UNIVERSITY OF CALIFORNIA,
IRVINE

Decoding Black Box Models to Find New Physics at the LHC

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Physics

by

Taylor James Faucett

Dissertation Committee:
Professor Daniel Whiteson, Chair
Professor Andrew Lankford
Professor David Kirkby

2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Page

ACKNOWLEDGMENTS

I would like to give a special thanks to my advisor, Daniel Whiteson, for always providing a supportive, engaging and productive environment for my work. In our research group, he is the black box that helps guide us to better high-level decision making. Additional thanks are owed to co-advisors and colleagues who I have had the pleasure of working with, including: Pierre Baldi (UCI), Jesse Thaler (MIT), Shih-Chieh Hsu (U of Washington), Julian Donini (Université Clermont-Auvergne), and Kyle Cranmer (NYU).

I would like to give an additional thanks to my parents, Larry and Kaye, who have provided constant support and encouragement throughout my long academic history.

Lastly, thanks for funding go to the U.S. Department of Energy Office of Science under Grant No. DE-SC0009920, the National Science Foundation under Award No. 1633631 and the Chateaubriand Fellowship of the Office for Science & Technology of the Embassy of France in the United States.

Chapter 3 of this dissertation is a reproduction of material as it appears in “Mapping Machine-Learned Physics into a Human-Readable Space”[1]. This work was directed and supervised by Daniel Whiteson and was co-written with Jesse Thaler from the Center for Theoretical Physics at MIT.

Chapter 4 of this dissertation is a reproduction of material as it appears in “Learning to identify electrons”[2]. This work was produced in collaboration with Daniel Whiteson, Jessica N. Howard, and Tony Tong from the Department of Physics & Astronomy at UC Irvine along with Pierre Baldi and Julian Collado from the Department of Information & Computer Sciences at UC Irvine.

Chapter 5 of this dissertation is a reproduction of material as it appears in “Learning to isolate muons”[3]. This work was produced in collaboration with Daniel Whiteson, Kevin Bauer, and Edmund Witkowski from the Department of Physics & Astronomy at UC Irvine along with Pierre Baldi and Julian Collado from the Department of Information & Computer Sciences at UC Irvine.

Chapter 6 of this dissertation is based on work supervised and directed by Daniel Whiteson and in collaboration with Shih-Chieh Hsu from the department of physics at the University of Washington.

Chapter 7 of this dissertation is a reproduction of material as it appears in “Parameterized Neural Networks for High-Energy Physics”[4]. This work was produced in collaboration with Pierre Baldi and Peter Sadowski from the department of Information & Computer Sciences at UC Irvine, Daniel Whiteson from the department of Physics & Astronomy at UC Irvine and Kyle Cranmer from the department of Physics at NYU.

VITA

Taylor James Faucett

EDUCATION

Doctor of Philosophy in Physics **2021**
University of California, Irvine *Irvine, CA*

Masters of Science in Physics **2015**
University of Hawaii, Manoa *Honolulu, HI*

Bachelor of Science in Physics **2009**
Westminster College *Salt Lake City, UT*

RESEARCH EXPERIENCE

Graduate Research Assistant **2015–2021**
University of California, Irvine *Irvine, California*

Graduate Research Assistant **2013–2015**
University of Hawaii, Manoa *Honolulu, HI*

TEACHING EXPERIENCE

Teaching Assistant **2015–2017**
University of California, Irvine *Irvine, CA*

Teaching Assistant **2011–2015**
University of Hawaii, Manoa *Honolulu, HI*

REFEREED JOURNAL PUBLICATIONS

Parameterized Machine Learning for High-Energy Physics **May 2016**

European Physical Journal C

Mapping machine-learned physics into a human-readable space **Feb 2021**

Physical Review D

Learning to identify electrons **Jun 2021**

Physical Review D

Learning to isolate muons **Oct 2021**

Journal of High Energy Physics

SOFTWARE

Average Decision Ordering `github.com/taylorFaucett/average-decision-ordering`

Python based implementation of a novel similarity metric that measures the relative similarity between the predictions of two machine learning models on the same input data.

ABSTRACT OF THE DISSERTATION

Decoding Black Box Models to Find New Physics at the LHC

By

Taylor James Faucett

Doctor of Philosophy in Physics

University of California, Irvine, 2021

Professor Daniel Whiteson, Chair

This work presents techniques for addressing the black box problem for deep learning in high-energy physics applications at the LHC. In an initial group of studies, a method is presented for translating a black box classifier using high-dimensional detector data into a minimal set of simple physics motivated features with equivalent classification performance. The strategy is first applied to a benchmark discrimination task for jets from a boosted W boson decay. The algorithm is then used on two active areas of standard model research at the LHC: electron identification and prompt muon isolation. Finally, the technique is applied to a beyond the standard model study of semi-visible jets produced via a theoretical dark quark hadronization process.

A second technique is presented, providing a method for the explicit embedding of physics parameters alongside measured features in a deep learning model. This architecture yields a parameterized classifier that can smoothly interpolate between physical features. The result is a simpler and more powerful machine learning classifier that can seamlessly incorporate expert physics knowledge into its learned solution. The parameterized network is applied to a benchmark classification task for $t\bar{t}$ decays.

Chapter 1

Introduction

The research motivation for physicists working at the Large Hadron Collider (LHC) can be broken into two main categories: probing of the Standard Model (SM) with ever increasing precision and the search for new particles in the Beyond the Standard Model (BSM) regime. Each of these goals are addressed with detector measured data and the identification of rare signatures in the presence of immense background.

To isolate and study such attenuated signals, large quantities of data are produced through high-energy proton collisions, measurements are collected by detectors (e.g. ATLAS and CMS) and state of the art data analysis techniques are leveraged to isolate signals contained in the resulting data. With every new discovery, the remaining unknown physics retreats farther into more obscure areas of study and experiments are forced to ramp up the accelerator luminosity. Upgrades to the LHC (run 3) are expected to increase the accelerator's performance by approximately 1.5 times its current instantaneous luminosity. In an attempt to address the increase in event sizes and data volume, machine learning (ML) methods have been adopted with the promise of improved accuracy in analysis, rapid execution times, and a more compact computational footprint. These benefits are well attested to in the literature and are used in all aspects of LHC operation and analysis, including: Event triggering [5–8], reconstruction and calibration [7, 9, 10], object identification [11–14], event selection [15–17], and simulation [18–20].

The adoption of ML techniques into HEP doesn't come without serious obstacles. While sophisticated architectures used in the literature demonstrate improvements in performance and speed when compared to standard analysis tools, the advantages often come at the expense of model simplicity and intelligibility. These types of opaque problem solving strategies are referred to as “black box” models and the loss of interpretability as the “black box problem”. Unlike some simpler analytical techniques, a black box model attempts to solve a problem by abstracting it to a high-dimensional space and fine-tuning the internal parameters of that abstraction to give optimal predictions. This kind of data representation can not be interpreted directly. Rather, one can think of a black box model as a computer generated function that maps inputs to predictions (often very accurately) using an unknown internal logic. Such a construction leads to complications in data selection, cleaning and pre-processing, model design, model optimization and tuning, and ultimately interpretation and understanding of the final data-driven solution.

Although transparent learning methods exist to combat these problems, they frequently perform worse than black box methods on important learning tasks. As a result, physicists often find themselves in the position of bargaining between the use of poorly performing interpretable models and better performing black box networks. The inevitable compromise between these two approaches is occasionally treated as an axiom akin to Heisenberg's uncertainty principle; One may not maximize, simultaneously, the performance and intelligibility of a machine learned model.

In this work, that notion is challenged through techniques for untangling and interacting with the learned solutions of black box networks. Rather than resign oneself to selecting between weak models and unintelligible ones, it becomes possible to implement a learning strategy that combines the benefits of both while minimizing their weaknesses.

Chapter 2

Deep Learning in High-Energy Physics

Studies performed at the Large Hadron Collider have the potential to address many of the fundamental questions posed in modern physics. Questions relating to the the fundamental composition of matter, basic interactions and forces between matter and a unified theory combining them. In the last decade, the LHC has produced high-precision measurements of many Standard Model (SM) particles and higher order corrections, discovered nearly 5 dozen new hadrons and peaked (both metaphorically and literally in the invariant mass distribution) with the confirmation of the Higgs boson. Future searches at the LHC will extend the experiment to more exotic physics, including studies of: Supersymmetry (SUSY), Dark Matter (DM), gravitons and a SM theory of gravity, extensions of the Higgs boson and quantum corrections to many SM processes.

With every new discovery, the search for undiscovered physics necessarily becomes more elusive. Studying infrequently created objects requires the production of large quantities of data in order to sift through a meaningful sample of events. Furthermore, a suite of sophisticated analysis techniques are required to isolate those signals in the presence of enormous quantities of background noise. The primary technique for generating large signal samples at the LHC involves the collision of protons at high energies and the careful measurement of the resulting particle shower it creates. Particles can then be assessed by precise detectors, recording their mass, energy and basic kinematics. This tracking data is processed for low-level analysis (i.e. object identification and reconstruction) as well as high-level analysis for the discovery of new particles.

2.1 Traditional & New Analysis Techniques

Discriminating new physics signals in the presence of background is often accomplished through the use of statistical hypothesis testing, classification or regression. Each of these tasks ultimately simplifies to the evaluation of the conditional probability ($p(\bar{x}|\bar{y})$) for observing rare events $\bar{x}$ given theory parameters $\bar{y}$. Evaluating this expression directly, especially in the context of high-dimensional/low-level (LL) feature spaces, is generally an intractable problem. One standard technique for addressing this complexity involves the computation of simple engineered variables from LL features based on theory considerations. This allows one to reduce the dimensionality of the problem into a high-level (HL) form. The HL features are then evaluated using a trusted algorithm, like a Boosted Decision Tree (BDT) as a part of the Toolkit for Multivariate Analysis(TMVA) [21] included in ROOT [22].

Dimensionality reduction techniques have the benefit that they can be tailored to the specific events or collection of events being studied. In the case of individual events, for example, reconstruction algorithms are typically used to generate calorimeter cluster and track data from the low-level detector measurements. Energy, momentum and particle identification information can then be reconstructed from the already reduced clusters and tracks. These simpler objects can be further processed according to the individual study being performed (e.g. jet clustering algorithms applied to reconstructed particle kinematics present in the hadronic calorimeter). This repeated dimensionality reduction process enables the application of simpler multivariate analysis methods on low-dimensional data and, historically, has produced better performance in analysis than studies on the original high-dimensional features. It is assumed, however, that some information must be lost in this reduction pipeline.

2.1.1 Moving from TMVA to Neural Networks

Given the expected information loss from dimensionality reduction techniques, methods for evaluating low-level information directly have been considered. The most ubiquitous of these tools in recent years is the deep neural network (DNN). In contrast to the multivariate techniques, a DNN attacks the problem by expanding the dimensionality of the problem into a higher-order abstract latent space. The motivation for this learning method is based on similar encoding schemes used in biological neural networks (i.e. the brain). Artificial Neural Networks (ANN) take in an input and process that information through neurons (nodes) to generate a high-dimensional representation of the solution. The biological connections between neurons in the brain are mimicked in an artificial neural network by weights applied between connections of nodes and those weights are systematically tuned to optimize performance as measured by an error/loss function.

Both shallow (single-layer) neural networks and deep neural networks are built from layers of neurons (described in Fig. 2.2), which function by multiplying inputs by their corresponding weight w_i. The weighted inputs are summed together and passed through an activation function that converts the combined features into an output value. The learning process is achieved through tuning of the networks neuron weights. Neurons are initialized with randomly assigned weight values and, for each epoch of training, the error/loss function is calculated. The error function measures the size of the discrepancy between the output vector

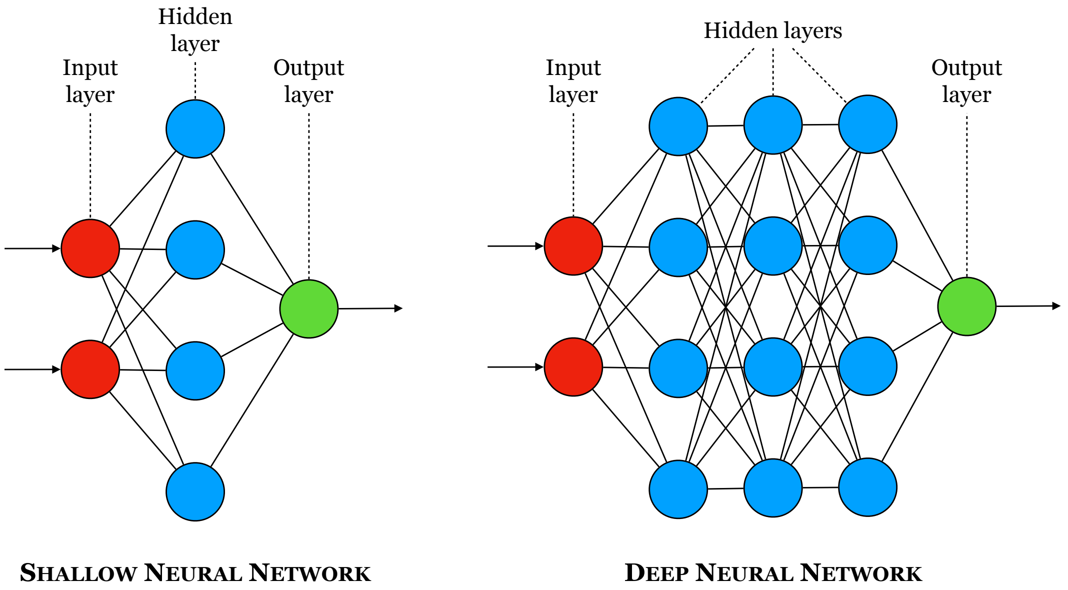


Figure 2.1: Architecture for a shallow single layer feed forward neural network (left) and a multi-layer deep neural network (right)

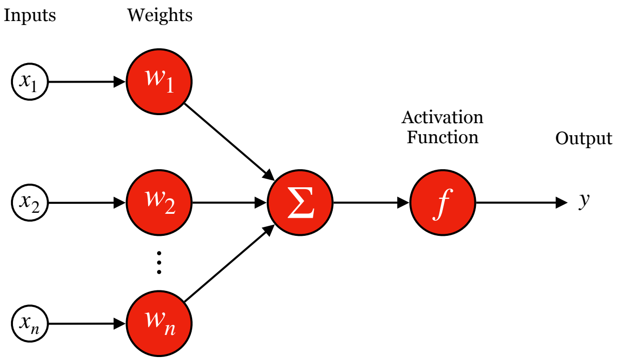


Figure 2.2: Diagram of a typical neural network node architecture.

and the truth while a loss function measures the impact of that discrepancy on the network. Depending on the optimization task (i.e. error function for regression vs loss function for classification), modifications to weights are determined through the minimization of the error/loss function. This is typically achieved using gradient descent to find its optimal local minimum value (i.e. weights are moved in the direction of the negative gradient of the error/loss function). Therefore, for each epoch of training, the weights are updated given an error/loss function E with,

$$w_{i+1} = w_i - \alpha \frac{\partial E}{\partial w_i} \tag{2.1}$$

where α, the learning rate, determines the size of steps taken when moving towards the minimum gradient. In the case of multi-layer deep neural networks, the calculation of gradients in individual layers is computationally costly. This is addressed through the use of backpropagation, in which the gradient calculation is pushed backwards through the network such that a gradient of the final hidden layer is computed first. Partial computations of the gradient from each layer are reused in the computation of the prior layer gradient. Error terms are computed for the initial layer in the backpropagation and the remaining errors are computed as a simple product sum of error terms from previous layers.

2.1.2 Moving to Higher Dimensions with Convolutional Networks

Although deep neural networks can be applied to the low-dimensional physics features discussed previously, the motivation for their use in this context is an aptitude for training on high-dimensional and non-linear problems. In the area of Computer Vision machine learning, Convolutional Neural Networks (CNN) have become a popular deep learning tactic for dealing with higher-order ($n-$dimensional) inputs. These are particularly powerful tools in

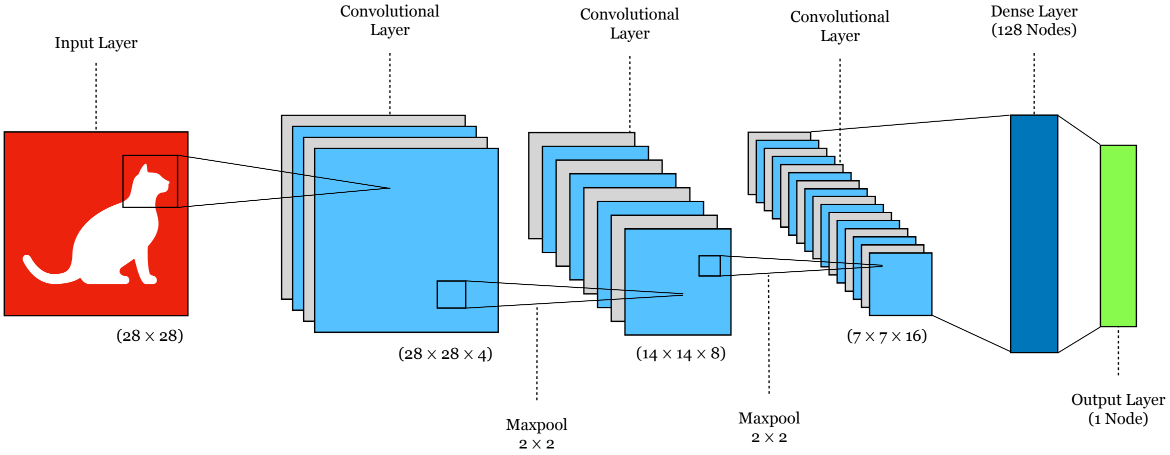


Figure 2.3: Example convolutional neural network architecture featuring a two-dimensional input image as a 28 × 28 array, convolutional layers interspersed with max-pooling layers, dense layer and a final output layer for binary prediction.

learning examples where spatial information is relevant in the data structure. This property lends itself nicely to clusters and tracks in a HEP context (examples are given in Sec. 2.1.3).

A typical CNN architecture for a binary classifier network is shown in Fig. 2.3. This example demonstrates an input for a 28 × 28 pixel image that is first passed to a convolutional layer. The convolutional layer down-samples the features to a smaller size through the application of a filter. An example filter demonstrating this down-sampling behavior is provided in Fig. 2.4. The filter is a grid of multiplicative factors that are applied to each pixel in the image. For a 3 × 3 filter applied to all inner pixels of the image, the original 5 × 5 matrix is reduced down to a 3 × 3 convolved feature. The size of the filter and the stride length (i.e. the step size between each shift of the filter) will ultimately decide the size reduction between image layers. The image is then further reduced by a pooling layer. Two commonly used pooling behaviors are the max-pooling and average-pooling method. As their names imply, the max and average pooling procedure involves systematically selecting subset filters of an image and isolating either the maximum or average value in the selected pixels. An example of this method is also given in Fig. 2.4. A series of convolutional layers and pooling layers

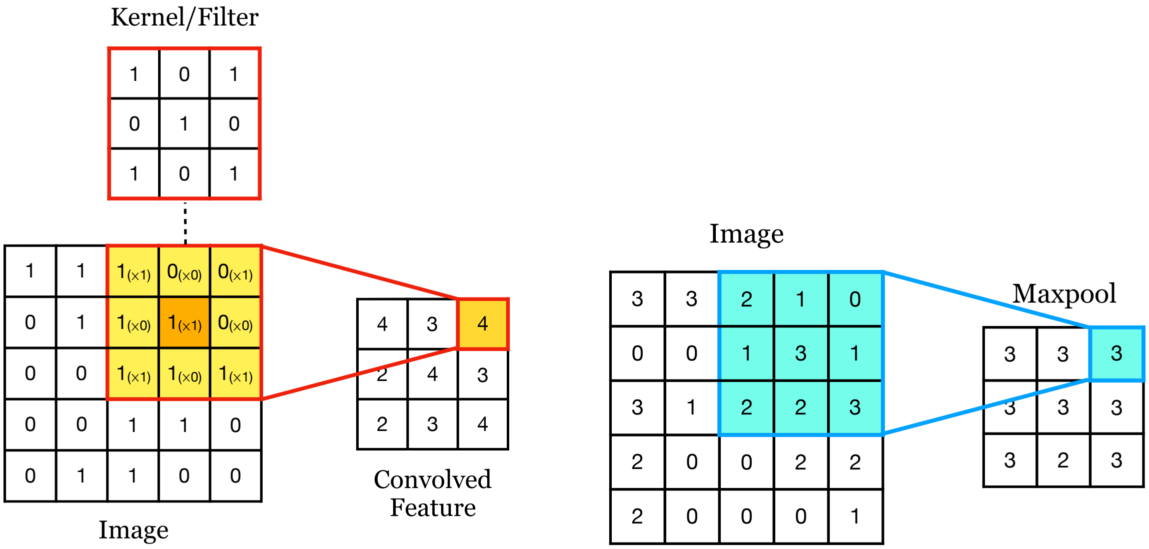


Figure 2.4: Application of a filter (left) and max-pooling layer (right) to reduce the dimensionality of an image feature. The convolution example demonstrates the reduction of a 5×5 image to a 3×3 subset through the use of a 3×3 filter. The max-pooling example is shown reducing a 5×5 image to a 3×3 subset.

are selected to systematically reduce the features spatial resolution and weights are tuned to minimize the Error/Loss. The spatial learning of a CNN is done through generalizations made across neighboring pixels as the filter sweeps over the inputs and imposes a form of weight sharing between them. Additionally, the pixel reduction results in fewer trainable parameters and helps the network isolate and learn separate components of the image at different scales. This form of deep learning is a good generalization approach for data with hierarchical or layered information.

2.1.3 Analysis Techniques for Hadronic Jets

A HEP classification task that naturally lends itself to a CNN is the discrimination of differing types of hadronic jets. Proton collisions at the LHC are capable of producing high energy quarks and gluons. These particles can be generated directly or via the production of other particles with decay modes to quarks and gluons (e.g. Higgs/W/Z). Due to color

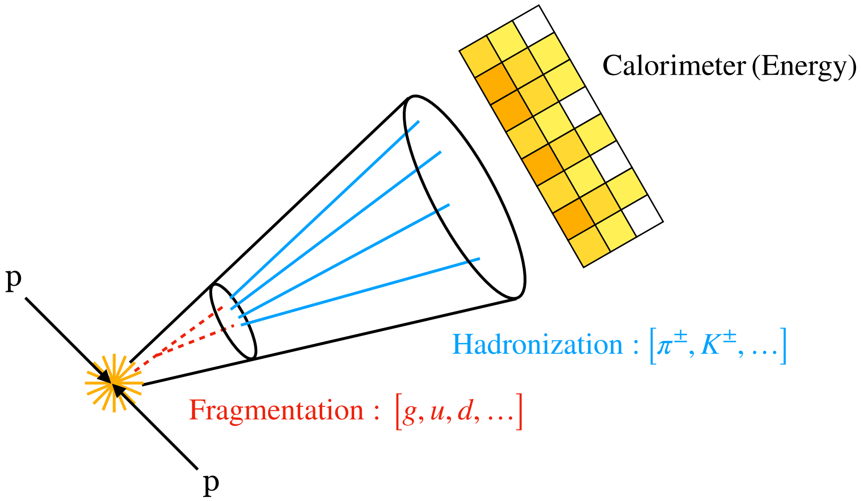

Calorimeter (Energy)
p
Hadronization : $[\pi^{\pm}, K^{\pm}, \ldots]$
Fragmentation : $[g, u, d, \ldots]$
p

Figure 2.5: Diagram for fragmentation (red) and hadronization (blue) generated from the collision of high-energy protons. The resulting particle shower energy and position is tracked by the cells of a calorimeter.

confinement, particles with non-zero color charge must undergo pair production with other color charged particles spontaneously generated from the vacuum to produce an overall color neural hadron. The hadrons are produced along the same primary axis of momentum of the original partons and this results in a collimated spray of particles. This jet of particles can then be measured in the tracker and calorimeters of the ATLAS detector (Fig. 2.5).

In practice, a more complete definition for jets exists that includes higher-order QCD corrections (in which additional real and virtual particles are included) and provides a pragmatic approach to measuring jets quantitatively. This working definition of a jet is is based on the selected hadrons that fall under a jet clustering algorithm and chosen clustering parameters (usually a jet radius R defined in (η, φ) space). The most commonly used of these jet definitions (and the default for applications in ATLAS) is the anti-kt algorithm [23]. A primary jet radius value is chosen (e.g. $R = 1$) and additional internal "subjets" can be defined by re-application of the anti-kt algorithm to the primary jet with a reduced radius parameter (e.g. $R = 0.2$).

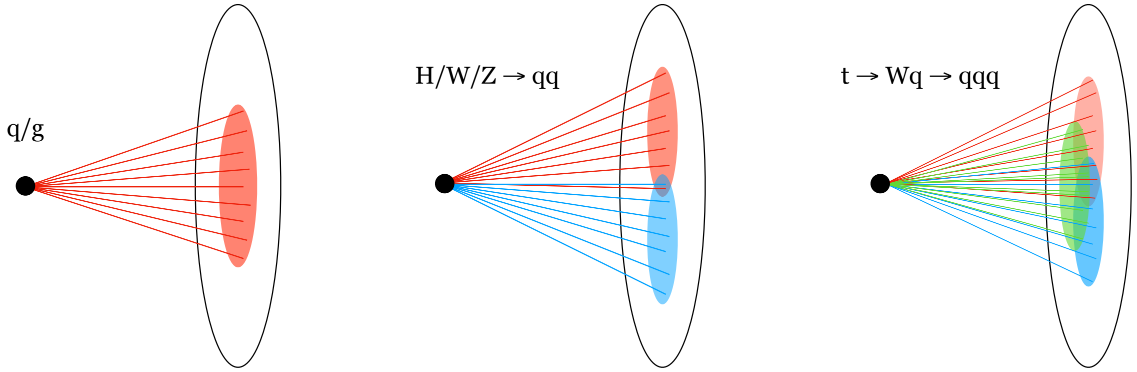


Figure 2.6: Representation of energy distribution patterns for various jets generated from quarks/gluons (left), dijets from Higgs/W/Z bosons (center) or a higher-order top quark decay to three quarks (right)

Jets are a useful object for detector analysis as their presence, size, energy depositions and multiplicity all give insights into the original process that created them. Examples of jet production and their characteristic energy depositions are given in Fig. 2.6. Much like other studied features, hadronic jet calorimeter data has also typically undergone dimensionality reduction to create physics-engineered features (Jet Substructure Observable (JSS)). An example of a common and general purpose observable for jet substructure is the jet invariant mass, which sums over all constituents in a jet to produce,

$$m^2 = \left(\sum_{i \in \text{jet}}^{N} p_i \right)^2 \tag{2.2}$$

However, other carefully tailored observables are described in the literature and these hold particular utility in specific discrimination tasks. Distinguishing between jets produced by quarks and gluons, for example, will benefit from the parameterization of calorimeter features as generalized angularities (GA). This class of feature is defined by the momentum and

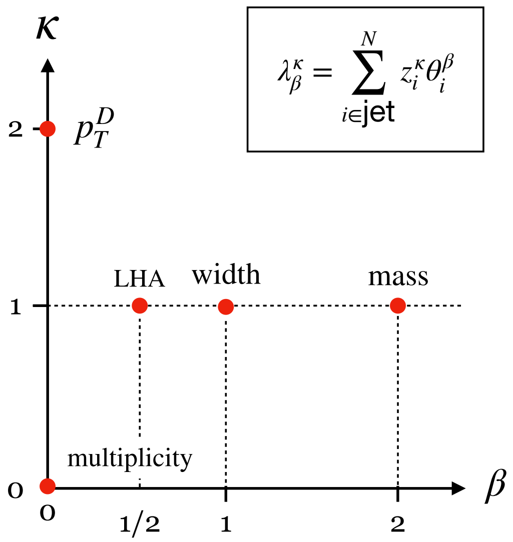

κ
$\lambda_\beta^\kappa = \sum_{i\in\text{jet}}^{N} z_i^\kappa \theta_i^\beta$
2
p_T^D
LHA
width
mass
1
multiplicity
0
0
1/2
1
2
β

Figure 2.7: Generalized angularity parameter space (κ and β), yielding jet substructure observables. Frequently used and named observables are charted by red dots and include: Les Houches Angularity (LHA), width, mass, p_{T}^{D} and multiplicity.

angular separation of constituents by,

$$z_i \equiv \frac{p_{T_i}}{\sum\limits_{j\in\text{jet}}^{N} p_{T_j}} \tag{2.3}$$

$$\theta_i \equiv \frac{R_{i,\hat{n}}}{R} \tag{2.4}$$

where $R_{i,\hat{n}}$ is the rapidity-azimuth distance to the jet axis. The GA substructure observables can then be calculated for different choice of κ and β parameter as,

$$\lambda_\beta^\kappa = \sum_{i\in\text{jet}}^{N} z_i^\kappa \theta_i^\beta \tag{2.5}$$

Examples of commonly selected features from the generalized angularity space are given in Fig. 2.7. In later literature, these angularity expressions are expanded to include many higher order summations over their constituents (sensitive to W/Z/Higgs boson related jets)[24]

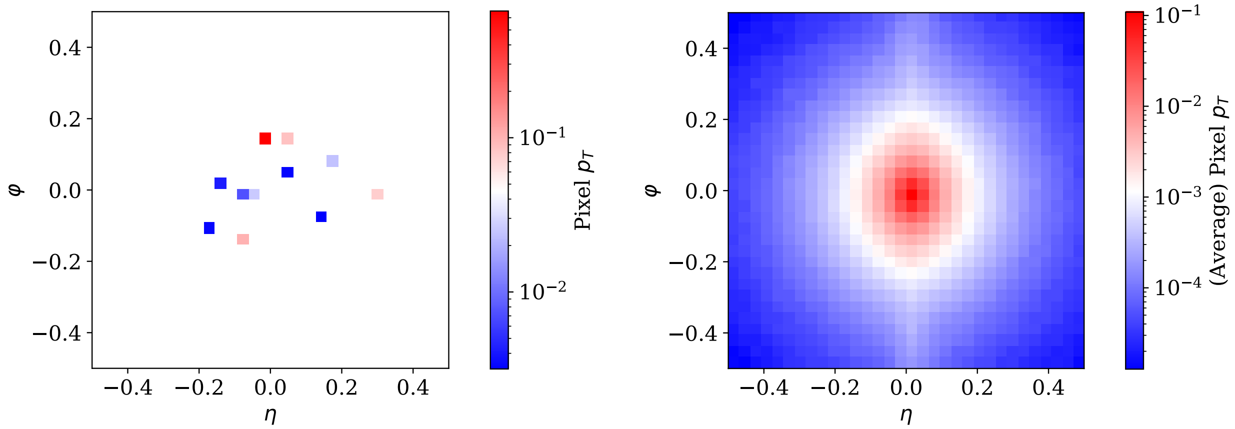


Figure 2.8: Jet image (left) and average over 1 Million jet images (right) produced from a QCD quark decay to a single prong jet in the hadronic calorimeter. This example jet image consists of a 32×32 grid of cells, spanning a space of $\Delta\eta = 0.8$ and $\Delta\varphi = 0.8$ of the hadronic calorimeter. Values are plotted logarithmically to account for the exponentially increasing constituent deposits in the center of the image.

and observables are developed for the untangling of multiple-prong jets that are often masked in the boosted regime[25].

2.1.4 Deep Learning and Jet Images

In contrast to the jet substructure approach to quantifying hadronic calorimeter data, an analog can be made to images and their use in computer vision machine learning. Hadrons produced in a detector like ATLAS will first pass through the inner detector and electromagnetic calorimeter before finally being absorbed by steel plates in the hadronic calorimeter (HCal). The HCal is interspersed with scintillator tiles that allow for the measurement of the position and energy of incident particles. This measurement region provides a grid of tracking cells across a width of $|\eta| < 3.2$ with spacing between cells of approximately $\Delta\eta \times \Delta\varphi = 0.1 \times 0.1$. Taking measurements directly from the calorimeter yields data parameterized similarly to that of a single-channel (grayscale) image composed of pixel values in an (x, y) grid. Exporting calorimeter measurements into a two-dimensional form, one can create a *jet image* (Fig. 2.8) in (η, φ) space.

These jet images undergo a relatively minimal set of pre-processing steps. Primarily, images are centered such that average p_{T} pixel value sits at $(0, 0)$. Note that although the calorimeter is taking energy measurements, jet images are translated into terms of their transverse momentum. Although translations in the φ direction amount to an invariant rotation about the detector's $z-$axis, a translation in the η direction is equivalent to a Lorentz boost along that z-axis. An artificial boost in the z-axis would modify the true measured energy during pre-processing. Converting pixel values to contain $p_{T,i} = E_i / \cosh{(\eta_i)}$ maintains translation invariance for movements inside of the (η, φ) plane.

Despite the minimal pre-processing and higher-dimensional nature of jet images, training convolutional networks with these features often demonstrates superior performance when compared to the physics-engineered jet substructure examples [26–28]. This is an important confirmation for the idea that deep learning methods can be used to access better solutions and potentially isolate novel physics from higher-dimensional datasets. However, it also motivates new important questions for the original application of jet substructure. Given the carefully catered design of JSS observables, why does the low-dimensional example perform worse? For a black box model, like a deep neural network using convolutional layers, this becomes a complicated question.

2.2 The Black Box Problem

For deep learning, a seemingly simple question like *"why has model X performed better than model Y"* is a generally complex, if not unanswerable, query. Entire classes of machine learning and artificial intelligence architectures function as opaque problem solvers. These models are considered "black box" methods and they can not be trivially evaluated, tested or understood. There is no bright line distinguishing between opaque and transparent learning methods and, at present, no simple way to quantify the degree to which one learned model is more opaque than any other.

2.2.1 What is a Black Box?

Despite being an unquantifiable property, model transparency still has some essential feature definitions. One fundamental aspect of transparency is the ability to "look inside" and understand how individual components function. A system may be highly complex but if it is possible, in principle, to track the function of individual parts and their connections to other components, it is not opaque. Although it's technically possible to "look inside" a deep neural network, doing so will reveal little about how the chosen weights are contributing to predictions. The initial hyper-parameters and network architecture are also easily observed, but this is similarly not instructive in understanding the solution. This interpretability problem is, in some sense, an intentional feature. By design, black box models are adept at finding subtle patterns for abstract relationships across many high-dimensional features and these discoveries are not well-defined in simple functional ways. The flexibility to find those patterns, however, makes it difficult to reverse engineer the solution.

Opaque networks also have the unfortunate property of masking how inputs are being used. Typically, for learning abstracted to a higher-dimensional space, the training set becomes a mélange of features that influence each other in unpredictable ways. The degree to which any one variable is used and the ways in which they are used with one another is unclear and can vary wildly with even minor tweaks of the network architecture or parameter selection. Consider, for example, a *holdout* analysis in which a network is trained with a set of input features followed by the training of an identical network with one input excluded[29]. If the network performs equally well in both scenarios, it would be tempting to conclude that the first network did not use the feature that was later removed. This, however, is an incorrect assumption (as will be seen in Sec. 6.4.2). In fact, with even a minor change in input features, one can radically alter the parameter space that the network is being optimized for. A network that draws a decision boundary in its own latent space can't be used, as reference, to conclude how a similar network chose its own decision boundary. This is true even for pairs of networks that differ only by a single input feature. A network trained on features X may find one solution to a problem. A similar network trained on features X' may find a different and equally accurate solution to the same question. Those networks ultimately have to be treated as unrelated.

2.2.2 The Problem with Black Boxes

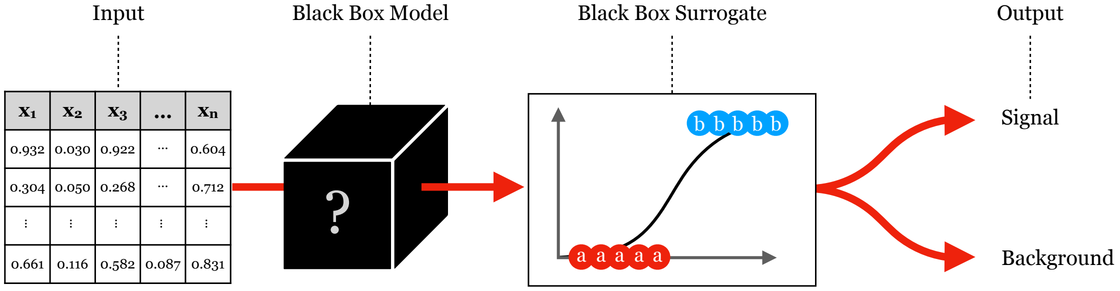
Ernest Rutherford is often attributed with the quote, "All science is either physics or stamp collecting". Although most likely apocryphal, the sentiment illustrates an important epistemic challenge for the use opaque learning methods. In the context of the sciences, and physics in particular, the ultimate objective of any study involves description, prediction and a fundamental understanding of the system being studied. To answer any question in physics requires more than observing and cataloging a phenomena. Opaque models can be a strong motivation for directing searches and for indicating more or less fruitful paths of research. However, satisfaction with a more accurate learning strategy at the expense of interpretability is antithetical to the greater project of understanding the universe.

Even if one is willing to ignore the intellectual problems of using uninterpreted opaque models, they also pose significant practical problems. One of the most significant drawbacks is a lack of trust in the learned solution. In critical applications, such as ML in healthcare, the ability to check a models solution for flaws or dubious results due to poor data collection/preprocessing is a necessity (for an example, see Ref. [30]). For less life-threatening analysis, this same concern over trust exists in the form of doubt about the validity of the data-driven solution. For a transparent learning approach, it is possible to evaluate the way in which a model is using its features. If a feature is being used in a way that seems unreasonable or lacks a good basis in the theory it seeks to model, that feature can be removed or constrained. Additionally, the effects of pre-processing can be studied and systematic uncertainties for those features can be addressed individually.

This kind of simple analysis is not possible with a black model. Features may contribute information to a prediction which is incorrect or introduced to the data through the application of dimensionality reduction or pre-processing. Those mistakes can not be evaluated by a black box model and can't be spotted by the physicist using it. Similarly, it becomes impossible to establish which features are truly useful, which are superfluous and which contain discrimination information when used in conjunction with other features. The general inability to connect learned information from an opaque model to scientific bedrock can impact learning on either a *global* or *local* level in a theory. To address these global/local interpretability problems, a few techniques have been developed that provide modest improvements to model transparency.

Global Model Interpretability

For a model trained on a full feature set, it must learn a wider and more comprehensive theory to make predictions. Understanding how the complete black box makes global choices can't be deduced by inspecting the model's local decisions. One approach to evaluating the global influence of a model on features is the use of a surrogate model [31, 32]. A black box surrogate involves the inclusion of some simple and transparent learning method (e.g. logistic regression) placed as a proxy between the black box model and the final output layer. The surrogate network then acts as an interpretable component learning from the outputs of the black box. The surrogate gives some interpretability to the black box itself, although



x_1	x_2	x_3	...	x_n
0.932	0.030	0.922	···	0.604
0.304	0.050	0.268	···	0.712
⋮	⋮	⋮	⋮	⋮
0.661	0.116	0.582	0.087	0.831

Figure 2.9: Diagram of a surrogate machine learning approach for global network interpretability.

no conclusions can be drawn about the original inputs. A cartoon example of a black box surrogate process is given in Fig. 2.9.

Although this approach can give some insights into the global approach of a black box model, the information accessible from the surrogate is limited. Additionally, performance is often lost in the process of converting solutions from the black box to the surrogate model. Ironically, this approach is particularly insubstantial when the learned information one is interested in is precisely that nuanced relationship which the black box model was necessary to tease out in the first place.

Local Model Interpretability

Rather than consider the global solution, one can study individual features and explore local relationships between them by probing smaller sections of the network. A popular approach to evaluating subsets of a model is given by *Local Interpretable Model-Agnostic Explanations* (LIME) [33]. This utility attempts to explain model decisions inside of a network's latent space by generating a new set of features in a region of interest with small perturbations. A separate surrogate model is then trained from the prediction of the black box on the perturbed features, weighted relative to their distance from the region being studied. The resulting surrogate model acts as a local simplified approximation for the black box in the

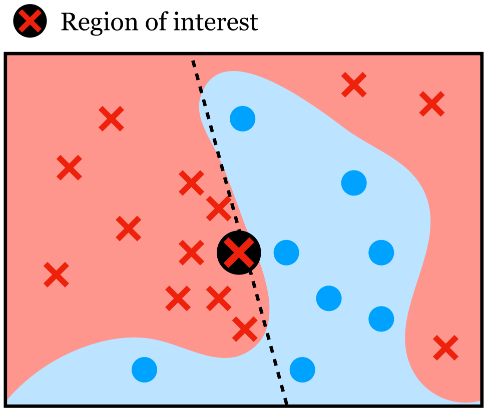


Figure 2.10: Simple non-linear latent space example with a locally valid linear classification boundary learnable through LIME/SHAP (or other local interpretability technique)

region of interest. Although the surrogate model is not accurate globally, it provides an interpretable learned version of that local section of the network. A simple cartoon example of a locally accurate linear decision surface in a non-linear space is given in Fig. 2.10

The approach to local interpretability in LIME is compelling but suffers from a few drawbacks. Defining the size of the local region and the weighted distance in the latent parameter space is non-trivial. Additionally, impacts to the model for even small changes in the latent space are often unstable [34]. This stability can be addressed through the use of *Shapley values*[35], a game theory technique for correctly apportioning the influence of features in gaining/losing performance in a learning task. The feature significance is addressed, with SHAP, by measuring contribution weight given variations in predictions measured through the permutation of a models input features.

A simple jet substructure learning example for predicting the mass of a hypothetical particle, m_X, using high-level features and SHAP analysis is given in Fig. 2.11. In this permutation tree, nodes represent individual models trained with the features indicated at the top (e.g. $\tau_{21}^{\beta=1}$, p_{T} or M_{jet}) and edges represent the marginal contribution (MC) for a feature to a model. SHAP values are measured for all models in the tree given some sample input, x_0.

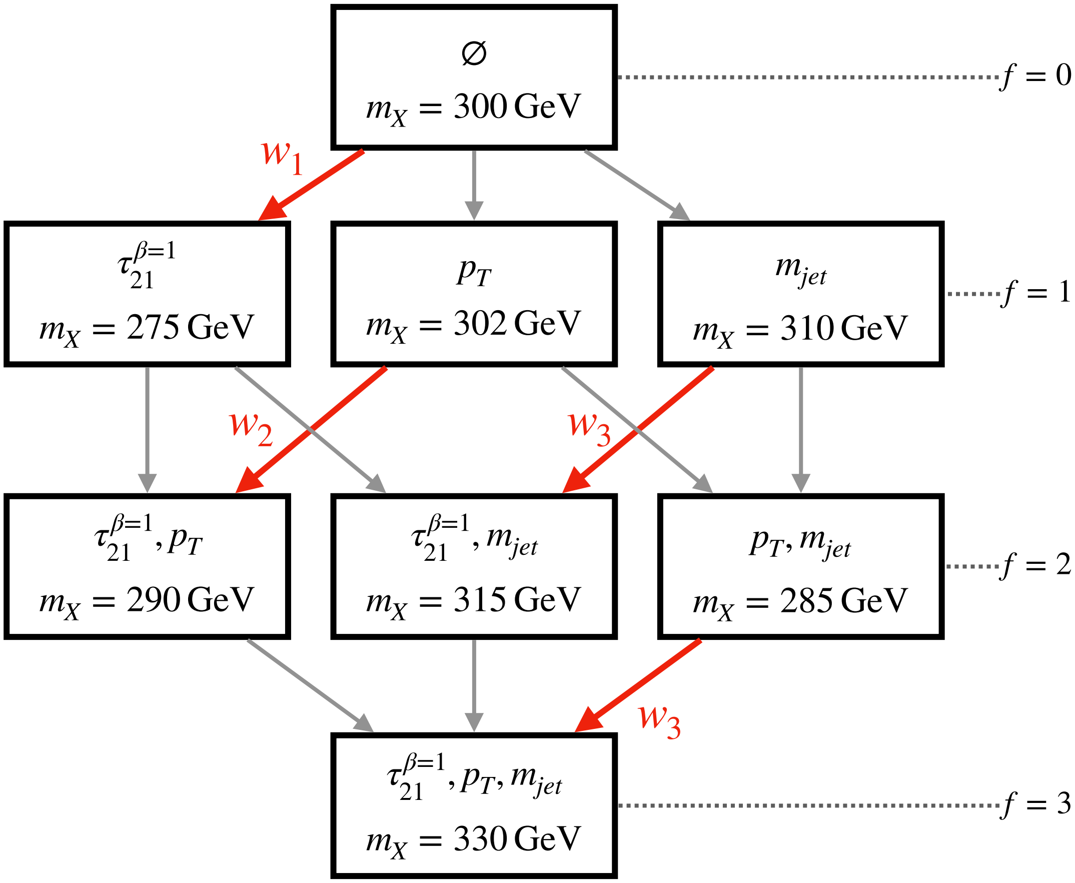


Figure 2.11: SHAP permutation tree for jet substructure observables $\left(\tau_{21}^{\beta=1}, p_{\mathrm{T}} \text{ and } M_{\mathrm{jet}}\right)$ for the example problem of predicting a hypothetical particle mass m_X. Each node represents a separate model, prediction and contributor in the final Shapley analysis.

In the initial *null model* ($\emptyset$), the base-line prediction is made by simply taking the average value of m_X. In the second level ($f = 1$), predictions on x_0 are made for a model including each feature separately. Subsequent levels ($f \geq 2$) then make predictions on x_0 for high-order mutual interactions between features. The marginal contribution for feature $\tau_{21}^{\beta=1}$ given a model trained with only that individual feature (layer $f = 1$) is given by

$$\begin{aligned}
\mathrm{MC}_{\tau_{21}^{\beta=1},\left\{\tau_{21}^{\beta=1}\right\}}(x_0) &= \mathrm{Predict}_{\left\{\tau_{21}^{\beta=1}\right\}}(x_0) - \mathrm{Predict}_{\emptyset}(x_0) && (2.6)\\
&= 275\,\mathrm{GeV} - 300\,\mathrm{GeV} && (2.7)\\
&= -25\,\mathrm{GeV} && (2.8)
\end{aligned}$$

marginal contribution values are calculated for every edge in the graph and then used to compute the overall SHAP value for a feature given input x_0. The SHAP value for the feature $\tau_{21}^{\beta=1}$ is computed by,

$$\begin{aligned}
\mathrm{SHAP}_{\tau_{21}^{\beta=1}}(x_0) =& w_1 \times \mathrm{MC}_{\tau_{21}^{\beta=1}\{\tau_{21}^{\beta=1}\}}(x_0) + w_2 \times \mathrm{MC}_{\tau_{21}^{\beta=1}\{\tau_{21}^{\beta=1},p_{\mathrm{T}}\}}(x_0) && (2.9)\\
&+ w_3 \times \mathrm{MC}_{\tau_{21}^{\beta=1}\{\tau_{21}^{\beta=1},M_{\mathrm{jet}}\}}(x_0) + w_4 \times \mathrm{MC}_{\tau_{21}^{\beta=1}\{\tau_{21}^{\beta=1},p_{\mathrm{T}},M_{\mathrm{jet}}\}}(x_0) && (2.10)
\end{aligned}$$

The marginal contributions are weighted according to three requirements. First, the weights should represent the probabilistic contribution for the given edges and so sum to unity

$$w_1 + w_2 + w_3 + w_4 = 1 \qquad (2.11)$$

Second, the sum of the marginal contribution for the sum of the weights in a one-feature model must be equal to a two-feature model, three-feature model, and so on. This then requires,

$$w_1 = w_2 + w_3 = w_4 \qquad (2.12)$$

Finally, the marginal contribution must be shared equally by all permutations in the level $(f = 1, 2, \ldots)$, which in this example requires: $w_2 = w_3$. Applying these three criteria, the weights for the simple three-feature example becomes: $\left(w_1 = \frac{1}{3}, w_2 = \frac{1}{6}, w_3 = \frac{1}{6}, w_4 = \frac{1}{3}\right)$. SHAP values can then be averaged over many samples of x_i in the modeled dataset. This contribution weighting scheme gives better apportionment for feature influence on the studied model. An example of the benefits of SHAP analysis are demonstrated, and used for partial analysis, in a dark matter learning example in Sec. 6.4.2.

In comparison to global model interpretation, these local analyses are often more illuminating due to the reduced scope of the problem. Popular techniques like LIME and SHAP can give some insights into how individual features are being utilized. However, these benefits are still limited and interpretation remains difficult. Knowledge of the marginal contribution of features in a small segment of the model still gives relatively little understanding as to how the features are being used generally, how they relate to one another and what aspects of the feature ultimately benefit the learned solution.

2.2.3 Black Box Applications in High-Energy Physics

An example of a black box model yielding performance improvements on a practical high-energy classification task is given at the end of Sec. 2.1.4. However, one might reasonably wonder how pervasive this black box problem is in HEP. Reviewing the literature, additional examples can be found in the areas of: event classification [26, 36–40], jet substructure studies [27, 28, 41–43], jet flavor classification [44–48], detector unfolding [49–51], and uncertainty estimation [52–56]. As computational techniques continue to improve and high-energy problems become more complex, examples will likely become more frequent.

Addressing the black box problem in specifically the context of high-energy physics serves a number of important purposes. First, it's crucial that information used by ML models is validated as being real and physical, as opposed to the accidental use of artifacts introduced to the data through simulation, processing, etc. Even in cases where training doesn't use simulated inputs [57–59], it is relevant to understand what information is being used and how. Translating a black box method into a simpler and understandable format allows for validation of those inputs. Second, an interpretable strategy based on HL inputs allows for more reliable estimates of systematic uncertainties. For a sufficiently small set of features, those observables can be studied and calibrated individually and knowledge can be extended to future problems solved with the same or similar inputs. Third, replacing a more sophisticated ML strategy with a simpler model with fewer inputs allows for faster interference and lower memory requirements at run-time [60, 61]. Lastly, if overlooked information in the LL data is physical, identifying it can provide new insights into the nature of the problem.

2.3 Finding a Path Forward

Addressing the black box problem in high-energy applications becomes an important but difficult task. Previous strategies have been proposed in the literature to draw connections between a learned NN strategy and existing HL observables. For example, one can compare a models performance both with and without an included HL observable [29] or use a technique of projecting a models decision surface along the HL observables [26]. Alternatively, one can expand the NN function in a basis of the input features [62–64]. These strategies are useful but are primarily limited to studying the structure of the NN in terms of already-identified HL observables. An important and distinct goal from these approaches is to demonstrate a more comprehensive search method which makes selections from a broad space of features and systematically maps a black box strategy into that lesser known space.

In Chp. 3, this objective is achieved through a technique for translating information learned by a black box ML strategy on LL inputs into a more meaningful and interpretable set of

HL observables. Rather than attempt to directly interpret the ML model, the network is used as a guide for the construction of a classifier which makes equivalent decisions while relying on a small set of simple physics motivated and human-interpretable features. These features are chosen through an iterative process from a large space of candidate observables. This approach is presented on a case study of jet classification [28], using convolutional neural networks to guide in the selection of a small set of HL observables called *energy flow polynomials* (EFP) [65]. The conclusion of Chp. 3 will show that a CNN trained on LL inputs can be translated into a more compact and meaningful set of EFPs. Ultimately, these results suggests a new set of HL observables that physicists should consider relevant to jet substructure classification. In Chps. 4 and 5, this technique is extended to two modern standard model searches for electron identification and prompt muon isolation. Finally, in Chp. 6, this approach is applied to a beyond the standard model search for semi-visible jets produced by a dark matter quark with partial accessibility to a dark sector regime.

Chapter 3

Translating Black Box Models To a Human Readable Space

3.1 Introduction

An outline of this chapter is as follows: In Sec. 3.2, a method for mapping an ML model's learned solution into a human-readable space is presented. This mapping involves the construction of a broad set of candidate HL observables from the EFP space. From this EFP space, a similarity metric, *average decision ordering* (ADO), is used to compare the relative decisions made by two classifiers. Finally, the ADO is used to demonstrate a method for iteratively mapping between HL observables and a fixed black box ML model according to the maximum ADO between them.

In Sec. 3.3, this method is demonstrated on a real-world HEP problem involving the discrimination between jets originating from a boosted W boson and those generated from light quarks or gluons. This is a well-studied problem in the area of jet substructure [42, 66–76], where both HL [24, 25, 77–80] and NN strategies [27, 28, 41] have proven effective.

The starting point of this analysis is Ref. [28], in which a small but persistent improvement in classification performance with a deeply-connected CNN is found when compared to a boosted decision tree (BDT) of HL observables. To augment this set of features for jet tagging, a search is made through the space of EFPs [81], which forms an (over)complete basis of collider observables that are infrared and collinear (IRC) safe. This search is then extended to include IRC-unsafe variants of the EFP observables that have been successful in past jet tagging studies [82–85].

In Secs. 3.4 and 3.5, results for the case study are presented. Starting with the set of six HL observables from [28], a black box guided strategy is used to identify a seventh HL observable that closes the performance gap with a CNN. A black box guided strategy is then attempted starting with just mass and transverse momentum of the jet, comparing the results to a brute force strategy of directly searching the space of EFPs and a guided search based on ground truth labels. This comparison shows that the black box guided strategy significantly outperforms the label guided search, reaching comparable performance to the brute force strategy with considerably reduced computational costs. These comparisons are then analyzed and physical interpretations of the translated ML strategy and its connection to the broader context are given in the discussion in Sec. 3.6.

3.2 Translating from Machine to Human

In order to map information between a HL and LL space, it's desirable to identify a small set of physically-motivated HL observables that, when combined into a joint classifier, will make the same classification choices as a DNN operating on LL features. The primary goal is to establish a set of HL features which, when combined, will maximize the classification performance by following the interpreted strategy from the black box NN. This will provide a more efficient training strategy when compared to training directly with "ground truth" information. If successful, the resulting features will have expressed the ML strategy more simply and transparently than any network using the LL inputs.

In the first step, a potential source of HL observables must be chosen from which a novel physics-motivated set of features can be selected. This step still requires human knowledge of the problem and insight from the physicist. This step, at present, can not be automated or outsourced to an algorithmic process. For the specific case of jet substructure, the space of Energy Flow Polynomials (EFPs) is selected as a suitable and comprehensive basis of HL observables [81]. For other ML tasks in HEP, a separate set of HL observables may be necessary as the basis for that feature space.

In this section, the algorithmic approach and statistical arguments for the method are presented with a focus on a binary classification problem. To evaluate the performance of the simple HL network and a black box NN, a metric needs to be chosen to evaluate the similarity of two classifiers and their decision surfaces. A variety of metrics exist for related objectives but, in this instance, a custom metric has been developed to directly solve this problem. The Average Decision Ordering (ADO) is introduced and is preferred for this task as it shares a conceptual simplicity to the Area Under the Curve (AUC) metric which is often used to benchmark ML classifiers against ground truth. With this tool, an explorable set of HL observables can be evaluated for its learning similarity with a black box NN and the results are mappable to a meaningful physical space.

3.2.1 Average Decision Ordering

The guided strategy proposed in the following sections will hinge on the use of a similarity metric capable of comparing two decision functions, $f(x)$ and $g(x)$. In this context, x represents any arbitrarily complex set of inputs used in either the black box NN or the physically-motivated HL observables. The similarity between these decision functions must reflect the classification task of interest which, in this case, is a binary classification problem. Because classification performance is invariant under any non-linear monotonic transformation of f or g, the similarity metric cannot be affected by such a transformation. As such, simple comparisons such as functional overlap or linear correlation are insufficient as metrics.

Similarly, it is not sufficient to simply compare the overall performance of the two classifiers for the equivalent dataset as this measurement does not provide insight into how the inputs are being used to make the scored predictions. As is discussed in Ref. [84], two decision functions are capable of using information from two distinct regions of the LL input space and making conflicting classification decisions (on a case by case basis) while achieving an overall similar level of performance. The key to this study is to map the specific decision making on features between two networks and not merely find networks with comparable classification performance.

To start, it's assumed that the decision functions $f(x)$ and $g(x)$ are real valued and that the final binary classification is determined by a threshold on the decision function output. Objects on one side of the decision surface will be labelled "signal" and those on the opposite side are labelled "background". Depending on the application, this threshold can be tuned to different points on the receiver operating characteristic (ROC) curve to optimize the signal acceptance vs background rejection. The overall classification performance is then measured by the Area Under the Cure (AUC) of the ROC. This is equivalent to the probability that a randomly selected signal/background pair is correctly ordered by a decision function f(x):

$$\mathrm{AUC}(f) = \int \mathrm{d}x\, \mathrm{d}x'\, p_{\mathrm{sig}}(x) p_{\mathrm{bkg}}\left(x'\right)\, \Theta\left(f(x) - f\left(x'\right)\right). \tag{3.1}$$

Where Θ is the Heaviside theta function (i.e. $\Theta(x < 0) = 0$ and $\Theta(x \geq 0) = 1$) and p_{sig} and p_{bkg} are the ground truth signal and background probability distributions. Using the AUC, a perfect decision function has AUC=1 and random guessing yields AUC $= \frac{1}{2}$.

The AUC is simply a measurement of the individual performance of a model in its predictions. In order to compare the classification behavior of two different decision functions, the decision surface defined by a threshold on the function output is considered. For each function, the set of thresholds defines a set of surfaces in x space. If the two decision functions have identical decision surfaces, they are effectively using the same information for classification. Note that the absolute output values for the decision functions are not relevant for determining whether the decision surfaces are similar. The relative locations of the decision surfaces are determined by the relative ordering of the two decision functions when evaluated at pairs of points in the input space. This notion can be efficiently encapsulated by the *decision ordering* (DO) for a pair of inputs x and x':

$$\mathrm{DO}\left(f, g, x, x'\right) = \Theta\left(\left(f(x) - f\left(x'\right)\right)\left(g(x) - g\left(x'\right)\right)\right), \tag{3.2}$$

where DO=1 corresponds to f and g having the same ordering and DO=0 corresponding to an inverted ordering. If two decision functions have DO=1 for all pairs (x, x'), then they are monotonically related to each other, have identical decision surfaces, and are therefore identical decision makers for the purposes of classification. To build a summary statistic, the DO metric can be averaged over all possible pairs of (x, x'), weighted by signal and background distributions, yielding the *average decision ordering* (ADO):

$$\mathrm{ADO}(f, g) = \int \mathrm{d}x\, \mathrm{d}x'\, p_{\mathrm{sig}}(x) p_{\mathrm{bkg}}\left(x'\right) \mathrm{DO}\left(f, g, x, x'\right). \tag{3.3}$$

This expression evaluates to 1 when the decision functions make the same relative classification decision for every pair, to 0 if the functions make the opposite classification for every pair, and to $\frac{1}{2}$ if there is no consistency in their orderings. Since a decision function can be trivially inverted, the case of ADO $= 0$ and ADO $= 1$ are equivalent. As such, the final measure can be mapped to ADO $\rightarrow 1 -$ ADO for all ADO $< \frac{1}{2}$. The ADO has a similar philosophy to Kendall's rank correlation coefficient [86], with the important difference that compared inputs are drawn from separate signal and background distributions.

3.2.2 Understanding the ADO

To gain intuition for the ADO, a comparison can be made to the AUC given in Eq. (3.1). While the AUC measures the probability that a single decision function orders objects correctly relative to ground truth, the ADO gives the probability that two decision functions order objects in the same way, even when done so incorrectly. For a likelihood ratio $f(x) = p_{\text{sig}}(x)/p_{\text{bkg}}(x)$, $f(x)$ is an optimal classifier by the Neyman-Pearson lemma, so an ADO $= 1$ implies that $g(x)$ defines the same optimal decision boundary as $f(x)$. In contrast to maximizing towards the AUC, where an ML application is looking for optimal performance, the guided strategy attempts to maximize ADO to optimize for decision similarity.

There are other similarity metrics that one could use, however they are less interpretable in terms of pure classification decisions. One alternate method for capturing similarity is to use the mutual information or, more appropriate to a binary classification problem, mutual information with the truth [84]. For the guided strategy presented here, whether two decision functions have the same quantity of information available for a classification task answers a different question than whether they are using that information in the same way. Even if $f(x)$ and $g(x)$ contain a high level of mutual information, this doesn't guarantee that they have equivalent decision boundaries. This is a crucial distinction to make in the context of deep networks where the flexibility and high dimensionality of the latent space allows for an ML strategy to find many different solutions to a problem using the same set of information.

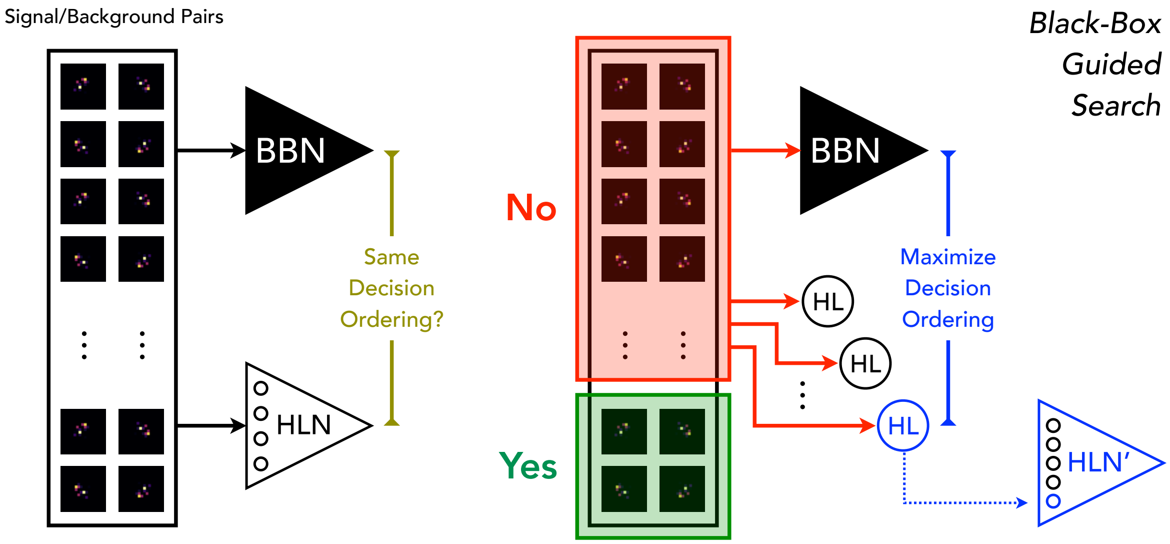


Figure 3.1: Schematic for the black box guiding strategy given in Sec. 3.2.3. In each iteration of the strategy, the decision ordering of signal and background pairs between a fixed black box network (BBN, black triangle) and a trainable network of HL observables (HLN, white triangle) is used to identify the subset (red box) in which pairs are differently ordered. From a large space of HL observables (circles), the one with the largest ADO in the differently ordered space (blue circle) is chosen for the next iteration. The schematic corresponds to the n = 4 iteration. Note that the BBN is not retrained in each iteration, but the network of HL observables is.

3.2.3 Black Box Guided Search Strategy

A graphical representation of the black box guided strategy is given in Fig. 3.1, where the goal is to find the HL observables that maximize the ADO relative to an already trained ML tool. The black box network is labelled “BBN” and is, in this context, a deep network acting on a set of LL inputs. Starting from a large set of physicist selected HL observables, set $\mathcal{S}$, the goal is to train a HL network (HLN) with the same decision surfaces as BBN. In the initial step ($n = 0$), the first observable (HL_1) is selected with the largest ADO with the BBN:

$$\mathrm{HL}_1 = \underset{\mathrm{HL}\in\mathcal{S}}{\mathrm{argmax}}\ \mathrm{ADO}\,(\mathrm{BBN}, \mathrm{HL})_{X_{\mathrm{all}}} \tag{3.4}$$

Here, X_{all} indicates that the full set of signal/background training pairs (x, x') is being used when computing ADO. The first observable, HL_1 is therefore the physics-motivated observable in the set $\mathcal{S}$ that best approximates the decision surfaces of the BBN. In the following step, $n = 1$, the focus of the search moves to finding the region of the feature space where the BBN disagrees with the current set of HL observables by isolating the subset of signal/background pairs X_1 that are differently ordered by both the BBN and HL_1:

$$X_1 = \{(x, x') \,|\, \text{DO}\left(\text{BBN}, \text{HL}_1, x, x'\right) = 0\} \tag{3.5}$$

The next HL observable (HL_2) can then be selected by isolating the largest ADO with the BBN when restricted to the X_1 subset:

$$\text{HL}_2 = \underset{\text{HL}\in\mathcal{S}}{\text{argmax}}\ \text{ADO}\left(\text{BBN}, \text{HL}\right)_{X_1} \tag{3.6}$$

For each subsequent step for $n > 1$, the HL observables are combined with the previously identified features in the previous steps into a joint network

$$\text{HLN}_n = \text{NN}\left(\text{HL}_1, \ldots, \text{HL}_n\right), \tag{3.7}$$

where the neural network, NN, is trained on the full signal/background training set with n HL observables as inputs. From this joint HLN, the differently ordered subset X_n is given by

$$X_n = \{(x, x') \,|\, \text{DO}\left(\text{BBN}, \text{HL}_n, x, x'\right) = 0\} \tag{3.8}$$

Because a new HLN is trained in every iteration, X_n may not be a strict subset of X_{n-1}. The next observable HL_{n+1} is determined by

$$\text{HL}_{n+1} = \underset{\text{HL}\in\mathcal{S}}{\text{argmax}}\ \text{ADO}\left(\text{BBN}, \text{HL}\right)_{X_n} \tag{3.9}$$

Note that the same BBN is used in each iteration but the changing subset X_n provides a different decision surface test at every step. This iterative process is repeated until $\mathrm{ADO}\,(\mathrm{BBN}, \mathrm{HLN}_{n+1})$ gives a value as close to 1 as desired.

Isolating the differently-classified pairs in Eq. (3.8) is similar in spirit to the boosting step of BDTs [87, 88]. This approach focuses only on the subspace of pairs where the BBN disagrees with the current set of HL observables, providing a look at new HL observables that make signal/background ordering decisions most similar to the BBN in that subspace.

The ADO, or some other similarity metric, is crucial to the comparison of the BBN and HL networks for this kind of translation between HL and LL information. In Sec. 3.5.3, an attempt to perform a similar iterative method will be shown using truth labels. Instead of the ADO, the label guided approach uses the AUC with respect to the ground truth information. The ADO proves to be more effective at this task and it is straightforward to see why this must be the case in the guided search. To the extent that the BBN is well trained, it represents the best approximation to the Neyman-Pearson optimal classifier. Achieving the correct DO relative to this optimal classifier for all signal/background sample pairs is the best one could hope to do with that dataset. Therefore, a guiding strategy which attempts to iteratively reduce the space of the HL model where inputs are differently ordered from the BBN will approach the same decision making as that optimal classifier.

In contrast, the AUC guided method captures the DO relative to truth labels. Unless the BBN is able to achieve an AUC = 1, there will necessarily be a subset of signal/background pairs that are incorrectly ordered even by the maximally performant classifier. Instead of reducing the size of the differently ordered space of features in the HL classifier, an AUC guided approach will inevitably become fixated on a space of differently ordered pairs which, in principle, can't be correctly ordered by any classifier. As such, an ADO guided approach is preferable as the HLN_n will be guided explicitly towards mimicking the most optimal solution to the problem as determined by the LL model.

The black box guided approach is considered a “greedy algorithm” (i.e. a strategy that makes the locally optimal solution at each step in an algorithm) and as such, it cannot identify situations where two HL observables could be combined simultaneously to match the BBN decision surface. This means that the algorithm is likely to prefer solutions which favor individually strong features as opposed to those which perform poorly on their own but perform well jointly with another set of observables. If the objective was to simply maximize performance, this would be an undesirable feature. However, in the context of mapping a black box ML strategy to a physically-interpretable space, this is beneficial. With this algorithm, observables chosen for their individual accessible information are more likely to lead to physical insight that can be understand as a stand alone input.

3.3 A Case Study in Jet Substructure

Armed with the guiding strategy given in Sec. 3.5, the method can be applied to a practical physics problem related to jet classification at the LHC. In this section, a test case is given for a boosted W boson classifier that compares jet substructure information to jet images as demonstrated in Ref. [28]. The EFP space is then introduced as a candidate space of HL observables from which to supplement the results of a classifier.

3.3.1 Boosted Boson Classification

Massive objects produced at the LHC often have enough transverse momentum that their decay products will become collimated. For an object with a hadronic decay mode, such as the W boson decaying to a quark and anti-quark pair ($W \to q\bar{q}$), the resulting jet in the detector consists of two clusters of energy (one from each of the fragmenting quarks). The substructure of these jets is distinct from those that arise during fragmentation of a single hard quark or gluon. Identifying these jets with nontrivial substructure has become an essential tool for probing the nature of collisions at the LHC [42, 66–76, 89].

There are many methods of representing the information contained within a jet. At it's most fundamental level, a jet is simply a collection of four-vectors representing the constituent particle kinematics that make up the jet, motivating set-based ML tools [81, 90–94]. Another popular approach is to describe a jet as a grid of calorimeter cells with energy depositions, giving rise to a "jet image" [27, 95]. In any of these low-level representations, the jet data is high-dimensional. This motivates the creation of high-level/low-dimensional observables that simplify and attempt to summarize the low-level information into a smaller dimensionality. Physicists have engineered such HL observables that incorporate knowledge of the nature of jets (see Ref. [24, 25, 70, 77–80, 96–101] for examples). Typical usage involves the application of cuts on one or more of these HL observables, or the combination of multiple observables in a shallow ML classifier.

In the context of jet classification, ML tools have been shown to outperform traditional strategies using HL observables through training on low-level inputs [102]. This result is not necessarily surprising as the HL observables, themselves, are just lower-dimensional representations of the low-level inputs. This fact also suggests that through the collection of many HL observables, one can find an informationally complete set of simple features that can match the performance of the LL classifiers [103–105].

This case study is based on the same datasets as Ref. [28]. These datasets correspond to a $\sqrt{s} = 14\,\text{TeV}$ proton-proton collision where hard scattering and resonance decay were generated using MadGraph5 v2.2.3 [106], showering and hadronization were performed with Pythia v6.426 [107] and the response of the detectors was simulated with Delphes v3.2.0 [108]. The boosted W signal is a diboson production ($pp \to W^+W^-$), which produces two fat jets each with 2-prong substructure. For a background, QCD dijets are produced from quark and gluons ($pp \to qq, qg, gg$), which typically generate 1-prong jets. The samples used in the production of jet images and HL observables do not include contamination from pileup (i.e. multiple proton-proton collisions per beam crossing).

Resulting jets are clustered using the anti-k_t algoirthm [23] with radius parameter $R = 1.2$, using FASTJET. The dataset contains 5×10^6 events with an equal split between signal and background samples. Following the procedure in Ref. [28], each jet is pixelated into a $32 \times 32 \times 1$ grid in the rapidity-azimuth plane and a jet image is formed from the transverse momentum (p_T) deposits in each cell. The jet images are then trimmed [109], where subjets of radius $R_\mathrm{sub} = 0.2$ are discarded for $p_\mathrm{T} < 3\%$ of the original jet. Final jet selection takes jets with trimmed momentum $p_\mathrm{T}^\mathrm{trim} \in [300, 400]$ GeV within the range $|\eta| < 5.0$.

From the trimmed jet constituents, six HL jet substructure observables are constructed: the trimmed jet mass (M_jet), four ratios of energy correlation functions $\Big(C_2^{\beta=1}$, $C_2^{\beta=2}$, $D_2^{\beta=1}$, $D_2^{\beta=2}\Big)$ [24, 79, 80], and the N-subjettiness ratio $\Big(\tau_{21}^{\beta=1}\Big)$ [25, 78]. These observables are well-established in the context of boosted W classification, including studies at ATLAS [110, 111] and CMS [112]. Distribution plots for each of the six high-level observables are shown in Fig. A.1. The performance of each of these individual observables in a simple DNN is given in Table 3.1. The trimmed jet mass is the most powerful single observable, since the 80.4 GeV mass peak is a characteristic feature of boosted W bosons. The ADO calculation described in Sec. 3.2.2 can give additional insights into the nature of these traditional observables. Given in Fig. 3.2, the pairwise ADO between each of the HL observables is given. Among the HL pairs measured, the most similar decisions (i.e. the nearest to ADO = 1) are the energy correlation functions $C_2^{\beta=1}$ with $C_2^{\beta=2}$ and $D_2^{\beta=1}$ with $D_2^{\beta=2}$. This is to be expected given that these observables share a common structure for their calculation with the exception of their choice of exponent β, which controls the relative weighting of angular information within the jet constituents. These pairs also have similar AUC values, as seen in Table 3.1. This feature is also not unusual as observables with very similar information and information accessible to a ML model in a similar way will often produce comparable performance. Comparing

Observable	AUC	ADO(CNN, Obs.)
M_{jet}	0.898 ± 0.004	0.807
$C_2^{\beta=1}$	0.660 ± 0.006	0.584
$C_2^{\beta=2}$	0.604 ± 0.007	0.548
$D_2^{\beta=1}$	0.790 ± 0.005	0.743
$D_2^{\beta=2}$	0.807 ± 0.005	0.762
$\tau_2^{\beta=1}$	0.662 ± 0.006	0.600
6HL	0.9504 ± 0.0002	0.971
CNN	0.9531 ± 0.0002	1
488HL	0.9535 ± 0.0002	0.978
7HL	0.9528 ± 0.0003	0.971

Table 3.1: Classification performance of the six HL observables studied in Ref. [28], as well as a 6HL joint classifier. The six HL observables face a small but statistically significant performance gap compared to the benchmark CNN. As discussed in Sec. 3.4.1, this performance gap is bridged by a seventh feature discovered using the black box guiding strategy. An additional HL network (488HL) is trained with a large set of EFPs for comparison and is discussed in Sec. 3.5.2. Uncertainty on the AUC is computed from 1 standard deviation of 10-fold cross validation. The decision similarity (ADO) to the benchmark CNN is also given. Details of the NN architectures provided in App. A.2

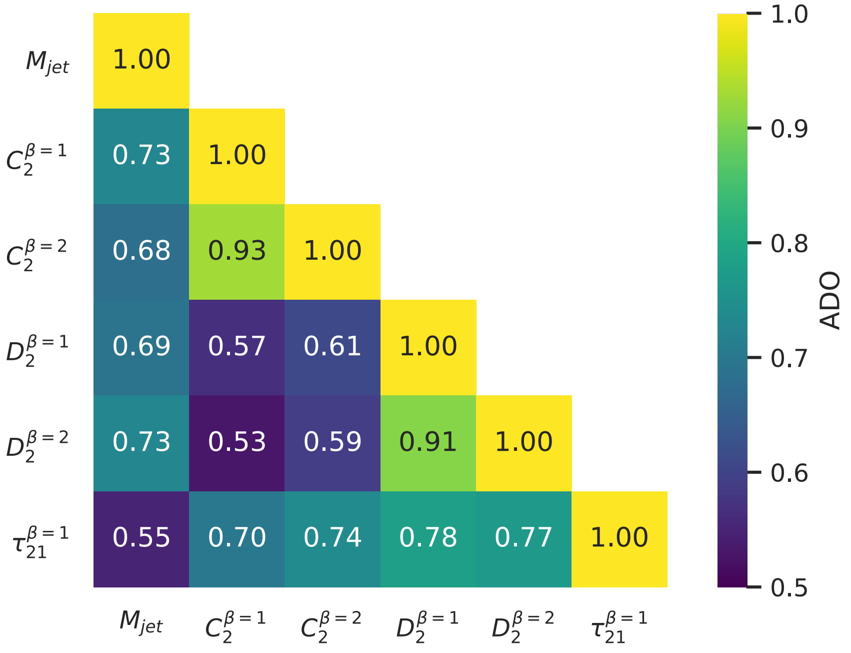


Figure 3.2: Similarity of classification decisions between the six traditional HL jet substructure observables as quantified by their relative ADO. Each feature is trained using a simple DNN with parameters given in App. A.4. A value of ADO=1 corresponds to the two networks trained on their respective observables having identical decision ordering for all signal/background pairs, while ADO=$\frac{1}{2}$ corresponds to no similarity between them. This comparison then shows a similar interpretation to that of the AUC, but with respect to the classification decisions between each other rather than to the ground truth.

observables in this way (i.e. in terms of both their performance with AUC and similarity with ADO) gives a more detailed picture about the degree of correlation between them in classification. Conversely, the pair of observables with the least similarity when used in training $\left(\text{i.e. nearest to ADO} = \frac{1}{2}\right)$ are M_{jet} with $\tau_{21}^{\beta=1}$ and C_2^{β} with D_2^{β}. In the comparison of M_{jet} with $\tau_{21}^{\beta=1}$, this is the expected result since N-subjettiness probes the degree of prong-like collimation explicitly, whereas mass is sensitive to the energies of the prongs and their relative angles. For the case of C_2^{β} and D_2^{β}, this is can be explained by the fact that the two observables have different scalings under boosts along the jet direction [80]. Its also apparent that pairs of observables that make dissimilar decisions can often be combined into more powerful joint classifiers. This can be seen in Fig. 3.3, where pairwise DNN classifiers have been trained for each combination of HL observables $(\text{NN}\,(\text{HL}_i, \text{HL}_j))$. Additional

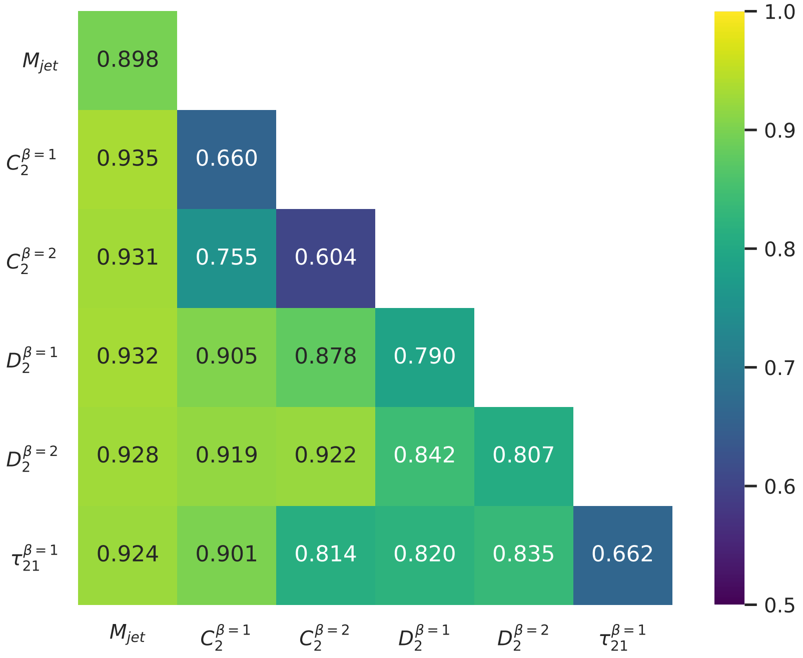


Figure 3.3: Classification performance of the six observables in Table 3.1 (on the diagonal entries) and the AUC for pairwise classification between coupled HL inputs on the off-diagonal entries when trained with a simple DNN with parameters given in App. A.4.

and more comprehensive studies of these six jet substructure observables can be found in Ref. [74] While these six engineered HL features are powerful jet substructure discriminants, they do not fully capture the information available in the original W boson tagging dataset. Using the calorimeter cells to construct a two-dimensional jet image, the performance of HL observables can be compared to powerful computer vision techniques [27, 28, 41, 43, 48, 53, 95, 113, 114]. In fact, Ref. [28] demonstrated directly that a deeply connected CNN using low-level jet images as inputs yields superior classification performance than the six HL observables combined by a BDT. The performance gain, although modest, is persistent and this makes it an ideal benchmark problem for the comparison of competing ML strategies. The CNN training on jet images was repeated with the parameters given in App. A.1 and a DNN on the 6HL combination (with the parameters given in App. A.2):

$$6\mathrm{HL} \equiv \mathrm{NN}\left(M_{\mathrm{jet}}, C_2^{\beta=1}, C_2^{\beta=2}, D_2^{\beta=1}, D_2^{\beta=2}, \tau_{21}^{\beta=1}\right) \tag{3.10}$$

A performance gap of 0.0027 ± 0.0003 in AUC is found, as reported in Table 3.1. Using the guided strategy, this gap in performance can be analyzed and understood. Specifically, has the CNN found a strategy similar to the existing HL observables or something unique? Does this gap in information indicated a minor optimization can be made but the two solutions are, otherwise, similar? Or does this difference in performance suggest a larger change in the way the HL observables (and by extension, physicists) should view the problem of jet substructure classification?

3.3.2 Energy Flow Polynomials

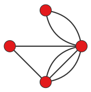
In order to map the CNN from Ref. [28] to a human readable space, it's first necessary to define the space of observables from which one can find a supplemental input relevant to the classification problem. This selection requires domain knowledge about the underlying physics of the problem as well as intuition for the kinds of information that might be missing from the existing inputs. It is also necessary that the potential observables are powerful individually given, as mentioned in Sec. 3.2.3, the black box guided strategy uses a greedy algorithm that is more sensitive to inputs which maximize ADO for individual steps in the algorithm.

The chosen set of HL observables is based on EFPs [81], which provides a large (formally infinite) set of parameterized engineered functions, inspired by previous work on energy correlation functions [24, 80, 115–118]. In the jet image representation, the EFPs are defined in terms of the momentum fraction z_i of a calorimeter cell i, as well as the pairwise angular distance between cells (i, j) with θ_{ij}. The EFPs are built in increasing levels of complexity, from simple sums over single cells to many higher-order combinations of momentum and pair-wise angles. An EFP is represented as a multigraph where:

$$\sum_{i=1}^{N} z_i \equiv \text{each node} \tag{3.11}$$

$$(\theta_{ij})^{\kappa} \equiv \text{each } k\text{-fold edge} \tag{3.12}$$

As an example, one can produces an EFP like:



$$\Bigg(\qquad \Bigg) = \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{c=1}^{N}\sum_{d=1}^{N} z_a z_b z_c z_d \theta_{ab}^3 \theta_{bc} \theta_{ac} \theta_{ad}^2. \tag{3.13}$$

From the graphical representation, it's possible to express both fully connected and disconnected graphs. For the purpose of this study, only connected graphs are considered.

Each EFP can be further modified according to two additional parameters (κ, β). The chosen value will impact the relative scaling of z_i and θ_{ij} according to,

$$z_i^{\kappa} = \left(\frac{p_{T_i}}{\sum\limits_j p_{T_j}} \right)^{\kappa}, \tag{3.14}$$

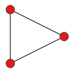
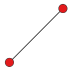
$$\theta_{ij}^{\beta} = \left(\Delta\eta_{ij}^2 + \Delta\varphi_{ij}^2 \right)^{\beta/2} \tag{3.15}$$

In this context, p_{T_i} is the transverse momentum of cell i, and η_{ij} (φ_{ij}) represents the pseudorapidity (azimuth) difference between cells i and j. Note that IRC-safe EFPs require that $\kappa = 1$, though additional examples are included into the search space that correspond to $\kappa \neq 1$ to allow for a search in a broader space of observables, as motivated by Refs. [82, 84, 85, 119]. Additionally, note that $\kappa > 0$ corresponds to Infrared safe (IR) but not Collinear (C) safe observables. Both zero and negative values of κ are included in this search in an attempt to explore different combinations of IR and C safety. The EFPs are generated using the EnergyFlow python package [120] to translate jet-image data (p_T, η, φ) into EFPs with varying graphs and choices of κ and β.

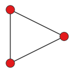
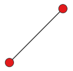
For the guided search, the EFP selections available consists of every combination of graph with (κ, β) values where $\kappa \in \left[-1, 0, \frac{1}{2}, 1, 2\right]$ and $\beta \in \left[\frac{1}{2}, 1, 2\right]$. Each of these 15 combinations of (κ, β) is then applied to the complete set of connected graphs with degree (i.e. number of edges) $d \leq 7$ along with all connected graphs with degree $d \leq 8$ with chromatic number $c = 4$ (to be defined in Sec. 3.4.2), for a total of 509 graphs. This combination of graphs and parameter choices yields a space of 7,635 EFPs from which to choose. However, due to the fact that some selections of κ and β can be swapped without any impact to the EFP values, degenerate EFPs are removed leaving a total of 7,545 unique observables.

It is important to emphasize that, although the EFP space constitutes a formally complete basis for IRC-safe jet classification, the primary goal is the pragmatic task of isolating individual observables that can map out the CNN behavior. In the ideal scenario, a CNN strategy is straightforwardly mapped into a single EFP, indicating that it can be expressed compactly in terms that can be easily understood by physicists. Failing that, it is still of value to accomplish a similar mapping using a small collection of discovered observables [103–105]. This outcome would still provide a significantly more physically meaningful interpretation of the data and reduction in data complexity when compared to the alternative low-level modeling.

In the event one is unable to make a mapping between the CNN strategy and a small number of simple EFPs, this could mean one of two things. First, it's possible that the CNN strategy simply can't be made to operate on the lower-dimensional HL space, requiring us to revisit the assumption that the HL space was sufficiently complete to capture all of the essential information for jet classification. Second, it could mean that the CNN strategy is encodable in terms of HL observables but with a more complex combination of inputs. As an example of this second option, consider the energy correlation ratios C_2 [24] and D_2 [80]. These can be written, in terms of EFPs, with the parameter $\kappa = 1$ as

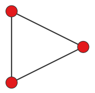



$$C_2^{(\beta)} = \frac{\left(\qquad\right)^{(\kappa=1,\beta)}}{\left[\left(\qquad\right)^{(\kappa=1,\beta)}\right]^2}, \tag{3.16}$$

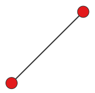



$$D_2^{(\beta)} = \frac{\left(\qquad\right)^{(\kappa=1,\beta)}}{\left[\left(\qquad\right)^{(\kappa=1,\beta)}\right]^3}, \tag{3.17}$$

where the graphs corresponds to:



$$\left(\qquad\right)^{(\kappa=1,\beta)} = \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{c=1}^{N} z_a z_b z_c \theta_{ab}^{(\beta)} \theta_{bc}^{(\beta)} \theta_{ca}^{(\beta)}, \tag{3.18}$$



$$\left(\qquad\right)^{(\kappa=1,\beta)} = \sum_{a=1}^{N}\sum_{b=1}^{N} z_a z_b \theta_{ab}^{(\beta)}. \tag{3.19}$$

The guided strategy, however, would not necessarily be able to identify these specific ratio combinations unless they were defined ahead of time. Therefore, whether or not the guided mapping is effective, one learns something about the nature of the physics problem either way.

3.4 Supplementing Existing Observables

In this section, the mapping strategy given in Sec. 3.2 is applied to find an additional HL observable to act as a supplement to the 6HL observables identified in Eq. (3.10) when used for boosted W boson classification. From Table 3.1, the small performance gap remains between the HL observables and the CNN trained on LL inputs. Using that CNN as a guide for selecting from the EFP space, the goal is to bridge that performance gap and recover any missing information lost in the simpler and lower-dimensionality feature space.

3.4.1 Black Box Guiding

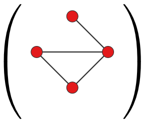
The first step in the black box guided strategy from Sec. 3.2 is to identify a subset of signal/background pairs that are differently ordered by the CNN and combination of 6HL inputs:

$$X_6 = \{(x, x') \,|\mathrm{DO}\,(\mathrm{CNN}, 6\mathrm{HL}, x, x') = 0\} \tag{3.20}$$

Although the dataset, when counted as signal/background pairs, has $6.25 \times 10^1 2$ samples to choose from, a subset of 5×10^7 is used to improve processing efficiency. From this subset, the pairs which are different isolated, X_6 is compared to each EFP according to their ADO:

$$\mathrm{HL}_7^{\text{black box}} = \underset{\mathrm{HL}\in\mathrm{EFP}}{\mathrm{argmax}}\,\mathrm{ADO}\,(\mathrm{CNN}, \mathrm{HL})_{X_6} \tag{3.21}$$

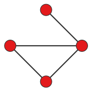
The results for this iteration are given in the first row of Table 3.2. The optimal EFP according to the ADO measured against the CNN in the X_6 subspace is:



$$^{\left(\kappa=2,\beta=\frac{1}{2}\right)} = \sum_{a,b,c,d=1}^{N} (z_a z_b z_c z_d)^2 \sqrt{\theta_{ab}\theta_{bc}\theta_{ac}\theta_{ad}}. \tag{3.22}$$

On its own, the EFP given in Eq. (3.22) only has a performance value of AUC=0.8031. However, when used as the seventh feature of an NN along with the original 6HL inputs,

$$7\mathrm{HL}_{\text{black box}} \equiv \mathrm{NN}\left[M_{\text{jet}}, \ldots, \tau_2^{\beta=1}, \left(\qquad \right)^{\left(\kappa=2,\beta=\frac{1}{2}\right)}\right], \tag{3.23}$$



the resulting 7HL set of inputs successfully closes the performance gap with the CNN by achieving AUC $= 0.9528 \pm 0.0003$, as given in Table 3.2. Interestingly, this occurs despite the ADO between $7\mathrm{HL}_{\text{black box}}$ and the CNN is only 0.971, suggesting that the two networks still make different decisions approximately 3% of the time. Although the black box guided process has provided an observable that closes the AUC performance gap, the remaining ADO gap implies that there is additional information not being captured.

The remaining rows shown in Table 3.2 give results for other choices of EFP from the set of 7,545 EFPs available sorted by ADO. The statistical uncertainties on the ADO are large enough that the precise ranking is not so meaningful, thoguh the overall trends are. One noteworthy feature is that many observables have a similar ADO to Eq. (3.22), but that they often feature $\kappa = 2$ and $\beta = \frac{1}{2}$ as their choice of parameters. A choice of $\kappa = 2$ corresponds to an IRC-unsafe EFP and this suggests that IRC-unsafe information may be particularly valuable for this supplemental input (though perhaps not uniquely so) for mapping the CNN strategy. Similarly, the choice of $\beta = \frac{1}{2}$ suggests the importance of probing small angle behavior. Other IRC-unsafe factors appear in these results as EFPs with $\kappa = 0$ and $\kappa = -1$ also perform quite well, especially with the constituent multiplicity appearing third on the list.

The best performing IRC-safe ($\kappa = 1$) choice doesn't appear until rank 5531 and incorporating this EFP as the supplemental observable doesn't result in the performance gap closing. Specifically, the EFPs in Eq. (3.18) and Eq. (3.19) with $\kappa = 1$ have a relatively small ADO in the X_6 subspace, never getting above 0.5279. This result is reasonable given that the

Rank	EFP	κ	β	c	$\mathrm{ADO(EFP, CNN)}_{X_6}$	AUC(EFP)	$\mathrm{ADO(6HL + EFP, CNN)}_{X_{\mathrm{all}}}$	AUC(6HL + EFP)
1		2	$\frac{1}{2}$	3	0.6207	0.8031	0.9714	0.9528 ± 0.0003
2		2	$\frac{1}{2}$	3	0.6205	0.8203	0.9714	0.9524
3		0	–	1	0.6205	0.6737	0.9715	0.9525
4		2	$\frac{1}{2}$	3	0.6199	0.8301	0.9715	0.9527
5		2	$\frac{1}{2}$	3	0.6197	0.8290	0.9714	0.9527
6		2	$\frac{1}{2}$	3	0.6196	0.8251	0.9715	0.9522
7		0	$\frac{1}{2}$	2	0.6187	0.7511	0.9715	0.9526
8		2	$\frac{1}{2}$	3	0.6184	0.8257	0.9712	0.9527
9		2	$\frac{1}{2}$	3	0.6182	0.8090	0.9714	0.9527
10		2	$\frac{1}{2}$	3	0.6180	0.8314	0.9714	0.9526
60		0	1	2	0.6163	0.7194	0.9715	0.9525
341		−1	$\frac{1}{2}$	4	0.6142	0.6286	0.9714	0.9509
589		0	2	2	0.6109	0.7579	0.9714	0.9523
3106		−1	–	1	0.5891	0.5882	0.9714	0.9510
3519		$\frac{1}{2}$	$\frac{1}{2}$	2	0.5664	0.7698	0.9715	0.9524
3521		$\frac{1}{2}$	–	1	0.5663	0.7093	0.9714	0.9522
5531		1	2	1	0.5290	0.7454	0.9714	0.9507
5554		1	$\frac{1}{2}$	2	0.5279	0.8210	0.9713	0.9505
5610		2	–	1	0.5245	0.7117	0.9714	0.9507
5657		1	1	3	0.5224	0.8257	0.9712	0.9506
5793		1	1	2	0.5191	0.8640	0.9714	0.9505
6052		1	2	3	0.5153	0.8500	0.9716	0.9504
7438		1	2	2	0.5011	0.8835	0.9716	0.9506

Table 3.2: A selection of EFPs, sorted by their similarity with the CNN and evaluated using the ADO in the differently-ordered subspace X_6. This corresponds to one iteration of the black box guiding method depicted in Fig. 3.1. Past the top 10, EFPs are shown when if they correspond to a dot graph, appear in the C_2/D_2 observables from Eq. (3.16) or Eq. (3.17), or have the highest ADO among graphs with a given value of κ, β, or chromatic number (c).

information, although potentially useful, is already sufficiently captured by the presence of C_2 and D_2 combinations in the original 6HL input features.

For completeness, distributions for the top three EFPs from Table 3.2 are given in Fig. 3.4 with both their complete form X_{all} and in the differently ordered subspace X_6. The first two observables show strong separation between signal and background when measured in the full space, as expected given their AUC is around 0.8. The third observable, constituent multiplicity, is a relatively poor discriminant by itself. When restricted to the X_6 subspace, there is only a modest residual separation power for these three observables. Despite this, that small separation power proves to be sufficient to bridge the performance gap with the CNN.

3.4.2 Physics Interpretation

Interpreting these results in a physics context, the first observation to make is that the κ-augmented EFP space is sufficiently comprehensive to close the performance gap between 6HL and the CNN. However, had attention been restricted to just the IRC-safe EFPs, this result would not have been the case. As mentioned in the previous section, the top ranked $\kappa = 1$ EFP included with 6HL was only capable of reaching AUC $= 0.9507$. Thus, IRC-unsafe information seems to be essential to closing the performance gap. It's worth noting that, as discussed in Ref. [121], CNNs are formally IRC safe. However, in order to map this IRC-safe behavior to the EFPs would require very high-point correlators which with, in principle, as many nodes as pixels in the original jet image. With IRC-unsafe EFPs, a similar matching to the CNN decision surface can be made with low-point correlators.

Fascinatingly, $\kappa = 2$ appears prominently in the top ten EFPs, though in a different form than previously considered in the literature. A key feature of $\kappa = 2$ EFPs is that they are more sensitive to higher energy particles. Looking through the top $\kappa = 2$ observables

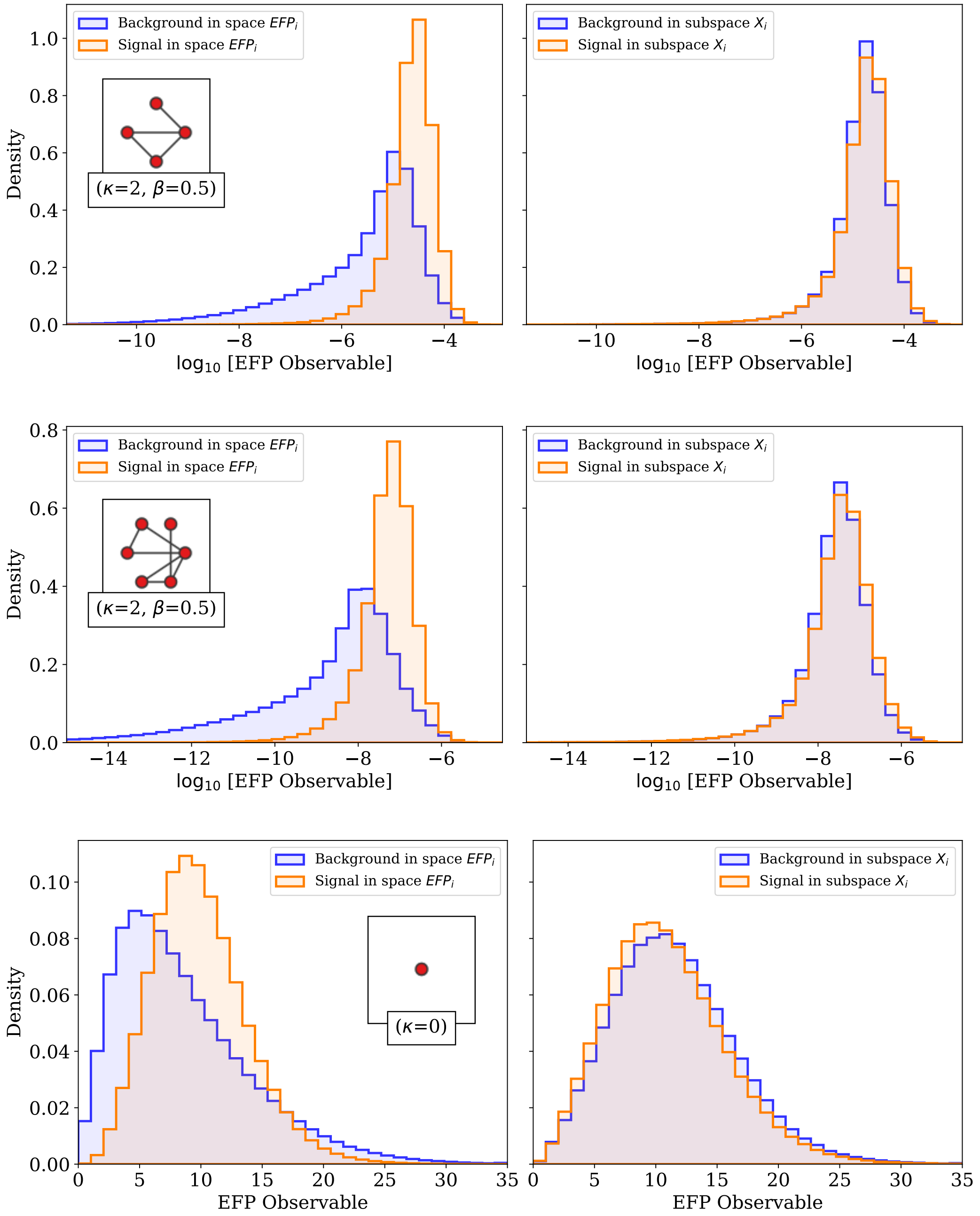


Figure 3.4: Top three EFPs selected from the black box guided search when seeking a seventh HL observable; These three observables correspond to the first three rows of Table 3.2. EFP distributions are shown with their signal and background separation for the entire dataset (left) and for the differently ordered subset X_6 (right).The top two observables, although not identical, have similar functional forms up to an overall rescaling. The third observable is the jet constituent multiplicity.

in Table 3.2, they have the common feature of sharing a chromatic number $c = 3$. The chromatic number is defined as the minimum number of colors needed to decorate the nodes of a graph such that no edge connects same-color nodes. If an EFP has chromatic number c, then it is only non-zero if the jet has at least c distinct particle directions. This makes chromatic number an effective probe for deviations from $(c-1)$-prong substructure. The $\kappa = 2$ and $c = 3$ EFPs found by the guided strategy therefore probe IRC-unsafe deviations from 2-prong substructure 9as one might expect for boosted W tagging), with a particular emphasis on the higher energy particles inside the jet.

By contrast, the only $\kappa = 2$ observable that has received any significant attention in the literature is p_T^D [119]. In the form of an EFP, the p_T^D is represented by a chromatic $c = 1$ graph with no edges:

$$\left(\quad \bullet \quad \right)^{(\kappa=2)} = \sum_{a=1}^{N} z_a^2. \tag{3.24}$$

In this instance, the choice of p_T^D in the guided search appears at rank 5610 when sorted by ADO. Evidently, generic IRC-unsafe information is not, by itself, useful for boosted W boson classification, but must be paired with the correct angular dependence to highlight the physics of interest. It is interesting that, in this instance, $\beta = \frac{1}{2}$ is the preferred angular exponent, since this choice appeared previously in the context of the Les Houches angularity for quark/gluon discrimination [85].

There are also $\kappa = 0$ observables in the top ten EFPs, including the well known constituent multiplicity

$$\left(\quad \bullet \quad \right)^{(\kappa=0)} = \sum_{a=1}^{N} 1. \tag{3.25}$$

The fact that a $\kappa = 0$ and $c = 1$ observable gives nearly identical performance to a class of $\kappa = 2$ and $c = 3$ observables is a surprising result. One interpretation for this is that it represents two complementary approaches to solving this jet classification task. On the one hand, boosted W bosons are 2-prong objects, so one expects to see chromatic $c = 3$ EFPs as the most relevant. Indeed, the numerators of Eq. (3.16) and Eq. (3.17) are $c = 3$ graphs that probe 2-prong substructure, which is part of the original motivation for the C_2 and D_2 observables. On the other hand, the background quark and gluon jets are 1-prong sensitive objects, and constituent multiplicity is well-known to be a powerful quark/gluon discriminant [85] (though sensitive to detector effects [122])

The next $\kappa = 0$ observable on the list has $c = 2$ and $\beta = \frac{1}{2}$, which is an IRC-unsafe probe of 1-prong substructure with an emphasis on collinear physics, which should also be an effective quark/gluon discriminant. This suggests an improvement to classification performance either with a refined probe of the W boson signal or a refined probe of the quark/gluon background sample, which happens to have a similar effect on the decision boundary.

In summary, by translating an ML strategy into a human-readable space, a new set of jet substructure observables have been identified which do not currently exist in the literature for boosted W boson studies. This also motivates further study into the area of IRC-unsafe observables, particularly those with parameter $\kappa = 2$. In Sec. 3.6, the implications towards future work in jet substructure observables is discussed.

3.5 Iteratively Mapping from Minimal Features

In the previous section, the use of a black box guided search was used to supplement the existing set of 6HL features to bridge the performance gap between it and the CNN. This jet substructure case study is unusual, however, in that it benefits from an existing and mature literature with many known observables. In many other studies of interest, one may find themselves starting from a minimal starting point and a need to build a set of HL observables from scratch.

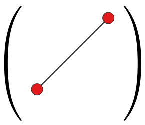
In this section, a black box guided search is performed with a minimal initial set of observables. Starting with transverse momentum p_{T} and jet mass M_{jet}, the guided search is used to build the remaining set of features purely with EFPs. Ultimately, the following sections show that even starting from a sparser initial point, a set of 7 physics-motivated is sufficient to match the performance of the CNN and the 7HL combination (which did not originally include p_{T}). However, the specific solution found through this approach will prove to be different from that in the previous sections in interesting ways.

This section is then ended with a comparison to a brute force approach in order to establish the beneficial computational efficiency inherent to the guided approach and a comparison to a label-guided search, demonstrating the performance improvements inherent to a low-level guided method.

3.5.1 Black Box Guiding

In this section, the black box guided approach used in Sec. 3.4.1 is repeated but starting with a smaller subset of initial features, p_{T} and M_{jet}. The choice for this minimal set is motivated, in the case of M_{jet}, by the simple fact that the W boson mass of 80.4 GeV is an important and obvious feature for the discrimination of boosted W bosons. As such, this works as a reasonable initial feature that would not require much prior knowledge or existing literature to select. The inclusion of p_{T} is made due to the fact that the EFPs and their inclusion of z_i given in Eq. (3.14) are effectively dimensionless. Therefore, the initial inputs need at least one HL observable with dimension scaled relative to the jet p_{T} in order to capture the W boson mass peak, and both the p_{T} and M_{jet} will contribute to this. Additionally, both observables are ubiquitous jet observables appearing in the majority of collider studies.

Starting with this minimal set of inputs creates a streamlined selection of EFPs. It's worth noting that, in theory, one could start with the exclusion of M_{jet} and attempt to re-derive it through the guided process in the EFP space. There exists an approximate equivalent to M_{jet} in the EFP observables in the form of:



$$^{(\kappa=1,\beta=2)} \approx \frac{M_{\text{jet}}^2}{p_{\text{T}}^2}. \tag{3.26}$$

However, because of the choice of θ_{ij} in Eq. (3.15), the representation in Eq. (3.26) is only approximately true and so the explicit addition of the mass information bypasses the need to produce multiple observables to properly capture this feature (and use it as a dimensional scale). A small study was done in which the guided search was attempted with just p_{T} or just M_{jet} to compare effectiveness with the use of both in the initial observables. Although both approaches could successfully recover the CNN performance, the selected EFPs in those processes tended to be repetitively "mass-like". Using both p_{T} and M_{jet}, in contrast, provided varied and unique observables.

Starting with a trained NN on just p_{T} and M_{jet}, the initial state network becomes:

$$\text{HLN}_0 \equiv \text{NN}\left(p_{\text{T}}, M_{\text{jet}}\right). \tag{3.27}$$

This gives a performance of AUC = 0.9119, which is considerably below the CNN performance for boosted W boson tagging. Restricting attention to the subset of events which are differently ordered relative to the CNN:

$$X_0 = \{(x, x') \mid \text{DO}\left(\text{CNN}, \text{HLN}_0, x, x'\right) = 0\} \tag{3.28}$$

The ADO between the CNN and HLN_0 is 0.9150, so X_0 contains 8.5% of the original X_{all} sample. Again, taking a random subset of 5×10^7 pairs in X_0, the differently ordered pairs are isolated and the EFP with the maximum ADO with the CNN is measured:

$$\mathrm{EFP}_n = \underset{\mathrm{EFP}\in\mathcal{S}}{\operatorname{argmax}} \, \mathrm{ADO}\left(\mathrm{CNN}, \mathrm{EFP}\right) X_{n-1}. \tag{3.29}$$

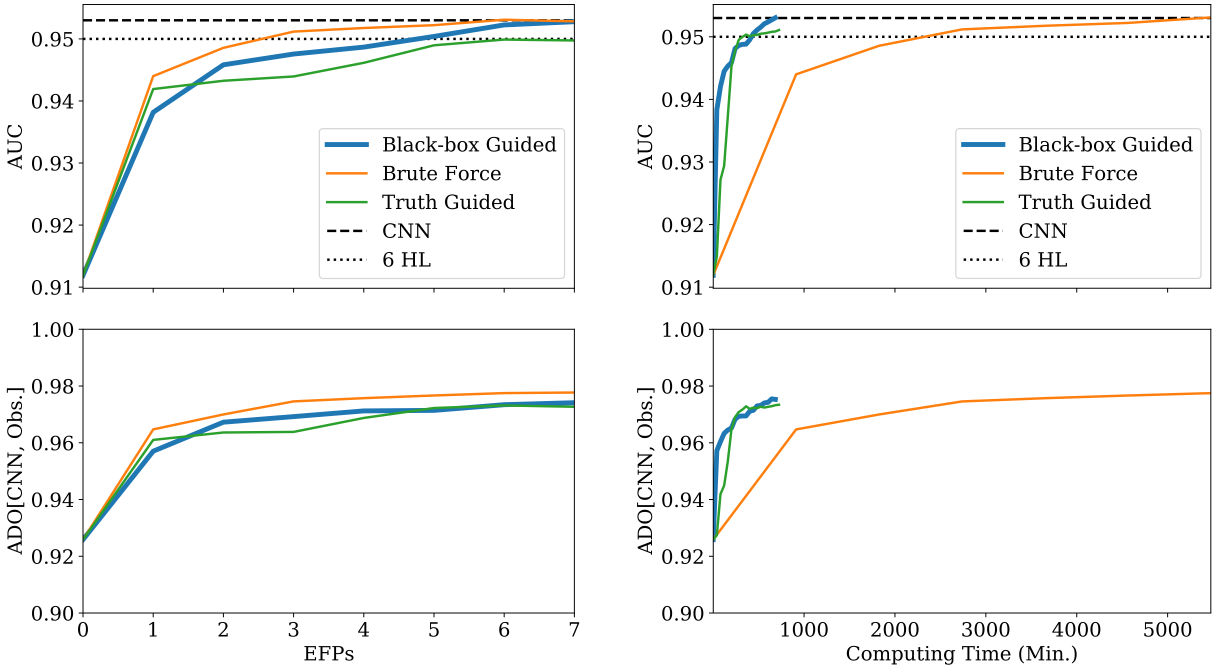
A joint classifier is trained with the included EFPs

$$\mathrm{HLN}_n \equiv \mathrm{NN}\left(p_{\mathrm{T}}, M_{\mathrm{jet}}, \mathrm{EFP}_1, \ldots, \mathrm{EFP}_n\right). \tag{3.30}$$

This allows for the identification of the remaining differently ordered subset of events.

$$X_n = \left\{(x, x') \,|\, \mathrm{DO}\left(\mathrm{CNN}, \mathrm{HLN}_n, x, x'\right) = 0\right\}, \tag{3.31}$$

where in each iteration a new random subset of 5×10^7 pairs is utilized. The primary computational cost in this procedure is in the training of the joint classifier which occurs n times for each iteration performed in Eq. (3.30). The AUC and ADO values for this minimal black box guided example are given in Fig. 3.5a and charted according for each iteration through the guided search. Additionally, the computational cost in terms of computing time is given in Fig. 3.5b. More details about the selected EFPs are given in Table 3.3. Note that by the fifth iteration, the AUC performance has matched that of the 6HL combination and, by the seventh iteration, the performance matches that of the CNN with an ADO = 0.974. This indicates that the guided search starting with p_{T} and M_{jet} has found a more similar solution to the CNN than the previous $7\mathrm{HL}_{blackbox}$ guided strategy. Considering this started from a minimal set of features, it is not surprising that the EFPs selected are qualitatively different from those chosen in Sec. 3.4. The physical interpretation of these EFPs is presented in Sec. 3.5.4.



(a) Maximum performance

(b) Cumulative computing time

Figure 3.5: Performance of the black box guided search strategy (left) and computing time comparison (right) for the mapping of a CNN solution into human-interpretable observables. Here, the process starts from just the basic jet features p_{T} and M_{jet} and iteratively add one EFP at a time. The performance is shown shown in terms of AUC (top) and ADO (bottom) as a function of the scan number. The performance of a brute force scan of the EFP space (Sec. 3.5.2) and a truth-label guided search (Sec. 3.5.3) are also shown. For reference, the performance of the CNN and of the existing 6HL features are indicated by horizontal lines.

n	EFP	κ	β	c	$\mathrm{ADO(EFP, CNN)}_{X_{n-1}}$	AUC(EFP)	$\mathrm{ADO\,(HLN}_n\mathrm{, CNN)}_{X_{\mathrm{all}}}$	$\mathrm{AUC\,(HLN}_n)$
0	$M_{\mathrm{jet}} + p_{\mathrm{T}}$	–	–	–	–	–	0.9259	0.9119
1		2	$\frac{1}{2}$	2	0.8144	0.8190	0.9570	0.9382
2		0	2	2	0.6377	0.8106	0.9673	0.9458
3		0	–	1	0.5460	0.6737	0.9692	0.9476
4		1	$\frac{1}{2}$	2	0.5274	0.8464	0.9712	0.9487
5		−1	–	1	0.5450	0.5882	0.9714	0.9504
6		1	$\frac{1}{2}$	4	0.5382	0.7678	0.9734	0.9523
7		−1	$\frac{1}{2}$	2	0.5561	0.5957	0.9741	0.9528

Table 3.3: The EFPs selected during each iteration (n) of the black box guiding strategy beginning from HLN_0, which uses just p_{T} and M_{jet}. For each iteration, the selected EFP is the one with the largest ADO with the CNN in the differently-ordered subspace X_{n-1}.

3.5.2 Comparison to Brute Force Search

An alternative approach to maximizing the ADO is to perform a *brute force* search through the space of EFPs to find a set that maximally matches the decisions of the CNN. This is a much more computationally expensive search than the black box guided strategy, but it has the potential to converge to a smaller set of EFPs if there is useful pairwise information between observables. In an absolute brute force search, one could construct the set of all possible combinations of EFPs and compare the ADO of every element in the set to the CNN. Given the size of the EFP set, this becomes an intractable approach. Instead, a greedy algorithm which incrementally builds the EFP set by brute force is done as a comparison. This approach, while still computationally expensive, is tractable.

Starting again with p_{T} and M_{jet}, a joint classifier is trained with the inclusion of each of the EFPs as an input:

$$\mathrm{NN}\left(p_{\mathrm{T}}, M_{\mathrm{jet}}, \mathrm{EFP}\right). \tag{3.32}$$

The EFP that yields the largest ADO with the CNN, evaluated on the full training set, is selected as the first EFP

$$\mathrm{EFP}_1 = \underset{\mathrm{EFP}\in\mathcal{S}}{\mathrm{argmax}}\,\mathrm{ADO}\left(\mathrm{CNN}, \mathrm{NN}\left(p_\mathrm{T}, M_\mathrm{jet}, \mathrm{EFP}\right)\right)_{X_\mathrm{all}} \tag{3.33}$$

This procedure is then repeated for many iterations, each time selecting a new EFP according to the maximum ADO with the CNN when testing with each EFP as a feature

$$\mathrm{EFP}_n = \underset{\mathrm{EFP}\in\mathcal{S}}{\mathrm{argmax}}\,\mathrm{ADO}\left(\mathrm{CNN}, \mathrm{NN}\left(p_\mathrm{T}, M_\mathrm{jet}, \ldots, \mathrm{EFP}_{n-1}, \mathrm{EFP}_n\right)\right)_{X_\mathrm{all}} \tag{3.34}$$

The key difference between this brute force search and the black box guided strategy is that the joint classifier is trained before evaluating the ADO, and the ADO is evaluated on the full training set instead of just a differently-ordered subset

The primary computational cost in the brute force approach comes from the training of a joint classifier for each of the candidate EFPs (n=7,545) for every iteration of the process. For computationally efficiency, a subset of the original pool of EFPs is selected. Reducing the EFPs to those with dimension $d \leq 5$ (54 graphs) with choices of $\kappa \in \left[\frac{1}{2}, 1, 2\right]$ and $\beta \in \left[\frac{1}{2}, 1, 2\right]$, the brute force can be performed on a more manageable 486 total candidate EFPs.

Results for the brute force search are given in Fig. 3.5b with a comparison of the AUC and ADO measured against the benchmark CNN. In the first few iterations, the AUC and ADO values are higher than for the black box guiding, achieving a comparable performance to the original 6HL result after the inclusion of a third EFP. The brute force process continues until it matches the CNN after 6 EFPs (for a total of 8 HL inputs). As one would expect, the brute force approach performs at least as well as the guided search given that it is involves trying effectively every combination of EFP one at a time. This computational cost, however, must be weighed against the marginal decrease in the number of EFPs required to match the CNN as well as the need to restrict the input space prior to exploring the performance. As shown in Fig. 3.5b, the brute force approach does not complete even a single iteration of the algorithm before the guided approaches have converged on a solution.

Finally, for completeness, al alternative to the brute force approach is considered in which a network is trained using every EFP (from the n=486 subset) in a single classifier. The performance of "488HL" is given in Table 3.1, with marginally better performance than the CNN. This indicates that the EFPs are effectively a complete basis for this task.

3.5.3 Comparison to Truth-Label Guiding

In the black box guided strategy, the CNN and the ADO similarity metrics are auxiliary tools to help identify a set of EFPs that maximize the classification performance. Assuming the EFP space is complete and labelled data exists, one could dispense with the CNN entirely and simply search the space of EFPs for the most performant combination of inputs according to a models AUC, in an approach similar to Ref. [123]. In comparison to the intractable brute force search, a *truth guided* search is done in which EFP selection is guided by the CNN according to its AUC using truth labels.

Analogously to decision ordering in Eq. (3.2), a *truth ordering* (TO) metric can be defined by pairs of signal/background samples x and x' and a decision function f:

$$\mathrm{TO}\left(f, x,' \right) = \Theta\left(f(x) - f\left(x'\right)\right), \tag{3.35}$$

where a value of TO = 1 corresponds to f correctly ordering the points and TO = 0 corresponding to inverted ordering. Starting with the minimal feature set of p_{T} and M_{jet}, a differently ordered subset is isolated from the training events:

$$Y_0 = \{(x, x')\, | \mathrm{TO}\, (\mathrm{HLN}_0, x, x') = 0\} \tag{3.36}$$

In each iteration, the EFP with the highest AUC in the incorrectly-ordered subspace is found,

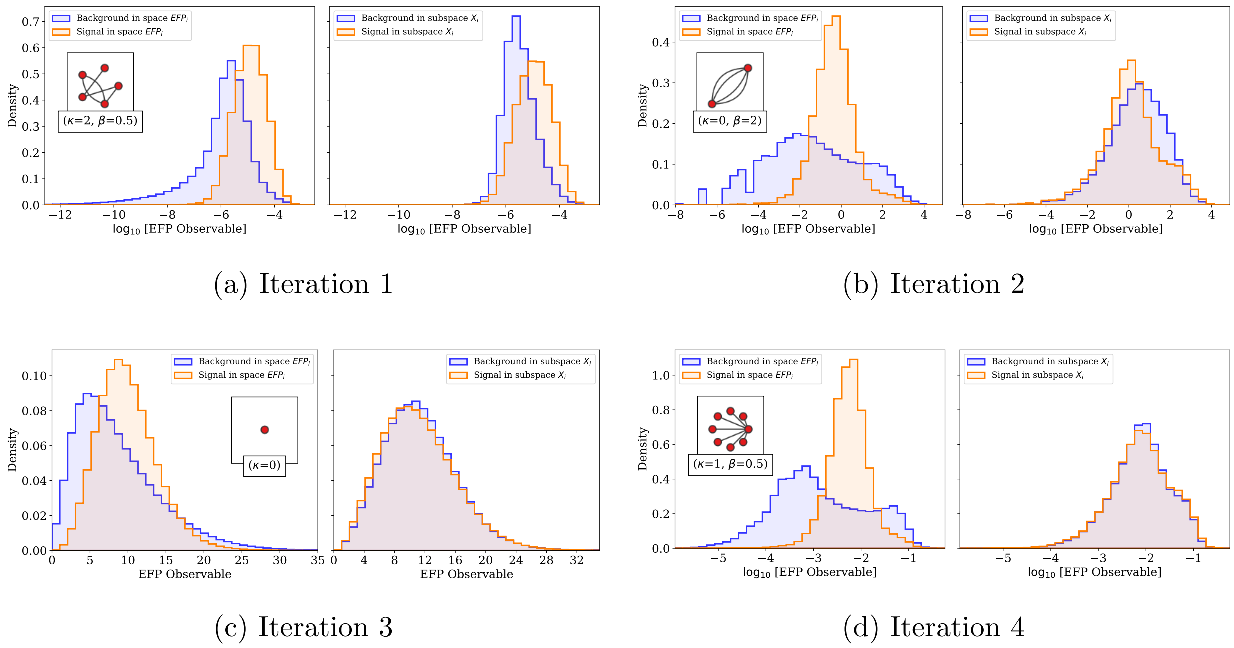
$$\mathrm{EFP}_n = \underset{\mathrm{EFP} \in \mathcal{S}}{\operatorname{argmax}}\, \mathrm{AUC}\, (\mathrm{EFP})_{Y_{n-1}}\,, \tag{3.37}$$

a new joint classifier is constructed, $\mathrm{HLN}_n \equiv \mathrm{HLN}_0 + n\mathrm{EFP}$, and the next incorrectly-ordered subset of events is isolated

$$Y_n = \{(x, x')\, | \mathrm{TO}\, (\mathrm{HLN}_n, x, x') = 0\} \tag{3.38}$$

Note that this procedure and the use of truth labels means the EFP selections are made independently from the CNN.

Results for the truth-label guided search are shown in Fig. 3.5a in terms of their performance (AUC) and ADO with the CNN. In the first iteration, classification performance is better than in the black box guided search. This makes sense given that the label guided method is trying to optimize for performance directly. After 7 iterations, though, the classification performance never rises above AUC = 0.951. As mentioned in Sec. 3.2.3, isolating the incorrectly-ordered pairs turns out to be counter productive since some of these pairs could never be ordered correctly even by the optimal classifier. This emphasizes the value of using the ADO relative to an already trained and optimized network as it focuses attention on event pairs that have a chance of being correctly ordered.



(a) Iteration 1

(b) Iteration 2

(c) Iteration 3

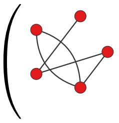
(d) Iteration 4

Figure 3.6: First four EFPs selected by the black box guided search when beginning with a minimal set of HL observables (p_{T} and M_{jet})

3.5.4 Physics Interpretation

By translating the CNN into a space of physically-motivated observables, one can gain physical insight into the observables used in the classification decision. In particular, the first few observables in Table 3.3 give us a glimpse at a possible alternative history for the field of jet substructure, if combinations like C_2 and D_2 had not been previously identified. Distributions of the EFPs found in the first four iterations are shown in Fig. 3.6.

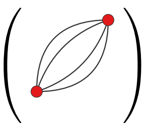
After p_{T} and M_{jet}, the first EFP selected by the black box guided strategy is:



$$\left(\qquad \right)^{\left(\kappa=2,\beta=\frac{1}{2}\right)} = \sum_{a,\ldots,e=1}^{N} \left(z_a z_b z_c z_d z_e\right)^2 \theta_{bd}\sqrt{\theta_{ab}\theta_{ac}\theta_{ce}}. \tag{3.39}$$

The fact that a $\kappa = 2$ observable shows up early in the iterative procedure bolsters the evidence from Sec. 3.4.1 that these kinds of observables are important for mapping the CNN strategy. This is a chromatic number $c = 2$ graph, so just like jet mass, it probes deviations from 1-prong substructure. However, it uses a 5-point correlator (unlike mass which is a 2-point correlator) and it uses the $\beta = \frac{1}{2}$ angular exponent (unlike mass which uses $\beta = 2$). Putting these together, Eq. (3.39) is an IRC-unsafe probe of hard, small-angle radiation.

The second EFP is also IRC unsafe and also corresponds to a $c = 2$ graph:

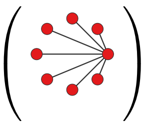


$$\left(\quad \right)^{(\kappa=0,\beta=2)} = \sum_{a,b=1}^{N} \theta_{ab}^{8}. \tag{3.40}$$

Here, though, $\kappa = 0$ and $\beta = 2$ are found, which is a probe of soft, wide-angle radiation. It is interesting that the black box guided strategy selects these two complementary $c = 2$ observable in the first two iterations, indicating the importance of 1-prong substructure probes even if the goal is to identify 2-prong boosted W bosons.

The third EFP is constituent multiplicity, as seen before in Eq. (3.25), which reinforces the idea that controlling the composition of the quark/gluon background is important for W tagging. These three observables, together with p_{T} and M_{jet}, yield an AUC of 0.9476. This is not as good as the 6HL combination, but still quite encouraging given that the black box guided strategy did not have any information about the ratio structures used to construct C_2 and D_2.

The main surprise from this study is that IRC-safe information was not selected by the black box guided search until the fourth iteration:



$$\left(\quad \right)^{\left(\kappa=1,\beta=\frac{1}{2}\right)} = \sum_{a,\ldots,h=1}^{N} z_a \cdots z_h \sqrt{\theta_{ab}\theta_{ac}\theta_{ad}\theta_{ae}\theta_{af}\theta_{ag}\theta_{ah}} \tag{3.41}$$

Moreover, it is a $c = 2$ graph, so still a probe of 1-prong substructure. Only in interaction six are there higher chromatic number graphs, but the guided search skips over the C_2/D_2-like graphs with $c = 3$ and goes straight to $c = 4$. The black box guided strategy has identified a very different strategy for boosted W boson tagging that nevertheless matches the 6HL combination with a comparable number of observables.

One interpretation of this result is that it simply reflects the “entropy” of the HL space. There are 4 times as many IRC-unsafe observables in the HL collection than IRC-safe ones, so just by random chance, one expects to see more unsafe observables in the scan. Indeed, there are IRC-safe observables that are highly ranked in the first three iterations, just not at the top of the list. Another interpretation is that the black box guided strategy is teaching us that IRC-unsafe information is more relevant for boosted W tagging than one might naively think. A related observation was made in Ref. [101], which introduced a color ring observable to identify color-singlet configurations. Intriguingly, when restricted to three particles, the angular structure of Eq. (3.39) defines similar decision boundaries to the color ring. Either way, by searching through a large space of HL observables in a systematic way, the black box guided strategy has given us a new perspective on an old problem in a human-readable format.

3.6 Discussion

In this chapter, a new technique has been proposed for mapping an ML solution into a space of human-interpretable observables. The guided strategies mitigates some of the well-founded concerns about black box approaches, while still allowing one to capitalize on the black box performance to efficiently guide the selections of HL observables. The end result is a set of HL observables that have a more direct physical interpretation and allow for a more transparent treatment of systematic uncertainties.

In the jet substructure case study, the black box guided strategy was shown to isolate information that is not captured by previous HL representations. Remarkably, only a single observable was needed to close the performance gap identified in Ref. [28], nearly duplicating the CNN strategy with a low-dimensional input representation. Beginning from a minimal set of basic jet observables (p_T and M_jet), the CNN behavior was condensed to a small set of EFP observables which reproduce its performance and very nearly match its decisions. It would be interesting to study the utility of the EFPs in more complicated contexts, such as event-wide classification tasks.

Interestingly, the structure of the selected EFPs differ in qualitative ways from the C_2 and D_2 jet substructure observables custom designed for boosted W boson classification. While these previous observables are based on fully connected graphs, the guided strategy picked out multi-node EFP graphs with relatively low chromatic number. While these previous observables use the IRC-safe choice of $\kappa = 1$, the guided strategy emphasized the importance of unsafe $\kappa = 2$ observables, particularly ones with non-trivial angular dependence. This motivates further physics studies of these exotic EFPs. It is worth emphasizing that these new observables were only identified because of considerations from a sufficiently large space of HL observables. There may be other hidden organizing principles to exploit for jet substructure studies, which motivates the construction of alternative sets of observables based on different physical principles than the EFPs. In particular, this search does not capitalize on the power counting and scaling properties of ratio/product observables [80, 118, 123–125], which may reveal more efficient HL observables for jet classification. It may also be beneficial to leverage first-principles knowledge about signal/background likelihood ratios [101, 126–131] to identify promising HL observables.

The informational gap in this benchmark problem could be closed using a single HL observable, suggesting that the CNN strategy was not relying on subtle correlations among the low-level features, but rather exploiting information encodable into a $\kappa = 2$ EFP. Thus, instead of a purely performance-oriented approach, it's important to adopt a strategy of using deep networks to establish performance benchmarks, but always seek to translate ML strategies into a more tractable space of well-motivated physical observables. If this proves to be impossible or impractical, it might be that the ML approach really is identifying genuinely new information, or more likely, that the space of physical observables needs to be augmented or optimized.

Chapter 4

Electron Identification with Interpretable Learning

4.1 Introduction

The production of electrons in high-energy collisions provides an essential view on precision studies in the SM [132, 133] and acts as a source of information for searches into new physics [134, 135]. A reliable method for identifying electrons in the presence of background data which mimics their characteristic signatures becomes a critical ML classification task.

The ATLAS detector tracks charged particles, like electrons, and features a dedicated electromagnetic calorimeter (ECal) for the measurement of their energies. The primary source of background noise masking signal events comes from the production of hadronic jets, which can obscure information as they deposit energy in both the ECal and hadronic calorimeter (HCal) or cause false positives from small fluctuations that mimic the presence of an electron. The ECal (and to a lesser extent the HCal) are designed with finally segmented calorimeter cells to allow for higher fidelity measurements. This design choice has the consequence of producing extremely high-dimensional data which is difficult to analyze directly. In an effort similar to that discussed in Sec. 3.3, a robust literature [136–138] has produced methods for converting raw detector measurements into high-level (HL) features designed to highlight electron signals and suppress background. This discovery motivates a similar set of questions to those originally posed in Sec. 3. Namely, how does the performance of electron classification in the case of LL data trained using computer vision techniques compare to the HL and physics-motivated features? If a similar performance gap is seen between the LL and HL forms of the data, can that information be recovered and translated into a HL and physically meaningful format?

In this chapter, a deeply connected convolutional neural network is given the task of distinguishing between electrons and hadronic jets through analysis of both ECal and HCal based jet images. This LL model sets a performance benchmark and is compared to the performance of a traditional set of HL inputs used for electron classification. Upon finding that the jet image networks outperform the HL input feature space, a guided search is performed to isolate EFP features as guided by both the ECal and HCal CNN. This allows for the creation of a new set of HL observables which are capable of matching the CNN performance while remaining as simple and interpretable 1-dimensional inputs.

4.2 Dataset and Generation

The HL and LL features used in this analysis comes from a simulated dataset created with publicly available fast simulation tools [108]. Although simulated samples do not generally match the same fidelity of those generated in a full simulation [139], the simulation is modified to more closely match the natural resolution of the ATLAS ECal and HCal. This provides a sufficiently realistic proof-of-principle analysis for the comparison of HL and LL features generated in the real detectors used at the LHC. The primary focus of this work is to compare the training and performance techniques on an equal footing. While the numerical results found here won't likely be perfectly reproducible in a realistic scenario, the general picture in regards to applying ML techniques to calorimeter measured data will be transferrable.

4.2.1 Processes and Simulation

Simulated signal samples for isolated electrons are generated from the production and decay of a Z' boson in hadronic collisions through $pp \to Z' \to e^+e^-$ as an energy of $\sqrt{s} = 13$ TeV. The Z' mass is set to $m_{Z'} = 20$GeV in order to efficiently produce electrons within the desired range of $p_T = [10,\ 30]$ GeV, where hadronic backgrounds are most significant. Background samples are simulated via a dijet production process $pp \to jj$. Events were generated with MadGraph v2.6.5 [106], decayed and showered with pythia v8.235 [107], with detector response described by delphes v3.4.1 [108] using root version 6.0800 [22].

Delphes configuration was chosen to best approximate the ATLAS detector design [140]. The central region of the calorimeter is closely modeled as this region is the source of the majority of the energy deposits. Future work of interest may include the creation of a more realistic collection of edge-case data by more closely modeling the outer edges of the calorimeters.

However, the critical separation between the electromagnetic and hadronic calorimeters and their distinct segmentation is maintained. The simulated electromagnetic calorimeter (ECal) has segmention of $(\Delta\varphi, \Delta\eta) = \left(\frac{\pi}{126}, 0.025\right)$ while the simulated hadronic calorimeter (HCal) is coarser, $(\Delta\varphi, \Delta\eta) = \left(\frac{\pi}{31}, 0.1\right)$. This approach allows for the determination of whether information about the structure of the many-particle jet is useful for suppressing their contribution. See Ref [141] for an analysis of the information contained in the shape of shower for individual particles.

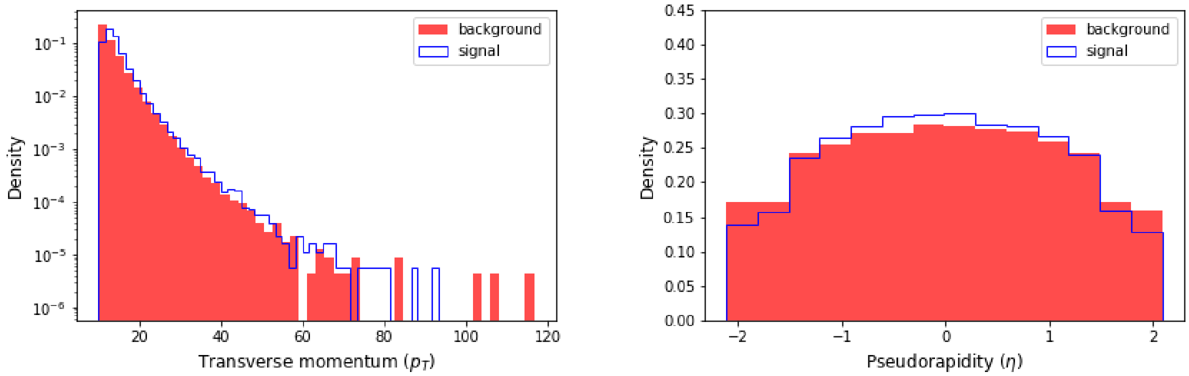


Figure 4.1: p_{T} (left) and η (right) distributions for electron candidate data for both signal and background simulated samples. Distributions are shown prior to re-weighting procedure.

No pile-up simulation was included in the generated data, as pileup subtraction techniques have been shown to be effective [142]. In total, 107k signal and 107k background objects were generated.

4.2.2 Electron Candidate Selection

Delphes standard electron identification procedures are used where loose electron candidates are selected from charge particle tracks which align with energy deposits in the ECal. An additional cut is placed on samples such that they have a minimum p_{T} of 10 GeV and fall within $|\eta| < 2.125$ to avoid edge effects when forming images (see Fig. 4.1). Finally, background objects undergo p_{T} re-weighting to guarantee p_{T} distributions between signal and background match. This is done to guarantee that ML classifiers can't make overly simple delineations between signal and background given a unique p_{T} characteristic signal.

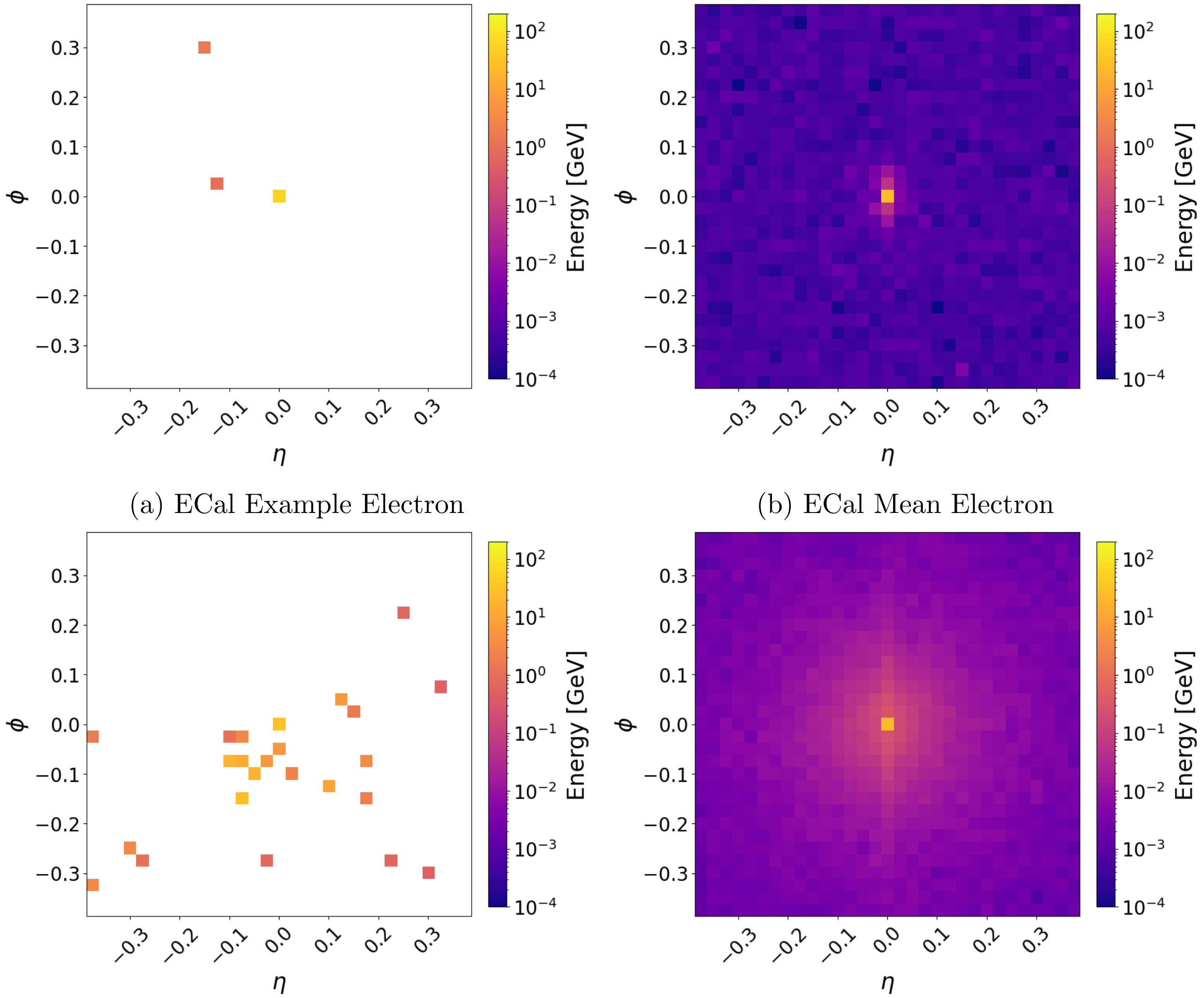
4.2.3 Image Formation

Much like the jet images produced in Ref. [28], the ECal and HCal measurements for electron identification naturally lend themselves to pixelation and computer vision analysis. A simple translation of the calorimeter to images would involve the assignment of a pixel to a cells energy, E. Alternatively, one could form images in which each pixel is represented by $E_{\mathrm{T}} = E/\cosh(\eta)$, which incorporates the object location into the pixel image relative to the collision point. For completeness, a version of both the ECal and HCal images are generated with both the simple energy-per-cell pixel representation and a version using E_{T}. This means that for each event, four LL inputs are available to train from: ECal E, ECal E_{T}, HCal E and HCal E_{T}. Each of the images are normalized to have a value range between $[0, 1]$, followed by a scaling by subtracting the mean image and dividing by its standard deviation.

The center of the calorimeter images is chosen such that the ECal cell with the largest E_{T} sits at the 9×9 cell region surrounding the track of the highest p_{T} electron in that event. This centering helps to account for the curvature in the path of the electron as it propagates between the tracker and calorimeter. All images form, in total, a 31×31 image. The HCal granularity is four times as course, and an 8×8 image covers the same physical region. Figs. 4.2 and 4.3 show example and mean images for the ECal and HCal, respectively.

4.3 Standard Classification Features

Data generation is performed to match the creation of a standard suite of electron identification features, as described in Ref. [136, 137], with some minor modifications. Since electron candidates are confined to the longitudinal range $|\eta| < 2.125$, only variables which are well defined in this range are used. Additionally, only variables which are based on information available in this simulation are selected so as to guarantee that comparisons are done so on an equal footing. Clustering is not used as it is unnecessary given the simplified nature of the Delphes simulation. In the instance that a HL feature requires the clustered energy, this is replaced with the total energy of the image. This is a reasonable proxy as the simplified



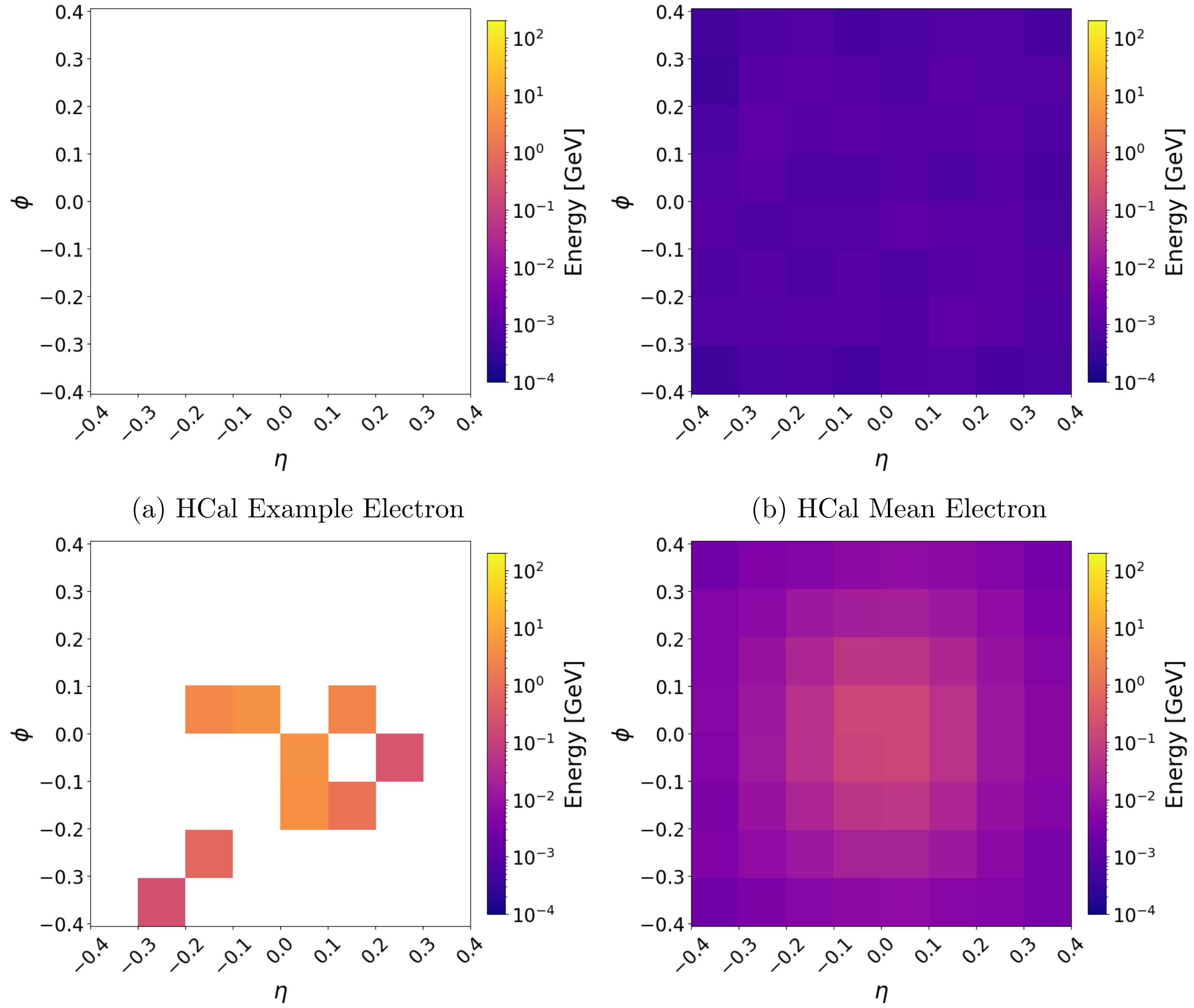
(a) ECal Example Electron

(b) ECal Mean Electron

(c) ECal Example Jet

(d) ECal Mean Jet

Figure 4.2: Individual (left) and average (right) ECal images for signal electrons (top) and corresponding ECal images for hadronic jet background samples (bottom).



(a) HCal Example Electron

(b) HCal Mean Electron

(c) HCal Example Jet

(d) HCal Mean Jet

Figure 4.3: Individual (left) and average (right) HCal images for signal electrons (top) and corresponding HCal images for hadronic jet background samples (bottom).

simulation of the calorimeter response in Delphes is not likely to deposit electron energy in multiple disconnected clusters.

All HL features are generated directly from the ECal and HCal image data, using E or E_{T} where necessary. Ultimately, seven HL features are produced, including: R_{had}, $\omega_{\eta 2}$, R_{φ}, R_{η}, $\sigma_{\eta\eta}$ and two isolation quantities. These observables represent a common strategy for suppressing objects with significant hadronic energy or extended energy deposits. Definitions of each feature are below, and distributions for signal and background samples are shown in Fig. 4.4.

Ratio of HCal and ECal Energy: $R_{\mathbf{had}}$

The feature R_{had} relates the transverse energy (E_{T}) in the electromagnetic calorimeter to that in the hadronic calorimeter. Specifically,

$$R_{\mathrm{had}} = \frac{\sum\limits_{i} E_{\mathrm{T},i}^{\mathrm{HCal}}}{\sum\limits_{j} E_{\mathrm{T},j}^{\mathrm{ECal}}} \tag{4.1}$$

where i and j run over the pixels in the HCal and ECal images, respectively.

Lateral Width of the ECal Energy Shower: $w_{\eta 2}$

The lateral width of the shower in the ECal, $w_{\eta 2}$, is calculated as

$$w_{\eta 2} = \sqrt{\frac{\sum\limits_{i} E_i (\Delta\eta_i)^2}{\sum\limits_{i} E_i} - \left(\frac{\sum\limits_{i} E_i \Delta\eta_i}{\sum\limits_{i} E_i}\right)^2} \tag{4.2}$$

where E_i is the energy of the i^{th} pixel in the ECal image and $\Delta\eta_i$ is the pseudorapidity of the i^{th} pixel in the ECal image measured relative to the image's center. The sum is calculated within an $(\Delta\eta \times \Delta\varphi) = (3 \times 5)$ cell window centered on the image's center.

Azimuthal and Longitudinal Energy Distributions: R_φ and R_η

To probe the distribution of energy in azimuthal (φ) and longitudinal (η) directions, R_φ and R_η are used. Qualitatively, these relate the total ECal energy in a subset of cells to the energy in a larger subset of cells extended in either φ or η, respectively. Specifically,

$$R_\varphi = \frac{E_{3\times3}}{E_{3\times7}}, \tag{4.3}$$

$$R_\eta = \frac{E_{3\times7}}{E_{7\times7}}, \tag{4.4}$$

where the subscript indicates the number of cells included in the sum in η and φ respectively. For example, $(\eta \times \varphi) = (3 \times 7)$ is a subset of cells which extends 3 cells in η and 7 in φ relative to the center of the image.

4.3.1 Lateral Shower Extension

An alternative probe of the distribution of energy in η is $\sigma_{\eta\eta}$ [136]

$$\sigma_{\eta\eta} = \sqrt{\frac{\sum\limits_i w_i (i_\eta - \bar{i_\eta})^2}{\sum\limits_i w_i}} \tag{4.5}$$

Where w_i is the weighting factor $|\ln(E_i)|$ with E_i being the ECal energy of the i^{th} pixel. The sum runs over the non-zero cells in the $(\eta \times \varphi) = (5 \times 5)$ subset of cells centered on the highest energy cell in the ECal. Here, i_η is measured in units of cells away from center, $\bar{i_\eta}$, as $i_\eta \in 0$, ± 1, or ± 2 when choosing $\bar{i_\eta} = 0$.

4.3.2 Isolation

Jets typically deposit significant energy surrounding the energetic core, while electrons from heavy boson decays are typically isolated in the calorimeter. Electrons may also appear inside jets in decays of B-mesons for example, but here the focus is on decays from real W and Z bosons. To assess the degree of isolation, the sum the ECal energy in cells within the angular range $\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < 0.3$ or 0.4, where $\Delta\eta$ and $\Delta\varphi$ are measured from a given cell's center and the center of the image.

4.4 Neural Network Architectures and Training

For each trained network, including those trained on LL or HL features, a sigmoid output layer is used to make binary classifications between an electron signal and jet background.

LL images are passed through a series of convolutional blocks with each block consisting of two convolutional layers with a 3×3 kernel, rectified linear units [143] as the activation function and a final 2×2 maxpooling layer. Outputs from a maxpooling layer are flattened and concatenated with the high-level inputs to form a high-dimensional vector. The high-dimensional vector is processed by a sequence of fully connected layers with rectified linear units using dropout [144, 145].

The final output is produced by a single logistic unit and interpreted as the probability of a signal classification relative to background. All architectures are trained with stochastic gradient descent to minimize the relative entropy between targets and outputs across

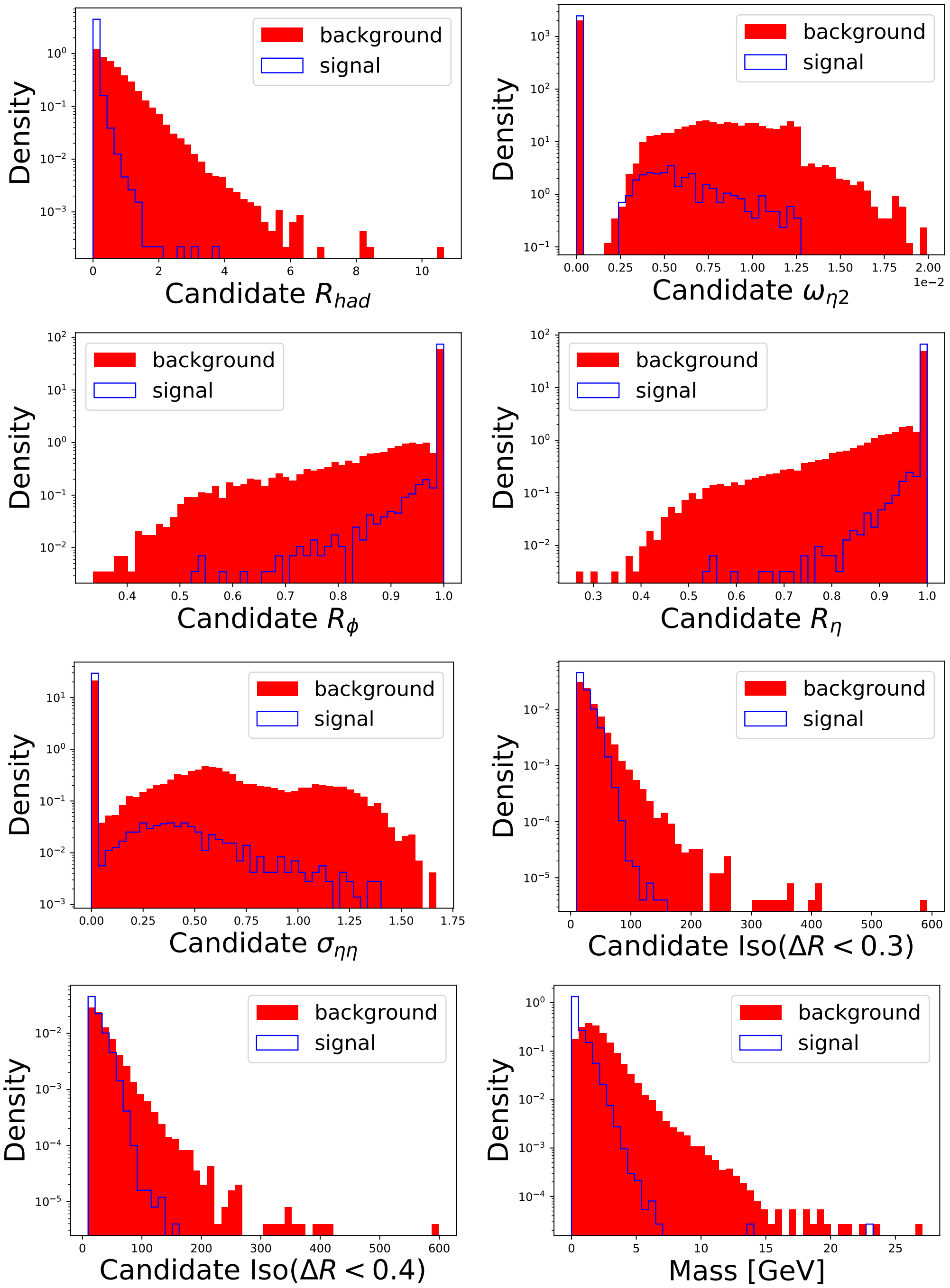


Figure 4.4: Electron signal (red) and jet background (blue) distributions for the seven generated HL features and mass.

all training examples. For HL networks, all combinations are trained and tuned as fully connected neural networks with a similar sigmoid unit at the top.

All trained models are implemented using KERAS [146] with TENSORFLOW [147] as the backend and trained with a batch size of 128 with the ADAM optimizer [148]. The weights for all the models were initialized using Glorot [149] uniform weights and each network was tuned using 150 iterations of bayesian optimizaton with the SHERPA hyperparameter optimization library [150]. Additional details about the hyperparameters and their optimization are given in Tables B.1, B.2 and B.3.

4.5 Performance

Preliminary studies showed that the use of images with both E and E_{T} information performed no better than networks with purely E_{T} information. As such, the LL inputs were simplified to only use E_{T} based images and the “image” results given in the following sections use this image subset. A full comparison of performance, as measured by the AUC for networks trained on both the E_{T} images and 7 HL features is given in Table 4.1 and the ROC curves for all networks given in Fig. 4.5.

Similar to the jet substructure example given in 3.3, the HL features (AUC = 0.945) fail to achieve the same performance as the LL image networks (AUC = 0.972). This suggests that there is information contained within the image representation of the data that does not translate into the standard HL features generated from the literature. Per the previous chapter, this type of performance gap is not an unexpected result. Futhermore, networks which train strictly on the ECal or Hcal data but not both perform worse than those which view the combination of images. This confirms suspicions that useful and unique information for electron classification appears in both calorimeters. Including the HL features in a

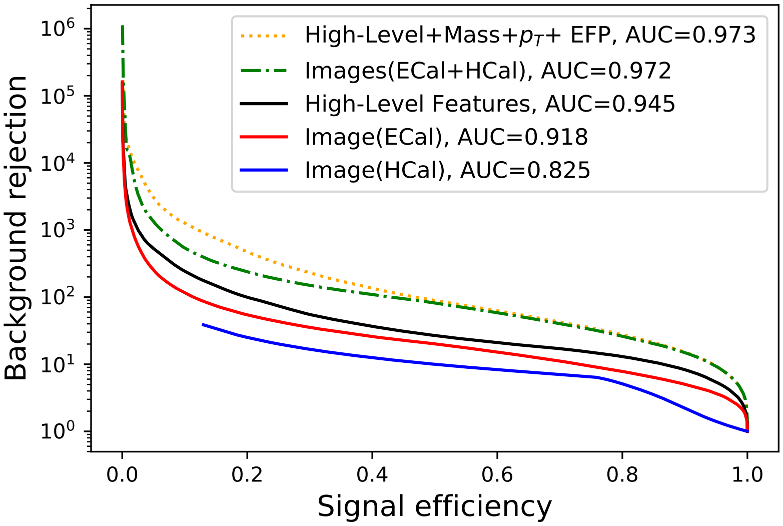


Figure 4.5: ROC curves for various networks trained on the electron identification task. Results include a CNN on HCal and Ecal images (solid blue and solid red, respectively), a DNN trained on 7 HL features (solid black), a CNN trained on the ECal and HCal simultaneously (dashed-dotted green) and a DNN trained with 7 HL features, mass and a black-box EFP (dotted orange).

network with the LL images, however, shows no significant improvement. This suggests that the information contained within the ECal and HCal images is sufficient to capture all of the information highlighted by the HL features.

4.6 Bridging the gap

The performance gap identified in Table 4.1 indicates the presence of information in the LL images not captured by the suite of existing high-level features. This situation naturally lends itself to the black-box guided method introduced in Sec. 3.5.1. The goal, as it was for jet substructure, is to find a set of interpretable and physically meaningful features with the same classification performance as a much more complex LL network. Crucially, this can be accomplished by using the demonstrated CNN on low-level information as a guide to isolate any new HL features.

Network Features				AUC
Images		7 Standard		
ECal	HCal	HL Features	Mjet	
	✓			0.82 ± 0.02
✓				0.918
✓	✓			0.972
✓	✓	✓		0.973
✓	✓	✓	✓	0.973
		✓		0.945
		✓	✓	0.956

Table 4.1: Training performance (defined by AUC) for the electron classification task using various combinations of HL and LL features. ECal and HCal images use strictly E_T information in their pixels. The seven HL observables consist of: R_had, $\omega_{\eta 2}$, R_φ, R_η, $\sigma_{\eta\eta}$, Iso($\Delta R < 0.3$), Iso($\Delta R < 0.4$). Uncertainties are calculated by a 95% confidence interval on 200 bootstrapped training examples and do not exceed ±0.001 unless otherwise specified.

Unlike the jet substructure problem used in Chp. 3, the HL features used here must be sensitive to the specific task they are meant to address. In contrast to a purely jet substructure analysis, the energy depositions produced in the calorimeters for produced electrons will be unique in comparison to jets, which can potentially exhibit a rich structure and comprise a mixture of jets from gluons, light quarks and heavy quarks. It is expected, for example, that features sensitive to jet substructure or quantity may provide strong discrimination power.

Considering the possible benefits of jet substructure information, the jet mass (M_jet) is included, despite not being a common electron identification feature. This added feature can be seen in Fig. 4.4 as showing strong signal/background separation. Furthermore, the inclusion of M_jet into a HL network provides an improvement in performance as seen in Table 4.1. This inclusion acts as a strong motivation to further explore lesser used features or observables common from the jet substructure literature.

4.6.1 Set of Observables

One could in principle consider an infinite number of jet observables. To organize the search, the Energy Flow Polynomials (EFPs) [81] are used again, with details and design outlined in Sec. 3.3.2. Recall that, in principle, the space is complete such that any jet observable can be described by one or more EFPs of some degree. In practice, only a finite subset can be explored. In this search, the space under consideration includes all observables with up to seven edges and with $\beta \in \left[\frac{1}{2}, 1, 2\right]$ and $\kappa \in [-1, 0, 1, 2]$. Each graph is computed and applied separately to the ECal or the HCal, effectively doubling the number of graphs used. In total, 15,090 graphs are available to the search. Note that one might suppose that a version of the EFPs which attempts to use both simultaneously during its generation might benefit from the encoding of pairwise information into the feature. A separate study was done with this approach with no measured performance improvement. As such, the final space of EFPs used were kept distinct so that individual benefits found in the electromagnetic or hadronic calorimeter could be observed and analyzed without the complication of untangling a “mixture” feature of calorimeter information.

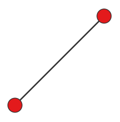
4.6.2 Searching for Observables

Rather than conduct a brute-force search (as studied and suggested against in Sec. 3.5.2) of this large space, the black-box guided method is applied to attempt to isolate an EFP based on an optimized CNN. The method described in Sec. 3.4.1 is followed identically with the one modification that the pool of EFPs undergoing comparison via ADO can include either HCal or ECal variants.

For all HLN$_n$ used in this search, models were trained with Keras [146] using Tensorflow [147] as the backend. Each model was built as a fully connected neural network of simple one dimensional input features and a single logistic unit output. The guided search requires training a new HLN$_n$ after each new EFP selection. Performing a full Bayesian optimization with Sherpa and bootstrapping each network becomes computationally expensive. Instead, a simpler architecture was found to be provide consistent, stable, performance. These networks consisted of 3 hidden layers, each with 50 rectified linear units, separated by 2 dropout layers using a dropout value of 0.25 and trained with a batch size of 128. The Adam optimizer [148] was used with learning rate of 0.001 and initialized with Glorot [149] normal weights.

4.6.3 IRC safe observables

In the initial search, a subset of the initial set of EFPs is made to isolate the impact of IRC safety to this specific classification task. This is done by restricting the guided search to only graphs with $\kappa = 1$, which amounts to 3,018 graphs. Starting first with the seven HL features (but excluding mass), the first graph selected in the black-box guided process is



$$\left(\qquad \right)^{\left(\kappa=1,\beta=\frac{1}{2}\right)} = \sum_{a,b=1}^{N} z_a z_b \sqrt{\theta_{ab}} \tag{4.6}$$

This graph has an ADO with the CNN of 0.802 when evaluated over the differently ordered subspace between CNN and HL, suggesting it is well aligned with the CNN strategy. Including this single feature with the original seven HL inputs yields a performance of AUC $= 0.970 \pm 0.001$, very nearly closing the gap with the CNN. This graph has a very similar structure to that of jet mass (Eq. (3.26)), a pairwise sum over cells which folds in angular separation. However, it bears an even closer resemblance to the Les Houches Angularity (LHA) [85], which similarly is sensitive to the distribution of energy away from the center, though with a smaller power of the angularity than jet mass. This suggests that the network

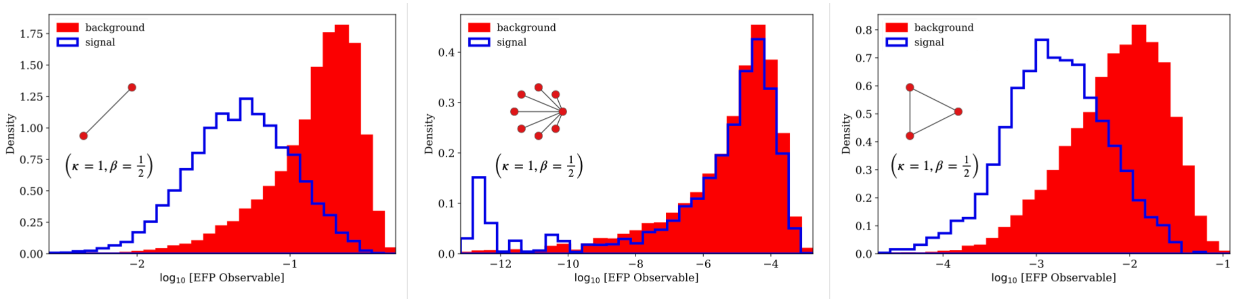
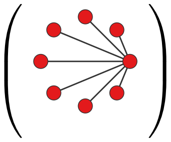


Figure 4.6: $\log_{10}$ distributions of the selected IRC-safe EFPs as chosen by the black-box guided strategy, for signal electrons and background jets.

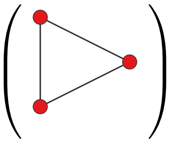
benefits from the addition of extra small angle information in the LHA in comparison to the jet mass.

If instead, one begins with the seven HL features with the inclusion of jet mass, the black-box method selects two graphs:



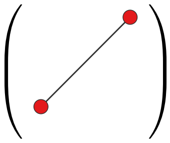
$$^{(\kappa=1,\beta=1)} = \sum_{a\cdots h=1}^{N} z_a \ldots z_h \theta_{ab}\theta_{ac}\theta_{ad}\theta_{ae}\theta_{af}\theta_{ag}\theta_{ah} \tag{4.7}$$

and



$$^{\left(\kappa=1,\beta=\frac{1}{2}\right)} = \sum_{a,b,c=1}^{N} z_a z_b z_c \sqrt{\theta_{ab}\theta_{bc}\theta_{ac}} \tag{4.8}$$

Which, when combined with the seven HL features and M_{jet}, gives ten observables that achieve an AUC of 0.971 ± 0.001 and nearly matching the performance of the CNN. Distributions of these observables for signal and background samples are shown in Fig. 4.6. As the EFPs are normalized, they are sensitive to relative distributions of energy rather than the overall scale. Per comments made in Sec. 3.5.1, the inclusion of the jet p_{T} sum as an observable should help give a relative scale for the dimensionless EFPs to be measured against. The jet p_{T}, when combined with the seven HL features and M_{jet} gives an AUC of 0.965. Performing the IRC safe guided search a second time with the inclusion of jet p_{T} now identifies the familiar graph,

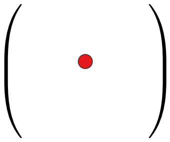


$$^{\left(\kappa=1,\beta=\frac{1}{2}\right)} = \sum_{a,b=1}^{N} z_a z_b \sqrt{\theta_{ab}} \tag{4.9}$$

The addition of this graph with HL features, M_{jet} and jet p_{T} reaches an AUC of 0.973±0.001, completely closing the gap.

4.6.4 Broader Scan

Remove the previous restriction, a broader search can be performed with the inclusion of EFPs which don't conform to IRC safety. Returning to the full set of generated EFPs, the parameter kappa can now include $\kappa \in [-1, 0, 1, 2]$. Beginning, once again, from the seven standard HL features and running a single iteration of the black-box guiding strategy, the selected EFP is:



$$^{(\kappa=2)} = \sum_{a=1}^{N} z_a^2 \tag{4.10}$$

This choice of EFP features no angular term (i.e. $\beta = 0$) and a sensitivity to wide angle information ($\kappa = 2$). In fact, this observable is an existing jet substructure and detailed in the literature as p_{T}^{D} [82, 119]. The original development for p_{T}^{D} was in the separation of quark and gluon jets. In the electron classification, when p_{T}^{D} is included with the original seven HL features, the performance reaches AUC $= 0.970 \pm 0.001$. Additional scans do not lead to any statistically significant improvements in performance.

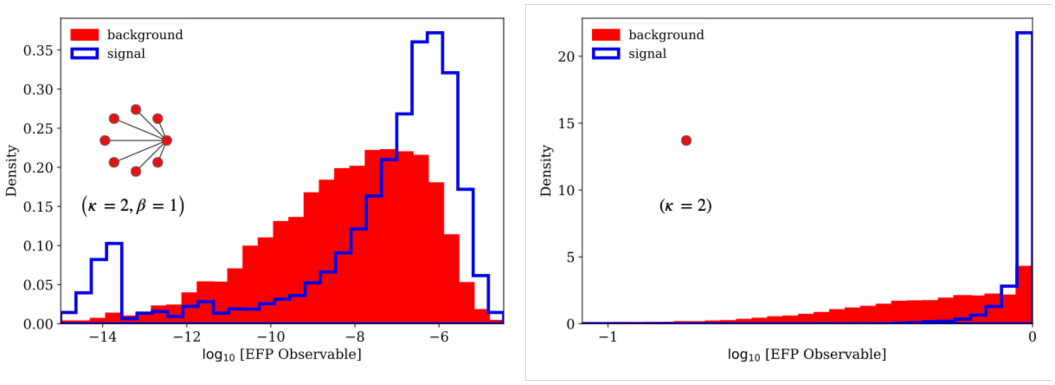
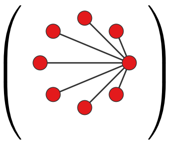


Figure 4.7: $\log_{10}$ distributions of the selected EFPs as chosen by the black-box guided strategy, regardless of IRC safety, for signal electrons and background jets.

With the inclusion of M_{jet} into the broader scan, the EFP chosen is



$$^{(\kappa=2,\beta=1)} = \sum_{a\cdots h=1}^{N} (z_a \ldots z_h)^2 \, \theta_{ab}\theta_{ac}\theta_{ad}\theta_{ae}\theta_{af}\theta_{ag}\theta_{ah} \tag{4.11}$$

Note that this selection is identical to the IRC Safe search result found in Eq. (4.7) but with a change in the momentum-fraction parameter to $\kappa = 2$. Distributions for these two IRC unsafe EFP observables are given in Fig. 4.7. Training a network with the seven HL features, M_{jet} and these two IRC unsafe selections results in an AUC of 0.971 ± 0.001. Additional iterations of the guided search don't provide any extra performance benefits. However, when beginning from the seven HL observbles, M_{jet}, jet p_{T} and the two IRC unsafe EFPs, the performance can be closed with an AUC of 0.973 ± 0.001.

See Table 4.2 for a summary of the additional observables needed to reach the performance of ≈ 0.97 in each case, and Table 4.3 for background rejection factors for several choices of signal efficiency.

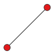
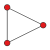
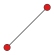
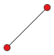
Base	Additions (κ, β)	AUC
7HL		0.945
7HL	 $(1, \frac{1}{2})$	0.970
7HL	(2,-)	0.970
7HL + M_{jet}		0.956
7HL + M_{jet}	$(1, 1)$  $(1, \frac{1}{2})$	0.971
7HL + M_{jet}	$(2, 1)$ (2,-)	0.971
7HL + M_{jet} + p_T		0.965
7HL + M_{jet} + p_T	 $(1, \frac{1}{2})$	0.973
7HL + M_{jet} + p_T	$(2, 1)$ (2,-)	0.973
CNN		0.972
CNN + 7HL	 $(1, \frac{1}{2})$	0.972
CNN + 7HL	(2,-)	0.973

Table 4.2: Summary of the performance of various networks considered. Uncertainty in the AUC value is ±0.001, estimated using bootstrapping.

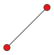
Features	AUC	$R_{\epsilon=0.5}$	$R_{\epsilon=0.75}$	$R_{\epsilon=0.9}$
7HL	0.945	32.98	15.78	8.80
7HL +  $(1, \frac{1}{2})$	0.970	88.63	34.73	15.07
CNN	0.973	94.07	36.89	15.93

Table 4.3: Performance of selected networks, in terms of the AUC value as well as background rejection (R) at several choices of signal efficiency (ϵ).

4.7 Discussion

Upon analyzing the results of a deep neural network on LL calorimeter data, once again it is observed that the physics-motivated features generated directly from the raw calorimeter measurements has lost important detail relevant for the classification of signal particles from background. Following the black-box guided procedure of Sec. 3.2.3, this information was recaptured as a simple and understandable collection of physical observables.

One of the first and most noticeable conclusions of the selected EFPs from the guided search is that they only ever preferred EFPs using the ECal information for their creation. This, alone, is an interesting result as the EFPs were strongly motivated for the information they provide relative to jet classification but the dataset most useful for constructing those jet substructure observables seems to come primarily from the electromagnetic calorimeter.

The first EFP chosen, in Eq. (4.9), is closely related to the Les Houches Angularity [85], and confirms suspicions that the non-trivial structure of the background object provides a useful handle for classification. The second observable, given by Eq. (4.10), is in fact a the jet substructure observable p_{T}^{D} [82, 119], and is not currently used in electron identification networks. This is an example of an IRC unsafe feature and was originally developed to help distinguish between quark and gluon jets. It effectively counts the number of hard particles, which is sensitive to the amount of color charge, where electrons and jets are clearly distinct. Both Les Houches Angularity and p_{T}^{D} display power to separate electrons from the jet backgrounds, by exploiting the structure and nature of the jet energy deposits.

The studies performed here use a simplified simulation of the detector, and notably lack an accurate description of the radiation of photons from electrons, which may result in an unrealistic pattern of energy deposition and secondary clusters. While the precise performance obtained here may depend at some level on the fidelity of the simulation used and the resulting limitations on the implementation of state-of-the-art high-level features, these results strongly suggest that these observables be directly studied in experimental contexts where more realistic simulation tools are available, or directly in data samples, using weakly supervised learning [151].

More broadly, the existence of a gap between the performance of state-of-the-art high-level features and CNN represents an opportunity to gather additional power in the effort to suppress lepton backgrounds. Rather than employing black-box CNNs directly, relevant observables from a large list of physically interpretable options can be isolated given the context of that CNN. This allows the physicist to understand the nature of the information being used and to assess its systematic uncertainty.

Chapter 5

Prompt Muon Isolation with Interpretable Learning

5.1 Introduction

Searches for new physics at the LHC frequently rely on the investigation of leptonic decays for heavy bosons. This is an advantageous search path due to the relatively low background rates and excellent momentum resolution in compared to hadronic final states. One such search example is that of *prompt muons*, which are produced from the decay of W, Z or other bosons. the primary background to a prompt muon process occurs within heavy-flavor jets and occurs most significantly at lower values of muon transverse momentum. Such searches have become critical in various searches of supersymmetry supersymmetry [152–154] and low-mass resonances [155].

Standard techniques for isolating prompt muons from non-prompt muon backgrounds typically involves the combination of data from multiple detector components [156, 157]. Critical to these strategies is the concept of isolation, which is sensitive to the presence of an associated jet that produces many tracks and calorimeter deposits. While the entire detector is worth studying [156], the focus here will be on the nature of the information available in the calorimeters. With access to calorimeter measurements, the standard method for deriving high-level features involves the measurement of:

$$I_\mu(R_0) = \sum_{i, R<R_0} \frac{p_{\mathrm{T}}^{\text{cell } i}}{p_{\mathrm{T}}^{\text{muon}}} \tag{5.1}$$

$$R = \sqrt{\Delta\varphi^2 + \Delta\eta^2} < R_0 \tag{5.2}$$

Here, I_μ is described as an *isolation cone* which is defined for some radial size R surrounding the muon [158]. Generally a single cone is used with some value of R_0 between 0.1 and 0.45 selected. This approach relies on identifying a typical characteristic of the signal, low calorimeter activity in the vicinity of the muon.

In an analogous situation to the electron classification scheme detailed in Chp. 4, the strong focus on features sensitive to the signal features may sacrifice classification performance missed out on the complex characteristics of the background sample. In addition to the electron example in the previous chapter, related work has demonstrated advantages for object classification tasks when focused as a background jet rejection problem when applied to photons [159, 160] and pions [161]. Furthermore, studies have shown that muons which fail the traditional isolation requirement can contain power to reveal new physics [162].

5.2 Approach and Dataset

In line with demonstrations from Chp. 3 and Chp. 4, the application of an advanced machine learning approach to low-level data can be expected to show benefits over the comparatively simplistic use of high-level isolation cones. In the presence of another performance gap between high-level and low-level information, useful features gleaned from the nuances of the background samples can be used to inform and improve prompt muon isolation.

For the task of muon isolation of prompt muons from background, isolation cones ($I_\mu\left(R_0\right)$) is a powerful discriminating feature and a conveniently simple high-level feature. However, it is likely that such simplicity comes at the cost of a not fully capturing the total complexity of prompt muon classification from calorimeter information. The presence of missing information in this high-level data can be established through the training of a benchmark low-level deep neural network. If and where there is a performance gap, a black-box guided search can be used to attempt to isolate new and efficient training features

5.2.1 Data generation

Samples of simulated prompt muons were generated via the process $pp \rightarrow Z' \rightarrow \mu^+\mu^-$ with a Z' mass of 20 GeV. Non-prompt muons were generated via the process $pp \rightarrow b\bar{b}$. Both samples are generated at a center of mass energy $\sqrt{s} = 13$ TeV. Collisions and heavy boson decays are simulated with MADGRAPH5 v2.6.5 [106], showered and hadronized with PYTHIA v8.235 [107], and the detector response simulated with DELPHES v3.4.1 [108] using the standard ATLAS card and ROOT version 6.0800 [22]. The classification of these objects is sensitive to the presence of additional proton interactions, referred to as pile-up events. The interactions are overlayed within the simulation with an average number of interactions per event of $\mu = 50$, as an estimate of LHC Run 2 experimental data.

Muons in the range $p_\mathrm{T} \in [10, 15]$ GeV with $|\eta| < 2.53$ were considered; see Fig. 5.1. To avoid inducing biases from artifacts of the generation process, signal and background events are weighted such that the distributions in p_T and η are uniform, using 32 bins in each dimension. Only events where a muon is identified as a track in the muon spectrometer are used. In total, 499,970 events were used, where 249,991 were signal and 249,979 were

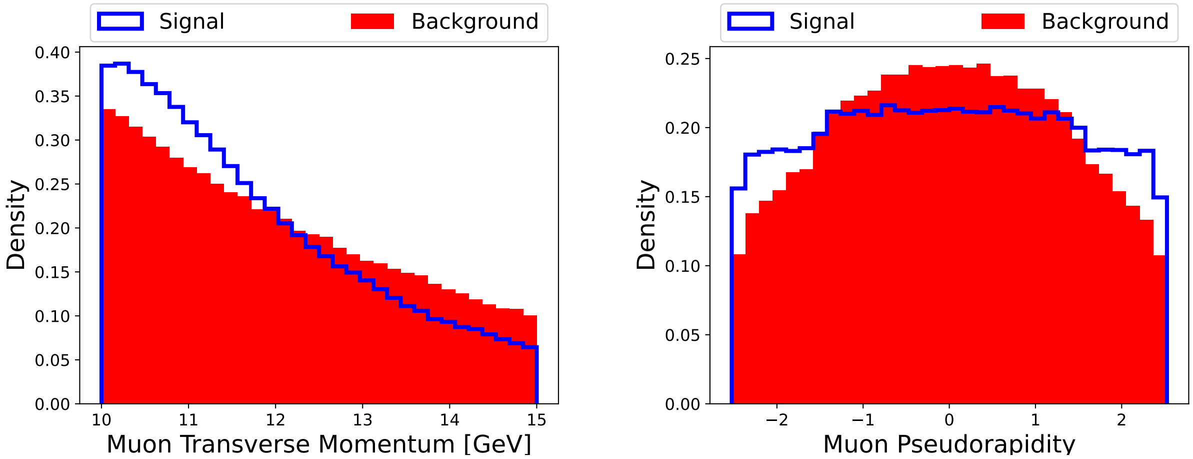
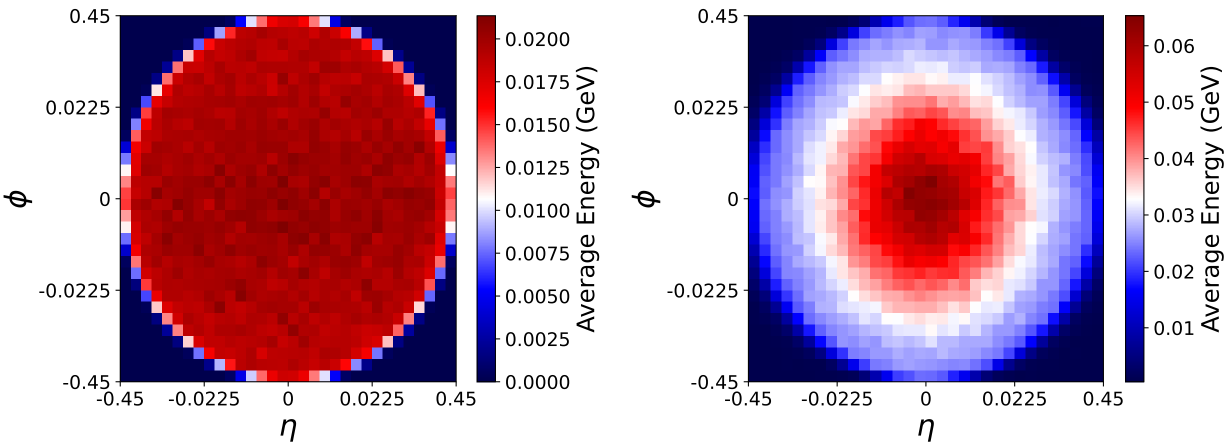


Figure 5.1: Transverse momentum (left) and pseudorapidity (right) distributions for prompt muon signal (blue) and non-prompt muon backgrounds (filled, red). Distributions are shown prior to any signal/background re-weighting.

background. Both the signal and background datasets are randomly split as: 83% training, 8.5% validation, and 8.5% testing sets. Following the same procedure described in Sec. 4.2.3, transverse energy E_{T} is taken from calorimeter measurements and assigned as pixels to produce images. For muonic calorimeter images, pixels are restricted to calorimeter cells up to a radius of $\Delta R = 0.45$ surrounding the muon location after propagating to the radius of the calorimeter. The images are divided into a 32×32 grid which roughly corresponds to calorimeter granularity used in the ATLAS and CMS detectors. Heat maps of the calorimeter energy deposits in $\eta - \varphi$ space for both signal prompt muons and background non-prompt muons are shown in Fig. 5.2. The signal calorimeter deposits are uniform and can be attributed to pileup whereas the background deposits appear largely radially symmetric with a dense core from the jet. Taking Eq. (5.2), isolation cones are calculated for 18 radii equally spaced in the range of 0.025 - 0.45. This calculation is performed using the pixel information taken from the low-level images to guarantee that the final comparison of high-level and low-level performance is on an equal footing. While it is true that some minimal information while be lost during the pixelation process for generating images, this study is primarily focused on making an equivalent comparison between calorimeter images and isolation cone features such that information can be mapped between them.



(a) Mean Prompt Muon

(b) Mean Non-Prompt Muon

Figure 5.2: Average calorimeter images for prompt muons (left) and muons produced through heavy-flavor jets (right) within $\Delta R = 0.45$ of the reconstructed muons. Cells consist of transverse energy E_{T} and are normalized to sum to unity.

5.3 Networks and Performance

In the same fashion as before, the high-level and low-level performance by training an optimal classifier using various deep learning classifiers. Starting with the high-level features, a network is initially trained using just a single isolation cone (set to $R_0 = 0.425$). This value was chosen as the best single performing isolation cone when compared, by brute force, to networks trained on every other choice of radius. This singular isolation cone trained via a deep neural network achieved an AUC of 0.787. This test was then followed by the training of 17 more networks, each time including an additional isolation cone via a greedy search (using the network architecture given in App. C.1). One might expect that with the inclusion of every new feature, performance would similarly continue to improve as different sensitivities to the radial information of the calorimeter is probed. On the second addition of an isolation cone, the performance only slightly increases relative to the previous single isolation cone network (AUC = 0.793). Results for this greedy search are given in Fig. 5.3 where an obvious plateau in performance is observed after the initial contribution from a few included isolation cones.

Various low-level networks were then trained using the calorimeter image data. This included a convolutional neural network (CNN) and, in contrast to the tests in Chps. 3 and 4, it is joined by an Energy Flow Network (EFN) and Particle Flow Network (PFN) [65]. Both the EFN and PFN are explicitly jet oriented deep learning architectures which aim to train a machine learned function solver inspired by the same momentum-fraction and angular separation equation composition as Energy Flow Polynomials (EFP). Specifically, for a learnable "per-particle" function Φ and latent space function F, an EFN trains a model to solve the function

$$\mathrm{EFN} = F\left(\sum_{i}^{M} z_i \Phi\left(\hat{p}_i\right)\right) \tag{5.3}$$

where z_i is the particle or constituent energy ($z_i = p_{\mathrm{T},i}$) and $\hat{p}_i$ is the angular information such that $\hat{p}_i = (\eta_i, \varphi_i)$. Alternatively, the PFN solves a function of the form

$$\mathrm{PFN} = F\left(\sum_{i}^{M} \Phi\left(p_i\right)\right) \tag{5.4}$$

where p_i is the particle information (i.e. four-momentum, charge, flavor). The power of Particle-Flow Networks (PFNs) relies on their ability to learn virtually any symmetric function of the towers. Their mathematical structure is naturally invariant under permutation of the input ordering, as it is built on a summation over the constituents

Using the same set of input for each, a CNN, EFN and PFN are all trained for prompt muon classification. The worst of these three is the CNN, which yields an AUC of 0.841. Despite it's poor performance relative to the other low-level networks, this result is still significantly improved over the isolation cone networks for any number of inputs. The PFN has the best performance at AUC 0.857 with the EFN in the middle (see Fig. 5.4 and Table 5.1 for complete results). This immediately suggests that there is significant additional information available for this classification task which is not well translated into isolation cones. Note that an important difference between the two Flow Networks involves a lack of IRC Safety

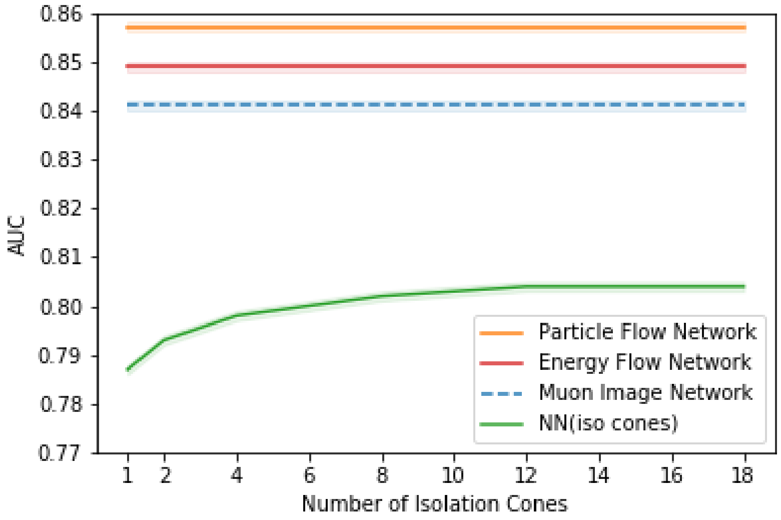


Figure 5.3: Classification performance for the greedy search with a deep neural network on high-level isolation cones (green) and low-level networks trained with a CNN (blue, dashed), Energy Flow Network (red) and Particle Flow Network (orange).

in a PFN but which is present in the EFN. Given the significant performance gap between the isolation cone and EFN and a modest improvement between the EFN and PFN, this strongly suggests that most of the information missing from the isolation cone network is in fact IRC Safe content.

These results support the conventional wisdom that a significant fraction of the information relevant for classification is captured by a single cone with an appropriately chosen characteristic radius. However, this also indicate that there is additional information in the radial distribution of energy, which can be captured by using multiple cones. Most importantly, it's clear, that even the inclusion of many isolation cones falls considerably short of the performance for networks with access to direct calorimeter cell information. The stark contrast in performance between these two approaches is likely due to discrepancies between the muon axis, isolation cone center and jet axis which is not translated to the isolation cones in their current form.

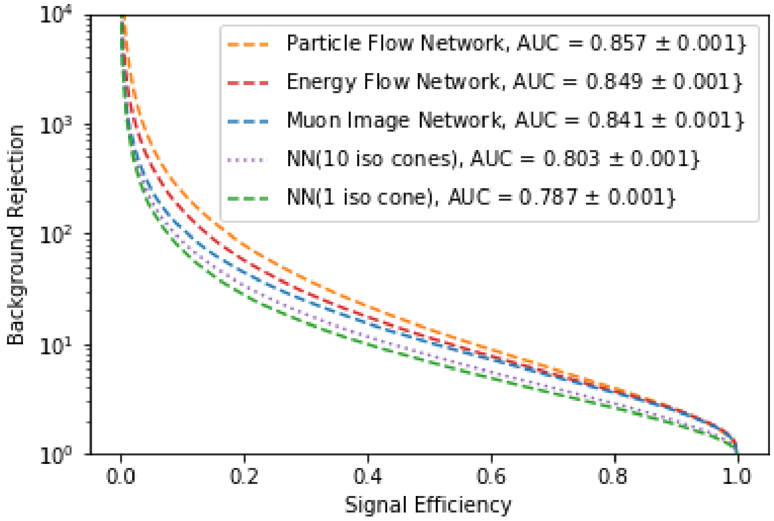


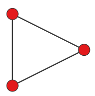
Figure 5.4: ROC curves for two examples of the greedy search with a deep neural network on high-level isolation cones (1 iso cone [green, dashed] and 10 iso cones [purple, dotted]) and low-level networks trained with a CNN (blue, dashed), Energy Flow Network (red, dashed) and Particle Flow Network (orange, dashed).

5.4 Analysis & Search Strategy

Given the success of the black-box guided strategy in both the jet substructure problem from Chp. 3 and electron classification task from Chp. 4, it is now applied to the discrimination problem for prompt muons. The discrepancy between EFN and PFN performance seen in the previous section suggests that the exploration of both IRC Safe and IRC Unsafe information will be of interest to this search. Therefore, distinct pools of candidate EFPs are generated: one with exclusively IRC safe ($\kappa = 1$) EFPs and another and with access to IRC Unsafe examples ($\kappa \neq 1$). Also, recall that $\kappa > 0$ generically corresponds to IR-safe but C-unsafe observables which are included in the IRC Unsafe search. For $\kappa < 0$, empty cells are omitted from the sum. In all examples of the guided search, 10 isolation cones are used corresponding to the initial 10 features found through the greedy search prior to a performance plateau. Additionally, image p_{T} (i.e. the summation over all image p_{T} values) is included as an input feature to add a necessary scaling factor for networks to understand the relative size of the otherwise dimensionless EFPs

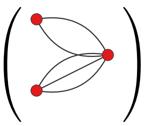
5.4.1 IRC Safe Observables

Starting with the IRC Safe example (a subset of the complete pool of candidate EFPs), the feature selection is also initially reduced to a smaller set of simpler EFPs. This is done to give initial preferential attention to simpler EFPs which, irrespective of performance, are generally easier to interpret and allow for much more direct comparisons to existing high-level discrimination features. From the starting group of IRC Safe EFPs, the subset with no more than 3 nodes, no more than 3 edges connected between nodes and parameters of $\kappa = 1$ and $\beta \in [1, 2]$ are allowed. Starting with ten isolation cones and p_{T}, a black-box guided search is performed with the PFN model as the low-level guide. The first EFP selected is a simple three-point correlator:

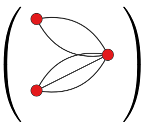


$$\left(\quad \right)^{(\kappa=1,\beta=1)} = \sum_{a,b,c=1}^{N} z_a z_b z_c \theta_{ab}\theta_{bc}\theta_{ca} \qquad (5.5)$$

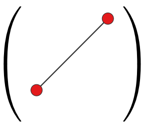
which, when combined with the ten isolation cones and p_{T}, yields an AUC of 0.838 and an ADO with the PFN of 0.891. The inclusion of just this single EFP marks a significant improvement relative to just using the radial information of the isolation cones. The subsequent scans produce two variants of one graph and a familiar looking two-point correlator:



$$^{(\kappa=1,\beta=2)} = \sum_{a,b,c=1}^{N} z_a z_b z_c \theta_{ab}^4 \theta_{bc}^6 \tag{5.6}$$



$$^{(\kappa=1,\beta=1)} = \sum_{a,b,c=1}^{N} z_a z_b z_c \theta_{ab}^2 \theta_{bc}^3 \tag{5.7}$$



$$^{(\kappa=1,\beta=2)} = \sum_{a,b=1}^{N} z_a z_b \theta_{ab} \tag{5.8}$$

For the graphs identified in iteration two and three (Eqs. (5.6) and (5.7), the improvement in performance might be due to additional edges corresponding to higher powers of the angular information. Their power may come from their sensitivity to the collimated radiation pattern of the jet. For the graph found in Eq. (5.8), this is once again the EFP most closely corresponding to jet mass (given in Eq. (3.26). Together with the isolation cones, these observables reach an AUC of 0.842 and an ADO with the PFN of 0.888, see Table 5.1.

This set of observables performs better than the CNN and largely closes the performance gap with the best calorimeter cell networks (PFN), indicating that angular information is especially relevant to the muon isolation classification task. Distributions of these EFPs for signal and background are shown in Fig. 5.5. Further scans in this limited space do not yield significant boost in AUC or ADO values. The strong result of the IRC-safe EFN indicates that it is possible to capture nearly all of the classification power using IRC-safe graphs, likely requiring graphs with complexity beyond what has been considered.

A scan guided by the CNN rather than the PFN yields very similar results, with identical choices for the first three EFPs.

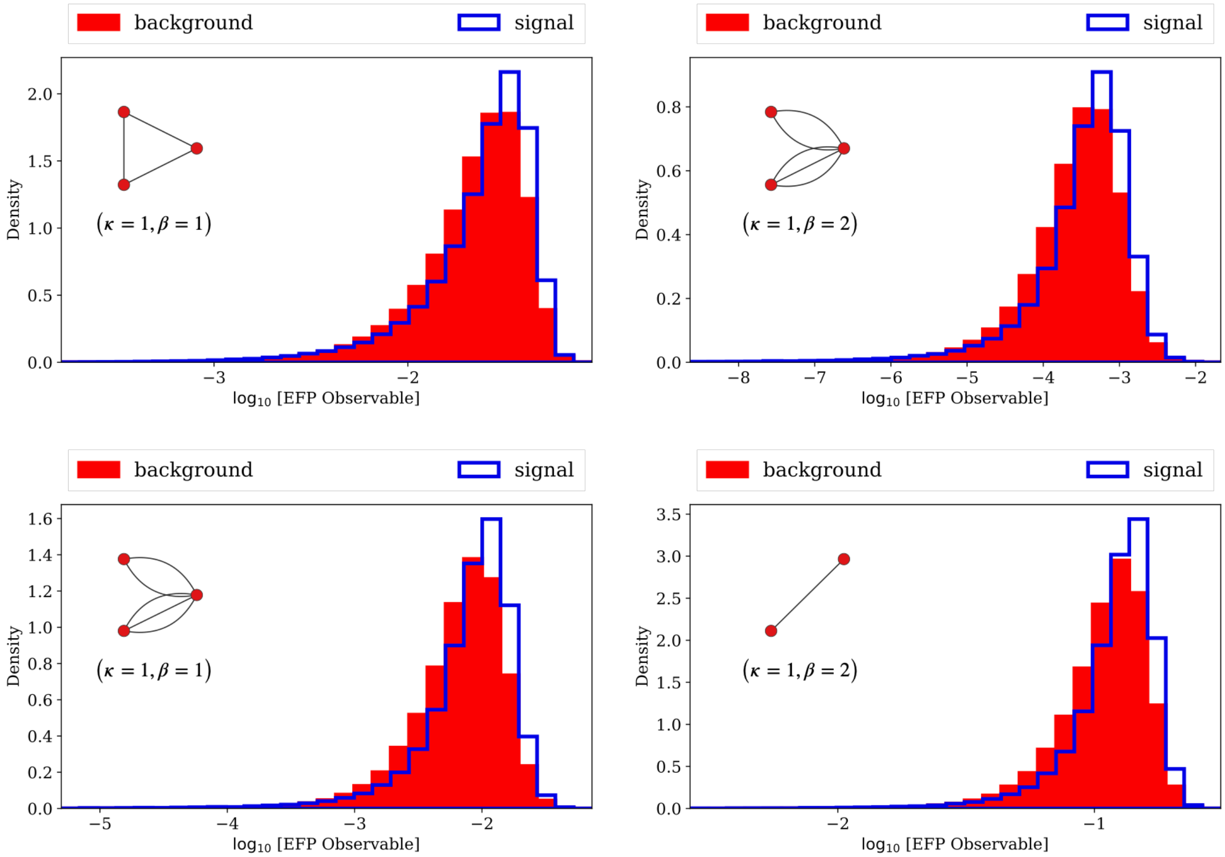


Figure 5.5: Distributions of the $\log_{10}$ for selected EFPs in the IRC Safe but restricted space of maximum 3 nodes and 3 edges group. EFPs were selected by a black-box guided process using a PFN model.

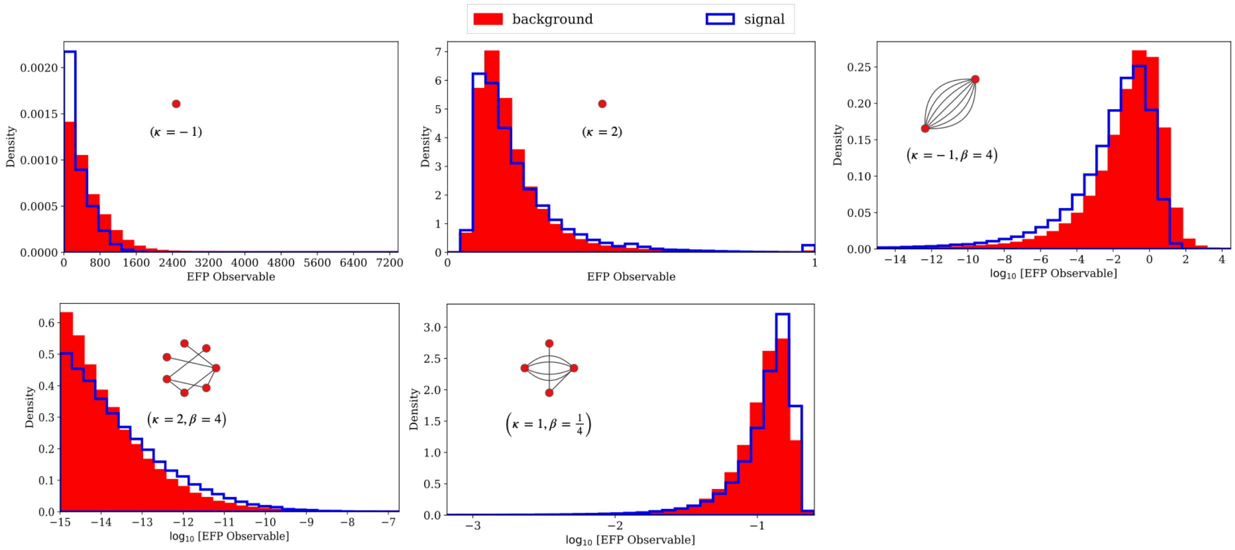


Figure 5.6: Distributions of the $\log_{10}$ for selected EFPs in the complete space of EFPs (not IRC Unsafe and no restrictions on complexity, κ or β). EFPs were selected by a black-box guided process using a PFN model.

5.4.2 IRC-unsafe Observables

To understand the nature of the remaining information used by the PFN but not captured by the isolation cones and the IRC-safe observables, the search space is expanded to include observables which are not IRC safe ($\kappa \in \left[-1, 0, \frac{1}{4}, \frac{1}{2}, 1, 2\right]$), with alternative angular powers ($\beta \in \left[\frac{1}{4}, \frac{1}{2}, 1, 2, 3, 4\right]$) and with up to $n = 7$ nodes and $d = 7$ edges.

A scan of these observables finds a set of five which, when combined with the isolation cones and p_{T} reach an AUC of 0.857. Fig. 5.6 gives the distribution for prompt and non-prompt muons parameterized as the selected five EFPs. Unconstrained in parameters, this process first selects two single point correlators with IRC Unsafe values of κ followed by three multi-point correlators with many more nodes and edges than in the IRC Safe reduced set. The preference for large edge counts in many selected EFPs might indicate a sensitivity for large-angle effects when isolating prompt muons. This analysis should be tempered by the observation that, due to the overlapping nature of the large space of EFPs, there are several

Input/Method				AUC	ADO	Parameters
Iso Cone(s)	p_{T}	EFP	Image			
1				0.787	0.860	40k
10				0.803	0.877	41k
10	✓			0.807	0.884	42k
10	✓	4 (simple)		0.842	0.888	42k
10	✓	5		0.857	0.900	43k
			CNN	0.841	0.950	167k
			EFN	0.849	0.951	453k
			PFN	0.857	1	453k

Table 5.1: Summary of performance (AUC) in the prompt muon classification task for various network architectures and input features. Statistical uncertainty in each case is ±0.001 with 95% confidence, measured using bootstrapping over 100 models. Uncertainty due to the initial conditions of the network is found to be negligible. Also shown are the number of parameters in each network.

sets of EFPs which achieve similar performance. One again, repeating this scan guided by the CNN rather than the PFN yields very similar performance results and EFP choices.

5.5 Discussion

The performance of the networks which use the low-level calorimeter cells indicates that information exists in these cells which is not captured by the isolation cones, see Table 5.1. A guided search through the space of IRC-safe EFPs closes most of the gap between these networks, giving us some insight as to the nature of the information. A broader search is able to complete the bridge, yielding the same performance as the low-level network, but employing IRC-unsafe EFPs.

A comparison of the network complexity for the various approaches is shown in Table 5.1. The set of high-level features (isolation cones and EFP graphs) matches the PFN performance with 10 times fewer parameters, supporting the notion that the high-level features are effectively summarizing the relevant low-level information.

With the application of deep neural networks on low-level calorimeter images, the separation of prompt muons from a non-prompt muon background was demonstrated to have superior power to traditional methods using isolation cones. Attempts to compensate for individual isolation cones with the inclusion of more variations in the characteristic radius were similarly unable to approach the low-level methods shown. This performance gap indicates the presence of considerable classification information accessible through potential non-radial structure in the calorimeter cells which significantly improves this isolation task.

Using a guided search with the best performing low-level network, a small set of IRC Safe EFPs was selected which nearly recovers the entire performance of the PFN. As these inputs are simple functions of the energy deposition, they can be physically interpreted, and the fidelity of their modeling can be studied in control regions in collider data. These boosts in the efficiency to identify prompt muons are extremely valuable to searches at the LHC, especially those with multiple leptons, where event-level efficiencies depend sensitively on object-level efficiencies.

Additional, albeit more complex EFPs, have been shown to give an even stronger boost to performance and can successfully match the most powerful low-level networks. The contrast between the IRC Safe and IRC Unsafe guided search results, along with the comparison of EFN and PFN results, provides useful insights into the relevance of IRC Safety for information relevant to prompt muon isolation. Their relationship suggests that most of the additional content missing from isolation cone based studies is accessible through purely IRC-safe information.

More broadly, the existence of a gap between the performance of state-of-the-art high-level features and networks using lower-level calorimeter information provides an improved solution for the suppression of lepton backgrounds. In contrast to using black-box networks directly, particle isolation can be translated and performed with physically understandable inputs. This allows the physicist to understand the nature of the information being used and to assess its systematic uncertainty.

Chapter 6

Exploring Dark Matter with Interpretable Learning

6.1 Introduction

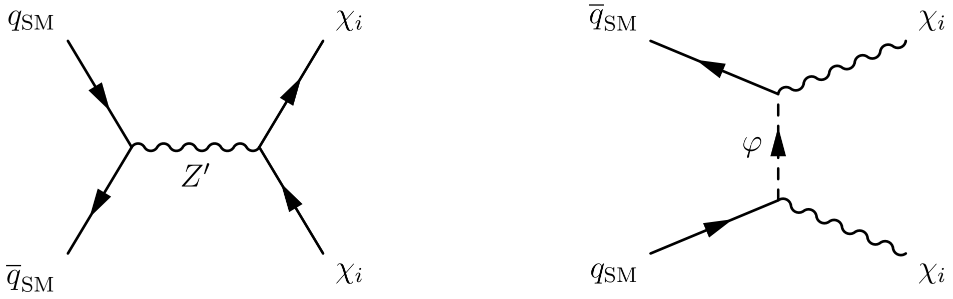
The microscopic nature of dark matter (DM) remains one of the most pressing open questions in modern physics [163–165], and a robust program of experiments search for evidence of its interaction with the Standard Model (SM) sector. These experiments typically assume that DM is neutral, stable and couples weakly to SM particles [166, 167]; in collider settings this predicts detector signatures in which weakly-produced DM particles are invisible, evidenced only by the imbalance of momentum transverse to the beam. No evidence of DM interactions has been observed to date.

However, while these assumptions are reasonable, the lack of observation motivates exploring scenarios in which one or more of them are relaxed. Specifically, if DM contains complex strongly-coupled hidden sectors, it may lead to the production of stable or meta-stable dark particles within hadronic jets [168, 169]. Depending on the fraction of the jet which results in dark-sector hadrons, it may be only “semi-visible” to detectors, leading to a unique pattern of energy deposits, or jet substructure.

A robust literature exists for the identification of jet substructure, with applications to boosted W-boson, Higgs boson and top-quark tagging, in which observables are designed to distinguish jets with a single core from those with several hard subjets due to the hadronic decay of the heavy boosted particle. While these observables have some power when adapted to the task of identifying semi-visible jets [170], no observables have yet been specifically designed to be sensitive to the unique energy patterns of semi-visible nature of jets.

In parallel, the rapid development of machine learning to the analysis of jet energy depositions has demonstrated that jet tagging strategies can be learned directly from lower-level jet constituents. Such learned models are naturally challenging to interpret or validate, especially given the high-dimensional nature of their inputs. However, techniques introduced in Chps. 3, 4 and 5 have been successful at translating the learned model into interpretable high-level observables.

In this chapter, the first study of deep networks trained to distinguish semi-visible jets from QCD background jets is shown using the patterns of their low-level jet constituents. A comparison is then made between a deep network's performance and that obtained for a similar network using a set of existing high-level observables. The relative performance between these two strategies is then translated between the deep network's approach into a small set of new observables which approximately replicate its decisions and performance. Interpretation of these observables can yield insight into the nature of the energy deposition inside semi-visible jets.



(a) s-channel

(b) t-channel

Figure 6.1: Dark Matter production via an s-channel and t-channel process yielding semi-visible jets

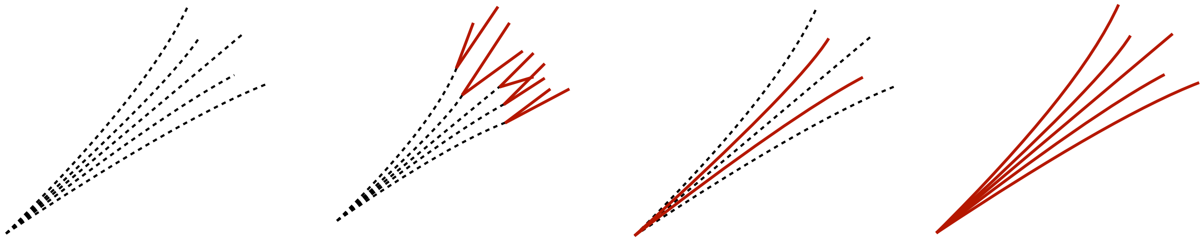
6.2 Exploring Semi-Visible jets

Following Ref. [168], the production of dark-sector quarks are considered with several flavors, ($\chi_i = \chi_{1,2}$), via a messenger section which features a Z' gauge boson that couples to both SM and DM sectors; see Fig. 6.1. The dark quarks produce QCD-like dark showers, involving many dark quarks and gluons which produce dark hadrons, some of which are stable or meta-stable and some of which decay into SM hadrons via an off-shell Z'

The detector signature of the resulting jet depends on the lifetime and stability of the dark hadrons, leading to the four possible states given in Fig. 6.2. Though the physics is complex and sensitively dependent on the details of the dark sector structure, a description of the dark and SM hadrons produced by a DM model quark can be encapsulated in the quantity $r_{\rm inv}$ (Eq. (6.1)), the ratio of dark stable hadrons to all hadrons in the jet:

$$r_{\rm inv} \equiv \left\langle \frac{\#\text{ of stable dark hadrons}}{\#\text{ of hadrons}} \right\rangle \tag{6.1}$$

Given the production of a dark quark which yields a jet, $r_{\rm inv}$ describes the relative quantity of hadrons in that jet which present as visible SM hadrons and which become invisible through access to the dark sector. An invisible fraction of $r_{\rm inv} = 0.0$, for example, corresponds to



(a) Stable Dark Jet (b) Long-lived Dark Jet (c) Semi-Visible Jet (d) Rapid Decay

Figure 6.2: jet production modes for dark quark hadronization into visible hadrons or the dark sector. For a dark decay, unstable dark hadrons decay to SM quarks (red, solid) while stable dark hadrons remain in the dark sector (black, dashed).

a dark quark producing a jet consisting of only visible hadrons. This is equivalent to the example of *Rapid Decay* given in Fig. 6.2d. Alternatively, an invisible fraction of $r_{\text{inv}} = 1.0$ describes a stable dark jet (Fig. 6.2a), in which the dark quark hadronizes exclusively in the dark sector and generates no visible particles. A stable dark jet would only be detectable through indirect measurements, primarily the presence of missing energy (MET). For any intermediate value of r_{inv}, jets will contain a visible and invisible fraction, leading to a mixture of energy deposits into the hadronic calorimeter and MET along the dijet axis (Fig. 6.2c).

6.3 Sample Generation and Data Processing

Samples of simulated events with semi-visible jets are generated using the modified Hidden Valley[171] model described in Ref. [172] for both an s-channel (Fig. 6.1a) and t-channel (Fig. 6.1b) process. Simulation of the hard process $pp \to Z' \to \chi_1\overline{\chi}_1$ at center-of-mass energy of $\sqrt{s} = 13\,\text{TeV}$ are performed in MadGraph5 [106] (v2.6.7) with `xqcut`=100 and the NNPDF2.3 LO PDF set[173]. The mediator mass is set to $M_{Z'} = 1.5\,\text{TeV}$ and the dark matter candidate mass of $M_\chi = 10\,\text{GeV}$. Distinct sets were generated for invisible fractions of $r_{\text{inv}} \in [0.0, 0.3, 0.6]$. Up to two extra jets are generated and MLM matched[174].

Showering and hadronization is performed with Pythia8 v8.244 [107] with detector simulation and reconstruction in Delphes v3.4.2 [108] using the default ATLAS card.

A sample of SM jets from a typical QCD processes is generated from a process of $p\,p \to j\,j$. The same simulation chain used for SVJ production is applied to the SM jets. Jets are clustered using the anti-k_{T} [23] algorithm in pyjet[175] with a jet-radius parameter of $R = 1.0$. Sub-jets are identified in each jet, using a jet-radius parameter $R_{\mathrm{sub}} = 0.2$, for later analysis of the jet substructure. Leading jets are required to have $p_{\mathrm{T}} \in [300, 400]$GeV and subjets are required to have a p_{T} of greater than 5% of the leading jet.

For each event generated, the leading jet is selected and truth matched to guarantee the presence of a dark quark within the region of $\Delta R < 1$. The final dataset produced for training classification models is, therefore, truth matched leading jets for the invisible fractions $r_{\mathrm{inv}} \in [0.0, 0.3, 0.6]$.

After all cuts and selection requirements, 2×10^6 simulated jets remain with a 50/50 split between signal (SVJ) and background (QCD). To avoid inducing biases from artifacts of the generation process, signal and background events are re-weighted such that the distributions in p_{T} and η are uniform. Distributions are given in Fig. 6.3, where weighted histograms show a matching spread for both parameters between signal/background.

6.3.1 High-Level Observables

A large set of jet substructure observables[75, 76, 176] have been proposed for a task different from the focus of this study, that of identifying jets with multiple hard subjets. Nevertheless, these observables may summarize the information content in the jet in a way that is relevant for the task of identifying semi-visible jets[170], and so serve as a launching point for the search for new observables.

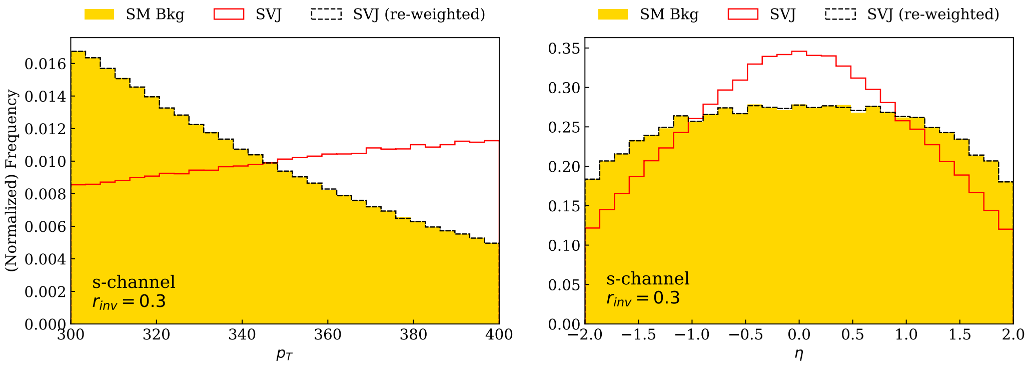


Figure 6.3: Distribution of SVJ transverse momentum (p_{T}) and pseudorapidity (η) for signal and background samples. An example for the s-channel process with $r_{\mathrm{inv}} = 0.3$ is given showing signal (red), background (yellow) and re-weighted signal (black, dashed) distributions.

This set of high-level observables includes: jet mass (M_{jet}), jet p_{T} sum, p_{T}^D, Les Houches Angularity (LHA), N-subjettiness ratios $\tau_{21}^{\beta=1}$ and $\tau_{32}^{\beta=1}$[25], and Energy Correlation function ratios $C_2^{\beta=1}$, $C_2^{\beta=2}$, $D_2^{\beta=1}$, $D_2^{\beta=2}$, e_2, e_3, e_{width}[24, 80, 118] and the splitting function z_g[176]. In each case, observables are calculated from the list of trimmed jet constituents described in Sec. 6.3. Descriptions for each observable and distributions are provided in Sec. D.1.

6.4 Machine Learning and Evaluation

For both the low-level trimmed jet constituents and high-level jet substructure observables, a variety of networks and architectures are tested. The goal in this step is to establish the maximum performance captured by each form of the data and seek a potential performance gap between the low-level and high-level features. Based on past success from work presented in Chps. 3, 4 and 5, a deep neural network using dense layers is used to train a binary classifier. Additionally, XGBoost[177] and LightGBM[178], two increasingly popular ML strategies in the HEP literature, are also tested. In addition to improved training and prediction speeds over deep networks using dense layers, XGBoost and LightGBM have shown good performance in training high-level classifiers with jet substructure for performing class separation on high-level features[179–181]. Measuring performance across these three networks in Table 6.1, LightGBM is selected as the the best performing high-level classifier.

In the case of low-level classifiers, convolutional neural networks on jet images are considered, motivated by results given in Refs. [27, 28, 95] and similarly good performance found in Chps. 3, 4 and 5. For the specific task of classifying jet substructure observables, Energy Flow Networks (EFN) and Particle Flow Networks (PFN) are a natural inclusion for low-level classifiers[65].

After preliminary testing, convolutional networks on jet images were found to perform generally worse than Flow Networks across all selections of r_{inv}. This matches the result found when training classifiers for muon isolation in Sec. 5.3. Given the Flow Networks explicit approach to learning the problem in terms of jet substructure functions, it's not surprising that it performs well in this task. Specifically, a PFN trained on low-level trimmed constituent features performs better than all other low-level and high-level strategies. ROC curves for both the PFN (low-level) and LightGBM (high-level) models are given in Fig. 6.4. Additional performance details of other high-level strategies are also given, for context, in Table 6.1. Details for network training and hyperparameter selection are provided in Sec. D.2.

6.4.1 Comparing High-Level and Low-Level

Comparing performance across selections of invisible fraction (r_{inv}), the low-level approach yields equivalent or superior performance to even the best high-level strategy. This result is reasonable given that the high-level features are generated from the same data used in the PFN training and, therefore, the high-level observables are strictly a subset of the information available to the low-level networks.

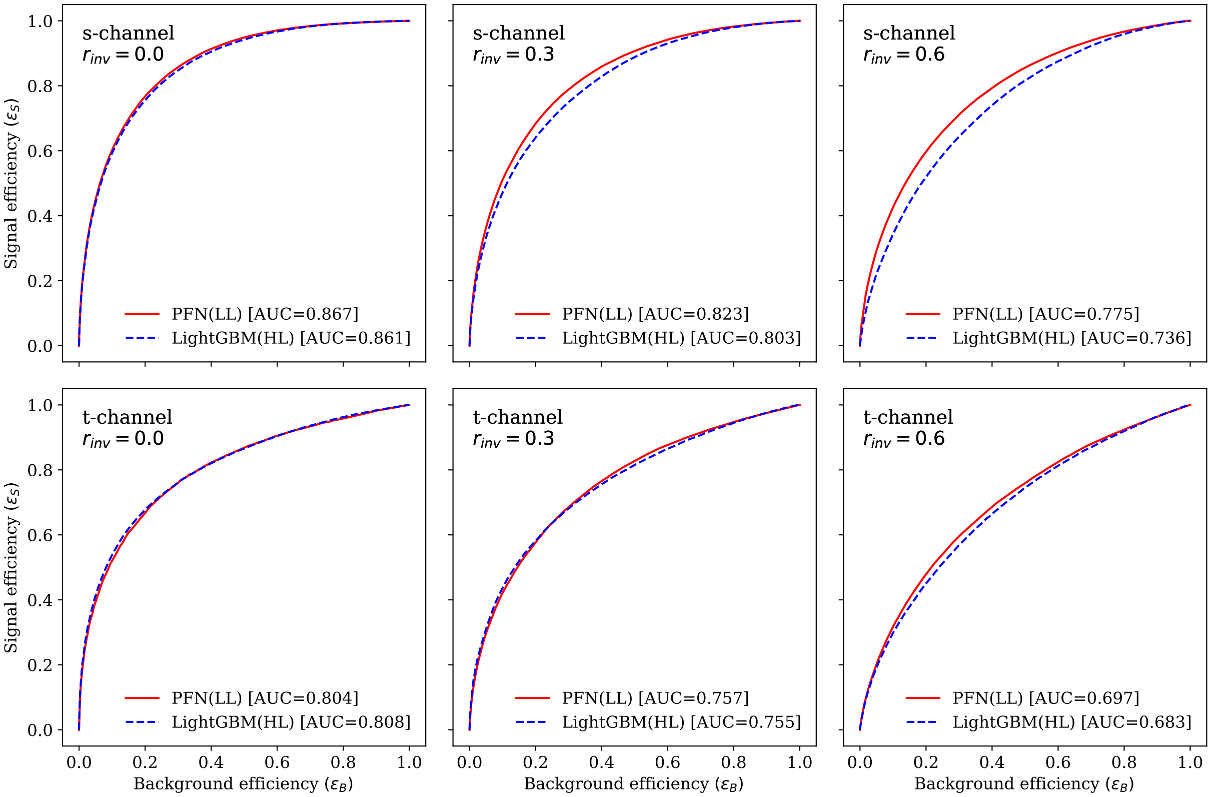


Figure 6.4: Background rejection (defined as the inverse of background efficiency) versus signal efficiency for the best performing PFN architecture (red solid line) and HL network (blue dashed line).

The largest performance gap appears when comparing the LightGBM model on high-level features for $r_{\text{inv}} = 0.6$ in the s-channel process (AUC = 0.736) with the PFN on low-level features (AUC = 0.775). However, a modest improvement in AUC is also captured in the case of the $r_{\text{inv}} = 0.0$ and $r_{\text{inv}} = 0.3$ invisible fractions in the s-channel process and $r_{\text{inv}} = 0.6$ in the t-channel process.

Two combinations from the t-channel process data (i.e. $r_{\text{inv}} = 0.0$ and $r_{\text{inv}} = 0.3$) have matching performance when comparing high-level and low-level classification. This matching AUC could mean that the peak performance is already captured by the broad set of jet substructure observables chosen for those specific regions of invisible fraction. However, the ADO between the LightGBM and PFN strategy is well below unity. This would imply that their individual solutions are distinct and there may be room to improve the classification accuracy by considering a network that uses both sets of features.

6.4.2 Initial Reduction with SHAP and LightGBM

The selections for high-level observable as given in Sec. 6.3.1 utilizes an "everything but the kitchen sink" approach and includes a broad and over-complete set of substructure observables. This can be a useful strategy where only training performance is considered, but the relative impact of individual features is generally obfuscated in the process. This makes interpretation of the physical meaning and context difficult. In an initial attempt to reduce the high-level feature set, quantify feature impact and optimize performance, SHAP analysis is applied to the high-level model.

SHAP (SHapley Additive exPlanations), originally discussed in Sec. 2.2.2, is a game theoretic approach to explaining the final predictive outputs for a classifier in terms of the original input features it receives[35]. This is accomplished, in the case of a high-level explainer, by training 2^N models (for all permutations of N input features) and measuring

s-channel

Features	$r_{\text{inv}} = 0.0$		$r_{\text{inv}} = 0.3$		$r_{\text{inv}} = 0.6$	
	ADO[PFN]	AUC	ADO[PFN]	AUC	ADO[PFN]	AUC
PFN(LL)	1	0.867	1	0.823	1	0.775
DNN(HL)	0.867	0.860	0.833	0.799	0.804	0.734
LightGBM(HL)	0.858	0.861	0.839	0.803	0.819	0.736
XGBoost (HL)	0.863	0.861	0.836	0.803	0.816	0.738

t-channel

Features	$r_{\text{inv}} = 0.0$		$r_{\text{inv}} = 0.3$		$r_{\text{inv}} = 0.6$	
	ADO[PFN]	AUC	ADO[PFN]	AUC	ADO[PFN]	AUC
PFN(LL)	1	0.812	1	0.757	1	0.697
DNN(HL)	0.810	0.801	0.798	0.753	0.774	0.676
LightGBM(HL)	0.845	0.808	0.804	0.755	0.790	0.683
XGBoost (HL)	0.834	0.803	0.813	0.757	0.787	0.682

Table 6.1: Summary of performance (AUC and ADO) in the SVJ classification task for various network architectures and input features. Statistical uncertainty in each case is less than ±0.002 with a 95% confidence, measured using bootstrapping over 200 models.

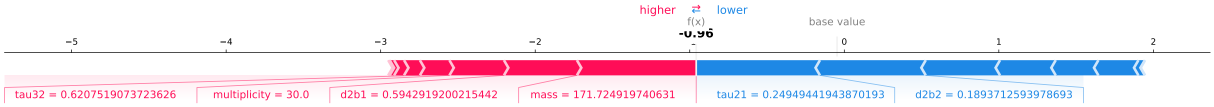


Figure 6.5: SHAP force plot for a single training sample using high-level jet substructure observables in a LightGBM model. Features for this example are taken from the s-channel process with invisible fraction $r_{\text{inv}} = 0.3$.

the weighted impact individual features contribute to the final prediction. A comparison can then be made, averaged over many training samples, to the mean contribution an input feature provides to a models predictive decisions.

Conveniently, tools have have been designed for the application of fast SHAP analysis on LightGBM models [182]. SHAP analysis was applied to each high-level model used across selections of r_{inv} in the SVJ dataset. An example force plot of jet substructure observables for an individual jet sample after evaluation with SHAP is given in Fig. 6.5. This figure shows the degree to which each input "pushes" or "pulls" the base value (i.e. the networks initialized prediction before evaluation by the model) towards its final predicted value. The particular jet given in Fig. 6.5 would suggest that N-Subjettiness $\left(\tau_{21}^{\beta=1}\right)$ and jet mass (M_{jet}) strongly impact the model towards predicting a lower value from the initial mean prediction. Conversely, e_2 and $\tau_{32}^{\beta=1}$ make the opposite suggestion in the model, and to a much smaller degree of overall influence. The remaining features had minimal impact on prediction for this singular event. Many of these force plots are measured and weighted across all events to produce a summary plot, given in Fig. 6.6. Applying this summary analysis for all examples of process and invisible fraction, a similar ordering of observables appears, with the most impactful inputs typically including: $\tau_{21}^{\beta=1}$, $C_2^{\beta=2}$, p_{T}, e_2 and multiplicity. On the other end of the spectrum, splitting function (z_g), e_{width}, and LHA are comparatively unimportant to the model predictions.

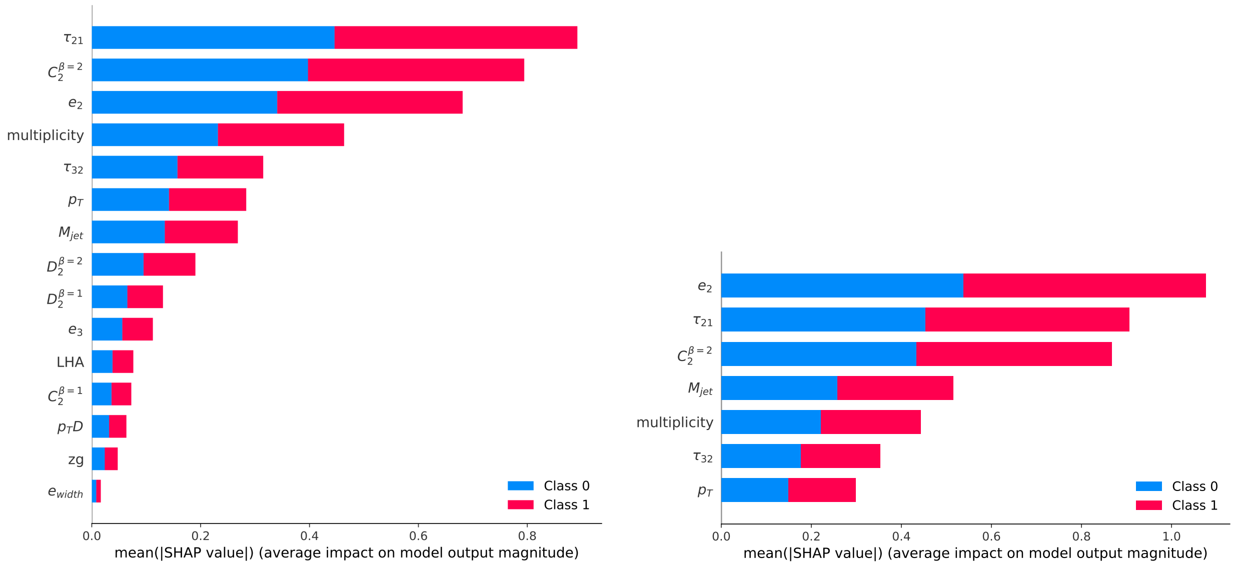


Figure 6.6: SHAP summary plot for a complete LightGBM model using 13 high-level features (left) and 7 high-level features (right) . Results are given for the example case of an s-channel process with invisible fraction $r_{\text{inv}} = 0.3$.

To evaluate the performance impact on a reduced sets of features, additional LightGBM models were trained with subsets of the 13 original high-level observables removed. The removed observables were chosen based on the lowest performing examples in the SHAP summary given in Fig. 6.6. After iteratively removing inputs until a change in performance was measured by AUC, a minimal set of seven jet substructure observables were found. These inputs were capable of maintaining the same AUC performance (within statistical uncertainty) across s-channel and t-channel processes for all r_{inv} given in Table 6.1. This reduced feature set includes:

$$7\,\text{HL} \equiv \left[e_2, \tau_{21}^{\beta=1}, C_2^{\beta=2}, p_{\text{T}}, M_{\text{jet}}, \tau_{32}^{\beta=1}, \text{multiplicity}\right], \tag{6.2}$$

The remaining input features could be removed without any change to classification accuracy. This suggests that SHAP analysis has effectively isolated unnecessary features that contain information already captured by the best performing inputs (or through pairwise combinations of them) and can generate a reduced set of observables to search from.

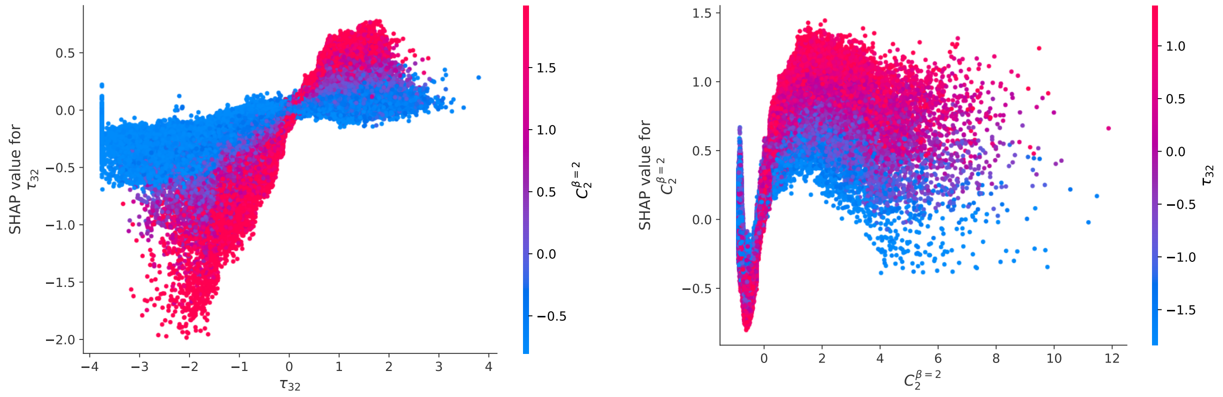


Figure 6.7: SHAP dependence plot for $\tau_{32}^{\beta=1}$ (top) and $C_2^{\beta=2}$ (bottom) for a LightGBM model trained on the reduced set of seven high-level. Features for this example are taken from the s-channel process with invisible fraction $r_{\text{inv}} = 0.3$.

However, interestingly, running a secondary SHAP analysis on the subset of 7HL features results in a different ordering of feature importance, as seen when comparing the full feature list and the reduced feature list (Fig. 6.6). This is an expected result as the best solution for a different set of inputs, even a subset of the larger example, will change for decisions with both individual and pairwise information. The increase in significance of e_2 is likely caused by the loss of similar information accessible through the dropping of e_3. Meanwhile, $\tau_{21}^{\beta=1}$ and $\tau_{32}^{\beta=1}$ contains overlapping information with one another and this should reduce their individual contributions when applied on the 7HL example.

Such interaction effects between pairs of inputs are captured by SHAP in *dependence* plots, like those shown in Fig. 6.7. For each observable, SHAP isolates the second observable with the strongest correlation between their measured SHAP values. Pairwise dependence for $\tau_{32}^{\beta=1}$ appears to be strongest with $C_2^{\beta=2}$ (and vice versa) which similarly suggests the inclusion of both, although leading to improved performance, introduces overlapping information.

6.5 Finding New Observables

From the benchmarking in Sec. 6.4.1, the PFN on calorimeter cells is seen to be the best performing strategy. However, the objective for this study is not merely to find the optimal binary classifier for this problem. Rather, the goal is to understand the underlying physics used by the PFN and to translate this information into a meaningful and low-dimensional physical feature, likely from the EFP space (Sec. 3.3.2).

From previous sections, specifically Sec. 3.4.1, it has been established that this task can be accomplished with a black box guided approach. However, given the success of SHAP demonstrated in Sec. 6.4.2, one might suspect that this procedure could be avoided by training a classifier on all candidate observables (i.e. the original high-level inputs and all EFPs) and allowing SHAP to measure the feature significance for those inputs. Given the size and scope of the EFP space, this kind of brute force method becomes problematic. Recall from Sec. 3.3.1, the ADO similarity for high-level features was measured between one another (Table 3.1 and Fig. 3.2). Jet substructure observables tend to be "over-complete" in their information and, as such, their pair-wise and higher order interactions and the use of mutual information between them complicates the task of disentangling their influence on a model. Additionally, the performance gap between high-level and low-level often hinges on very small optimizations found by the low-level network. Attempts to use large sets of training features often fails to find improvements through these minor trends in the data as the dimensionality of the input data grows. Therefore, the greedy search is used in an attempt to isolate individual EFPs by iterative measurement of the ADO between competing networks.

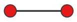
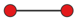
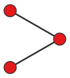
6.5.1 Guided Iteration

A black box guided search is applied for each invisible fraction of r_{inv}, where the PFN is used as a low-level guide for the selection of EFPs to be trained by a high-level LightGBM classifier. In all cases where performance between the high-level and low-level strategy improves, the guided search reaches its best result with the inclusion of just one extra EFP and yields no significant benefit with additional iterations. The selected EFP for each combination of process and invisible fraction are given in Table 6.2. Separate trials including both 7HL and 13HL were performed with no significant difference in AUC performance or ADO similarity with the PFN.

Comparing both AUC performance and ADO similarity between matching processes and invisible fractions, some modest improvements are found. Starting with the broadest search, including EFPs with dimension $d \leq 5$ and a wide selection of IRC Safe and unsafe parameters with $\beta \in \left[\frac{1}{2}, 1, 2\right]$ and $\kappa \in [-1, 0, 1, 2]$ (Table 6.2), an overall trend is found for the inclusion of the IRC unsafe choice of $\kappa = -1$. Due to this parameters influence over the momentum fraction, a negative value for κ provides an observable sensitive to soft emissions and small p_{T} features. The dot graph consistently selected in the s-channel process holds no angular dependence. For those selections that do in the t-channel process, the parameter $\beta = \frac{1}{2}$ is always chosen. A small value of this angular parameter would imply benefits for observables sensitive to small angle resolution.

A second instance of this search was run with a smaller collection of EFPs. Specifically, observables with no more than three nodes and edges were included and only parameters of $\kappa = 1$ and $\beta \in [1, 2]$ were studied. This modification tasks the network with finding a similar solution with simpler graph structures and exclusively IRC Safe information. It also biases training benefits to come primarily through graph construction as the momentum-fraction parameter becomes fixed and variety of angular resolution is reduced. Comparing results

Full EFP Set ($d \leq 5$ with parameters $\kappa \in \left[-1, 0, \frac{1}{2}, 1, 2\right]$ and $\beta \in \left[\frac{1}{2}, 1, 2\right]$)

r_{inv}	s-channel			t-channel		
	Included EFP	ADO[PFN]	AUC	Included EFP	ADO[PFN]	AUC
0.0	$(\kappa = -1)$	0.870	0.865	 $\left(\kappa = -1, \beta = \frac{1}{2}\right)$	0.851	0.822
0.3	$(\kappa = -1)$	0.840	0.805	 $\left(\kappa = -1, \beta = \frac{1}{2}\right)$	0.815	0.759
0.6	$(\kappa = -1)$	0.820	0.739	 $\left(\kappa = -1, \beta = \frac{1}{2}\right)$	0.801	0.689

Reduced EFP Set ($d \leq 3$, no more than 3 nodes with parameters $\kappa = 1$ and $\beta \in [1, 2]$)

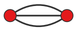
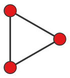
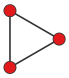
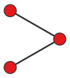
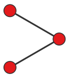
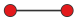
r_{inv}	s-channel			t-channel		
	Included EFP	ADO[PFN]	AUC	Included EFP	ADO[PFN]	AUC
0.0	 $(\kappa = 1, \beta = 1)$	0.868	0.863	 $(\kappa = 1, \beta = 2)$	0.846	0.822
0.3	 $(\kappa = 1, \beta = 2)$	0.839	0.804	 $(\kappa = 1, \beta = 2)$	0.807	0.759
0.6	 $(\kappa = 1, \beta = 1)$	0.819	0.736	 $(\kappa = 1, \beta = 1)$	0.792	0.687

Table 6.2: Results for a guided search classifying SVJ from QCD background when trained via a LightGBM model using settings given in Sec. D.2.3. Examples are given for both the complete set of generated EFPs (top) and for a reduced set of IRC-Safe and simple graphs (bottom).

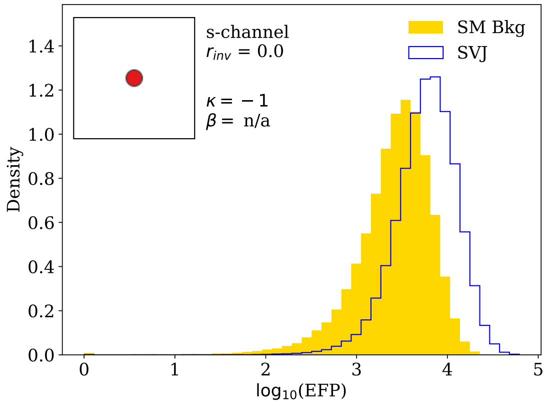


Figure 6.8: SVJ signal (blue) and QCD background (yellow, solid) distributions for the dot graph selected in the $r_{\text{inv}} = 0.0$ processes for the s-channel process.

in the bottom half of Table 6.2, one sees similar performance with only small degradations. Consistent with results in Chps. 3, 4 and 5, it is not unusual to find that the flexibility of a DNN can isolate approximately equivalent performance with a different set of features.

Full EFP Space and s-channel

In the s-channel process, invisible fraction $r_{\text{inv}} = 0.0$ increases its AUC and closes the performance gap with the PFN within statistical uncertainty. The remaining choices of r_{inv} see only marginal improvements. In each case, the *dot* graph with IRC Unsafe selection of $\kappa = -1$ is preferred. This graph is expressed as a sum over constituents in Eq. (6.3).

$$\left(\quad \bullet \quad \right)^{(\kappa=-1)} = \sum_{a=1}^{N} \frac{1}{z_a} \tag{6.3}$$

This graph is, in effect, simply a measure of the sum of the inverse p_{T} of the jet constituents and works to improve network sensitivity to small p_{T} features. The signal and background distributions for the dot graph in the $r_{\text{inv}} = 0.0$ case is given in Fig. 6.8, where good separation between signal and background is visible. Moving to the reduced set of EFPs for the s-channel selects a two-point and pair of three-point correlators. The AUC in the $r_{\text{inv}} = 0.0$ no longer closes the gap with the PFN, suggesting that the IRC Unsafe information was advantageous for the classification task. The primary benefit in the s-channel process appearing in $r_{\text{inv}} = 0.0$ provides an interesting comparison. Recall that the choice of $r_{\text{inv}} = 0.0$ corresponds to the production of a jet from a dark quark which yields only visible hadrons. The strength of the EFP found in Eq. (6.3) might suggest that when all hadrons are produced in the visible spectrum of a SVJ decay, soft emissions are not well represented in the jet substructure observables given in 13HL and this is a useful discriminator between SVJ jets and QCD jets. As the data moves away from purely visible hadrons and begins to include additional background noise, that low p_{T} information might be obscured to the model, explaining the difficulty in isolating mixed visible and semi-visible hadrons.

Mass-Like Observables for t-channel (0% and 30% Invisible Fractions)

Under the t-channel process, improvements in the $r_{\text{inv}} = 0.0$ and $r_{\text{inv}} = 0.3$ case are both seen when using the same two-point correlator. Although there was only a minor difference between the LightGBM and PFN performance, the inclusion of a mass-like observable results in increased performance for the high-level inputs. This is accomplished with the inclusion of the EFP in Eq. (6.4), which is similar in structure to the EFP's closest approximation to jet mass, Eq. (6.5). The change in value for κ and β suggests that although jet mass is already present in the 13HL and 7HL subset, sensitivity to small angular separation and low

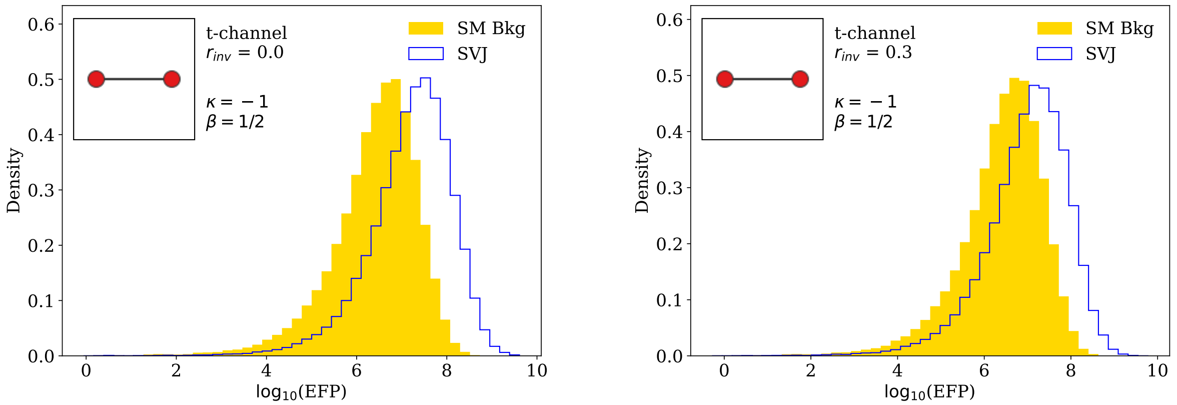


Figure 6.9: SVJ signal (blue) and QCD background (yellow, solid) distributions for the mass-like graph selected in the t-channel process with invisible fraction $r_{\text{inv}} = 0.0$ (left) and $r_{\text{inv}} = 0.3$ (right)

p_{T} constituents might be an improved discriminator to include.

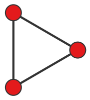
$$\left(\bullet\!\!-\!\!-\!\!\bullet\right)^{\left(\kappa=1,\beta=\frac{1}{2}\right)} = \sum_{a,b=1}^{N} z_a z_b \sqrt{\theta_{ab}} \tag{6.4}$$

$$\left(\bullet\!\!-\!\!-\!\!\bullet\right)^{(\kappa=1,\beta=2)} \approx \frac{M_{\text{jet}}^2}{p_{\text{T}}^2} \tag{6.5}$$

In the case of the $r_{\text{inv}} = 0.0$, specifically, the inclusion of the mass-like graph with 7HL leads to better performance than the original PFN. This result is surprising but not inexplicable. Given the low ADO between 7HL and the PFN, one can see that the two models have arrived at relatively divergent solutions to the problem. The guided search suggesting a missing feature sensitive to low-p_{T} paired with the simplicity of a much lower dimensional set of features might yield improved training results. This extra performance should, in principle, be recoverable by a PFN (likely with more rigorous hyperparameter optimization). Signal and background distributions are given for both mass-like graphs in Fig. 6.9 Comparing to the reduced set of EFPs, both $r_{\text{inv}} = 0.0$ and $r_{\text{inv}} = 0.3$ select a three-point correlator and find equivalent performance to the larger IRC Unsafe group. It's interesting to note that the

$r_{\mathrm{inv}} = 0.0$ graph selected shares a similar design to one of the already included observables. Specifically, the three point correlator e_3 exists in the EFP spaces with graph,

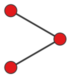
$$e_3^{(\beta)} = \tag{6.6}$$



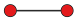
Note, however, that the selected EFP modifies β to have a wider angle sensitivity of $\beta = 2$.

Selections in the t-channel for (60% Invisible Fraction)

Finally, for the invisible fraction $r_{\mathrm{inv}} = 0.6$, performance improvements are seen in the t-channel process but with some performance gap remaining. from the wider set, a three-point correlator is chosen in the IRC Unsafe test and another mass-like EFP is chosen in the IRC Safe test.



$$\Bigg(\qquad\Bigg)^{\left(\kappa=-1,\beta=\frac{1}{2}\right)} = \sum_{a,b,c=1}^{N} \left(z_a z_b z_c\right)^{-1} \sqrt{\theta_{ab}\theta_{bc}\theta_{ac}} \tag{6.7}$$



$$\Bigg(\qquad\Bigg)^{(\kappa=1,\beta=1)} = \sum_{a,b=1}^{N} z_a z_b \sqrt{\theta_{ab}} \tag{6.8}$$

In the case of the mass-like EFP selection, the angular parameter β as been reduced to probe smaller angle resolution. Distributions for both examples are given in Fig. 6.10

6.5.2 SHAP Measurements of Included EFPs

Temporarily returning to SHAP analysis, it's possible to once again check the feature significance of the selected observables but with the inclusion of a black box guided EFP. For those examples where only minimal improvement is made, the SHAP score of the included

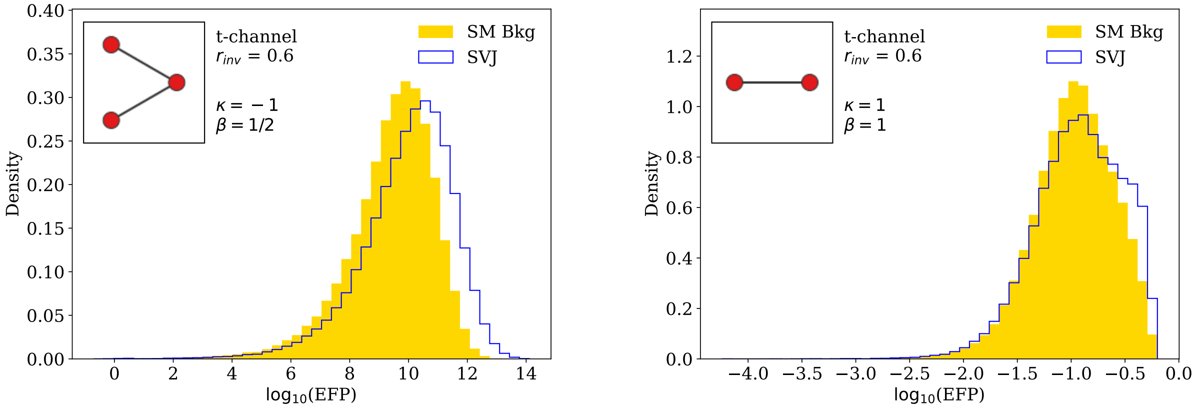
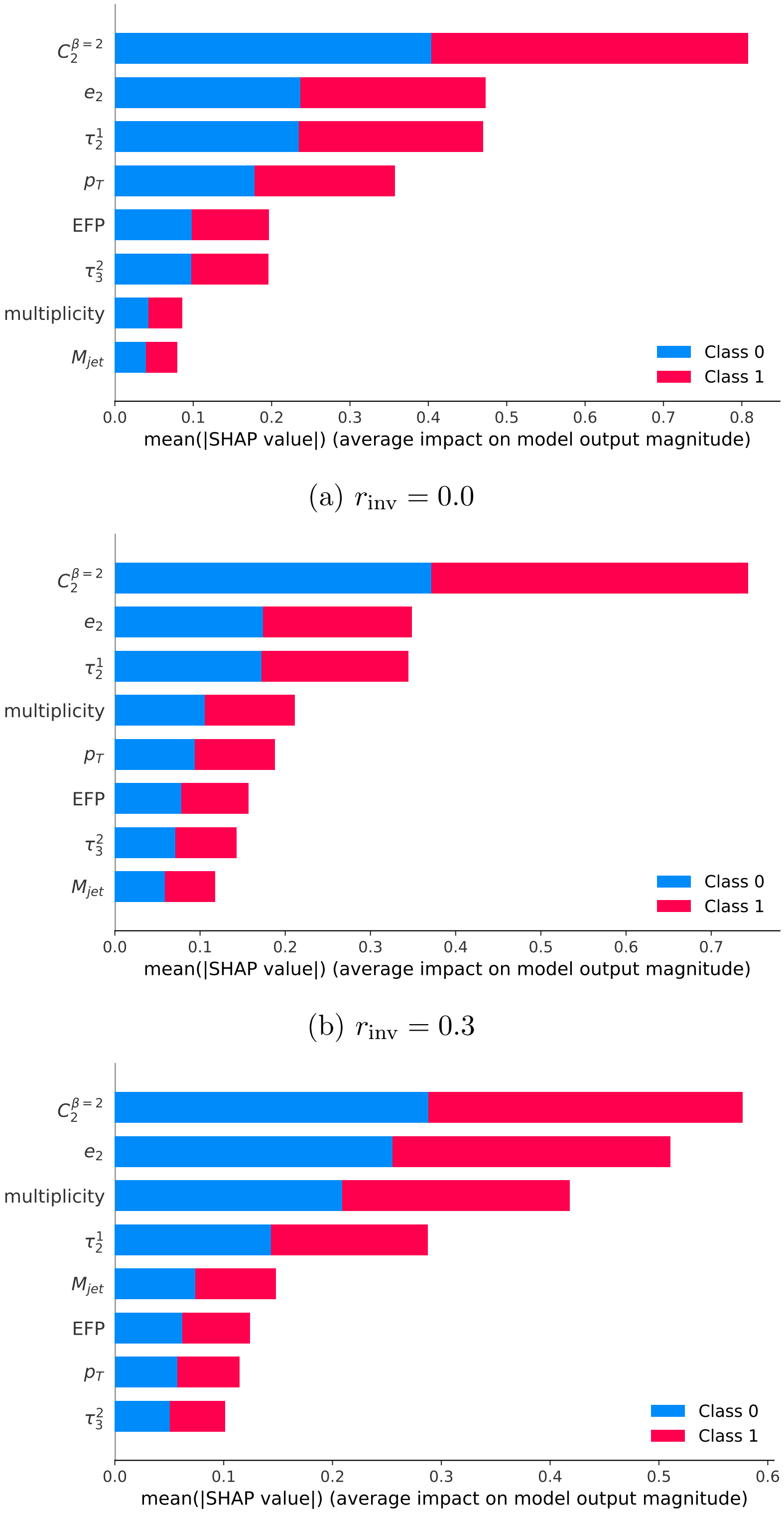


Figure 6.10: SVJ signal (blue) and QCD background (yellow, solid) distributions for the triangular graph selected in the s-channel process (top) and t-channel process (botom) with invisible fraction $r_{\text{inv}} = 0.6$

EFP appears with the lowest score. This result is consistent with the fact that these models, after the inclusion of the best performing EFP, did not utilize this information to benefit classification accuracy in a significant manner. In the case of the t-channel results, where performance gains were made for all invisible fractions, the SHAP score for the included EFPs was much more dramatic, as seen in Fig. 6.11. Comparing SHAP scores after EFP selection lends credence to many of the inferred benefits from Secs. 6.5.1 and 6.5.1. Recall that in the case of $r_{\text{inv}} = 0.0$ and $r_{\text{inv}} = 0.3$ in the t-channel process, a mass-like graph was chosen with a modification to parameters κ and β. The model trained with the inclusion of this EFP (Figs. 6.11a and 6.11b) sees a decrease in the relative model contribution by M_{jet} when compared to the same SHAP analysis without the EFP, Fig. 6.6. In contrast, for invisible fraction $r_{\text{inv}} = 0.6$ where a three-point correlator was selected, M_{jet} remains a more important feature for signal/background discrimination.



(a) $r_{\text{inv}} = 0.0$

(b) $r_{\text{inv}} = 0.3$

(c) $r_{\text{inv}} = 0.6$

Figure 6.11: SHAP performance for LightGBM models on 7HL + 1 EFP in the t-channel process derived through black box guiding.

6.6 Discussion

Starting with the initial benchmark performance numbers, Table 6.1, a low-level strategy is shown to provide additional classification performance in the task of separating SVJ jets from QCD jets. This low-level solution, however, lacks the physical insights into the problem that are available to the use of high-level jet substructure observables. With minimal literature present for the explicit problem of SVJ classification, a broad set of observables were collected and tested.

Trying all possible observables present in the literature is, in some sense, just as bad as using a low-level strategy in the context of model intelligibility. An interpretability framework (like SHAP) can help isolate the relative strength of used features so that low-performing observables can be removed. This simplifies the context of the problem to a more understandable collection of variables and, depending on the complexity of the classification task, may yield performance improvements.

Performing a black box guided search to leverage the classification power of the PFN, model improvements were found for a number of selected EFPs. In most cases, these selections were low dimensional and shared properties with existing jet substructure observables while emphasizing the benefits in a change of momentum-fraction or angular separation parameter.

Chapter 7

Parameterized Neural Networks for Physics Engineered Features

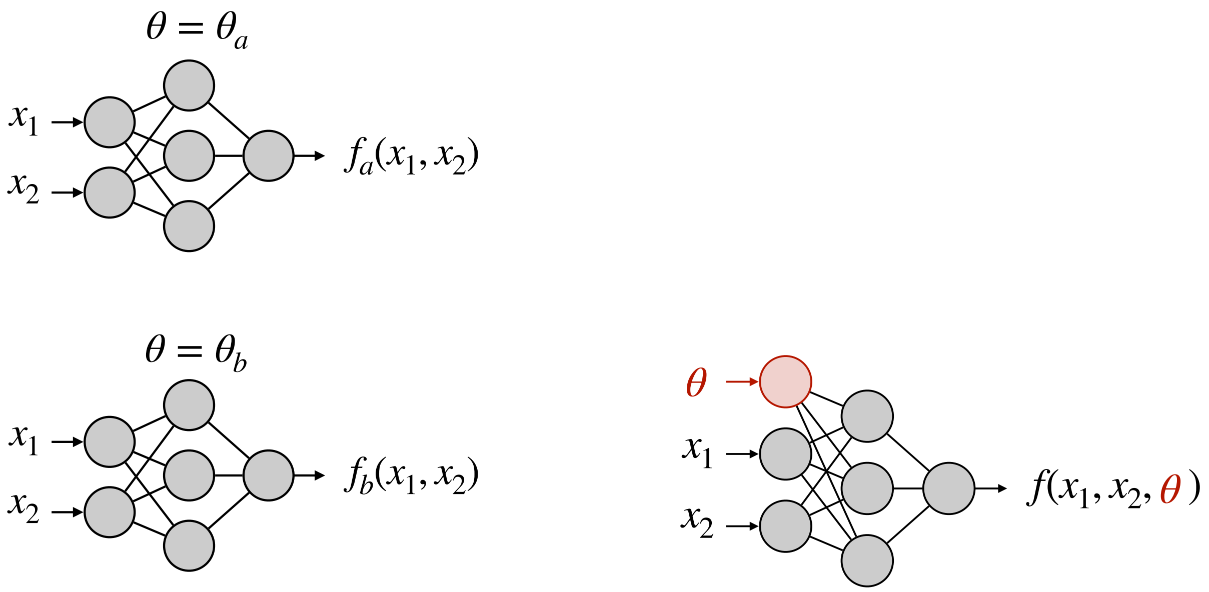
7.1 Introduction

In previous chapters, many examples of neural networks applied to high-energy classification tasks have been given. However, each of these examples has involved individual networks being trained to solve an isolated problem with a specific set of properties. A network trained in that way will not be easily extended to another set of problems, despite any similarities they might share. One common example in high-energy physics is seen in the case of a supervised learning problem for a signal sample with a range of possible masses. A network trained to distinguish signal sample with a characteristic resonance mass from background events would be generally incapable of making the same predictions for another set of signal events with a new mass. The existing solution to this problem is either the training of multiple networks with a separate signal for each signal of interest [17, 183] or the training of a single network that attempts to learn a general solution from a broad array of signal examples. Neither of these approaches are ideal as the former becomes cumbersome and models fail to learn the larger context of the problem and the later, conversely, tends to overgeneralize the problem and perform worse at specific values of interest to narrowly tailored networks.

In this chapter, a method for training a *parameterized neural network* is given in which a single network can be taught to learn the broader classification task and to smoothly interpolate those learned results to mass regions not explicitly given in the signal data. This is accomplished by training modification of the input features to include both traditional event-level features but, additionally, a parameterized input feature relevant to the underlying problem such as a new particle's mass.

7.2 Network Design and Training

A typical ML model takes in features as an input vector $\bar{x}$, where those input features are calculated from a set of event-level quantities. The model then generates a function $f\left(\bar{x}\right)$ which transforms those feature inputs to a prediction. These inputs can, additionally, be modified by some parameter $\bar{\theta}$ and a model which trains on both event-level features and parameters can be defined as $f\left(\bar{x}, \bar{\theta}\right)$. In this framing, a simple network trained and evaluated on a set of inputs $\bar{x}_0$ will evaluate to a real number $f\left(\bar{x}_0\right)$. However, for that network parameterized by θ, outputs for $f\left(\bar{x}_0, \bar{\theta}\right)$ will change as a function of the parameter θ. A graphical representation of the traditional and parameterized approach to these models is given in Fig. 7.1. Given the simplicity of the task, the trained network uses just one hidden layer with three nodes and a sigmoid activation function.



(a) Individual and separate networks for distinct parameters θ_a and θ_b

(b) Single network with a θ input parameter (*parameterized neural network*)

Figure 7.1: *Left*, separate individual networks trained on the same event-level input features but for some unique signal parameter $\theta = \theta_a$ or $\theta = \theta_b$. These networks learn the problem without the context of the parameter θ and will perform poorly for any other choices of θ. *Right* provides a single network which learns parameter θ in addition to inputs (x_1, x_2) during training. This network learns the problem for specific choices of θ_a and θ_b as well as being capable of generalizing to choices of θ not explicitly included in the training examples.

In the case of supervised learning, the model is additionally given an input y, which represents the label for the target class, such that the full network input has the form $\left(\bar{x}, \bar{\theta}, y\right)$. This is in contrast to traditional supervised learning which would, instead, take the form $(\bar{x}, y)$.

7.3 Toy Example

The construction and application of a parameterized network can be shown by a simple toy example in which a supervised model is trained given a simple input feature $\bar{x}$ which is modified by a parameter $\bar{\theta}$. For the input feature, consider a 1-D gaussian distribution feature $\bar{x}$ and distribution mean parameter θ. Features pulled from the gaussian distribution

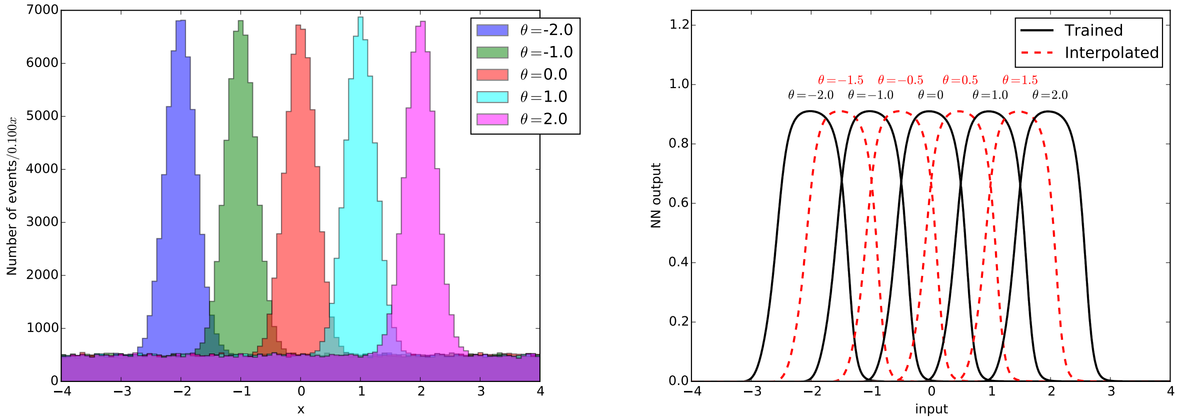


Figure 7.2: *Left*, Toy signal examples (gaussian distributions) at various θ parameters with a uniform background. *Right*, Neural network response as a function of the input features at various parameters θ. Solid curves (Trained) represent explicitly trained signal samples while dashed red lines (Interpolated) are predictions made for parameters θ which the network hasn't seen during training.

represent a signal sample with label $y = 1$ and background samples, $y = 0$ are taken from a uniform background. Signal samples are generated with a width $\sigma = 0.25$ at means of $\theta = -2, -1, 0, 1, 2$.

As demonstrated in Fig. 7.2, the network successfully learns the problem for features it learns directly (whole value parameters of θ) while also learning how to generalize this solution to parameter regions not provided in the training data $\left(\theta = -\frac{3}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}\right)$. In this simple example, the signal/background classification is as accurate for parameters of θ where data does exist in the training set as for values where it does not.

7.4 One-Dimensional Physical Example

A natural extension of the toy example given in Sec. 7.3 can be found in the case of a supervised learning problem for a new particle with an unknown mass. One such example, shown in Fig. 7.3, is the classification problem of two similar top quark decays ($t\bar{t}$) with

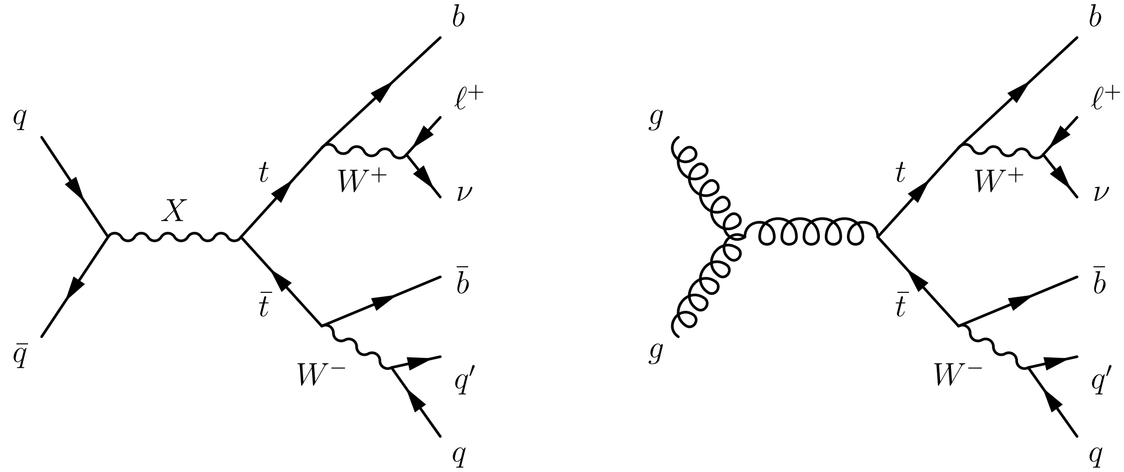


Figure 7.3: Feynman diagrams for the production and decay of a hypothetical $X \to t\bar{t}$ (left) and the dominant background process for top pair production (right). Both processes yield a pair of top quarks and the same final state of a single charged lepton (ℓ), neutrino (ν) and quarks (q, b)

identical final state productions but differing mass distributions. In this example, one can consider the dominant decay mode of a $t\bar{t}$ which yields $t\bar{t} \to W^+bW^-\bar{b} \to qq'b\ell\nu\bar{b}$. In comparison, a new particle X will generate the same final state but with distinct kinematics due to its intermediate resonance.

Similar to the toy example, this problem can be simplified to a one-dimensional problem. Specifically, a network can be trained with a single event-level feature by calculating the reconstructed resonance mass m_{WWbb} using methods given in Ref. [184]. Events are simulated at parton level using MADGRAPH5 [106], showered with PYTHIA8 [173] and reconstructed using DELPHES [108] using the default ATLAS configuration for simulation. Distributions are given in Fig. 7.4 for the calculated mass m_{WWbb} from reconstructed events for the background process and signal process with several selection of m_X, the hypothetical X particle mass. It's clear that the signal distribution is distinct from background given this mass as a classification feature.

In this situation, a “traditional” approach would involve one of three methods

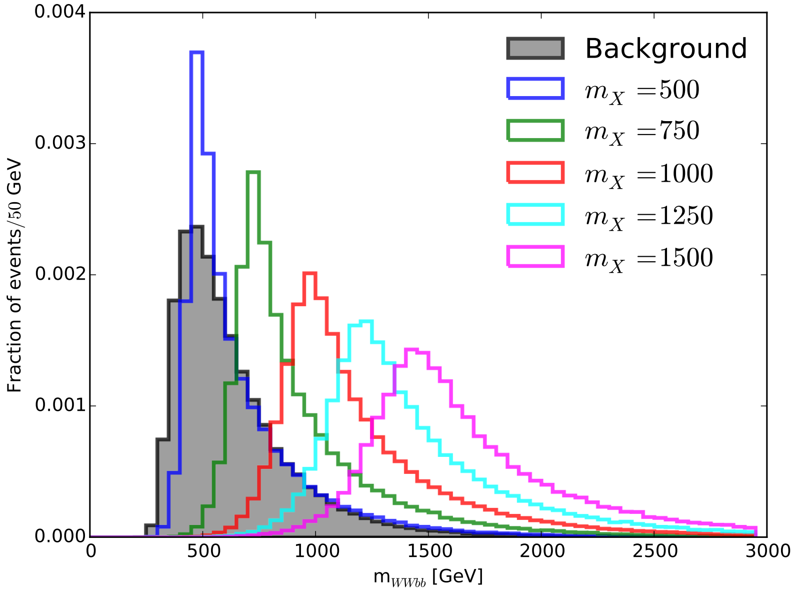


Figure 7.4: Mass distributions for the calculated resonance M_{WWbb} for signal events given various masses of particle X and a background distribution from the standard decay process.

1. Train one model at a midpoint in the mass range M_{WWbb} in an effort to find a generalized solution that attempts to capture an “average” solution to the problem [83, 185]. Performance at or near the training data region will be ideal but will degrade for other choices of mass

2. Train a single model with a mixed group of signal samples for all choices of mass. This approach learns a more generalized problem and performs better for examples outside of the training data. However, the performance of events from within the region of the training data suffers compared to a more focused network.

3. Train a separate model for each selection of mass, as done in Refs. [17, 83]. This method will give good results for all explicit mass selections (i.e. those present in the training data). However, performance outside of the known mass regions will suffer and results in discontinuities in selection efficiencies across the mass region and an inability to interpolate between trained mass regions.

For comparison, one can utilize a parameterized network which introduces the true mass of the hypothetical particle M_X as an additional input parameter. More generally, for any network trained with n input features and m parameters, it's possible to train a network of size $n+m$ incorporating event-level features and related parameters to build a parameterized model.

It's worth noting that a similar method with similar goals was used in Ref. [186]. However, in this study the results in application to a Boosted Decision Tree (BDT) gave worse sensitivity at the trained "true" parameter regions when compared to individually trained algorithms at those parameter values. Additionally, this work did not show an ability to interpolate between regions given a high-dimensional input feature space.

In the application of a parameterized neural network to the top decay problem, a multi-layer perceptron is used from PyLearn2 [187], with outputs treated with a regressor method and logistic activation function. The input and output data are normalized to a range of zero and one using scikit-learn `minmaxscaler` [188]. All neural networks are trained with 1 hidden layer, each with 3 nodes and using Nesterov's method for stochastic gradient descent [189]. The learning rate is set to 0.01, momentum set to 0.9 and minibatch size is set to treat each point individually (i.e. minibatch size of 1). For each network and choice of mass, training is done on 100k samples.

For each model, the signal-background classification performance is measured according to the predictions ROC curve (signal and background efficiency) and this is compared to the same metric for interpolated regions with the parameterized network. Performance for the parameterized network is also compared to a network trained at a fixed mass value, m_X^0 for mass values away from selected mass. For example, Fig. 7.5 shows the ROC curve for a single fixed network trained at true mass value $m_X^0 = 750\,\text{GeV}$ (black points) and this is compared to a parameterized network trained at surrounding mass values but excluding fixed networks parameter (i.e. parameterized network $m_X = 500$, 1000, 1250, 1500 GeV (black line).

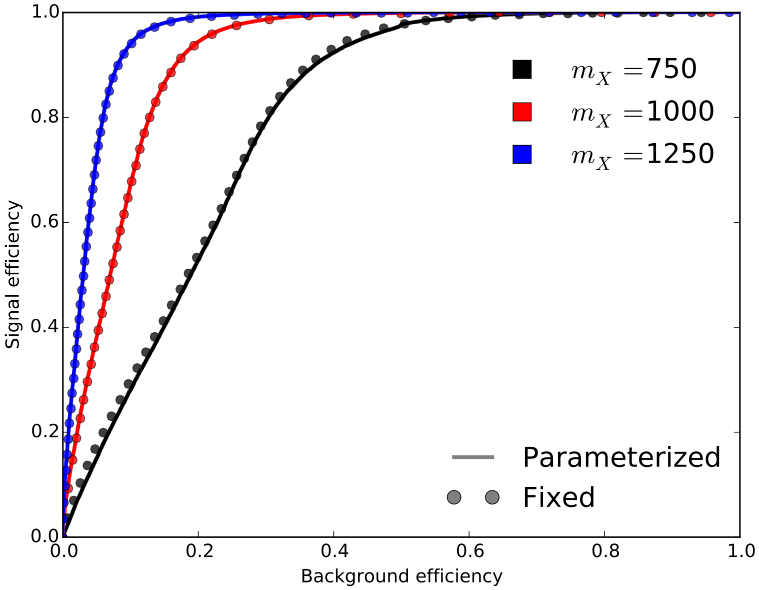


Figure 7.5: ROC curves for fixed networks (points) at various true mass values compared to predictions of a parameterized network (solid line) when trained at all true mass values *except* those of the fixed networks parameter.

For each mass, a *fixed* network is trained using exclusively data from that true mass region m_X^0. The performance is measured by its ROC curve and compared to a parameterized network making predictions at that same true mass value. However, for each mass region the parameterized network is not allowed to train on the input features of for that mass point but, instead, trains on inputs from nearby mass points. The parameterized network is then asked to interpolate predictions at m_X^0 and the ROC curve is measured on those predictions. For example, Fig. 7.5 compares a fixed network at $m_X^0 = 750\,\mathrm{GeV}$ to a parameterized network trained at $m_X = 500$, 1000, 1250, 1500 GeV but excluding $m_X = 750\,\mathrm{GeV}$. The parameterized network's mass parameter is then set to the true value of interest, $m_X^0 = 750\,\mathrm{GeV}$ and predictions are made using event level data for that mass. Despite having never been trained on data for this mass parameter, the signal/background efficiency closely mirrors the same performance seen for the fixed network.

For this example problem, a single parameterized network can be shown as capable of reproducing the performance of a fixed network even when being tasked with interpolating those results purely from neighboring/related data regions. This conclusion can't, however, be generalized far beyond the scope of the given training examples. When applied to other examples or more exotic mass regions outside the range of the training data, it's recommended to use a statistical hold-out test to check the relative performance of training/test/validation data. The application of such a statistical test for intermediate values of m_X entails knowing the distribution for the parameterized network output at that given parameter point (i.e. $p\left(f\left(x, m_X\right) \mid m_X\right)$. A discussion on *parameterized calibration* is given in detail by Ref. [190]. Note that a complete statistical comparison of both the fixed network and parameterized network would require having access to data at the mass region of interest. Generating data samples for parameters of interest is the most straight forward, albeit computationally expensive, strategy for establishing this comparison. An approximate, but more computationally efficient approach involves the use of an interpolation algorithm to construct the parameterized distribution [191–193]. This approach co-opts a common approach to dealing with uncertainties with respect to nuisance parameters.

7.5 High-Dimensional Physical Example

The previous sections demonstrate a functioning parameterized network but in the case of relatively simple one-dimensional examples with a clearly defined relationship between the event-level data and the parameter a network needs to learn to interpolate between. This same method, however, can also be applied to higher dimensional examples with similarly powerful interpolation results and high accuracy when compared to dedicated fixed networks.

Starting with the same signal and background process given in Fig. 7.3, the set of trainable features is expanded to include LL kinematics which correspond to the result of reconstruction algorithms, and HL features, which benefit from the domain of external physics knowledge. The LL features consist of 22 inputs comprised of:

- lepton leading momenta,
- momenta of the four leading jets,
- b-tagging jet information,
- missing transverse momentum magnitude
- missing transverse momentum angle
- jet multiplicity

HL features are then calculated from the LL information to generate a set of 5 invariant mass values for intermediate objects, which include:

- mass $m_{\ell\nu}$ for the process $W \to \ell\nu$
- mass m_{jj} for the process $W \to qq'$
- mass m_{jjj} for the process $t \to Wb \to bqq'$
- mass $m_{j\ell\nu}$ for the process $t \to Wb \to \ell\nu b$
- mass m_{WWbb} for the process $X \to t\bar{t}$

For both the HL and LL datasets, several distributions are given in Fig. 7.6

A parameterized neural network was trained using the *Blocks framework* [194–196] with seven million events for training and one million for validation. The dataset was a 50/50 admixture of signal and background. The network used five hidden layers, each with 500 hidden rectified

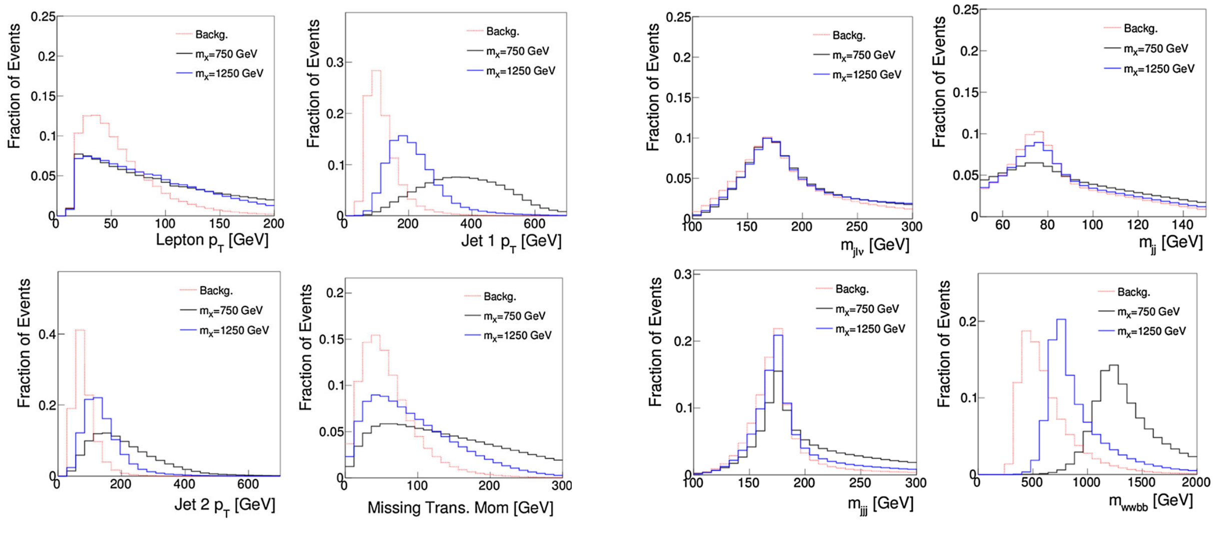


Figure 7.6: LL (left) and HL (right) feature distributions for the hypothetical particle decay $X \to t\bar{t}$ for particle masses $m_X = 750\,\text{GeV}$ and $m_x = 1250\,\text{GeV}$ and the dominant background process

linear units and a logistic output. Parameters were initialized with a Gaussian distribution with a mean of zero and width of 0.1, updated using stochastic gradient descent and trained in mini-batches of size 100 with momentum 0.5. The learning rate was initialized at 0.1 with a decay factor of 0.89 per epoch and training was finalized after 200 epochs.

Because of the high dimensionality of the problem, the networks dependence on the true mass parameter, m_X, is difficult to visualize. As such, the Area Under the Curve (AUC) was measured for networks trained at different mass signals to compare relative performance between networks. A parameterized network was trained with masses m_X = 500, 750, 1000, 1250, 1500 GeV with 7M training samples followed by a test prediction at $m_X = 1000\,\text{GeV}$. The parameterized networks performance is then compared, in all cases with a train set of 7M samples, to a fixed network trained at $m_X = 1000\,\text{GeV}$ and a *broad* non-parameterized network trained with all mass points (m_X = 500, 750, 1000, 1250, 1500 GeV) in its input features. The ROC curves for networks using LL inputs is given in Fig. 7.7, along with an additional parameterized network which receives the additional mass point of interest,

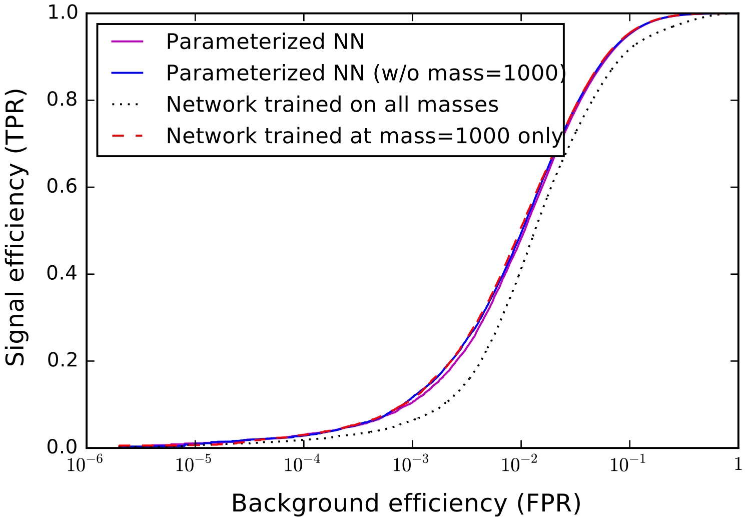


Figure 7.7: Comparison of the signal-to-background discrimination for a parameterized network, fixed network and a network trained across multiple mass points. The parameterized network is trained with masses of $m_X = 500,\ 750,\ 1000,\ 1250,\ 1500\,\text{GeV}$. This is presented in contrast to a fixed network trained at $m_X = 1000\,\text{GeV}$ and a non-parameterized network trained with the same mass points, $m_X = 500,\ 750,\ 1000,\ 1250,\ 1500\,\text{GeV}$. Results shown in this figure are for the case of LL features only but additional tests (not shown here) yield identical performance when incorporating both LL and HL features into all three networks.

$m_X = 1000\,\text{GeV}$, as an input. This comparison shows that both parameterized networks match the performance of the fixed network for the prediction at $m_X = 1000\,\text{GeV}$. Further, the network which excludes this mass point from the input features can easily interpolate between mass points for this prediction and suffers no degradation in predictions when compared to those which train for it explicitly. The more broadly trained network, however, loses performance relative to the fixed network and parameterized networks.

Conversely, Fig. 7.8 shows performance between the parameterized network, fixed network and broad network for predictions at all mass points present in the dataset. In contrast with the fixed network, the parameterized network demonstrates a superior ability to make predictions for mass points away from $m_X = 1000\,\text{GeV}$ as it makes interpolations away from the fixed mass training region. Furthermore, these interpolations outside of the central mass

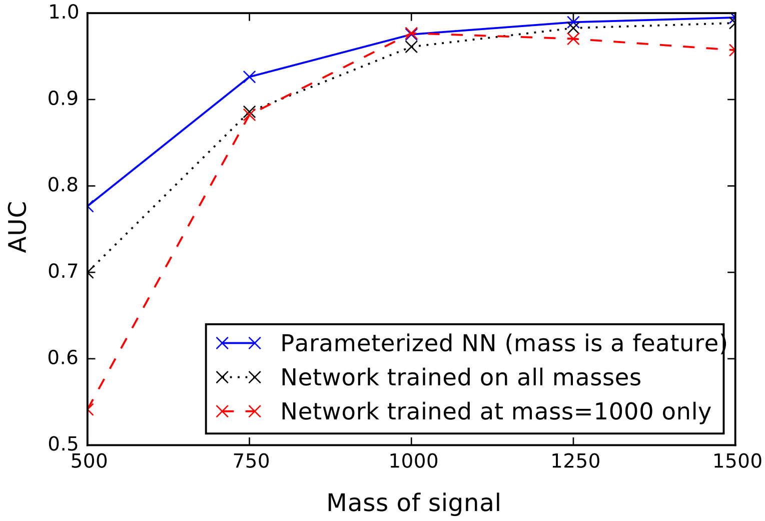


Figure 7.8: Comparison of network prediction performance for a fixed network, parameterized network and network trained on all masses at various mass points. The AUC at a mass signal of $m_X = 1000\,\text{GeV}$ is similar for the fixed network and parameterized network. Outside of this mass region, the performance of the parameterized network exceeds the other as it smoothly interpolates outside of the original test region.

region yield better results than the broad network and the falloff in performance as one makes predictions outside of the center mass region are less severe.

These results show improved results when compared to either of the traditional approaches to training supervised neural networks for either HL or LL features. Additionally, the parameterized network offers a simpler and more convenient approach to a traditional method like in the case of training multiple separate fixed networks.

Finally, it's notable that from past works, deep networks [26, 36] have been shown to be capable of achieving the same performance with HL features as with the initial kinematic information (i.e. LL four-vectors). As is mentioned in Fig. 7.7, the comparison in performance between the parameterized network, fixed network and broad network is identical in both the HL and LL cases.

7.6 Discussion

The parameterized network outlined in this chapter represents a novel approach to training neural networks for common applications in high-energy physics which gives improved performance and a simplified design compared to traditional approaches to the same problem. Although the example given in Sec. 7.5 uses a single parameter, θ, the method is easily generalized to a higher set of input parameter spaces.

Parameterized networks are also capable of improving performance as a function of nuisance parameters that describe systematic uncertainties when, in contrast, traditional networks are optimized for a single parameter value. The parameterized network allows for statistical tests using profile likelihood ratio tests [197] to choose networks corresponding to the profiled values of the nuisance parameters [190].

Chapter 8

Conclusion

The work contained in the previous chapters have examined smarter approaches and more targeted studies of both high-level and low-level physics data as modeled by opaque learning strategies. These techniques offer the potential for more instructive and powerful machine learning at the LHC with the opportunity to uncover entirely new physics from the datasets collected in particle detectors. These novel strategies can contribute insights at various stages of analysis, from the early model design and feature selection to later analysis and interpretation.

However, the methods detailed here are far from comprehensive and tools addressing interpretability in machine learning, and HEP in particular, are largely in their infancy. The complications brought about by a dearth of black box solving methods promises to only become greater as physicists expand their search to more exotic phenomena in ever larger and more intricate data. In that light, this work should be seen as an illustration of the critical importance of interpretability in machine learning for high-energy applications, a motivation for future techniques for achieving this goal and a general reflection of “best practices” for viewing opaque learning in the field.

Physicists should be more wary, in general, at the prospect of purely black box learning that can obscure how a problem is solved in exchange for performance improvements. In general, one should evaluate whether that "extra" information captured by a deep learning technique is representative of real physics that can be accounted for by the user deploying the model.

The ultimate goal for research in high-energy physics should not to be the development of artificial intelligence physicists which (or should we say *who*?) can blindly process raw data and make statements about the structure of the Universe without being able to communicate the intermediate steps. Instead, machine learning should be used to identify the gaps in human knowledge and, where possible, guide the search for better human-engineered approaches.

Bibliography

[1] Taylor Faucett, Jesse Thaler, and Daniel Whiteson. "Mapping machine-learned physics into a human-readable space". In: *Physical Review D* 103 (3 Feb. 2021).

[2] Julian Collado et al. "Learning to identify electrons". In: *Physical Review D* 103 (11 June 2021).

[3] Julian Collado et al. "Learning to isolate muons". In: *Journal of High Energy Physics 2021 2021:10* 2021 (10 Oct. 2021).

[4] Pierre Baldi et al. "Parameterized neural networks for high-energy physics". In: *The European Physical Journal C* 76 (5 May 2016).

[5] J K Kohne et al. "Realization of a second level neural network trigger for the H1 experiment at HERA". In: *Nucl. Instrum. Meth.* A389 (1997).

[6] Sean Benson et al. "The LHCb Turbo Stream". In: 664.8 (Dec. 2015).

[7] Johannes Albrecht et al. *HEP Community White Paper on Software trigger and event reconstruction.* 2018.

[8] V V Gligorov and M Williams. "Efficient, reliable and fast high-level triggering using a bonsai boosted decision tree". In: 8.02 (Feb. 2013).

[9] Georges Aad et al. "A neural network clustering algorithm for the ATLAS silicon pixel detector". In: *JINST* 9 (2014).

[10] Carsten Peterson. “Track Finding With Neural Networks”. In: *Nucl. Instrum. Meth.* A279 (1989).

[11] Halina Abramowicz, Allen Caldwell, and Ralph Sinkus. “Neural network based electron identification in the ZEUS calorimeter”. In: *Nucl. Instrum. Meth.* A365 (1995).

[12] Vardan Khachatryan et al. “Observation of the Diphoton Decay of the Higgs Boson and Measurement of Its Properties”. In: *Eur. Phys. J. C* 74 (2014).

[13] V M Abazov et al. “First measurement of sigma ($p\bar{p} \to Z$) . Br($Z \to \tau\tau$) at $\sqrt{s} =$ 1.96 TeV”. In: *Phys. Rev.* D71 (2005).

[14] Denis Derkach, Mikhail Hushchyn, and Nikita Kazeev. “Machine Learning based Global Particle Identification Algorithms at the LHCb Experiment”. In: *EPJ Web of Conferences* 214 (Sept. 2019).

[15] P. Abreu et al. “Classification of the hadronic decays of the Z^0 into b and c quark pairs using a neural network”. In: *Physics Letters B* 295 (Dec. 1992).

[16] V.M. Abazov et al. “Search for single top quark production at D0 using neural networks”. In: *Physics Letters B* 517.3 (2001).

[17] T. Aaltonen et al. “Evidence for a Particle Produced in Association with Weak Bosons and Decaying to a Bottom-Antibottom Quark Pair in Higgs Boson Searches at the Tevatron”. In: *Physical Review Letters* 109 (Aug. 2012).

[18] Michela Paganini, Luke de Oliveira, and Benjamin Nachman. “Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters”. In: *Physical Review Letters* 120.4 (Jan. 2018).

[19] Michela Paganini, Luke de Oliveira, and Benjamin Nachman. “CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks”. In: *Physical Review D* 97 (Dec. 2017).

[20] Philip Ilten, Mike Williams, and Yunjie Yang. “Event generator tuning using Bayesian optimization”. In: *Journal of Instrumentation* 12 (Oct. 2016).

[21] Andreas Hoecker et al. “TMVA: Toolkit for Multivariate Data Analysis”. In: *PoS* ACAT (2007).

[22] R. Brun and F. Rademakers. “ROOT: An object oriented data analysis framework”. In: *Nucl. Instrum. Meth. A* 389 (1997).

[23] Matteo Cacciari, Gavin P Salam, and Gregory Soyez. “The anti-ktjet clustering algorithm”. In: *Journal of High Energy Physics* 2008 (Apr. 2008).

[24] Andrew J Larkoski, Gavin P Salam, and Jesse Thaler. “Energy Correlation Functions for Jet Substructure”. In: *arXiv* (Apr. 2013).

[25] Jesse Thaler and Ken Van Tilburg. “Identifying boosted objects with N-subjettiness”. In: *Journal of High Energy Physics 2011 2011:3* 2011 (Mar. 2011).

[26] Pierre Baldi, Peter Sadowski, and Daniel Whiteson. “Searching for Exotic Particles in High-Energy Physics with Deep Learning”. In: *Nature Commun.* 5 (2014).

[27] Luke de Oliveira et al. “Jet-images — deep learning edition”. In: *JHEP* 07 (2016).

[28] Pierre Baldi et al. “Jet Substructure Classification in High-Energy Physics with Deep Neural Networks”. In: *Phys. Rev.* D93.9 (2016).

[29] Spencer Chang, Timothy Cohen, and Bryan Ostdiek. “What is the Machine Learning?” In: *Phys. Rev. D* 97.5 (2018).

[30] Rich Caruana et al. “Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-Day Readmission”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’15. Sydney, NSW, Australia: Association for Computing Machinery, 2015. ISBN: 9781450336642.

[31] Yash Patel, Tomas Hodan, and Jiri Matas. “Learning Surrogates via Deep Embedding”. In: (July 2020). eprint: 2007.00799.

[32] Ivo Couckuyt et al. “Automatic surrogate model type selection during the optimization of expensive black-box problems”. In: *Proceedings of the 2011 Winter Simulation Conference (WSC)*. 2011.

[33] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “”Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. In: (Feb. 2016). eprint: `1602.04938`.

[34] Muhammad Rehman Zafar and Naimul Mefraz Khan. *DLIME: A Deterministic Local Interpretable Model-Agnostic Explanations Approach for Computer-Aided Diagnosis Systems*. 2019. arXiv: `1906.10263 [cs.LG]`.

[35] Scott M Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017.

[36] P. Baldi, P. Sadowski, and D. Whiteson. “Enhanced Higgs Boson to $\tau^+\tau^-$ Searches with Deep Learning”. In: *Physical Review Letters* 114 (Mar. 2015).

[37] Roberto Santos et al. “Machine learning techniques in searches for $t\bar{t}h$ in the $h \to b\bar{b}$ decay channel”. In: *JINST* 12.04 (2017).

[38] A. Aurisano et al. “A Convolutional Neural Network Neutrino Event Classifier”. In: *JINST* 11.09 (2016).

[39] Timothy Cohen, Marat Freytsis, and Bryan Ostdiek. “(Machine) Learning to Do More with Less”. In: *JHEP* 02 (2018).

[40] M. Andrews et al. “End-to-End Event Classification of High-Energy Physics Data”. In: *J. Phys. Conf. Ser.* 1085.4 (2018).

[41] Leandro G Almeida et al. “Playing Tag with ANN: Boosted Top Identification with Pattern Recognition”. In: *JHEP* 07 (2015).

[42] Andrew J. Larkoski, Ian Moult, and Benjamin Nachman. “Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning”. In: *Phys. Rept.* 841 (2020).

[43] Gregor Kasieczka et al. “Deep-learning Top Taggers or The End of QCD?” In: *JHEP* 05 (2017).

[44] Daniel Guest et al. “Jet Flavor Classification in High-Energy Physics with Deep Neural Networks”. In: *Phys. Rev.* D94 (2016).

[45] *Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment.* Tech. rep. Geneva: CERN, Mar. 2017.

[46] C M S Collaboration. *Heavy flavor identification at CMS with deep neural networks.* Mar. 2017.

[47] A M Sirunyan et al. “Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV”. In: *JINST* 13 (2018).

[48] Patrick T Komiske, Eric M Metodiev, and Matthew D Schwartz. “Deep learning in color: towards automated quark/gluon jet discrimination”. In: *JHEP* 01 (2017).

[49] Anders Andreassen et al. “OmniFold: A Method to Simultaneously Unfold All Observables”. In: *Phys. Rev. Lett.* 124 (2020).

[50] Kaustuv Datta, Deepak Kar, and Debarati Roy. “Unfolding with Generative Adversarial Networks”. In: (Oct. 2018).

[51] Marco Bellagente et al. “Invertible Networks or Partons to Detector and Back Again”. In: (Oct. 2020).

[52] Christoph Englert et al. “Machine Learning Uncertainties with Adversarial Neural Networks”. In: *Eur. Phys. J. C* 79 (2019).

[53] James Barnard et al. “Parton Shower Uncertainties in Jet Substructure Analyses with Deep Neural Networks”. In: *Phys. Rev. D* 95 (2017).

[54] Sven Bollweg et al. "Deep-Learning Jets with Uncertainties and More". In: *SciPost Phys.* 8 (2020).

[55] Benjamin Nachman. "A guide for deploying Deep Learning in LHC searches: How to achieve optimality and account for uncertainty". In: *SciPost Phys.* 8 (2020).

[56] Gregor Kasieczka et al. "Per-Object Systematics using Deep-Learned Calibration". In: (Oct. 2020).

[57] Eric M. Metodiev, Benjamin Nachman, and Jesse Thaler. "Classification without labels: Learning from mixed samples in high energy physics". In: *JHEP* 10 (2017).

[58] Anders Andreassen et al. "JUNIPR: a Framework for Unsupervised Machine Learning in Particle Physics". In: *Eur. Phys. J. C* 79 (2019).

[59] Jack H Collins, Kiel Howe, and Benjamin Nachman. "Anomaly Detection for Resonant New Physics with Machine Learning". In: *Phys. Rev. Lett.* 121 (2018).

[60] Cristian Buciluundefined, Rich Caruana, and Alexandru Niculescu-Mizil. "Model Compression". In: Association for Computing Machinery, 2006.

[61] Javier Duarte et al. "Fast inference of deep neural networks in FPGAs for particle physics". In: *JINST* 13 (2018).

[62] A. A. Alemi et al. "Deep Variational Information Bottleneck". In: *arXiv* (Dec. 2016).

[63] Stefan Wunsch et al. "Identifying the relevant dependencies of the neural network response on characteristics of the input space". In: *Comput. Softw. Big Sci.* 2.1 (2018).

[64] Thomas Roxlo and Matthew Reece. "Opening the black box of neural nets: case studies in stop/top discrimination". In: (Apr. 2018).

[65] Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler. "Energy Flow Networks: Deep Sets for Particle Jets". In: *JHEP* 01 (2019).

[66] M H Seymour. "Tagging a heavy Higgs boson". In: Nov. 1991.

[67] Michael H. Seymour. “Searches for new particles using cone and cluster jet algorithms: a comparative study”. In: *Zeitschrift für Physik C Particles and Fields 1994 62:1* 62 (Mar. 1994).

[68] J. M. Butterworth, B. E. Cox, and J. R. Forshaw. “WW scattering at the CERN LHC”. In: *Physical Review D* 65 (May 2002).

[69] Jonathan M. Butterworth, John R. Ellis, and Are R. Raklev. “Reconstructing sparticle mass spectra using hadronic decays”. In: *Journal of High Energy Physics* 2007 (May 2007).

[70] Jonathan M. Butterworth et al. “Jet Substructure as a New Higgs-Search Channel at the Large Hadron Collider”. In: *Physical Review Letters* 100 (June 2008).

[71] A. Abdesselam et al. “Boosted objects: a probe of beyond the standard model physics”. In: *The European Physical Journal C 2011 71:6* 71 (June 2011).

[72] A Altheimer et al. “Jet substructure at the Tevatron and LHC: new results, new tools, new benchmarks*”. In: *Journal of Physics G: Nuclear and Particle Physics* 39 (May 2012).

[73] Jessie Shelton. “TASI Lectures on Jet Substructure”. In: (Feb. 2013).

[74] D. Adams et al. “Towards an understanding of the correlations in jet substructure”. In: *The European Physical Journal C 2015 75:9* 75 (Sept. 2015).

[75] Roman Kogler et al. “Jet substructure at the Large Hadron Collider”. In: *Rev. Mod. Phys.* 91 (Dec. 2019).

[76] Simone Marzani, Gregory Soyez, and Michael Spannowsky. “Looking Inside Jets”. In: 958 (2019).

[77] Yanou Cui, Zhenyu Han, and Matthew D. Schwartz. “W-jet tagging: Optimizing the identification of boosted hadronically-decaying W bosons”. In: *Physical Review D* 83 (Apr. 2011).

[78] Jesse Thaler and Ken Van Tilburg. “Maximizing boosted top identification by minimizing N-subjettiness”. In: *Journal of High Energy Physics 2012 2012:2* 2012 (Feb. 2012).

[79] G Aad et al. “Measurement of the cross-section of high transverse momentum vector bosons reconstructed as single jets and studies of jet substructure in pp collisions at = 7 TeV with the ATLAS detector”. In: *New Journal of Physics* 16 (Nov. 2014).

[80] Andrew J. Larkoski, Ian Moult, and Duff Neill. “Power counting to better jet observables”. In: *Journal of High Energy Physics 2014 2014:12* 2014 (Dec. 2014).

[81] Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler. “Energy flow polynomials: A complete linear basis for jet substructure”. In: *JHEP* 04 (2018).

[82] Francesco Pandolfi. “Search for the Standard Model Higgs Boson in the $H \to ZZ \to l^+l^-q\bar{q}$ Decay Channel at CMS”. PhD thesis. New York: Zurich, ETH, 2012.

[83] Serguei Chatrchyan et al. “Search for Z’ Resonances Decaying to t tbar in Dilepton + Jets Final States in pp Collisions at $\sqrt{s}$ =7 TeV”. In: *Phys.Rev.D* 87 (Apr. 2013).

[84] Andrew J Larkoski, Jesse Thaler, and Wouter J Waalewijn. “Gaining (Mutual) Information about Quark/Gluon Discrimination”. In: *JHEP* 11 (2014).

[85] Philippe Gras et al. “Systematics of quark/gluon tagging”. In: *Journal of High Energy Physics 2017 2017:7* 2017 (July 2017).

[86] M G KENDALL. “A New Measure of Rank Correlation”. In: *Biometrika* 30 (1938).

[87] Robert E. Schapire. “The Strength of Weak Learnability”. In: *Machine Learning* 5.2 (1990).

[88] Y. Freund. “Boosting a Weak Learning Algorithm by Majority”. In: *Information and Computation* 121.2 (1995).

[89] A. Altheimer et al. “Boosted objects and jet substructure at the LHC. Report of BOOST2012”. In: *The European Physical Journal C 2014 74:3* 74 (Mar. 2014).

[90] Huilin Qu and Loukas Gouskos. “Jet tagging via particle clouds”. In: *Phys. Rev. D* 101 (Mar. 2020).

[91] Eric A Moreno et al. “JEDI-net: a jet identification algorithm based on interaction networks”. In: *Eur. Phys. J. C* 80 (2020).

[92] Vinicius Mikuni and Florencia Canelli. “ABCNet: An attention-based method for particle tagging”. In: *Eur. Phys. J. Plus* 135 (2020).

[93] Alexander Bogatskiy et al. *Lorentz Group Equivariant Neural Network for Particle Physics*. 2020.

[94] Jonathan Shlomi et al. “Secondary Vertex Finding in Jets with Neural Networks”. In: (Oct. 2020).

[95] Josh Cogan et al. “Jet-Images: Computer Vision Inspired Techniques for Jet Tagging”. In: *JHEP* 02 (2015).

[96] David E. Kaplan et al. “Top Tagging: A Method for Identifying Boosted Hadronically Decaying Top Quarks”. In: *Phys. Rev. Lett.* 101 (Oct. 2008).

[97] Leandro G. Almeida et al. “Substructure of high-p_T jets at the LHC”. In: *Phys. Rev. D* 79 (Apr. 2009).

[98] Stephen D. Ellis, Christopher K. Vermilion, and Jonathan R. Walsh. “Recombination algorithms and jet substructure: Pruning as a tool for heavy particle searches”. In: *Phys. Rev. D* 81 (May 2010).

[99] Mrinal Dasgupta et al. “Towards an understanding of jet substructure”. In: *Journal of High Energy Physics* 2013.9 (2013).

[100] Andrew J Larkoski et al. “Soft drop”. In: *Journal of High Energy Physics* 2014 (May 2014).

[101] Andy Buckley et al. “An Optimal Observable for Color Singlet Identification”. In: *SciPost Phys.* 9 (2020).

[102] Anja Butter et al. “The Machine Learning Landscape of Top Taggers”. In: *SciPost Phys.* 7 (2019).

[103] Kaustuv Datta and Andrew Larkoski. “How Much Information is in a Jet?” In: *JHEP* 06 (2017).

[104] Liam Moore et al. “Reports of My Demise Are Greatly Exaggerated: N-subjettiness Taggers Take On Jet Images”. In: *SciPost Phys.* 7 (2019).

[105] J. A. Aguilar-Saavedra and B. Zaldıévar. “Jet tagging made easy”. In: *The European Physical Journal C* 80.6 (2020).

[106] Johan Alwall et al. *MadGraph 5 : Going Beyond.* 2011.

[107] Torbjorn Sjostrand, Stephen Mrenna, and Peter Z. Skands. “PYTHIA 6.4 Physics and Manual”. In: *JHEP* 0605 (2006).

[108] J. de Favereau et al. “DELPHES 3, A modular framework for fast simulation of a generic collider experiment”. In: *JHEP* 02 (2014).

[109] David Krohn, Jesse Thaler, and Lian-Tao Wang. “Jet trimming”. In: *Journal of High Energy Physics* 2010 (Feb. 2010).

[110] Atlas Collaboration. “Identification of boosted, hadronically decaying W bosons and comparisons with ATLAS data taken at $\sqrt{s} = 8$ TeV”. In: *The European Physical Journal C* 76.3 (2016).

[111] Morad Aaboud et al. “Performance of top-quark and W-boson tagging with ATLAS in Run 2 of the LHC”. In: *Eur. Phys. J. C* 79 (2019).

[112] CMS collaboration. “Identification techniques for highly boosted W bosons that decay into hadrons”. In: *Journal of High Energy Physics* 2014.12 (2014).

[113] *Quark versus Gluon Jet Tagging Using Jet Images with the ATLAS Detector*. Tech. rep. Geneva: CERN, July 2017.

[114] Sebastian Macaluso and David Shih. “Pulling out all the tops with computer vision and deep learning”. In: *Journal of High Energy Physics* 2018.10 (2018).

[115] Andrea Banfi, Gavin P Salam, and Giulia Zanderighi. “Principles of general final-state resummation and automated implementation”. In: 2005.03 (Mar. 2005).

[116] Guy Gur-Ari, Michele Papucci, and Gilad Perez. *Classification of Energy Flow Observables in Narrow Jets.* 2011.

[117] Martin Jankowiak and Andrew J. Larkoski. “Jet substructure without trees”. In: *Journal of High Energy Physics* 2011.6 (2011).

[118] Ian Moult, Lina Necib, and Jesse Thaler. “New angles on energy correlation functions”. In: *Journal of High Energy Physics* 2016.12 (2016).

[119] Serguei Chatrchyan et al. “Search for a Higgs boson in the decay channel H to ZZ^* to $q\bar{q}$ $\ell^-\ell^+$ in pp collisions at $\sqrt{s} = 7$ TeV”. In: *JHEP* 04 (2012).

[120] Patrick T Komiske. *Energy Flow.* (Visited on 05/13/2021).

[121] Suyong Choi, Seung J Lee, and Maxim Perelstein. “Infrared Safety of a Neural-Net Top Tagging Algorithm”. In: *JHEP* 02 (2019).

[122] Gregor Kasieczka et al. “Quark-Gluon Tagging: Machine Learning vs Detector”. In: *SciPost Phys.* 6 (2019).

[123] Kaustuv Datta and Andrew J Larkoski. “Novel Jet Observables from Machine Learning”. In: *JHEP* 03 (2018).

[124] Kaustuv Datta, Andrew Larkoski, and Benjamin Nachman. “Automating the construction of jet observables with machine learning”. In: *Phys. Rev. D* 100 (Nov. 2019).

[125] A.M. Sirunyan et al. “Identification of heavy, energetic, hadronically decaying particles using machine-learning techniques”. In: 15.06 (June 2020).

[126] Davison E Soper and Michael Spannowsky. “Finding physics signals with event deconstruction”. In: *Phys. Rev. D* 89 (2014).

[127] Davison E Soper and Michael Spannowsky. “Finding physics signals with shower deconstruction”. In: *Phys. Rev. D* 84 (2011).

[128] Davison E Soper and Michael Spannowsky. “Finding top quarks with shower deconstruction”. In: *Phys. Rev. D* 87 (2013).

[129] Danilo Ferreira de Lima et al. “Quark-gluon tagging with shower deconstruction: Unearthing dark matter and Higgs couplings”. In: *Phys. Rev. D* 95 (Feb. 2017).

[130] Andrew J Larkoski and Eric M Metodiev. “A Theory of Quark vs. Gluon Discrimination”. In: *JHEP* 10 (2019).

[131] Gregor Kasieczka et al. “Towards Machine Learning Analytics for Jet Substructure”. In: *JHEP* 09 (2020).

[132] Georges Aad et al. “Measurement of $W^{\pm}$ and Z-boson production cross sections in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector”. In: *Phys. Lett. B* 759 (2016).

[133] Eduardo Silva Almeida et al. “Electroweak Sector Under Scrutiny: A Combined Analysis of LHC and Electroweak Precision Data”. In: *Phys. Rev. D* 99.3 (2019).

[134] Georges Aad et al. “Search for supersymmetry in final states with jets, missing transverse momentum and one isolated lepton in $\sqrt{s} = 7$ TeV pp collisions using 1 fb^{-1} of ATLAS data”. In: *Phys. Rev. D* 85.1 (2012).

[135] Serguei Chatrchyan et al. “Search for Supersymmetry in pp Collisions at $\sqrt{s}$=8 TeV in Events with a Single Lepton, Large Jet Multiplicity, and Multiple b Jets”. In: *Phys. Lett. B* 733 (2014).

[136] Vardan Khachatryan et al. “Performance of Electron Reconstruction and Selection with the CMS Detector in Proton-Proton Collisions at $\sqrt{s} = 8$ TeV”. In: *JINST* 10.06 (2015).

[137] *Electron efficiency measurements with the ATLAS detector using the 2015 LHC proton-proton collision data.* Tech. rep. ATLAS-CONF-2016-024. Geneva: CERN, June 2016.

[138] M. Hushchyn and V. Chekalina. “Particle-identification techniques and performance at LHCb in Run 2”. In: *Nucl. Instrum. Meth. A* 936 (2019).

[139] S. Agostinelli et al. “GEANT4: A Simulation toolkit”. In: *Nucl. Instrum. Meth. A* 506 (2003).

[140] G. Aad et al. “The ATLAS Experiment at the CERN Large Hadron Collider”. In: *JINST* 3 (2008).

[141] Luke De Oliveira, Benjamin Nachman, and Michela Paganini. “Electromagnetic Showers Beyond Shower Shapes”. In: *Nucl. Instrum. Meth. A* 951 (2020).

[142] Peter Berta et al. “Particle-level pileup subtraction for jets and jet shapes”. In: *JHEP* 06 (2014).

[143] Vinod Nair and Geoffrey E Hinton. “Rectified Linear Units Improve Restricted Boltzmann Machines”. In: ed. by Johannes Furnkranz and Thorsten Joachims. Omnipress, 2010.

[144] Geoffrey E. Hinton et al. “Improving neural networks by preventing co-adaptation of feature detectors”. In: *CoRR* abs/1207.0580 (2012).

[145] P Baldi and P Sadowski. “The Dropout Learning Algorithm”. In: *Artificial Intelligence* 210C (2014).

[146] François Chollet et al. *Keras.* 2015.

[147] Martıén Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* 2015.

[148] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *CoRR* abs/1412.6980 (2014).

[149] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: ed. by Yee Whye Teh and D. Mike Titterington. Vol. 9. JMLR.org, 2010.

[150] Lars Hertel et al. “Sherpa: Robust Hyperparameter Optimization for Machine Learning”. In: *SoftwareX* (2020).

[151] Lucio Mwinmaarong Dery et al. “Weakly Supervised Classification in High Energy Physics”. In: *JHEP* 05 (2017).

[152] Morad Aaboud et al. “Search for electroweak production of supersymmetric states in scenarios with compressed mass spectra at $\sqrt{s} = 13$ TeV with the ATLAS detector”. In: *Phys. Rev.* D97.5 (2018).

[153] Robert Schoefbeck. “Search for supersymmetry with extremely compressed spectra with the ATLAS and CMS detectors”. In: *Nuclear and Particle Physics Proceedings* 273-275 (2016).

[154] Vardan Khachatryan et al. “Search for supersymmetry in the vector-boson fusion topology in proton-proton collisions at $\sqrt{s} = 8$ TeV”. In: *JHEP* 11 (2015).

[155] Issac Hoenig, Gabriel Samach, and David Tucker-Smith. “Searching for dilepton resonances below the Z mass at the LHC”. In: *Phys. Rev. D* 90 (2014).

[156] A. M. Sirunyan et al. “Particle-flow reconstruction and global event description with the CMS detector”. In: *JINST* 12.10 (2017).

[157] Joosep Pata et al. “MLPF: Efficient machine-learned particle-flow reconstruction using graph neural networks”. In: (Jan. 2021).

[158] Georges Aad et al. “Muon reconstruction performance of the ATLAS detector in proton–proton collision data at $\sqrt{s}$ =13 TeV”. In: *Eur. Phys. J.* C76.5 (2016).

[159] Roel Aaij et al. “Search for Dark Photons Produced in 13 TeV pp Collisions”. In: *Phys. Rev. Lett.* 120.6 (2018).

[160] Zachary Hall and Jesse Thaler. “Photon isolation and jet substructure”. In: *JHEP* 09 (2018).

[161] ATLAS Collaboration. *Deep Learning for Pion Identification and Energy Calibration with the ATLAS Detector*. Tech. rep. ATL-PHYS-PUB-2020-018. Geneva: CERN, July 2020.

[162] Christopher Brust et al. "Identifying boosted new physics with non-isolated leptons". In: *JHEP* 04 (2015).

[163] ATLAS Collaboration. "Search for associated production of a Z boson with an invisibly decaying Higgs boson or dark matter candidates at $\sqrt{s} = 13$ TeV with the ATLAS detector". In: (Nov. 2021). eprint: `2111.08372`.

[164] Janik von Ahnen. "Dark Sector searches with jets". In: (Nov. 2021). eprint: `2111.00270`.

[165] Vasiliki A Mitsou. "Overview of searches for dark matter at the LHC". In: *Journal of Physics: Conference Series* 651 (Nov. 2015).

[166] J. G. de Swart, G. Bertone, and J. van Dongen. "How dark matter came to matter". In: *Nature Astronomy* 1.3 (Mar. 2017).

[167] Stefano Giagu. "WIMP Dark Matter Searches With the ATLAS Detector at the LHC". In: *Frontiers in Physics* 7 (2019).

[168] Timothy Cohen, Mariangela Lisanti, and Hou Keong Lou. "Semi-visible Jets: Dark Matter Undercover at the LHC". In: (Feb. 2015).

[169] Timothy Cohen, Joel Doss, and Marat Freytsis. "Jet substructure from dark sector showers". In: *Journal of High Energy Physics* 2020.9 (Sept. 2020).

[170] Deepak Kar and Sukanya Sinha. "Exploring Jet Substructure in Semi-visible jets". In: *arXiv* (July 2020).

[171] Matthew J Strassler and Kathryn M Zurek. "Echoes of a hidden valley at hadron colliders". In: *Physics Letters B* 651 (Aug. 2007).

[172] Timothy Cohen et al. "LHC searches for dark sector showers". In: *Journal of High Energy Physics* 2017.11 (2017).

[173] P Skands, S Carrazza, and J Rojo. "Tuning PYTHIA 8.1: the Monash 2013 tune". In: *The European Physical Journal C* 74 (Aug. 2014).

[174] Michelangelo L Mangano, Mauro Moretti, and Roberto Pittau. "Multijet matrix elements and shower evolution in hadronic collisions: -jets as a case study". In: *Nuclear Physics B* 632.1-3 (June 2002).

[175] Matteo Cacciari, Gavin P Salam, and Gregory Soyez. "FastJet user manual". In: *The European Physical Journal C* 72 (Mar. 2012).

[176] Andrew Larkoski et al. "Exposing the QCD Splitting Function with CMS Open Data". In: *Physical Review Letters* 119 (Sept. 2017).

[177] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. New York, NY, USA: ACM, 2016.

[178] Guolin Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[179] Daniel Alvestad et al. "Beyond Cuts in Small Signal Scenarios - Enhanced Sneutrino Detectability Using Machine Learning". In: (Aug. 2021).

[180] Dimitri Bourilkov. "Machine and deep learning applications in particle physics". In: *International Journal of Modern Physics A* 34.35 (Dec. 2019).

[181] Alan S. Cornell et al. *Boosted decision trees in the era of new physics: a smuon analysis case study*. 2021.

[182] Scott M Lundberg et al. "Explainable machine-learning predictions for the prevention of hypoxaemia during surgery". In: *Nature Biomedical Engineering* 2.10 (2018).

[183] Serguei Chatrchyan et al. “Combined results of searches for the standard model Higgs boson in pp collisions at $\sqrt{s}$ =7 TeV”. In: *Phys.Lett.B* 710 (Mar. 2012).

[184] Georges Aad et al. “Search for a multi-Higgs-boson cascade in $W^+W^-b\bar{b}$ events with the ATLAS detector in pp collisions at $\sqrt{s} = 8$ TeV”. In: *Phys.Rev.D* 89 (Feb. 2014).

[185] G. Aad et al. “Search for $W' \to t\bar{b}$ in the lepton plus jets final state in proton–proton collisions at a centre-of-mass energy of $\sqrt{s} = 8$ TeV with the ATLAS detector”. In: *Physics Letters, Section B: Nuclear, Elementary Particle and High-Energy Physics* 743 (Apr. 2015).

[186] V. M. Abazov et al. “Search for the standard model Higgs boson in tau lepton final states”. In: *Physics Letters B* 714 (Aug. 2012).

[187] Ian J. Goodfellow et al. “Pylearn2: a machine learning research library”. In: (Aug. 2013).

[188] F Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011).

[189] Yu. Nesterov. “Gradient methods for minimizing composite functions”. In: *Mathematical Programming 2012 140:1* 140 (Dec. 2012).

[190] Kyle Cranmer, Juan Pavez, and Gilles Louppe. “Approximating Likelihood Ratios with Calibrated Discriminative Classifiers”. In: (2015).

[191] A. L. Read. “Linear interpolation of histograms”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 425 (Apr. 1999).

[192] Kyle Cranmer et al. *HistFactory: A tool for creating statistical models for use with RooFit and RooStats.* 2012.

[193] M. Baak et al. “Interpolation between multi-dimensional histograms using a new non-linear moment morphing method”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 771 (Jan. 2015).

[194] Bart van Merriënboer et al. “Blocks and Fuel: Frameworks for deep learning”. In: (June 2015).

[195] Frédéric Bastien et al. “Theano: new features and speed improvements”. In: (Nov. 2012).

[196] James Bergstra et al. “Theano: A CPU and GPU Math Compiler in Python”. In: *Proceedings of the 9th Python in Science Conference* (2010).

[197] Glen Cowan et al. “Asymptotic formulae for likelihood-based tests of new physics”. In: *The European Physical Journal C 2011 71:2* 71 (Feb. 2011).

[198] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. “Exact solutions to the nonlinear dynamics of learning in deep linear neural networks”. In: *CoRR* abs/1312.6120 (2013).

[199] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting.” In: *Journal of Machine Learning Research* 15 (2014).

[200] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. “Deep Sparse Rectifier Neural Networks”. In: vol. 15. PMLR, Oct. 2011.

[201] G Aad et al. “Measurement of soft-drop jet observables in pp collisions with the ATLAS detector at $\sqrt{s}$ =13 TeV”. In: *Physical Review D* 101 (Mar. 2020).

Appendix A

Mapping Machine-Learned Physics

A.1 Jet Substructure Observables and Performance

High-level jet features and the ROC curves establishing benchmark performance for the jet classification task are reproduced from here from Ref. [28] for context in the black-box guided application of the same data set

A.2 Network Architectures and Hyperparameters

For consistency, all neural networks use a set of common settings regardless of model design or input. $N = 5 \times 10^6$ event samples are used with 70% for training, 15% for validation and 15% for testing. Training samples are pre-processed via Sci-kit's `StandardScaler`[188]. The Adam optimizer is used with learning rate $= 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e - 07$, and amsgrad turned off. An sigmoid activation is used in the final layer to make predictions of signal or background and all measurement uncertainties are measured by a 10-fold cross validation as described in Sec. A.5. Models are trained with early stopping on validation loss

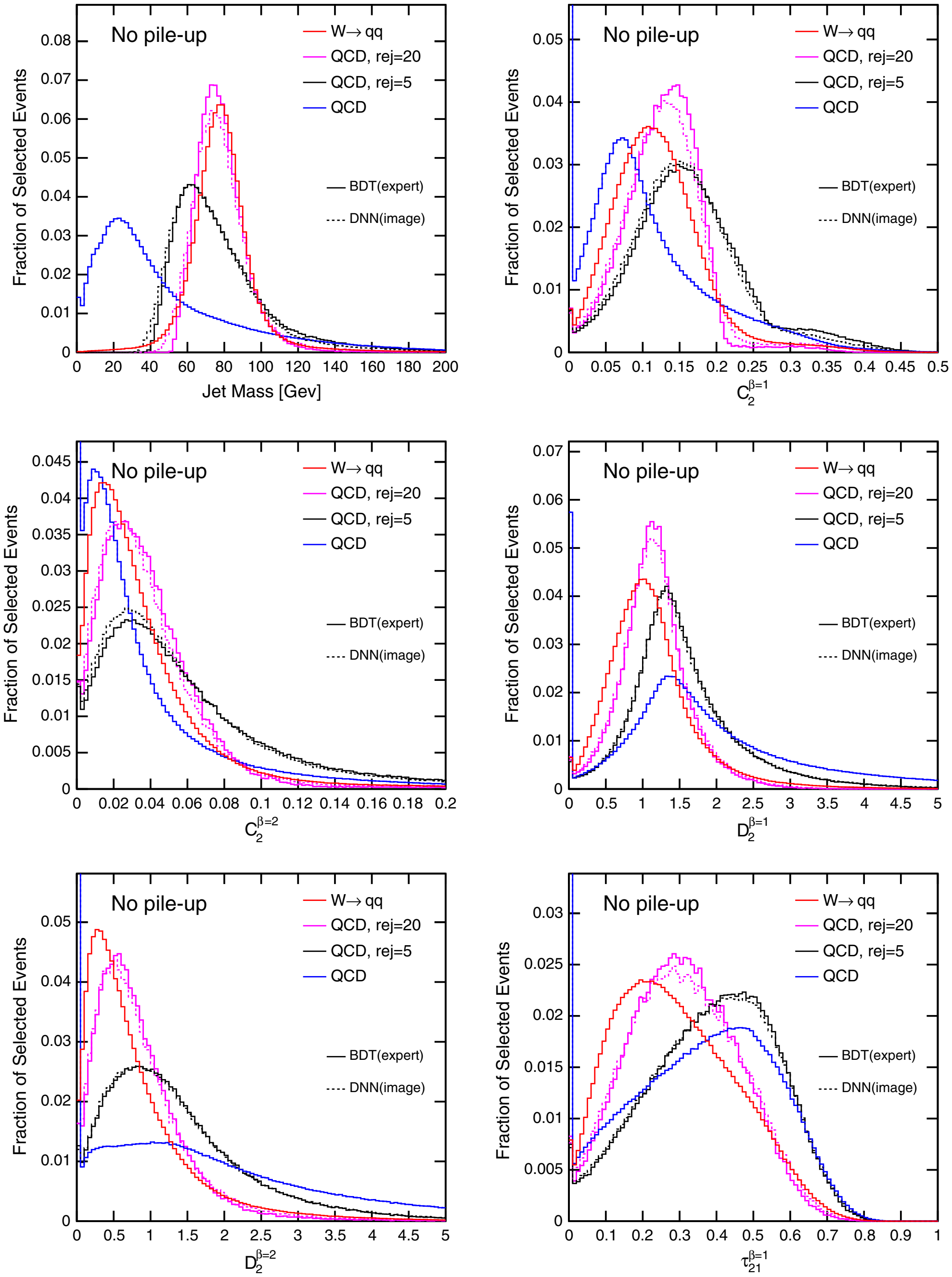


Figure A.1: Distributions for simulated high-level jet substructure observables used to discriminate between jet pairs produced via $W \to qq$ processes (red) and QCD jets from single quarks or gluons (blue).

Parameter	Value
Num. of hidden layers	5
Num. of hidden units	500
kernel size	(4,4)
strides	(1,1)
dropout layers	3
dropout rate	0.2
padding	VALID
kernel initializer	GLOROT NORMAL
activation	RELU
kernel constraint	`max_norm(3)`

Table A.1: Hyperparameter values used in CNN training on LL inputs for the benchmark network used in the black box guided search on jets from a boosted W as given in Chp. 3.

with a patience of 30 epochs and models are saved for best results only. Finally, dropout layers are included in between each convolutional or dense layer used in the trained models.

A.3 Baseline Convolutional Neural Network

For all convolutional neural networks, their inputs undergo a log transformation prior to processing with the `StandardScaler`. Each CNN uses the parameters given in Table A.1

A.4 Baseline Dense Neural Network

For all dense neural networks on HL inputs, including both EFPs and jet substructure observables, parameters used are given in Table A.2

Parameter	Value
Num. of hidden layers	3
Num. of hidden units	300
dropout layers	2
dropout rate	0.5
activation	RELU
kernel constraint	`max_norm(3)`

Table A.2: Hyperparameter values used in DNN training on all HL inputs for the benchmark network and guided networks used in the blackbbox guided search on jets from a boosted W as given in Chp. 3.

A.5 K-fold Validation

To calculate uncertainties on the trained model prediction accuracy, the bootstrap cross-validation package in SCI-KIT is used to equally divide the test set 10 times and measure the performance across 10 bootstrapping iterations. Averages and standard deviations are then calculated from these 10 iterations to define the central value and uncertainties of the AUC.

Appendix B

Electron Identification

B.1 Neural Network Hyperparameters and Architecture

Parameter	Range
Num. of conv. blocks	[1, 4]
Num. of filters	[8, 128]
Num. of dense layers	[1, 3]
Num. of hidden units	[1, 200]
Learning rate	[0.0001, 0.01]
Dropout	[0.0, 0.5]

Table B.1: Hyperparameter ranges for bayesian optimization of convolutional networks in the electron identification classifier.

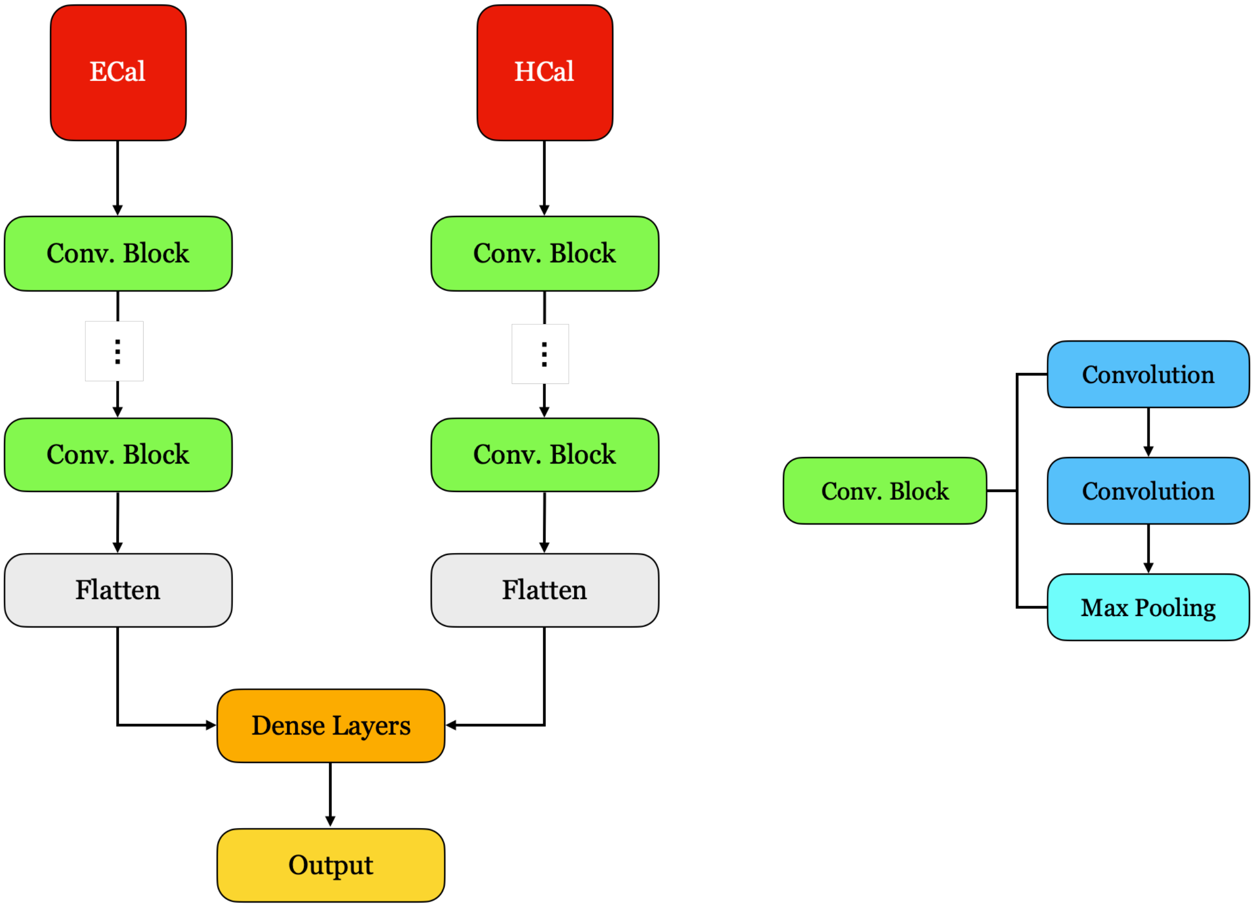


Figure B.1: Diagram of the architecture of the convolutional neural network (left) and diagram of a convolutional block (right) appearing in the final network architecture for the electron identification classifier.

Parameter	Range
Num. of dense layers	[1, 8]
Num. of hidden units	[1, 200]
Learning rate	[0.0001, 0.01]
Dropout	[0.0, 0.5]

Table B.2: Hyperparameter ranges for bayesian optimization of fully connected networks in the electron identification classifier

features	conv.	filters	dense	hidden	LR	DP
ECal	3	117	2	160	0.0001	0.0
Hcal	2	27	2	84	0.01	0.5
Ecal+HCal	3	47	2	146	0.0001	0.0
HL	-	-	5	149	0.001	0.0019

Table B.3: Best hyperparameters found per model in the electron identification classifier.

Appendix C

Prompt Muon Isolation

C.1 Neural network architectures

All networks were trained in Tensorflow[147] and Keras[146]. The networks were optimized with Adam [148] for up to 100 epochs with early stopping. For all networks except the PFNs, the weights were initialized using orthogonal weights[198]. Hyperparameters were optimized using Bayesian optimization with the Sherpa hyperparameter optimization library [150]. The variables and ranges for the hyperparameters are shown in tables C.1 and C.2.

Below are further details regarding the networks which use images and those which use isolation and EFP observables.

C.2 Muon Image Networks

The pixelated images were preprocessed to have zero mean and unit standard deviation. We tried rotating the images as in [28] but performance was considerably lowered by this

Parameter	Range	Value
Num. of convolutional blocks	[1, 4]	3
Num. of filters	[16, 128]	48
Num. of fully connected layers	[2, 4]	2
Number of hidden units	[25, 200]	74
Learning rate	[0.0001, 0.01]	0.0003
Dropout	[0.0, 0.5]	0.2388

Table C.1: Hyperparameter ranges for bayesian optimization of convolutional networks in the muon isolation classifier.

preprocessing step. The best muon image network structure begins with three convolutional blocks. Each block contains three convolutional layers with 48 filters with rectified linear units [143], followed by a 2x2 pooling layer. Afterwards there are two fully connected layers with 74 rectified linear units and a final layer with a sigmoidal logistic activation function to classify signal vs background. The model had dropout [145, 199] with value 0.2388 on the fully connected layers and an initial learning rate of 0.0003 and batch size of 128.

C.3 Particle-Flow Networks

The Particle Flow Network (PFN) is trained using the `energyflow` package[65]. Input features are taken from the muon image pixels and preprocessed by subtracting the mean and dividing by the variance. The PFN uses 3 dense layers in the per-particle frontend module and 3 dense layers in the backend module. Each layer uses 100 nodes, `relu` activation and `glorot_normal` initializer. The final output layer uses a sigmoidal logistic activation function to predict the probability of signal or background. The `Adam` optimizer is used with a learning rate of 0.0001 and trained with a batch size of 128.

Parameter	Range	ISO Value
Num. of layers	[2, 8]	2
Num. of hidden units	[1, 200]	197
Learning rate	[0.0001, 0.01]	0.0003
Dropout	[0.0, 0.5]	0.0547

Table C.2: Hyperparameter ranges for Bayesian optimization of fully connected networks in the muon isolation classifier.

C.3.1 Isolation Cone and EFP Networks

The isolation inputs and EFPs are preprocessed by subracting the mean and dividing by the variance. We trained neural networks with two to eight fully connected hidden layers depending on the hyperparameter value and a final layer with a sigmoidal logistic activation function to predict the probability of signal or background.

For the minimal set of isolation inputs, the best model we found had 2 fully connected layers with 197 rectified linear hidden units[200] and a learning rate of 0.0003 and dropout rate of 0.0547.

C.3.2 ADO comparison

In Fig. C.1, the ADO between the various networks is shown.

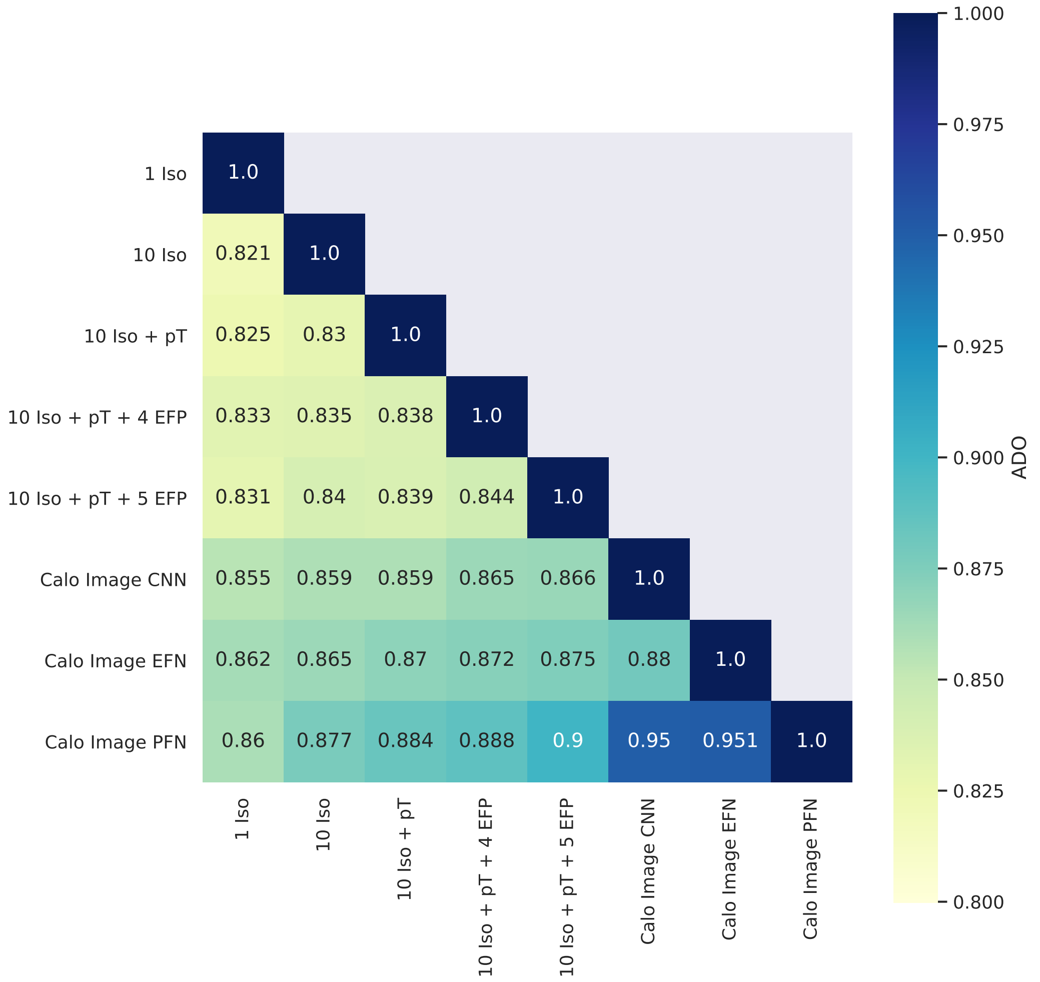


Figure C.1: Comparison of the similarity of decisions made by pairs of networks, as quantified by the Average Decision Ordering (ADO) in the muon isolation classifier.

Appendix D

Semi-visible Jet Identification

D.1 Jet Substructure Observables

D.1.1 Fundamental JSS

A common standard classification feature used across applications is the invariant jet mass, defined as,

$$M_{\text{jet}} = \sqrt{E^2 - \|\mathbf{p}\|^2} \tag{D.1}$$

Additionally, the sum of jet p_T constituents is included as a JSS observable both in the initial HL inputs and along with EFPs to give ML algorithms a relative scale for dimensionless EFP features to train with. The jet p_T sum is calculated by

$$p_\text{T} = \sum_{i \in \text{jet}}^{N} p_{\text{T,i}} \tag{D.2}$$

Distributions for both observables are shown in Fig. D.1.

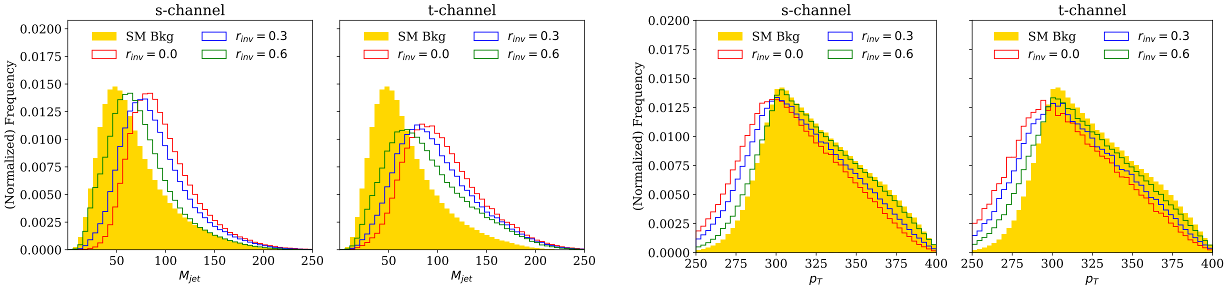


Figure D.1: Distribution plots for background (yellow) and signal (red, green, blue) processes for high-level features of M_{jet} and p_{T} in the semi-visibile HL jet classifier.

D.1.2 Generalized Angularities

A variety of existing JSS observables are defined by choices of κ and β parameters from the momentum fraction (z_i) and angular separation (θ_i) of a Generalized Angularity (GA) expression,

$$\lambda_\beta^\kappa = \sum_{i \in \text{jet}}^{N} z_i^\kappa \theta_i^\beta \tag{D.3}$$

The Les Houches Angularity (LHA) is defined from the GA expression with parameters $(\kappa = 1, \beta = 1/2)$ and p_{T}^D with $(\kappa = 2, \beta = 0)$. Written explicitly, these become

$$\text{LHA} = \sum_{i \in \text{jet}}^{N} z_i \theta_i^{1/2} \tag{D.4}$$

$$p_{\text{T}}^D = \sum_{i \in \text{jet}}^{N} z_i^2 \tag{D.5}$$

an additional value, e_{width} is produced by choices of $(\kappa = 1, \beta = 1)$,

$$e_{\text{width}} = \sum_{i \in \text{jet}}^{N} z_i \theta_i \tag{D.6}$$

$$\tag{D.7}$$

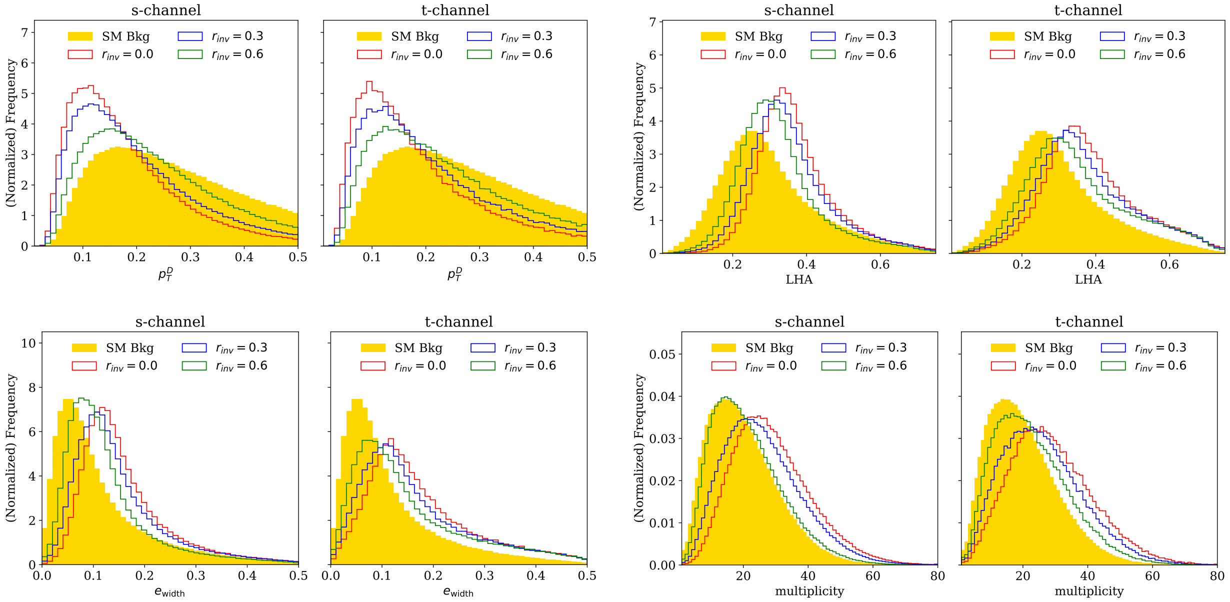


Figure D.2: Distribution plots for background (yellow) and signal (red, green, blue) processes for high-level features of p_{T}^{D}, LHA, e_{width} and multiplicity in the semi-visibile jet HL classifier.

Lastly, the multiplicity (although technically defined as simply the total number of constituents in the jet) can be expressed in this same generalized form for $(\kappa = 0, \beta = 0)$

$$\text{multiplicity} = \sum_{i \in \text{jet}}^{N} 1 \tag{D.8}$$

Distributions for all GA observables are shown in Fig. D.2

D.1.3 Energy Correlation

For Energy Correlation and their corresponding ratios, we start with the simple Energy Correlation functions $\mathrm{ECF}_1, \mathrm{ECF}_2$ and ECF_3,

$$\mathrm{ECF}_1 = \sum_i p_{\mathrm{T,i}} \tag{D.9}$$

$$\mathrm{ECF}_2^\beta = \sum_{i<j} p_{\mathrm{T,i}}\, p_{\mathrm{T,j}}\, (\theta_{ij})^\beta \tag{D.10}$$

$$\mathrm{ECF}_3^\beta = \sum_{i<j<k} p_{\mathrm{T,i}}\, p_{\mathrm{T,j}}\, p_{\mathrm{T,k}}\, (\theta_{ij}\theta_{ik}\theta_{jk})^\beta \tag{D.11}$$

and the related ratios are given by,

$$e_2^\beta = \frac{\mathrm{ECF}_2^\beta}{(\mathrm{ECF}_1)^2} \tag{D.12}$$

$$e_3^\beta = \frac{\mathrm{ECF}_3^\beta}{(\mathrm{ECF}_1)^3} \tag{D.13}$$

from these ratios, we then compute the energy correlation ratios C_2 and D_2

$$C_2 = \frac{e_3}{(e_2)^2} \tag{D.14}$$

$$D_2 = \frac{e_3}{(e_2)^3} \tag{D.15}$$

Distributions for all Energy Correlation observables are shown in Fig. D.3

D.1.4 N-Subjettiness

Unlike the previous JSS observables, the N-Subjettiness is based on iterating over the events candidate subjets. Given subjets isolated via clustering, for N candidate subjets, the N-

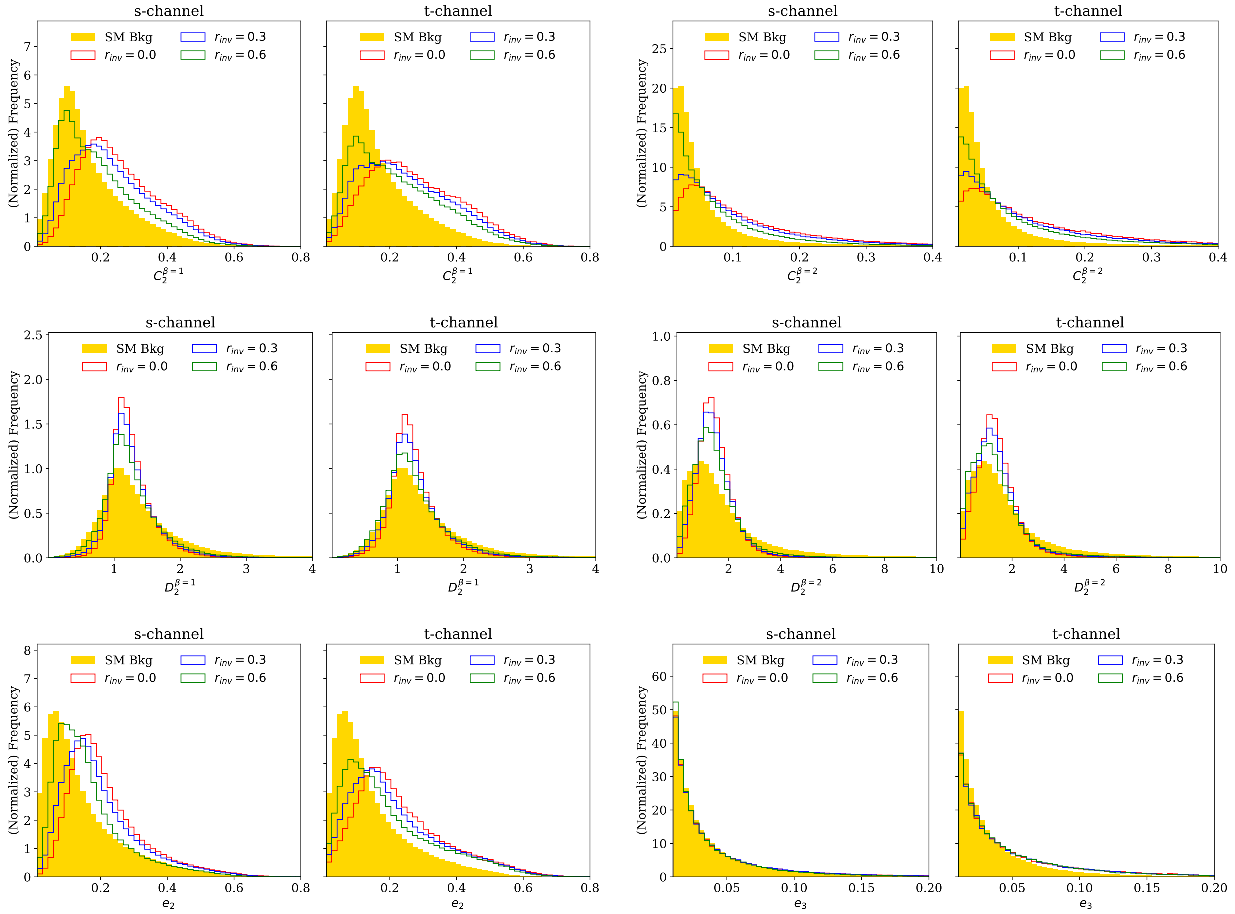


Figure D.3: Distribution plots for background (yellow) and signal (red, green, blue) processes for high-level features of energy correlation functions ($C_2^{\beta=1}$, $C_2^{\beta=2}$, $D_2^{\beta=1}$ and $D_2^{\beta=2}$) and pairs e_2 and e_3 in the semi-visibile jet HL classifier.

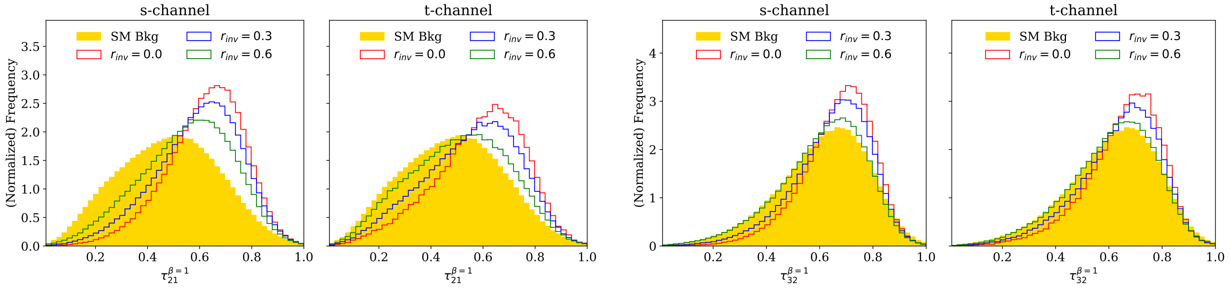


Figure D.4: Distribution plots for background (yellow) and signal (red, green, blue) processes for high-level features of N-subjettiness ($\tau_{21}^{\beta=1}$ and $\tau_{32}^{\beta=1}$) in the semi-visibile jet HL classifier.

subjettiness (τ_N) is defined as,

$$\tau_N = \frac{1}{d_0} \sum_k p_{\mathrm{T,k}} \mathrm{min}\left(\Delta\theta_{1,k}, \Delta\theta_{2,k}, \ldots, \Delta\theta_{N,k}\right) \tag{D.16}$$

where we define the normalization factor d_0 by,

$$d_0 = \sum_k p_{\mathrm{T,k}}\, R_0 \tag{D.17}$$

where R_0 is the characteristic jet radius used during clustering. Finally, the N-subjettiness ratios used are defined by

$$\tau_{21}^{\beta=1} = \frac{\tau_2}{\tau_1} \tag{D.18}$$

$$\tau_{32}^{\beta=1} = \frac{\tau_3}{\tau_2} \tag{D.19}$$

Distributions for both N-subjettiness observables are shown in Fig. D.4

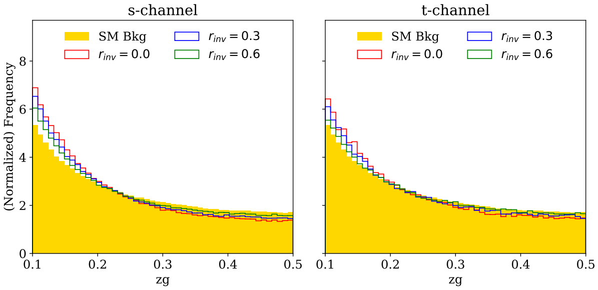


Figure D.5: Distribution plots for background (yellow) and signal (red, green, blue) processes for high-level features splitting function z_g in the semi-visibile jet HL classifier..

D.1.5 Groomed Momentum Splitting Fraction

The splitting fraction is described in terms of the Soft Drop grooming technique in Ref. [201]. The feature is calculated using `energyflow`[120] with the Cambridge/Aachen algorithm using a jet radius of $R = 1$ and Soft Drop parameters of $\beta = 0$ and $z_{\text{cut}} = 0.1$.

A distribution for z_g is given in Fig. D.5

D.2 ML Architectures

D.2.1 Deep Neural Networks

All deep neural networks were trained in Tensorflow[147] and Keras[146]. The networks were optimized with Adam [148] for up to 100 epochs with early stopping. For all networks, weights were initialized using orthogonal weights[198]. Hyperparameters were optimized using bayesian optimization with the Sherpa hyperparameter optimization library [150].Tables D.1 and D.2

s-channel

Parameter	Range	Value (by r_{inv})		
		0.0	0.3	0.6
Learning Rate	[0.00001, 0.001]	0.001	0.001	0.001
Dropout	[0, 0.5]	0.1	0.0	0.0
Dense Layers	[2,8]	4	7	4
Dense Units	[20,200]	100	200	200

t-channel

Parameter	Range	Value (by r_{inv})		
		0.0	0.3	0.6
Learning Rate	[0.00001, 0.001]	0.001	0.001	0.001
Dropout	[0, 0.5]	0.0	0.0	0.0
Dense Layers	[2,8]	4	7	4
Dense Units	[20,200]	200	200	200

Table D.1: Hyperparameter ranges for bayesian optimization of high-level deep neural networks

D.2.2 High-Level DNN

All HL features are preprocessed with Scikit's `StandardScaler` before training.

Deep Neural Networks

trained with two to eight fully connected hidden layers depending on the hyperparameter value and a final layer with a sigmoidal logistic activation function to predict the probability of signal or background. Individual parameters are given in Table D.1

Particle-Flow Networks

The Particle Flow Network (PFN) is trained using the `energyflow` package[120]. Input features are taken from the trimmed jet constituents and preprocessed by centering the in $(\eta - \varphi)$ space to the average p_T and normalizing pixel values to 1. The PFN uses 3 dense layers in the per-particle frontend module and 3 dense layers in the backend module. Both frontend and backend layers use 300 hidden nodes per layer. Each layer uses `relu` activation and `glorot_normal` initializer. The final output layer uses a sigmoidal logistic activation function to predict the probability of signal or background. The `Adam` optimizer is used and trained with a batch size of 128. Individual parameters are given in Table D.2

D.2.3 Boosted Learning Models

HL features are, again, preprocessed with Scikit's `StandardScaler` before training. Except where indicated, default settings are used.

LightGBM

All applications of LightGBM are trained using regression for binary log loss classification using Gradient Boosting Decision Trees. Performance is measured by the AUC metric for a maximum of 5000 boosting rounds and early stopping set to 100 rounds against AUC improvements.

XGBoost

All applications of XGBoost are trained using using the gradient tree booster and settings of `eta`=0.1, `subsample`=0.5, `base_score`=0.1, `gamma`=0.0, `max_depth`=6. Performance is s-channel

Parameter	Range	Value (by r_{inv})		
		0.0	0.3	0.6
Learning Rate	[0.00001, 0.001]	0.001	0.001	0.001
Latent Dropout	[0, 0.5]	0.2	0.2	0.2
Filter Dropout	[0, 0.5]	0.2	0.2	0.2
Phi Size	[100, 300]	300	300	300
Filter Size	[100, 300]	300	300	300

t-channel

Parameter	Range	Value (by r_{inv})		
		0.0	0.3	0.6
Learning Rate	[0.00001, 0.001]	0.001	0.001	0.001
Latent Dropout	[0, 0.5]	0.2	0.2	0.2
Filter Dropout	[0, 0.5]	0.2	0.2	0.2
Phi Size	[100, 300]	300	300	300
Filter Size	[100, 300]	300	300	300

Table D.2: Hyperparameter ranges for bayesian optimization of energy flow and particle flow networks trained on s-channel (top) and t-channel (bottom).

measured by the AUC metric for a maximum of 5000 boosting rounds and early stopping set to 100 rounds against AUC improvements.