# Multi-Jet Event classification with Convolutional neural network at Large Scale

**Jiwoong Kim[2], Chang-Seong Moon[2], Hokyeong Nam[2], Junghwan Goh[3], Dongsung Bae[3], Changhyun Yoo[3], Sungwon Kim[4], Tongil Kim[4], Hwidong Yoo[4], Soonwook Hwang[1], Kihyeon Cho[1], Jaegyoon Hahm[1], Hunjoo Myung[1], Minsik Kim[1], Taeyoung Hong[1]**

[1] Korea Institute of Science and Technology Information, 245, Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea
[2] Department of Physics, Kyungpook National University, 80 Daehakro, Bukgu, Daegu 41566, Republic of Korea
[3] Department of Physics, Kyung Hee University, 26 Kyungheedae-ro, Dongdaemun-gu, Seoul 02447, Republic of Korea
[4] Department of Physics, Yonsei University, 50 Yonsei-ro Seodaemun-gu, Seoul, 03722, Republic of Korea

E-mail: jiwoong.kim@cern.ch

**Abstract.** We present an application of Scalable Deep Learning to analyze simulation data of the LHC proton-proton collisions at 13 TeV. We built a Deep Learning model based on the Convolutional Neural Network (CNN) which utilizes detector responses as two-dimensional images reflecting the geometry of the Compact Muon Solenoid (CMS) detector. The model discriminates signal events of the R-parity violating Supersymmetry (RPV SUSY) from the background events with multiple jets due to the inelastic QCD scattering (QCD multi-jets). With the CNN model, we obtained x1.85 efficiency and x1.2 expected significance with respect to the traditional cut-based method. We demonstrated the scalability of the model at a Large Scale with the High-Performance Computing (HPC) resources at the Korea Institute of Science and Technology Information (KISTI) up to 1024 nodes.

## 1. Introduction

In the High Energy Physics (HEP) data analysis, there are successful applications of Machine Learning (ML) algorithms that significantly improve event classification performance compared to the traditional methods based on the expert's knowledge. ML algorithms such as the Boosted Decision Tree (BDT), Shallow Neural Network, or similar algorithms have been used in the HEP data analysis. Recently, the Deep Neural Network (DNN) or Deep Learning is widely adopted because it is able to be applied to data with complicated data structures such as images, videos, natural languages, or sensors. There are studies to apply DNNs in analyzing low-level information such as the position and momentum of particles that pass through the detector, which give higher efficiency to select signal events compared to the ML algorithms with conventional feature variables based on the physics knowledge [1, 2].

In order to train the DNN model at a reasonable time scale with the Big Data from the LHC and the future High-Luminosity LHC (HL-LHC), we studied the performance of the DNN from

the viewpoint of the physics performance but also the computing performance with available computing resources. In Korea, the largest High-Performance Computing (HPC) resources for scientific research – Nurion – [3] are provided by the supercomputing center at the Korea Institute of Sciences and Technology Information (KISTI). We studied scalability to train the model on the Nurion system by measuring speed-up factors by increasing the number of computing nodes.

## 2. Model

The major purpose of this study is to do binary classification between signal and background. The signal process is RPV gluino pair production. We assume the gluino mass equals 1.4 TeV. In this process, each gluino decays as follows: $\tilde{g} \to \tilde{t} \to tbs$. The top quark decays to a b-jet and W boson. Subsequently, the W boson decays hadronically, thus there are 10 jets in the final state. The significant number of SM backgrounds makes it difficult to search the RPV SUSY process. The major background in this analysis is QCD multi-jet process which also has almost 10 jets in the final state.

The proton-proton collision events with a center-of-mass energy of 13 TeV in the LHC ring are generated using the PYTHIA8 [4]. The detector simulation assuming the CMS detector [5] is performed using DELPHES 3.4 [6] with the default "CMS detector configuration". 330 000 RPV SUSY events and 20 000 000 QCD multi-jet events are generated. Furthermore, an average of 32 interactions per proton bunch crossing (pile-up) are considered. Before making images, a baseline selection is applied event-by-event to select the event of interest and increase training performance. We reference all the cut criteria and physics variables of baseline selections from the CMS cut-based analysis [7]. Finally, the signal efficiency obtained from the CNN method is compared with the efficiency from the cut-based method [7]. To obtain results of the cut-based method, the signal-region selected in [7] is used. Table 1 shows the number of selected events after passing the baseline selection and basic information of the dataset.

Table 1: Summary of dataset: The QCD multi-jet samples are generated in different scalar sum of transverse momentum ($H_T$) ranges. The number of selected events is the remaining events after passing the baseline selection. These events are used as inputs of DNN model. There are roughly 450 000 training images, 150 000 validation images, and 150 000 testing images. The QCD multi-jet sample with $H_T < 1000$ GeV are all excluded after the baseline selection.

| dataset | Cross section [pb] | Number of generated events | Number of selected events |
|---|---|---|---|
| RPV | 0.02530 | 330 599 | 294 762 |
| QCD ($H_T$ 1000-1500 GeV) | 1207 | 15 466 225 | 37 091 |
| QCD ($H_T$ 1500-2000 GeV) | 119.9 | 3 368 613 | 137 805 |
| QCD ($H_T$ 2000-Inf GeV) | 25.24 | 3 250 016 | 280 279 |

After the baseline selection, the information of final-state particles is interpreted as $224 \times 224$ or $64 \times 64$ pixel images corresponding to an azimuthal angle $|\phi| < 3.15$ and a pseudorapidity $|\eta| < 2.5$ which is defined as $\eta = -\ln[\tan(\frac{\theta}{2})]$. All sub-detectors of the CMS, i.e. tracker, electromagnetic calorimeter, and hadronic calorimeter are considered as 3 different channels likewise RGB colors.

We exploit symmetry when we build the neural network structure instead of building a typical 2D image classification model because the detector image has a circular symmetry according to the $\phi$ direction. Therefore, we focus on the padding method that works well with the circular symmetry of detector images. The CNN models incorporate the "zero-padding" process to maintain the size of the feature map by attaching zeros to the outline of the image. However, this can lead to loss of information along to $\eta$ direction because of the $\phi$ symmetry. Therefore,

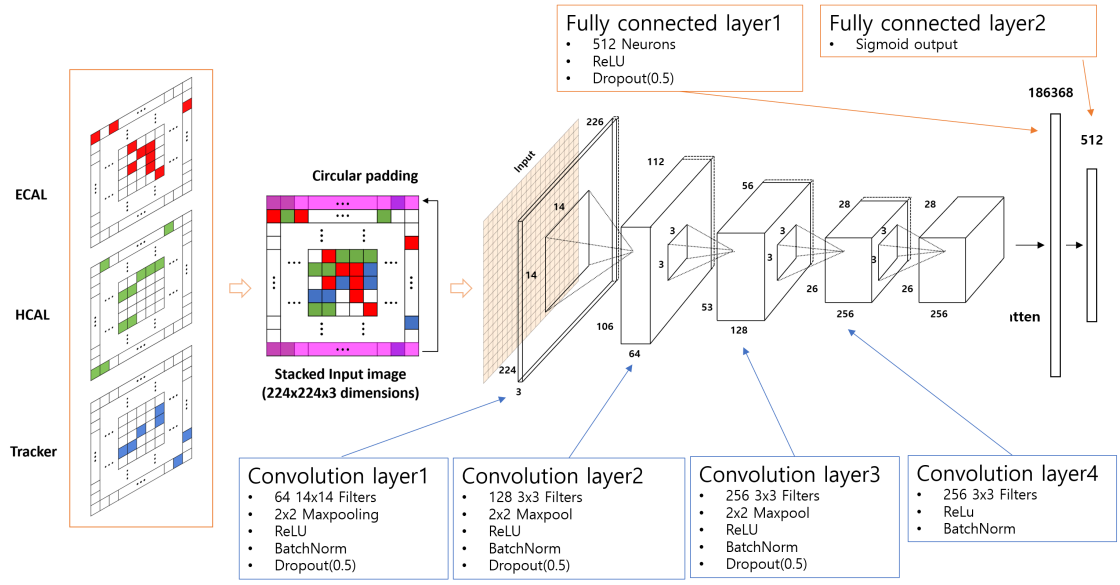


Figure 1: Illustration of the CNN architecture. Before the training step, image data are prepared by merging three sub-detector images: tracker, ECAL, and HCAL. The ECAL and HCAL images are weighted by energy measured in the detector, the tracker image is weighted by transverse momentum, and finally, these three images are merged as an image with 224x224x3 dimension

we adopt a new padding technique namely the "circular padding" wherein we still use the "zero-padding" in the $\eta$ direction, however, in the $\phi$ direction the left column is patched to the right column in the same order. The circular-padding scheme with the CNN architecture used to classify $224 \times 224$ pixel images is illustrated in Figure 1.

At the start of the training, two weights of the events are considered. First, is the normalization factor calculated as a cross-section divided by the number of generated events. This implies the number of expected events considering the probability of the production of a particular physical process and acceptance efficiency after the baseline selection. The second is the re-scale weight designed to balance the two input classes. After the normalization, background events are re-scaled to match the weighted sum of background events equal to the number of selected signal events.

We use the "Data parallelism" for the distributed training. In this method, the training model is copied to all the worker nodes, and the data are divided and distributed to all the nodes. When the training stage begins, the gradient of each node is calculated for each batch, and when the gradient calculations of all the nodes are completed, the model is updated with the average of all the gradients (Synchronous distributed training) [8]. Because "Data parallelism" and "Synchronous distributed training" are used, the overall batch size (effective batch size) is equal to batch size times the number of nodes. Therefore, we scale the training model conserving the overall batch size equals 32K (Strong scaling). The CNN model is trained using PyTorch[9] DNN library with Horovod[10] distributed training framework.

## 3. Results

We compare the classification performance between the CNN method and the cut-based method. The signal efficiency using the CNN method is 1.845 (1.803) times higher than that of using the cut-based method for $224 \times 224$ ($64 \times 64$) images. We obtain an average AUC of 0.9903 (0.9914) for the $224 \times 224$ ($64 \times 64$) pixel images. The left panel of Figure 2 shows the ROC curve of the

CNN method using 224 × 224 pixel image. In addition, we compare the expected significance from the cut-based method and the CNN method in the same background rejection point (False positive rate of 0.003503). The expected significance of the CNN method is 1.2 times higher than that of the cut-based method. Furthermore, we investigate the number of expected events in the discovery level ($5\sigma$) from the CNN results. There are 258 (259) signals and 2476 (2473) backgrounds, for 224 × 224 pixel images and 64 × 64 pixel images.

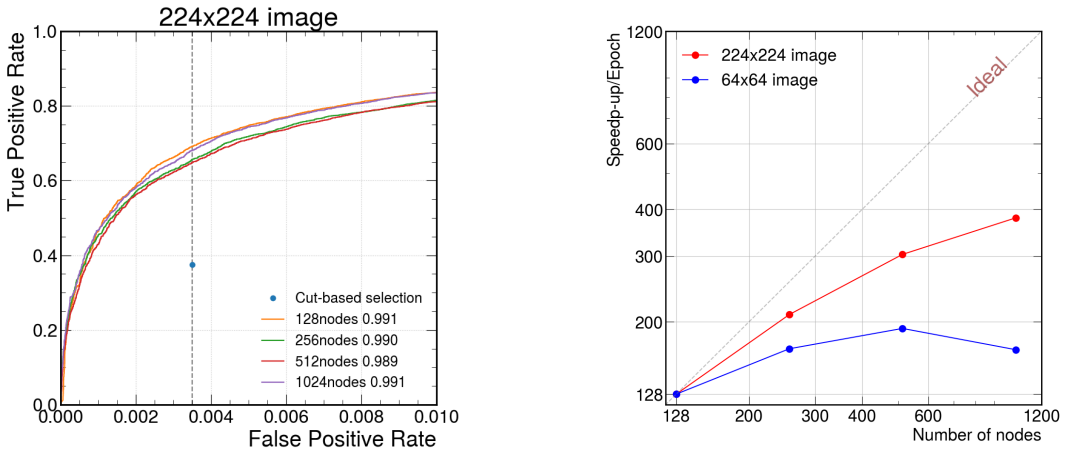


Figure 2: ROC curve of training results from 224 × 224 (left) and speedup according to the number of nodes (right). ROC curves of the CNN results with a different number of nodes are represented as solid lines. The True positive rate (TPR) point in the given False positive rate (FPR) from the cut-based method is represented as a blue point. Each AUC value according to the number of nodes is shown in the legend. The results of the CNN method with multi-nodes show higher signal efficiency (TPR) than that of the cut-based method. The training using 128 nodes shows the highest AUC. Also, each node shows a similar ROC curve, which means that performances of multi-node are stable. The scalability is checked using a strong scale and visualized in the right panel. The "speedup" (y-axis) is calculated as the training time of the number of nodes divided by that of the single node. The linear gray line in the plot is an ideal speedup. We check the scalability of our model up to 1024 nodes using 224 × 224 pixel images.

We demonstrate the scalability of our CNN model using 224 × 224 pixel images up to 1024 nodes in the Nurion system. The scalability is checked using two ways, i.e. training time and the "speedup" factor which is equal to the training time of N (N>1) nodes divided by that of a single node. As the number of nodes increases, the training time decreases, and the "speedup" factor increases. Figure 2 shows the ROC curve and speedup plot using strong scaling methods. Also, we compare the training time of 1024 KNL nodes with a single GPU machine (TITAN RTX). The training time per epoch using 1024 nodes is found to be 25 times faster than that observed using the GPU.

## 4. Conclusion

In this study, mainly, the CNN performance is compared with the conventional cut-based method. We demonstrate that the CNN overperforms the cut-based method in both signal efficiency and expected significance. Furthermore, we verify the scalability of our CNN model at the Nurion up to 1024 parallel "Xeon Phi" CPUs to speedup the training.

## Acknowledgments

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (Grants No. 2018R1A6A1A06024970, No. 2020R1A2C1012322 and Contract NRF-2008-00460). This research used resources of the National Supercomputing Center and the computing resources of the Global Science experimental Data hub Center (GSDC) in Korea Institute of Science and Technology Information (KISTI), along with supercomputing resources and technical support.

## References

[1] Baldi P, Sadowski P and Whiteson D 2014 *Nat. Commun.* **5** 1–9
[2] Baldi P, Sadowski P and Whiteson D 2015 *Phys. Rev. Lett.* **114** 111801
[3] Top500 2018 NURION - CRAY CS500, INTEL XEON PHI 7250 68C 1.4GHZ, INTEL OMNI-PATH `https://www.top500.org/system/179421/`
[4] Sjöstrand T, Ask S, Christiansen J R, Corke R, Desai N, Ilten P, Mrenna S, Prestel S, Rasmussen C O and Skands P Z 2015 *Comput. Phys. Commun.* **191** 159–177
[5] Adolphi R *et al. JINST volume=803, pages=S08004, year=2008, sorting=none*
[6] De Favereau J, Delaere C, Demin P, Giammanco A, Lemaitre V, Mertens A and Selvaggi M 2014 *JHEP* **2014** 57
[7] CMS collaboration 2016 Search for r-parity-violating susy in final states with zero or one lepton and large multiplicity of jets and b-tagged jets URL `https://cds.cern.ch/record/2205147`
[8] Goyal P, Dollár P, Girshick R, Noordhuis P, Wesolowski L, Kyrola A, Tulloch A, Jia Y and He K 2017 *arXiv preprint arXiv:1706.02677*
[9] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L *et al.* 2019 *Advances in neural information processing systems* pp 8026–8037
[10] Sergeev A and Del Balso M 2018 *arXiv preprint arXiv:1802.05799*