

# *Area-median background subtraction*

Grégory Soyez

IPhT, CEA Saclay

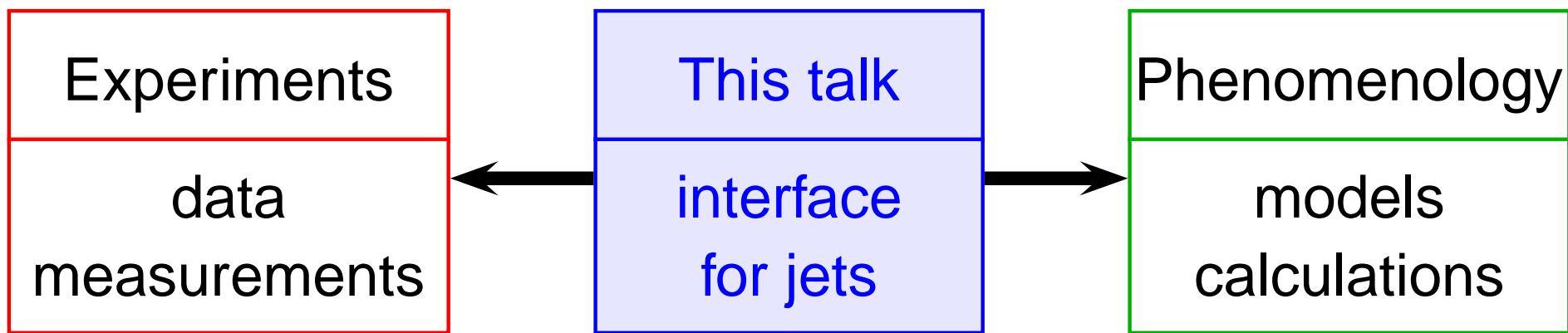
with Matteo Cacciari, Gavin Salam

Paris Jet workshop — July 3rd 2013

# *Generic idea behind this talk*



# *Generic idea behind this talk*



should be  
reproducible  
& pQCD safe

public  
standard  
documented

should be  
free of  
biases

# *FastJet framework*



# *FastJet (in a nutshell)*

[M.Cacciari, G.Salam, GS, 2007-2013]

- Initially fast  $k_t$  (& recombination algs.)
- Grown way beyond that:
  - plugins for most other jet definitions
  - jet areas and background subtraction (see below)
  - tools for jet substructure
  - more to come...
- (standard) framework for jet manipulation for both theorists and experimentalists
- FastJet 3.0.4 released in June 2013  
see [www.fastjet.fr](http://www.fastjet.fr)

# *A practical “warming-up” example*

---

example/01-basic.cc

3 fundamental objects of FastJet:

- PseudoJet: 4-momentum
- JetDefinition: the algorithm and its parameters (or a plugin)
- ClusterSequence: handles the clustering

# FastJet contrib (New: Feb 2013)

- fastjet.fr
- fastjet-contrib
- contrib svn

## FastJet Contrib

The fastjet-contrib space is intended to provide a common location for access to 3rd party extensions of FastJet.

**Download** the current version: [fjcontrib-1.003](#) (released 1 May 2013), which contains [these contributions](#). Changes relative to earlier versions are briefly described in the [NEWS](#) file.

Package	Version	Information
GenericSubtractor	1.1.0	README NEWS
JetFFMoments	1.0.0	README NEWS
VariableR	1.0.1	README NEWS
Nsubjettiness	1.0.2	README NEWS
EnergyCorrelator	1.0.1	README NEWS

- a quick and uniform access to 3rd-party code
- contributors are welcome (please contact us)

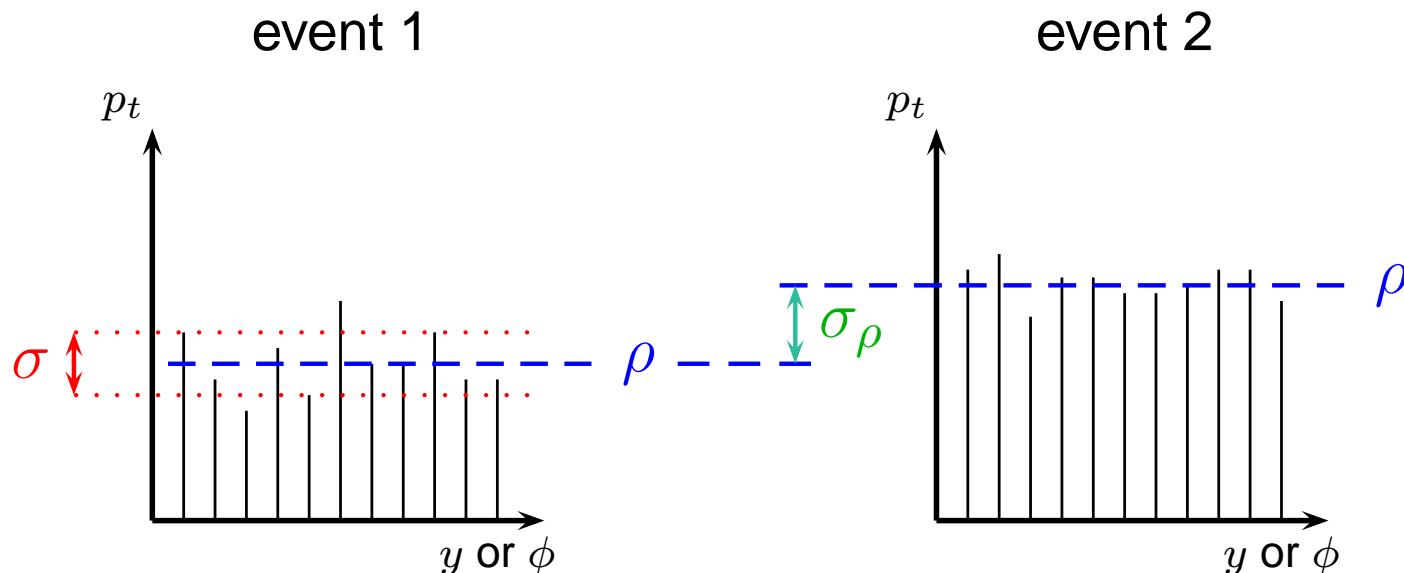


# *Jets in soft background preamble/basic considerations*

# Basic characterisation

Background mostly characterised by 3 numbers (\*):

- $\rho$ : the average activity in an event (per unit area)
- $\sigma$ : the intra-event fluctuations (per unit area)
- $\sigma_\rho$ : the event-to-event fluctuations of  $\rho$



# *Basic characterisation*

Background mostly characterised by 3 numbers (\*):

- $\rho$ : the average activity in an event (per unit area)
- $\sigma$ : the intra-event fluctuations (per unit area)
- $\sigma_\rho$ : the event-to-event fluctuations of  $\rho$

Jet of momentum  $p_t$  and area  $A$ :

one event:  $p_t \rightarrow p_t + \rho A \pm \sigma \sqrt{A}$

event average:  $p_t \rightarrow p_t + \langle \rho \rangle A \pm \sigma_\rho A \pm \sigma \sqrt{A}$

# Basic characterisation

Background mostly characterised by 3 numbers (\*):

- $\rho$ : the average activity in an event (per unit area)
- $\sigma$ : the intra-event fluctuations (per unit area)
- $\sigma_\rho$ : the event-to-event fluctuations of  $\rho$

Jet of momentum  $p_t$  and area  $A$ :

$$\text{one event: } p_t \rightarrow p_t + \boxed{\rho A} \pm \boxed{\sigma \sqrt{A}}$$

$$\text{event average: } p_t \rightarrow p_t + \boxed{\langle \rho \rangle A} \pm \boxed{\sigma_\rho A \pm \sigma \sqrt{A}}$$

$p_t$  shift

$p_t$  smearing  
resolution degradation

correct event-by-event reduces smearing

# The clock is ticking

HYDJET simulations		$\rho$ (GeV) ( $y=0, 0-10\%$ )	$\sigma$ (GeV)	$\sigma_\rho$ (GeV)	$\sigma_{\text{jet}}$ (GeV) (anti-kt, $R=0.4$ )
LHC 2.76 TeV	all	250	18	36	16
	charged only	147	12.5	22	11.3
Data LHC 2.76 TeV		$\rho$ (GeV) ( $y=0, 0-10\%$ )	$\sigma$ (GeV)	$\sigma_\rho$ (GeV)	$\sigma_{\text{jet}}$ (GeV) (anti-kt, $R=0.4$ )
ALICE, charged only I201.2423		138		18.5	11.2
CMS I205.0206					5.2 ( $R=0.3 + \text{NR}$ )
ATLAS I208.1967					12.5

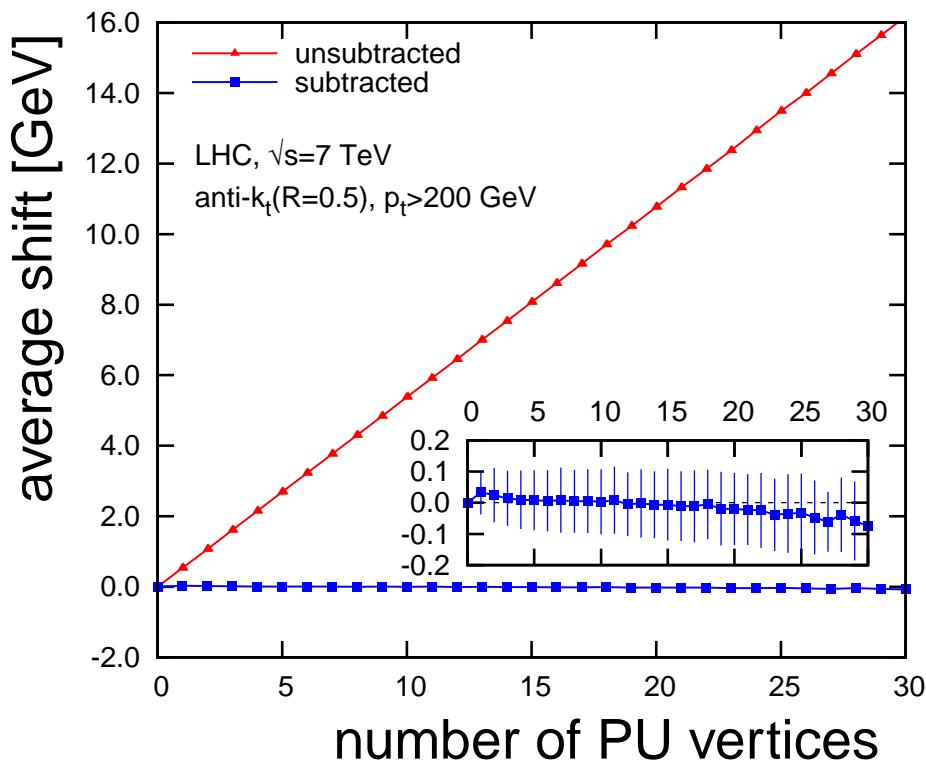
Only background-induced component, no calorimeter effects

It would be nice to fill that table during this workshop

## *Area-median subtraction*

# Achievements illustration (PU)

average  $p_t$  shift



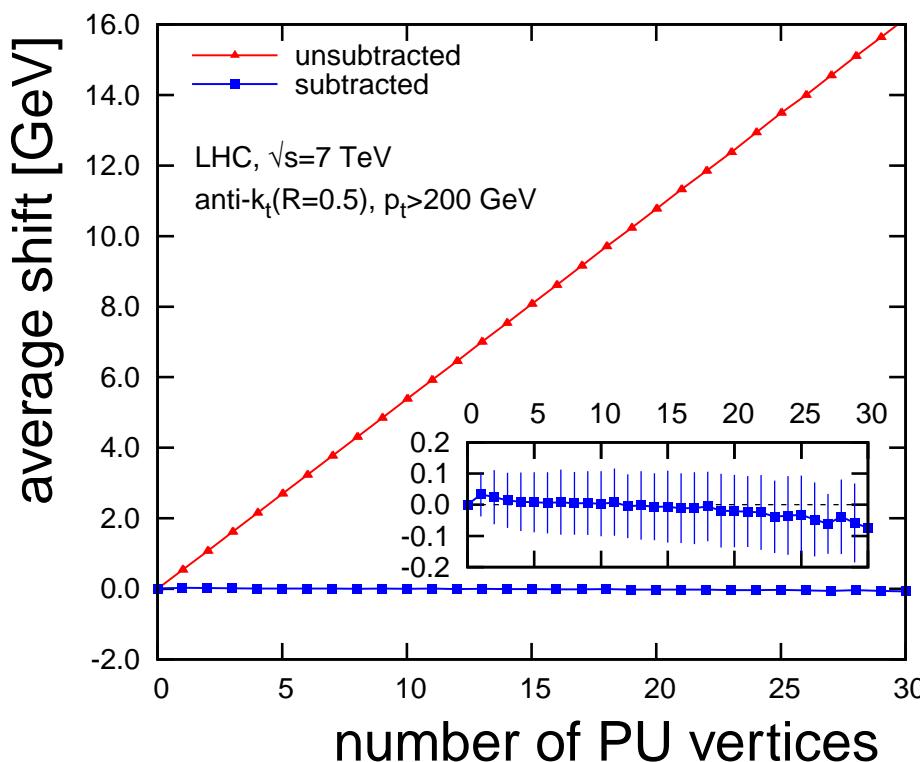
No subtraction

area-median subtraction

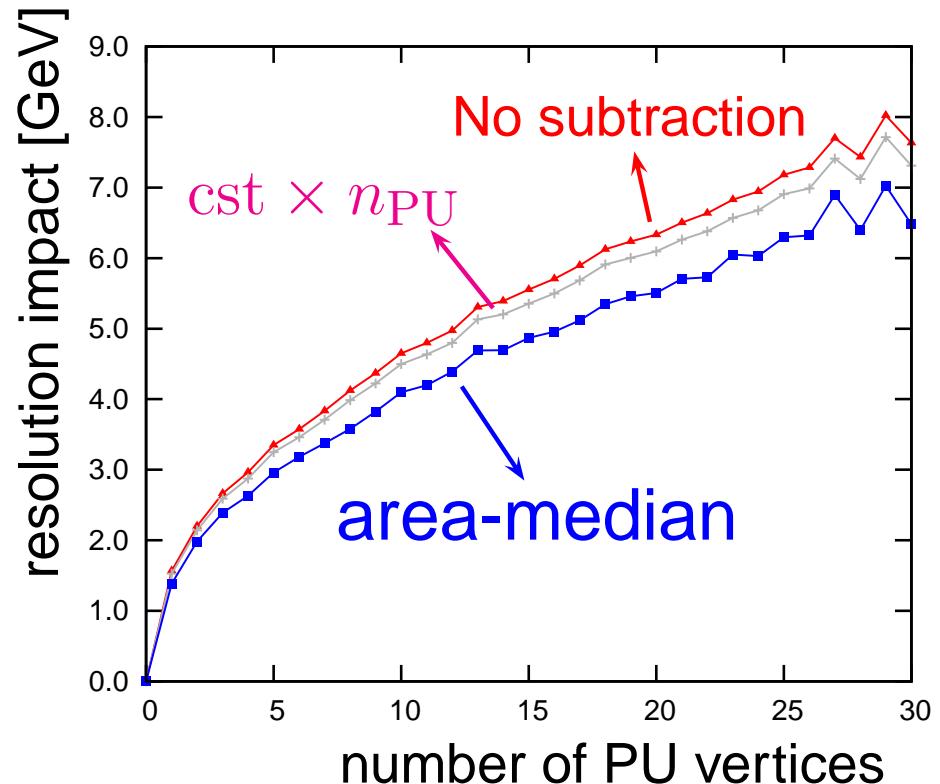
corrected for shift

# Achievements illustration (PU)

average  $p_t$  shift



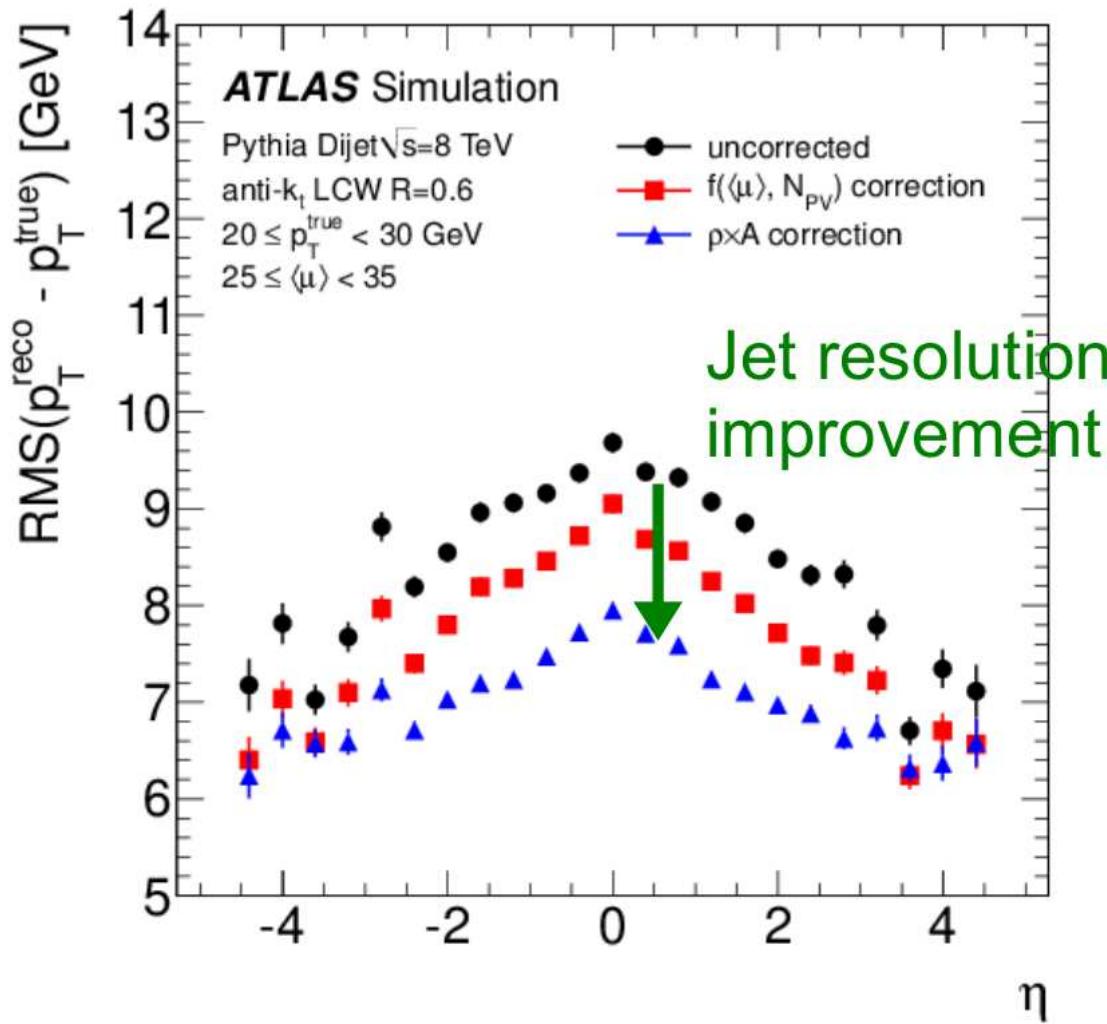
impact on resolution



corrected for shift

resolution improved

# PU subtraction as seen by ATLAS



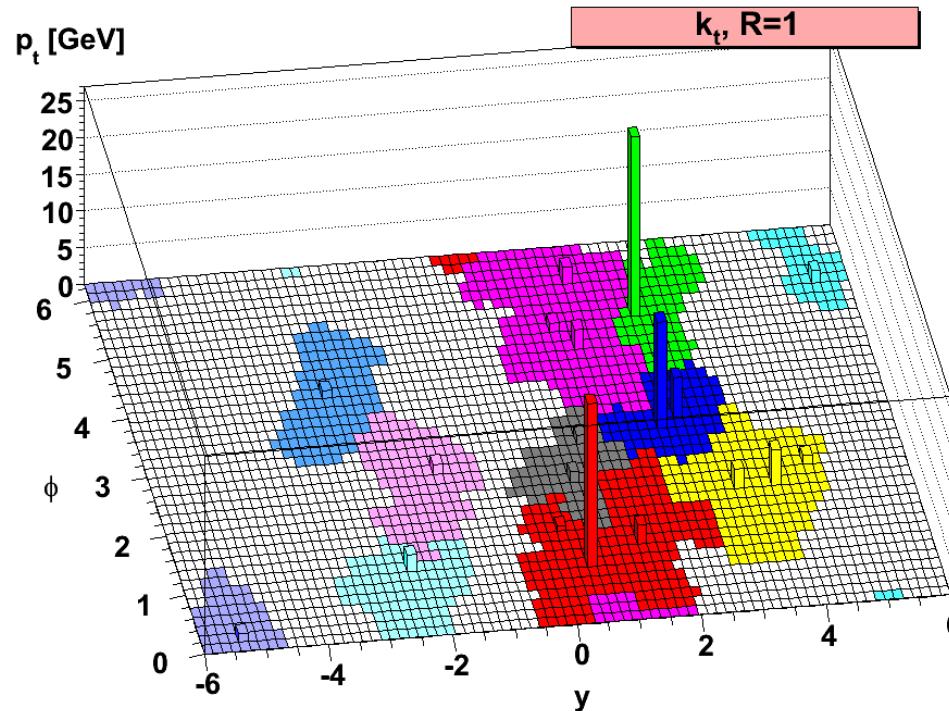
[B. Petersen, ATLAS Status report for the LHCC, 2013]

# Area-median subtraction

[M.Cacciari, G.P. Salam, 07; M.Cacciari, G.P. Salam, GS, 2008]

$$p_{t,\text{jet}}^{(\text{sub})} = p_{t,\text{jet}} - \rho_{\text{est}} A_{\text{jet}}$$

- jet area: available with jet clustering (per jet)



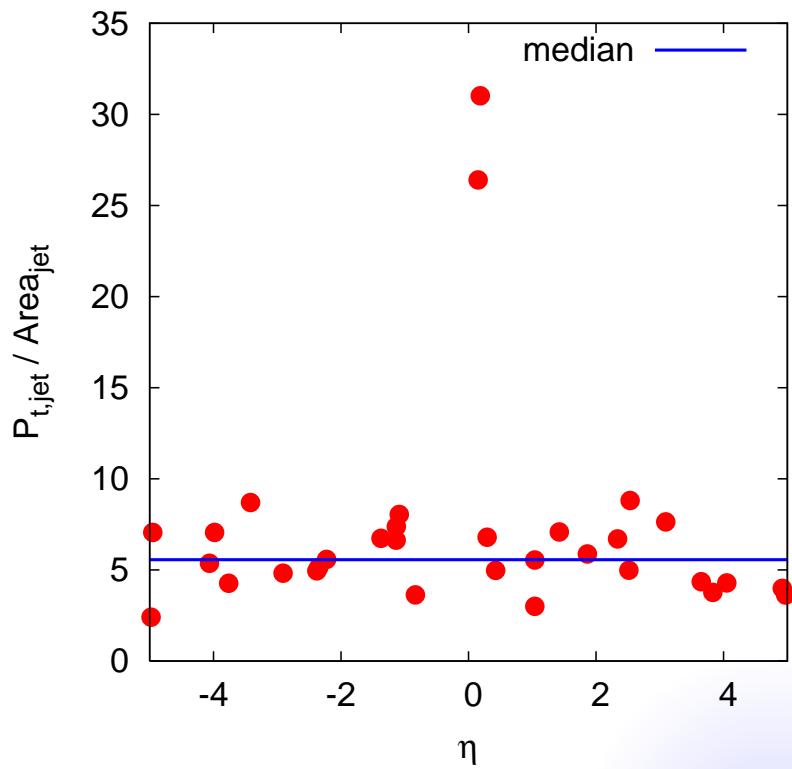
# Area-median subtraction

[M.Cacciari, G.P. Salam, 07; M.Cacciari, G.P. Salam, GS, 2008]

$$p_{t,\text{jet}}^{(\text{sub})} = p_{t,\text{jet}} - \rho_{\text{est}} A_{\text{jet}}$$

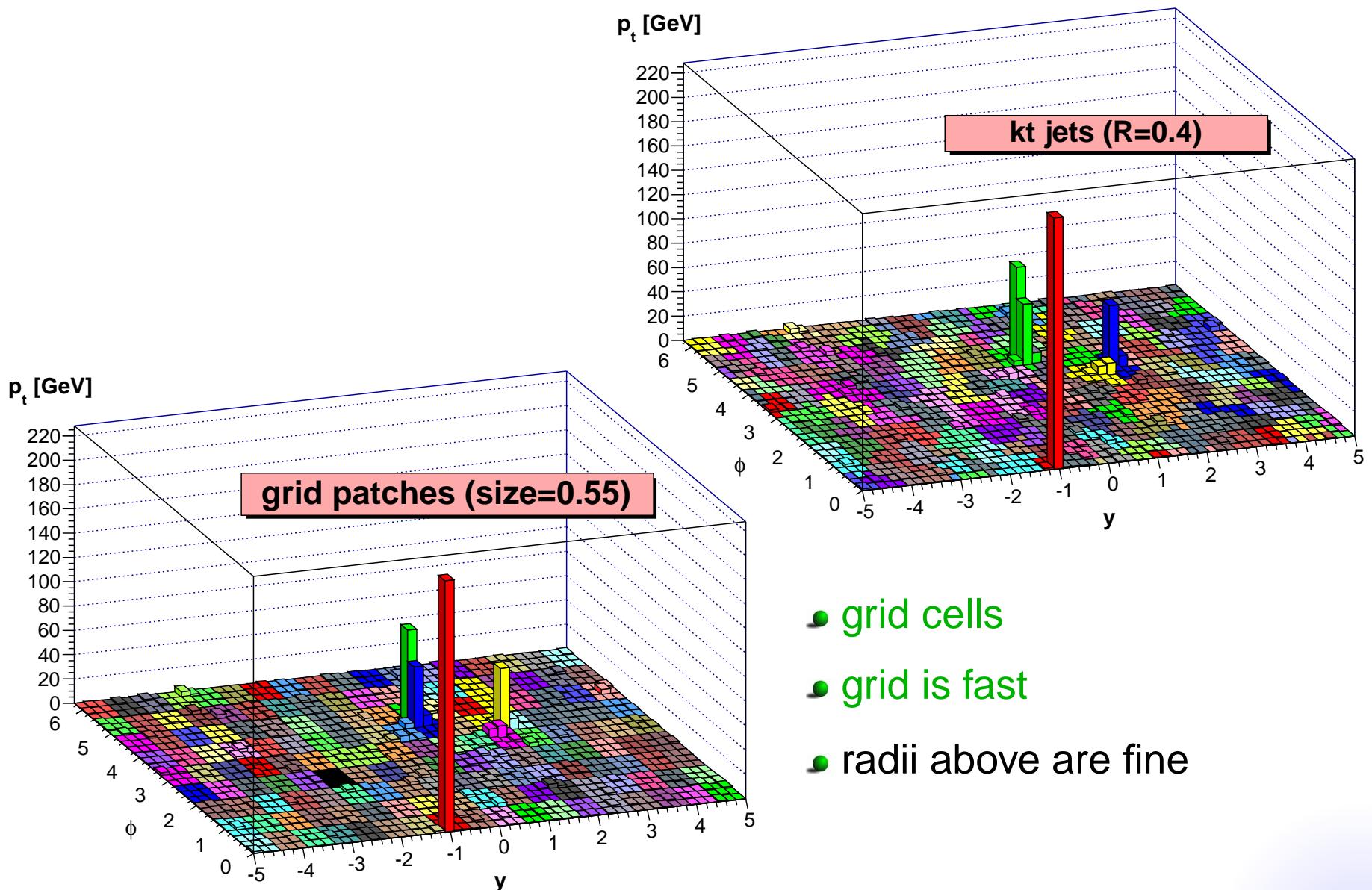
- jet area: available with jet clustering (per jet)
- $\rho_{\text{est}}$ , the background  $p_t$  density per unit area (per event)
  - break the event in patches of similar size  
e.g. cluster with  $k_t$
  - Estimate  $\rho_{\text{bkg}}$  using

$$\rho_{\text{est}} = \underset{j \in \text{patches}}{\text{median}} \left\{ \frac{p_{t,j}}{A_j} \right\}$$



# Patches of similar size

Patches are typically one of the 2 following types



# *Practical implementation (area)*

```
// where to place the ghosts
GhostedAreaSpec area_spec(ymax, nrepeat, ghost_area);

// how to compute areas
AreaDefinition area_def(active_area_explicit_ghosts,
                        area_spec);

// perform the clustering including area
ClusterSequenceArea cs(particles, jet_def, area_def);

// take (e.g.) the first jet
PseudoJet jet = cs.inclusive_jets()[0];
if (jet.has_area()){...}
jet.area();
jet.area_4vector();
jet.is_pure_ghost();
```

[see also example/06-areas.cc]

# *Practical implementation ( $\rho_{\text{est}}$ )*

## jets as patches

```
Selector rho_range; // patches acceptance
JetDefinition jet_def; // kt_algorithm, R = 0.4
AreaDefinition area_def; // see above
JetMedianBackgroundEstimator bge(rho_range, jet_def,
                                 area_def);
```

## grid cells as patches

```
double y_max; // patches acceptance
double cell_size; // typically 0.55
GridMedianBackgroundEstimator bge(y_max, cell_size);
```

## usage

```
double rho = bge.rho(); // bkg density
double sigma = bge.sigma(); // bkg fluctuations
```

# Full specification

---

- For the area computation
  - ghost acceptance (typically the particle  $y_{\max}$ )
  - numer of repetitions (use 1)
  - ghost area (default 0.01, smaller better)
  - type of area (active area with explicit ghosts)
- For the  $\rho$  estimation
  - patch definition ( $k_t$  algorithm,  $R = 0.4$ )
  - patch acceptance (typically  $y_{\max} - R$  or  $y_{\max}$ )



## *Beyond basic usage*

# *Recent developments*

---

## Improvements/extensions of the method

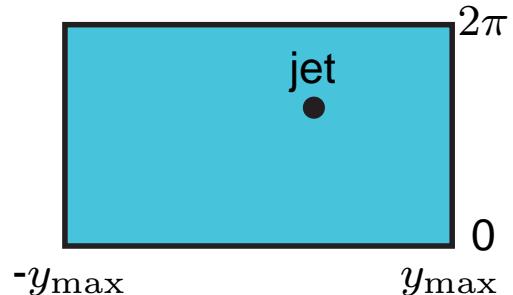
- Methods to handle **positional dependence of  $\rho$**   
Directly relevant for the LHC (e.g. rapidity dependence)  
[M.Cacciari,G.Salam,GS,2010-2011]
- Subtraction for jet mass and jet shapes  
Important for jet tagging (“ $q$  v.  $g$  jet”,  $b$  jet, top jet,  $H \rightarrow b\bar{b}$ )  
[GS,G.Salam,J.Kim,S.Dutta,M.Cacciari,2013]
- Subtraction of **fragmentation function (moments)**  
Useful for quenching in  $PbPb$  collisions  
[M.Cacciari,P.Quiroga,G.Salam,GS,2012]

# *Positional dependence*

 $\rho_{\text{est}}(\text{jet})$

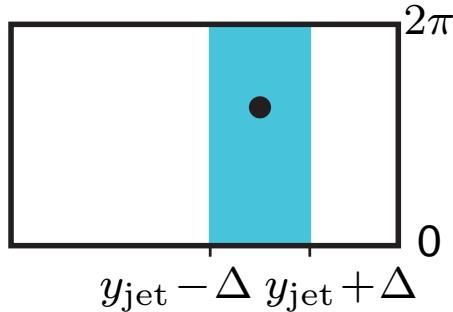
# Rapidity/positional dependence

Global



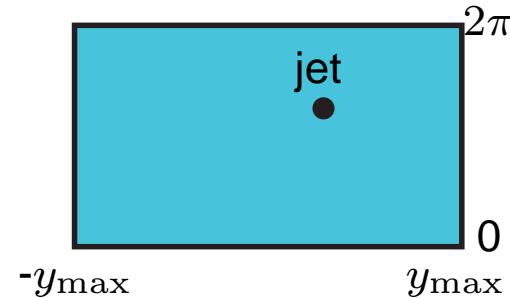
$$\rho_{\text{est}} = \underset{\text{all patches}}{\text{median}} \left\{ \frac{p_{t,j}}{A_j} \right\}$$

Local (strip, doughnut, ...)



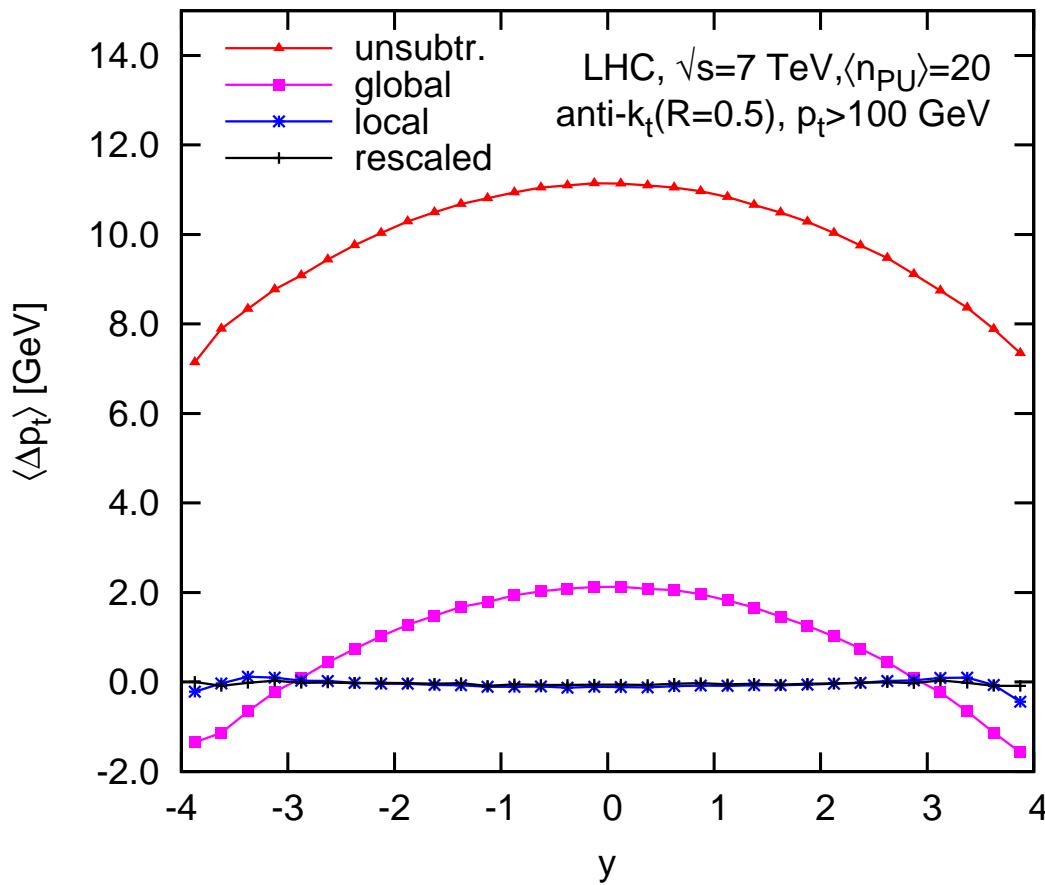
$$\rho_{\text{est}}(y) = \underset{\text{local patches}}{\text{median}} \left\{ \frac{p_{t,j}}{A_j} \right\}$$

Rescaled



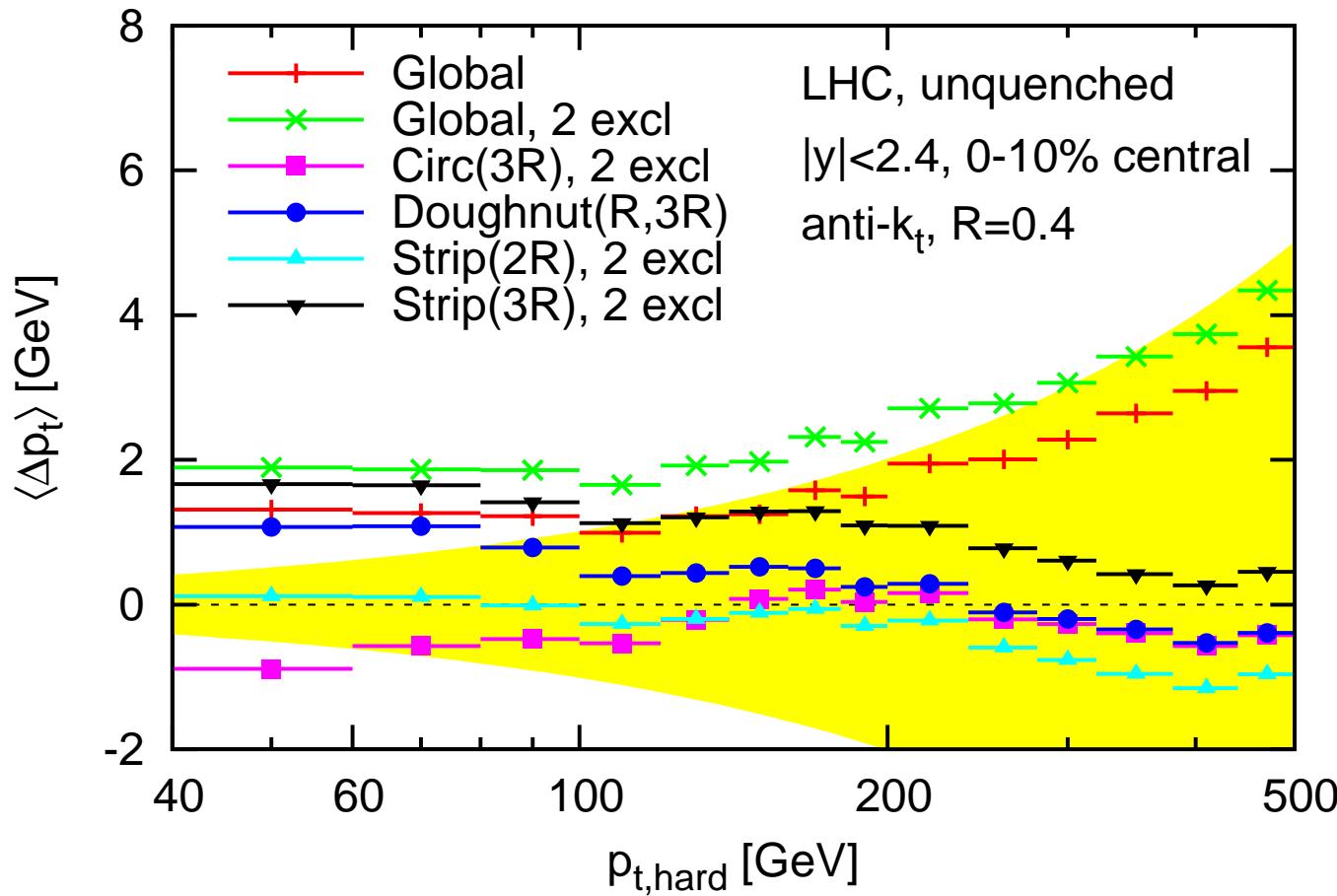
$$\rho_{\text{est}}(y) = f(y) \underset{\text{all patches}}{\text{median}} \left\{ \frac{p_{t,j}}{f(y_j) A_j} \right\}$$

# Rapidity dependence in $pp$



particles and ghosts up to  $|y|=5$ ; jets: anti- $k_t$ ( $R=0.5$ ),  $p_{t,\text{hard}} > 200$  GeV,  $|y|<4$   
 $\rho_{\text{est}}$  from  $k_t(R=0.4)$  jets or cells(0.55) up to  $|y|=4$ , excluding the 2 hardest patches  
'rescaling' uses a 4th order polynomial fitted on minbias events  
'local' is a strip of half-width 1.5  
Events from Pythia8(4C); matching using "common constituents > 50% hard  $p_t$ "

# Rapidity dependence in $PbPb$



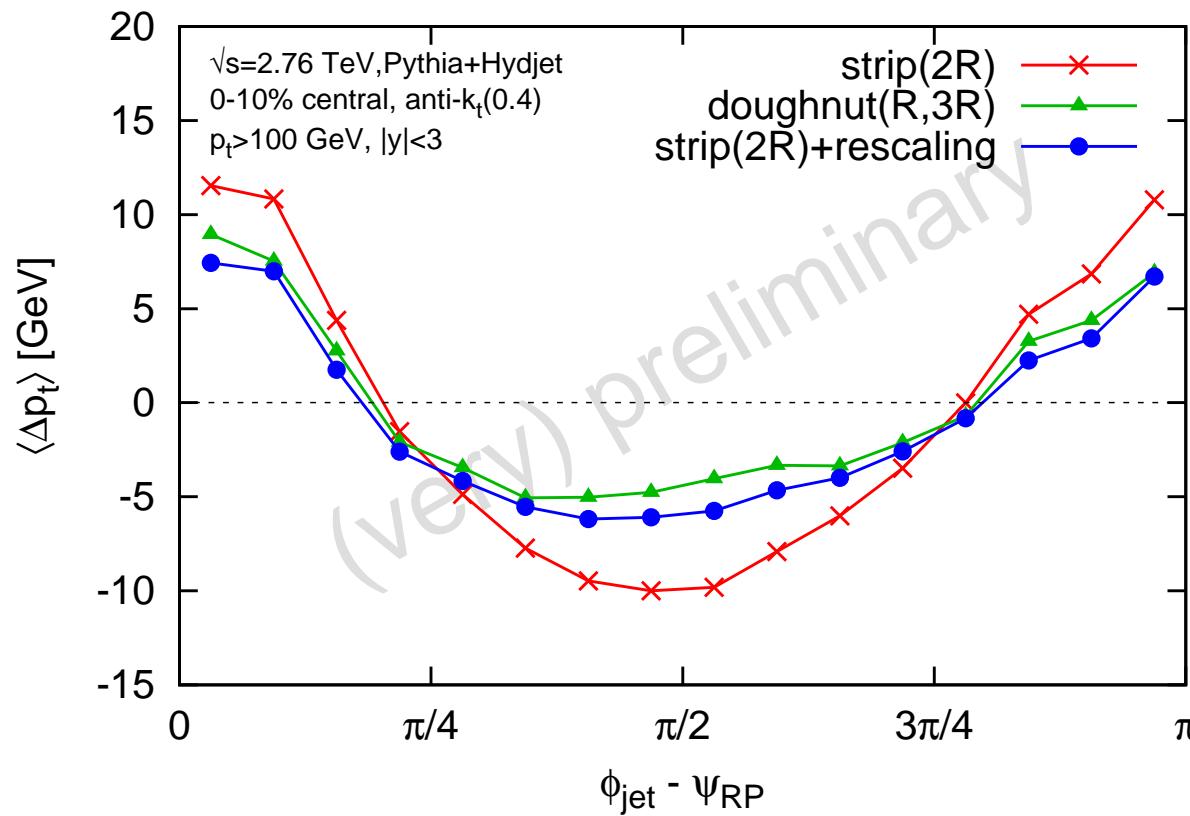
$\neq$  estimates may help assessing the  $\rho_{\text{est}}$  uncertainty

Yellow band is 1% error. Ghosts up to  $|y|=4.2$ ; jets: anti- $k_t$ ( $R=0.4$ ),  $|y|<2.4$

$\rho_{\text{est}}$  from  $k_t(R = 0.5)$  jets  $|y|=4$ , excluding the 2 hardest patches

Events simulated at 5.5 TeV using Pythia 6.4 dijet events embedded in Hydjet 1.6

# Azimuthal dependence in $PbPb$

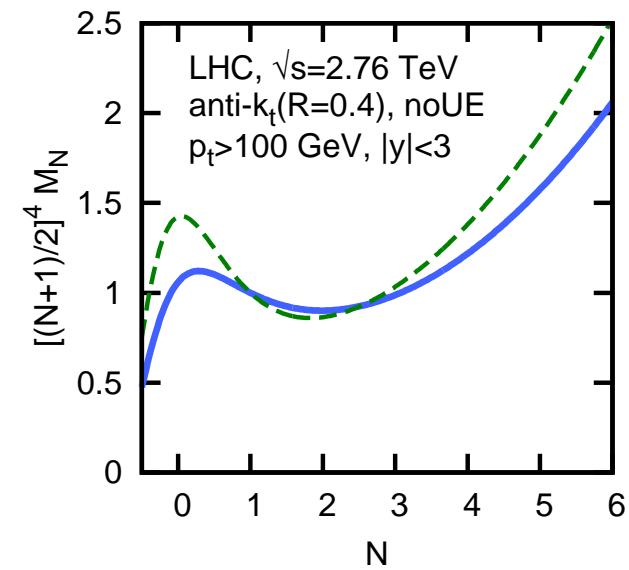
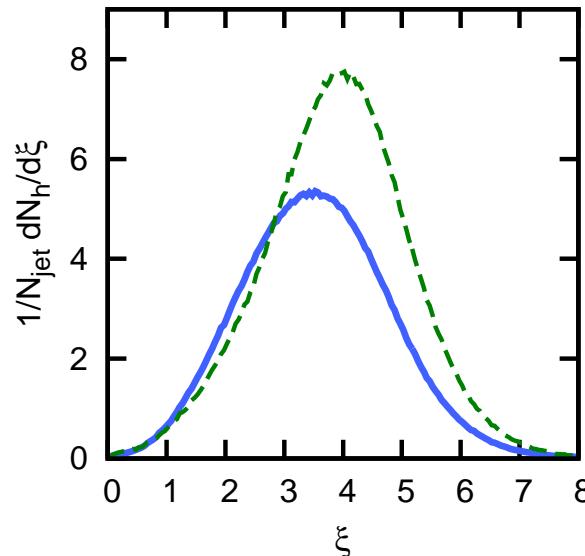
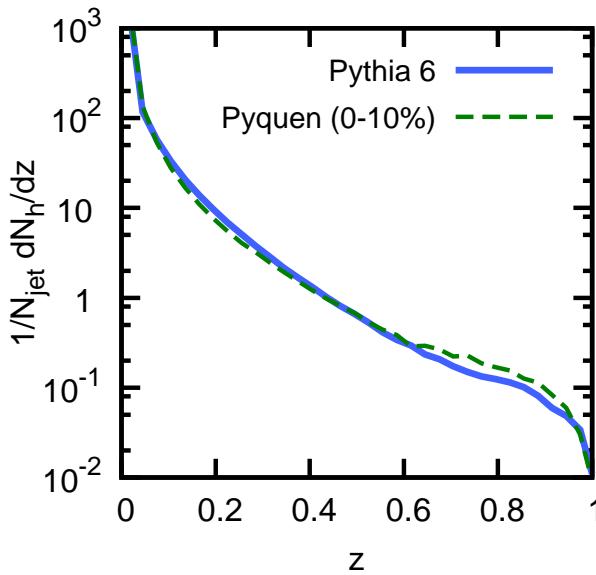


- improvement but obvious room for improvement
- very rough  $\phi$  rescaling...
- Rescaling in FastJet/area-median can be what you decide works best** e.g. Yue-Shi's fit/training/modelling

# *Fragmentation function*

# Fragmentation function in HI

Example ( $z = p_{t,\text{part}}/p_{t,\text{jet}}$ ;  $\xi = \log(1/z)$ ):



- $dN/dz$ ,  $dN/d\xi$  “count” particles in the jet
- Moments (make sense event by event)

$$M_N = \int_0^1 dz z^N \frac{1}{N_{\text{jet}}} \frac{dN}{dz} = \frac{\sum_i p_{t,i}^N}{p_t^N}$$

# *“Standard” background subtraction*

---

Underlying idea:

- measure the medium where it is not affected by the hard jets
- subtracts that from the fragmentation function

Simple test:

region transverse to the dijet event with the same area

# *Subtraction in moment space*

Alternative approach:  
use jet-area-based techniques in moment space

Introduce a new background property  $\rho_N$

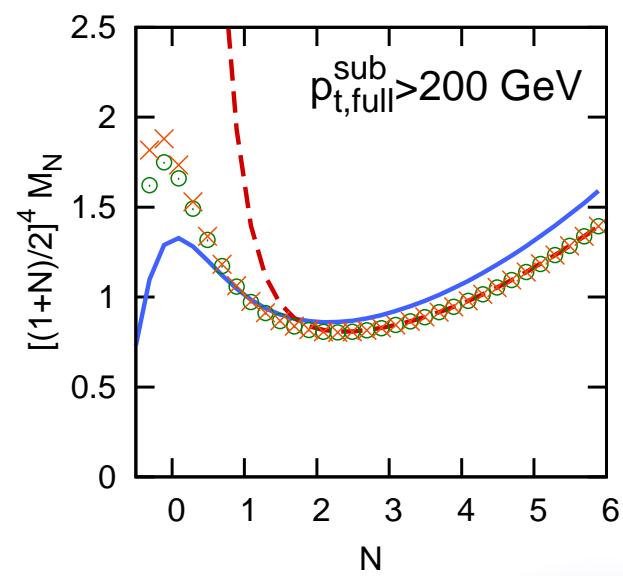
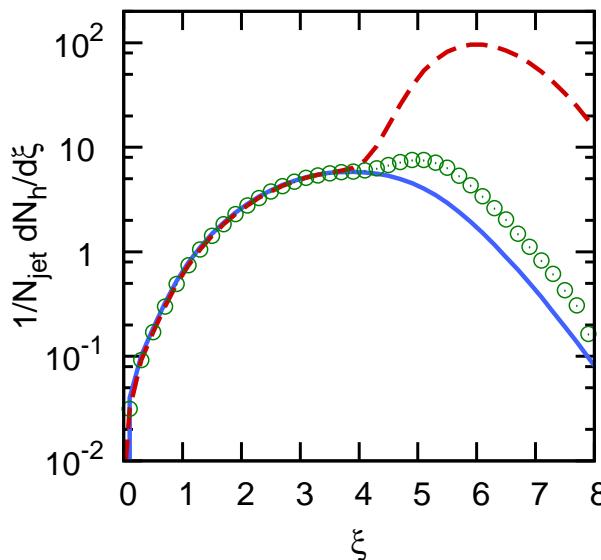
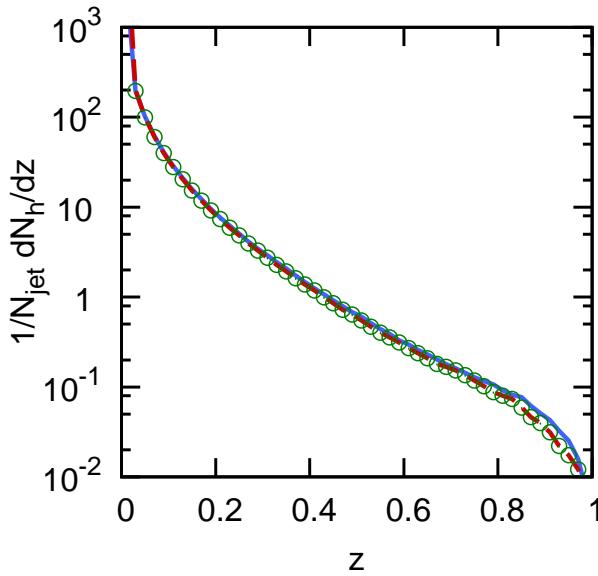
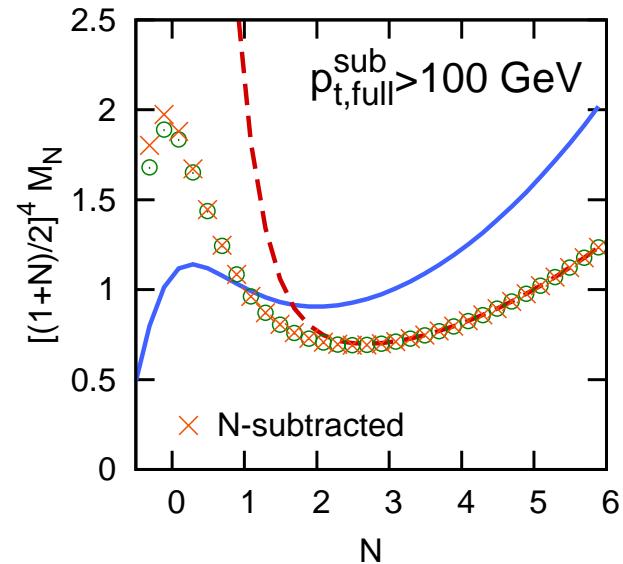
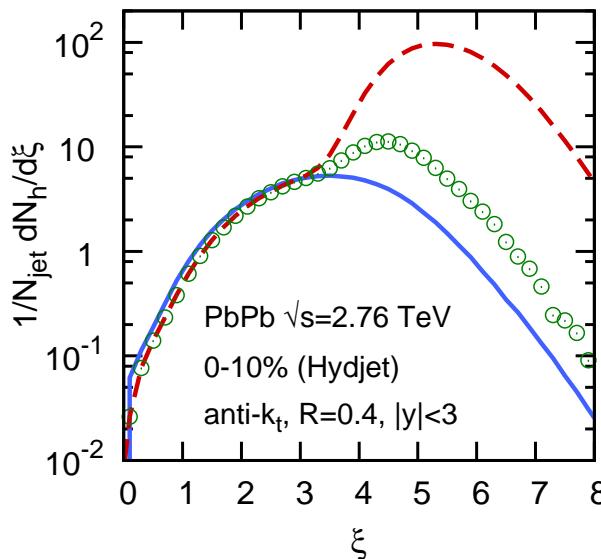
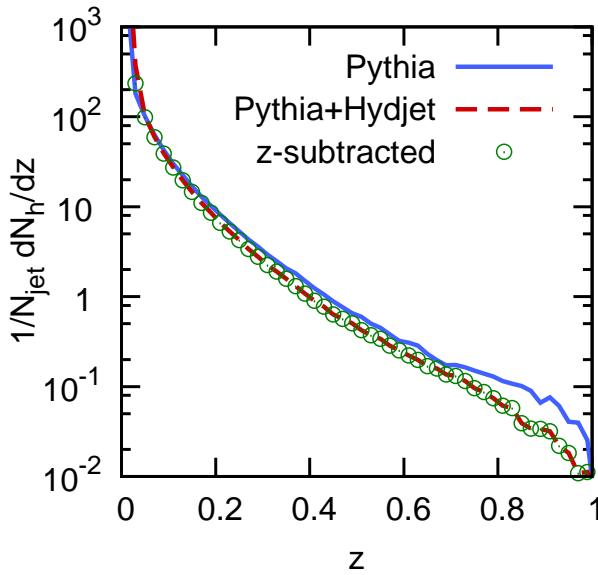
$$\rho = \underset{\text{patches}}{\text{median}} \left\{ \frac{p_{t,\text{patch}}}{A_{\text{patch}}} \right\}$$

$$\rho_N = \underset{\text{patches}}{\text{median}} \left\{ \frac{\sum_{i \in \text{patch}} p_{t,i}^N}{A_{\text{patch}}} \right\}$$

and subtract using

$$M_N^{\text{sub}} = \frac{\sum_{i \in \text{jet}} p_{t,i}^N - \rho_N A}{(p_t - \rho A)^N}$$

# Fragmentation function subtraction



similar improvement for both methods

# Improved subtraction

Problem:

- steeply falling jet spectrum
- cut on  $p_{t,\text{full}}^{\text{sub}}$  tends to pick smaller  $p_{t,\text{hard}}$  with upwards fluctuations

Consequences:

- $p_{t,\text{jet}}$  overestimated i.e.  $z$  underestimated:  
→ underestimation at large  $N$
- extra soft particles in the medium:  
→ overestimation at small  $N$

# Improved subtraction

A correction to these effects can be analytically computed in moment space assuming fluctuations are small ( $\sigma/p_t \ll 1$ )

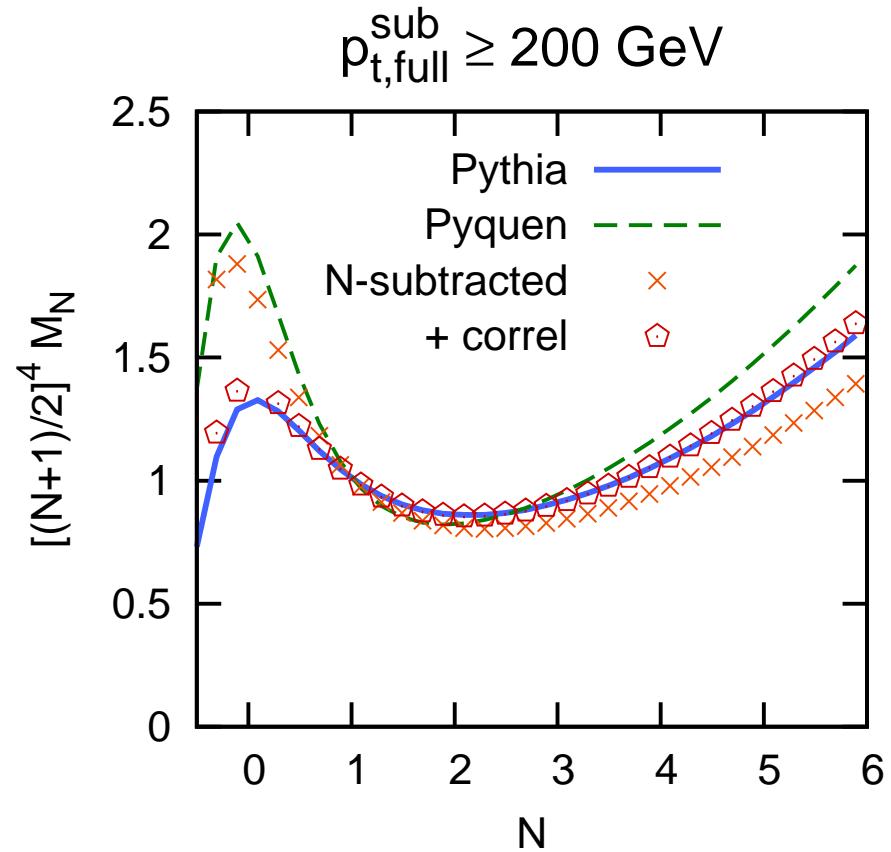
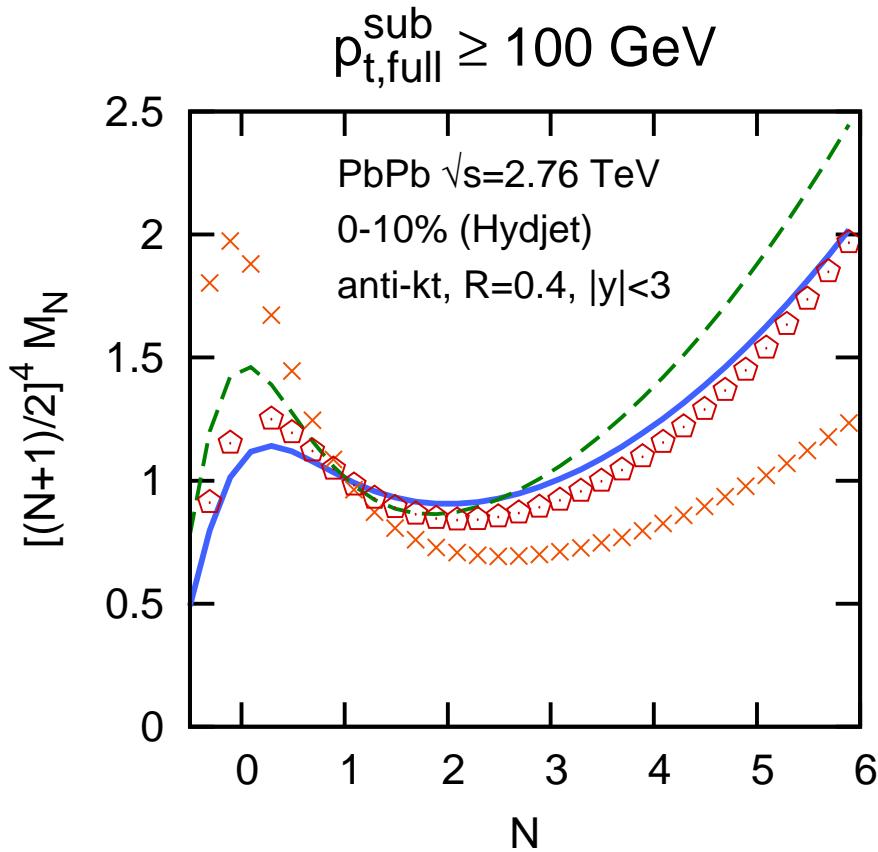
$$M_N^{\text{sub,imp}} = M_N^{\text{sub}} \times \left( 1 + N \frac{\sigma^2 A}{\mu p_{t,\text{jet}}} \right) - r_N \frac{\sigma \sigma_N A}{\mu p_{t,\text{jet}}^N}$$

- $r_N, \sigma_N$  computed from patches in the event  
(uses of the correlations between  $\sum_i p_{t,i}^N$  and  $p_t^N$ )
- $\mu$  is the hardness of the jet spectrum (from  $pp$ )

For  $\sigma/p_t \ll 1$ , OK to approximate the “signal” by an exponential and the background by a Gaussian. Fluctuations of  $\sum_i p_{t,i}^N$  correlated with those of  $p_t^N$

# Improved subtraction

$$M_N^{\text{sub,imp}} = M_N^{\text{sub}} \times \left( 1 + N \frac{\sigma^2 A}{\mu p_{t,\text{jet}}} \right) - r_N \frac{\sigma \sigma_N A}{\mu p_{t,\text{jet}}^N}$$



Nice improvement  
Easily done in moments (local in  $N$  v. matrix-like in  $z$ )!

# Practical implementation

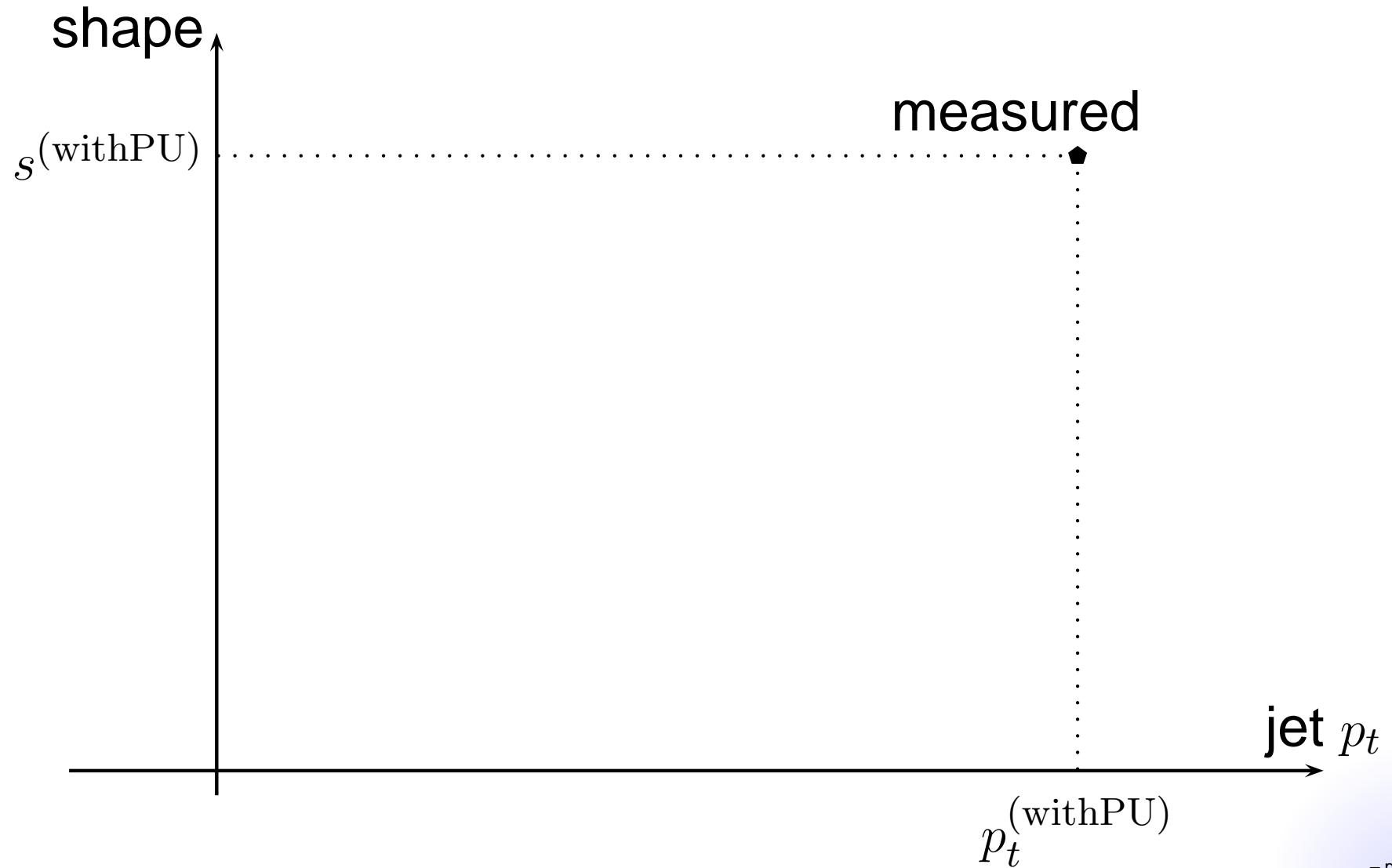
```
JetFFMoments is a FastJet contrib  
#include <fastjet/contrib/JetFFMoments.hh>  
  
// The main object for computing FF moments  
contrib::JetFFMoments ffms(Nmin, NMax, nNs, &bge);  
  
// set the improvement  
double mu = 25.0; // OK in the 150-200 GeV range  
ffms.set_improved_subtraction(mu, rho_range,  
                               event,jet_def,area_def);  
// Note: in FJ-3.1, ffms.set_improved_subtraction(mu)  
  
// Get the moments for a given jet  
vector<double> ffm = ffms(jet);
```

## *jet shapes*

*(function of the jet constituents  
e.g. the jet mass, broadening, ...)*

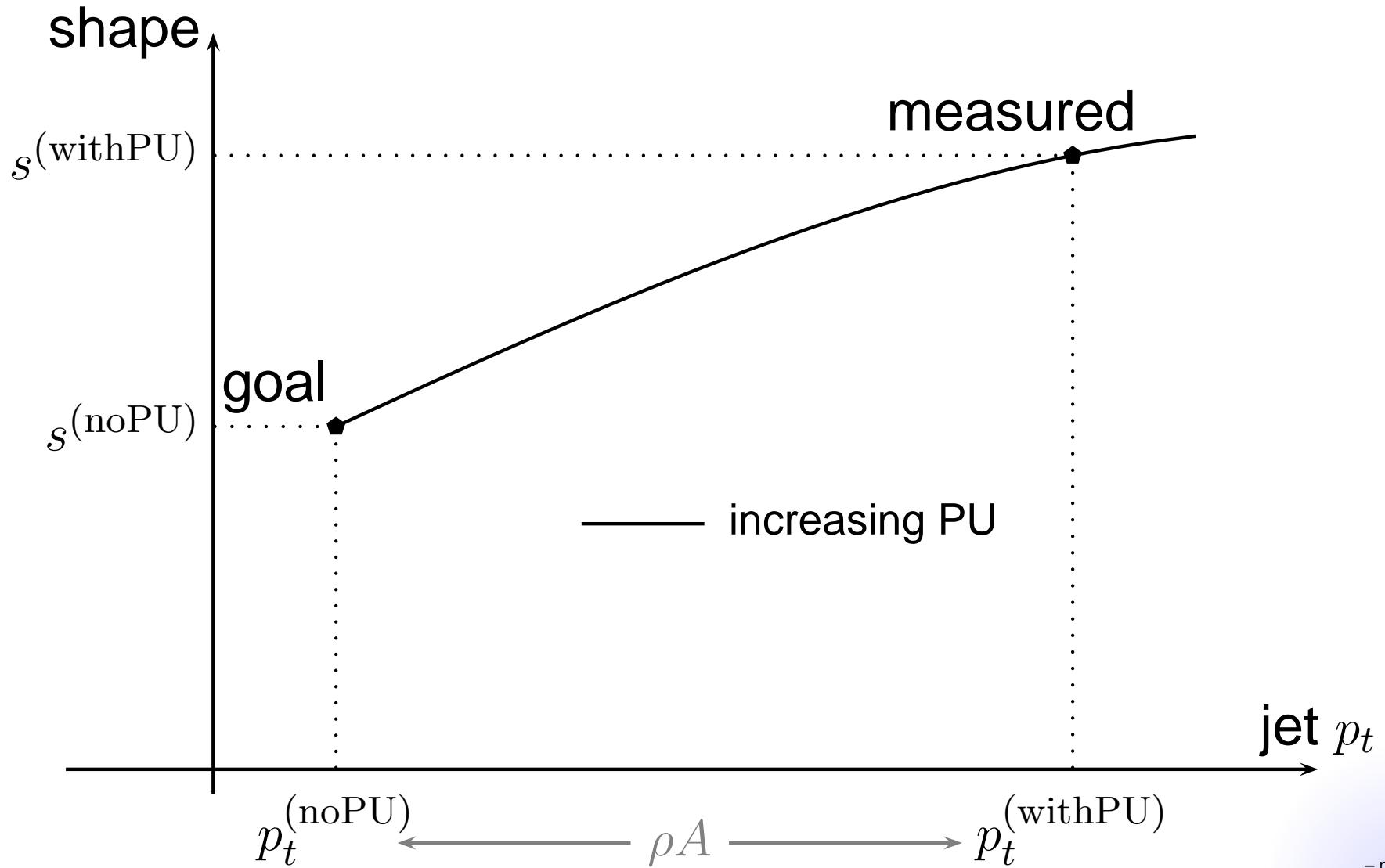
# Idea: area-median + extrapolation to 0 PU

[M.Cacciari, S.Dutta, J.Kim, G.Salam, GS, 2013]



# Idea: area-median + extrapolation to 0 PU

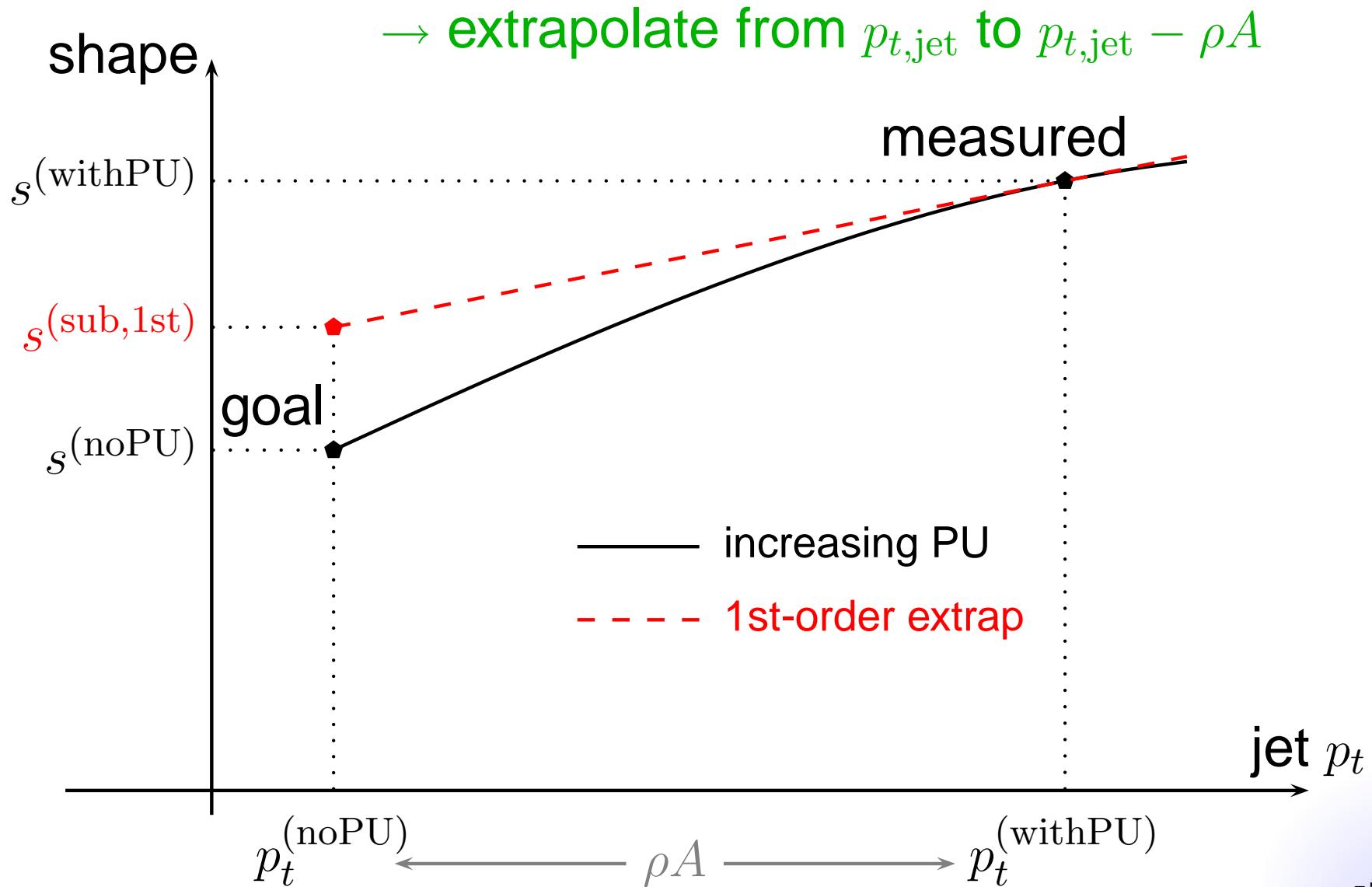
[M.Cacciari, S.Dutta, J.Kim, G.Salam, GS, 2013]



# Idea: area-median + extrapolation to 0 PU

[M.Cacciari, S.Dutta, J.Kim, G.Salam, GS, 2013]

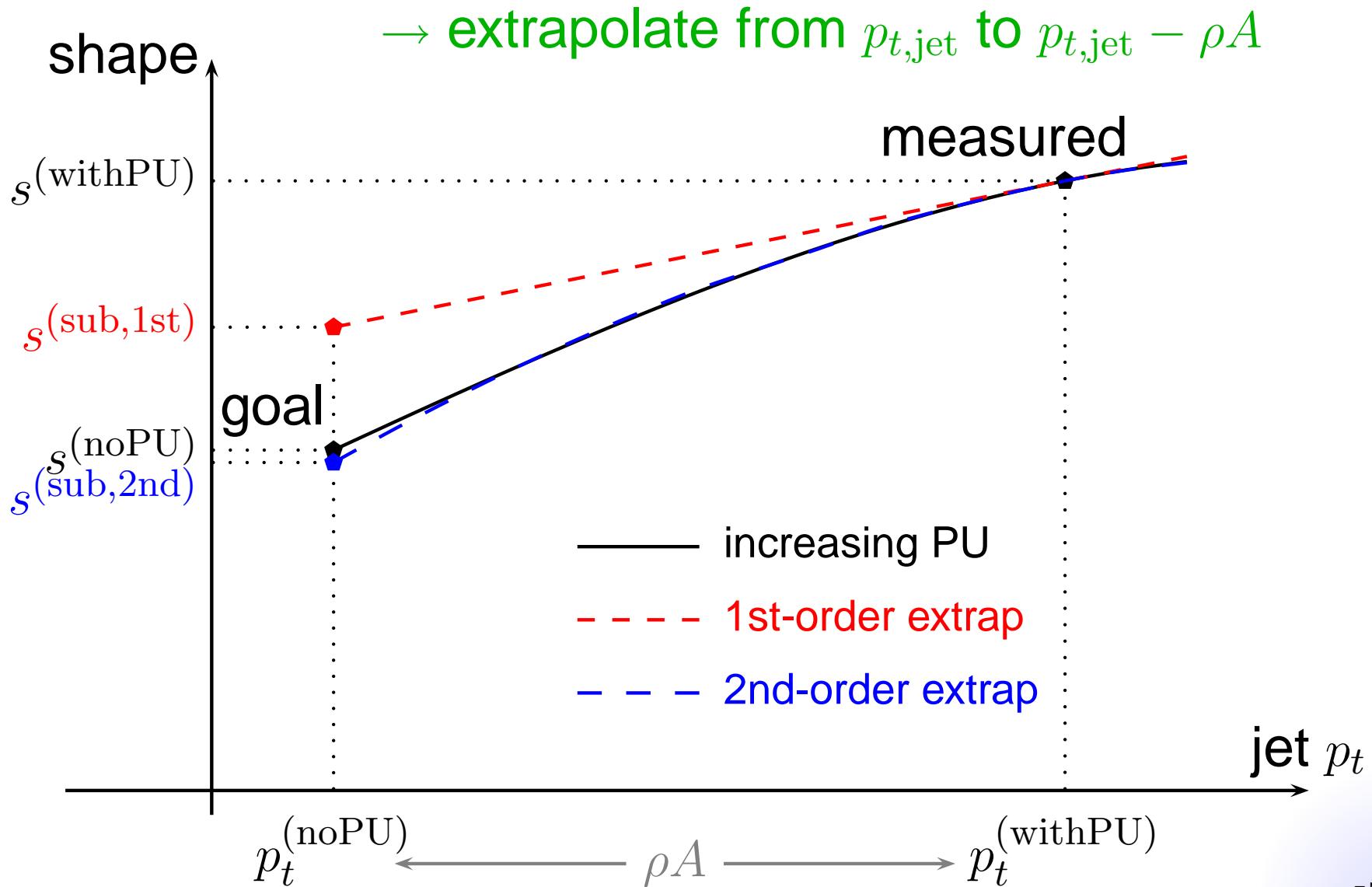
knowledge of the derivatives wrt uniform shift of PU



# Idea: area-median + extrapolation to 0 PU

[M.Cacciari, S.Dutta, J.Kim, G.Salam, GS, 2013]

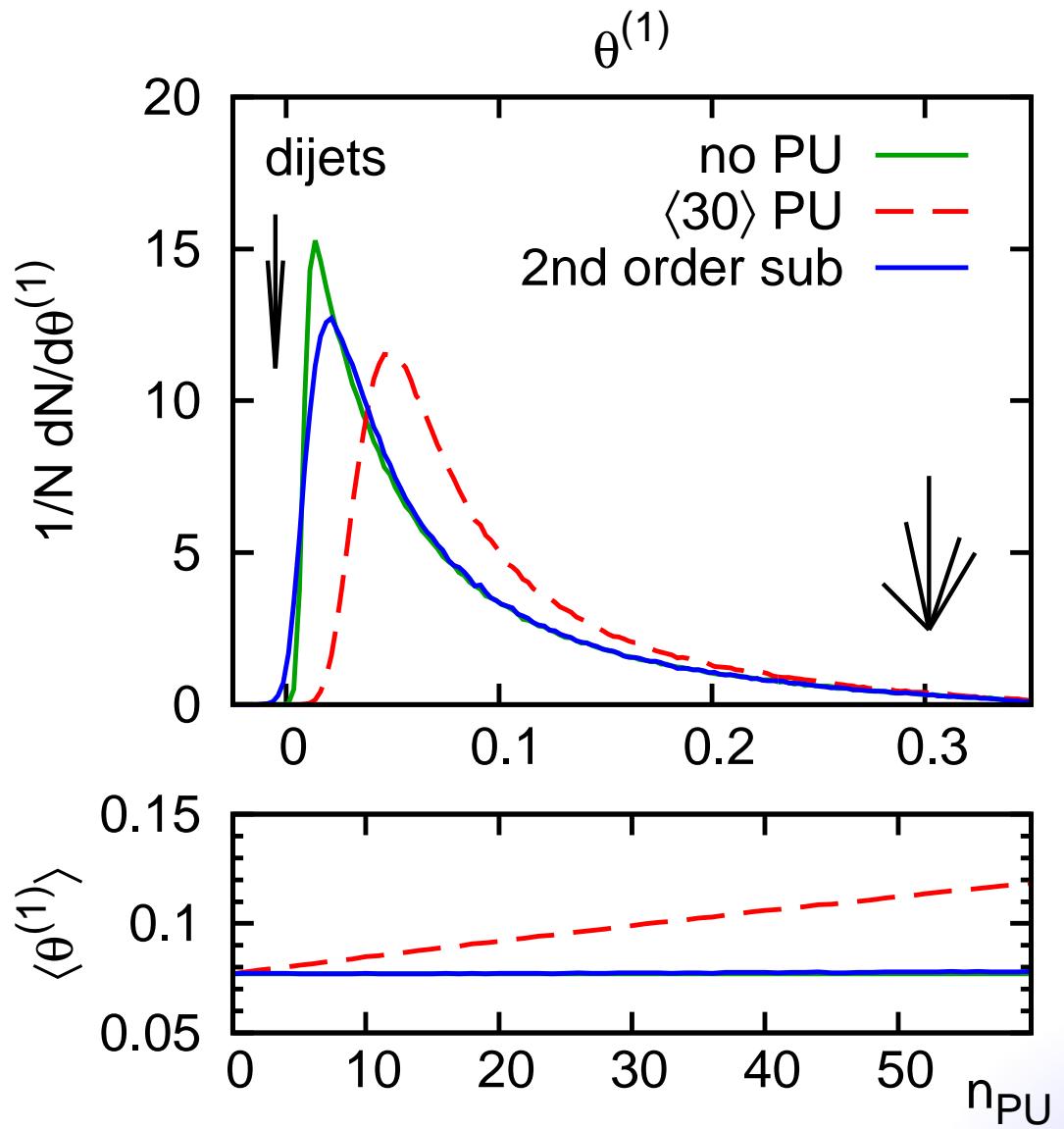
knowledge of the derivatives wrt uniform shift of PU



# Example: quark/gluon discrimination

Use a cut on girth/broadening/width  $\theta^{(1)} < 0.05$

$$\theta^{(1)}(\text{jet}) = \frac{1}{\tilde{p}_t} \sum_{i \in \text{jet}} p_{t,i} \Delta R_{i,\text{jet}}$$

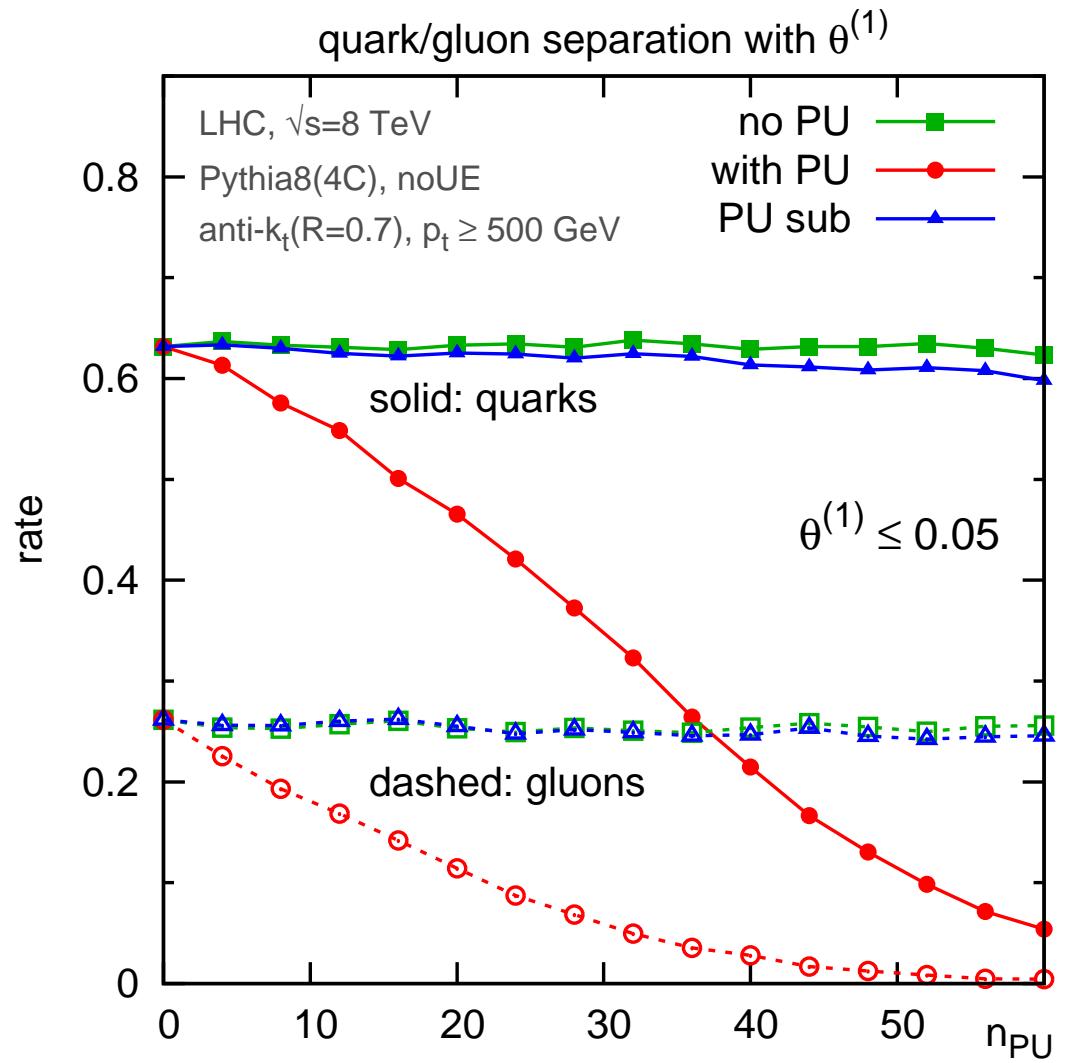


# Example: quark/gluon discrimination

Use a cut on girth/broadening/width  $\theta^{(1)} < 0.05$

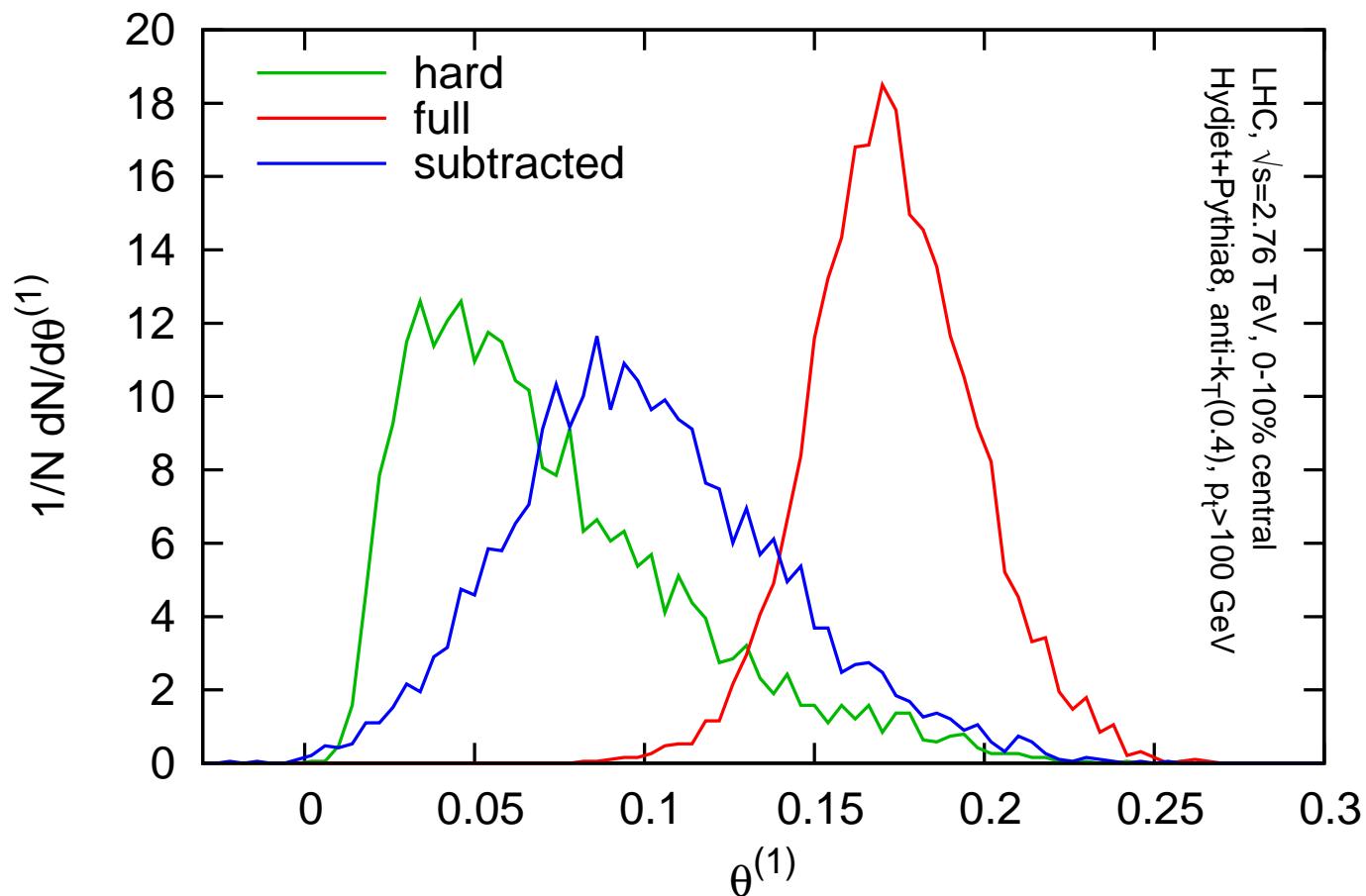
$$\theta^{(1)}(\text{jet}) = \frac{1}{\tilde{p}_t} \sum_{i \in \text{jet}} p_{t,i} \Delta R_{i,\text{jet}}$$

efficiencies very well recovered



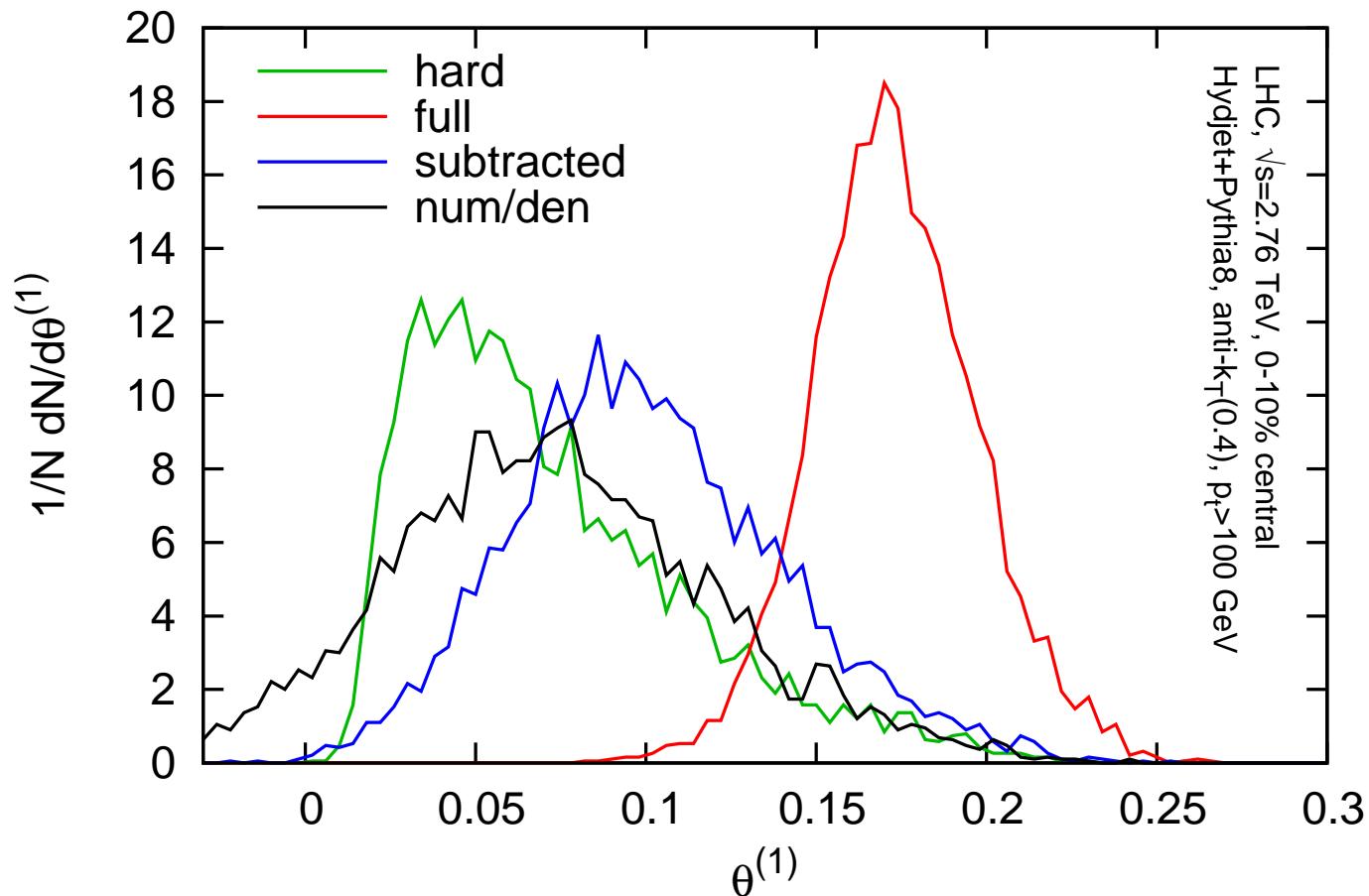
# *Application to heavy-ions (preliminary)*

Does this works for heavy-ions ( $\rho A / p_t \sim 1$ )



# *Application to heavy-ions (preliminary)*

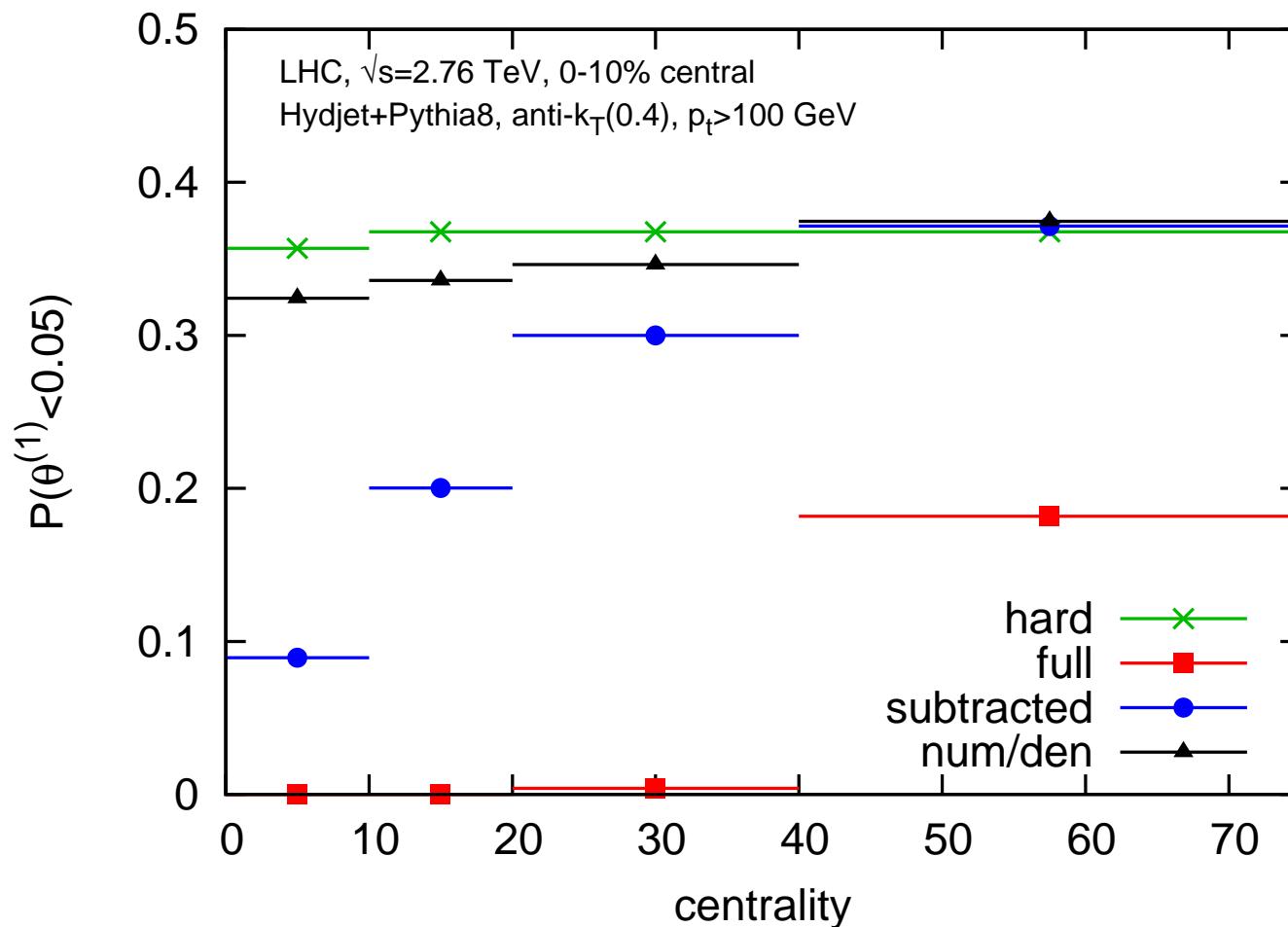
Does this works for heavy-ions ( $\rho A / p_t \sim 1$ )



Subtract numerator and denominator  
independently works decently

# *Application to heavy-ions (preliminary)*

Does this works for heavy-ions ( $\rho A/p_t \sim 1$ )



efficiencies well recovered

# Practical implementation

```
GenericSubtractor is a FastJet contrib  
#include <fastjet/contrib/GenericSubtractor.hh>  
  
// The main object for computing shape subtraction  
contrib::GenericSubtractor gen_sub(&bge);  
  
// Subtract a given shape  
double shape_sub = gen_sub(shape, jet);  
  
// Optional 3rd argument  
GenericSubtractorInfo info;  
double shape_sub = gen_sub(shape, jet, info);  
double shape_unsub = info.unsubtracted();  
double shape_1st = info.first_order_subtracted();
```

## *Extra remarks*

# Filtering

---

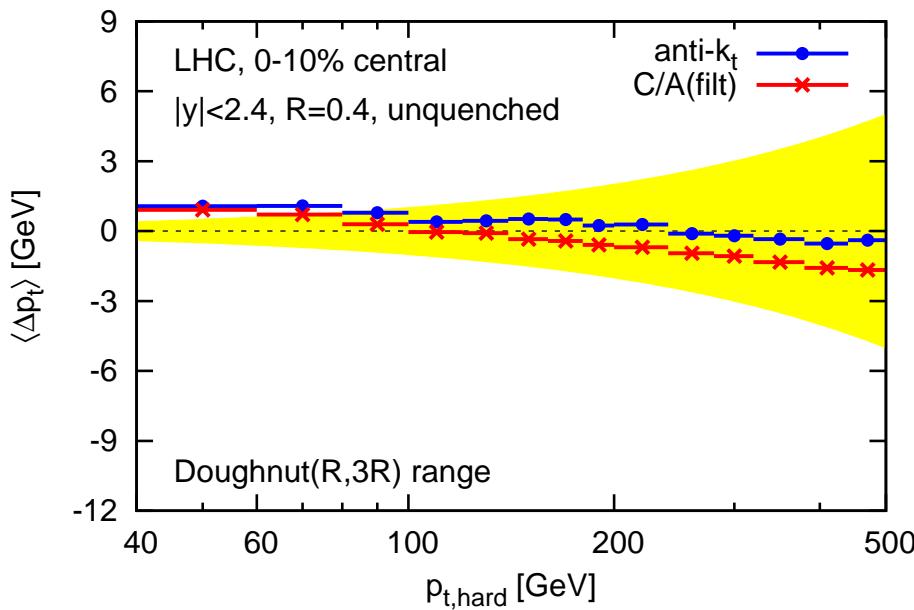
Filtering works as follows:

- Start with a given jet (usually Cambridge/Aachen)
- re-cluster it into smaller subjets (typically C/A( $R/2$ ))
- keep the 2 hardest subjets to form the filtered jet

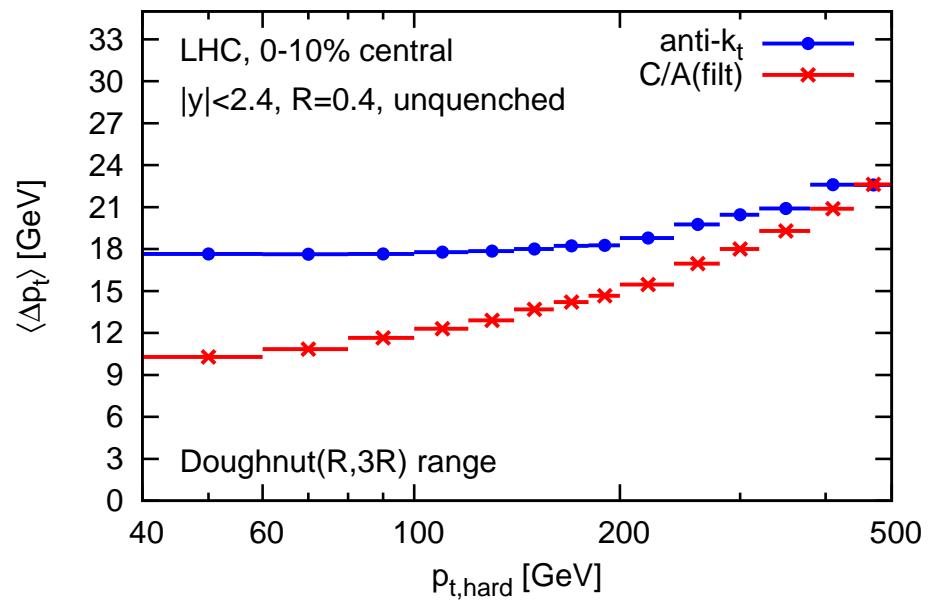
Idea: keep most of the perturbative radiation while reducing background sensitivity (reduced jet area)

# Filtering

average  $p_t$  shift



impact on resolution



corrected for shift

resolution improved

Note: compensation between

- back-reaction
- hardest selection bias

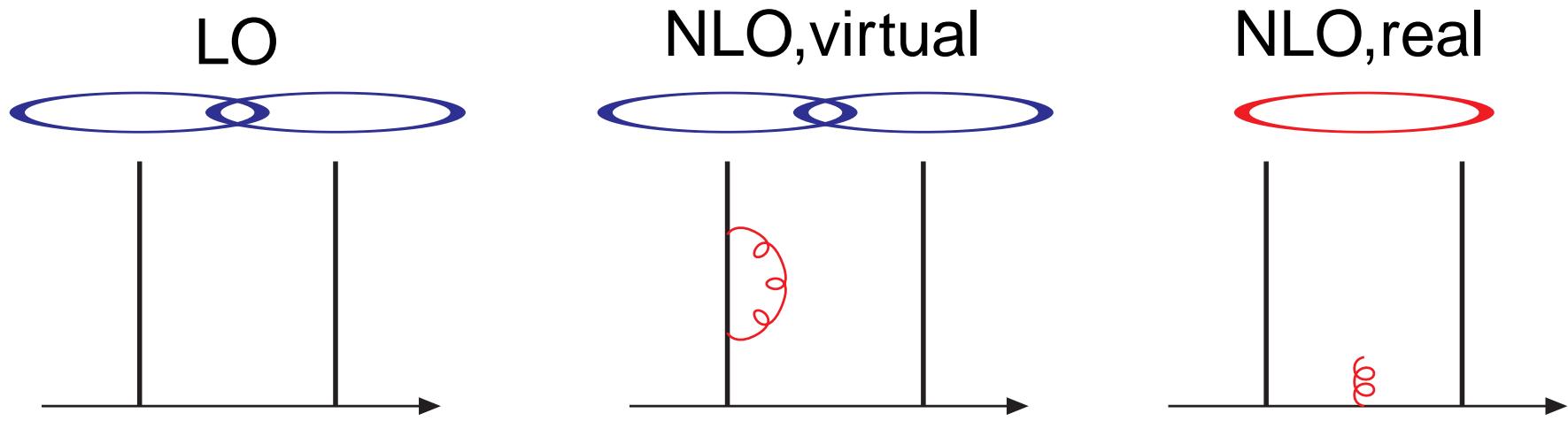
Note: better  $\phi$  response

# *Conclusion and perspectives*

## Area-median background subtraction method

- Simple, public, versatile
- 2 (factorized) ingredients
  - jet area: background contamination  $\propto A$
  - use of the median to estimate  $\rho$
- Key properties
  - corrects for shift (benchmarks OK)
  - reduces resolution smearing
- Many possible extensions/applications
  - positional dependence (local or rescaling)
  - fragmentation function
  - jet shapes

## Example: JetClu



cancellation between real and virtual spoiled

JetClu, ATLASCone:	IR unsafety with 2+1 particles	(NLO for inclusive jet x-sect)
CDF/D0MidPoint:	IR unsafety with 2+1 particles	(NNLO for inclusive jet x-sect)
CMSIterativeCone:	collinear unsafety with 2+1 particles	(NNLO for inclusive jet x-sect)

Origin: incomplete determination of the stable (*i.e.* self-consistent) cones

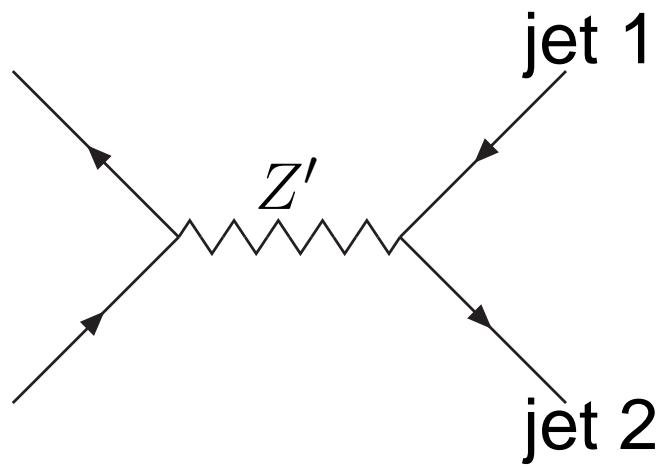
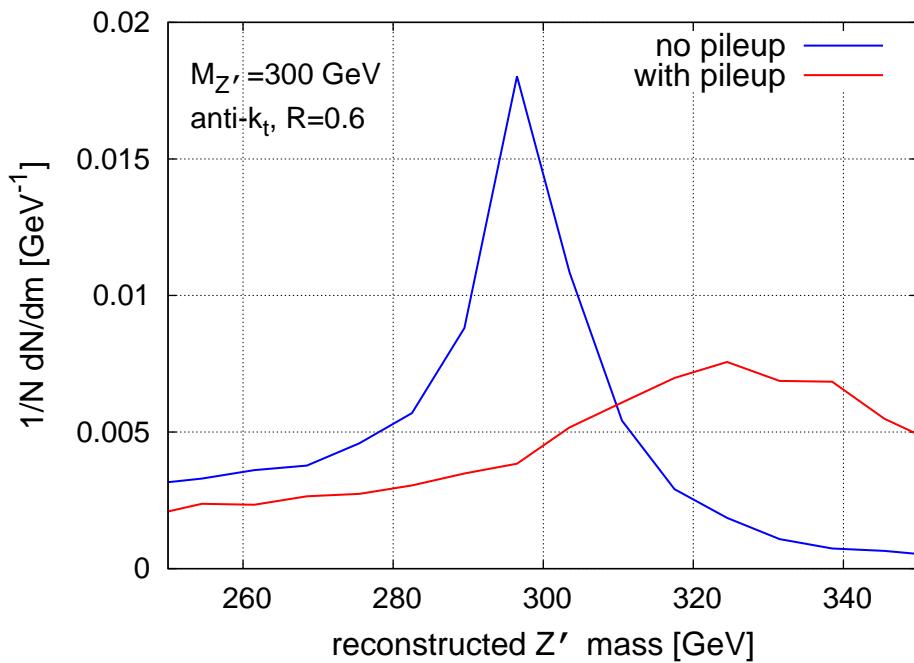
## The FastJet lemma

If  $(i, j)$  is the pair that minimize the  $k_t$  distance and  $k_{t,i} < k_{y,j}$ , then  $j$  is  $i$ 's nearest neighbour

Proof: assume it is not, then  $\exists k$  s.t.  $\Delta R_{ik} < \Delta R_{ij}$  and

$$\begin{aligned}\min(k_{t,i}^2, k_{t,l}^2) \Delta R_{il}^2 &\leq k_{t,i}^2 \Delta R_{il}^2 \\ &\leq \min(k_{t,i}^2, k_{t,j}^2) \Delta R_{il}^2 \\ &< \min(k_{t,i}^2, k_{t,j}^2) \Delta R_{ij}^2\end{aligned}$$

# Illustration of PU effects



- Shift due to the “ $\rho A$ ” term
- Smearing due to the “ $\sigma_\rho A$ ” and “ $\sigma \sqrt{A}$ ” terms

# Heavy ions

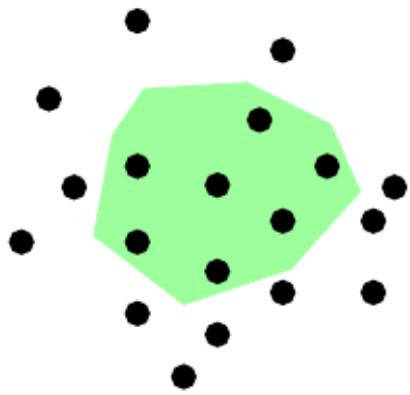
Note: same considerations for “spectator  $p$  and  $n$ ”  
in heavy ion collisions

Typical case: anti- $k_t$   $R = 0.4$ , 20 PU or 0–10% centrality

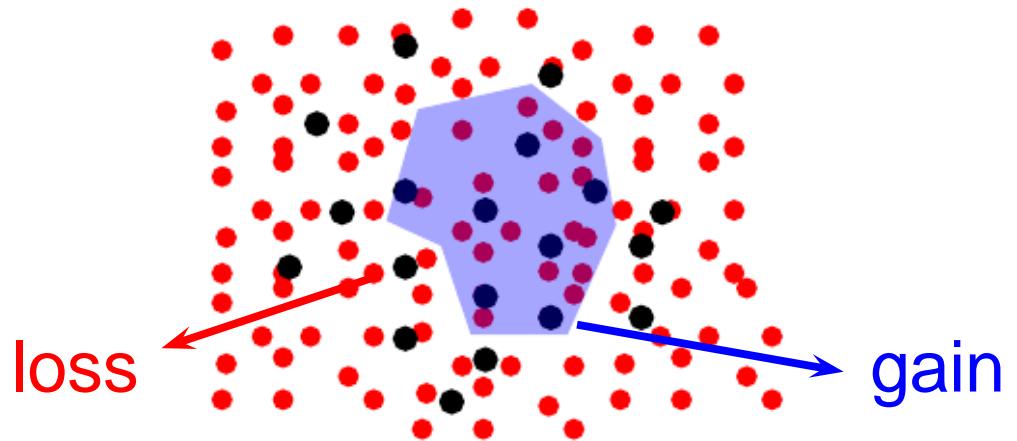
<i>Estimates</i>	LHC, $pp$	LHC, $PbPb$
$\rho$	15 GeV	200 GeV
$\sigma_\rho$	4 GeV	40 GeV
$\sigma$	5 GeV	20 GeV
$A_{\text{jet}}$	0.5	0.5
$\delta p_{t,\text{jet}}$	7.5 GeV	100 GeV
$\sigma_{\text{jet}}$	3.5 GeV	16 GeV

# *Back reaction*

No background



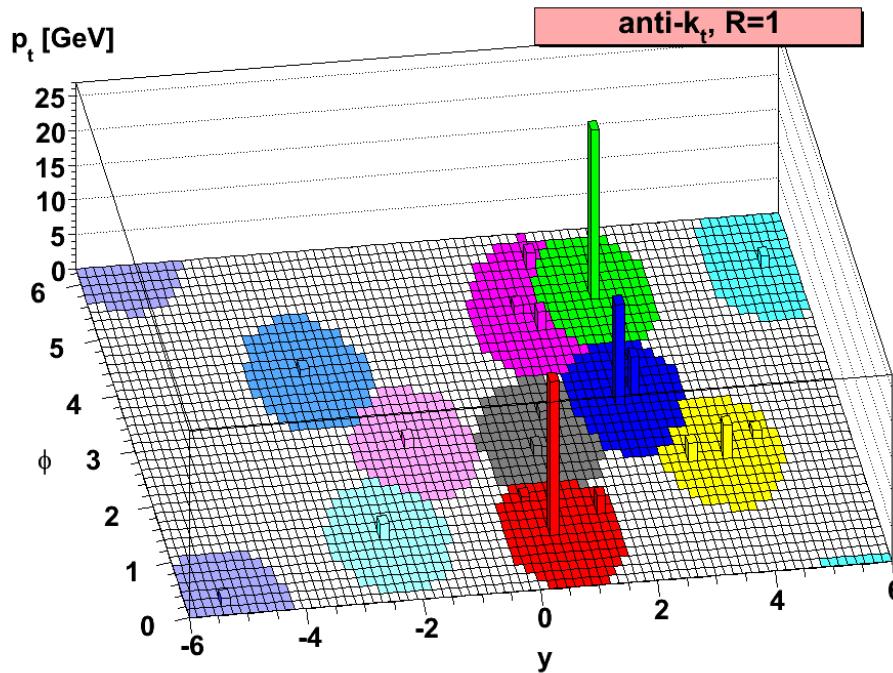
With background



Negligible for anti- $k_t$   
(a nice consequence of its soft resilience)

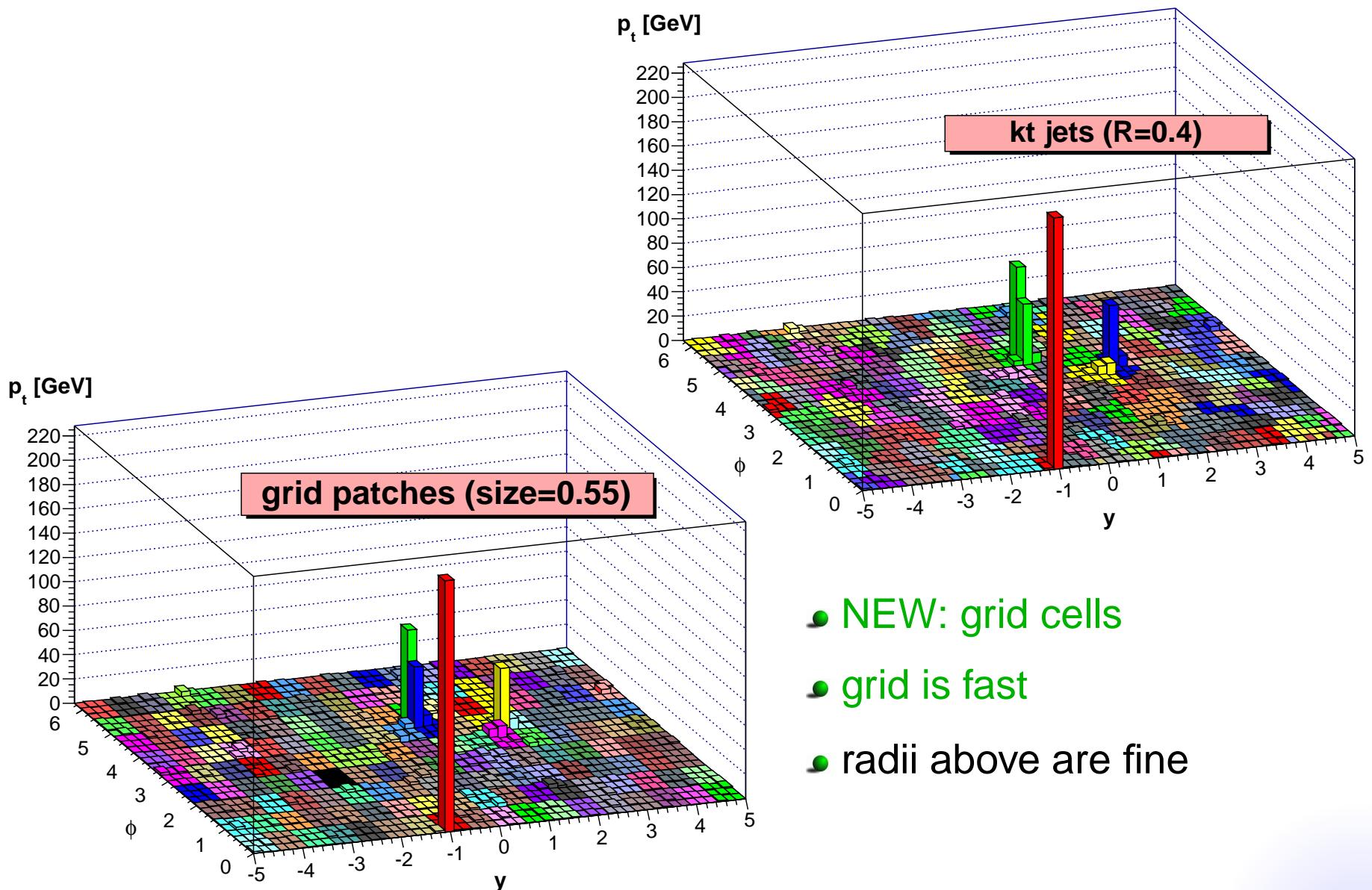
# Determining jet area

- jet area: available with jet clustering
  - add a dense coverage of particles with tiny  $p_t$  ( $\equiv$  area quanta)
  - jet area  $\propto$  number of these “ghosts” in the jet



# Patches of similar size

Patched are typically one of the 2 following types



# *Jet shape subtraction*

No pileup:  $s_{\text{noPU}}(\text{jet}) = s(\{p_{t,i}\}_{\text{hard}})$

With pileup:  $s_{\text{wPU}}(\text{jet}) = s(\{p_{t,i}\}_{\text{hard}}, \{p_{t,i}\}_{\text{PU}})$

- Assume  $\rho \ll p_t$  and expand in series of  $\rho$
- PU  $\propto$  ghosts  $\Rightarrow \partial_\rho \propto \partial_{\text{ghostscale}}$
- numerically: vary the ghost scale to compute the derivatives

# Jet shape subtraction

No pileup:  $s_{\text{noPU}}(\text{jet}) = s(\{p_{t,i}\}_{\text{hard}})$

With pileup:  $s_{\text{wPU}}(\text{jet}) = s(\{p_{t,i}\}_{\text{hard}}, \{p_{t,i}\}_{\text{PU}})$

- Assume  $\rho \ll p_t$  and expand in series of  $\rho$
- PU  $\propto$  ghosts  $\Rightarrow \partial_\rho \propto \partial_{\text{ghostscale}}$
- numerically: vary the ghost scale to compute the derivatives

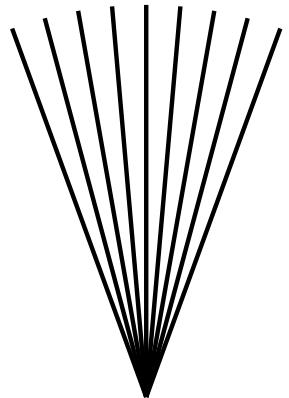
Subtraction:

$$\begin{aligned} s_{\text{sub}}(\text{jet}) &= s_{\text{wPU}}(\text{jet}) - \rho a_{\text{ghost}} \partial_{\text{ghostscale}} s_{\text{wPU}}(\text{jet}) \\ &\quad + \frac{1}{2} (\rho a_{\text{ghost}})^2 \partial_{\text{ghostscale}}^2 s_{\text{wPU}}(\text{jet}) + \dots \end{aligned}$$

# *Computing derivatives: back to the concepts*

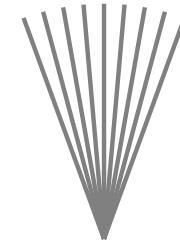
Idea of area-median subtraction:

pile-up



ghosts

$$\propto \rho \times$$



# *Computing derivatives: back to the concepts*

Idea of area-median subtraction:

pile-up

ghosts

$$\{p_i^\mu\} \propto \rho \times \{g_i^\mu\}$$

# Computing derivatives: back to the concepts

Idea of area-median subtraction:

pile-up

ghosts

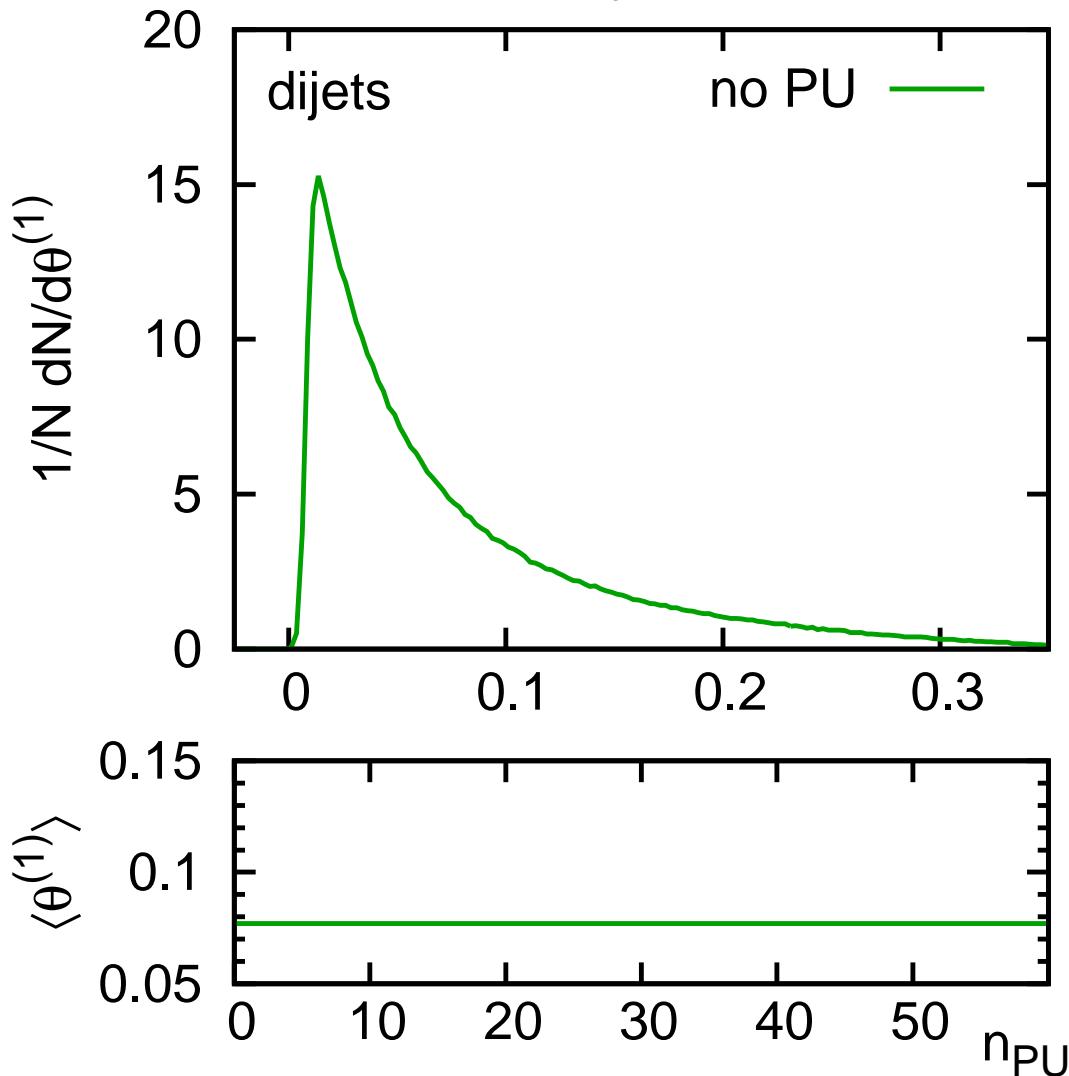
$$\{p_i^\mu\} \sim \rho \times \{g_i^\mu\} \frac{a_{\text{ghost}}}{\bar{g}_t}$$

The set of ghosts mimics the uniform background  
→ use that to compute the derivatives numerically

$\bar{g}_t$  = ghost  $p_t$  = ghostscale

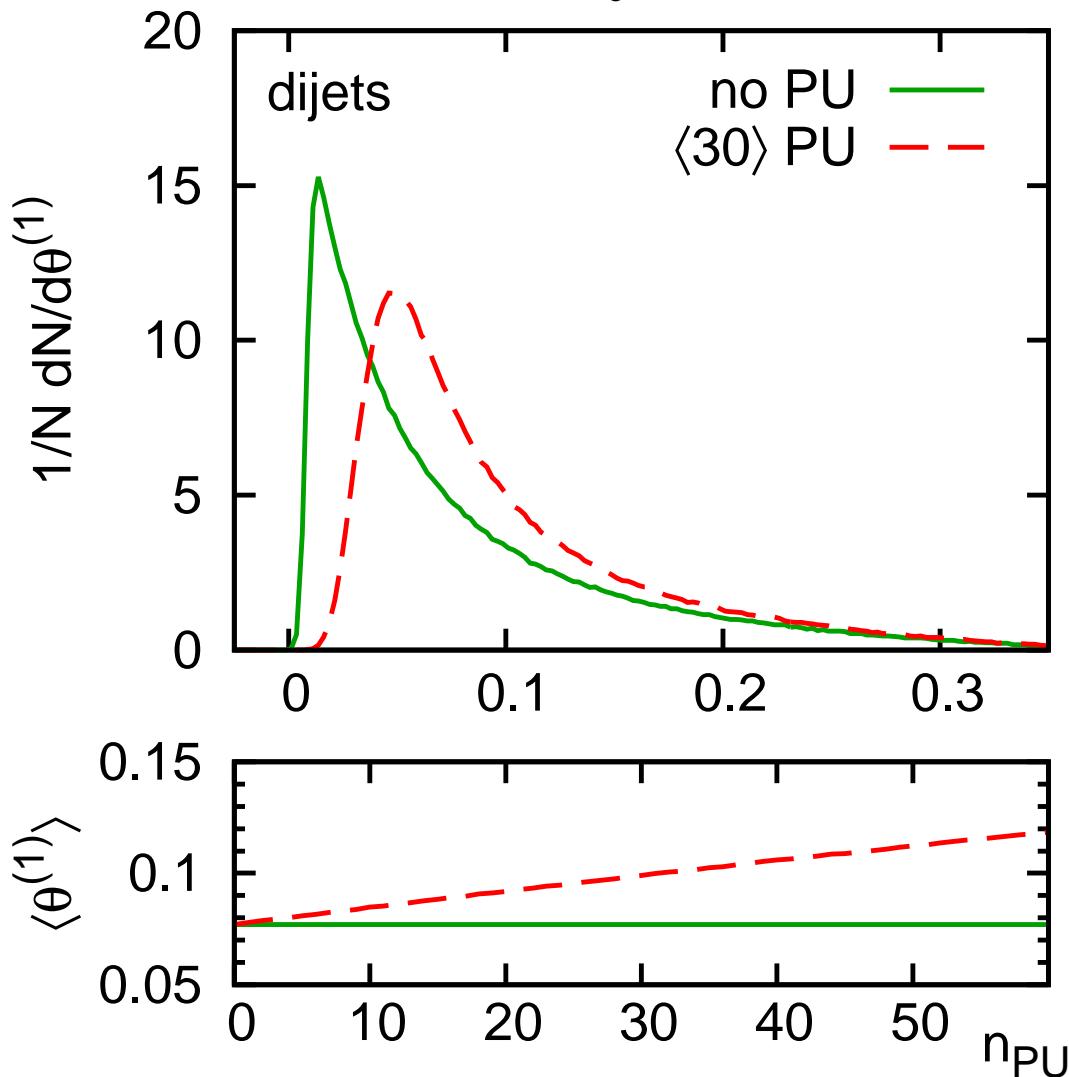
# detailed example: angularities

$$\theta^{(\alpha)} = \frac{\sum_{i \in \text{constits}} p_{t,i} \Delta R_{i,\text{jet}}^{\alpha}}{\sum_{i \in \text{constits}} p_{t,i}}$$



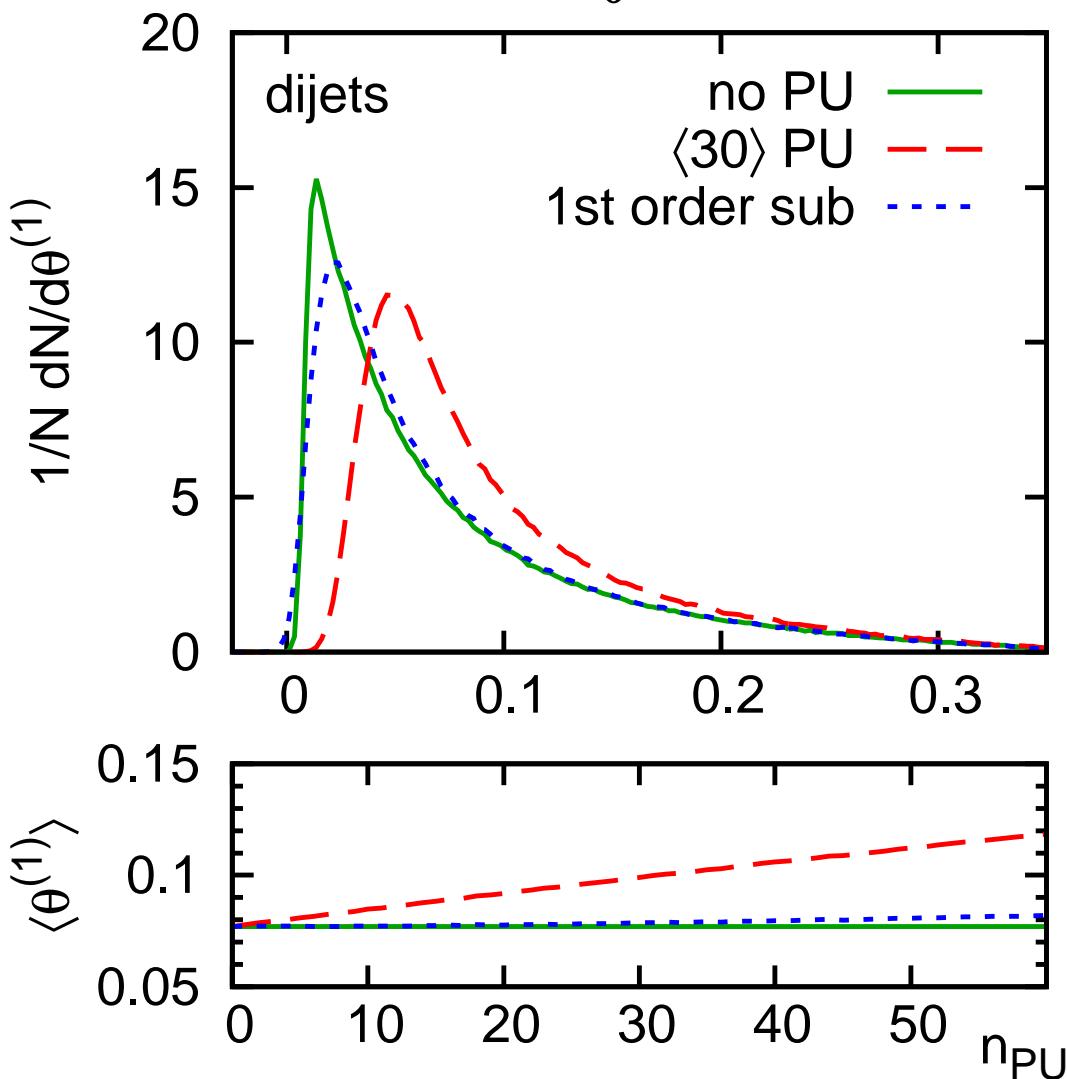
# detailed example: angularities

$$\theta^{(\alpha)} = \frac{\sum_{i \in \text{constituents}} p_{t,i} \Delta R_{i,\text{jet}}^{\alpha}}{\sum_{i \in \text{constituents}} p_{t,i}}$$



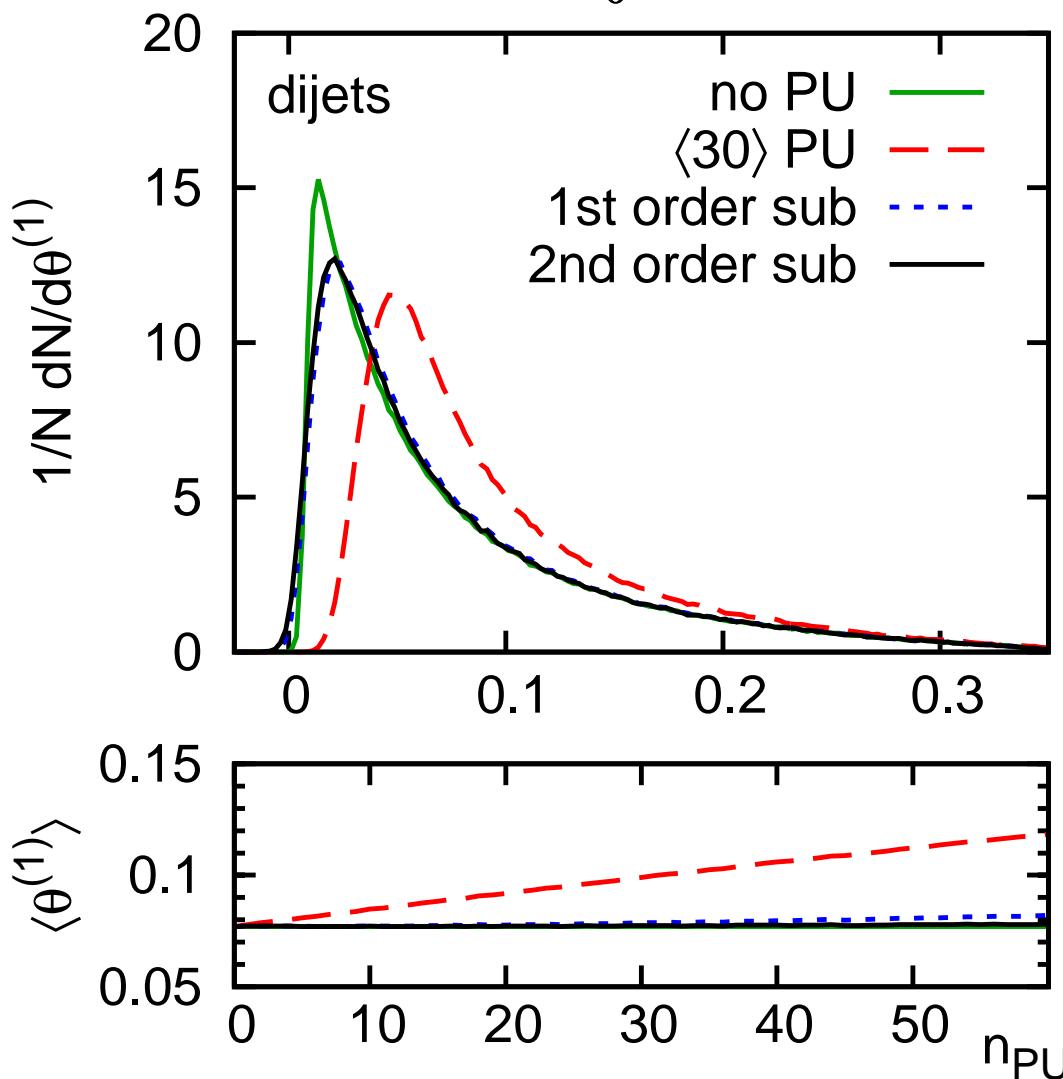
# detailed example: angularities

$$\theta^{(\alpha)} = \frac{\sum_{i \in \text{constituents}} p_{t,i} \Delta R_{i,\text{jet}}^{\alpha}}{\sum_{i \in \text{constituents}} p_{t,i}}$$

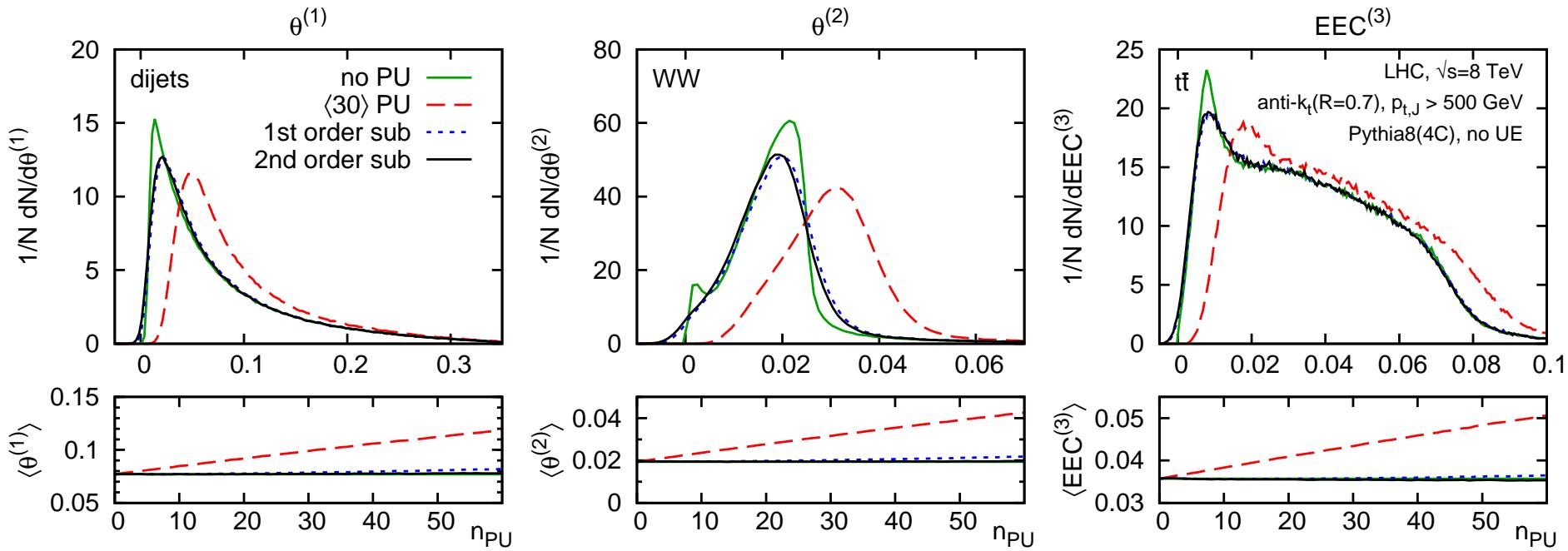


# detailed example: angularities

$$\theta^{(\alpha)} = \frac{\sum_{i \in \text{constituents}} p_{t,i} \Delta R_{i,\text{jet}}^{\alpha}}{\sum_{i \in \text{constituents}} p_{t,i}}$$

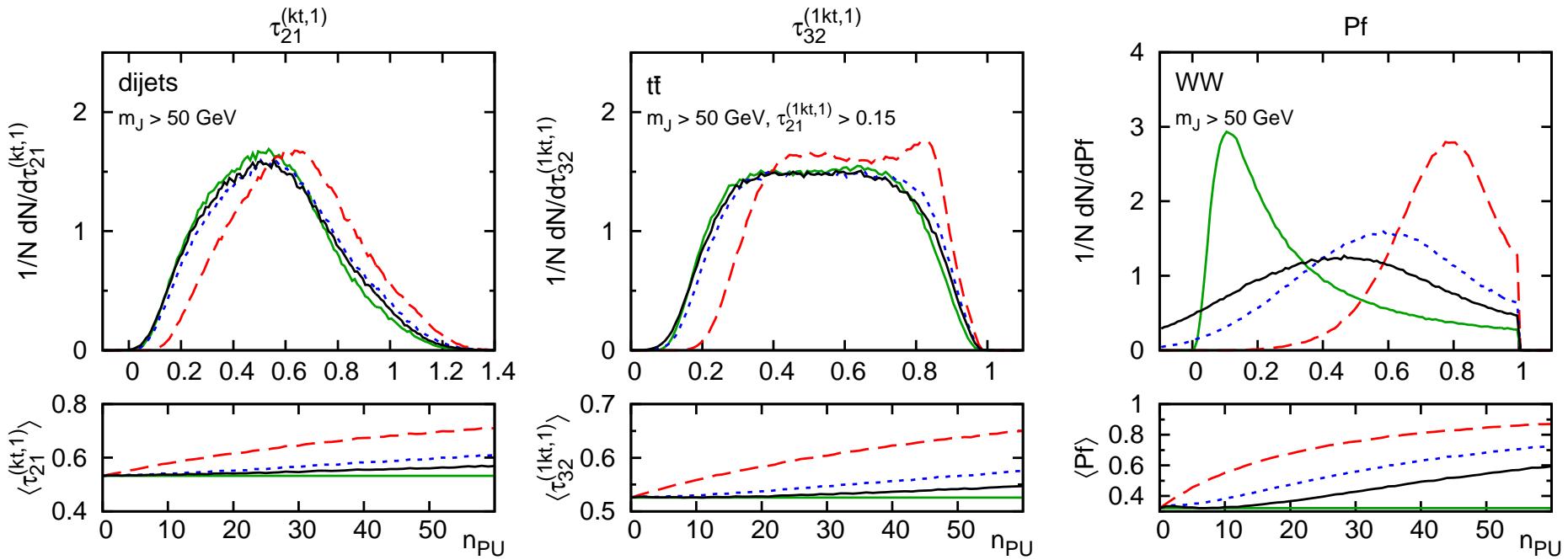


# Shape subtraction examples (1/2)



- PU has a 50-100% effect (5-10% for  $p_t$ )
- Subtraction works (very) well generically
- Broadening of sharp peaks (PU fluctuations!)

# Shape subtraction examples (2/2)



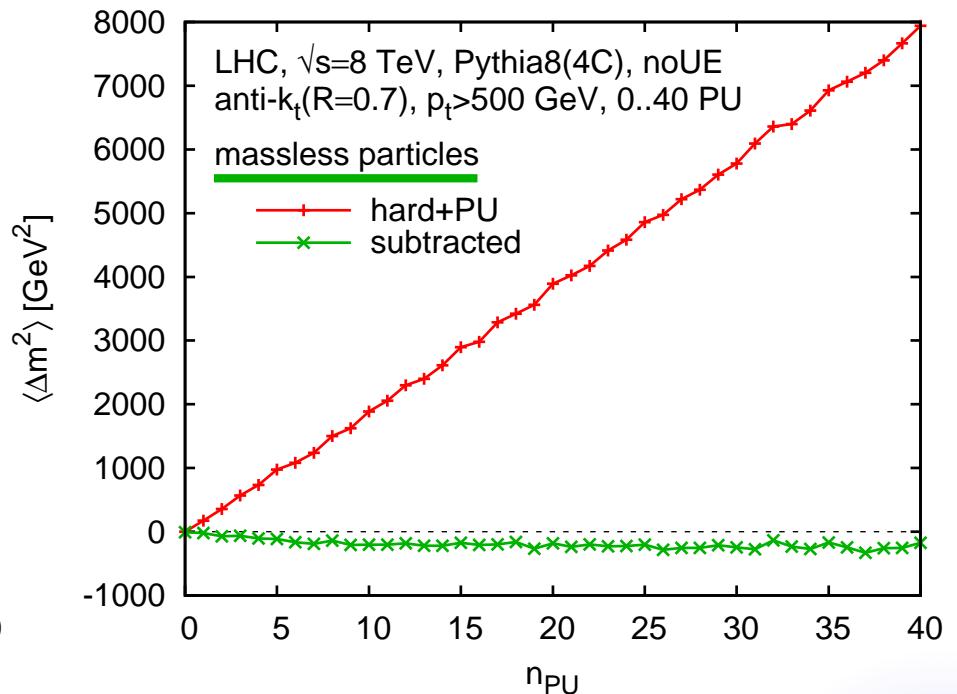
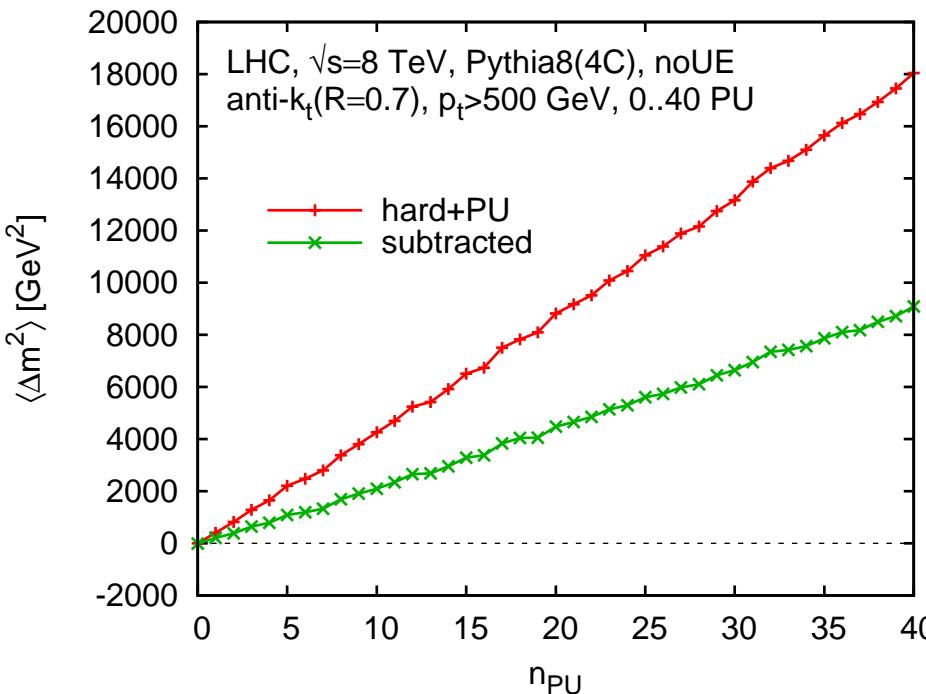
- 2nd derivative sometimes helpful
- Planar flow very poor
  - very PU-sensitive! (from  $Pf = 0$  to  $Pf = 1$ )
  - series expansion breaks for  $n_{PU} \gtrsim 15$

# *Jet mass: the problem*

$$p_{\text{jet}}^{\mu,(\text{sub})} = p_{\text{jet}}^\mu - \rho_{\text{est}} A_{\text{jet}}^\mu$$

applies to the jet 4-momentum

How do we do for the jet mass?



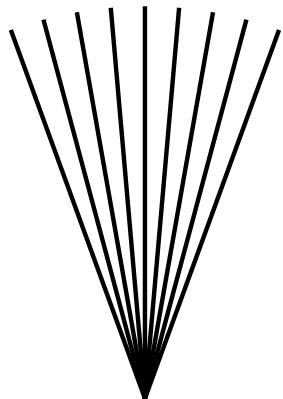
OK for massless particles... but if one want massive ones...

# Back to the concepts in pictures

area-median subtraction:

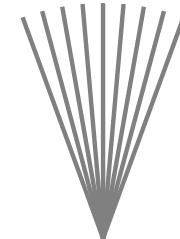
$$m_t = \sqrt{p_t^2 + m^2}$$

pile-up



ghosts

$\propto \rho \times$



$$\sum_i p_i^\mu$$

$(\textcolor{green}{p}_t \cos(\phi_i), \textcolor{green}{p}_t \sin(\phi_i),$   
 $\textcolor{red}{m}_t \sinh(y_i), \textcolor{red}{m}_t \cosh(y_i))$

$$\sum_i g_i^\mu$$

$g_t(\cos(\phi_i), \sin(\phi_i),$   
 $\sinh(y_i), \cosh(y_i))$

Need an extra term to account for  $m_t \neq p_t$

## *Back to the concepts: subtraction*

New 4-vector subtraction formula:

$$p_{\text{sub}}^\mu = p^\mu - \rho A^\mu - \rho_m A_m^\mu$$

with

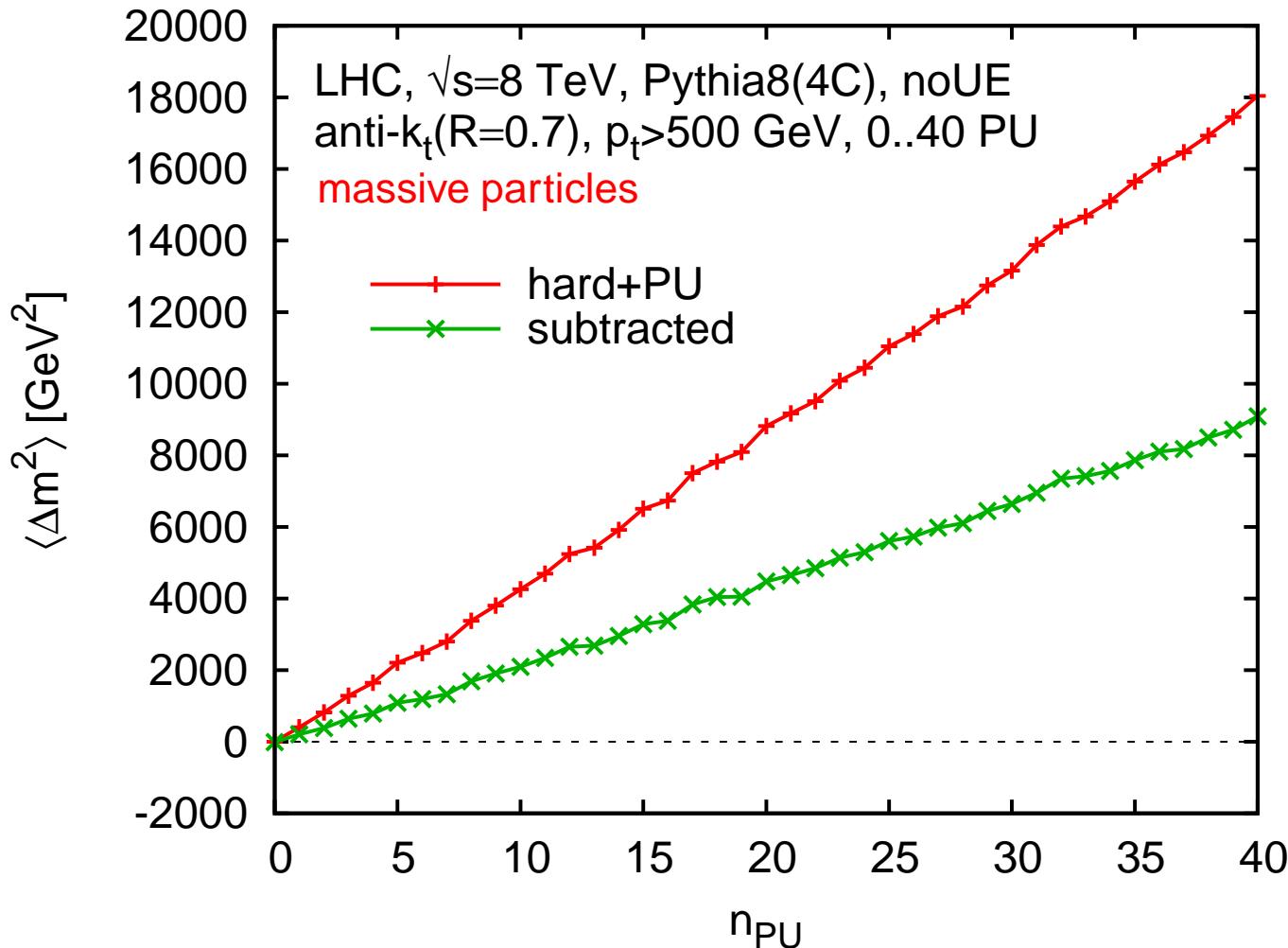
$$\rho = \underset{j \in \text{patches}}{\text{median}} \left\{ \frac{p_{t,j}}{A_t} \right\}$$

$$\rho_m = \underset{j \in \text{patches}}{\text{median}} \left\{ \frac{\sum_{i \in j} m_{t,i} - p_{t,i}}{A} \right\}$$

$$A_m^\mu \equiv (0, 0, A_z, A_E)$$

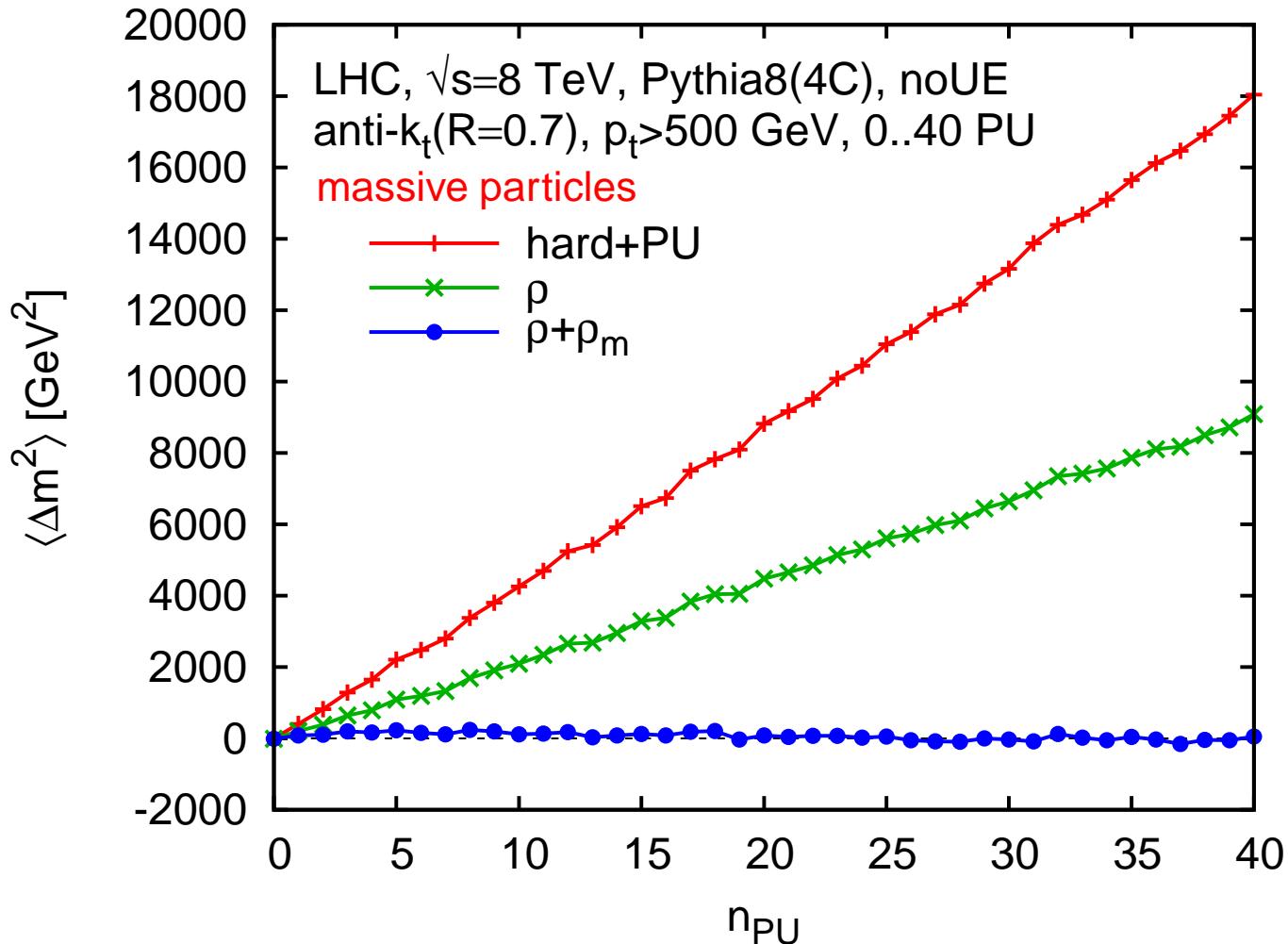
# Jet mass subtraction revisited

$$p_{\text{sub}}^\mu = p^\mu - \rho A^\mu$$



# Jet mass subtraction revisited

$$p_{\text{sub}}^\mu = p^\mu - \rho A^\mu - \rho_m A_m^\mu$$



# *Usage in tools*

---

Subtraction can be used internally in tools

Examples:

- subtract each subject before selecting the  $n$  hardest in filtering
- subtract each subject before applying the  $p_t$  cut in trimming
- subtract at each unclustering step in pruning

# Usage in tools

Subtraction can be used internally in tools

grooming+subtraction  
can prove very powerful

