Investigating the improvement of Instanton process classification via machine-learning methods

Nicholas Mitchell

Merton College, University of Oxford
nicholas.mitchell@merton.ox.ac.uk

October 1, 2021

Abstract

Yang-Mills theories within the Standard Model predict currently-undiscovered topological objects, for example the Instanton. Recently proposals show that it would be fruitful to search for Instantons in proton-proton collisions at the LHC. Instantons are expected to behave as "soft-bombs" of very many low-energy jets, and might lie hidden within data already recorded. This work demonstrates the improved sensitivity of Machine Learning methods trained on Monte Carlo simulation over threshold cuts to classify proton-proton collision events and better distinguish the Instantons from perturbative QCD backgrounds.

1 Introduction

The Standard Model predicts a pseudo-particle called an Instanton [1]. These pseudo-particles are non-perturbative topological objects in quantum chromodynamics (QCD) that describe transitions between classically-degenerate vacua in Minkowski spacetime. Instantons are expected to decay into multiple gluons and all kinematically-available quark-antiquark pairs, therefore it is expected that events with Instantons present have more charged particle tracks and a more spherical momentum distribution.

Events involving Instantons may already be present in current recorded LHC data but remain clouded by background events. Instanton events may be filtered for by selecting a signal region specified by thresholds for particular event parameters to improve the ratio of Instanton events to background events.

This project is aimed at improving the classification of Instanton processes by opting for machine-learning methods instead of simple threshold cuts. A better classification method would help improve the significance of Instanton detection in future studies. Section 2 outlines the methods by which this comparison will take place, Section 3 provides the results of this project.

2 Methods

Particle collision events are measured by measuring charged-particle tracks originating from the collision process. The information about these tracks can be used to calculate various event-level features. Instanton events are expected to be characterised by certain behaviours in these quantities. For example, Instanton events are expected to have more tracks and more spherical track distributions. By searching for events which exhibit these features, Instanton signal events may be found.

2.1 Monte Carlo Samples

Simulated samples for signal and background events are taken from Monte Carlo (MC) simulations. The background sample is generated by the EPOS [2] generator and the signal sample [3] is generated by Sherpa [4] with a cut for Instanton masses greater than 50GeV.

Data used in Figure 7 is a sample taken from LHC ATLAS collisions at $\sqrt{s}$=13 TeV with one collision per bunch crossing for clean analysis.

To remove biases due to the $m_{instanton} \geq$ 50GeV slice, all samples are sliced for the sum of scalar transverse momentum (ST)$\geq$ 50GeV.

2.2 Features

The following features (i.e. event-level variables that may help distinguish between Instanton events and background events) were used in this analysis. Figure 1 shows normalised histograms for each of these features, demonstrating the separation between signal and background events.

2.2.1 Number of tracks (or track multiplicity, or multiplicity)

The number of tracks is simply the number of recorded tracks of an event.

2.2.2 ST

The ST is the sum of scalar transverse momenta of an event: $ST \coloneqq \sum_{i \in Tracks} |\underline{p}_i^T|$, where $\underline{p}_i^T$ is the momentum in the transverse plane. All masses and momenta are measured in MeV in natural units unless stated otherwise.

2.2.3 Invariant mass

The invariant mass of an event is the total invariant mass of all track 4-momenta summed.

2.2.4 Mass per track

This is the invariant mass divided by the number of tracks of an event. Although it is a degenerate feature, i.e. it can be exactly determined by other features, the mass per track is still used as it is a physically-relevant quantity and ML methods cannot perform operations such as division between features, therefore non-linear functions or combinations of other features may provide additional distinguishing power.

2.2.5 Transverse Mass

The transverse mass for an event is calculated as

$$m_{transverse}^2 \coloneqq \left(\sum_{i \in Tracks} |\underline{p}_i^T| \right)^2 - \left| \sum_{i \in Tracks} \underline{p}_i^T \right|^2 . \tag{1}$$

This is the invariant mass of the event if all track 4-momenta are projected into the transverse plane as 1+2 momenta, then assumed to be null.

2.2.6 Transverse mass per track

This is the transverse mass divided by the number of tracks of an event.

2.2.7 Magnitude of mean pseudorapidity

The magnitude of mean pseudorapidity is $|\langle \eta \rangle|$, where η is the pseudorapidity of a track and the mean $\langle \bullet \rangle$ is taken over all tracks of an event. The absolute value is taken since $\langle \eta \rangle$ has a symmetric distribution about zero as it is related directly to the momentum ratio between the two colliding protons which collide head-on with similar energies.

2.2.8 Standard deviation of pseudorapidity

σ_η is the standard deviation of track pseudorapidities of an event.

2.2.9 Sphericity scalars (S,A,C,D)

The sphericity tensor is defined as

$$S^{\alpha\beta} \coloneqq \frac{\sum_{i \in Tracks} |\underline{p}_i|^{r-2} p_i^\alpha p_i^\beta}{\sum_{i \in Tracks} |\underline{p}_i|^r}, \tag{2}$$

where r determines the momentum weighting of tracks. $r = 1$ is chosen for this analysis as it provides a good separation of signal versus background, as well as suppressing a secondary peak in event shapes at zero generated by low numbers of high-momentum tracks. The eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ of $S^{\alpha\beta}$ are then used to calculate four scalar event shapes S,A,C,D as:

$$S \coloneqq \frac{3}{2}(\lambda_2 + \lambda_3), \tag{3}$$

$$A \coloneqq \frac{3}{2}\lambda_3, \tag{4}$$

$$C \coloneqq 3(\lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_1\lambda_3), \tag{5}$$

$$D \coloneqq 27\lambda_1\lambda_2\lambda_3, \tag{6}$$

2.2.10 Thrust

The thrust is defined as

$$T \coloneqq \max_{\underline{t}} \frac{\sum_{i \in Tracks} |\underline{t} \cdot \underline{p}_i|}{\sum_{i \in Tracks} |\underline{p}_i|}, \tag{7}$$

where $\underline{t}$ is a unit vector called the thrust axis also defined here as the unit vector that maximises this fraction. This is calculated here in 2D to avoid longitudinal momentum weighing the thrust axis along the beam axis, so the thrust axis is uniquely specified by a single azimuthal angle and this may be simplified to

$$T \coloneqq \max_{\varphi} \frac{\sum_{i \in Tracks} |\underline{p}_i^T \cos(\varphi - \theta_i)|}{\sum_{i \in Tracks} |\underline{p}_i^T|}, \tag{8}$$

where θ_i is the azimuthal angle of track i and φ is the azimuthal angle of the thrust axis.

2.2.11 Broadening

The broadening in 3D is defined by

$$B \coloneqq \frac{\sum_{i \in Tracks} |\underline{t} \times \underline{p}_i|}{\sum_{i \in Tracks} |\underline{p}_i|}. \tag{9}$$

In the 2D transverse plane, this may be simplified to

$$B \coloneqq \frac{\sum_{i \in Tracks} |\underline{p}_i^T \sin(\varphi - \theta_i)|}{\sum_{i \in Tracks} |\underline{p}_i^T|}, \tag{10}$$

where φ is the azimuthal angle of the thrust axis as defined by the thrust in Equation 8.

The last six features (S, A, C, D, T and B) are called 'event shapes' as they describe the distribution of track momenta after the collision.

2.2.12 Other features

Other features were considered but will not be included in the final study for various reasons. These include:

- p^T-weighted pseudorapidity

  The p^T-weighted pseudorapidity for a track i is defined as $\tilde{\eta}_i \coloneqq |\underline{p}_i^T|\eta_i / \sum_{j \in Tracks} |\underline{p}_j^T|$, then the absolute value of the mean $|\langle\tilde{\eta}\rangle|$ and standard deviation $\sigma_{\tilde{\eta}}$ may be calculated and used as new features.

  The p^T-weighted pseudorapidity variables will not be used further as they are still similar in distribution to the unweighted pseudorapidity variables therefore should not provide much new distinguishing power. In addition, the p^T weighting causes $\tilde{\eta}$-related features to become correlated with other features, both kinematic features and event shapes (especially with Thrust and Broadening). The main power of the pseudorapidity features is their lack of correlation with all other features, which is lost under the p^T-weighting.

- Transverse sphericity

  The sphericity tensor defined by Equation 2 may be calculated in the 2D transverse plane by replacing $\underline{p}_i$ with $\underline{p}_i^T$. This 2D sphericity tensor then has only two eigenvalues, so only two nonzero definitions for scalar sphericity event shapes remain: $S_{2D} \coloneqq 2\lambda_2$ and $C_{2D} \coloneqq 4\lambda_1\lambda_2$. These two features are actually degenerate since $\sum_i \lambda_i = 1$ for eigenvalues of $S^{\alpha\beta}$, therefore $C_{2D} = S_{2D}(2 - S_{2D})$, so only one of these features is useful. The distribution does have some separation and is not perfectly correlated with other features, however the 3D sphericity event shapes are preferable as they should capture more information about the momentum distribution of the event.

- Number of displaced tracks

  The number of displaced tracks is the number of tracks with a longitudinal impact parameter d_0 greater than some minimum threshold d_0^{min}. This is motivated physically as an approximation to the number of secondary vertices if $d_0^{min} \sim 0.02mm$ [5], around 3 times the resolution of the d_0 measurement to identify tracks that have not originated from the primary vertex. This feature was not used as it appeared highly correlated with the number of tracks for $d_0^{min} = 0.02mm$. Although much larger values of d_0^{min} may be chosen, these appear arbitrary therefore the number of displaced tracks is not included.

2.2.13 Feature Histograms

Normalised histograms for each of these features for a background sample and signal sample are shown in Figure 1 to demonstrate the difference in distributions between signal and background samples, therefore providing some motivation that the selected features are sensible inputs to a classifier.

The feature histograms with luminosity weighting at an integrated luminosity of 1 pb^{-1} are show in Figure 2 to demonstrate the magnitude differences between the signal and background.

Figure 3 shows correlation heatmaps between pairs of features for each sample. The four main blocks along the diagonal show that there are four groups of features which are highly correlated with the other features in that group. Although this means the classifiers might still work well with just four features (one from each block/group), all of the features are still used as imperfect correlations mean that there is still some additional information gained by these additional features.

2.3 Cuts

A cut is a selection of bounds on event observables in which the desired Instanton processes are more prominent over background processes compared to the uncut sample.

The best cuts applied should have a maximal true positive rate (TPR) for a given false positive rate (FPR). This is achieved by providing a set of sensible slices [5] of varying strictness on each feature (for example, a minimum number of tracks bound as we expect Instanton signal events to have more tracks) then generating a set of cuts as every combination of these feature slices, as detailed in Appendix 6.2. These cuts are then plotted on a receiver-operating-characteristic (ROC) plot and fitted with an upper envelope which is effectively the ROC curve for possible cuts. The area under the ROC curve (AUC) then lends itself as a sensible and threshold-invariant (as different points along the ROC curve correspond to different thresholds) scalar measure between 0 and 1 of how well the classifier is performing.

These cuts could be further improved as an AUC-optimisation problem, but the selection of slices provided should give a sensible overview of cut performance.

2.4 Machine-Learning Methods

The Python library ‘scikit-learn’ [6] is employed as the machine-learning (ML) framework for this project [7] as it allows many ML methods to be applied with ease. Various ML methods are applied, but the two with best performance for this project are neural networks and boosted decision trees. Two of each of these methods are employed, each with different algorithms and hyperparameters tuned roughly to give optimal performance. Details of the specific ML methods and hyperparameters used can be found in Appendix 6.1.

One of the main disadvantages of ML methods over cuts are that ML methods are much harder to interpret as they act as a black boxes, i.e. are complicated so an understanding of how they are distinguishing between signal and background events is difficult. One way to attempt to understand how the ML method is classifying events is feature importance , a measure of how strongly different features affect the classification output. A feature’s importance [8] is the change in output score (e.g. classification accuracy) when values for that feature are permuted through different events (i.e. the column of the feature in question for all events is randomly shuffled). More important features will have a larger effect on classifier output when their values are permuted, therefore a greater feature importance as defined above. The feature importance I^f is defined [9] as

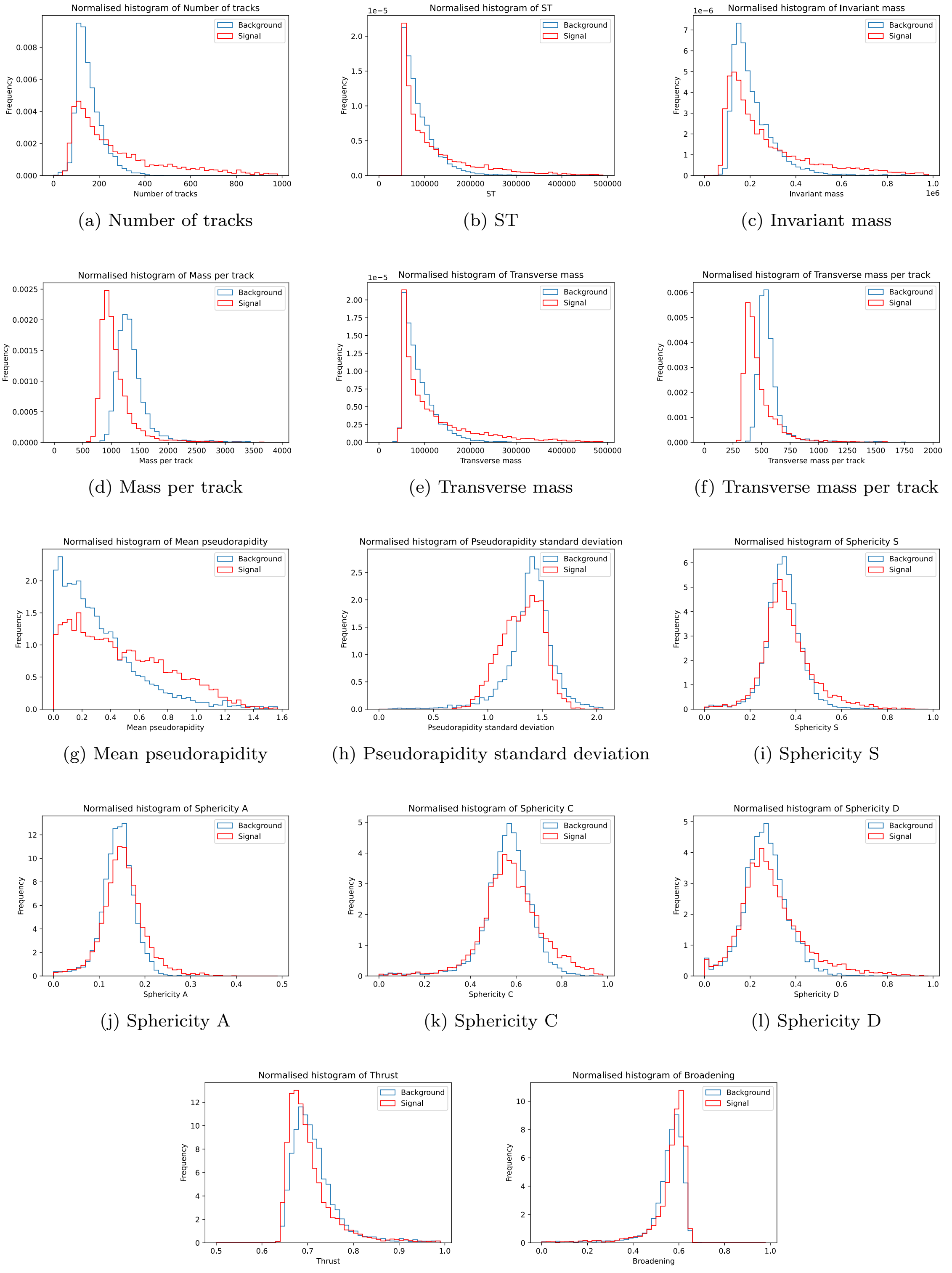
$$I^f := s - \frac{1}{N}\sum_{n=1}^{N} s_n^f, \tag{11}$$

where s is the original classifier score, s_n^f is the classifier score achieved by a random shuffle, labelled n, of the values of feature f, and N is the number of random shuffles which should be some reasonably large integer to average shuffles over (set to 100 in this study).

2.5 Background Estimation via ABCDisCo

The background within the signal-dominated region as determined by the ML classifier or cuts may be estimated using the ABCD method [10]. This requires two variables that are independent in the background sample and define the signal region well by two simple cuts. Choosing the output of an ML classifier as one of these variables may lead to a better background approximation as the signal and background regions should be more appropriately dominated than those regions as selected by cuts.

When using an ML classifier as one or both of these variables, decorrelation in the background with the second feature could be achieved by training the network without that feature (or any other correlated features). Figure 3 shows that both of the pseudorapidity-related features are almost completely independent of all other features. This suggests that the mean pseudorapidity may be a good choice for the second feature as it is



(a) Number of tracks

(b) ST

(c) Invariant mass

(d) Mass per track

(e) Transverse mass

(f) Transverse mass per track

(g) Mean pseudorapidity

(h) Pseudorapidity standard deviation

(i) Sphericity S

(j) Sphericity A

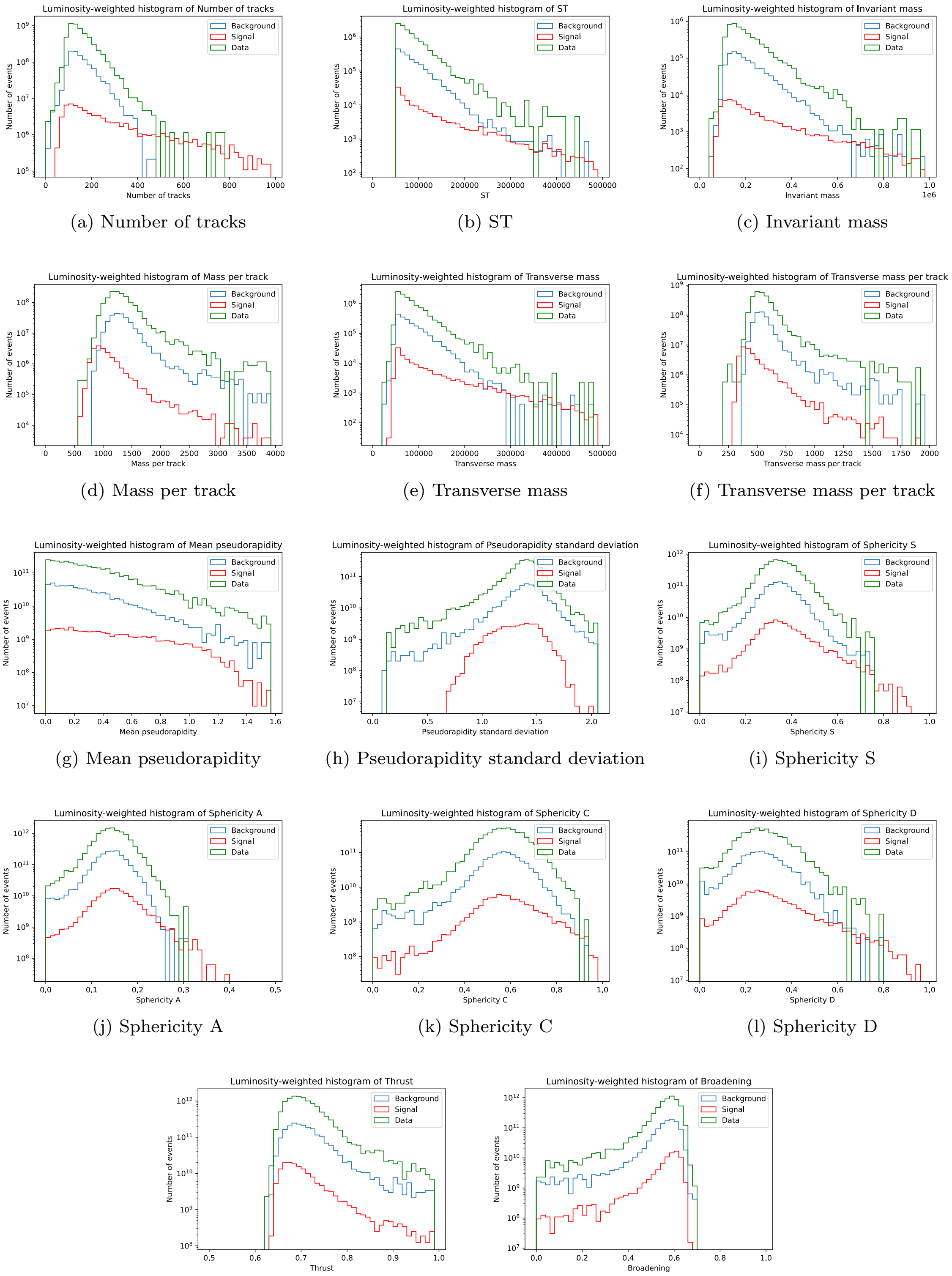
(k) Sphericity C

(l) Sphericity D

(m) Thrust

(n) Broadening

Figure 1: Normalised histograms for signal and background samples each feature



(a) Number of tracks

(b) ST

(c) Invariant mass

(d) Mass per track

(e) Transverse mass

(f) Transverse mass per track

(g) Mean pseudorapidity

(h) Pseudorapidity standard deviation

(i) Sphericity S

(j) Sphericity A

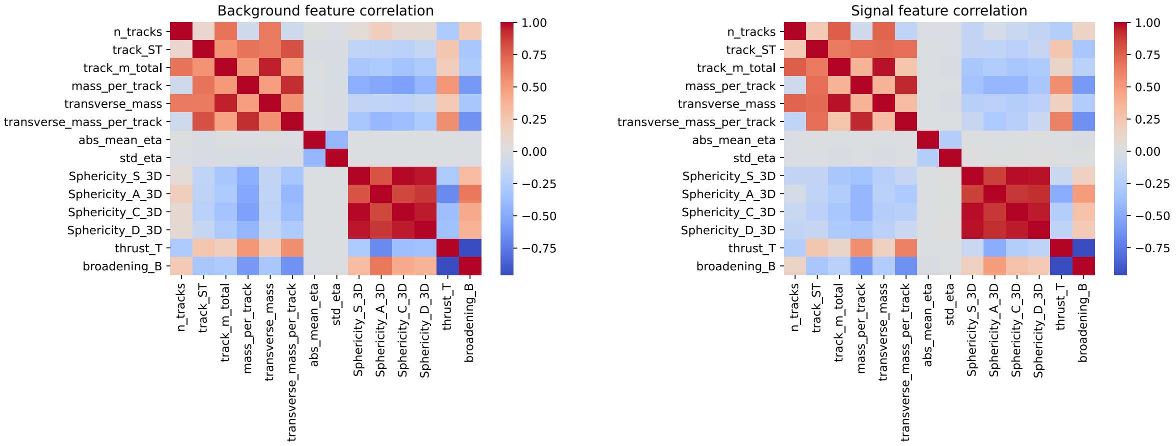
(k) Sphericity C

(l) Sphericity D

(m) Thrust

(n) Broadening

Figure 2: Luminosity-weighted histograms of event-level features for background and signal samples with 2.12×10^{10} background events and 1.55×10^{9} signal events, calculated by the product of an integrated luminosity of $1\ pb^{-1}$, the predicted cross section, and the ratio of sample remaining after the $ST \geq 50\text{GeV}$ slice is applied. The data histogram has been scaled to the sum of expected events from the background and signal samples. These histograms also show that the MC background simulation matches up well with data in all of these observables.



(a) Background

(b) Signal

Figure 3: Correlation heatmaps between pairs of features for each sample. The product moment correlation coefficient (PMCC) is used to measure correlation

uncorrelated in the background with all other features and separates the signal and background reasonably well in Figure 1.g.

If this first method does not work, then decorrelation can be enforced by using the DisCo method as discussed for the ABCD method in [10]. Unfortunately scikit-learn is not apt for customising the loss function or backpropagation routine as required by the DisCo method, so another ML framework such as Keras [11] may have to be used instead.

The DisCo method is a technique for training a ML classifier while forcing the output to be decorrelated with another feature by modifying the classifier's loss function to

$$L_{DisCo} \coloneqq L_{classifier}(f, y_{target}) + \alpha \; dCorr^2(f, g)|_{background}, \tag{12}$$

where L_{DisCo} is the classifier's new loss function, f is the classifier output, y_{target} is the Boolean target, g is the feature that the DisCo method aims to decorrelate against, $L_{classifier}(f, y_{target})$ is the classifier's original loss function, $dCorr^2(f, g)|_{background}$ is the *distance correlation* (a measure of correlation between 0 and 1 corresponding to no correlation and perfect correlation respectively) between f and g evaluated over the background samples, and α is a parameter describing the importance of decorrelation over classification. This is useful for the ABCD method where we desire two decorrelated features with strong separation in signal and background, which a good classifier should have.

A disadvantage of single DisCo, where g is an original feature, is that g may not be well separated in the signal and background. A better approach may be to use double DisCo, where g is another classifier output such that it is also well separated, hopefully giving a good separation in both f and g to reduce signal contamination in the control regions.

3 Results

3.1 Comparison of Machine-Learning Methods and Cuts

Figure 4 shows the ROC curve comparison between five ML methods and simple cuts. The best AUC of ML methods is 0.96, a significant improvement to the AUC of the cuts' upper envelope of 0.81. This demonstrates that the ML methods are outperforming simple cuts and would improve statistical significance for Instanton detection in future studies. A description of applied cuts and the best cuts (those that lie on the upper envelope) can be found in Appendix 6.2.

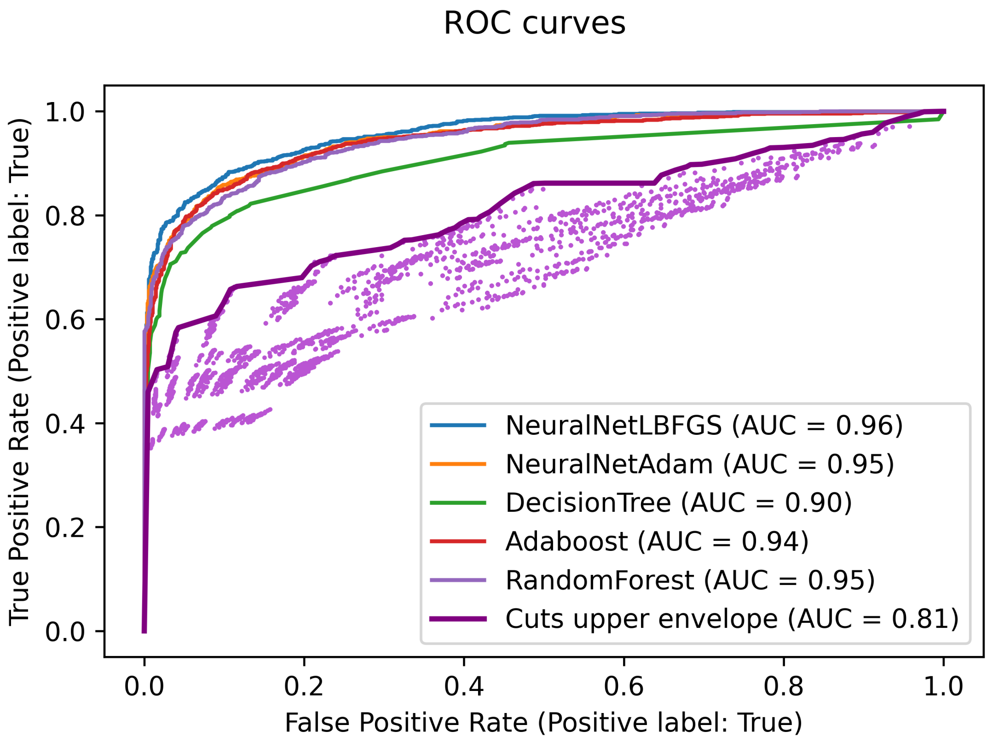


Figure 4: ROC curves comparing five ML methods and the upper envelope of cuts. Better classifiers have ROC curves closer to the top left of the plot, therefore a greater area under the ROC curve (AUC).

Figure 5 demonstrates the signal-to-background ratio in the signal region for various TPRs for each classifier. Similarly to the ROC curve, this plot also demonstrates the ML classifiers outperforming the cuts. The curves zigzag and end abruptly at low TPRs due to FPR→ 0 where the change in 1/FPR is discrete and becomes larger.

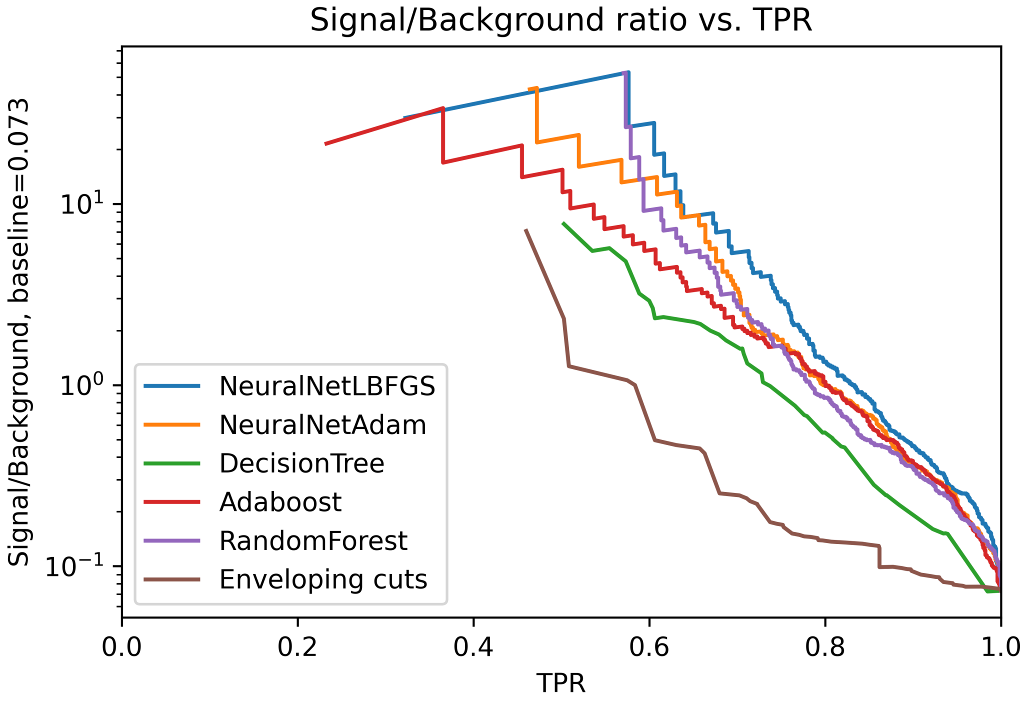


Figure 5: Signal-to-background ratio versus TPR for each classifier.

Figure 6 shows the feature importances for the five ML methods used. It is not surprising that the number of tracks and mass per track are the dominant classifying features for most ML classifiers as they appear to discriminate well between signal and background in Figure 1. The pseudorapidity-related features have small importances as although they are slightly distinct, the large overlap in histograms suggests that they do not distinguish well.

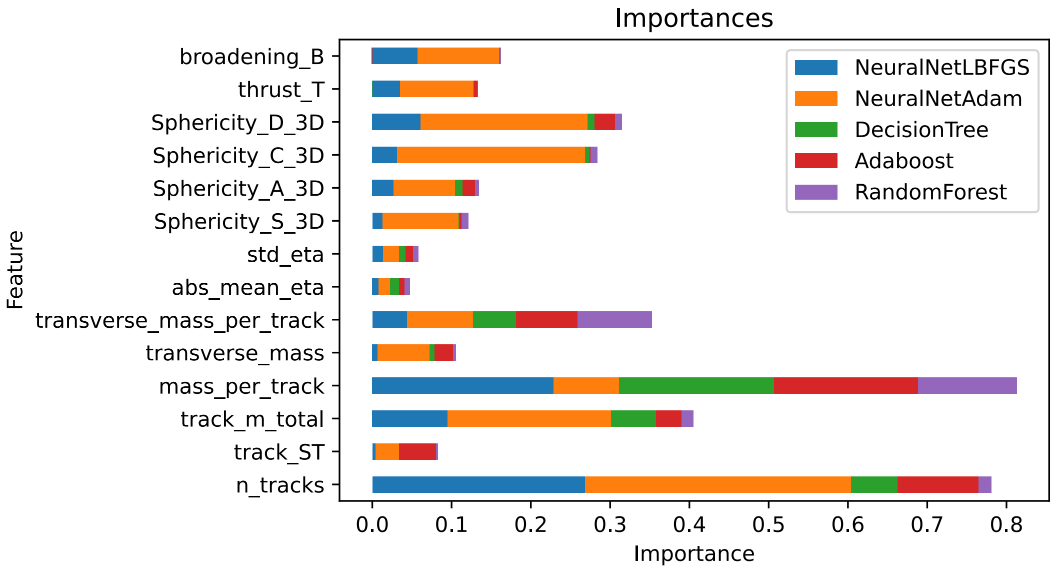
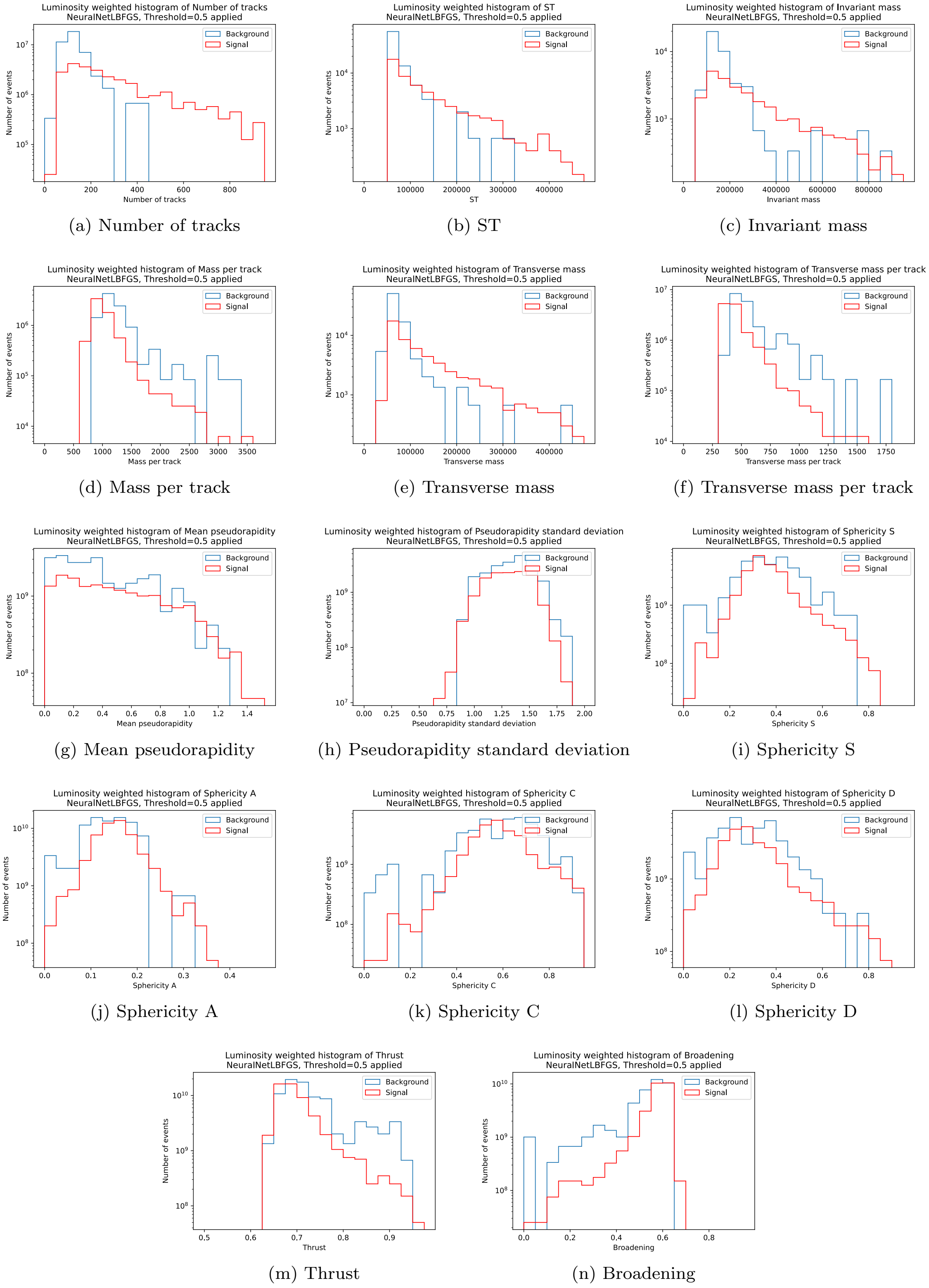


Figure 6: Feature importances of five ML methods used.

Figure 7 shows example histograms of signal and background samples after an ML classifier has been applied as a demonstration of how ML methods act as good filters for Instanton events.

3.2 Background Estimation via ABCDisCo

Figure 8 shows a classifier's predictions for different strengths of decorrelation with mean pseudorapidity in the background with the top plots demonstrating the capability of the classifier for an ABCD background estimation with single DisCo. Subplot (a) demonstrates the classifier with no decorrelation, (b) demonstrates the classifier's performance with a roughly optimal α, (c) shows a classifier beginning to perform badly due to a large α, and (d) shows a classifier with such a high α that the classifier desires only decorrelation with very little care for classification.


Luminosity weighted histogram of Number of tracks NeuralNetLBFGS, Threshold=0.5 applied

(a) Number of tracks

Luminosity weighted histogram of ST NeuralNetLBFGS, Threshold=0.5 applied

(b) ST

Luminosity weighted histogram of Invariant mass NeuralNetLBFGS, Threshold=0.5 applied

(c) Invariant mass

Luminosity weighted histogram of Mass per track NeuralNetLBFGS, Threshold=0.5 applied

(d) Mass per track

Luminosity weighted histogram of Transverse mass NeuralNetLBFGS, Threshold=0.5 applied

(e) Transverse mass

Luminosity weighted histogram of Transverse mass per track NeuralNetLBFGS, Threshold=0.5 applied

(f) Transverse mass per track

Luminosity weighted histogram of Mean pseudorapidity NeuralNetLBFGS, Threshold=0.5 applied

(g) Mean pseudorapidity

Luminosity weighted histogram of Pseudorapidity standard deviation NeuralNetLBFGS, Threshold=0.5 applied

(h) Pseudorapidity standard deviation

Luminosity weighted histogram of Sphericity S NeuralNetLBFGS, Threshold=0.5 applied

(i) Sphericity S

Luminosity weighted histogram of Sphericity A NeuralNetLBFGS, Threshold=0.5 applied

(j) Sphericity A

Luminosity weighted histogram of Sphericity C NeuralNetLBFGS, Threshold=0.5 applied

(k) Sphericity C

Luminosity weighted histogram of Sphericity D NeuralNetLBFGS, Threshold=0.5 applied

(l) Sphericity D

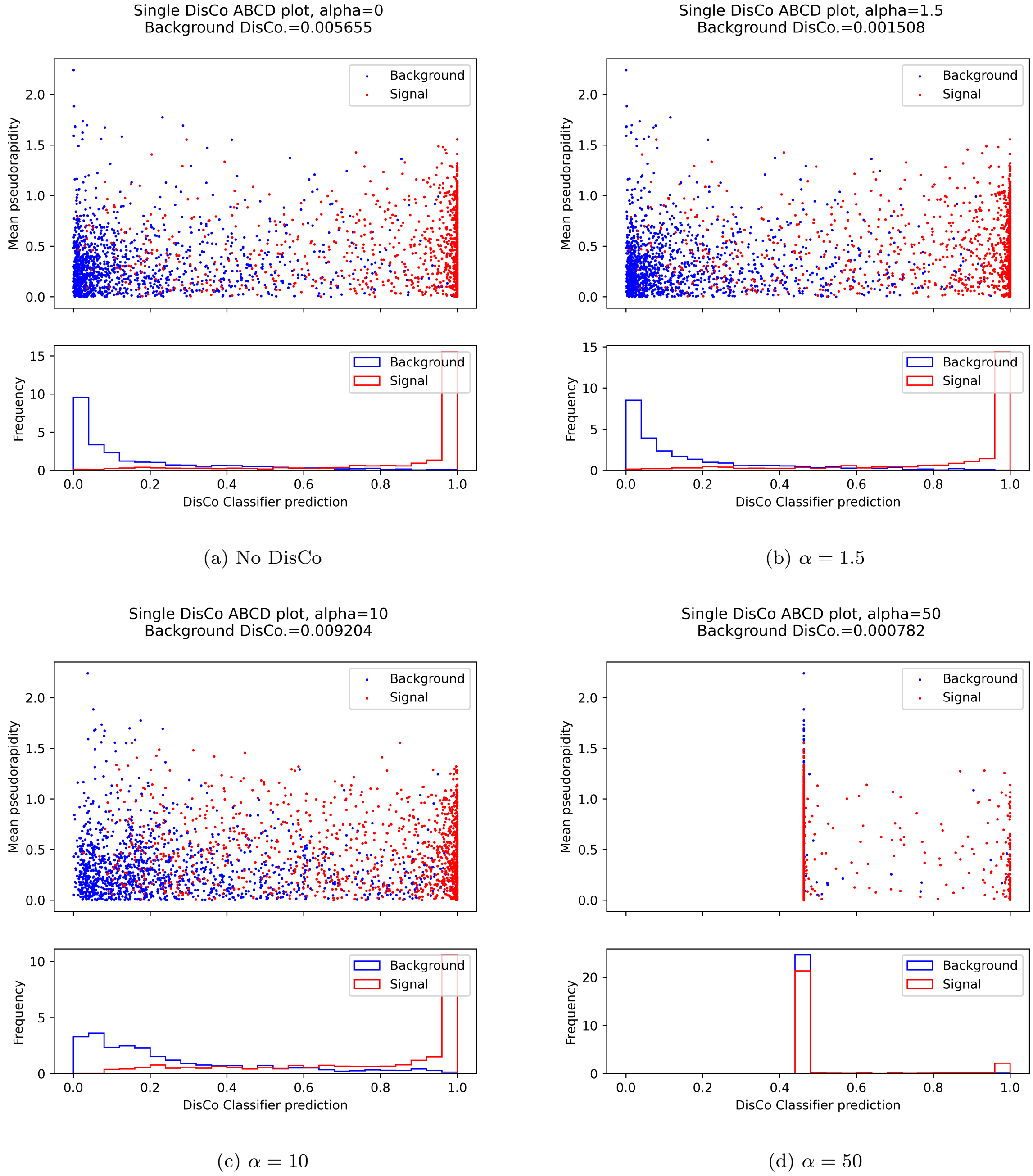
Luminosity weighted histogram of Thrust NeuralNetLBFGS, Threshold=0.5 applied

(m) Thrust

Luminosity weighted histogram of Broadening NeuralNetLBFGS, Threshold=0.5 applied

(n) Broadening

Figure 7: Luminosity-weighted histograms of event-level features for background and signal samples after an example ML classifier applied with 2.11×10^9 background events and 1.33×10^9 signal events, giving a signal-to-background ratio of 0.63. The ML classifier is operating at a threshold of 0.5, giving a TPR=0.86 and FPR=0.10. The bin size has been scaled up by a factor of 5/2 due to small sample sizes.



(a) No DisCo

(b) $\alpha = 1.5$

(c) $\alpha = 10$

(d) $\alpha = 50$

Figure 8: Plots of classifiers with different importances of decorrelation (values of α) for single DisCo.

This behaviour may be seen in Figure 9, where there is an optimal value (at the first minima) of α for which the classifier still distinguishes well between signal and background, then become unstable for larger values of α where the classifier starts losing all separation between signal and background, only attempting to minimise correlation. Although the initial decorrelation is very small, this figure shows that small DisCo corrections can improve decorrelation, improving the reliability of the background estimate via the ABCD method.

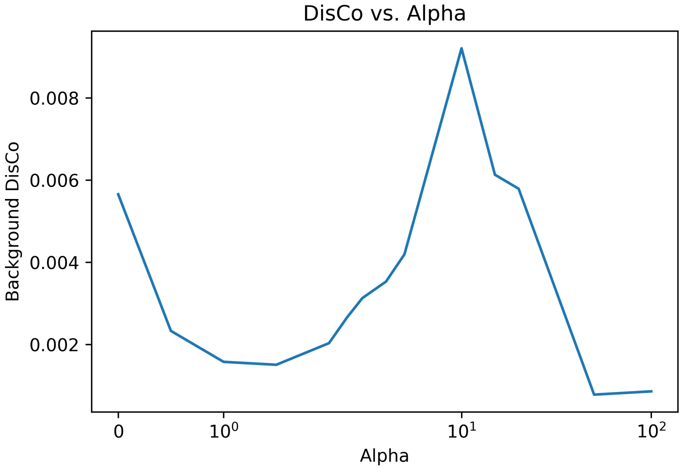


Figure 9: Plot of Distance Correlation between classifier output and mean pseudorapidity versus α for various classifiers for single DisCo.

Two classifiers A and B are used, trained sequentially. Classifier A is trained first on pseudorapidity and event-shape features such that the output has some separation but leaves the strongest features for classifier B. Classifier B is then trained on all of the features but forced to be decorrelated against the output of classifier A in the background.

The double DisCo approach is shown in Figure 10. Subplot (a) shows classifier A's output, then (b) shows outputs of classifier B versus classifier A as an ABCD plot with no DisCo. The noticable correlation of $dCorr^2 = 0.19$ with no DisCo motivates the use of the double DisCo method. (b) shows the ABCD plot with an optimal α used for decorrelation, then (c) shows the dangers of high α ruining classifier B's output.

Figure 11.a shows the correlation versus α for double DisCo, showing an optimal value of α similar to single DisCo. Using this optimal value of α, the ABCD regions may be defined with good separation and little background correlation as demonstrated in Figure 11.b. This provides a strong signal region defined by thresholds on these two classifiers, with decorrelation in the background enforced by double DisCo enabling the ABCD method to provide a good estimate of background contamination in the signal region.

4 Conclusion

Machine-learning methods are demonstrated to be an improvement over threshold cuts in classifying Instanton events in ATLAS pp collisions with a best AUC of 0.96 versus 0.81 respectively, therefore may improve statistical significance of Instanton detection in future studies. The number of tracks and mass per track appear to be key features in classifying Instanton events with other features like event shapes having lower distinguishing power between signal and background events.

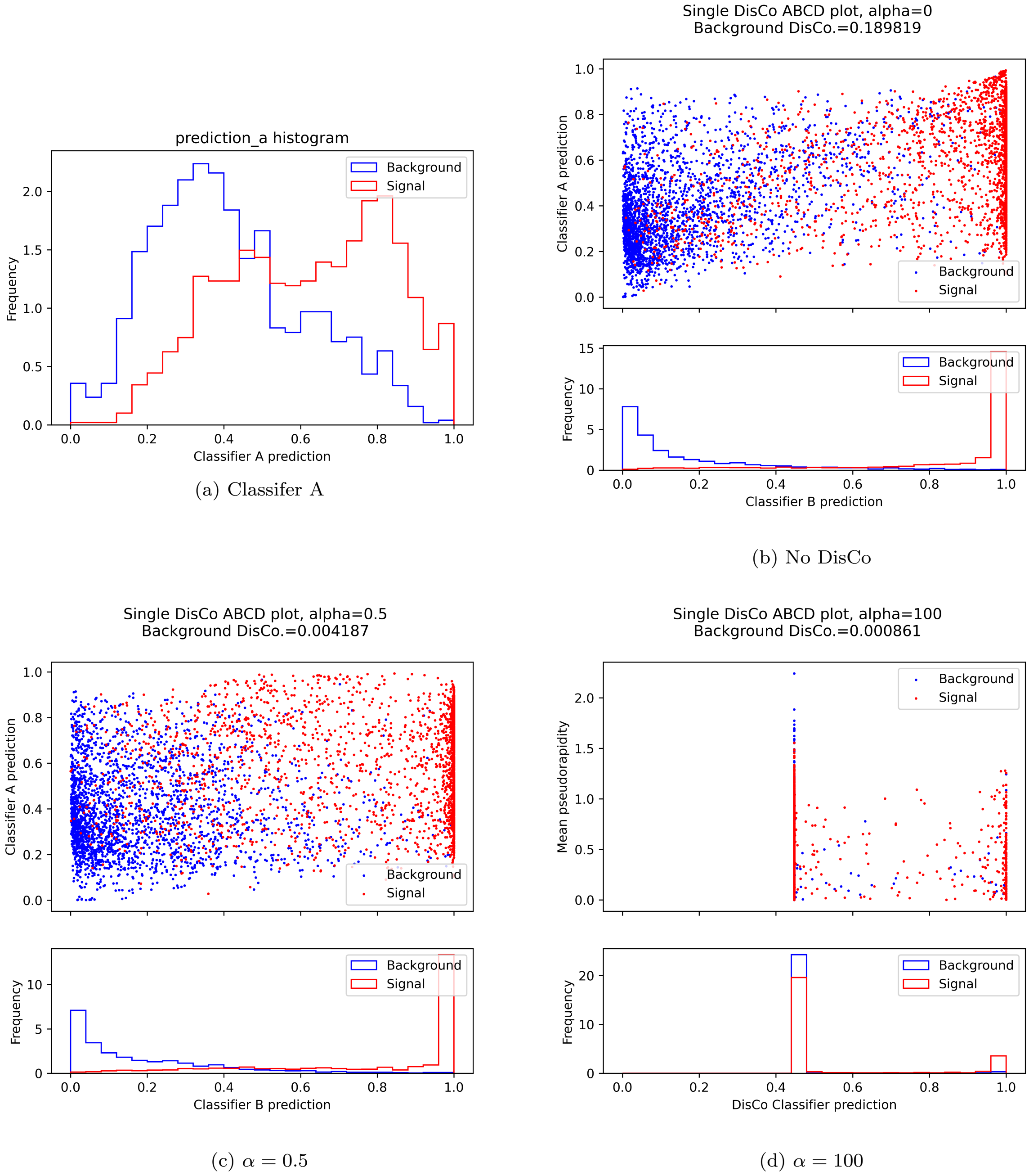
The classifier output may be used as one or both features for the ABCD method to estimate background contamination in the signal region. The single DisCo method appears to work well to decorrelate the classifier output against the other feature used. The double DisCo method appears to be particularly useful for the ABCD method as it provides a good separation in two classifier outputs that are forced to be independent in the background.

This study was limited to a Monte Carlo event generator of Instanton masses greater than 50 GeV. Future work may be able to apply and combine ML methods on various signal samples of different Instanton mass regimes in order to better classify Instanton events.

This work demonstrates the capability of ML methods to improve upon cuts but future work may want to attempt to choose a set of features more carefully. For example, more independent features could be included that have not been considered here, or features that do not provide sufficient improvement to the classifier to justify their use could be removed. A small set of specific working points (i.e. set of features, classifier, and thresholds) may want to be chosen.

5 Acknowledgements

I would like to thank Alan Barr for providing, organising and guiding me through this project. I would also like to thank Ynyr Harris for his regular advice, guidance and samples. This project was funded by the Merton College Summer Project scheme to which I am very grateful.



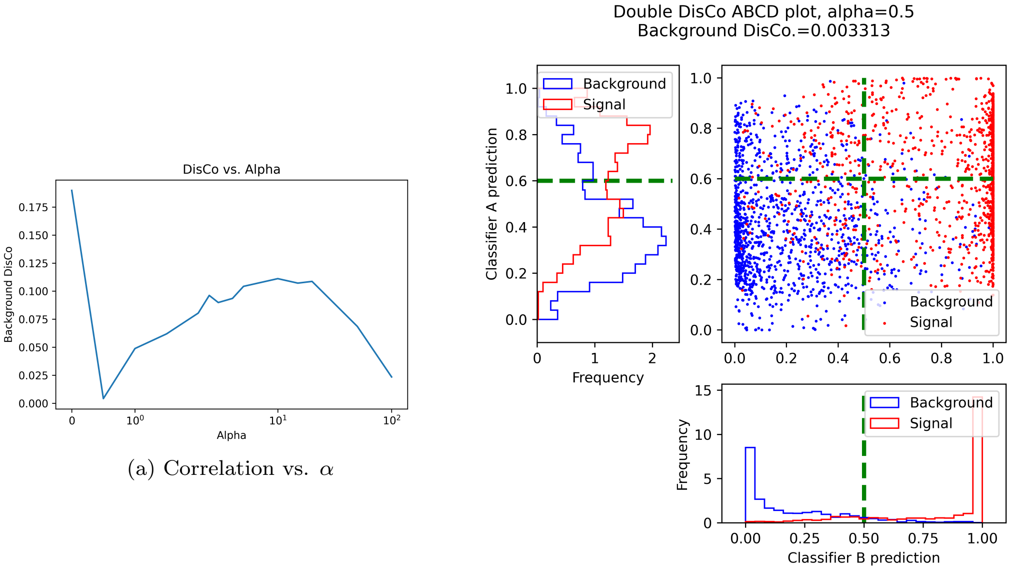
(a) Classifer A

(b) No DisCo

(c) $\alpha = 0.5$

(d) $\alpha = 100$

Figure 10: Plots of classifiers with different importances of decorrelation (values of α) for double DisCo. Classifier A (y-axis) is trained first on pseudorapidity and event-shape features, then classifier B (x-axis) is trained on all features while being forced to decorrelate with classifier A.



(a) Correlation vs. α

(b) ABCD with double DisCo

Figure 11: (a) Plot of Distance Correlation between classifier A output and classifier B versus α for various classifiers for double DisCo, showing an optimal value of $\alpha \approx 0.5$. (b) Example ABCD (signal region A in top right) plot with two classifier outputs with ABCD regions selected with green lines.

6 Appendices

6.1 Machine-Learning Methods used

Classifier name	Scikit-Learn name	Hyperparameters
NeuralNetAdam	MPLClassifier	solver='adam'
NeuralNetLBFGS	MPLClassifier	solver='lbfgs'
DecisionTree	DecisionTreeClassifier	max_depth=7

Random Forest	RandomForestClassifier	max_depth=7,
		n_estimators=400,
		max_features=3

Adaboost	AdaBoostClassifier	n_estimators=400

Table 1: ML methods used.

6.2 Cuts applied

Possible slices are defined on each feature in Table 2. Minimum slice bounds mean that all events below that value are removed, whereas maximum slice bounds mean that all events above that value are removed.

Each possible combination of these slices is then taken to form the set of cuts (10,080 cuts for these set of slices). These cuts are applied, plotted on the ROC curve in Figure 4 and fitted with an upper envelope as the cuts best ROC curve. The cuts forming the envelope are given in Table 3.

Letter	Feature	Bound	Slice bound	No. slices
a	Number of tracks	Min.	None,15,25,30,45,180	6
b	$\lvert\langle\eta\rangle\rvert$	Min.	None, 0.05, 0.075, 0.1, 0.2	5
c	Mass per track	Max.	None,2000,1700,1600,1500,1400,1300	7
d	Transverse mass per track	Max.	None,700,600,550,525,500,475,450	7
e	σ_η	Max.	None,1.8,1.75,1.7,1.65,1.6	6

Table 2: Possible slices used. Values were chosen to maximise AUC and cover the FPR interval evenly. Other slices on invariant mass and sphericity were also considered, but were removed since they did not sit on the upper envelope.

Index	a	b	c	d	e	TPR	FPR
8586	180.0	0.000	1600.0	450.0	0.00	0.459984	0.004751
8436	180.0	0.000	0.0	475.0	0.00	0.502829	0.015835
1101	0.0	0.100	2000.0	450.0	1.70	0.508488	0.029295
7053	45.0	0.000	1300.0	450.0	1.70	0.574778	0.039588
5274	30.0	0.000	1500.0	450.0	0.00	0.583670	0.042755
4261	25.0	0.075	1500.0	475.0	1.80	0.606306	0.089470
2009	15.0	0.000	1300.0	475.0	1.60	0.630558	0.098971
2005	15.0	0.000	1300.0	475.0	1.80	0.657235	0.107680
6756	45.0	0.000	0.0	475.0	0.00	0.662894	0.115598
2003	15.0	0.000	1300.0	500.0	1.60	0.679871	0.197150
7041	45.0	0.000	1300.0	500.0	1.70	0.702506	0.209026
7038	45.0	0.000	1300.0	500.0	0.00	0.710590	0.219319
3585	25.0	0.000	1500.0	500.0	1.70	0.714632	0.228820
1902	15.0	0.000	1500.0	500.0	0.00	0.722716	0.239905
3534	25.0	0.000	1600.0	500.0	0.00	0.722716	0.241489
1996	15.0	0.000	1300.0	525.0	1.65	0.737268	0.307997
7035	45.0	0.000	1300.0	525.0	1.70	0.744543	0.317498
313	0.0	0.000	1300.0	525.0	1.80	0.751819	0.325416
7032	45.0	0.000	1300.0	525.0	0.00	0.752627	0.334917
5306	30.0	0.000	1400.0	525.0	1.75	0.762328	0.368171
218	0.0	0.000	1500.0	525.0	1.75	0.770412	0.378464
1705	15.0	0.000	0.0	525.0	1.80	0.776071	0.388757
7683	45.0	0.075	1300.0	0.0	1.70	0.783347	0.395091
7681	45.0	0.075	1300.0	0.0	1.80	0.791431	0.405384
2640	15.0	0.075	1300.0	0.0	0.00	0.792239	0.416469
3987	25.0	0.050	1300.0	0.0	1.70	0.797090	0.420428
5665	30.0	0.050	1300.0	0.0	1.80	0.806791	0.432304
1973	15.0	0.000	1300.0	0.0	1.60	0.822959	0.445764
292	0.0	0.000	1300.0	0.0	1.65	0.842361	0.463183
291	0.0	0.000	1300.0	0.0	1.70	0.851253	0.475059
5329	30.0	0.000	1300.0	0.0	1.80	0.860954	0.486936
288	0.0	0.000	1300.0	0.0	0.00	0.861762	0.501188
3610	25.0	0.000	1400.0	700.0	1.65	0.861762	0.638163
1924	15.0	0.000	1400.0	0.0	1.65	0.877122	0.646873
243	0.0	0.000	1400.0	0.0	1.70	0.886823	0.664291
1922	15.0	0.000	1400.0	0.0	1.75	0.893290	0.678543
6961	45.0	0.000	1400.0	0.0	1.80	0.897332	0.683294
240	0.0	0.000	1400.0	0.0	0.00	0.898141	0.697546
3556	25.0	0.000	1500.0	0.0	1.65	0.908650	0.739509
5235	30.0	0.000	1500.0	0.0	1.70	0.919159	0.761679
3554	25.0	0.000	1500.0	0.0	1.75	0.925627	0.776722
3553	25.0	0.000	1500.0	0.0	1.80	0.929669	0.783056
192	0.0	0.000	1500.0	0.0	0.00	0.930477	0.799683
1827	15.0	0.000	1600.0	0.0	1.70	0.934519	0.836105
1825	15.0	0.000	1600.0	0.0	1.80	0.945028	0.858274
99	0.0	0.000	1700.0	0.0	1.70	0.945837	0.874901
6725	45.0	0.000	0.0	0.0	1.60	0.952304	0.889153
97	0.0	0.000	1700.0	0.0	1.80	0.956346	0.897862
1731	15.0	0.000	2000.0	0.0	1.70	0.959580	0.911322
5044	30.0	0.000	0.0	0.0	1.65	0.977365	0.927158
6723	45.0	0.000	0.0	0.0	1.70	0.987874	0.949327
5043	30.0	0.000	0.0	0.0	1.70	0.988682	0.950911
5042	30.0	0.000	0.0	0.0	1.75	0.995150	0.967538
5041	30.0	0.000	0.0	0.0	1.80	0.999192	0.975455
5040	30.0	0.000	0.0	0.0	0.00	1.000000	0.998416

Table 3: Set of cuts forming the upper envelope. Column heads a-e refer to the cut letters in Table 2.

References

[1] Edward Shuryak. Lectures on nonperturbative QCD (Nonperturbative Topological Phenomena in QCD and Related Theories). *https://arxiv.org/abs/1812.01509*, 2020.

[2] S. Porteboeuf, T. Pierog, and K. Werner. Producing Hard Processes Regarding the Complete Event: The EPOS Event Generator, 2010.

[3] Ynyr Harris. Instanton analysis. *https://gitlab.cern.ch/yharris/instantonanalysis*, Aug 2021.

[4] Enrico Bothmann, Gurpreet Singh Chahal, Stefan Höche, Johannes Krause, Frank Krauss, Silvan Kuttimalai, Sebastian Liebschner, Davide Napoletano, Marek Schönherr, Holger Schulz, and et al. Event generation with Sherpa 2.2. *SciPost Physics*, 7(3), Sep 2019.

[5] Simone Amorosoa, Deepak Karb, and Matthias Schott. How to discover QCD instantons at the LHC. *https://arxiv.org/pdf/2012.09120.pdf*, 2020.

[6] Scikit learn (Python library). *https://scikit-learn.org/stable/*, 2021.

[7] Nicholas Mitchell. Investigating the improvement of Instanton process classification via machine-learning methods. *https://gitlab.cern.ch/nimitche/instanton-ml-summer-project*, Sep 2021.

[8] Leo Breiman. Random Forests. *https://doi.org/10.1023/A:1010933404324*, pages 19–20, 2001.

[9] Scikit-learn. Permutation feature importance. *scikit-learn.org/stable/modules/permutation_importance*, 2021.

[10] Gregor Kasieczka, Benjamin Nachman, Matthew D. Schwartz, and David Shih. ABCDisCo: Automating the ABCD method with Machine Learning. *https://arxiv.org/pdf/2007.14400.pdf*, 2020.

[11] Keras (Python library). *https://keras.io/*, 2020.