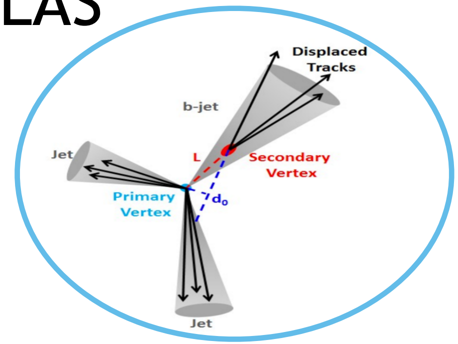
NN Optimisation for Flavour Tagging in ATLAS

Hammers and Nails - Machine Learning & HEP
20th July 2017

Marie Lanfermann, Andrea Coccaro, Tobias Golling

Introduction to flavour tagging in ATLAS

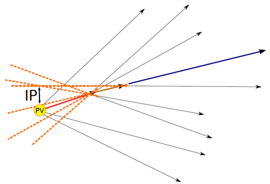


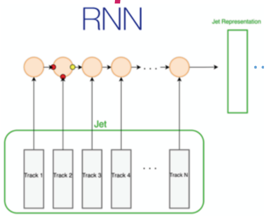
Higher Level Tagger
What do we want?

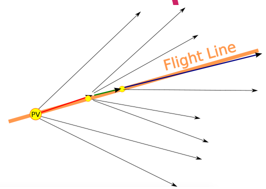
- **b-tagging**
- **c-tagging**
- Robust tagger (data/MC comparison)
- **Optimisation** and **Generalisation**
  - Good performance over **full kinematics region**
  - Good for **various physics searches**
- As little total work as possible

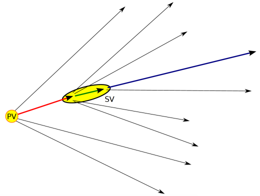
Details next Wednesday by Tobias

Kinematics









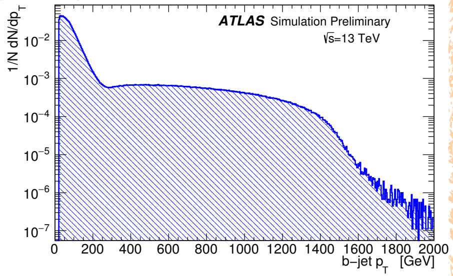
Protocol

Hybrid ttbar/Z' sample

Pre-processing | Training incl. loss monitoring | Evaluation

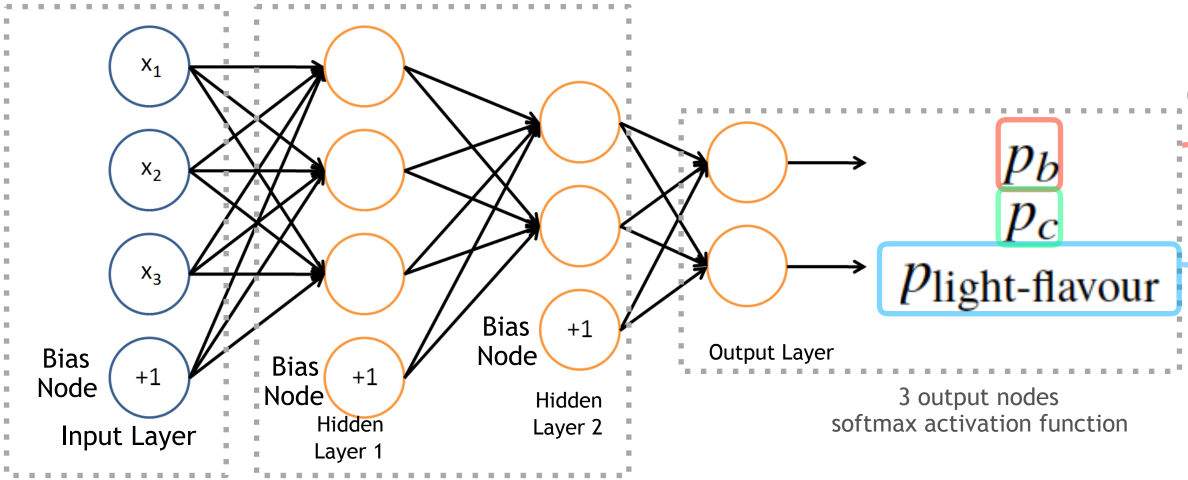
Pure samples of ttbar or Z'

- Pre-processing:
  - Reweight in 2D kinematics to b-jet distribution <—**treating flavours on equal footing**
  - Default values:
    - No values far from non-default values but rather **set to mean of non-default values**
    - Introduce **binary default-check variables** (to propagate information on the values being defaults)
- Training (Hybrid ttbar/Z' sample):
  - Interesting phase space **up to O(1TeV)**
    - **Available statistics: 5.1 M training jets, 1.3 M validation jets**
  - Weighs are used in the back propagation update (training & validation set)
- Evaluation (separate pure samples of ttbar or Z'):
  - **Available statistics: ttbar: 6.5 M jets; Z': 4.3 M**



DL1 - General Overview

Training using
ttbar-Z' **Hybrid sample**



higher level variables

6-10 hidden layers
Mixtures of
MaxoutDense and Dense layers
ReLU activation function

3 output nodes
softmax activation function

Performance evaluation
on **pure ttbar and Z' samples**

Building the
final DL1 discriminant:
Reducing output dimensionality

b-tagging: $$DL1b\,f_{\text{c-jets}} = \ln\left(\frac{p_b}{f_{\text{c-jets}}\cdot p_c + (1-f_{\text{c-jets}})\cdot p_{\text{light-flavour}}}\right)$$

c-tagging: $$DL1c\,f_{\text{b-jets}} = \ln\left(\frac{p_c}{f_{\text{b-jets}}\cdot p_b + (1-f_{\text{b-jets}})\cdot p_{\text{light-flavour}}}\right)$$

—> **Increased Flexibility:**
+ Background weighing tuneable after training
+ Same training usable for b- and c-tagging

NN config file size ~1MB

Grid Search

- Keras sequential model
  - 3 output nodes
- Theano backend
- **Adam** optimiser
  - Minimise categorical cross-entropy loss
- General settings:
  - **ReLU** activation function (softmax for output layer)
  - Mixture of **Maxout and Dense layers**
  - **BatchNormalisation**
  - **Dropout** (training) for robustness
    - 1st layer: 10% of nodes masked
    - Other hidden layers: 20% masked
  - 100 training epochs

- Varied:
  - Number of hidden layers, layer type sequencing, number of nodes, learning rate

Grid Search

Parameter	Varied value
Number of hidden layers (nodes per layer)	[*5* (48(MO)-36-24-12(MO)-6), *6* (57(MO)-48-36-24-12(MO)-6), *7* (72(MO)-57-48-36-24(MO)-12-6), *8* (72(MO)-57-48-36-24(MO)-18-12-6), 78(MO)-66(MO)-57-48-36-24(MO)-12-6, 72(MO)-57-60-48-36-24(MO)-12-6, 78(MO)-66(MO)-57-48-36-24(MO)-12-6), *9* (72(MO)-57-60-48-36-24(MO)-18-12-6), *10* (78(MO)-66-57-60-48-36-24-18-12-6, 78(MO)-66-57-60-48-36-24(MO)-18-12-6, 78(MO)-66(MO)-57-60-48-36-24(MO)-18-12-6)]
Learning rate	[0.001, 0.0005, 0.0001]

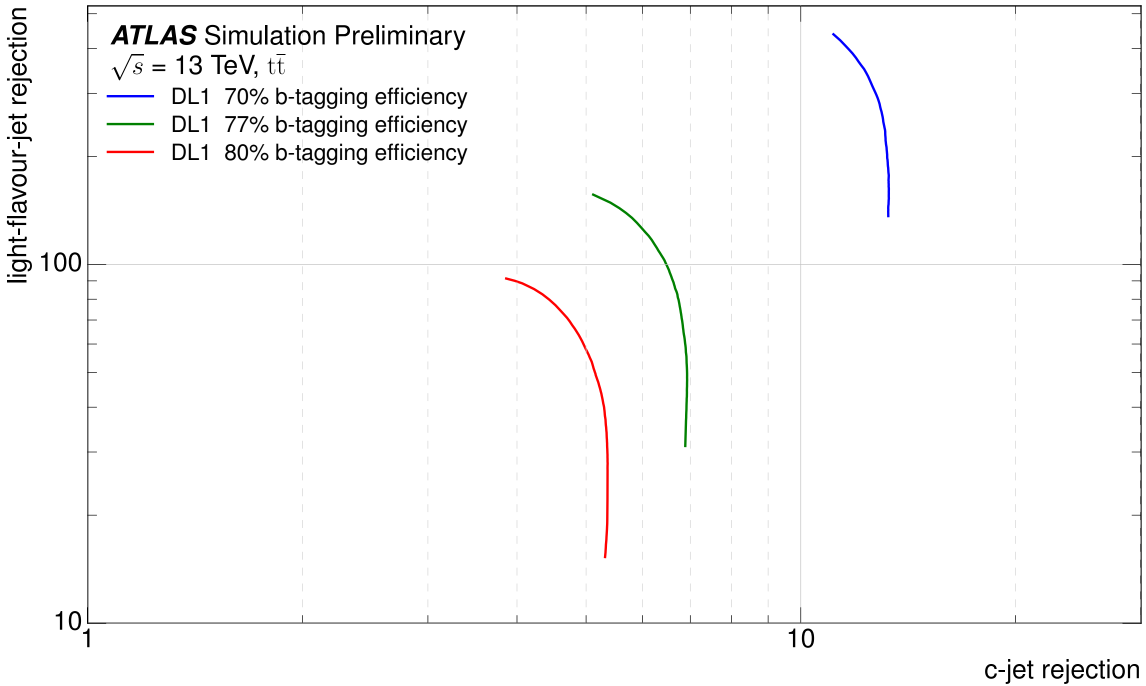
MO: Maxout layer

—> Approximately 100k trainable parameters

Information accessible after construction Kears model via: model.summary()

Optimisation Procedure

1. **Sanity Checks:**
    1. #(training samples) > #(free parameters of the model)
    2. Loss development on training and validation set is monitored
2. Performance evaluated on test sets
3. **Extend training** for best performing configuration
    - Performance after different number of epochs evaluated on test sets (using Keras ModelCheckpoints)



b-tagging

Tuneable after training:

Background fraction of the final DL1 discriminant can be adapted for physics performance interests by moving along the lines

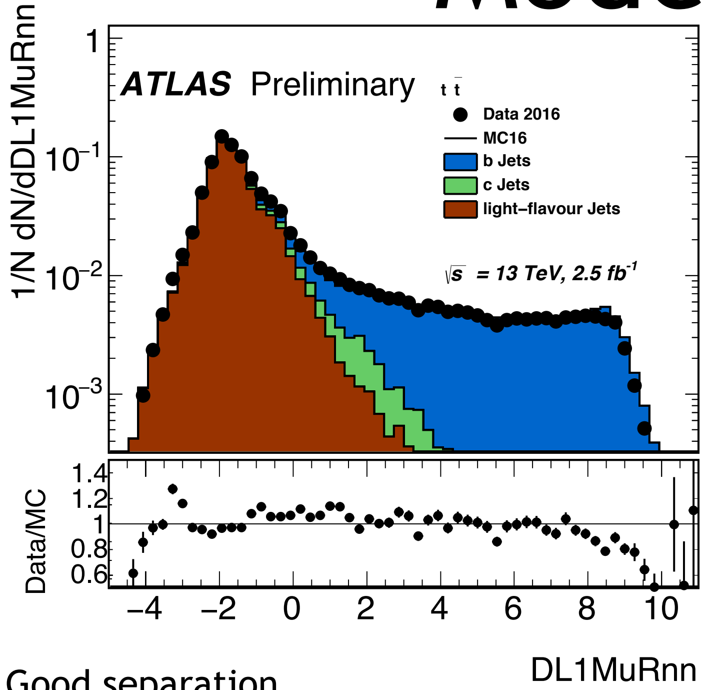
Iso-efficiency curve = Scan over full range of $f_{\text{c-jets}}$

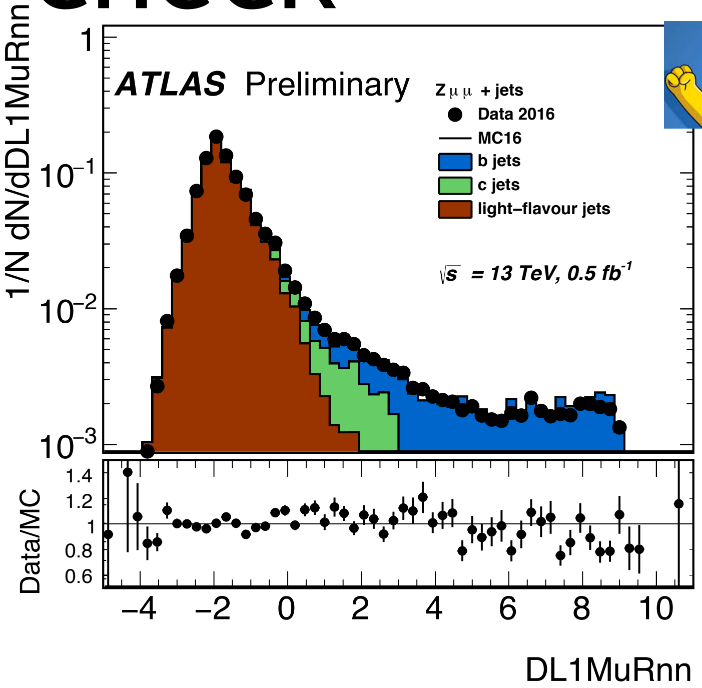
$$DL1bf_{\text{c-jets}} = \ln\left(\frac{p_b}{f_{\text{c-jets}} \cdot p_c + (1 - f_{\text{c-jets}}) \cdot p_{\text{light-flavour}}}\right)$$

Discussion on performance improvements next Wednesday by Tobias

Modelling check

Generalisation





- Good separation
- Simulation describes the data within 20% with some localised differences for low and high values
- To be checked with more data

Conclusions

- Novel highly flexible tagger **ready to be used on 2017 data**
  - Only one training
  - Tuneable after training
- Calibration analysis starting
- The theorem “There’s no free lunch” holds

The more inputs are taken into account, the more the NN approach gains in performance w.r.t. a BDT

References:

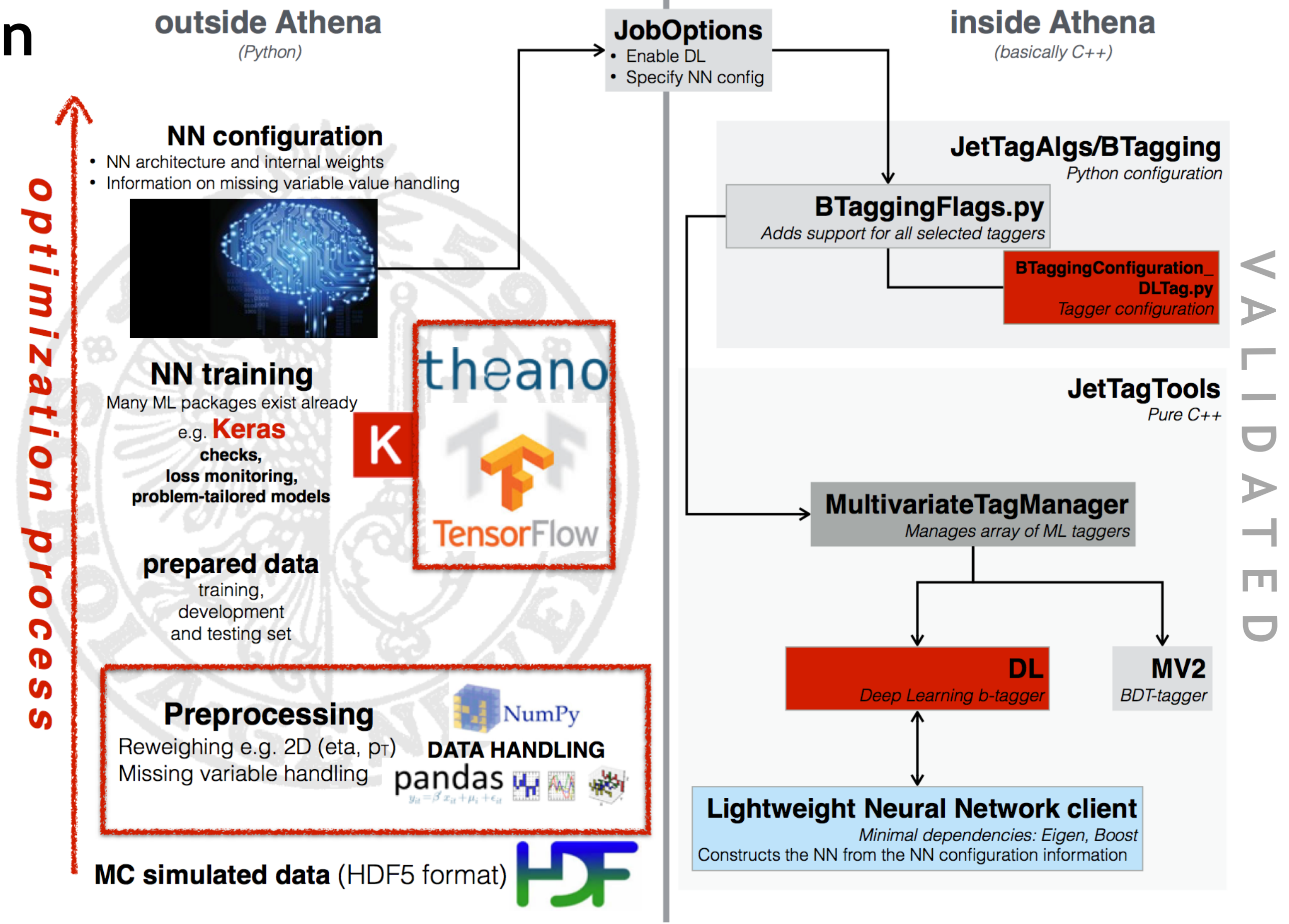
Deep Learning in the ATLAS experiment, ATL-PHYS-SLIDE-2017-477, http://cds.cern.ch/record/2274065.

Optimisation and performance studies of the ATLAS b-tagging algorithms for the 2017-18 LHC run, ATL-PHYS-PUB-2017-013, http://cds.cern.ch/record/2273281.

Identification of Jets Containing b-hadrons with Recurrent Neural Networks at the ATLAS experiment, ATL-PHYS-PUB-2017-003, http://cds.cern.ch/record/2255226.

BACKUP

The DL1 chain

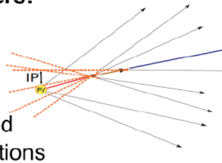


Overview

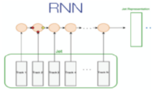
Selection of lower level taggers:

Impact Parameter (IP) based

IP2D, IP3D:
Log-likelihood ratios using flavour hypotheses computed from summed track contributions extracted from simulation-derived templates

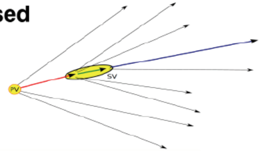


RNNIP:
Parallel approach which feeds raw tracks into a Recurrent Neural Network (RNN) and exploits correlations between the tracks

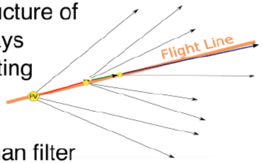


Secondary Vertex (SV) based

SV1:
Reconstructs inclusive secondary vertices

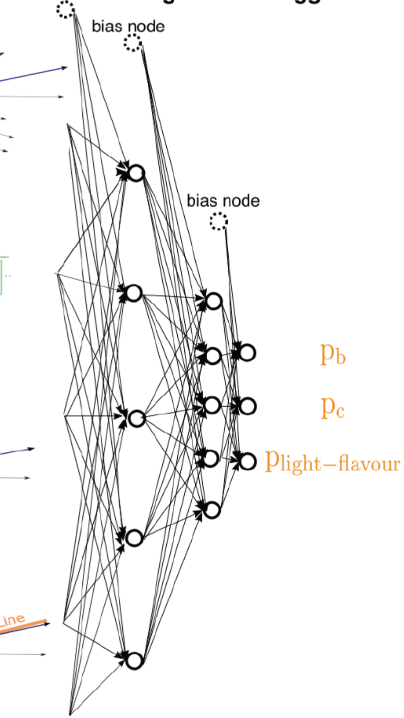


JetFitter:
Exploits the topological structure of weak b- and c-hadron decays inside the jet by approximating the b-hadron or c-hadron flight path with PV, SV and tertiary vertex using a Kalman filter



Also used:
- Jet kinematics
- Information on muons produced in b/c decays

DL1 higher level tagger:



Initial Calo-Jet Cuts

$$p_T^{\text{jet, calib}} > 20\ \text{GeV} \quad (1)$$

$$|\ \eta^{\text{jet, calib}}\ | < 2.5 \quad (2)$$

$$\text{if}\ |\ \eta^{\text{jet, calib}}\ | < 2.4 \text{ and } p_T^{\text{jet, calib}} < 60\ GeV : \text{JVT}^{\text{jet}} > 0.59 \quad (3)$$

Standard jet cuts for calo-jets, see equations 1, 2 and 3, are applied using the calibrated kinematic variables. For training, the uncalibrated kinematic variables are used as inputs.

Akt4EMTopo jets

Input modelling check