MVA results in the $2e/\mu + 1\tau_{had}$ channel

Christian Kirfel, Pablo Martinez Agullo

31st August 2021

UNIVERSITÄT BONN

Introduction

I Summary of MVA methods developed for the dileptau channel

II Explanation of strategies and ongoing efforts

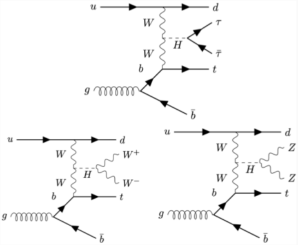
III Comparison of neural network and BDT approaches

IV Discussion of future strategy

Used configuration

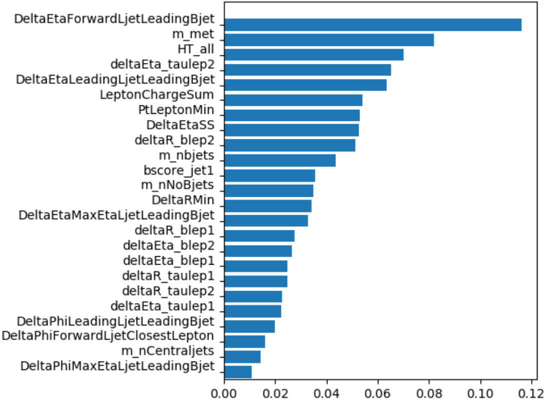
At tHqLoop level some cuts are applied accounting for the geometric acceptance of the triggers thresholds.

- Jets
  - $p_T(jet) > 20.0$ GeV
  - $|\eta(jet)| < 4.50$
- b-jet
  - $p_T(b-jet) > 20.0$ GeV
  - $|\eta(b-jet)| < 2.50$
  - DL1r_PC with eff_70_70
- Lepton
  - $p_T(e) > 10.0$ GeV
  - $|\eta(e)| < 2.47$ and $|\eta(e)| \notin [1.37, 1.52]$
  - $p_T(\mu) > 10.0$ GeV
  - $|\eta(\mu)| \in [0.01, 2.50]$
  - $p_T(\tau) > 20.0$ GeV
  - $|\eta(\tau)| < 2.50$ and $|\eta(\tau)| \notin [1.37, 1.52]$
- E_T^{miss}: no cut (up to 800 GeV)
- 1-6 jets and 0-2 b-jets



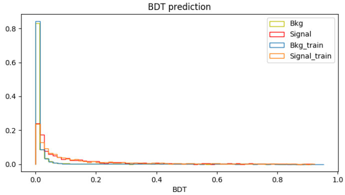
- Tight leptons
- $p_T(Lepton1) > 28.0$ GeV
- $p_T(Lepton2) > 20.0$ GeV
- $p_T(Lepton3) > 20.0$ GeV

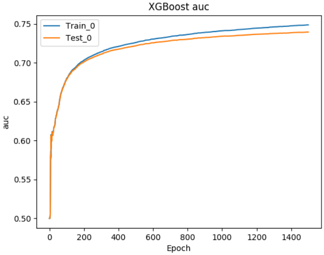
BDT Features



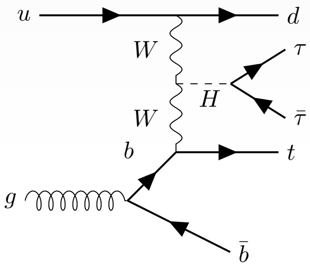
BDT results

- Using XGBoost library
- Optimised using the genetic algorithm
- Reduced learning rate to avoid overfitting
- Using only positive weights only an AUC of 0.75 is reached
- All events: AUC drops to 0.53





General ML selection



- n-jets: 2 (b-jets: 1)
- b-jet WP: 70 DL1r
- nLeptons & nTaus: 2e/μ 1τ_{had}
- $E_{\text{T,miss}}$: no cut (to 800 GeV)

- jets:
  - p_T > 35 GeV
  - $|\eta| < 4.5$
  - EMPFlow
- electrons:
  - p_T > 20 GeV leading 27 GeV
  - $|\eta| < 2.5$ not in 1.37 - 1.52
  - WP: LooseAndBLayerLH ; isolation: no requirement
- muons:
  - p_T > 20 GeV leading 27 GeV
  - $0.01 < |\eta| < 2.5$
  - WP: Loose ; isolation: no requirement
- taus:
  - p_T > 20 GeV leading 27 GeV
  - $|\eta| < 2.5$ not in 1.37 - 1.52
  - WP: RNNLoose
  - ASG recommended OLR (τ_{had} remove jets)

Features and weight setup

- Absolute weights for training because
  → *bestandmoststableresultsPreliminaryselectionofvariables*

eta_jf	forward jet eta
pt_jf	forward jet transverse momentum
mass_jf	forward jet mass
phi_jf	forward jet phi
eta_b	b-jet eta
pt_b	b-jet transverse momentum
phi_b	b-jet phi
HvisMass	mass of LorentzV sum of hadronic taus
m_met	Missing energy
Reco_w_mass_2	Reconstructed mass of the W case 1
Reco_w_mass_1	Reconstructed mass of the W case 2

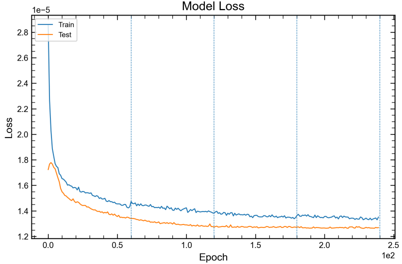
deltaRTau	Delta R of the hadronic taus
deltaPhiTau	Delta phi of the hadronic taus
HvisPt	pt of LorentzV sum of hadronic taus
HvisEta	eta of LorentzV sum of hadronic taus
TvisMass	mass of reconstructed top
TvisPt	pt of visible top
TvisEta	eta of visible top
M_b_jf	Mass of LorentV sum of b and jf
HT	Sum of transverse energies
lep_Top_pt	Light lepton pt
lep_Top_eta	Light lepton eta

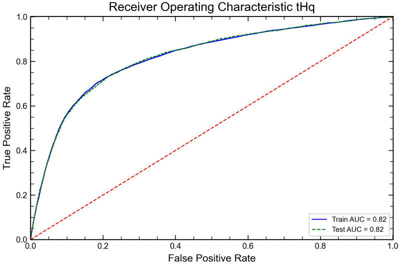
Hyperparameters

- Optimised by small grid search
- More thorough optimisation using evolutionary method scheduled, method is in place

Hyperparameter	Setting
Model	Categorical
Nodes	120
Layers	6
Dropout	0.65
Batchnormalisation	On
Activation	elu
Output activation	sigmoid
Batch size	1000
Optimisation	Adam
Weight Initialisation	Lecun Normalisation
K-folds	4

NN results for tHq vs all backgrounds





- Good separation, no overtraining
- AUC of 0.82 consistent with test sample
- Outperfoming overall BDT performance

Motivation for a categorical classification

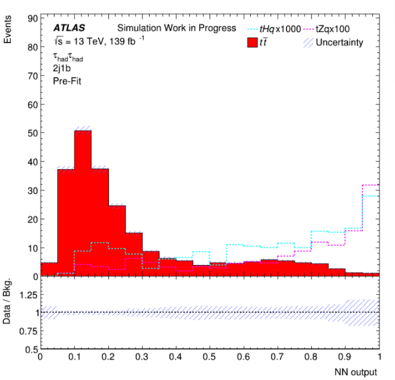
Underlying problem

- Treating tZq as background lead to heavy missclassification
- Giving tZq a different training label should decrease the problem

Technicalities

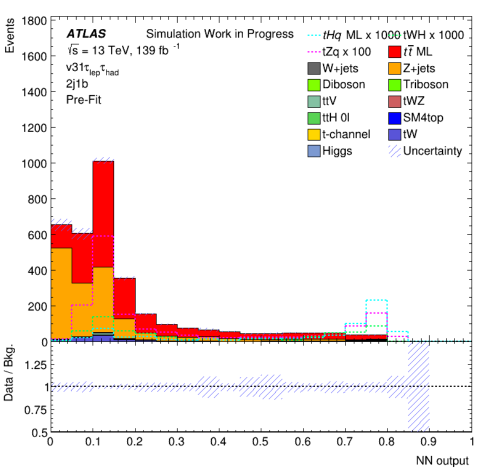
- Use OneHotEncoding to give labels to signal, background and tZq
- Plot response and ROC for every label separately
- The final response is a vector. One component per label likelihood
- All components add up to 1

Previous results



- tHq against other processes
- ttbar dominates the training
- tZq misclassified as signal
- Possible approaches:
  - multiple networks
  - multiple targets
  - reweighting samples

Response plot



- Majority of tZq is no longer missclassified
- Part of tZq is still missclassified but tHq also gets a cleaner peak

Summary

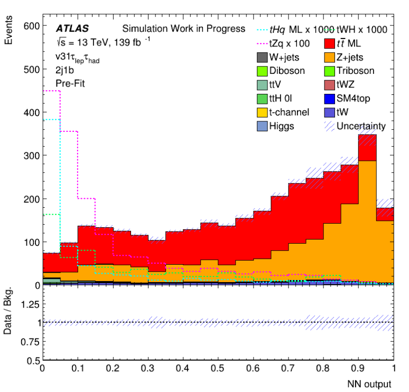
Feature	Neural Network	BDT
AUC, positive weights only	-	0.75
AUC	0.82	0.52
Feature optimisation	not yet	done
Evolutionary optimisation	in place	done
Reduction of tZq missclassification	done	-
Write response to tree	done	not yet

Backup

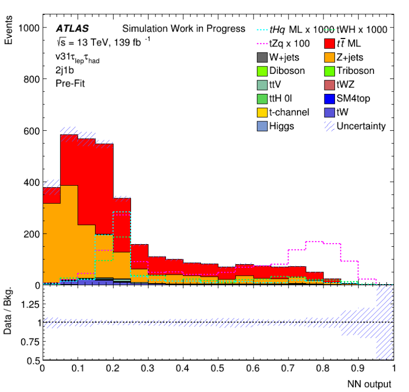
Evolutionary neural networks

- Starting with a set of random configurations
- Evaluate the results of the first generation and generate a new generation based on AUC
- Repeat until a good configuration is reached
- Advantages:
    - Decrease user bias for hyperparameter choice
    - Optimised to run on worker nodes
    - Quick discarding of bad configurations
    - User friendly for unexperienced students

Responses

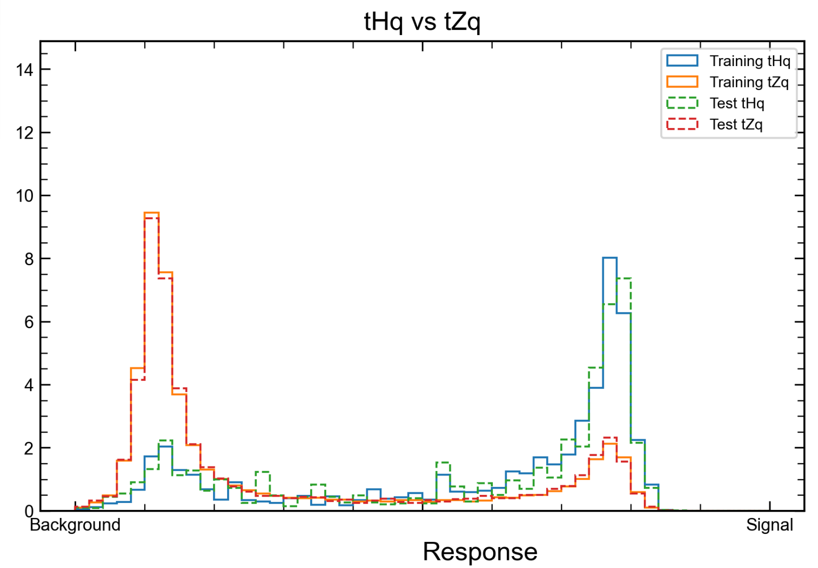


Background response



tZq response

tHq versus tZq



BDT summary

- A cut a bit below 0.2 would remove around 99% of bkg events and 80% of signal. Having just the 1% of the bkg and 20% of the signal would greatly increase our significance.
- With the cut on the BDT we would have (approx.): bkg/sg = 4877/20 = 243 Improved by a factor 71 to the before-presel scenario. Including BDT score in the trees
- Using all events, not just the ones with positive weights reduces AUC surprisingly significantly
- In the future the BDT should be tested in specific regions or for specific backgrounds

Genetic Algorithm

1. **Initialise Population**: Defines a matrix with pop_size rows (we use 30) and n(hyperparam. to optimize) columns. Each element is a random number within a range. Each row is a set of hyperparameters.
2. **Fitness function**: For each row (set of hyperparameters) the GA evaluates the fitness function (Z_n). In our study $z_n = \frac{1}{1-AUC} - log\_loss$.
3. **Selection and Drop**: Ranks each row according the fitness function and removes the worst half of the initial population.
4. **Diversify population**: Duplicate the the remaining rows (so that we have pop_size rows again) and modify the new ones to achieve diversity in the population. There are two techniques to do this
   - **Cross pair**: Randomly combine the new copies. There is a probability a specific value could be exchanged between two rows.
   - **Mutation**: A value could be modified to avoid local minimum. There is a probability the algorithm multiplies a specific value by a random value from a normal distribution.
5. **Drop duplicate and Renew population**: Drop duplicate if any and add new rows until the number of rows matches that of the initial population.
6. **Iterate**: Do all these steps again
7. **Results**: We keep the best set of hyperparams following these metrics: Fitness function value (Zn), AUC, y LogLoss