

# Jets

Matteo Cacciari

LPTHE Paris and Université Paris Diderot

**Lecture 1 - Jet algorithms**

**Lecture 2 - Jet substructure**

[Includes material from  
Gavin Salam and Grégory Soyez]

## ▶ Jet algorithms

- ▶ How are jets made

## ▶ Jet substructure

- ▶ What's inside them

# 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**

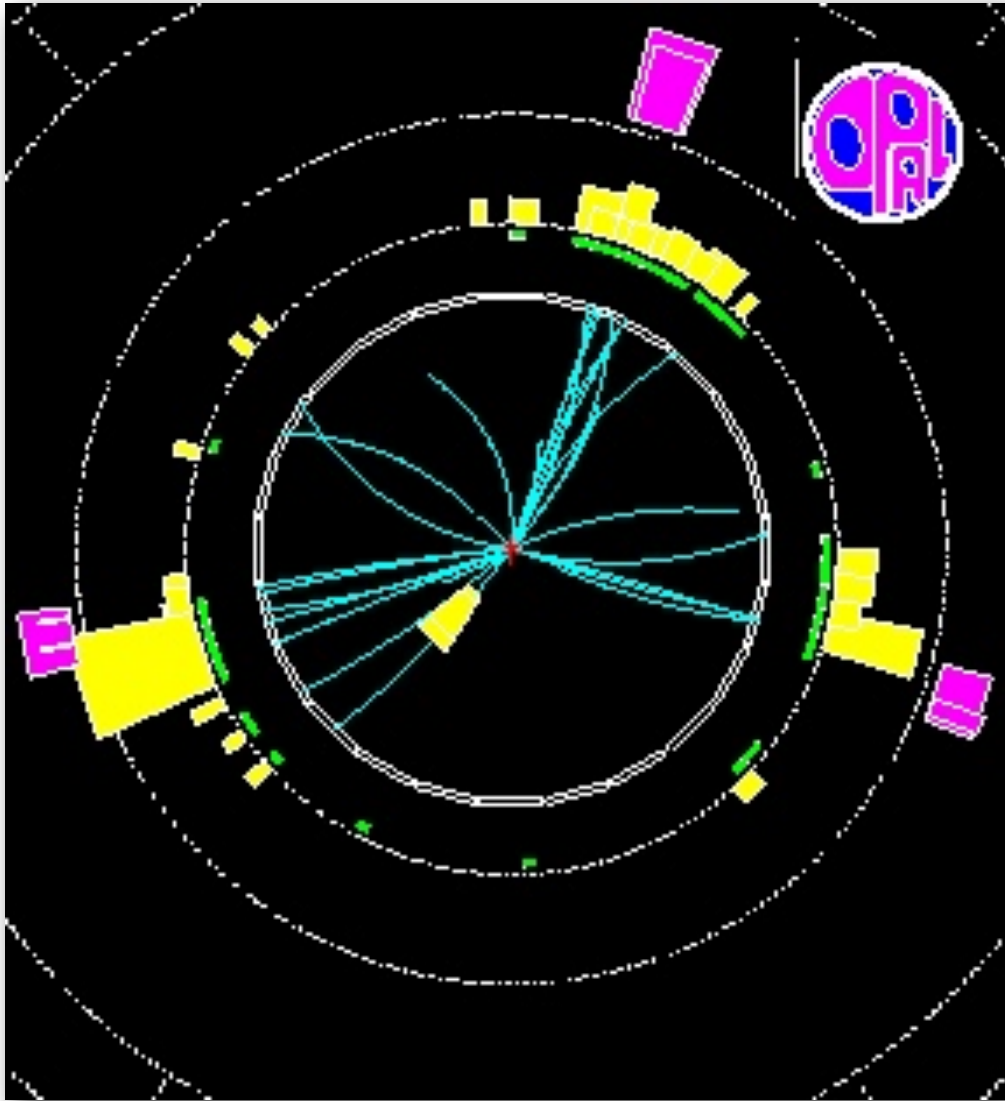




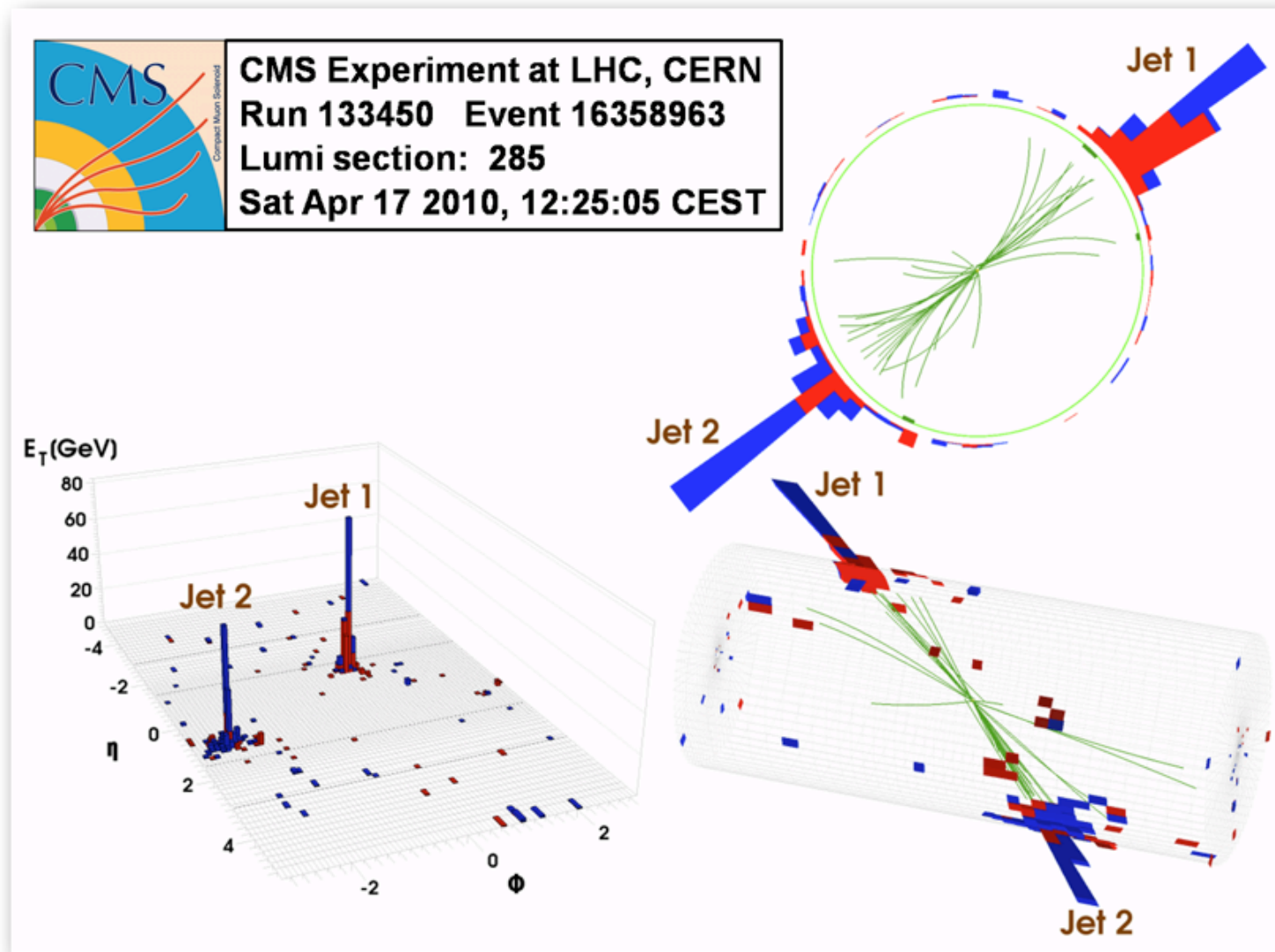
## From PETRA to LEP

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

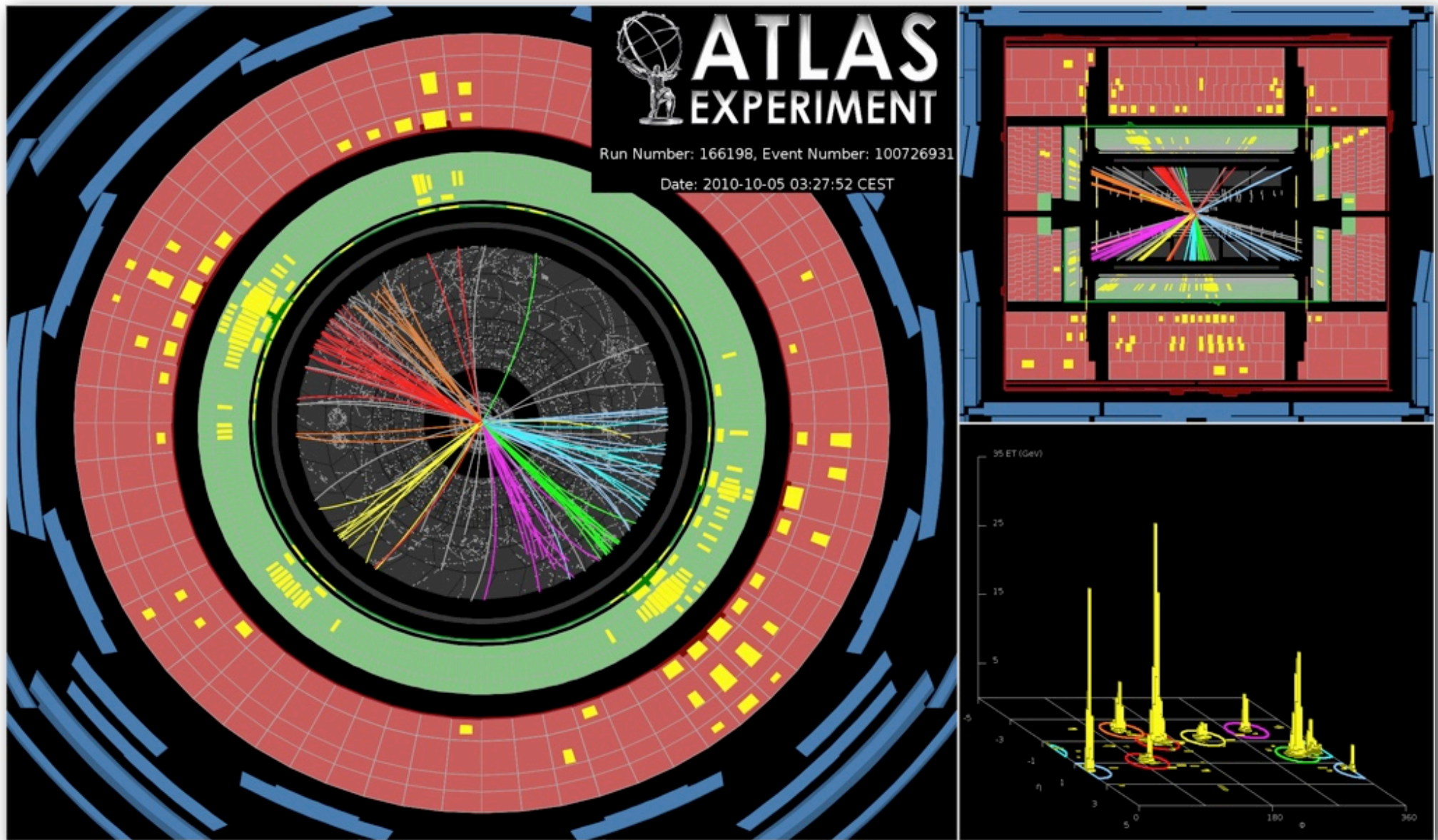
The classification of particles into jets is best done using a **clustering algorithm**



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



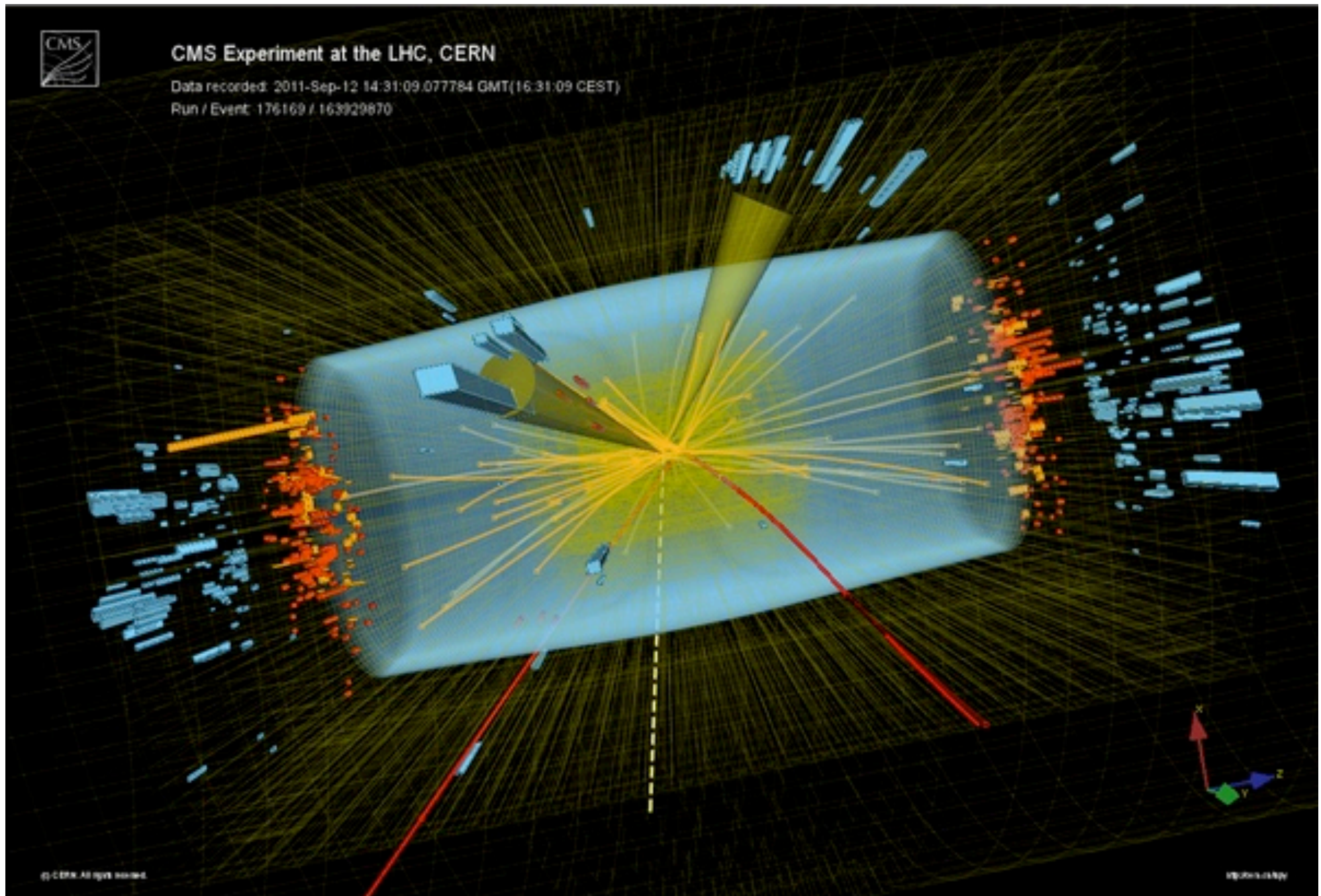
Dijet event from CMS



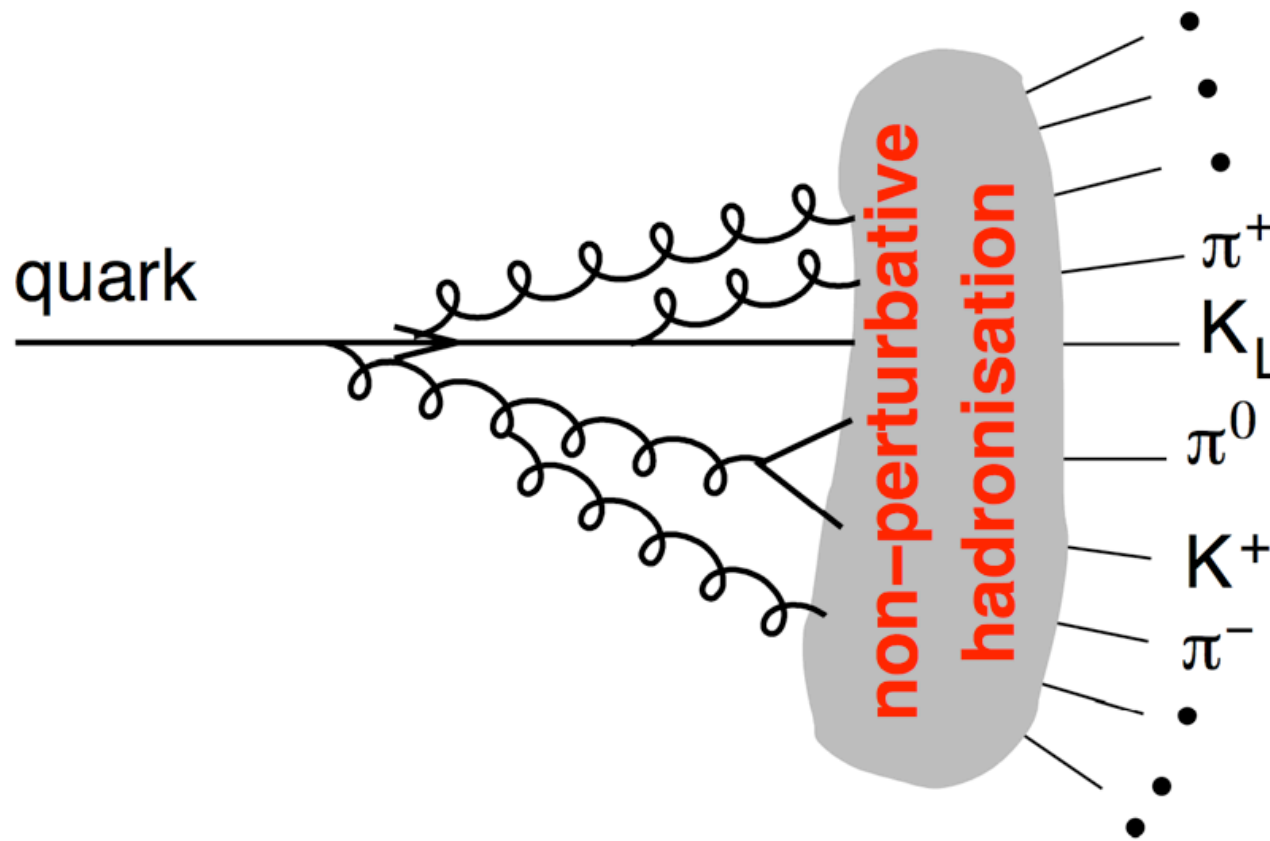
8(!) jets event from ATLAS



# Jets @ LHC



# Why do jets happen?



Gluon emission

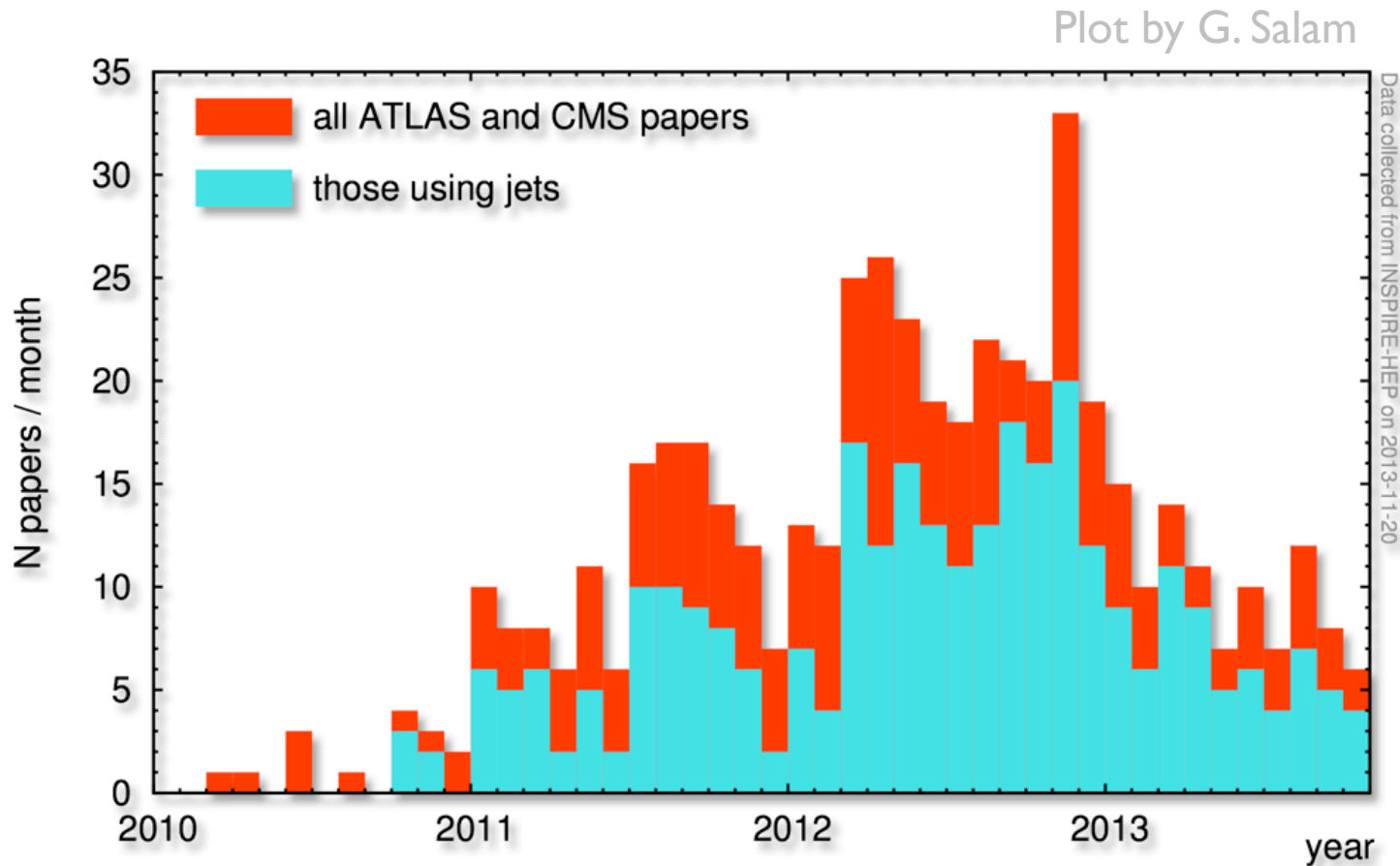
$$\int \alpha_s \frac{dE}{E} \frac{d\theta}{\theta} \gg 1$$

Non-perturbative physics

$$\alpha_s \sim 1$$

# The pervasiveness of jets

- ▶ ATLAS and CMS have each published **400+** papers since 2010
  - ▶ More than **half** of these papers make use of **jets**
  - ▶ **60%** of the **searches** papers makes use of **jets**

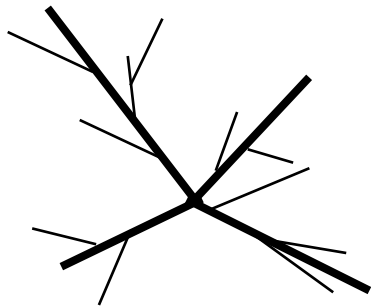


(Source: INSPIRE.  
Results may vary when  
employing different search  
keywords)

## Why are jets so important?

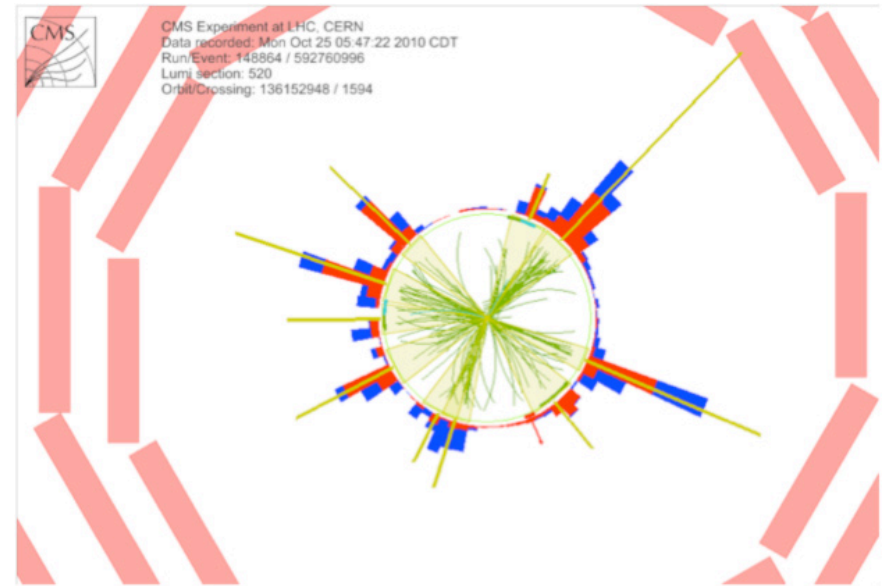
# Taming reality

Multileg + PS



QCD predictions

??



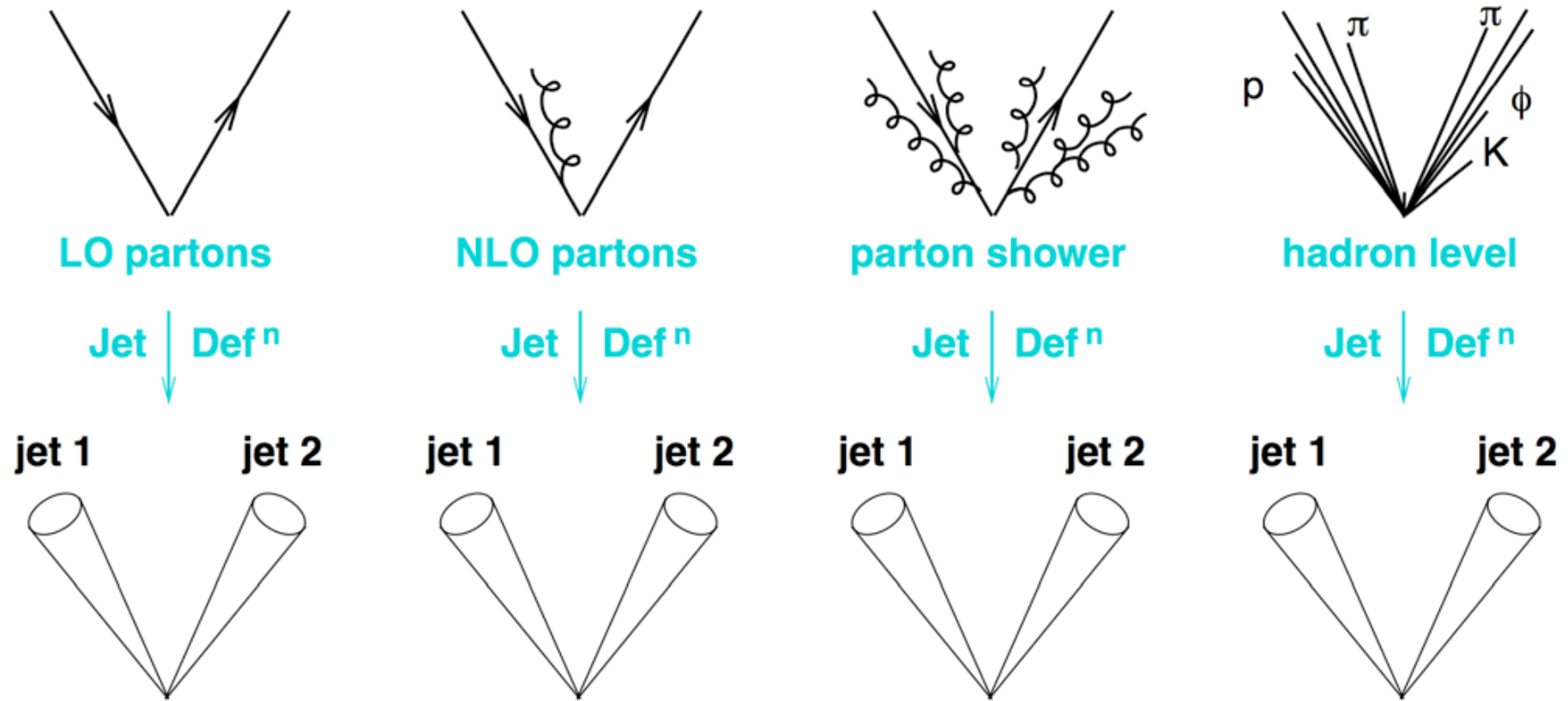
Real data

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

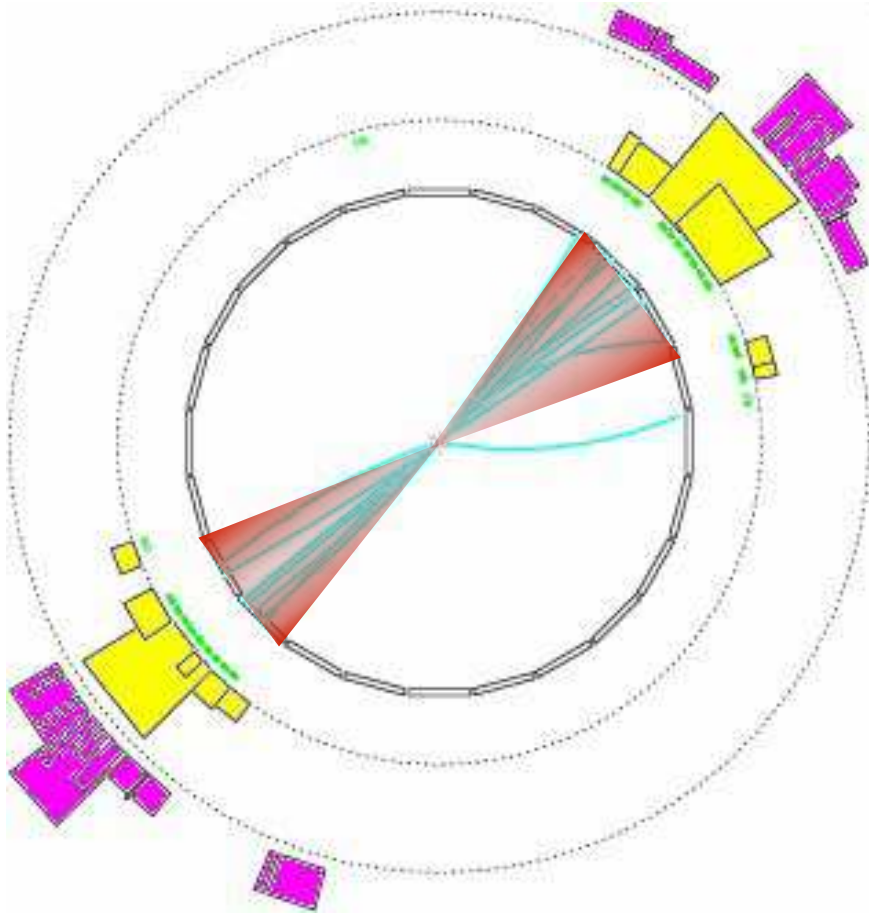


Projection to jets should be resilient to QCD effects

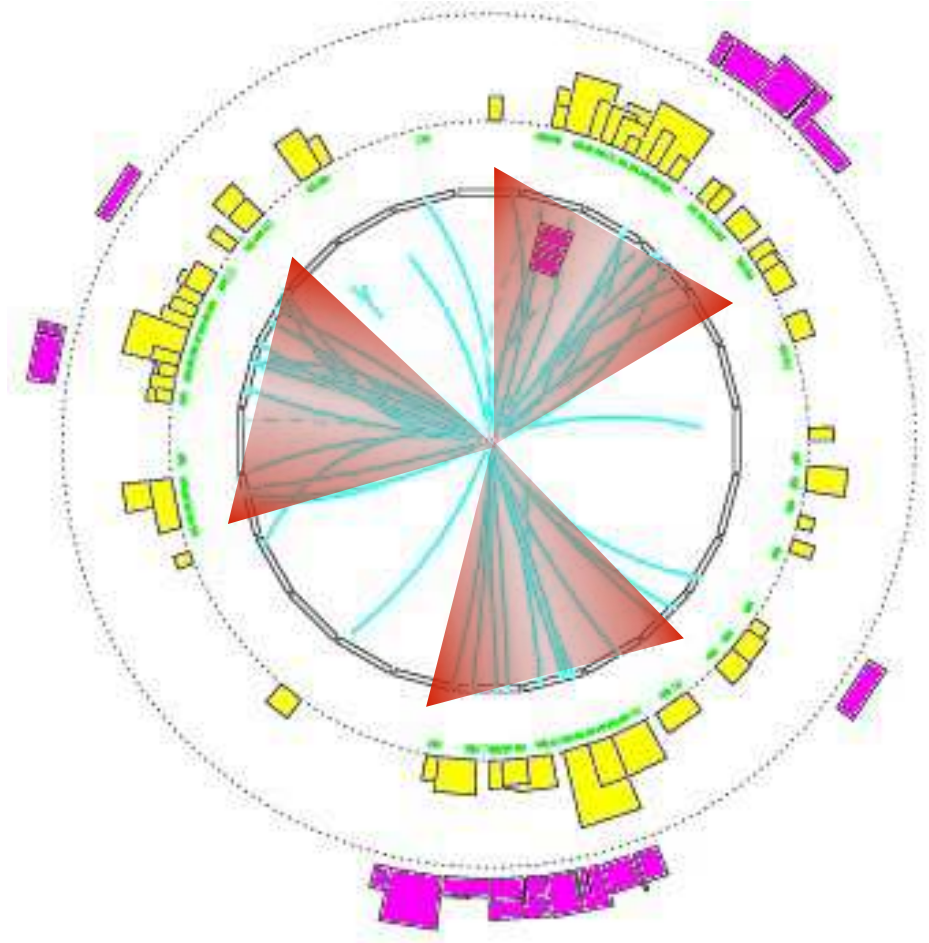
**NB: projections are NOT unique:  
a jet is NOT EQUIVALENT to a parton**



# Reconstructing jets is an ambiguous task

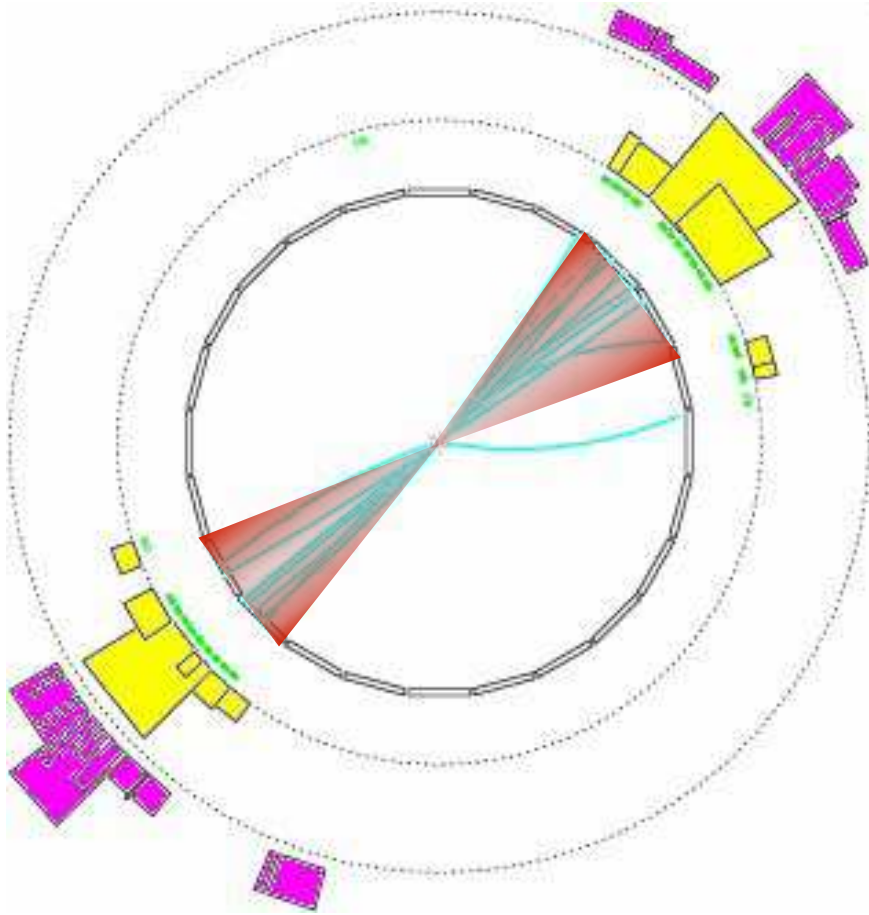


2 clear jets

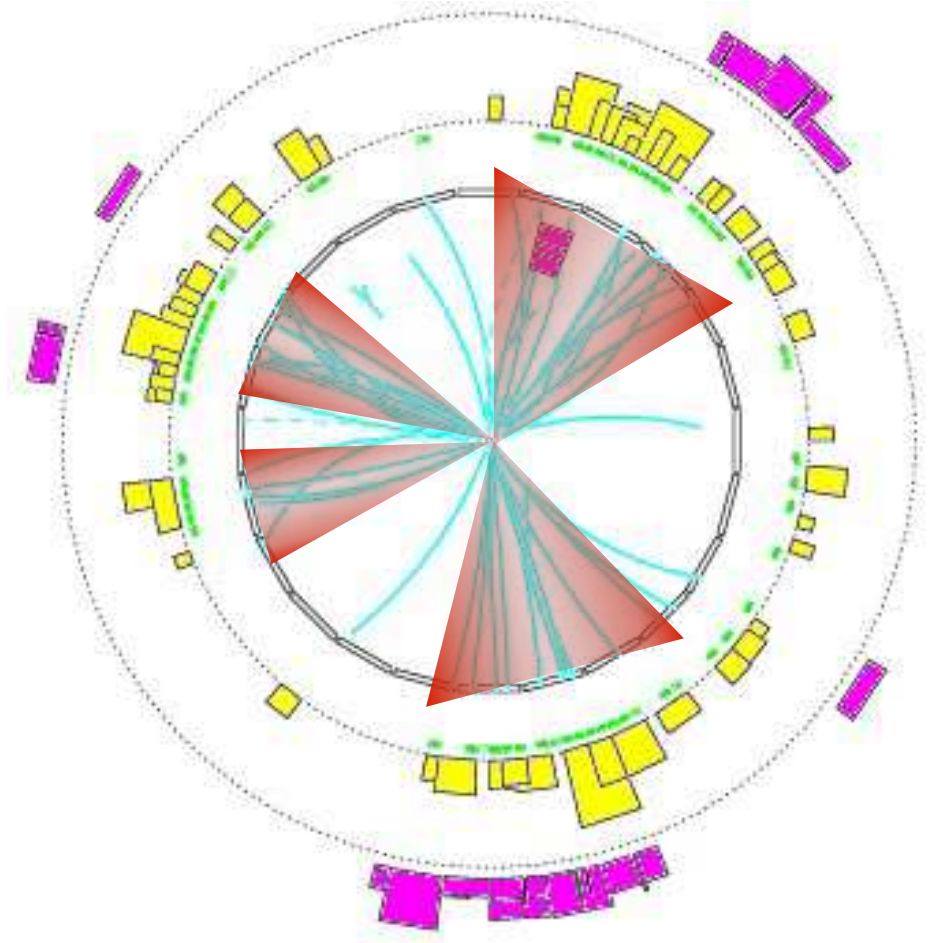


3 jets?

# Reconstructing jets is an ambiguous task



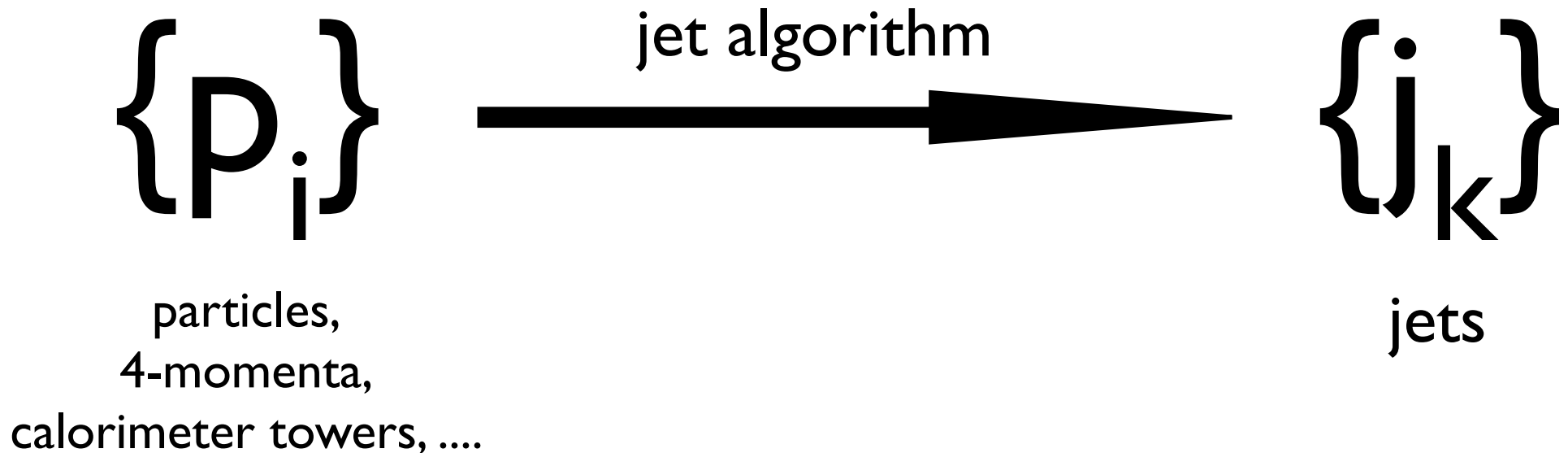
2 clear jets



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:



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

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**:

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

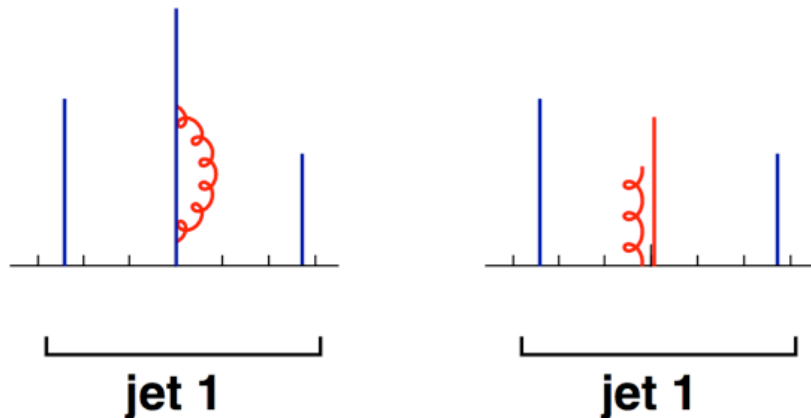
$$O(X; p_1, \dots, p_n \parallel p_{n+1}) \rightarrow O(X; p_1, \dots, p_n + p_{n+1})$$

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

## Collinear Safe

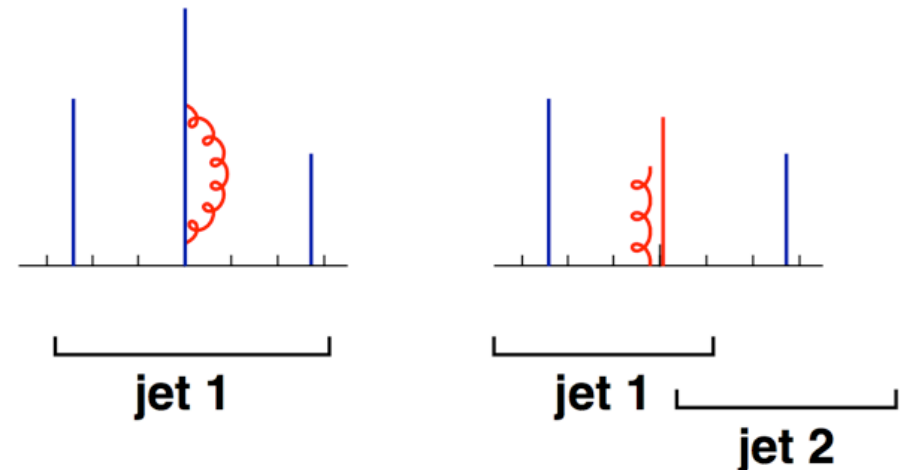


$$\alpha_s^n \times (-\infty)$$

$$\alpha_s^n \times (+\infty)$$

**Infinites cancel**

## Collinear Unsafe



$$\alpha_s^n \times (-\infty)$$

$$\alpha_s^n \times (+\infty)$$

**Infinites do not cancel**

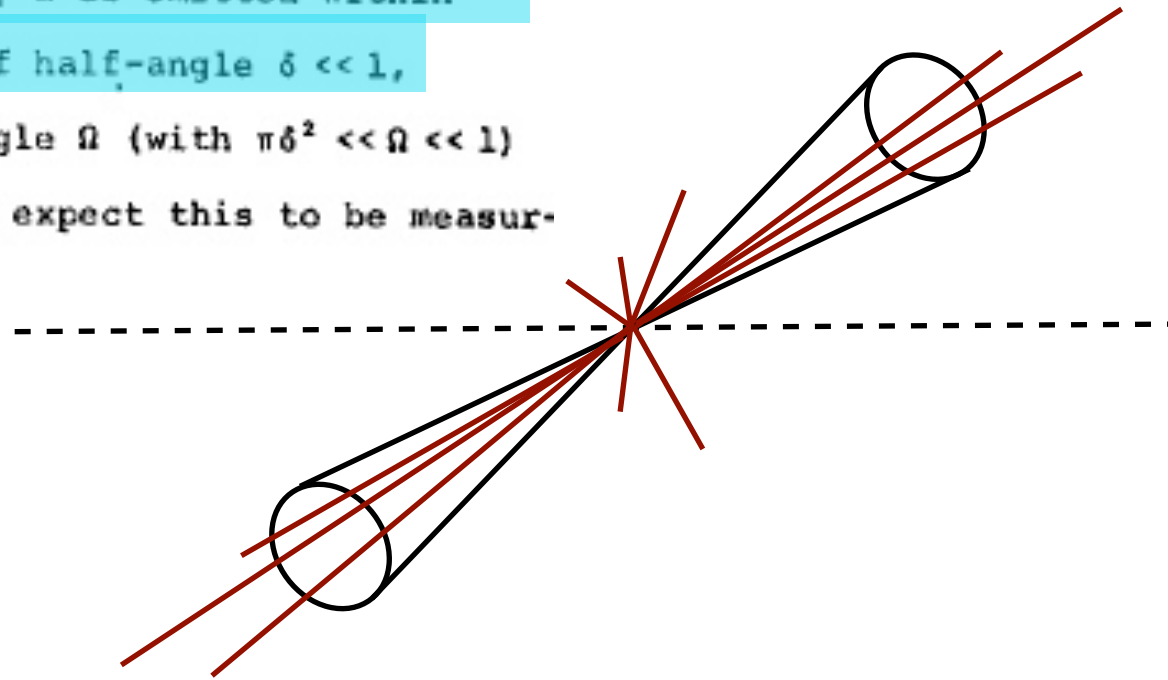
Perturbative calculations of jet observable will only be possible with collinear (and infrared) safe jet definitions



# Cone algorithms

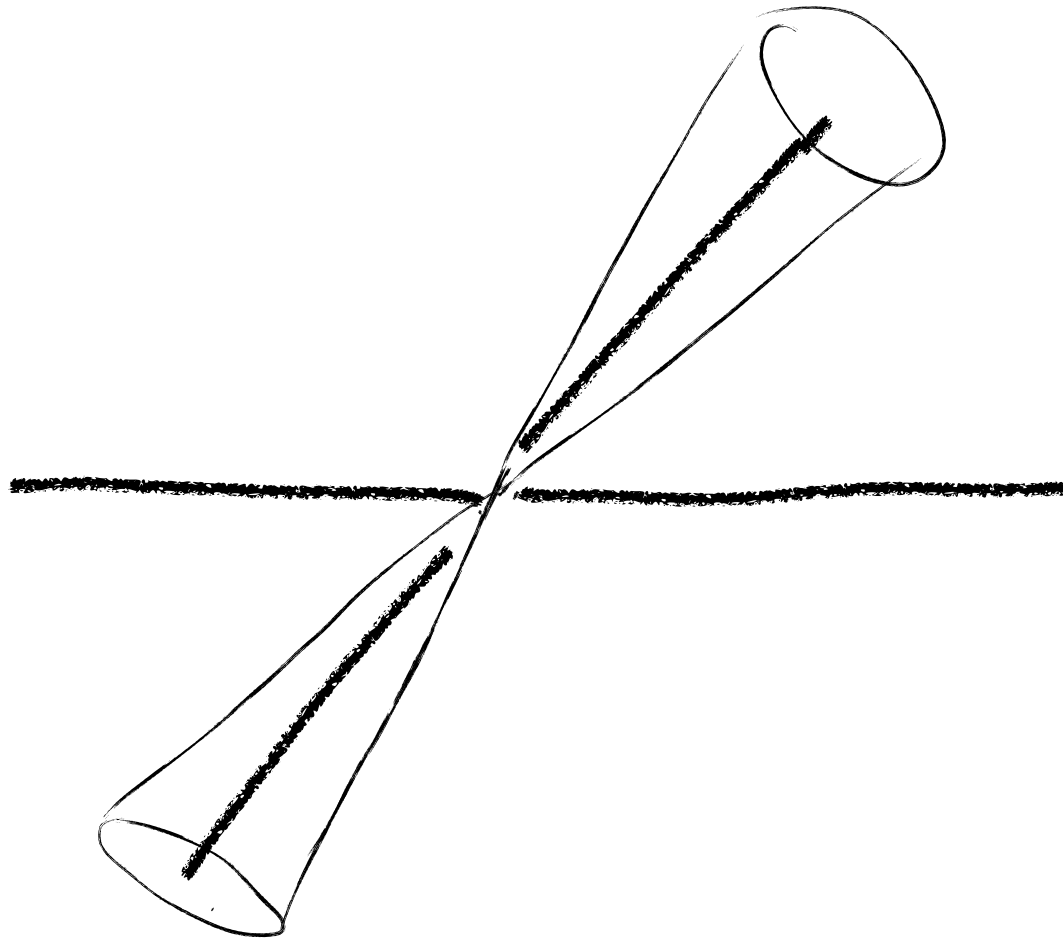
The first rigorous definition of cone jets in QCD is due to Stermann and Weinberg  
Phys. Rev. Lett. **39**, 1436 (1977)

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  $\Omega$  (with  $\pi\delta^2 \ll \Omega \ll 1$ )  
at an angle  $\theta$  to the  $e^+e^-$  beam line. We expect this to be measur-



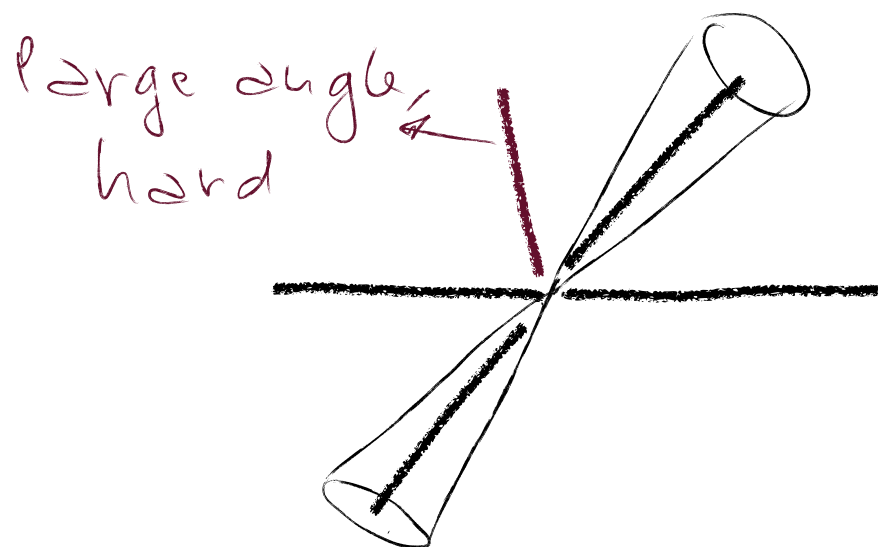
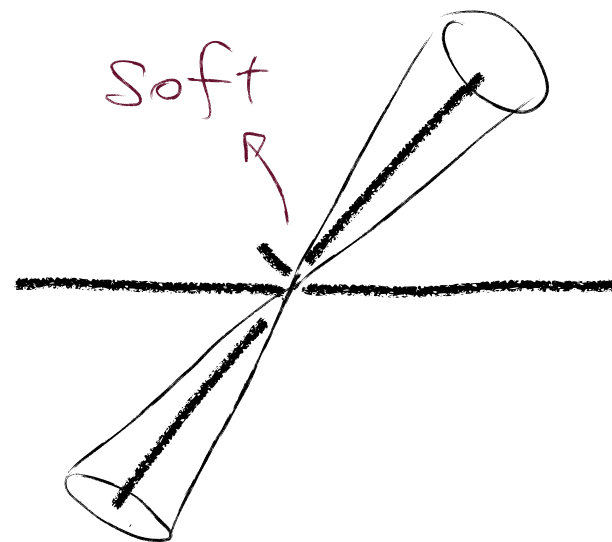
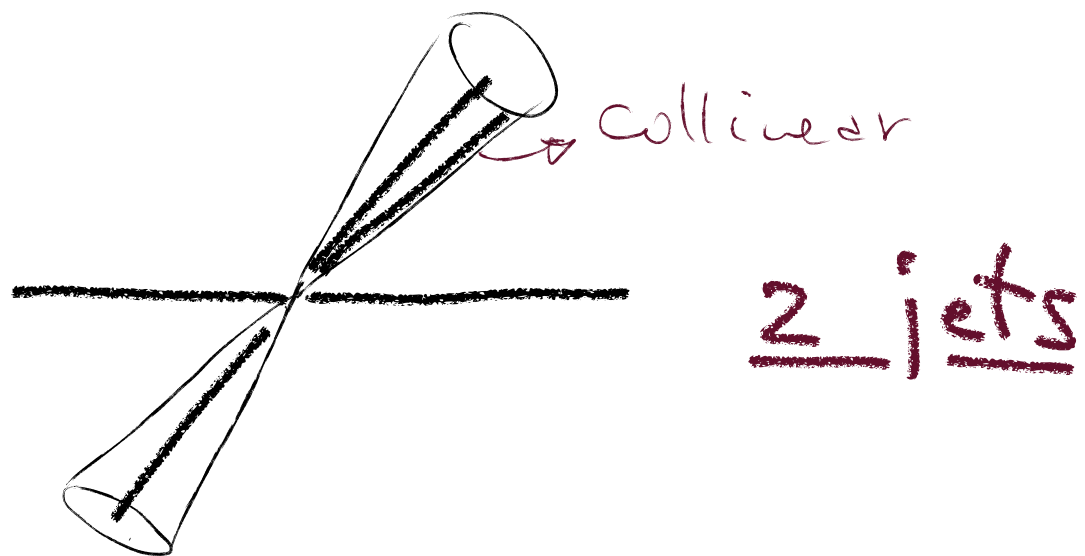
$$\sigma(E, \theta, \Omega, \epsilon, \delta) = (d\sigma/d\Omega)_0 \Omega \left[ 1 - (g_E^2/3\pi^2) \left\{ 3\ln \delta + 4\ln \delta \ln 2\epsilon + \frac{\pi^3}{3} - \frac{5}{2} \right\} \right]$$

2 particles = 2 jets  
(+ virtual corrections)





3 particles  $\geq$  ?



3 jets

The Stermann-Weinberg definition is “inclusive enough”  
for IRC safety

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

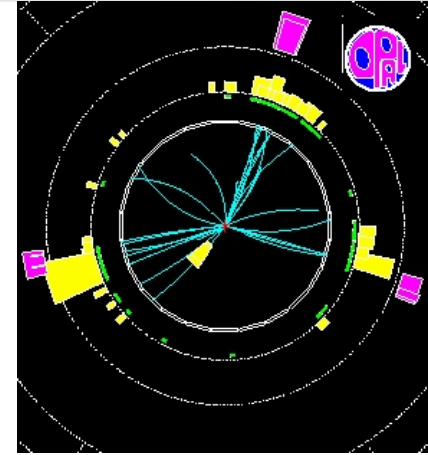
What happens in a more general case, where more than  
two jets are likely to exist?

**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  
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)

# 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  $\Leftrightarrow$  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

FERMILAB-Conf-90/249-E  
[E-741/CDF]

## Toward a Standardization of Jet Definitions \*

\* To be published in the proceedings of the 1990 Summer Study on High Energy Physics, *Research Directions for the Decade*, Snowmass, Colorado, June 25 - July 13, 1990.

Several important properties that should be met by a jet definition are [3]:

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 section at any order of perturbation theory;
5. Yields a cross section that is relatively insensitive to hadronization.

**Speed**

**Infrared and  
collinear safety**

[Addition of a soft particle or a collinear splitting should not change final hard jets]

# A little history

- ▶ Cone-type jets were introduced first in QCD in the 1970s (Sterman-Weinberg '77)
- ▶ In the 1980s cone-type jets were adapted for use in hadron colliders (SppS, Tevatron...) → iterative cone algorithms
- ▶ LEP was a golden era for jets: new algorithms and many relevant calculations during the 1990s
  - ▶ Introduction of the 'theory-friendly'  $k_t$  algorithm
    - ▶ sequential recombination type algorithm, IRC safe
    - ▶ it allows for all order resummation of jet rates
  - ▶ Several accurate calculations in perturbative QCD of jet properties: rates, jet mass, thrust, ....

# Finding stable cones

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

Unfortunately, this takes  $N^{2^N}$  operations:  
the time taken is the age of the universe for only 100 objects

An approximate way out is to use **seeds** (e.g. à la k-means)  
However, the final result can depend on the choice of the seeds and,  
such jet algorithms usually turn out to be **IRC unsafe**

# Finding cones

Different procedures for placing the cones lead to **different cone algorithms**

NB: their properties and behaviour can **vastly differ**:  
there isn't '**a**' cone algorithm, but rather many of them

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** ( $\text{IC}_{\text{mp}}$ -SM) (CDF MidPoint, D0 Run II)
- \*  $\text{IC}_{\text{mp}}$  with **split-drop** ( $\text{IC}_{\text{mp}}$ -SD) (PxCone)
- \* **Seedless** cone with **split-merge** (SC-SM) (SISCone)

All, except SISCone, are approximate

All, except SISCone, are infrared or collinear 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:

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

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

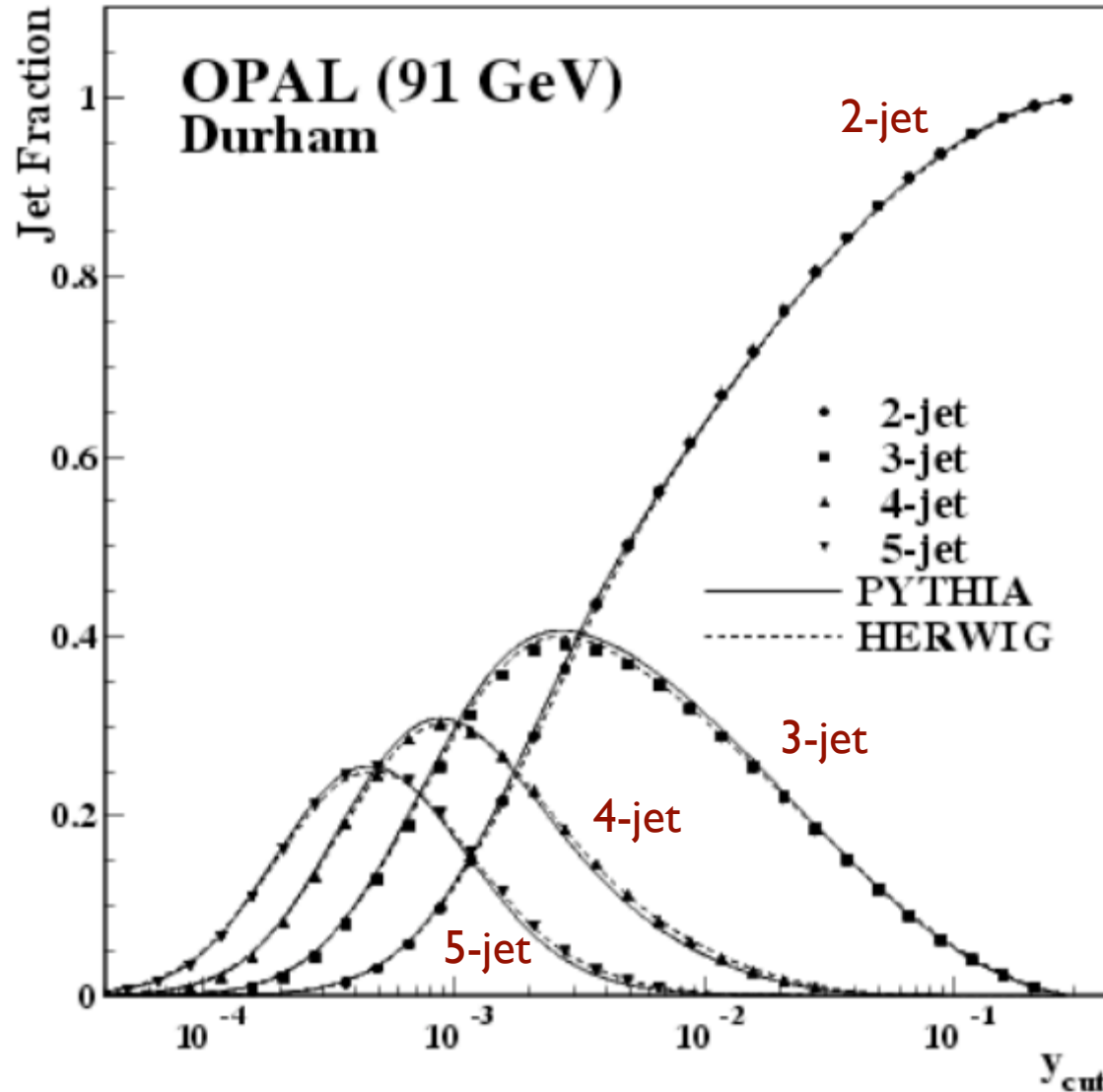
Distance:

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

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  $y_{\text{cut}}$ , recombine  $i$  and  $j$  into a single new particle ('pseudojet'), and repeat
- ▶ If no  $y_{\min} < y_{\text{cut}}$  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  $y_{\text{cut}}$ )

Resummed calculations for distributions of  $y_{\text{cut}}$  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:

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

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

# The common wisdom circa 2005

- ▶ Cone algorithms are IRC unsafe
  - ➔ because, to make them reasonably fast, they were usually implemented via approximate methods using seeds
- ▶ Sequential recombination algorithms (i.e.  $k_t$ ) are slow and too susceptible to background contamination
  - ➔ because they scale like  $N^3$
  - ➔ because they tend to collect soft particles up to large distances from centre
  - ➔ because they were often run with  $R=1$  and compared to cones with  $R=0.5$ !

The solution to the speed problem came from considering the clustering problem from a geometrical rather than from a combinatorial point of view

- Sequential recombination algorithms could be implemented with  **$O(N^2)$**  or even  **$O(N \ln N)$**  complexity rather than  $O(N^3)$

[MC, Salam, 2006]

- Cone algorithms could be implemented exactly (and therefore made IRC safe) with  **$O(N^2 \ln N)$**  rather than  $O(N^{2N})$  complexity

[Salam, Soyez 2007]

# $k_t$ algorithm in hadron collisions

(Inclusive and longitudinally invariant version)

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

- ▶ Calculate the distances between the particles:  $d_{ij}$
- ▶ Calculate the beam distances:  $d_{iB}$
- ▶ 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

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

**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- $k_t$  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

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

## $p > 0$

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

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

## $p = 0$

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

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

## $p < 0$

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

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

# IRC safe algorithms

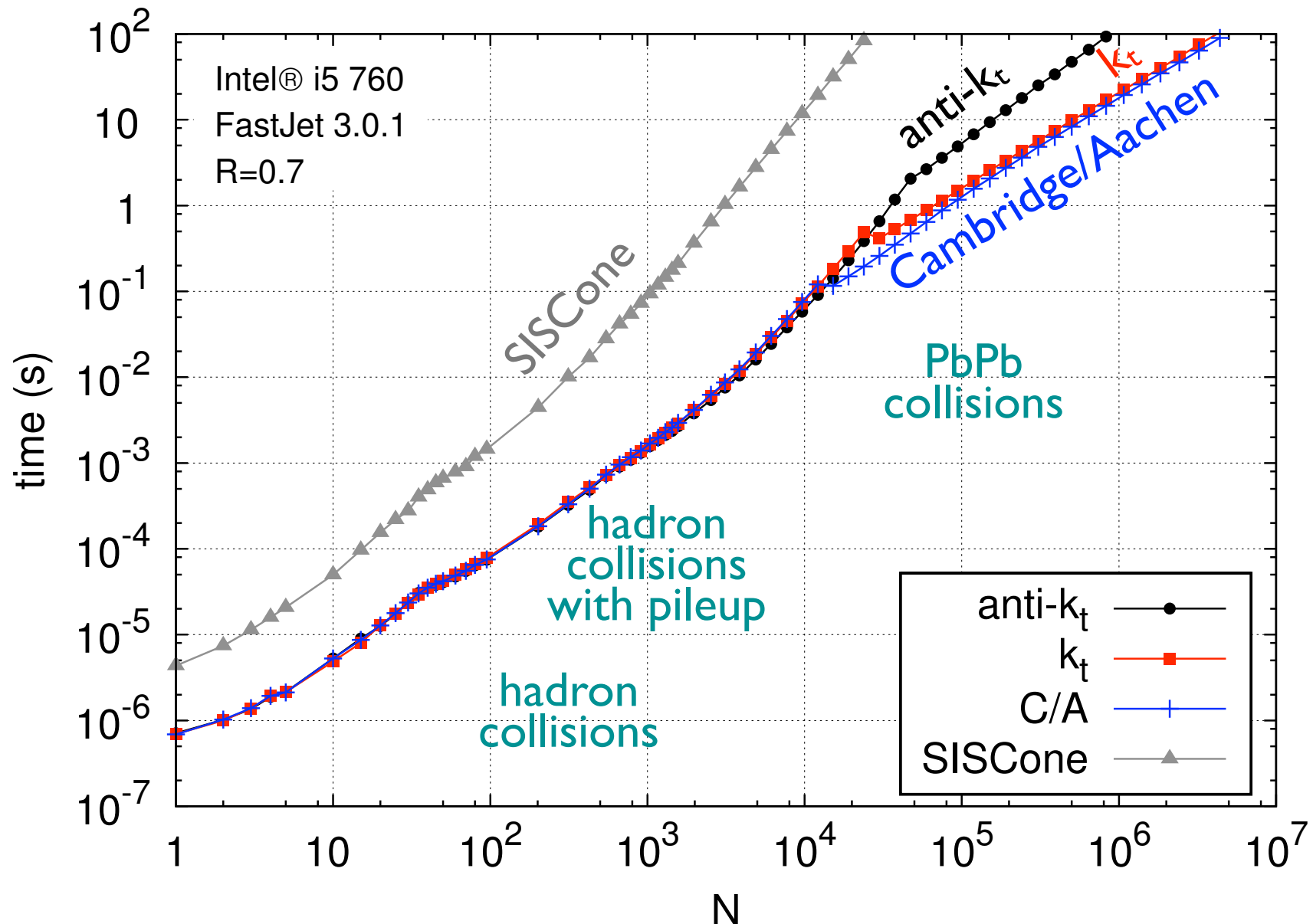
$k_t$	<p>SR</p> $d_{ij} = \min(p_{ti}^2, p_{tj}^2) \Delta R_{ij}^2 / R^2$ <p>hierarchical in rel <math>p_t</math></p>	<p>Catani et al '91 Ellis, Soper '93</p>	$N \ln N$
Cambridge/ Aachen	<p>SR</p> $d_{ij} = \Delta R_{ij}^2 / R^2$ <p>hierarchical in angle</p>	<p>Dokshitzer et al '97 Wengler, Wobish '98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

'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

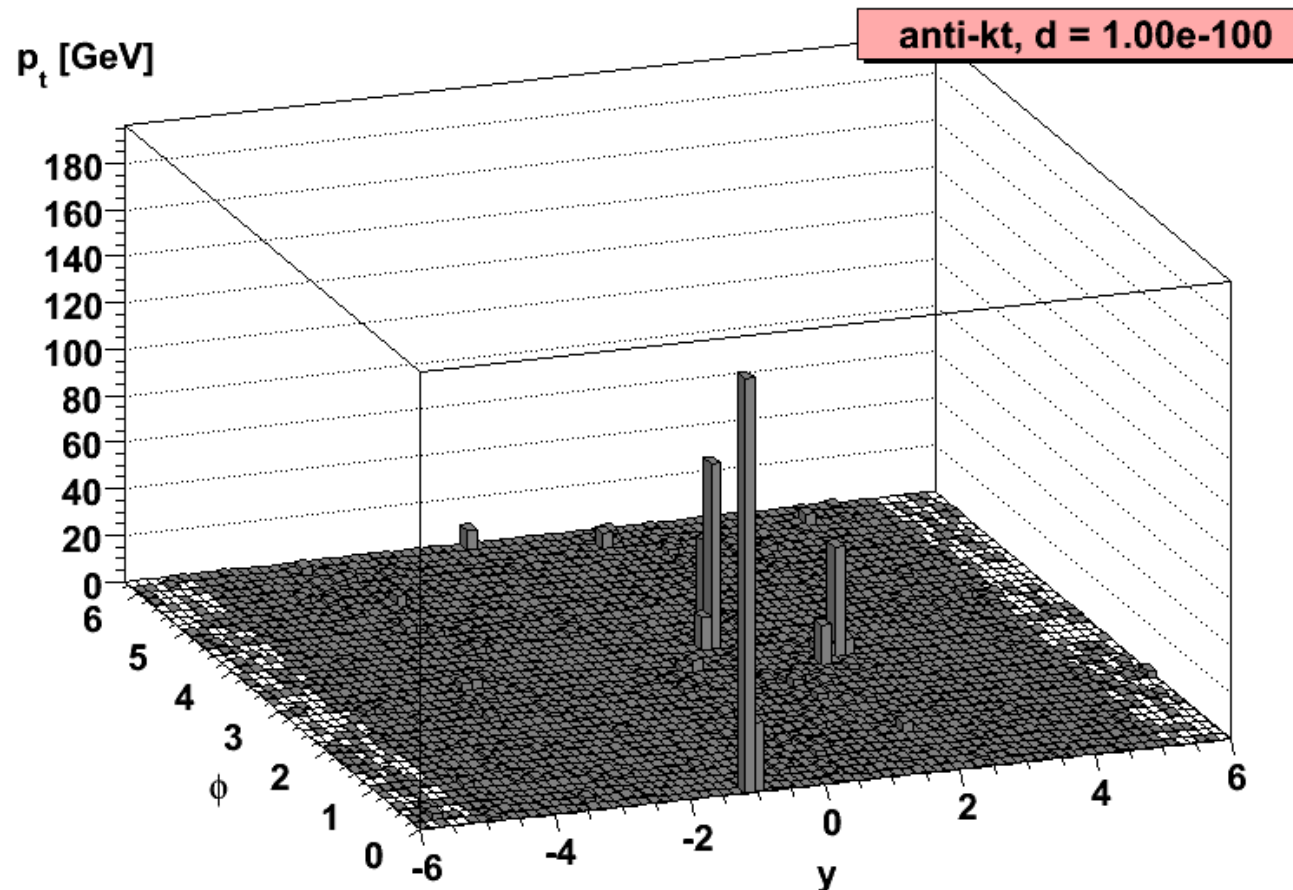


Clustering grows  
around hard cores

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

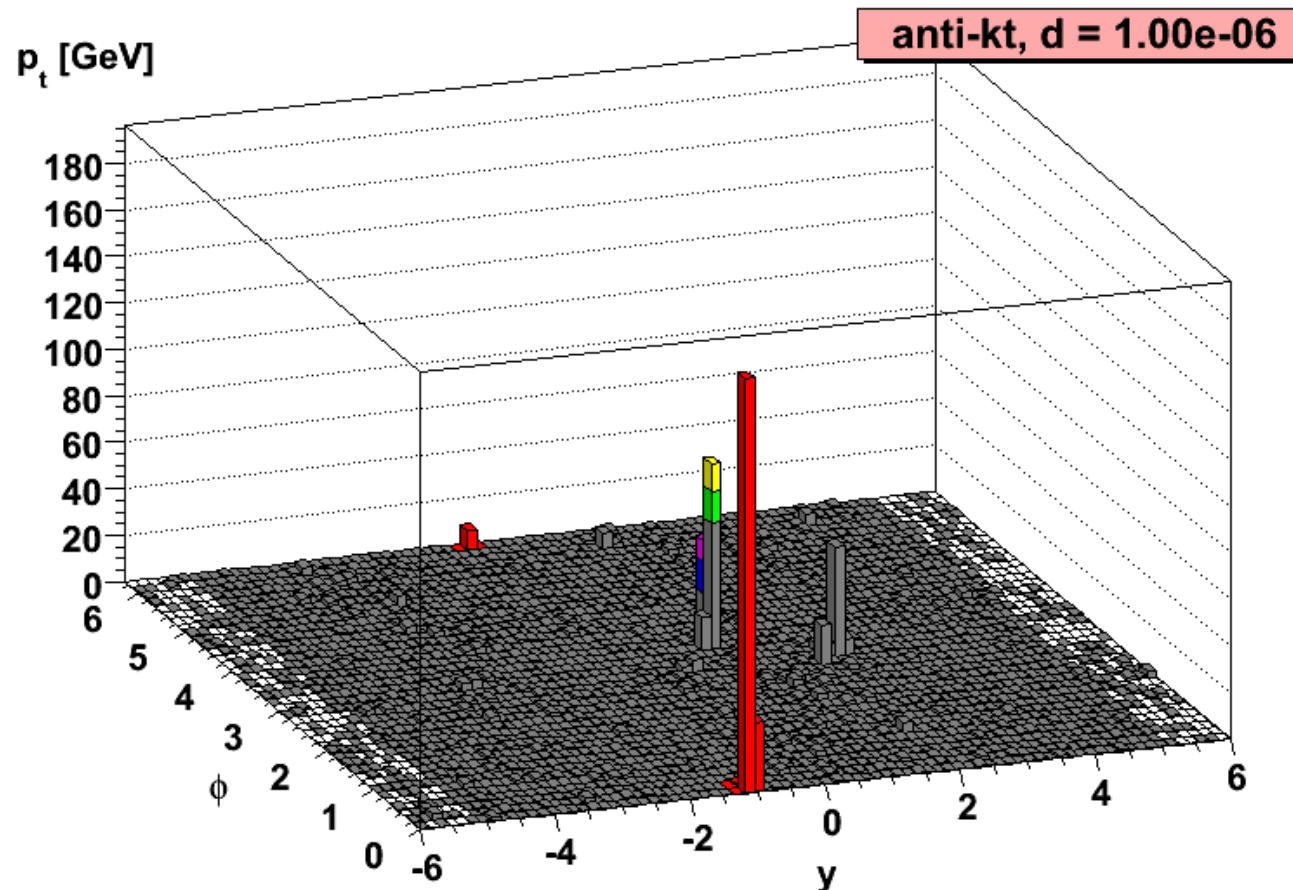
Clustering grows  
around hard cores

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



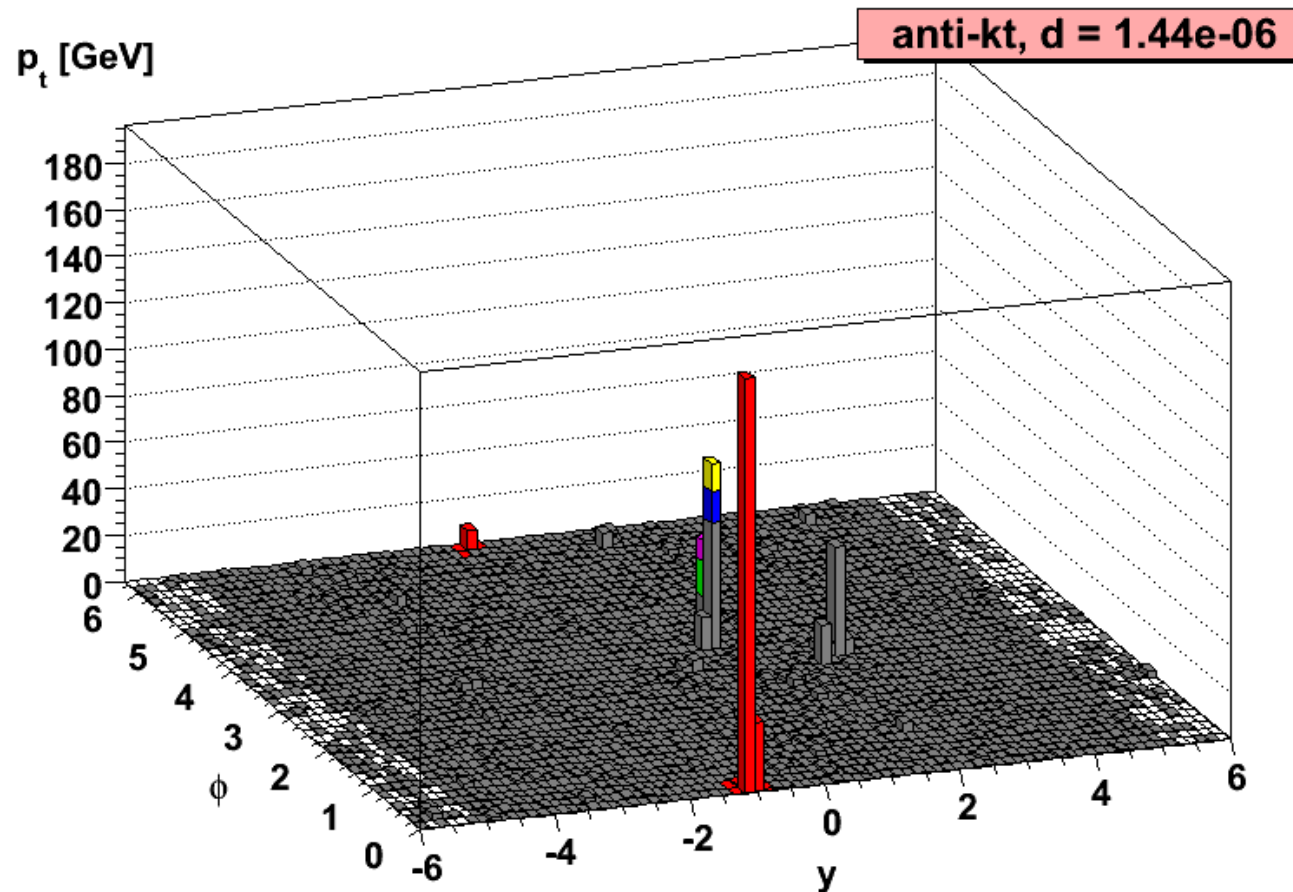
Clustering grows  
around hard cores

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



Clustering grows  
around hard cores

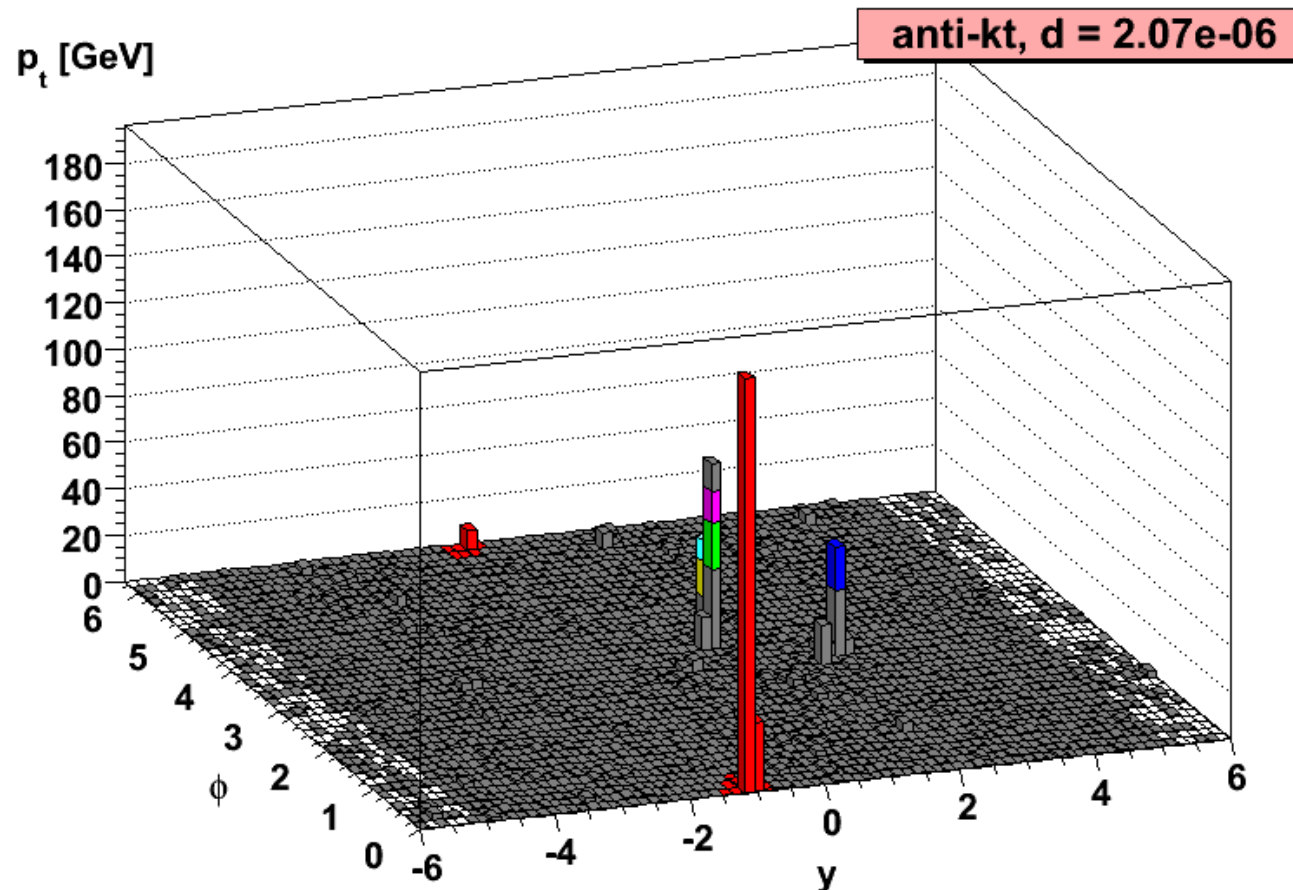
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$





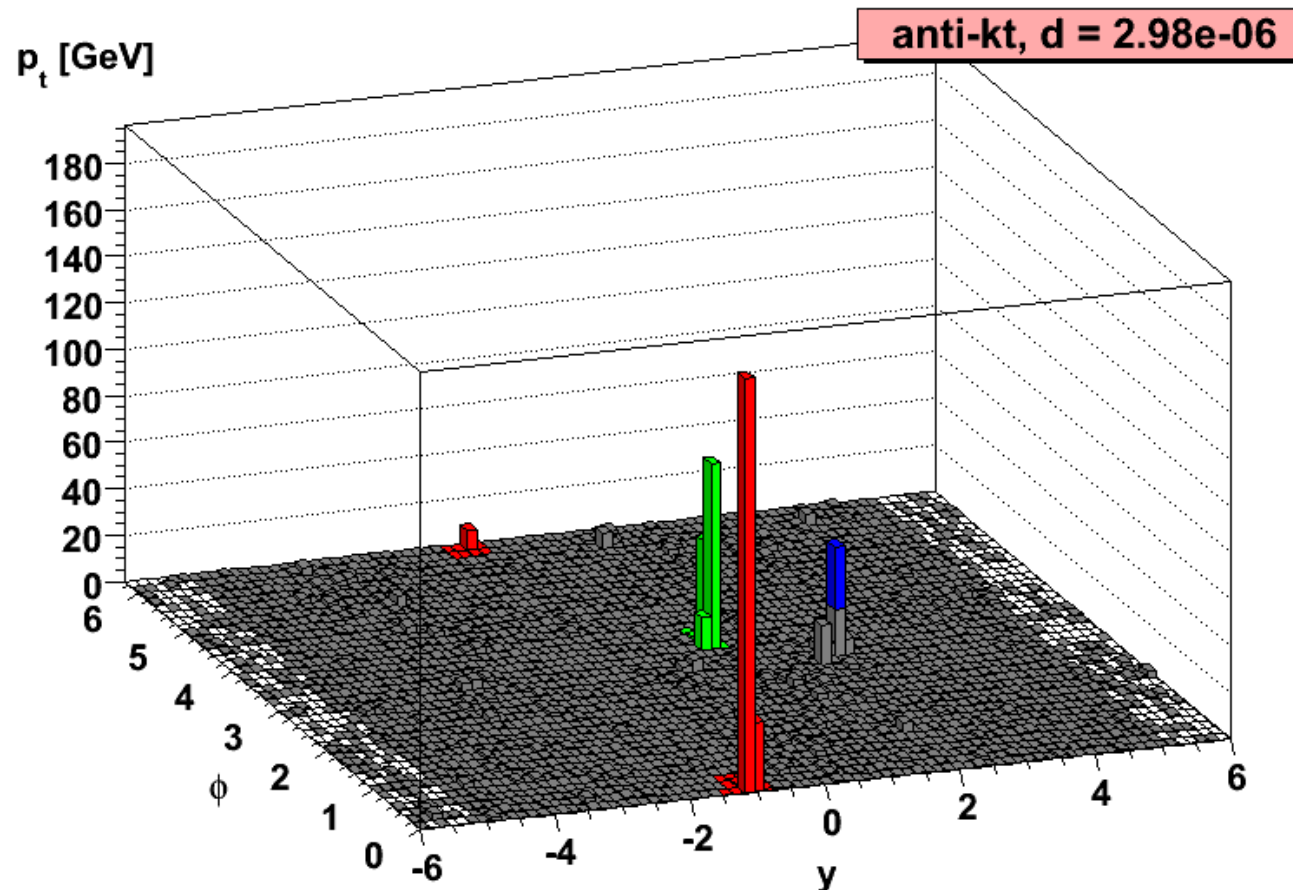
Clustering grows  
around hard cores

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



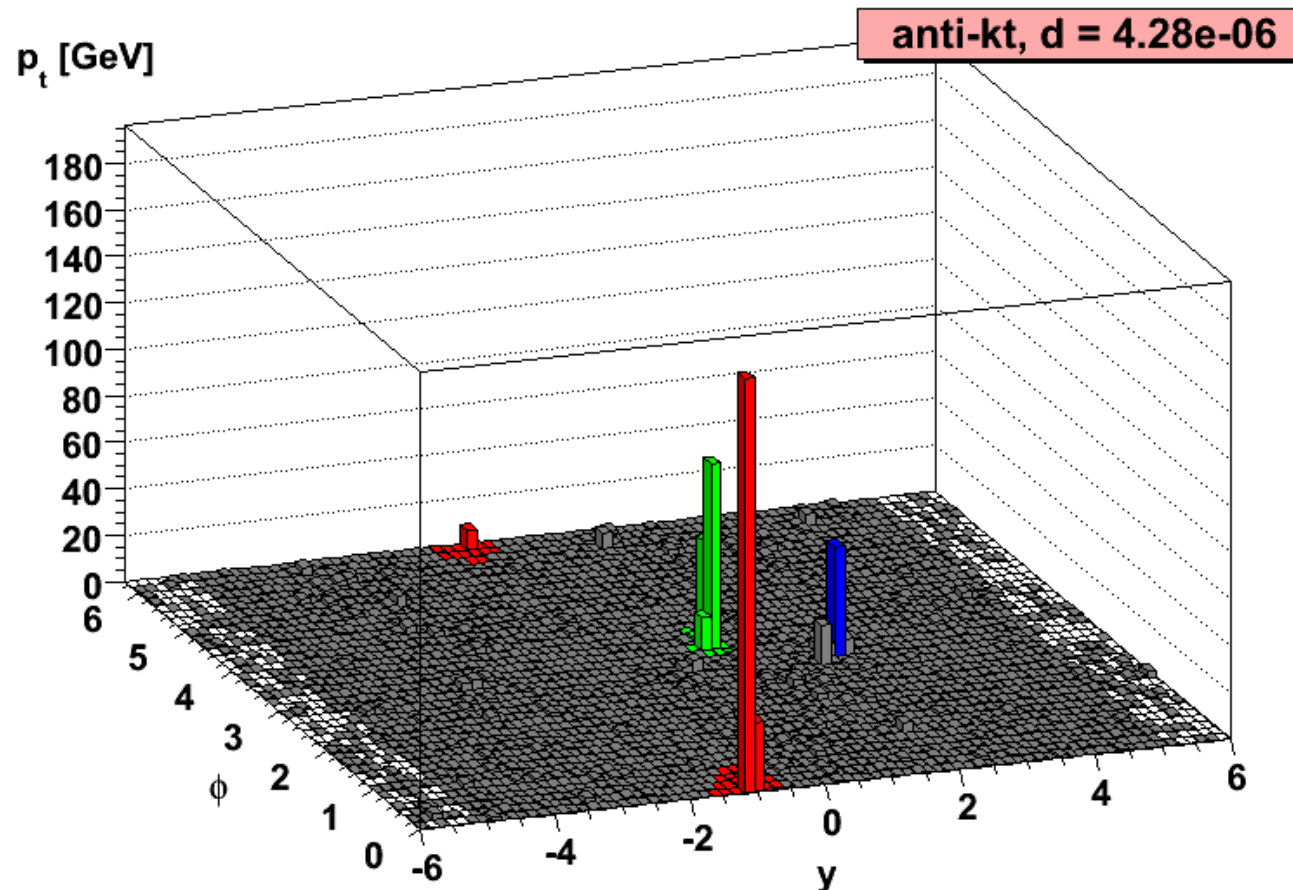
Clustering grows  
around hard cores

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



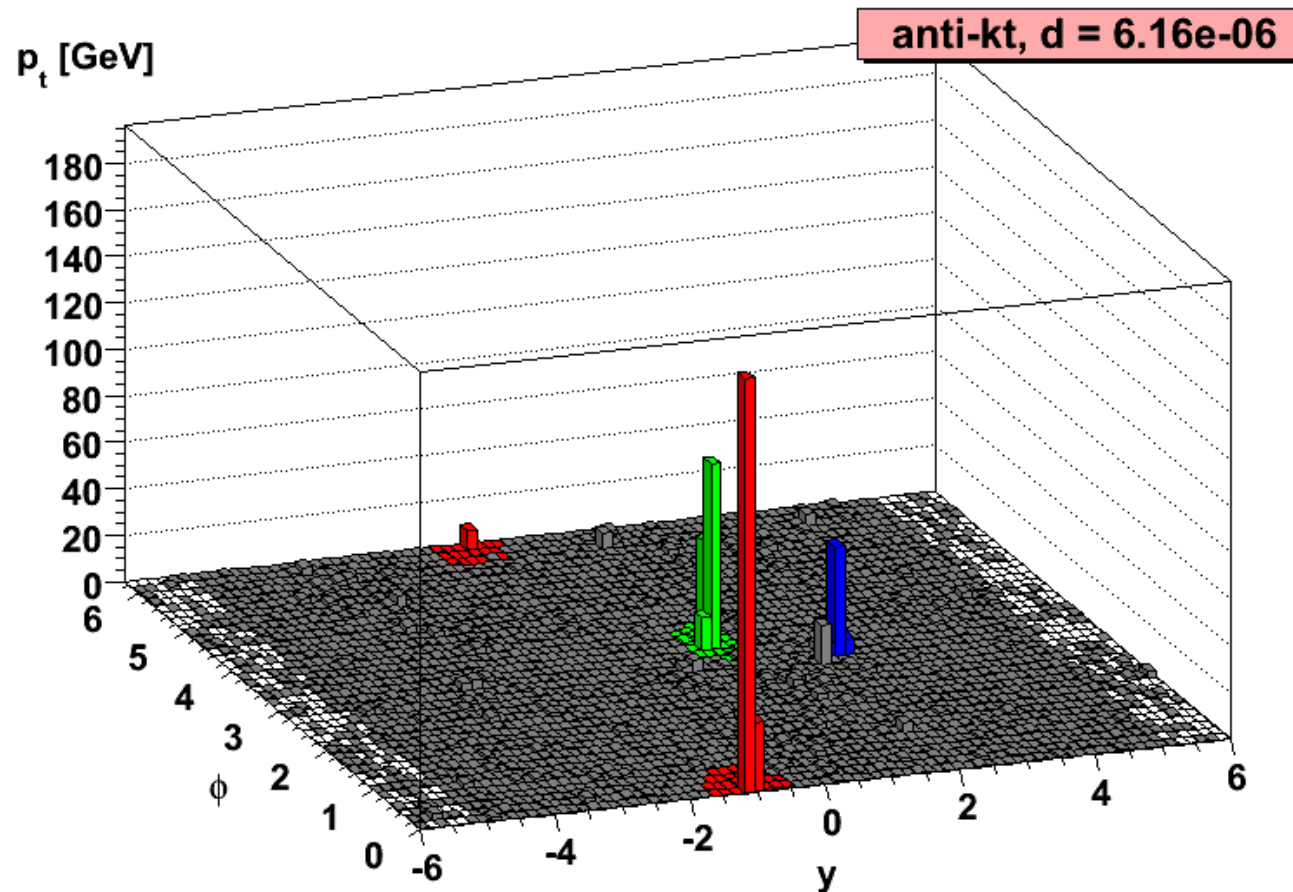
Clustering grows  
around hard cores

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



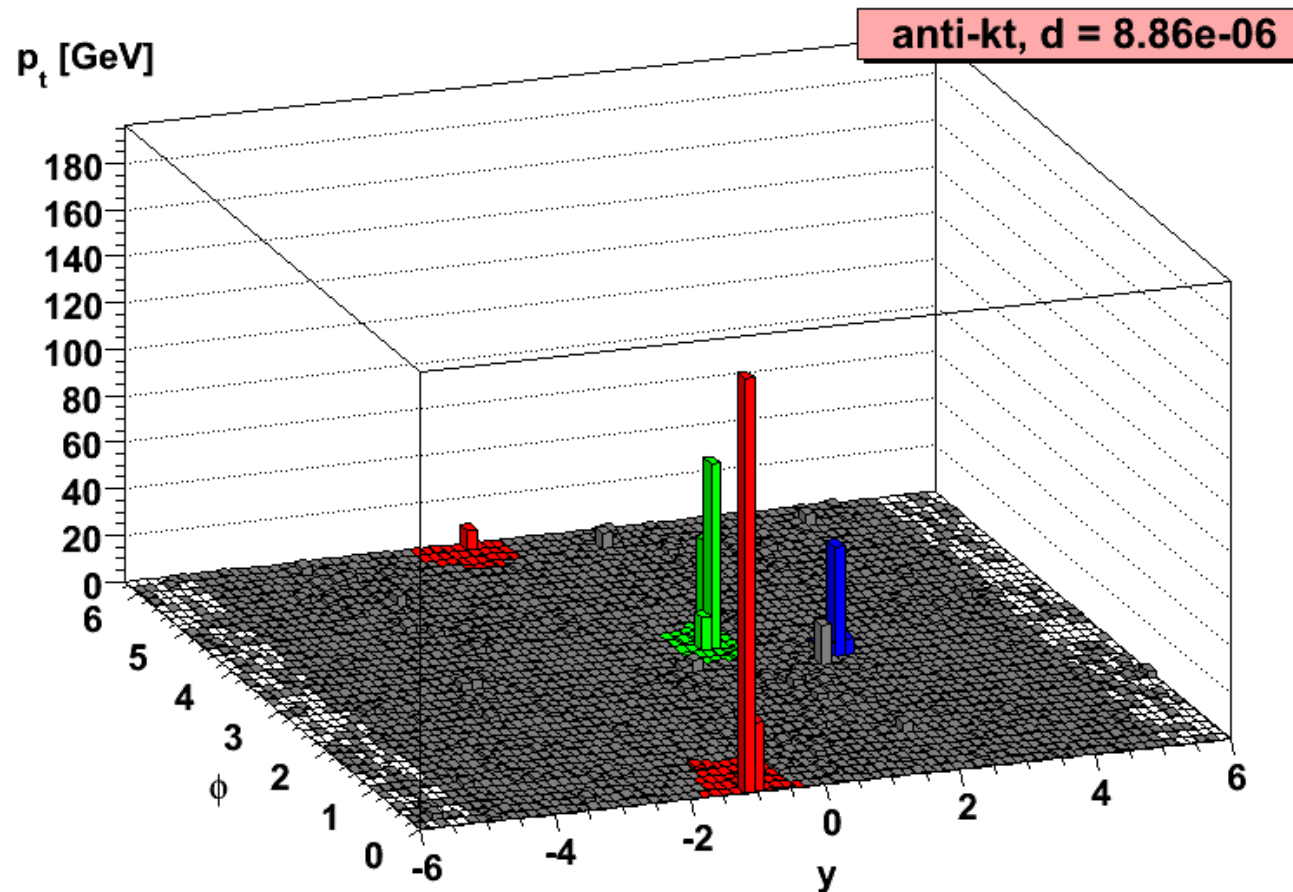
Clustering grows  
around hard cores

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



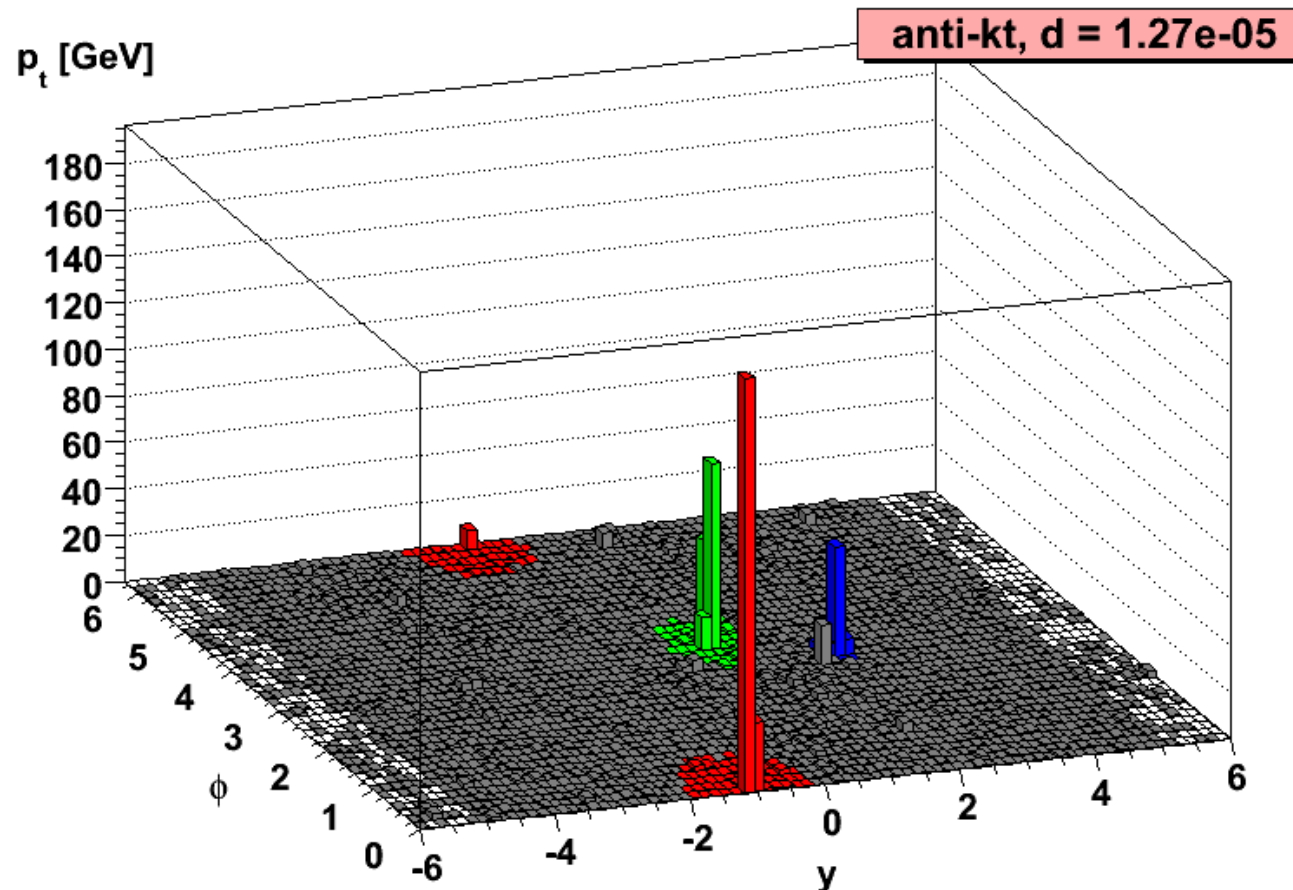
Clustering grows  
around hard cores

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



Clustering grows  
around hard cores

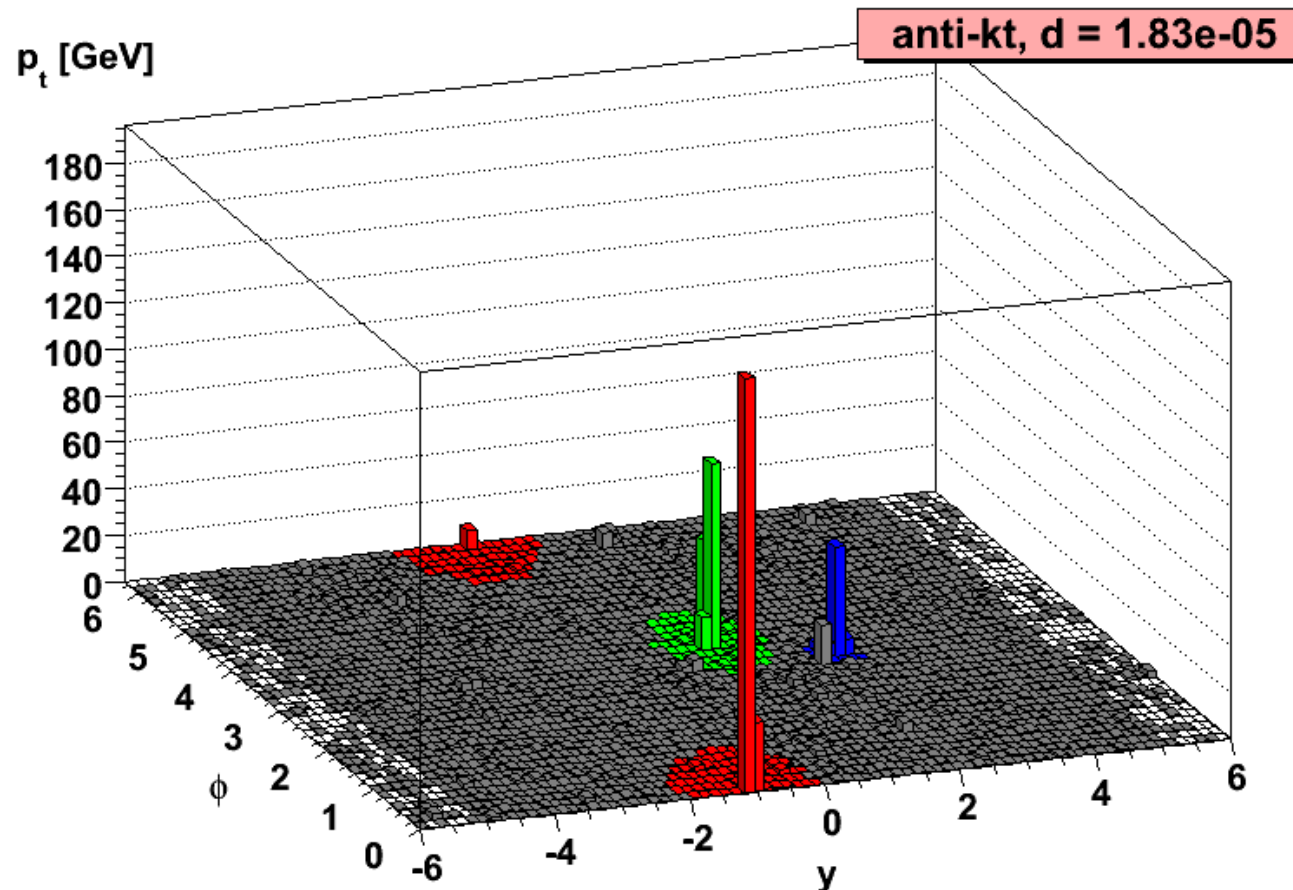
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$





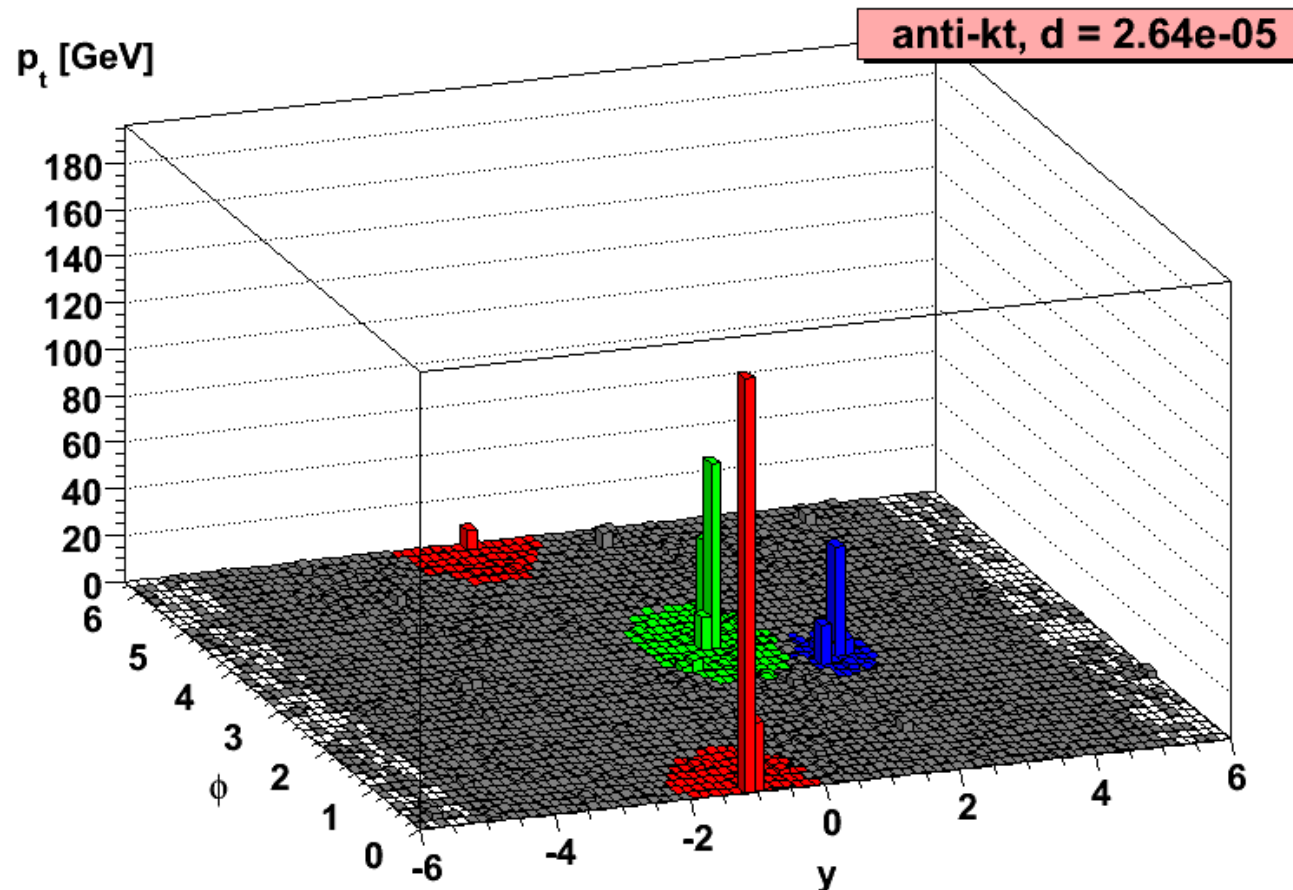
Clustering grows  
around hard cores

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



Clustering grows  
around hard cores

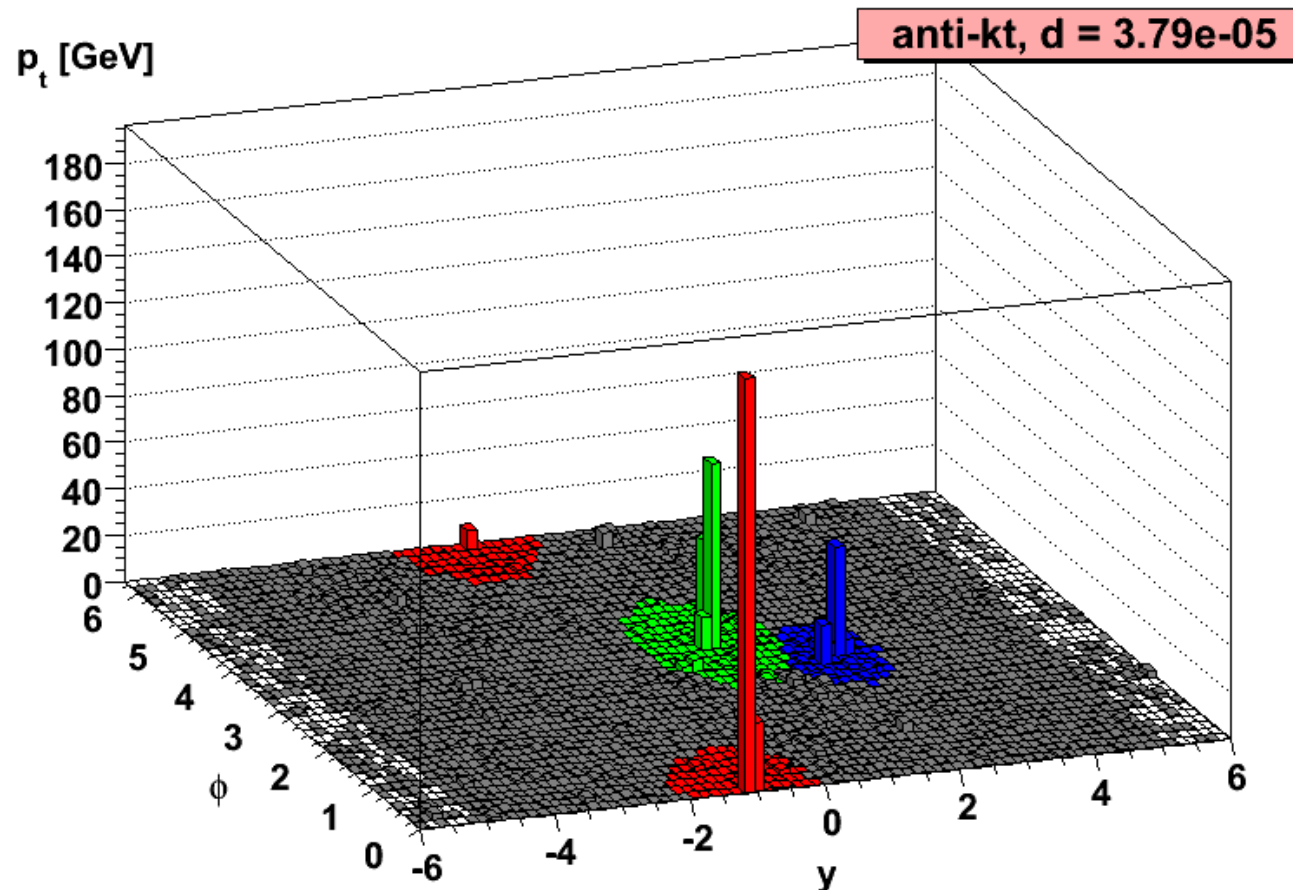
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$





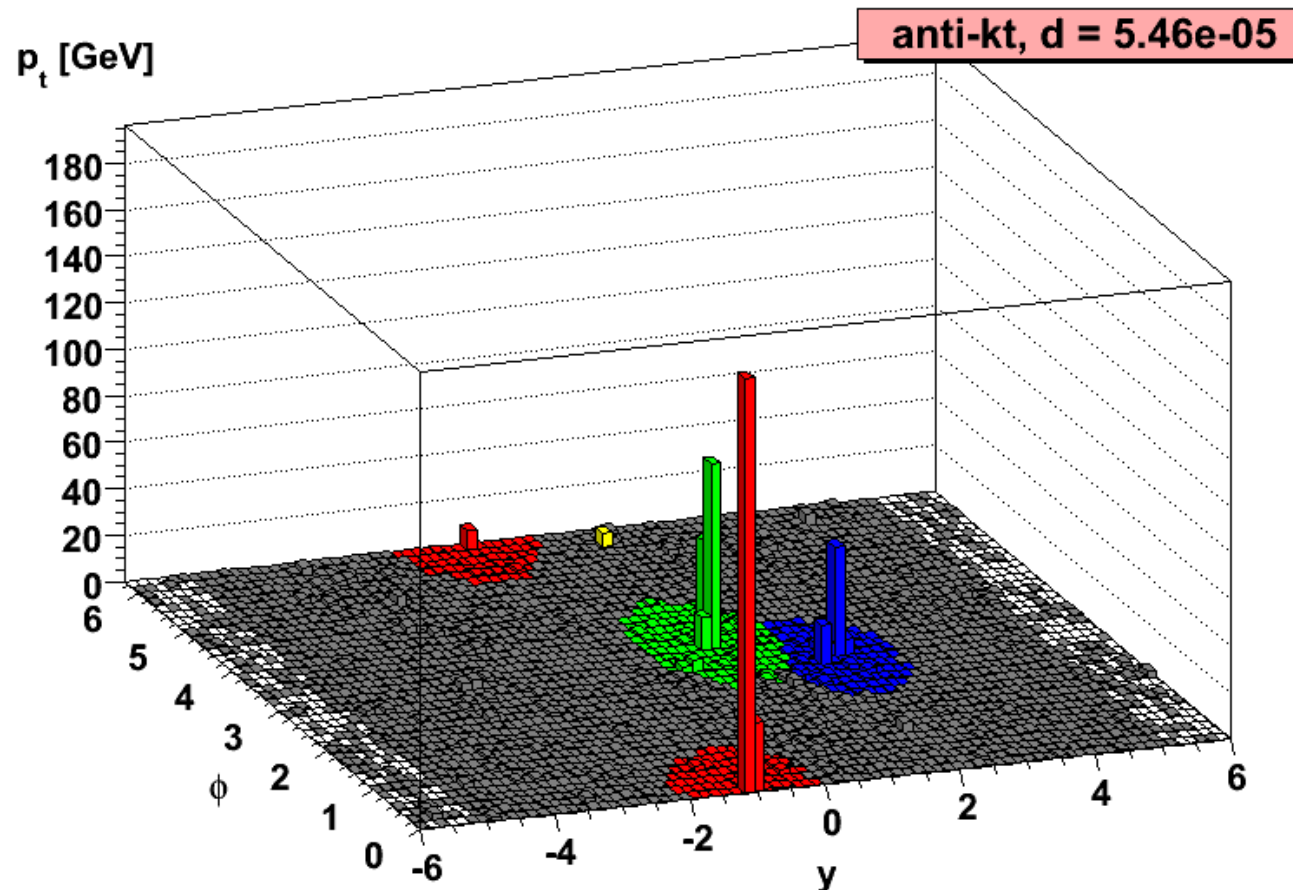
Clustering grows  
around hard cores

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



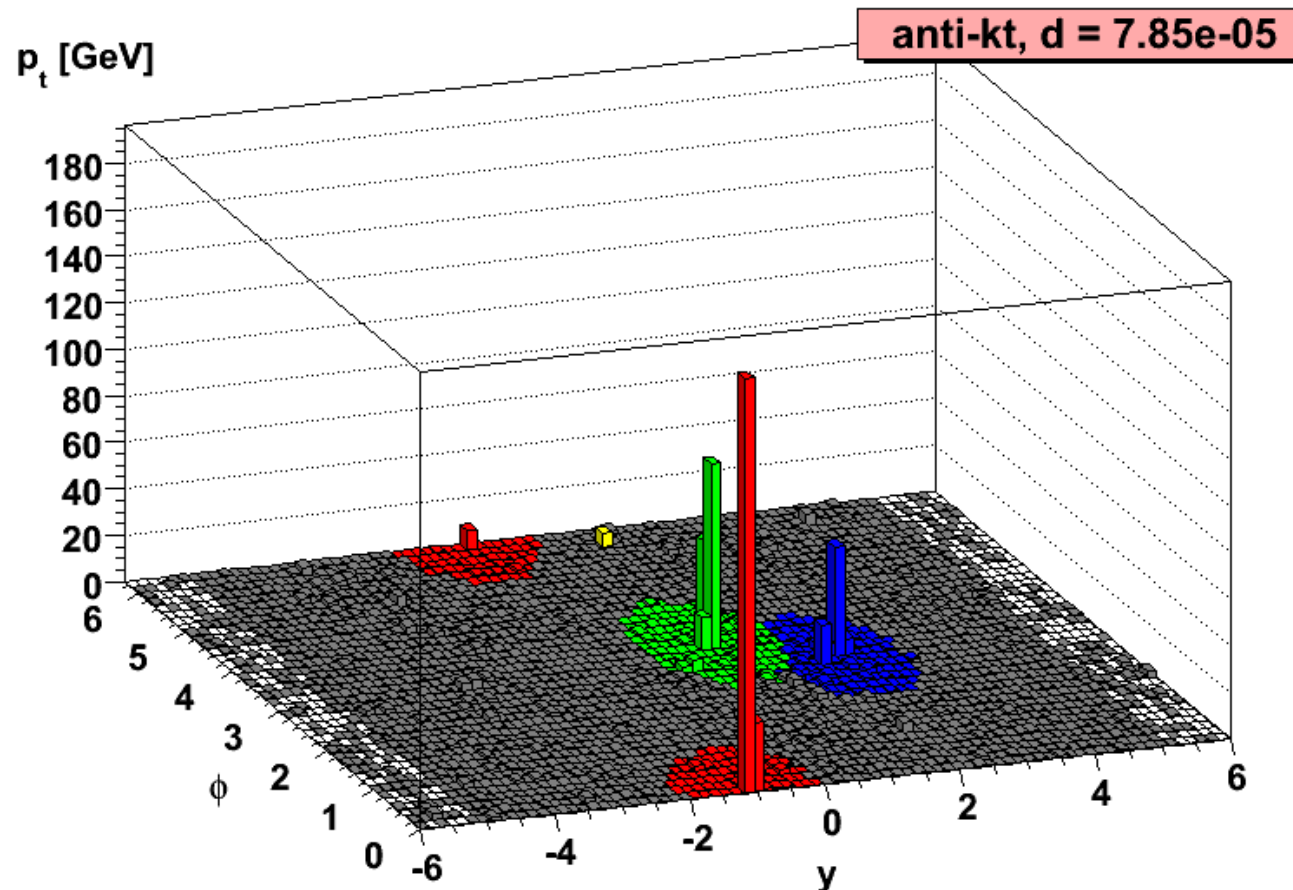
Clustering grows  
around hard cores

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



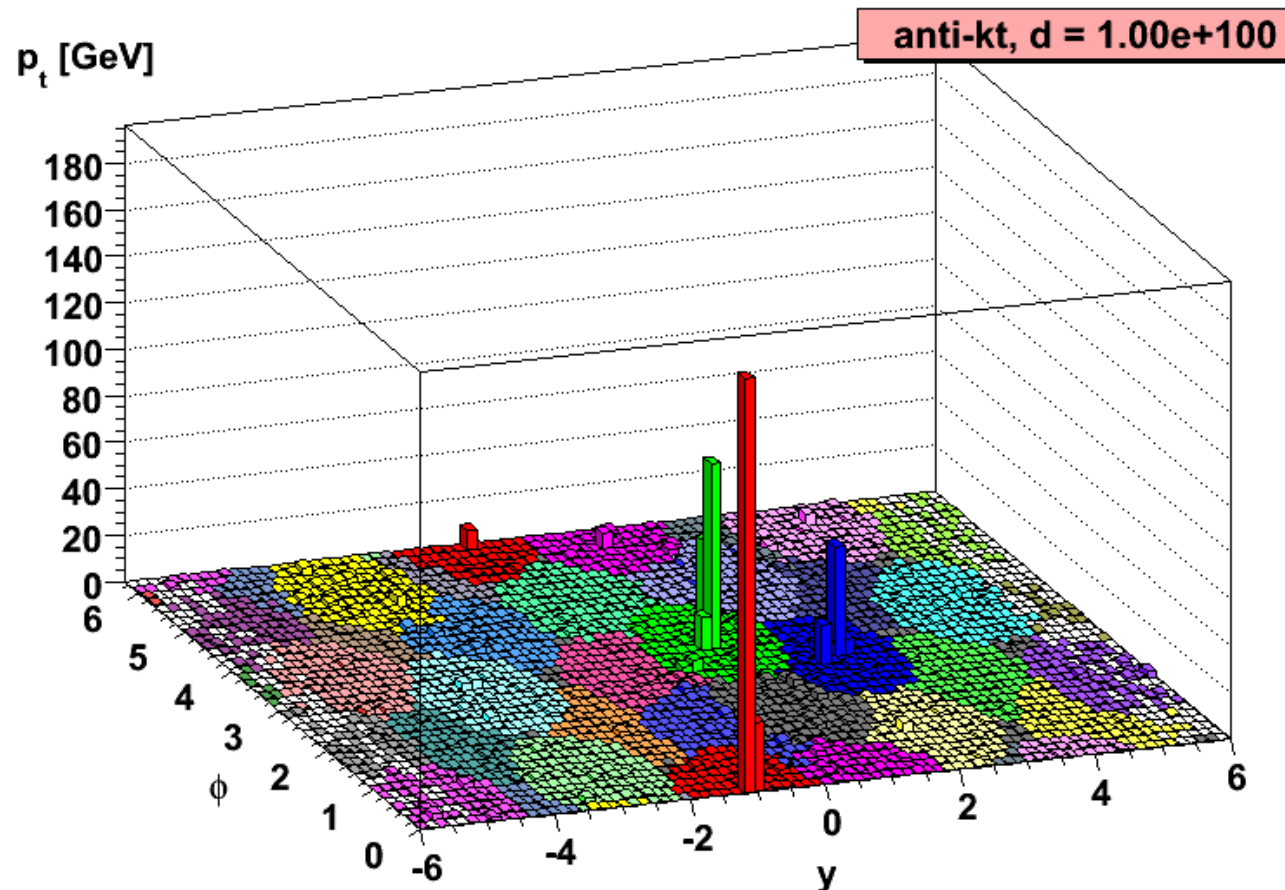
Clustering grows  
around hard cores

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

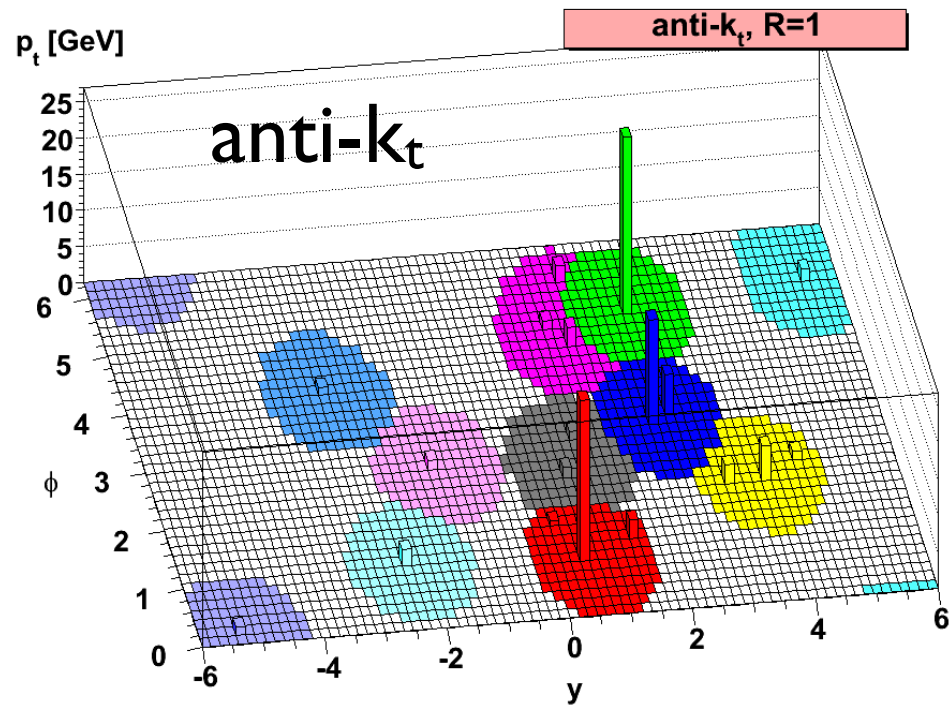
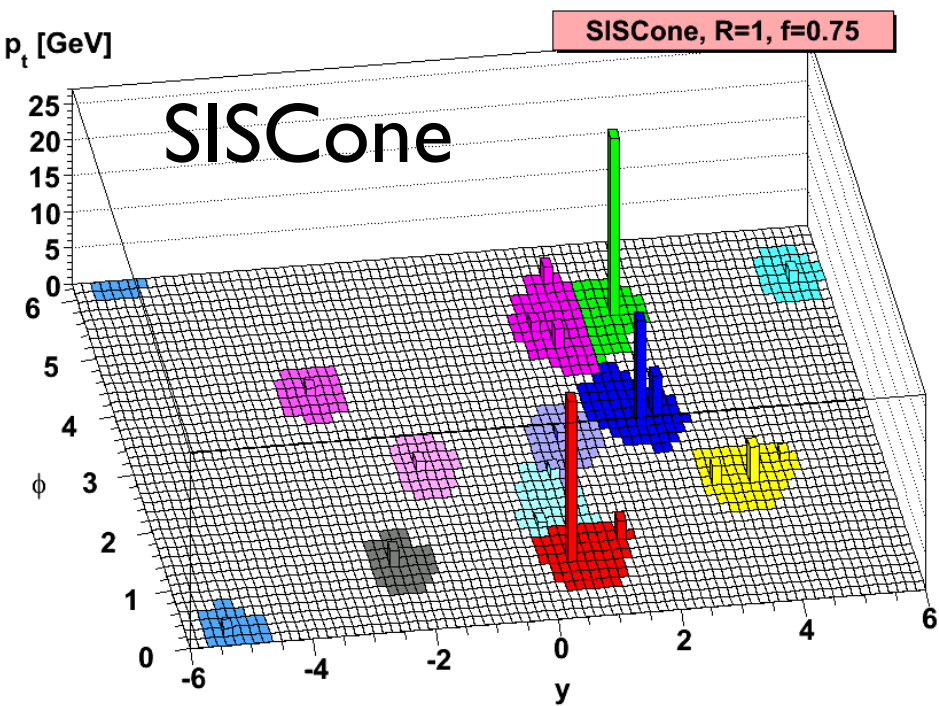
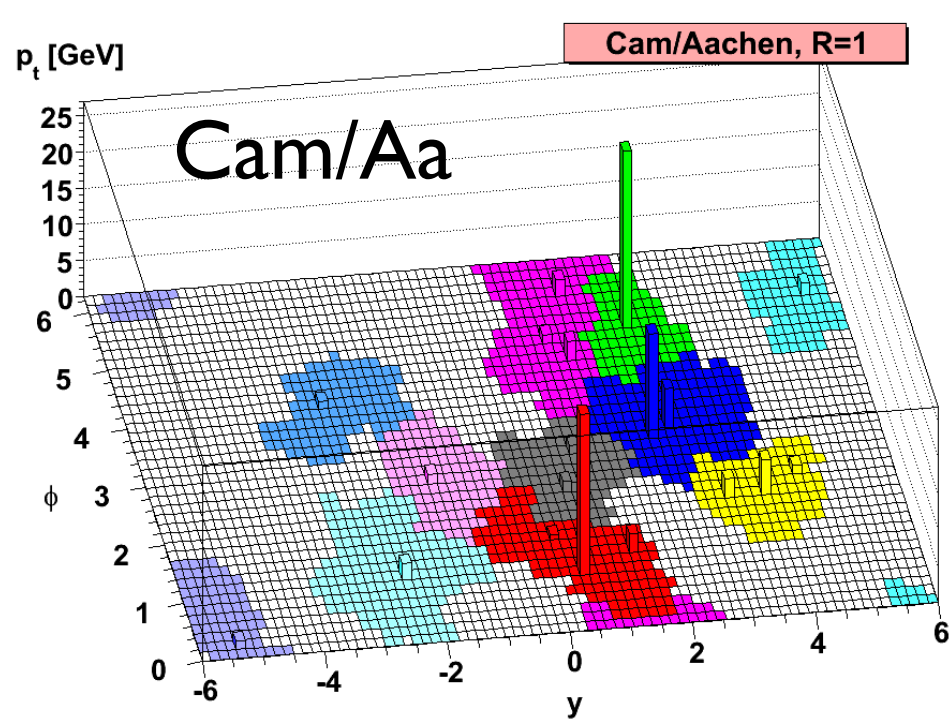
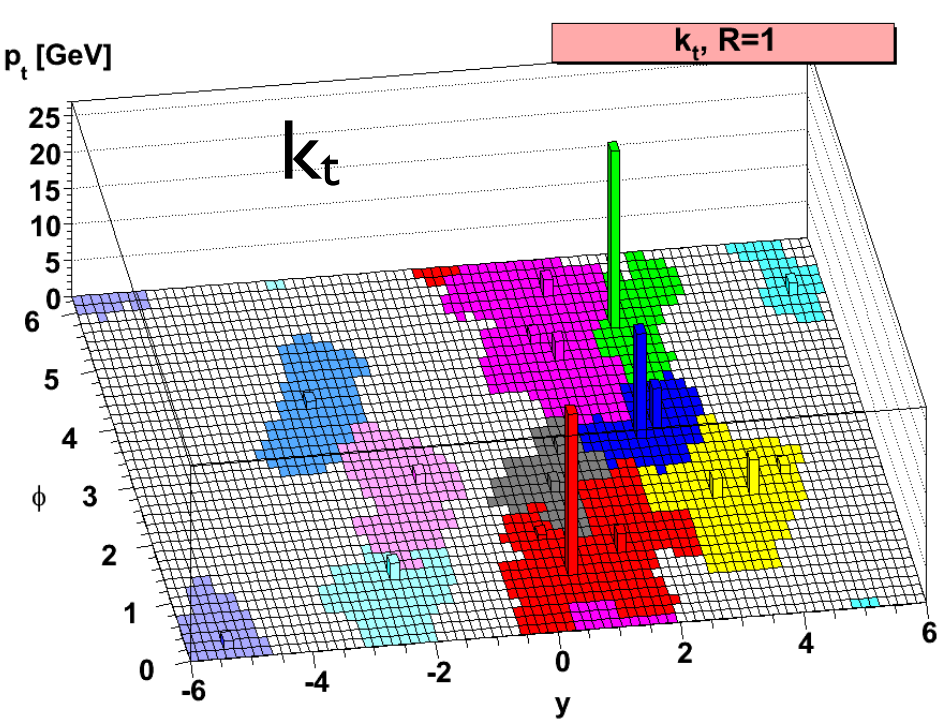


Clustering grows  
around hard cores

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

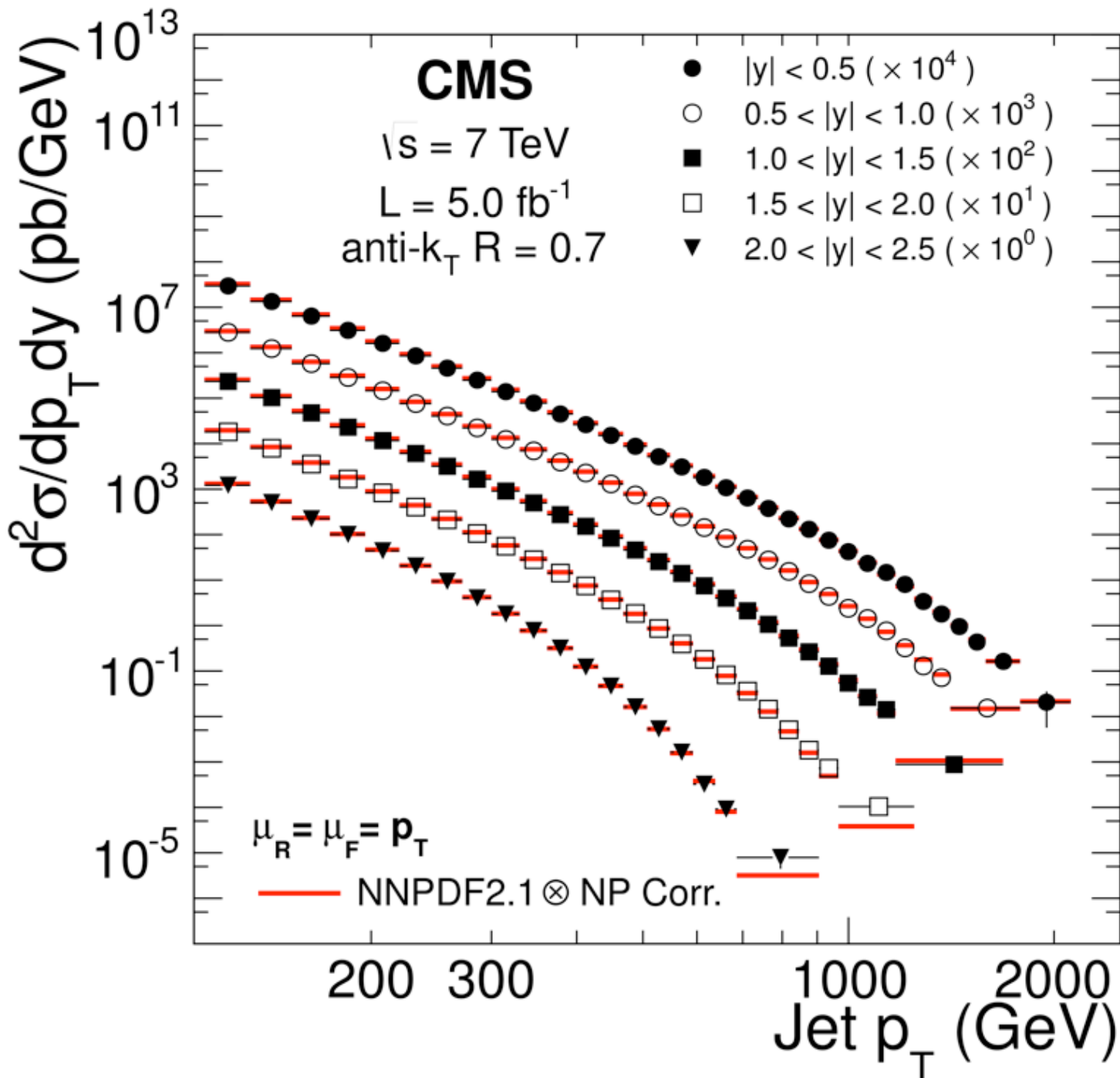


**Anti- $k_t$  gives  
circular jets  
("cone-like")  
in a way that's  
infrared safe**





# Example of jet observable



Inclusive  
jet cross  
section

Excellent  
theory-data  
agreement over  
many orders of  
magnitude

# Recap of Lecture I

- ▶ 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

# Jets

Matteo Cacciari

LPTHE Paris and Université Paris Diderot

Lecture 1 - Jet algorithms

**Lecture 2 - Jet substructure**

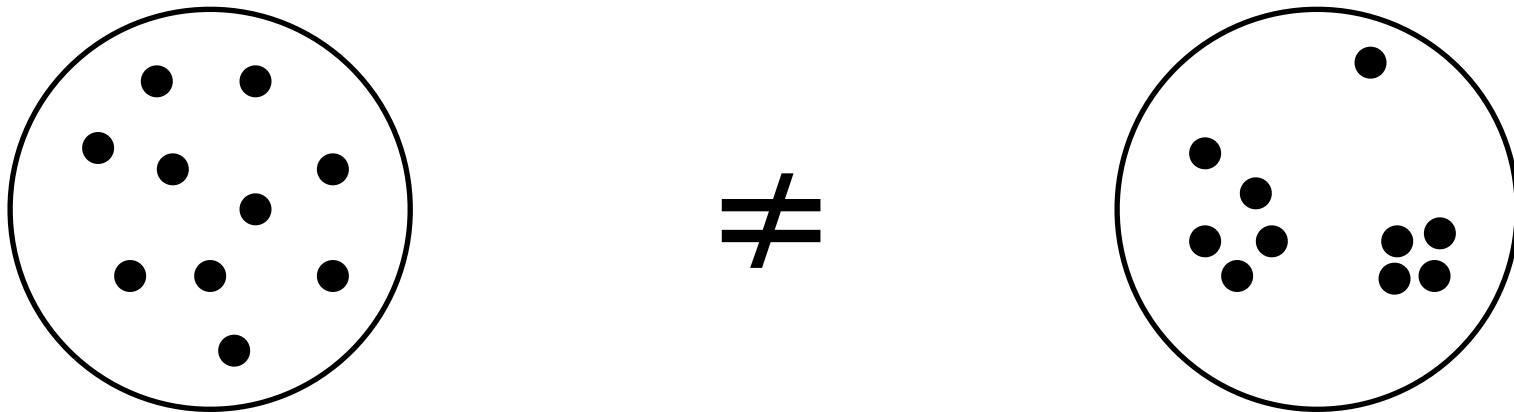
[Includes material from  
Gavin Salam and Grégory Soyez]



# Jet substructure

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)

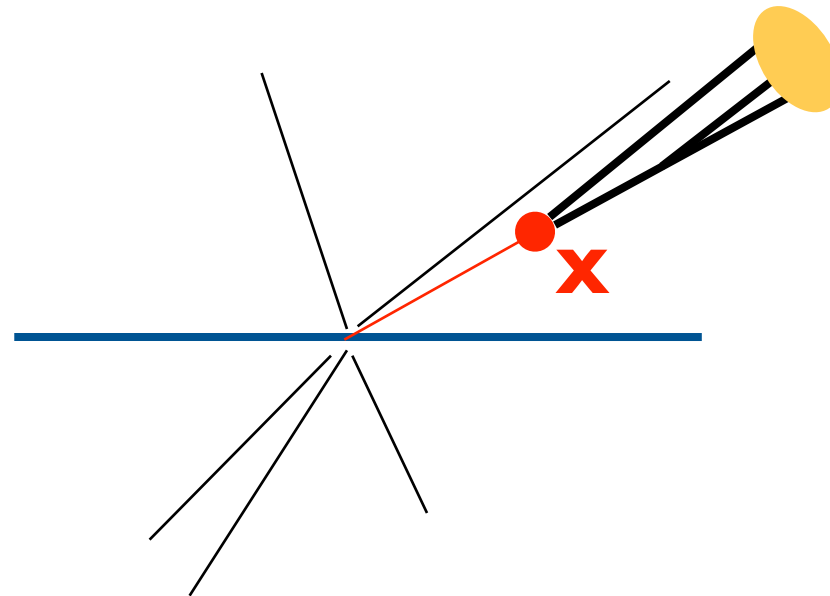
What is the **arrangement** of the constituents inside the jet?



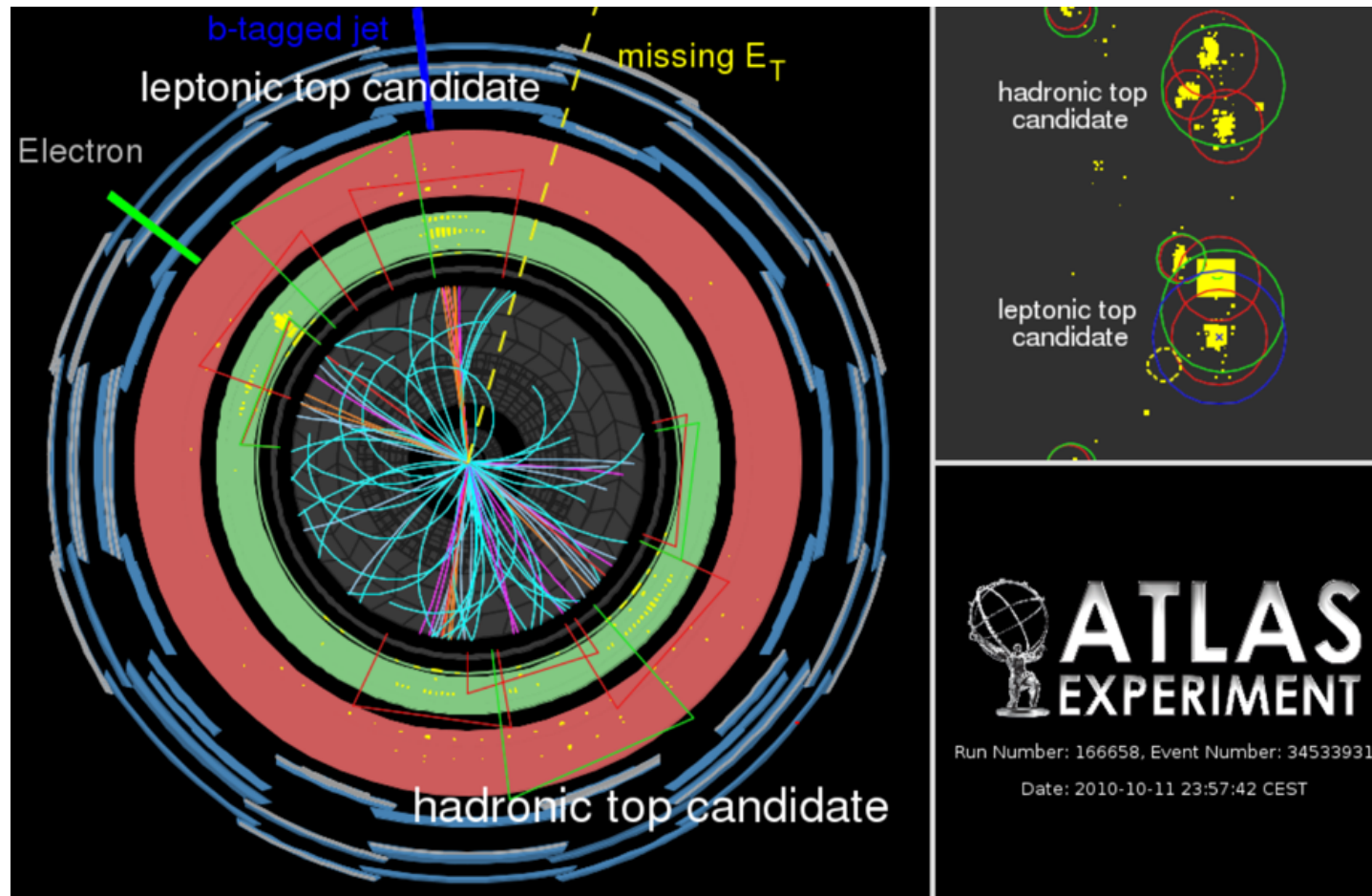
# Jet substructure

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

Topic revived about 10 years  
ago in order to study boosted objects



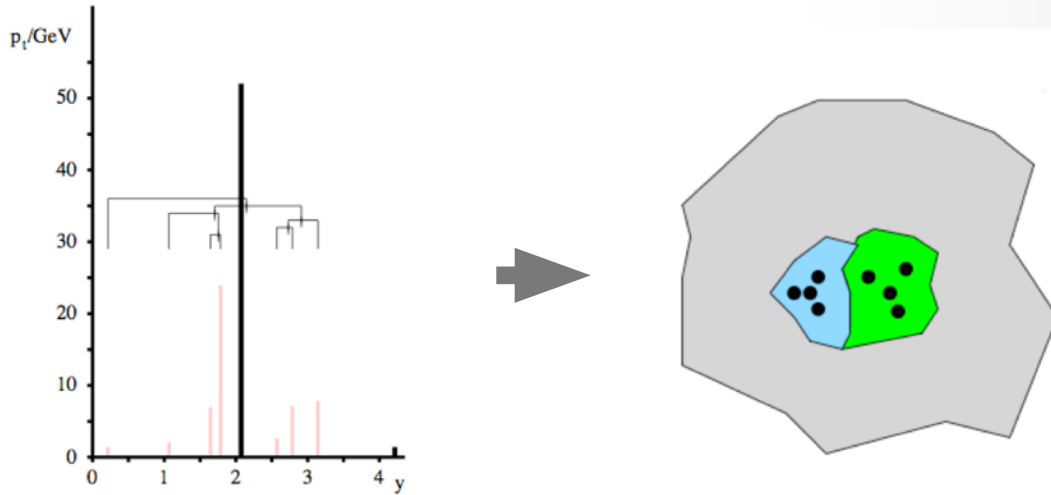
# Jet substructure



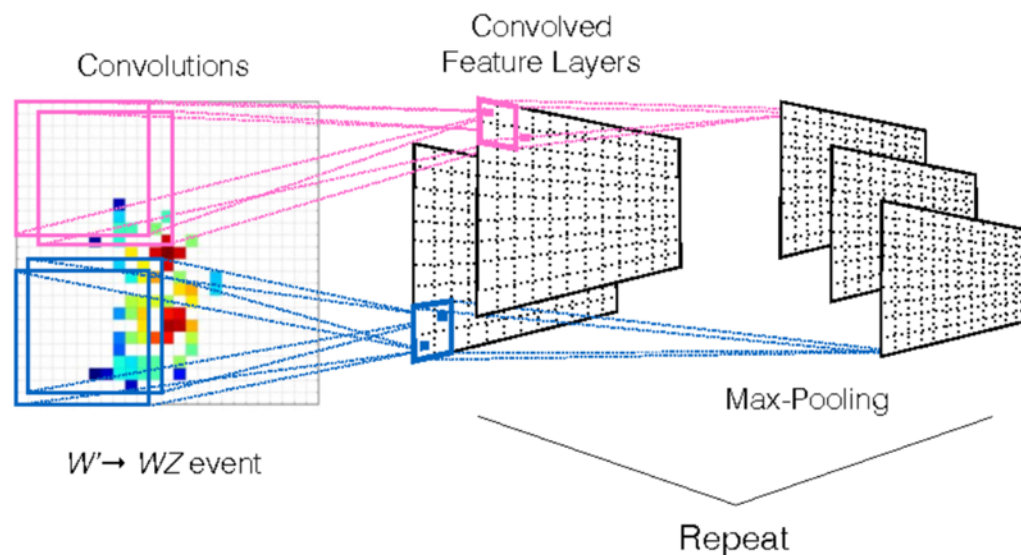
The past ten years have seen an explosion in jet substructure studies, i.e. **how radiation is arranged within jets**, and what it can tell us

# Jet substructure

Jet  
declustering

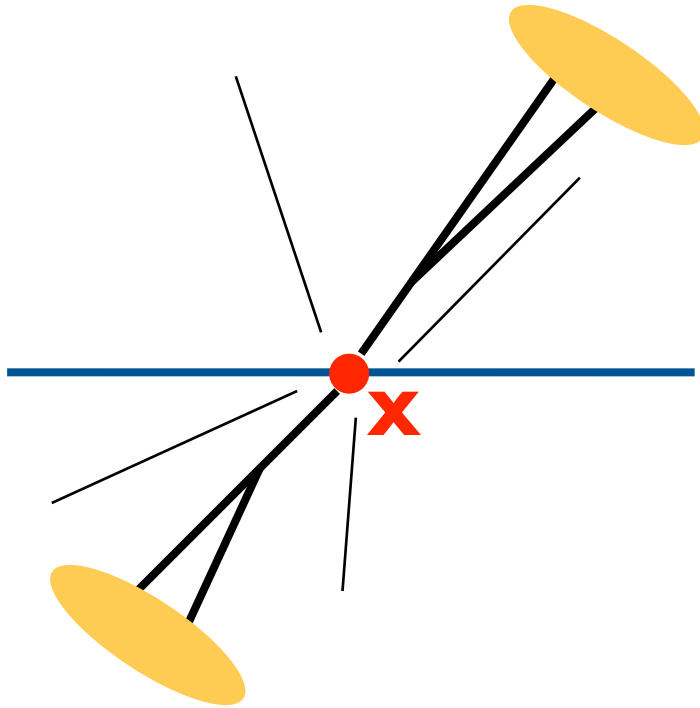


Jet shapes  
(calculate a function from  
radiation distribution)



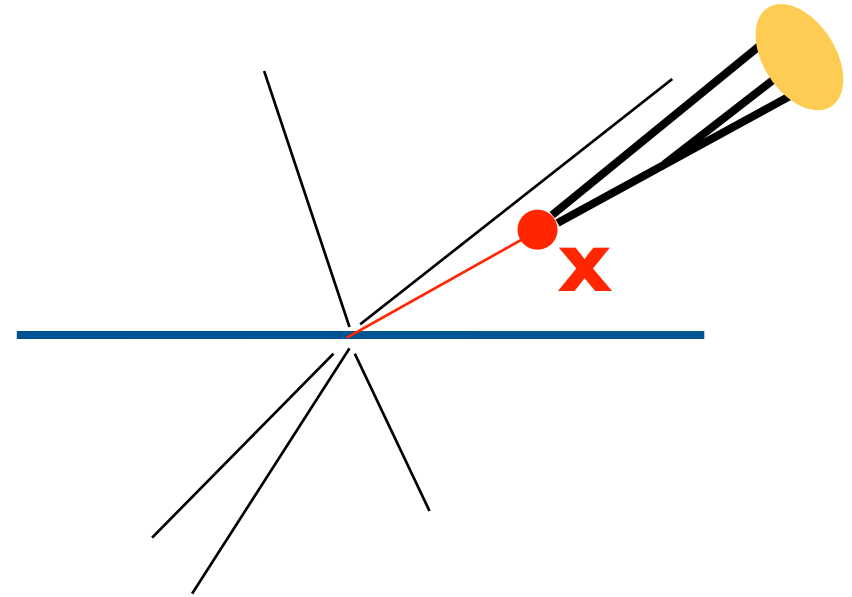
Machine learning

# Why boosted objects



Heavy particle X at **rest**

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

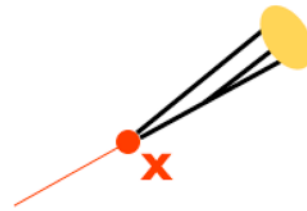
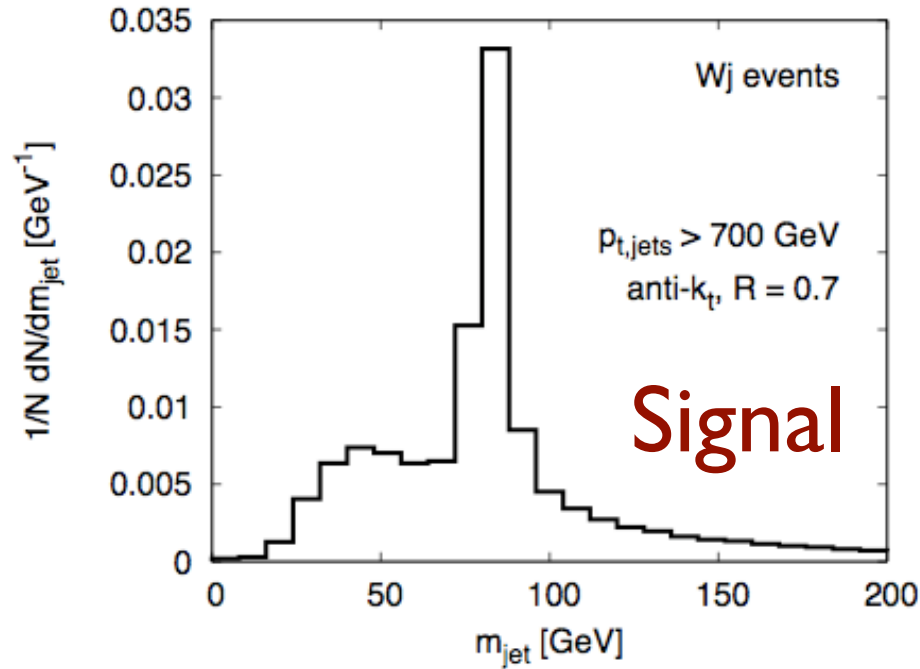


**Boosted** heavy particle X

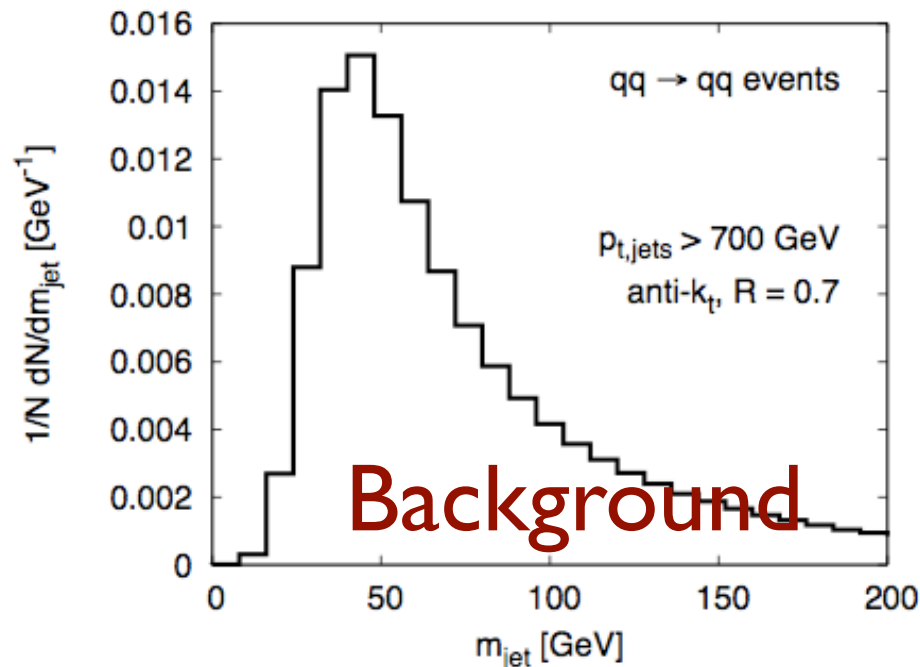
Cross section very much reduced, but acceptance better and some backgrounds smaller/reducible

# Mass of a single jet

G. Salam



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

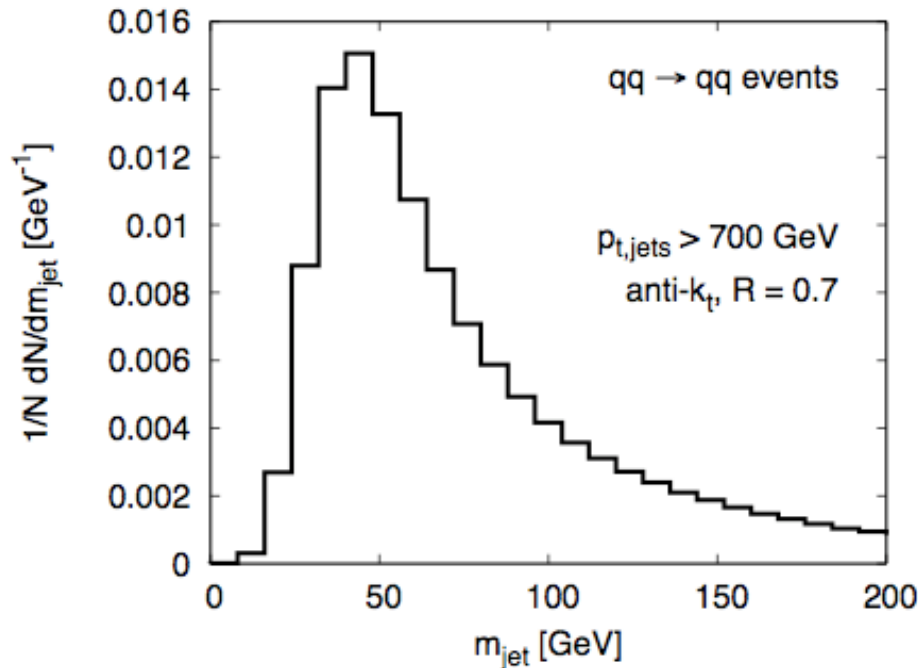


... but pure QCD jets can be massive too:

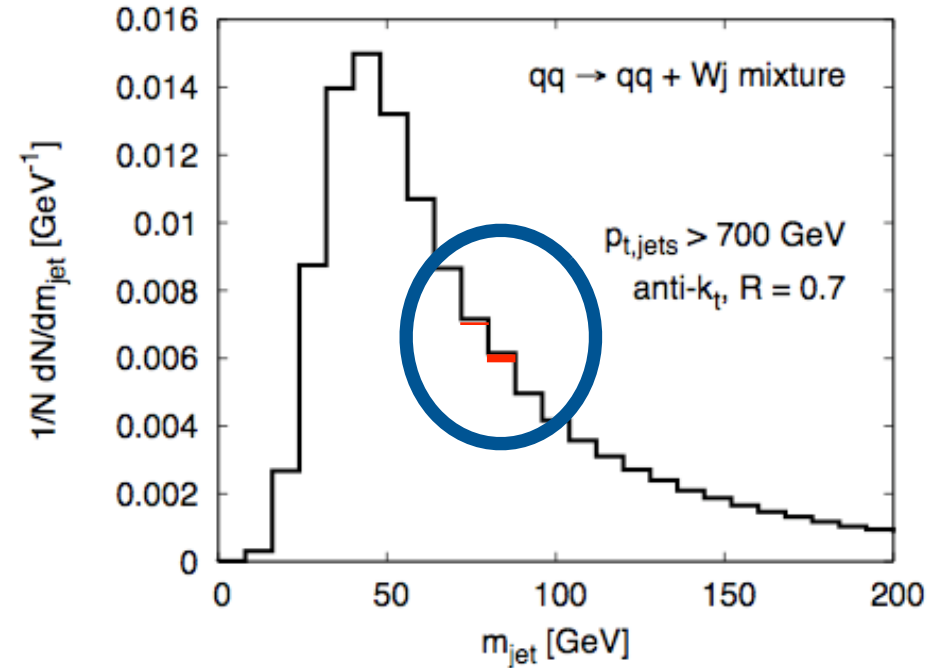
$$\frac{dN}{d \ln m} \sim \alpha_s \ln \frac{p_t R}{m} \times \text{Sudakov}$$

# Mass of a single jet

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



**Background only**



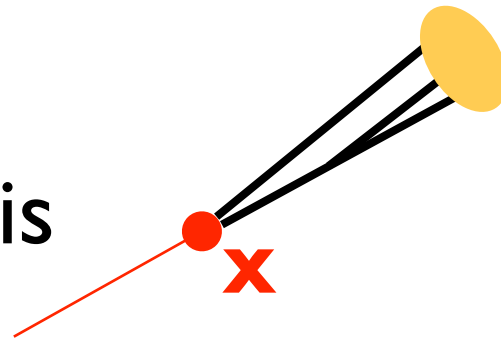
**Signal + background**

**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

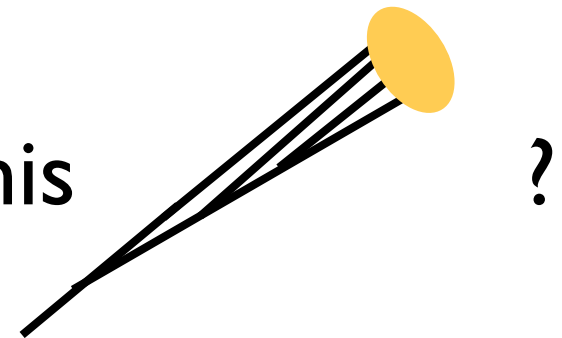


How to tell this



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

## ► 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

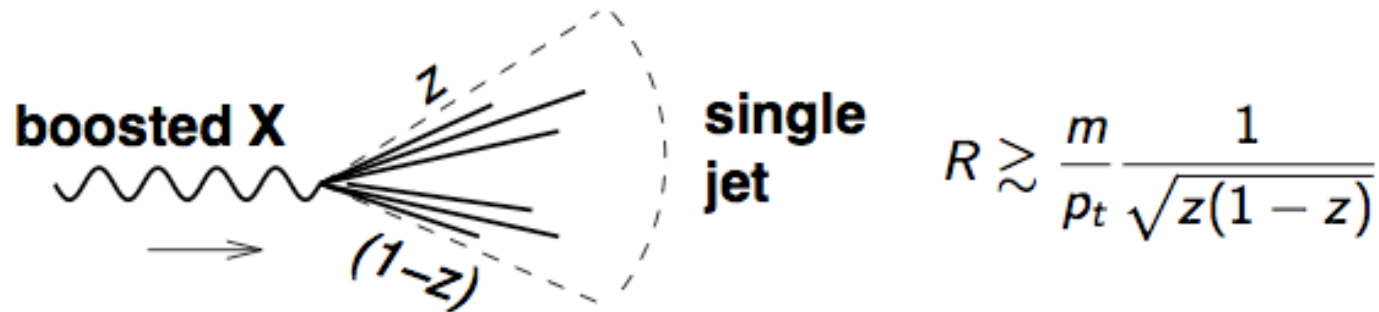
- procedure that might not return an output jet (i.e. it either tags a heavy particle originating the jet or returns zero. This is used to identify a specific particle originating the jet.)

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:  $m \sim 100 \text{ GeV}$ ,  $p_t \sim 500 \text{ GeV}$

(e.g. electroweak particle from decay of  $\sim 1 \text{ TeV}$  BSM particle)



- ▶ 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

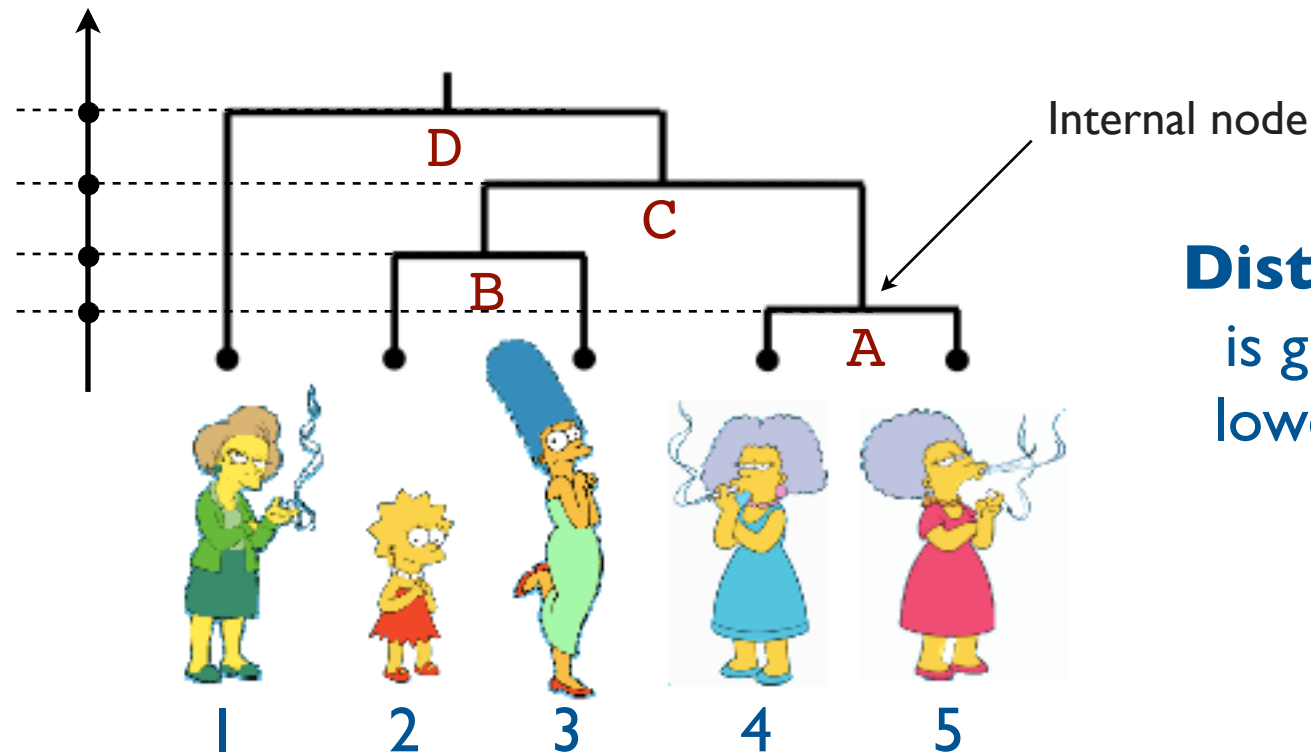
# What jets to use for substructure?

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



**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

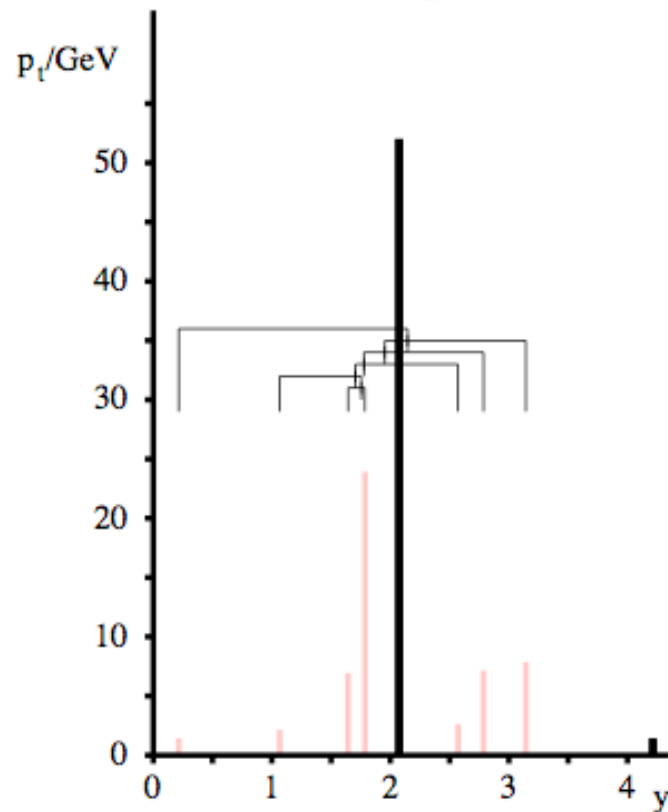
The **clustering sequence** is 4-5 (A), 2-3 (B), 23-45 (C), 1-2345 (D)

anti-kt



# Hierarchical substructure

## anti- $k_t$ algorithm



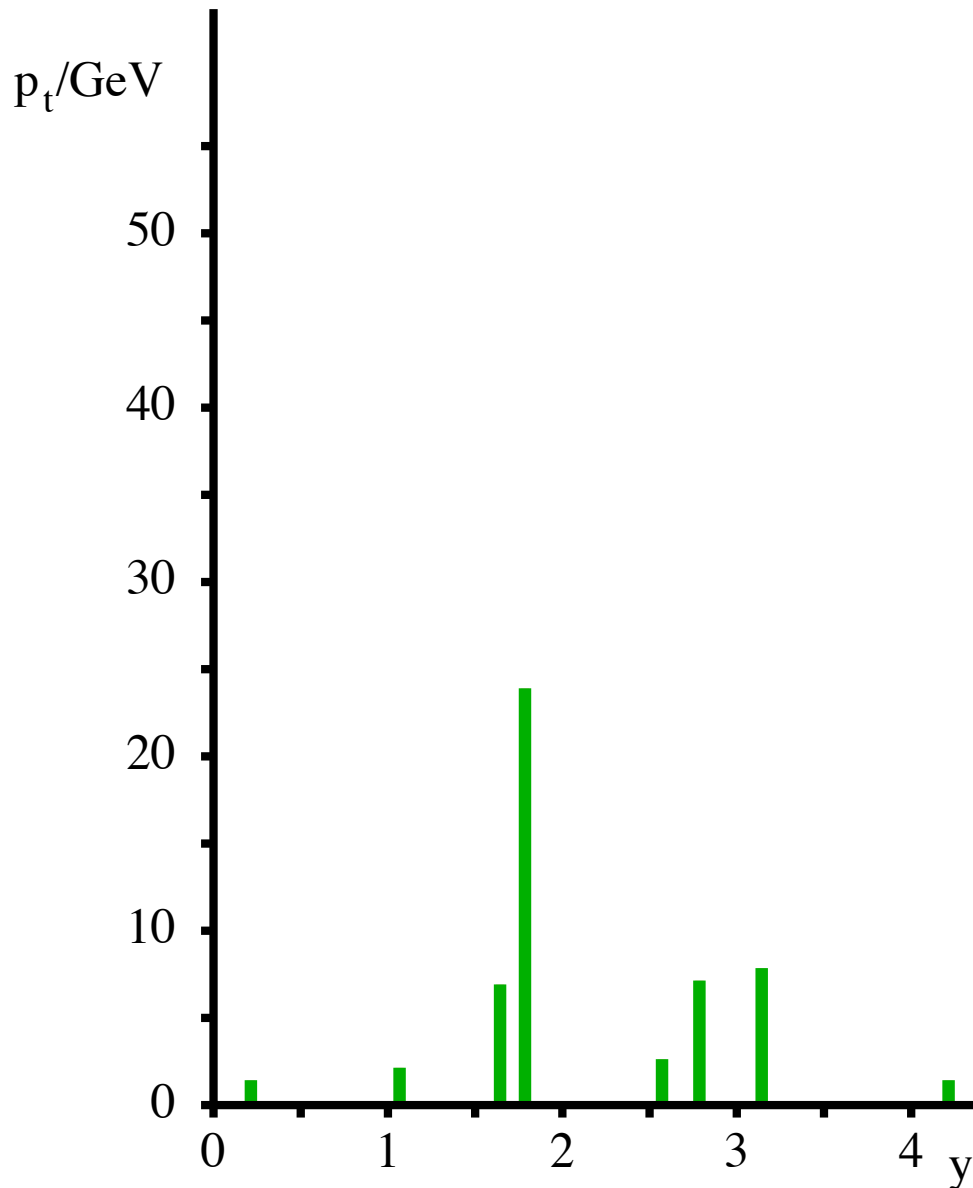
## Anti- $k_t$ distance measure

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

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

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm



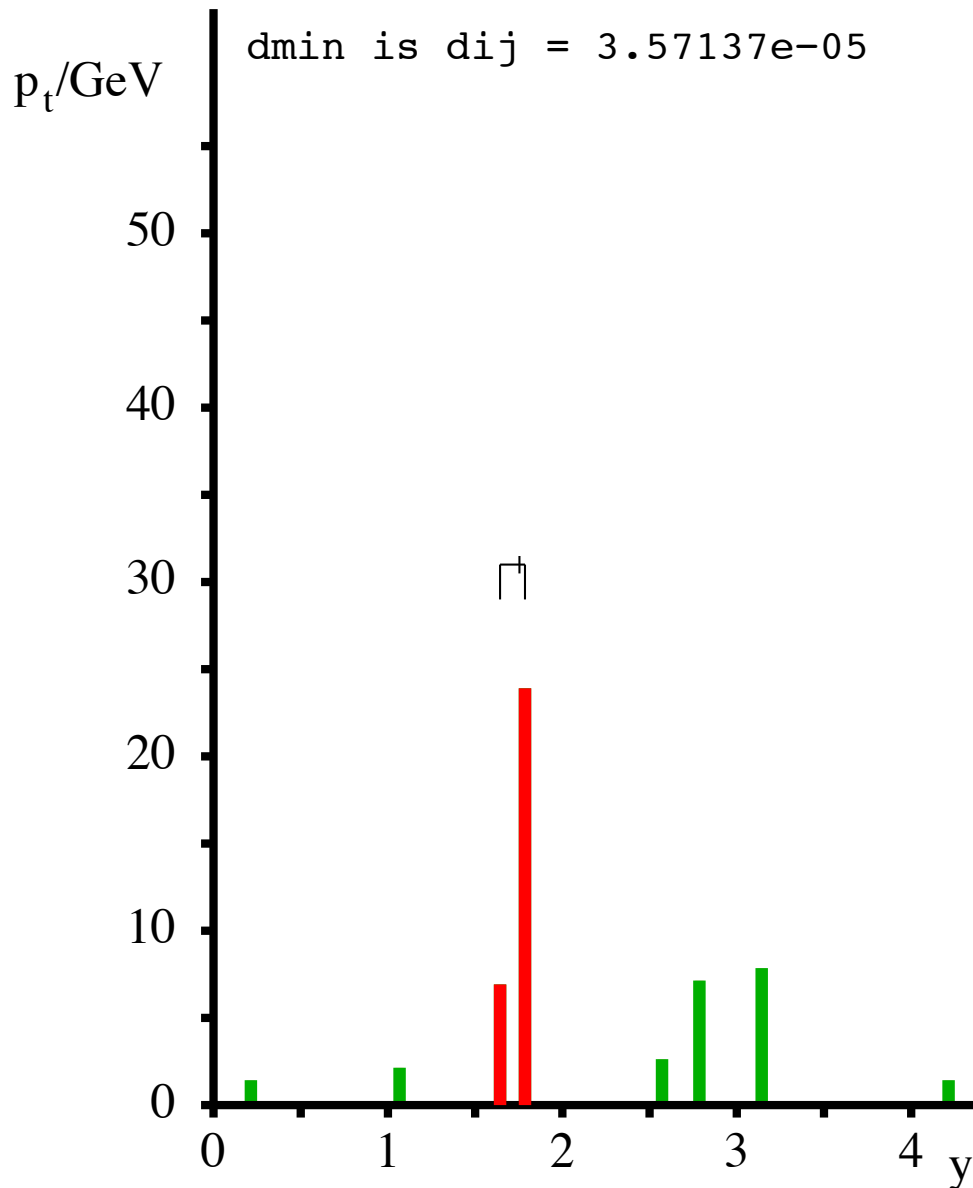
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm

dmin is dij = 3.57137e-05

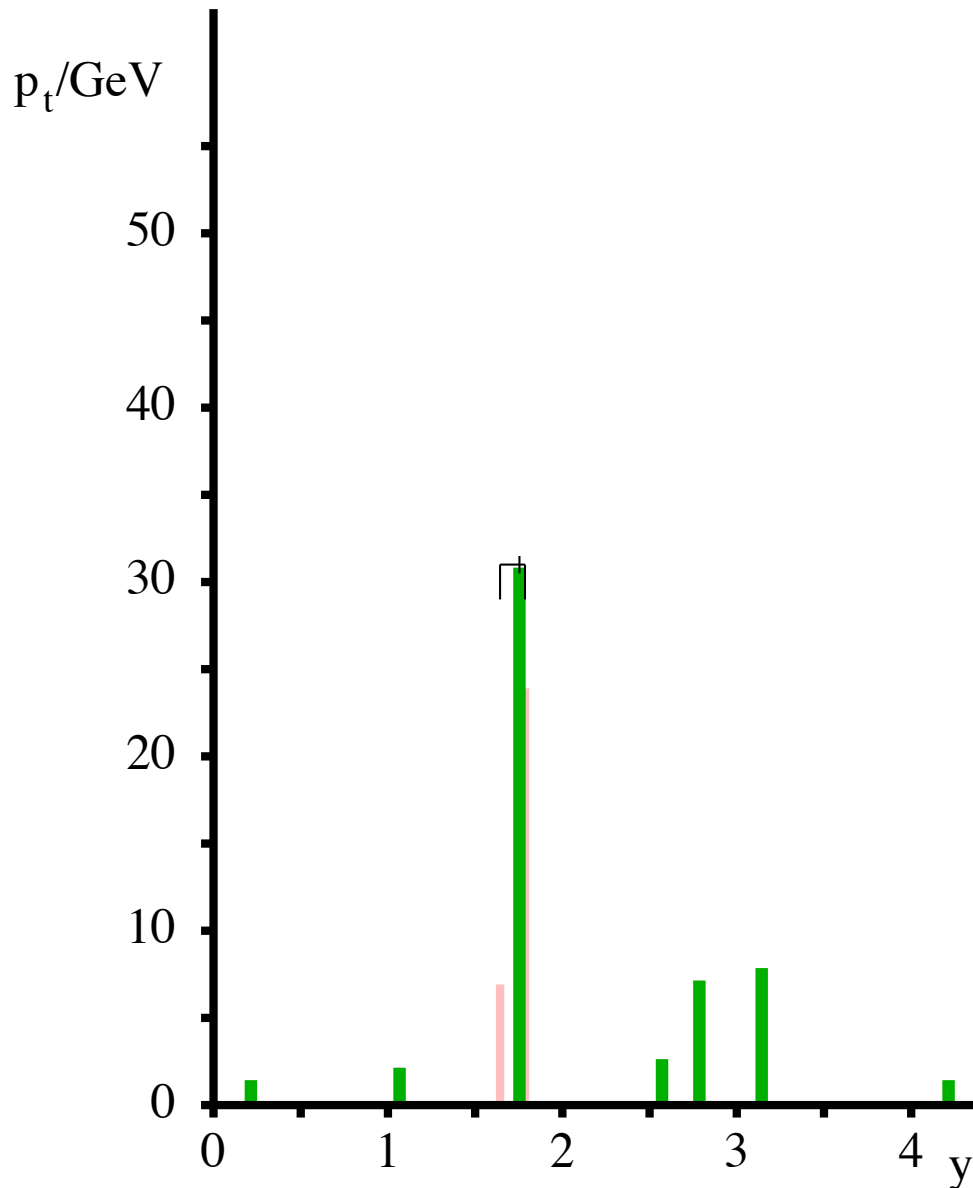


How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm



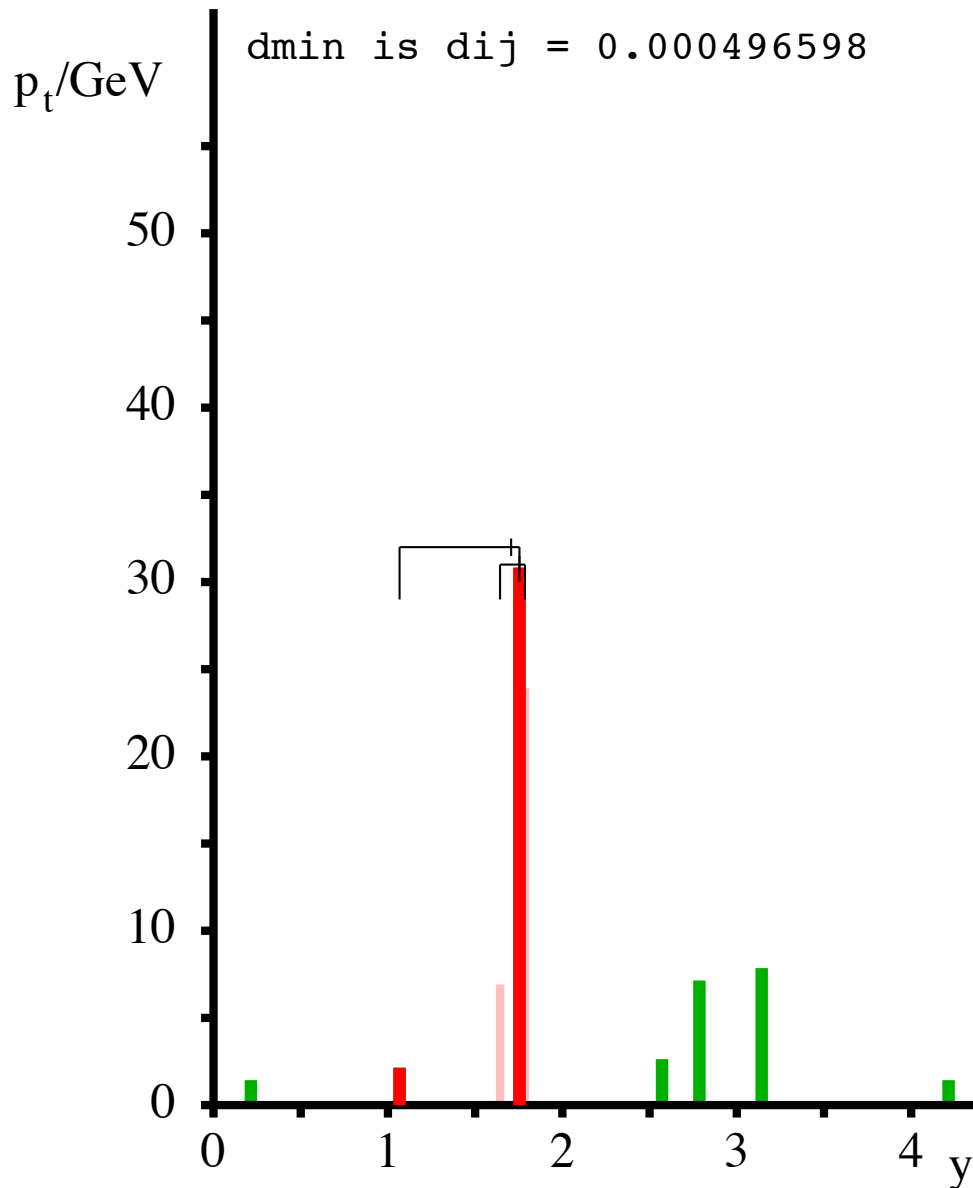
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm

$d_{\min}$  is  $d_{ij} = 0.000496598$

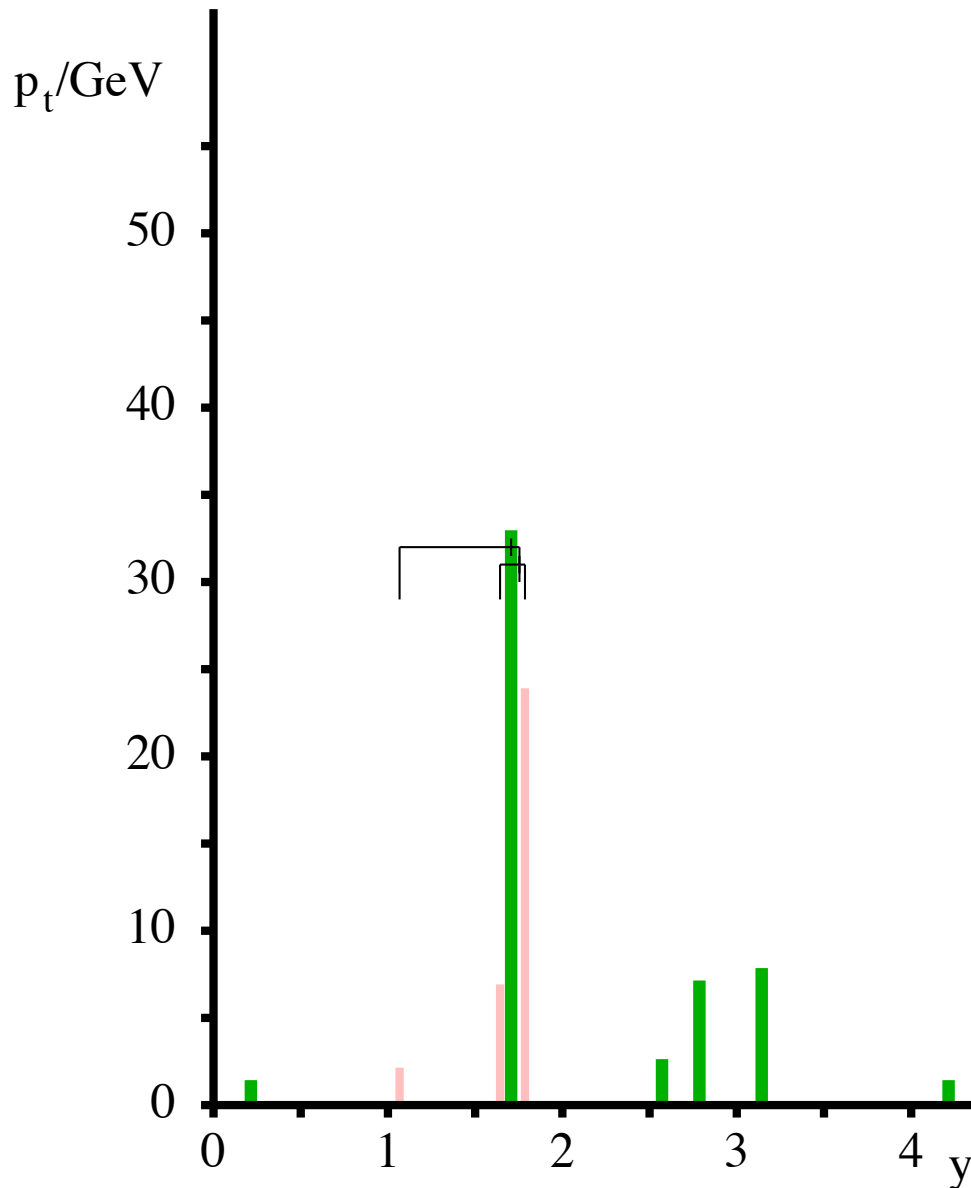


How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm



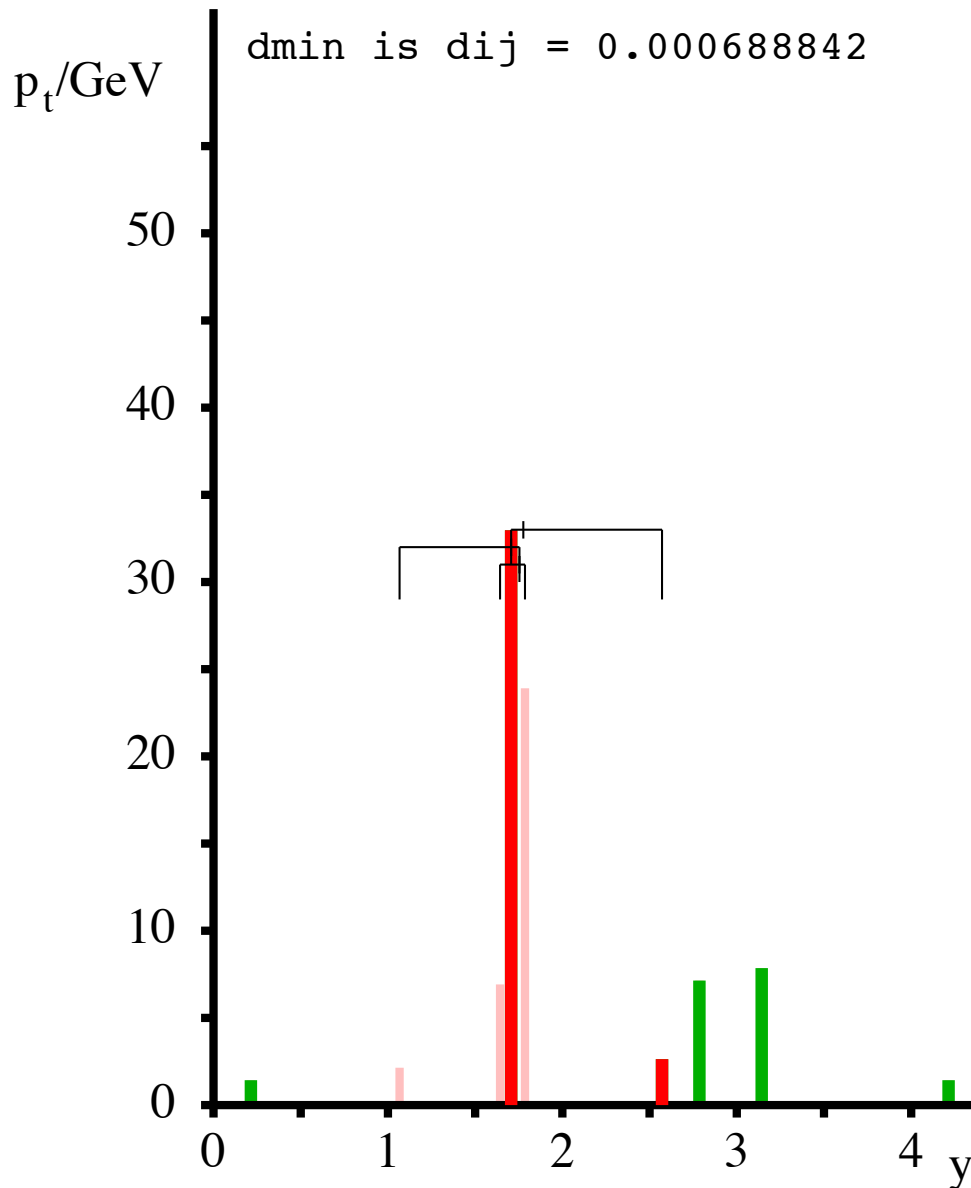
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm

$d_{\min}$  is  $d_{ij} = 0.000688842$



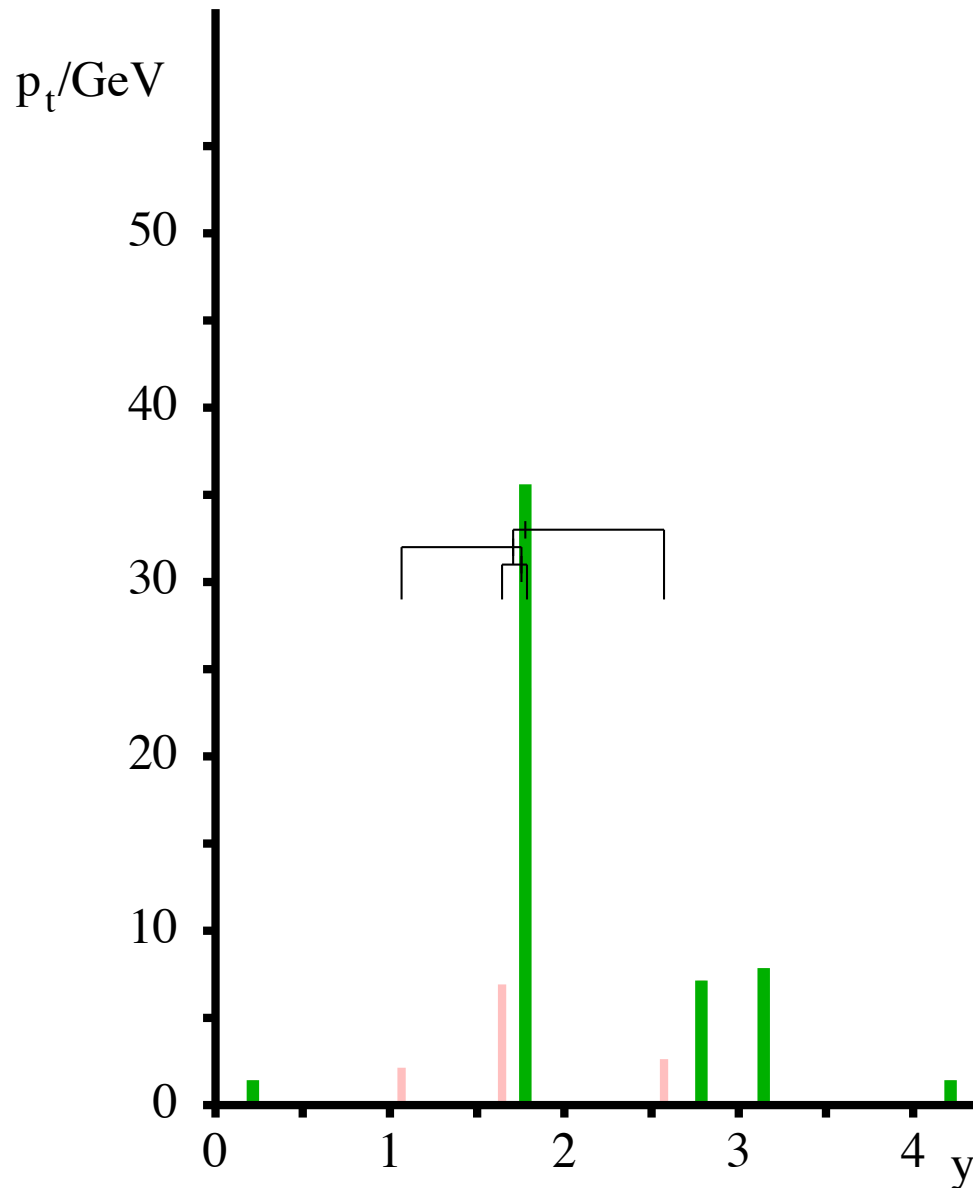
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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



# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm



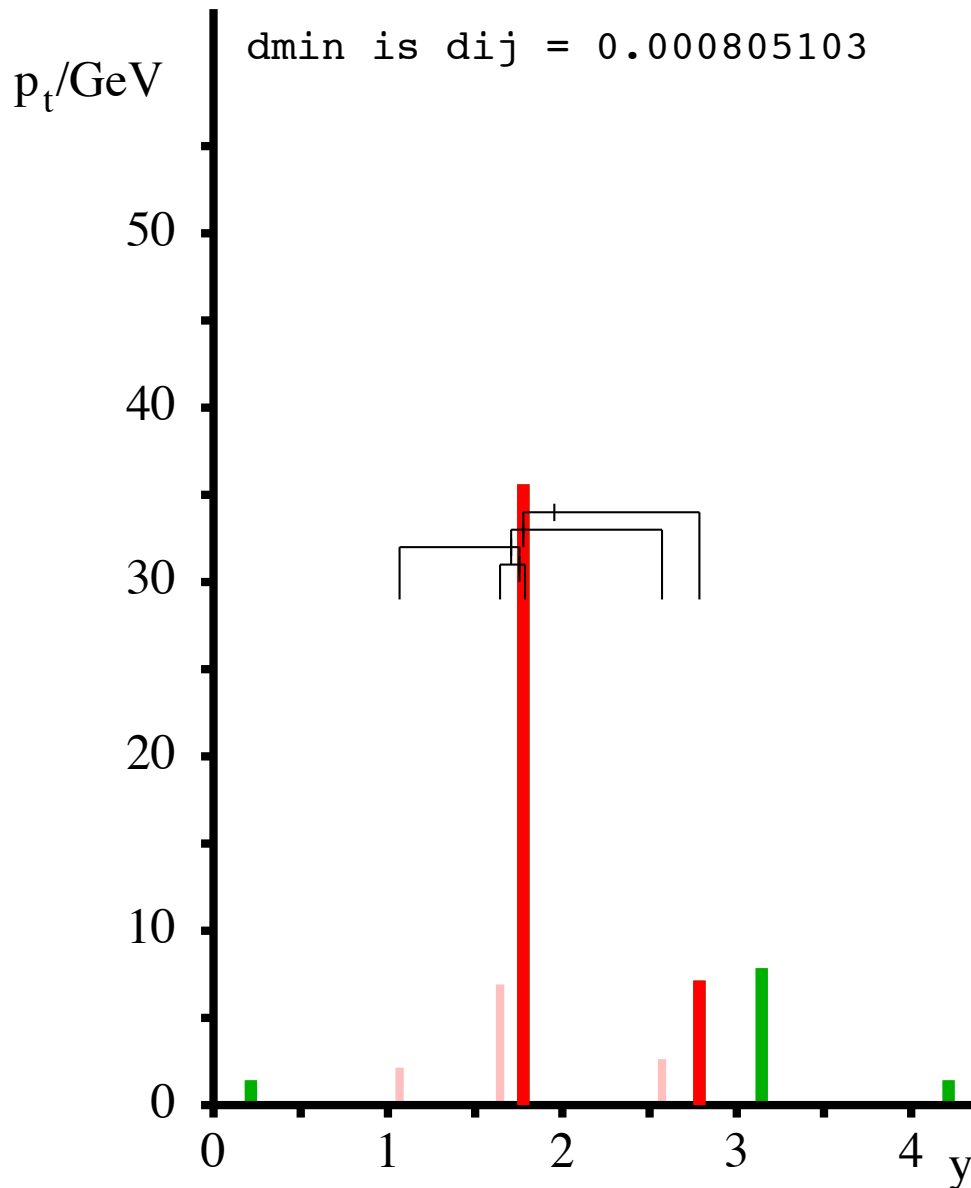
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm

dmin is dij = 0.000805103



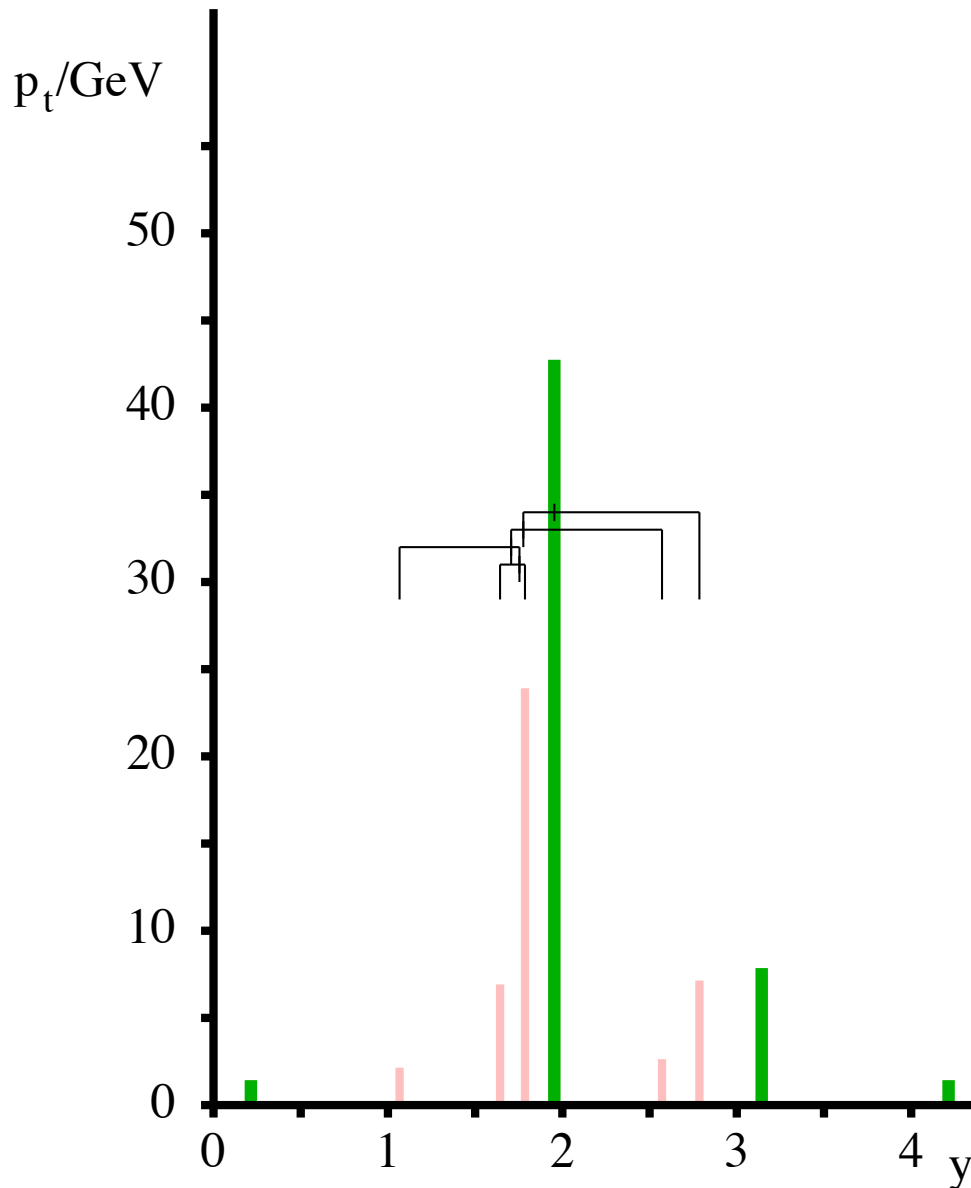
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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.*

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm



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

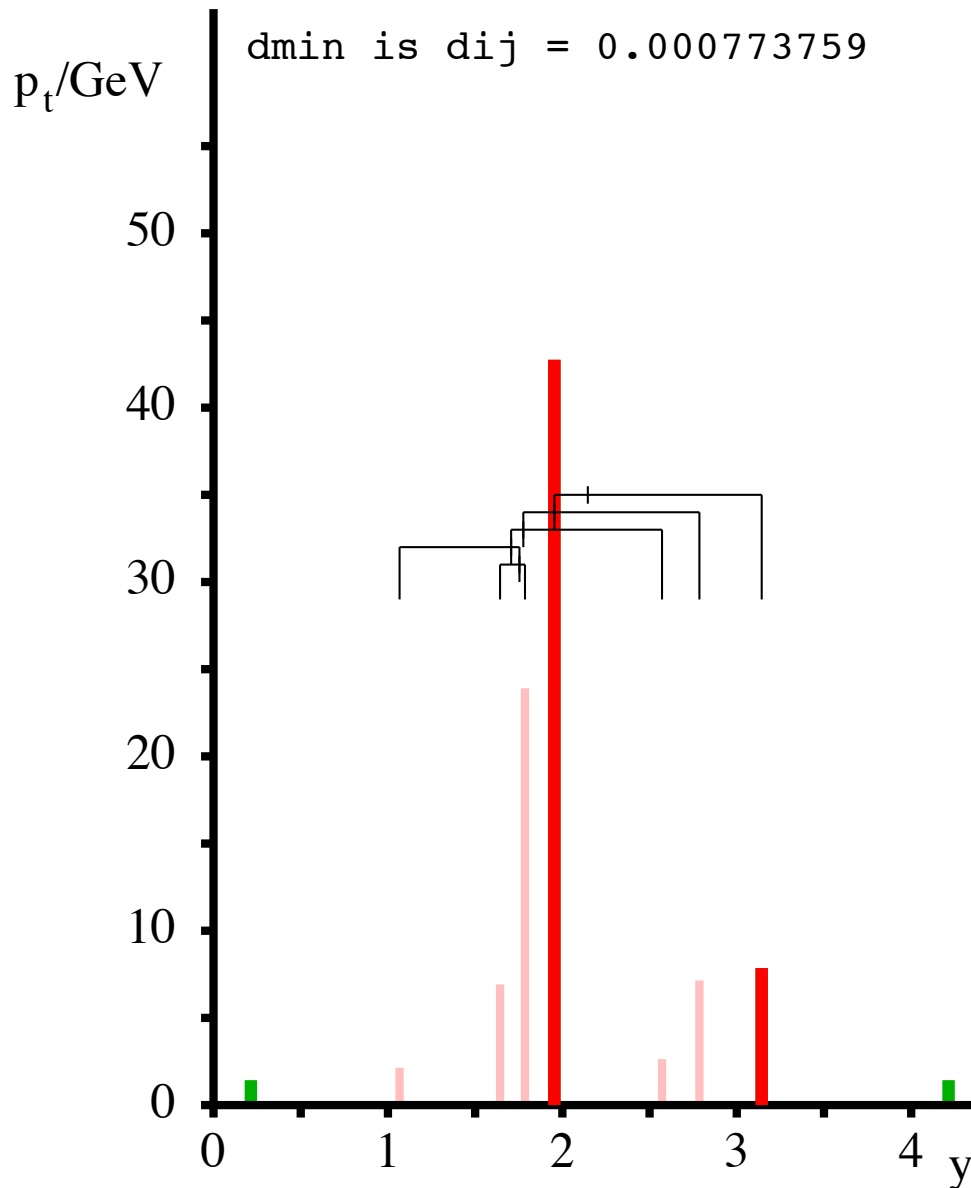
This is crucial for identifying the kinematic variables of the partons in the jet (e.g.  $z$ ).

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

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm

dmin is dij = 0.000773759



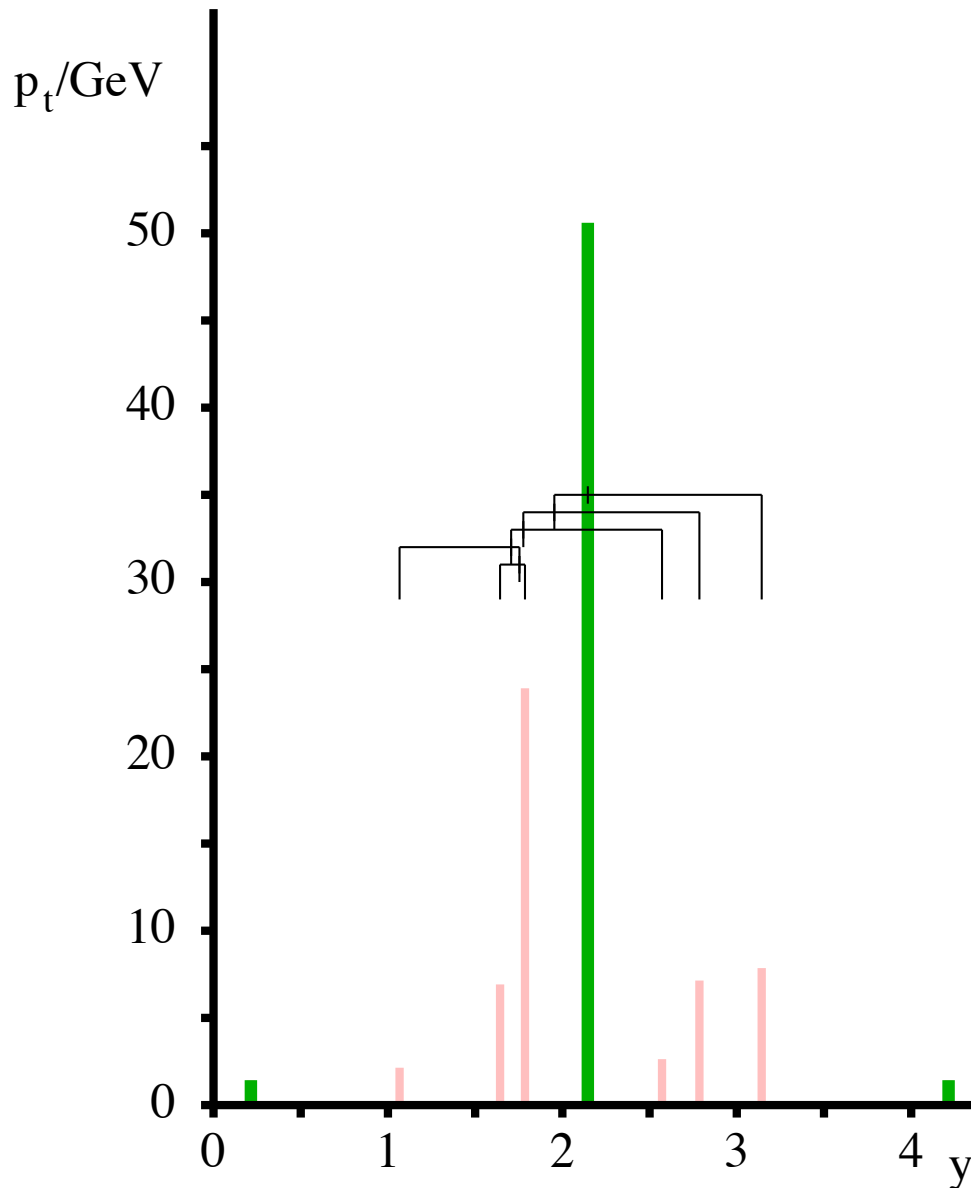
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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.*

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm



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

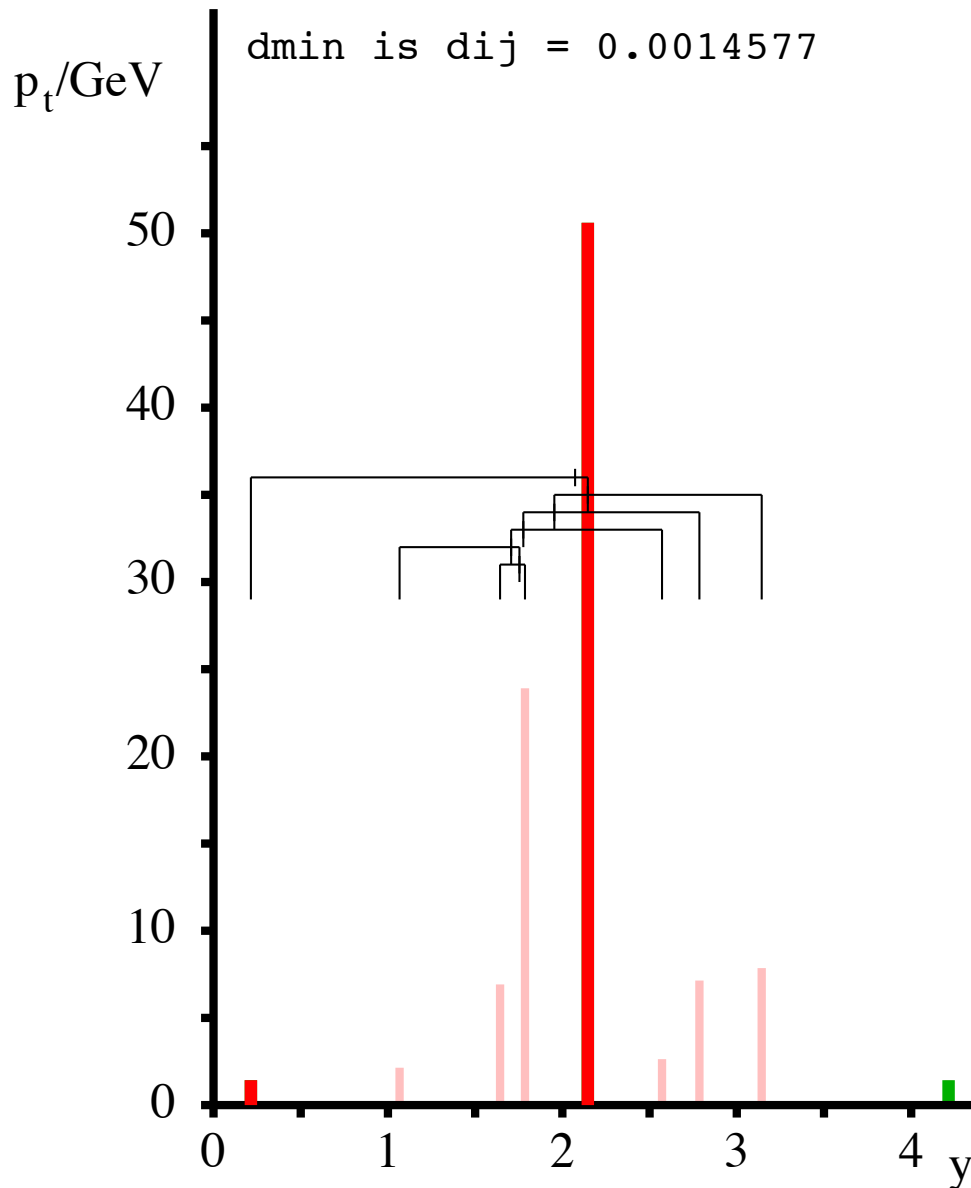
This is crucial for identifying the kinematic variables of the partons in the jet (e.g.  $z$ ).

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

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm

dmin is dij = 0.0014577



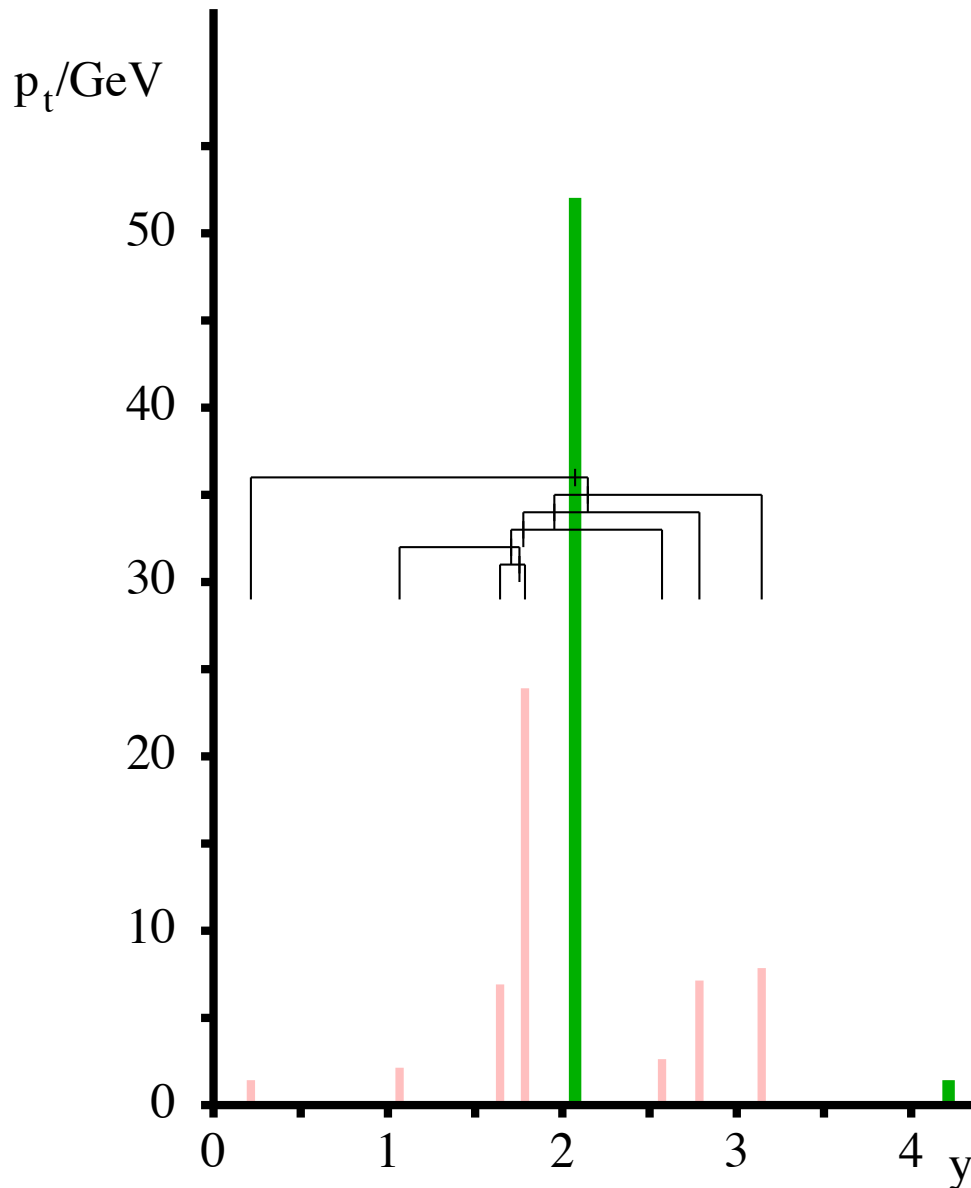
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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.*

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm



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.  $z$ ).

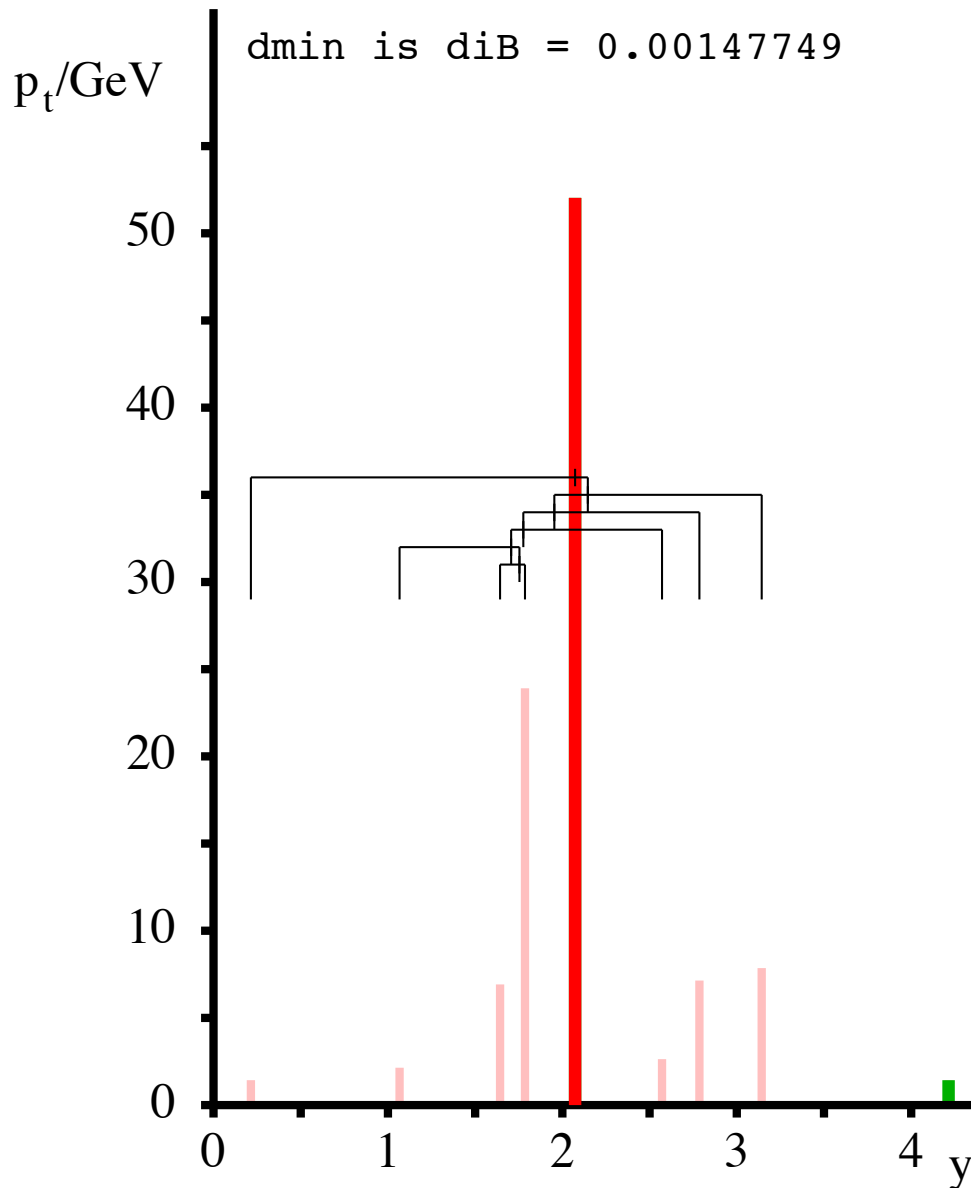
*Anti- $k_t$  gradually makes its way through the secondary blob  $\rightarrow$  no clear identification of substructure associated with 2nd parton.*



# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm

dmin is diB = 0.00147749



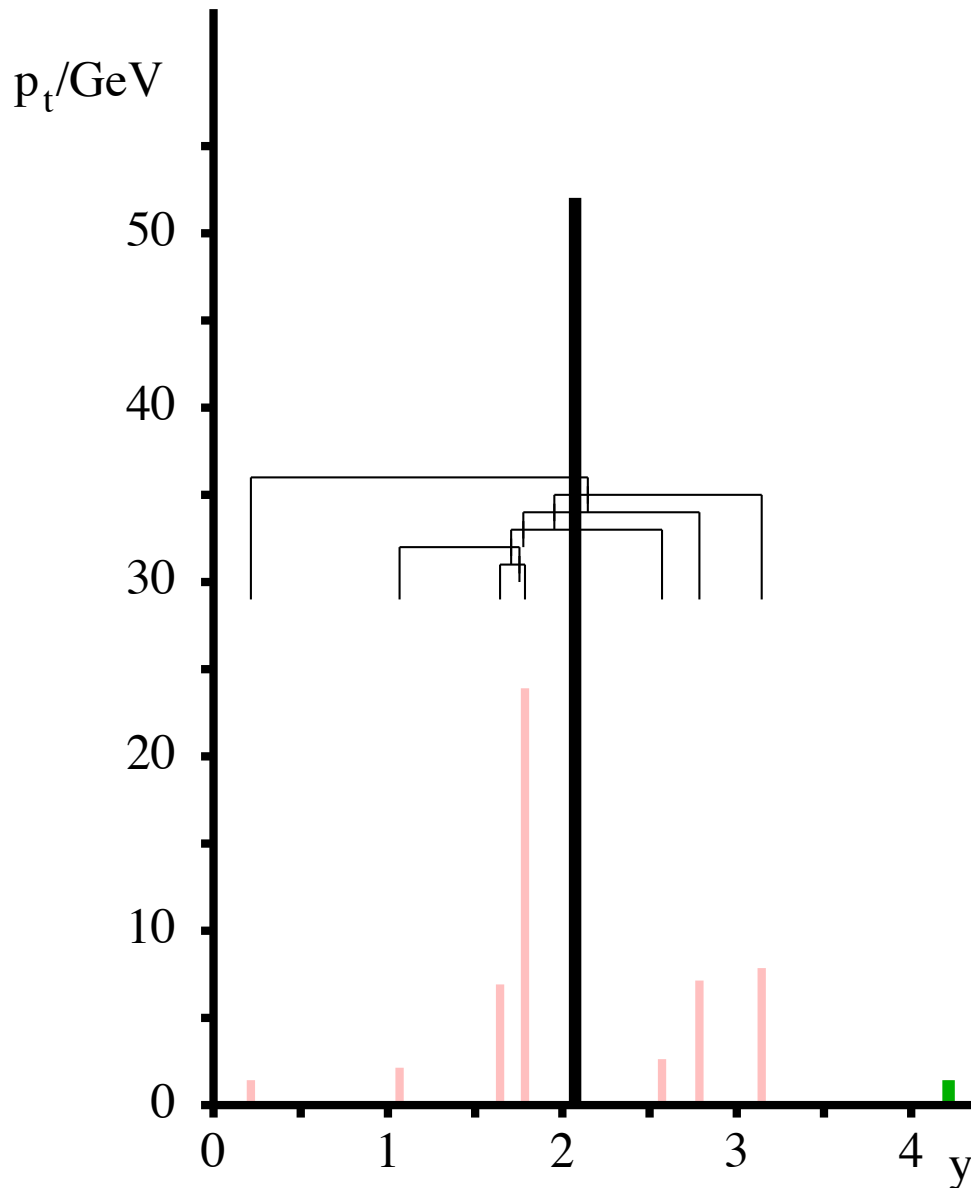
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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.*

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm



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

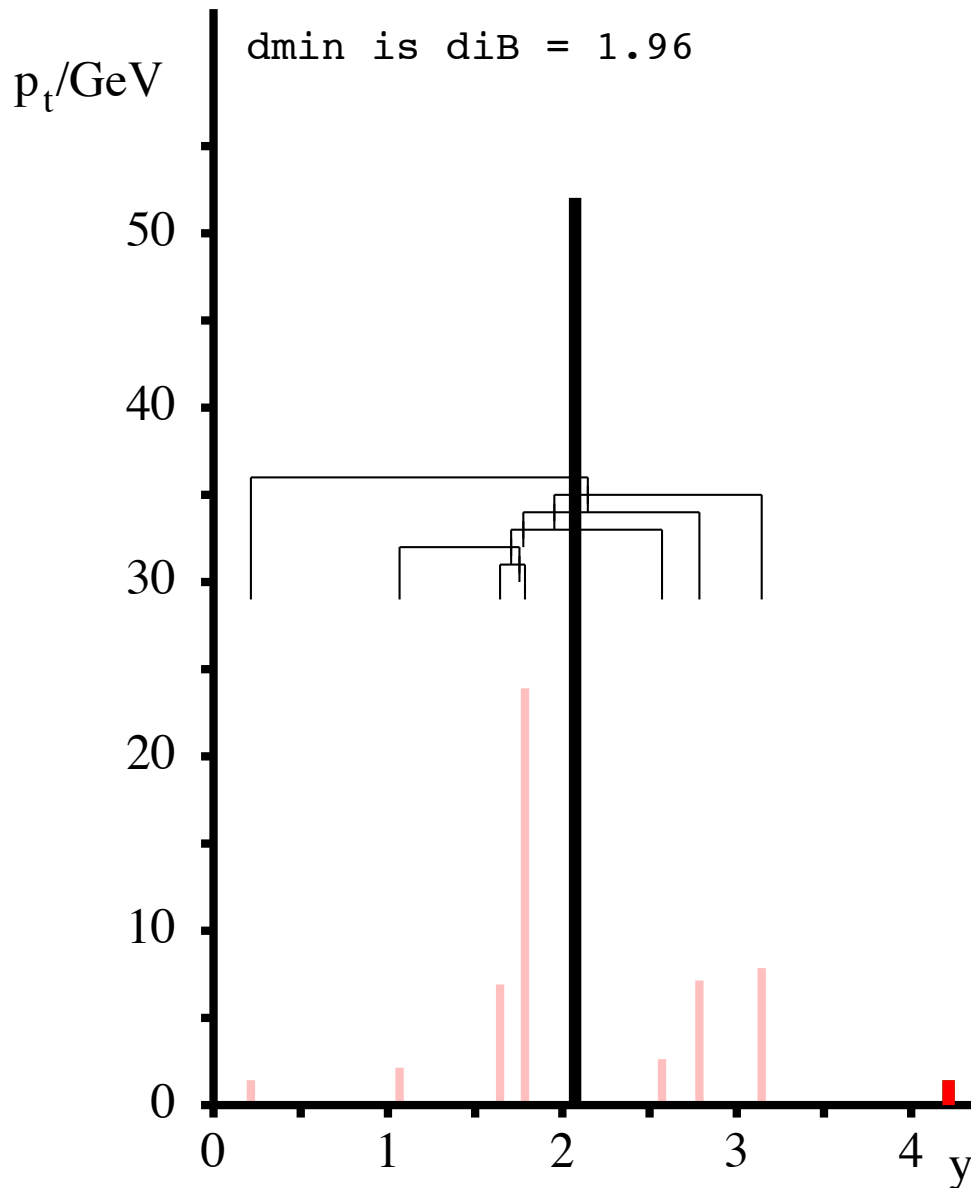
This is crucial for identifying the kinematic variables of the partons in the jet (e.g.  $z$ ).

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

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm

$d_{\min}$  is  $d_{iB} = 1.96$



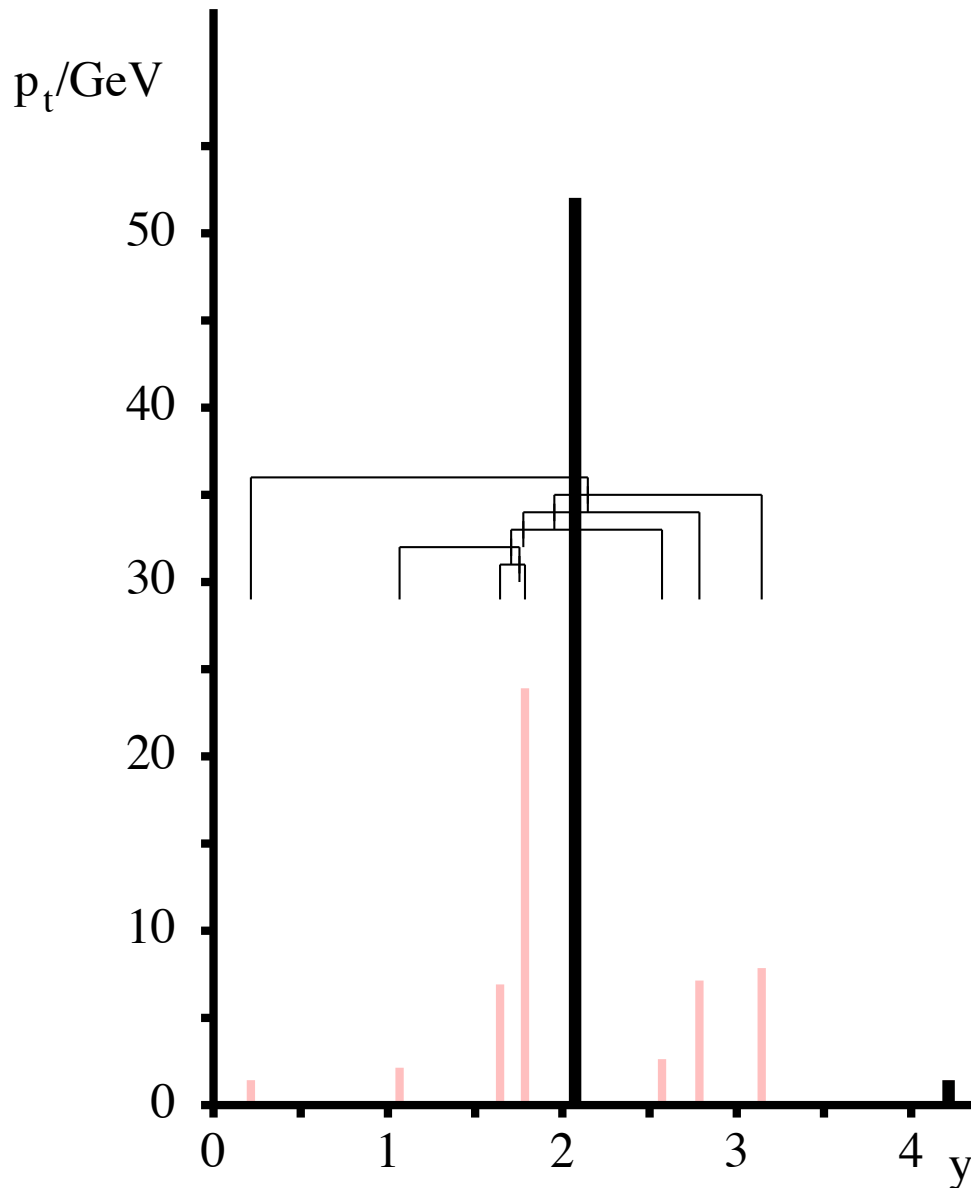
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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.*

# Identifying jet substructure: try out anti- $k_t$

## anti- $k_t$ algorithm



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

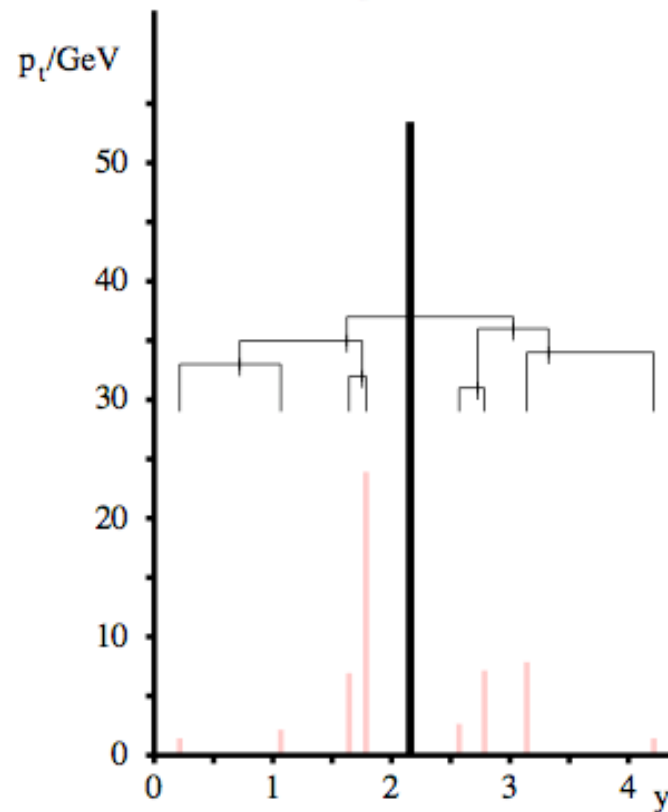
This is crucial for identifying the kinematic variables of the partons in the jet (e.g.  $z$ ).

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

$kt$

# Hierarchical substructure

## $k_t$ algorithm



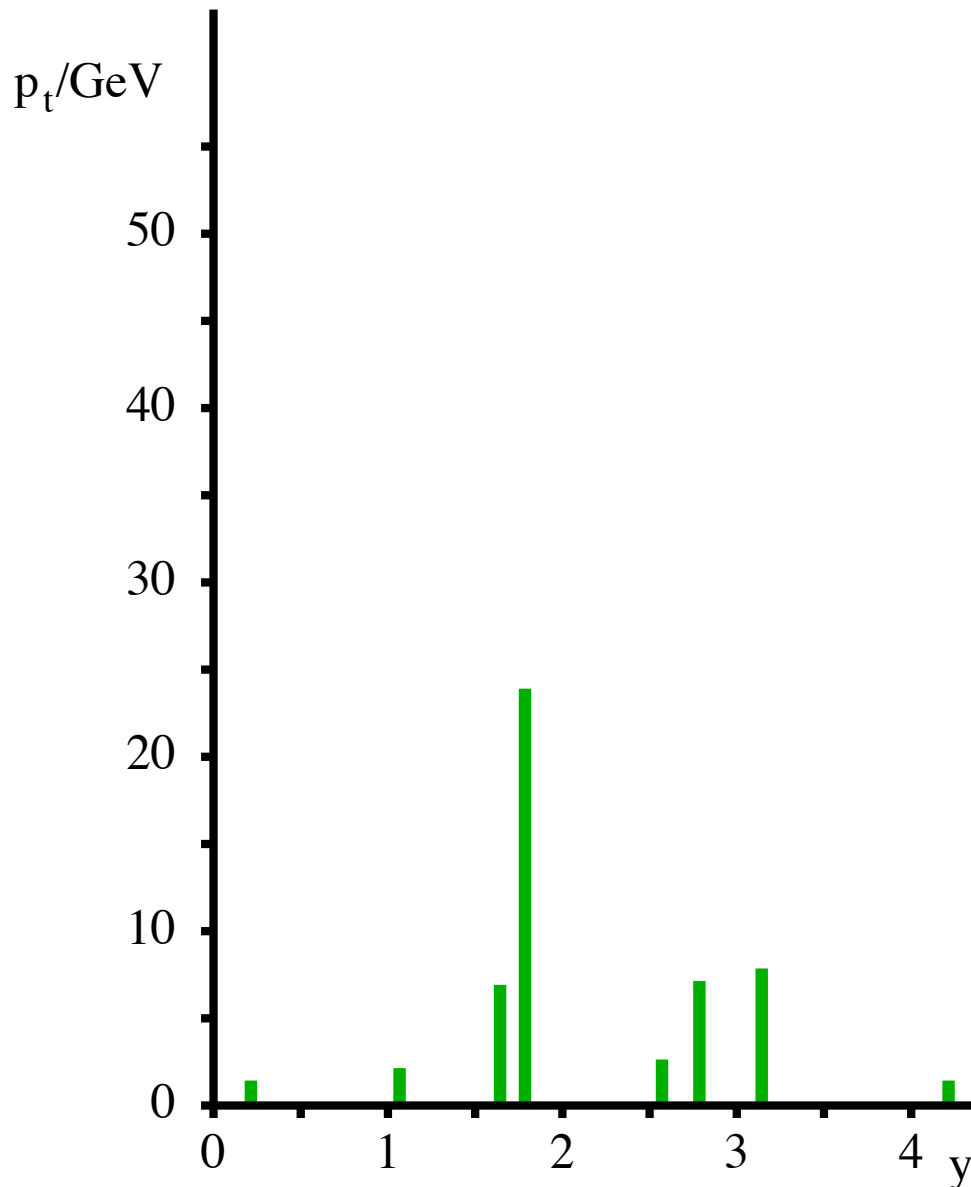
$k_t$  distance measure

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

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

# Identifying jet substructure: try out $k_t$

## $k_t$ algorithm



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

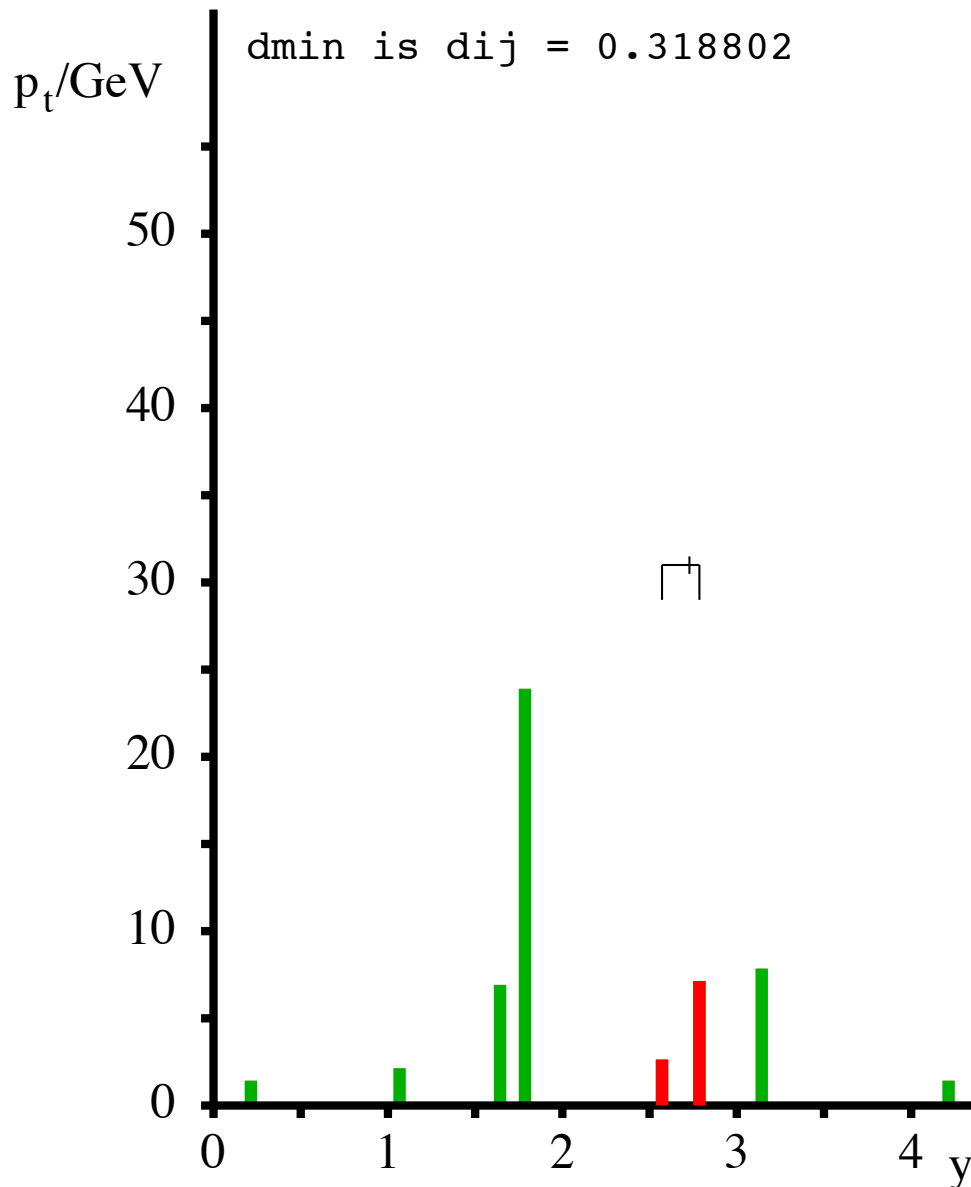
This is crucial for identifying the kinematic variables of the partons in the jet (e.g.  $z$ ).



# Identifying jet substructure: try out $k_t$

## $k_t$ algorithm

dmin is dij = 0.318802

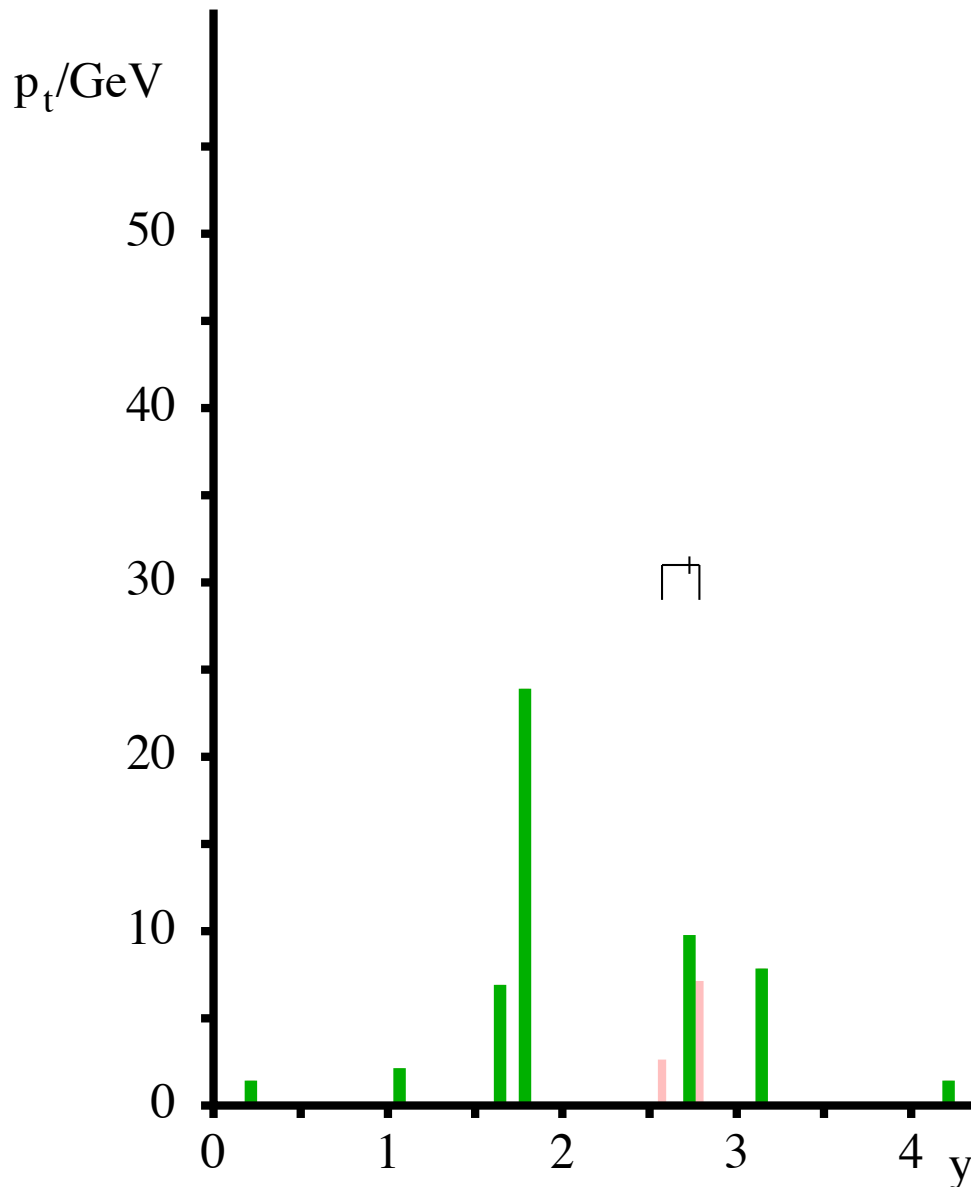


How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: try out $k_t$

## $k_t$ algorithm



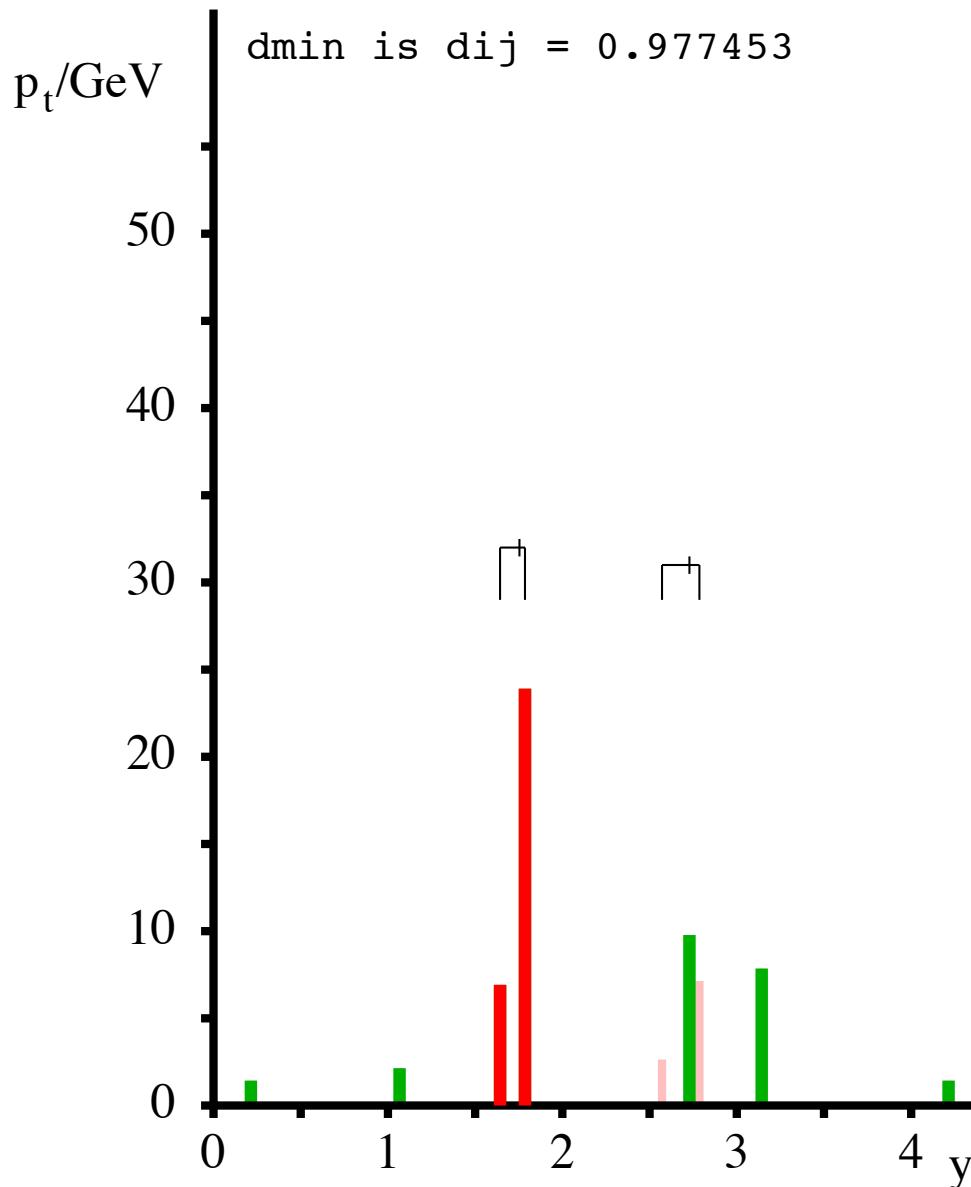
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: try out $k_t$

## $k_t$ algorithm

dmin is dij = 0.977453

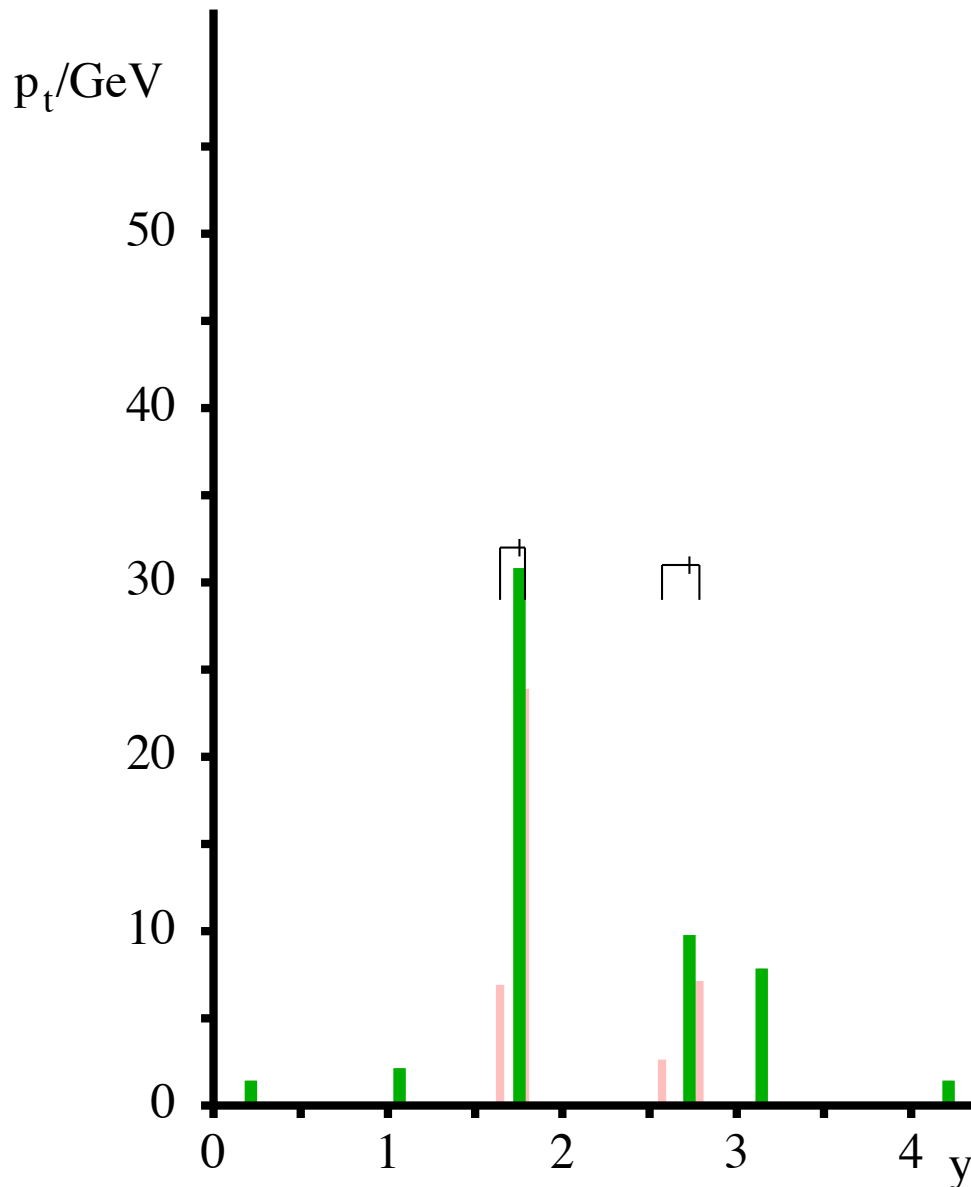


How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: try out $k_t$

## $k_t$ algorithm



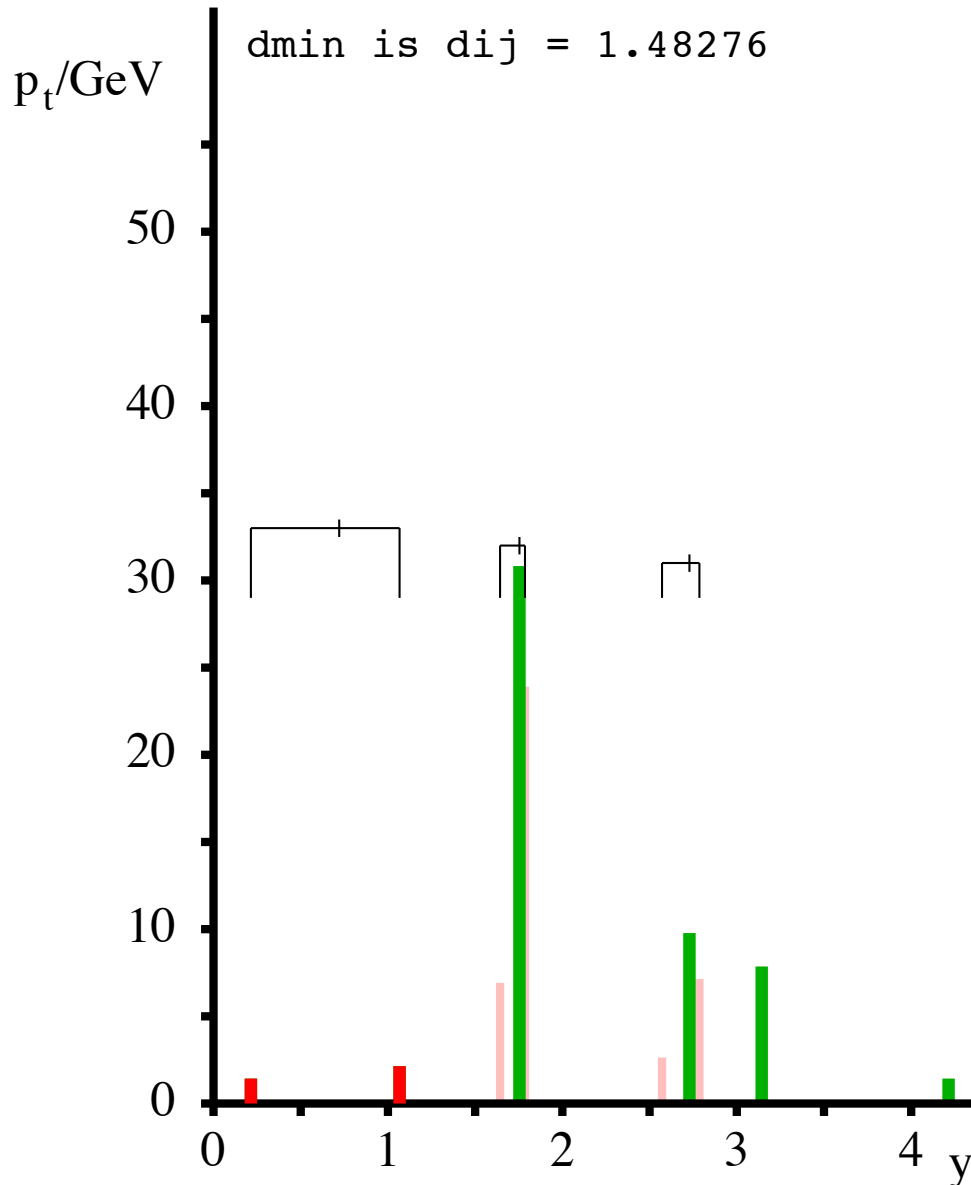
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: try out $k_t$

## $k_t$ algorithm

$d_{\min}$  is  $d_{ij} = 1.48276$



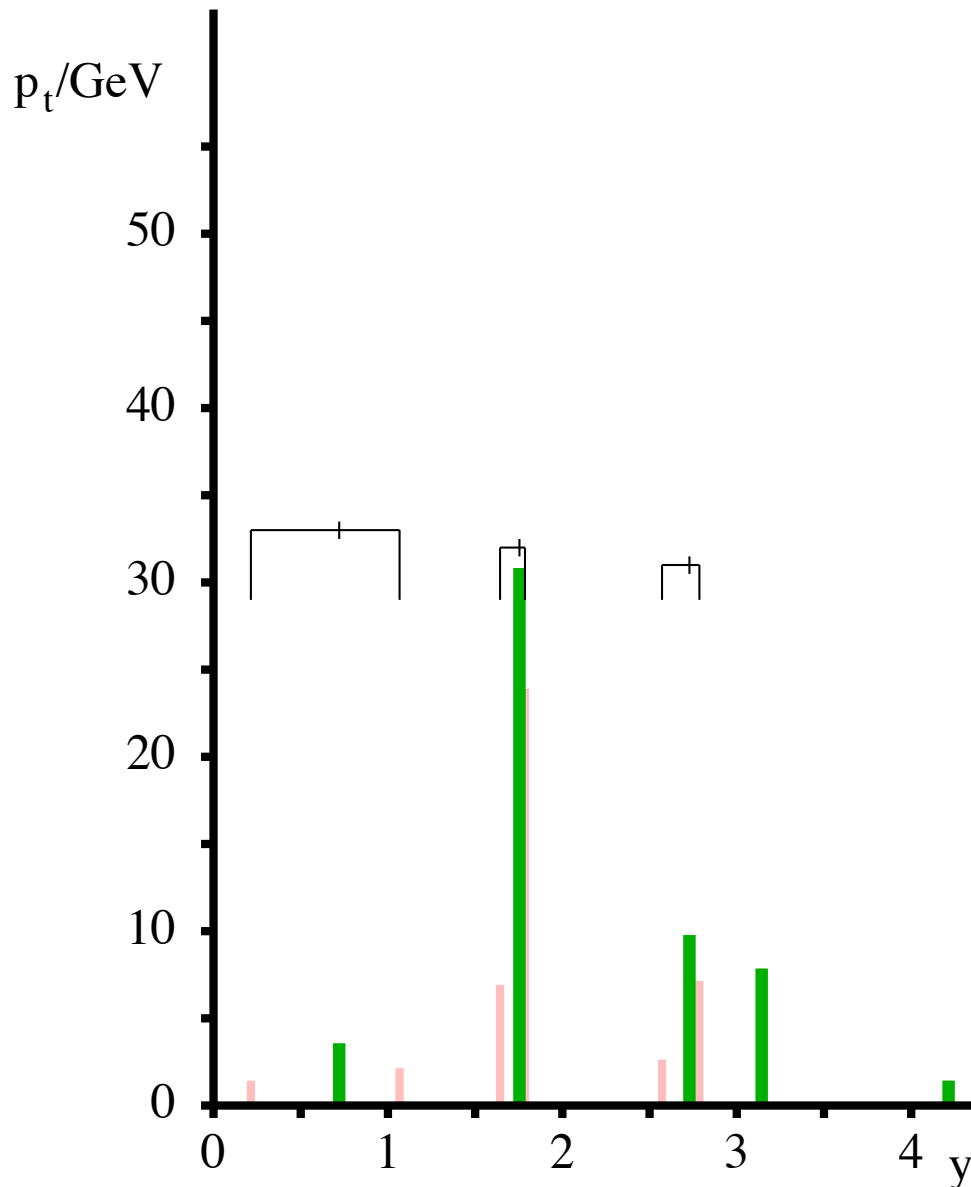
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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 $k_t$

## $k_t$ algorithm



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

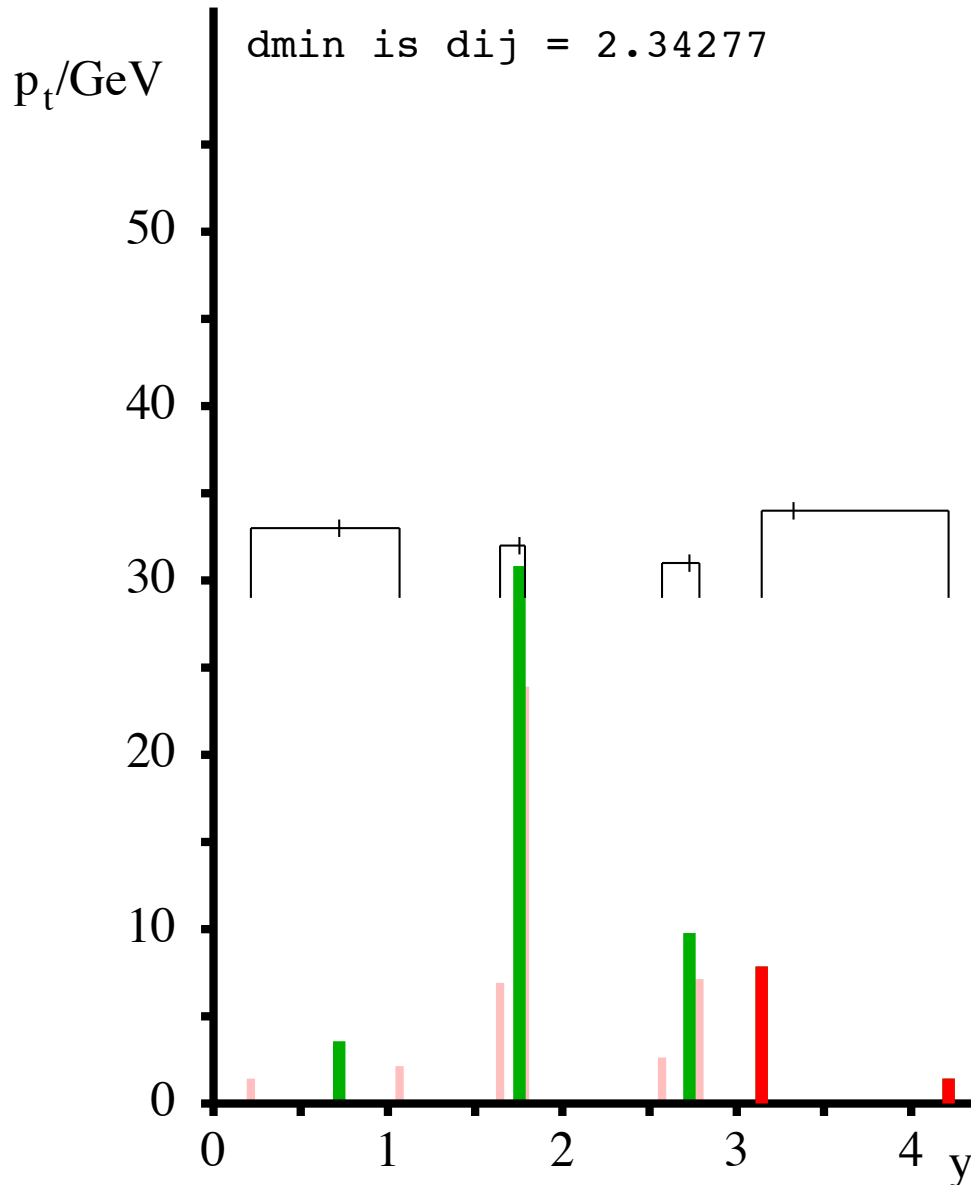
This is crucial for identifying the kinematic variables of the partons in the jet (e.g.  $z$ ).

$k_t$  clusters soft “junk” early on in the clustering

# Identifying jet substructure: try out $k_t$

## $k_t$ algorithm

dmin is dij = 2.34277



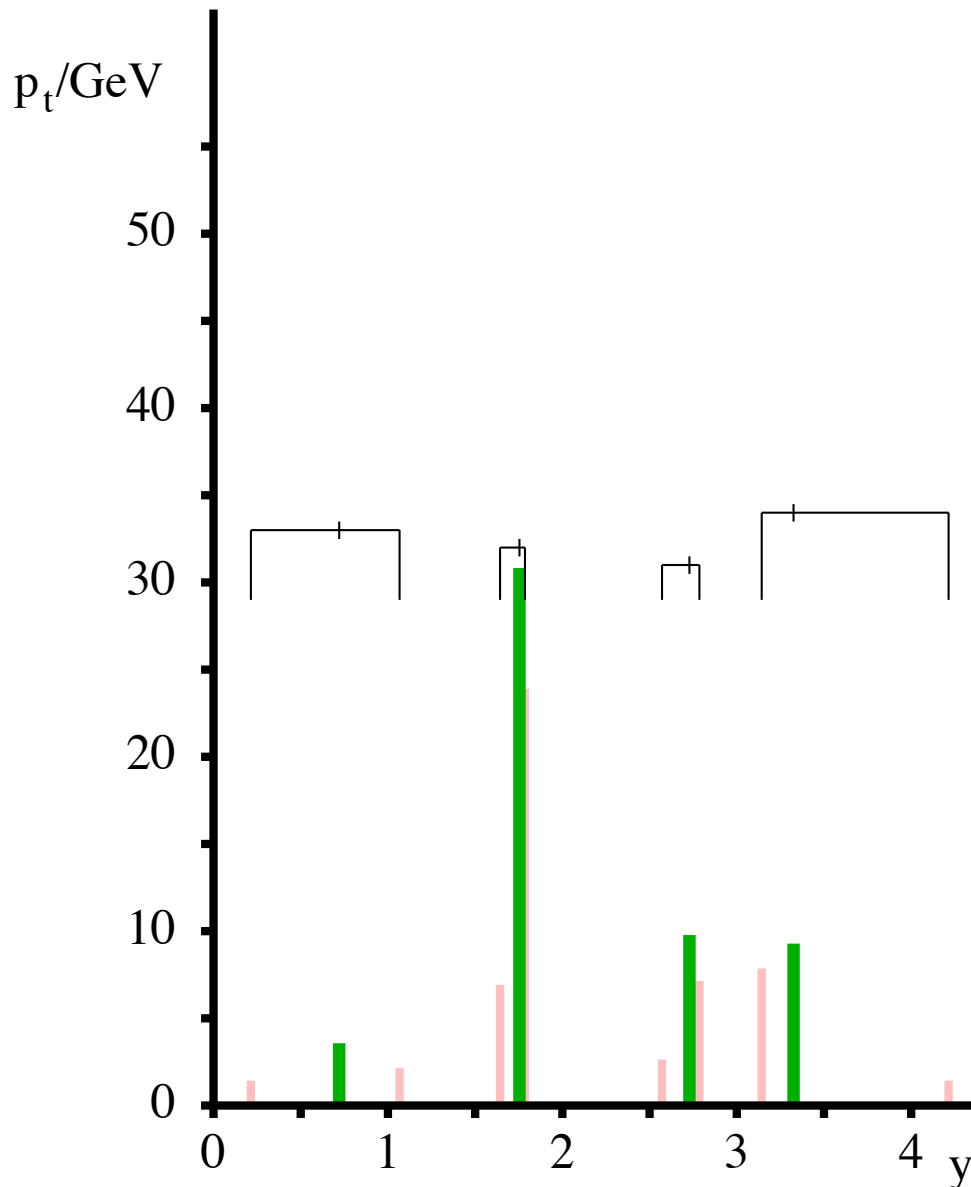
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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 $k_t$

## $k_t$ algorithm



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.  $z$ ).

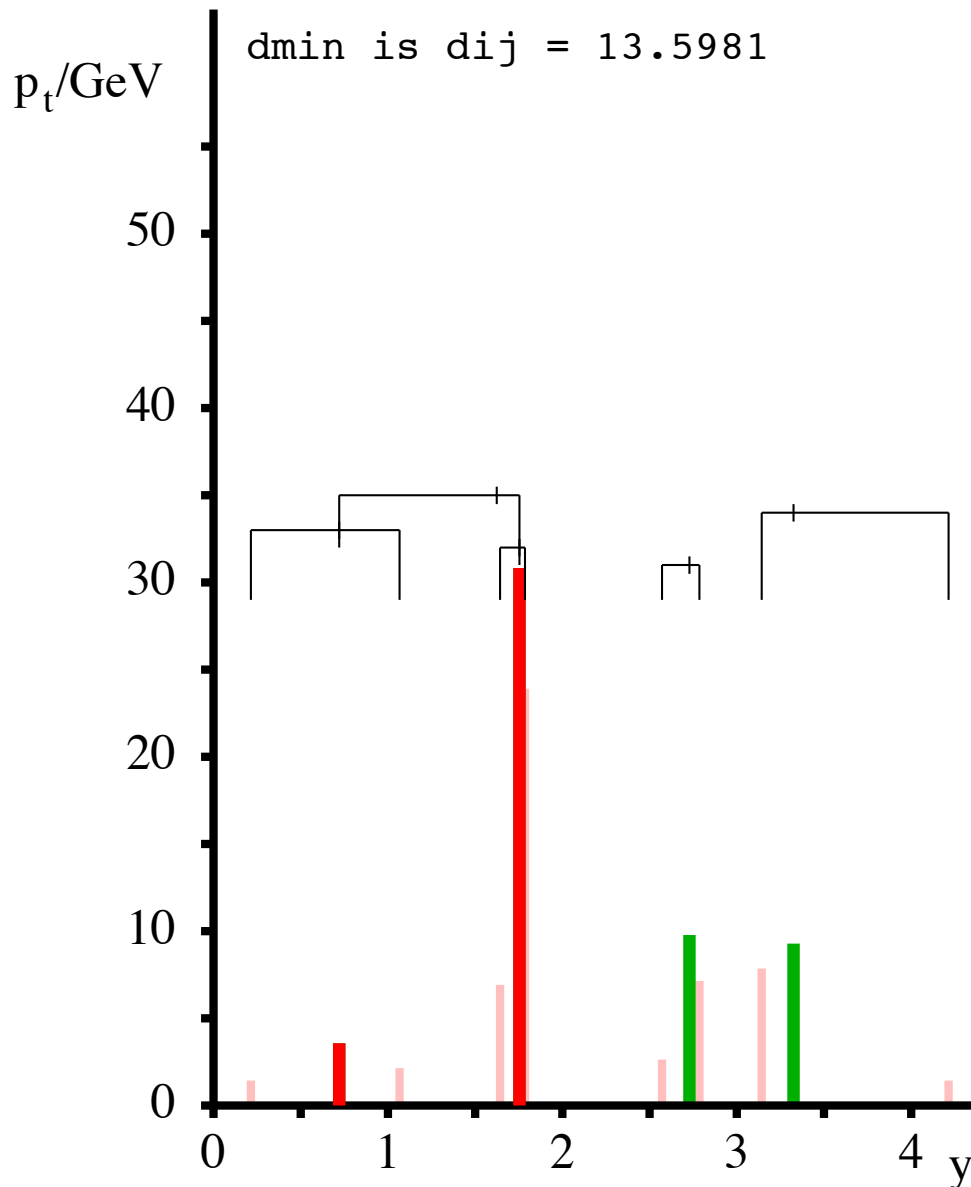
$k_t$  clusters soft “junk” early on in the clustering



# Identifying jet substructure: try out $k_t$

## $k_t$ algorithm

dmin is dij = 13.5981



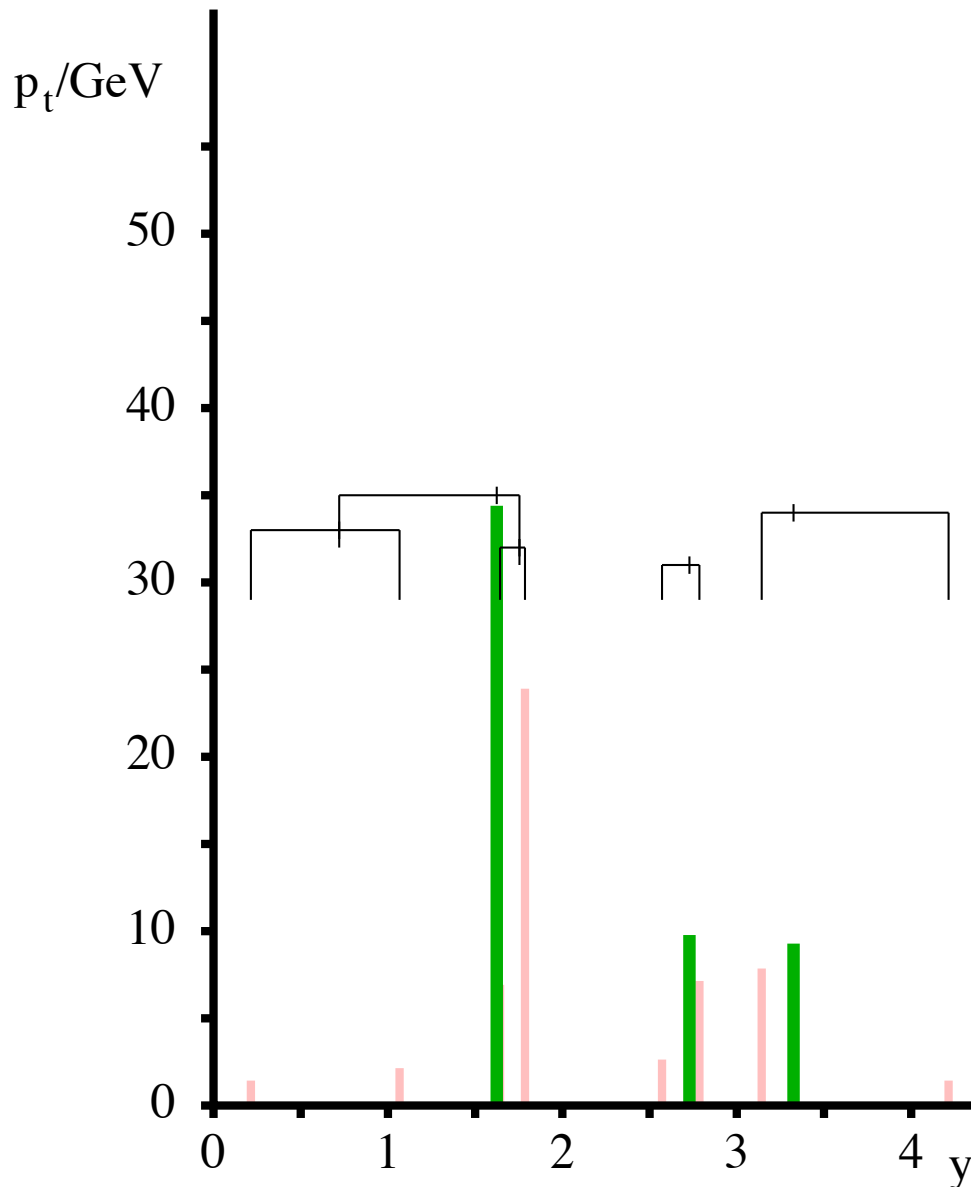
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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 $k_t$

## $k_t$ algorithm



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

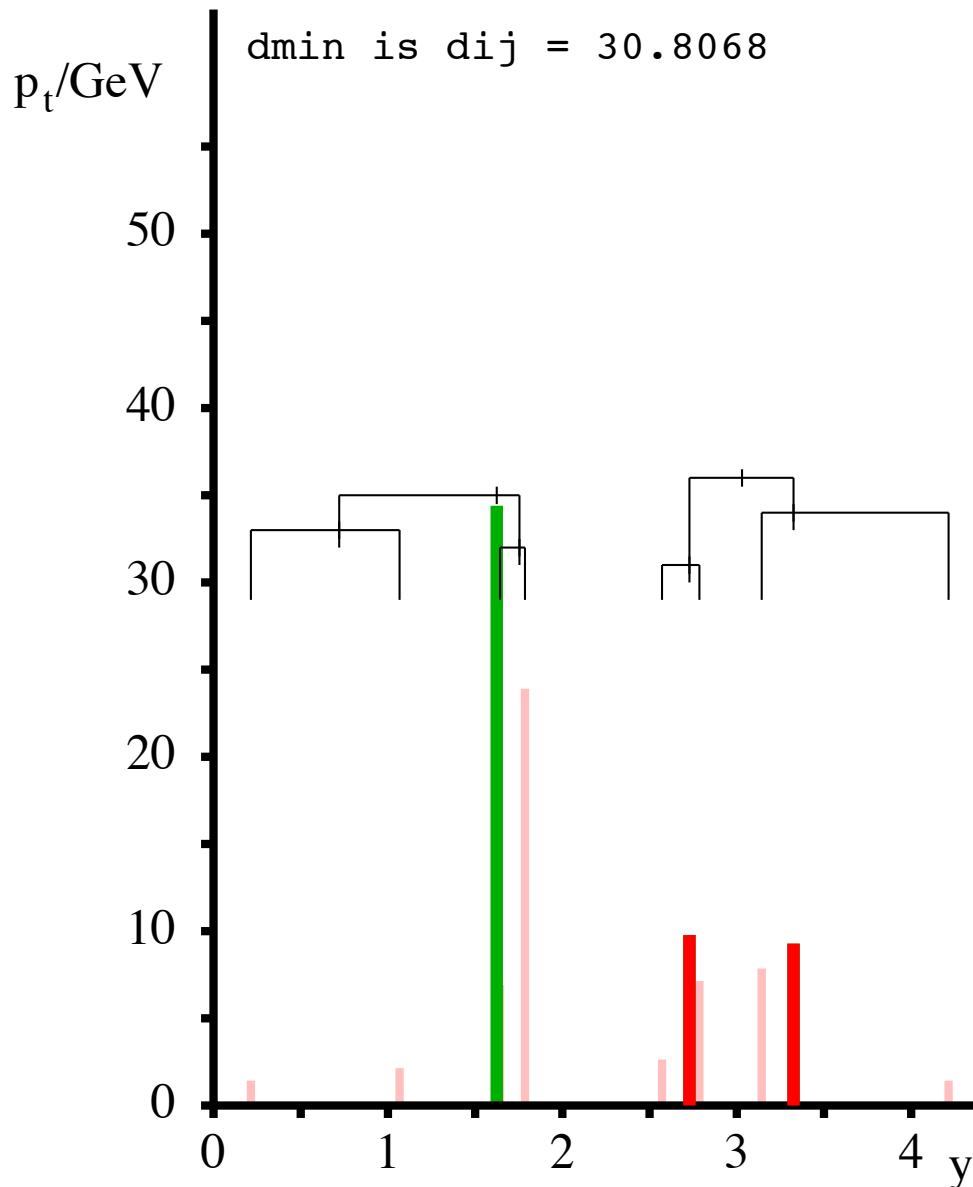
This is crucial for identifying the kinematic variables of the partons in the jet (e.g.  $z$ ).

$k_t$  clusters soft “junk” early on in the clustering

# Identifying jet substructure: try out $k_t$

## $k_t$ algorithm

$d_{\min}$  is  $d_{ij} = 30.8068$



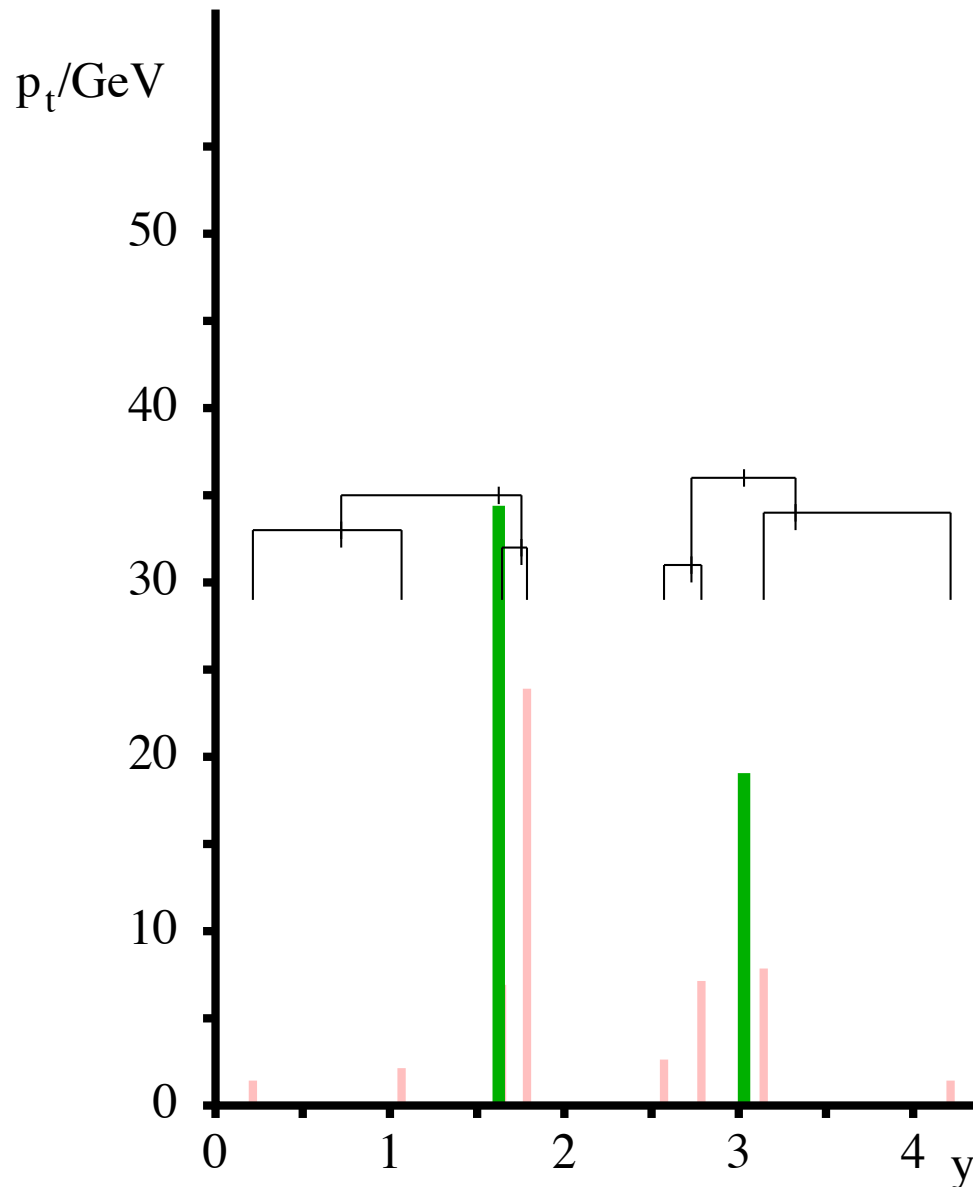
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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 $k_t$

## $k_t$ algorithm



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

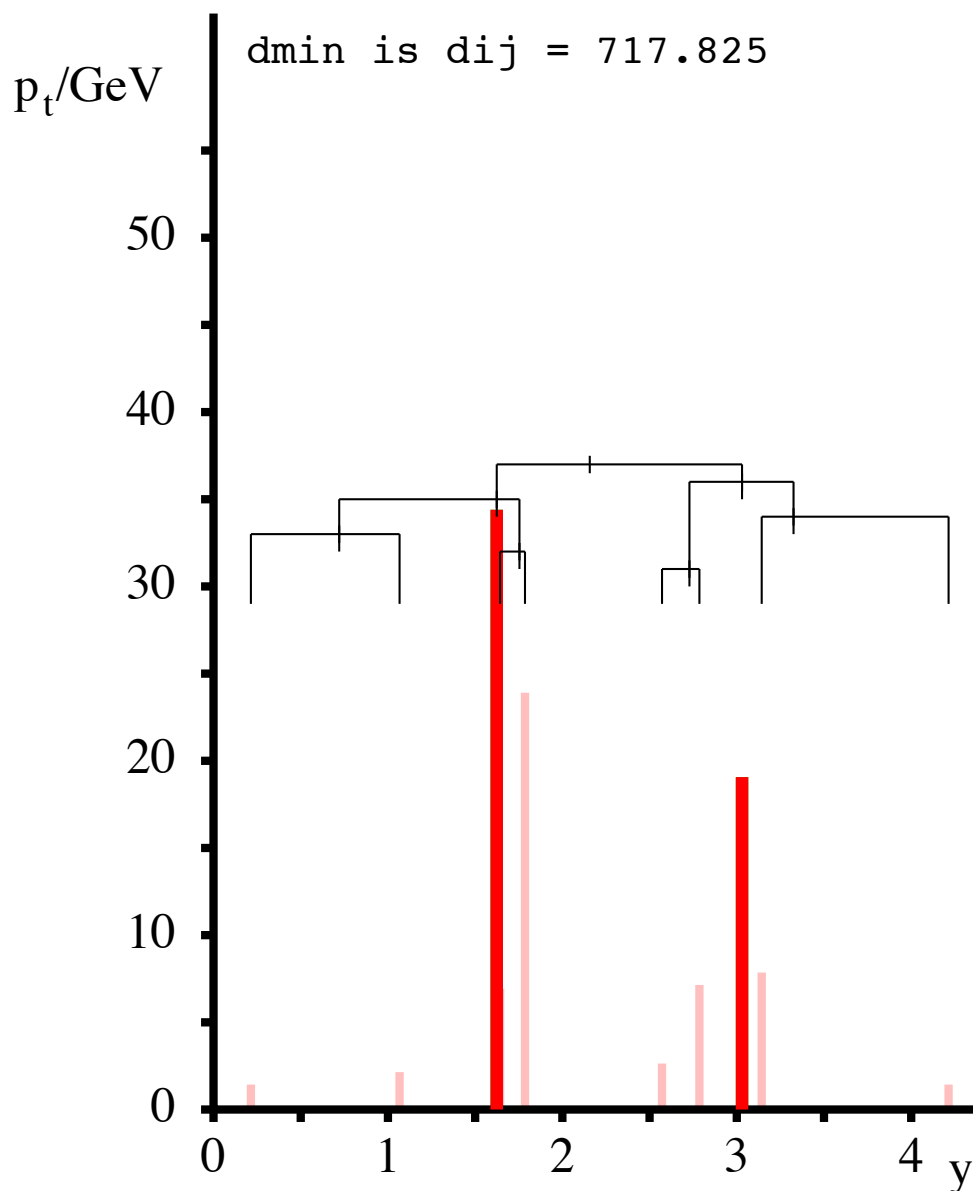
This is crucial for identifying the kinematic variables of the partons in the jet (e.g.  $z$ ).

$k_t$  clusters soft “junk” early on in the clustering

# Identifying jet substructure: try out $k_t$

## $k_t$ algorithm

dmin is dij = 717.825



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

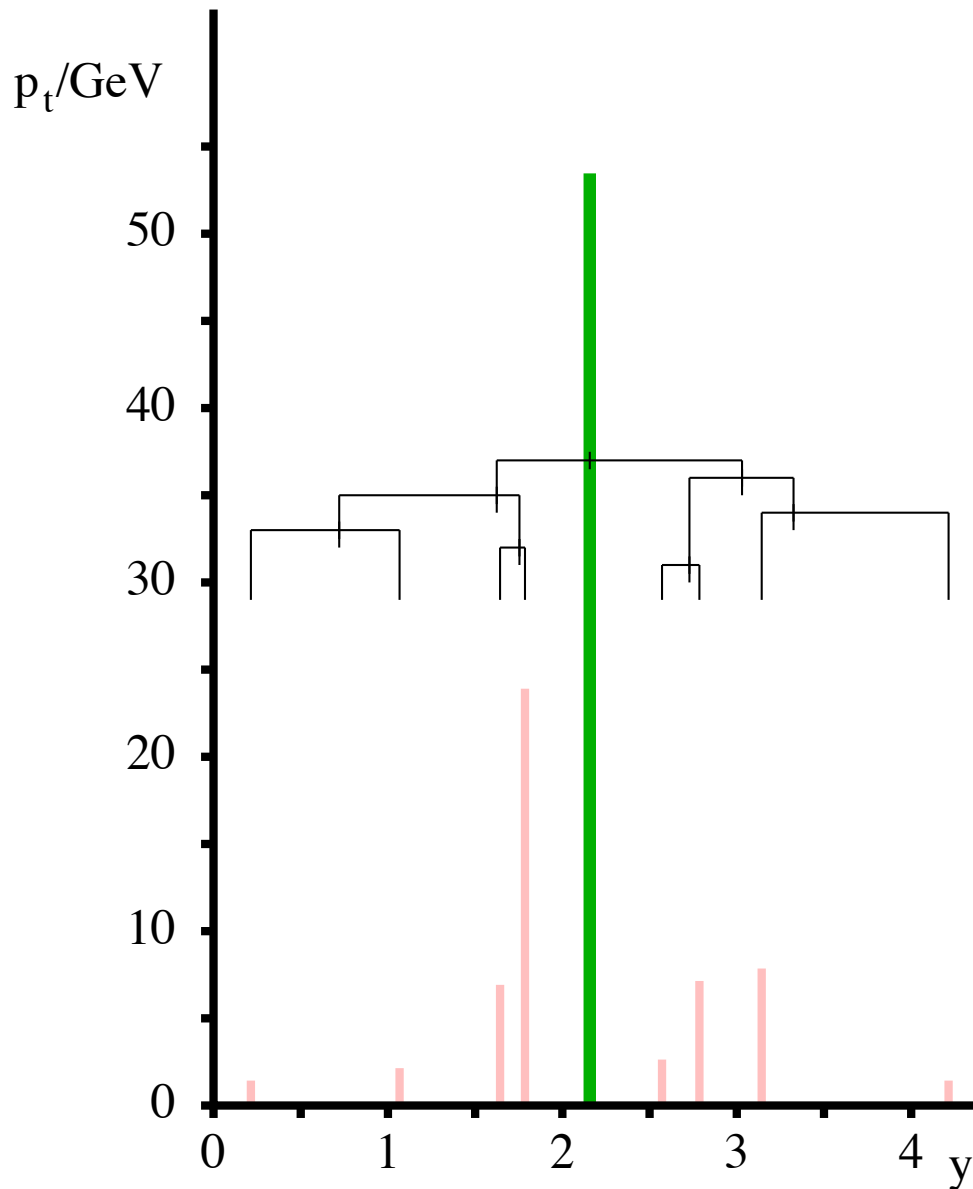
This is crucial for identifying the kinematic variables of the partons in the jet (e.g.  $z$ ).

$k_t$  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 $k_t$

## $k_t$ algorithm



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.  $z$ ).

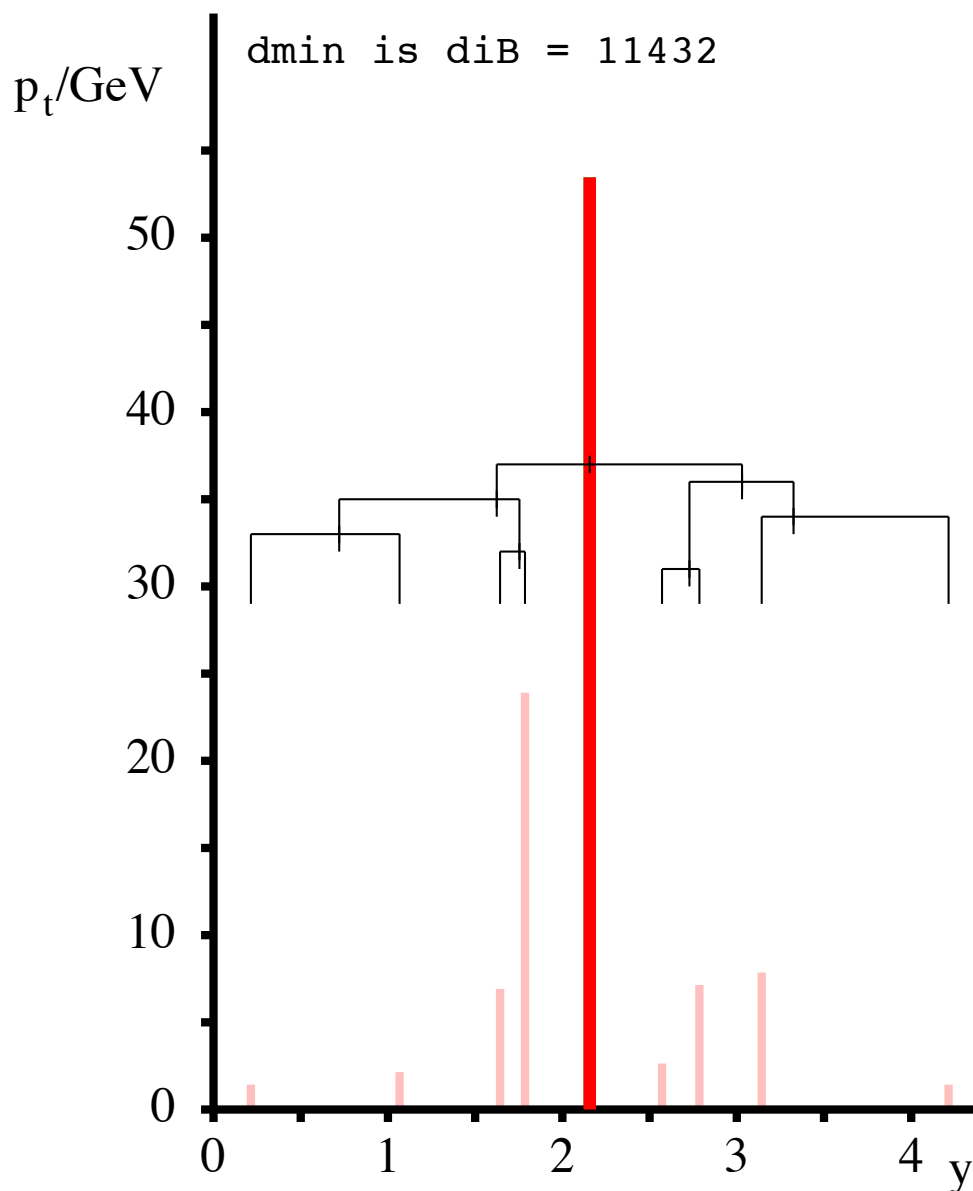
$k_t$  clusters soft “junk” early on in the clustering

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

# Identifying jet substructure: try out $k_t$

## $k_t$ algorithm

dmin is diB = 11432



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

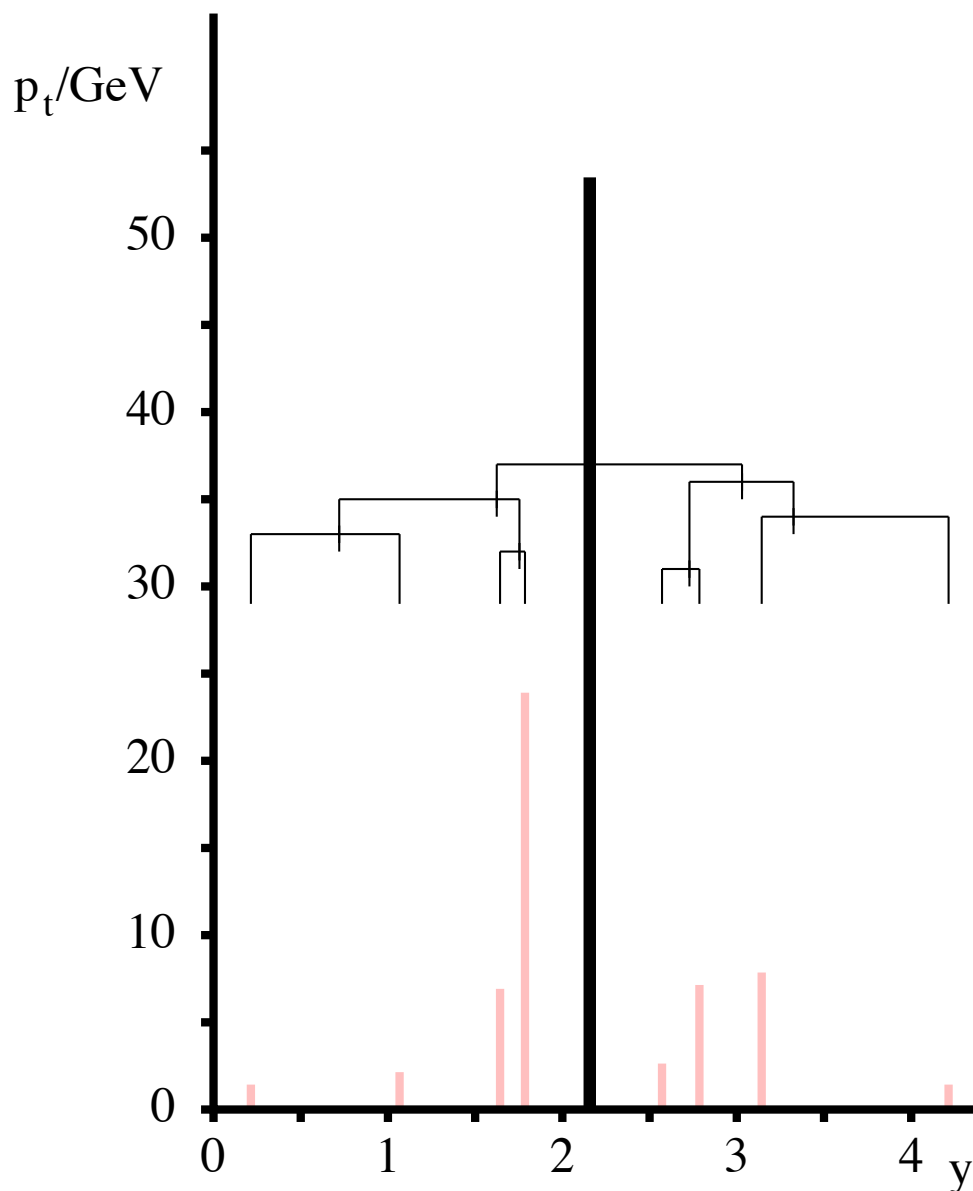
This is crucial for identifying the kinematic variables of the partons in the jet (e.g.  $z$ ).

$k_t$  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 $k_t$

## $k_t$ algorithm



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

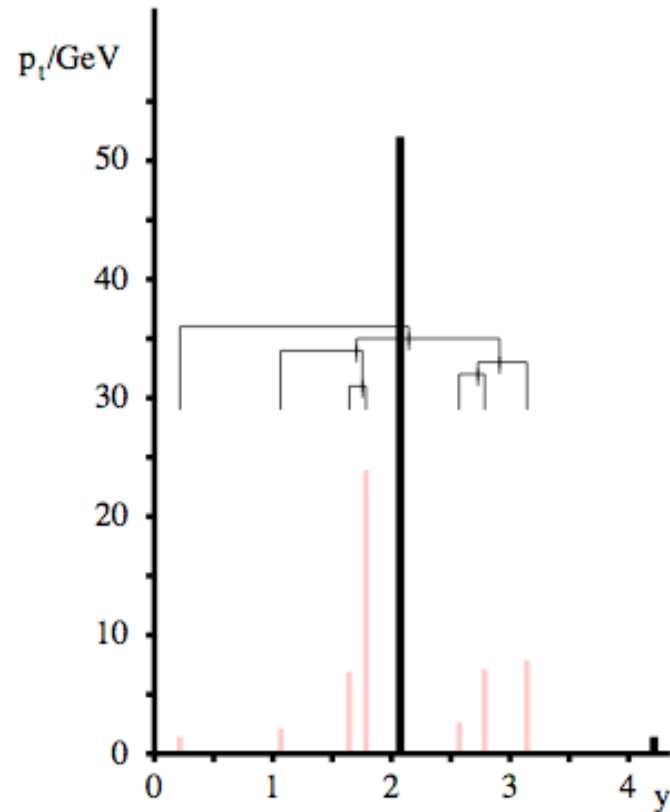
Butterworth, Cox & Forshaw '02



## Cambridge/Aachen

# Hierarchical substructure

## Cambridge/Aachen



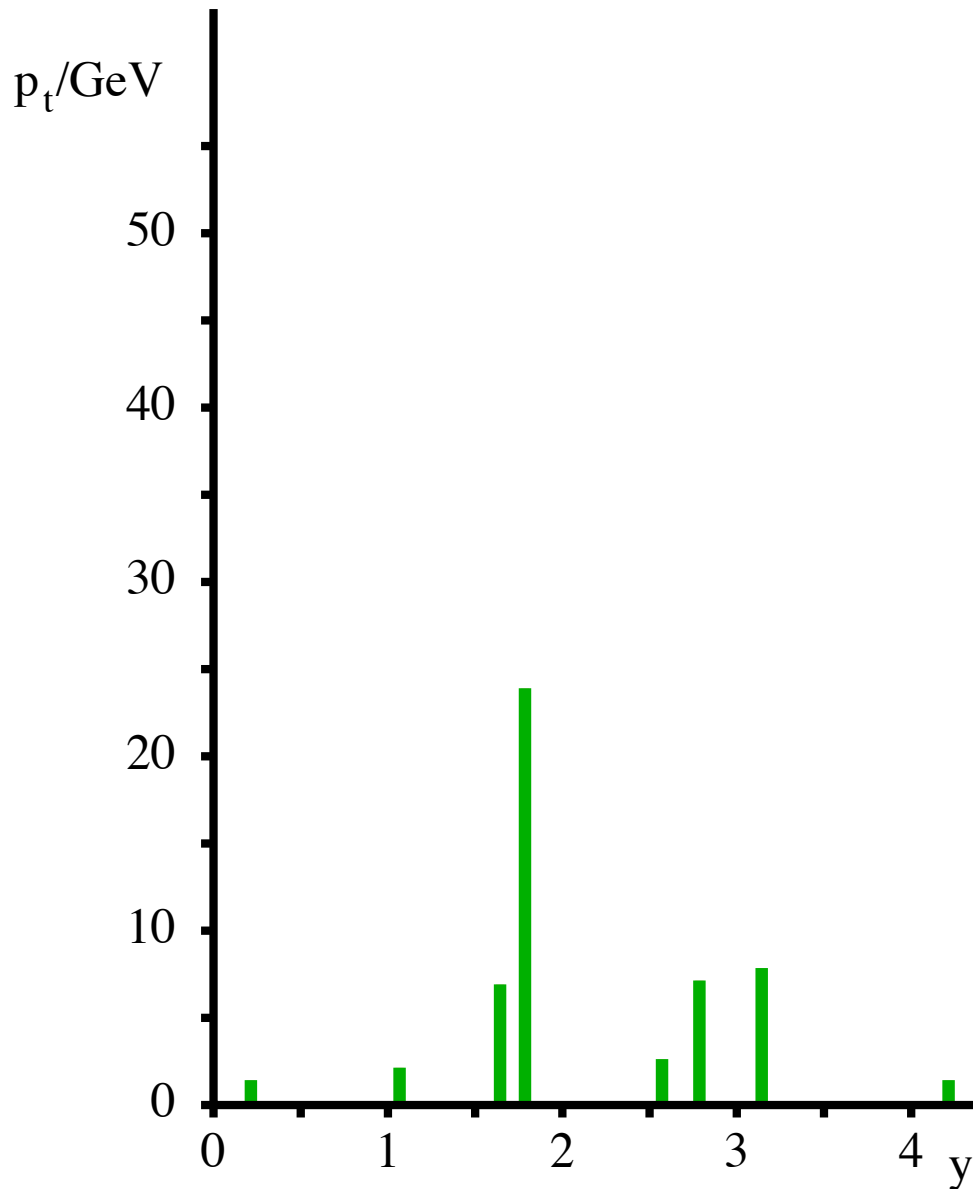
C/A distance measure

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

Cluster by merging  
the **closest** particles

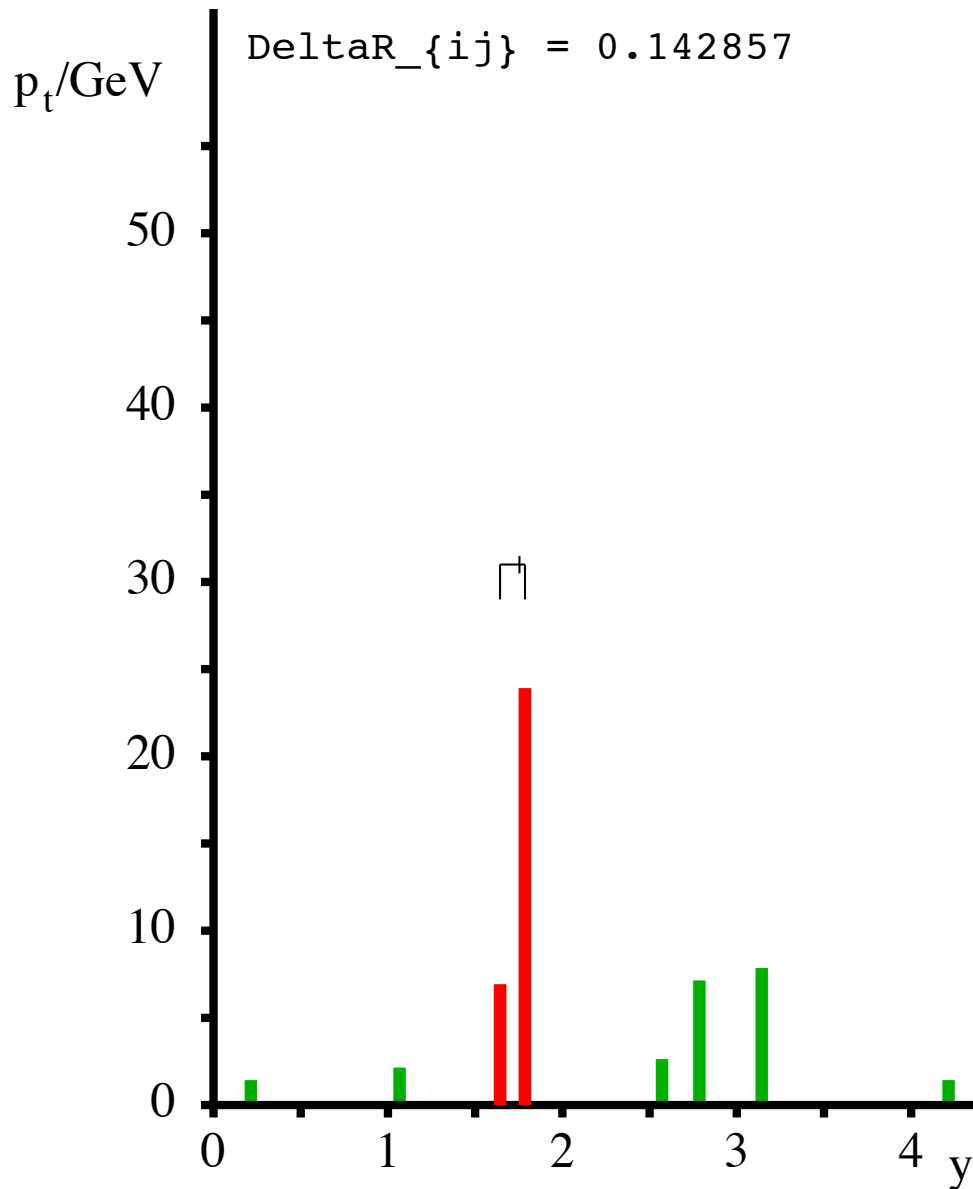
# Identifying jet substructure: Cam/Aachen

## Cambridge/Aachen algorithm



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

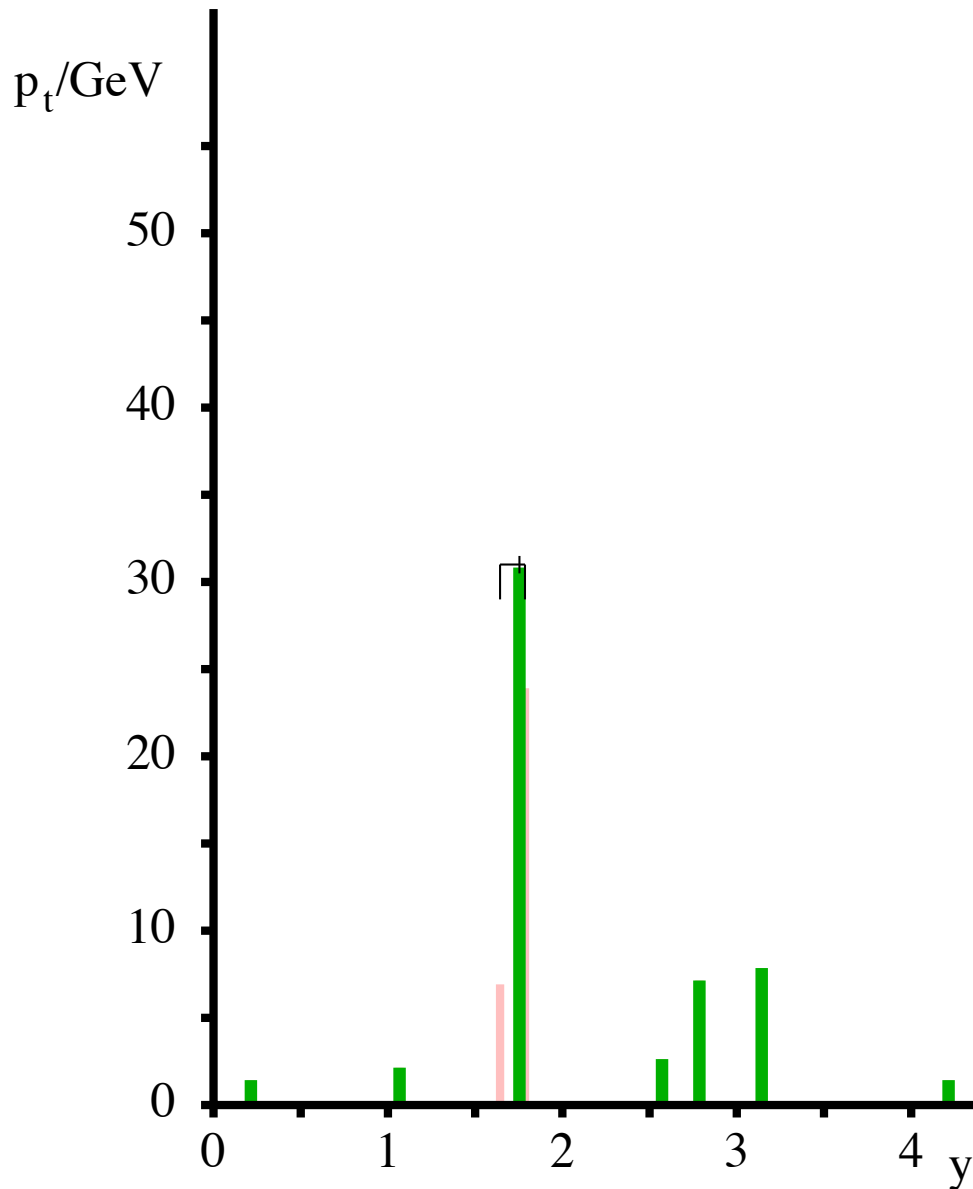
## Cambridge/Aachen algorithm



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

# Identifying jet substructure: Cam/Aachen

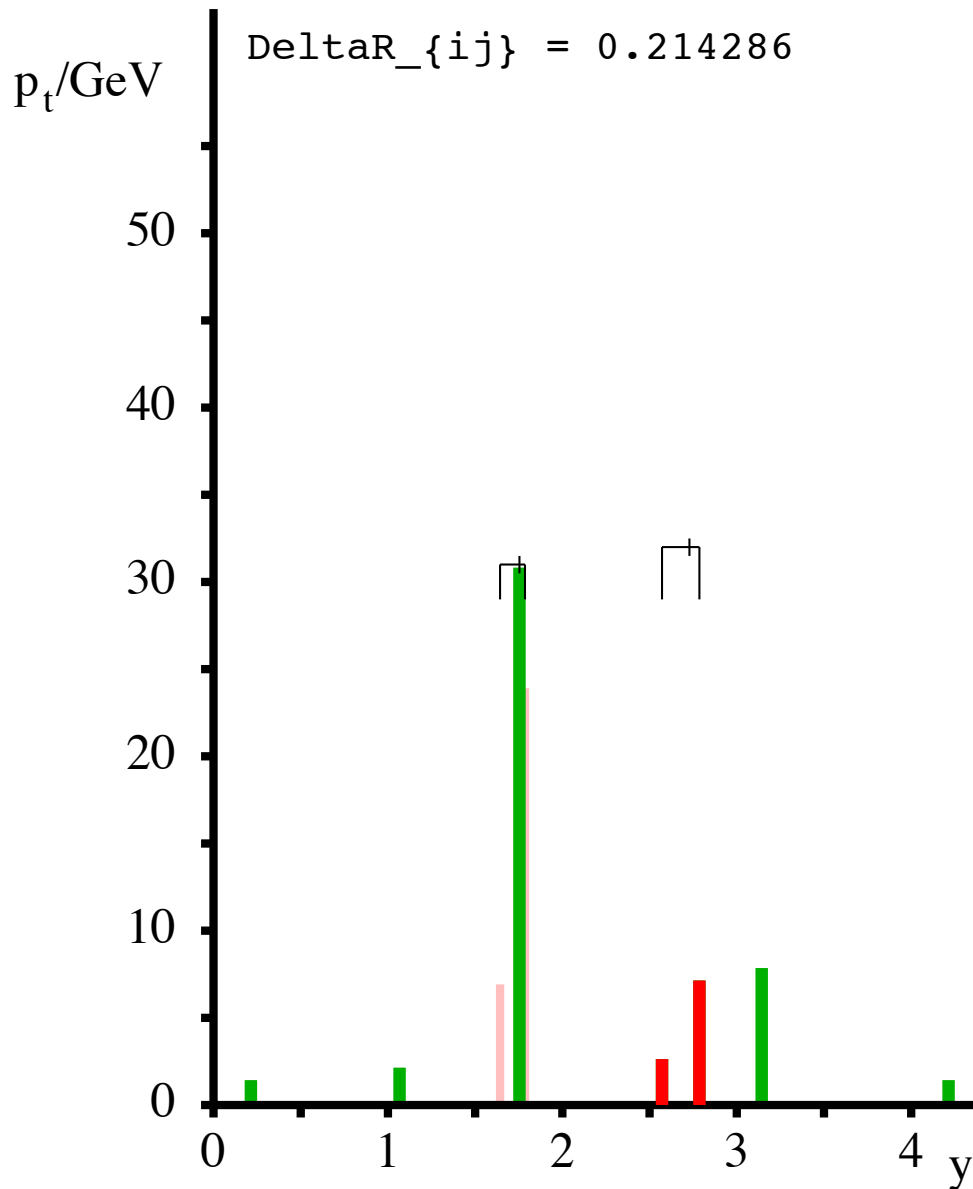
## Cambridge/Aachen algorithm



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

# Identifying jet substructure: Cam/Aachen

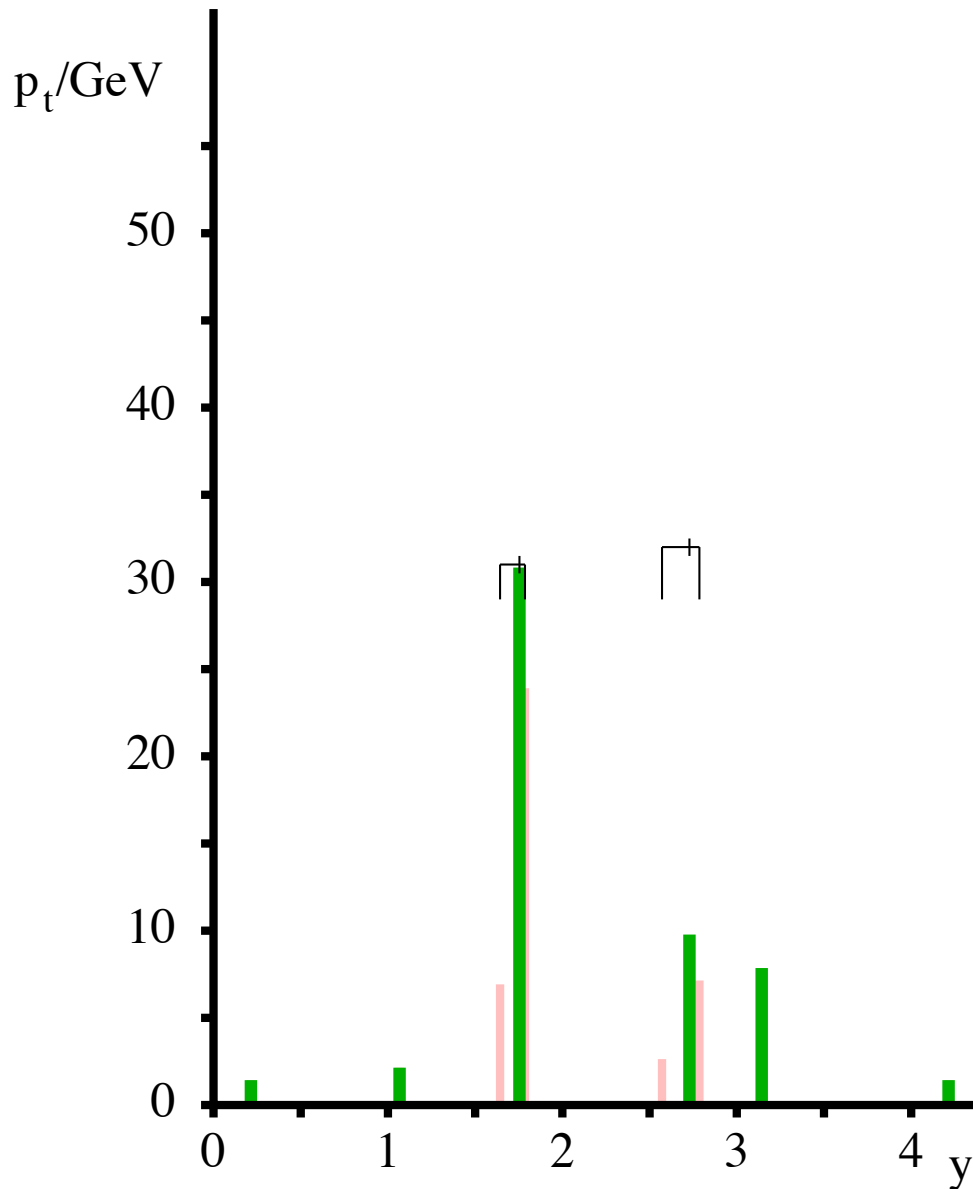
## Cambridge/Aachen algorithm



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

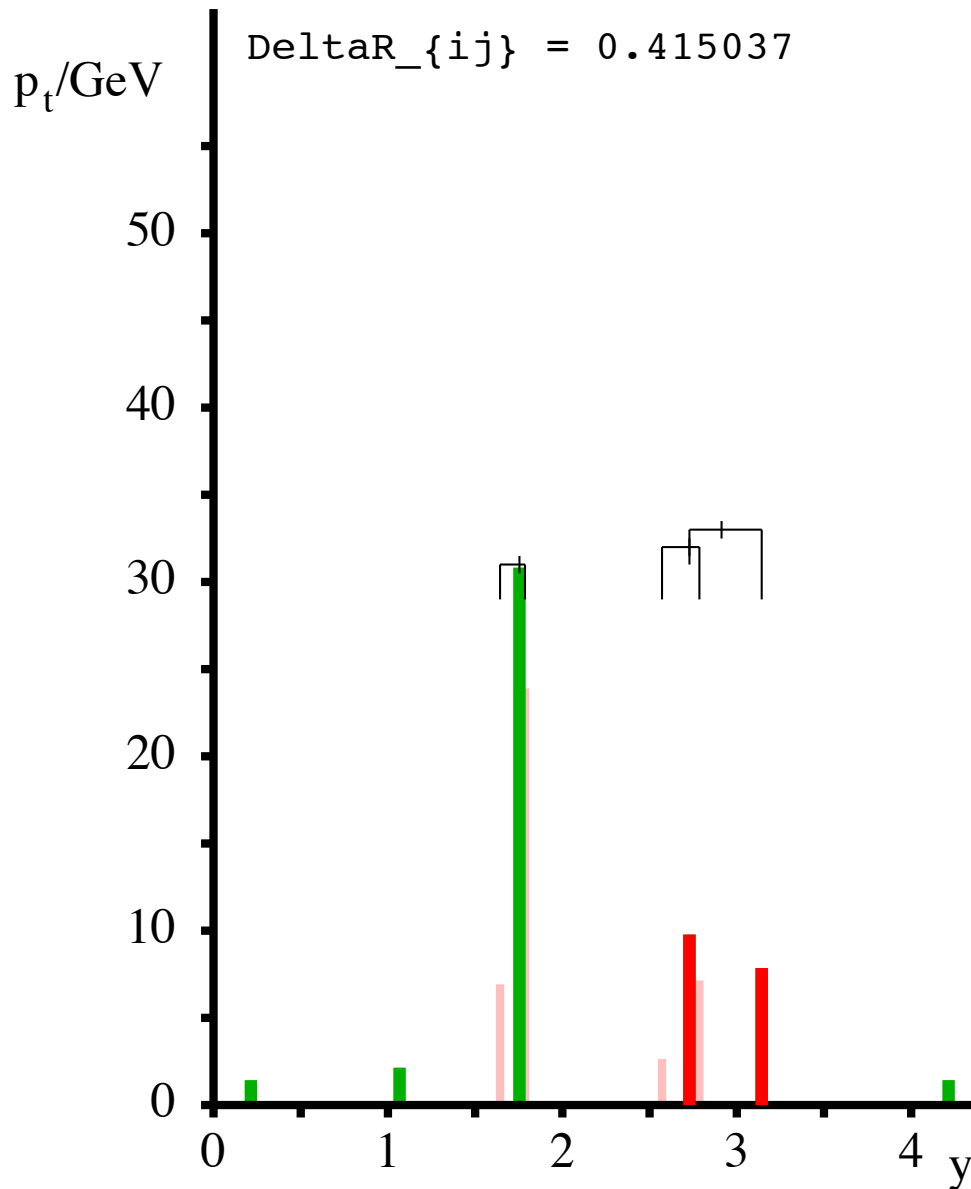
# Identifying jet substructure: Cam/Aachen

## Cambridge/Aachen algorithm



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

## Cambridge/Aachen algorithm

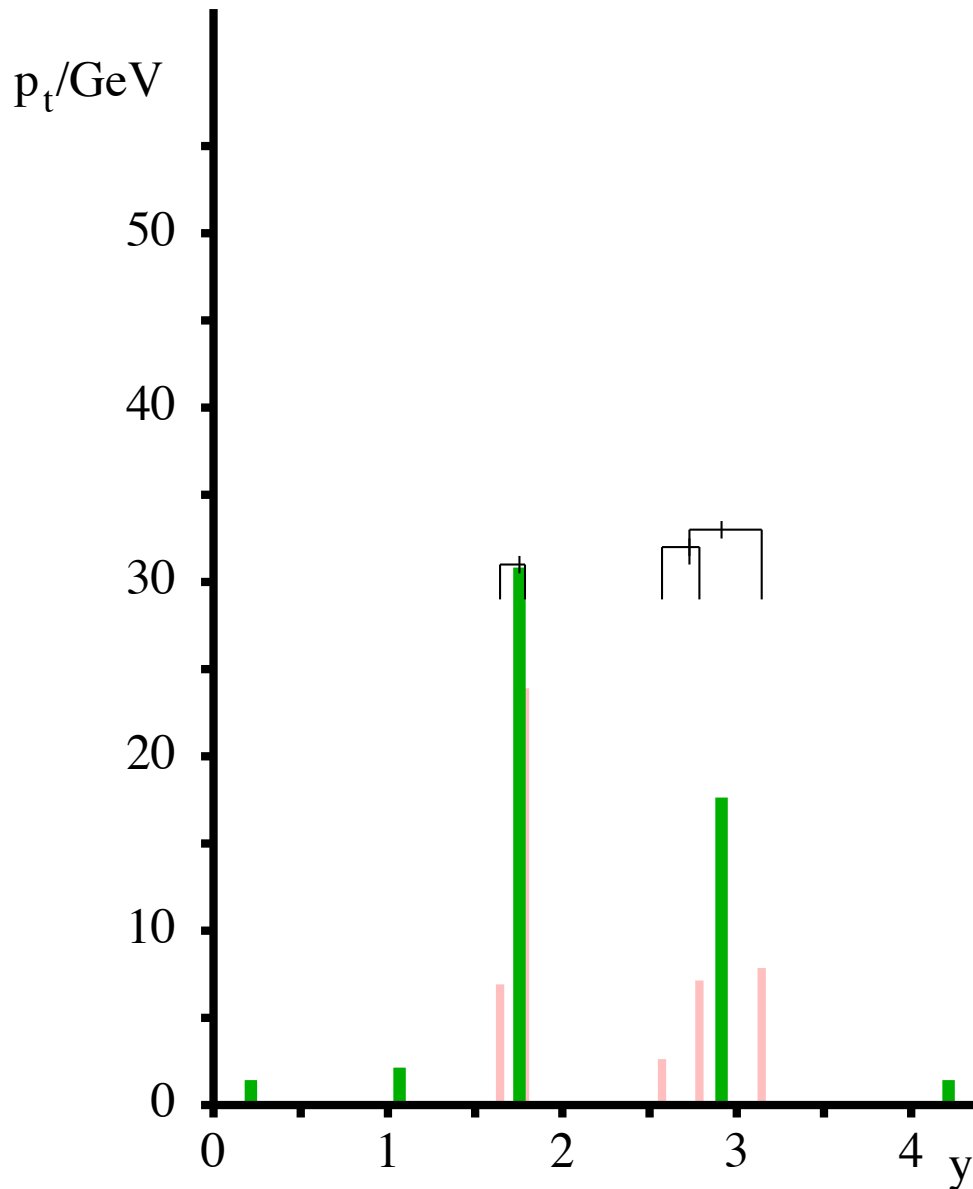


How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?



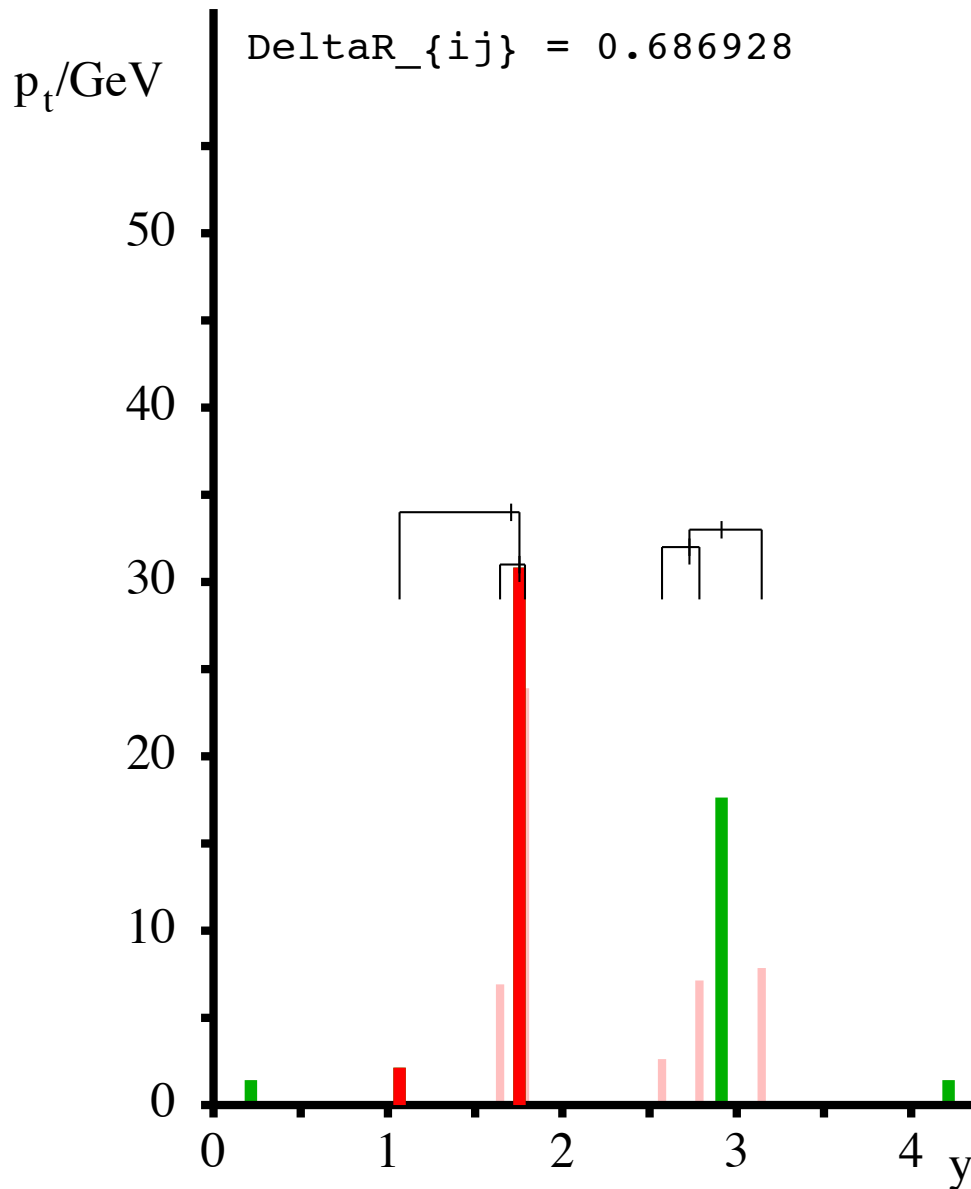
# Identifying jet substructure: Cam/Aachen

## Cambridge/Aachen algorithm



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

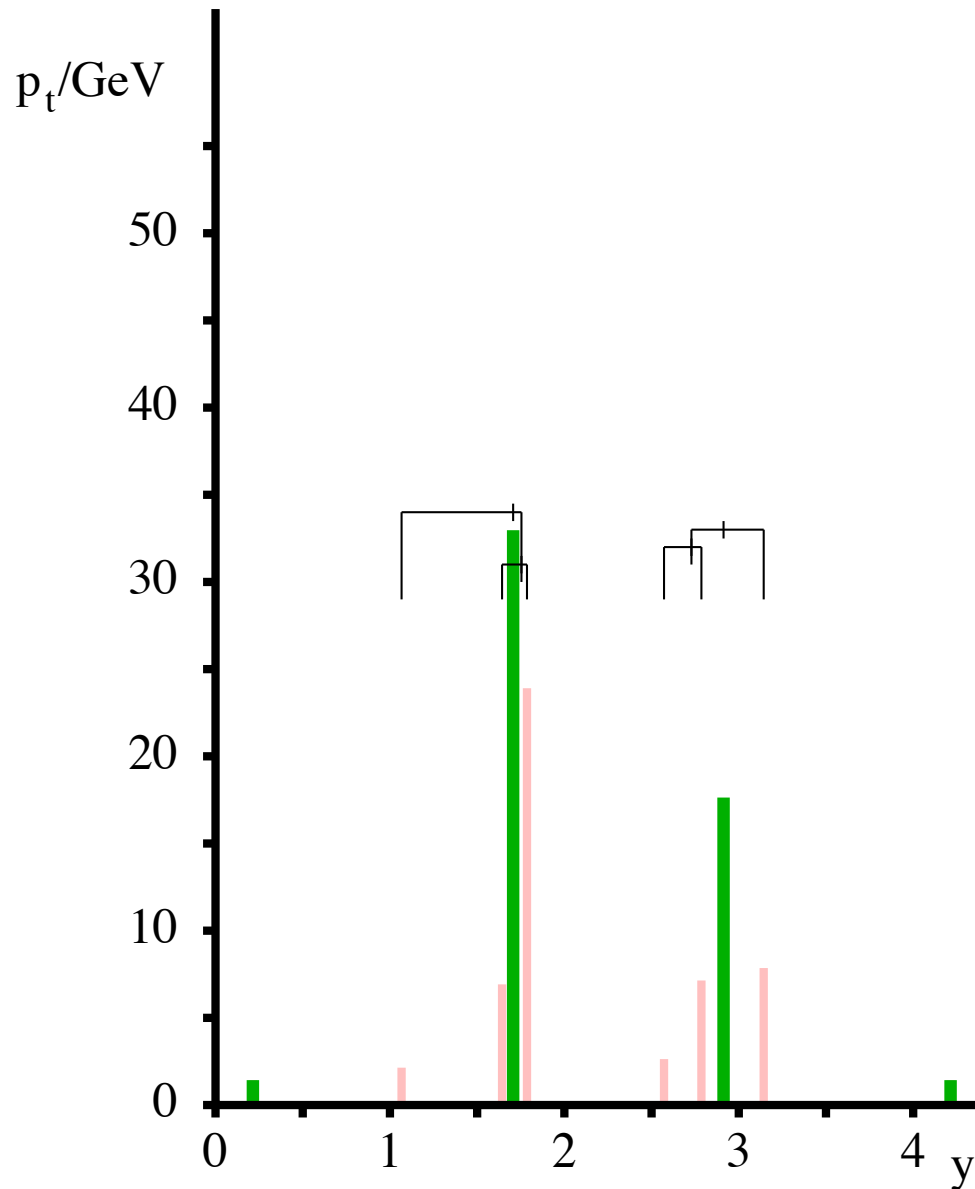
## Cambridge/Aachen algorithm



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

# Identifying jet substructure: Cam/Aachen

## Cambridge/Aachen algorithm

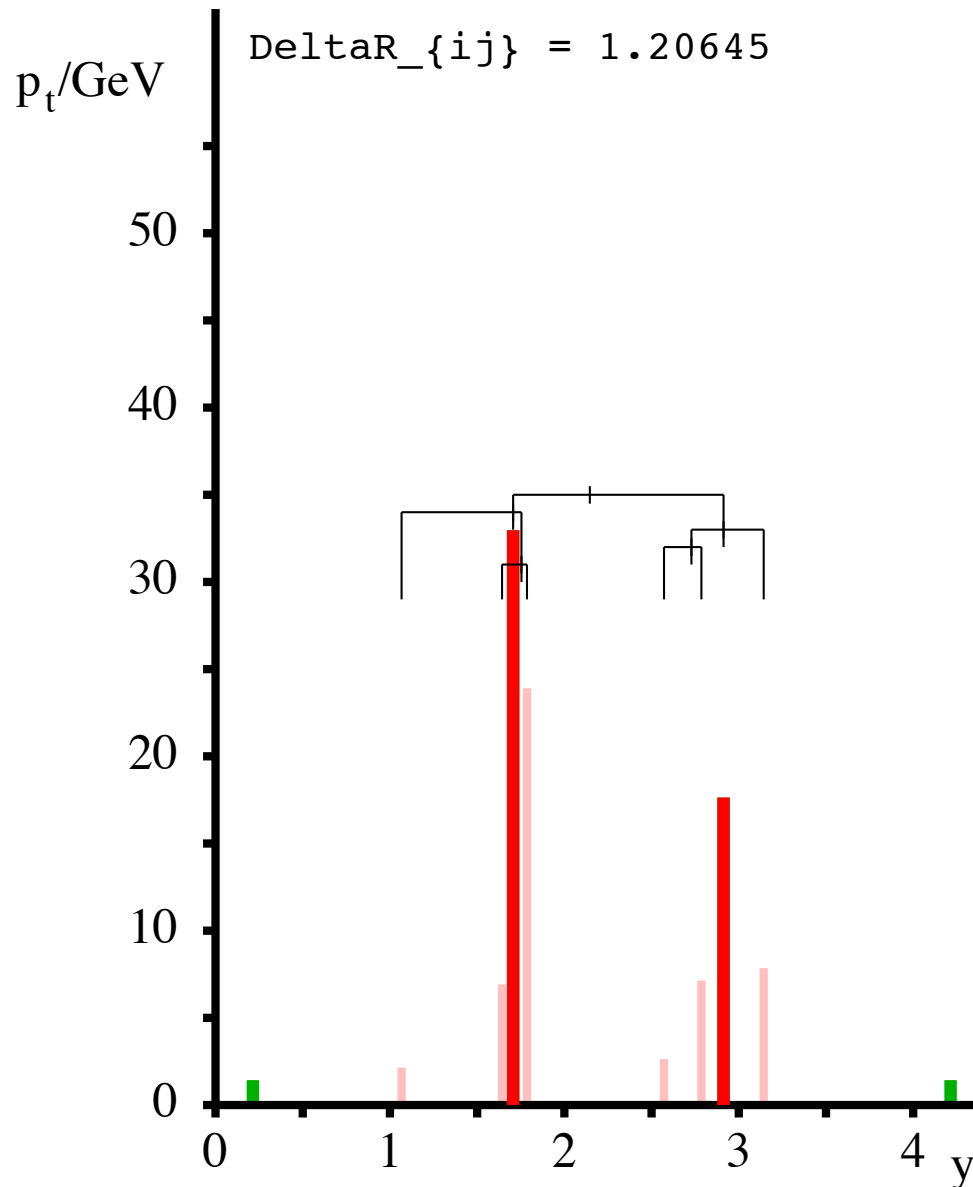


How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination

# Identifying jet substructure: Cam/Aachen

## Cambridge/Aachen algorithm

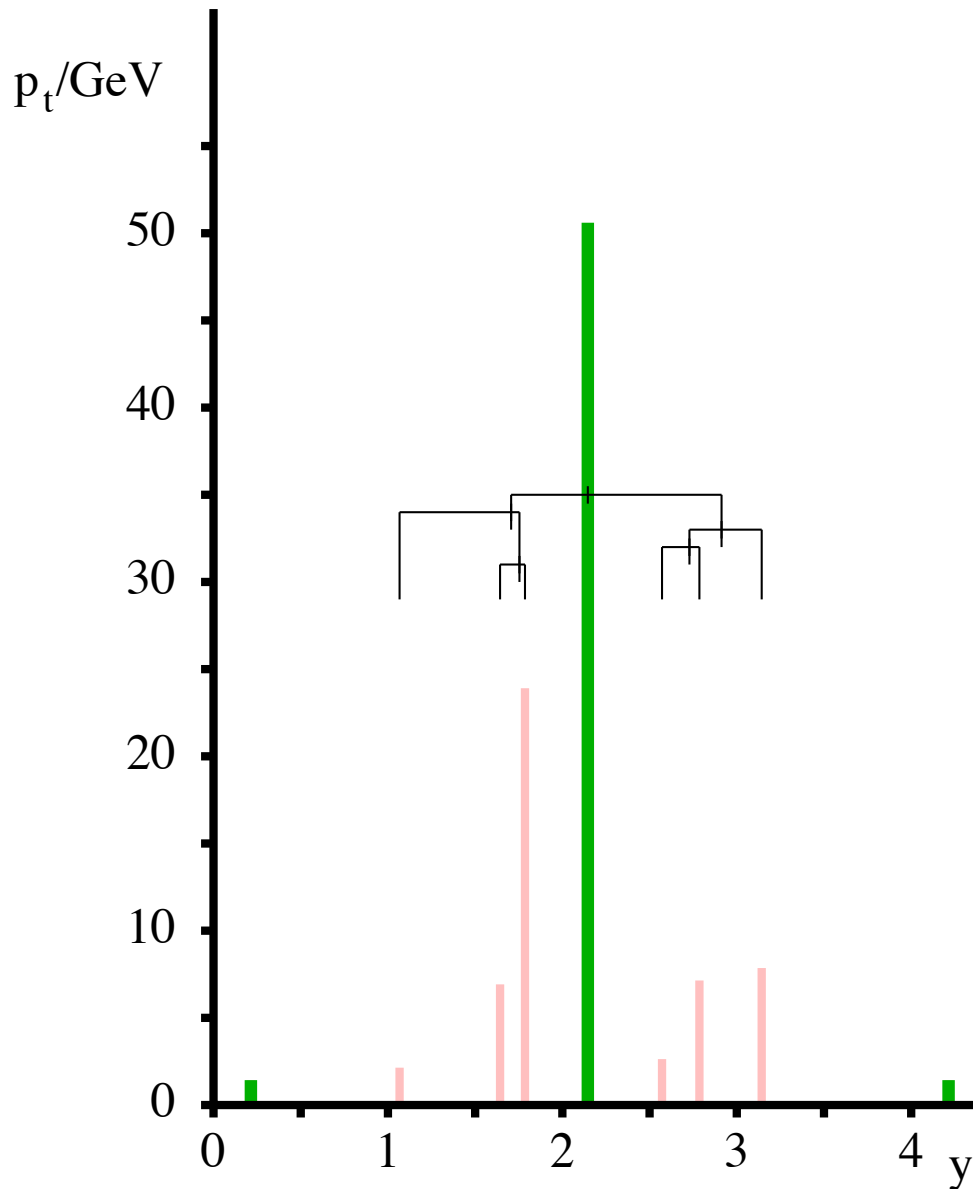


How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: Cam/Aachen

## Cambridge/Aachen algorithm

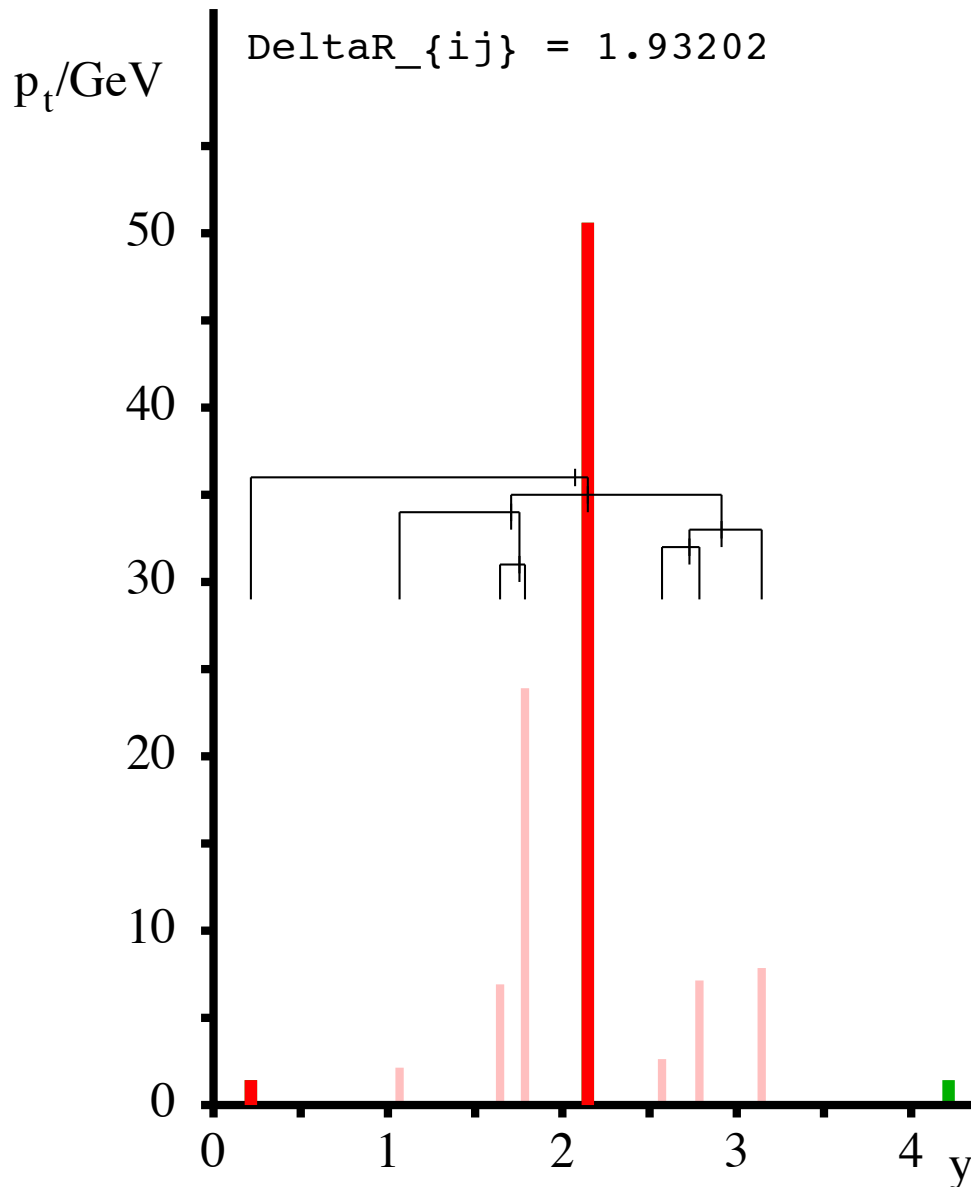


How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: Cam/Aachen

## Cambridge/Aachen algorithm

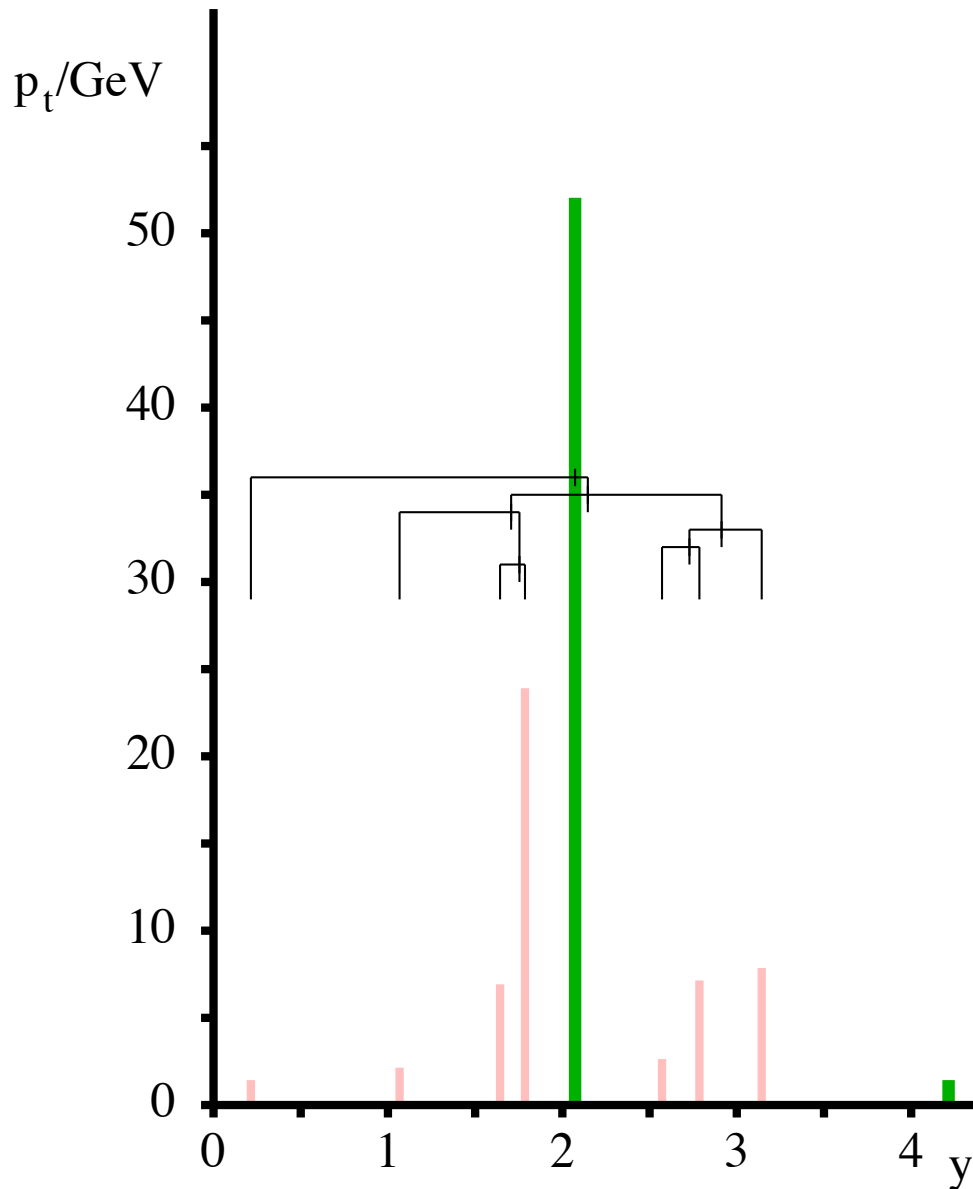


How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: Cam/Aachen

## Cambridge/Aachen algorithm

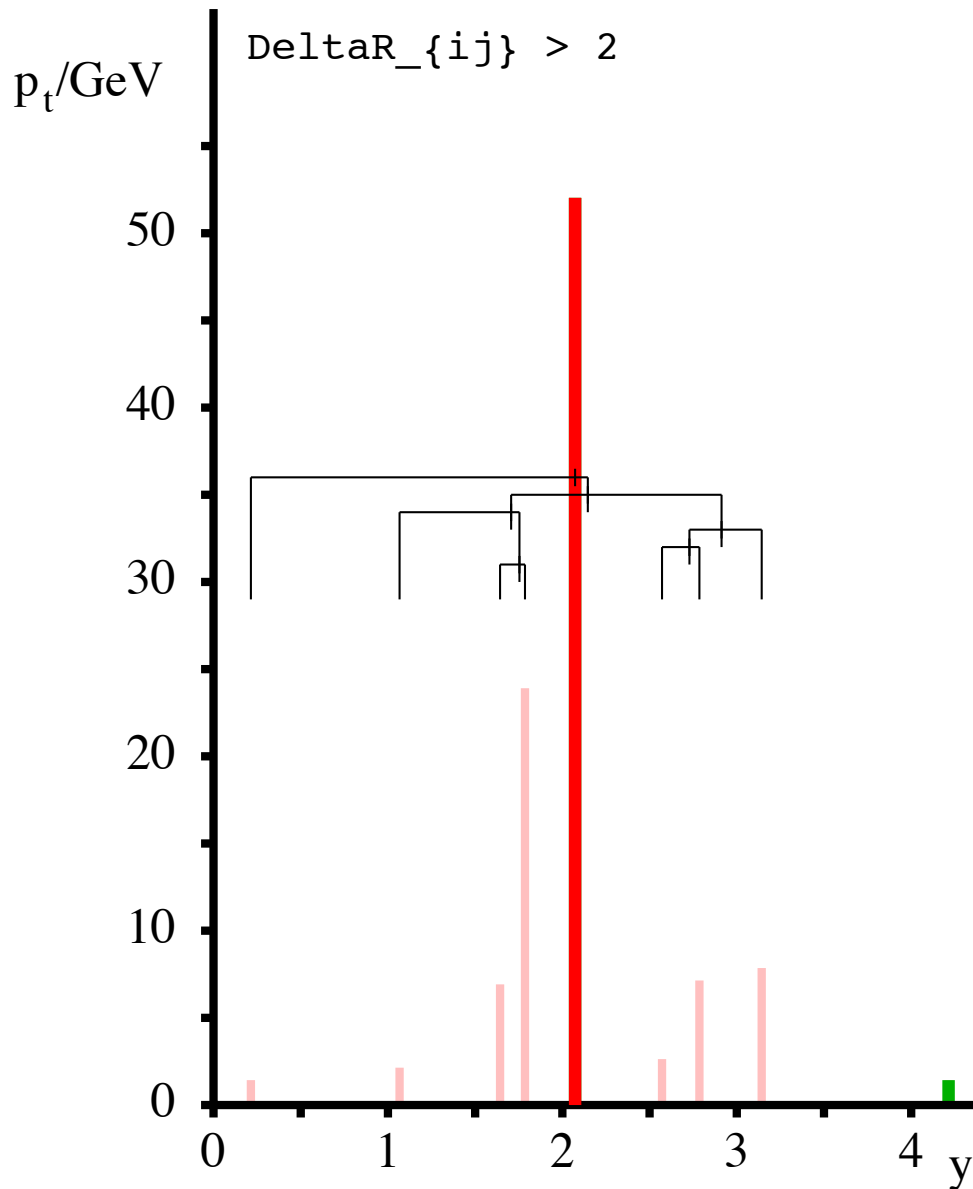


How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: Cam/Aachen

## Cambridge/Aachen algorithm



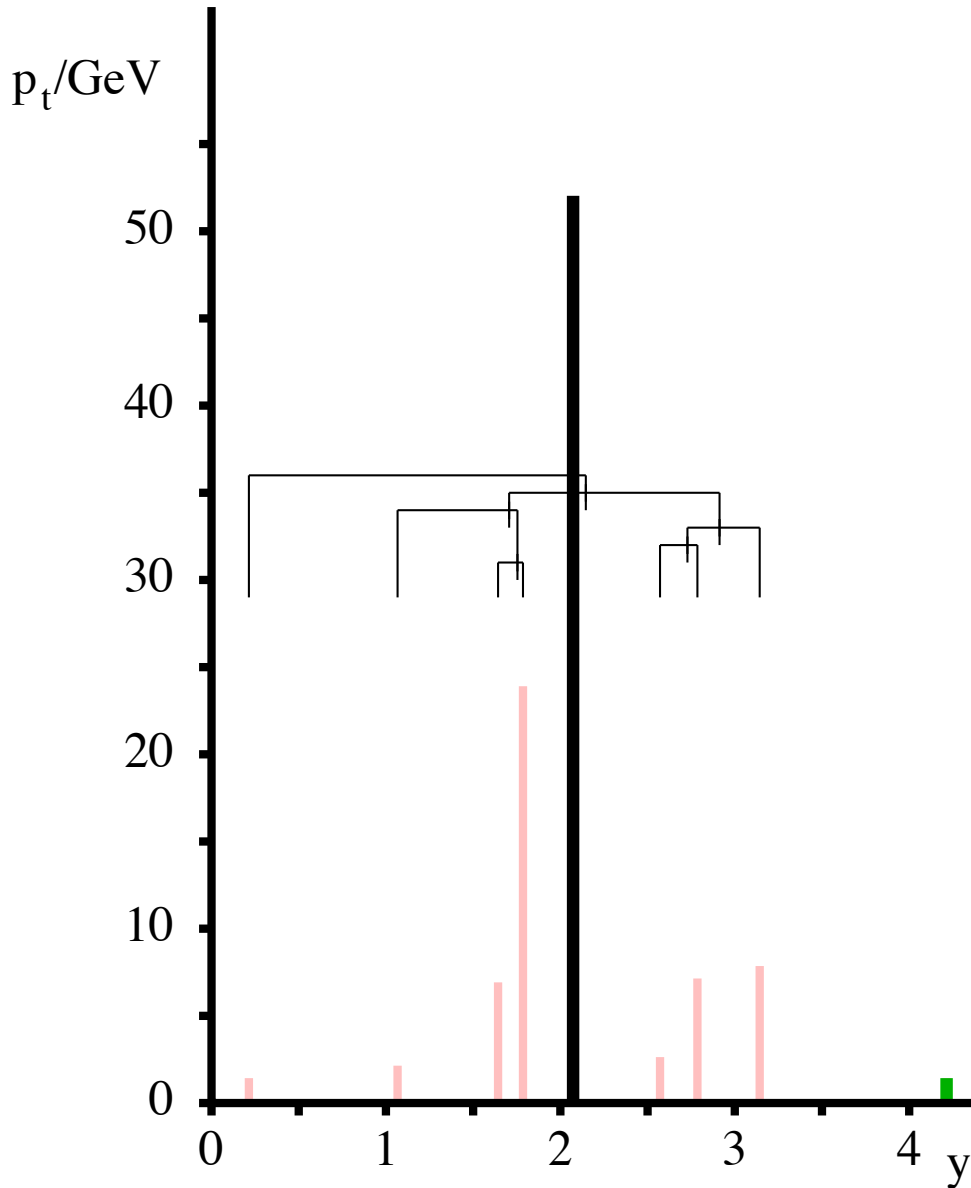
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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



# Identifying jet substructure: Cam/Aachen

## Cambridge/Aachen algorithm

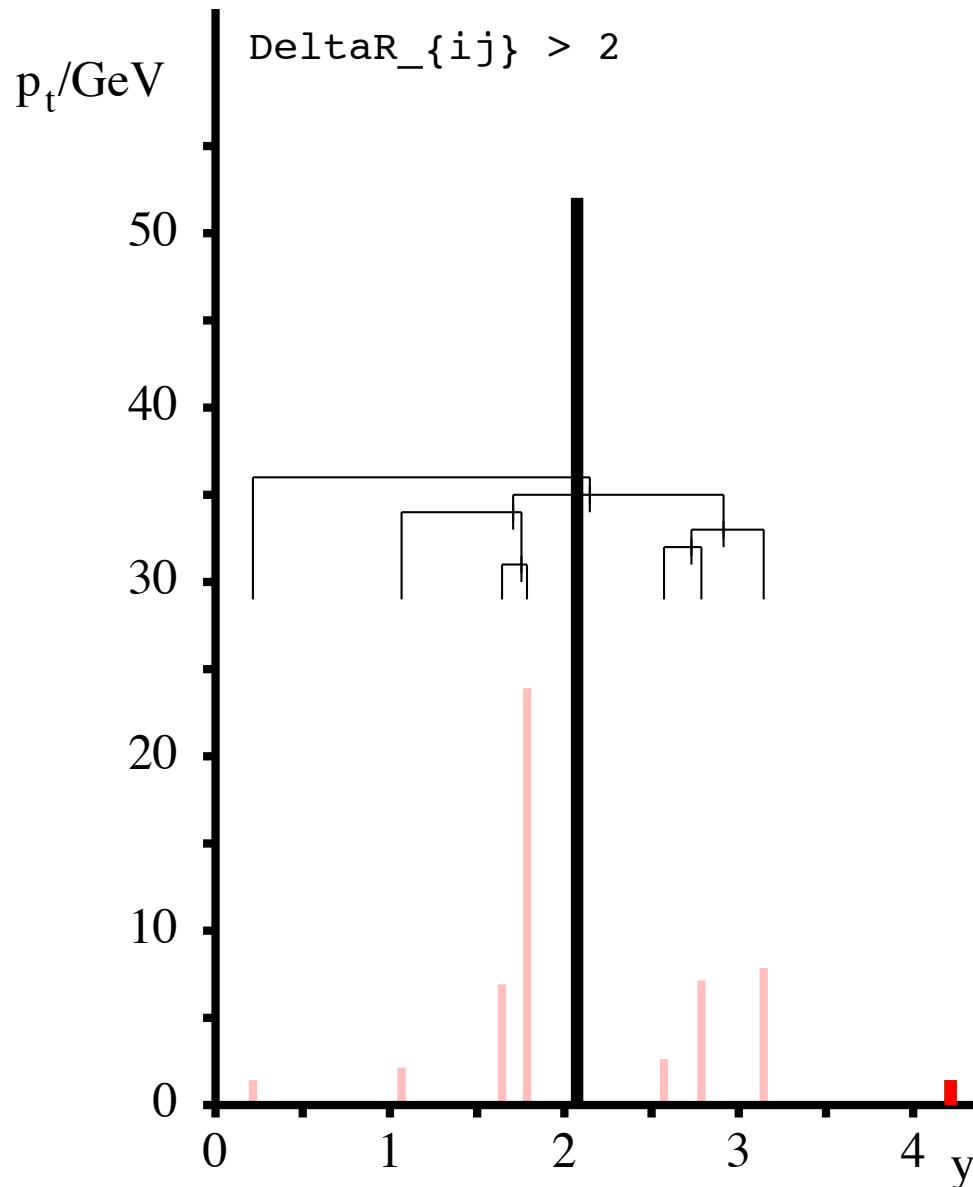


How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

# Identifying jet substructure: Cam/Aachen

## Cambridge/Aachen algorithm

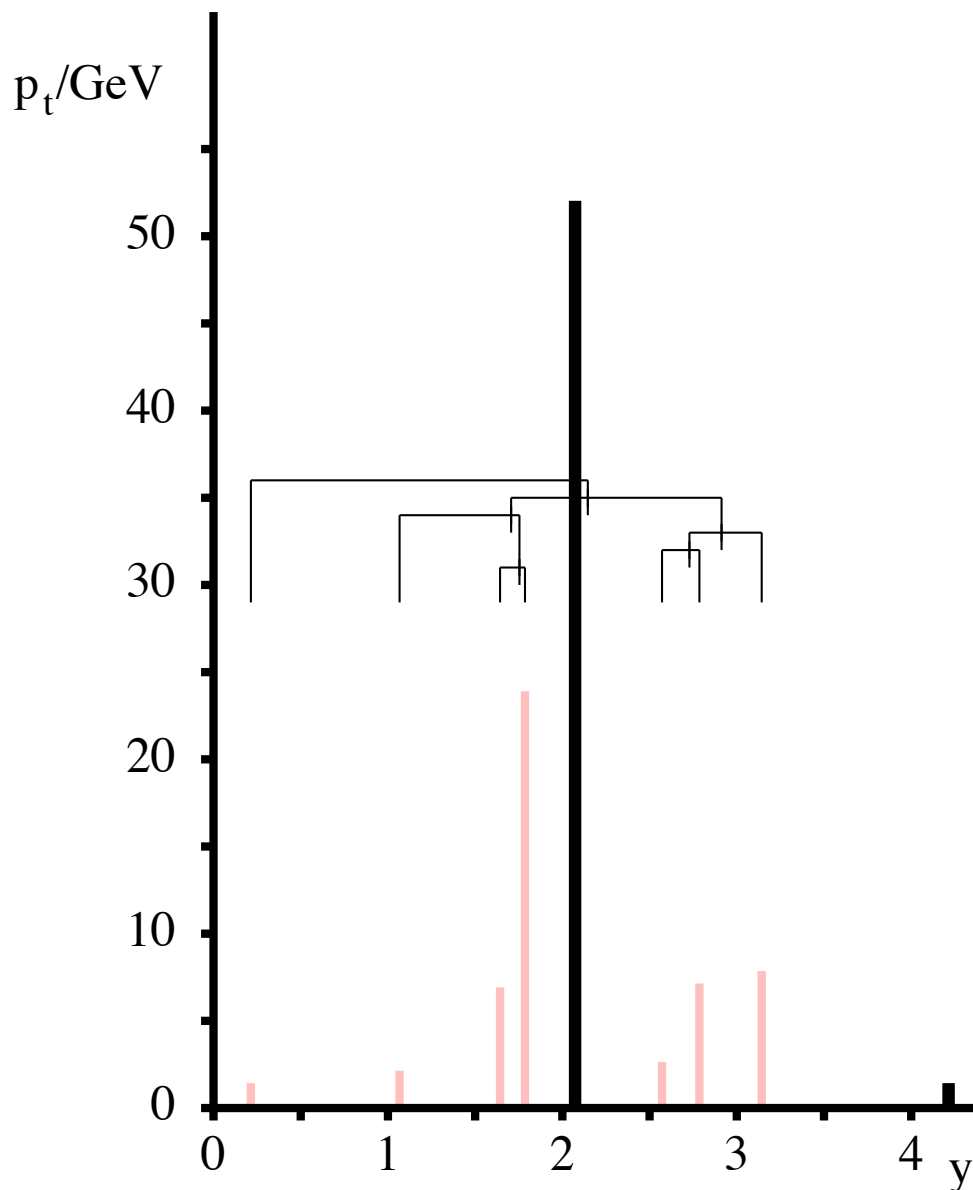


How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

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

## Identifying jet substructure: Cam/Aachen

# Cambridge/Aachen algorithm



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

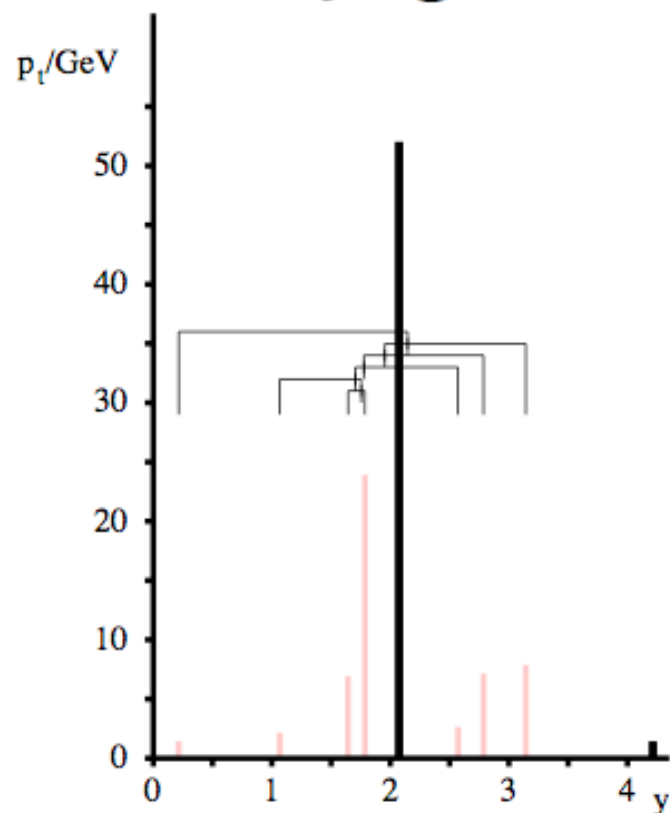
C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk

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

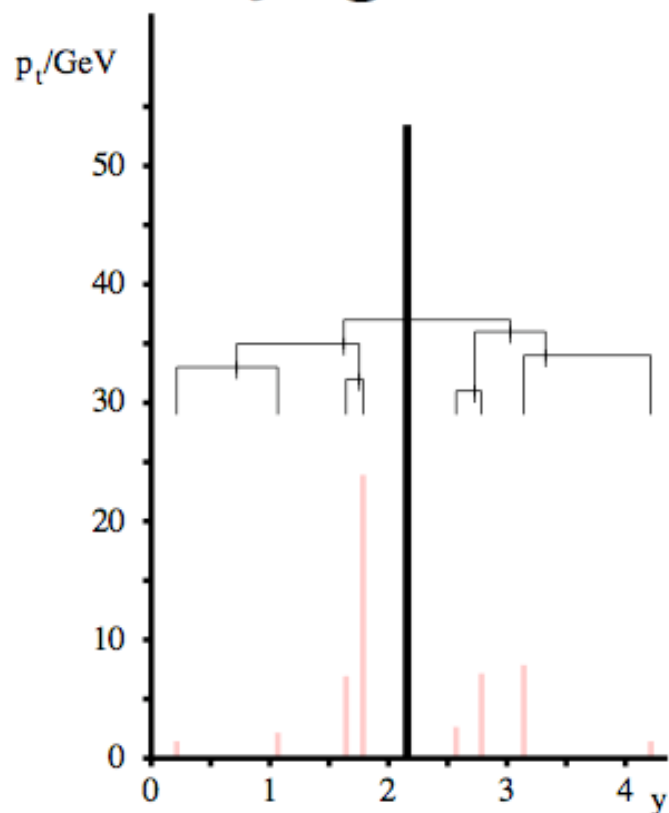
Butterworth, Davison, Rubin & GPS '08  
Kaplan, Schwartz, Rehman & Tweedie '08  
Butterworth, Ellis, Rubin & GPS '09  
Ellis, Vermilion & Walsh '09

# Hierarchical substructure

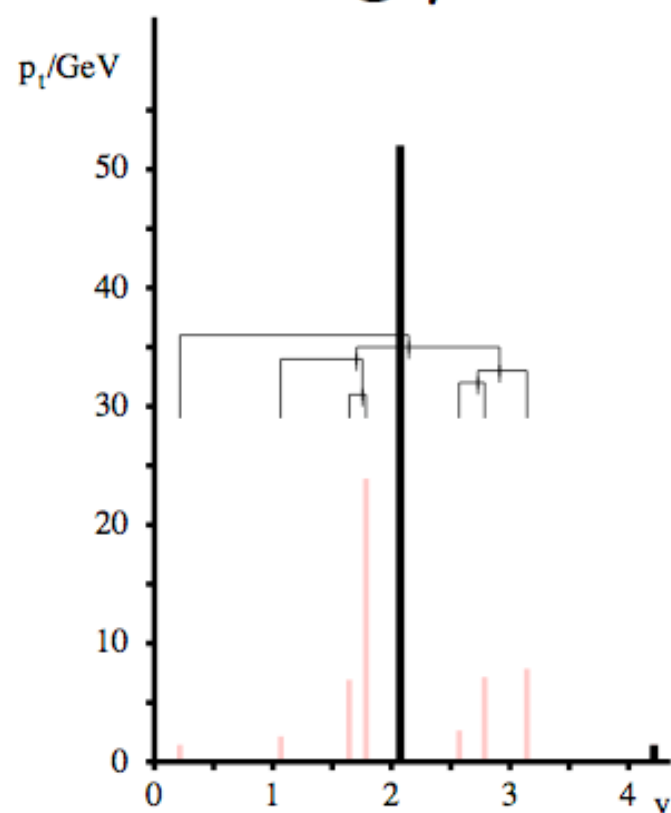
**anti- $k_t$  algorithm**



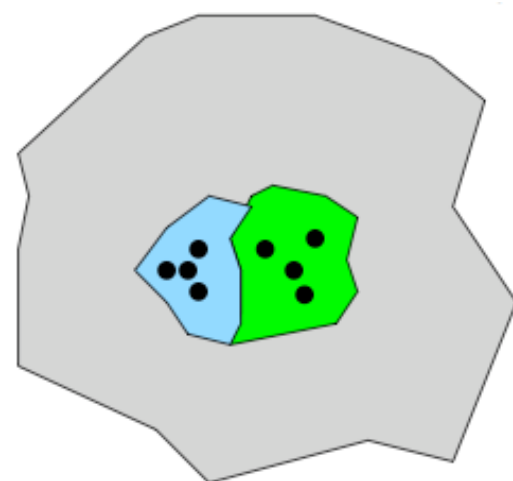
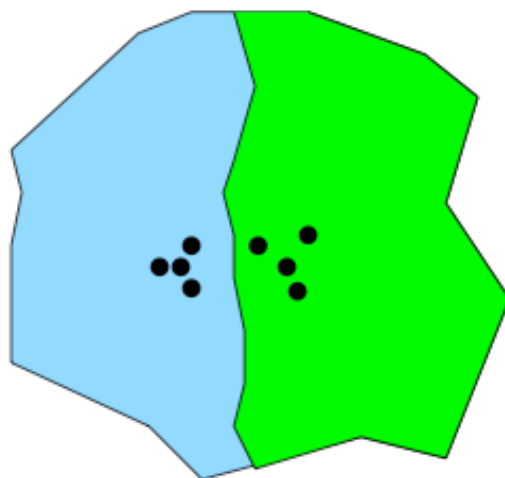
**$k_t$  algorithm**



**Cambridge/Aachen**



Undo the last  
clustering step(s)



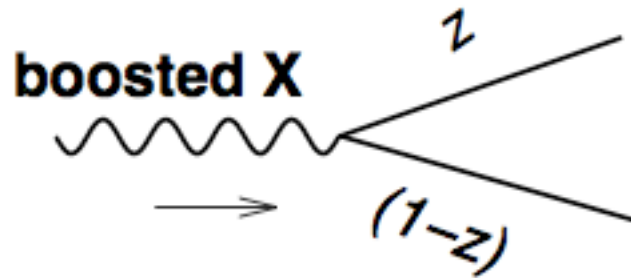
# The IRC safe algorithms

	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

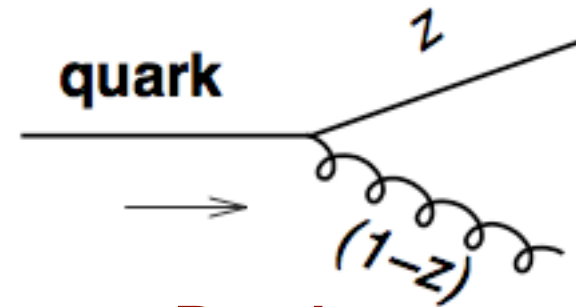
A possible approach for reducing the QCD background is to identify the two prongs of the heavy particle decay, and put a cut on their momentum fraction



**Signal:**

$$P(z) \sim 1$$

Will split mainly  
**symmetrically**



**Background:**

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

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

Will split mainly  
**asymmetrically**

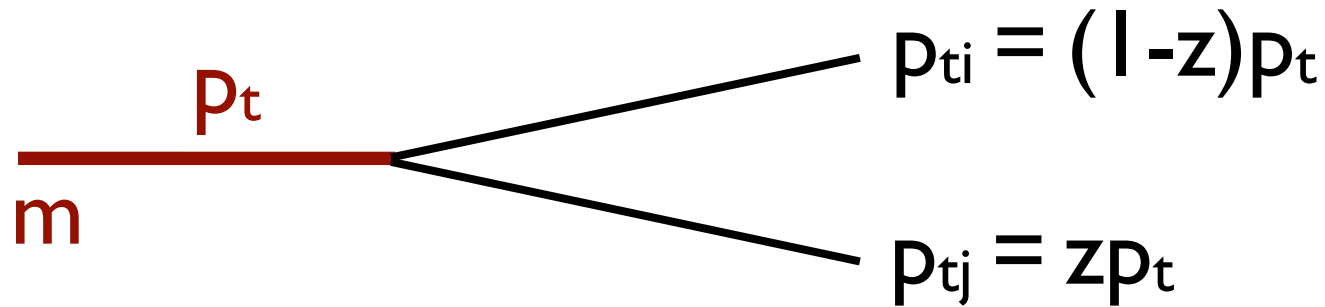
Potential tagger: **asymmetric splitting**

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

$k_t$  distance: 
$$d_{ij}^{(p_{tj} \leq p_{ti})} \simeq \frac{z}{1-z}m^2$$

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 wish 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.*

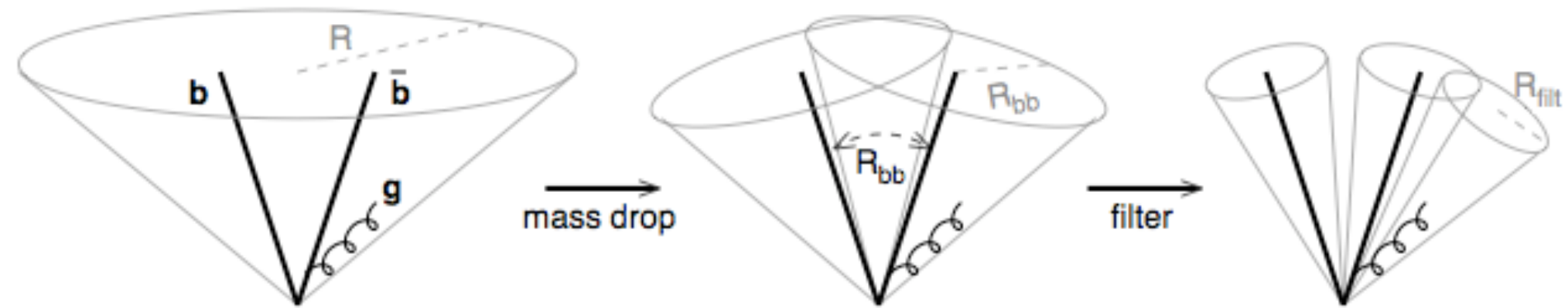
A possible approach is to use a **Mass-Drop** requirement:  
the clustering is **progressively undone**, and a splitting is **the relevant one** if  
**both subjects are much less massive than their combination**



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

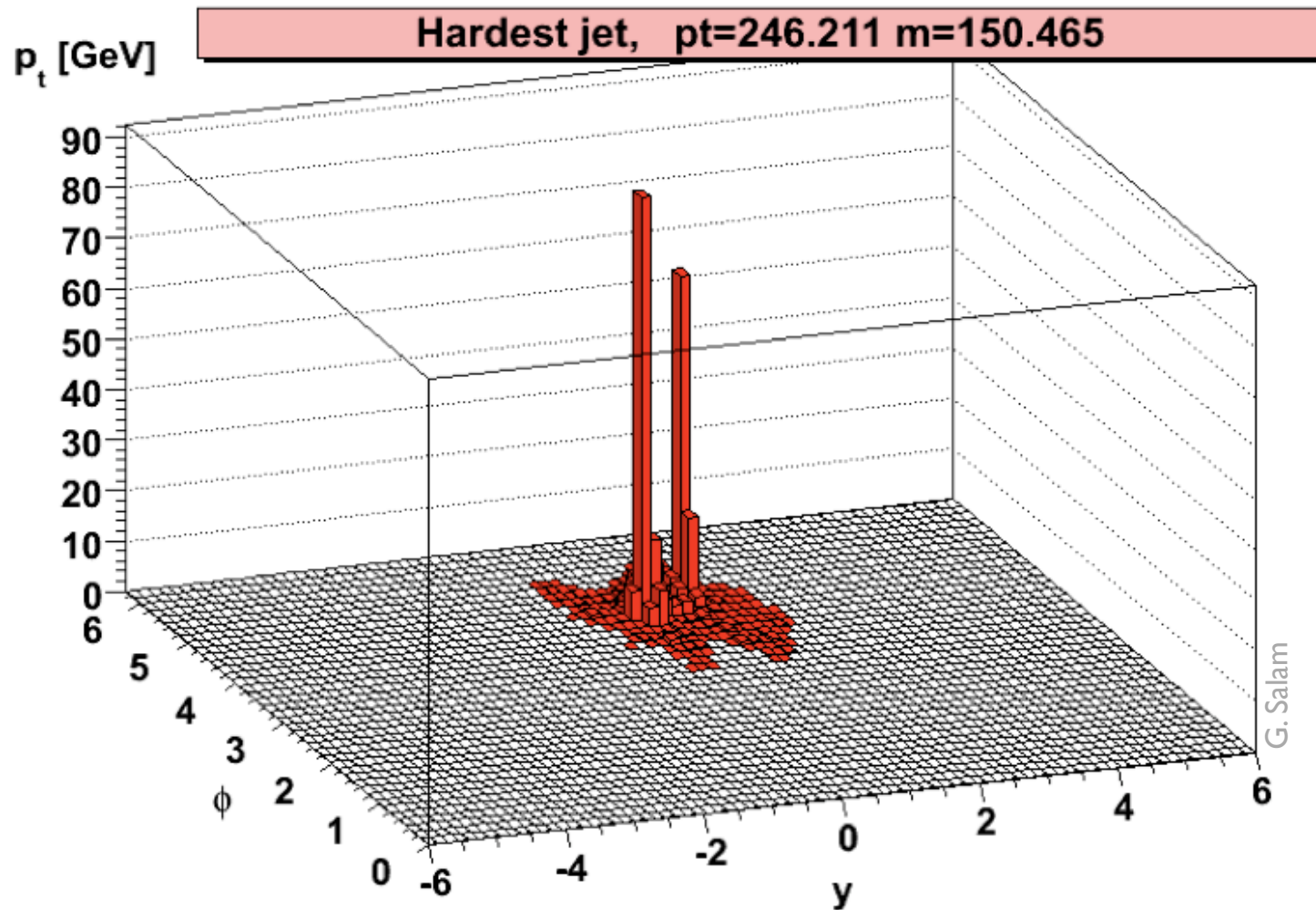
# The BDRS tagger/groomer

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**')

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

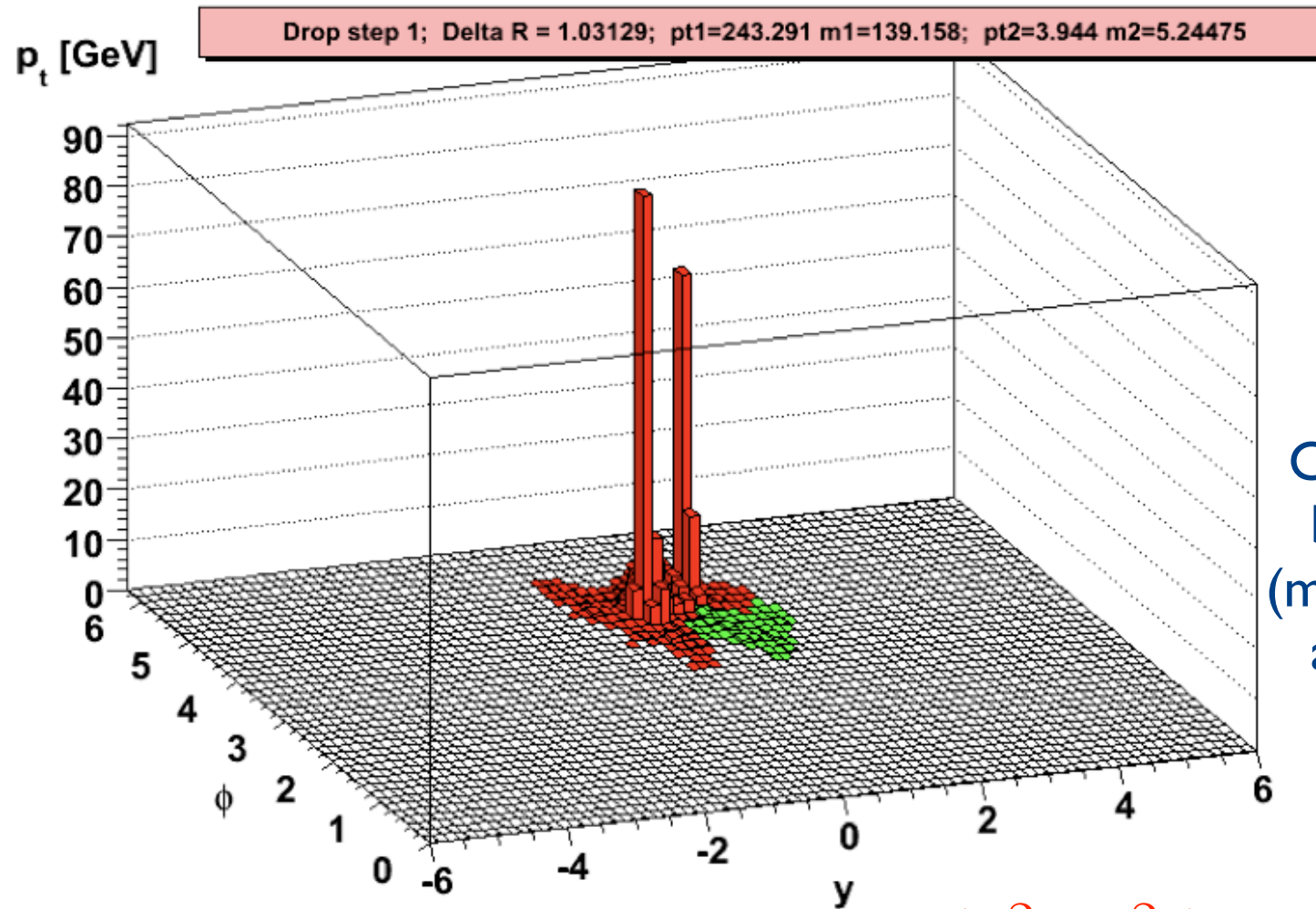


Start with the  
hardest jet

Use C/A with  
large  $R=1.2$

$$m_j = 150 \text{ GeV}$$

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

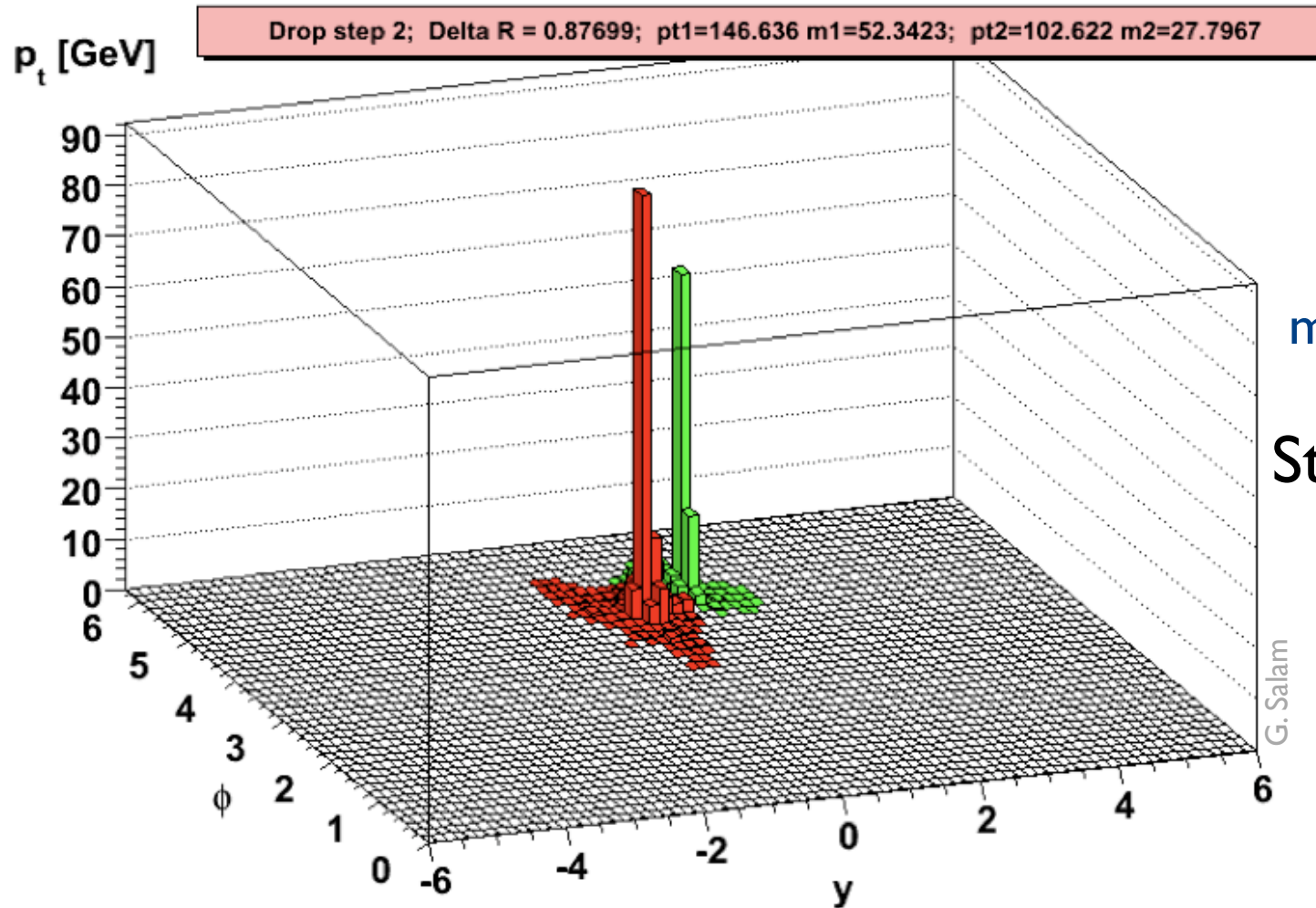


Undo last step of clustering

Check how the mass splits between the two subjects ( $m_1 = 139$  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

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



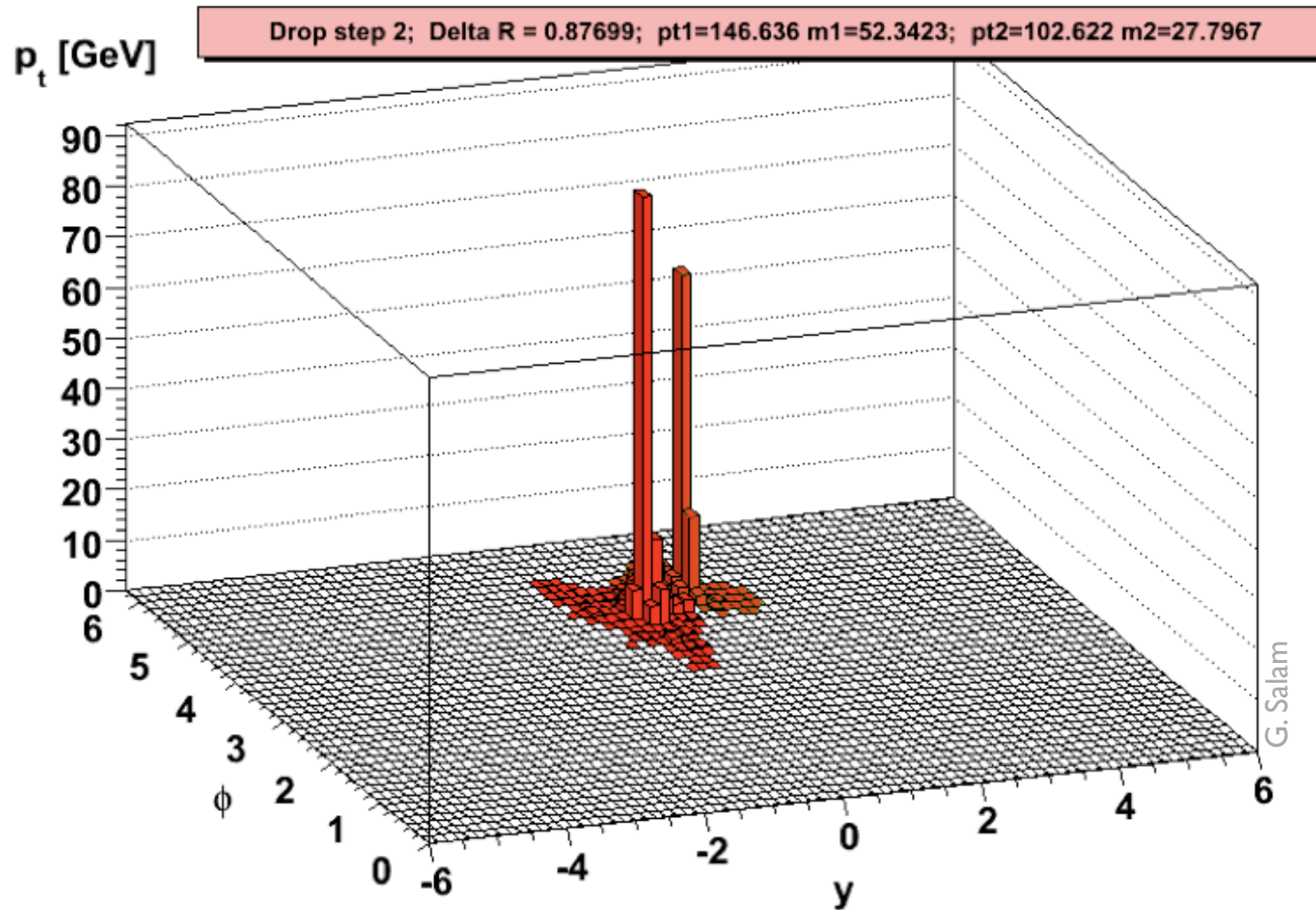
$$m_1 = 52 \text{ GeV}, m_2 = 28 \text{ GeV}$$

Stop when a **large mass drop** is observed  
(and **recombine** these two jets)

[NB. Parameters used  $\mu = 0.67$  and  $y_{\text{cut}} = 0.09$ ]

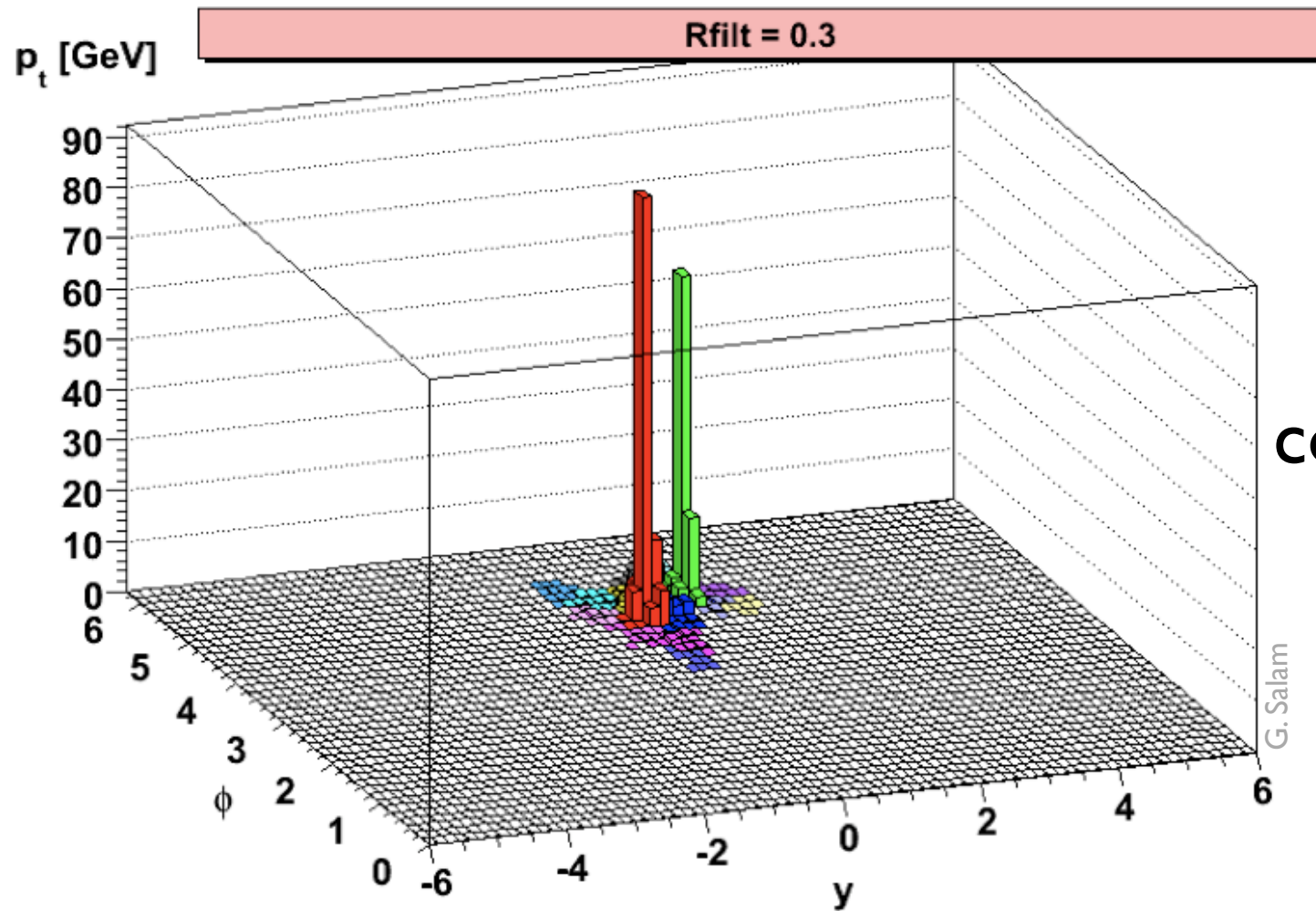


$pp \rightarrow ZH \rightarrow \nu\nu b\bar{b}$



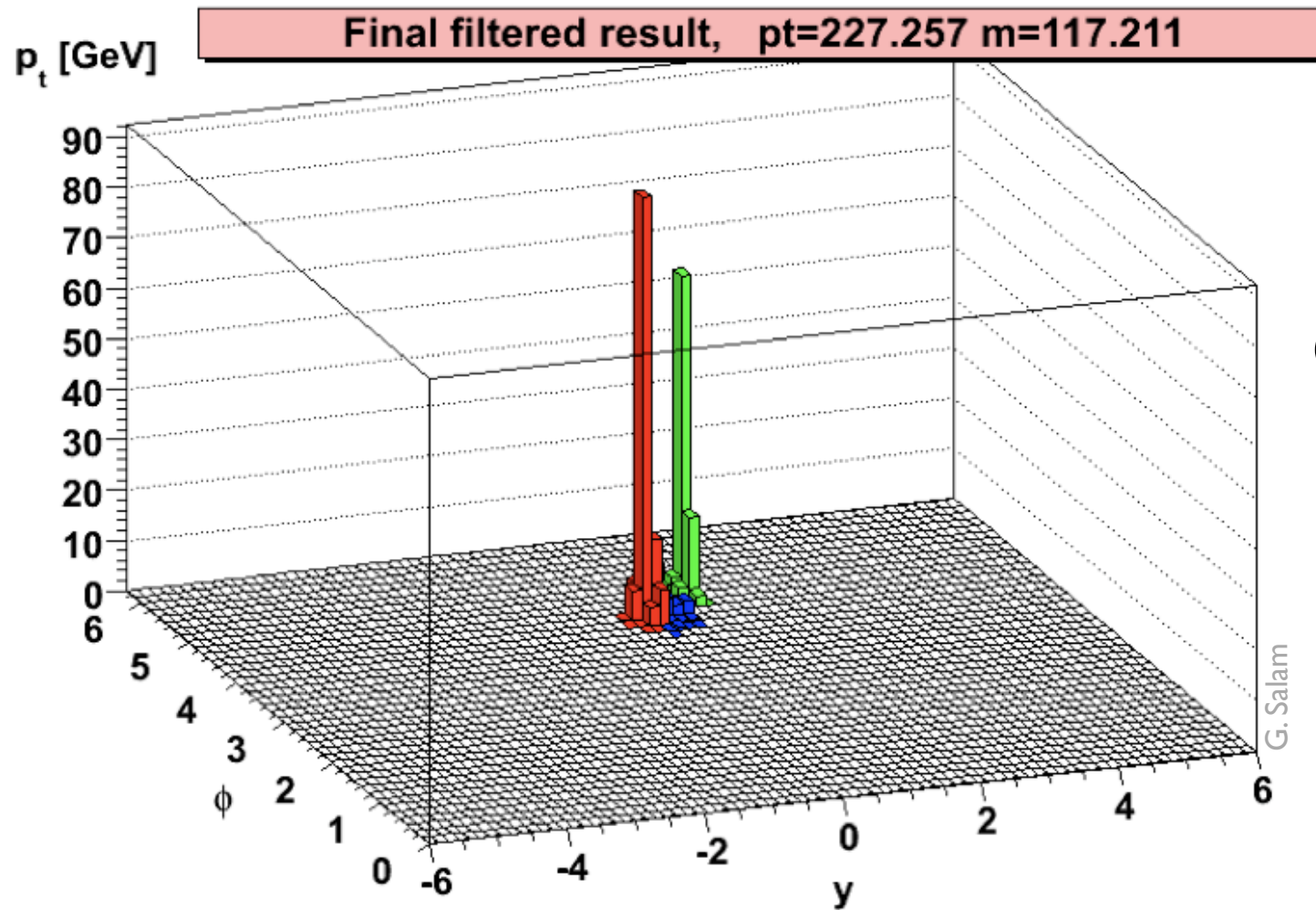
Start with the recombined jet

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



Recluster the  
constituents with  $R_{\text{filt}}$

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



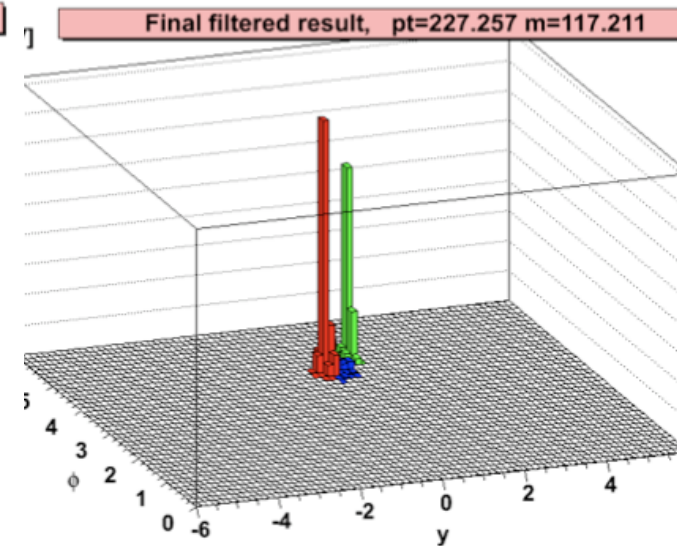
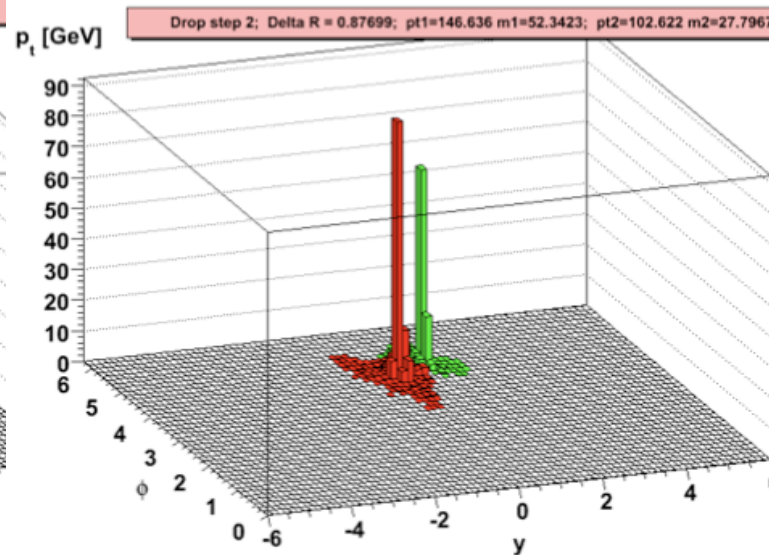
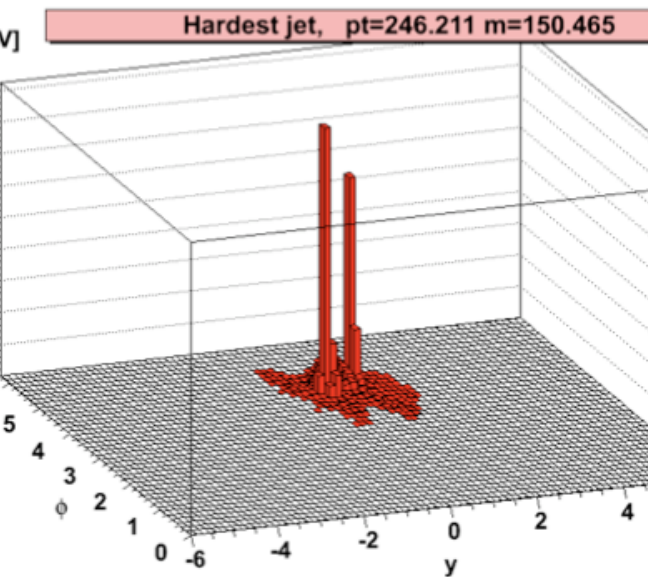
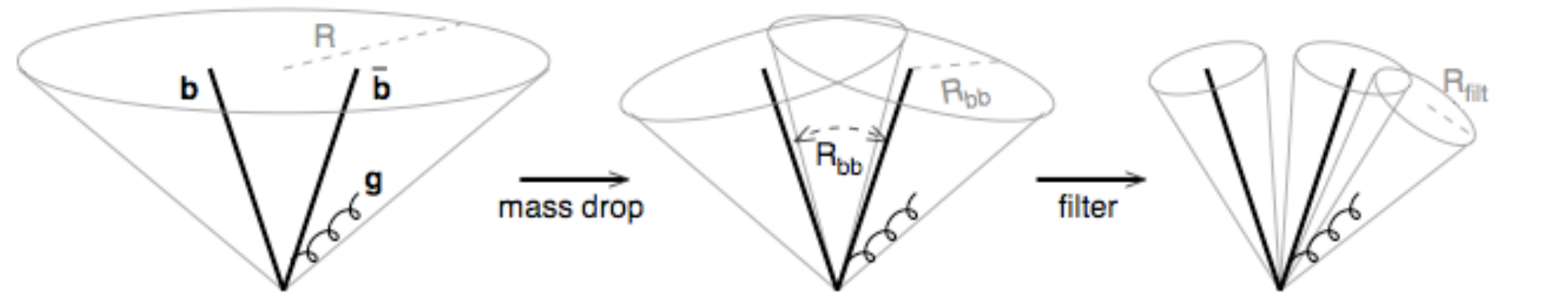
Only keep the  $n_{\text{filt}}$  hardest jets

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

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

# Visualisation of BDRS

Butterworth, Davison, Rubin, Salam, 2008



Cluster with a large  $R$

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

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



# First taggers/groomers

## ► Mass Drop + Filtering

Butterworth, Davison, Rubin, Salam, 2008

Decluster with mass drop and asymmetry conditions

Recluster constituents into subjects at distance scale  $R_{\text{filt}}$ , retain  $n_{\text{filt}}$  hardest subjects

## ► Jet ‘trimming’

Krohn, Thaler, Wang, 2009

Recluster constituents into subjects at distance scale  $R_{\text{trim}}$ ,

retain subjects with  $p_{t,\text{subject}} > \epsilon_{\text{trim}} p_{t,\text{jet}}$

## ► Jet ‘pruning’

S. Ellis, Vermilion, Walsh, 2009

While building up the jet, discard softer subjects when  $\Delta R > R_{\text{prune}}$

and  $\min(p_{t1}, p_{t2}) < \epsilon_{\text{prune}} (p_{t1} + p_{t2})$

**Aim: limit contamination from QCD background while retaining bulk of perturbative radiation**

**Trimming and pruner are a priori groomers, but can become taggers when combined with an invariant mass window test (if you can groom everything then there’s no heavy particle in the jet)**

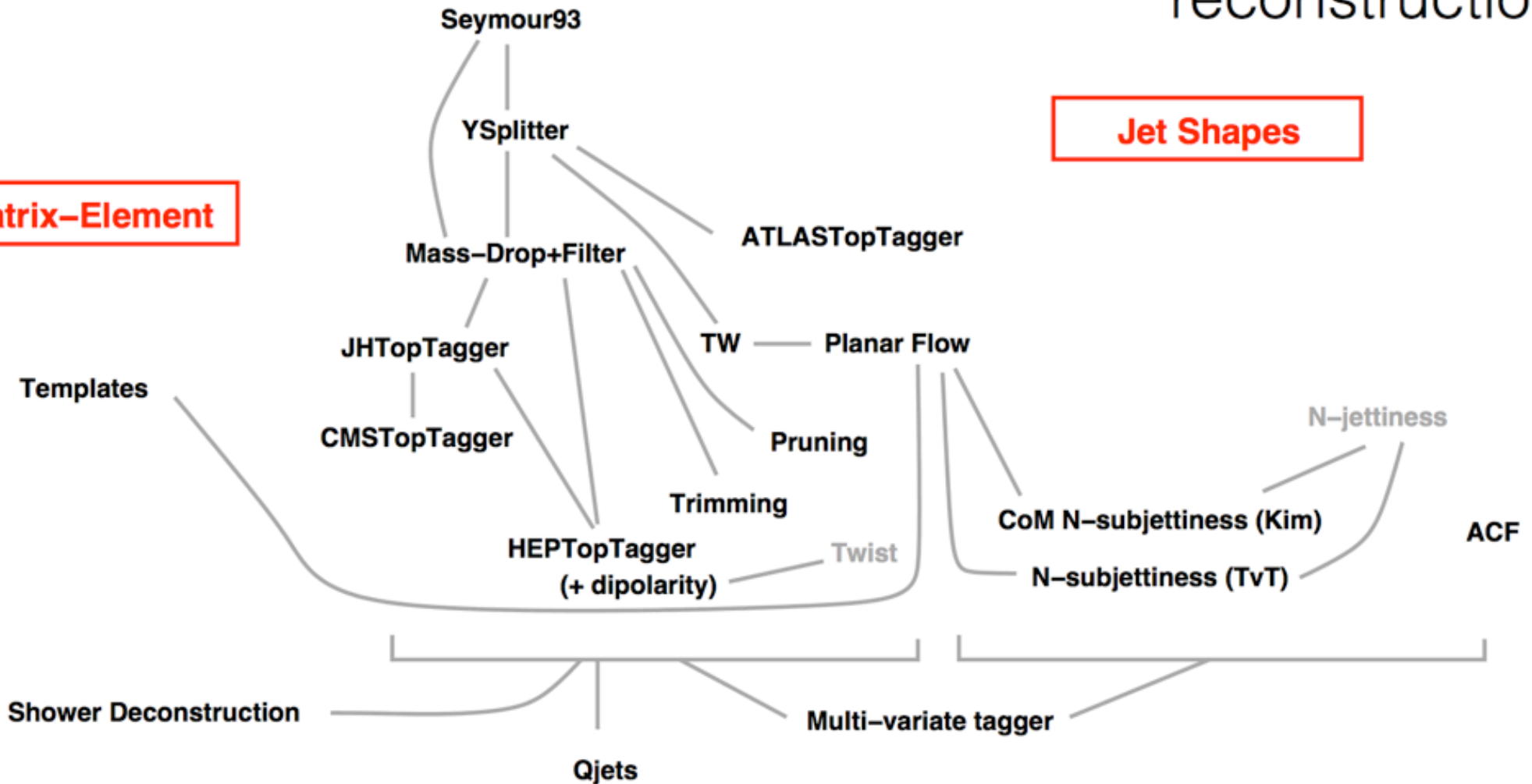
# The jet substructure maze

Some of the tools developed  
for boosted W/Z/H/top  
reconstruction

## Jet Declustering

## Jet Shapes

## Matrix-Element



Slide by G. Salam, now a few years old

# Soft Drop declustering

Larkoski, Marzani, Soyez, Thaler, 2014

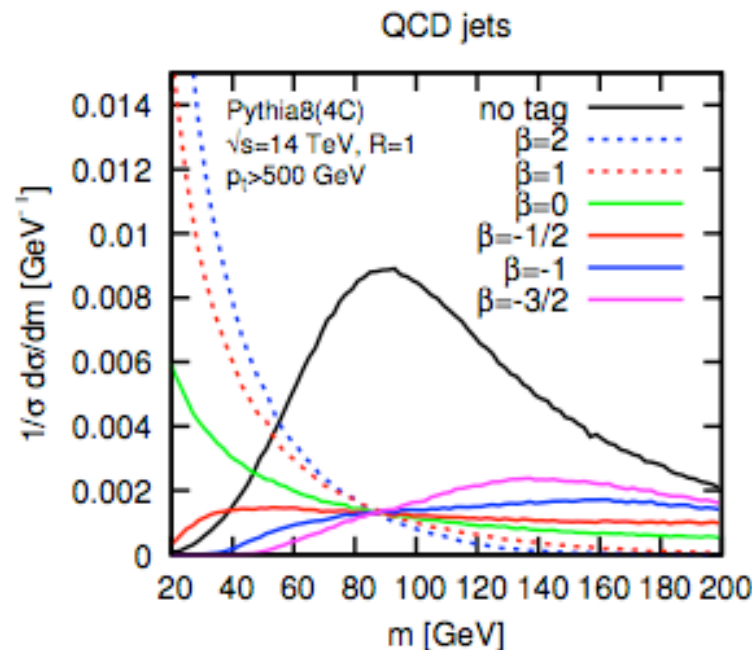
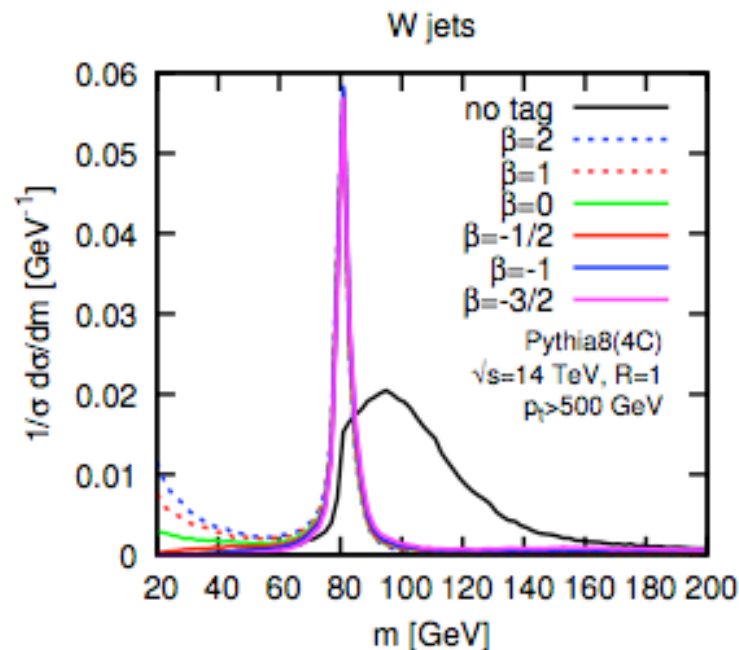
Decluster and drop softer constituent unless

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

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

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

# Alternatives to hierarchical substruct.

- ▶ If what we are interested in is the structure of the constituents of a jet, the “jet” itself is not the most important feature.
- ▶ A different algorithm, or simply the study of the constituents in a certain patch will also do. Selected alternatives are:
  - ▶ Use of jet-shapes to characterise certain features
    - ▶ e.g. *N-subjettiness*: how many subjects a jets appears to have  
Thaler, van Tilburg, 2011
  - ▶ Alternative ways of clustering
    - ▶ e.g. *Qjets*: the clustering history not deterministic, but controlled by random probabilities of merging. Can be combined with, e.g. pruning  
Ellis, Hornig, Roy, Krohn, Schwartz, 2012
  - ▶ Use information from matrix element
    - ▶ e.g. *shower deconstruction*: use analytic shower calculations to estimate probability that a certain configuration comes from signal or from background  
Soper, Spannowsky, 2011
  - ▶ Use event shapes mimicking jet properties
    - ▶ e.g. *JetsWithoutJets*, mimicking trimming  
Bertolini, Chen, Thaler, 2013

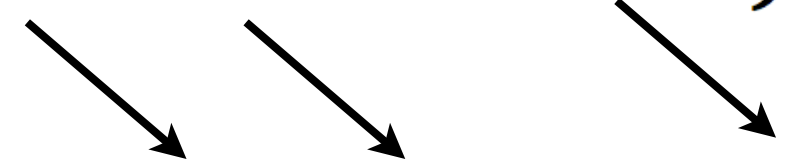
# N-subjettiness

Thaler, van Tilburg, 2010

$$\tau_N^{(\beta)} = \sum_i p_{Ti} \min \left\{ R_{1,i}^\beta, R_{2,i}^\beta, \dots, R_{N,i}^\beta \right\}$$

Sum over constituents of a jet

Distances to axes of N subjets



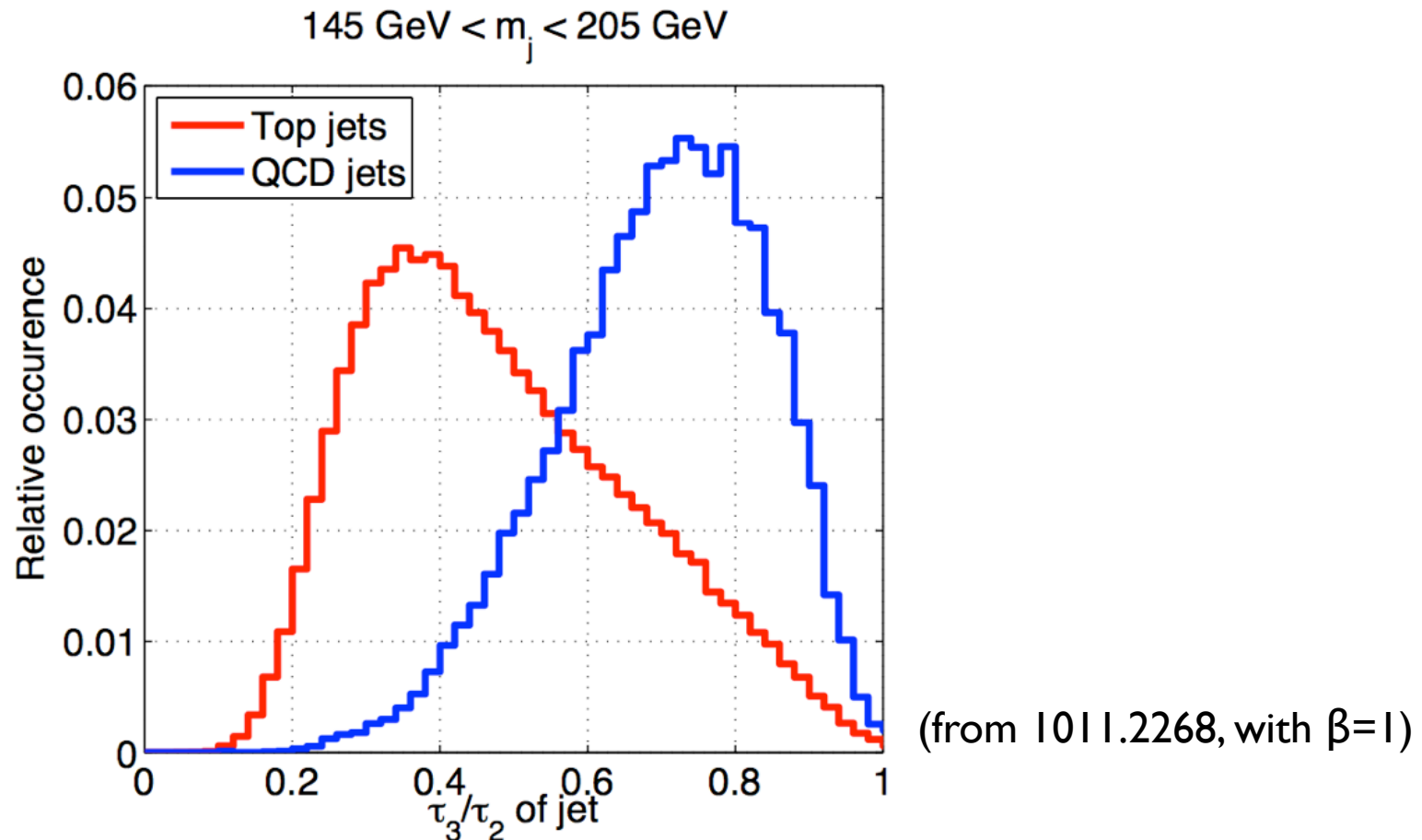
$\tau_N$  measures departure from N-parton energy flow:  
*if a jet has N subjets,  $\tau_{N-1}$  should be much larger than  $\tau_N$*

# N-subjettiness

Thaler, van Tilburg, 2010

$$\tau_{N,N-1}^{(\beta)} \equiv \frac{\tau_N^{(\beta)}}{\tau_{N-1}^{(\beta)}}$$

A jet with a **small**  $\tau_{N,N-1}$   
is more likely to have  
N than N-1 subjects



# Energy correlation functions

*Probes of N-prong structures without requiring identification of subjects*

$$ECF(N, \beta) = \sum_{i_1 < i_2 < \dots < i_N \in J} \left( \prod_{a=1}^N p_{T i_a} \right) \left( \prod_{b=1}^{N-1} \prod_{c=b+1}^N R_{i_b i_c} \right)^\beta$$

Angular ( $y$ - $\varphi$ ) distances  
between constituents

$ECF(N+1)$  is zero if there are only  $N$  particles

*More generally, if there are  $N$  subjects one expects  $ECF(N+1)$  to be much smaller than  $ECF(N)$   
[because radiation will be mainly soft/collinear to subjects]*

# Discriminators

$$r_N^{(\beta)} \equiv \frac{\text{ECF}(N+1, \beta)}{\text{ECF}(N, \beta)}$$

small for N prongs:  
if N hard partons, small if radiation  
only soft-collinear

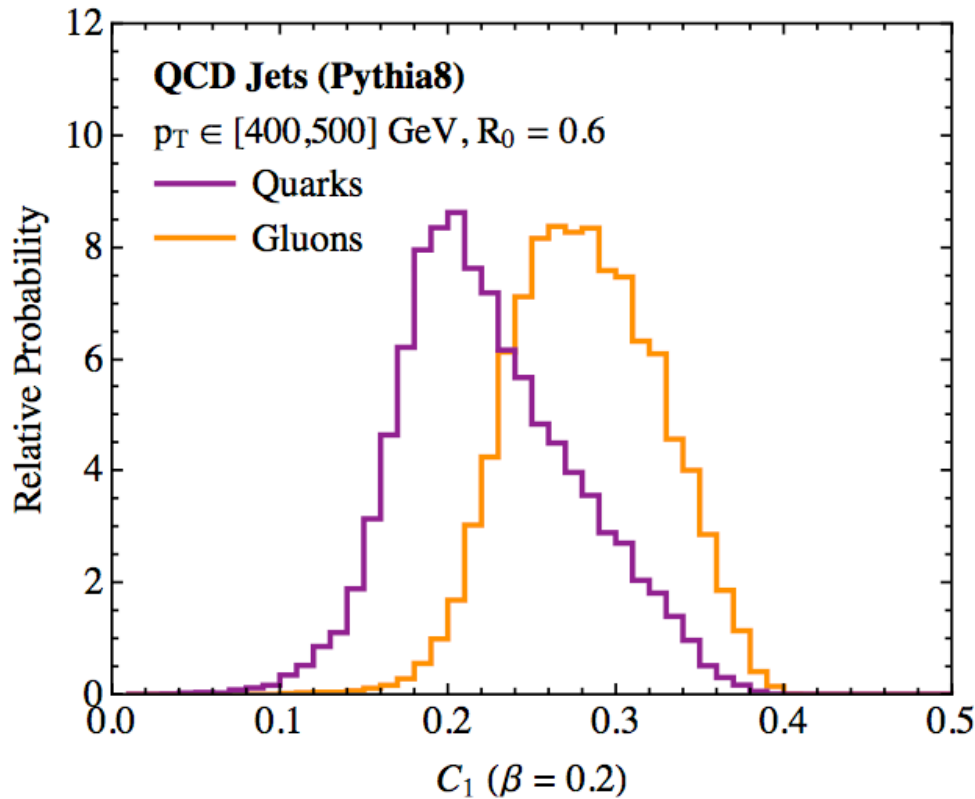
$$C_N^{(\beta)} \equiv \frac{r_N^{(\beta)}}{r_{N-1}^{(\beta)}} = \frac{\text{ECF}(N+1, \beta) \text{ECF}(N-1, \beta)}{\text{ECF}(N, \beta)^2}$$

A jet with a **small**  $C_N$  is more likely  
to have N prongs and at most soft/coll radiation

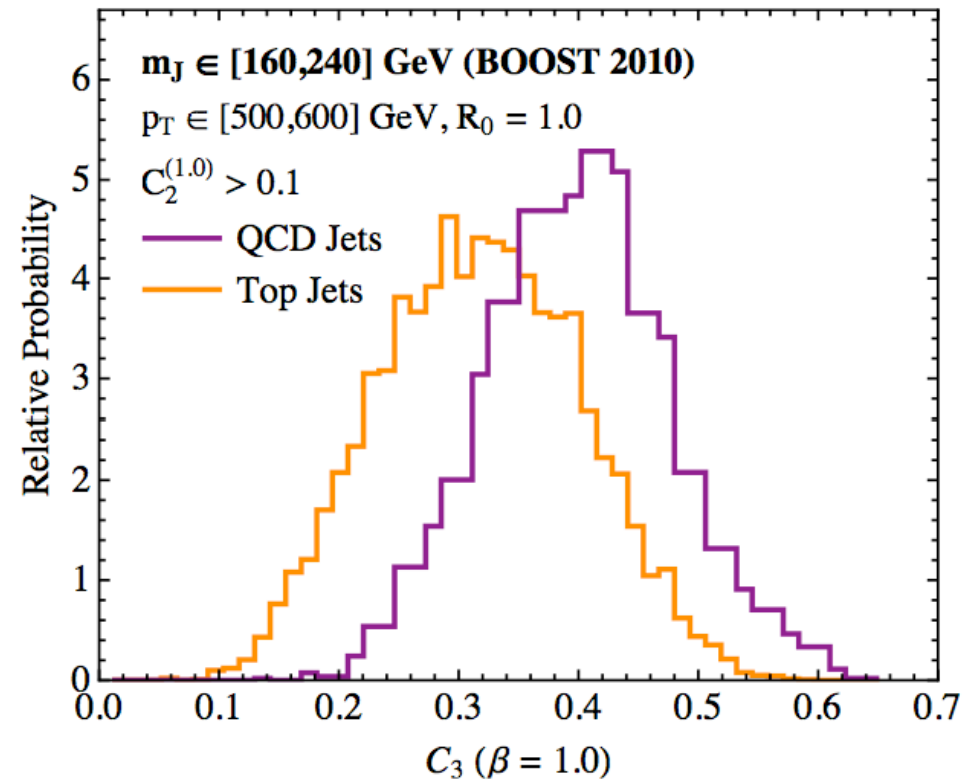


$C_1$ 

quark-gluon discriminator

 $C_3$ 

top tagging



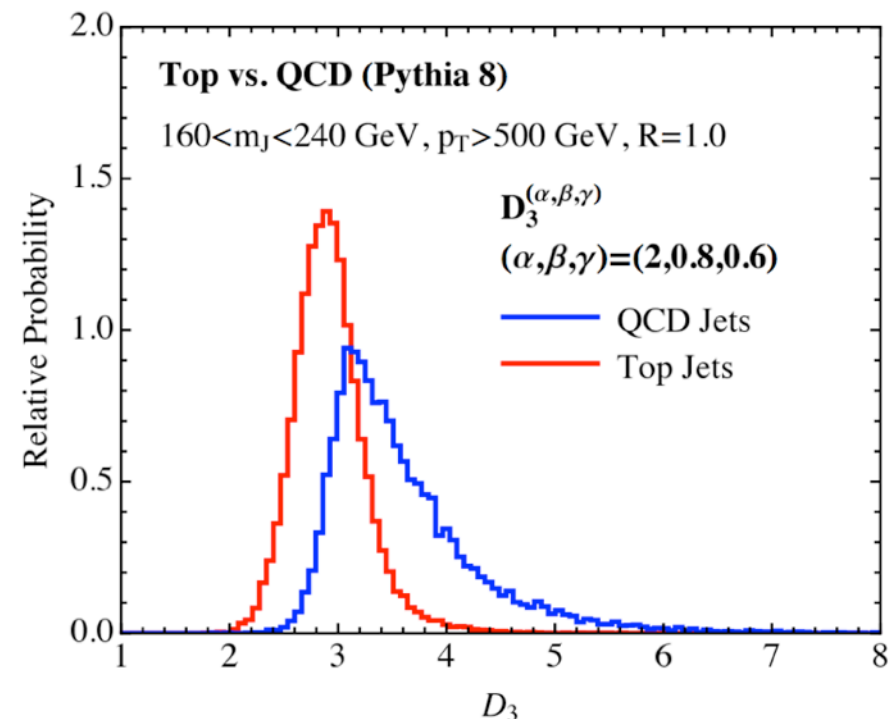
Note different values of  $\beta$   
 (chosen to maximise discriminating power)

# The D functions are variations of the C ones

Instead of  $C_2^{(\beta)} = \frac{e_3^{(\beta)}}{(e_2^{(\beta)})^2}$   $C_3^{(\beta)} = \frac{e_4^{(\beta)} e_2^{(\beta)}}{(e_3^{(\beta)})^2}$

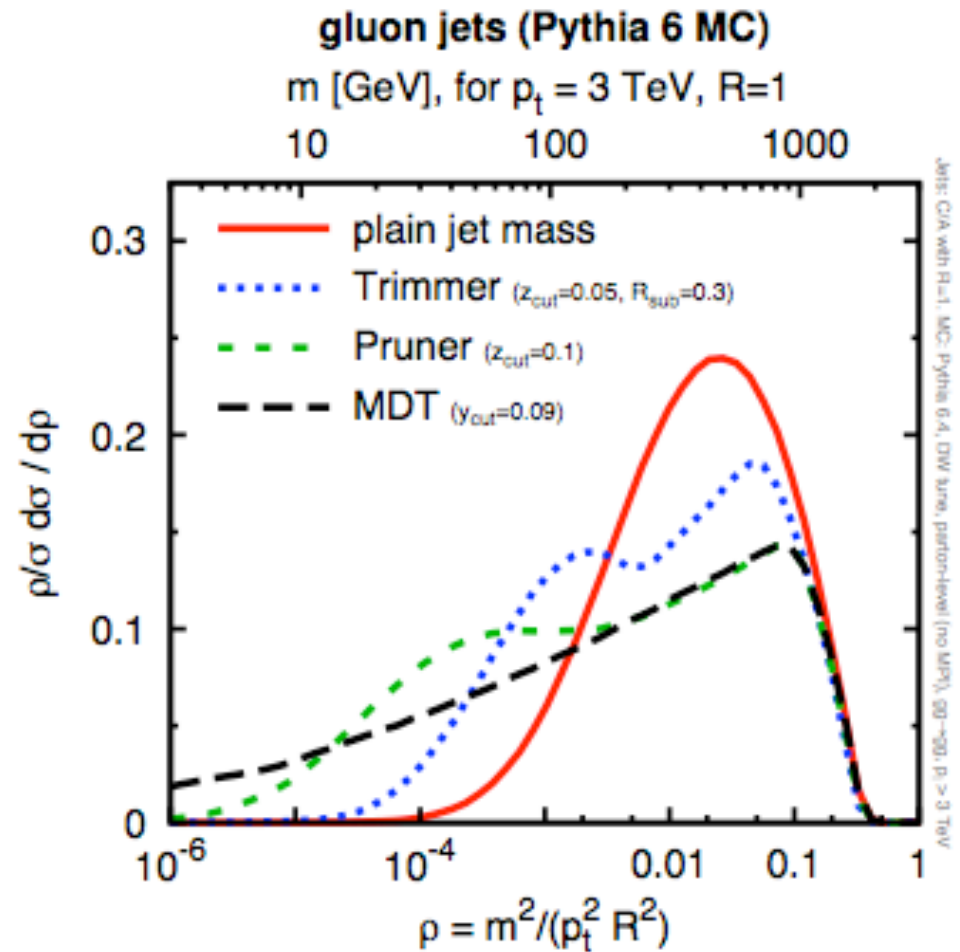
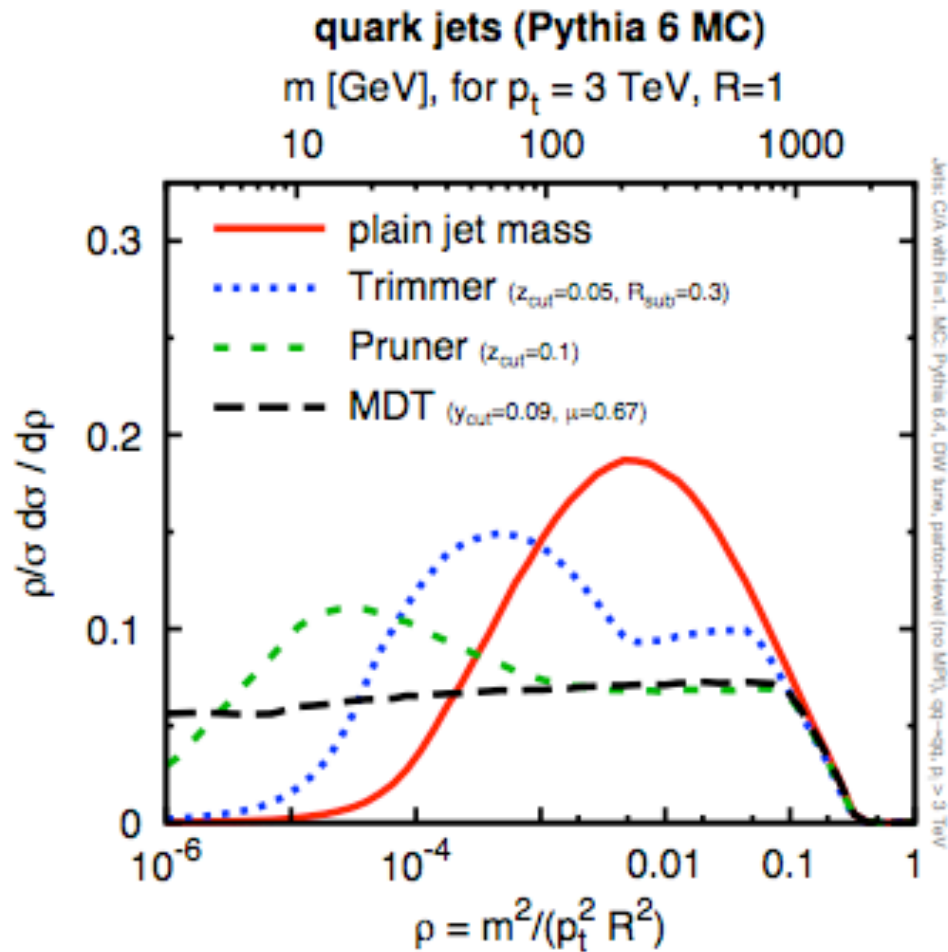
define  $D_2^{(\beta)} = \frac{e_3^{(\beta)}}{(e_2^{(\beta)})^3}$   $D_3^{(\alpha,\beta,\gamma)} = \frac{e_4^{(\gamma)} (e_2^{(\alpha)})^{\frac{3\gamma}{\alpha}}}{(e_3^{(\beta)})^{\frac{3\gamma}{\beta}}} + x \frac{e_4^{(\gamma)} (e_2^{(\alpha)})^{\frac{2\gamma}{\beta}-1}}{(e_3^{(\beta)})^{\frac{2\gamma}{\beta}}} + y \frac{e_4^{(\gamma)} (e_2^{(\alpha)})^{\frac{2\beta}{\alpha}-\frac{\gamma}{\alpha}}}{(e_3^{(\beta)})^2}$

Attempt to improve the discriminating power, and to account for different regions of phase space of radiation  
[also, gives an idea of increasing ‘sophistication’, or complexification]



# Robustness of substructure tools

Dasgupta, Fregoso, Marzani, Salam, 2013



Tools that are considered (or can be seen in Monte Carlo tests) to behave ‘similarly’ could cease to do so in different parameter regions

# Analytic calculations of jet substructure

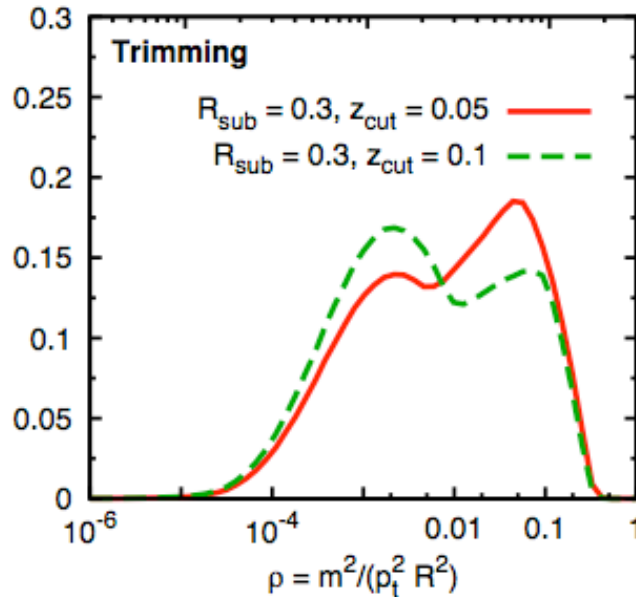
Dasgupta, Fregoso, Marzani, Salam, 2013

## Monte Carlo

Pythia 6 MC: gluon jets

$m$  [GeV], for  $p_t = 3$  TeV,  $R = 1$

10 100 1000



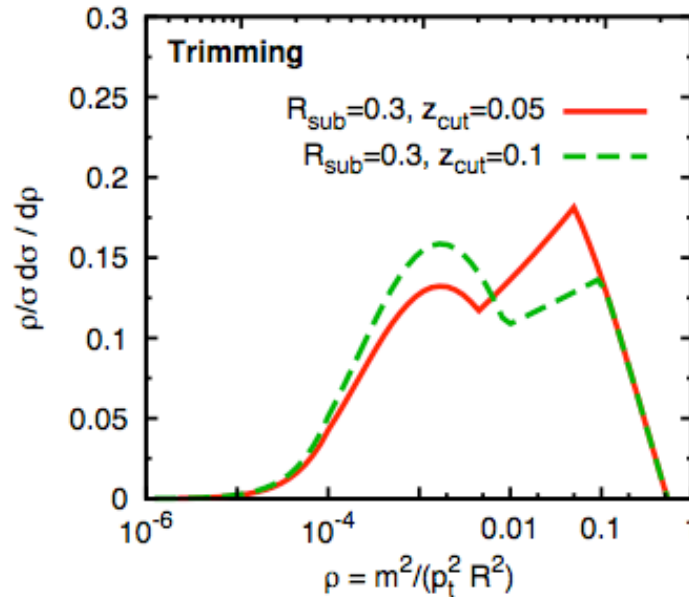
## Analytic

(resummed pQCD)

Analytic Calculation: gluon jets

$m$  [GeV], for  $p_t = 3$  TeV,  $R = 1$

10 100 1000



- Analytical understanding of ‘kinks’ in distributions
- Check of Monte Carlo predictions
- Other analytical investigations: Rubin 2010 (filtering), Walsh, Zuberi 2011 (jet substructure with SCET), Feige Schwartz, Stewart, Thaler 2012 (N-subjettiness), Dasgupta, Marzani, Powling 2013 (groomed jet mass), ...

$$\frac{1}{\sigma} \frac{d\sigma}{dm^2}^{(\text{trim}, \text{LO})} = \frac{\alpha_s C_F}{\pi} \int_0^1 dz p_{gq}(z) \int \frac{d\theta^2}{\theta^2} \delta(m^2 - z(1-z)p_t^2 \theta^2) \times$$

$$\times \left[ \Theta(z - z_{\text{cut}}) \Theta(1 - z - z_{\text{cut}}) \Theta(\theta^2 - R_{\text{sub}}^2) + \Theta(R_{\text{sub}}^2 - \theta^2) \right] \Theta(R^2 - \theta^2)$$

# Recap of Lecture 2

The big news of the past few years has been the development of 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 proven their worth in ‘Standard Model’ analyses. They are now being implemented in BSM searches