Bayesian Networks — at the LHC

Tilman Plehn

Universität Heidelberg

Aachen 2/2020

Nothing is ever new

Machine learning and top tagging

– 1991: NN-based quark-gluon tagger [Lönnblad, Peterson, Rögnvaldsson]

USING NEURAL NETWORKS TO IDENTIFY JETS

Leif LÖNNBLAD*, Carsten PETERSON** and Thorsteinn RÖGNVALDSSON***

Department of Theoretical Physics, University of Lund, Sölvegatan 14A, S-22362 Lund, Sweden

Received 29 June 1990

A neural network method for identifying the ancestor of a hadron jet is presented. The idea is to find an efficient mapping between certain observed hadronic kinematical variables and the quark-gluon identity. This is done with a neuronic expansion in terms of a network of sigmoidal functions using a gradient descent procedure, where the errors are back-propagated through the network. With this method we are able to separate gluon from quark jets originating from Monte Carlo generated e^+e^- events with $\sim 85\%$ approach. The result is independent of the MC model used. This approach for isolating the gluon jet is then used to study the so-called string effect.

– but unclear how to define quarks vs gluons

Nothing is ever new

Machine learning and top tagging

– 1991: NN-based quark-gluon tagger [Lönnblad, Peterson, Rögnvaldsson]

USING NEURAL NETWORKS TO IDENTIFY JETS

Leif LÖNNBLAD*, Carsten PETERSON** and Thorsteinn RÖGNVALDSSON***

Department of Theoretical Physics, University of Lund, Sölvegatan 14A, S-22362 Lund, Sweden

Received 29 June 1990

A neural network method for identifying the ancestor of a hadron jet is presented. The idea is to find an efficient mapping between certain observed hadronic kinematical variables and the quark-gluon identity. This is done with a neuronic expansion in terms of a network of sigmoidal functions using a gradient descent procedure, where the errors are back-propagated through the network. With this method we are able to separate gluon from quark jets originating from Monte Carlo generated e^+e^- events with $\sim 85\%$ approach. The result is independent of the MC model used. This approach for isolating the gluon jet is then used to study the so-called string effect.

– but unclear how to define quarks vs gluons

– top jets from $t \to bq\bar{q}'$ vs QCD jets

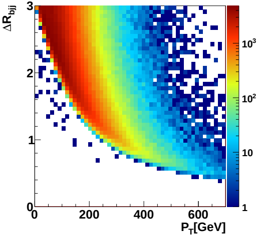
motivation: $Z' \to t\bar{t}$ with $p_{T,t} > 300$ GeV

theory: top decays perturbative QCD

experiment: labelled semileptonic $t\bar{t}$ events

simulation: fast and high-quality MC data

$\Rightarrow$ Fat top jets perfect ML playground



Jet image machines

Next step in LHC analyses [Cogan etal, Oliveira, Nachman etal, Baldi, Whiteson etal (2014/15)]

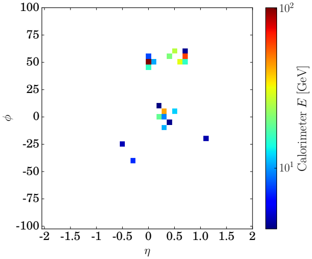
– why intermediate high-level variables?

– as much data as possible

– calorimeter output as image

– eventually, adding tracker output

$\Rightarrow$ Deep learning = modern networks on low-level observables

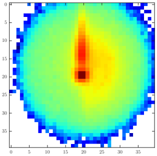


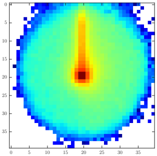
Jet image machines

Next step in LHC analyses [Cogan etal, Oliveira, Nachman etal, Baldi, Whiteson etal (2014/15)]

- why intermediate high-level variables?
- as much data as possible
- calorimeter output as image
- eventually, adding tracker output

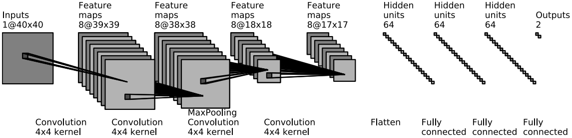
$\Rightarrow$ Deep learning = modern networks on low-level observables





Convolutional network [Kasieczka, TP, Russell, Schell; Macaluso, Shih]

- image recognition standard ML task
- rapidity vs azimuthal angle, colored by energy deposition
- 40 $\times$ 40 bins through calorimeter resolution



Theory work?

4-vector input — graph CNN [Butter, Kasieczka, TP, Russell; much better versions by now]

– physics objects from calorimeter and tracker

– distance measure known from e&m [alternatively: Erdmann, Rath, Rieger]

Inspired by QFT

– input 4-vectors $(k_{\mu,i})$

– jet algorithm $\longrightarrow$ combination layer

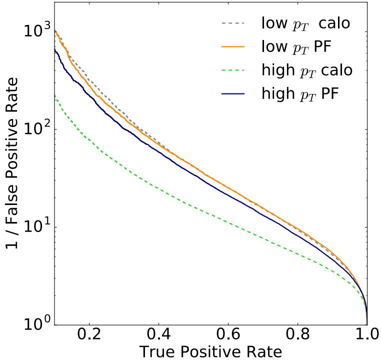
$$k_{\mu,i} \xrightarrow{\text{CoLa}} \widetilde{k}_{\mu,j} = k_{\mu,i}\, C_{ij}$$

– observables $\longrightarrow$ Lorentz layer

$$\tilde{k}_j \xrightarrow{\text{LoLa}} \hat{k}_j = \begin{pmatrix} m^2(\tilde{k}_j) \\ p_T(\tilde{k}_j) \\ \vdots \end{pmatrix}$$

$\Rightarrow$ Learn Minkowski metric

g =diag(0.99±0.02, −1.01±0.01, −1.01±0.02, −0.99±0.02)



Jet classification done

SciPost Physics Submission

The Machine Learning Landscape of Top Taggers

G. Kasieczka (ed)[1], T. Plehn (ed)[2], A. Butter[2], K. Cranmer[3], D. Debnath[4], B. M. Dillon[5], M. Fairbairn[6], D. A. Faroughy[5], W. Fedorko[7], C. Gay[7], L. Gouskos[8], J. F. Kamenik[5,9], P. T. Komiske[10], S. Leiss[1], A. Lister[7], S. Macaluso[3,4], E. M. Metodiev[10], L. Moore[11], B. Nachman,[12,13], K. Nordström[14,15], J. Pearkes[7], H. Qu[8], Y. Rath[16], M. Rieger[16], D. Shih[4], J. M. Thompson[2], and S. Varma[6]

1 Institut für Experimentalphysik, Universität Hamburg, Germany
2 Institut für Theoretische Physik, Universität Heidelberg, Germany
3 Center for Cosmology and Particle Physics and Center for Data Science, NYU, USA
4 NHECT, Dept. of Physics and Astronomy, Rutgers, The State University of NJ, USA
5 Jozef Stefan Institute, Ljubljana, Slovenia
6 Theoretical Particle Physics and Cosmology, King's College London, United Kingdom
7 Department of Physics and Astronomy, The University of British Columbia, Canada
8 Department of Physics, University of California, Santa Barbara, USA
9 Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia
10 Center for Theoretical Physics, MIT, Cambridge, USA
11 CP3, Universitéxx Catholique de Louvain, Louvain-la-Neuve, Belgium
12 Physics Division, Lawrence Berkeley National Laboratory, Berkeley, USA
13 Simons Inst. for the Theory of Computing, University of California, Berkeley, USA
14 National Institute for Subatomic Physics (NIKHEF), Amsterdam, Netherlands
15 LPTHE, CNRS & Sorbonne Université, Paris, France
16 III. Physics Institute A, RWTH Aachen University, Germany

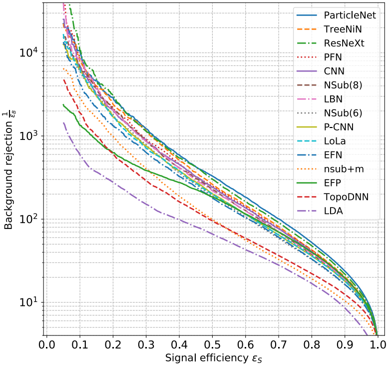
gregor.kasieczka@uni-hamburg.de
plehn@uni-heidelberg.de

July 24, 2019

Abstract

Based on the established task of identifying boosted, hadronically decaying top quarks, we compare a wide range of modern machine learning approaches. Unlike most established methods they rely on low-level input, for instance calorimeter output. While their network architectures are vastly different, their performance is comparatively similar. In general, we find that these new approaches are extremely powerful and great fun.

Content



– many networks successful [ask Martin]

– which direction to follow?

⇒ Error bars, maybe? [Nachman 1909.03081]

Jet classification with error bars

Jet-by-jet uncertainties

– $(60\pm??)\%$ top, uncertainty from training

– probability for test event $p(c^*|C)$ [classifier output C, network ω]

$$p(c^*|C) = \int d\omega \; p(c^*|\omega, C) \; p(\omega|C) = \int d\omega \; p(c^*|\omega, C) \; q(\omega)$$

– loss function from minimizing Kullbeck-Leibler divergence [Bayes' theorem]

$$\begin{aligned} \text{KL}[q(\omega), p(\omega|C)] &= \int d\omega \; q(\omega) \; \log \frac{q(\omega)}{p(\omega|C)} \\ &= \int d\omega \; q(\omega) \; \log \frac{q(\omega)p(C)}{p(C|\omega)p(\omega)} \\ &= \underbrace{\text{KL}[q(\omega), p(\omega)]}_{\text{L2-regularization}} + \underbrace{\log p(C) \int d\omega \; q(\omega)}_{\text{normalization of } q\text{, irrelevant}} - \underbrace{\int d\omega \; q(\omega) \log p(C|\omega)}_{\text{likelihood, maximized}} \end{aligned}$$

$$\Rightarrow L = \text{KL}[q(\omega), p(\omega)] - \int d\omega \; q(\omega) \log p(C|\omega)$$

Jet classification with error bars

Jet-by-jet uncertainties

- (60$\pm$??)% top, uncertainty from training
- probability for test event $p(c^*|C)$ [classifier output C, network ω]

$$p(c^*|C) = \int d\omega \; p(c^*|\omega, C) \; p(\omega|C) = \int d\omega \; p(c^*|\omega, C) \; q(\omega)$$

- loss function from minimizing Kullbeck-Leibler divergence [Bayes' theorem]

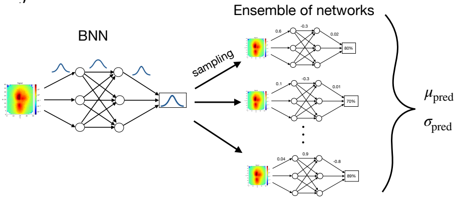
$$\begin{aligned} \text{KL}[q(\omega), p(\omega|C)] &= \int d\omega \; q(\omega) \; \log \frac{q(\omega)}{p(\omega|C)} \\ &= \int d\omega \; q(\omega) \; \log \frac{q(\omega)p(C)}{p(C|\omega)p(\omega)} \\ &= \underbrace{\text{KL}[q(\omega), p(\omega)]}_{\text{L2-regularization}} + \underbrace{\log p(C) \int d\omega \; q(\omega)}_{\text{normalization of } q\text{, irrelevant}} - \underbrace{\int d\omega \; q(\omega) \log p(C|\omega)}_{\text{likelihood, maximized}} \end{aligned}$$

$$\Rightarrow L = \text{KL}[q(\omega), p(\omega)] - \int d\omega \; q(\omega) \log p(C|\omega)$$

$\Rightarrow$ sample ω to extract $(\mu_\text{pred}, \sigma_\text{pred})$

check prior independence

check frequentist many-networks



Jet classification with error bars

Jet-by-jet uncertainties

- $(60\pm??)\%$ top, uncertainty from training
- probability for test event $p(c^*|C)$ [classifier output C, network ω]

$$p(c^*|C) = \int d\omega\; p(c^*|\omega, C)\; p(\omega|C) = \int d\omega\; p(c^*|\omega, C)\; q(\omega)$$

$\Rightarrow$ sample ω to extract $(\mu_\text{pred}, \sigma_\text{pred})$

Complication with classification

- sigmoid to map on closed interval $[0, 1]$

$$\text{Sigmoid}(x) = \frac{e^x}{1+e^x}$$

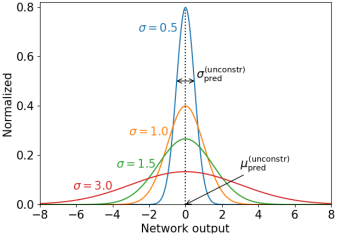
- predictive mean

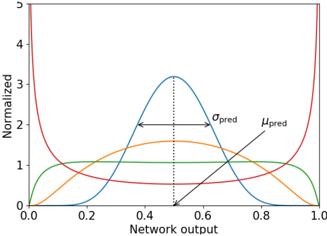
$$\mu_\text{pred} = \int_{-\infty}^{\infty} d\omega\; \text{Sigmoid}(\omega)\; G_{\mu,\sigma}(\omega)$$
$$= \int_0^1 dx\; \frac{x}{x(1-x)}\; G_{\mu,\sigma}\left(\log \frac{x}{1-x}\right) \in [0, 1]$$

- predictive standard deviation

$$\sigma_\text{pred} \approx \mu_\text{pred}\,(1-\mu_\text{pred})\;\sigma_\text{pred}^\text{(unconstr)}$$

$\Rightarrow$ Additional complication...



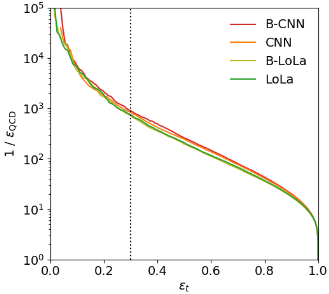


Statistics & systematics

Training statistics [Bollweg, Haussmann, Kasieczka, Luchmann, TP, Thompson]

– Bayesian version of DeepTop and LoLa

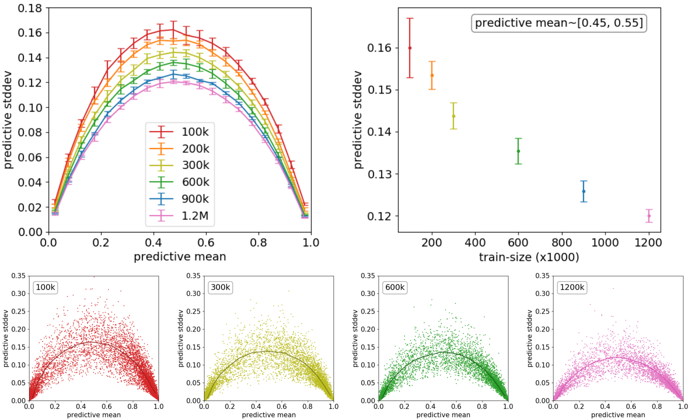
– similar performance as deterministic network
training time somewhat increased



Statistics & systematics

Training statistics [Bollweg, Haussmann, Kasieczka, Luchmann, TP, Thompson]

- Bayesian version of DeepTop and LoLa
- similar performance as deterministic network
  training time somewhat increased
- correlation between μ_{pred} and σ_{pred} [toy network, 10k jets]
- increasing training statistics [parabola from closed interval output]



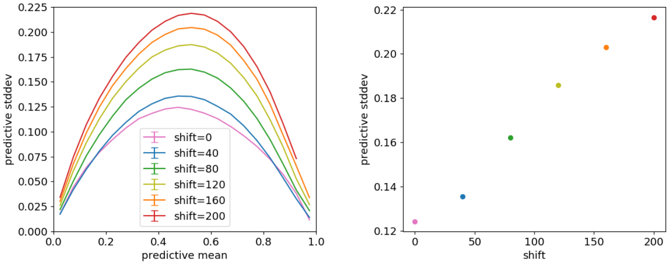
Statistics & systematics

Training statistics [Bollweg, Haussmann, Kasieczka, Luchmann, TP, Thompson]

- Bayesian version of DeepTop and LoLa
- similar performance as deterministic network
  training time somewhat increased
- correlation between μ_{pred} and σ_{pred} [toy network, 10k jets]
- increasing training statistics [parabola from closed interval output]

Noise/pile-up

- increasing pile-up, stable [LoLa, ordered constituents]



Statistics & systematics

Training statistics [Bollweg, Haussmann, Kasieczka, Luchmann, TP, Thompson]

– Bayesian version of DeepTop and LoLa

– similar performance as deterministic network
training time somewhat increased

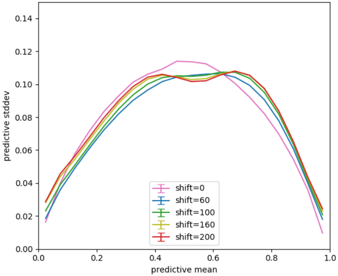
– correlation between μ_{pred} and σ_{pred} [toy network, 10k jets]

– increasing training statistics [parabola from closed interval output]

Noise/pile-up

– increasing pile-up, stable [LoLa, ordered constituents]

– increasing pile-up, unstable [DeepTop, jet image]



Statistics & systematics

Training statistics [Bollweg, Haussmann, Kasieczka, Luchmann, TP, Thompson]

– Bayesian version of DeepTop and LoLa

– similar performance as deterministic network
  training time somewhat increased

– correlation between μ_{pred} and σ_{pred} [toy network, 10k jets]

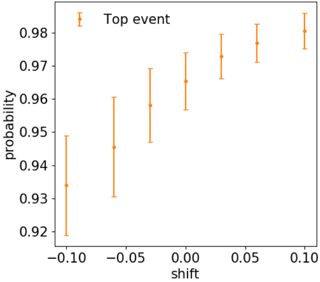
– increasing training statistics [parabola from closed interval output]

Jet energy scale

– systematics effect in test sample

1– shift of hardest constituent

– adversarial example: hierarchical subjets = top



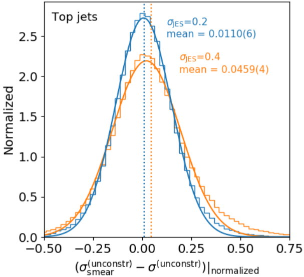
Statistics & systematics

Training statistics [Bollweg, Haussmann, Kasieczka, Luchmann, TP, Thompson]

– Bayesian version of DeepTop and LoLa
– similar performance as deterministic network
training time somewhat increased
– correlation between μ_{pred} and σ_{pred} [toy network, 10k jets]
– increasing training statistics [parabola from closed interval output]

Jet energy scale

– systematics effect in test sample
1– shift of hardest constituent
– adversarial example: hierarchical subjets = top
2– uncorrelated shift of all constituents
– tiny degradation for signal
⇒ Better control needed



Jet measurements with error bars

Regression: measure $p_{T,t}$ [Kasieczka, Luchmann, TP (soon)]

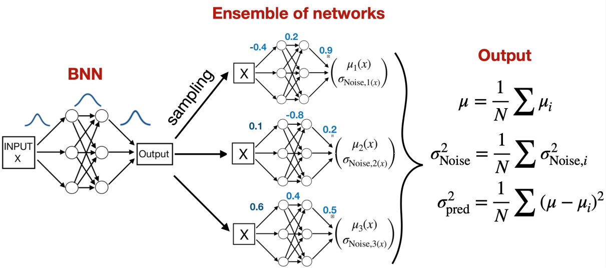
– effect of noisy and size-limited data separated

σ_{pred}: limited training sample

σ_{noise}: statistical behavior of training data [Gaussian likelihood]

$$\log p(C|\omega) \to \log p(C|\mu, \sigma_{\text{noise}}) = \frac{(C-\mu)^2}{2\sigma_{\text{noise}}^2} + \frac{1}{2}\log \sigma_{\text{noise}}^2 + \text{const}$$

$$\sigma_{\text{tot}}^2 = \sigma_{\text{pred}}^2 + \sigma_{\text{noise}}^2 \quad \text{[all Gaussian]}$$



Jet measurements with error bars

Regression: measure $p_{T,t}$ [Kasieczka, Luchmann, TP (soon)]

– effect of noisy and size-limited data separated

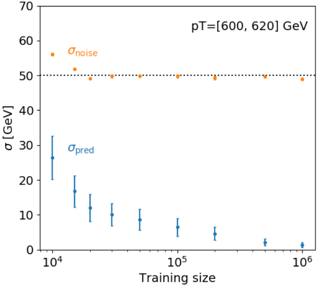
σ_{pred}: limited training sample

σ_{noise}: statistical behavior of training data [Gaussian likelihood]

$$\log p(C|\omega) \to \log p(C|\mu, \sigma_{\text{noise}}) = \frac{(C-\mu)^2}{2\sigma^2_{\text{noise}}} + \frac{1}{2}\log \sigma^2_{\text{noise}} + \text{const}$$

$\sigma^2_{\text{tot}} = \sigma^2_{\text{pred}} + \sigma^2_{\text{noise}}$ [all Gaussian]

– sample size dependence [statistics saturating]



Jet measurements with error bars

Regression: measure $p_{T,t}$ [Kasieczka, Luchmann, TP (soon)]

– effect of noisy and size-limited data separated

σ_{pred}: limited training sample

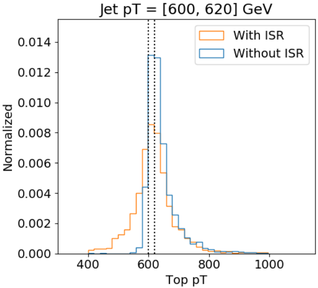
σ_{noise}: statistical behavior of training data [Gaussian likelihood]

$$\log p(C|\omega) \to \log p(C|\mu, \sigma_{\text{noise}}) = \frac{(C-\mu)^2}{2\sigma^2_{\text{noise}}} + \frac{1}{2}\log\sigma^2_{\text{noise}} + \text{const}$$

$\sigma^2_{\text{tot}} = \sigma^2_{\text{pred}} + \sigma^2_{\text{noise}}$ [all Gaussian]

– sample size dependence [statistics saturating]

– comparison with $p_{T,t}$ vs $p_{T,j}$



Jet measurements with error bars

Regression: measure $p_{T,t}$ [Kasieczka, Luchmann, TP (soon)]

– effect of noisy and size-limited data separated

σ_{pred}: limited training sample

σ_{noise}: statistical behavior of training data [Gaussian likelihood]

$$\log p(C|\omega) \to \log p(C|\mu, \sigma_{\text{noise}}) = \frac{(C-\mu)^2}{2\sigma^2_{\text{noise}}} + \frac{1}{2} \log \sigma^2_{\text{noise}} + \text{const}$$

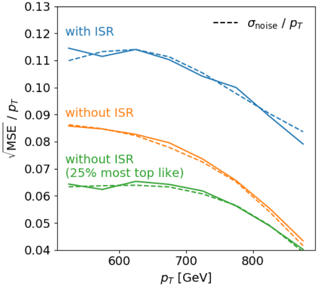
$$\sigma^2_{\text{tot}} = \sigma^2_{\text{pred}} + \sigma^2_{\text{noise}}$$ [all Gaussian]

– sample size dependence [statistics saturating]

– comparison with $p_{T,t}$ vs $p_{T,j}$

– dependence on ISR and top-ness

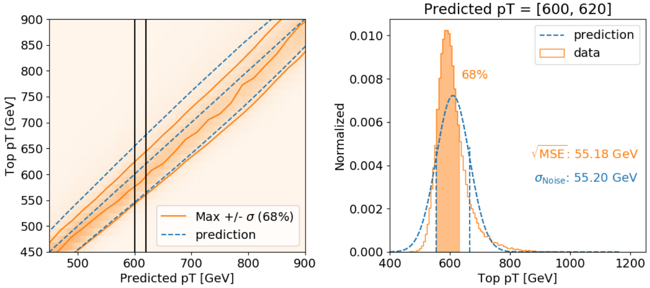
⇒ Accurate error estimate



Jet calibration

Calibration means error propagation

– training on smeared data??
  better: training with smeared labels [p_T measured elsewhere, with error]
– Gaussian noise over $p_{T,t}$ label [2, 4, 6 . . . 10%]
– distribution of extracted $p_{T,t}$
  correlation extending to error bars
  slice with expected non-Gaussian tail from QCD radiation

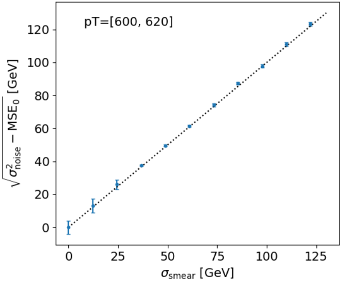


Jet calibration

Calibration means error propagation

- training on smeared data??
  better: training with smeared labels [p_T measured elsewhere, with error]
- Gaussian noise over $p_{T,t}$ label [2, 4, 6 . . . 10%]
- distribution of extracted $p_{T,t}$
  correlation extending to error bars
  slice with expected non-Gaussian tail from QCD radiation
- effect from calibration uncertainty alone
  trace label smearing to network output
  making sense of σ_{noise}

$\Rightarrow$ Works!



Looking into the future

Machine learning a great tool box...

...LHC physics really is big data

...imagine recognition is a starting point

...performance in tagging solved

...time for (more) interesting questions

...Bayesian networks do uncertainties better than current methods