DESY Hamburg
September 22, 2009

Recent Progress in Jets
a.k.a.
The path from Clustering to Jetography

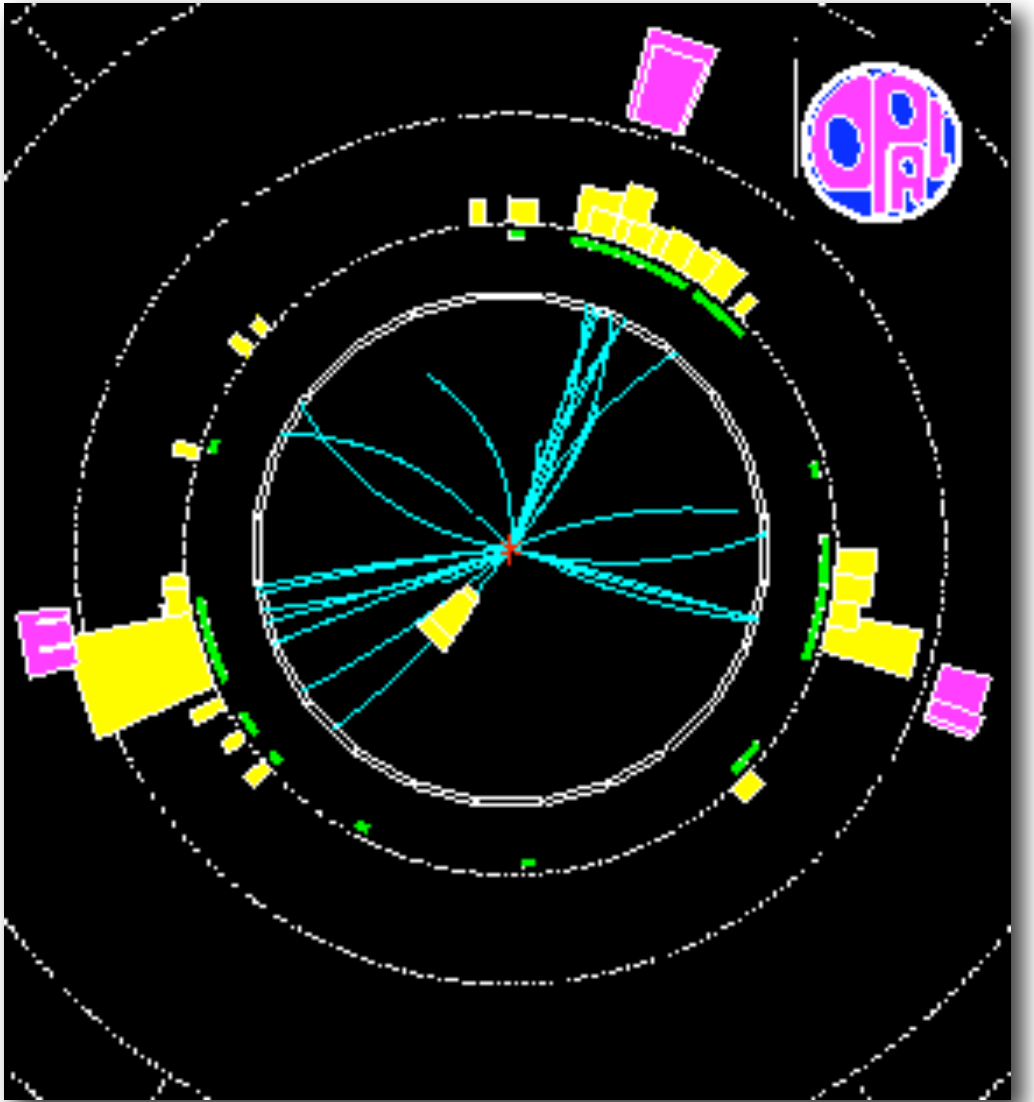
Matteo Cacciari
LPTHE - Paris 6,7 and CNRS

Contains work done in collaboration with G. Salam, G, Soyez, J. Rojo

Outline

- Introductory remarks and review
- New jet algorithms (anti-k_t, SISCone) and FastJet
- New properties/tools: areas, backreaction, filtering
- Two examples: *jet quality for a mass peak* and *Higgs search using jet substructure*
- Summary/conclusions

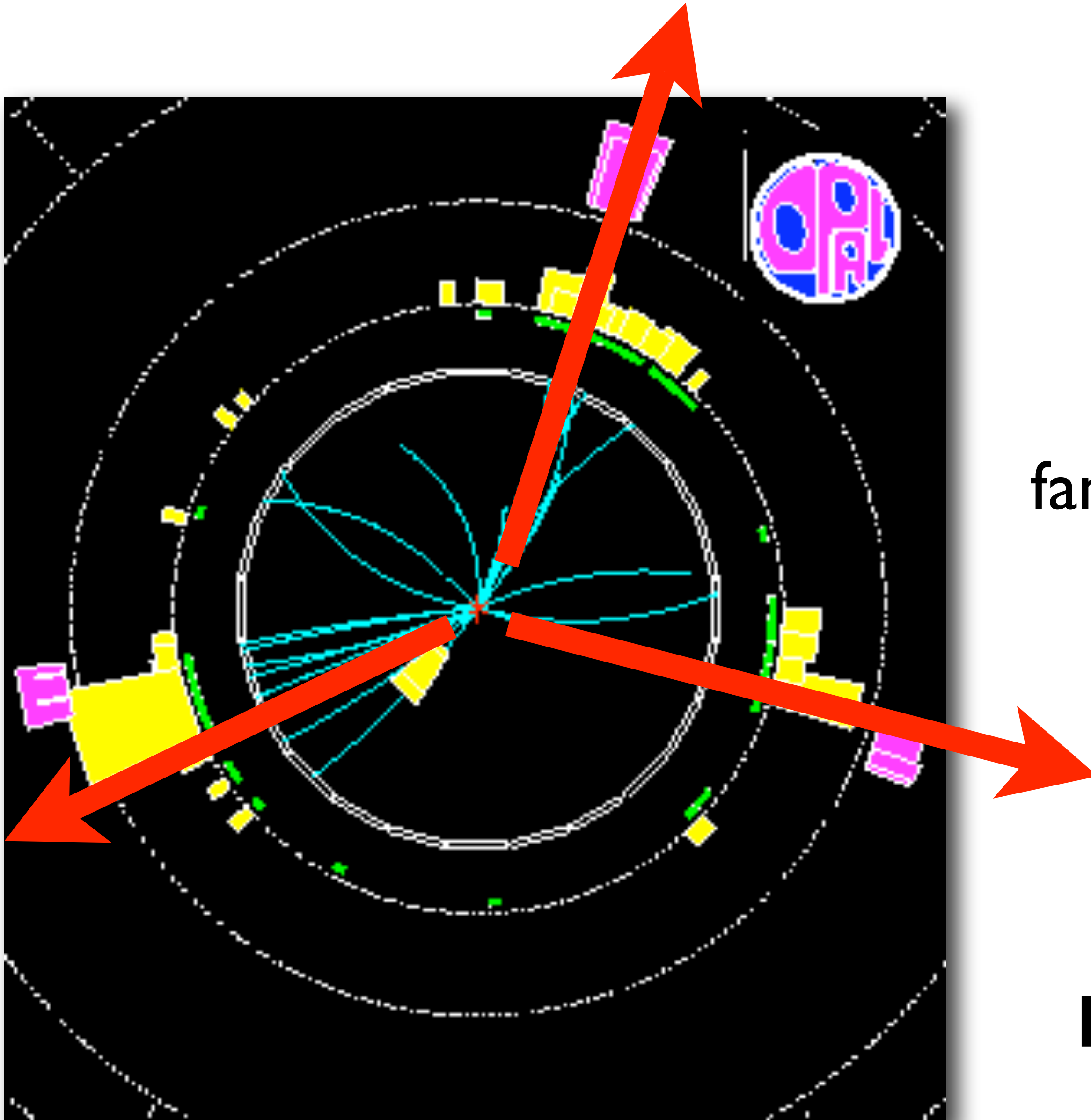
Why jets



A jet is something that happens in high energy events:

a collimated bunch of hadrons flying roughly in the same direction

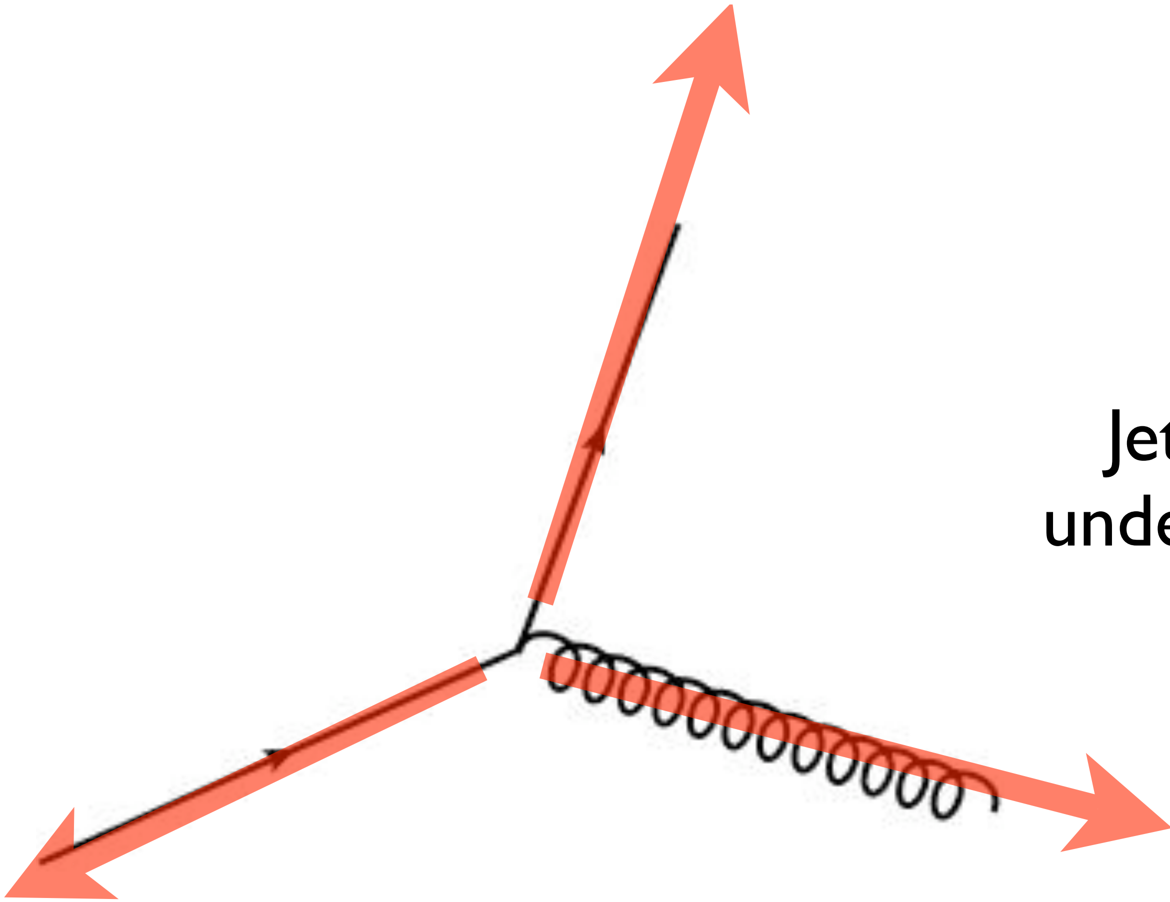
Note: hundreds of hadrons contain **a lot** of information. More than we can hope to make use of



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



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

The 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

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

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, …...

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

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

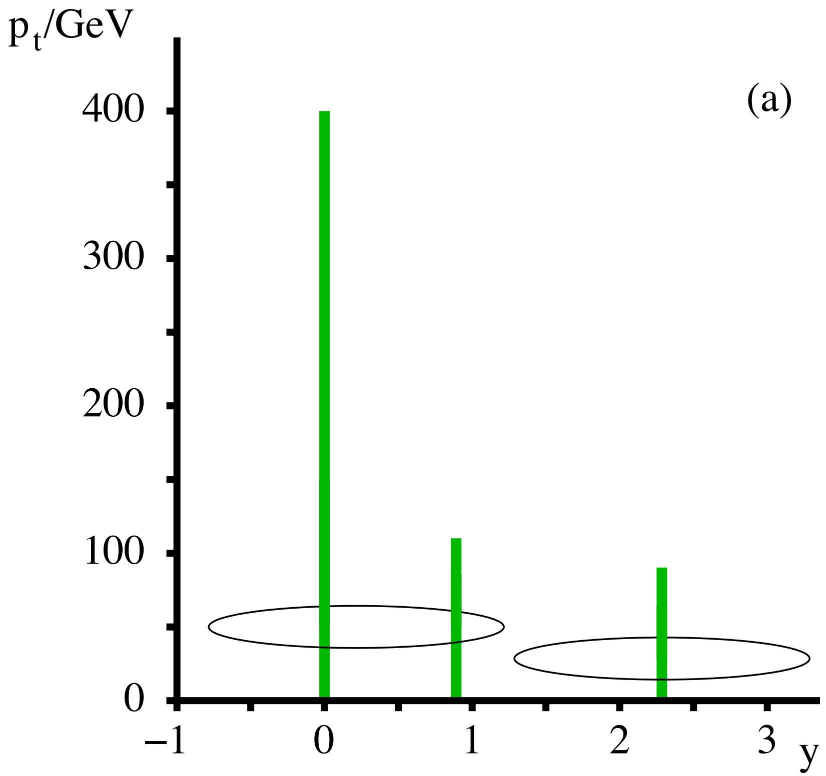
Too slow.
Resort to approximate methods

Example of IC-SM: MidPoint Cone

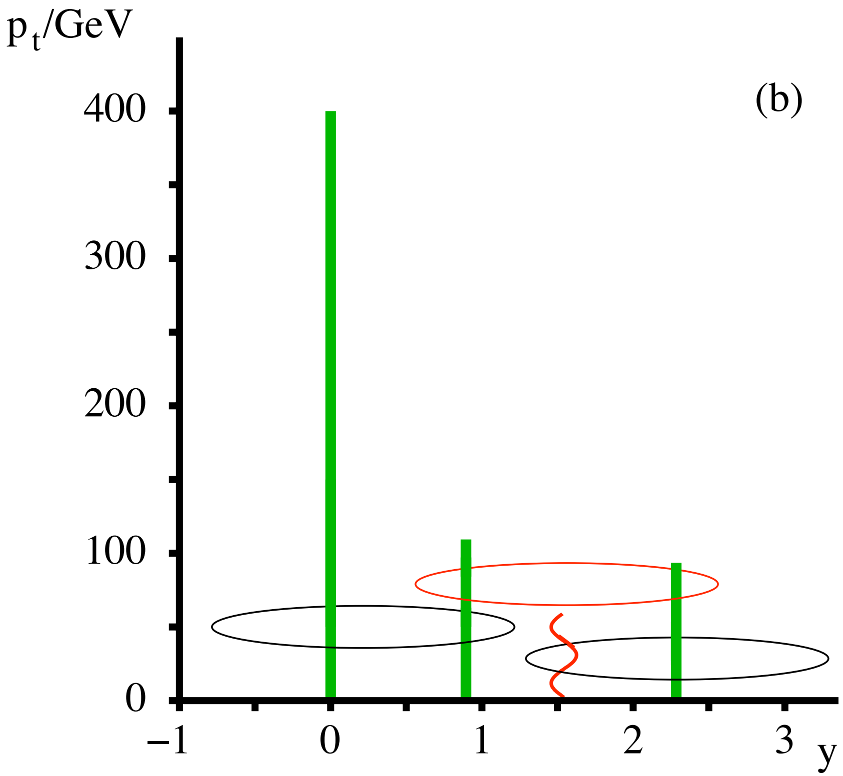
- Begin with seed particles
- Cluster particles into cone if $\Delta R < R$
- Iterate until stable (i.e. axis coincide with sum of momenta) cones found
- Start new search cones at midpoint of stable cones
- Merge jets if overlapping energy is > f times the energy of the smaller jet

Use of seeds is the most problematic issue

MidPoint infrared unsafety



Three hard particles clustered into **two** cones by the MidPoint algorithm



Addition of a **soft** particle changes the hard jets: **three** stable cones are now found

G. Salam

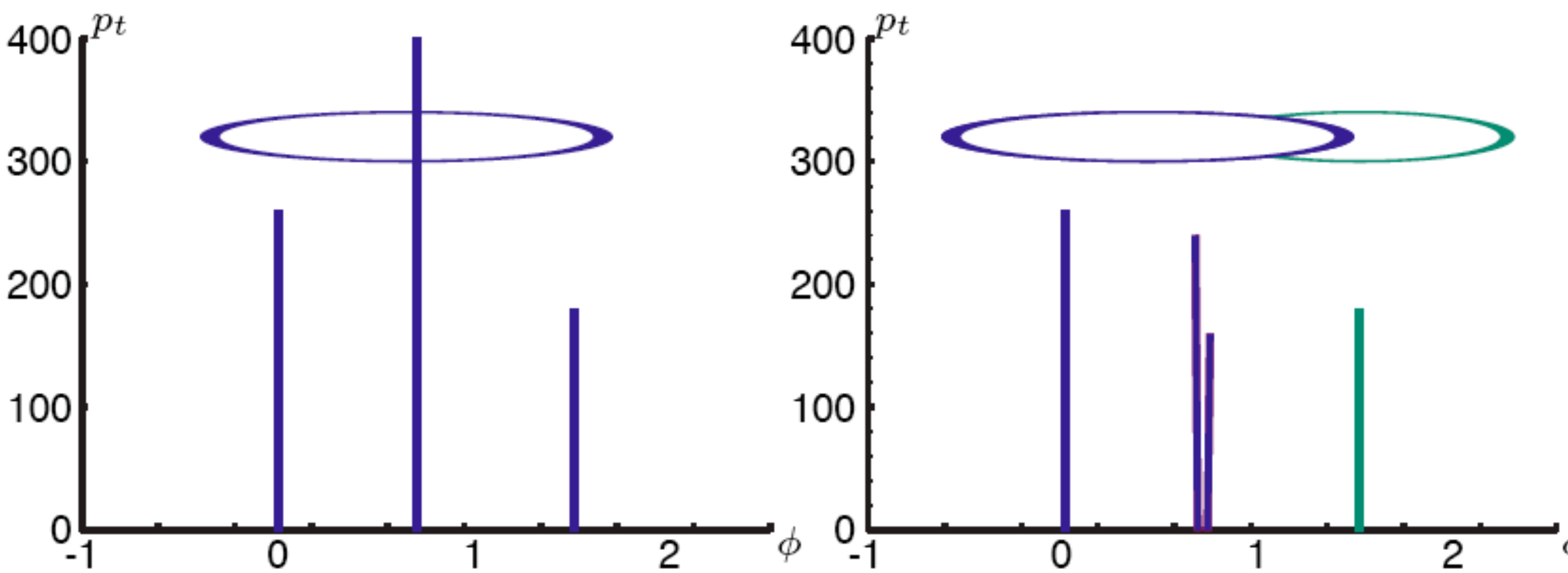
The problem is that the specific stable-cone search procedure used by MidPoint cannot find **all** possible stable cones

Example of IC-PR (e.g. CMS cone)

- Begin with hardest particle as seed
- Cluster particles into cone if $\Delta R < R$
- Iterate until stable (i.e. axis coincide with sum of momenta) cones found
- Eliminate constituents of jet and start over from hardest remaining particle

NB. This is a very different algorithm from previous one.
Many physics aspects differ.

IC-PR cone collinear unsafety



G. Soyez

Splitting the hardest particle **collinearly** changes the number of final jets

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	$\mathrm{IC}_{se,mp}$-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

- There isn't **one** cone algorithm, but rather many different cones, which can behave quite distinctly from one another
- Essentially all of the cones commonly used are <u>unsafe</u> at some point. The best ones only fail at NNLO (3+1), others already at NLO (2+1)

Examples:

	Last meaningful order		
	ATLAS cone [IC-SM]	MidPoint [IC$_{mp}$-SM]	CMS it. cone [IC-PR]
Inclusive jets	LO	NLO	NLO
$W/Z + 1$ jet	LO	NLO	NLO
3 jets	**none**	LO	LO
$W/Z + 2$ jets	**none**	LO	LO
m_{jet} in $2j + X$	**none**	**none**	**none**

Calculations cost real money:
~ 100 theorists ×15 years ≈100 M€

Using unsafe jet tools essentially renders them useless

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 kt 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 $N^2 \ln N$

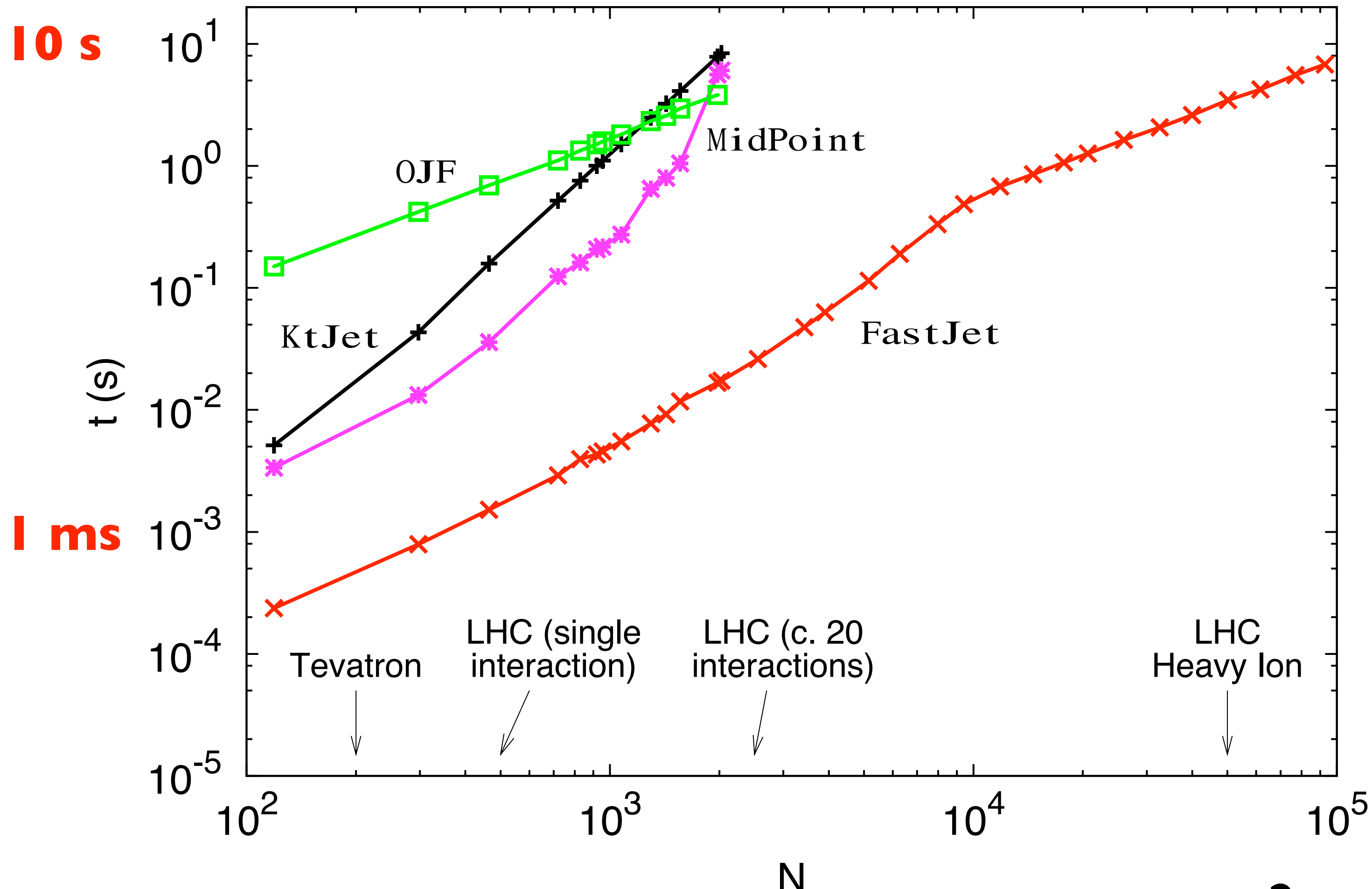
  (Salam, Soyez, arXiv: 0704.0292)

Both implementations (and a lot more) available via FastJet

www.fastjet.fr

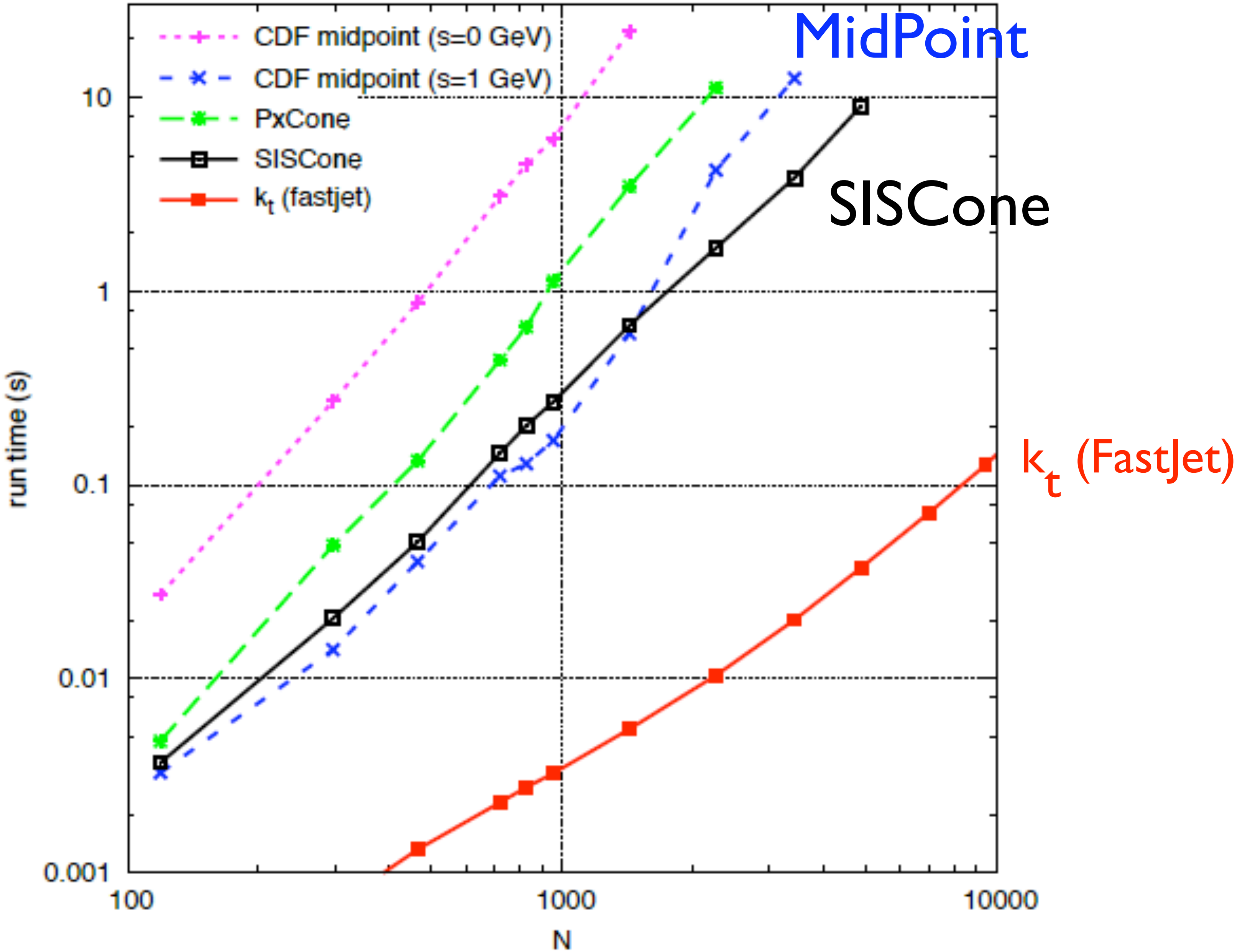
FastJet performance (k_t)

Time taken to cluster N particles:



- Almost two orders of magnitude gain at small N (related $O(N^2)$ implementation)
- Large-N region now reachable

SISCone performance



SISCone as fast as MidPoint → no penalty for infrared safety!

Cones Infrared (un)safety

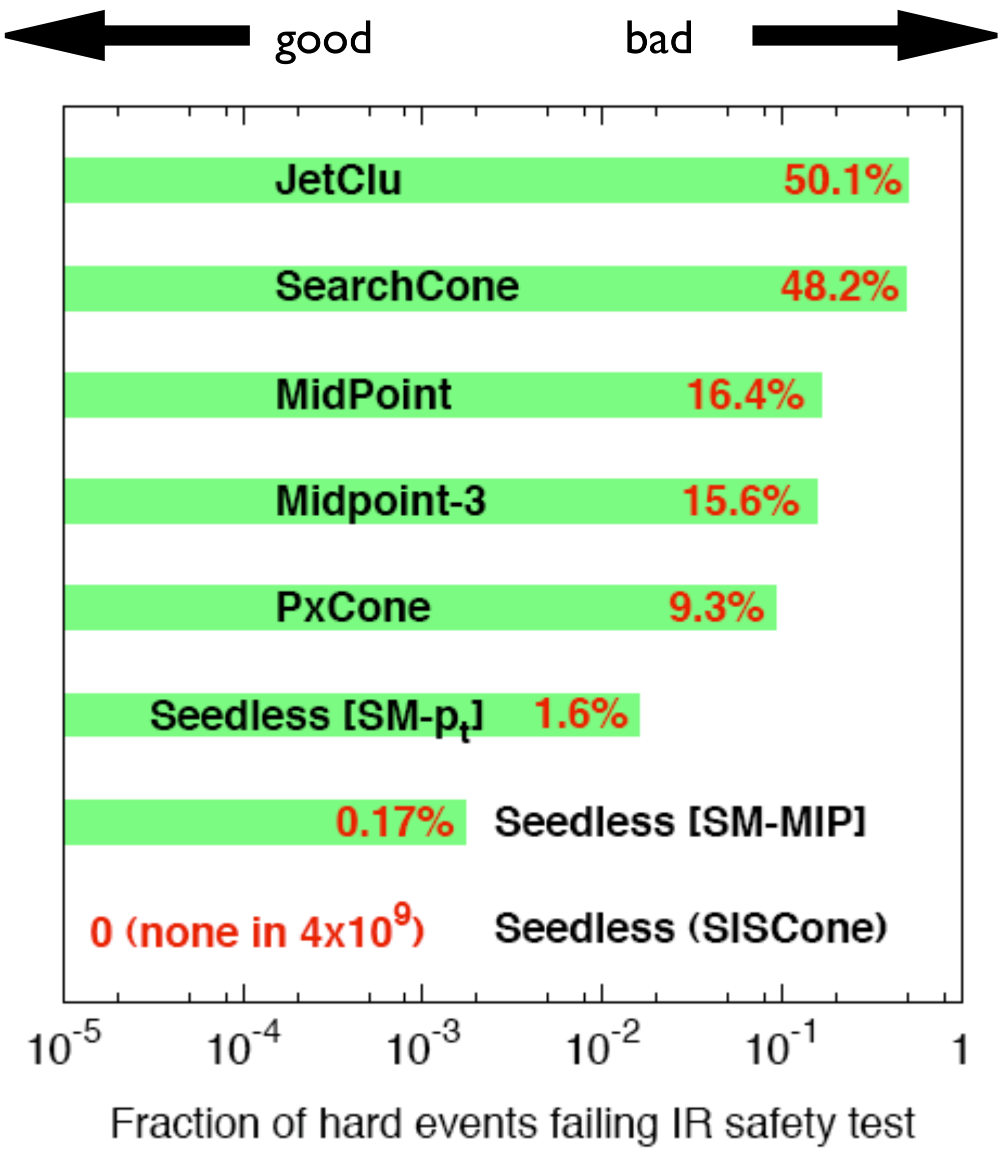
Q: How often are the hard jets changed by the addition of a soft particle?

A:

- Generate event with $2 < N < 10$ hard particles, find jets
- Add $1 < N_{soft} < 5$ soft particles, find jets again [repeatedly]
- If the jets are different, algorithm is IR unsafe.

Unsafety level	failure rate
2 hard + 1 soft	$\sim 50\%$
3 hard + 1 soft	$\sim 15\%$
SISCone	IR safe !

Be careful with split–merge too



Salam & Soyez

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	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 $d_{ij} = \min(k_{ti}^{-2}, k_{tj}^{-2})\Delta R_{ij}^2/R^2$ 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)

Replacements

If you care about IRC safety but don't want to stray too far from algorithms used so far, these are possible replacements:

JetClu ATLAS Cone MidPoint	⟶	SISCone	(As fast, but IRC safe)
Iterative Cone (PR)	⟶	anti-k_t	(Gives regular cones too, but IRC safe)

In addition, k_t and Cambridge/Aachen can provide further flexibility

Different algorithms spanning a series of *different and complementary characteristics*: should be enough for most purposes

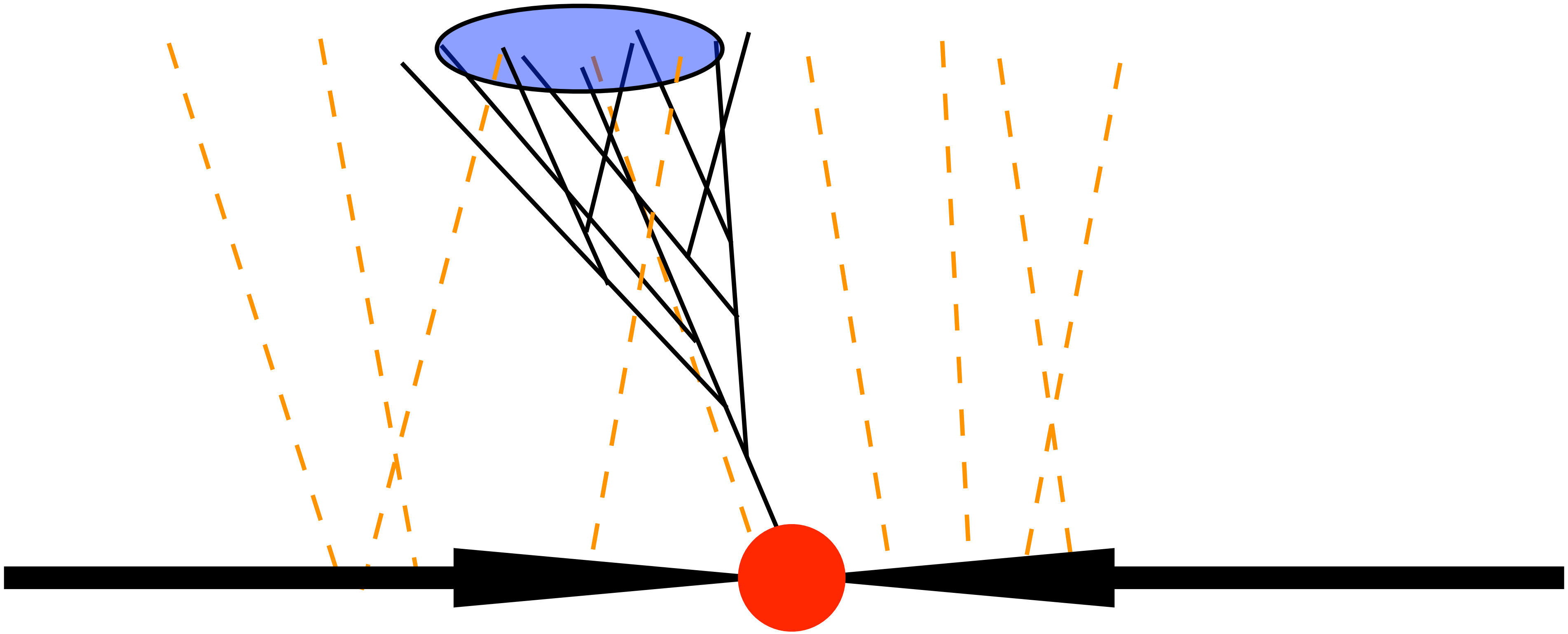
One should probably try to concentrate on these, both for analytical understanding and practical use in experiments, rather than using IRC unsafe ones

So far, old jet clustering, just better and/or faster

High speed and infrared safety allow for a **qualitatively new use** of jet clustering, through **new features: jet areas**

It's an example of **making a different use of jets:**

Jet areas: the physics case



In a realistic set-up underlying event (UE) and pile-up (PU) from multiple collisions produce many soft particles which can 'contaminate' the hard jet

p_T (jet) ~ p_T (parton) + Average underlying momentum density × 'size' of the jet

Not one, but three **definitions** of a jet's size:

MC, Salam, Soyez, arXiv:0802.1188

- Voronoi area
- Passive area — Mimics effect of **pointlike** radiation
- Active area — Mimics effect of **diffuse** radiation

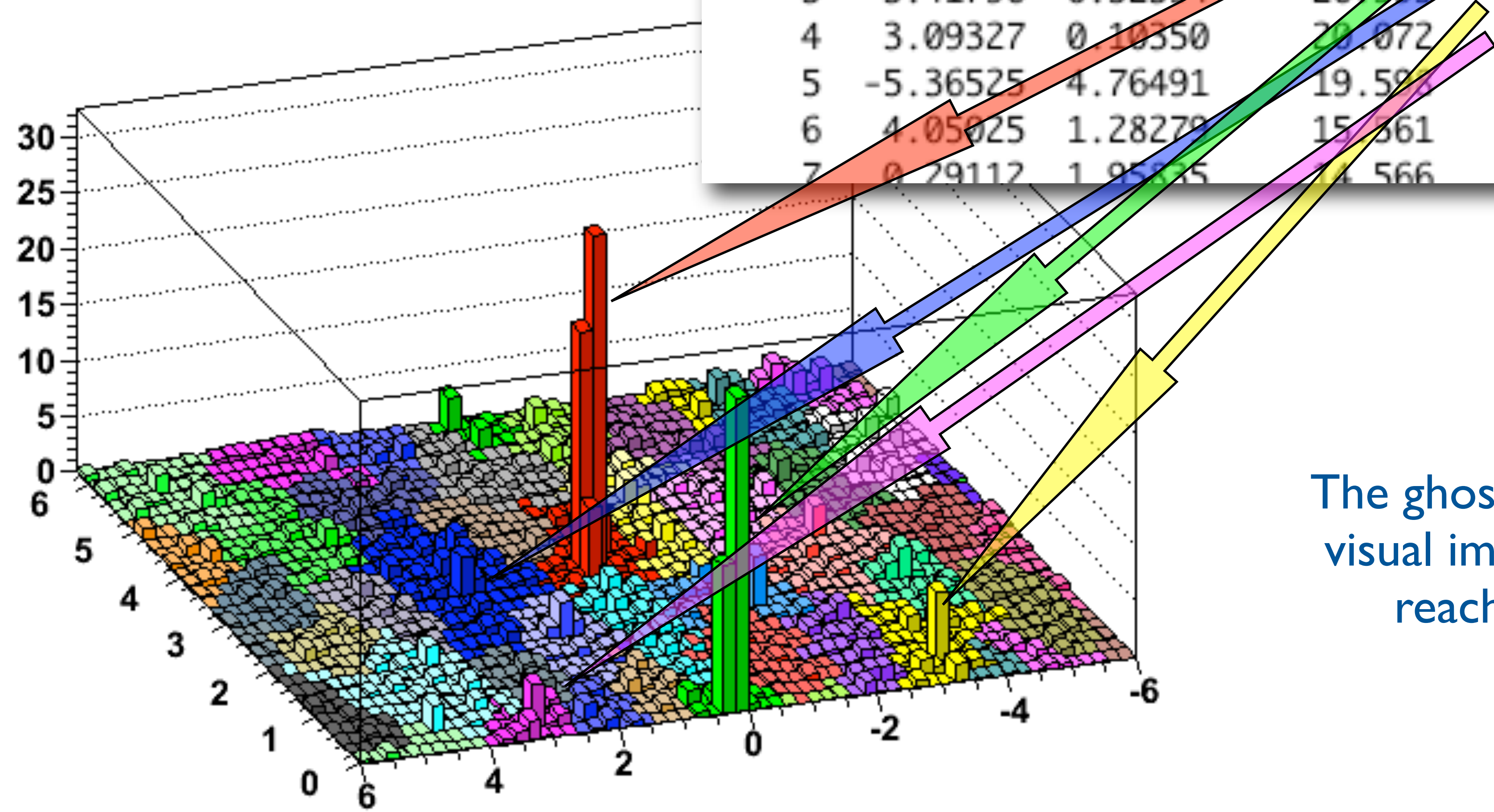
(In the large number of particles limit all areas converge to the same value)

Details

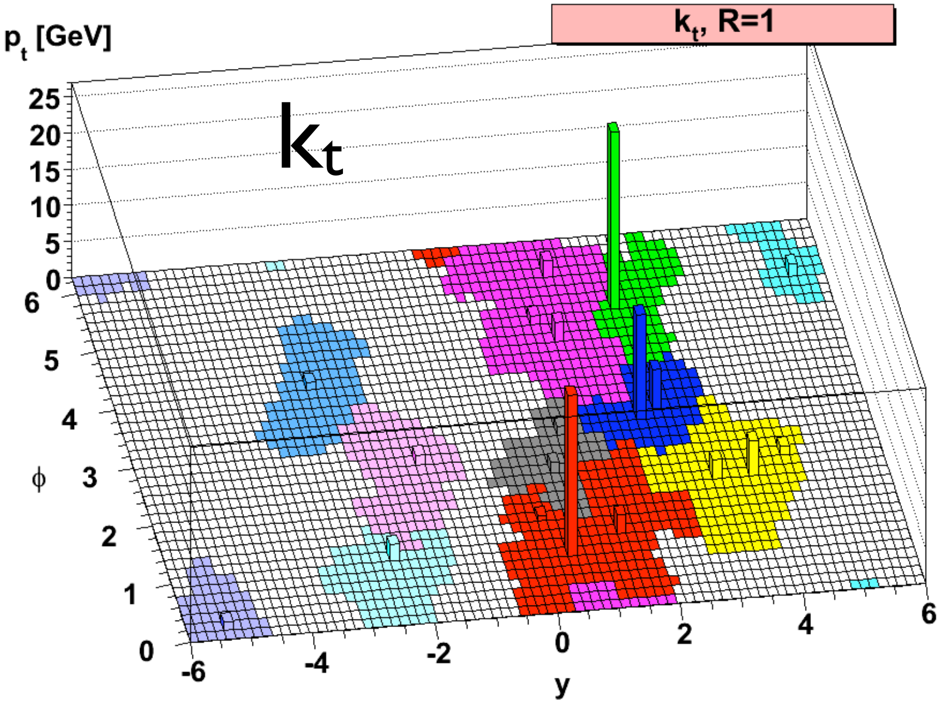
Jet active areas

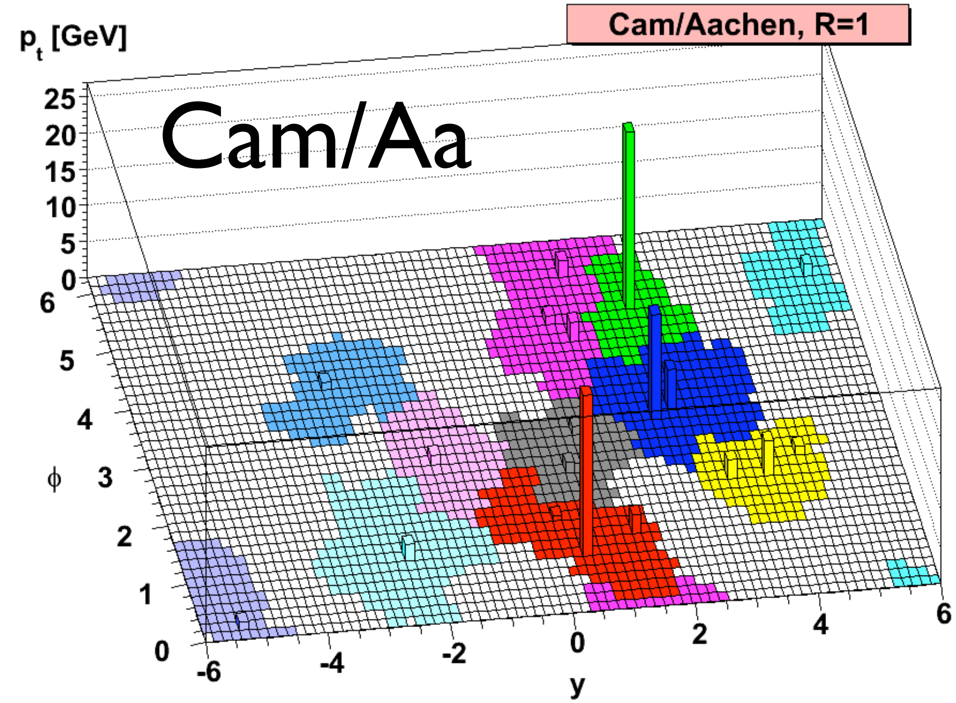
Active areas are calculated by adding thousands of `ghost' particles, clustering them with the event, and counting how many end up in a given jet

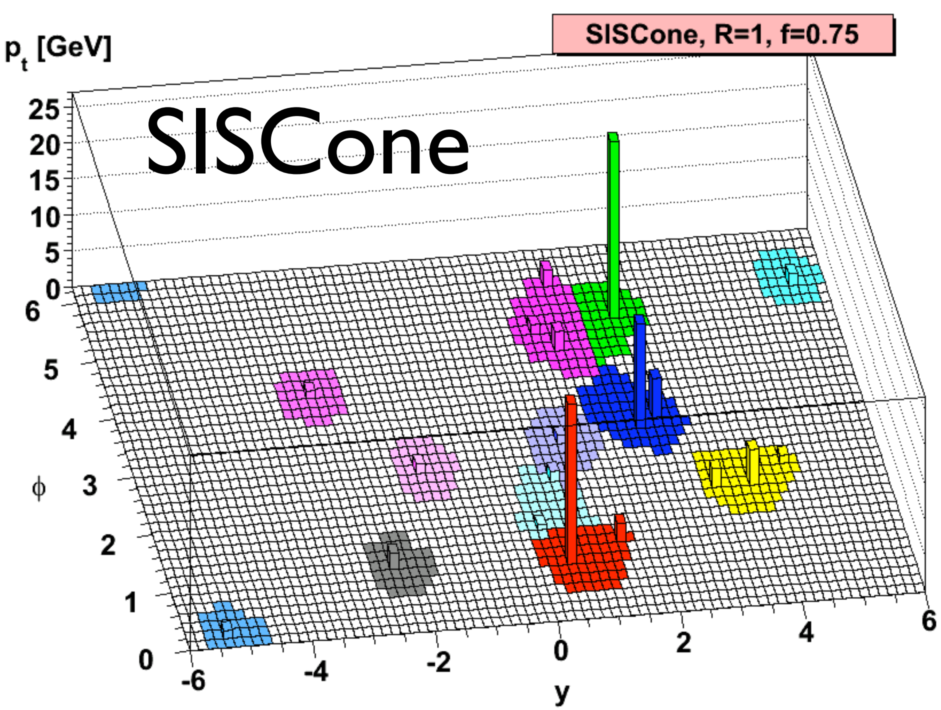
```
iev 0 (irepeat 24): number of particles = 1428
strategy used =  NlnN
number of particles = 9051
Total area: 76.0265
Expected area: 76.0265
 ijet    eta        phi         Pt          area  +-   err
   0    0.15050   3.24498     69.970      2.625 +-   0.020
   1    0.18579   0.13150     59.133      1.896 +-   0.020
   2    2.33840   3.23960     31.976      4.749 +-   0.028
   3   -3.41796   0.52394     26.595      3.084 +-   0.021
   4    3.09327   0.10350     20.072      2.688 +-   0.023
   5   -5.36525   4.76491     19.598      2.780 +-   0.012
   6    4.05025   1.28279     15.561      3.592 +-   0.028
   7    0.29112   1.95835     14.566      2.114 +-   0.018
```

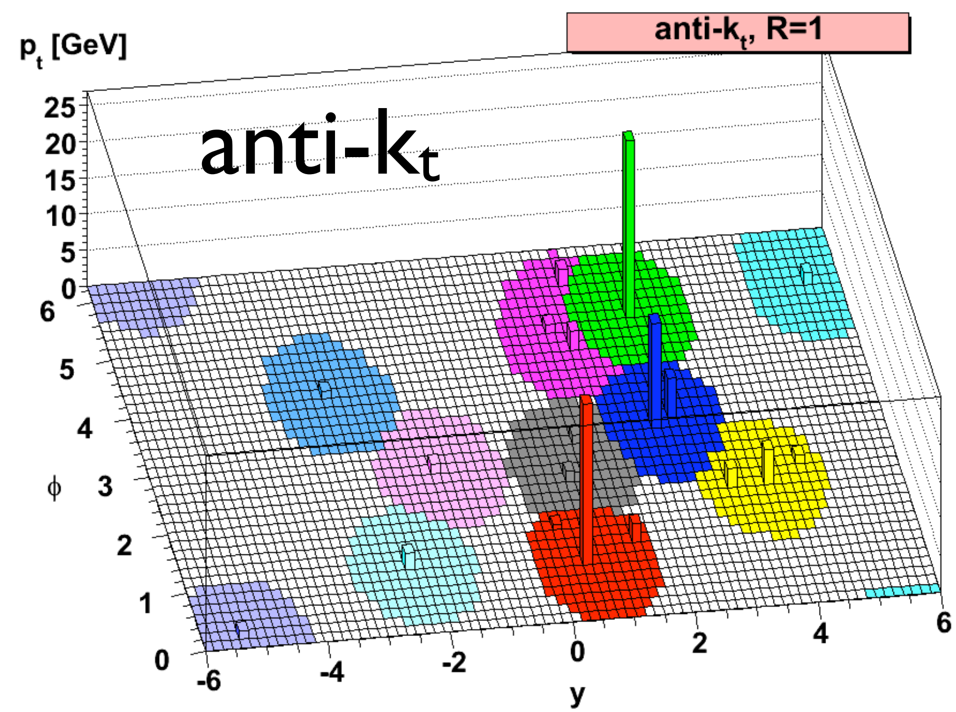


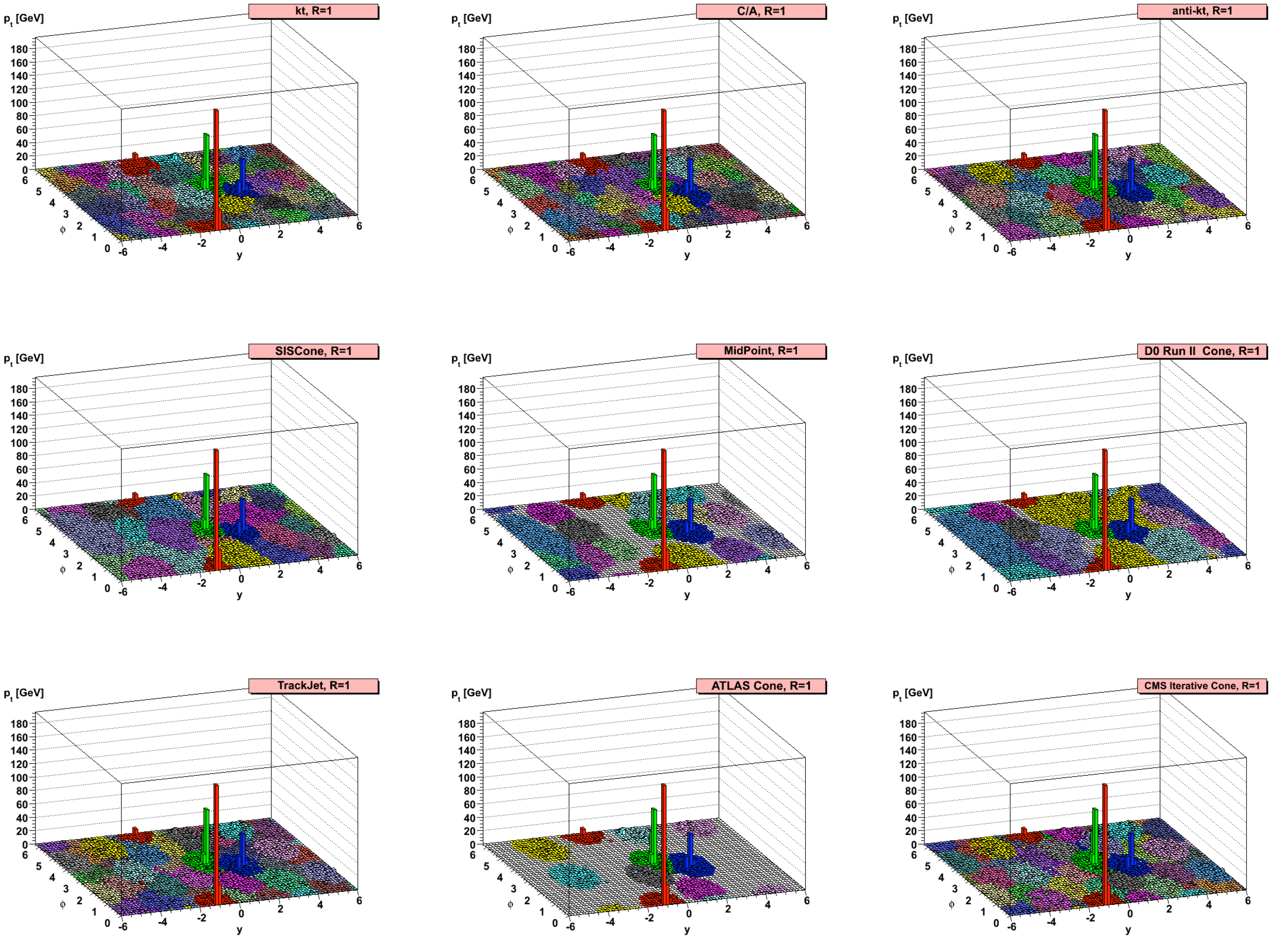
The ghost can also give a visual impression of the reach of each jet


k$_t$, R=1
k$_t$
p$_t$ [GeV]
φ
y


Cam/Aachen, R=1
Cam/Aa
p$_t$ [GeV]
φ
y


SISCone, R=1, f=0.75
SISCone
p$_t$ [GeV]
φ
y


anti-k$_t$, R=1
anti-k$_t$
p$_t$ [GeV]
φ
y


kt, R=1
C/A, R=1
anti-kt, R=1
SISCone, R=1
MidPoint, R=1
D0 Run II Cone, R=1
TrackJet, R=1
ATLAS Cone, R=1
CMS Iterative Cone, R=1
p_t [GeV]
ϕ
y

Jet areas: the link to physics

The definition of **active area** mimics the behaviour of the jet-clustering algorithms in the presence of a **large number of randomly distributed soft particles**

This is like underlying event or pileup.

Tools needed to implement it:

1. An **infrared safe jet algorithm** (the ghosts should not change the jets)
2. A reasonably **fast implementation** (we are adding thousands of ghosts)

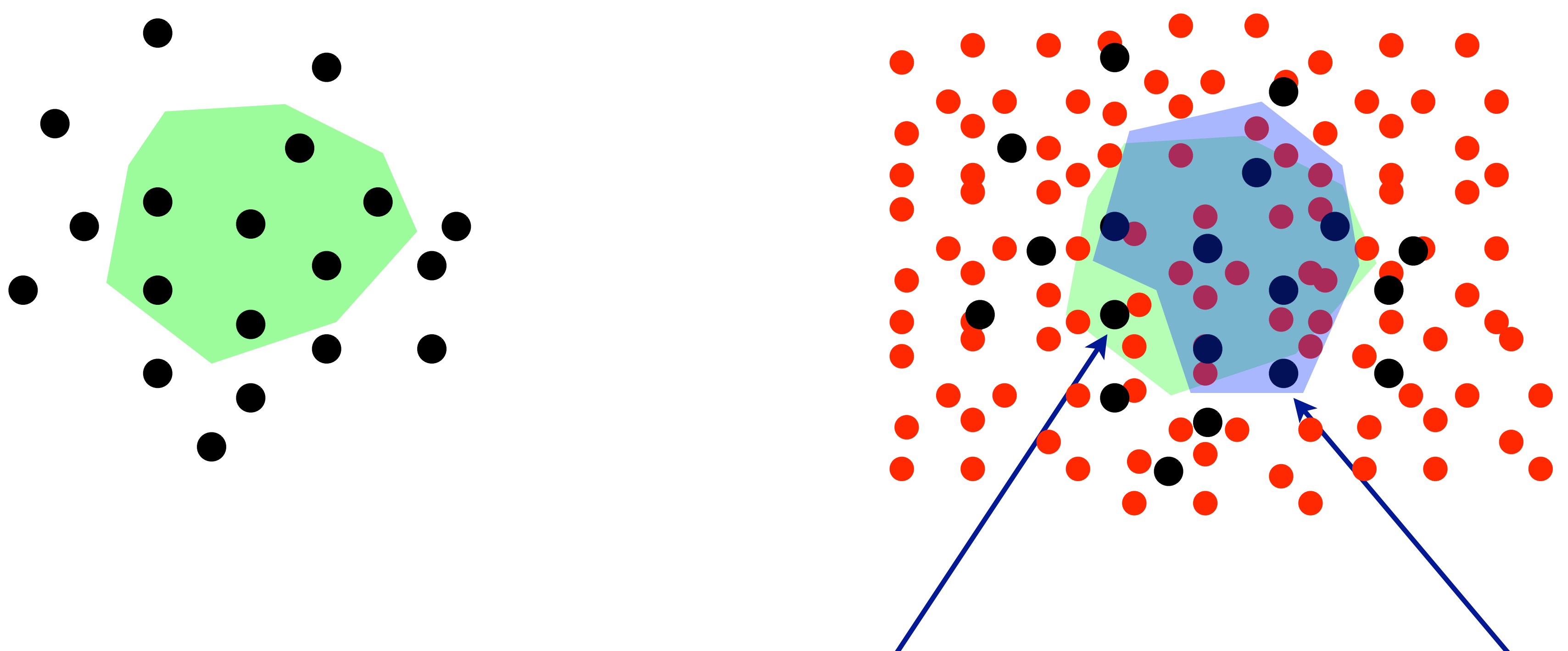
Both are available

Backreaction

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

Without background

With background

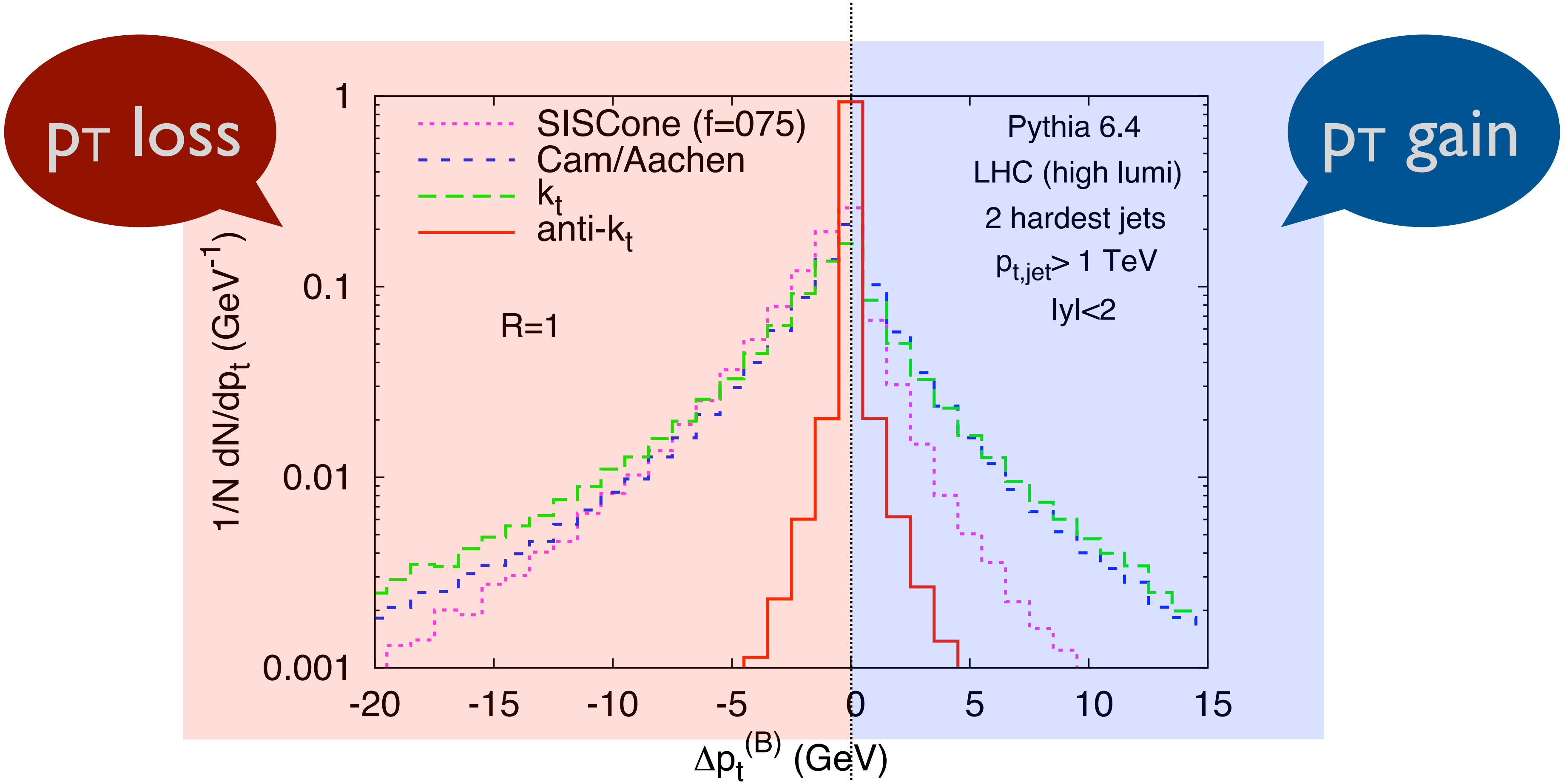


Backreaction **loss**

Backreaction **gain**

Backreaction

MC, Salam, Soyez, arXiv:0802.1188



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

Underlying event and pileup determination and subtraction

Measurement of background level

MC, Salam, arXiv:0707.1378

$$\rho \equiv \text{median}\left[\left\{\frac{p_t^{jet}}{\text{Area}_{jet}}\right\}\right]$$

(over a single event)

Taking the median of the distribution is a way to get rid of (part of the) possible bias from the few hard jets

(This can still be tweaked/improved)

For ρ to be non-zero at the *perturbative* level one would need at least as many hard jets as 'empty jets'.
This can be shown to happen at order α_s^n, with

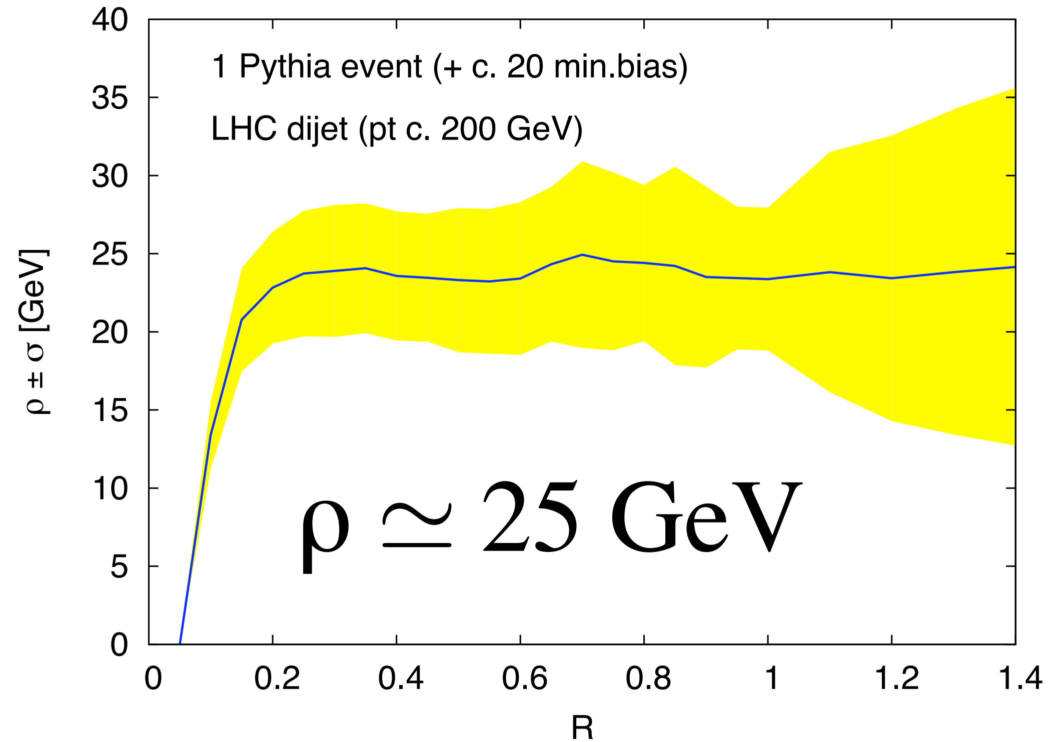
$$n \simeq \frac{4y_{\max}}{(x_{\rm pg}+x_{\rm sp})R^2} \simeq 2.94\frac{y_{\max}}{R^2}$$

Factors related to typical areas of pure-ghost and single-particle jets

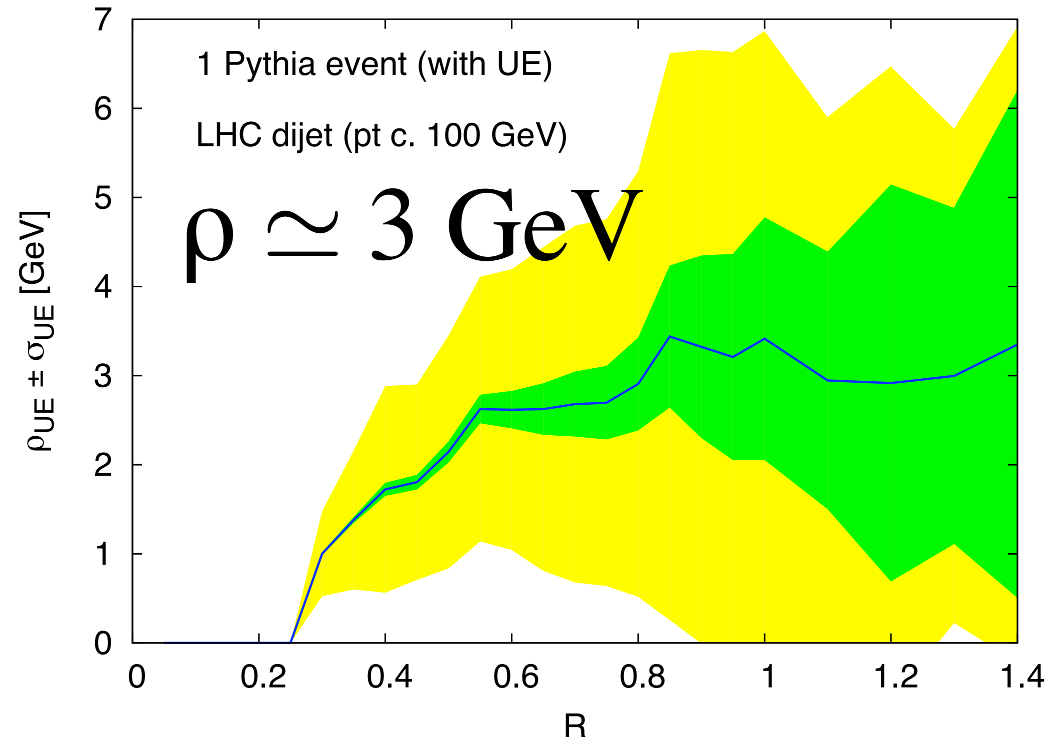
This gives n ~ 24--47 for y_{max} = 4 and R = 0.5--0.7

Examples of background measurements

Pileup @ LHC



UE @ LHC



When 'measuring' the background, R should be not too small (too many empty jets) and not too large (too few jets, biased by the hard particles)

Theoretical estimates and empirical evidence point to R ~ 0.5--0.6 for Underlying Event measurement

(Also, a 'sensible' jet alg like k_t or Cambridge/Aachen should be used)

Background subtraction

[MC, Salam, arXiv:0707.1378]

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

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

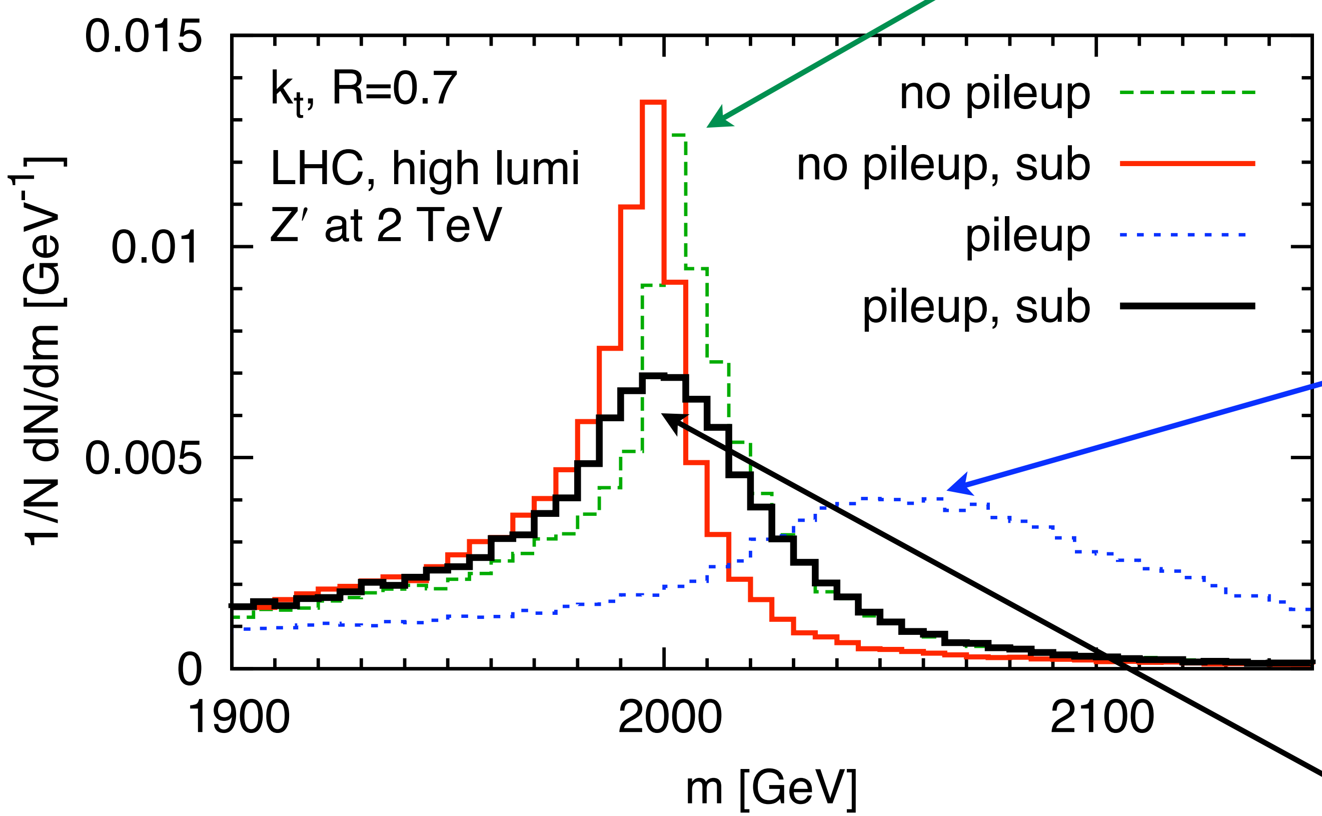
ρ being calculated on an event-by-event basis, this procedure will generally improve the resolution of, say, a mass peak

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

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

Which R to choose?

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

R-dependent effects

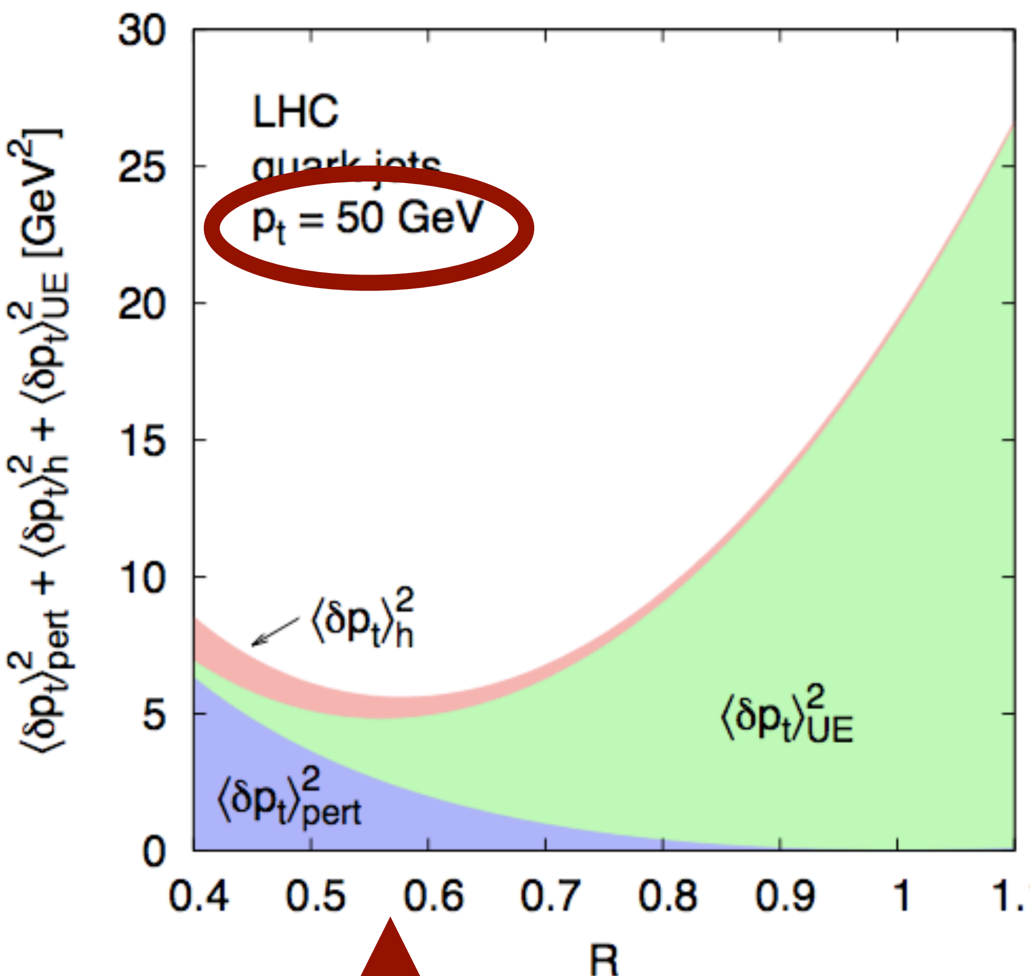
Perturbative radiation: $$\Delta p_t \simeq \frac{\alpha_s(C_F, C_A)}{\pi} p_t \ln R$$

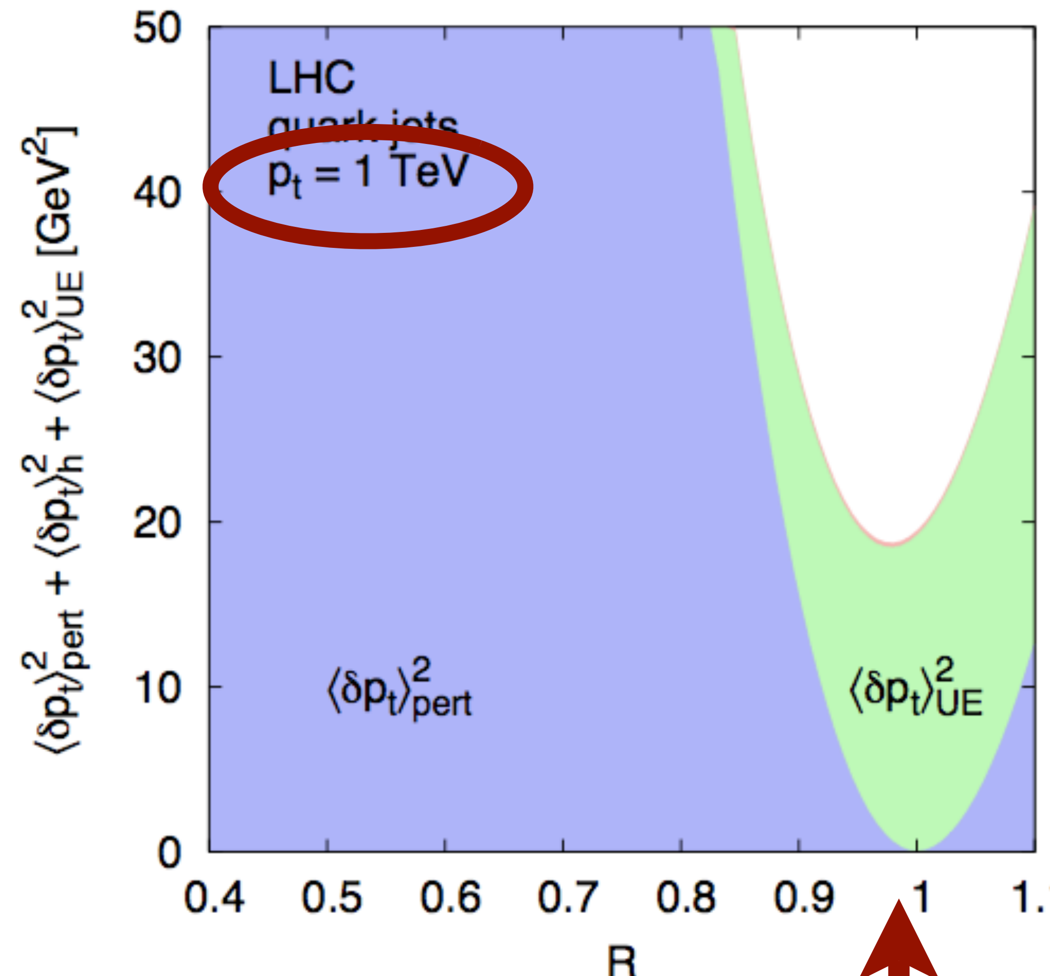
Hadronisation: $$\Delta p_t \simeq \frac{(C_F, C_A)}{R} \times 0.4 \text{ GeV}$$

Underlying Event: $$\Delta p_t \simeq \frac{R^2}{2} \times (\underset{\text{Tevatron}}{2.5} \text{ — } \underset{\text{LHC}}{15} \text{ GeV})$$

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

Minimize $\Sigma(\Delta p_t)^2$





Best R

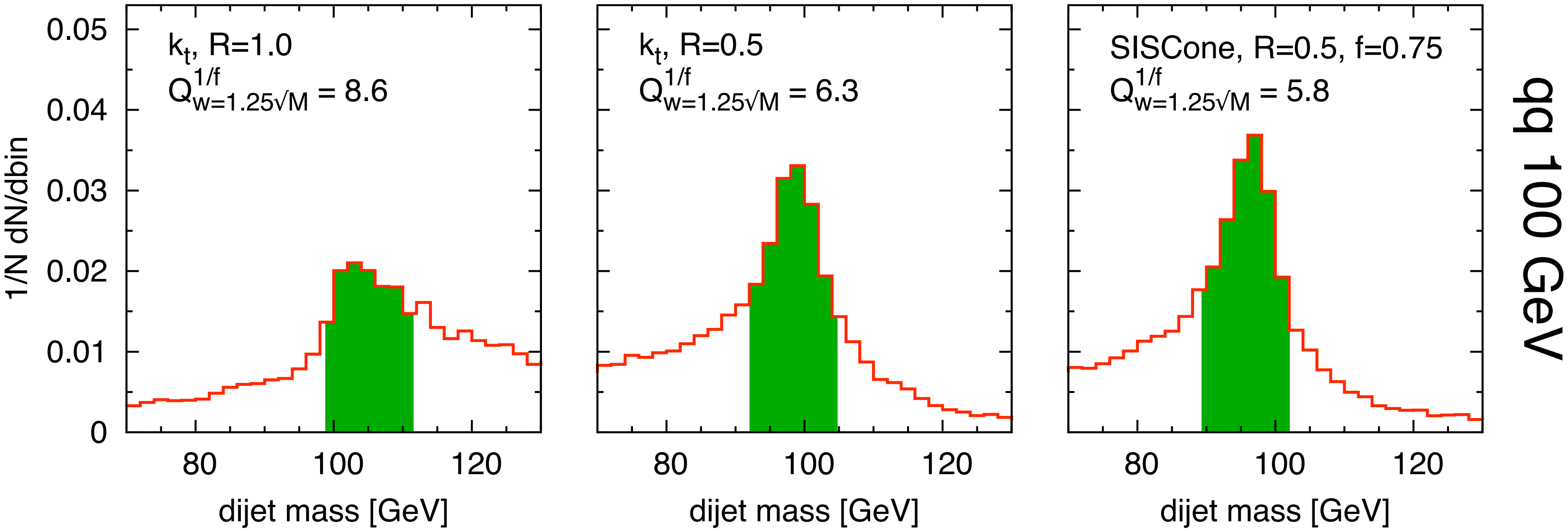
Best R

Dasgupta, Magnea, Salam, arXiv:0712.3014

Reconstruction of a di-jet mass peak

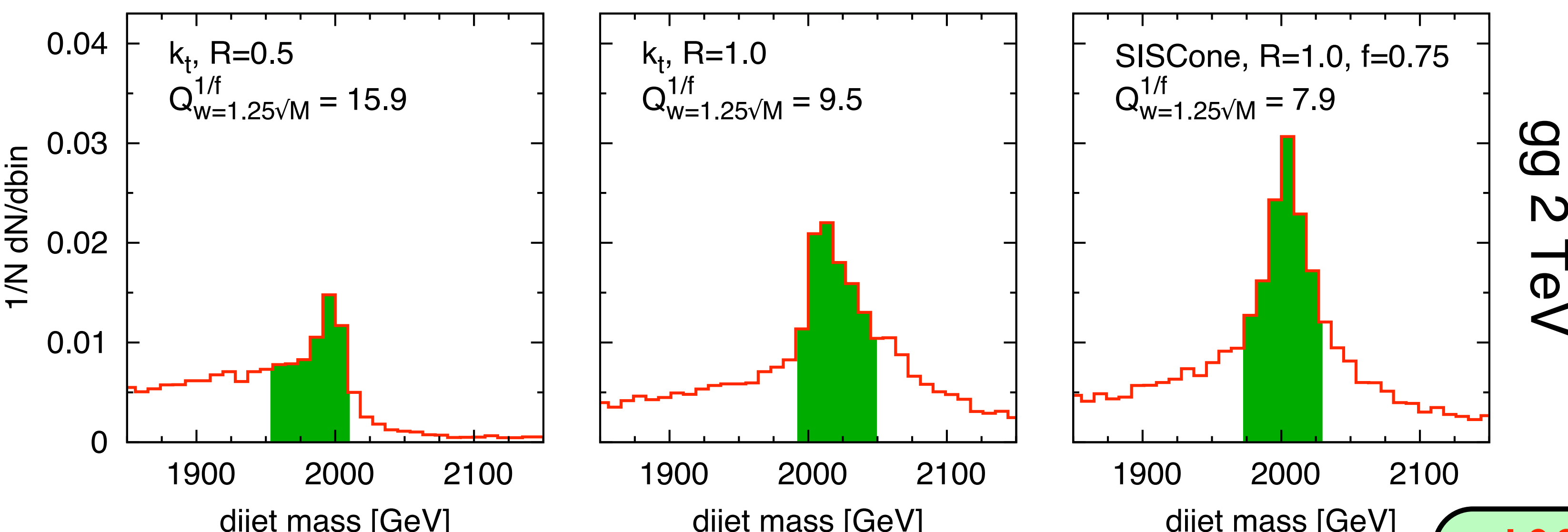
MC, Rojo, Salam, Soyez, arXiv:0810.1304

R=1: BAD   R=0.5: BETTER



qq jets at 100 GeV

R=0.5: BAD   R=1: BETTER

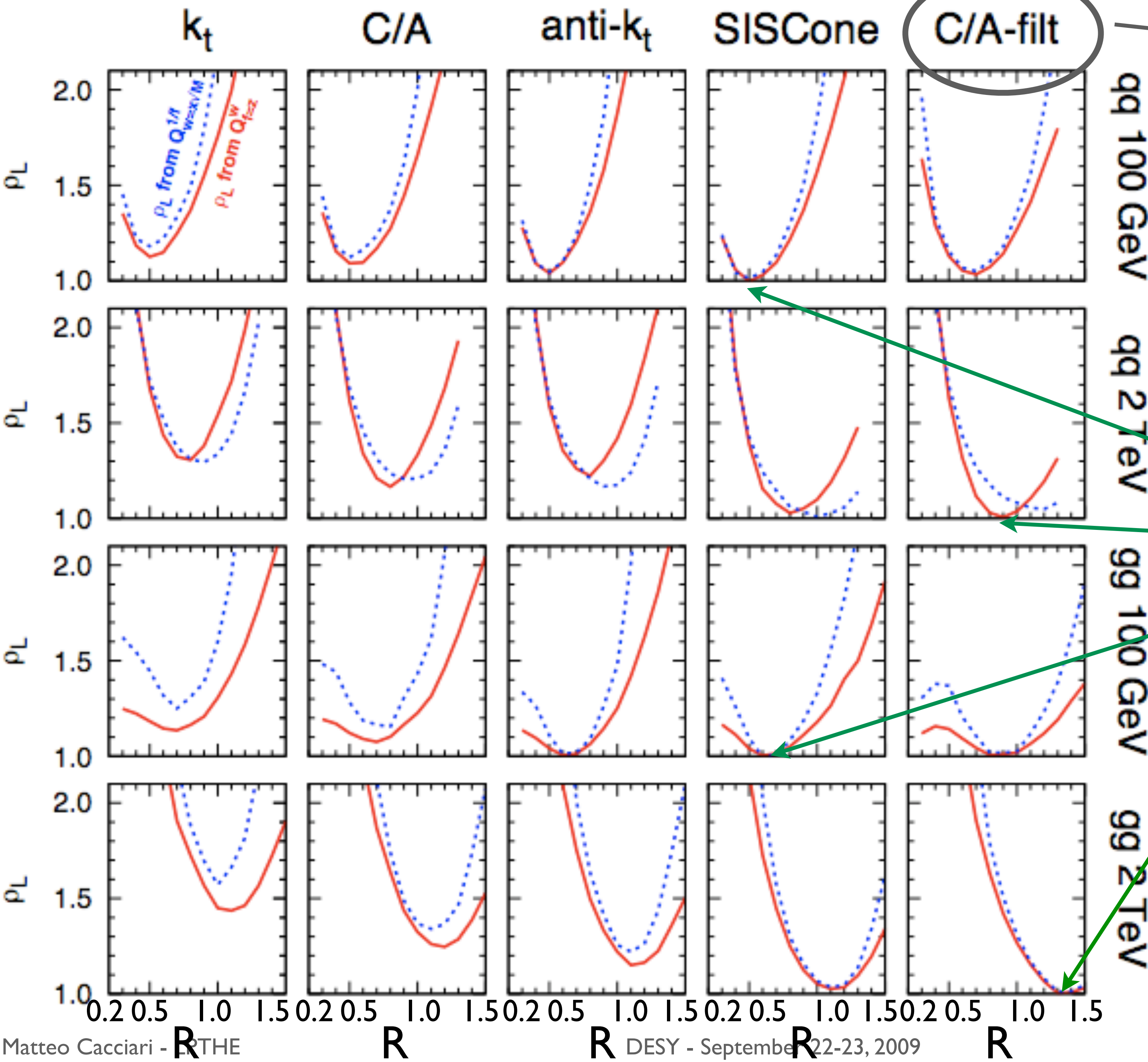


gg jets at 2 TeV

~ 100000 similar plots at http://quality.fastjet.fr

Gluons (and heavy objects) prefer larger R

Effective luminosity ratios



Example of a 'third-generation' algorithm

Lower is better

Best

Different algs, different R, different performances

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 $x_{filt}R$
- Retain only the n_{filt} hardest subjets

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

Jets substructure in Higgs searches

$H \to b\bar{b}$ in the WH/ZH channels

Usually considered hopeless:

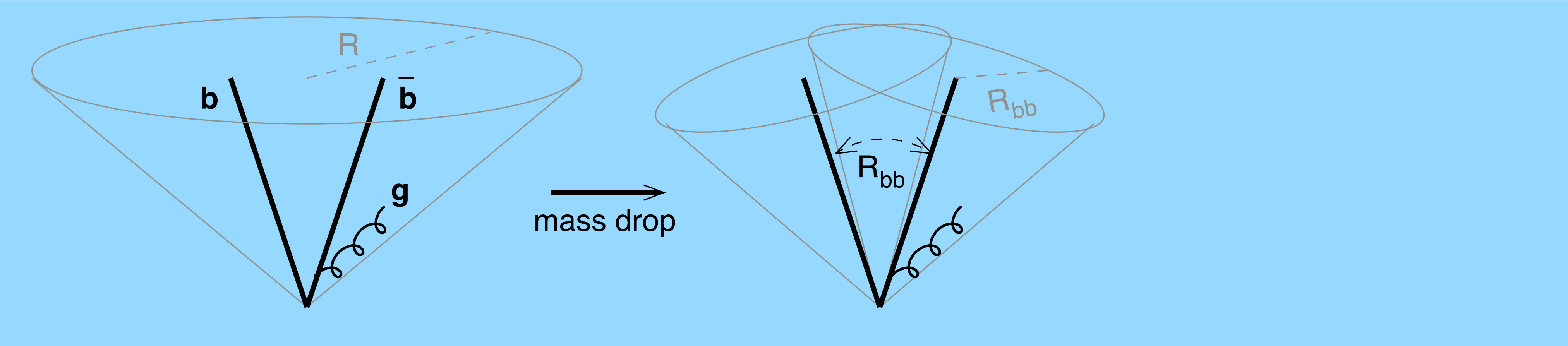
Conclusion (ATLAS TDR):

"The extraction of a signal from $H \to b\bar{b}$ decays in the WH channel will be very difficult at the LHC, even under the most optimistic assumptions [...]"

New way of going at it:
use large p_t (boosted Higgs), exploit jet substructure

Butterworth, Davison, Rubin, Salam, arXiv:0802.2470

#2: The jet analysis



Start with high-p_t jet

1. Undo last stage of clustering ($\equiv$ reduce R): $J \to J_1, J_2$

2. If $\max(m_1, m_2) \lesssim 0.67m$, call this a **mass drop** [else goto 1]

   Automatically detects correct $R \sim R_{bb}$ to catch angular-ordered radn.

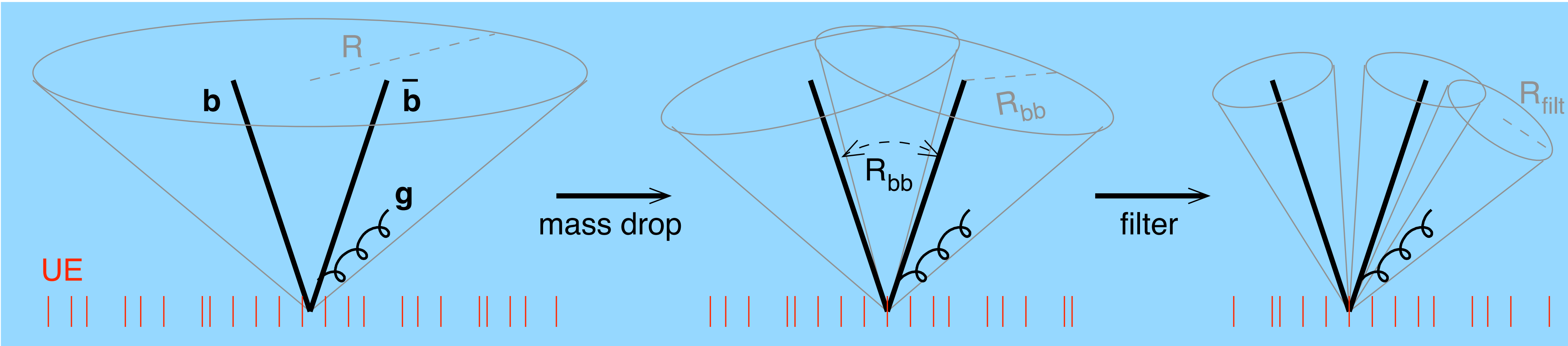
3. Require $y_{12} = \frac{\min(p_{t1}^2, p_{t2}^2)}{m_{12}^2} \Delta R_{12}^2 \simeq \frac{\min(z_1, z_2)}{\max(z_1, z_2)} > 0.09$ [else goto 1]

   dimensionless rejection of asymmetric QCD branching

4. Require each subjet to have b-tag [else reject event]

   Correlate flavour & momentum structure

#3: jet filtering



At moderate p_t, R_{bb} is quite large; *UE & pileup degrade mass resolution*

$\delta M \sim R^4 \Lambda_{UE} \frac{p_t}{M}$ [Dasgupta, Magnea & GPS '07]

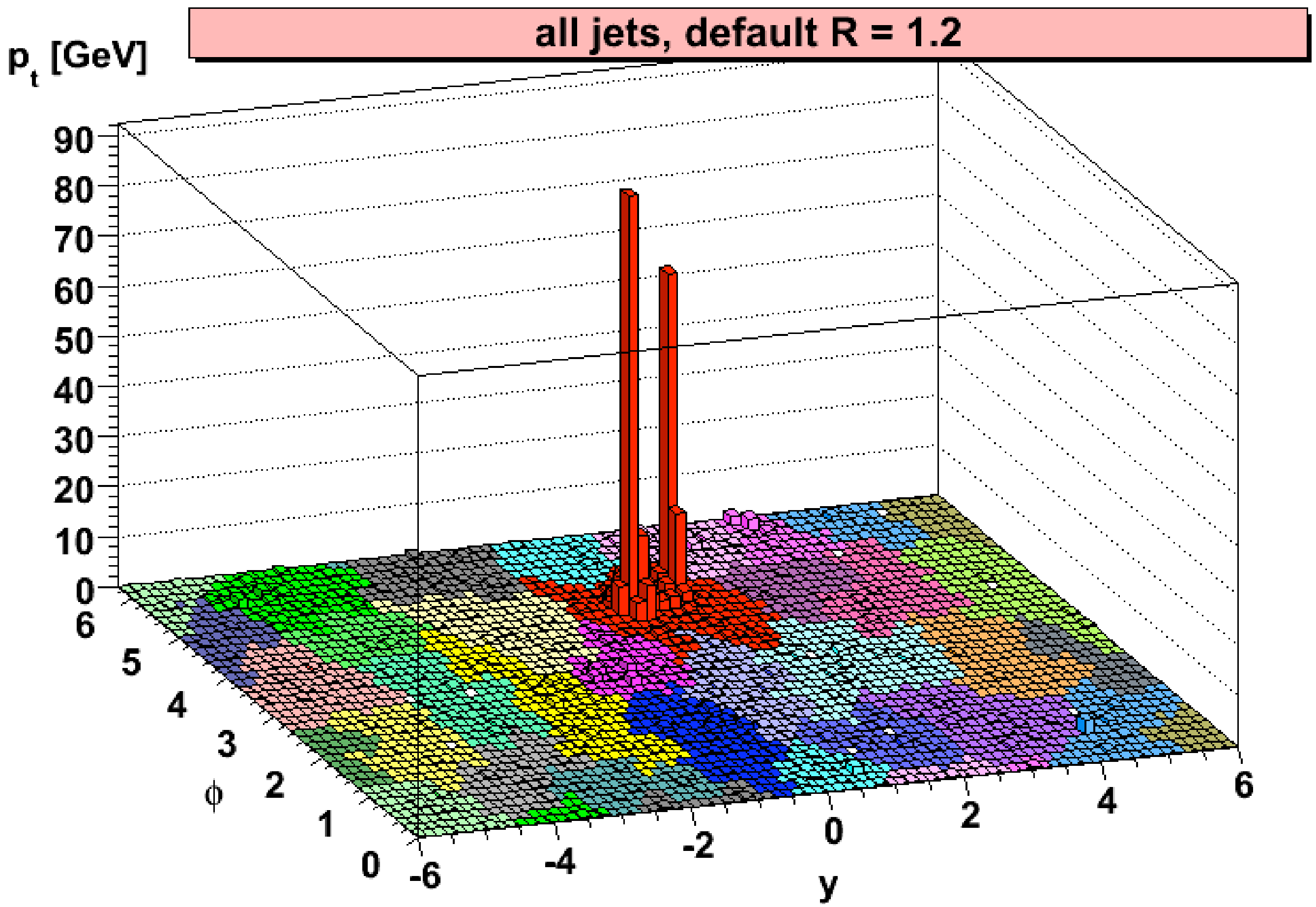
Filter the jet

- Reconsider region of interest at smaller $R_{filt} = \min(0.3, R_{b\bar{b}}/2)$
- Take **3** hardest subjets — b, $\bar{b}$ and leading order gluon radiation

$pp \rightarrow ZH \rightarrow \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV

SIGNAL

Herwig 6.510 + Jimmy 4.31 + FastJet 2.3



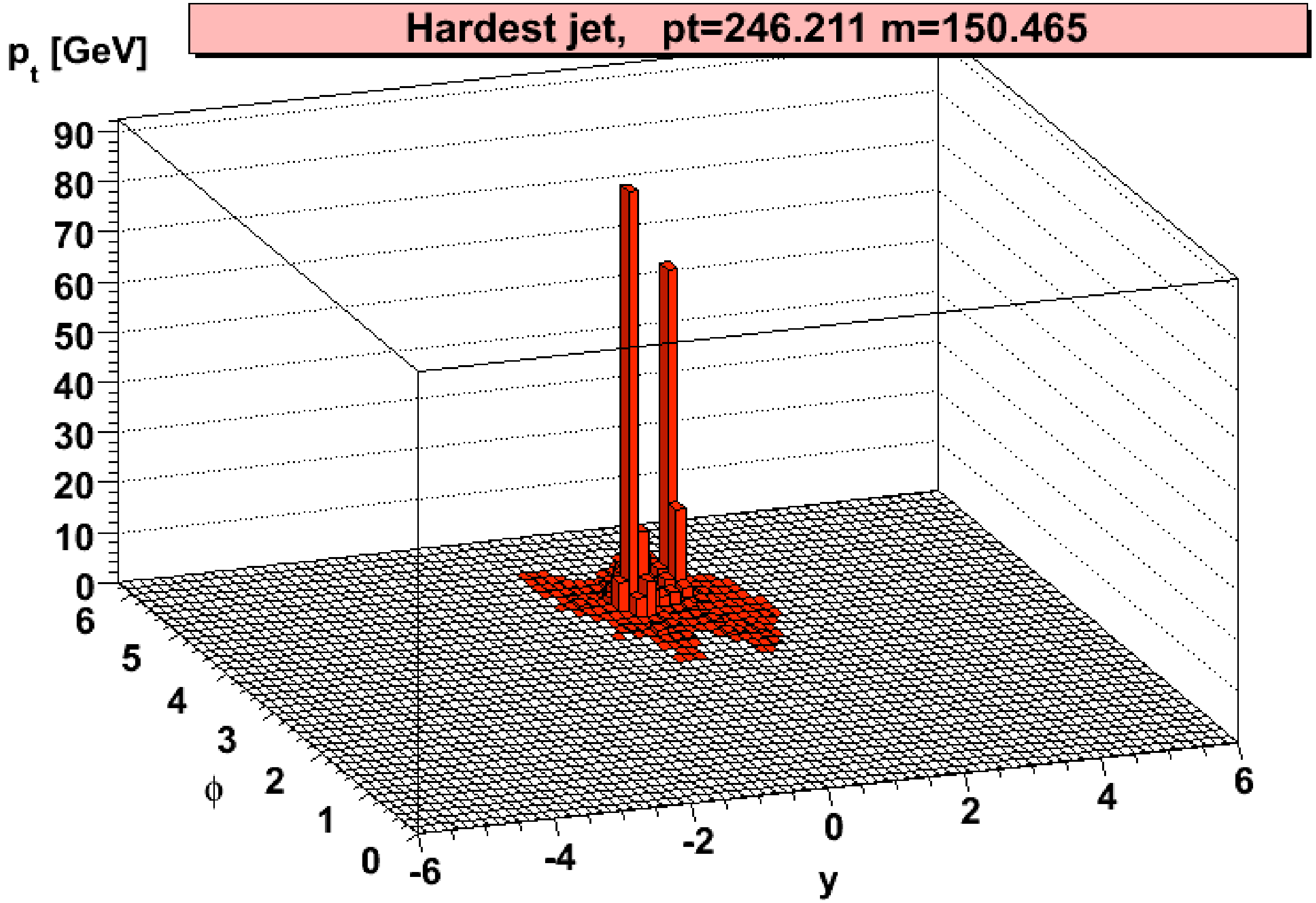
Zbb BACKGROUND

Cambridge/Aachen, R=1.2

arbitrary norm.

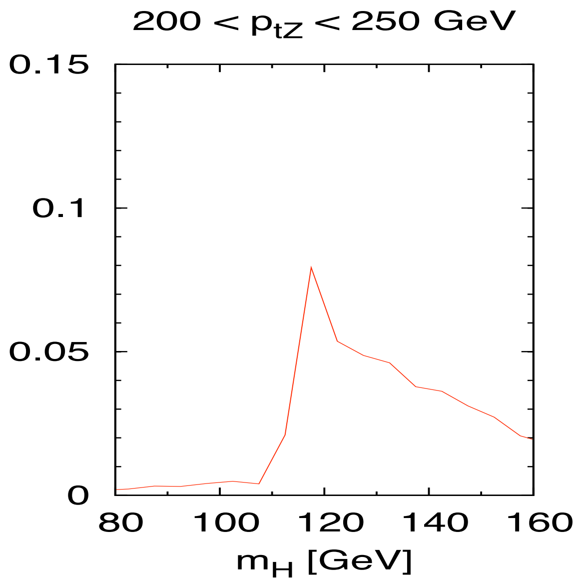
$pp \to ZH \to \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV

Herwig 6.510 + Jimmy 4.31 + FastJet 2.3

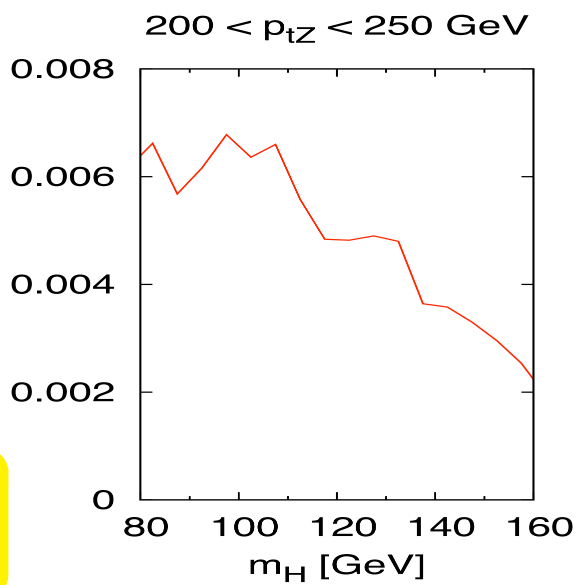


Consider hardest jet, $m = 150$ GeV

SIGNAL



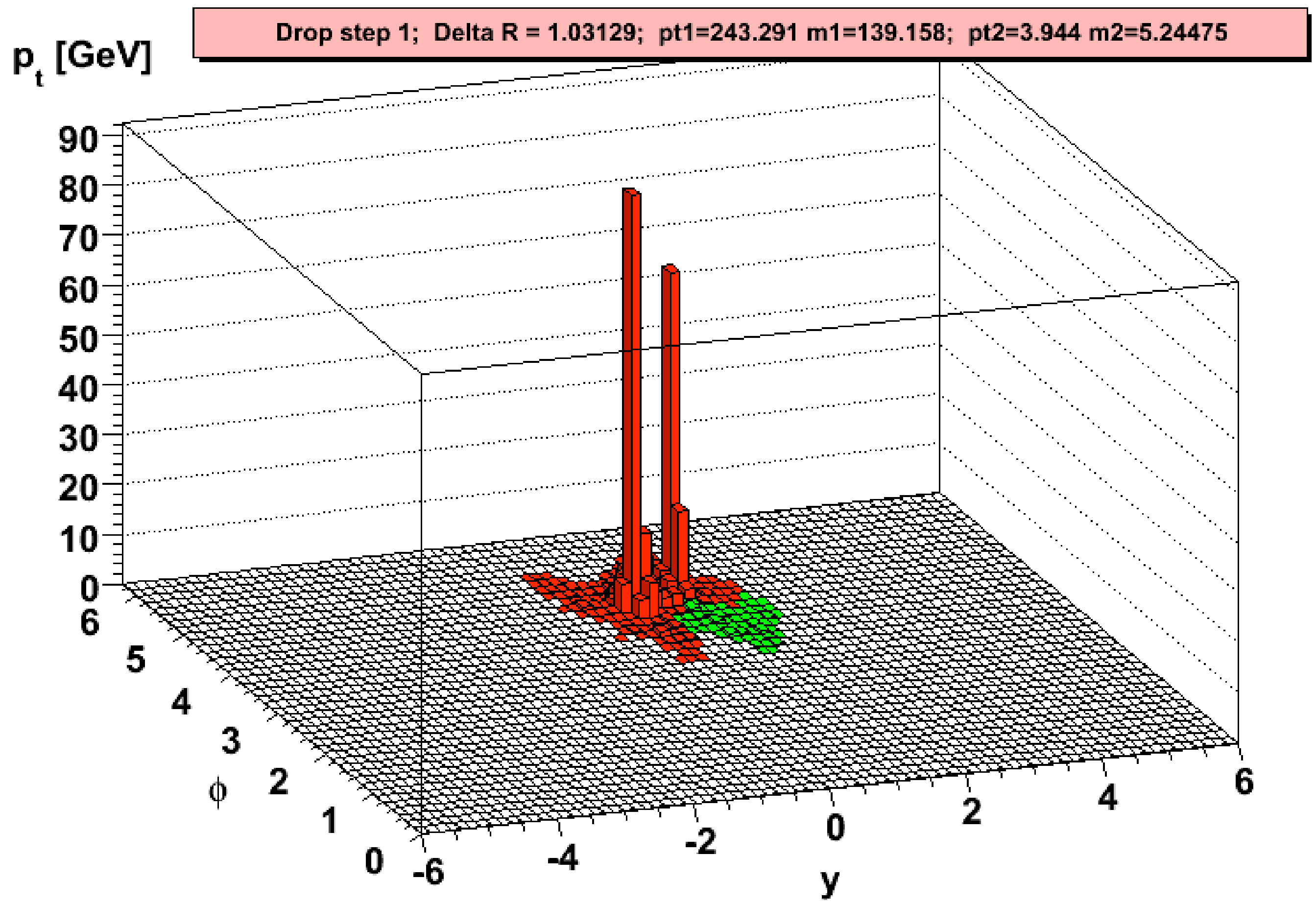
Zbb BACKGROUND



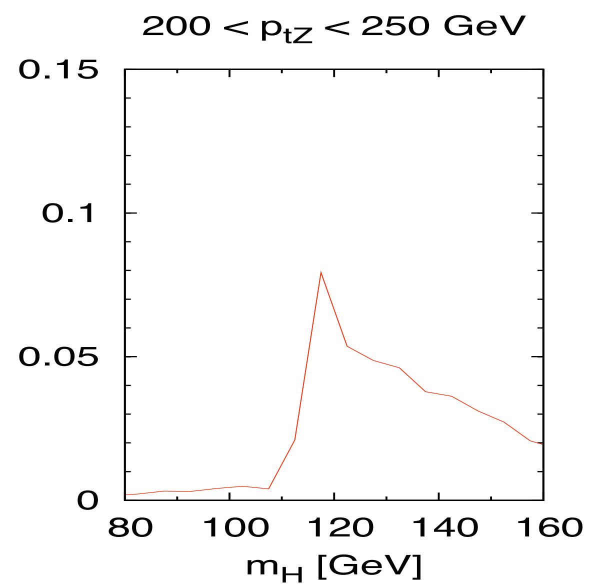
arbitrary norm.

$pp \to ZH \to \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV

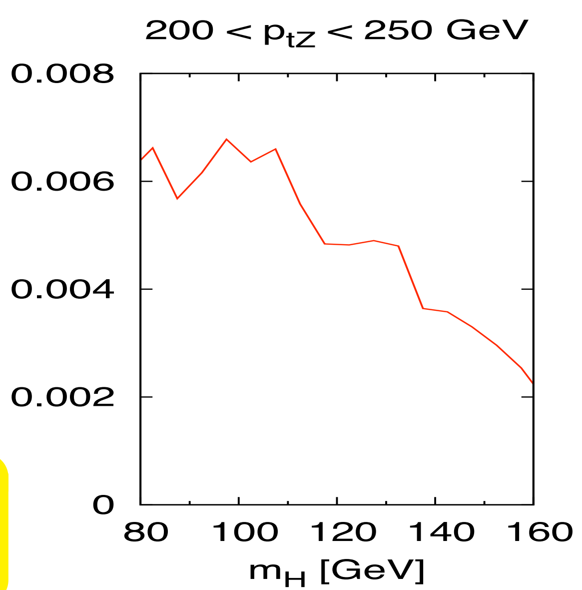
Herwig 6.510 + Jimmy 4.31 + FastJet 2.3



SIGNAL



Zbb BACKGROUND

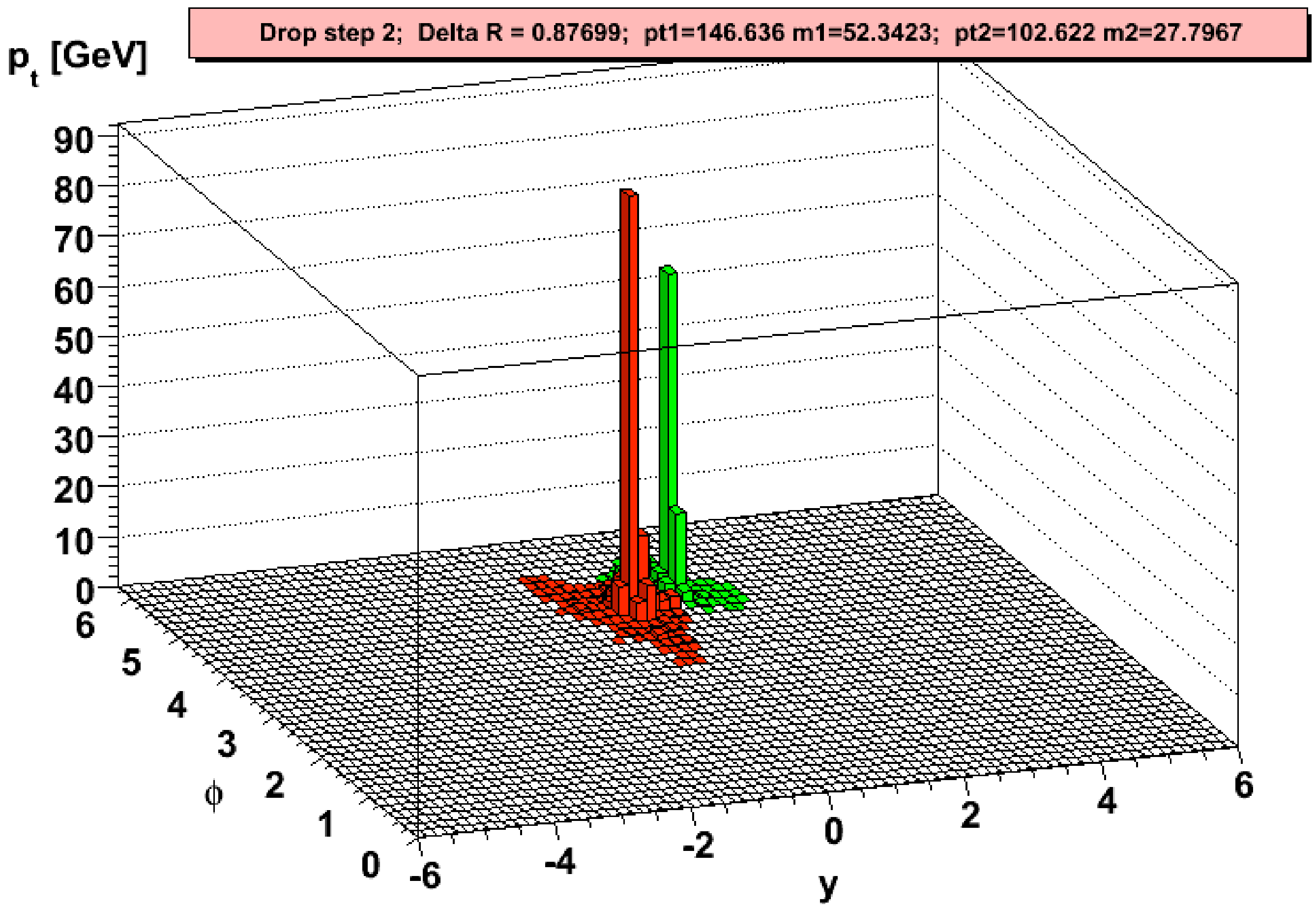


split: $m = 150$ GeV, $\dfrac{\max(m_1, m2)}{m} = 0.92 \to$ repeat

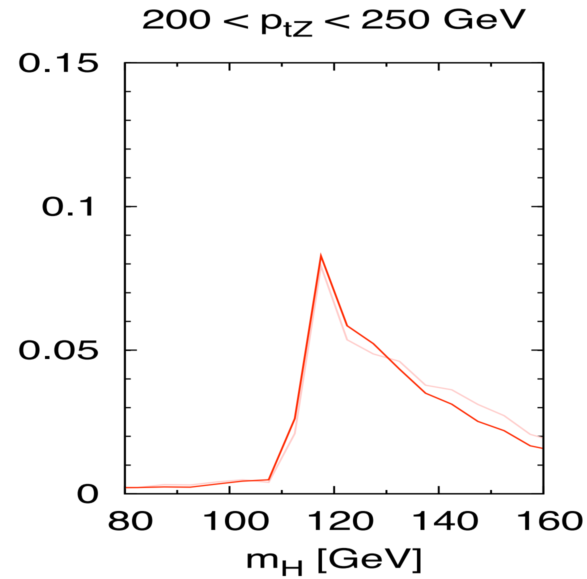
arbitrary norm.

$pp \to ZH \to \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV

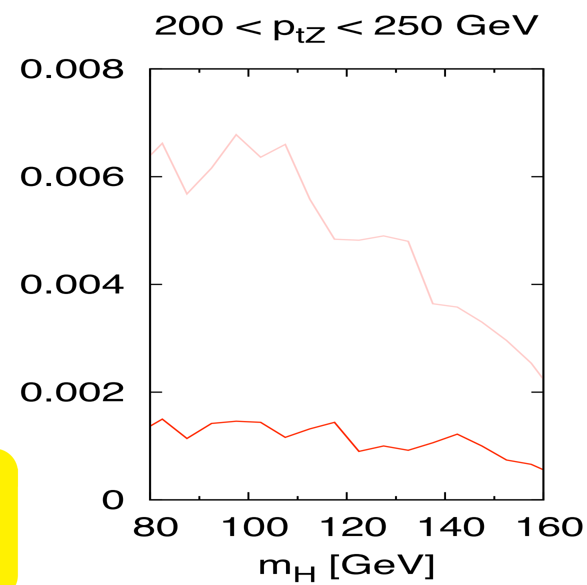
Herwig 6.510 + Jimmy 4.31 + FastJet 2.3



SIGNAL



Zbb BACKGROUND



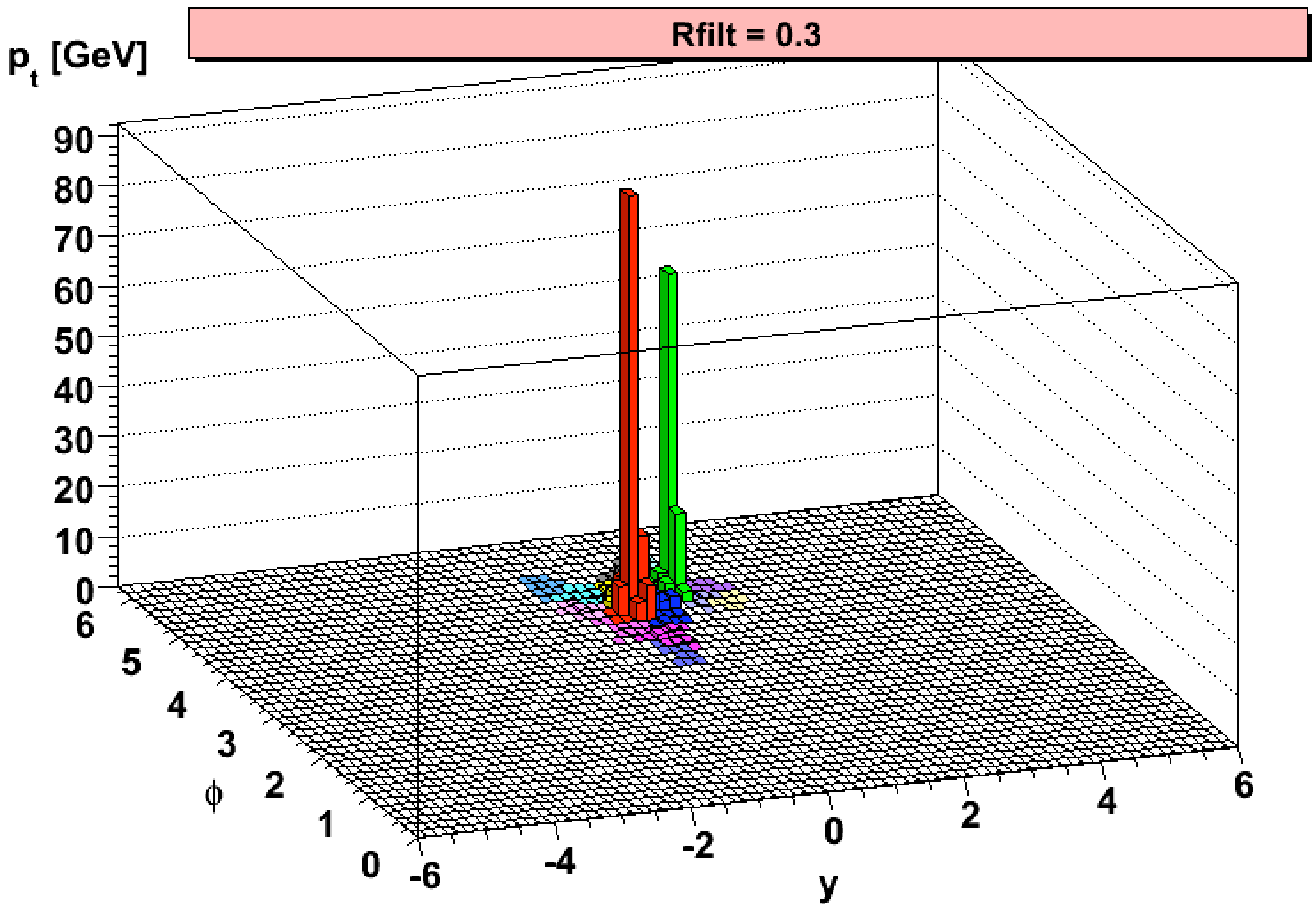
split: $m = 139$ GeV, $\frac{\max(m_1, m2)}{m} = 0.37 \to$ mass drop

$y_{12} = 0.7$ -- OK

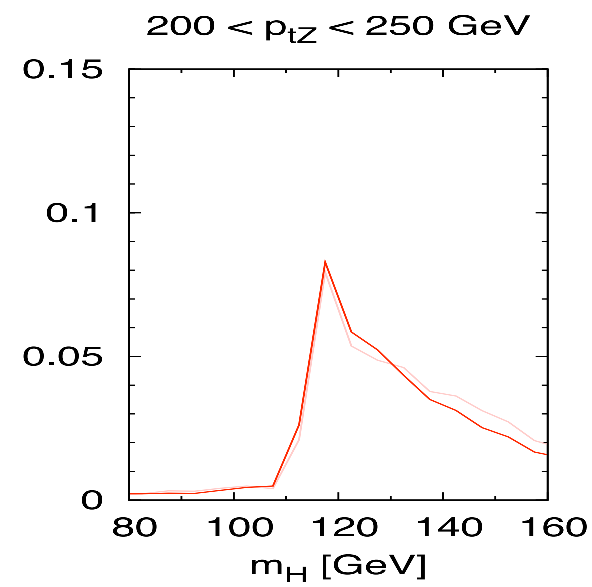
arbitrary norm.

$pp \to ZH \to \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV

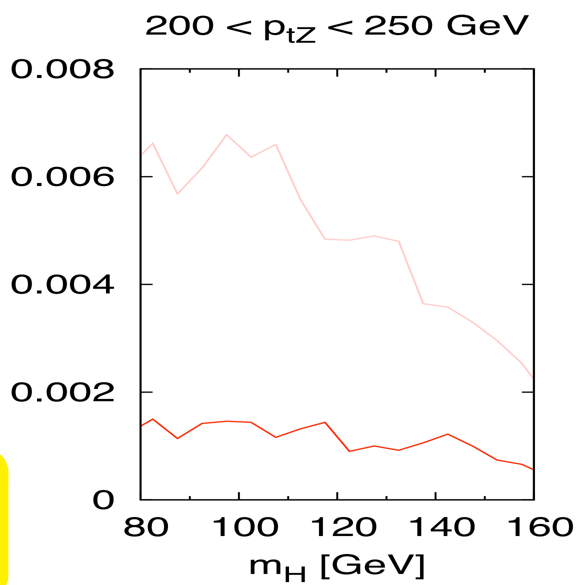
Herwig 6.510 + Jimmy 4.31 + FastJet 2.3



SIGNAL



Zbb BACKGROUND

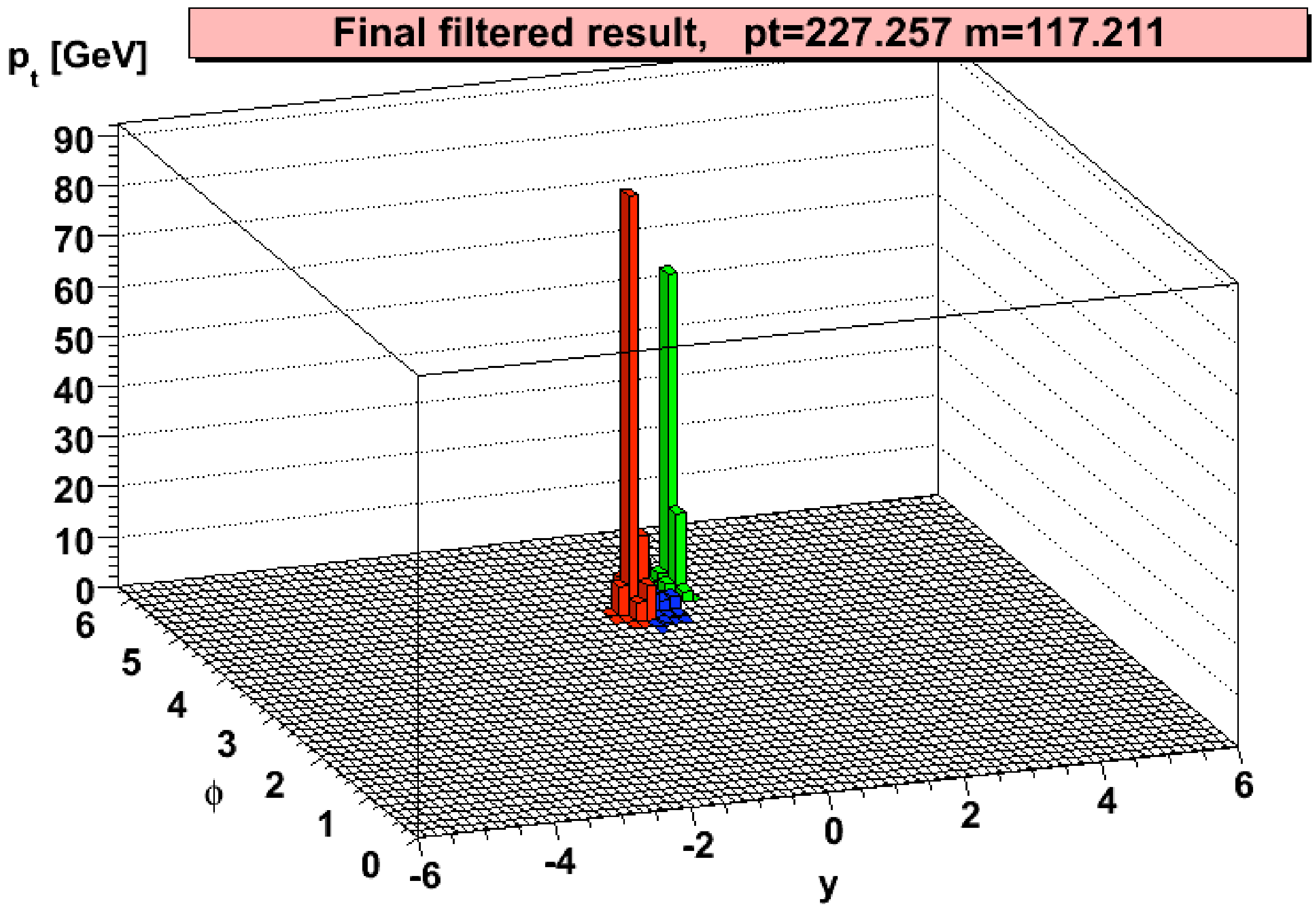


$R_{filt} = 0.3$

arbitrary norm.

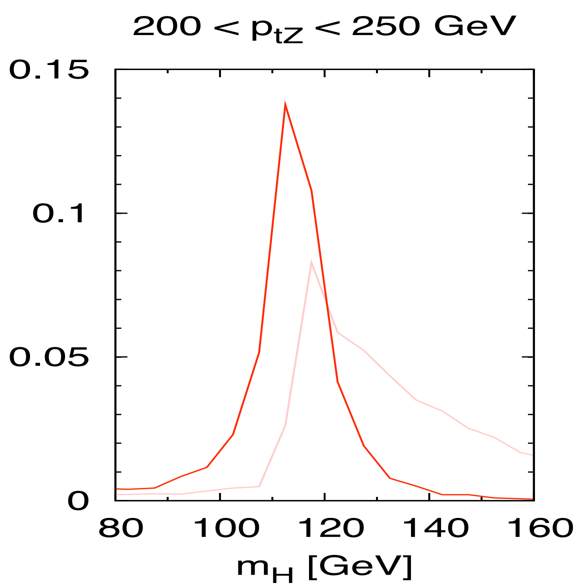
$pp \to ZH \to \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV

Herwig 6.510 + Jimmy 4.31 + FastJet 2.3

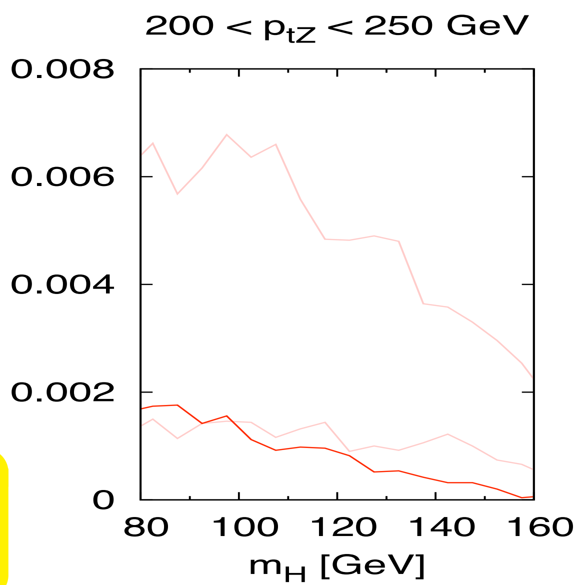


$R_{filt} = 0.3$: take 3 hardest, **m = 117 GeV**

SIGNAL

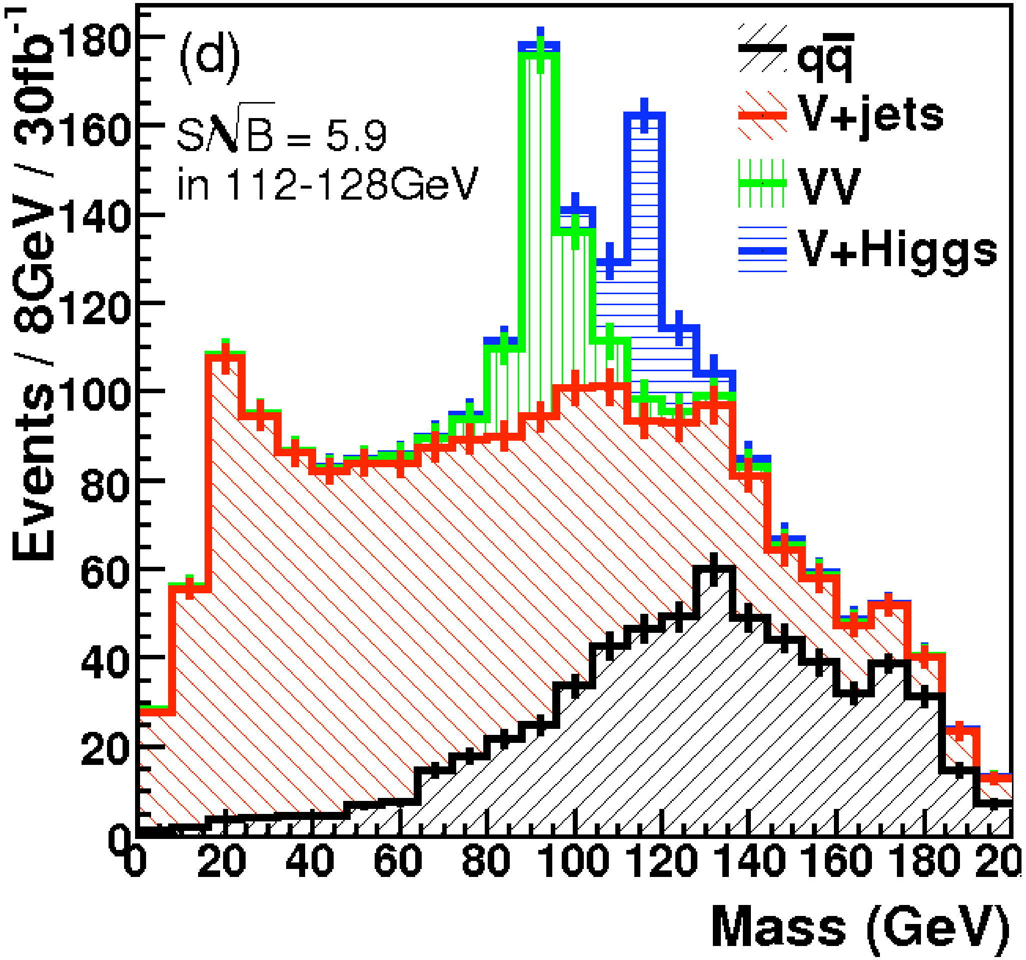


Zbb BACKGROUND



arbitrary norm.

combine HZ and HW, $p_t > 200$ GeV



Common cuts

- $p_{tV}, p_{tH} > 200$ GeV
- $|\eta_H| < 2.5$
- $[p_{t,\ell} > 30$ GeV, $|\eta_\ell| < 2.5]$
- No extra ℓ, b's with $|\eta| < 2.5$
- Real/fake b-tag rates: 0.7/0.01
- $S/\sqrt{B}$ from 16 GeV window

3 channels combined

Note excellent VZ, $Z \to b\bar{b}$ peak for calibration

NB: $q\bar{q}$ is mostly $t\bar{t}$

At 5.9σ for 30 fb^{-1} this looks like a possible new channel for light Higgs discovery. **Deserves serious exp. study!**

The IRC safe algorithms

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

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, and study its characteristics.

- ☑ They can be used to subtract the noise from the hard jets, improving the quality of kinematical reconstructions.

- ☑ ‘Third-generation’ algorithms look promising.

Extra material

The FastJet algorithm

To improve the speed of the algorithm we must find more efficiently which particle is "close" to another and therefore gets combined with it

Observation (MC, G.P. Salam, hep-ph/0512210):

If i and j form the smallest $d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta\eta^2 + \Delta\phi^2}{R^2}$

and

$k_{ti} < k_{tj}$ $\Rightarrow$ $\Delta R_{ij} \leq \Delta R_{ik}$ $\forall\ k \neq j$

i.e. j is the **geometrical** nearest neighbour of i

Translation from mathematics:

When a particle gets combined with another, and has the smallest kt, its partner will be its **geometrical nearest neighbour** on the cylinder spanned by y and ϕ

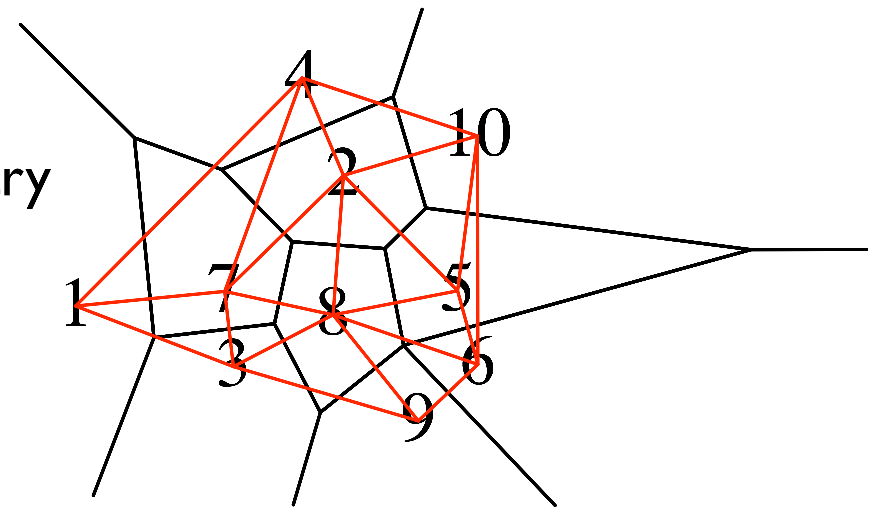
This means that we need to look for partners only among the O(N) nearest neighbours of all particles (a few neighbours each × N particles)

The FastJet algorithm

Our problem has now become a **geometrical** one:
how to find efficiently the (nearest) neighbour(s) of a point

Widely studied problem in computational geometry
Tool: **Voronoi diagram**

Definition: each cell contains the locations which have the given point as nearest neighbour



The **dual** of a Voronoi diagram is a **Delaunay triangulation**

Key feature: once the Voronoi diagram is constructed, <u>the nearest neighbour of a point will be in one of the O(1) cells sharing an edge with its own cell</u>

Example : the G(eometrical) N(earest) N(eighbour) of point 7 will be found among 1,4,2,8 and 3 (it turns out to be 3)

The FastJet algorithm

MC and G.P. Salam, hep-ph/0512210

Construct the Voronoi diagram of the N particles using the CGAL library **O(N lnN)**

Find the GNN of each of the N particles. Construct the d_{ij} distances, store the results in a priority queue (C++ map) **O(N lnN)**

Merge/eliminate particles appropriately

Update Voronoi diagram and distances' map **O(lnN)**

repeat N times

Overall, an O(N ln N) algorithm

NB. Results identical to standard k_t algorithm. This is NOT a new jet-finder.

back

The SISCone algorithm

[Salam, Soyez, arXiv: 0704.0292]

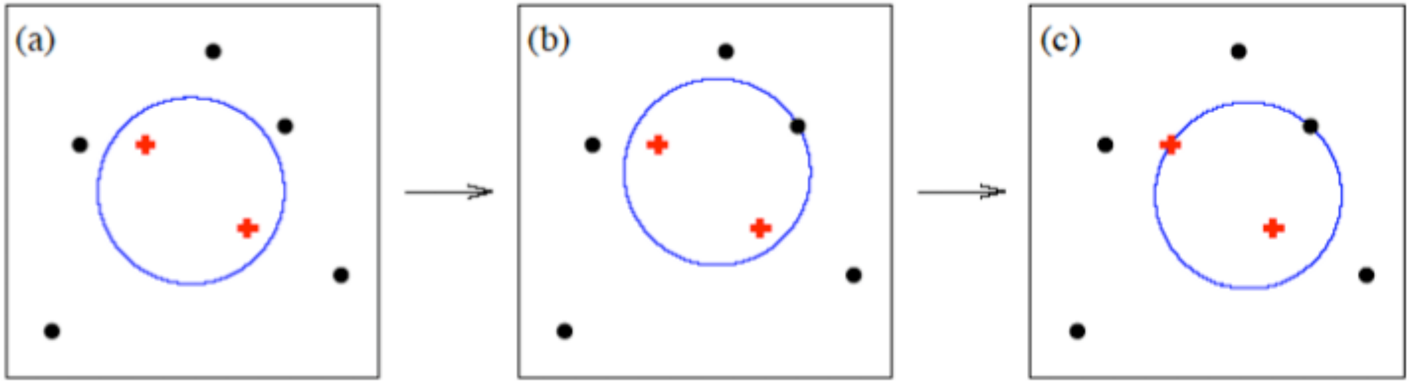
Checking **all particles** in an event to test for stable combinations

(i.e. the axis of the cone containing a subset of particles coincides with the momentum sum)

takes $O(N2^N)$ time

Solution: once more, transform into a geometrical problem

1. Find all distinct way of enclosing a set of particles in a y-ϕ circle
2. Check, for each enclosure, if it corresponds to a stable cone



Finding all distinct circular enclosures of a set of points is geometry:
move it until you hit a point, then rotate it until one of the points hits the edge

Result: Seedles Infrared Safe Cone (SISCone)

[runs in $O(N^2 \ln N)$ time, similar to MidPoint (N^3)]

<Theoretical interlude>

Passive Area

Add a **single** ghost* particle to the event.
Move it around.
Check if it gets clustered in a given jet J.

$$a(J) \equiv \int dy\, d\phi\, f(g(y,\phi),J) \qquad\qquad f(g,J) = \begin{cases} 1 & g \in J \\ 0 & g \notin J \end{cases}$$

* ghost particle: particle with infinitesimally small momentum with respect to all other particles in the event (in practice, O(10^{-100} GeV))

Active Area

Add **many** ghost particles in random configurations to the event.
Cluster many times. *Allow ghosts to cluster among themselves too.*
Count how many ghosts on average get clustered into a given jet J.

$$A(J\,|\,\{g_i\}) = \frac{N_g(J)}{\nu_g}$$

Number of ghosts in jet J

Active area of a single ghosts configuration

Ghost density

$$A(J) = \lim_{\nu_g \to \infty} \langle A(J\,|\,\{g_i\}) \rangle_g$$

Active area

Jet areas: the single hard particle case

[MC, Salam, Soyez]

A jet of 'radius' R will surely have area πR^2, right?

Well, it depends.....

Passive areas of a single hard particle are indeed $\boldsymbol{\pi R^2}$

However, **active** areas are not:

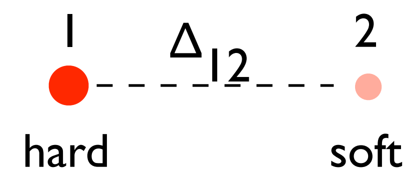
- $k_t \rightarrow 0.81\ \pi R^2$
- Cam/Aa $\rightarrow 0.81\ \pi R^2$
- SISCone $\rightarrow \pi R^2 / 4$
- anti-$k_t \rightarrow \pi R^2$

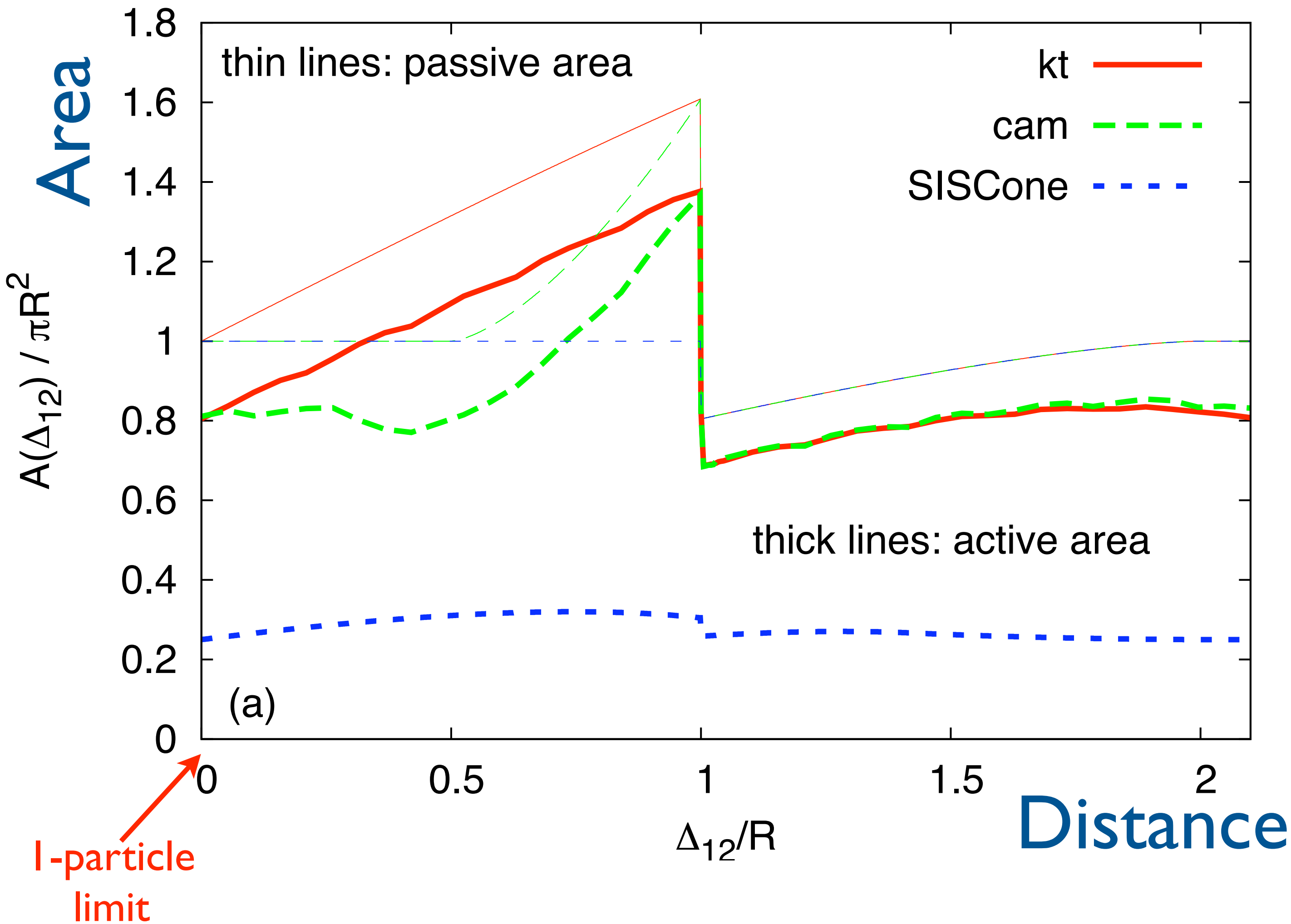
Recall that 'area' is how much background a a jet can pick up.
Its knowledge is essential in order to subtract it from measurements

Jet areas: a second hard(ish) particle

Real events have more than a single hard particle.

Add a second (soft) one at a distance Δ_{12}





The jet area depends on the distance between the particles

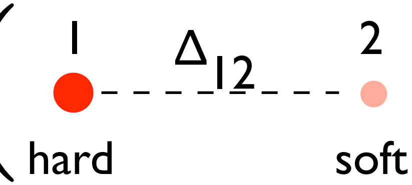
Note very small active area for SISCone!

Passive areas (and SISCone's active area) can be calculated **analytically**, while the others are obtained numerically

Jet areas: anomalous dimensions

Finally, weigh the probability of emission of the soft particle with the leading-order QCD matrix element:

$$\langle \Delta area \rangle = \int C_1 \frac{\alpha_s(p_{t2}\Delta_{12})}{\pi} \frac{dp_{t2}}{p_{t2}} \left[\frac{d\Delta_{12}}{\Delta_{12}}\right]_+ \Bigg(\qquad \Bigg)$$



The result is an **anomalous dimension**.

Areas change with transverse momentum of the jet in a predictable way:

$$\boxed{\langle \Delta area \rangle = \mathrm{d} \; \frac{C_1}{\pi b_0} \ln \frac{\alpha_s(Q_0)}{\alpha_s(R p_{t1})}}$$

In a similar way one can also predict the evolution of the dispersion, calculating

$$\langle \Delta\, area^2 \rangle = \mathrm{s}^2 \, \frac{C_1}{\pi b_0} \ln \frac{\alpha_s(Q_0)}{\alpha_s(R p_{t1})}$$

Passive areas: analytical results

MC, Salam, Soyez, arXiv:0802.1188

d:

$$d_{k_t,R} = \left(\frac{\sqrt{3}}{8} + \frac{\pi}{3} + \xi\right) R^2 \simeq 0.5638\,\pi R^2\,,$$

$$d_{\mathrm{Cam},R} = \left(\frac{\sqrt{3}}{8} + \frac{\pi}{3} - 2\xi\right) R^2 \simeq 0.07918\,\pi R^2\,,$$

$$d_{\mathrm{SISCone},R} = \left(-\frac{\sqrt{3}}{8} + \frac{\pi}{6} - \xi\right) R^2 \simeq -0.06378\,\pi R^2\,,$$

Negative! SISCone jets shrink!

s^2:

$$s^2_{k_t,R} = \left(\frac{\sqrt{3}\pi}{4} - \frac{19}{64} - \frac{15\zeta(3)}{8} + 2\pi\xi\right) R^4 \simeq (0.4499\,\pi R^2)^2\,,$$

$$s^2_{\mathrm{Cam},R} = \left(\frac{\sqrt{3}\pi}{6} - \frac{3}{64} - \frac{\pi^2}{9} - \frac{13\zeta(3)}{12} + \frac{4\pi}{3}\xi\right) R^4 \simeq (0.2438\,\pi R^2)^2\,,$$

$$s^2_{\mathrm{SISCone},R} = \left(\frac{\sqrt{3}\pi}{12} - \frac{15}{64} - \frac{\pi^2}{18} - \frac{13\zeta(3)}{24} + \frac{2\pi}{3}\xi\right) R^4 \simeq (0.09142\,\pi R^2)^2\,.$$

with $\xi \equiv \dfrac{\psi'(1/6) + \psi'(1/3) - \psi'(2/3) - \psi'(5/6)}{48\sqrt{3}} \simeq 0.507471$

Jet areas: passive v. active

	area/πR^2		dispersion		d or D		s or S	
	passive	active	passive	active	passive	active	passive	active
	$a(1PJ)$	$A(1PJ)$	$\sigma(1PJ)$	$\Sigma(1PJ)$	d	D	s	S
k_t	1	0.81	0	0.28	0.56	0.52	0.45	0.41
Cam/Aachen	1	0.81	0	0.26	0.08	0.08	0.24	0.19
SISCone	1	1/4	0	0	−0.06	0.12	0.09	0.07
anti-k_t	1	1	0	0	0	0	0	0

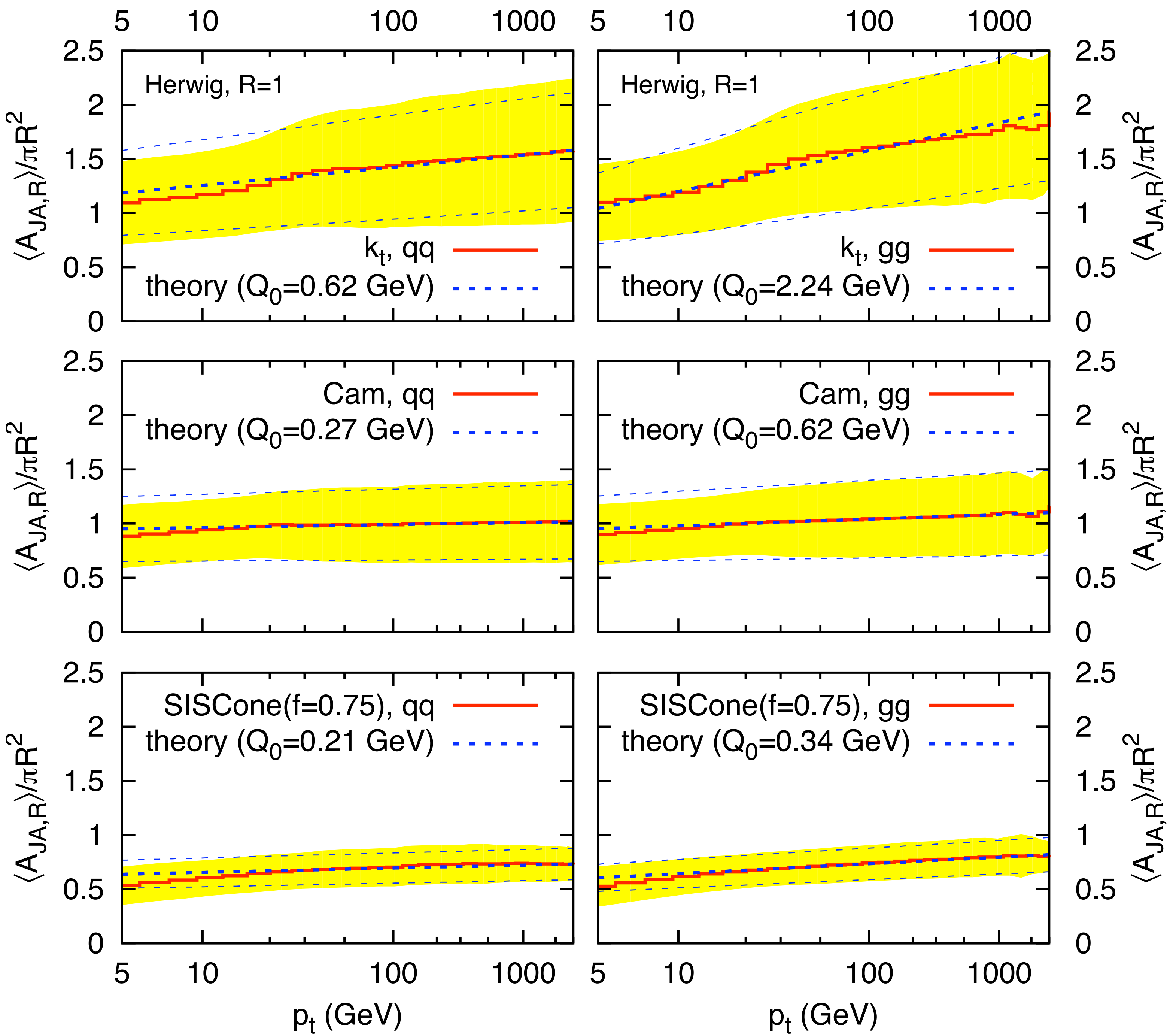
single hard particle

emission of a second perturbative particle (coeff. of anomalous dimension)

Some remarkable features

- SISCone has very small active area
- SISCone's anomalous dimension changes from negative for passive area to positive for active area
- k_t has largest anomalous dimension
- anti-k_t has constant area (null anomalous dimension): it's a **perfect cone**

Jet areas scaling violations



Averages and dispersions evolution from Monte Carlo simulations (dijet events at LHC) in good agreement with simple LL calculations

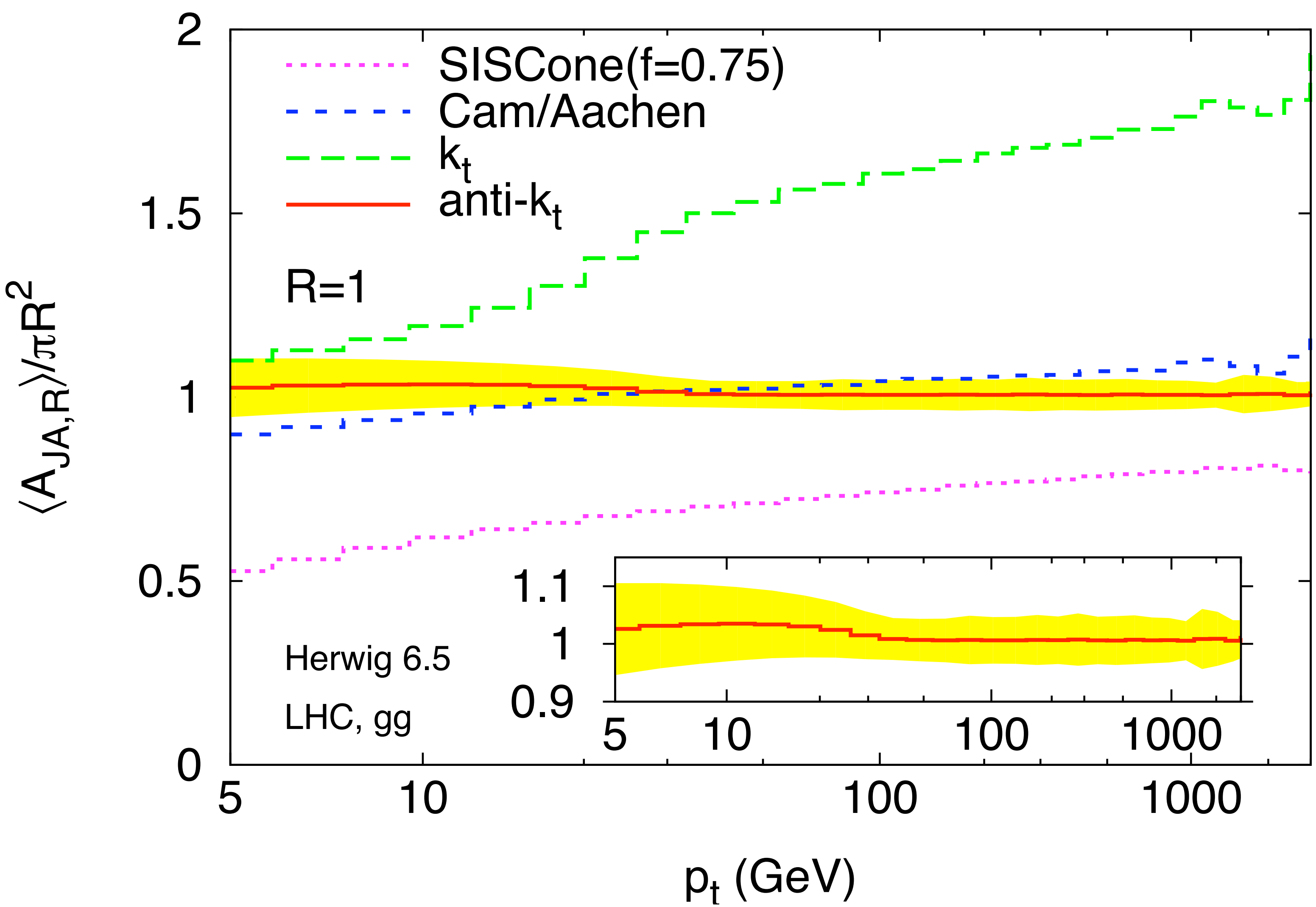
Area scaling violations are a legitimate observable.

(Though it might not be the best place where to measure α_s ....)

Jet areas scaling violations

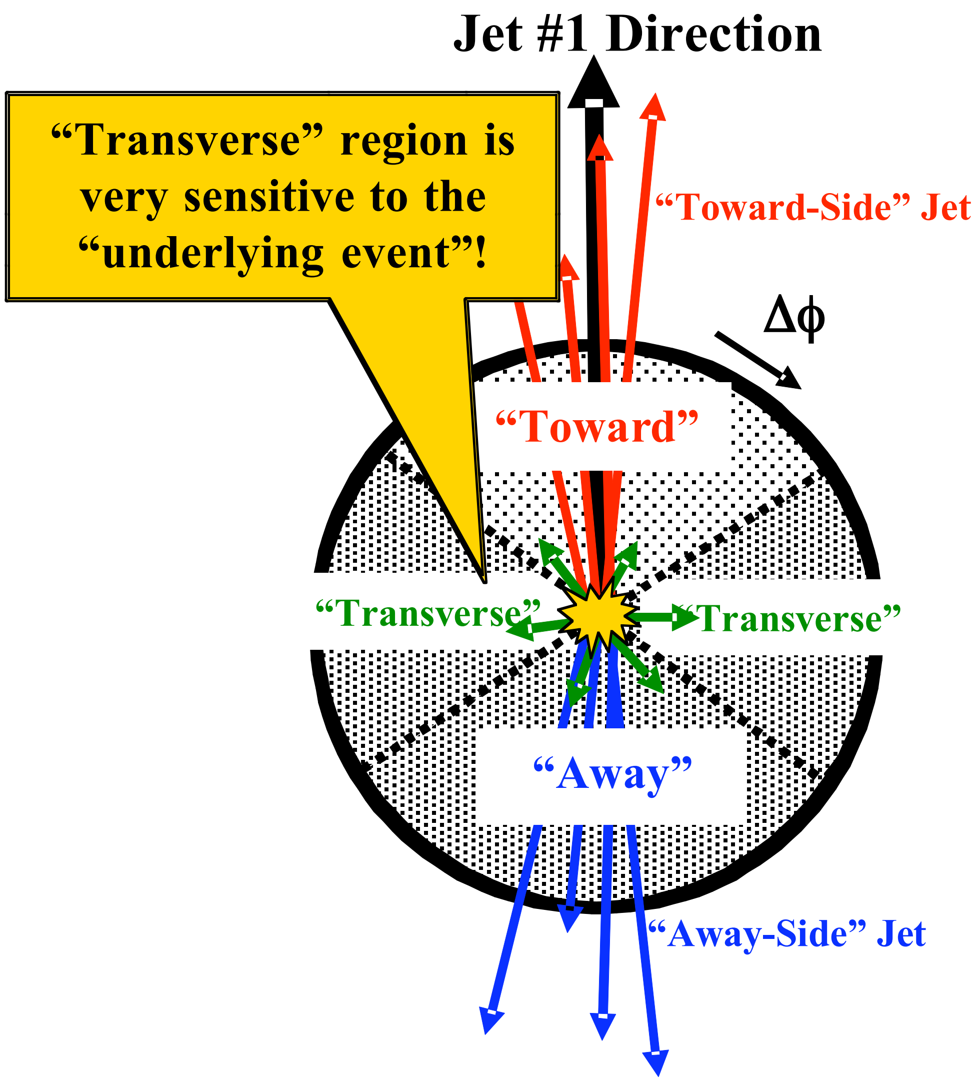
MC, Salam, Soyez, arXiv:0802.1189

Check anti-k_t behaviour: scaling violations indeed absent, as predicted



</Theoretical interlude>

Underlying event measurement



Marchesini-Webber idea:
look at transverse region to measure underlying event

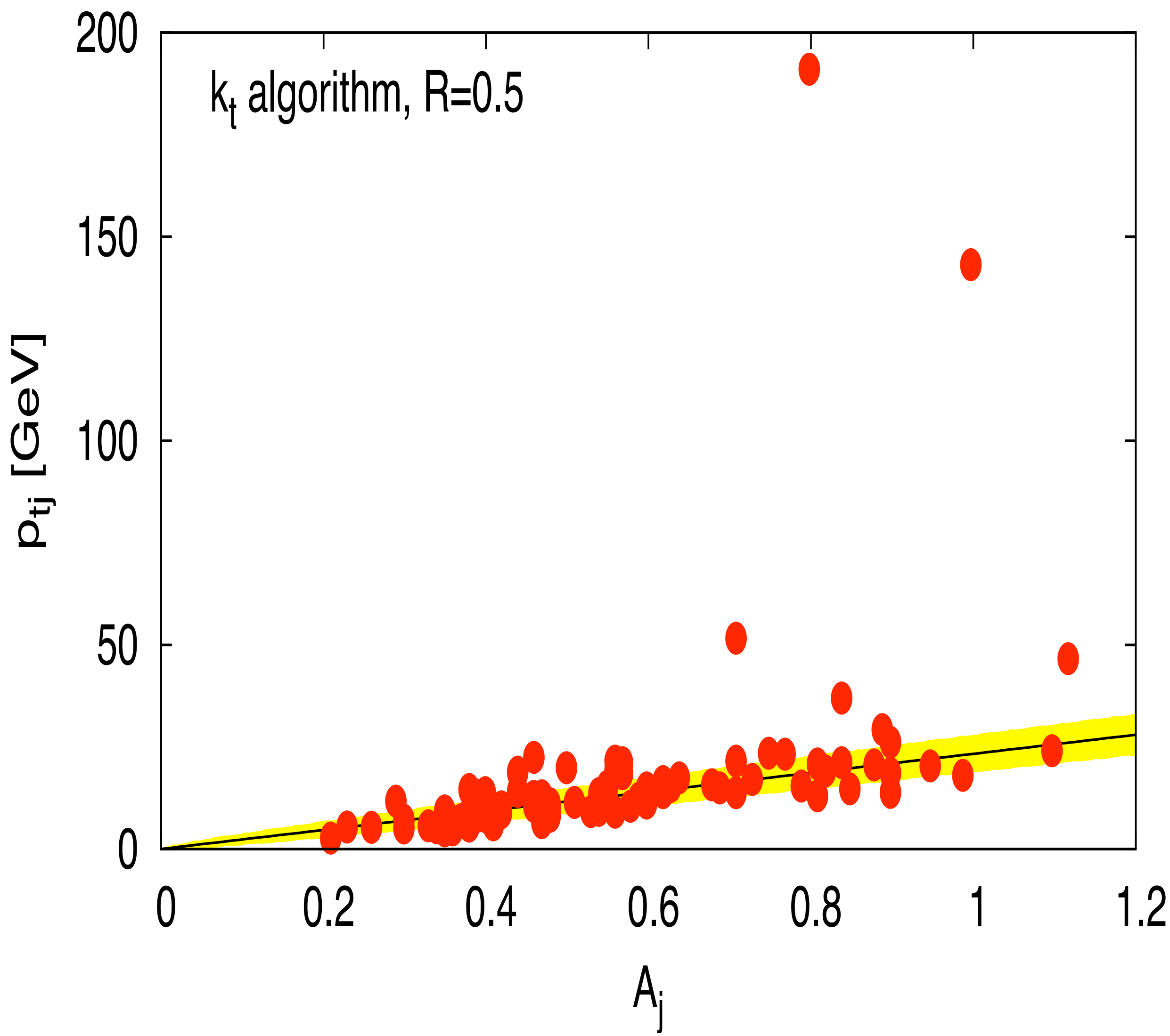
Topological selection

The jets are classified as belonging to the noise on the ground of their **position**

LHC: dijet event + high-lumi pilup

a few hard particles and many softer ones

(a similar picture applies to the Underlying Event)

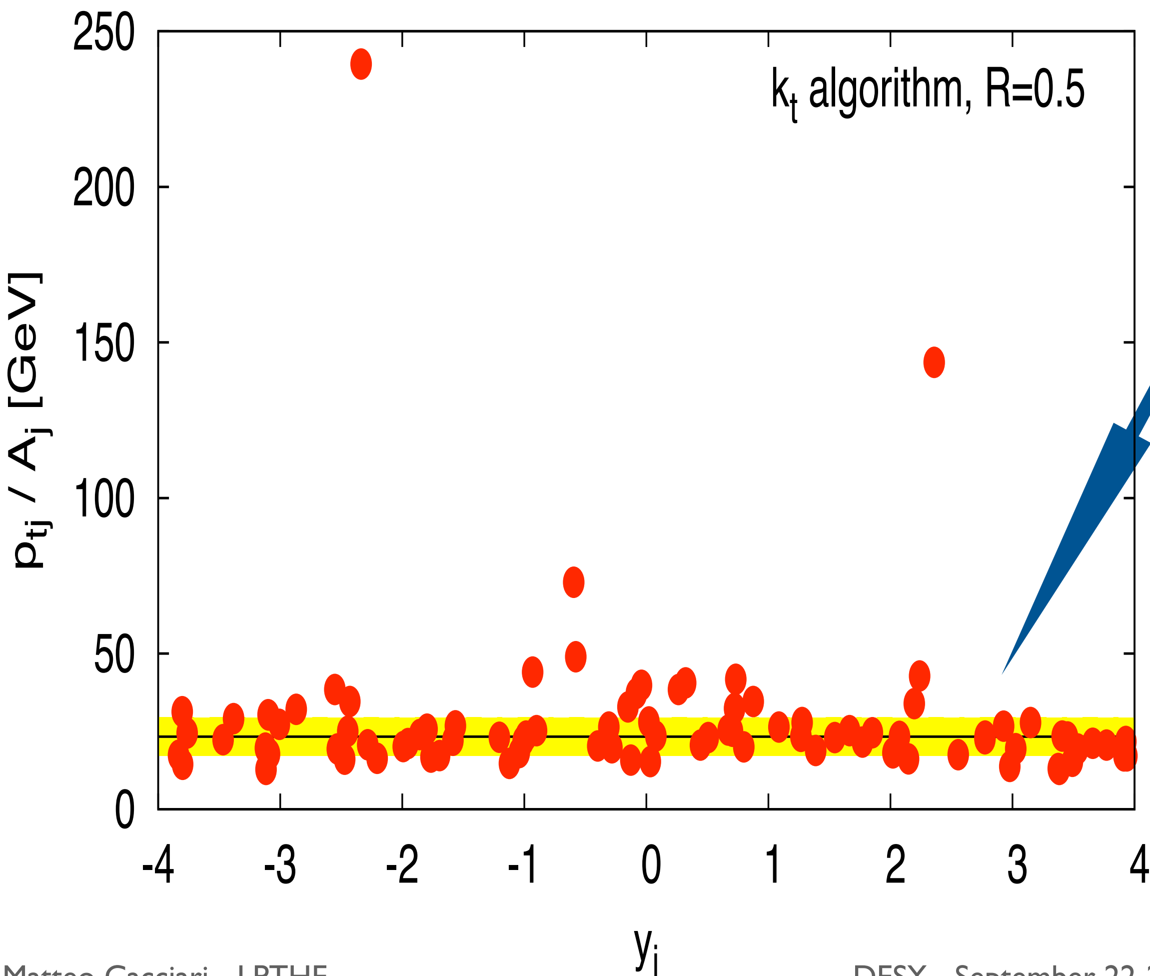


The jets adapt to the surrounding environment

They can have very different areas

The key observation

p_T/Area is fairly constant, except for the hard jets



The distribution of background jets establishes its own average momentum density **ρ**

(NB. this is true on an event-by-event basis)

Dynamical selection

The jets are classified as belonging to the noise on the ground of their **characteristics**