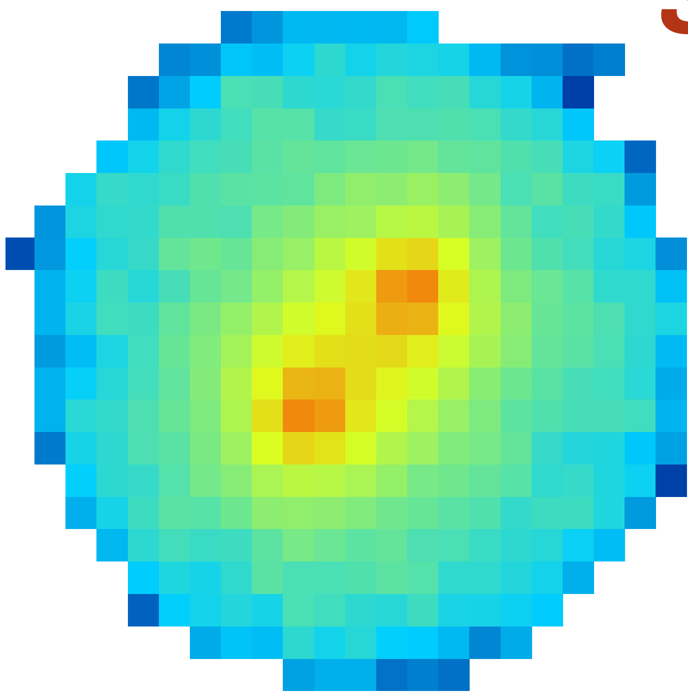
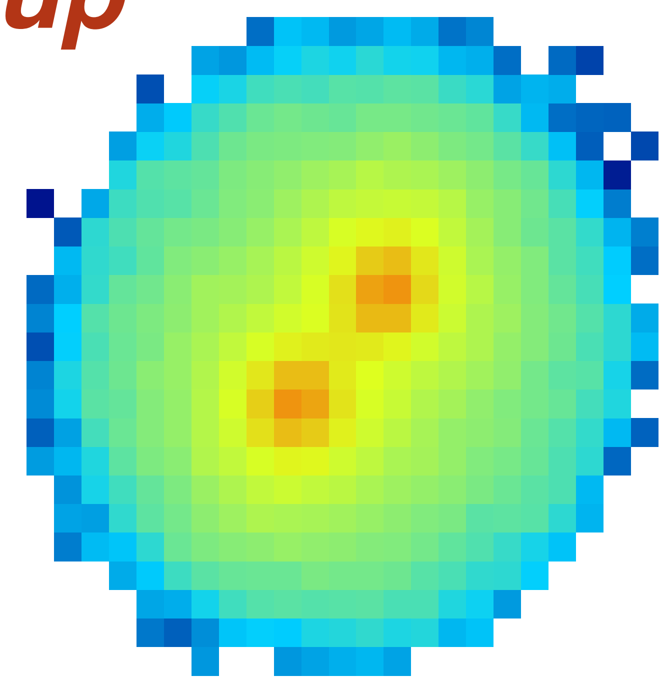
Machine learning, computer vision, and probabilistic models in jet physics

Luke de Oliveira[1], Michael Kagan[2], Carolyn Kim[3], Lester Mackey[3], Benjamin Nachman[2], Francesco Rubbo[2], Conrad Stansbury[2], Ariel Schwartzman[2], Michael Zhu[3]



SLAC-Stanford HEP-ML group



Data Science @ LHC

Thursday, November 12, 2015

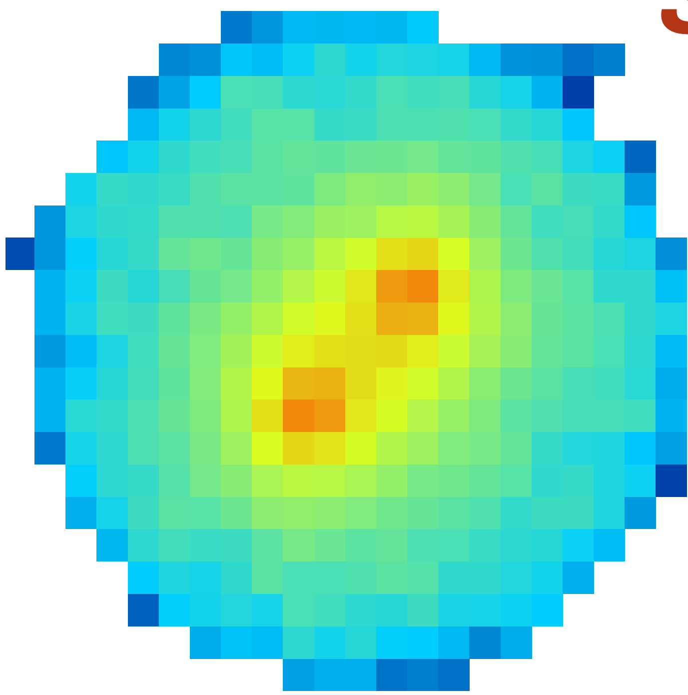
[1]*Stanford Institute for Computational and Mathematical Engineering (ICME)*
[2]*SLAC National Accelerator Center, Stanford University*
[3]*Department of Statistics, Stanford University*

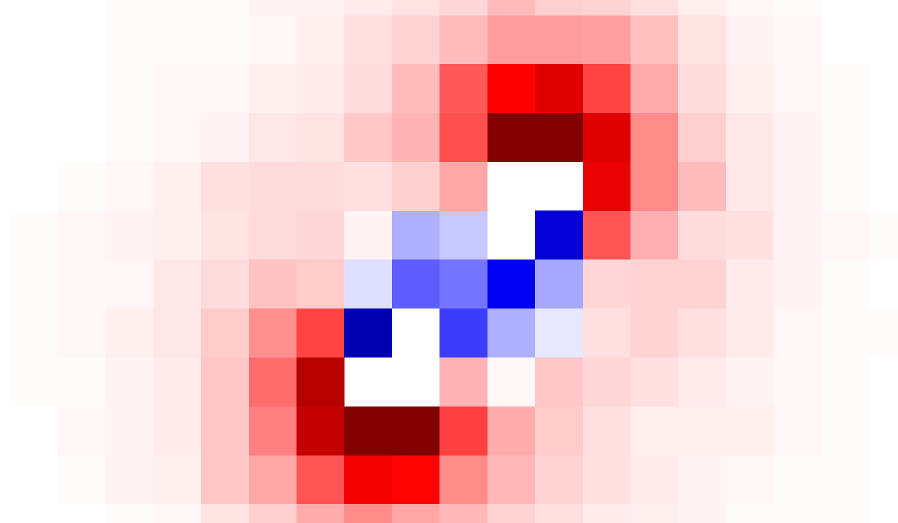
also known as…

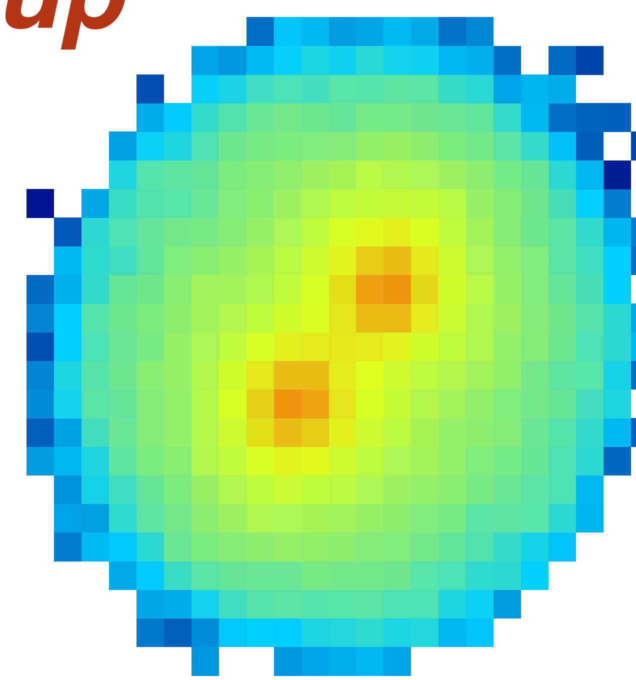
*What can we **learn** from machine **learning** about jets?*

Luke de Oliveira[1], Michael Kagan[2], Carolyn Kim[3], Lester Mackey[3], Benjamin Nachman[2], Francesco Rubbo[2], Conrad Stansbury[2], Ariel Schwartzman[2], Michael Zhu[3]

SLAC-Stanford HEP-ML group



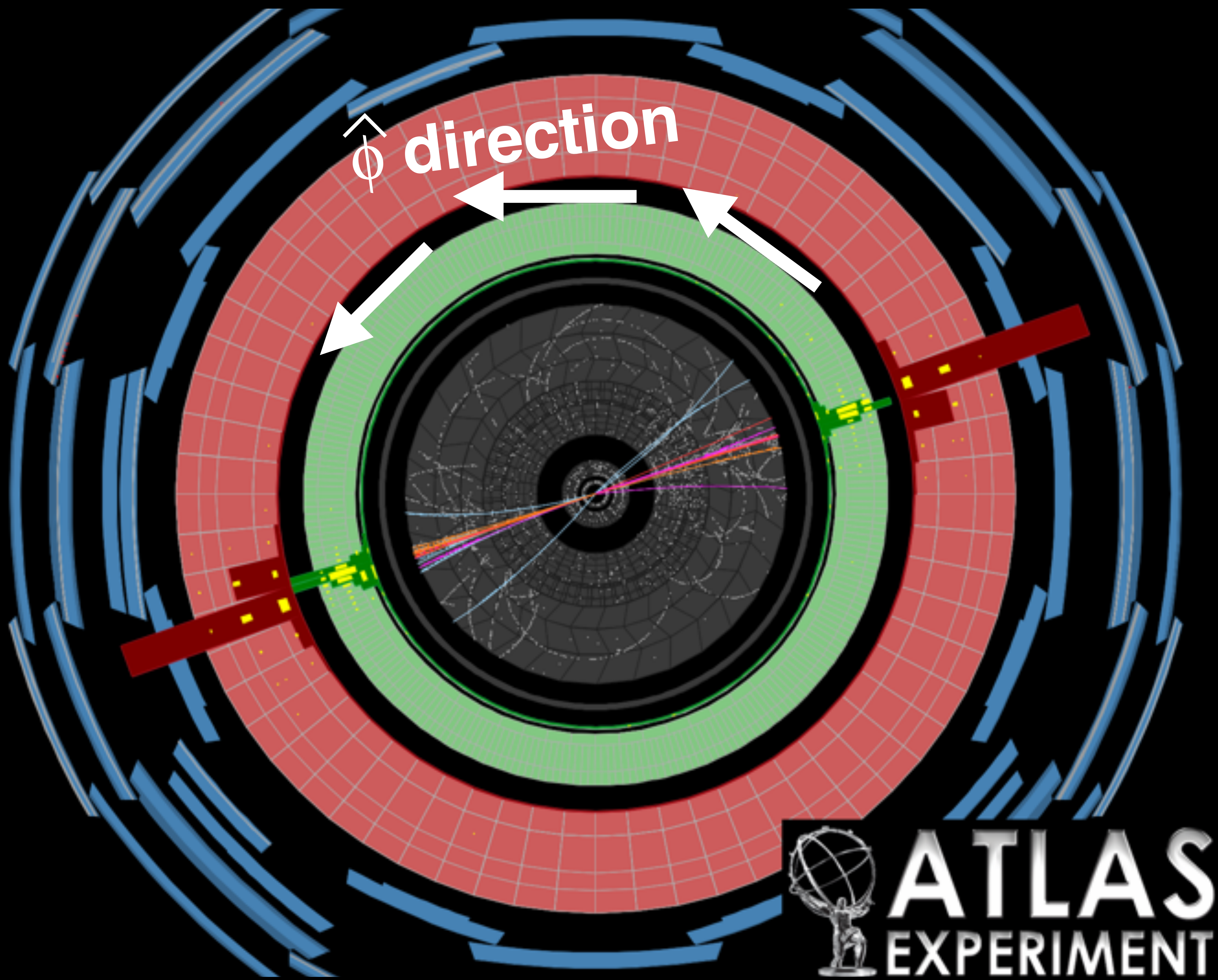




Data Science @ LHC

Thursday, November 12, 2015

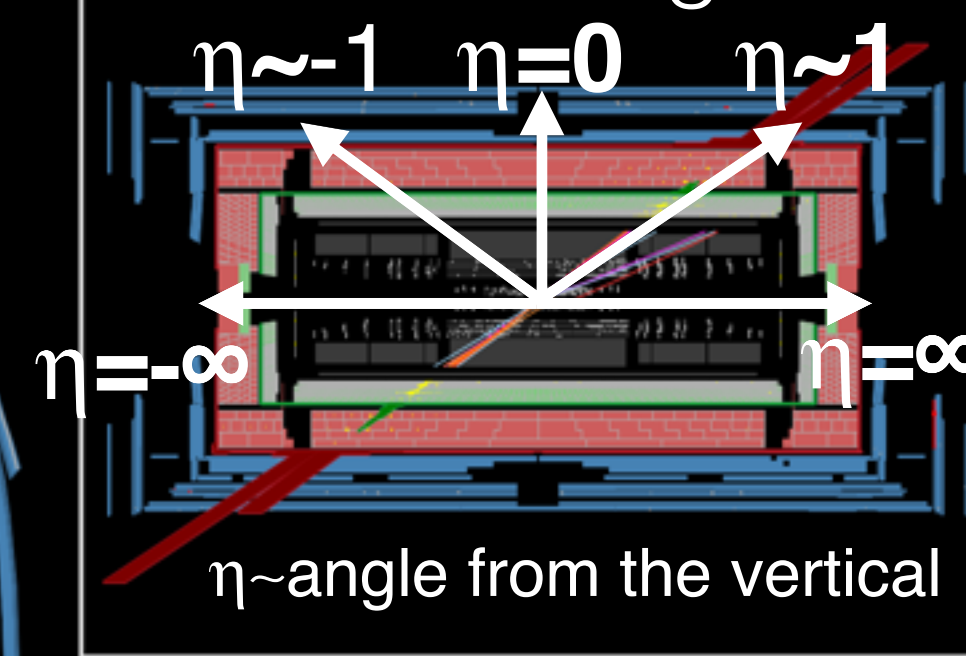
[1]*Stanford Institute for Computational and Mathematical Engineering (ICME)*
[2]*SLAC National Accelerator Center, Stanford University*
[3]*Department of Statistics, Stanford University*

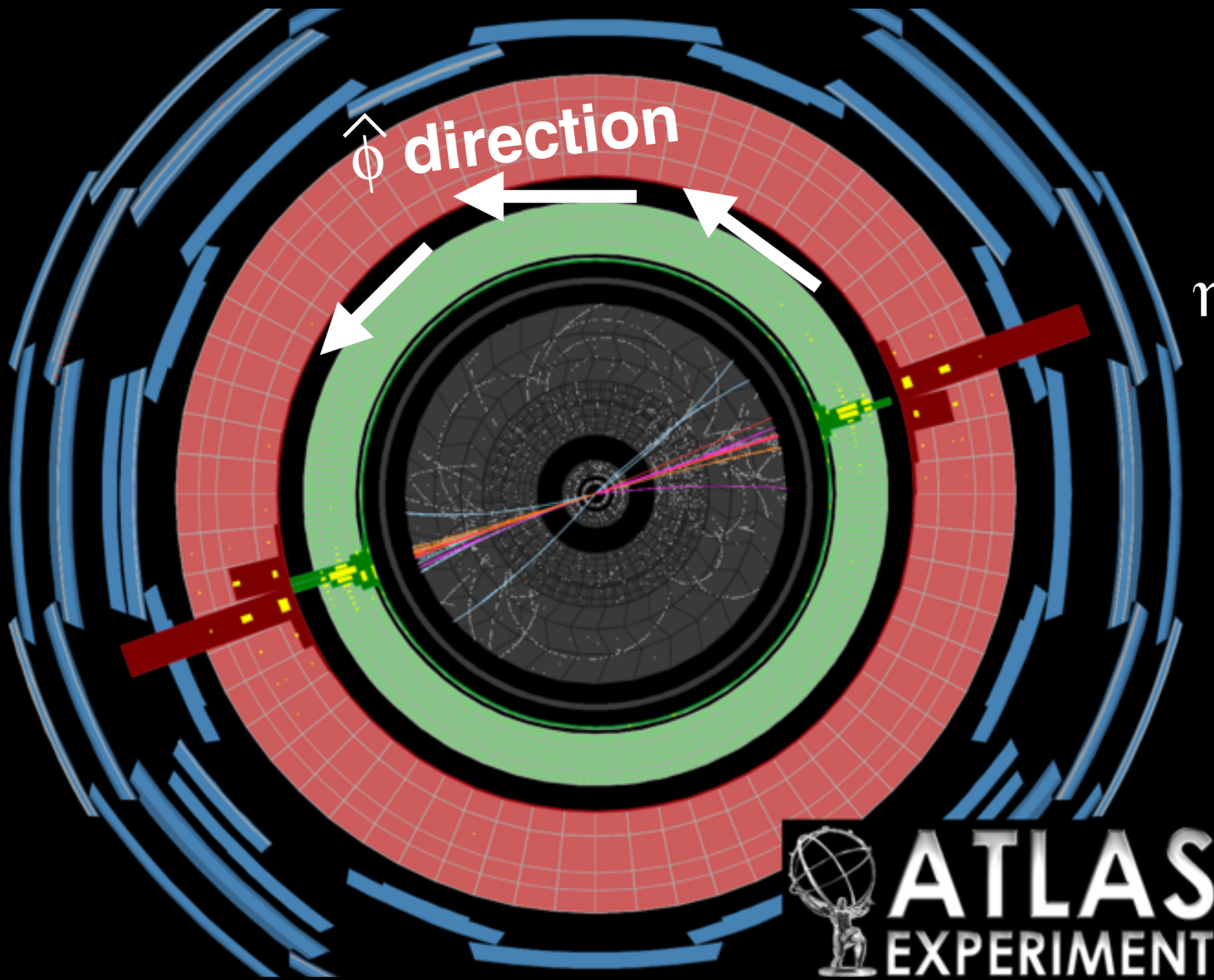


beam into and
out of the page

Orientation Part I

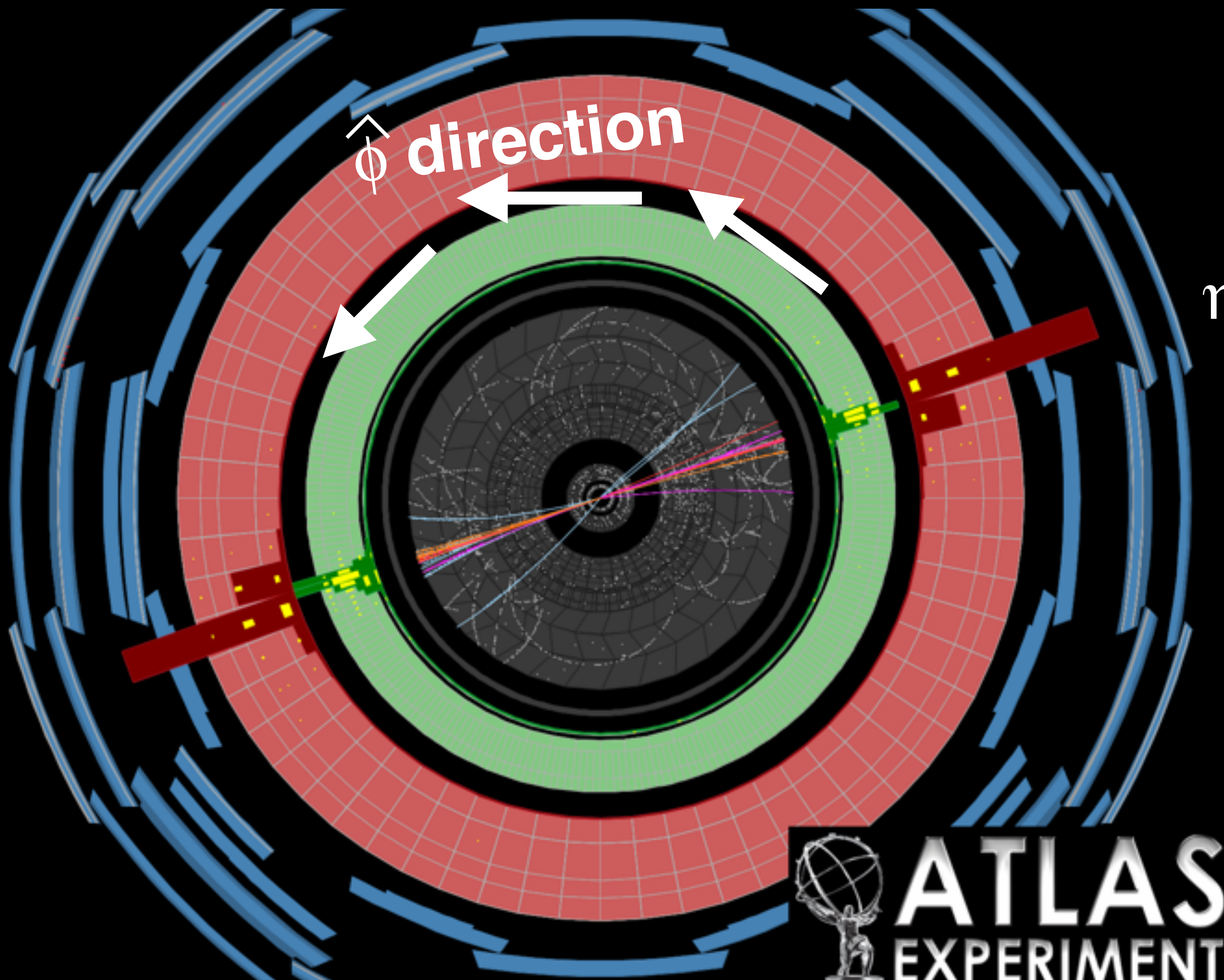
beam from
left and right





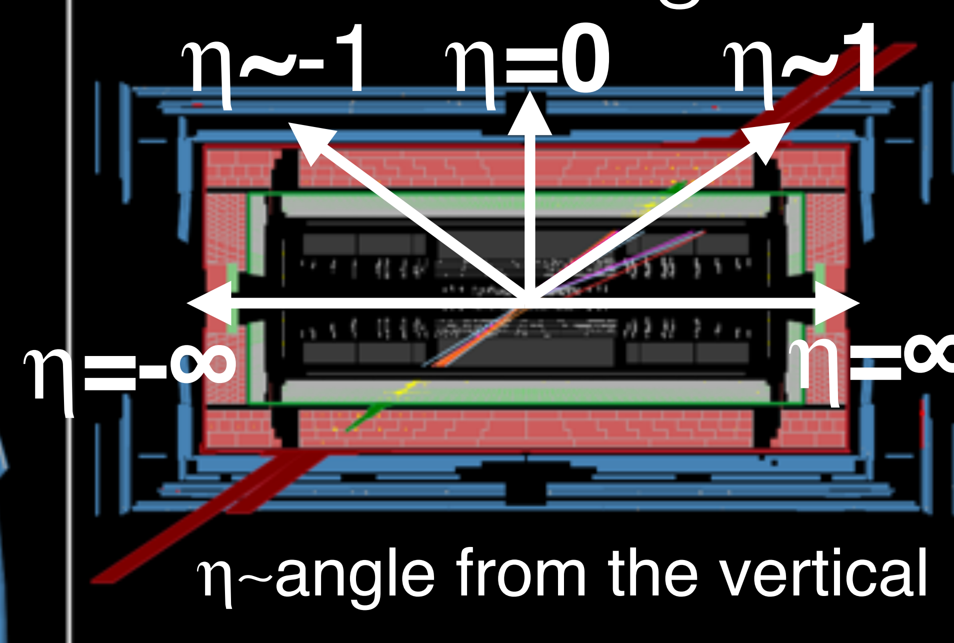
beam into and
out of the page

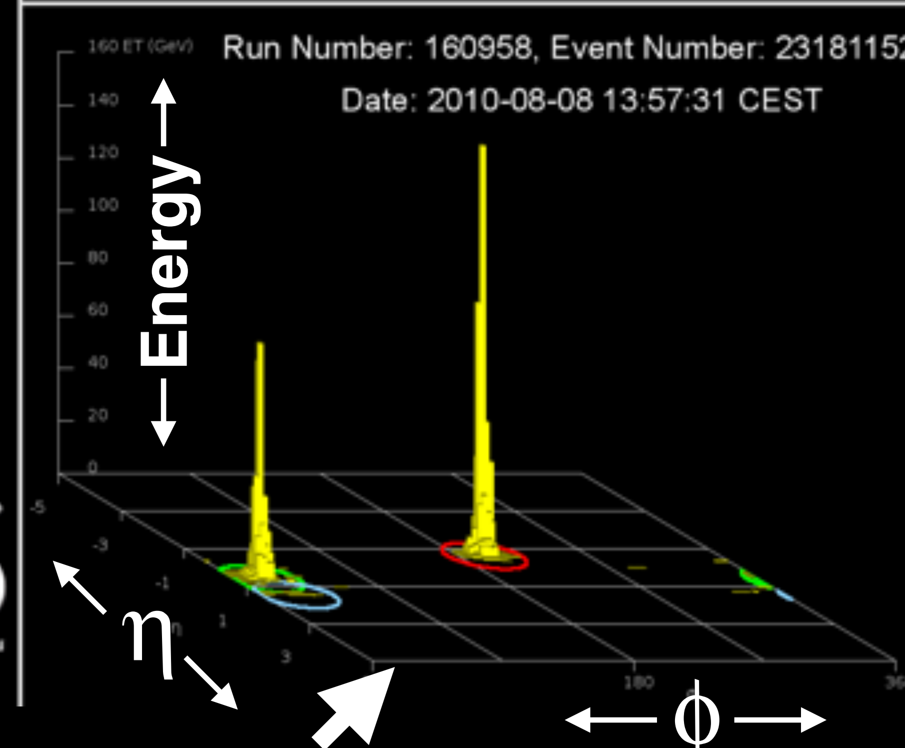
Orientation Part I



beam into and out of the page

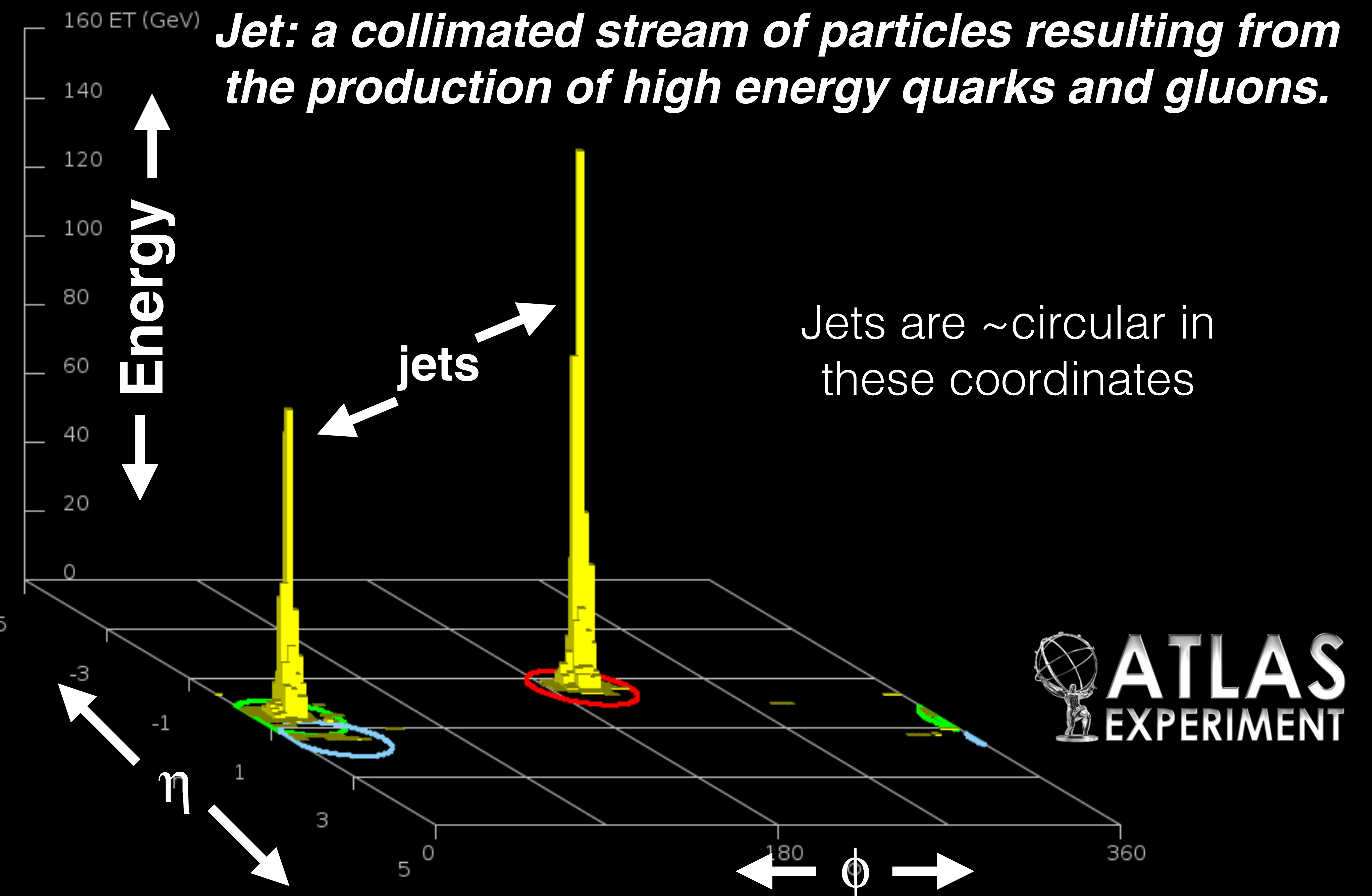
beam from left and right





“Unroll” the calorimeter - this is where we naturally think about jets.

Jet: a collimated stream of particles resulting from the production of high energy quarks and gluons.



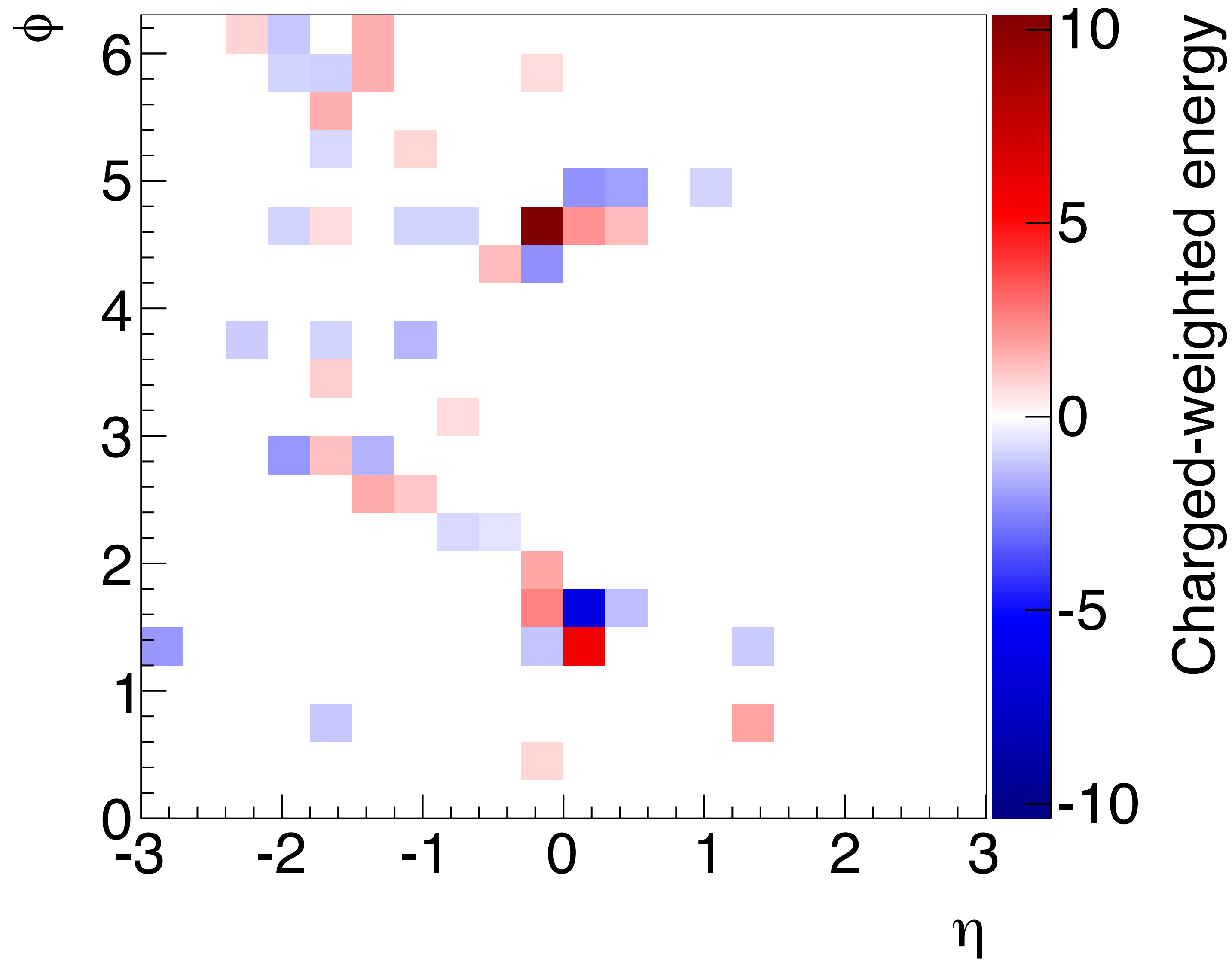
Jets are ~circular in these coordinates

Jet structure contains information about the quarks & gluons

However, jets are not unique!

Jets are defined by clustering algorithms

This is the ~entire calorimeter



Easy

How many jets?
Where are they?
How big are they?
What is their origin?

Hard

Jet structure contains information about the quarks & gluons

However, jets are not unique!

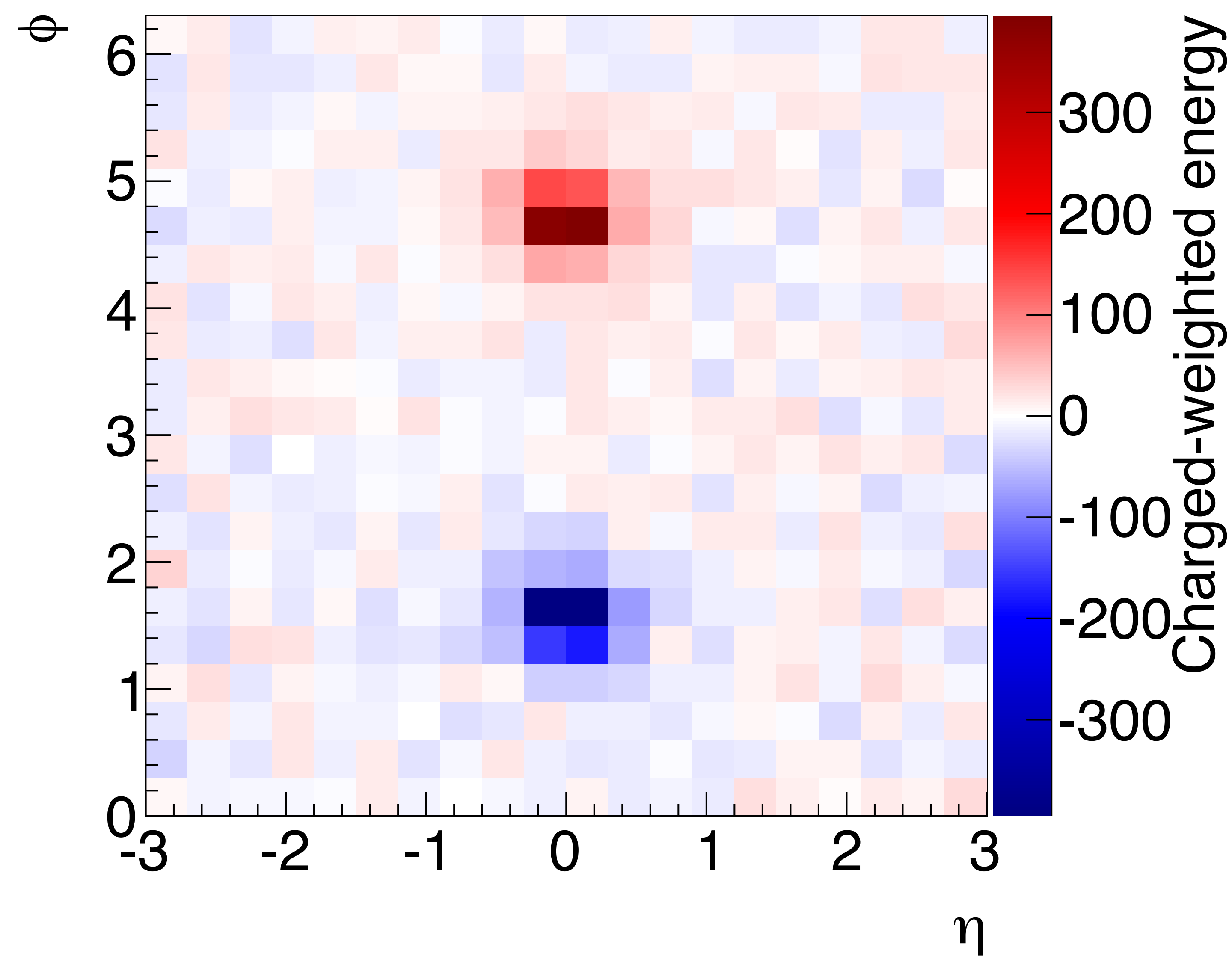
Jets are defined by clustering algorithms

Easy

How many jets?
Where are they?
How big are they?
What is their origin?

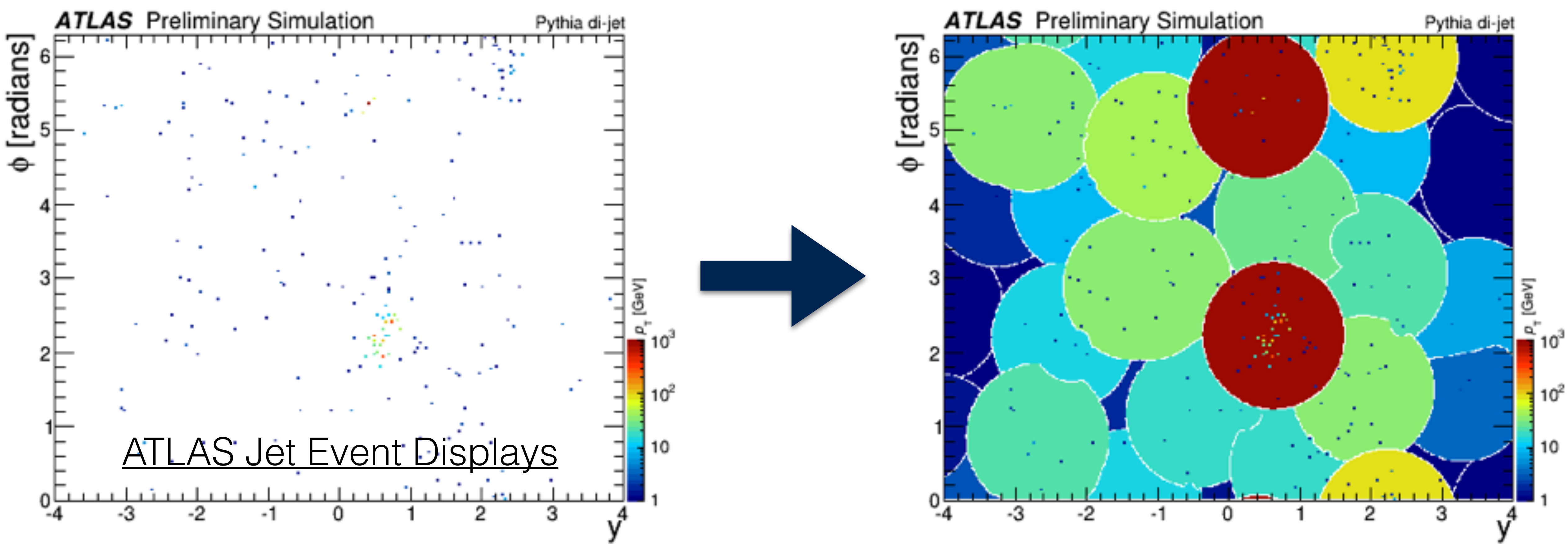
Hard

10k events



ML is no stranger to jet physics - custom unsupervised learning techniques are used to cluster jets.

See e.g. the anti-kt algorithm



However, the extensive ML toolkit can be used to **enhance** and **enrich** the study of jets and their *sub*structure.

Optimization

The bottom line is performance, but also, can we build new, ***better*** *(simple?)* ***features****?*

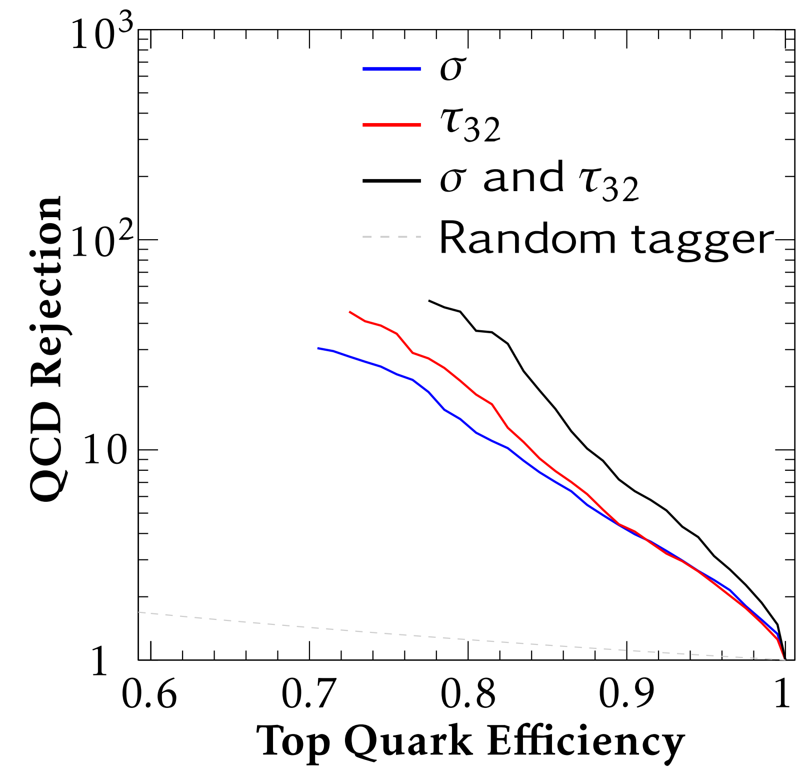


Fig. ROC from Fuzzy jets [1].

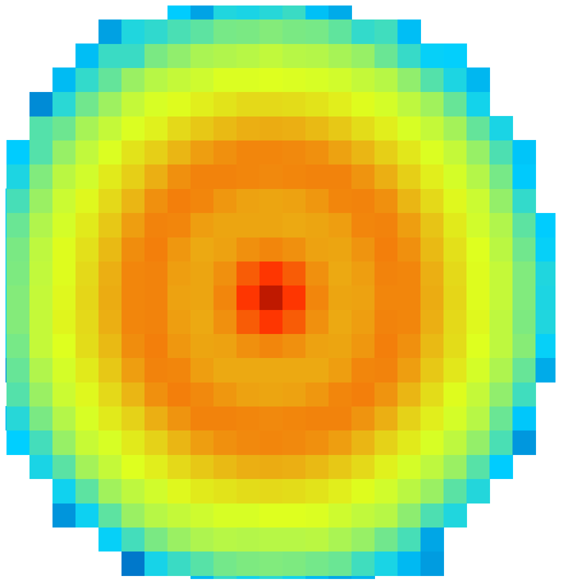


Fig. Unrotated W *jet image*

Teaching the Learning

We don't want the ML to re-learn the basics of special relativity.

Goal: boost performance with ***domain-specific input****.*

Learning from the Learning

The core of our work is to extract information about what the ML is learning. A key component of this is **visualization**.

(You have seen many teasers already!)

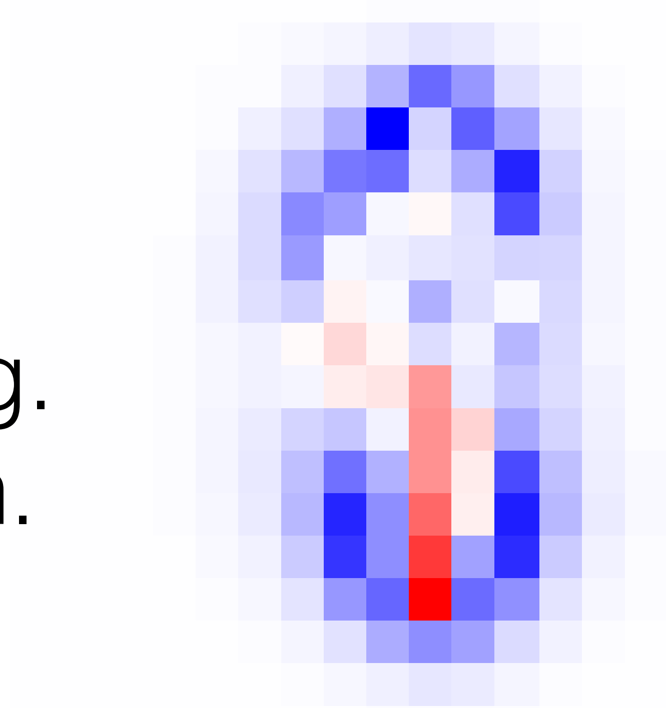


Fig. Fisher jet from W versus QCD [2].

Fixed representations

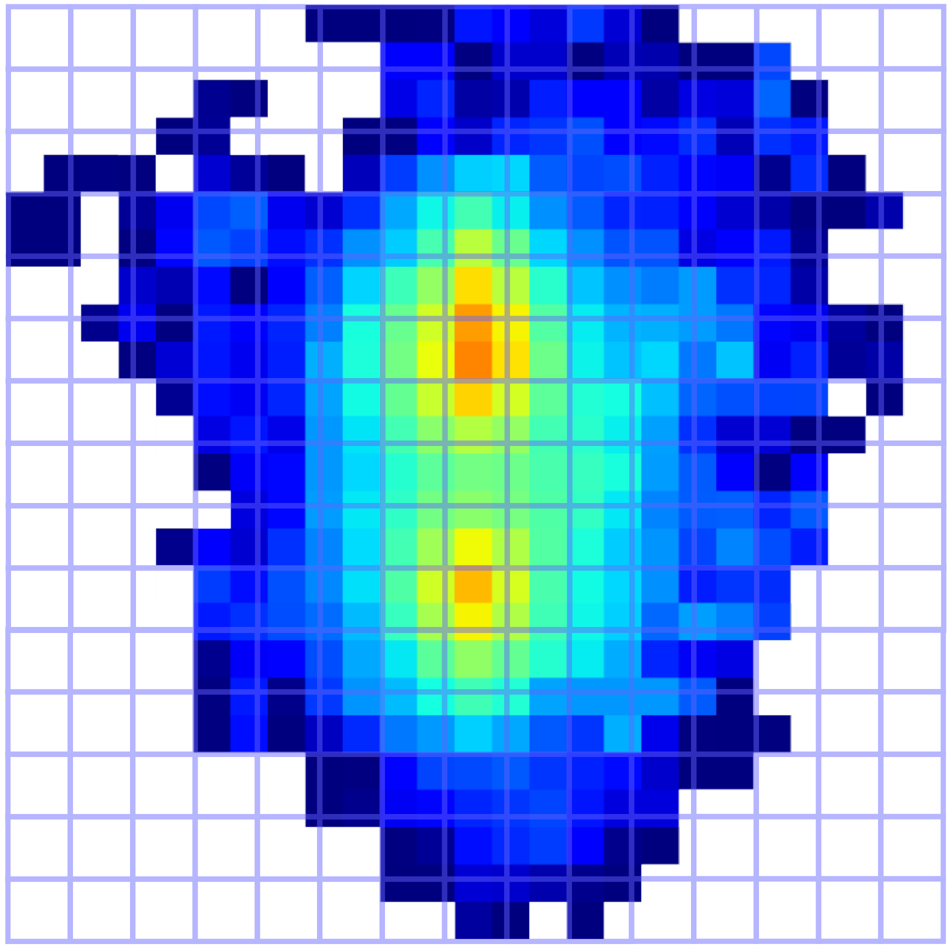
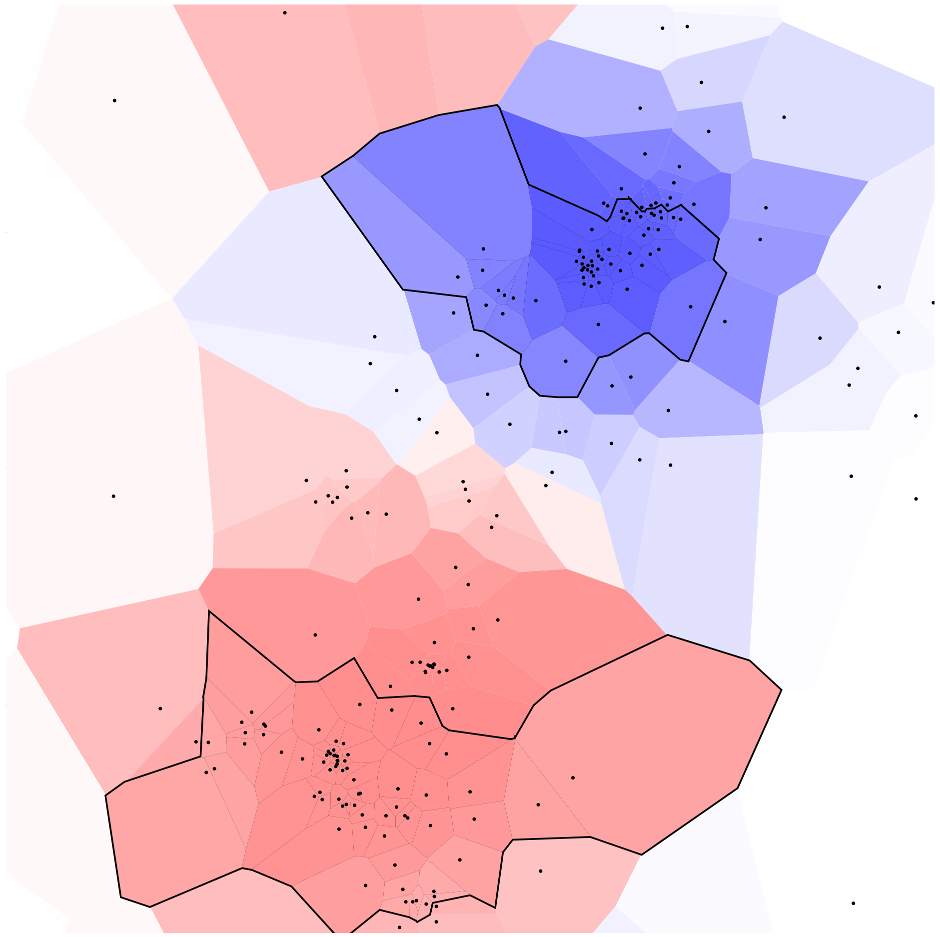


Fig. Processed W *jet image* [2].

Learned representations

Fig. Fuzzy jets from top quark events [1].

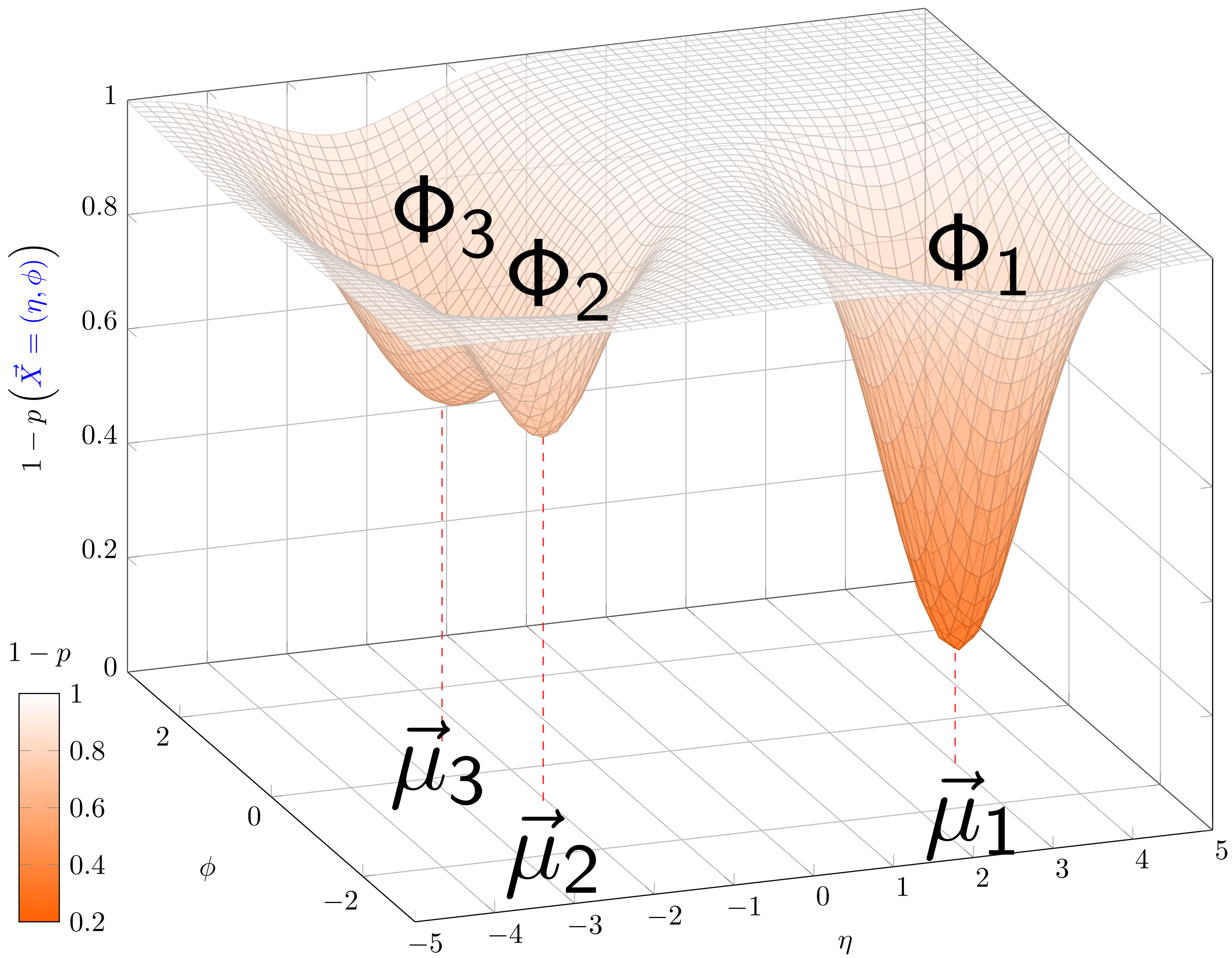


Example: Jet Images

Example: Fuzzy jets

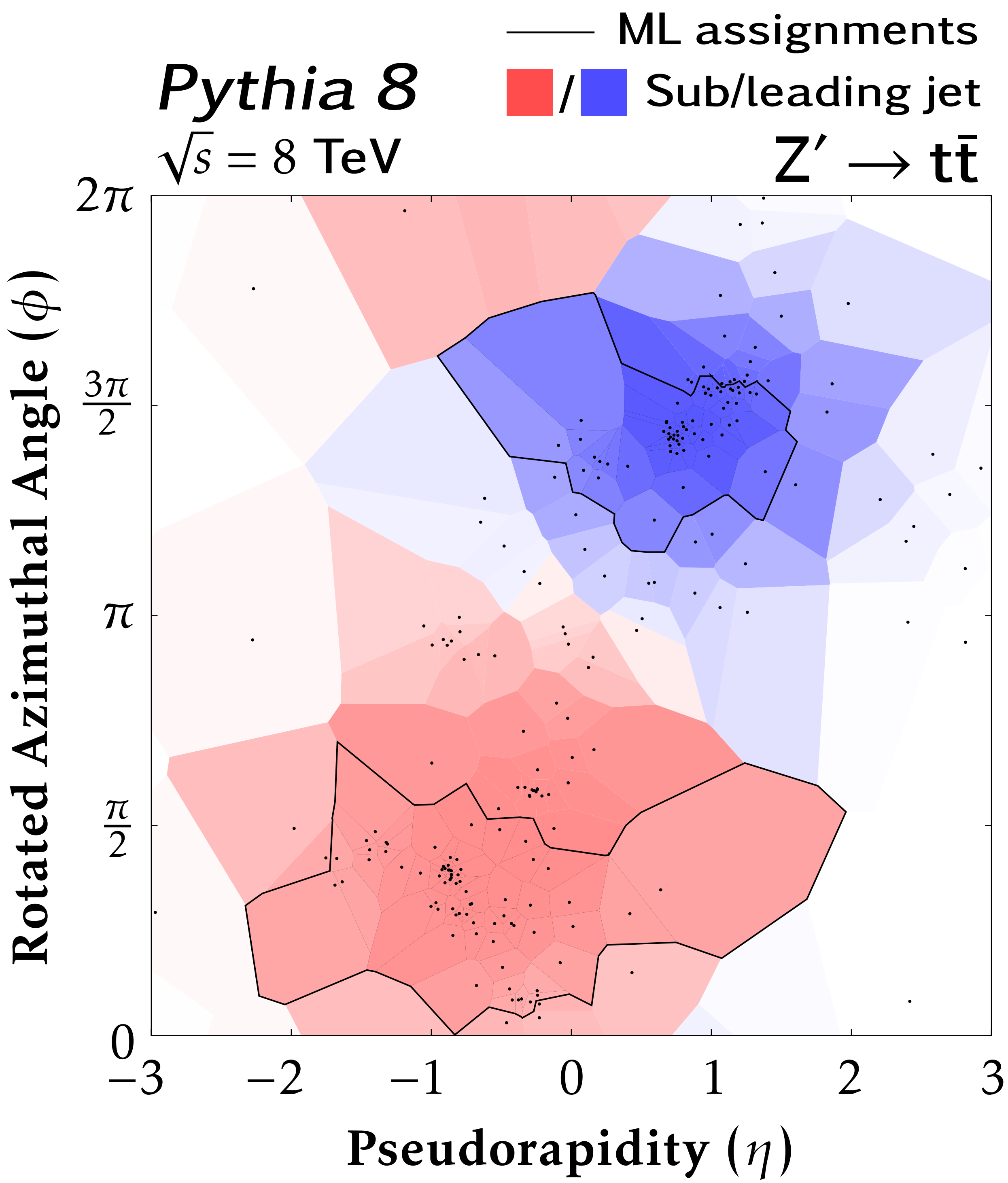
What are fuzzy jets?

Postulate an event likelihood and then minimize given the measured particles.



In machine learning, this is called a ***mixture model***

example likelihood with Gaussians and k = 3



Color intensity = probability of belonging to the red/blue jet

Outcome: location and shape of jets

Compute *Pr(particle i ∊ jet j)*

There are no hard memberships!

One technical slide: the (log) likelihood

Learn: θ and π

jet properties

number of inputs

α = 1 implies **IRC safety**

number of jets

$$\log \mathcal{L}(\{p_{T,i},\rho_i\}|\theta) = \sum_{i=1}^{m} p_{T,i}^{\alpha} \log\left(\sum_{j=1}^{k} \pi_j f(\rho_i|\theta_j)\right),$$

Modification to the usual mixture model paradigm

f = Gaussian (as an example)

π = prior (initialized as uniform)

Domain specific modification

Learn: θ and π

jet properties

number of inputs

number of jets

α = 1 implies **IRC safety**

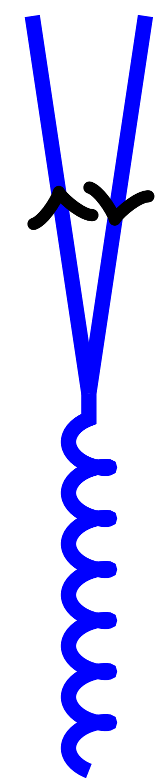
$$\log \mathcal{L}(\{p_{T,i}, \rho_i\}|\theta) = \sum_{i=1}^{m} p_{T,i}^{\alpha} \log\left(\sum_{j=1}^{k} \pi_j f(\rho_i|\theta_j)\right),$$

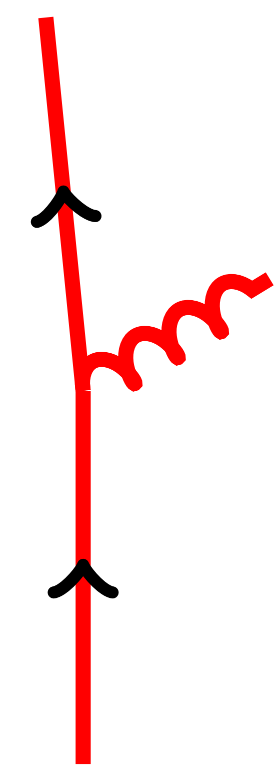
Algorithm must be insensitive to

soft particles (IR-safe)

collinear splittings (C-safe)

This modification does not spoil the ML and ensures IRC safety

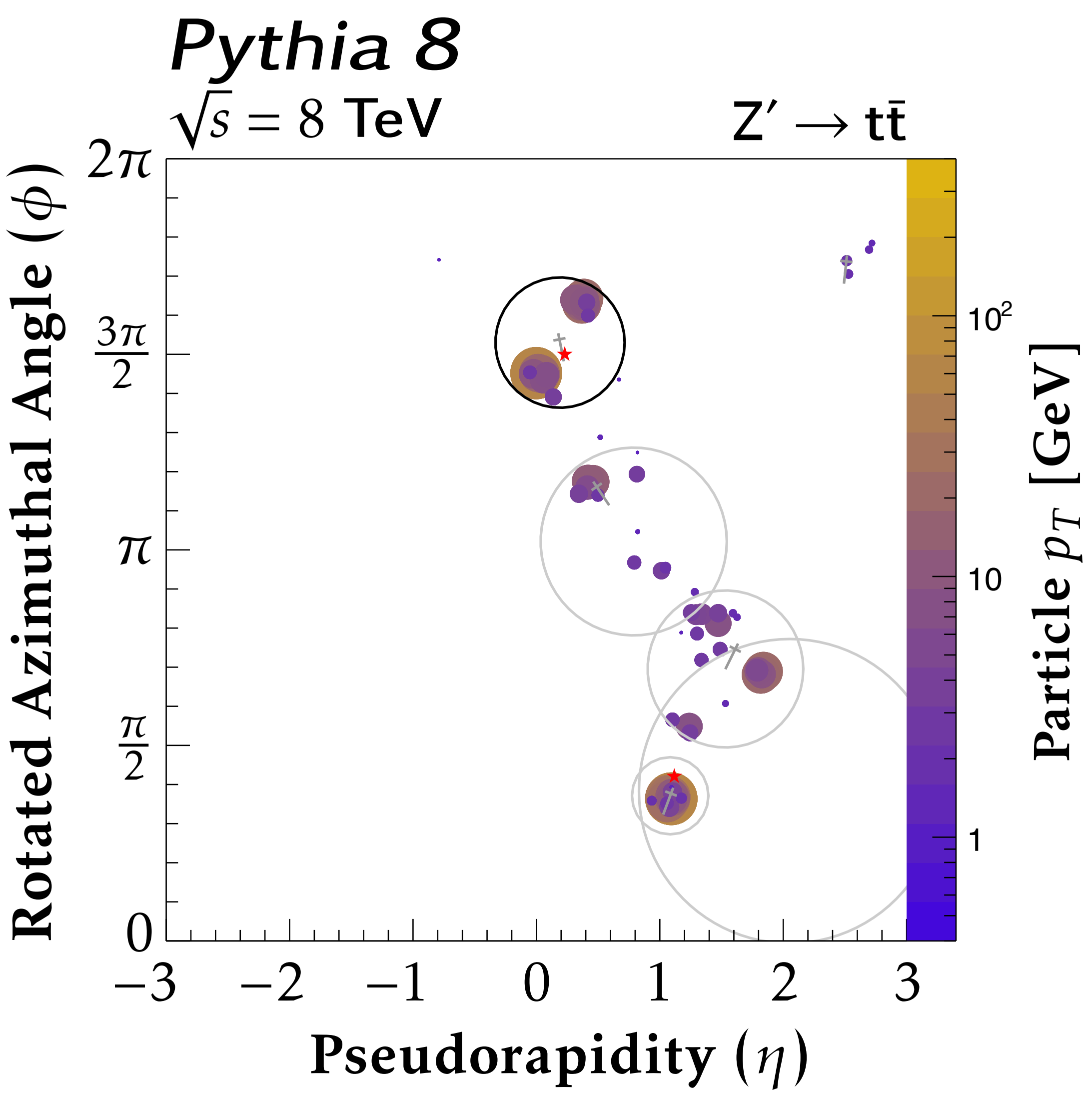




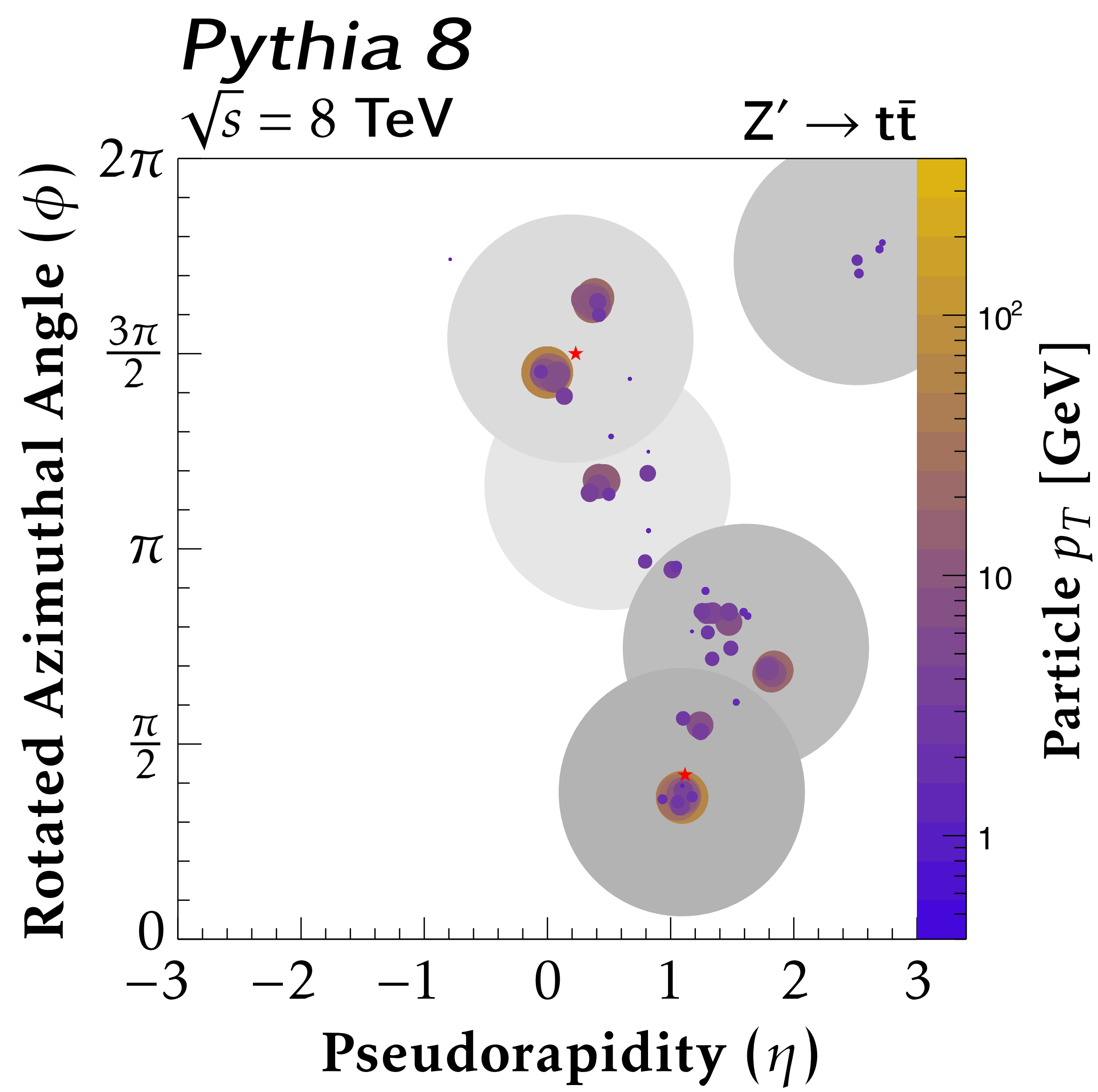
Fuzzy jets vary in size and can overlap!

(in general they can vary in shape, but we using circles here)

Fuzzy Jets

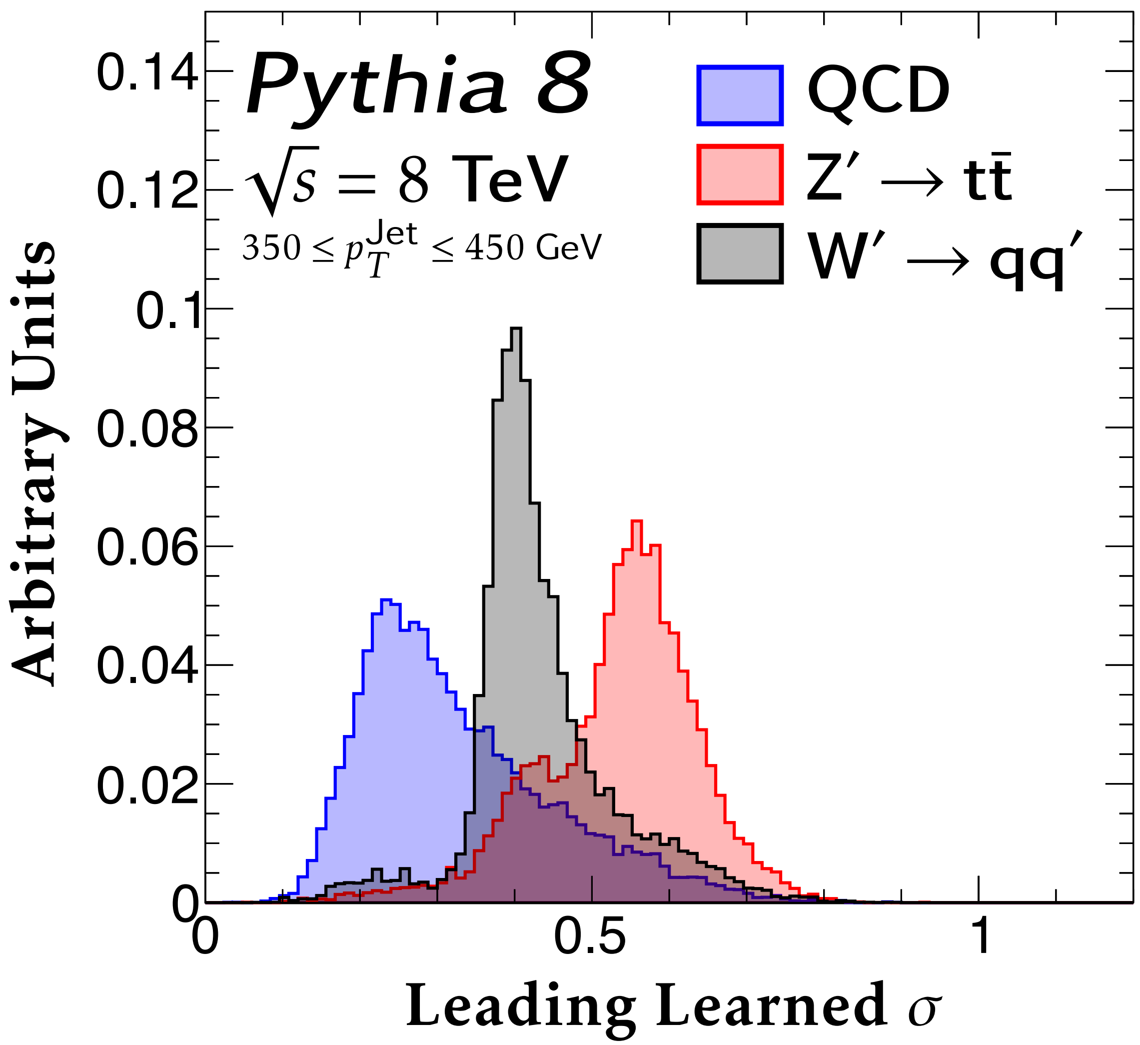


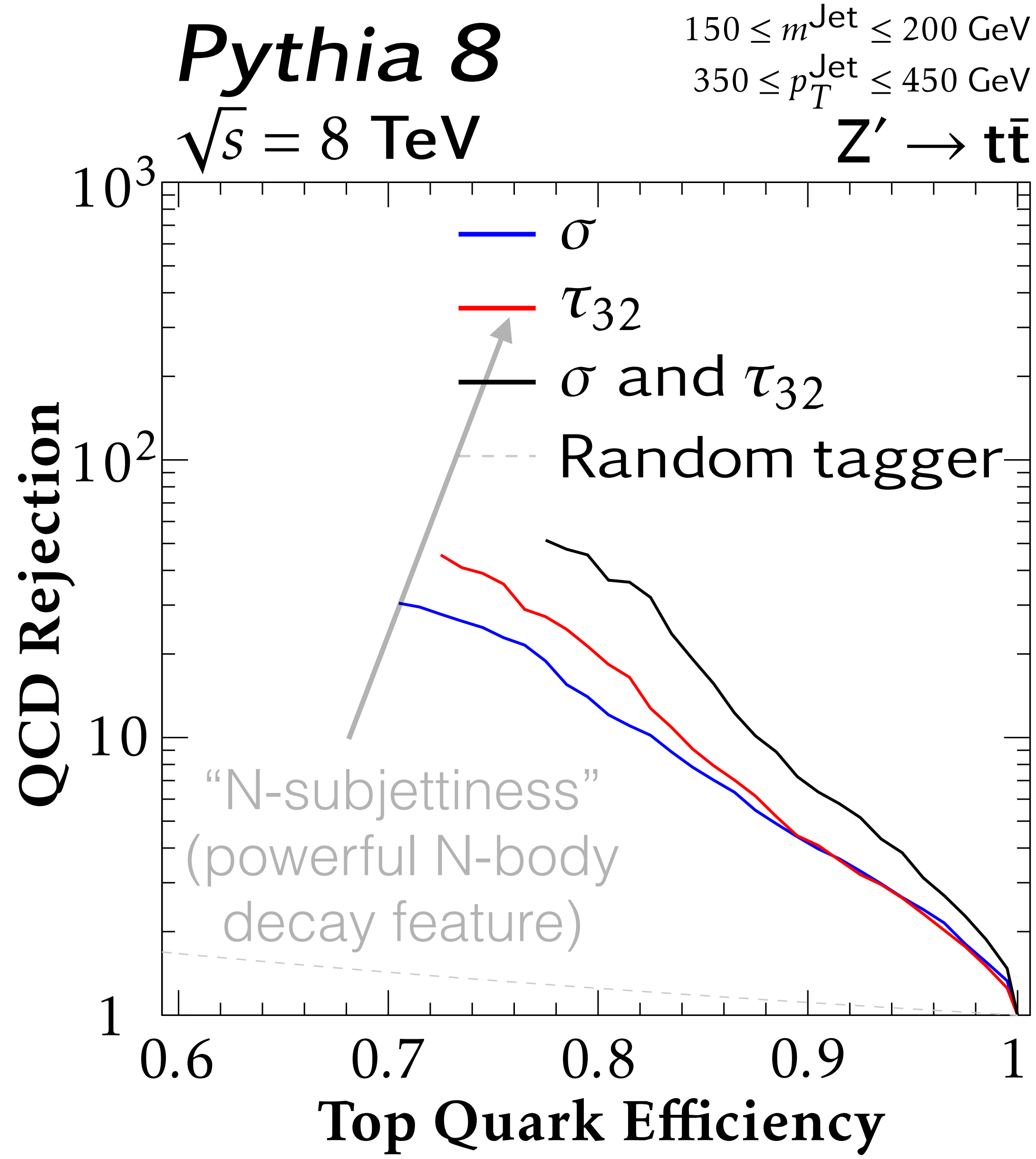
anti-k_t



Initialize the EM algorithm with anti-k_t jets

One useful variable is the size σ of the leading fuzzy jet

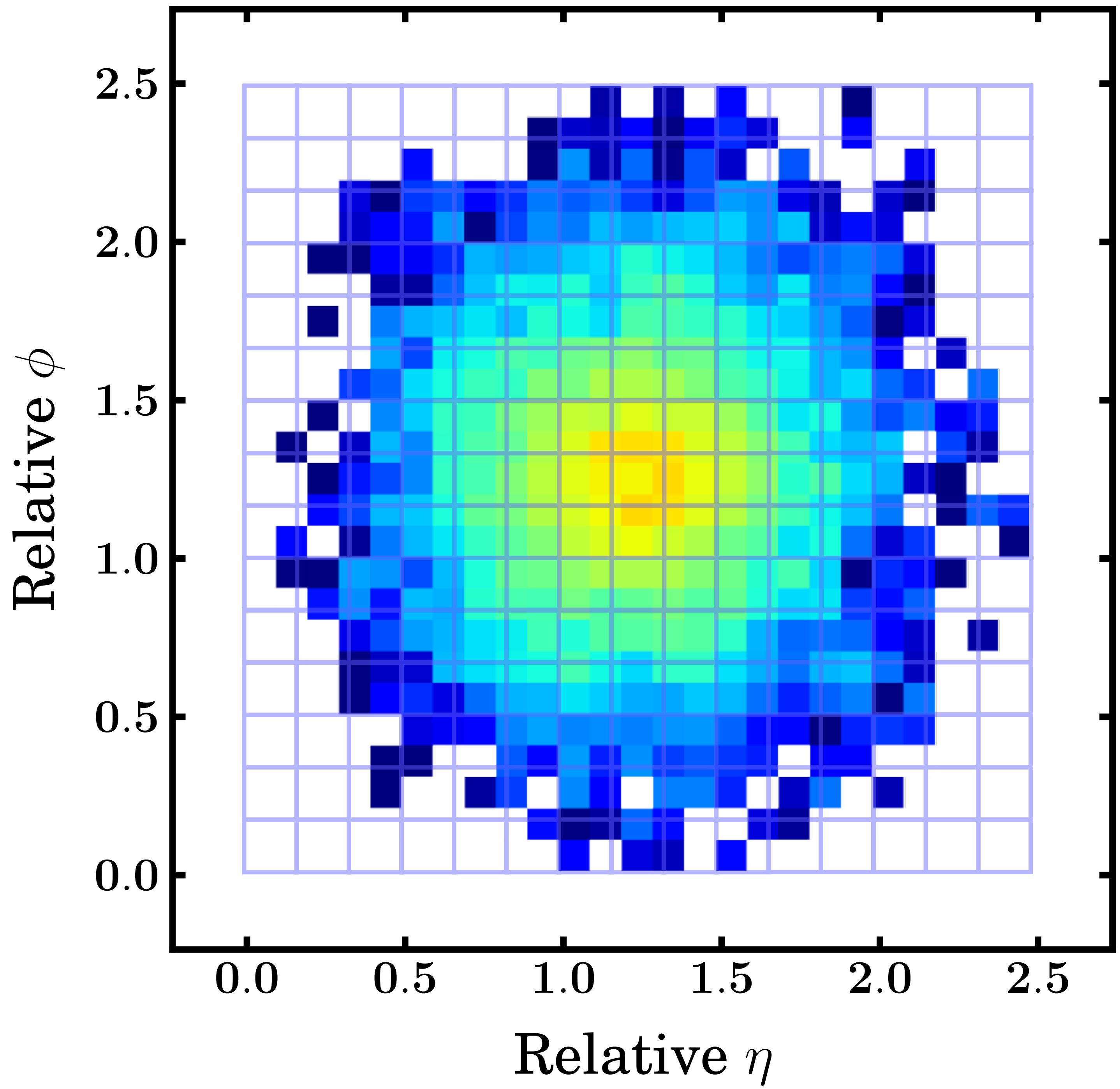




The leading size scale with m/p_T whereas the generic fuzzy jet is rather independent of the process (and is large).

Simple, but powerful paradigm proposed by M. Kagan et al. [2]

Idea: use **image processing** techniques on **jets**!



Pre-processing and the symmetries of space-time

Pre-process

(Pixelate)

Real detectors are already pixelated!

Translations in η are **boosts** along z

Translate

Radiation is symmetric about the jet axis

Translations in ϕ are **rotations** in space

Rotate

Re-grid

Need to convert the rotated grid into a grid!

Flip

Parity symmetry about the jet axis

Pre-processing and the symmetries of space-time

Translations in η are **boosts** along z

Translations in ϕ are **rotations** in space

Need to convert the rotated grid into a grid!

Pre-process

(Pixelate)

Translate

Rotate

Re-grid

Flip

Real detectors are already pixelated!

Radiation is symmetric about the jet axis

Parity symmetry about the jet axis

Pre-processing and the symmetries of space-time

Pre-process

Real detectors are already pixelated!

(Pixelate)

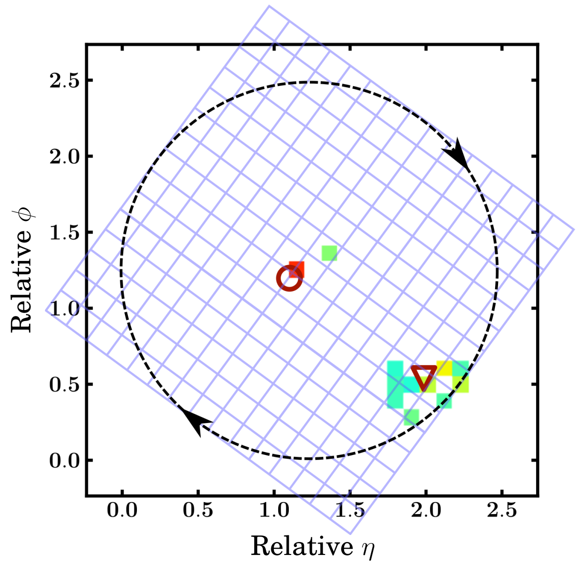
Translations in η are **boosts** along z

Translate

Radiation is symmetric about the jet axis

Translations in ϕ are **rotations** in space

Rotate



Re-grid

Need to convert the rotated grid into a grid!

Flip

Parity symmetry about the jet axis

Pre-processing and the symmetries of space-time

Pre-process

Real detectors are already pixelated!

(Pixelate)

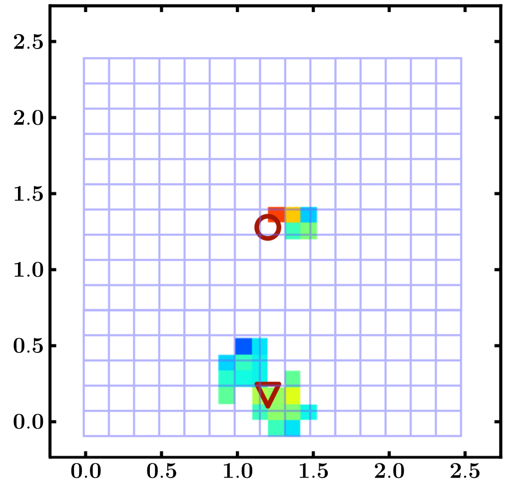
Translations in η are **boosts** along z

Translate

Radiation is symmetric about the jet axis

Translations in ϕ are **rotations** in space

Rotate



Re-grid

Need to convert the rotated grid into a grid!

Flip

Parity symmetry about the jet axis

Pre-processing and the symmetries of space-time

Pre-process

(Pixelate)

Real detectors are already pixelated!

Translations in η are **boosts** along z

Translate

Radiation is symmetric about the jet axis

Translations in ϕ are **rotations** in space

Rotate

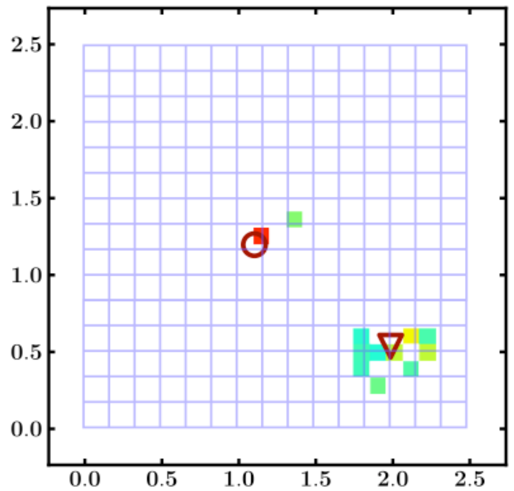
Re-grid

Need to convert the rotated grid into a grid!

Flip

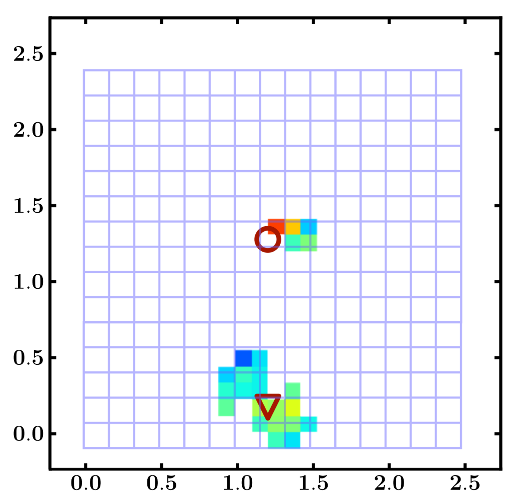
Parity symmetry about the jet axis

Pre-processing and the symmetries of space-time



Translations in η are **boosts** along z

Translations in ϕ are **rotations** in space



Need to convert the rotated grid into a grid!

Pre-process

(Pixelate)

Translate

Rotate

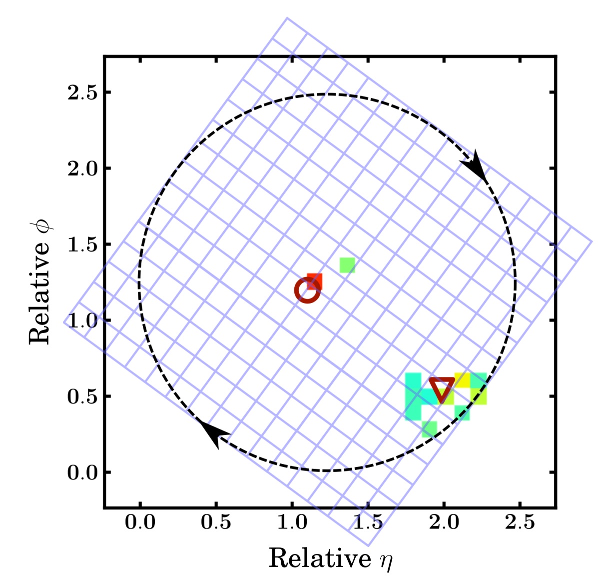
Re-grid

Flip

show one at a time

Real detectors are already pixelated!

Radiation is symmetric about the jet axis



Parity symmetry about the jet axis

Pre-process

(Pixelate)

↓

Translate

↓

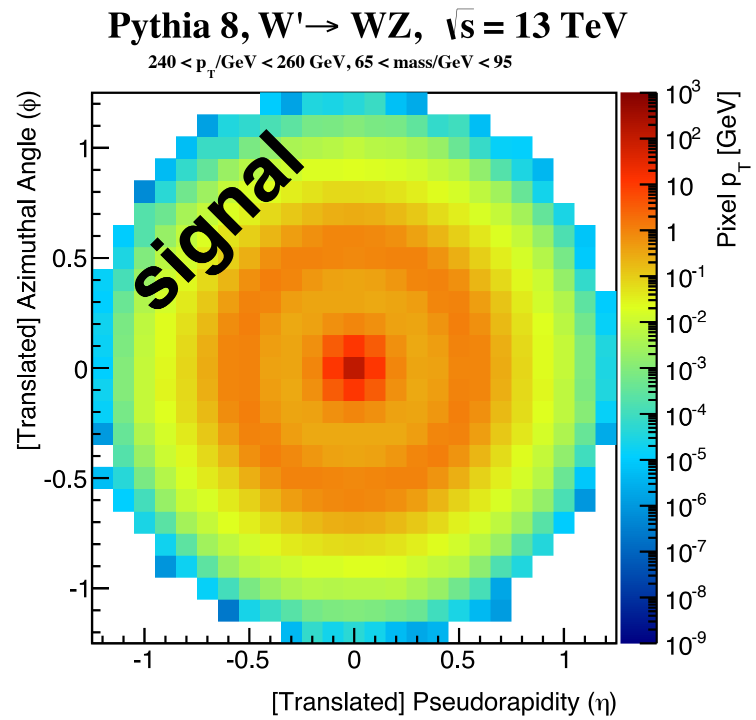
Rotate

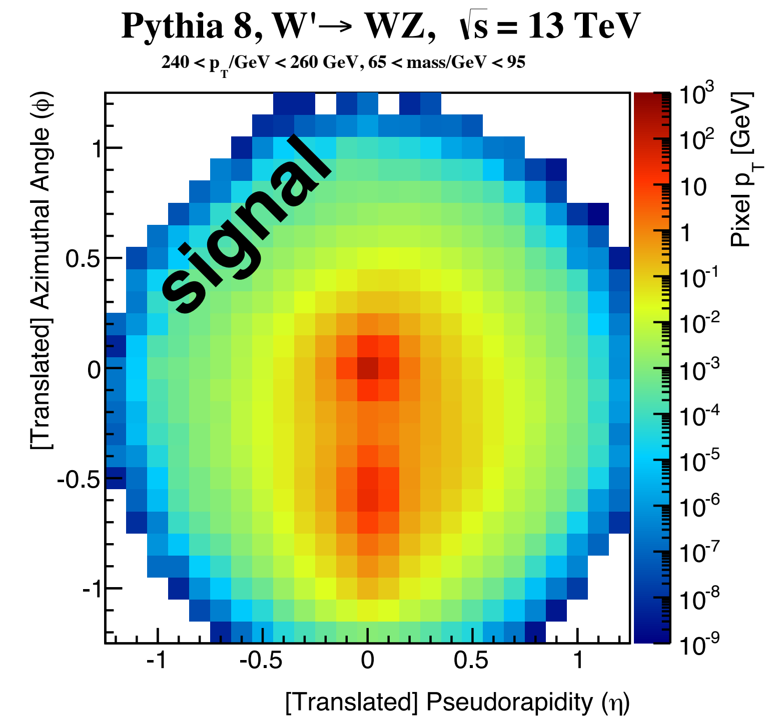
↓

Re-grid

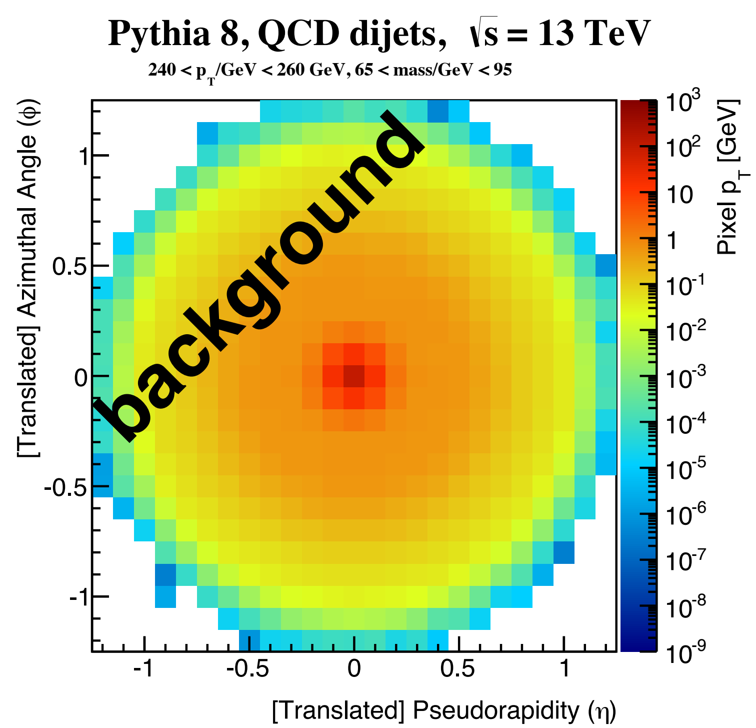
↓

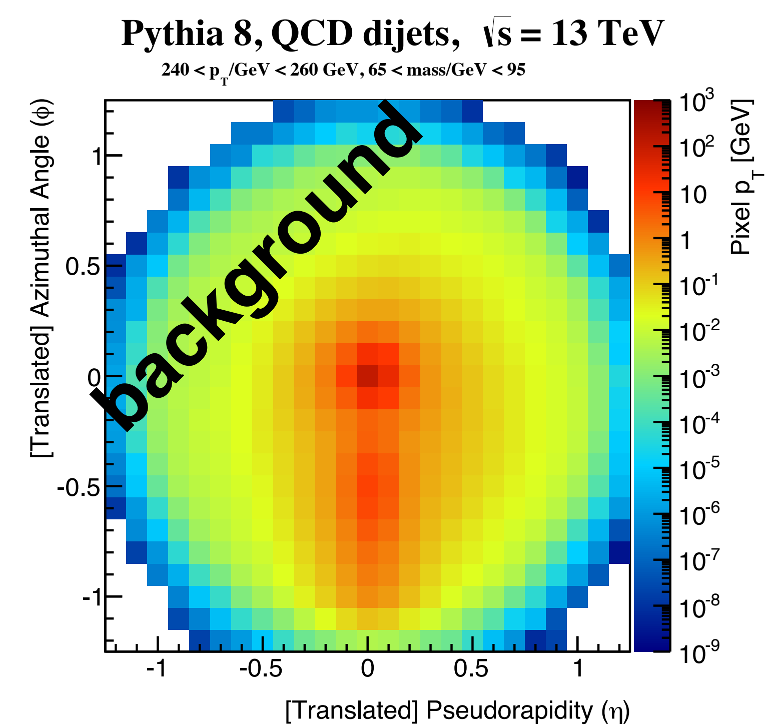
Flip





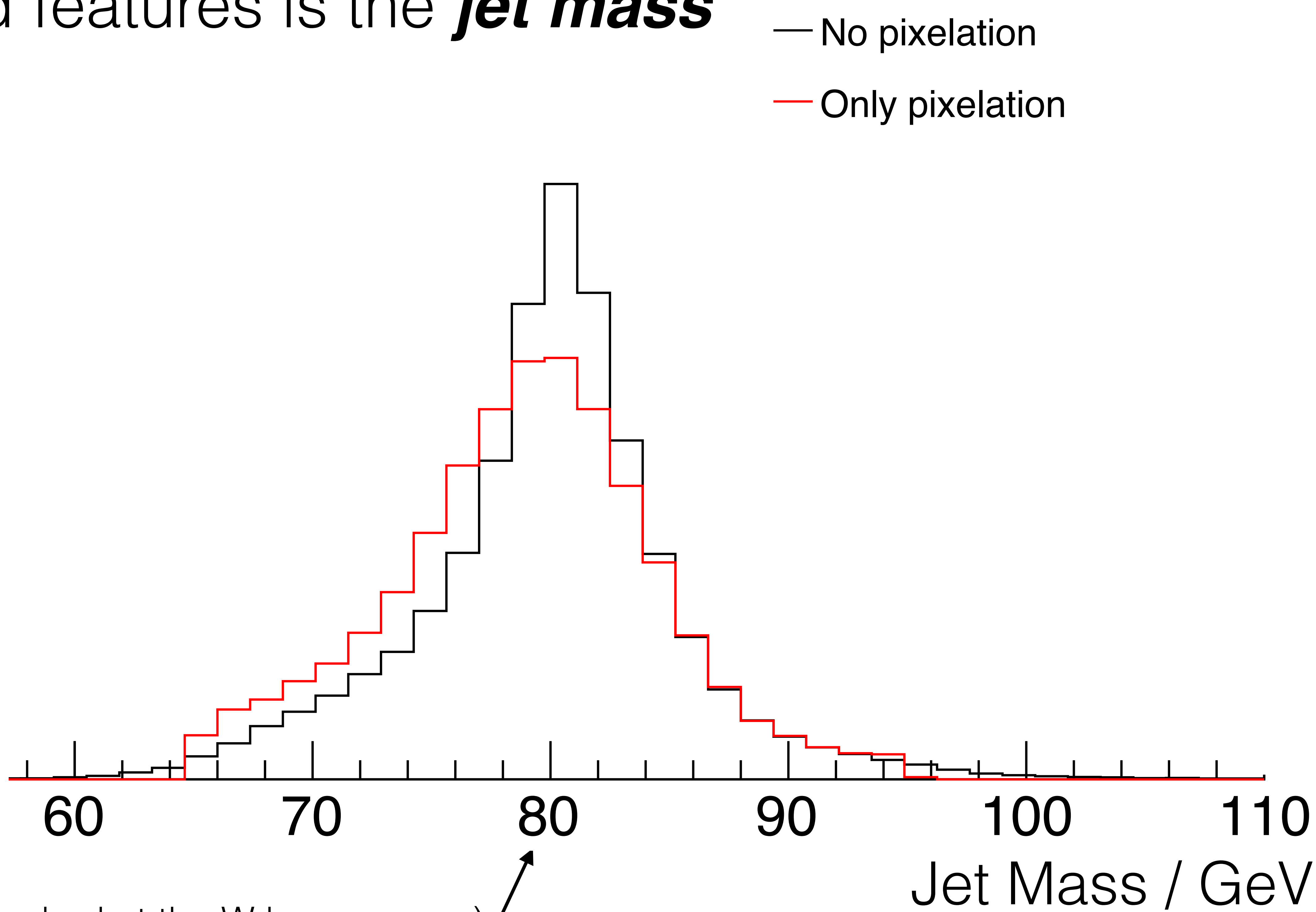
Rotate*, *Re-grid* & *Flip





One of the most useful physics-inspired features is the ***jet mass***

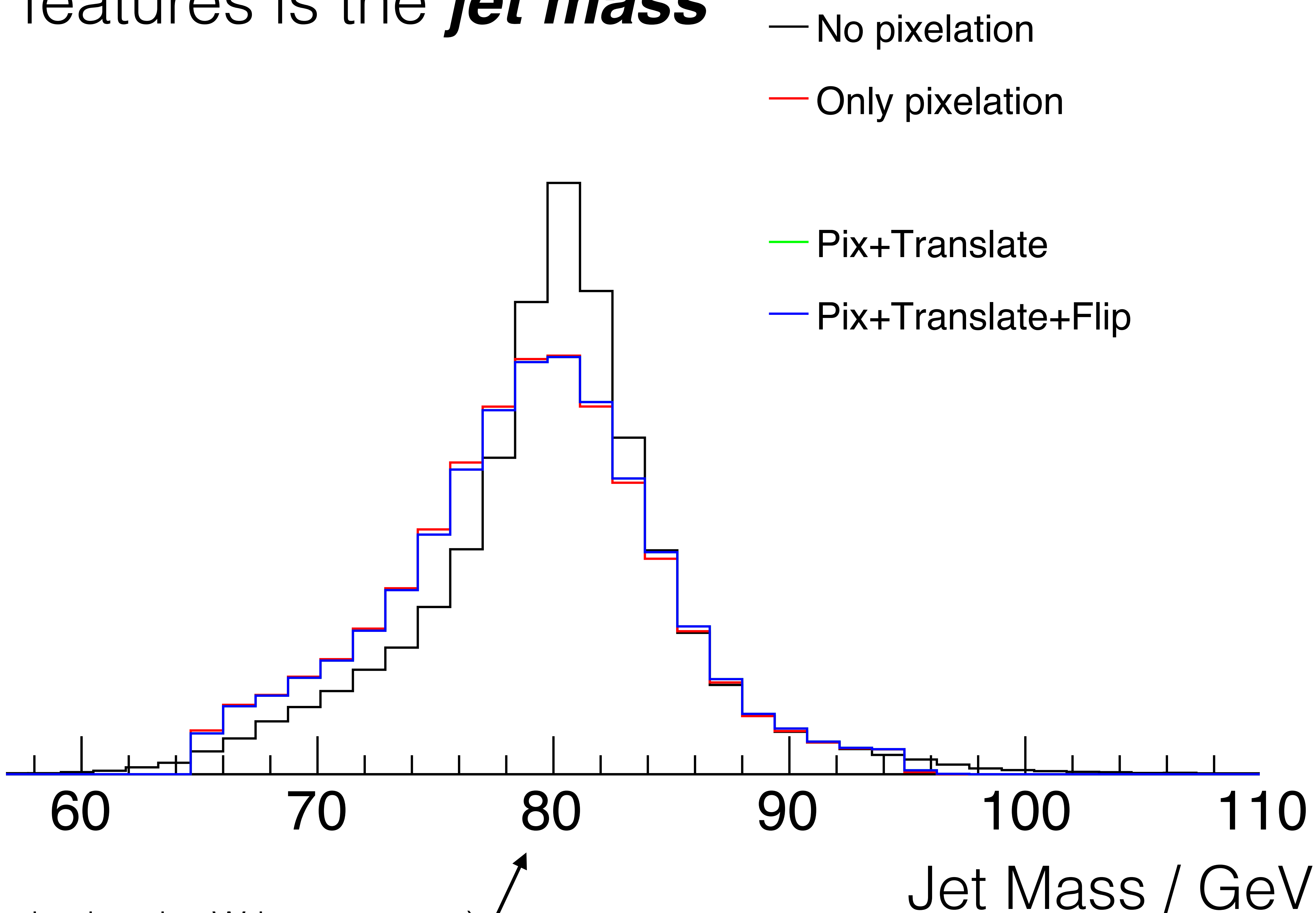
$$m_{jet}^2 \sim \sum_{i<j} E_i E_j \theta_{ij}^2$$



(distribution peaked at the W boson mass)

One of the most useful physics-inspired features is the ***jet mass***

$$m_{jet}^2 \sim \sum_{i<j} E_i E_j \theta_{ij}^2$$

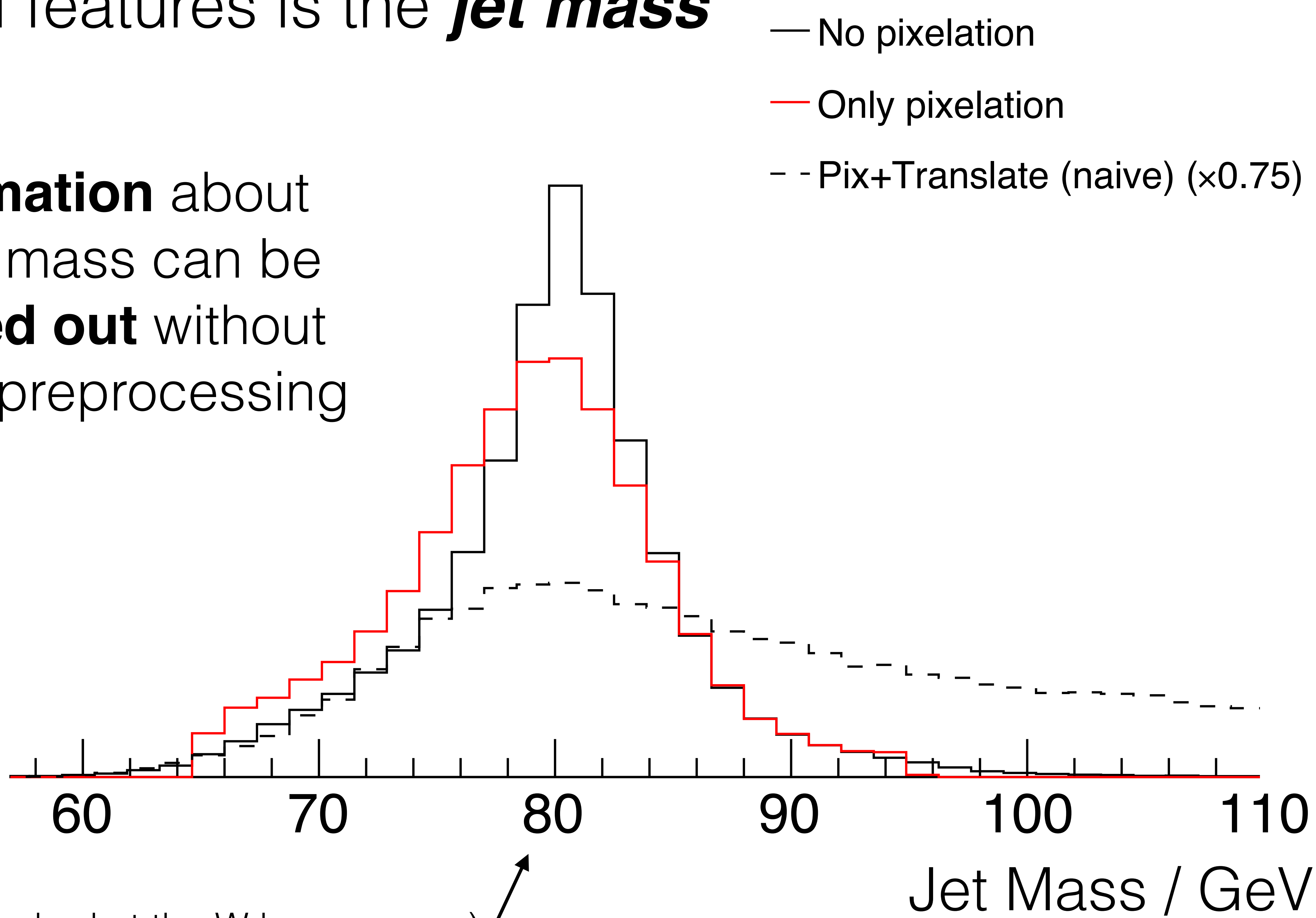


(distribution peaked at the W boson mass)

One of the most useful physics-inspired features is the ***jet mass***

$$m_{jet}^2 \sim \sum_{i<j} E_i E_j \theta_{ij}^2$$

Information about the jet mass can be **washed out** without care in preprocessing

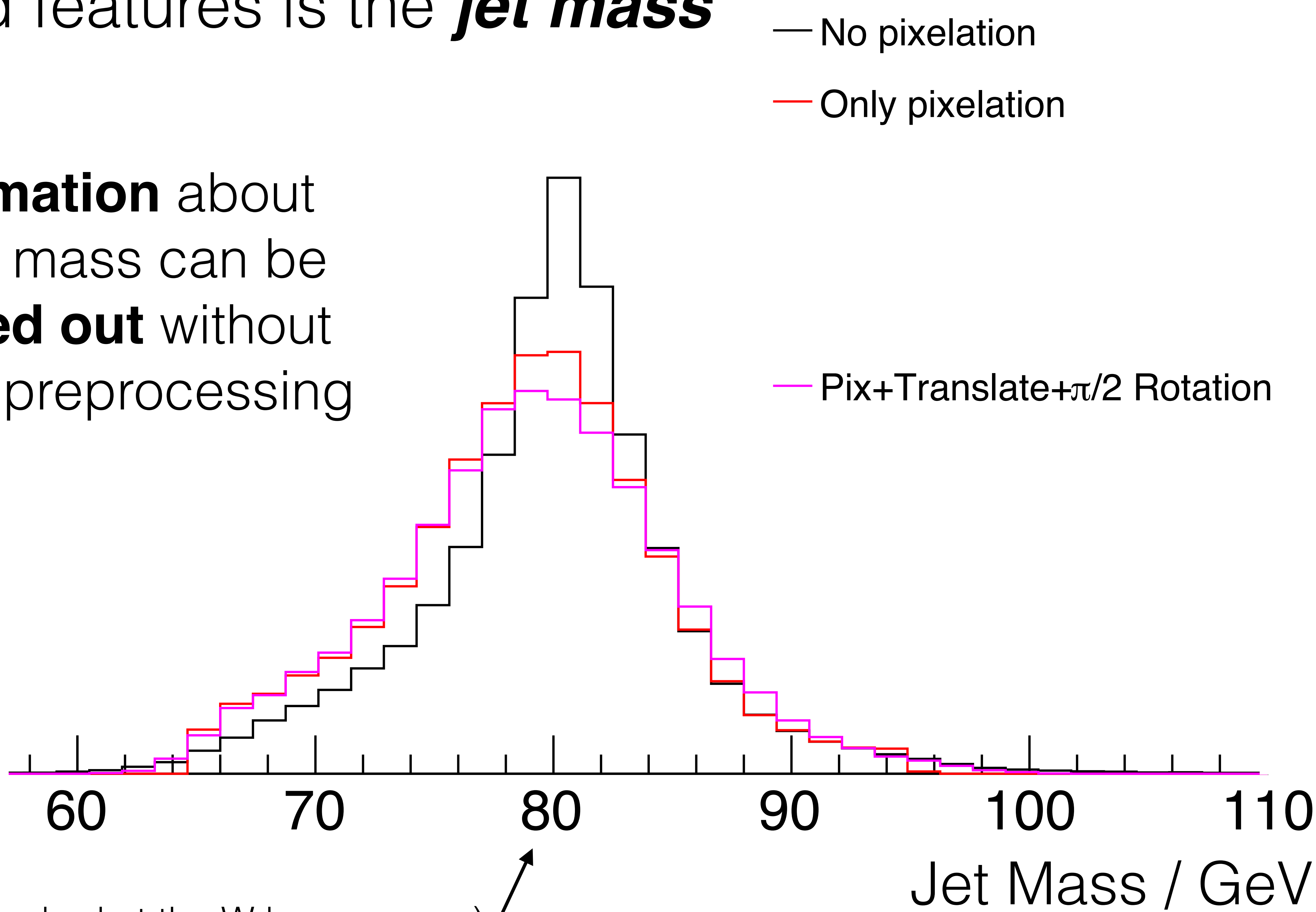


(distribution peaked at the W boson mass)

One of the most useful physics-inspired features is the ***jet mass***

$$m_{jet}^2 \sim \sum_{i<j} E_i E_j \theta_{ij}^2$$

Information about the jet mass can be **washed out** without care in preprocessing

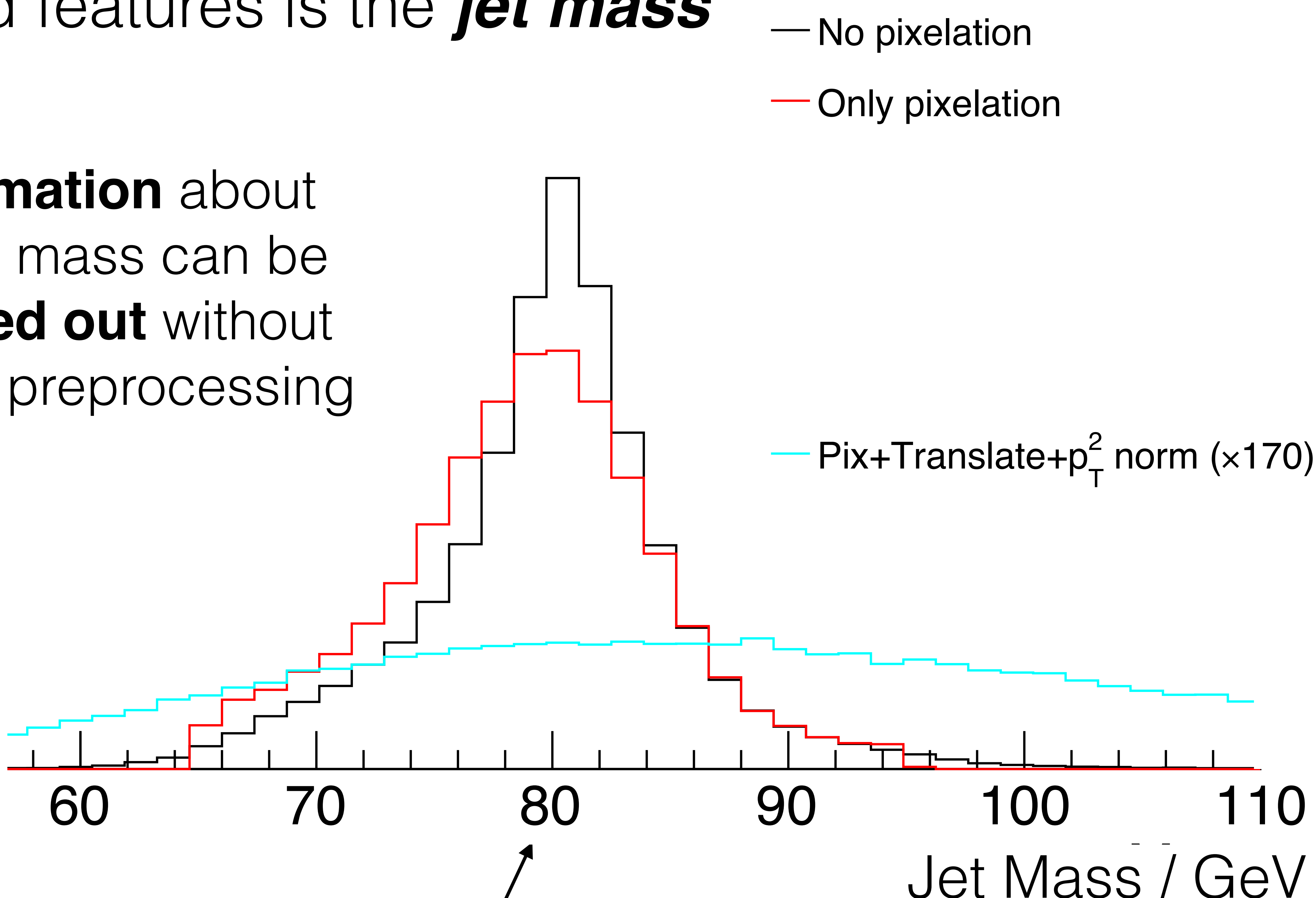


(distribution peaked at the W boson mass)

One of the most useful physics-inspired features is the ***jet mass***

$$m_{jet}^2 \sim \sum_{i<j} E_i E_j \theta_{ij}^2$$

Information about the jet mass can be **washed out** without care in preprocessing

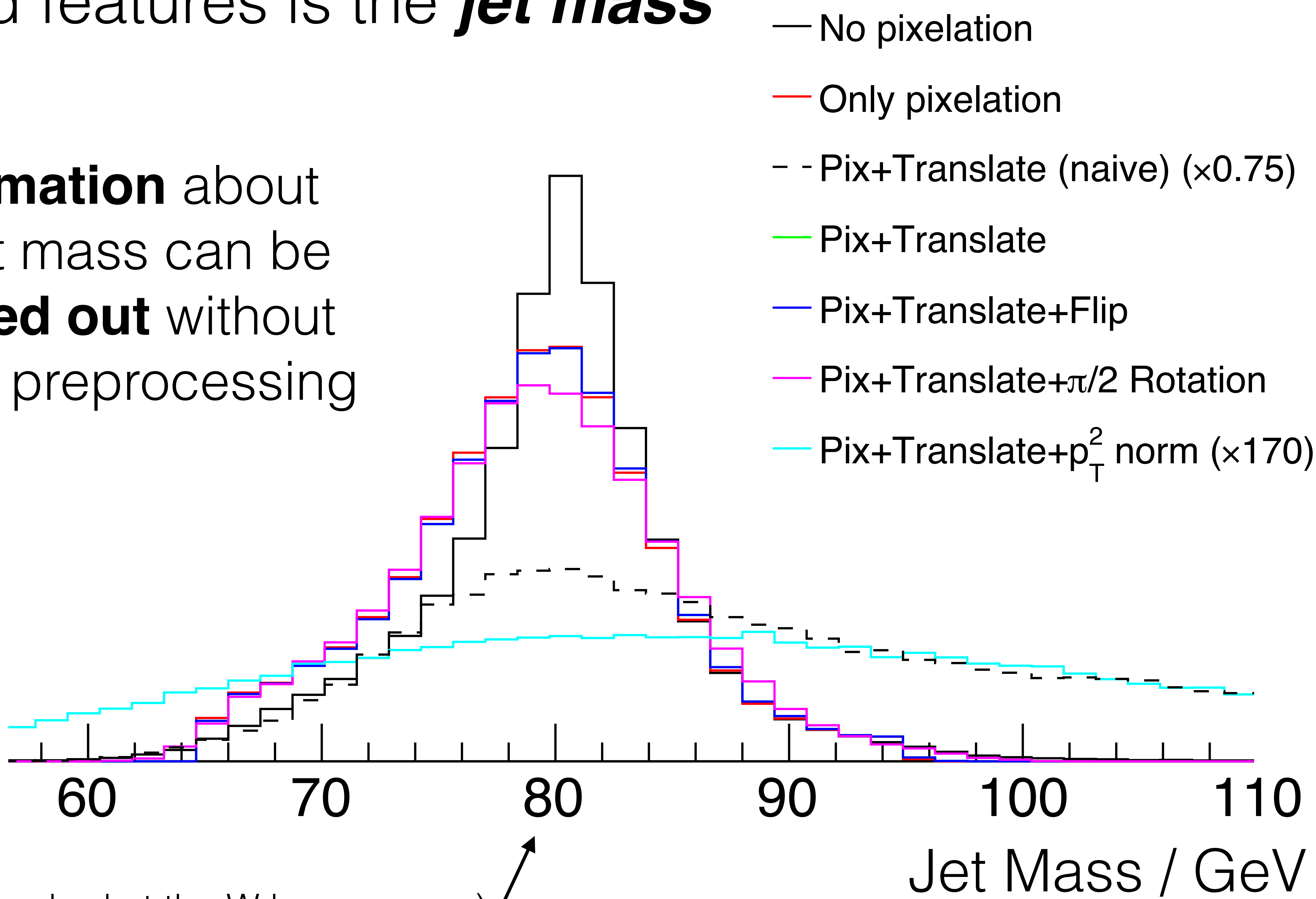


(distribution peaked at the W boson mass)

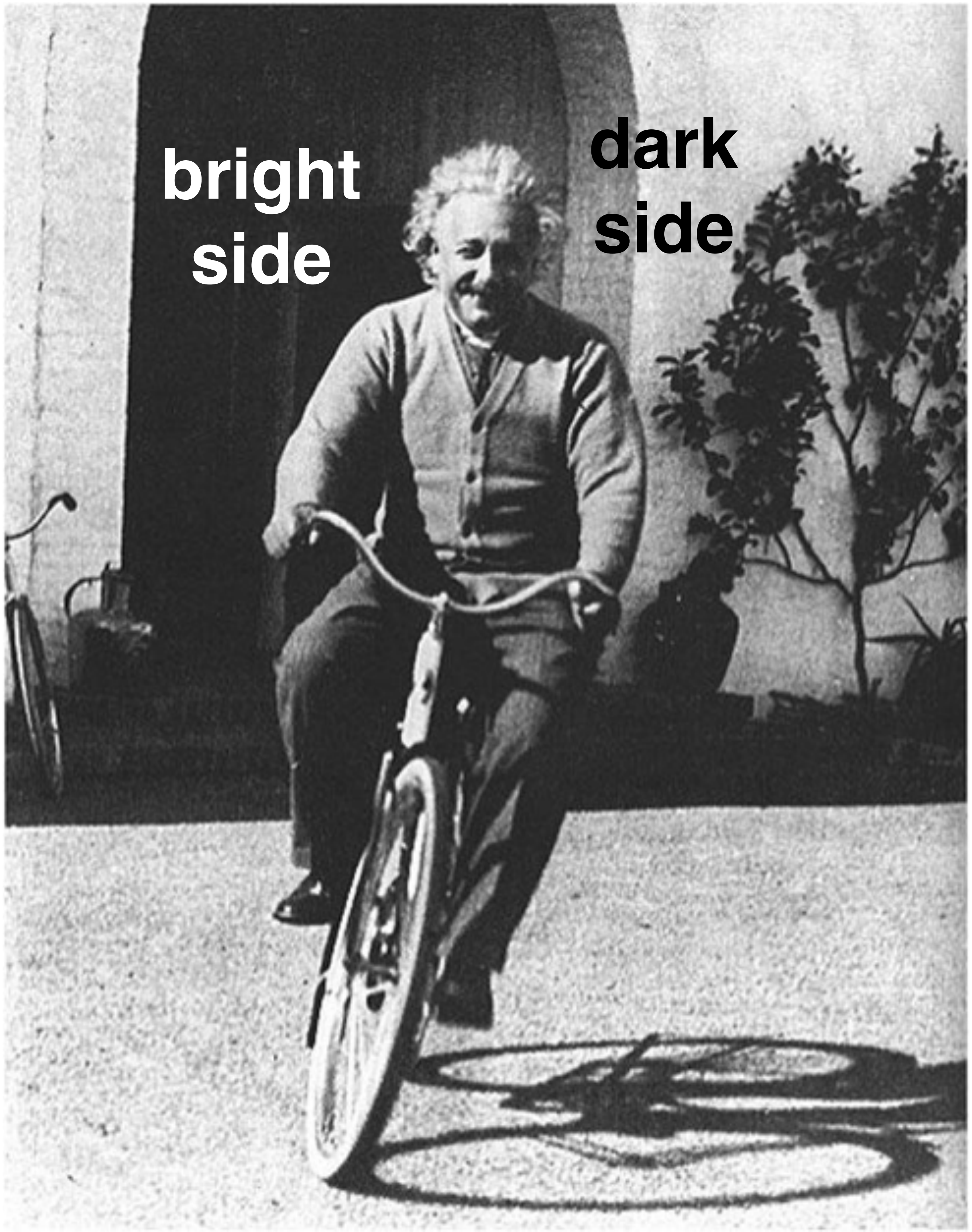
One of the most useful physics-inspired features is the ***jet mass***

$$\mathbf{m_{jet}^2 \sim \sum_{i<j} E_i E_j \theta_{ij}^2}$$

Information about the jet mass can be **washed out** without care in preprocessing



(distribution peaked at the W boson mass)



In both pictures, total intensity of Einstein's face is about the same.

However, his face's **image mass** is quite different!

Photos from: http://mentalfloss.com/article/49222/11-unserious-photos-albert-einstein

Intuition via analogy

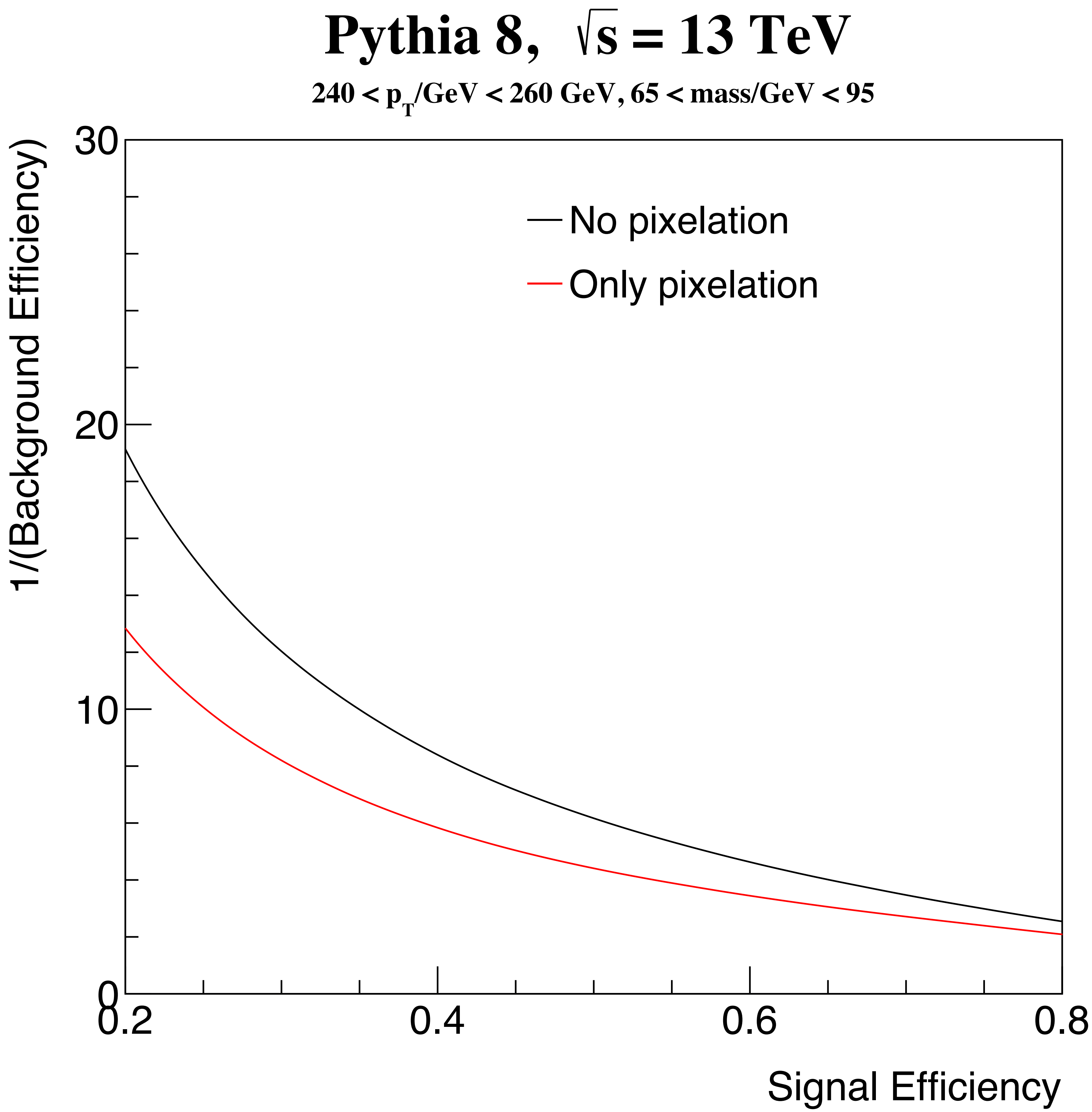
In both pictures, total intensity of Einstein's face is about the same.

In standard computer vision, you likely don't want to be sensitive to this! …not the case for jet images!

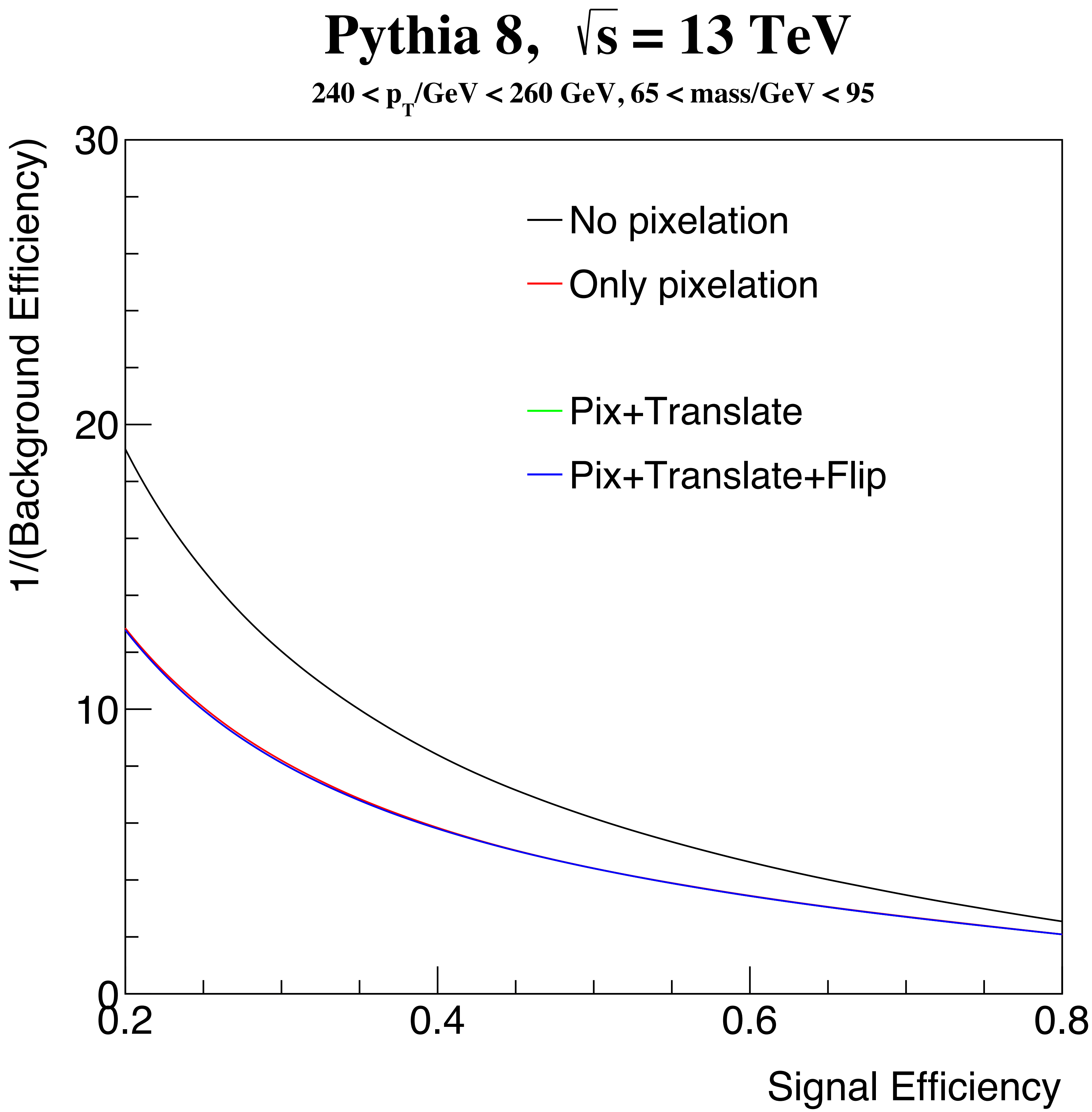
However, his face's **image mass** is quite different!

Photos from: http://mentalfloss.com/article/49222/11-unserious-photos-albert-einstein

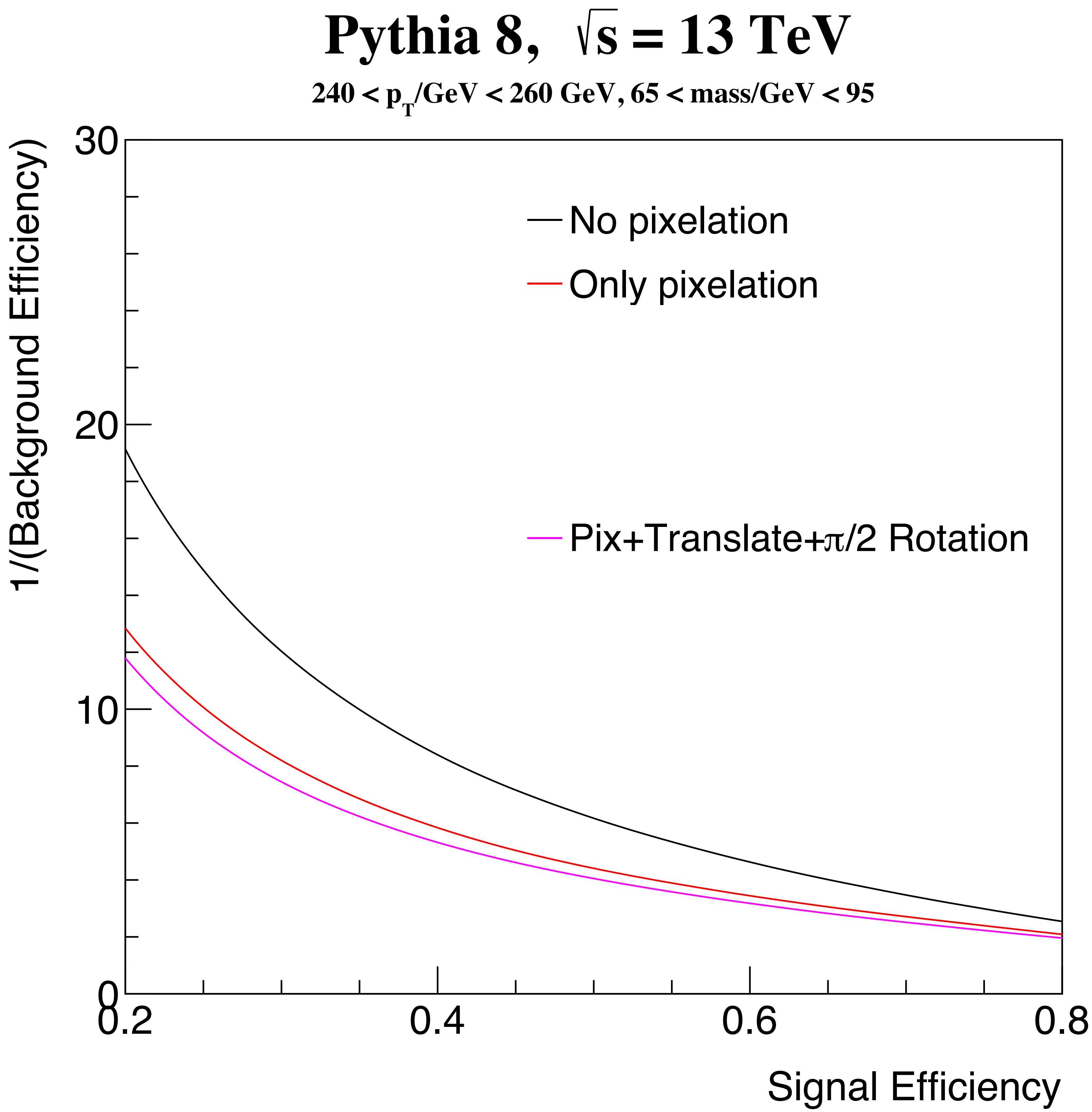
Pre-processing and the symmetries of space-time



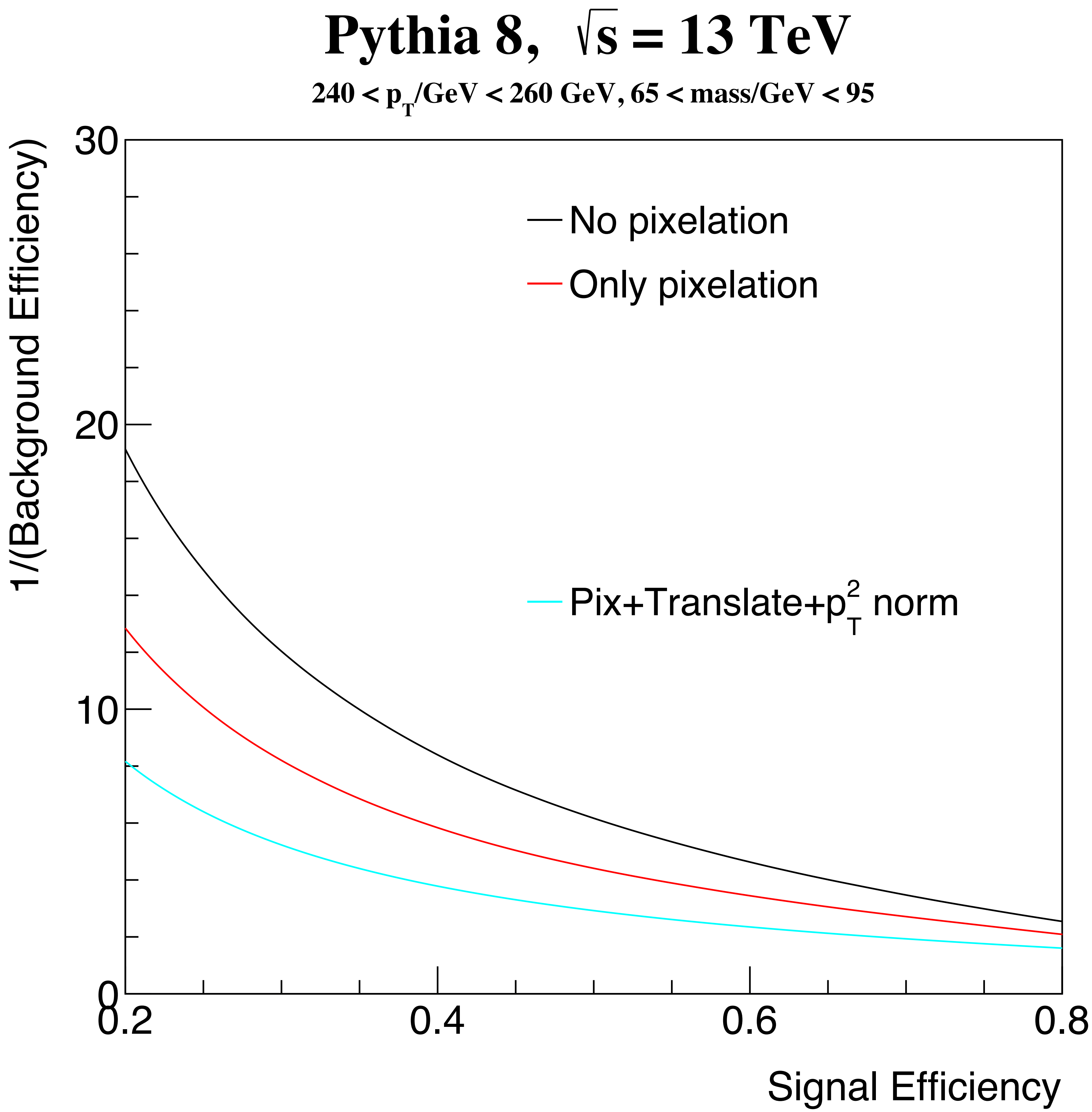
Pre-processing and the symmetries of space-time

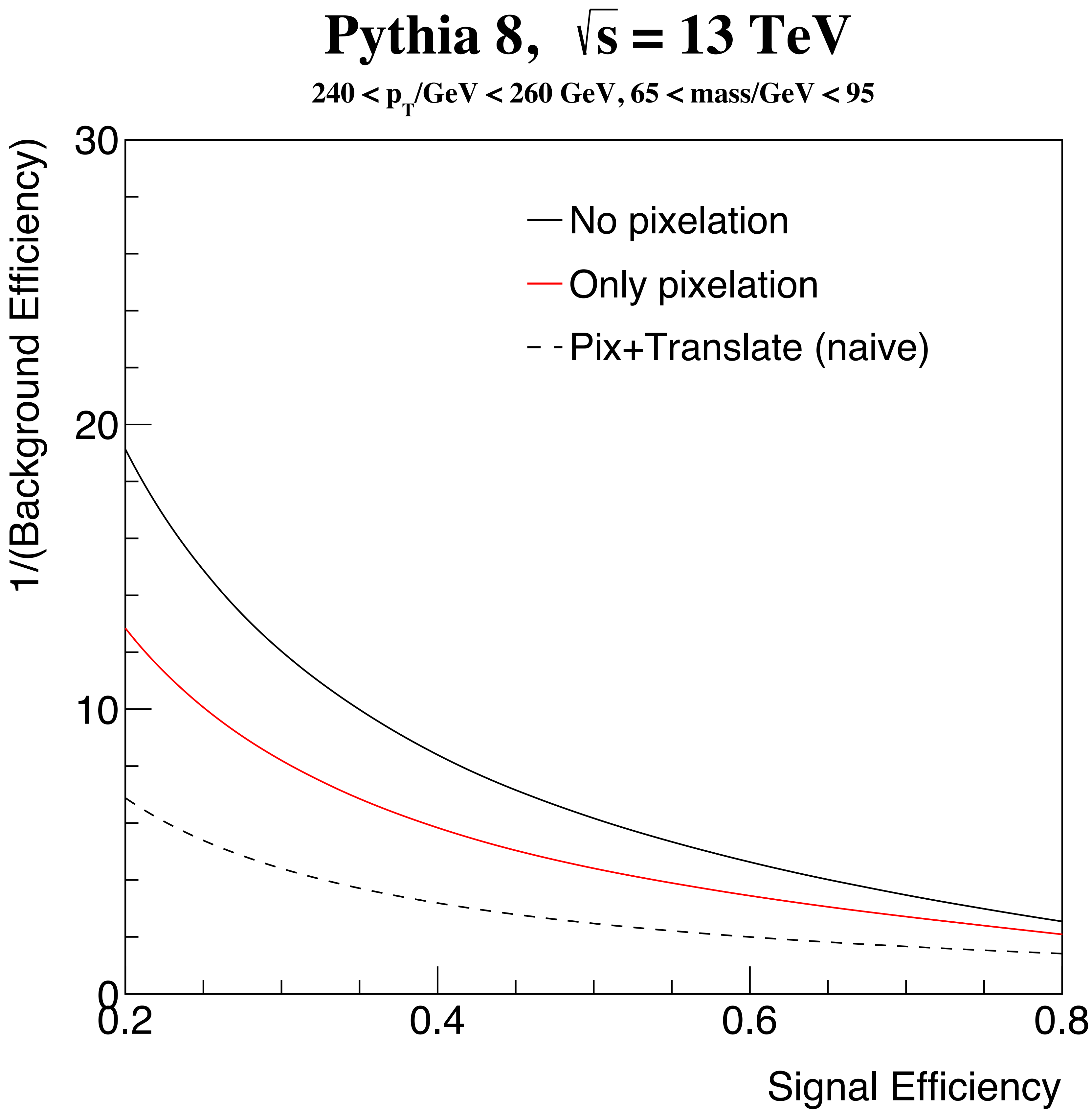


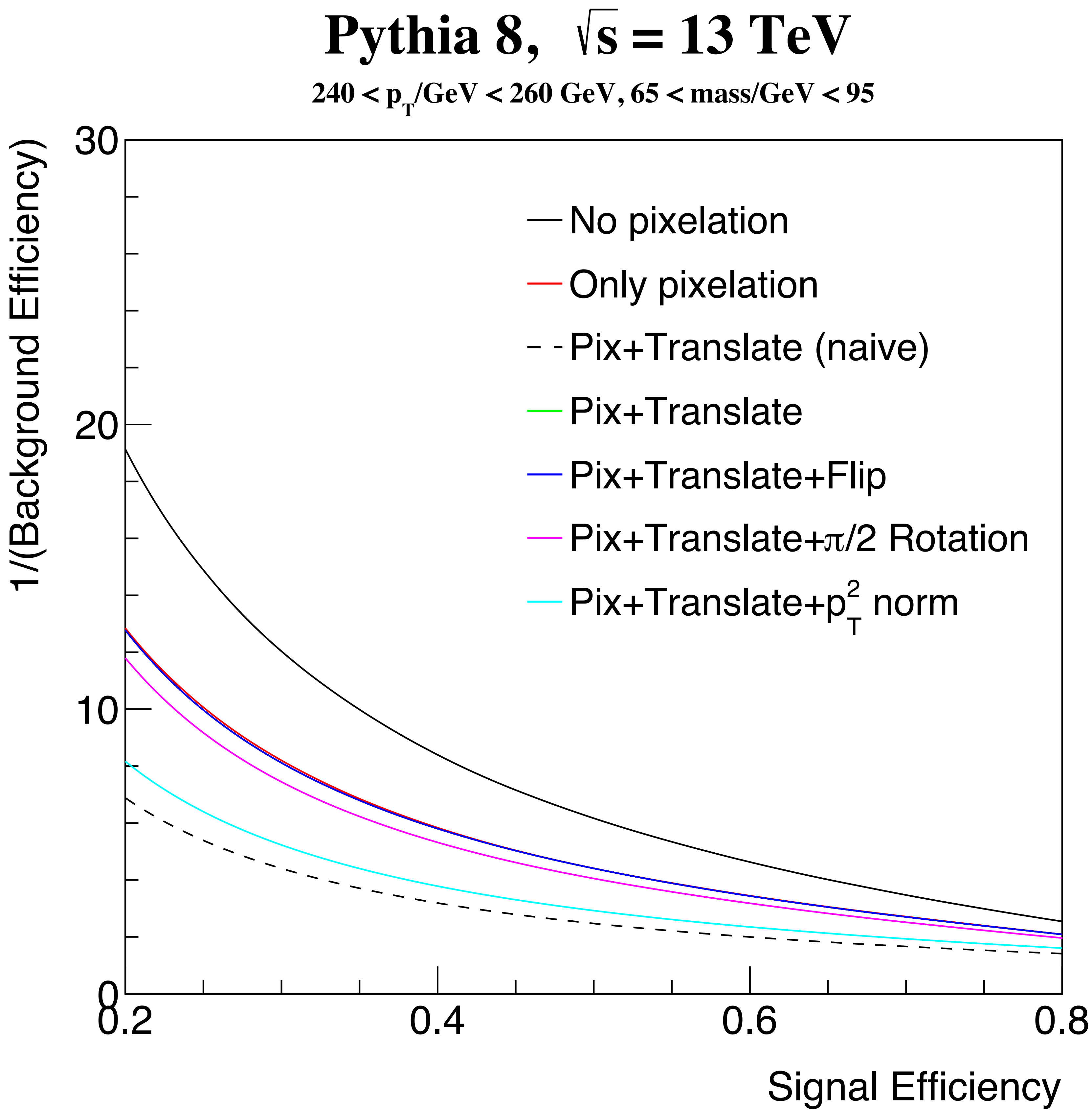
Pre-processing and the symmetries of space-time



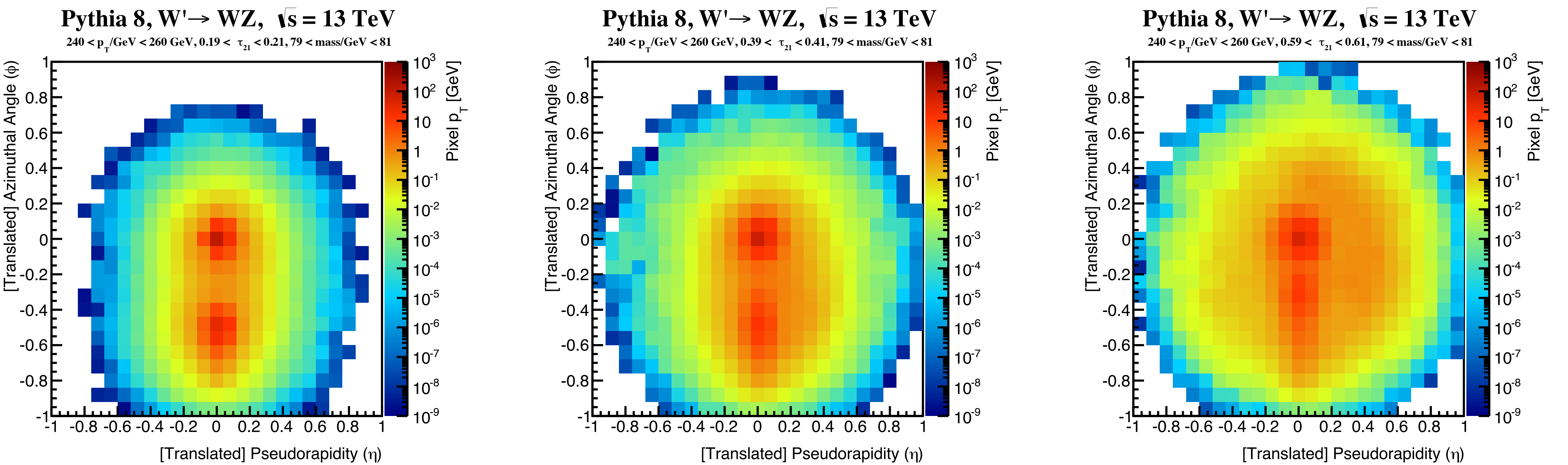
Pre-processing and the symmetries of space-time




Pythia 8, $\sqrt{s}$ = 13 TeV
240 < p_T/GeV < 260 GeV, 65 < mass/GeV < 95
1/(Background Efficiency)
30
20
10
0
0.2
0.4
0.6
0.8
Signal Efficiency
No pixelation
Only pixelation
Pix+Translate (naive)

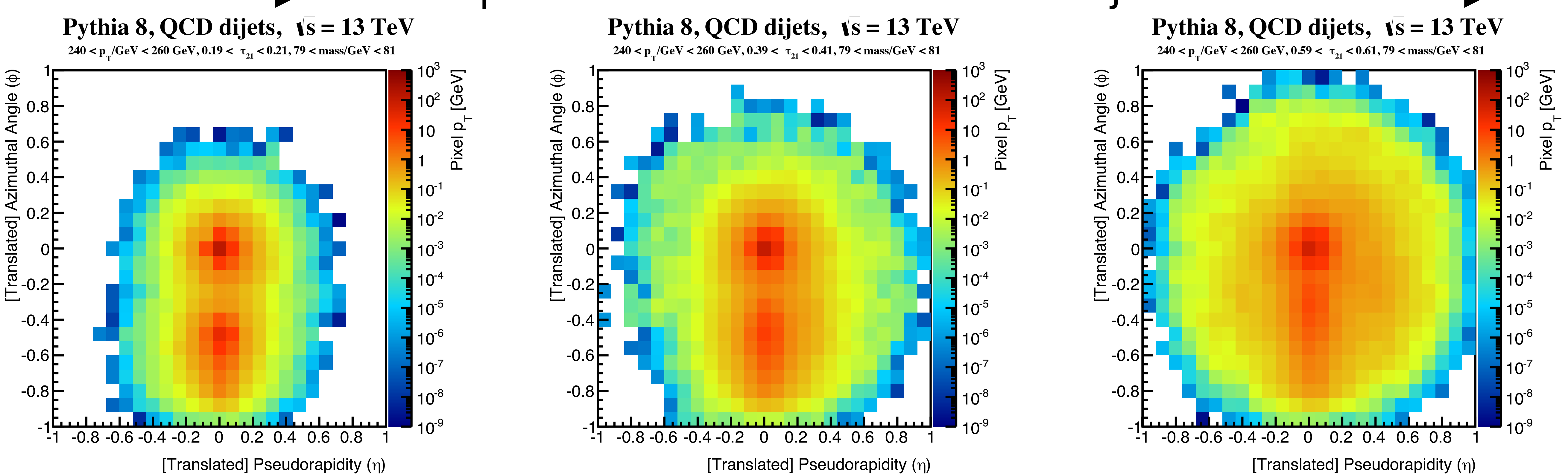

Pythia 8, $\sqrt{s}$ = 13 TeV
240 < p_T/GeV < 260 GeV, 65 < mass/GeV < 95
1/(Background Efficiency)
30
20
10
0
0.2
0.4
0.6
0.8
Signal Efficiency
No pixelation
Only pixelation
Pix+Translate (naive)
Pix+Translate
Pix+Translate+Flip
Pix+Translate+π/2 Rotation
Pix+Translate+p_T^2 norm

Where is the discrimination?

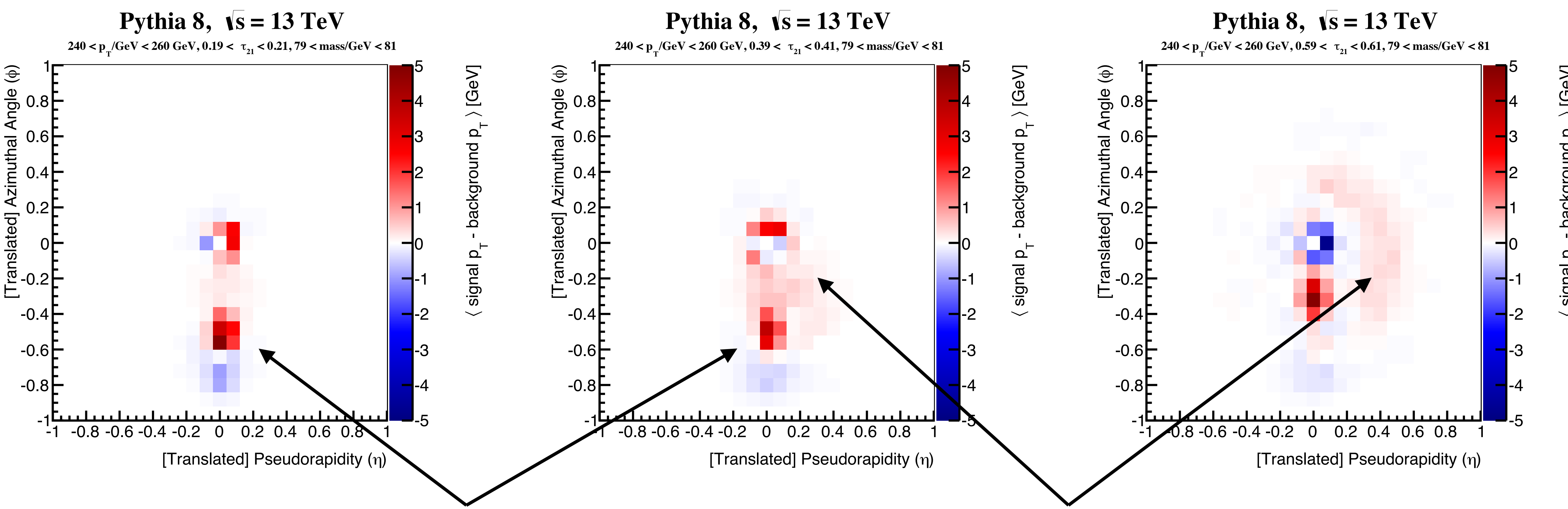


The jet image paradigm allows us to **visualize** this information!

less pronounced second subjet



You can **see** the physics!



The distance
between subjets
is slightly different

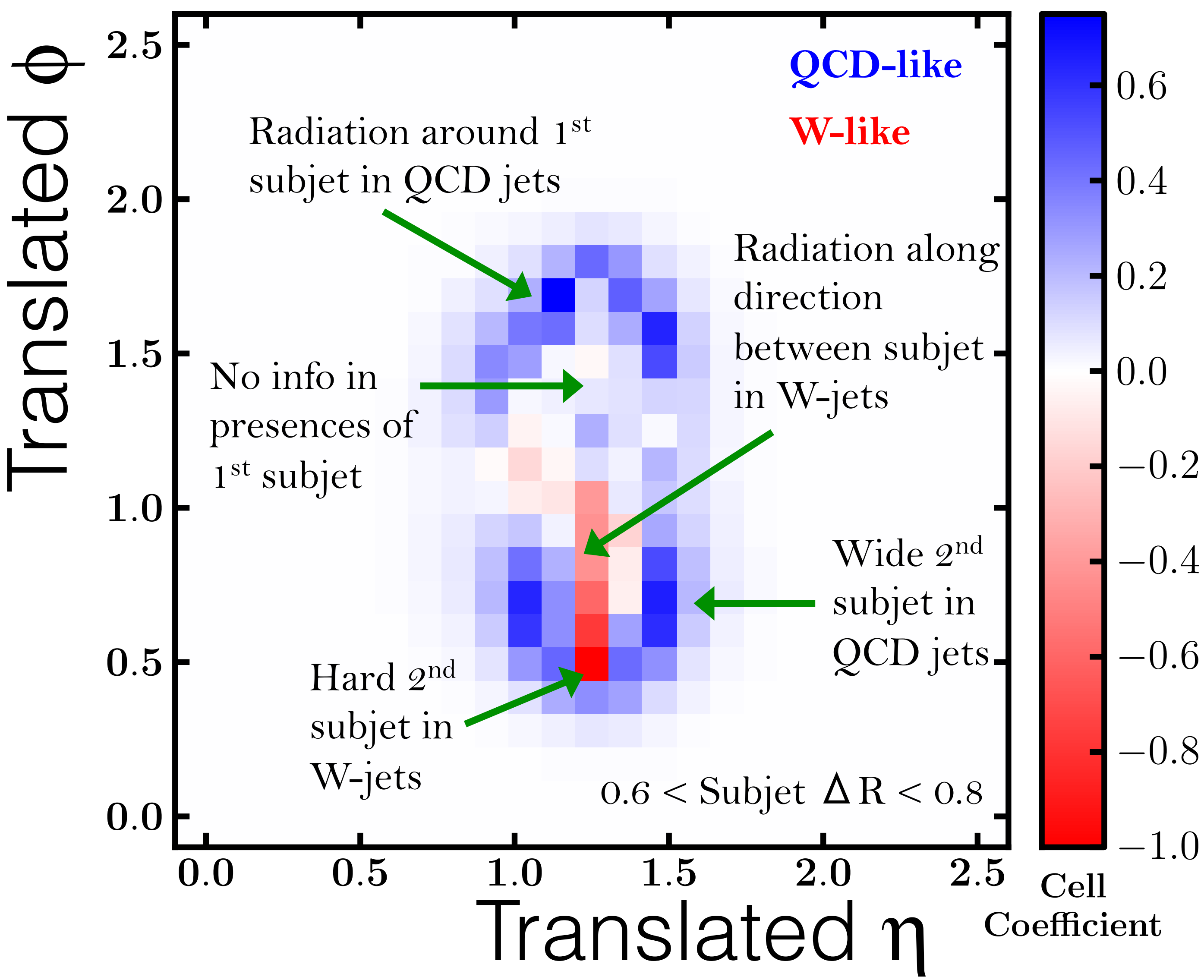
gluon jet background
is a color octet,
diffuse radiation

less pronounced second subjet

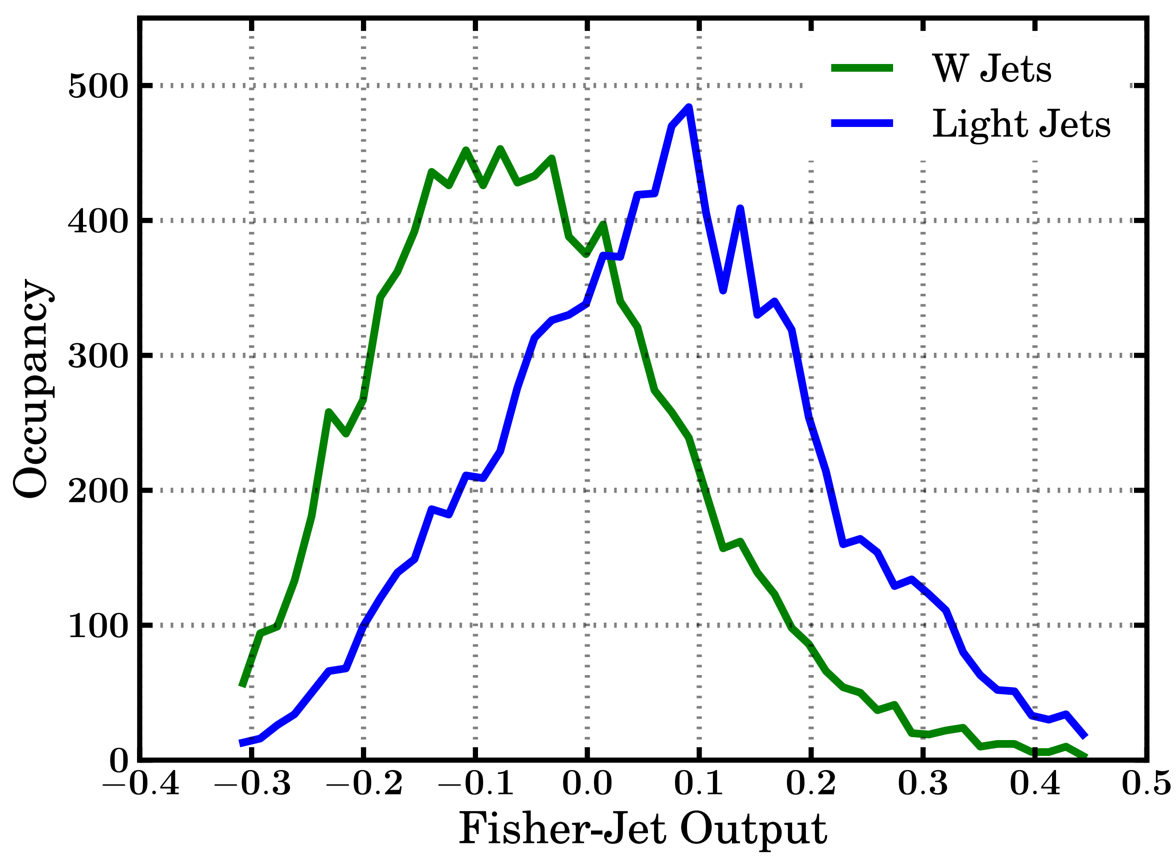
Analogous to facial recognition with Fisher Faces, construct a **Fisher Jet:**

Direction in the n x n image space that maximizes the between class variance over the within class variance

Directly interpretable!



The discriminant is the projection of any image onto the Fisher Jet

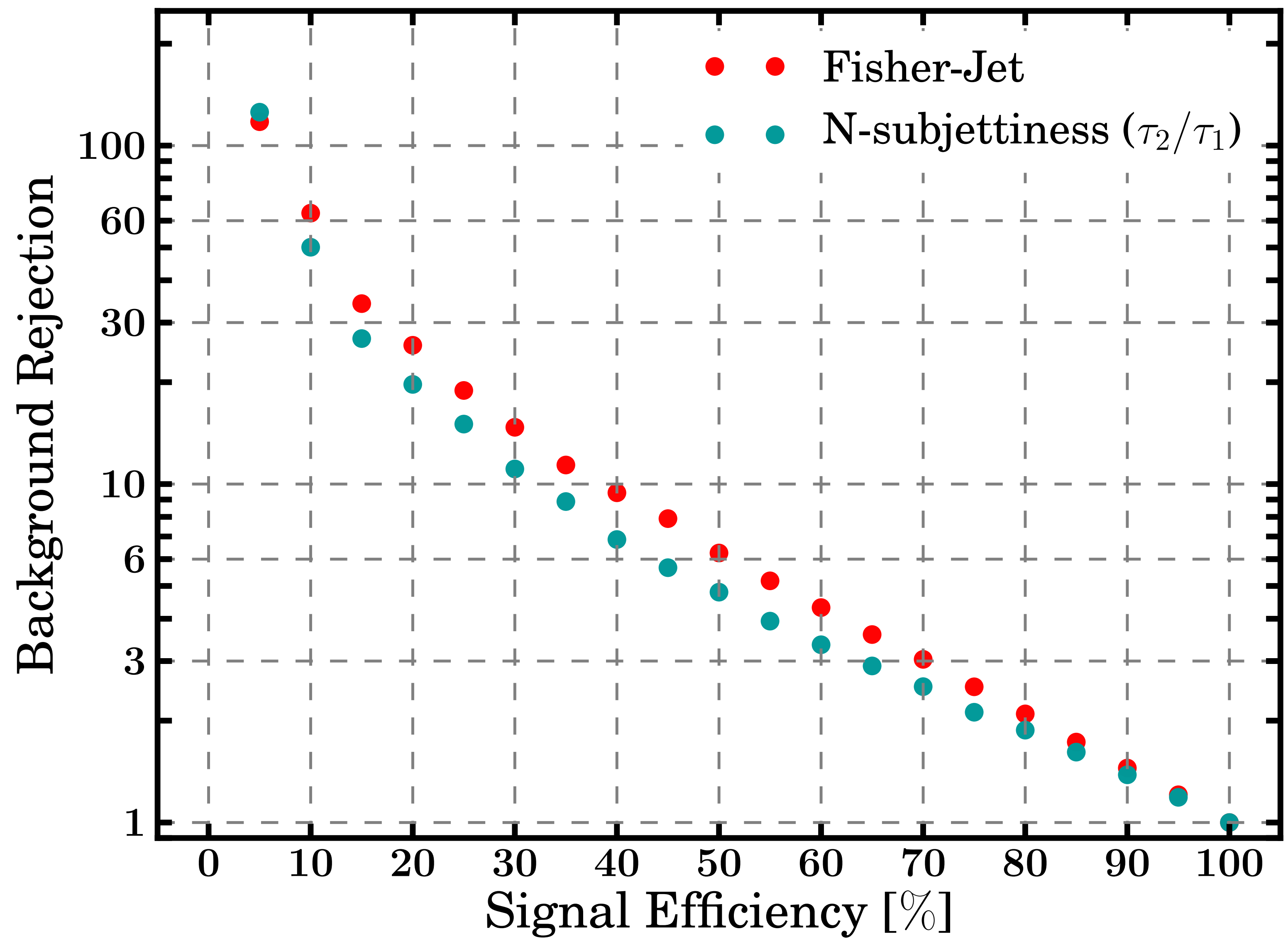


(b) Fisher-Jet Discriminant Output

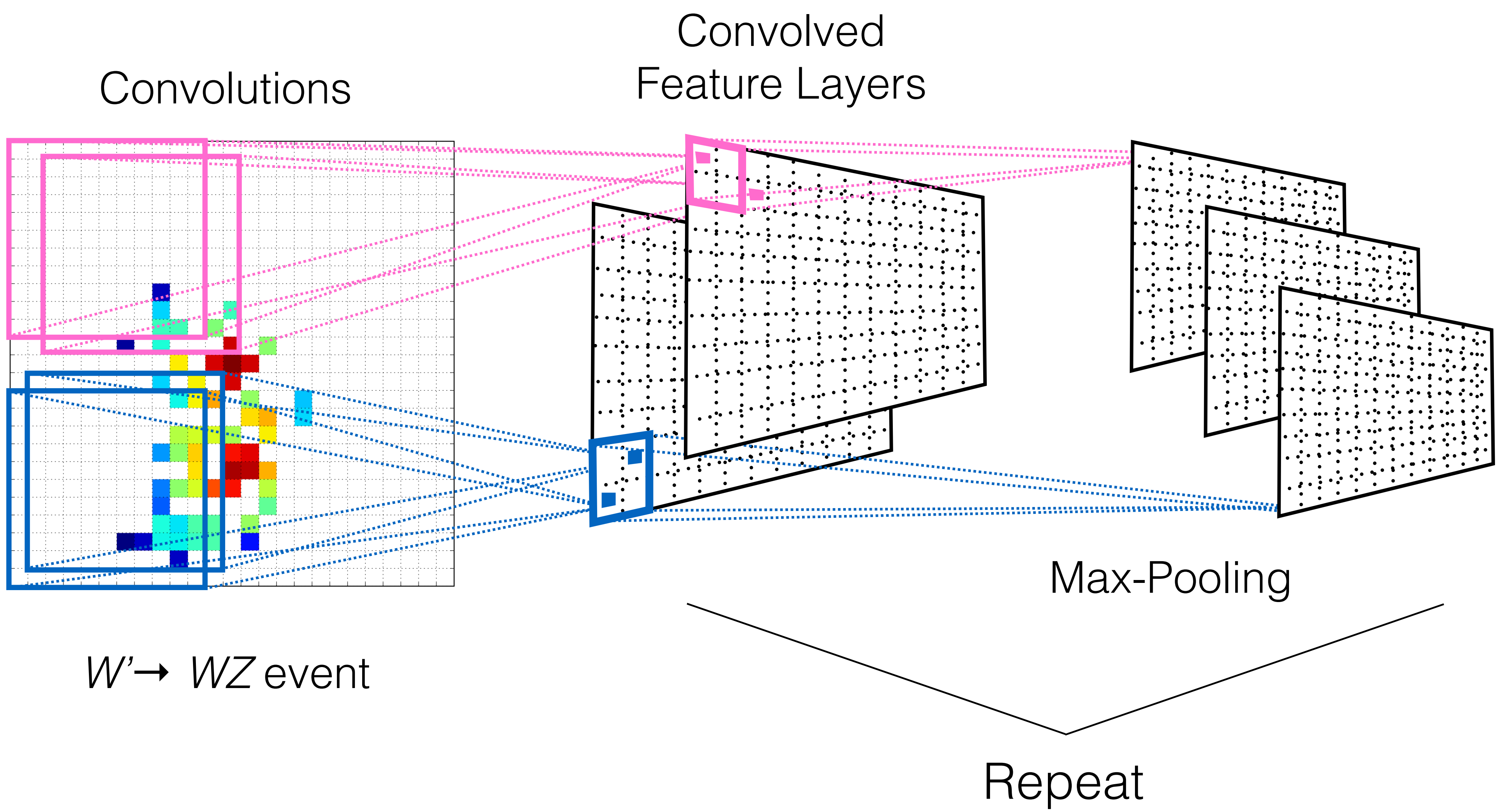
Slightly worse performance as N-subjettines out-of-the-box;

Add in a coarse ΔR binning to surpass τ_{21}

Image analogy: eyes get further apart the farther away you are!



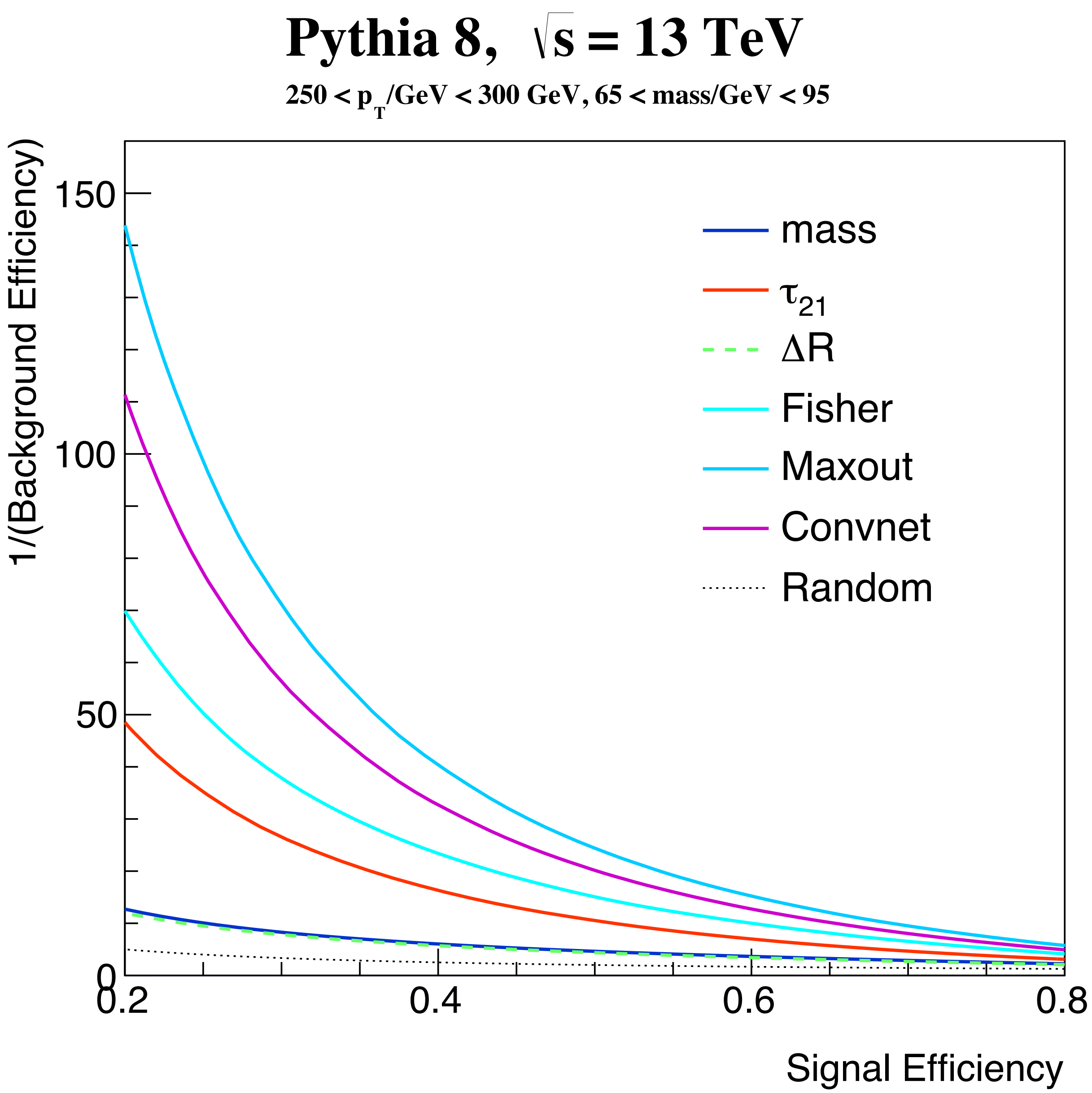
Even more non-linearity: Going Deep

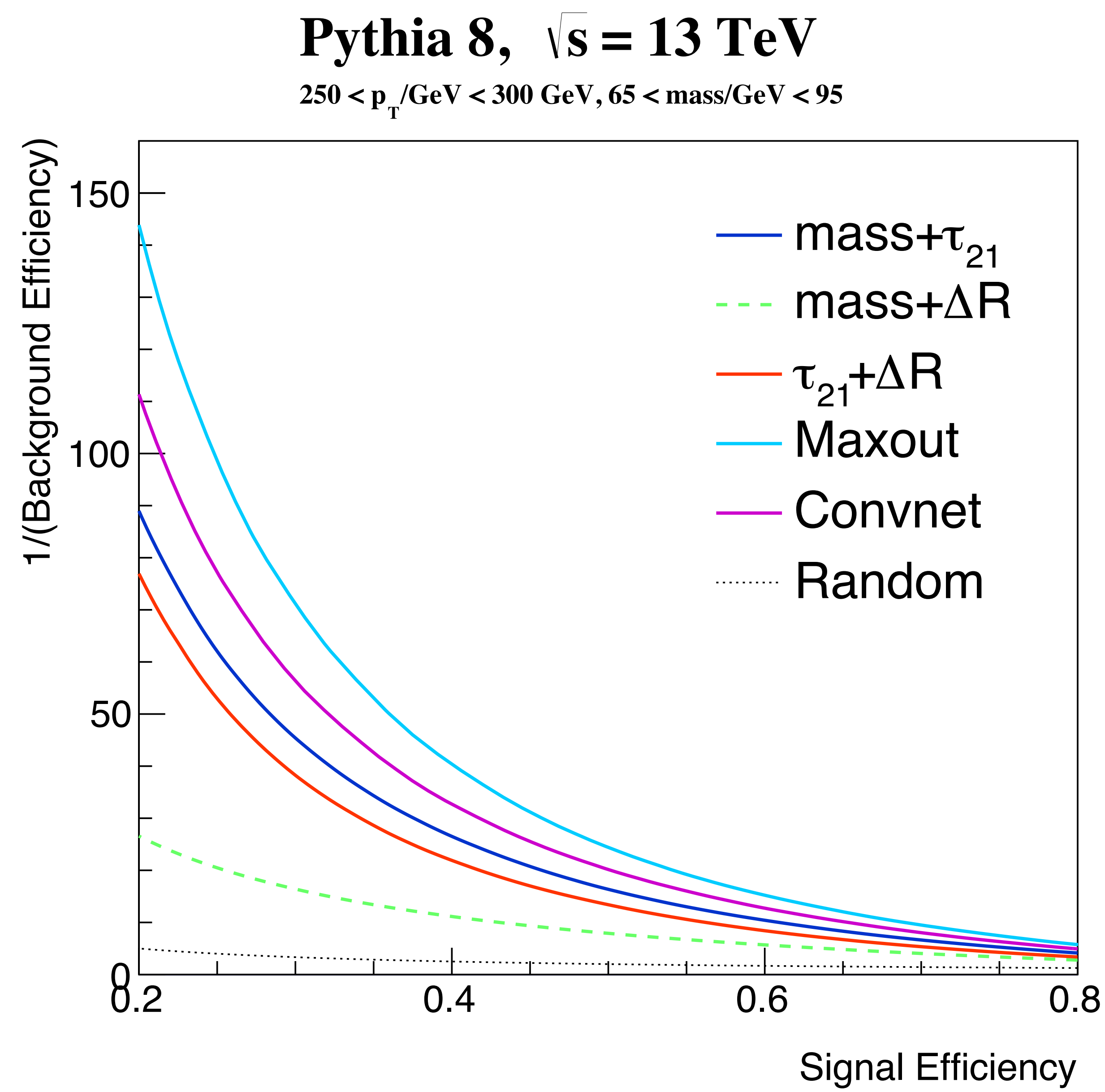


Apply deep learning techniques on jet images! [3]

convolutional nets are a standard image processing technique; also consider maxout

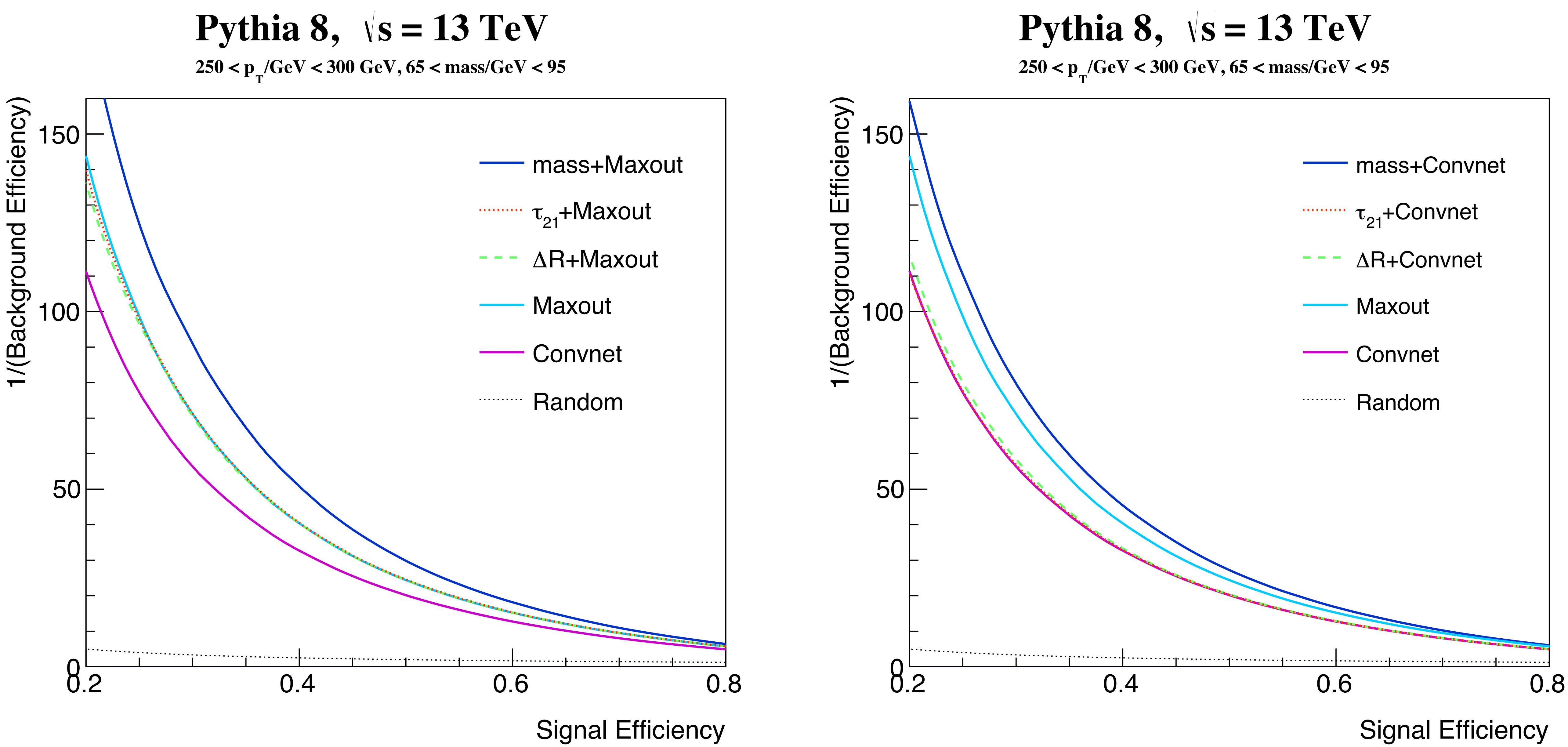
Performance





Out-performs standard and well-performing features.
Maxout out-performs Convnet (more on this shortly)

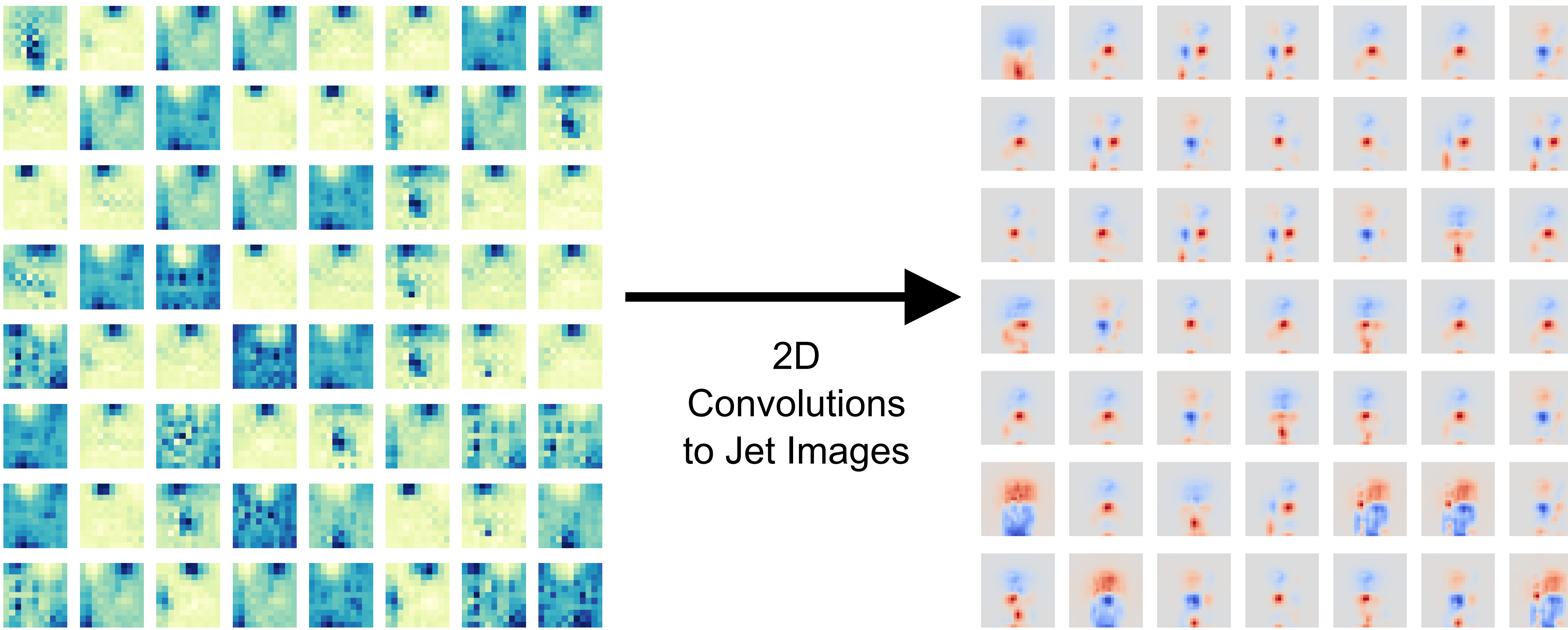
Performance and a first look at what is learned



A first indication that the networks are efficiently learning angular information, but not all there is about the jet mass.

(N.B. only 3 coarse bins of mass are needed to achieve the boost!)

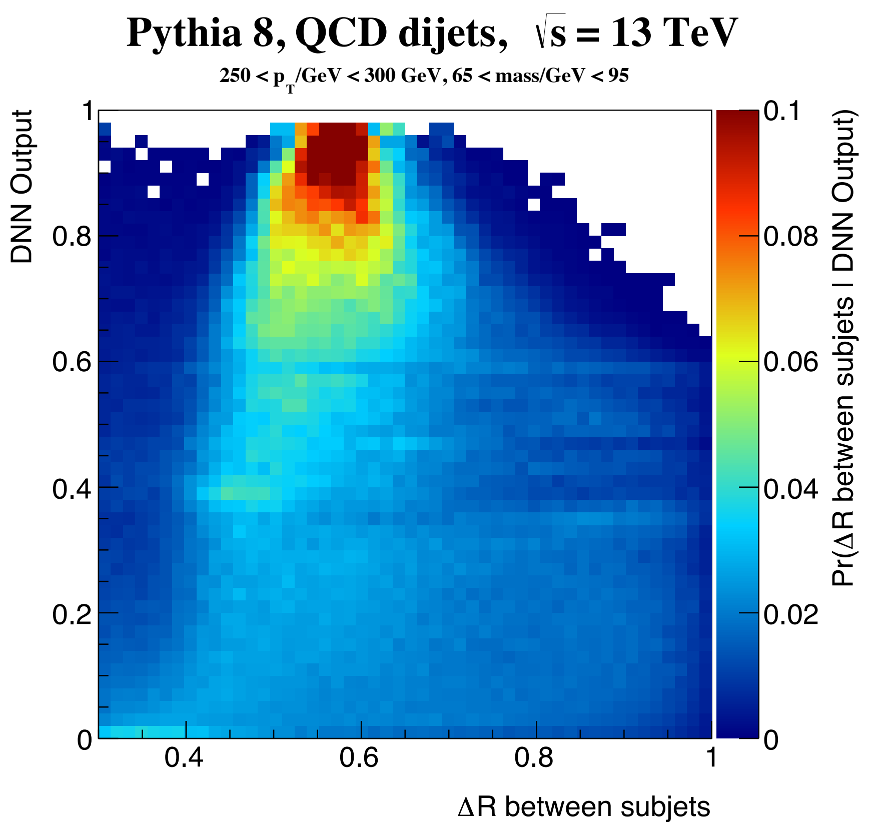
Advantage of CNN is that we can visualize the filters

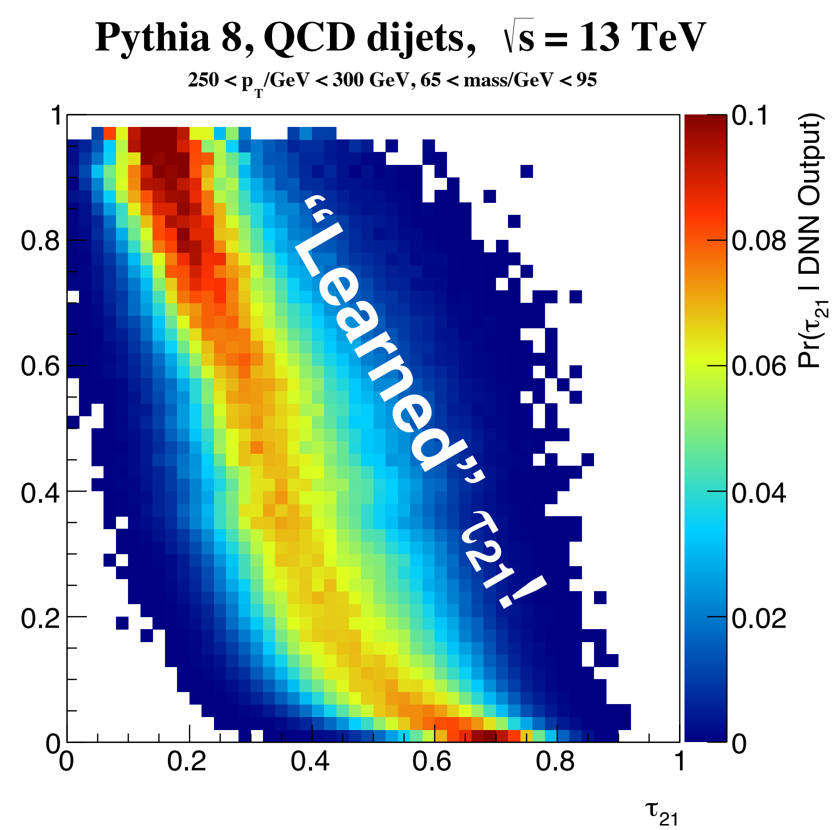


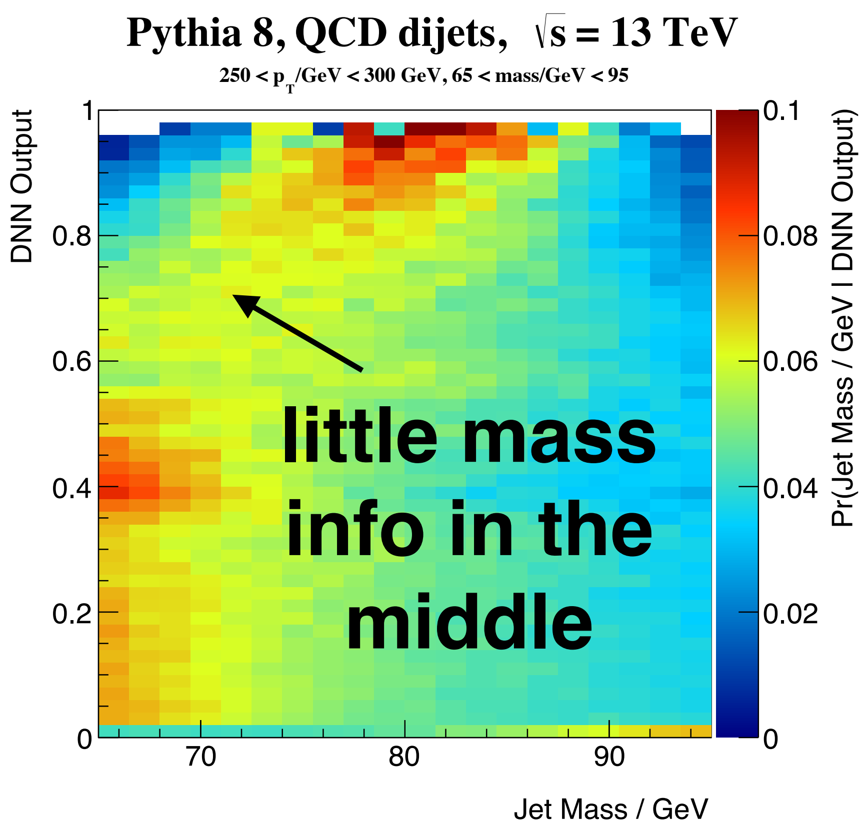
Data very sparse; convolution paradigm does not work as intended (need large filters).

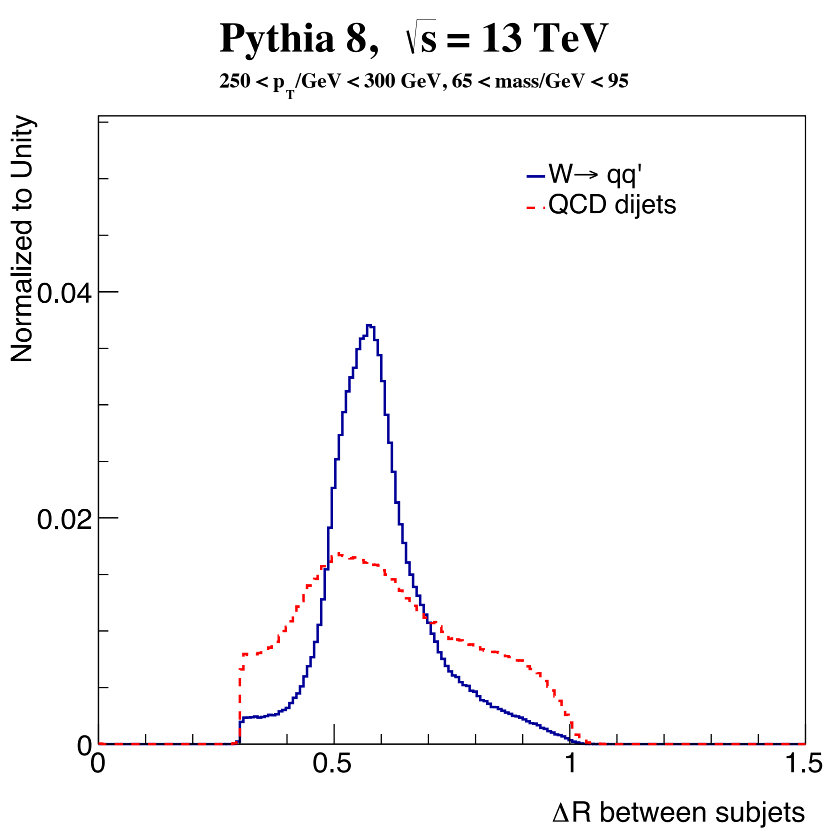
However, we can apply the new technique for visualization learned information by convolving the filters with the images

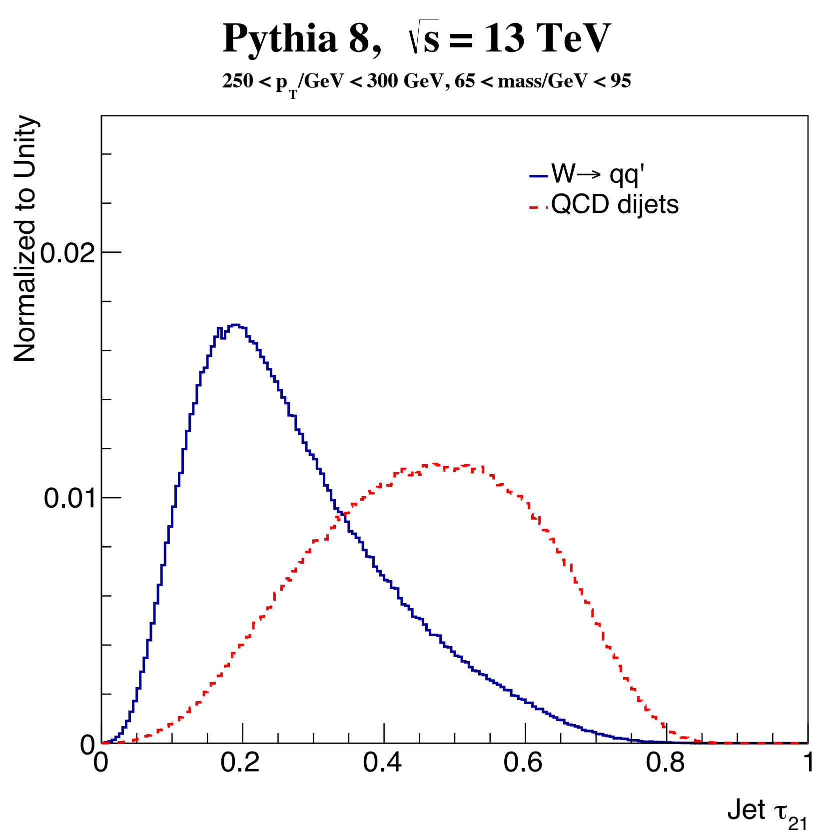
Learning about learning

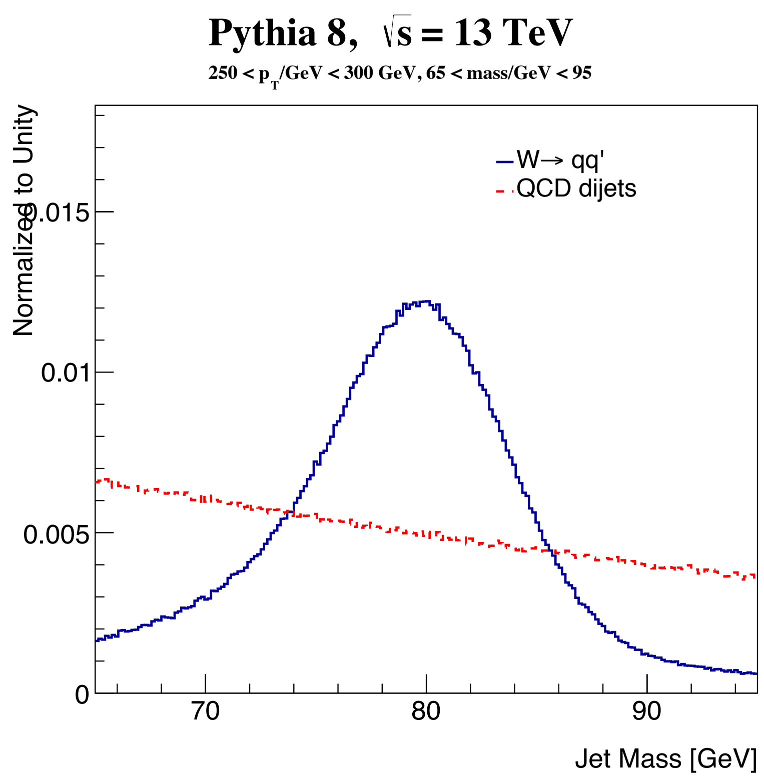




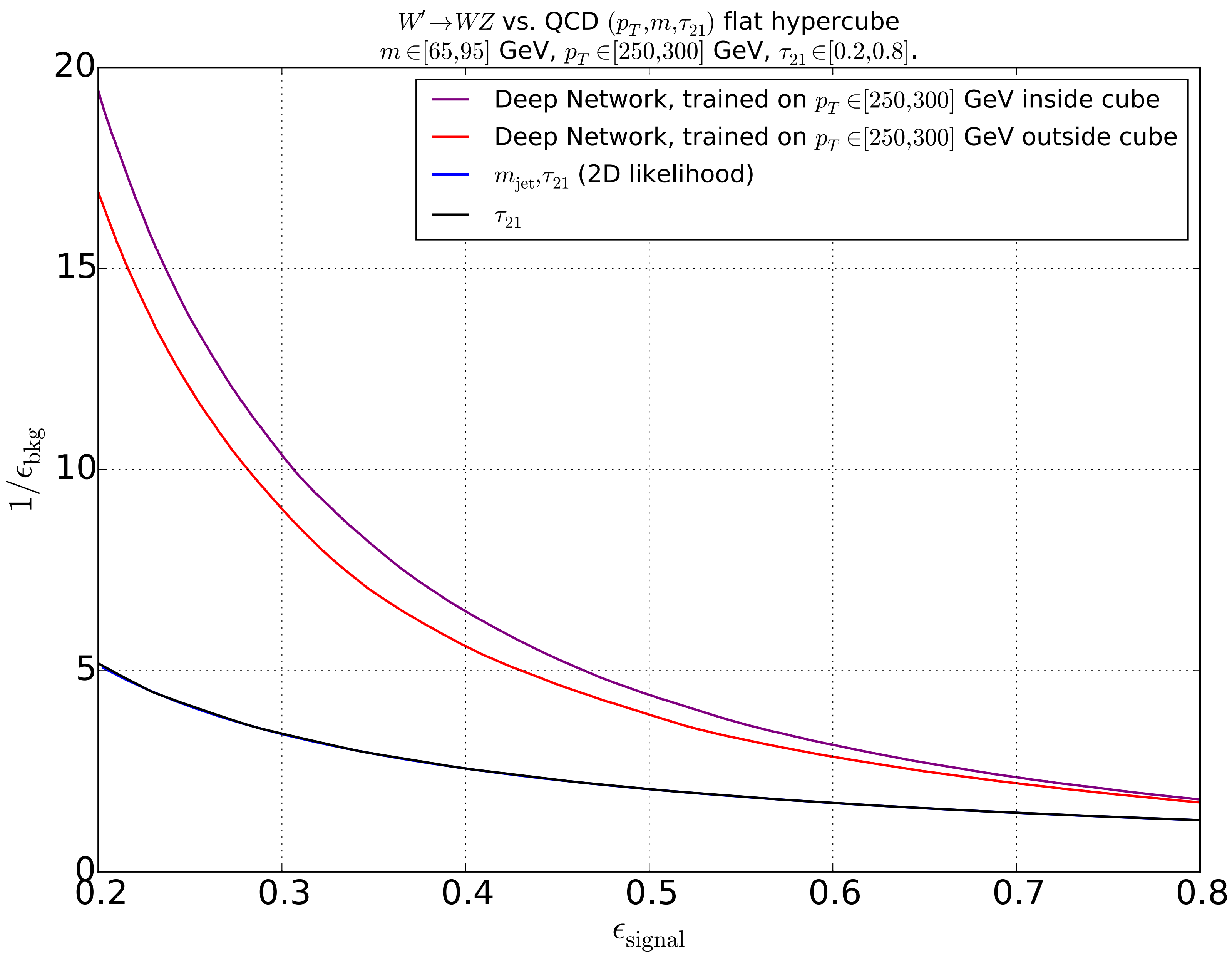






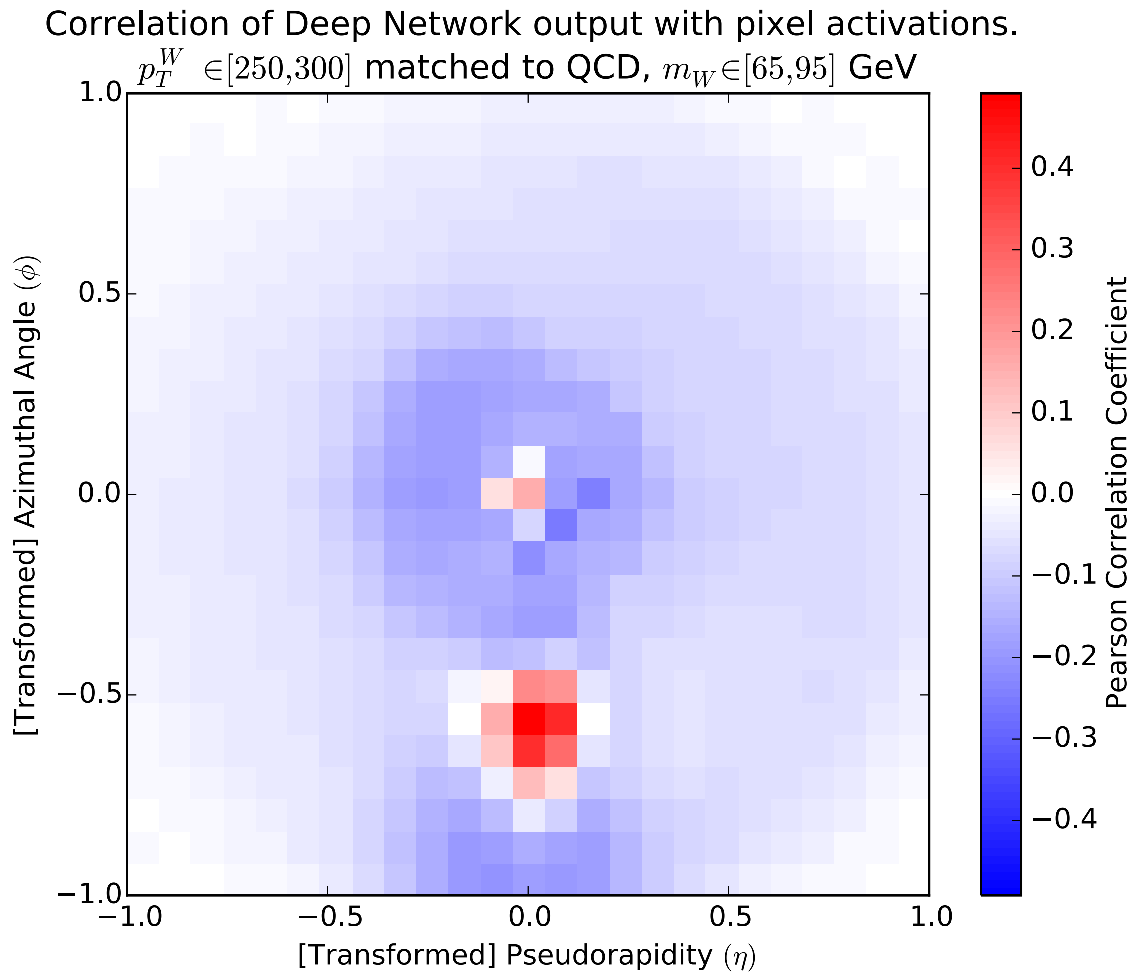


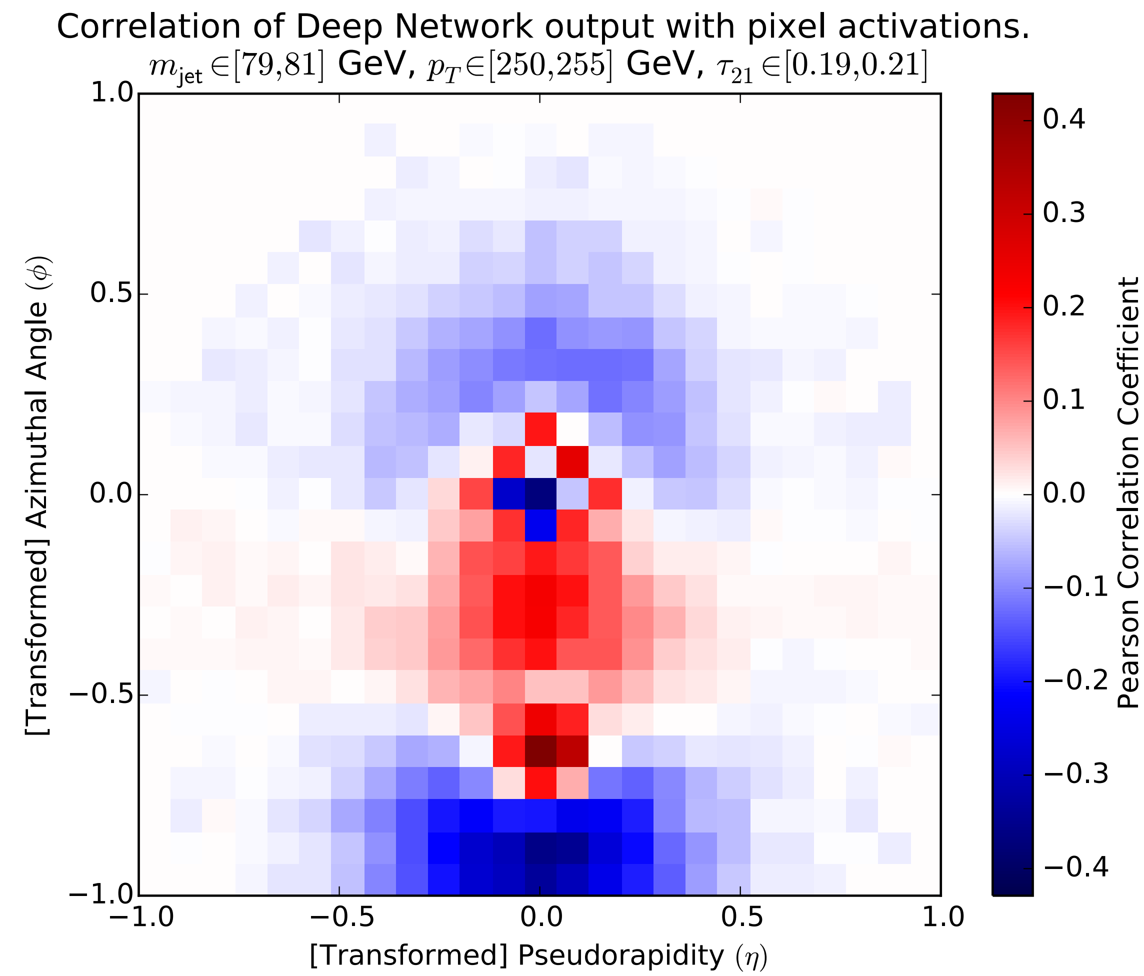
Learning about learning



There is clearly something learned beyond (mass and) τ_{21}. There is certainly physics to learn: colorflow, etc.

Learning about learning



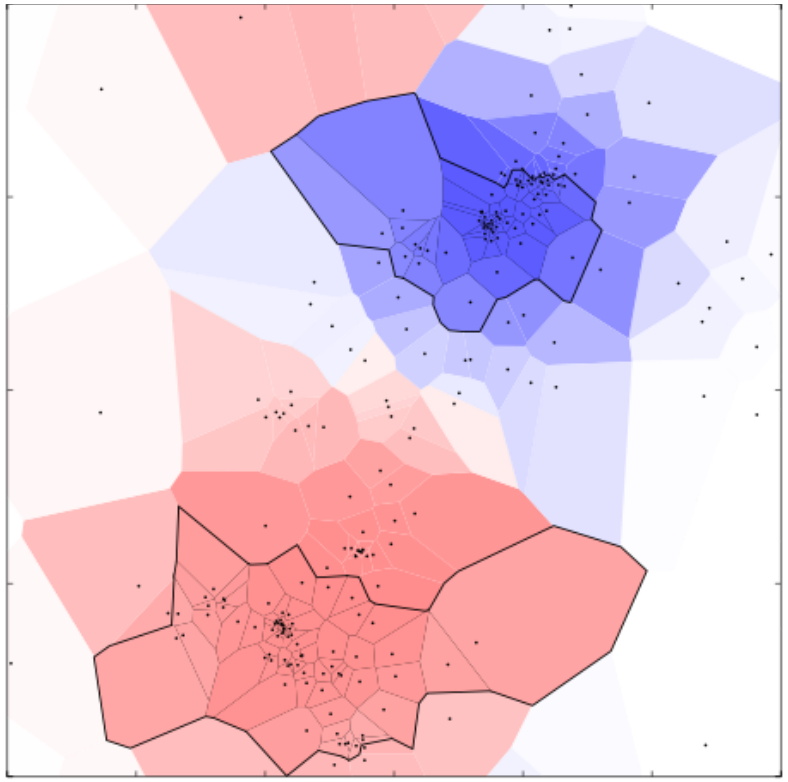


Pixel-by-pixel correlation between the network output and pixel intensity: linear in z-axis but non-linear spatial information. There is clearly some information about the colorflow embedded in the neural network!

Beyond optimizing discrimination, can we **learn** what ML algorithms **learn** about physics?

-What can be gained by looking at data in new ways?

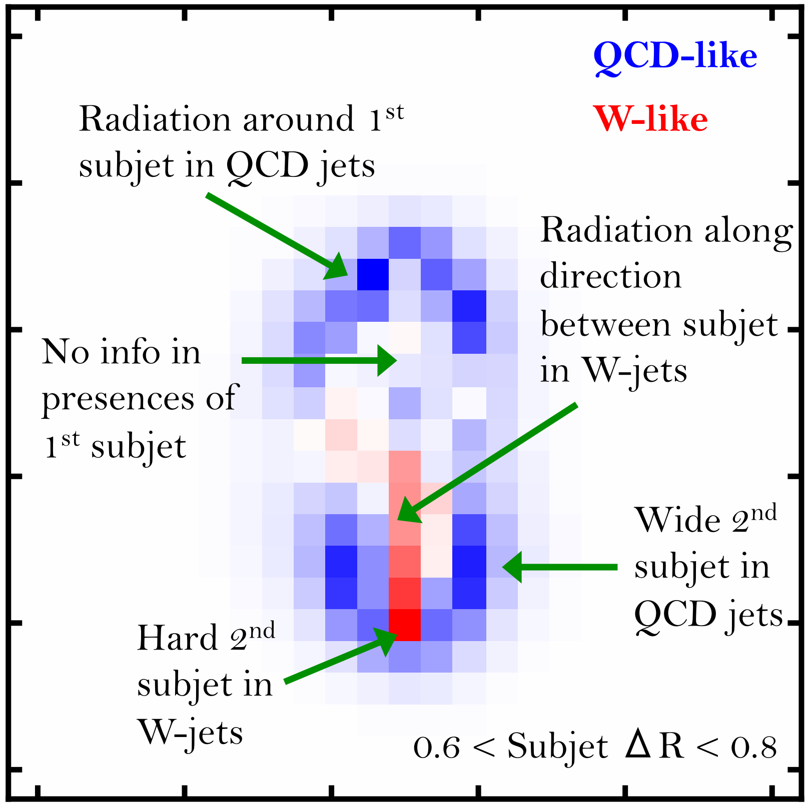
-How to adapt ML algorithms to physics?



We have shown two examples:

Blurring jet clustering algorithms with **Fuzzy Jets**

-IRC safe likelihood-based approach to jet clustering



Physics meets Computer Vision: **Jet Images**

-Powerful discrimination

-Intuitive visualization to understand what physics has been learned

Our goal: Continue to use powerful ML techniques to learn **about** physics at the LHC

For more information **http://stanford.edu/group/hepml/**

References:

[1] L. Mackey, B. Nachman, A. Schwartzman, and C. Stansbury, *Fuzzy Jets*

Submitted to JHEP. *Preprint: 1509.02216 [hep-ph].*

[2] J. Cogan, M. Kagan, E. Strauss, A. Schwartzman, *Jet-Images: Computer Vision Inspired Techniques for Jet Tagging*

JHEP 02 (2015) 118. *Preprint: 1407.5675 [hep-ph].*

[3] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, and A. Schwartzman, *Jet Image: Deep Learning Edition*

In Preparation.

All simulations were performed with Pythia 8 and we made extensive use of Fastjet for jet clustering. The n-subjettiness ratios are used as a baseline.

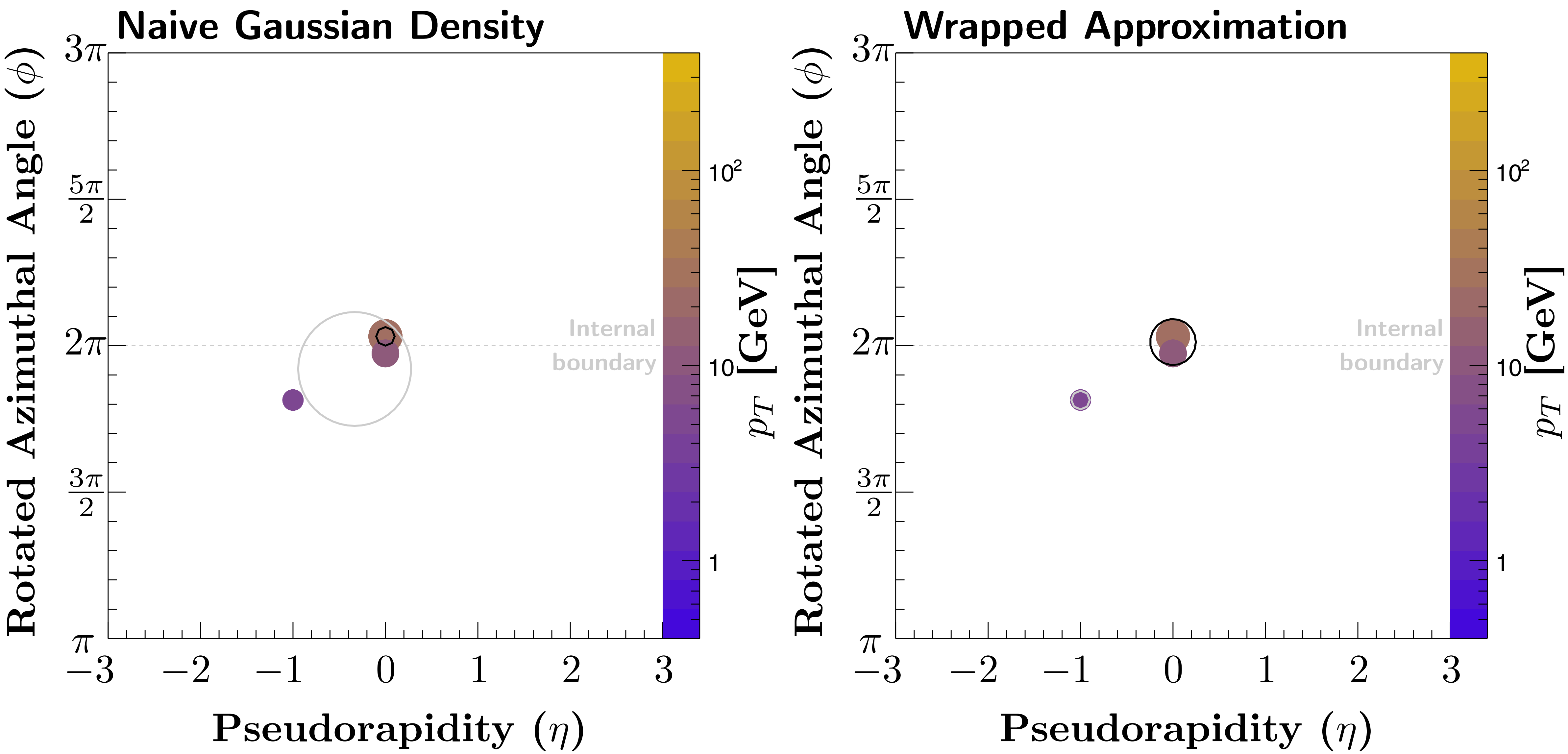
T. Sjöstrand, S. Mrenna and P. Skands, Comput. Phys. Comm. 178 (2008) 852 [arXiv:0710.3820]

M. Cacciari, G.P. Salam and G. Soyez, Eur. Phys. J. C72 (2012) 1896 [arXiv:1111.6097]

J. Thaler and K. Van Tilburg, JHEP 1103 (2011) 015 [arXiv:1011.2268]

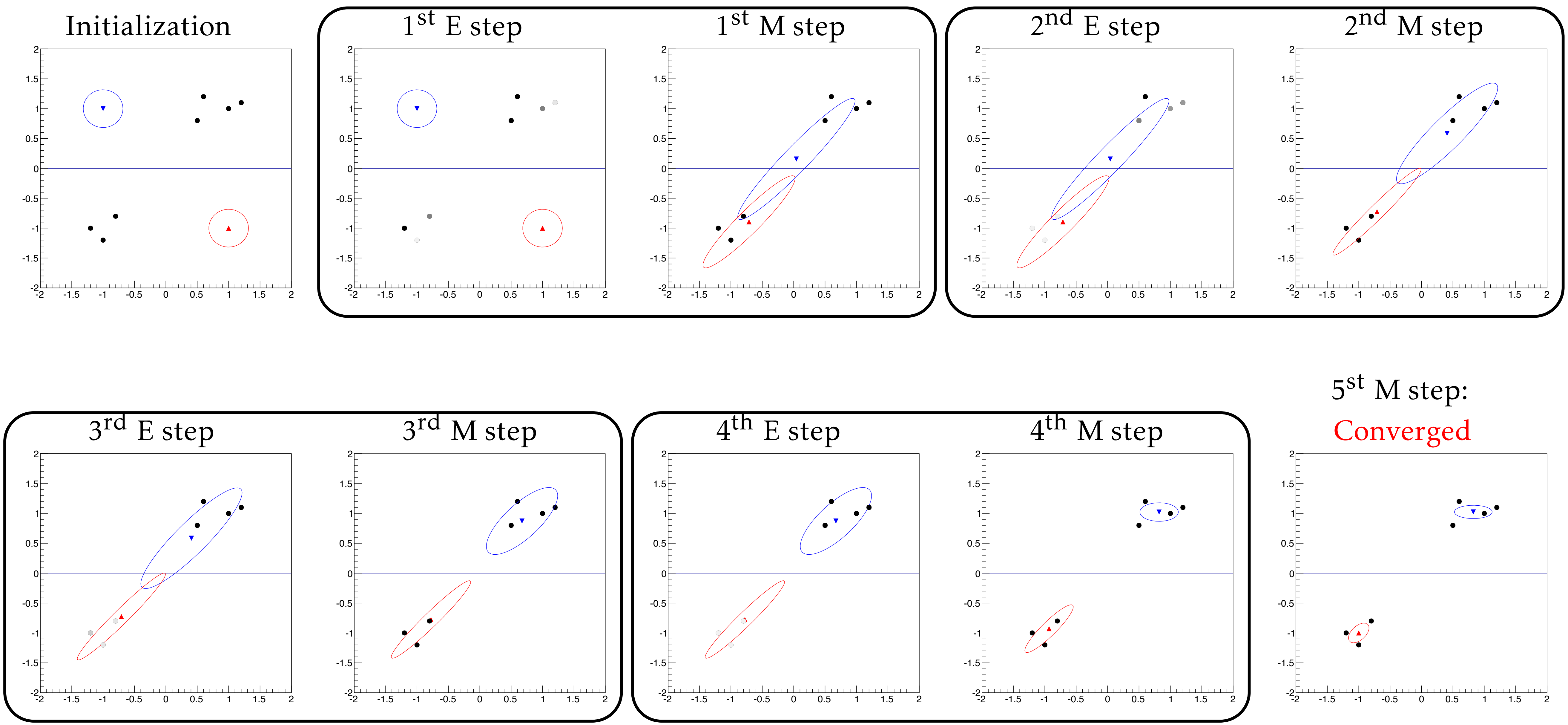
BACKUP

Food for thought - mixture modeling on a cylinder



Running fuzzy jets clustering

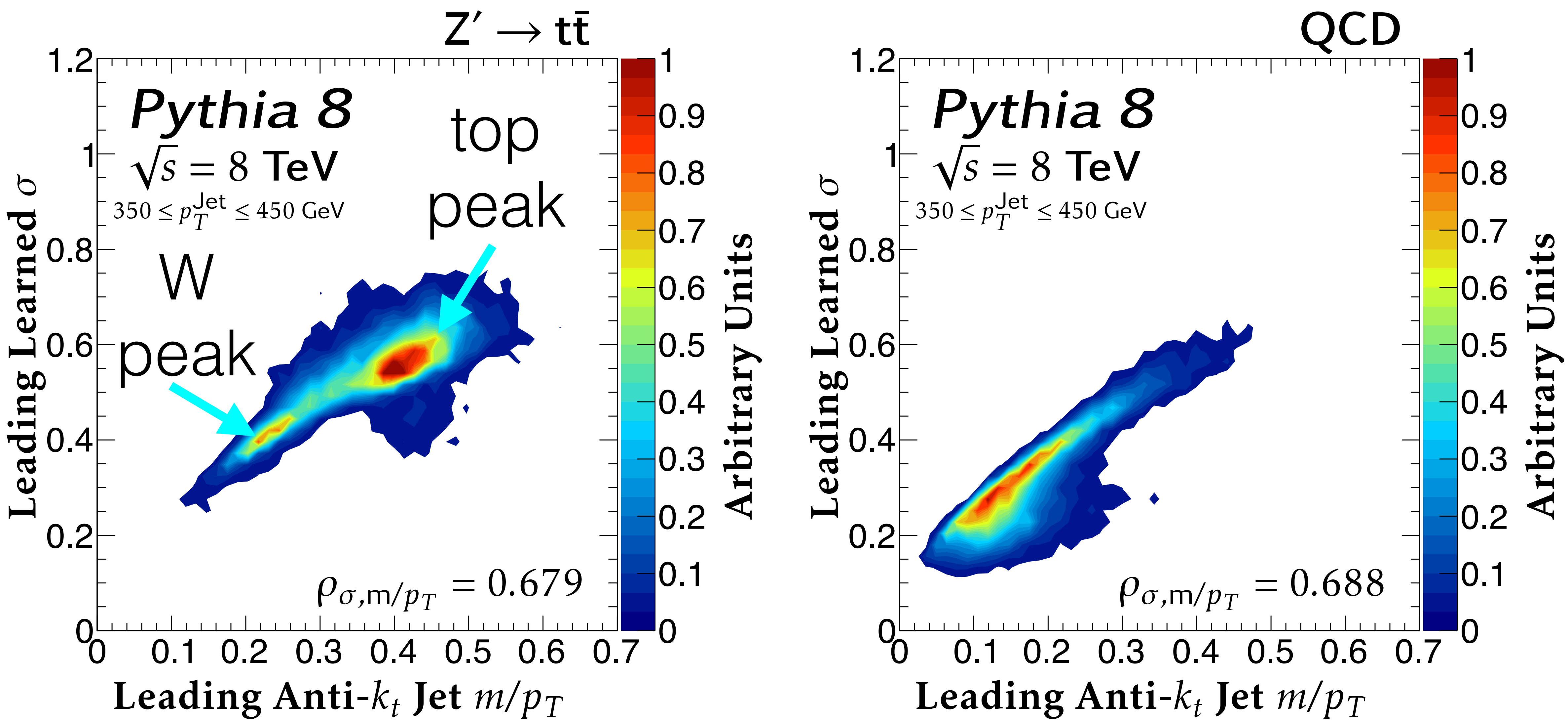
To minimize the IRC safe likelihood, we use an iterative procedure called the EM algorithm (illustrated below)



E step: Given jet locations, compute the probability for a particle i to belong to jet j.

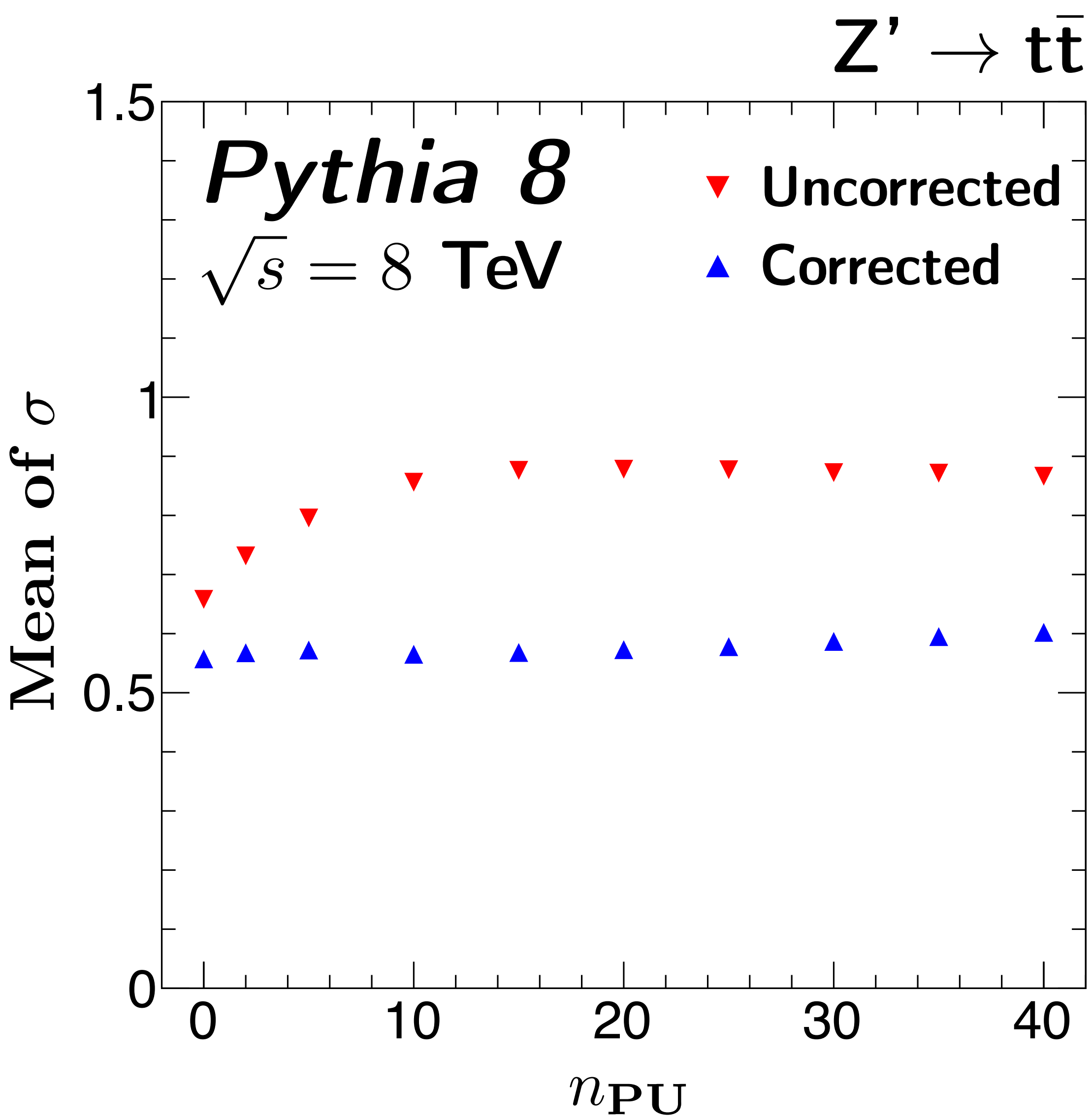
M step: Given the probabilities, compute the jet properties of jet j.

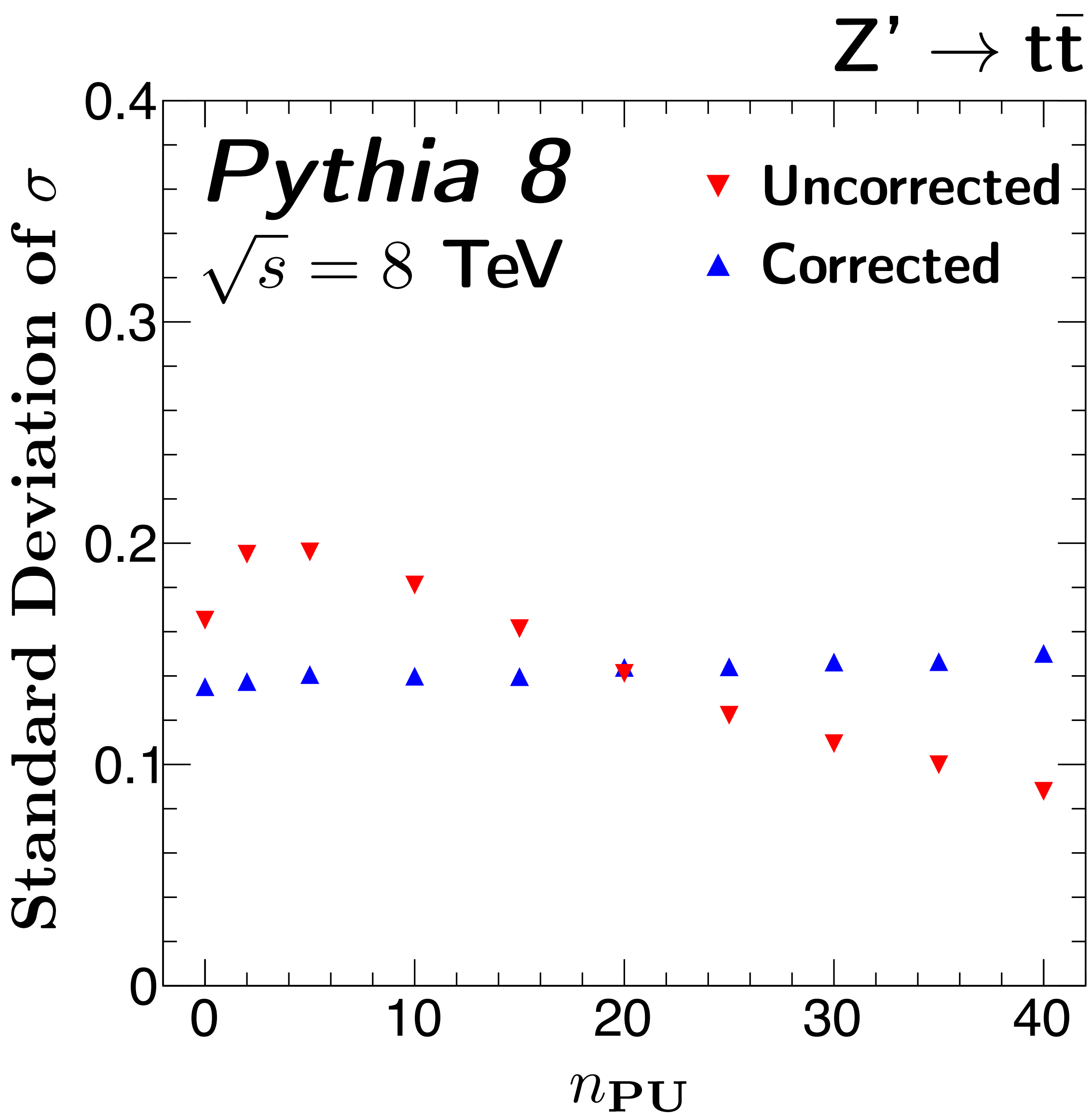
What is the relationship to anti-k_t quantities?



The leading size scale with m/p_T, but is not exactly the same (so there is something to add).

Pileup


Z' → $t\bar{t}$
Pythia 8
$\sqrt{s} = 8$ TeV
Uncorrected
Corrected
Mean of σ
n_{PU}
0
0.5
1
1.5
0
10
20
30
40


Z' → $t\bar{t}$
Pythia 8
$\sqrt{s} = 8$ TeV
Uncorrected
Corrected
Standard Deviation of σ
n_{PU}
0
0.1
0.2
0.3
0.4
0
10
20
30
40