Exploring Machine Learning Models to Improve the Classification of Displaced Hadronic Jets in the ATLAS Calorimeter

by

Rodrigue de Schaetzen

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

BSc, Combined Honours Computer Science and Physics

in

THE FACULTY OF SCIENCE
(Physics and Astronomy)

The University of British Columbia
(Vancouver)

April 2020

Abstract

The Large Hadron Collider (LHC) has yet to find new physics that could address the Standard Model's (SM) large open questions such as the composition of Dark Matter and the matter-antimatter asymmetry of the universe. There have been recent searches for Hidden Sector (HS) particles through the investigation of pair-production of neutral long-lived particles (LLPs) in proton-proton collisions. The ATLAS collaboration recently published results using a partial dataset from a search for paired LLP decays that produce displaced hadronic jets in the ATLAS calorimeter. Several classification models have been studied to identify these LLP decays, including boosted decision trees and LSTMs. In this analysis, 1D convolutional layers were added to an existing model architecture, which significantly improved the performance. Following hyperparameter optimization, the proposed model achieved a ROC AUC score of 0.97; a 10% relative improvement over the previous model.

Table of Contents

List of Tables

List of Figures

Glossary

ATLAS A Toroidal LHC ApparatuS

AUC Area Under Curve

BIB Beam Induced Background

CMS Compact Muon Solenoid

CNN Convolutional Neural Network

ECAL Electromagnetic Calorimeter

FPR False Positive Rate

HCAL Hadronic Calorimeter

HS Hidden Sector

ID Inner Detector

IID Independent and Identically Distributed

LHC Large Hadron Collider

LLP long-lived particles

LSTM Long Short-Term Memory

QCD Quantum Chromodynamics

RNN Recurrent Neural Network

ROC Receiver Operating Characteristic

SM Standard Model

CV Cross Validation

p_T Transverse momentum

Acknowledgments

Thank you to my supervisor Dr. Alison Lister and to Ph.D student Félix Cormier for their feedback and continued support during this project.
I would like to also thank my roommate and my dad for their insightful comments and suggestions for this paper.

Chapter 1

Introduction

This chapter provides a brief introduction to the world of particle physics and the motivation for this study.

1.1 The Standard Model

The Standard Model (SM) of particle physics describes our current understanding of the fundamental particles and their interactions. However, many open questions such as the matter-antimatter asymmetry of the universe [4], and the composition of Dark Matter [5] cannot be answered by the Standard Model, rendering it an incomplete model. In response, physicists have proposed new models which extend the SM in order to address some of its limitations. One class of such models are Hidden Sector (HS) models [6] [7]. These models predict a new set of particles, only weakly coupled to the SM, resulting in experimental signatures containing particles not charged under the SM (so not visible in the detectors) that decay to SM particles with a measurable lifetime. HS models could provide answers to the open SM questions presented above, in particular the nature of Dark Matter.

The components of the SM shown in Figure 1.1 are the fundamental building blocks of all ordinary matter. For example, subatomic particles such as protons and neutrons are each composed of three valence quarks and held together in a band state by the strong force, i.e. gluons. This is the definition of a hadron.

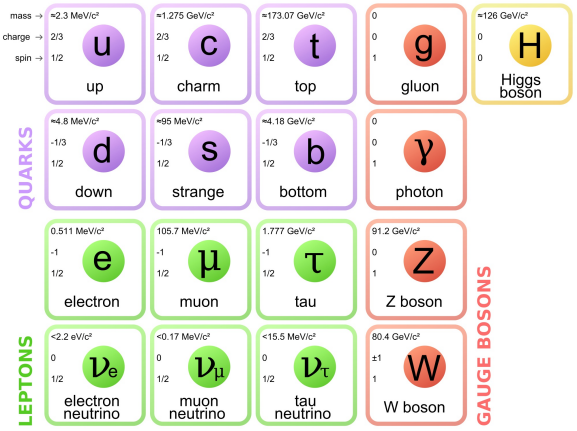

Figure 1.1: The Standard Model of particle physics. It describes our current understanding of the fundamental particles and their interactions. The model is considered incomplete due to its inability to answer several major questions about the universe, most notably the composition of Dark Matter. Image taken from [1].

1.2 The ATLAS Experiment

This study will use simulation from A Toroidal LHC ApparatuS (ATLAS), a general-purpose particle detector containing many layers of sub-detectors. It is located at the world's largest particle accelerator, the Large Hadron Collider (LHC) situated along the Swiss-French border. Along with Compact Muon Solenoid (CMS), another detector around the collider, ATLAS measurements observed the long-predicted Higgs Boson in 2012 [8]. Inside the LHC, bunches of protons are accelerated to near light speeds in opposite directions around the ring and collide with each other at specific points. Specifically, during Run 2 of the LHC (2015-2018) the center of mass energy was 13 TeV for proton-proton collisions. This tremendous amount of energy along with a frequency of 10^8 physics-related events per second, generated countless particles that either decayed in the detectors, or left the apparatus undetected.

1.2.1 ATLAS detector sub-systems

Due to the variety of particles that arise from proton collisions, many layers of different detectors combined in ATLAS provide clues to identifying a particle. These clues make up what is known as a particle signature. Figure 1.2 illustrates the subsystems of ATLAS. Their descriptions are provided in the list below. It should be emphasized that the proton-proton collisions occur at the center of the detector.

1. **Inner Detector (ID):** The innermost layer of the detector measures the trajectories of electrically-charged particles. From the curvature of the reconstructed trajectories their momentum can be inferred.

2. **Electromagnetic Calorimeter (ECAL):** Calorimeters are devices designed to absorb and measure the energy of particles that reach them. The topology of calorimeters consists of clusters of cells. As such, the specific clusters a particular particle hits is of high relevance to particle identification. The ECAL is the first of two calorimeters in ATLAS and is specialized in measuring the energy of photons and electrons.

3. **Hadronic Calorimeter (HCAL):** The HCal measures the energy of hadrons (e.g. protons, neutrons).

4. **Muon Spectrometer:** The main purpose of the final layer is to track muons, an elementary particle similar to the electron as they, along with neutrinos, are not stopped in the calorimeters. The muon spectrometer is also segmented. The detector principle is similar to the ID in that it allows the reconstruction of the trajectory of charged particles in a magnetic field. The parts of the muon spectrometer exhibiting energy deposits (hits) consistent with a charged particle are called muon segments. Note, any charged particle that has not yet decayed in the calorimeters will be detected in this final layer.

1.2.2 Jets

A major component of hadron collider experiments is the reconstruction and analysis of jets which are roughly cone-shaped clusters of particles. Typically, jets

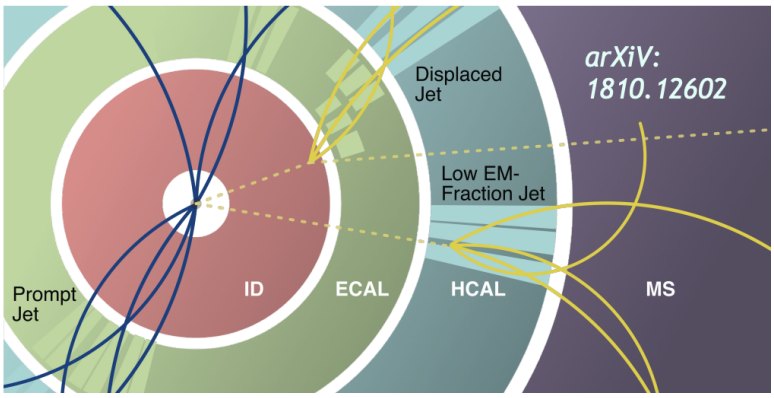

Figure 1.2: A diagram of a slice of ATLAS while looking down into the cylinder. The innermost layer (ID) is composed of detectors that track trajectories of charged particles. The next two layers are the electromagnetic calorimeter (ECal) and hadronic calorimeter (HCal). Calorimeters are designed to absorb the energy of particles that pass through them, eventually stopping the particle. Electrons and photons are absorbed in the ECal, while hadrons get absorbed in the HCal. The final layer is the muon spectrometer which is designed to measure muons and any other charged particle that has not yet been stopped. Figure from [2].

originate from quarks or gluons (elementary particles) that decay and radiate then form into hadrons (composite particles). The layers of the ATLAS detector make it possible to piece together the low-level and high-level characteristics of a particular jet. Tracks, calorimeter cluster deposits, and muon segments constitute the low level constituents of a jet. The high-level jet variables used in this study are given by the four-momentum of the jets, namely:

- **Pseudorapidity (η):** Describes the angle in relation to the axis of the detector cylinder, and thus beam axis.
- **Transverse momentum (p_t):** The component of momentum transverse/perpendicular to the axis of the detector.
- **Angle (ϕ):** The angle in the transverse plane.

1.3 The Central Problem

To date, no physics outside of the Standard Model has been discovered at the LHC. For this reason, researchers have broadened their search to more complex particle signatures. In particular, a number of studies search for particles that decay to SM particles only after a measurable distance in the detector. Many extensions to the SM theorize the existence of such long-lived particles (LLP). A paper published by the ATLAS collaboration ([9]) considers a heavy neutral boson decaying to a pair of neutral LLPs. In their search, the signature of interest is LLPs decaying in the ATLAS calorimeters. A model capturing the complexities of this signal was developed so that it could be differentiated from the highly abundant background. The initial classification model consisted of a Boosted Decision Tree; a relatively simple machine learning algorithm. An ongoing analysis has revamped this model to leverage the recent developments of highly complex machine learning algorithms. This approach builds off the successes of similar complex models applied in other physics analyses [10], [11].

In this analysis, the application of a novel machine learning algorithm to the current classification model is explored. Specifically, this paper proposes a modified architecture consisting of adding 1D convolutional layers. The goal is to further improve the LLP jet classification model. An improvement to the model would increase the discovery potential of a Hidden Sector particles which could answer some of the Standard Model open questions.

Chapter 2

Theory

2.1 Long Lived Particles

Figure 2.1 displays the Feynman Diagram of the theorized HS model generating the signature of interest. Two protons (p) collide to form a heavy neutral boson (Φ). The boson then decays to two long-lived scalar particles (s) which in turn each decay to a fermion-antifermion pair (f, $\bar{f}$). Both the boson and the long-lived particles are invisible to the detector, indicated by the dotted lines in the diagram. The four fermion final state is the observable signature inside ATLAS. Due to the long lifetime of s considered in this model, each LLP decaying to a fermion-antifermion pair is postulated to decay to a displaced jet just before or in the first layers of the ATLAS calorimeters thus depositing most of their energy in the HCal. This restriction can be expressed by a high ratio of energy deposited in the HCal relative to the ECal (Equation 2.1). Other expected characteristics of the signal jet include a lack of tracks and narrow jet widths. A model exhibiting some of these discussed signal jet features is shown in Figure 2.2. It should be emphasized that this study is interested in searching for pairs of these types of jets.

$$CalRatio = \frac{E_{HCal}}{E_{ECal}} \tag{2.1}$$

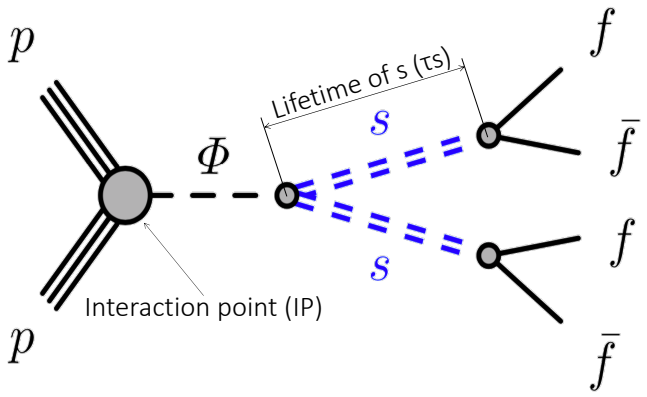

Figure 2.1: The Feynman diagram of the theoretical particle decays that create the signature of interest. Two protons (p) collide to form a heavy scalar boson (Φ). The boson decays to two neutral long-lived particles (s) which then both decay to a fermion-antifermion pair (f, $\bar{f}$). Overall this model serves as a benchmark for searching paired LLPs. Figure taken, with permission, from ATLAS analysis team.

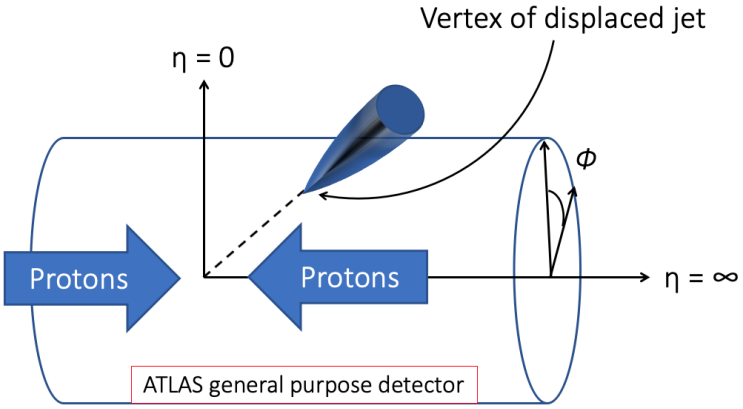

Figure 2.2: A diagram depicting a trackless-displaced jet. The dotted line again indicates no detection by the detector. Important high-level variables measured at ATLAS are also shown here. Pseudorapidity (η) is a spatial coordinate related to the angle of the particle in relation to the beam axis. ϕ is the angle of a particle in the transverse plane.

2.2 Backgrounds

Two types of jets which mimic signal are considered in this analysis.

2.2.1 Quantum Chromodynamics (QCD)

Although the least probable to resemble signal, QCD is the most abundant form of background. QCD multi-jets are simply decays to the SM from proton-proton collisions. A cluster of neutrons for example, decaying to other hadrons in the HCal could confuse the classification model with signal. Detector measurement errors could also contribute to a QCD jet reassembling signal.

2.2.2 Beam Induced Background (BIB)

BIB stems from muons generated from proton interactions with the collider beam gas or the collimators. This occurs prior to the protons reaching the ATLAS detector. These muons travelling parallel to the beam pipe could deposit energy in calorimeters creating a trackless jet.

2.3 Supervised Machine Learning

A multi-class classification model is needed to classify jets (this is a jet by jet not event level classification) as either signal, QCD, or BIB. The complexity of the classification problem at hand requires the unprecedented pattern-identification ability of novel machine learning algorithms. In the context of classification, supervised machine learning consists of systematically tuning weights of a function that describes the relationship between some input and discrete output by comparing the predicted outputs to ground truth. The variables that describe a particular input are referred to as features and the discrete classes the model is trying to predict are called labels. In the context of this analysis, the features are all the low-level and high-level variables that describe a particular jet e.g. track, constituent, muon segment, and p_T. The jet types (i.e. Signal, QCD, or BIB) are the labels.

$$L(y,\hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}(y_{ij} * log(\hat{y}_{ij})) \tag{2.2}$$

A supervised model tunes/learns its weights via a loss function which determines how close the predictions are to the truth values. The loss function for a typical multi-classification problem, and the one used in this study, is given by Equation 2.2. It is known as the categorical cross entropy loss function. $\hat{y}$ is the predicted value, y is the truth label, M is the number of classes, and N is the number of samples. Discrete outputs with n possible labels are converted to an array of length n. The value 1 is set to the index corresponding to the discrete class and 0s are set everywhere else, a technique called one-hot encoding. Given this description, it is simple to verify the presented loss function is a sum of separate loss for each class label per observation. Iterative optimization algorithms make it possible for the model loss to be minimized. The optimizer used in this analysis is Nadam [12]. The whole procedure of minimizing the loss function and tuning weights of the model is referred to as the training phase. It is crucial for the data used for model training to be independent from the data used for testing to avoid a biased model. A common metaphor used to illustrate this constraint is a student getting access to the answers of an exam prior to writing it versus the student getting access to similar previous exams. When a model has excessively tuned its weights to the point it has learned the complexities of the noise in the training data, the model is said to have overfitted. A standard technique to monitor model performance during training is to regularly evaluate the model on a separate dataset called the validation dataset. This provides insight into how well the model is doing and indicates the possibility of overfitting.

2.4 Deep Learning

Deep learning is an area of machine learning inspired by the complex neural networks of human brains. Networks are composed of interconnected layers of artificial neurons or nodes. In a simple feed-forward neural network, each node has an associated weight and bias, and its output is fed into an non-linear function called an activation function. The correct weights and biases of these nodes to match any given input to its corresponding output is determined during model training. Multiple layers of nodes and large numbers of nodes in each layer make it possible for the network to automatically learn highly complex and discriminative features.

This capability relates to the dominance of deep learning for complex classification tasks, in comparison to other machine learning algorithms.

The following two subsections provide technical details of the specific deep learning algorithms discussed throughout this paper.

2.4.1 Recurrent Neural Network (RNN)

Recurrent Neural Networks are a class of artificial neural networks with a powerful ability to model sequential data. The assumption made in standard neural networks is that the data is Independent and Identically Distributed (IID). In other words, no information/context is lost if samples are randomly selected from a dataset. In many problems however, this is not the case. For example, a model that tries to predict the next word in a sentence needs to have information about the parts of the sentence that came before it. RNNs solve this problem by not making the IID assumption and instead retain information on the inputs the model has seen so far by having loops in the network nodes. The output of the model is therefore dependent on both the current input and the inputs before it. As a result, these models perform exceptionally well in finding patterns in data containing variable length sequences.

2.4.2 1D Convolutional Neural Network (CNN)

Convolutional Neural Networks are a class of artificial neural networks centered around the idea of convolutions. Series of filters or feature detectors capture localized information as they move across an input such as pixels of an image. This operation is called a convolution. The filters themselves are nothing more than a matrix with adjustable weights that produce an output when multiplied by portions of an input matrix. The goal is for a network to tune these filters such that high-level features are extracted.

In the case of 1D CNNs, the width of a filter is the width of the input matrix. This implies the filter can only move across rows and not across columns (i.e. 1 dimension). The filter is also referred to as the kernel and the height of the filter is called the kernel size. The example of a 1D Convolutional layer shown in Figure 2.3 has a filter with kernel size 3.

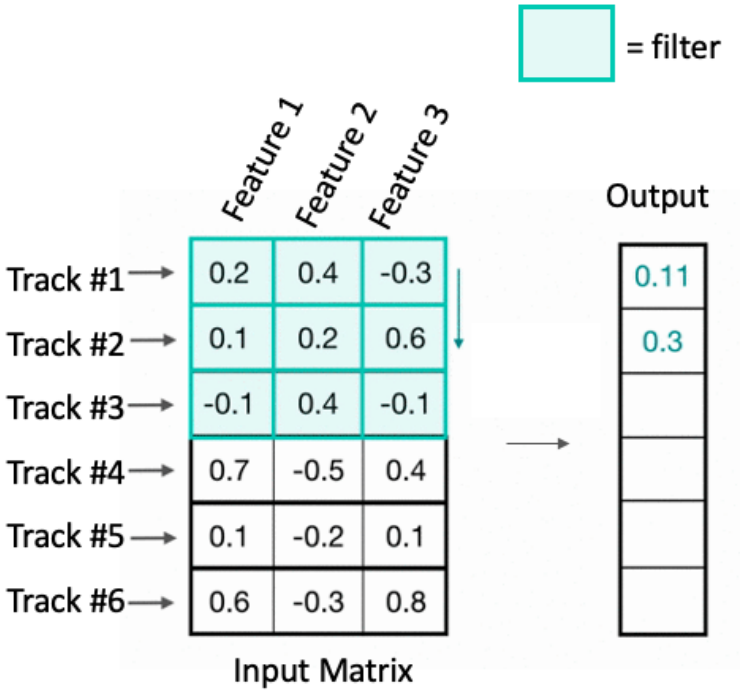

Figure 2.3: Diagram depicting a single 1D Convolutional layer with 1 filter of kernel size 3. At each step of the convolution, matrix multiplication is applied between the filter matrix and the portion of the input matrix covered by the filter. This operation is repeated as the filter moves down row by row along the input matrix. Original diagram taken from [3].

Chapter 3

Methods

The Python programming library Keras [13], with the TensorFlow [14] back-end, was used throughout this study to implement and modify networks.

3.1 Current Network Configuration

The current architecture of the LLP tagger developed by the UBC ATLAS group is a deep Recurrent Neural Network. It leverages Long Short-Term Memory (LSTM) Networks, specialized RNN layers capable of learning long-term dependencies, an ability standard RNN layers are known to lack [15]. These specialized layers solve the issue of being unable to learn the connections between relevant information and the current inputs when the gaps (i.e. how much data has been fed into the model since a particular set of inputs) between the two are too big. This makes LSTMs highly effective at learning dependencies across arbitrary sized sequential data.

Each jet fed into the network is truncated at 20 tracks, 30 constituents, and 30 muon segments. Features of these jet components consist of the various possible measurements made by the ATLAS detector such as transverse momentum, pseudorapidity, angle in the transverse plane, layer fraction, and timing information. Figure 3.1 provides a graphical representation of the current architecture.

In addition to a preconfigured network, preprocessed and transformed data was also available at the start of the project. The number of events and the distinction between simulated and real data is shown in Table 3.1. Details specifying the

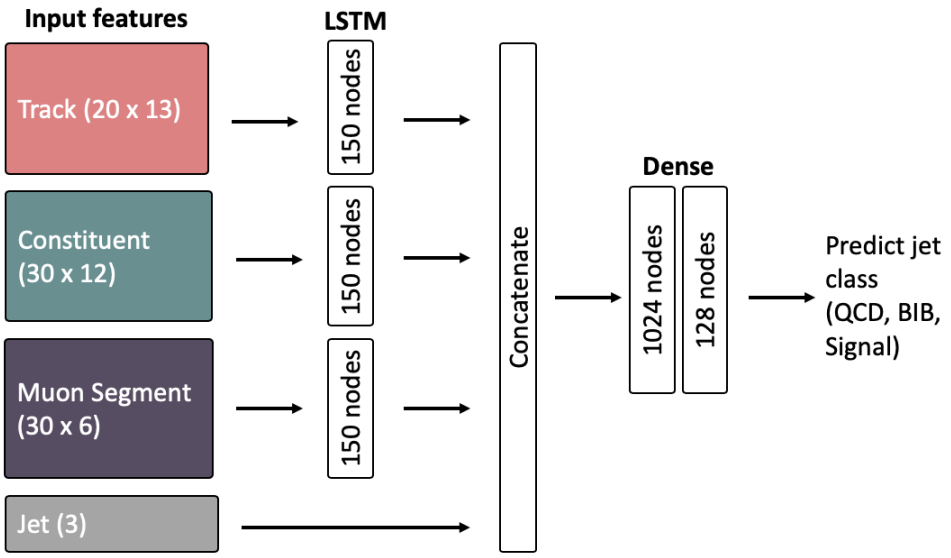

Figure 3.1: A simplified diagram depicting the deep learning based LLP tagger developed by the UBC ATLAS collaboration. The model leverages Long Short-Term Memory (LSTM) networks, specialized RNN layers capable of learning long-term dependencies. Consequently, the original network will be referred to as the LSTM model. The full architecture is displayed in Appendix A.1.

Event Type	Simulated/Real	Number of Events
Signal	Simulated	660,134
QCD	Simulated	766,056
BIB	Real	661,176

Table 3.1: Presented here are the number of events available per jet class, and whether the event is simulated or real. In total, the full dataset consists of 2,087,366 events.

methods used to generate simulated events is described in [9]. In essence, the pipeline presented above effectively served as the starting point of this study.

3.2 Exploring the Ordering of Transverse Momentum

LSTMs are particularly well-suited for temporal modeling, where inputs feeding into the network layers are expected to be ordered in some way. By default, tracks and constituents are sorted by descending p_T during the pre-processing phase, to take advantage of this fact. Although muon segments also feed into an LSTM, it is not possible to sort them by p_T since it is a missing feature. It is worth noting that this ordering is somewhat arbitrary and does not translate to any physical meaning. However, since some variables can be more accurately modeled than others, there is likely an optimal ordering to the inputs. Hence, the first study focused on determining whether transverse momentum is an appropriate ordering for inputs tracks and constituents. Models were trained with three different p_T-ordered datasets: descending, ascending, and random.

3.3 Modifying Model Architecture

The CMS collaboration published a paper [16] on training a deep neural network to classify b jets using proton-proton collision data measured with the CMS detector. The model architecture presented in their paper is a feedforward neural network consisting of CNN, LSTM, and Dense layers. Specifically, the CNN layers are 1D convolution filters with kernel size 1. Although several studies [17], [18] have shown this unified architecture is highly effective in applications that benefit from both temporal and spatial modeling, the former does not apply to the CNN layers in the b jet tagger. Instead, they perform global feature extraction and dimensionality reduction, without a spatial aspect since these filters capture a single row at a time. The addition of these 1D convolutional layers output highly discriminating and compressed features which feed into the LSTMs.

Inspired by this, in this second study, the addition of 1D convolutional layers with kernel size 1 to the current LLP tagger model is explored. Henceforth, the proposed model will be referred to as Conv1D + LSTM. The inputs track, constituent, and muon segment now feed into Conv1D layers before passing through the LSTMs. The number of Conv1D layers and filters were initialized to match the configuration outlined in [16]. An initial comparison was made to verify the addition of Conv1D layers does indeed improve the performance of the network.

Hyperparameter	Values	Count
Learning rate	$0.000025, 0.00005, 0.0001, 0.0002, 0.0004$	5
Regularization	$0.001, 0.0025, 0.005, 0.01$	4
Final Conv1D layer	$16, 12, 8$ for Constituent and Track $8, 6, 4$ for Muon Segment	3

Table 3.2: Hyperparameter search space for the grid search. The count column displays the number of different values tested per hyperparameter. In total $5x4x3 = 60$ unique model configurations were trained. Final Conv1D layer represents the number of respective filters for each of the 3 CNNs in the network and are referenced by their input data. Note, in attempt to reduce the search space (and to reduce computational load) this was treated as a single hyperparameter.

Following this step, the hyperparameters of the proposed architecture were optimized through a grid search, an effective yet computationally expensive model optimization technique. The search space consisted of 5 values for learning rate, 4 values for regularization, and 3 values for the number of filters in the final Conv1D layer. The specific values tested are shown in Table 3.2. Note, the learning rate and regularization values used for training the model in the previous study were 0.00005 and 0.001 respectively. The following are short descriptions for each hyperparameter part of the search space:

- **Learning Rate:** Size of the adjustments made to the model weights with respect to the loss gradient. Also known as step size.
- **Regularization:** An additional term to the loss function which penalizes model complexity to avoid overfitting.
- **Number of filters in final Conv1D layer:** Determines the width of the input matrices feeding into the LSTMs. This was part of the grid search to validate the usefulness of dimensionality reduction. For example, the Track input is a 20x13 matrix which reduces to 20x8 if the final Conv1D layer contains 8 filters.

3.4 Model Metrics

The next two subsections provide descriptions of two metrics this analysis used to evaluate model performance.

3.4.1 Accuracy

Recall, the final output of the network is a probability for each jet class. A given jet is considered accurately labeled if the ground-truth label matches the jet class with the highest probability. In order to measure a model's accuracy over a given dataset, model predictions for every jet are gathered. The accuracy is simply calculated as the number of correctly labelled jets divided by the total number of jets classified.

3.4.2 ROC AUC

A useful tool commonly used in binary classifier systems is a Receiver Operating Characteristic (ROC) curve. It provides graphical insight into the classification performance across all possible threshold values. The Area Under Curve (AUC) is defined as the total area under a ROC curve and represents the degree of separability. A value of 1 corresponds to a perfect classifier while 0.5 is equivalent to a randomly guessing network.

The following is a description of a specialized ROC curve tailored to the multi-class LLP tagger. This curve is generated by plotting QCD rejection against LLP efficiency. LLP tagging efficiency is defined as the fraction of times the network correctly tagged a jet as signal (Equation 3.1) while QCD rejection is equivalent to 1 over the False Positive Rate (FPR) (Equation 3.2). Table 3.3 defines the terms in these equations. The final jet class is integrated into the plot via the quoted BIB efficiency, a discrete value corresponding to the proportion of true BIB jets classified as BIB. This value determines the initial cut/separation to the three jet class distributions prior to generating the ROC curve. For this analysis, the BIB efficiency was set to 0.968 to be consistent with the BIB efficiency achieved in the previous analysis [9].

$$\textit{LLP Tagging Efficiency} = \frac{TP}{TP+FN} \tag{3.1}$$

	True Signal	True QCD background
Classified as Signal	True Positive (TP)	False Positive (FP)
Classified as QCD	False Negative (FN)	True Negative (TN)

Table 3.3: Table describing the terms in Equations 3.1 and 3.2.

$$QCD\ Rejection = \frac{FP+TN}{FP} \tag{3.2}$$

3.5 K-Fold Cross Validation

K-Fold Cross Validation (CV) is a powerful statistical technique used throughout this study to validate and compare models. It consists of randomly splitting the training data into k partitions and training a model for each possible training-validation pair. This produces k different models, such that each model is trained on $k-1$ partitions of the data, and each partition acts as the validation set for exactly one model. This technique results in a less biased, or more accurate estimate of the model skill than other methods.

Chapter 4

Results and Discussion

4.1 Re-ordering Transverse Momentum

The model trained with random p_T ordered data performed poorly in comparison to the models trained with ordered p_T data. Results of a 5-Fold cross validation are shown in Figure 4.1. Descending p_T seemed to be slightly better than ascending p_T and as a result the descending p_T dataset was used in the subsequent study.

From the results, transverse momentum appears to be a suitable ordering for the track and constituent inputs. It is worth observing that the mean performance was slightly higher in models trained with descending ordered data compared to ascending. Repeated experiments showed conflicting results and it is still unclear whether this small difference is significant. Regardless, the improved performance with sorted data compared to unsorted data is enough to justify either direction of ordering.

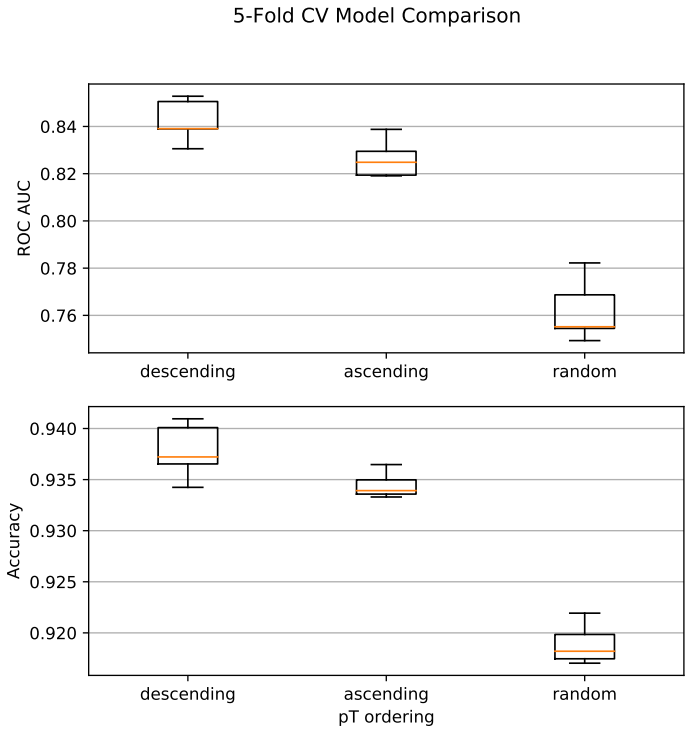

Figure 4.1: 5-Fold cross validation comparing the impact of p_T ordering on the model performance. The distribution of the model metrics are represented as boxplots. The mean is shown by the orange line, the top and bottom of the box represent the 75^{th} and 25^{th} percentile respectively, and the whiskers represent the maximum and minimum. Models trained with sorted p_T data performed better in both accuracy and ROC AUC.

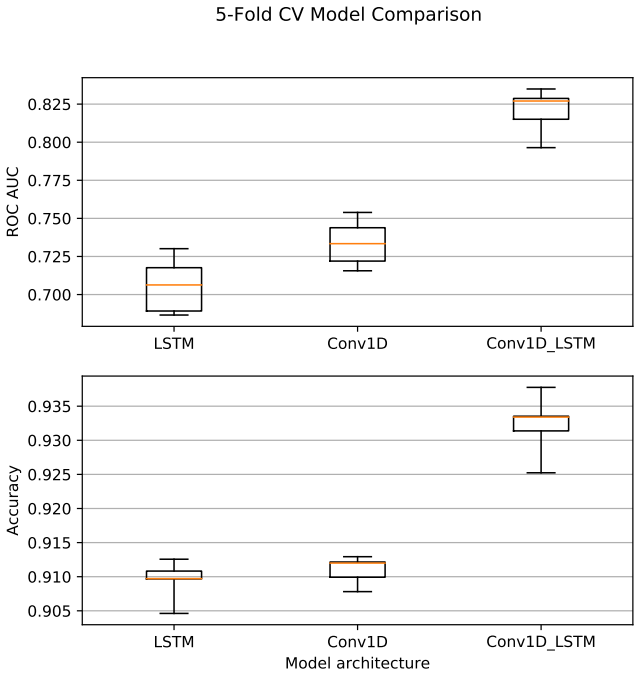

Figure 4.2: 5-Fold cross validation comparing model architectures. The combined architecture Conv1D + LSTM outperformed the models LSTM and Conv1D.

4.2 Modifying Model Architecture

A 5-Fold cross validation was performed to evaluate the new proposed architecture of adding 1D Convolutional layers. In addition to training the original and new model architectures, a third model consisting of Conv1D, GlobalAveragePooling1D, and Dense layers (referred to as the Conv1D model) was also trained. The Conv1D layer configuration including the number of filters for each layer, was set to closely match the configuration outlined in the DeepJet b tagging algorithm [16] due to the similar input and output shapes of the 1D CNNs. The Conv1D + LSTM architecture outperformed both models in accuracy and ROC AUC. It was also found that the Conv1D model (no LSTM layers) and the LSTM model achieved similar performance. Results of the 5-Fold cross validation are shown in Figure 4.2.

4.3 Optimizing Hyperparameters

Following the completion of the grid search, correlations were calculated between the hyperparameters from the search space and the model evaluation metrics. A strong positive correlation was found between learning rate and model performance. Their relationship is shown in Figure 4.3 which plots the mean ROC AUC.

Though far less significant, a negative correlation was found between regularization and model performance. Figure 4.4 plots the average ROC against regularization for each of the different learning rate values from the search space. A curve was plotted for each learning rate to better visualize the effect of changing the regularization. A subtle trend in better performance is seen when decreasing regularization. Finally, little dependency was found on varying the number of filters in the final Conv1D layers with the model metrics.

The final adjustment made to the Conv1D + LSTM architecture was decreasing the number of nodes in the LSTM layers resulting in a decrease in trainable parameters. The motivation for this adjustment comes from the general notion that the more parameters there are to train, the more likely it is for the model to overfit to the training data. This reduction from 150 to 60 nodes in the LSTM decreased the variability of the model performance, and increased the mean classification ability. Results from a 4-Fold CV comparing the effect of this adjustment are displayed in Figure 4.5.

To summarize, the best Conv1D + LSTM model was trained with a learning rate of 0.0004, regularization of 0.001, and 60 nodes for the LSTM layers. All the modifications made to the original LSTM network are reflected in the model architecture diagram shown in Figure 4.6. Table 4.1 provides a summary of the relative improvements achieved by the proposed model. There is a 10% ROC AUC improvement in comparison to the original model, clearly indicating the architecture Conv1D + LSTM is a major improvement to the deep learning LLP tagger.

4.3.1 Insights from Grid Search

Learning rate is arguably the most important hyperparameter to tune on a given network and its optimal value is highly dependent on the model architecture, optimizer, and loss function. Too small of a value can result in the loss converging to

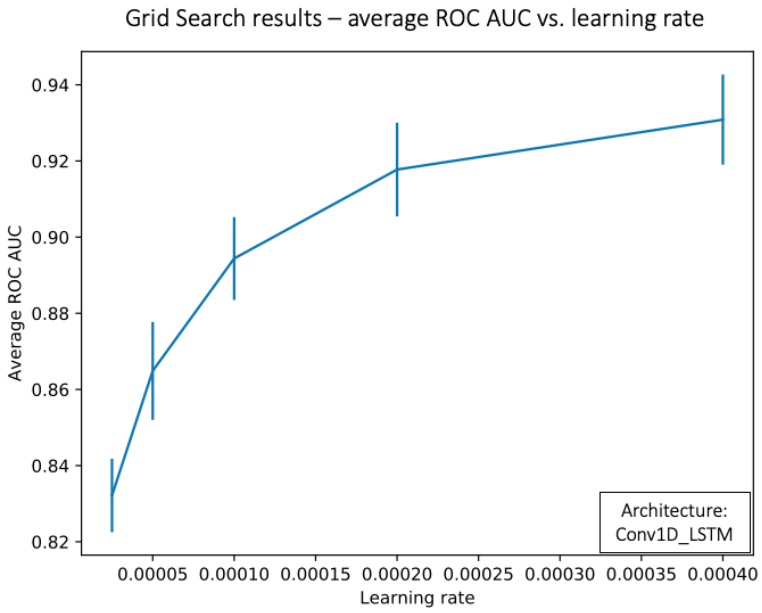

Figure 4.3: Grid search results showing the positive correlation between learning rate and model performance. 12 different models were trained for each learning rate value. Prior to the grid search, the learning rate was set to 0.00005 which achieved the second lowest mean performance in this experiment.

Model	ROC AUC	Accuracy
Conv1D + LSTM optimized	0.96	0.97
Conv1D + LSTM	0.92	0.96
LSTM	0.87	0.94

Table 4.1: Table to highlight the improvements achieved by this study. Results indicate the architecture Conv1D + LSTM prior to the grid search provides a 5-6% increase in ROC AUC in comparison to the LSTM model. The proposed model with optimized hyperparameters attains a relative improvement of 10%. Note, the number of epochs were doubled to 200 to ensure all models have converged in order to record maximum model performance.

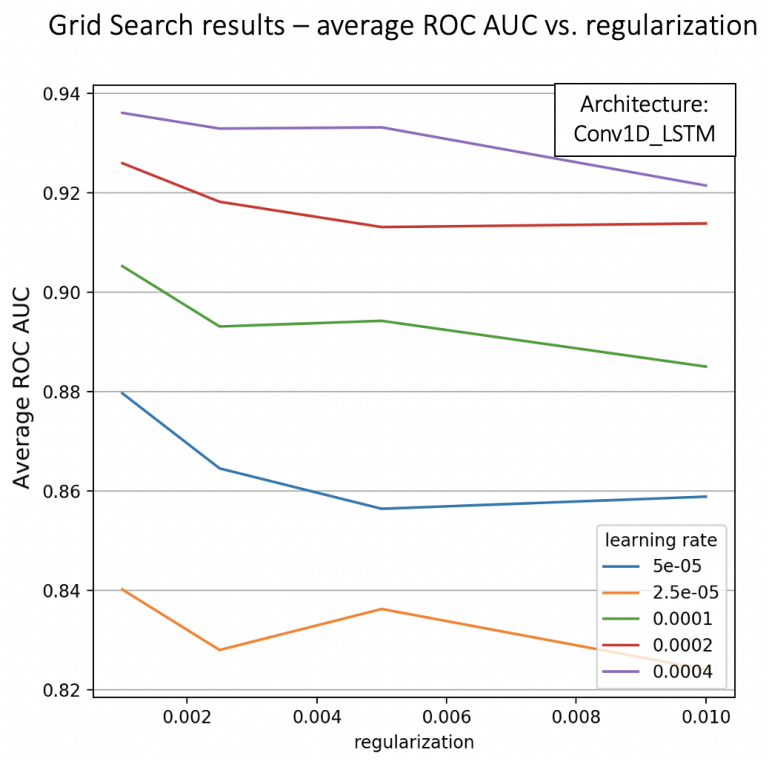

Figure 4.4: Average ROC AUC scores plotted against the tested regularization values from the grid search. A curve is displayed for each of the learning rates to highlight the effect of varying the regularization. 3 different models were trained for each of the regularization + learning rate configurations.

local minimum, unable to climb out due to the small step size. As such, dedicated time was spent tuning this value for the Conv1D + LSTM model. Results from the grid search strongly indicated the optimal learning rate for the proposed architecture was much larger than the starting value based on the training configuration from the previous study.

Although regularization was found to have a slight dependency on the model metrics, additional experiments would be required to validate this trend (via a larger search space) and to determine whether 0.001 is truly the optimal value. A final improvement to the hyperparameters would be finding the optimal number of LSTM

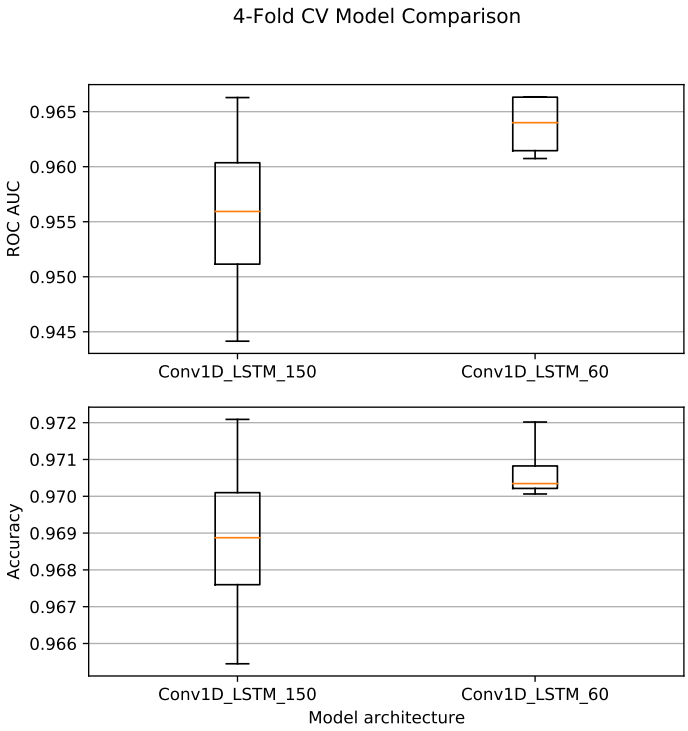

Figure 4.5: Impact on the Conv1D + LSTM architecture when decreasing the number of nodes in the LSTM layers from 150 nodes to 60 nodes. A 4-Fold CV was performed to evaluate this modification on the model performance.

nodes, since only one other value was tested other than the initial setting. Along with the other improvements discussed above, further hyperparameter optimizations could be performed with a randomized search resulting in drastically lower runtime.

Another major result from the parameter search is the fact that adding additional filters to the final Conv1D layer did not seem to improve the model. A layer containing 8 filters seemed to output the same amount of meaningful information as a layer with 16 filters. Therefore, the null correlation found between the tested number of filters and model performance validates the dimensionality reduction ability of the Conv1D layers. To conclude this part of the discussion, the

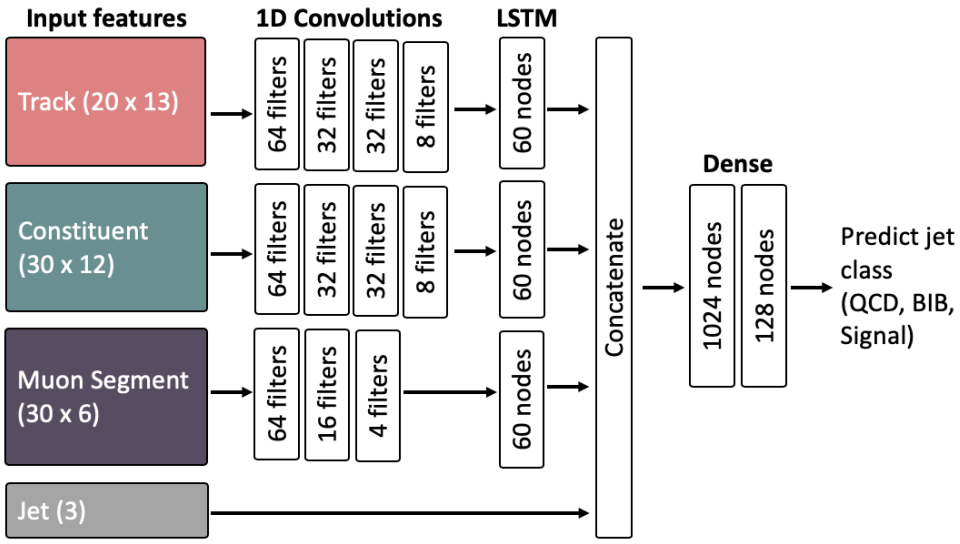

Figure 4.6: Diagram of the optimized proposed model architecture.

improvements to the LSTM model with the added sequential Conv1D layers can be attributed to the resulting feature extraction and compression.

4.3.2 Possible New Metrics

Aside from ways to further improve the model, extensions to this analysis should consider exploring new metrics that could better quantify model performance, or provide more insight when making comparisons. One example would be extracting/optimizing for the signal efficiency for a small fixed set of QCD rejection values, rather than calculating the efficiency over all values of rejection (as done for the ROC curve and accuracy number). Setting fixed thresholds would output more explicit measures of the network's ability to differentiate the three jet classes. The added constraint would also make it easier to explicitly optimize for signal efficiency.

Another useful metric would be to calculate the ratio of signal event count to the square root of the background event count. This metric, often referred to as "significance", is commonly used in particle physics research. Optimizing this ratio implies maximizing the signal event count while minimizing the uncertainty

expressed by the denominator. The counting of events and the uncertainty are associated with the fact that counting signal events with some probability of background obeys a Poisson distribution. The higher the significance, the higher the confidence is to reject the null hypothesis that signal is purely a result of statistical fluctuation of the background. In other words, an increase in significance would directly translate to a higher probability of finding new physics. Though a promising approach, this technique is nontrivial since it requires information on the expected number of signal and background events (i.e. the relative cross-sections).

Chapter 5

Conclusions

Many extensions to the Standard Model suggest the existence of long lived particles that only interact with the SM through a weakly-coupled mediator. The extended lifetime of these particles and the weak interaction with the SM would result in displaced hadronic jets in the ATLAS detector. Ongoing and published research by the ATLAS collaboration, have presented machine learning models to classify displaced jets in the ATLAS calorimeters. In this analysis, a modified architecture was proposed with the aim to improve the model performance.

Prior to exploring new models, an experiment was performed to verify the decision to sort certain input data by transverse momentum due to the nature of LSTM layers expecting ordered inputs. The experiment consisted of training models on a descending, ascending and random p_T ordered datasets. Models trained on the ordered datasets performed significantly better in both accuracy and ROC AUC.

A deep recurrent neural network developed as part of the ongoing LLP search, was established as the benchmark for comparing model performance and developing an improved architecture. This analysis explored the addition of 1D Convolutional layers to the deep learning based LLP tagger. A 5-Fold cross validation showed the proposed Conv1D + LSTM model substantially outperformed the original network.

Finally, a grid search was performed to optimize the hyperparameters of the improved model. A larger learning rate, a slightly smaller regularization value, and a decrease in LSTM nodes were found to further enhance the model performance.

Overall, the optimized Conv1D + LSTM model achieved a 10% increase in ROC AUC in comparison to the original model.

Extensions to this analysis should consider exploring other metrics for evaluating and comparing models. Significance and signal efficiency at specific QCD rejection values are two proposed metrics which could offer better insight to the discovery potential.

Bibliography

[1] Wikipedia contributors. Standard model — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Standard_Model&oldid=943664780, 2020. [Online; accessed 6-March-2020]. → pages vi, 2

[2] Lawrence Lee, Christian Ohm, Abner Soffer, and Tien-Tien Yu. Collider searches for long-lived particles beyond the standard model. *Progress in Particle and Nuclear Physics*, 106:210–255, May 2019. → pages vi, 4

[3] Cezanne Camacho. Cnns for text classification - gif. https://cezannec.github.io/assets/cnn_text/conv_maxpooling_steps.gif. [Online; accessed 1-April-2020]. → pages vii, 11

[4] Yanou Cui and Brian Shuve. Probing baryogenesis with displaced vertices at the lhc. *Journal of High Energy Physics*, 2015(2), Feb 2015. → pages 1

[5] Keith R. Dienes, Shufang Su, and Brooks Thomas. Distinguishing dynamical dark matter at the lhc. *Physical Review D*, 86(5), Sep 2012. → pages 1

[6] Yuk Fung Chan, Matthew Low, David E. Morrissey, and Andrew P. Spray. Lhc signatures of a minimal supersymmetric hidden valley. *Journal of High Energy Physics*, 2012(5), May 2012. → pages 1

[7] Matthew Baumgart, Clifford Cheung, Joshua T Ruderman, Lian-Tao Wang, and Itay Yavin. Non-abelian dark sectors and their collider signatures. *Journal of High Energy Physics*, 2009(04):014–014, apr 2009. → pages 1

[8] The ATLAS Collaboration. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, 716(1):1–29, Sep 2012. → pages 2

[9] The ATLAS Collaboration. Search for long-lived neutral particles in pp collisions at $\sqrt{s} = 13\ tev$ s = 13 tev that decay into displaced hadronic jets in the atlas calorimeter. *The European Physical Journal C*, 79(6), Jun 2019. → pages 5, 13, 16

[10] Shannon Egan, Wojciech Fedorko, Alison Lister, Jannicke Pearkes, and Colin Gay. Long short-term memory (lstm) networks with jet constituents for boosted top tagging at the lhc, 2017. → pages 5

[11] Wahid Bhimji, Sasha Farrell, Thorsten Kurth, Michela Paganini, Mr Prabhat, and Evan Racah. Deep neural networks for physics analysis on low-level whole-detector data at the lhc. *Journal of Physics: Conference Series*, 1085, 11 2017. → pages 5

[12] Timothy Dozat. Incorporating nesterov momentum into adam. 2015. → pages 9

[13] François Chollet et al. Keras. https://github.com/fchollet/keras, 2015. → pages 12

[14] M. Abadi, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. → pages 12

[15] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994. → pages 12

[16] The CMS Collaboration. Performance of the DeepJet b tagging algorithm using 41.9/fb of data from proton-proton collisions at 13TeV with Phase 1 CMS detector. Nov 2018. → pages 14, 20

[17] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description, 2014. → pages 14

[18] Tara Sainath, Oriol Vinyals, Andrew Senior, and Hasim Sak. Convolutional, long short-term memory, fully connected deep neural networks. pages 4580–4584, 04 2015. → pages 14

Appendix A

Supporting Materials

A.1 Python Code

The GitHub repository **rdesc/deep-learning-llp-tagger** contains the python scripts that were used for pre-processing data, generating plots, and for training, evaluating, and testing models.

A.2 Complete Model Diagrams

The following three figures consist of complete diagrams of the three different architectures discussed in this paper. These diagrams were generated via the **plot_model** method from **keras.utils**.

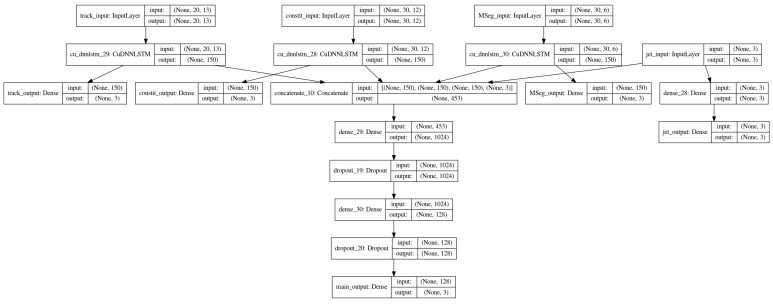

Figure A.1: Complete diagram of the LSTM model

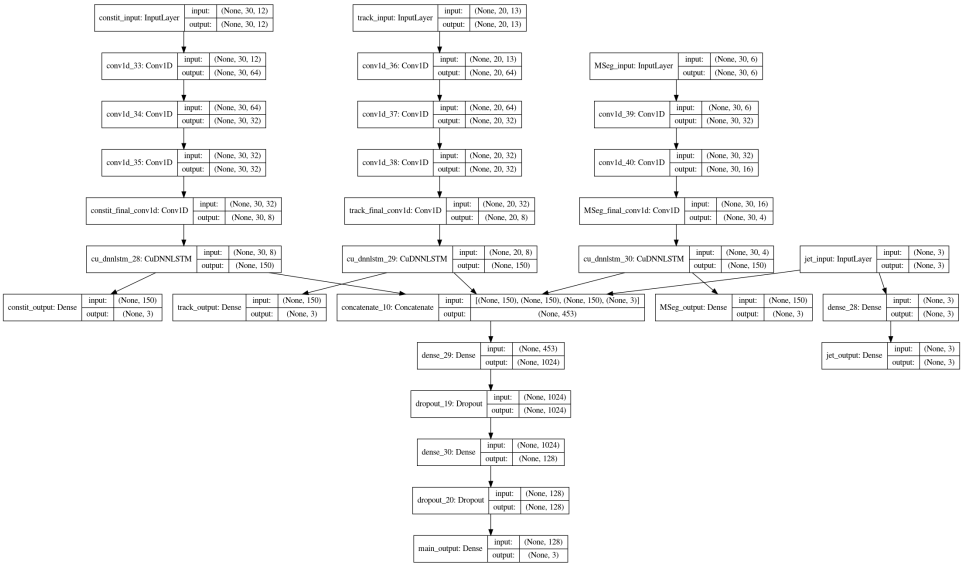

Figure A.2: Complete diagram of the Conv1D + LSTM model.

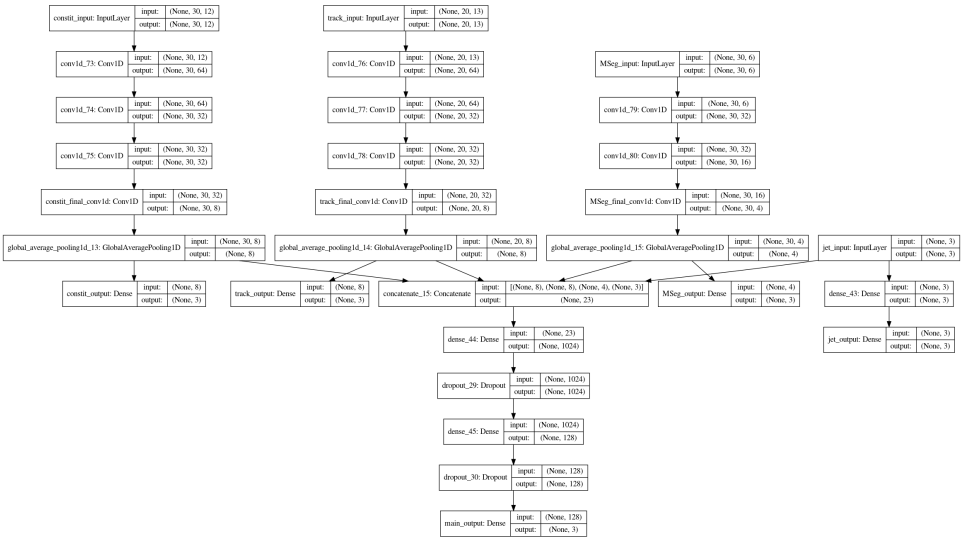

Figure A.3: Complete diagram of the Conv1D model. **GlobalAveragePooling1D** layers were used to flatten the features.