

## **UC Irvine**

### **UC Irvine Electronic Theses and Dissertations**

#### **Title**

Machine Learning in Physics

#### **Permalink**

<https://escholarship.org/uc/item/0dv9n88t>

#### **Author**

Collado Umana, Julian

#### **Publication Date**

2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,  
IRVINE

Machine Learning in Physics

DISSERTATION

submitted in partial satisfaction of the requirements  
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Julian Collado Umana

Dissertation Committee:  
Distinguished Professor Pierre Baldi, Chair  
Professor Daniel Whiteson  
Professor Xiaohui Xie

2021

Chapter 2 © 2016 American Physical Society (APS)  
Chapter 3 © 2021 American Physical Society (APS)  
Chapter 4 © 2021 Journal of High Energy Physics (JHEP)  
All other materials © 2021 Julian Collado Umana

# DEDICATION

To my wife for supporting me so I do not fall down and when I do fall, for helping me get  
back up.

To my parents for always believing in me.

# TABLE OF CONTENTS

	Page
<b>LIST OF FIGURES</b>	<b>v</b>
<b>LIST OF TABLES</b>	<b>vii</b>
<b>ACKNOWLEDGMENTS</b>	<b>viii</b>
<b>VITA</b>	<b>x</b>
<b>ABSTRACT OF THE DISSERTATION</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Jet Flavor Classification in High-Energy Physics with Deep Neural Networks</b>	<b>5</b>
2.1 Abstract . . . . .	5
2.2 Introduction . . . . .	6
2.3 Classification and Dimensionality . . . . .	7
2.4 Data . . . . .	9
2.5 Methods . . . . .	14
2.5.1 Machine Learning Approaches . . . . .	14
2.5.2 Hardware and Software Implementations . . . . .	21
2.6 Results . . . . .	21
2.7 Discussion . . . . .	29
<b>3 Learning to Identify Electrons</b>	<b>32</b>
3.1 Abstract . . . . .	32
3.2 Introduction . . . . .	33
3.3 Overview . . . . .	34
3.4 Dataset Generation . . . . .	35
3.4.1 Processes and Simulation . . . . .	36
3.4.2 Electron Candidate Selection . . . . .	36
3.4.3 Image Formation . . . . .	37
3.5 Standard Classification Features . . . . .	38
3.5.1 Lateral Shower Extension: $\sigma_{\eta\eta}$ . . . . .	41
3.5.2 Isolation . . . . .	41

3.6	Neural Network Architectures and Training . . . . .	41
3.7	Performance . . . . .	43
3.8	Bridging the gap . . . . .	44
3.8.1	Set of Observables . . . . .	45
3.8.2	Searching for Observables . . . . .	46
3.8.3	IRC safe observables . . . . .	47
3.8.4	Broader Scan . . . . .	49
3.9	Discussion . . . . .	50
3.10	Neural Network Hyperparameters and Architecture . . . . .	53
<b>4</b>	<b>Learning to Isolate Muons</b>	<b>55</b>
4.1	Abstract . . . . .	55
4.2	Introduction . . . . .	56
4.3	Approach and Dataset . . . . .	57
4.3.1	Data generation . . . . .	58
4.4	Networks and Performance . . . . .	59
4.5	Analysis . . . . .	64
4.5.1	Search Strategy . . . . .	64
4.5.2	IRC Safe Observables . . . . .	66
4.5.3	IRC-unsafe Observables . . . . .	68
4.6	Discussion . . . . .	69
4.7	Conclusions . . . . .	70
<b>5</b>	<b>Conclusion</b>	<b>72</b>
	<b>Bibliography</b>	<b>74</b>
	<b>Appendix A Learning to Identify Muons Appendix</b>	<b>81</b>

# LIST OF FIGURES

		Page
2.1	Distributions in simulated samples of high-level jet flavor variables widely used to discriminate between jets from light-flavor and heavy-flavor quarks. . . . .	13
2.2	Top: Distribution of the number of tracks associated to a jet in simulated samples. Bottom: Distribution of the number of vertices associated to a jet in simulated samples, before and after removing tracks which exceed the maximum allowed value of 15. . . . .	15
2.3	Feedforward neural network architecture. In the first layer, connections of the same color represent the same value of the shared weight. The others layers are fully connected without shared weights. . . . .	17
2.4	Architecture of the Long Short Term Memory networks as described in the text.	19
2.5	Architecture of the outer recursive networks as described in the text. . . . .	20
2.6	Signal efficiency versus background rejection (inverse of efficiency) for deep networks trained on track-level, vertex-level or expert-level features. The top pane shows the performance for $b$ -quarks versus light-flavor quarks, the bottom pane for $b$ -quarks versus $c$ -quarks. . . . .	24
2.7	Signal efficiency versus minimum jet $p_T$ relative to light quarks (top) or charm quarks (bottom). In each case, efficiency is shown for fixed values of background rejection for networks trained with only expert features or networks trained with all features (tracks, vertices and expert features). . . . .	25
2.8	Signal efficiency versus minimum jet pseudo-rapidity relative to light quarks (top) or charm quarks (bottom). In each case, efficiency is shown for fixed values of background rejection for networks trained with only expert features or networks trained with all features (tracks, vertices and expert features). . . . .	26
2.9	Rejection of light quarks (top) or charm quarks (bottom) versus minimum jet $p_T$ . In each case, rejection is shown for fixed values of signal efficiency for networks trained with only expert features or networks trained with all features (tracks, vertices and expert features). . . . .	27
2.10	Rejection of light quarks (top) or charm quarks (bottom) versus minimum jet pseudo-rapidity. In each case, rejection is shown for fixed values of signal efficiency for networks trained with only expert features or networks trained with all features (tracks, vertices and expert features). . . . .	28

2.11	Distributions of expert-level features for heavy-flavor and light-flavor classes. Also shown are distributions of light-flavor and charm jets surviving network threshold selections chosen to give rejection of 10 and 50, for networks using only expert information and networks using expert information in addition to lower-level information. . . . .	29
3.1	Distribution of generated electron candidate $p_T$ and $\eta$ for simulated signal and background samples, before reweighting to match spectra. . . . .	37
3.2	Images in the electromagnetic calorimeter for signal electrons (top) and background jets (bottom). On the left are individual examples, on the right are mean images. See Fig. 3.3 for corresponding hadronic calorimeter images. . .	38
3.3	Images in the hadronic calorimeter for signal electrons (top) and background jets (bottom). On the left are individual examples, on the right are mean images. See Fig. 3.2 for corresponding electromagnetic calorimeter images. . .	39
3.4	Distribution of signal electron (red) and background jets (blue) for seven existing typically-used high-level features, as well as for mass. . . . .	42
3.5	Comparison of the performance in electron identification for networks with varying sets of input features. Shown is the signal efficiency versus background rejection, and the AUC, for networks which use the existing set of expert high-level features (see text for details), networks which use HCal or ECal images, or both. . . . .	43
3.6	$\log_{10}$ distributions of the selected IRC-safe EFPs as chosen by the black-box guided strategy, for signal electrons and background jets. . . . .	49
3.7	$\log_{10}$ distributions of the selected EFPs as chosen by the black-box guided strategy, regardless of IRC safety, for signal electrons and background jets. . .	50
3.8	Diagram of the architecture of the convolutional neural network. . . . .	54
3.9	Diagram of convolutional block appearing in network architecture, see Fig 3.8.	54
4.1	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 $R = 0.4$ . The color of each cell represents the sum of the $E_T$ of the calorimeter deposits within the cell. . . . .	60
4.2	Comparison of classification performance using the performance metric AUC between Particle-Flow networks trained on unordered lists of calorimeter deposits (orange, 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. . . . .	62
4.3	Background rejection versus signal efficiency for Particle-Flow networks trained on unordered lists of calorimeter deposits (orange, solid), 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). . . . .	63
4.4	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. . . . .	68



# LIST OF TABLES

	Page
2.1 Performance results for networks using track-level, vertex-level or expert-level information. In each case the jet $p_T$ and pseudorapidity are also used. Shown for each method is the Area Under the Curve (AUC), the integral of the background efficiency versus signal efficiency, which have a statistical uncertainty of 0.001 or less. Signal efficiency and background rejections are shown in Figs. 2.6-2.10. . . . .	23
3.1 Electron classification power (AUC) for networks with various feature sets. Images refer to low-level pixel data. Standard features are the high-level (HL) features typically used ( $R_{\text{had}}, \omega_{\eta 2}, R_\phi, R_\eta, \sigma_{\eta\eta}, \text{Iso}(\Delta R < 0.3), \text{Iso}(\Delta R < 0.4)$ ), as described in the text. All AUC values have an uncertainty of $\pm 0.001$ unless otherwise specified. . . . .	44
3.2 Summary of the performance of various networks considered. Uncertainty in the AUC value is $\pm 0.001$ , estimated using bootstrapping. . . . .	50
3.3 Hyperparameter ranges for bayesian optimization of convolutional networks .	53
3.4 Hyperparameter ranges for bayesian optimization of fully connected networks	53
3.5 Best hyperparameters found per model. . . . .	53
4.1 Summary of performance (AUC) in the prompt muon classification task for various network architectures and input features. Statistical uncertainty in each case is $\pm 0.001$ with 95% confidence, measured using bootstrapping over 200 models. Uncertainty due to the initial conditions of the network is found to be negligible. . . . .	69

# ACKNOWLEDGMENTS

For chapter 2, I would like to thank my co-authors Daniel Guest, Shih-Chieh Hsu and Gregor Urban, Pierre Baldi and Daniel Whiteson for their contribution to this work. This chapter is a reprint of the material as it appears in Physics Review D (PRD) as "Jet Flavor Classification in High-Energy Physics with Deep Neural Networks". Pierre Baldi and Daniel Whiteson are co-authors listed in these publications which directed and supervised research which forms the basis for the thesis/dissertation. Additionally I would like to thank David Kirkby, Gordon Watts, Shimon Whiteson, David Casper, and Kyle Cranmer for useful comments and helpful discussion. I would like to thank Yuzo Kanomata for computing support. I also wish to acknowledge a hardware grant from NVIDIA and NSF grant IIS-1321053 to PB.

For chapter 3, I would like to thank my co-authors Jessica N. Howard, Taylor Faucett, Tony Tong, Daniel Whiteson and Pierre Baldi for their contribution to this work. This chapter is a reprint of the material which has been submitted to Physics Review D (PRD) as "Learning to Identify Electrons". Pierre Baldi and Daniel Whiteson are co-authors listed in these publications which directed and supervised research which forms the basis for the thesis/dissertation. Additionally I would like to thank Jesse Thaler, Ian Moulton and Tim Tait for helpful discussions. I wish to acknowledge a hardware grant from NVIDIA. This material is based upon work supported by the National Science Foundation under grant number 1633631. The work of JC and PB is in part supported by grants NSF 1839429 and NSF NRT 1633631 to PB. The work of JNH is in part supported by grants DE-SC0009920, DGE-1633631, and DGE-1839285. TT wants to thank Undergraduate Research Opportunities Program at UCI for grant number 02399s1.

For chapter 4, I would like to thank my co-authors Kevin Bauer, Edmund Witkowski, Taylor Faucett, Daniel Whiteson and Pierre Baldi for their contribution to this work. This chapter is a reprint of the material which has been submitted to Journal of High Energy Physics (JHEP) as "Learning to Isolate Muons". Pierre Baldi and Daniel Whiteson are co-authors listed in these publications which directed and supervised research which forms the basis for the thesis/dissertation. Additionally I would like to thank Michael Fenton, Dan Guest and Jesse Thaler for providing valuable feedback and insightful comments and Yuzo Kanomata for computing support. We also wish to acknowledge a hardware grant from NVIDIA. This material is based upon work supported by the National Science Foundation under grant number 1633631. DW is supported by the DOE Office of Science. The work of JC and PB in part supported by grants NSF 1839429 and NSF NRT 1633631 to PB.

I would like to thank the American Physical Society (APS), owner of the journal Physics Review D, for allowing me to use the articles or a portion of the articles published in their journals, in a thesis or dissertation without requesting permission from APS, provided the bibliographic citation and the APS copyright credit line are given on the appropriate pages.

I would like to thank the International School for Advanced Studies (SISSA - Trieste, Italy), owner of the Journal of High Energy Physics (JHEP), for allowing me to include the Article (all or part) in a research thesis or dissertation, in a thesis or dissertation.

I would like to deeply thank my advisor Pierre Baldi who chose to take a chance and give me an opportunity to prove myself in my graduate studies. This opportunity has completely changed my life and I will always be grateful for it.

I would like to thank my co-advisor Daniel Whiteson for going above and beyond in terms of advice and support.

I would like to thank Peter Sadowski for his outstanding help and support, specially during my first year at UCI.

I would also like to thank the MAPS: Machine Learning and Physical Sciences National Science Foundation/UCI Graduate Training Program (grant number 1633631) for funding, training and support during my PhD.

# VITA

**Julian Collado Umana**

## EDUCATION

**Doctor of Philosophy in Computer Science** **2021**  
University of California Irvine *Irvine, California*

**Bachelor in Physics** **2013**  
University of Costa Rica *Montes de Oca, San Jose*

## RESEARCH EXPERIENCE

**Graduate Research Assistant** **2014–2021**  
University of California, Irvine *Irvine, California*

## TEACHING EXPERIENCE

**Teaching Assistant** **2014–2021**  
University of California Irvine *Irvine, California*

## INDUSTRY EXPERIENCE

**Data Scientist Intern** **Summer 2019**  
Electronic Arts *Redwood City, California*

**Data Analyst Intern** **Summer 2018**  
Blizzard Entertainment *Irvine, California*

**Data Scientist Intern** **Summer 2017**  
Blackberry-Cylance *Irvine, California*

**Engineering Intern** **Summer 2016**  
Canon Information and Imaging Solutions *Irvine, California*

**REFEREED JOURNAL PUBLICATIONS** **Deep learning, dark knowledge, and dark matter**  
Journal of Machine Learning Research

**Jet flavor classification in high-energy physics with deep neural networks** 2016

Physics Review D

**Sherpa: Robust hyperparameter optimization for machine learning** 2020

SoftwareX

**Sparse Autoregressive Model for Scalable Generation of Sparse Images in Particle Physics** 2021

Physics Review D

**REFEREED CONFERENCE PUBLICATIONS**

**Convolutional Neural Networks for Energy and Vertex Reconstruction in DUNE** Dec 2019

33rd Conference on Neural Information Processing Systems (NeurIPS), Machine Learning and the Physical Sciences Workshop

**Sparse Image Generation with Decoupled Generative Models** Dec 2019

33rd Conference on Neural Information Processing Systems (NeurIPS), Machine Learning and the Physical Sciences Workshop

**Deep-Learning-Based Kinematic Reconstruction for DUNE** Dec 2020

34th Conference on Neural Information Processing Systems (NeurIPS), Machine Learning and the Physical Sciences Workshop

**SOFTWARE**

**Sherpa** <https://github.com/sherpa-ai/sherpa>  
*Python Hyperparameter Optimization Library.*

# ABSTRACT OF THE DISSERTATION

Machine Learning in Physics

By

Julian Collado Umana

Doctor of Philosophy in Computer Science

University of California, Irvine, 2021

Distinguished Professor Pierre Baldi, Chair

What is the universe made of? This is the core question particle physics aims to answer by studying the fundamental blocks of the universe. To study these blocks we require colliding particles at approximately the speed of light which produces high dimensional data in the order of peta-bytes per second, presenting considerable challenges in data processing and analysis. In order to validate or refute physical theories, it is necessary to distinguish the particles created in the collisions from the background noise. The data is processed through a complex pipeline with multiple non-interpretable data representations like images, sequences and graphs, at each level of processing. At the end of the pipeline there is a set of interpretable high-level features created using physics-motivated heuristics, which are analyzed using conventional statistical methods to make a classification. The multiple levels of data processing and representations opens the possibility of using techniques like deep learning to obtain improvements which in time will enable new discoveries in particle physics.

In this thesis, we show it is possible to bypass the dimensionality reduction step of traditional methods by using deep learning directly in the low-level detector data. This approach outperforms the-state-of-the-art methods in particle physics problems such as jet flavor classification, electron classification, and muon classification by 1.6%, 3.0% and 8.7% respectively. In addition, we show it is possible to achieve this performance using neural networks while

maintaining the interpretability of high-level features, by using a recently developed technique to map the deep network into a space of physically interpretable high-level features that reproduce the performance of the deep network.

# Chapter 1

## Introduction

Particle identification at high-energy collisions provides an essential evidence for precision studies of the Standard Model [5, 12] as well as for searches for new physics [3, 32]. Being able to identify particles of interest and separate them from their background is a critical element of the data analysis toolkit for particle physics. This separation allows more precise measurements particle properties, which in turn can help refute or validate physical theories. However, high-energy collision experiments produce gargantuan amounts of data with high dimensionality, making particle classification very difficult.

Traditional methods aim to reduce the dimensionality of the problem by using physics-motivated heuristics in order to simplify the representation of the data, and afterwards use machine learning techniques to classify the particles. Advances in machine learning such as the development of deep learning methods have shown promising pathways to improve classification performance in areas like computer vision [78, 57]. Some data in high-energy physics can be represented as an image or a sequence, which opens the possibility of using deep learning methods in these datasets [23, 66]. Since deep learning methods can take advantage of large amounts of data and in many cases are able to surpass the performance



of human-designed features, they are a promising option to complement or surpass the performance of traditional methods in high-energy physics.

In this thesis, I work with an interdisciplinary team of computer scientists, statisticians and physicists to apply and develop new deep learning models for high-energy physics. The common methodology throughout this work will be going directly to the low-level detector representation of the data and using custom deep learning models to improve the state-of-the-art results and afterwards try to understand what the network is learning.

However, deep neural networks work as a black box, which means the improvement in classification performance comes at the cost of interpretability compared to the physics-inspired heuristics of traditional methods. To be able to understand the nature of the information being used by the network, we analyze the performance of several models at different levels of data processing and from different sensors. In addition, we use energy flow polynomials (EFPs) [67] to create an infinite set of physically interpretable high-level variables from low-level detector information. We then use another recent method which uses neural networks to reduce this infinite set to a tractable number of variables which capture the performance of the black box network trained on low-level information [46]. Our results provide evidence that there is no need to sacrifice interpretability in order to achieve high performance with neural networks in high-energy physics. Specifically, we found that in some cases it is possible to completely match the performance of the network using interpretable features, while in others we achieve a boost in performance but are not able to match the black box model. This suggests the need of more complex variables beyond energy flow polynomials.

In the following chapters, we will show evidence of the strong potential of deep neural networks for high-energy physics applications. We will also show how in some cases it is possible to use these models to recover interpretable variables useful for physical theories.

In chapter 2, we do a jet flavor classification study in which we show it is possible to surpass

the performance of traditional classification methods by using deep learning models directly in lower-level detector information. This suggests the dimensionality reduction performed by the traditional methods is sacrificing or distorting crucial information in the data. Furthermore, by doing an analysis at different levels of complexity, we are able to quantify the amount of information lost at each level of data processing. Finally, our results show the best performance is obtained when we combine the deep learning model using low-level detector information with the high level physics-inspired heuristics. While in principle all of the information exists in the lowest-level features and it should be possible to train a network which matches or exceeds this performance without expert knowledge, this is neither necessary nor desirable. Expert knowledge exists and is well-established, and there is no reason to discard it.

In chapter 3, in a study to distinguish electrons from jet background, we show traditional methods are overlooking important information that can be exploited by deep learning methods used directly in electromagnetic and hadronic calorimeter deposits. We then use a recently developed technique to map the deep network into a space of physically interpretable variables [67][46]. We show that by using a small number of these new variables in addition to the ones used by the traditional method we are able to close most of the gap between the traditional method and the neural network. This provides evidence that it is not strictly necessary to sacrifice interpretability to gain performance while using neural networks.

In chapter 4, we perform a study to distinguish prompt muons produced in heavy boson decay and muons produced in association with heavy-flavor jet production. Similarly to chapter 3, we want to study if there is additional information in the calorimeter deposits that is currently not captured by traditional methods. We trained four types of deep learning models; dense neural networks on traditional interpretable high-level variables, convolutional neural networks on calorimeter images, energy flow networks and particle flow networks on unordered sets of calorimeter deposits. All deep learning models surpass the performance

of traditional methods providing critical improvements to muon classification techniques. In addition we use the same method as in chapter 3 to create new interpretable variables and map the network performance to this space. We found that while the new variables were not able to fully close the gap, an analysis of the new variables and the networks provides crucial insights on what type of information needs to be added to the traditional method and where the current method for creating new variables could be expanded.

# Chapter 2

## Jet Flavor Classification in High-Energy Physics with Deep Neural Networks

### 2.1 Abstract

Classification of jets as originating from light-flavor or heavy-flavor quarks is an important task for inferring the nature of particles produced in high-energy collisions. The large and variable dimensionality of the data provided by the tracking detectors makes this task difficult. The current state-of-the-art tools require expert data-reduction to convert the data into a fixed low-dimensional form that can be effectively managed by shallow classifiers. We study the application of deep networks to this task, attempting classification at several levels of data, starting from a raw list of tracks. We find that the highest-level lowest-dimensionality expert information sacrifices information needed for classification, that the performance of current state-of-the-art taggers can be matched or slightly exceeded by deep-network-based

taggers using only track and vertex information, that classification using only lowest-level highest-dimensionality tracking information remains a difficult task for deep networks, and that adding lower-level track and vertex information to the classifiers provides a significant boost in performance compared to the state-of-the-art.

## 2.2 Introduction

The search for new particles and interactions at the energy frontier is a rich program with enormous discovery potential. The power to discover this hypothetical new physics relies crucially on the ability to infer the nature of the interaction and the particles produced from the data provided by the detectors which surround the point of collision. One critical element is jet flavor classification, the distinction between hadronic jets produced from light-flavor and heavy-flavor quarks. Such classification plays a central role in identifying heavy-flavor signals and reducing the enormous backgrounds from light-flavor processes [4, 8].

Jets originating from heavy-flavor quarks tend to produce longer-lived particles than those found in jets from light-flavor quarks; these long-lived particles have decays which are displaced from the primary vertex. To identify such vertices, the central tracking chamber measures the trajectories of charged particles which allows for the reconstruction of vertex locations. The large and varying number of particles in a jet leads to a difficult classification problem with large and variable dimensionality without a natural ordering. The first step in typical approaches involves vertex-finding algorithms [85], which transform the task into one of reduced, but still variable, dimensionality. Finally, most state-of-the-art jet flavor classification tools used by experiments [7, 31] rely heavily on expert-designed features which fix and further reduce the dimensionality before applying shallow machine-learning techniques. Such techniques have excellent performance, but are primarily motivated by historical limitations in the ability of shallow learning methods to handle high- and variable-dimensionality

datasets.

Recent applications of deep learning to similar problems in high-energy physics [23, 24, 75, 18], combined with the lack of a clear analytical theory to provide dimensional reduction without loss of information, suggests that deep learning techniques applied to the lower-level higher-dimensional data could yield improvements in the performance of jet-flavor classification algorithms. General methods for designing and applying recurrent and recursive neural networks to problems with data of variable size or structure have been developed in Refs. [20, 52, 47, 48, 21], and applied systematically to a variety of problems ranging from natural language processing [80], to protein structure prediction [19, 83, 44, 72] to prediction of molecular properties [71, 45] and to the game of go [86]; previous studies have discussed the extension of such strategies to tasks involving tracks in high energy physics [40, 15].

In this paper, we apply several deep learning techniques to this problem using a structured dataset with features at three levels of processing (tracks, vertices, expert), each of which is a strict function of the previous level(s). The data at the highest level of processing, with smallest dimensionality, is intended to mirror the typical approach used currently by experimental collaborations. The multi-layered structure of the dataset allows us to draw conclusions about the information loss at each stage of processing, and to gauge the ability of machine learning tools to find solutions in the lower- and higher-dimensional levels. These lessons can guide the design of flavor-tagging algorithms used by experiments.

## 2.3 Classification and Dimensionality

The task of the machine learning (ML) algorithm is to identify a function  $f(\bar{x}) : \mathbb{R}^N \rightarrow \mathbb{R}^1$  whose domain is the observed data at some level of processing (with potentially very large dimensionality  $N$ ) and which evaluates to a single real value that contains the information

necessary to perform the classification. Perfect classification is not expected; instead, the upper bound is performance which matches classification provided by the true likelihood ratio between heavy-flavor ( $b$ ) and light-flavor quarks ( $q$ ):  $P(\bar{x}|b)/P(\bar{x}|q)$  evaluated in the high-dimensional domain.

Though we lack knowledge of an analytical expression for the likelihood, in principle one could recover such a function from labeled datasets with trivial algorithms, by estimating the likelihood directly in the original high-dimensional space. In practice, this requires an enormous amount of data, making it impractical for problems with anything but the smallest dimensionality in their feature space.

Machine learning plays a critical role in approximating the function  $f(\bar{x})$  which reduces the dimensionality of the space to unity by finding the critical information needed to perform the classification task. Such a function may disregard some of the information from the higher-dimensional space if it is not pertinent to the task at hand. However, for very high dimensional spaces (greater than  $\approx 50$ ), the task remains very difficult, and until the recent advent of deep learning it appeared to be overwhelming, though it can still require the generation of large samples of training data.

It would be very powerful to compare the performance of a given solution to the theoretical upper limit on performance, provided by the true likelihood. Unfortunately, without knowledge of the true likelihood, it is difficult to assess how well the ML algorithm has captured the necessary information. For this reason, in the studies presented here and in earlier work [23, 24, 18], we built structured datasets with at least two levels of dimensionality: an initial sample with lower-level data at high dimensionality and a reduced sample with expert features at lower dimensionality. Importantly, the expert features are a strict function of the lower-level features, so that they contain a subset of the information. The expertise lies solely in the design of the dimensionality-reducing function, without providing any new information.

This structure allows us to draw revealing conclusions about the information content of the intermediate and expert-level information and the power of classifiers to extract it. Since the higher-level data contains a subset of the information and benefits from expert knowledge, it can provide the basis for a performance benchmark for the tools using lower-level data in place of the unknown true likelihood. Therefore, if the performance of tools using lower-level data fails to match that of tools using the higher-level data (or a combination of both kinds of data), then we may conclude that the tools using the lower-level data have failed to extract the complete information. On the other hand, if the performance of tools using lower-level data exceeds that of tools using the higher-level data, then we may conclude that the higher-level data does not contain all of the information relevant to the classification task, or that it has transformed the problem into a more difficult learning task for the algorithms considered. Regardless of the reason, in this case the transformation to the higher-level lower-dimensional data has failed in its goal.

## 2.4 Data

Training samples were produced with realistic simulation tools widely used in particle physics. Samples were generated for three classes of jet:

- light-flavor: jets from  $u, d, s$  quarks or gluons;
- charm: jets from  $c$  quarks;
- heavy-flavor: jets from  $b$  quarks.

Collisions and immediate decays were generated with MADGRAPH5 [13] v2.2.3, showering and hadronization simulated with PYTHIA [79] v6.428, and response of the detectors simulated with DELPHES [39] v3.2.0. Studies with additional  $pp$  interactions (*pileup*) are reserved for



future work; here we assume that pileup effects will not alter the relative performance of the different methods, and is not likely to have a large impact at luminosities recorded to date, given effective techniques to isolate pileup tracks and vertices from the vertices of interest to this study.

The detector simulation was augmented with a simple tracking model that smears truth particles to yield tracks similar to those expected at ATLAS [2]. Tracks follow helical paths in a perfectly homogeneous 2 T magnetic field. No attempt was made to account for material interactions or remove strange hadrons. As a result the tracking model lacks the sophistication of models developed by LHC collaborations while retaining enough realism to run vertex reconstruction and compare the relative performance of various machine learning approaches.

Jets are reconstructed from calorimeter energy deposits with the anti- $k_T$  clustering algorithm [28] as implemented in FastJet [29], with a distance parameter of  $R = 0.4$ . Tracks are assigned to jets by requiring that they be within a cone of  $\Delta R \equiv (\Delta\eta^2 + \Delta\phi^2)^{1/2} < 0.4$  of the jet axis. Jets are labeled by matching to partons within a cone of  $\Delta R < 0.5$ . If a  $b$  or  $c$  quark is found within this cone the jet is labeled heavy or charm flavor respectively, with  $b$  taking precedence if both are found. Otherwise the jet is labeled light-flavor.

To reconstruct secondary vertices, we use the adaptive vertex reconstruction algorithm implemented in RAVE v6.24 [85, 84]. The algorithm begins by fitting a primary vertex to the event and removing all compatible tracks. For each jet, secondary vertices are then reconstructed iteratively: a vertex is fit to a point that minimizes  $\chi^2$  with respect to all tracks in the jet, less compatible tracks are down-weighted, and the vertex fit is repeated until the fit stabilizes.

Since a  $b$ -hadron decay typically cascades through a  $c$ -hadron, jets may include multiple secondary vertices. To account for this, tracks with large weights in the secondary vertex fit

are removed and the fit is repeated with the remaining tracks. The process repeats until all tracks are assigned to a secondary vertex.

As described earlier, we organize the information in three levels of decreasing dimensionality and increasing pre-processing using expert knowledge where each level is a strict function of the lower-level information. The classification is done per-jet rather than per-event, and at every level the transverse momentum and pseudorapidity of the jet is included.

The lowest-level information considered is the list of reconstructed tracks. Each helical track has five parameters in addition to a  $5 \times 5$  symmetric covariance matrix with 15 independent entries. The number of tracks varies from 1 to 33 in these samples, with a mean of 4.

The intermediate-level information comes from the output of the vertexing algorithm. The features are the vertex mass, number of tracks associated to the vertex, the fraction of the total energy in jet tracks which is associated to those tracks, vertex displacement, vertex displacement significance, and angular separation in  $\Delta\eta$  and  $\Delta\phi$  with respect to the jet axis for each vertex. In cases where both low and intermediate level features are used the track to vertex association weight is also included. The number of vertices varies from 1 to 13 in these samples, with a mean of 1.5.

The highest-level information is designed to model the typical features used in current experimental applications; see Fig. 2.1 for distributions of these features for each jet class. There are fourteen such features:

- The  $d_0$  and  $z_0$  significance of the 2nd and 3rd tracks attached to a vertex, ordered by  $d_0$  significance.
- The number of tracks with  $d_0$  significance greater than  $1.8\sigma$ .
- The JETPROB [35] light jet probability, calculated as the product over all tracks in the jet of the probability for a given track to have come from a light-quark jet.

- The width of the jet in  $\eta$  and  $\phi$ , calculated for  $\eta$  as

$$\left( \frac{\sum_i p_{Ti} \Delta\eta_i^2}{\sum_i p_{Ti}} \right)^{1/2}$$

and analogously for  $\phi$ .

- The combined vertex significance,

$$\frac{\sum_i d_i / \sigma_i^2}{\sqrt{\sum_i 1 / \sigma_i^2}}$$

where  $d$  is the vertex displacement and  $\sigma$  is the uncertainty in vertex position along the displacement axis.

- The number of secondary vertices.
- The number of secondary-vertex tracks.
- The angular distance  $\Delta R$  between the jet and vertex.
- The decay chain mass, calculated as the sum of the invariant masses of all reconstructed vertices, where particles are assigned the pion mass.
- The fraction of the total track energy in the jet associated to secondary vertices <sup>1</sup>

The dataset consists of 10 million labeled simulated jets. The corresponding target labels are “light-flavor”, “charm”, and “heavy-flavor”. The data contains 44, 11, 45 percent of each class respectively. This data is available from the UCI Machine Learning in Physics Web portal at <http://mlphysics.ics.uci.edu/>.

---

<sup>1</sup>The vertex energy fraction is not a strict fraction; it can be greater than unity if tracks are assigned to multiple vertices.

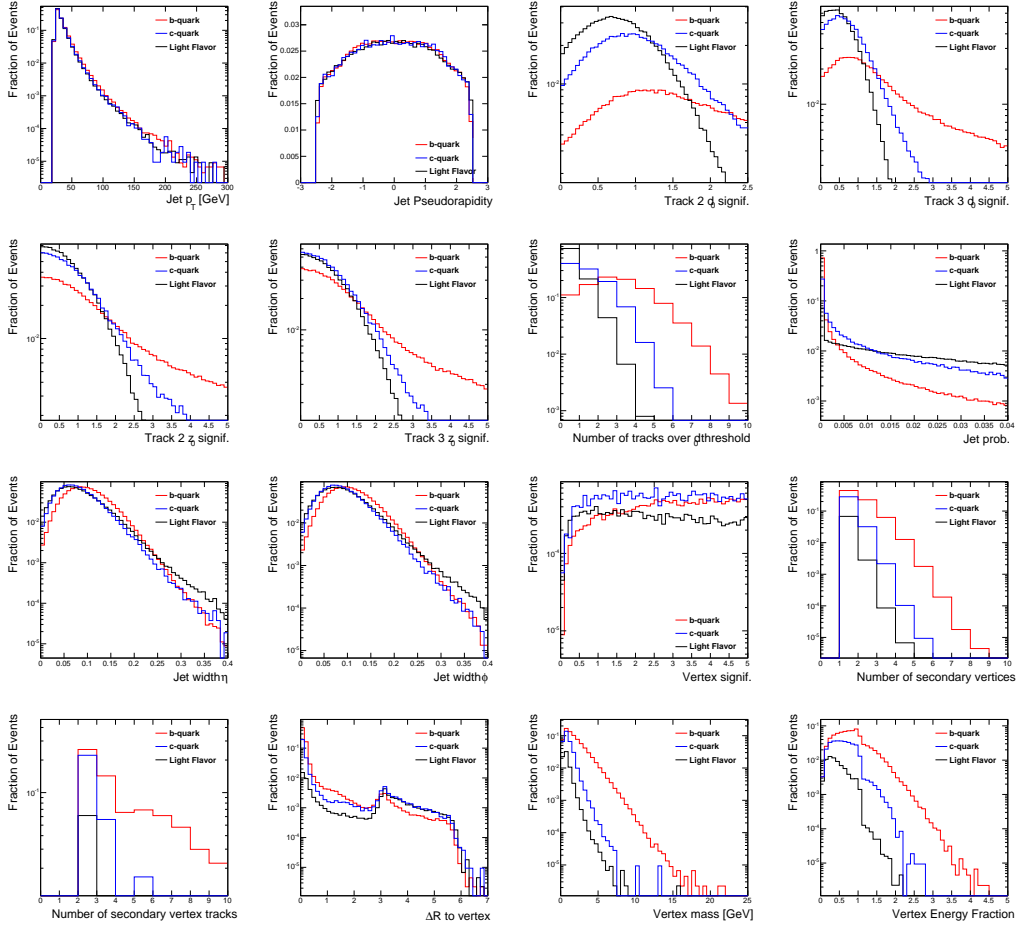


Figure 2.1: Distributions in simulated samples of high-level jet flavor variables widely used to discriminate between jets from light-flavor and heavy-flavor quarks.

## 2.5 Methods

In the experiments, we typically use 8 million samples for training, one million for validation, and one million for testing. Since there are three labels but we are interested in the study of signal vs background and classification, the labels are converted to binary by mapping bottom quark to one, and both charm and light quark to zero. We study the light-quark and charm-quark rejection separately.

### 2.5.1 Machine Learning Approaches

To each simulated collision is attached a set of tracks and a set of vertices. This poses challenges for a machine learning approach in that the size of these sets is variable as seen in Fig. 2.2 and the sets are unordered, although as usual an arbitrary order is often used to list their elements. To address and explore these challenges we use three different deep learning approaches: feedforward neural networks, recurrent neural networks with LSTM (Long Short Term Memory) units, and outer recursive neural networks.

#### Feedforward Neural Networks

The track feature set and the vertex feature set have variable size for a given collision. However, the structure of feedforward networks requires a fixed-size input to make predictions. Thus the use of feedforward neural networks requires first an arbitrary ordering and then a capping of the size of the input set, with zero padding for sets that are smaller than the capped size. To resolve the arbitrary ordering the tracks were sorted by decreasing absolute  $d_0$  significance. This ordering also ensures that tracks from a secondary vertex, which typically have large  $d_0$ , are unlikely to be removed by the capping. Random ordering before adding the padding was also tested but the performance was lower than using the absolute

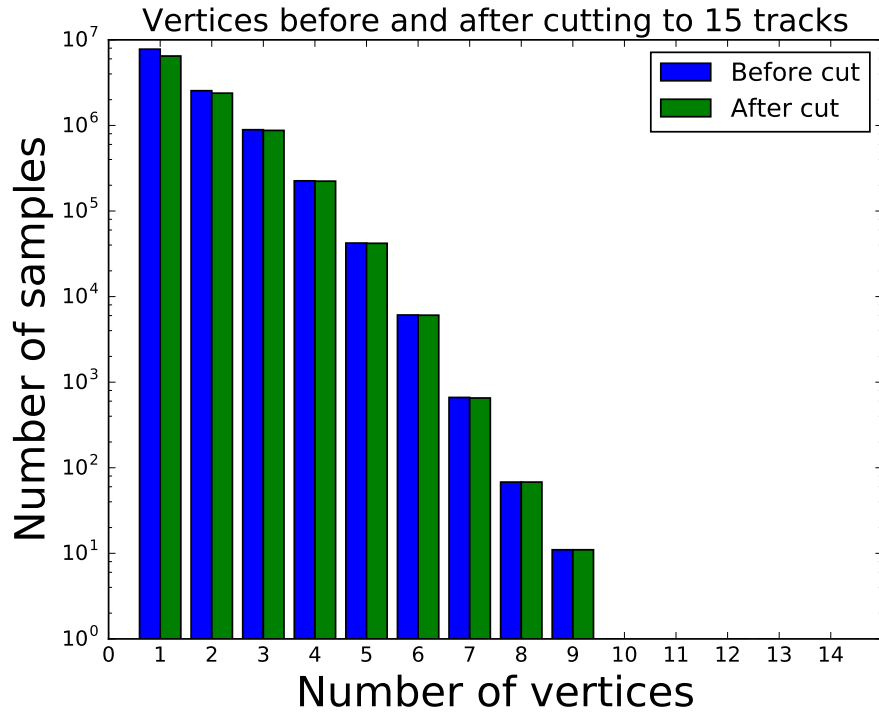
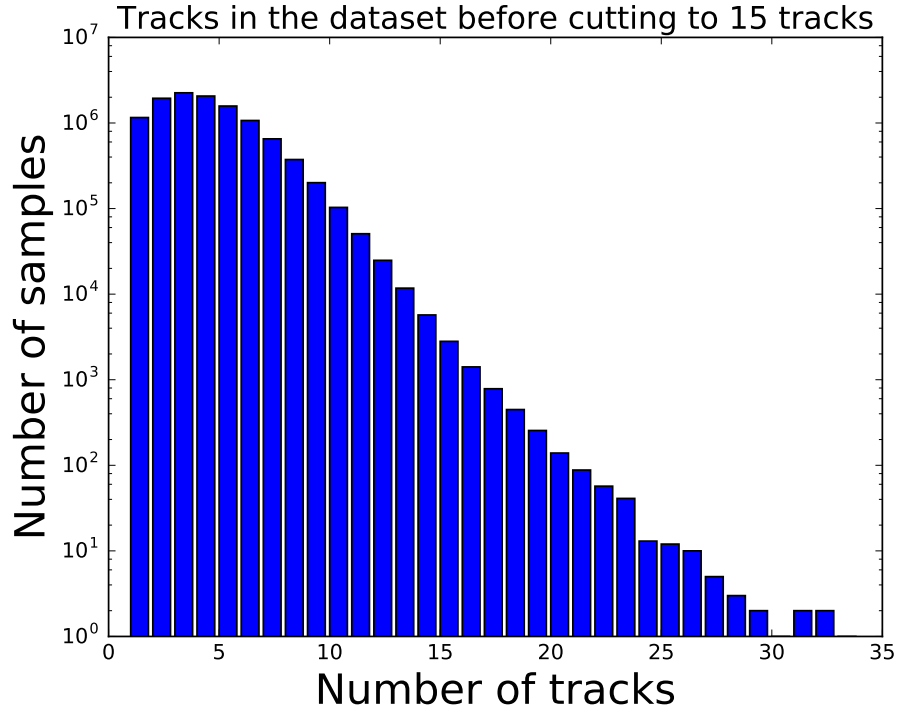


Figure 2.2: Top: Distribution of the number of tracks associated to a jet in simulated samples. Bottom: Distribution of the number of vertices associated to a jet in simulated samples, before and after removing tracks which exceed the maximum allowed value of 15.

$d_0$  significance ordering.

To create a fixed size input, the number of tracks was limited to 15, from a maximum of 33. Using 15 as the cutoff value ensures that 99.97% of the samples preserve all their original tracks; see Fig. 2.2. Tracks are associated to vertices by concatenating the track parameters with those from the associated vertex. Before training, the samples are preprocessed by shifting and scaling such that each feature has a mean of zero and a standard deviation of one. Jets with fewer than 15 tracks are zero-padded after preprocessing. After the cut on the number of tracks, the maximum number of vertices is 12 with an average of 1.5; see Fig. 2.2.

The feedforward neural networks were trained on 8 million training samples with one million more for validation using stochastic gradient descent with mini-batches of 100 samples. They were trained for 100 epochs and the best model was chosen based on the validation error. Momentum for the weights updated was used and linearly increased from zero to a final value over a specified number of epochs. Learning rate decayed linearly from 0.01 to a final value starting and finishing at a specified number of epochs. Dropout (in which nodes are removed during training) with values of  $p$  from 0.0 to 0.5 were used at several combinations of layers to add regularization [60, 22]. These networks had 9 fully connected hidden layers with rectified linear units [50, 62].

Shared weights for each track object were used at the first layer to preserve information about the structure of the data; see Fig 2.3. When adding the vertex and high level variables to the tracks, these were also included within the set of variables with shared weights. The weights for all but the last layer were initialized from a uniform distribution between  $[-\sqrt{6/C}, \sqrt{6/C}]$  where  $C$  is the total number of incoming and outgoing connections [49]. The weights for the last layer were initialized from a uniform distribution between -0.05 and 0.05. A manual optimization was performed over all the hyperparameters to find the best model.

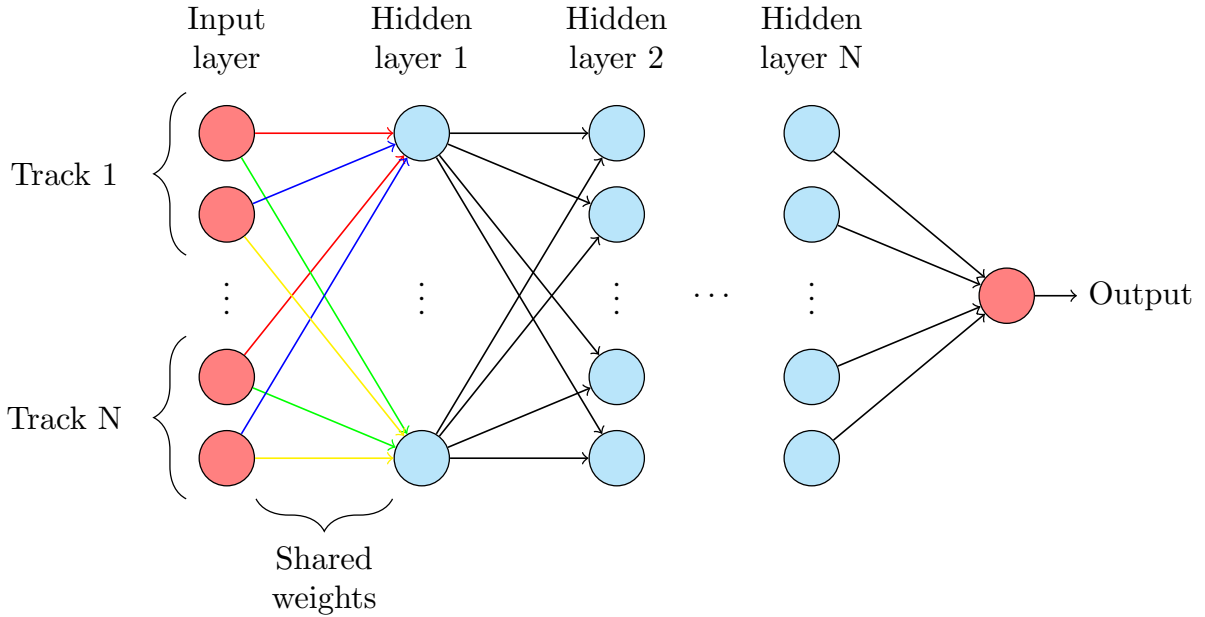


Figure 2.3: Feedforward neural network architecture. In the first layer, connections of the same color represent the same value of the shared weight. The others layers are fully connected without shared weights.

## LSTM Networks

A natural approach to handling variable-sized input is to use recursive neural networks. Broadly speaking, there are two classes of approaches for designing such architectures, the inner approach and the outer approach [16]. In the inner approach, neural networks are used inside the data graphs to crawl the corresponding edges and compute the final output. This process requires the data graphs to be directed and acyclic. Since here the data consists of a set of vertices and tracks, we first convert the data into a sequence by ordering the vertices and tracks as described previously and then use recursive neural networks for sequences, in combination with Long Short Term Memory units [48, 54] to better capture long range dependencies. In the underlying acyclic graph, the variables associated with each node are a function of the variables associated with the parent nodes. Each such function can be parameterized by a neural network. Because the directed acyclic graph has a regular structure, the same network can be applied at different locations of the graph, ultimately



producing the LSTM grid network in Figure 2.4.

We follow the standard implementation of LSTMs with three gates (input, forget, output) and initialize the connections to random orthonormal matrices. The input data consists of a sequence of concatenated track, vertex, and expert features (or different sub-combinations thereof) which are sorted by their absolute  $d_0$  significance, as was the case with the fully connected models. The main difference is that we do not need zero-padding as the LSTM networks can handle sequences of arbitrary length, though we retain the same maximum of 15 tracks for comparability. The final model consists of one LSTM layer comprising between 50 and 500 neurons, and a feedforward neural network with one to four hidden layers that receives its input from the LSTM network and produces the final predictions (where each layer has between 50 and 500 units). We add dropout layers in between the LSTM and each hidden fully connected layer. For hyperparameter-optimization we performed a random search over these parameters as well as the individual dropout rates that are part of the model. We trained the LSTM networks for 100 epochs using SGD with a momentum of 0.9 and decay the step-size parameter from initially  $2 \cdot 10^{-3}$  down to  $10^{-4}$  over the course of training.

## Outer Recursive Networks

Alternatively, to handle inputs of variable size, we can use an outer recursive approach, where neural networks are built in a direction perpendicular to the original data graph, with horizontal weight sharing. The outer approach can be used to build more symmetric deep architectures; see Fig. 2.5. For instance, in our case the input consists of up to 15 tracks, from which we can sample all possible pairs of tracks and use a shared neural network that processes these in the first layer of the outer approach. In this case, there are at most  $\binom{15}{2} = 105$  unordered pairs, or 210 ordered pairs, which is manageable especially considering that there is a single network shared by all pairs. Using ordered pairs would yield the most

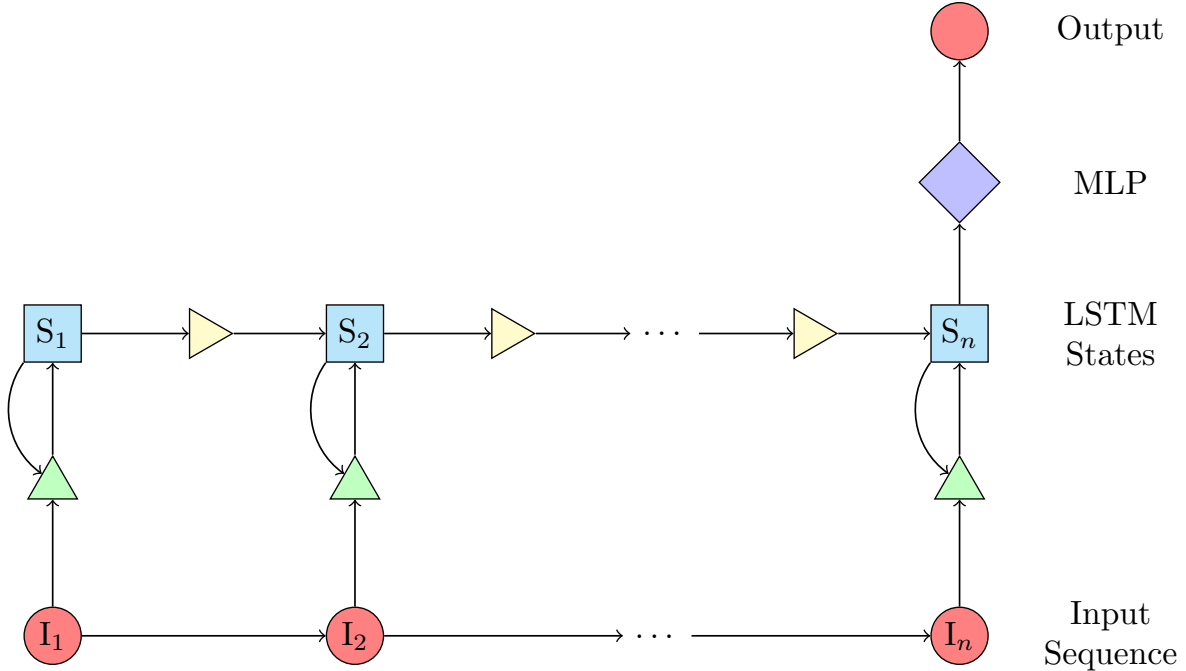


Figure 2.4: Architecture of the Long Short Term Memory networks as described in the text.

symmetric overall network. At the next level of the architecture, one can for instance use a network for each track  $t_i$  that combines the outputs of all the networks from the first layer associated with pairs containing  $t_i$ , and so forth. In the second level of the outer architecture, for simplicity here we use a fully connected feedforward network that computes the final output using the outputs of all the pair networks. More specifically, for each data sample we compute the list of stacked track features for all 210 pairs and process each pair with a shared nonlinear hidden layer (with 5 to 20 neurons). The resulting outputs for all pairs are then concatenated and fed into a multilayer perceptron as was the case for the LSTM models, with one to four hidden layers containing between 100 and 600 hidden units. We again use dropout layers in between the hidden layers and optimize the dropout rates and network depth and size using random search.

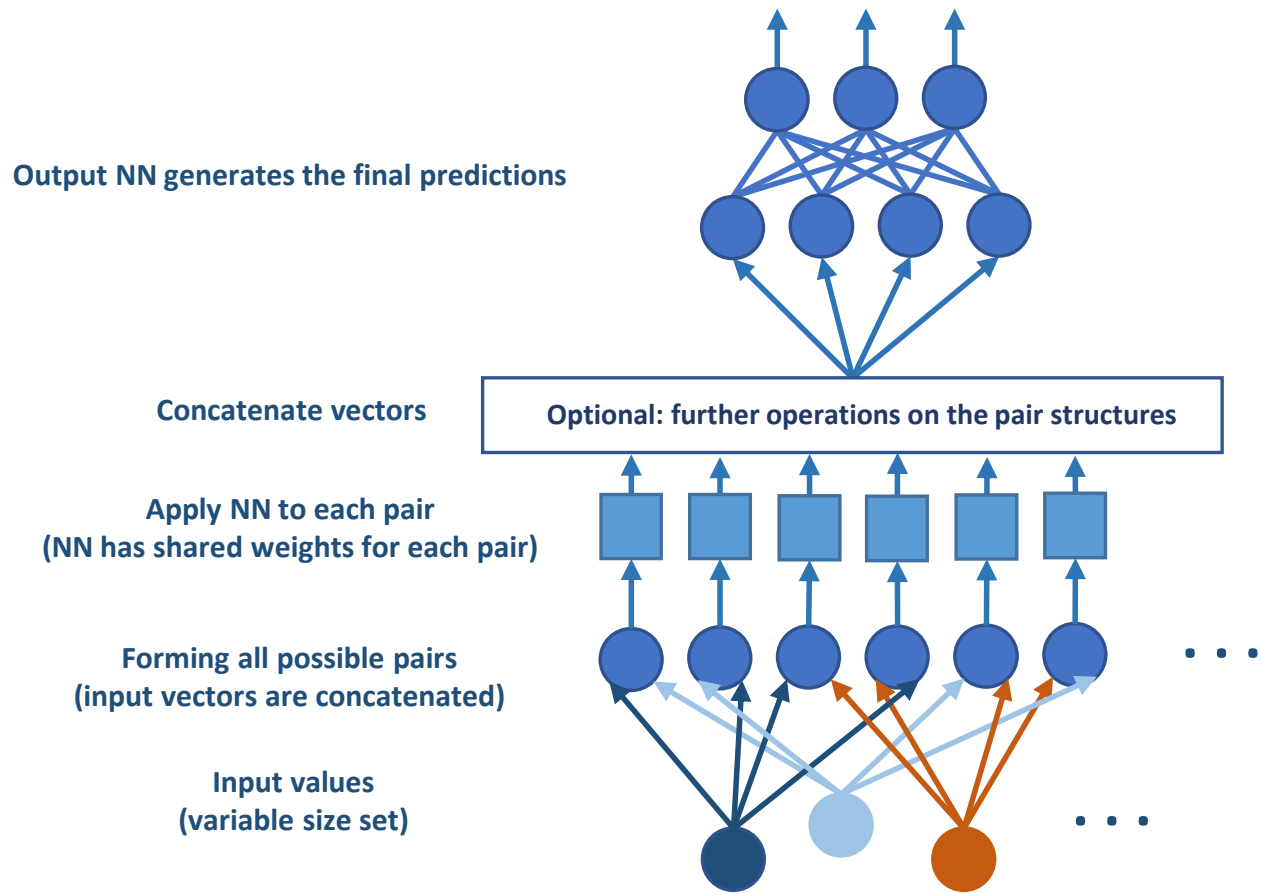


Figure 2.5: Architecture of the outer recursive networks as described in the text.

## 2.5.2 Hardware and Software Implementations

All computations were performed using machines with 16 Intel Xeon cores, NVIDIA Titan graphics processors, and 64 GB memory. All neural networks were trained using the GPU-accelerated Theano software library [82] and, for the feed forward neural networks, also the Keras software library [33].

## 2.6 Results

The best feedforward neural networks have 9 fully connected hidden layers with 400 rectified linear units and a single sigmoid unit at the end. On the first layer the networks have shared weights. The first five tracks have one set of shared weights per track, tracks 6 to 10 have a second set of shared weights per track and the last five tracks have a third set of shared weights per track. They have a momentum term of 0 which starts to linearly increase at the first epoch and reaches its final value of 0.5 at epoch 100. Initially, the learning rate is set at 0.01 and, starting at epoch 80, it is linearly decreased to a final value of 0.001 at epoch 100. Dropout was used in the first two layers with a value of  $p=0.3$ . The same architecture was used across all the combinations of features except in the case of using only high level features, in which case the first layer is fully connected without any shared weights.

We found that the main characteristic of the best LSTM models is a relatively small size of the hidden state representation of the LSTM module (about 70 units), while the size of the MLP, which is sitting on top of it, is of secondary importance for overall performance of the model. The best models using the outer recursive approach contain between two and three hidden layers on top of the shared-weight layer (which operates on all paired tracks) and those contain 17 or more neurons.

Final results are shown in Table 2.1. The metric used is the Area Under the Curve (AUC),

calculated in signal efficiency versus background efficiency, where a larger AUC indicates better performance. In Fig. 2.6, the signal efficiency is shown versus background rejection, the inverse of background efficiency. Figures 2.7 and 2.8 show the efficiency versus jet  $p_T$  and pseudorapidity for fixed values of background rejection. Figures 2.9 and 2.10 show the rejection versus jet  $p_T$  and pseudorapidity for fixed values of signal efficiency.

The results can be analyzed to draw conclusions regarding the power of the learning algorithms to extract information at different levels of preprocessing, and to compare the three learning approaches.

The state-of-the-art performance is represented by the networks which use only the expert-level features. Networks using only tracking or vertexing features do not match this performance, though networks using both tracking and vertexing do slightly exceed it. In addition, networks which combine expert-level information with track and/or vertex information outperform the expert-only benchmark, in some cases by a significant margin.

For any given set of features, the feedforward deep networks most often give the best performance, though in some cases by a small margin over the LSTM approach. This may be somewhat unexpected since LSTMs were created to handle variable sized input data as is the case here. We must note, however, that unlike truly sequential data like speech or text there is no natural order in the data that we are working on. The tracks have been ordered by absolute  $d_0$  significance, which tends to cluster tracks belonging to the same vertex, but a sequential model with this ordering may not be superior to processing tracks in parallel, as in the connected DNN with tied weights.

While one cannot probe the strategy of the ML algorithm, it is possible to compare distributions of events categorized as signal-like by the different algorithms in order to understand how the classification is being accomplished. To compare distributions between different algorithms, we study simulated events with equivalent background rejection, see Fig. 2.11

Table 2.1: Performance results for networks using track-level, vertex-level or expert-level information. In each case the jet  $p_T$  and pseudorapidity are also used. Shown for each method is the Area Under the Curve (AUC), the integral of the background efficiency versus signal efficiency, which have a statistical uncertainty of 0.001 or less. Signal efficiency and background rejections are shown in Figs. 2.6-2.10.

Inputs			Technique	AUC
Tracks	Vertices	Expert		
✓			Feedforward	0.916
✓			LSTM	0.917
✓			Outer	0.915
	✓		Feedforward	0.912
	✓		LSTM	0.911
	✓		Outer	0.911
✓	✓		Feedforward	0.929
✓	✓		LSTM	0.929
✓	✓		Outer	0.928
		✓	Feedforward	0.924
		✓	LSTM	0.925
		✓	Outer	0.924
✓		✓	Feedforward	0.937
✓		✓	LSTM	0.937
✓		✓	Outer	0.936
	✓	✓	Feedforward	0.931
	✓	✓	LSTM	0.930
	✓	✓	Outer	0.929
✓	✓	✓	Feedforward	0.939
✓	✓	✓	LSTM	0.939
✓	✓	✓	Outer	0.937

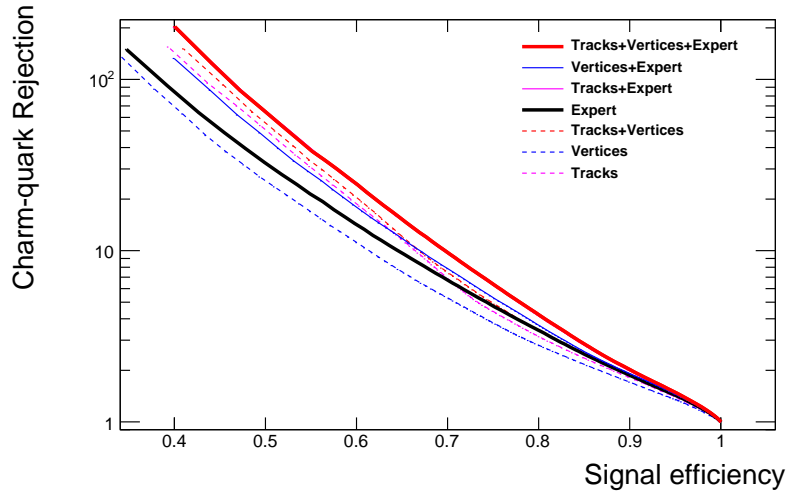
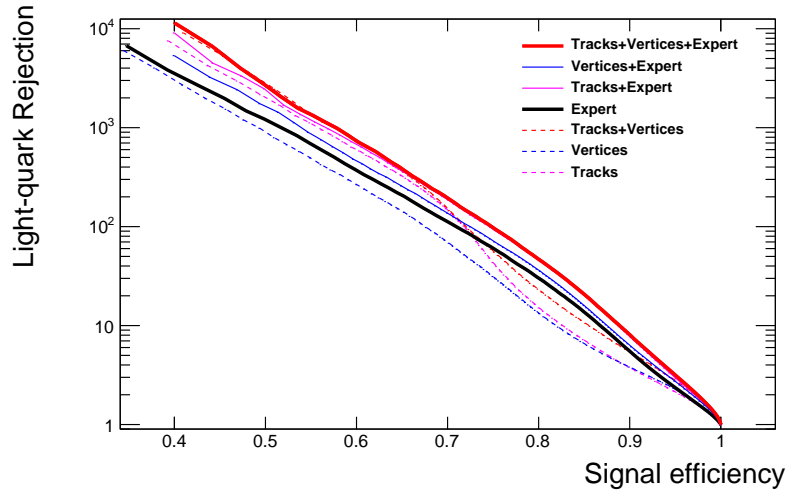


Figure 2.6: Signal efficiency versus background rejection (inverse of efficiency) for deep networks trained on track-level, vertex-level or expert-level features. The top pane shows the performance for  $b$ -quarks versus light-flavor quarks, the bottom pane for  $b$ -quarks versus  $c$ -quarks.

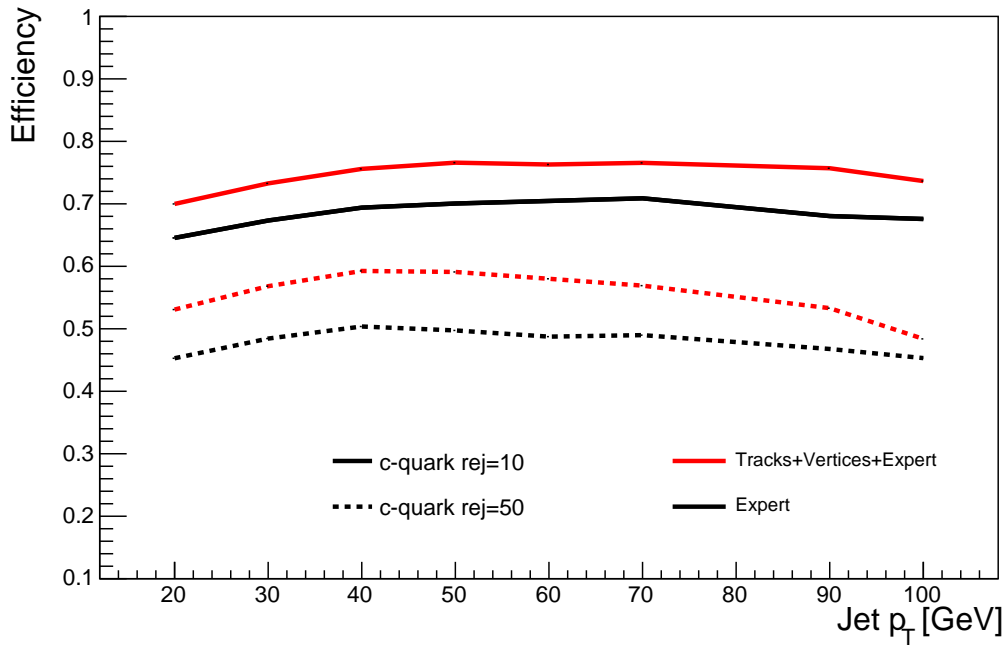
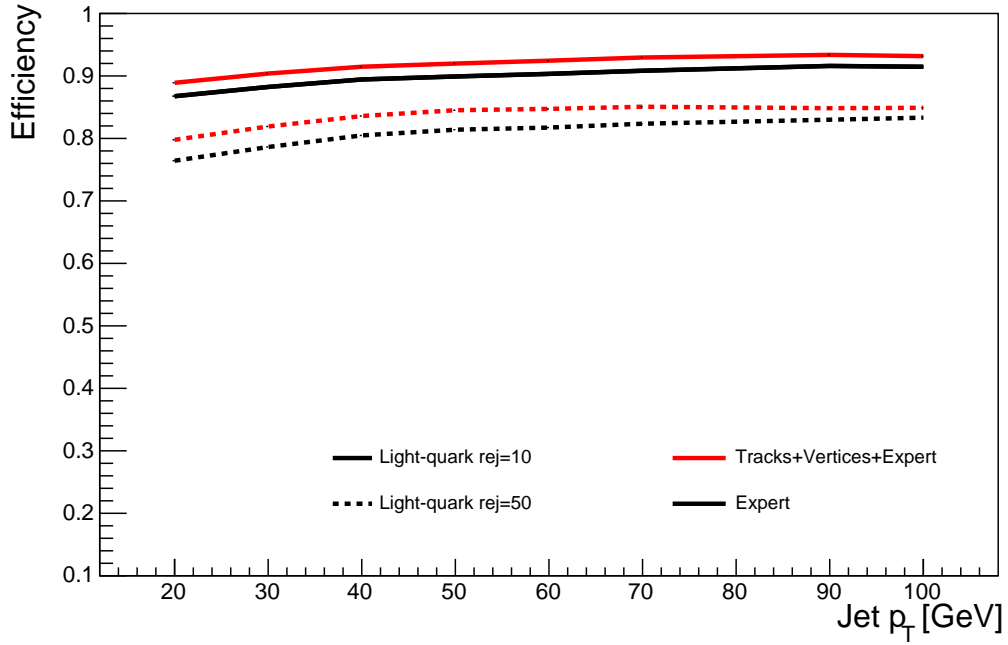


Figure 2.7: Signal efficiency versus minimum jet  $p_T$  relative to light quarks (top) or charm quarks (bottom). In each case, efficiency is shown for fixed values of background rejection for networks trained with only expert features or networks trained with all features (tracks, vertices and expert features).



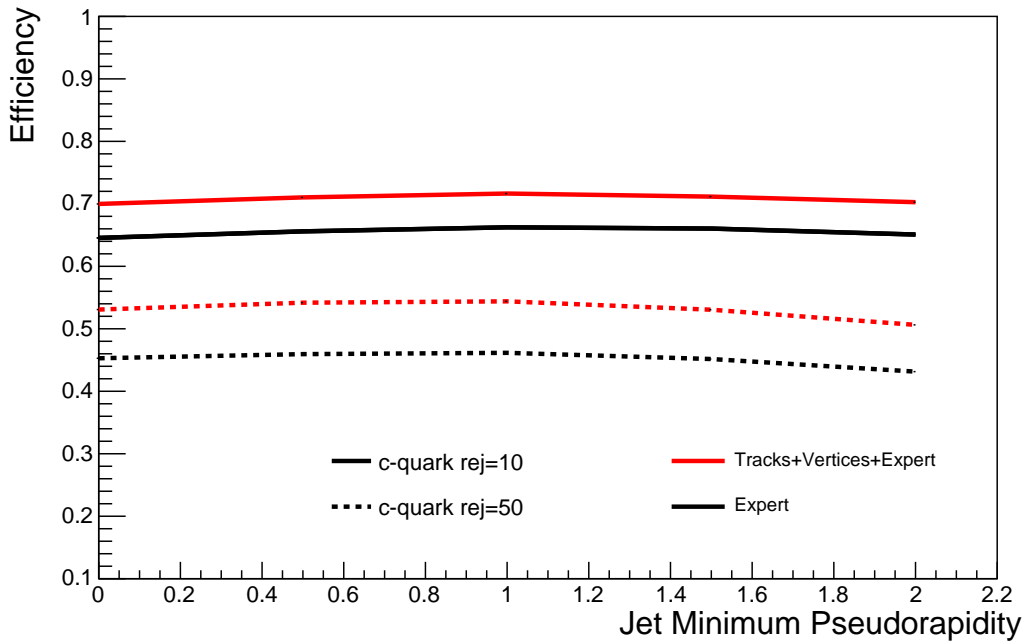
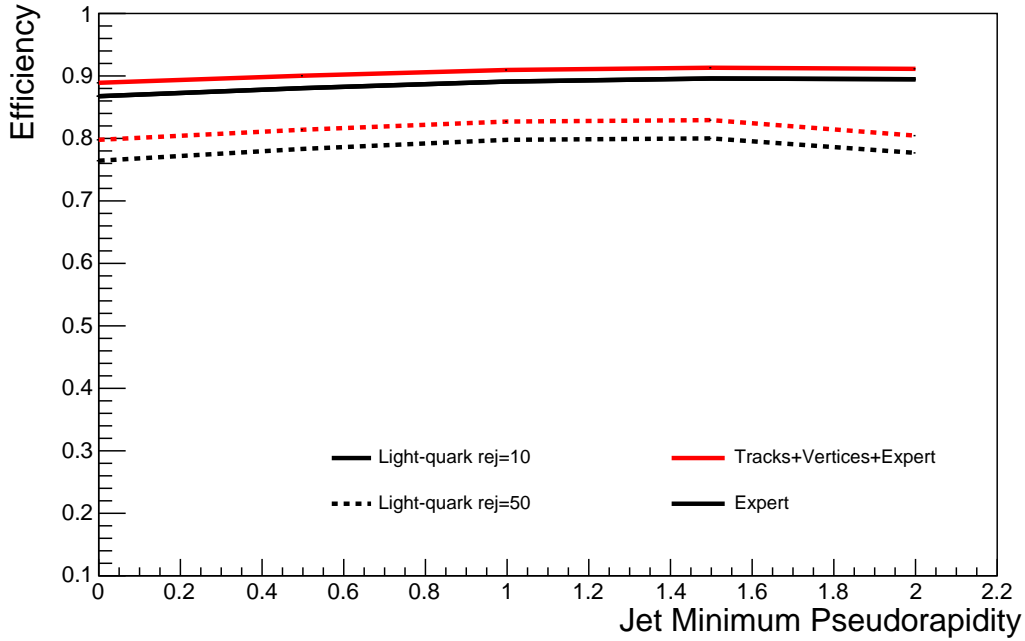


Figure 2.8: Signal efficiency versus minimum jet pseudo-rapidity relative to light quarks (top) or charm quarks (bottom). In each case, efficiency is shown for fixed values of background rejection for networks trained with only expert features or networks trained with all features (tracks, vertices and expert features).

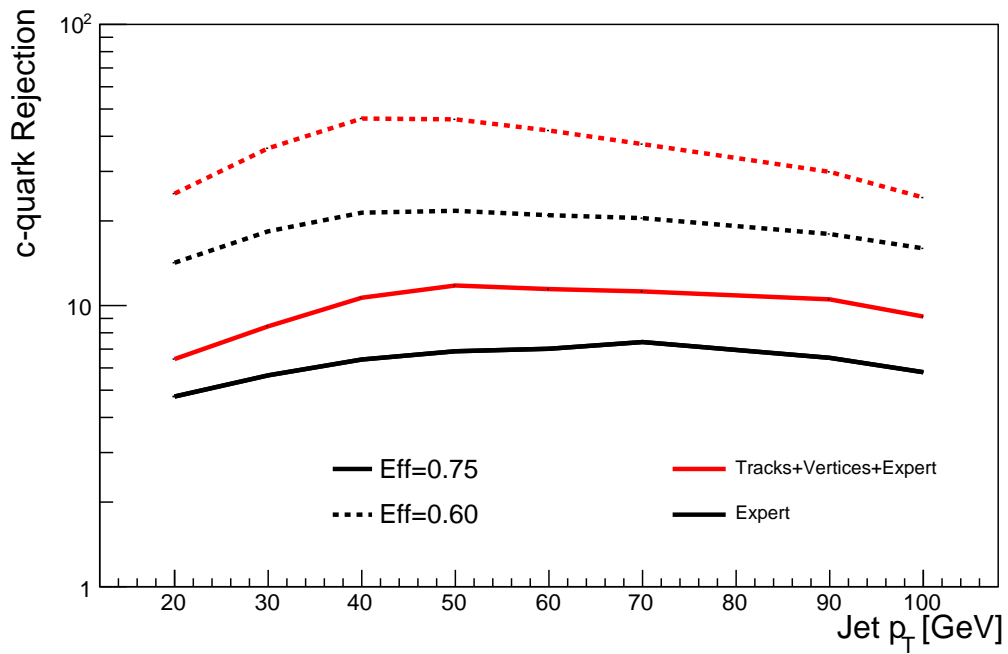
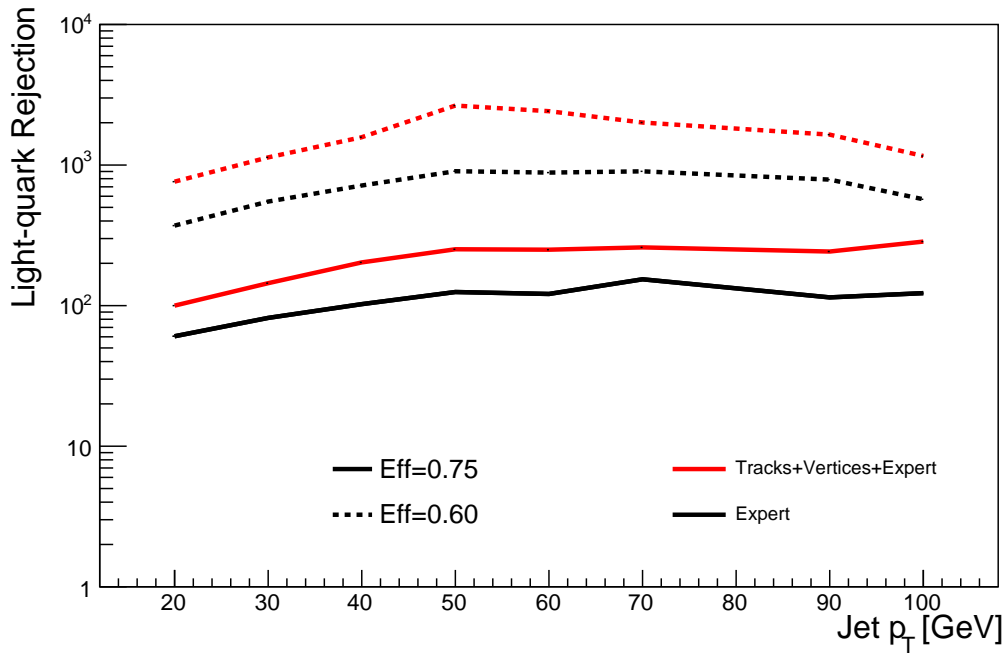


Figure 2.9: Rejection of light quarks (top) or charm quarks (bottom) versus minimum jet  $p_T$ . In each case, rejection is shown for fixed values of signal efficiency for networks trained with only expert features or networks trained with all features (tracks, vertices and expert features).

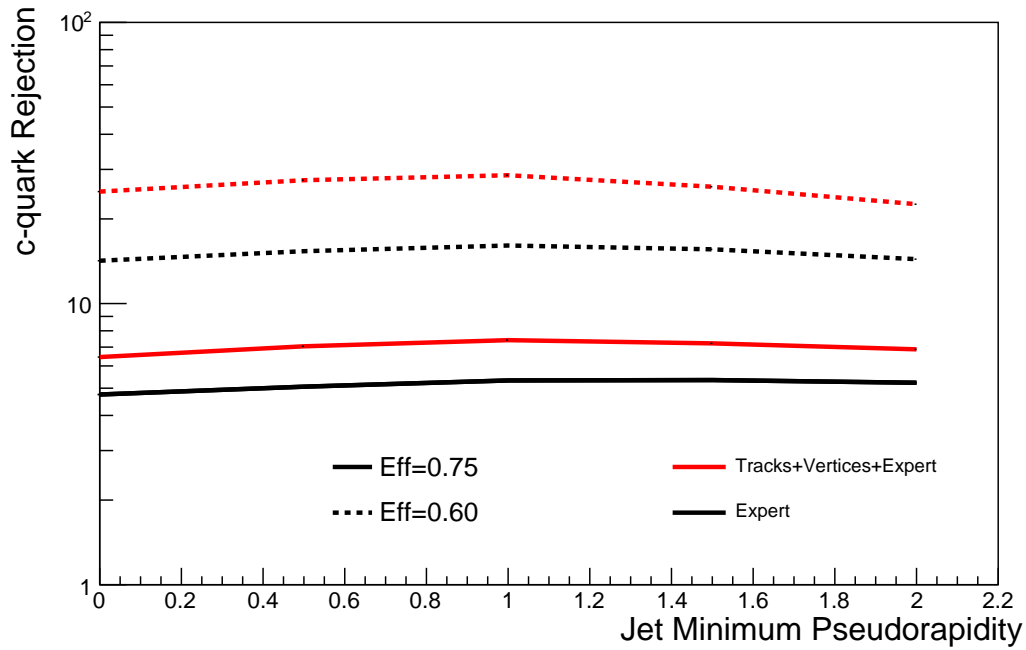
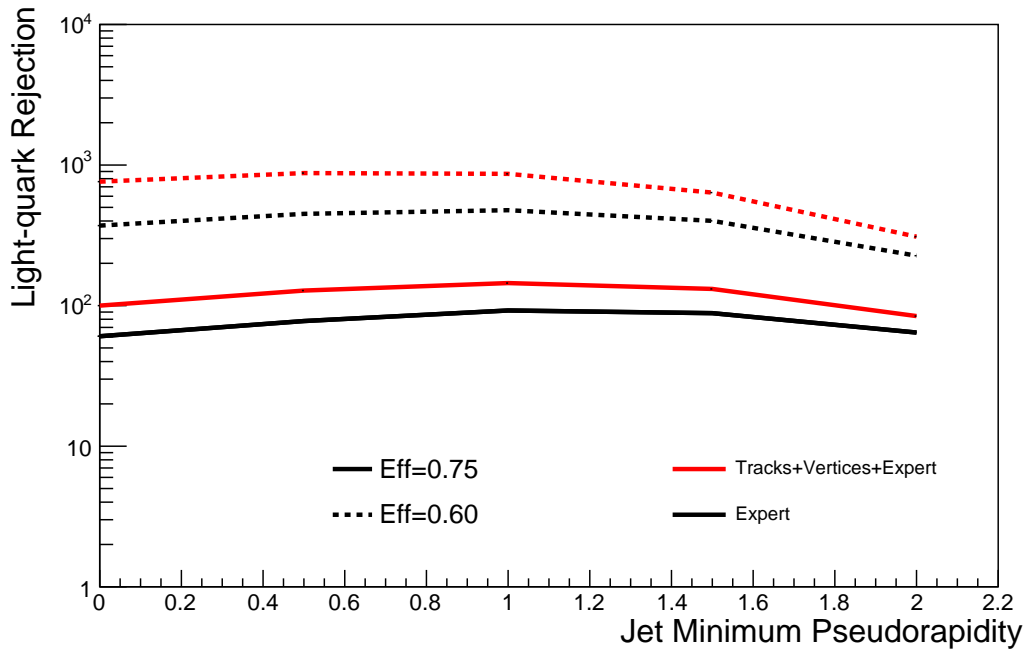


Figure 2.10: Rejection of light quarks (top) or charm quarks (bottom) versus minimum jet pseudo-rapidity. In each case, rejection is shown for fixed values of signal efficiency for networks trained with only expert features or networks trained with all features (tracks, vertices and expert features).

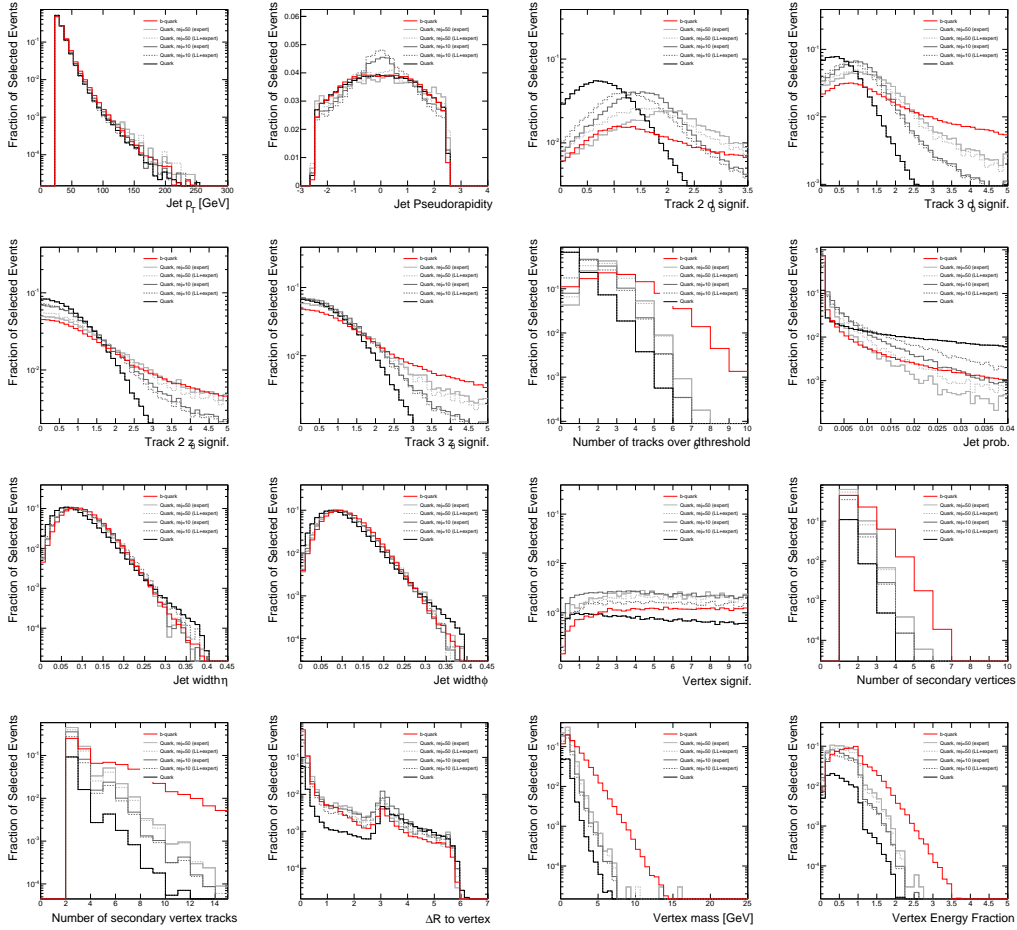


Figure 2.11: Distributions of expert-level features for heavy-flavor and light-flavor classes. Also shown are distributions of light-flavor and charm jets surviving network threshold selections chosen to given rejection of 10 and 50, for networks using only expert information and networks using expert information in addition to lower-level information.

for a comparison of the selected regions in the expert features for classifiers with and without the lower-level information.

## 2.7 Discussion

Our experiments support four conclusions.

**The existing expert strategies for dimensional reduction sacrifice or distort useful information.** Networks which include lower-level information outperform networks using exclusively higher-level information. For example, if the vertex-level information contained all of the classification power of the track-level information but with lower dimensionality, one would expect the vertex-only network to match the performance of the tracks-and-vertex network, as the lower-dimensional problem should be simpler to learn. Instead, networks using tracks and vertices outperform those which use only vertices. Similarly, networks using tracks and expert features outperform those with only expert features. We note that these conclusions apply to the expert strategies considered here, and in the case of the simulated environment we have studied; however, we feel that both are representative of the current state-of-the-art.

**The task remains a challenge for deep networks.** Networks which use only the lower-level information do not match the performance of networks which use the higher-level information. Since the higher-level features are strict functions of the lower-level features, the lower-level features are a superset of the information contained in the high-level features. The performance of the networks which use the high-level features then provides a baseline against which to measure the ability of the network to extract the relevant information in the more difficult higher-dimensional space of lower-level features. Networks using only track information do not match the performance of those which use only the high-level features (but note that track-only networks outperform vertex-only networks, giving a clue as to the area of difficulty).

**Networks using track and vertex information outperform those with expert features.** Networks trained with track and vertex information but without the benefit of expert-level guidance and dimensional reduction manage to achieve better performance than those which use only expert-level features. This is remarkable, as the dimensionality of the tracks+vertices features is very large and expert-only networks represent the current state-

of-the-art. Note, however, that for high signal efficiency ( $> 75\%$ ) the expert-only networks outperform the networks using tracks+vertices.

**Networks which combine expert features with low-level information have the best performance.** Combining the lowest-level information for completeness with the low-dimensional hints from expert features significantly outperforms the state-of-the-art networks which use only expert features. While in principle all of the information exists in the lowest-level features and it should be possible to train a network which matches or exceeds this performance without expert knowledge, this is neither necessary nor desirable. Expert knowledge exists and is well-established, and there is no reason to discard it.

In addition, this expert guidance encourages the network to identify discrimination strategies based on well-understood properties of the jet flavor problem and decreases the likelihood of relying on learning strategies based on spurious or poorly-modeled corners of the space. We note that the use of high-dimensional lower-level data will require careful validation of the simulation models; reasonable strategies exist, such as a combination of the validation of individual features in one-dimensional projections with validation of the network output in control samples, which probes the use of information in multi-feature correlations.

These improvements in the performance of the tagger can give important boosts to physics studies which rely on the identification of jet flavor.

# Chapter 3

## Learning to Identify Electrons

### 3.1 Abstract

We investigate whether state-of-the-art classification features commonly used to distinguish electrons from jet backgrounds in collider experiments are overlooking valuable information. A deep convolutional neural network analysis of electromagnetic and hadronic calorimeter deposits is compared to the performance of typical features, revealing a  $\approx 5\%$  gap which indicates that these lower-level data do contain untapped classification power. To reveal the nature of this unused information, we use a recently developed technique to map the deep network into a space of physically interpretable observables. We identify two simple calorimeter observables which are not typically used for electron identification, but which mimic the decisions of the convolutional network and nearly close the performance gap.

## 3.2 Introduction

Production of electrons in high-energy collisions provides an essential handle on precision studies of the Standard Model [5, 12] as well as for searches for new physics [3, 32]. The identification of electrons, and their separation from backgrounds which mimic their signature, is therefore a critical element in the data analysis toolkit, especially at lower transverse momentum, where the backgrounds rise rapidly [61].

In collider experiments, electrons are identified by an isolated track which aligns with a localized energy deposition, primarily in the electromagnetic calorimeter. The primary source of backgrounds is the production of hadronic jets, which typically feature multiple tracks and extended energy deposition in both electromagnetic and hadronic calorimeters, but can fluctuate to mimic electrons. The tracker and calorimeters, however, are very finely segmented, producing high-dimensional data which is difficult to analyze directly. A mature literature [63, 36] contains higher-level features designed by physicists to highlight the distinct signature of the electron and suppress the backgrounds. The higher-level features define a lower-dimensional feature space.

Recent strides in machine learning for physics, particularly the advent of deep learning [23, 55, 17] and image-processing techniques [34, 18, 41, 42], have demonstrated that high-level features designed by domain experts may not always fully capture the information available in the lower-level high-dimensional data. Specifically, the rich but subtle structure of the deposition of energy by jets provides a powerful potential handle for discrimination. Given their role as the dominant background, this suggests that additional classification power may be gained by applying image-based deep learning techniques to electrons.

In this study, we apply deep convolutional neural networks (CNNs) to the task of distinguishing between electrons and jets, using separate images from the electromagnetic and hadronic calorimeters. Due to the black-box nature of their operation, we do not propose



to use CNNs in place of the high-level features. Instead, we apply CNNs as probe the information content of the low-level data in comparison to the high-level features. We show that the classification performance of the image-based CNNs exceeds the performance of the high-level features in common use by Large Hadron Collider (LHC) experiments, by a small, but significant, margin. We then identify the source of the untapped information and construct novel high-level features that capture it.

This paper is organized as follows. In Section 2, we outline our approach. In Section 3, we discuss the details of our image generation process and the corresponding dataset used for CNN experiments. In Section 4, we review the existing state-of-the-art ATLAS and CMS high-level features, which we combine to derive our benchmark performance. In Section 5, we provide details of neural network architectures and training. In Section 6, we discuss the performance of these networks. In Section 7, we search for new high-level features to bridge the gap between CNNs and standard features. In Section 8, we summarize and discuss the results, providing an intuitive understanding of the underlying landscape.

### 3.3 Overview

This study explores whether low-level, high-dimensional,  $\mathcal{O}(10^3)$ , calorimeter data contains information useful for distinguishing electrons from a major background not captured by the standard suite of high-level features designed by physicists. Similar studies in jets or flavor tags have revealed such gaps [55, 18].

We probe this issue using a simulated dataset created with publicly available fast simulations tools [38]; while such samples do not typically match the fidelity of those generated with full simulations [11], we refine the calorimeter description for this study and find the modeling sufficiently realistic for a proof-of-principle analysis. Our focus is on comparing physically

motivated, high-level features to low-level image techniques on equal footing. While we anticipate that the numerical results will be different when evaluated in a fully realistic scenario, the broad picture will likely remain the same. The technique described here is fairly general and applicable to more realistic experimental scenarios, so that valuable lessons can be learned in the present context.

We reproduce the standard suite of electron identification features, as described in Refs. [63, 36], in the context of our simulated description. We then compare their combined performance to that of deep convolutional neural networks (CNNs) which have been trained to analyze the lower-level calorimeter cells using image recognition techniques [34, 41, 18]. We do not advocate for the use of CNNs to replace high-level features whose designs are grounded in physics; CNNs are difficult to interpret and the low level and large dimensionality of the input makes validation of the features and definition of systematic uncertainties nearly impossible. Instead, here we use the power of CNNs as a probe, to test whether further information is present in the low-level data. Having identified a gap, we then explore a complete space of novel high-level features, Energy Flow Polynomials (EFPs) [67] to interpret and bridge the gap.

## 3.4 Dataset Generation

In this section, we describe the process of generating simulated signal and background datasets, reproducing the standard suite of high-level features, and forming pixelated images from the electromagnetic and hadronic calorimeter deposits.

### 3.4.1 Processes and Simulation

Simulated samples of isolated electrons are generated from the production and electronic decay of a  $Z'$  boson in hadronic collisions,  $pp \rightarrow Z' \rightarrow e^+e^-$  at  $\sqrt{s} = 13$  TeV. We set  $m_{Z'}$  to 20 GeV in order to efficiently produce electrons in the range  $p_T = [10, 30]$  GeV, where hadronic backgrounds are significant. Simulated samples of background jets are generated via generic dijet production. Events were generated with MADGRAPH v2.6.5 [13], decayed and showered with PYTHIA v8.235 [79], with detector response described by DELPHES v3.4.1 [38] using ROOT version 6.0800 [26].

Our configuration of DELPHES approximates the ATLAS detector. For this initial study, we model only the central layer of the calorimeters where most energy is deposited; future work will explore more detailed and realistic detector simulation. However, we maintain the critical separation between the electromagnetic and hadronic calorimeters and their distinct segmentation. Our simulated electromagnetic calorimeter (ECal) has segmentation of  $(\Delta\phi, \Delta\eta) = (\frac{\pi}{126}, 0.025)$  while our simulated hadronic calorimeter (HCal) is coarser,  $(\Delta\phi, \Delta\eta) = (\frac{\pi}{31}, 0.1)$ . This approach allows us to investigate whether information about the structure of the many-particle jet is useful for suppressing their contribution. See Ref [42] for an analysis of the information contained in the shape of shower for individual particles.

No pile-up simulation was included in the generated data, as pileup subtraction techniques have been shown to be effective [25]. In total, we generated 107k signal and 107k background objects.

### 3.4.2 Electron Candidate Selection

We use DELPHES' standard electron identification procedures where loose electron candidates are selected from charged particle tracks which align with energy deposits in the ECal. We

required object to have track  $p_T > 10$  GeV and  $|\eta| < 2.125$ , to avoid edge effects when forming calorimeter images, see Fig. 3.1. For later training, the background objects are reweighted to match the  $p_T$  distribution of the signal.

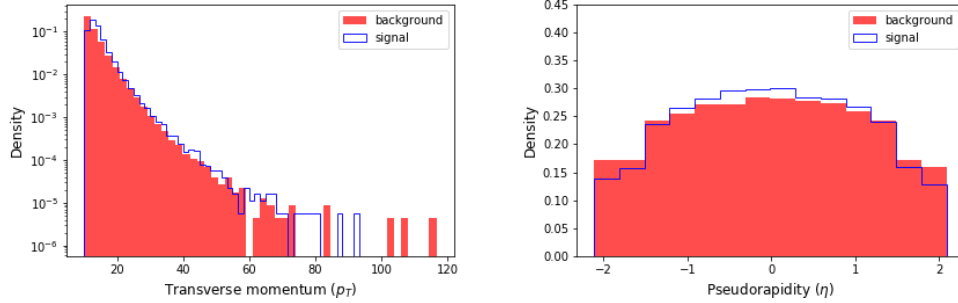


Figure 3.1: Distribution of generated electron candidate  $p_T$  and  $\eta$  for simulated signal and background samples, before reweighting to match spectra.

### 3.4.3 Image Formation

The cells of the calorimeter can be naturally organized as pixels of an image, allowing for use of powerful image-processing techniques. Each pixel contains the energy deposited in one cell. Alternatively, one may form images in which each cell represents  $E_T = E/\cosh\eta$ , which folds in the location of the object relative to the collision point. For completeness we initially consider images in which pixels represent  $E$  and images where pixels represent  $E_T$ . Additionally, we create separate images for the ECal and HCal, in order to preserve the separate and powerful information they offer. In total, four images are created for each electron candidate: ECal  $E$ , ECal  $E_T$ , HCal  $E$ , HCal  $E_T$ .

The center of a calorimeter image is chosen to be the ECal cell with largest transverse energy deposit in the  $9 \times 9$  cell region surrounding the track of the highest  $p_T$  electron in that event. This accounts for the curvature in the path of the electron as it propagates between the tracker and the calorimeter. The ECal image extends fifteen pixels in either direction, forming a  $31 \times 31$  image. The HCal granularity is four times as coarse, and an

$8 \times 8$  image covers the same physical region. Figures 3.2 and 3.3 show example and mean images for the ECal and HCal, respectively.

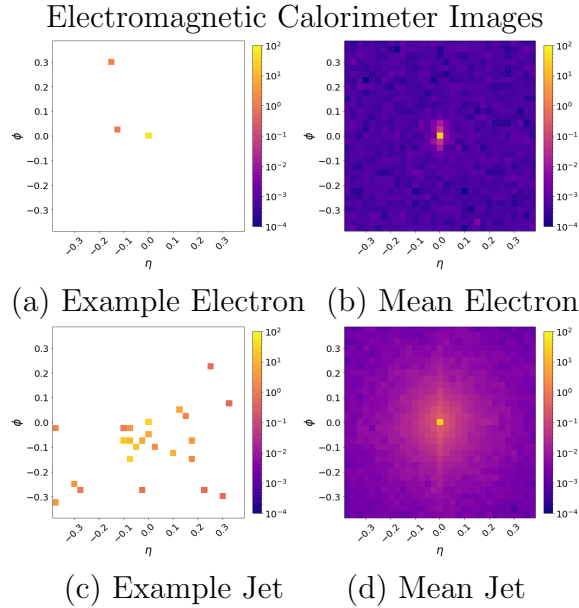


Figure 3.2: Images in the electromagnetic calorimeter for signal electrons (top) and background jets (bottom). On the left are individual examples, on the right are mean images. See Fig. 3.3 for corresponding hadronic calorimeter images.

### 3.5 Standard Classification Features

To assess the performance of the high-level classification features typically used by ATLAS [36] and CMS [63] which identify electrons and reject jet backgrounds, we reproduce their form here, where relevant.

Since electron candidates are confined to the longitudinal range  $|\eta| < 2.125$ , we only consider variables that are well-defined in this range. Additionally, we only consider variables which are based on information included in our simulation, to ensure the comparison uses information on equal footing. In addition, we do not perform clustering; where a feature calls for the cluster energy, we replace it with the total energy of the candidate image, a reasonable proxy for the cluster in our less-finely segmented simulation. All high-level features are calculated

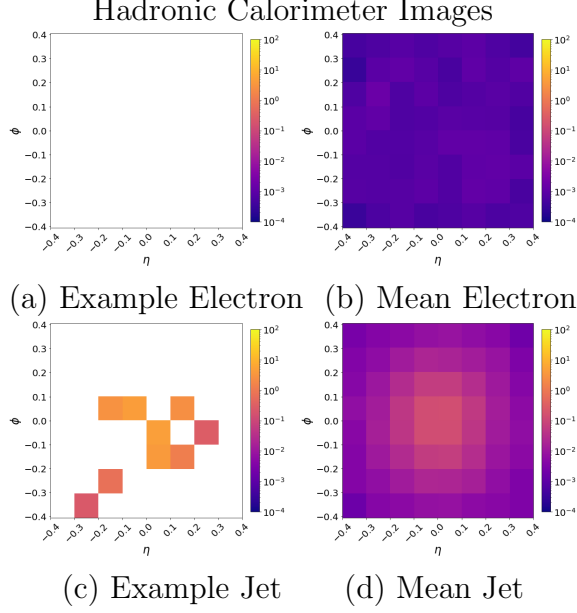


Figure 3.3: Images in the hadronic calorimeter for signal electrons (top) and background jets (bottom). On the left are individual examples, on the right are mean images. See Fig. 3.2 for corresponding electromagnetic calorimeter images.

from the ECal and HCal images, using  $E$  or  $E_T$  images where appropriate.

We reproduce seven features:  $R_{\text{had}}$ ,  $\omega_{\eta 2}$ ,  $R_\phi$ ,  $R_\eta$ ,  $\sigma_{\eta\eta}$ , and two isolation quantities. Together, these capture the typical strategies of suppressing objects with significant hadronic energy or extended energy deposits. Definitions of each feature are below, and distributions for signal and background samples are shown in Fig. 3.4.

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

The feature  $R_{\text{had}}$  relates the transverse energy ( $E_T$ ) in the electromagnetic calorimeter to that in the hadronic calorimeter. Specifically,

$$R_{\text{had}} = \frac{\sum_i E_{T,i}^{\text{HCal}}}{\sum_j E_{T,j}^{\text{ECal}}} \quad (3.1)$$

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

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

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

$$w_{\eta 2} = \sqrt{\frac{\sum_i E_i (\eta_i)^2}{\sum_i E_i} - \left(\frac{\sum_i E_i \eta_i}{\sum_i E_i}\right)^2} \quad (3.2)$$

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

**Azimuthal and Longitudinal Energy Distributions:  $R_\phi$  and  $R_\eta$**

To probe the distribution of energy in azimuthal ( $\phi$ ) and longitudinal ( $\eta$ ) directions, we calculate two features  $R_\phi$  and  $R_\eta$ . Qualitatively, these relate the total ECal energy in a subset of cells to the energy in a larger subset of cells extended in either  $\phi$  or  $\eta$ , respectively. Specifically,

$$R_\phi = \frac{E_{3 \times 3}}{E_{3 \times 7}}, \quad R_\eta = \frac{E_{3 \times 7}}{E_{7 \times 7}} \quad (3.3)$$

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

### 3.5.1 Lateral Shower Extension: $\sigma_{\eta\eta}$

An alternative probe of the distribution of energy in  $\eta$  is  $\sigma_{\eta\eta}$ . Specifically,

$$\sigma_{\eta\eta} = \sqrt{\frac{\sum_i w_i (\eta_i - \bar{\eta})^2}{\sum_i w_i}} \quad (3.4)$$

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

### 3.5.2 Isolation

Jets typically deposit significant energy surrounding the energetic core, where electrons are typically isolated in the calorimeter. To assess the degree of isolation, we sum the ECal energy in cells within the angular range  $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2} < 0.3$  or  $0.4$ , where  $\Delta\eta$  and  $\Delta\phi$  are measured from a given cell's center and the center of the image.

## 3.6 Neural Network Architectures and Training

We construct multi-layer neural networks that accept low-level images, or high-level features, or both, with a sigmoidal logistic unit as their output unit to classify between signal and background.

Each image input is passed through a number of convolutional blocks, with each block consisting of two convolutional layers with  $3 \times 3$  kernels, rectified linear units [51] as the activation function, and a final  $2 \times 2$  maxpooling layer. Finally, the outputs of the maxpooling



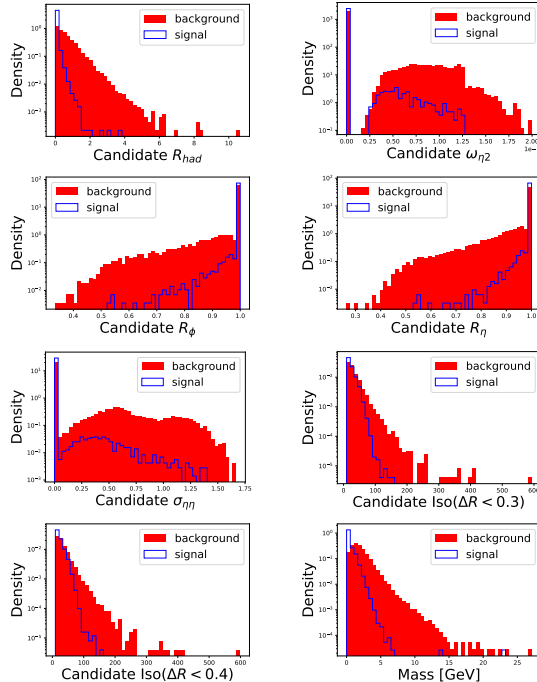


Figure 3.4: Distribution of signal electron (red) and background jets (blue) for seven existing typically-used high-level features, as well as for mass.

layer are flattened and concatenated with the high-level inputs to form a high-dimensional vector. This high-dimensional vector is then processed by a sequence of fully connected layers with rectified linear units, using dropout[59, 22]. The final output is produced by a single logistic unit and it can be interpreted as the probability of the input belonging to the signal class. The entire architecture is trained by stochastic gradient descent to minimize the relative entropy between the targets and the outputs, across all training examples.

For each combination of high-level variables, we also train and tune multi-layer, fully connected, neural networks with a similar sigmoidal logistic unit at the top,

All models were implemented using KERAS [33] with TENSORFLOW [10] as the backend and trained with a batch size of 128 with the ADAM optimizer [65]. The weights for all the models were initialized using orthogonal weights and each network was tuned using 150 iterations of bayesian optimizaton with the SHERPA hyperparameter optimization library [58]. Additional details about the hyperparameters and their optimization are given in Tables 3.3, 3.4 and 3.5.

## 3.7 Performance

Initial studies indicated that having images that reflect both  $E$  and  $E_T$  provided no performance boost, so only results with  $E_T$ -based images are shown here and used for further studies. A comparison of the performance of the image networks and the seven standard high-level features ( $R_{\text{had}}, \omega_{\eta 2}, R_\phi, R_\eta, \sigma_{\eta\eta}, \text{Iso}(\Delta R < 0.3), \text{Iso}(\Delta R < 0.4)$ ) is shown in Fig. 3.5 and described in Table 3.1.

Networks combining the standard high-level features (AUC of 0.945) do not match the performance of a network which analyzes the lower-level data expressed as images (0.972), indicating that the images contain additional, untapped information relevant to the identification of electrons. This is not unexpected, and is in line with similar results for jet substructure or flavor tagging [55, 18]. Networks which see only one of the ECal or HCal images but not both do not match this performance, supporting the intuition that both calorimeters contribute valuable information. Adding the HL features to the CNN, however, gives an almost negligible boost in performance, suggesting that the CNN has succeeded in capturing the power of the HL features.

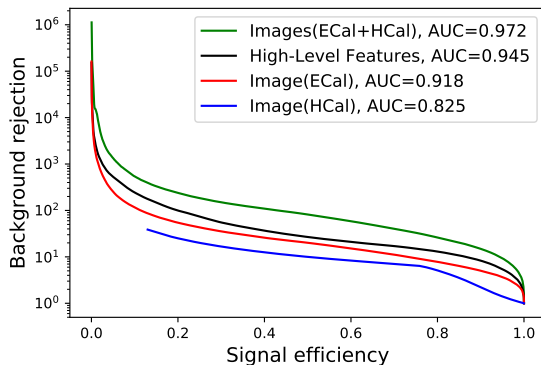


Figure 3.5: Comparison of the performance in electron identification for networks with varying sets of input features. Shown is the signal efficiency versus background rejection, and the AUC, for networks which use the existing set of expert high-level features (see text for details), networks which use HCal or ECal images, or both.

Table 3.1: Electron classification power (AUC) for networks with various feature sets. Images refer to low-level pixel data. Standard features are the high-level (HL) features typically used ( $R_{\text{had}}$ ,  $\omega_{\eta 2}$ ,  $R_{\phi}$ ,  $R_{\eta}$ ,  $\sigma_{\eta\eta}$ ,  $\text{Iso}(\Delta R < 0.3)$ ,  $\text{Iso}(\Delta R < 0.4)$ ), as described in the text. All AUC values have an uncertainty of  $\pm 0.001$  unless otherwise specified.

Network Features				AUC
Images		7 Standard HL Features	$M_{\text{jet}}$	
ECal	HCal			
	✓			$0.82 \pm 0.02$
✓				0.918
✓	✓			0.972
✓	✓	✓		0.973
		✓		0.945
		✓	✓	0.956

### 3.8 Bridging the gap

The performance of the deep CNN reveals that there is information in the low-level image that is not captured by the suite of existing high-level features. The goal, however, is not to replace the suite of features with an image-based network whose decisions are opaque to us and may not align with real physical principles. Instead, our aim is to identify new high-level features which bridge the gap between the existing performance and the superior performance of the CNN.

We note that the design of the high-level features focuses on highlighting the characteristics of the signal electrons, localized energy depositions primarily in the ECal without significant structure. The background, however, is due to jets, which potentially can exhibit a rich structure and comprise a mixture of jets from gluons, light quarks, and heavy quarks. Each parton may produce jets with a distinct structure and varying probability to mimic electrons. We hypothesize that features which are sensitive to the structure of the jet, or subclasses of jets, may provide additional discrimination power.

We first consider the powerful feature of jet mass,  $M_{\text{jet}}$ , which is not often applied to electron identification, but has a distinct marginal distribution for electrons and jets, see Fig. 3.4.

Including it in a network of HL features provides a small but distinct boost in performance, see Table 3.1, indicating that it contains useful information for this classification task not duplicated by the standard seven HL features. This encourages us to explore further the space of jet observables as a way to understand the source of additional classification power of the CNN.

### 3.8.1 Set of Observables

One could in principle consider an infinite number of jet observables. To organize our search, we use the Energy Flow Polynomials (EFPs) [67], a large (formally infinite) set of parameterized engineered functions, inspired by previous work on energy correlation functions [70], which sum over the contents of the cells scaled by relative angular distances.

These parametric sums are described as the set of all isomorphic multigraphs where:

$$\text{each node} \Rightarrow \sum_{i=1}^N z_i, \tag{3.5}$$

$$\text{each } k\text{-fold edge} \Rightarrow (\theta_{ij})^k. \tag{3.6}$$

The observable corresponding to each graph can be modified with parameters  $(\kappa, \beta)$ , where

$$(z_i)^\kappa = \left( \frac{p_{Ti}}{\sum_j p_{Tj}} \right)^\kappa, \tag{3.7}$$

$$\theta_{ij}^\beta = (\Delta\eta_{ij}^2 + \Delta\phi_{ij}^2)^{\beta/2}. \tag{3.8}$$

Here,  $p_{Ti}$  is the transverse momentum of cell  $i$ , and  $\eta_{ij}$  ( $\phi_{ij}$ ) is pseudorapidity (azimuth) difference between cells  $i$  and  $j$ . The original IRC-safe EFPs require  $\kappa = 1$ , however we consider examples with  $\kappa \neq 1$  to explore a broader space of observables. Also, note that

$\kappa > 0$  generically corresponds to IR-safe but C-unsafe observables.<sup>1</sup>

In principle, the space is complete, such that any jet observable can be described by one or more EFPs of some degree; in practice, the space is infinite and only a finite subset can be explored. We consider EFPs up to degree  $d = 7$  and with  $\beta$  values of  $[\frac{1}{2}, 1, 2]$  and  $\kappa$  values of  $[-1, 0, 1, 2]$ . We consider each graph as applied to the ECal or the HCal separately, effectively doubling the number of graphs<sup>2</sup>.

### 3.8.2 Searching for Observables

Rather than conduct a brute-force search of this large space, we aim to leverage the success of the CNN and find observables which mimic its decisions. We follow the black-box guided algorithm of Ref. [46], which isolates the portion of the input space where the CNN and existing HL features disagree and searches for a new observable that matches the decisions of the CNN algorithm in that subspace.

The subspace is defined as input pairs  $(x, x')$  that have a different relative ordering between the CNN and the network of  $n$  HL features ( $\text{HLN}_n$ ). Mathematically, we express this using the *decision ordering* (DO)

$$\text{DO}[f, g](x, x') = \Theta\left((f(x) - f(x'))(g(x) - g(x'))\right), \quad (3.9)$$

where  $f(x)$  and  $g(x)$  are classification functions such as the CNN or the  $\text{HLN}_n$ , such that  $\text{DO} = 0$  corresponds to inverted ordering and  $\text{DO} = 1$  corresponds to the same ordering. The

---

<sup>1</sup>For  $\kappa < 0$ , empty cells are omitted from the sum.

<sup>2</sup>We also explored a version where ECal and HCal information were used simultaneously by each graph, but found no improvement.

focus of our investigation are the set of pairs  $X_n$  where the two classifiers disagree, defined as


$$X_n = \left\{ (x, x') \mid \text{DO}[\text{CNN}, \text{HL}_n](x, x') = 0 \right\}. \quad (3.10)$$

As prescribed in Ref. [46], we scan the space of EFPs to find the observable that has the highest average decision ordering (ADO) with the CNN when averaged over the disordered pairs  $X_n$ . The selected EFP is then incorporated into the new network of HL features,  $\text{HLN}_{n+1}$ , and the process is repeated until the ADO or AUC plateaus.

For all  $\text{HLN}_n$  used in this search, models were trained with KERAS [33] using TENSORFLOW [10] as the backend. Each model was built as a fully connected neural network of simple one dimensional input features and a single logistic unit output. These networks consisted of 3 hidden layers, each with 50 rectified linear units, separated by 2 dropout layers using a dropout value of 0.25 and trained with a batch size of 128. The ADAM optimizer [65] was used with learning rate of 0.001 and initialized with glorot normal weights.

### 3.8.3 IRC safe observables


We begin our search by considering only the observables which are IRC safe, with  $\kappa = 1$ . Beginning with the seven HL features, the first graph selected is:




$$= \sum_{a,b=1}^N z_a z_b \theta_{ab}^{\frac{1}{2}}$$

with  $\beta = \frac{1}{2}$ . This graph has an ADO of 0.802 with the CNN over the input subspace where the CNN disagrees with the seven HL, suggesting that it is well aligned with the CNN's strategy. Adding it to the seven HL features achieves an AUC of  $0.970 \pm 0.001$ , very nearly closing the gap with the CNN performance of 0.972. This graph is very closely related to jet mass, a pairwise sum over cells which folds in angular separation, but more closely resembles the Les Houches Angularity variable [53], which similarly is sensitive to the distribution of energy away from the center, though with a smaller power of the angularity than jet mass, which suggests that it enhances small angles. Additional scans do not identify EFP observables with a useful ADO and do not contribute to the AUC.

If instead, we begin with the seven HL features as well as the jet mass, the procedure selects two graphs:

$$
= \sum_{a \dots h=1}^N z_a z_b z_c z_d z_e z_f z_g z_h \theta_{ab} \theta_{ac} \theta_{ad} \theta_{ae} \theta_{af} \theta_{ag} \theta_{ah}$$

and

$$
= \sum_{a,b,c=1}^N z_a z_b z_c \theta_{ab}^{\frac{1}{2}} \theta_{bc}^{\frac{1}{2}} \theta_{ac}^{\frac{1}{2}}$$

When combined with the seven HL features and  $M_{\text{jet}}$ , this set of ten observables achieves an AUC of  $0.971 \pm 0.001$ , almost matching the CNN performance. Distributions of these observables for signal and background samples are shown in Fig. 3.6.

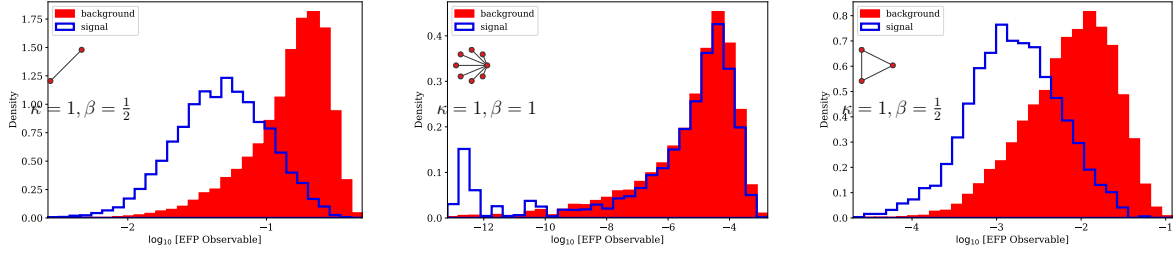


Figure 3.6:  $\log_{10}$  distributions of the selected IRC-safe EFPs as chosen by the black-box guided strategy, for signal electrons and background jets.

### 3.8.4 Broader Scan

In this Section, we present a scan of a larger set of EFPs, including values of  $\kappa$  which lead to IRC unsafe observables,  $\kappa = [-1, 0, 1, 2]$ .

Beginning from the seven standard HL features, the first pass selects a simple observable:

$$\bullet = \sum_{a=1}^N z_a^2$$

with no angular terms at all, but  $\kappa = 2$ . This is known in the jet substructure literature as  $p_T^D$  [74, 30] and was originally developed to help distinguish between quark and gluon jets. When combined with the other seven HL features, this observable also reaches a performance of  $0.970 \pm 0.001$ . Further scans do not lead to statistically significant improvements in AUC.

If instead, we begin from the seven standard HL features and  $M_{\text{jet}}$ , we find  $\star$ , this time with  $\kappa = 2$  as well as the simpler  $p_T^D$ . Distributions of these two IRC unsafe EFP observables for signal and background are shown in Fig. 3.7. Together with the seven HL and  $M_{\text{jet}}$ , these 10 observables reach a performance of  $0.971 \pm 0.001$ . Further scans do not lead to statistically significant improvements in AUC.



Table 3.2: Summary of the performance of various networks considered. Uncertainty in the AUC value is  $\pm 0.001$ , estimated using bootstrapping.

Base	Additions $(\kappa, \beta)$		(AUC)	
7HL			0.945	
7HL	$+M_{\text{jet}}$		0.956	
7HL		$\swarrow (1, \frac{1}{2})$	0.970	
7HL	$+M_{\text{jet}}$	$\star (1, 1)$	$\triangleright (1, \frac{1}{2})$	0.971
7HL		$\cdot (2, -)$	0.970	
7HL	$+M_{\text{jet}}$	$\star (2, 1)$	$\cdot (2, -)$	0.971
CNN			0.972	

See Table 3.2 for a summary of the additional observables needed to reach the performance of  $\approx 0.97$  in each case.

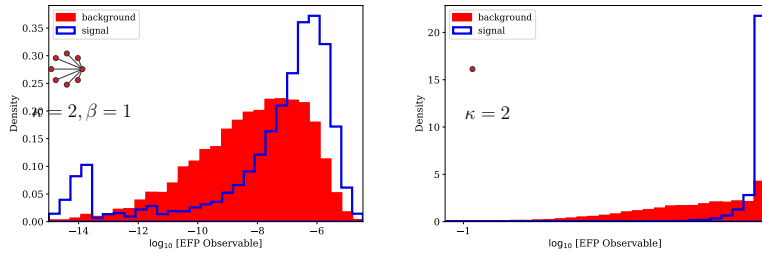
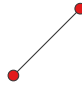


Figure 3.7:  $\log_{10}$  distributions of the selected EFPs as chosen by the black-box guided strategy, regardless of IRC safety, for signal electrons and background jets.

### 3.9 Discussion


Our deep neural networks indicate that low-level calorimeter data represented as images contains information useful for the task of electron identification that is not captured by the standard set of high-level features as implemented here.

A guided search [46] through the EFP space identified two EFP observables calculated on the ECal cells which mimic the CNN strategy and bridge the gap. Observables on the HCal information were not helpful to the classification task. The first,



$$= \sum_{a,b=1}^N z_a z_b \theta_{ab}^{\frac{1}{4}}$$

is closely related to the Les Houches Angularity [53], and confirms our suspicion that the non-trivial structure of the background object provides a useful handle for classification. The second observable,  $p_T^D$  [74, 30],



$$= \sum_{a=1}^N z_a^2$$

with  $\kappa = 2$  is not IRC safe, and was originally developed to help distinguish between quark and gluon jets. It effectively counts the number of hard particles, which is sensitive to the amount of color charge, where electrons and jets are clearly distinct.

Both Les Houches Angularity and  $p_T^D$  display power to separate electrons from the jet backgrounds, by exploiting the structure and nature of the jet energy deposits. While the precise performance obtained here may depend at some level on the fidelity of the simulation used and the resulting limitations on the implementation of state-of-the-art high-level features, these results strongly suggest that these observables be directly studied in experimental contexts where more realistic simulation tools are available, or directly in data samples, using weakly supervised learning [43].

More broadly, the existence of a gap between the performance of state-of-the-art high-level features and CNN represents an opportunity to gather additional power in the battle to suppress lepton backgrounds. Rather than employing black-box CNNs 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.

Any boost in electron identification performance is extremely valuable to searches at the LHC, especially those with multiple leptons, where event-level efficiencies depend sensitively on object-level efficiencies.

All code and data used in this project is available at: <https://github.com/TDHTTTT/EID>, as well as through the UCI Machine Learning in Physics web portal at: <http://mlphysics.ics.uci.edu/>.

---

## 3.10 Neural Network Hyperparameters and Architecture

Table 3.3: Hyperparameter ranges for bayesian optimization of convolutional networks

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

Table 3.4: Hyperparameter ranges for bayesian optimization of fully connected networks

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

Table 3.5: Best hyperparameters found per model.

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

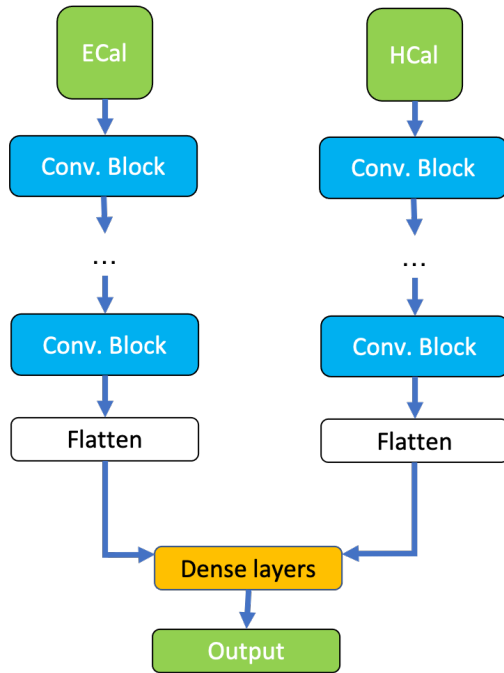


Figure 3.8: Diagram of the architecture of the convolutional neural network.

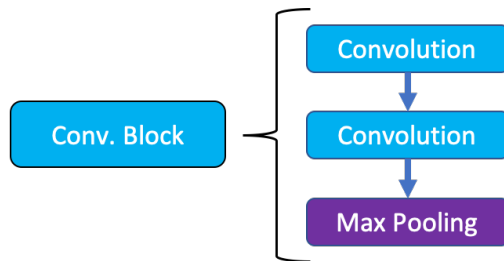


Figure 3.9: Diagram of convolutional block appearing in network architecture, see Fig 3.8.

# Chapter 4

## Learning to Isolate Muons

### 4.1 Abstract

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 partially close the performance gap with networks which analyze the calorimeter cells directly. These observables are theoretically well-defined and can be applied to studies of collider data. The remaining performance gap suggests the need for a new class of calorimeter-based observables.

## 4.2 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 [1, 77, 64] as well as low-mass resonances [61].

The current state of the art strategy for distinguishing prompt and non-prompt muons in experimental searches is the robust and simple approach of measuring the isolation of the muon in the calorimeter, as

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

within a cone  $R = \sqrt{\Delta\phi^2 + \Delta\eta^2} < R_0$  surrounding the muon [6], where typically a single cone is used, and values of  $R_0$  range from 0.1-0.4. This approach focuses on identifying a typical characteristic of the signal, low calorimeter activity in the vicinity of the muon.

This 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 [9, 56], pions [14] or electrons [37]. Other studies have shown that muons which fail the traditional isolation requirement can

contain significant power to reveal new physics [27].

At the same time, there have been significant advances in machine learning techniques and their applications in physics [23, 17], 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 [34, 18] or as an unordered list [68].

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 develop a set of simple, interpretable calorimeter features which capture a good fraction of that additional classification power. We suggest a new class of calorimeter observables which may capture the remaining unused information.

### 4.3 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, especially when the dimensionality of the data is large and the training is done with simulated samples, as is the case for networks which directly use the calorimeter cells. This 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 [46]. These high-level observables are more compact, physically interpretable, can be validated in data, and allow the straightforward assessment and propagation of systematic uncertainties.

### 4.3.1 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 [13], showered and hadronized with PYTHIA [79], and the detector response simulated with DELPHES [39] using the standard ATLAS card. 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 future LHC experimental data.

Muons in the range  $p_T \in [10, 15]$  GeV were considered, and the signal samples are weighted such that the transverse muon momentum distributions match that of the background. Only events where a muon is identified as a track in the muon spectrometer are used. In total

there were 91,592 events used, where 47,616 were signal and 43,976 were background.

Calorimeter deposits can be represented as images where each pixel value represents the  $E_T$  deposited by a particle [34]. 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.

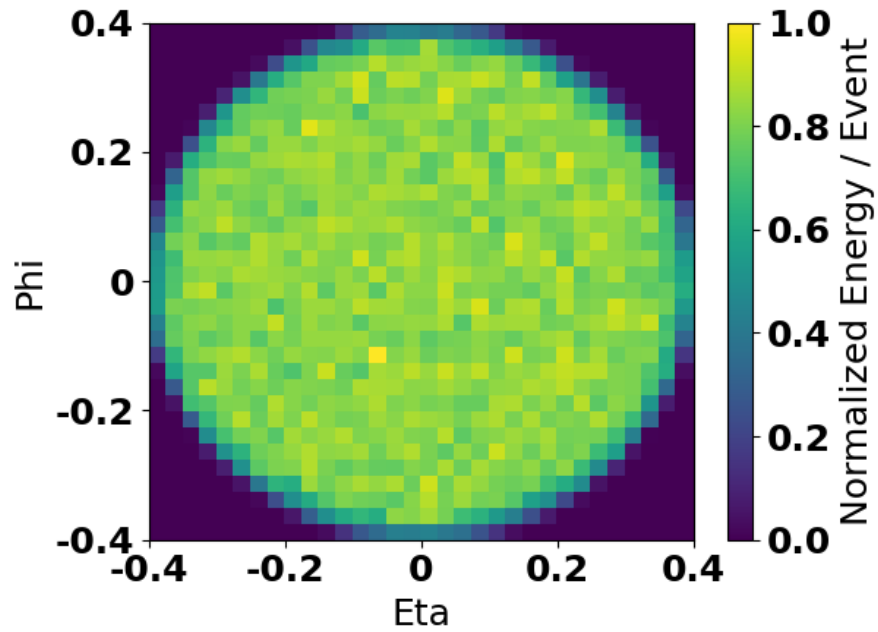
We choose a 32x32 grid, which approximately corresponds with the calorimeter granularity of ATLAS and CMS. Heat maps of the calorimeter energy deposits in  $\eta - \phi$  space for both signal prompt muons and background non-prompt muons are shown in Fig. 4.1. The signal calorimeter deposits are uniform 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.

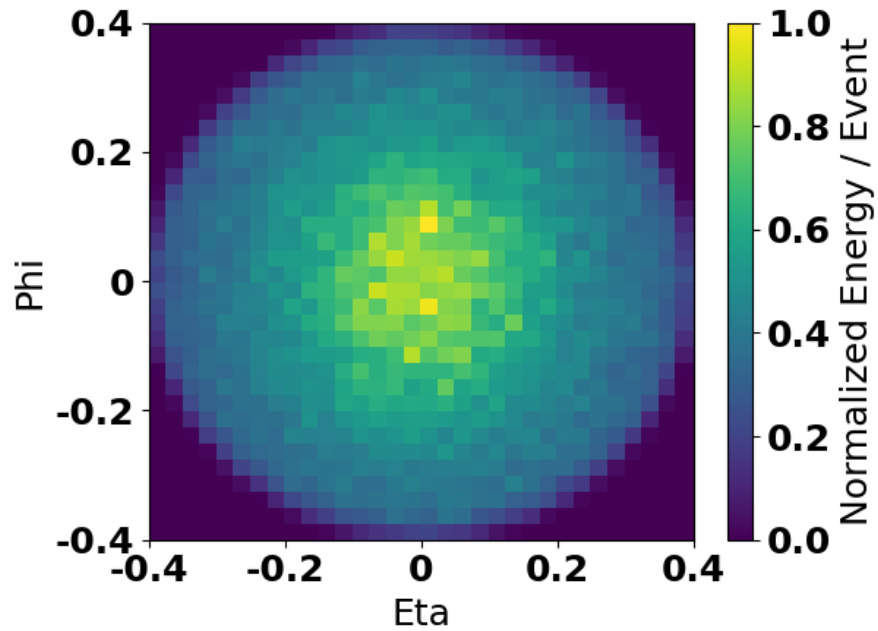
Crucially, these isolation observables and all other calorimeter observables are calculated directly from the pixels of the muon images, ensuring that they contain a strict subset of the information available. This allows for direct and revealing comparisons of the performance between networks trained with the images and those trained with  $I_\mu$ . Note that pixelization of the detector may incur some loss of information relative to the underlying segmentation of the calorimeter, but our studies examines the relative powers of the techniques, rather absolute comparisons with more realistic scenarios.

## 4.4 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. Accessing the



(a) Mean Prompt Muon



(b) Mean Non-prompt Muon

Figure 4.1: 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  $R = 0.4$ . The color of each cell represents the sum of the  $E_T$  of the calorimeter deposits within the cell.

calorimeter information at the lowest-level and highest-dimensionality, particle-flow networks (PFN) [68] operate on unordered lists of calorimeter cells, while convolutional networks (CNN) are applied to the muon images [34, 18]. Smaller 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 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 standard approach of using a single isolation cone yields an AUC of 0.780 for the optimal cone size,  $R_0 = 0.425^1$ . The muon image network achieves a significantly higher performance, with an AUC of 0.842, and the particle flow network reaches 0.848. 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 [68] (EFN), which enforces infra-red and collinear (IRC) safety, achieves nearly the same performance, 0.843. This suggests that most of the additional information beyond the isolation cones is IRC-safe.

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 0.785. To estimate the full information available in the cones, we perform a greedy search through all 18 cones; we find that a set of 8 cones,  $[0.15, 0.175, 0.2, 0.225, 0.25, 0.3, 0.35, 0.4]$ , yields another small boost in classification power up to an AUC of 0.794, as shown in Fig. 4.2. Performance was fairly insensitive to the specific choices of cone sizes, and does not grow significantly beyond eight cones. However, a gap remains between performance of isolation cones and the calorimeter cell networks as seen Fig. 4.3 and Table 4.1.

These results support the conventional wisdom that a significant fraction of the information

---

<sup>1</sup>Similar performance was seen for other cone sizes.

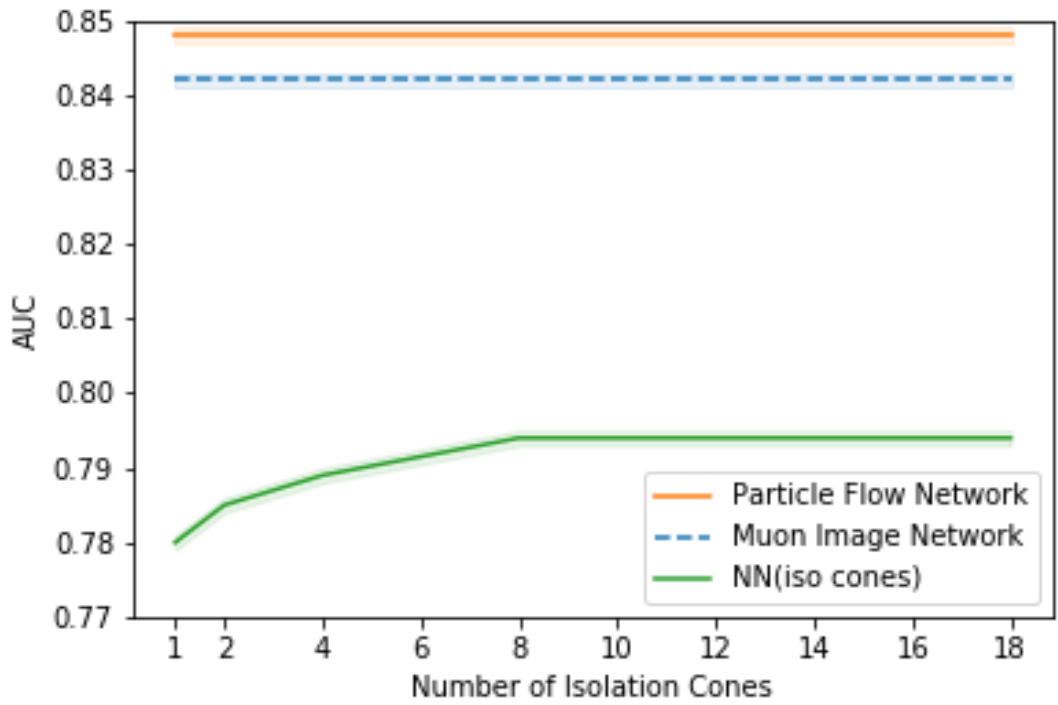


Figure 4.2: Comparison of classification performance using the performance metric AUC between Particle-Flow networks trained on unordered lists of calorimeter deposits (orange, solid), convolutional networks trained on muon images (blue, dashed) and networks which use increasing numbers of isolation cones (green, solid). For each number of cones, the optimal set is chosen.

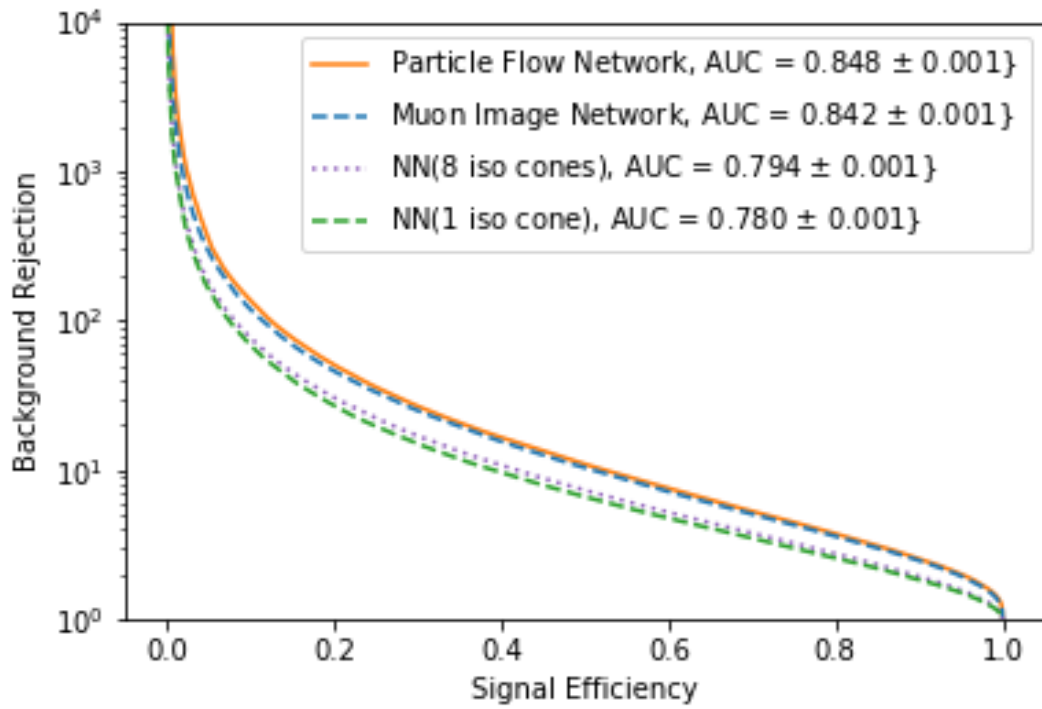


Figure 4.3: Background rejection versus signal efficiency for Particle-Flow networks trained on unordered lists of calorimeter deposits (orange, solid), convolutional networks trained on muon images (blue, dashed), networks trained on a set of isolation cones (purple, dotted) and the benchmark approach, a single isolation cone approach (green, dashed).

relevant for classification is captured by a single, simple cone. However, they also indicate that there is additional information in the radial distribution of energy, which can be captured by using multiple cones. Most intriguingly, 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 by isolation cones.

## 4.5 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.

### 4.5.1 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 [46]. This allows us to understand the nature of the information being used as well as to represent it more compactly.

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) [67], a formally infinite set of parameterized engineered functions, inspired by previous work on energy correlation functions [70], which sum over the contents of the cells scaled by relative angular distances.

These parametric sums are described as the set of all isomorphic multigraphs where:

$$\text{each node} \Rightarrow \sum_{i=1}^N z_i, \tag{4.1}$$

$$\text{each } k\text{-fold edge} \Rightarrow (\theta_{ij})^k. \tag{4.2}$$

The observable corresponding to each graph can be modified with parameters  $(\kappa, \beta)$ , where

$$(z_i)^\kappa = \left( \frac{p_{Ti}}{\sum_j p_{Tj}} \right)^\kappa, \tag{4.3}$$

$$\theta_{ij}^\beta = (\Delta\eta_{ij}^2 + \Delta\phi_{ij}^2)^{\beta/2}. \tag{4.4}$$

Here,  $p_{Ti}$  is the transverse momentum of cell  $i$ , and  $\Delta\eta_{ij}$  ( $\Delta\phi_{ij}$ ) is pseudorapidity (azimuth) difference between cells  $i$  and  $j$ . 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 observables<sup>2</sup>.

In principle, the space spanned by the EFPs is complete, such that any jet observable can

---

<sup>2</sup>Also, note that  $\kappa > 0$  generically corresponds to IR-safe but C-unsafe observables. For  $\kappa < 0$ , empty cells are omitted from the sum.



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. [46], which iteratively assembles a set of EFPs that mimic the decisions of another guiding network (the CNN or 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:


$$\text{DO}[f, g](x, x') = \Theta\left((f(x) - f(x'))(g(x) - g(x'))\right), \quad (4.5)$$

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. [46], 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,  $\text{HLN}_{n+1}$ , and the process is repeated until the ADO plateaus.


### 4.5.2 IRC Safe Observables


We begin our search by considering only a small set of simple observables, those which are IRC safe ( $\kappa = 1$ ), have a simple angular weighting ( $\beta \in [1, 2]$ ), and are limited to a small number of nodes  $n \leq 3$  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:

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

which, when combined with the isolation cones and  $\sum p_T$ , yields an AUC of 0.813 and an ADO with the CNN of 0.897, a significant boost relative to just using the radial information of the isolation cones. The subsequent scans produce variants of this observable :

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

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

$$
= \sum_{a,b=1}^N z_a z_b \theta_{ab}$$

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 0.821 and an ADO with the CNN of 0.908, see Table 4.1.

This set of observables partially closes the performance gap with the calorimeter cell net-

works, indicating that angular information is relevant to the muon isolation classification task, but fails to fully match its performance. Further scans in this limited space do not yield significant boost in AUC or ADO values. Distributions of these EFPs for signal and background are shown in Fig. 4.4.

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

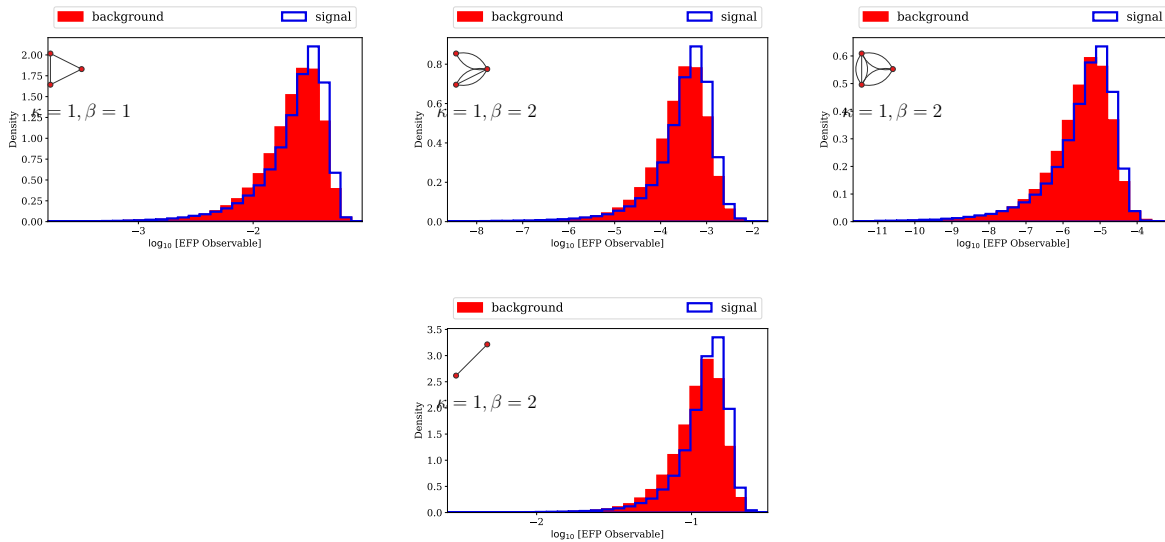


Figure 4.4: Distributions of the  $\log_{10}$  of the selected IRC-safe EFPs as chosen by the black-box guided strategy, for prompt (signal) muons and non-prompt (background) muons.

### 4.5.3 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 10 which, when combined with the isolation cones and  $\sum p_T$  reach an AUC of 0.827. Due to the overlapping nature of the large space of EFPs,

Table 4.1: Summary of performance (AUC) in the prompt muon classification task for various network architectures and input features. Statistical uncertainty in each case is  $\pm 0.001$  with 95% confidence, measured using bootstrapping over 200 models. Uncertainty due to the initial conditions of the network is found to be negligible.

Method	AUC	ADO[PFN]
Single Iso Cone	0.780	0.865
8 Iso	0.794	0.885
8 Iso + $\sum p_T$ + 1 IRC-safe EFPs	0.813	0.897
8 Iso + $\sum p_T$ + 4 IRC-safe EFPs	0.821	0.908
8 Iso + $\sum p_T$ + 10 IRC-unsafe EFPs	0.827	0.923
Calo image CNN	0.842	0.949
Calo cell Energy-Flow Net	0.843	0.947
Calo cell Particle-Flow Net	0.848	1

there are many sets which achieve similar performance. Rather than focusing on the specific EFPs selected, we take the value of this plateau as a measure of the power contained in our finite subset of the formally infinite space of EFPs. Again, a similar scan guided by the CNN rather than the PFN yields very similar results.

## 4.6 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 4.1. A guided search through the space of EFPs closes approximately half of the gap between these networks, giving us some insight as to the nature of the information. However, given that the set of EFPs are formally complete, the remaining gap presents an interesting puzzle. Why is there no EFP which can capture the information used by the calorimeter-cell networks?

One clue lies in the assumptions that underly the claim that EFPs are a complete basis for IRC safe observables. Specifically, it is assumed that the calorimeter cell inputs are rotationally and translationally invariant, such that a transformation does not affect the value of the observable. In this case, however, an important element of the learning task violates

that assumption: the location of the muon at the center of the image. As a consequence, the EFPs do not have access to the information about the relative angle between a cell and the muon location, which is clearly important for this task<sup>3</sup>. Instead, they can only access angular information between cells. The particle-flow network, in contrast, does not assume this invariance, and can learn that the angle relative to the center of the image is important. An extension of the EFP sets which includes an additional node of another class, to indicate the location of the muon, would likely close the performance gap, but is beyond the scope of this work.

## 4.7 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 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 reliably extrapolated from control regions in collider data. Any boost in muon identification performance is extremely valuable to searches at the LHC, especially those with multiple leptons, where event-level efficiencies depend sensitively

---

<sup>3</sup>We thank Jesse Thaler for discussions on this point.

on object-level efficiencies.

Additional, non-IRC safe EFPs provide a further modest boost in performance, but does not close the gap with the PFN and CNN, suggesting that additional information remains to be extracted. It is possible that the remaining information could be captured by more complex observables we have not included in our EFP subset, or require an extension of the EFP observables to include information such as the location of the muon. 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. While these studies were performed with simulated samples, similar studies can be performed using unsupervised methods [43, 73] on samples of collider data, which we leave to future studies.

# Chapter 5

## Conclusion

The rapid development of deep learning methods in recent years has opened new possibilities to improve classification performance in areas such as computer vision, natural language processing, medicine, and others. In this thesis, we show it is possible to use deep learning methods to improve the performance of particle identification and reconstruction in high-energy physics experiments. This is possible by using neural networks directly in high dimensional low-level detector information, instead of following the approach of traditional methods by using physics-based heuristics to reduce the dimensionality of the data beforehand. Our results suggest traditional methods are losing or distorting information when performing the dimensionality reduction.

We have shown that neural networks are able to improve performance, at the cost of interpretability. However, it is also possible to recover some or all of this performance by using the same network to create new interpretable variables which imitate the decisions of the network. In the case where it was not possible to close the gap between the new interpretable variables and the network, the results suggest the need and a roadmap for more complex interpretable variables. Therefore, it is possible to use neural networks to improve

performance while maintaining interpretability, which is of paramount importance for using machine learning models for physics.

My thesis is one of the first steps in showing the effectiveness of deep learning in high-energy physics. The improvements I have shown by applying deep learning methods are of critical importance to high-energy physics experiments such as the Large Hadron Collider (LHC) since the improvements of classification performance provide evidence to refute or validate physical theories. The complexity of the detectors, the experiments and data representations in high-energy physics and physical sciences offers great opportunities for future specialized developments of deep learning models and I have no doubt this field will continue to grow in the future.



# Bibliography

- [1] 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.
- [2] G. Aad et al. The ATLAS Experiment at the CERN Large Hadron Collider. *JINST*, 3:S08003, 2008.
- [3] G. Aad et al. Search for supersymmetry in final states with jets, missing transverse momentum and one isolated lepton in  $\sqrt{s} = 7$  TeV pp collisions using  $1 \text{ fb}^{-1}$  of ATLAS data. *Phys. Rev. D*, 85(1):012006, 2012. [Erratum: *Phys.Rev.D* 87, 099903 (2013)].
- [4] G. Aad et al. A search for top squarks with R-parity-violating decays to all-hadronic final states with the ATLAS detector in  $\sqrt{s} = 8$  TeV proton-proton collisions. *Arxiv*, 2016.
- [5] G. Aad et al. Measurement of  $w^\pm$  and  $z$ -boson production cross sections in  $pp$  collisions at  $\sqrt{s} = 13$  tev with the atlas detector. *Phys. Lett. B*, 759:601–621, 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. Performance of  $b$ -Jet Identification in the ATLAS Experiment. *JINST*, 11(04):P04008, 2016.
- [8] G. Aad et al. Search for single production of a vector-like quark via a heavy gluon in the  $4b$  final state with the ATLAS detector in  $pp$  collisions at  $\sqrt{s} = 8$  TeV. *Phys. Lett.*, B758:249–268, 2016.
- [9] R. Aaij et al. Search for Dark Photons Produced in 13 TeV  $pp$  Collisions. *Phys. Rev. Lett.*, 120(6):061801, 2018.
- [10] 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. Watkenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

- [11] S. Agostinelli et al. GEANT4: A Simulation toolkit. *Nucl. Instrum. Meth. A*, 506:250–303, 2003.
- [12] E. S. Almeida, A. Alves, N. Rosa-Agostinho, O. J. P. Eboli, and M. C. Gonzalez-Garcia. Electroweak Sector Under Scrutiny: A Combined Analysis of LHC and Electroweak Precision Data. *Phys. Rev. D*, 99(3):033001, 2019.
- [13] 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.
- [14] 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.
- [15] P. Baldi. <https://indico.cern.ch/event/395374/>, 2015. DataScience@LHC.
- [16] P. Baldi. The inner and outer approaches to the design of recursive neural architectures. *Data Mining and Knowledge Discovery*, 32, 01 2018.
- [17] P. Baldi. *Deep Learning in Science: Theory, Algorithms, and Applications*. Cambridge University Press, Cambridge, UK, 2020. In press.
- [18] 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, 2016.
- [19] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15:937–946, 1999.
- [20] P. Baldi and Y. Chauvin. Hybrid modeling, hmm/nn architectures, and protein applications. *Neural Comput.*, 8(7):1541–1565, Oct. 1996.
- [21] P. Baldi and G. Pollastri. The principled design of large-scale recursive neural network architectures—dag-rnns and the protein structure prediction problem. *J. Mach. Learn. Res.*, 4:575–602, Dec. 2003.
- [22] P. Baldi and P. Sadowski. The dropout learning algorithm. *Artificial Intelligence*, 210:78–122, May 2014.
- [23] P. Baldi, P. Sadowski, and D. Whiteson. Searching for Exotic Particles in High-Energy Physics with Deep Learning. *Nature Communications*, 5:4308, 2014.
- [24] P. Baldi, P. Sadowski, and D. Whiteson. Enhanced higgs boson to  $\tau^+\tau^-$  search with deep learning. *Phys. Rev. Lett.*, 114:111801, Mar 2015.
- [25] P. Berta, M. Spousta, D. W. Miller, and R. Leitner. Particle-level pileup subtraction for jets and jet shapes. *JHEP*, 06:092, 2014.

- [26] R. Brun and F. Rademakers. ROOT: An object oriented data analysis framework. *Nucl. Instrum. Meth. A*, 389:81–86, 1997.
- [27] 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.
- [28] M. Cacciari, G. P. Salam, and G. Soyez. The anti- $k_t$  jet clustering algorithm. *JHEP*, 04:063, 2008.
- [29] M. Cacciari, G. P. Salam, and G. Soyez. FastJet User Manual. *Eur. Phys. J.*, C72:1896, 2012.
- [30] S. Chatrchyan et al. Search for a Higgs boson in the decay channel  $H$  to  $ZZ^{(*)}$  to  $q$   $\bar{q}$   $\ell^- \ell^+$  in  $pp$  collisions at  $\sqrt{s} = 7$  TeV. *JHEP*, 04:036, 2012.
- [31] S. Chatrchyan et al. Identification of b-quark jets with the CMS experiment. *JINST*, 8:P04013, 2013.
- [32] S. Chatrchyan et al. Search for Supersymmetry in  $pp$  Collisions at  $\sqrt{s}=8$  TeV in Events with a Single Lepton, Large Jet Multiplicity, and Multiple b Jets. *Phys. Lett. B*, 733:328–353, 2014.
- [33] F. Chollet et al. Keras. <https://keras.io>, 2015.
- [34] J. Cogan, M. Kagan, E. Strauss, and A. Schwartzman. Jet-Images: Computer Vision Inspired Techniques for Jet Tagging. *JHEP*, 02:118, 2015.
- [35] A. Collaboration. Impact parameter-based b-tagging algorithms in the 7 TeV collision data with the ATLAS detector: the TrackCounting and JetProb algorithms. Technical Report ATLAS-CONF-2010-041, CERN, Geneva, Jul 2010.
- [36] T. A. Collaboration. Electron efficiency measurements with the ATLAS detector using the 2015 LHC proton-proton collision data. Technical Report ATLAS-CONF-2016-024, CERN, Geneva, Jun 2016.
- [37] J. Collado, J. N. Howard, T. Faucett, T. Tong, P. Baldi, and D. Whiteson. Learning to Identify Electrons. 11 2020.
- [38] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, and M. Selvaggi. DELPHES 3, A modular framework for fast simulation of a generic collider experiment. *JHEP*, 02:057, 2014.
- [39] J. de Favereau et al. DELPHES 3, A modular framework for fast simulation of a generic collider experiment. *JHEP*, 1402:057, 2014.
- [40] L. de Oliveira. <https://indico.cern.ch/event/395374/>, 2015. DataScience@LHC.
- [41] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, and A. Schwartzman. Jet-Images — Deep Learning Edition. *Journal of High Energy Physics*, 2016.

- [42] L. De Oliveira, B. Nachman, and M. Paganini. Electromagnetic Showers Beyond Shower Shapes. *Nucl. Instrum. Meth. A*, 951:162879, 2020.
- [43] L. M. Dery, B. Nachman, F. Rubbo, and A. Schwartzman. Weakly Supervised Classification in High Energy Physics. *JHEP*, 05:145, 2017.
- [44] P. Di Lena, K. Nagata, and P. Baldi. Deep architectures for protein contact map prediction. *Bioinformatics*, 28:2449–2457, 2012.
- [45] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2215–2223. Curran Associates, Inc., 2015.
- [46] T. Faucett, J. Thaler, and D. Whiteson. Mapping Machine-Learned Physics into a Human-Readable Space. 10 2020.
- [47] P. Frasconi, M. Gori, and A. Sperduti. A general framework for adaptive processing of data structures. *Trans. Neur. Netw.*, 9(5):768–786, Sept. 1998.
- [48] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.
- [49] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS’10)*. Society for Artificial Intelligence and Statistics, 2010.
- [50] X. Glorot, A. Bordes, and Y. Bengio. Deep Sparse Rectifier Neural Networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume JMLR W&CP 15, Fort Lauderdale, FL, USA, 2011.
- [51] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In G. Gordon, D. Dunson, and M. Dudík, editors, *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, 2011. PMLR.
- [52] C. Goller and A. Kuchler. Learning task-dependent distributed representations by back-propagation through structure. *IEEE International Conference on Neural Networks*, page 347–352, 1996.
- [53] P. Gras, S. Höche, D. Kar, A. Larkoski, L. Lönnblad, S. Plätzer, A. Siódmok, P. Skands, G. Soyez, and J. Thaler. Systematics of quark/gluon tagging. *JHEP*, 07:091, 2017.
- [54] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. *arXiv preprint arXiv:1503.04069*, 2015.

- [55] D. Guest, J. Collado, P. Baldi, S.-C. Hsu, G. Urban, and D. Whiteson. Jet Flavor Classification in High-Energy Physics with Deep Neural Networks. *Physical Review D*, 94, 2016.
- [56] Z. Hall and J. Thaler. Photon isolation and jet substructure. *JHEP*, 09:164, 2018.
- [57] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [58] 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>.
- [59] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.
- [60] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, July 2012.
- [61] 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.
- [62] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision*, pages 2146–2153, Sept. 2009.
- [63] V. Khachatryan et al. Performance of Electron Reconstruction and Selection with the CMS Detector in Proton-Proton Collisions at  $\sqrt{s} = 8$  TeV. *JINST*, 10(06):P06005, 2015.
- [64] 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.
- [65] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [66] P. T. Komiske, E. M. Metodiev, and M. D. Schwartz. Deep learning in color: towards automated quark/gluon jet discrimination. *JHEP*, 01:110, 2017.
- [67] P. T. Komiske, E. M. Metodiev, and J. Thaler. Energy flow polynomials: A complete linear basis for jet substructure. *JHEP*, 04:013, 2018.
- [68] P. T. Komiske, E. M. Metodiev, and J. Thaler. Energy Flow Networks: Deep Sets for Particle Jets. *JHEP*, 01:121, 2019.
- [69] P. T. Komiske, E. M. Metodiev, and J. Thaler. Metric space of collider events. *Physical Review Letters*, 123(4), 7 2019.

- [70] A. J. Larkoski, G. P. Salam, and J. Thaler. Energy Correlation Functions for Jet Substructure. *arXiv.org*, Apr. 2013.
- [71] A. Lusci, G. Pollastri, and P. Baldi. Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules. *Journal of chemical information and modeling*, 53:1563–1575, 2013.
- [72] C. Magnan and P. Baldi. Sspro/accpro 5: Almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning, and structural similarity. *Bioinformatics*, 30:2592–2597, 2014.
- [73] E. M. Metodiev, B. Nachman, and J. Thaler. Classification without labels: Learning from mixed samples in high energy physics. *JHEP*, 10:174, 2017.
- [74] F. Pandolfi. *Search for the Standard Model Higgs Boson in the  $H \rightarrow ZZ \rightarrow l^+l^-q\bar{q}$  Decay Channel at CMS*. PhD thesis, Zurich, ETH, New York, 2012.
- [75] P. Sadowski, J. Collado, D. Whiteson, and P. Baldi. Deep Learning, Dark Knowledge, and Dark Matter. In *Proceedings of the NIPS 2014 Workshop on High-energy Physics and Machine Learning*, volume 42 of *Proceedings of Machine Learning Research*, pages 81–87, Montreal, Canada, (2015). PMLR. <http://proceedings.mlr.press/v42/sado14.html>.
- [76] 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.
- [77] 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).
- [78] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [79] T. Sjostrand, S. Mrenna, and P. Z. Skands. PYTHIA 6.4 Physics and Manual. *JHEP*, 0605:026, 2006.
- [80] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the conference on empirical methods in natural language processing*, 1631:1642, 2013.
- [81] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [82] T. D. Team. Theano: A python framework for fast computation of mathematical expressions. *arXiv e-prints arXiv:1605.02688*, 2016.
- [83] A. Tegge, Z. Wang, J. Eickholt, and J. Cheng. Nncon: improved protein contact map prediction using 2d-recursive neural networks. *Nucleic acids research*, 37:515–518, 2009.

- [84] W. Waltenberger. RAVE: A detector-independent toolkit to reconstruct vertices. *IEEE Trans. Nucl. Sci.*, 58:434–444, 2011.
- [85] W. Waltenberger, W. Mitaroff, F. Moser, B. Pflugfelder, and H. V. Riedel. The RAVE/VERTIGO vertex reconstruction toolkit and framework. *J. Phys. Conf. Ser.*, 119:032037, 2008.
- [86] L. Wu and P. Baldi. Learning to play go using recursive neural networks. *Neural Networks*, 21(9):1392 – 1400, 2008.

# Appendix A

## Learning to Identify Muons Appendix

### A.1 Neural Network Architectures

All networks were trained in Tensorflow[10] and Keras[33]. The networks were optimized with Adam [65] for up to 100 epochs with early stopping. For all networks except the PFNs, the weights were initialized using orthogonal weights[76]. Hyperparameters were optimized using bayesian optimization with the Sherpa hyperparameter optimization library [58]. 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.

#### A.1.1 Muon Image Networks

The pixelated images were preprocessed to have zero mean and unit standard deviation. The best muon image network structure begins with three convolutional blocks. Each block contains two convolutional layers with 56 filters with rectified linear units [51], followed by



a 2x2 pooling layer. Afterwards there are four fully connected layers with 178 rectified linear units and a final layer with a sigmoidal logistic activation function to classify signal vs background. The model had dropout [81, 22] with value 0.2062 on the fully connected layers and an initial learning rate of 0.0002 and batch size of 128.

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

Parameter	Range	Value
Num. of convolutional blocks	[1, 3]	3
Num. of filters	[16, 128]	56
Num. of fully connected layers	[2, 5]	4
Number of hidden units	[25, 200]	178
Learning rate	[0.0001, 0.01]	0.0002
Dropout	[0.0, 0.5]	0.2062

### A.1.2 Particle-Flow Networks

The Particle Flow Network (PFN) is trained using the `energyflow` package[69]. 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.

### A.1.3 Isolation Cone Networks

The isolation inputs 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 4 fully connected layers with 179 rectified linear hidden units[51] and a learning rate of 0.0002 and dropout rate of 0.0160.

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

Parameter	Range	ISO Value
Num. of layers	[2, 8]	4
Num. of hidden units	[1, 200]	179
Learning rate	[0.0001, 0.01]	0.0002
Dropout	[0.0, 0.5]	0.0160

#### A.1.4 Isolation Cone and EFP Networks

For all trained models using a combination of isolation cone and EFP features, a single architecture was chosen. The isolation and EFP inputs are preprocessed by subtracting the mean and dividing by the variance. Each trained neural network uses four fully connected hidden layers with 150 rectified linear hidden units[51] and a learning rate of 0.0001. Finally, an output layer with sigmoidal logistic activation function is used to predict the probability of signal or background.