iTHEPHY
Innovative Team - Teaching for Physics

Classification in Particle Physics Using Machine Learning

Irene Cagnoli

Aaron van der Graaf

Gianluca Bianco

Florian Mausolf

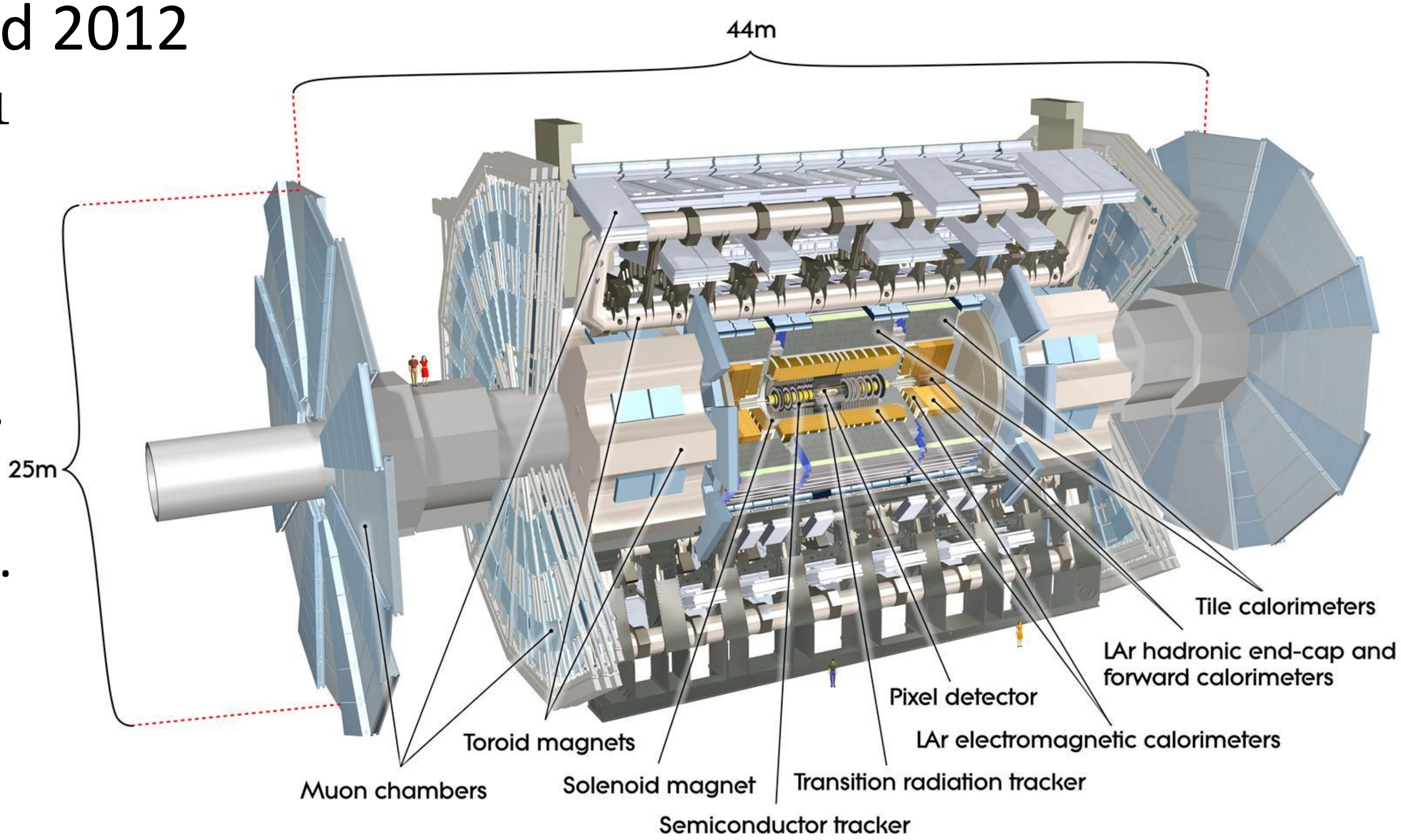
ATLAS

Higgs challenge

iTHEPHY Innovative Team-Teaching for Physics

ATLAS Analysis: Evidence for $H \rightarrow \tau\tau$ decays (2015)

- Data taken in 2011 and 2012
- $4.5\ \mathrm{fb}^{-1}$ and $20.3\ \mathrm{fb}^{-1}$
- 7 TeV and 8 TeV
- ATLAS detector
  - Inner tracking system.
  - Electromagnetic and hadronic calorimeters.
  - Muon spectrometer.

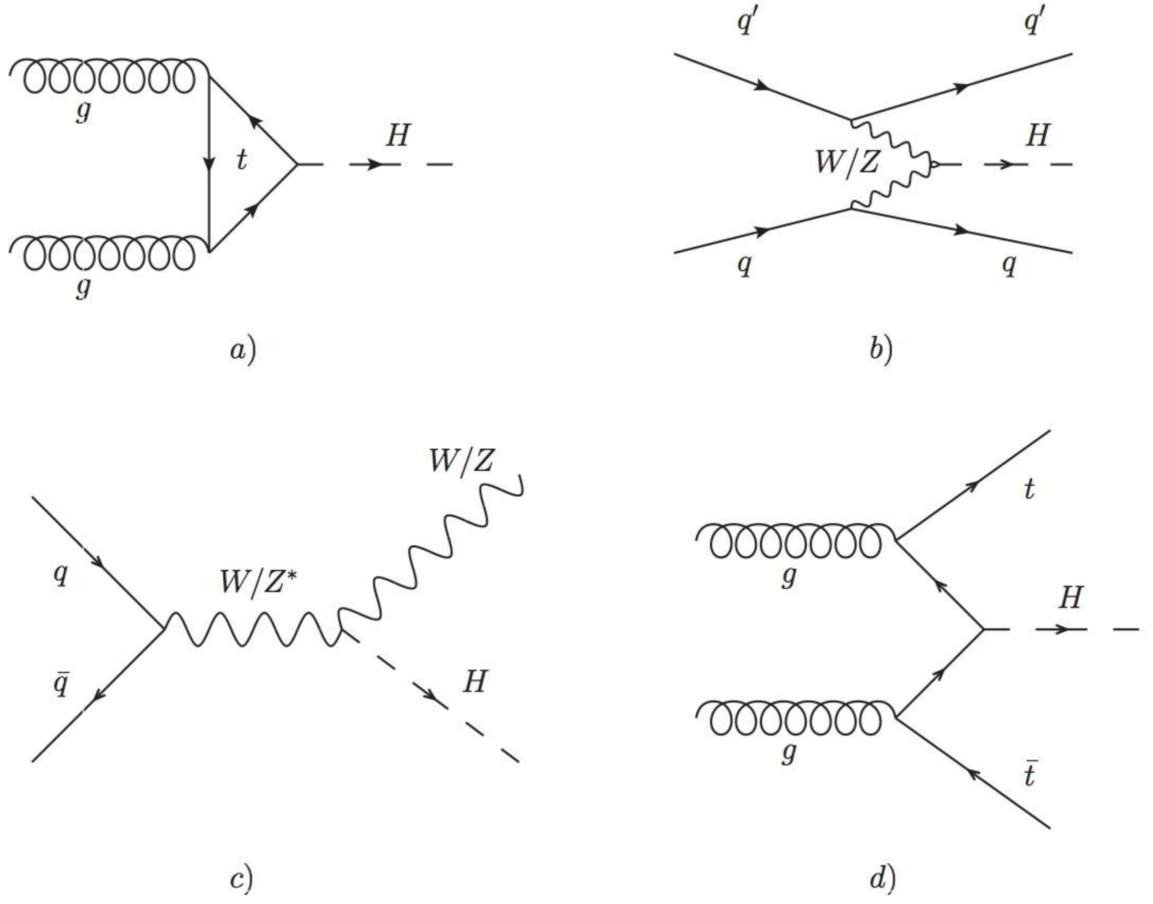


Motivation

- Higgs boson found in 2012
- Properties have to be investigated
- Yukawa coupling to fermions was not yet proved
  - $H \rightarrow \bar{b}b$ with 2.1 σ significance by CMS
  - $H \rightarrow \tau\tau$ with 3.2 σ significance by CMS

Higgs boson production at the LHC

a) Gluon fusion
b) Vector boson fusion (VBF)
c) Associated production with a vector boson
d) Production with a top-quark pair



Event selection

- Leptonic and hadronic decays are considered
  - $\tau \rightarrow l\bar{\nu}\nu$ or $\tau \rightarrow \nu\ hadrons$
- Two categories: VBF and Boosted
  - VBF: two separated high p_T jets
  - Boosted: High p_T Higgs boson candidate
- Three decay modes: $\tau_{lep}\tau_{lep}, \tau_{lep}\tau_{had}, \tau_{had}\tau_{had}$

Simulation and background events

- Signal and background events are simulated
  - NNLO QCD, NLO electroweak corrections for signal events
- Most important background in all channels:
  - $Z \to \tau\tau$, fake τ, $Z \to ll$
  - Other backgrounds: top quarks, $W + jets$, diboson background, …

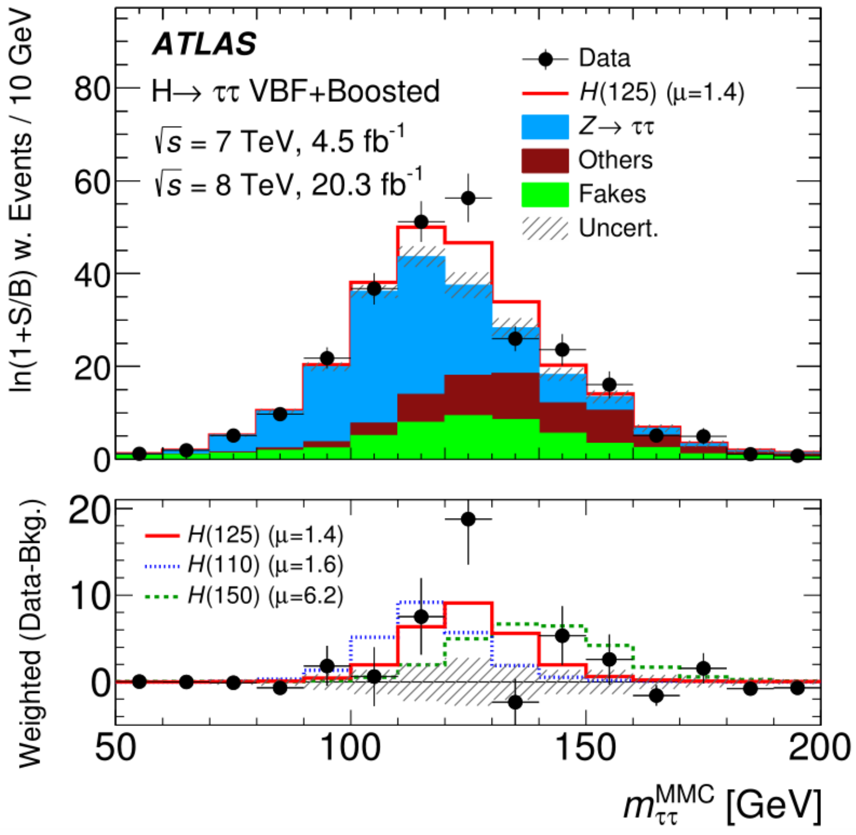
Analysis strategy

- Multivariate analysis using Boosted Decision Trees (BDT)
- Cross checked using cut based analysis
- Separate BDT trained for each category and channel
- 6-10 input variables used, depending on channel

iTHEPHY
Innovative Team - Teaching for Physics

Results

- Signal strength
  $$\mu = 1.43^{+0.27}_{-0.26}(stat)^{+0.32}_{-0.25}(syst) \pm 0.09(theo)$$
- Background-only hypothesis:
  $$p_0 = 2.7 \times 10^{-6}$$
- Deviation from background expectation at 4.5σ (expected 3.4σ)



Kaggle Higgs boson machine learning challenge

- Challenge based on ATLAS simulation
- Simplifications:
  - $\tau_{lep}\tau_{had}$ only
  - Only $Z \rightarrow \tau\tau, \bar{t}t$ and $W + jets$ background included
  - b-tagged jets are rejected
  - Other small simplifications are applied
- 13 derived and 17 primitive variables are included
- Callenge is evaluated using approximate mean significance (AMS)

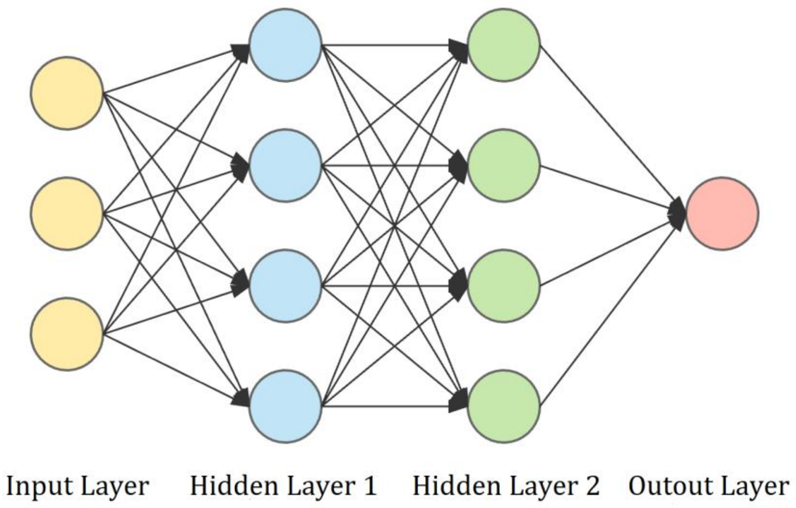
$$AMS = \sqrt{2\left(\left(s + b + b_{reg}\right)\ln\left(1 + \frac{s}{b+b_{reg}}\right) - s\right)} \approx \frac{s}{\sqrt{b}} \qquad (b_{reg} = 10)$$

Dataset

- Events are labelled as signal and background
- Events are splitted into three subsets with normalised weights
  - Training set: $250\ 000$ events
  - Validation set: $100\ 000$ events
  - Test set: $450\ 000$ events
- Missing values are set to a dummy

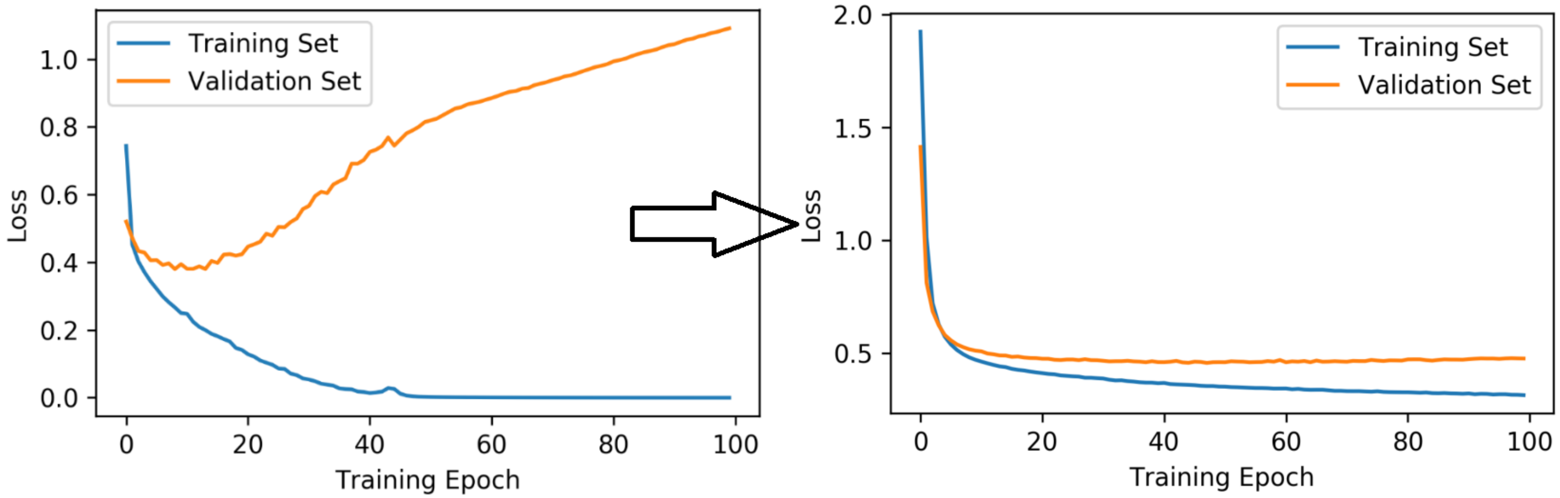
Neural Networks

- Neural Networks (NN) classify n-dimensional input vectors into a discriminant
- n nodes in the input layer
- Hiden layers
- All nodes are connected with weights
- Each layer: $\vec{y} = \text{activation}\left(W\vec{x} + \vec{b}\right)$
- Classification is evaluated using loss function

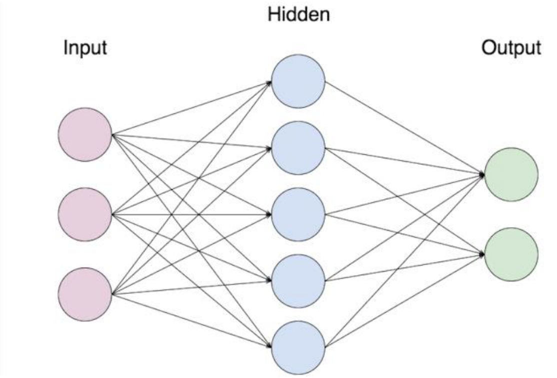


Training and regularisation

- Training: Gradient descent on the loss by adapting weights and bias
  - $w^{i+1} = w^i - \mathrm{h} \times \nabla \mathrm{L}(\mathrm{w^i})$, $\quad b^{i+1} = b^i - \mathrm{h} \times \nabla \mathrm{L}(\mathrm{b^i})$
  - Learning rate h: free parameter
- NN learn better from scaled data
- Overtraining: Model can adapt too closely to the training set
- Regularisation methods: Dropout layers, L1 and L2 regularisation



Machine Learning
Results Group 2.1



Irene Cagnoli
Aaron van der Graaf

Data Prepreation

- Adding feature: Category (if event is VBF/Boosted)
- Cleaning missing data
- Converted label from string to binary
- Standard Scaler before any splitting
- Splitting into TrainingSet, ValidationSet and TestSet
- Splitting data into 3 subsets depeding on jet number (0jet, 1jet, 2+jet)
- Cleaning useless features in subsets

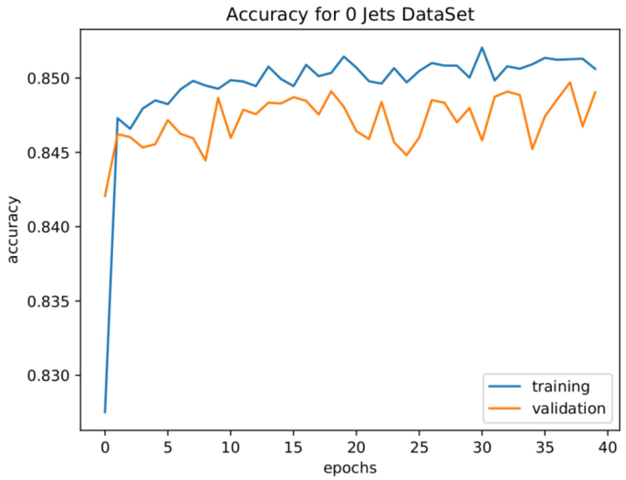
DNN Design

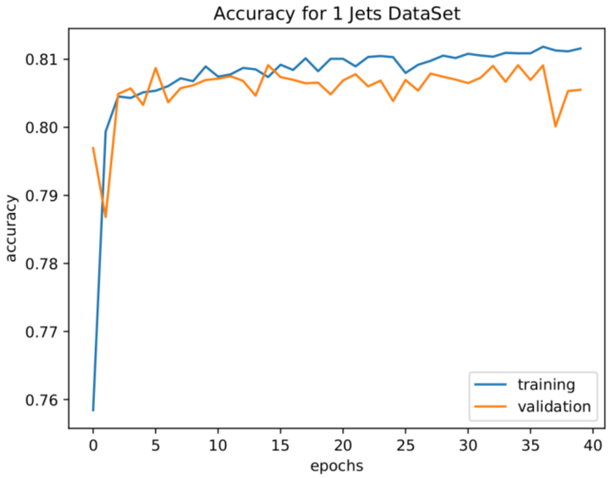
Trained a DNN for each subset 0jet ,1jet, 2+jets, all DDNs use the same structure

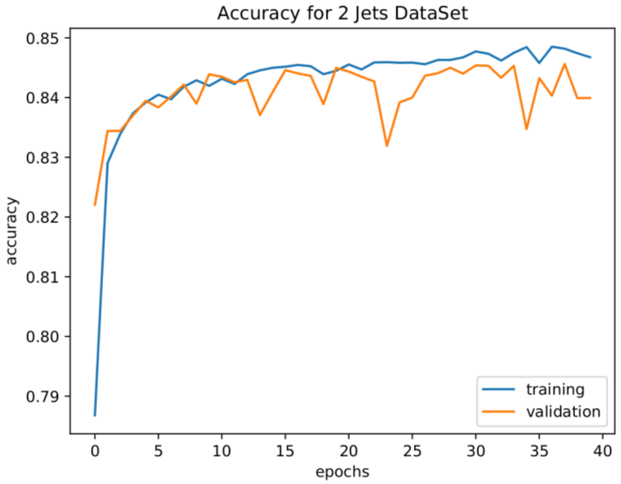
- fully-connected network
- Input dimension: 24 (0jets), 27 (1jet), 31 (2+jets)
- 7 hidden layers dimension: 64,128, 128, 64, 32, 16, 8 neurons
- 37441 (jet0), 37633 (jet1), 37889 (2+jets) trainable parameters
- Activation functions: ReLU except last layer sigmoid
- L2 regularisation $\lambda = 0.001$
- Loss: binary crossentropy
- Optimizer: adam
- Metrics: accuracy

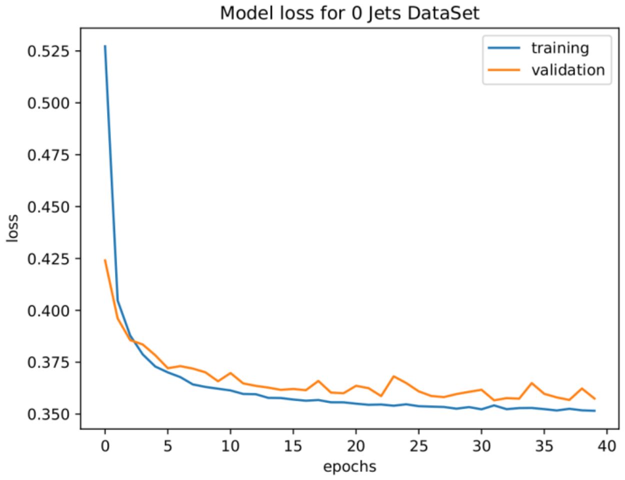
iTHEPHY
Innovative Team-Teaching for Physics

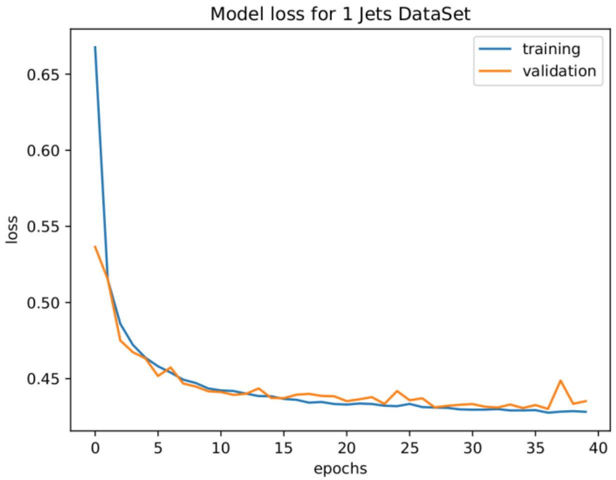
DNN Perfomance

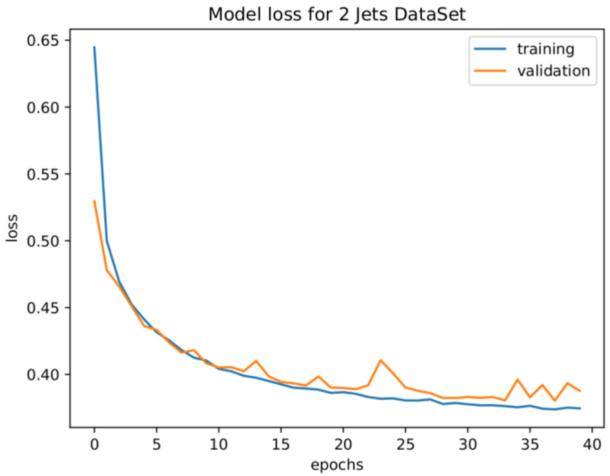




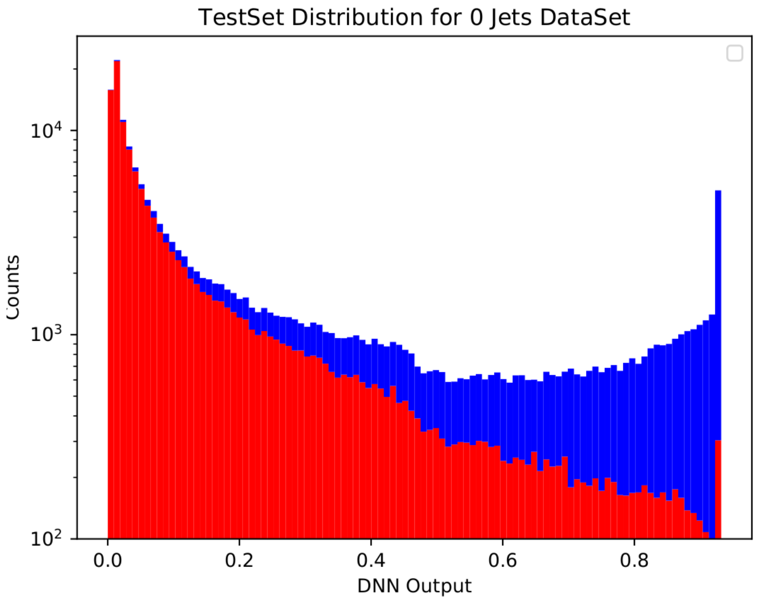




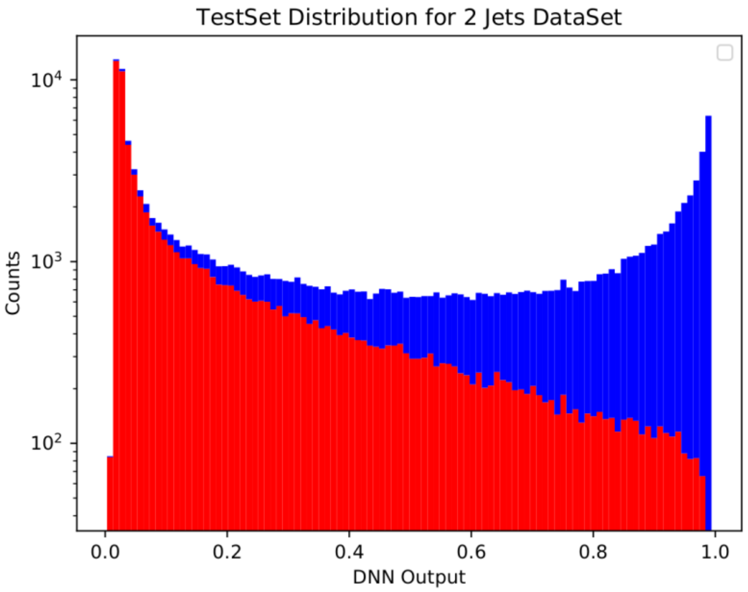




DNN Output



TestSet Distribution for 0 Jets DataSet
Counts
DNN Output

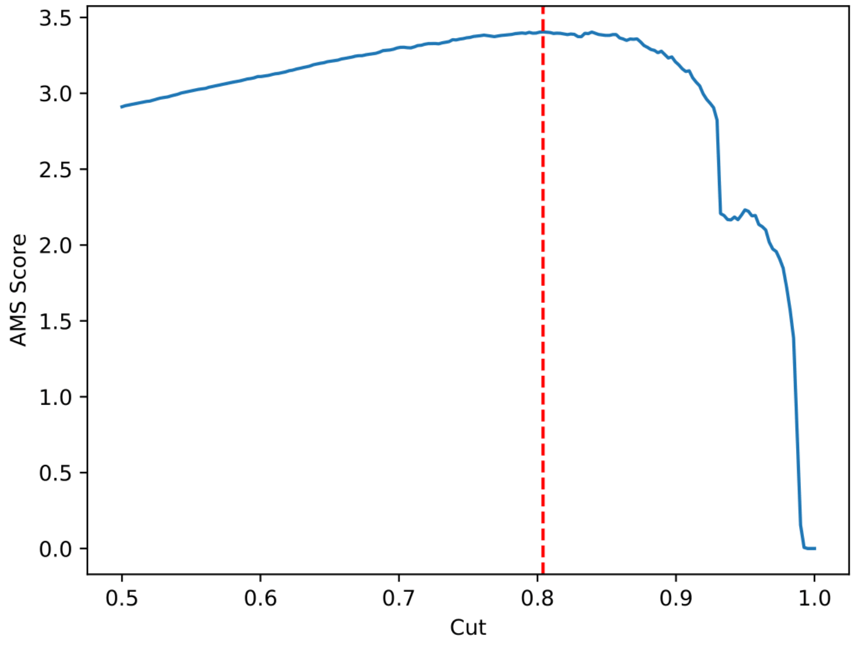


TestSet Distribution for 2 Jets DataSet
Counts
DNN Output

TestSet Distribution for 1 Jets DataSet
Counts
DNN Output

AMS Score

- Used checkpoints: going back to epoch with minimum in validation loss
- AMS Score: 3.4099



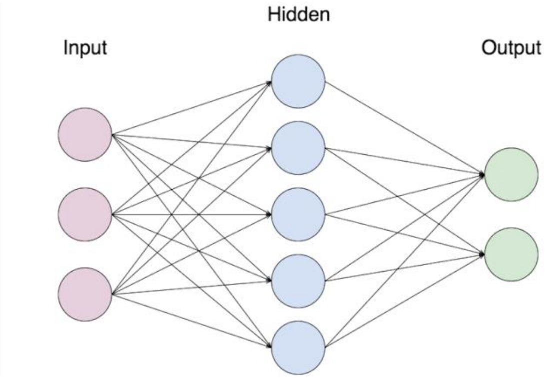
Improvement Ideas:

- Hyperparameter Optimisation
- Combine with ML-algorithm

iTHEPHY
Innovative Team - Teaching for Physics

Machine Learning Results Group 2.2

Higgs challenge



Keras

Gianluca Bianco
Florian Mausolf

Our strategy

- **Step 0**: Scale data -> Standard Scaler

$$z = \frac{x-\mu}{s}$$

  - Where x: feature, μ: mean, s: standard deviation

- **Step 1**: use 1 deep NN for the whole dataset
  - Low AMS
  - Too many useless jet variables
  - DNNs implemented in *Keras* from *Tensorflow*

Our Deep Neural Network

- **Step 2**: Eliminate useless variables by splitting dataset according to jet number:
  - 0 jets: ~100 000 events
  - 1 jet: ~80 000 events
  - $\geq$ 2 jets: ~70 000 events
- Perform a DNN classification for each subset
- Find best hyperparameters by grid search

Subset	Hidden Layers	Nodes	Dropout rate	Regularisation
0 Jets	6	32, 64, 128, 64, 32, 8	0.1	L1, $\lambda = 0.0001$
1 Jet	5	64, 64, 64, 32, 8	0.1	L1, $\lambda = 0.0001$
2 Jets	6	64, 128, 64, 64, 32, 8	0.1	L2, $\lambda = 0.0001$

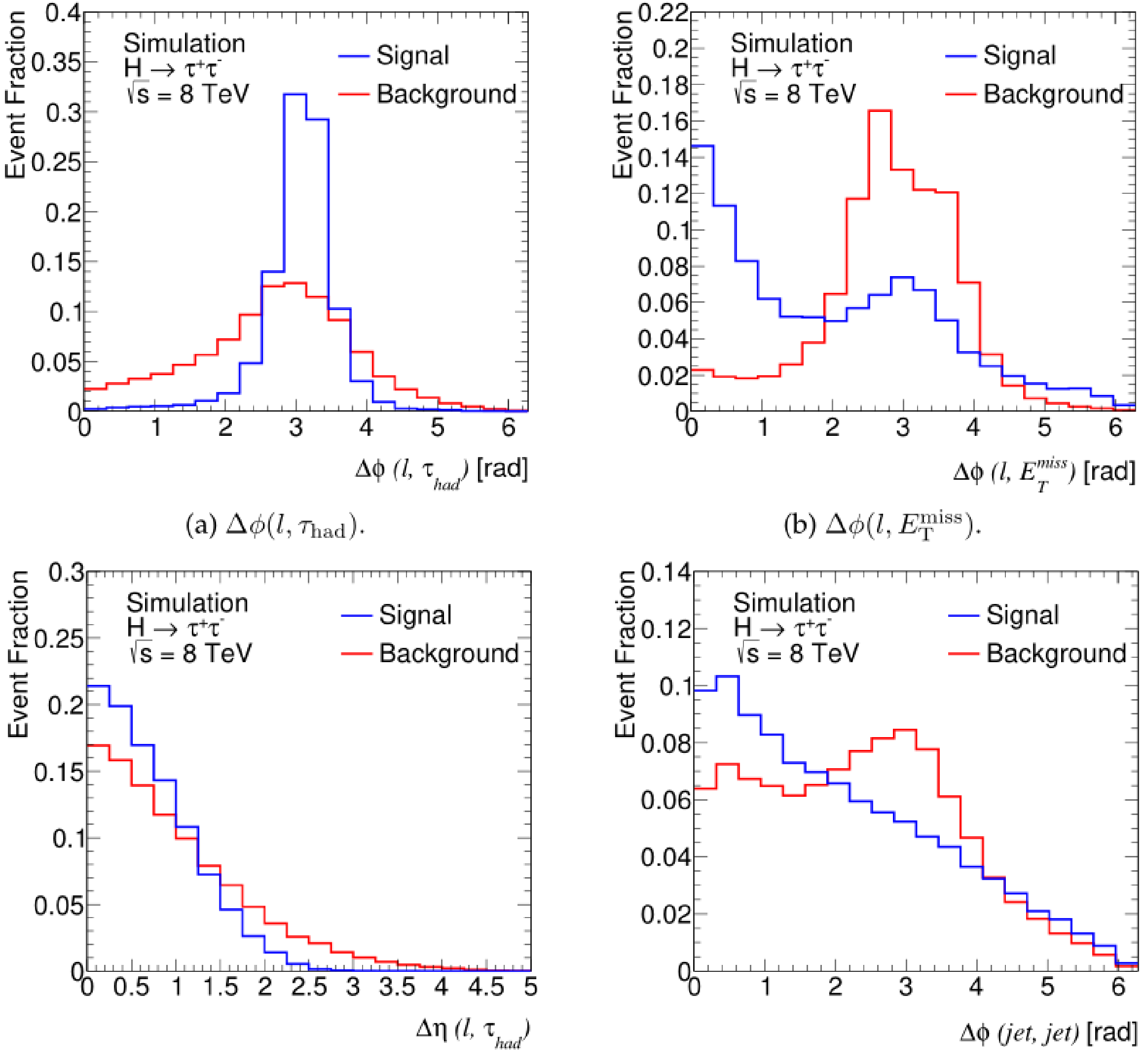
Other DNN parameters

- Optimizer: Adam
- Loss function: binary crossentropy
- Activation function for hidden layers: ReLu
- Sigmoid output
- Early stopping on validation accuracy:

$$Acc = \frac{tp + tp}{tp + tn + fp + fn}$$

Further optimisation

- **Step 3**: Improved classification by changing angular variables
- All new features have a separation power between signal and background

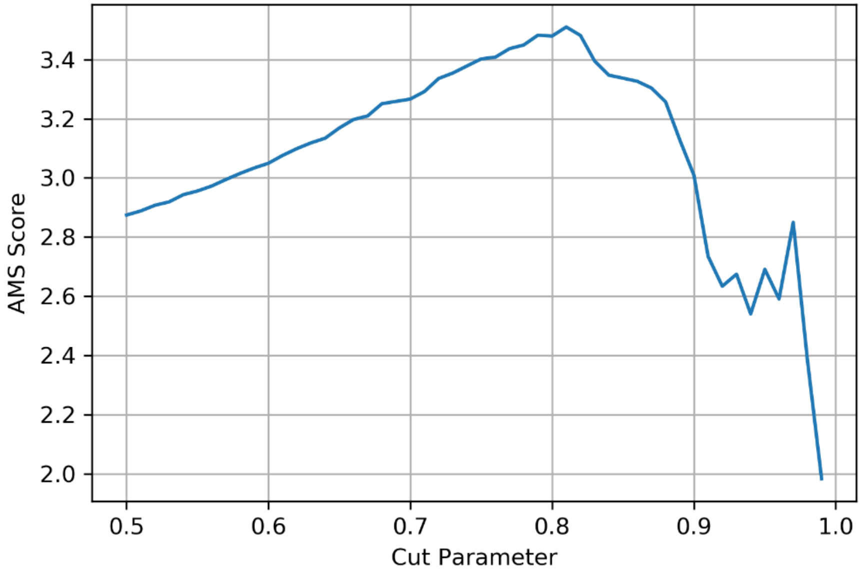


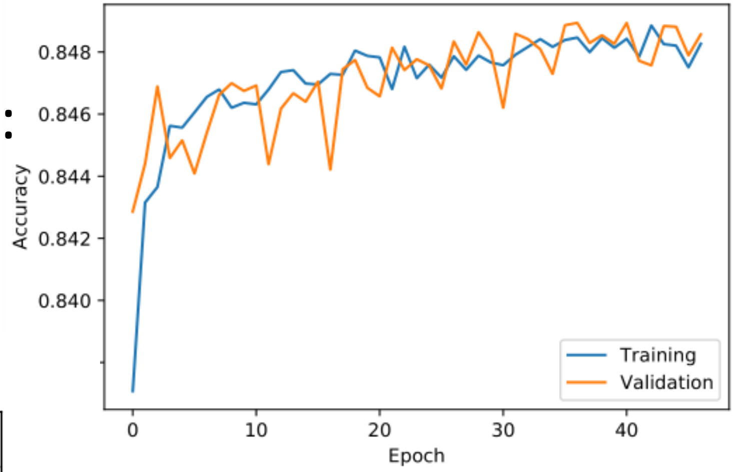
(a) $\Delta\phi(l, \tau_{\text{had}})$. (b) $\Delta\phi(l, E_{\text{T}}^{\text{miss}})$.

(c) $\Delta\eta(l, \tau_{\text{had}})$. (d) $\Delta\phi(jet, jet)$.

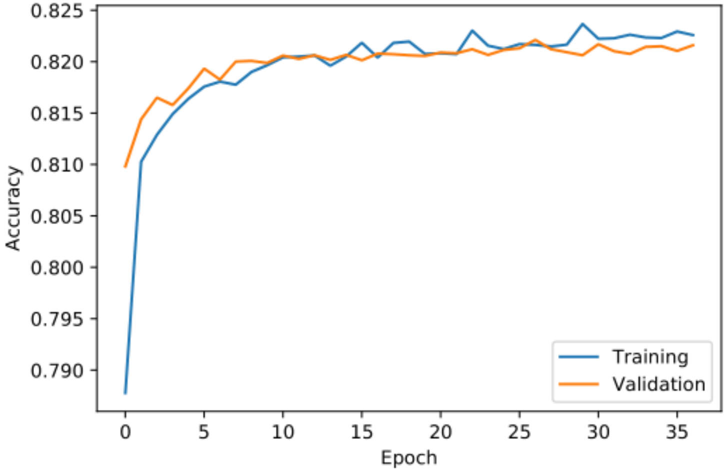
Learning curves and evaluation

- AMS score for the combined DNNs:
  3.55 at a cut parameter of 0.83

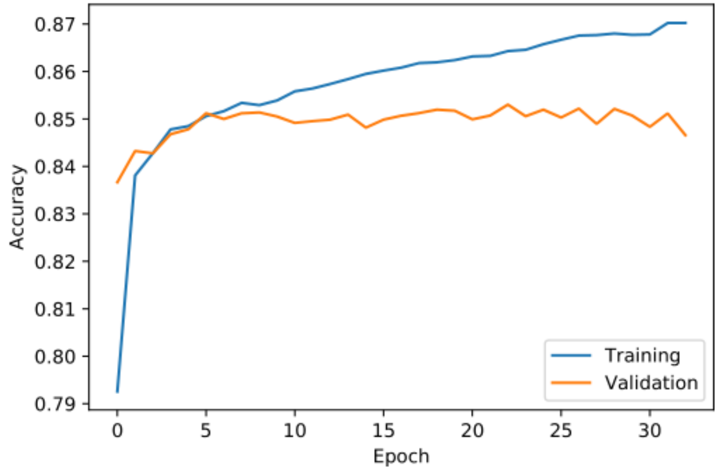




(a) 0 Jets.



(b) 1 Jet.



(c) 2 Jets.

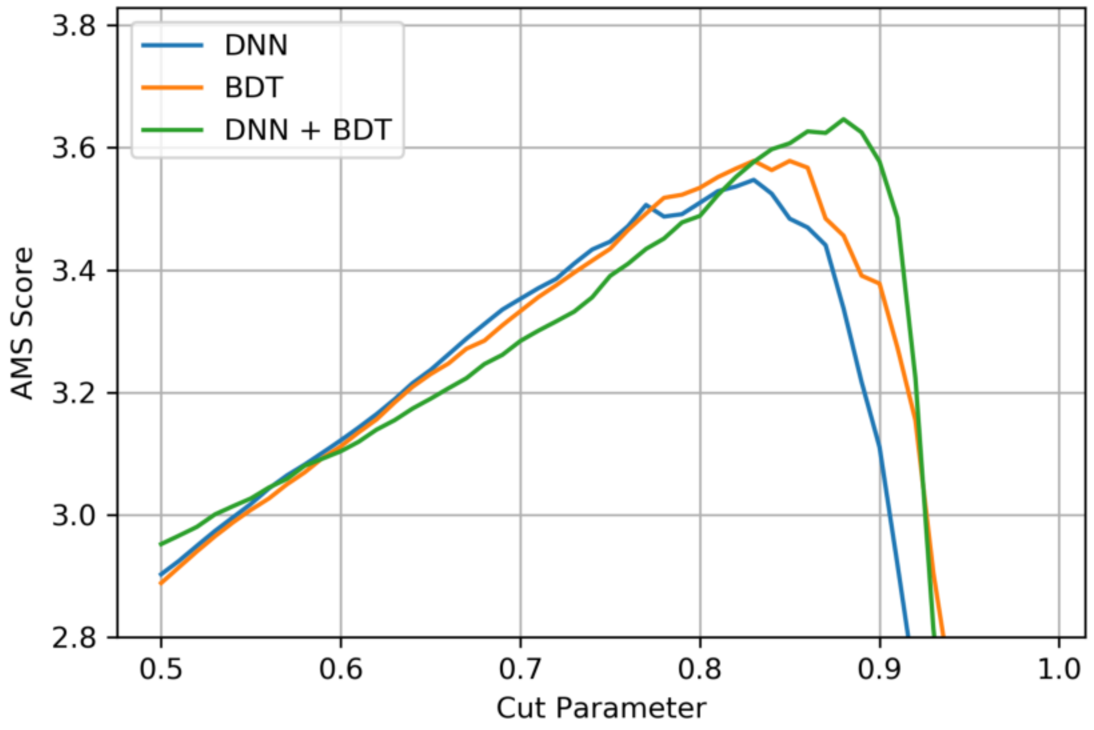
Comparison to a BDT model

- **Extra step**: try also a different model, based on BDT
- BDT combines many weak learners (decision trees) to a strong classifier
- Implemented in *Scikit-Learn: HistGradientBoostingClassifier*
  - 90 trees with up to 50 nodes
  - Up to 50 leaves per tree
  - L2 regularization: $\lambda = 0.5$
  - Learning rate: 0.1
  - Loss: binary crossentropy
  - Up to 50 bins per feature for the histogram
- AMS score for the BDT model:
  - 3.58 at a cut parameter of 0.83

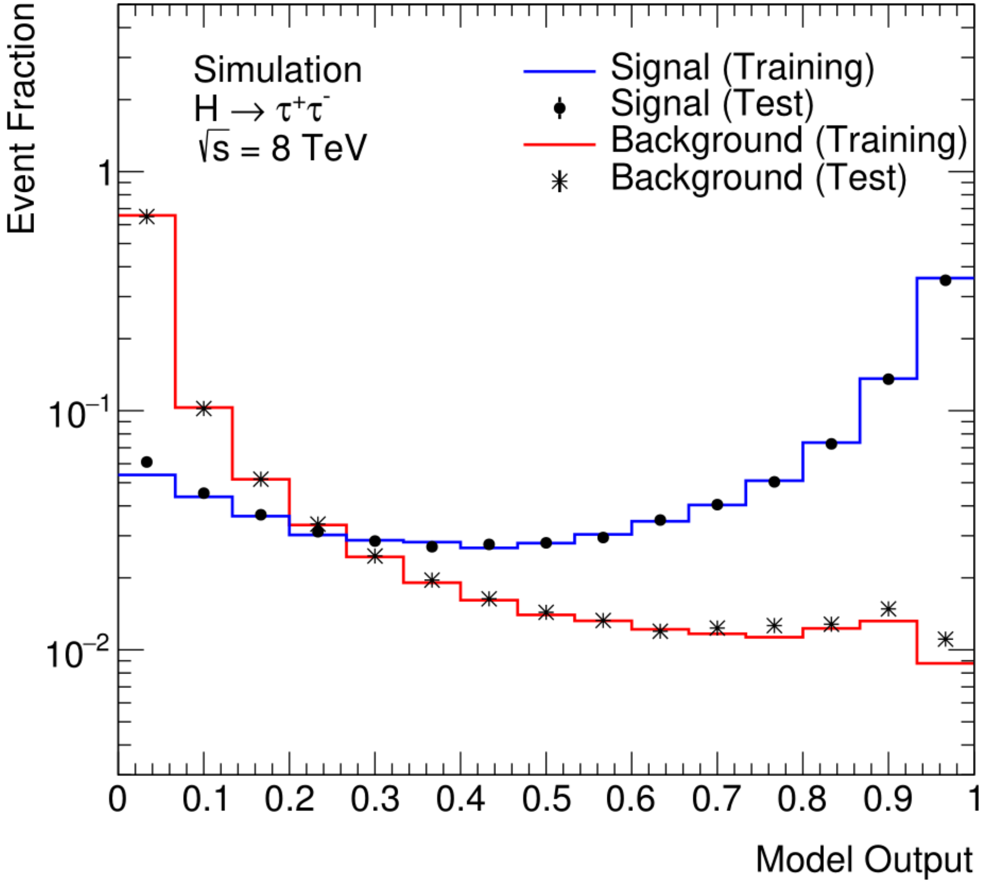
Final results

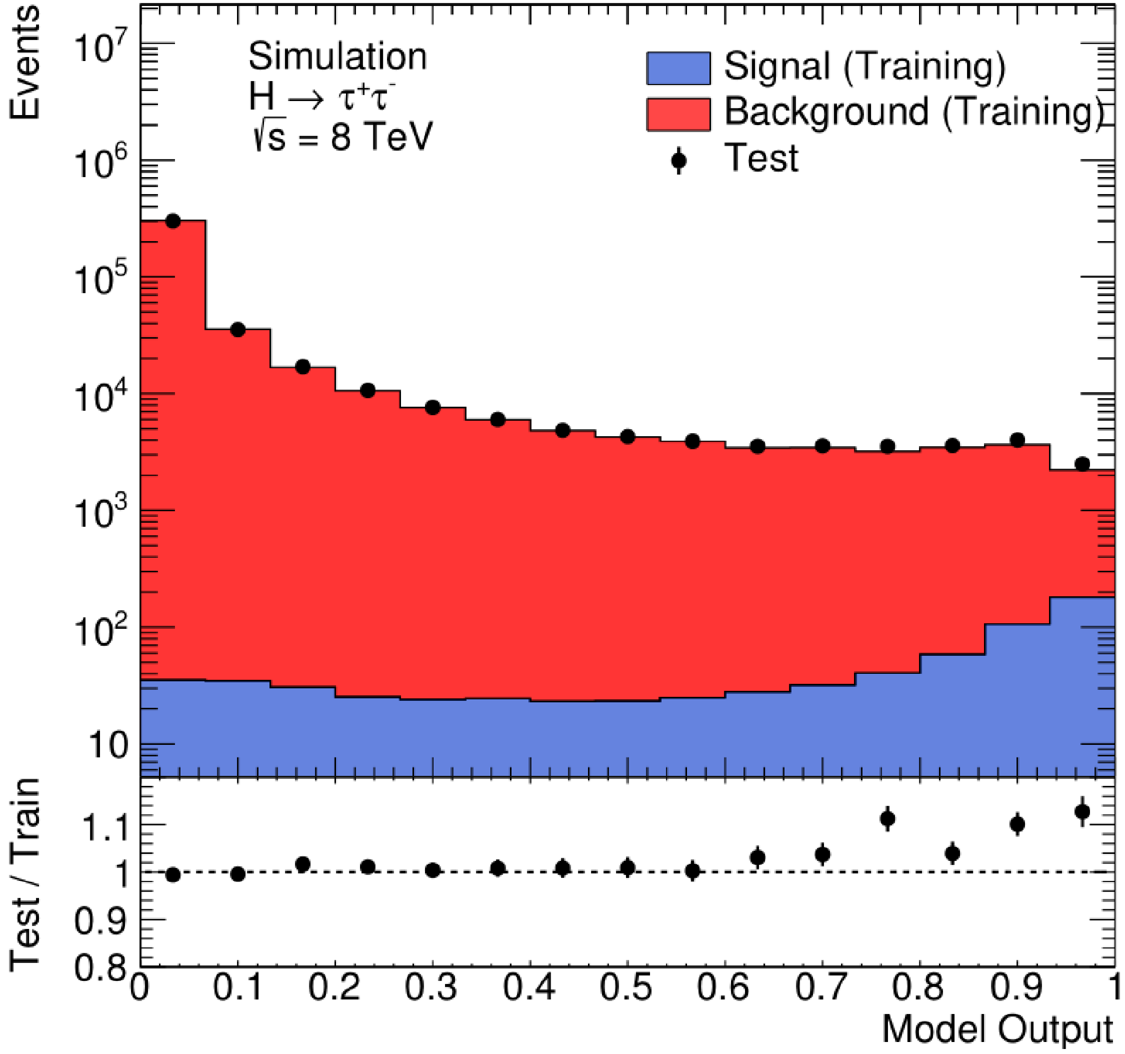
- **Final Step:** Combination of both models (BDT and DNNs) reaches the highest AMS
- Combined using logistic regression

- Final AMS: 3.65 at 0.88
- Kaggle rank 445 of 1784 (unofficially)



Model outputs: Training vs. test set





Backup Slides

BDT Model

BDT consider an additive model of M trees in the following form:

$$F(x) = \sum_{m=1}^{M} \gamma_m h_m(x)$$

where h_m denotes the $m-$th decision tree and γ_m is the step length. The model is created iteratively in the following way:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

where the h_m tries to minimise the loss function L via

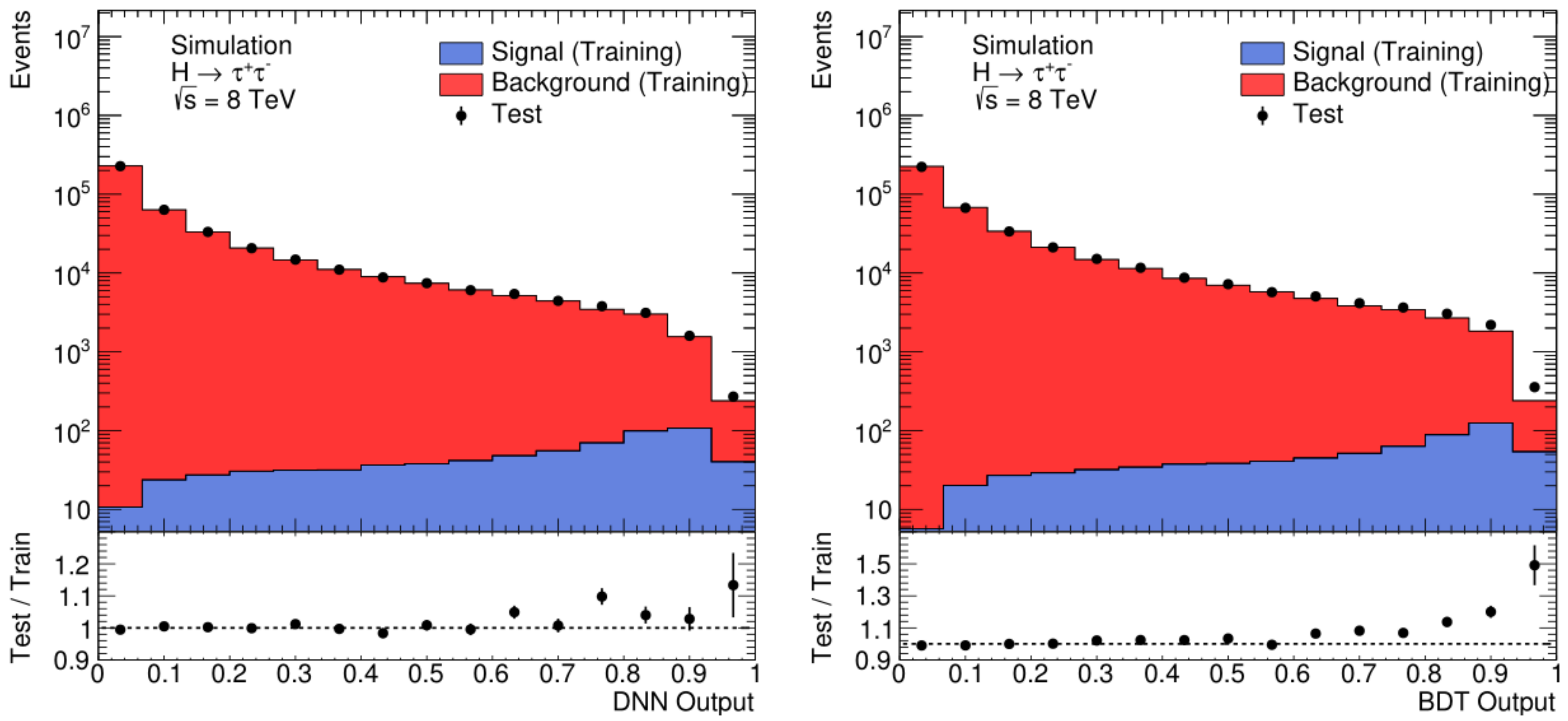
$$h_m = \arg\min_h \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + h(x_i)),$$

where n is the number of training samples and y_i is the $i-$th label

Gradient boosting attempts to solve the minimisation numerically. This method is similar to the one used for DNNs. The step length γ_m is chosen as:

$$\gamma_m = \arg\min_\gamma \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) - \gamma \frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}\right)$$

Overtraining test: DNN vs. BDT



(a) DNN output.

(b) BDT output.