

UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

Learning to Isolate Muons and Address Broken Symmetries with Encouraged Invariance

Permalink

<https://escholarship.org/uc/item/16s2p3z2>

Author

Witkowski, Edmund

Publication Date

2023

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

Learning to Isolate Muons and Address Broken Symmetries with Encouraged Invariance

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Physics

by

Edmund Witkowski

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

2023

DEDICATION

To Isaac, my first cat, who was with me for 18 years before passing away in 2020.

TABLE OF CONTENTS

	Page
LIST OF FIGURES	v
LIST OF TABLES	viii
ACKNOWLEDGMENTS	ix
VITA	xi
ABSTRACT OF THE DISSERTATION	xiii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Artificial Neural Networks	4
2 Learning to Isolate Muons	10
2.1 Introduction	11
2.2 Approach and Dataset	12
2.2.1 Data generation	13
2.3 Networks and Performance	17
2.4 Analysis	20
2.4.1 Search Strategy	20
2.4.2 IRC Safe Observables	23
2.4.3 IRC-unsafe Observables	25
2.5 Discussion	26
2.6 Conclusions	27
3 Learning to Isolate Muons in Data	29
3.1 Introduction	30
3.2 Dataset	33
3.3 Methods	39
3.4 Results	44
3.5 Conclusions	48

4	Learning Broken Symmetries with Resimulation and Encouraged Invariance	50
4.1	Introduction	51
4.2	Dataset	57
4.3	Encouraged Invariance	62
4.4	Results	64
4.5	Conclusions	67
	Bibliography	69
	Appendix A Appendix: Learning to Isolate Muons	76
	Appendix B Appendix: Learning to Isolate Muons in Data	80
	Appendix C Appendix: Learning Broken Symmetries with Encouraged Invariance	86

LIST OF FIGURES

	Page
1.1 The CMS detector, which consists of silicon trackers, a crystal electromagnetic calorimeter, a hadron calorimeter, a solenoid, and muon sub-detectors[33].	2
1.2 An example of a jet image produced from data collected by the CMS experiment[1].	3
1.3 A simple fully connected, feed-forward neural network, consisting of an input layer taking an input vector \mathbf{x} , two hidden layers, and an output layer yielding a scalar output y	5
1.4 Input nodes with elements from \mathbf{x} are used to calculate the value of an output node y , using weights \mathbf{w} . This illustration does not include the activation function or bias which may additionally be applied to the value y	6
1.5 Examples of commonly used activation functions. (Left) A rectified linear unit (ReLU), which is commonly used as the activation for hidden layers in modern networks. This introduces non-linearity and mitigates some issues which commonly arise during training with other functions. (Right) A sigmoid function, which ensures that values are in the range 0 to 1, commonly applied at the output layer of binary classifiers. (Bottom) A tanh function, which outputs values between -1 and 1 , is favored in cases where it is desirable to have an output centered at 0 and where both positive and negative signals may be informative.	7
2.1 Distributions of muon transverse momentum (top) and pseudorapidity (bottom) for signal and background samples. Afterwards, the distributions are weighted to make both samples uniform.	15
2.2 Mean calorimeter images for signal prompt muons (top) and muons produced within heavy-flavor jets (bottom), in the vicinity of reconstructed muons within a cone of $R = 0.45$. The color of each cell represents the sum of the E_T of the calorimeter deposits within the cell.	16
2.3 Comparison of classification performance using the metric AUC between Particle-Flow networks trained on lists of calorimeter deposits (orange, solid), Energy-Flow networks trained on lists of calorimeter deposits (red, solid), convolutional networks trained on muon images (blue, dashed) and networks which use increasing numbers of isolation cones (green, solid). For each number of cones, the optimal set is chosen.	18

2.4	Background rejection versus signal efficiency for Particle-Flow networks trained on lists of calorimeter deposits (orange, dashed), Energy-Flow networks trained on lists of calorimeter deposits (red, dashed), convolutional networks trained on muon images (blue, dashed), networks trained on a set of isolation cones (purple, dotted) and the benchmark approach, a single isolation cone approach (green, dashed).	19
2.5	Distributions of the \log_{10} of the selected IRC-safe EFPs as chosen by the black-box guided strategy, for prompt (signal) muons and non-prompt (background) muons.	24
2.6	Distributions of the \log_{10} of the selected EFPs as chosen by the black-box guided strategy, for prompt (signal) muons and non-prompt (background) muons.	25
3.1	Histogram of the dimuon invariant mass near the Z boson peak, for events in data with identical (yellow) or opposite (blue) electric charges. The unshaded area indicates the region for the oppositely-charge pairs which comprises our “prompt muon abundant” sample. The grey, shaded region for the oppositely-charge pairs, as well as the entire region for identically-charged pairs, comprise our “prompt muon moderate” sample.	33
3.2	Histograms of muon p_T and pseudorapidity η in the two samples with varying fractions of prompt muons, as defined in text and Fig. 3.1.	35
3.3	The average image of hadronic activity in the vicinity of an identified muon, in angular coordinates of azimuthal angle ϕ and pseudorapidity η , for our two training samples, one which is dominated by prompt muons (top) and a second which has a more moderate mixture of prompt and non-prompt muons (bottom). The muon itself is excluded from these visualizations, but the energies are normalized by that of the muon.	37
3.4	Histograms of the muon isolation (defined in Eq. 3.1) for each of our training samples, one of which is dominated by prompt muons, for two choices of isolation cone radius parameter $R_0 = 0.025$ (top) and $R_0 = 0.45$ (bottom).	38
3.5	A visualization of the masses overlaid with the fit and its prompt / non-prompt components. The shaded regions indicate events which are included in the relatively less prompt sample. Here we fit the full CMS sample used in the study, finding that it is $95.6 \pm 0.6\%$ prompt overall.	42
3.6	Isolation network performance shown as a function of number of input cones. Performance of the PFN and best performing high-level network are shown as benchmarks. ROC AUC is shown for each model (top) as well as the signal efficiency at a fixed background efficiency (bottom).	46
3.7	Distribution of the EFP observable identified in the search described by the text. Samples shown are separated by class using the sPlots weighting technique after applying a 50% background efficiency cut according to the outputs of the 9 isolation cone network. Also shown is the graph representation of the EFP.	47

3.8	Comparison of the performance of the networks described in Table 3.1, via ROC curves. Shown is background rejection (inverse of efficiency) versus signal efficiency.	49
4.1	Demonstration of the breaking of a continuous rotation symmetry by the pixelization of a realistic detector. (Left) shows an ideal detector that performs no binning, while (Right) is a realistic detector which produces a pixelated image. (Top) shows a jet incident on the detector, and the images produced by each. (Middle) shows the image produced by the realistic detector rotated by an angle that is not a multiple of $\frac{\pi}{2}$. Rotating a pixelated image by such an angle results in artifacts and does not produce a detector image which reflects the true symmetry. (Bottom) shows the case where the jet itself is rotated pre-detector, producing an image which accurately represents the symmetry of the problem. Though it is not closed under rotation, it avoids introducing artifacts from post-detector rotation.	54
4.2	Demonstration of the significance of rotation-induced artifacts on less sparse (left) and more sparse (right) images by comparing original (top) and rotated images (bottom). Visually, the image on the left (CIFAR-10[64]) appears relatively unchanged after the rotation. The artifacts in the image on the right are far more prominent, and so might have relatively more influence on any learned strategy.	55
4.3	Visualization of dataset generation process. (Top) An example of an event before pixelization, where the size of each deposit is proportional to its energy. (Bottom) The same event, after the deposits have been distributed over a small area to simulate shower width effects which can lead to deposits over adjacent pixels. In the bottom pane the size of the deposits is arbitrary.	59
4.4	(Top) The event shown in Fig. 4.3 with uniform binning (Top) and rectangular binning (Bottom) applied.	60
4.5	The event shown in Fig. 4.3 rotated at 45 degrees from their original orientation with uniform binning (Top) and rectangular binning (Bottom) applied. Applying rotations prior to binning (Left) avoids the interpolation artifacts which arise from applying rotations after binning (Right), where the images look relatively blurrier and more washed out.	61
4.6	Performance of FCNs (left) and PFNs (right) trained on uniformly binned data (top) or non-uniformly binned data (bottom) as a function of training set size. Though results depend on the nature of the task and the structure of the network, pre-detector augmentation and resimulation typically improves the learning rate, and encouraged invariance provides a further boost in learning.	65

LIST OF TABLES

	Page
2.1 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 inputs to and parameters of each network.	26
3.1 Comparison of the performance of the various networks discussed in the text. Performance is measured through ROC AUC, as well as signal efficiency (TPR) at 50% background efficiency. Standard error is evaluated to be $\lesssim 1 \times 10^{-3}$ for both metrics over a 1σ confidence interval (see Sec. 4.3 for details on calculation). While the reported performance values refer only to testing done on CMS data, the “EFP Scan” column indicates whether the EFP inputs used were identified as useful by a scan over CMS or simulated data. These results correspond to the ROC curves in Fig 3.8.	48
4.1 The ROC AUC performance for models with various augmentation strategies described in the text, trained and evaluated on events with uniform or non-uniform pixelization, shown for the smallest and largest training set sizes tested.	67

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Daniel Whiteson, for giving me the opportunity to work on interesting problems in physics, for providing me with guidance and support in developing my research and writing skills, and for overall making my experience in grad-school positive and productive.

I would also like to thank my other co-authors who directly contributed to the work presented in this thesis. These are Julian Collado, Kevin Bauer, Taylor Faucett, Pierre Baldi, and Benjamin Nachman. Additionally, I am grateful for the many people who have provided useful feedback on our work over the years, including Aishik Ghosh, Mike Fenton, Alexis Romero, Jason Baretz, Kevin Greif, Jacob Hollingsworth, Jessica Howard, Jesse Thaler, Chase Shimmin, Dan Guest, David Shih, Troels Peterson, Gregor Kasieczka, and Yuzo Kanomata.

I would like to acknowledge that I received funding from the University of California, Irvine, in the form of TA stipends and a department fellowship, and that a portion of my work was funded by the National Science Foundation through the MAPS Training Program. The work presented here was done with support from the Department of Energy Office of Science under contract DE-AC02-05CH11231, the National Science Foundation under grants 1633631, 1839429, and a hardware grant from NVIDIA.

Much of the work presented here was made possible through the use of the Greenplanet facilities, and I am thankful to Nathan Crawford for quickly fixing any technical issues we encountered as they arose. Further, a portion of this work makes extensive use of data provided freely by the CMS collaboration, through CMS Open Data[1]. Finally, all of our work could not have been done without the use of python[90], along with many publically available libraries, including Pytorch[72], Pytorch-Lightning[46], Tensorflow[12], Numpy[54], Scipy[91], Scikit-Learn[74], Madgraph[15], Pythia[86], and Delphes[42].

I am also grateful to David Kirkby and Andrew Lankford, for agreeing to serve on my defense committee, as well as Timothy Tait and Pierre Baldi for serving on my advancement committee.

I would also like to thank David McGee and Romulo Ochoa from The College of New Jersey, for not only the excellent physics education they provided, but for allowing me to work with them in their optics labs as my introduction to scientific research, and encouraging me to pursue a graduate degree. I would like to thank Regina Demina and Sergey Korjenevski at the University of Rochester, for letting me assist them in contributing to the CMS collaboration, and for providing me with a very interesting and enjoyable summer research experience. I would also like to thank Mu-Chun Chen, who gave me my first research opportunity at UCI and allowed me to do exciting work on the theoretical side of high-energy physics.

I am grateful to my parents, the rest of my family, and my friends, for all of the help they have provided as I pursued my degree. Particularly, I would like to thank Christian

Courtney and Kyle Alinea, who have been very supportive and remained two of my closest friends throughout much of my life. I would also like to express my gratitude towards the friends that I have gained since coming to UCI, who have made grad-school a much more fun and pleasant experience. These include Beda, Nitish, Igor, Ben, Angelina, Jose, Rob, Bri, Olivia, Corey, Aishik, Arianna, Dillon, Eric, Daniel, Genevieve, Michael, Kenny, and Abby.

Finally, I would like to thank my cat Wolfie, for always being supportive and overall being a great friend.

VITA

Edmund Witkowski

EDUCATION

Doctor of Philosophy in Physics University of California, Irvine	2023 <i>Irvine, California</i>
Bachelor of Science in Physics The College of New Jersey	2017 <i>Ewing, New Jersey</i>
Studies in Chemistry Rider University	2013 - 2014 <i>Lawrenceville, New Jersey</i>

RESEARCH EXPERIENCE

Graduate Research Assistant University of California, Irvine	2019–2023 <i>Irvine, California</i>
Undergraduate Research Assistant The College of New Jersey,	2015-2017 <i>Ewing, New Jersey</i>
Undergraduate Research Assistant University of Rochester	2016 <i>Rochester, New York</i>

TEACHING EXPERIENCE

Teaching Assistant University of California, Irvine	2017–2023 <i>Irvine, California</i>
Tutor The College of New Jersey	2015–2017 <i>Ewing, New Jersey</i>

REFEREED JOURNAL PUBLICATIONS

Learning Broken Symmetries with Resimulation and Encouraged Invariance 2023

Edmund Witkowski, Daniel Whiteson

Learning to Isolate Muons in Data 2023

Edmund Witkowski, Benjamin Nachman, Daniel Whiteson
Physical Review D

Learning to Isolate Muons 2021

Julian Collado, Kevin Bauer, Edmund Witkowski, Taylor
Faucett, Daniel Whiteson, Pierre Baldi
Journal of High Energy Physics

CONFERENCE TALKS

Learning Broken Symmetries with Resimulation and Encouraged Invariance Nov 2023

Edmund Witkowski, Daniel Whiteson
ML4Jets2023

Learning to Isolate Muons in Data Nov 2022

Edmund Witkowski, Benjamin Nachman, Daniel Whiteson
ML4Jets2022

CONFERENCE POSTERS

Fabrication and Characterization of DR1 SU8 Holographic Thin Films June 2017

Benjamin Campos, Edmund Witkowski, David McGee
Laser World of Photonics

Photopatterned Surface Relief Gratings in Azobenzene-Amorphous Polycarbonate Thin Films March 2016

Benjamin Campos, Edmund Witkowski, David McGee
APS March Meeting

ABSTRACT OF THE DISSERTATION

Learning to Isolate Muons and Address Broken Symmetries with Encouraged Invariance

By

Edmund Witkowski

Doctor of Philosophy in Physics

University of California, Irvine, 2023

Professor Daniel Whiteson, Chair

We demonstrate techniques to improve the performance of data driven methods used in collider experiments, through the use of neural networks. First, using a simulated muon dataset, we probe the discriminating power of the typically used isolation observable by comparing it to neural networks trained on full event details. By performing a search over the space of Energy Flow Polynomials (EFPs), a set of scalar observables which performs similarly to the full information is identified. This methodology is then applied to real collider data obtained from CMS Open Data. The CMS data lacks event level class labels, necessitating the use of CWoLa, a weakly supervised training method, along with an sPlots-based performance evaluation method. Once again, we successfully identify a minimal set of scalar observables capable of outperforming isolation. Finally, a novel data-augmentation scheme is introduced. Symmetries present in an ideal dataset may be broken by detector effects, leading to lower quality augmented copies. We perform augmentation pre-detection in simulation, and further encourage invariance across augmented copies during training. We find that synthesizing examples this way leads to faster convergence, and that encouraging invariance yields further performance gains.

Chapter 1

Introduction

1.1 Background and Motivation

The Standard Model of particle physics has demonstrated remarkable success in explaining many of the fundamental aspects of the universe, but it has some significant shortcomings. These include not leading to predictions consistent with general relativity, lacking a viable dark matter candidate, and not providing an explanation for baryon asymmetry[66]. The pursuit of a more complete model has led to the development of many experiments designed to search for new fundamental physics. These include the ATLAS and CMS experiments at CERN, which collect and analyze large amounts of detector data produced by the Large Hadron Collider (LHC). These experiments curate massive datasets, with ATLAS alone reporting 10,000 TB of data a year[18]. This data potentially holds clues which could lead to the discovery of new physics, and constrain the space of theories which should be considered. In order to identify useful information within the vast quantity collected, it is natural to employ the techniques that have been developed in the realm of data science. The application of artificial neural networks to high-energy physics experiments is a

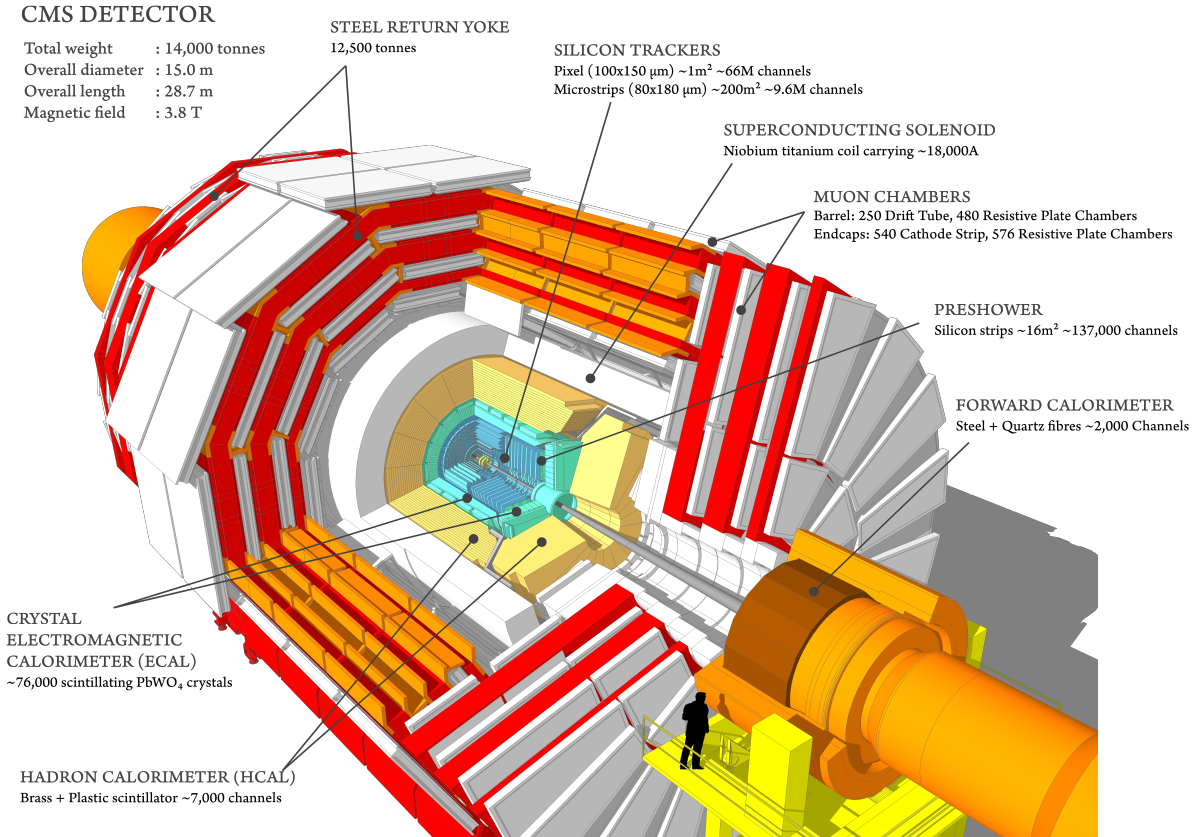


Figure 1.1: The CMS detector, which consists of silicon trackers, a crystal electromagnetic calorimeter, a hadron calorimeter, a solenoid, and muon sub-detectors[33].

relatively recent development, yielding impressive results in tasks ranging from simulation to classification[10, 53, 39, 58] Neural networks will be further detailed in section 1.2, but at a high level, they use an iterative process to automatically “learn” a mapping, given some training data and a properly defined loss function to guide the network to the desired solution. This process can present certain advantages over more traditional methods.

A specific example arises when working with jet images, which inherently possess high dimensionality. High dimensional data notoriously presents significant issues for many numerical and machine learning methods; this is a phenomenon referred to as “the curse of dimensionality.” In order to mitigate this, the high dimensional jets are typically reduced to theoretically motivated scalar observables. While these can be quite powerful, they do not necessarily capture all of the useful information present in the data. However, as we will

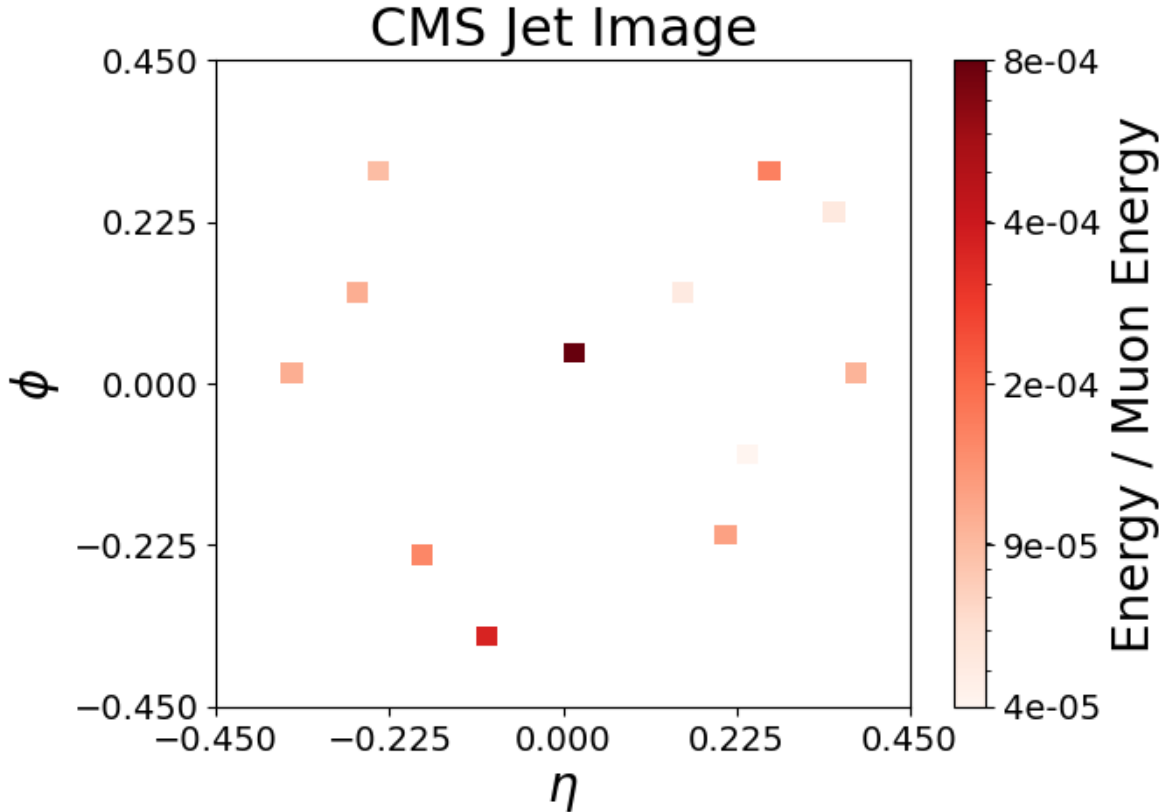


Figure 1.2: An example of a jet image produced from data collected by the CMS experiment[1].

demonstrate in Chapters 2 and 3, neural networks can not only be used to directly perform dimensionality reduction in a more optimal way, but they may also assist in identifying the optimal scalar observables in a given set. Using existing observables may be preferable, as they typically have a clearer physical meaning, highlighting one of the weaknesses of neural networks, which is a lack of human interpretability. Another disadvantage of neural networks is that they typically require very large amounts of training data, and can be quite computationally intensive to train, optimize, and evaluate. Even in the case that a large amount of collider data is available, this may not be the case in a given region of interest. This motivates the investigation of methods for improving the performance of neural networks in the presence of limited training data. Previous work has found that symmetries present in a dataset may be leveraged to improve data efficiency. These techniques typically involve

synthesizing new training examples by applying transformations to the existing data[81], or integrating the structure of a symmetry into the design of a network[30]. However, directly applying these solutions to jet images can lead to less than ideal performance. Symmetries expected to be present in jet images may be slightly broken by detector effects, and attempting to apply transformations after this may not be straightforward or may give rise to artifacts. Chapter 4 presents a data-augmentation technique which mitigates these issues by modifying data produced in simulation, prior to the introduction of detector effects.

This thesis aims to further delve into these challenges, and elaborate on our proposed solutions. Our overarching goal is to explore and demonstrate how neural networks can be harnessed both to enhance traditional methods in high-energy physics analyses, and to optimize neural network methodologies specifically for this context. In order to ensure clarity and lay the groundwork for the studies that follow, we now provide a broad overview of neural networks and how they operate.

1.2 Artificial Neural Networks

Artificial neural networks are parameterized machine learning models loosely inspired by biological neural networks[68]. They are typically organized as graphs consisting of nodes, referred to as neurons, interconnected by edges representing weights. Each neuron contains a value, either provided as input or computed within the network, and the weights control the strength of the interactions between connected neurons. These are organized into layers, with the first referred to as the input layer, the last as the output layer, and with a number of “hidden” layers in between. At each layer, a scalar bias may be added and an activation function applied, expanding the class of functions which the network may model. According to various formulations of universal approximation theorems, a network containing a single hidden layer and an appropriate non-linear activation function is capable of approximating

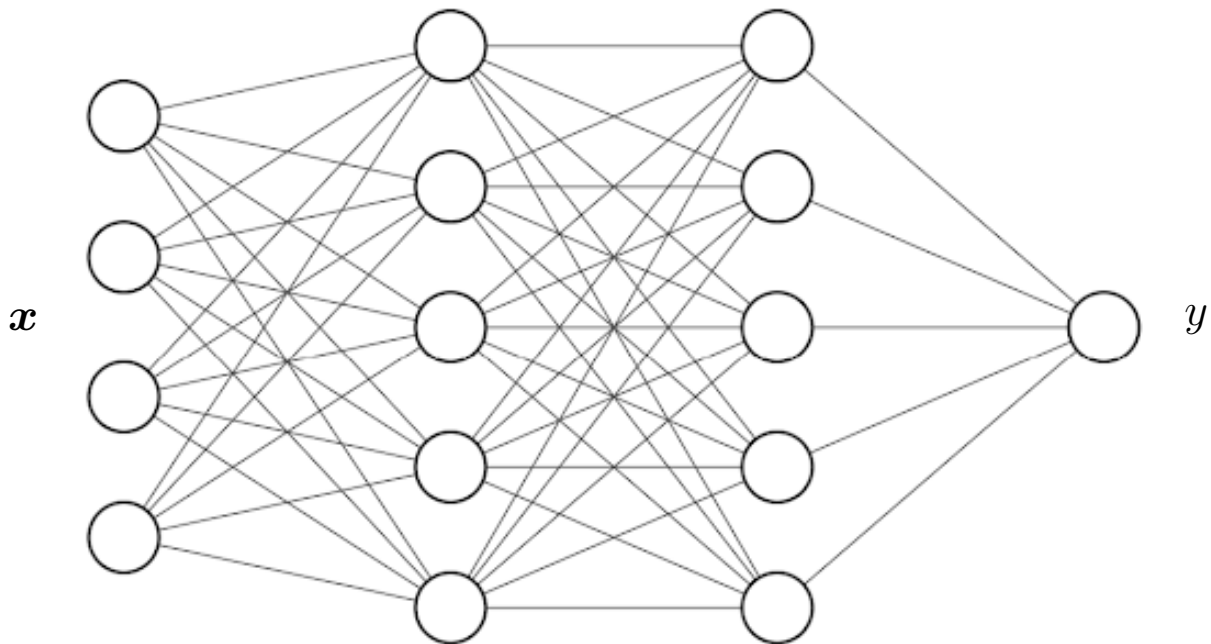


Figure 1.3: A simple fully connected, feed-forward neural network, consisting of an input layer taking an input vector \mathbf{x} , two hidden layers, and an output layer yielding a scalar output y .

a broad range of continuous functions, specifically those that are Borel measurable, to arbitrary precision[57]. In practice, the use of multiple hidden layers is found to allow iterative optimization methods to converge to a solution more efficiently than a single layer. This gives rise to the many-layered “deep” neural networks, which form the basis of many successful neural network solutions.

More formally, a network may be represented as a mapping F , which takes an input vector \mathbf{x} from an input space \mathcal{X} , to an output vector \mathbf{y} in an output space \mathcal{Y} . For a network composed of N layers, where the i th layer applies a transformation f_i , a network may be expressed as a nested composition of its layers

$$F(\mathbf{x}) = f_N \circ f_{N-1} \circ \dots \circ f_2 \circ f_1(\mathbf{x}) \tag{1.1}$$

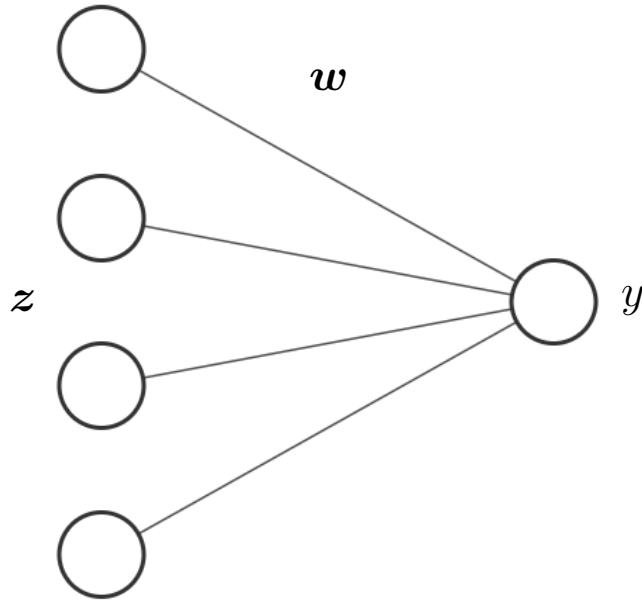


Figure 1.4: Input nodes with elements from \mathbf{x} are used to calculate the value of an output node y , using weights \mathbf{w} . This illustration does not include the activation function or bias which may additionally be applied to the value y .

. A given layer in a basic neural network consists of an activation function σ_i applied to the linear transformation defined by the weights \mathbf{W}_i and biases b_i .

$$f_i(\mathbf{z}) = \sigma_i(\mathbf{W}_i\mathbf{z} + b_i) \tag{1.2}$$

The specific choice of activation function may vary depending on the task and type of layer, but it should be noted that for most learning algorithms it is required to be differentiable.

Certain aspects of the network, termed hyperparameters, are typically specified manually. These include the number of neurons in each layer, as well as the number of hidden layers used, among other things. In contrast, the weights and biases are optimized iteratively through a backpropagation algorithm, allowing the network to find a solution which yields the desired results. The first step of this process is known as the forward pass, in which inputs are fed through the network, their respective outputs are evaluated, and then passed to a “loss” function. The loss function can take different forms depending on the task, but

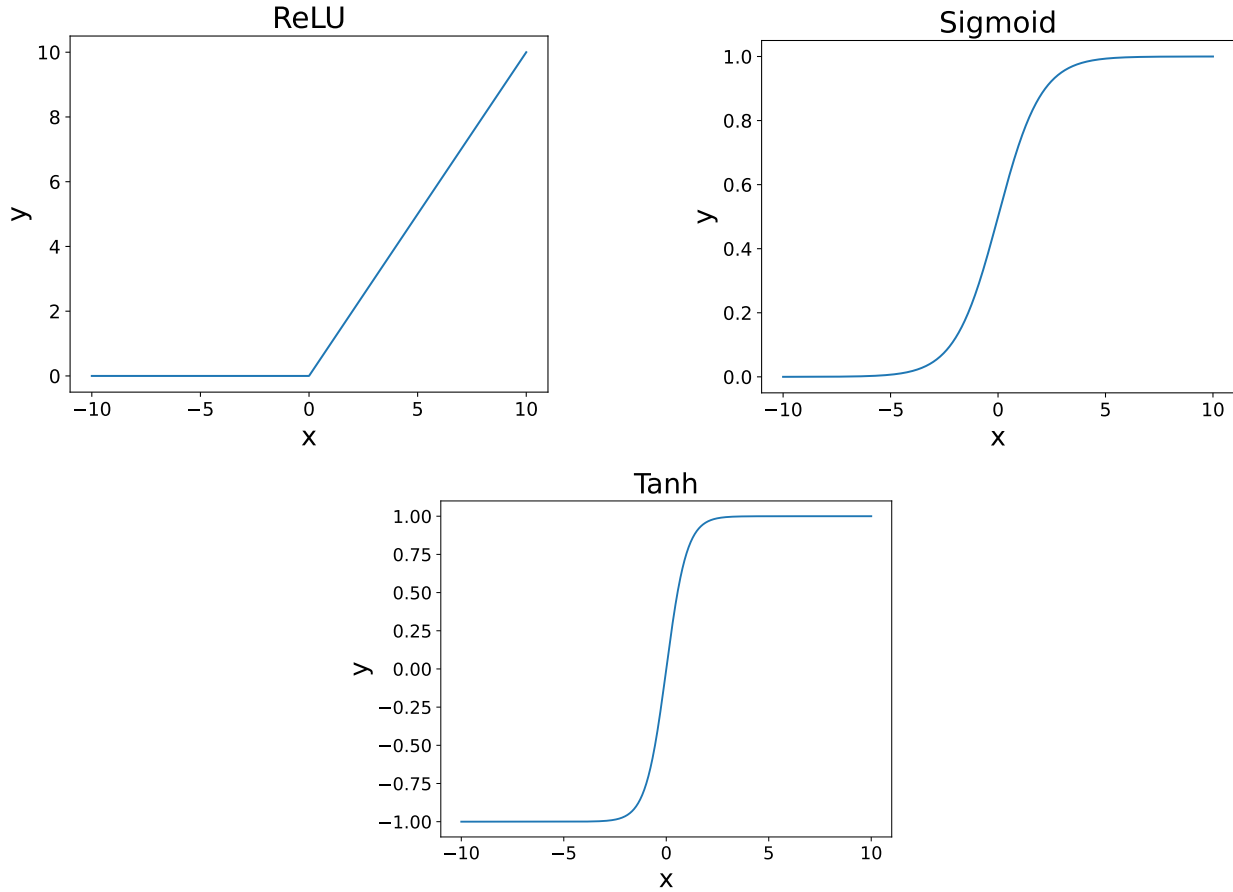


Figure 1.5: Examples of commonly used activation functions. (Left) A rectified linear unit (ReLU), which is commonly used as the activation for hidden layers in modern networks. This introduces non-linearity and mitigates some issues which commonly arise during training with other functions. (Right) A sigmoid function, which ensures that values are in the range 0 to 1, commonly applied at the output layer of binary classifiers. (Bottom) A tanh function, which outputs values between -1 and 1 , is favored in cases where it is desirable to have an output centered at 0 and where both positive and negative signals may be informative.

generally it quantifies how close the outputs are to the desired solution. Next, the gradients of the loss with respect to the parameters of the model are evaluated, necessitating that the chosen loss function should be differentiable. During the forward pass, values computed at each step are stored, and used with an automatic differentiation algorithm to compute the gradients. A network can be represented as a computational graph, with nodes being mathematical operations, and edges being the data which these operations act on. As long as each operation is differentiable, the chain rule allows gradients to be computed in

succession as the graph is traversed, typically starting from the output in the case of neural networks. This process is particularly efficient, and is crucial for the feasibility of training large networks. The gradients indicate how the model parameters should be updated in order to decrease the loss, increasing performance. This update process is broadly referred to as gradient descent, and the exact details of how the update is done can vary depending on the optimization algorithm used. Different algorithms may introduce various hyperparameters to control aspects of the optimization, among these typically being a learning rate, which scales the size of each update. As the gradients are computed and parameters are updated starting from the output layer, this is known as the backwards pass, and serves as the namesake of the backpropagation algorithm. This is done repeatedly over a training dataset, and stopped when the training loss is determined to be sufficiently minimal and stable. Gradient descent is not without challenges, however. Issues such as exploding gradients, where values grow too large and become numerically unstable, or vanishing gradients, where the gradients become extremely small and no longer contribute meaningfully to updates, can hinder convergence. These issues are largely mitigated through the use of activation functions such as rectified linear units (ReLU), and modern optimizers. Another significant issue is overfitting, where a model is learned which closely reproduces the training data, but does not generalize well to new examples. Neural networks are frequently largely over-parameterized, further exacerbating the likelihood of overfitting. There are several common ways to mitigate this. For example, a simple and frequently used method is to evaluate the model on a validation dataset after each update. This data is not used to update the model itself, and so if the training loss continues to decrease while the validation loss either stabilizes or begins to increase, it can indicate that the model is beginning to overfit and training should be halted. Other techniques to avoid overfitting include using dropout during training, where some neurons are randomly ignored at each iteration, or the inclusion of regularization terms in the loss which can encourage smaller weights and simplify the model.

While these principles can be broadly applied to the understanding of many artificial neural network methods, and are relied on greatly in the studies presented here, other modes of learning do exist which require additional steps or significant changes to the overall procedure. For our purposes, we will be exploring fully supervised learning, where each training example comes with a label that is supplied to the loss function, as well as weakly supervised learning, where these individual labels are unavailable. In the weakly supervised setting, we present a method which relies on macroscopic sample information to synthesize training labels, which are provided to the loss as in the fully supervised case. In terms of network architectures, we use fully connected neural networks, convolutional neural networks, and a more physics specific architecture known as a Particle Flow Network, which is deeply related to the other two. Fully connected networks present a straightforward design, in which every neuron in a given layer is connected to every neuron in the following layer. Convolutional neural networks apply a convolution operation to the data at each layer, sliding a window composed of trainable weights over groups of neurons. This allows the network to better capture spatial relationships, making it well suited to tasks involving images, such as those constructed from collider data. In the work that follows, the methods presented here lay the foundation for techniques to improve high-energy physics analyses.

Chapter 2

Learning to Isolate Muons

This chapter is heavily based on work previously published in collaboration with Julian Collado, Kevin Bauer, Taylor Faucett, Daniel Whiteson, and Pierre Baldi[38].

Distinguishing between prompt muons produced in heavy boson decay and muons produced in association with heavy-flavor jet production is an important task in analysis of collider physics data. We explore whether there is information available in calorimeter deposits that is not captured by the standard approach of isolation cones. We find that convolutional networks and particle-flow networks accessing the calorimeter cells surpass the performance of isolation cones, suggesting that the radial energy distribution and the angular structure of the calorimeter deposits surrounding the muon contain unused discrimination power. We assemble a small set of high-level observables which summarize the calorimeter information and close the performance gap with networks which analyze the calorimeter cells directly. These observables are theoretically well-defined and can be studied with collider data.

2.1 Introduction

Searches for new physics and precision tests of the Standard Model at hadron colliders have long relied on leptonic decays of heavy bosons, due to the relatively low background rates and excellent momentum resolution compared to hadronic final states. In the case of muons, the primary source of background to prompt muons (those from W, Z or other bosons) is production within a heavy-flavor jet. This non-prompt background is largest at lower values of muon transverse momentum, which has become important in searches for supersymmetry [4, 78, 59] as well as low-mass resonances [56].

The current state of the art strategy for distinguishing prompt and non-prompt muons in experimental searches involves techniques which integrate information from multiple detector systems [82, 73]. 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 [83], here we focus on the nature of the information available in the calorimeter. There, the traditional approach is to use a robust and simple method, measuring:

$$I_\mu(R_0) = \sum_{i, R < R_0} \frac{p_T^{\text{cell } i}}{p_T^{\text{muon}}}$$

within a cone $R = \sqrt{\Delta\phi^2 + \Delta\eta^2} < R_0$ surrounding the muon [6]. Typically a single cone is used, with values of R_0 in the 0.1-0.45 range. This approach relies on identifying a typical characteristic of the signal, low calorimeter activity in the vicinity of the muon.

The traditional strategy, however, focuses on the simple nature of the signal and may overlook the rich set of characteristics offered by the background object, which can provide handles for additional rejection power. Related work, which approaches similar object classification tasks as a background jet rejection problem, has shown significant improvement in background

discrimination when applied to photons [10, 53], pions [17] or electrons [39]. Other studies have shown that muons which fail the traditional isolation requirement can contain power to reveal new physics [29].

At the same time, there have been significant advances in machine learning techniques and their applications in physics [22, 19], specifically in the context of jet classification tasks, which take a fuller view of the object by directly analyzing the low-level calorimeter energy deposits, representing them either as a type of image [36, 20] or as a list [62].

It seems likely, therefore, that these machine learning strategies may identify the presence of significant additional calorimetric rejection power in the context of prompt muon identification. In this paper, we apply machine learning tools similar to those developed for jet calorimeter analysis to the task of distinguishing muons due to heavy boson decay from those produced within a heavy-flavor jet, analyze the nature of the information being used, and assemble a set of interpretable calorimeter features which capture that additional classification power.

2.2 Approach and Dataset

The observable $I_\mu(R_0)$ is a powerful discriminator which reduces a large amount of information to a single high-level scalar. However, it is possible that it fails to capture the fullness of the calorimeter information available to distinguish prompt muons from those which are produced within a jet. To probe whether information has been lost, we compare the performance of deep neural networks which access the full calorimeter information to shallow networks which use one or more isolation cones.

Neural network decisions are notoriously difficult to reverse-engineer [32, 23, 76, 95, 13], especially when the dimensionality of the data is large, as is the case for networks which

directly use the low-level calorimeter cells. Understanding the nature of the decisions is particularly vital when the training is done with simulated samples, as it leads to valid concerns about the application of such complex strategies to collider data.

In this study, our goal is not to develop deep networks for use in collider data. Instead, we apply these deep networks as a probe, to measure a loose upper bound on the possible classification performance, and provide insight into whether information has been lost in the reduction of the calorimeter cells to isolation cones.

Where information has been lost, we attempt to capture it, not by applying the deep network, but by assembling a small set of new high-level (HL) observables that bridge the performance gap and reproduce the classification decisions of the calorimeter cell networks [47]. These high-level observables are more compact, physically interpretable, can be validated in data, and allow for the straightforward assessment and propagation of systematic uncertainties.

2.2.1 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 [15], showered and hadronized with PYTHIA v8.235 [86], and the detector response simulated with DELPHES v3.4.1 [42] using the standard ATLAS card and ROOT version 6.0800 [28]. The classification of these objects is sensitive to the presence of additional proton interactions, referred to as pile-up events. We overlay such interactions 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 $p_T \in [10, 15]$ GeV with $|\eta| < 2.53$ were considered; see Fig. 2.1. To

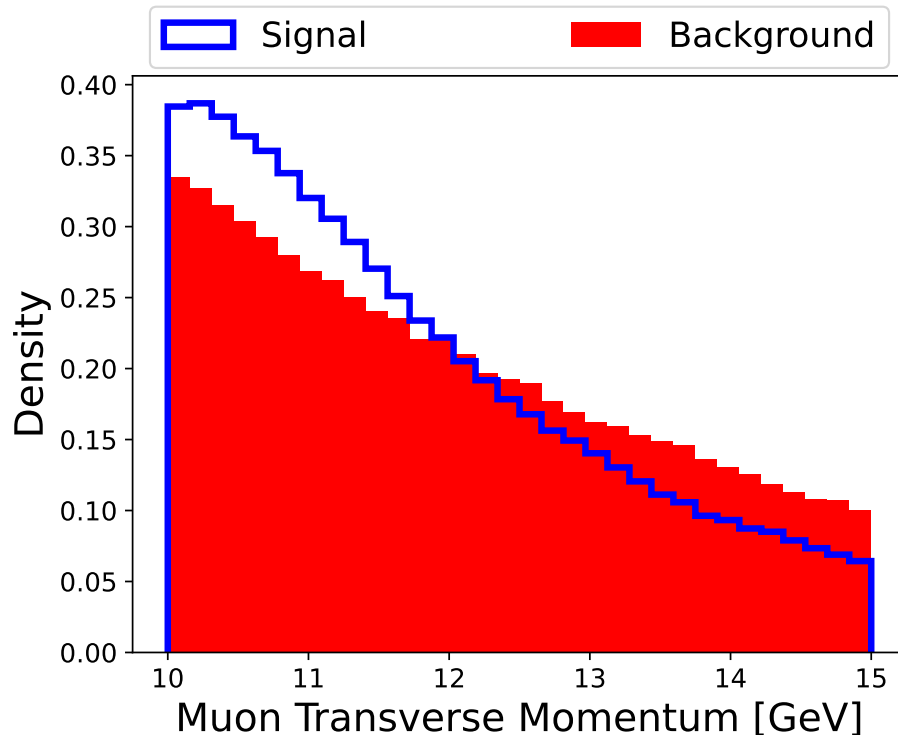
avoid inducing biases from artifacts of the generation process, signal and background events are weighted such that the distributions in p_T and η are uniform, using 32 bins in each dimension. Only events where a muon is identified as a track in the muon spectrometer are used. In total, 499,970 events were used, where 249,991 were signal and 249,979 were background. Both the signal and background datasets are randomly split as: 83% training, 8.5% validation, and 8.5% testing sets.

Calorimeter deposits can be represented as images where each pixel value represents the E_T deposited by a particle [36]. Images are formed by considering cells in the calorimeter within a cone of radius up to $\Delta R = 0.45$ surrounding the muon location after propagating to the radius of the calorimeter.

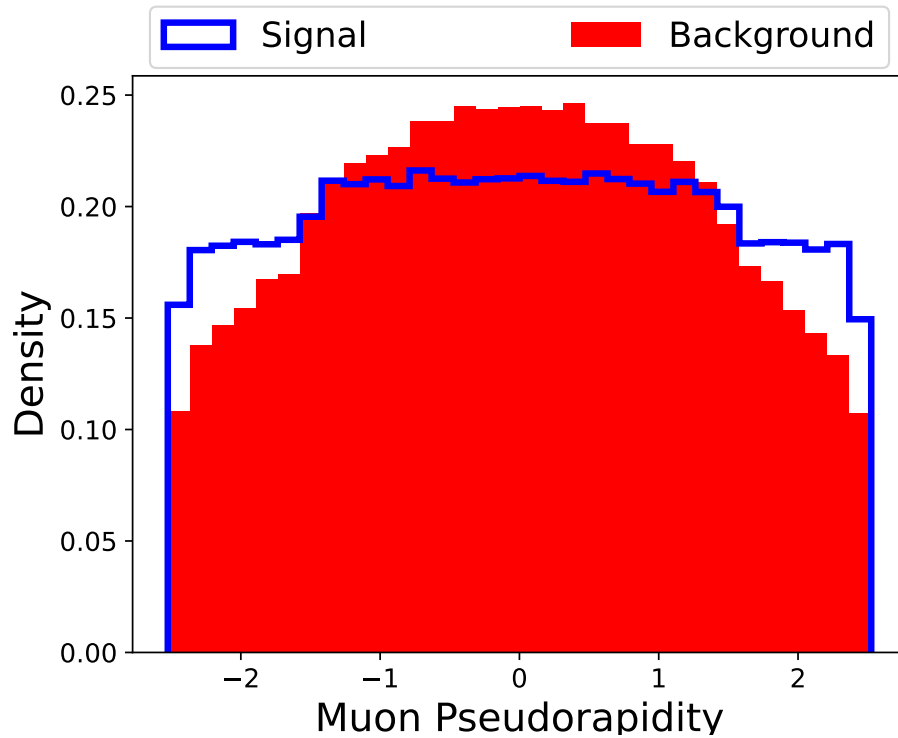
We use a 32x32 grid, which approximately corresponds with the calorimeter granularity of ATLAS and CMS. Heat maps of the calorimeter energy deposits in $\eta - \phi$ space for both signal prompt muons and background non-prompt muons are shown in Fig. 2.2. The signal calorimeter deposits are uniform and can be attributed to pileup whereas the background deposits appear largely radially symmetric with a dense core from the jet.

We calculate the standard muon isolation observable $I_\mu(R_0)$ for a set of cones with $0.025 \leq R_0 \leq 0.45$ in 18 equally spaced steps.

Crucially, these isolation observables and all other calorimeter observables are calculated directly from the pixels of the muon images, ensuring that they contain a strict subset of the information available. This allows for direct and revealing comparisons of the performance between networks trained with the images and those trained with I_μ . Note that pixelization of the detector may incur some loss of information relative to the underlying segmentation of the calorimeter. However, this work focuses on examining the relative power of different techniques, rather than identifying the best performance under the most realistic scenario.

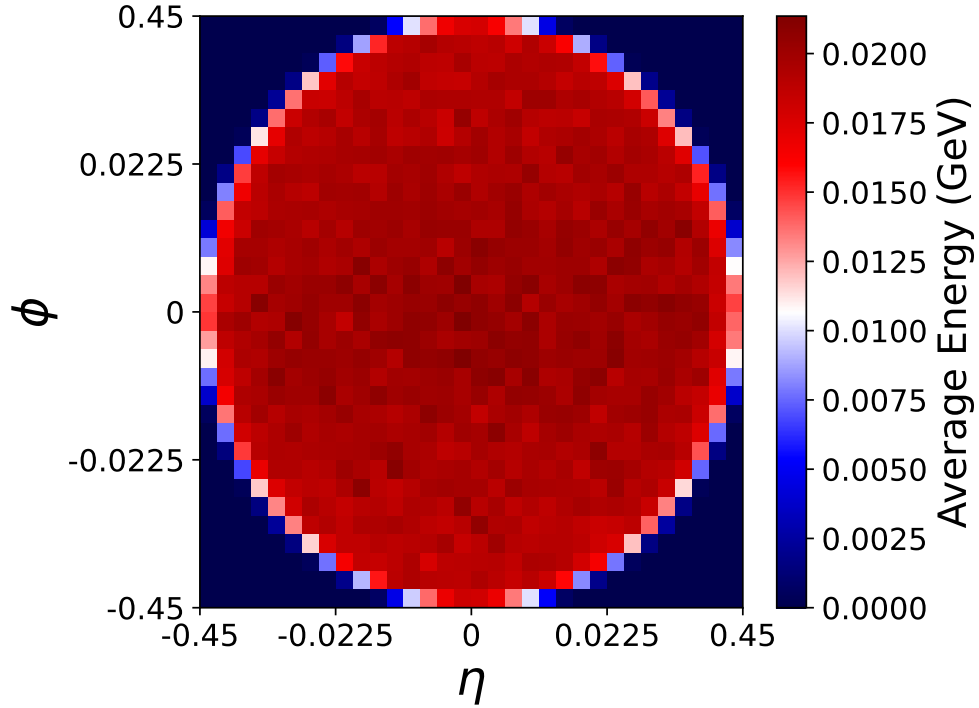


(a)

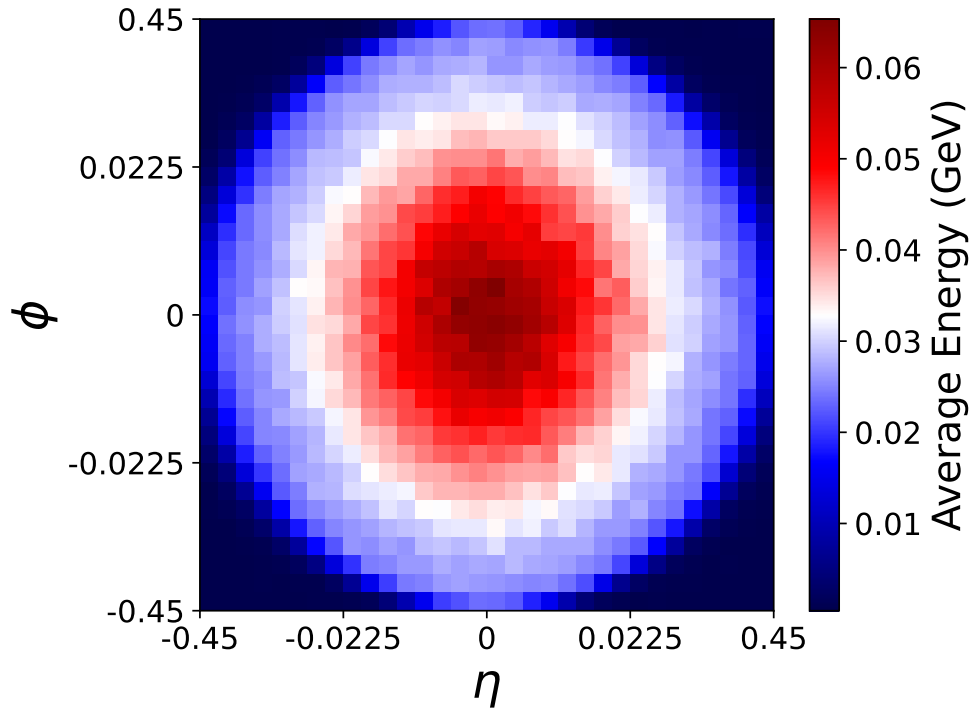


(b)

Figure 2.1: Distributions of muon transverse momentum (top) and pseudorapidity (bottom) for signal and background samples. Afterwards, the distributions are weighted to make both samples uniform.



(a) Mean Prompt Muon



(b) Mean Non-prompt Muon

Figure 2.2: Mean calorimeter images for signal prompt muons (top) and muons produced within heavy-flavor jets (bottom), in the vicinity of reconstructed muons within a cone of $R = 0.45$. The color of each cell represents the sum of the E_T of the calorimeter deposits within the cell.

2.3 Networks and Performance

We apply several strategies to the task of classifying prompt and non-prompt muons, using both low-level calorimeter information and higher-level isolation quantities. We evaluate the performance of each approach by comparing the integral of the ROC (Receiver Operating Characteristic) curve, known as the AUC (Area Under the Curve). The uncertainty for the AUC is calculated by training 100 randomly initialized models with the same hyperparameters on different bootstraps of the data. In this case, we seek to determine the statistical uncertainty due to the stochastic training method, rather than any systematic uncertainty due to the calorimeter resolution.

For the high-level quantities, the standard approach of using a single isolation cone yields an AUC of 0.787 for the optimal cone size, $R_0 = 0.425$ ¹. We hypothesized that additional cones would provide useful information about the radial energy distribution. Including a second cone with a distinct R_0 value as input to a small neural network (see Appendix A) slightly improves performance, with an AUC of 0.793. To estimate the full information available in the cones, we perform a greedy search through all 18 cones; we find that a set of 10 cones² yields another small boost in classification power up to an AUC of 0.803, as shown in Fig. 2.3. Performance was fairly insensitive to the specific choices of cone sizes, and does not grow significantly beyond 10 cones. Feed-forward dense networks are trained to use the information in one or more isolation cones (see the Appendix for details on network architectures and training).

We next examine whether additional information is available by applying strategies which access the calorimeter information at the lowest-level and highest-dimensionality. Convolutional networks (CNN) are applied to the muon images [36, 20, 19]. As an alternative, we apply particle-flow networks (PFN) [62], which are mathematically structured as sums over

¹Similar performance was seen for other cone sizes.

² $R_0 = [0.025, 0.05, 0.075, 0.125, 0.15, 0.225, 0.275, 0.325, 0.425, 0.45]$

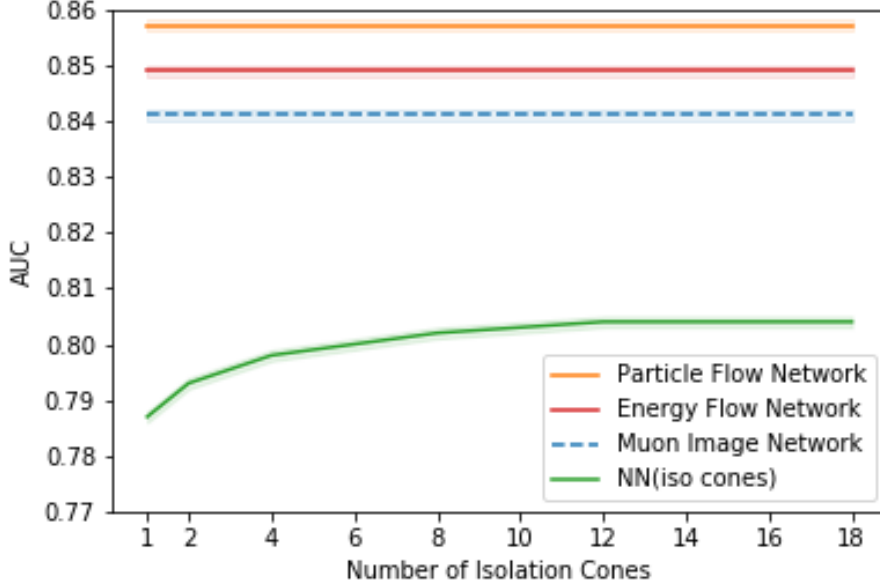


Figure 2.3: Comparison of classification performance using the metric AUC between Particle-Flow networks trained on lists of calorimeter deposits (orange, solid), Energy-Flow networks trained on lists of calorimeter deposits (red, solid), convolutional networks trained on muon images (blue, dashed) and networks which use increasing numbers of isolation cones (green, solid). For each number of cones, the optimal set is chosen.

inputs and thus are invariant to permutations of the inputs.

The muon image CNN achieves a significantly higher performance than the isolation-only networks, with an AUC of 0.841, and the particle flow network reaches 0.857, see Fig. 2.4 and Table 2.1. This immediately suggests that there is significant additional information available to distinguish between the prompt and non-prompt muons beyond what is summarized in the isolation cones. A more restricted version of the PFN, an Energy-Flow Network [62] (EFN), which enforces infra-red and collinear (IRC) safety, achieves nearly the same performance, 0.849. This suggests that most of the additional information beyond the isolation cones is IRC-safe.

These results support the conventional wisdom that a significant fraction of the information relevant for classification is captured by a single, simple cone. However, they also indicate

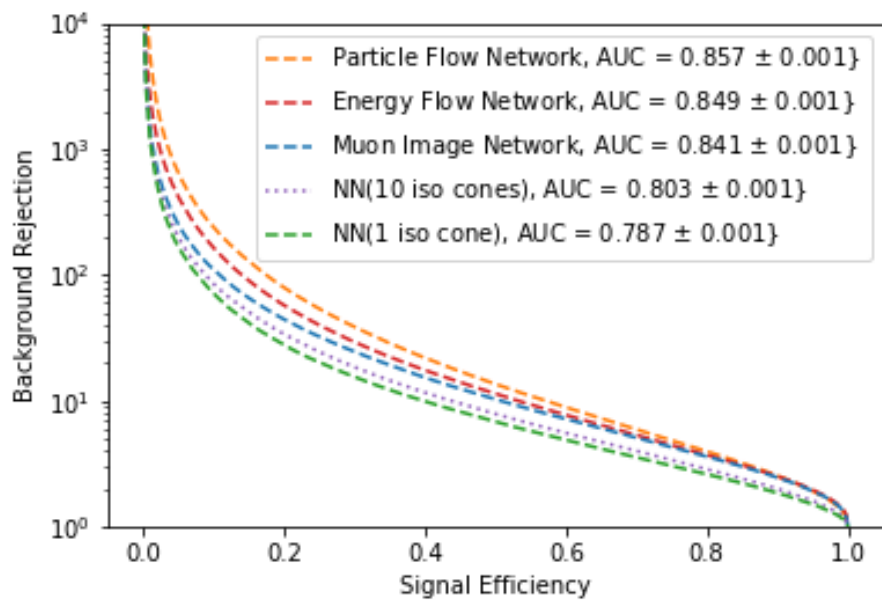


Figure 2.4: Background rejection versus signal efficiency for Particle-Flow networks trained on lists of calorimeter deposits (orange, dashed), Energy-Flow networks trained on lists of calorimeter deposits (red, dashed), convolutional networks trained on muon images (blue, dashed), networks trained on a set of isolation cones (purple, dotted) and the benchmark approach, a single isolation cone approach (green, dashed).

that there is additional information in the radial distribution of energy, which can be captured by using multiple cones. However, even many cones fail to match the performance of the networks which use the calorimeter cell information directly, suggesting that there is additional non-radial information relevant to the classification task not captured in the isolation cones. This is likely due to a difference between the muon axis, the center of the isolation cones, and the jet axis.

2.4 Analysis

The networks which use the calorimeter cells directly have the most powerful performance, but our aim is not simply to optimize classification performance in this particular simulated sample. Instead, we seek to understand the nature of the learned strategy in order to validate it and translate it into simpler, more easily interpretable high-level features which can be studied in other datasets, real or simulated. In addition, this understanding can reveal how well the strategy is likely to generalize to other kinds of jets that are not represented by this background sample, such as charm jets.

The CNN and PFN results indicate that the radially symmetric isolation cones are failing to utilize some information which is relevant to the classification task. In this section, we search for additional high-level observables which capture this information.

2.4.1 Search Strategy

Interpreting the decisions of a deep network with a high-dimensional input vector is notoriously difficult. Instead, we attempt to translate its performance into a smaller set of interpretable observables [47]. This allows us to understand the nature of the information being used as well as to represent it more compactly.

One might imagine exploring a set of physically-motivated quantities, such as the relative p_T between the jet and the muon or the energy-weighted average distance between the jet and calorimeter cells. These particular quantities were considered and found to not contribute significant power in addition to the isolation cones.

Instead, we use a systematic approach and explore a formally complete set of observables. As the background non-prompt muons are due to jet production, we search within a set of observables originally intended for analysis of jets: the Energy Flow Polynomials (EFPs) [61], a formally infinite set of parameterized engineered functions, inspired by previous work on energy correlation functions [65], which sum over the contents of the cells scaled by relative angular distances. An EFP for a jet with M constituents which considers N correlators with angular connections k, l is written as:

$$\text{EFP} = \sum_{i_1=1}^M \dots \sum_{i_N=1}^M z_{i_1}^\kappa \dots z_{i_N}^\kappa \prod_{k,l} \theta_{i_k i_l}^\beta$$

where

$$(z_i)^\kappa = \left(\frac{p_{Ti}}{\sum_j p_{Tj}} \right)^\kappa, \quad (2.1)$$

$$\theta_{ij}^\beta = (\Delta\eta_{ij}^2 + \Delta\phi_{ij}^2)^{\beta/2}. \quad (2.2)$$

Here, p_{Ti} is the transverse momentum of cell i , and $\Delta\eta_{ij}$ ($\Delta\phi_{ij}$) is the pseudorapidity (azimuth) difference between cells i and j . These parametric sums correspond to the set of all isomorphic multigraphs where:

$$\text{each node} \Rightarrow \sum_{i=1}^N z_i, \quad (2.3)$$

$$\text{each } k\text{-fold edge} \Rightarrow (\theta_{ij})^k. \quad (2.4)$$

As the EFPs are normalized, they capture only the relative information about the energy deposition. For this reason, in each network that includes EFP observables, we include as an additional input the sum of p_T over all cells, to indicate the overall scale of the energy deposition.

The original IRC-safe EFPs require $\kappa = 1$. To more broadly explore the space, we consider examples with $\kappa \neq 1$ to explore a broader space of observables³.

In principle, the space spanned by the EFPs is complete, such that any jet observable can be described by one or more EFPs of some degree. One might consider simply searching this space for all possible combinations of EFPs for a set which maximizes performance for this task. Such a search is computationally prohibitive; instead, we follow the black-box guided algorithm of Ref. [47], which iteratively assembles a set of EFPs that mimic the decisions of another guiding network (the PFN in our case) by isolating the portion of the input space where the guiding network disagrees with the isolation network, and finding EFPs which mimic the guiding network’s decisions in that subspace.

Here, the agreement between networks $f(x)$ and $g(x)$ is evaluated over pairs of (x, x') by comparing their relative classification decisions, expressed mathematically as:

$$\text{DO}[f, g](x, x') = \Theta\left((f(x) - f(x'))(g(x) - g(x'))\right), \quad (2.5)$$


and referred to as *decision ordering* (DO). A DO= 0 corresponds to inverted decisions over all input pairs and DO= 1 corresponds to the same decision ordering. As prescribed in Ref. [47], we scan the space of EFPs to find the observable that has the highest average decision ordering (ADO) with the guiding network when averaged over disordered pairs. The selected EFP is then incorporated into the new network of HL features, HLN_{n+1} , and

³Also, note that $\kappa > 0$ generically corresponds to IR-safe but C-unsafe observables. For $\kappa < 0$, empty cells are omitted from the sum.

the process is repeated until the ADO plateaus.


2.4.2 IRC Safe Observables

As the elements of the EFP space are not orthogonal, there are potentially many combinations of EFP observables which capture the relevant information. As simpler EFPs may be more conducive to theoretical interpretation, we begin our search in a restricted subset of the EFP space. Specifically, we consider those which are IRC safe ($\kappa = 1$), have a simple angular weighting ($\beta \in [1, 2]$), and $n \leq 3$ fewer nodes with at most three edges between nodes. We also include $\sum p_T$, where the summation is over all calorimeter cells in the image, to set the scale accompanying the normalized EFPs. The first EFP observable identified is a simple three-point correlator:




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


which, when combined with the isolation cones and $\sum p_T$, yields an AUC of 0.838 and an ADO with the CNN of 0.891, a significant boost relative to just using the radial information of the isolation cones. The subsequent scans produce variants of this observable :



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



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



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

with 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. Together with the isolation cones, these observables reach an AUC of 0.842 and an ADO with the PFN of 0.888, see Table 2.1.

This set of observables partially closes the performance gap with the best calorimeter cell networks, indicating that angular information is relevant to the muon isolation classification task, but fails to fully match its performance. Distributions of these EFPs for signal and background are shown in Fig. 2.5. 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 we have considered.

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

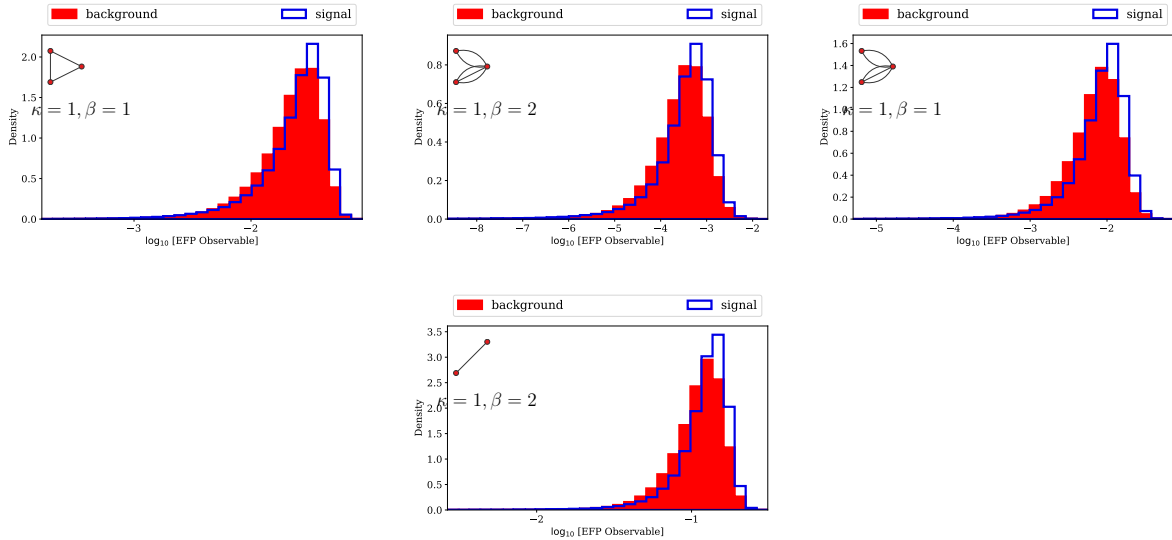


Figure 2.5: Distributions of the \log_{10} of the selected IRC-safe EFPs as chosen by the black-box guided strategy, for prompt (signal) muons and non-prompt (background) muons.

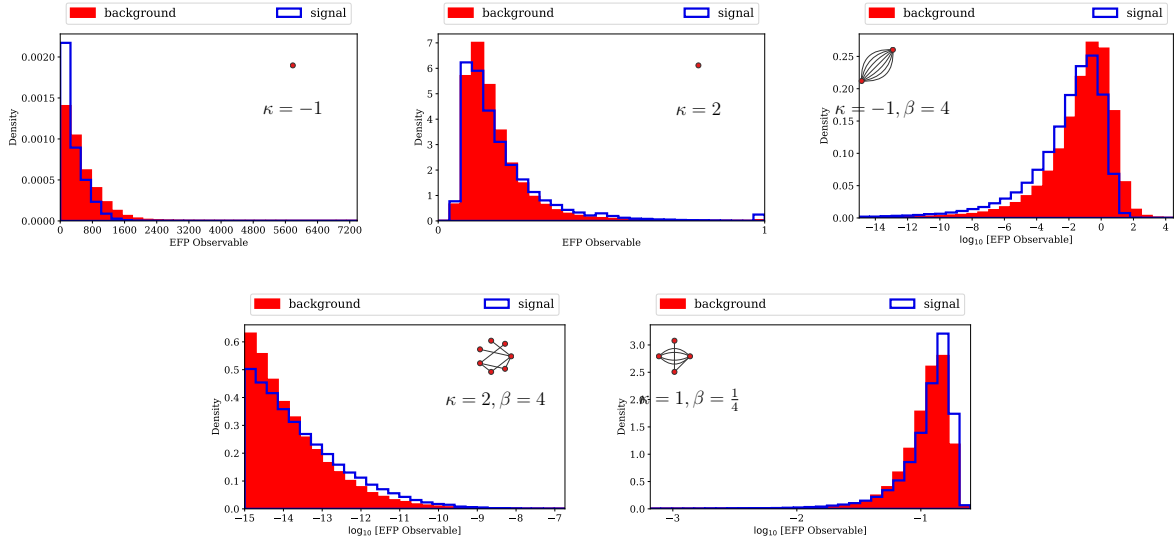


Figure 2.6: Distributions of the \log_{10} of the selected EFPs as chosen by the black-box guided strategy, for prompt (signal) muons and non-prompt (background) muons.

2.4.3 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, we expand the search space to include observables which are not IRC safe ($\kappa \in [-1, 0, \frac{1}{4}, \frac{1}{2}, 1, 2]$), with alternative angular powers ($\beta \in [\frac{1}{4}, \frac{1}{2}, 1, 2, 3, 4]$) and with up to $n = 7$ nodes and $d = 7$ edges.

A scan of these observables finds a set of 5 which, when combined with the isolation cones and $\sum p_T$ reach an AUC of 0.857. Figure 2.6 shows the EFP graphs as well as their distributions for prompt and non-prompt muons. They include single point-graphs, with no angular powers, as well as a two-point correlators with large angular power sensitive to high-angle effects, and more complex graphs with multiple nodes. We note that due to the overlapping nature of the large space of EFPs, there are several sets of EFPs which achieve similar performance. Again, a similar scan guided by the CNN rather than the PFN yields very similar results.

Table 2.1: 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 inputs to and parameters of each network.

Method	N_{inputs}	AUC	ADO [PFN]	N_{Params}
Single Iso Cone	1	0.787	0.860	40k
10 Iso	10	0.803	0.877	41k
10 Iso, $\sum p_T$	11	0.807	0.884	42k
10 Iso, $\sum p_T$, 4 simple EFPs	15	0.842	0.888	42k
10 Iso, $\sum p_T$, 5 EFPs	16	0.857	0.900	43k
Calo image CNN	1024	0.841	0.950	167k
Calo cell Energy-Flow Net	102	0.849	0.951	453k
Calo cell Particle-Flow Net	102	0.857	1	453k

2.5 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 2.1. 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. The multi-point correlators may be sensitive to the width of the jet, due to the momentum of the constituents relative to the jet axis, as a result of b - and c -quark decays.

A comparison of the network complexity for the various approaches is shown in Tab. 2.1. 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.

2.6 Conclusions

We have applied deep networks to low-level calorimeter deposits surrounding prompt and non-prompt muons in order to estimate the amount of classification power available and to probe whether the standard methods are fully capturing the relevant information.

The performance of the calorimeter cell networks significantly exceeds the benchmark approach, a single isolation cone. The use of several isolation cones provides some improvement, suggesting that there is additional useful information in the full radial energy distribution. However, a substantial gap remains, hinting that there is non-radial structure in the calorimeter cells which provides useful information for classification. We map the strategy of the calorimeter cell networks into a set of energy flow polynomials, finding four IRC-safe, simple three-point correlators which capture a significant amount of the missing information. As they 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. Any boost in the efficiency to identify prompt muons is extremely valuable to searches at the LHC, especially those with multiple leptons, where event-level efficiencies depend sensitively on object-level efficiencies.

Additional, more complex EFPs provide a further modest boost in performance, closing the gap with the PFN. The strong performance of the IRC-safe EFN suggests that most of the additional information beyond the isolation cones is IRC-safe.

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 represents an opportunity to gather additional power in the battle to suppress lepton backgrounds. Rather than employing black-box deep networks directly, we have demonstrated the power of using them to identify the relevant observables from a large list of physically interpretable options. This allows the physicist to understand the nature of the information being used and to assess

its systematic uncertainty. Here we have focused on two-dimensional projections of the calorimeter response, but longitudinal information expressed in three dimensions may offer additional power in future work. While these studies were performed with simulated samples, similar studies can be performed using unsupervised methods [44, 69] on samples of collider data, which we leave to future studies.

Chapter 3

Learning to Isolate Muons in Data

This chapter is heavily based on work previously published in collaboration with Benjamin Nachman and Daniel Whiteson[93].

We use unlabeled collision data and weakly-supervised learning to train models which can distinguish prompt muons from non-prompt muons using patterns of low-level particle activity in the vicinity of the muon, and interpret the models in the space of energy flow polynomials. Particle activity associated with muons is a valuable tool for identifying prompt muons, those due to heavy boson decay, from muons produced in the decay of heavy flavor jets. The high-dimensional information is typically reduced to a single scalar quantity, isolation, but previous work in simulated samples suggests that valuable discriminating information is lost in this reduction. We extend these studies in LHC collisions recorded by the CMS experiment, where true class labels are not available, requiring the use of the invariant mass spectrum to obtain macroscopic sample information. This allows us to employ Classification Without Labels (CWoLa), a weakly supervised learning technique, to train models. Our results confirm that isolation does not describe events as well as the full low-level calorimeter information, and we are able to identify single energy flow polynomials capable of closing

the performance gap. These polynomials are not the same ones derived from simulation, highlighting the importance of training directly on data.

3.1 Introduction

Data collected in hadronic collisions offer a significant opportunity to precisely test the Standard Model (SM) and to search for physics beyond the SM (BSM). The identification of muons resulting from electroweak boson decays (called ‘prompt’) is a crucial part of many such studies, as muons are typically well measured and have low rates of background. An important source of background for these events comes from muons produced within jets from decays in flight. This ‘non-prompt’ background is largest at the lower end of the muon transverse momentum spectrum, which has become important in searches for supersymmetry [4, 78, 59, 2, 89, 8] as well as for low-mass resonances [56, 3, 11, 88].

Prompt muons tend to have less nearby detector activity as compared to muons from jets, which are found near hadrons from the rest of the jet. The concept of *isolation* is therefore important to much of the work involving the discrimination of prompt muons from the non-prompt backgrounds. A complete description of the isolation requires capturing the high-dimensional data in the vicinity of the muon. In practice, high-dimensional data are challenging to analyze and isolation is typically reduced to a scalar quantity [82, 83, 9]. However, in the reduction from a high-dimensional (low-level) representation of the data to a lower-dimensional (high-level) one, information can be lost.

Deep learning with low-level inputs has been demonstrated to exceed the performance of engineered high-level observables on a number of tasks in high energy physics, starting with Refs. [23, 43] and now including many studies [49]. In the context of prompt muon identification, deep neural networks were able to outperform classical isolation definitions

using simulated data – by as much as 50% in non-prompt background rejection at a prompt muon efficiency of 50% [38]. This was achieved by processing all of the calorimeter cells¹ in the vicinity of the muon, corresponding to roughly 1000 dimensions per event. Significant suppression of non-prompt backgrounds with a deep learning approach has the potential to improve the precision and sensitivity of many measurements and searches involving muons at the Large Hadron Collider (LHC).

However, previous studies were based on simulations, with relatively simple detector effects. Hadronic final states are complex and difficult to model, so it is reasonable to be concerned that the performance of a deep learning-based isolation strategy trained on simulated events may depend on details of the simulation which are not faithful reproductions of collider data. Scale factors derived using standard tag-and-probe methods [5, 84] may correct the efficiency, but the performance in data would be suboptimal [31]. Achieving optimal performance in data requires training with data. The limitation is that data are not labeled as prompt or not-prompt, so the *supervised* machine learning strategies used in previous studies and which require such labels cannot be applied to data.

We propose to overcome this limitation with *weakly supervised* learning. In contrast to supervised learning, where every event is labeled with certainty as prompt or non prompt, weakly supervised learning is trained with noisy labels, which describe the overall composition of the sample but not individual events. Specifically, we use the Classification Without Labels (CWoLa) [69] approach to weak supervision where two samples of training events are prepared. One sample is dominated by prompt muons, and receives the noisy label of ‘signal’ (and will be called ‘prompt abundant’); the second sample, while still mostly containing prompt muons, has a relatively higher fraction of non-prompt muons and receives the noisy label of ‘background’ (and will be called ‘prompt moderate’). Under mild assumptions, training a standard classifier with these noisy labels converges to the same classifier

¹The previous work mentioned here only used calorimeter information, though this study considers both calorimeter and track information.

found in a supervised setting. While weak supervision has been used previously for data analysis [85, 70, 41, 40, 7], these studies only used 2-18 inputs. Our goal is to approach the muon isolation problem with weak supervision directly on low-level, high-dimensional ($\mathcal{O}(100)$) inputs. While the inputs are high dimensional enough to hold a large number of detected objects, this is only necessary for a small number of events, as on average the inputs have $\mathcal{O}(10)$ non-zero entries.

Even if proven effective in data, deep networks operating on low-level observables can be opaque. To improve the interpretability and compactness of the network, we follow Ref. [38], bridging the performance gap between the low-level observables and classical isolation variables through a small set of additional high-level observables identified by the decisions of a network operating at the low-level. We search for new high-level observables among the Energy Flow Polynomials (EFP) [61], a set of relatively simple combinations of energies and angles of reconstructed objects within the isolation cone. EFP observables are identified automatically using the Average Decision Ordering (ADO) method [48], which uses the decisions of the low-level network as a guide. While still complex, the resulting EFP is more physically interpretable than the original deep neural network. Interestingly, the first EFP selected through this process was not identified in the previous study as a top candidate for closing the corresponding gap in simulation [38]. This is one more reason why it is essential here to train directly on data.

This paper is organized as follows. Section 3.2 introduces the dataset, which is from the CMS experiment [33, 1]. Then, Sec. 4.3 describes the machine learning strategy. Numerical results are presented in Sec. 4.4. The paper ends with conclusions and outlook in Sec. 4.5.

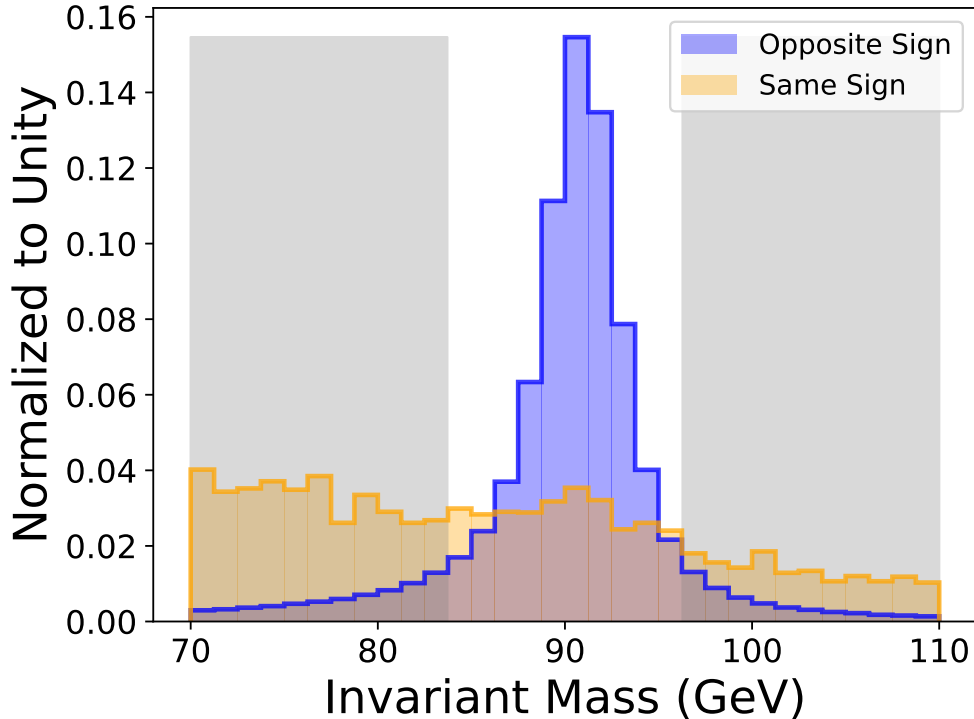


Figure 3.1: Histogram of the dimuon invariant mass near the Z boson peak, for events in data with identical (yellow) or opposite (blue) electric charges. The unshaded area indicates the region for the oppositely-charge pairs which comprises our “prompt muon abundant” sample. The grey, shaded region for the oppositely-charge pairs, as well as the entire region for identically-charged pairs, comprise our “prompt muon moderate” sample.

3.2 Dataset

Proton-proton collisions at $\sqrt{s} = 8$ TeV were recorded in 2012 and curated by the CMS Collaboration and made available through the CERN Open Data Portal [1]. The number of collisions corresponds to 19.5 fb^{-1} . Reconstruction was performed with the Particle Flow (PF) algorithm [83], which integrates calorimeter and tracker information to approximate individual particle four-vectors. The PF algorithm also assigns a particle identification (PID) from one of the following types: muon, charged hadron, neutral hadron, photon, or pileup. For the charged PF objects, the sign of the charge is reconstructed. PF object momenta are represented by their transverse momentum (p_T), pseudorapidity (η), and azimuthal angle (ϕ).

We select events with exactly two muons, both with $p_T \geq 25$ GeV, $|\eta| \leq 2.1$, and with a dimuon invariant mass between 70 and 110 GeV to accommodate the Z boson mass of 90 GeV [94]. Events are separated into two samples which have different mixtures of prompt and non-prompt muon events, as is required by the CWoLa method. One sample, with a higher fraction of non-prompt muons, consists of all events in which the muons have identical electric charge, as well as events with muon pairs of opposite electric charge but reconstructed invariant mass far from the Z boson invariant mass, below 84 GeV or above 96 GeV. This sample is referred to as the “prompt muon moderate sample.” The remaining events, which are almost entirely prompt muons, form the complementary sample and are referred to as the “prompt muon abundant sample.” These regions are illustrated in Fig 3.1. The opposite sign sample is almost entirely from Z boson decays and so is peaked at the Z boson mass. The same sign sample is mostly from decays in flight and has a nearly smooth and steeply falling spectrum.

In order to ensure that the two samples have similar kinematic distributions, event weights are computed so that the muon p_T and η spectra are the same between the prompt-enriched and non-prompt-enriched samples. The unbinned likelihood ratio is estimated using a two-dimensional Kernel Density Estimator with Gaussian kernels. The pre-weighted spectra are displayed in Fig. 3.2. The p_T spectrum is peaked near $m_Z/2$ and the sharp features in the muon histogram are due to detector acceptance effects. We additionally validate the core assumption of CWoLa (see Sec. 4.3) – that the (non)prompt muons look the same in both samples – using samples of simulated muons; see Appendix B.

Once events are selected, they are formatted to be used as inputs to the neural networks. The low-level inputs are comprised of the p_T , η , ϕ , and PID for each constituent within a 0.45 radius around a given muon. We additionally preprocess the low-level input by centering on the muon and dividing the momenta by the muon transverse momentum. A visualization of the momentum in the vicinity of the muon, not including the muon itself, for both samples

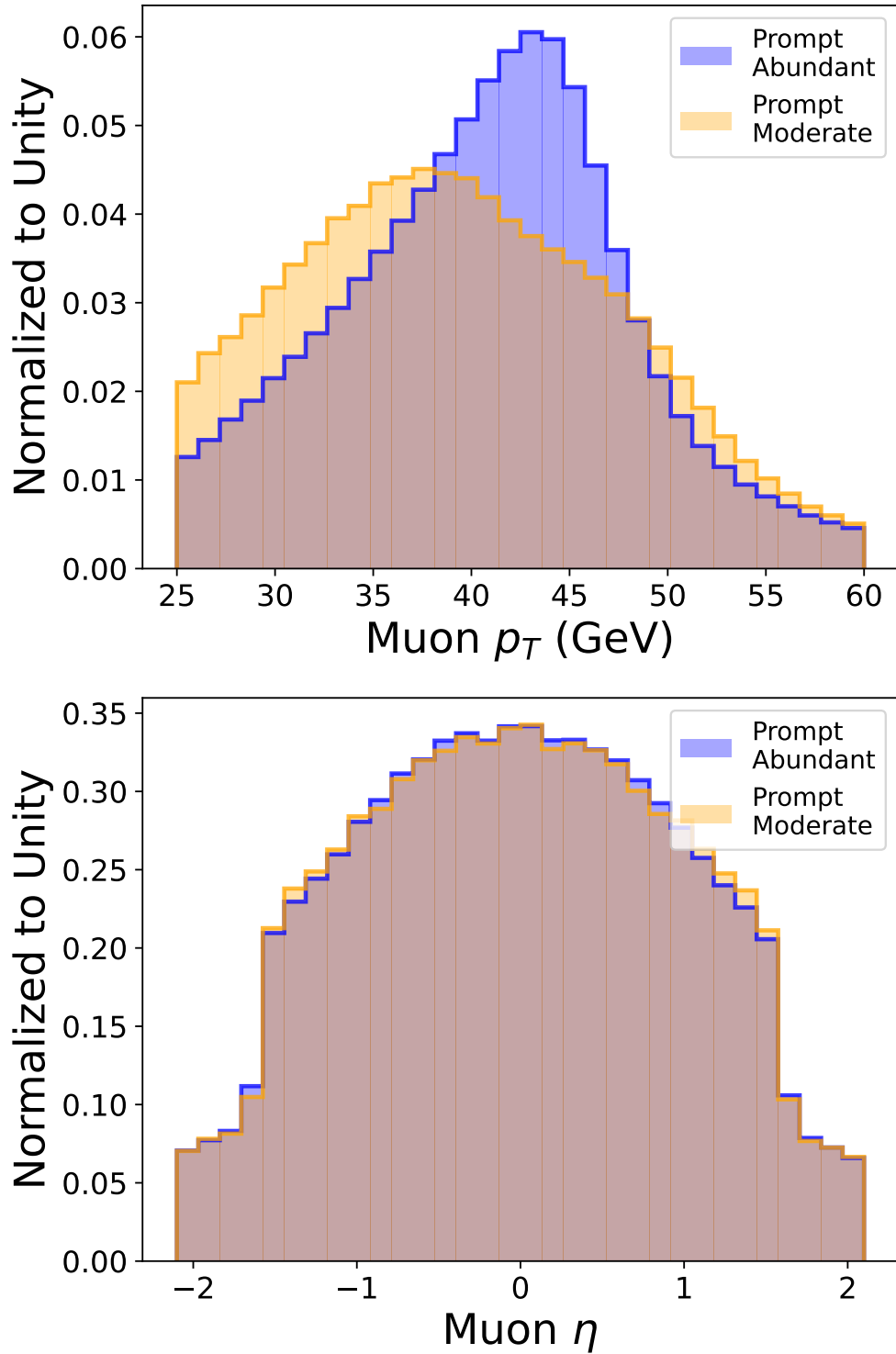


Figure 3.2: Histograms of muon p_T and pseudorapidity η in the two samples with varying fractions of prompt muons, as defined in text and Fig. 3.1.

is shown in Fig. 3.3. We see that the sample means per pixel have distinct distributions, with the more prompt sample being more uniform.

Traditional, high-level scalar observables are calculated from the low-level data. These observables include the summed p_T of non-muon objects in an event, isolation, and EFP observables. We calculate isolation as defined in Eq. 3.1, where h_{\pm} and h_0 denote charged and neutral hadrons, respectively. This definition quantifies the activity around a muon within a given radius strictly in terms of Particle Flow objects and treats the objects differently according to their Particle Flow ID. The expression is composed of terms which sum over the transverse momenta of the non-muon Particle Flow candidates within the chosen radius, and the result is normalized by the muon momentum. Pileup is mitigated by subtracting half of its sum from the neutral hadron and photon sums, and clamping the result of this subtraction at 0. Distributions of the isolation for two choices of cone radius are shown in Fig 3.4. The larger of the two choices of radius tends to yield larger isolation values, as one might expect.

$$I_{\mu}(R_0) = \left[\sum_{i,R < R_0}^{N_{h_{\pm}}} p_{T,h_{\pm}}^i + \max\left(0, \sum_{i,R < R_0}^{N_{h_0}} p_{T,h_0}^i + \sum_{i,R < R_0}^{N_{\gamma}} p_{T,\gamma}^i - \frac{1}{2} \sum_{i,R < R_0}^{N_{\text{pileup}}} p_{T,\text{pileup}}^i\right) \right] / p_{T,\text{muon}} \quad (3.1)$$

We calculate isolation quantities for a set of radii from 0.025 - 0.45 in steps of 0.025. CMS has previously studied isolation at radius of 0.3 [83], which is included in our generated set.

While in principle the demonstration of weak supervision as a technique for learning to improve muon isolation beyond cone-based quantities could use simulation instead of data, we have chosen to use collider data for a number of reasons. First, realistic simulation of muon isolation is very challenging, for both the prompt and non-prompt categories; see App. B. Second, a demonstration in data can confirm (or refute) the results of earlier studies

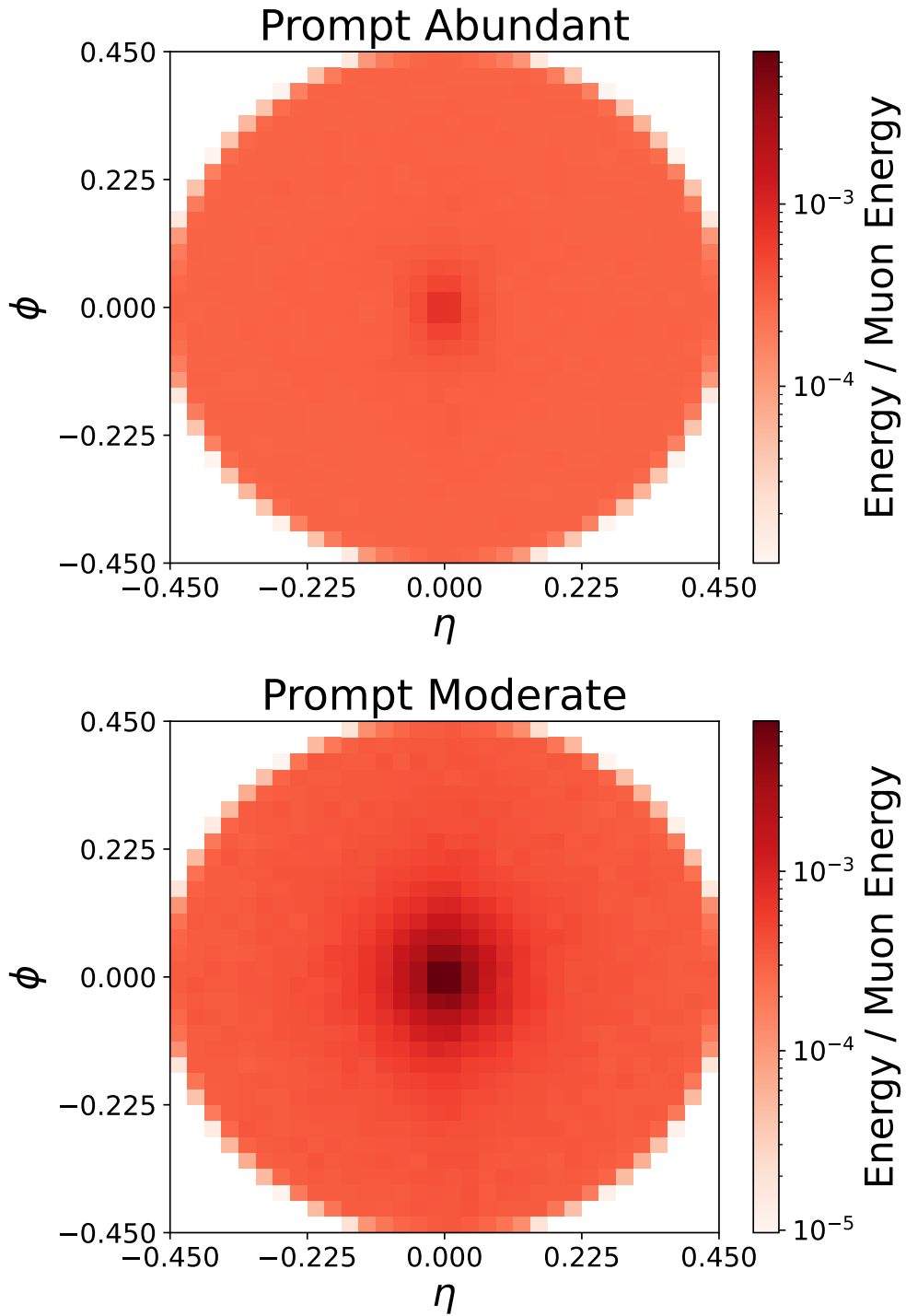


Figure 3.3: The average image of hadronic activity in the vicinity of an identified muon, in angular coordinates of azimuthal angle ϕ and pseudorapidity η , for our two training samples, one which is dominated by prompt muons (top) and a second which has a more moderate mixture of prompt and non-prompt muons (bottom). The muon itself is excluded from these visualizations, but the energies are normalized by that of the muon.

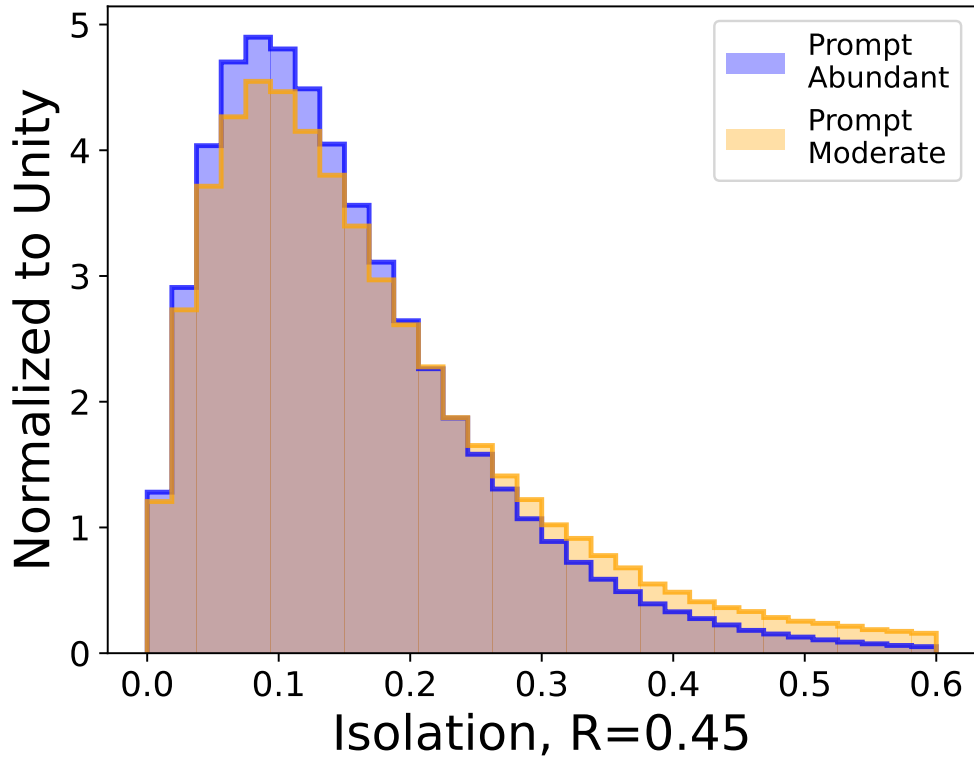
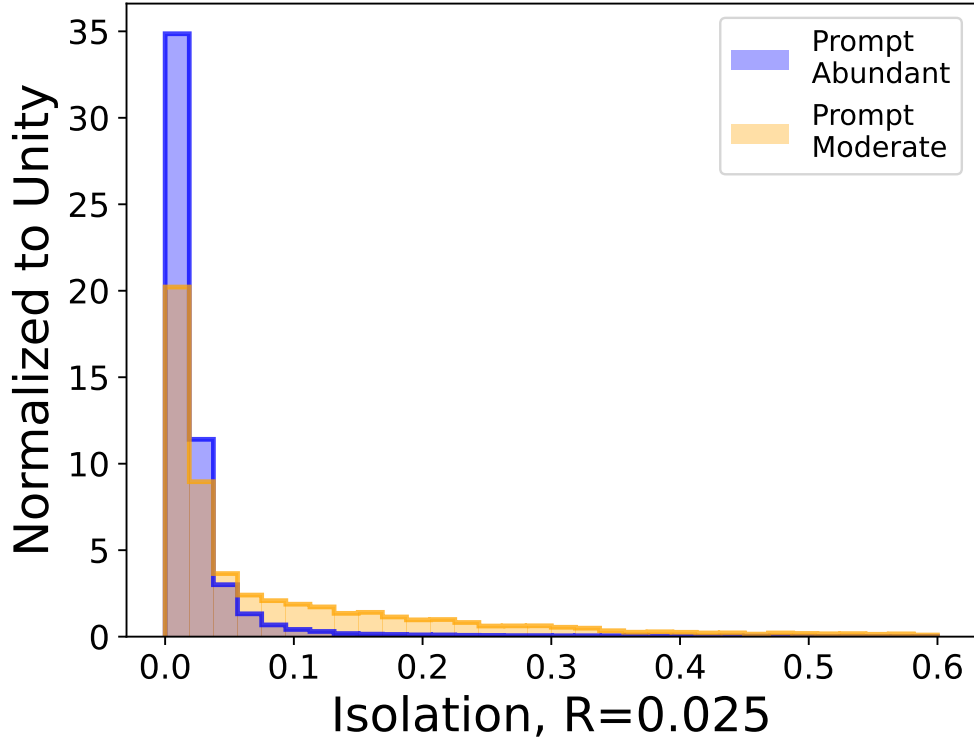


Figure 3.4: Histograms of the muon isolation (defined in Eq. 3.1) for each of our training samples, one of which is dominated by prompt muons, for two choices of isolation cone radius parameter $R_0 = 0.025$ (top) and $R_0 = 0.45$ (bottom).

in simulation, which showed a significant gap between the power of isolation cones and full use of the lower-level data. If such a gap exists in collider data, it would indicate that additional information is available in nature; the interpretation of that gap in terms of EFP observables will provide clues as to the physical processes involved, and the size of the gap can motivate a further study in a complete experimental context. For this reason, we also do not estimate systematic uncertainties, which would be required before application to searches and measurements. As a data-driven method, there are no simulation-based uncertainties, but there would be method closure uncertainties related to the underlying assumptions of CWoLa and sPlots.

3.3 Methods

Classification Without Labels (CWoLa) defines a weakly supervised setting which relies on the principle that given two classes, an optimal classifier may be obtained by training to discriminate between two samples composed of different mixtures of the classes, rather than training directly on two pure class samples. This technique only requires that the two samples have different class mixtures, and these mixtures do not need to be known in order for training to proceed. The essential assumption is that class fraction is the only feature that determines the different properties of the two samples. This means that the spectrum of radiation around the muon for prompt leptons is identical for the prompt muon abundant and the prompt muon moderate samples. Similarly, the probability density for hadrons around the muon for non-prompt leptons should be the same within the prompt muon abundant and the prompt muon moderate samples. We expect this to be the case here, since the invariant mass of the muons and their relative electric charges should statistically independent from the radiation pattern around the muons given the prompt status. This expectation is validated in simulation in App. B.

While CWoLa does not need class labels to derive a classifier, some class information is required to determine the performance of the method. The only information needed is the proportion of prompt muons in each sample; from this information, it is possible to characterize the full tradeoff between signal efficiency and background rejection. The prompt-muon fraction is measured directly from the data in each sample by modeling the invariant mass distribution as a mixture model with two components: one peaking component of Z bosons which decay to two prompt muons, and a second, non-peaking component. The invariant mass spectrum is fit using a Voigt profile and an exponential function for the respective components. Fitting is done with Scipy v1.7.3 [91] and visually demonstrated in Fig 3.5, where the fit is applied to the full dataset, finding an overall prompt fraction of $95.6 \pm 0.6\%$, where the error bar corresponds to 1σ statistical. In the “prompt muon abundant” sample, the prompt fraction is measured to be 98.9%; in the “prompt muon moderate” sample, the prompt fraction is measured to be 56.0%. This is the first application of weak supervision in particle physics where the relative proportions have also been extracted directly from data.

Characterizing the network performance is non-trivial without pure samples. To measure the efficiency of a varying network threshold in the prompt and non-prompt samples, one could fit the distribution of the invariant mass of events surpassing each threshold. Measurement of the efficiencies of each class allows calculation of performance metrics, such as the standard Receiver Operating Characteristic (ROC) and its associated statistics. However, fits are expensive and stochastic. Fitting the mass spectrum for each threshold output can be avoided using the sPlots technique [75], which can decompose the prompt and non-prompt contributions to distributions of the network output given weights from the single invariant mass fit into the full sample. sPlots assumes that the variable being weighted is statistically independent of the invariant mass, within the individual classes. This is approximately true for our discriminating variables, such as model outputs, and so the method can be applied. Once the variable has been separated by the components, the resulting histograms may be

integrated to calculate true and false positive rates, and construct a ROC curve. Performance is evaluated through the Area Under the Curve (AUC) and the signal efficiency at 50% background efficiency. While we do not perform a full determination of the uncertainty, we do consider statistical sources of uncertainty from the training and from the fit². While not an uncertainty per se [71], the statistical variation from the finite size of the training dataset³ gives a sense for the stability and optimality of the result. This effect is estimated using bootstrapping [45] with 100 event ensembles with a new classifier trained per ensemble. Additionally, we propagate the statistical uncertainty from the fit in each ensemble by sampling 100 times from the fitted parameter covariance matrix. Metrics are recomputed and averaged across each ensemble, and we report the 1σ confidence intervals according to the resulting set of values.

We consider two types of neural networks: high-level networks with an increasing list of engineered observables (such as isolation) and low-level networks that process the full muon image. For the high-level networks, one of our goals is to determine the minimal set of isolation observables that will saturate the performance. To do this, we start by training a network using the single isolation cone corresponding to the largest radius in our set and subsequently train networks with incrementally smaller cones included as inputs. The summed event p_T is included as an input in all of these sets, in order to be sensitive to overall normalization effects.

The low-level networks take the full, high-dimensional representations of the events as inputs. We use the deep sets architecture [96] implemented as Particle Flow Networks (PFNs) [62] to process these data. This architecture was chosen because the inputs are a permutation-invariant, variable-length set of four-vectors and so a point-cloud model is the natural choice for processing them. Deep sets models are composed of two fully connected networks. The

²While these are the only sources of uncertainty quantified in Table 3.1, other sources are present, such as a bias due to imperfect description of the mass distribution by the fit function.

³The random initialization of the network is also folded into this estimation.

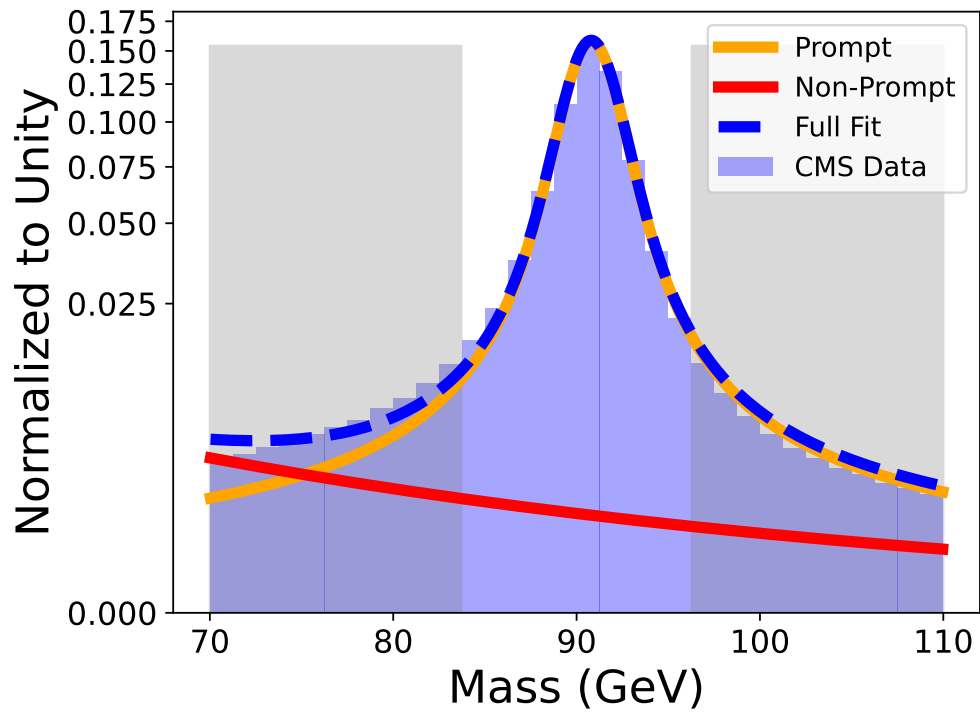


Figure 3.5: A visualization of the masses overlaid with the fit and its prompt / non-prompt components. The shaded regions indicate events which are included in the relatively less prompt sample. Here we fit the full CMS sample used in the study, finding that it is $95.6 \pm 0.6\%$ prompt overall.

first network embeds each particle flow object (represented by $(p_{\text{T}}, \eta, \phi, \text{PID})$) into a latent space. The second network processes the sum of these latent space vectors across all inputs. The sum operation is permutation invariant and can readily process variable-length inputs.

Additionally, we strive to close the gap in performance between low- and high-level networks using relatively simple variables. Energy Flow Polynomials (EFPs) [61] serve as a set of potential variables for this purpose. EFPs are a set of parameterized functions which sum over objects within an event, where each term is weighted using the angular relations between these objects. EFPs can be represented using graphs, where

$$\text{each node} \Rightarrow \sum_{i=1}^N z_i, \quad (3.2)$$

$$\text{each } k\text{-fold edge} \Rightarrow (\theta_{ij})^k. \quad (3.3)$$

$$(z_i)^\kappa = \left(\frac{p_{\text{T}i}}{\sum_j p_{\text{T}j}} \right)^\kappa, \quad (3.4)$$

$$\theta_{ij}^\beta = (\Delta\eta_{ij}^2 + \Delta\phi_{ij}^2)^{\beta/2}. \quad (3.5)$$

When $\kappa = 1$ the EFPs form a basis for Infrared and Collinear (IRC)-safe observables [61]. We compute a set of EFPs which contains IRC-safe, as well as unsafe, information, using the same parameterizations as in Ref. [38]: $\kappa \in [-1, 0, \frac{1}{4}, \frac{1}{2}, 1, 2]$ and $\beta \in [\frac{1}{4}, \frac{1}{2}, 1, 2, 3, 4]$, for graphs with up to $n = 7$ nodes and up to $d = 7$ edges.

We use the Average Decision Ordering (ADO) [48] metric to determine which EFPs from this generated set might bridge the performance gap to the PFN. ADO compares two classifiers on signal and background input pairs, measuring how often the classifiers rank the inputs in the same way. This is quantified with a Heaviside step function on many different pairs, and the results are averaged to obtain the ADO. The ADO can be interpreted as the probability

that a given pair will be ordered in the same way by the two classifiers. This is intuitively similar to the AUC metric, which measures the probability that a given signal example will be ranked higher than a given background example. While AUC can be seen as comparing a classifier to the truth, the ADO compares two classifiers to one another without regard for correct ordering. To avoid training a large set of new high-level networks, one for each EFP being considered as an additional observable, we follow the strategy of Ref. [48] and search for EFPs which have a high ADO with our PFN for the subset of events where the PFN and the high-level network disagree. In general, this process can be iterated, selecting new observables until the ADO no longer improves.

3.4 Results

The performance of each network is measured through ROC AUC as well as the signal efficiency at a fixed background efficiency of 50%. Fig. 3.6 illustrates the effects of including additional isolation cones as network input features. Adding cones tends to increase performance up until nine cones are used, after which there is no clear further gain in AUC. There is a significant performance gap between the network which uses nine cones and the PFN, which respectively yield AUCs of $0.848(1)^4$ and $0.874(1)$, as well as signal efficiencies of $0.939(1)$ and $0.957(1)$. This suggests that isolation cones alone do not capture all discriminating information available in the low-level data. This is consistent with previous results shown on simulation [38], and it is notable that it holds for real collider data.

We use the ADO metric to search among the EFP observables for ways to close the gap with the PFN performance. Note that the EFPs lack the built-in radial symmetry of the isolation cones, and so may contain additional useful information. The networks using EFP features are also provided the nine largest isolation cones and the summed event p_T . Remarkably,

⁴The reported error values should be understood as rounded to 1×10^{-3} from values calculated to be $\lesssim 1 \times 10^{-3}$.

the ADO search method is able to identify a *single* IRC-safe EFP which obtains an AUC of 0.871(1) and signal efficiency of 0.953(1), almost fully closing the gap in AUC to the PFN from 0.026 to 0.003. The graph representation of this EFP, as well as class distributions separated through the sPlots technique, are illustrated in Fig. 3.7. This EFP corresponds to parameters $\kappa = 1$ and $\beta = 0.25$, and the full expression is provided in Eq. 3.6.

$$\sum_{a,b,c,d=1}^N z_a z_b z_c z_d (\theta_{ab} \theta_{ac} \theta_{bd} \theta_{cd}^4)^{1/4} \quad (3.6)$$

An additional scan is done over the quadratic EFPs included in our full set of calculated EFPs, as these are simple in structure and are therefore more interpretable. This identifies another single EFP with $\kappa = 1$ and $\beta = 0.25$ which yields performance close to that of the one identified by the first ADO search, at an AUC of 0.870(1) and signal efficiency of 0.956(1). We further check the performance of sets of EFPs identified as useful by previous work done on simulation [38], which selected an IRC-safe set of EFPs, as well as a set not restricted to be safe. The IRC-safe set yields an AUC of 0.868(1) with a signal efficiency of 0.949(1), while the unsafe set yields an AUC of 0.865(1) with a signal efficiency of 0.954(1). While these sets identified in simulation close much of the performance gap, they require more features and are outperformed by the EFPs identified directly on the CMS data, underscoring the importance of training in data.

A full summary of performance across the methods used is presented in Table 3.1, as well as depicted in Fig. 3.8. Our results indicate that we are able to construct a minimal set of high-level observables which perform comparably to the low-level inputs, allowing for the use of more physically intuitive features and less complex networks without making concessions regarding performance.

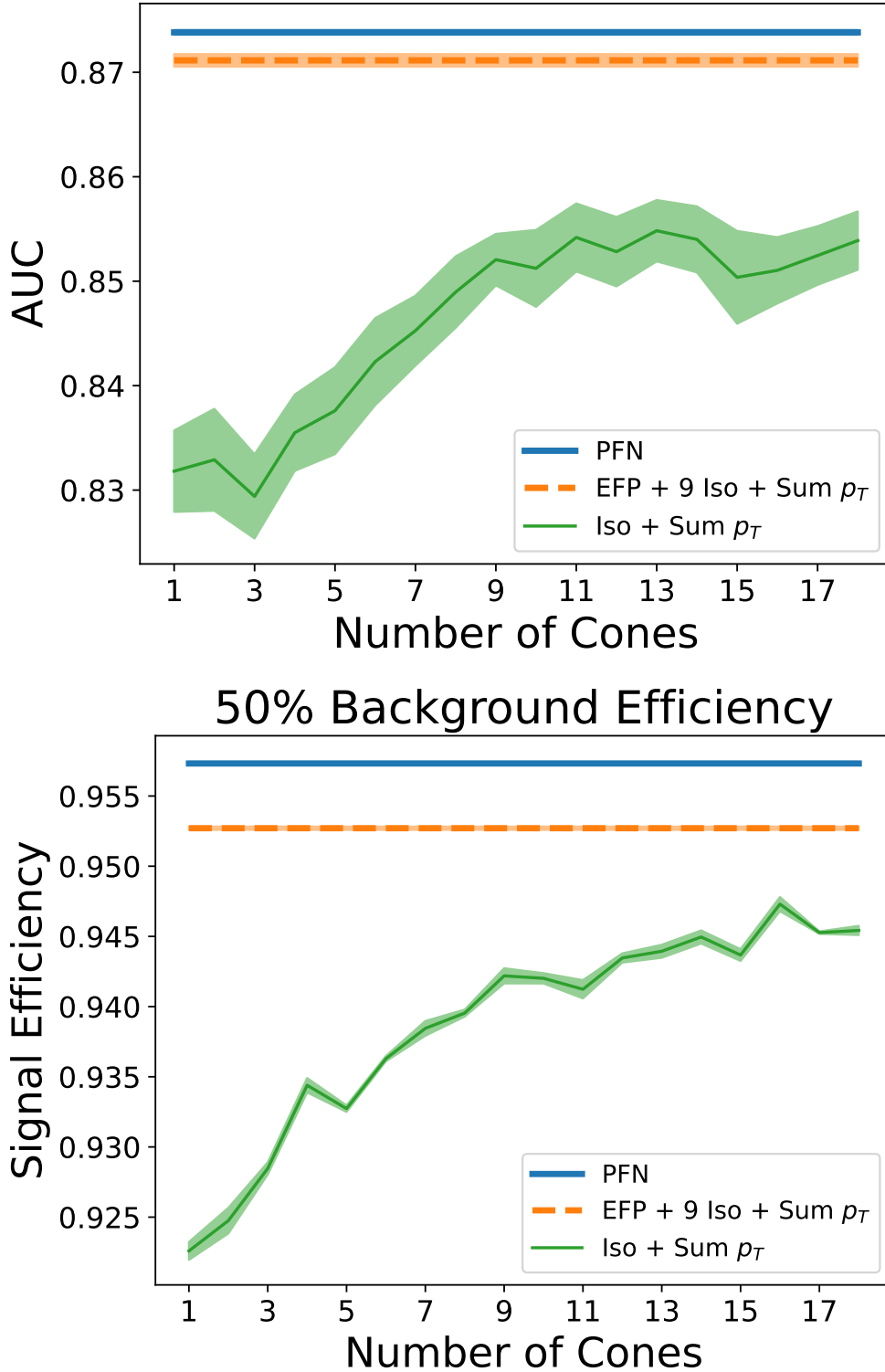


Figure 3.6: Isolation network performance shown as a function of number of input cones. Performance of the PFN and best performing high-level network are shown as benchmarks. ROC AUC is shown for each model (top) as well as the signal efficiency at a fixed background efficiency (bottom).

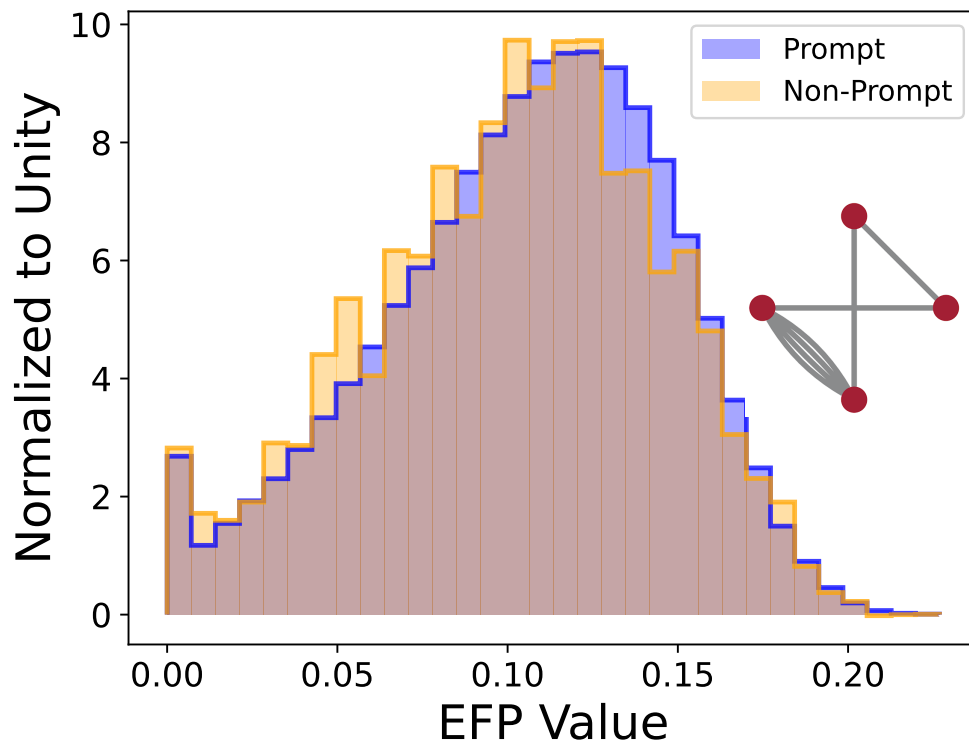


Figure 3.7: Distribution of the EFP observable identified in the search described by the text. Samples shown are separated by class using the sPlots weighting technique after applying a 50% background efficiency cut according to the outputs of the 9 isolation cone network. Also shown is the graph representation of the EFP.

Input features	AUC	TPR	EFP Scan
Single Iso Cone + $\sum p_T$	0.835	0.922	
9 Iso, $\sum p_T$	0.848	0.939	
9 Iso, $\sum p_T$, ADO EFP	0.871	0.953	CMS
9 Iso, $\sum p_T$, Quadratic EFP	0.870	0.956	CMS
9 Iso, $\sum p_T$, 4 IRC-safe EFP	0.868	0.949	Sim
9 Iso, $\sum p_T$, 5 EFP	0.865	0.954	Sim
Full details PFN	0.874	0.957	

Table 3.1: Comparison of the performance of the various networks discussed in the text. Performance is measured through ROC AUC, as well as signal efficiency (TPR) at 50% background efficiency. Standard error is evaluated to be $\lesssim 1 \times 10^{-3}$ for both metrics over a 1σ confidence interval (see Sec. 4.3 for details on calculation). While the reported performance values refer only to testing done on CMS data, the ‘‘EFP Scan’’ column indicates whether the EFP inputs used were identified as useful by a scan over CMS or simulated data. These results correspond to the ROC curves in Fig 3.8.

3.5 Conclusions

On collision data from the LHC, we apply neural networks to the problem of prompt muon discrimination. We investigate how much information is present in high and low-level representations of the data, finding that the traditionally used scalar isolation does not capture all useful classification information present at the low-level. Furthermore, we find that another high-level set of observables, the EFPs, may be used to create a network which performs almost as well as one operating at the low-level, while providing the advantage of being less complex and more human interpretable. In addition to being notable for using real rather than simulated data, this study demonstrates the use of weakly supervised training methods with CWoLa on low-level features, as well as performance evaluation without having access to individual class labels. Future work may include investigating the interpretation of the observables selected here, exploring how much information might be captured by other types of high-level observables, and the generalizability of these results. While our study indicates that additional information is available beyond the use of simple cones, and the identification of a single EFP observable which captures that information allows for simple application and interpretation, further work would be required before implementation within

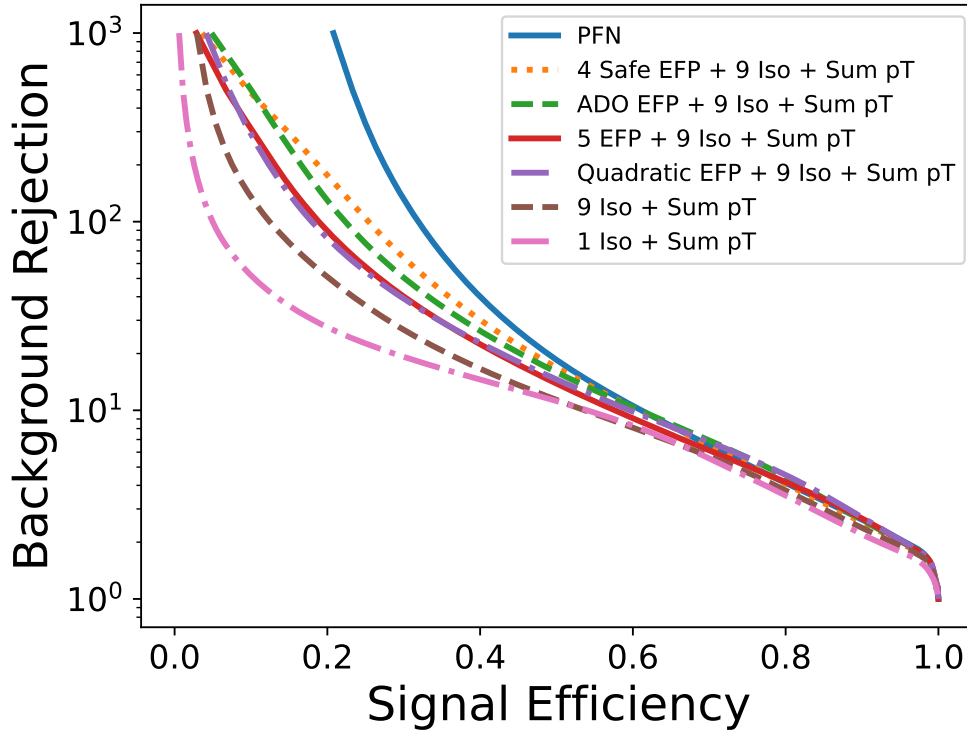


Figure 3.8: Comparison of the performance of the networks described in Table 3.1, via ROC curves. Shown is background rejection (inverse of efficiency) versus signal efficiency.

an experimental context. A robust estimate of the systematic uncertainties involved has not been done, which would be necessary to establish the optimal observables. Our result does not replace work by the experimental collaborations, but motivates further study.

Code and Data

The code for this paper can be found at https://github.com/Edwit4/learning_to_isolate_muons_in_data. The datasets will be provided upon reasonable request to the authors.

Chapter 4

Learning Broken Symmetries with Resimulation and Encouraged Invariance

This chapter is heavily based on work previously published in collaboration with Daniel White-son.

Recognizing symmetries in data allows for significant boosts in neural network training. In many cases, however, the underlying symmetry is present only in an idealized dataset, and is broken in the training data, due to effects such as arbitrary and/or non-uniform detector bin edges. Standard approaches, such as data augmentation or equivariant networks fail to represent the nature of the full, broken symmetry. We introduce a novel data-augmentation scheme that respects the true underlying symmetry and avoids artifacts by augmenting the training set with transformed pre-detector examples whose detector response is then resimulated. In addition, we encourage the network to treat the augmented copies identically, allowing it to learn the broken symmetry. While the technique can be extended to other sym-

metries, we demonstrate its application on rotational symmetry in particle physics calorimeter images. We find that neural networks trained with pre-detector rotations converge to a solution more quickly than networks trained with standard post-detector augmentation, and that networks modified to encourage similar internal treatment of augmentations of the same input converge even faster.

4.1 Introduction

Evidence for new physics and subtle features of the Standard Model are often hidden in high-volume, high-dimensional datasets produced at the Large Hadron Collider and in other high-intensity particle beams. Traditional methods of data analysis reduce the dimensionality of the data with engineered features which exploit our physical understanding of the task. While powerful, these heuristics often rely on simplifying assumptions which fail to fully capture the available information. Recently, artificial neural networks have demonstrated the capacity to exceed the performance of engineered features [23, 43, 49]. However, training such networks often requires vast quantities of data or computational resources, which can be problematic in practice.[16] There may only be a limited amount of data available for a given region of interest, or there may be computational limitations on how much data can feasibly be processed or generated with simulation programs. Learning strategies which are more efficient, reaching the performance plateau with fewer learning cycles or on smaller training samples, are therefore of great value to the particle physics research program.

Evidence for new physics and subtle features of the Standard Model are often hidden in high-volume, high-dimensional datasets produced at the Large Hadron Collider and in other high-intensity particle beams. Traditional methods of data analysis reduce the dimensionality of the data with engineered features which exploit our physical understanding of the task. While powerful, these heuristics often rely on simplifying assumptions which fail to fully

capture the available information. Recently, artificial neural networks have demonstrated the capacity to exceed the performance of engineered features [23, 43, 49]. However, training such networks often requires vast quantities of data or computational resources, which can be problematic in practice[16]. There may only be a limited amount of data available for a given region of interest, or there may be computational limitations on how much data can feasibly be processed or generated with simulation programs. Learning strategies which are more efficient, reaching the performance plateau with fewer learning cycles or on smaller training samples, are therefore of great value to the particle physics research program.

Efficiency may be gained by leveraging physical symmetries present in the data, where the data are closed under some transformation. This is typically done by enforcing equivariance in latent space operations or by requiring invariance in classification output.[37, 14, 51, 26, 80, 79, 25] If the symmetry group is understood exactly, the network structure might incorporate it, effectively constraining the functional search space. A contrasting strategy is data augmentation, expanding the training set to explicitly include transformations of the original data, allowing the network to infer the symmetry. However, in many cases the symmetry is exact only in an idealized scenario and in practice is *broken* by asymmetries such as the data-collection devices. For example, the exact and continuous translational and rotational symmetry of an idealized image of a cat is broken into a discrete symmetry by the camera's pixel edges. Arbitrary shifts or rotations of the cat only generate shifted or rotated versions of the image when they reflect the discrete symmetry of the pixel geometry. If the pixels themselves are not identical, the symmetry is broken further. What was a powerful, continuous and exact symmetry is broken into a less effective, discrete and approximate symmetry, which hinders our ability to exploit it to boost training efficiency.

Detection of particle energies by detectors presents a widespread and important example of broken symmetries, especially in the context of detection of a jet's energy deposition by a grid of calorimeter cells. A given jet is equally likely to have any orientation around its axis,

and rotation of its constituent particles around the axis changes none of its crucial physical observables. An idealized detector would respect this symmetry, but a realistic calorimeter is composed of discrete, non-uniform cells. The continuous symmetry is broken such that the detector pattern is preserved only under rotations of the particles by multiples of $\frac{\pi}{2}$. Rotations of the observed image by arbitrary angles, as is often done in data-augmentation strategies, introduce artifacts from the double-pixelization and fail to generate training examples which demonstrate the true symmetry, as demonstrated in Fig. 4.1. Enforcing symmetry in the network’s latent operations faces a similar hurdle, as the training data do not reflect the true continuous symmetry, only the more limited, broken symmetry.

Certain tasks may be relatively more resilient to these artifacts than others. For example, these effects will be less prominent for datasets containing high resolution images with pixels of uniform shape, where an approximate rotation will closely resemble the true rotation. These artifacts may also be less impactful for non-sparse images, where the exact value of a given pixel could be less vital in observing macroscopic structure, as illustrated in Fig 4.2. However, for calorimeter images, which may be low resolution, use non-uniform pixels, or be sparse, such artifacts are often significantly detrimental. In pixelated calorimeter images, an artifact-free augmented dataset cannot not be created from the post-detector data alone; a network trained on the post-detector augmentations is learning the wrong symmetry. The true symmetry, before the symmetry is broken, is not demonstrated in these augmentations [92].

We propose a novel data-augmentation technique which allows a network to learn a broken symmetry, assisting the learning process and increasing data efficiency. In traditional data augmentation, examples are synthesized directly from the post-detection training data, resulting in non-physical artifacts such that the augmented examples do not reflect the desired symmetry, making it challenging for the network infer their relationship. Perhaps more importantly, the post-detector augmented data do not represent the expected detector pat-

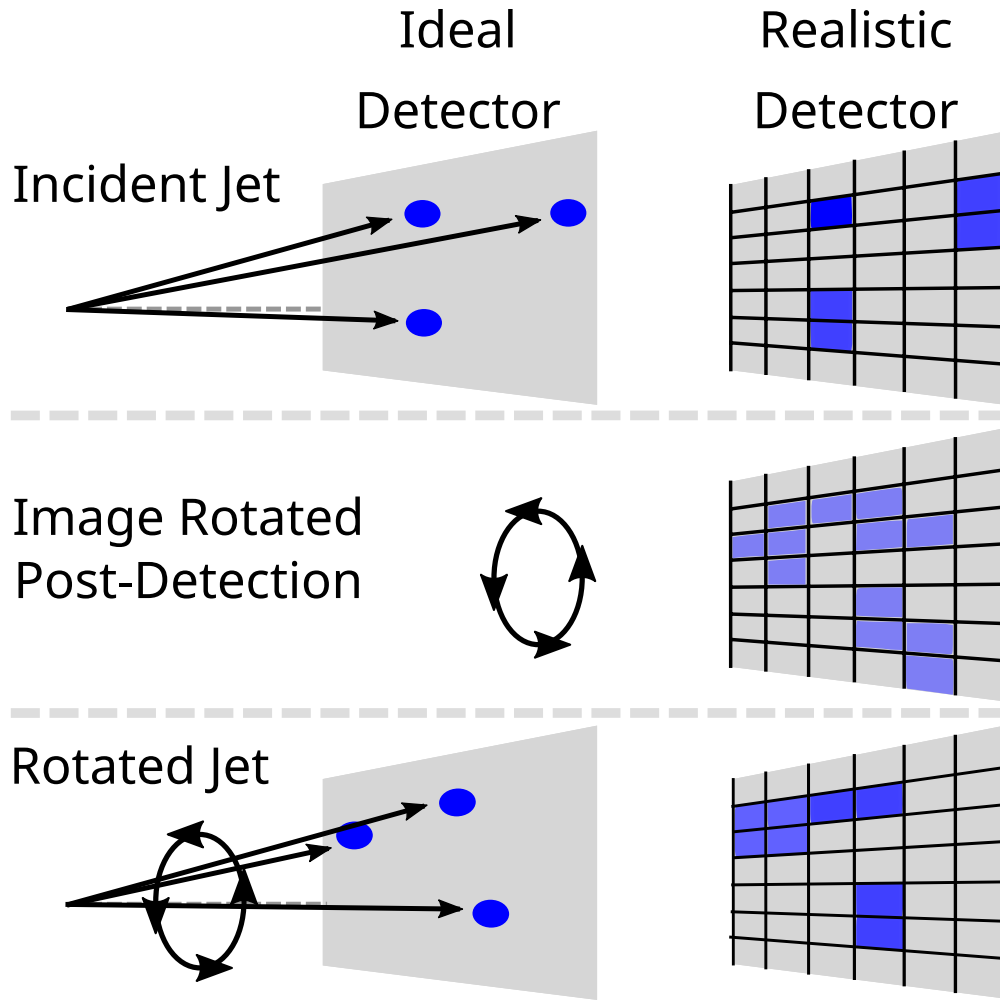


Figure 4.1: Demonstration of the breaking of a continuous rotation symmetry by the pixelization of a realistic detector. (Left) shows an ideal detector that performs no binning, while (Right) is a realistic detector which produces a pixelated image. (Top) shows a jet incident on the detector, and the images produced by each. (Middle) shows the image produced by the realistic detector rotated by an angle that is not a multiple of $\frac{\pi}{2}$. Rotating a pixelated image by such an angle results in artifacts and does not produce a detector image which reflects the true symmetry. (Bottom) shows the case where the jet itself is rotated pre-detector, producing an image which accurately represents the symmetry of the problem. Though it is not closed under rotation, it avoids introducing artifacts from post-detector rotation.

terns under the true detector transformation, such that a network which attempts to infer the symmetry is learning the wrong thing. We introduce two modifications to the traditional approach: pre-detector augmentation and encouraged invariance. If the data are generated with a simulator, as is the case with calorimeter images, pre-detector augmentation applies

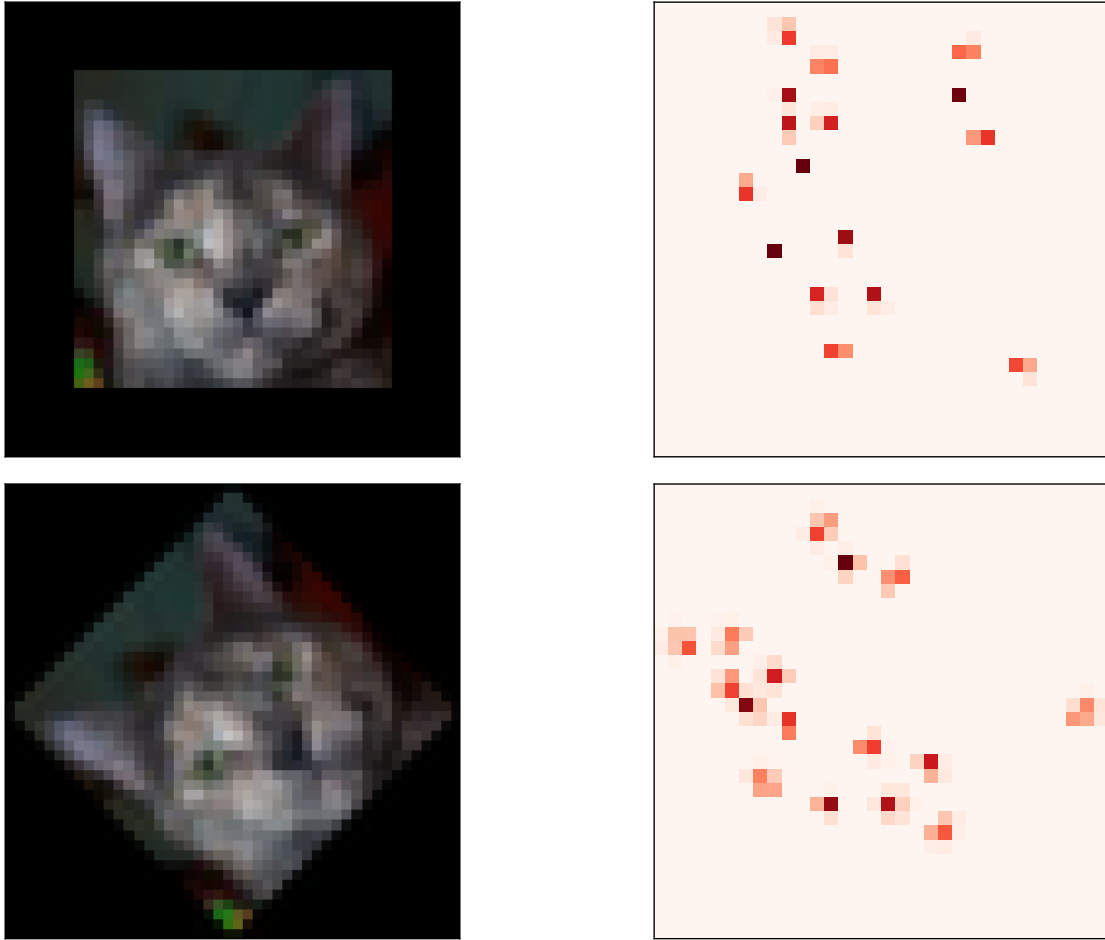


Figure 4.2: Demonstration of the significance of rotation-induced artifacts on less sparse (left) and more sparse (right) images by comparing original (top) and rotated images (bottom). Visually, the image on the left (CIFAR-10[64]) appears relatively unchanged after the rotation. The artifacts in the image on the right are far more prominent, and so might have relatively more influence on any learned strategy.

the transformation before the symmetry is broken, rather than after; this is also referred to as re-simulation [34]. The post-detector data are still not closed under the transformation, but now accurately represent the set of expected detector signatures under the true symmetry. Our second modification is to the loss function, penalizing the network for violating invariance across the pre-detector augmented examples of a given image during training. This effectively provides crucial missing information, indicating to the network which set of images are expected to produce the same output. In this way the symmetry that was initially hidden by the detector is exposed during training, and the network might learn to classify in an approximately invariant manner. If the simulation used to create the training data represents the physical process well, then a network trained using this technique might be fine tuned and applied to real world data, though this step is not explored in this study. We explore the effectiveness of this technique on a simplified toy dataset designed to have similar properties to calorimeter images, which is computationally cheaper to produce than using a full simulation pipeline. In order to probe the relative data efficiency of each method, the performance of training with a post-detection augmented dataset is compared to that of a pre-detection augmented dataset, as well as to a network where output invariance is encouraged, across a range of training set sizes.

The organization of this paper is as follows. Sec. 4.2 provides the details of the dataset used, summarizing how it is generated and the structure of the resulting data. Sec. 4.3 covers the proposed method for encouraging model output invariance, as well as model implementation and evaluation. Sec. 4.4 presents numerical results along with their discussion. Finally, Sec. 4.5 ends with conclusions, summarizing findings and future outlook.

4.2 Dataset

To reduce the computational and time costs associated with generating jet images through a full simulation pipeline, we use a simplified toy dataset to evaluate the viability of the proposed methods. The structure of the signal and background are inspired by jet substructure tasks (e.g. [67]), but not intended to be physically realistic.

This dataset is generated using Python v3.10.11[90] and Numpy v1.22.3[54]. A given toy example, referred to as an “event”, is composed of a list of simulated energy deposits, or intensity values with associated 2D spatial coordinates. In total, 3000 events consisting of 16 deposits each are generated, with half belonging to a signal class sample, and the other half belonging to a background sample. The intensity values are drawn from a uniform distribution of values between 0 and 1 for both classes. For background events, the location of every deposit is drawn from a uniform distribution over a disk of radius 1. For signal events, initially only a single deposit is drawn from the uniform disk distribution. Subsequent deposits are then drawn from a 2D Gaussian centered on this deposit, with equal variances of 0.3 in each dimension. If the location drawn happens to be outside of the radius of the disk, it is redrawn until it is inside the disk. This results in the signal events having an internal structure distinct from what is found in the background sample, in a spirit similar to the task of tagging jets with sub-structure [38]. To increase the complexity and realism of the problem, noise is added to the signal events by drawing a number of additional events from the uniform disk distribution, independent of the other deposits within the event. The noise deposits only differ in their spatial distribution, with the intensities values being drawn from the same distribution across all deposits. This means that the network cannot learn to separate the noise based on this information, and the overall normalization remains similar between the two classes.

To mimic the physical extent of the shower created by an incident particle, each deposit

is given a width by drawing 32 points from Gaussian distributions, with equal variances of 1×10^{-4} in each dimension, centered on each of the chosen locations. The intensity corresponding to a given deposit is then distributed evenly across these points. An example of an event generated before and after the spreading process may be seen in Fig. 4.3.

Pre-detector augmented images are created by applying rotations in 45° increments to the events at this step, creating a set of 8 copies for every event. The intensities are then binned according to their position for a simplified detector response, creating the final pixelated event images. Two variants of the pixelated events are created, with one using square binning on a 32×32 grid, and the other using rectangular binning on a 4×32 grid. An example of a single event with each binning scheme is shown in Fig. 4.4.

Rectangular pixel geometries are present in real detectors[50], and this non-uniformity could exaggerate the degree to which the effects of rotation are obscured by binning. Traditionally, in computer vision tasks where augmentations are applied, it is done directly to pixelated images. In our pipeline this would be equivalent to applying the rotation after the binning step. By applying rotations prior to the binning step, we are able to augment the dataset with synthetic examples which have been transformed exactly as expected, without introducing non-physical effects. For comparison, we create another augmented dataset where augmentations are applied this way using OpenCV v4.7.0[27], with the default bilinear interpolation method for rotating pixels. It should be noted that other interpolation methods, as well as foregoing interpolation entirely in the case that re-binning is not required to produce valid network inputs, may result in different performance. While the performance may change, none of these methods will be capable of recovering information lost through pixelization, and so will not match the output obtained by applying rotations pre-detection. The differences between applying rotations before and after the detector step are demonstrated in Fig. 4.5. This leads to higher quality synthetic examples, as well as a way to expose a symmetry hidden by the binning to a neural network during training. This may only be done in

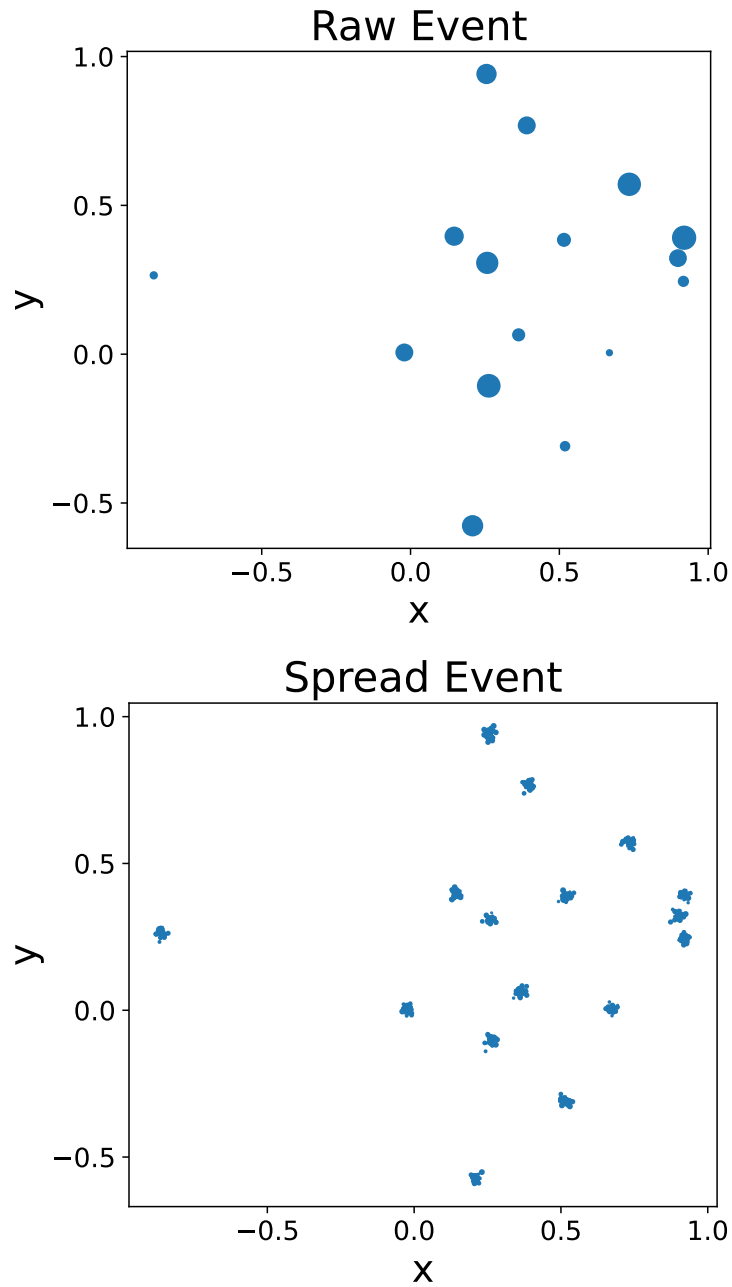


Figure 4.3: Visualization of dataset generation process. (Top) An example of an event before pixelization, where the size of each deposit is proportional to its energy. (Bottom) The same event, after the deposits have been distributed over a small area to simulate shower width effects which can lead to deposits over adjacent pixels. In the bottom pane the size of the deposits is arbitrary.

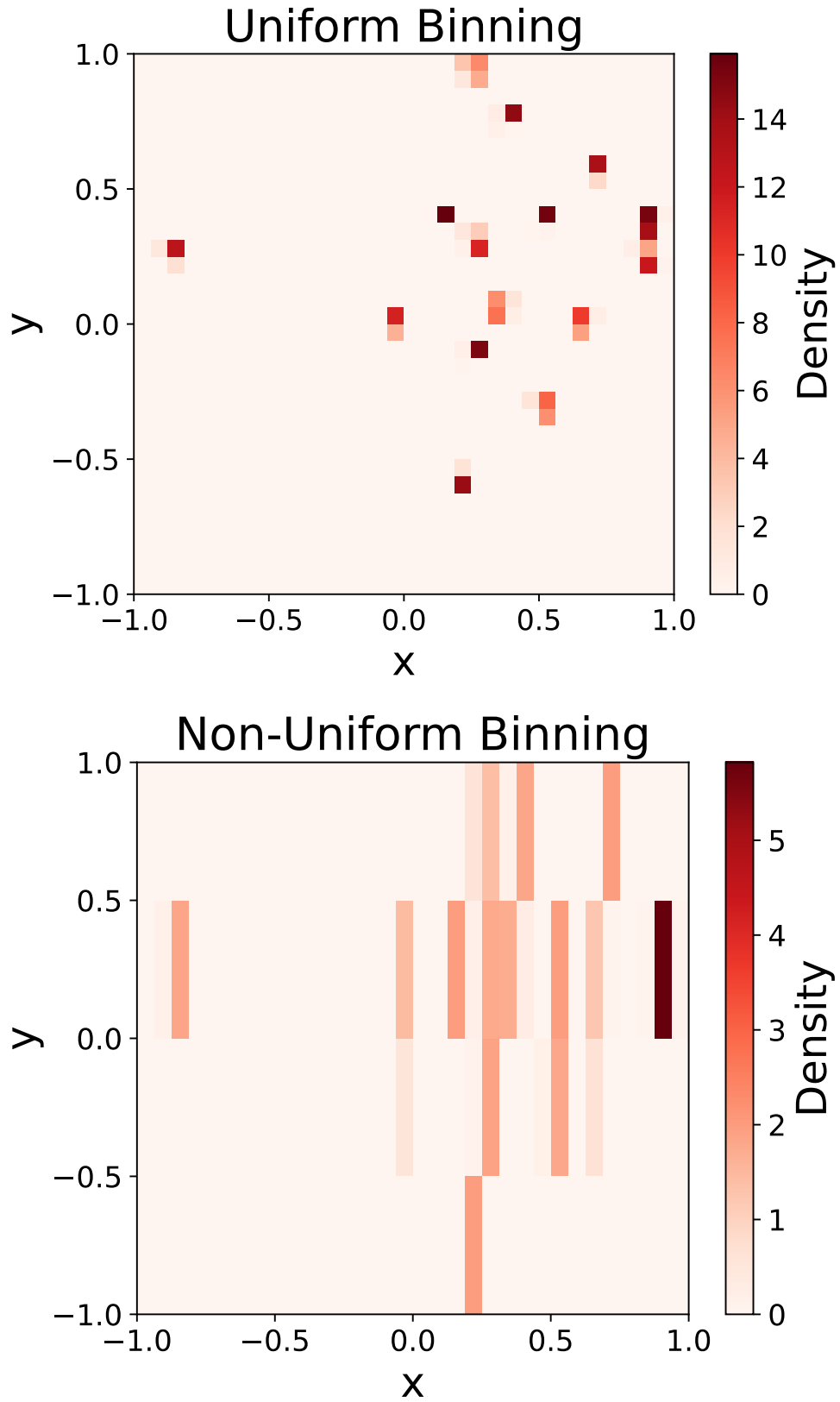


Figure 4.4: (Top) The event shown in Fig. 4.3 with uniform binning (Top) and rectangular binning (Bottom) applied.

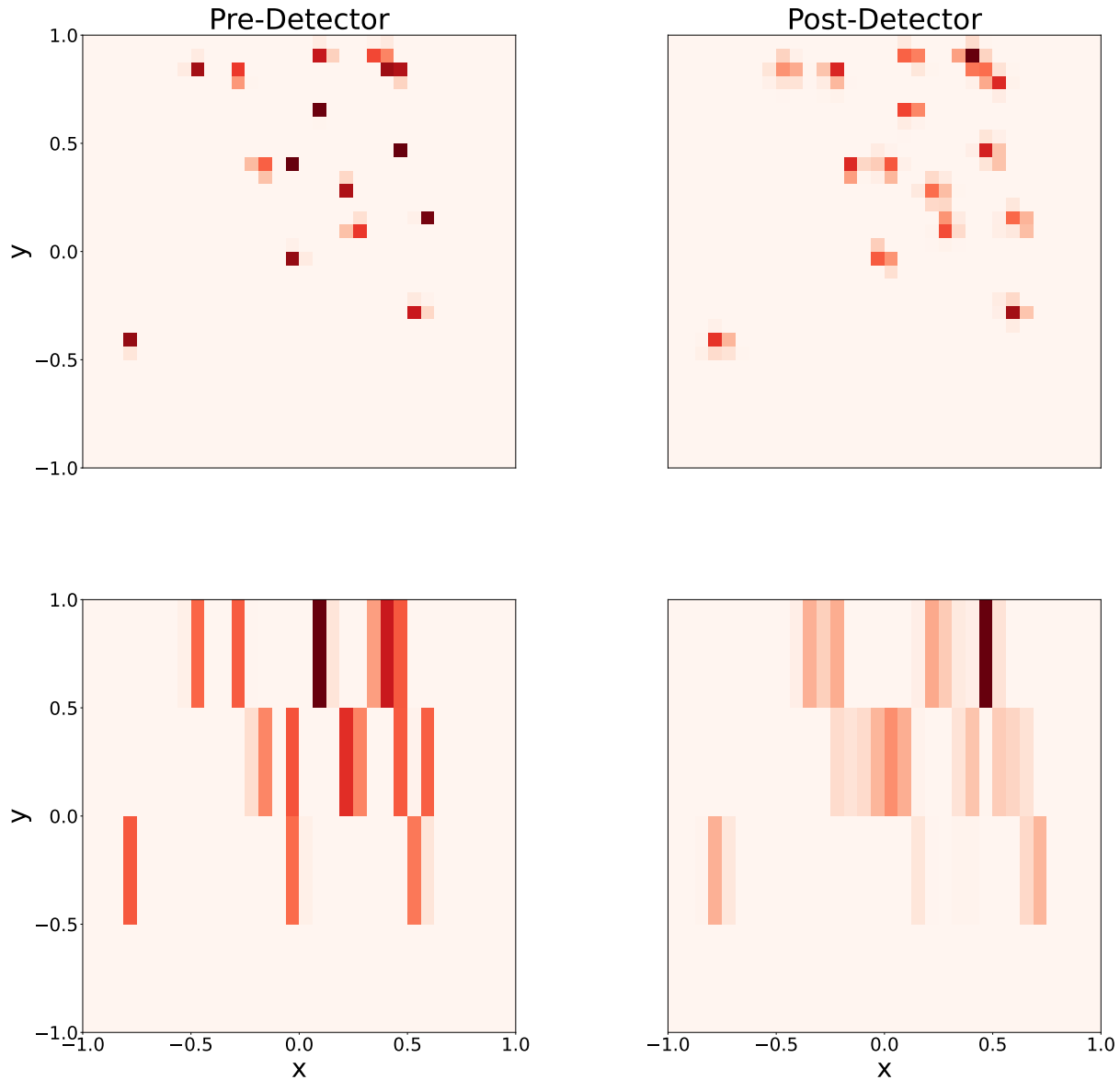


Figure 4.5: The event shown in Fig. 4.3 rotated at 45 degrees from their original orientation with uniform binning (Top) and rectangular binning (Bottom) applied. Applying rotations prior to binning (Left) avoids the interpolation artifacts which arise from applying rotations after binning (Right), where the images look relatively blurrier and more washed out.

cases where the data is accessible prior to the step which results in pixelization, which could be the case for simulation but not for real world collider data.

4.3 Encouraged Invariance

We extend our novel learning method beyond the pre-detector data augmentation procedure outlined above, by the use of a loss function which explicitly encourages a neural network to learn a classification invariant to augmentations applied prior to the binning. This is achieved by adding an additional component to the usual classification loss function, which penalizes differences in outputs across all of the augmented variants for a given event. This takes the form shown in Eq. 4.1, where a and b are scalar weight parameters, L_{cls} is a typical loss function used for classification, and L_{inv} is a loss function responsible for encouraging invariance.

$$L = aL_{\text{cls}} + bL_{\text{inv}} \tag{4.1}$$

For all models presented here, binary cross-entropy is used for L_{cls} , and mean squared error is used for L_{inv} . Training is constrained to process all augmented copies for a given event within the same batch before a gradient update, producing a network output for each individual copy. The standard deviation is computed across the outputs produced by augmented copies of a given event. The outputs themselves are passed to L_{cls} , and the standard deviations are passed to L_{inv} . L_{inv} penalizes for non-zero deviations, and in the case of mean squared error this is done by comparing the computed values to zero. This effectively introduces more information to the training, by encouraging the network to treat all augmented versions of the jet identically, even if the post-detector data are not simply related under transformation in a way that would allow the network to infer it.

For models that are not trained with encouraged invariance, we effectively use $a = 1$ and $b = 0$ reducing the expression to the usual classification loss, and otherwise the weighting is tuned during hyperparameter optimization. Fully Connected Networks (FCNs) and Particle Flow Networks[62] (PFNs), of the deep sets architecture[96], are used to process the data, implemented and trained with Pytorch v2.0.0[72]. While the total number of pixels in an event is fixed, the number of non-zero pixels may differ between events. PFNs are a natural choice for this type of data, as they take a permutation invariant set of variable length inputs, and have been shown to yield good performance with collider data in previous studies[38]. The hyperparameters of each model are selected by performing 5-fold cross-validation with a random search over a set of learning rates, batch sizes, layer sizes, and loss term weights with early stopping based on the validation loss. Specifically, the learning rates are allowed to vary from 1×10^{-3} to 1×10^{-6} in steps of powers of 10, batch sizes from 32 to 1024 and layer sizes from 32 to 512 in steps of powers of 2, and loss term weights are drawn from the set $\{0.01, 1, 10, 100, 300\}$. A minimum change of 1×10^{-3} in the validation loss with a patience of 10 epochs is used as the early stopping criteria. This is done at the largest training size in order to obtain stable parameters in a computationally efficient manner. Performance is then measured by constructing the standard Receiver Operating Characteristic (ROC) and evaluating the ROC AUC and signal efficiency at a fixed background efficiency of 50%, over 100 event ensembles using the optimized hyperparameters. Error due to statistical uncertainty is estimated from these ensembles to 1σ . Both architectures are evaluated in this way for the uniform and non-uniform image binning schemes across a range of training set sizes. Each model is evaluated with augmentations applied prior to and after binning, and the pre-detector variants are additionally evaluated with invariance encouraged.

4.4 Results

Learning strategies which take advantage of symmetries in the data can provide efficiencies in training, reaching a performance plateau with smaller training samples. We explore the relative power of several methods by evaluating the AUC with respect to the proportion of the full dataset used to train. Figure 4.6 shows the dependence of the ROC AUC for FCNs and PFNs on the training set proportion for uniform pixels or non-uniform pixels. Additional figures and results for signal efficiency at 50% background efficiency are included in the appendix. For simplicity, we will focus on the AUC performance in the discussion that follows, as the trends found in the signal efficiency scans are very similar.

Scans are performed across a range of training set sizes, from 480 to 3000 unique events, not counting the 8 synthetic variants of each image that are included in the data augmentation approaches. We observe that in all cases the performance increases as the training set size increases, and that not using any augmentation yields the lowest performance, as might be expected.

In the case of uniform pixelization, the FCNs show a clear trend across the various methods. At every training set size, there is a gain from one method to the next, with post-detection augmentations giving the smallest gain, followed by pre-detection augmentations, and then by encouraged invariance with the largest gain. The PFN outperforms the FCN, which may not be surprising, as it is a more complex network that is better suited for this dataset. Further, performance quickly converges for both augmentation schemes as well as encouraging invariance. It appears that for uniformly pixelated data, where an approximate post-detection rotation leads to something close to a true pre-detection rotation, coupling any of the tested training methods with a more powerful network overcomes much of the challenge presented by the broken symmetry.

When non-uniform pixelization is used, differences between methods are apparent for both

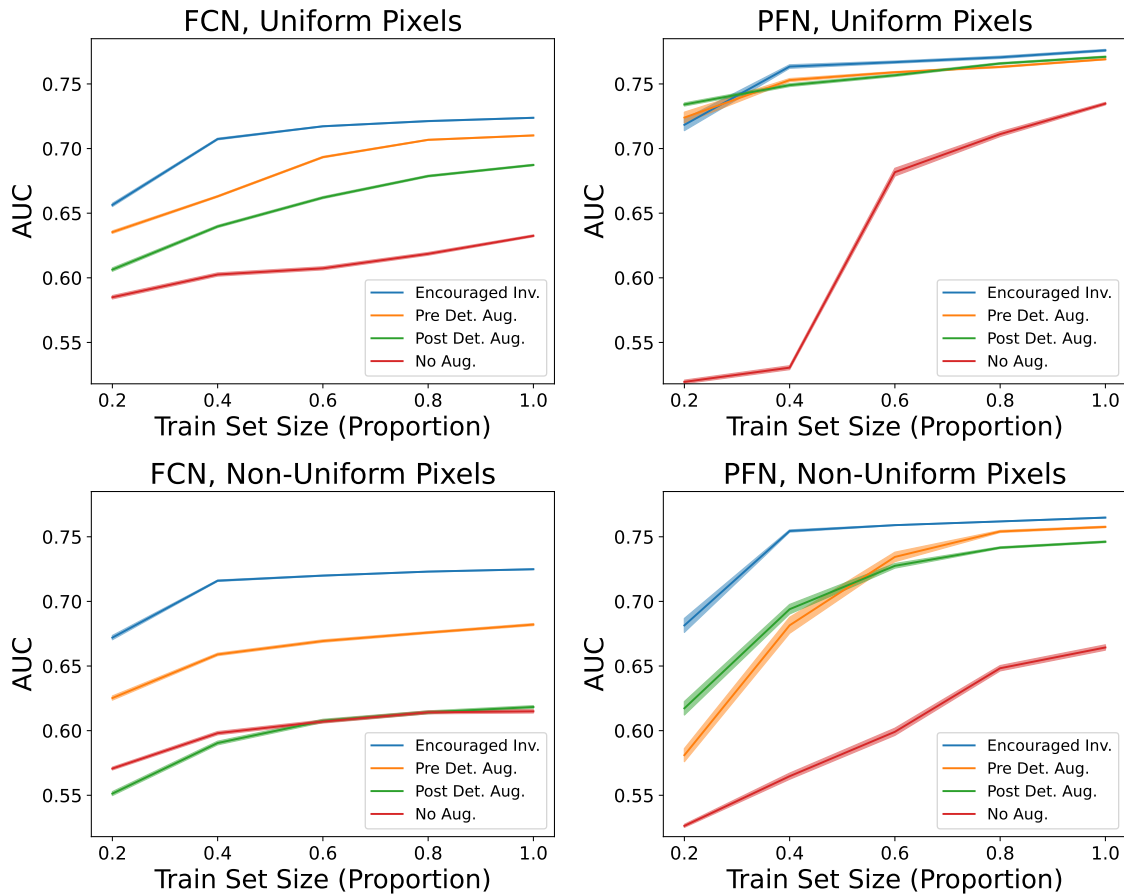


Figure 4.6: Performance of FCNs (left) and PFNs (right) trained on uniformly binned data (top) or non-uniformly binned data (bottom) as a function of training set size. Though results depend on the nature of the task and the structure of the network, pre-detector augmentation and resimulation typically improves the learning rate, and encouraged invariance provides a further boost in learning.

types of networks. The FCN shows no clear gain in performance when using post-detection augmentations, but does benefit from pre-detection augmentations, and sees the largest improvement from using encouraged invariance. Interestingly, the PFN not only demonstrates a gain from using the post-detection augmented data, but shows a comparable gain from the pre-detection augmentations. This suggests that there is useful information in the post-detection augmentations, but that it is harder for the FCN to take advantage of it, than it is for the PFN. Unlike with uniform pixelization, the performance of the PFN does not completely converge across every method, but it does approach convergence more rapidly than the FCN.

These findings suggest that the differences in these methods are more apparent when training data is limited and when the symmetry breaking process is more exaggerated. As including pre-detection augmented image variants in the training set, and encouraging invariance, consistently yield performance boosts for the FCN, these techniques may be especially beneficial in a computationally limited setting where the use of a simpler network is favored, even in the case that the symmetry of interest is only lightly obscured. Notably, the performance gains obtained through the use of the pre-detection augmentations suggests that synthesizing new examples this way does indeed lead to higher quality augmented copies than applying the transformations post-detection. It is of additional significance that across both pixelization schemes, encouraging invariance almost always yields the best performance. This shows that training with explicit symmetry awareness meaningfully enhances the training process, in a way that is achievable even when the symmetry is obscured in the final data, to the point of not being exactly recoverable due to loss of information. While these methods do require access to a way to apply transformations during data generation, it is not necessary to attempt to understand the specific details of how the transformation ends up being represented in the final data, which in itself may present an advantage. The AUC values for the smallest and largest training set sizes are tabulated in Table 4.1.

Arch.	Augm.	Uniform bins		Non-uniform bins	
		small set	large set	small set	large set
FCN	None	0.585(1)	0.632(1)	0.571(1)	0.615(1)
	post-det	0.606(2)	0.687(1)	0.551(1)	0.618(1)
	pre-det.	0.635(1)	0.710(1)	0.625(2)	0.682(1)
	enc. inv.	0.656(1)	0.724(1)	0.672(2)	0.725(1)
PFN	None	0.519(1)	0.735(1)	0.526(1)	0.664(2)
	post-det	0.734(1)	0.771(1)	0.617(5)	0.746(1)
	pre-det	0.724(4)	0.770(1)	0.581(5)	0.758(1)
	enc. inv.	0.718(4)	0.776(1)	0.681(5)	0.765(1)

Table 4.1: The ROC AUC performance for models with various augmentation strategies described in the text, trained and evaluated on events with uniform or non-uniform pixelization, shown for the smallest and largest training set sizes tested.

4.5 Conclusions

We propose a method for training neural networks with greater data efficiency, in the case that a symmetry known to be present in the data at some point during its generation is broken in the representation that is ultimately observed. By creating an augmented dataset where the relevant transformation is applied at a step in the generation process when the symmetry is fully represented, higher quality synthetic examples may be obtained. This information can be further leveraged by explicitly encouraging invariance across augmented variants of a given example through the loss during training.

We successfully demonstrate this method on a toy problem designed to probe the viability of these techniques for use with collider data. In the case that a rotational symmetry is obscured by a detector-like binning process, training on a dataset which uses the higher quality augmentations results in better performance, especially with simpler networks. Further, encouraging invariance can allow for even more data efficient training, showing that a network may be trained in a symmetry aware way even if the symmetry is not perfectly represented in the observed data. Our results indicate that the utility of this technique depends on factors such as the amount of data available, the degree to which a symmetry is hidden within the

data, and the type of network used.

Further work is necessary to determine how well this method can be used with both simulated and real world collider data. Since pre-detector intervention and resimulation cannot be applied directly to real world data, it must also be explored how well performance gains are preserved when transferring from models trained on simulation. This study also does not explore the optimization of the number of augmented copies created. Using more augmented copies may serve to close some of the performance gap between the models which use synthetic examples, and those which encourage invariance. This effect would likely be dependent on the dataset, and depending on the degree to which performance is dependent on it, it may be worth optimizing along with other hyperparameters.

More generally, our findings suggest that it may be possible to improve the learning efficiency in other scenarios in which a symmetry is only approximately realized, such as Lorentz symmetry [24, 30].

Code and Data

The code for this paper can be found at https://github.com/Edwit4/learning_broken_symmetries. The datasets will be provided upon reasonable request to the authors.

Bibliography

- [1] CERN Open Data Portal.
- [2] Search for direct pair production of sleptons and charginos decaying to two leptons and neutralinos with mass splittings near the W -boson mass in $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector. 9 2022.
- [3] Search for a new Z' gauge boson in 4μ events with the ATLAS experiment. 1 2023.
- [4] M. 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. *Phys. Rev.*, D97(5):052010, 2018.
- [5] G. Aad et al. Muon reconstruction performance of the ATLAS detector in proton–proton collision data at $\sqrt{s} = 13$ TeV. *Eur. Phys. J. C*, 76(5):292, 2016.
- [6] G. Aad et al. Muon reconstruction performance of the ATLAS detector in proton–proton collision data at $\sqrt{s} = 13$ TeV. *Eur. Phys. J.*, C76(5):292, 2016.
- [7] G. Aad et al. Dijet resonance search with weak supervision using $\sqrt{s} = 13$ TeV pp collisions in the ATLAS detector. *Phys. Rev. Lett.*, 125(13):131801, 2020.
- [8] G. Aad et al. Searches for electroweak production of supersymmetric particles with compressed mass spectra in $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector. *Phys. Rev. D*, 101(5):052005, 2020.
- [9] G. Aad et al. Muon reconstruction and identification efficiency in ATLAS using the full Run 2 pp collision data set at $\sqrt{s} = 13$ TeV. *Eur. Phys. J. C*, 81(7):578, 2021.
- [10] R. Aaij et al. Search for Dark Photons Produced in 13 TeV pp Collisions. *Phys. Rev. Lett.*, 120(6):061801, 2018.
- [11] R. Aaij et al. Searches for low-mass dimuon resonances. *JHEP*, 10:156, 2020.
- [12] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga,

- S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [13] G. Agarwal, L. Hay, I. Iashvili, B. Mannix, C. McLean, M. Morris, S. Rappoccio, and U. Schubert. Explainable AI for ML jet taggers using expert variables and layerwise relevance propagation. *JHEP*, 05:208, 2021.
- [14] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 37–45, Los Alamitos, CA, USA, dec 2015. IEEE Computer Society.
- [15] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H. S. Shao, T. Stelzer, P. Torrielli, and M. Zaro. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *JHEP*, 07:079, 2014.
- [16] L. Alzubaidi, J. Zhang, A. Humaidi, et al. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *J Big Data*, 8:53, 2021.
- [17] ATLAS Collaboration. Deep Learning for Pion Identification and Energy Calibration with the ATLAS Detector. Technical Report ATL-PHYS-PUB-2020-018, CERN, Geneva, Jul 2020.
- [18] ATLAS Collaboration. Software & computing, 2023. Accessed: 2023-09-29.
- [19] P. Baldi. *Deep Learning in Science*. Cambridge University Press, Cambridge, UK, 2021.
- [20] P. Baldi, K. Bauer, C. Eng, P. Sadowski, and D. Whiteson. Jet Substructure Classification in High-Energy Physics with Deep Neural Networks. *Phys. Rev.*, D93(9):094034, 2016.
- [21] P. Baldi and P. Sadowski. The dropout learning algorithm. *Artificial Intelligence*, 210C:78–122, 2014.
- [22] P. Baldi, P. Sadowski, and D. Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, 5, 2014.
- [23] P. Baldi, P. Sadowski, and D. Whiteson. Searching for Exotic Particles in High-Energy Physics with Deep Learning. *Nature Commun.*, 5:4308, 2014.
- [24] A. Bogatskiy, B. Anderson, J. T. Offermann, M. Roussi, D. W. Miller, and R. Kondor. Lorentz Group Equivariant Neural Network for Particle Physics. 6 2020.
- [25] A. Bogatskiy et al. Symmetry Group Equivariant Architectures for Physics. In *Snowmass 2021*, 3 2022.

- [26] A. Bogatskiy, T. Hoffman, D. W. Miller, and J. T. Offermann. PELICAN: Permutation Equivariant and Lorentz Invariant or Covariant Aggregator Network for Particle Physics. 11 2022.
- [27] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [28] R. Brun and F. Rademakers. ROOT: An object oriented data analysis framework. *Nucl. Instrum. Meth. A*, 389:81–86, 1997.
- [29] C. Brust, P. Maksimovic, A. Sady, P. Saraswat, M. T. Walters, and Y. Xin. Identifying boosted new physics with non-isolated leptons. *JHEP*, 04:079, 2015.
- [30] A. Butter, G. Kasieczka, T. Plehn, and M. Russell. Deep-learned Top Tagging with a Lorentz Layer. *SciPost Phys.*, 5(3):028, 2018.
- [31] C. Cesarotti, Y. Soreq, M. J. Strassler, J. Thaler, and W. Xue. Searching in CMS Open Data for Dimuon Resonances with Substantial Transverse Momentum. *Phys. Rev. D*, 100(1):015021, 2019.
- [32] S. Chang, T. Cohen, and B. Ostdiek. What is the Machine Learning? *Phys. Rev. D*, 97(5):056009, 2018.
- [33] S. Chatrchyan et al. The CMS Experiment at the CERN LHC. *JINST*, 3:S08004, 2008.
- [34] D. Chirkin. Event reconstruction in IceCube based on direct event re-simulation. In *33rd International Cosmic Ray Conference*, page 0581, 2013.
- [35] F. Chollet et al. Keras. <https://keras.io>, 2015.
- [36] J. Cogan, M. Kagan, E. Strauss, and A. Schwartzman. Jet-Images: Computer Vision Inspired Techniques for Jet Tagging. *JHEP*, 02:118, 2015.
- [37] T. Cohen and M. Welling. Group equivariant convolutional networks. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR.
- [38] J. Collado, K. Bauer, E. Witkowski, T. Faucett, D. Whiteson, and P. Baldi. Learning to isolate muons. *JHEP*, 21:200, 2020.
- [39] J. Collado, J. N. Howard, T. Faucett, T. Tong, P. Baldi, and D. Whiteson. Learning to Identify Electrons. 11 2020.
- [40] J. H. Collins, K. Howe, and B. Nachman. Anomaly Detection for Resonant New Physics with Machine Learning. *Phys. Rev. Lett.*, 121(24):241803, 2018.
- [41] J. H. Collins, K. Howe, and B. Nachman. Extending the search for new resonances with machine learning. *Physical Review D*, 99(1), jan 2019.

- [42] J. de Favereau et al. DELPHES 3, A modular framework for fast simulation of a generic collider experiment. *JHEP*, 1402:057, 2014.
- [43] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, and A. Schwartzman. Jet-images — deep learning edition. *JHEP*, 07:069, 2016.
- [44] L. M. Dery, B. Nachman, F. Rubbo, and A. Schwartzman. Weakly Supervised Classification in High Energy Physics. *JHEP*, 05:145, 2017.
- [45] B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1 – 26, 1979.
- [46] W. Falcon and T. P. L. team. PyTorch Lightning. <https://github.com/Lightning-AI/lightning>, Mar. 2019.
- [47] T. Faucett, J. Thaler, and D. Whiteson. Mapping Machine-Learned Physics into a Human-Readable Space. 10 2020.
- [48] T. Faucett, J. Thaler, and D. Whiteson. Mapping Machine-Learned Physics into a Human-Readable Space. 10 2020.
- [49] M. Feickert and B. Nachman. A Living Review of Machine Learning for Particle Physics. 2 2021.
- [50] A. Gaudiello. Atlas pixel ibl modules construction experience and developments for future upgrade. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 796:56–59, 2015. Proceedings of the 10th International Conference on Radiation Effects on Semiconductor Materials Detectors and Devices.
- [51] R. Gens and P. M. Domingos. Deep symmetry networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.
- [52] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.
- [53] Z. Hall and J. Thaler. Photon isolation and jet substructure. *JHEP*, 09:164, 2018.
- [54] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020.
- [55] L. Hertel, J. Collado, P. Sadowski, J. Ott, and P. Baldi. Sherpa: Robust hyperparameter optimization for machine learning. *SoftwareX*, 2020. Software available at: <https://github.com/sherpa-ai/sherpa>.

- [56] I. Hoenig, G. Samach, and D. Tucker-Smith. Searching for dilepton resonances below the Z mass at the LHC. *Phys. Rev. D*, 90:023, 2014.
- [57] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.
- [58] J. N. Howard, S. Mandt, D. Whiteson, and Y. Yang. Learning to simulate high energy particle collisions from unlabeled data. *Scientific Reports*, 12(1), may 2022.
- [59] V. Khachatryan et al. Search for supersymmetry in the vector-boson fusion topology in proton-proton collisions at $\sqrt{s} = 8$ TeV. *JHEP*, 11:189, 2015.
- [60] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [61] P. T. Komiske, E. M. Metodiev, and J. Thaler. Energy flow polynomials: A complete linear basis for jet substructure. *JHEP*, 04:013, 2018.
- [62] P. T. Komiske, E. M. Metodiev, and J. Thaler. Energy Flow Networks: Deep Sets for Particle Jets. *JHEP*, 01:121, 2019.
- [63] P. T. Komiske, E. M. Metodiev, and J. Thaler. Energy flow networks: deep sets for particle jets. *Journal of High Energy Physics*, 2019(1), Jan 2019.
- [64] A. Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009.
- [65] A. J. Larkoski, G. P. Salam, and J. Thaler. Energy Correlation Functions for Jet Substructure. *arXiv.org*, Apr. 2013.
- [66] H. M. Lee. Lectures on physics beyond the standard model. *Journal of the Korean Physical Society*, 78(11):985–1017, may 2021.
- [67] Y. Lu, A. Romero, M. J. Fenton, D. Whiteson, and P. Baldi. Resolving extreme jet substructure. *JHEP*, 08:046, 2022.
- [68] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [69] E. M. Metodiev, B. Nachman, and J. Thaler. Classification without labels: Learning from mixed samples in high energy physics. *JHEP*, 10:174, 2017.
- [70] V. M. Mikuni. Collider Physics Measurements in High Jet Multiplicity Final States, 2021. Presented 2021.
- [71] B. Nachman. A guide for deploying Deep Learning in LHC searches: How to achieve optimality and account for uncertainty. *SciPost Phys.*, 8:090, 2020.

- [72] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [73] J. Pata, J. Duarte, J.-R. Vlimant, M. Pierini, and M. Spiropulu. MLPF: Efficient machine-learned particle-flow reconstruction using graph neural networks. 1 2021.
- [74] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [75] M. Pivk and F. R. Le Diberder. SPlot: A Statistical tool to unfold data distributions. *Nucl. Instrum. Meth. A*, 555:356–369, 2005.
- [76] T. Roxlo and M. Reece. Opening the black box of neural nets: case studies in stop/top discrimination. 4 2018.
- [77] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *CoRR*, abs/1312.6120, 2013.
- [78] R. Schoefbeck. Search for supersymmetry with extremely compressed spectra with the atlas and cms detectors. *Nuclear and Particle Physics Proceedings*, 273-275:631 – 637, 2016. 37th International Conference on High Energy Physics (ICHEP).
- [79] C. Shimmin. Particle Convolution for High Energy Physics. 7 2021.
- [80] C. Shimmin, Z. Li, and E. Smith. Rethinking SO(3)-equivariance with Bilinear Tensor Networks. 3 2023.
- [81] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.
- [82] A. M. Sirunyan et al. Particle-flow reconstruction and global event description with the CMS detector. *JINST*, 12(10):P10003, 2017.
- [83] A. M. Sirunyan et al. Particle-flow reconstruction and global event description with the CMS detector. *JINST*, 12(10):P10003, 2017.
- [84] A. M. Sirunyan et al. Performance of the CMS muon detector and muon reconstruction with proton-proton collisions at $\sqrt{s} = 13$ TeV. *JINST*, 13(06):P06015, 2018.
- [85] A. M. Sirunyan et al. Measurement of the $t\bar{t}b\bar{b}$ production cross section in the all-jet final state in pp collisions at $\sqrt{s} = 13$ TeV. *Phys. Lett. B*, 803:135285, 2020.
- [86] T. Sjostrand, S. Mrenna, and P. Z. Skands. PYTHIA 6.4 Physics and Manual. *JHEP*, 0605:026, 2006.

- [87] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [88] A. Tumasyan et al. Search for low-mass dilepton resonances in Higgs boson decays to four-lepton final states in proton–proton collisions at $\sqrt{s} = 13$ TeV. *Eur. Phys. J. C*, 82(4):290, 2022.
- [89] A. Tumasyan et al. Search for supersymmetry in final states with two or three soft leptons and missing transverse momentum in proton-proton collisions at $\sqrt{s} = 13$ TeV. *JHEP*, 04:091, 2022.
- [90] G. Van Rossum and F. L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.
- [91] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
- [92] A. G. Wilson and P. Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *CoRR*, abs/2002.08791, 2020.
- [93] E. Witkowski, B. Nachman, and D. Whiteson. Learning to isolate muons in data. 2023.
- [94] R. L. Workman et al. Review of Particle Physics. *PTEP*, 2022:083C01, 2022.
- [95] S. Wunsch, R. Friese, R. Wolf, and G. Quast. Identifying the relevant dependencies of the neural network response on characteristics of the input space. *Comput. Softw. Big Sci.*, 2(1):5, 2018.
- [96] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. Salakhutdinov, and A. Smola. Deep sets, 2018.

Appendix A

Appendix: Learning to Isolate Muons

A.0.1 A. Neural Network Architectures

All networks were trained in Tensorflow[12] and Keras[35]. The networks were optimized with Adam [60] for up to 100 epochs with early stopping. For all networks except the PFNs, the weights were initialized using orthogonal weights[77]. Hyperparameters were optimized using Bayesian optimization with the Sherpa hyperparameter optimization library [55]. The variables and ranges for the hyperparameters are shown in tables A.1 and A.2.

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

B. Muon Image Networks

The pixelated images were preprocessed to have zero mean and unit standard deviation. We tried rotating the images as in [20] but performance was considerably lowered by this 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 [52], 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 [87, 21] with value 0.2388 on the fully connected layers and an initial learning rate of 0.0003 and batch size of 128.

Table A.1: Hyperparameter ranges for bayesian optimization of convolutional networks

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

C. Particle-Flow Networks

The Particle Flow Network (PFN) is trained using the `energyflow` package[63]. 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.

D. Isolation Cone and EFP Networks

The isolation inputs and EFPs are preprocessed by subtracting 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[52] and a learning rate of 0.0003 and dropout rate of 0.0547.

Table A.2: Hyperparameter ranges for Bayesian optimization of fully connected networks

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

A.0.2 ADO comparison

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

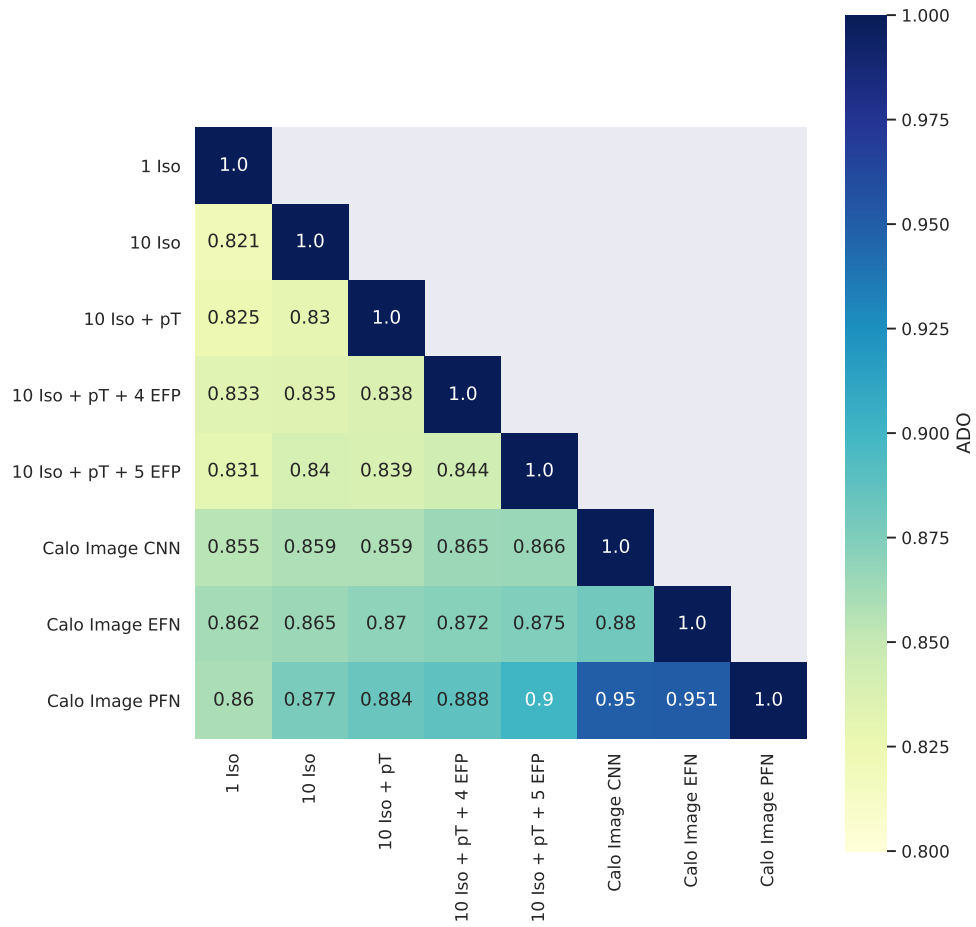


Figure A.1: Comparison of the similarity of decisions made by pairs of networks, as quantified by the Average Decision Ordering (ADO) [48], defined in the text.

Appendix B

Appendix: Learning to Isolate Muons in Data

CWoLa assumes that the mixed samples are generated in such a way that a given component feature is distributed the same way in one sample as it is in the other. While we cannot explicitly demonstrate this on an unlabeled dataset, we can use a simulated dataset similar to the experimental data to probe whether we can reasonably expect this assumption to hold.

We simulate events where prompt muons are generated by the process $pp \rightarrow Z \rightarrow \mu^+\mu^-$, and non-prompt muons by $pp \rightarrow b\bar{b}$. A center of mass energy of $s = \sqrt{(13)} \text{ TeV}$ is used. Madgraph5, Pythia, and Delphes are used respectively for collision and heavy boson decay simulation, showering and hadronization, and the detector simulation, with pile-up included. In total we generate 22766 events, where half are prompt and the other half are non-prompt events. The muon transverse momentum and pseudorapidity distributions for this dataset are shown in Fig B.1, and the average event images are shown in Fig B.2, where quantities are separated between the prompt and non-prompt distributions.

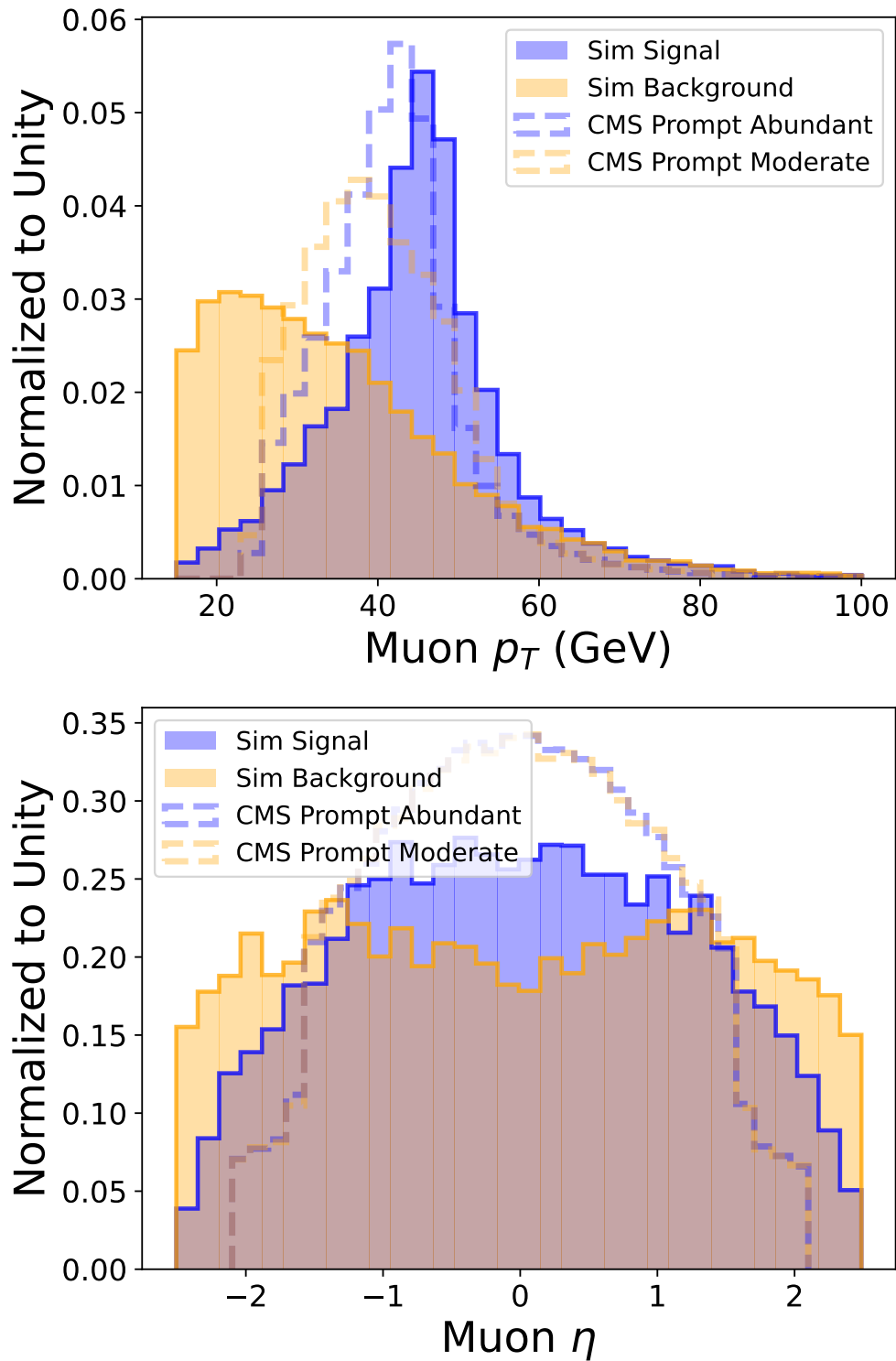


Figure B.1: (MG5+Pythia+Delphes) Distributions of the simulated muon transverse momentum and pseudorapidity.

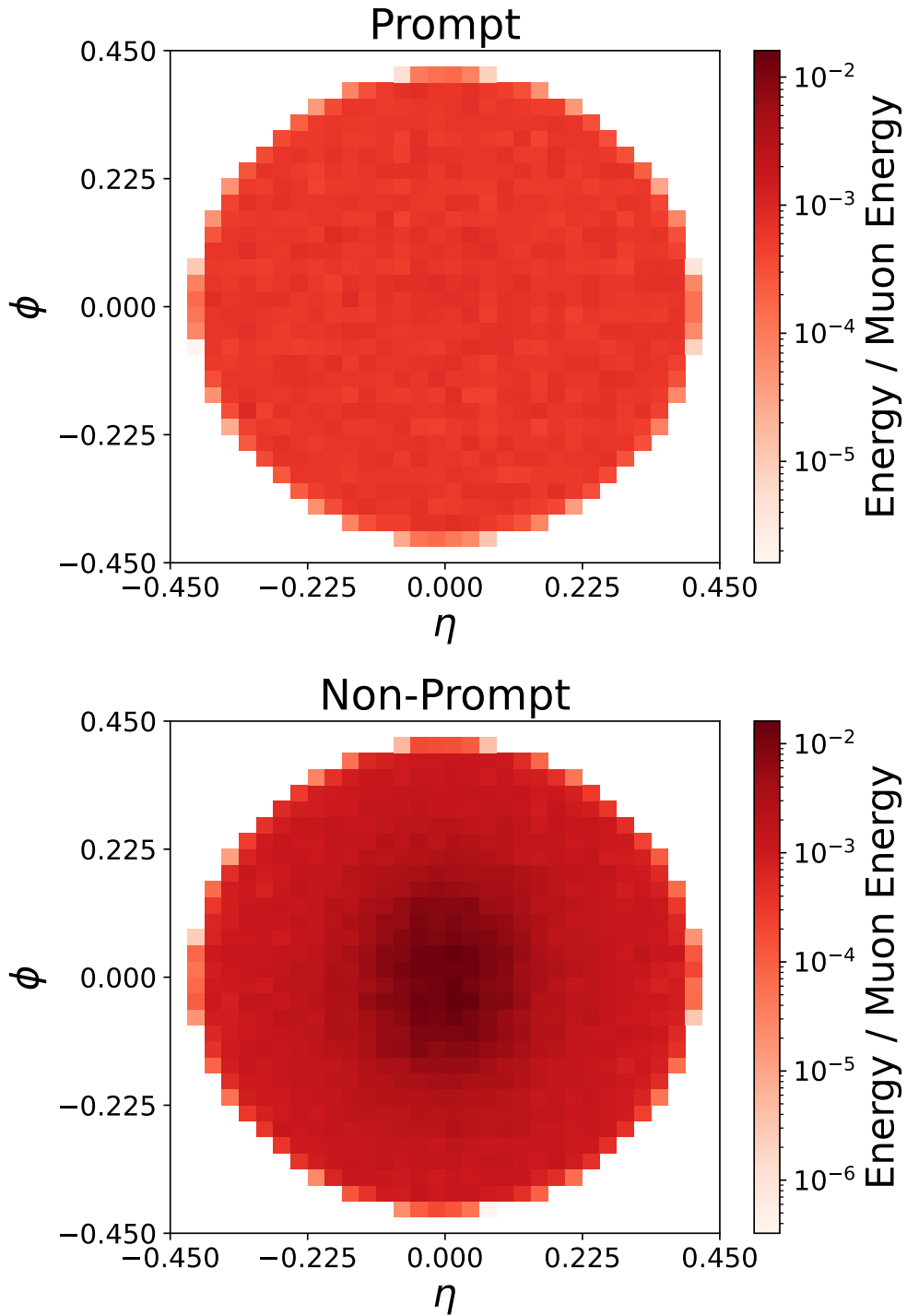


Figure B.2: (MG5+Pythia+Delphes) Average event images similar to Fig 3.3, but for the simulated dataset and separated by prompt and non-prompt events.

Using the simulated dataset, we compute one of the features included in our models which use the CMS dataset, the summed transverse momentum of the objects in an event. We see in Fig B.3 that the component distributions do approximately match across the samples for the simulated dataset. Similarly, the class components of a network classifier should be distributed the same way, regardless of which mixed sample the events were drawn from. We check this by training a PFN using the simulated dataset and looking at the distributions of the outputs, as shown in Fig B.4. Once again we see that the distributions depend on the class rather than the mixed sample to which events belong.

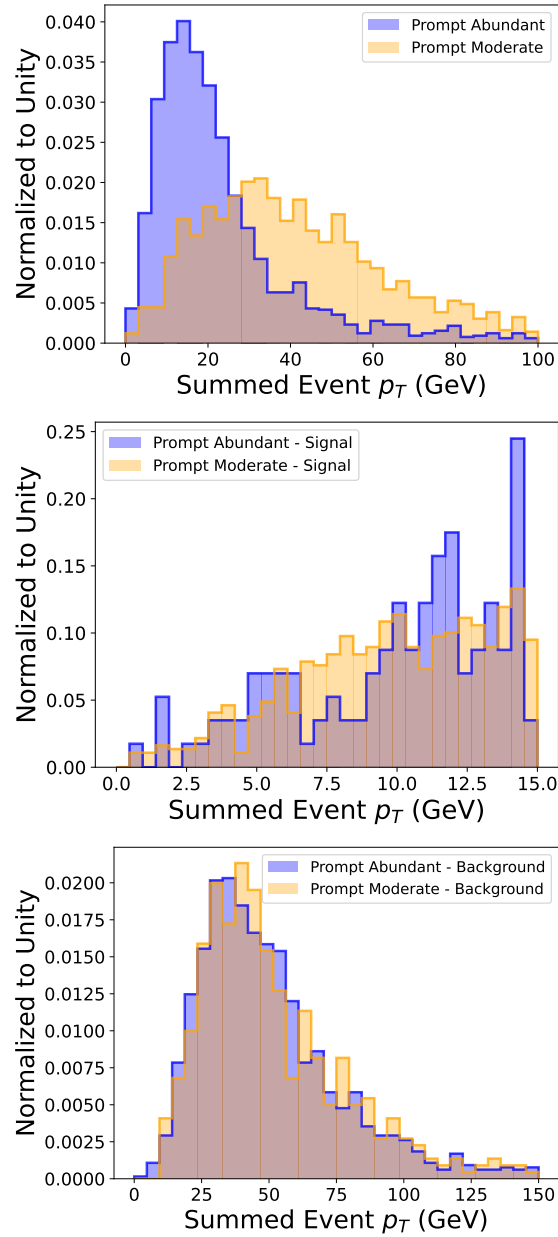


Figure B.3: (MG5+Pythia+Delphes) (Top) The total summed event p_T distributions for two simulated mixed samples. (Middle) Only the signal components of the two simulated mixed samples. (Bottom) Only the background components of the two simulated mixed samples. We see that while the class proportions are different, the signal and background distributions are approximately the same across the samples.

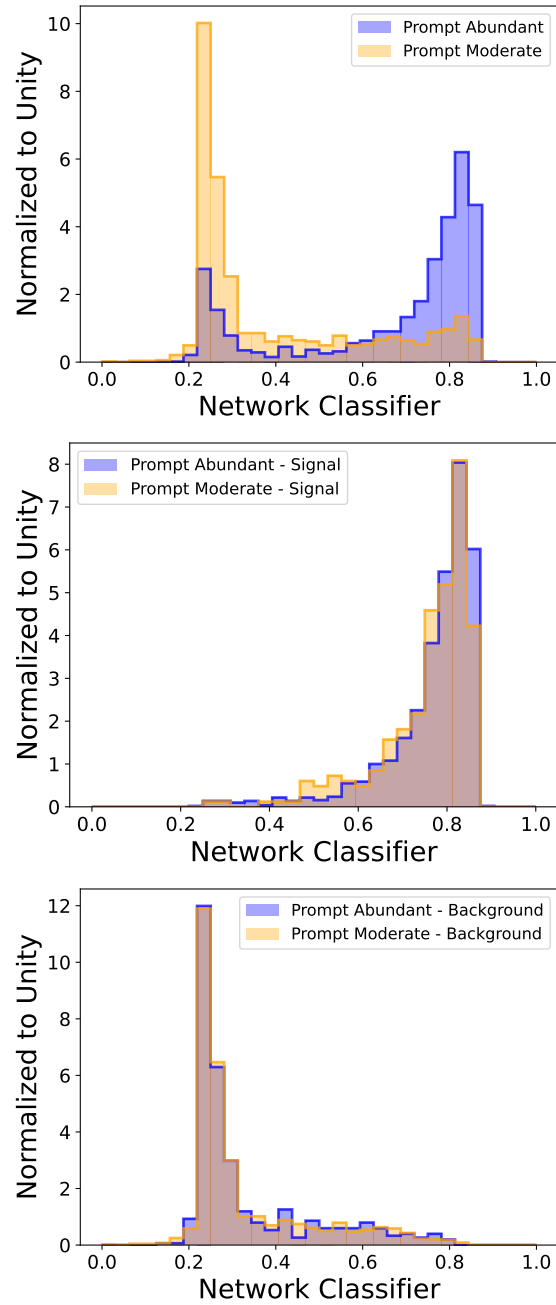


Figure B.4: (MG5+Pythia+Delphes) Similar to Fig B.3, but demonstrating that network output distributions for each class match across mixed samples.

Appendix C

Appendix: Learning Broken Symmetries with Encouraged Invariance

Signal efficiencies are tabulated in Table C.1. The results presented in these tables are illustrated in Figs. C.1 and C.2.

The signal efficiencies at fixed background efficiency are depicted in Fig. C.3 separated by network architecture and binning scheme.

The averages of each sample may be seen in Fig. C.4 with uniform binning and non-uniform binning.

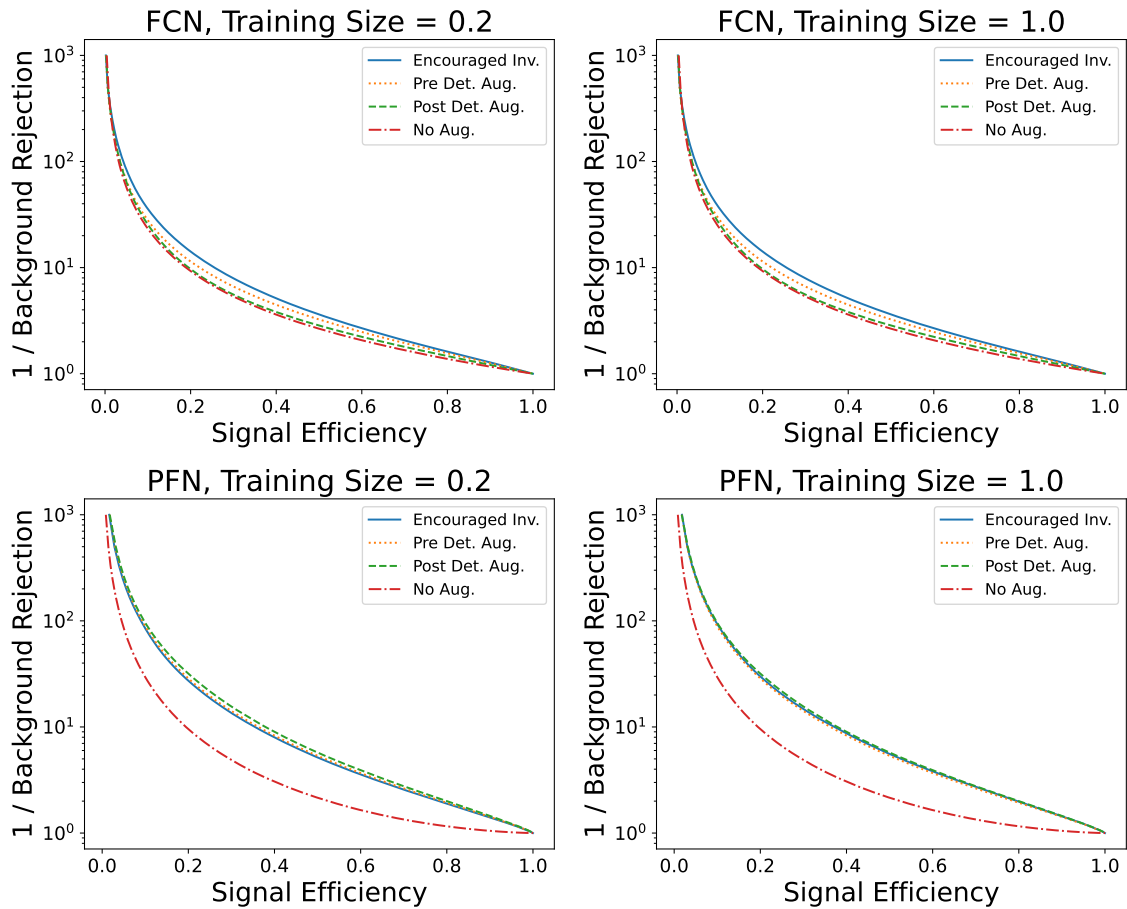


Figure C.1: Performance of FCNs (left) and PFNs (right) trained on a small training set size (top) or a large training set size (bottom) using uniformly binned data.

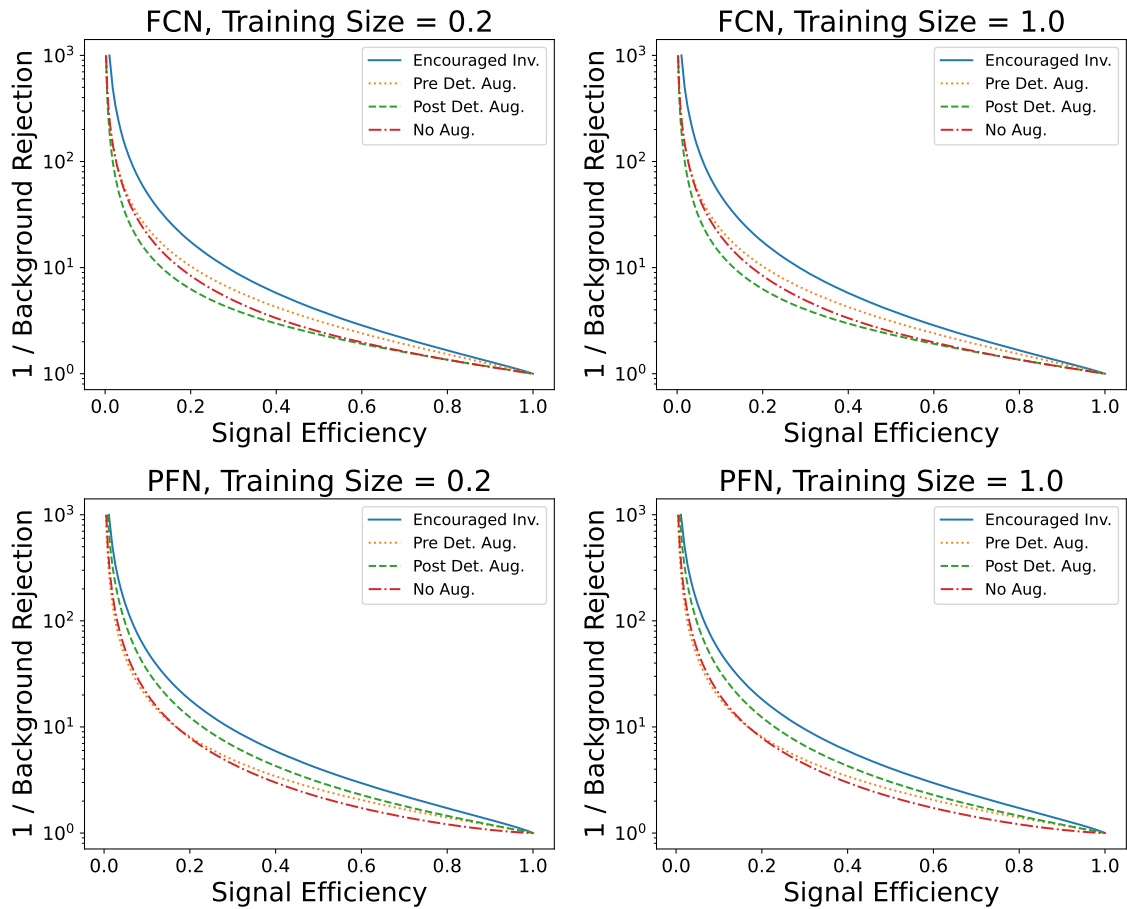


Figure C.2: Performance of FCNs (left) and PFNs (right) trained on a small training set size (top) or a large training set size (bottom) using non-uniformly binned data.

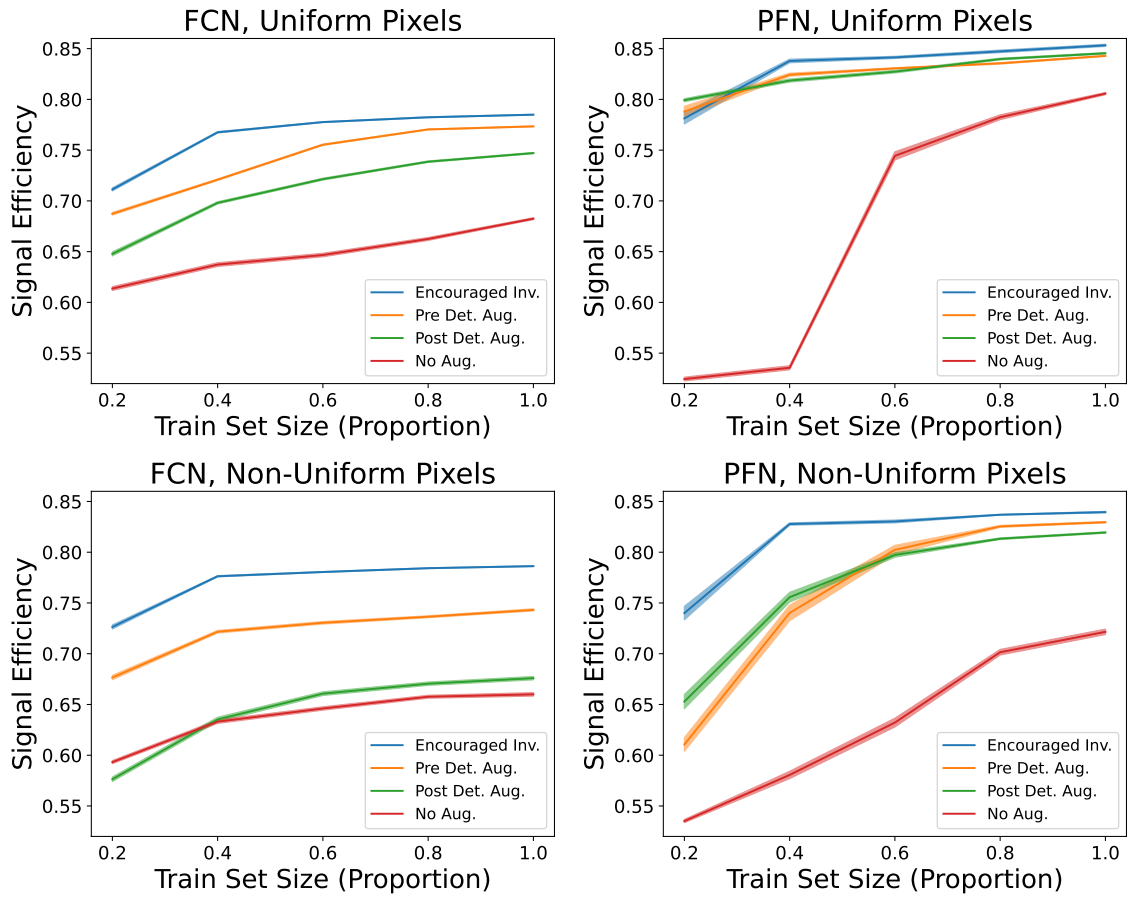


Figure C.3: Performance of FCNs (left) and PFNs (right) trained on uniformly binned data (top) or non-uniformly binned data (bottom) as a function of training set size.

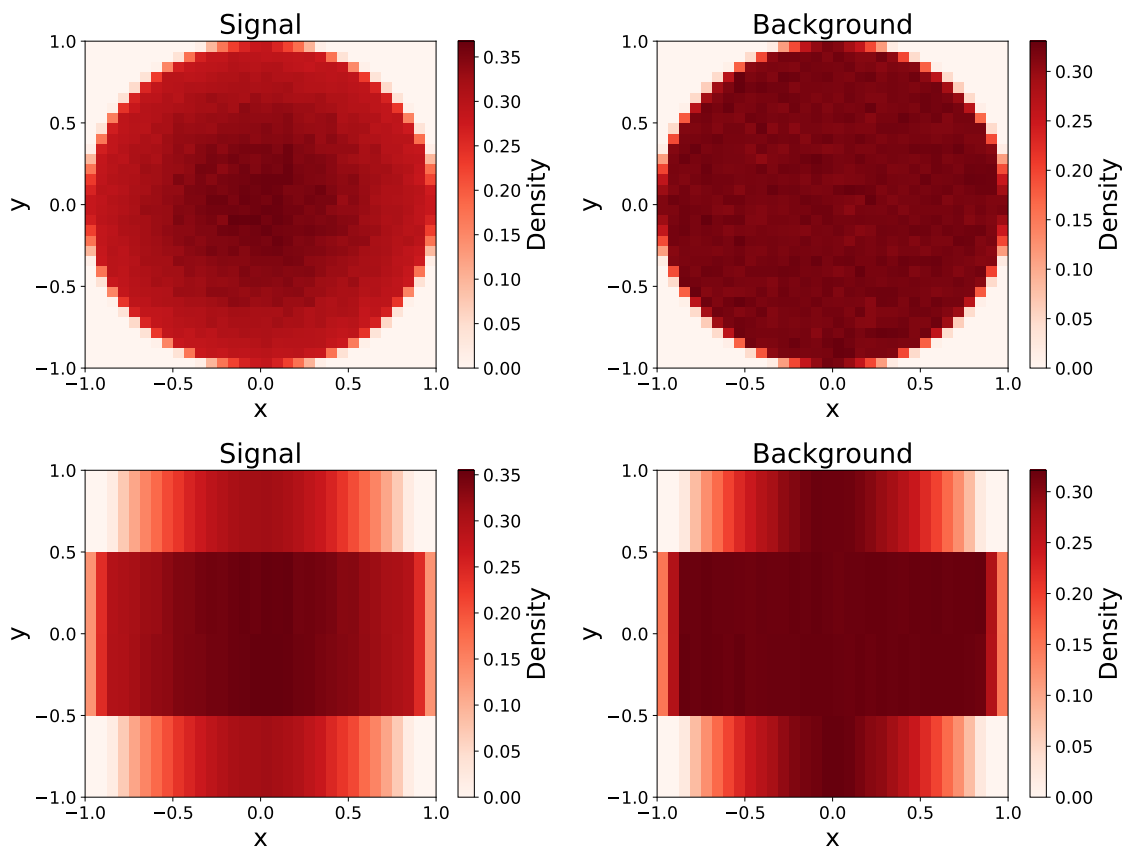


Figure C.4: Average images of the signal (Left) and background samples (Right), with uniform pixelization (Top) and non-uniform pixelization (Bottom).

Arch.	Augm.	Uniform bins		Non-uniform bins	
		small set	large set	small set	large set
FCN	None	0.614(2)	0.682(1)	0.593(1)	0.660(2)
	post-det.	0.648(2)	0.747(1)	0.576(2)	0.676(2)
	pre-det.	0.687(1)	0.773(1)	0.677(2)	0.743(1)
	enc. inv.	0.711(1)	0.785(1)	0.726(2)	0.786(1)
PFN	None	0.524(2)	0.806(1)	0.535(2)	0.722(3)
	post-det.	0.799(1)	0.845(1)	0.653(7)	0.819(1)
	pre-det.	0.788(5)	0.843(1)	0.611(7)	0.830(1)
	enc. inv.	0.781(5)	0.853(1)	0.740(7)	0.840(1)

Table C.1: A summary of the signal efficiency at a fixed background efficiency of 50%, shown for the smallest and largest training set sizes tested and both pixelization schemes.