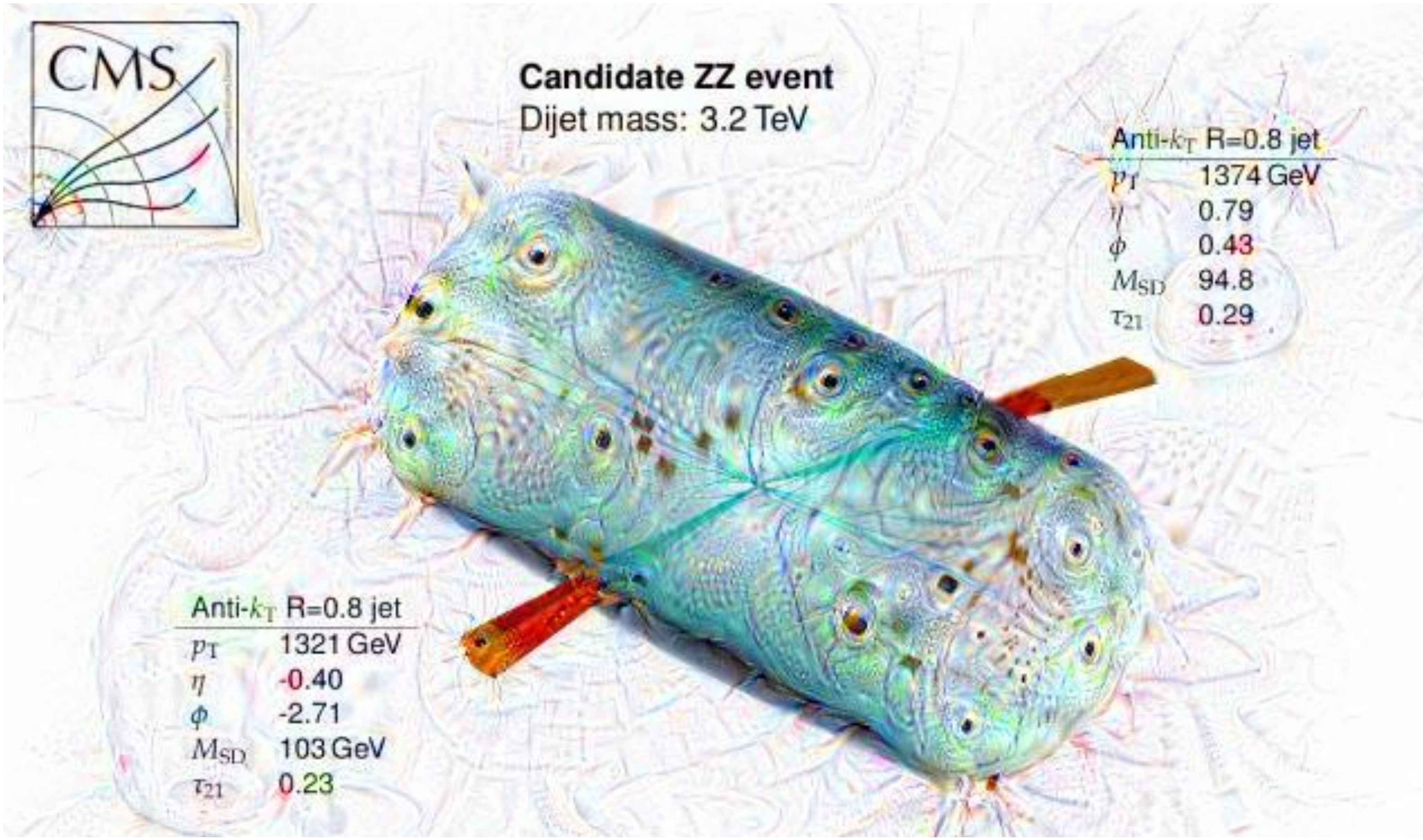


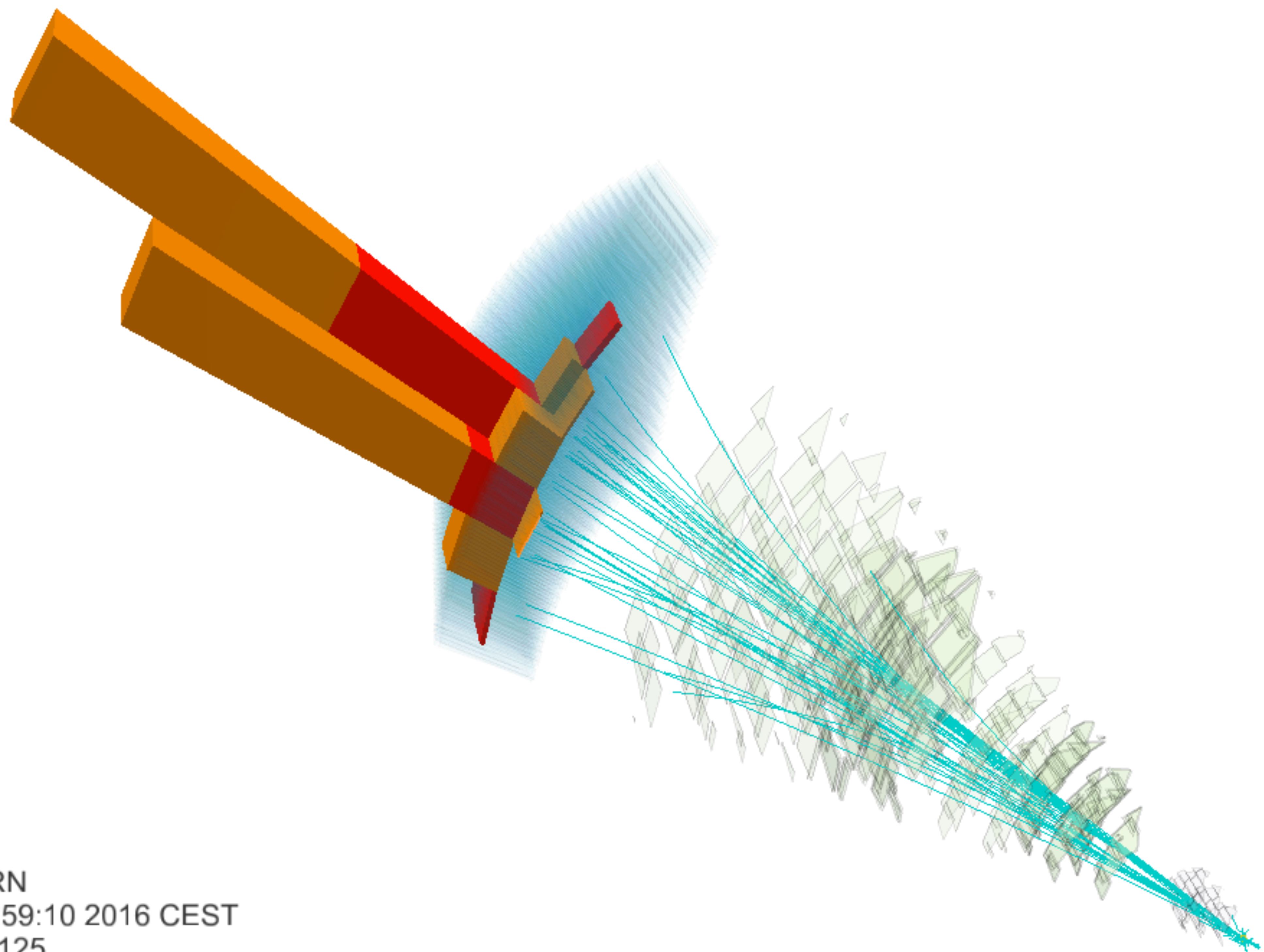
Lorentz Invariance Based DNN for W-tagging

Thea Klæboe Årrestad

PHYSIK INSTITUT
UNIVERSITÄT ZÜRICH

Joint CMS/LHCb Seminar
Physik Institut, May 4th

W/Z-tagging in CMS



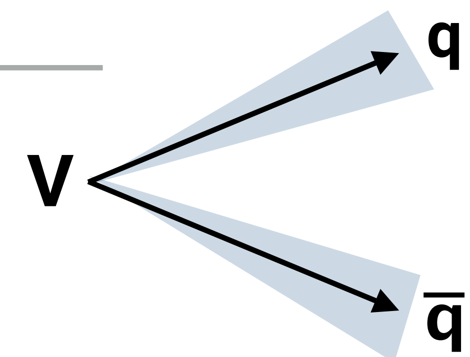
CMS-PAS-B2G-17-001

CMS Experiment at LHC, CERN
Data recorded: Mon Jul 18 19:59:10 2016 CEST
Run/Event: 276950 / 1080730125
Lumi section: 573

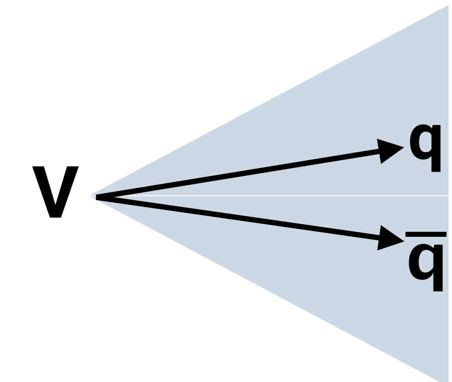
$V \rightarrow q\bar{q}$?

Universität Zürich[UZH]

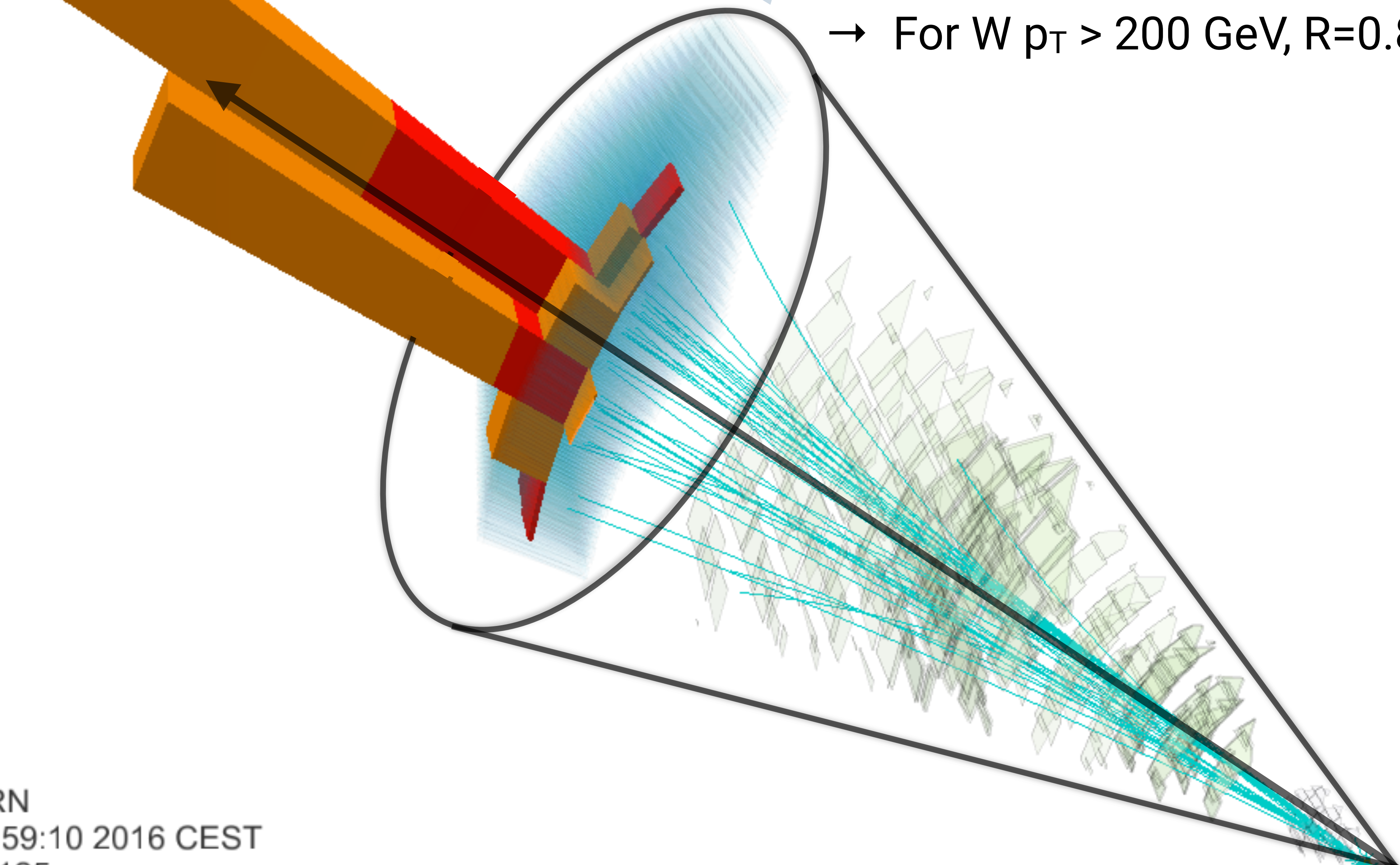
W/Z-tagging in CMS



$$p_{T,V} \gtrsim \frac{2m_V}{R}$$



→ For W p_T > 200 GeV, R=0.8

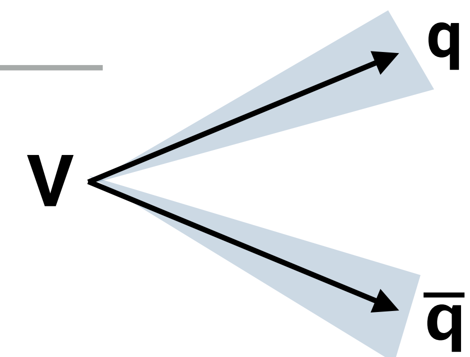


V→q$\bar{q}$?

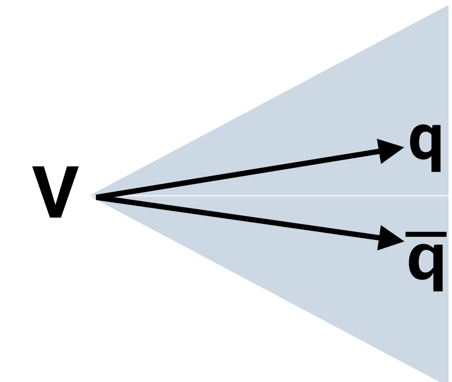
CMS-PAS-B2G-17-001

CMS Experiment at LHC, CERN
Data recorded: Mon Jul 18 19:59:10 2016 CEST
Run/Event: 276950 / 1080730125
Lumi section: 573

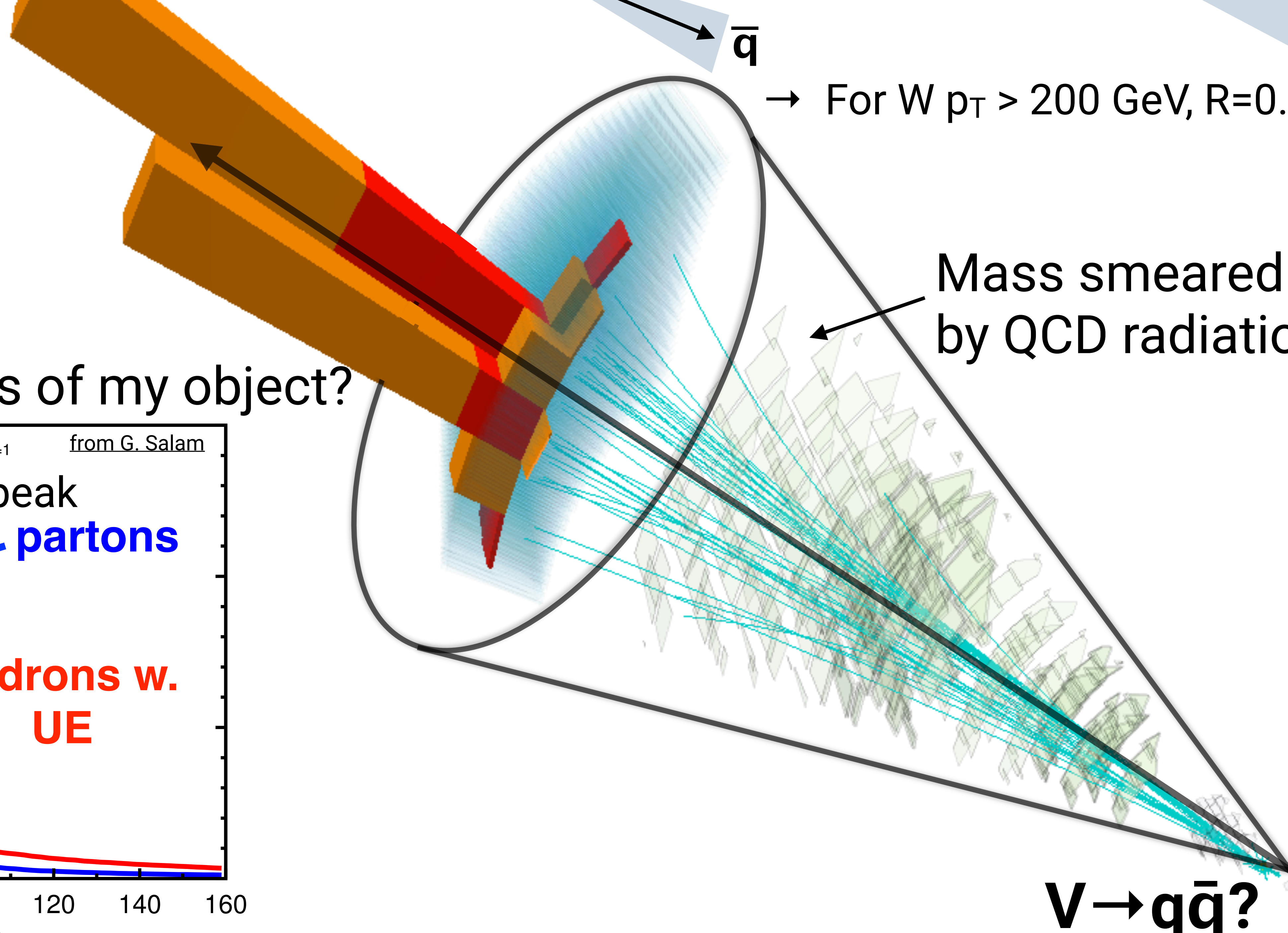
W/Z-tagging in CMS



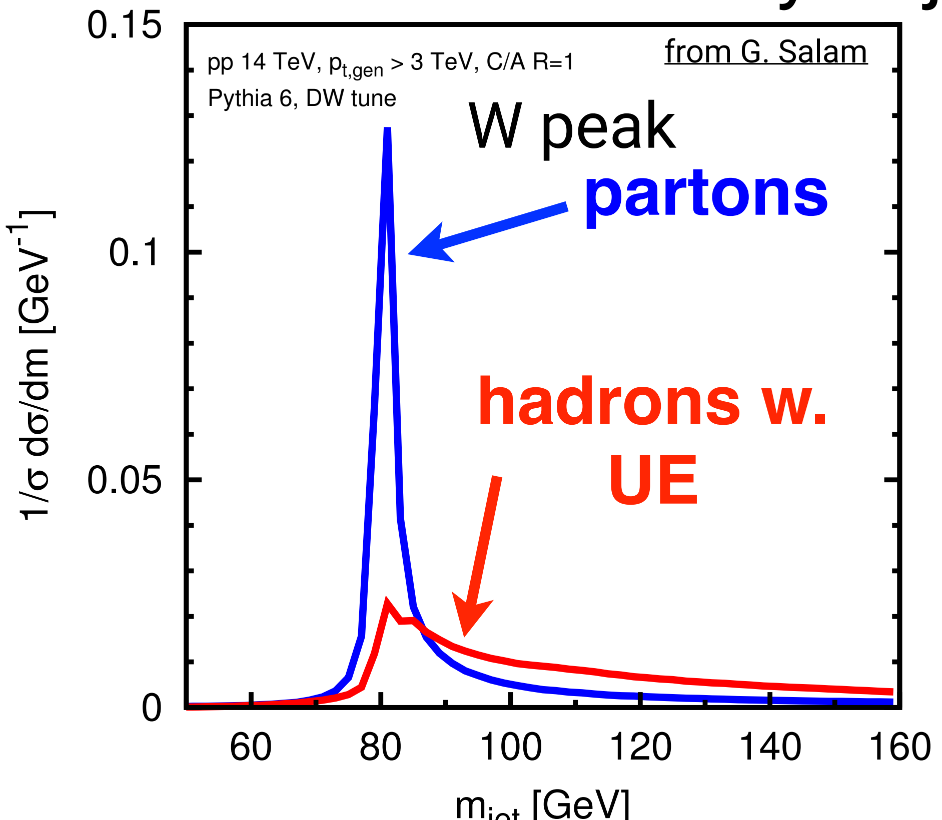
$p_{T,V} \gtrsim \frac{2m_V}{R}$



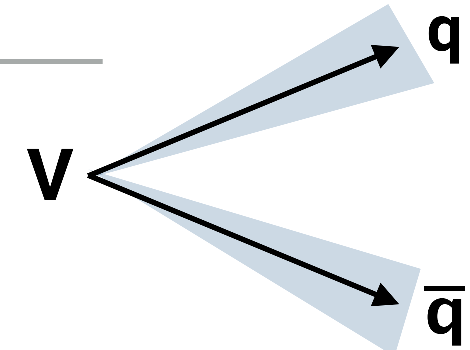
→ For W p_T > 200 GeV, R=0.8



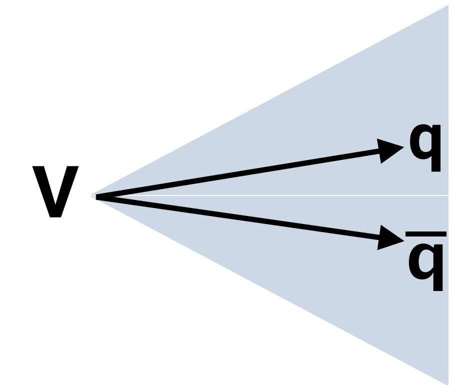
1) What's the mass of my object?



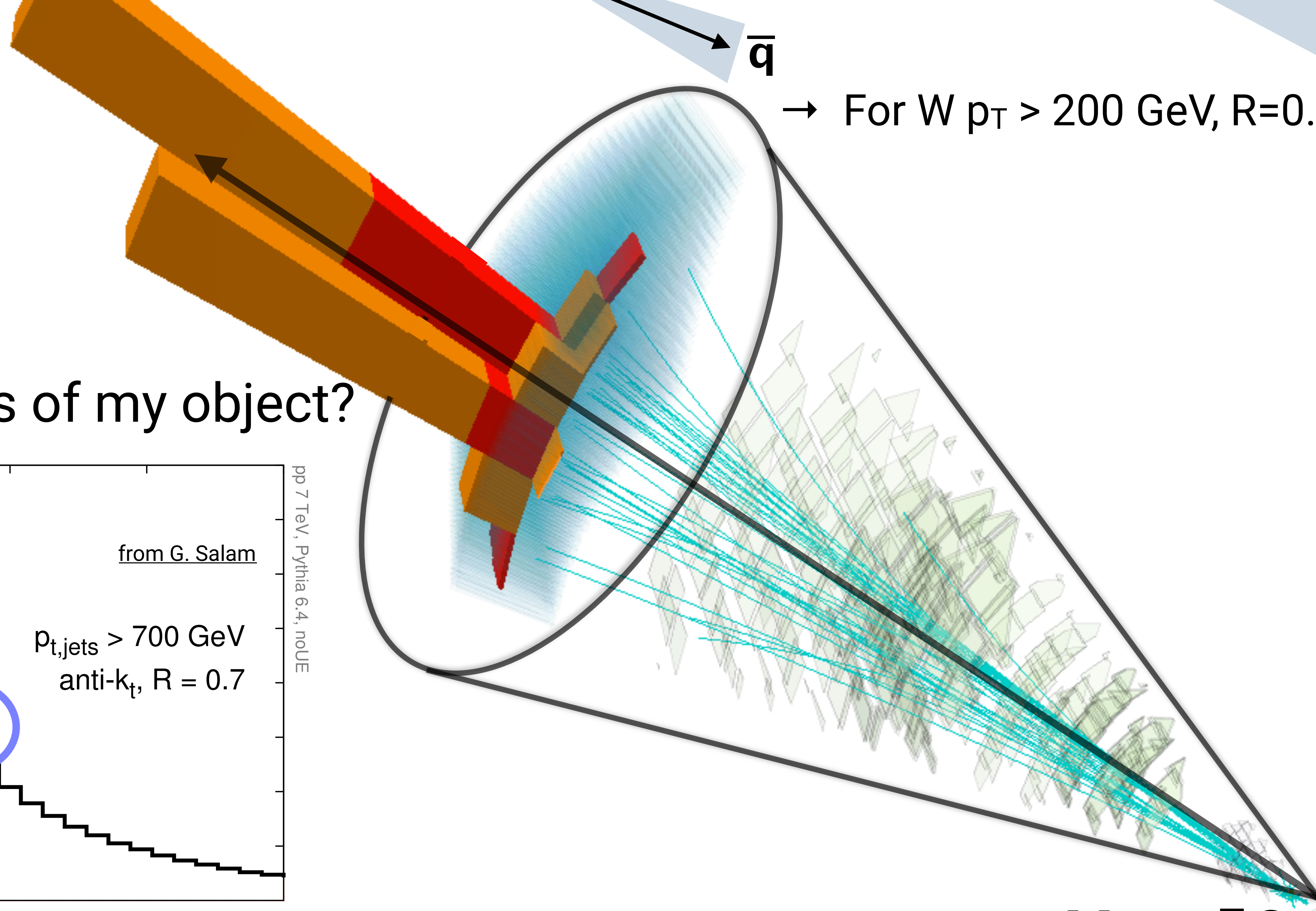
W/Z-tagging in CMS



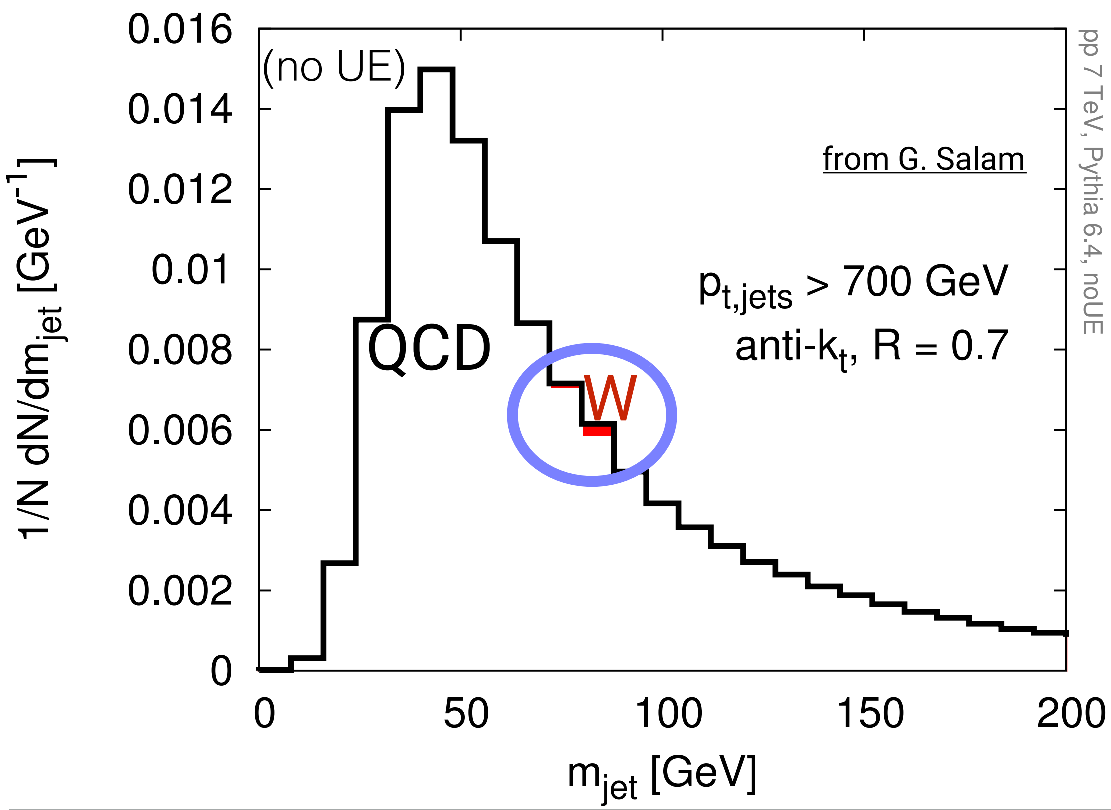
$$p_{T,V} \gtrsim \frac{2m_V}{R}$$



→ For W p_T > 200 GeV, R=0.8

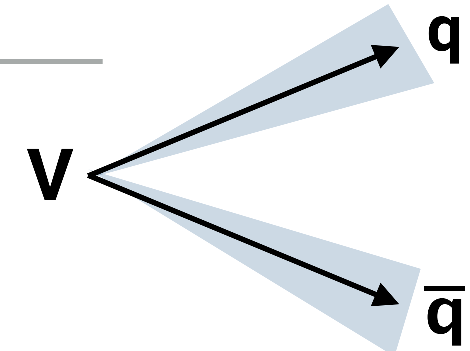


1) What's the mass of my object?

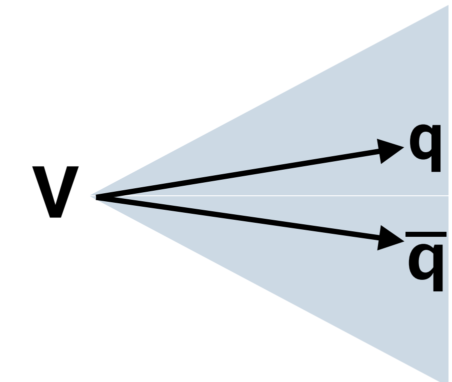


$V \rightarrow q\bar{q}$?

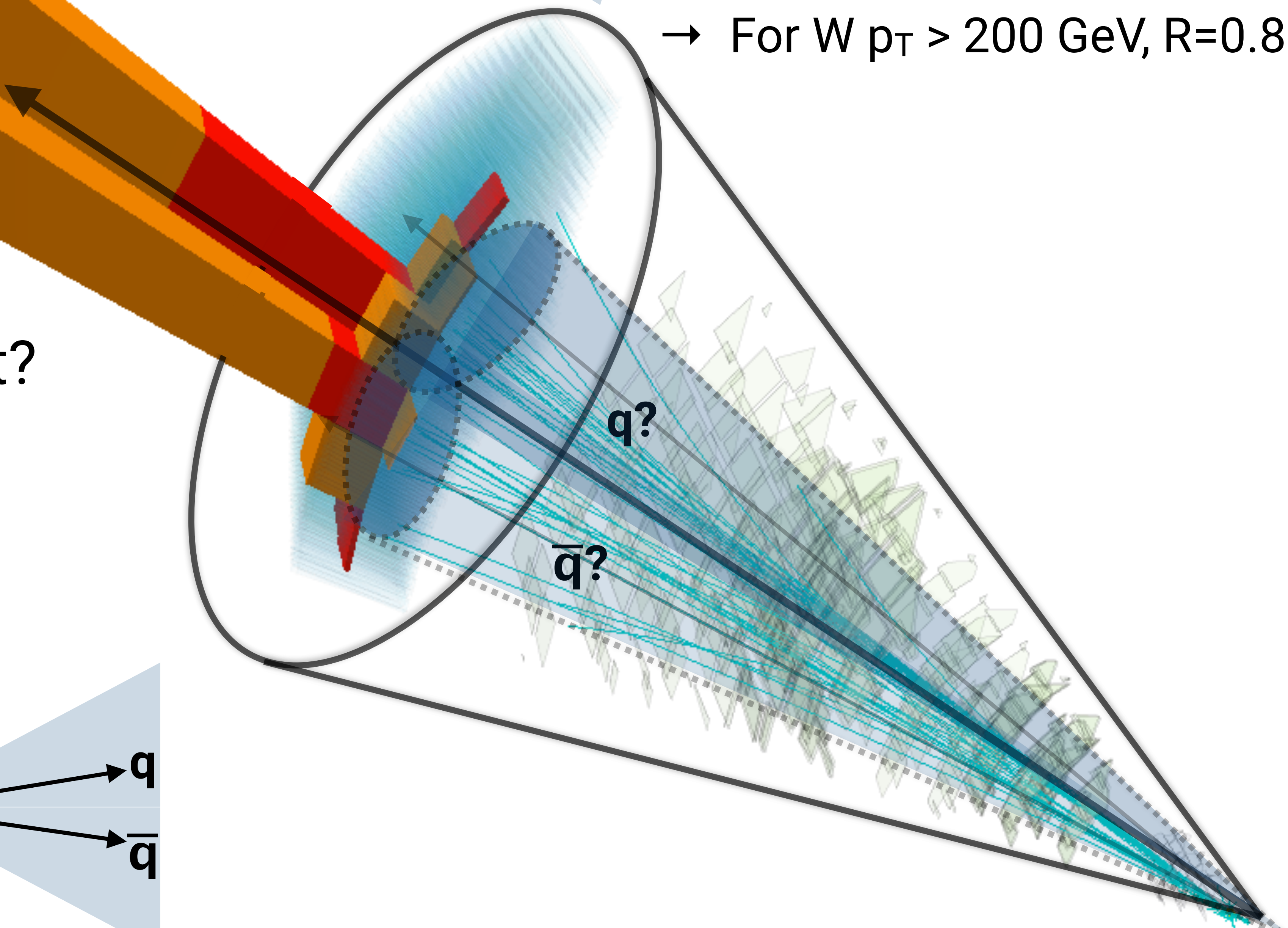
W/Z-tagging in CMS



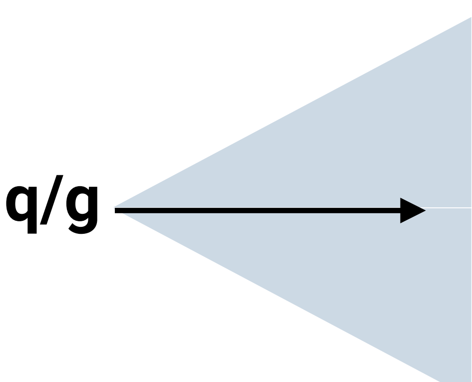
$$p_{T,V} \gtrsim \frac{2m_V}{R}$$



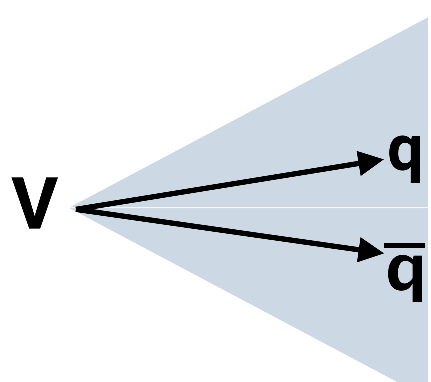
→ For W p_T > 200 GeV, R=0.8



2) Can I peak inside the jet?



vs.

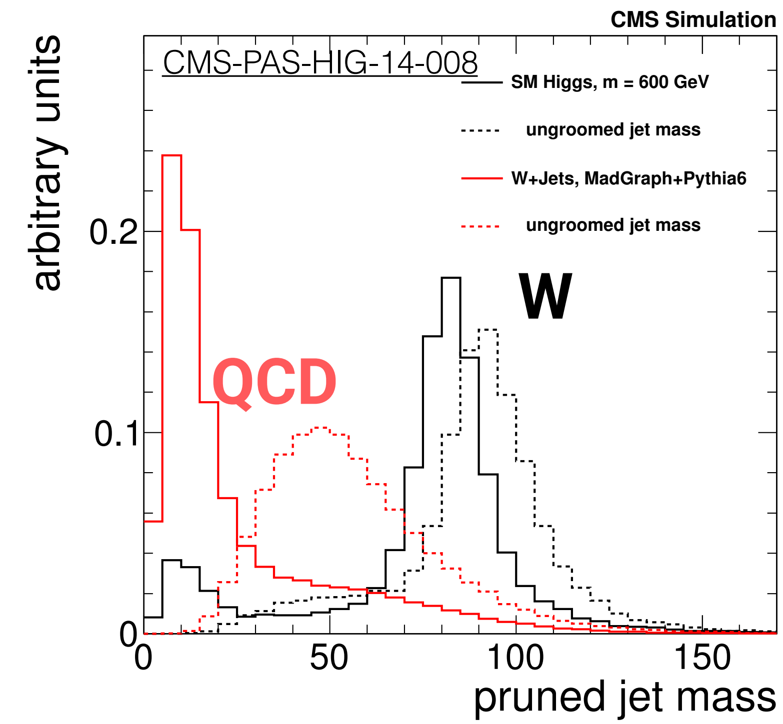


V→q$\bar{\text{q}}$?

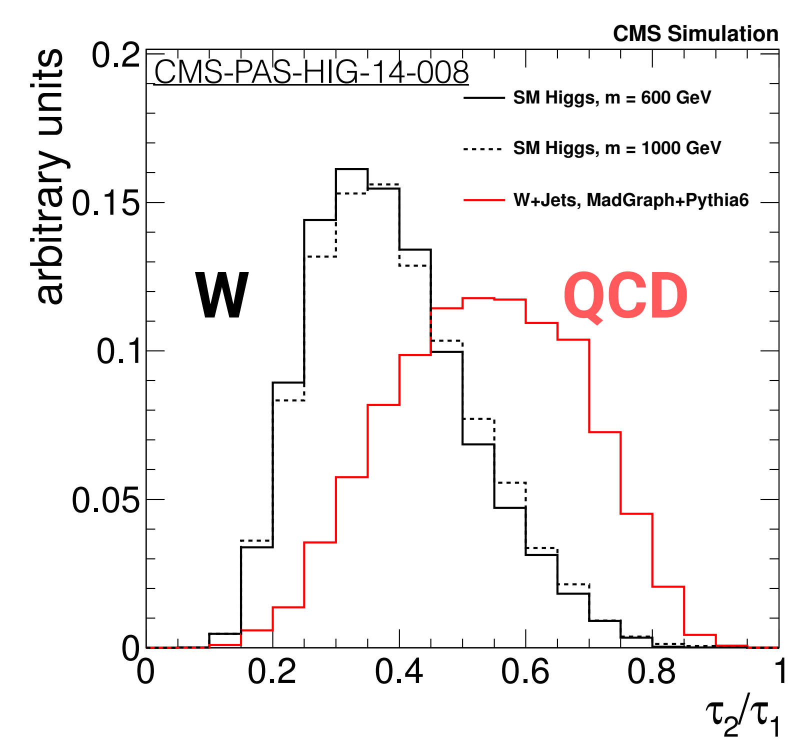
Traditional W-tagging

Cut-based taggers trying to answer

- Q: What's the mass of my object?
  A: *Grooming (pruning/SD): Remove soft and wide angle jet constituents*



- Q: Is there substructure?
  A: *N-subjettiness ++: Distance between jet constituents and hard subjet axes*



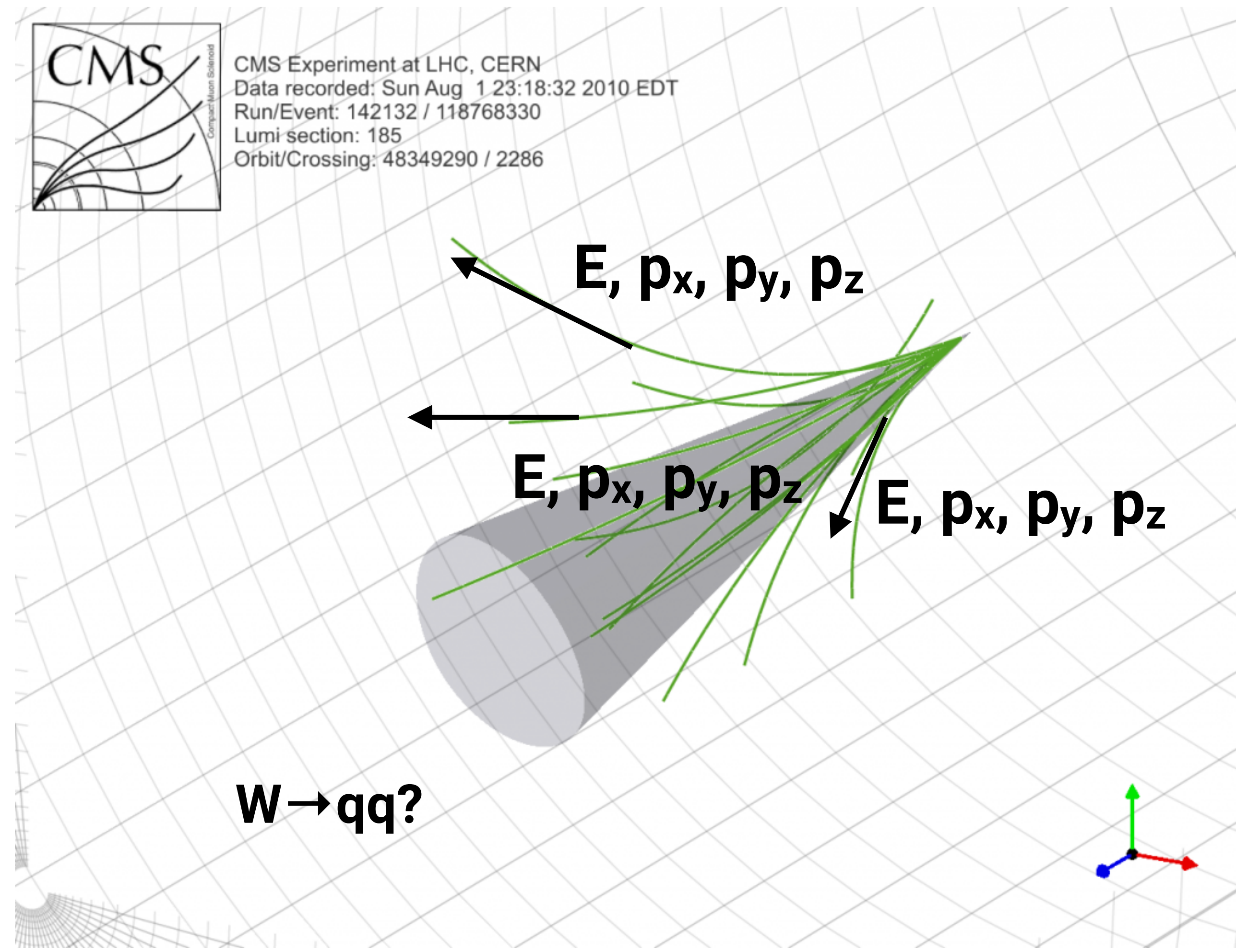
If a human can think up such algorithms, I bet a machine can too

- give DNN information it needs to design its own grooming/substructure algorithms

LoLa

DNN working with Lorentz vectors introduced for top-tagging by T. Plehn, G. Kasieczka et. Al (arXiv:1707.08966)

- physics based deep neural network
- **does not**: throw huge amounts of inputs into NN and eliminate through rankings
- **does**: analyse jet constituents directly, teach NN distances in Minkowski space



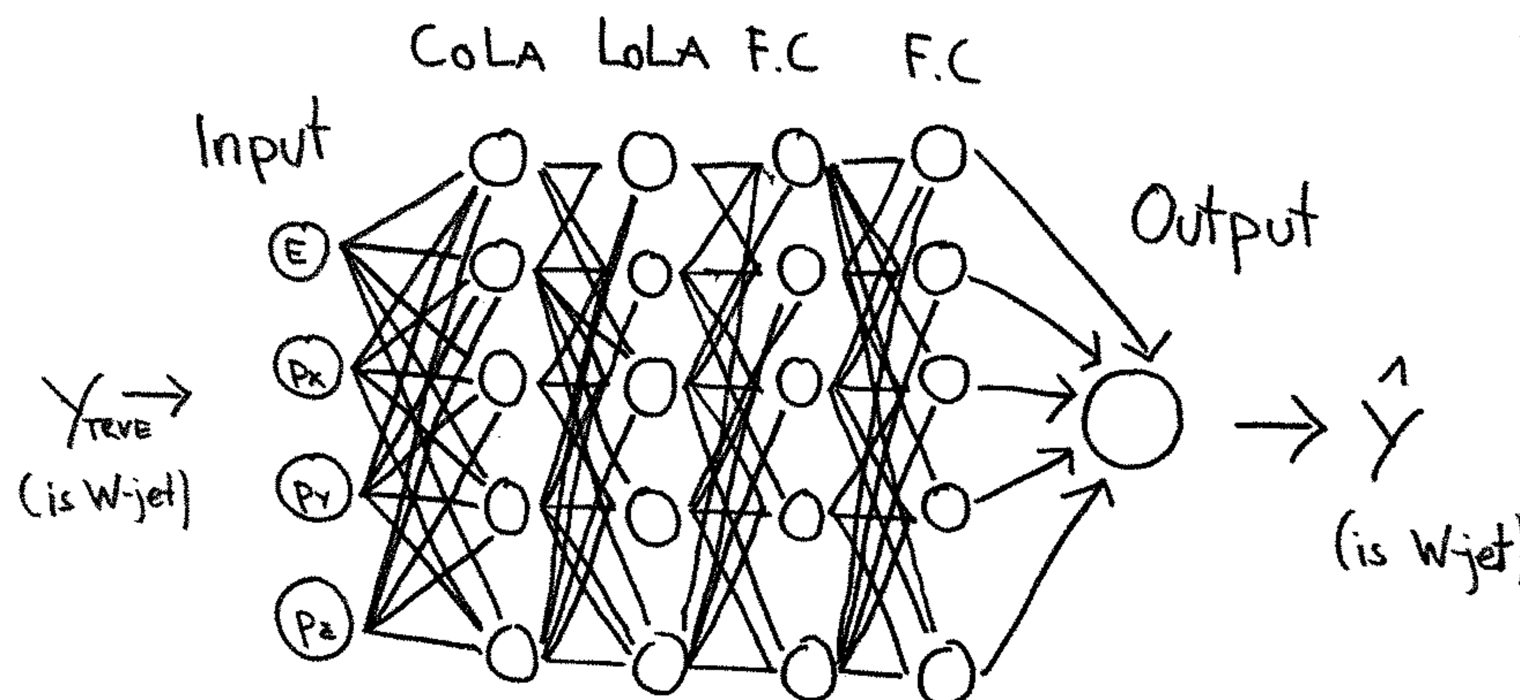
All substructure/grooming algorithms in CMS based on jet constituent 4-vectors

- by giving DNN tools to do jet substructure, can we learn substructure from LoLa instead of other way around

Network structure

4-layer DNN doing supervised learning with fixed-size input vectors

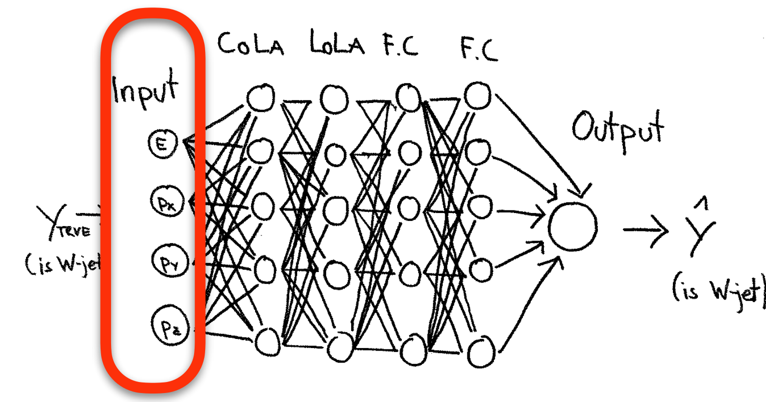
- feed forward sequential network
- Two novel layers (CoLa and LoLa) doing jet clustering and implementing Minkowski metric



Technicalities

- Keras w/ Theano backend (on Amazon)
- Loss function: Categorical crossentropy (output :W-jet/QCD probability)
- ADAM optimiser (adapt learning rate of model parameters during training)

Input



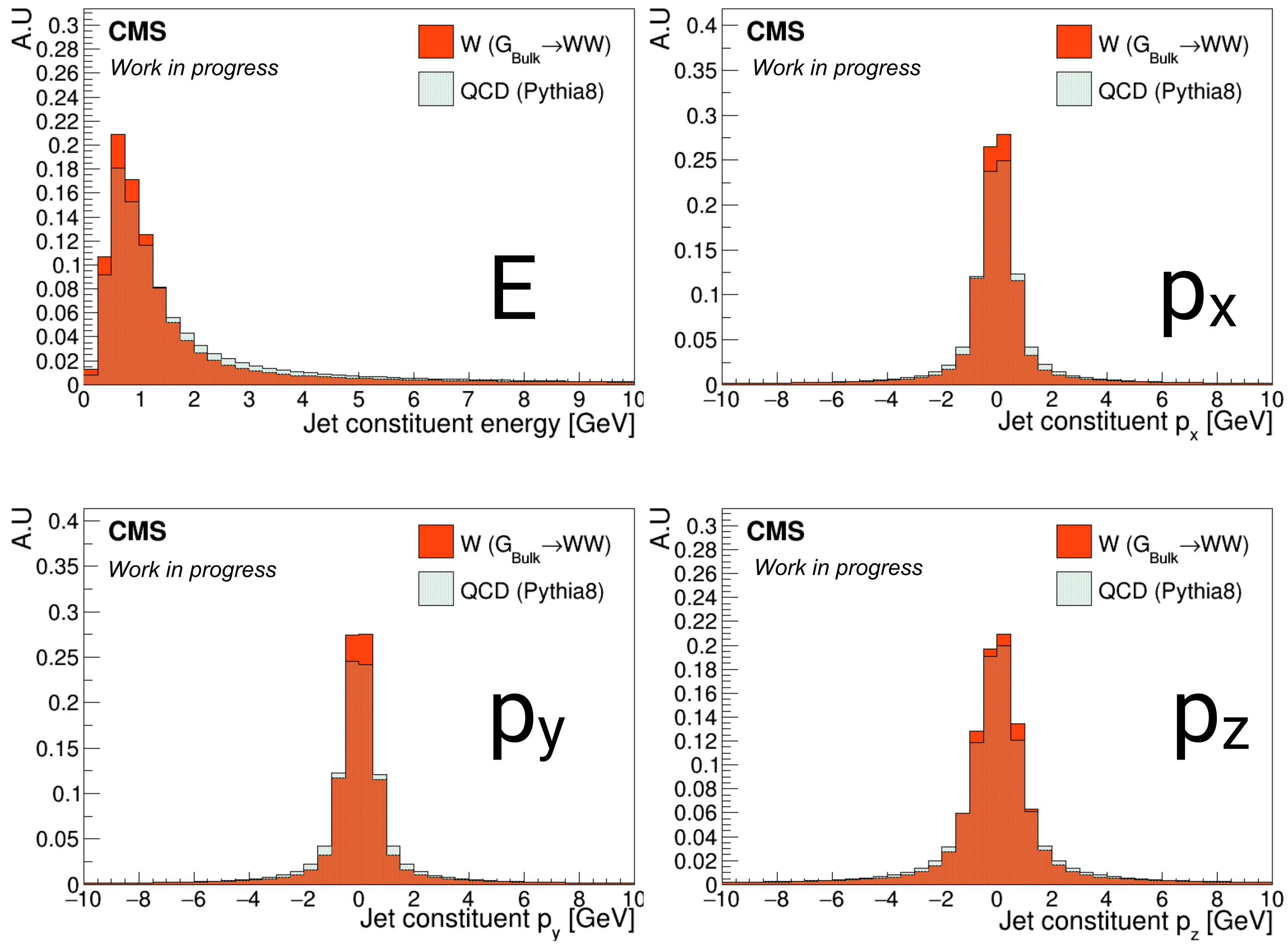
Four input features

- 4-vectors of the N=20 highest-p_T jet constituents of AK8 jets

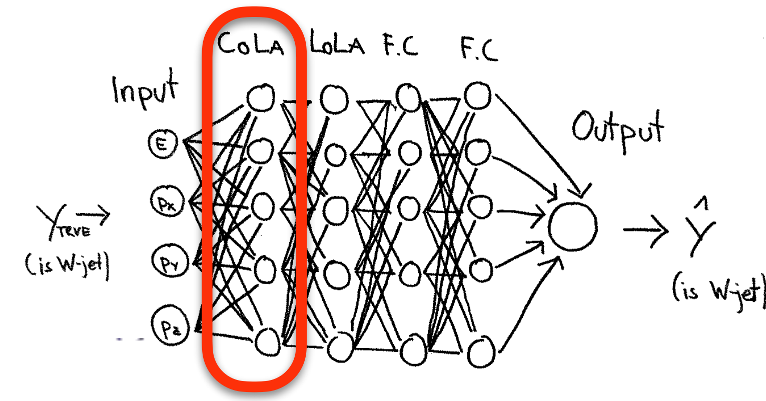
4x20 matrix $k_{\mu,i}$ for each jet

$$(k_{\mu,i}) = \begin{pmatrix} k_{0,1} & k_{0,2} & \cdots & k_{0,N} \\ k_{1,1} & k_{1,2} & \cdots & k_{1,N} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,N} \\ k_{3,1} & k_{3,2} & \cdots & k_{3,N} \end{pmatrix}$$

(4 Features x 20 constituents)



Combination Layer (CoLa)



COLA: E.g for 2 constituents

$$k_{\mu,i} = \begin{bmatrix} E_1 & E_2 \\ p_x^1 & p_x^2 \\ p_y^1 & p_y^2 \\ p_z^1 & p_z^2 \end{bmatrix} \quad (4,2)$$

$$\tilde{k}_{\mu,j} = k_{\mu,i}\boxed{C_{ij}} = \overbrace{\begin{bmatrix} E_1 & E_2 \\ p_x^1 & p_x^2 \\ p_y^1 & p_y^2 \\ p_z^1 & p_z^2 \end{bmatrix}}^{k_{\mu,i}}_{(4,2)} \cdot \overbrace{\begin{bmatrix} 1 & 1 & 0 & w_{4,1} \\ 1 & 0 & 1 & w_{4,2} \end{bmatrix}}^{\textbf{CoLa}}_{(2,M=4)} = \begin{bmatrix} E_1+E_2 & E_1 & E_2 & w_{4,1}E_1 + w_{4,2}E_2 \\ p_x^1+p_x^2 & p_x^1 & p_x^2 & w_{4,1}p_x^1 + w_{4,2}p_x^2 \\ p_y^1+p_y^2 & p_y^1 & p_y^2 & w_{4,1}p_y^1 + w_{4,2}p_y^2 \\ p_z^1+p_z^2 & p_z^1 & p_z^2 & w_{4,1}p_z^1 + w_{4,2}p_z^2 \end{bmatrix}$$

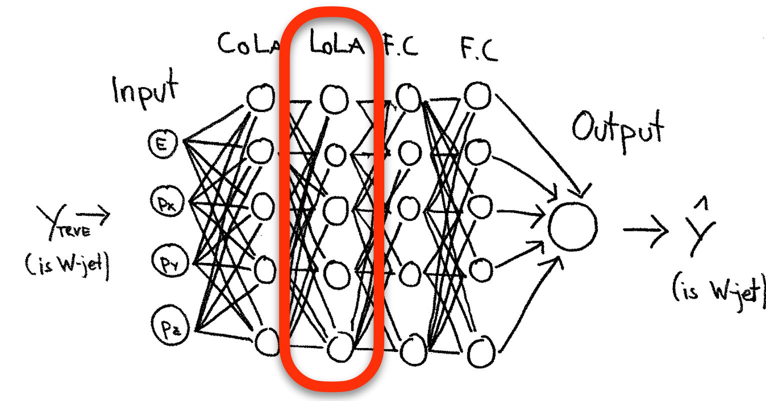
ΣE, Σp_x, Σp_y, Σp_z — p_4^1 — p_4^2 — linear combinations of momenta

Linear combinations similar to jet-clustering

- Sum of all momenta
- Each original constituent momenta
- Linear combinations + **trainable weights**. Can make subjets!

! Can "weight" constituents away, reconstruct hard subjets → groomer

Lorentz Layer (LoLa)



LoLA:

LoLa

$$\tilde{k}_{wj} = \begin{bmatrix} \Sigma E & E_1 & E_2 & w_{4,1}E_1 + w_{4,2}E_2 \\ \Sigma p_x & p_x^1 & p_x^2 & w_{4,1}p_x^1 + w_{4,2}p_x^2 \\ \Sigma p_y & p_y^1 & p_y^2 & w_{4,1}p_y^1 + w_{4,2}p_y^2 \\ \Sigma p_z & p_z^1 & p_z^2 & w_{4,1}p_z^1 + w_{4,2}p_z^2 \end{bmatrix}_{(4,4)} \rightarrow \hat{k}_j = \begin{bmatrix} m^2(\tilde{k}_j) \\ p_T(\tilde{k}_j) \\ w^E_{jm} E(\tilde{k}_m) \\ w^{1d}_{jm} \Sigma d^2_{jm} \\ w^{2d}_{jm} \Sigma d^2_{jm} \\ w^{3d}_{jm} \min d^2_{jm} \\ w^{4d}_{jm} \min d^2_{jm} \end{bmatrix}_{(7,4)} = \begin{bmatrix} g_{\mu\nu} P^{\mu}_{\Sigma p_4} P^{\nu}_{\Sigma p_4} & \cdots \\ \left|\sqrt{\Sigma p_x^2 + \Sigma p_y^2}\right| & \cdots \\ w_{jm} \Sigma E & \cdots \\ w_1^{min} \cdot & \cdots \\ w_2^{min} \cdot & \cdots \\ w_1^{sum} \cdot & \cdots \\ w_2^{sum} \cdot & \cdots \end{bmatrix}$$

$$\left\{ (\tilde{k}_{\Sigma} - \tilde{k}_m)_{\mu}\, g^{\mu\nu}\, (\tilde{k}_{\Sigma} - \tilde{k}_m)_{\nu} \cdots \right.$$

Minkowski metric explicitly used for m^2 and d

Maps CoLa output onto

- $m^2 + p_T$ of each column ("jet", constituents, hard subjets)
- Energy of all constituents (with trainable weight)
- Distance between all particles (2*min+ 2*sum)
  → n-subjetiness

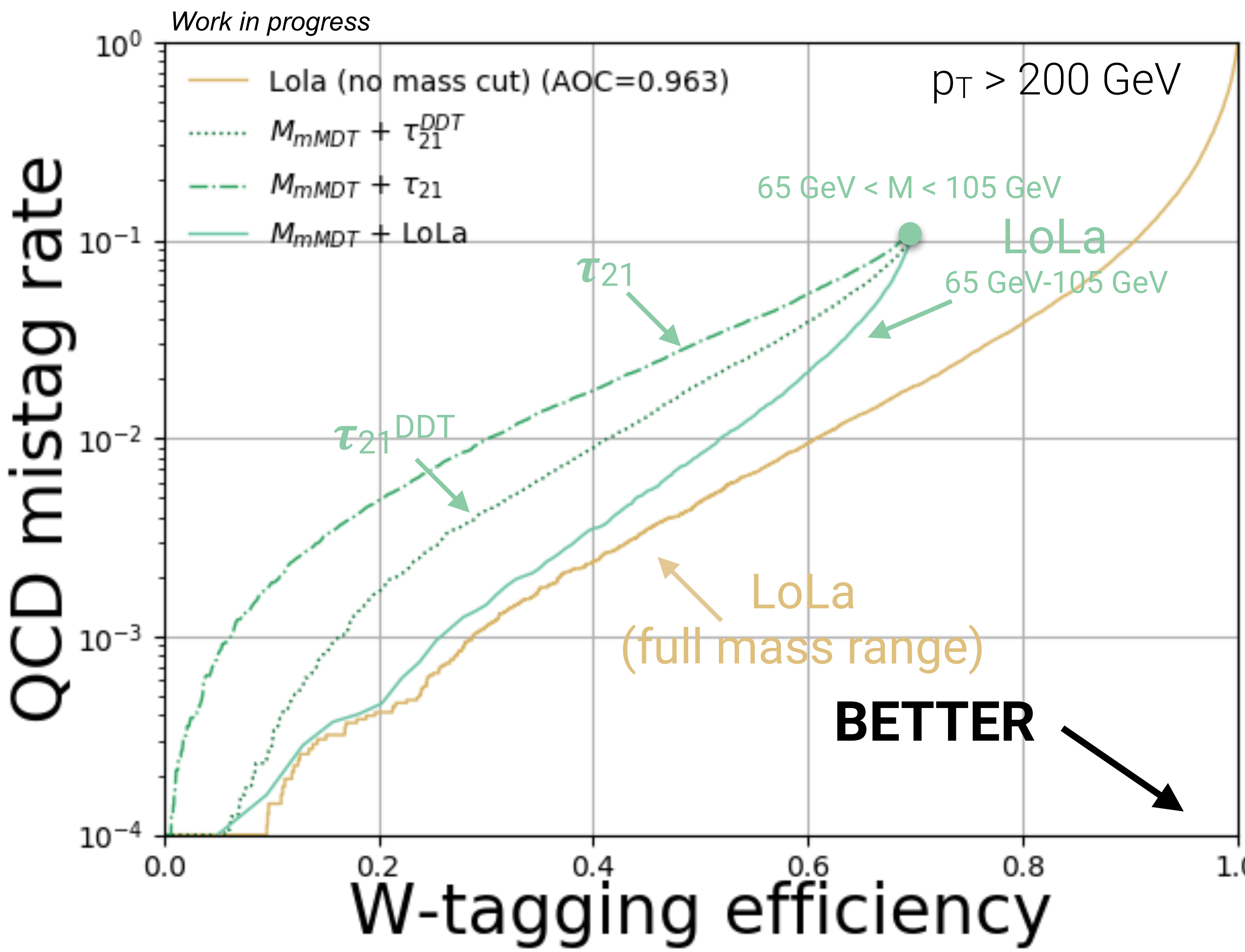
Overall performance

Compare performance to most commonly used cut based V-taggers

- LoLa performs significantly better than current baseline
  - 20% higher ε_S at given ε_B compared to best cut-based
  - no need for mass window, increased signal acceptance

For two-W final state, 43% increase in signal efficiency*

We all know DNNs do better. Whats next?



*B2G-17-001

Beyond performance

Beyond performance

Three things to consider when making a DNN tagger:

- is the absolute performance better (compared to common methods, a standard BDT)?

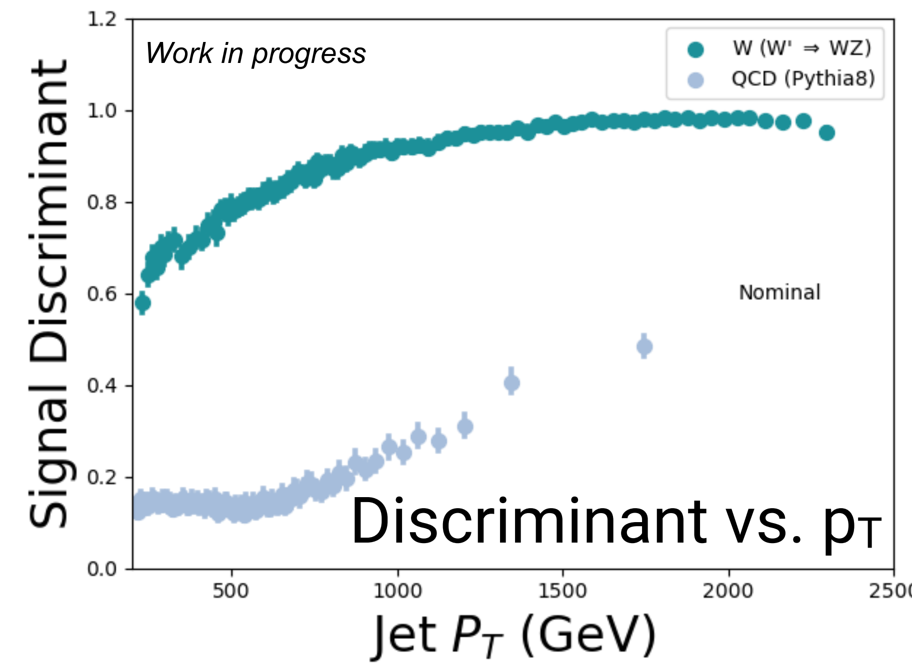
Beyond performance

Three things to consider when making a DNN tagger:

- is the absolute performance better (compared to common methods, a standard BDT)?

- is the tagger p_T-dependent?

Output strongly correlated with p_T/mass

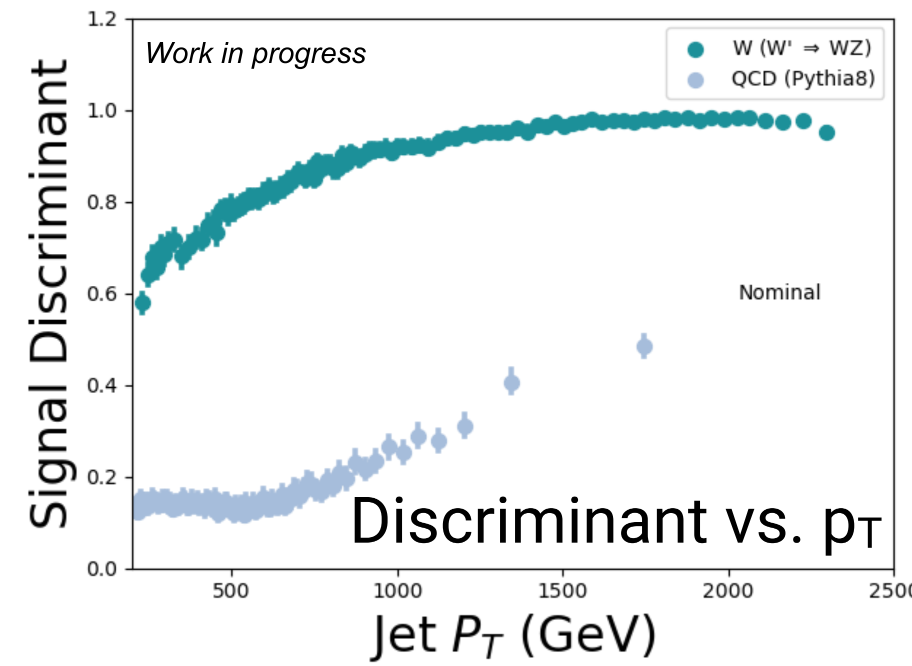


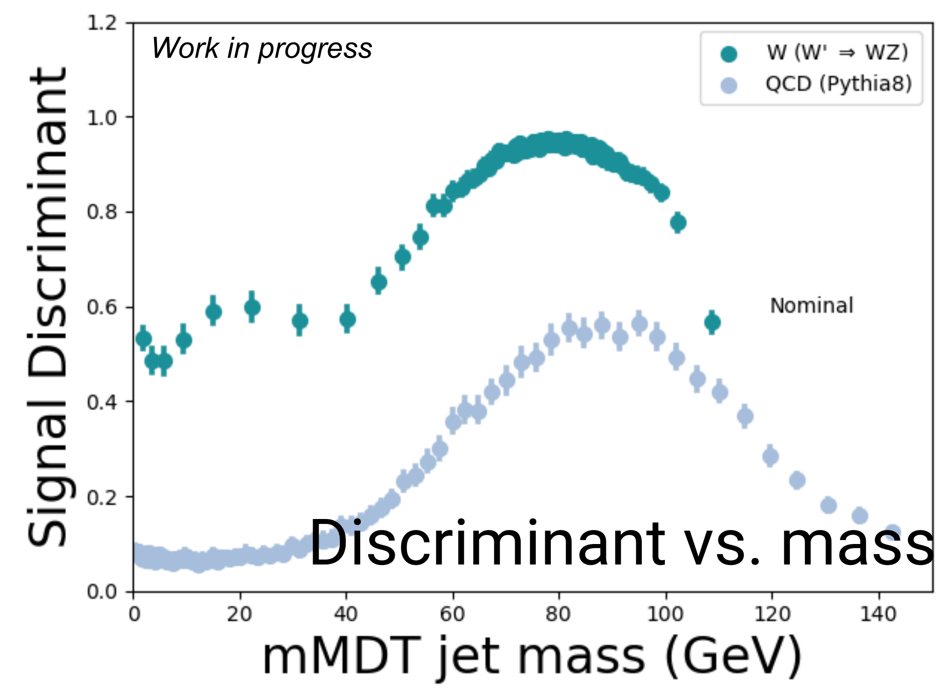
Beyond performance

Three things to consider when making a DNN tagger:

- is the absolute performance better (compared to common methods, a standard BDT)?
- is the tagger p_T-dependent?
- does the tagger sculpt the mass spectrum?

Output strongly correlated with p_T/mass





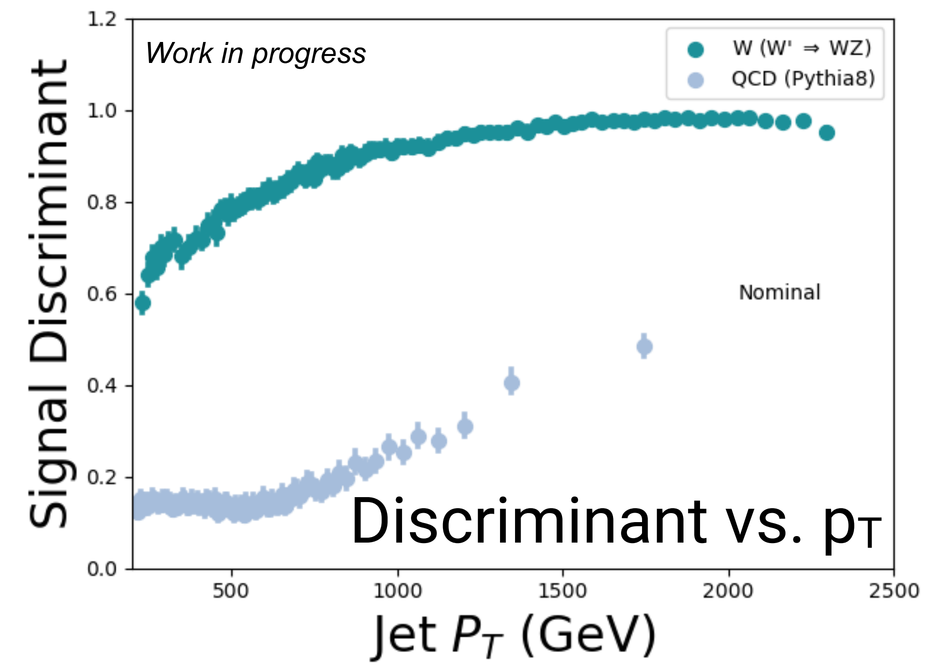
Beyond performance

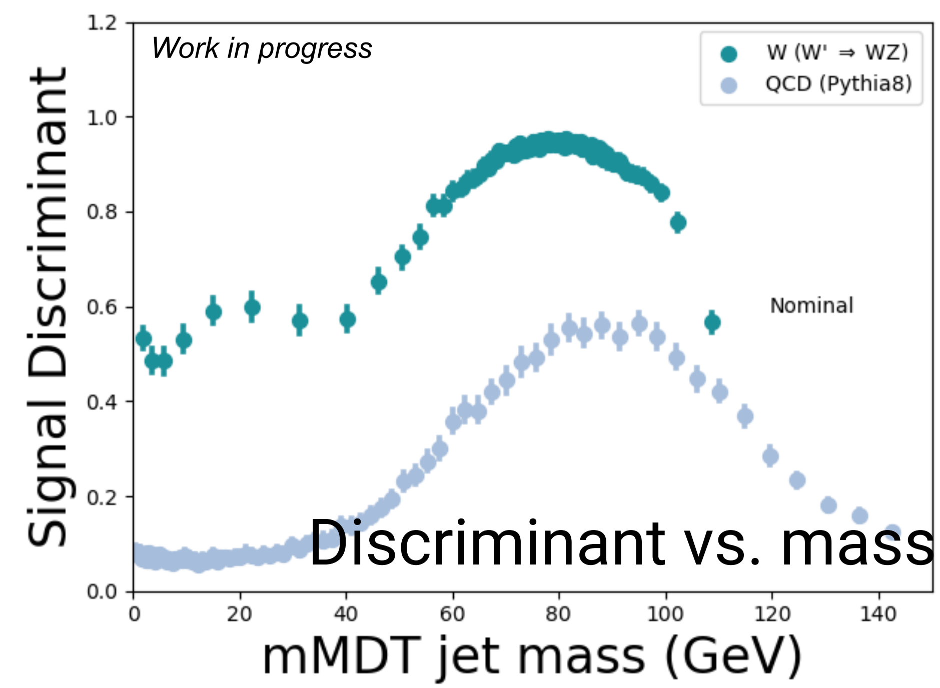
Three things to consider when making a DNN tagger:

- is the absolute performance better (compared to common methods, a standard BDT)?
- is the tagger p_T-dependent?
- does the tagger sculpt the mass spectrum?

These three measures are equally important in quantifying performance

Output strongly correlated with p_T/mass





Beyond performance

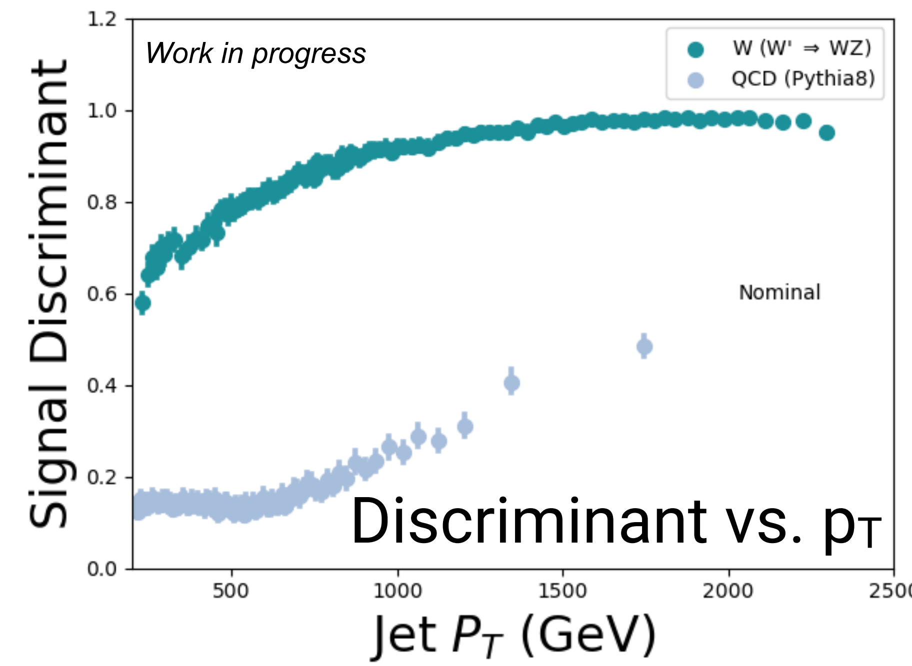
Three things to consider when making a DNN tagger:

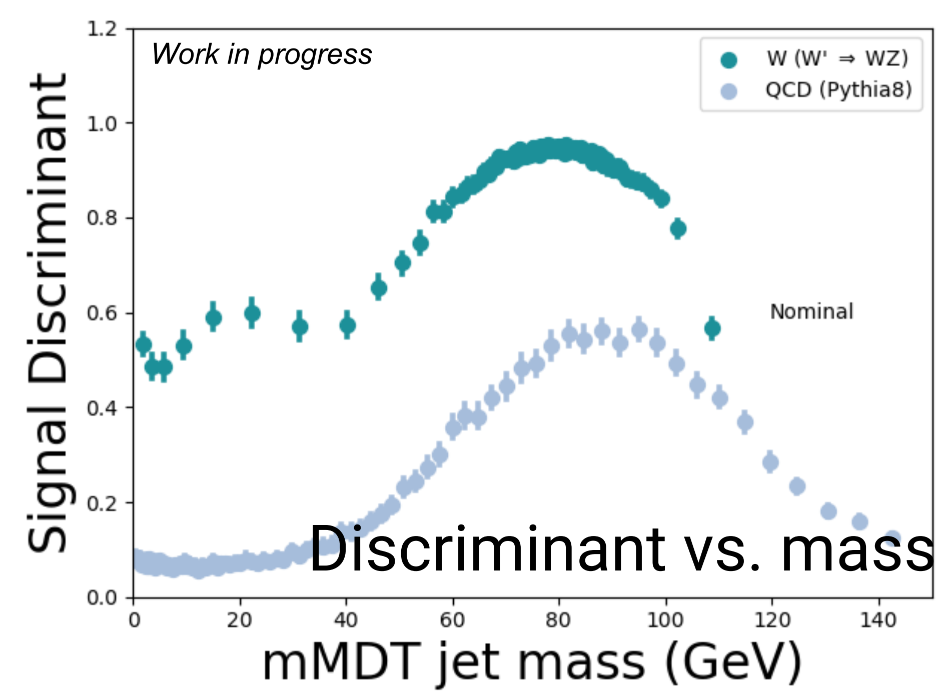
- is the absolute performance better (compared to common methods, a standard BDT)?
- is the tagger p_T-dependent?
- does the tagger sculpt the mass spectrum?

These three measures are equally important in quantifying performance

A DNN will naturally learn that p_T and mass are discriminating variables unless you penalise it for it!

Output strongly correlated with p_T/mass





Beyond performance

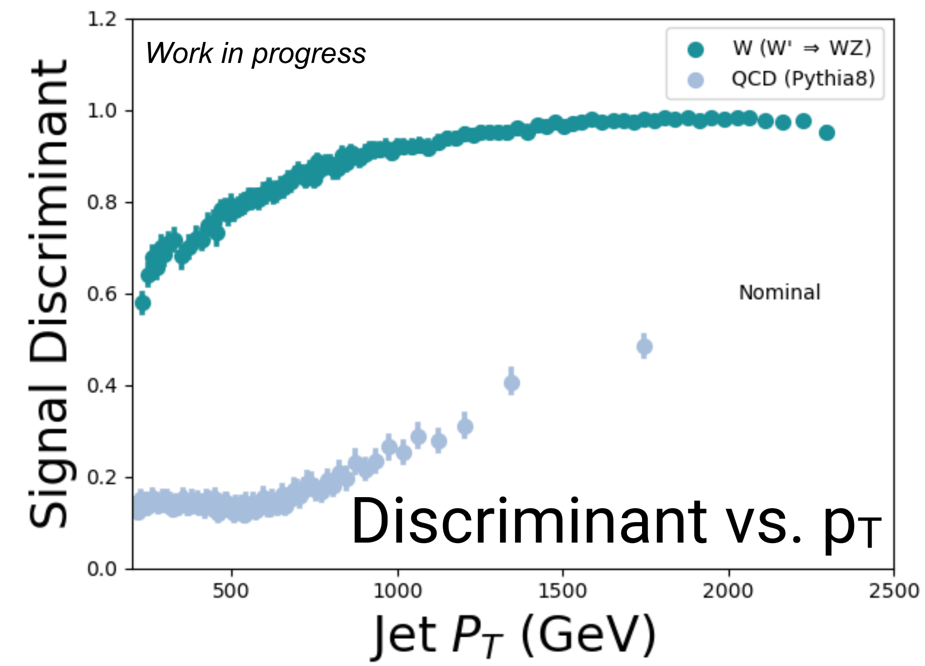
Three things to consider when making a DNN tagger:

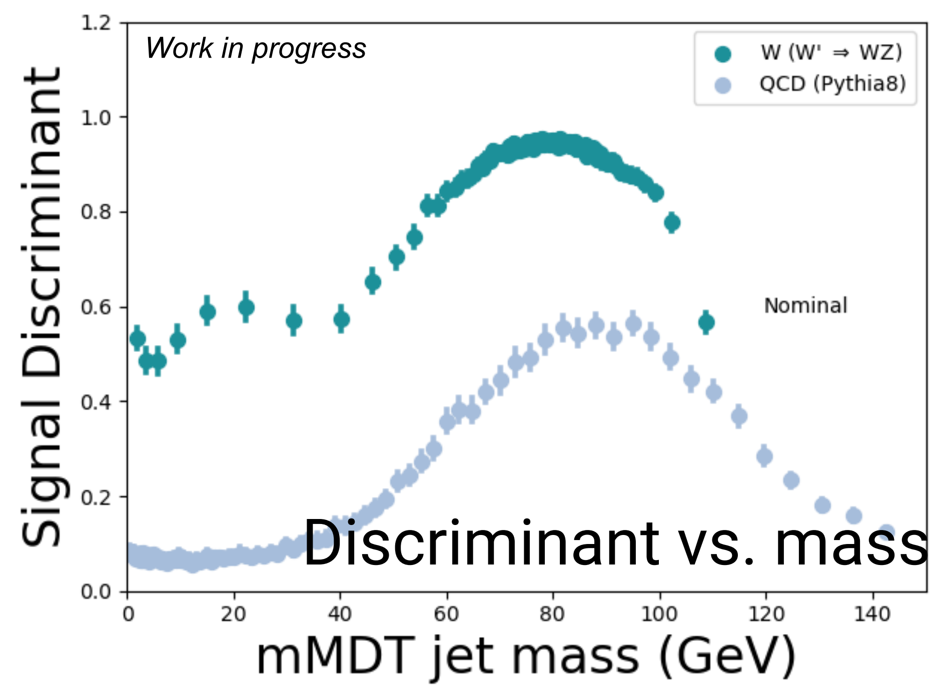
- is the absolute performance better (compared to common methods, a standard BDT)?
- is the tagger p_T-dependent?
- does the tagger sculpt the mass spectrum?

These three measures are equally important in quantifying performance

A DNN will naturally learn that p_T and mass are discriminating variables unless you penalise it for it!

Output strongly correlated with p_T/mass





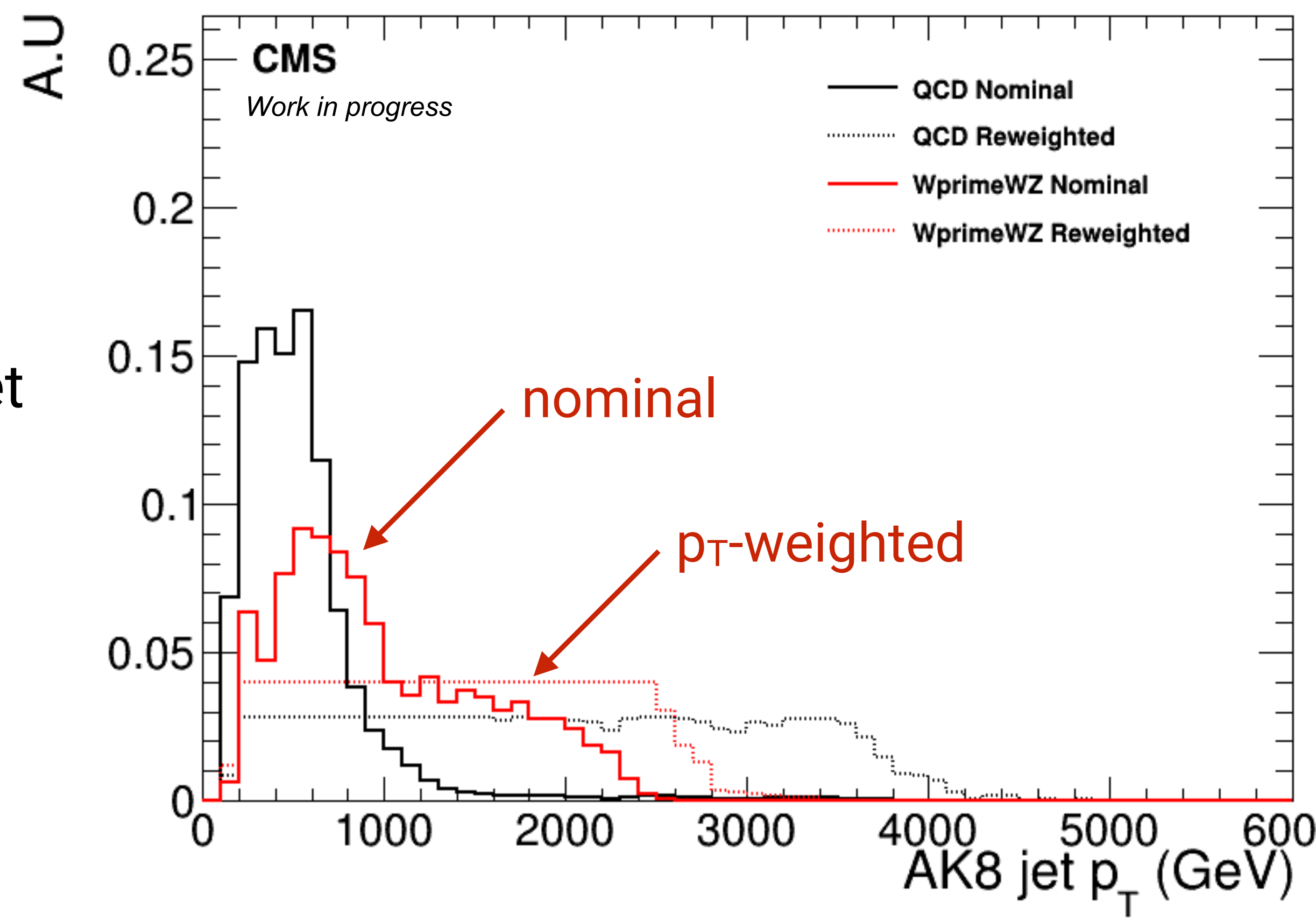
p_T dependence

p_T-dependence is a problem because

- signal efficiency is variable, requires working point scaling with p_T
- p_T (tagger validation region) != p_T (signal region)

One method to cope: reweight training set event-by-event to be flat in p_T-space

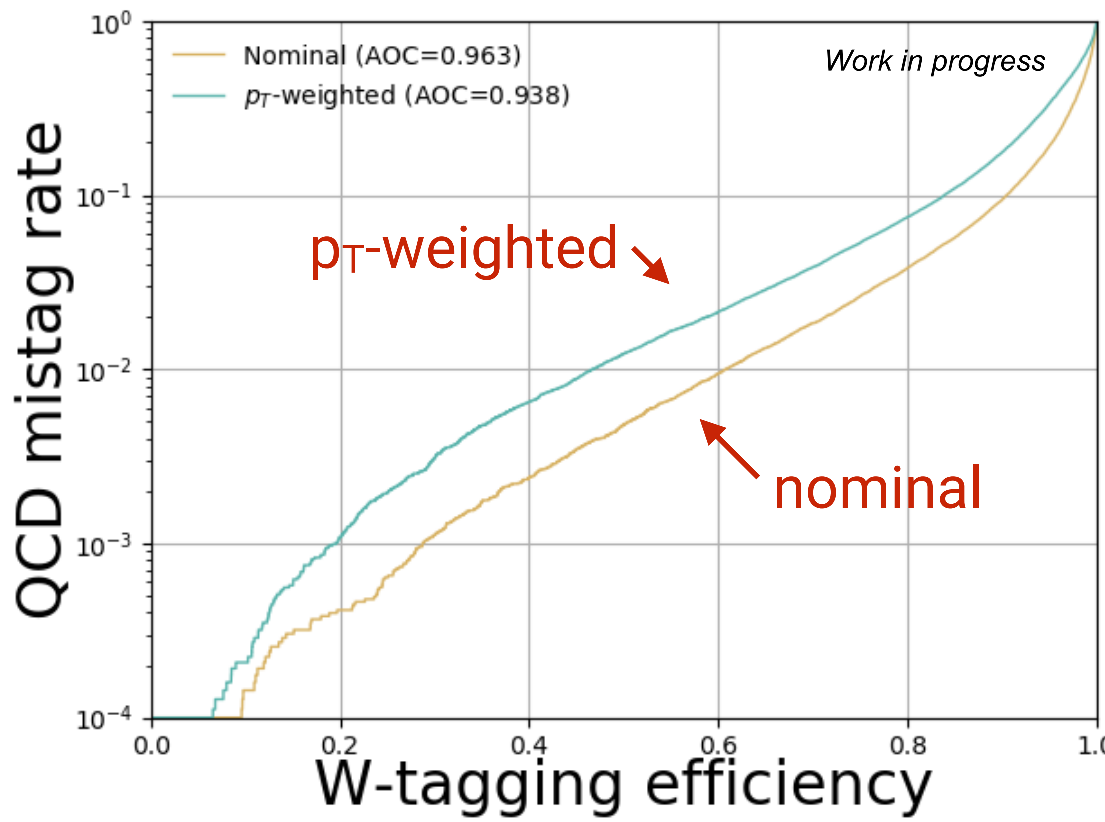
- passed as sample weights to training

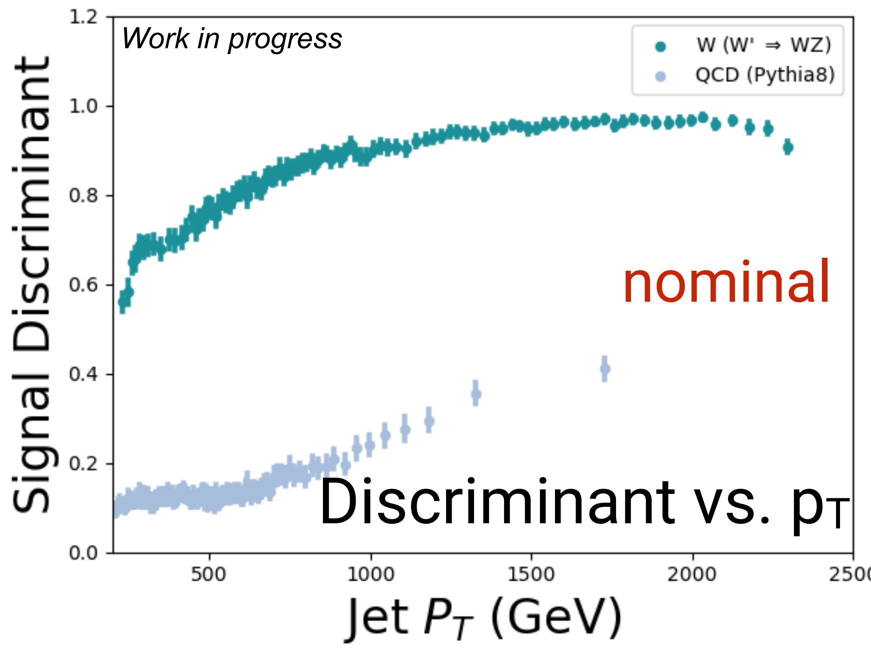


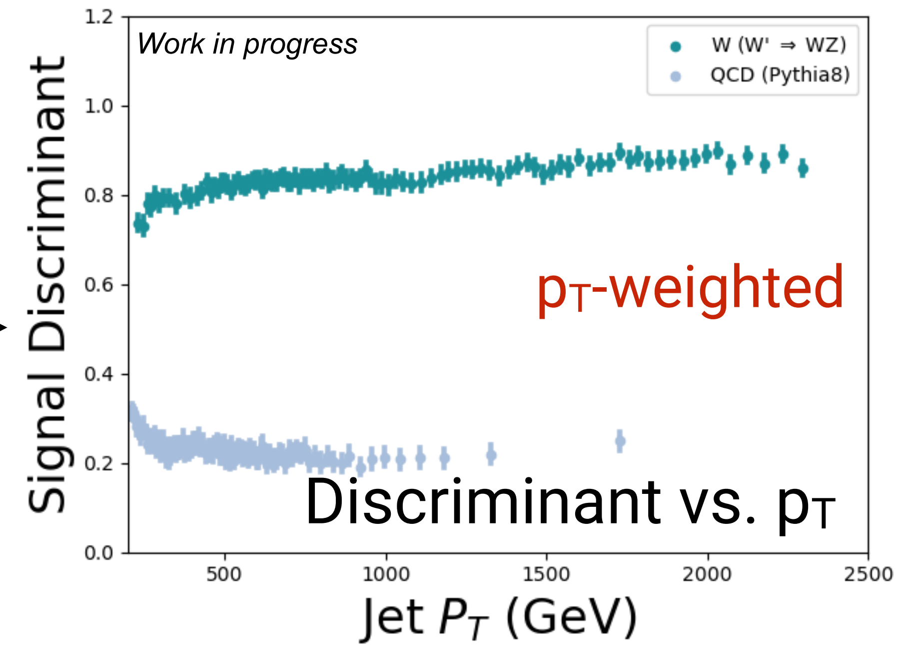
p_T dependence

Such strategies yields loss in overall performance, but reduced p_T-dependence

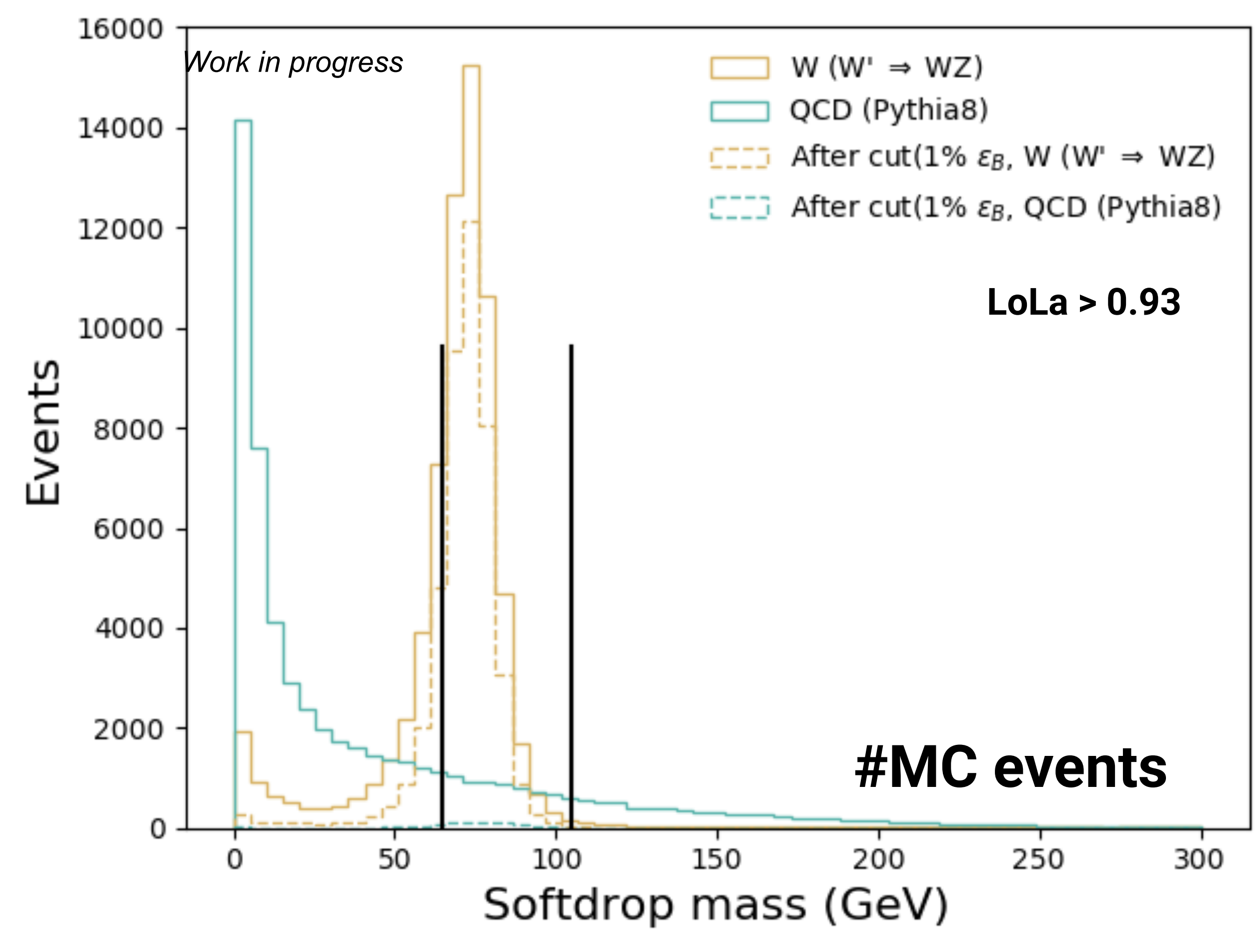
No “truth” for which solution is better before running full analysis including systematics for p_T-dependent tagging







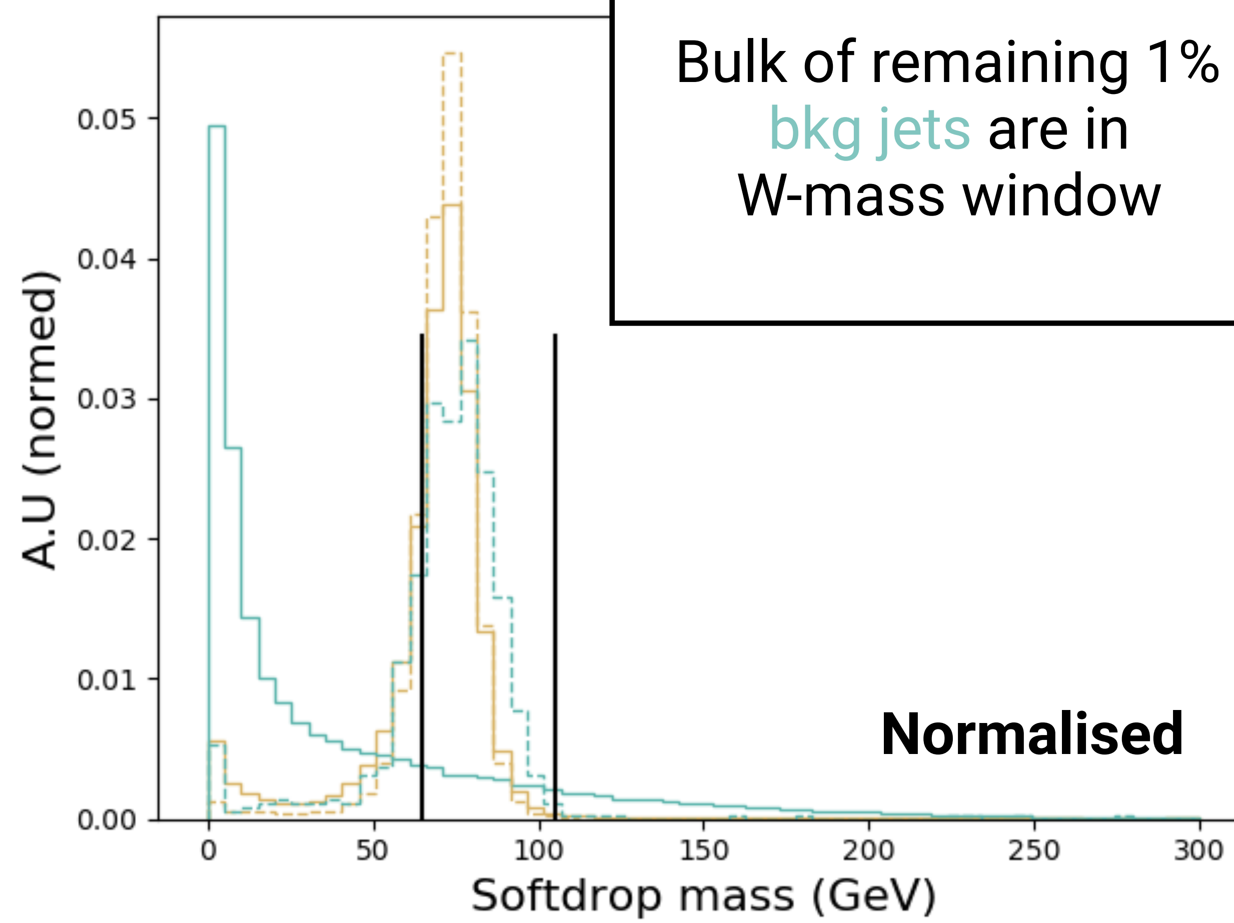
Mass sculpting



Mass sculpting

Universität Zürich[UZH]

Bulk of remaining 1% bkg jets are in W-mass window



Mass sculpting

I smart DNN will learn W-mass

- good! Clearly W-mass != q/g-jet mass

Unfortunately, we often estimate background in mass sidebands

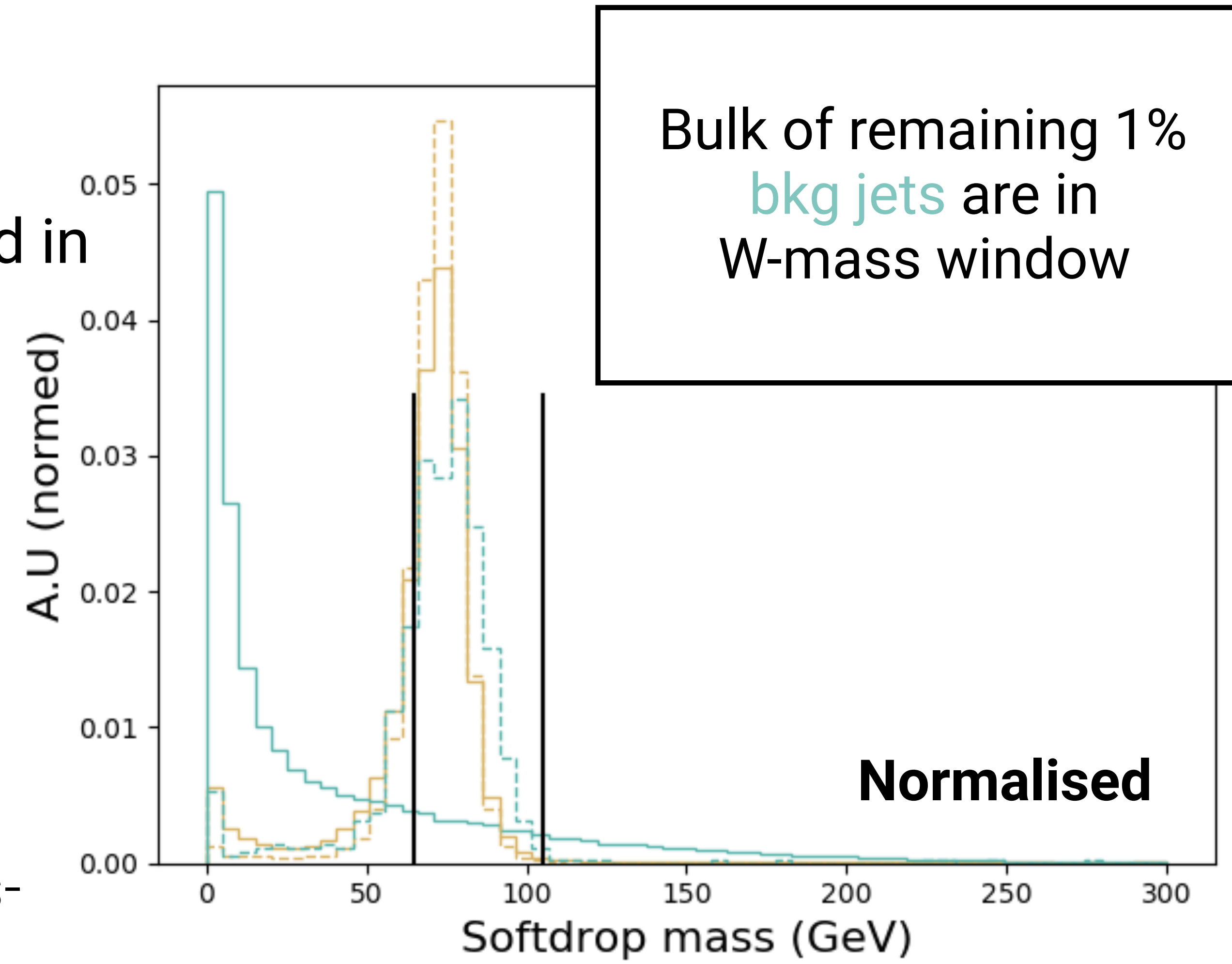
- bad! After cut on tagger, mass is sculpted making background difficult to constrain

Mass-dependence in itself not a problem, background rate uncertainties are

- trade-off between efficiency and (analysis-dependent) systematics.

Hot topic in ML: adversarial NNs that penalise loss if mass is learned (see C. Shimmin et. Al)

- loss in efficiency, gain in analysis sensitivity



Model grooming

Despite common beliefs, a DNN is NOT a black box

- series of multiplications/additions and pre-computed activation functions
- you can (and should) read out the weights of your model for each layer (or feature)*

Does network learn something (un)expected?

- with physics-based trainable weights like in LoLa, easier to disentangle

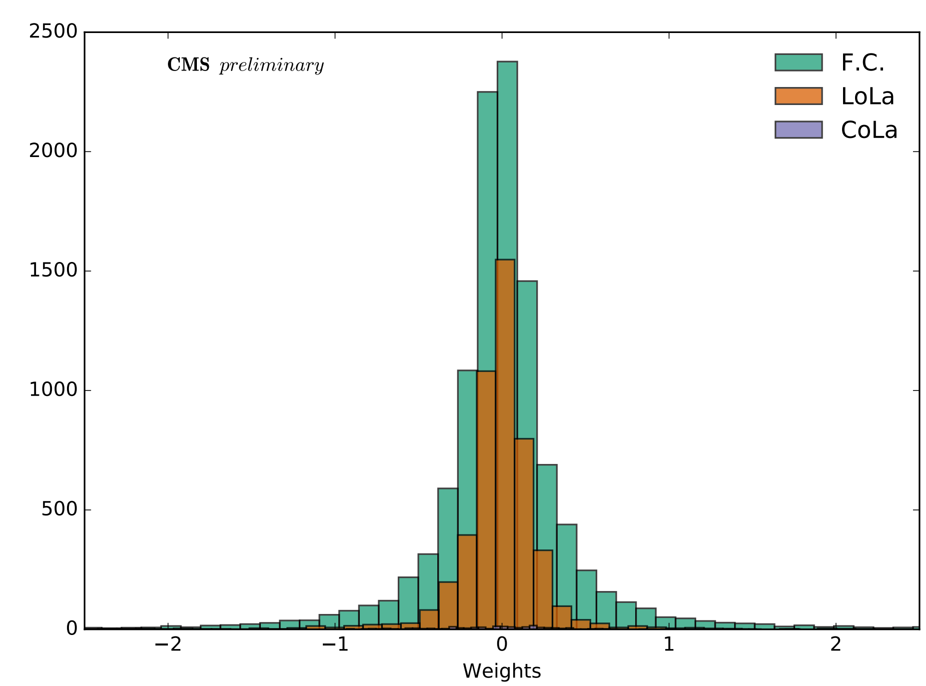
Also allows you to prune your DNN

- remove ~zero-weights from network. Reduces processing time with same performance

Input — Layer 1

$$\begin{bmatrix}x_1\\x_2\end{bmatrix} \rightarrow \sigma\left(\underbrace{\begin{bmatrix}W_{1,1} & W_{2,1}\\ W_{1,2} & W_{2,2}\end{bmatrix}}_{\text{trainable weights}}\begin{bmatrix}x_1\\x_2\end{bmatrix} + \underbrace{\begin{bmatrix}b_1\\b_2\end{bmatrix}}_{\text{trainable biases}}\right)$$

How are my features **x** weighted?



*model.layers[i].output
model.get_layer(layer_name).output

Summary and outlook

The idea behind LoLa is to give DNN the rules of Minkowski space, jet clustering and substructure and let it do the rest

analyse constituents directly with large set of trainable weights

For use in tagging, absolute performance is not a sufficient measure

- p_T-dependence + mass-sculpting resilience may be equally important depending on the analysis performed
- should strive to implement taggers in a full analysis chain before making final decisions (p_T-reweighting, mass penalising, etc.)

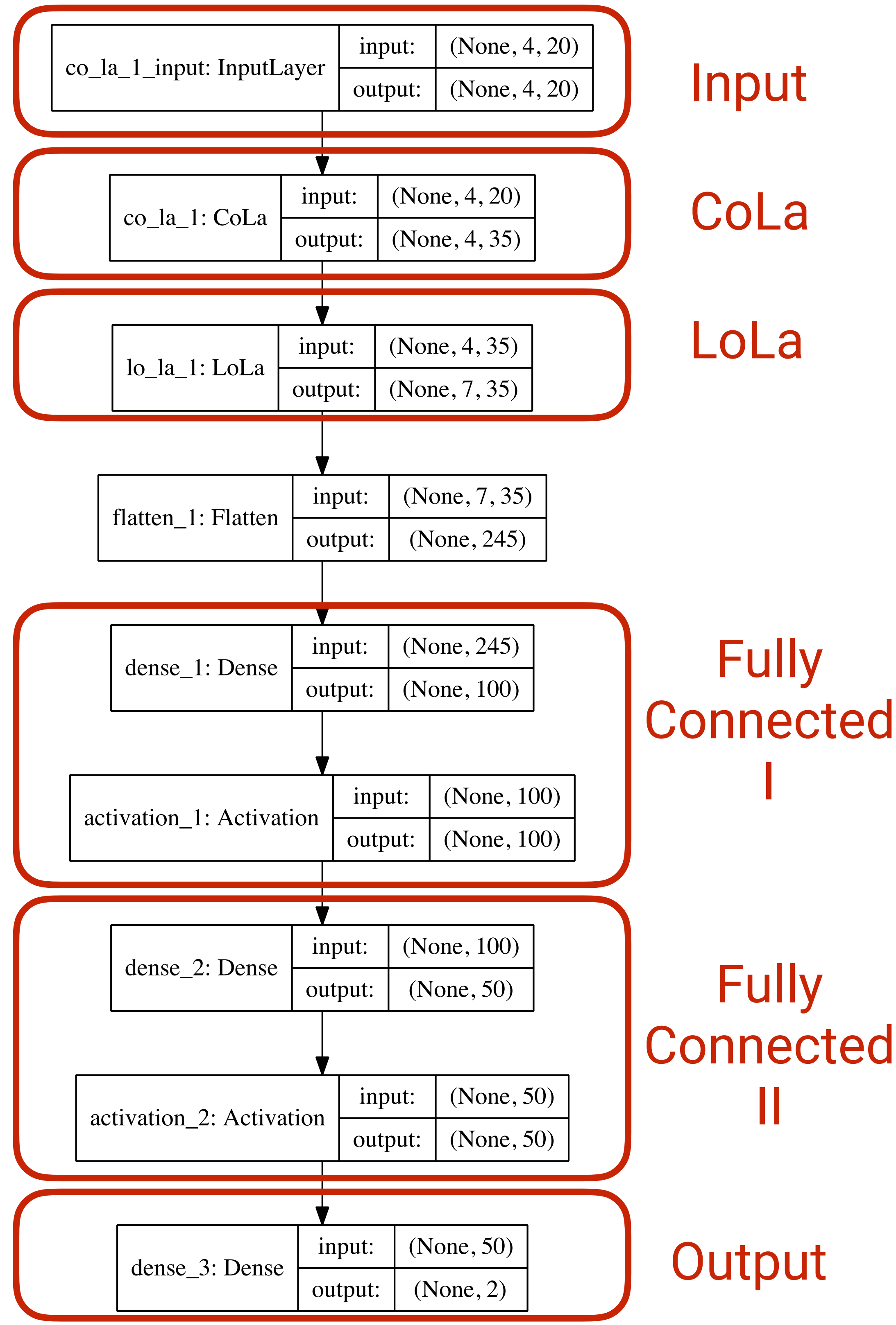
The question “What can we learn from the machine?” is getting more interesting than “What can we teach the machine?”

- by probing layer-wise LoLa output, hope to learn something new about substructure!

Backup

Model

- 4 layer DNN doing supervised learning with fixed-size input vectors
  - feed forward sequential network
  - Two novel layers (CoLa and LoLa) implementing Minkowski metric and “substructure” calculations (see later) and two fully connected layers
- Technicalities
  - Keras with Theano backend
  - Loss function: categorical crossentropy
  - ADAM optimiser (adapt learning rate of model parameters during training)
- Train 200k + Test 60k + Val 60k on AWS



Model summarised

Input:
4-vectors of N = 20 highest p_T jet constituents of AK8 jets

Combination layer(CoLa):
- Sum of all momenta
- Each original momentum
- 15 trainable weights C

Lorentz layer(LoLa):
Compute kinematics for CoLa output.
1+4 additional trainable weights

$$(k_{\mu,i}) = \begin{pmatrix} k_{0,1} & k_{0,2} & \cdots & k_{0,N} \\ k_{1,1} & k_{1,2} & \cdots & k_{1,N} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,N} \\ k_{3,1} & k_{3,2} & \cdots & k_{3,N} \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 & C_{1,N+2} & \cdots & C_{1,M} \\ 1 & 0 & 1 & & \vdots & C_{2,N+2} & \cdots & C_{2,M} \\ \vdots & \vdots & \vdots & \ddots & 0 & \vdots & & \vdots \\ 1 & 0 & 0 & \cdots & 1 & C_{N,N+2} & \cdots & C_{N,M} \end{pmatrix}$$

$$\tilde{k}_j \xrightarrow{\text{LoLa}} \hat{k}_j = \begin{pmatrix} m^2(\tilde{k}_j) \\ p_T(\tilde{k}_j) \\ \boxed{w_{jm}^{(E)}} E(\tilde{k}_m) \\ \boxed{w_{jm}^{(d)}} d_{jm}^2 \end{pmatrix}$$

$k_{\mu,i}$

(4 x 20)

$$k_{\mu,i} \xrightarrow{\text{CoLa}} \tilde{k}_{\mu,j} = k_{\mu,i}\, C_{ij}$$

(4 x 35)

↓

Summing and weighting all constituent should allow network to calculate subjet axes

$$\tilde{k}_j \xrightarrow{\text{LoLa}} \hat{k}_j$$

(7 x 35)

↓

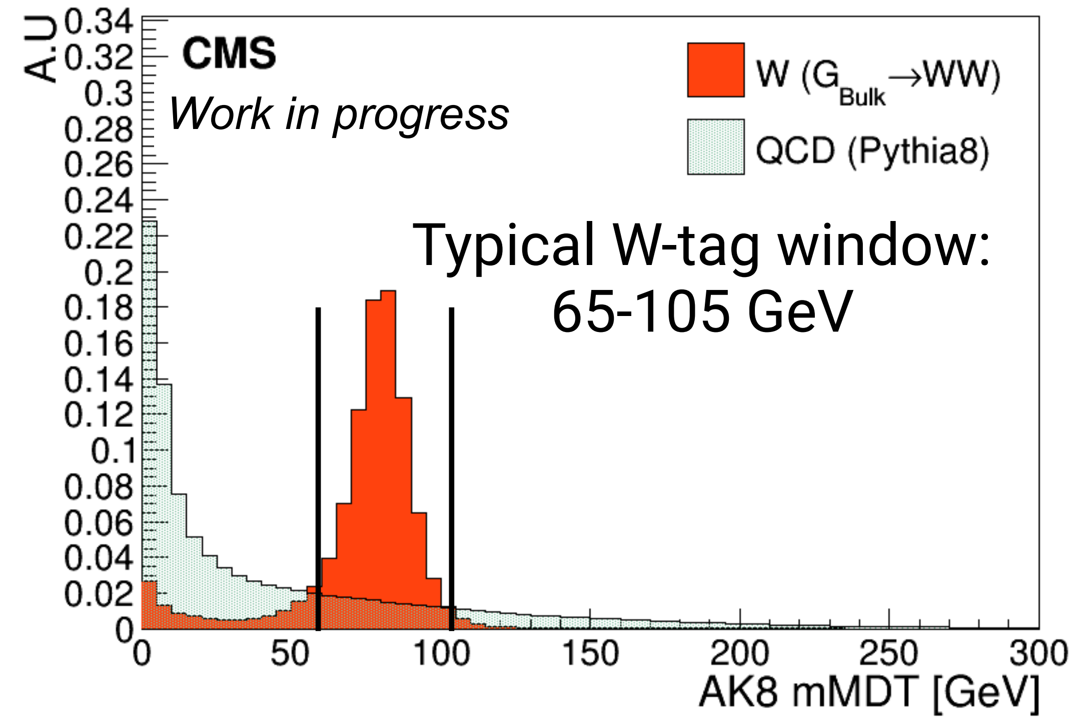
d^2_{jm} and m^2 use Minkowski distance

$$d_{jm}^2 = (\tilde{k_j} - \tilde{k_m})_\mu g^{\mu\nu} (\tilde{k_j} - \tilde{k_m})_\nu$$

The basic setup

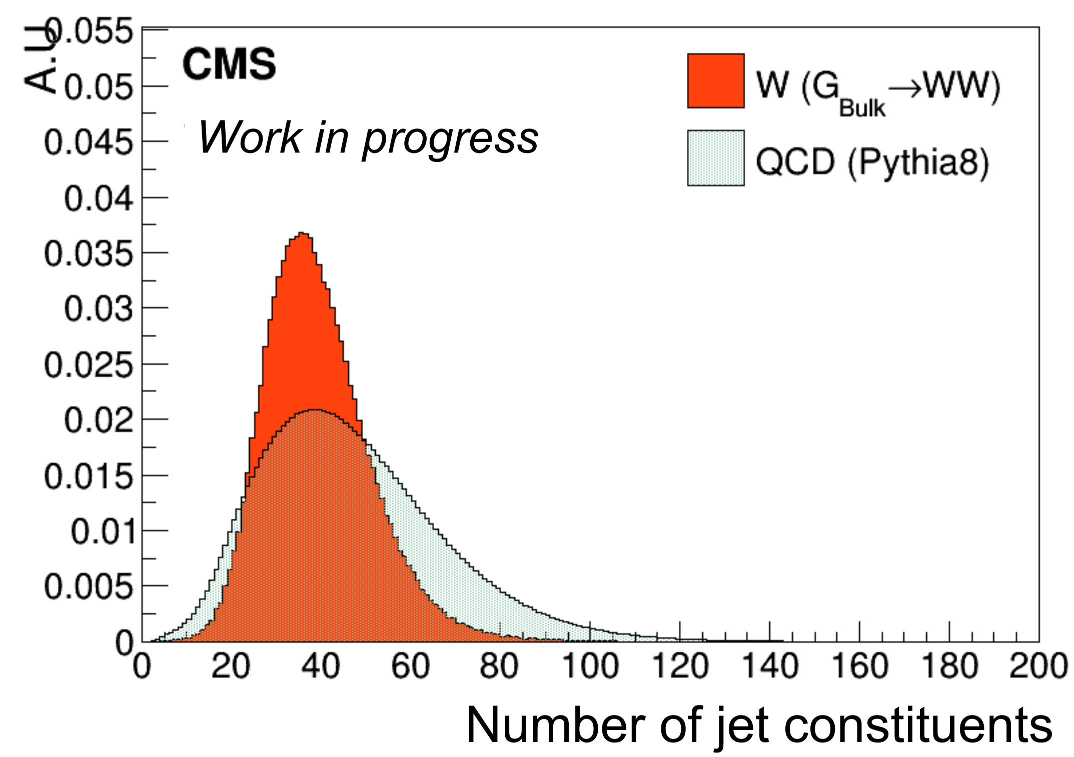
Signal

- 320k fully merged hadronic W-jets (AK8) from W'→WZ →4q ($M_{W'}$ = 0.6-4.5 TeV)
- why small training set?→ Do not mix signal samples until one is understood (can change with W polarisation etc.)



Background

- QCD Pythia8 non-W jets
- Danger: Jet substructure strongly depends on shower generators (different description of gluon radiation). Different QCD MC might yield different results



Disclaimer: The following contains student work in progress studies and not CMS approved results

What does LoLa learn?

Universität Zürich[UZH]

<u>Input:</u>
4-vectors of N = 20 highest p_T jet constituents

<u>Combination layer(CoLa):</u>
- Sum of all momenta

<u>Lorentz layer(LoLa):</u>
Compute kinematics for CoLa output.

E_i ... E_N
p_x
p_y
p_z

$$(k_{\mu,i}) = \begin{pmatrix} k_{0,1} & k_{0,2} & \cdots & k_{0,N} \\ k_{1,1} & k_{1,2} & \cdots & k_{1,N} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,N} \\ k_{3,1} & k_{3,2} & \cdots & k_{3,N} \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 & C_{1,N+2} & \cdots & C_{1,M} \\ 1 & 0 & 1 & & \vdots & C_{2,N+2} & \cdots & C_{2,M} \\ \vdots & \vdots & \vdots & \ddots & 0 & \vdots & & \vdots \\ 1 & 0 & 0 & \cdots & 1 & C_{N,N+2} & \cdots & C_{N,M} \end{pmatrix}$$

$$\tilde{k}_j \xrightarrow{\text{LoLa}} \hat{k}_j = \begin{pmatrix} m^2(\tilde{k}_j) \\ p_T(\tilde{k}_j) \\ w_{jm}^{(E)} E(\tilde{k}_m) \\ w_{jm}^{(d)} d_{jm}^2 \end{pmatrix}$$

HOW DOES THIS BIAS MASS AND P_T?!

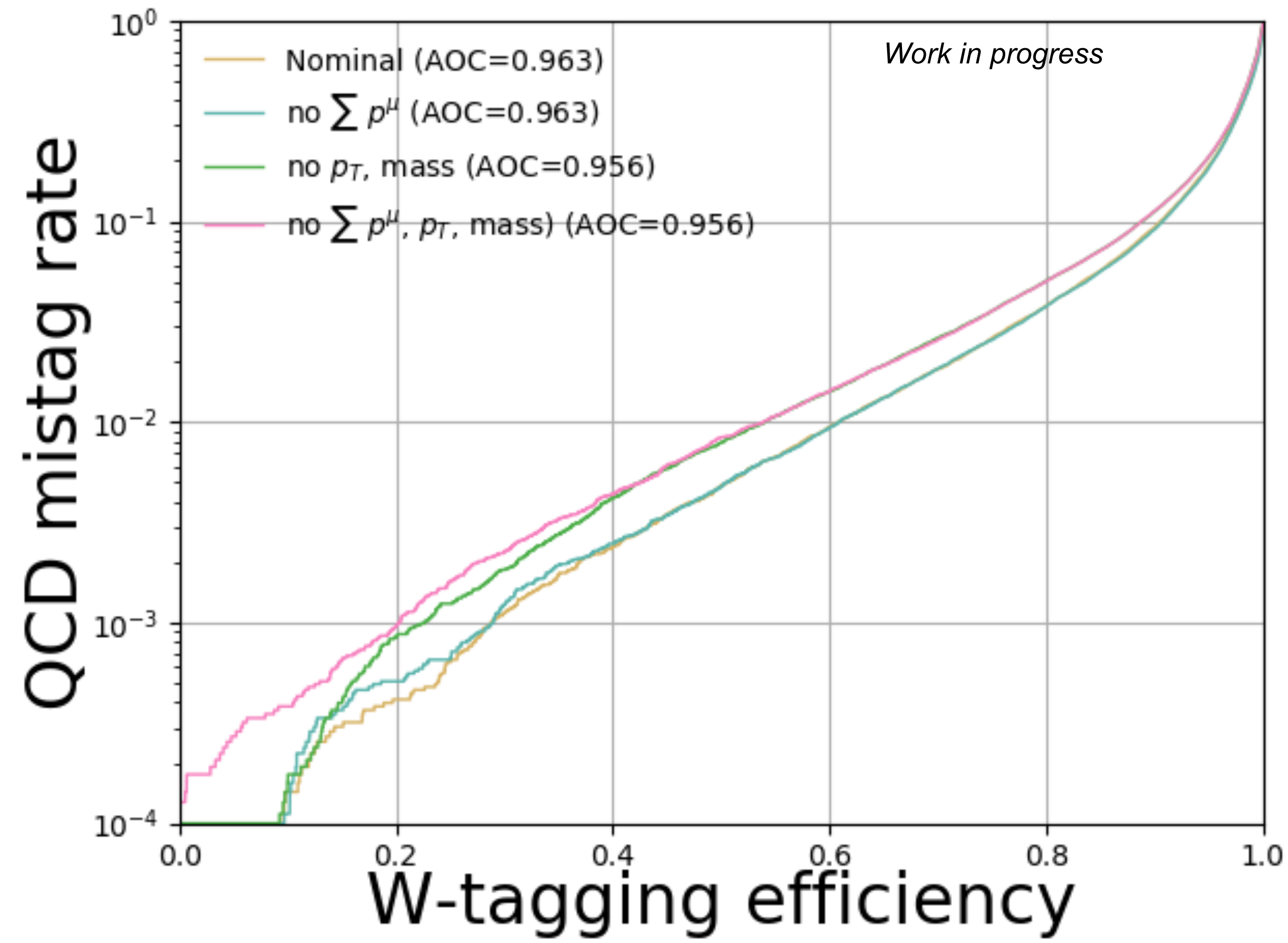
Summing and weighting all constituent should allow network to calculate subjet axes

d^2_{jm} and m^2 use Minkowski distance

$$d_{jm}^2 = (\tilde{k}_j - \tilde{k}_m)_\mu g^{\mu\nu} (\tilde{k}_j - \tilde{k}_m)_\nu$$

What does LoLa learn?

- Compare nominal training to training after removing variables sensitive to mass and p_T
- Remove CoLa column that passes sum of all 4-momentum (“jet” 4-vector)
  - not much impact on overall performance
  - not much information taken from LoLa “n-subjettiness”
- Remove Lola mass and p_T variables reduce performance significantly
  - worst when removing jet 4-vector, mass and p_T



LoLas future

Study LoLa output column-wise to understand what LoLa is learning

- picking up substructure or not?

Study discriminating power for longitudinally versus transversally polarised W bosons
→W_T vs W_L tagger?

As part of fun

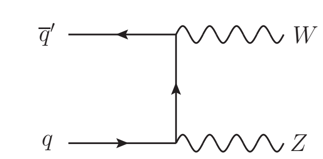
- train LoLa to do Pythia QCD vs. Herwig to understand where shower differences arise?

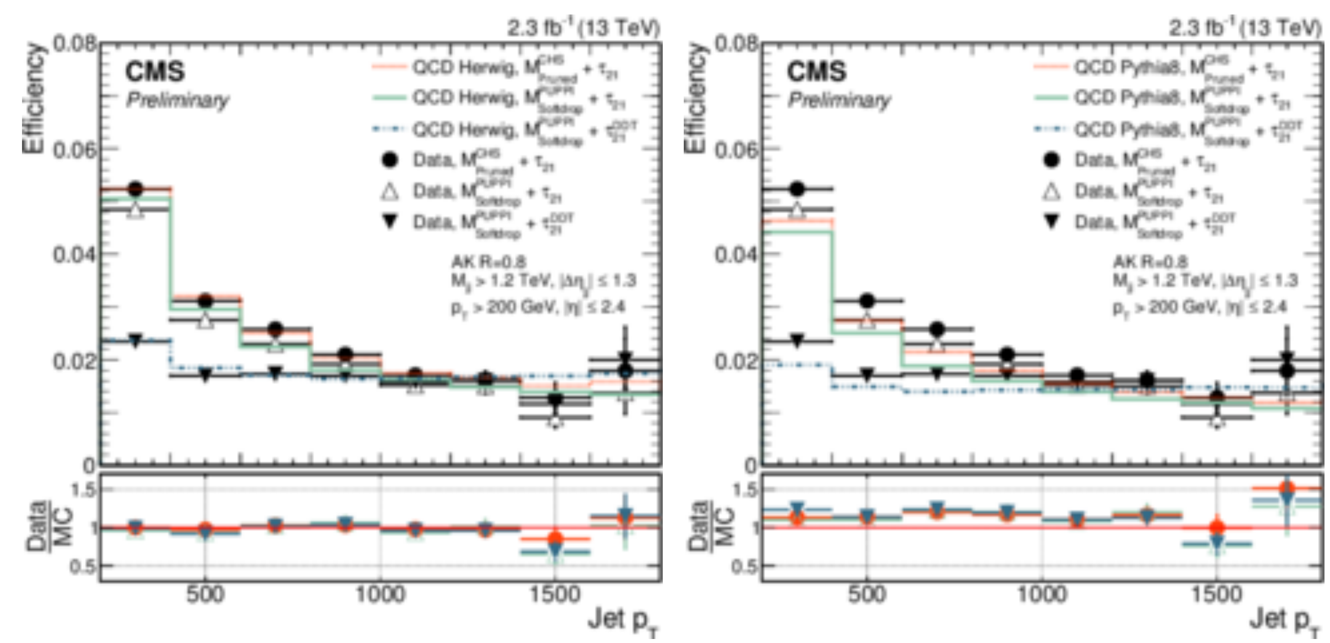
Energy enhanced new-physics effects in **longitudinal channel**

$$\frac{\mathcal{A}_{LL}^{\mathrm{SM\,+\,BSM}}(q\bar{q}\to WZ)}{\mathcal{A}_{LL}^{\mathrm{SM}}(q\bar{q}\to WZ)} \sim 1 + a_q^{(3)} E^2$$

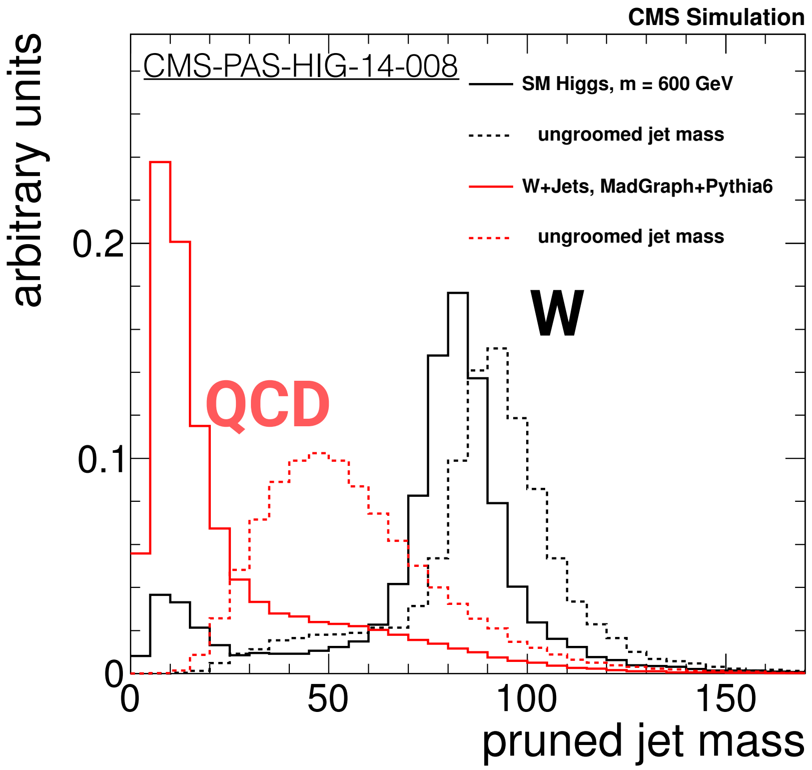
… but **transverse** channels **dominate** the SM cross section

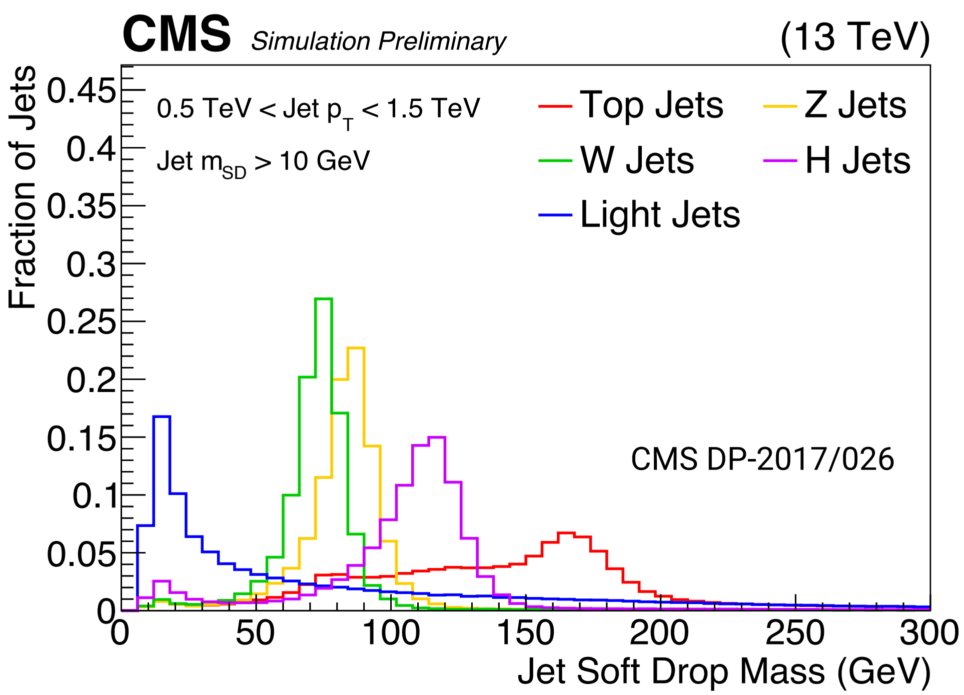
large cross section
due to t-channel singularity
(only there for transverse)





What's the mass of my object?





Pruning

arxiv:0912.0033

- Remove soft, wide-angle radiation
  - recluster jet with C-A, remove recombination if

$$\frac{\min(p_{T1}, p_{T2})}{p_{T1}+p_{T2}} < \boxed{0.1} \text{ and } \Delta_{1,2} > \boxed{0.5} \cdot \frac{2m}{p_T}$$

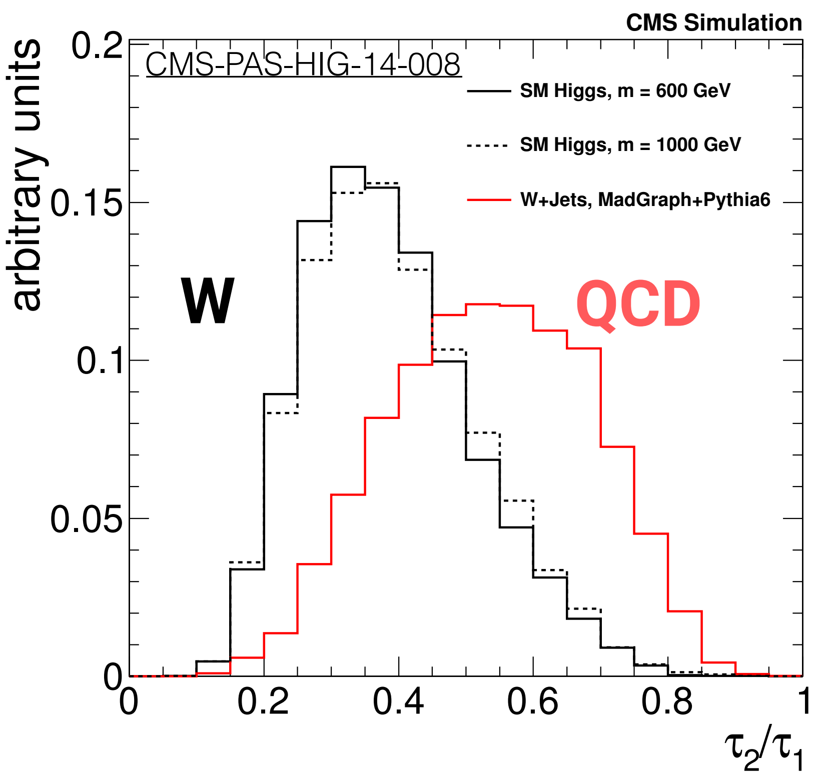
Modified Mass Drop Tagger (aka Softdrop, β=0)

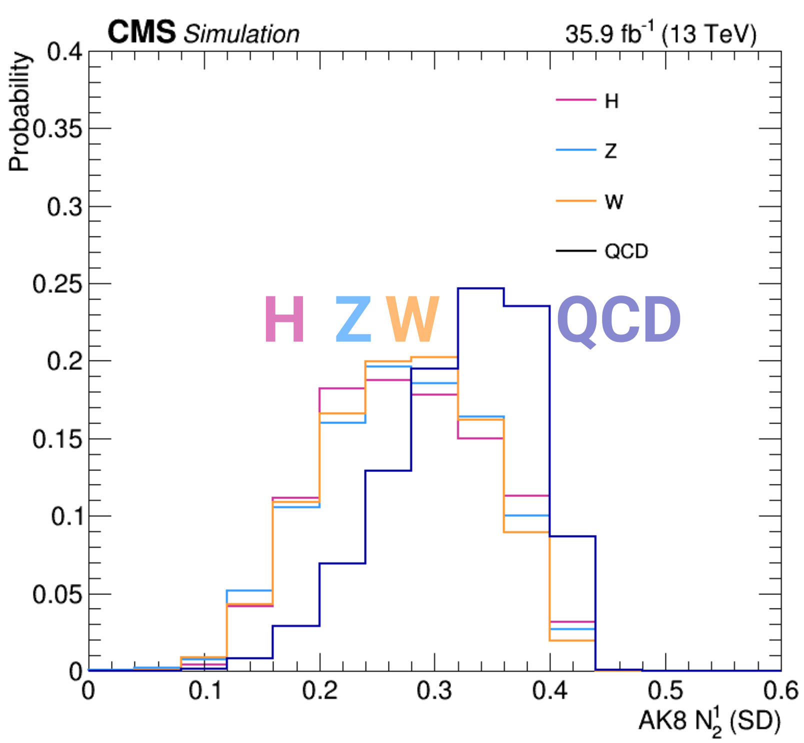
arxiv:1402.2657
arXiv:1307.0007

- Remove all soft emission
  - decluster with C-A, remove recombination if

$$\frac{\min(p_{T1}, p_{T2})}{p_{T1}+p_{T2}} < \boxed{0.1}$$

Can I peak inside the jet?





N-subjettiness τ_{21}

arxiv:1011.2268

- How compatible jet is with having N axis
  - p_T-weighted distance between constituents and N axes
  - small τ_2/τ_1: more two- than one-prong like

Ratio of Energy Correlation Functions N_2

arxiv:1305.0007

- Sensitive to N-particle correlations within jet
  - like τ_2/τ_1, but avoid definition of subjet axes
  - less dependent on p_T and p_T^2/m^2

$$N_2 = \frac{{}_2e_3}{({}_1e_2)^2} \qquad e_2 = \sum_{1\le i<j\le n_J} z_i z_j \, \theta_{ij}$$

pairwise angles

between n constituents

Mass: Softdrop

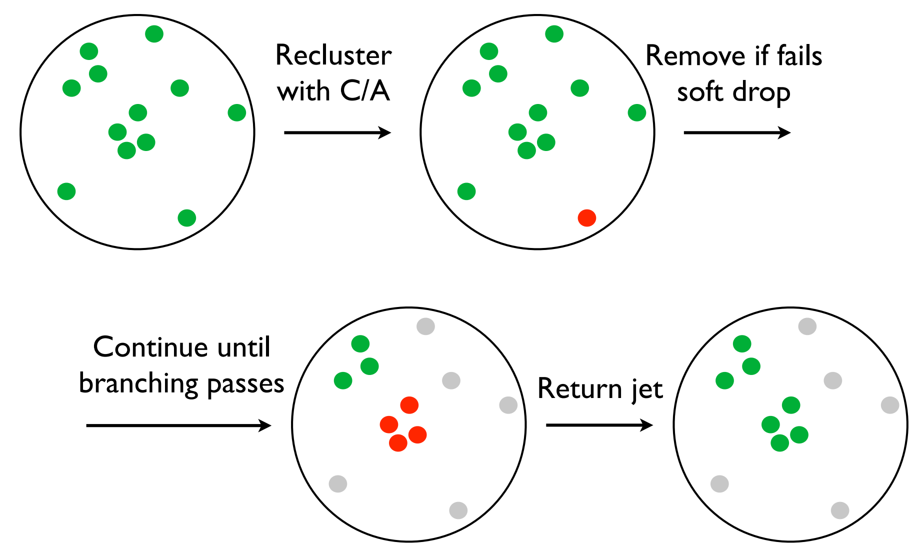
- Reclusterjet with C-A algorithm. Then decluster and check if subjets pass

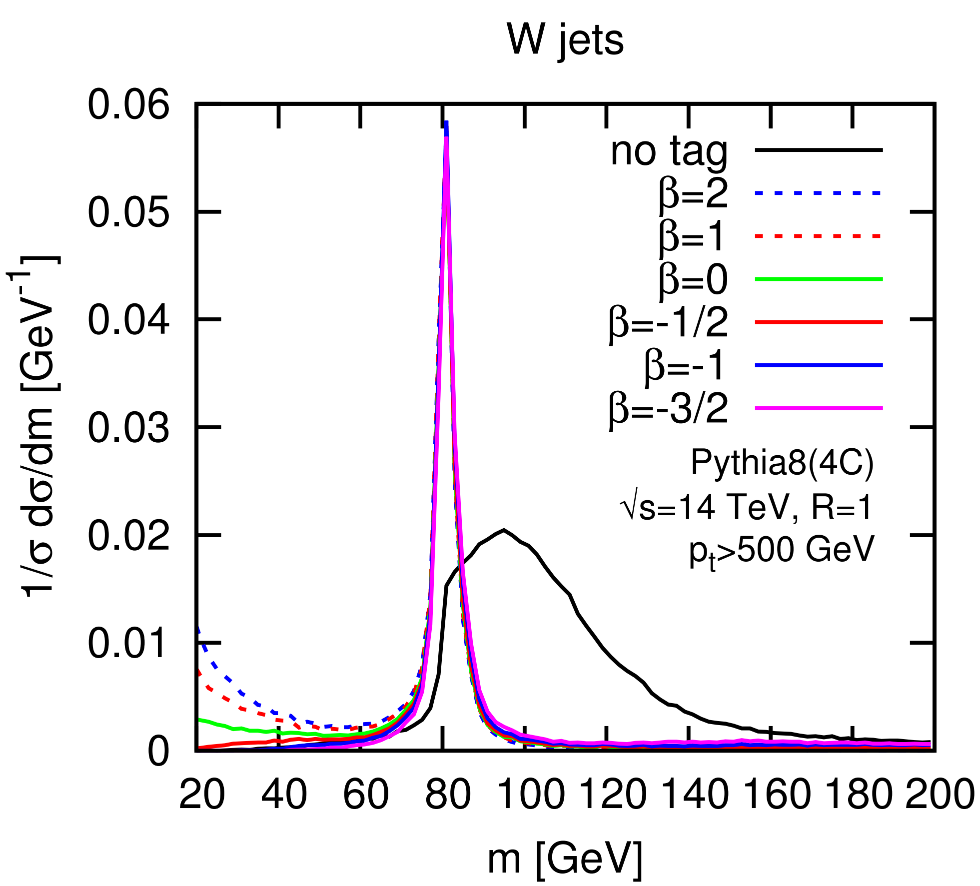
$$\frac{\min(p_{T1}, p_{T2})}{p_{T1}+p_{T2}} > z_{\text{cut}} \left(\frac{\Delta R_{12}}{R_0}\right)^{\beta}$$

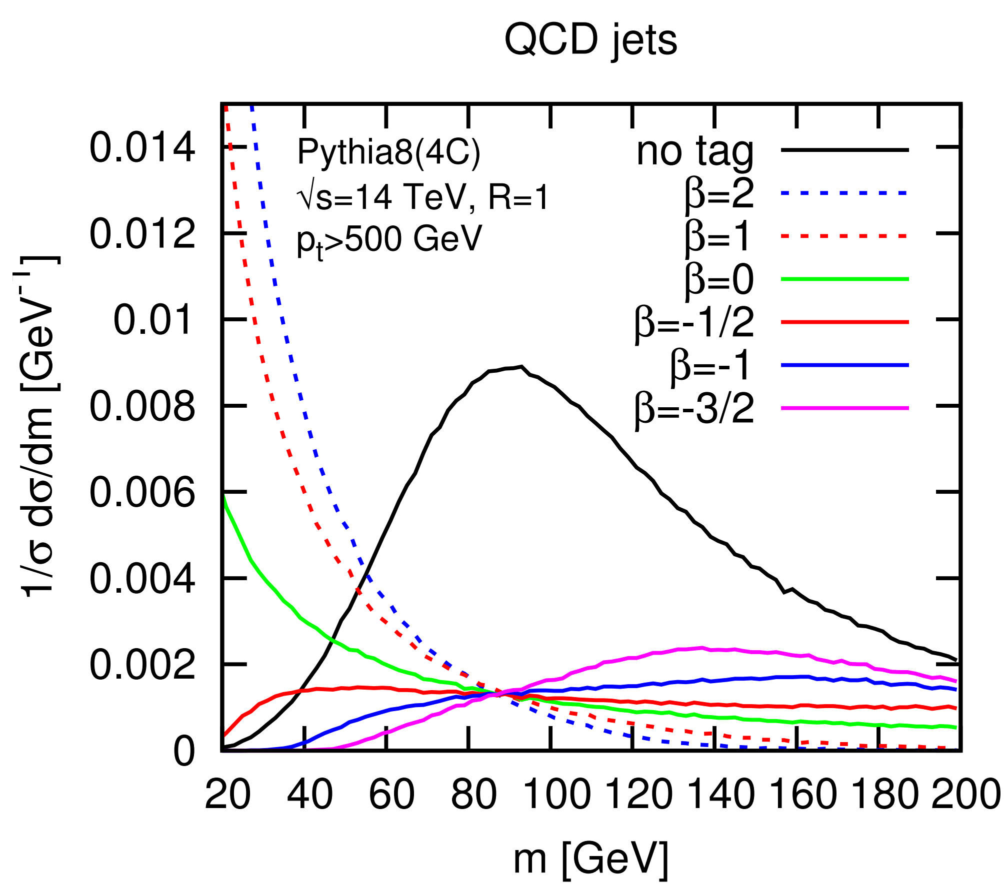
Soft threshold

Angular exponent

- in CMS β=0, z_{cut} = 0.1 (modified Mass Drop)







Tuned parameters: z_{cut} and β

β = ∞
no grooming

β > 0
soft, wide angle removed
some soft-collinear removed

β = 0
all soft emissions removed
modified Mass Drop limit

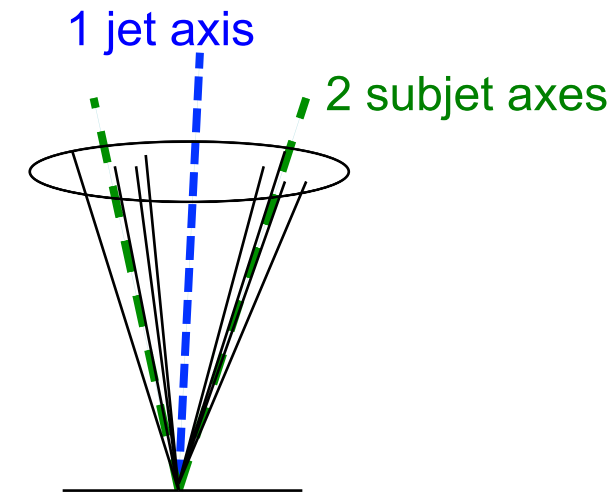
CMS default

β < 0
all soft and collinear emissions removed

Universität Zürich[UZH]

Substructure: N-subjettiness

- p_T-weighted sum over all constituents of the distance w.r.t the closest of N axes in a jet

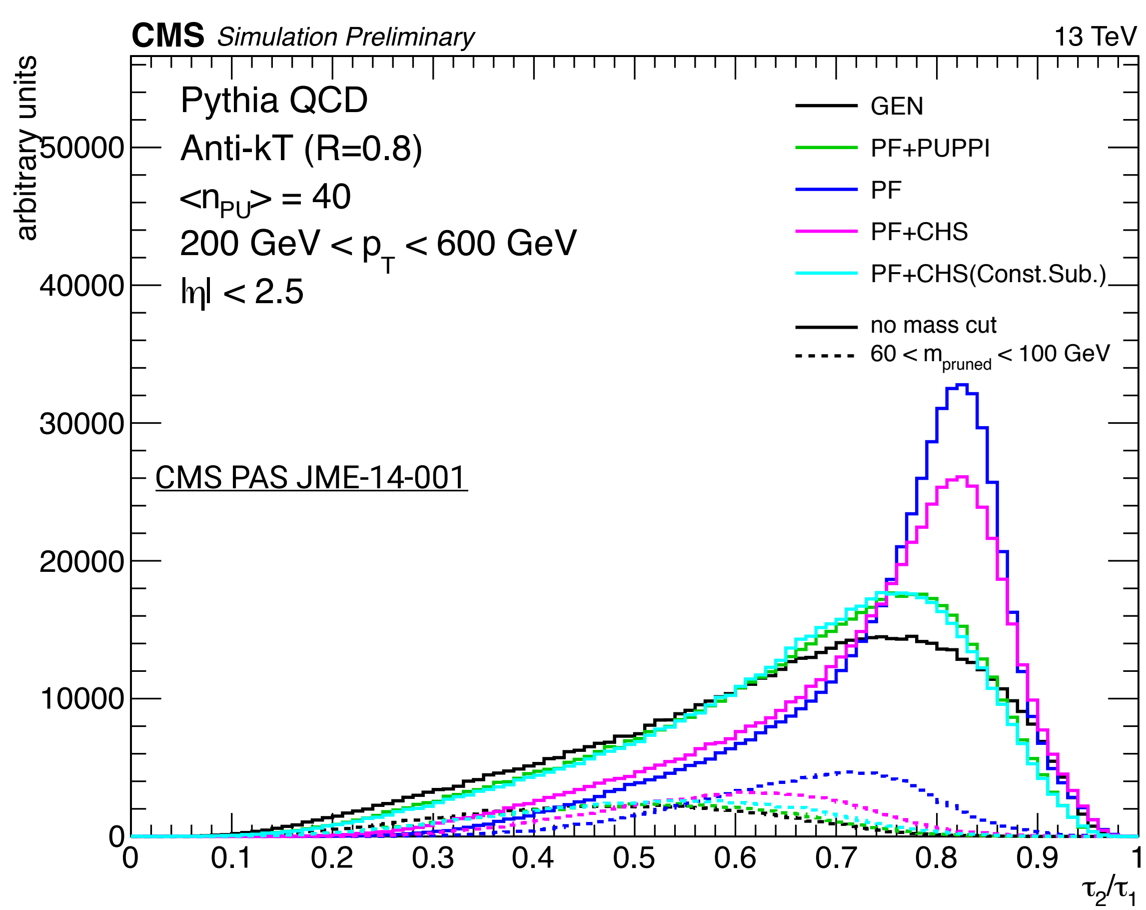


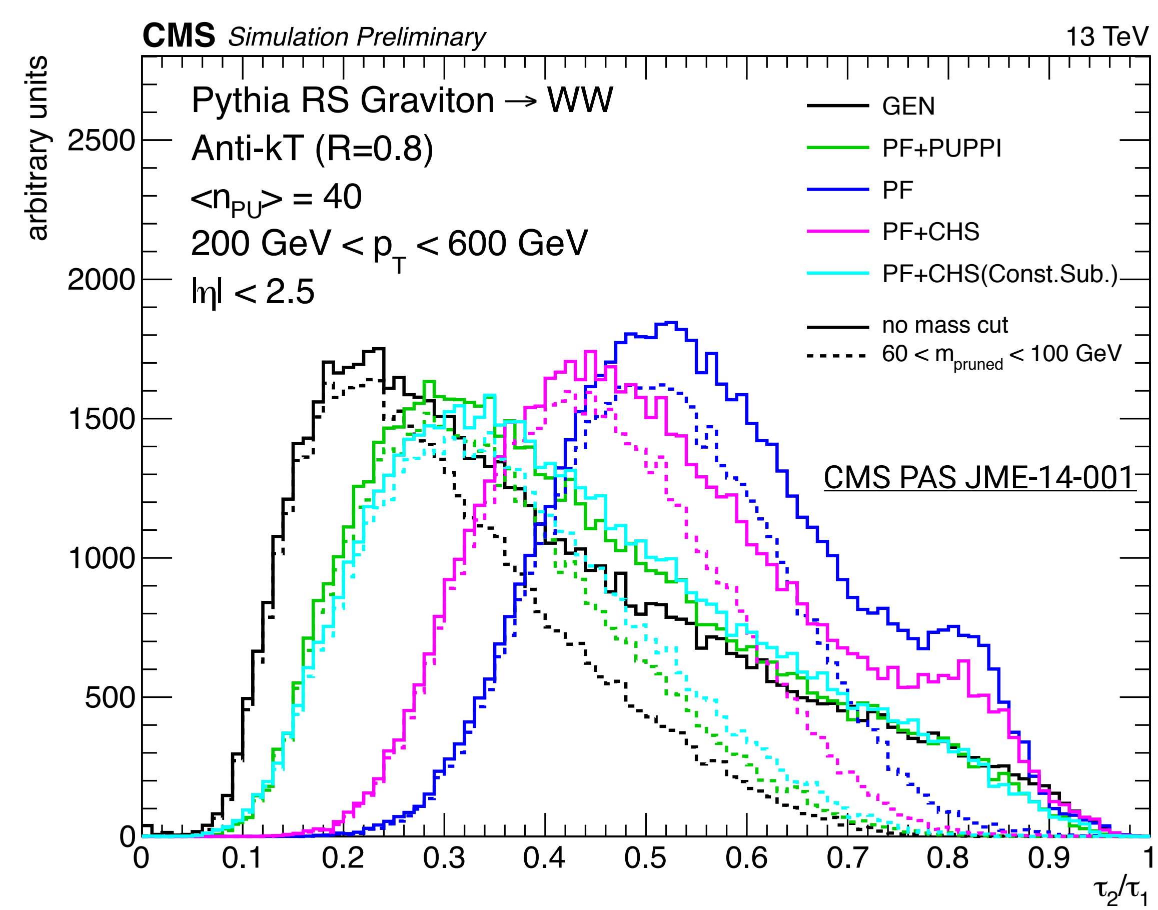
$$\tau_N = \frac{1}{d_0} \sum_k p_{T,k} \min\left((\Delta R_{1,k}), (\Delta R_{2,k}) \ldots (\Delta R_{N,k})\right)$$

Distance between momentum of constituent k w.r.t momentum of rest-frame subjet N

Each constituent assigned to nearest subjet!

  - axis obtained by undoing last (N-1) steps of clustering algorithm
  - small τ_N indicates compatibility with N axes hypothesis





Energy correlation functions (EFCs)

- Signal jets satisfy the inequality ${}_2e_3 << (e_2)^2$, explaining the definition of the N2 observable

$$N_2^\beta = \frac{{}_2e_3^\beta}{({}_1e_2^\beta)^2}$$

\# particles

\# angles

- Less discriminating power after grooming applied

$${}_1e_2^1 = \sum_{1\le i<j\le n_J} z_i z_j \Delta R_{ij}$$

$${}_2e_3^1 = \sum_{1\le i<j<k\le n_J} z_i z_j z_k \min\{\Delta R_{ij}\Delta R_{ik}, \Delta R_{ij}\Delta R_{jk}, \Delta R_{ik}\Delta R_{jk}\}$$