PROCEEDINGS OF SCIENCE

A study of foundation models for event classification in collider physics

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

[a] *High Energy Accelerator Research Organization (KEK), Computing Research Center, 1-1, Oho, Tsukuba, Ibaraki, Japan*

[b] *International Center for Elementary Particle Physics (ICEPP), The University of Tokyo, 7-3-1, Hongo, Bunkyo, Tokyo, Japan*

E-mail: tomoe.kishimoto@kek.jp

This study aims to improve the performance of event classification in collider physics by introducing a foundation model based on deep learning. Event classification is a typical problem in collider physics, where the goal is to distinguish the signal events from the background events as much as possible in order to search for new phenomena in nature. A foundation model refers to a pre-trained model, which is usually trained on a large amount of unlabelled data, and then transferred to downstream tasks and fine-tuned. In applying this foundation model concept to collider physics, the following novelties are introduced in this study. First, the real particle collision data collected by the CMS experiment are used to train the foundation model. A self-supervised learning technique is introduced to process this unlabelled data. Second, data augmentation techniques based on physics knowledge are applied during the training process of the foundation model. This paper describes details of the self-supervised learning and data augmentation techniques used in this study, and shows the performance improvement in event classification by introducing this foundation model.

International Symposium on Grids and Clouds (ISGC2025)
16 -21 March 2025
Academia Sinica Grid Computing Centre (ASGC), Taipei, Taiwan

*Speaker

 ISSN 1824-8039 . Published by SISSA Medialab.

PoS(ISGC2025)001

1. Introduction

In collider physics, a significant number of events are produced from particle collisions using high-energy accelerators such as the Large Hadron Collider (LHC) [1]. Event classification is a typical problem in collider physics, where the goal is to distinguish the signal events from the background events as much as possible in order to search for new phenomena in nature. Although Deep Learning (DL) can provide significant discrimination power in this event classification by exploiting its large parameter space, a large amount of data is necessary to maximize its performance. The training data are typically generated using Monte Carlo (MC) simulations based on theoretical models of signal and background processes, as well as detailed detector geometries. Because there are many data analyses that target various signal events, such as Higgs boson measurements and new particle searches [2], generating sufficient training data using MC simulations is computationally expensive. To address this problem, we propose the use of a foundation model that enables an efficient training for the target event classification even with a small amount of data.

A foundation model refers to a pre-trained model, which is actively discussed in other fields such as natural language processing [3]. A foundation model is usually trained on a large amount of unlabelled data, and then transferred to downstream tasks and fine-tuned. It can be assumed that the downstream tasks will achieve an efficient training even with a small amount of data if common features are involved. By applying this foundation model concept to the event classification problem, we can reduce the computational cost associated with generating the training data using MC simulation. We have developed a self-supervised learning approach using real particle collision data and data augmentation techniques to build a foundation model for event classification. This paper describes the details of the self-supervised learning and data augmentation techniques employed in constructing the foundation model and demonstrates their effectiveness in improving event classification performance. The remainder of this paper is organized as follows: Section 2 describes the related work, including the novelties of this study. Section 3 summarizes the datasets used in the study. Section 4 provides details of the DL model. Section 5 presents the proposed pre-training strategy and the experimental results. Finally, Section 6 summarizes the findings of the study.

2. Related work

DL has been successfully adapted to event classification in collider physics, and has outperformed traditional machine learning methods, such as Boosted Decision Tree [4]. A previous study has also reported that DL models can provide discriminative power for different signal events by applying the transfer learning technique [5]. This study on the transferability of DL models has indicated that the effectiveness of transfer learning strongly depends on the similarity between the source and target event types. For example, transfer learning between different Higgs-related events results in significant performance gains, while transfer learning from Higgs events to searches for new particles shows limited effectiveness. These findings lead to the need for a well-generalized foundation model that can be applied to many types of events. The following novel contributions are made in this study to build such a foundation model:

(1) The real particle collision data collected by the CMS experiments [6] are used to train the foundation model. The advantages of using real data are as follows: First, there is no need

to generate a large amount of training data using MC simulations, which saves computing resources. Second, the potential bias introduced by the arbitrary selection of physics process for MC simulation is mitigated. It can be assumed that the foundation model is optimized for the selected physics event if the MC simulation data are used. This bias is mitigated by using the real data because many physics events are included in the real data. This will ensure the transferability of foundation models for many data analyses.

(2) Data augmentation techniques based on physics knowledge are applied during the training of foundation models. Data augmentation is a well-established method for expanding datasets in the image processing field [7]. Data augmentation plays an important role because the amount of data needs to be increased as much as possible to build a robust foundation model. The Lorentz transformations are applied to the events based on our physics knowledge to enhance data variations.

In a previous paper, we reported a preliminary study based on the first novelty, which is the use of real data [8]. In this preliminary study, only one type of event classification was evaluated because we focused on the proof-of-concept validation of pre-training strategy. In this updated study, we have extended the evaluation to four different event classification tasks, as described in the next section, to discuss the generalization capability of the foundation model. In addition, we have updated the training strategy of the foundation model as discussed in Section 5.

3. Datasets

There are two phases of training in this study: pre-training and event classification. In the pre-training phase, a foundation model is trained using the real data, as described earlier. This foundation model is then transferred to the event classification phase, where the model is fine-tuned using MC simulation data corresponding to the target physics processes. Both the pre-training and event classification phases utilize the CMS Open Data, which consists of proton-proton collision events collected at a center of mass energy $\sqrt{s}$= 13 TeV in 2016, along with the corresponding MC simulation samples. Table 1 summarizes the datasets and event selection criteria used in this study, where n^{particle} denotes the number of reconstructed particles (hereafter referred to as objects) in each event. Four types of signal sample are selected for the event classification phase. The background sample for all event classification tasks is the $t\bar{t}$+jets [9] process. The number of events allocated for training, validation, and testing for each dataset is 1.0×10^6, 1.0×10^5, and 1.0×10^5 events, respectively.

In this study, the input objects consist of lepton (electron and muon), τ lepton, light-jet, b-jet, and missing E_{T} (hereafter MET). The four-momenta (E, η, ϕ, mass), charge, and object type for each object are used as input variables. The ϕ is converted to ($\sin\phi$, $\cos\phi$) to handle the periodicity correctly. The object type is represented in a one-hot vector format: lepton, τ, light-jet, b-jet, or MET. Log transformation is applied to E and mass to fit the values within a reasonable range.

4. Deep learning model

Figure 1 shows an overview of the input data structure and the proposed DL model. The input data are prepared in a two-dimensional format of size $n \times m$, where n is the number of objects and

Table 1: Summary of datasets.

Phase		Data samples	Event selections
Pre-training		Collision data [10, 11]	$(n^{\text{lepton}} \geq 1) + (n^{\text{jet}} \geq 2) + (n^{\text{bjet}} \geq 1)$
Event Classification	(1)	H^+tb [12]	$(n^{\text{lepton}} \geq 1) + (n^{\text{jet}} \geq 4) + (n^{\text{bjet}} \geq 1)$
	(2)	H^+HW [13]	$(n^{\text{lepton}} \geq 1) + (n^{\tau} \geq 1) + (n^{\text{jet}} \geq 3) + (n^{\text{bjet}} \geq 1)$
	(3)	ttH [14]	$(n^{\text{lepton}} \geq 1) + (n^{\text{jet}} \geq 4) + (n^{\text{bjet}} \geq 2)$
	(4)	ttH [14]	$(n^{\text{lepton}} \geq 2) + (n^{\text{jet}} \geq 2) + (n^{\text{bjet}} \geq 1)$

m is the number of feature variables. For each event, the top four leptons, τ leptons, and b-jets are selected in order of p_{T}. The top eight light-jets are also selected in order of p_{T}, as light-jets are typically observed more frequently than the other object types. MET is treated as a single object, and thus the number of MET objects is always one. Zero-padding is applied to obtain a fixed length of input data when the number of objects for a given object type is less than the maximum. The feature variables for each object are (E, η, $\sin\phi$, $\cos\phi$, mass, charge, and one-hot vector for five object types). As a result, the number of objects (n) and the number of feature variables (m) are 21 and 11, respectively.

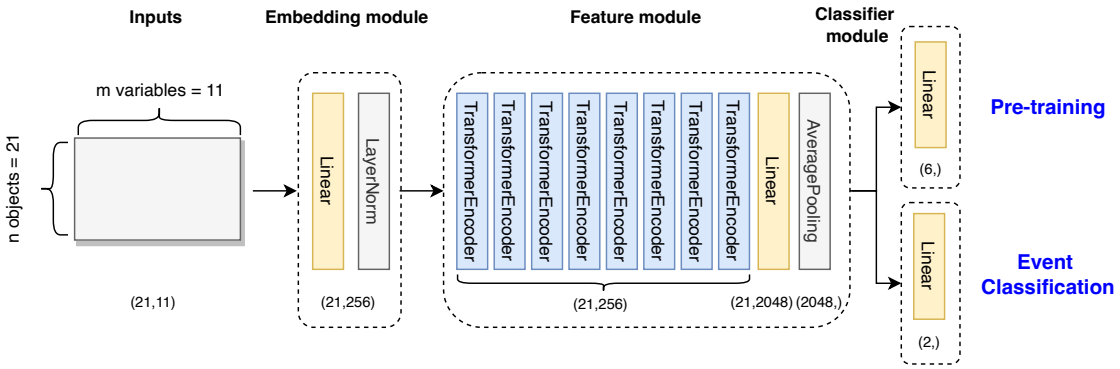


Figure 1: Overview of proposed DL model

The model consists of three modules: embedding, feature, and classifier modules. In the embedding module, the input feature variables for each object are embedded through a fully connected (linear) layer, and the outputs are then fed to the feature module. The transformer encoder layer [15] is employed in the feature module to exchange information among the objects. The zero-padded objects are masked in the transformer encoders. The feature module outputs are equivariant to permutations of the input object order, which is important since each event is represented as an unordered set of objects. The classifier module consists of a linear layer that outputs predictions depending on the training phase: pre-training or event classification. The total number of trainable parameters in the model is approximately 11 M.

5. Experiments

The code for this experiment was implemented using PyTorch [16]. The training settings were the same for both the pre-training and event classification phases. The cross-entropy loss function was used as the loss function. The SGD algorithm [17] was used as an optimizer, and the learning rate was decreased from 0.01 to 0.0001 by the cosine annealing algorithm [18]. The batch size was set to 512. The training was performed up to 500 epochs for pre-training and 100 epochs for event classification. The best epoch for the validation data was used as the final model parameters.

As described above, pre-training is performed using the real data. A self-supervised learning technique is employed to handle this unlabelled data. In this study, one object in each event is randomly replaced with another object of the same type from a different event during mini-batch preparation. The DL is then trained to predict the type of object that was replaced. Figure 2 illustrates an example of this object replacement procedure, where a lepton is replaced for prediction. The total number of label classes is six, including the case where no replacement is applied. In our previous study [8], the object types were randomly masked, and the DL model was trained to predict the masked object types using a multi-label classification approach. However, this method was not effective for handling leptons and MET since their types could be easily predicted from their mass information. In contrast, the new replacement-based approach achieves more robust training by requiring the DL model to learn the relationships among objects within the event.

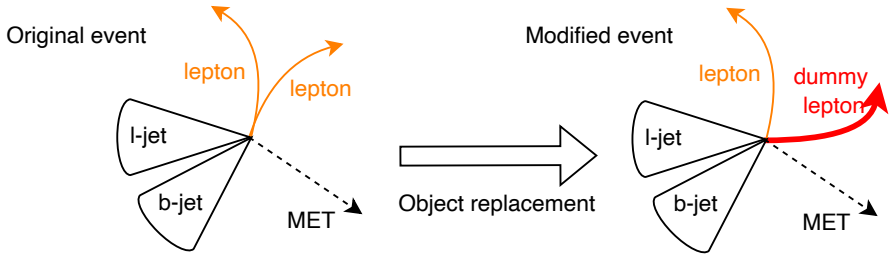


Figure 2: Illustration of the object replacement procedure used for self-supervised learning.

A data augmentation technique based on Lorentz transformations is applied during pre-training to enhance data variations. The DL model is expected to maintain consistent label predictions even after the Lorentz transformations are applied since the physics processes are invariant under the transformations. The following three types of Lorentz transformations are randomly applied to the events in this study:

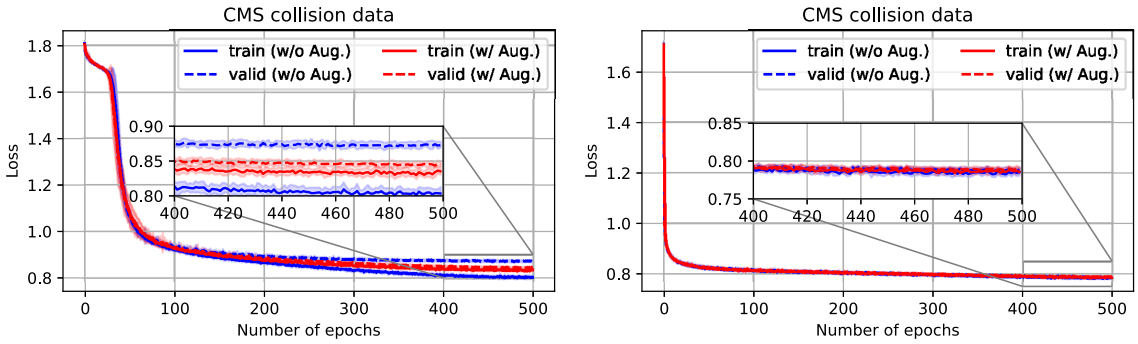
(1) Azimuthal rotation around the beam axis,

(2) Inversion of the beam-axis direction,

(3) Lorentz boost along the beam axis with β values ranging from 0 to 0.3.

Figure 3 shows the loss values during pre-training with and without data augmentation. By comparing the loss values of the training and validation datasets, it can be confirmed that data augmentation helps suppress overfitting when the number of available events is limited (Figure 3 (a)).

On the other hand, the effect of data augmentation becomes negligible when sufficient data are available (Figure 3 (b)).



(a) **(b)**

Figure 3: The loss values during pre-training with data augmentation (red lines) and without data augmentation (blue lines). The solid and dashed lines indicate the loss values for training and validation datasets, respectively. (a) 3.0×10^4 and (b) 1.0×10^6 events of the real data are used.

In the event classification phase, all input objects and feature variables without the replacement are used as a standard supervised classification problem. The weight parameters of the embedding and feature modules, which are obtained by pre-training, are used as the initial model parameters for event classification. These weight parameters are then fine-tuned during the event classification phase. The classifier module is trained from scratch because it is assumed that the embedding and feature modules extracted common knowledge, and the classifier module is highly dependent on the target task.

6. Results

Figure 4 shows the Area Under the Curve (AUC) values observed on the test datasets in event classification. The AUC values are shown for each signal dataset in terms of the number of events used for training in the event classification phase. The orange and green markers represent the AUC values with and without pre-training, respectively, where 1.0×10^6 events were used for pre-training. These results confirm that pre-training leads to significant improvements across all datasets, indicating that the pre-trained model generalizes well and is transferable to different types of analyses. The initial assumption was that transfer learning is particularly effective when the number of events is limited, whereas its performance gains would converge when a sufficient number of events are available. Figure 4 (b) supports this tendency; however, the other datasets show consistent improvements even when 1.0×10^6 events are available for event classification. Therefore, an important future task is to investigate whether the performance improvements in event classification converge as more data are used in datasets (1), (3), and (4).

Figure 5 shows the performance gains from pre-training in event classification, as a function of the number of events used in pre-training. The y-axis indicates the differences in AUC values with and without pre-training, that is ($\mathrm{AUC}^{\text{w/ pre-training}} - \mathrm{AUC}^{\text{w/o pre-training}}$). 3.0×10^4 events

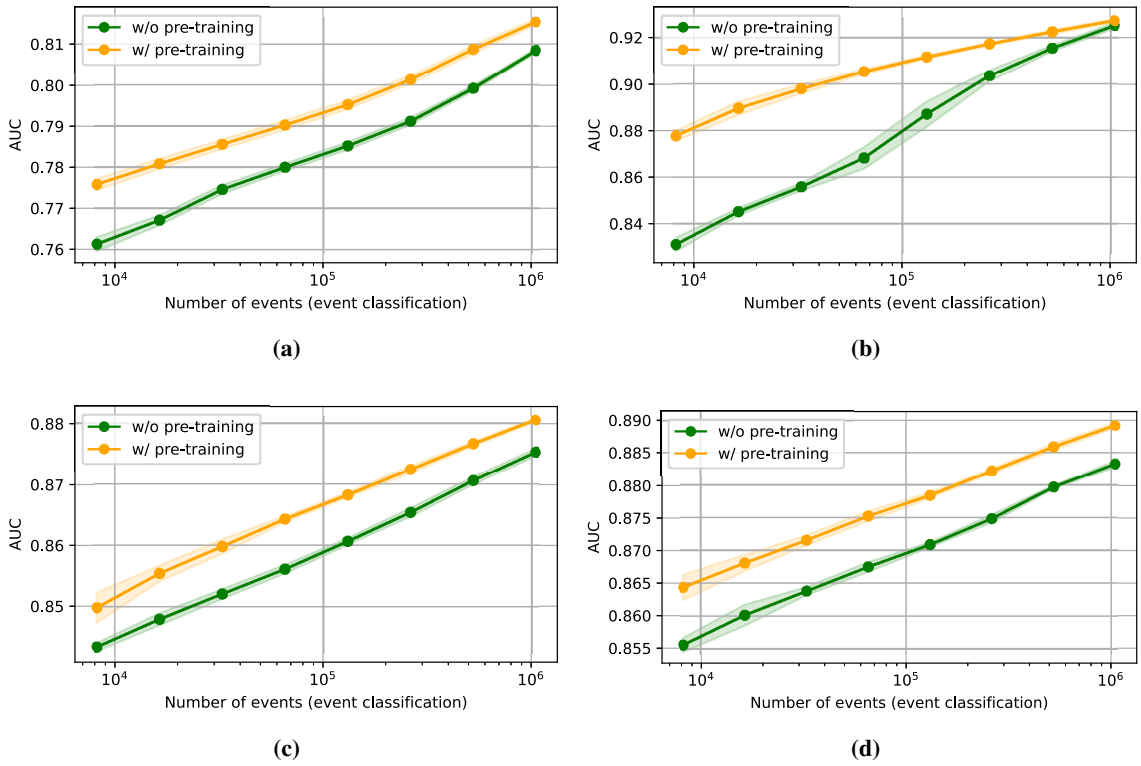


Figure 4: The AUC values in terms of the number of events in event classification. 1.0×10^6 events were used in pre-training. The error bands indicate one standard deviation of ten runs. The figures (a), (b), (c), and (d) indicate (1), (2), (3), and (4) datasets in Table 1, respectively.

were used in event classification. The red and blue markers indicate the values with and without data augmentation, respectively. The scalability of performance improvements with respect to the number of events used in pre-training is consistently observed across all datasets. Data augmentation results in higher average performance; however, the improvements remain within one standard deviation of ten independent runs.

7. Conclusion

In this study, the concept of foundation models was applied to event classification in collider physics. The proposed foundation model was successfully pre-trained on real data by using a self-supervised learning approach, in which an object within an event was replaced with one from another event for predictions. In addition, the data augmentation techniques based on the Lorentz transformations were applied during pre-training, which effectively suppressed overfitting when the number of available events was limited.

Our experiments confirmed that the AUC values in event classification are improved by introducing the pre-trained model across all four datasets. This finding demonstrates that the pre-trained model generalizes well across different types of analyses. Furthermore, the experimental results also indicate that improvements would be greater when more data are available in pre-training.

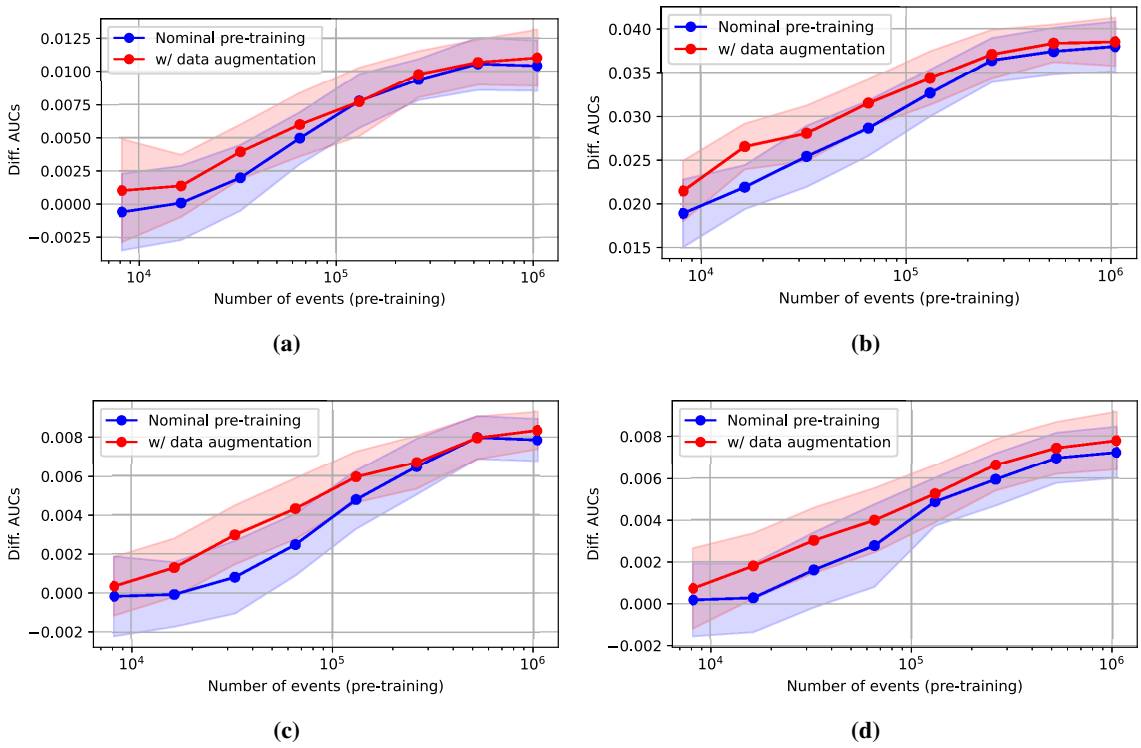


Figure 5: The improvements in AUC values in terms of the number of events in pre-training. The y-axis indicates the differences in AUC values with and without pre-training. The red and blue lines indicate the values with and without data augmentation, respectively. 3.0×10^4 events were used in event classification. The error bands indicate one standard deviation of ten runs. The figures (a), (b), (c), and (d) indicate (1), (2), (3), and (4) datasets in Table 1, respectively.

Investigating this scalability with larger models and more data remains an important direction for future work. This study may contribute to reducing the demand for computing resources for future collider experiments since the need for generating a large amount of training data by simulation can be eliminated.

References

[1] L. Evans and P. Bryant, *LHC Machine*, *Journal of Instrumentation* **3** (2008) S08001.

[2] "ATLAS Summary plots history." Available: https://atlaspo.cern.ch/public/summary_plots/, 2021.

[3] R. Bommasani, D.A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx et al., "On the opportunities and risks of foundation models." Available: https://crfm.stanford.edu/assets/report.pdf, 2021.

[4] P. Baldi, P. Sadowski and D. Whiteson, *Searching for Exotic Particles in High-Energy Physics with Deep Learning*, *Nature Commun.* **5** (2014) 4308 [1402.4735].

[5] T. Kishimoto, M. Morinaga, M. Saito and J. Tanaka, *Application of transfer learning to event classification in collider physics*, *PoS* **ISGC2022** (2022) 016.

[6] CMS collaboration, *The CMS Experiment at the CERN LHC*, *JINST* **3** (2008) S08004.

[7] C. Shorten and T.M. Khoshgoftaar, *A survey on image data augmentation for deep learning*, *J. Big Data* **6** (2019) 60.

[8] T. Kishimoto, M. Morinaga, M. Saito and J. Tanaka, *Pre-training strategy using real particle collision data for event classification in collider physics*, 2023.

[9] "CMS Collaboration (2024). Simulated dataset TTJets_TuneCP5_13TeV-amcatnloFXFX-pythia8 in NANOAODSIM format for 2016 collision data. CERN Open Data." Available: https://opendata.cern.ch/record/67731.

[10] "CMS Collaboration (2024). BTagCSV primary dataset in NANOAOD format from RunG of 2016 (/BTagCSV/Run2016G-UL2016_MiniAODv2_NanoAODv9-v1/NANOAOD). CERN Open Data." Available: https://opendata.cern.ch/record/30517.

[11] "CMS Collaboration (2024). BTagCSV primary dataset in NANOAOD format from RunH of 2016 (/BTagCSV/Run2016H-UL2016_MiniAODv2_NanoAODv9-v1/NANOAOD). CERN Open Data." Available: https://opendata.cern.ch/record/30550.

[12] "CMS Collaboration (2024). Simulated dataset ChargedHiggs_HplusTB_HplusToTB_M-300_TuneCP5_13TeV_amcatnlo_pythia8 in NANOAODSIM format for 2016 collision data. CERN Open Data." Available: https://opendata.cern.ch/record/33867.

[13] "CMS Collaboration (2024). Simulated dataset ChargedHiggs_HplusTB_HplusToHW_M-300_MH2_M-200_TuneCP5_13TeV_amcatnlo_pythia8 in NANOAODSIM format for 2016 collision data. CERN Open Data." Available: https://opendata.cern.ch/record/33833.

[14] "CMS Collaboration (2024). Simulated dataset ttHTobb_M125_TuneCP5_13TeV-powheg-pythia8 in NANOAODSIM format for 2016 collision data. CERN Open Data." Available: https://opendata.cern.ch/record/67645.

[15] PyTorch TransformerEncoderLayer. Available: https://pytorch.org/docs/stable/generated/torch.nn.TransformerEncoderLayer.html, 2021.

[16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan et al., *Pytorch: An imperative style, high-performance deep learning library*, in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett, eds., pp. 8024–8035, Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[17] S. Ruder, "An overview of gradient descent optimization algorithms." Available: https://arxiv.org/abs/1609.04747, 2016. 10.48550/ARXIV.1609.04747.

[18] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts." Available: https://arxiv.org/abs/1608.03983, 2016. 10.48550/ARXIV.1608.03983.