Physics at the LHC
DESY Hamburg
June 7, 2010

Jet Physics

The path from Clustering to Jetography

Matteo Cacciari
LPTHE - Paris 6,7 and CNRS

Purposes

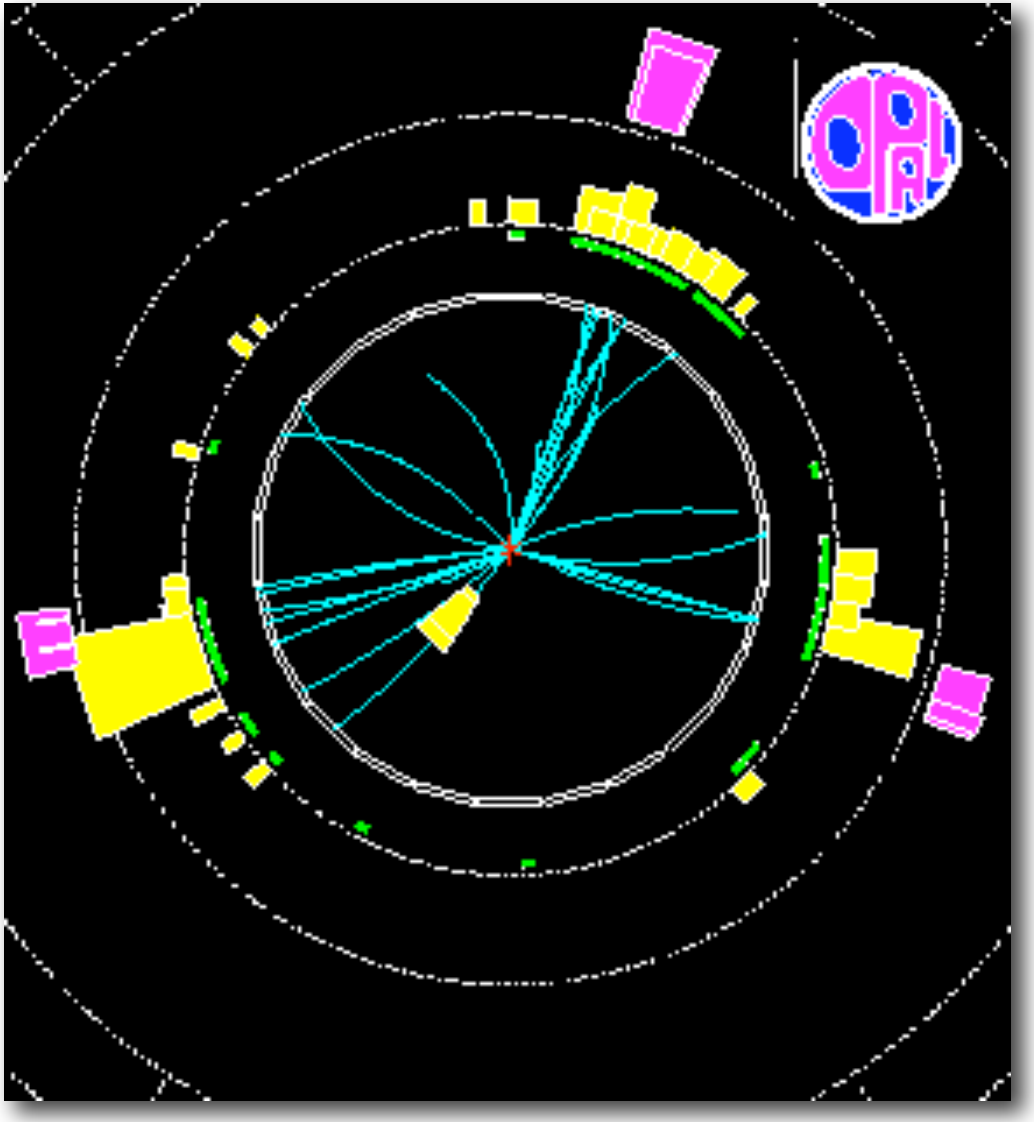
Jets can serve two purposes:

- 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

Outline

- Introductory remarks and review
- New jet algorithms (anti-k_t, SISCone) and FastJet
- New properties/tools: areas, backreaction, background subtraction, taggers, filtering (trimming, pruning)
- Examples: *reconstruction of a mass peak, Higgs search using jet substructure*

Why jets

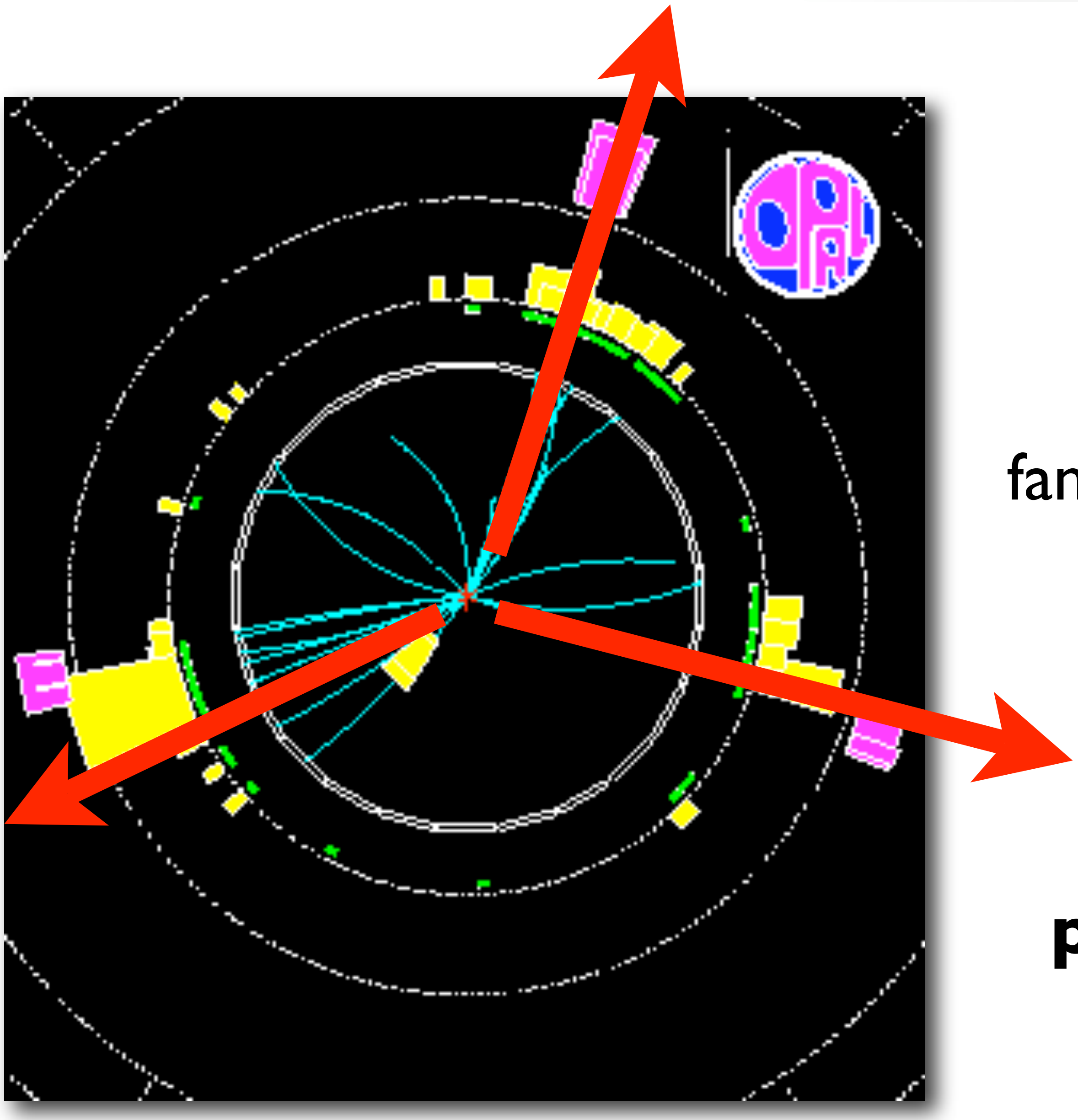


A jet is something that happens in high energy events:

a collimated bunch of hadrons flying roughly in the same direction

Note: hundreds of hadrons contain **a lot** of information. More than we can hope to make use of

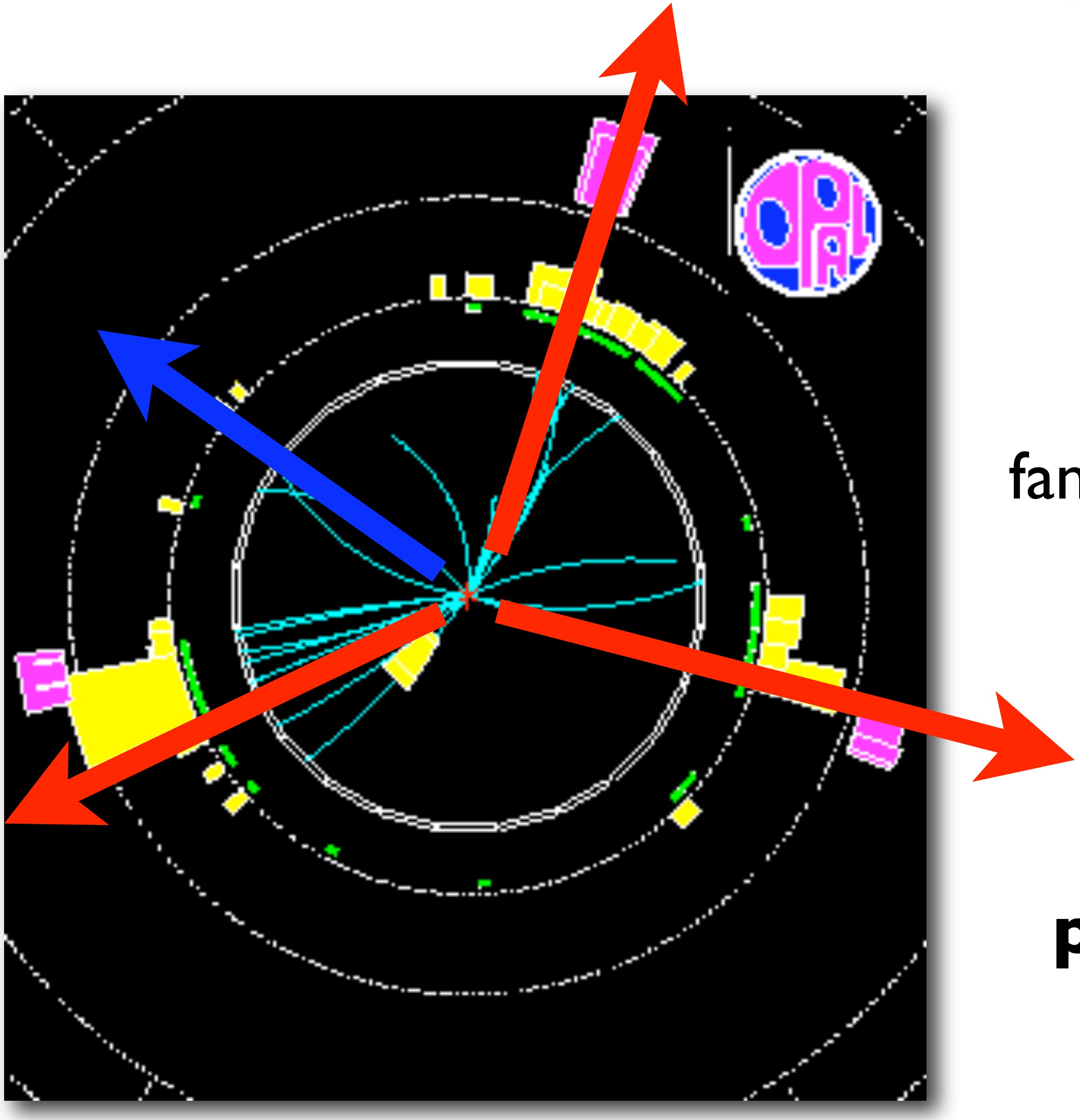
Why jets



Often you don't need a fancy algorithm to 'see' the jets

But you do to give them a **precise** and **quantitative** meaning

Why jets

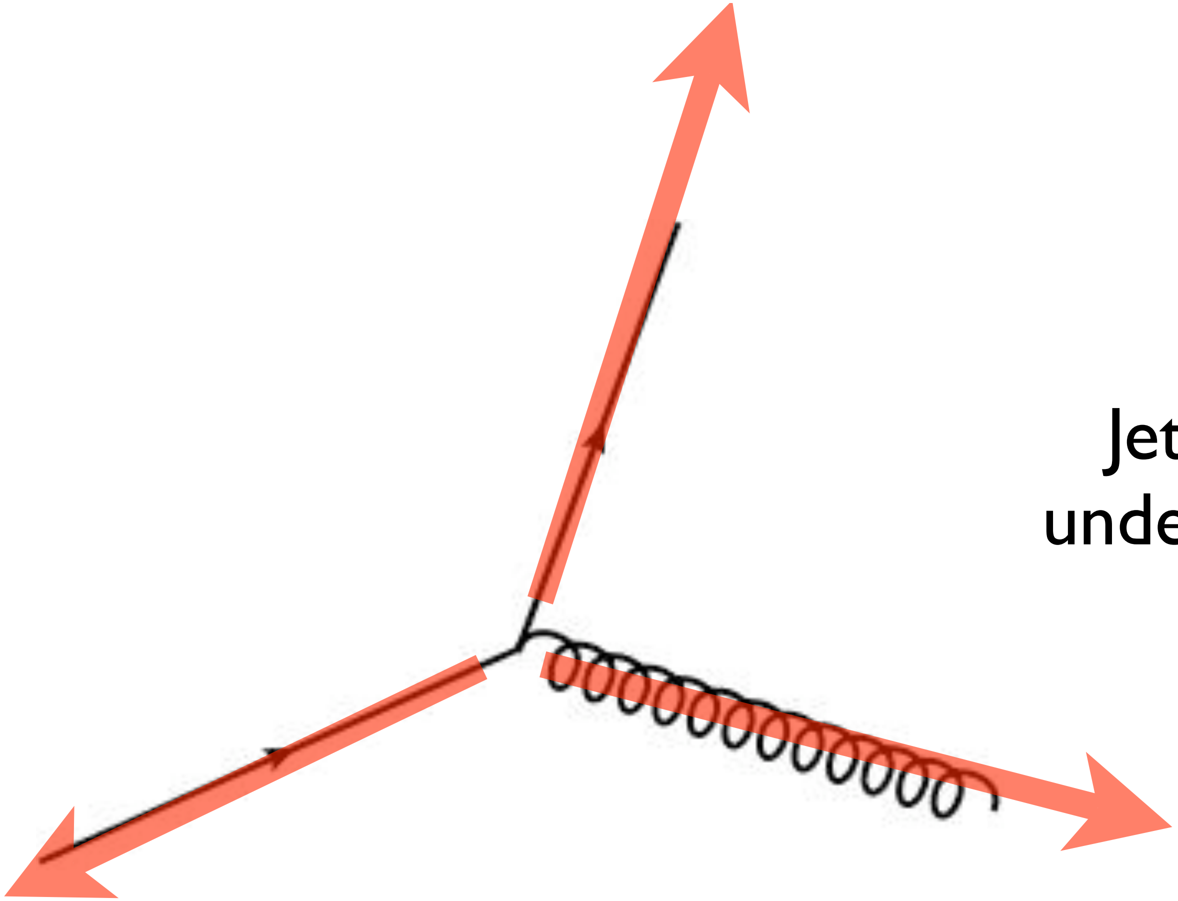


Often you don't need a fancy algorithm to 'see' the jets

But you do to give them a **precise** and **quantitative** meaning

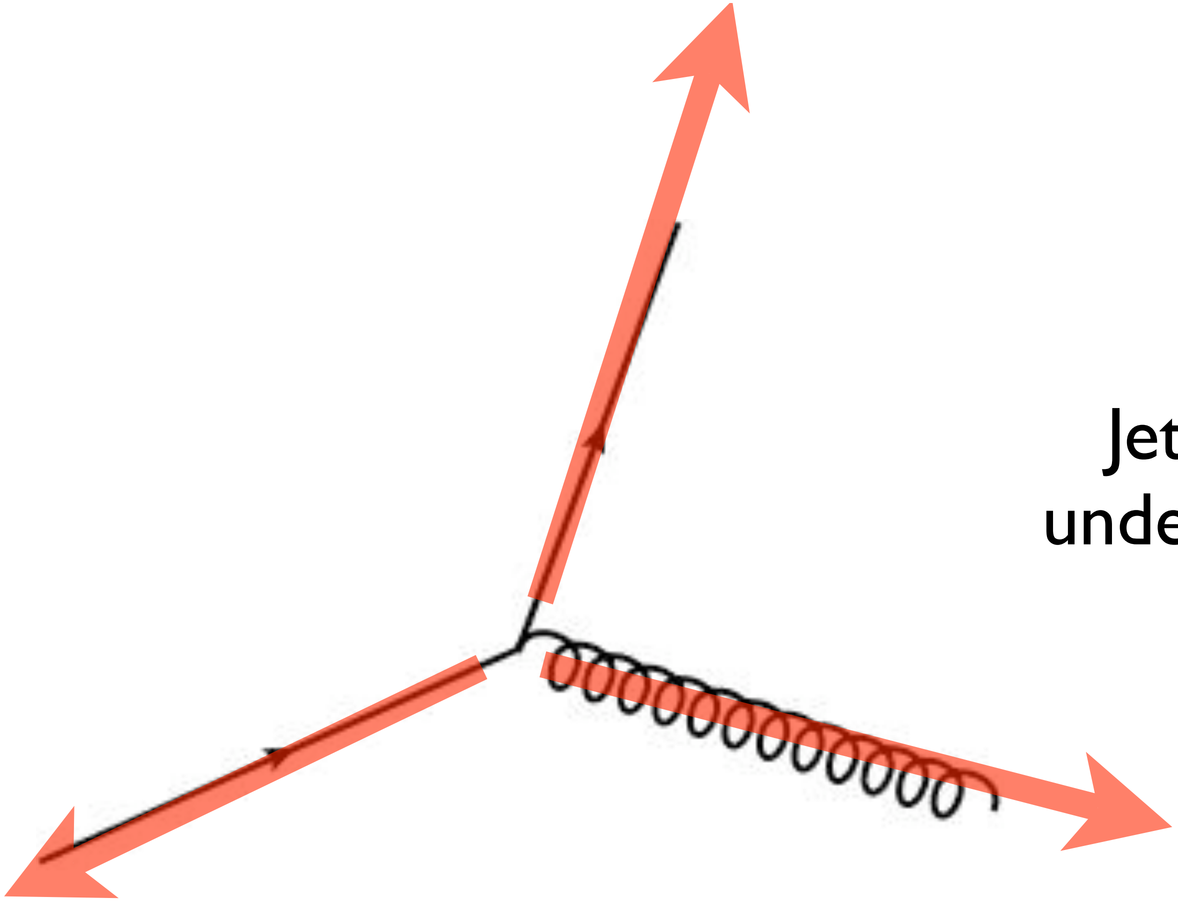
For instance: are there 3 or 4 jets in this event?

Why jets



Jets are usually related to an underlying perturbative dynamics (i.e. quarks and gluons)

Why jets



Jets are usually related to an underlying perturbative dynamics (i.e. quarks and gluons)

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

Jet 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

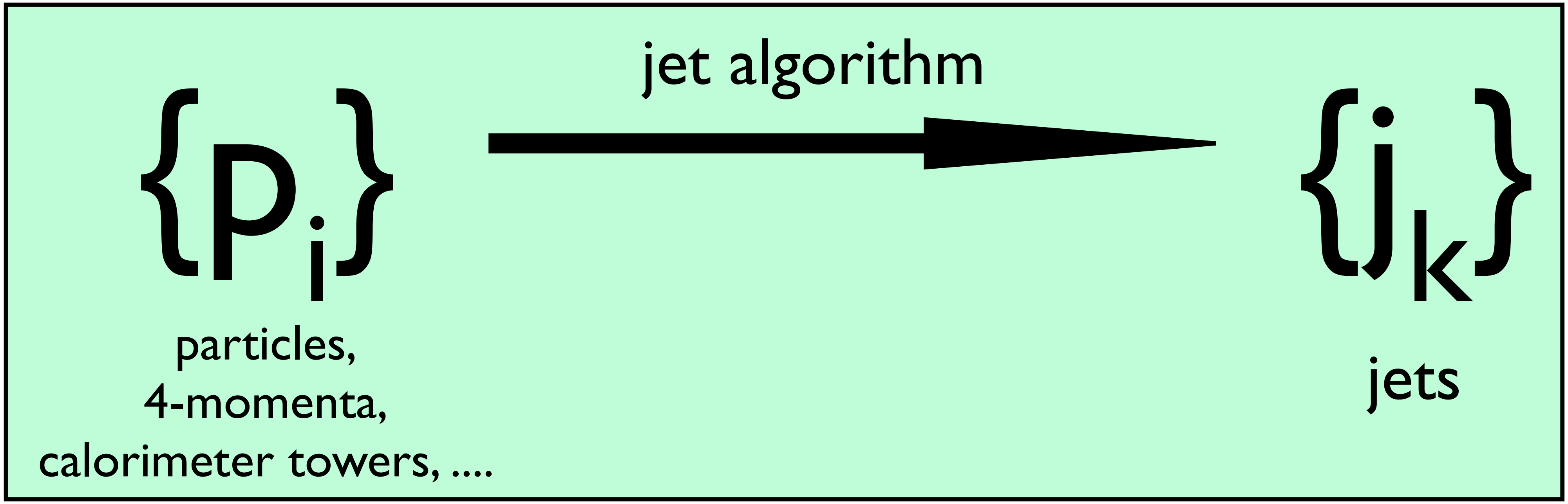
Jet definition

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

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

jets

Jet definition



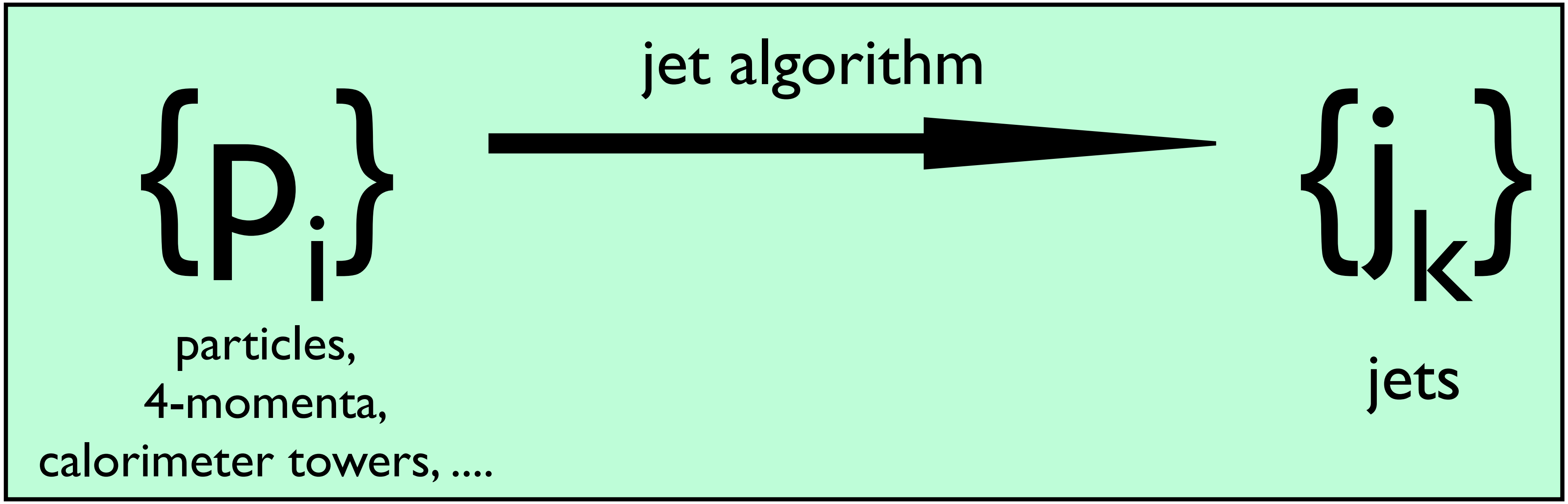
$\{p_i\} \xrightarrow{\text{jet algorithm}} \{j_k\}$

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

jets

+ parameters (usually at least the radius R)

Jet definition



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

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

jets

+ parameters (usually at least the radius R)

+ recombination scheme

Jet definition

Les Houches 2007 proceedings, arXiv:0803.0678

jet definition

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

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

jets

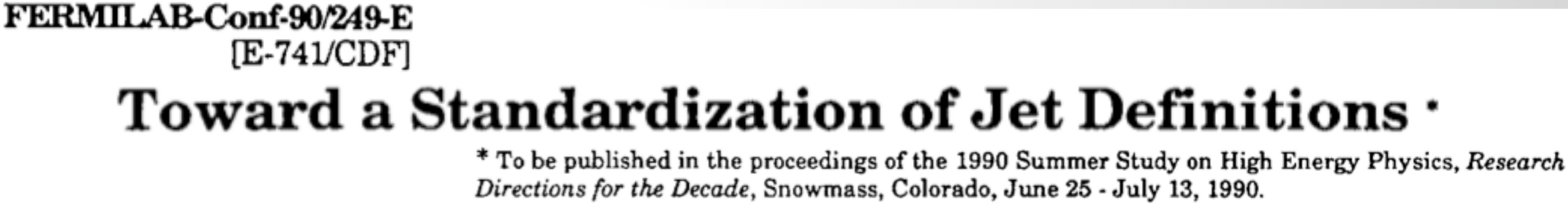
+ parameters (usually at least the radius R)

+ recombination scheme

Reminder: running a jet definition gives a well defined physical observable, which we can measure and, hopefully, calculate

Observables

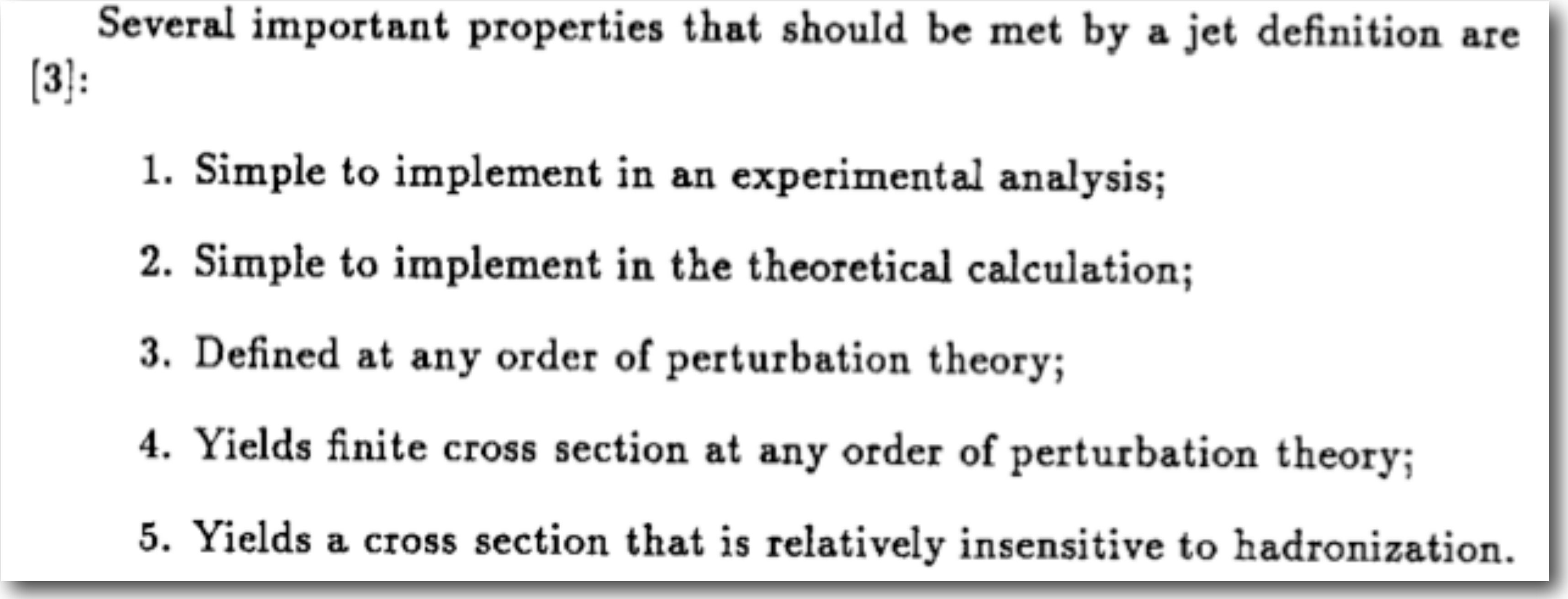
Snowmass



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

Toward a Standardization of Jet Definitions *

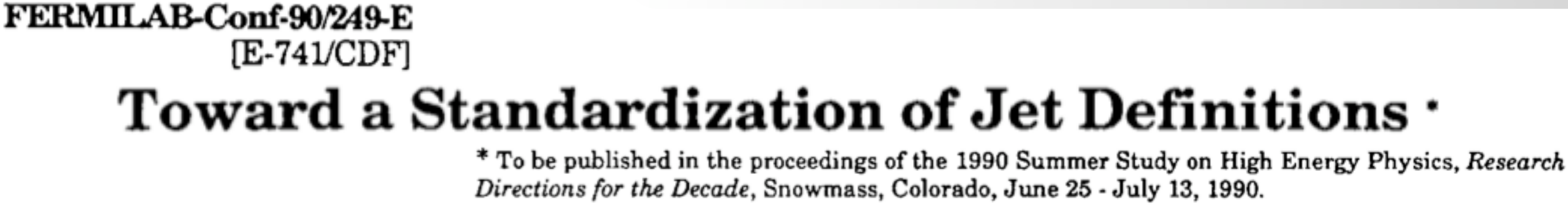
* To be published in the proceedings of the 1990 Summer Study on High Energy Physics, *Research Directions for the Decade*, Snowmass, Colorado, June 25 - July 13, 1990.



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.

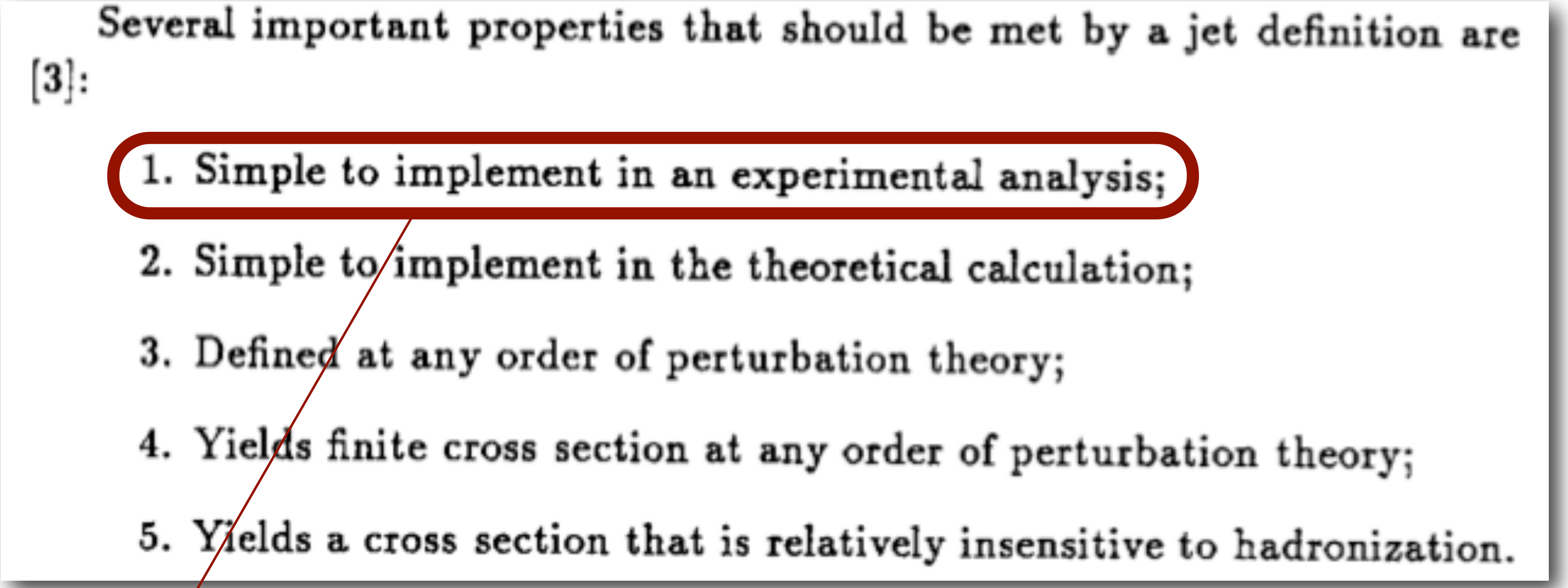
Snowmass



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.

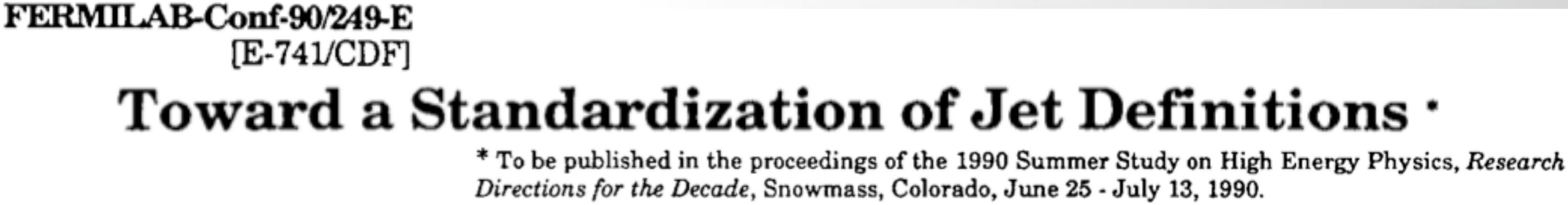


Several important properties that should be met by a jet definition are [3]:

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

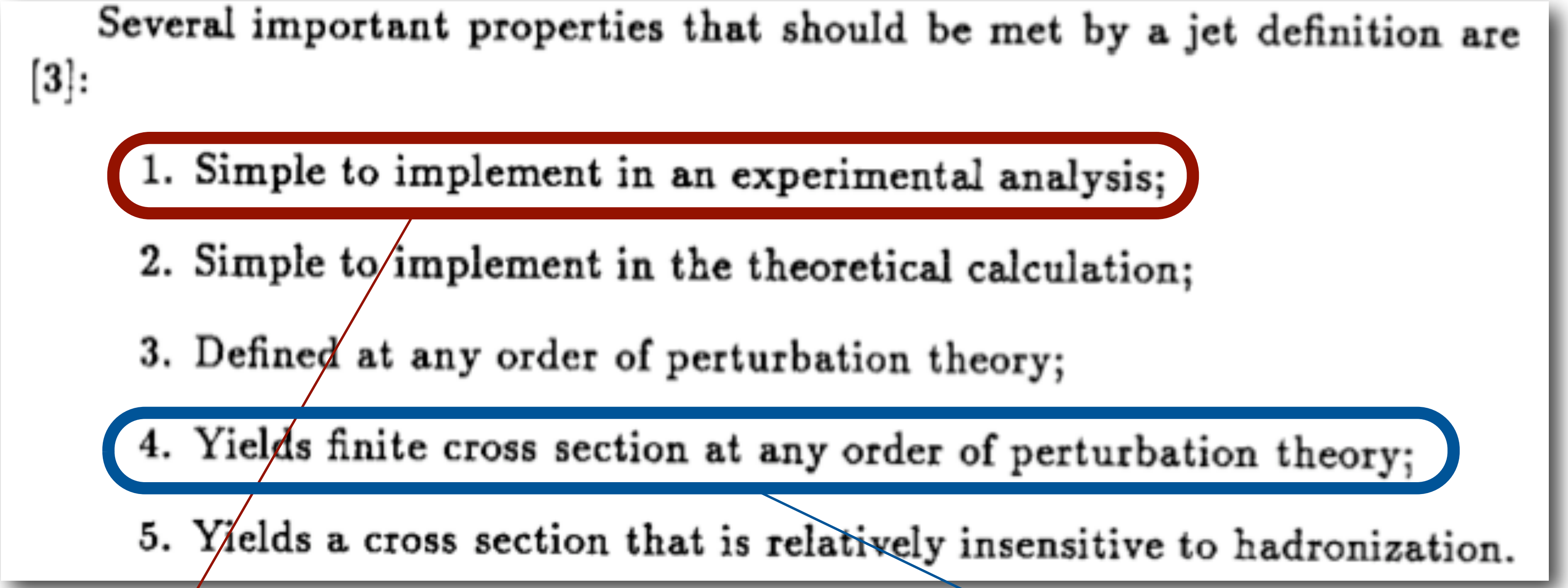
Snowmass



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]:

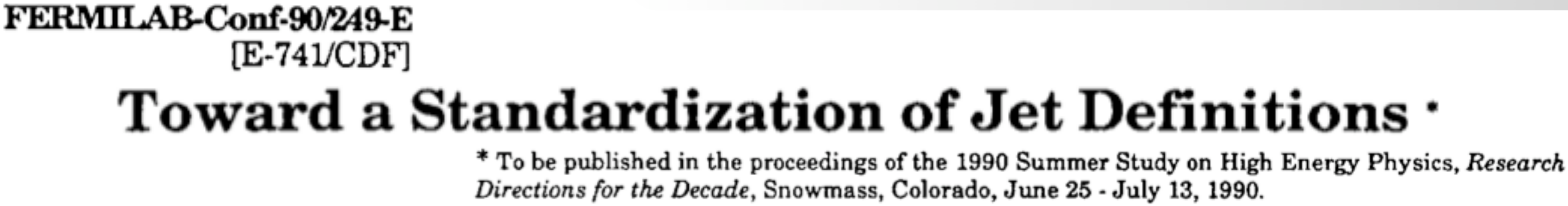
1. Simple to implement in an experimental analysis;
2. Simple to implement in the theoretical calculation;
3. Defined at any order of perturbation theory;
4. Yields finite cross section at any order of perturbation theory;
5. Yields a cross section that is relatively insensitive to hadronization.

Speed

Infrared and collinear safety

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

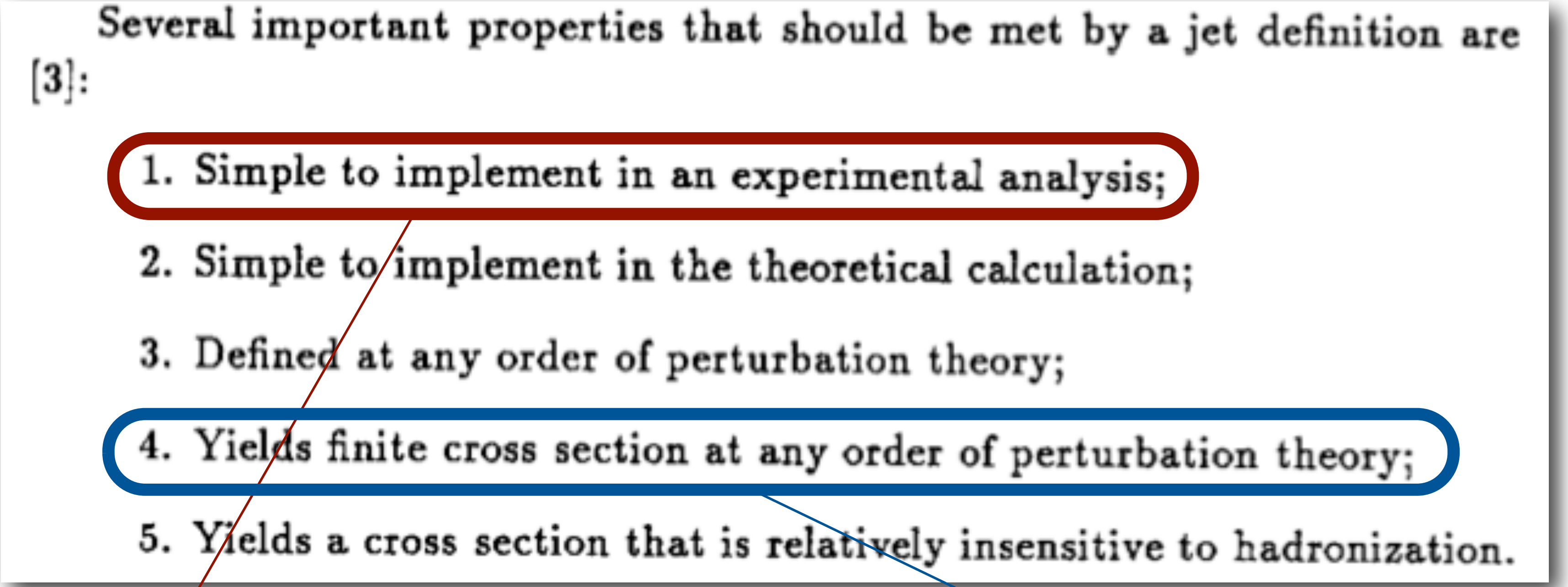
Snowmass



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]

Snowmass set standards, but didn't provide solutions

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

Work because of mapping closeness ⇔ QCD divergence

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

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

Work 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)

Work because QCD only modifies energy flow on small scales

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

Cone algorithms

Finding all stable cones (and hence produce an infrared and collinear (IRC) safe cone algorithm) would naively take $N2^N$ operations

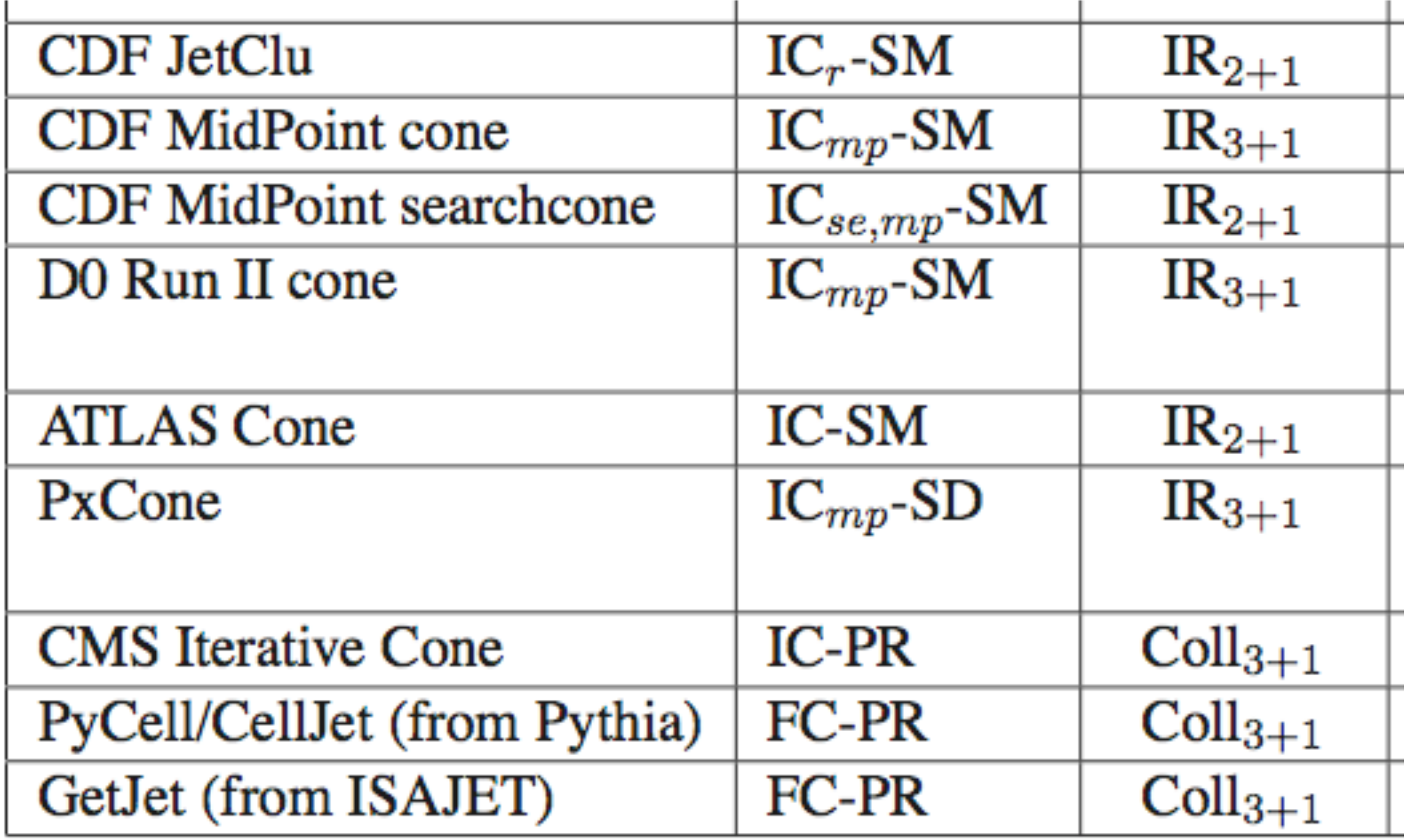
This is roughly the age of the universe for just 100 particles

Too slow.
Resort to approximate methods

A long list of cones (all eventually unsafe)

Les Houches 2007 proceedings, arXiv:0803.0678

‘First-generation’ algorithms



CDF JetClu	IC_r-SM	IR_{2+1}
CDF MidPoint cone	IC_{mp}-SM	IR_{3+1}
CDF MidPoint searchcone	$IC_{se,mp}$-SM	IR_{2+1}
D0 Run II cone	IC_{mp}-SM	IR_{3+1}
ATLAS Cone	IC-SM	IR_{2+1}
PxCone	IC_{mp}-SD	IR_{3+1}
CMS Iterative Cone	IC-PR	$Coll_{3+1}$
PyCell/CellJet (from Pythia)	FC-PR	$Coll_{3+1}$
GetJet (from ISAJET)	FC-PR	$Coll_{3+1}$

IC = Iterative Cone
SM = Split-Merge
SD = Split-Drop
FC = Fixed Cone
PR = Progressive Removal

type of algorithm

safety issue

IR_{n+1} : unsafe when a soft particle is added to n hard particles in a common neighbourhood

$Coll_{n+1}$: unsafe when one of n hard particles in a common neighbourhood is split collinearly

Recombination algorithms: k_t

Longitudinally invariant k_t:

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

- Calculate the distances between the particles: $d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta y^2 + \Delta\phi^2}{R^2}$
- Calculate the beam distances: $d_{iB} = k_{ti}^2$
- Combine particles with smallest distance or, if d_{iB} is smallest, call it a jet
- Find again smallest distance and repeat procedure until no particles are left

This is infrared and collinear safe, but finding all the distances is an N^2 operation, to be repeated N times

⇒ naively, the k_t jet algorithm scales like N^3

Faster than the cone, but still too slow:
about 60 seconds for 4000 particles

FastJet and SISCone

Both the N^3/speed problem of k_t and the $N2^N$/speed/IRC safety of the cone were solved by shifting the problem from combinatorics to geometry

- k_t was made fast by reducing the problem to near-neighbour searches, and using Voronoi diagrams to reduce complexity to $N \ln N$

(MC, Salam, hep-ph/0512210)

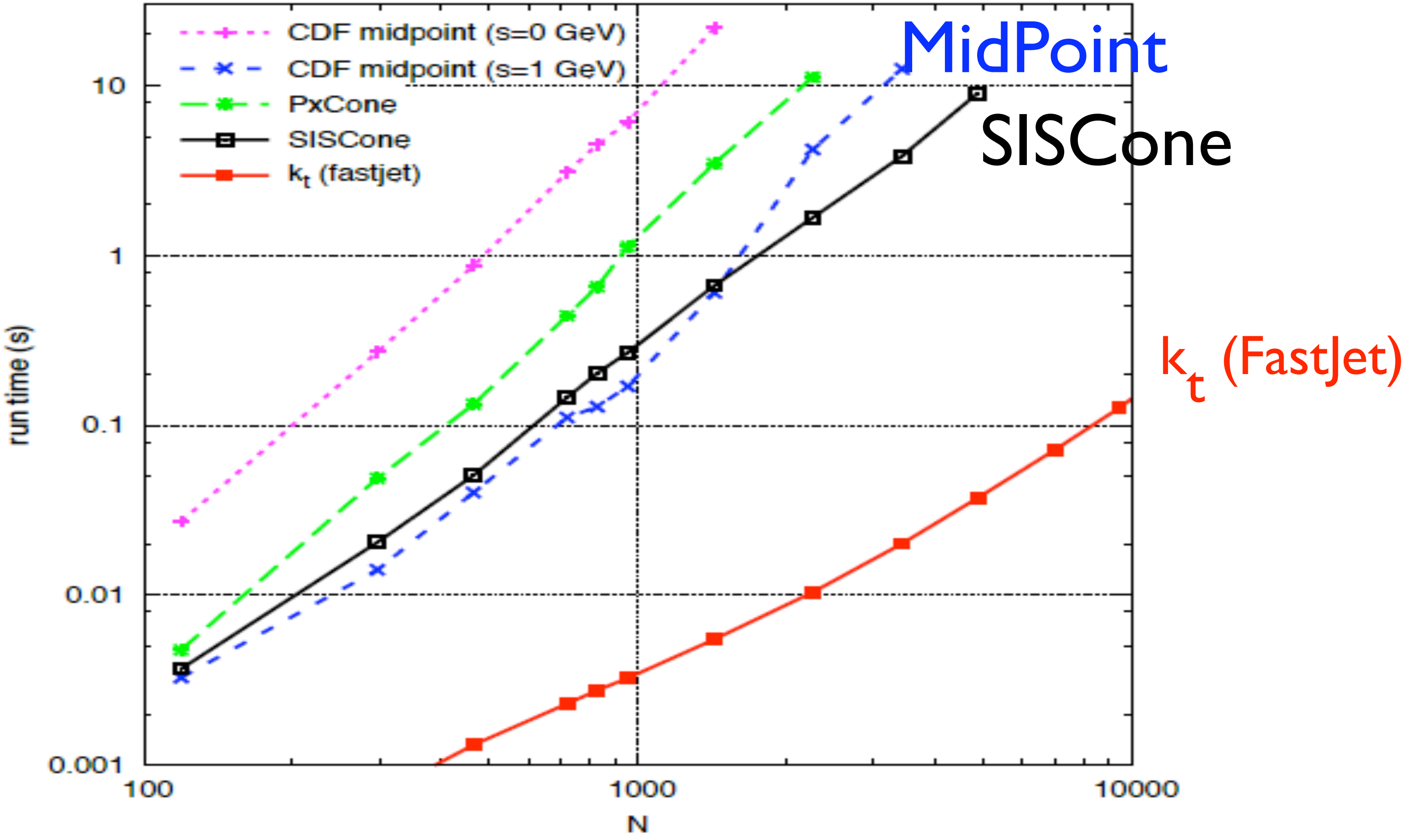
- Cone was made fast (and IRC safe) by inventing circular enclosures to find stable cones and reduce complexity to $N^2 \ln N$

(Salam, Soyez, arXiv: 0704.0292)

Both implementations (and a lot more) available via FastJet

www.fastjet.fr

Timing performances



- SISCone as fast as MidPoint → no penalty for infrared safety
- Large-N region (HI collisions) now doable

Beyond k_t

One can generalise the k_t distance measure:

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

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

Beyond k_t

One can generalise the k_t distance measure:

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

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

Beyond k_t

One can generalise the k_t distance measure:

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

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 with 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

The IRC safe algorithms

k_t	SR $d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2$ hierarchical in rel p_t	Catani et al '91 Ellis, Soper '93	$N\ln N$
Cambridge/ Aachen	SR $d_{ij} = \Delta R_{ij}^2/R^2$ hierarchical in angle	Dokshitzer et al '97 Wengler, Wobish '98	$N\ln N$
anti-k_t	SR $d_{ij} = \min(k_{ti}^{-2}, k_{tj}^{-2})\Delta R_{ij}^2/R^2$ 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	$N^2\ln N$

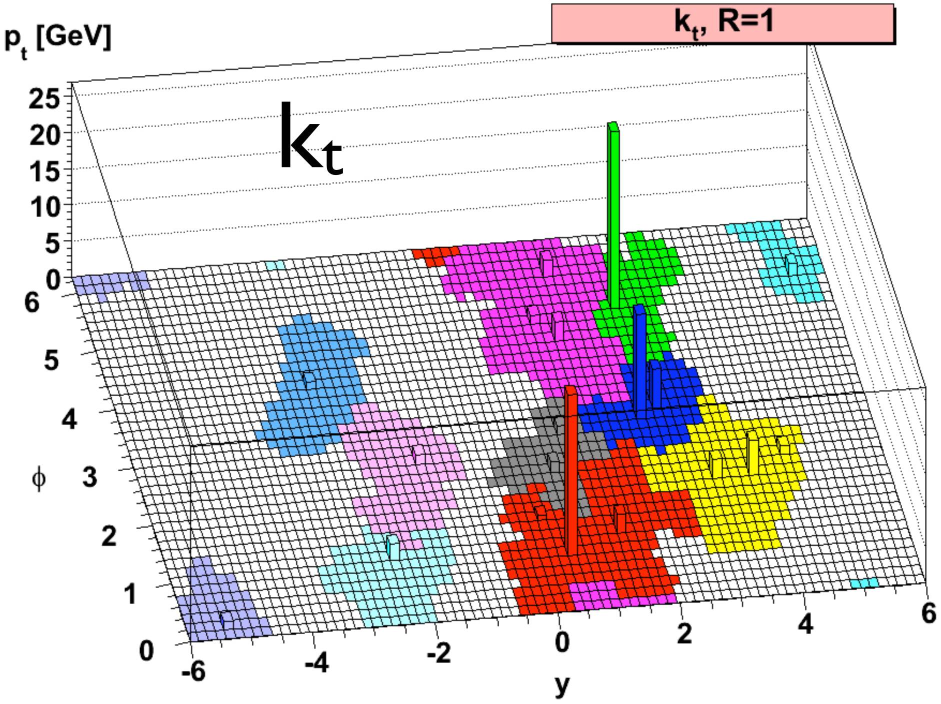
The IRC safe algorithms

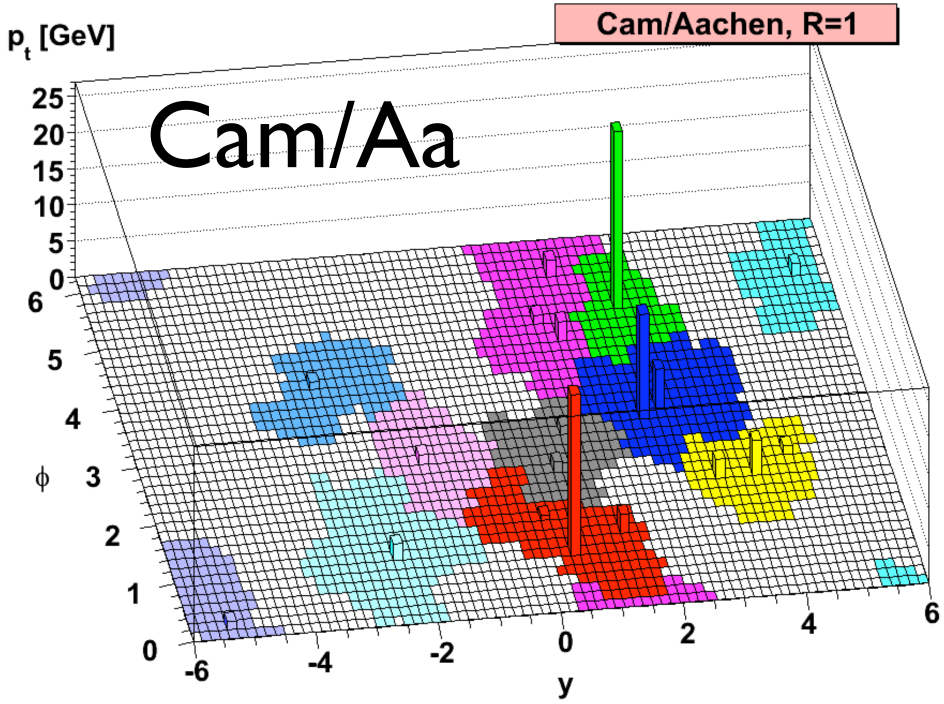
k_t	SR $d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2$ hierarchical in rel p_t	Catani et al '91 Ellis, Soper '93	$N\ln N$
Cambridge/ Aachen	SR $d_{ij} = \Delta R_{ij}^2/R^2$ hierarchical in angle	Dokshitzer et al '97 Wengler, Wobish '98	$N\ln N$
anti-k_t	SR $d_{ij} = \min(k_{ti}^{-2}, k_{tj}^{-2})\Delta R_{ij}^2/R^2$ 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	$N^2\ln N$

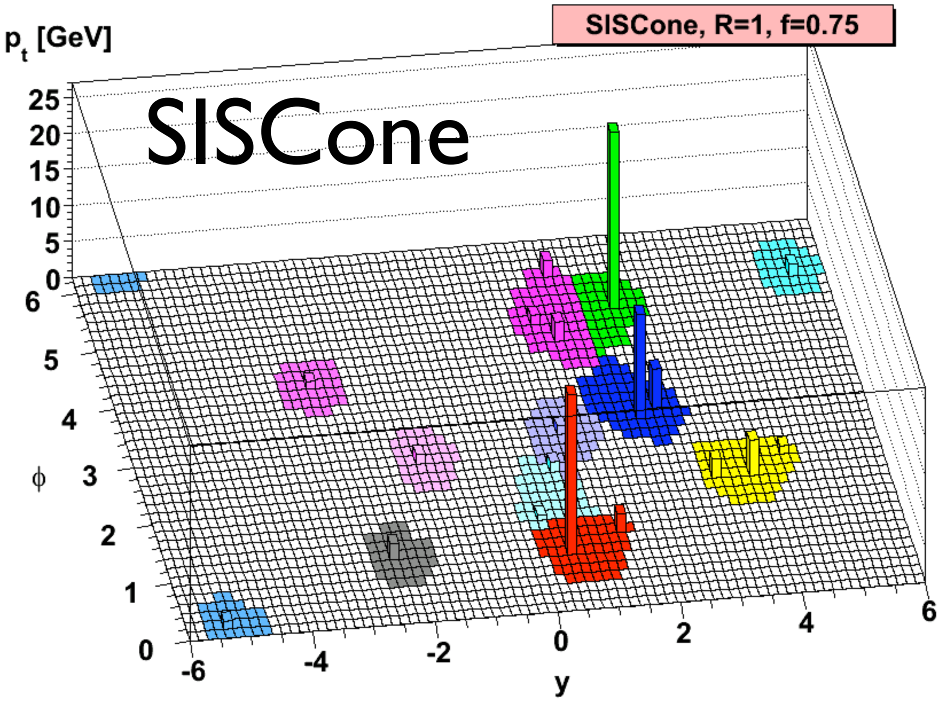
We call these algs 'second-generation' ones

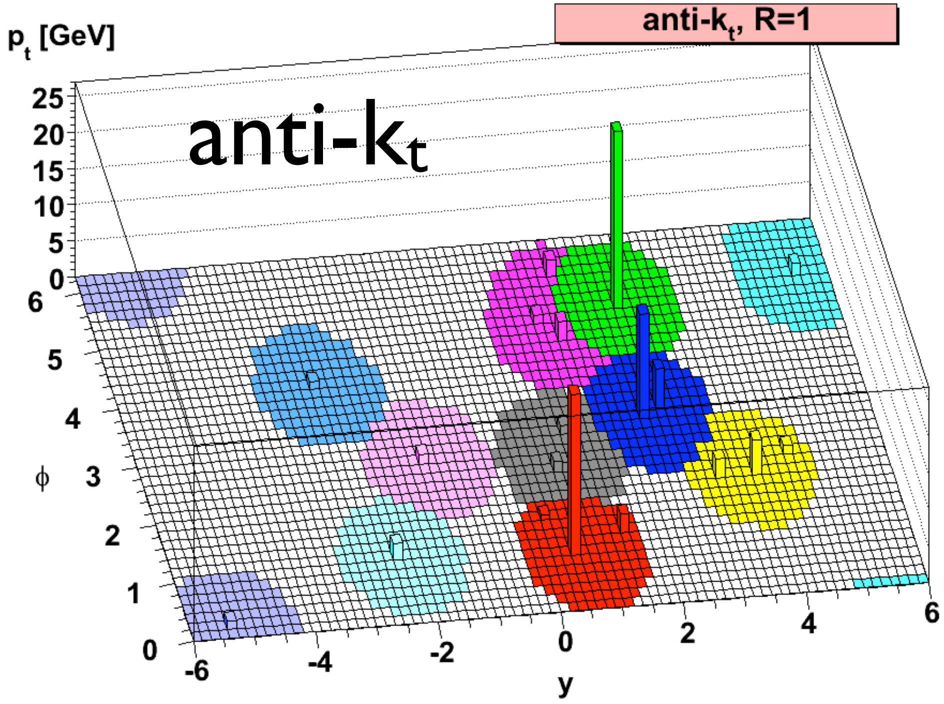
All are available in FastJet, http://fastjet.fr

(As well as many IRC unsafe ones)


k$_t$, R=1
k$_t$
p$_t$ [GeV]
25
20
15
10
5
0
ϕ
6
5
4
3
2
1
0
-6
-4
-2
0
2
4
6
y


Cam/Aachen, R=1
Cam/Aa
p$_t$ [GeV]
25
20
15
10
5
0
ϕ
6
5
4
3
2
1
0
-6
-4
-2
0
2
4
6
y


SISCone, R=1, f=0.75
SISCone
p$_t$ [GeV]
25
20
15
10
5
0
ϕ
6
5
4
3
2
1
0
-6
-4
-2
0
2
4
6
y


anti-k$_t$, R=1
anti-k$_t$
p$_t$ [GeV]
25
20
15
10
5
0
ϕ
6
5
4
3
2
1
0
-6
-4
-2
0
2
4
6
y

Jets

CMS and ATLAS have settled on one of the new algorithms as their default

Amazingly, they will be using the same one, **anti-k_t**

Jets

CMS and ATLAS have settled on one of the new algorithms as their default

Amazingly, they will be using the same one, **anti-k_t**

However, and perhaps unsurprisingly,
CMS will use **R = 0.5** and **0.7**, and ATLAS **0.4** and **0.6**!

While the use of an IRC safe algorithm is welcome, it's fairly unfortunate that there isn't some overlap for immediate comparisons
(I'm told discussion are in progress to fix this)

Tools

Remove soft contamination from a hard jet

Tag heavy objects originating the jet

Eventually leading to 'third-generation' jet algorithms

Jet substructure as tagger

Studying the ***jet substructure***

(i.e. the subjets obtained by undoing the clustering of a sequential recombination algorithm)

can lead to **identification capabilities** of specific objects (as opposed to 'standard' QCD background)

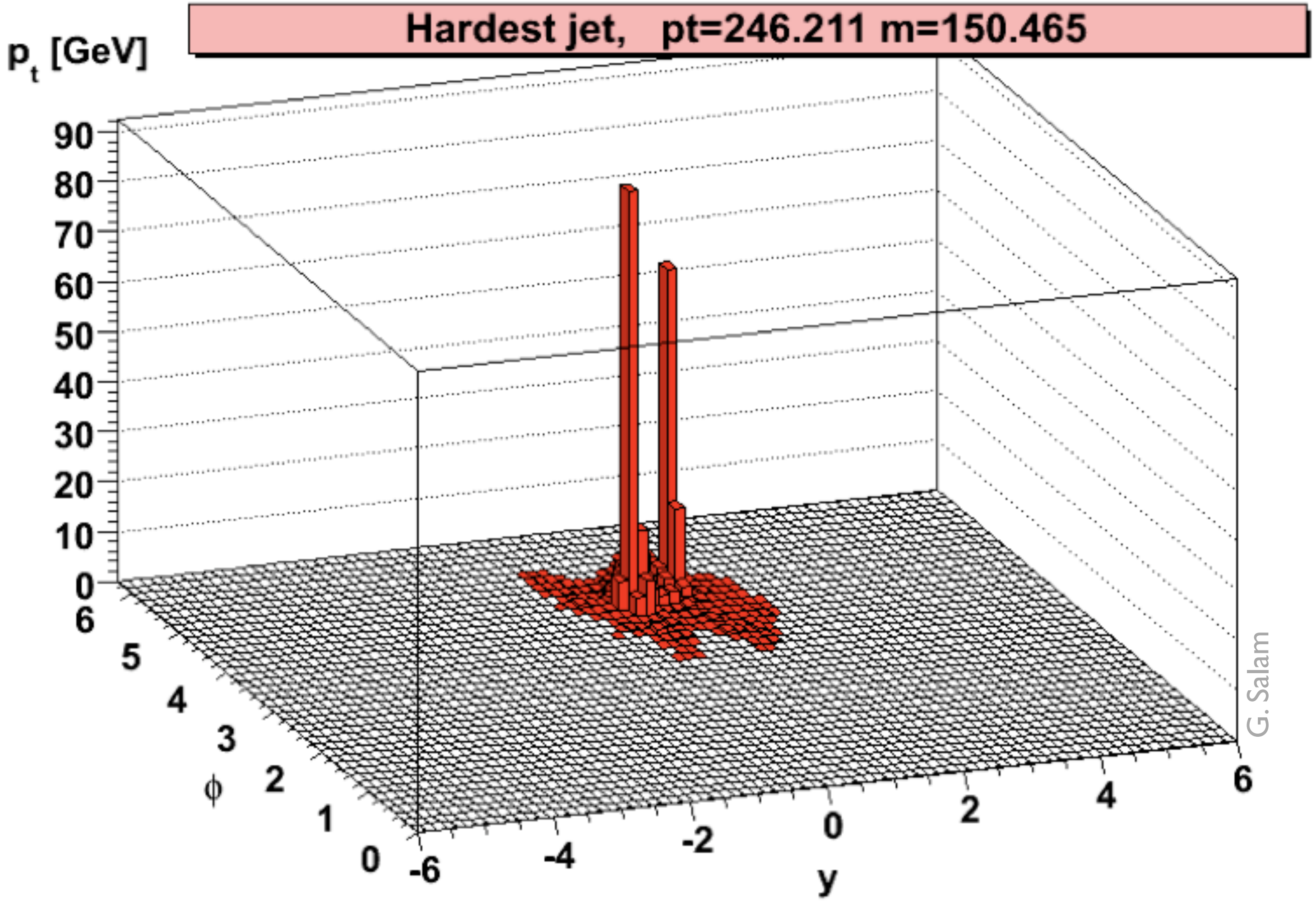
- Boosted Higgs tagger — Butterworth, Davison, Rubin, Salam, 2008
- Boosted top tagger — Kaplan, Rehermann, Schwartz, Tweedie, 2008; Thaler, Wang, 2008; G. Broojmans, ATLAS 2008
- Moderately boosted top and Higgs tagger — Plehn, Salam, Spannowsky, 2009

Common feature: start with a 'fat jet', decluster it and check if it contains a complex 'hard' substructure

Boosted Higgs tagger

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

Butterworth, Davison, Rubin, Salam, 2008



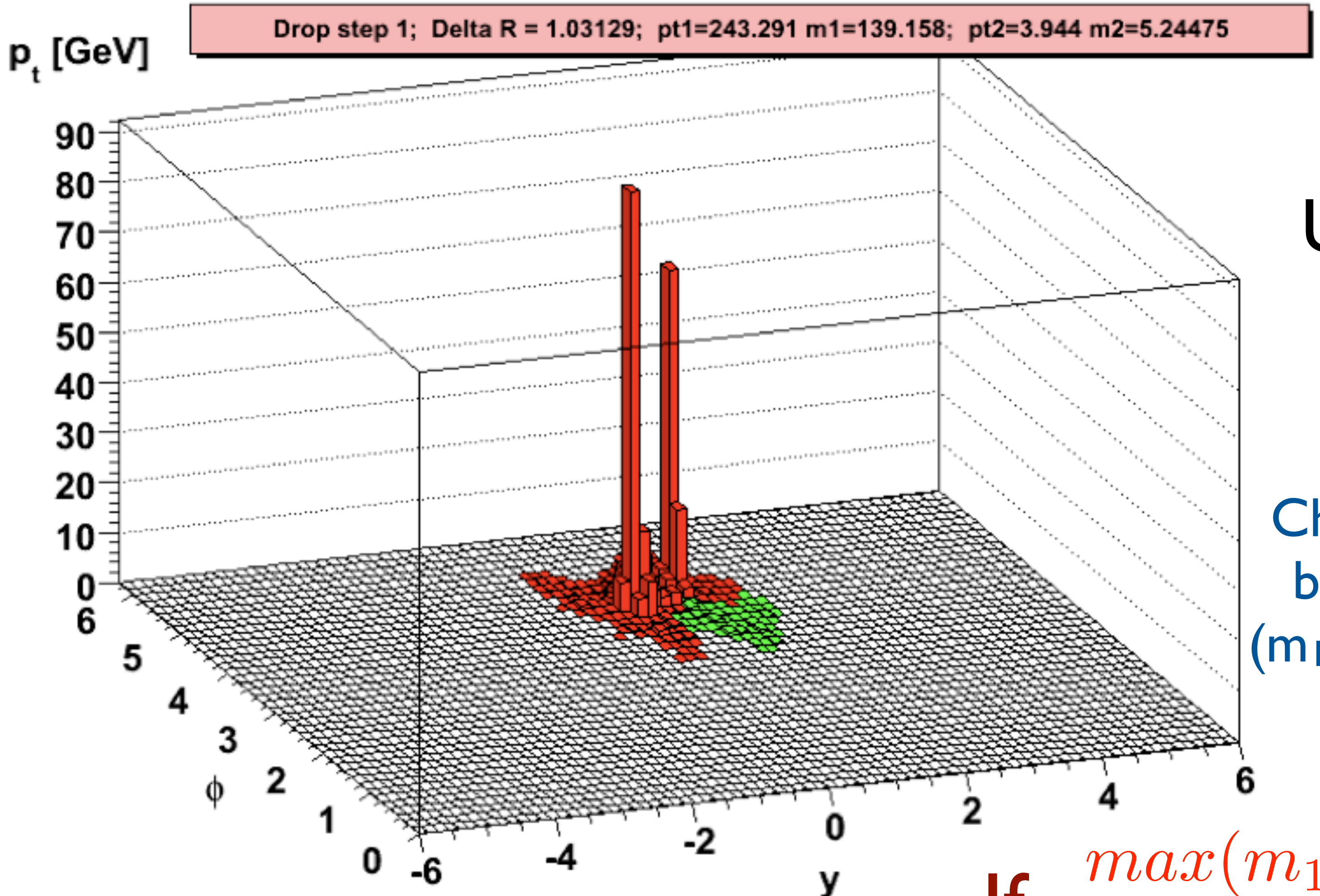
Start with the hardest jet

Use C/A with large R=1.2

m = 150 GeV

Boosted Higgs tagger

pp →ZH → ννbb



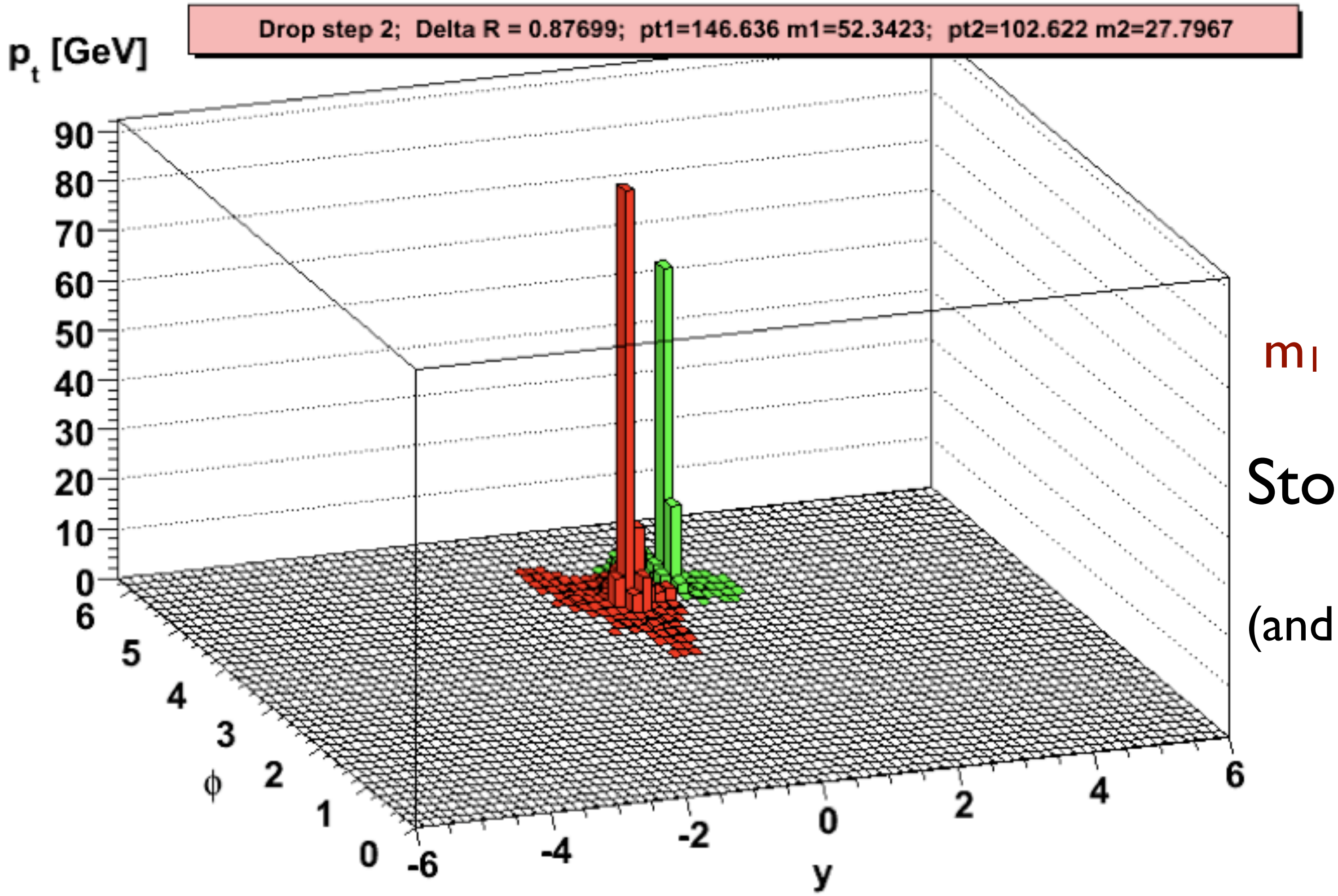
Undo last step of clustering

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

If $\dfrac{max(m_1, m_2)}{m} > \mu$ repeat

Boosted Higgs tagger

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



m_1 = 52 GeV, m_2 = 28 GeV

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

Jet substructure as filter

The ***jet substructure***
can be exploited to help **removing contamination**
from a soft background

- Jet 'filtering' — Butterworth, Davison, Rubin, Salam, 2008
- Jet 'pruning' — S. Ellis, Vermilion, Walsh, 2009
- Jet 'trimming' — Krohn, Thaler, Wang, 2009

Aim: limit sensitivity to background while retaining bulk of perturbative radiation

(Filtering, trimming and pruning are actually quite similar)

Cambridge/Aachen with filtering

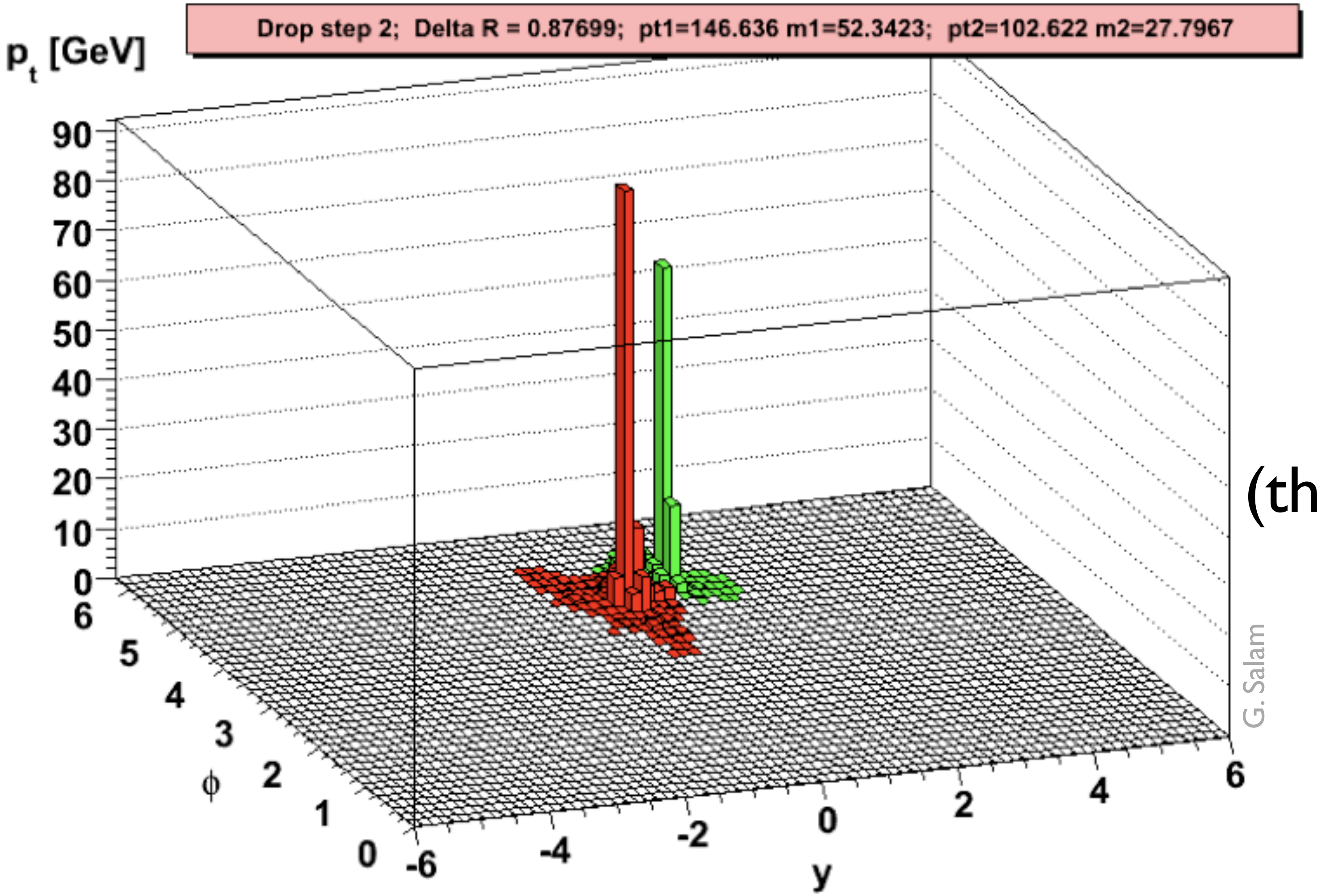
Butterworth, Davison, Rubin, Salam, arXiv:0802.2470

An example of a **third-generation** jet algorithm

- Cluster with C/A and a given R
- Undo the clustering of each jet down to subjets with radius $\mathbf{x_{filt}R}$
- Retain only the $\mathbf{n_{filt}}$ hardest subjets

Filtering in action

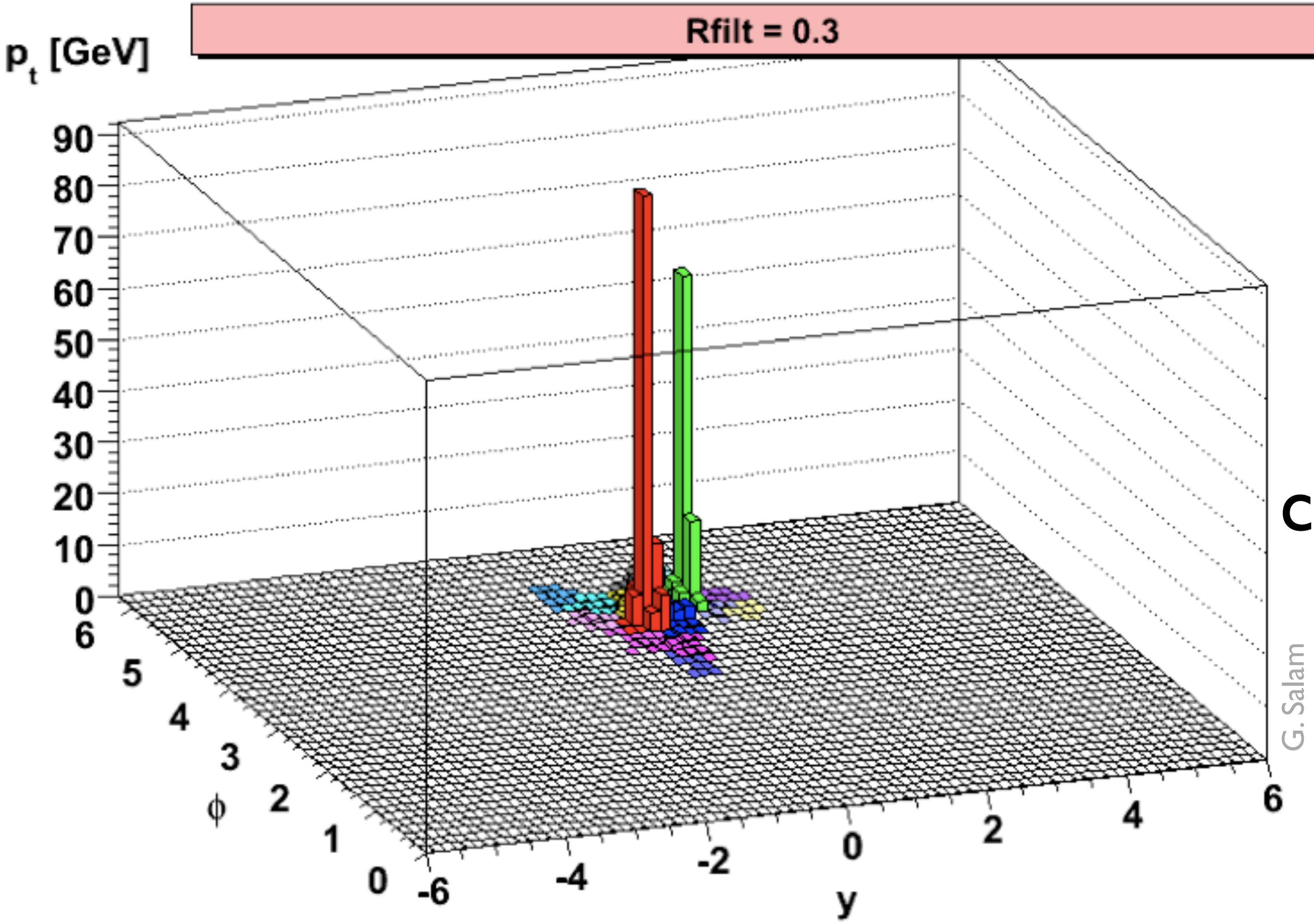
Butterworth, Davison, Rubin, Salam, arXiv:0802.2470



G. Salam

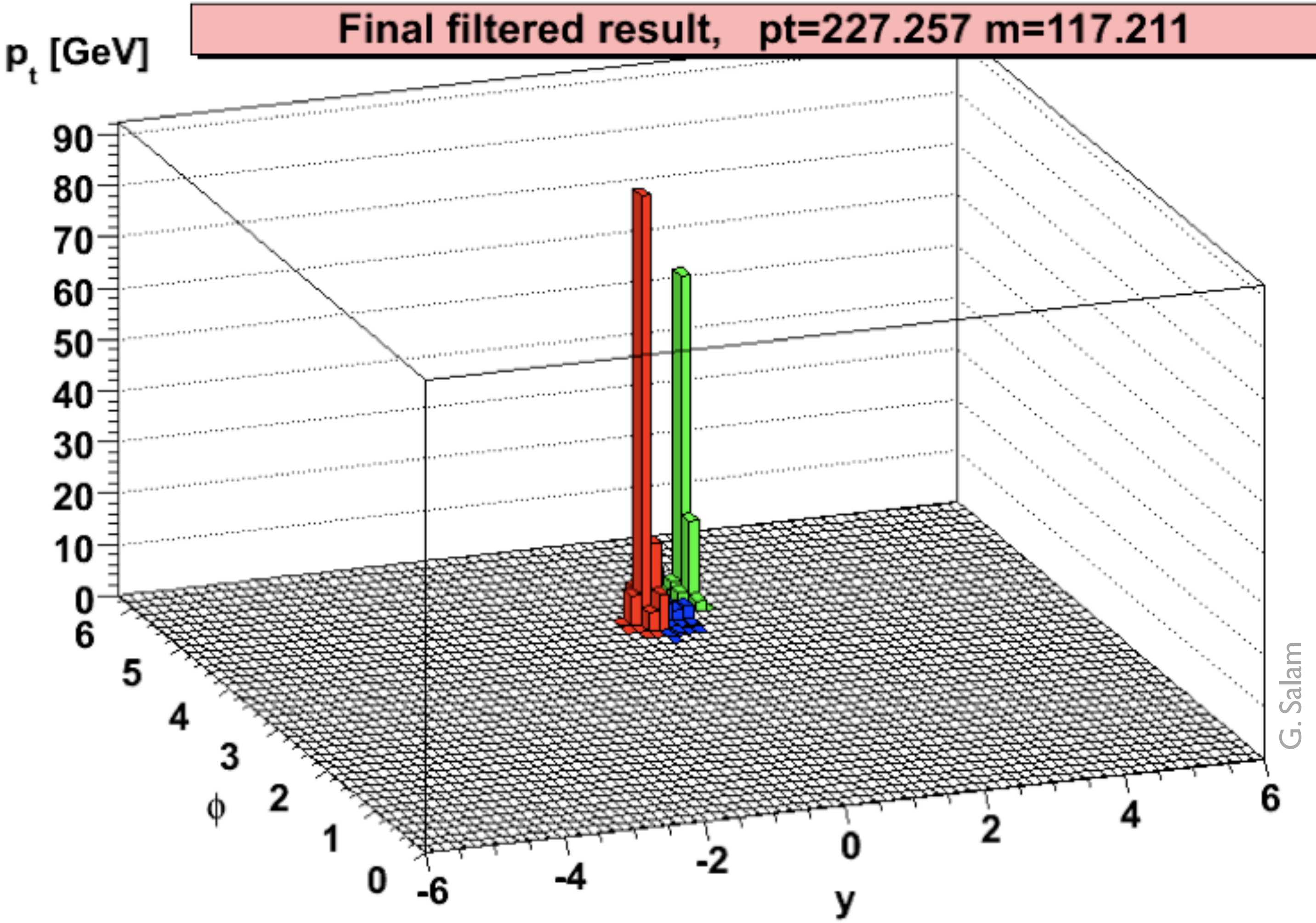
Start with a jet
(the sum of these two)

Filtering in action



Recluster the contituents with $R_{\rm filt}$

Filtering in action

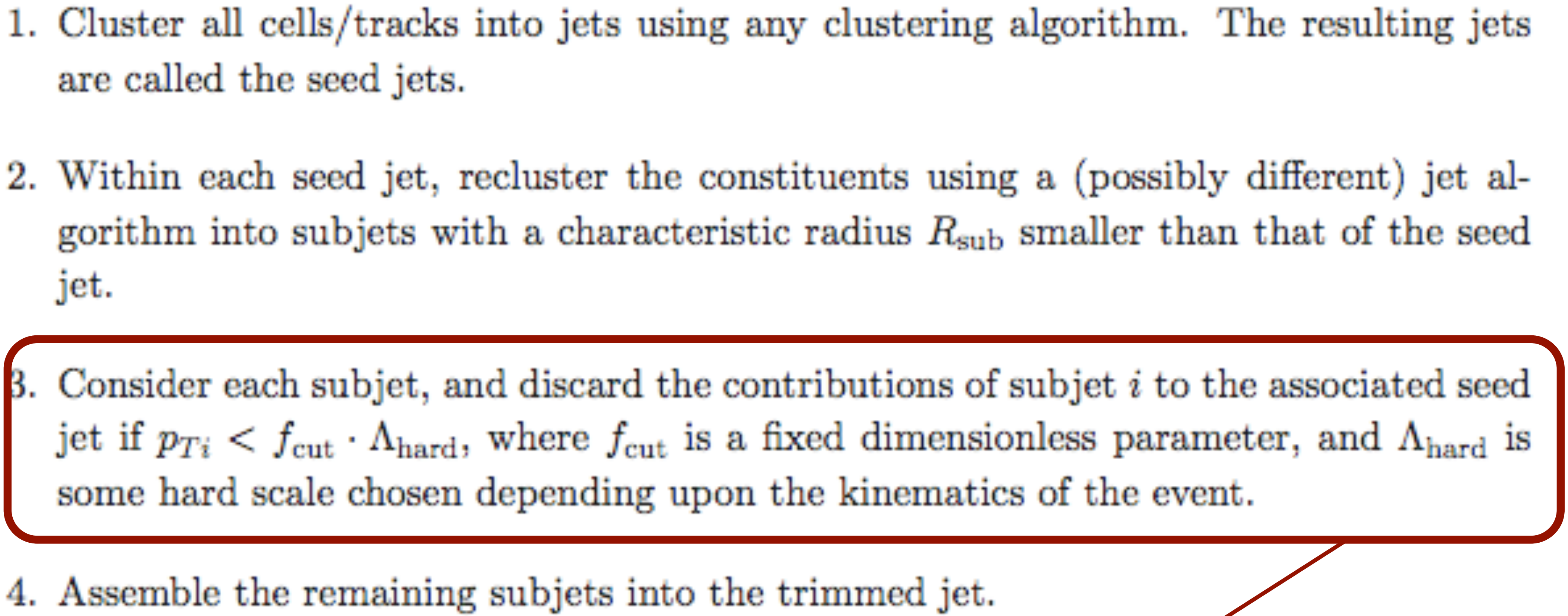


Only keep the n_{filt} hardest jets

The low-momentum stuff surrounding the hard particles has disappeared

Jet trimming

Krohn, Thaler, Wang, 2009

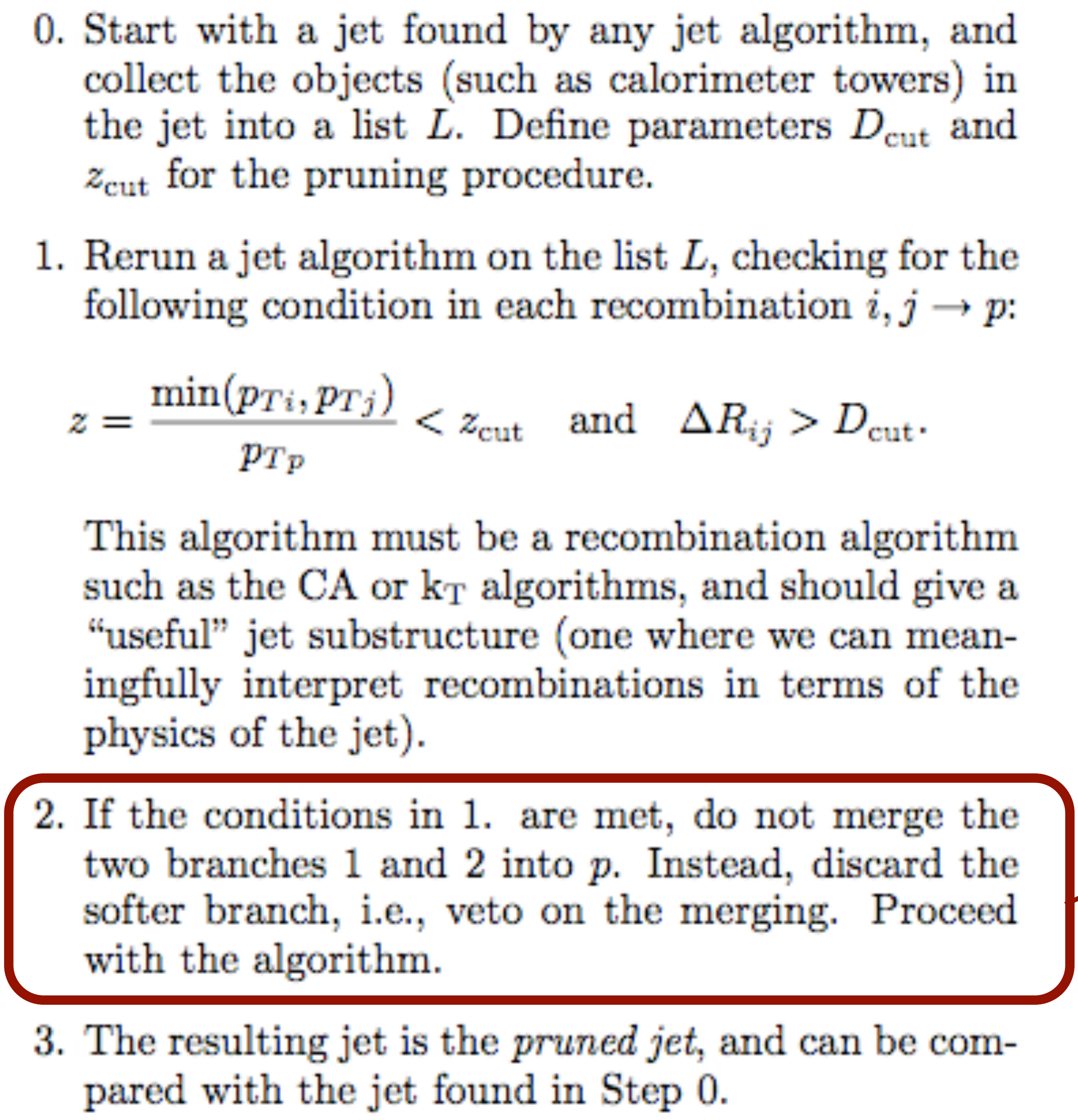


1. Cluster all cells/tracks into jets using any clustering algorithm. The resulting jets are called the seed jets.
2. Within each seed jet, recluster the constituents using a (possibly different) jet algorithm into subjets with a characteristic radius R_{sub} smaller than that of the seed jet.
3. Consider each subjet, and discard the contributions of subjet i to the associated seed jet if $p_{Ti} < f_{\text{cut}} \cdot \Lambda_{\text{hard}}$, where f_{cut} is a fixed dimensionless parameter, and Λ_{hard} is some hard scale chosen depending upon the kinematics of the event.
4. Assemble the remaining subjets into the trimmed jet.

Different condition for retaining jets
(p_T-cut rather than n_{filt} hardest)
with respect to filtering

Jet pruning

S. Ellis, Vermilion, Walsh, 2009



0. Start with a jet found by any jet algorithm, and collect the objects (such as calorimeter towers) in the jet into a list L. Define parameters $D_{\rm cut}$ and $z_{\rm cut}$ for the pruning procedure.

1. Rerun a jet algorithm on the list L, checking for the following condition in each recombination $i, j \to p$:

$$z = \frac{\min(p_{Ti}, p_{Tj})}{p_{Tp}} < z_{\rm cut} \quad \text{and} \quad \Delta R_{ij} > D_{\rm cut}.$$

   This algorithm must be a recombination algorithm such as the CA or k_T algorithms, and should give a "useful" jet substructure (one where we can meaningfully interpret recombinations in terms of the physics of the jet).

2. If the conditions in 1. are met, do not merge the two branches 1 and 2 into p. Instead, discard the softer branch, i.e., veto on the merging. Proceed with the algorithm.

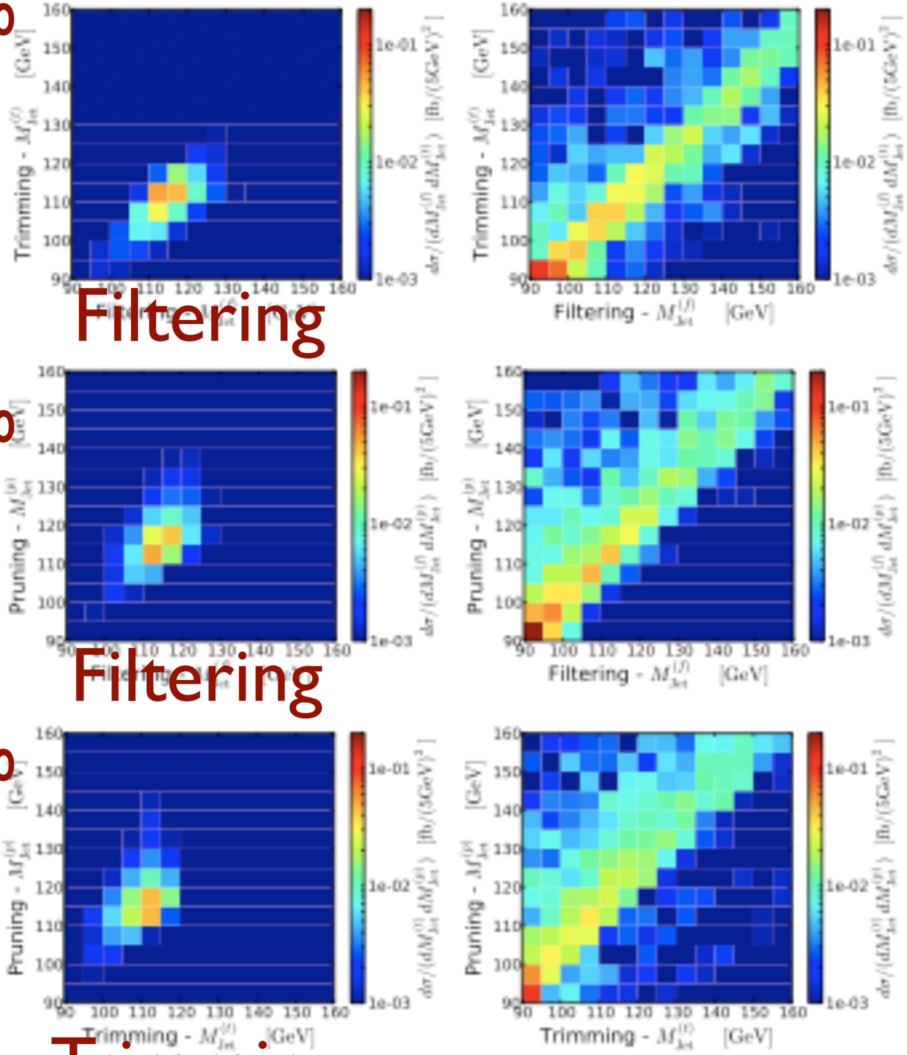
3. The resulting jet is the *pruned jet*, and can be compared with the jet found in Step 0.

Exclude soft stuff and large angle recombinations from clustering

filtering + pruning + trimming

Filtering, trimming and pruning are identical in aim and spirit (‘clean up’ a jet, keeping the hard core but getting rid of soft contamination’) but differ in details

ZH signal bkgd



Trimming

Filtering

Pruning

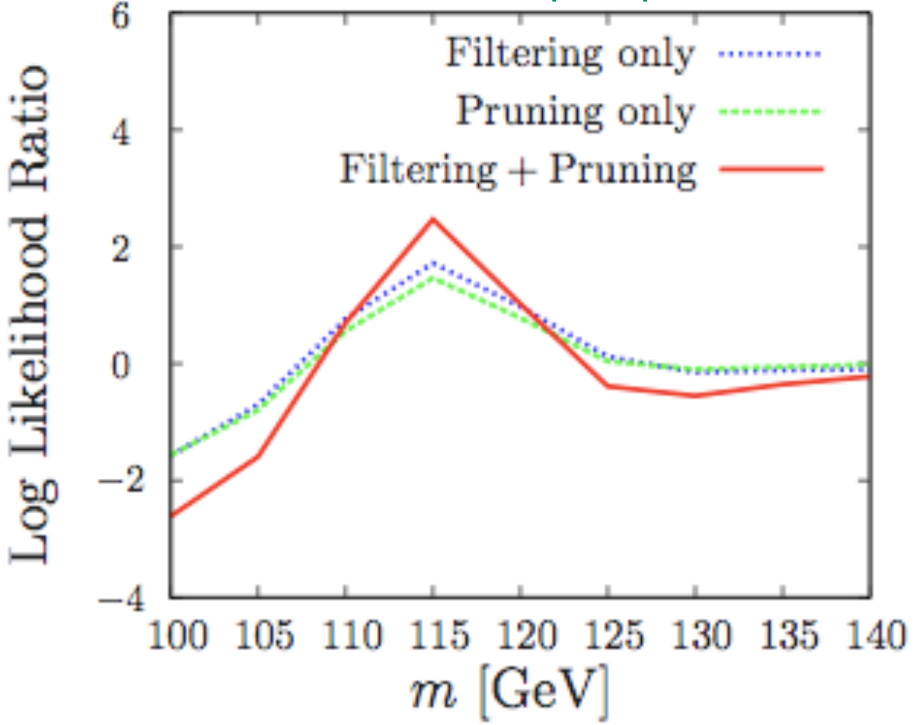
Filtering

Pruning

Trimming

This allows for combining them, obtaining an enhanced signal/ background significance

Soper, Spannowski, 2010



Jet area as a tool

Underlying event and pileup determination and subtraction

UE characterisation

Jet algorithms like k_t or Cambridge/Aachen allow one to determine
on an event-by-event basis
the **"typical" level of transverse momentum density**
of a **roughly uniform background noise**:

$$\rho \equiv \underset{\text{(over a \underline{single} event)}}{\text{median}} \left[\left\{ \frac{p_t^{jet}}{\text{Area}_{jet}} \right\} \right]$$

MC, Salam, 2007

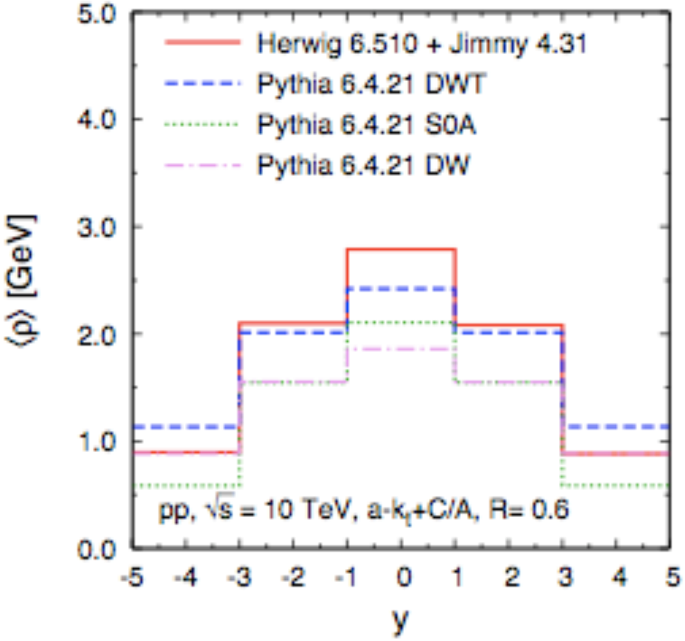
This ρ value can, in turn, be used to characterise the UE

Since this measurement is done with the jets, it is alternative/complementary to the usual analyses done using charged tracks (à la R. Field)

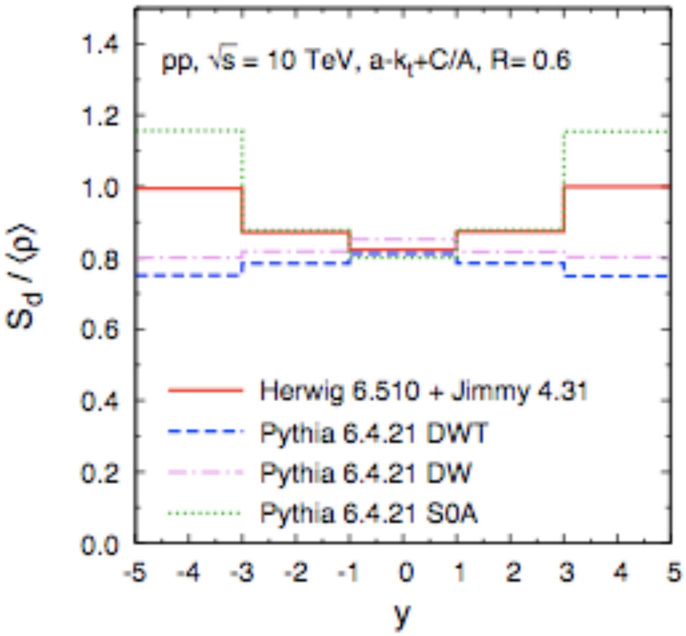
MC, Salam, Sapeta, 2009

UE characterisation

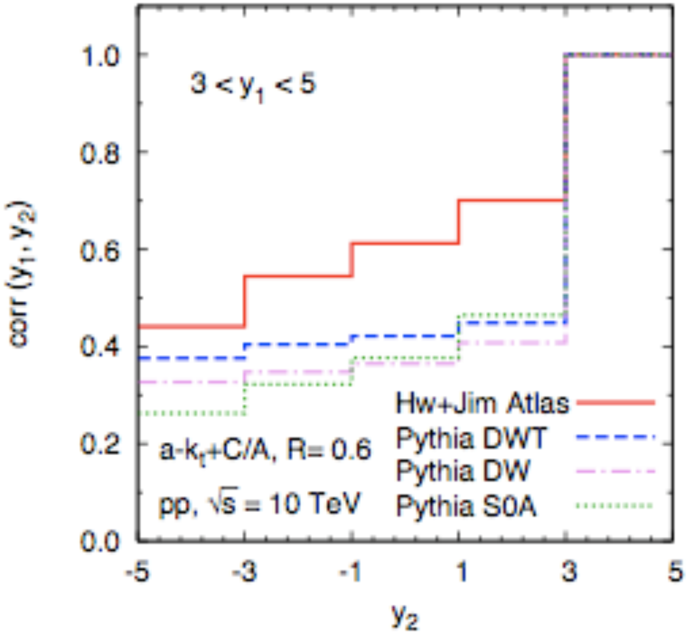
UE level



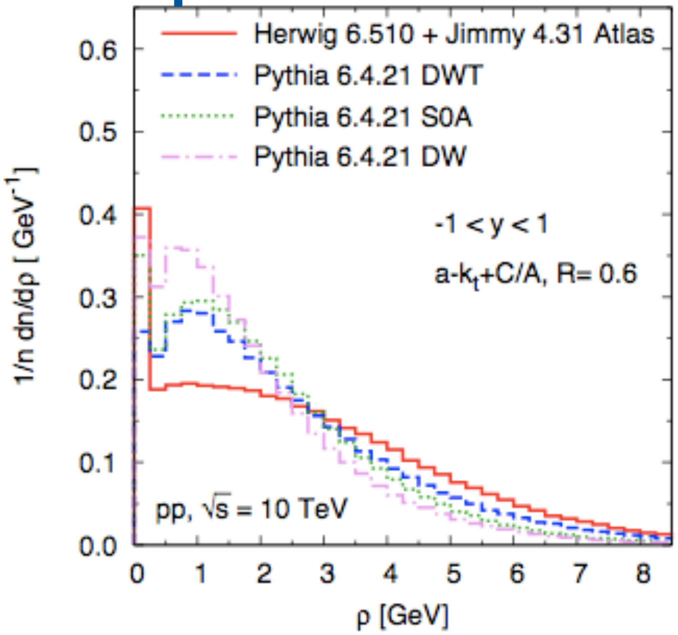
Its fluctuations



Correlations at different rapidities



ρ distribution



Potential discriminating power between different Pythia tunes, Herwig, etc

Useful for tuning MCs

Background subtraction

[MC, Salam, arXiv:0707.1378]

Once measured, the background density can be used to correct the transverse momentum of the hard jets:

$$p_T^{\text{hard jet, corrected}} = p_T^{\text{hard jet, raw}} - \rho \times \text{Area}_{\text{hard jet}}$$

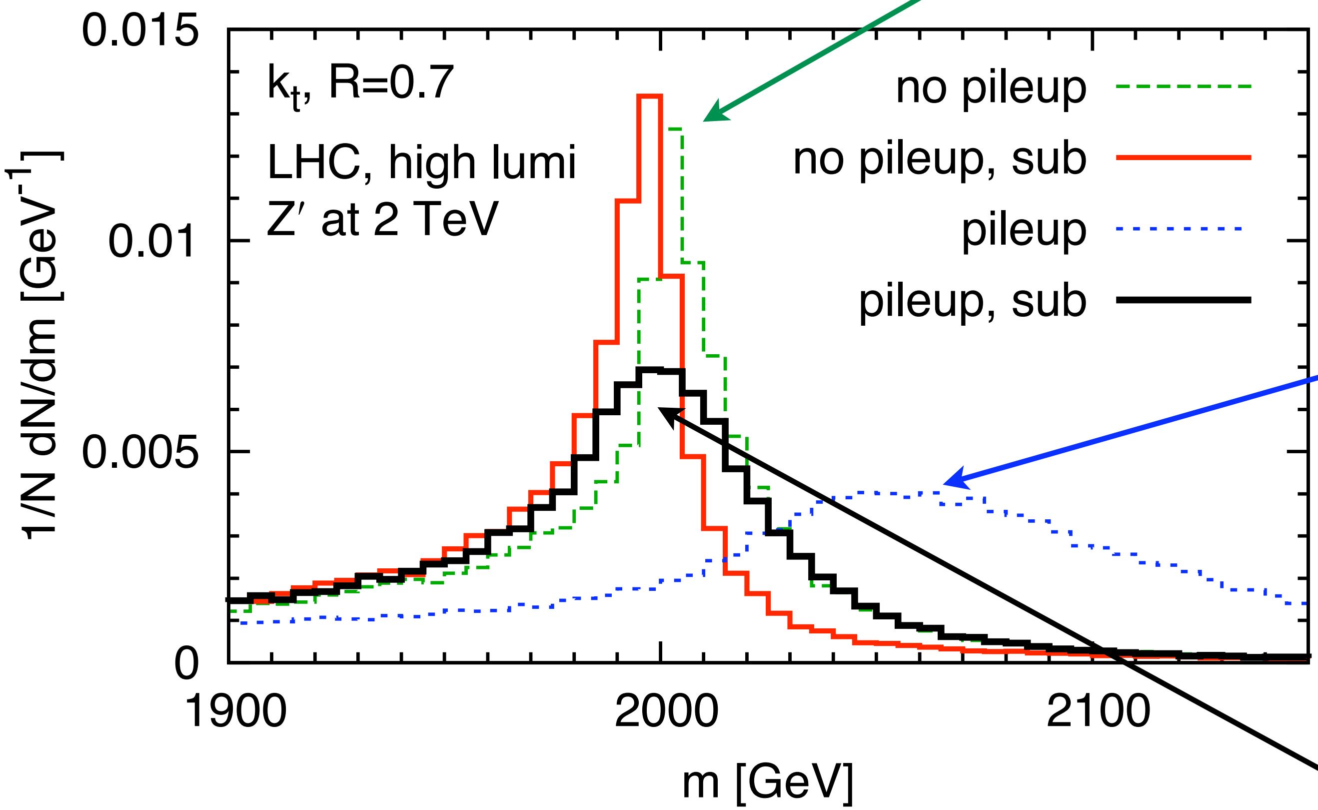
ρ being calculated on an event-by-event basis,
this procedure will generally improve the resolution of, say, a mass peak

NB. Also be(a)ware of *backreaction*
(immersing a hard jet in a soft background may cause some particles belonging to the hard event to be lost from (backreaction loss) or added to (backreaction gain) the jet).
Small effect for UE, larger for pileup, can be very important for heavy ions.
Analytical understanding of this effect available (MC, Salam, Soyez, arXiv:0802:1188)

Example of pileup subtraction

Let’s discover a leptophobic Z’ and measure its mass:

MC simulation:
m = 2000 GeV, width ~ 10 GeV



Naive measurement with PU:
m ~ 2050 GeV, width ~ 60 GeV

Measurement after subtraction:
m ~ 2000 GeV, width ~ 25 GeV

Mass reconstruction

The value of R matters because it affects,
in opposite ways, a number of things:

Small R:

Limit underlying event and pileup contamination

Better resolve many-jets events

Large R:

Limit perturbative radiation loss (‘out-of-cone’)

Limit non-perturbative hadronisation effects

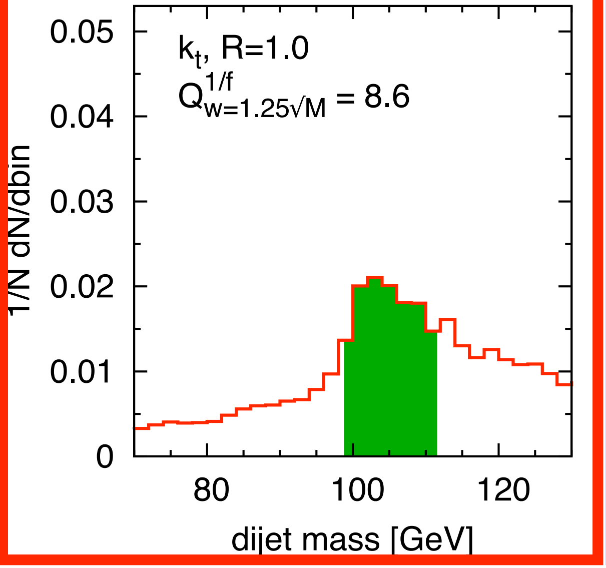
The best compromise will in general depend on the specific observable

Analytical estimates,
Dasgupta, Magnea, Salam, arXiv:0712.3014

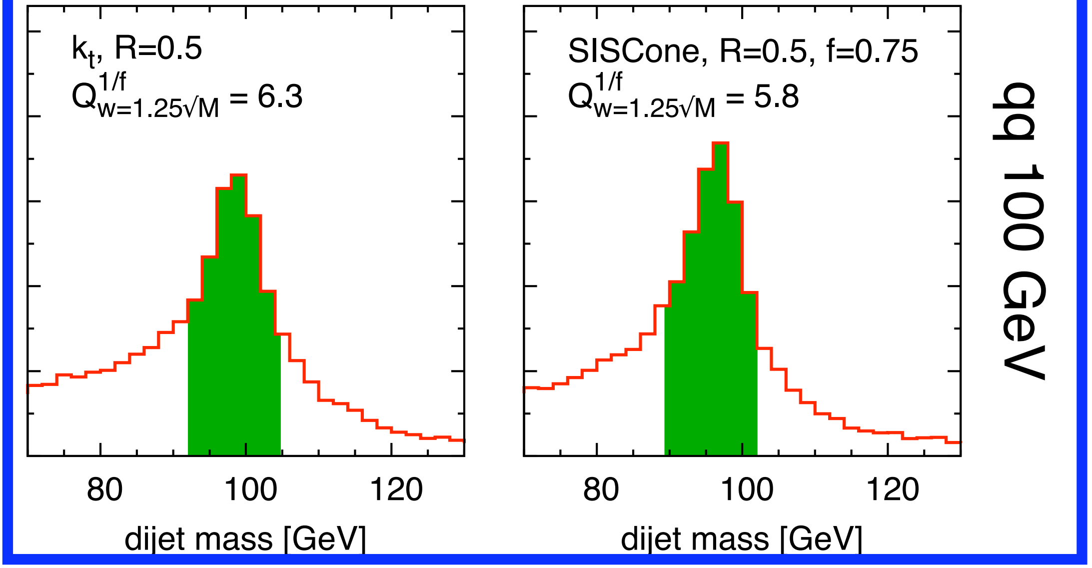
Mass reconstruction

MC, Rojo, Salam, Soyez, 2008

R=1: BAD



R=0.5: BETTER

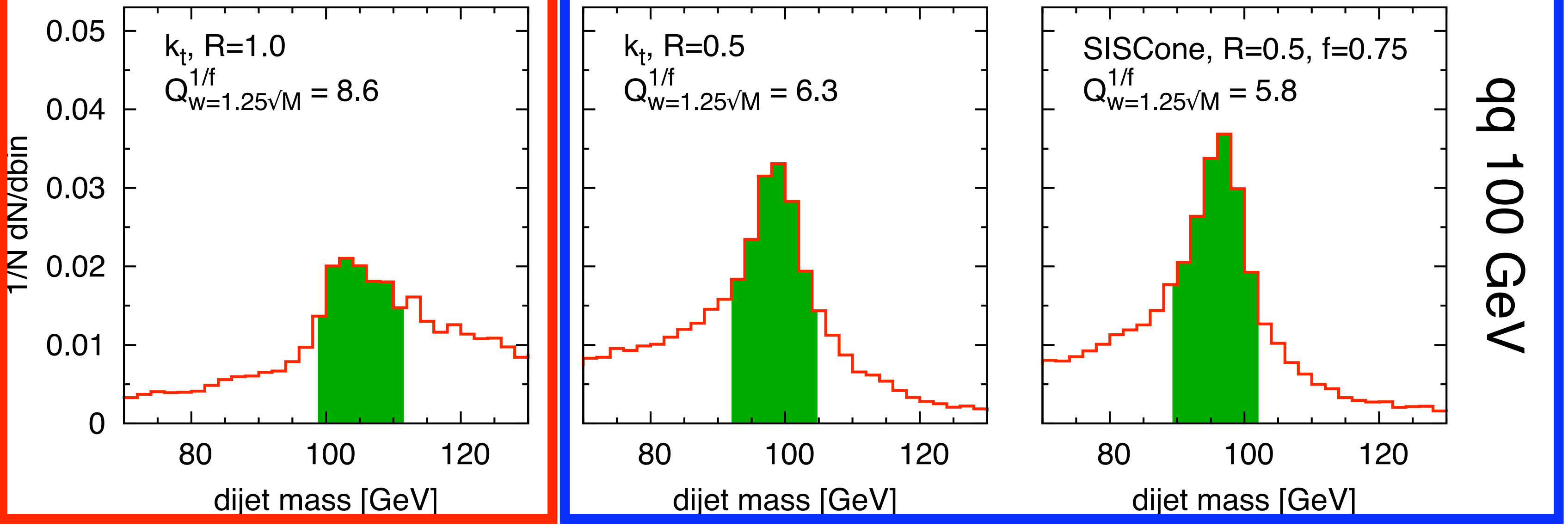


qq jets
at 100 GeV

Mass reconstruction

R=1: BAD

R=0.5: BETTER



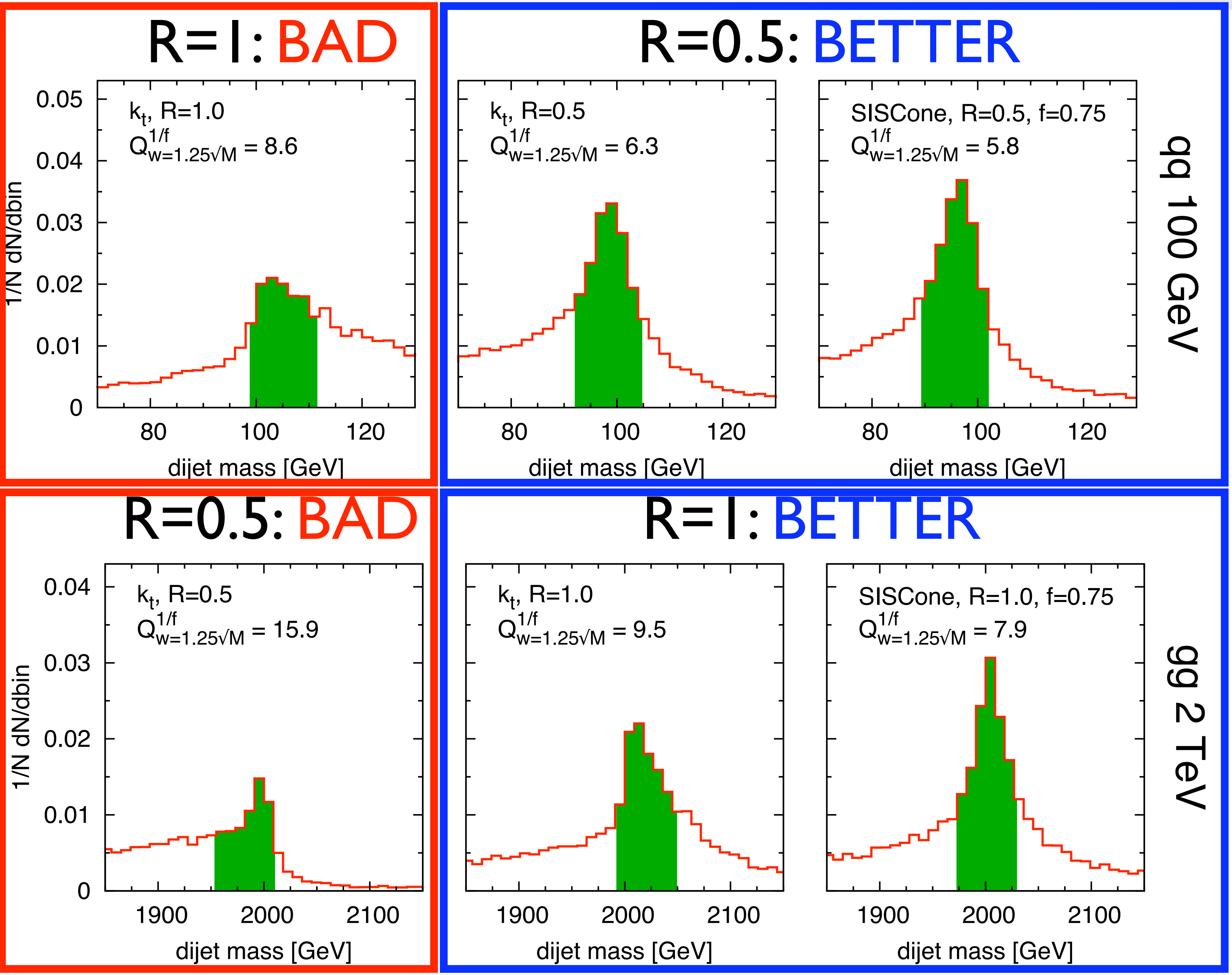
MC, Rojo, Salam, Soyez, 2008

qq jets
at 100 GeV

gg jets
at 2 TeV

Mass reconstruction

MC, Rojo, Salam, Soyez, 2008


R=1: BAD

R=0.5: BETTER

R=0.5: BAD

R=1: BETTER

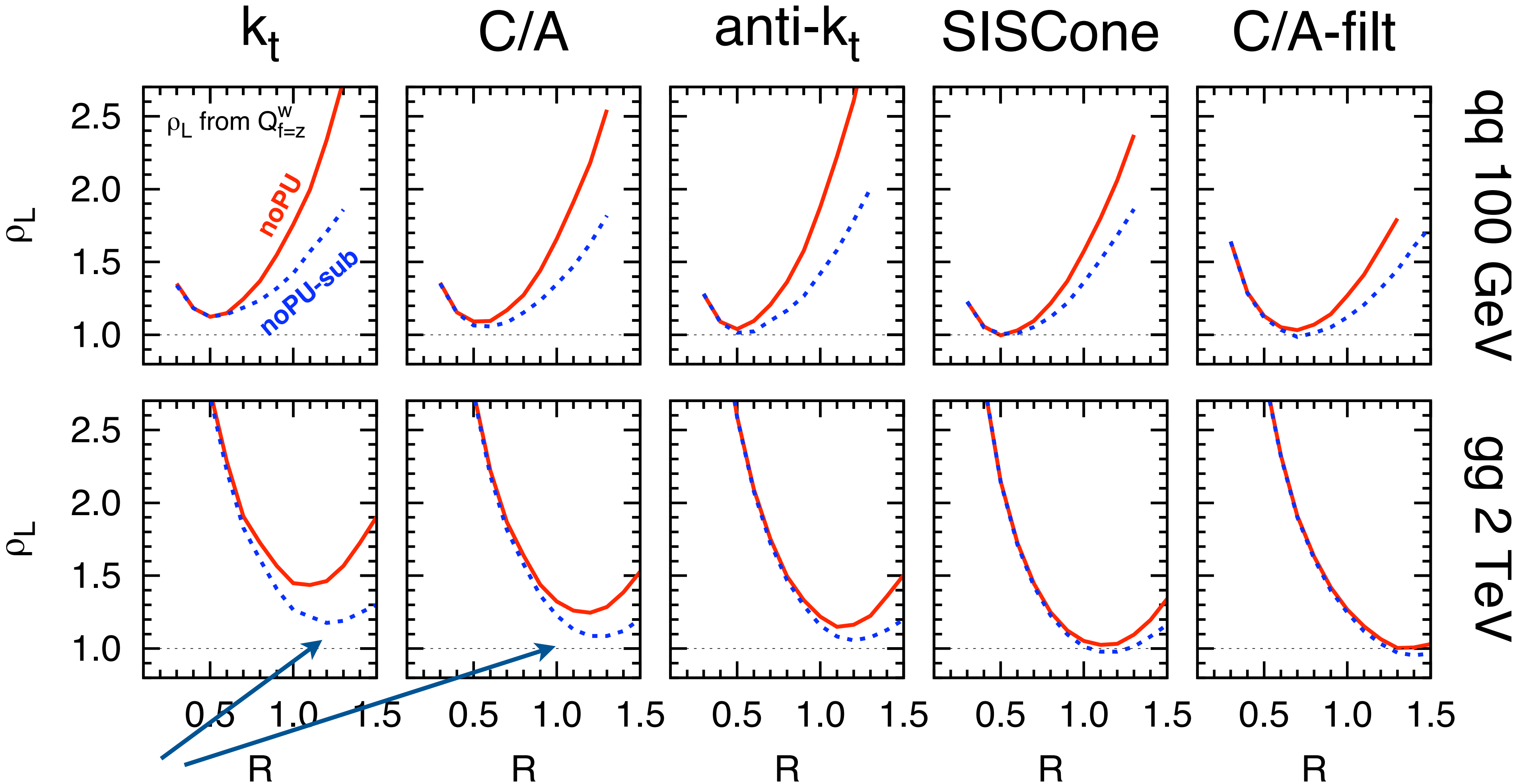
qq jets
at 100 GeV

gg jets
at 2 TeV

Gluons (and heavy objects) prefer larger R

Background subtraction

ρ_L = factor of luminosity needed to obtain equivalent signal/background significance



Effect of subtraction: bad jet definitions are improved.

Gain in effective luminosity about 20-30%

(i.e. they were 'bad' due to a large extent to their behaviour with respect to the background)

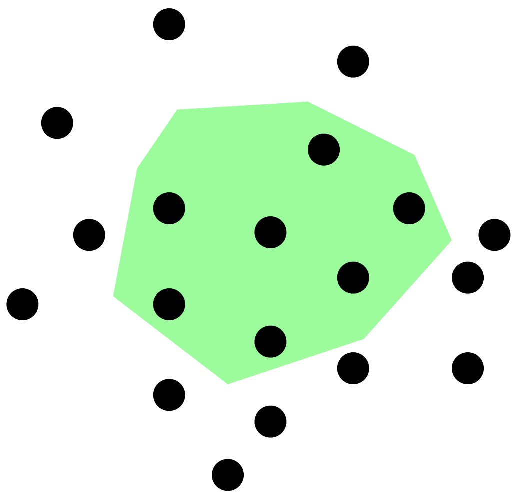
Filtering can help in mass reconstruction too

(and can also be further subtracted)

Backreaction

“How (much) a jet changes when immersed in a background”

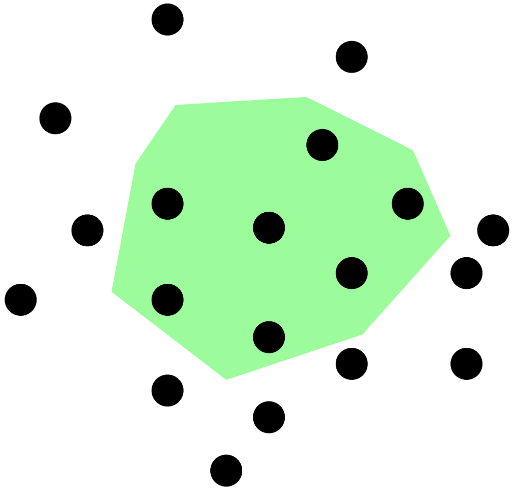
Without background

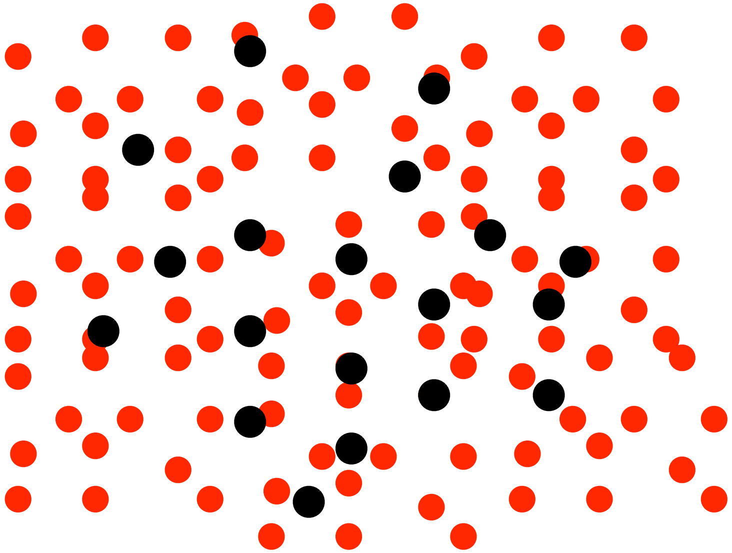


Backreaction

"How (much) a jet changes when immersed in a background"

Without background

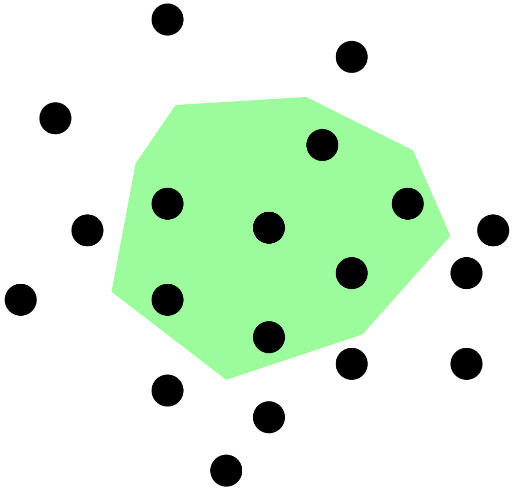




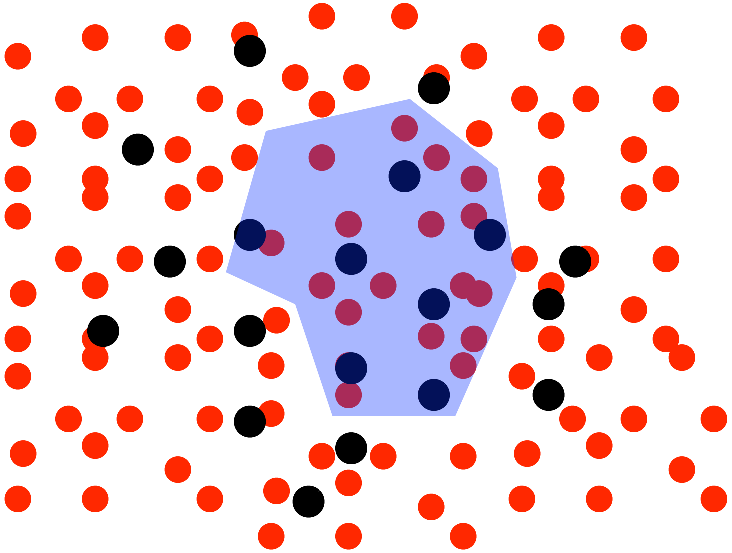
Backreaction

"How (much) a jet changes when immersed in a background"

Without background

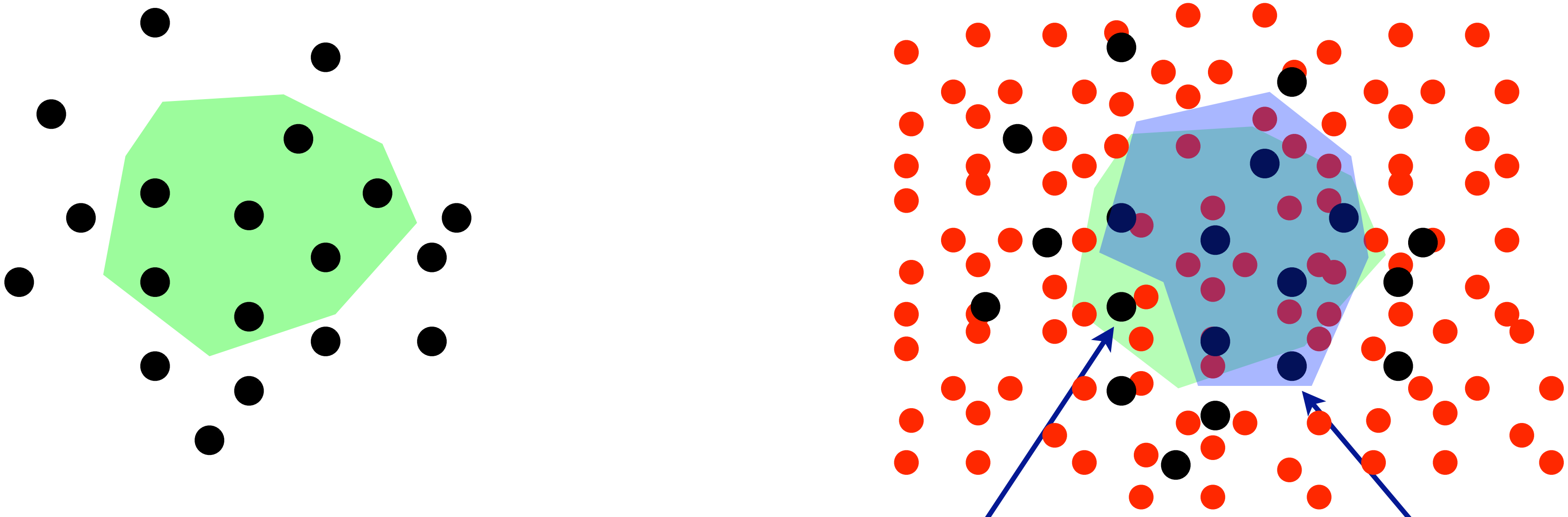


With background



Backreaction

"How (much) a jet changes when immersed in a background"



Without background

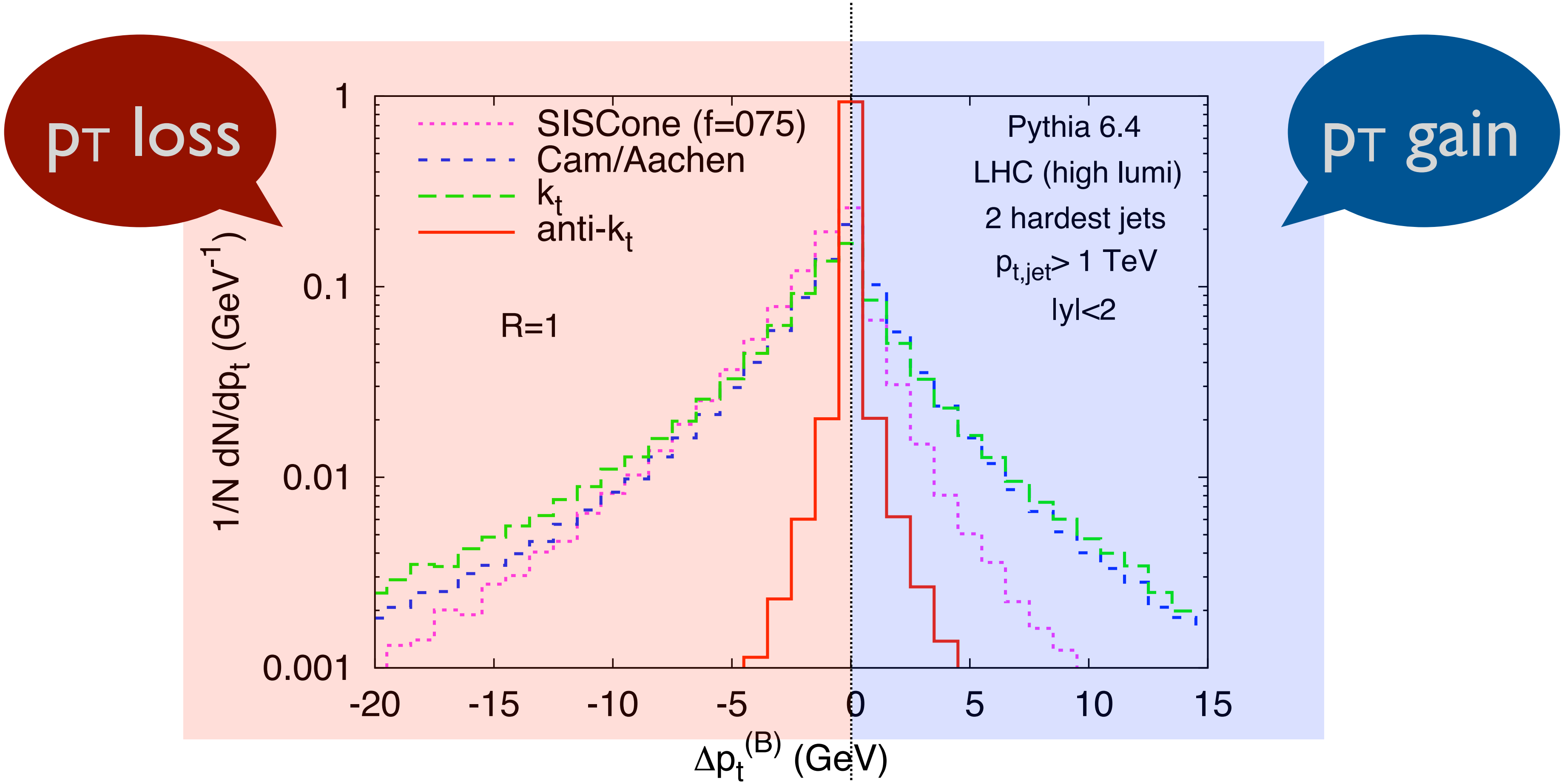
With background

Backreaction **loss**

Backreaction **gain**

Backreaction

MC, Salam, Soyez, arXiv:0802.1188



Anti-k_t jets are much more resilient to changes from background immersion

Heavy Ions

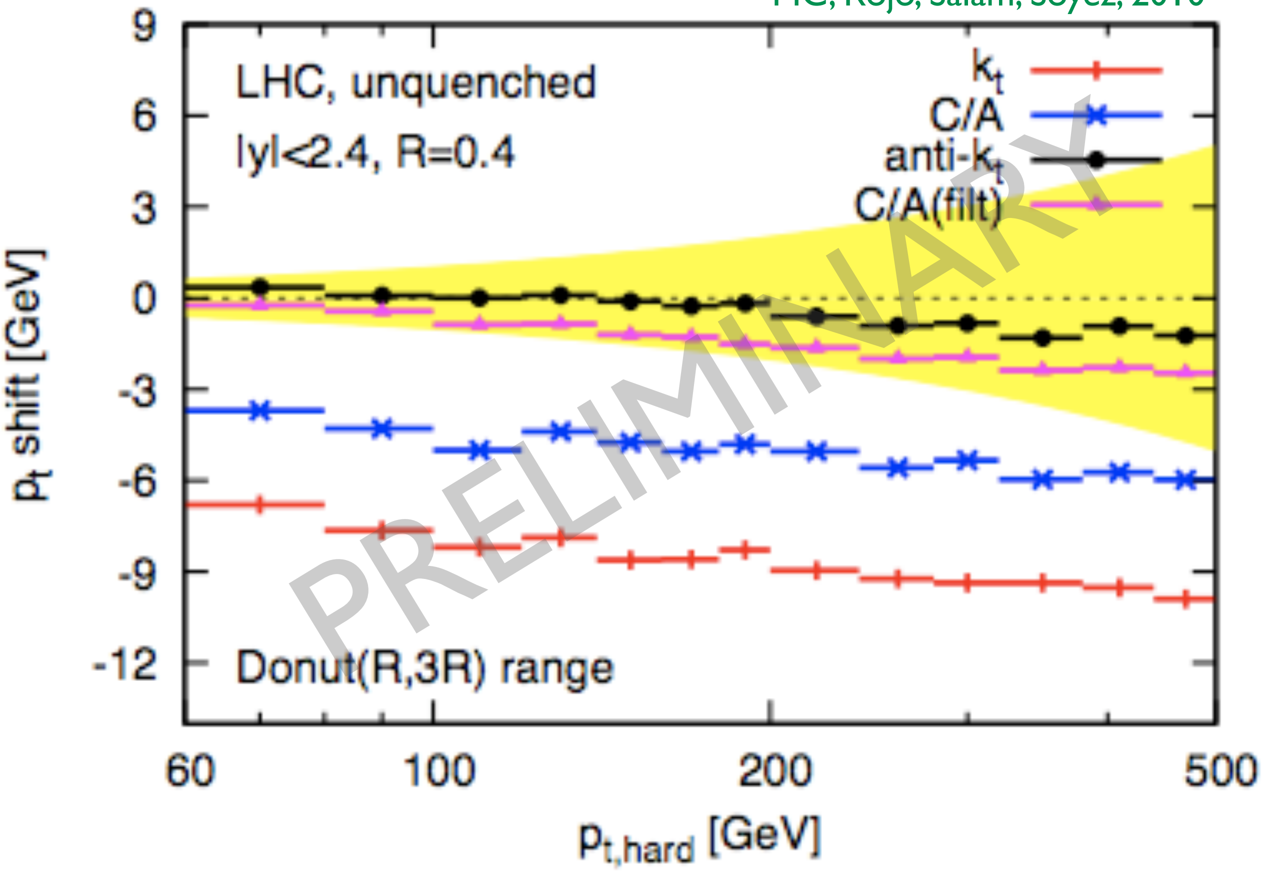
Many of the jet techniques just described can also be considered for heavy ions

In particular, one can perform background determination and subtraction
(about 250 GeV per unit area at LHC!)

Residual bias after subtraction

(anti-k_t doing better due
to its soft-resiliency leading to
vanishing back-reaction)

MC, Rojo, Salam, Soyez, 2010



Flexibility

All IRC safe algorithms are equal, but some are more equal than others

Depending on the analysis you wish to perform,
a jet definition might give better results than others

The IRC safe algorithms

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

Conclusions

Conclusions

☑ An extensive set of fast, IRC safe jet algorithms exists, offering replacements for the IRC unsafe ones.

Conclusions

☑ An extensive set of fast, IRC safe jet algorithms exists, offering replacements for the IRC unsafe ones.

☑ They offer ample **flexibility** in choosing the most effective jet definition for any given analysis.

Conclusions

☑ An extensive set of fast, IRC safe jet algorithms exists, offering replacements for the IRC unsafe ones.

☑ They offer ample **flexibility** in choosing the most effective jet definition for any given analysis.

☑ They can be used to estimate the level of a uniformly distributed noise, study its characteristics, and subtract from the hard jets, improving the quality of kinematical reconstructions.

Conclusions

☑ An extensive set of fast, IRC safe jet algorithms exists, offering replacements for the IRC unsafe ones.

☑ They offer ample **flexibility** in choosing the most effective jet definition for any given analysis.

☑ They can be used to estimate the level of a uniformly distributed noise, study its characteristics, and subtract from the hard jets, improving the quality of kinematical reconstructions.

☑ ‘Third-generation’ algorithms exploiting jet substructure look very promising.