Pre-training strategy using real particle collision data for event classification in collider physics

Tomoe Kishimoto
Computing Research Center, High Energy Accelerator Research Organization, KEK
1-1 Oho, Tsukuba, Ibaraki, Japan
tomoe.kishimoto@kek.jp

Masahiro Morinaga
International Center for Elementary Particle Physics, ICEPP, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
morinaga@icepp.s.u-tokyo.ac.jp

Masahiko Saito
International Center for Elementary Particle Physics, ICEPP, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
saito@icepp.s.u-tokyo.ac.jp

Junichi Tanaka
International Center for Elementary Particle Physics, ICEPP, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
jtanaka@icepp.s.u-tokyo.ac.jp

Abstract

This study aims to improve the performance of event classification in collider physics by introducing a pre-training strategy. Event classification is a typical problem in collider physics, where the goal is to distinguish the signal events of interest from background events as much as possible to search for new phenomena in nature. A pre-training strategy with feasibility to efficiently train the target event classification using a small amount of training data has been proposed. Real particle collision data were used in the pre-training phase as a novelty, where a self-supervised learning technique to handle the unlabeled data was employed. The ability to use real data in the pre-training phase eliminates the need to generate a large amount of training data by simulation and mitigates bias in the choice of physics processes in the training data. Our experiments using CMS open data confirmed that high event classification performance can be achieved by introducing a pre-trained model. This pre-training strategy provides a potential approach to save computational resources for future collider experiments and introduces a foundation model for event classification.

1 Introduction

In collider physics, a significant number of events[1] are produced from particle collisions using high-energy accelerators such as the Large Hadron Collider (LHC) [1]. Event classification, which

[1]The term "event" corresponds to "image" in image classification.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

separates interesting signal events from background events, plays a crucial role in data analysis. Even though Deep Learning (DL) can provide significant discrimination power in this classification problem 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 signal and background process theories. Because there are many data analyses that target various signal events, such as Higgs boson measurements and new phenomena searches [2], preparing a large amount of training data using MC simulations for all analyses is computationally expensive. For example, next-generation LHC experiments require extensive computing resources for MC simulations [3]. Therefore, there is a need for a technique that maximizes DL performance even with a small amount of training data.

The transfer learning (TL) technique is a promising approach to reduce computational costs because it enables efficient training of target tasks even with a small amount of data [4]. The DL model comprises a stack of layers with nonlinear functions. The initial layers learn the local features of the data, and the subsequent layers learn the global features. This behavior indicates that the knowledge of the local features learned while solving a problem can be transferred and put to use to solve a different set of problems that involve the common local features. This is how the TL technique functions and helps reduce computational demand. In the present analysis workflow for collider physics, dedicated DL models for each data analysis are trained from scratch, indicating that a large amount of training data is required for each data analysis. Thus, the demand on computing resources for generating the training data can be saved when a pre-trained model that can be transferred to many data analyses could be built.

This study presents a strategy for building a pre-trained model for event classification and performance improvement. The CMS open data [5] from real particle collisions (hereafter called real data) collected by the CMS experiment [6] at the LHC were used in pre-training by employing a self-supervised learning technique. The remainder of this paper is organized as follows: Section 2 describes the related work, including the advantages of using real data. Section 3 summarizes the datasets used in the study. Section 4 provides details of the DL model and the proposed pre-training strategy. Section 5 presents the experimental results. Finally, Section 6 summarizes the findings of the study.

2 Related work

DL has been successfully adapted for event classification in collider physics. A previous study has reported that DL outperforms traditional machine learning methods, such as Boosted Decision Trees, by discovering powerful features and providing better discrimination power [7]. Another study on transferability of DL models to different signal events has reported that DL provides discriminative power to other signals that vary kinematically [8]. Application of the TL technique for event classification using graph neural network architecture [9] was investigated and found that it enabled us to to examine the transferability between event classifications with different numbers of reconstructed particles [10]. In contrast to the aforementioned works, this study introduces a novel approach that utilizes real data for pre-training. 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 for pre-training, which saves computing resources.
- Second, choice of the physics process for the MC simulation that is applied for pre-training can be arbitrary because it is assumed that the pre-trained model will be optimized for the chosen physics process. The bias of this choice is mitigated by using real data because many physics processes are included in the real data. This will ensure the transferability of pre-trained model for many data analyses.

However, due to limited availability of true information in real data, in comparison to MC simulation data, an alternative training method based on a self-supervised technique was developed, as discussed in Section 4.1. Building a pre-trained model using self-supervision has been actively discussed in other fields, such as natural language processing, as a foundation model [11]. This study introduces a similar idea for data analysis in collider physics.

3 Datasets

As mentioned in the previous sections, there are two phases of training in this study: pre-training and event classification. Details of the datasets for each phase are described below.

Pre-training dataset: The single electron and single muon datasets in the CMS open data [12, 13], which are collision events at a center of mass energy $\sqrt{s}$ = 13 TeV in 2015, were used in the pre-training. The electron and muon are simply called lepton in this study. The following loose event selections were applied to the reconstructed particles (hereafter called objects[2]) in an event:

- There is at least one loose lepton with $p_{\mathrm{T}} > 10$ GeV.
- There are at least two b-jets with $p_{\mathrm{T}} > 20$ GeV.
- There are at least two light-jets with $p_{\mathrm{T}} > 20$ GeV.

These event selections were made with an aim to obtain events with a topology similar to that of the target event classification task. The selected events were split into approximately10^6, 10^5, and 10^5 events for the training, validation, and testing datasets, respectively.

Event classification dataset: The training data were produced using simulations: the collision events were generated by MadGraph5_aMC@NLO [15] at $\sqrt{s}$ = 13 TeV, showering and hadronization were performed by Pythia8 [16], and the detector response was simulated by Delphes [17]. Two-Higgs-doublet model [18], which introduces additional Higgs bosons, H^0, A and $H^{\pm}$, was used as the signal event. The top-pair production of the Standard Model was used as the background event. The observed objects per event were one lepton, one missing transverse momentum (MET) due to undetected neutrino, two b-jets, and two light-jets. Total events generated independently for the training, validation, test phases for each signal and background process were approximately 5×10^5, 5×10^4, and 5×10^4 events, respectively.

Six objects in an event were used in this study: a lepton, a MET, two b-jets, and two light-jets. The four-momenta of each object (p_{T}, η, ϕ, mass) and object-type were used as input variables. The ϕ was converted to ($\sin\phi$, $\cos\phi$) to handle the periodicity correctly. The object-type was represented in a one-hot vector format: lepton, MET, b-jet, or light-jet. Log transformation was applied to p_{T} and mass to fit the values within a reasonable range.

4 DL model

Figure 1 shows an overview of the input data and the proposed DL model. The input data were prepared in a two-dimensional format of $n \times m$, where n was the number of objects and m was the number of feature variables. The model consisted of three modules: embedding, feature, and classifier modules. In the embedding module, the input feature variables for each object are embedded in a fully connected (linear) layer, and the outputs are then fed to the feature module. The transformer encoder layer [19] is employed in the feature module to exchange information between the objects, which is influenced by the natural language processing field. The feature module outputs are equivariant to permutations of objects. The classifier module consists of a linear layer and outputs predictions depending on the phase: pre-training or event classification. The total number of trainable parameters was approximately 1.7 M.

4.1 Proposed pre-training strategy

As mentioned earlier, a self-supervised learning technique was employed in the pre-training phase to handle unlabeled real data. In the pre-training phase, the object-type (lepton, MET, b-jets, or light-jets) in the feature variables was randomly masked by zeros when preparing a mini-batch: the probability of an object-type getting masked is 0.65. It was possible that all object-types in an event were masked or not masked at all. The DL model was trained to predict the masked object-types as a multi-label classification problem. All input feature variables, including the object-type, were used in the target event classification phase. The weight parameters of the embedding and feature modules, which were obtained by pre-training, were used as the initial parameters for event classification. These weight parameters were fine-tuned during the event classification phase. The classifier module

[2]Collider physics terminologies are described in [14]

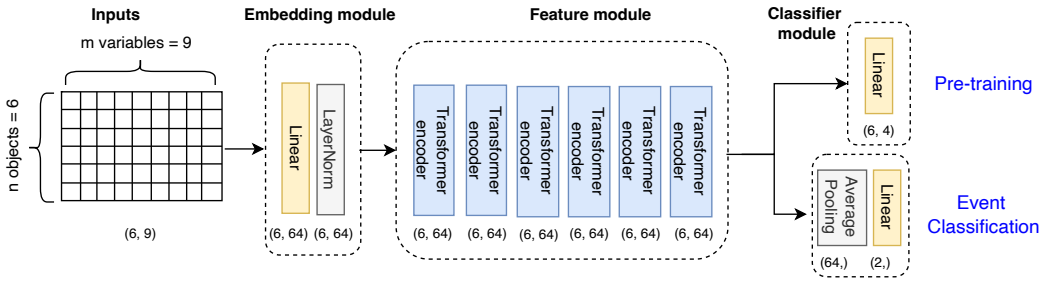


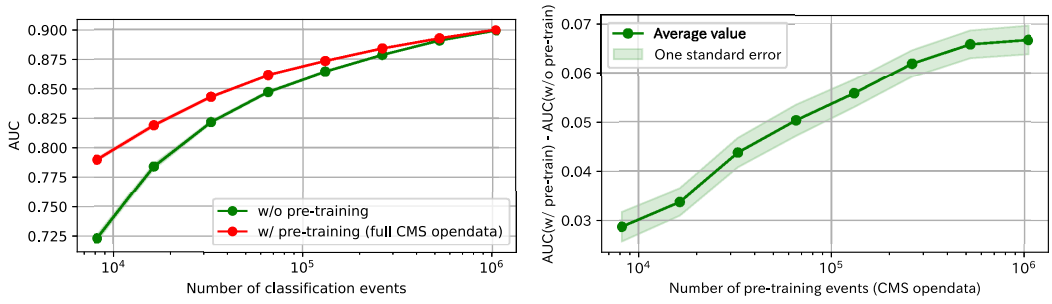
Figure 1: Overview of the proposed DL model. The numbers in the bracket indicate output shapes.

was trained from scratch because it was assumed that the embedding and feature modules extracted common knowledge, and that the classifier module was highly dependent on the target task.

5 Experiments

The model proposed in this experiment was implemented using PyTorch [20] and is available in [21]. The training settings were the same for the pre-training and event classification phases. The cross-entropy loss function was used as the loss function. Best epoch for the validation data is used as the final weight parameter after training for 100 epochs. The SGD algorithm [22] was used as an optimizer, and the learning rate was decreased from 0.01 to 0.0001 by the cosine annealing algorithm [23]. The batch size was set to 1,024. Other hyperparameters, such as the number of nodes and multi-heads in the transformer encoder layer, were optimized through a grid search using an event classification dataset without pre-training. All the executions used a local cluster of NVIDIA Tesla A100 cards, and training speed of approximately 90 batches/s was accomplished using an A100 card.

Figure 2 (a) shows the observed AUC values in the event classification task, with and without pre-training, that were obtained from the test dataset. The AUC values are shown in terms of the number of events used in the event classification phase, where all training events in the CMS open data were used for pre-training. Significant improvement in the performance could be evidenced by introducing pre-training when the number of events in the event classification was small (approximately 10^4 events). However, when the number of events increased to10^6 events and higher, it can be observed that the with and without pre-training performance curves converged. Furthermore, Figure 2 (b) shows the AUC values with respect to the number of events used in the pre-training phase, where approximately 10^4 events were used in the event classification phase. It can be observed that the improvements achieved by introducing the pre-training phase increase as the number of events in the pre-training phase increases.



(a) (b)

Figure 2: (a) AUC values of the event classification task in terms of the number of classification events. The red and green points show the values with and without pre-training. (b) Differences of AUC values with and without pre-training in terms of the number of pre-training events. The points and error bands in both figures indicate the average value and one standard error of 50 runs.

Following have been identified as limitations related to this study. First, the scaling behavior shown in Figure 2 (b) encourages a pre-training with a larger number of events; however, adding more events to the pre-training phase is difficult because the number of events available in the CMS open data itself is limited. Second, better optimization of the DL model and training strategy are required to achieve higher transfer learning efficiency. Furthermore, our interest lies in adapting the pre-trained model to different signal events to evaluate the generalization of the model. This will be the subject of future research.

6 Conclusion

In this study, a pre-training strategy that uses real data for event classification is discussed. The proposed model was successfully trained using real data by employing a self-supervised learning technique in which the object-types were masked for predictions. Our experiments confirmed that the AUC values in event classification could be improved by introducing a pre-trained model when the number of available events is small. Further, the experiments also indicated that improvements would be greater when more data were available in pre-training. In our experiment, event classification was performed using only one dataset. More physics processes must be investigated for event classification to improve the generalization of transferability, which is a subject for future research.

This work can potentially contribute to reducing the demand for computing resources for future collider experiments because the need for generating a large amount of training data by simulation can be eliminated. This study aims to solve the problem of pure fundamental science, and we do not expect our study to result in a negative social impact.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP22K14050.

References

[1] Lyndon Evans and Philip Bryant. LHC Machine. *Journal of Instrumentation*, Vol. 3, No. 08, pp. S08001–S08001, aug 2008.

[2] ATLAS Summary plots history. Available: `https://atlaspo.cern.ch/public/summary_plots/`, 2021. (Accessed: Sep. 6, 2023).

[3] ATLAS Collaboration. ATLAS Software and Computing HL-LHC Roadmap. Available: `http://cds.cern.ch/record/2802918`, Mar 2022. (Accessed: Sep. 6, 2023).

[4] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 10, pp. 1345–1359, 2010.

[5] Lassila-Perini, Kati, Lange, Clemens, Carrera Jarrin, Edgar, and Bellis, Matthew. Using cms open data in research - challenges and directions. *EPJ Web Conf.*, Vol. 251, p. 01004, 2021.

[6] S. Chatrchyan, et al. The CMS Experiment at the CERN LHC. *JINST*, Vol. 3, p. S08004, 2008.

[7] Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for Exotic Particles in High-Energy Physics with Deep Learning. *Nature Commun.*, Vol. 5, p. 4308, 2014.

[8] M. Crispim Romão, N. F. Castro, R. Pedro, and T. Vale. Transferability of deep learning models in searches for new physics at colliders. *Phys. Rev. D*, Vol. 101, p. 035042, Feb 2020.

[9] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. Available: https://arxiv.org/abs/1806.01261, 2018.

[10] Tomoe Kishimoto, Masahiro Morinaga, Masahiko Saito, and Junichi Tanaka. Application of transfer learning to event classification in collider physics. *PoS*, Vol. ISGC2022, p. 016, 2022.

[11] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. Available: https://crfm.stanford.edu/assets/report.pdf, 2021.

[12] CMS collaboration (2021). SingleElectron primary dataset in AOD format from RunD of 2015 (/SingleElectron/Run2015D-08Jun2016-v1/AOD). Available: `http://opendata.cern.ch/record/24103`, 2021. (Accessed: Sep. 6, 2023).

[13] CMS collaboration (2021). SingleMuon primary dataset in AOD format from RunD of 2015 (/SingleMuon/Run2015D-16Dec2015-v1/AOD). Available: `http://opendata.cern.ch/record/24102`, 2021. (Accessed: Sep. 6, 2023).

[14] The ATLAS Collaboration. The ATLAS experiment at the CERN large hadron collider. *Journal of Instrumentation*, Vol. 3, No. 08, pp. S08003–S08003, aug 2008.

[15] 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. *JHEP*, Vol. 07, p. 079, 2014.

[16] 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. *Comput. Phys. Commun.*, Vol. 191, pp. 159–177, 2015.

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

[18] G.C. Branco, P.M. Ferreira, L. Lavoura, M.N. Rebelo, Marc Sher, and Joao P. Silva. Theory and phenomenology of two-higgs-doublet models. *Physics Reports*, Vol. 516, No. 1, pp. 1–102, 2012. Theory and phenomenology of two-Higgs-doublet models.

[19] PyTorch TransformerEncoderLayer. Available: `https://pytorch.org/docs/stable/generated/torch.nn.TransformerEncoderLayer.html`, 2021. (Accessed: Sep. 6, 2023).

[20] 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*, pp. 8024–8035. Curran Associates, Inc., 2019.

[21] hepfoundation. Available: `https://github.com/ktomoe/hepfoundation/`, 2023. (Accessed: Oct. 30, 2023).

[22] Sebastian Ruder. An overview of gradient descent optimization algorithms. Available: https://arxiv.org/abs/1609.04747, 2016.

[23] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. Available: https://arxiv.org/abs/1608.03983, 2016.