The FastJet ***jet package***

Grégory Soyez

Brookhaven National Laboratory

M. Cacciari, G.P. Salam, G. Soyez,

`http://www.lpthe.jussieu.fr/~salam/fastjet`

Jet definitions

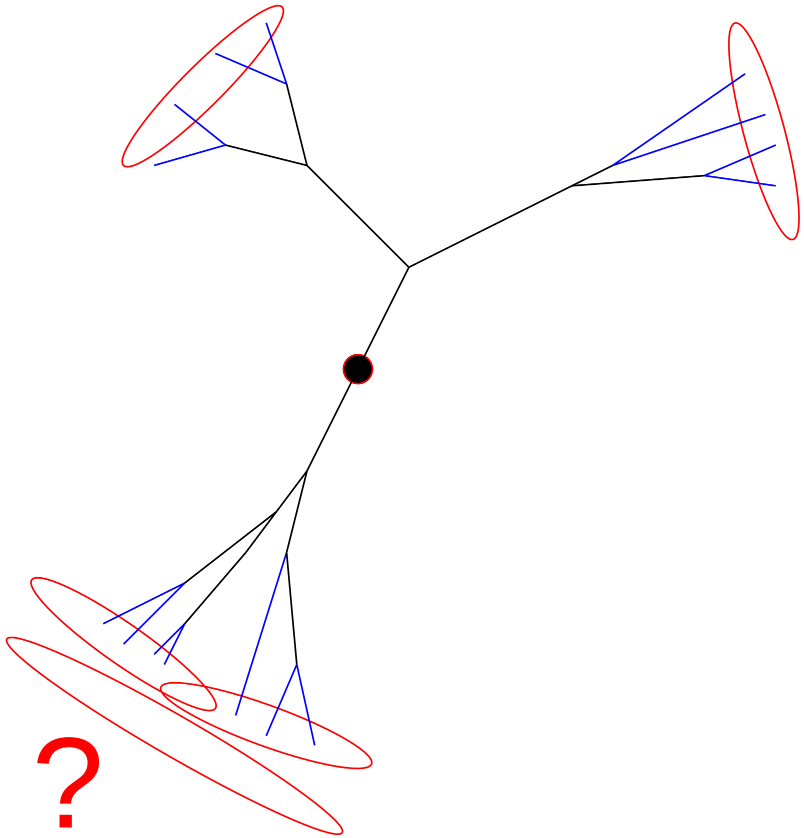
Aim: Study hard processes

- QCD backgrounds, top quark physics
- Higgs, physics beyond the standard model

Define jets: parton $\leftrightarrow$ jet

But: partons are ambiguous

Hence: Multiple definitions of a “jet”



Two classes of algorithms

Class 1: recombination	Cass 2: cone
Successive recombinations of the "closest"[(a)] pair of particle	find directions of energy flow $\equiv$ stable cones[(b)]
Nice perturbative behaviour	Small sensitivity to soft radiation (UE,PU)
Often used in $e^\pm e^\pm$, $e^\pm p$	Often used in pp

(a) Distance:

$$k_t:\quad d_{i,j} = \min(k_{t,i}^2, k_{t,j}^2)(\Delta\phi_{i,j}^2 + \Delta y_{i,j}^2)$$

$$\text{Aachen/Cam.:}\quad d_{i,j} = \Delta\phi_{i,j}^2 + \Delta y_{i,j}^2$$

(b) **stable cones** (radius R) such that:
the total momentum of its contents points in the direction of its centre

How the cone works...

- Seeded (iterative) approaches: iterate from an initial position until stable
  - seed = initial particle
  - seed = midpoint between stable cones found at first step
- One has to deal with overlapping stable cones: 2 subclasses

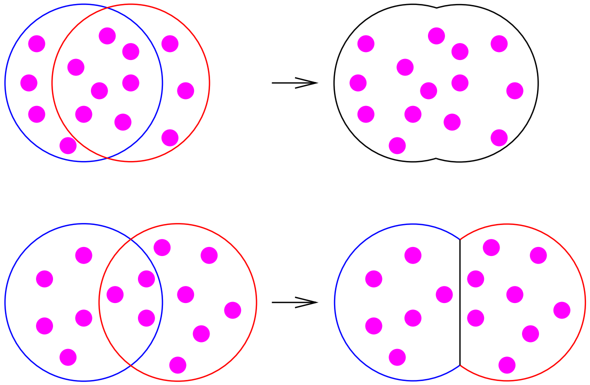
How the cone works...

- Seeded (iterative) approaches: iterate from an initial position until stable
  - seed = initial particle
  - seed = midpoint between stable cones found at first step

Class 2(a): cone with split-merge (ex.: JetClu, Atlas, MidPoint):

$\tilde{p}_{t,\text{shared}} > f\,\tilde{p}_{t,\text{min}}$

$\tilde{p}_{t,\text{shared}} \leq f\,\tilde{p}_{t,\text{min}}$

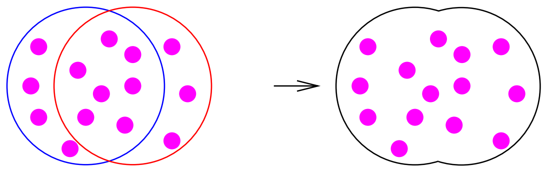


How the cone works...

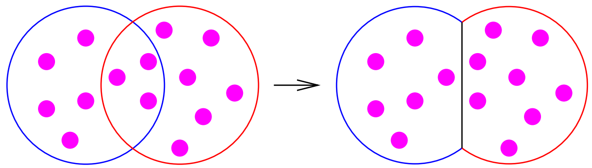
- Seeded (iterative) approaches: iterate from an initial position until stable
  - seed = initial particle
  - seed = midpoint between stable cones found at first step

Class 2(a): cone with split-merge (ex.: JetClu, Atlas, MidPoint):

$\tilde{p}_{t,\text{shared}} > f\,\tilde{p}_{t,\text{min}}$



$\tilde{p}_{t,\text{shared}} \leq f\,\tilde{p}_{t,\text{min}}$



Class 2(b): cone with progressive removal (ex.: Iterative Cone)

- iterate from the hardest seed
- remove the stable cone as a jet and start again

Idea: “regular/circular” jets

Progress in jet definitions

- Recombination algorithms
  - 20th century: the k_t algorithm is too slow – $\mathcal{O}\left(N^3\right)$
  - Today: fast implementation using algorithmic geometry – $\mathcal{O}\left(N\ \log(N)\right)$ (the 'fast' in FastJet)

[M.Cacciari,G.P.Salam,06]

- Cone algorithms
  - 20th century: infrared-and/or-collinear (IRC) unsafe
  - Today:
    - SISCone: IRC-safe replacement for JetClu,MidPoint-type algs.

      [G.P.Salam,GS.,07]
    - anti-k_t: IRC-safe replacement for Iterative-Cone algorithms

      [M.Cacciari,G.P.Salam,GS.,08]

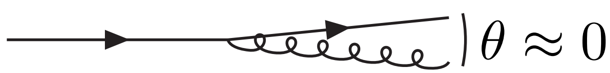
SISCone ***and anti-***$\mathrm{k_t}$

QCD divergences

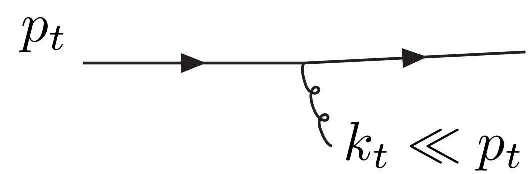
QCD probability for gluon bremsstrahlung at angle θ and $\perp$-mom. k_t:

$$dP \propto \alpha_s \frac{d\theta}{\theta} \frac{dk_t}{k_t}$$

Two divergences:



Collinear



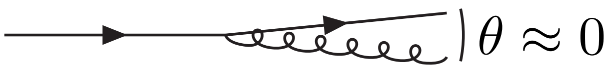
Soft

QCD divergences

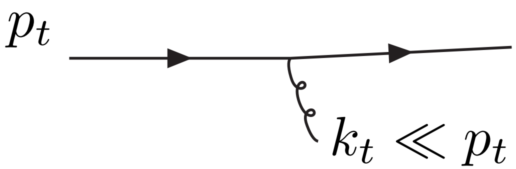
QCD probability for gluon bremsstrahlung at angle θ and $\perp$-mom. k_t:

$$dP \propto \alpha_s \frac{d\theta}{\theta} \frac{dk_t}{k_t}$$

Two divergences:



Collinear



Soft

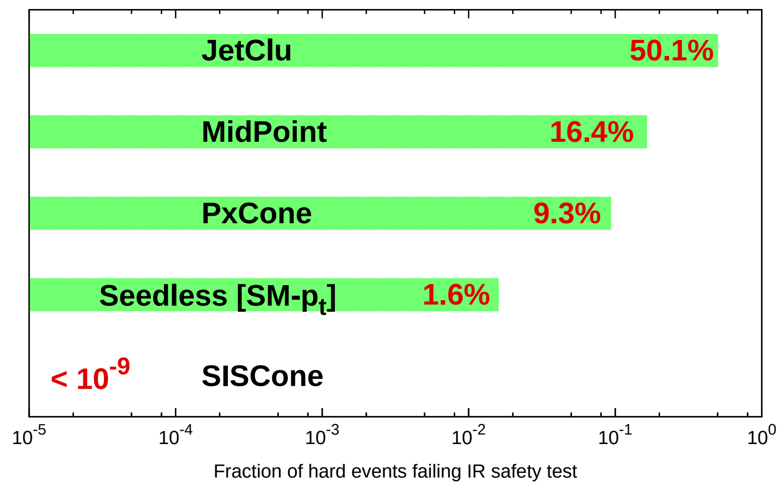
For pQCD to make sense, the (hard) jets (or stable cones) should not change when

- one has a collinear splitting
  i.e. replaces one parton by two at the same place (η, ϕ)
- one has a soft emission *i.e.* adds a very soft gluon

IRC unsafety

Random “hard+soft” events:

- JetClu,ATLAS cone: 50% failure
- MidPoint, iter. cone: 15% failure



Midpoint and the iterative cone IR or Collinear unsafe† at $\mathcal{O}(\alpha_s^4)$ ($\mathcal{O}(\alpha_s^3)$ for JetClu)

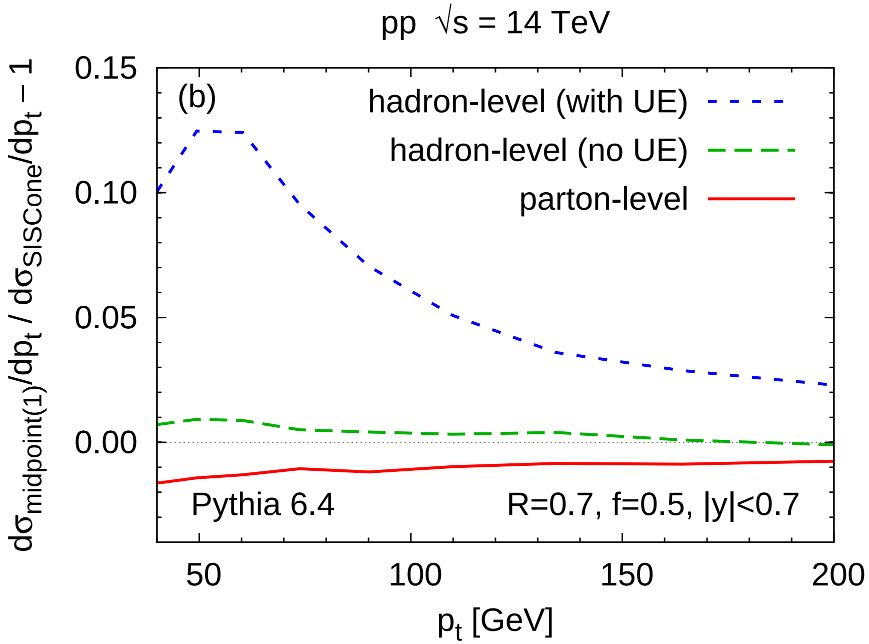
Observable	1st miss cones at	Last meaningful order
Inclusive jet cross section	NNLO	NLO
3 jet cross section	NLO	LO (NLO in NLOJet)
$W/Z/H$ + 2 jet cross sect.	NLO	LO (NLO in MCFM)
jet masses in 3 jets	LO	none (LO in NLOJet)

$\Rightarrow$ The IR-unsafety issue will matter at LHC

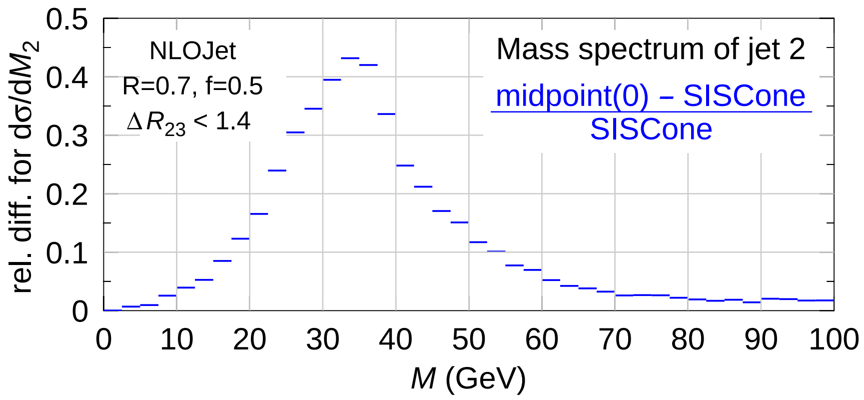
+ We do not want the theoretical efforts to be wasted

SISCone vs. MidPoint

Inclusive (midpoint/SISCone-1)



Masses in 3-jet events



- Inclusive cross-section:
  - Effect of a few percents
  - Less sensitivity to the UE
- More exclusive processes: effects $\sim 45\%$ (Important for LHC!)
- HI : mass $\leftrightarrow$ substructure $\leftrightarrow$ quenching

Anti-k_t vs. the iterative cone

Come back to recombination-type algorithms:

$$d_{ij} = \min(k_{t,i}^{2p}, k_{t,j}^{2p})\ \left(\Delta\phi_{ij}^2 + \Delta\eta_{ij}^2\right)$$

- $p = 1$: k_t algorithm
- $p = 0$: Aachen/Cambridge algorithm
- $p = -1$: anti-k_t algorithm [M.Cacciari, G.Salam, G.S.,JHEP 04 (08) 063]

Anti-k_t vs. the iterative cone

Come back to recombination-type algorithms:

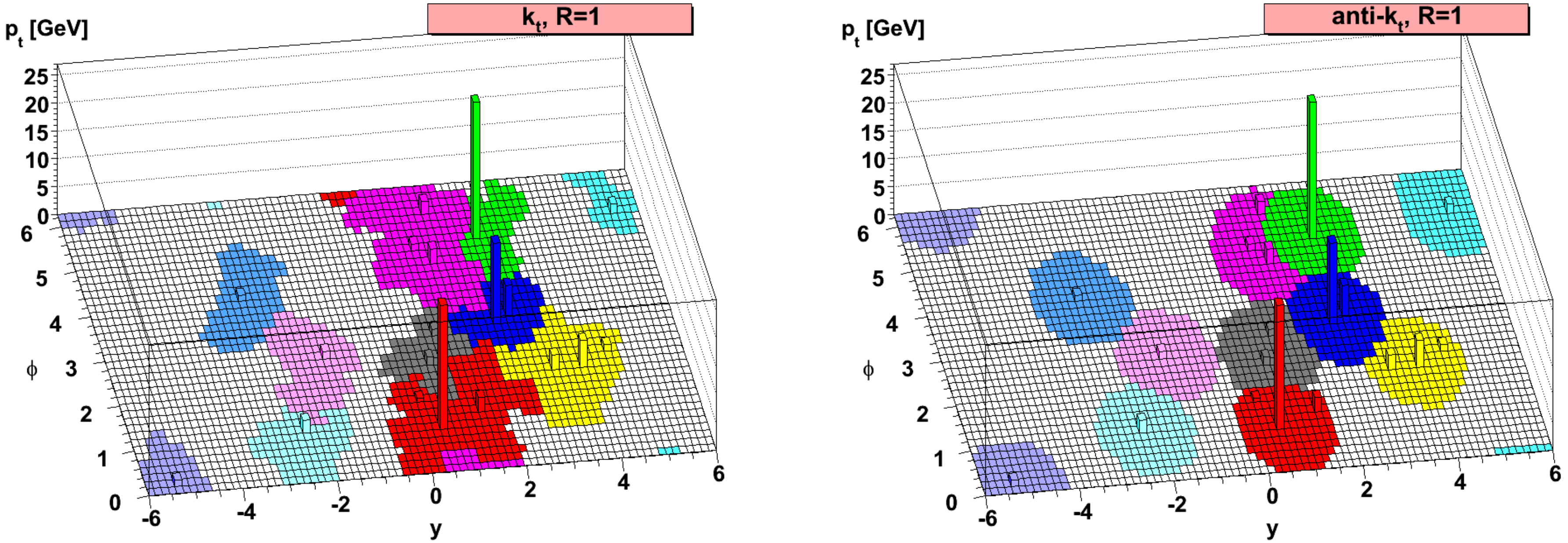
$$d_{ij} = \min(k_{t,i}^{2p}, k_{t,j}^{2p})\ (\Delta\phi_{ij}^2 + \Delta\eta_{ij}^2)$$

- $p = 1$: k_t algorithm
- $p = 0$: Aachen/Cambridge algorithm
- $p = -1$: anti-k_t algorithm [M.Cacciari, G.Salam, G.S.,JHEP 04 (08) 063]

Why should that be related to the iterative cone ?!?

- "large $k_t \Rightarrow$ small distance"
  i.e. hard partons "eat" everything up to a distance R
  i.e. circular/regular jets, jet borders unmodified by soft radiation
- infrared and collinear safe

Hard event + homogeneous soft background



anti-k_t is soft-resilient

FastJet

The FastJet *package*

- FastJet is an interface for running jet clustering

- FastJet is an interface for running jet clustering
- Jet algorithms:
  - Native: recombination algs. (fast implementation)
    - k_t
    - Cambridge/Aachen
    - anti-k_t
    - e^+e^- algorithms in preparation
  - Plugins:
    - SISCone
    - CDF JetClu and CDF MidPoint (IRC unsafe)
    - PxCone (IRC unsafe)
    - D0 run II cone (IRC unsafe)

The FastJet *package*

- FastJet is an interface for running jet clustering
- Jet algorithms:
  - Native: recombination algs. (fast implementation)
    - k_t
    - Cambridge/Aachen
    - anti-k_t
    - e^+e^- algorithms in preparation
  - Plugins:
    - SISCone
    - CDF JetClu and CDF MidPoint (IRC unsafe)
    - PxCone (IRC unsafe)
    - D0 run II cone (IRC unsafe)
- Computation of jet areas and pileup subtraction

Algorithm speed

Idea: use geometric arguments

	recomb.	cone
before:	Naive: $\mathcal{O}\left(N^3\right)$	Naive: $\mathcal{O}\left(N2^N\right)$ Midpoint: $\mathcal{O}\left(N^3\right)$
now:		

Algorithm speed

Idea: use geometric arguments

	recomb.	cone
before:	Naive: $\mathcal{O}\left(N^3\right)$	Naive: $\mathcal{O}\left(N2^N\right)$ Midpoint: $\mathcal{O}\left(N^3\right)$
now:	Factorisation: $\mathcal{O}\left(N^2\right)$	

One can factorise the k_t-dependent part

For the purely geometric part ($\equiv$ Cam/Aachen): iteration less coslty

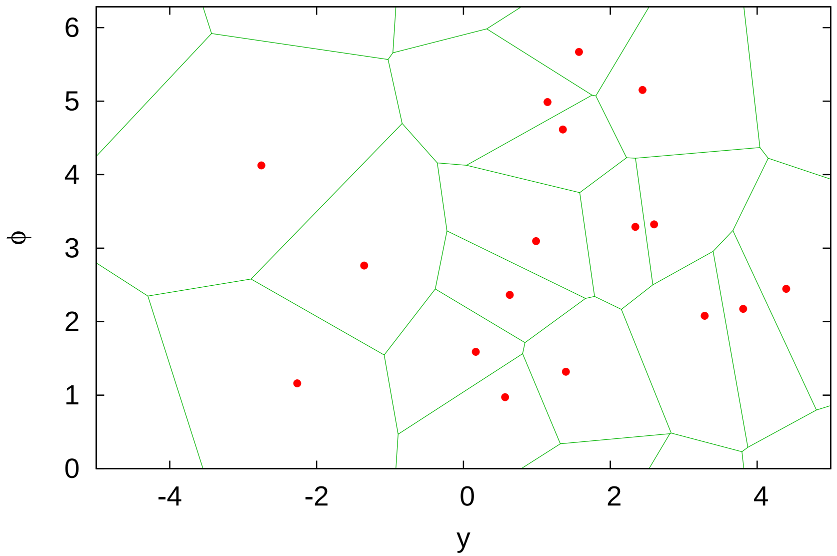
$$\Rightarrow \mathcal{O}\left(N^2\right)$$

Algorithm speed

Idea: use geometric arguments

	recomb.	cone
before:	Naive: $\mathcal{O}\left(N^3\right)$	Naive: $\mathcal{O}\left(N2^N\right)$ Midpoint: $\mathcal{O}\left(N^3\right)$
now:	Factorisation: $\mathcal{O}\left(N^2\right)$ Voronoi: $\mathcal{O}\left(N\log(N)\right)$	

Dynamic Nearest Neighbour using the Voronoi diagram



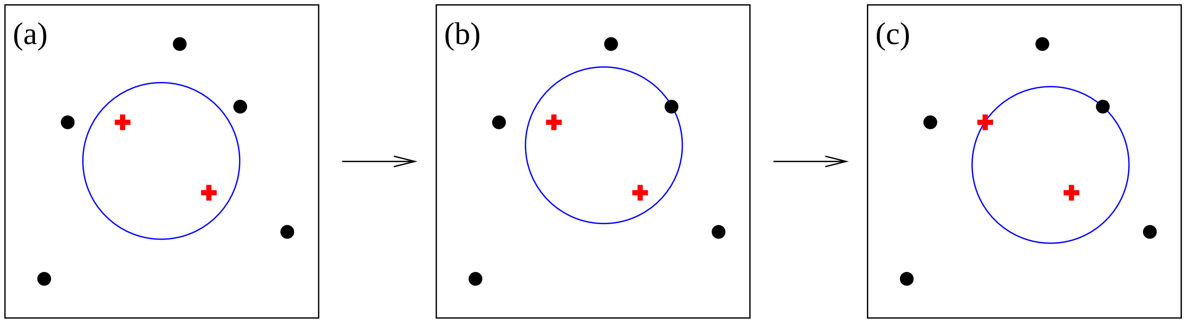
[M. Cacciari, G. Salam, 06]

Algorithm speed

Idea: use geometric arguments

	recomb.	cone
before:	Naive: $\mathcal{O}\left(N^3\right)$	Naive: $\mathcal{O}\left(N2^N\right)$ Midpoint: $\mathcal{O}\left(N^3\right)$
now:	Factorisation: $\mathcal{O}\left(N^2\right)$ Voronoi: $\mathcal{O}\left(N\log(N)\right)$	SISCone: $\mathcal{O}\left(N^2\log(N)\right)$

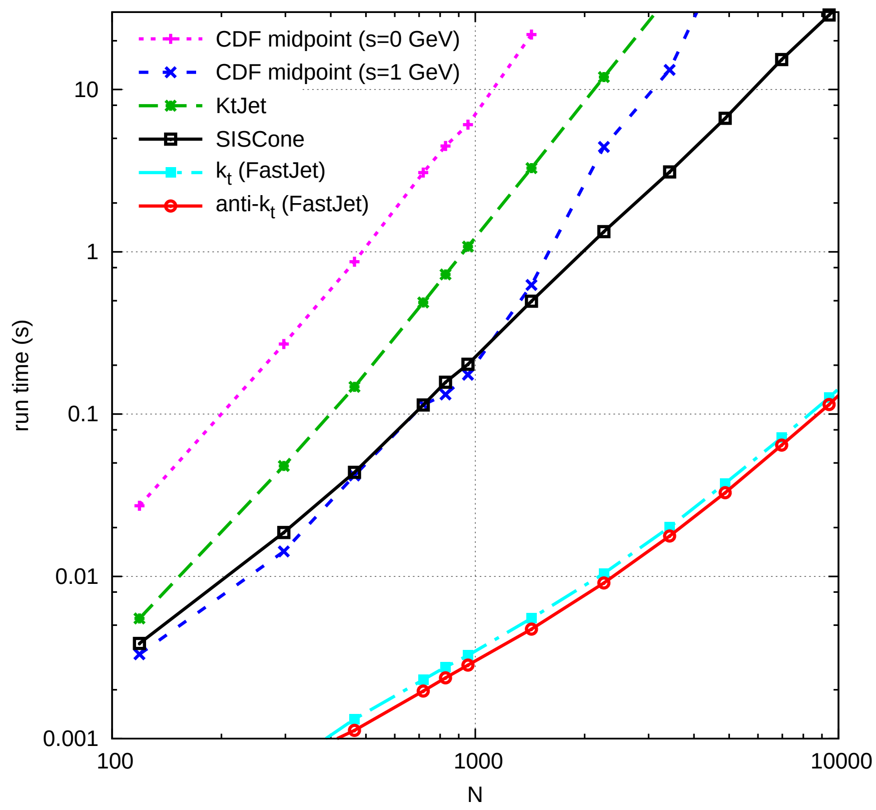
Every enclosure moved to touch two points



Enumerate enclosures $\equiv$ enulerate pairs of points

Algorithm speed (2)

Execution timings:



- SISCone (at least) as fast as Midpoint (with a 1 GeV seed threshold)
- FastJet k_t much faster than KtJet ($\mathcal{O}\left(N^3\right)$)
- anti-$k_t \approx k_t$ (still faster than SISCone)

Clustering done using 3 major classes

- Class #1: fastjet::PseudoJet
  Used to deal with 4-vectors
  PseudoJet p;p.px(), p.py(), p.pz(), p.E(), p.rap(), p.phi(),...

- Class #2: fastjet::JetDefinition(*"algorithm", "parameters"*)
  Used to define the clustering recipe i.e. algorithm + parameters (e.g. R)

- Class #3: fastjet::ClusterSequence(vector$\langle$fastjet::PseudoJet$\rangle$, fastjet::JetDefinition)
  Really perform the clustering

FastJet *usage example*

Example: Cluster particles from the command line and print out jets

```
#include <iostream>
#include <vector>
#include <fastjet/PseudoJet.hh>
#include <fastjet/ClusterSequence.hh>


using namespace std;

void main(){
  // read the particles
  vector<fastjet::PseudoJet> particles;
  double px, py , pz, E;
  while (cin >> px >> py >> pz >> E)
    particles.push_back(fastjet::PseudoJet(px,py,pz,E));

  // declare a jet definition
  double R = 0.5;
  fastjet::JetDefinition jet_def(kt_algorithm, R);

  // perform the clustering
  fastjet::ClusterSequence clust_seq(particles, jet_def);

  // retrieve the jets and print them out
  double ptmin = 0.0;
  vector<fastjet::PseudoJet> jets = sorted_by_pt(clust_seq.inclusive_jets(ptmin));

  for (unsigned int i=0;i<jets.size();i++)
    cout << jets[i].perp() << " " << jets[i].rap() << " " << jets[i].phi() << endl;


}
```

FastJet *usage example*

Example: Cluster particles from the command line and print out jets

```
#include <iostream>
#include <vector>
#include <fastjet/PseudoJet.hh>
#include <fastjet/ClusterSequence.hh>
#include <fastjet/SISConePlugin.hh>

using namespace std;

void main(){
  // read the particles
  vector<fastjet::PseudoJet> particles;
  double px, py , pz, E;
  while (cin >> px >> py >> pz >> E)
    particles.push_back(fastjet::PseudoJet(px,py,pz,E));

  // declare a jet definition
  double R = 0.5;
  fastjet::JetDefinition jet_def = new SISConePlugin(R,0.75);

  // perform the clustering
  fastjet::ClusterSequence clust_seq(particles, jet_def);

  // retrieve the jets and print them out
  double ptmin = 0.0;
  vector<fastjet::PseudoJet> jets = sorted_by_pt(clust_seq.inclusive_jets(ptmin));

  for (unsigned int i=0;i<jets.size();i++)
    cout << jets[i].perp() << " " << jets[i].rap() << " " << jets[i].phi() << endl;

  delete jet_def.plugin();
}
```

Jet area

Everyone has an idea of what a jet area is

but can we define that properly?

[M. Cacciari, G. Salam, G.S., JHEP 04 (2008) 5]

[M. Cacciari, G. Salam, Phys. Lett. B 659 (2008) 119]

Area definition

- Idea: add soft particles (ghosts) and look in which jets they are caught

jet area = region where it catches ghosts

Area definition

- Idea: add soft particles (ghosts) and look in which jets they are caught

jet area = region where it catches ghosts

- 2 definitions
  - Passive area
    add one ghost and look where it ends. repeat to cover the (y, ϕ) plane
  - Active area
    add a large amount of ghosts and cluster everything
    also gives purely ghosted jets

Area definition

- Idea: add soft particles (ghosts) and look in which jets they are caught

jet area = region where it catches ghosts

- 2 definitions
  - Passive area

    add one ghost and look where it ends. repeat to cover the (y, ϕ) plane
  - Active area

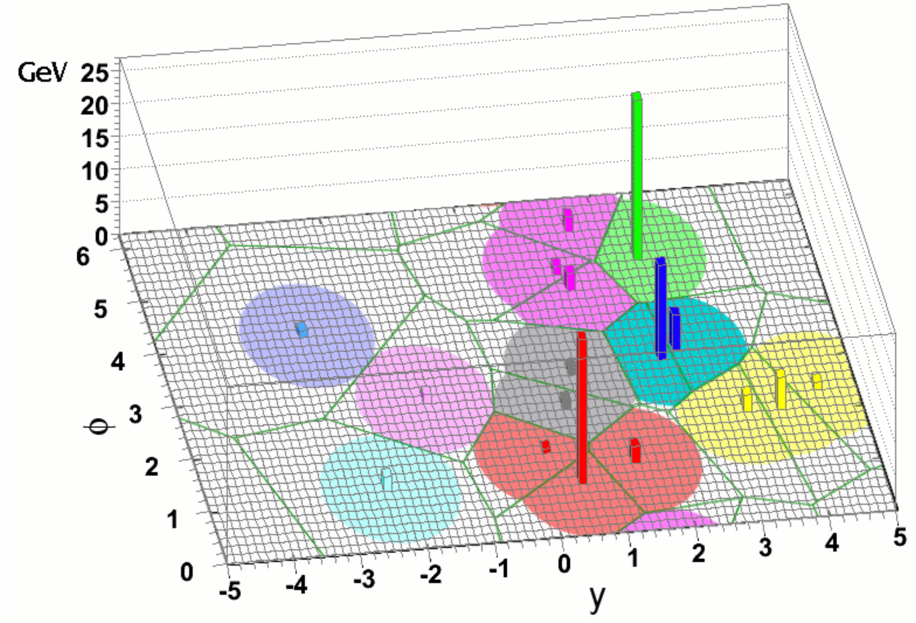
    add a large amount of ghosts and cluster everything

    also gives purely ghosted jets
- Both definitions agree for dense events
- Both practical and tractable analytically

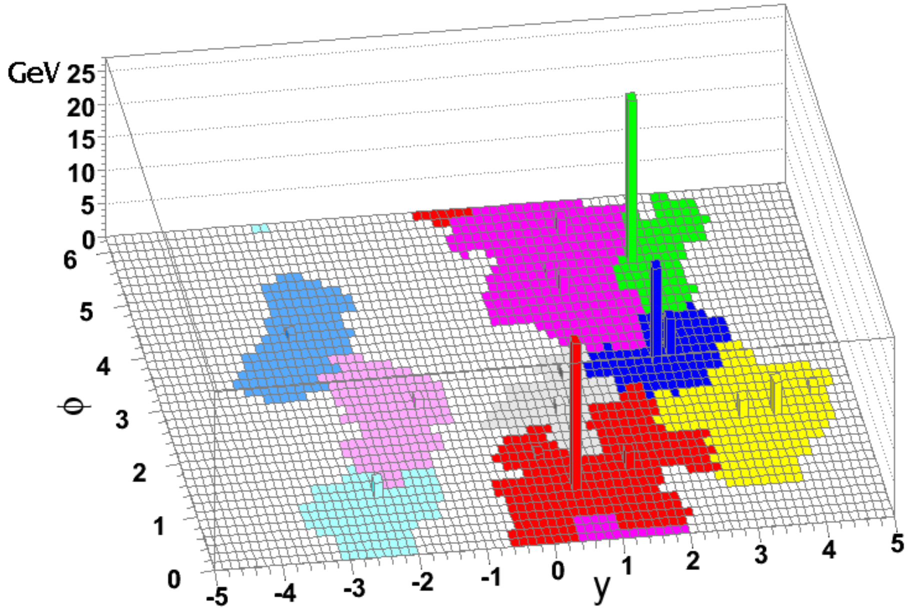
Example

- Hard event + ghost added at each point of the grid
- Jet areas = shaded regions

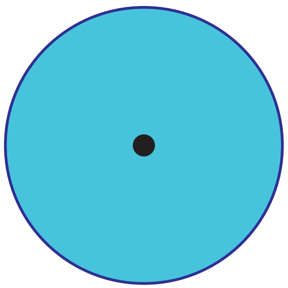
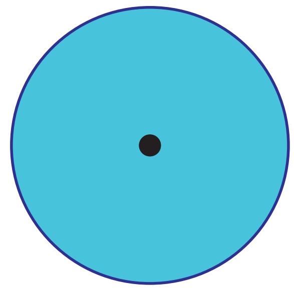
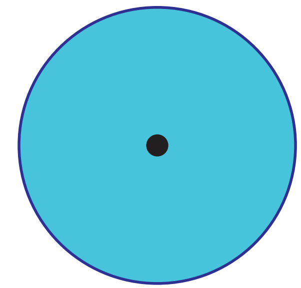
k_t algorithm, passive area



k_t algorithm, active area

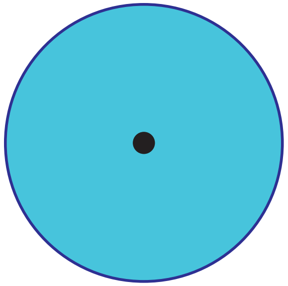
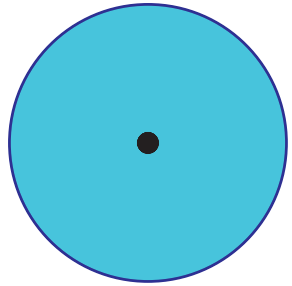
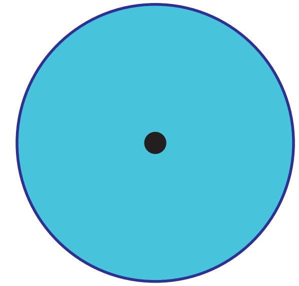
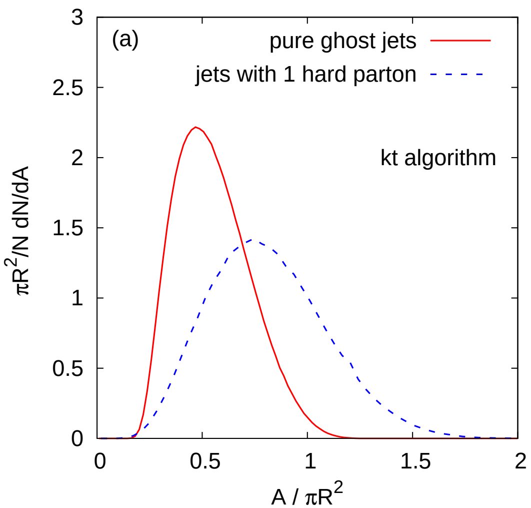
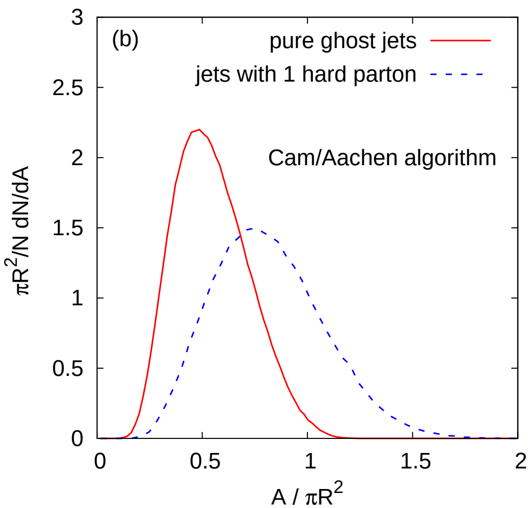
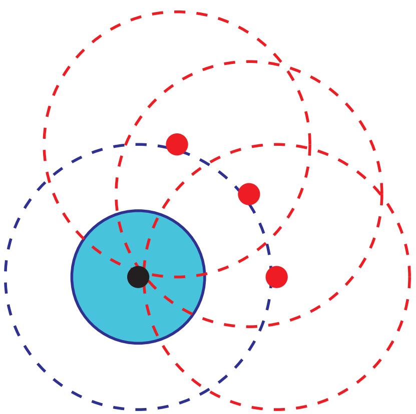


Examples: 1-particle cases

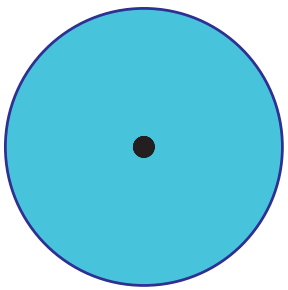
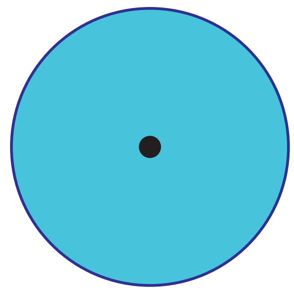
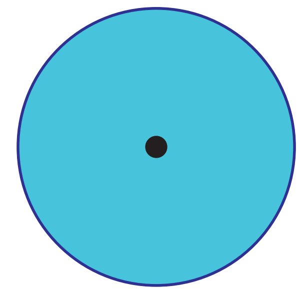
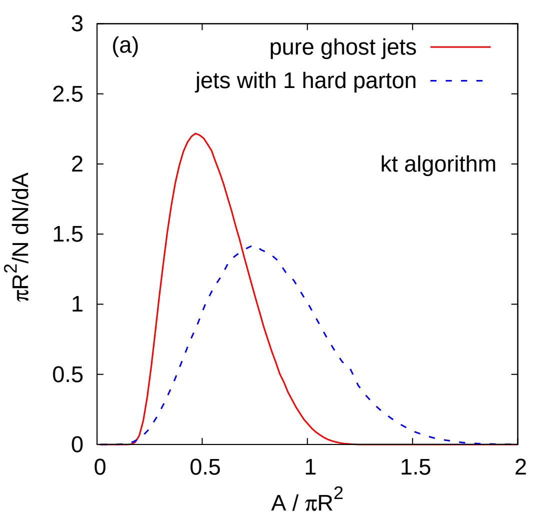
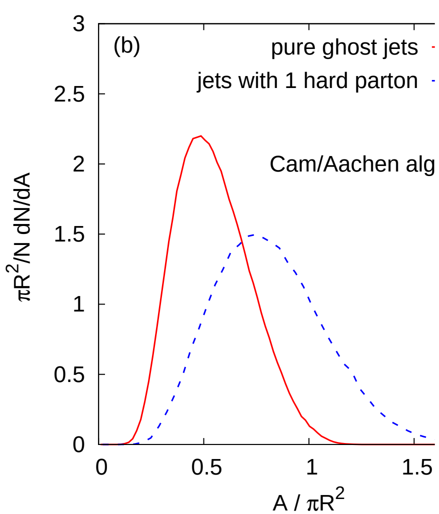
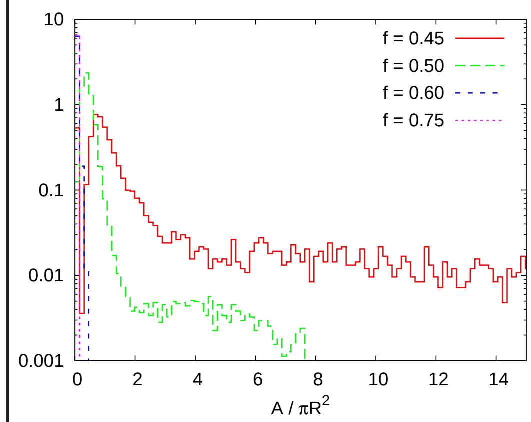
	k_t	Aac/Cam	cone
Passive	 πR^2	 πR^2	 πR^2
Active			

Examples: 1-particle cases

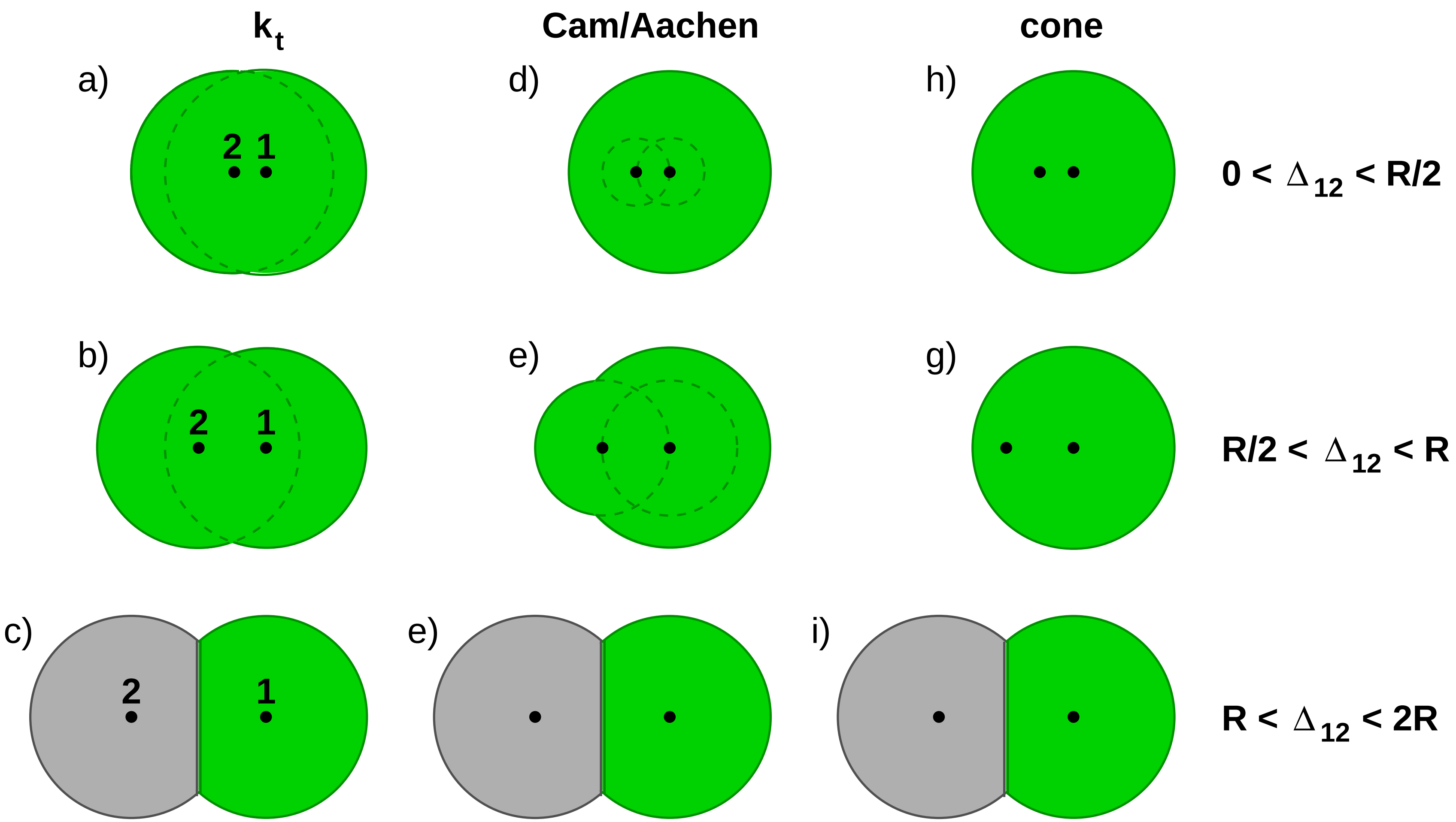
BROOKHAVEN NATIONAL LABORATORY

	k_t	Aac/Cam	cone
Passive	 πR^2	 πR^2	 πR^2
Active	 $\frac{A_{\text{hard}}}{\pi R^2} \approx 0.812 \pm 0.277$ $\frac{A_{\text{ghost}}}{\pi R^2} \approx 0.554 \pm 0.174$	 $\frac{A_{\text{hard}}}{\pi R^2} \approx 0.814 \pm 0.261$ $\frac{A_{\text{ghost}}}{\pi R^2} \approx 0.551 \pm 0.176$	 $\frac{A_{\text{hard}}}{\pi R^2} = 0.25$

Examples: 1-particle cases

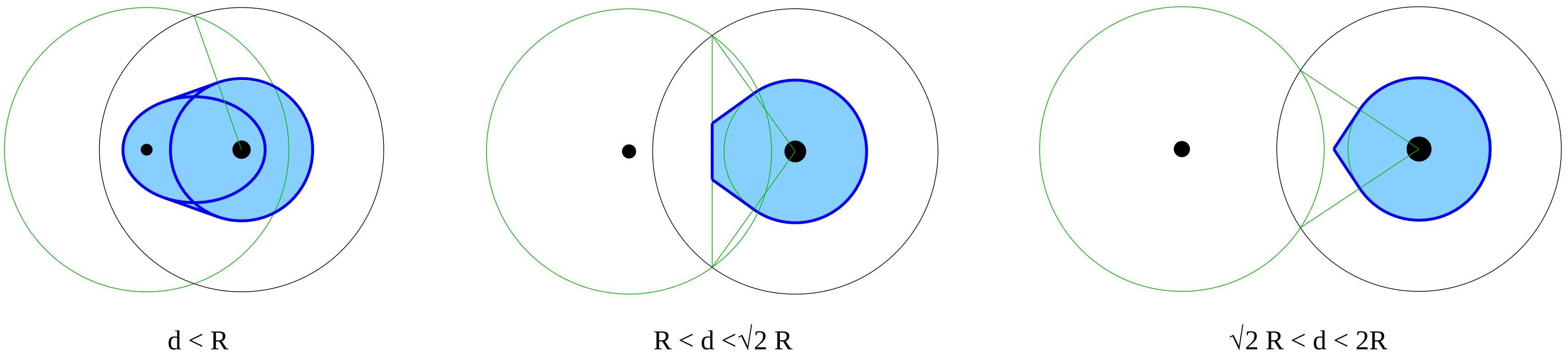
	k_t	Aac/Cam	cone
Passive	 πR^2	 πR^2	 πR^2
Active	 $\frac{\mathcal{A}_{\text{hard}}}{\pi R^2} \approx 0.812 \pm 0.277$ $\frac{\mathcal{A}_{\text{ghost}}}{\pi R^2} \approx 0.554 \pm 0.174$	 $\frac{\mathcal{A}_{\text{hard}}}{\pi R^2} \approx 0.814 \pm 0.$ $\frac{\mathcal{A}_{\text{ghost}}}{\pi R^2} \approx 0.551 \pm 0.$	 $\mathcal{A}_{\text{ghost}}$ depends on f possible monster jets!

[Passive area](): 1 hard particle + 1 soft



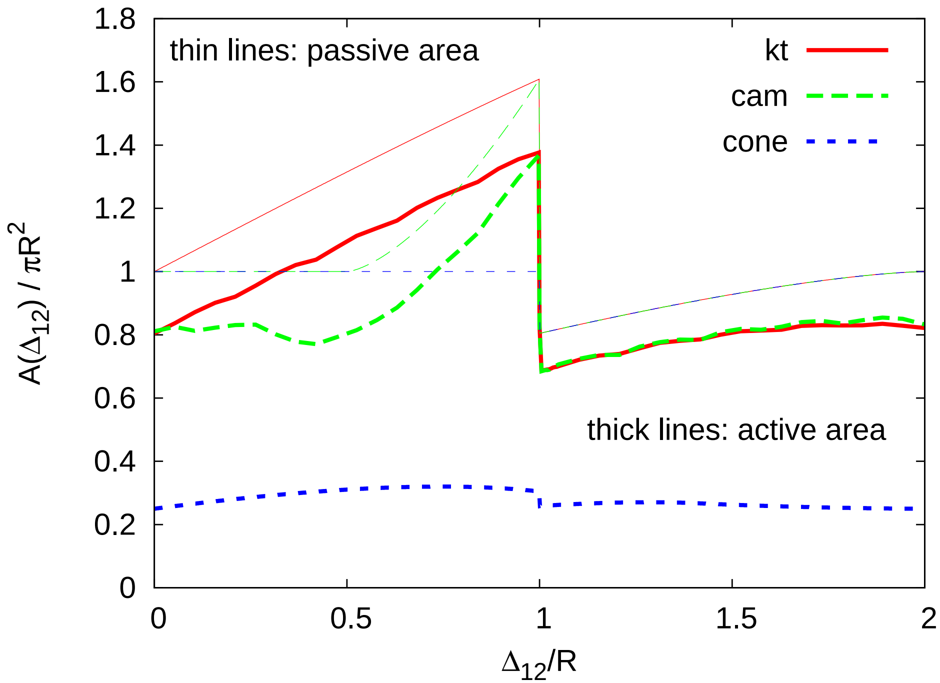
2-particle cases

Active area: 1 hard particle + 1 soft: analytic result for cone only



Alltogether, we have:

- Area $\neq$ cst. πR^2
- Δ_{12} dependence under control



Area scaling violations

QCD probability of emitting a small-angle soft gluon:

$$\frac{dP}{d\Delta_{12} dp_{t,2}} = C_{F,A} \frac{2\alpha_s}{\pi} \frac{1}{\Delta_{12}} \frac{1}{p_{t,2}}$$

Hence the average area is

$$\langle \mathcal{A}(p_{t,1}, R) \rangle = \mathcal{A}_{\text{1hard}}(R) + \int d\Delta\, dp_{t,2} \frac{dP}{d\Delta_{12} dp_{t,2}} \left[\mathcal{A}_{\text{hard+1 soft}}(\Delta, R) - \pi R^2\right]$$

Area scaling violations

QCD probability of emitting a small-angle soft gluon:

$$\frac{dP}{d\Delta_{12}dp_{t,2}} = C_{F,A}\frac{2\alpha_s}{\pi}\frac{1}{\Delta_{12}}\frac{1}{p_{t,2}}$$

Hence the average area is

$$\begin{aligned}\langle\mathcal{A}(p_{t,1},R)\rangle &= \mathcal{A}_{\text{1hard}}(R) + \int d\Delta\, dp_{t,2}\, \frac{dP}{d\Delta_{12}dp_{t,2}}\left[\mathcal{A}_{\text{hard+1 soft}}(\Delta,R) - \pi R^2\right] \\ &= \frac{C_{F,A}}{\pi b_0}\log\left(\frac{\alpha_s(Q_0)}{\alpha_s(Rp_t)}\right)\pi R^2\, d\end{aligned}$$

- Scaling viloation

Area scaling violations

QCD probability of emitting a small-angle soft gluon:

$$\frac{dP}{d\Delta_{12} dp_{t,2}} = C_{F,A} \frac{2\alpha_s}{\pi} \frac{1}{\Delta_{12}} \frac{1}{p_{t,2}}$$

Hence the average area is

$$\begin{aligned} \langle \mathcal{A}(p_{t,1}, R) \rangle &= \mathcal{A}_{1\text{hard}}(R) + \int d\Delta \, dp_{t,2} \, \frac{dP}{d\Delta_{12} dp_{t,2}} \left[\mathcal{A}_{\text{hard+1 soft}}(\Delta, R) - \pi R^2\right] \\ &= \frac{C_{F,A}}{\pi b_0} \log\left(\frac{\alpha_s(Q_0)}{\alpha_s(R p_t)}\right) \pi R^2 \, d \end{aligned}$$

- Scaling viloation
- gluon > quark

Area scaling violations

QCD probability of emitting a small-angle soft gluon:

$$\frac{dP}{d\Delta_{12}dp_{t,2}} = C_{F,A}\frac{2\alpha_s}{\pi}\frac{1}{\Delta_{12}}\frac{1}{p_{t,2}}$$

Hence the average area is

$$\begin{aligned}\langle\mathcal{A}(p_{t,1},R)\rangle &= \mathcal{A}_{1\text{hard}}(R) + \int d\Delta\, dp_{t,2}\,\frac{dP}{d\Delta_{12}dp_{t,2}}\left[\mathcal{A}_{\text{hard+1 soft}}(\Delta,R) - \pi R^2\right]\\ &= \frac{C_{F,A}}{\pi b_0}\log\left(\frac{\alpha_s(Q_0)}{\alpha_s(Rp_t)}\right)\pi R^2\, d\end{aligned}$$

d	passive	active
k_t	0.5638	0.519
Cam	0.07918	0.0865
SISCone	-0.06378	0.1246
anti-k_t	0	0

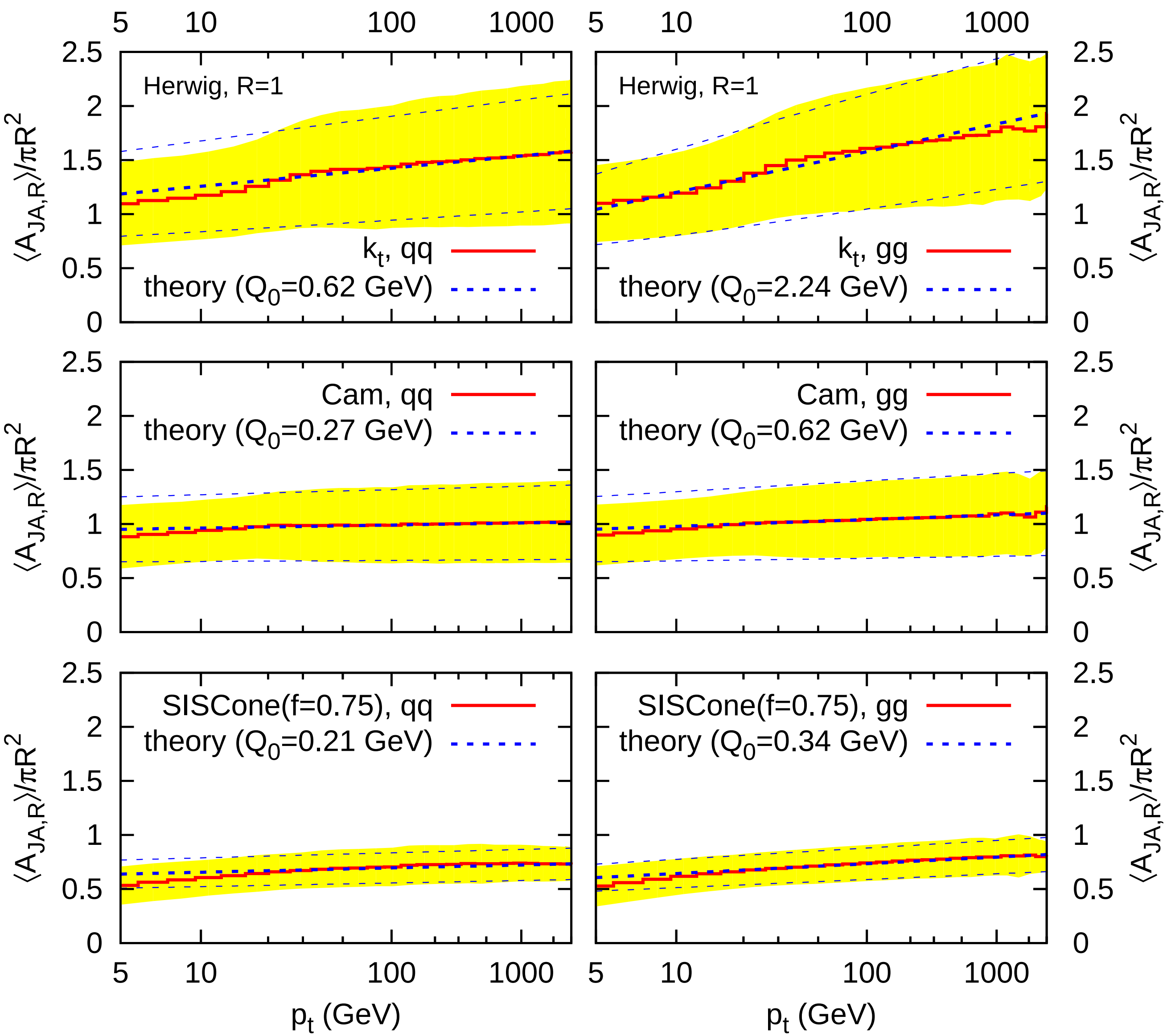
- Scaling viloation
- gluon > quark
- with know LO anomalous dimension

“Real-life” anomalous dimension

Herwig simulations:

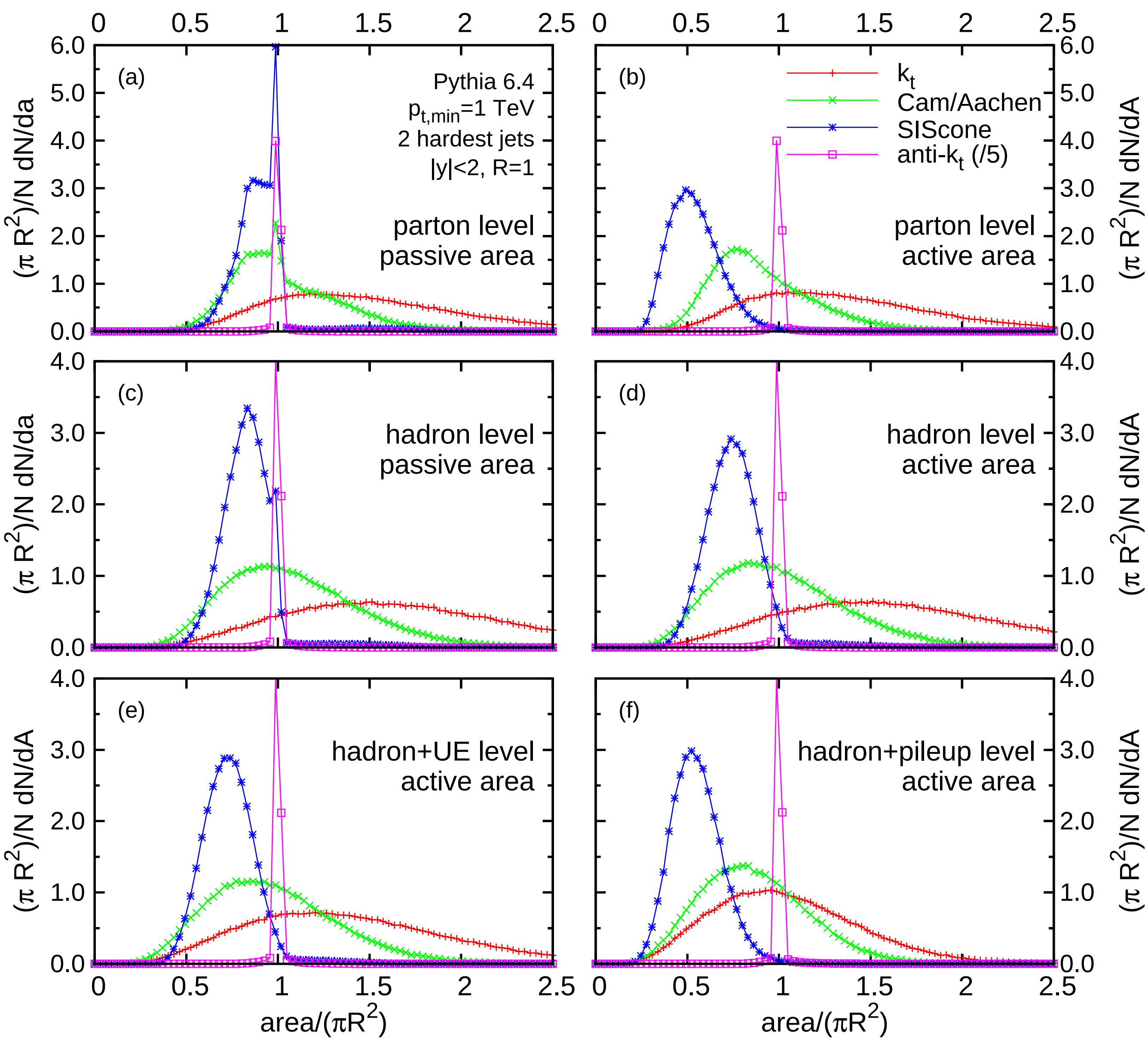
at hadron+UE level:

area vs. p_t of the jet



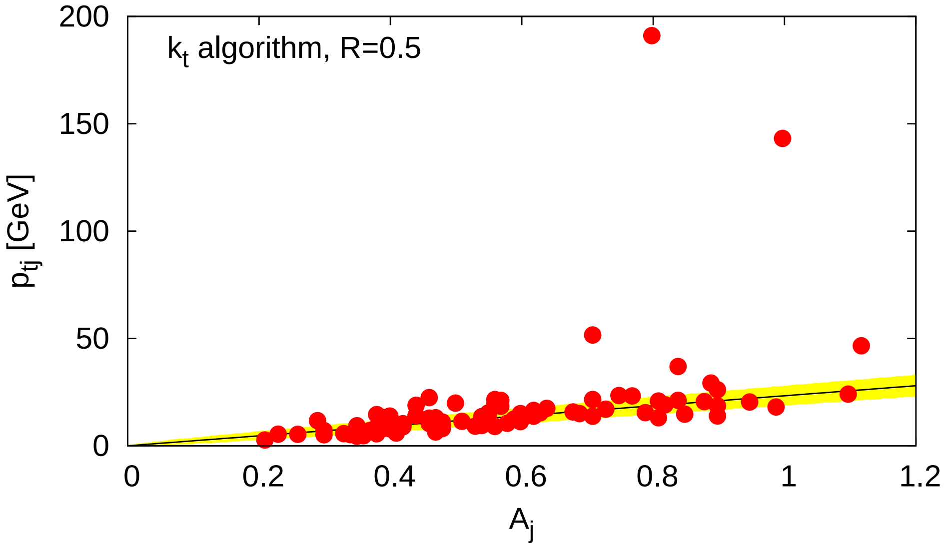
- good agreement with LO predictions
- for flucts. too
- k_t bigger $\Rightarrow$ NLO?

Area histograms



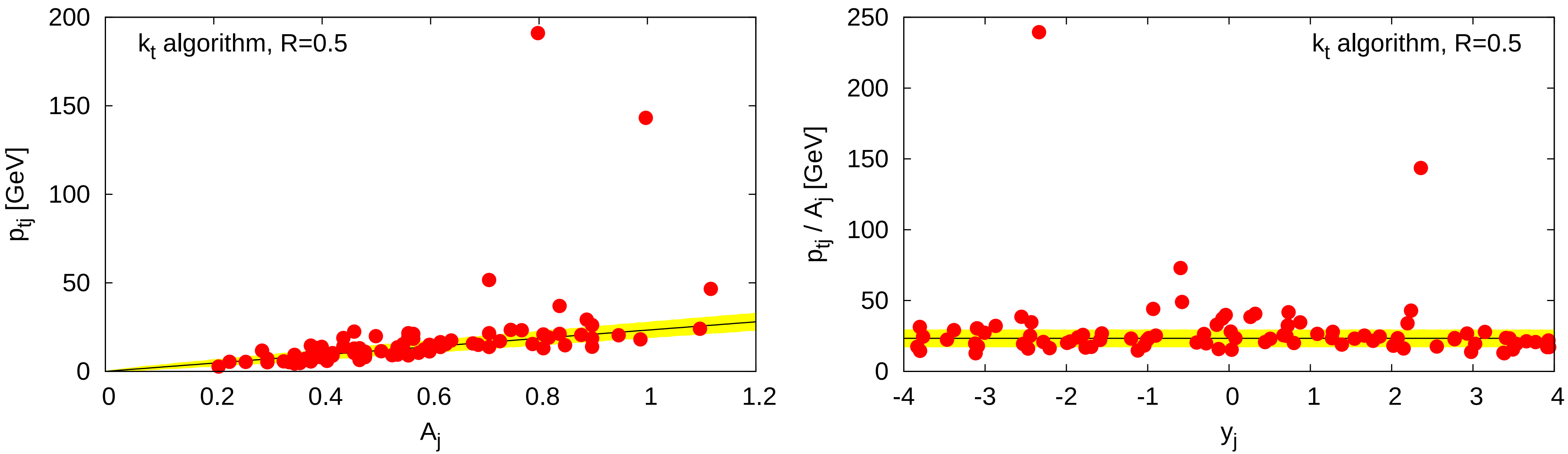
What can area be used for?

Dense event with pile-up (or uniform soft background):



What can area be used for?

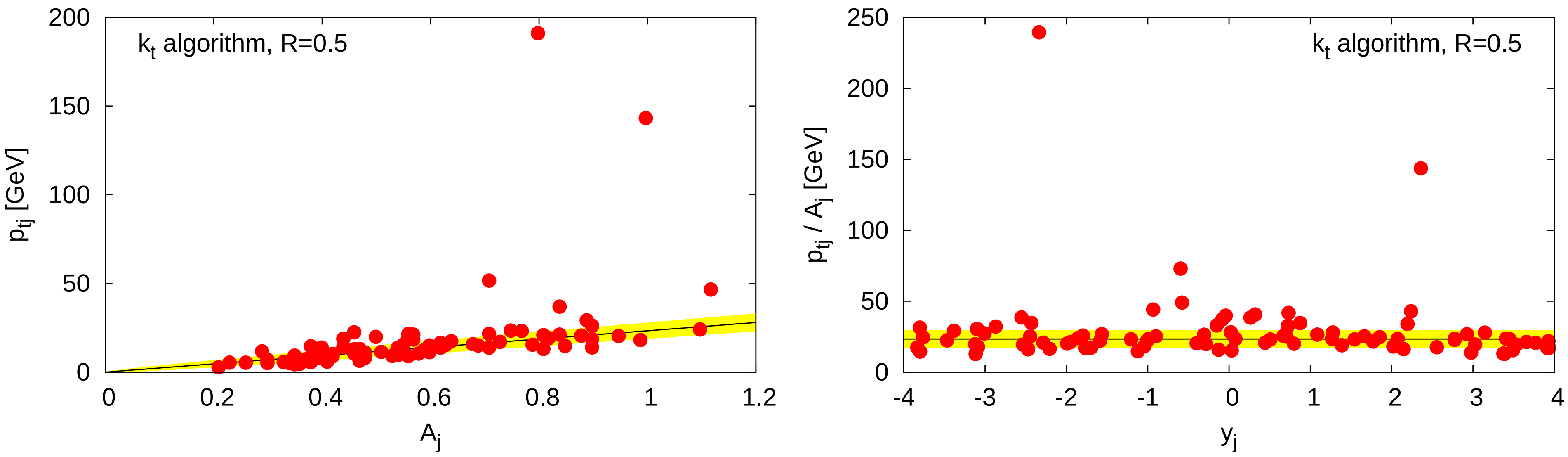
Dense event with pile-up (or uniform soft background):



- For pure pile-up jets: Area $\propto p_t$ of the jet
- p_t/area is constant $\rightarrow \rho$ = median p_t/area

What can area be used for?

Dense event with pile-up (or uniform soft background):



- For pure pile-up jets: Area $\propto p_t$ of the jet
- p_t/area is constant $\rightarrow \rho$ = median p_t/area

Area can be used to subtract pileup:

$$p_{t,\mathrm{corrected}} = p_{t,\mathrm{bare}} - \rho \; \mathrm{area}$$

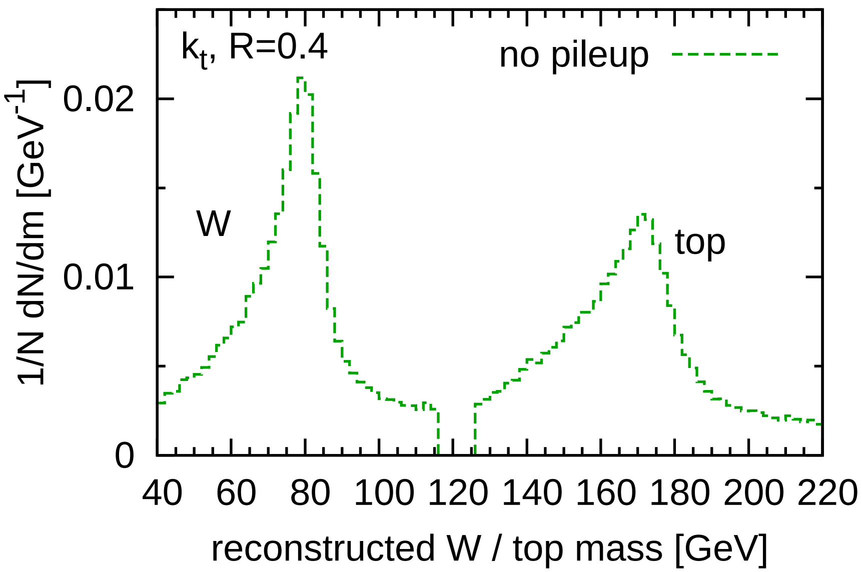
FastJet *implementation*

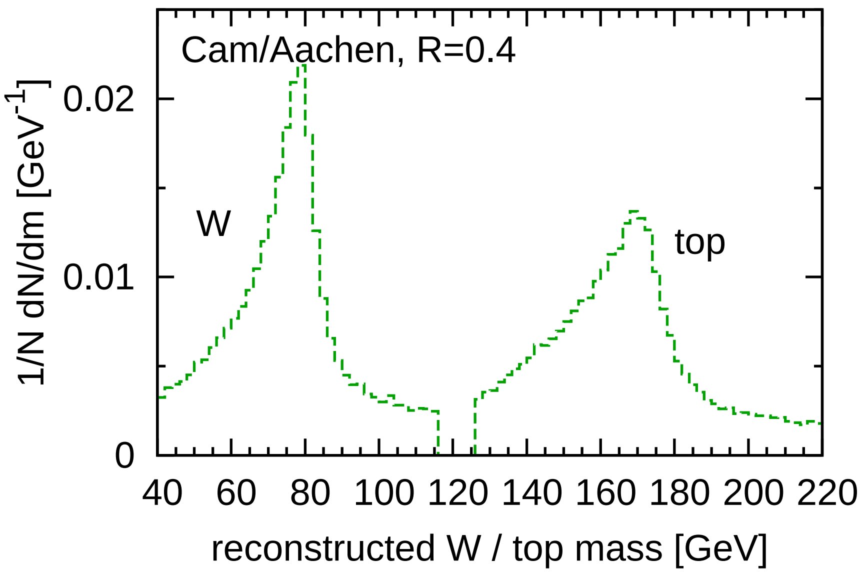
Implementation in FastJet using the AreaDefinition and sf ClusterSequenceArea classes

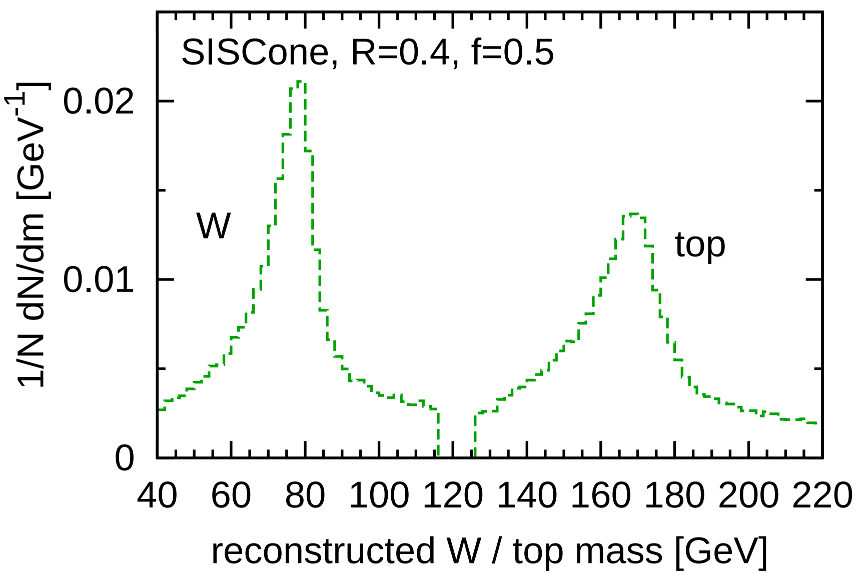
```
// declare usual FastJet tools: particles and jet definition
vector<fastjet::PseudoJet> particles;
fastjet::JetDefinition jet_def(kt_algorithm, R);

// define the type of area you want
fastjet::GhostedAreaSpec area_spec(maxrap_ghost, num_repeat, ghost area);
fastjet::AreaDefinition area_def(active_area, area_spec);

// perform the clustering with area computation
fastjet::ClusterSequenceArea clust_seq(particles, jet_def, area_def);

// get the median background per unit area (i.e. rho)
double rho = clust_seq.median_pt_per_unit_area_4vector(maxrap_ghost);

// retrieve the jets and do the subtraction
vector<fastjet::PseudoJet> jets = sorted_by_pt(clust_seq.inclusive_jets(ptmin))
fastjet::PseudoJet jet_sub = jets[0] - rho * clust_seq.area_4vector(jets[0]);
```

Subtraction in action

$t\bar{t} + W \qquad (t\bar{t} \to \ell^+ \nu_\ell b + q\bar{q}\bar{b}) \qquad (W \to q\bar{q})$





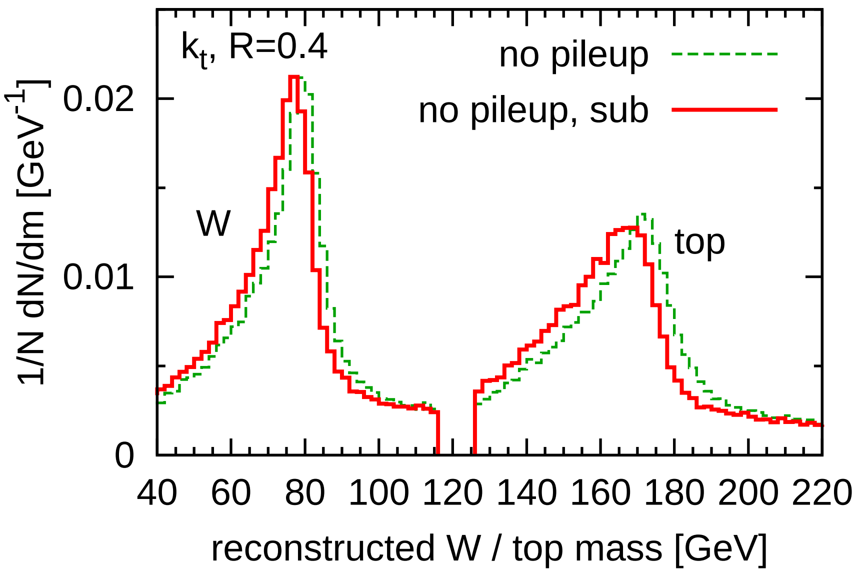


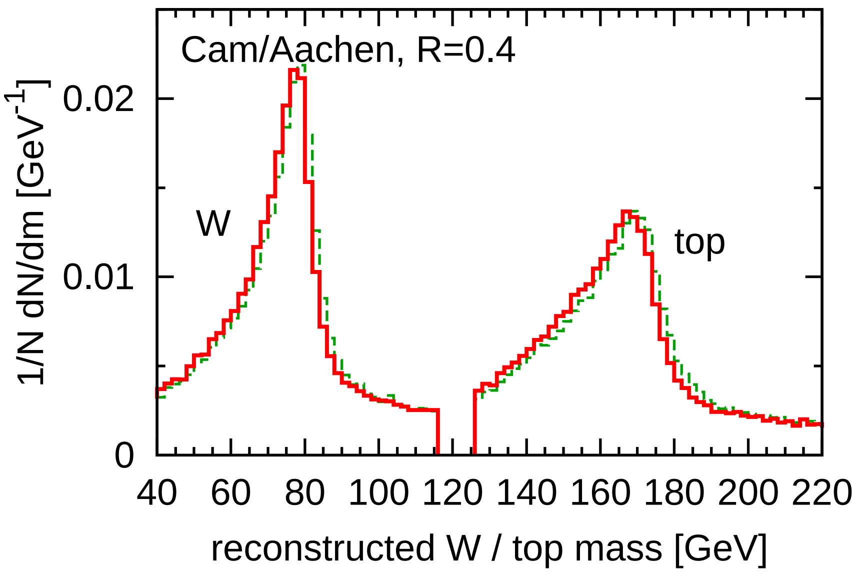
LHC at high lumi

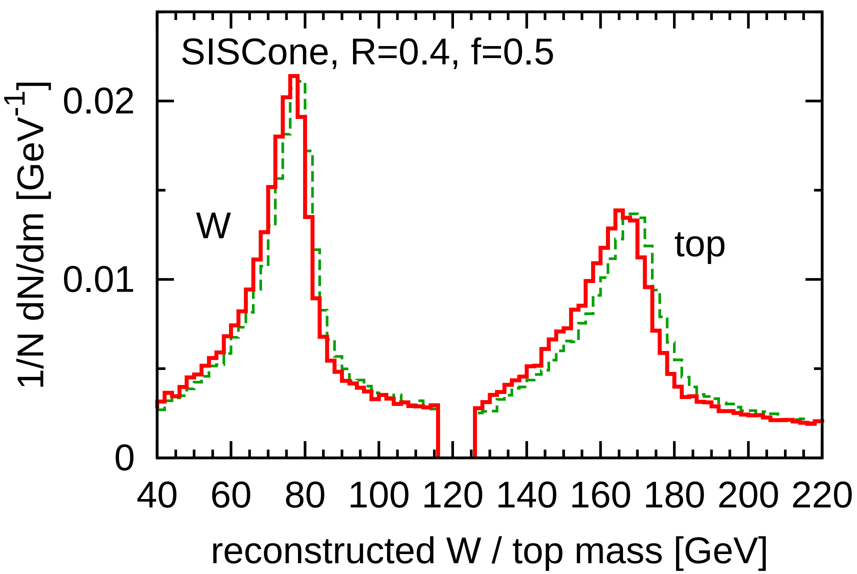
no pileup $\Rightarrow$ good result

Subtraction in action

$t\bar{t} + W$ $(t\bar{t} \to \ell^+ \nu_\ell b + q\bar{q}\bar{b})$ $(W \to q\bar{q})$







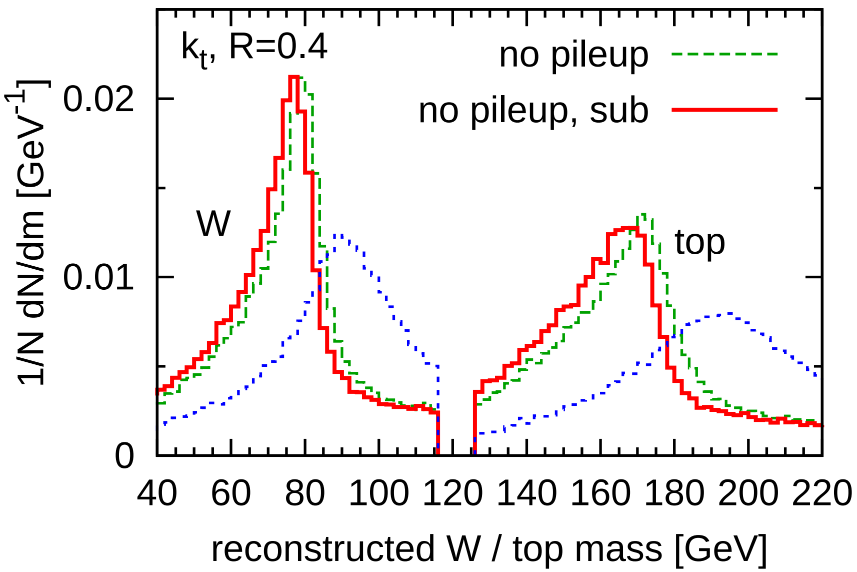
LHC at high lumi

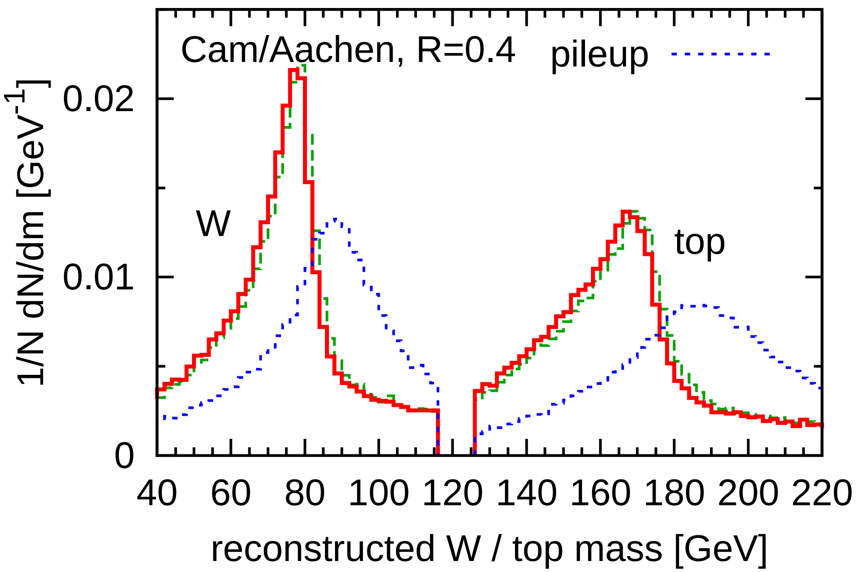
no pileup $\Rightarrow$ good result

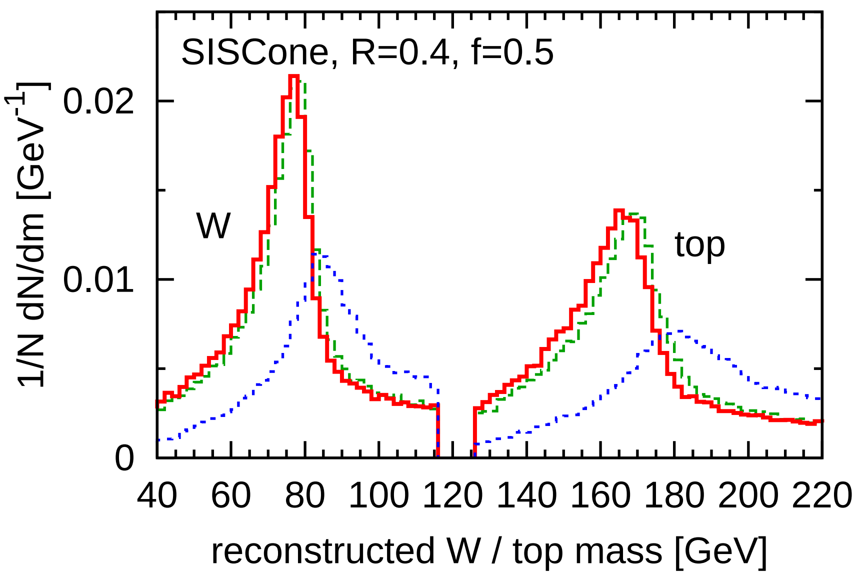
$\Rightarrow$ no subtraction effect

Subtraction in action

$t\bar{t} + W$ $\quad (t\bar{t} \to \ell^+\nu_\ell b + q\bar{q}\bar{b})$ $\quad (W \to q\bar{q})$







LHC at high lumi

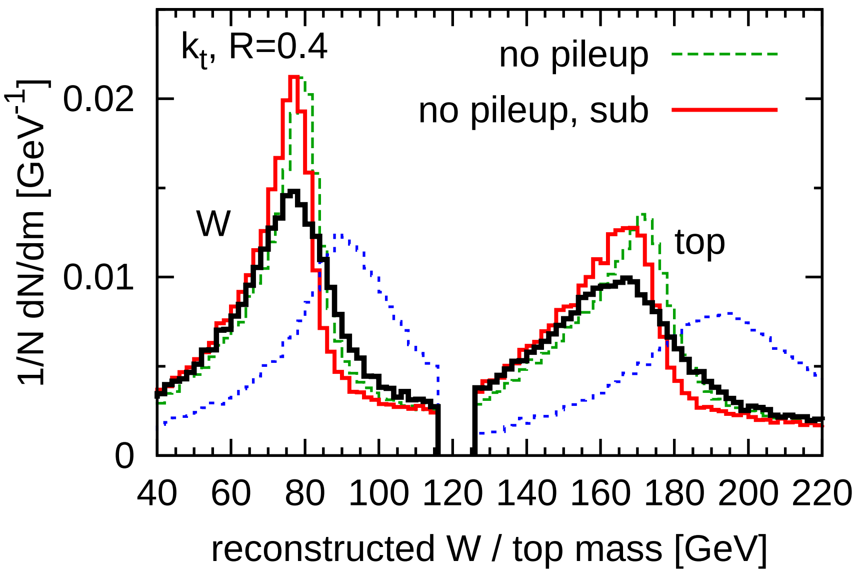
no pileup $\Rightarrow$ good result

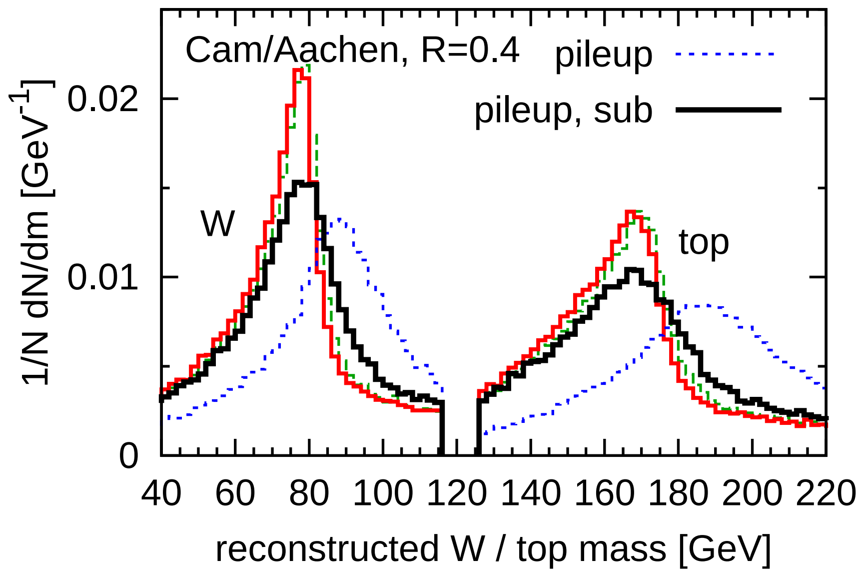
$\Rightarrow$ no subtraction effect

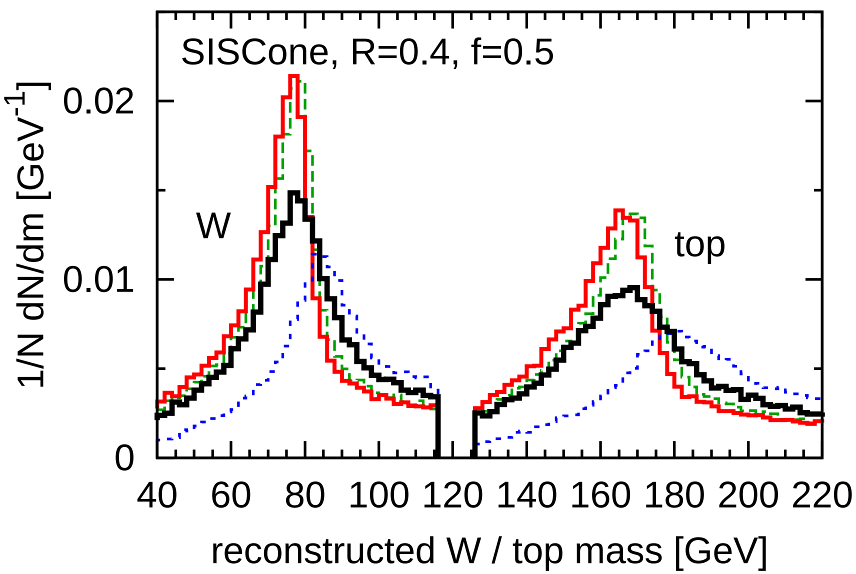
pileup $\Rightarrow$ poor result

Subtraction in action

$t\bar{t} + W$ $\quad (t\bar{t} \to \ell^+\nu_\ell b + q\bar{q}\bar{b}) \quad (W \to q\bar{q})$







LHC at high lumi

no pileup $\Rightarrow$ good result

$\Rightarrow$ no subtraction effect

pileup $\Rightarrow$ poor result

$\Rightarrow$ subtraction works

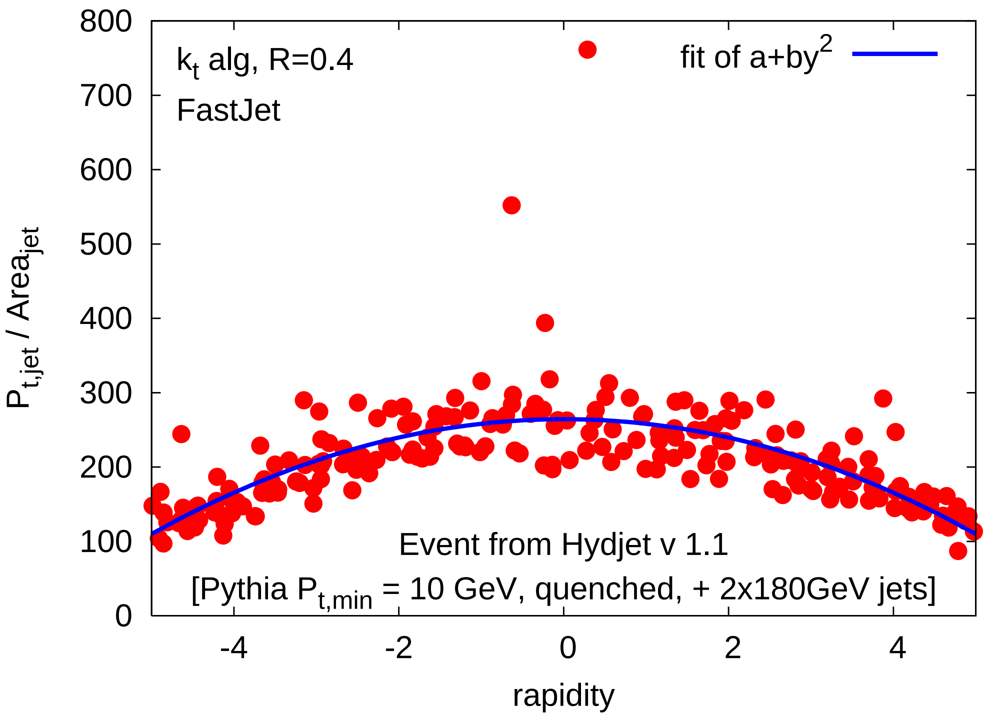
Subtraction in Heavy Ions

For heavy-ion collisions at LHC:

- Large background (~ 250 GeV/unit area)
- Not really uniform

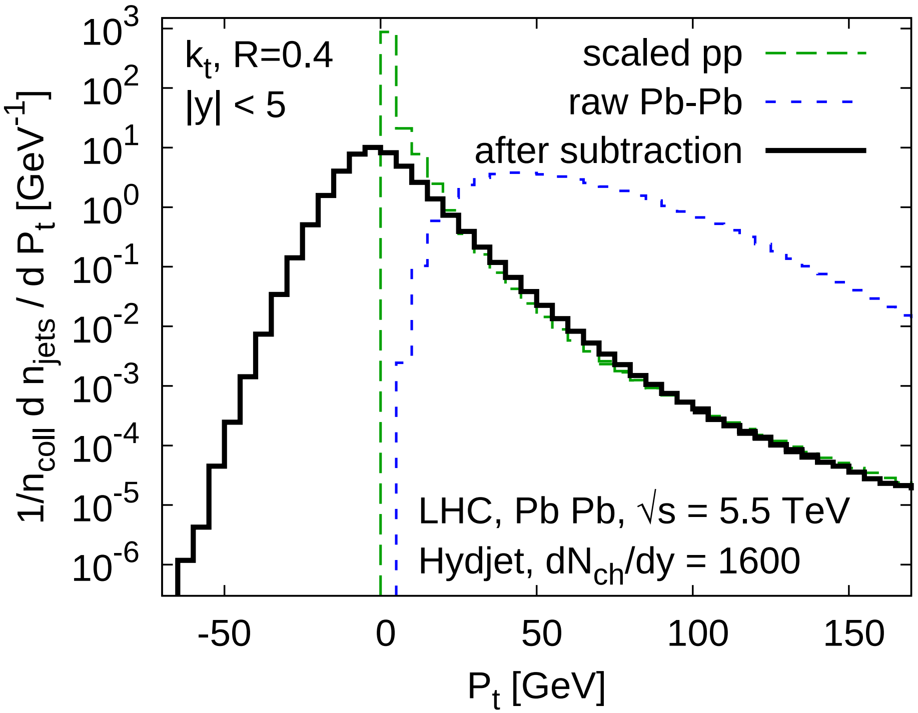
For heavy-ion collisions at LHC:

- Large background (~ 250 GeV/unit area)
- Not really uniform
- Background: p_t/area $\approx$ parabolic in rapidity



For heavy-ion collisions at LHC:

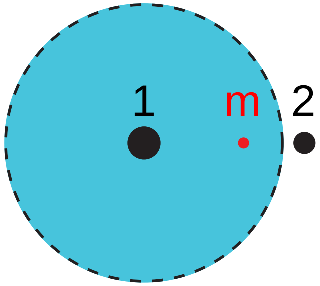
- Large background (~ 250 GeV/unit area)
- Not really uniform
- Background: p_t/area $\approx$ parabolic in rapidity
- After subtraction:



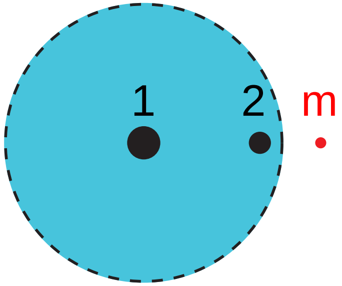
Back-reaction

Back reaction: when adding soft background to a hard event some hard parton can be associated to another hard jet.

- Point-like or diffuse background
- gain: p_2 gained when adding p_1



- loss: p_2 lost when adding p_1

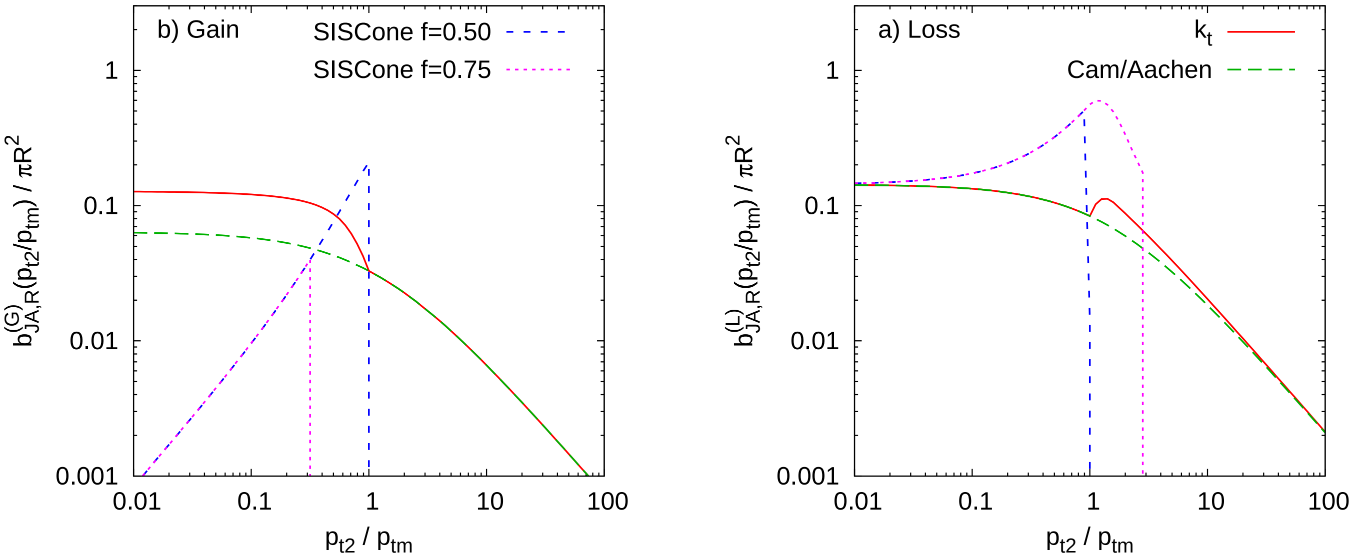


Back-reaction (2)

Under analytic control!

- Probability of gain/loss ($\rho \equiv$ background density)

$$\frac{dP^{(G,L)}}{dp_{t,2}} = \frac{2\alpha_s C_{F,A}}{\pi} \frac{1}{p_{t,2}} b^{(G,L)}(p_{t,2}/\rho)$$



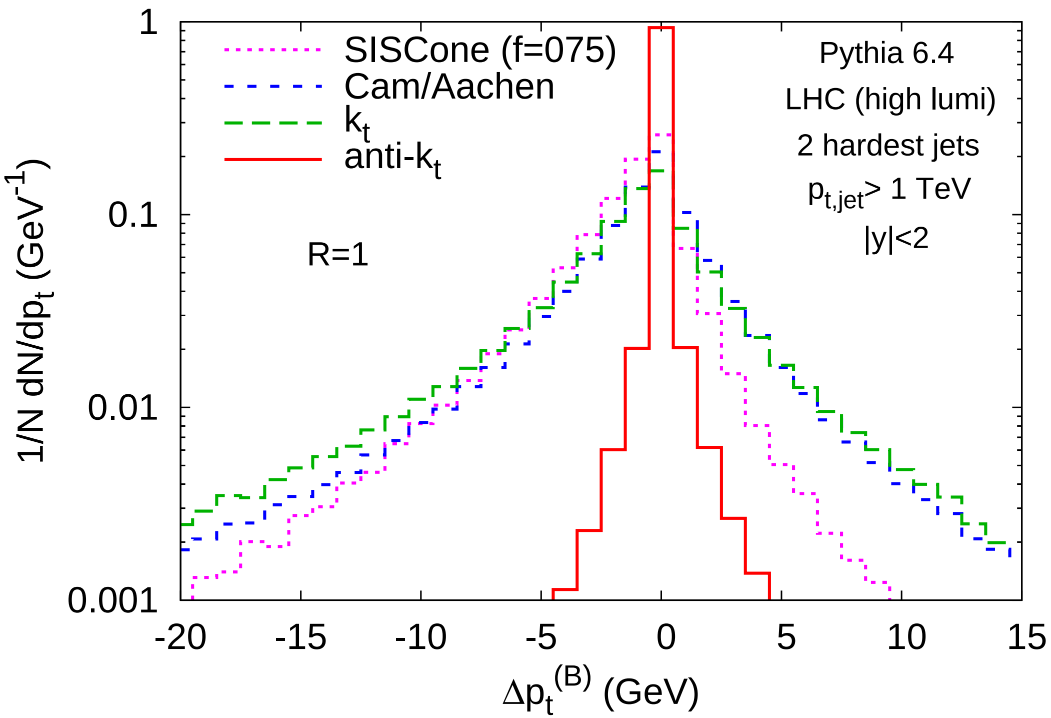
$$p_{t,m} \ll p_{t,2} \ll p_{t,1} : \quad \begin{cases} \propto p_{t,m}/p_{t,2} & \text{for } k_t, \text{Cam} \\ \approx 0 & \text{for anti}-k_t, \text{SISCone} \end{cases}$$

Under analytic control!

- Probability of gain/loss ($\rho \equiv$ background density)

$$\frac{dP^{(G,L)}}{dp_{t,2}} = \frac{2\alpha_s C_{F,A}}{\pi} \frac{1}{p_{t,2}} b^{(G,L)}(p_{t,2}/\rho)$$

- Shift in p_t due to back-reaction: $\Delta p_t^{(G-L)}$



anti-k_t $\ll$ SISCone < k_t, Cam

Back-reaction (2)

Under analytic control!

- Probability of gain/loss ($\rho \equiv$ background density)

$$\frac{dP^{(G,L)}}{dp_{t,2}} = \frac{2\alpha_s C_{F,A}}{\pi} \frac{1}{p_{t,2}} b^{(G,L)}(p_{t,2}/\rho)$$

- Shift in p_t due to back-reaction: $\Delta p_t^{(G-L)}$

$$\left\langle \Delta p_t^{(G-L)} \right\rangle = \mathcal{B} \cdot \rho \cdot \frac{C_{F,A}}{\pi b_0} \cdot \log\left(\frac{\alpha_s(\rho R^3)}{\alpha_s(p_{t,1} R)}\right)$$

$\mathcal{B}/\pi R^2$	pointike	diffuse
k_t	$\sqrt{3}/4 \approx 0.14$	≈ 0.10
Cam	$\sqrt{3}/4 \approx 0.14$	≈ 0.10
SISCone	0 (+NLO)	0 (+NLO)
anti-k_t	0 (+power corr.)	0 (+power corr.)