iSTEP Summer School
Tsinghua University - Beijing
July 2016

Jet Reconstruction

Jet algorithms and jet substructure

Matteo Cacciari
LPTHE Paris
Université Paris Diderot

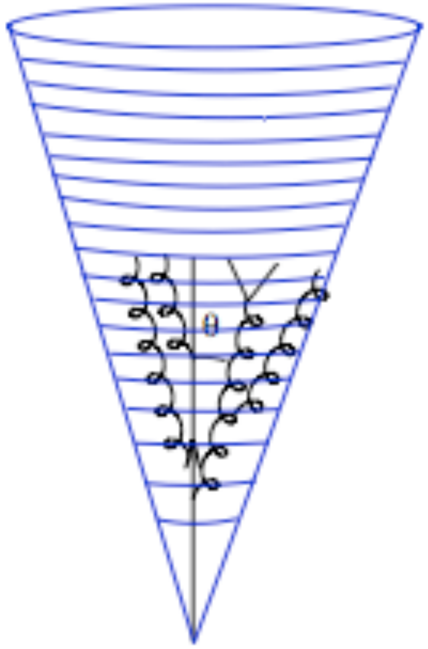
Includes material from
Gavin Salam and Grégory Soyez

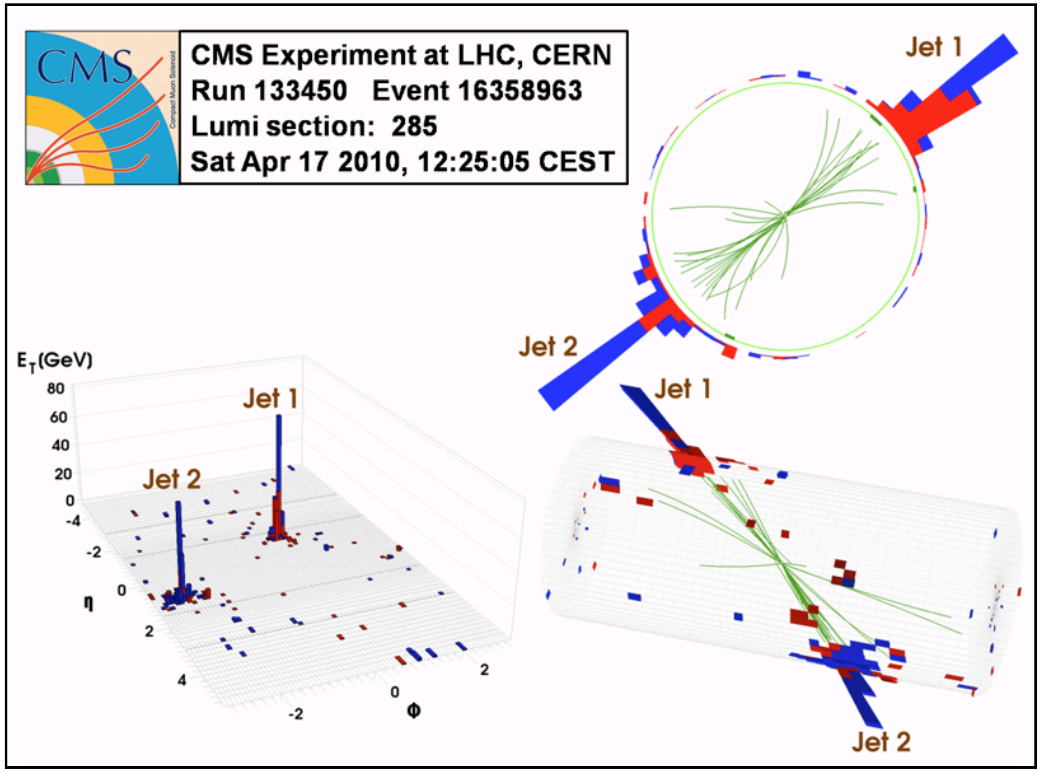
Outline

- Jet algorithms
  - How are jets made
- Jet substructure
  - What's inside them

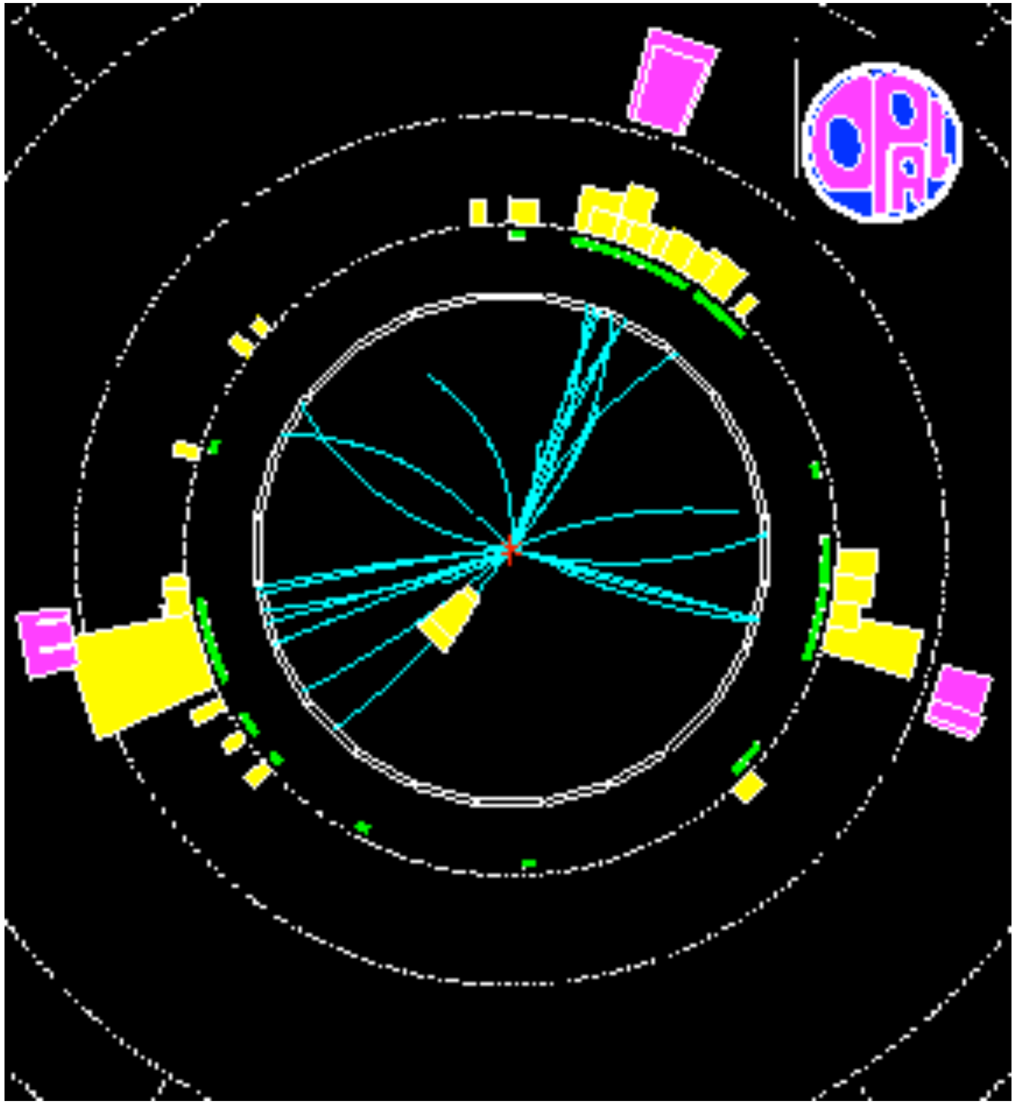
What is a jet?


AIRFRANCE




CMS
CMS Experiment at LHC, CERN
Run 133450 Event 16358963
Lumi section: 285
Sat Apr 17 2010, 12:25:05 CEST
Jet 1
Jet 2
Jet 1
Jet 2
Jet 1
Jet 2
E_T(GeV)
80
60
40
20
0
-4
-2
0
2
4
η
-2
0
2
Φ

Why jets

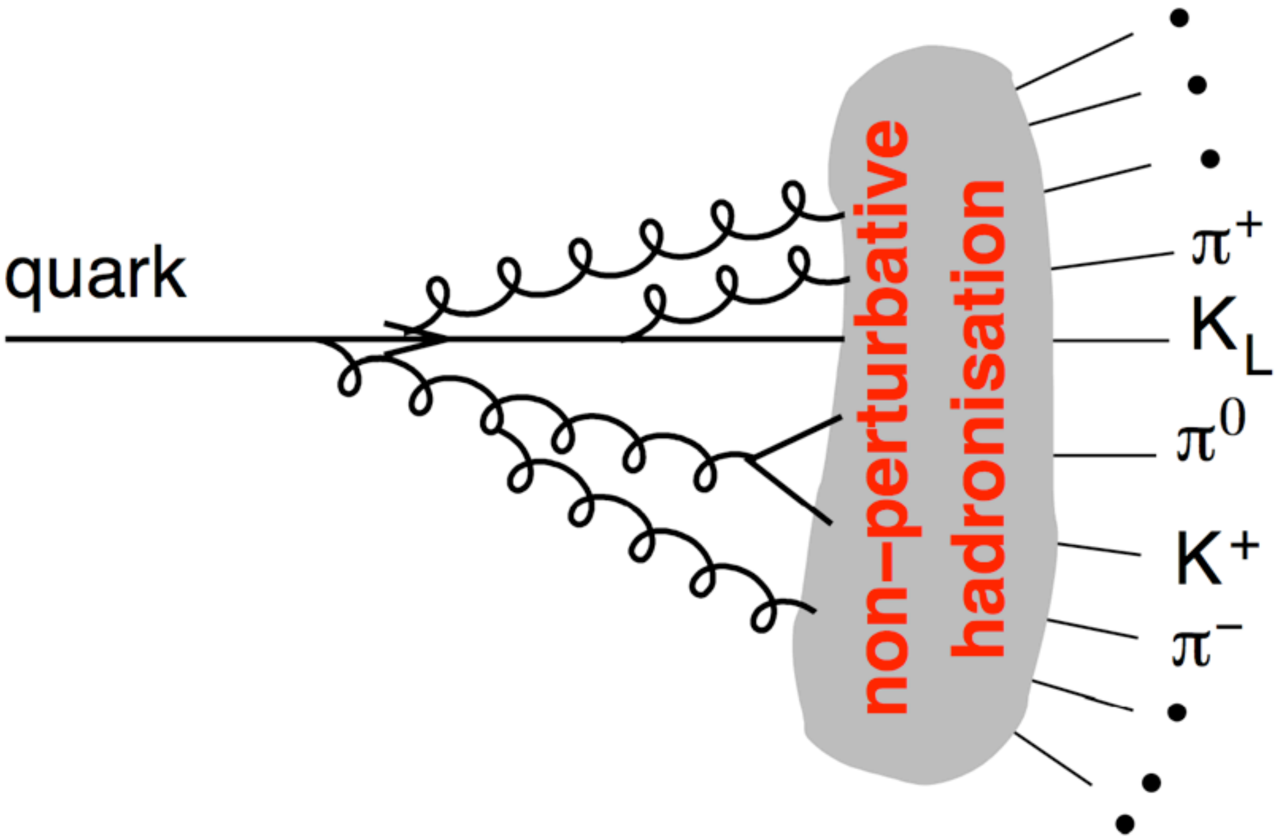


A **jet** is something that happens in high energy events: **a collimated bunch of hadrons flying roughly in the same direction**

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**

Why do jets happen?



Gluon emission

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

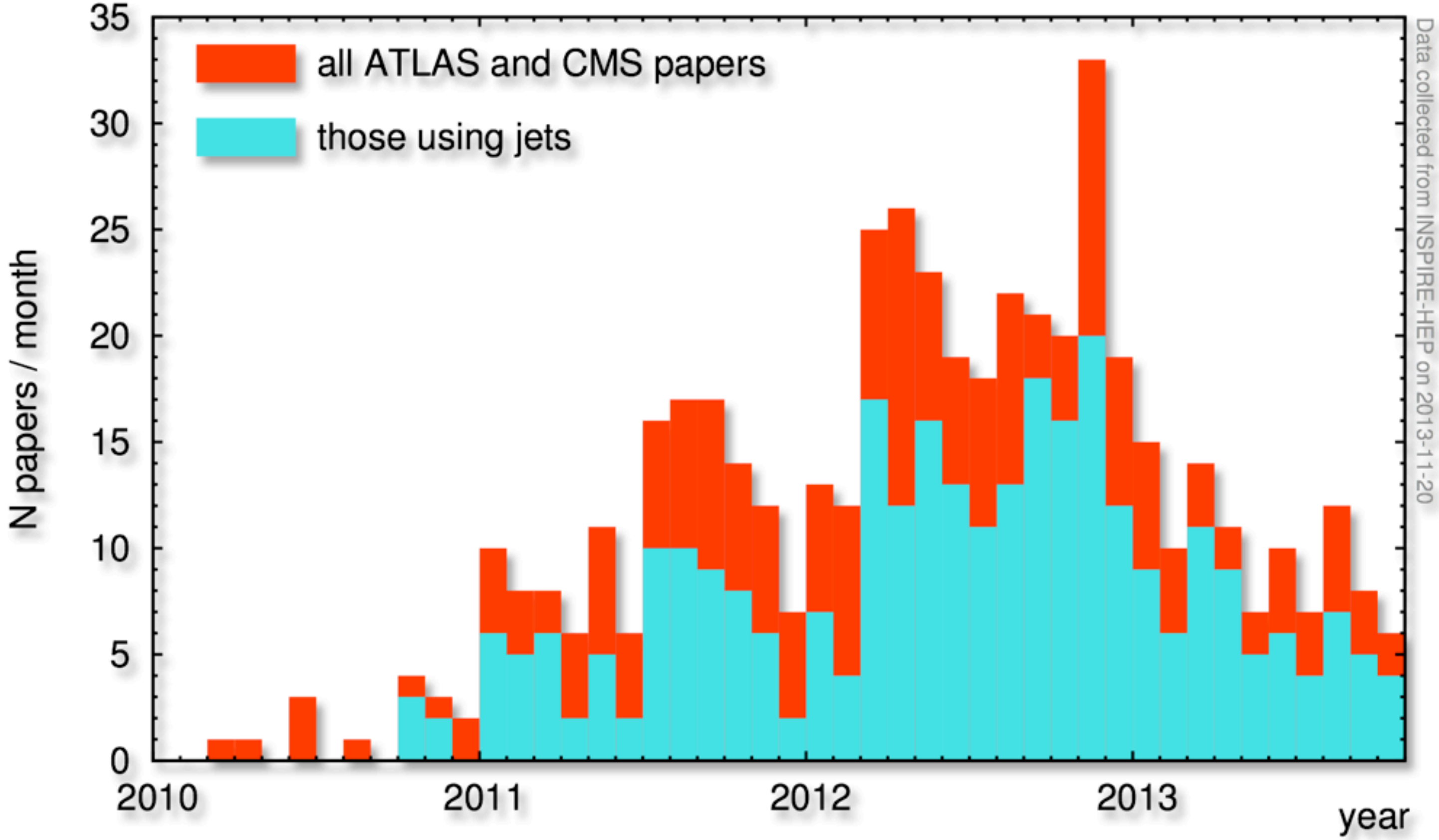
Non-perturbative physics

$$\alpha_s \sim 1$$

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**

Plot by G. Salam

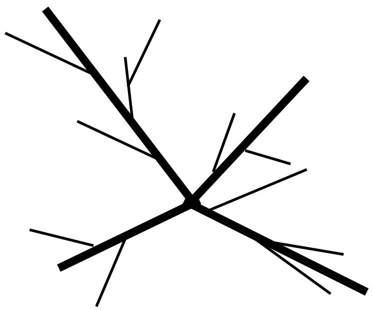


(Source: INSPIRE. Results may vary when employing different search keywords)

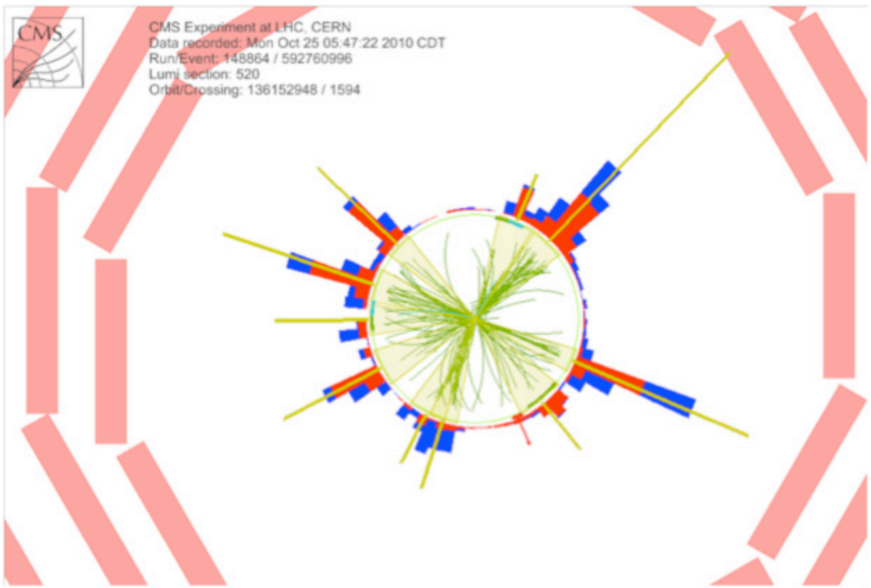
Why are jets so important?

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**

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

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

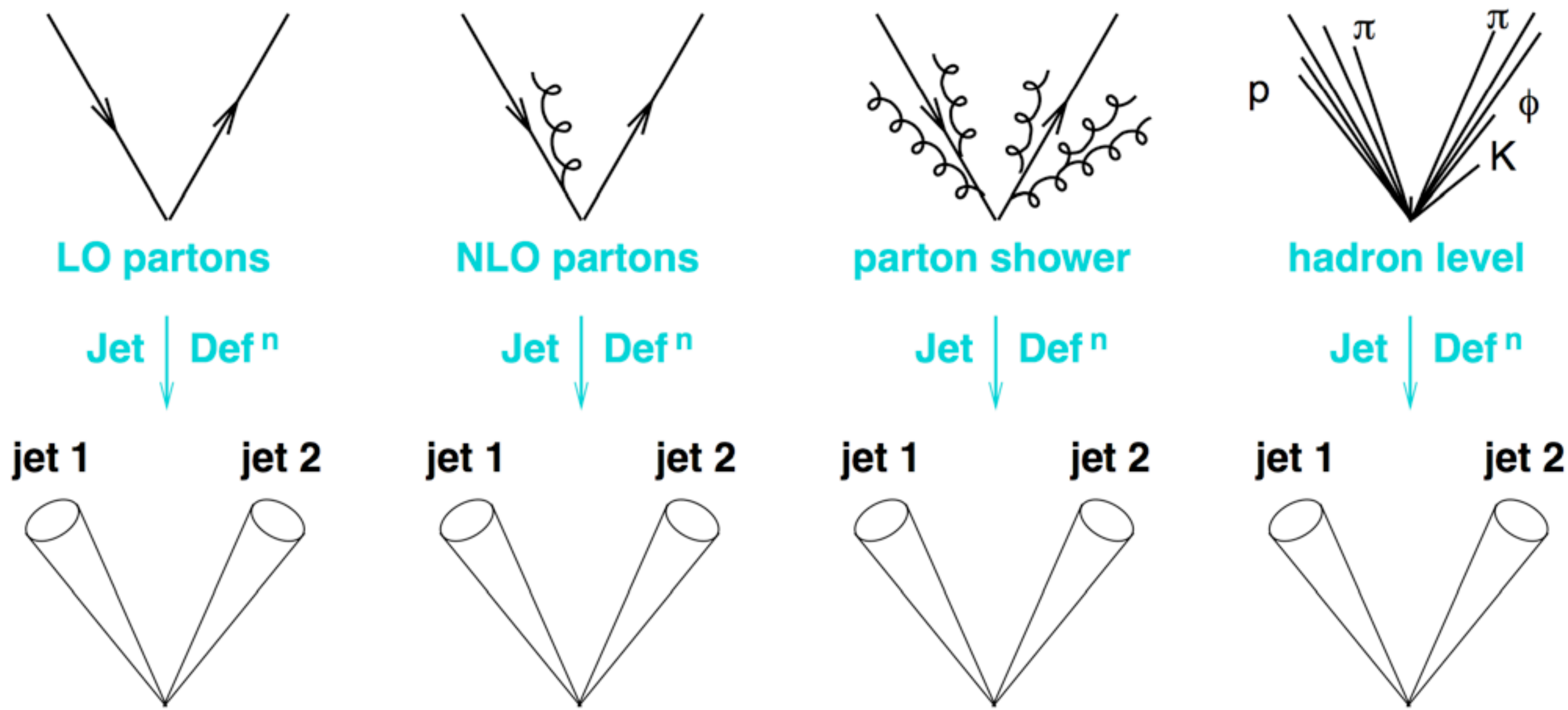
Most algorithms contain a resolution parameter, **R**, which controls the extension of the jet

Jet Definition

A jet algorithm
+
its parameters (e.g. R)
+
a recombination scheme
=
a **Jet Definition**

```
/// define a jet definition
JetDefinition jet_def(JetAlgorithm jet_algorithm,
                      double R,
                      RecombinationScheme rec_sch = E_scheme);
```

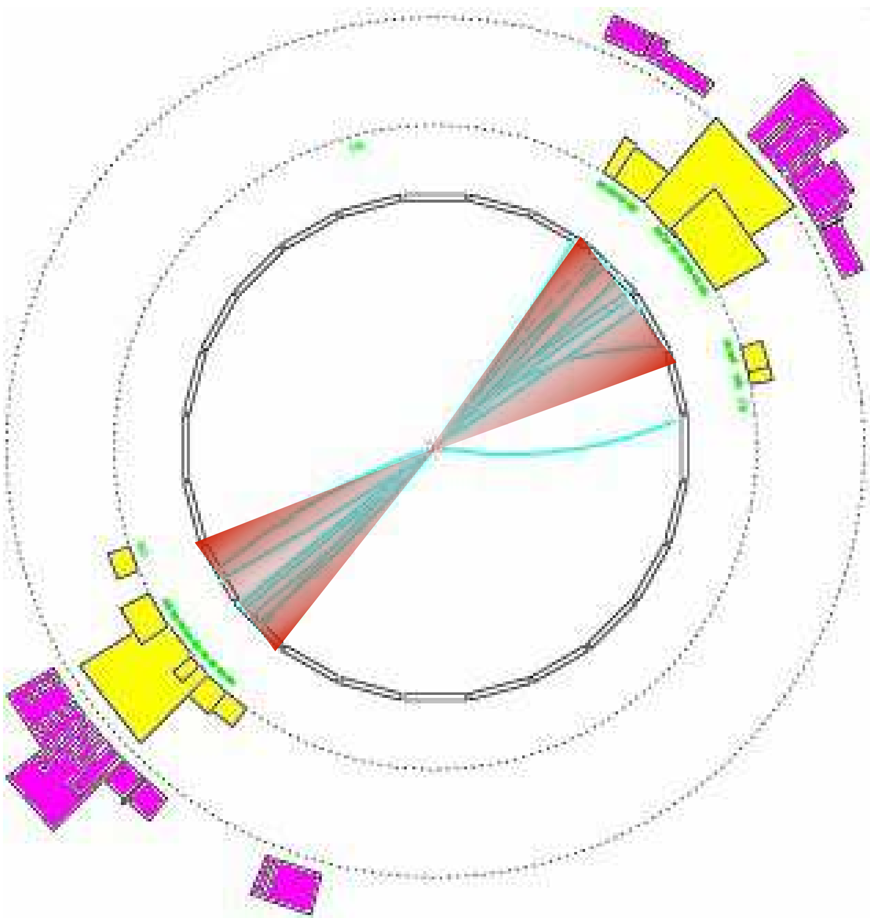
Jet definitions as projections



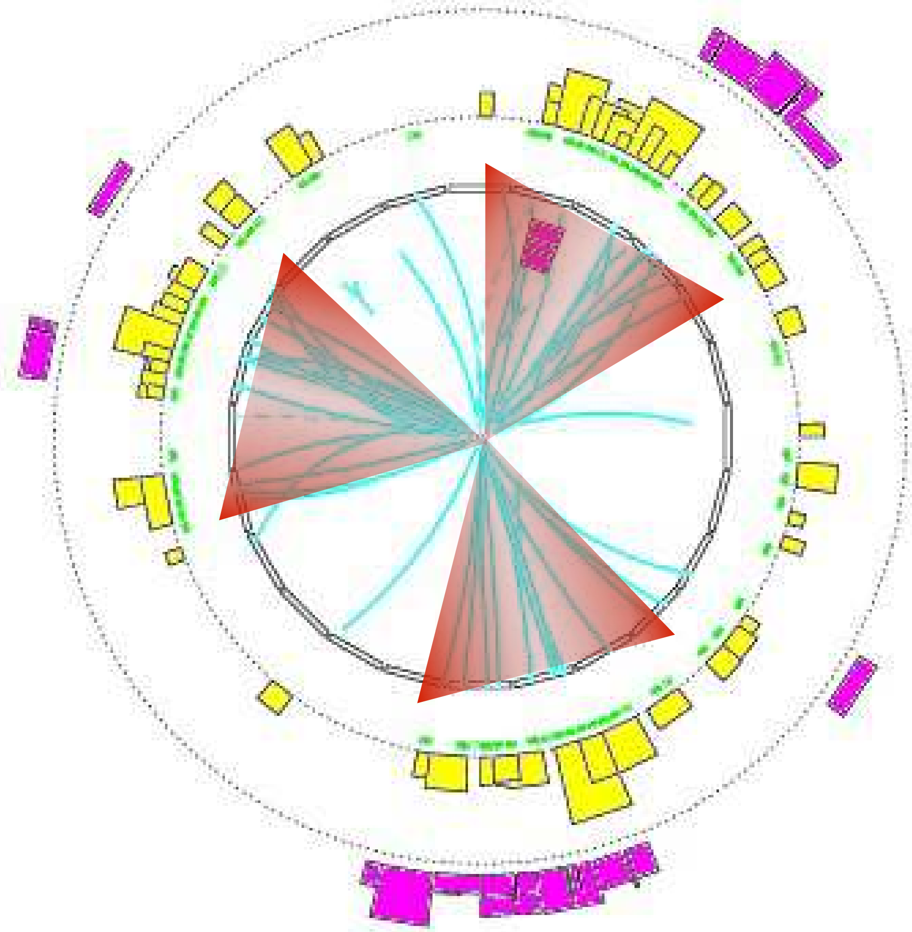
Projection to jets should be resilient to QCD effects

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

Reconstructing jets is an ambiguous task

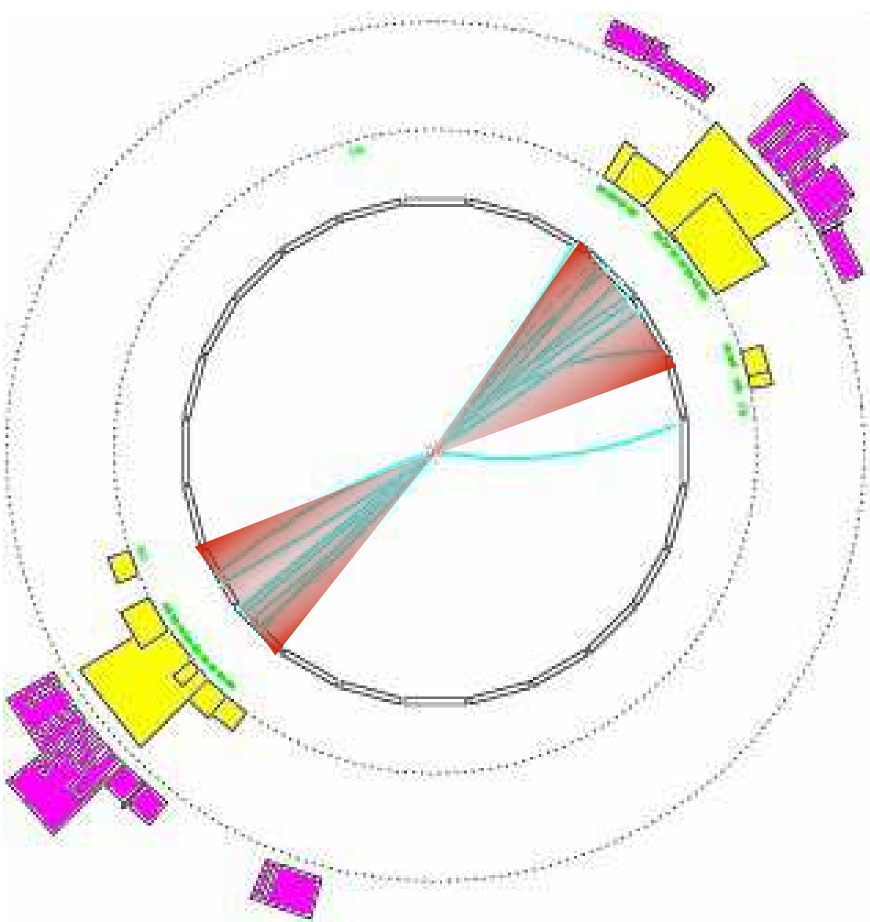


2 clear jets

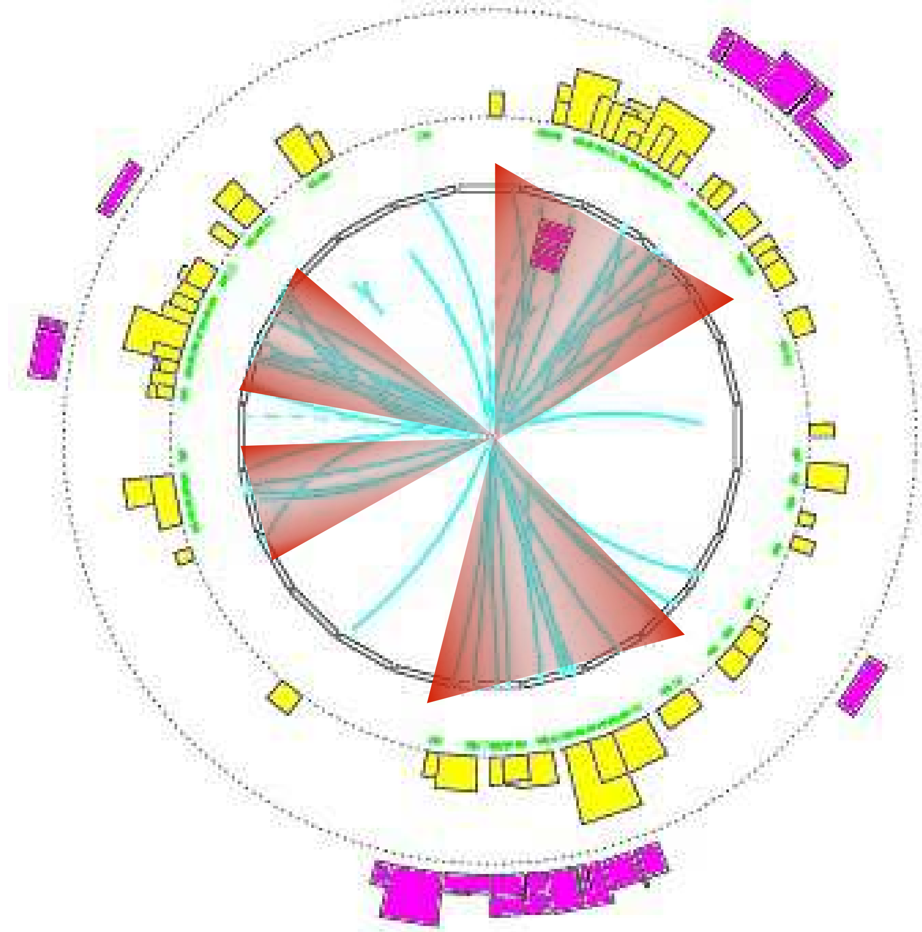


3 jets?

Reconstructing jets is an ambiguous task



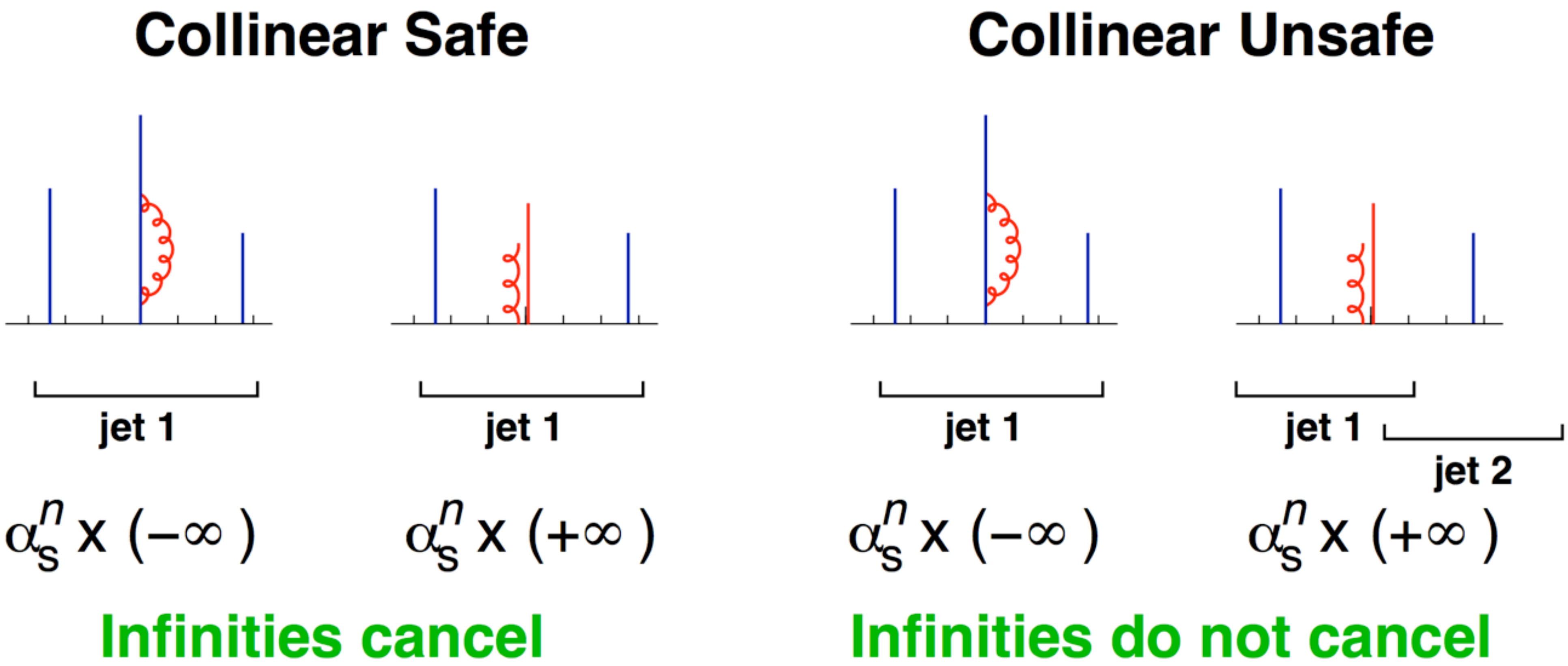
2 clear jets



3 jets?

or 4 jets?

Reconstructing jets must respect rules



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

IRC safety

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

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

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

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

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

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, …..

- ## 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…...

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

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, ...

e^+e^- k_t (Durham) algorithm

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

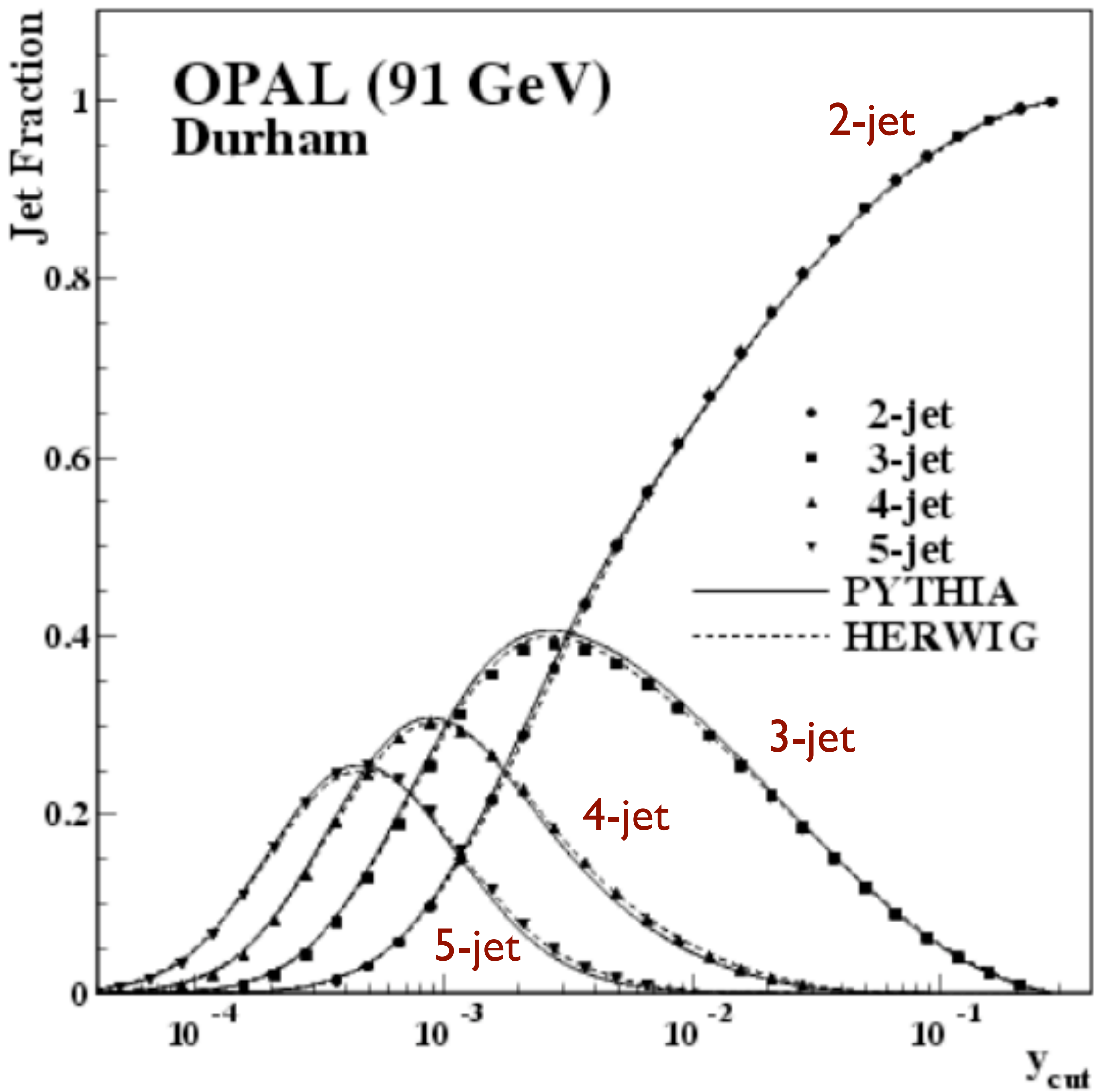
Distance:

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

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

Jet challenges at the LHC

The LHC environment differs from the LEP one (and even the Tevatron) under many respects

- Number of final state particles much larger (order 10^3)

  → needs a fast algorithm

- Many higher order calculations (NLO, NNLO) available

  → needs an IRC-safe algorithm

- Presence of background (underlying event and pileup)

  → needs small/known susceptibility and/or ability to subtract background

- Jets often initiated by a large-momentum heavy particle

  → needs capability to distinguish boosted objet jet from QCD jet

Two parameters, *R* and $p_{t,min}$
(These are the two parameters in essentially every widely used hadron-collider jet algorithm)

$$d_{ij} = \min(p_{ti}^2, p_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad \Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$$

Sequential recombination algorithm

1. Find smallest of d_{ij}, d_{iB}
2. If *ij*, recombine them
3. If *iB*, call i a jet and remove from list of particles
4. repeat from step 1 until no particles left

Only use jets with $p_t > p_{t,min}$

Inclusive k_t algorithm
S.D. Ellis & Soper, 1993
Catani, Dokshitzer, Seymour & Webber, 1993

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 → 0 ⇒ clustered first, no effect on jets

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

p = 0

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

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

p < 0

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

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

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)

Jet clustering in FastJet

```
/// define a jet definition
JetDefinition jet_def(JetAlgorithm jet_algorithm,
                      double R,
                      RecombinationScheme rec_sch = E_scheme);
```

jet_algorithm can be any one of the four IRC safe algorithms, or also most of the old IRC-unsafe ones, for legacy purposes

```
/// create a ClusterSequence, extract the jets
ClusterSequence cs(input_particles, jet_def);
vector<PseudoJet> jets = sorted_by_pt(cs.inclusive(jets));
...
// pt of hardest jet
double pt_hardest = jets[0].pt();
...
// constituents of hardest jet
vector<PseudoJet> constits = jets[0].constituents();
```

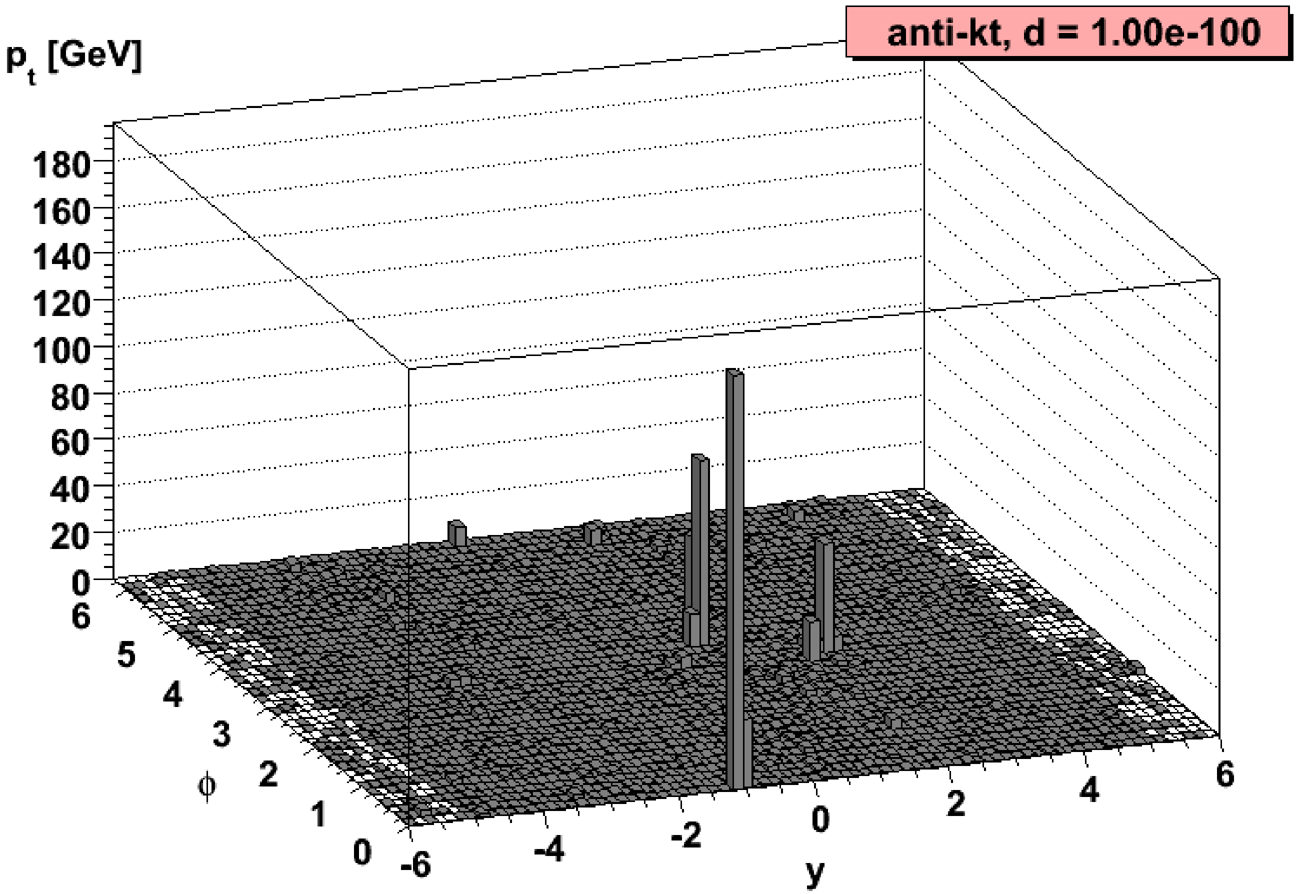
Clustering grows around hard cores

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

Anti-k_t in action

Clustering grows around hard cores

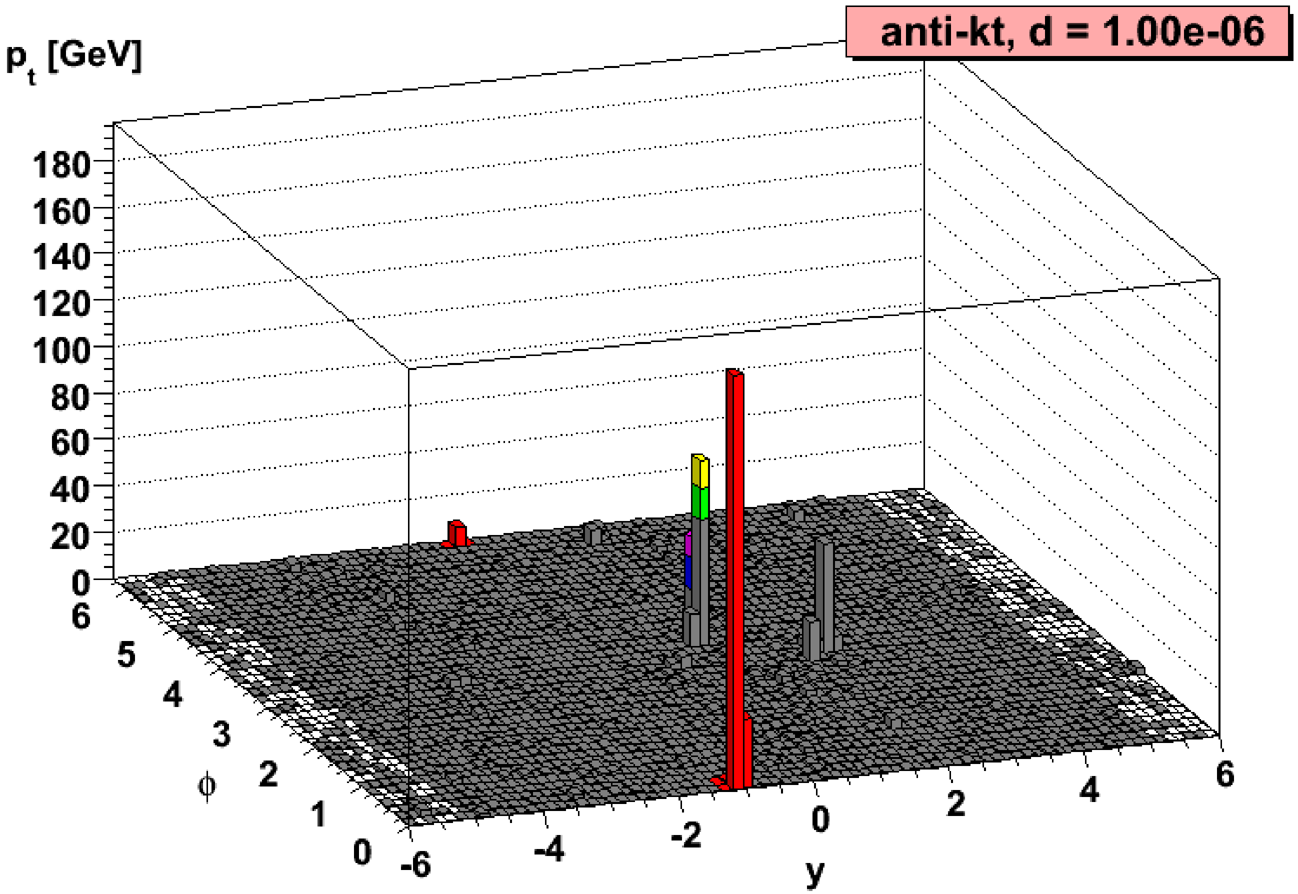
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

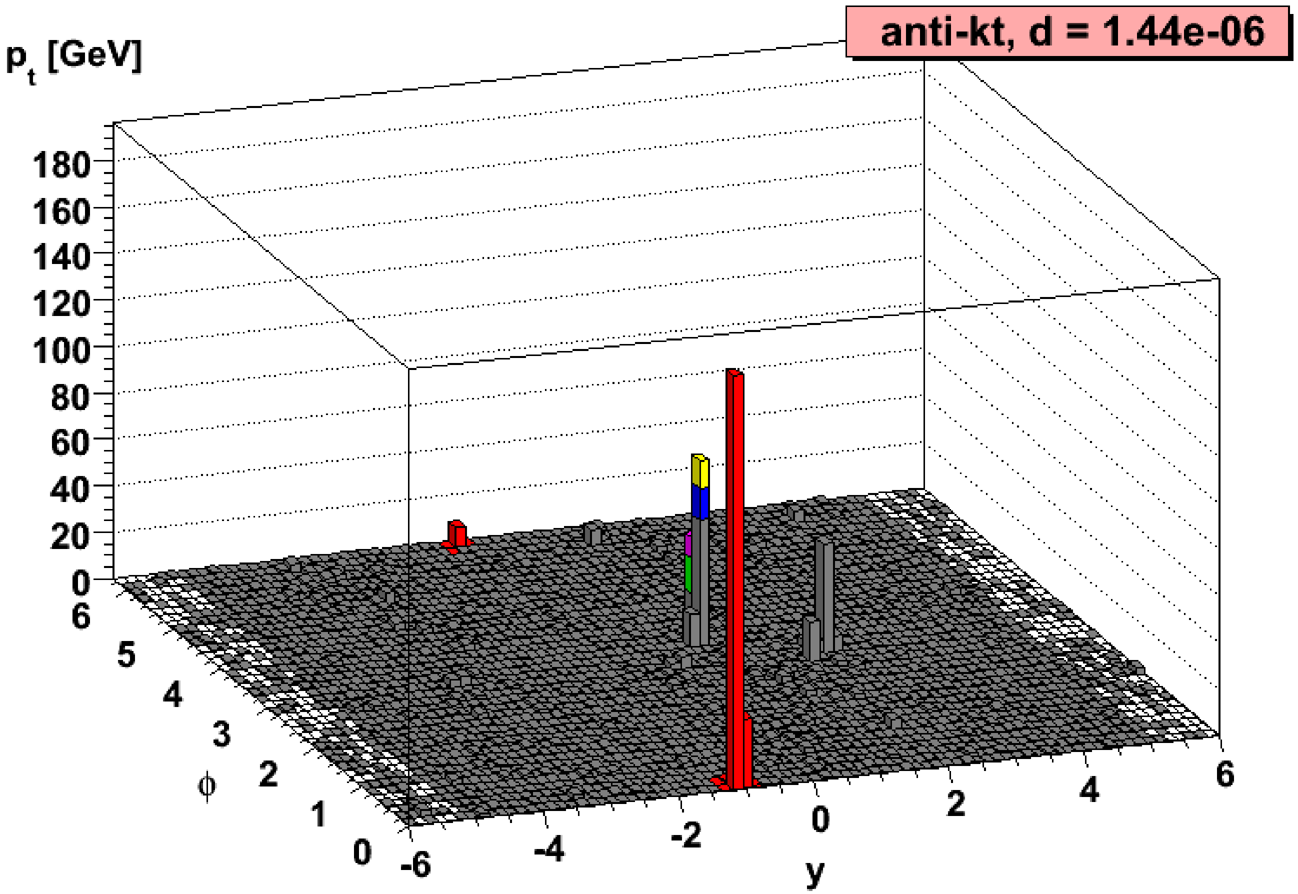
Clustering grows around hard cores

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



Clustering grows around hard cores

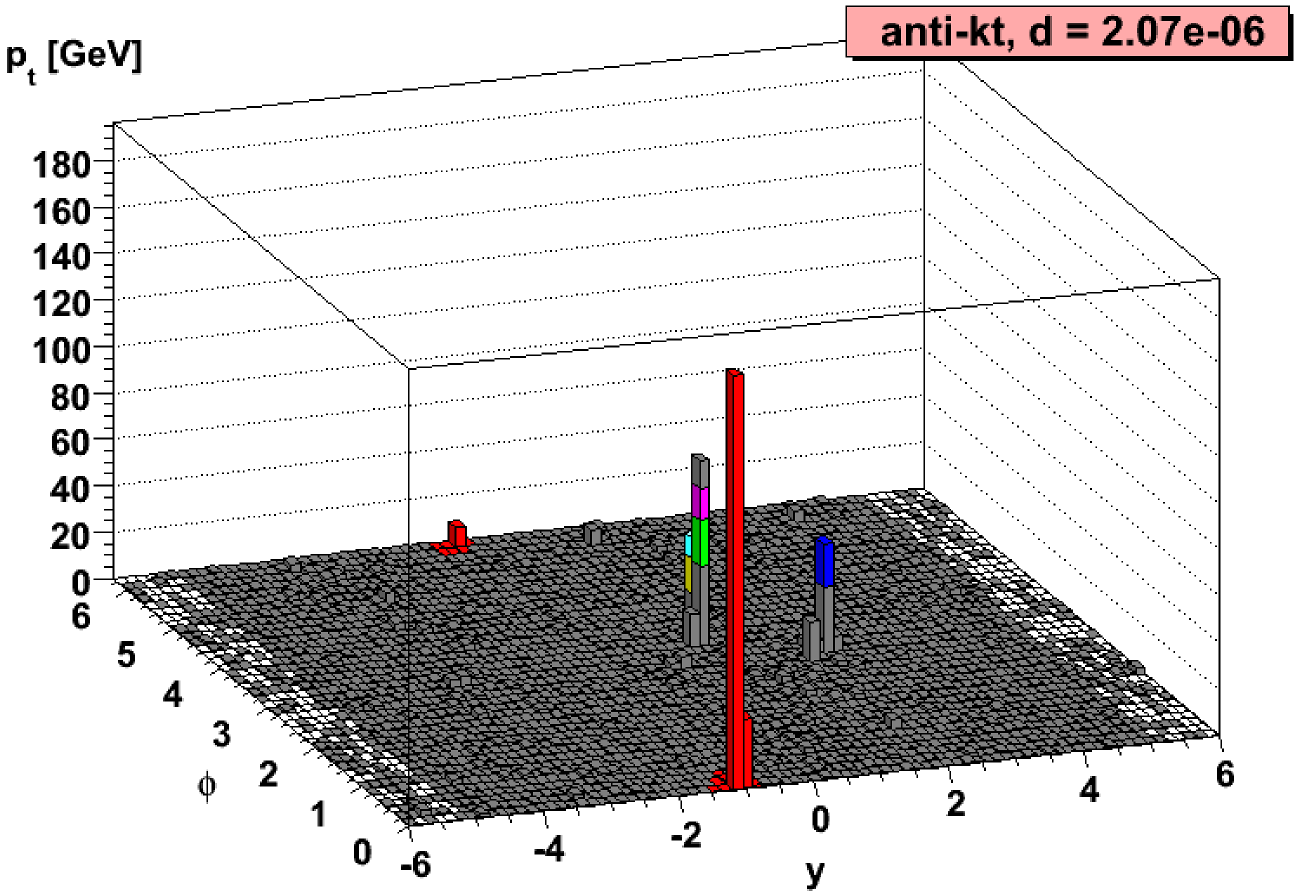
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

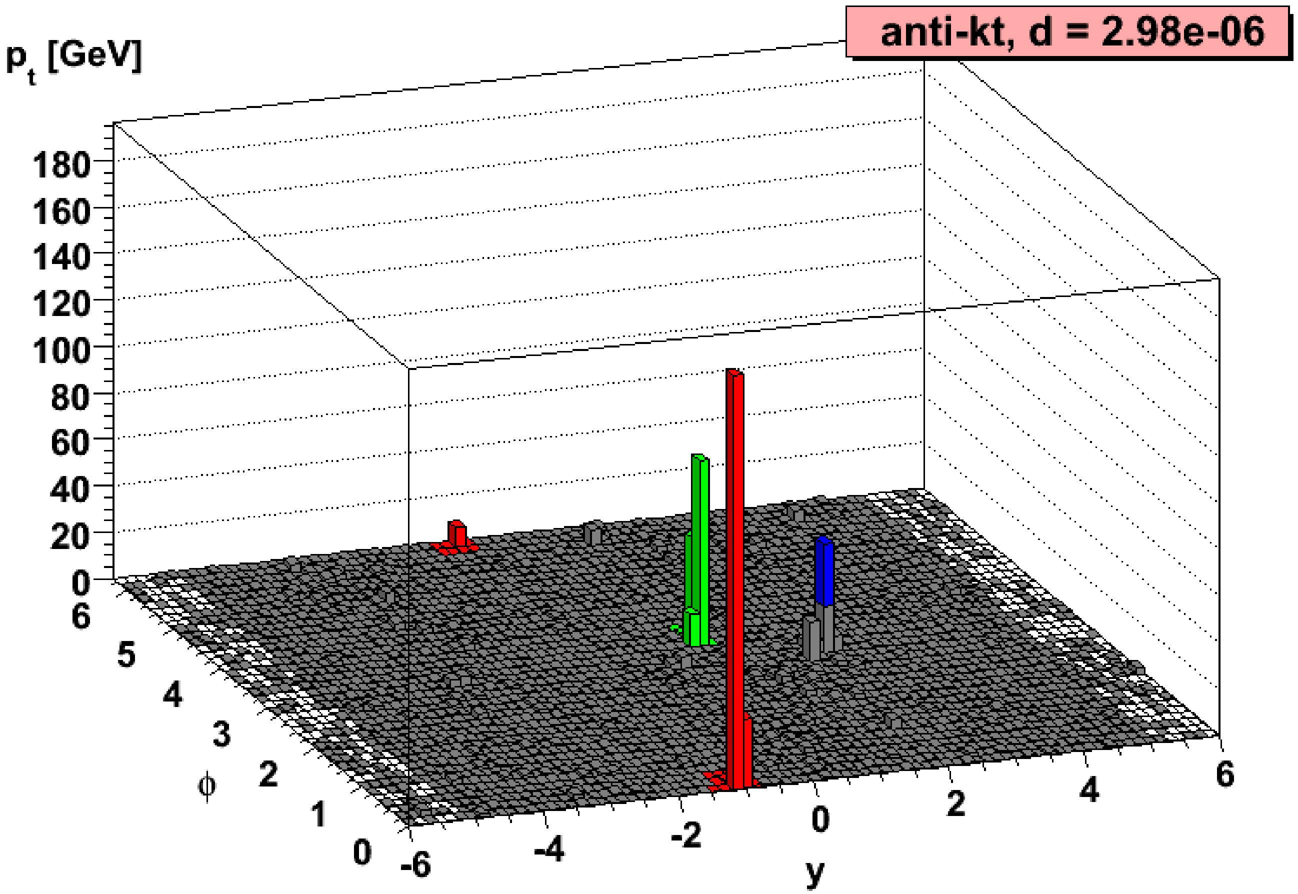
Clustering grows around hard cores

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



Clustering grows around hard cores

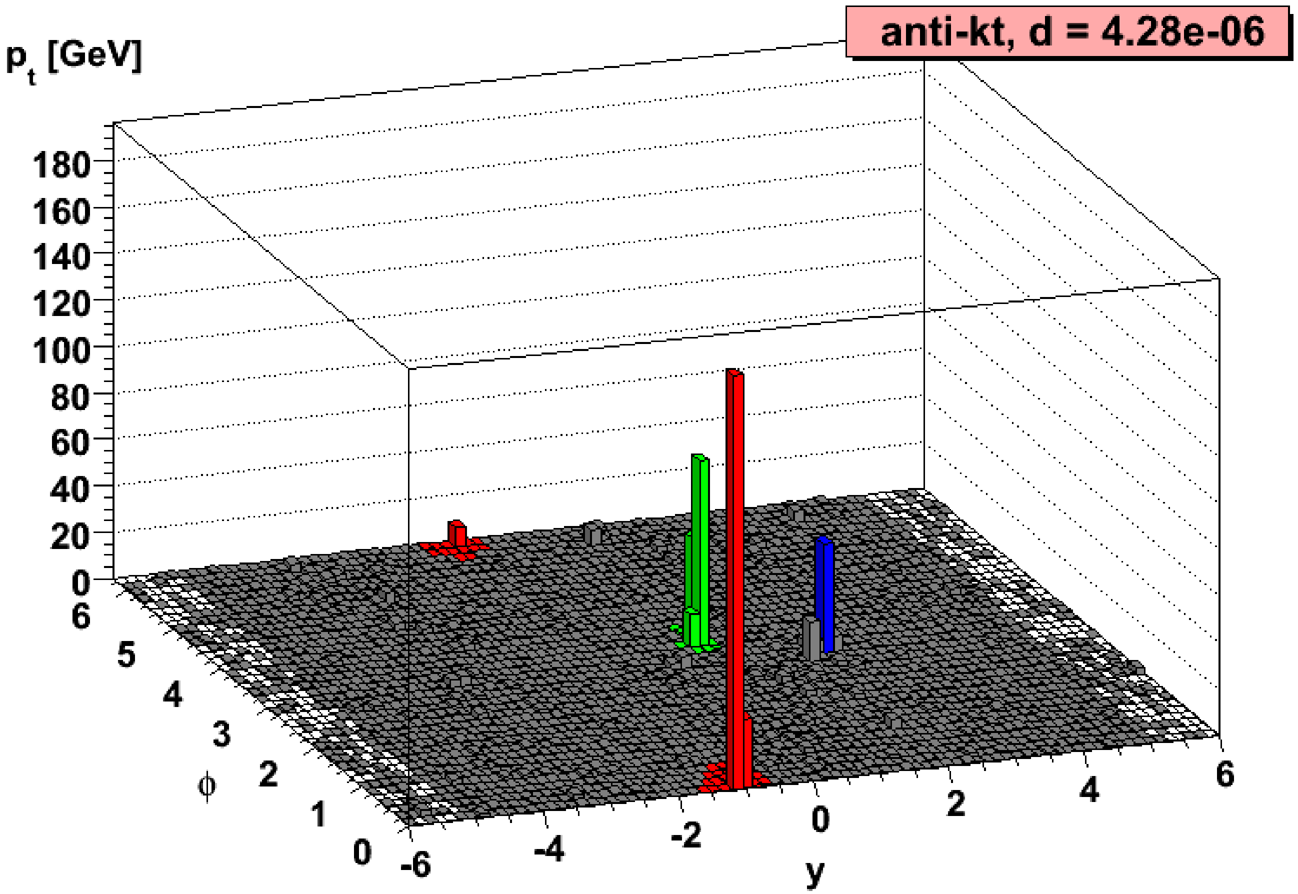
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

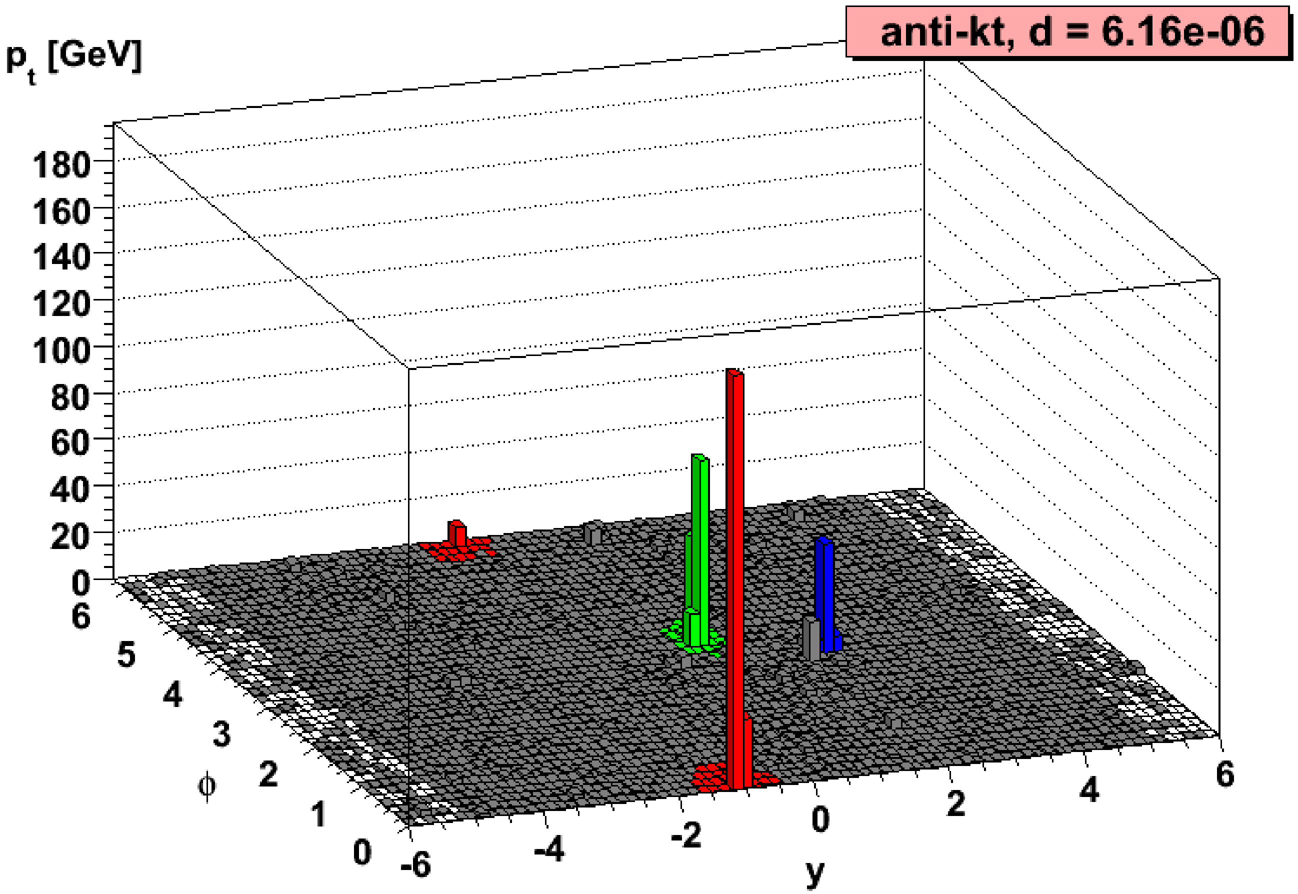
Clustering grows around hard cores

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



Clustering grows around hard cores

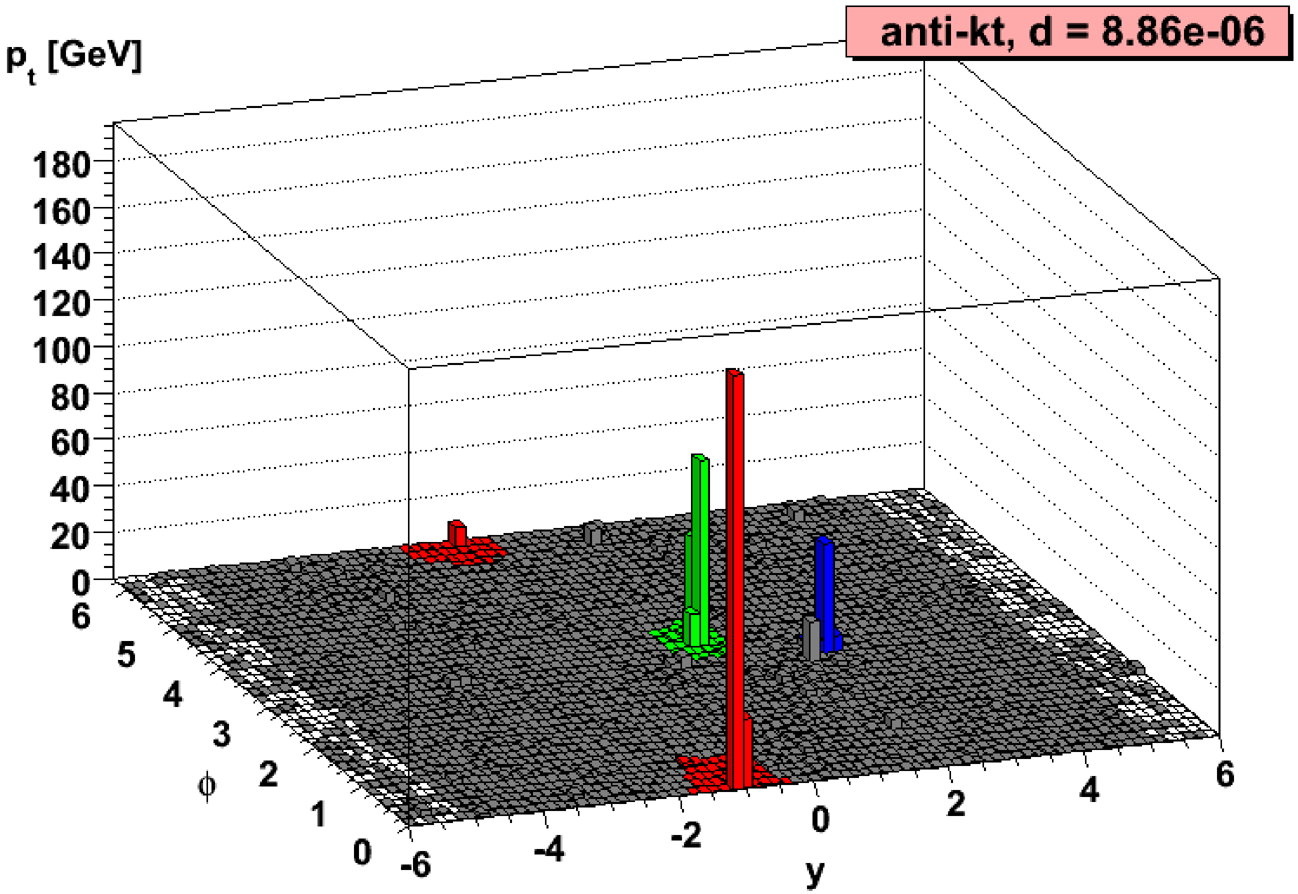
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

Clustering grows around hard cores

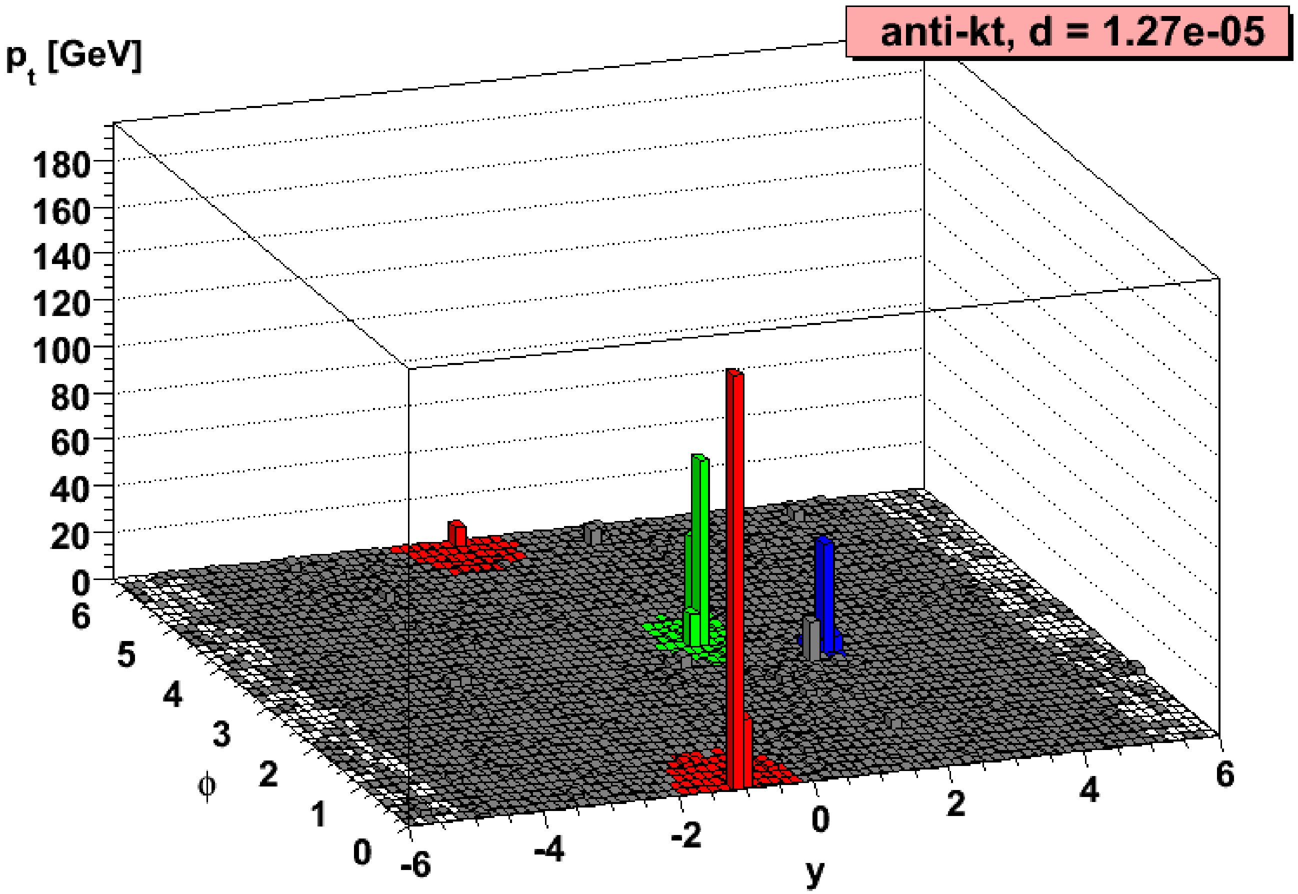
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

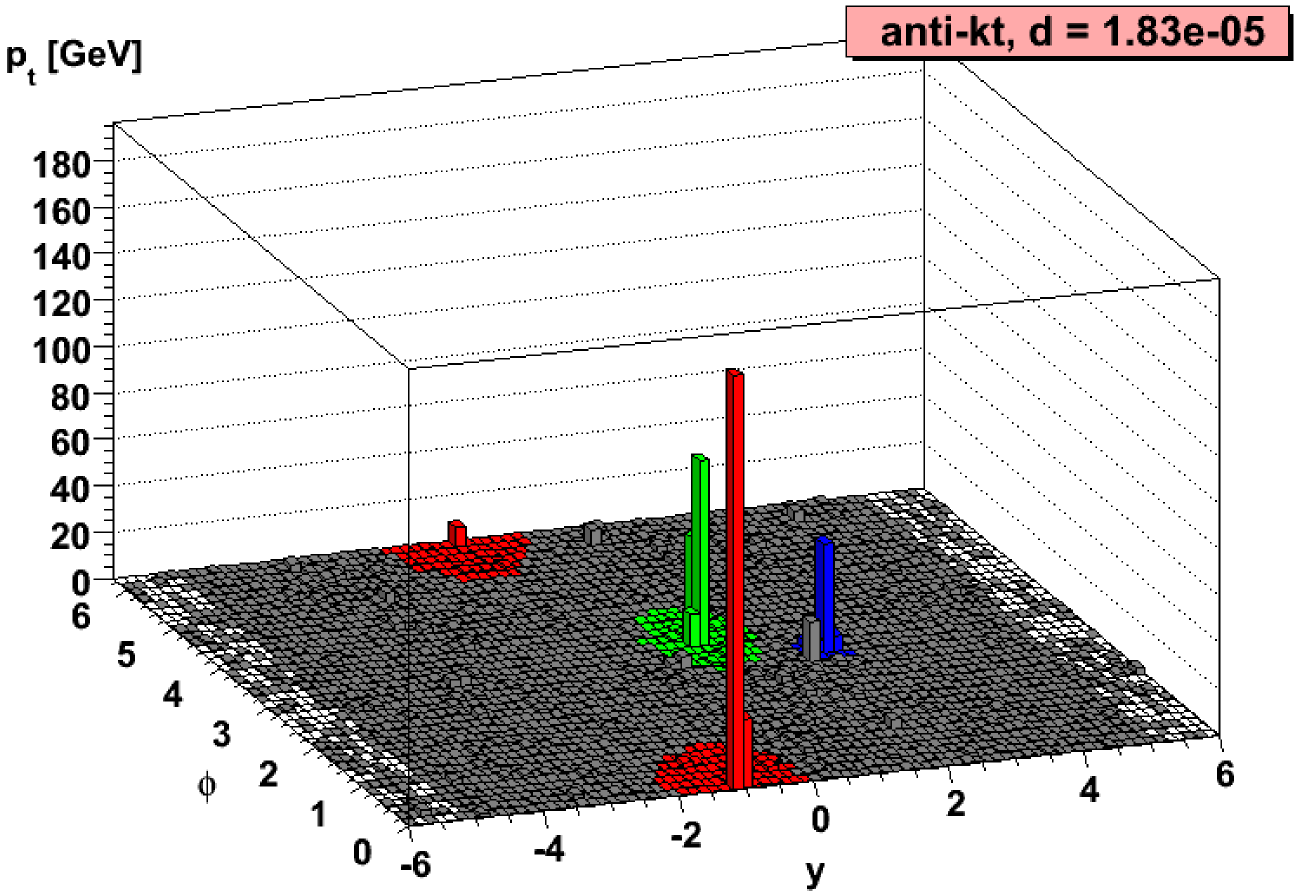
Clustering grows around hard cores

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



Clustering grows around hard cores

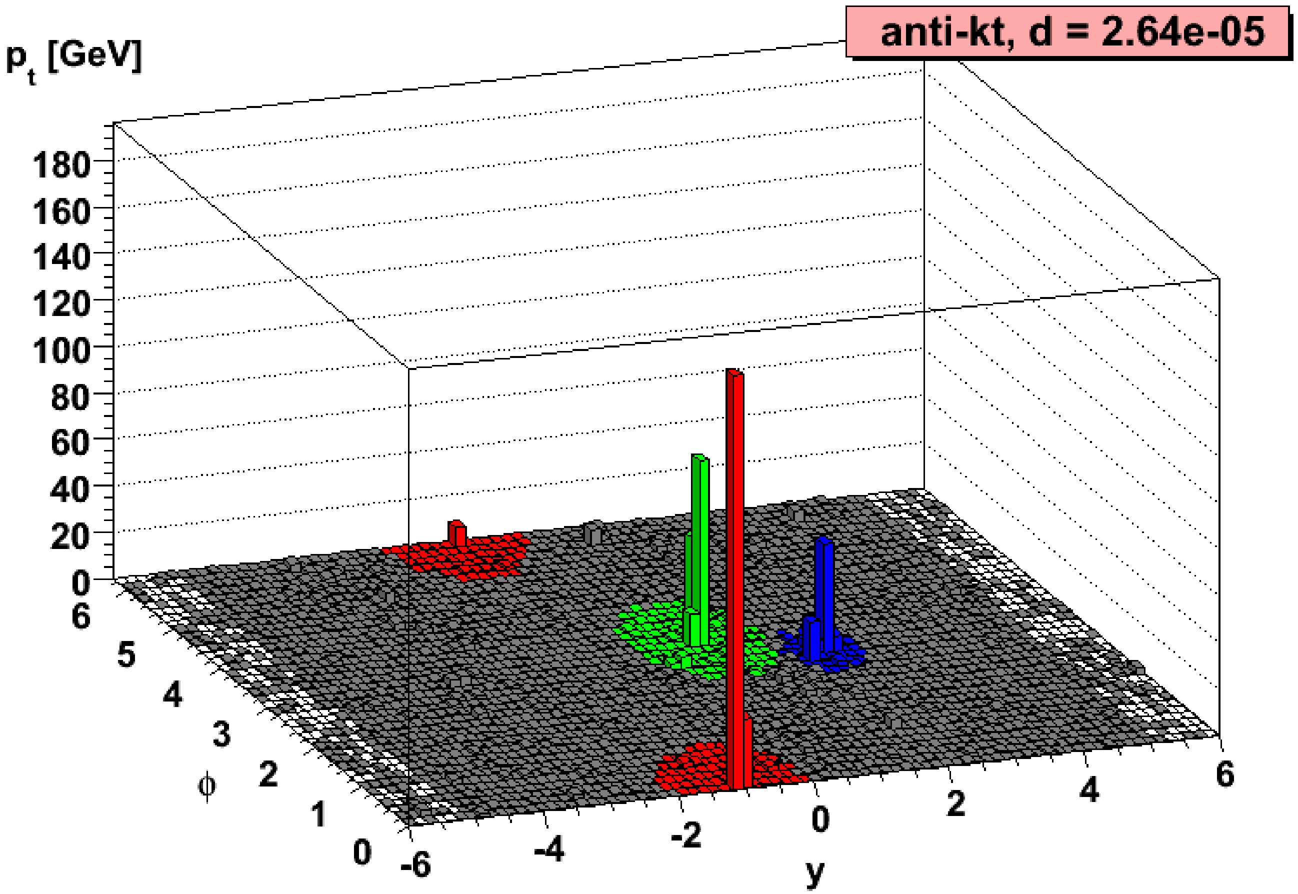
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

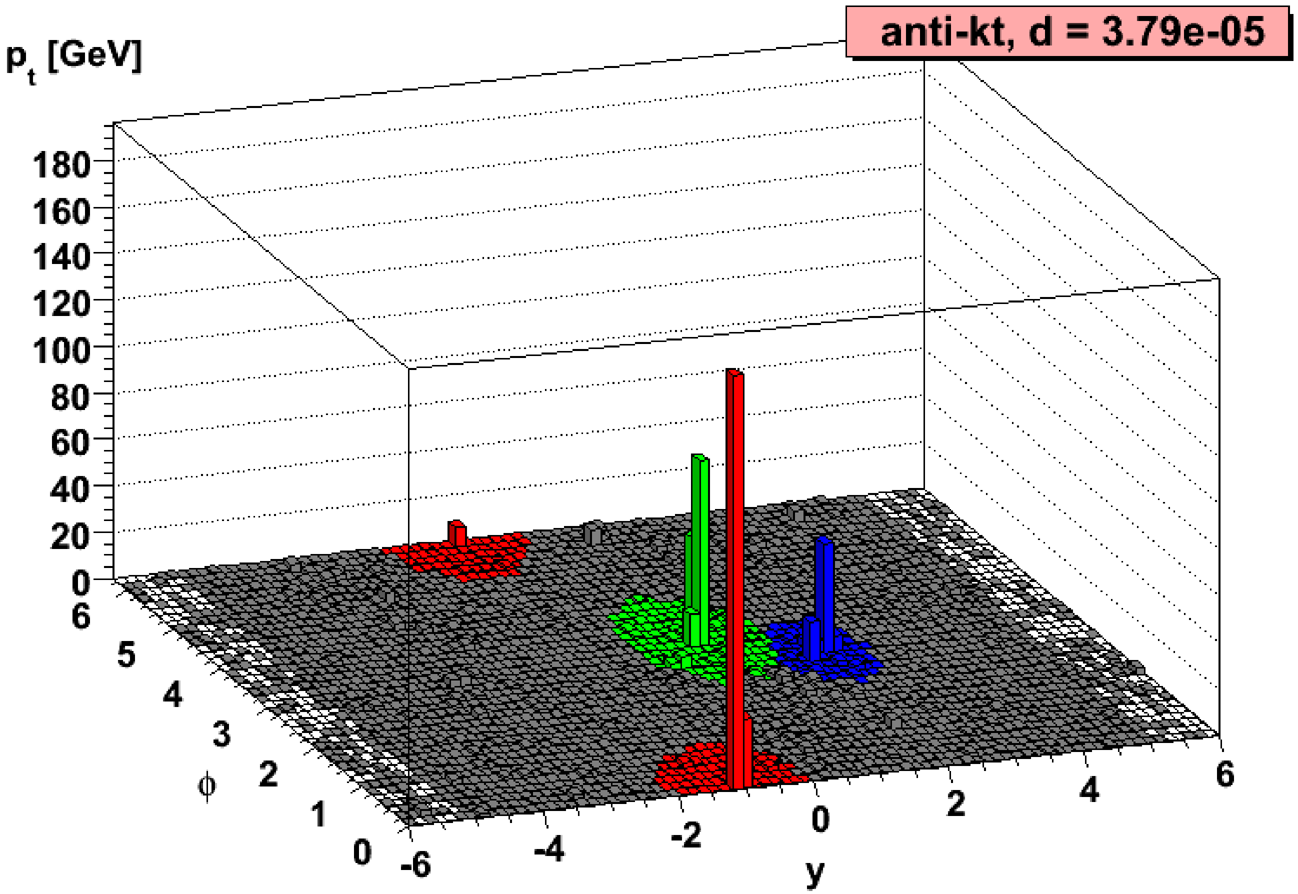
Clustering grows around hard cores

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



Clustering grows around hard cores

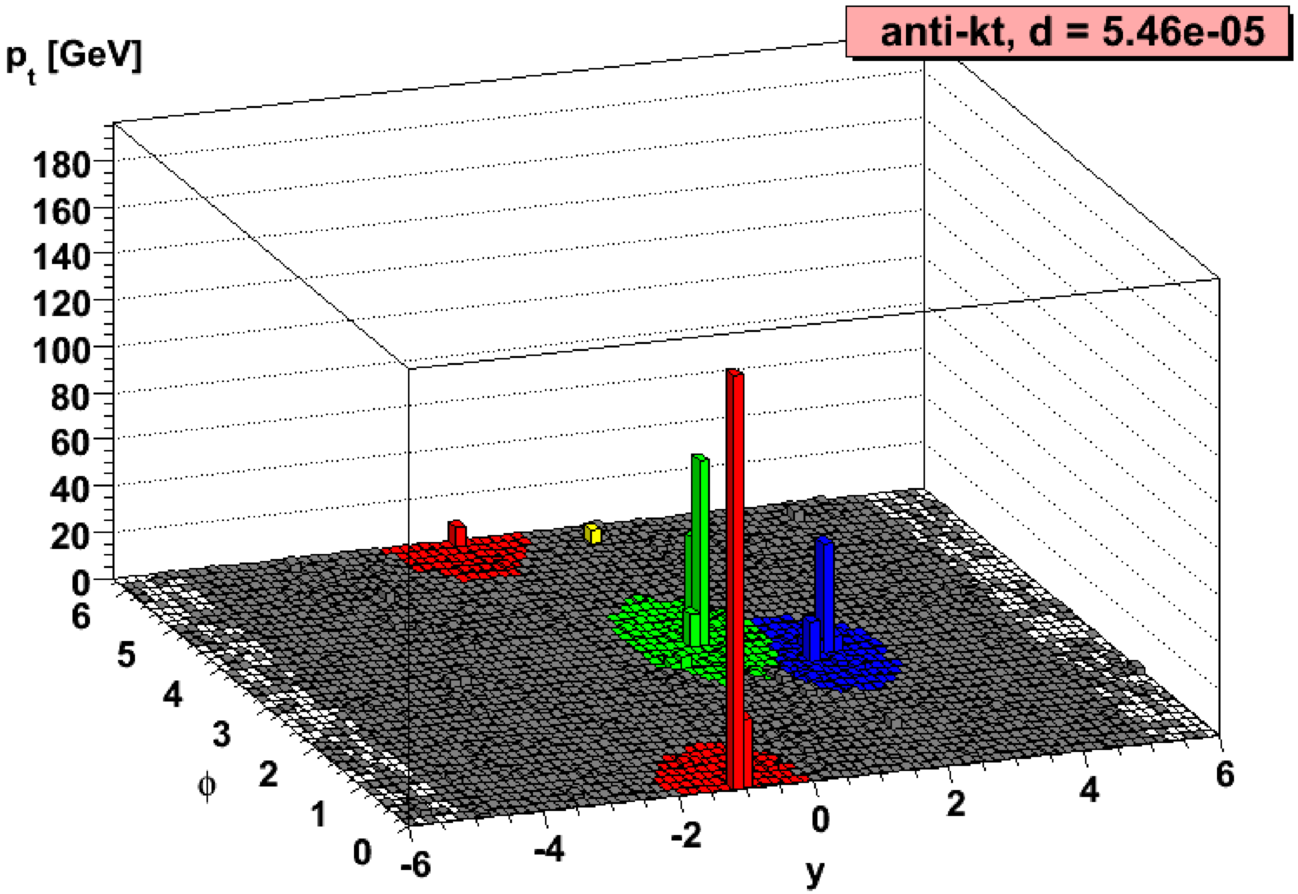
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

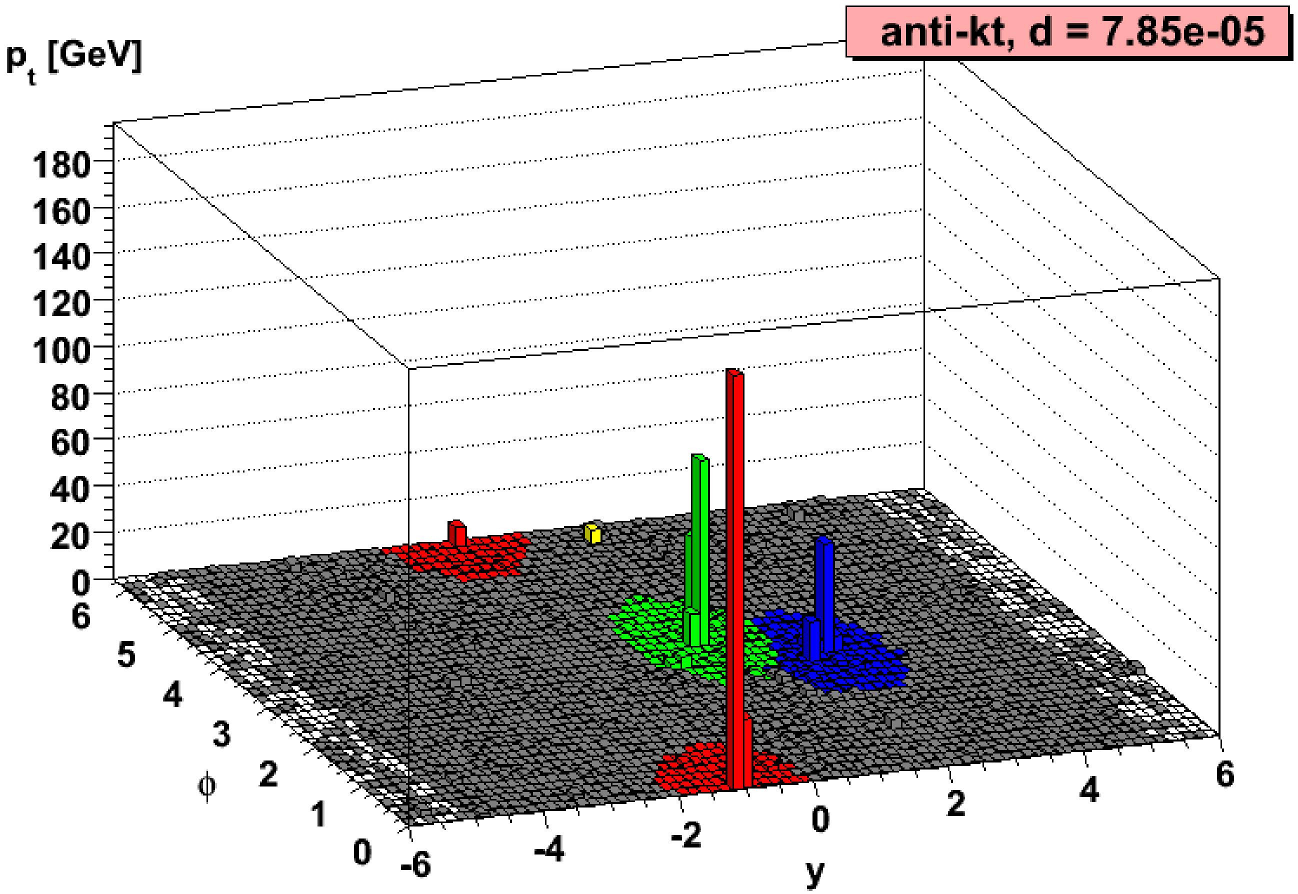
Clustering grows around hard cores

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



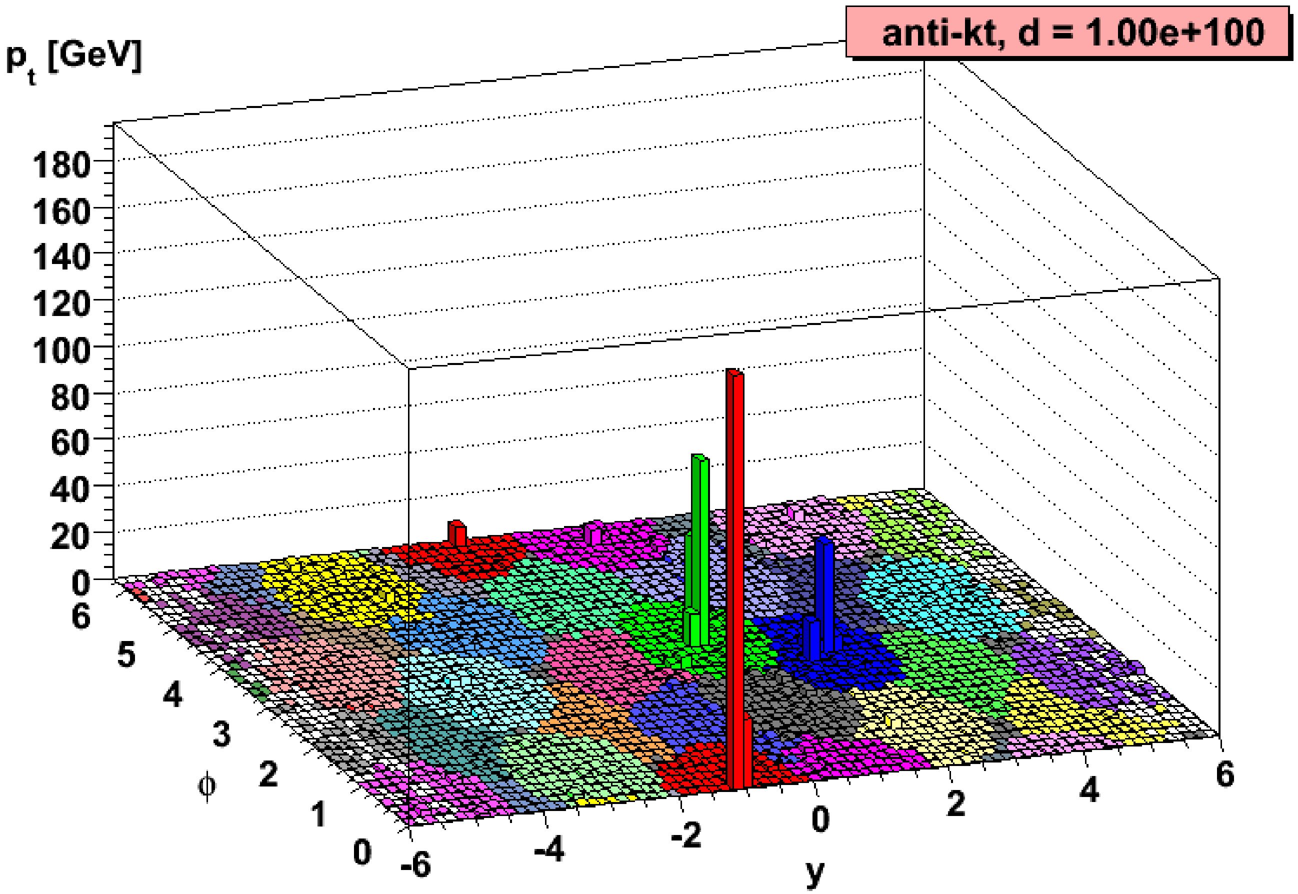
Clustering grows around hard cores

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

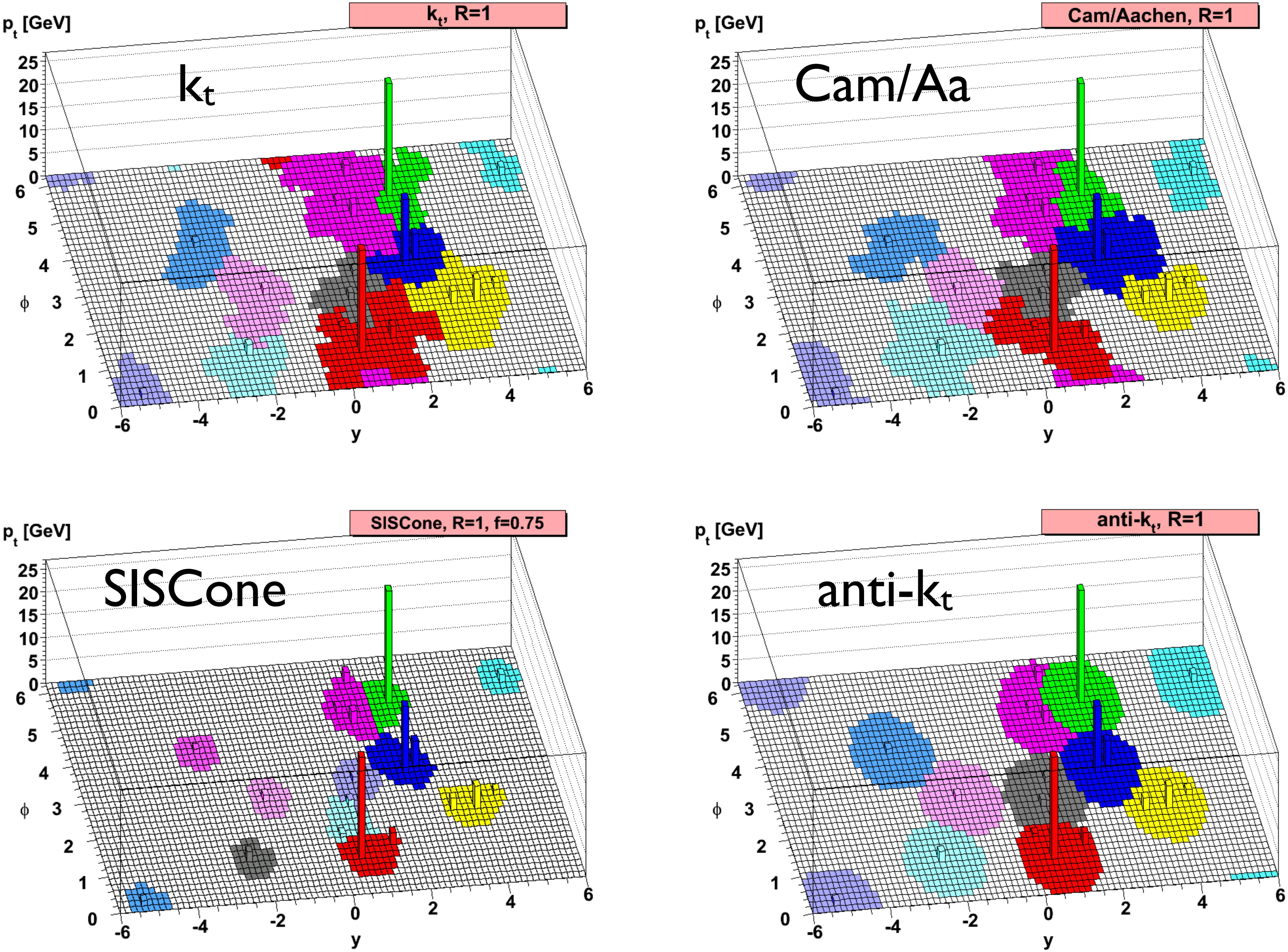


Clustering grows around hard cores

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

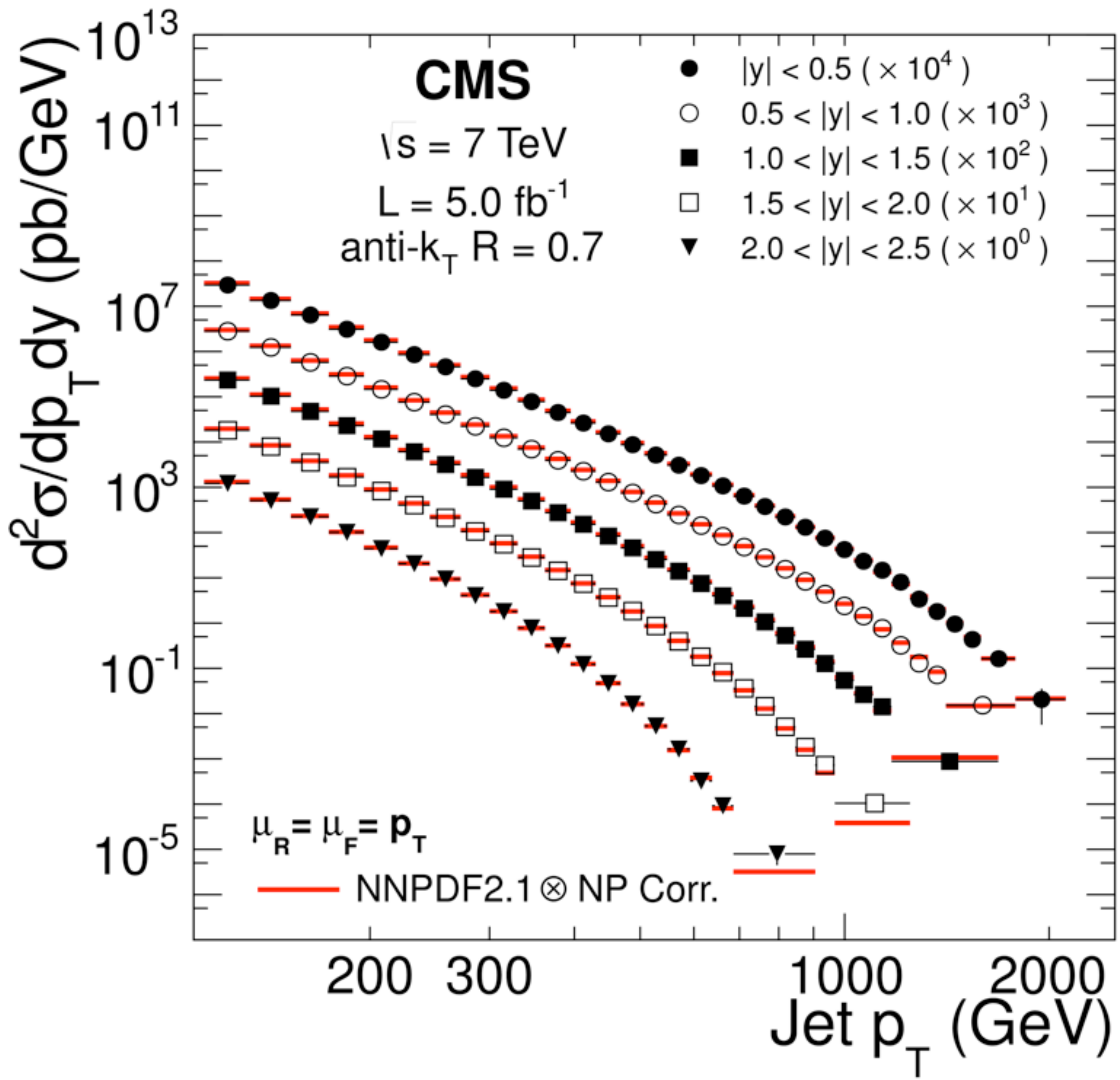


Anti-k_t gives circular jets ("cone-like") in a way that's infrared safe



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

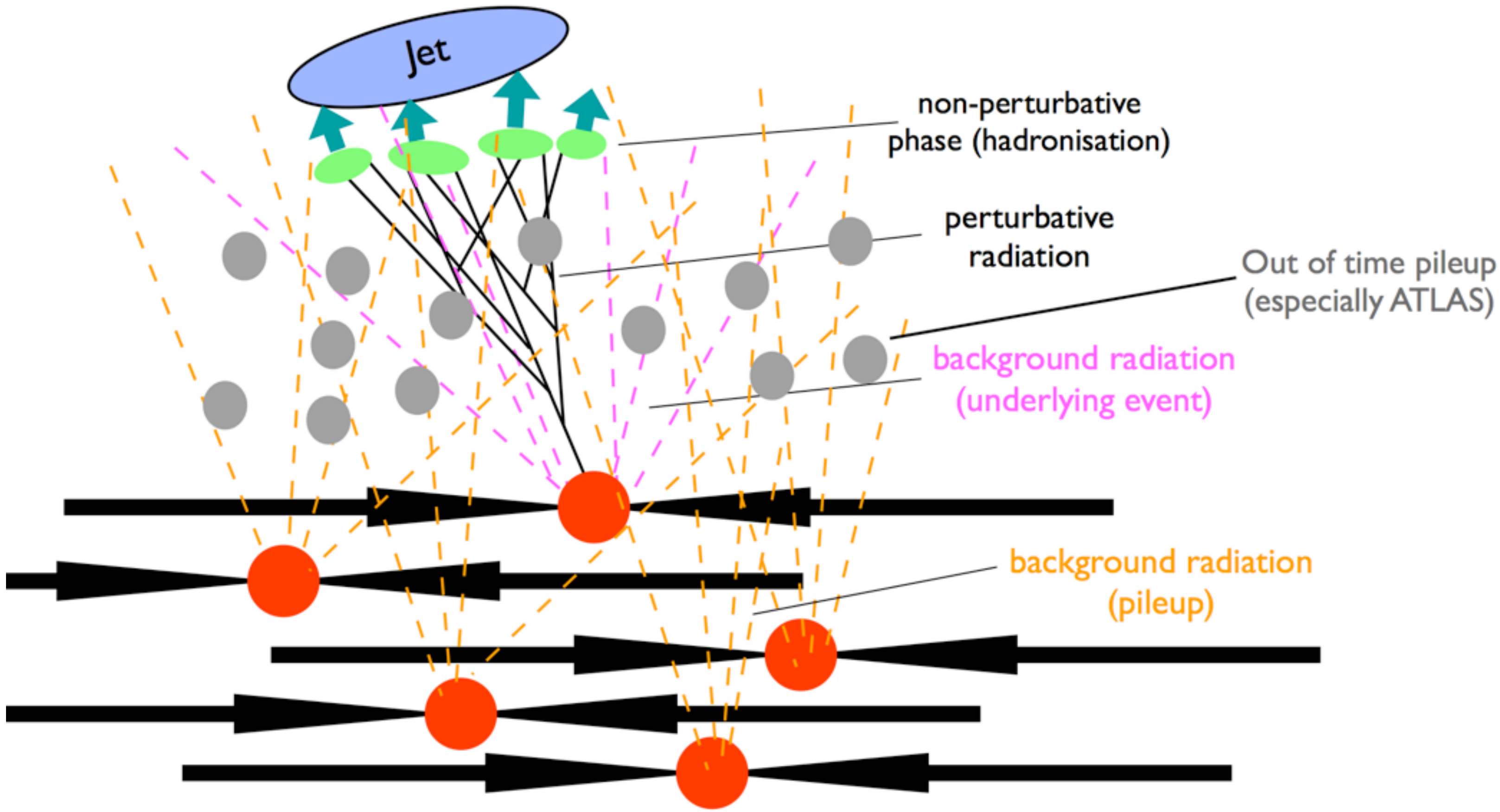


Inclusive jet cross section

Excellent theory-data agreement over many orders of magnitude

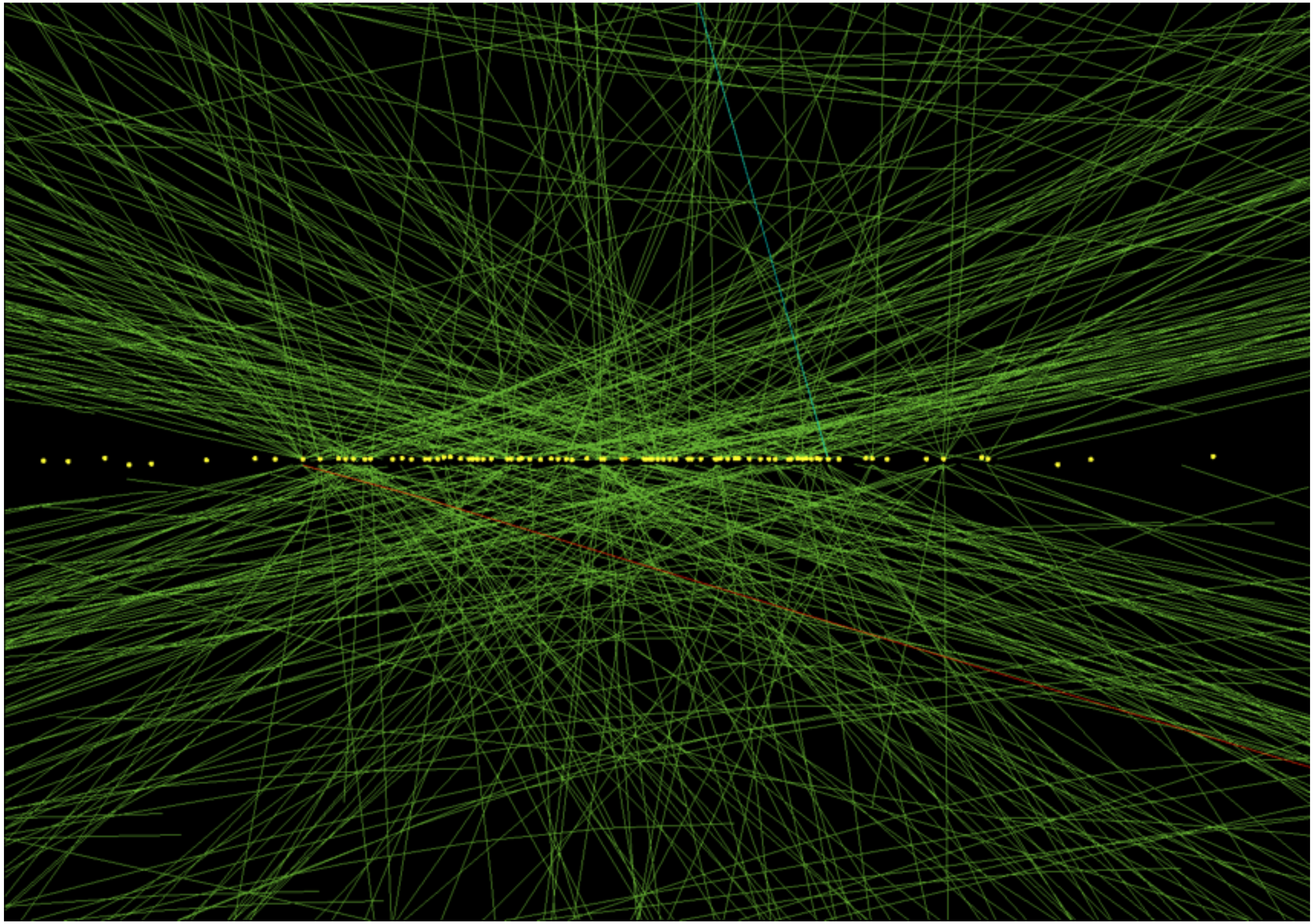
Background

Many 'things' can be clustered into (or lost from) a jet other than what we want (typically, perturbative radiation from a parton)



Ideally we'd like to be able to correct for these effects

Pileup



78-vertices event from CMS

https://cds.cern.ch/record/1479324

Pileup can deposit several tens of GeV (or even hundreds, in a heavy ion collision) into a medium-sized jet

Background subtraction

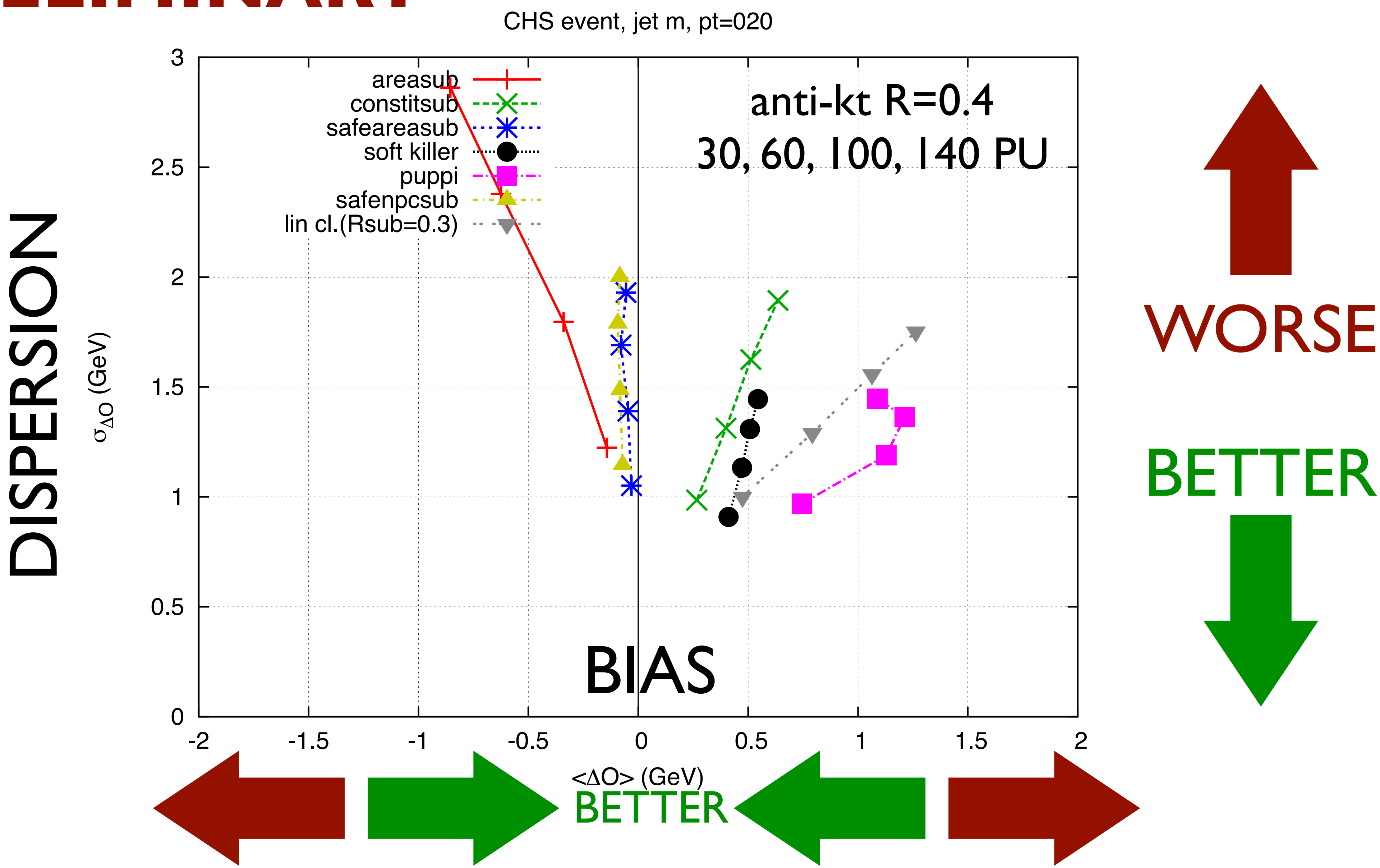
Many advanced tool have been developed in the past ten years to subtract background from jets

This can be done, with varying level of precision and bias, either at the **observable** level (measure a quantity, e.g. a jet p_t, and then correct its value), or at the **constituents** level (select only the 'good' constituents of the jet before measuring the quantity)

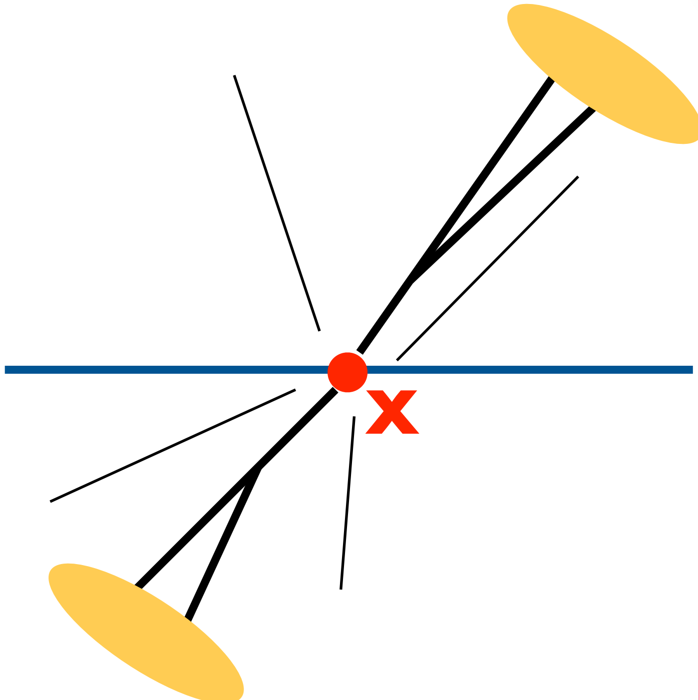
Comparisons of pileup subtractions

from the CERN pileup mitigation workshop, https://indico.cern.ch/event/306155/

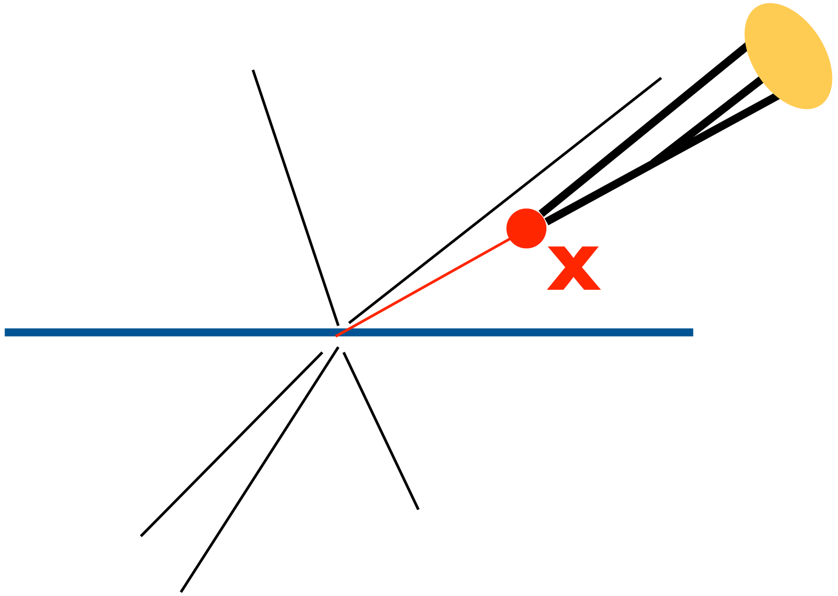
PRELIMINARY



Why boosted objects



Heavy particle X at **rest**



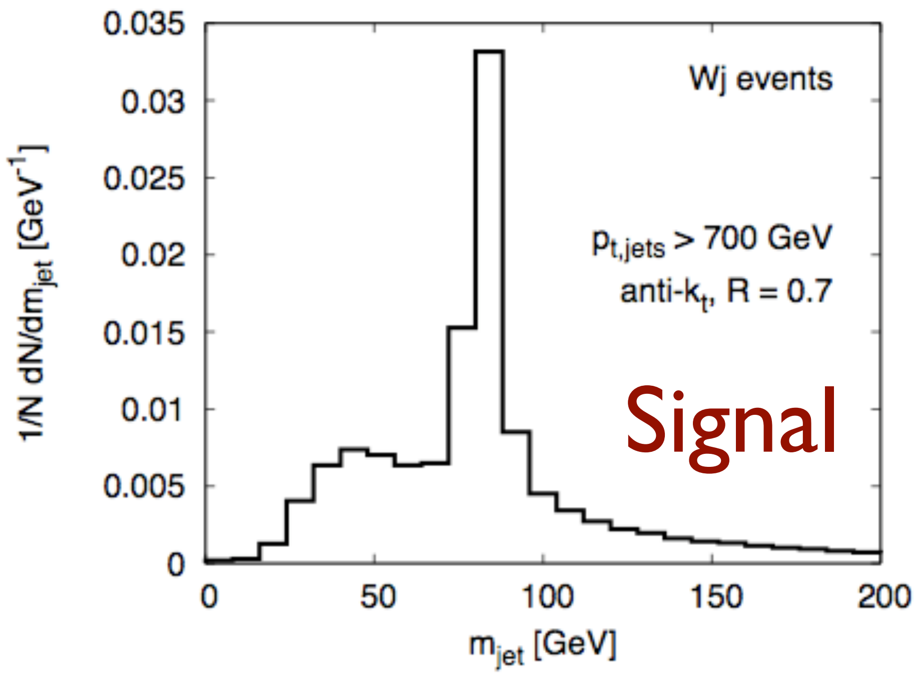
Boosted heavy particle X

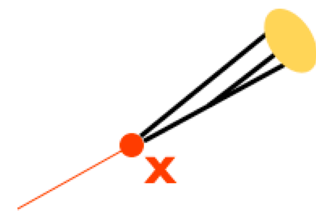
Easy to resolve jets and calculate invariant mass, but signal very likely swamped by background (eg H→bb v. tt →WbWb)

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

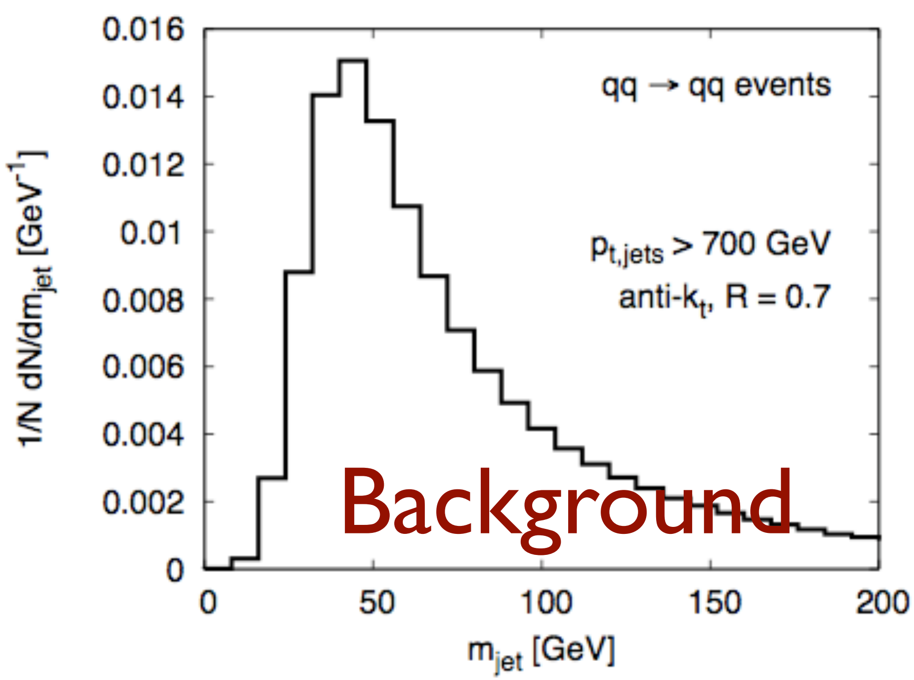
Mass of a single jet

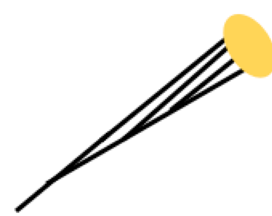
G. Salam





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



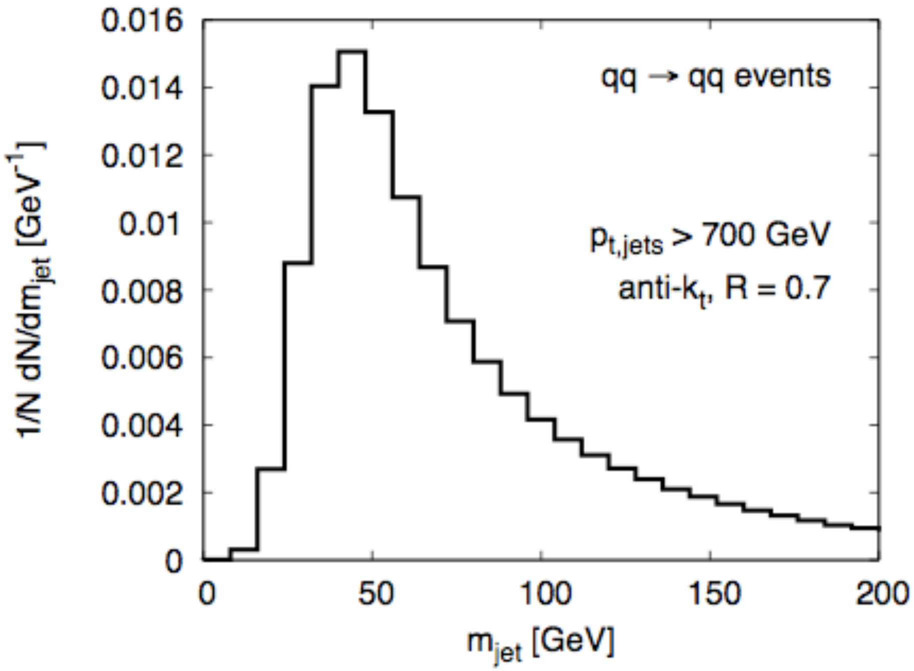


... but pure QCD jets can be massive too:

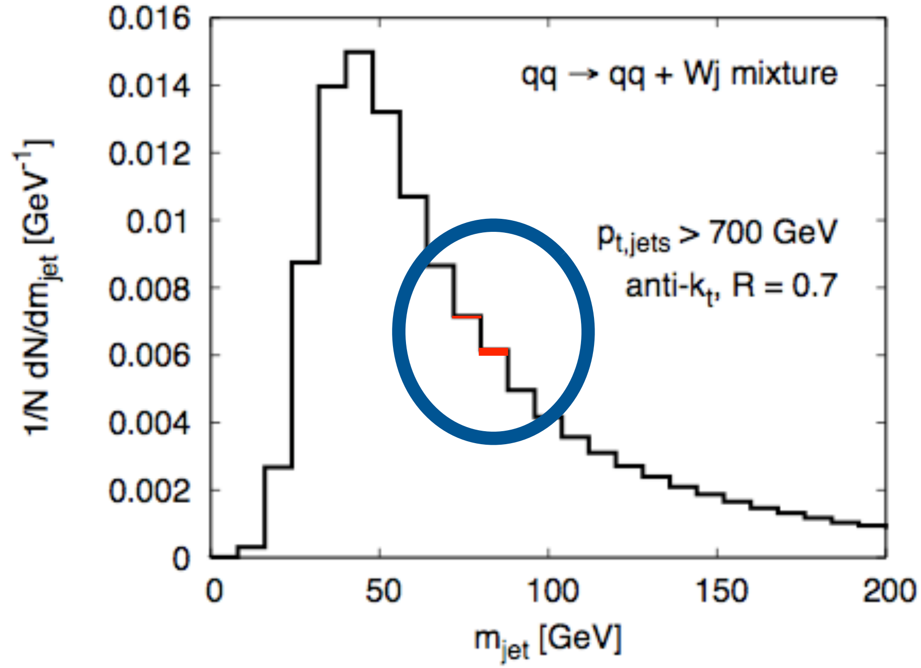
$$\frac{dN}{d\ln m} \sim \alpha_s \ln \frac{p_t R}{m} \times \text{Sudakov}$$

Mass of a single jet

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



Background only

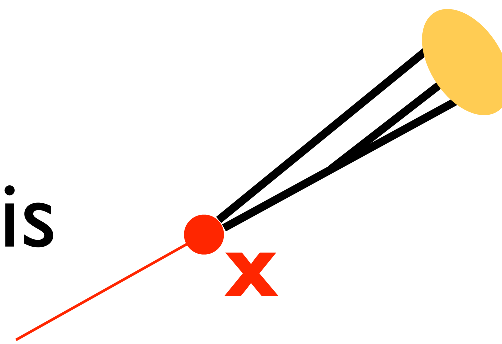


Signal + background

Practically identical

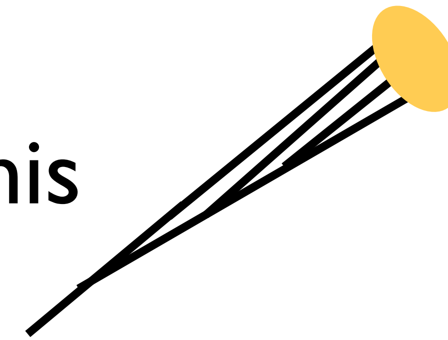
This means that one can't rely on the invariant mass only.
An appropriate strategy must be found to reduce the background and enhance the signal

How to tell this



Decay of a heavy (boosted) object

from this



Light parton fragmentation

?

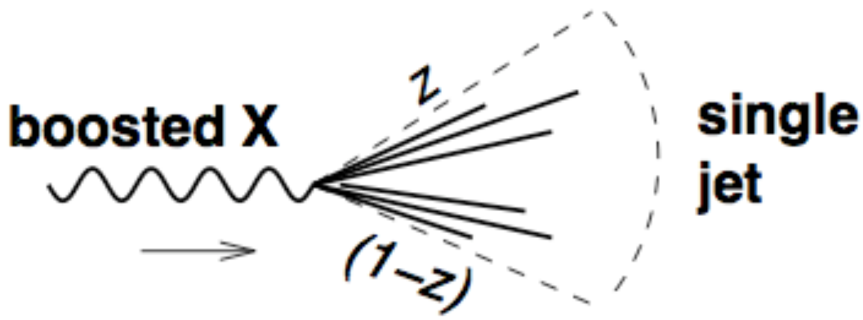
Tagging and Grooming

- 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*

Why substructure

Scales: m ~ 100 GeV, p_t ~ 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 (< $2m/p_t$ ~ 0.4) to resolve two prongs
- need large R (>~ $3m/p_t$ ~ 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

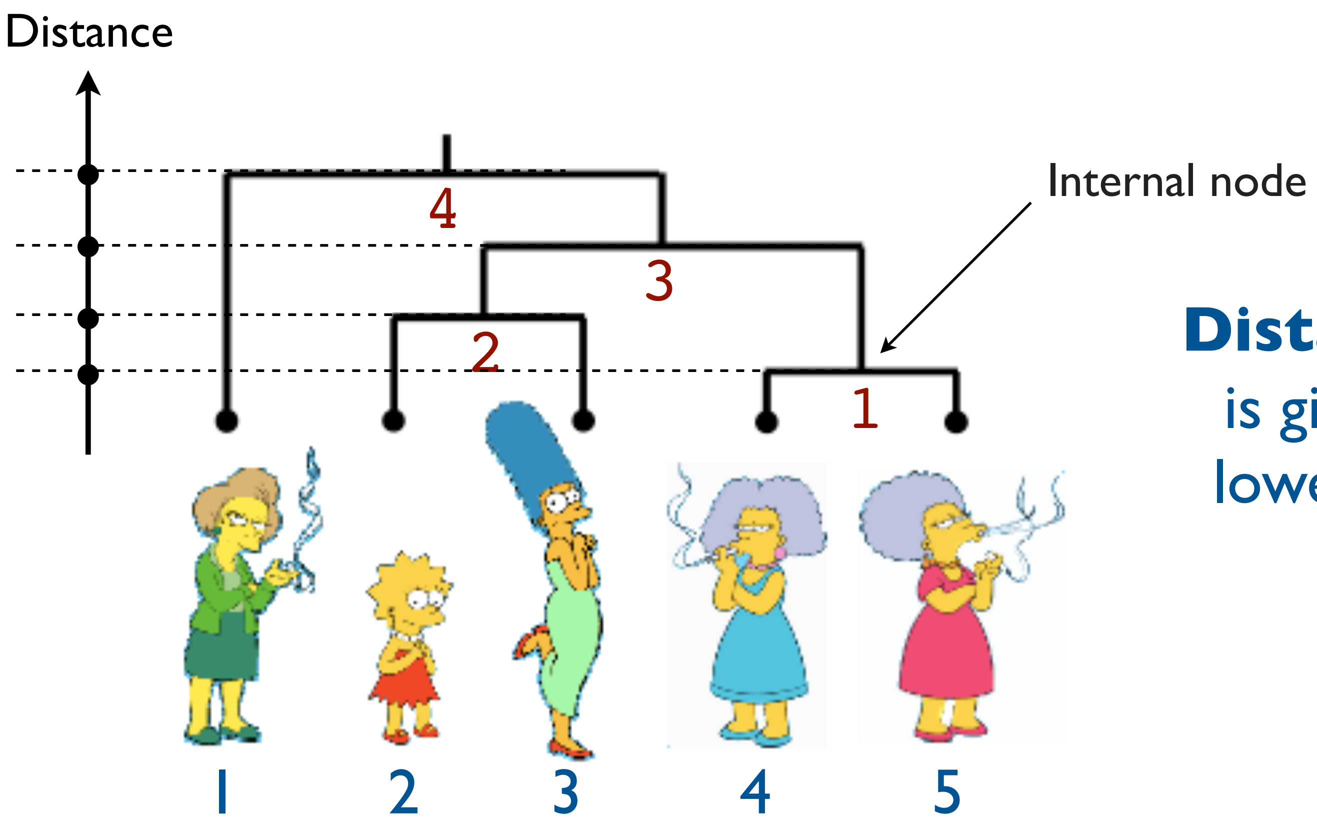
Let an algorithm find the 'right' substructure

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.

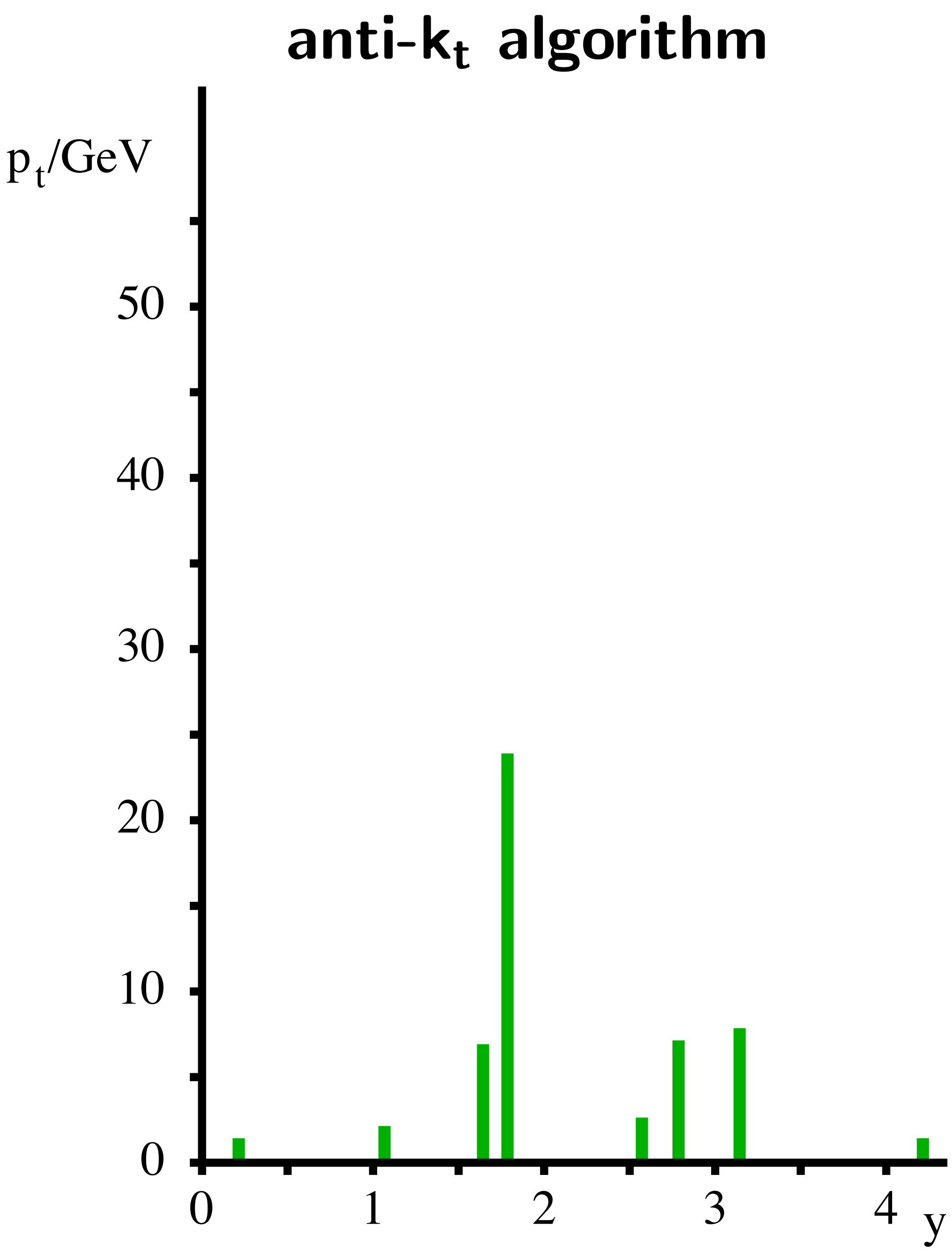
Order of clustering here is 1,2,3,4

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

First try

anti-kt

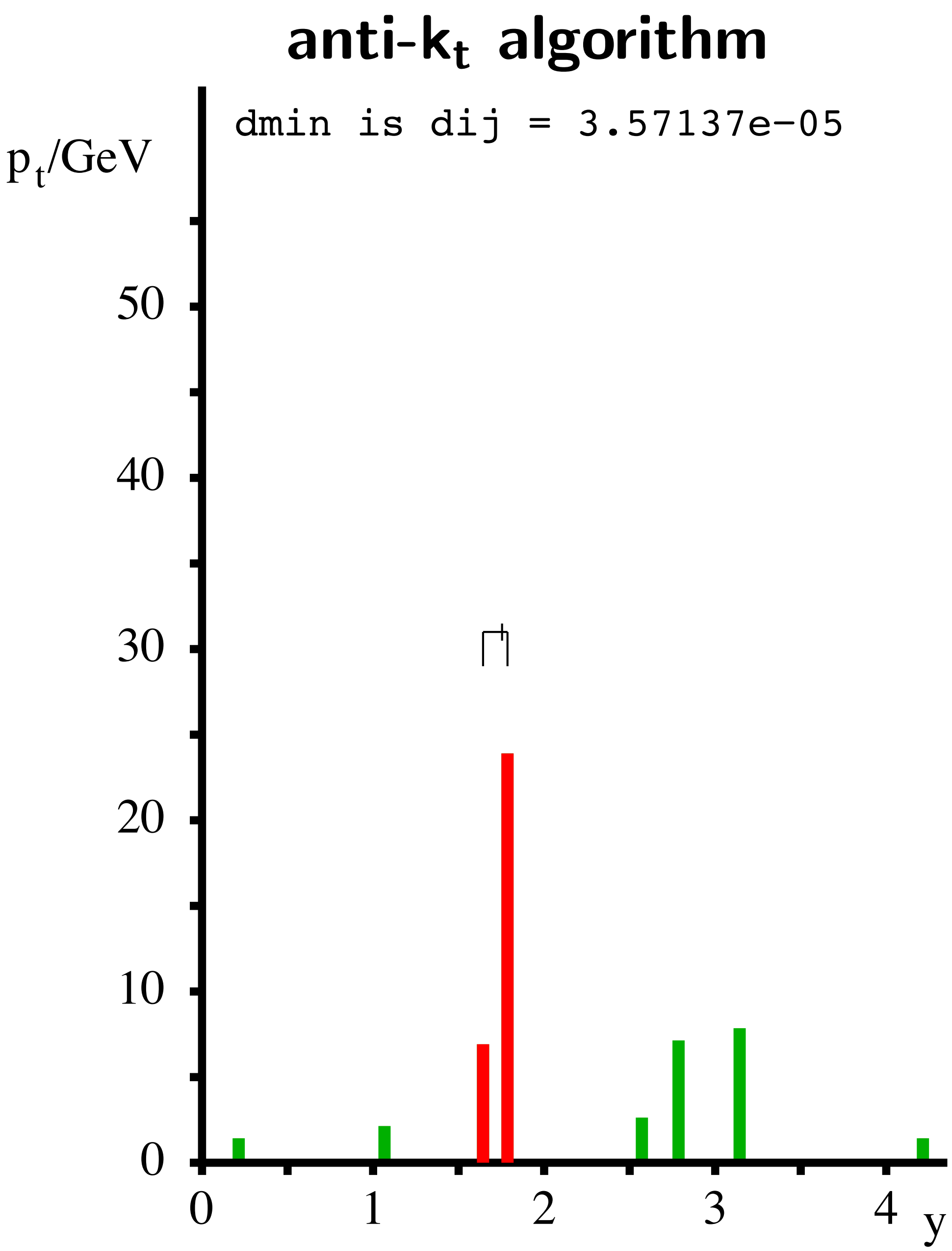
Identifying jet substructure: try out anti-k_t



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).

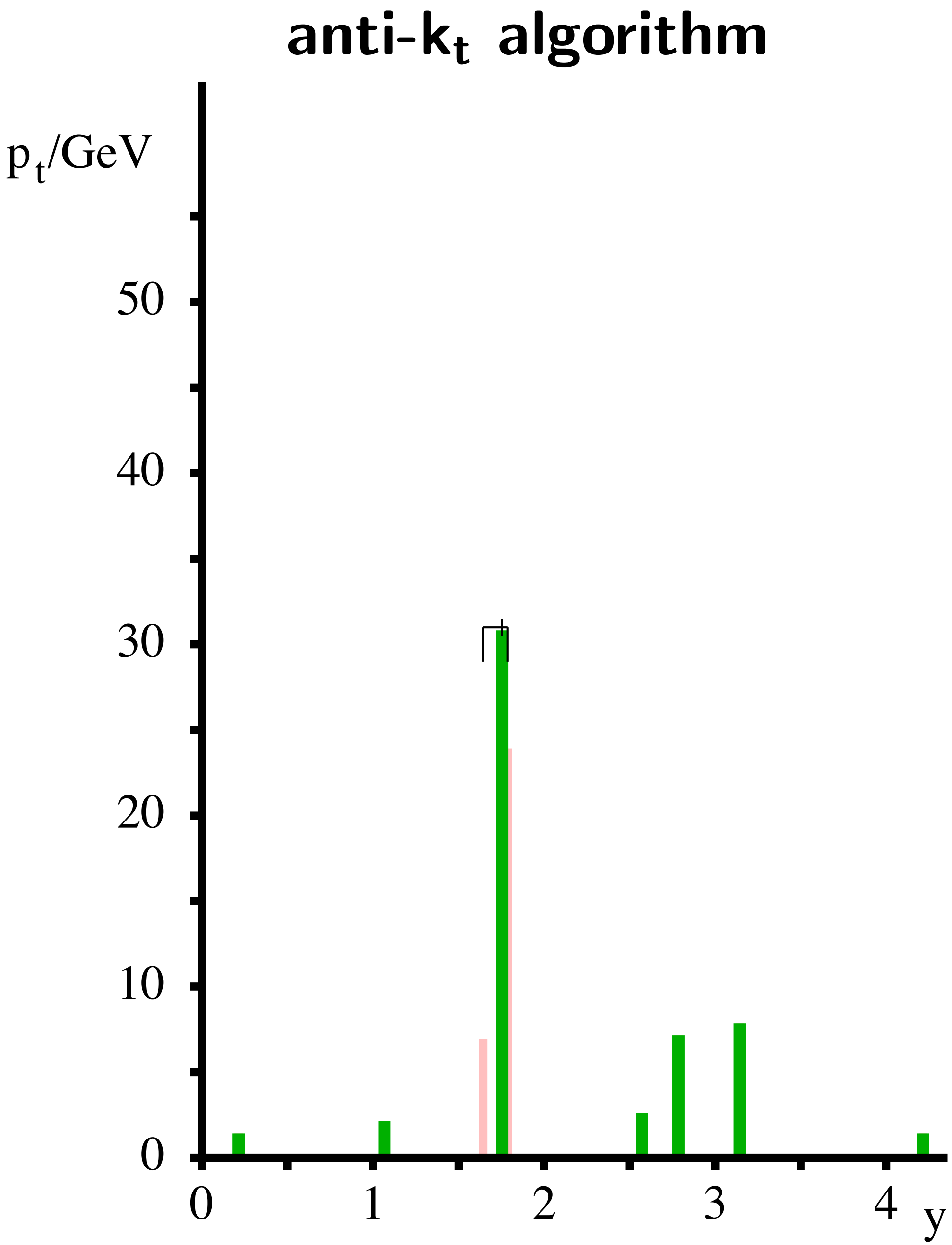
Identifying jet substructure: try out anti-k_t



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).

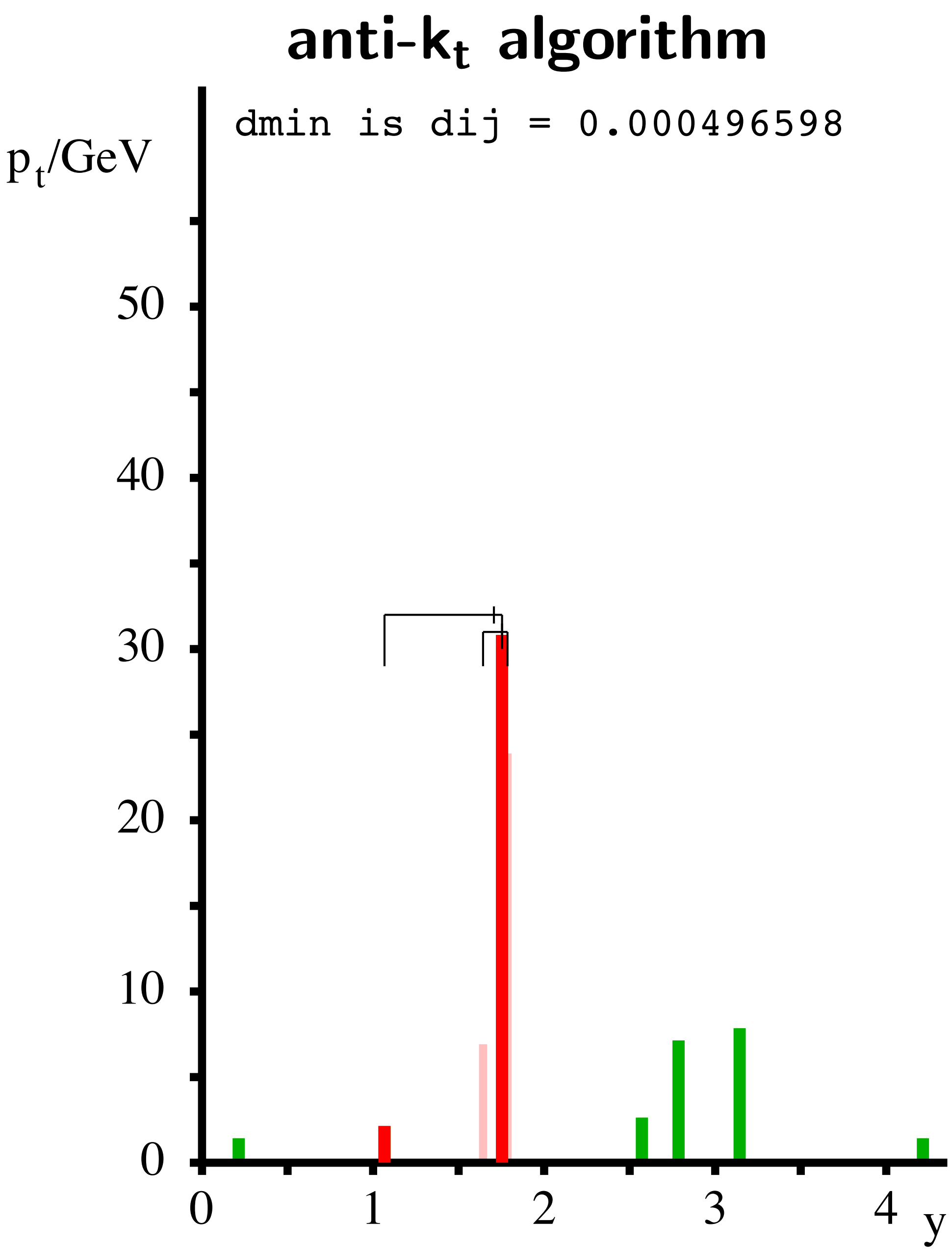
Identifying jet substructure: try out anti-k_t



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).

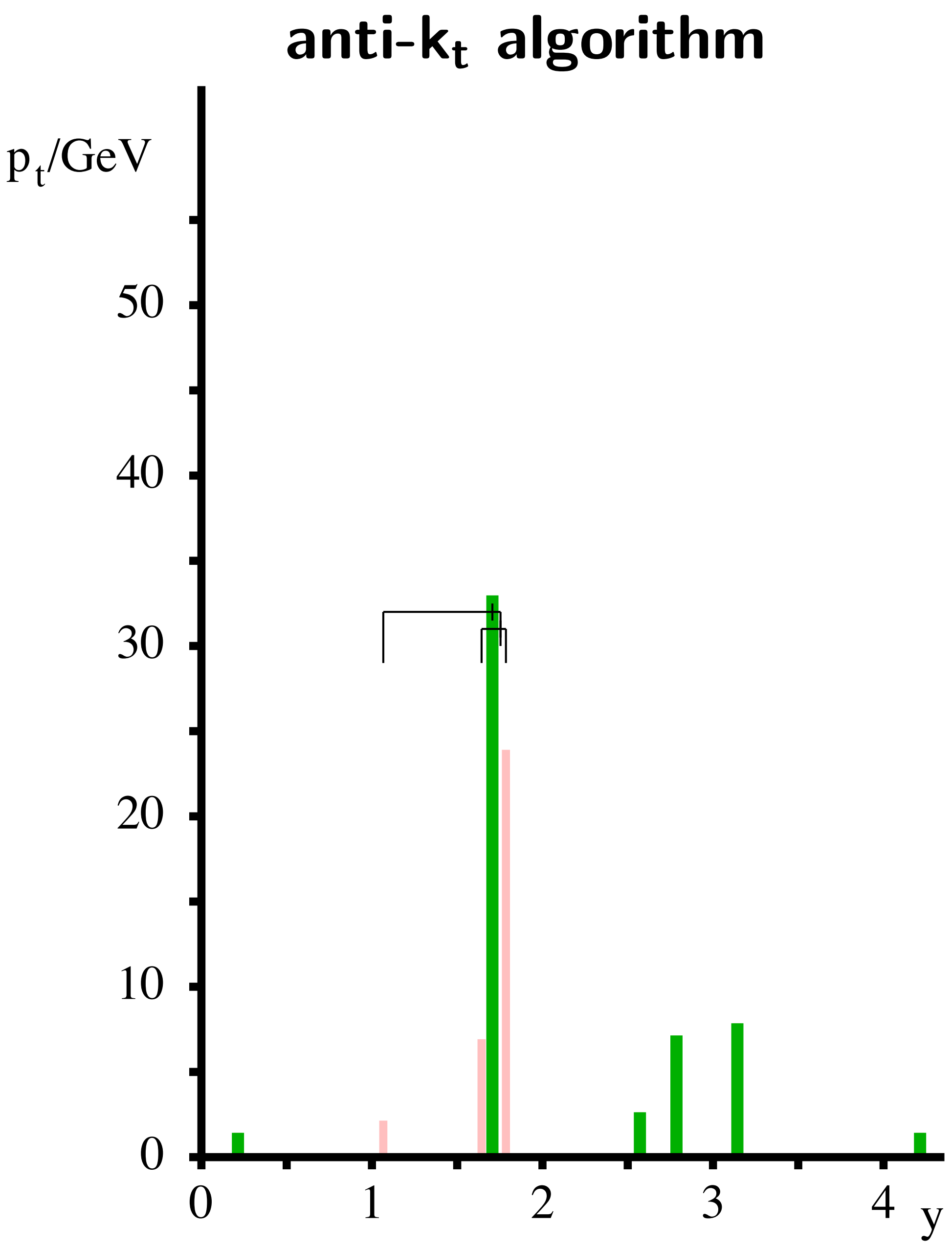
Identifying jet substructure: try out anti-k_t



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).

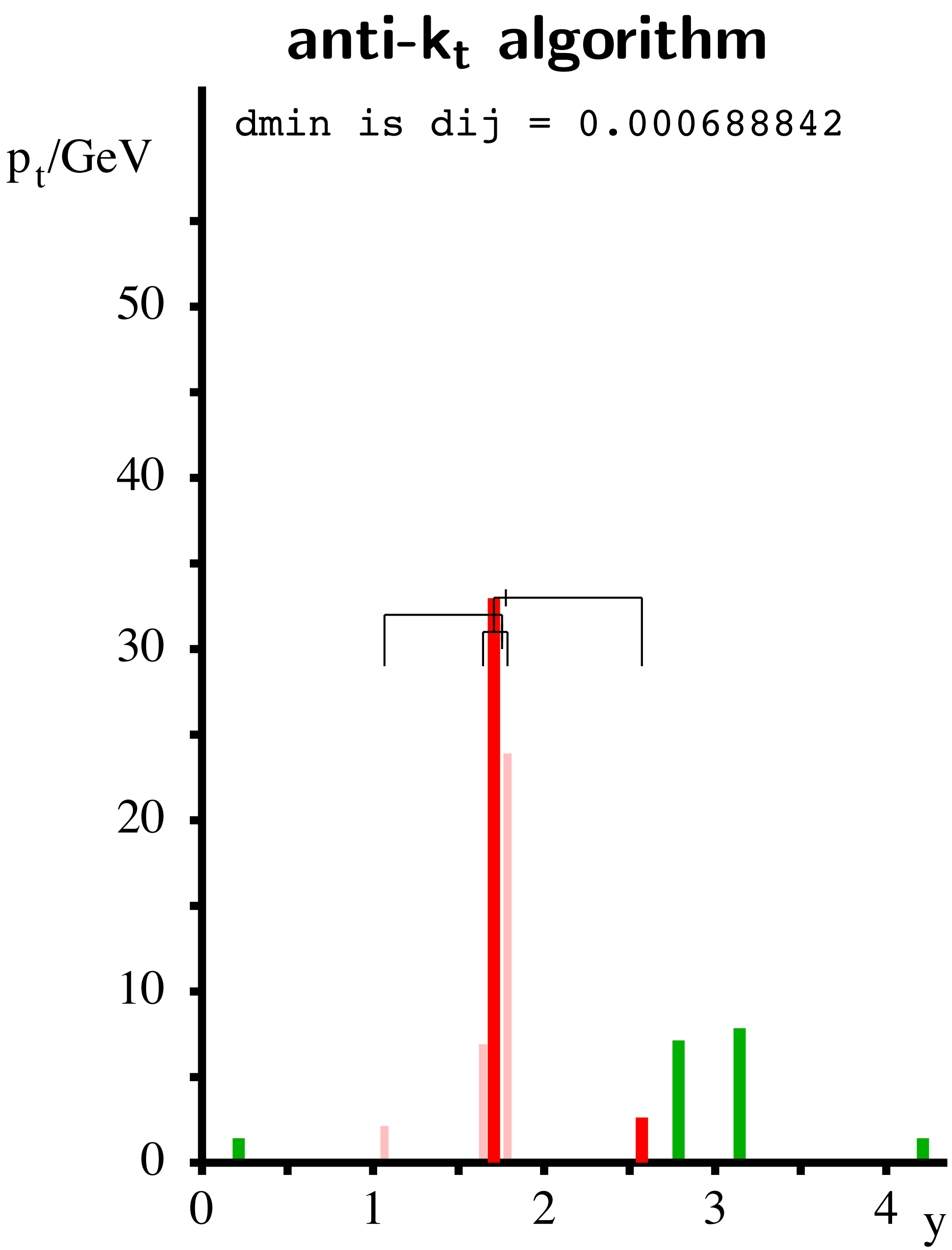
Identifying jet substructure: try out anti-k_t



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

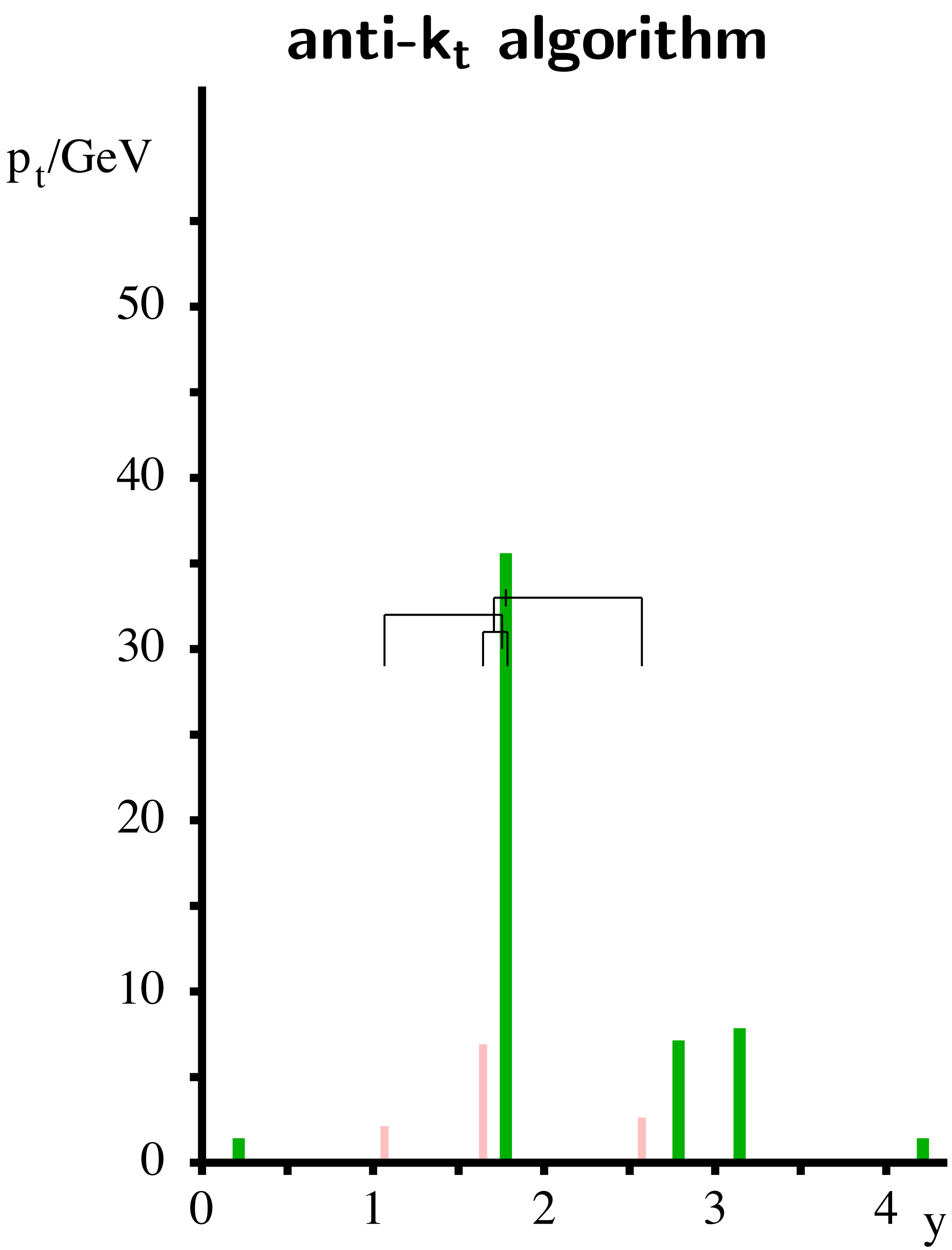
This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Identifying jet substructure: try out anti-k_t



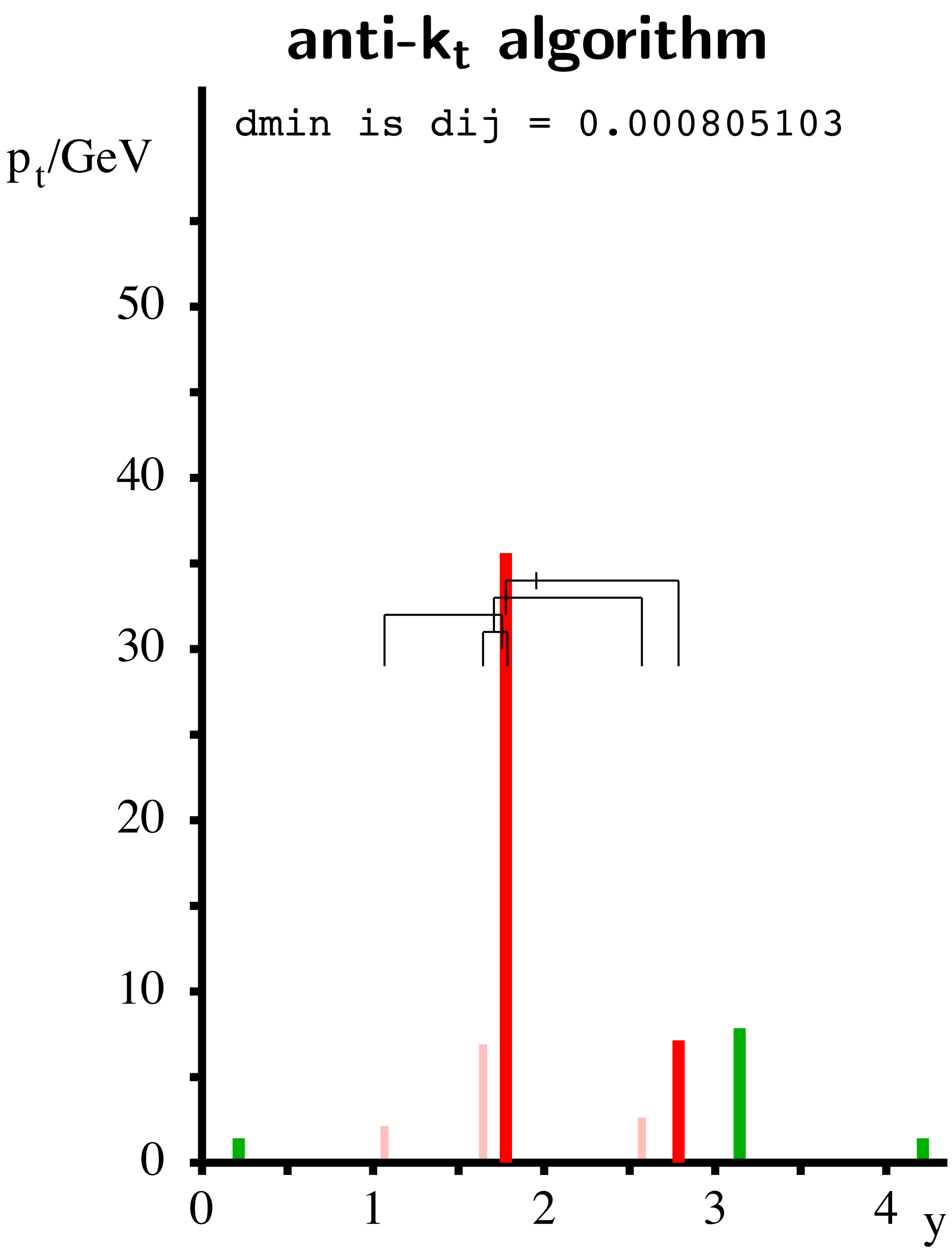
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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).

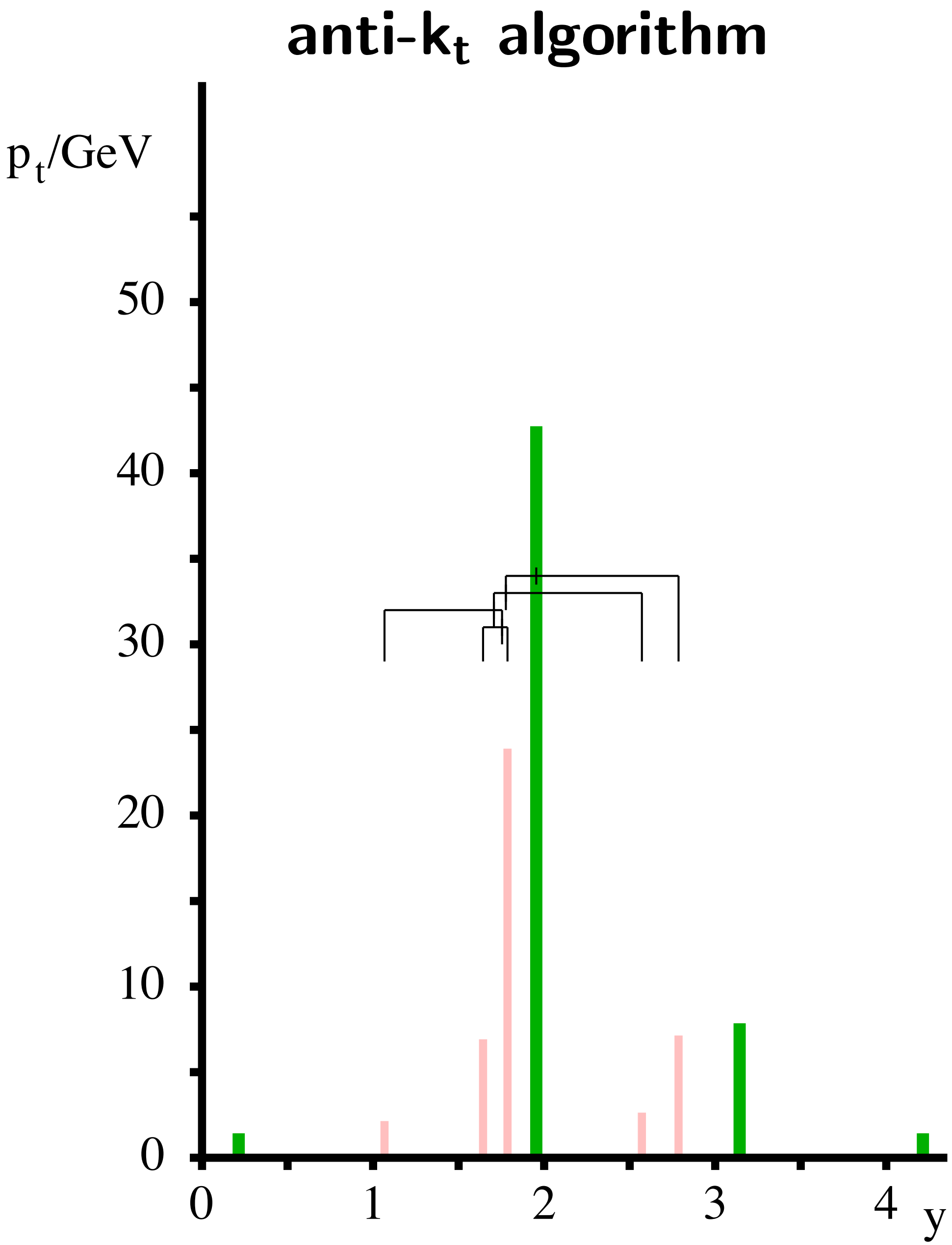


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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.

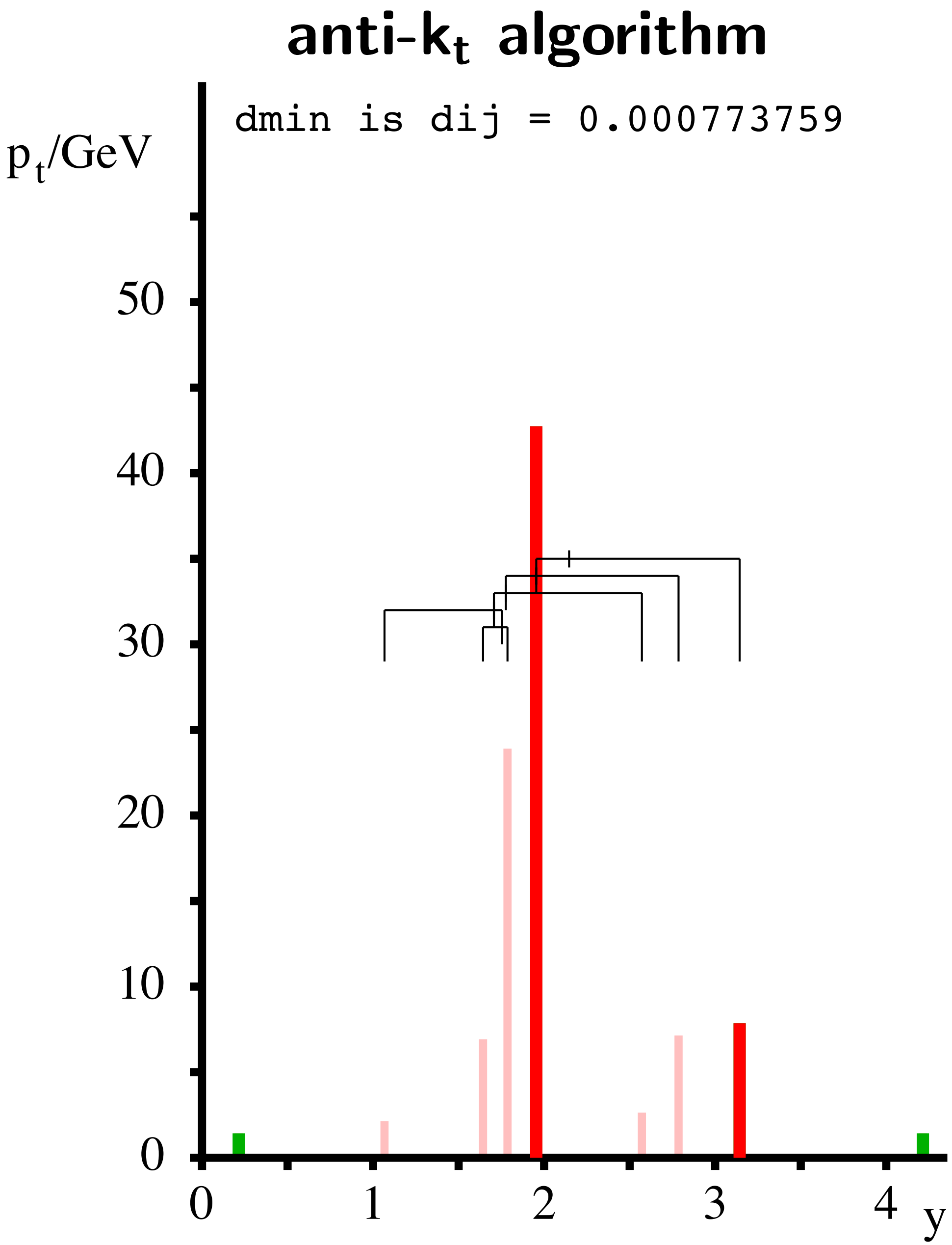
Identifying jet substructure: try out anti-k_t



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.

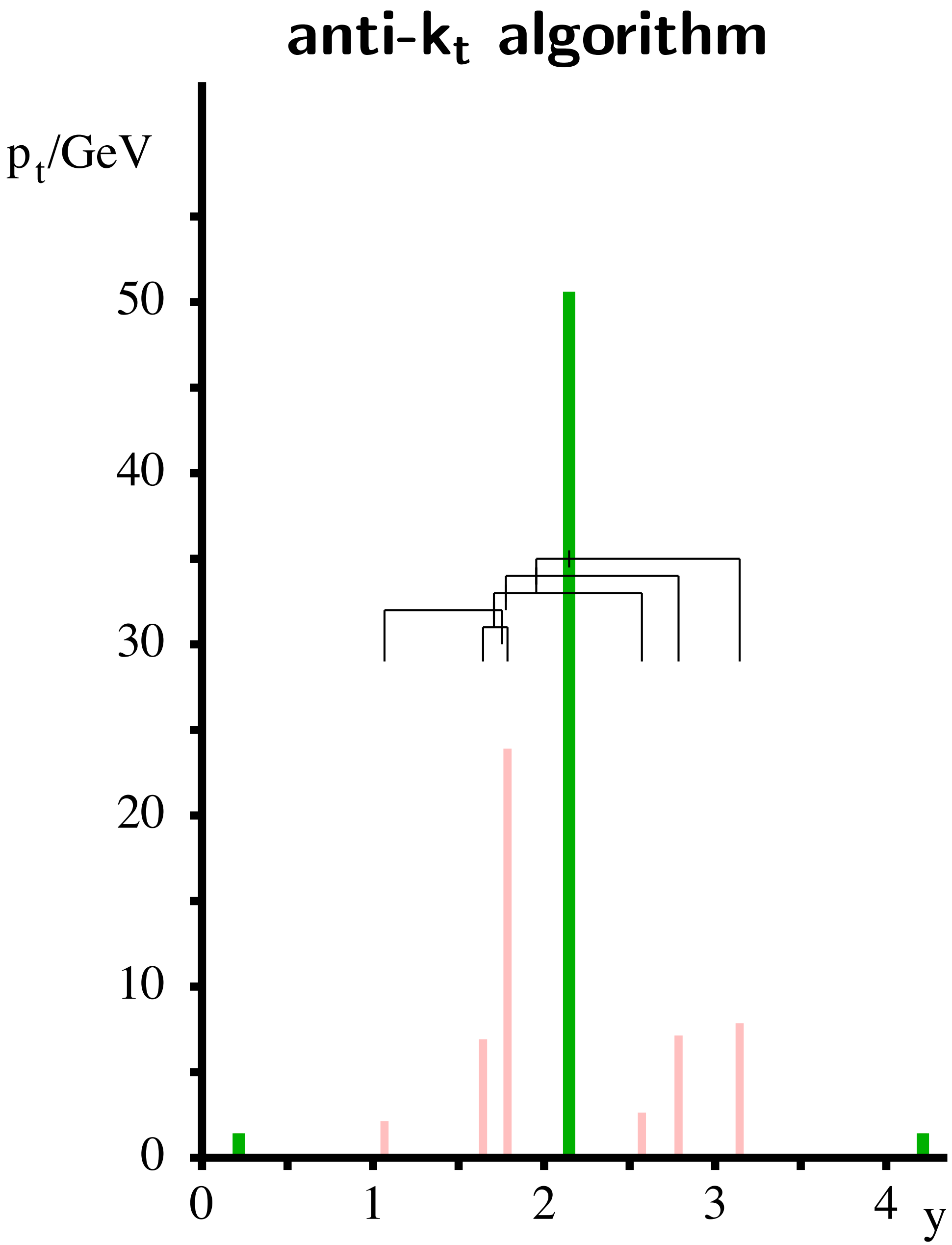


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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.

Identifying jet substructure: try out anti-k_t

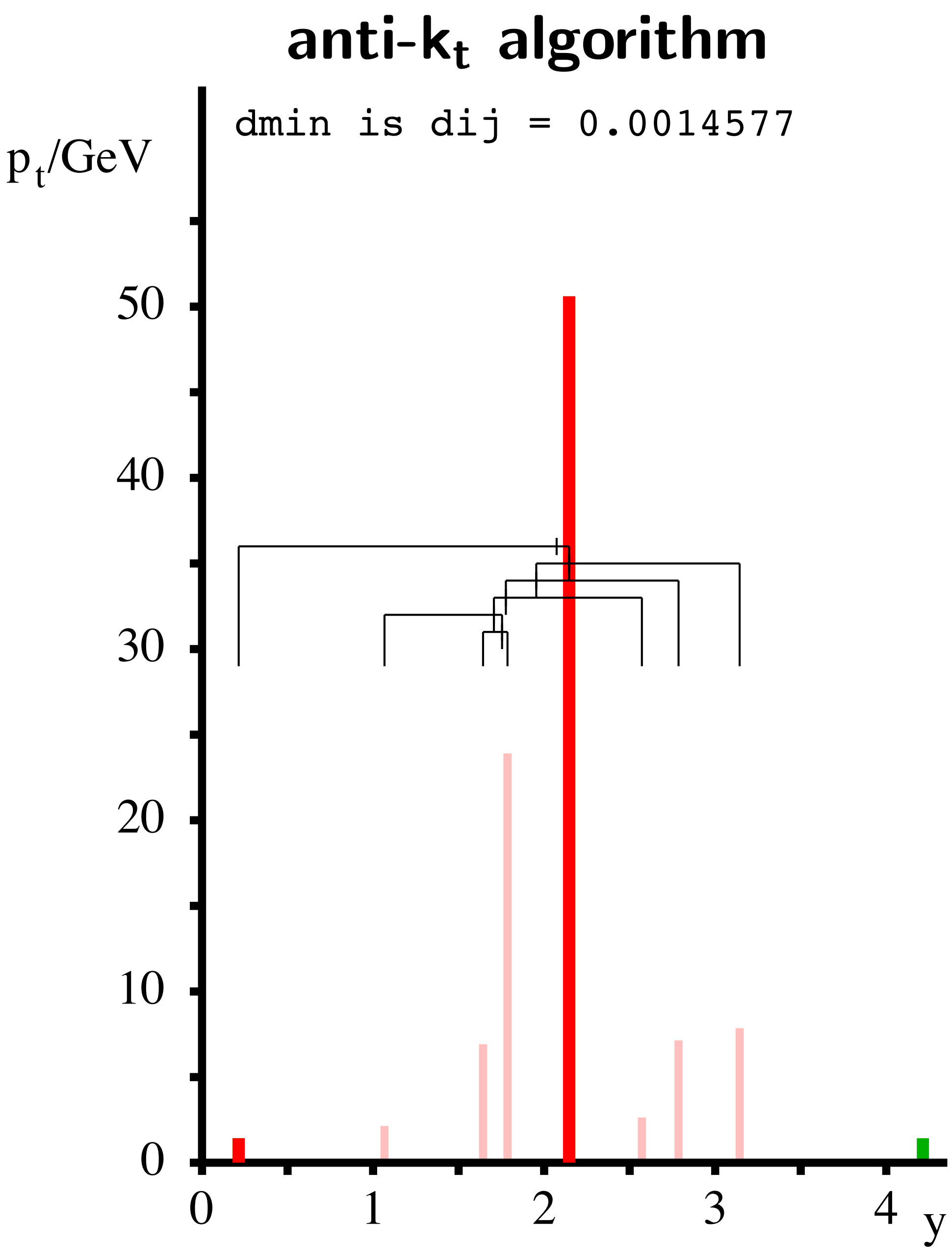


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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.

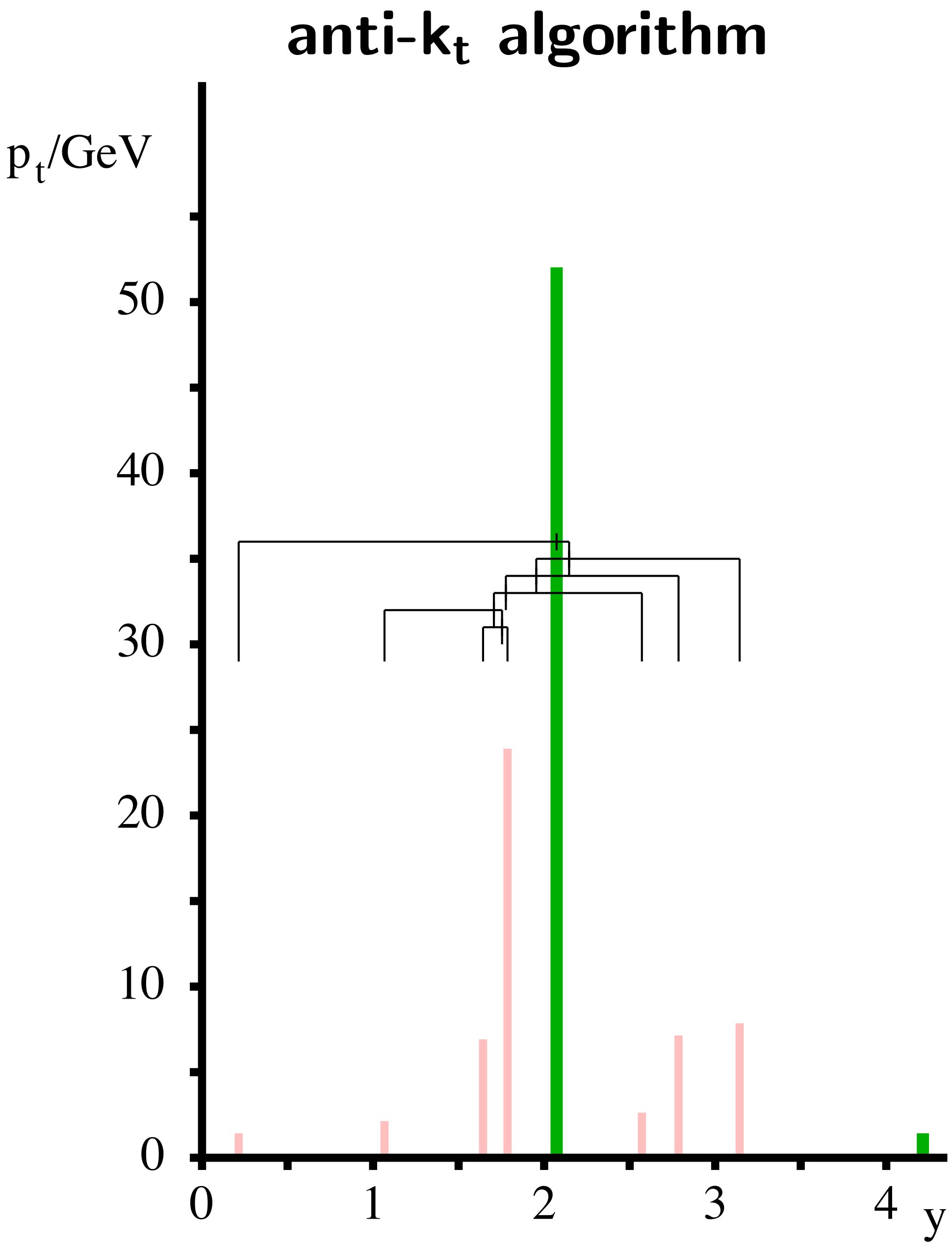
Identifying jet substructure: try out anti-k_t



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.

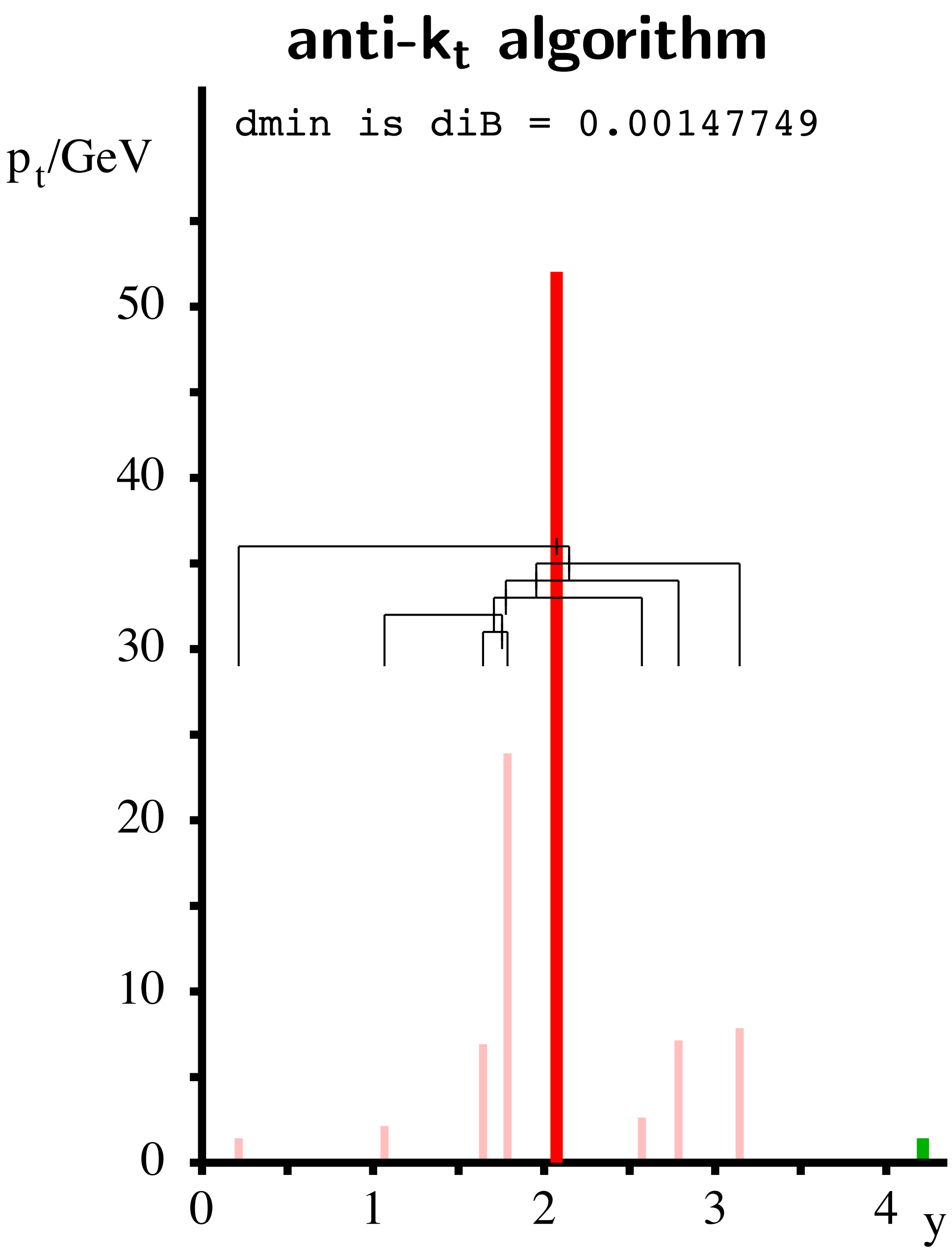


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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

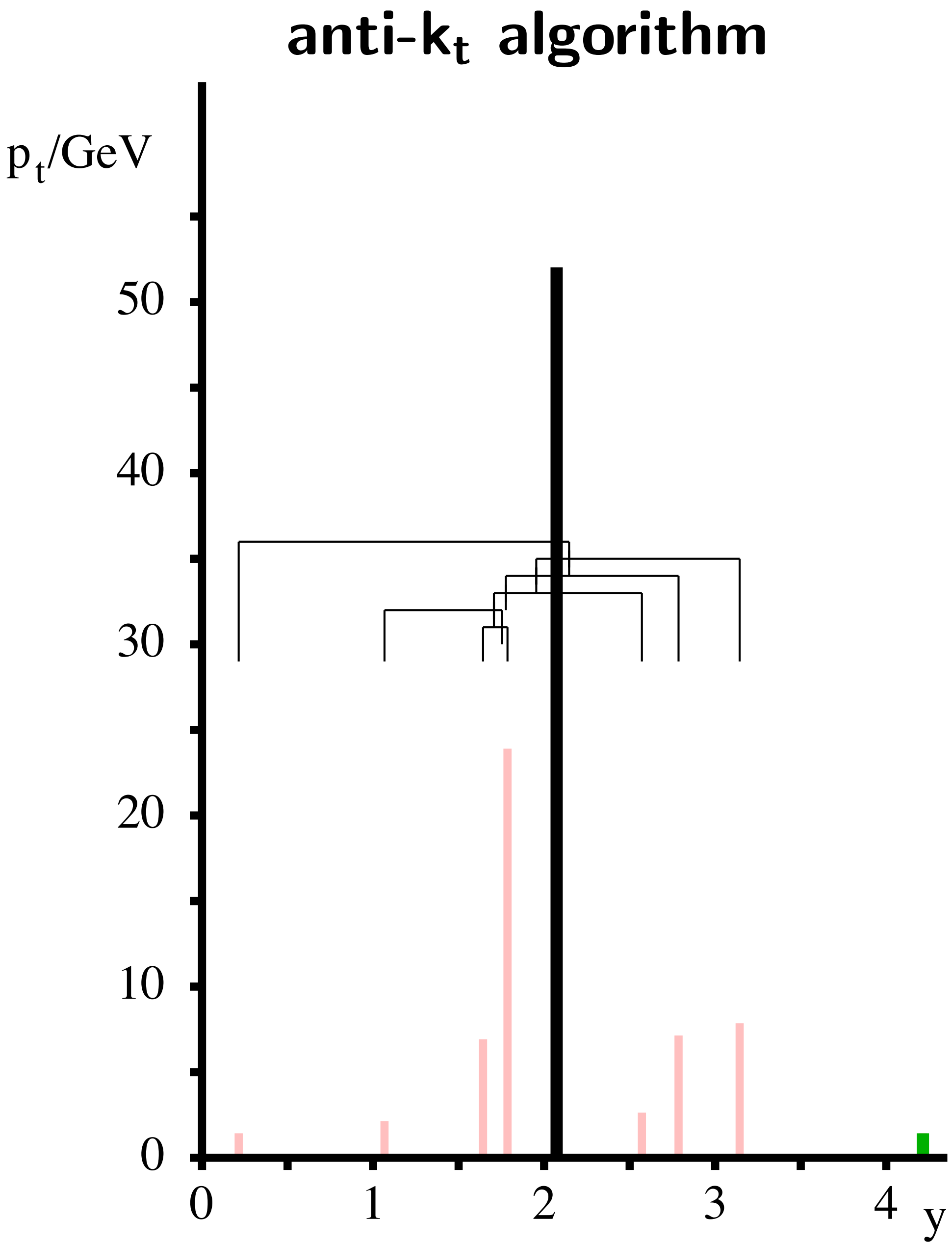


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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.

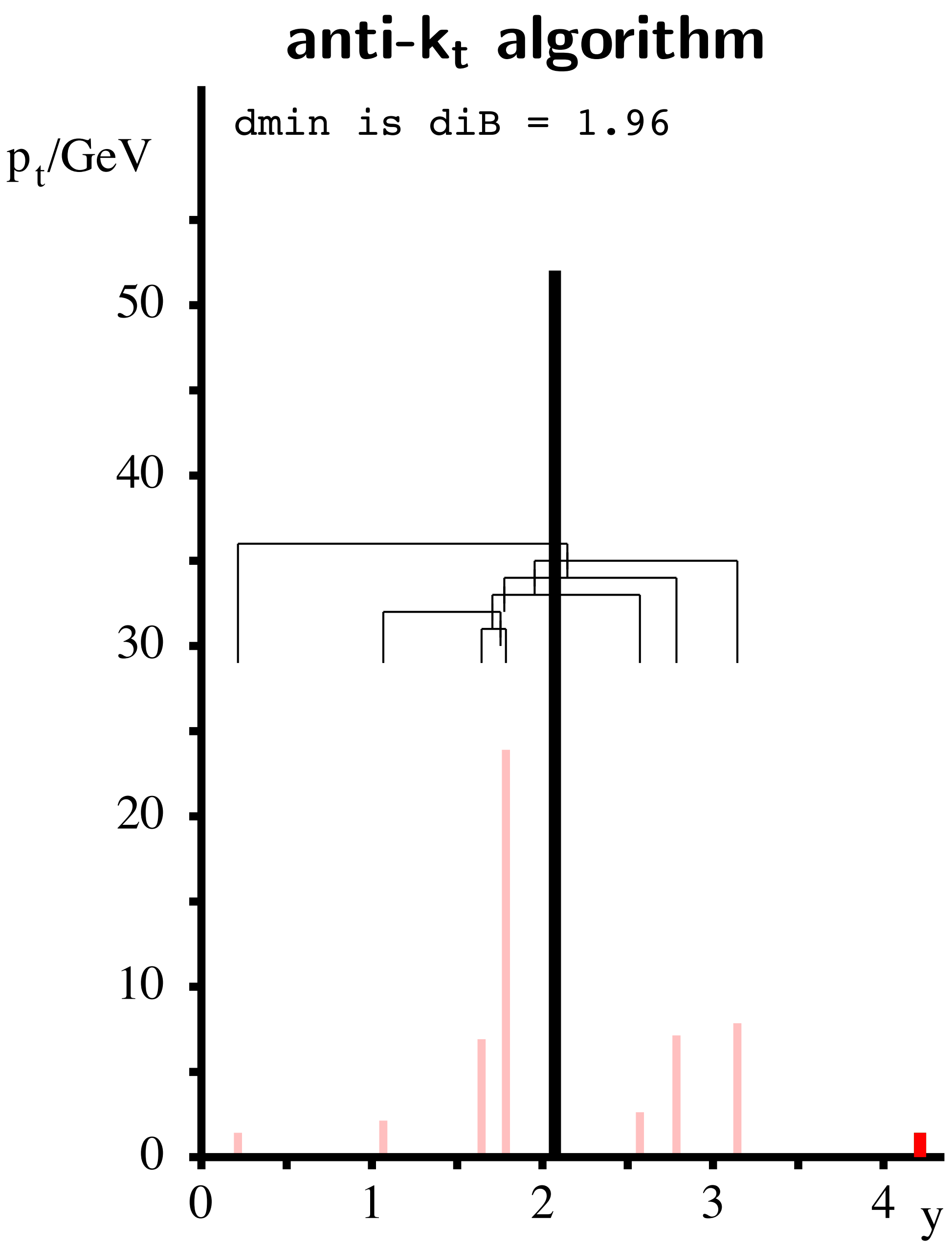
Identifying jet substructure: try out anti-k_t



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).

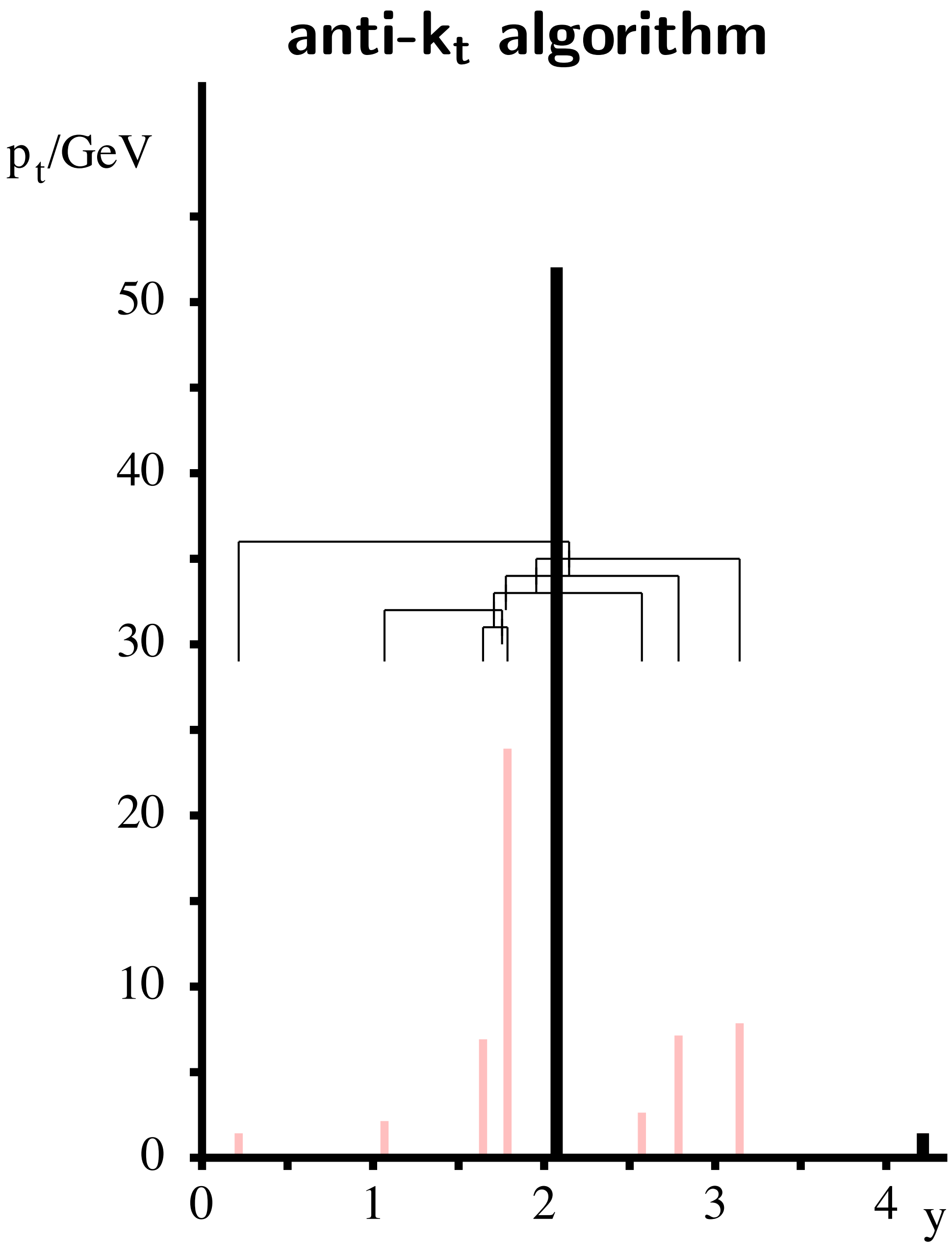
Anti-k_t gradually makes its way through the secondary blob $\rightarrow$ no clear identification of substructure associated with 2nd parton.



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 → 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?

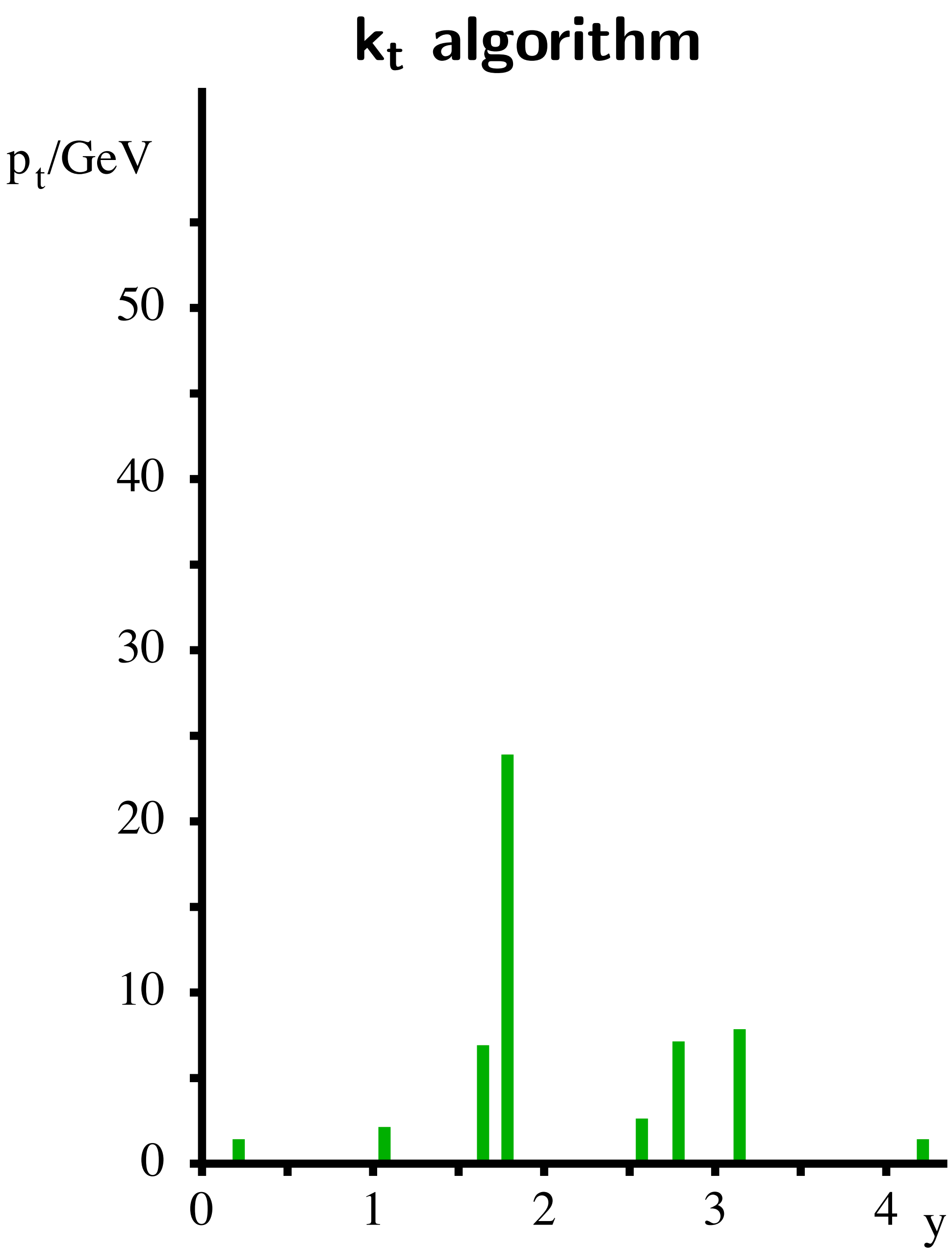
This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

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

Second try

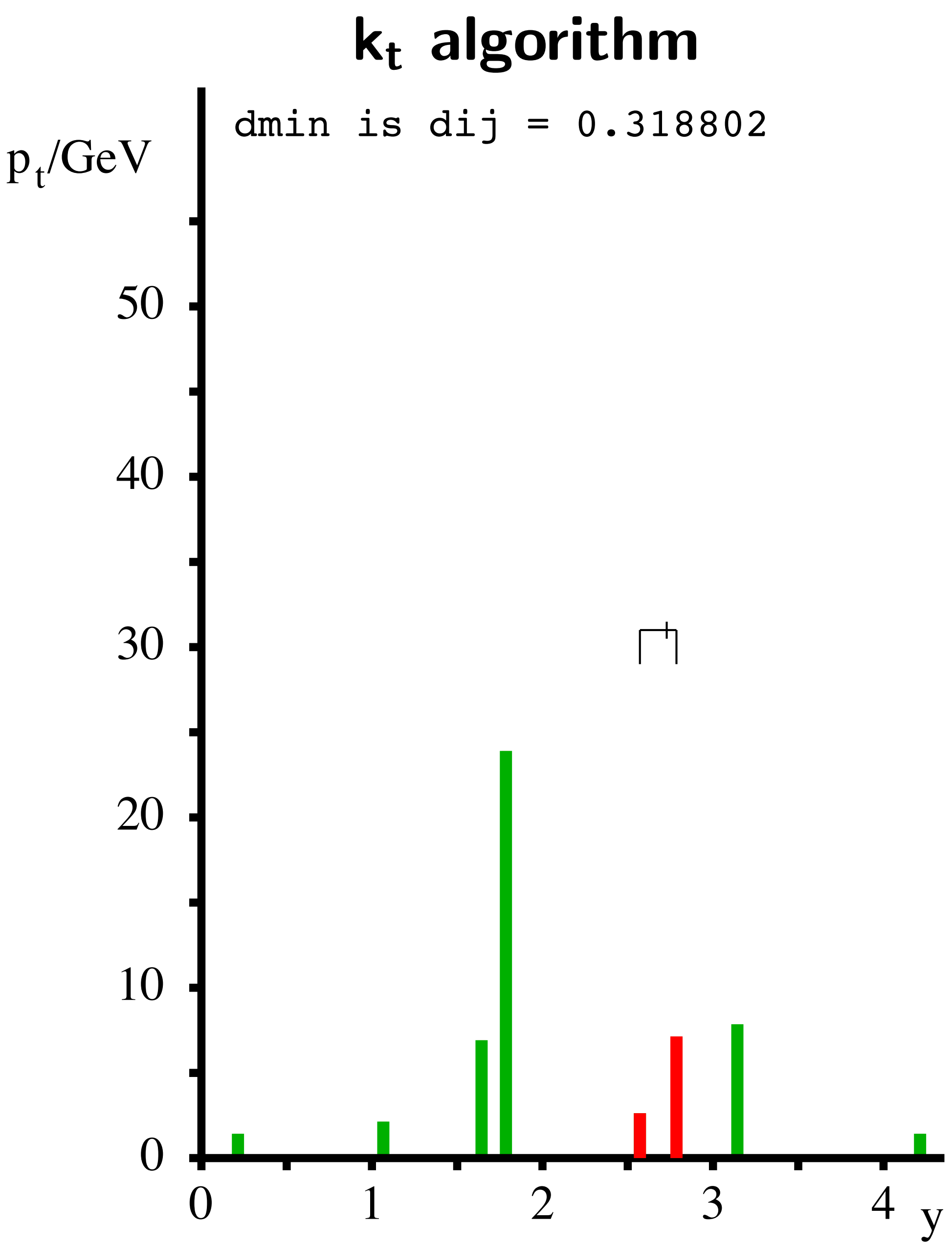
kt

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?

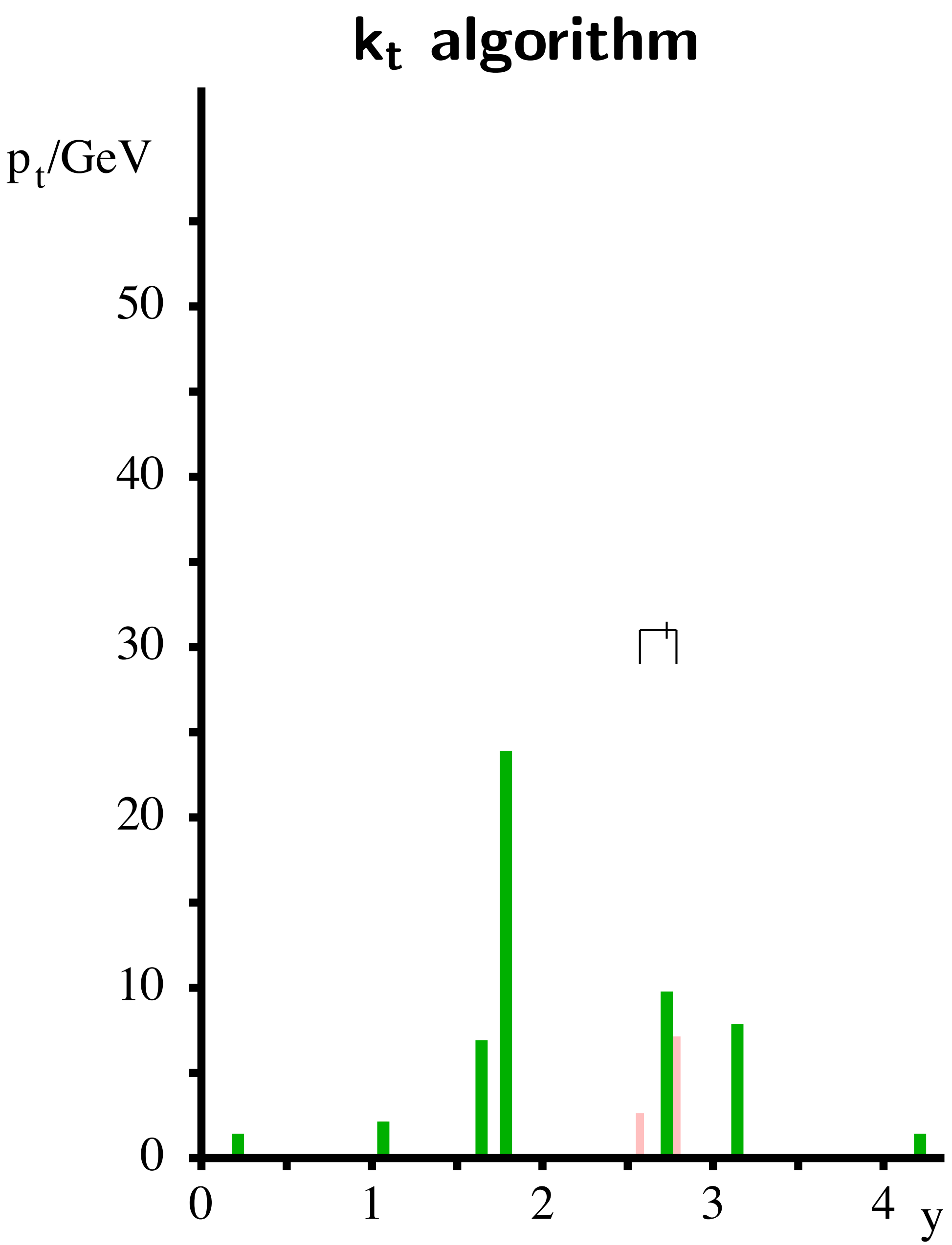
This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).



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

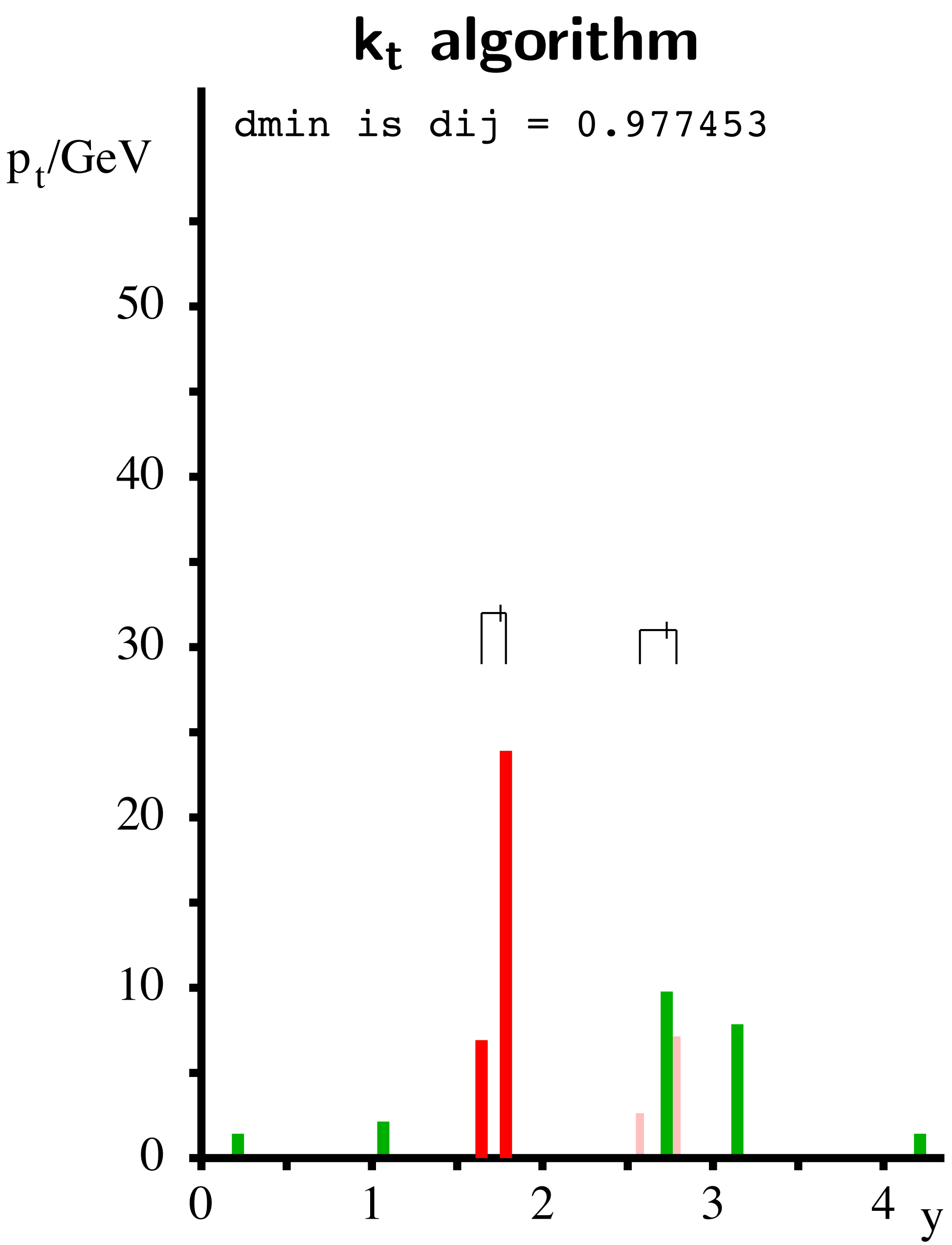
This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Identifying jet substructure: try out k_t



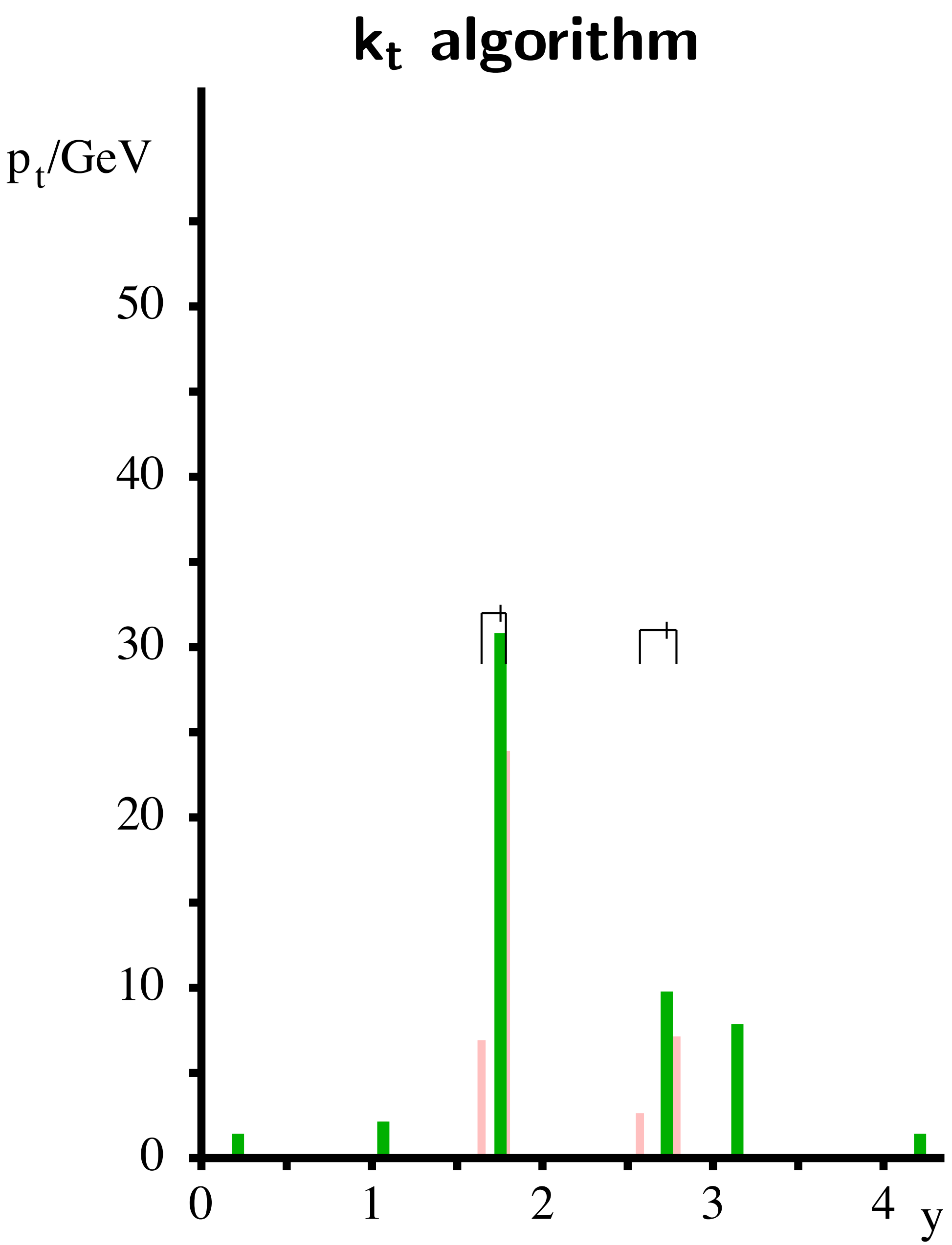
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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



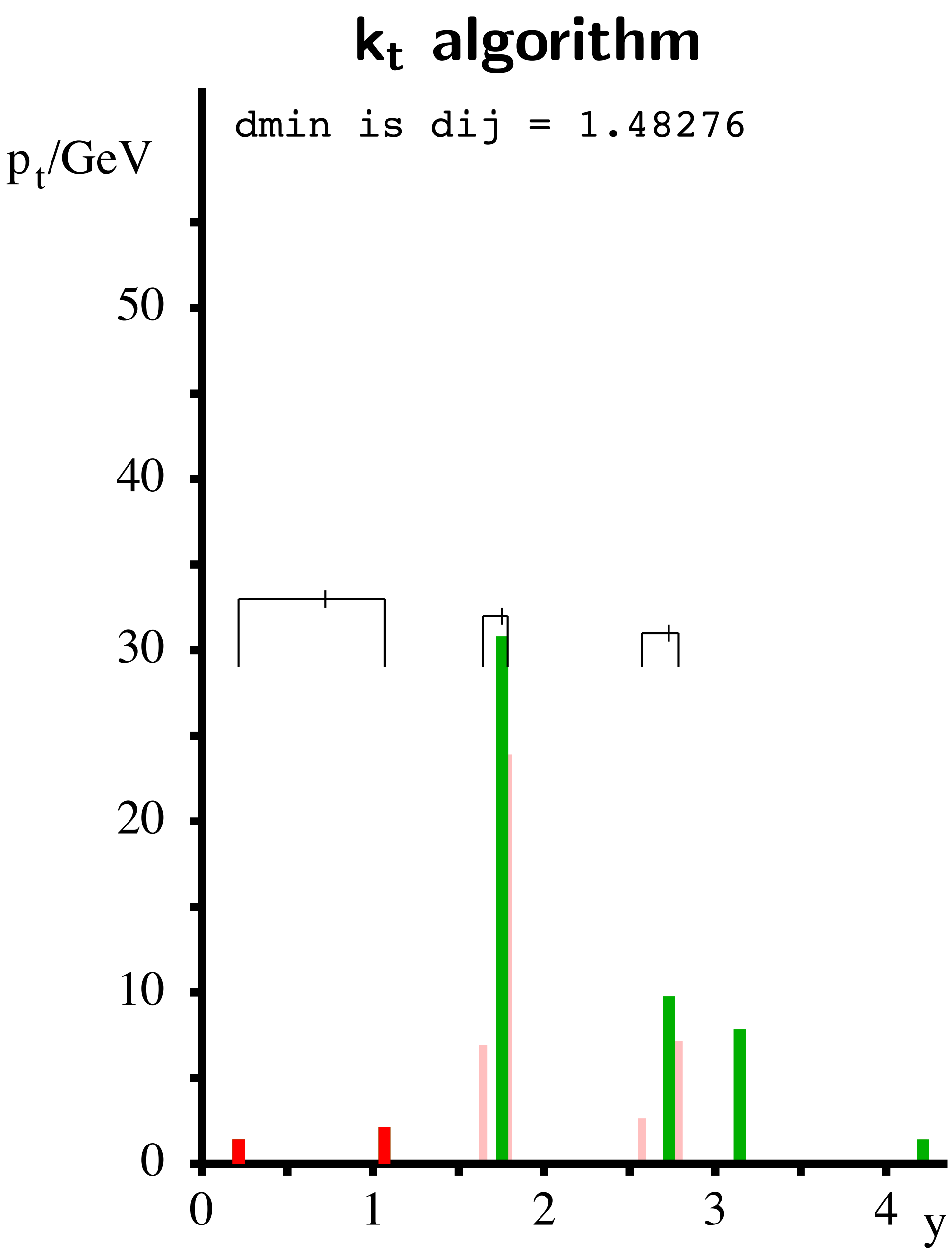
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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



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

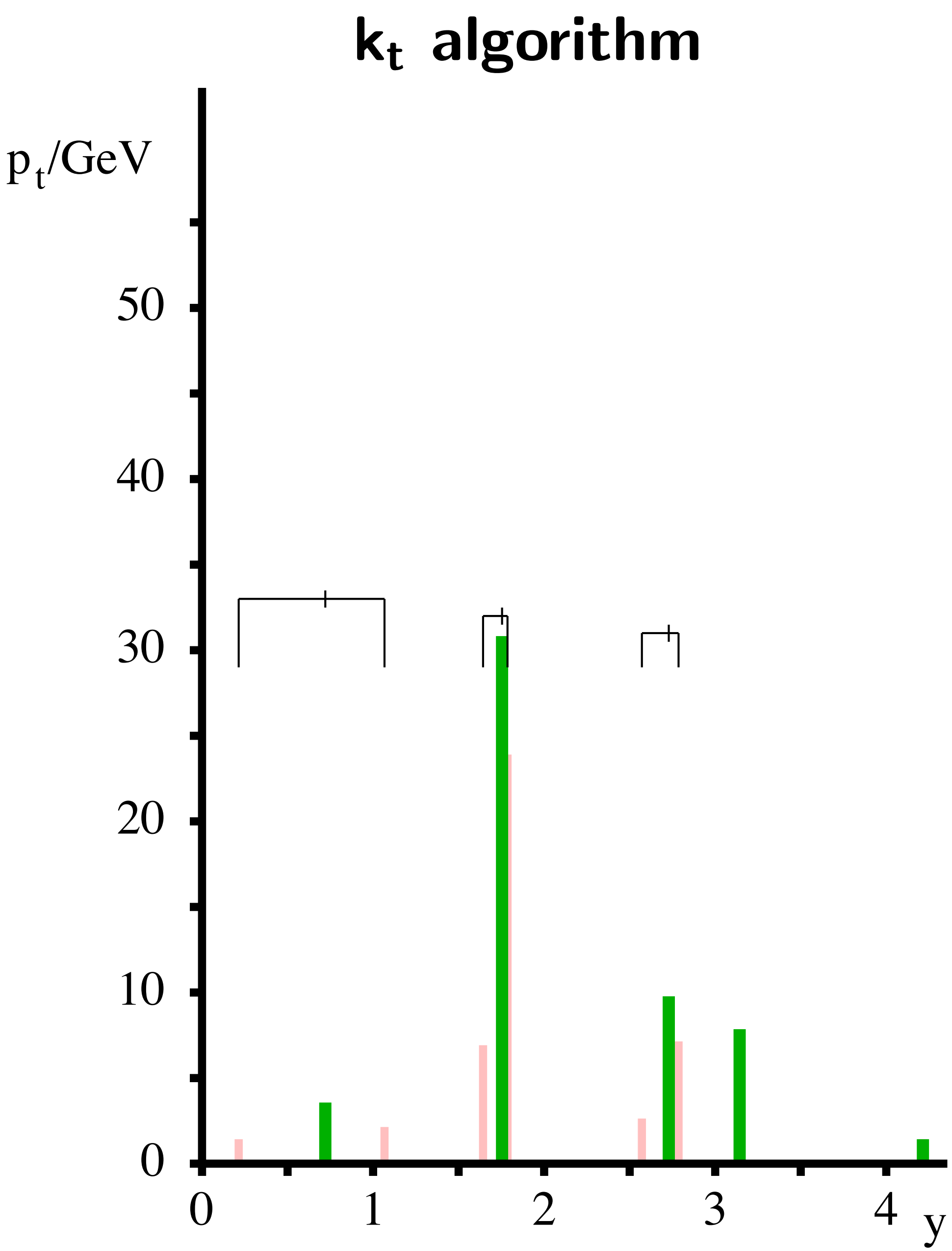
This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).



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).

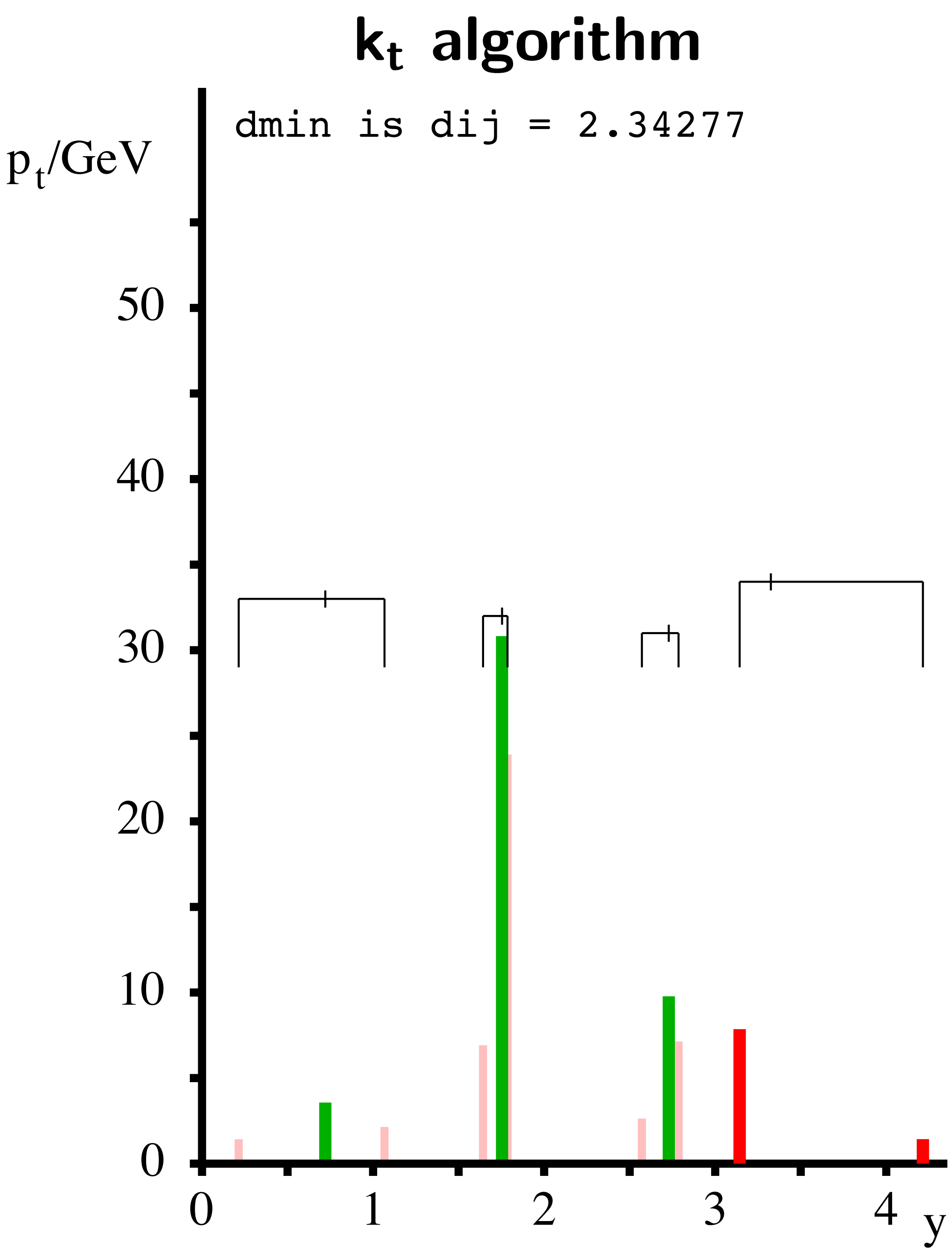
k_t clusters soft "junk" early on in the clustering



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

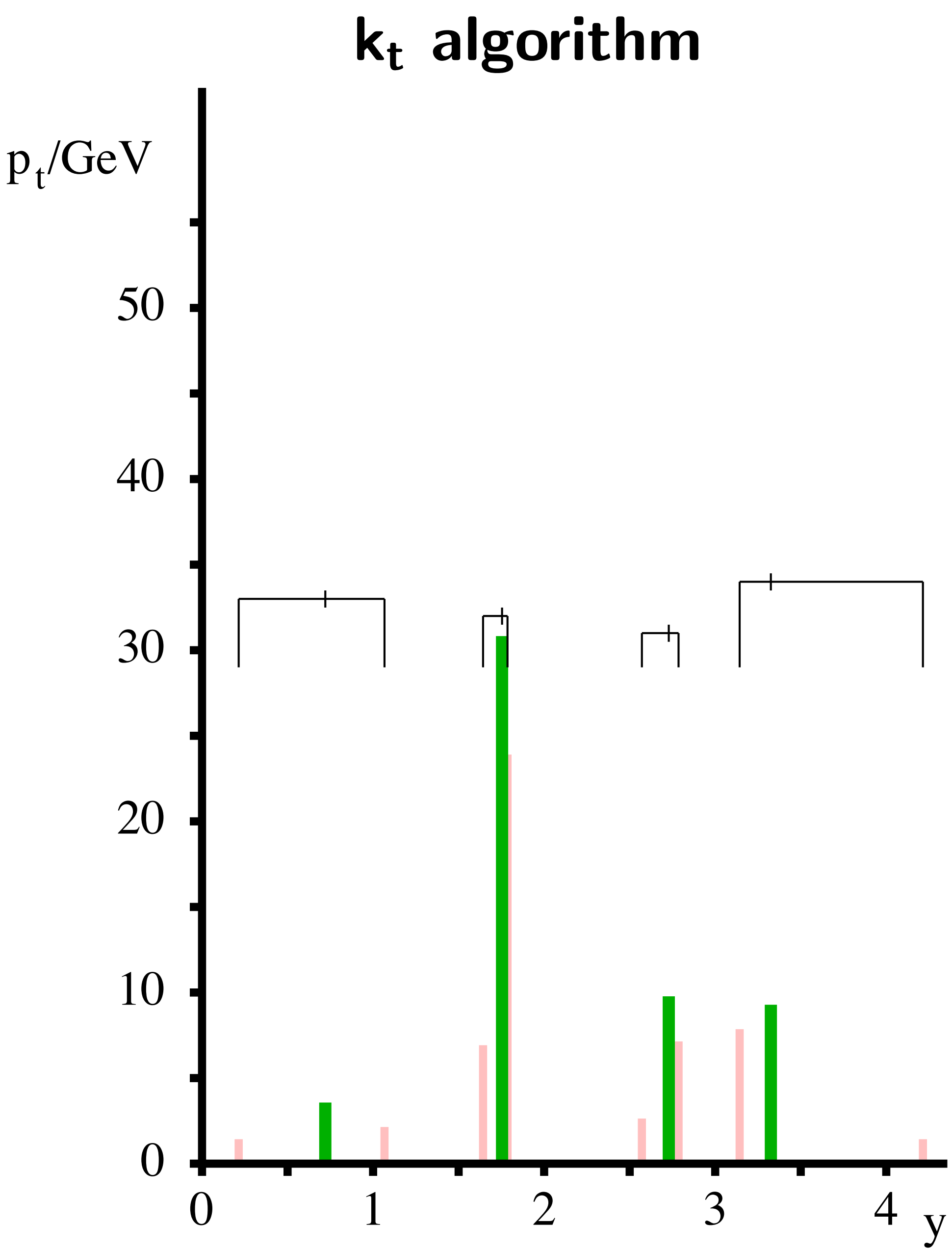


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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

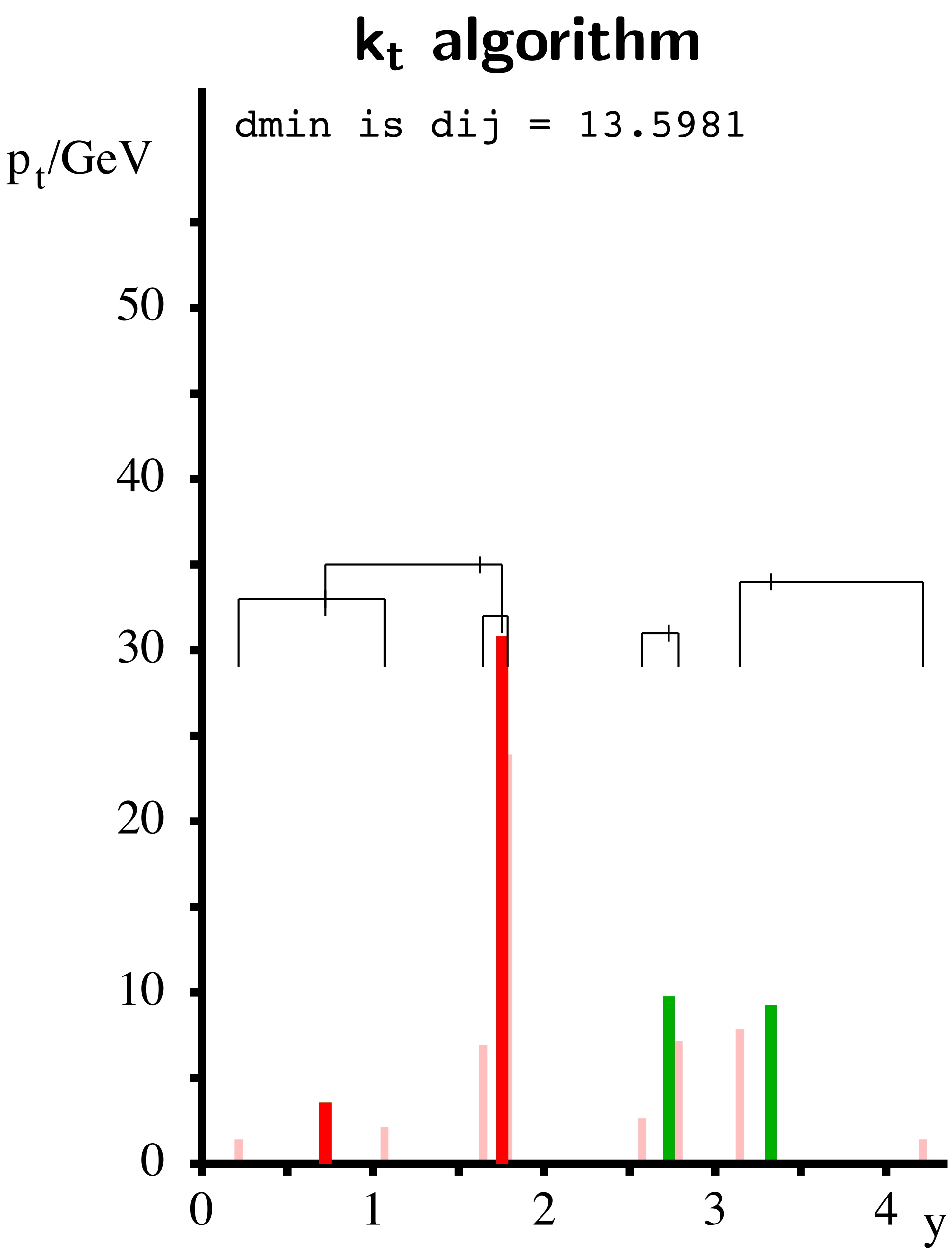
Identifying jet substructure: try out k_t



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).

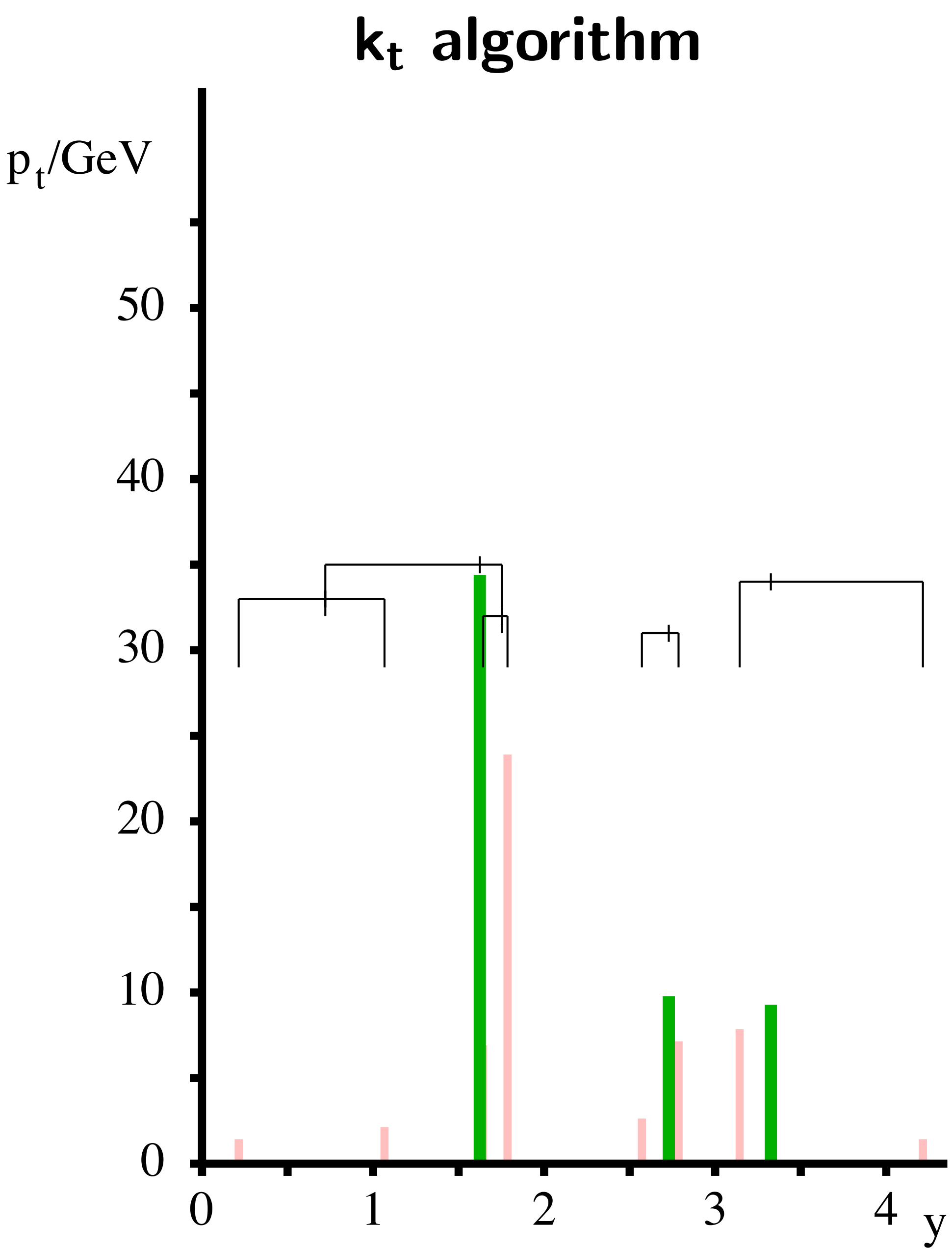
k_t clusters soft "junk" early on in the clustering



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).

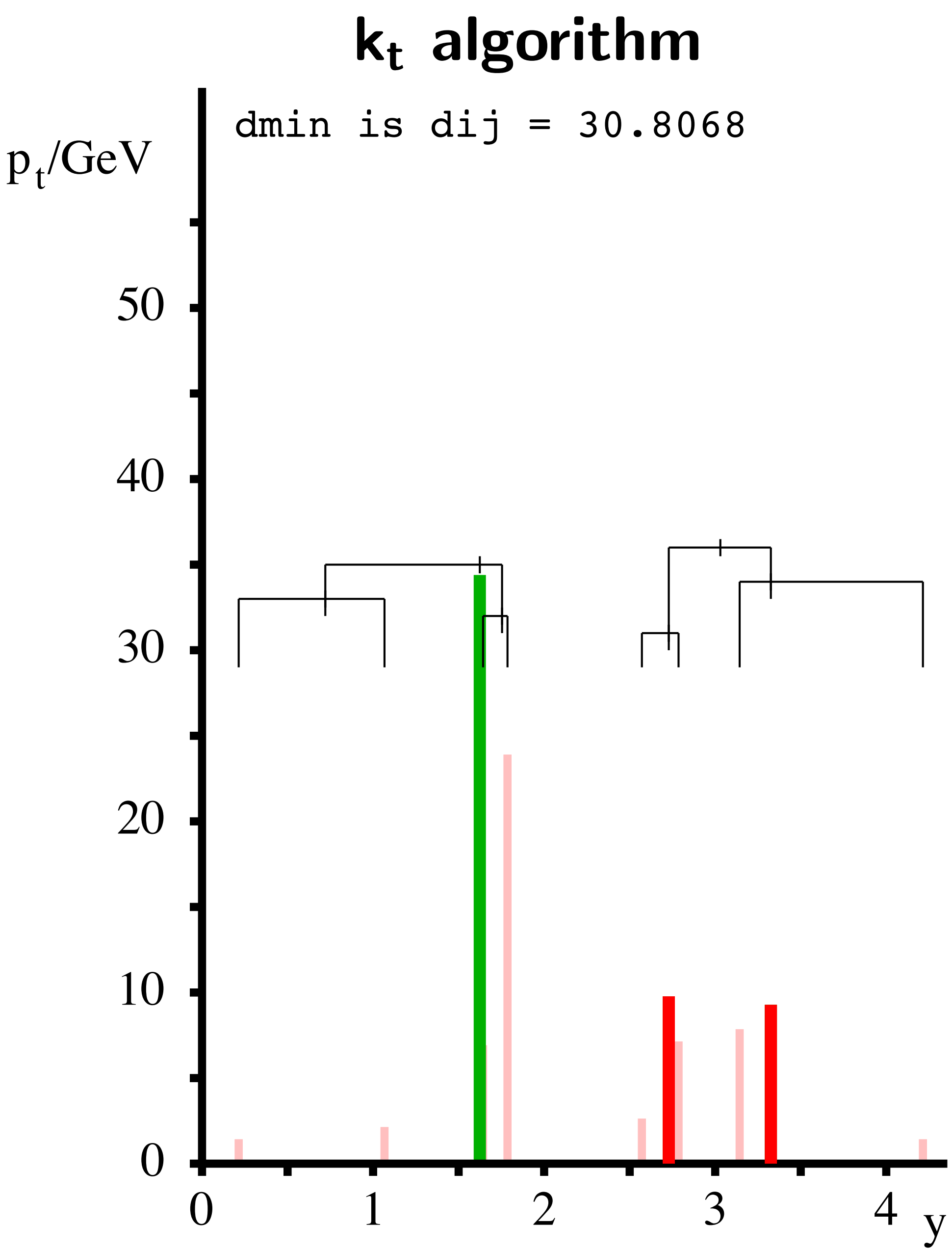
k_t clusters soft "junk" early on in the clustering



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

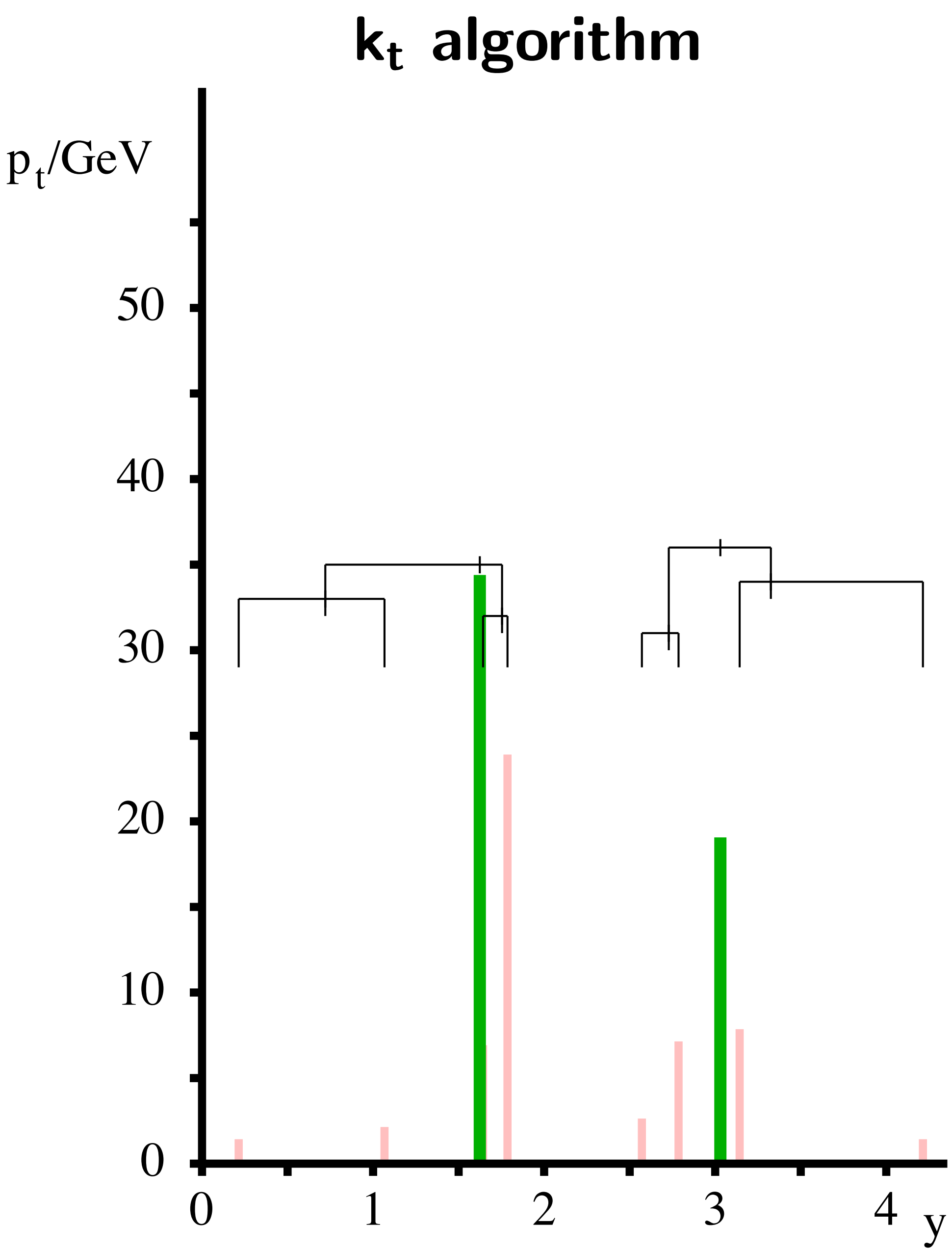


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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

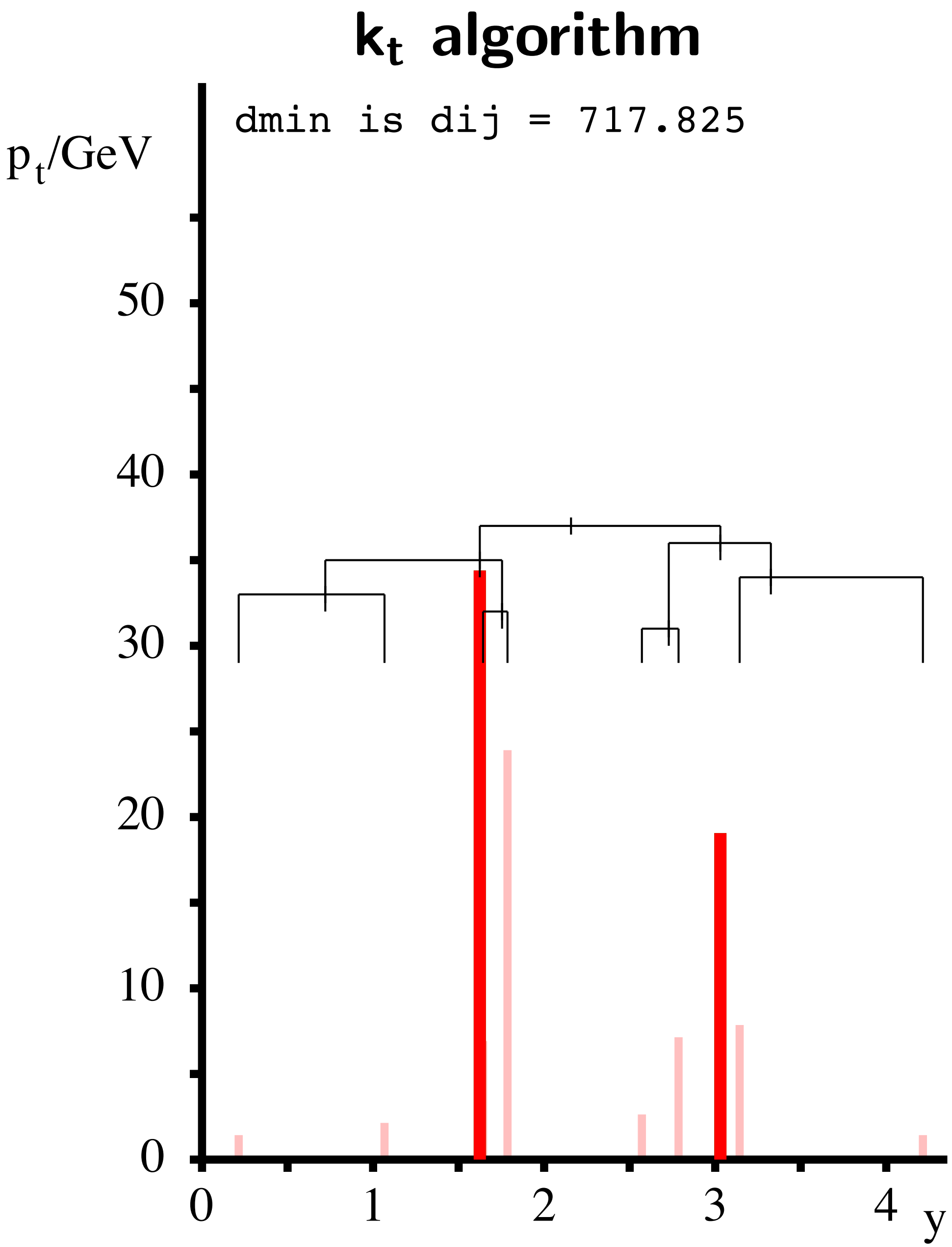
Identifying jet substructure: try out k_t



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

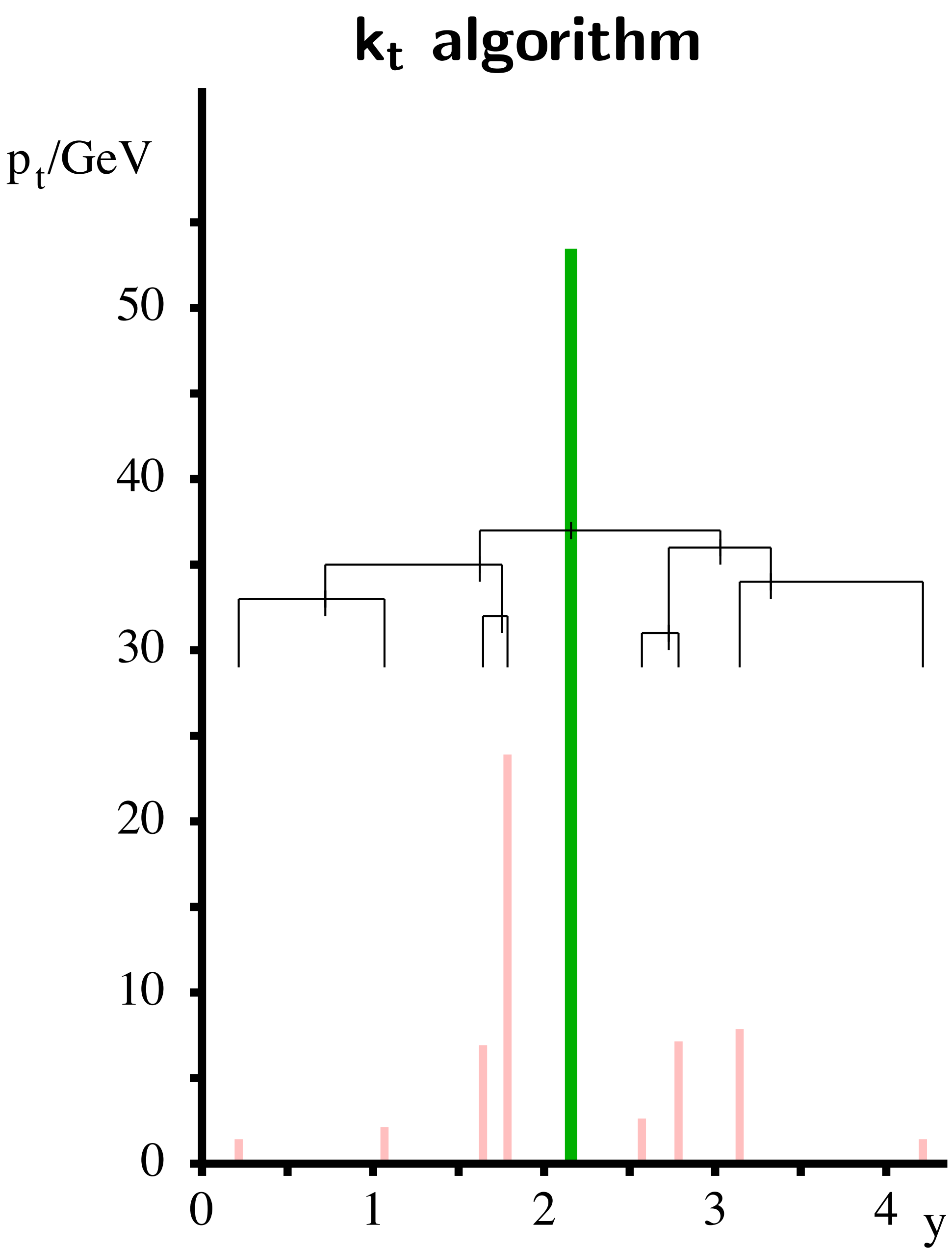


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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

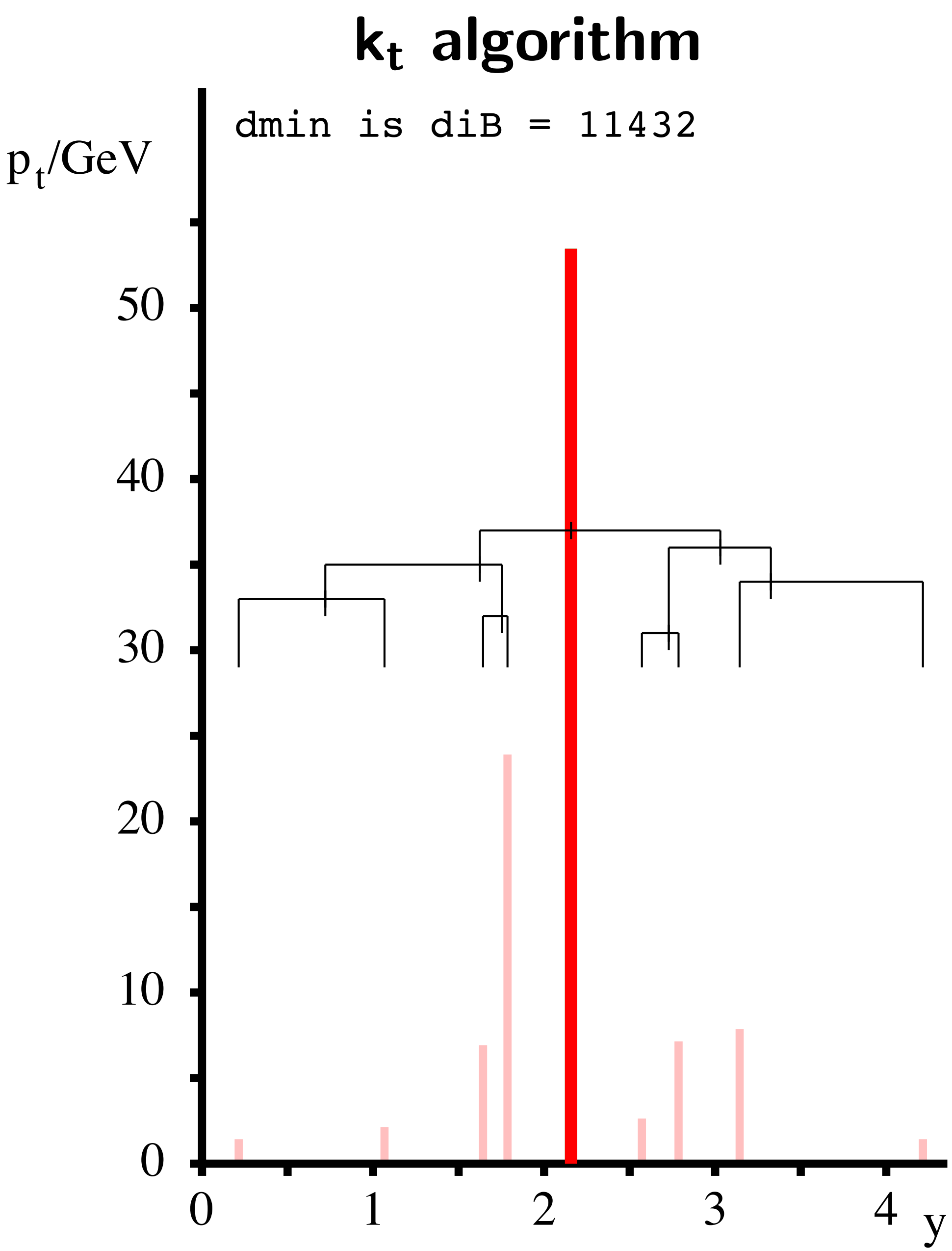


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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

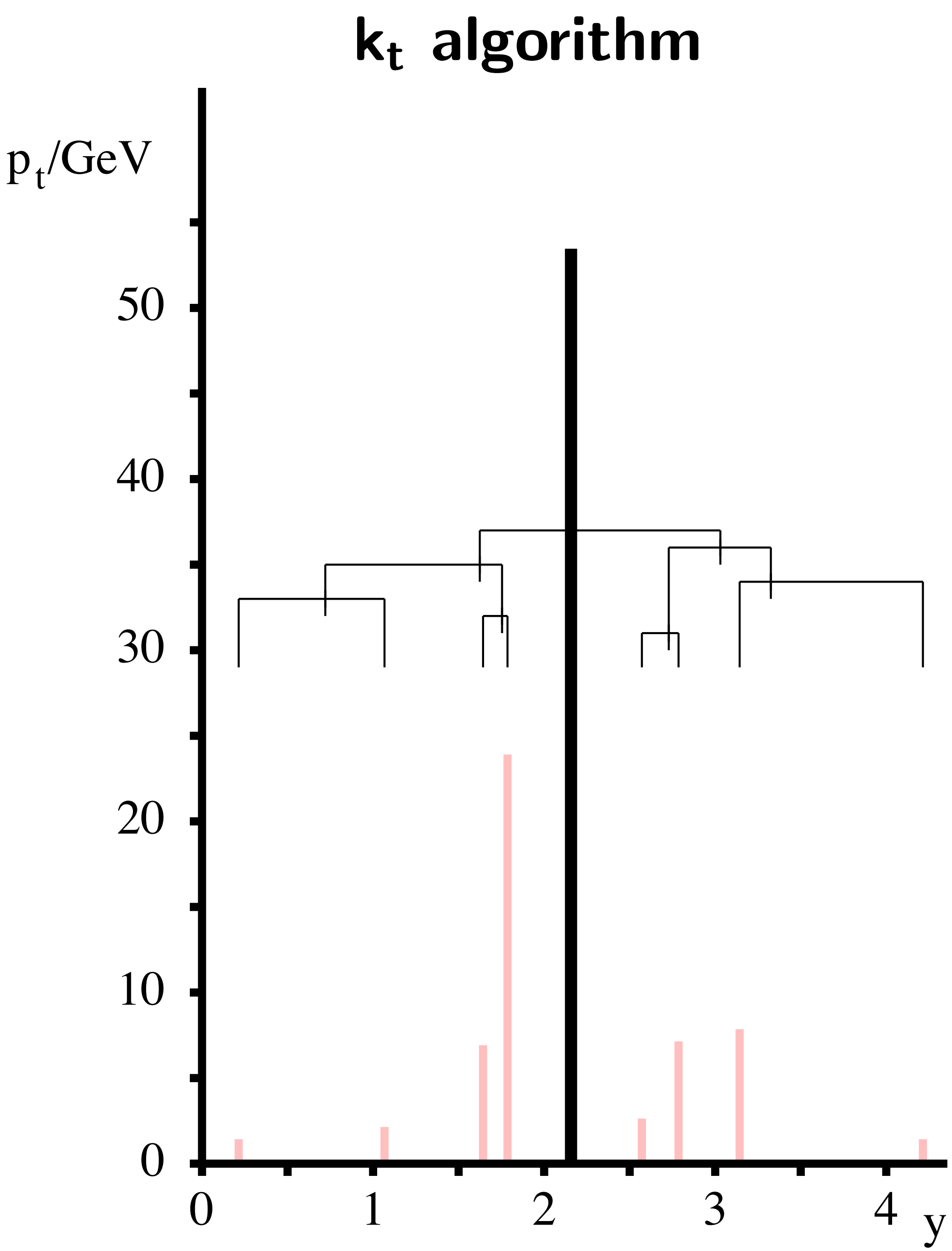


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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

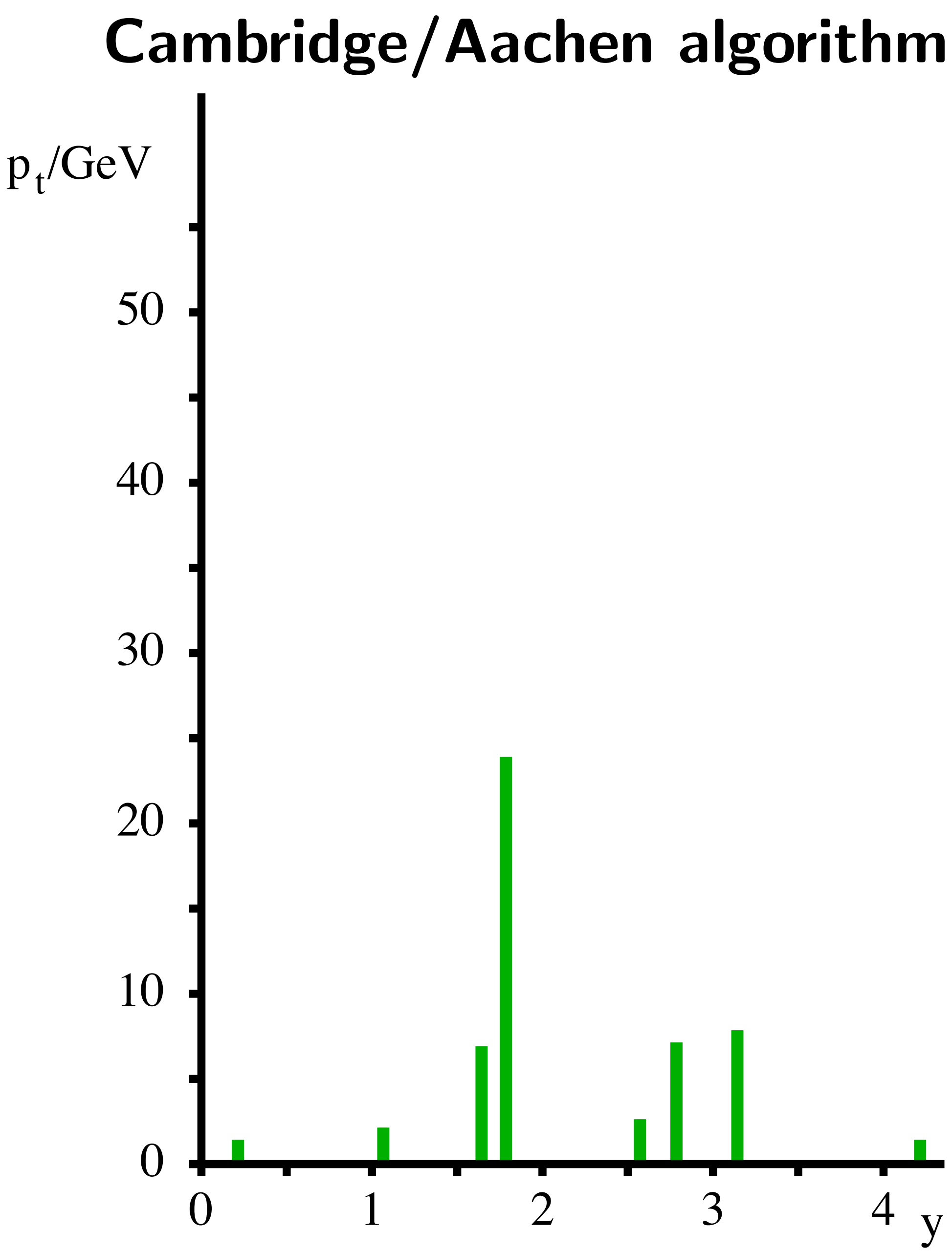
Its last step is to merge two hard pieces. Easily undone to identify underlying kinematics

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

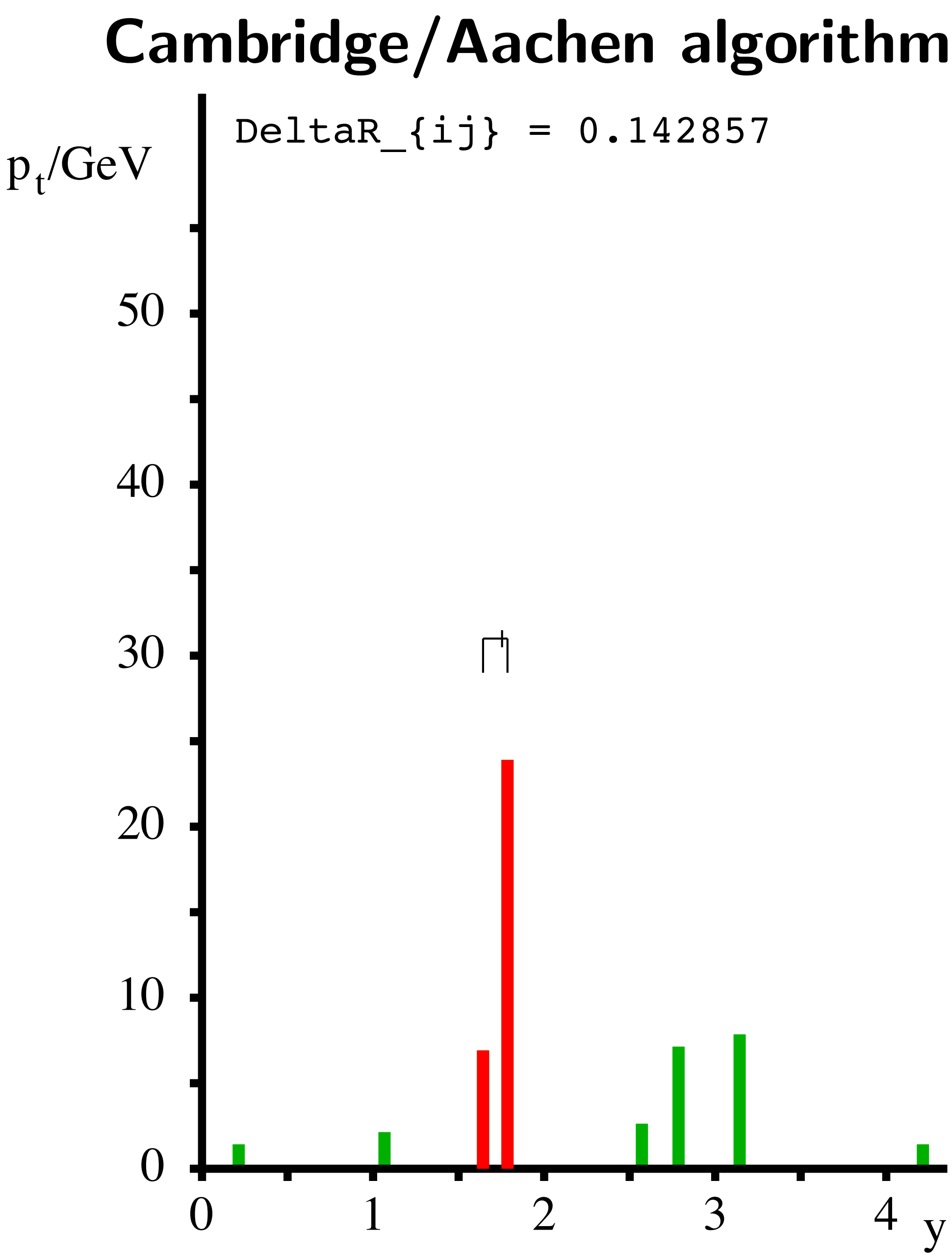
Seymour '93

Butterworth, Cox & Forshaw '02

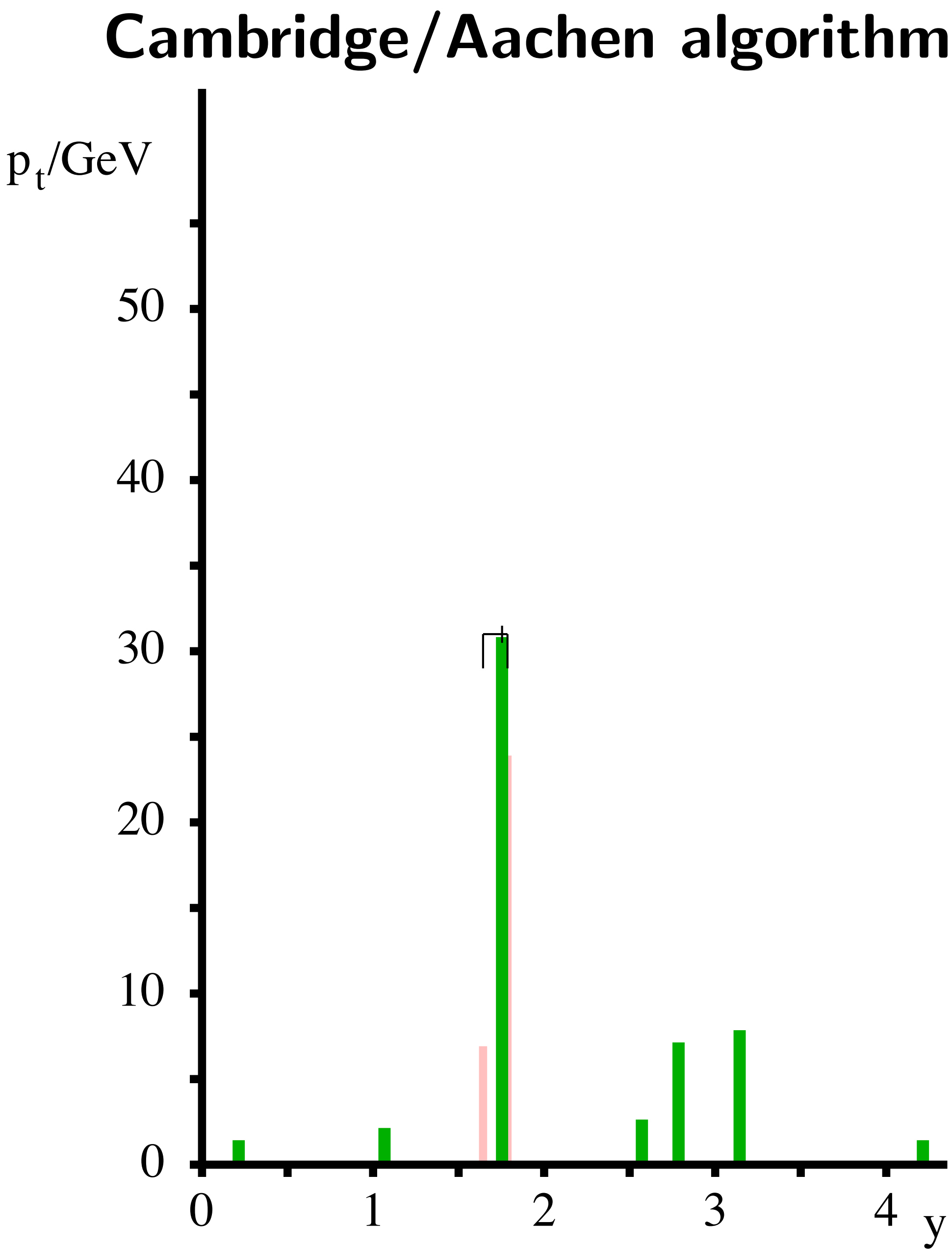
Cambridge/Aachen



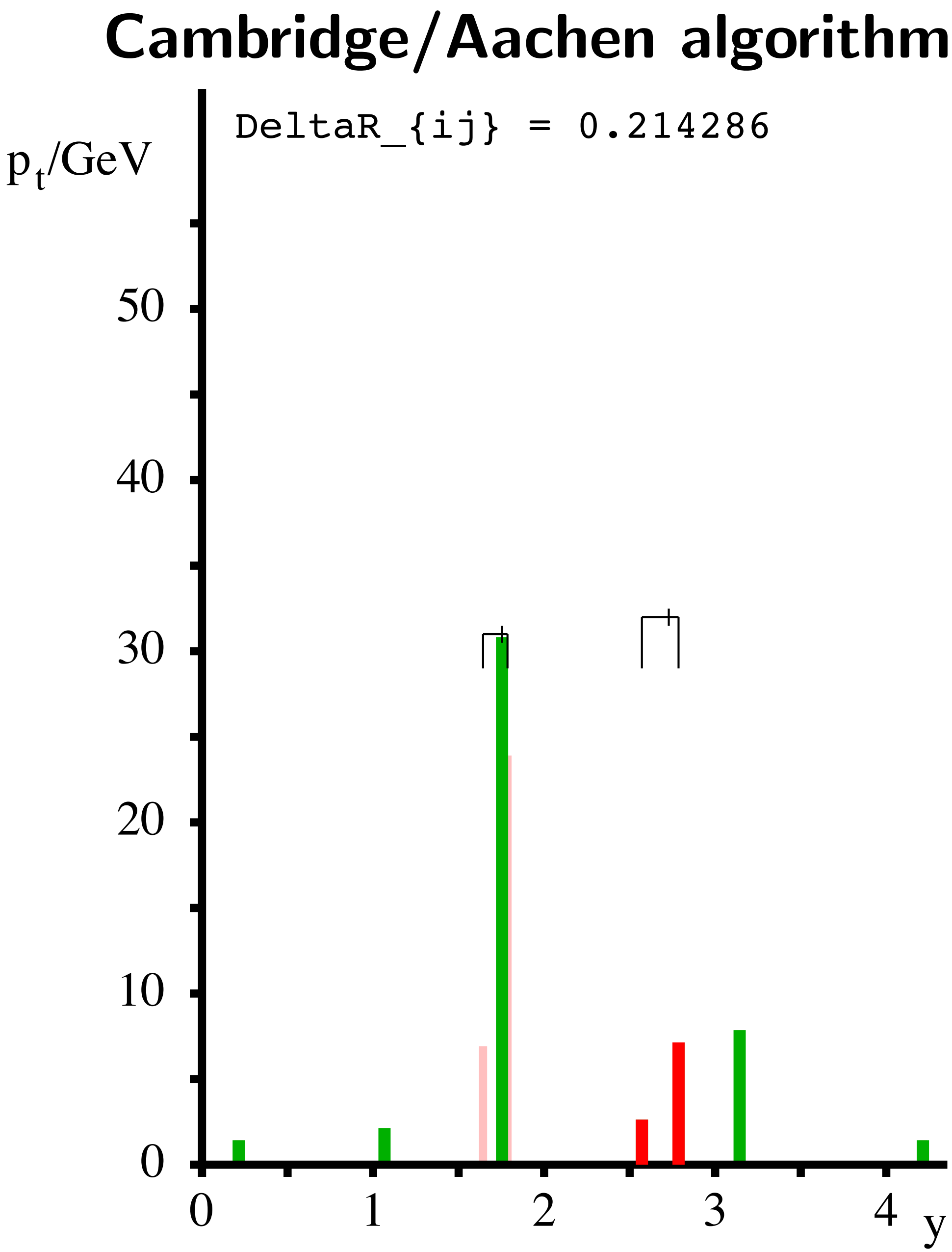
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?



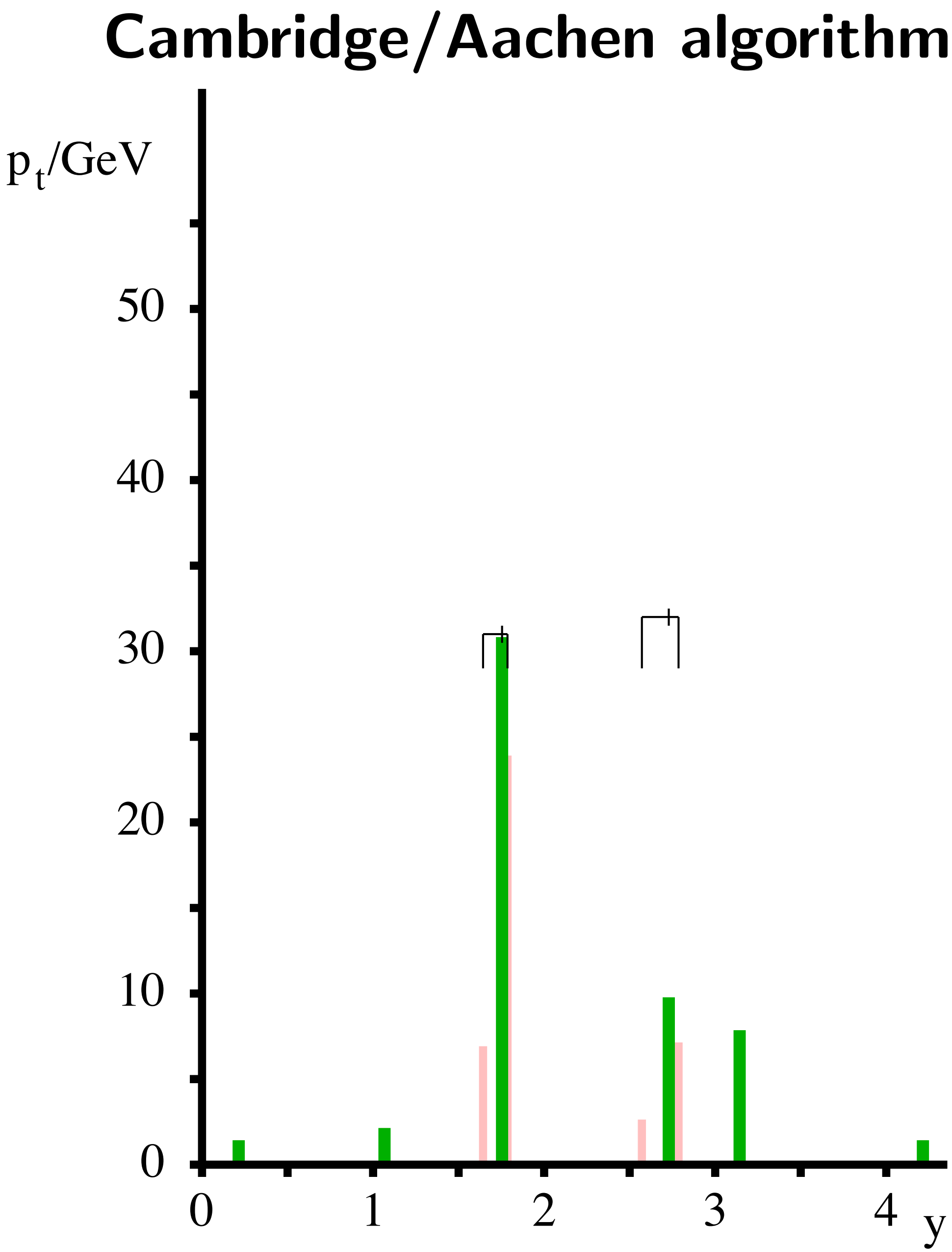
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?



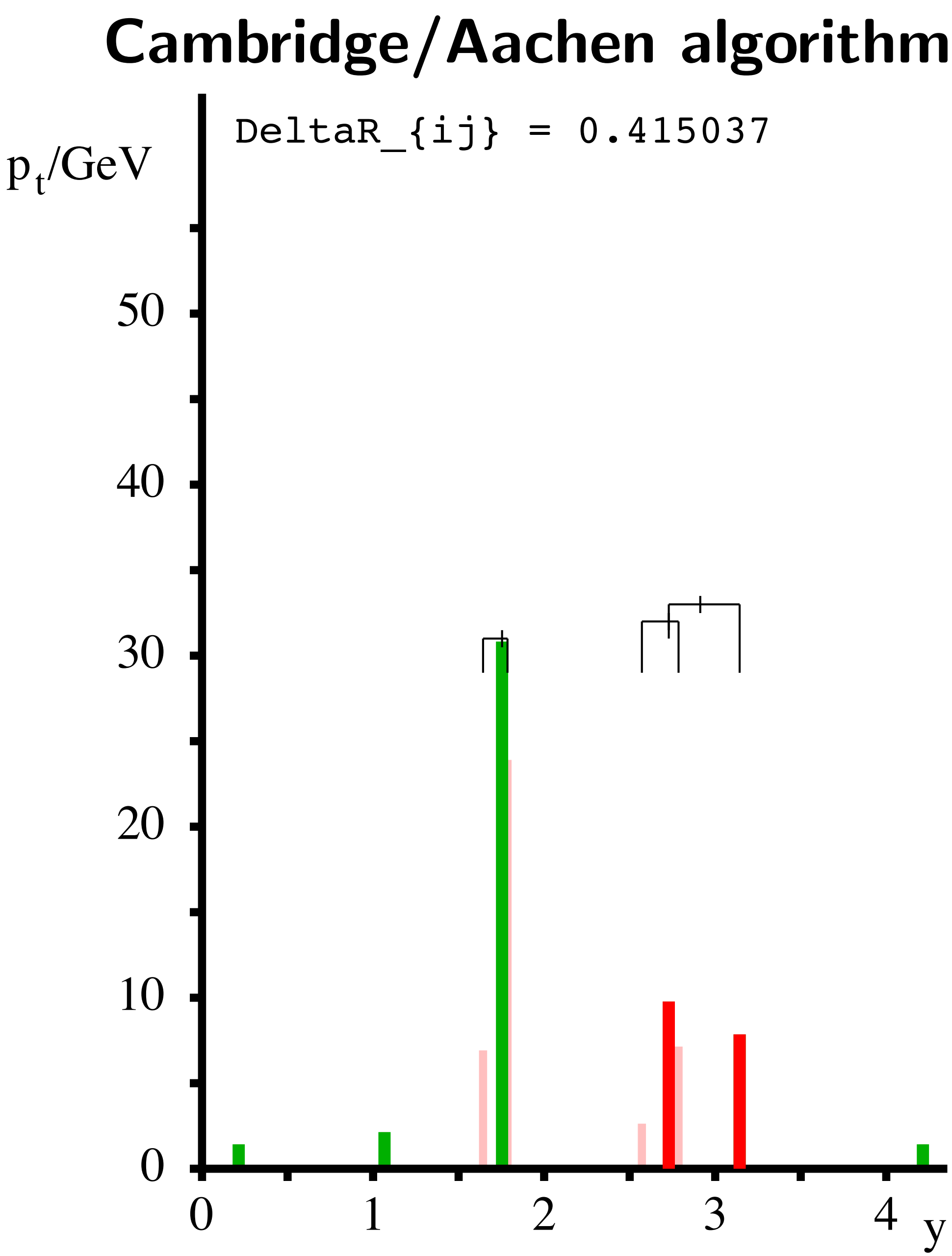
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?



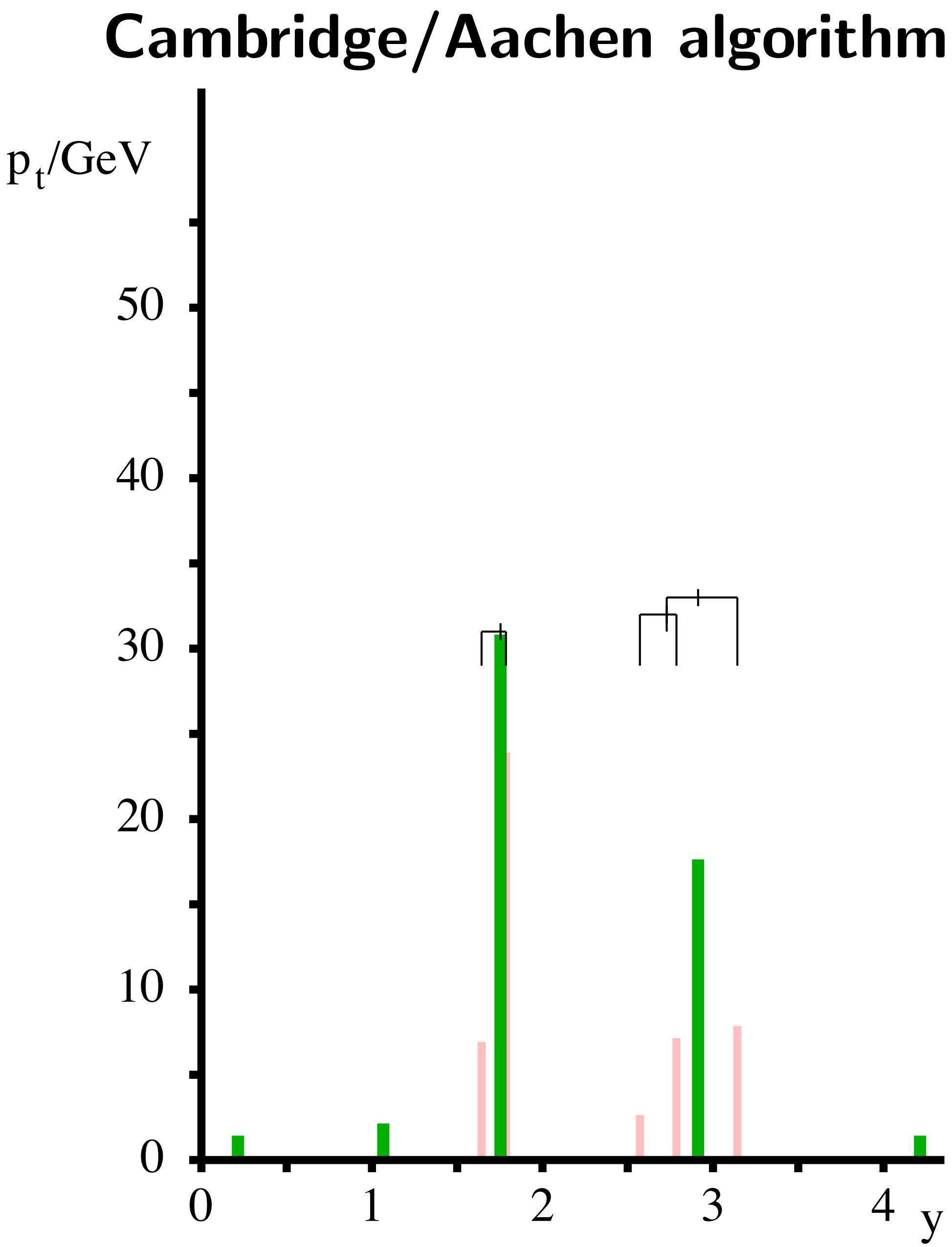
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?



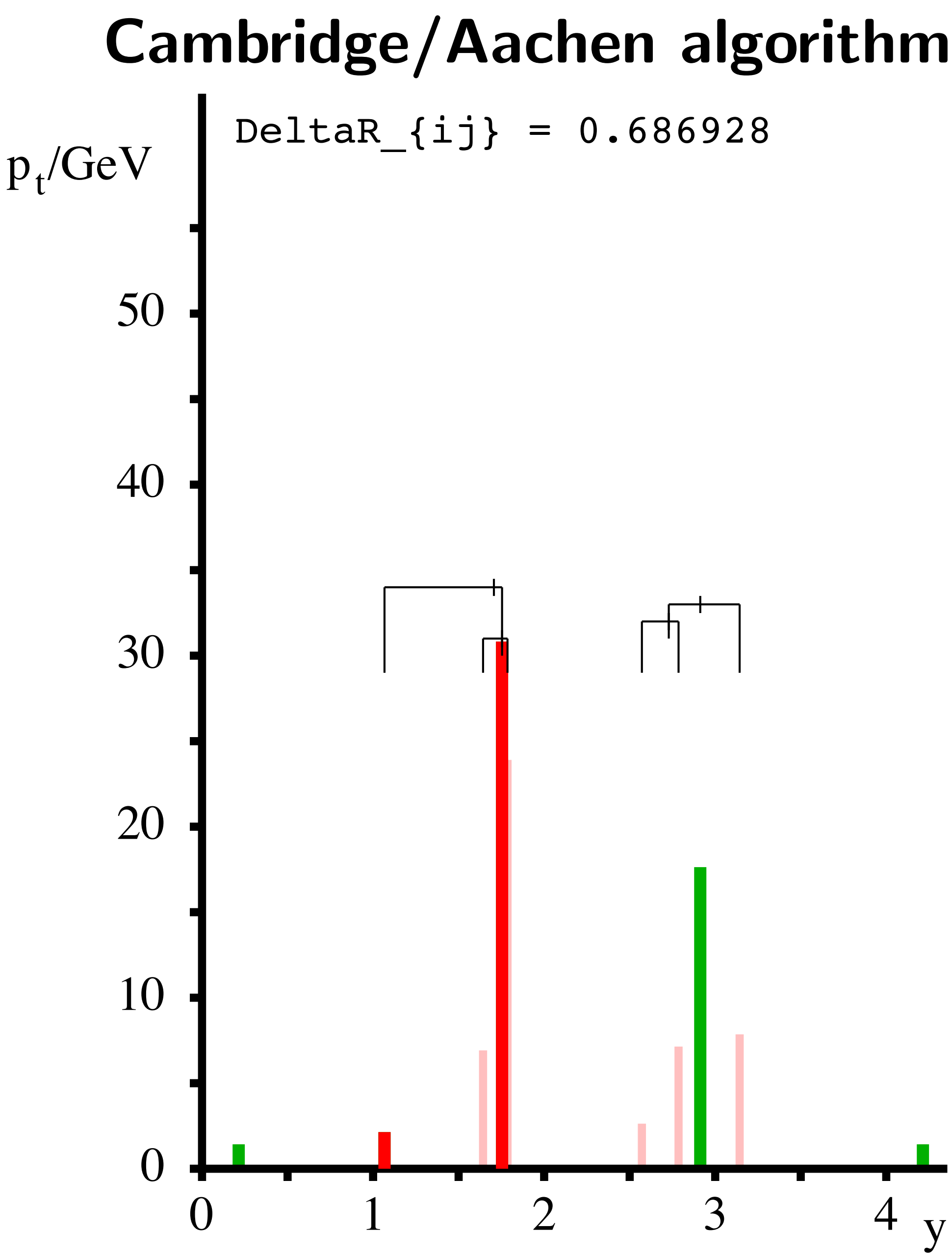
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?



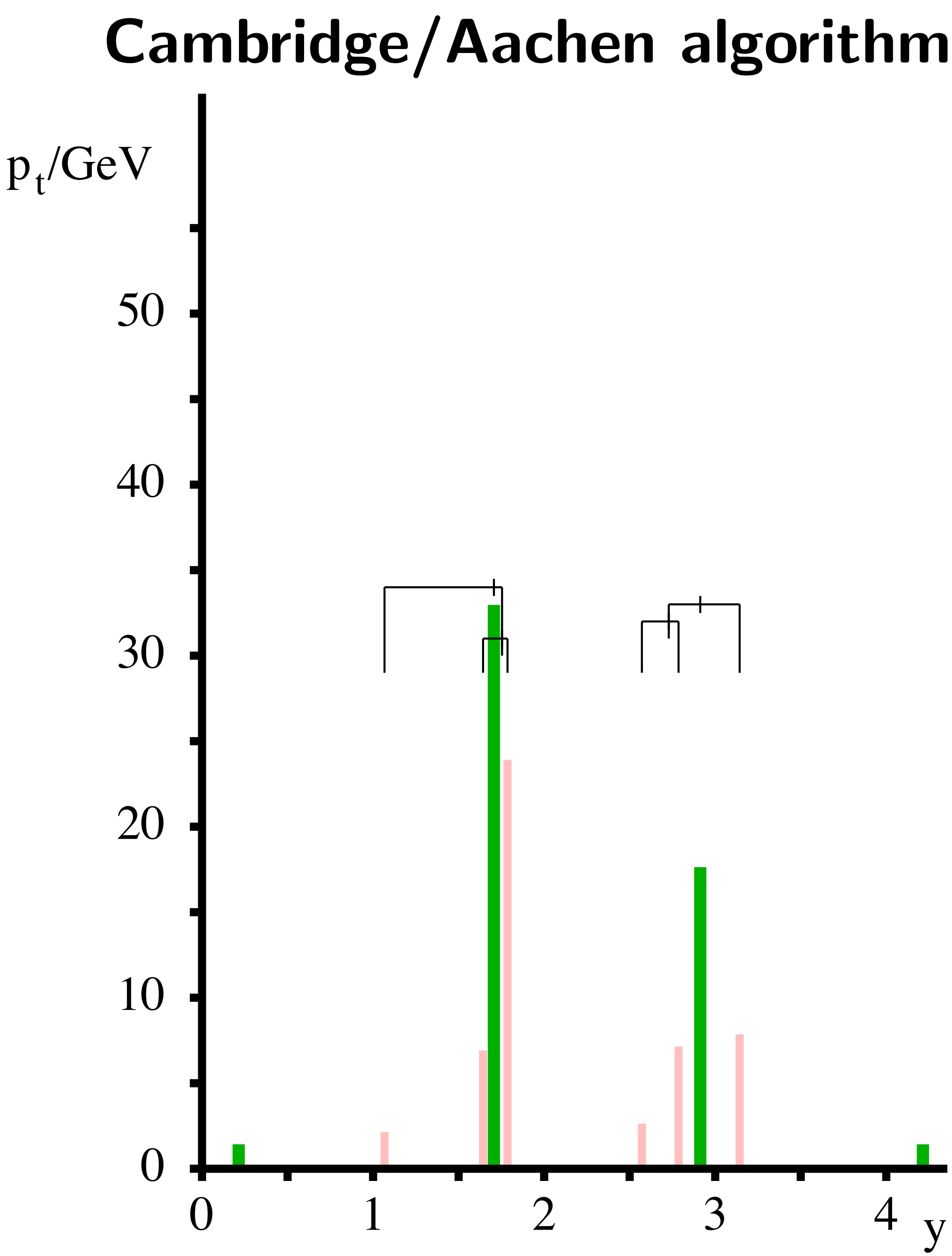
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?



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

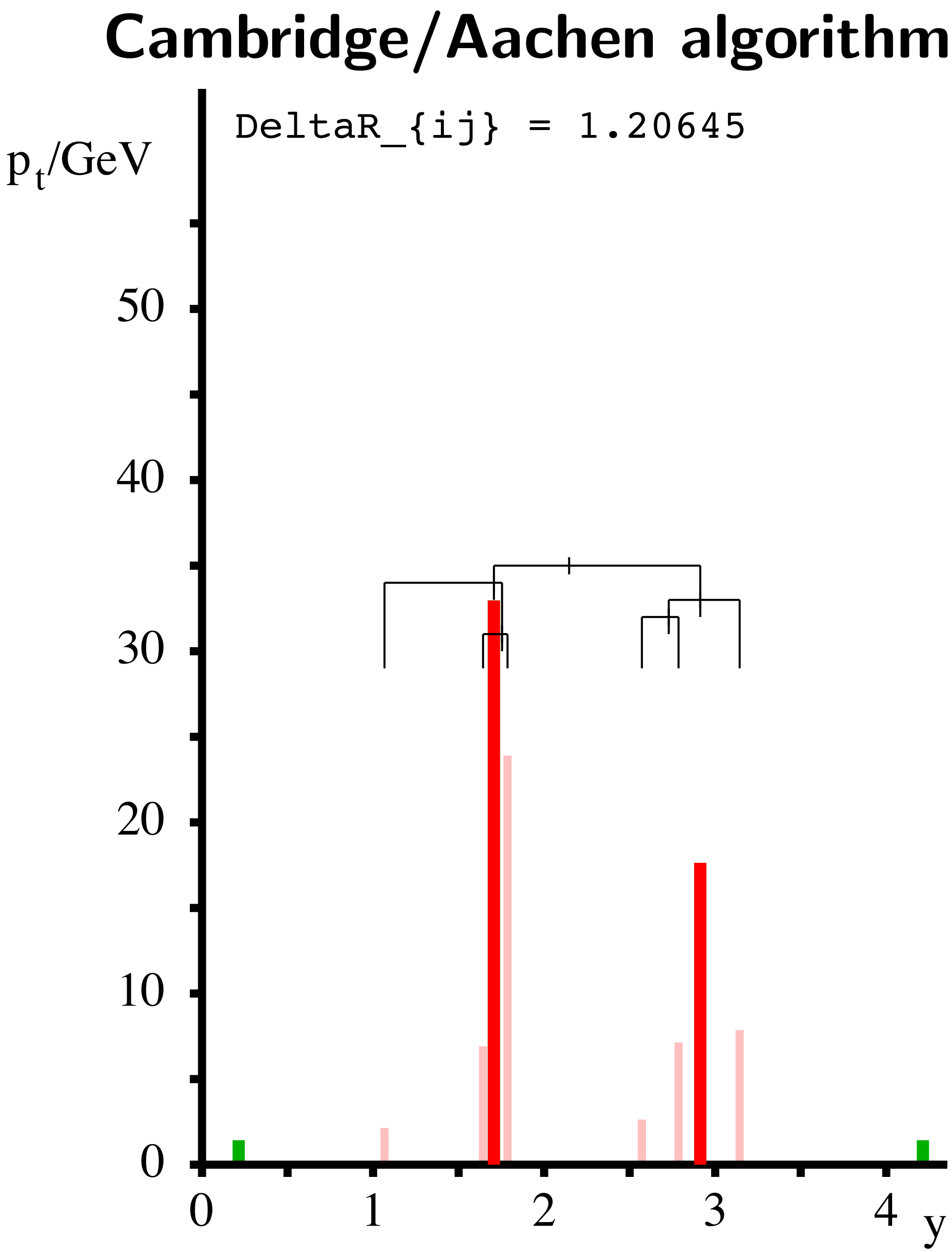


How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?



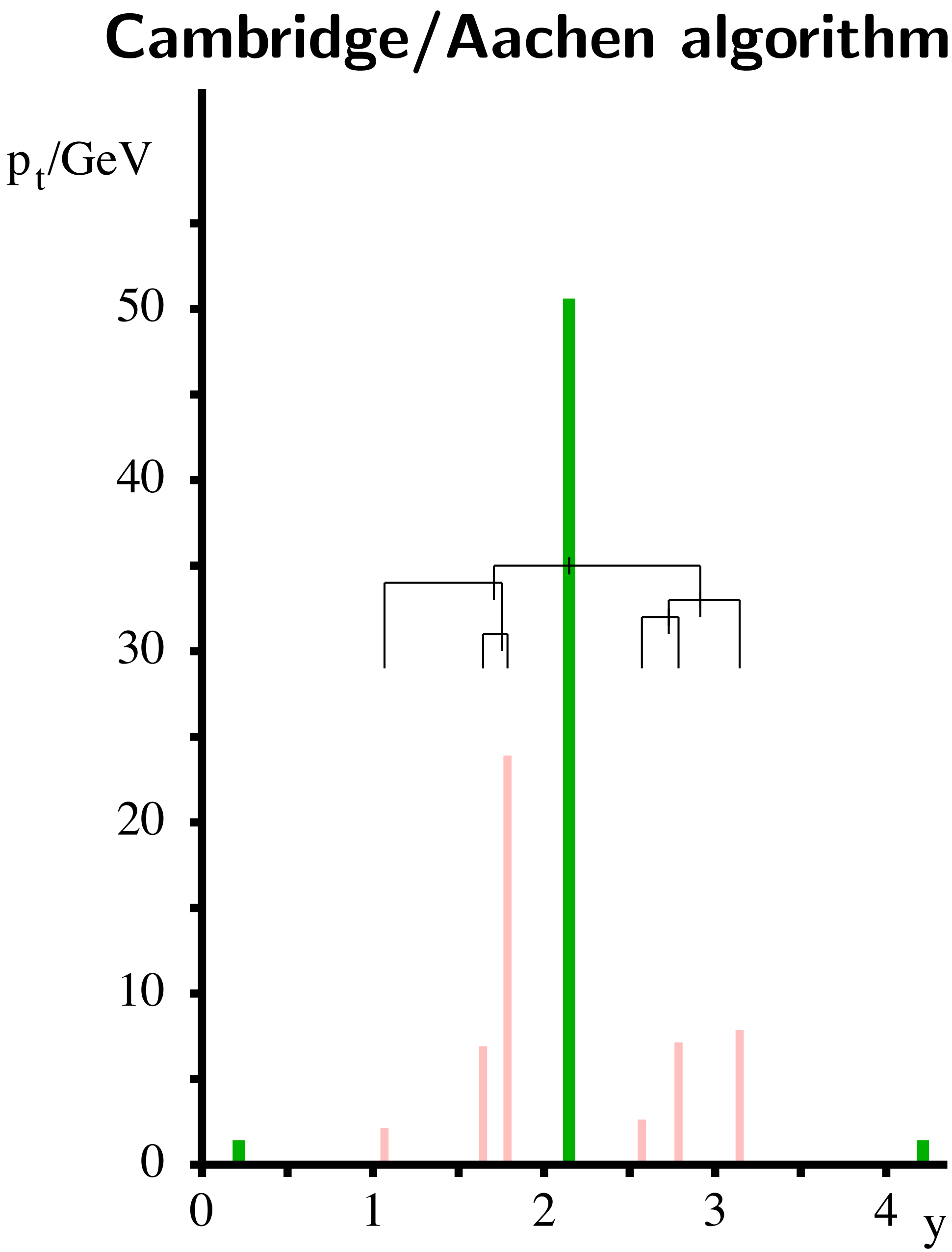
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination



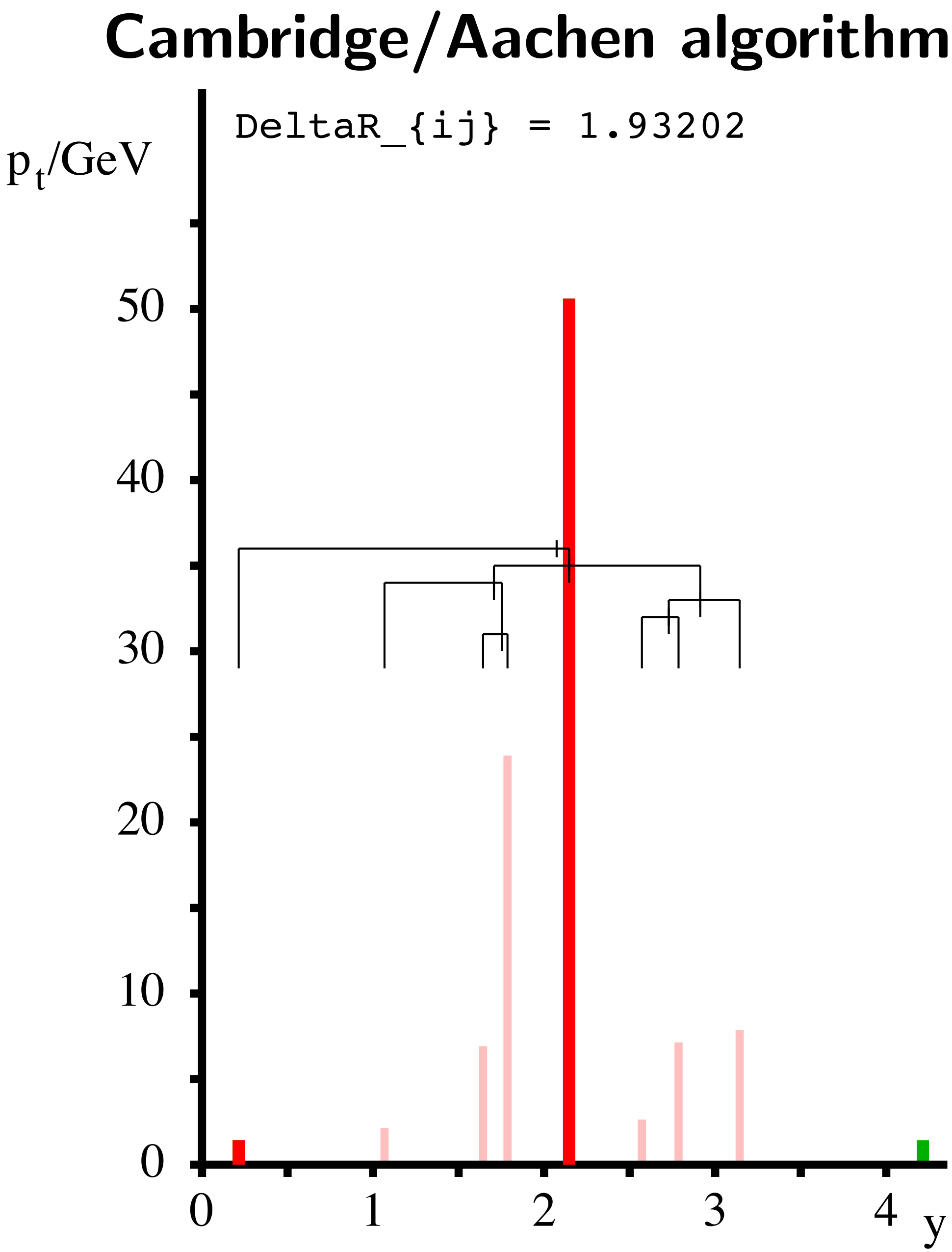
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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



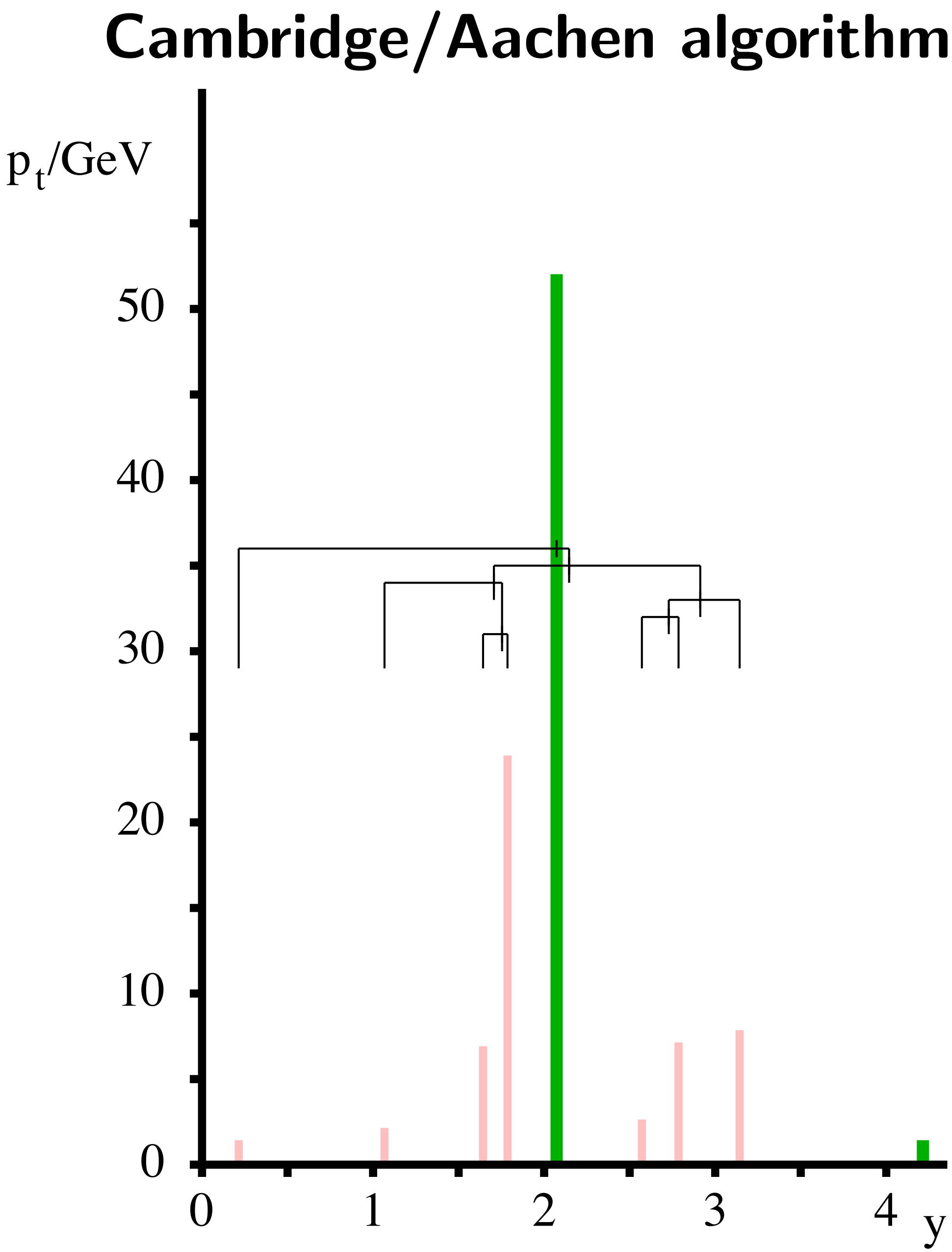
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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



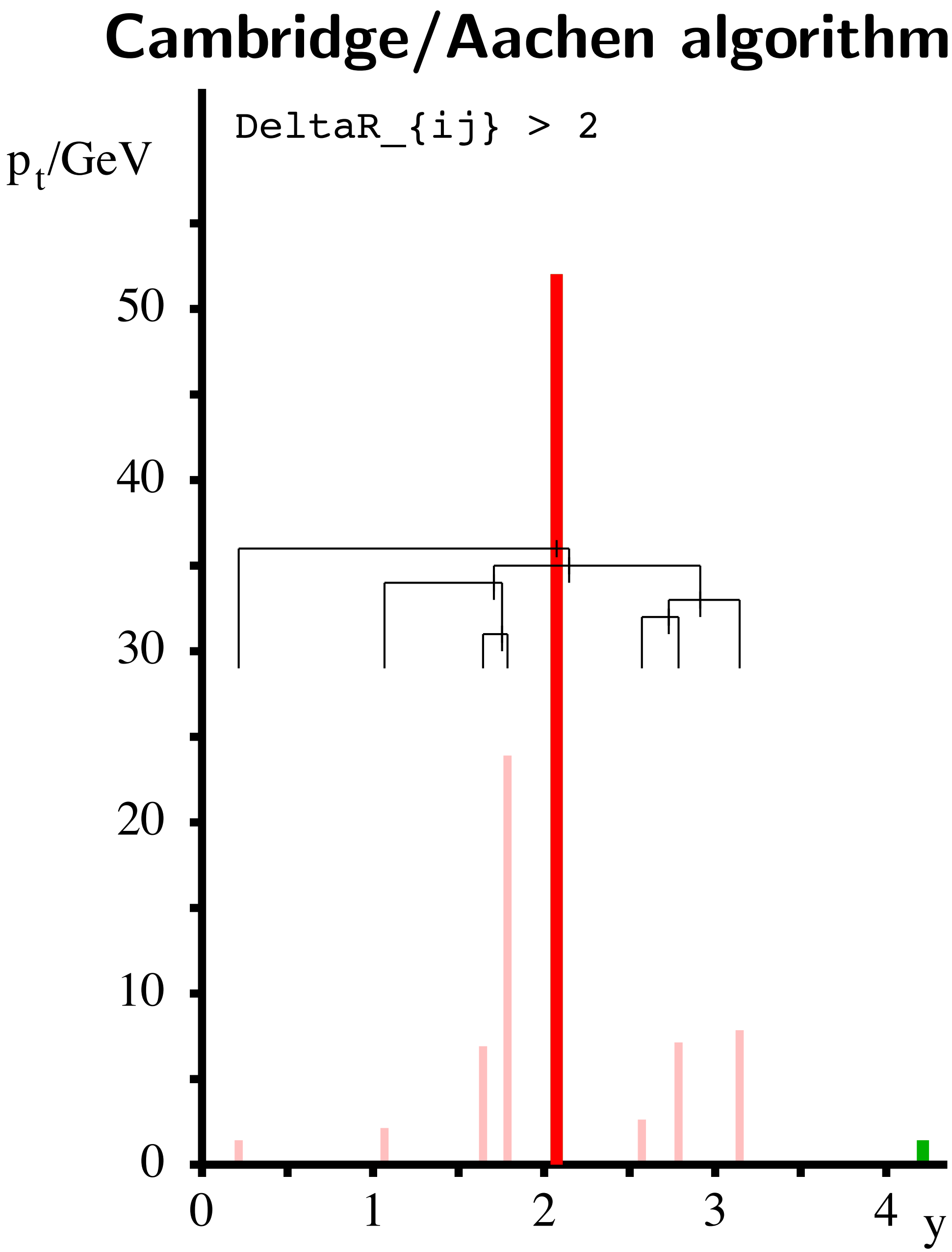
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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



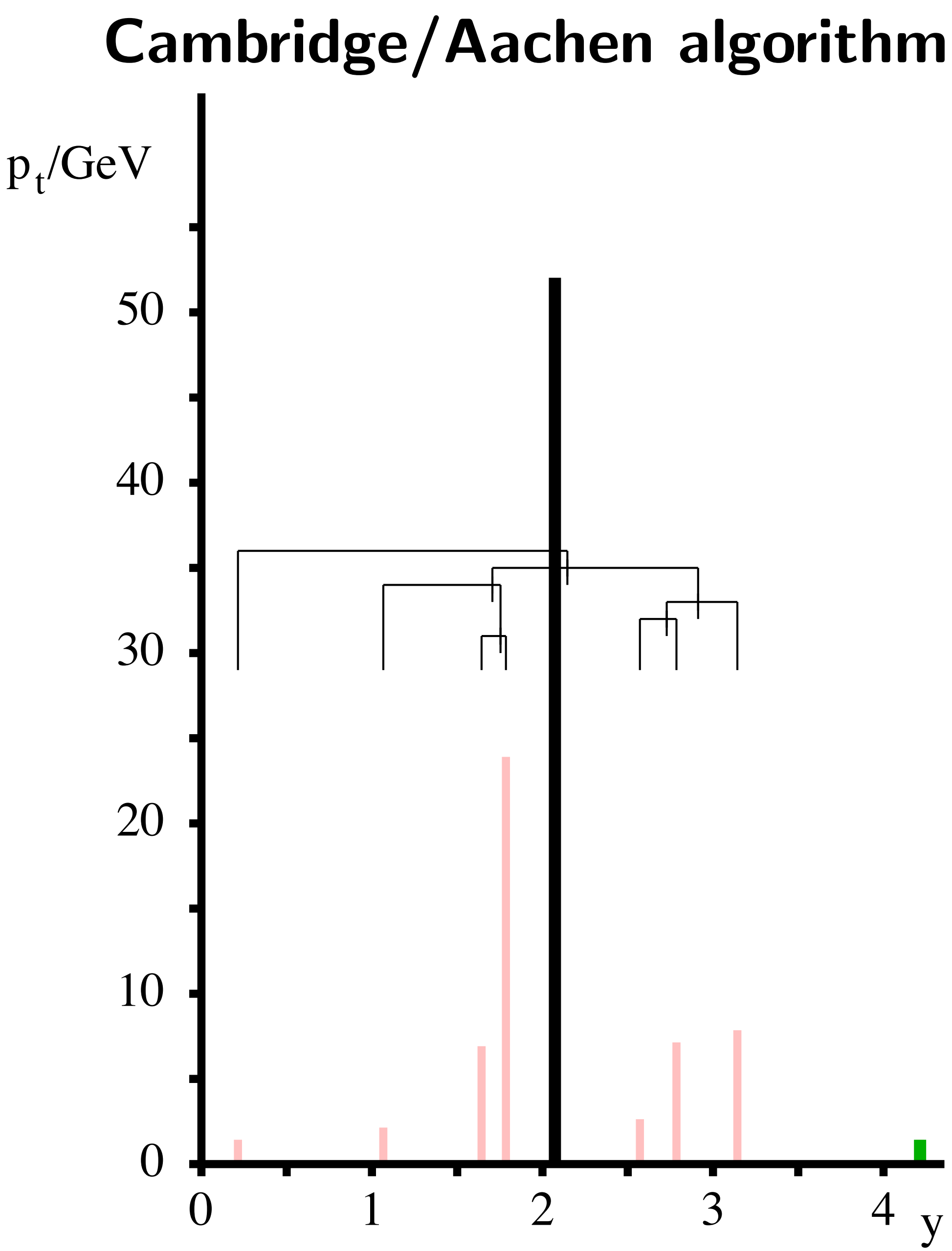
How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

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?

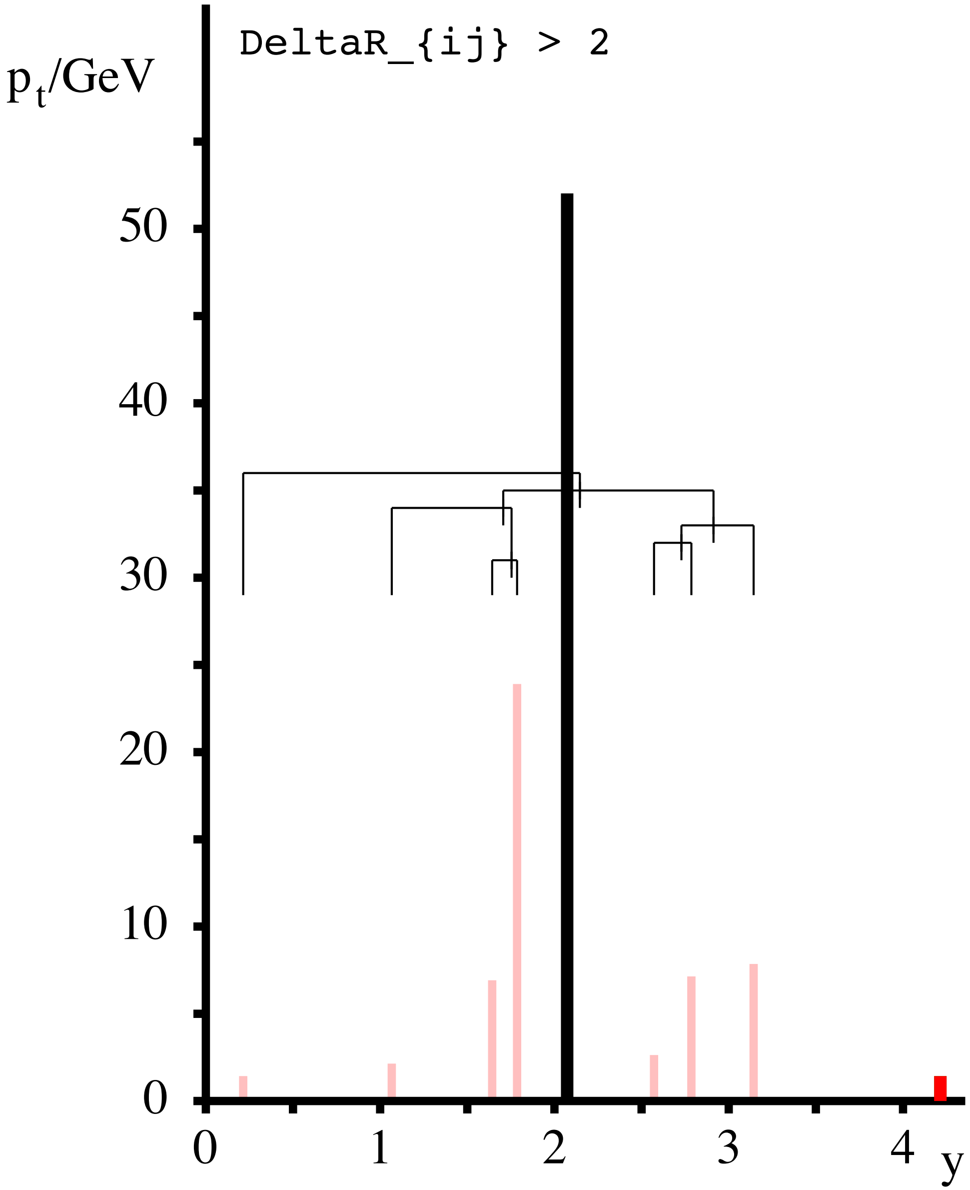
C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk



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

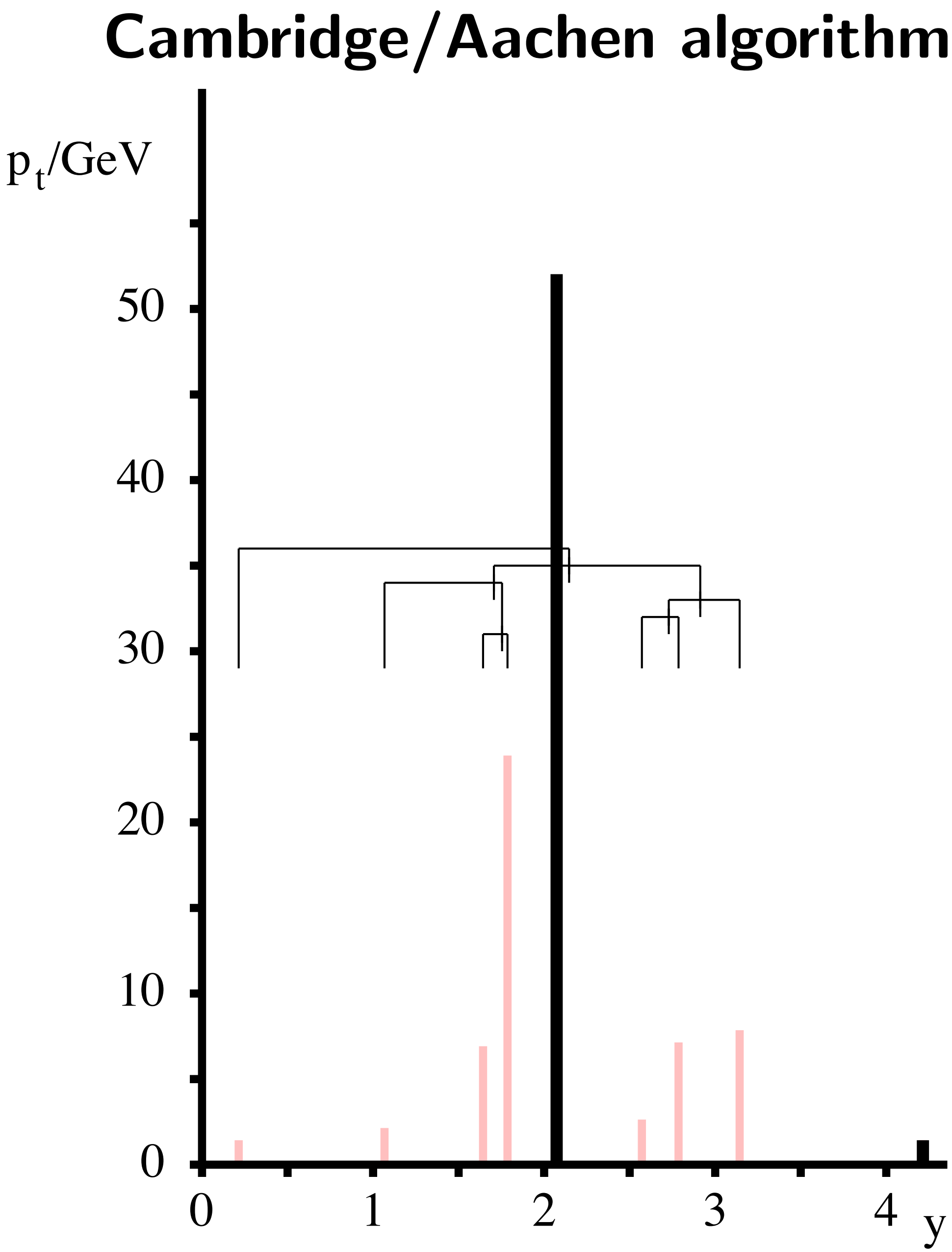
C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk

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?

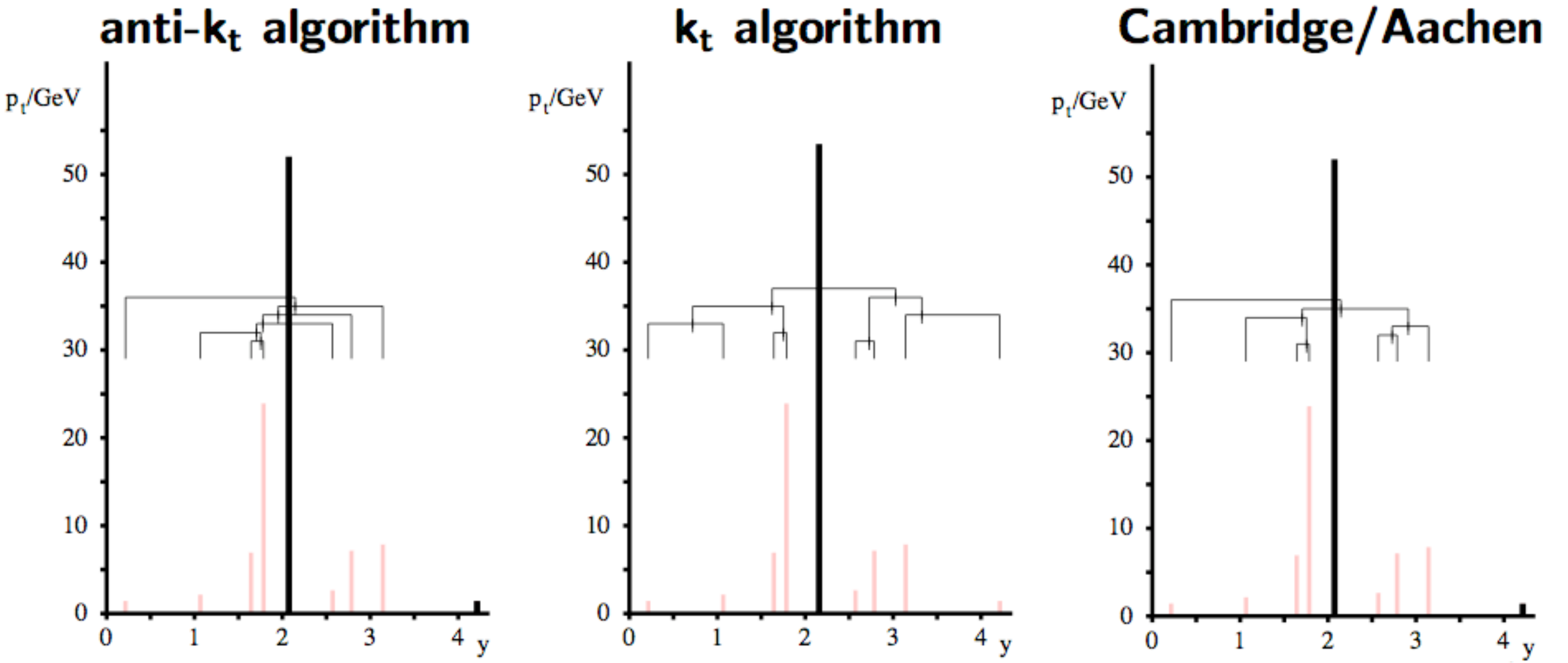
C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk

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

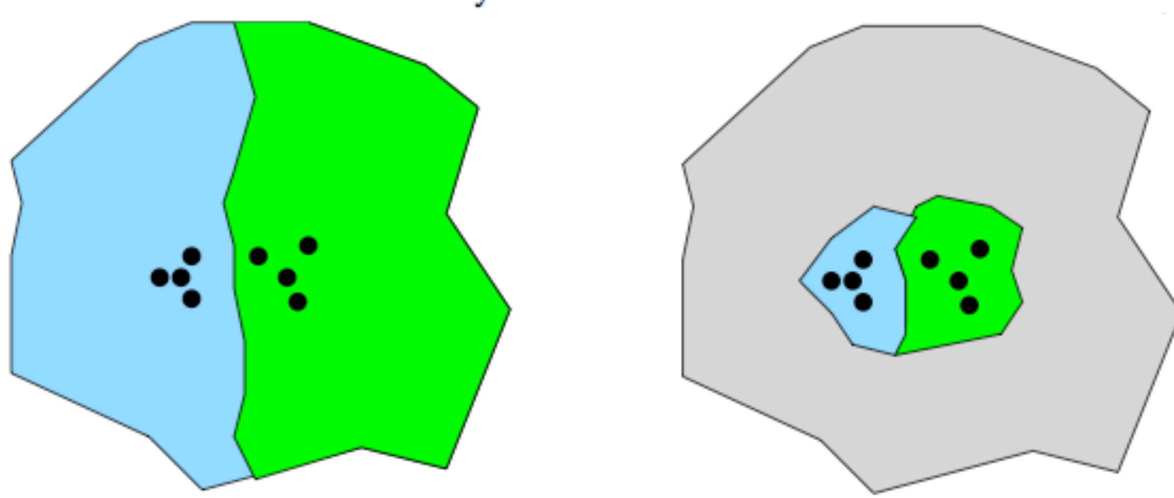
Butterworth, Davison, Rubin & GPS '08
Kaplan, Schwartz, Reherman & Tweedie '08
Butterworth, Ellis, Rubin & GPS '09
Ellis, Vermilion & Walsh '09

Slide by
Gavin Salam

Hierarchical substructure



Undo the last clustering step(s) →



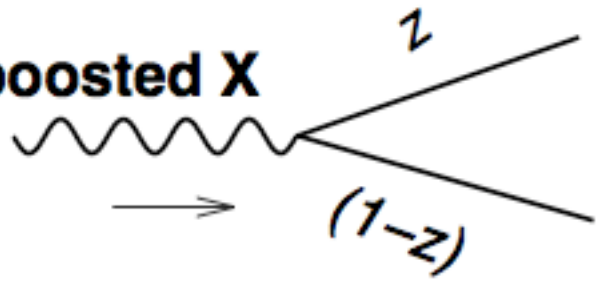
The IRC safe algorithms

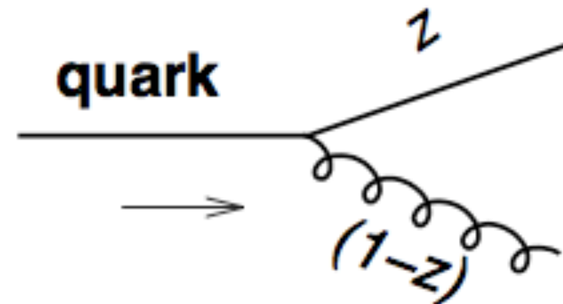
	Speed	Regularity	UE contamination	Backreaction	Hierarchical substructure
k_t	☺ ☺ ☺	☂	☂☂	☁☁	☺ ☺
Cambridge /Aachen	☺ ☺ ☺	☂	☂	☁☁	☺ ☺ ☺
anti-k_t	☺ ☺ ☺	☺ ☺	☁/ ☺	☺ ☺	✘
SISCone	☺	☁	☺ ☺	☁	✘

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**

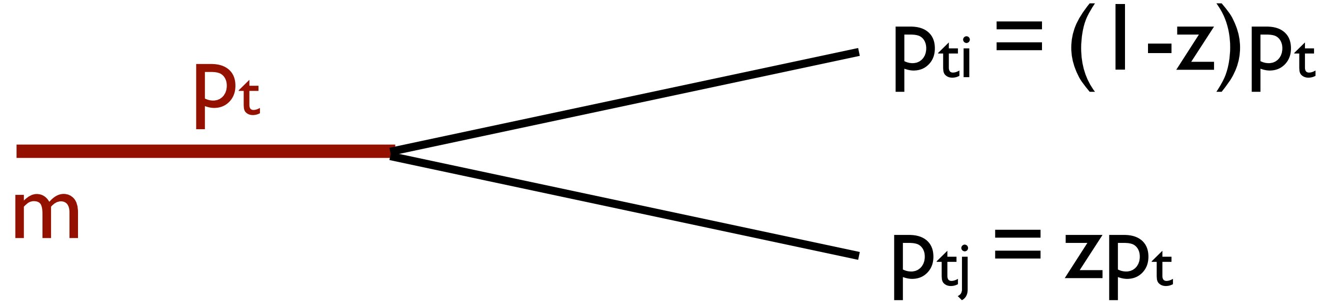
Potential tagger: asymmetric splitting

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

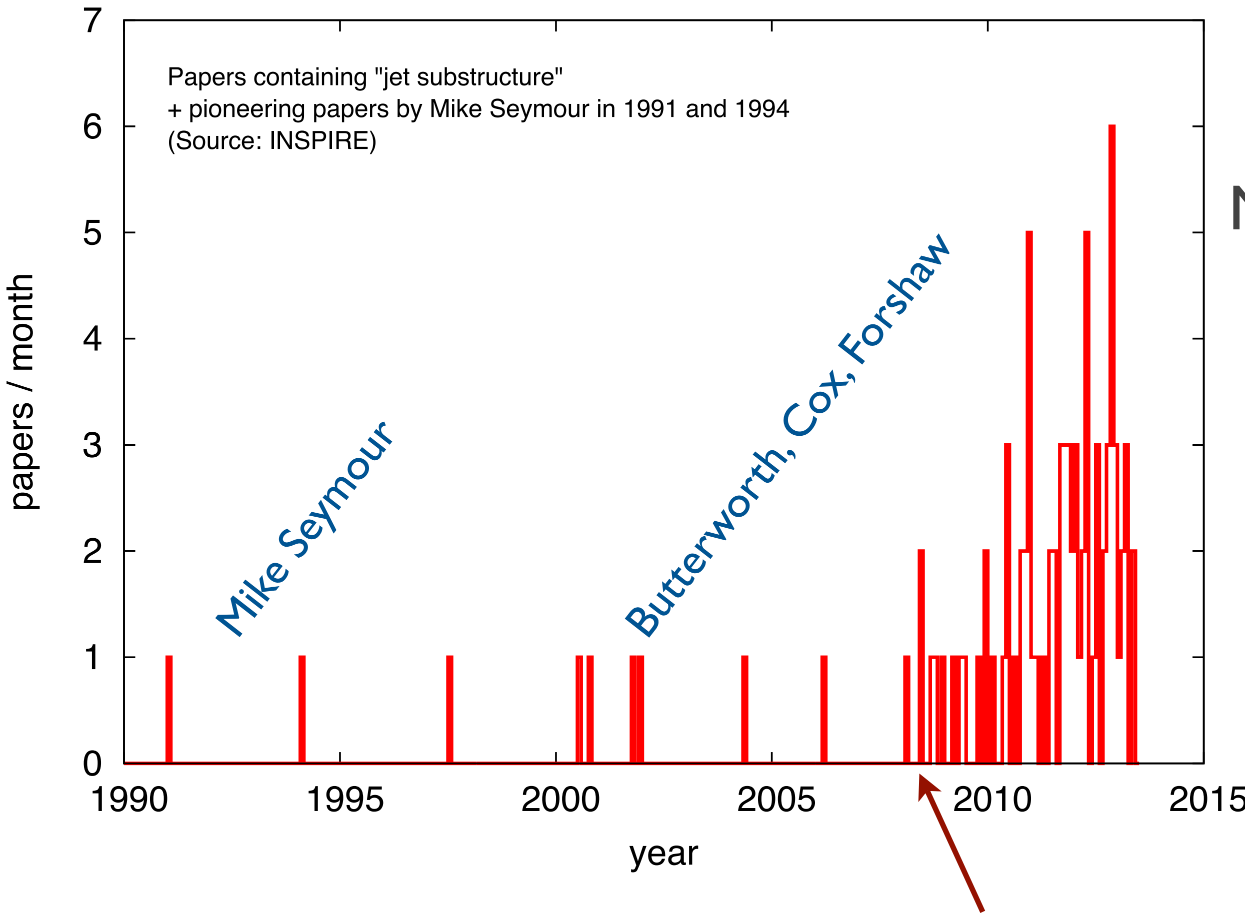
For a given mass, the **background** will have smaller distance d_{ij} than the signal, i.e. it will tend to **cluster earlier** in the k_t algorithm

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

'Jet substructure' papers in INSPIRE

Number of papers containing the words 'jet substructure'



More than 100 papers since 2008

(+ some background noise)

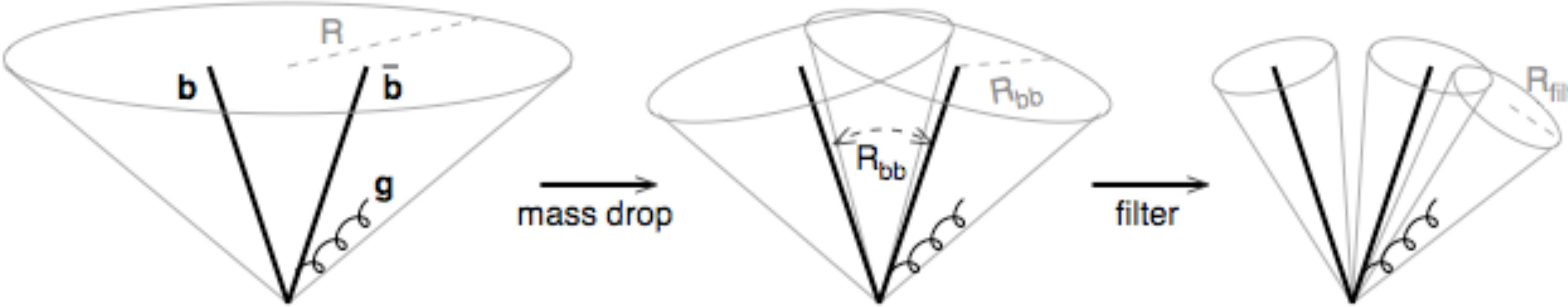
Pioneered by M. Seymour in the early '90s, rebooted by BDRS paper

15. **Jet substructure as a new Higgs search channel at the LHC.**
Jonathan M. Butterworth, Adam R. Davison (University Coll. London), Mathieu Rubin, Gavin P. Salam (Paris, LPTHE).
Published in **Phys.Rev.Lett. 100 (2008) 242001**
e-Print: **arXiv:0802.2470 [hep-ph]**

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

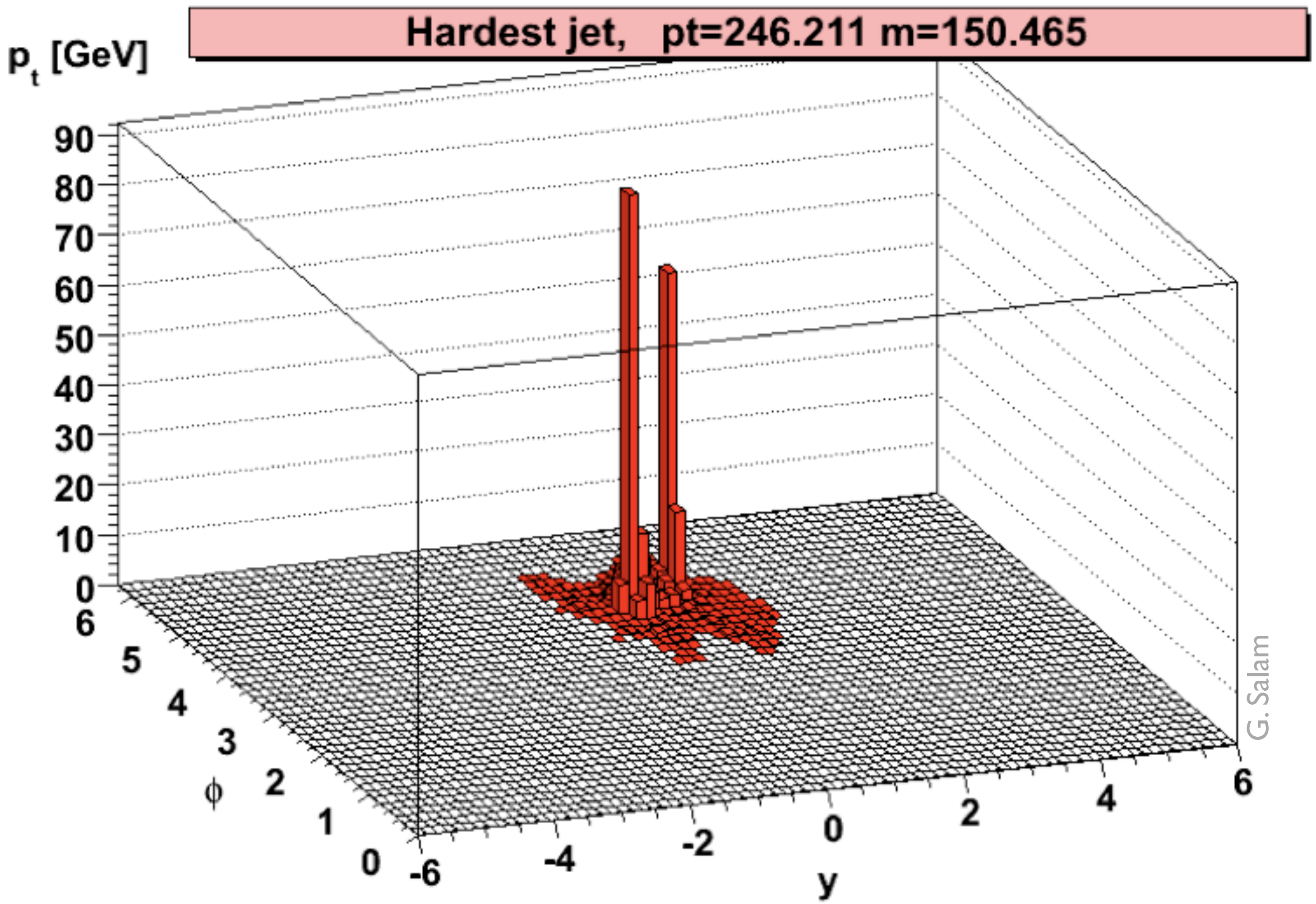
The BDRS tagger/groomer

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**')

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

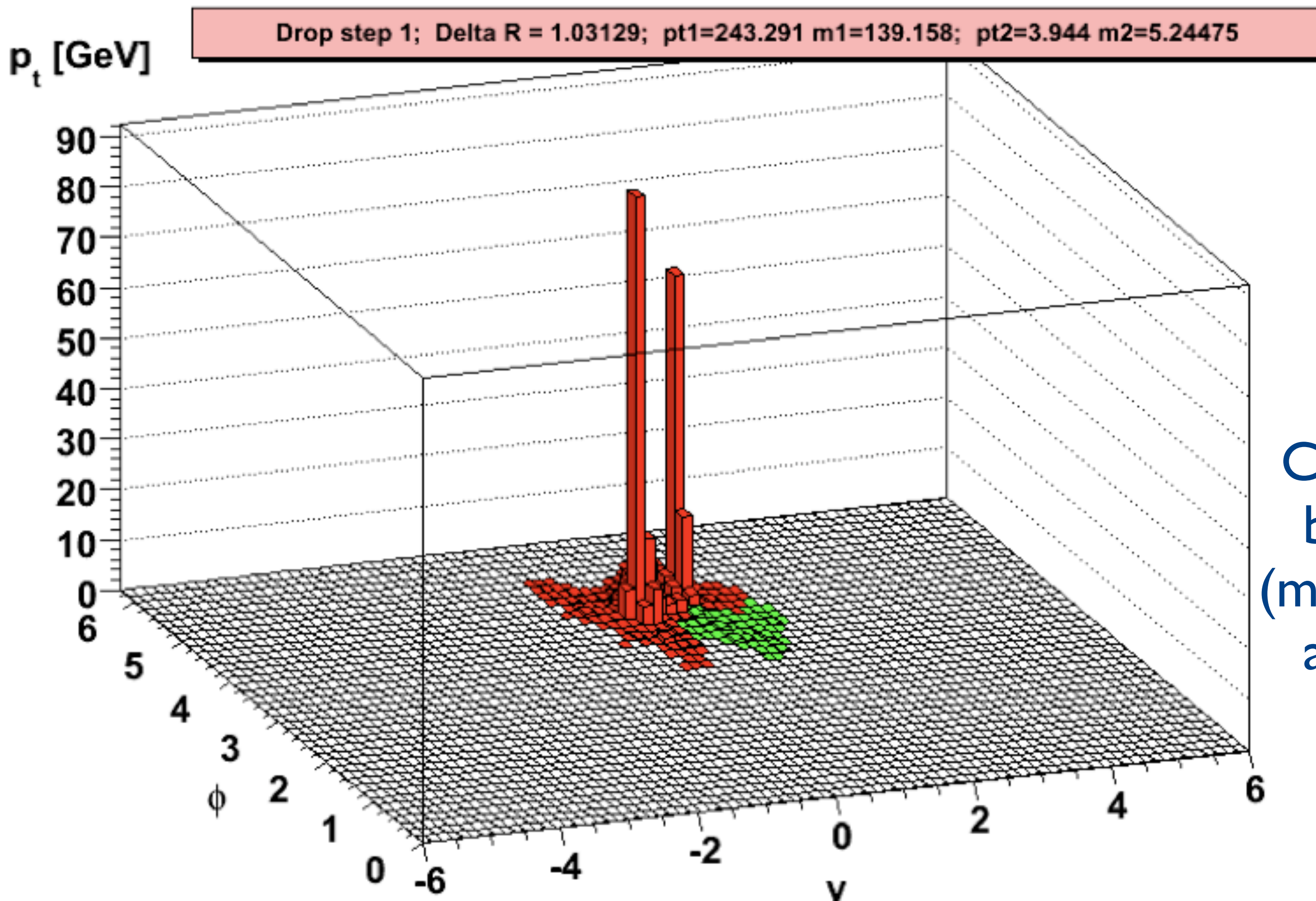


Start with the hardest jet

Use C/A with large R=1.2

m_j = 150 GeV

pp → ZH → ννbb

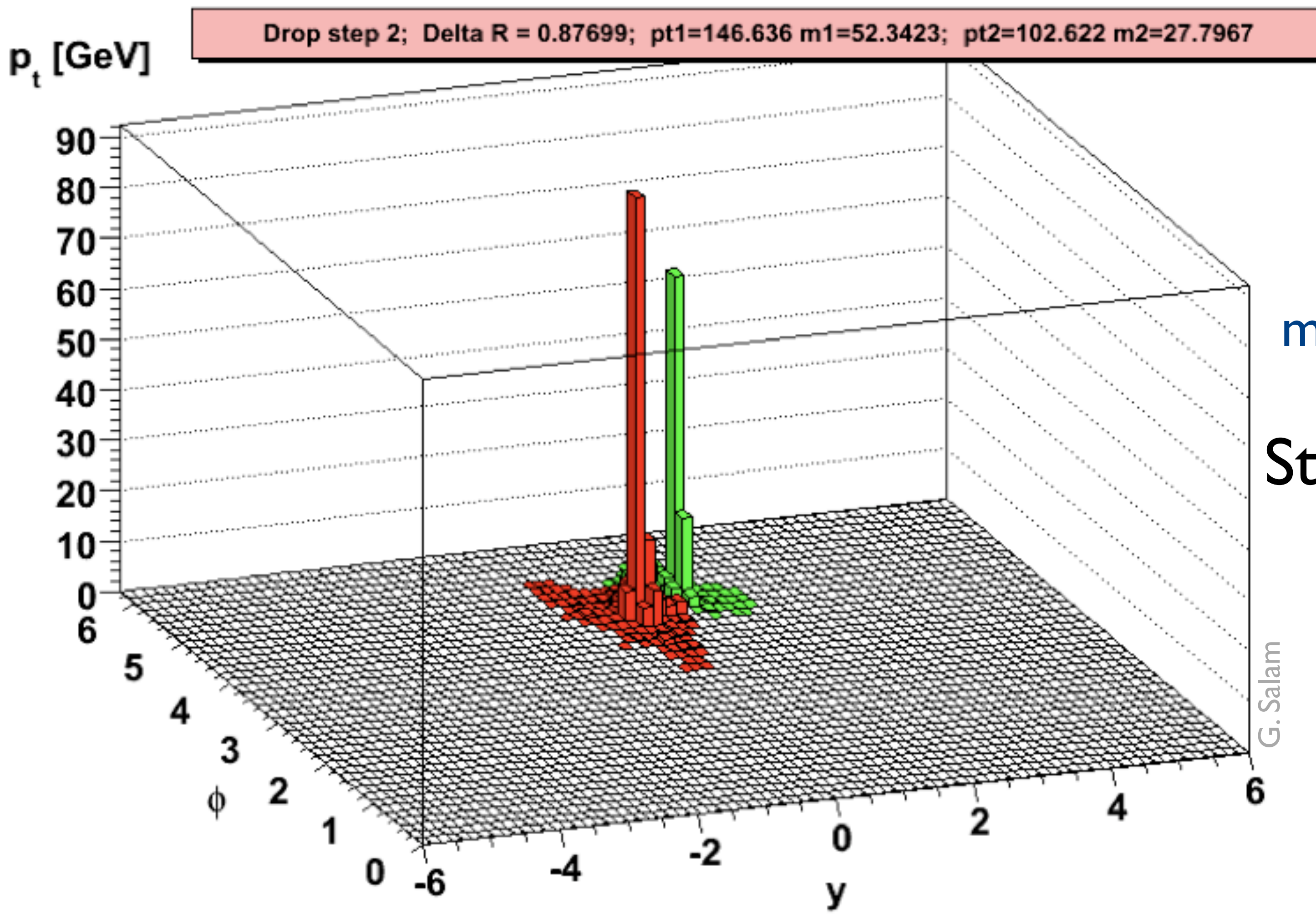


Undo last step of clustering

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

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

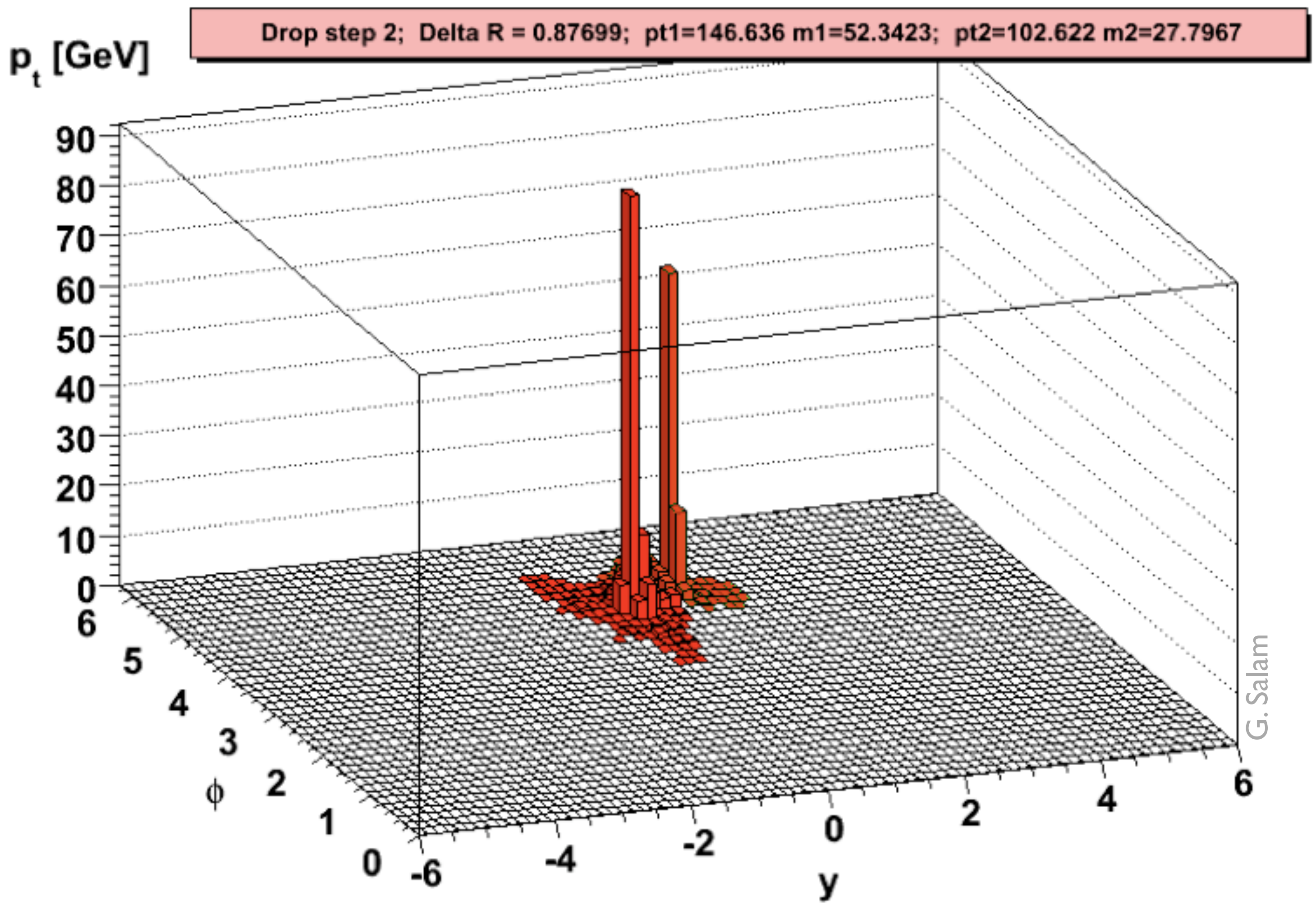


m_1 = 52 GeV, m_2 = 28 GeV

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

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

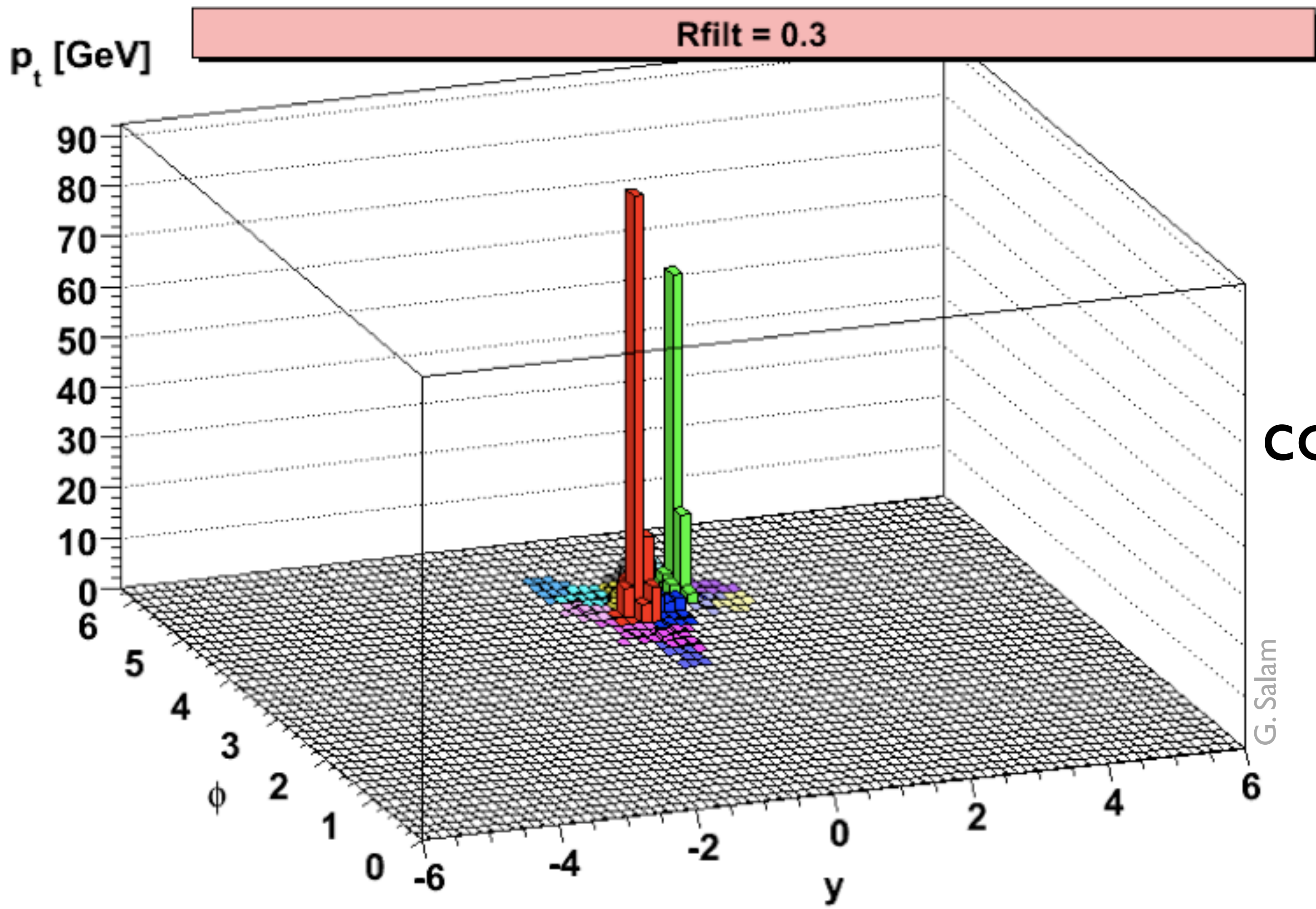
pp → ZH → ννbb



Start with the recombined jet

BDRS: filtering

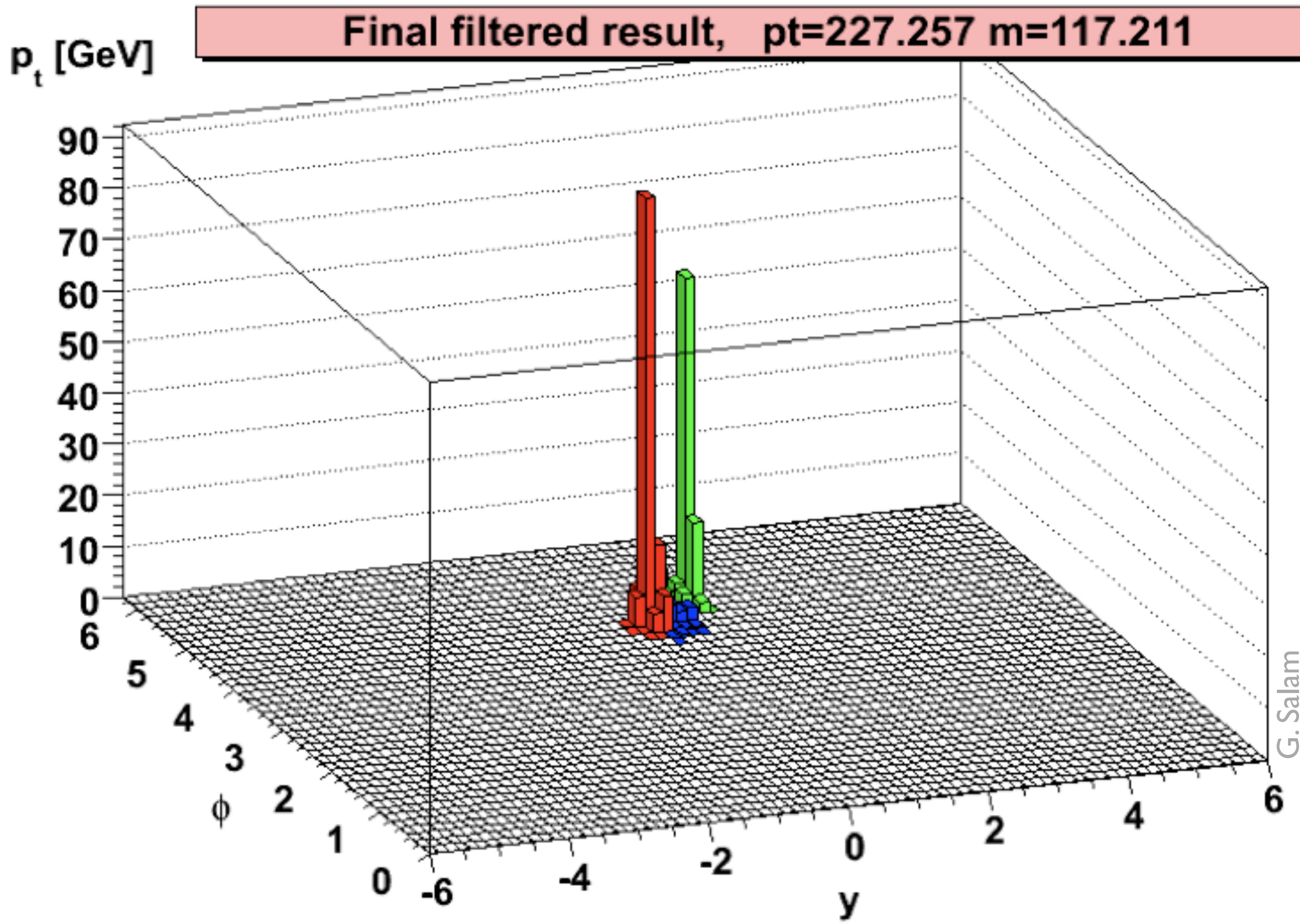
$pp \to ZH \to \nu\nu bb$



Recluster the contituents with R_{filt}

BDRS: filtering

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



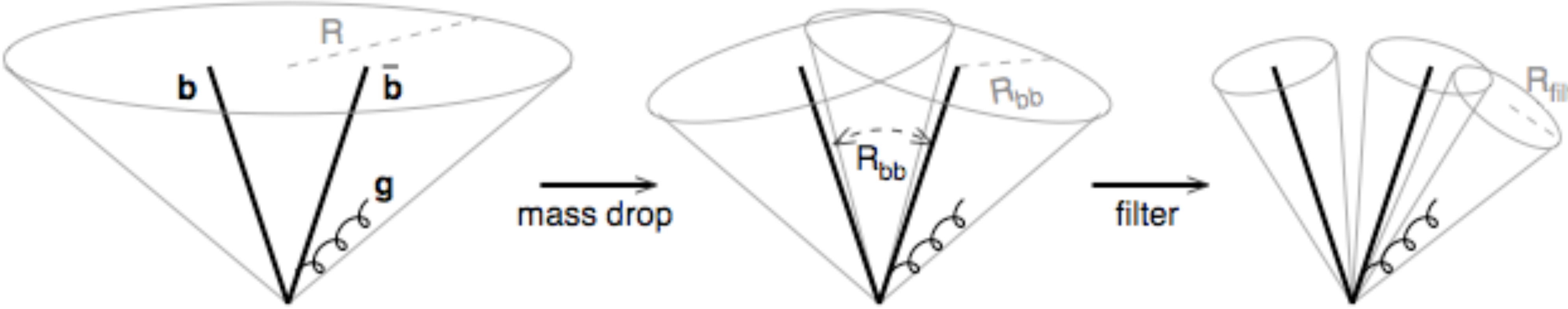
Only keep the n_{filt} hardest jets

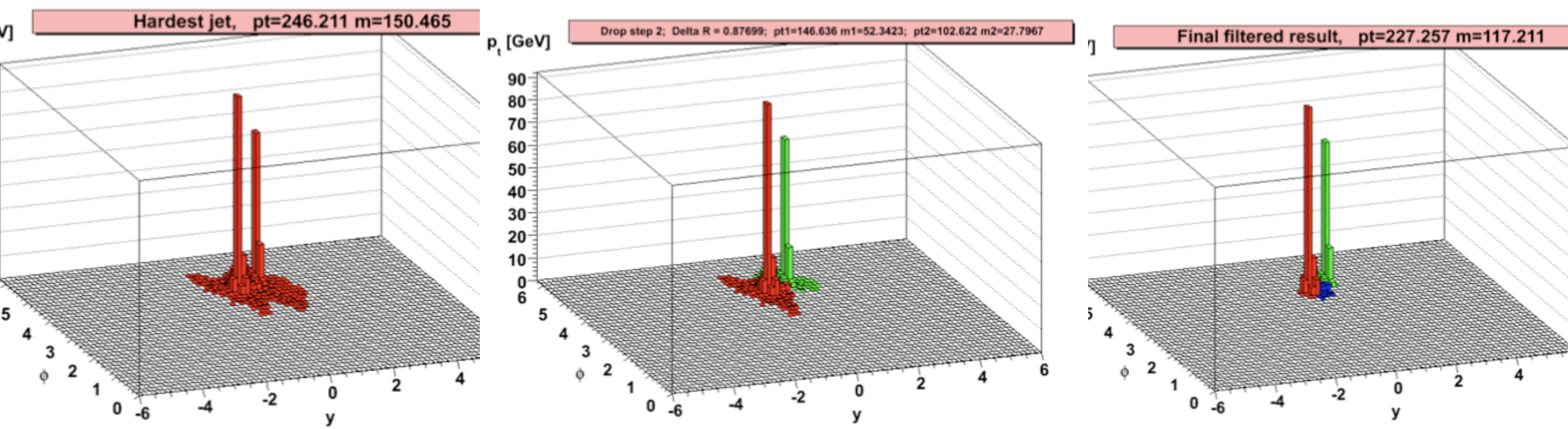
The low-momentum stuff surrounding the hard particles has been removed

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

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

In FastJet

```
#include "fastjet/tools/MassDropTagger.hh"
#include "fastjet/tools/Filter.hh"

JetDefinition jet_def(cambridge_algorithm, 1.2);
ClusterSequence cs(input_particles, jet_def);

// define the tagger and use it
MassDropTagger md_tagger(0.667, 0.09);
PseudoJet tagged = md_tagger(jets[0]);

// define the filter and use it
Filter filter(0.3,SelectorNHardest(3));
Pseudojet higgs = filter(tagged);     // this is the Higgs!!
```

The real analysis is slightly more refined (b-tagging, dynamical filter radius, etc) but the main features are already present here

First taggers/groomers

- **Mass Drop + Filtering** — Butterworth, Davison, Rubin, Salam, 2008

  Decluster with mass drop and asymmetry conditions
  Recluster constituents into subjets at distance scale R_{filt}, retain n_{filt} hardest subjets

- **Jet 'trimming'** — Krohn, Thaler, Wang, 2009

  Recluster constituents into subjets at distance scale R_{trim},
  retain subjets with $p_{t,subjet} > \varepsilon_{trim}\, p_{t,jet}$

- **Jet 'pruning'** — S. Ellis, Vermilion, Walsh, 2009

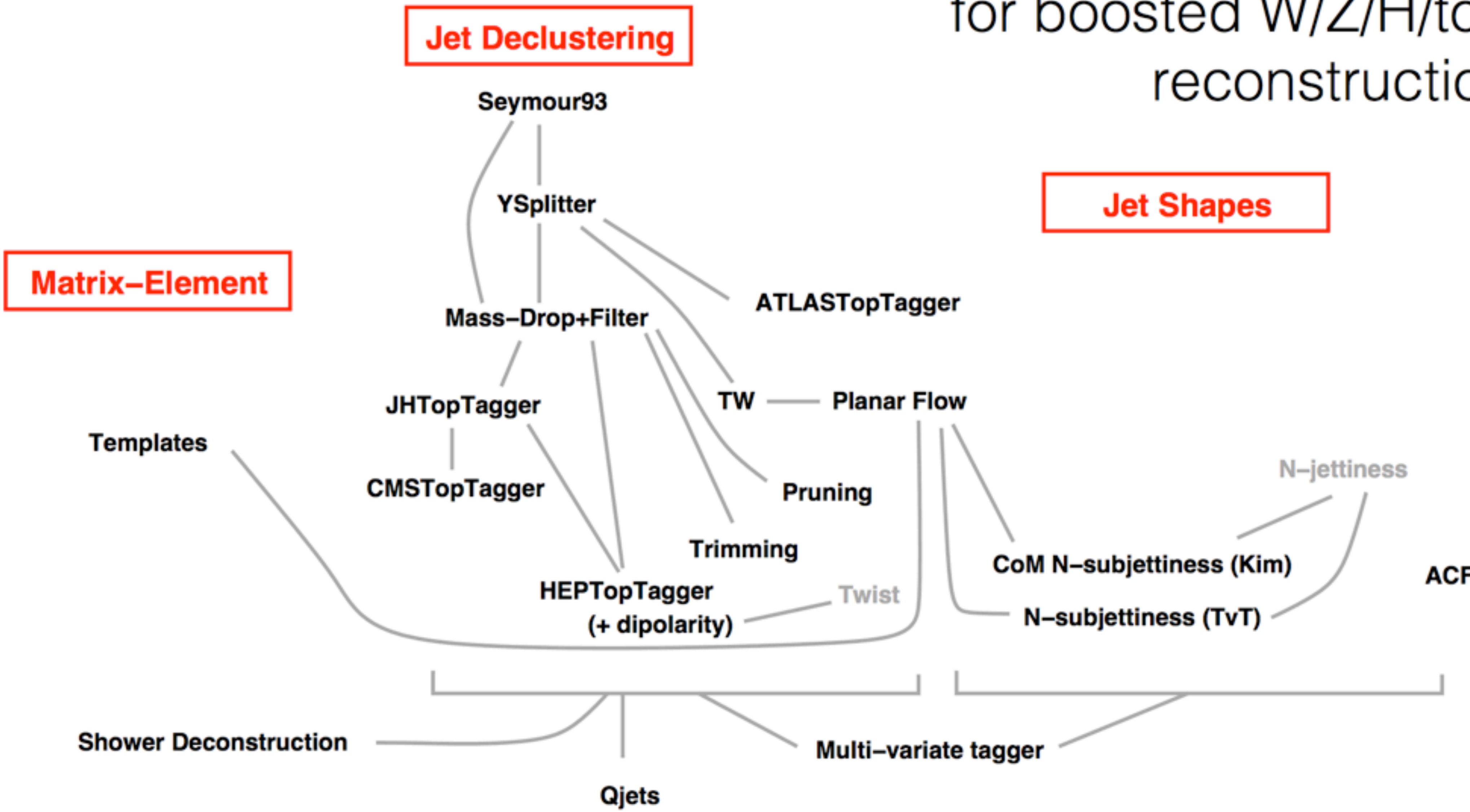
  While building up the jet, discard softer subjets when $\Delta R > R_{prune}$
  and $\min(p_{t1},p_{t2}) < \varepsilon_{prune}\,(p_{t1}+p_{t2})$

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



Slide by G. Salam, now a few years old

Conclusion part I

- A number of different IRC-safe jet algorithms exist
  - They all try to be good proxies for hard partons, but they have different characteristics, especially with respect to soft particles

- Jets from all algorithms inevitably suffer from pileup contamination
  - Techniques exist to subtract it, either at jet-level, or at particle-level

- Both the jet algorithms and many pileup subtraction techniques are packaged aither in FastJet or in fjcontrib contributions
  - **Use of standard algorithms and packages** (either directly or through interfaces) **should be privileged**, as it ensures reproducibility

http://fastjet.fr http://fastjet.hepforge.org/contrib/

Conclusions part 2

The big news of the past few years has been the emergence of jet-based taggers and groomers

- They have proven their worth in ‘Standard Model’ analyses
- They are being implemented in BSM searches

The tutorial will offer you the opportunity to play with clustering and substructure