RUTGERS
THE STATE UNIVERSITY
OF NEW JERSEY

Pulling Out All the Tops with Computer Vision and Deep Learning

Sebastian Macaluso
NHETC, Rutgers University

Machine Learning for Jet Physics
Fermilab - November 15, 2018

SM & David Shih [arXiv:1803.00107]

Boosted top and QCD jets

- Collimated decay products of boosted particles reconstructed as a “fat jet”.

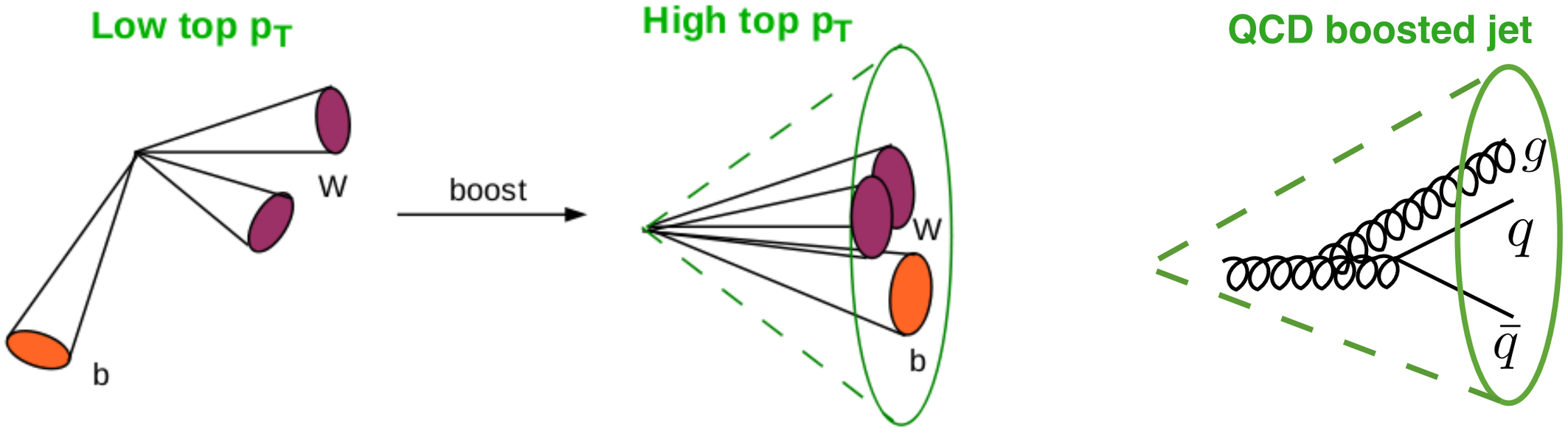


Fig. from www.quantumdiaries.org

- Typical classifiers use physical observables, e.g jet mass, etc
  - ⋆ Traditional methods include the HEPTopTaggerV2, CMSTT, …

- Apply them on a cut and count analysis or as high-level inputs of multivariate machine learning algorithms (BDTs).

New methods applying computer vision and deep learning

- Deep Neural Network (NN) classifier between boosted top and QCD jets.
- Low level inputs (e.g. momentum 4-vectors of jet constituents):
  - ⋆ NN “creates” most useful physical observables
  - ⋆ NN finds the optimal cuts on observables
- Starting point: baseline tagger inspired by
  “Deep Top” Tagger of Kasieczka et al [arXiv: 1701.08784]
  → top tagger comparable to state-of-the-art BDT’s.
- Our NN significantly outperforms conventional methods.

Convolutional Neural Networks (CNN)

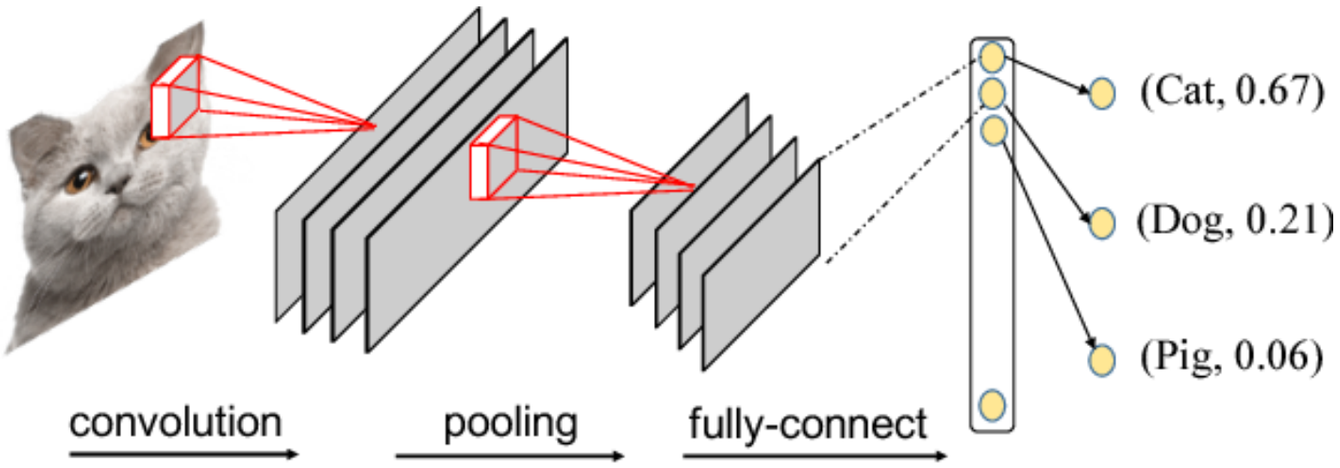
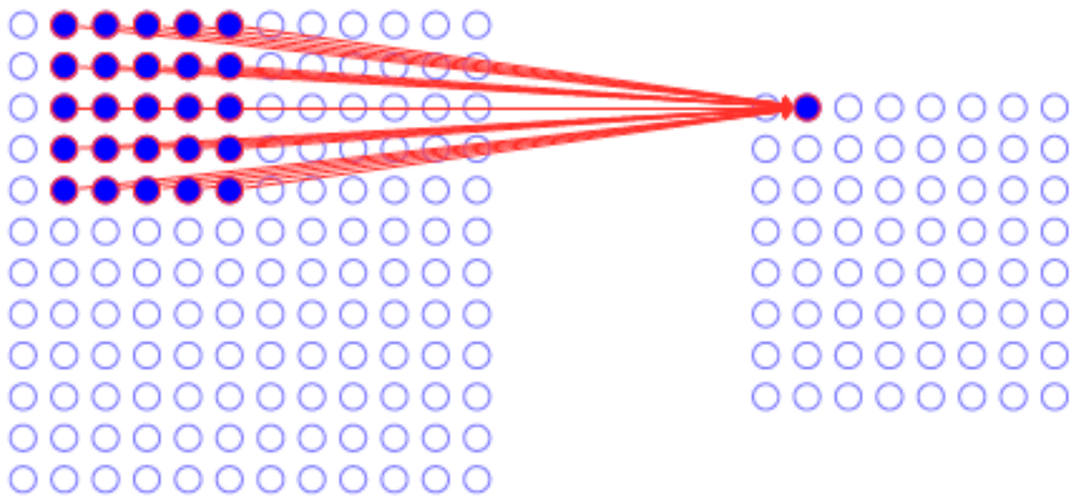


Fig. from arXiv:1712.01670

- Led to breakthroughs in computer vision:
  - Pixel level labeling of images for autonomous vehicles.
  - Google's automatic captioning of images.
  - Facebook's DeepFace project.
  - Microsoft surpassing human-level performance on ImageNet classification.

- Implement locality.
- Three basic ideas:
  - ★ **Filters**
  - ★ **Shared weights**
  - ★ **Pooling**



Pipeline

Event generation

- All-hadronic $t\bar{t}$ (signal) and QCD (background) dijet events.
- Pythia+Delphes (FastJet and CMS card) +PyROOT.

Generate and preprocess images

- (Python+Numpy)

CNN implementation and analysis

- Keras+Tensorflow on an NVidia Tesla K80 GPU.

Jet Samples

[Kasieczka et al, ’17] [CMS-PAS-JME-15-002]

	DeepTop	CMS
Jet sample	14 TeV $p_T \in (350, 450)$ GeV, $\lvert\eta\rvert < 1$ $R = 1.5$ anti-k_T calo-only match: $\Delta R(t, j) < 1.2$ merge: NONE	13 TeV $p_T \in (800, 900)$ GeV, $\lvert\eta\rvert < 1$ $R = 1$ anti-k_T particle-flow match: $\Delta R(t, j) < 0.6$ merge: $\Delta R(t, q) < 0.6$
Image	40×40 $\Delta\eta = 4$, $\Delta\phi = \frac{10}{9}\pi$	37×37 $\Delta\eta = \Delta\phi = 3.2$
Colors	p_T^{calo}	$(p_T^{neutral}, p_T^{track}, N_{track}, N_{muon})$

First 3 colors used in [Komiske, Metodiev, Schwartz ’16]

Jets as images

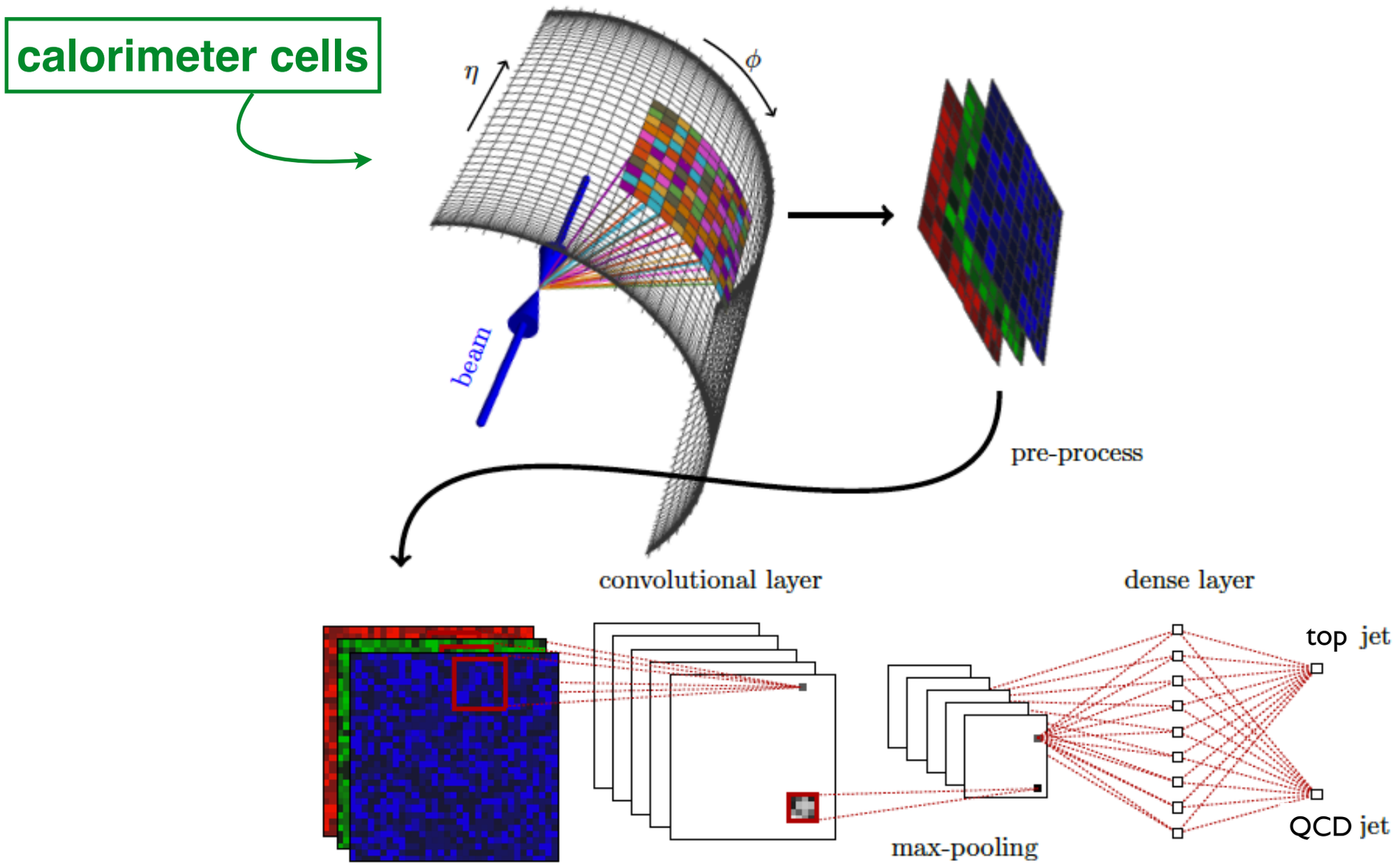
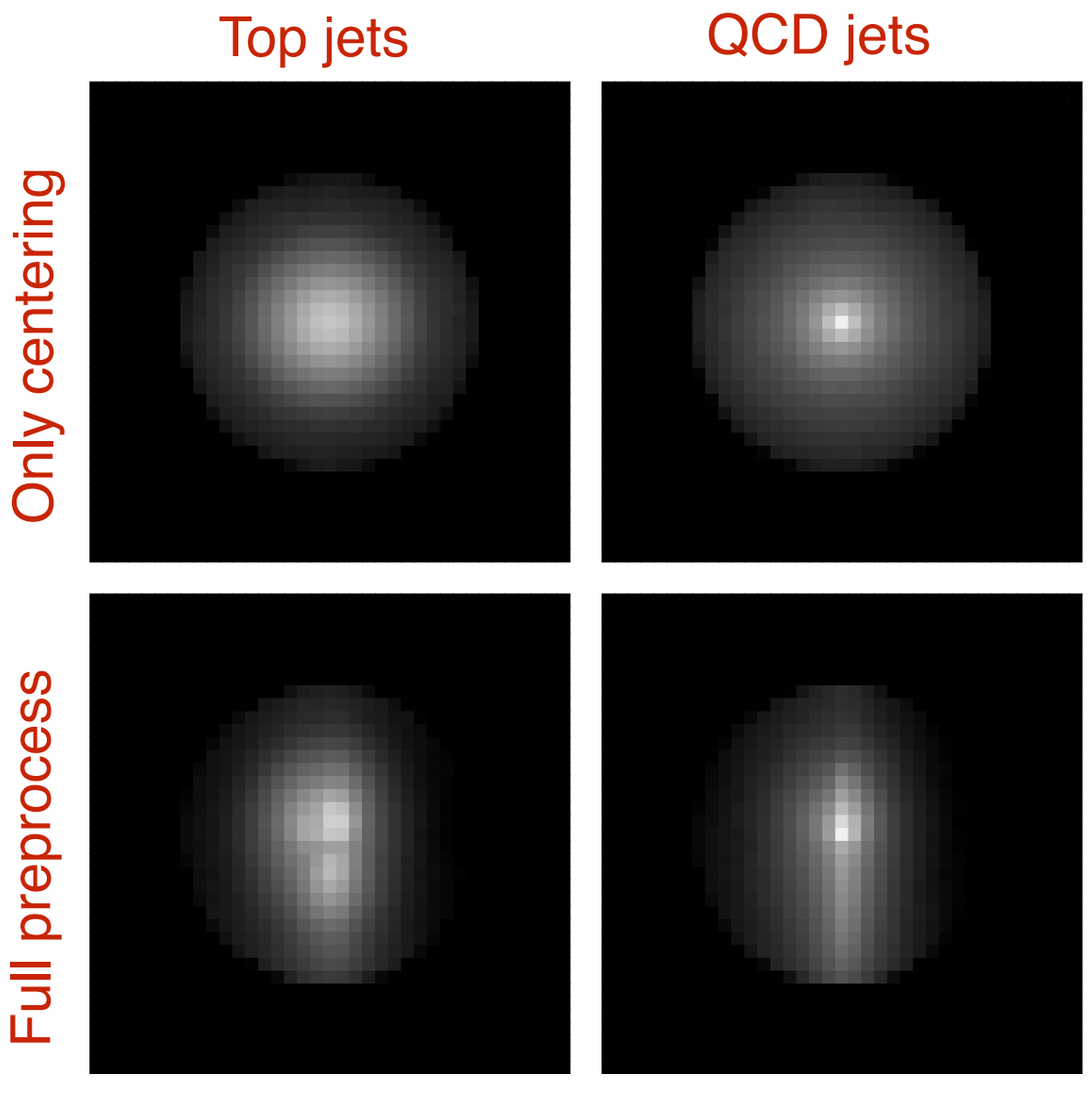


Fig. from [Komiske, Metodiev & Schwartz ’16]

Jet images

- Image preprocessing: center → rotate → flip → normalize → pixelate.



Average of 100k grayscale jets (total p_T in each pixel).

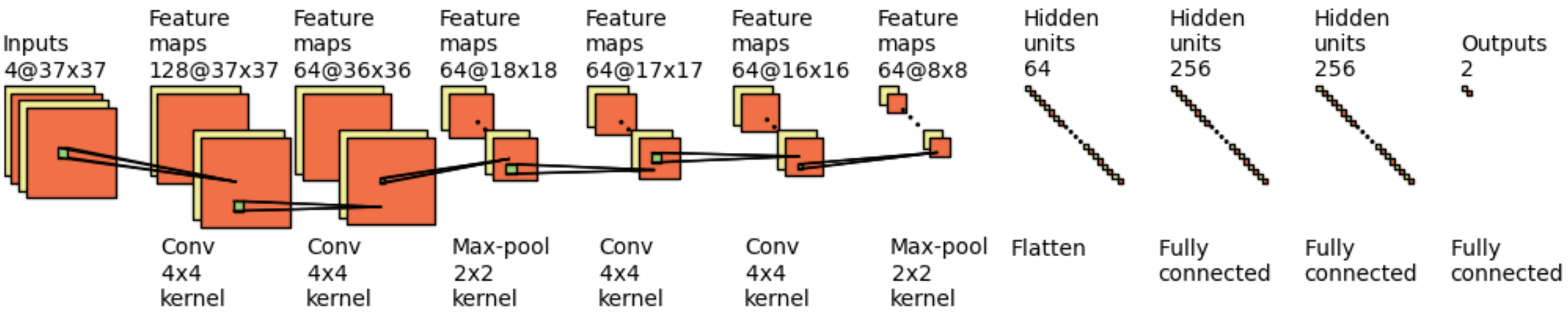
Neural Network Architecture

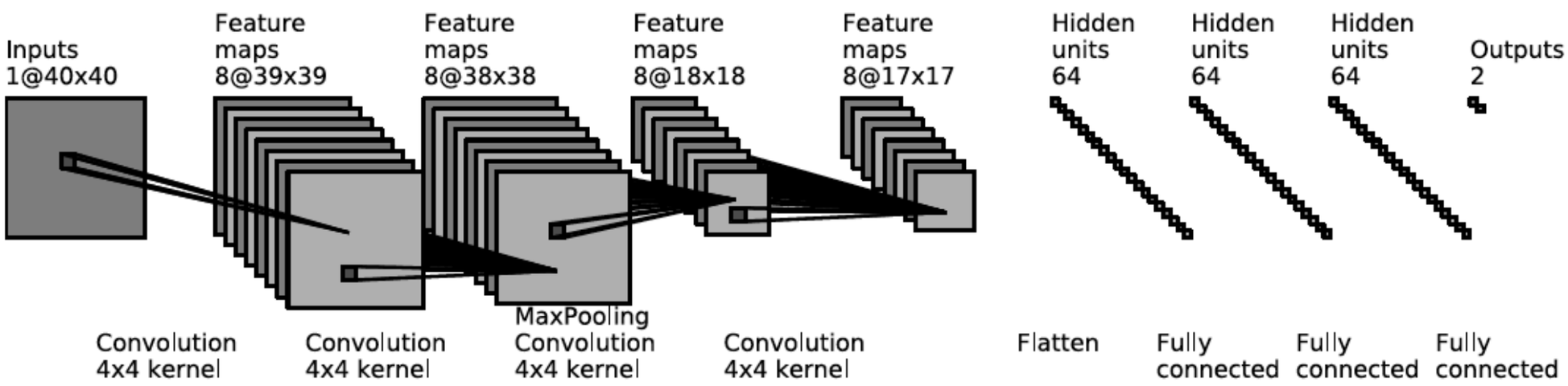
Input image (4 colors)

Low level features (lines, edges, etc)

Complex and more abstract concepts

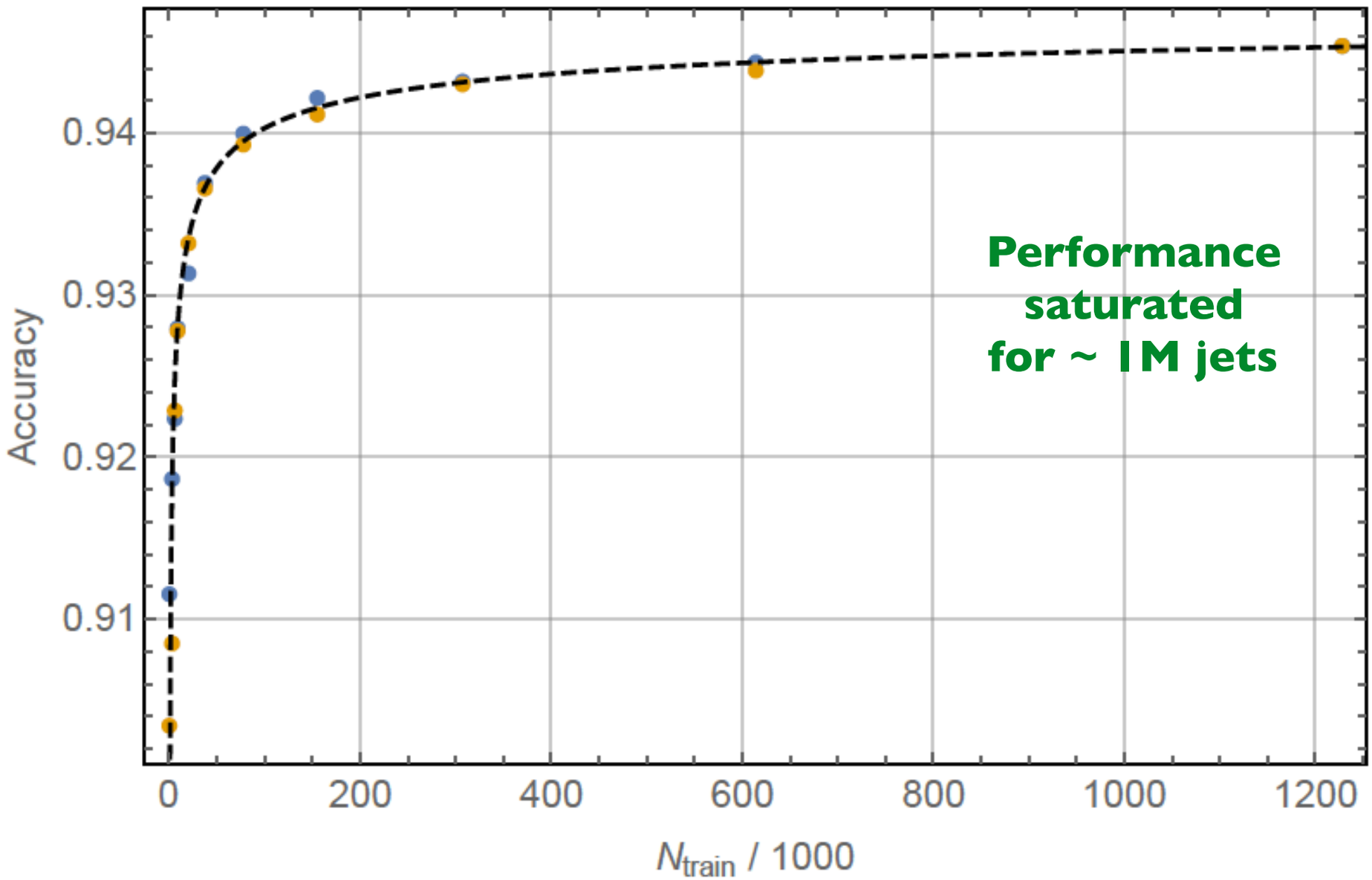
Output (top or Not top)





“Deep Top” Tagger [arXiv: 1701.08784]

Sample size

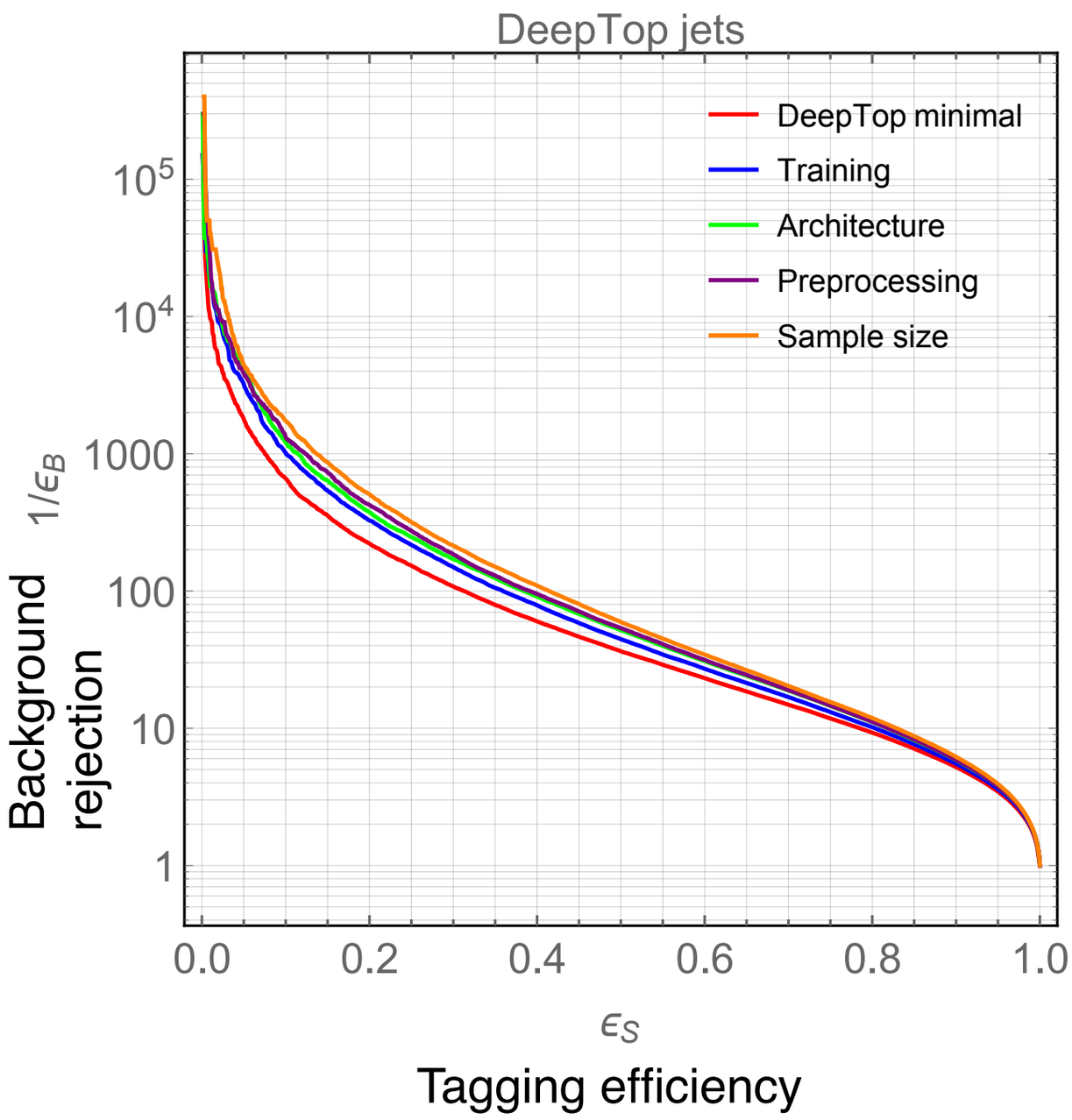


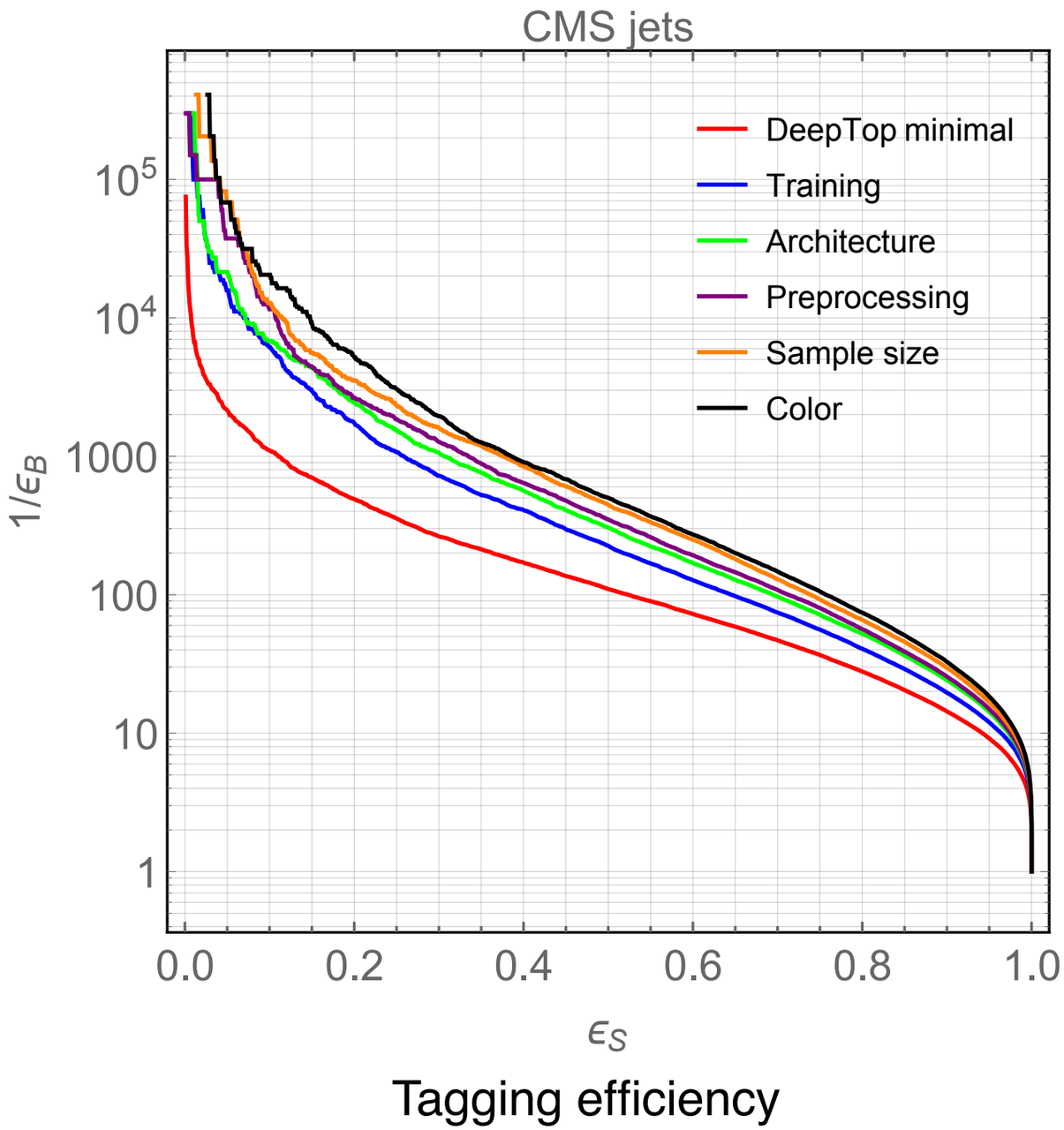
Least-squares fit → $a + b/N_{train}^{c}$ [Cortes et al, '94]

Improvements

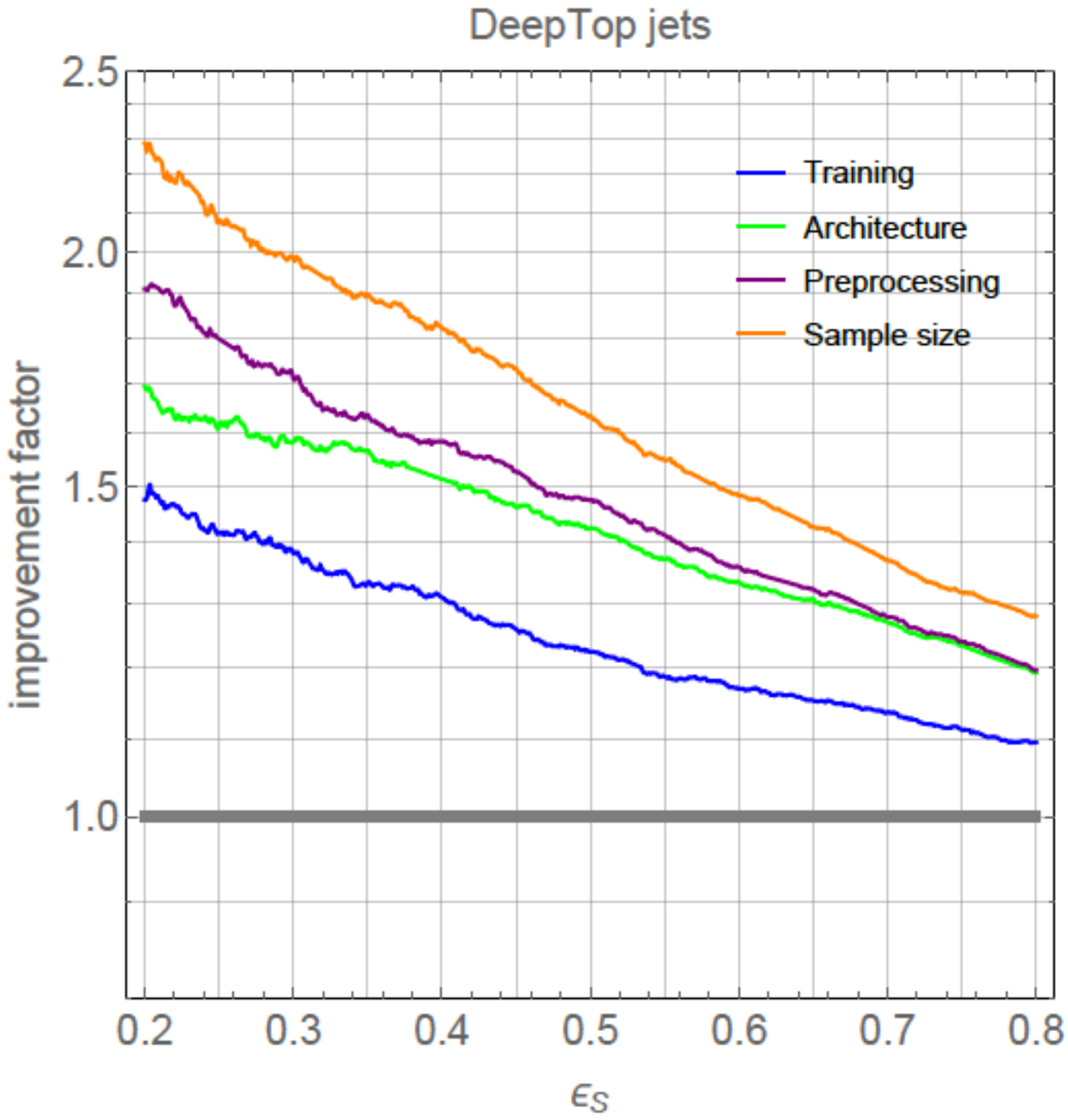
	DeepTop minimal	Our final tagger
Training	SGD $\eta = 0.003$ minibatch size=1000 MSE loss	AdaDelta $\eta = 0.3$ with annealing schedule minibatch size=128 cross entropy loss
CNN architecture	8C4-8C4-MP2-8C4-8C4-64N-64N-64N	128C4-64C4-MP2-64C4-64C4-MP2-64N-256N-256N
Preprocessing	pixelate→center → normalize	center→rotate→flip → normalize→pixelate
Sample size	150k+150k	1.2M+1.2M
Color	$p_T^{calo} = p_T^{neutral} + p_T^{track}$	$(p_T^{neutral}, p_T^{track}, N_{track}, N_{muon})$

Improvements

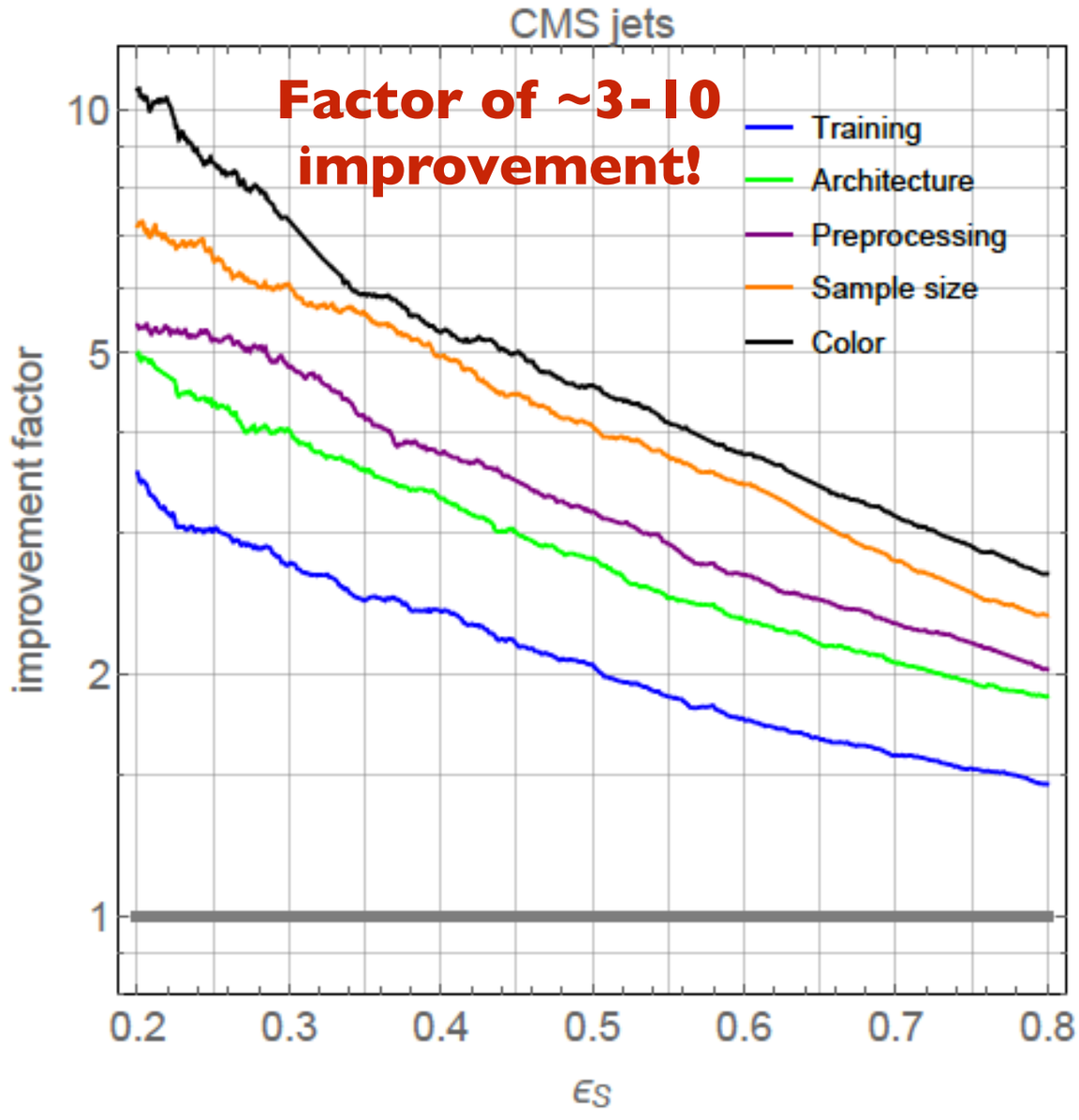




Improvements

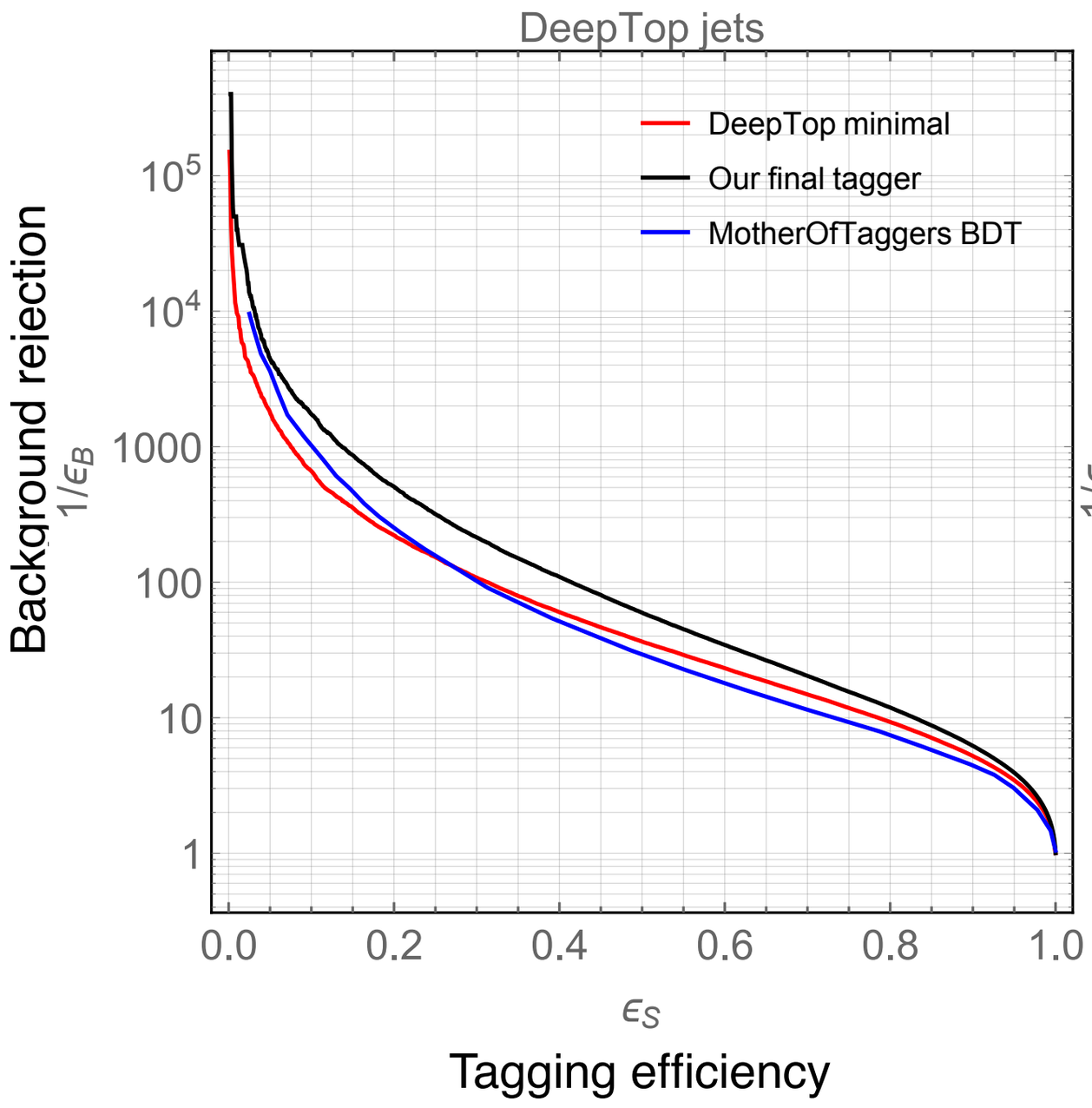


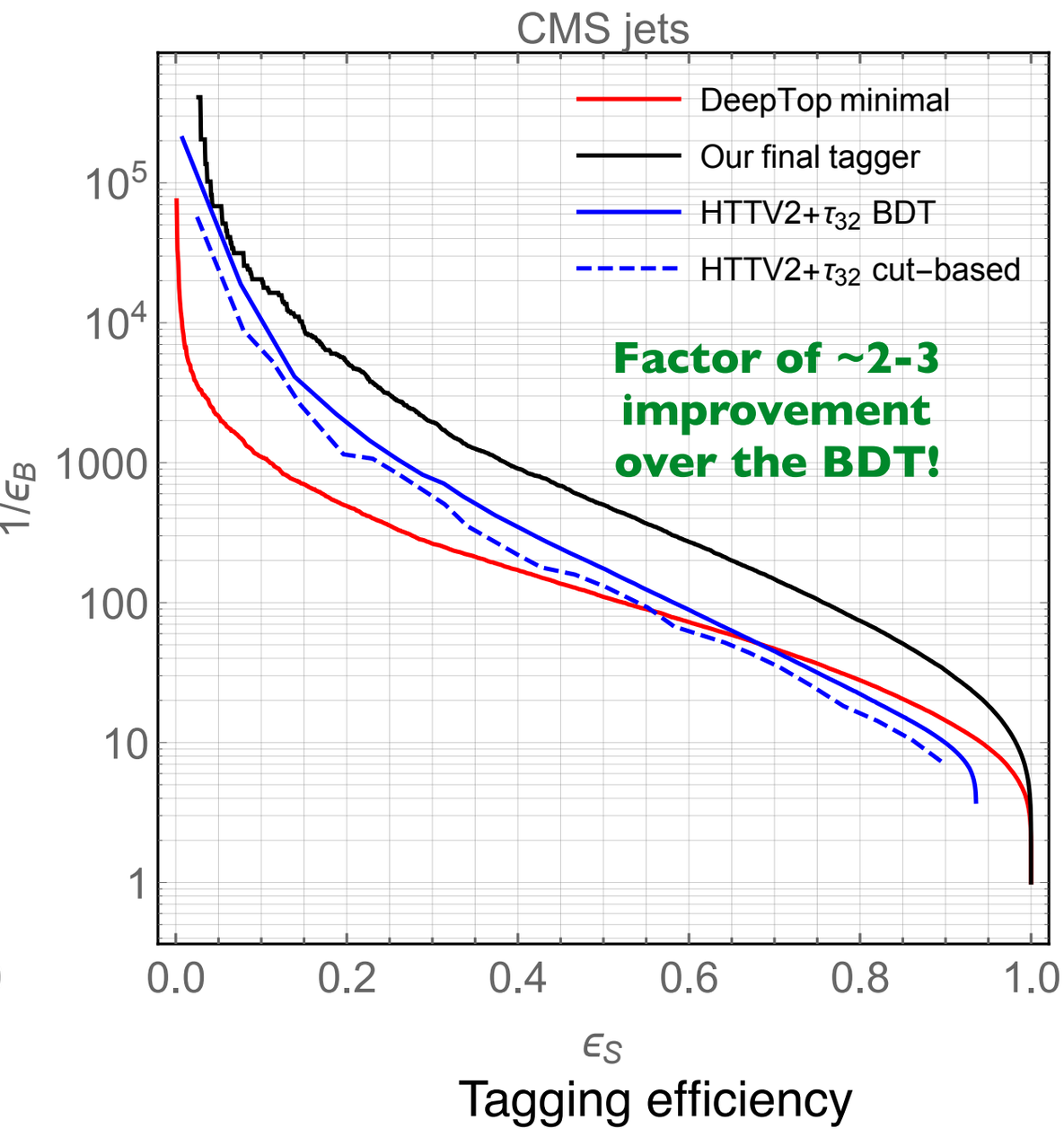
Tagging efficiency



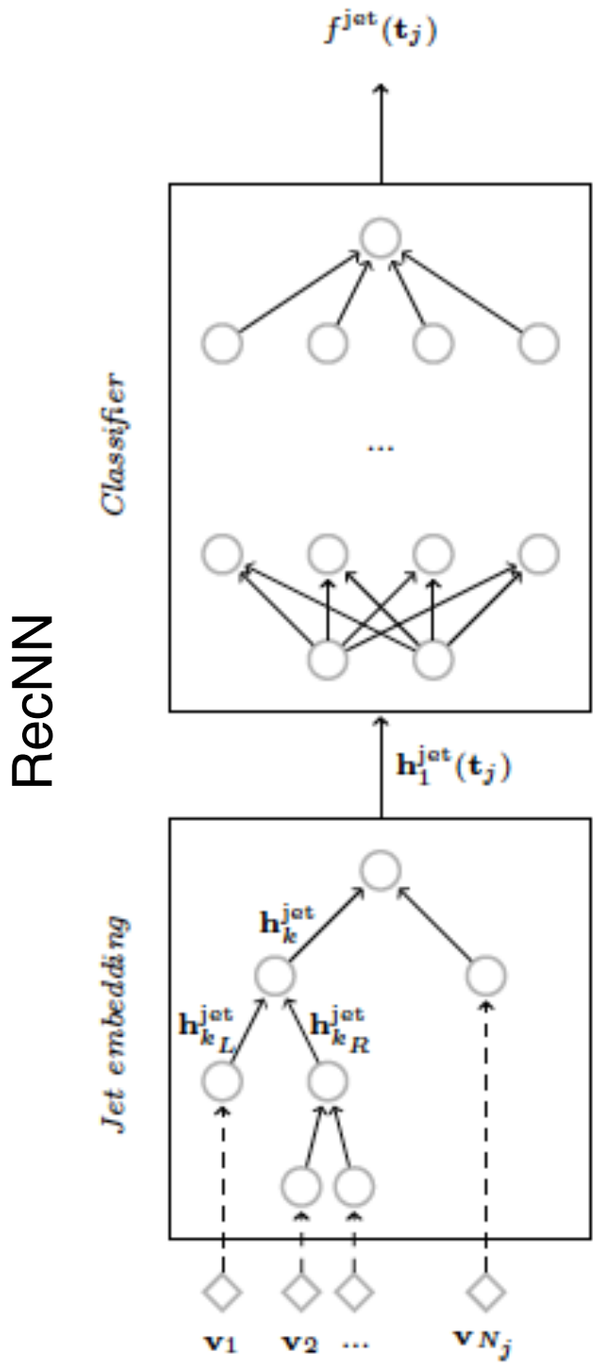
Tagging efficiency

Final comparison





What else can be said about top tagging?



[Louppe, Cho, Becot, Cranmer '17]

- Tagger for both boosted and non-boosted tops?
- Full events? Images vs trees?

Recursive Neural Networks

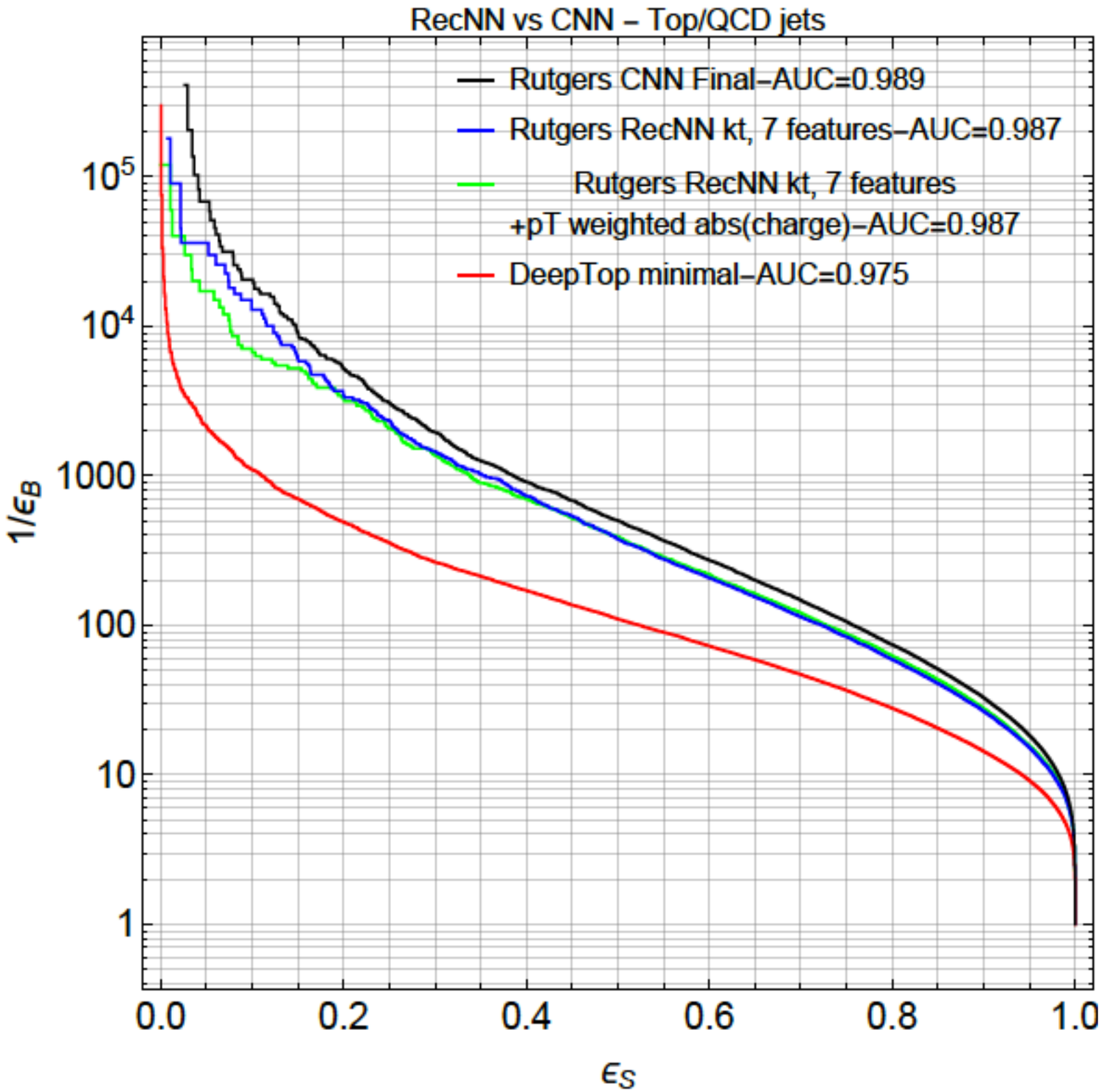
$$\mathbf{h}_k^{\text{jet}} = \begin{cases} \mathbf{u}_k & \text{if } k \text{ is a leaf} \\ \sigma\left(W_h \begin{bmatrix} \mathbf{h}_{k_L}^{\text{jet}} \\ \mathbf{h}_{k_R}^{\text{jet}} \\ \mathbf{u}_k \end{bmatrix} + b_h\right) & \text{otherwise} \end{cases}$$

$$\mathbf{u}_k = \sigma\left(W_u g(\mathbf{o}_k) + b_u\right)$$

- CMS sample.
- Pyroot+FastJet
- PyTorch batch training implementation (initial code in Numpy).
- RecNN much fewer trainable parameters: ~10000 vs ~1 M

SM & Matt Buckley, Dipsikha Debnath & David Shih, work in progress …

Comparison



So far …. boosted jets

- Recluster with kt and C/A algorithm better performance than anti-kt
- No preprocessing
- 7 features: lpl, eta, phi, E, E/jet_E, pT, theta.
- Added new info: charge, abs(charge), muon id, etc but performance does not improve → upper bound on tagging accuracy?

Summary

- CNN top tagger significantly outperforms state-of-the-art conventional top taggers based on high level inputs.
- Methodology could be straightforwardly extended to the classification of other types of boosted objects such as *W/Z* bosons, Higgses, and BSM particles.
- Interesting to do a more complete architecture and hyperparameter scan, with access to a GPU cluster, to see if the NN performance could be further improved.
- Other architectures could allow for new implementations.

Thanks for your attention!

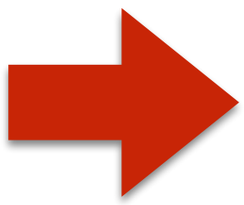
Motivations

- We have seen the relevance of top quarks in:
  - ★ Searches for new physics.
  - ★ The hierarchy problem and contributions to the higgs mass.

- Heavy particles decays typically give signatures with boosted objects.
  - ★ Top partners can naturally produce boosted top quarks in their decays.

- Gains in luminosity at LHC allow to apply harder cuts and still have enough statistics.

- Signatures with boosted top jets are predicted from Standard Model processes
  - ★ Single top
  - ★ Top pair production



Good motivations to build a highly effective boosted top tagger.

Top quark phenomenology

Jet → collimated spray of energetic charged and neutral hadrons.

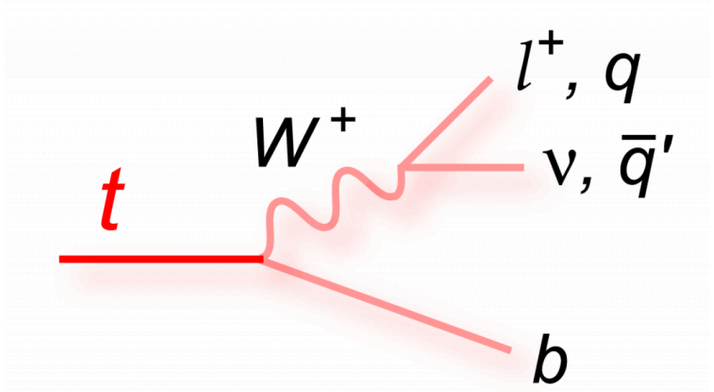


Fig. from www.quantumdiaries.org

- Leptonic decay → *b* quark jet+isolated lepton+MET.
- Hadronic decay → 3 jets.

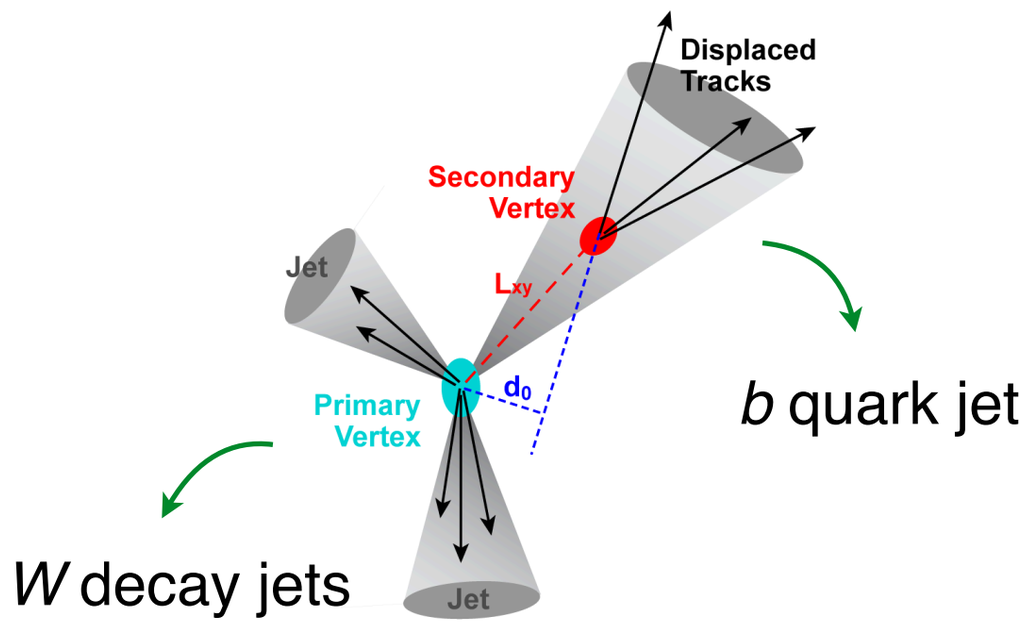
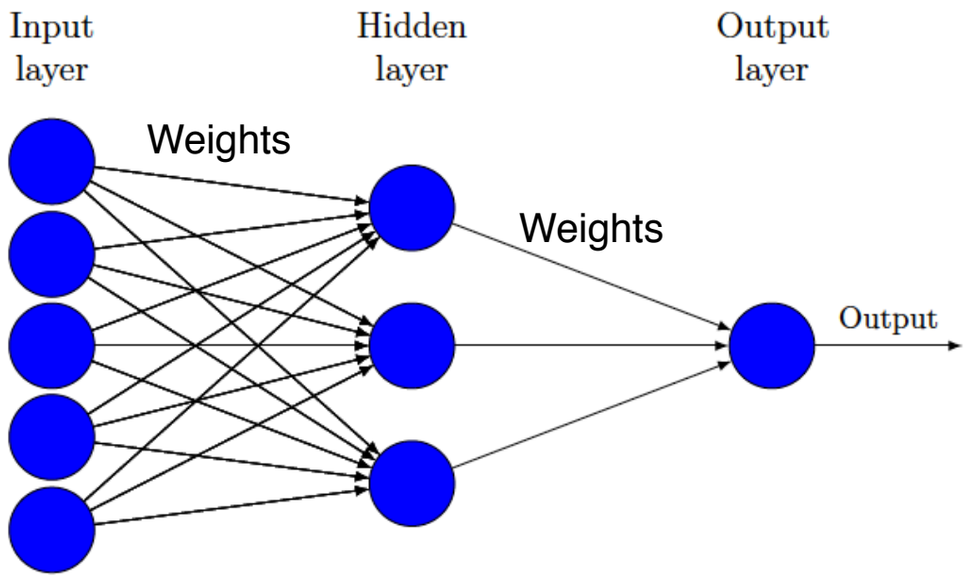


Fig. from www.amva4newphysics.wordpress.com

Neural Networks

- Learn real world notions by:
  - ★ Organizing its features in a hierarchical way.
  - ★ Building connections among them.
  - ★ Go from simple features to more complex and abstract ones.
  - ★ Features are arranged in *layers* and the connections are called *weights*.



- Loss function encodes the difference between the output and the true value.

Cross entropy

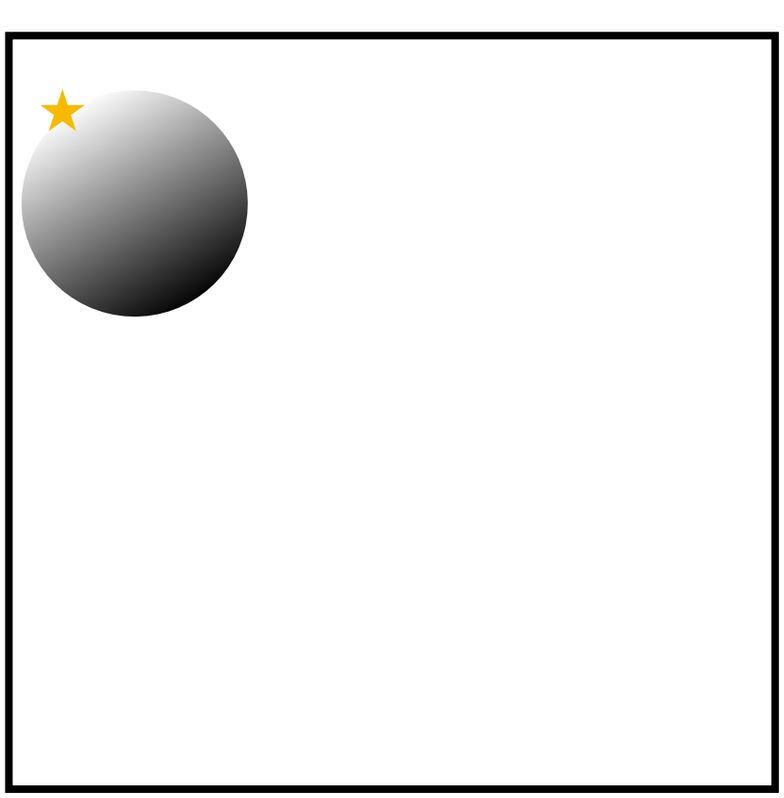
$$f(a, y) = -(y \log a + (1 - y) \log(1 - a))$$

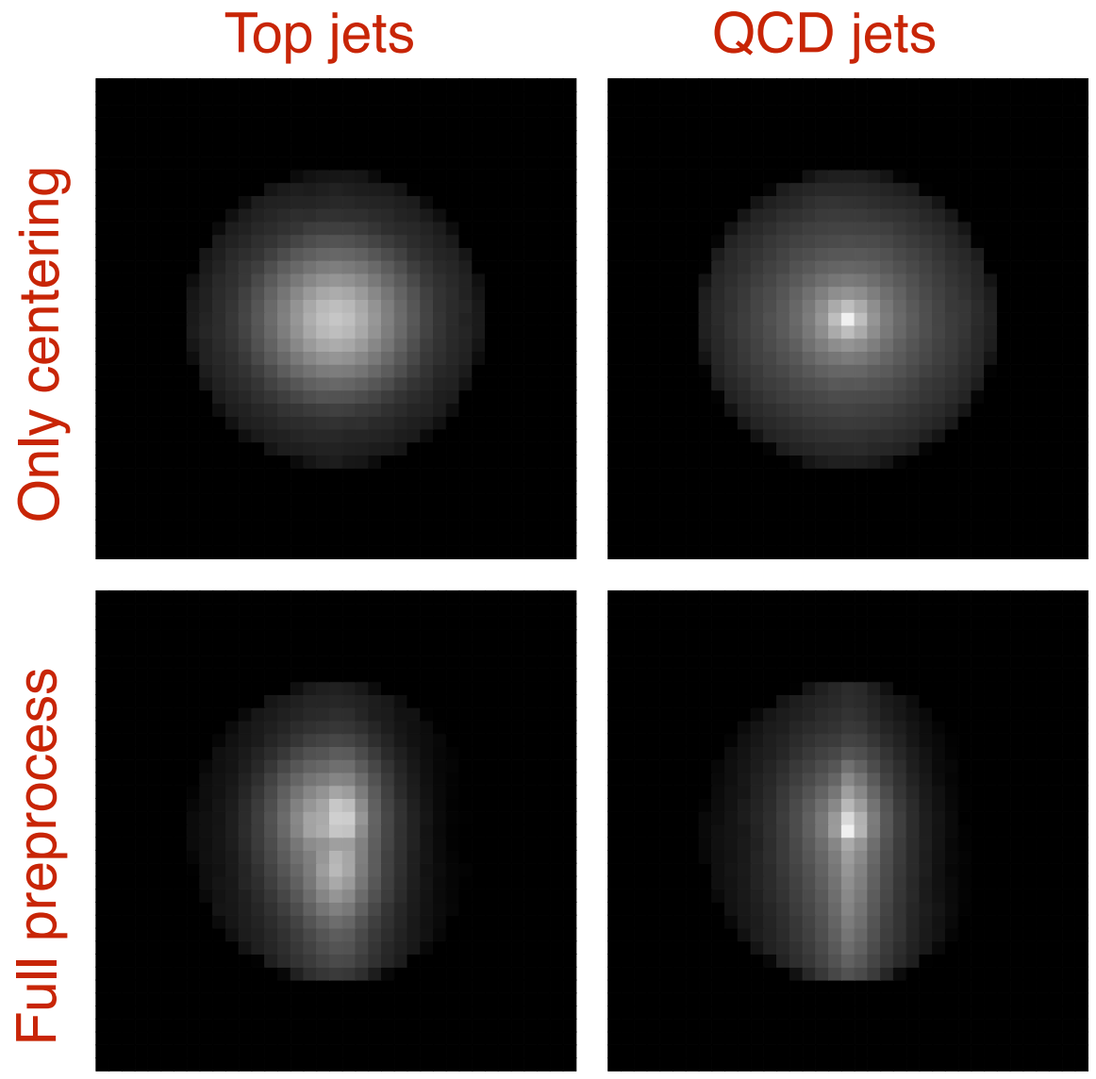
Mean-squared-error

$$f(a, y) = (a - y)^2$$

Jet images

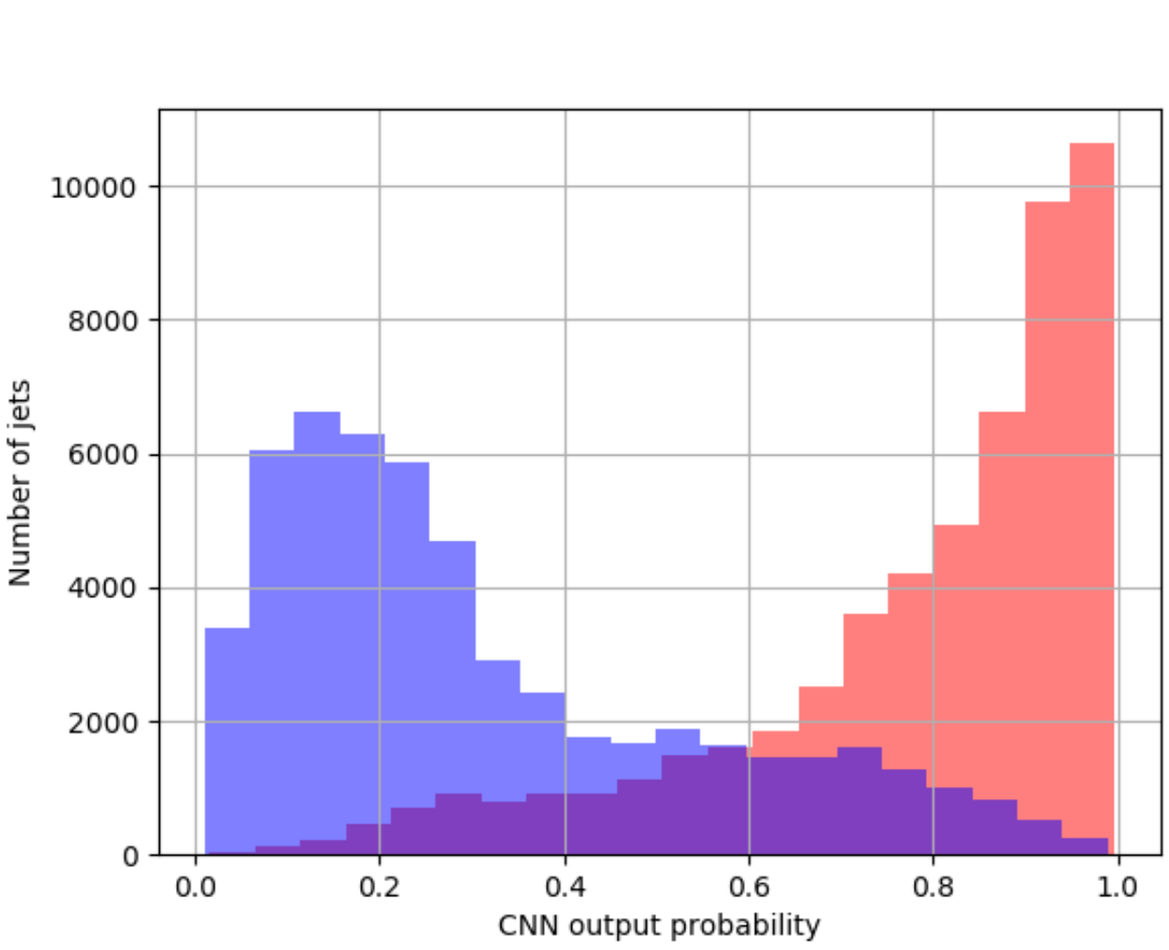
- Image preprocessing: center → rotate → flip → normalize → pixelate.

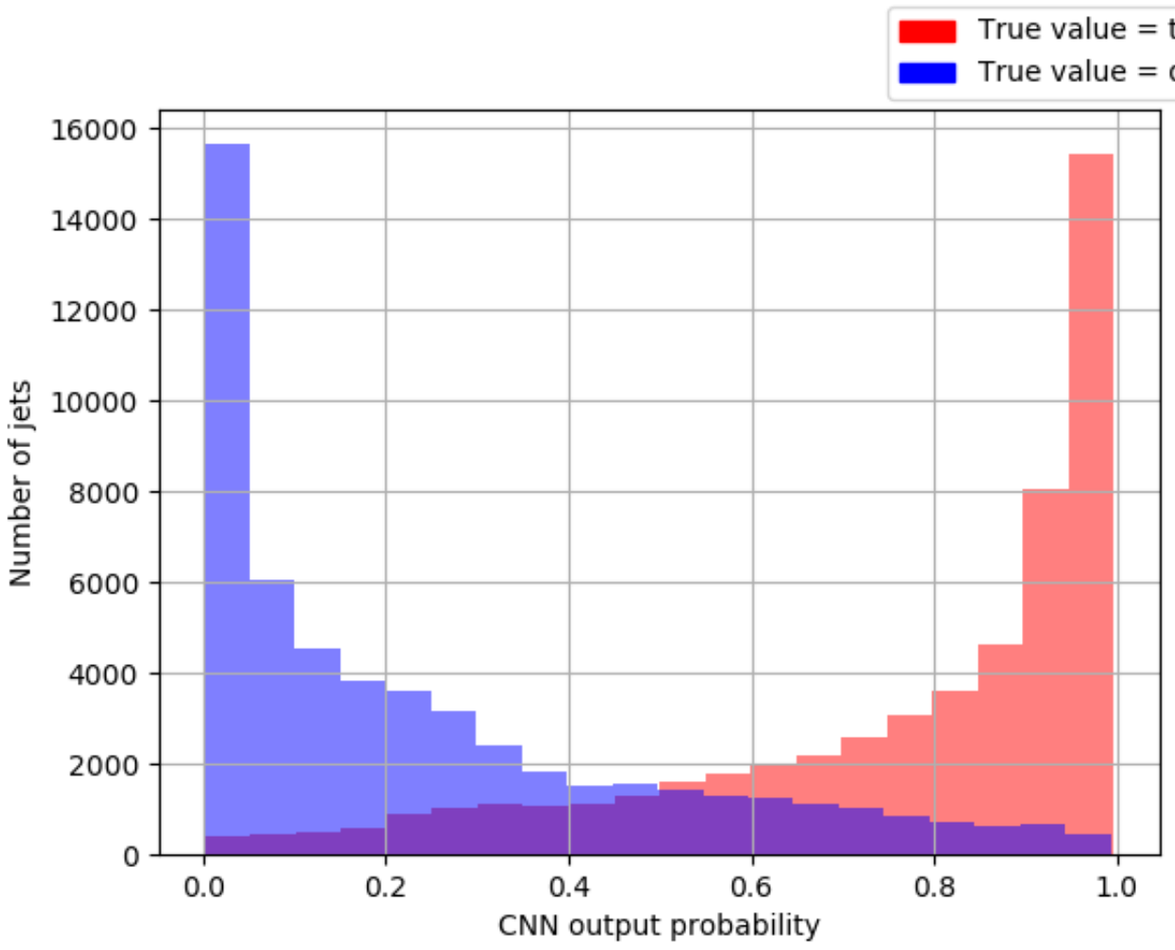




Average of 100k grayscale jets (total p_T in each pixel).

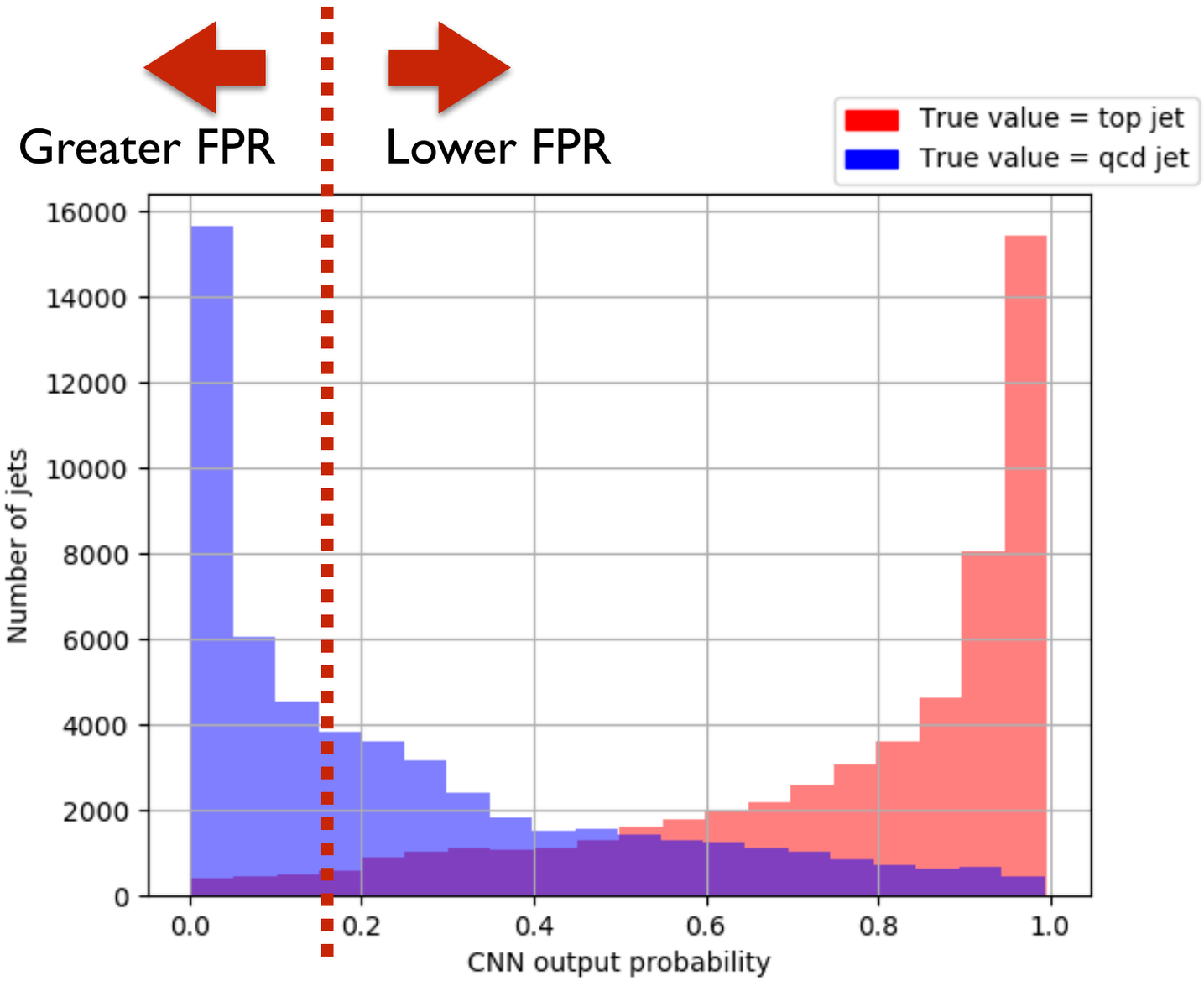
Finding the optimal weights

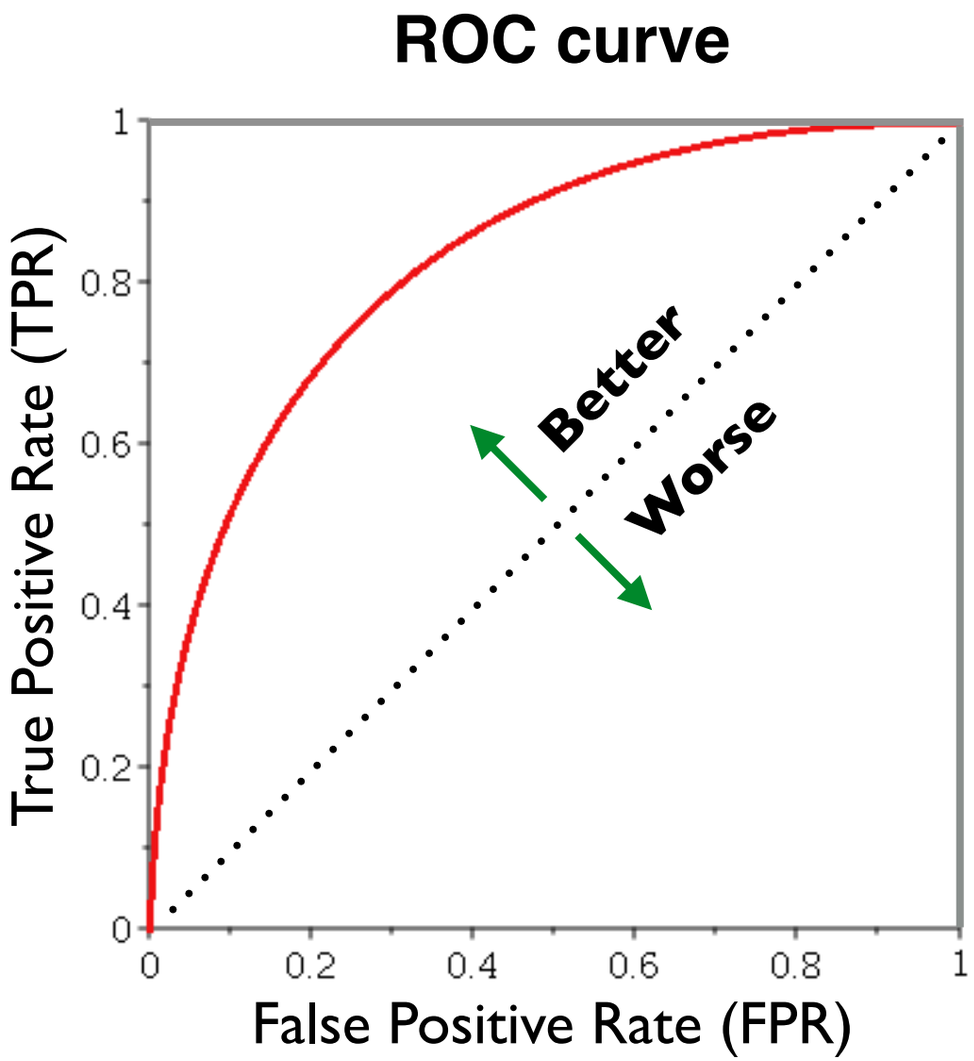




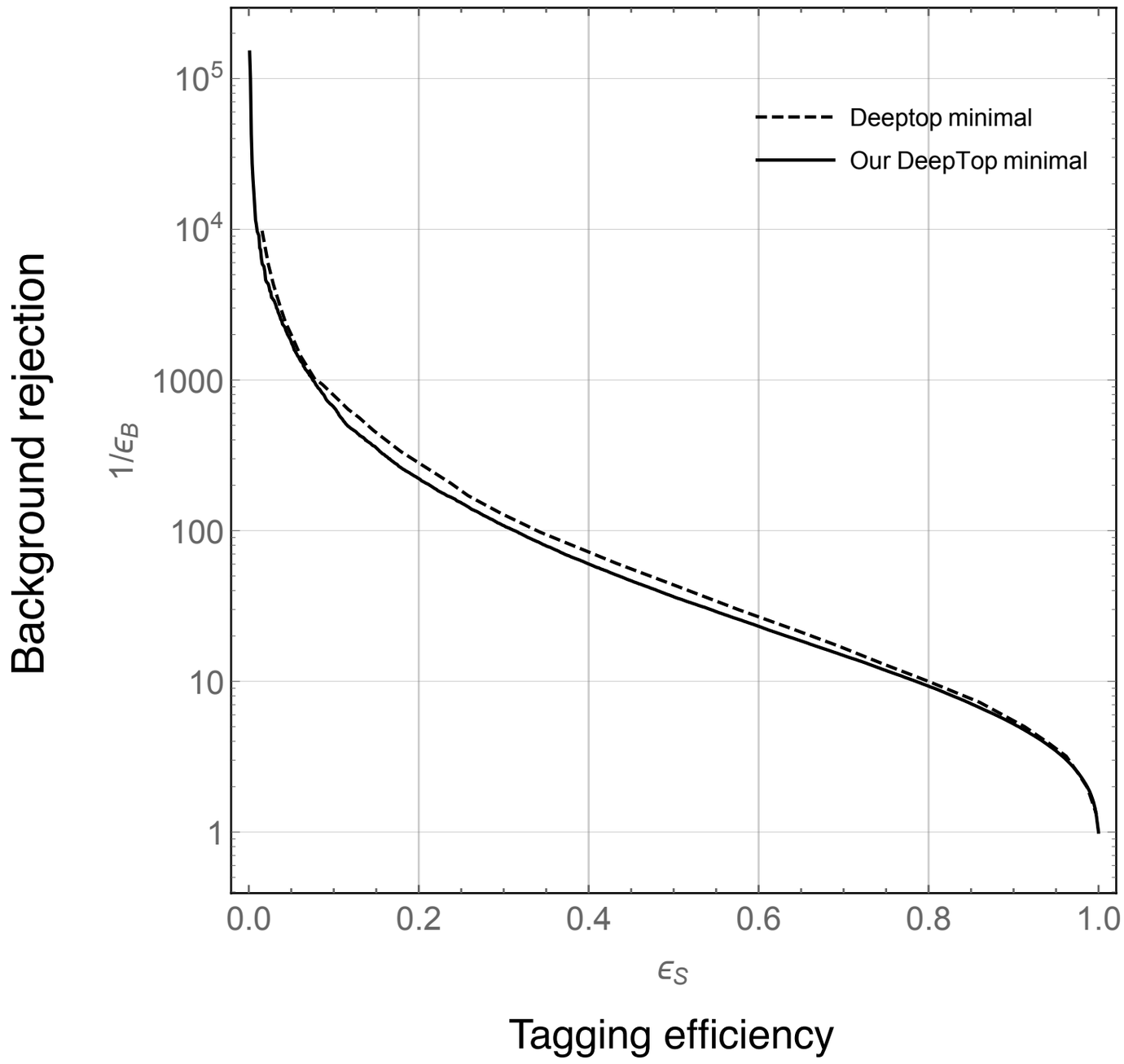
Goal : maximize the split between signal and background output probabilities

ROC curves

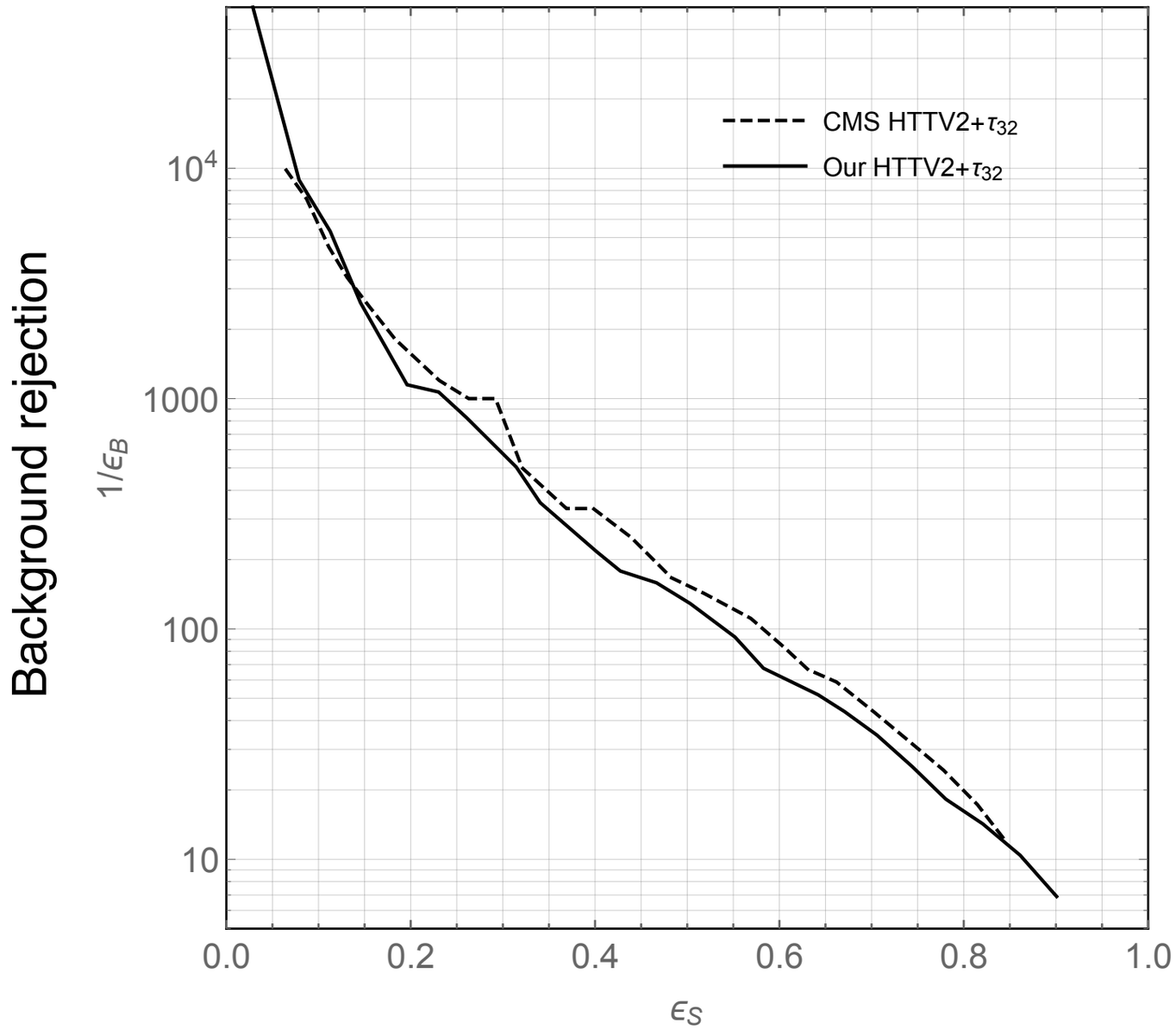

Greater FPR
Lower FPR
True value = top jet
True value = qcd jet
Number of jets
CNN output probability
0
2000
4000
6000
8000
10000
12000
14000
16000
0.0
0.2
0.4
0.6
0.8
1.0


ROC curve
True Positive Rate (TPR)
False Positive Rate (FPR)
Better
Worse
0
0.2
0.4
0.6
0.8
1

Deeptop validation



HTTV2+N-subjettiness validation



Merge requirement