# ttH(bb) search at ATLAS: Studying Training Bias in BDT Algorithm

Ariel Kuperman, University of Melbourne, Australia

Supervisors: Dr. Paul Glaysher, Dr. Judith Katzy

DESY ATLAS Group

September 5, 2018

### Abstract

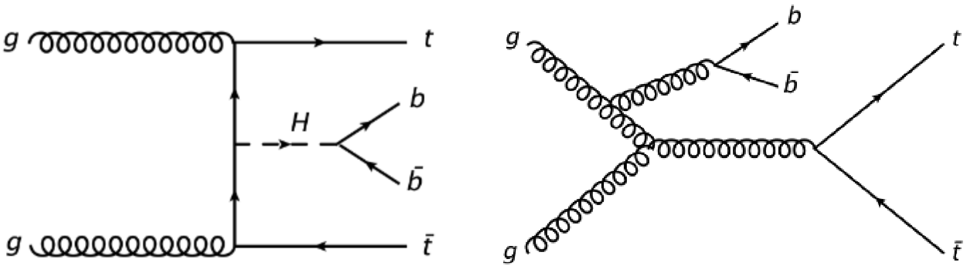
The ttH(bb) search at ATLAS suffers from a prohibitively small signal-to-background ratio. This led to the study of multivariate analysis tools such as Boosted Decision Trees, in order to reduce the uncertainty in the measured signal-strength parameter. The systematic bias introduced in the BDT from the choice of background event generator was investigated, by analyzing the effect of penalizing poorly-modelled regions of phase-space. The method was found to be partially effective against starkly different choices of generator, with scope for further improvement and development outlined.

# Contents

# 1. Introduction

## 1.1. Motivation

The detection of the Higgs boson at the Large Hadron Collider (LHC) in 2012 heralded a rich new era of precision High Energy Physics, wherein probing the Higgs sector allows us to explore the boundaries of the Standard Model and verify its predictions. In particular, measurement of the Yukawa coupling between the top quark and the Higgs boson - the two heaviest particles in the Standard Model - may serve as a probe into the nature of the Higgs field, and has garnered interest in recent years as a measurement potentially sensitive to Physics Beyond the Standard Model. In so being, the focus of this analysis is the Higgs boson production channel associated with two top quarks (referred to as ttH), shown in Fig 1a.



(a) ttH(bb) signal

(b) tt+bb dominant background

Figure 1: Relevant Feynman diagrams in the ttH analysis.

This particular production mode enables the direct investigation of the Top-Higgs Yukawa coupling, as opposed to indirect measurements such as those involving a virtual top loop, common in other Higgs production channels. The Standard Model predicts a contribution of only around 1% from the ttH production mode to the total Higgs boson production cross-section. As such, the Higgs boson decay to b-quark jets is selected in this analysis, as it amounts to the largest branching ratio for the Higgs boson's decay, at $\Gamma(\mathrm{H} \rightarrow \mathrm{bb}) = 58\%$ [1].

The dominant background in this analysis is top-quark production with associated b-jets, shown in Fig 1b. The background is estimated to be 3 orders of magnitude larger than the signal [2], which yields a significantly small signal-to-background ratio. This, coupled with the extreme similarity of final state product signatures, poses a major challenge in detecting and classifying events from the large plethora of data collected at the LHC. These considerations thus provide the motivation for introducing multivariate analysis techniques, discussed further in section 2.

## 1.2. The ttH(bb) Analysis at ATLAS

The ATLAS search for the ttH(bb) signal was performed in 2017 utilizing 36.1 $\text{fb}^{-1}$ of $pp$ collision data at $\sqrt{s} = 13$ TeV, collected at the LHC in 2015 and 2016. In May, 2018, the Collaboration announced the ratio of measured ttH(bb) signal cross-section to the Standard Model expectation to be $\mu = 0.84^{+0.64}_{-0.61}$, the large uncertainty found to stem largely from the uncertainty on the Monte Carlo modelling of the tt+bb background [2].

The full analysis involved sub-classifying the data into separate regions of phase-space, based on the events' number of jets and the number of b-tagged jets. In the present study we focus solely on the most sensitive signal region, namely the semileptonic decay channel (ie, containing only one electron or muon from the top decay), and identified to contain at least 6 jets and at least 4 b-tagged jets, consistent with the expected decay product profile (shown in Fig 2). We specialize to this region in order to test features of the multivariate analysis step with the highest possible signal statistics.

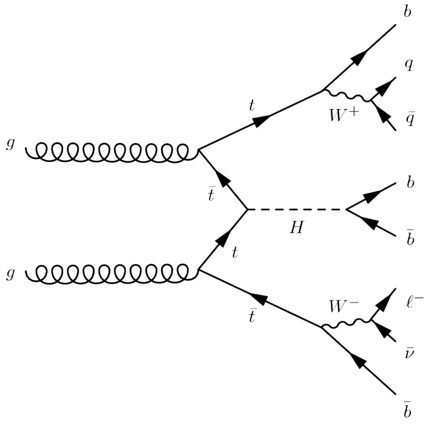


Figure 2: ttH(bb) nominal decay mode: 6 hadronic jets with 4 b-tagged jets

The prohibitively small signal-to-background ratio in this analysis engenders the need for multivariate analysis techniques, such as Boosted Decision Trees (BDTs), in order to classify signal and background events. The BDTs are generated ('trained') with simulated Monte Carlo data, and are then applied to the real and simulated datasets. The output of the BDT may then be utilized to perform a full fit of the theoretical simulation to the data, and thus extract the so-called 'signal-strength parameter' $\mu$ defined above. An example of the output of this procedure is shown in Figure 3, published by the ATLAS Collaboration. Even in the region with most classified signal data (shown in red) on the rightmost bin in the plot, the uncertainty in the background (shown as a hashed histogram) is of the same order as the classified signal, which clearly obscures the possibility of making a precise measurement. Consequently, reducing the systematic errors introduced via the BDT is of paramount importance in reducing the error in $\mu$.

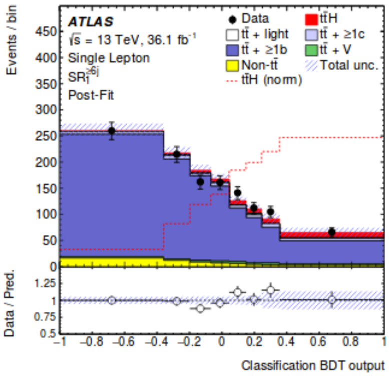


Figure 3: BDT output showing a comparison between theoretical prediction and data [2]. The real data is shown as black dots, predicted background in blue, predicted signal in red, and uncertainty on the background as a hashed histogram. The size of the uncertainty on the background obscures any prospect of obtaining precise signal measurements, motivating the study of this uncertainty.

# 2. Multivariate Analysis

## 2.1. Boosted Decision Trees

A decision tree is a machine learning technique utilized to detect patterns in data. It operates as a binary classifier, in which a data set is successively split in an attempt to maximally classify the entries as either one of two classes, in our case ‘Signal’ or ‘Background’. At each branching point (‘node’) in the tree, the classifier detects which variable and value would most optimally yield a cut (by minimizing a pre-determined loss function), and thus branches of the tree are created according to whether events lie above or below this cut-off value. This is performed iteratively until a pre-determined depth, or until the data set is completely classified into one of the classes. A schematic decision tree is shown in Figure 4.

In order to generate a decision tree, the data set is generally split into a **training** and a **testing** subset, whose true classification is already previously known. The training subset is utilized to generate the tree architecture, and the testing subset is used to verify the efficacy of the tree’s classifications. A perfect decision tree would thus classify all of the testing dataset with exact correspondence to their true classification.

The advantage of utilizing decision trees as a multivariate analysis tool is their simplicity and comprehensibility, in comparison with other machine learning techniques. However, decision trees are also relatively unstable with respect to statistical fluctuations in the training data, and are generally regarded as weak classifiers [3]. Hence, one generally constructs a stronger classifier by combining many decision trees, which may be achieved through a variety of methods. We focus in particular on Boosted Decision Trees, which are frequently utilized in High Energy Physics statistical analyses.

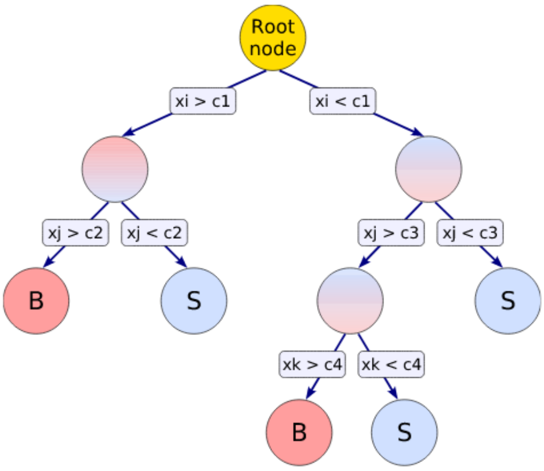


Figure 4: Schematic Decision Tree [3]. At each node, the optimal variable x and cut-off c are determined and data is classified along the branches of the tree. Each event begins at the root node and eventually reaches a final node ('leaf'), where the data is classified as either Signal or Background.

Boosting involves combining multiple decision trees (in the order of hundreds) through an effective weighted average of classification outputs. Most common boosting algorithms generate decision trees successively by accounting for the mistakes that the previous classifier makes, as demonstrated in Figure 5. Upon combining many weak classifiers, one is thus able to obtain a stronger classifier, namely the Boosted Decision Tree (BDT).

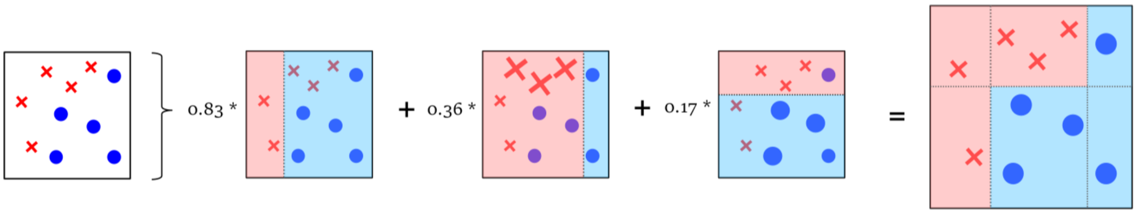


Figure 5: Schematic BDT for classifying data as either a red cross or a blue dot. The data set is classified across the two-variable phase space by multiple successive decision trees, each taking the mistakes of the previous classifier into account as it performs its own classification. The larger data points represent a larger weighting given to the corresponding events upon classification. Upon combining ('boosting') all weak classifiers, we obtain a stronger classifier.

The resulting quantity obtained is the BDT Discriminant Response, also called the Classification Score. This score, measured from -1.0 to 1.0, is a probabilistic measure of an event being signal-like (output = 1.0) or background-like (output = -1.0). A sample

BDT output distribution is shown in Figure 6. An ideal classifier would thus classify all true signal events (shown in blue) as signal-like, with a distribution skewed mostly to the right, and all true background events (shown in red) as background-like, with the corresponding distribution skewed mostly to the left. Indeed, the overlap of the two distributions may serve as an indication for the performance of the BDT classification, which is further discussed in Section 2.2.

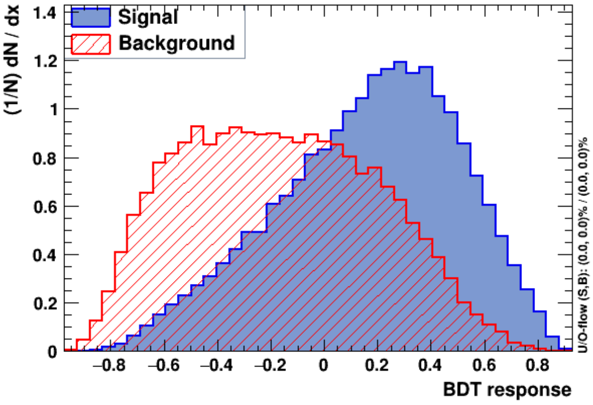


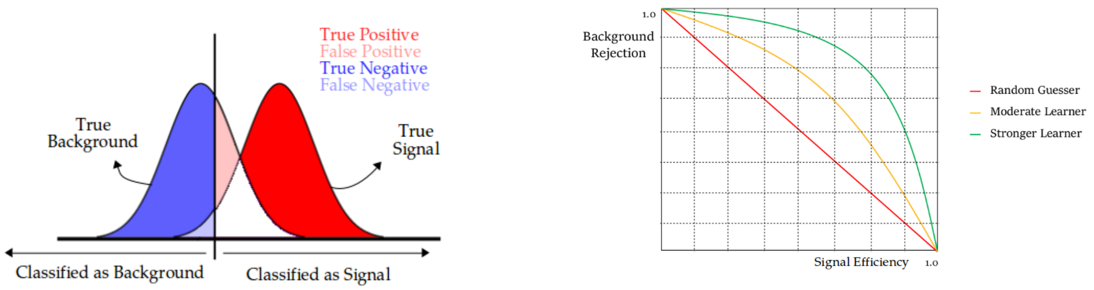
Figure 6: BDT Classification Score for ttH signal and tt+bb background. The blue (red) histograms correspond to normalized distributions of true signal (background) event estimates.

The Boosting algorithm utilized in this analysis is the AdaBoost scheme, developed by Freund and Schapire (1996) and implemented within the TMVA Multivariate Analysis Tool interfaced within Root, the nominal data analysis software utilized in HEP. We make this choice of Boosting algorithm due to its proven success as a robust boosting scheme [4].

## 2.2. Classifier Performance

Once a BDT is trained, its performance may be evaluated by analyzing its Receiver Operating Characteristic (ROC) curve, or analogously the area under the ROC curve (AUROC). The ROC Curve is a measure of the BDT's background rejection versus its signal efficiency. Signal efficiency is defined as the ratio of correctly classified signal events ('true positives') by incorrectly classified signal events ('false negatives'), whereas background rejection is defined as the ratio of incorrectly classified background events ('false positives') by correctly classified background events ('true negatives'). A visual definition of such classification regions is shown in Figure 7a, and examples of different ROC curves may be seen in Figure 7b.

A completely random classifier equally classifies signal and background events correctly and incorrectly. However, an ideal classifier would perform as to maximize the signal efficiency and background rejection. This would be analogous to observing a distinct separation in the two BDT output distributions shown in Figure 6, and is reflected



(a) Classification measures used in calculating signal efficiency and background rejection.

(b) Sample ROC Curves. A random guesser performs equally badly in classifying events correctly or incorrectly; a strong learner pushes the ROC curve towards (1.0, 1.0).

Figure 7: Measuring BDT performance.

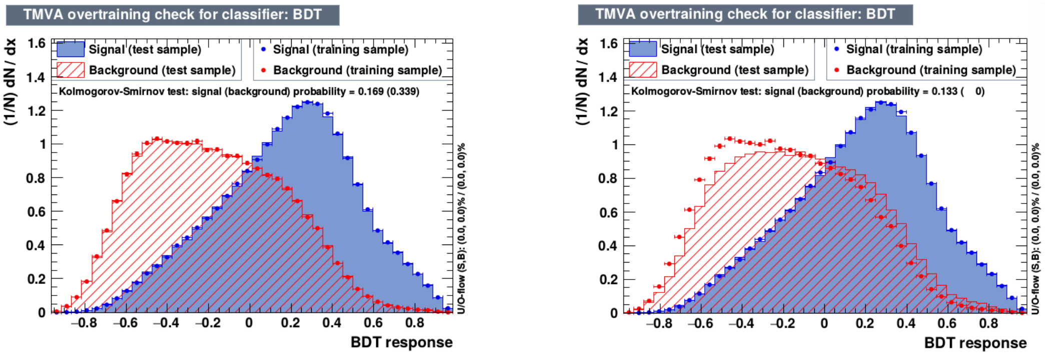
in an AUROC value as close to 1.0 as possible. As such, we may quantify the performance of a BDT as a classifier in terms of an AUROC value: the closer to 1.0, the better the BDT's performance.

## 2.3. Overtraining and Bias

A highly important feature to take into account when generating and utilizing BDTs is the risk of **statistical overtraining**, which refers to generating decision trees that are too specific to the training data. If the classification becomes over-specialized to the training dataset, the capability to accurately classify other data is lost. This may be checked for by utilizing the testing dataset, and analyzing discrepancies in the BDT outputs between training and testing sets. An example of overtraining is shown in Figure 8.

The overtraining may be quantified by performing a *k-fold Cross-Validation Check* [5]. In general, a dataset is actually split into k randomly sampled subsets (called 'folds'), trained on k-1 folds, and tested with the remaining fold. The variance in the AUROC values for each of the BDT outputs can thus be utilized as an indication for overtraining. In this analysis, however, we refrain from performing such checks, as they have been previously carried out in other work. We hence utilize only visual clues to detect significant changes in statistical overtraining.

Nevertheless, an important source of bias often not accounted for is **systematic overtraining**, and is particularly pernicious in the ttH(bb) analysis. This bias arises from the use of simulated Monte Carlo (MC) data used to train our BDTs, as we do not have access to real-world data whose true classification as 'Signal' or 'Background' is previously known. However, multiple MC event generators exist to model the dominant tt+bb background, none of which model reality with 100% accuracy. As such, making



(a) A BDT displaying no overtraining.

(b) A BDT displaying clear overtraining.

Figure 8: An example of overtraining. The BDT output for the training dataset is shown as a dotted distribution, and for the testing set as filled histograms. On the left we see agreement between the two overlaid distributions, indicating little-to-no overtraining. However on the right a clear discrepancy may be seen, indicating this BDT has been overtrained.

a particular choice of MC event generator induces a systematic overtraining bias in the BDT. In fact, this particular bias was determined to be the leading cause of the uncertainty in the signal-strength $\mu$ published by the ATLAS Collaboration in May 2018 [2]. Seeking to reduce this systematic bias thus became the focus of the present study.

It is also worth mentioning in passing that the aforementioned k-fold Cross Validation Check is blind to systematic overtraining in the BDT, and as such is of no use in mitigating this issue. This leads one to introduce a novel concept for mitigating systematic bias, discussed in detail in Section 3.

# 3. Reducing BDT Systematic Training Bias

## 3.1. Penalty Weighting

The concept behind this study is to introduce **penalty weights** to events that fall in regions of discrepancy between different MC event generators, when training the BDT. Through penalizing such events, poorly-modelled regions of phase-space are given a lower priority in the training, which would consequently reduce the dependence of the BDT classifier on the particular choice of MC generator, and thus mitigate the systematic bias discussed in Section 2.3. Even though some classification power is likely to be sacrificed (given that one would now train a BDT with lower sensitivity to outlying events), the expected reduction in the BDT's model dependence is projected to have a significant impact in reducing the large uncertainty in $\mu$.

This study makes use of the following MC event generators for the tt+bb background:

- `PowHeg+Pythia 8` (PP8)
- `Sherpa`
- `aMC@Nlo` (aMC)
- `PowHeg+Herwig 7` (PoH7)

We take PP8 as the nominal generator in this study, due to its observed closeness to real data observed in previous work. The generators differ primarily in their number of final state partons calculated by the hard process, the algorithms used to develop parton showers, the modelling of multi-parton interactions, and the modelling of hadronisation, among other features. The signal sample for this study is also generated using PP8.

The kinematic variables listed in Appendix A are utilized to train the BDT in this analysis. Previous work has been conducted to investigate and verify this particular choice of training variables [6].

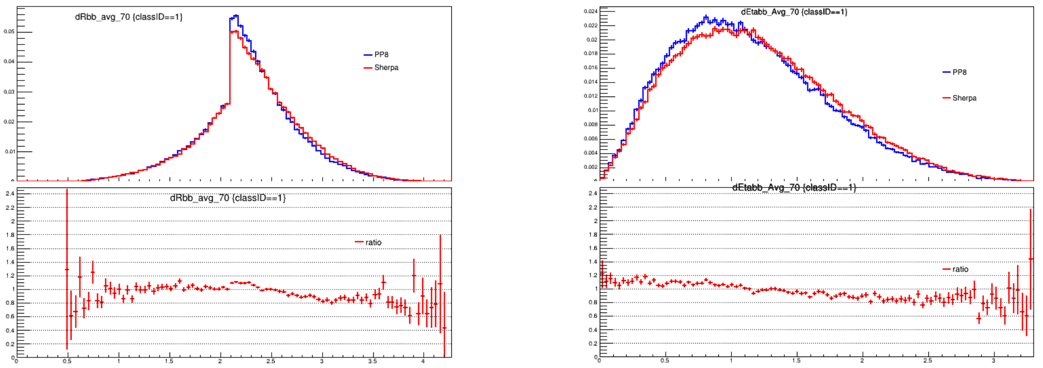
The proposed method to introduce this penalty weighting into the training of the BDT is as follows:

1. Obtain ratios of binned distributions of all input variables from two different generators: the Nominal Generator (PP8) and the Test Generator (Sherpa, aMC or PoH7). These ratios are used to define the penalty weights.
2. For each event:
   - For each variable, determine the corresponding bin the event falls into and assign it the corresponding weight.
   - Compound all weights.
   - Assign the event its overall weight when training the BDT.
3. Train the BDT with and without the penalty weighting, and test with the nominal and test generators without penalty weights, to hopefully observe a reduced discrepancy between the two testing sets.

The above steps are discussed in more detail in the following section.

## 3.2. Implementation

The distributions of each input variable generated by the Nominal and Testing Generators were plotted to the same binning and their ratios taken, an example of which may be seen in Figure 9. The discrepancies between the two generators, in this case PP8 and Sherpa, may be clearly seen: whereas for one variable (Fig. 9a) the generators seem to yield a similar event profile, they yield distinctly different distributions for other variables (Fig. 9b). The calculated ratios are then utilized as the penalty weight for



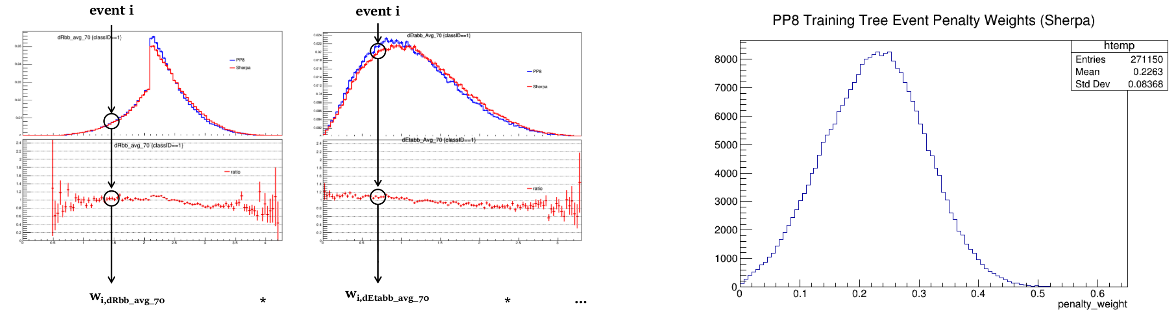
(a) Average ΔR between any two b-jets

(b) Average $\Delta\eta$ between any two b-jets

Figure 9: Obtaining ratios for penalty weighting. Events generated by PP8 are shown in blue, and by Sherpa in red. The bottom half of the diagrams shows the ratios of the distributions, which are utilized to define the penalty weights.

that variable. If a ratio is computed to be above 1, the reciprocal value is taken, such that events are always penalized by the relative difference between the two generators.

Each event is determined to come from a certain bin in the aforementioned distributions, and the corresponding weight for the variable is obtained (Fig. 10a). Upon combining all weights across all variables, the events are assigned an overall total weight, to be used in training the BDT. Figure 10b shows an example distribution of overall penalty weights to be assigned to the BDT training set. Whereas the distributions computed for this analysis do not take the errors in the ratio values into consideration, one should not expect these to be highly significant in the overall analysis, since these regions of high error arise from regions of low bin statistics (which by definition are statistically rare).



(a) Each event is identified in the binned input variable distributions and the corresponding weight from each variable is assigned. All weights are then compounded together.

(b) Overall penalty weight distribution for training BDT, computed using ratios of PP8 against Sherpa.

Figure 10: Assigning penalty weighting to events prior to training the BDT.

Lastly, the third step involved training the BDT with and without penalty weights, and cross-testing. The resulting distributions using PP8 and PoH7 are shown in Figure 11. On the top left we observe the BDT being tested and trained with PP8 and no penalty; it appears to be very robust against statistical overtraining. The top right shows testing with PoH7, whereupon we may notice a discrepancy between training and testing, as expected. However, once we introduce penalty weighting we may see that the testing and training sets for both generators deviate significantly from one another. This reflects the expected reduction in classification power from introducing the penalty weighting to the training dataset. This is explored further in Section 4.1.

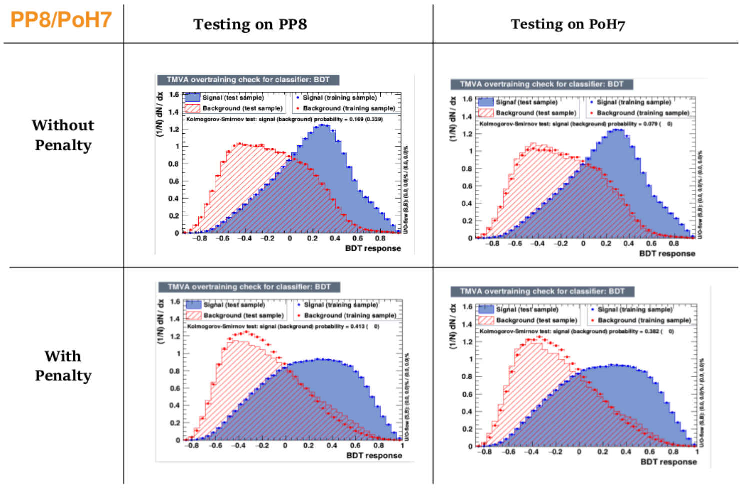


Figure 11: BDT Outputs comparing training using PP8 with and without penalty weights, and testing against PP8 and PoH7. A discussion of quantitative results is deferred to Section 4.1.

# 4. Results and Discussion

## 4.1. BDT Performance

The above methodology was implemented across all three testing generators, and tables similar to Figure 11 generated. The corresponding AUROC values for each testing set is recorded in Table 1, as an estimate for the BDT classification power. In particular, the expectation of reduced performance was confirmed; the BDT generally performs as a worse classifier upon implementing penalty weights in training. However, one may also see that the reduction in its classification power is indeed small (of the order of at most ∼8%) such that this method may still be implemented without compromising the classification.

It is also worthwhile noting that Sherpa and PP8 appear to be the closest generators in terms of BDT classification power, whereas aMC seems to be the most distinct compared to PP8. This matches expectation from known properties of these generators.

| **AUROC** | Test on PP8 | Test on Sherpa | Test on PoH7 | Test on aMC |
|---|---|---|---|---|
| No Penalty | 0.769 | 0.767 | 0.782 | 0.729 |
| With Sherpa Penalty | 0.752 | 0.749 | - | - |
| With PoH7 Penalty | 0.756 | - | 0.770 | - |
| With aMC Penalty | 0.759 | - | - | 0.708 |

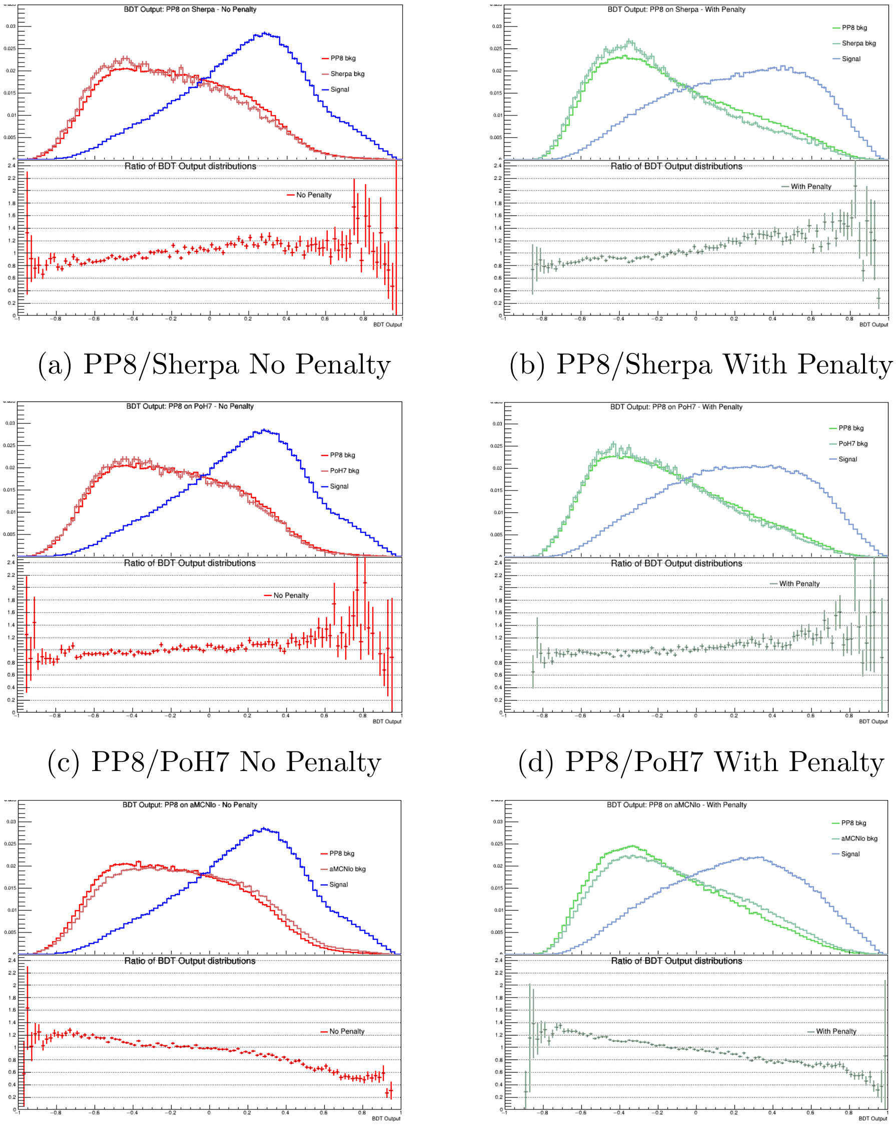
Table 1: AUROC values for BDT classification of testing samples, with and without penalty weights. A clear trend is recognized in that classification power goes down as penalties are added; however the reduction is deemed to be sufficiently small as to not compromise the BDT. Sherpa is recognized as the closest generator to PP8, and aMC as the most distinct.

A visual representation of the BDT output distributions for each of the testing generators is shown in Figure 12. The distributions from PP8 and the alternative generator are overlaid, and the signal distribution is shown in blue. The ratio of the distributions is shown in the bottom half of each plot. The left-hand column represents training without penalties, and the right-hand column with penalty weights added.

The ratio between the two background generator output responses, shown in the bottom half of figures 12a-12f, is an indicator of systematic bias in the BDT. If the ratio were exactly 1.0 (within uncertainty due to limited size of the testing sample), this would be an indication that the algorithm is entirely model independent. Hence, a deviation from 1.0 may serve as a measure for systematic bias in the BDT.

For a better comparison, Figures 13-15 show Figures 12a-12f overlaid for each generator, along with a line of best fit fitted to the ratio plots. It is worthwhile noting that although this line of best fit may serve as a rough indicator of the deviation of the ratios from 1.0, there is no reason to believe that the ratios should follow a linear trend. Hence the lines of best fit are included for completeness, but not for further analysis.

Inspecting the plots displayed in Figures 13-15, one may observe that introducing penalty weighting in training seems to smear the signal distributions towards a more spread-out classification, and the background distributions to be more populated deeper into the background- and signal-like regions. The large number of events in the cross-over region between distributions (from -0.2 to 0.2) is suppressed, and events are pushed further towards more signal- or background-like classification. The cross-over region represents particularly indistinct events, such that the multiple decision trees used in the BDT on average classify them as signal or background with commensurate certainty. This change may be due to the penalty weights effectively eliminating outlying background events, that would otherwise be used by the BDT to discern similar-looking signal and background events. Thus, events are now more coarsely classified by the BDT such that previously indistinct events fall (both correctly and incorrectly) in more signal- or background-like regions. This reflects the expected reduction in the BDT's classification power.



(a) PP8/Sherpa No Penalty

(b) PP8/Sherpa With Penalty

(c) PP8/PoH7 No Penalty

(d) PP8/PoH7 With Penalty

(e) PP8/aMC No Penalty

(f) PP8/aMC With Penalty

Figure 12: Cross Testing Results. The training is always performed with PP8, and BDT outputs of the two background testing sets are overlaid. On the left-hand side no penalty is used in training, on the right penalty weights are applied.

An unexpected effect of introducing penalty weighting was the impact on the signal distributions, despite no direct weighting ever being applied to signal samples. Despite obtaining a more spread distribution, we appear to also obtain a higher signal-to-background ratio in the signal-like regions, which are of primary interest. A promising prospect for future study would be to analyze this ratio as a figure of merit and study its dynamics as different testing generators are used.

Secondly, one may see that introducing penalty weighting in training the BDT is successful for the aMC generator (Fig. 15). In the signal-sensitive region of the BDT classifier output (from ∼0.6 onwards), one may observe a ∼20% improvement in reducing the discrepancy between testing sets. This may directly reflect in a proportional reduction in the uncertainty in $\mu$.

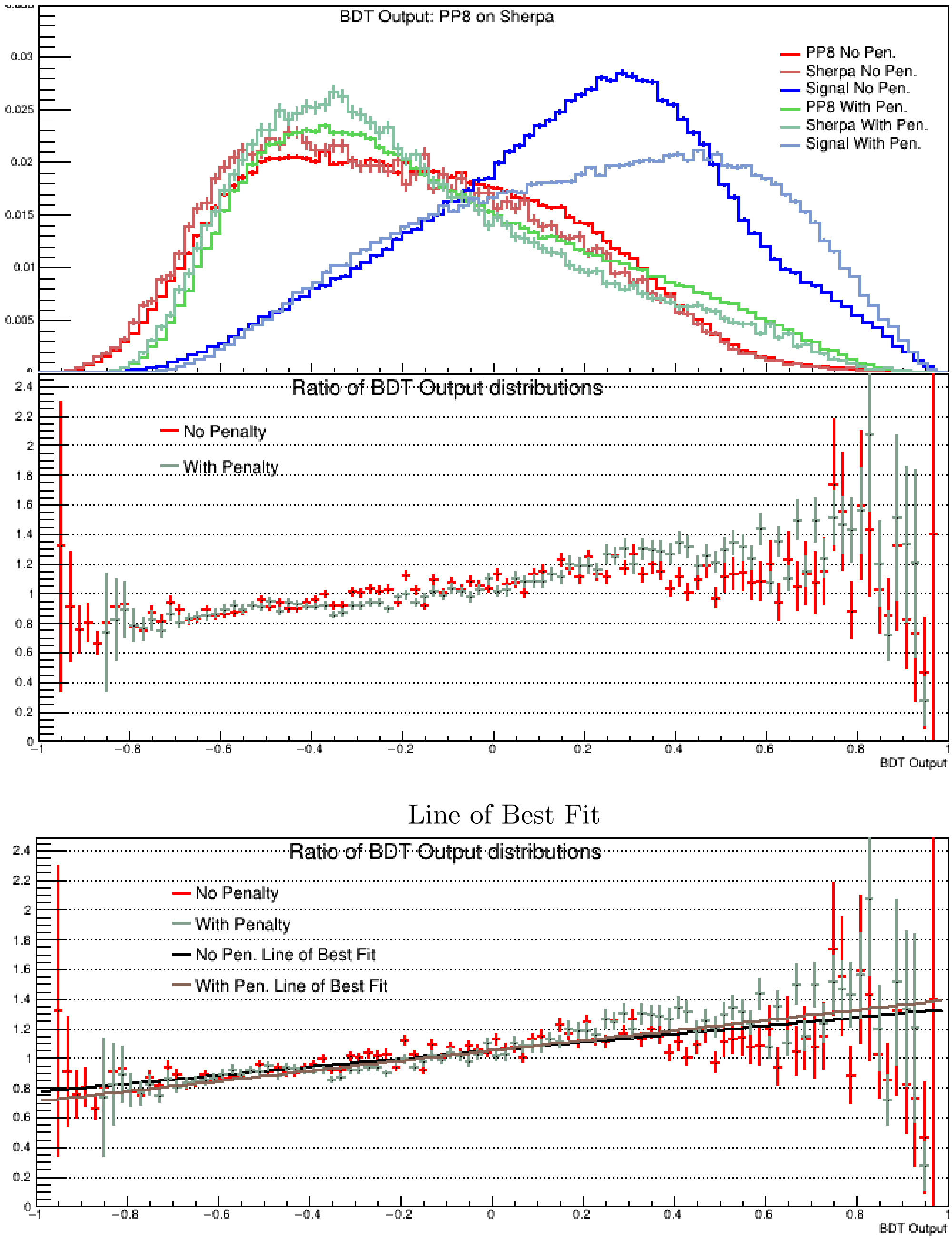
However, the penalty weighting method does not appear to work for Sherpa or PoH7 (Figs. 13, 14). We observe little-to-no improvement in reducing the systematic bias, and since there is a definite loss in classification power as discussed above, this method is rendered unfavourable for these generators. This might be because PoH7 and Sherpa are similar enough to PP8 to begin with, such that the correction caused by the weighting is too coarse to add any differentiating power between the two. The current implementation of the penalty weighting thus seems to overcompensate given a relatively small initial difference between generators.

It is also worthwhile noting the upwards trend in the ratios for Sherpa and PoH7, and the downwards trend in the ratios using aMC. This may be attributed to the initial similarities between Sherpa or PoH7 and PP8, and the relatively large initial difference between aMC and PP8. Since Sherpa and PoH7 are initially similar to PP8, the BDT classifies them as background-like, as it trains on a similar-looking dataset which it calls 'Background'. However, given the difference between PP8 and aMC, the BDT classifies aMC as less background-like, and consequently as more signal-like. Hence, since the ratios are calculated as PP8/Test Generator, we obtain the observed trend. As such, the penalty weighting in its current implementation seems to perform better for initially starkly different generators.

It is interesting to note in Figure 13, however, that the BDT appears to classify a larger proportion of the PP8 cross-over region events as signal-like compared to Sherpa after the penalties are introduced. It thus appears that the BDT performs better in classifying Sherpa events, suggesting that the penalty weighting might be morphing the training set to be more Sherpa-like than PP8-like. However, this does not seem to be the case for aMC, which might possibly be due to the large initial differences in generators. In the case of PoH7, the limited number of statistics obscures the prospect of drawing meaningful comparisons. A prospect for further investigation would be to analyze how different penalty weight algorithms affect this trend, and whether training on a different generator also induces the same observed effects.

The above observations motivate the prospect of refining the weighting algorithm, in order to detect any improvement in Sherpa and PoH7 and to investigate the performance of the BDT in classifying cross-over region events after penalty weights are added. Unfortunately, a detailed analysis lies beyond the scope of the present study. However, a brief attempt at modifying the weighting algorithm was performed, whose result and suggestions for further work are discussed in Section 4.2 below.

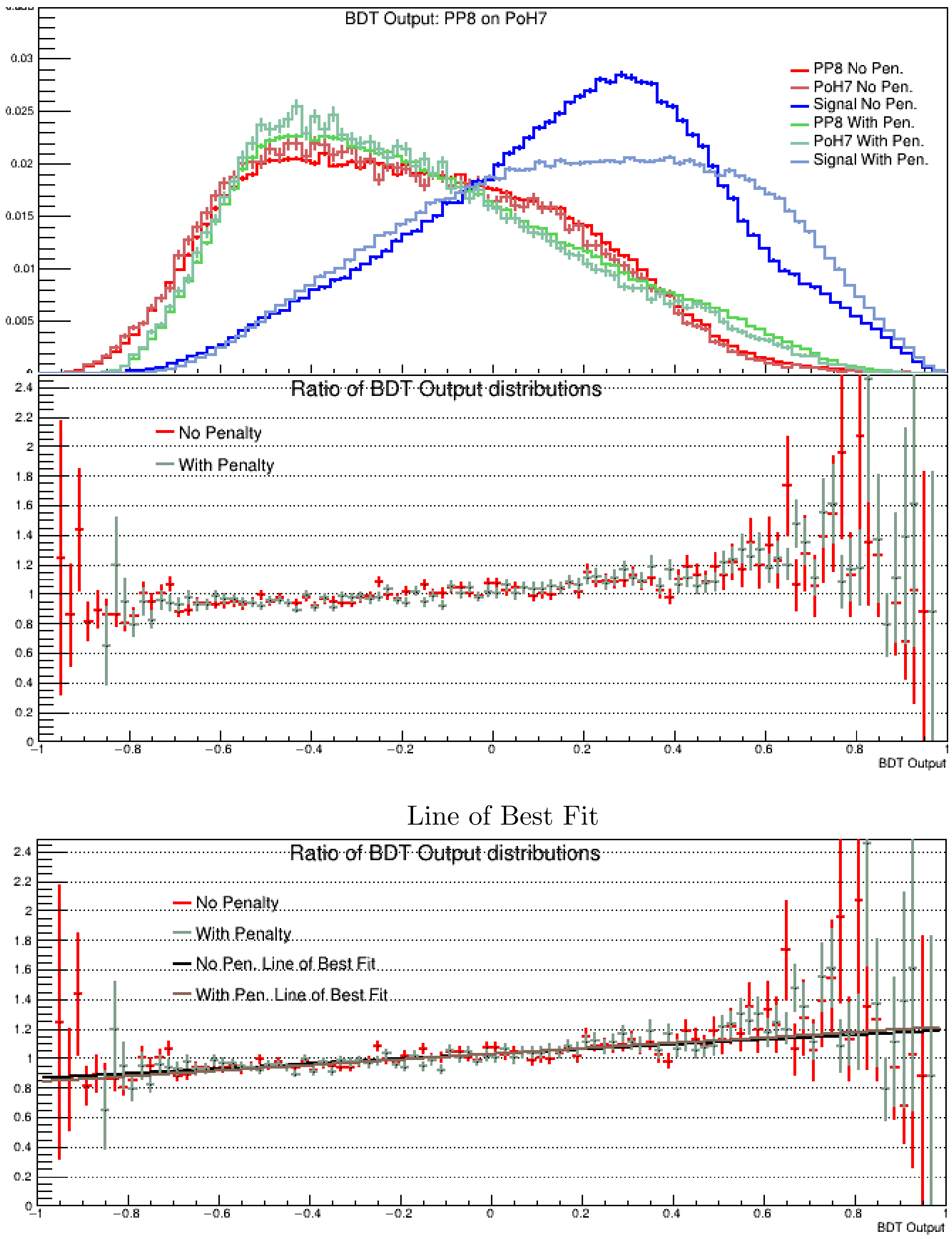
Training: PP8 no penalty, PP8 with penalty (compared to Sherpa). Testing: PP8, Sherpa



Line of Best Fit

Figure 13: Top: Results of BDT Output for **PP8/Sherpa** with and without penalty weights, overlaid. Middle: ratios of background distributions with and without penalty weights, overlaid. Bottom: ratios with added line of best fit. The distributions indicate the penalty weighting mostly impacts the cross-over region around [-0.2, 0.2], as discussed in the text. The penalty method appears to be ineffective in combating systematic bias.

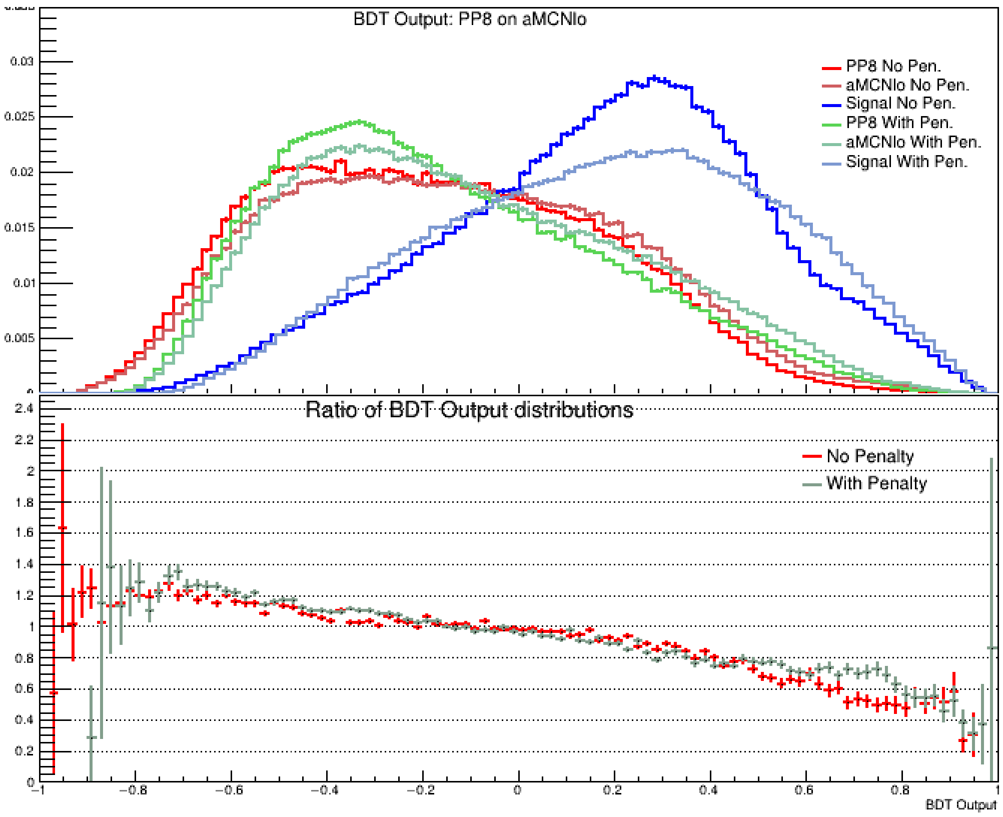
Training: PP8 no penalty, PP8 with penalty (compared to PoH7). Testing: PP8, PoH7



Line of Best Fit

Figure 14: Top: Results of BDT Output for **PP8/PoH7** with and without penalty weights, overlaid. Middle: ratios of background distributions with and without penalty weights, overlaid. Bottom: ratios with added line of best fit. The two generators appear to perform very similarly, indicating that the current implementation of the weighting algorithm is ineffective against initially similar generators.

Training: PP8 no penalty, PP8 with penalty (compared to aMC). Testing: PP8, aMC



Line of Best Fit

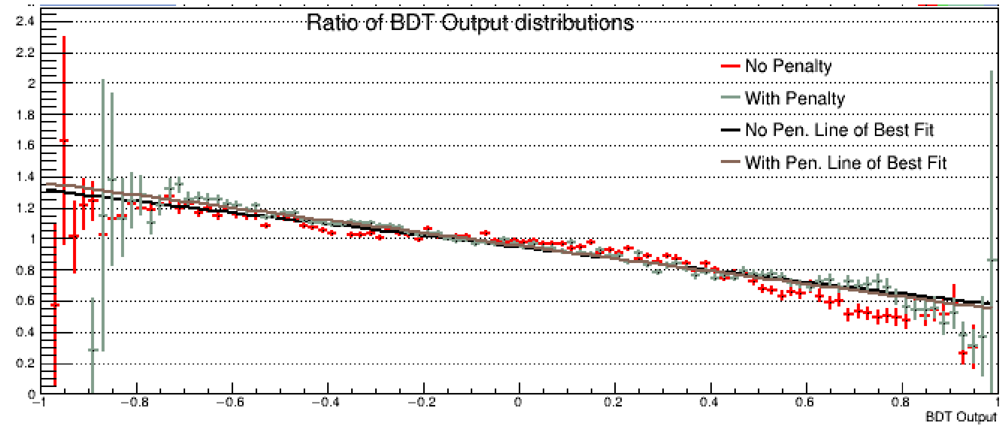
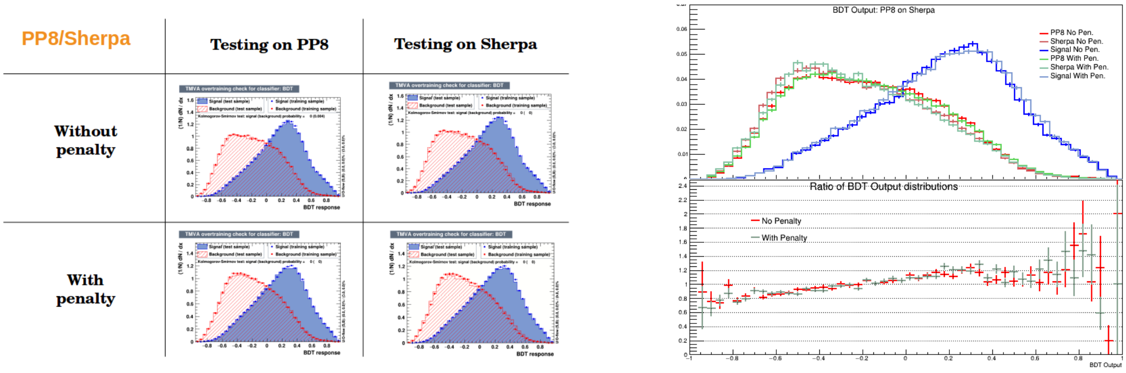


Figure 15: Top: Results of BDT Output for **PP8/aMC** with and without penalty weights, overlaid. Middle: ratios of background distributions with and without penalty weights, overlaid. Bottom: ratios with added line of best fit. The penalty weights appear to induce a significant decrease in systematic bias in the signal-sensitive region, potentially due to stark initial differences between the generators.

## 4.2. Modifying the Weighting Algorithm

The initial implementation of the penalty weights was performed only as a proof of principle for this method of reducing systematic bias in BDT training. As discussed above, the pilot implementation compounds all penalty weights across all variables together in equal measure as $w_{overall} = \prod_{i=1}^{N} w_i$, where N is the number of input variables shown in Appendix A and $w_i$ the weight for the $i^{\text{th}}$ variable.

As a second approach, rather than compounding all weights, the variable with the highest weighting was identified and only its weight was assigned to the event in training, such that $w_{overall} = w_{highest}$. This method was tested with Sherpa as the testing generator, and the results are shown in Figure 16.



(a) Results of Cross-Testing. We observe a very small shift in the BDT response when the $w_{overall} = w_{highest}$ algorithm is used, almost negligible.

(b) BDT outputs overlaid and the ratios with and without penalty shown. The penalty weights have a negligible effect on the BDT response.

Figure 16: New Penalty Weighting Algorithm. Training was performed with PP8, and testing with Sherpa. A coarser binning is utilized for clarity.

As the figure demonstrates, this particular algorithm appears to be too weak to generate any significant change to the systematic bias in the BDT training. This may serve as an indication that the optimal weighting method is somewhere in between compounding all variables and using only one – or that utilizing a different algorithm might be beneficial.

A suggestion for future research would be to further investigate ways to refine the weighting algorithm, perhaps giving larger priority to certain variables' weights if they are deemed to be particularly sensitive to the differences in the generators. In particular, one could define the overall weight as $w_{overall} = \prod_{i=1}^{N} c_i w_i$, where $c_i$ is some weighting related to the variables' separation power, or a parameter that quantifies the differences between how the generators model different variables. An optimization could be performed to these $c_i$'s to determine the most optimal weighting algorithm.

# 5. Summary, Conclusions and Outlook

The prohibitively small Signal/Background ratio in the ttH(bb) analysis led to the implementation of multivariate analysis tools such as classification BDTs. However, large uncertainties in the dominant background modelling reflected large uncertainties in the signal-strength parameter, thus motivating the study of model dependence in the BDT. A novel method for reducing the systematic bias in the BDT training was investigated, consisting of implementing penalty weights to events that fall in regions of discrepancy between different MC background event generators. The current implementation of the method was shown to be effective for test generators with large initial discrepancy (PP8 vs. aMC), but ineffective against others with smaller initial discrepancy (PP8 vs. PoH7, PP8 vs. Sherpa).

Repeating the study with an increased number of MC generator statistics would be beneficial in mitigating statistical ambiguities in these results (particularly for the PoH7 generator). Alternatively, the weighting algorithm could be modified as to ignore regions of low statistics when compounding the penalty weights.

Further study is also required to investigate the implementation of the weighting algorithm, with particular focus on optimizing the method for compounding the weights. Further analysis on the effect of the weighting on the overall BDT performance is also advised, in particular through performing the training with a different generator, as to check whether trends in the results match those observed in the current study.

# Acknowledgement

I would like to express my appreciation to my supervisors Dr. Paul Glaysher and Dr. Judith Katzy, for their constant attention, dedication and assistance throughout. Many thanks to the DESY ATLAS group, for ensuring a feeling of welcome and camaraderie throughout the development of this project. Special thanks go to the DESY Summer Student Program organizers, for arranging such an enjoyable and educational experience. Lastly, I'd like to express my gratitude to DESY and to the city of Hamburg, for welcoming the author. This study was undertaken as part of the DESY Summer Student Program 2018.

# References

[1] D. de Florian et al. (2017), *Handbook of LHC Higgs Cross Sections: 4. Deciphering the Nature of the Higgs Sector*, CERN Yellow Reports: Monographs, Vol. 2, (CERN–2017–002-M).
arXiv: 1610.07922 [hep-ph].

[2] ATLAS Collaboration (2018), *Search for the Standard Model Higgs boson produced in association with top quarks and decaying into a bb pair in pp collisions at* $\sqrt{s}$*=13 TeV with the ATLAS detector* , Phys. Rev., Vol. D97, No. 07, p. 07206.
arXiv: 1712.08895 [hep-ex].

[3] A. Hoecker et al.(2007), *TMVA - Toolkit for Multivariate Data Analysis*,
arXiv: physics/0703039 [physics.data-an].

[4] Wyner et al. (2017), *Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers*, Journal of Machine Learning Research, Vol. 18, 1-33.

[5] A. Bagoly et al. (2017), *Machine Learning Developments in ROOT*, J. Phys., Conf. Ser. 898, p. 072046.

[6] S. An. (2017), *Optimising Variable Selection for Machine Learning Analysis in ATLAS ttH Search*, Online.
Available at: http://www.desy.de/f/students/2017/reports/SitongAn.pdf. [Accessed: 30-08-18].

# A. Variable List for BDT Training

| Variable Name | Variable Description |
|---|---|
| nHiggs30_70 | Number of b-jet pairs with invariant mass within 30 GeV of the Higgs boson mass |
| nJets_Pt40 | Number of jets with $p_\mathrm{T} \geq 40$ GeV |
| pT_jet5 | $p_\mathrm{T}$ of fifth leading jet |
| HT_jets | Scalar sum of jet $p_\mathrm{T}$ |
| HT_all | Scalar sum of all $p_\mathrm{T}$ |
| H1_all | Second Fox-Wolfram moment computed using all jets and charged leptons |
| dEtajj_MaxdEta | Maximum $\Delta\eta$ between any two jets |
| Centrality_all | Scalar sum of the $p_\mathrm{T}$ divided by the sum of E for all jets and the lepton |
| dRbb_avg_70 | Average $\Delta R$ for all b-tagged jets |
| Mbb_MindR_70 | Invariant mass of the combination of any two b-jets with the smallest $\Delta R$ |
| Mbj_MaxPt_70 | Invariant mass of the combination of jet and b-jet with the largest vector sum $p_\mathrm{T}$ |
| dRbb_MaxPt_70 | $\Delta R$ between the two b-jets with the largest vector sum $p_\mathrm{T}$ |
| dRbb_HiggsMass_70 | $\Delta R$ between b-jets from the Higgs candidate |
| dRlepbb_MindR_70 | $\Delta R$ between the lepton and the combination of the two b-tagged jets with the smallest $\Delta R$ |
| Aplanarity_jets | $1.5\lambda_2$, where $\lambda_2$ is the second eigenvalue of the momentum tensor built with all jets |
| Mjj_MindR | Invariant mass of the combination of any two jets with the smallest $\Delta R$ |
| dRbj_Wmass_70 | $\Delta R$ between a b-jet and any other jet from the W boson candidate |
| Mbj_Wmass_70 | Invariant mass of a b-jet and any other jet from the W boson candidate |
| Mbj_MindR_70 | Mass of the combination of any jet and b-jet with the smallest $\Delta R$ |
| dRlj_MindR | $\Delta R$ between any jet and a b-jet with the smallest $\Delta R$ |
| pT_jet3 | $p_\mathrm{T}$ of the jet with third largest $p_\mathrm{T}$ |
| dRbb_MaxM_70 | $\Delta R$ between the two b-jets with the largest invariant mass |
| H4_all | Fifth Fox-Wolfram moment computed using all jets and charged leptons |
| Aplanarity_bjets_70 | As Aplanarity_jets, for b-tagged jets |
| Mjjj_MaxPt | Invariant mass of three jets with largest vector sum $p_\mathrm{T}$ |
| Mbb_MaxM_70 | Largest invariant mass of the combination of any two b-jets |
| Mjj_MinM | Smallest invariant mass of the combination of any two jets |
| dEtabb_Avg_70 | Average $\Delta\eta$ for all b-jets |

Table 2: List of variables used in the BDT. All b-tagged variables correspond to a b-tagging at 70% efficiency.