Fast Fourier Convolutional Neural Network (F-FCNN) for Boosted W $\rightarrow qq'$ Jet Substructure Classification

Resham Lal Sohal Jr., Jun Zhang, and Ramneet Kaur

In Fulfillment of the Requirements for PH-541 Final Project

Abstract

Collimated streams of high energy particles, known as jets, play a crucial role in the study of the properties of quarks and gluons, which are the building blocks of protons and other hadrons. After a proton–proton collision, quarks and gluons hadronise and radiate, producing jets of particles. To determine whether a detected jet is a result of a single low-mass particle or multiple decay objects from a massive particle, physicists need to separate the jets produced by the interaction of interest. This classification is critical for understanding the nature of the particles produced in the collision.

Physicists have developed advanced algorithms for identifying and analyzing jets. The latest generation of algorithms can detect individual particles within the jets and identify substructures of jets within jets. To date, numerous techniques have been employed for the classification of jets, including deep neural networks, attention-based neural network, and transformers. In an attempt to explore a new method, this project used Fast Fourier Convolutional Neural Network (F-FCNN). Results showed that F-FCNN trained on low level calorimeter images performed on par with XGBoost Classifier trained on high level calorimeter images, with an Area Under the Curve (AUC) score of 0.940 and 0.932, respectively.

1 Brief Background

At Large Hadron Collider, proton collisions lead to the production of new particles that allow us to explore the nature of our Universe. The primary focus is on the rare interactions that result in the production of elementary particles. To maximize chances of those rare interactions, LHC does not simply collide single protons, but rather large bunches of protons. Multiple protons interact when the bunches collide.

Particles from the interaction of interest are captured alongside particles from several other interactions, so-called pileup interactions. Particles that live for a tiny fraction of a second are reconstructed by looking at their decay products. These decay products often produce sprays or “jets” of particles.

Jets are the experimental signatures of quarks and gluons produced in the head on proton-proton collisions. As quarks and gluons have a net colour charge and cannot exist freely due to colour-confinement, they are not directly observed in Nature. Instead, they come together to form colour-neutral hadrons, a process called hadronization that leads to a collimated spray of hadrons called a jet. The identification of the jets from different objects is one of the major challenges at LHC. Radiation patterns of colourless objects (W/Z/H) differs from quark or gluon jets.

ML-based techniques can be customized to identify the origins of jets. For instance, we can look at the energy and direction measurements of the particles within a jet and project it onto two dimensions, so it looks like a regular image. The same machine learning algorithms that can tell the difference between two different faces in facial recognition can also tell the difference between different types of jets.

If the original particle has a high momentum, the jets originating from its decay products will merge. To perform our analyses, it becomes necessary to identify what kind of particle created the jets in our event. In some cases, the jet can have multiple distinct substructures or ”prongs”. Prongs are essentially separate streams of particles that are produced in different directions within the jet. Jet substructure uses a set of tools to exploit information from the radiation pattern inside these jets.

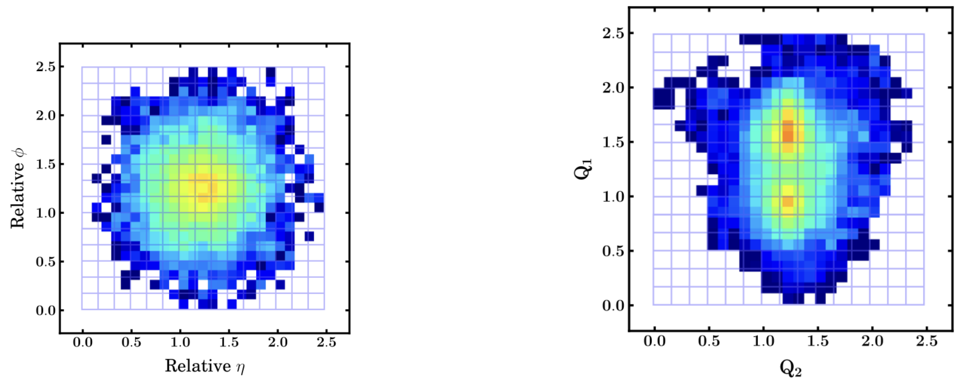

Figure 1: Jet image of a hadronically decaying W boson before (left) and after (right) preprocessing.

A jet image is a pixelated grayscale image, where the pixel intensity represents the transverse momentum of all particles that deposited energy in a particular location. Grayscale images are then expanded to include additional layers ('colors') to encode more information. Jet tagging approach focuses on learning the structure of the transverse energy distribution of a jet image. Accurate classification of jet images relies on the use of appropriate preprocessing techniques. Fig. 1 [1] shows the jet images before and after preprocessing.

2 Motivation and Element of Novelty

At sufficiently large boost above an order of $p_T > 200GeV$, the final state hadrons from the $W \rightarrow qq'$ decay merge into a single jet, and the traditional analysis techniques relying on resolved jets are no longer applicable. However, in such cases, the analysis of jet substructure can be used to identify those jets arising from decays of W bosons. Considering the fact that LHC will be transitioning to a high-luminosity upgrade [2], the problem may be even more difficult to handle due to an increased pile-up factor, reaching now up to 140 simultaneous pp interactions.

Josh Cogan (Sep 2015) [1] introduced a novel approach to jet tagging and classification through the techniques inspired by computer vision. This approach to jet tagging attempted to learn the structure of the transverse energy distribution of the pixels of the jet image. These jet images were preprocessed is for accurate jet-image classification. An example of a jet image before and after preprocessing is shown in figure 1.

Pierre Baldi (April 2016) [3] used DNN for jet substructure classification for jets. The samples of jets from $W \rightarrow qq'$ decay and jets from single quarks and gluons were generated in the range of $p_T \in [300, 400]GeV$. DNNs achieved an AUC score of 0.94 for classification of these sample jets. Boosted Decision Trees were used to train the data using 6 high level variables. BDTs achieved an accuracy of 0.932.

Recently (2022)[4], simple machine learning algorithms such as k-nearest neighbors (kNN) and support vector machine (SVM) were also employed to classify the jets. For 200k jets with pileup level of 80 and $p_T \in [400, 550]GeV$, the best AUC score is 0.800.

For the case of jet images, the natural machine learning tool for analyzing images is the convolution neural network (CNN). Modern machine learning algorithms trained on jet images can outperform standard feature-driven approaches to jet tagging. Moreover, improved accuracy in predicting jets, resulting from rare interactions, can provide new opportunities for discovering unexplored areas of physics.

Fourier analysis can be a powerful tool in jet analysis because it allows us to analyze the patterns that occur within jets. The distribution of particles within a jet often exhibit a pattern of radial symmetry, with particle density being highest near the center of the jet, and lowest further from it. This pattern of radial symmetry can be described mathematically using a Fourier series. In the case of a jet with radial symmetry, the Fourier series would include terms with low frequencies to represent the higher density, and higher frequencies to represent the lower density. Furthermore, jet substructure algorithms generally rely on identifying repeating structures within the jet. These structures can also be analyzed using Fourier analysis. By performing a Fourier transform on a jet image or any other representation of the jet, one can identify the frequencies that correspond to the periodic patterns within the jet. This can help in characterizing the substructure of the jet.

In an attempt to sort things out, we try to implement Fourier Convolutional Neural Networks for jet classification in $W \to qq'$ decay. This technique has never been used for classification of jets from the hardronic decay of W bosons so far.

3 Fourier Tranformation

The continuous Fourier transformation is as follows

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt. \tag{1}$$

In real life, we can only collect finite number of data points, so we have to use the discrete version of the continuous transformation above, i.e, Discrete Fourier Transformation(DFT), which is

$$\begin{aligned} F(\omega) &= \int_{0}^{(N-1)T} f(t) e^{-j\omega t} dt \\ &= f(0) + f(1)e^{-j\omega T} + f(2)e^{-2j\omega T} + \cdots + f(N-1)e^{-j\omega(N-1)T} \\ &= \sum_{k=0}^{N-1} f(k) e^{-jk\omega t}. \end{aligned} \tag{2}$$

One may notice that the range of the integral above is not from $-\infty$ to ∞. The rationale behind is that we assume the data points are periodic. We can generalize the above one-dimensional DFT to two-dimensional DFT [6], which can be written as

$$F_{i_1,i_2} = \sum_{j_1}^{m_u} \sum_{j_2}^{n_u} e^{-2j\pi(\frac{i_1 j_1 n_u + i_2 j_2 m_u}{n_u m_u})} f_{j_1,j_2}. \tag{3}$$

From eq (3) , it is known that the computational complexity of the DFT goes to $O(n^2)$, but this can be reduced considerably using an algorithm called Fast Fourier Transform (FFT). proposed by Cooley and Turkey. Accordingly, one can reduce the computational complexity of the DFT to $O(n\log_2 n)$.

Below, we showed how FFT works intuitively by using its matrix form [7].

$$\begin{aligned} \begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \end{pmatrix} &= \begin{pmatrix} W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 \\ W^0 & W^4 & W^2 & W^6 & W^1 & W^5 & W^3 & W^7 \\ W^0 & W^0 & W^4 & W^4 & W^2 & W^2 & W^6 & W^6 \\ W^0 & W^4 & W^6 & W^2 & W^3 & W^7 & W^1 & W^5 \\ W^0 & W^0 & W^0 & W^0 & W^4 & W^4 & W^4 & W^4 \\ W^0 & W^4 & W^2 & W^6 & W^5 & W^1 & W^7 & W^3 \\ W^0 & W^0 & W^4 & W^4 & W^6 & W^6 & W^2 & W^2 \\ W^0 & W^4 & W^6 & W^2 & W^7 & W^3 & W^5 & W^1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_4 \\ a_2 \\ a_6 \\ a_1 \\ a_5 \\ a_3 \\ a_7 \end{pmatrix} \\ &= \begin{pmatrix} 1 & \cdot & \cdot & \cdot & W^0 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & W^1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & W^2 & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & W^3 \\ 1 & \cdot & \cdot & \cdot & W^4 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & W^5 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & W^6 & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & W^7 \end{pmatrix} \begin{pmatrix} 1 & \cdot & W^0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & W^2 & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & W^4 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & W^6 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & W^0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & W^2 \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & W^4 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & W^6 \end{pmatrix} \begin{pmatrix} 1 & W^0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & W^4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & W^0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & W^4 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & W^0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & W^4 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & W^0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & W^4 \end{pmatrix} \begin{pmatrix} a_0 \\ a_4 \\ a_2 \\ a_6 \\ a_1 \\ a_5 \\ a_3 \\ a_7 \end{pmatrix}, \end{aligned} \tag{4}$$

The idea is that the matrix form contains sparse sub-matrices (lots of zeros). With this said, multiplying by the dense (no zeros) matrix on top is more expensive than multiplying by the three sparse matrices on the bottom. For $N = 2^r$, the factorization would involve r matrices of size $N \times N$, each with 2 non-zero elements. Thus, the FFT algorithm reduces the computational complexity from N^2 to $2Nr = 2N\log_2 N$.

3.1 Fourier Convolutional Neural Network

Convolutional Neural Networks (CNNs)[5] is a state-of-the-art machine learning algorithm that is capable of achieving good results for many computer vision tasks, for example, in the case of image and video analysis. Due to the significant recent advancements in the availability of processing power, the implementation of CNNs has become more practical. For instance, with Graphics Processing Units (GPUs), one can deal with the heavy computations required by the convolution operation that comes along with the algorithm.

However, there are downsides to CNNs. One of the main limiting aspects of CNNs is the computational cost of updating a large number of convolution parameters. Further, in the spatial domain, larger images take exponentially longer than smaller images to train on CNNs due to the operations involved in convolution methods. As a result, CNNs are often not a viable solution for large image computer vision tasks.

In the traditional Convolution Neural Networks (CNNs) [5], convolutions between images u^j and kernel functions κ^i are carried out via the sliding window approach. That is, a kernel matrix of some abritrary size is moved across an input image. The convolution is computed as the sum of the Hadamard product $\odot$ of the image patch with the kernel:

$$\tilde{z}^{i,j}_{k_1,k_2} = \sum_{\ell_1=\lfloor -m_k/2 \rfloor}^{\lfloor m_k/2 \rfloor} \sum_{\ell_2=\lfloor -n_k/2 \rfloor}^{\lfloor n_k/2 \rfloor} \kappa^i_{\ell_1,\ell_2} \odot u^j_{k_1-\ell_1,k_2-\ell_2}. \tag{5}$$

A Fast Fourier Convolution Neural Network (F-FCNN) [6], was proposed whereby training was conducted entirely within the Fourier domain, and the convolution above is replaced by

$$\mathcal{F}(z) = \mathcal{F}(\kappa) \odot \mathcal{F}(u), \tag{6}$$

where $\mathcal{F}$ represents the DFT introduced above. The equation above shows that after the DFT, the convolution simply becomes element-wise product which thus decreases the number of operations.

This paper aims to implement this method for jet substructure classification [3] and figure out underlying benefits, if there are any, that comes with it.

4 Dataset Description and Model Architecture

4.1 Dataset

Samples of boosted $W \to qq'$ were acquired from an existing work [3], with a center of mass energy $\sqrt{s} = 14TeV$ using the diboson production and decay process $pp \to W^+W^- \to qqqq$ leading to two pairs of quarks; each pair of quarks are collimated and lead to a single jet. Samples of jets originating from single quarks and gluons were simulated using the $pp \to qq, qg, gg$ process. In both cases, jets were generated in the range of $p_T \in [300, 400]\ GeV$.

All classifiers were trained on a training data set containing 10 million instances. The test data set, on the other hand, consists of 5M test instances. Two sets of features are available: the low-level calorimeter images, and the high-level derived features. There are also two versions of the dataset, one with pile-up and one without pile-up.

The low-level calorimeter images are the jet images, which were characterized by the energy deposits at different points on an approximately cylindrical calorimeter surface. The high-level derived features include 6 high level jet variables. These variables are the invariant mass of the trimmed jet, N-subjettiness ($\tau_{21}^{\beta=1}$), and 4 energy correlation functions ($C_2^{\beta=1}$,$C_2^{\beta=2}$,$D_2^{\beta=1}$,$D_2^{\beta=2}$).

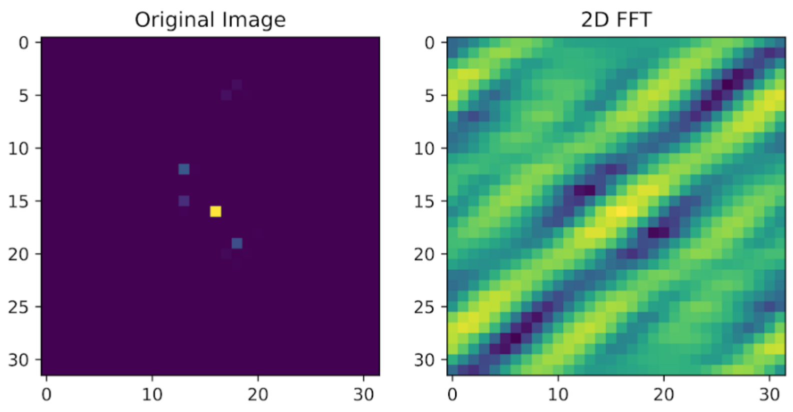

Figure 2: A depiction of how Fourier Transforms work on a QCD jet image. On the right, the image was transformed from the spatial domain to a frequency domain

4.2 Fast Fourier Convolutional Neural Network (F-FCNN)

Presented in this section are the deliverables set forth for the problem. Fast Fourier Convolutional Neural Network (F-FCNN) was used for jet substructure classification. The model introduced involves roughly 2.1M trainable parameters, with an outline of its architecture described below.

1. **Input Layer.** Contrary to a typical shape convention for image classifications, the corresponding shape notation from the dataset takes the form (channel, height, width). For the input layer, a Fourier Convolution (FC) takes in a shape of (1,32,32), with 32 filters, a batch size of 64, and initial kernel size of 3x3. For reference, the 2D-FC method used in this project was adopted from a Python compatible library called DeepSaki [8], so as to not implement everything from scratch.

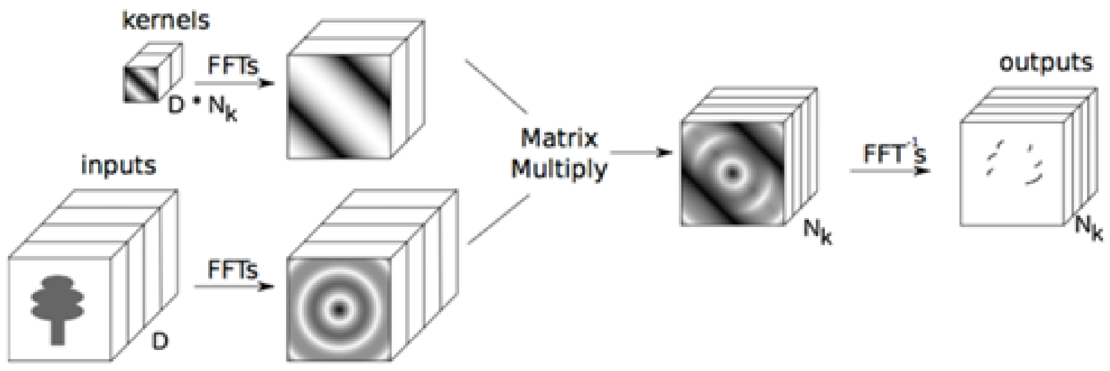

Figure 3: A general representation of how F-FCNN works [10].

2. **Subsequent Convolution Layers.** For the next few convolutions, the following was implemented:
   (a) An FFC with 32 filters and kernel size of 3x3
   (b) Max pooling of size 2x2
   (c) Two FC with 64 filters each and kernel size of 3x3
3. **Fully-connected Layers.** The output of the last convolution was flattened and fed to a first dense layer with 128 neurons. After which, a dropout factor of 0.2 was introduced, with an output fed to another dense layer consisting of 32 neurons.

4. **Output Layer.** To avoid overfitting, another dropout factor of 0.2 was used and results were fed to a single neuron with a sigmoid activation function. Such an activation function was used due to the binary classification nature of the problem.

For both the FFC and dense layers, the weights were initialized from a truncated normal distribution. Zero padding was also employed to the Fourier kernels to ensure that its of the same size as the image dimension.

4.3 Extreme Gradient Boosting (XGB)

XGB is a machine learning library that allows parallel tree boosting and can solve problems involving binary classification. An XGB classifier was used to fit the derived high-level features of shape notation (instances, features). An outline of the model parameters that were used to train the XGB classifier is shown below for reference.

Booster	n estimators	min split loss	max depth	min child weight	learning rate
Gradient Boosting Tree	240	0.1	15	0.025	0.07

Table 1: XGB classifier parameters used for training.

5 Training

To take realistic scenario into account, the dataset with pileup (addition of in-time pp interactions) was used for the entire project. For clarity, two types of datasets were used, namely low-level calorimeter energy hits and high-level features. The idea behind the training phase is shown below.

5.1 Low-level Calorimeter Energy Deposits

As previously mentioned, the pixel size associated to this generated dataset is 32 x 32. A Python library called Dask [11] was used to load the training dataset in chunks. Specifically, 80% was used for training while the remaining 20% was set for validation. The model was compiled using Adam optimizer with an initial learning rate of 0.001. Binary cross entropy was used as a loss function and the model performance was monitored using accuracy as a performance metric. To allow the model to converge to an optimal solution, a learning rate scheduler and early stopping callbacks were also introduced. The initial learning rate was decreased by a factor 0.1, provided that no changes of validation loss were observed for three epochs. The model was then trained for 100 epochs, where the validation loss plateaued after training for about 44 epochs. Moreover, the model was tested on 5M instances on the given test dataset, and a Receiver Operating Characteristic − Area Under the Curve (ROC AUC) was plotted as a measure of performance due to its interpretability.

5.2 Derived High-Level Features

For this dataset, the same split was done as above. XGB classifier was used for training, with its performance evaluated on a test set containing 5M instances. In total, there were six features, and the impact of each feature was explored by plotting an associated ROC curve. The AUC results obtained here were then compared to the results obtained by the F-FCNN.

6 Results

In this section, the results of the afformentioned methodology were presented. It was found out that the neural network designed in this project performed slightly better than the XGB classifier that was trained on all features. While it is true that the result difference is only of small margin, as shown in Table 2, one should take note that F-FCNN was trained on low level inputs. This implies that the network was able to extract relevant features that contributed to a better background and signal discrimination compared to an expert-derived features.
On another note, a traditional 2D-convolutional neural network (CNN) was also made based on the F-FCNN architecture, having roughly the same number of trainable parameters. As summarized in

Model	AUC
F-FCNN	0.940
Traditional Conv2D	0.946
Stacked FFConv2D and Conv2D	0.9395
XGB (trained on hjgh-level features)	0.932

Table 2: AUC comparison of models used in this study.

Table 2, the traditional CNN performed slightly better than F-FCNN. It also outperformed F-FCNN in terms of training time by as much as six times in magnitude, which means that the FFT Convolution adopted from DeepSaki was not optimized. As a matter of fact, such an outcome is consistent with a GPU Speedtest comparison [12], noting that a Pytorch FFT implementation is 4.5 times faster than a GPU-supported Tensorflow implementaion. Moreover, it turned out that a simpler model will cater the needs of this problem better, as opposed to the model complexity that F-FCNN offers. We have figured out that this was the case as we experimented on using a stacked model consisting of F-FCNN and Conv2D layers. With this method, however, no significant boost of performance noticed.

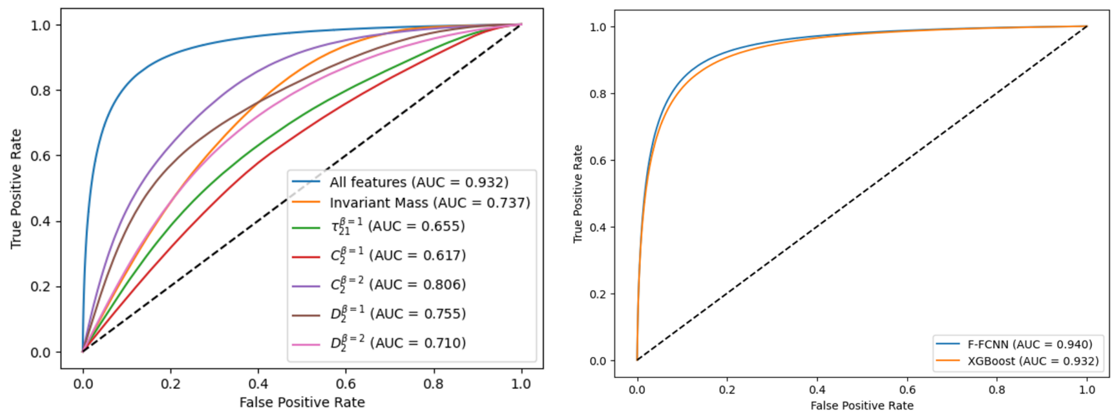

Figure 4: ROC curve for each high-level feature trained using XGB classifier solely *(left)*, and graphical representation of the tabulated results in Table 2 *(right)*.

The impact of each feature in distinguishing whether a jet is a result of boosted W $\rightarrow qq^{'}$ or quark/gluon topology is shown in the figure above. It is clear from the graph that the energy correlation function $C_2^{\beta=2}$ is the most important feature that one can use to effectively classify a jet as a signal or background. This rationale may only be valid for this particular jet tagging problem, as particle physicists treats N-subjettiness as the default feature [16] for any analysis involving boosted particle decays. However, this feature alone may be insufficient when the average pp interactions is increased significantly. Thus, to acquire a maximum AUC, it is of best interest to make use of all the high-level input features.

As an additional step, a little modification was done to potentially boost the proposed model's performance. A marriage of a Convolutional Neural Network and XGB classifier was implemented. This kind of technique is called Convolutional-XGBoost, where the idea behind is to use CNN as a feature extractor and let XGB do the jet classification job. Apparently, as recorded by existing literature [13, 14, 15], this method both leads to a performance boost (if the model is simple), or declined performance (if the model of interest is way too complex).

With this tweak, only $\approx +0.01$ AUC was noticed despite having the best XGB parameters. A confusion matrix report is shown on the next page for reference. As one can notice, an F-FCNN + XGB model became better in signal jet classification, while a traditional CNN + XGB became better in background jet classification. One can argue that such results have further implications on the applications of machine learning algorithm in particle physics. For instance, representing a dataset in Fourier domain could potentially be good for quests involving exotic particle searches.

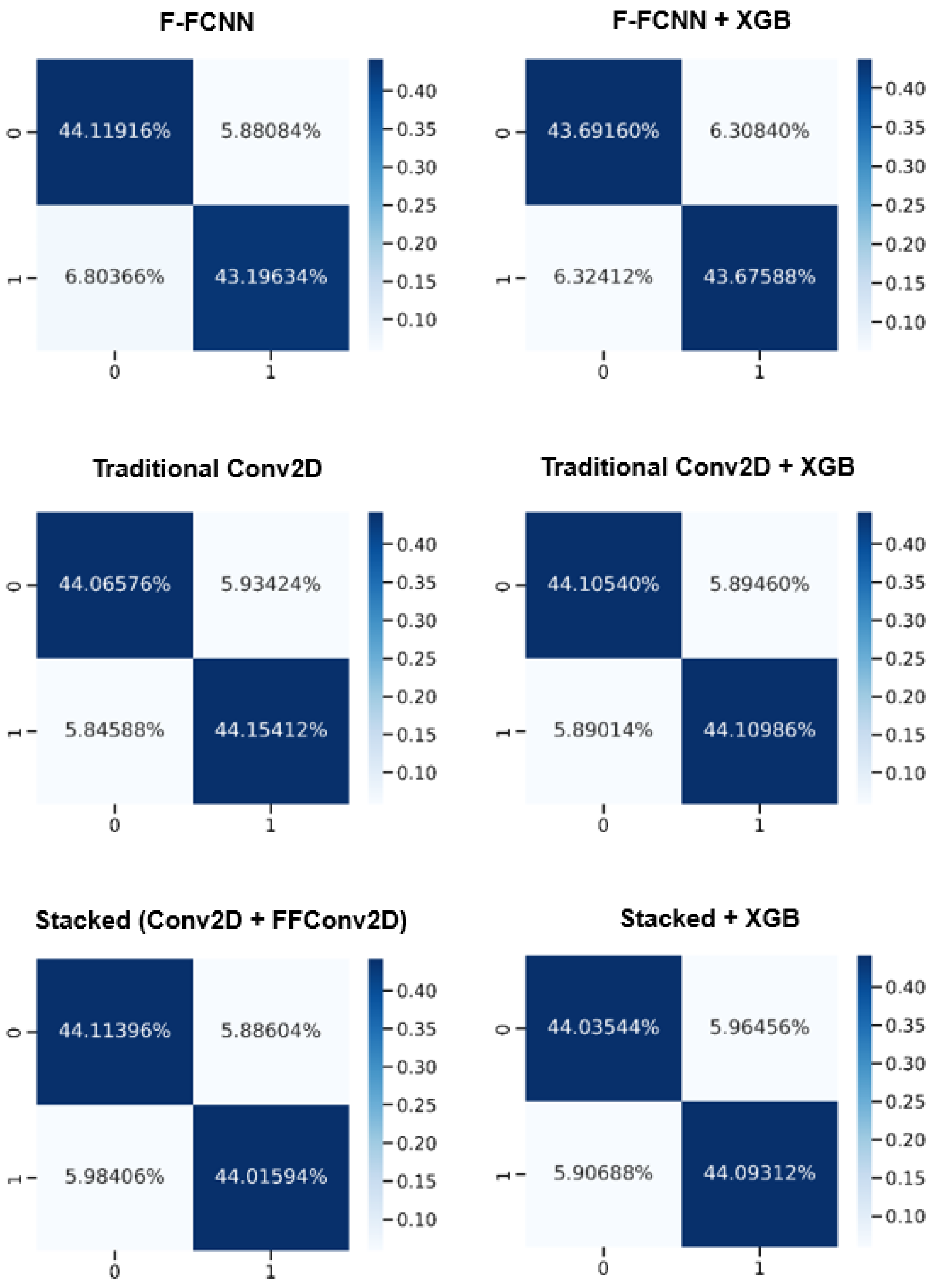

Figure 5: Confusion Matrix Reports for Model alone (*Left*) and its Convolutional-XGB counterpart (*Right:*)

7 Conclusions and Recommended Further Work

In this paper, we proposed an existing machine learning algorithm for boosted jet substructure classification. With the results presented, it was apparent that F-FCNN, trained on low-level features, performed on par with a boosted decision tree trained on expert-derived features, though it may not be an ideal tool to use. While the researchers were hoping for a better accuracy boost for a training time trade-off, when compared to the traditional CNN model [17, 18], this happened to be not the case. An underlying factor that could have affected such an outcome can be attributed to size of the jet images. To get around the issue of training time while preserving the accuracy for small images, it is advisable to crop the resulting convolution instead of applying a zero kernel padding. While the process involved in cropping could possibly truncate some important features [9], it could possibly lead to a faster training time. As a further work, one can also investigate the use of a Fourier pooling layer instead of a Max pooling layer and check how the performance is affected by such a tweak. An extra computational effort involving F-FCNN+XGB model, trained on low-level features, is worth looking at, as performance boosts can be obtained.

References

[1] Cogan, J., Kagan, M., Strauss, E. et al., "Jet-images: computer vision inspired techniques for jet tagging", J. High Energ. Phys. 2015, 118 (2015). https://doi.org/10.1007/JHEP02(2015)118

[2] HL-LHC Preliminary Design Report: FP7 HL-LHC Design Study, CERN-ACC-2014-0300, Retrieved from: $http : //cds.cern.ch/record/1972604/files/CERN - ACC - 2014 - 0300.pdf$. Accessed on April 12, 2022.

[3] P. Baldi, K. Bauer, C. Eng, P. Sadowski and D. Whiteson, "Jet Substructure Classification in High-Energy Physics with Deep Neural Networks," Phys. Rev. D **93**, no.9, 094034 (2016) doi:10.1103/PhysRevD.93.094034 [arXiv:1603.09349 [hep-ex]].

[4] Cai, T., Cheng, J., Craig, K., Craig, N. "Which metric on the space of collider events?," Phys. Rev. D 105, 076003 (2022) doi:10.1103/PhysRevD.105.076003 https://link.aps.org/doi/10.1103/PhysRevD.105.076003

[5] A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M. D. Messier, E. Niner, G. Pawloski, F. Psihas, A. Sousa and P. Vahle. (2016). "A Convolutional Neural Network Neutrino Event Classifier," JINST **11**, no.09, P09001 doi:10.1088/1748-0221/11/09/P09001 [arXiv:1604.01444 [hep-ex]].

[6] H. Pratt, B. Williams, F. Coenen, and Y. Zheng. (2017). "FCNN: Fourier Convolutional Neural Networks," ECML/PKDD .

[7] H. Pratt, B. Williams, F. Coenen, and Y. Zheng. (1995). "Fourier Transforms and the Fast Fourier transform (FFT) algorithm," Computer Graphics

[8] S. Kirch. (2022). "DeepSaki Python Library for Deep Learning." Retrieved from: $https : //github.com/sascha - kirch/DeepSaki/$. Accessed on April 12, 2022.

[9] Y. Han and B.W. Hong. (2021). "Deep Learning Based on Fourier Convolutional Neural Network Incorporating Random Kernels," Electronics **10**(16), 2004. DOI: https://doi.org/10.3390/electronics10162004

[10] T. Highland. (2015). "Efficient Training of Small Kernel Convolutional Neural Networks using Fast Fourier Transform," Wrigth State University, Browse all Theses and Dissertations, 1398. Retrieved from $https : //corescholar.libraries.wright.edu/etd_all/1398$

[11] Dask Array Documentation. Retrieved from: $https : //docs.dask.org/en/stable/array.html$ Accessed on April 18, 2023.

[12] T. Aarholt. (2019). "Speed-testing python implementations of the 2D FFT on CPU and GPU using Tensorflow 2, PyTorch, CuPy, Numpy and FFTW." Retrieved from: $https : //thomasaarholt.github.io/fftspeedtest/fftspeedtest.html$

[13] J. Azpur (2021). "CNNs and Gradient Boosting for Image Classification". Retrieved from https://jonaac.github.io/works/deepxgboost.html. Accessed on March 28, 2023.

[14] M.S. Khan, N. Salsabil, M.G.R. Alam et al. (2022). "CNN-XGBoost fusion-based affective state recognition using EEG spectrogram image analysis," Sci Rep 12, 14122. DOI: https://doi.org/10.1038/s41598-022-18257-x

[15] M. Babayomi, O.A. Olagbaju, A.A. Kadiri (2023). Convolutional XGBoost (CXGBOOST) Model for Brain Tumor Detection. Electrical Engineering and Systems Science, Image and Video Processing. DOI: https://doi.org/10.48550/arXiv.2301.02317 [arXiv:2301.02317v1]

[16] The CMS Collaboration. (2022). "Workshop on Physics Objects: Jet Substructure.". Retrieved from $https : //cms - opendata - workshop.github.io/workshop2022 - lesson - physics - objects/06 - substructure/index.html$. Accessed on April 12, 2023.

[17] L. Chi, B. Jiang, Y. Mu. (2020). "Fast Fourier Convolution,". 34th Conference on Nueral Information Processing Systems (NeurIPS). ISBN: 9781713829546.

[18] V. Nair, M. Chatterjee, N. Tavakoli, A. S. Namin and C. Snoeyink (2020). "Optimizing CNN using Fast Fourier Transformation for Object Recognition,". 19th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 2020, pp. 234-239. DOI: 10.1109/ICMLA51294.2020.00046.