The (very) basics

- Hadronically decaying top quarks
- Contained in one (large-R) jet
- How to distinguish from light quark/gluon jets?

![Image of hadronically decaying top quarks and jet reconstruction](image)

**Figure 1:** Distance between reconstructed jet and closest generated top quark (left). Jet reconstruction efficiency (right) as function of the generated top quark $p_T$. The efficiency is defined as the fraction of top quarks for which a reconstructed jet with $p_T > 200$ GeV can be found within $\Delta R < 1.2$ ($\Delta R < 0.6$) for CA15 (AK8) jets. Superimposed is the fraction of merged top quarks as function of $p_T$ for the two thresholds used: 0.8 (0.6) at low (high) boost. All distributions are made using hadronically decaying top quarks with $p_T > 200$ GeV.

Matching particles to the generator-level is performed by selecting the closest reconstructed jet in $\Delta R$ to a truth-level parton or top quark. The maximum distance depends on the jet size and corresponds to $\Delta R < 1.2$ ($\Delta R < 0.6$) for $R = 1.5$ ($R = 0.8$) jets. These values approximately correspond to the minima of the $\Delta R$ distribution between jets and truth-level top quarks, as shown in Fig. 1 (left).

The jet cone sizes are at typical values used in top quark reconstruction [24]. Fig. 1 (right) shows the efficiency with which a reconstructed jet with $p_T > 200$ GeV is found within $\Delta R < 1.2$ ($\Delta R < 0.6$), for CA15 (AK8) jets, with respect to a hadronically decaying top quark as function of the quark $p_T$. The efficiency reaches a plateau close to 100% at about 280 GeV (380 GeV) for jets with $R = 1.5$ ($R = 0.8$). In fact, at low $p_T$ the larger cone size allows for the collection of all decay products from the top quark. In contrast, at high $p_T$ the top jet is more collimated and the smaller cone size reduces the amount of additional radiation clustered into the jet.

To evaluate the properties of top-tagging methods in an unbiased way over a large phase space, taggers are studied in exclusive regions of $p_T$, defined by the transverse momentum of the matched truth-level parton (top quark, light quark, or gluon). Then, a weight is assigned to each truth-level object such that the resulting distributions in $p_T$- and $h$-space are approximately flat. As the distribution of top-jet candidates becomes more central with higher $p_T$ values, we tighten different selection criteria on $|h|$ as the $p_T$ increases. Finally, to avoid the study of incompletely merged top quarks (i.e. top quarks for which not all decay products are contained in a jet), we restrict our selection to top quarks where the two quarks from the W boson decay and the b quark have a maximum distance from the top quark direction ($\Delta R_{(t, q)}$)).

**CMS PAS JME-15-002**
Overview

• QCD Inspired Jet Substructure Variables
• Top Tagging Algorithms
• Machine Learning Approaches
Jet Substructure Variables
Jet Grooming

- Remove
  - soft radiation
  - underlying event
  - pile up
- from jet to access top mass

Prior to the application of this procedure, no distinct features are present in the jet mass distribution, whereas afterwards, a clear mass peak that corresponds to the jet mass for comparison. The technical details of this grooming algorithm are described in section 3.3.1. Single-jet invariant mass distribution for Cambridge–Aachen (C/A) ATLAS jets with $R=1.2$, 600 simulated events containing highly boosted hadronically decaying $Z bosons before and after the application of a grooming procedure referred to as mass-drop filtering with respect to the ungroomed large-$R$ jets for comparison. The normalization of the groomed distribution includes the pile-up events are also included. The often dominant multi-jet background and the heavy-particle decay, which increases physics in recent studies of the phenomenological implications of such tools in searches for new.
Trimming

• Recluster constituents with \( R=0.2 \)

• Remove subjets with less than 5% of jet \( p_T \)

• ATLAS Default
mMDT

Softdrop

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

- Find hard substructure using step-wise unclustering
- No pure soft divergences
- Analytically calculable to high precision

\[
\frac{1}{z} \quad \text{Soft} \quad \text{Soft-Collinear}
\]

\[
\log \frac{1}{z_{\text{cut}}} \quad \beta > 0 \quad \beta = 0 \quad \beta < 0
\]

\[
\log R_0 / \theta
\]

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

**quark jets (Pythia 6 MC)**

\[
m [\text{GeV}], \text{for } p_t = 3 \text{ TeV}, R=1
\]

**gluon jets (Pythia 6 MC)**

\[
m [\text{GeV}], \text{for } p_t = 3 \text{ TeV} R=1
\]
\[ \tau_N = \frac{1}{d_0} \sum_k p_{T,k} \min \{ \Delta R_{1,k}, \Delta R_{2,k}, \cdots, \Delta R_{N,k} \} \]

- n-subjettiness: Small when compatible with n-prong substructure
- Used for top-tagging: \( \frac{\tau_3}{\tau_2} \)
- Recent ideas:
  - Dichroic n-subjettiness = ratio of n-subjettiness with different grooming (JHEP 1703 022)
  - Use for jet clustering (XCones: JHEP 1511 07)
ECFs

\[ N_i^{(\beta)} = \frac{2e_i^{(\beta)}}{\left(e_i^{(\beta)}\right)^2} \]

\[ N_3 = \frac{2e_4^{(\beta)}}{\left(e_3^{(\beta)}\right)^2} \]

top tagging

\[ v e_n^{(\beta)} = \sum_{1 \leq i_1 < i_2 < \ldots < i_n \leq n_J} z_{i_1} z_{i_2} \cdots z_{i_n} \prod_{m=1}^{v} \min_{s < t \in \{i_1, i_2, \ldots, i_n\}} \left\{ \theta_{st}^{(\beta)} \right\} \]

- Energy correlation functions
- Replacing n-subjettiness for heavy-resonance identification
- Wide range of other uses
Inclusive
Top Taggers
**HEPTopTagger (V2)**

- **OptimalR-Algorithm:**
  - Start with C/A, $R=1.5$ seed fat-jet
  - Perform unclustering to identify *small fat-jets* with $R=0.5$ to $R=1.5$ (in steps of 0.1) and run HEPTopTagger on each of them
  - Calculate: $R_{\text{min}} = \text{Smallest cone size for which the mass differs by less than 20\% from the mass at } R=1.5$
  - Get: $R_{\text{opt, calc}} (p_T)$. Result of fitting $R_{\text{opt}}$ as function of $p_T$ for signal jets
  - Output observables:
    - Top candidate mass: $m(R=R_{\text{opt}})$
    - $W / \text{top mass ratio: } f_W(R=R_{\text{opt}})$
    - $R_{\text{opt}}$ difference: $R_{\text{opt}} - R_{\text{opt, calc}} (p_T)$
HOTVR

- Heavy Object Tagger with Variable R
- Jet algorithm and heavy resonance tagger
- Jet clustering with variable R
- Removal of soft clusters during clustering (mass jump)
  - Stable jet mass distribution
- Tagger stable over large pT range
- Usable for different heavy hadronic resonances

\[ \eta \]

\[ \phi \]

\[ 200 < p_T < 400 \text{ GeV} \]
\[ N_{\text{sub}} \geq 2 \quad N_{\text{sub}} \geq 3 \]
- \( \text{t}\bar{\text{t}} \)
- \( \text{t}\bar{\text{t}} \)
- QCD
- QCD

\[ 600 < p_T < 800 \text{ GeV} \]
\[ N_{\text{sub}} \geq 2 \quad N_{\text{sub}} \geq 3 \]
- \( \text{t}\bar{\text{t}} \)
- \( \text{t}\bar{\text{t}} \)
- QCD
- QCD
**BEST**

- Boosted Event Shape Tagger (BEST)
- Boost jet constituents individually into each reference frame corresponding to particle origin hypothesis (t, W, Z, H)
- Calculate angular distributions in boosted frame
- Fox-Wolfram Moments, Sphericity, Aplanarity, Isotropy, Thrust, ..
- Use NN for simultaneous classification
Shower Deconstruction

- Similar to the matrix element method
- Assigns signal and background probabilities to individual jets instead of events
- Microjets are built by clustering the constituents of the fat jet into smaller jets with the $k_T$-algorithm
- Variable $X$: probability quotient, a set of microjets in a fat-jet was created by the decay of a top quark, divided by the probability that it was created by light quarks and gluons:

\[ X(p_N) = \frac{P(p_N|S)}{P(p_N|B)} \]

![Shower deconstruction diagram](image)

**Top**

**QCD**

Finding top quarks with shower deconstruction
DE Soper, M Spannowsky
PRD 87 054012

Finding physics signals with event deconstruction
DE Soper, M Spannowsky
PRD 89 094005
Deep Learning
Different Types of Problems

• What do we want to learn?
  • Classification
  • Generation
  • Regression

• What do we have available?
  • Labelled examples (*supervised learning*)
  • Limited labelled examples (*weakly supervised learning*)
  • No examples (*unsupervised learning*)

*Top tagging is a supervised classification problem*
Approaches

• **Fully connected** networks

  • *Jet Constituents for Deep Neural Network Based Top Quark Tagging*
  J Pearkes, W Fedorko, A Lister, C Gay
  1704.02124

• **See Shannon’s talk**

• **Image** recognition

• **Natural language** processing

  • *Boosted Top Tagging with Long Short-Term Memory Networks*
    To be published

• **See Shannon’s talk**

• **Lorentz vectors**
Image approach

- Jets = 2d grayscale images:
  - 1 pixel = 0.1 in eta, 5 degree in phi
  - pixel energy: calorimeter ET
- Preprocessing (for illustration!)
  - Center maximum
  - Rotate so that second maximum is 12 o’clock
  - Flip so that third maximum is on the right side
  - Crop to 40x40 pixels

Deep-learning Top Taggers or The End of QCD?

Origins:
Jet-Images: Computer Vision Inspired Techniques for Jet Tagging
J Cogan, M Kagan, E Strauss, A Schwartzman
arXiv:1407.5675
Jet-Images – Deep Learning Edition
Ld Oliveira, M Kagan, L Mackey, B Nachman, A Schwartzman
JHEP 1607 069
Playing Tag with ANN

- Early development towards deep learning for jet physics
- Calorimeter images used in fully connected network

Playing Tag with ANN: Boosted Top Identification with Pattern Recognition
LG Almeida, M Backovic, M Cliche, SJ Lee, M Perelstein
JHEP 1507 (2015) 086

Figure 1. Graphical representation of the Artificial Neural Network (ANN).

Networks in the context of image recognition, see for example [42]. Mathematically, the ANN can be thought of as a succession of non-linear transformations:

\[ h^{(1)}_{i} = f(W^{(1)}_{ij} \boldsymbol{e}_{j} + b^{(1)}_{i}) \]

...\[ h^{(l)}_{i} = f(W^{(l)}_{ij}h^{(l-1)}_{j} + b^{(l)}_{i}) \]

\[ Y = f(W^{(O)}_{j}h^{(l)}_{j} + b^{(O)}) \],

where \( f \) is the so-called activation function, chosen to be

\[ f(z) = \frac{1}{1 + e^{-z}}. \]

The inputs \( \boldsymbol{e}_{i} \) are simply the normalized energy deposits defined above, rearranged in a single 900-dimensional vector: \( \boldsymbol{e}_{i} = \sum_{a=30}^{a+b} \).

The weights \( W^{(L)}_{ij} \) and the biases \( b^{(L)}_{i} \) are numbers determined by the training procedure, which we will now describe.

To train the network, we use a set of \( N/2 \) top and \( N/2 \) QCD jets, where \( N \) is a large number. For the \( i \)-th jet, we assign the "target output" variable:

\[ y_{i} = 1 \] if it is a top jet,
\[ y_{i} = 0 \] if it is a QCD jet.

Training consists of adjusting the weights so that the actual outputs of the ANN \( Y_{i} \) correspond as close as possible to the target outputs \( y_{i} \), across the training set. To quantify the error, we use the logarithmic loss variable

\[ \text{Log-loss} = \frac{1}{N} \sum_{i=1}^{N} (y_{i} \log(Y_{i}) + (1 - y_{i}) \log(1 - Y_{i})). \]

The goal of training is to choose weights that minimize this function. We use the back-propagation algorithm [43], combined with gradient-descent minimization. In its simplest version, the algorithm can be summarized as follows [44]:

1. Initialize the weights of each link to small random values.

3. In Eq. (3.1) and below, repeated indices are always summed over.

Figure 3. Distributions of the ANN output \( O \) on top (red) and QCD (blue) jet samples in three representative \( p_{T} \) ranges. All distributions are normalized to unit area.
Convolutional Network

- **How to build a convolutional network**
  - Chain multiple conv layers
  - Reduce image resolution in between
  - Use multiple masks per layer
  - Add linear ANN in the end

Convolutional (conv) layer

(This is still a network. We just use a fancy idea to decide which nodes to connect to each other)
Performance

- Train a BDT on a set of standard tagging variables

SoftDrop + n-subjettiness:
\[
\{ m_{sd}, m_{fat}, \tau_2, \tau_3, \tau_2^{sd}, \tau_3^{sd} \}
\]

MotherOfTaggers:
\[
\{ m_{sd}, m_{fat}, m_{rec}, f_{rec}, \Delta R_{opt}, \tau_2, \tau_3, \tau_2^{sd}, \tau_3^{sd} \}
\]
Lorentz Vector Based Learning

Input is a $p_T$ sorted list of Lorentz four-vectors: (calo towers or particle flow objects)

$\mathbf{k}_{\mu,i} = \begin{pmatrix} E_0 & E_1 & \cdots & E_N \\ p_{x,0} & p_{x,1} & \cdots & p_{x,N} \\ p_{y,0} & p_{y,1} & \cdots & p_{y,N} \\ p_{z,0} & p_{z,1} & \cdots & p_{z,N} \end{pmatrix}$

**Combination Layer (CoLa):** create linear combinations: $k_{\mu,i} \xrightarrow{\text{CoLa}} \tilde{k}_{\mu,j} = k_{\mu,i} C_{ij}$

**Lorentz Layer (LoLa):** Use resulting matrix to extract physics features.
Main assumption is the Minkowski metric

- Per pseudo-jet variables: $\tilde{k}_{\mu,i} \rightarrow \tilde{k}_{0,i}$
  $\tilde{k}_{\mu,i} \rightarrow \tilde{k}_{\mu,i} \tilde{k}_{\nu,i} \eta^{\mu\nu}$

- Trainable sums: $\tilde{k}_{\mu,i} \rightarrow \tilde{k}_{0,i} A_{i,j}$

- Sum of differences: $\tilde{k}_{\mu,i} \rightarrow \sum_j (\tilde{k}_i - \tilde{k}_j)_{\mu,j} (\tilde{k}_i - \tilde{k}_j)_{\nu} \eta^{\mu\nu} B_{i,j}$

**Diagram:**

- CoLa: $\mathbf{C} = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 1 \end{pmatrix}$
  - Sum of all constituents
  - Trainable linear combinations

- Diagonal matrix (pass-through constituents)

**Equation:**

$k_{\mu,i} \xrightarrow{\text{CoLa}} \tilde{k}_{\mu,j} = k_{\mu,i} C_{ij}$
For the soft and hard selections we have tested values of epochs. We then consider the two standard ranges of epochs. Training terminates either after 200 epochs or if the performance does not improve for five epochs, typically after several tens of epochs.

The network is trained using \( \text{Adam} \) with a learning rate of 0.001. Training terminates either after 200 epochs or if the performance does not improve for five epochs, typically after several tens of epochs. The learning rate is used to scale the gradients of the loss function with respect to the weights of the network. The Adam optimizer is a variant of stochastic gradient descent that maintains an exponentially weighted average of the gradient and the squared gradient to adapt the learning rate during training.

We independently train five copies of the DeepTopLoLa [20] tagger, because we eventually include track-based corrections to these towers with \( \text{Delphes3} \) [31] back-end, the new, fast, and flexible, as well as their ordered mean transverse momentum for the soft and hard fat jet selections of Eq.(8).

The fat jet constituents at the particle flow level are significantly improved by the new method to significantly improve over these two approaches is that the pixelled image removes information related to the Minkowski metric is that it combines different features: two subjets are Minkowski-close if

\[ g = \text{diag}(0.99 \pm 0.02, -1.01 \pm 0.01, -1.01 \pm 0.02, -0.99 \pm 0.02) \]
Other Ideas

- Only covered works targeting top tagging
- Many other interesting ideas out there

  - *QCD-Aware Recursive Neural Networks for Jet Physics* (1702.00748)
  
  - *Classification without labels: Learning from mixed samples in high energy physics* (1708.02949)
  
  - *Pileup Mitigation with Machine Learning (PUMML)* (1707.08600)
  
  - *CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks* (1705.02355)
  
  - ...
Conclusions

• Tasting menu of boosted top taggers

• QCD Inspired variables:
  • Mass: Trimming / mMDT / Softdrop
  • Prongs: n-subjettiness / ECFs

• Inclusive Taggers:
  • HEPTopTagger, HOTVR, Shower Deconstruction, BEST

• Deep learned:
  • Fully connected, Images, LSTMs, LoLa
The End.

Thank you
Bonus Slides
A Very Simple Network

\[ y = f(f(x_1)w_1 + f(x_2)w_2) \]

\[ f(x) = \Theta(x) \cdot x \]
How do networks learn?

- **Backpropagation + Gradient descent**
- Pass input \((x_1, x_2)\) to ANN
- Calculate output \((y)\) and difference to true value \((\hat{y})\)
  This is the loss function \(L\)
- Find gradient of loss function with respect to weights
- Use gradient to find new weights

\[
L(y, \hat{y}) = (y - \hat{y})^2
\]

\[
w'_i = w_i + \alpha \cdot \frac{\partial L}{\partial w_i}
\]