Deep-learned Top Tagging using Lorentz Invariance and Nothing Else

Anja Butter

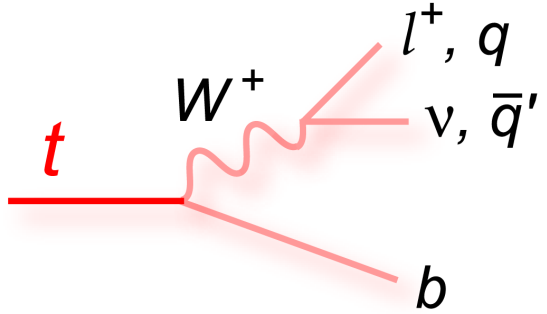
ITP, Universität Heidelberg

Based on arXiv:1707.08966
with Gregor Kasieczka, Tilman Plehn and Michael Russell

Top Tagging

- top quark is crucial for new physics and SM searches
  - large Yukawa coupling, Hierarchy problem

- Identify hadronically decaying top quarks from QCD jets
- Large p_T leads to boosted topology
  - $\rightarrow$ all decay products contained in one jet



https://www-d0.fnal.gov

- Standard approach: top tagging algorithms using mass drop, 3-prong structure
- Aim: improve performance with neural networks based on images/ Lorentz vectors

Outline

- Machine learning
- Image based top tagging
- Neural network using Lorentz vectors
  - + Includes tracking information
  - + Increases performance for strongly boosted tops

Machine learning

Definition by Arthur Samuel (1959)

The field of study that gives computers the ability to learn without being explicitly programmed.

Machine learning for top tagging

Definition by Tom Mitchell (1997)

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

T = distinguish between a top and a gluon jet

E = experience in distinguishing between the two cases

P = efficiency (mistagging rate)

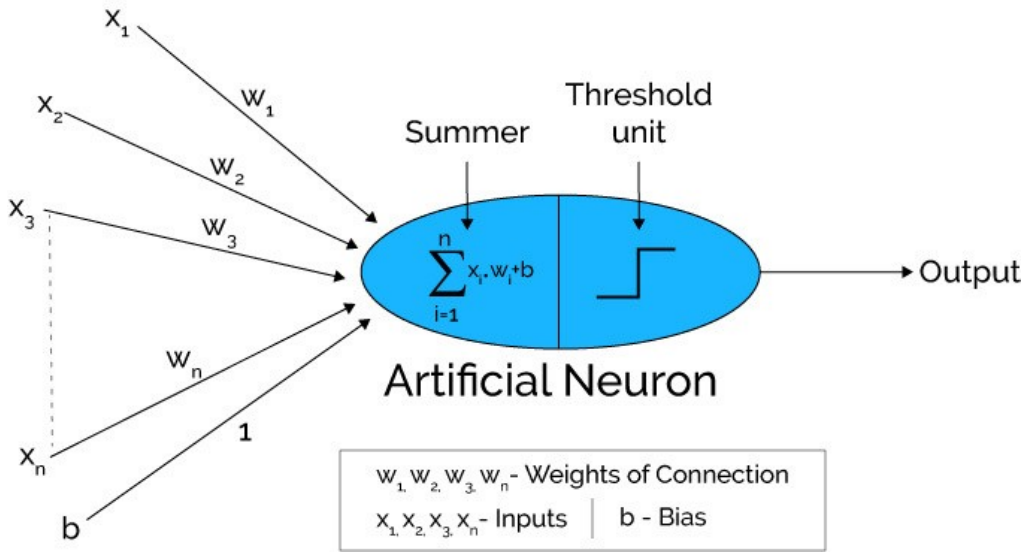
Some name-dropping

- Supervised Learning
  - Regression - Predict functional dependency, eg. probability of rain given temperature, pressure, etc.
  - Classification - top or QCD jet
- Unsupervised Learning - categorizing genes that look similar

Some typical techniques:

- Decision trees
- Support vector machines
- Neural networks

Neuron

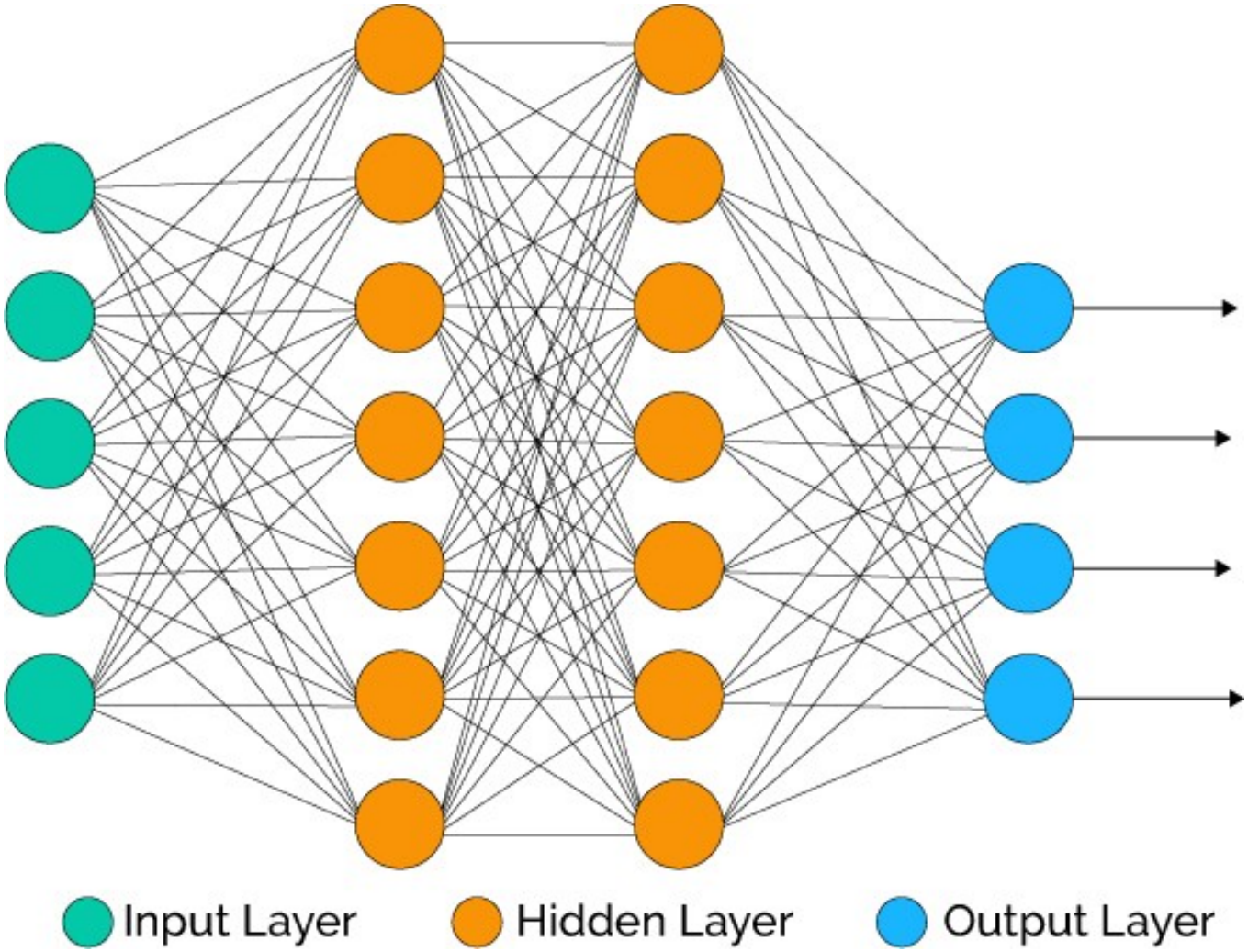


https://hackernoon.com/overview-of-artificial-neural-networks-and-its-applications-2525c1addff7

Popular activation functions:

- Binary step: $\Theta(x)$
- Re(ectified) L(inear) U(nit): $\Theta(x)x$
- Logistic: $\dfrac{1}{1+e^x}$

Neural network

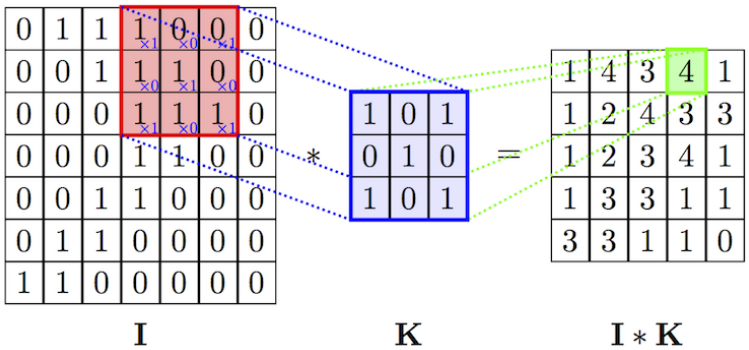


https://hackernoon.com/overview-of-artificial-neural-networks-and-its-applications-2525c1addff7

Types of layers

- Previous slide: Fully connected Layer

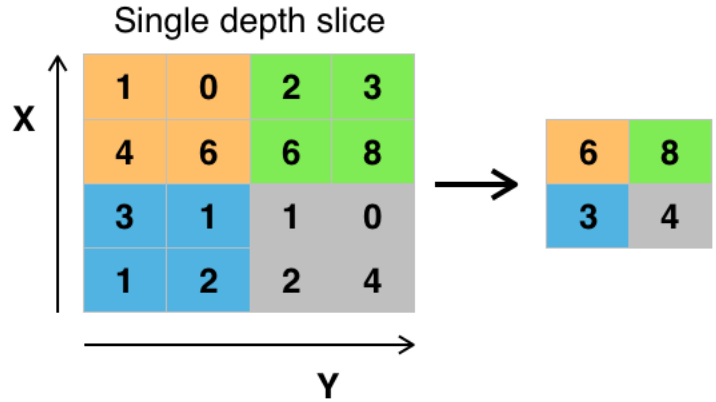
- Convolutional Layer



https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html

The network learns the parameters of K

- (Max)Pooling Layer



https://commons.wikimedia.org/wiki/File:Max_pooling.png

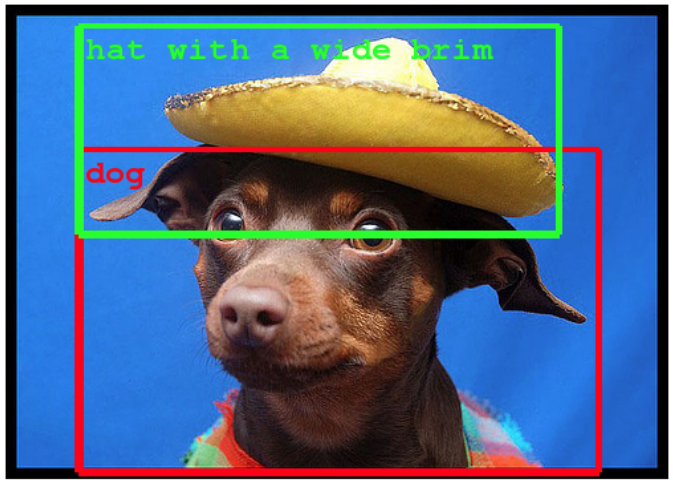
- Flattening layer $(n \times n) \rightarrow n^2$

How does the neural network learn?

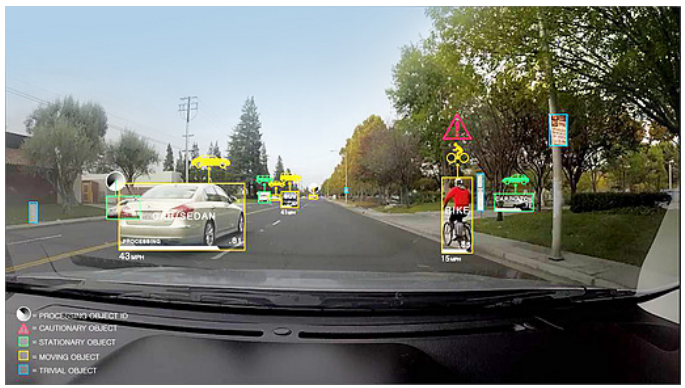
- Input: Labeled dataset
- Set up the neural network
  - Choose layers
  - Choose activation function
- Training
  - Train weights w_{ij}
  - Minimize cost function: eg. $E = \frac{1}{2}(o_i - t_i)^2$ with output o_i and true value t_i
  - Backpropagation: Adjust w_{ij} via
    $$\Delta w_i = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial net} \frac{\partial net}{\partial w_{ij}}$$
- Test performance with test sample to avoid overtraining

Applications

- AlphaGo Zero
- Face/Voice recognition
- Translation
- Object classification
- Text generation
- Autonomous driving
- Predicting earthquakes
- Art (Deep Dreaming)

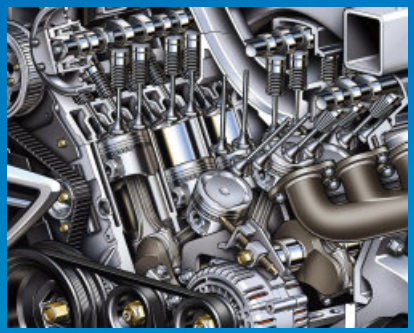


https://machinelearningmastery.com/inspirational-applications-deep-learning/



http://articles.sae.org/13996/


DeepDreamGenerator



https://www.getyourguide.com


DeepDreamGenerator

Back to Physics

Reminder:
We want to distinguish top from QCD jet events

Technical setup

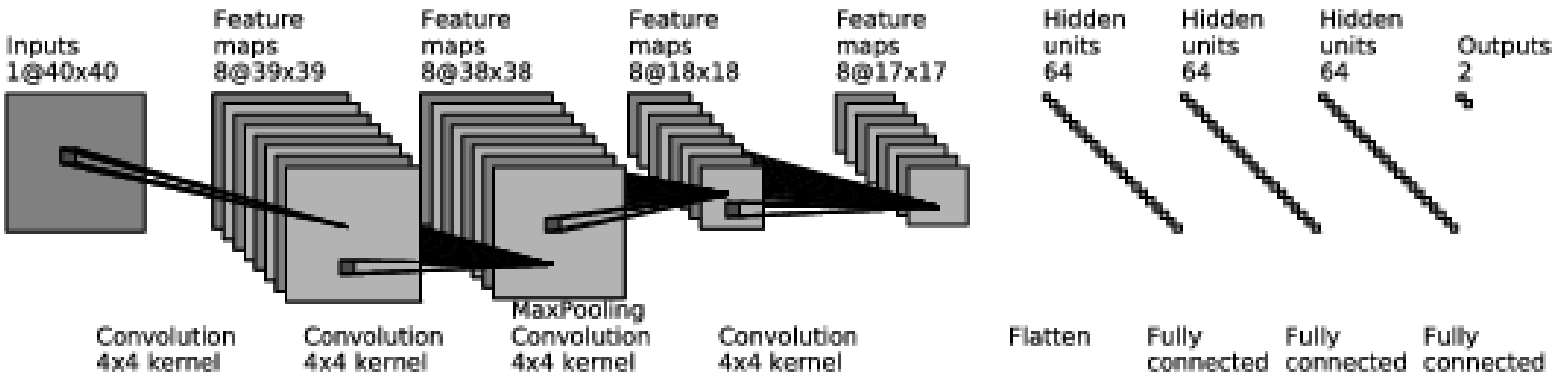
- 14 TeV hadronic ttbar vs QCD, both simulated with Pythia 8 + Delphes 3
- Cluster with FastJet3 anti-k_T with $R = 1.5$ (smooth shape)
- Re-cluster constituents with R=1.5 C/A jet
- $|\eta_{fat}| < 1.0$, $p_{T,fat} = 350 \ldots 450$ GeV
- Input: images, calorimeter E_T in azimuthal vs rapidity plane (5° in ϕ, 0.1 in η)

Deep-learning Top Taggers or The End of QCD?

arxiv:1701.08784

Gregor Kasieczka, Tilman Plehn, Michael Russell, Torben Schell

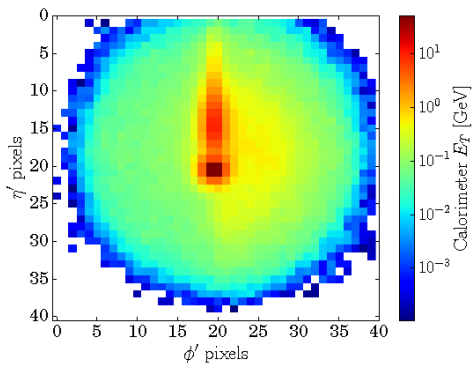
- Network architecture



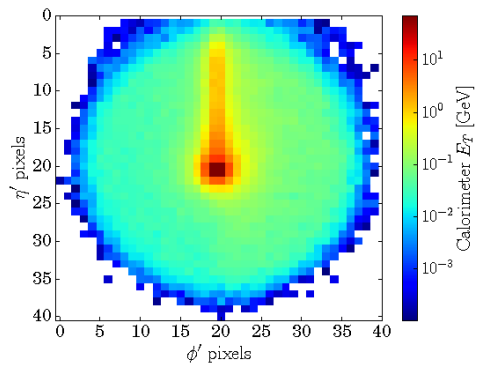
- Optimize hyperparameters (size and number of kernels, layers, nodes, etc)

What does the network learn?

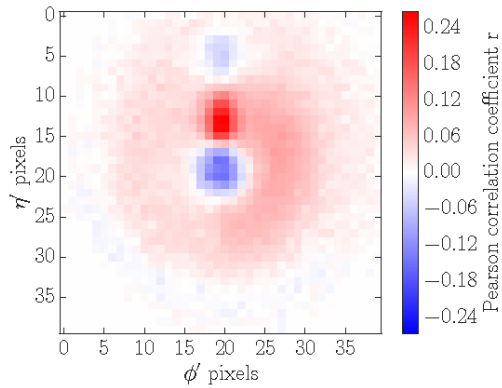
- Consider preprocessed images:
  - center maximum
  - rotate second maximum to 12 o'clock
  - flip third maximum to right side
  - overlay of multiple images



signal



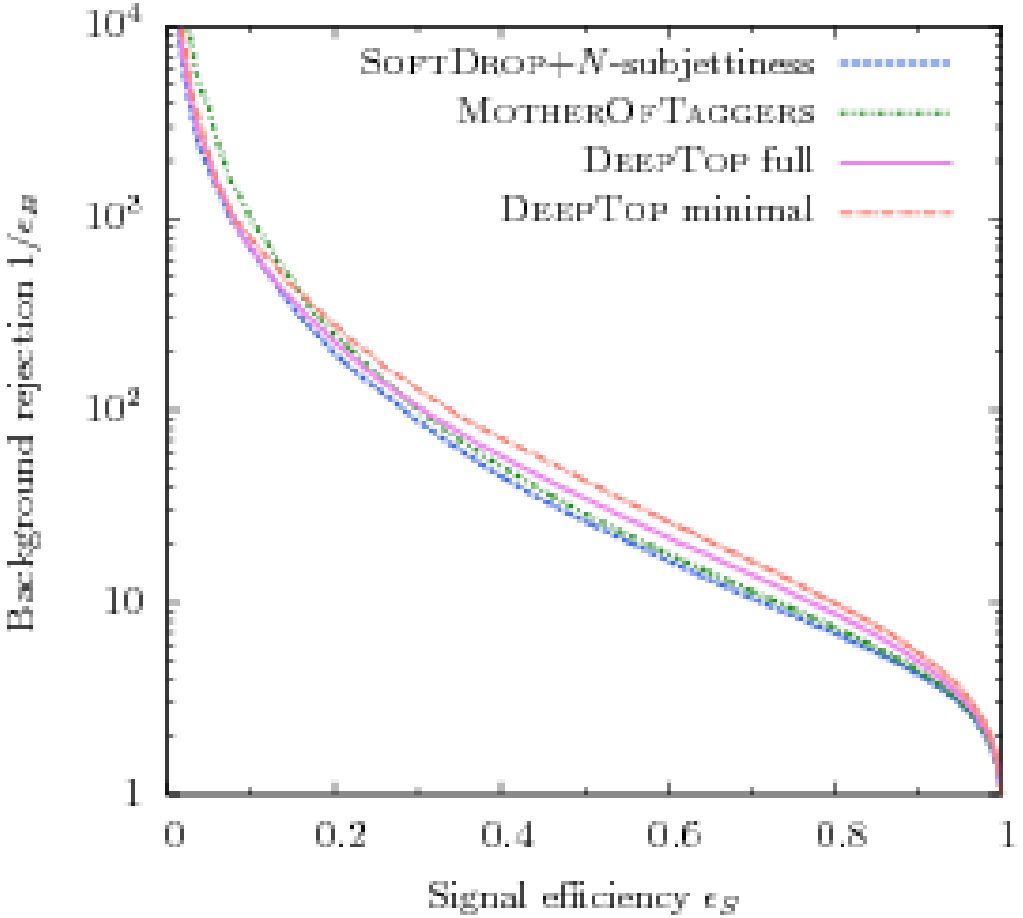
background



Pearson correlation coefficient r_{ij}

$$r_{ij} = \frac{\sum_{\text{images}}(x_{ij} - \bar{x}_{ij})(y - \bar{y})}{\sqrt{\sum_{\text{images}}(x_{ij} - \bar{x}_{ij})^2}\sqrt{\sum_{\text{images}}(y - \bar{y})^2}}$$

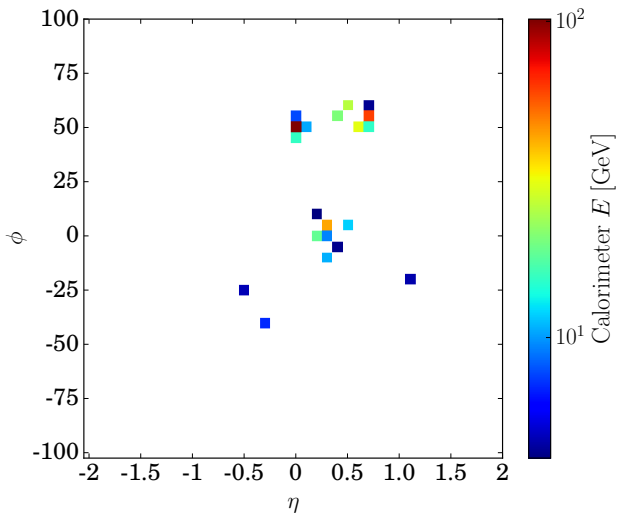
Result



- The neural network outperforms the QCD based taggers
- Mother of Taggers: BDT trained on standard tagging variables like masses and subjettiness

Room for improvement

- Coarse information (most bins are empty)
- No tracking information
  - Different resolution of calorimeter and tracking system
  - Tracking would lead to too many, too sparsely distributed pixels
- No physics!



New input

→ Use Lorentz vectors instead of images (from calorimeter or particle flow objects)

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

- Sorted by p_T
- Take into account vector properties with new Layer structure → CombinationLayer and LorentzLayer based on Minkowski metric

CoLa

Inspired by jet algorithms to reconstruct substructures:

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

with

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

1. Sum of all momenta
2. Original momenta
3. Trainable linear combination of Lorentz vectors

LoLa

Transform Lorentz vectors into physics motivated objects.

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

with

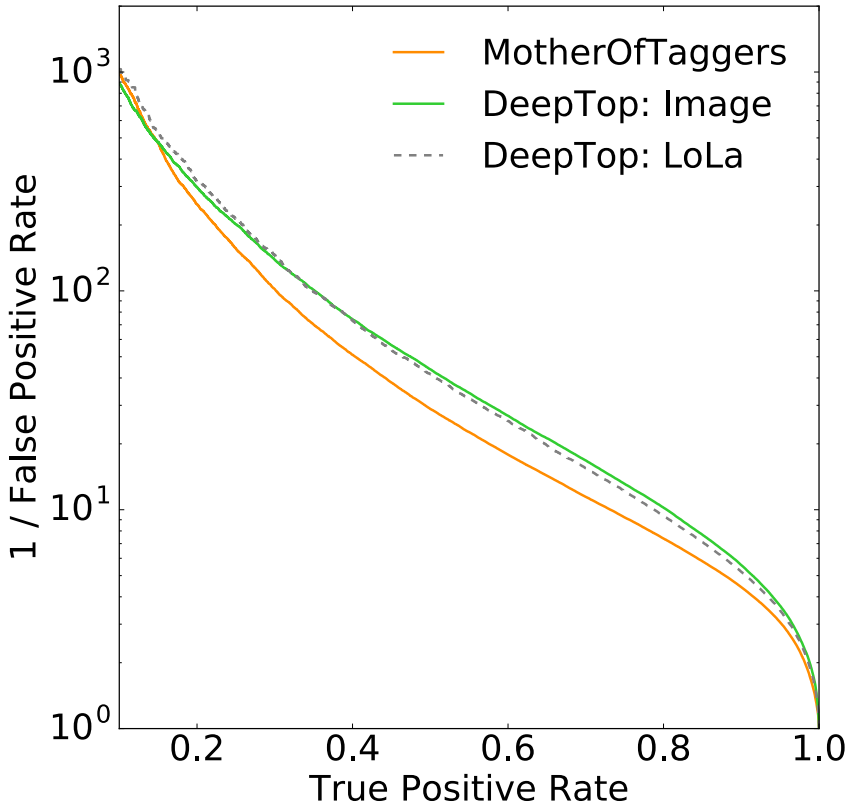
$$d_{jm}^2 = (\tilde{k}_j - \tilde{k}_m)_\mu\, \eta^{\mu\nu}\, (\tilde{k}_j - \tilde{k}_m)_\nu$$

- Use sum and minimum over index m
- Flexible list of objects, easy to extend

Framework

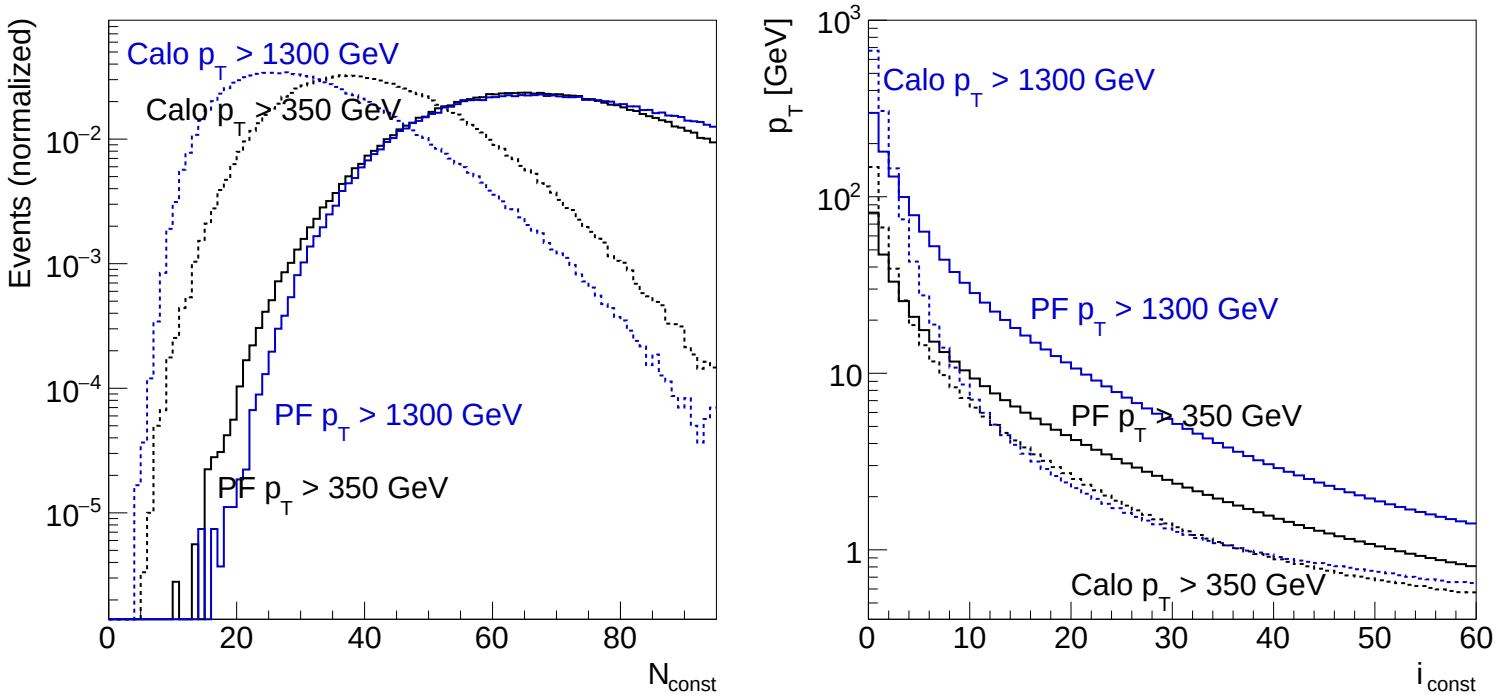
- Input: Calorimeter / Particle Flow
- Preprocessing:
  - Pythia8 + Delphes3
  - FastJet3 anti-k_T with $R = 1.5$
  - $|\eta_{fat}| < 1.0$, $p_{T,fat} = 350 \ldots 450$ GeV or $p_{T,fat} = 1300 \ldots 1400$ GeV
- CoLa, LoLa, 2 fully connected layers
- 180 000 training events, 60 000 test and 60 000 validation events

LoLa vs Image



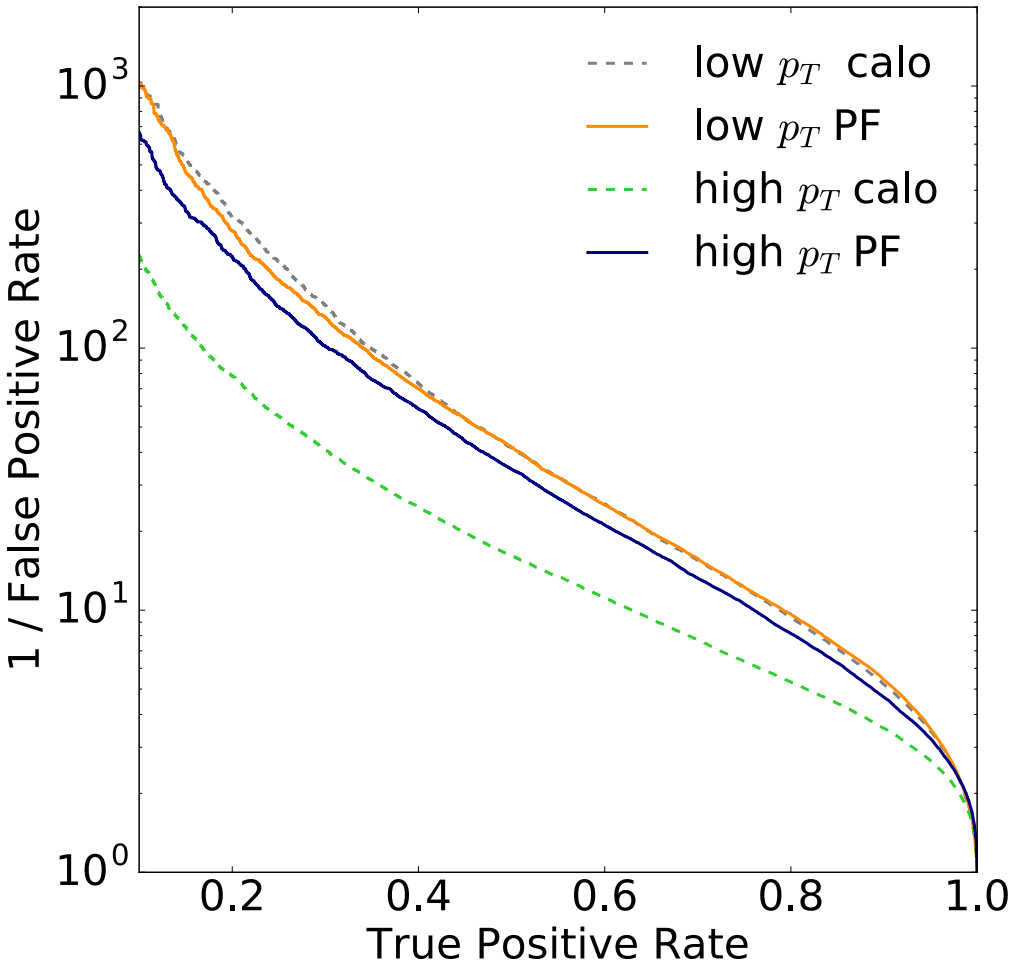
- Performance using only calorimeter information for $p_{T,fat} = 350 \ldots 450$ GeV
- Same performance as image based convolutional neural network, less complex
  [factor 3 to 8 less weights, factor 10 to 20 less inputs]

p_T dependance of jet constituents



- Tagging performance saturates with leading $N = 20$ constituents

Calo vs PF at low and high p_T



- For high p_T jets the additional tracking information in the particle flow object becomes crucial

Learning the Minkowski metric :)

- Requiring a diagonal metric to determine m^2 and d^2_{jm} with freely trainable entries yields *upon normalization*:

$$\eta = \text{diag}(\quad 0.99 \pm 0.02,$$
$$-1.01 \pm 0.01, -1.01 \pm 0.02, -0.99 \pm 0.02).$$

- The error is determined using 5 independent runs.

$\rightarrow$ recover sign difference (+,-,-,-)

$\rightarrow$ recover equal absolute values

Conclusion & Outlook

- Machine Learning is fun and efficient
  - $\rightarrow$ Image based approach easily outperforms standard taggers
- New, fast, flexible DeepTopLoLa tagger
- LoLa is competitive with image based approach
- Large performance gain of PF objects for strongly boosted top quarks with respect to calorimeter based objects
- For the future: Flexibility allows to easily include new features like *b*-tagging