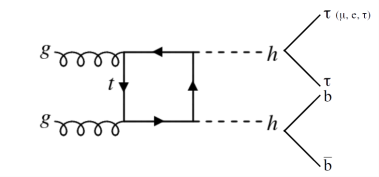
Di-Higgs searches with Machine Learning

Miguel Bengala and Rodrigo Santo
Supervisors: Michele Gallinaro and Giles Strong

6th September 2018

Introduction

- Goal
  - $\rightarrow$ explore the potential of advanced machine learning methods to project the expected discovery significance of non-resonant di-Higgs production in HL-LHC using the upgraded CMS detector
- Task
  - $\rightarrow$ classify events into "$\mu\,\tau_h\,b\,b/e\,\tau_h\,b\,b/\tau\,\bar{\tau}\,b\,\bar{b}$ decay of di-Higgs" versus "background", optimising the approximate median significance (AMS)
- Data
  - $\rightarrow$ samples produced via Monte Carlo generator of di-Higgs and several background channels ($t\,\bar{t}$ inclusive, SM Higgs, DY to di-Lepton, di-Boson WW and ZZ, W+jets, vector boson VH, single top)



Work Guidelines

- The data previously described was fed to deep neural networks (DNN) in order to build a classifier
- Several recent methods in DNN were applied to evaluate their efficiency
- The study was first performed for the Higgs ML challenge
  - $\rightarrow$ simulated LHC collision data with features characterising events detected by ATLAS of Higgs $\tau\,\bar{\tau}$ decay
- Used not only as a benchmark of the performance of each model and its optimisations but also to get us familiarised with DNN concepts

Deep Neural Network

- Basic classifier:
  - → Deep Neural Network with 3 hidden layers, each with 100 neurons
  - → Output layer of a single neuron
  - → Ensemble of 10 networks is trained on 50% of the data, using cross-validation, for 65 epochs
    - Models pre-trained without sample weights
    - Models weighted according to loss on validation data
  - → Remaining data is used to test the classifier and optimise the threshold

Feature selection

- Train only on the low-level final-state features plus multiplicity features
  - → give the best performance, since the high-level features can be implicitly computed by the network
  - → final set of 52 selected features

  - p_x, p_y, p_z, $|p|$, mass, energy and transverse mass of the hadronic tau τ_h, the muon and di-Higgs: 21 features;
  - p_x, p_y, p_z, $|p|$, mass and energy of both b-jets, $h_{b\,\bar{b}}$ and $h_{\tau\,\bar{\tau}}$: 24 features;
  - p_x, p_y, $|p|$ of missing transverse momentum: 3 features.
  - s_T the scalar sum of $\vec{p}_T^{\,miss}$, muon p_T and the transverse energy of both b-jets and the τ_h: 1 feature;
  - total number of jets, number of b-jets and number of tau-jets: 3 features.

Feature selection

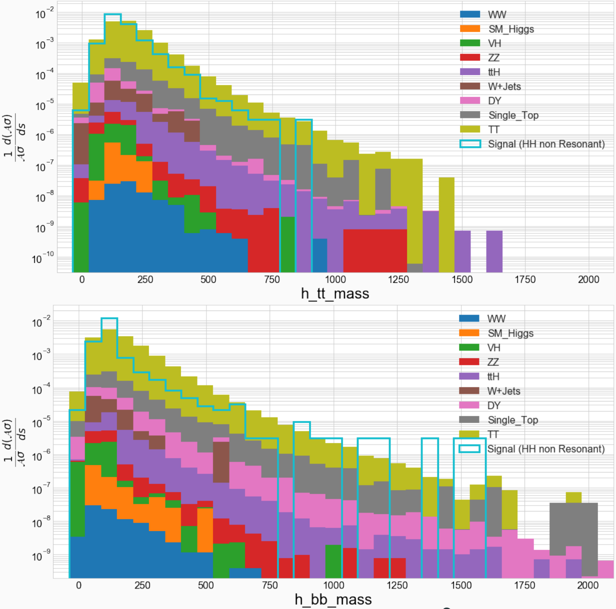


Figure 1: *$h_{\tau\bar{\tau}}$ mass and $h_{b\bar{b}}$ mass [GeV/c^2] ($\mu\,\tau_h\,b\,b$ channel)*

Feature selection

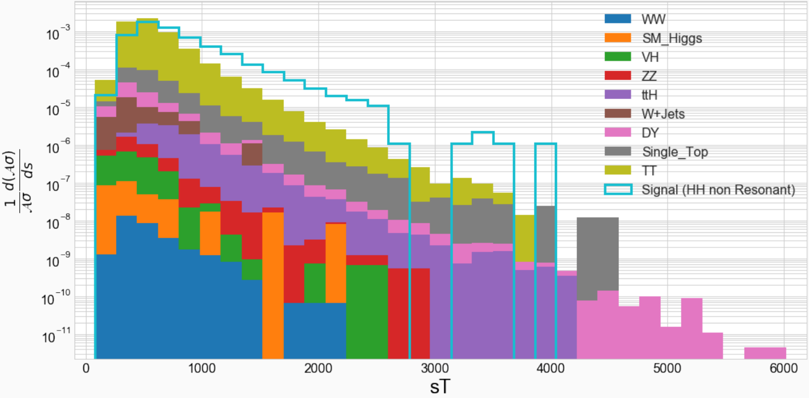


Figure 2: s_T $(\mu\, \tau_h\, b\, b$ *channel)*

Evaluation

- Performance was evaluated using the AMS (approximate median significance):
  - → approximation of the significance, more accurate than $signal/\sqrt{background}$
  - → background uncertainty accounts for the statistical uncertainty and assumes a 10 % systematic uncertainty on normalisation
  - cut is required to accept at least 10 background events in order to ensure correct statistical uncertainties
- Final result uses binned prediction in Higgs Combine, not only to calculate significance but also limits.

Improvements

- Three different activation functions tested:
  - $\rightarrow$ ReLU [1]
  - $\rightarrow$ SELU [2]
  - $\rightarrow$ Swish-1 [4]
- Learning rate finder
- Learning Rate schedules: Cyclical LR and Cosine Annealing for decaying LR
- Data Augmentation (ϕ and/or axis symmetry)

Learning Rate Finder

Learning Rate Finder

- Choosing the right learning rate improves training time and convergence:
  - A tiny learning rate leads to underfitting: the model cannot adequately capture the underlying structure of the data
  - A high learning rate leads to overfitting: the model corresponds too closely to the training data, and may therefore fail for additional data

Learning Rate Finder

- To find the optimal value, the model is trained while the learning rate is increased from a small value.
- The loss calculated on the validation data is evaluated.
- According to Smith (2015) [5], the optimal learning rate is the highest at which the loss is still decreasing.

Learning Rate Finder

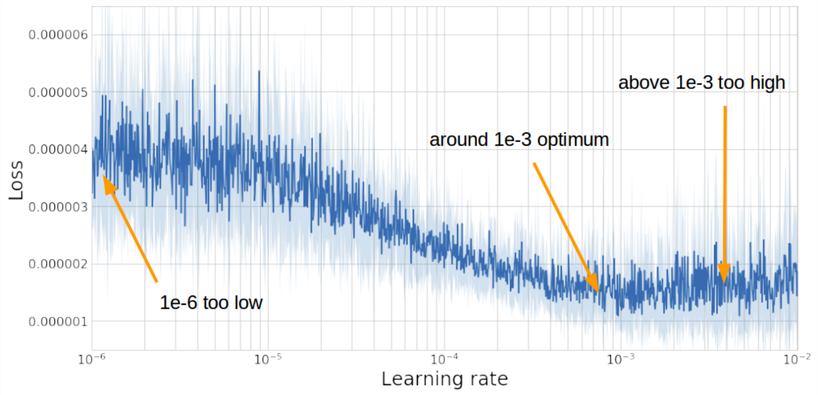


Figure 3: *Loss on validation data in function of the learning rate for SELU, using Cross Validation on 10 folds*

Learning Rate Finder - results

- 1×10^{-3} chosen as the optimum learning rate
- Three different activation functions were tested:

	ReLU	SELU	Swish-1
AMS	0.9018	1.8417	0.9974
Threshold	0.9906	0.9988	0.9943

Table 1: *AMS and cut using each activation function in an ensemble of 10 classifiers and setting the learning rate to the optimum value found.*

- SELU performed clearly better
- It was the activation function used when performing the following tests

Learning Rate Scheduling

Learning Rate Scheduling

- It's common to adjust the learning rate during training, decreasing it once the validation loss becomes flat
- Recent papers (Smith 2015) [5] suggest:

  $\rightarrow$ cycling between low and high bounds using triangular function
- Loshchilov & Hutter (2016) [3] take this further and introduce the cosine annealing

- In cosine annealing schedule the learning rate decays as a cosine function, restarting once it reaches zero.

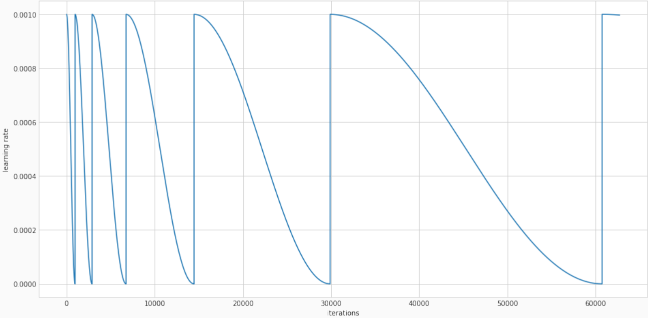


Figure 4: *Cosine Annealing schedule with multiplicity 2 and* 1×10^{-3} *learning rate*

Cosine Annealing

- A multiplicity factor of 2 and an initial learning rate of 1×10^{-3} were used

	Const. Learning Rate	Cosine Annealing
AMS	1.8417	2.6681
Threshold	0.9988	0.9991

Table 2: *AMS and cut using a constant learning rate and a cosine annealing schedule in an ensemble of 10 classifiers.*

Data Augmentation

Data Augmentation

- By performing rotations of the events over an angle ϕ and axis symmetries, "new" data can be created, without changing the underlying class

	with Data Aug.	without Data Aug.
AMS	2.7147	2.6681
Threshold	0.9992	0.9991

Table 3: *AMS and cut with and without a data augmentation routine.*

Final Classifier Predictions

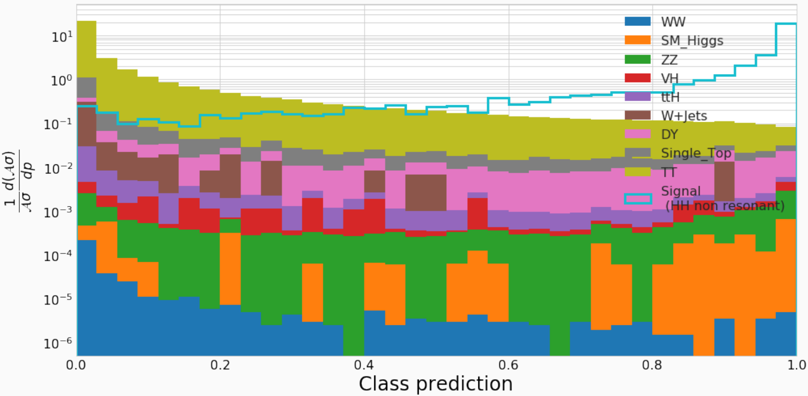


Figure 5: *Class predictions*

Summary

- DNN proved to be a good method to distinguish signal and background in the context of this problem
- New techniques were implemented successfully:
  - → Learning rate finder
  - → Cosine annealing schedule
  - → Data Augmentation
- We were able to predict the expected discovery significance of non-resonant di-Higgs production in HL-LHC using the CMS detector with its proposed upgrades
- We are preparing an analysis note describing our methods and results, to be ultimately included in the Yellow Report

Backup Slides

Event selection

Channels and selection

- Three channels: $\mu\,\tau_h\,b\,b$, $e\,\tau_h\,b\,b$, and $\tau\,\bar{\tau}\,b\,\bar{b}$
- $\mu\,\tau_h\,b\,b$ ($e\,\tau_h\,b\,b$) requires:
  - Exactly: 1 primary muon (electron), 0 veto muons, and 0 veto electrons
  - At least 1 hadronic tau of opposite charge to primary lepton (highest p_T tau chosen in case of multiple)
  - At least 2 b-jets (select pair with invariant mass closest to 125 GeV)
- $\tau\,\bar{\tau}\,b\,\bar{b}$ requires:
  - Exactly: 0 veto muons and 0 veto electrons
  - At least 2 hadronic taus of opposite charge (highest p_T taus chosen in case of multiple)
  - At least 2 b-jets (select pair with invariant mass closest to 125 GeV)

Object definitions

Lepton	Min. p_T [GeV]	Max. $\|\eta\|$	Max. iso [GeV]
Primary μ	23	2.1	0.15
Primary e	27	2.1	0.1
Veto e/μ	10	2.4	0.3
Hadronic tau	**Min. p_T [GeV]**	**Max. $\|\eta\|$**	
$\ell\,\tau_h\,b\,b$	20	2.3	
$\tau\,\bar{\tau}\,b\,\bar{b}$	45	2.1	

- Jets (b and τ) are taken from the `JetsPUPPI` collections
- b jets are defined using the medium working point with the mid timing detector and required to meet: $p_T > 30\,\mathrm{GeV}$ and $|\eta| < 2.4$
- Missing p_T, muons, and electrons are taken from the `PuppiMissingET`, `MuonLoose`, and `Electron` collections, respectively, i.e. the `CHS` versions are not used

Backgrounds used to train

- $t\,\bar{t}$
- Single Top
- Di-boson ZZ
- Drell-Yan to di-Lepton
- ttH

AMS - Approximate Median Significance

$$R = 2 * (((s+b) * \log((s+b) * (b+\sigma)/(b^2 + (s+b) * \sigma))) \\ - (b^2/\sigma * \log(1 + (\sigma * s/(b * (b+\sigma)))))) \quad (1)$$

$$AMS = \sqrt{R} \quad (2)$$

- s and b: unnormalized true positive and false positive rates, respectively
- σ: product of background uncertainty and false positive rate

Feature importance

h_bb_mass	0.1786212980747223
t_1_mT	0.14690006375312806
t_1_mass	0.10203024595975876
h_tt_mass	0.07916492968797684
h_tt_pT	0.06658982336521149
t_0_pT	0.0640179842710495
diH_mT2	0.06373308822512627
t_0_mass	0.03829675018787384
diH_mass	0.03458205610513687
h_bb_pT	0.02743470259010792
nJets	0.02400735765695572
t_0_mT	0.01972164325416088
diH_pT	0.018863273970782756
mPT_pT	0.018004904687404632
sT	0.012860003858804703
b_0_pT	0.012574289739131928
dShapeP	0.012289392948150634
b_1_mass	0.012003678735345602
nTauJets	0.010288166627287865
nBJets	0.009431023709475994
b_0_mass	0.0054306150414049625
aplanorityP	0.004858777765184641
t_1_pT	0.004572654701769352
maxJetPT	0.0042869404423981905
hT	0.0031436745543032885
maxJetEta	0.00285836907569319
minJetEta	0.002571428520604968
t_0_eta	0.0017155119450762868
b_0_eta	0.0017146944534033536
b_1_pT	0.0017146944534033536
aplanarityP	0.001714285695925355
maxJetMass	0.0017142856726422907
meanJetPT	0.0014289801707491278
meanJetEta	0.001143265888094902
minJetMass	0.0008575516054406762
centrality	0.0008575516054406762
diH_eta	0.0008571428479626775
sphericityA	0.0008571428246796131
mPT_phi	0.0005722460802644492
b_1_phi	0.0005718373227864504
meanJetMass	0.0005718373227864504
t_0_phi	0.0005714285653084517
upsilonA	0.0005714285653084517
spherocityA	0.0005714285653084517
spherocityP	0.0005714285653084517
b_1_eta	0.0002861230401322246
minJetPT	0.0002857142826542258
t_1_eta	0.0002857142826542258
h_bb_phi	0.0002857142826542258
eVis	0.0002857142826542258
diH_phi	0.0002857142826542258
dShapeA	0.0002857142826542258
sphericityP	0.0002857142826542258

Features - s_T i

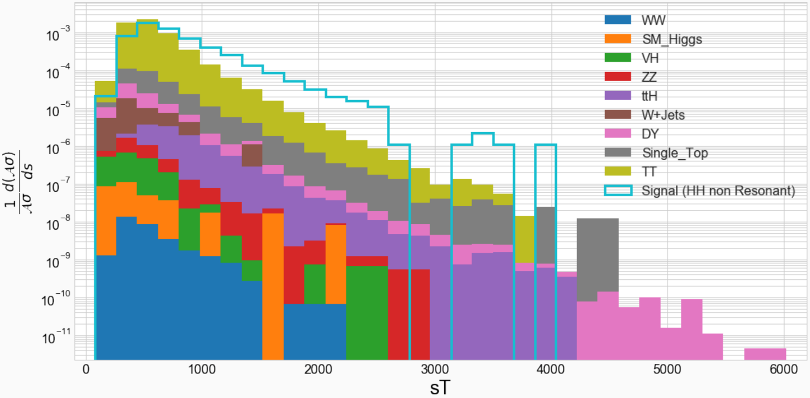


Figure 6: $\mu\,\tau_h\,b\,b$ *channel*

Features - s_T ii

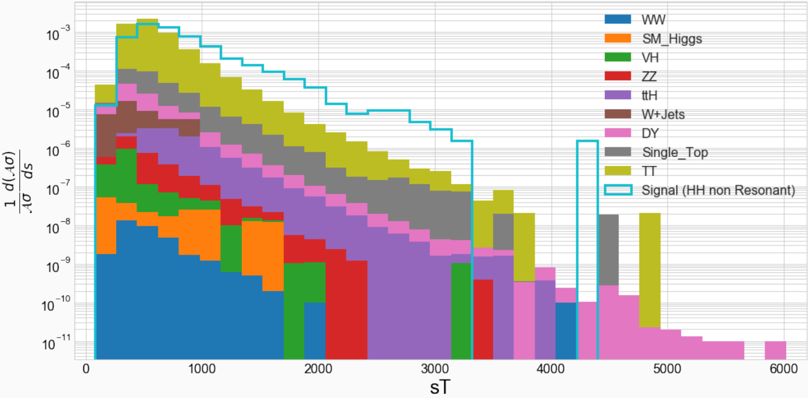


Figure 7: $e\,\tau_h\,b\,b$ *channel*

Features - s_T iii

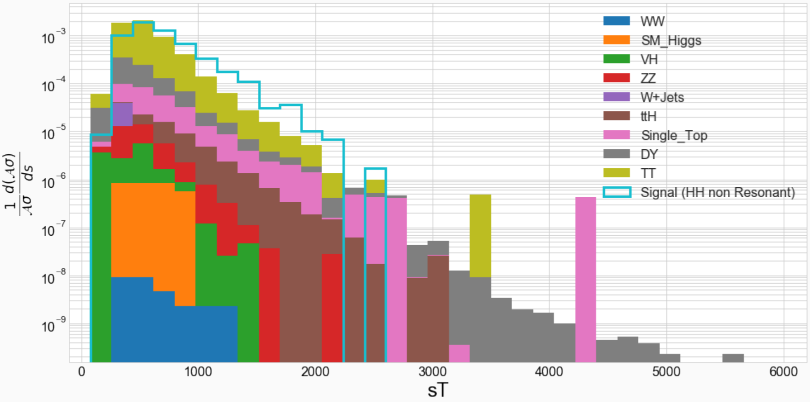


Figure 8: $\tau\,\bar{\tau}\,b\,\bar{b}$ *channel*

Features - p_T of μ, e, τ i

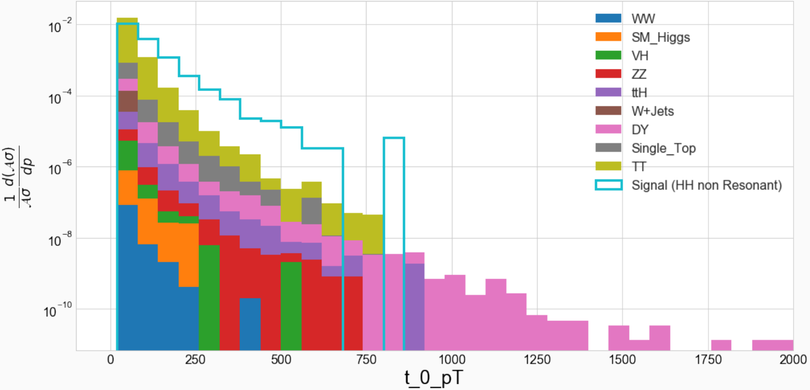


Figure 9: $\mu\,\tau_h\,b\,b$ *channel*

Features - p_T of μ, e, τ ii

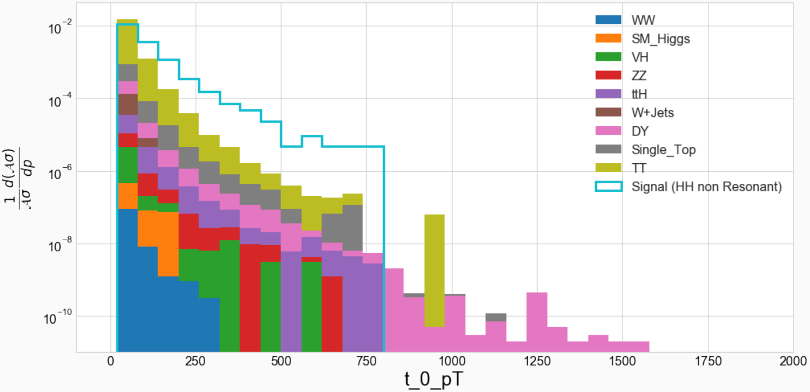


Figure 10: *$e\,\tau_h\,b\,b$ channel*

Features - p_T of μ, e, τ iii

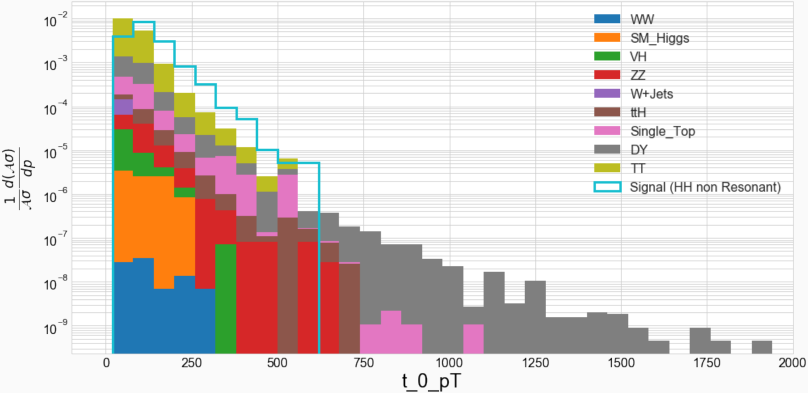


Figure 11: $\tau\,\bar{\tau}\,b\,\bar{b}$ *channel*

Features - $h_{\tau\bar{\tau}}$ mass i

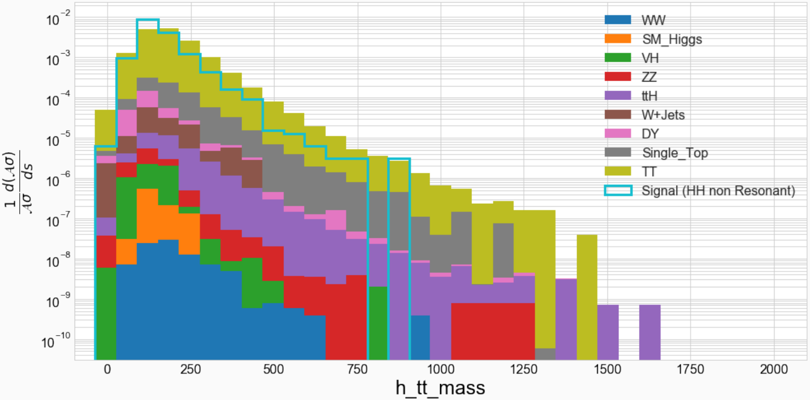


Figure 12: $\mu\,\tau_h\,b\,b$ *channel*

Features - $h_{\tau\bar{\tau}}$ mass ii

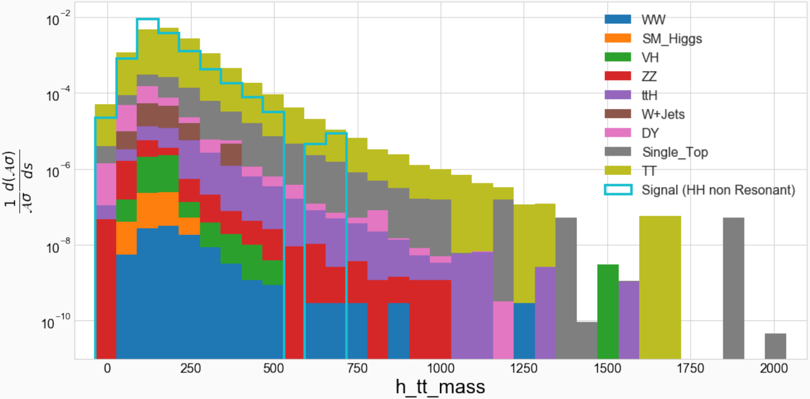


Figure 13: $e\,\tau_h\,b\,b$ *channel*

Features - $h_{\tau\bar{\tau}}$ mass iii

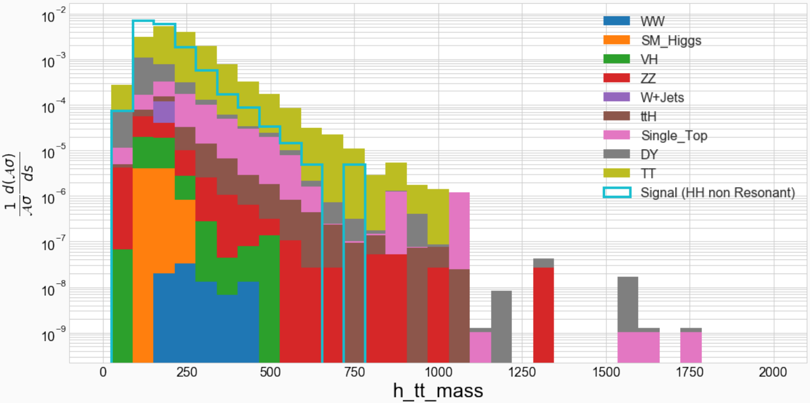


Figure 14: $\tau\,\bar{\tau}\,b\,\bar{b}$ *channel*

Features - $h_{\tau\bar{\tau}}$ mass (linear) i

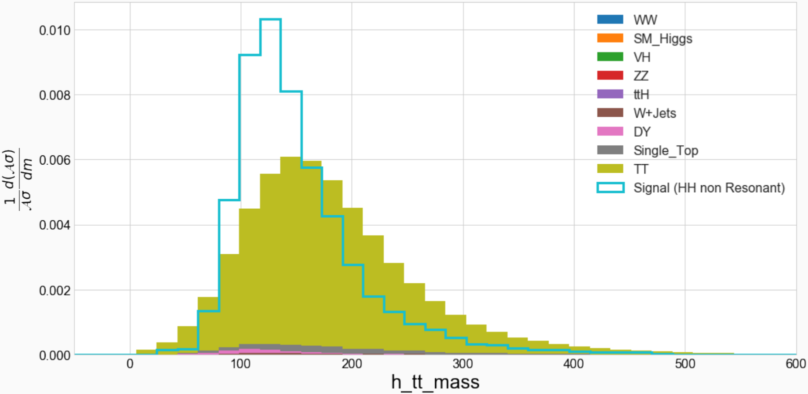


Figure 15: $\mu\,\tau_h\,b\,b$ *channel*

Features - $h_{\tau\bar{\tau}}$ mass (linear) ii

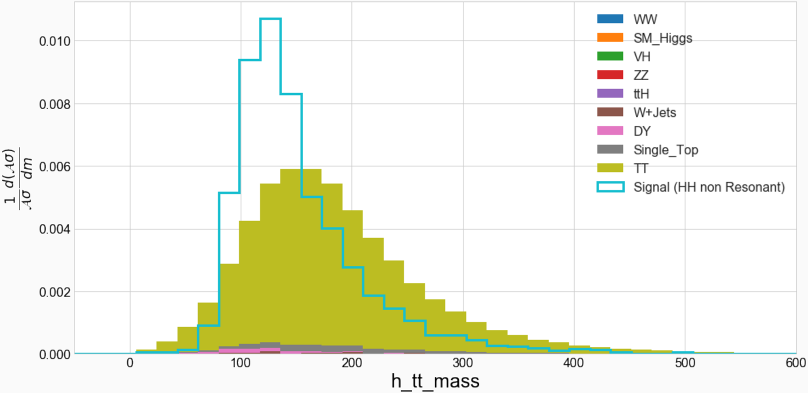


Figure 16: *$e\,\tau_h\,b\,b$ channel*

Features - $h_{\tau\bar{\tau}}$ mass (linear) iii

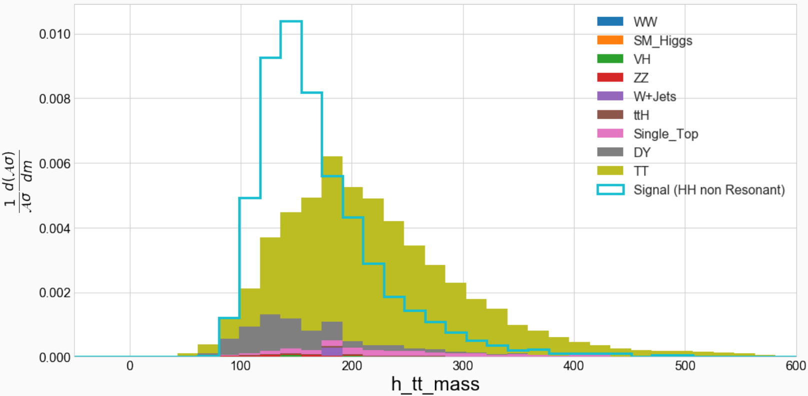


Figure 17: $\tau\,\bar{\tau}\,b\,\bar{b}$ *channel*

Features - $h_{b\,\bar{b}}$ mass i

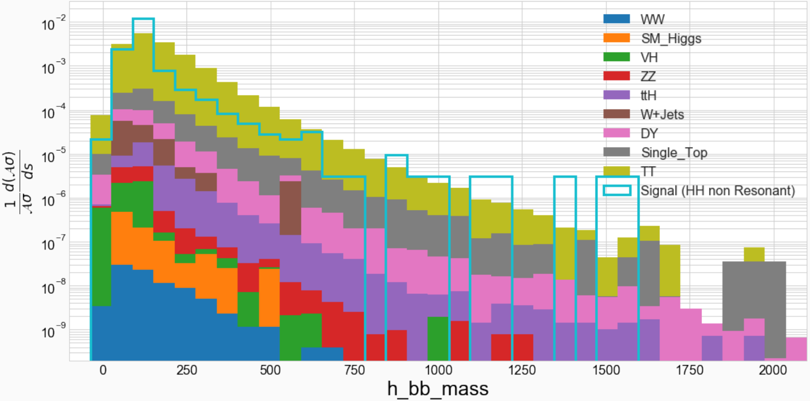


Figure 18: *$\mu\,\tau_h\,b\,b$ channel*

Features - $h_{b\bar{b}}$ mass ii

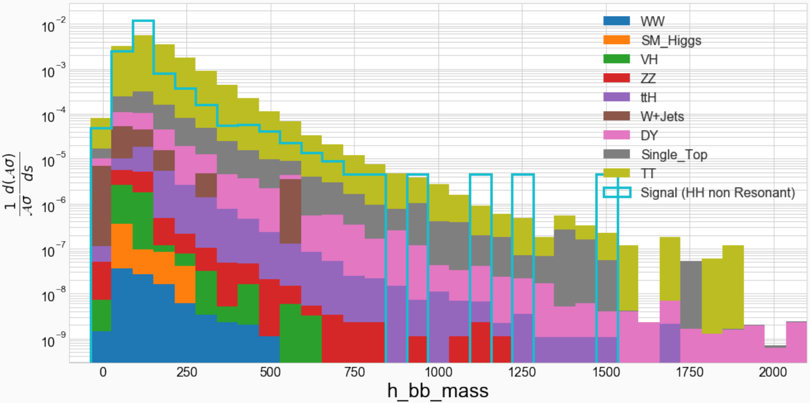


Figure 19: *$e\,\tau_h\,b\,b$ channel*

Features - $h_{b\,\bar{b}}$ mass iii

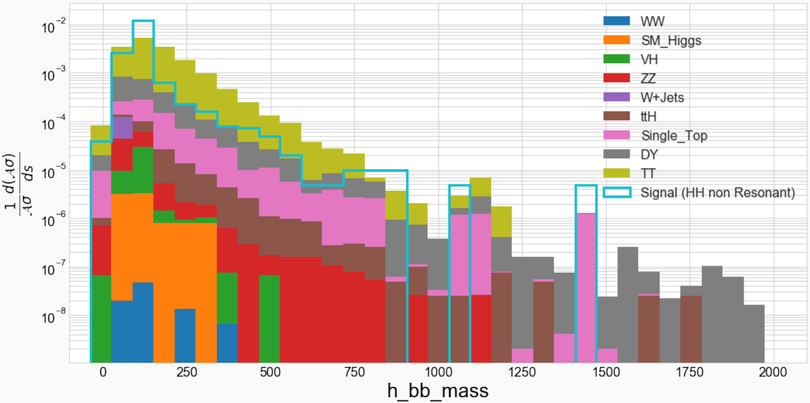


Figure 20: $\tau\,\bar{\tau}\,b\,\bar{b}$ *channel*

Features - $h_{b\bar{b}}$ mass (linear) i

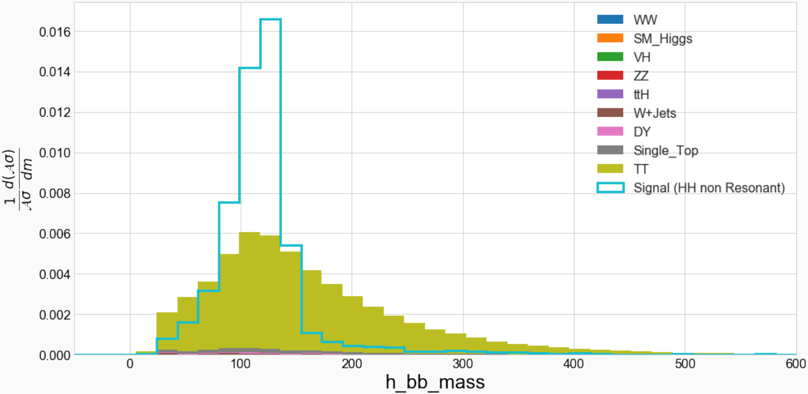


Figure 21: $\mu\,\tau_h\,b\,b$ *channel*

Features - $h_{b\,\bar{b}}$ mass (linear) ii

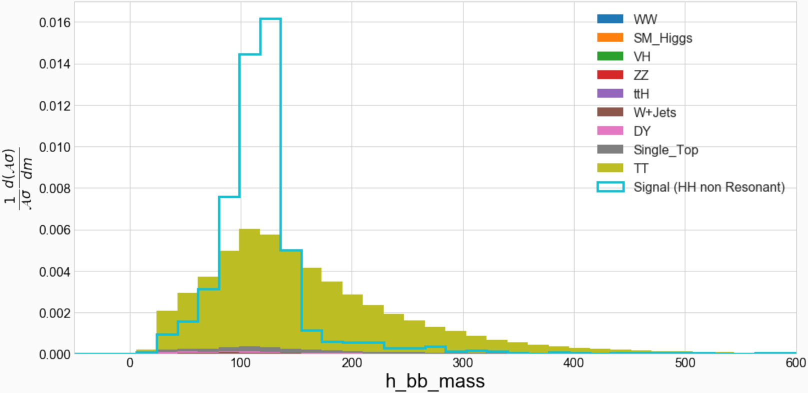


Figure 22: *$e\,\tau_h\,b\,b$ channel*

Features - $h_{b\,\bar{b}}$ mass (linear) iii

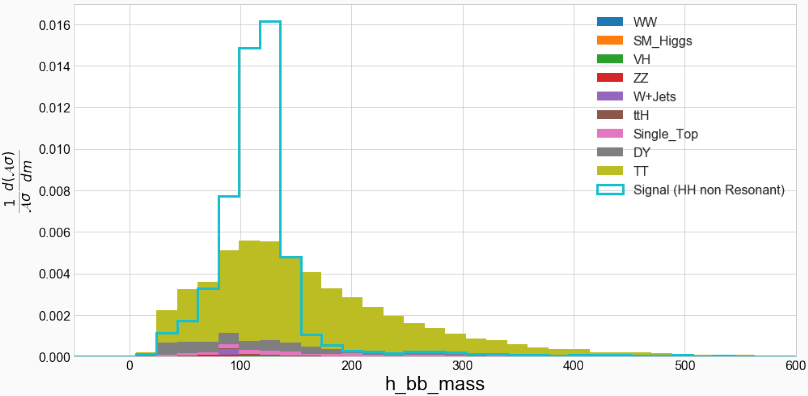


Figure 23: $\tau\,\bar{\tau}\,b\,\bar{b}$ *channel*

Final Classifier Predictions (linear)

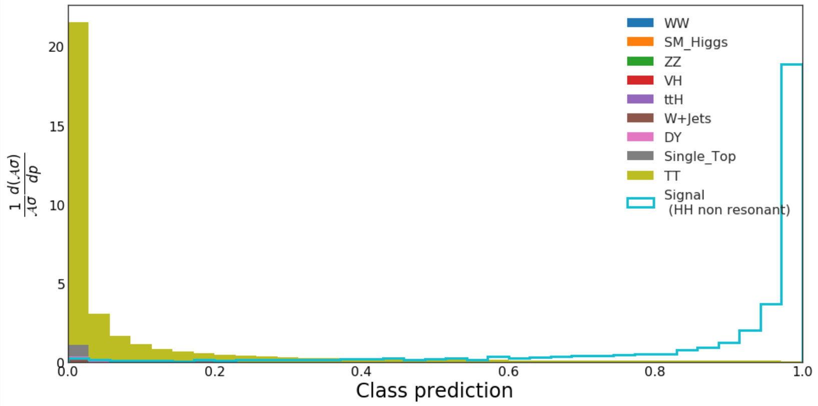


Figure 24: *Class predictions*

References i

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee.
Understanding deep neural networks with rectified linear units.
CoRR, abs/1611.01491, 2016.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter.
Self-normalizing neural networks.
CoRR, abs/1706.02515, 2017.

Ilya Loshchilov and Frank Hutter.
SGDR: stochastic gradient descent with restarts.
CoRR, abs/1608.03983, 2016.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le.
Searching for activation functions.
CoRR, abs/1710.05941, 2017.

Leslie N. Smith.
No more pesky learning rate guessing games.
CoRR, abs/1506.01186, 2015.