CHRISTIAN MICHELSEN
NIELS BOHR INSTITUTE
UNIVERSITY OF COPENHAGEN

A PHYSICIST'S APPROACH TO MACHINE LEARNING – UNDERSTANDING THE BASIC BRICKS

SUPERVISOR:
TROELS PETERSEN
NIELS BOHR INSTITUTE
UNIVERSITY OF COPENHAGEN

Christian Michelsen

HTTPS://GITHUB.COM/CHRISTIANMICHELSEN

First printing, December 2019

Abstract

Here will be a decent abstract at some point™.

Contents

Foreword

The background for this masters's thesis, is that it is part of a so-called 4+4 Ph.D. project (also known as an integrated Ph.D.). The Ph.D. dissertation is about the use of machine learning and deep learning in the field of ancient genomics. Here ancient DNA is sampled and analysed with the hope of finding patterns, structure, in the genome which were previously unknown. The overall goal is two-fold. On the big scale it is the better understand human history in the broadest sense of the word history. Where did we come from, where did we go. On a much smaller scale, the goal is to understand local history and migration patterns; how did we end up where we did. It is with this background that this project should be seen: as an introduction to the general use of applied machine learning.

Part I

The first part of this thesis deals with the introductory theory of machine learning and its predictive power in estimating Danish housing prices.

Part II

The second part of this thesis deals with particle physics and the discriminatory power of machine learning for quark-gluon identification and subsequent analysis.

5. Quark Gluon Analysis

As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves; as I have shown elsewhere, the phenomena should only be used as a canon for our understanding. The paralogisms of practical reason are what first give rise to the architectonic of practical reason. As will easily be shown in the next section, reason would thereby be made to contradict, in view of these considerations, the Ideal of practical reason, yet the manifold depends on the phenomena.

5.1 Data Preprocessing

The data consists of 43 data files taken between 1991 and 1995 totalling 3.5 GB (Data). Along with this comes 125 files based on Monte Carlo (MC) simulations (8.4 GB) and additional 42 MC-files with only b-quark events (MCb) simulated (2.1 GB). The data files which are in the form of *Ntuples*, ROOT's data format [27], are converted to HDF5-files by using uproot [7]. While iterating over the Ntuples, some basic cuts are applied before exporting the data to HDF5. The first one being that the (center of mass) energy E in the event has to be within $90.8\,\mathrm{GeV} \leq E \leq 91.6\,\mathrm{GeV}$ to only use the Z peak data. The second one being that the sum of the momenta p_{sum} in each event is $32\,\mathrm{GeV} \leq p_{\mathrm{sum}}$ to remove any $Z \to \tau^+\tau^-$ events. To ensure a primary vertex, at least two good tracks are required where a good track is defined as having 7 TPC hits and $\geq$ 1 silicon hit. Finally it is required that the cosine of the thrust axis polar angle, which is the angle between the trust axis and the beam, is less than or equal to 0.8 to avoid any low angle events since the detector performance worsens significantly in that region. These cuts were standard requirements for the ALEPH experiment.

One last cut which was experimented with was the threshold value for *jet matching*. The jet matching is the process of matching the jet with one of the final state quarks. The jet is said to be matched if the dot product of between the final quark momentum and the jet momentum is more than then threshold value. Higher thresholds means cleaner jets but at the expense of less statistics. A jet matching threshold of 0.90 was found to be a good compromise between purity and quantity where 97.8 % of all 2-jet events are matched and 96.7 % of all other jets were matched[1].

The data structure is quite differently structured in the Ntuples compared to normal structured data in the form of tidy data [81]. The data is organized such that one iterates over each event where the variables are variable-length depending on the number of jets in the events; this is also known as *jagged* arrays. The data is un-jagged[2] before exporting to HDF5-format and only the needed variables are kept. This reduces the total output file to a 2.9 GB HDF5-file for both Data, MC, and MCb.

The number of events for each number of jets can be seen in Table 5.1 for the Data and in Figure 5.2 for the MC and MCb.

[1] Compare this to 98.5 % and 97.8 % for a threshold of 0.85 or 95.9 % and 93.9 % for a threshold of 0.95.

[2] Such that e.g. a 3-jet event will figure as three rows in the dataset.

	jets	events
2	2 359 738	1 179 869
3	3 619 290	1 206 430
4	854 336	213 584
5	52 775	10 555
6	510	85
Total	6 886 649	2 610 523

Table 5.1: The dimensions of the dataset for the actual Data. The numbers in the jet columns are the number of events multiplied with the number of jets; e.g. $85 \cdot 6 = 510$.

	jets	events
2	7 293 594	3 646 797
3	10 780 890	3 593 630
4	2 241 908	560 477
5	103 820	20 764
6	588	98
Total	20 420 800	7 821 766

Table 5.2: The dimensions for the MC and MCb datasets.

5.2 *Explanatory Data Analysis*

Since the machine learning models are only trained on the three vertex variables `projet`, `bqvjet`, and `ptljet` – see chapter 4 for a deeper introduction to these variables – these variables will be the primary focus of this section. Given the fact that MC-simulated data exists, the truth of each simulated event is also known. This allows us visualize the difference between the different types of quarks. In the MC simulation each event are generated such that the type of quark, or *flavor*, is known and assigned the variable `flevt`. The mapping from flavor to `flevt` is:

Flavor:	*bb*	*cc*	*ss*	*dd*	*uu*
`flevt`:	5	4	3	2	1

In addition to knowing the correct flavor, we define that an event is *q-matched* if one, and only one, of the jets are assigned to one of the quarks, one, and only one, of the jets are assigned to the other quark, and no other jets are matched to any of the quarks. We can then define what constitutes a b-jet: if it has `flevt` $= 5$, the entire event is q-matched, and the jet is matched to one of the quarks. Similarly we define c-jets only with the change that `flevt` $= 5$, and uds-jets with `flevt` $\in \{1,2,3\}$. A gluon jet is defined as an any-flavor event which is q-matched but the jet is not assigned to any of the quarks. Strictly speaking, this means that g-jet is not 100 % certain of being a gluon, however, since the MC simulation does not contain this information this is the only option. Due to the q-match criterion this also means that some jets are assigned the label "non-q-matched" which is regarded as background. The distribution of different types of jets can be seen in Table 5.3 and shown as relative numbers in Table B.1 in the appendix.

With the criteria defined above for what constitutes a specific type of jet the 1D-distributions for the three vertex variables is plotted in Figure 5.1. For all three subplots the histograms are show with a logarithmic y-axis, all b-jets in blue, c-jets in red, g-jets in green and all jets in orange. In fully opaque color are shown the distributions for 2-jet events, in dashed (and lighter color) 3-jet

	b	c	uds	g	non-q-matched
2	2 713 454	944 380	2 125 900	0	1 509 860
3	2 433 878	964 212	2 129 218	3 365 969	1 887 613
4	326 264	156 332	336 548	1 012 198	410 566
5	10 332	5960	12 668	54 525	20 335
6	42	26	52	320	148
Total	5 483 970	2 070 910	4 433 012	4 604 386	3 828 522

Table 5.3: Number of different types of jets for MC and MCb. See also Table B.1 in the appendix for relative numbers.

events, and in semi-transparent 4-jet events. In the left subplot the `projet` variable is plotted where it can be seen that high values of `projet` tend to indicate b-jets. In the middle subplot `bqvjet` is plotted which shares many similarities with the `projet` -variables, including that high values indicate b-jets. In the right subplot the `ptljet` is plotted. This variable has many zeros in it which correlates with mostly with gluon[3] and large values are mostly due to b-jets. In general it is clear to se how the differences in distribution between the 2-, 3-, and 4-jet events are minor, with the one exception of 2-jet events which does not contain any gluons at all.

[3] Around 98 % of all g-jets are zeros compared to $\sim$ 82 % for c-jets and $\sim$ 70 % for b-jets.

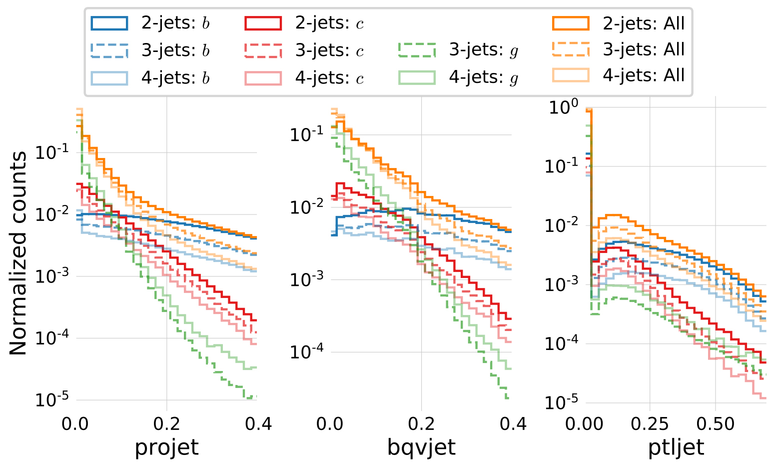


Figure 5.1: Normalized histograms of the three vertex variables: `projet` , `bqvjet` , and `ptljet` . In blue colors the variables are shown for true b-jets, in red for true c-jets, in green for true g-jets, and in orange for all of the jets (including non q-matched). In fully opaque color are shown the distributions for 2-jet events, in dashed (and lighter color) 3-jet events, and in semi-transparent 4-jet events. Notice the logarithmic y-axis, that there are no g-jets for 2-jet events (as expected), and that all of the distributions are very similar not matter how many jets.

Even though there are only three vertex variables, it is difficult to properly get an intuition about how easily separated they different types of jets are. Since there are millions of points a single 3D scatter plot quickly becomes overcrowded in one wants to plot all jets. We apply dimensionality reduction from the three dimensions down to two dimensions by using the UMAP algorithm [55]. Within recent years the field of dimensionality reduction algorithms has grown a lot from just the typical (linear) principal component analysis to also include non-linear algorithms. The t-SNE algorithm [75] deserves an honorable mention since this algorithm revolutionized the usage of (nonlinear) dimensionality reduction algorithms in e.g. bioinformatics [73, 79] yet its mathematical foundation has strongly been improved with the never, faster UMAP algorithm [55] which usage is also expanding [20, 21, 33].

The aim of UMAP, short for Uniform Manifold Approximation and Projection, is to correctly identify and preserve the structure, or topology, of the high-dimensional feature space in a lower-dimensional output space. It does so by trying to stitch together local manifolds in the high-dimensional feature space such that the difference between the high- and low-dimensional representations is minimized according to the cross-entropy such that both global structure and local structure is preserved [55]. Compared to t-SNE the approach in UMAP has an algebraic topological background compared to the more heuristic approach taken by t-SNE. Note that the UMAP algorithm is not provided any information about which jets are which types.

The UMAP algorithm has several hyperparameters, where two of the most important ones are the number of neighbors `n_neighbors` which controls the priority between correctly preserving the global versus the local structure, and the `min_dist` which defines how tightly together UMAP is allowed to cluster the points in the low-dimensional representation. To properly choose the best combination of `n_neighbors` and `min_dist` a grid search with `n_neighbors` $\in \{10, 50, 100, 250\}$ and `min_dist` $\in \{0, 0.2, 0.5\}$ is performed. This is shown for 4-jet events in Figure B.1 in the appendix. In this case the choice of best combination of `n_neighbors` and `min_dist` is subjective at best, but it was judged by the author that `n_neighbors` $= 250$ and `min_dist` 0.2 gave the best compromise between preserving local and global structure. The results of running UMAP on 4-jet events can be seen in Figure 5.4. Here the millions of points are plotted using Datashader [8] to avoid over-plotting and colored according to the jet type. From the figure it is seen how there are some clear, blue b-jet clusters, however, most of the data seem to be a mix of g- and uds-jets. The plots with the same UMAP parameters for 3-jet and 2-jet events are seen in Figure 5.3 and 5.4.

These figures suggests that it should be possible to discriminate the b-jets from the other jets somewhat, however, no clear separation is expected. The t-SNE algorithm was also tested but showed inferior performance compared to UMAP, see Figure B.2 in the appendix for an example of this.

The correlation between the vertex variables can be seen in Figure 5.5, where the upper diagonal shows the linear correlation ρ and the lower diagonal shows the (estimate of the) MIC non-linear correlation MIC_e. Here it ca be seen that `projet` and `bqvjet` correlate mostly whereas the other variables correlate a lot less. Had they all correlated a lot, it would be more difficult to extract any meaningful insights from the system at it would contain less information.

figures/quarks/df_UMAP-X=1120952-n_neighbors=25

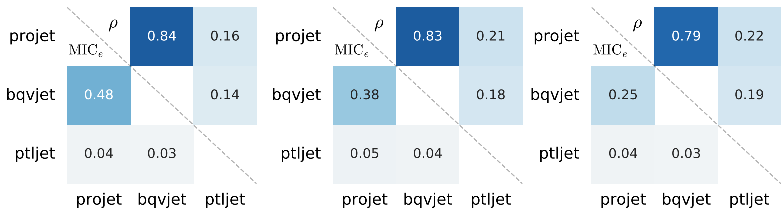
Figure 5.2: Visualization of the vertex variables for the different categories: true b-jets in blue, true c-jets in red, true uds-jets in green, true g-jets in orange, and non q-matched events in purple. The clustering is performed with the UMAP algorithm which outputs a 2D-projection. This projection is then visualized using the Datashader which takes takes care of point size, avoids over and underplotting, and color intensity.

figures/quarks/df_UMAP-X=1078089-n_neighbors=25

Figure 5.3: UMAP visualization of vertex variables for 3-jet events.

figures/quarks/df_UMAP-X=729358-n_neighbors=250

Figure 5.4: UMAP visualization of vertex variables for 2-jet events.



(a) 2-jet events (b) 3-jet events (c) 4-jet events

Figure 5.5: Correlation of the three vertex variables for 2-, 3- and 4-jet events.

5.3 *Loss and Evaluation Function*

In contrary to the housing prices subproject the goal in this project is to predict the class of particles, or the types of jets, where the so-called *signal* observations[4] are often assigned the label 1 and *background* observations 0. The combination of this being a *classification* problem (compared to a regression problem) along with the fact all the variables are actual measurements from a particle physics accelerator means that the issue of outliers is negligible. This also means that the problem of finding a robust loss function is non-existent since the in classification loss is already bounded in the $[0, 1]$-interval.

Classically *accuracy* is often used as loss function for classification which is simply the fraction of correct predictions, however, accuracy as a metric suffers a lot when handling *imbalanced* data: when the ratio between the number of instances of each class is not approximately $(50 : 50)\%$. The problem is that if the sample contains $90\,\%$ background and only $10\,\%$ signal, then a simple model which simply predicts everything to be background will have a $90\,\%$ accuracy.

To circumvent this issue, the area under the ROC curve (AUC) is used, where the ROC[5] curve is the the *signal efficiency* $\varepsilon_{\mathrm{sig}}$ of the ML model plotted as a function of the *background efficiency* $\varepsilon_{\mathrm{bkg}}$. The definition of these two measures are:

$$\varepsilon_{\mathrm{sig}} = \frac{S_{\mathrm{sel}}}{S_{\mathrm{tot}}}, \qquad \varepsilon_{\mathrm{bkg}} = \frac{B_{\mathrm{sel}}}{B_{\mathrm{tot}}}, \tag{5.1}$$

where S_{sel} are signal events that were also selected (predicted) as signal by the ML model, $S_{\mathrm{tot}} = S_{\mathrm{sel}} + S_{\mathrm{rej}}$ is the total number of signal events (the selected and rejected), and likewise for background events B. Within the machine learning community the signal efficiency is called the true positive rate (TPR) and the background efficiency the false positive rate (FPR). For the rest of this project, the AUC will be the evaluation function $f_{\mathrm{eval}} = \mathrm{AUC}$, however, since this metric does not work on single observations it cannot be used as the loss function. Instead we will use the *log-loss* as the loss function[6] which not not only is differentiable for single predictions, compared to AUC, but also takes the certainty of the prediction into account. When using tree-based algorithms or neural networks one can extract not only whether or not a single observation is classified as signal or background but also a prediction score. This is a number in the $[0, 1]$-interval and the closer to 1 the score is, the

[4] Often called *signal events*, however, this term would require that each event constitutes a single data point in the dataset which it does not here.

[5] Receiver Operating Characteristic.

[6] In the context of machine learning this is the same as the *cross entropy*.

more certain the model is of the prediction being signal. Given the prediction score $\hat{y}$ and the true label y, the log-loss $\ell_{\log}$ is calculated as:

$$\ell_{\log} = -y \log \hat{y} - (1-y) \log (1-\hat{y}). \tag{5.2}$$

This is visualized in Figure 5.6. Here it can be seen how the loss changes as a function of the prediction score. Notice that when $y = 0$ the loss for $\hat{y} = 1$ diverges towards ∞ and likewise with $y = 1$ and $\hat{y} = 0$ (since $\log 0$ diverges to $-\infty$).

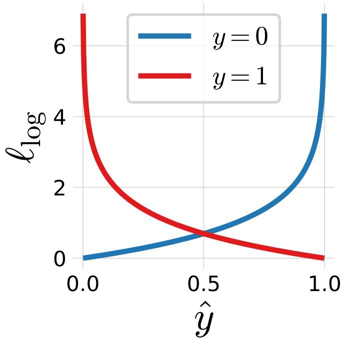


Figure 5.6: Plot of the log-loss $\ell_{\log}$.

5.4 b-Tagging Analysis

The ability to discriminate between the different types of particles produced in a collision is obviously import to understand the results. Today much work go into tagging algorithms from b-tagging in ATLAS and CMS [66] but this work started even decades ago. That b-quarks are tagged specifically is both due to b-quarks having more unique characteristics compared to e.g. c-quarks and are thus easier to tag, but also the fact that b-quarks are the second-heaviest of the quarks and are measured to better understand CP[7]-violation at LHC-b, contributes to the choice of tagging b-quarks. In ALEPH Proriol et al. [60] started the work of comparing different methods for b-tagging already in 1991. They concluded that a neural network had the best performance compared to e.g. a linear (Fisher) discriminant. The neural network used was a 3-layer neural network (NN) trained on nine variables and the output `nnbjet`. For this of this project this pre-trained network will be called NNB.

[7] Short for charge-parity.

The data are split[8] into training and test sets in such a way that the individual jets in an event are not split. As such, the splitting is performed at event-scale in a $(80:20)\%$ train-test ratio.

[8] After removing all low-energy jets such that all events that contain any jets with an energy of less than 4 GeV are removed.

5.4.1 b-Tagging Hyperparameter Optimization

Compared to the housing prices dataset, the number of observations N is a lot larger, although the dimensionality M is much smaller ($3 \ll 143$). Therefore both XGBoost (XGB) and LightGBM (LGB) were included as models initially since their performance in the housing dataset was very similar but LightGBM was expected to quite a lot faster on this dataset, which also turned out to be the case. The models were hyperparameter optimized (HPO) using random search (RS) since the Bayesian optimization (BO) did not show any performance gains compared to RS. They were run with 5-fold cross validation and early stopping with a patience of 100. The PDFs for the random search for the LightGBM model can be seen in Table 5.4, and the ones for XGBoost in Table B.2 in the appendix. The random search has been run with 100 iterations for LightGBM and only 10 for XGBoost since XGBoost is slow at fitting datasets of this size[9]. The results of the HPO for 3-jet and 4-jet events can be seen in Figure 5.7. For 3-jets it can be seen how most of the iterations share about the same performance within 1σ, however some

Hyperparameter	Range
`subsample`	$\mathcal{U}(0.4, 1)$
`colsample_bytree`	$\mathcal{U}_{\text{trunc}}(0.4, 1, 2)$
`max_depth`	$\mathcal{U}_{\text{int}}(-5, 63)$
`num_leaves`	$\mathcal{U}_{\text{int}}(7, 4095)$

Table 5.4: Probability Density Functions for the random search hyperparameter optimization process for the LightGBM model. For an explanation of $\mathcal{U}_{\text{trunc}}$, see section 5.5. All negative values of `max_depth` are interpreted as no max depth by both LGB and XGB.

[9] See page 70 for a discussion of the timings.

iterations have a significantly decrease in performance. For 4-jets there are not any iterations which share the same bad performance relative to the others as some of the 3-jets.

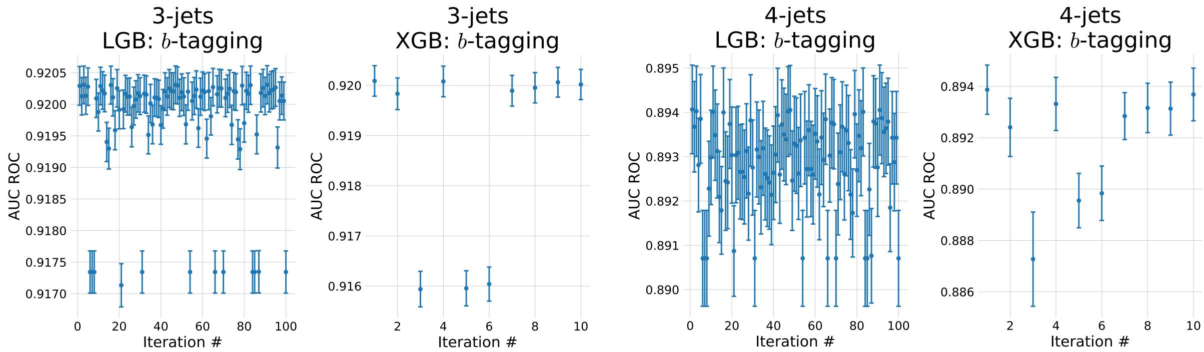


Figure 5.7: Hyperparameter Optimization results of b-tagging with random search. From left to right, we have A) 100 iterations of RS with LGB on 3-jets, B) 10 iterations of RS with XGB on 3-jets, C) 100 iterations of RS with LGB on 4-jets, D) 10 iterations of RS with XGB on 4-jets. Notice the different ranges on the y-axes.

The relationship between the different hyperparameters in 4-jet events can be seen in the parallel coordinate plot in Figure 5.8. First of all the importance of the column downsampling `colsample_bytree` variable is significant: all of the low-performing hyperparameter sets have a low value of this hyperparameter. Since $M = 3$ for the vertex variables this makes logical sense; using only $\mathrm{int}(\sim 0.5 \cdot 3) = 1$ variable[10] the model cannot properly learn the structure in the data. Compared to the column downsampling, the other hyperparameters are notably less important. The same overall conclusion can be inferred in the 3-jet case, see Figure B.3 in the appendix.

[10] See section 5.5 for a deeper discussion about the `colsample_bytree` hyperparameter.

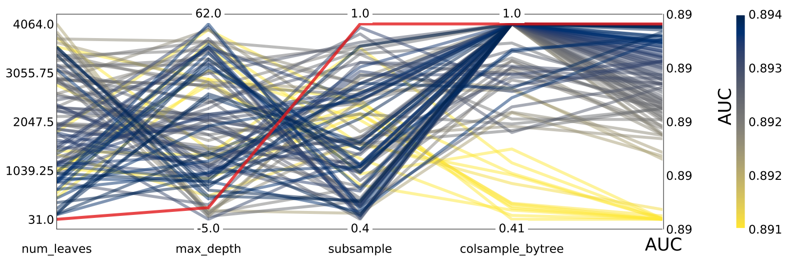


Figure 5.8: Hyperparameter optimization results of b-tagging for 4-jet events. The results are shown as parallel coordinates with each hyperparameter along the x-axis and the value of that parameter on the y-axis. Each line is an event in the 4-dimensional space colored according to the performance of that hyperparameter as measured by AUC from highest AUC in dark blue to lowest AUC in yellow. The single best hyperparameter is shown in red.

5.4.2 *b-Tagging Results*

The prediction score for the b-tagging models is usually called the b-tag and will be written as β_{tag}. The distribution of β_{tag} for the two HPO-optimized models, LGB and XGB, together with the pre-trained neural network NNB can be seen in Figure 5.9 for 4-jet events and in B.4 in the appendix for 3-jet events. Notice the strong match between the NNB and LGB models. The XGB model has almost no high b-tags $\beta_{\mathrm{tag}} > 0.8$, but a majority of b-tags in the very low end. This indicates that the XGBoost has focussed on the background events compared to the signal events, whereas the NNB and LGB models have focused more on the signal events.

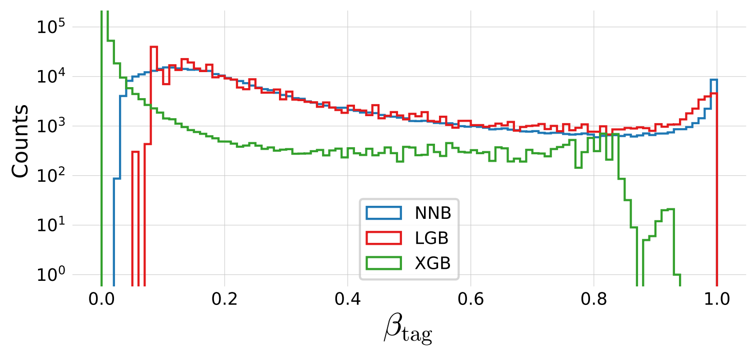


Figure 5.9: Histogram of b-tag scores β_{tag} in 4-jet events for NNB (the neural network pre-trained by ALEPH, also called `nnbjet`) in blue, LGB in red, and XGB in green.

Even though the distributions of b-tags are different between the three models, the real performance plot for classification is the ROC curve seen in Figure 5.10 for 4-jet events. Here the signal efficiency ε_{sig} is plotted as a function of the background efficiency ε_{sig} with the AUC shown in the bottom right corner. The LGB and XGB models performs similarly well with an AUC $= 0.896$ compared to the NNB with AUC $= 0.884$. The differences between the models are even smaller for 3-jet events seen in Figure B.5 in the appendix. In general the LGB and XGB models are so similar that they cannot be distinguished from another in any of the plots and their difference in AUC is on the forth decimal point. However, the LGB model is several times faster than the XGB model. In comparison, 10 iterations of HPO using RS on 3-jet events with XGB took more almost 34 hours on HEP[11] compared to just 23 hours for 100 iterations for LGB. The same performance difference was seen in 4-jet events where the timings were 4 hours for XGB compared to 2.5 hours for LGB, and thus XGB is dropped in all subsequent analysis.

[11] The local computing cluster.

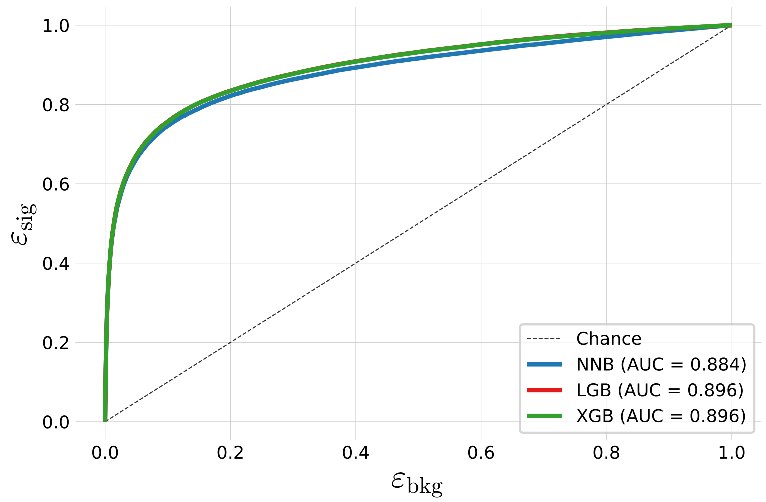


Figure 5.10: ROC curve of the three b-tag models in 4-jet events for NNB (the pre-trained neural network trained by ALEPH, also called `nnbjet`) in blue, LGB in red, and XGB in green. In the legend the area under curve (AUC) is also shown. Notice that the LGB and XGB models share performance and it is thus due to overplotting that only the green line for XGB can be seen. In the machine learning community the background efficiency ε_{bkg} is sometimes know as the false positive rate (FPR) and the signal efficiency ε_{sig} as the true positive rate (TPR).

5.4.3 *b-Tagging Model Inspection*

To get a better understanding of the trained LGB model, the global SHAP feature importances can be seen in Figure 5.11 for 4-jet events. First of all it is noted that the `projet` has global feature importance of 57.32 %, `bqvjet` 29.16 %, and `ptljet` 13.52 %. For all three variables it is seen how most of the points have many small feature values which has a negative impact on the model output however small. Especially the `ptljet` has many features with a low value (0 in fact) yet this does not pull the model too much towards background events compared to if a jet has a high value of `ptljet` which has a strong, positive impact on the output prediction.

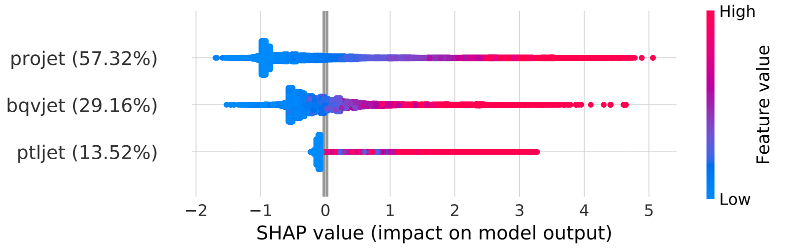


Figure 5.11: Global feature importances for the LGB b-tagging algorithm on 4-jet events. The normalized feature importance is shown in the parenthesis and shov ne SHA .

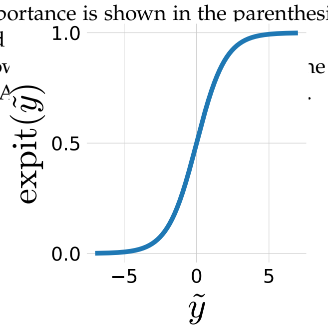


Figure 5.12: The expit function.

In regression, the model output is a continuous prediction $\hat{y}_{\text{reg}} \in \mathbb{R}$. In classification what is actually happening under the hood is that the model predicts a value $\tilde{y} \in \mathbb{R}$ which is transformed to a number in the $[0,1]$-interval via the *expit* function:

$$\text{expit}(\tilde{y}) = \frac{e^{\tilde{y}}}{1+e^{\tilde{y}}} \equiv p, \tag{5.3}$$

where p is a number in the $[0,1]$-interval. The expit function is also sometimes known as the logistic function and is visualized in Figure 5.12. Its inverse is the *logit* function:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \tilde{y}, \tag{5.4}$$

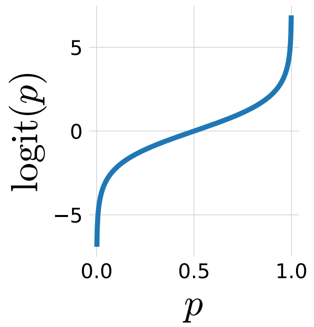


Figure 5.13: The logit function.

which is visualized in Figure 5.13. The fraction in equation (5.4) is called the *odds* and the logit-transformed value of p, $\text{logit}(p) = \tilde{y}$, is thus sometimes called the *log-odds*. It is in this log-odds space that LightGBM makes its predictions and the SHAP values in Figure 5.11 are also in log-odds space. The additivity[12] of SHAP is in this log-odds space.

With this in mind, single predictions of the LGB b-tagging model can be understood with SHAP which Figure 5.14 is an example of. This figure shows the logic behind the models prediction for this particular jet. That the bias is negative reflects that there is a majority of background compared to signal[13]. This particular event has `projet` $= 1.003$, `bqvjet` $= 0.529$, and `ptljet` $= 0$. In the plot it is seen how this high value of `projet` has the greatest impact on the model prediction, while the medium value of `bqvjet` also pushes the model prediction towards a signal-prediction. The four bars in the left part of the plot are all in log-odds space and their sum is shown as the blue bar to right, where the right y-axis shows the value in probability space $p \in [0,1]$. This jet was in fact a b-jet.

[12] See also section 2.8.

[13] There are 22.1 % b-jets in the 3-jet training set.

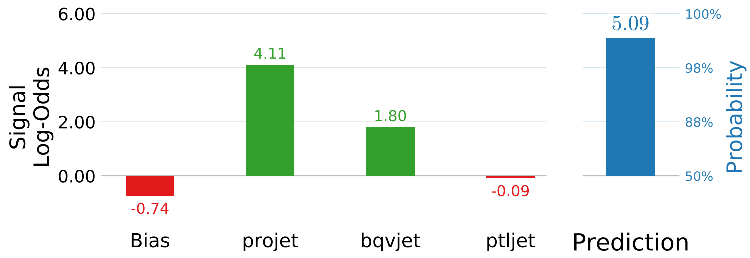


Figure 5.14: Model explanation for the 3-jet b-tagging LGB model for a b-like jet. The first column is the bias of the training set which acts as the naive prediction baseline, the rest are the input data variables. On the right hand side of the plot is the model prediction shown. The left part of the plot is shown in log-odds space, the right part in probability space. The negative log-odd values are shown in red, positive ones in green, and the prediction value in blue.

5.5 Truncated Uniform PDF

Initially when plotting the HPO performance as a function of iteration, it was seen how there were three significant plateaus, where the highest plateau (i.e. highest AUC value and thus best score) was only seen in the very first iteration. It was quickly realized that this was due to the very first iteration was being run with the default values of the LGB and XGB models in the custom implementation by the author. However, what was not understood was why this value was performing so much better than random sets of hyperparameters[14]. During the debugging process the column downsampling `colsample_bytree` was diagnosed to be the culprit. The default value is `colsample_bytree` $= 1$, however, the probability density function (PDF) used in RS for this parameter was $\mathcal{U}(0.4, 1)$ which was expected to give the same performance as the default value for large values of `colsample_bytree`. By inspecting the source code of LightGBM it was realized that the model takes the integer of the column downsampling multiplied with the total number of features if the column downsampling is less than 1 [6]. This means that no matter how close to 1 the column downsampling get, the integer value of the total number of columns get floored to 2 at max, compared to when the column downsampling is exactly 1 which it only is for the default values.

To deal with this problem a new PDF was developed on top of the existing ones in Scipy, the truncated uniform PDF: $\mathcal{U}_{\text{trunc}}(a, b, c)$. This PDF first generates a random number x from a uniform distribution between a and c. Then if x is larger than b it is floored to b. In this way, it is possible to both get values of x in the interval $[a, b]$ but also values exactly equal to b. The value of c controls how often these "overflow" values of x are generated.

[14] LightGBM and XGBoost of course have chosen their default parameters smartly, however, it one would not expect them to outperform other sets of hyperparameters that clearly.

5.6 g-Tagging Analysis

The trained b-tagging LGB model is a jet-based model which provides a b-tag score β_{tag} to a jet. This also means that each of the jets e.g. a 4-jet event can get a b-tag: $\boldsymbol{\beta}_{\text{tag}} = [\beta_{\text{tag}_1}, \beta_{\text{tag}_2}, \beta_{\text{tag}_3}, \beta_{\text{tag}_4}]$. Using $\boldsymbol{\beta}_{\text{tag}}$ one can train a new model on the events, compared to individual jets, where signal events are defined to be q-matched events where the gluons are assigned the $n-2$ lowest b-tag scores for n-jet events; e.g. $\boldsymbol{\beta}_{\text{tag}} = [0.95, 0.89, 0.15, 0.07]^\top$ for the four jets

$[b, \bar{b}, g, g]$. This event-based process will be called g-tagging and the trained model will return a g-tag score written as γ_{tag}. Compared to the b-tagging LGB model, this model will allow one to extract entire events which contains a clear identification of gluons versus non-gluons.

5.6.1 *Permutation Invariance*

Since the b-tags are only based on the vertex variables, the goal of the g-tag is to also be constructed in an un-biased way with respect to the jet energy E_{jet}. However, even though $\beta_{\text{tag}} \perp\!\!\!\perp E_{\text{jet}}$ and $\gamma_{\text{tag}} = f(\beta_{\text{tag}})$, it turned out that $\gamma_{\text{tag}} \not\!\perp\!\!\!\perp E_{\text{jet}}$, where $a \perp\!\!\!\perp b$ is defined to mean that a is independent[15] of b and f is an unknown function. This was because the ordering of the jets within the event was energy-dependent: they sorted according to their E_{jet}.

[15] And $\not\!\perp\!\!\!\perp$ means not independent.

This meant that the different components in $\boldsymbol{\beta}_{\text{tag}}$ had different importances, even though they should be equally important. Instead of defining $\boldsymbol{\beta}_{\text{tag}}$ as a vector it should instead be seen as a set[16] $\boldsymbol{\beta}_{\text{tag}} = \{\beta_{\text{tag}_1}, \ldots, \beta_{\text{tag}_n}\}$. The g-tagging model trained on the events should thus be *permutation invariant*[17] with regards to the input variables. The category of permutation invariant (and equivariant[18]) neural networks in the deep learning community has seen an huge development within recent years where the paper from Zaheer et al. [82] in 2017 was quite influential, however also other examples exists [62, 40]. Yet, the same development cannot be said to have happened within the more classic machine learning field.

[16] Since sets have no inherent order.

[17] $f(\mathbf{x}) = f(\tau(\mathbf{x}))$ for any permutation τ on an input vector $\mathbf{x}$.

[18] $\tau(f(\mathbf{x})) = f(\tau(\mathbf{x}))$ for any permutation τ on an input vector $\mathbf{x}$.

Although not being a novel software-technical solution, the problem was circumvented by two simple, different approaches: 1) by simply shuffling the inputs variables independently for each observation (row) in the dataset, and 2) training on all possible permutations of the variables in the dataset. The second approach can be seen as a feature augmentation technique where the data is artificially increased with factor of n factorial: $N \to n! \cdot N$ where N is the number of observations (rows) and n is the number of jets. These two methods were tested along with the original order of the dataset.

5.6.2 *g-Tagging Hyperparameter Optimization*

Four LightGBM models, two for 3-jet events and two for 4-jet events, were trained and hyperparameter optimized for both the the energy ordered and shuffled[19] data sets with 100 iterations of random search with the same PDFs as for the b-tagging, see Table 5.4, and 5-fold cross validation and early stopping with a patience of 100. The results of the HPO can be seen in Figure 5.15. Here the two 3-jets models are seen in the two plots to the left, and the two 4-jets to the right. The very left plot shows the 3-jet energy-ordered (no permutation) performance as a function of iteration number, which was also where the issues mentioned in section 5.5 were first discovered. Here the difference between the how many

[19] The method with all permutations was trained using the same hyperparameters as the best ones for the shuffled model.

of the three variables, the three b-tags, are included is seen as three clear plateaus. The three plateaus are also seen in the 3-jet events that were shuffled, however, with more variation in each plateau, along with a drop in performance. For the 4-jet events the plateaus are not as apparent but it can still be seen how some of the iterations how a significantly lower score than others. The parallel plots for the four fits can be seen in Figure B.7–B.10 in the appendix.

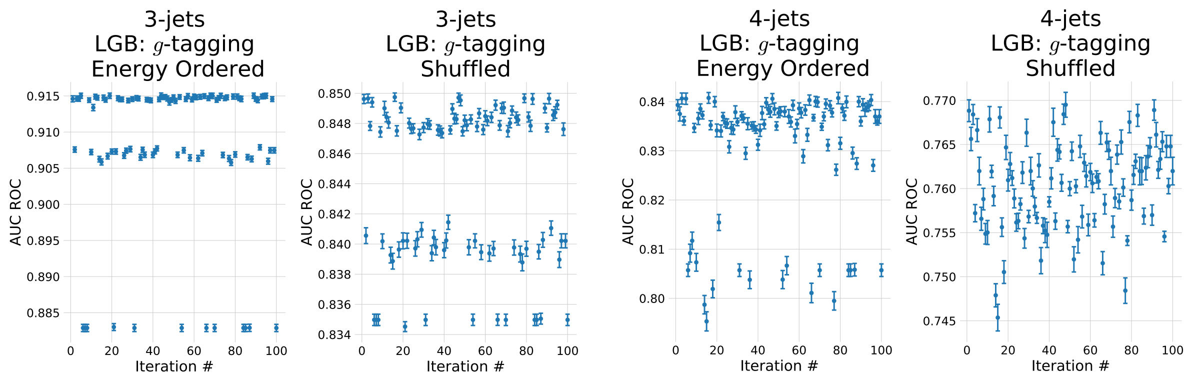


Figure 5.15: Hyperparameter Optimization results of g-tagging with 100 iterations of random search with LGB. From left to right, we have A) 3-jet events energy-ordered (no permutations), B) 3-jet events row-shuffled, C) 4-jet events energy-ordered, D) 4-jet events row-shuffled. Notice the different ranges on the y-axes.

5.6.3 *PermNet*

In addition to the LGB models, a permutation invariant neural network called PermNet based on the Deep Sets paper [82] implemented in Tensorflow [10] by Faye [36] was also tested. Zaheer et al. [82] showed that $f(X)$ is permutation invariant if and only if it can be decomposed in the following way:

$$f(X) = \rho\left(\sum_{x\in X}\phi(x)\right). \tag{5.5}$$

for suitable transformations ρ and ϕ (which the neural network learns[20]). The PermNet was trained using three layers[21] with leaky ReLU [54] as the activation function and ADAM [49] as the optimizer optimizing the log-loss. The network was trained with early stopping with a patience of 50 epochs and a batch size of 128. A visual overview of the PermNet architecture can be seen in Figure B.11 in the appendix.

[20] This is possible since neural networks are universal function approximators [44].

[21] Where the two hidden layers have 128 and 64 neurons in each.

5.6.4 *1D Comparison of LGB and PermNet*

To better understand the difference between the difference between the LGB and PermNet models, a small comparison was made. This comparison was constructed by summing the b-tag scores in the n-jet event together $\sum_i^n \beta_{\mathrm{tag}_i}$. The β_{tag_i} are summed together since this turns the problem into a 1D problem that is easy to visualize, the sum of numbers is a permutation invariant function, and is similar to the simplest functions of ρ and ϕ in equation (5.5): the identity function. The 1D models are fit to the training events and then a linear scan from $\sum_i^n \beta_{\text{tag}_i} = 0.4$ to 3.1 is made to see how the predicted g-tags γ_{tag} distribute. This is shown in Figure 5.16 for 4-jet events. Here the value of γ_{tag} is shown for the two models together with the fraction of signal to background in each bin. If the g-tag score should resemble a true probability it would be expected to follow the signal ratio, e.g. a model should predict $\gamma_{\text{tag}} = 0.9$ if there is 90 % signal in that bin. In the figure it is seen how the PermNet does a great job at fitting the signal fraction, however, the LGB model also does a decent job. Remember that none of these models were shown the signal fraction explicitly, only the b-tag sum and a signal-or-background label. The distribution of signal and background together with the distribution of cuts made by the LGB model can be seen in Figure B.12 in the appendix. The similar plots for 3-jet events are plotted in Figure B.13 and B.14, both in the appendix.

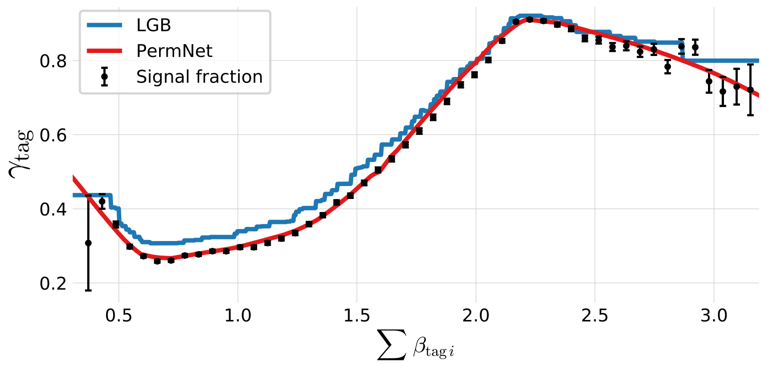


Figure 5.16: Plot of the (1D) g-tag scores for 4-jet events as a function of $\sum \beta_i$ for the LGB model in blue and the PermNet model in red. The signal fraction (based on the signal and background histograms in Figure B.12) is plotted as black error bars where the size of the error bars is based on the propagated uncertainties of the signal and background histogram assuming Poissonian statistics.

It can be concluded, at least in 1D, that both LGB and PermNet are able to capture the inherent structure in the (1D) data.

5.6.5 *g-Tagging Results*

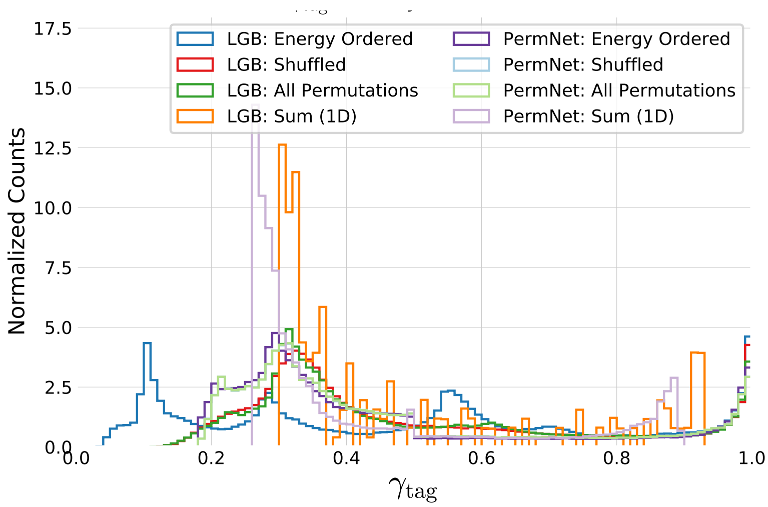


Figure 5.17: Histogram of g-tag scores (model prediction) in 4-jet events for XGB: Energy Ordered in blue, XGB: Shuffled in red, XGB: All Permutations in green, XGB: Sum 1D in orange, PermNet: Energy Ordered in purple, PermNet: Shuffled in light-blue, PermNet: All Permutations in light-green, PermNet: Sum 1D in light-purple. Here XGB and PermNet are the two different type of models and "Energy Ordered", "Shuffled", "All Permutations", and "Sum 1D" are the different methods used for making the input data permutation invariant.

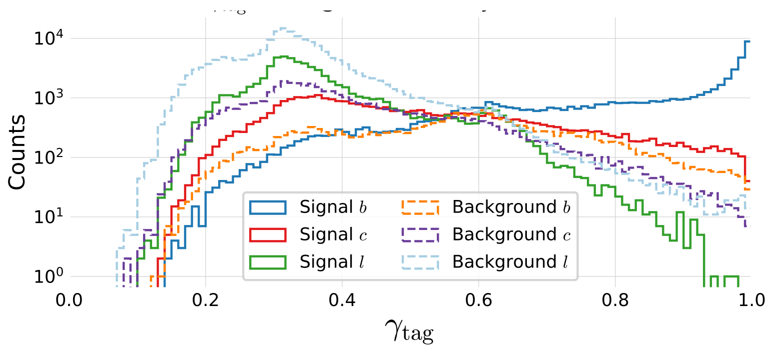


Figure 5.18: Histogram of g-tag scores (model prediction) from the XGB-model in 4-jet events for b signal in blue, c signal in red, l signal in green, b background in orange, c background in purple, l background in light-blue.

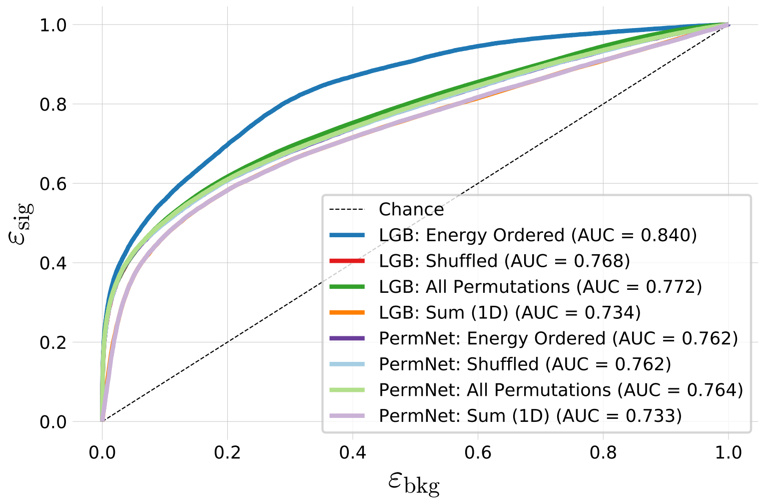


Figure 5.19: ROC curve of the eight g-tag models in 4-jet events. First one in dashed black is the ROC curve that you get by random chance. The colors are the same as in Figure 5.17 and in the legend also the Area Under the ROC curve (AUC) is shown. Notice that the XGB model which uses the energy ordered data produced the best model, however, this model is not permutation invariant. Of the permutation invariant models (the rest), the XGB model trained on all permutations of the b-tags performs highest. The lowest performing models are the two models trained only on the 1-dimensional sum of b-tags, as expected, however, still with a better performance than expected by the author.

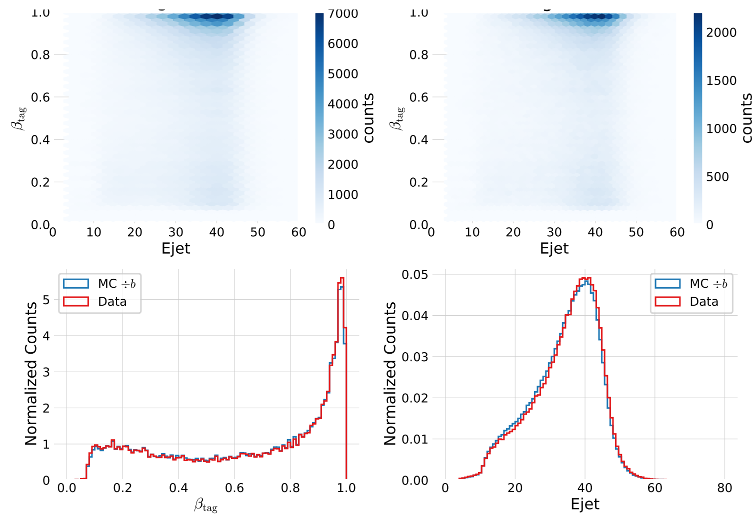


Figure 5.20: Comparison of the b-tag and jet energy (`Ejet`) distributions for Monte Carlo (MC) versus data. In the top row the 2D-distributions are shown for MC on the left (without the extra MCb samples) and data on the right. In the bottom row the 1D marginal distrubtions are shown for the b-tag and the jet energy with data in red and Monte Carlo ones in blue. Notice the the almost identical distributions in b-tag.

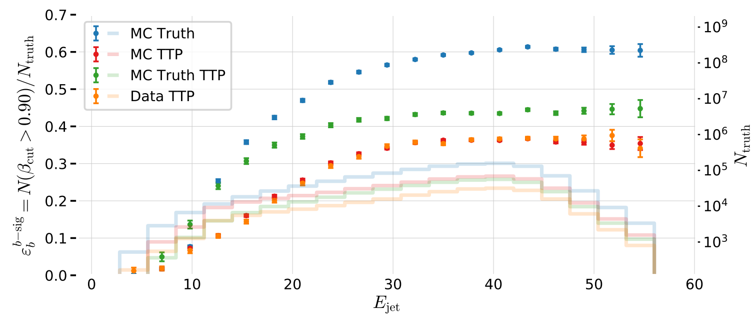


Figure 5.21: Efficiency of the b-tags for b-jets in the b-signal region for 3-jet events, $\varepsilon_b^{b-\mathrm{sig}}$, as a function of jet energy `Ejet` . The b-signal region is defined as $\beta > 0.9$. In the plot the efficiencies are shown for MC Truth in blue, MC TTP in red, MC Truth TTP in green, and Data TTP in orange. The efficiencies (the errorbars) can be read off on the left y-axis and the counts (histograms) on the right y-axis. The abbreviation TTP is short for "Tag, Tag, Probe" where two jets in a event are used as tags and the probe is then used for further analysis. Notice how both MC TTP and Data TTP follow each other closely.

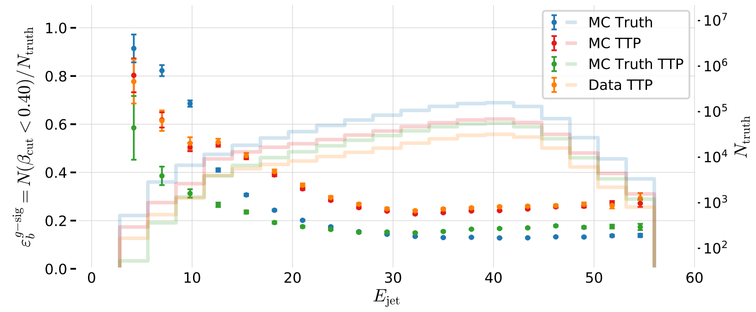


Figure 5.22: Efficiency of the b-tags for b-jets in the g-signal region for 3-jet events, $\varepsilon_b^{g-\text{sig}}$, as a function of jet energy `Ejet`. The g-signal region is defined as $\beta < 0.4$. In the plot the efficiencies are shown for MC Truth in blue, MC TTP in red, MC Truth TTP in green, and Data TTP in orange. The efficiencies (the errorbars) can be read off on the left y-axis and the counts (histograms) on the right y-axis. The abbreviation TTP is short for "Tag, Tag, Probe" where two jets in a event are used as tags and the probe is then used for further analysis. Notice how both MC TTP and Data TTP follow each other closely.

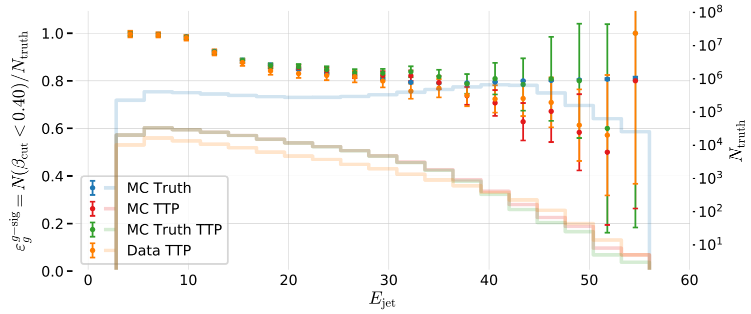


Figure 5.23: Efficiency of the b-tags for g-jets in the g-signal region for 3-jet events, $\varepsilon_g^{g-\text{sig}}$, as a function of jet energy `Ejet`. The g-signal region is defined as $\beta < 0.4$. In the plot the efficiencies are shown for MC Truth in blue, MC TTP in red, MC Truth TTP in green, and Data TTP in orange. The efficiencies (the errorbars) can be read off on the left y-axis and the counts (histograms) on the right y-axis. The abbreviation TTP is short for "Tag, Tag, Probe" where two jets in a event are used as tags and the probe is then used for further analysis. Notice how both MC TTP and Data TTP follow each other closely.

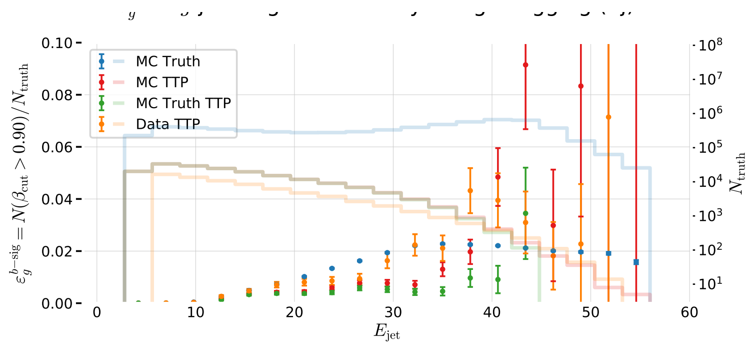


Figure 5.24: Efficiency of the b-tags for g-jets in the b-signal region for 3-jet events, $\varepsilon_g^{b-\text{sig}}$, as a function of jet energy `Ejet`. The b-signal region is defined as $\beta > 0.9$. In the plot the efficiencies are shown for MC Truth in blue, MC TTP in red, MC Truth TTP in green, and Data TTP in orange. The efficiencies (the errorbars) can be read off on the left y-axis and the counts (histograms) on the right y-axis. The abbreviation TTP is short for "Tag, Tag, Probe" where two jets in a event are used as tags and the probe is then used for further analysis.

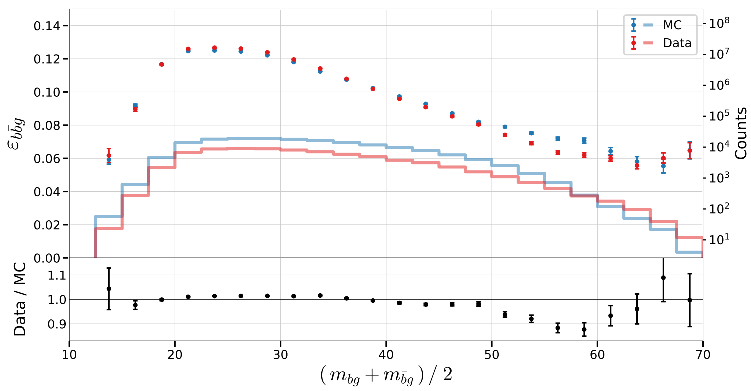


Figure 5.25: Proxy efficiency of the g-tags for $b\bar{b}g$ 3-jet events as a function of the mean of the two invariant masses m_{bg} and $m_{\bar{b}g}$. The proxy efficiency $\varepsilon_{b\bar{b}g}$ is measured by finding $b\bar{b}g$-events where $\beta_b > 0.9$, $\beta_{\bar{b}} > 0.9$, and $\beta_g < 0.4$. and then calculating $\varepsilon_{b\bar{b}g} = \varepsilon_b^{b-\text{sig}} \cdot \varepsilon_{\bar{b}}^{b-\text{sig}} \cdot \varepsilon_g^{g-\text{sig}}$. In the top plot $\varepsilon_{b\bar{b}g}$ is shown for MC in blue and Data in red where the counts in each bin can be read on right y-axis. In the bottom plot the ratio between Data and MC is shown.

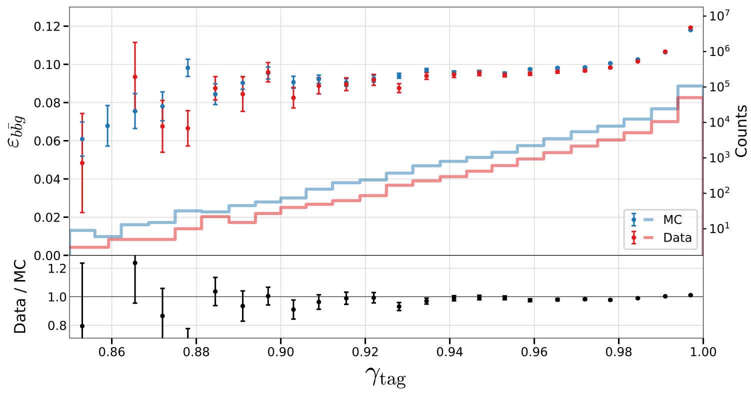


Figure 5.26: Proxy efficiency of the g-tags for $b\bar{b}g$ 3-jet events as a function of the event's g-tag. The proxy efficiency $\varepsilon_{b\bar{b}g}$ is measured by finding $b\bar{b}g$-events where $\beta_b > 0.9$, $\beta_{\bar{b}} > 0.9$, and $\beta_g < 0.4$. and then calculating $\varepsilon_{b\bar{b}g} = \varepsilon_b^{b-\text{sig}} \cdot \varepsilon_{\bar{b}}^{b-\text{sig}} \cdot \varepsilon_g^{g-\text{sig}}$. In the top plot $\varepsilon_{b\bar{b}g}$ is shown for MC in blue and Data in red where the counts in each bin can be read on right y-axis. In the bottom plot the ratio between Data and MC is shown.

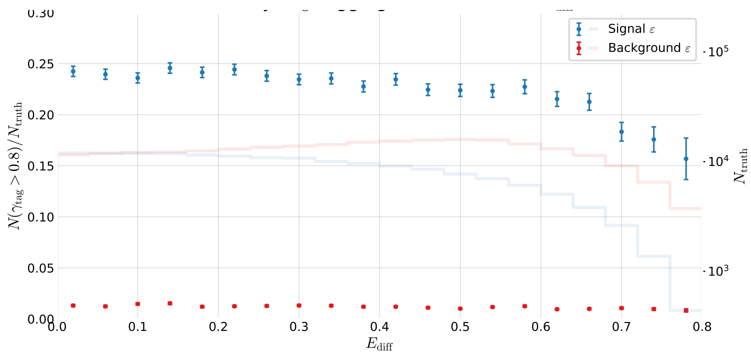


Figure 5.27: Efficiency of the g-tags for 4-jet events as a function of normalized gluon gluon jet energy difference in Monte Carlo. The efficiency is measured as the number of events with a g-tag higher than 0.8 ($\gamma > 0.8$) out of the total number and the normalized gluon gluon jet energy difference A is $A = \frac{E_{g\max} - E_{g\min}}{E_{g\max} + E_{g\min}}$ where $E_{g\max}$ ($E_{g\min}$) refers to the energy of the gluon with the highest (lowest) energy. The efficiency is plotted for signal events according to MC Truth in blue and background events according to MC Truth in red.

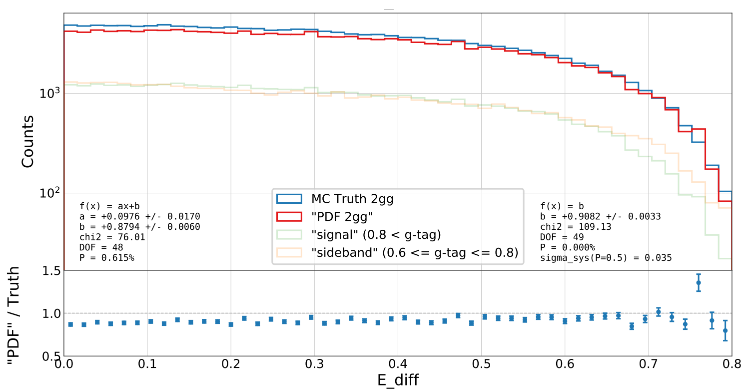


Figure 5.28: Closure plot between MC Truth and the corrected g-tagging model in 4-jet events for the normalized gluon gluon jet energy difference. The corrected g-taggingg model is described in further detail in section XXX **TODO!**. In the top part of the plot the MC Truth is shown in blue, the corrected g-tagging model `"PDF 2gg"` in red, the g-signal distribution in semi-transparent green and the g-sideband distribution in semi-transparent orange. In the bottom part of the plot the ratio between MC Truth and the output of the corrected g-tagging model is shown. The normalized gluon gluon jet energy difference A is $A = \frac{E_{g\max} - E_{g\min}}{E_{g\max} + E_{g\min}}$ where $E_{g\max}$ ($E_{g\min}$) refers to the energy of the gluon with the highest (lowest) energy.

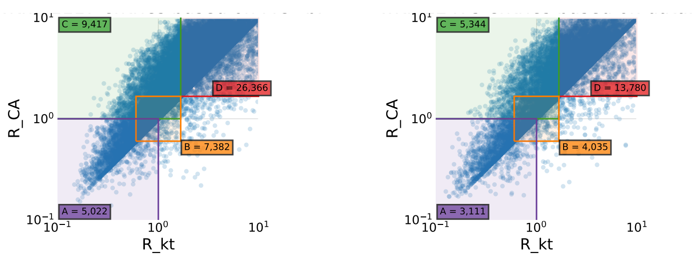


Figure 5.29: R kt CA overview XXX **TODO!**

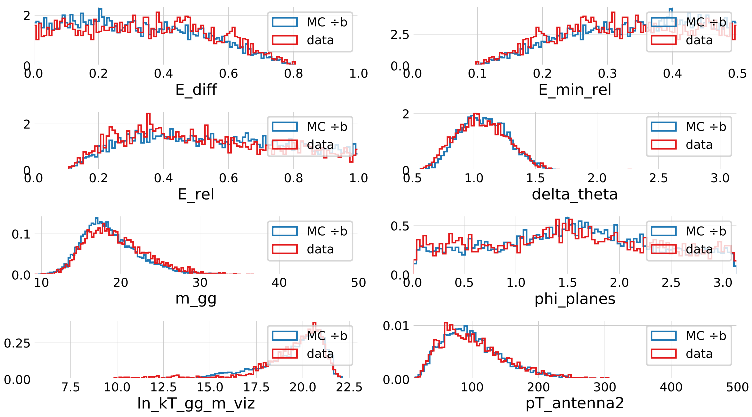


Figure 5.30: R kt CA cut region A XXX **TODO!**

B. *Quarks vs. Gluons Appendix*

	b	c	uds	g	non-q-matched
2	37.2 %	12.9 %	29.1 %	0.0 %	20.7 %
3	22.6 %	8.9 %	19.7 %	31.2 %	17.5 %
4	14.6 %	7.0 %	15.0 %	45.1 %	18.3 %
5	10.0 %	5.7 %	12.2 %	52.5 %	19.6 %
6	7.1 %	4.4 %	8.8 %	54.4 %	25.2 %

Table B.1: Number of different types of jets for MC and MCb written in relative numbers such that each row sum to 100 %. See also Table 5.3.

figures/quarks/viz_UMAP_test_0.5_input2b_njet=4_algorithm=UMAP.pdf

Figure B.1: Grid search of the two parameters `n_neighbors` and `min_dist` for the UMAP algorithm run on 4-jet events. For an explanation of these, see section 5.2.

figures/quarks/viz_TSNE_MULTI_test_0.5_input2b_njet=4_algorithm=tsne_multi.pdf

Figure B.2: Visualization of the t-SNE algorithm as a function of the `perplexity` parameters for 4-jet events.

Hyperparameter	Range
`subsample`	$\mathcal{U}(0.4, 1)$
`colsample_bytree`	$\mathcal{U}_{\text{trunc}}(0.4, 1, 2)$
`max_depth`	$\mathcal{U}_{\text{int}}(1, 20)$
`min_child_weight`	$\mathcal{U}_{\text{int}}(0, 10)$

Table B.2: Probability Density Functions for the random search hyperparameter optimization process for the XGBoost model.

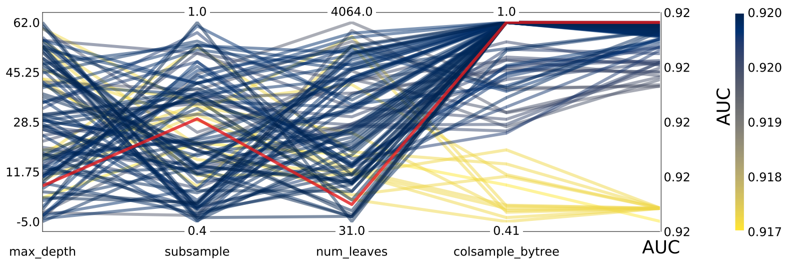


Figure B.3: Hyperparameter optimization results of b-tagging for 3-jet events. The results are shown as parallel coordinates with each hyperparameter along the x-axis and the value of that parameter on the y-axis. Each line is an event in the 4-dimensional space colored according to the performance of that hyperparameter as measured by AUC from highest AUC in dark blue to lowest AUC in yellow. The single best hyperparameter is shown in red.

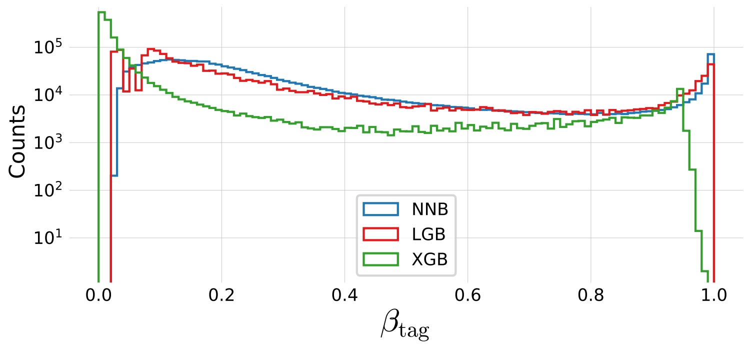


Figure B.4: Histogram of b-tag scores β_{tag} in 3-jet events for NNB (the neural network pre-trained by ALEPH, also called `nnbjet`) in blue, LGB in red, and XGB in green.

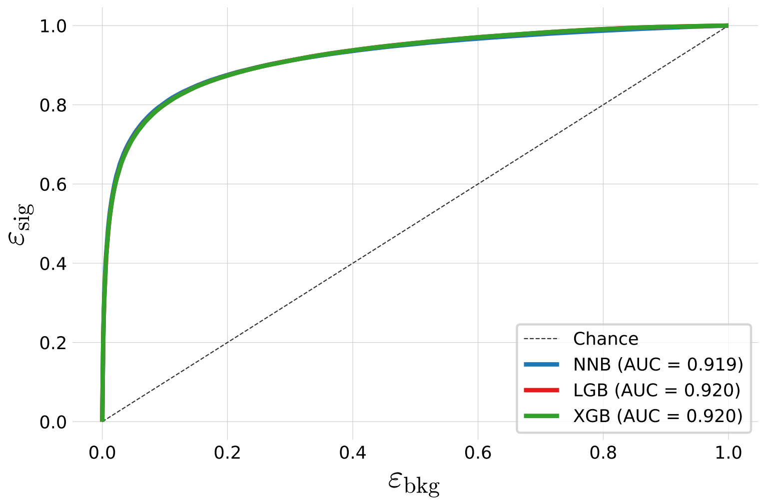


Figure B.5: ROC curve of the three b-tag models in 3-jet events for NNB (the pre-trained neural network trained by ALEPH, also called `nnbjet`) in blue, LGB in red, and XGB in green. In the legend the area under curve (AUC) is also shown. Notice that the LGB and XGB models share performance and it is thus due to overplotting that only the green line for XGB can be seen. In the machine learning community the background efficiency $\varepsilon_{\mathrm{bkg}}$ is sometimes know as the false positive rate (FPR) and the signal efficiency $\varepsilon_{\mathrm{sig}}$ as the true positive rate (TPR).

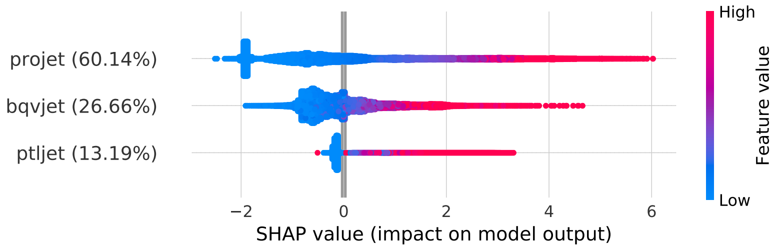


Figure B.6: Global feature importances for the LGB b-tagging algorithm on 3-jet events. The normalized feature importance is shown in the parenthesis and the each dot is an observation showing the dependance between the SHAP value and the feature's value.

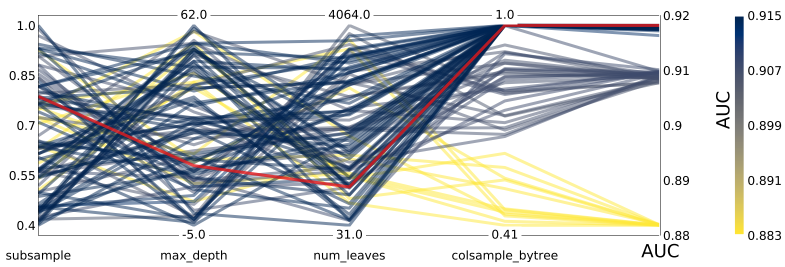


Figure B.7: Hyperparameter optimization results of g-tagging for 3-jet events for energy ordered jets.

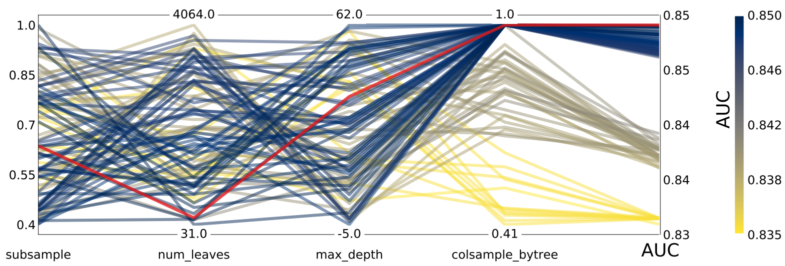


Figure B.8: Hyperparameter optimization results of g-tagging for 3-jet events for (row) shuffled jets.

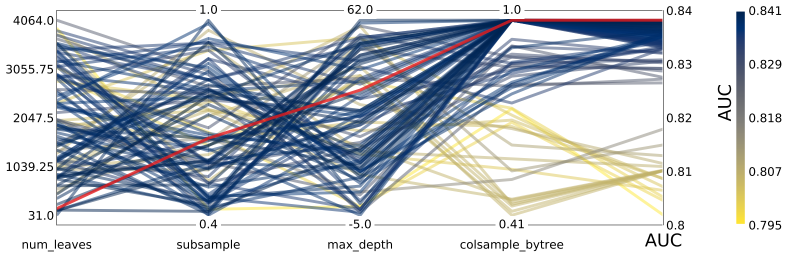


Figure B.9: Hyperparameter optimization results of g-tagging for 4-jet events for energy ordered jets.

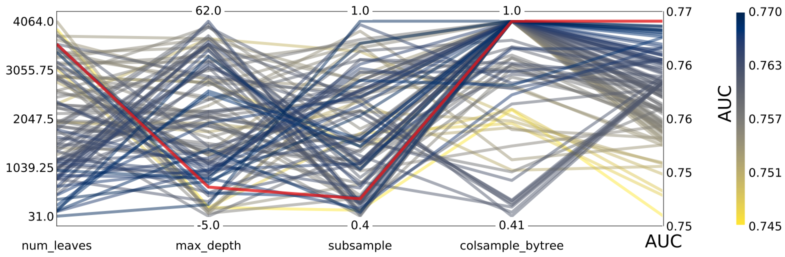


Figure B.10: Hyperparameter optimization results of g-tagging for 4-jet events for (row) shuffled jets.

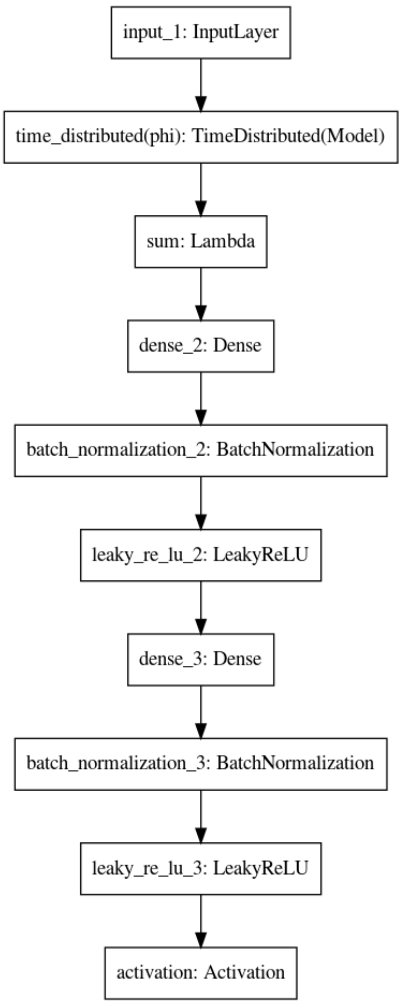


Figure B.11: Architecture of the PermNet neural network.

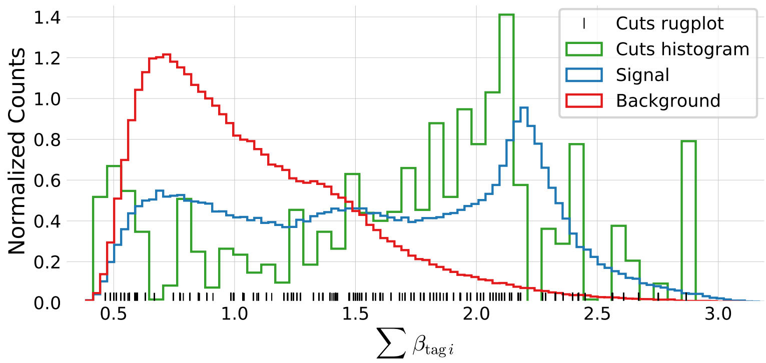


Figure B.12: Histogram of the distribution of signal in blue and background in red for the 1-dimensional sum of b-tags for 4-jet events. A histogram of the cut values from the LGB model trained on this data is shown in green together with a rug plot of the cut values in black. Notice how most of the cuts match up with the signal peak at around a $\sum \beta_i \sim 2.1$.

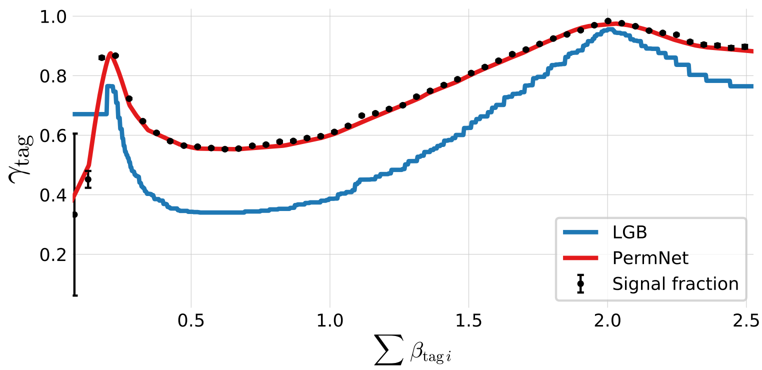


Figure B.13: Plot of the (1D) g-tag scores for 3-jet events as a function of $\sum \beta_i$ for the LGB model in blue and the PermNet model in red. The signal fraction (based on the signal and background histograms in Figure B.14) is plotted as black error bars where the size of the error bars is based on the propagated uncertainties of the signal and background histogram assuming Poissonian statistics.

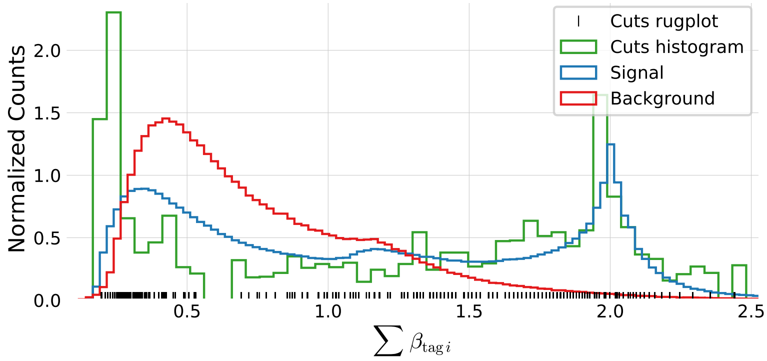


Figure B.14: Histogram of the distribution of signal in blue and background in red for the 1-dimensional sum of b-tags for 3-jet events. A histogram of the cut values from the LGB model trained on this data is shown in green together with a rug plot of the cut values in black. Notice how most of the cuts match up with the signal peak at around a $\sum \beta_i \sim 2.1$.

List of Figures

List of Tables

Bibliography

[1] Advanced Topics in Machine Learning (ATML). URL https://kurser.ku.dk/course/ndak15014u.

[2] Allstate Claims Severity - Fair Loss. URL https://kaggle.com/c/allstate-claims-severity.

[3] Dmlc/xgboost. URL https://github.com/dmlc/xgboost.

[4] HEP meets ML award | The Higgs Machine Learning Challenge. URL https://higgsml.lal.in2p3.fr/prizes-and-award/award/.

[5] The Large Electron-Positron Collider | CERN. URL https://home.cern/science/accelerators/large-electron-positron-collider.

[6] Microsoft/LightGBM. URL https://github.com/microsoft/LightGBM/blob/b3975555d7023fd05f8e56326905fe7b185109de/src/treelearner/serial_tree_learner.cpp#L282.

[7] Scikit-hep/uproot. URL https://github.com/scikit-hep/uproot.

[8] Datashader: Revealing the Structure of Genuinely Big Data. URL https://github.com/holoviz/datashader.

[9] O. . Www.OIS.dk - Din genvej til ejendomsdata. URL https://www.ois.dk/.

[10] M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems. URL http://tensorflow.org/.

[11] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin. *Learning From Data*. AMLBook. ISBN 978-1-60049-006-4.

[12] D. Albanese, S. Riccadonna, C. Donati, and P. Franceschi. A practical tool for maximal information coefficient analysis. 7. ISSN 2047-217X. doi: 10.1093/gigascience/giy032. URL https://doi.org/10.1093/gigascience/giy032.

[13] E. Anderson. The Species Problem in Iris. 23(3):457–509. ISSN 00266493. doi: 10.2307/2394164. URL www.jstor.org/stable/2394164.

[14] B. Andersson, G. Gustafson, G. Ingelman, and T. Sjöstrand. Parton fragmentation and string dynamics. 97(2):31–145. ISSN 0370-1573. doi: 10.1016/0370-1573(83)90080-7. URL http://www.sciencedirect.com/science/article/pii/0370157383900807.

[15] S. R. Armstrong. A Search for the standard model Higgs boson in four jet final states at center-of-mass energies near 183-GeV with the ALEPH detector at LEP. URL http://wwwlib.umi.com/dissertations/fullcit?p9910371.

[16] M. Awad and R. Khanna. Support Vector Regression. In M. Awad and R. Khanna, editors, *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, pages 67–80. Apress. ISBN 978-1-4302-5990-9. doi: 10.1007/978-1-4302-5990-9_4. URL https://doi.org/10.1007/978-1-4302-5990-9_4.

[17] R. J. Barlow. *Statistics: A Guide to the Use of Statistical Methods in the Physical Sciences (Manchester Physics Series)*. WileyBlackwell, reprint edition. ISBN 0-471-92295-1. URL http://www.amazon.co.uk/Statistics-Statistical-Physical-Sciences-Manchester/dp/0471922951%3FSubscriptionId%3D13CT5CVB80YFWJEPWS02%26tag%3Dws%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0471922951.

[18] J. T. Barron. A General and Adaptive Robust Loss Function. URL http://arxiv.org/abs/1701.03077.

[19] W. Bartel et al. Experimental study of jets in electron-positron annihilation. 101(1):129–134. ISSN 0370-2693. doi: 10.1016/0370-2693(81)90505-0. URL http://www.sciencedirect.com/science/article/pii/0370269381905050.

[20] E. Becht, C.-A. Dutertre, I. W. H. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell. Evaluation of UMAP as an alternative to t-SNE for single-cell data. page 298430, . doi: 10.1101/298430. URL https://www.biorxiv.org/content/10.1101/298430v1.

[21] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I. W. H. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell. Dimensionality reduction for visualizing single-cell data using UMAP. 37 (1):38–44, . ISSN 1546-1696. doi: 10.1038/nbt.4314. URL https://www.nature.com/articles/nbt.4314.

[22] J. Bergstra and Y. Bengio. Random Search for Hyper-parameter Optimization. 13:281–305. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=2188385.2188395.

[23] C. Bierlich. Rope hadronization, geometry and particle production in pp and pA Collisions. URL https://lup.lub.lu.se/search/ws/files/18474576/thesis.pdf.

[24] Bolighed. Bolighed - usikkerhed i data-vurderingen. URL https://bolighed.dk/om-bolighed/spoergsmaal-og-svar/#boligvaerdi.

[25] L. Breiman. Random Forests. 45(1):5–32. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

[26] E. Brochu, V. M. Cora, and N. de Freitas. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. URL http://arxiv.org/abs/1012.2599.

[27] R. Brun and F. Rademakers. ROOT — An object oriented data analysis framework. 389(1):81–86. ISSN 0168-9002. doi: 10.1016/S0168-9002(97)00048-X. URL http://www.sciencedirect.com/science/article/pii/S016890029700048X.

[28] A. Buckley, J. Butterworth, S. Gieseke, D. Grellscheid, S. Hoche, H. Hoeth, F. Krauss, L. Lonnblad, E. Nurse, P. Richardson, S. Schumann, M. H. Seymour, T. Sjostrand, P. Skands, and B. Webber. General-purpose event generators for LHC physics. 504(5):145–233. ISSN 03701573. doi: 10.1016/j.physrep.2011.03.005. URL http://arxiv.org/abs/1101.2599.

[29] C. Burgard. Standard model of physics | TikZ example. URL http://www.texample.net/tikz/examples/model-physics/.

[30] D. Buskulic et al. An investigation of Bdo and Bso oscillation. 322(4):441–458. ISSN 0370-2693. doi: 10.1016/0370-2693(94)91177-0. URL http://www.sciencedirect.com/science/article/pii/0370269394911770.

[31] T. Chen and C. Guestrin. XGBoost: A Scalable Tree Boosting System. pages 785–794. doi: 10.1145/2939672.2939785. URL http://arxiv.org/abs/1603.02754.

[32] T. A. Collaboration. Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. 716(1):1–29. ISSN 03702693. doi: 10.1016/j.physletb.2012.08.020. URL http://arxiv.org/abs/1207.7214.

[33] A. Diaz-Papkovich, L. Anderson-Trocmé, C. Ben-Eghan, and S. Gravel. UMAP reveals cryptic population structure and phenotype heterogeneity in large genomic cohorts. 15(11). ISSN 1553-7390. doi: 10.1371/journal.pgen.1008432. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6853336/.

[34] S. D. DST. Price Index (EJ14) - Statistics Denmark. URL https://www.dst.dk/en/Statistik/emner/priser-og-forbrug/ejendomme.

[35] D. et al. Buskulic. A precise measurement of hadrons. 313(3): 535–548. ISSN 0370-2693. doi: 10.1016/0370-2693(93)90028-G. URL http://www.sciencedirect.com/science/article/pii/037026939390028G.

[36] F. Faye. Frederik Faye / deepcalo. URL https://gitlab.com/ffaye/deepcalo.

[37] R. A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. 7(2):179–188. ISSN 2050-1439. doi: 10.1111/j.1469-1809.1936.tb02137.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x.

[38] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In P. Vitányi, editor, *Computational Learning Theory*, pages 23–37. Springer Berlin Heidelberg. ISBN 978-3-540-49195-8. AdaBoost.

[39] S. L. Glashow. Partial-symmetries of weak interactions. 22(4): 579–588. ISSN 0029-5582. doi: 10.1016/0029-5582(61)90469-2. URL http://www.sciencedirect.com/science/article/pii/0029558261904692.

[40] N. Guttenberg, N. Virgo, O. Witkowski, H. Aoki, and R. Kanai. Permutation-equivariant neural networks applied to dynamics prediction. URL http://arxiv.org/abs/1612.04530.

[41] A. E. Harvey and S. Peters. Estimation Procedures for Structural Time Series Models. doi: 10.1002/for.3980090203.

[42] T. Hastie and R. Tibshirani. Generalized Additive Models: Some Applications. 82(398):371–386. ISSN 01621459. doi: 10.2307/2289439. URL www.jstor.org/stable/2289439.

[43] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer-Verlag, 2 edition. ISBN 978-0-387-84857-0. URL //www.springer.com/la/book/9780387848570.

[44] K. Hornik. Approximation capabilities of multilayer feedforward networks. 4(2):251–257. ISSN 0893-6080. doi: 10.1016/0893-6080(91)90009-T. URL http://www.sciencedirect.com/science/article/pii/089360809190009T.

[45] P. Huber and E. Ronchetti. *Robust Statistics*. Wiley Series in Probability and Statistics. Wiley. ISBN 978-1-118-21033-8. URL https://books.google.dk/books?id=j1OhquR_j88C.

[46] S. Hviid, Juul. Working Paper: A regional model of the Danish housing market. URL http://www.nationalbanken.dk/en/publications/Pages/2017/11/Working-Paper-A-regional-model-of-the-Danish-housing-market.aspx.

[47] F. James and M. Roos. Minuit – a system for function minimization and analysis of the parameter errors and correlations. 10:343–367. doi: 10.1016/0010-4655(75)90039-9.

[48] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc. URL http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf.

[49] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. URL http://arxiv.org/abs/1412.6980.

[50] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. 49(4): 764–766. ISSN 0022-1031. doi: 10.1016/j.jesp.2013.03.013. URL http://www.sciencedirect.com/science/article/pii/S0022103113000668.

[51] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 4768–4777. Curran Associates Inc. ISBN 978-1-5108-6096-4. URL http://dl.acm.org/citation.cfm?id=3295222.3295230.

[52] S. M. Lundberg, G. G. Erion, and S.-I. Lee. Consistent Individualized Feature Attribution for Tree Ensembles - SHAP. . URL http://arxiv.org/abs/1802.03888.

[53] S. M. Lundberg, G. G. Erion, and S.-I. Lee. Consistent Individualized Feature Attribution for Tree Ensembles. . URL http://arxiv.org/abs/1802.03888.

[54] A. L. Maas. Rectifier nonlinearities improve neural network acoustic models.

[55] L. McInnes and J. Healy. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. URL http://arxiv.org/abs/1802.03426.

[56] I. Mulalic, H. Rasmussen, J. Rouwendal, and H. H. Woltmann. The Financial Crisis and Diverging House Prices: Evidence from the Copenhagen Metropolitan Area. ISSN 1556-5068. doi: 10.2139/ssrn.3041272. URL https://www.ssrn.com/abstract=3041272.

[57] Particle Data Group et al. Review of Particle Physics. 98 (3):030001. doi: 10.1103/PhysRevD.98.030001. URL https://link.aps.org/doi/10.1103/PhysRevD.98.030001.

[58] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. 12:2825–2830.

[59] E. Polley and M. van der Laan. Super Learner In Prediction. URL https://biostats.bepress.com/ucbbiostat/paper266.

[60] J. Proriol, J. Jousset, C. Guicheney, A. Falvard, P. Henrard, D. Pallin, P. Perret, and B. Brandl. TAGGING B QUARK EVENTS IN ALEPH WITH NEURAL NETWORKS (comparison of different methods : Neural Networks and Discriminant Analysis). page 27.

[61] A. Purcell. Go on a particle quest at the first CERN webfest. URL https://cds.cern.ch/record/1473657.

[62] S. Ravanbakhsh, J. Schneider, and B. Poczos. Deep Learning with Sets and Point Clouds. URL http://arxiv.org/abs/1611.04500.

[63] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti. Detecting Novel Associations in Large Data Sets. 334(6062):1518–1524. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1205438. URL http://science.sciencemag.org/content/334/6062/1518.

[64] P. J. Rousseeuw and C. Croux. Alternatives to the Median Absolute Deviation. 88(424):1273–1283. ISSN 0162-1459. doi: 10.1080/01621459.1993.10476408. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1993.10476408.

[65] A. Salam. Weak and electromagnetic interactions. In *Selected Papers of Abdus Salam*, volume Volume 5 of *World Scientific Series in 20th Century Physics*, pages 244–254. WORLD SCIENTIFIC. ISBN 978-981-02-1662-7. doi: 10.1142/9789812795915_0034. URL https://www.worldscientific.com/doi/abs/10.1142/9789812795915_0034.

[66] L. Scodellaro. B tagging in ATLAS and CMS. URL http://arxiv.org/abs/1709.01290.

[67] L. Shapley. A value for n-person games. In *The Shapley Value*, volume 28 of *Annals of Math Studies*, pages 307–317. doi: 10.1017/CBO9780511528446.003.

[68] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and

P. Z. Skands. An Introduction to PYTHIA 8.2. 191:159–177. ISSN 00104655. doi: 10.1016/j.cpc.2015.01.024. URL http://arxiv.org/abs/1410.3012.

[69] S. J. Taylor and B. Letham. Forecasting at scale. doi: 10.7287/peerj.preprints.3190v2. URL https://peerj.com/preprints/3190.

[70] i. team. Iminuit – A python interface to minuit. URL https://github.com/scikit-hep/iminuit.

[71] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. 58(1):267–288. ISSN 0035-9246. URL www.jstor.org/stable/2346178.

[72] A. Tikhonov. *On the Stability of Inverse Problems*, volume vol. 39 of *Doklady Akademii Nauk SSSR*.

[73] S. Toghi Eshghi, A. Au-Yeung, C. Takahashi, C. R. Bolen, M. N. Nyachienga, S. P. Lear, C. Green, W. R. Mathews, and W. E. O'Gorman. Quantitative Comparison of Conventional and t-SNE-guided Gating Analyses. 10. ISSN 1664-3224. doi: 10.3389/fimmu.2019.01194. URL https://www.frontiersin.org/articles/10.3389/fimmu.2019.01194/full.

[74] d. L. M. J. van, E. C. Polley, and A. E. Hubbard. Super Learner. 6(1). ISSN 1544-6115. doi: 10.2202/1544-6115.1309. URL https://www.degruyter.com/view/j/sagmb.2007.6.issue-1/sagmb.2007.6.1.1309/sagmb.2007.6.1.1309.xml.

[75] L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. 9:2579–2605. ISSN ISSN 1533-7928. URL http://www.jmlr.org/papers/v9/vandermaaten08a.html.

[76] F. van Veen. The Neural Network Zoo. URL http://www.asimovinstitute.org/neural-network-zoo/.

[77] V. Vapnik. Principles of Risk Minimization for Learning Theory. In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, pages 831–838. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-222-9. URL http://dl.acm.org/citation.cfm?id=2986916.2987018.

[78] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors. SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python. page arXiv:1907.10121. URL https://ui.adsabs.harvard.edu/abs/2019arXiv190710121V/abstract.

[79] I. Wallach and R. Lilien. The protein–small-molecule database, a non-redundant structural resource for the analysis of protein-ligand binding. 25(5):615–620. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp035. URL https://academic.oup.com/bioinformatics/article/25/5/615/183421.

[80] S. Weinberg. A Model of Leptons. 19(21):1264–1266. doi: 10.1103/PhysRevLett.19.1264. URL https://link.aps.org/doi/10.1103/PhysRevLett.19.1264.

[81] H. Wickham. Tidy data. 59(10):1–23. ISSN 1548-7660. doi: 10.18637/jss.v059.i10. URL https://www.jstatsoft.org/v059/i10.

[82] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola. Deep Sets. URL http://arxiv.org/abs/1703.06114.

Index