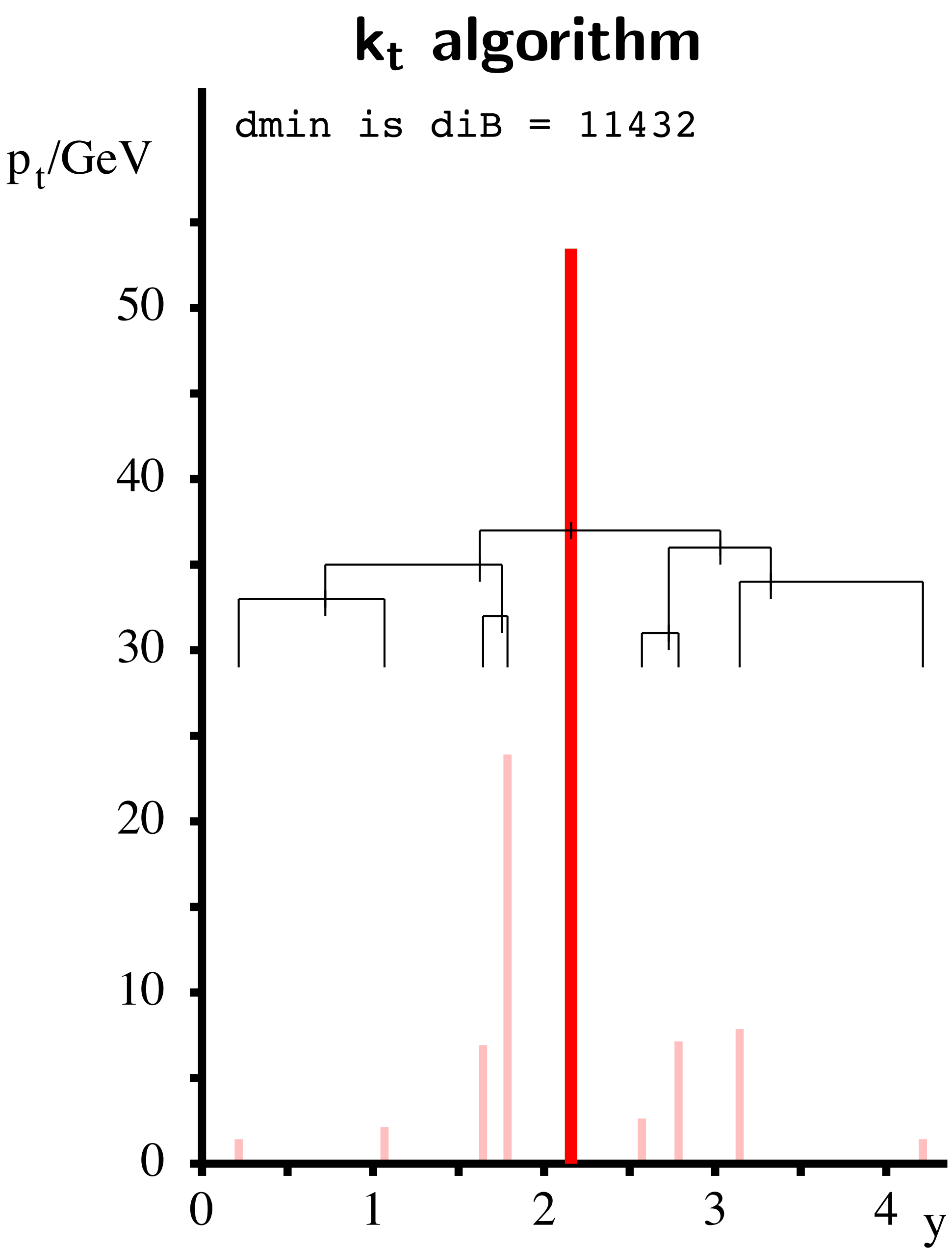
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

k_t clusters soft "junk" early on in the clustering

Its last step is to merge two hard pieces. Easily undone to identify underlying kinematics

Identifying jet substructure: try out k_t



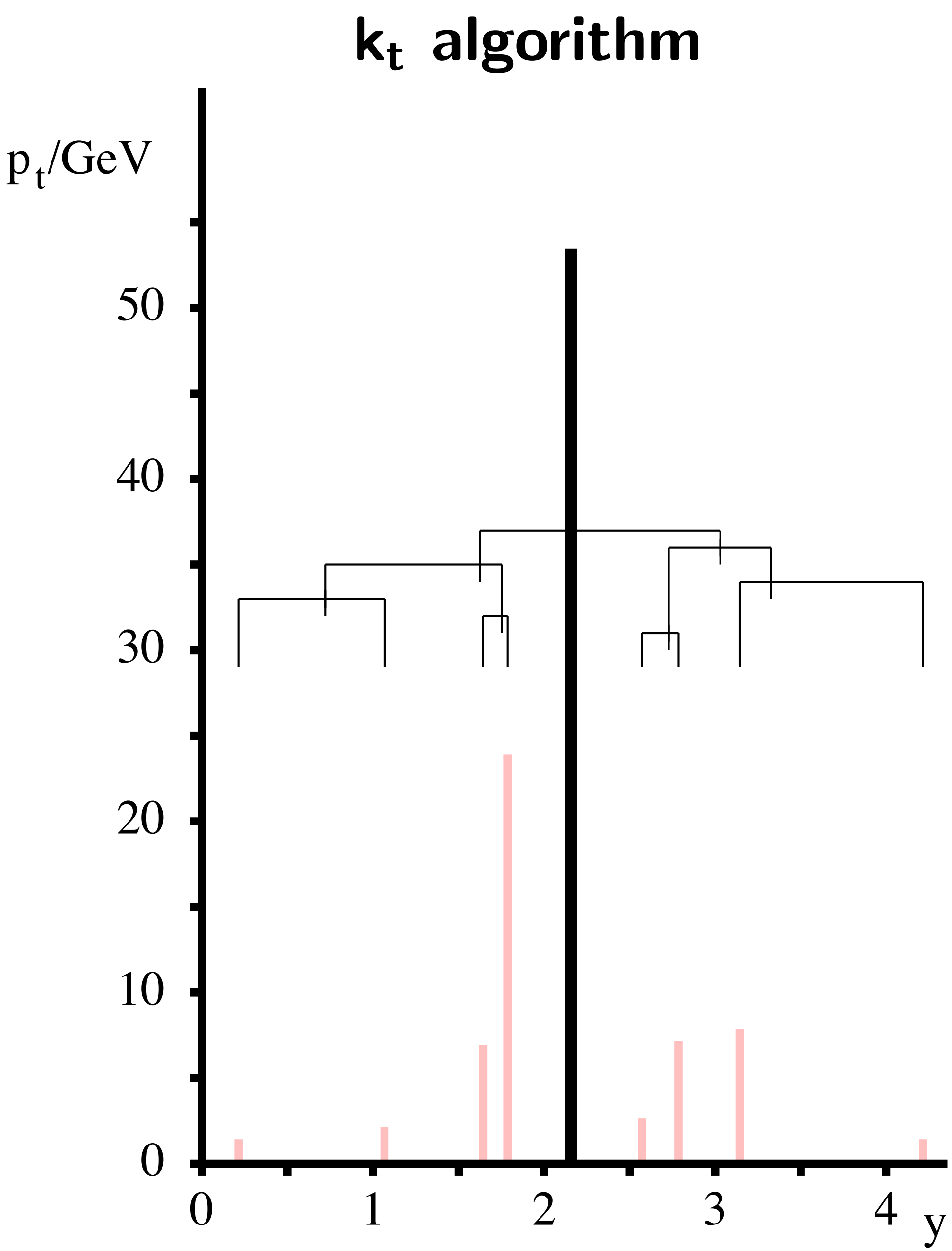
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

k_t clusters soft "junk" early on in the clustering

Its last step is to merge two hard pieces. Easily undone to identify underlying kinematics

Identifying jet substructure: try out k_t



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

k_t clusters soft "junk" early on in the clustering

Its last step is to merge two hard pieces. Easily undone to identify underlying kinematics

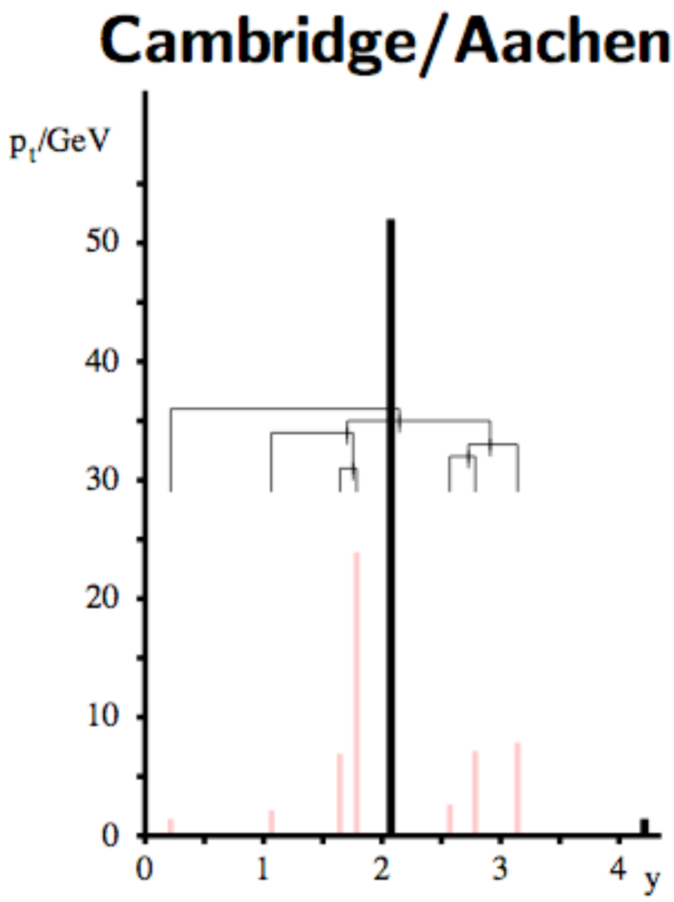
This meant it was the first algorithm to be used for jet substructure.

Seymour '93

Butterworth, Cox & Forshaw '02

Cambridge/Aachen

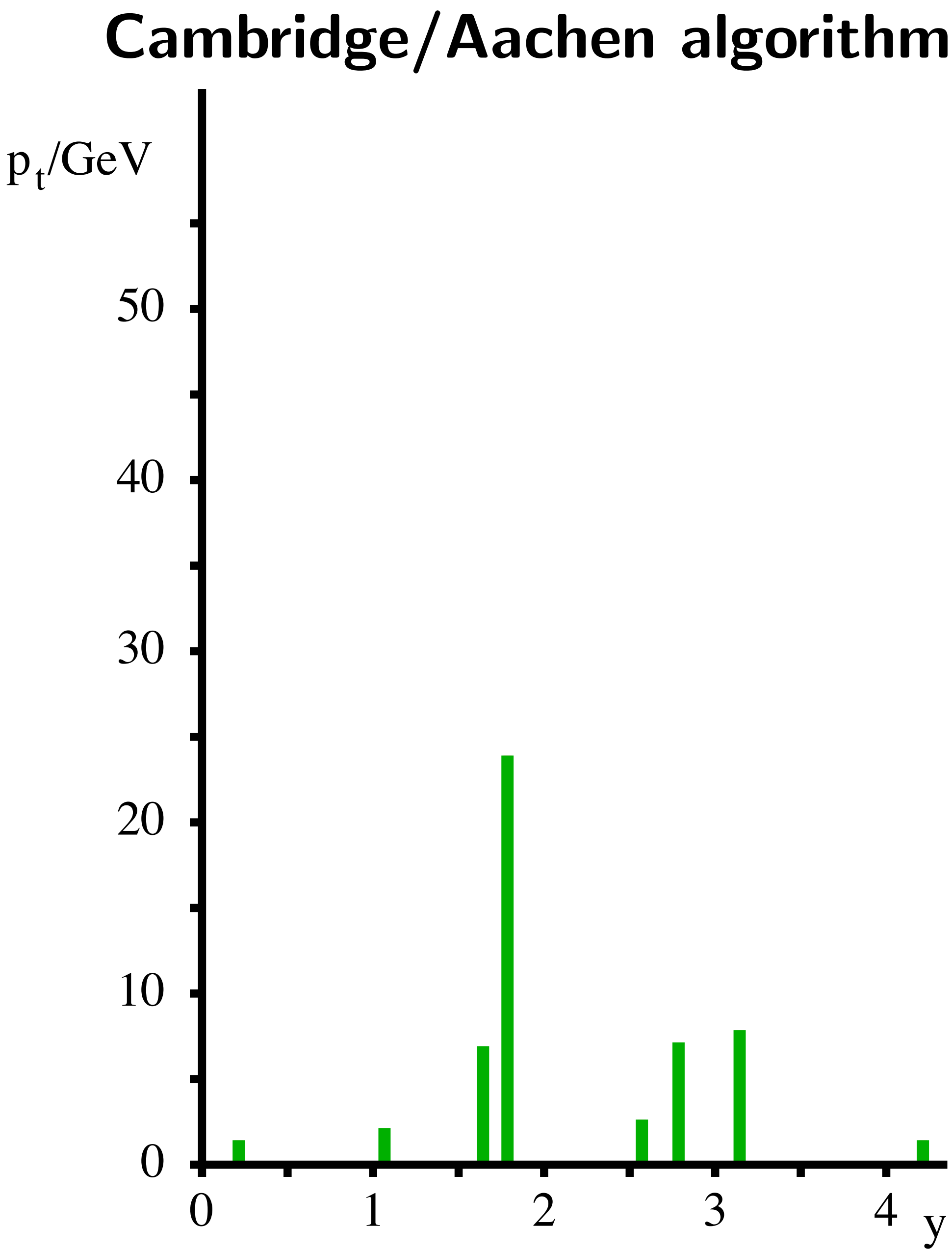
Hierarchical substructure



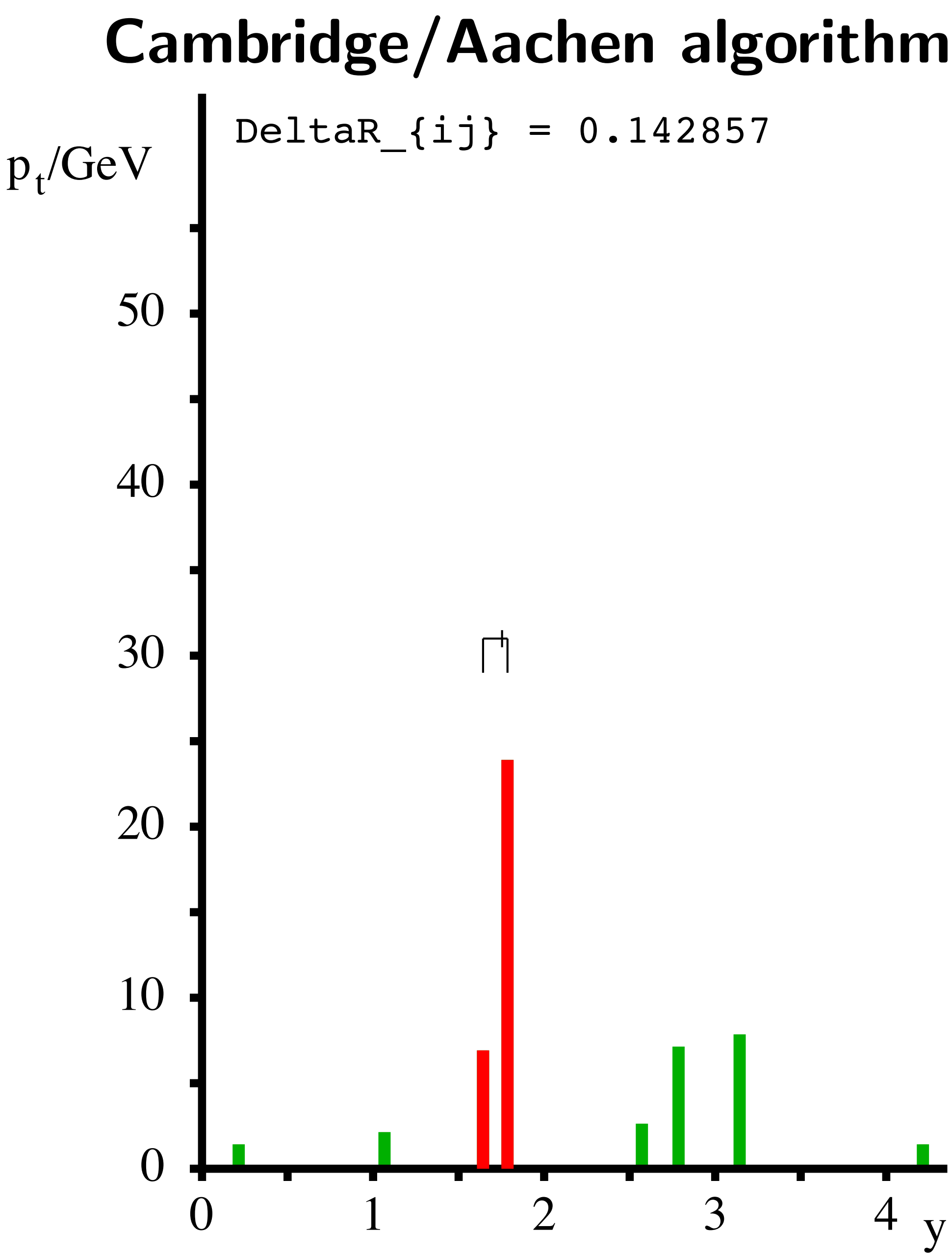
C/A distance measure

$$d_{ij} = \frac{\Delta y^2 + \Delta\phi^2}{R^2}$$

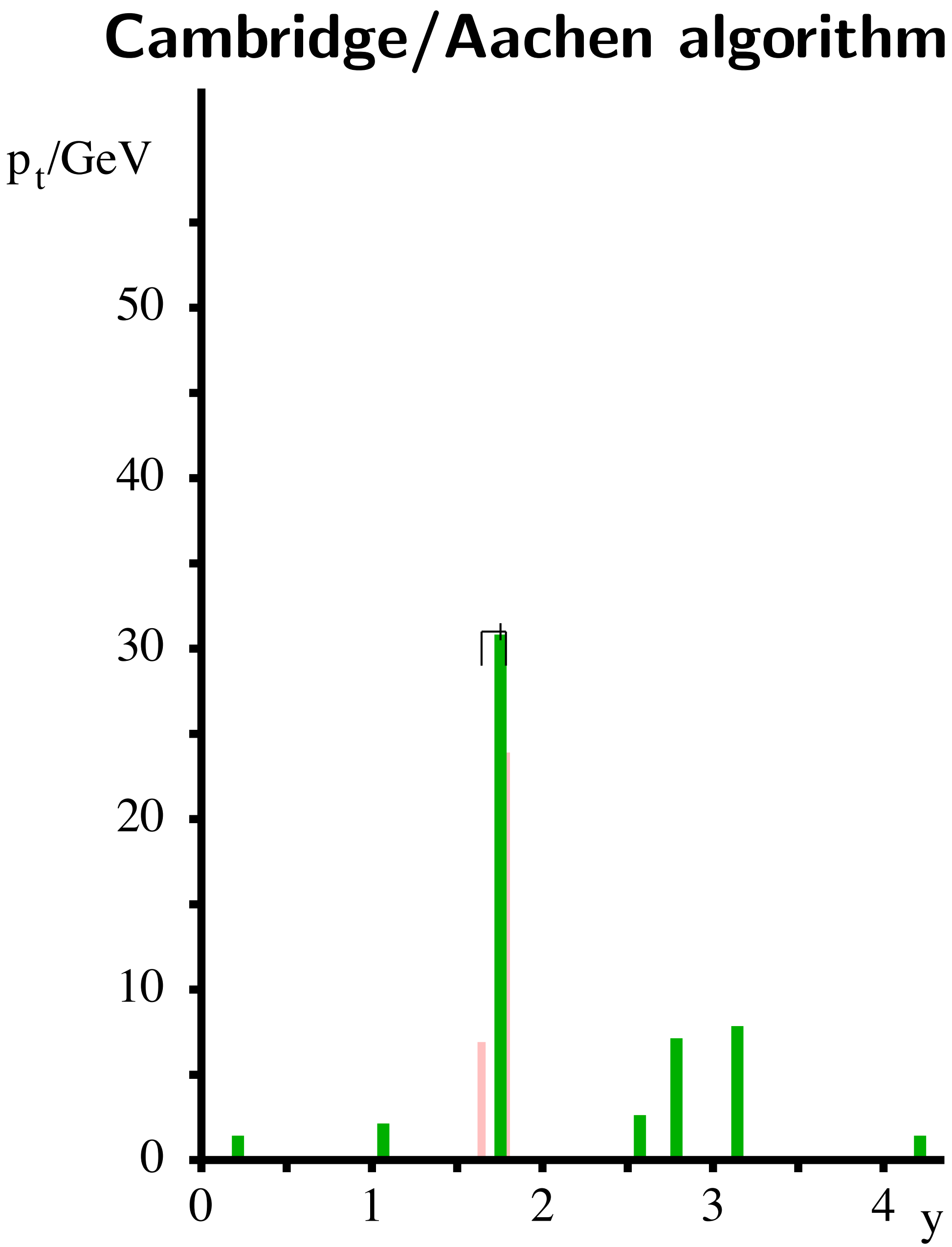
Cluster by merging the **closest** particles



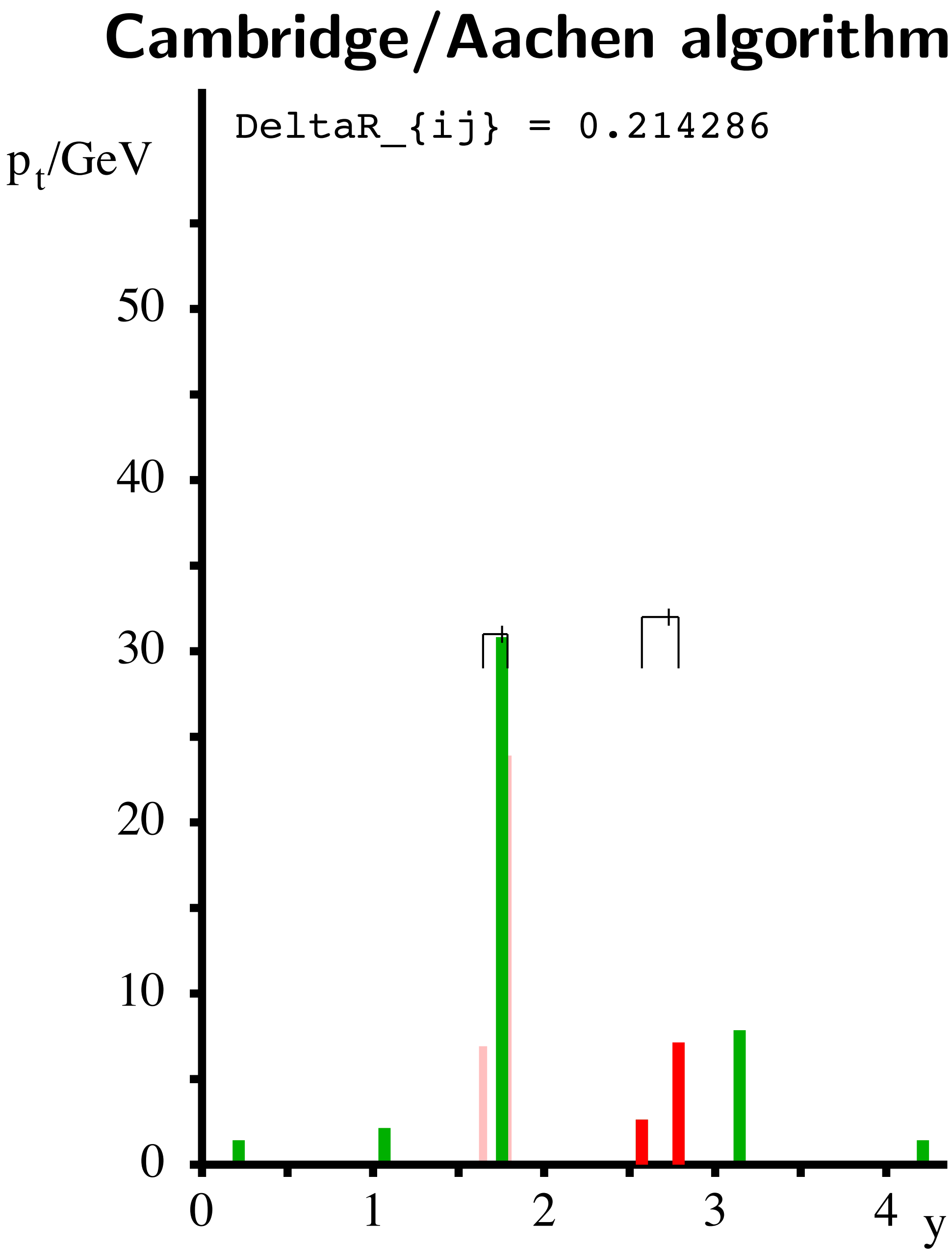
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?



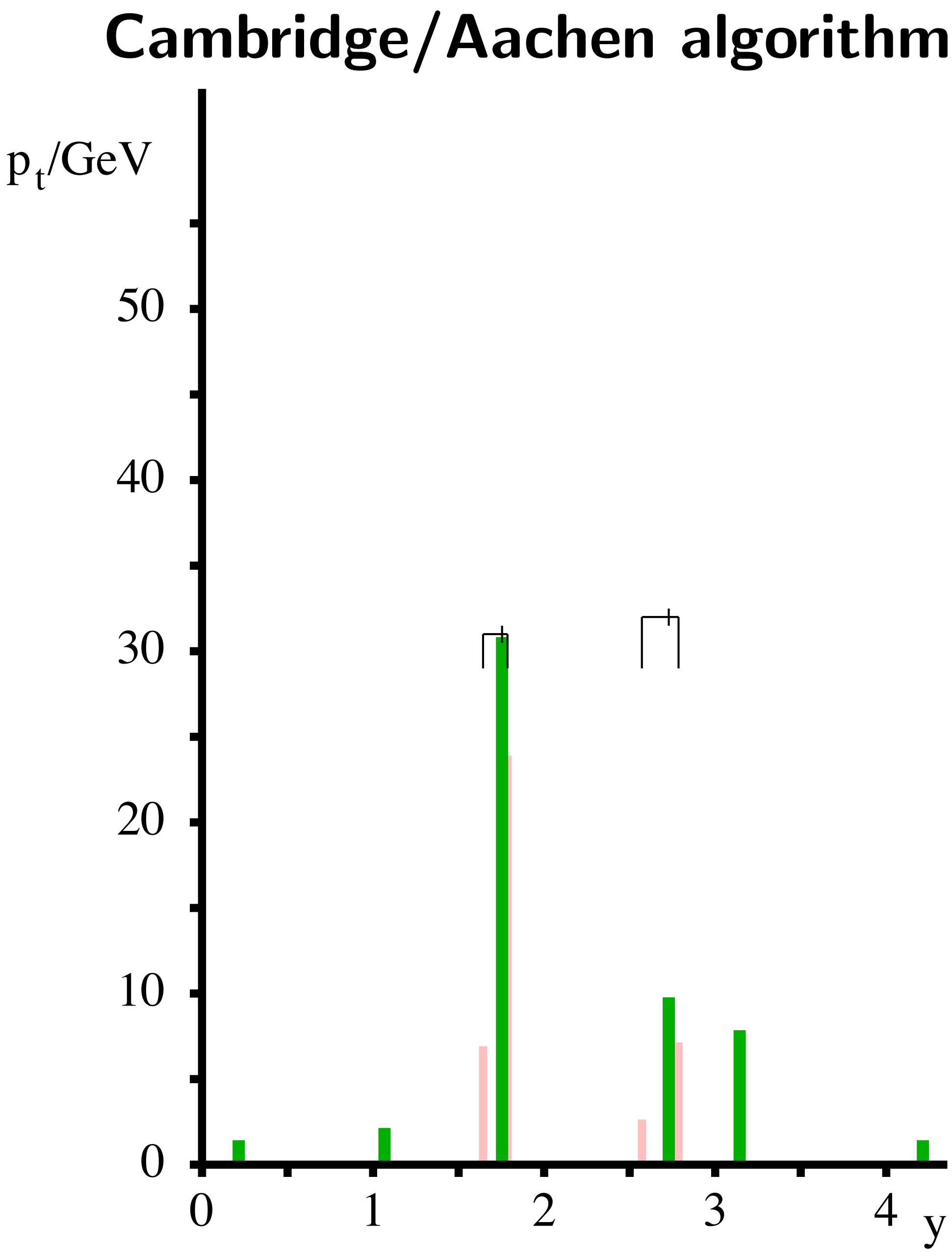
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?



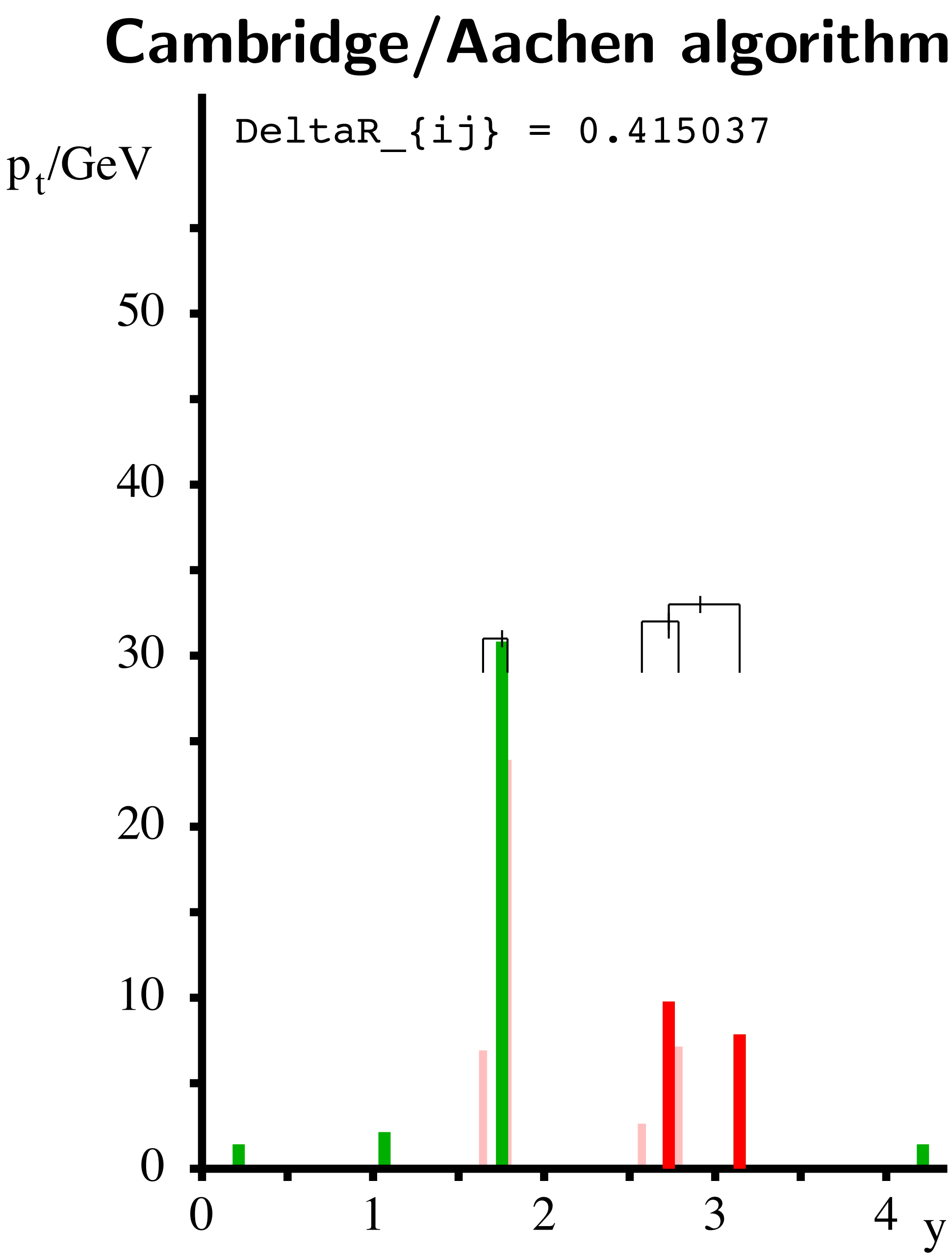
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?



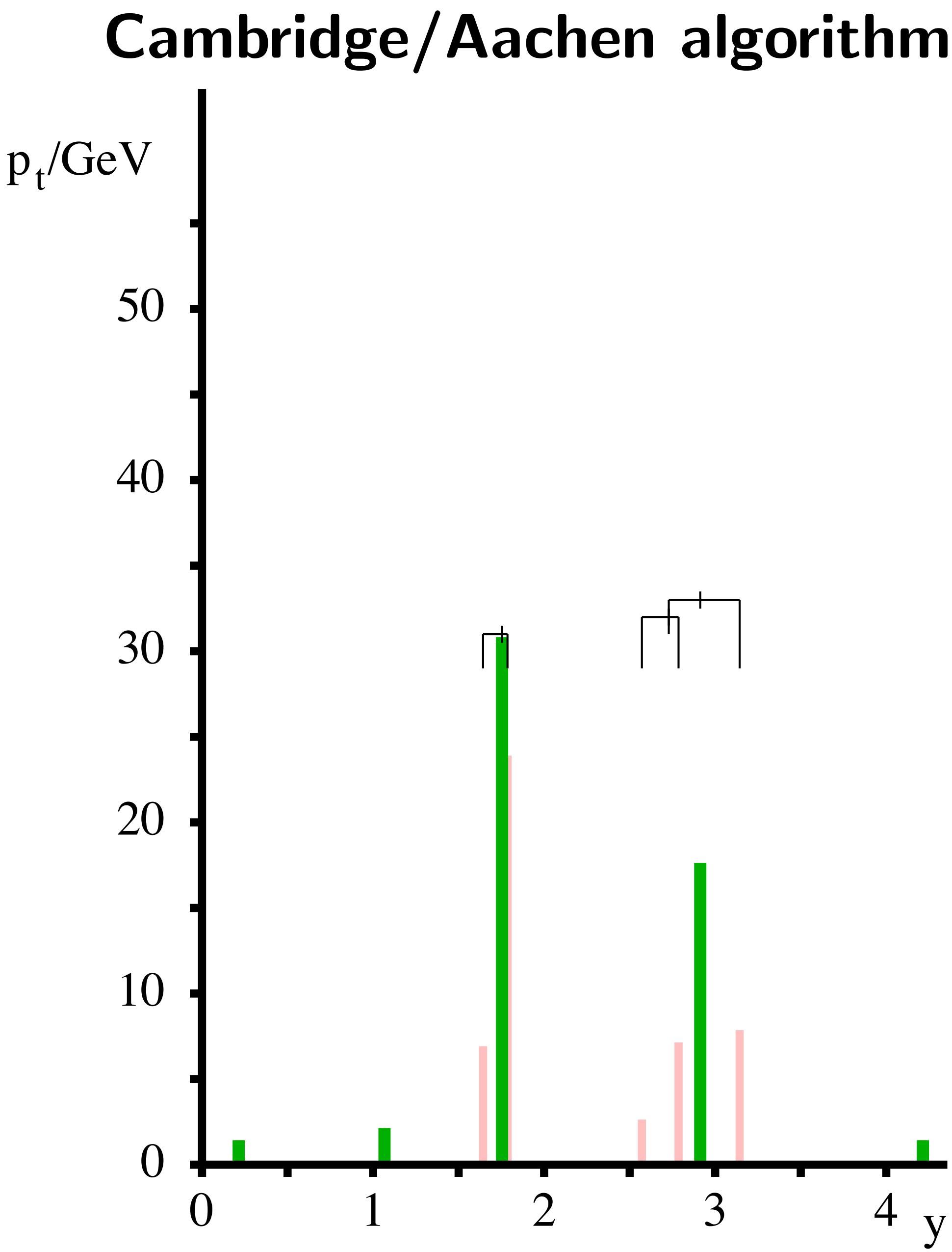
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?



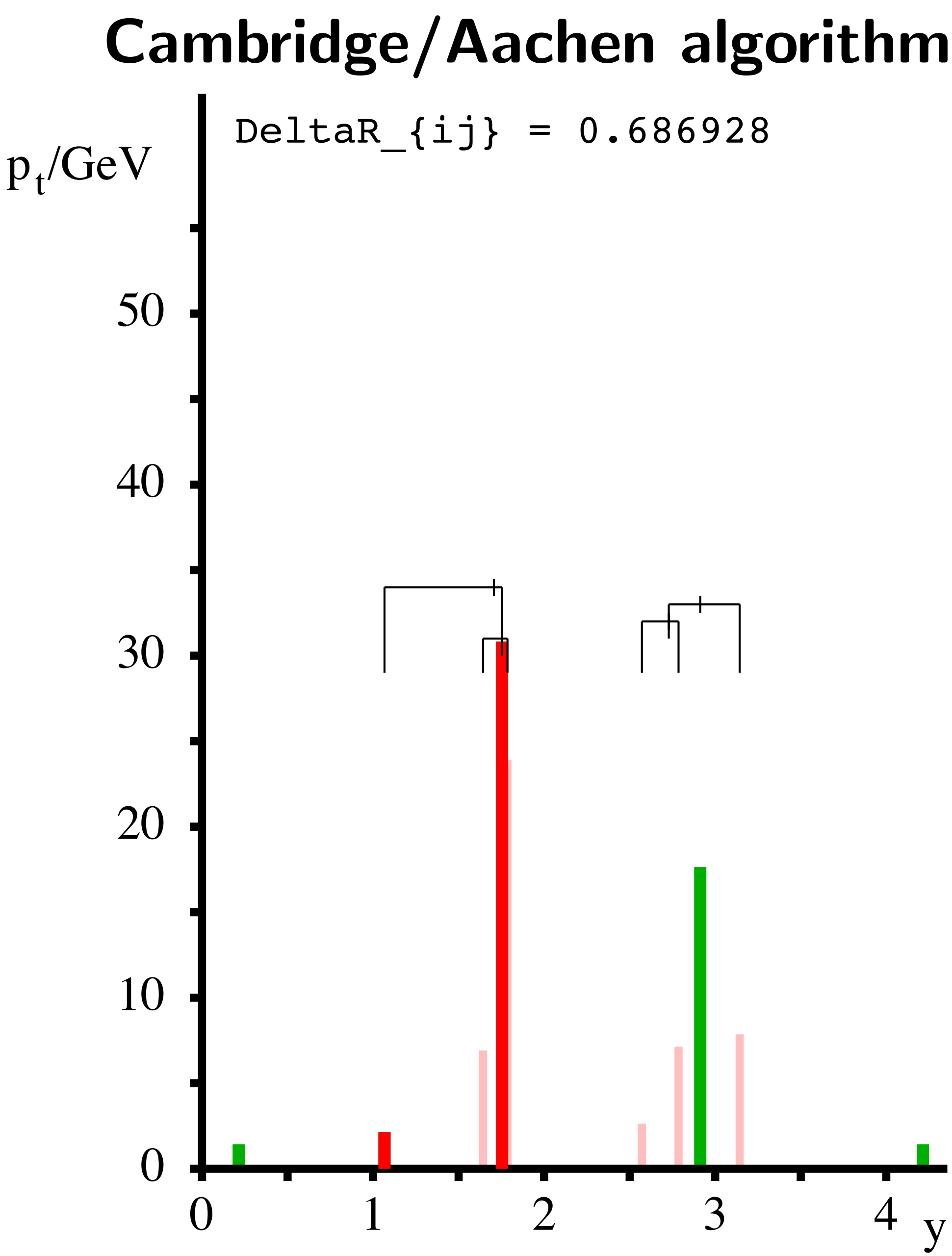
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?



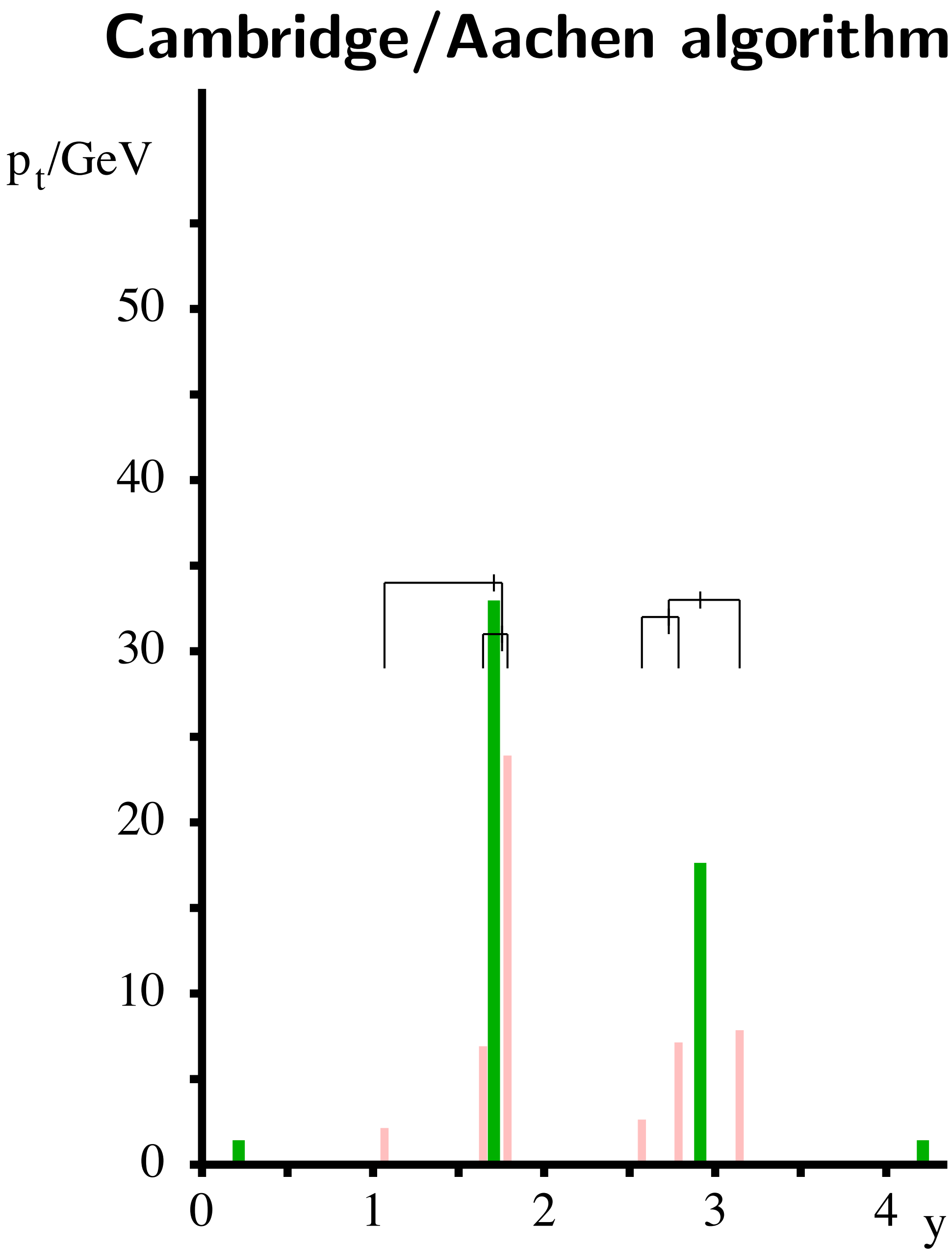
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

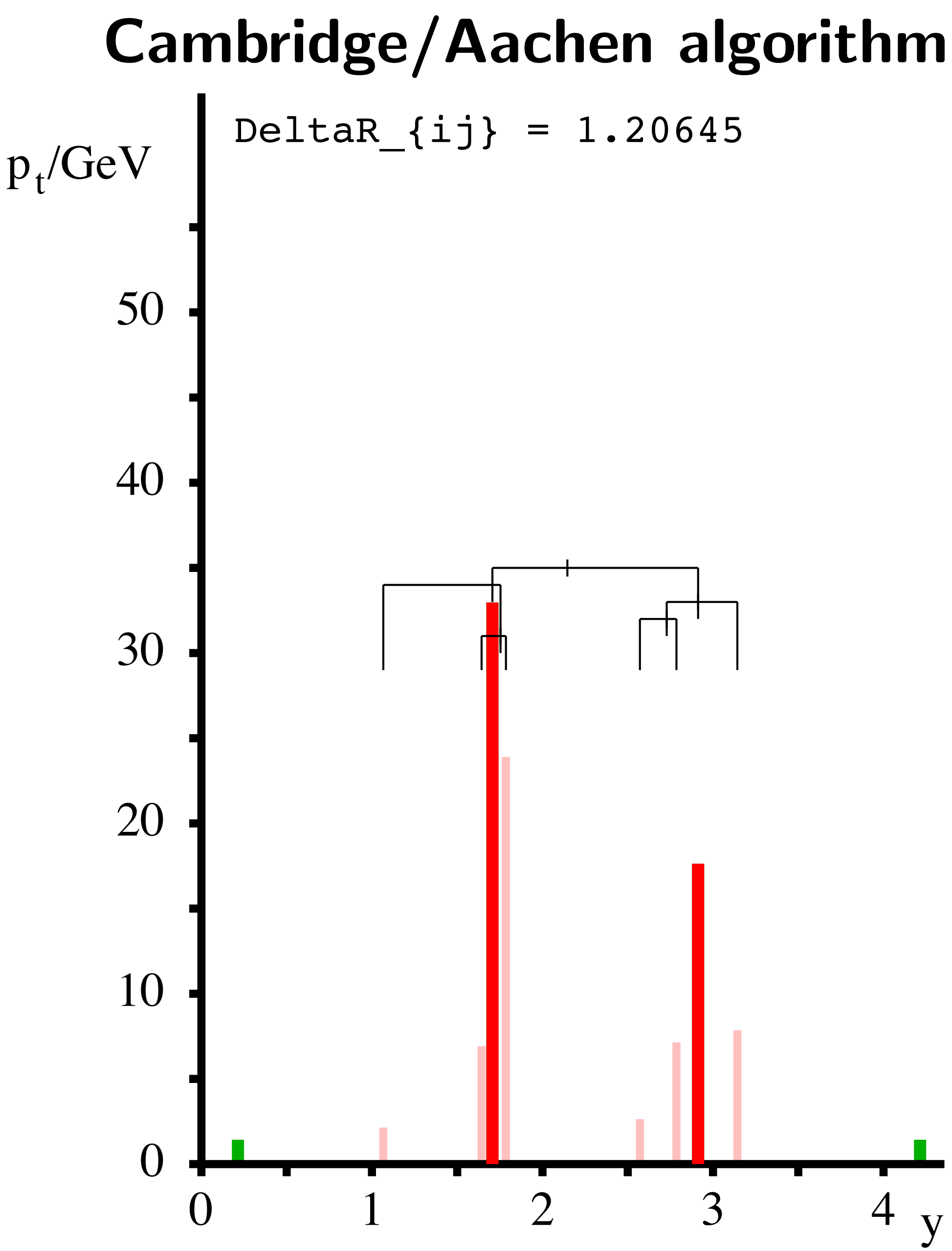


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?



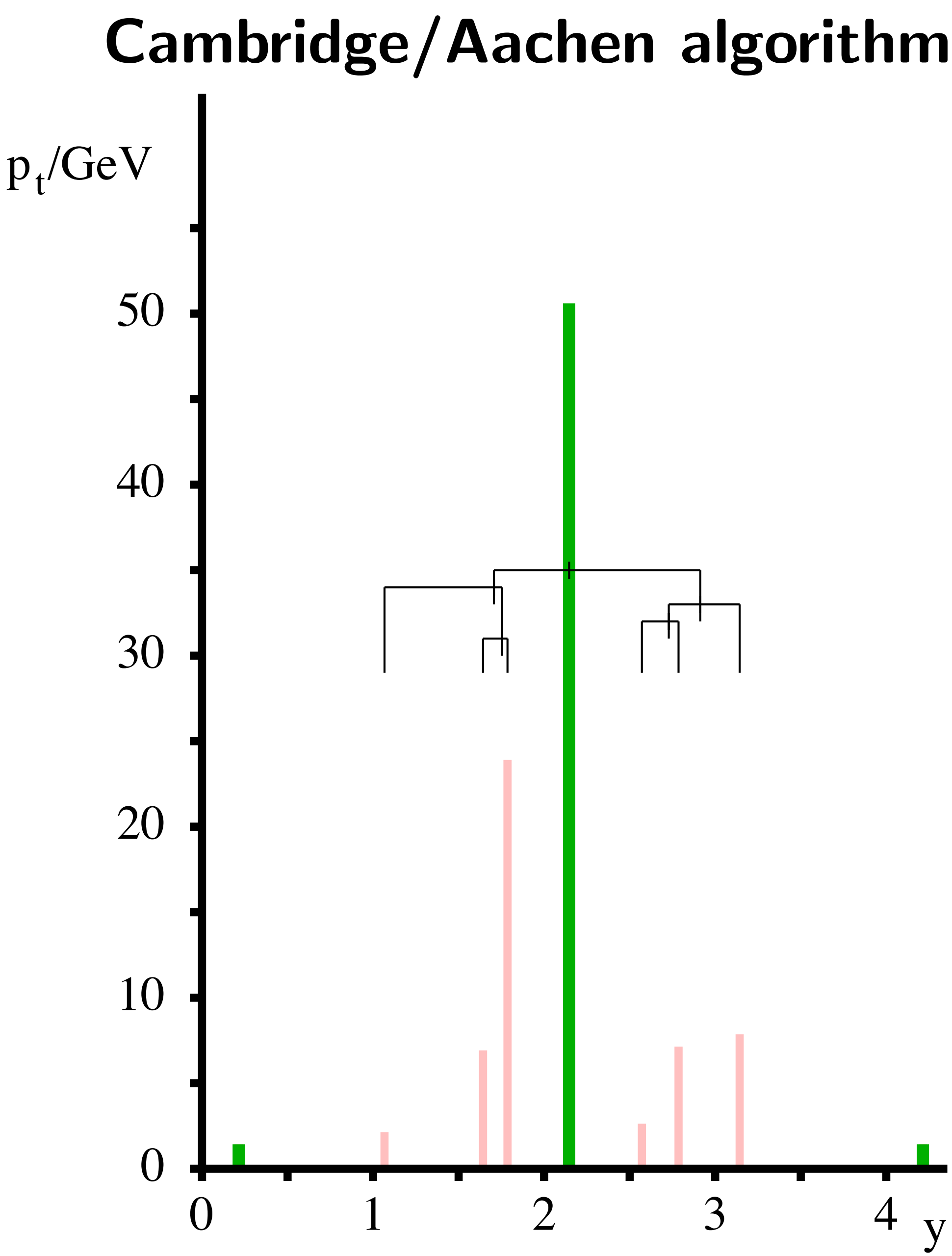
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination



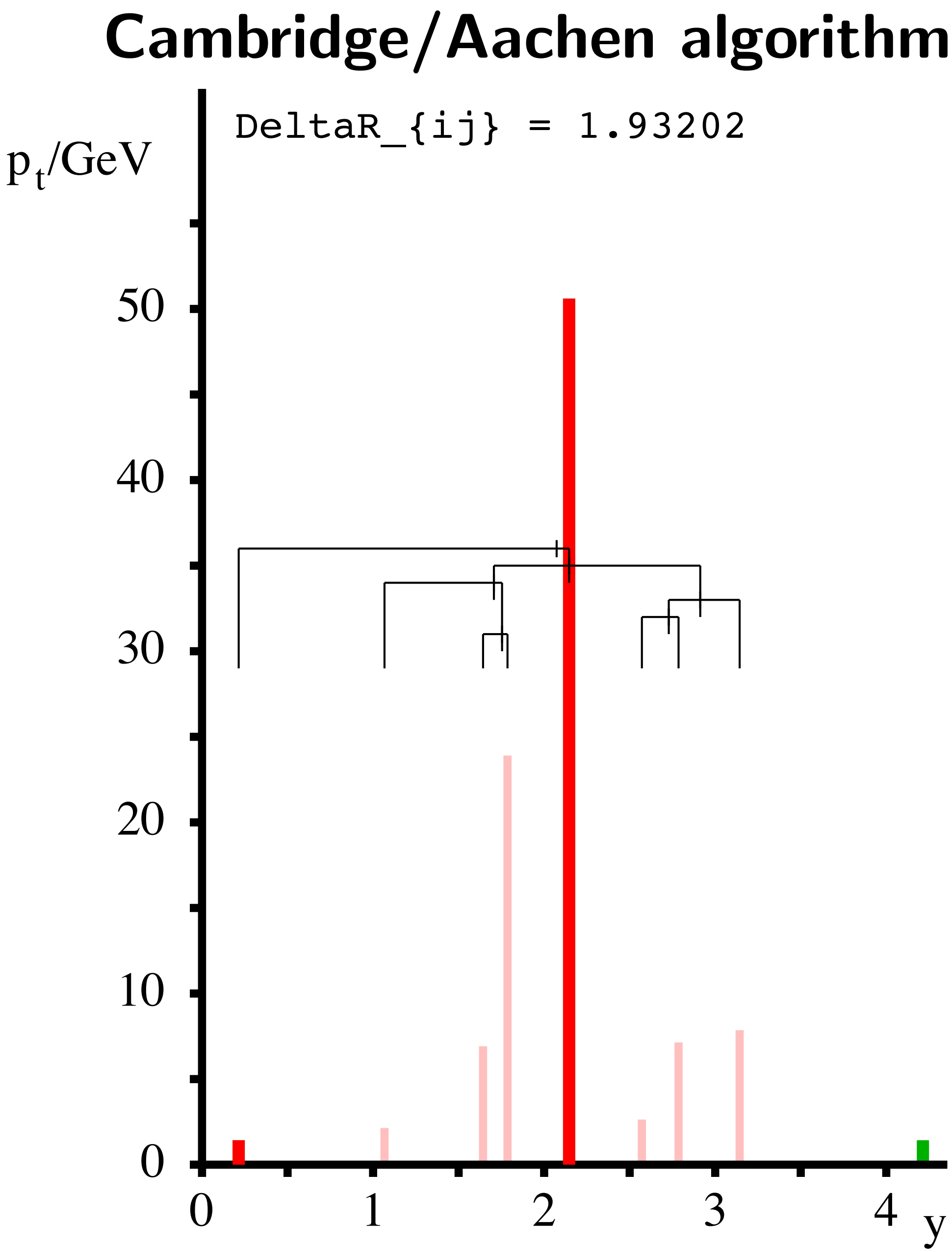
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination, joins them



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

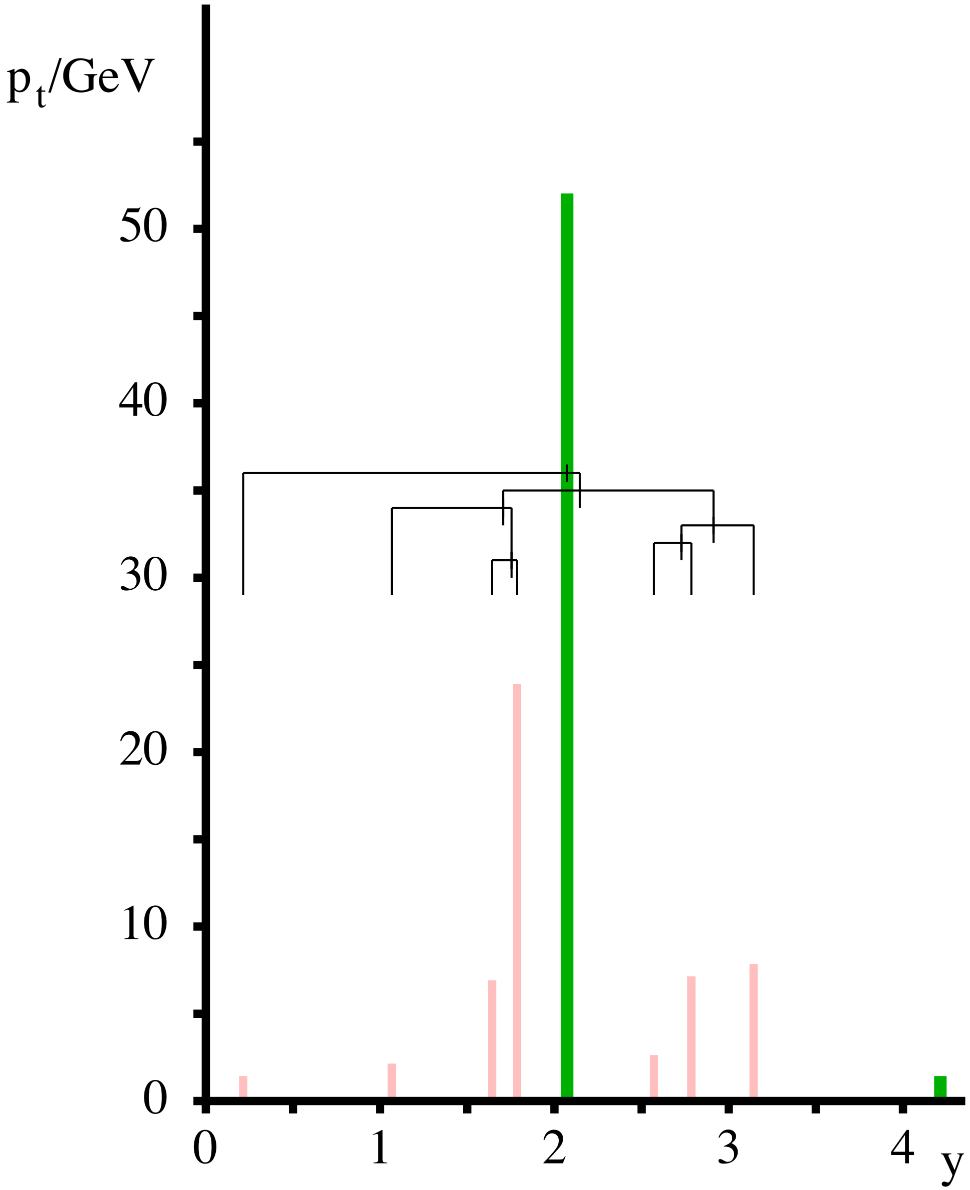
C/A identifies two hard blobs with limited soft contamination, joins them



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk

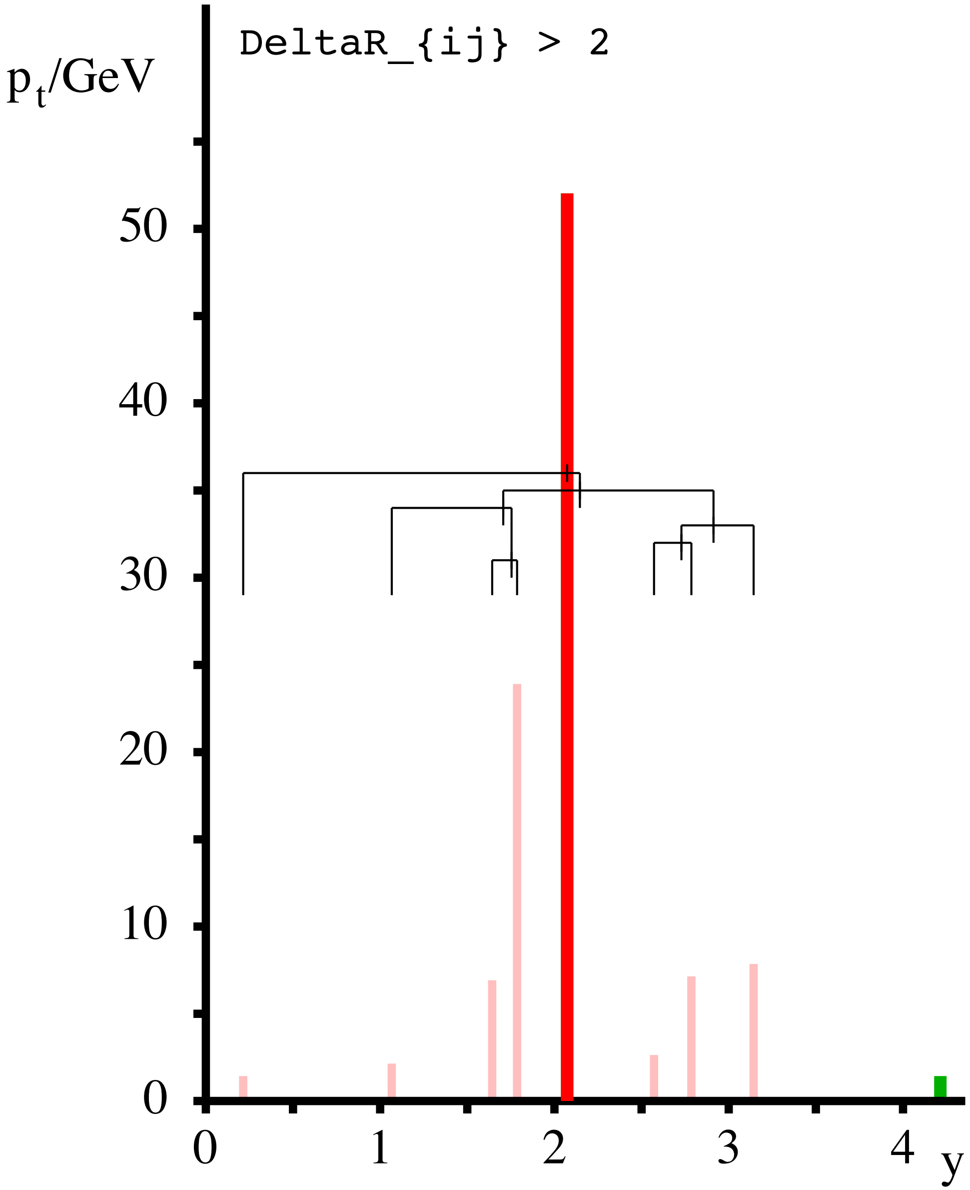
Cambridge/Aachen algorithm



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

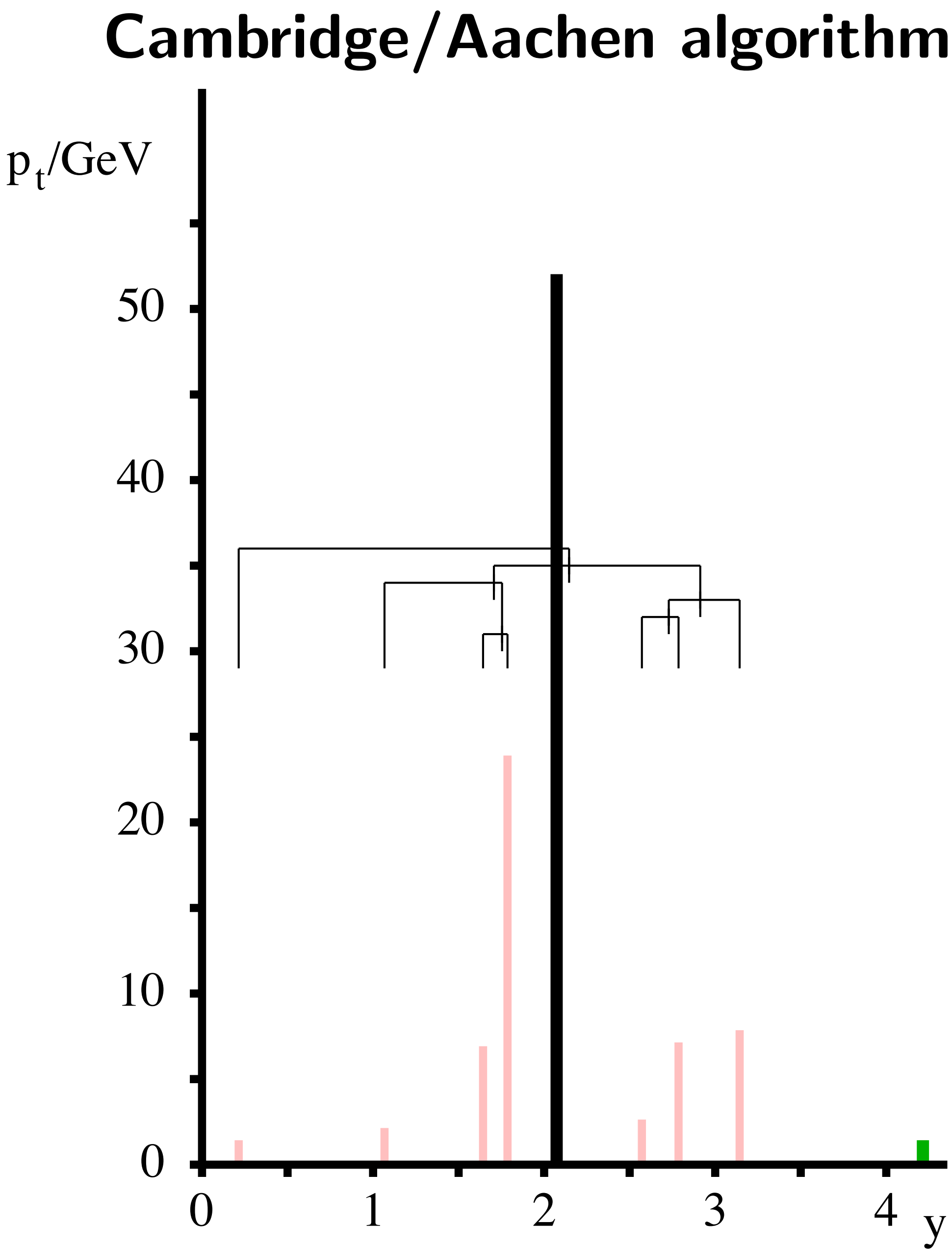
C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk

Cambridge/Aachen algorithm



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

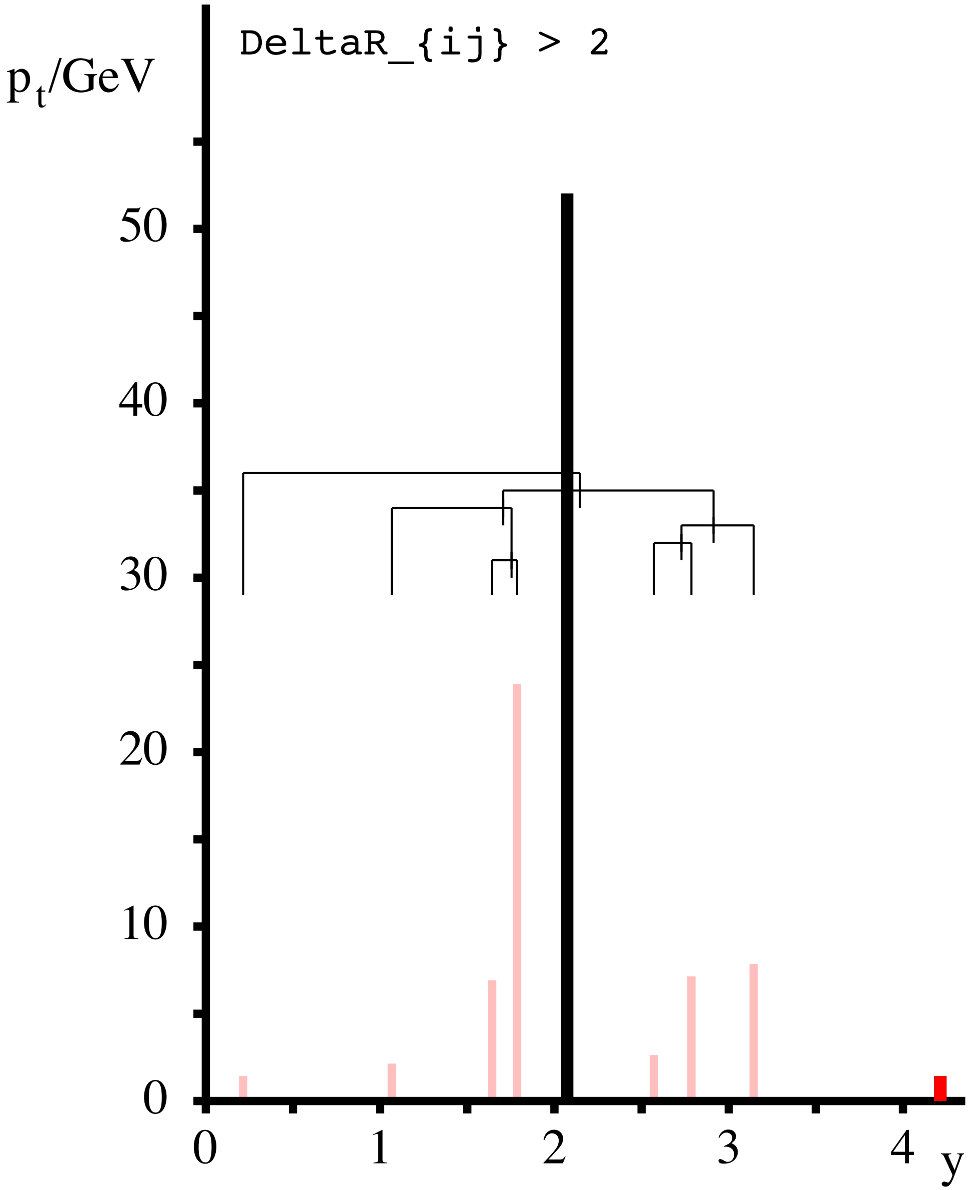
C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

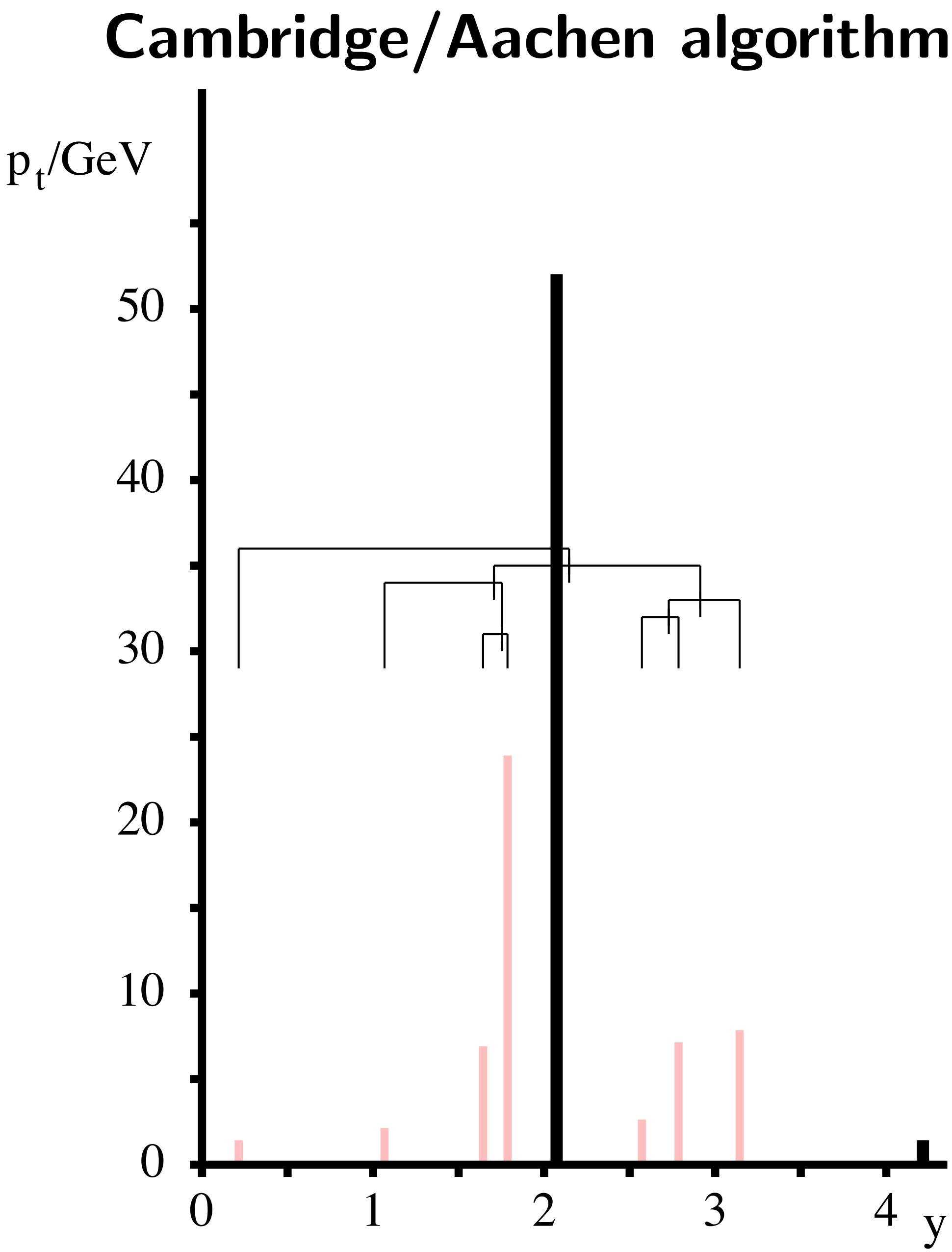
C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk

Cambridge/Aachen algorithm



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk



How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk

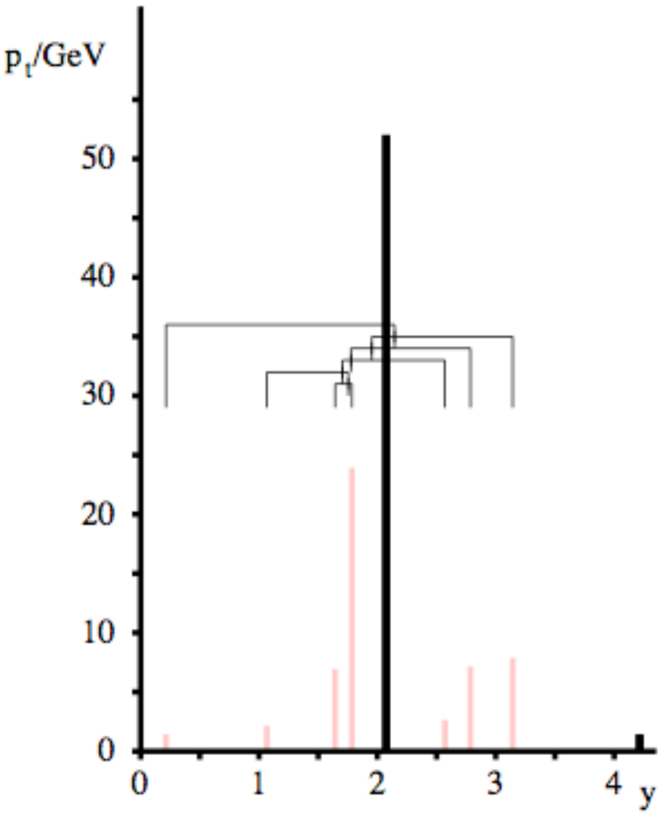
The interesting substructure is buried inside the clustering sequence — it's less contaminated by soft junk, but needs to be pulled out with special techniques

Butterworth, Davison, Rubin & GPS '08
Kaplan, Schwartz, Reherman & Tweedie '08
Butterworth, Ellis, Rubin & GPS '09
Ellis, Vermilion & Walsh '09

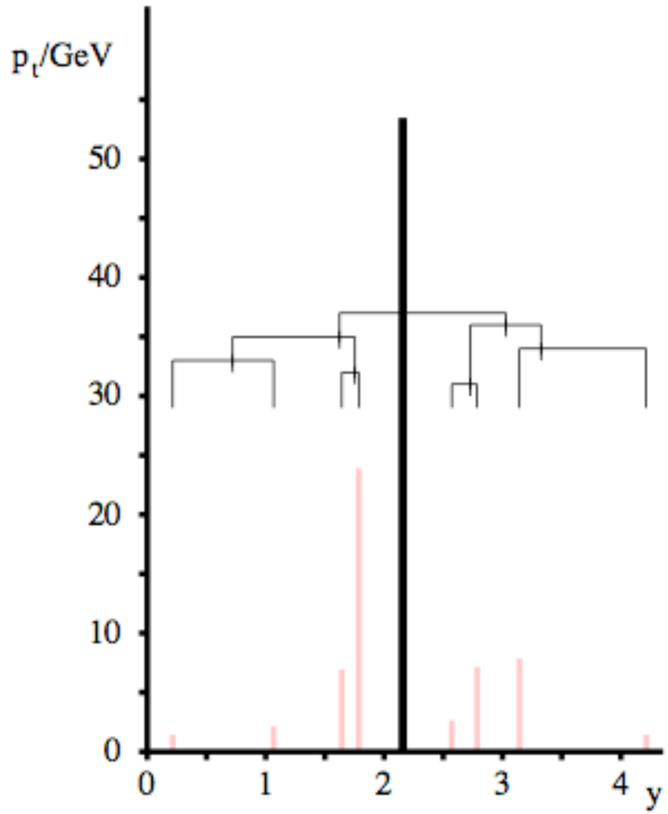
Slide by
Gavin Salam

Hierarchical substructure

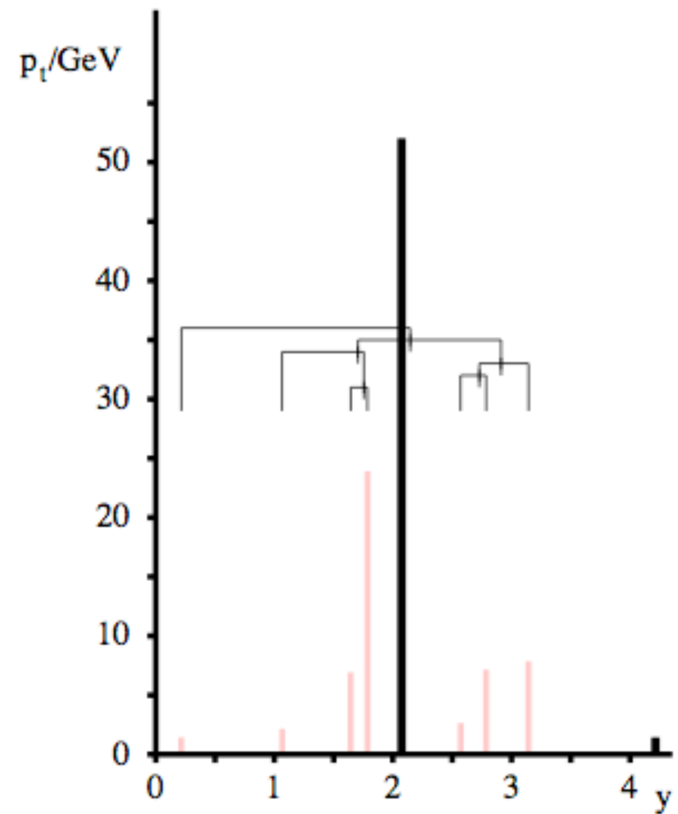
anti-k_t algorithm



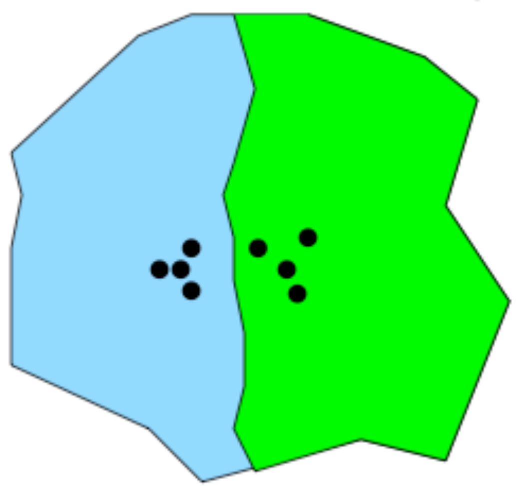
k_t algorithm

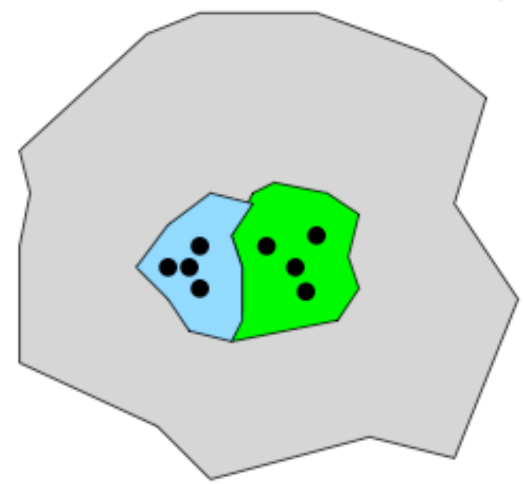


Cambridge/Aachen



Undo the last clustering step(s) →





The IRC safe algorithms

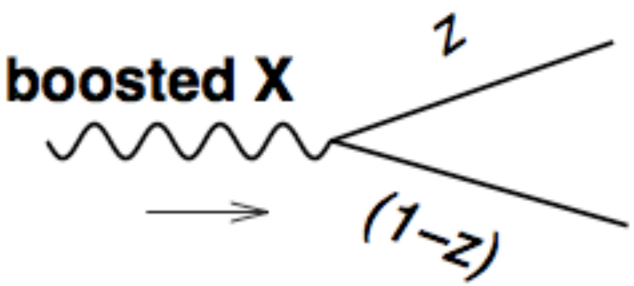
	Speed	Regularity	UE contamination	Backreaction	Hierarchical substructure
k_t	☺ ☺ ☺	☂	☂☂	☁☁	☺☺ ☺☺
Cambridge /Aachen	☺ ☺ ☺	☂	☂	☁☁	☺ ☺ ☺
anti-k_t	☺ ☺ ☺	☺☺ ☺☺	☁/☺	☺☺ ☺☺	✗
SISCone	☺☺	☁	☺☺ ☺☺	☁	✗

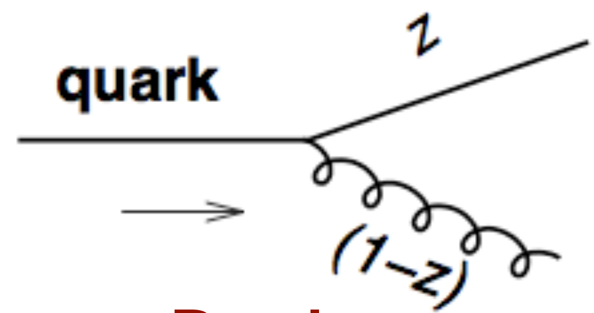
Array of tools with different characteristics.

Pick the right one for the job

QCD v. heavy decay

A possible approach for reducing the QCD background is to identify the two prongs of the heavy particle decay, and put a cut on their momentum fraction





Signal:

$$P(z) \sim 1$$

Will split mainly **symmetrically**

Background:

$$P(z) \sim \frac{1+z^2}{1-z} \qquad P(z) \sim \frac{1+(1-z)^2}{z}$$

Will split mainly **asymmetrically**

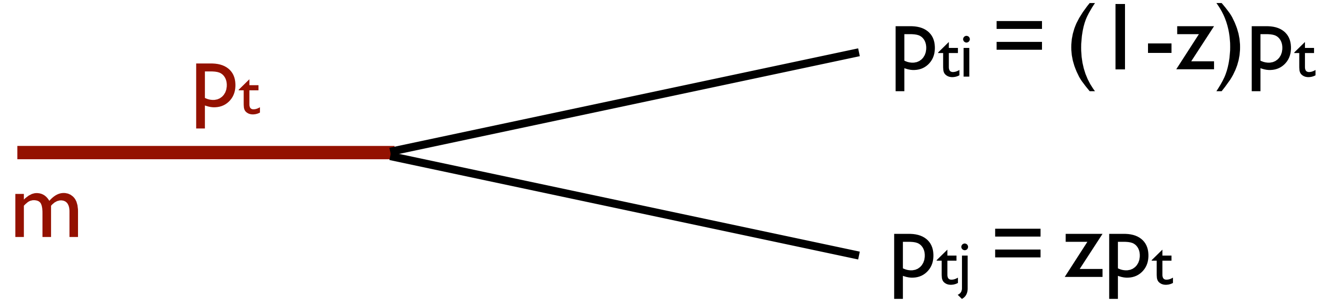
Potential tagger: asymmetric splitting

Possibly implemented via a cut on

$$y = min(p_{ti}^2, p_{tj}^2)\frac{\Delta R_{ij}^2}{m^2} \simeq \frac{min(p_{ti}, p_{tj})}{max(p_{ti}, p_{tj})}$$

Splittings and distances

Quasi-collinear splitting ($p_{tj} < p_{ti}$)



Invariant mass: $$m^2 \simeq p_{ti} p_{tj} \Delta R_{ij}^2 = (1-z) z p_t^2 \Delta R_{ij}^2$$

k_t distance: $$d_{ij} \overset{(p_{tj} < p_{ti})}{=} z^2 p_t^2 \Delta R_{ij}^2 \simeq \frac{z}{1-z} m^2$$

For a given mass, the **background** will have smaller distance d_{ij} than the signal, i.e. it will tend to **cluster earlier** in the k_t algorithm

Potential tagger: last clustering in k_t algorithm

This is where the hierarchy of the k_t algorithm becomes relevant.
QCD radiation is clustered first, and only at the end the symmetric, large-angle splittings due to decays are reclustered

Alternative algorithms

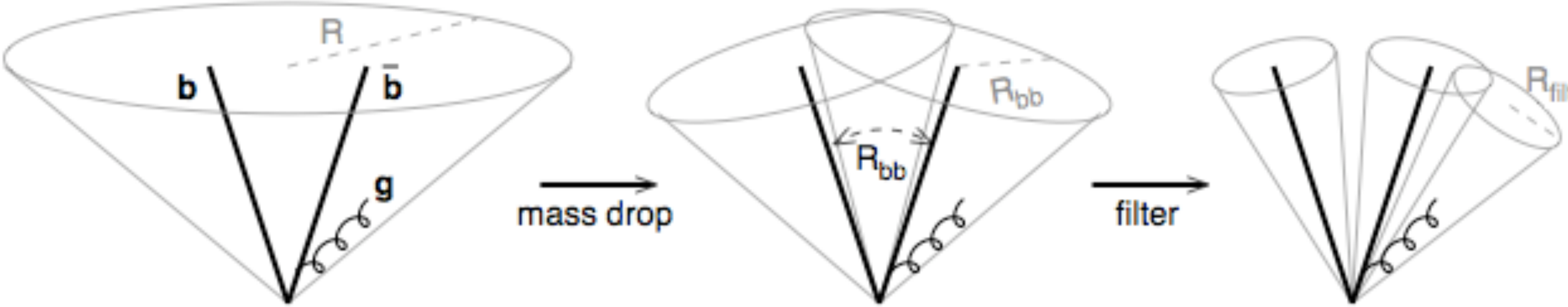
- Suppose that for some reasons (which will become clearer later) one does not with to use the k_t algorithm
  - *One must then find a way to determine what the* ***relevant splitting*** *(i.e. the one due to the decay, not to QCD radiation) is.*

A possible approach is to use a Mass-Drop requirement:
the clustering is **progressively undone**, and a splitting is the relevant one if both subjects are much less massive than their combination

$pp \to ZH \to \nu\bar{\nu}b\bar{b}$

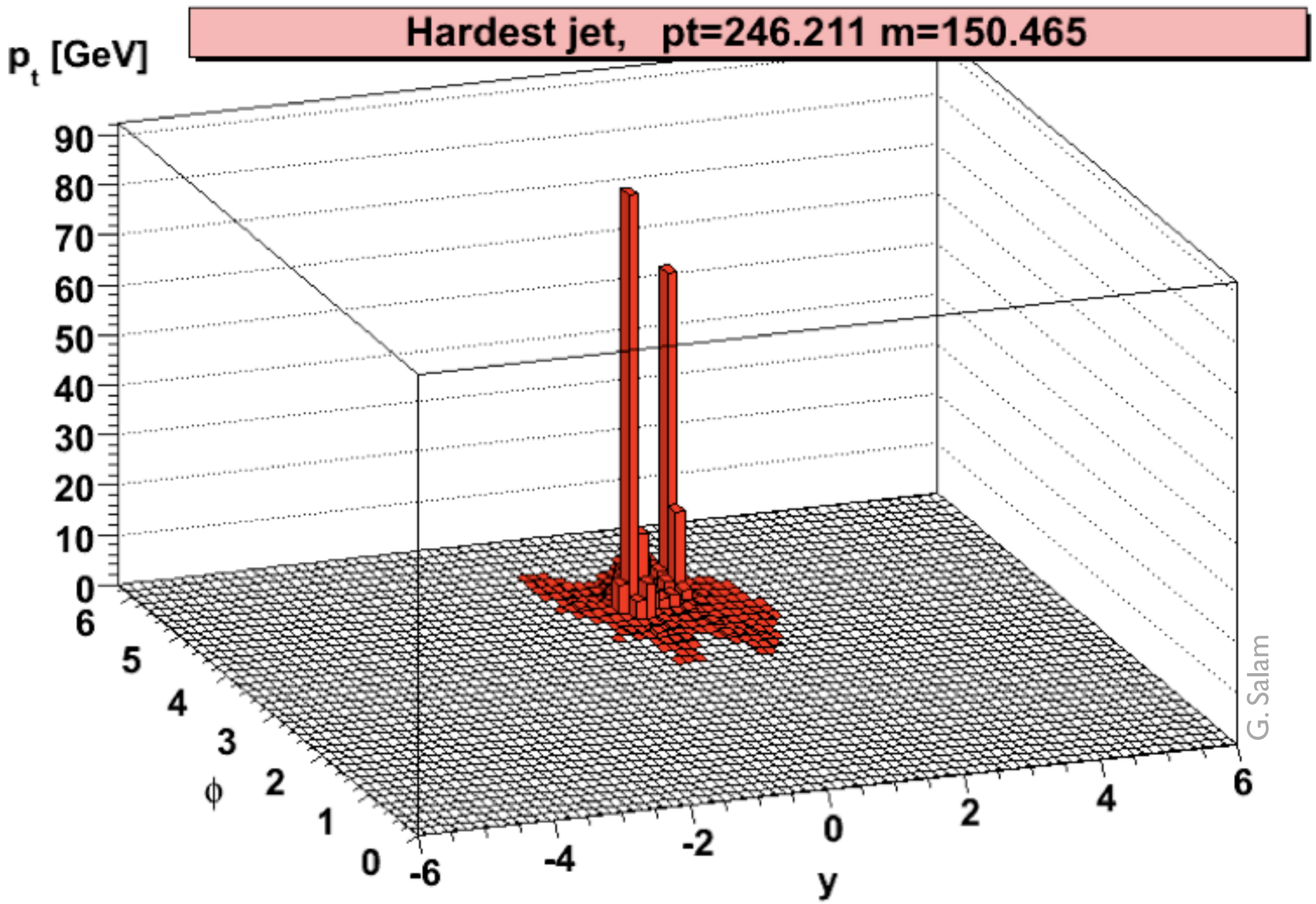
The BDRS tagger/groomer

Butterworth, Davison, Rubin, Salam, 2008



- A two-prong tagger/groomer for boosted Higgs, which
  - Uses the **Cambridge/Aachen** algorithm (because it's 'physical')
  - Employs a **Mass-Drop** condition, as well as an **asymmetry cut** to find the **relevant splitting** (i.e. '**tag**' the heavy particle)
  - Includes a post-processing step, using 'filtering' (introduced in the same paper) to clean as much as possible the resulting jets of UE contamination ('**grooming**')

$pp \to ZH \to \nu\bar{\nu}b\bar{b}$

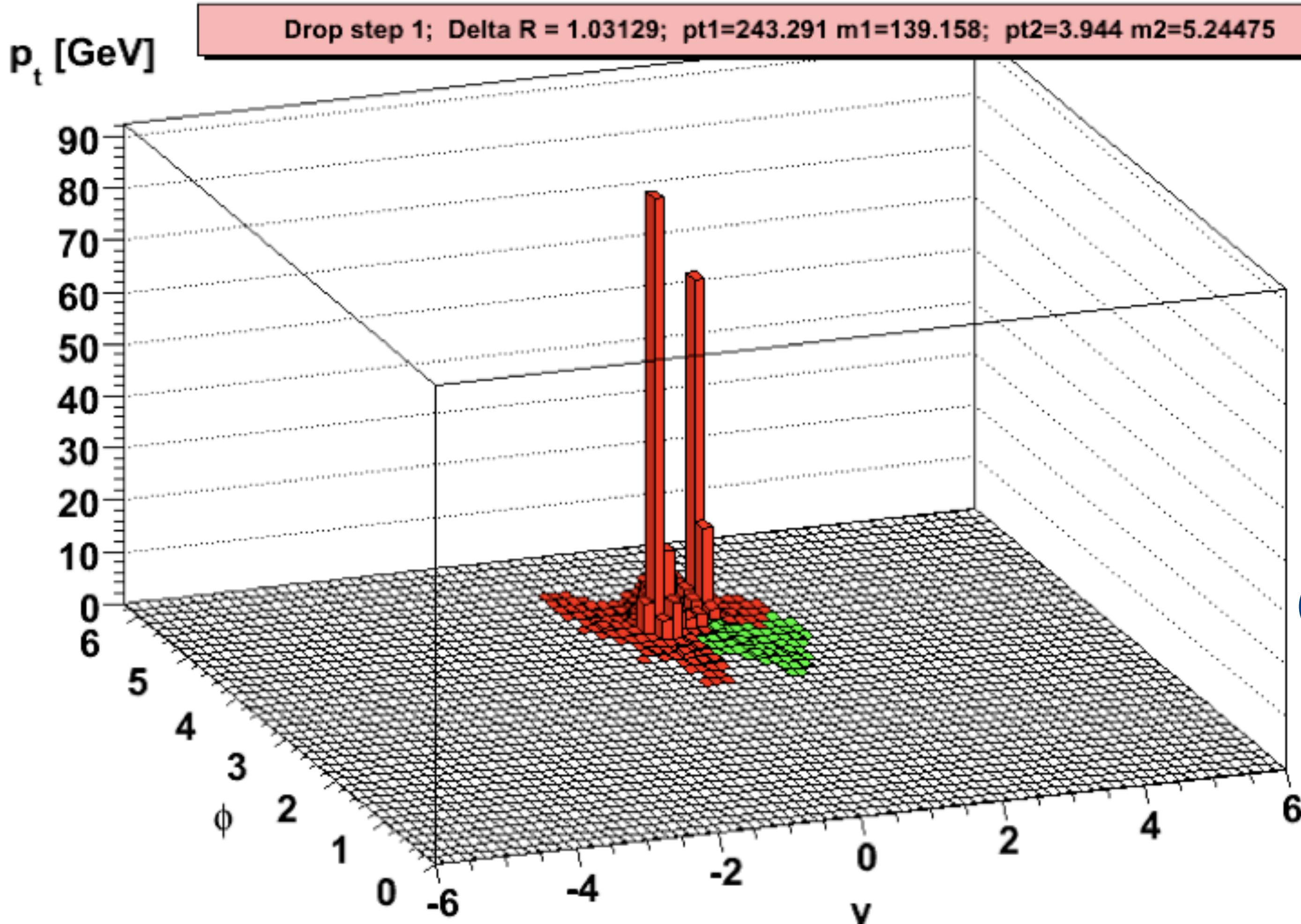


Start with the hardest jet

Use C/A with large R=1.2

m_j = 150 GeV

$pp \to ZH \to \nu\nu bb$

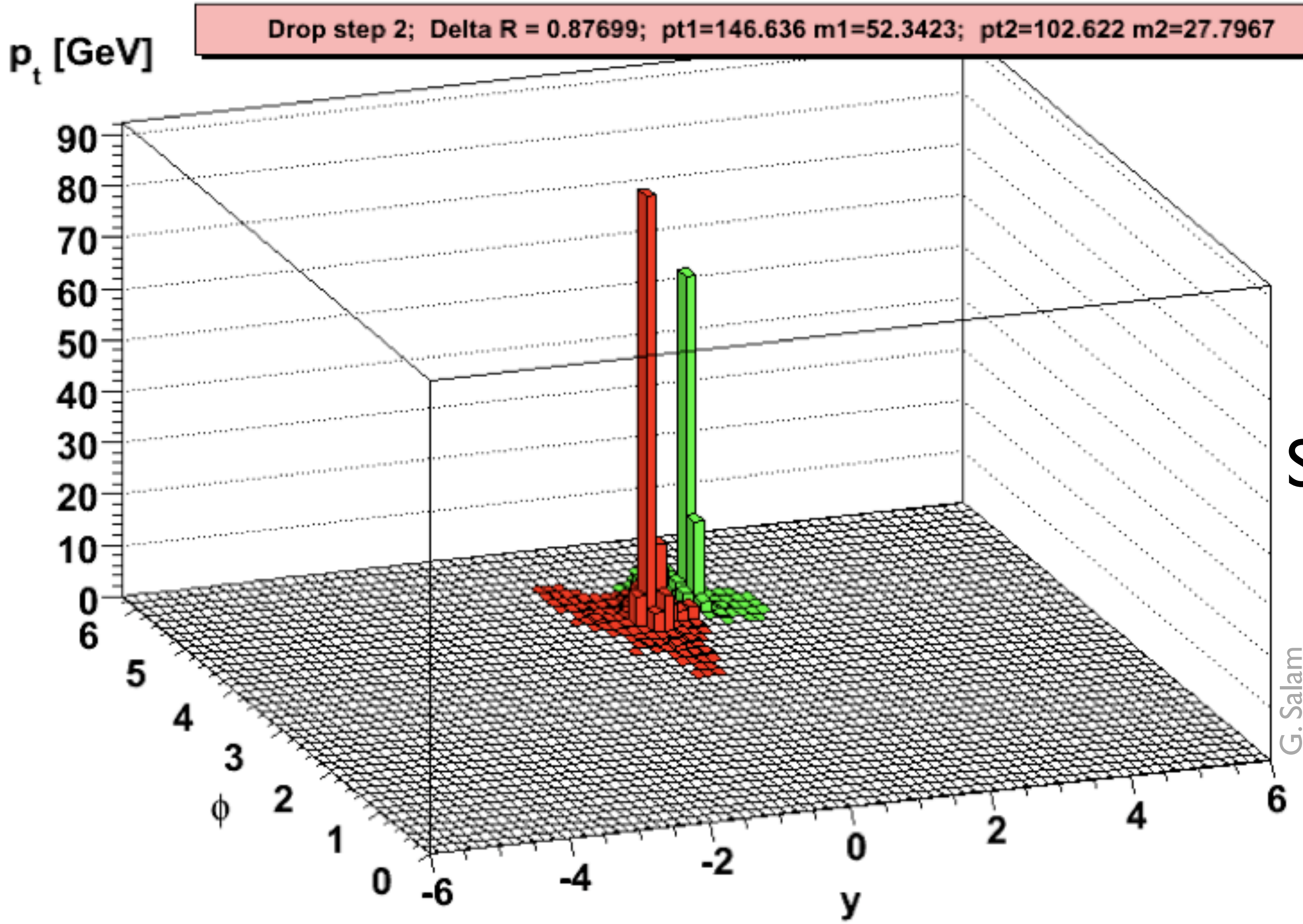


Undo last step of clustering

Check how the mass splits between the two subjets (m_1 = 139 GeV, m_2 = 5 GeV) and how asymmetric the splitting is

If $\frac{max(m_1, m_2)}{m_j} > \mu$ or $\frac{min(p_{t1}^2, p_{t2}^2)}{m_j^2} \Delta R_{12}^2 < y_{cut}$ repeat

$pp \to ZH \to \nu\nu bb$



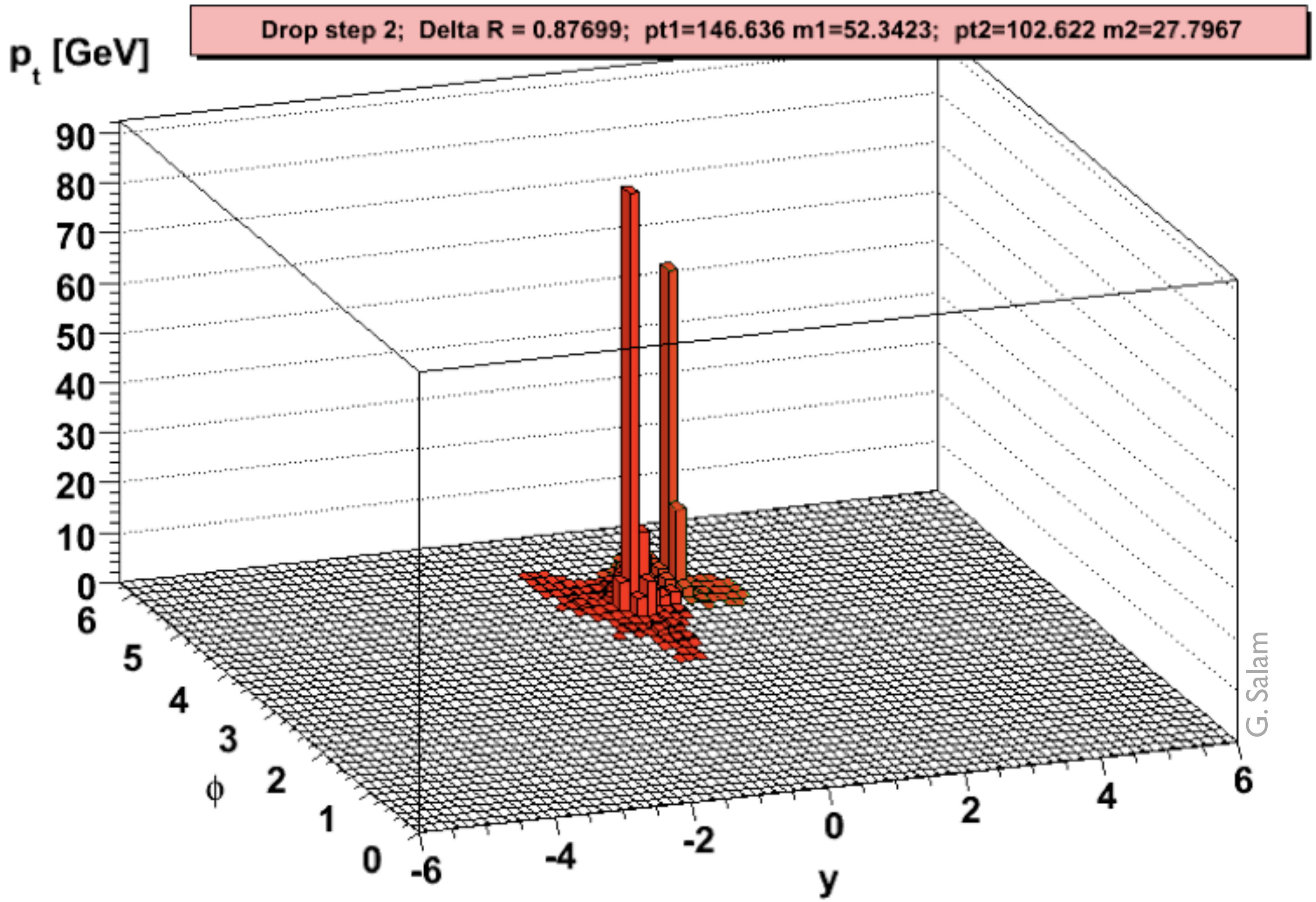
m_1 = 52 GeV, m_2 = 28 GeV

Stop when a large mass drop is observed

(and **recombine** these two jets)

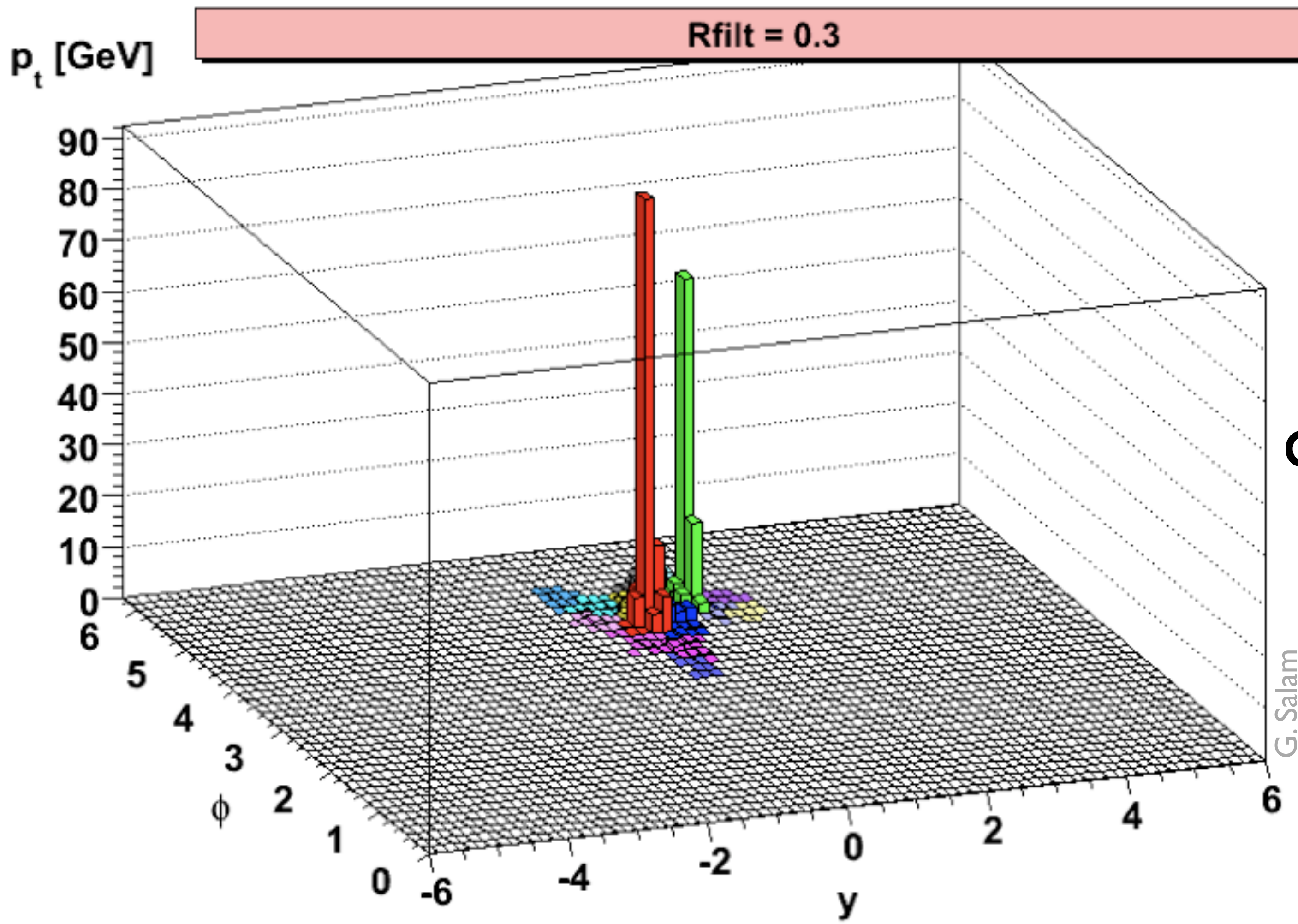
[NB. Parameters used μ = 0.67 and y_{cut} = 0.09]

pp → ZH → ννbb



Start with the recombined jet

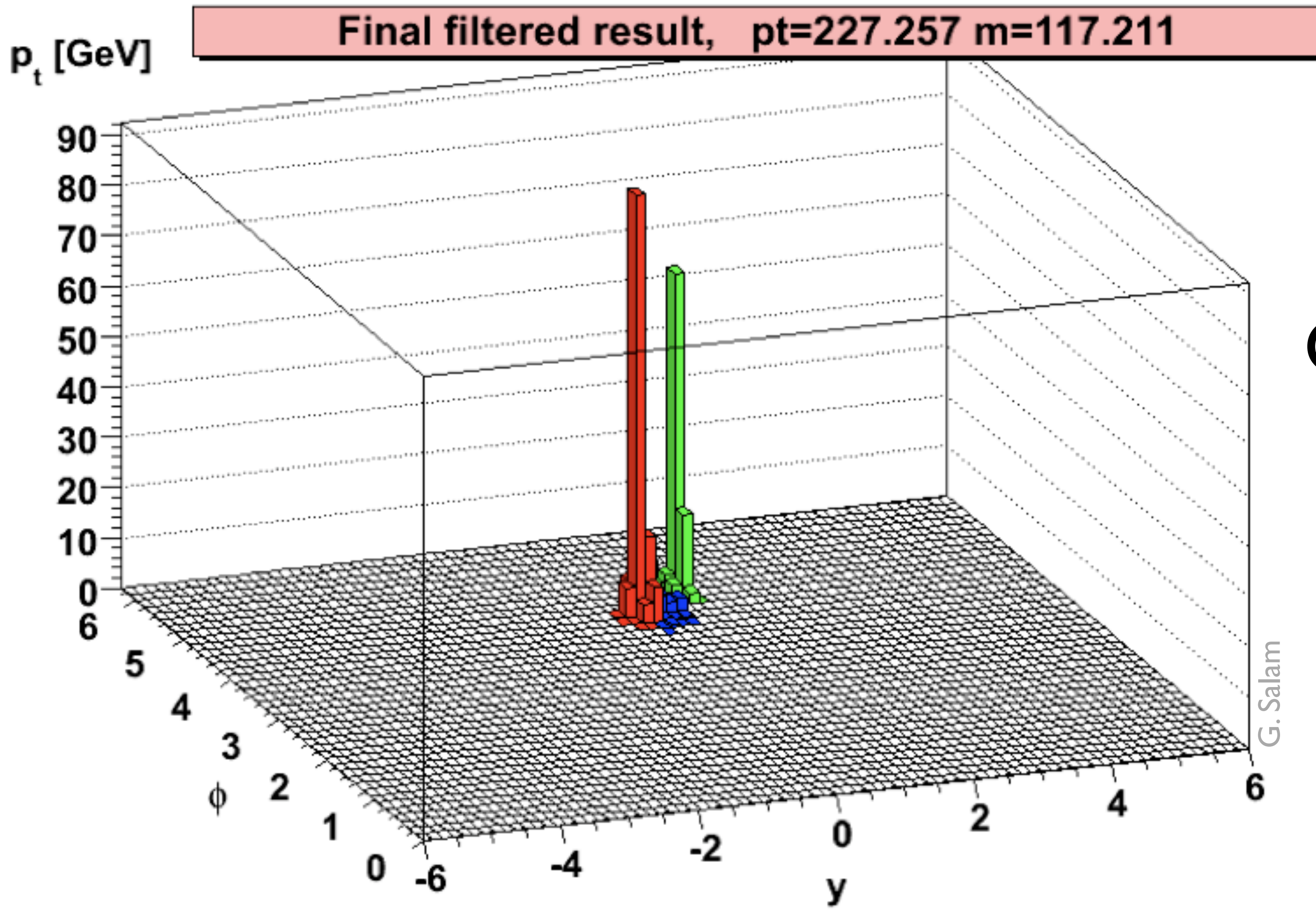
$pp \to ZH \to \nu\nu bb$



Recluster the contituents with R_{filt}

BDRS: filtering

$pp \rightarrow ZH \rightarrow \nu\nu bb$



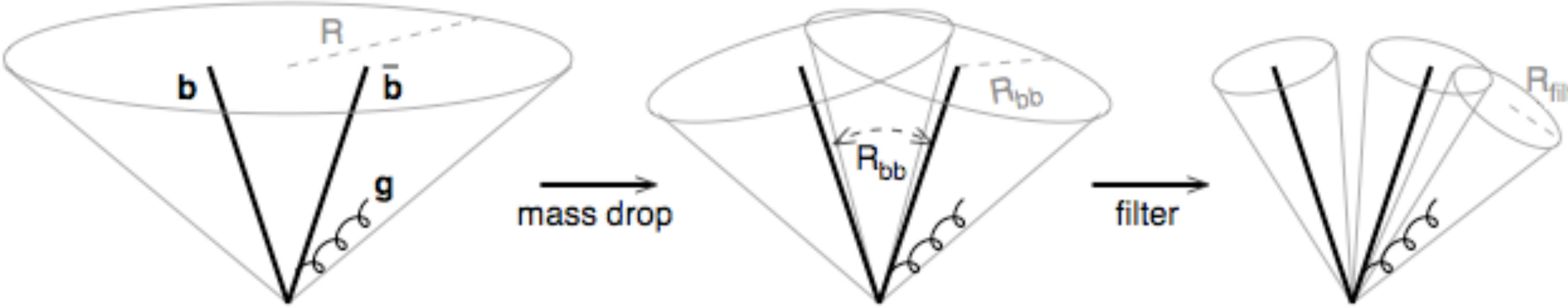
Only keep the n_{filt} hardest jets

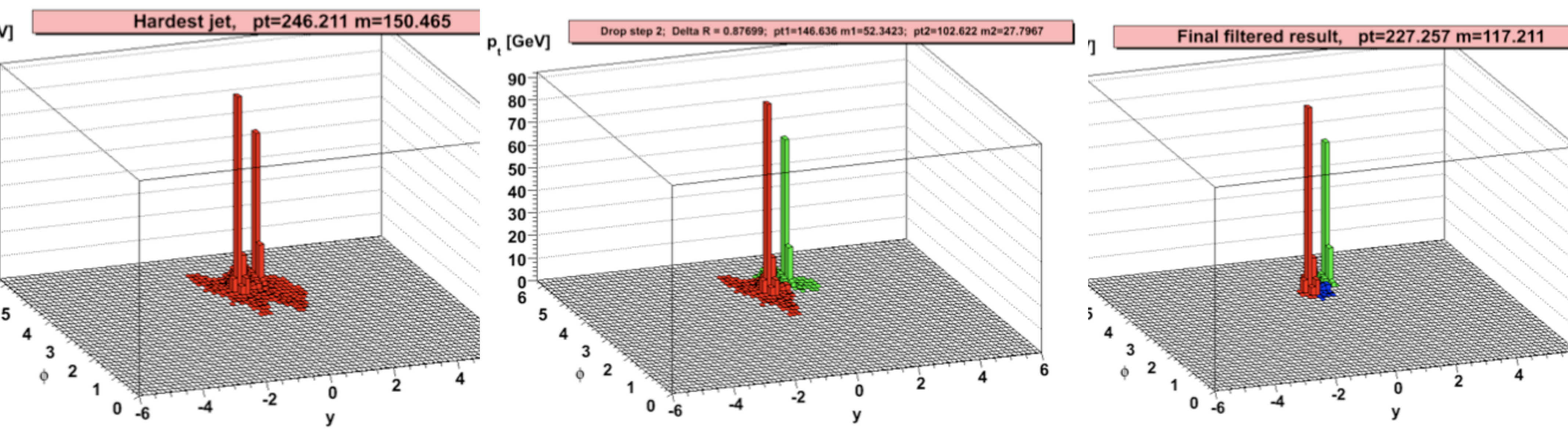
The low-momentum stuff surrounding the hard particles has been removed

$pp \to ZH \to \nu\bar{\nu}b\bar{b}$

Visualisation of BDRS

Butterworth, Davison, Rubin, Salam, 2008





Cluster with a large R

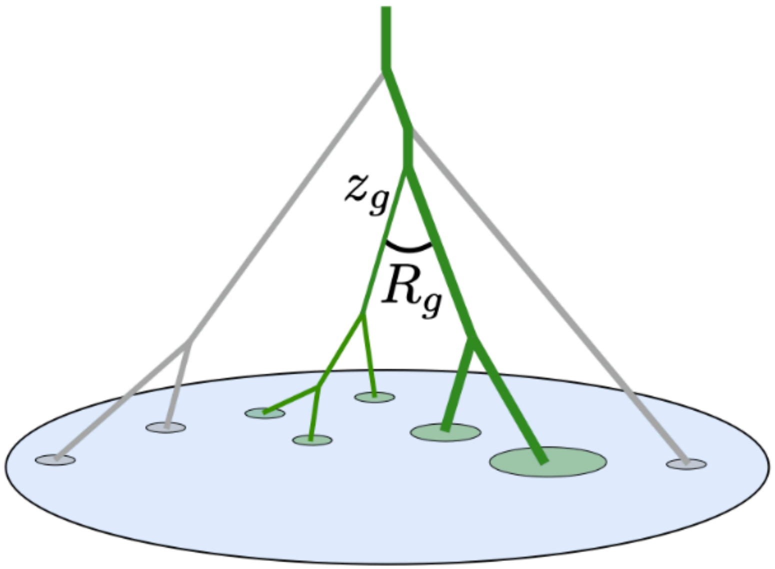
Undo the clustering into subjets, until a large asymmetry/mass drop is observed: tagging step

Re-cluster with smaller R, and keep only 3 hardest jets: grooming step

Larkoski, Marzani, Soyez, Thaler, 2014

A generalisation of the (modified) Mass-Drop tagger.
Progressively decluster and drop constituent unless

$$\text{Soft Drop Condition:} \quad \frac{\min(p_{T1}, p_{T2})}{p_{T1} + p_{T2}} > z_{\rm cut} \left(\frac{\Delta R_{12}}{R_0} \right)^{\beta}$$



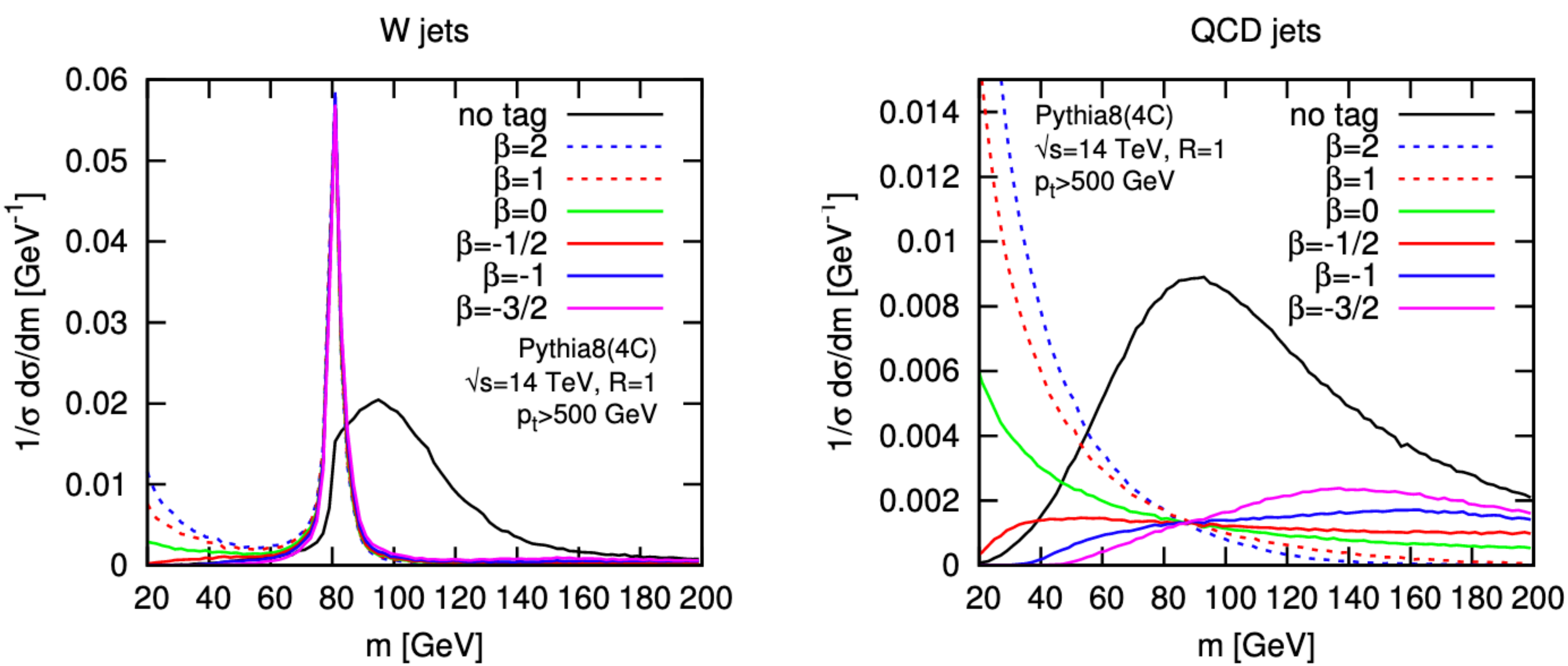
[Drawing from 2106.04589]

i.e. remove large-angle soft radiation from a jet

Larkoski, Marzani, Soyez, Thaler, 2014

The paper contains

- ✓ analytical calculations and comparisons to Monte Carlos
- ✓ study of effect of non-perturbative corrections
- ✓ performance studies



Example of SoftDrop performance when used as a boosted W tagger

Take home points

The big news of the past fifteen years has been the development of robust taggers and groomers using properties of jet substructure, through

- declustering
- jet shapes
- direct analysis of images (machine learning)

These techniques have been commissioned by experimental collaborations and have proven their worth in 'Standard Model' analyses. They are now being implemented in BSM searches