Research Report

Lawrence Livermore National Laboratory

Improving High-Energy Particle Detectors with Machine Learning

LLNL-TR-814867[1]

Michael Heinz, Aaron Angerami, Piyush Karande, and Ron Soltz

Abstract

Microseconds after the Big Bang, the universe existed in a state called the quark-gluon plasma (QGP). To experimentally study its properties, the QGP is recreated in high-energy nuclear collisions at the LHC, and the particles produced from the QGP are reconstructed from their energy deposition in the ATLAS calorimeter. This requires both classifying the particles and calibrating their deposited energy. The objective of this project is to improve the reconstruction by using machine learning techniques, where the energy depositions of clusters of cells, formed by ATLAS topo-clustering methods, are treated as three-dimensional images when inputted to neural networks. This approach significantly improves the calibration of deposited energies when cross-validating while training, and models trained on idealized data predict the calibrated energies of particles in more complex data sets well. Additionally, implementation of a data generator using `uproot` allows the program to load input data into memory as needed while training or predicting, significantly reducing the amount of memory used. The data generator also allows for use of multiprocessing to speed up training and evaluating. This work illustrates that using machine learning methods for both classification and calibration has the potential to significantly improve particle reconstruction.

1 Introduction

One of the main goals of high-energy nuclear physics is to understand properties of the quark-gluon plasma (QGP), the state of matter in which the universe existed microseconds after the Big Bang. To experimentally study its properties, the QGP is recreated in high-energy nuclear collisions at the Large Hadron Collider (LHC) at CERN. In these collisions, the QGP produces thousands of particles, whose energy deposition is recorded in the ATLAS calorimeter system. The ATLAS calorimeter system is discretized, with six layers of cells which record energy deposition using different methods [1]. The first three layers of cells comprise the LAr Electromagnetic Barrel (EMB), which mostly reads the energy deposition from charged particles produced by electromagnetic showers, while last three layers of cells make up the tile calorimeter, where particles produced by hadronic showers deposit most of their energy. The energy deposition in the calorimeter is used to try to reconstruct the original particles produced in the collision.

The first step in this particle reconstruction is clustering the cells in the calorimeter together using the ATLAS topo-clustering method, where each cluster is supposed to represent an original particle [1]. However, due to electronic noise and other sources of fluctuations in energy such as

[1]Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344.

energy deposited from previous events or other collisions in the same event, the clustering strategy uses energy cutoffs in order to extract the significant signal from that noise background. As a result, some of the energy deposited by the particle can be lost in cells that are not clustered due to the clustering strategy. Additionally, energy can be lost in inactive material near the cluster and, for hadronic showers, there is a loss of signal in the deposited energy due to the character of the ATLAS calorimeter. Thus, the energy of the topo-cluster must be calibrated to represent the true energy of the particle which deposited the energy in the calorimeter. Also, since the calibration depends on if the shower is electromagnetic or hadronic, it is essential to classify the cluster correctly [2]. In this project, the goal is to reconstruct the particles produced from showers of both charged ($\pi^{\pm}$) and neutral (π^0) pions, which are the most common particles produced by high-energy collisions. The charged pions produce electromagnetic showers, while the neutral pions produce hadronic showers.

The classification method and "local cell weighting" (LCW) energy calibration method currently used in ATLAS use traditional binning techniques based on a small number of hand-picked calorimeter observables [2]. The objective of this project is to improve the reconstruction of particles by using machine learning techniques, where the energy depositions of clusters of cells are treated as three-dimensional images when fed into neural networks.

This report is organized as follows. The reconstruction of particles using machine learning and how it compares to previous reconstruction techniques is described in Section 2. Section 3 discusses the implementation of a data generator which uses `uproot` and its benefits.

2 Reconstruction of Particles Using Machine Learning

2.1 Background

The first step to reconstructing particles is to form topo-clusters from cells in the calorimeter [1]. The topo-clustering algorithm starts proto-clusters with cells which have significant signals, called "seed cells", whose signal exceeds a seeding threshold S. After this, if a neighboring cell has a signal larger than the growth threshold N, the cell is included in the proto-cluster and its neighboring cells are also considered. Here, a neighboring cell means either an adjacent cell in the same layer or a cell in a different layer which overlaps with the current cell. As the proto-clusters grow, two proto-clusters are combined if they both include a cell whose signal is larger than N. Once the proto-cluster has no more neighboring cells whose signal is larger than N, the proto-cluster stops growing. However, neighboring cells who pass a principal threshold P which is smaller than N are still included in the cluster. If these proto-clusters have two or more local maxima, they are split in the three spatial directions, and cells which are shared between two clusters have their signal split between them. After the splitting is complete, the algorithm is finished, and the remaining clusters are called the topo-clusters.

Once the topo-clusters are formed, the total cluster energy is calculated, and their kinematic properties coordinates are calculated be weighted averages based on the cells energy, weighting cells with a higher signal more [1]. In the classification and energy calibration methods used so far, the clusters are then binned based on a hand-picked subset of these variables, which is different for classification and energy calibration. The classification and energy calibration for each bin is determined from simulated training data, where the true energy, deposited energy, and classification of each particle are known. The objective of this project is to improve the classification and energy calibration of these clusters by training neural networks on the training data using image recognition

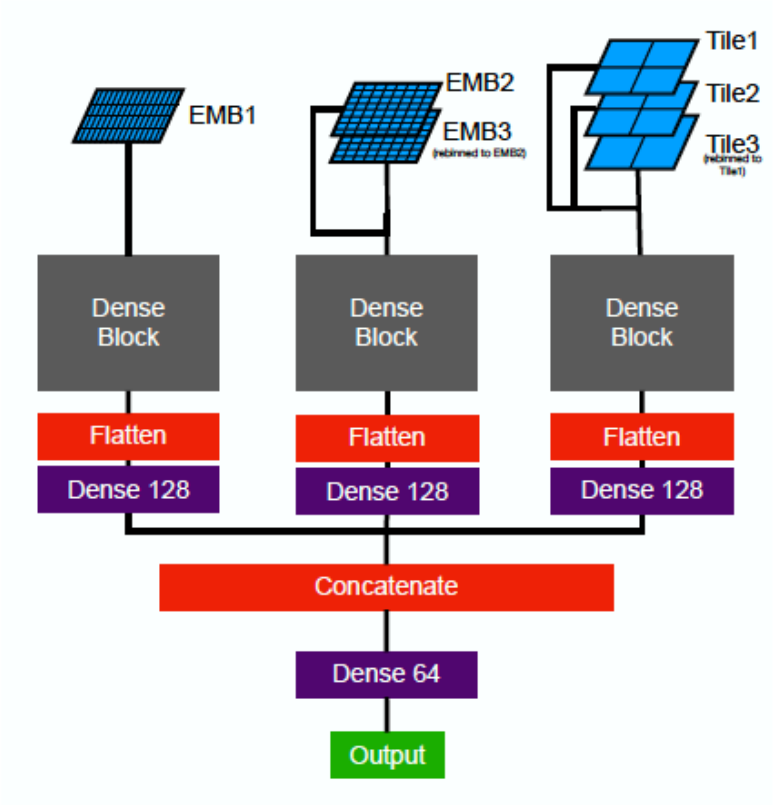

Figure 1: A sketch of the DenseNet model used for both classification and energy calibration by regression [2].

techniques on the energy depositions which are treated as three-dimensional images.

2.2 ATLAS Calorimeter Images

For this study, only the central portion of the detector is considered [2]. In this portion of the detector, the first sampling layer of the EMB calorimeter (EMB1) has a 128-by-4 grid of cells. On the other hand, the second sampling layer (EMB2) is a 16-by-16 grid, and the third sampling layer is a 8-by-16 grid. Finally, the three tile layers, Tile0, Tile1, and Tile2, are 4-by-4, 4-by-4, and 2-by-4 grids, respectively, in this portion of the detector.

2.3 Model

The neural network model used for both classification and energy calibration in this project is a DenseNet, as seen in Figure 1. The Dense Block for the DenseNet model is described in Ref. [3]. For both tasks, the energy depositions of the six calorimeter layers are fed into the neural network as three inputs: a one-channel image of the EMB1 layer at full resolution, a two-channel image of the EMB2 and EMB3 layers, where the EMB3 layer is upscaled to the EMB2 resolution, and a three-channel of all the tile layers, where the Tile2 layer is upscaled to the resolution of the Tile0 and Tile1 layers.

The output and loss function of the neural network differ for classification and energy calibration by regression. For classification, the model is trained using categorical crossentropy loss and outputs an array of two elements which describe the probability that the cluster comes from a charged pion or a neutral pion. The cluster is identified as coming from a neutral pion shower if its probability to come from one is larger than the classification cutoff p, where the default value for p is 0.5. For energy calibration by regression, the model is trained using mean-squared error and outputs a single value which represents the calibrated energy of the cluster, usually scaled for purposes of normalization.

2.4 Results

2.4.1 Classifying

Training of the classifier was done using cross-fold validation, which is when the trained network is tested on data it has not seen after each epoch to prevent overfitting. When using a classification cutoff $p = 0.5$, the classifier model has 94.57% accuracy when predicting the type of pion for the validation data. However, it only correctly classified 88.47% of the clusters from π^0 showers, while it correctly classified 96.25% of the clusters from charged pion showers. This is most likely because most of the training data came from charged pion showers, so the model was penalized more if it was incorrectly classifying charged pions.

On the other hand, the classifier model has an accuracy of 94.06% on the validation data when the classification cutoff p is taken to be 0.3. Although the overall accuracy is lower, the model now correctly classifies 93.70% of clusters from π^0 showers and 94.17% of clusters from $\pi^{\pm}$ showers. Going forward, investigation into which value of the classification cutoff p is best will be valuable.

2.4.2 Regression for Energy Calibration

Training of the regression model for energy calibration was done separately for neutral pions and charge pions as their energy is deposited in different ways in the calorimeter. The results of the cross-fold validation while training on the original training sample as well as inference on different collision samples compared to previous methods for energy calibration are shown in Figure 2. The cross-validation for the training sample (solid red) shows significant improvement in calculating the calibrated energy compared to the original cluster energy (dashed red) and the LCW calibrated energy (dashed-dotted red). Notice that the median original cluster energy is significantly smaller than the true energy for clusters with higher true energy, and this is properly corrected for by the LCW method. However, the LCW method still significantly overpredicts the true energy for clusters whose energy is less than 1 GeV. On the other hand, the DenseNet regression model predicts the true energy extremely well for clusters with true energy even as low as 0.3 GeV. The overprediction of the calibrated energy for clusters with low energies is likely due to the large influence of noise in cluster formation and on cluster energy at these low energies.

The DenseNet regression model trained on the original training sample was also used to predict the true energies of clusters for a different single pion sample, which is affected differently by noise compared to the original training sample due to a different noise cutoff. As seen in Figure 2, the DenseNet regression model (solid blue) fails to predict the true energy as well as the LCW calibration model (dashed-dotted blue). Still, the DenseNet regression model still predicts the true energy better than the original cluster energy (dashed blue) at high energies.

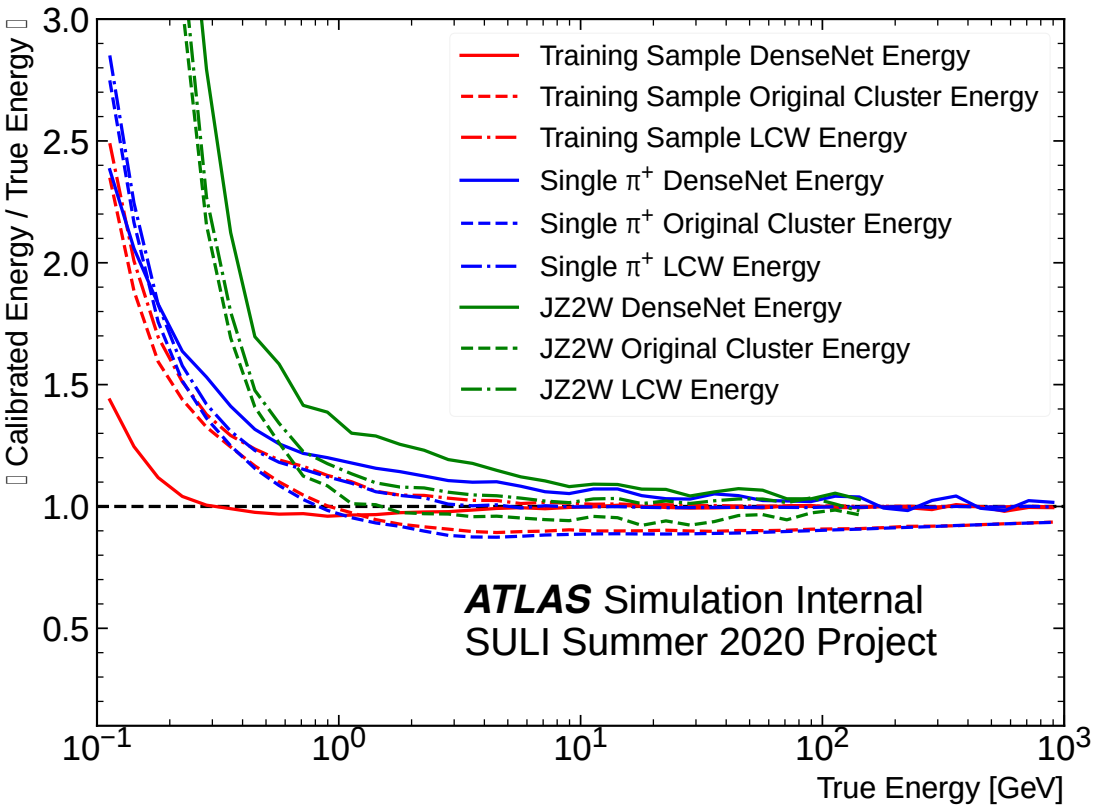

Figure 2: The ratio between the calibrated and true energy as a function of true energy, comparing inference with the model trained on the original idealized training sample versus previous methods of energy calibration. Results are displayed for the original training sample (red), a different single pion sample (blue), and a jet sample (green).

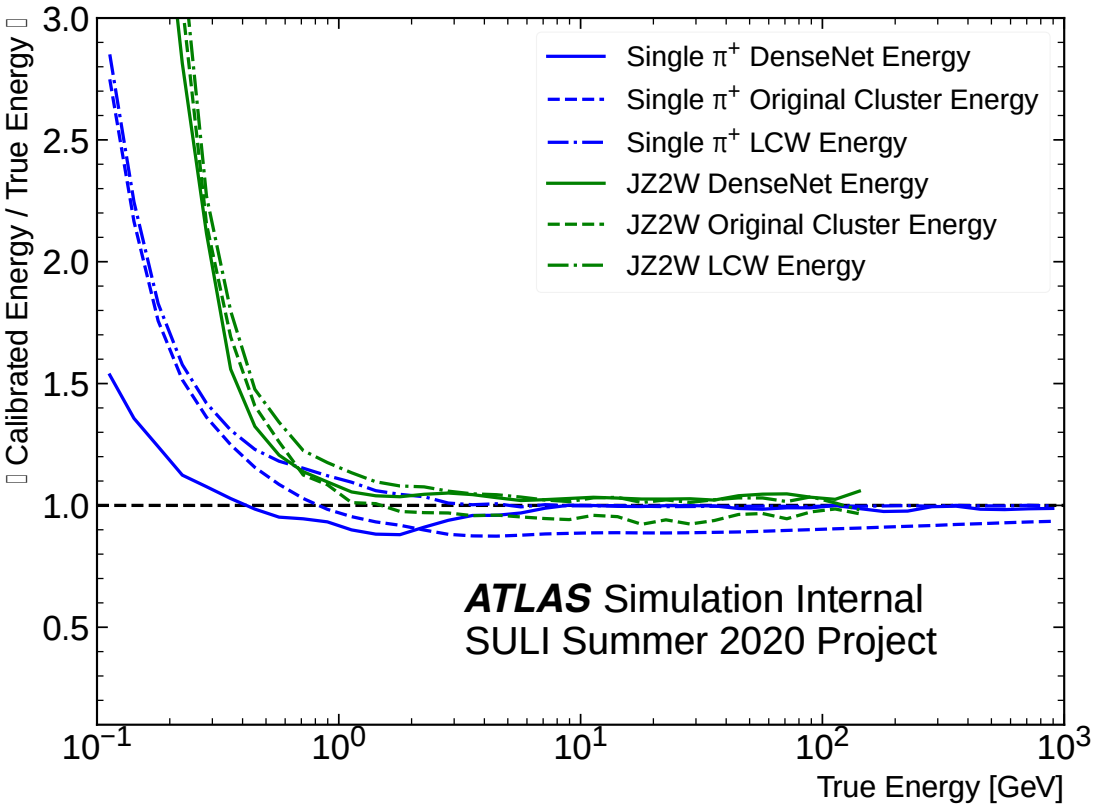

Figure 3: The ratio between the calibrated and true energy as a function of true energy, comparing inference with the model trained on the single pion sample versus previous methods of energy calibration. Results are displayed for a single pion sample (blue) and a jet sample (green).

Finally, the classifier and regression model were used in succession to predict the true energy of clusters created from a jet sample, JZ2W, which includes both neutral pion (electromagnetic) showers and charged pion (hadronic) showers. For each cluster, the classifier was used, with classification cutoff $p = 0.5$, to determine if the cluster cam from a charged or neutral pion. Depending on the classification, the corresponding regression model was used to predict the cluster's true energy. Again, the DenseNet regression model (solid green) predicts the cluster energy relatively well at high cluster energies, but fails to predict the energy as well as the LCW model (dashed-dotted green).

Although the DenseNet model fails to predict the true cluster energy as well as the LCW model for the latter two samples, the results are still promising: the DenseNet model still predicts the cluster energy relatively well even though the latter two samples are affected differently by noise compared to the training sample and the jet sample includes clusters produced by both neutral and charged pions.

2.4.3 Training on a Different Sample

Due to the issues in predicting the cluster energies for the single pion sample with the DenseNet model trained on the training data, another DenseNet model was trained on the single pion data itself to see if there were differences in behavior. The results of cross-validation and prediction with this model are shown in Figure 3. As seen from the cross-validation (solid blue), the DenseNet model is much better at predicting the cluster energies for the single pion data when being trained on a portion of the same data rather than the training data. It is also still an improvement to the LCW model at low energies. However, it is valuable to note that the cross-validation curve for the single pion sample in Figure 3 is not as good as the one for the training sample in Figure 2, indicating that there is perhaps some trouble in modeling the energy calibration for the single pion sample.

Once again, the classifier, which was still trained on the original training sample, and the regression were applied in succession to calibrate the cluster energies for the JZ2W jet sample. Here, the DenseNet model (solid green) actually improves on the LCW model (dashed-dotted green), as seen in Figure 3. Since the only change in this calibration was the data on which the regression model was trained, it is likely that there is an intrinsic difference between the original training sample and the latter two samples outside of the noise cutoff mentioned previously. Still, the results of the DenseNet predictions for the jet sample are incredibly promising, showing that machine learning techniques could lead to large improvements in particle reconstruction in the future.

2.5 Future Work

The use of machine learning methods for classification and energy calibration shows great potential. In the future, it will be valuable explore the classification cutoff p and its effect on energy calibration for jet samples with multiple types of particles. Additionally, the difference between the training sample and single pion sample should be examined to try to explain the difficulty that the DenseNet model trained on the training sample had in predicting the cluster energies for the single pion sample.

3 Implementing a Data Generator

So far, the training or testing data was fully loaded into memory as NumPy arrays to be passed into the TensorFlow `fit` and `evaluate` functions. When the number of clusters in the sample becomes large, this clearly takes up a lot of memory, so the problem cannot be scaled as easily. To solve this issue, a memory-efficient data generator using `uproot` was implemented. `uproot` is a reader and a writer of the ROOT file format, which is the format of choice for high-energy nuclear collision simulations data. `uproot` uses NumPy to cast blocks of data from ROOT files to NumPy arrays, and its `iterate` method allows the user to read specific entries from the ROOT files as needed. The data generator utilizes `uproot` by declaring a batch size and then iterating over the data in that batch size, using `iterate` to read in the specific block of data each iteration.

Since the data is loaded each iteration, the data generator was initially much slower per epoch than simply loading in all the data into NumPy arrays. However, the data generator allows use of multiprocessing, which speeds up the training and evaluating by allowing processes to be done in parallel. When using eight multiprocessing workers and a batch size of 1028, evaluation using the data generator was almost as fast as using full NumPy arrays.

3.1 Future Work

Moving forward, it will be valuable to implement the ability to shuffle the input data in the data generator, as shuffling data after each epoch is a common technique used when training to allow the model to see the data in various orders over time.

References

[1] The ATLAS Collaboration. Topological cell clustering in the ATL calorimeters and its performance in LHC Run 1. *Eur. Phys. J. C*, 77(490), 2017.

[2] The ATLAS Collaboration. Deep Learning for Pion Identification a Energy Calibration with the ATLAS Detector. Technical Report ATL-PHYS-PUB-2020-018, Jul 2020.

[3] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.