Deep learning with tops

or

How I learned to stop worrying and love neural networks

Michael Russell

in collaboration with Gregor Kasieczka,
Tilman Plehn & Torben Schell

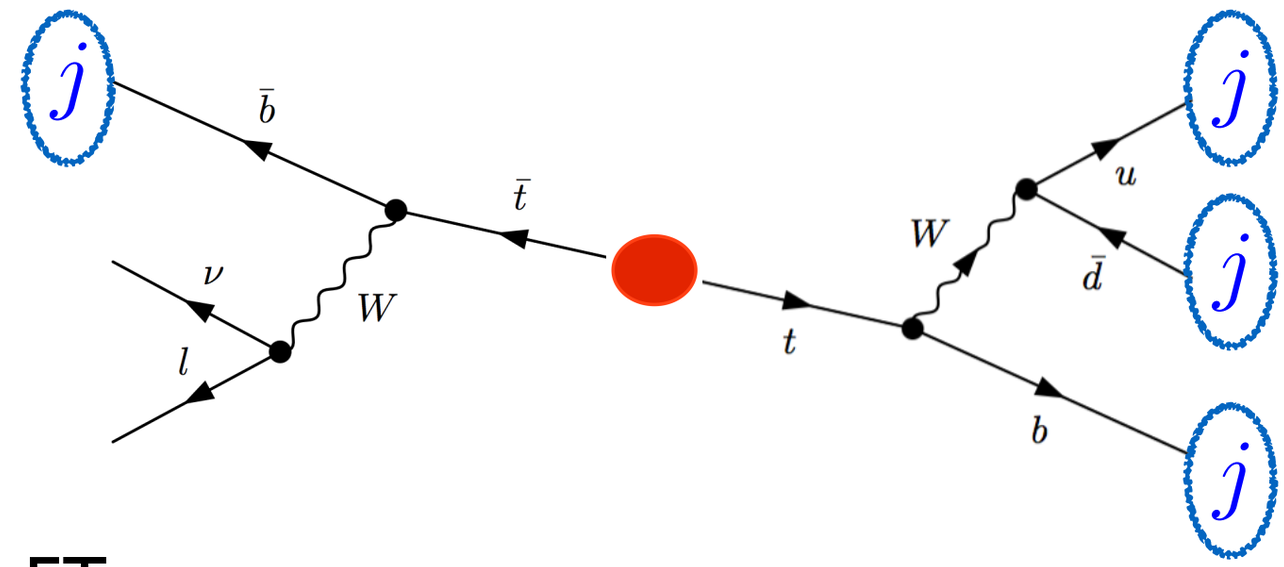
University of Glasgow

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

MCnet meeting, CERN, November 2016

Why tag tops?

Old-fashioned top reconstruction



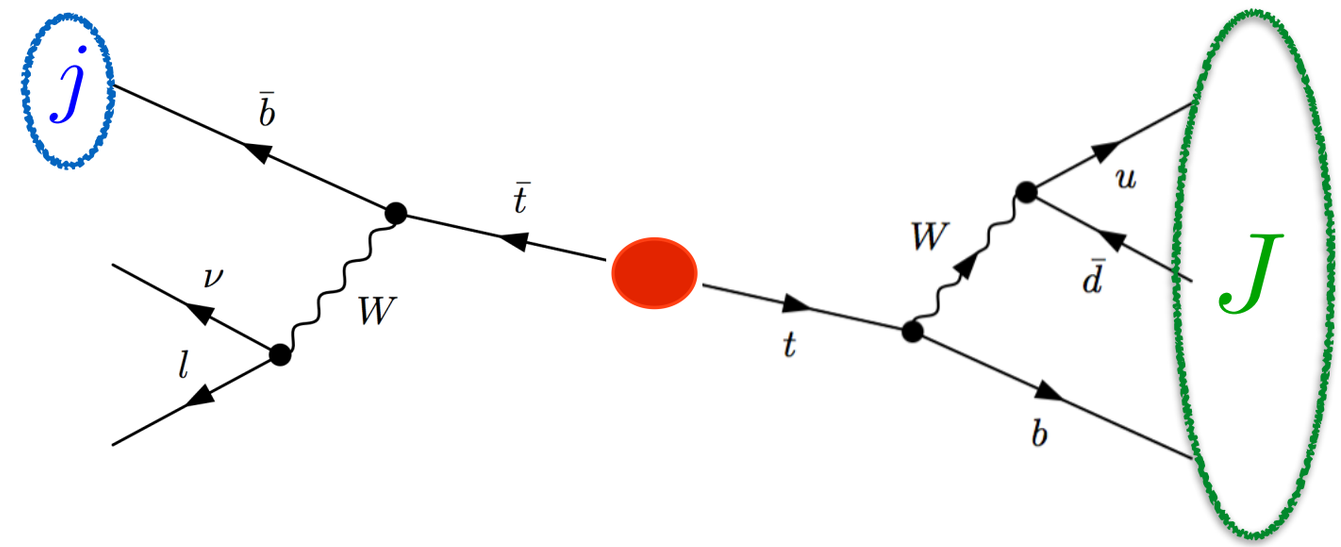
- Look for 4 light jets (2 b-tags)
- Look for isolated lepton, missing ET
- Reconstruct masses

BUT

- ~15% of total cross-section from pT > 200 GeV
- Region of interest for NP searches
- Standard reconstruction methods falter here

Why tag tops?

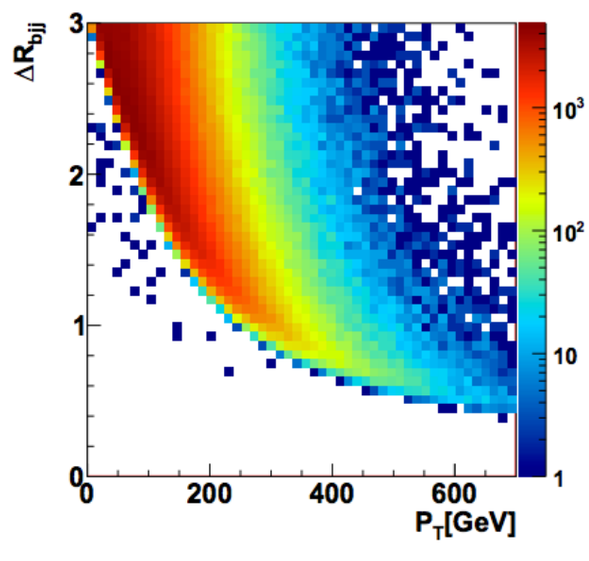
Modern top reconstruction



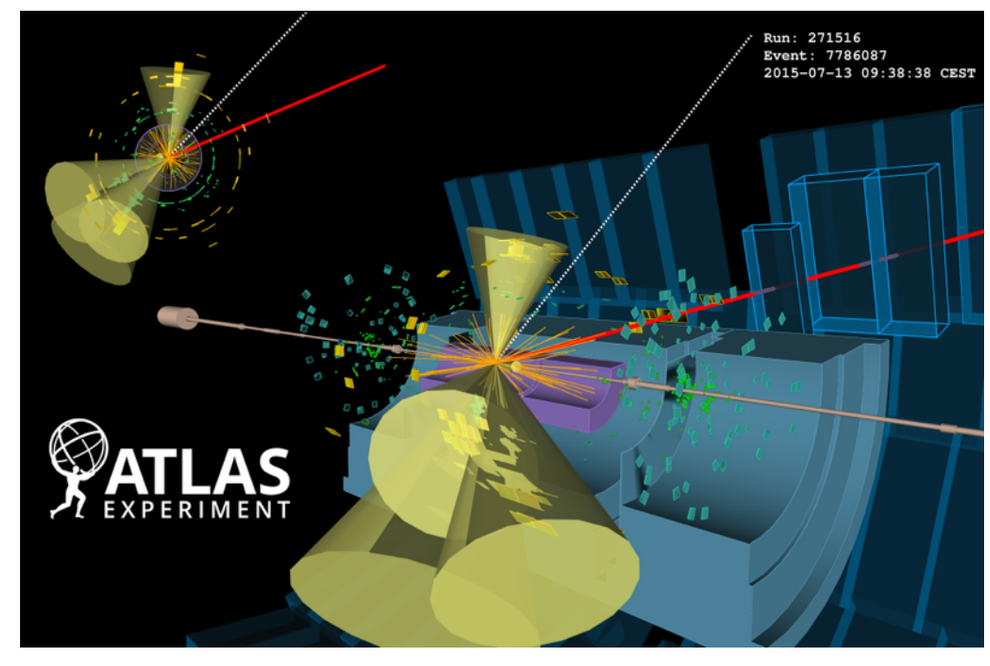
- Look for (tag) a FAT jet
- Look for isolated lepton, missing ET
- Reconstruct masses

How fat is fat?

$$\Delta R \sim \frac{2m}{p_T}$$



1006.2833



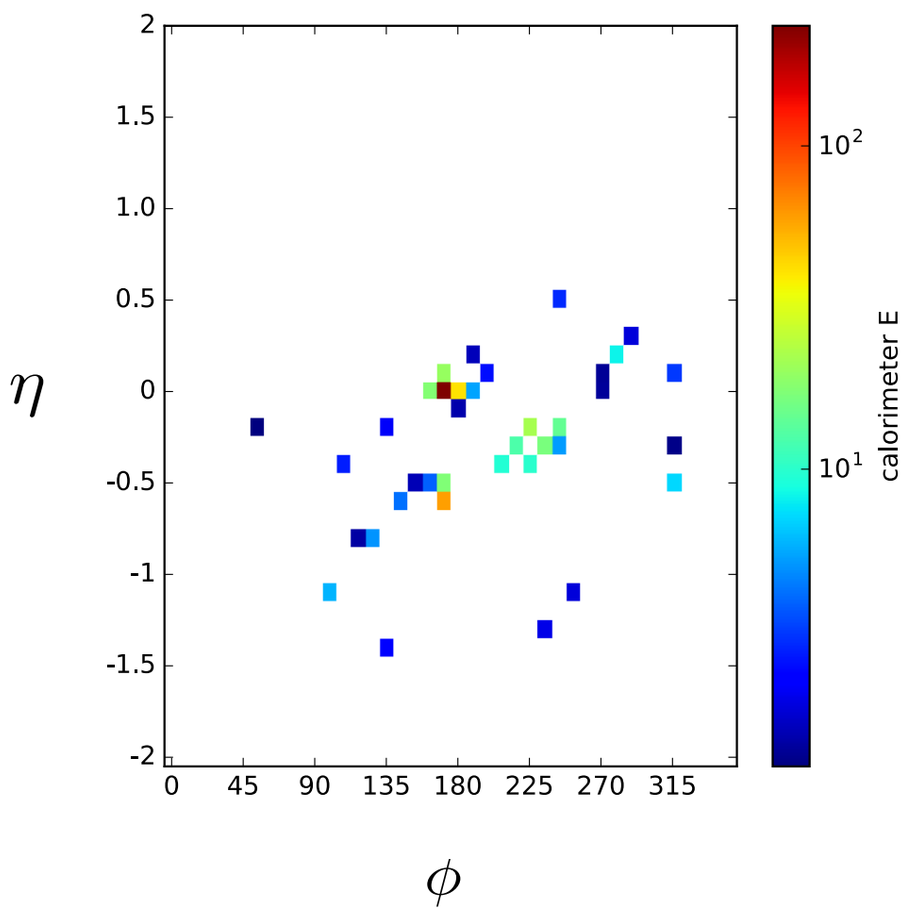
An example: HEPTopTagger

1006.2833

- Take a Cambridge-Aachen (R=1.5,pT > 200 GeV) fat jet
- Undo last step of clustering; apply mass-drop max(m1,m2) < mu*mJ
- If not fulfilled, keep heavier of 1 and 2. Otherwise keep both.
- Keep applying mass-drop until all subjets have mass < 30 GeV
- Filter subjets (combine into triplets)
- Top mass window requirement on subjets
- Re-cluster subjets with C-A, keep 5 leading-pT subjets
- W-mass requirement on subjets, if passed, jet is a top candidate

Missing information?

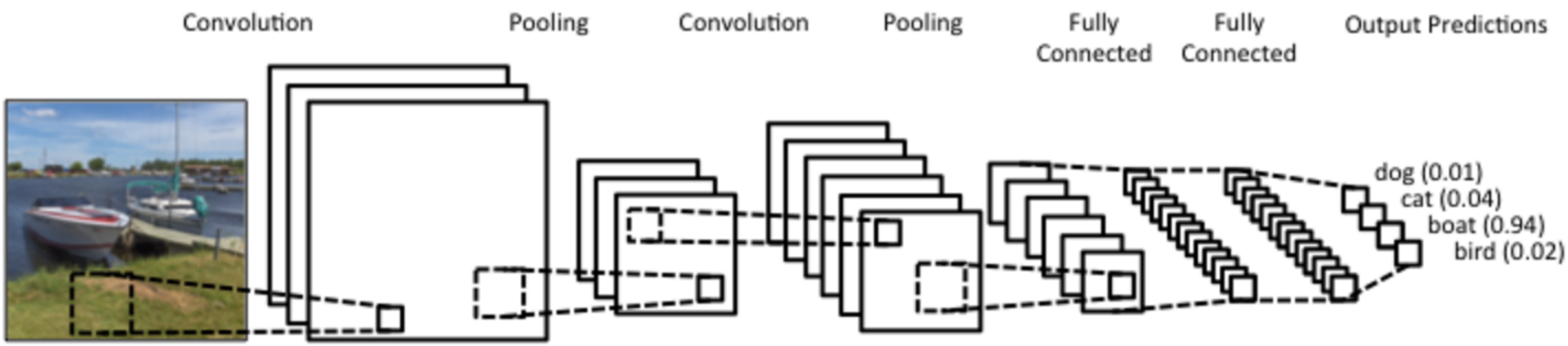
- Taggers interested in hard substructures of the event
- Most taggers end up with a handful of subjets to analyse
- On a grid of eta-phi ~ 0.1 by 0.1, much more activity
- Do we lose information this way?



Every pixel tells a story

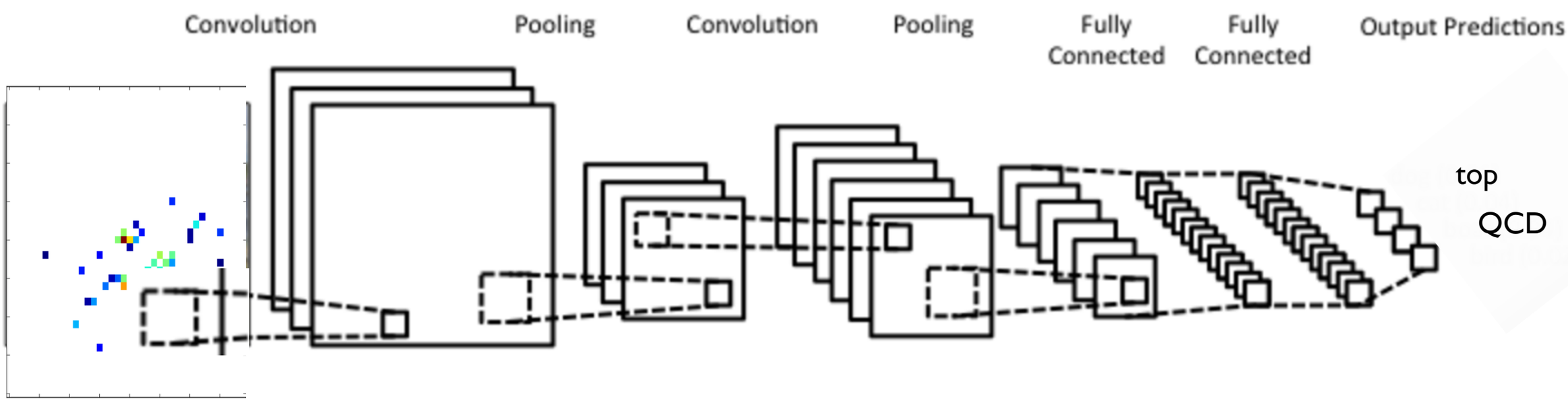
How can we keep track of ALL the features of a jet?

Convolutional neural networks



sauce: http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

The jet image



Put calorimeter on grid of 0.1 by 0.1

see also 1407.5675
1511.05190

fill pixels with jet constituents

calorimeter energy specify pixel density

Analysis setup

Signal: Top pair, all-hadronic

Background: QCD dijets

PYTHIA8 generation
Delphes detector simulation

Cluster into fatjets

C-A $\Delta R = 1.5$
$350\ \mathrm{GeV} < p_{T,J} < 450\ \mathrm{GeV}$
$|\eta_J| < 1.0$

Store additional kinematic information of jets

$$m_J, m_{J[filt]}, m_{J[sd]}, \tau_{32}, \tau_{32}^{sd}, \tau_{32}^{filt}$$

convert to jet image

feed to neural network

Preprocessing

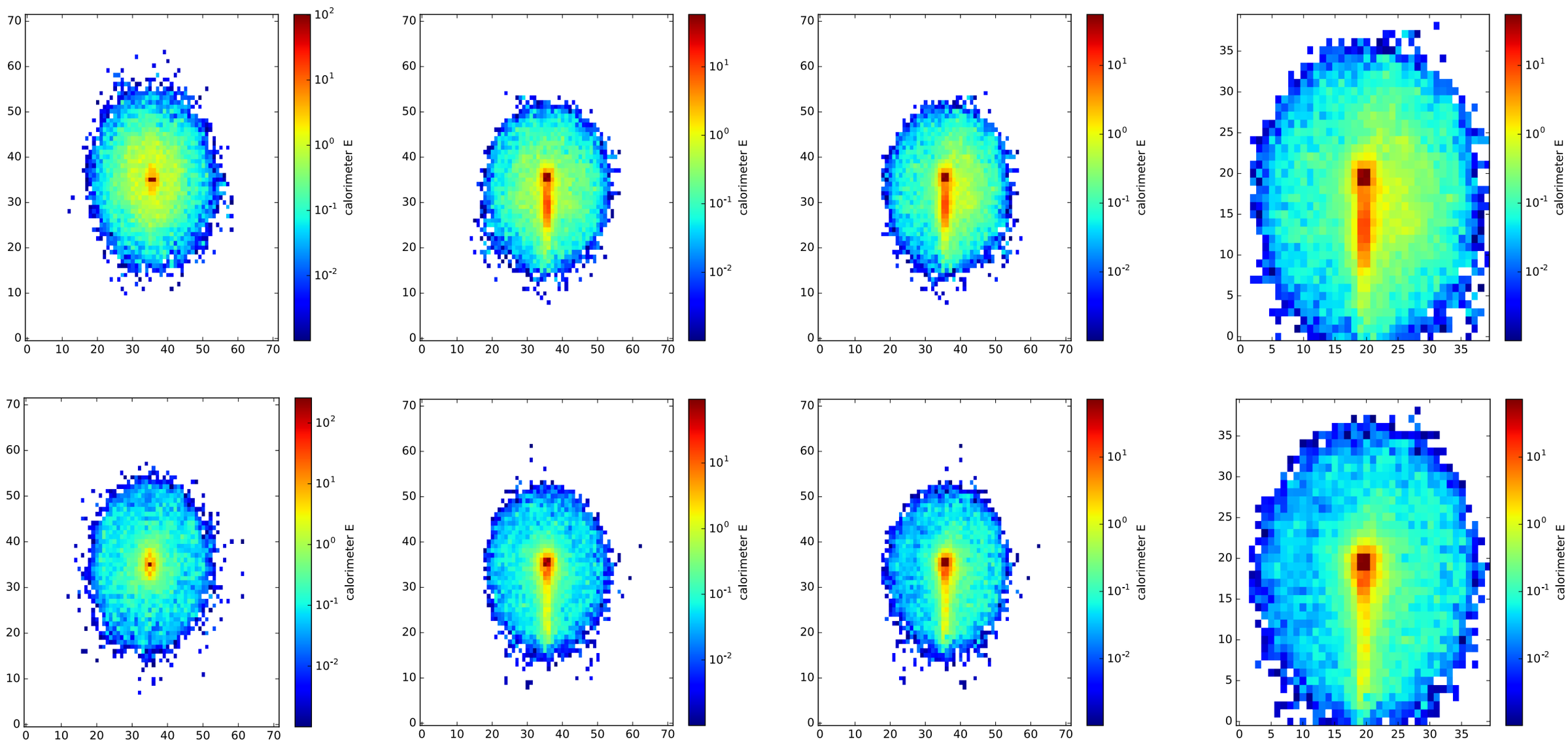
DNNs do not perform well on ‘raw’ images

Just as in computer vision, massage images before inputting to network

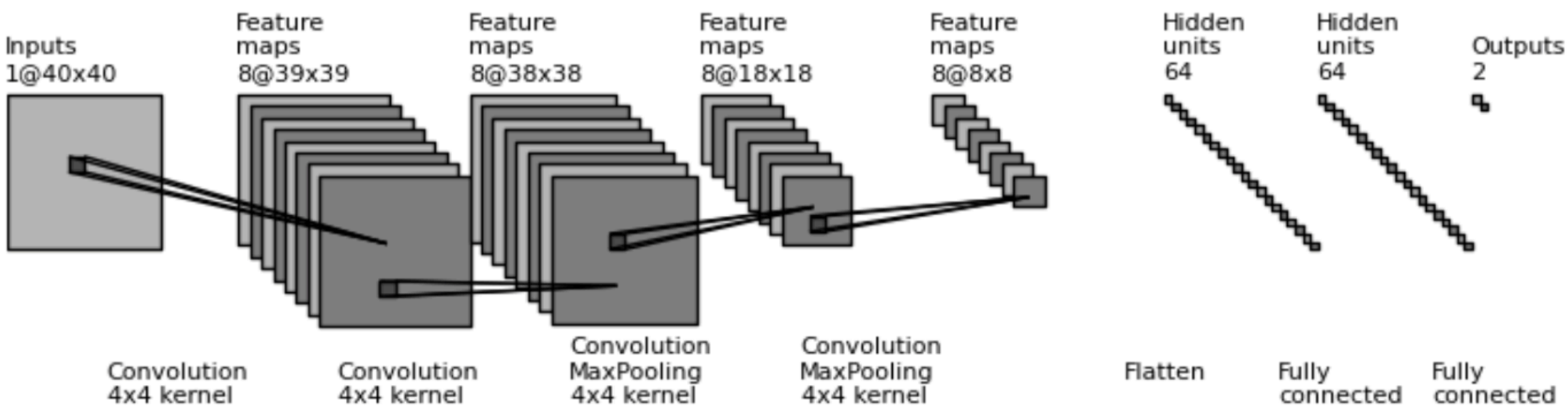
Shift | Rotate | Flip | Chop

S

B



Network architecture*



KERAS with Theano backend

300,000 signal and background images

Vary network 'hyperparameters'

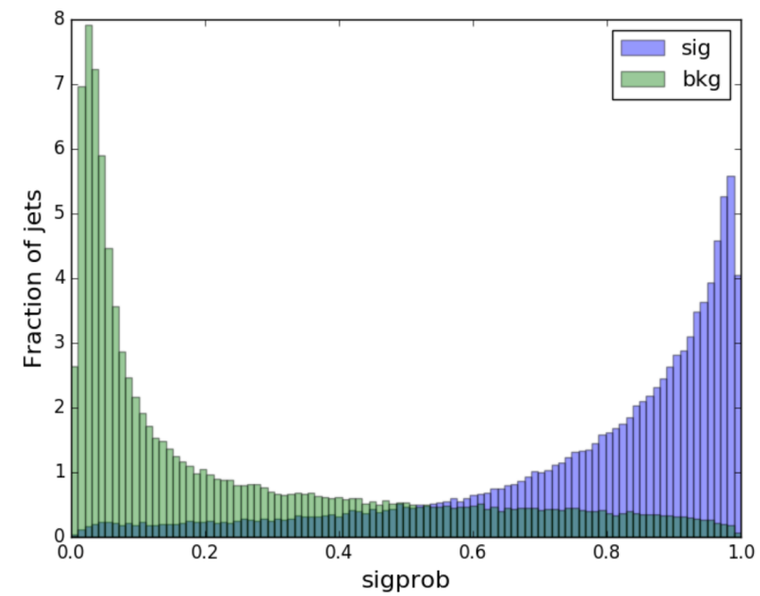
* don't ask

Performance

Final output from network

0 = certain background

1 = certain signal

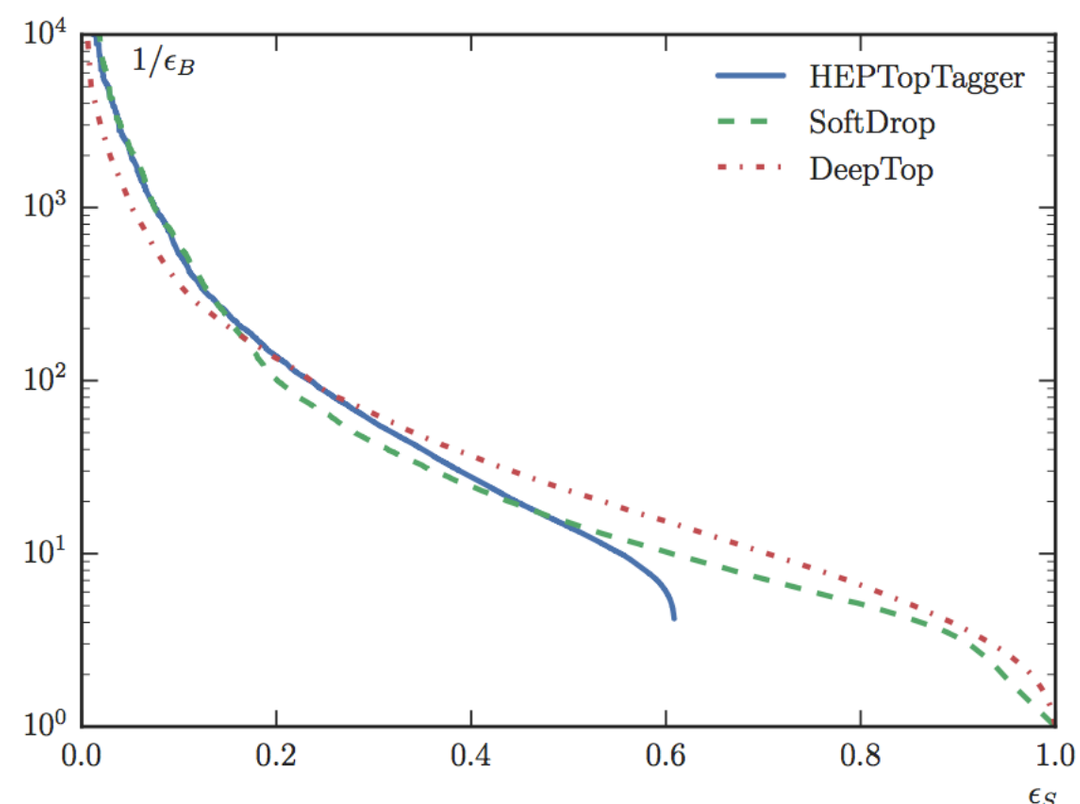


convert to efficiency curve

Benchmark against standard taggers

similar performance in high-purity regime

DNN far superior in high-acceptance regime!



Playing around

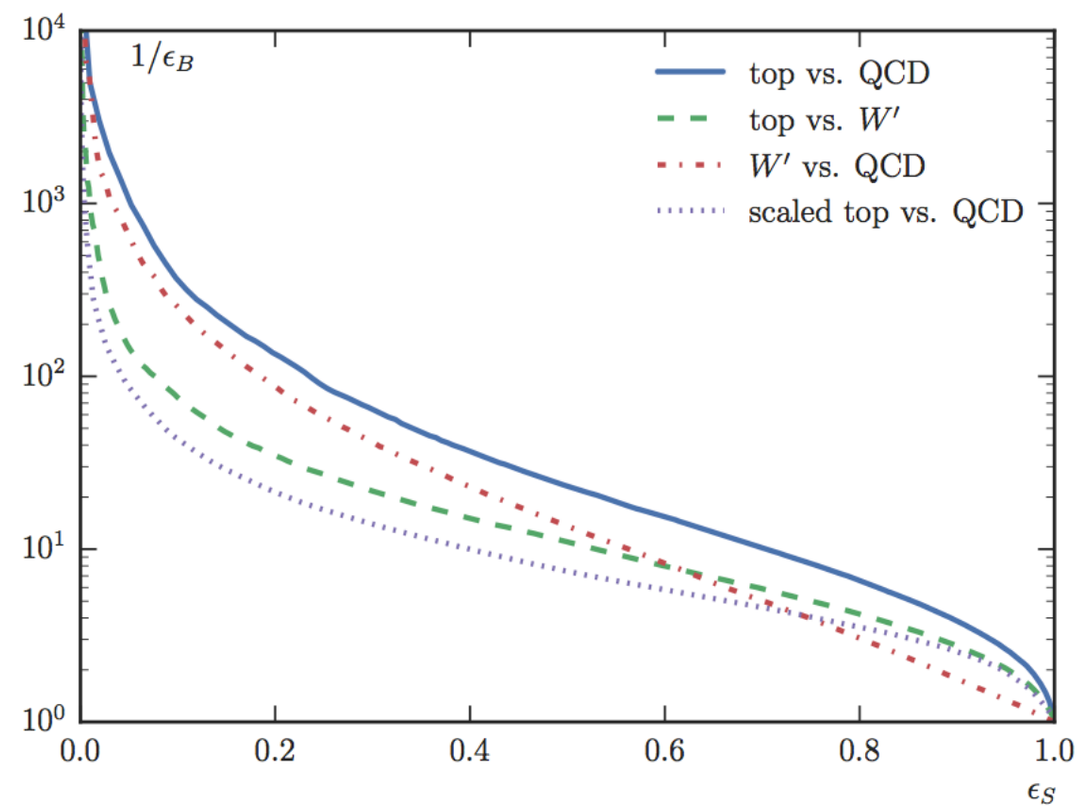
What is the network actually learning?

absolute mass scales, or mass drop?

- scale top and W mass by same factor

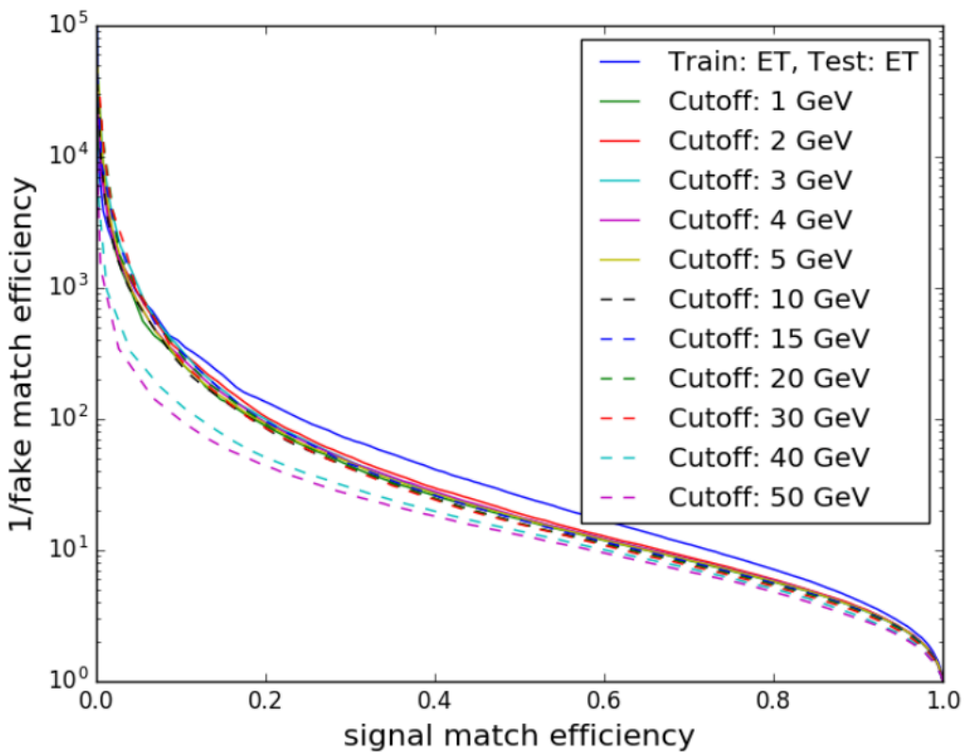
pronginess?

- replace tops with W's

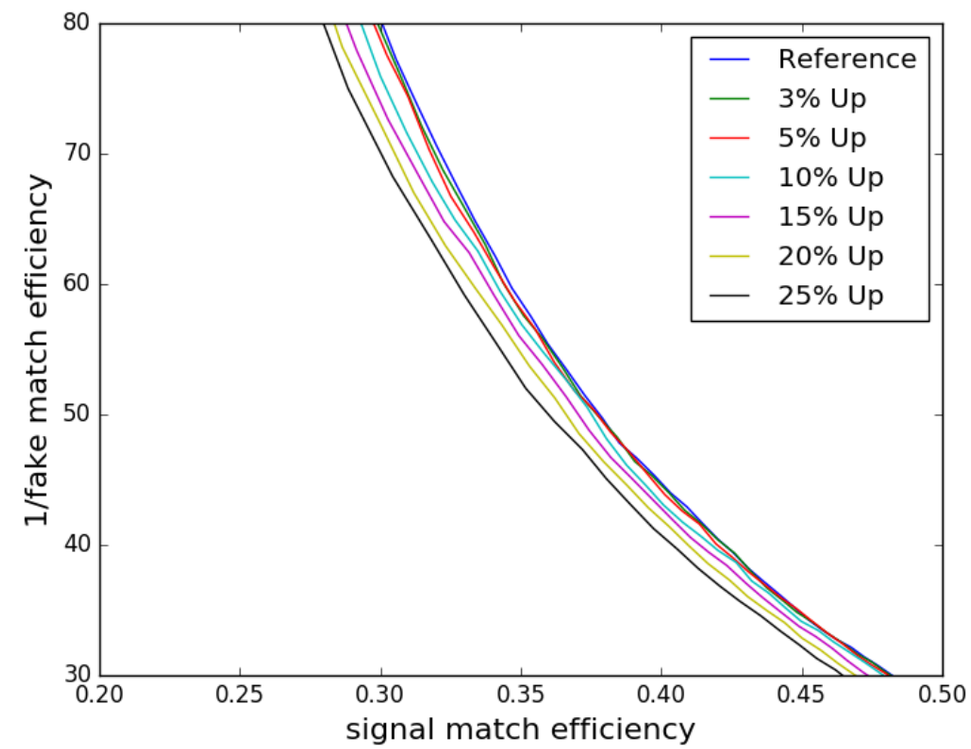


Playing around

How sensitive are results to experimental effects?



vary calorimeter threshold



vary jet energy scale

Summary

Jet images are a powerful new paradigm for boosted tagging

Comparable performance to QCD-inspired taggers

But a sacrifice in understanding of what is learned

Many open questions

What are these networks really learning?

What is the ultimate reach of this approach?

Can we move from MC to data?