PROCEEDINGS OF SCIENCE

Study for jet flavor tagging by using machine learning

Masahiro Morinaga,[a,b,*] **Masahiko Saito,**[a,b] **Junichi Tanaka,**[a,b] **Sanmay Ganguly**[a,b] **and Tomoe Kishimoto**[a,b,c]

[a] *International Center for Elementary Particle Physics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan*

[b] *Institute for AI and Beyond, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan*

[c] *Computing Research Center, High Energy Accelerator Research Organization, 1-1 Oho, Tsukuba, Japan*

E-mail: morinaga@icepp.s.u-tokyo.ac.jp

In particle collisions like the Large Hadron Collider (LHC), a large number of physical objects, called jets, are created. They originated from hadrons, gluons, or quarks, and it is important to identify their origin. For example, a b-jet produced from a bottom quark has features, which can be used for its identification called a "b-tagging" algorithm, enabling precise measurement of the Higgs boson and search for other new particles from the beyond the standard model. Machine learning models have been proposed by various researchers to identify jet flavors, but only for specific flavor classification, e.g., classification of the bottom quark and other quarks/gluons (b-tagging), or classification of quarks and gluons (quark and gluon separation). In this study, we propose a method and show its results, where we extend the classification to all flavors except top quark: b/c/s/d/u/g at once using a modern method based on a recently developed training strategy for image recognition models.

International Symposium on Grids & Clouds 2022 (ISGC 2022)
21 - 25 March, 2022
*Online, Academia Sinica Computing Centre (ASGC), Taipei, Taiwan****

*Speaker

PoS(ISGC2022)031

1. Intorduction

Particle identification is an important technique in particle physics experiments, such as the Large Hadron Collider [1]. At the LHC, jets are produced from the decay products of quarks and gluons. Once the flavor of the jet is determined, various physical processes can be identified. For example, bottom quark tagging (b-tagging) is useful for measuring Higgs boson properties, because approximately 58% of Higgs bosons decay into bottom quark pairs.

Machine learning techniques have historically been used for particle physics experiments [2, 3]. In recent years, deep learning has been used extensively and has dramatically improved particle identification capabilities compared to traditional neural networks. Jet flavor identification is the problem of classifying jet flavors (up quark, down quark, etc.) using jet features as input. Since jets are reconstructed from calorimeter clusters and charged particle tracks, high-level features such as collision parameters of charged particle tracks and average distances between each track have been used as input. The development of deep learning has improved the performance of methods that provide only low-level information to neural networks and also perform feature extraction.

There are two main types of jet flavor tagging: quark and gluon (q/g) tagging and bottom quark (b) tagging. The q/g-tagging classifies quarks from gluons, while the b-tagging classifies b-jets, c-jets (heavier flavor jets), and other quarks (lighter flavor jets). This q/g-tagging and b-tagging run on different models because they have different features for classifying quarks and gluons, b/c-jets and light-jets. For practical purposes, however, it is desirable to have a single model that can classify all flavors. Jet Image [4] is a method that creates images from jet components (clusters, tracks) and uses image recognition techniques to identify jet flavors.

In this study, we studied models and learning methods for classifying jet flavors, including heavy flavors and gluons. For versatility and practicality, we used a jet image classification method based on a model that has achieved remarkable results in the image recognition field.

This paper is organized as follows. Section 2 describes related works. Section 3 summarizes the datasets used in this study. Section 4 provides details of the proposed model. Section 5 gives the results of the experiments. Section 6 is for the conclusion.

2. Related work

Machine learning such as Boosted Decision Tree (BDT) was used for jet-flavor classification tasks but is being replaced by developments in deep learning. Jet Image [4] pixelized jets using the relationship between the jet components and their position relative to the jet center to create an image with track momentum, cluster momentum, and the number of components in each pixel represented in RGB. The features extracted by the CNN from the imaged jet are then used to perform flavor classification of the jet. Although Jet Image uses CNNs, we use a relatively new model, gated MLP (gMLP) [5] in this study, because the performance of newer models such as Vision Transformer (ViT) [6] and MLPMixer [7] improved significantly over the past few years. These new models are based on a similar structure called the Metaformer [8].

Compared to previous studies, our study is novel in the following respects:

- A neural network model based on the Metaformer structure is proposed.

- A model that classifies all quark flavors (except top quark) and gluons is constructed.
- Decrease correlation between jet property and neural network output using the FiLM layer [9].

3. Datasets and input definition

The training data sample in this study was produced with particle physics simulations: proton-proton collision events generated by MadGraph4_aMC@NLO [10] at the center of mass energy of 13 TeV, with showering and hadronization performed by Pythia8 [11] and detector response simulated by Delphes [12]. The quark pairs and the gluon pair are produced by Fig. 1 (a) and (b) respectively.

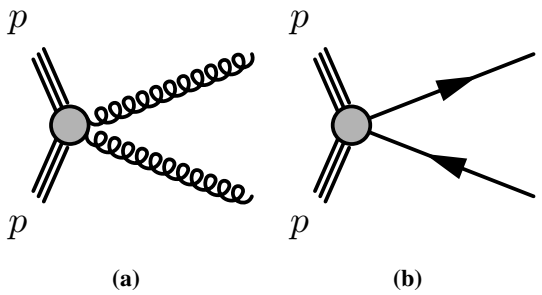


Figure 1: (a) Feynman diagram for a gluon pair production. (b) Feynman diagram for a quark pair production.

The hadronic jet is reconstructed with the anti-k_{T} algorithm [13] with a radius parameter $\Delta R = 0.4$. Jet is required to have transverse momentum greater than 20 GeV and absolute value of pseudorapidity less than 2.4, i.e. $p_{\mathrm{T}} \geq 20\,\mathrm{GeV}$ and $|\eta| \leq 2.4$.

Charged tracks and hadron clusters in the reconstructed jet were converted to images at η, ϕ = 0.0125 units. The jet image proposed in this paper has five channels, the channel is the third dimension of the image, e.g. RGB in photography. The number of tracks, electromagnetic clusters and hadronic clusters is assigned to the 1st, 2nd and 3rd channels. The 4th channel consists of a sum of the transverse momentum of charged tracks and hadronic clusters. The 5th channel is a sum of the transverse impact parameter of charged tracks. The transverse impact parameter is defined as the distance of the closest approach of the track to a reference point, in the r-ϕ projection.

A preprocessing is applied to the jet image. The preprocessing consists of simple normalization and standardization described as follows:

$$x'_{i,j,c} = (x_{i,j,c} - \mu_{i,j,c})/\sigma_{i,j,c} \tag{1}$$

where $x_{i,j,c}$ is a value of (i, j)-th pixel of the c-th channel, $\mu_{i,j,c}$ and $\sigma_{i,j,c}$ are a mean and standard deviation of (i, j)-th pixel of the c-th channel. The mean and standard deviation are calculated using the same number of samples for all flavors per pixel.

The input for the FiLM layer is a set of jet properties, transverse momentum, pseudorapidity, the number of charged tracks and the number of hadronic clusters. The distributions of transverse momentum and pseudorapidity are similar, but the number of charged tracks and neutral clusters have different distributions among the flavors.

4. Neural Network Model

The neural network model proposed in this paper is briefly shown in Fig. 2. The jet image is decomposed into 4x4 patches and features suitable for classification are acquired in the mixing layer. The patch is a container of pixels in a larger form, in this study, an image patch has 4x4 pixels. The original image consists of 32 x 32 pixels and the patches consist of 4 x 4 pixels. This means that there are 64 patches in one image. The mixing layer consists of a residual connection with a special component to mixture the features with a spatial relation. The model repeats the mixing layer L times, an inner dimension of the mixing layer is 64 in this study. To mix spatial and channel dimensions, the mixing layer has the spatial gating unit (SGU) described in Fig. 2.

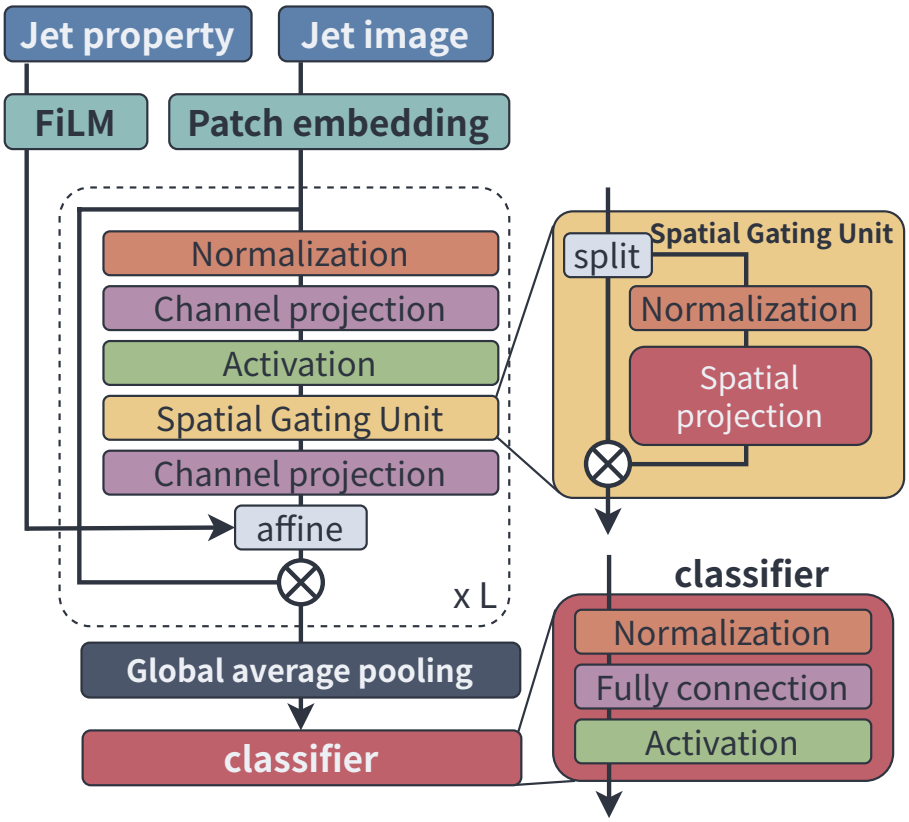


Figure 2: Model structure proposed in this paper

In the mixing layer, the Layer Normalization [14], channel projection, the GELU activation [15], the spatial gating unit and channel projection are applied in that order.

The FiLM layer repeats simple multi-layer perceptons (MLP) stacked 32 layers with 8 inner dimensions without any activation and normalization layers. The output from the FiLM layer is used for affine transformation before the residual connection of the mixing layer. The input to the classifier is passed through the global average pooling layer after the final layer of the mixing.

The classifier consists of simple MLP with a fully connected layer, the Layer Normalization and the GELU activation. In order to correct an imbalance in sample size between flavors, the class balanced loss [16] is utilized. The class balanced loss calculates a softmax cross-entropy loss as

follows:

$$\mathcal{L}_{\mathrm{CB}}(z, y) = -\frac{1-\beta}{1-\beta^{n_i}} \log\left(\frac{\exp(z_y)}{\sum_{j=1}^{C}\exp(z_j)}\right) \tag{2}$$

where β is a hyperparameter that should be given before training, n_i is the number of samples in the training with a label of i, C is the total number of flavors, z and y are a logit of the output and label of the training sample.

5. Experiments and results

Our codes for this experiment are implemented using PyTorch [17]. NVIDIA Tesla A100 and V100 are used for all execution.

Table 1: The hyper-parameters of the training.

#of samples per flavor	1048576
#of epoch	100
Batch size	2,048
Loss function	CB softmax cross-entropy
β	0.9999
Optimizer	RAdam [18]
β	$(0.9, 0.999)$
weight decay	0.01
lr scheduler	cosine annealing wramup [19]
warmup period	5 epochs
learning rate	0.01 - 0.0001

Table 1 summarizes hyper-parameters for the training. The validation sample is randomly taken from 10% of the training sample. The training is performed up to 100 epochs, and the best epoch for the validation data is used as the final weight parameters.

Fig. 3 shows the accuracies of the event classification during the training. It can be confirmed that the training data and validation data show similar performance from Fig. 3, so the overfitting is well suppressed. At the beginning of the training, the values are unstable due to the relatively high learning rate. In terms of accuracy, the b-jet has the best performance, followed by the c-jet and g-jet. The u-jet and d-jet are almost inseparable, so the labels were merged and learned. Surprisingly, s-jet and u/d-jet are somewhat separable and their performance is as well as u/d-jet. The best result is the last epoch, which is adopted as the model weight in this study.

Fig. 4 shows the confusion matrix between true labels and predicted labels by the proposed model. The purity is the percentage of each true label in the classified sample. The efficiency is the percentage of each true label that indicates which label it was classified as. As can be seen from Fig. 4, b-jet and c-jet, u/d-jet and s-jet are very similar and are often confused. This trend is also seen in other flavor tagging algorithms, where the separation of b-jet and c-jet is a major challenge.

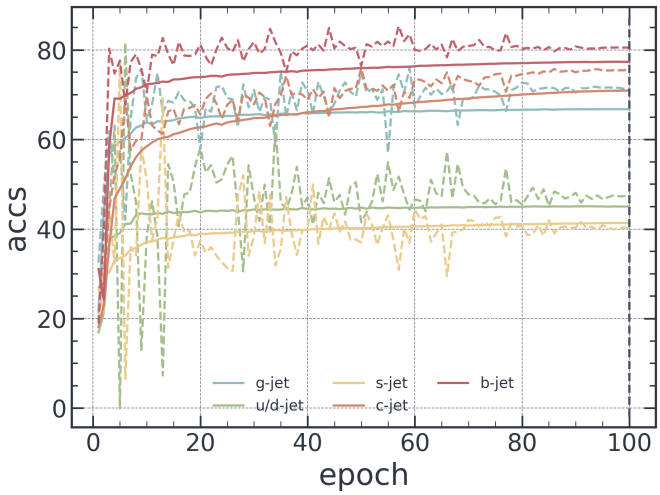
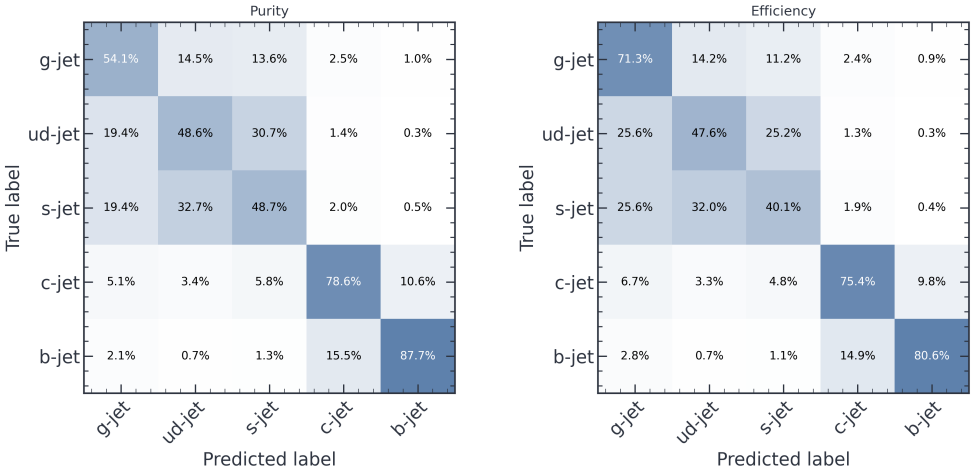


Figure 3: Accuracies over epochs during the training. The solid and broken lines are accuracies for the training and the validation respectively.



(a) Purity **(b)** Efficiency

Figure 4: Confusion matrix between true labels and predicted labels, (a) and (b) show a value of purity and efficiency respectively. The purity means a ratio of true positive labels and true labels, the efficiency stands for

Regarding the separation of u/d-jet and s-jet, the s-jet decays to K-hadrons and has a relatively long lifetime, so it would be better if this feature could be exploited more strongly. If the classification of the jets were random, the accuracy of each would be about 20%, but in this case, the accuracy of u/d-jet and s-jet is approximately 40-50% each, so we can conclude that the classification is working well.

Fig. 5 shows the Receiver Operating Characteristic (ROC) curve for all labels. The ROC curves in Fig. 5 are computed with each label as true positive and the sum of the other labels as false positive. As can be seen in Fig 5, b-jet tagging performance is the best, while s-jet tagging

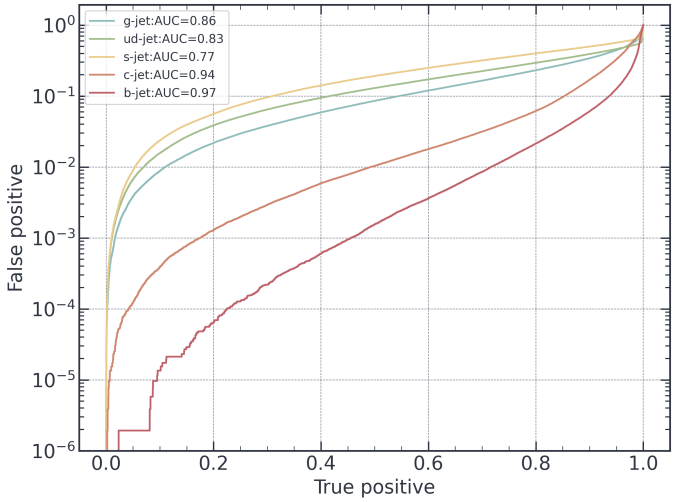
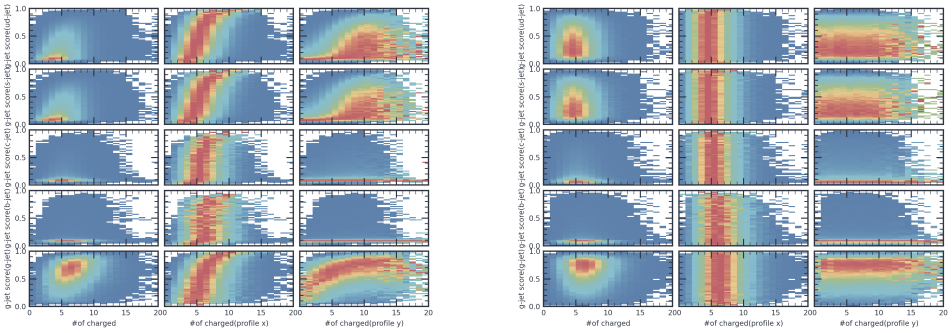


Figure 5: ROC curve for all labels against all other labels.

performance is the worst. The Area Under Curve (AUC) is calculated by an area under the ROC curve.



(a) Without FiLM layer **(b)** With FiLM layer

Figure 6: 2D histograms for the g-jet score and the number of charged tracks for without the FiLM layer (a) and with the FiLM layer (b) respectively. In each figure, for the row from top to bottom, u/d-jet, s-jet, c-jet, b-jet and g-jet are shown. For the column from left to right, a nominal 2D histogram, x-axis profiled histogram and y-axis profiled histogram are shown.

Fig. 6 shows a comparison of histograms showing the correlation between the g-jet scores and the number of charged tracks with and without the FiLM layer. If the FiLM is not applied, the score distribution is strongly correlated with jet transverse momentum, but once the FiLM is applied, there is little correlation, indicating that the FiLM layer is functioning. When the flavor tagging algorithm is applied to classification, strong dependence on jet characteristics such as momentum will result in strong bias from the distribution of the training samples, and the expected performance will not be achieved. Furthermore, if the distribution differs significantly between the real data and the training samples, the performance difference in the real data will be large, requiring a large

systematic error.

6. Conclusion

In this paper, jet-flavor tagging using the latest image recognition techniques is discussed. We proposed a model that classifies all jet flavors with little correlation with kinematic variables. We confirm that the FiLM layer reduces the jet kinematics correlation. While it is possible to simultaneously classify the flavors of jets containing gluons, we believe that additional efforts are needed to further improvements for the separation of b-jets from c-jets and u/d-jets from s-jets, respectively. We expect that a model that accounts for quark and gluon decay would improve the classification of individual flavors, which is a future subject.

References

[1] Lyndon Evans and Philip Bryant. LHC Machine. *JINST*, 3:S08001, 2008.

[2] Kim Albertsson et al. Machine Learning in High Energy Physics Community White Paper. *J. Phys. Conf. Ser.*, 1085(2):022008, 2018.

[3] Alexander Radovic, Mike Williams, David Rousseau, Michael Kagan, Daniele Bonacorsi, Alexander Himmel, Adam Aurisano, Kazuhiro Terao, and Taritree Wongjirad. Machine learning at the energy and intensity frontiers of particle physics. *Nature*, 560:41–48, 2018.

[4] Luke de Oliveira, Michael Kagan, Lester Mackey, Benjamin Nachman, and Ariel Schwartzman. Jet-images — deep learning edition. *JHEP*, 07:069, 2016.

[5] Hanxiao Liu, Zihang Dai, David R. So, and Quoc V. Le. Pay attention to mlps, 2021.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[7] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision, 2021.

[8] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision, 2021.

[9] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer, 2017.

[10] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.-S. Shao, T. Stelzer, P. Torrielli, and M. Zaro. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *Journal of High Energy Physics*, 2014(7), jul 2014.

[11] Torbjörn Sjöstrand, Stefan Ask, Jesper R. Christiansen, Richard Corke, Nishita Desai, Philip Ilten, Stephen Mrenna, Stefan Prestel, Christine O. Rasmussen, and Peter Z. Skands. An introduction to PYTHIA 8.2. *Computer Physics Communications*, 191:159–177, jun 2015.

[12] Michele Selvaggi. DELPHES 3: A modular framework for fast-simulation of generic collider experiments. *Journal of Physics: Conference Series*, 523:012033, jun 2014.

[13] Matteo Cacciari, Gavin P Salam, and Gregory Soyez. The anti-$k_t$jet clustering algorithm. *Journal of High Energy Physics*, 2008(04):063–063, apr 2008.

[14] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[15] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2016.

[16] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples, 2019.

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[18] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond, 2019.

[19] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2016.