UNIVERSITÉ
DE GENÈVE

FACULTÉ DES SCIENCES
Section de physique

Physics applications of AI exam

Jet identification in ATLAS

Dimitri Moulin

16 juin 2022

Summary

1 Introduction

The search for new physics, new particles and interactions is highly dependent on our capacity to infer the nature of the interaction and the particles created during the collision recorded by the detector. One key element for collider physics is *jet identification*, a jet can be pictured as a collection of particles detected next to each other. Jets can originate from several sources such as a quark or a gluon (QCD), a boson (W/Z) which then decays to two quarks or from a top quark decaying to a b-quark and a W boson.
The application of machine learning (ML) techniques and deep learning (DL) to high energy physics (HEP) and particularly to particle identification is more and more common. The capacity of deep learning algorithms to extract relevant features from high dimensional data allows to exploit most of the information contained in the data obtained by the detector. The use of physics-inspired models [3], [4] shows how physical understanding can be used to improve algorithms. The use of higher level variables, such as the four-momentum of a jet can also help the network in its identification task.

In this report, we perform jet classification on jets recorded by the ATLAS detector using both low-level variables (p_T, η, ϕ) and higher level variables such as energy correlation factors using a deep neural network with fully connected layers. Another analysis is also performed using image representations of jets built from the detector response using a convolutional neural network (CNN). Finally an attempt is made to combine these two approaches using the CNN predictions as an aditionnal input to the fully connected model. Both approaches are done using binary classification (QCD jets vs WZ) and multi classification with the main goal being QCD jet identification. A comparison is then made to determine which type of classification performs best.

This report is organised as follows. Section 2 describes the datasets used to train and evaluate the algorithms as well as the input variables and the data-processing. Section 3 introduces the algorithms based on a fully-connected architecture and a CNN. Finally, Section 4 summarizes the results and compares the different approaches.

2 Data

The dataset consists of several jet variables for each type of jet : QCD, W/Z and TOP. The low-level reconstructed variables include jet transverse momentum p_T, as well as the detector's angles ϕ, η and the reconstructed mass m. Other higher level variables are included such as jet N-subjettiness $\tau_{1,2,3}$, energy correlation factors ECF2/3 and splitting scales d_{12}, d_{23}. All of these variables a thoroughly defined in [1]. Figure 1 shows the distribution of the variables for each class of jets.

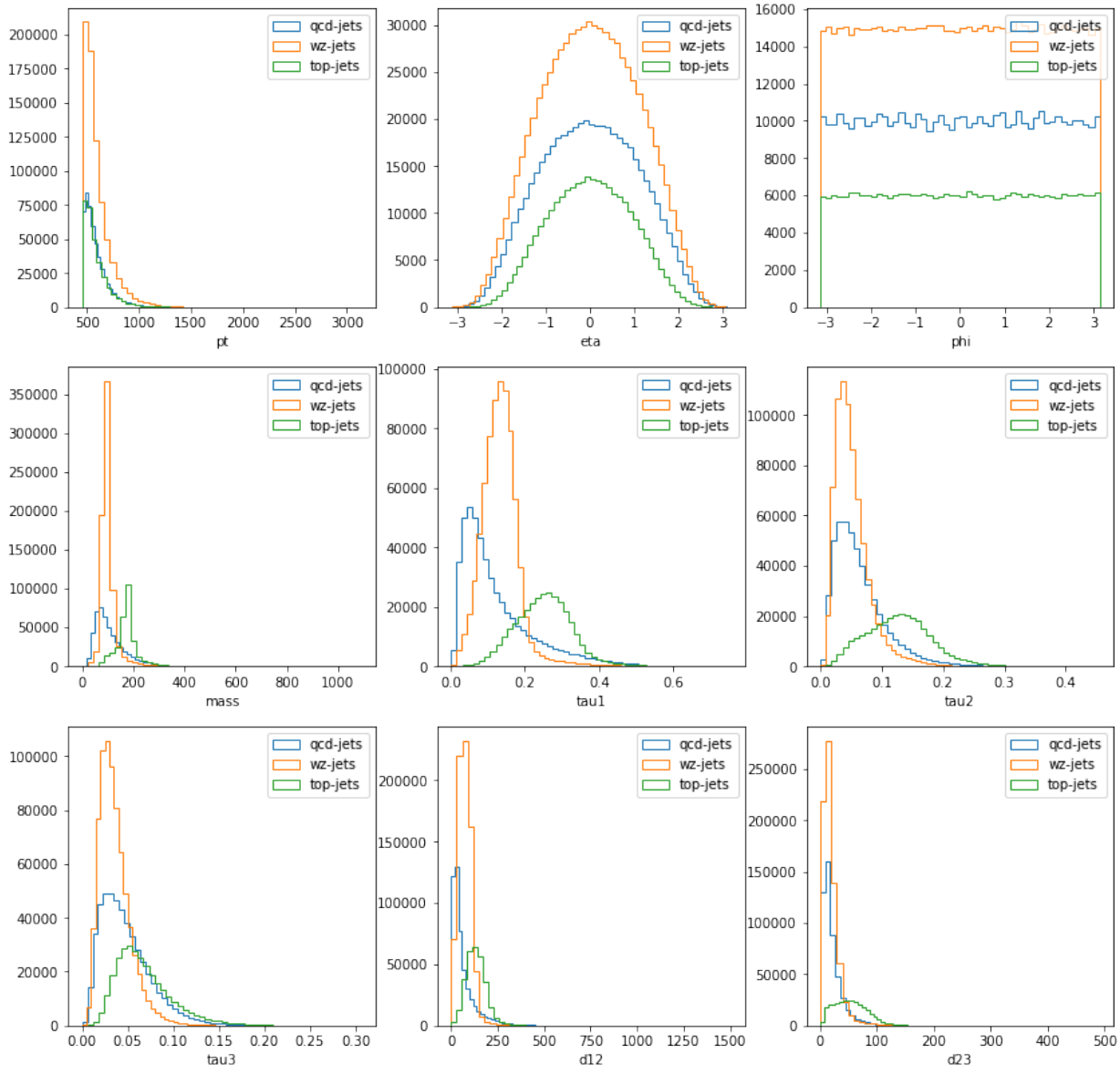

Figure 1.- Distribution of variables used for classification for each class

As we can see, some variables show similar distributions for each class, meaning that they could not be easily used for discrimination. The provided dataset does not contain any ill value hence only normalisation is required. Figure 2 shows the variable distributions after normalization.

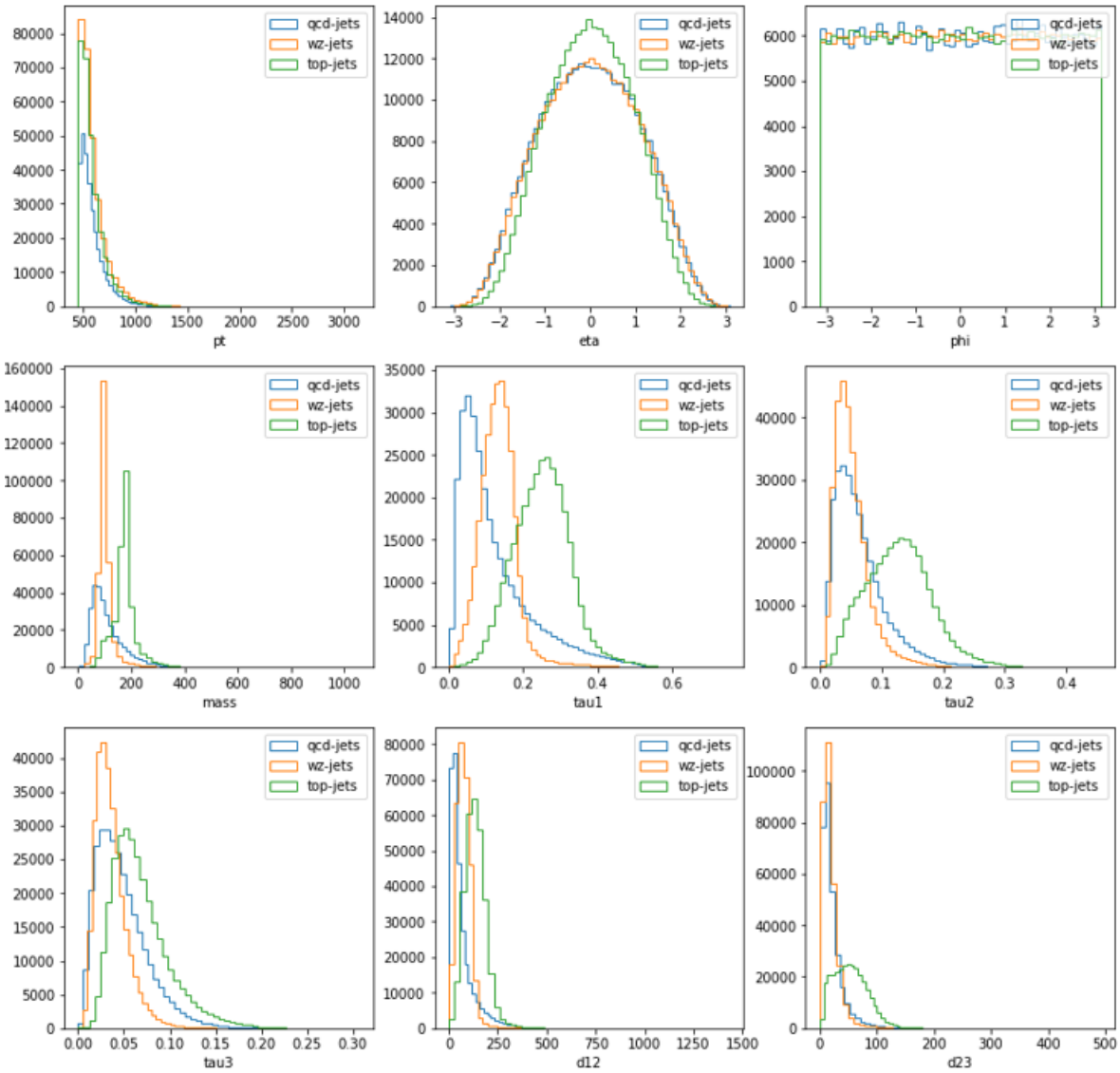

Figure 2.- Distribution of variables after normalization. The four momentum variables p_T, η, ϕ are know following a similar distribution.

Now that the data is normalized, we can see the four momentum components (p_T, η, ϕ) have similar distribution for each class, but the mass is still not the same for each class. Since we want the model to be sensitive to the substructure but not the four-momentum, the last step consists to sample the mass distribution to match for the three classes. Figure 3 shows the mass distribution after sampling.

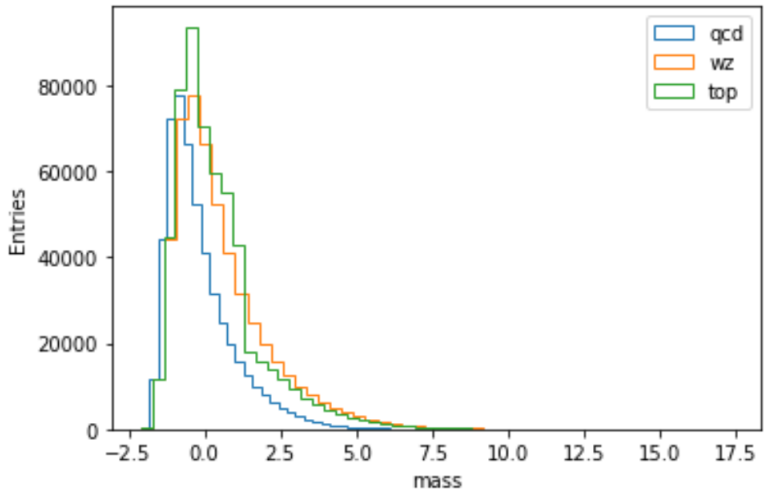

Figure 3.- Mass distribution for each class after sampling.

The presented variables will then be used as inputs to a fully connected network presented in section 3, both for a binary and a multi classification task.

The second part of the available data consists of the jet constituents. Similarly to what has already been done in [2], jet image representations can be built using the jet energy deposits in the eta/phi plane, which where then binned to form a 19 × 19 pixels image. Examples of individual images for each jet are shown in figure 4, and averages of jet images are shown in figure 5.

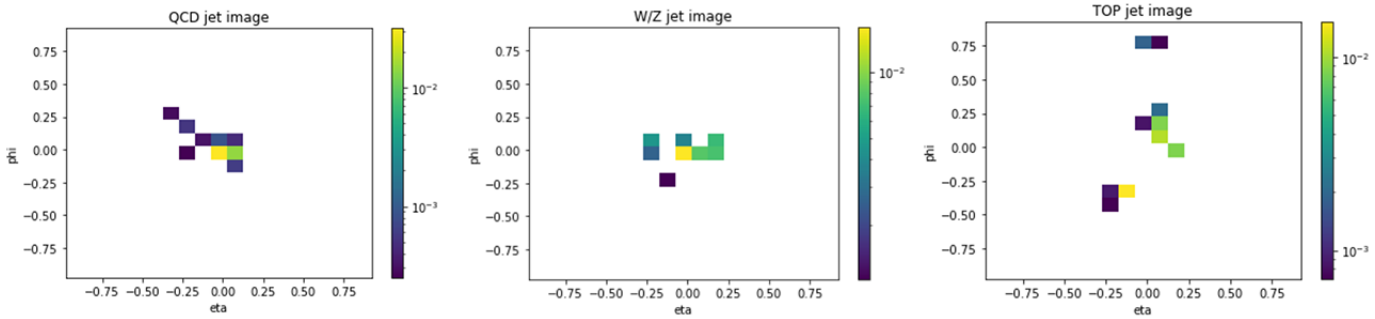

Figure 4.- Typical jet images from QCD, W/Z and TOP jets.

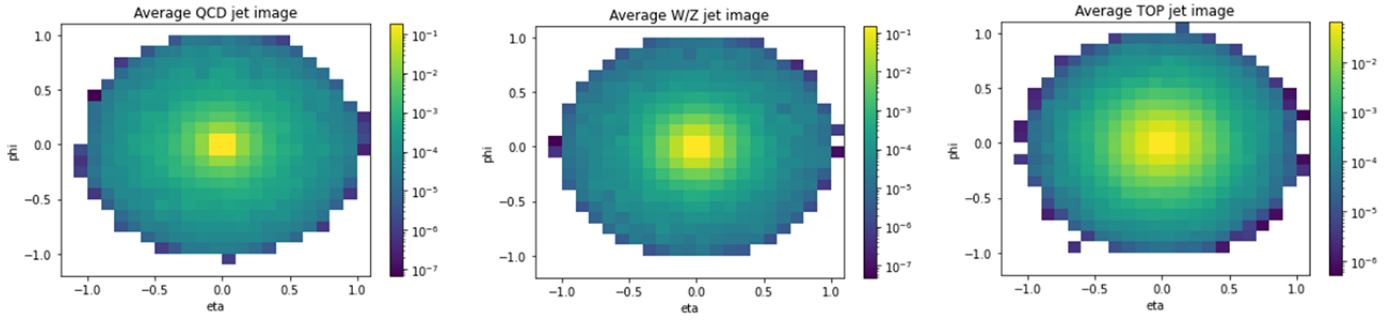

Figure 5.- Average of 50'000 jet images for QCD, W/Z, and TOP jets

As we can see, the images are already centered and need no further preprocessing other than pixel normalization to be used for training. The final step will then be to use the predictions from the CNN for each jet as new inputs used with the low and high level variables presented earlier.

3 Training

We now present the deep learning models that will be used for classification. For each model the hyperparameters where optimized by hand, based on three metrics : the training and validation loss and the area under curve (AUC) of the receiver operating characteristic (ROC) curve. The main goal was to maximize the AUC while preventing overfitting. In order to prevent the model of overfitting, dropout and batch normalization where applied during test sessions, and only dropout was kept for the final models. Different activation functions where tried such as the *leaky relu* unit, but were not found to be more efficient than the classical *relu* unit. It was found that the *batch size* would highly impact oscillations in the training and validation loss. Depending on the number of samples, using small batches (132) was found to reduce the fluctuations in the losses hence enhancing the efficiency of features such as *ReduceLrOnPlateau* which reduces the learning rate when a given metric (here : the training loss) does not improve for a few epochs.

3.1 Fully-connected neural network

The first type of model is a conventional fully connected model. It was trained on a training data set of 900'000 samples and was balanced using class weights.

Binary classification

For the binary classification task, the network consists of hidden layers of *relu* activation units and a *sigmoid* output with a binary cross-entropy loss function. The weights of the network were updated using the ADAM optimizer, and were initiated using the default *glorot uniform* initializer. The learning rate was set to 0.0001 and decreased by a factor 0.8 if the training loss did not improve after 3 epochs. Training was stopped if the validation loss failed to improve after 10 epochs. A sketch of the network architecture is given in figure 6. The network consists of 6 subsequent fully connected layers of [512, 256, 128, 64, 32, 16] units each. This network has about 180'000 trainable parameters and was found to be the best to perform for binary classification both on the loss optimization and on tagging efficiency (presented later on).

Multiclass classification

For multiclass classification, the network consists again of hidden *relu* activation units and a *softmax* output activtion with a categorical cross-entropy loss function. The weights were initiated using the default *glorot uniform* initializer and were updated using the ADAM optimizer. The learning rate was set to 0.0001 and decreased by a factor of 0.8 if the training loss did not improve after 3 epochs. Training was stopped if the validation loss failed to improve after 10 epochs. The multi class model has the same architecture than the binary one, with only difference in the output activation. It has about 180'000 trainable parameters and was found to be the best to perform for multi classification both on the loss optimization and on tagging efficiency.

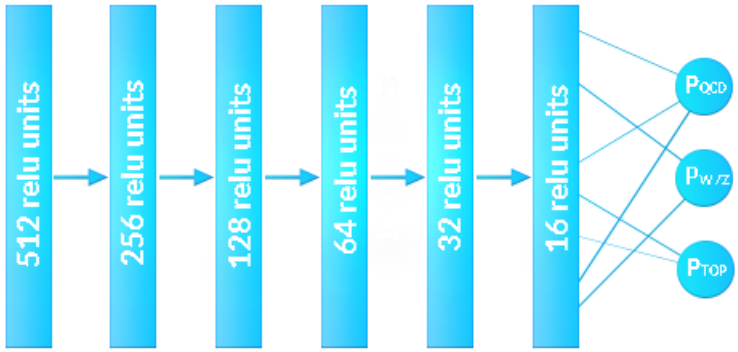

Figure 6.- Scheme of the network architecture. Each layer is composed of *relu* activation units. The output activation is a *softmax* unit. A dropout of 0.2 is applied between each layer.

3.2 Convolutional neural network

Moving on to using jet as images, we choose to implement a convolutional neural network consisting of two sequential [Conv + Conv + Max-Pool] layers followed by four fully connected dense layers of *relu* units, with *softmax* units as the output activation, and categorical crossentropy as loss. A sketch of the used architecture is given in figure 8.

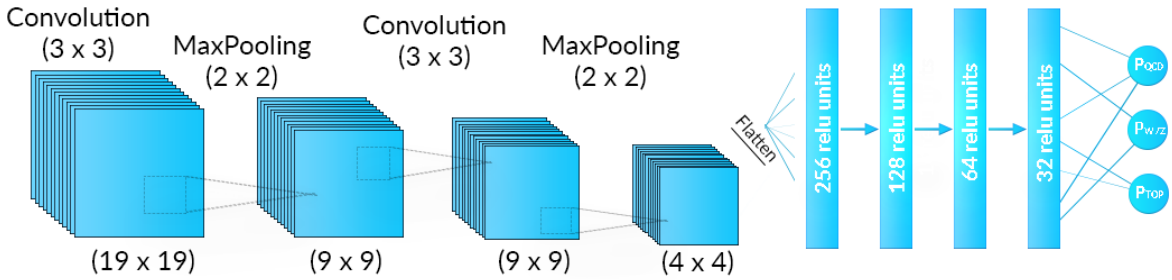

Figure 7.- Scheme of the architecture used for image classification. It consists of two sequential convolutional layers followed by a max pooling layer. This operation is repeated once and then followed by four fully connected layers of *relu* units.

The convolutional layers are using 32 filters with kernel sizes of 3×3 with *relu* activation units. The fully connected layers consist of a sequence of [256, 128, 64, 32] *relu* activation units. A dropout of 0.2 is used between each fully connected layer for regularization. The number of filters and the size of the kernel was optimized by hand, based on the evolution of the loss function, the AUC of the ROC curve and the tagging efficiency.
The model was trained over 100 epochs, with a learning rate initially set to 0.0001 and was decreased by a factor 0.8 if the training loss failed to improve after 3 epochs. Weights were initialized by the default *glorot uniform* initializer and were updated using the ADAM optimizer. The model weights and architecture were saved during training using keras callbacks in order to be used to get predictions on images in a broader network described right after.

3.3 Using CNN predictions as new inputs

Finally, an attempt is made to use the predictions from the CNN model on a set of jet images as additionnal inputs to the fully connected dense layers presented in section 3.1. In order to do so, we train a CNN as described in section 3.2 and save the model architecture with its current weights using callbacks (save at each epoch). After that, we select the best epoch to load based on the loss evolution per epoch. We then get the predictions of this model on a set of images and use the predictions as additional inputs along the variables described in section 2 in a fully connected model.

In order to make sure the CNN has not trained on the images to predict on, we make use of random seeds when splitting the data. The predictions are then made on the original *test set* from the dataset the CNN was trained on. We then split this test set into a new training, validation and test sets (fraction : 0.6, 0.2, 0.2) to train the full model. Because of this, if the original CNN was trained on 180'000 samples, the full model will train on 36'000 samples, considerably reducing the number of training samples and possibly affecting the performance of the network.

4 Results and discussion

The best performance for QCD jet identification is achieved by the multi class fully connected model using the mix of low and high level variables, without the CNN predictions as inputs. Each model was trained using the maximum number of jets that the computer would accept. This means that the fully connected networks (without images) were trained using 900'000 jets, whereas the CNN were trained on 180'000 jet images.
Final results are shown in table 1 with AUC used as a metric for QCD identification. The metric used is the Area Under the Curve (AUC), calculated in signal efficiency versus background efficiency, where a larger AUC indicates a better performance. In figure 8 the tagging efficiency, i.e : the signal efficiency versus background rejection is shown for Multi Class and CNN model. Figure 9 shows the tagging efficiency for Binary and CNN model on a QCD vs WZ dataset. Figure 10 shows the tagging efficiency of the Binary model and Multi class model for binary classification on a sample of QCD vs rest jets. Figure 11 displays the tagging efficiency of the Multi-Class model with and without CNN predictions as input, and figure 12 shows the AUC of the ROC curve of the Multi Class model for different mass values.

Inputs			Model	AUC
Variables	Images	CNN predictions		
✓			Multi Class FF	0.88
✓			Binary FF	0.87
	✓		Multi CNN	0.85
	✓		Binary CNN	0.85
✓		✓	Multi Class FF	0.85

Table 1.- AUC scores for each model for QCD identification. Each model was trained on samples of QCD, WZ, TOP jets. Binary model was trained on a sample of QCD and WZ jets.

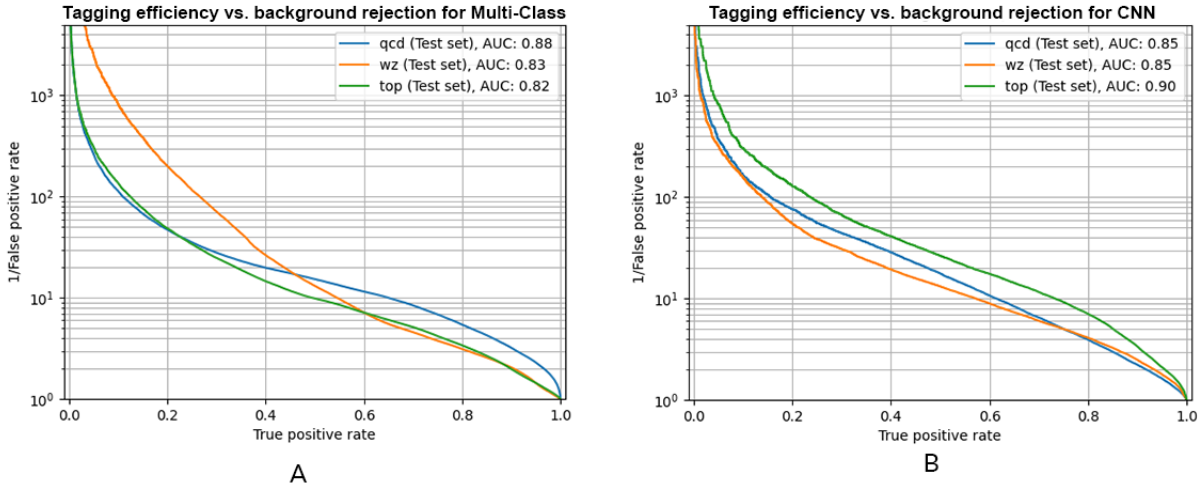

Figure 8.- Tagging efficiency for Multi-Class (A) and CNN (B) models. Both models where trained on 180'000 jets, using samples of QCD, WZ and TOP.

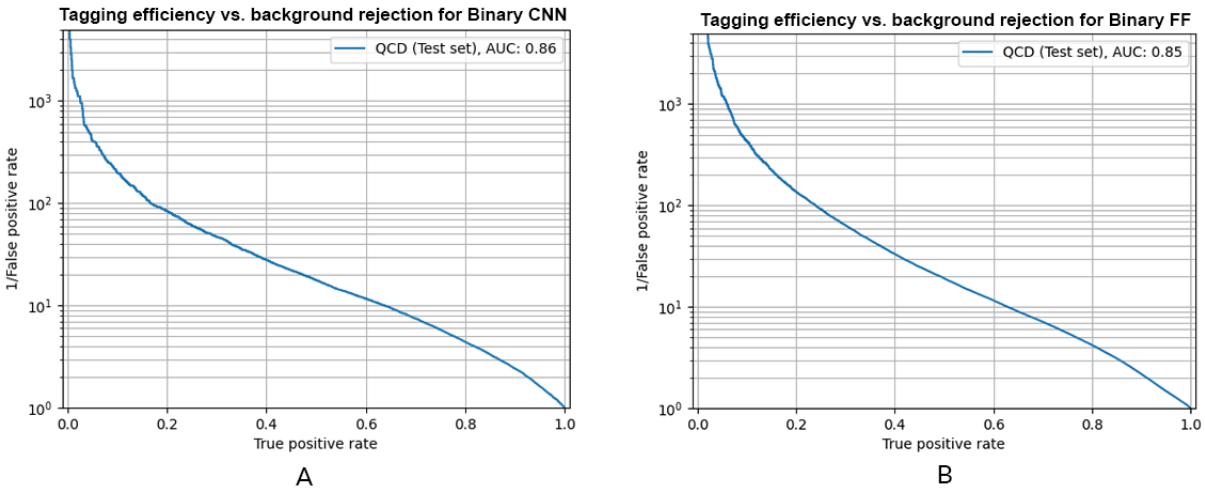

Figure 9.- Tagging efficiency for CNN (A) and Binary FF (B) models on a sample of 180'000 jets containing QCD, WZ class.

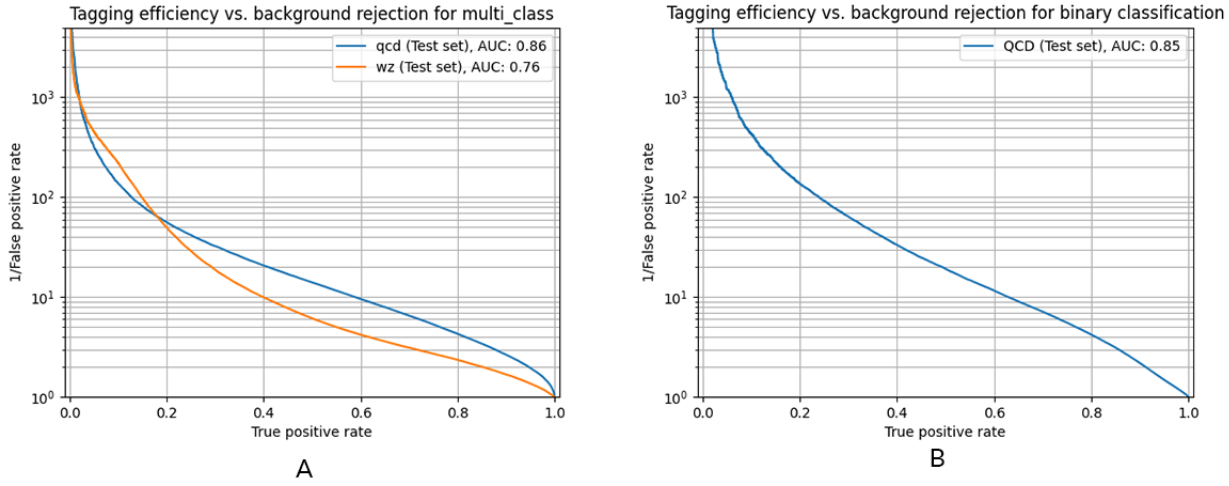

Figure 10.- Tagging efficiency for Multi-Class (A) and Binary (B) models. Both models are evaluated on the same sample containing QCD, WZ, TOP jets for a binary classification task (QCD vs Rest).

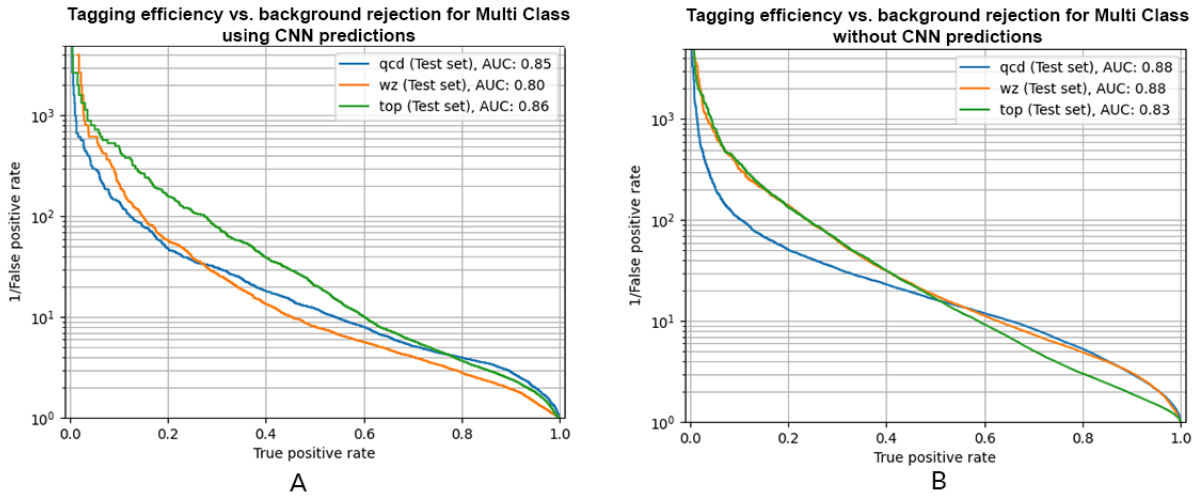

Figure 11.- Tagging efficiency for Multi-Class models with (A) and without (B) CNN predictions on images. Both models are evaluated on a sample of QCD, WZ and TOP jets.

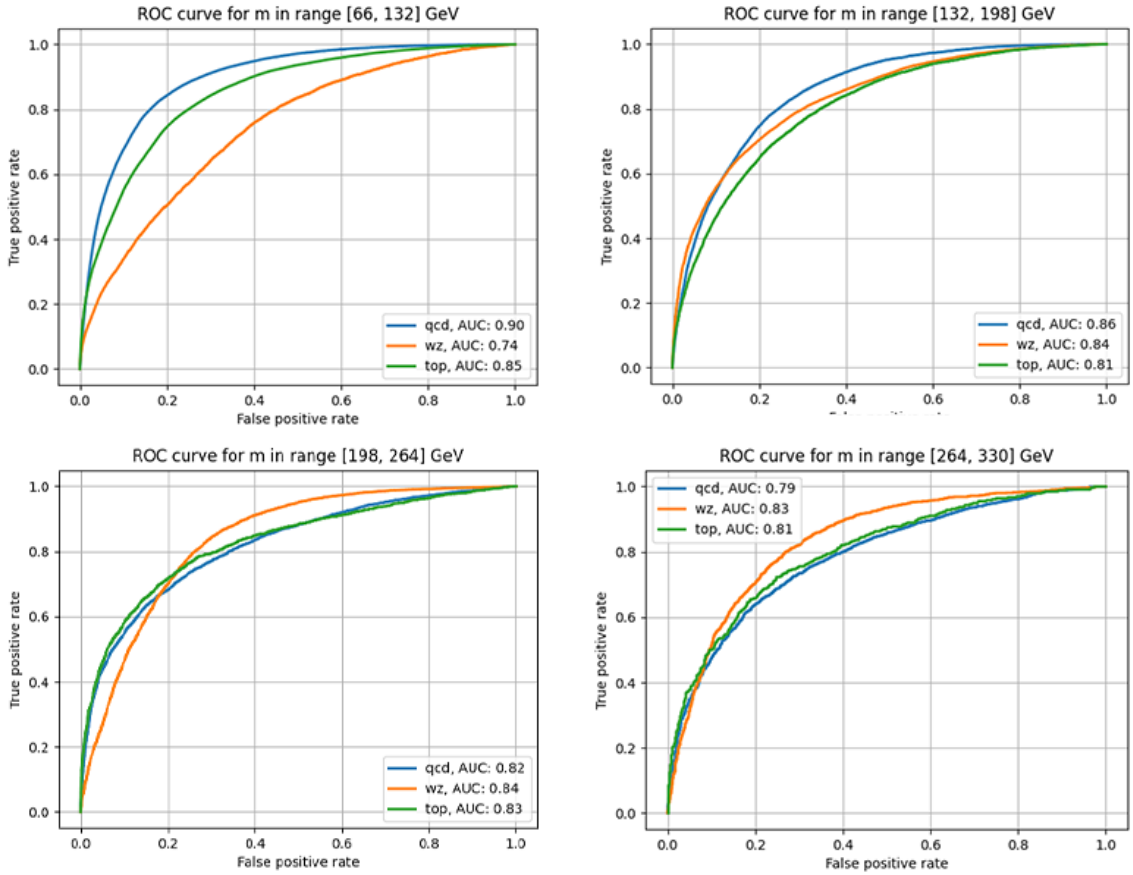

Figure 12.- Roc curve for different mass ranges for the Multi Class model.

The metric used to evaluate the performance of the model is the tagging efficiency. It seems to be the most relevant in particle physics since we are always trying to balance signal and background. Taking figure 8 as an example, if the tagging efficiency is 0.6 (we correctly label 60% of QCD jets), the Multi class model would mistake 1 out of 10 jets for QCD jets. The CNN model would also mistake 1 out of 10 QCD jets for this score, showing both models are viable. For figure 9, we can see for binary classification, both models show very close background rejection at every

tagging efficiency above 0.4. Figure 10 indicate the for a tagging efficiency of 0.6, the Multi class model mistakes 1 out of 9 jets for QCD, while the Binary model only does that for 1 out of 10 jets, showing slightly better performance. Finally, figure 12 shows the computed AUC for different mass ranges. As can we see, the AUC for QCD jets is still dependent on the mass, although for low and high mass values (below 50 GeV and above 300 GeV), the AUC is mainly reduced by the lack of samples, the model is still quite performant for the mass between 66 GeV and 264 GeV for QCD jet identification.

Overall, we can see that all three models show a similar performance for reasonable tagging scores. The Binary model shows slightly better performance on a one vs rest classification task.

The CNN model shows good performance both on binary and multi class classification, showing that jet images can be used as a good source for classifying jets.

Suprisingly, using the CNN predictions as additional input to the other variables has not shown to be successful as shown by figure 11. One possibility is that the number of training samples was too much reduced by the double splitting of the data described in section 2. Another possibility is that all of the information contained in the images is already fully captured by the other variables although this seems unlikely.

5 Conclusion

In this report, we have presented a study about using deep learning for jet identification using two different types of variables : a mix of low and high level variables in a fully connected network and image representations of jets using a CNN. The metrics used to optimize the models are the model's AUC of the ROC curve and loss evolution over epochs. The metric used to evaluate the performance of the model is the tagging efficiency. Our results show that both approaches are viable and show similar performance both for binary and multi class classification. The model's dependency with the jet four momentum is explored through its AUC score for the ROC curve for different mass values, and shows a slight dependency with mass values ranging from 66 to 264 GeV. This dependency could be reduced by using a better binning method for the mass distributions of the three classes. Finally, an attempt was made to use the predictions of the CNN model as an additionnal input to low (high) level variables which has not been successful.

Bibliographie

[1] ATLAS collaboration. Measurement of jet-substructure observables in top quark, W boson and light jet production in proton-proton collisions at $\sqrt{s}=13$ TeV with the ATLAS detector. *Journal of High Energy Physics 2019*, 8 (Aug. 2019), 33. http://arxiv.org/abs/1903.02942.

[2] Baldi, P., Bauer, K., Eng, C., Sadowski, P., and Whiteson, D. Jet Substructure Classification in High-Energy Physics with Deep Neural Networks. *Physical Review D 93*, 9 (May 2016), 094034. http://arxiv.org/abs/1603.09349.

[3] Bogatskiy, A., Anderson, B., Offermann, J. T., Roussi, M., Miller, D. W., and Kondor, R. Lorentz Group Equivariant Neural Network for Particle Physics. http://arxiv.org/abs/2006.04780, June 2020.

[4] Dreyer, F. A., and Qu, H. Jet tagging in the Lund plane with graph networks. http://arxiv.org/abs/2012.08526, Feb. 2021.