Cluster Counting Algorithm for the CEPC Drift Chamber using LSTM and DGCNN

Authors: Zhefei Tian, Guang Zhao, Linghui Wu, Zhenyu Zhang, Xiang Zhou, Shuiting Xin, Shuaiyi Liu, Gang Li, Mingyi Dong, Shengsen Sun, Guang Zhao, Zhenyu Zhang

Date: 2025-02-05T00:00:00+00:00

Abstract

Particle identification (PID) of hadrons plays a crucial role in particle physics experiments, especially for flavor physics and jet tagging. The cluster counting method, which measures the number of primary ionizations in gaseous detectors, represents a promising breakthrough in PID. However, developing an effective reconstruction algorithm for cluster counting remains a major challenge. In this study, we address this challenge by proposing a cluster counting algorithm based on long short-term memory and dynamic graph convolutional neural networks for the CEPC drift chamber. Leveraging Monte Carlo simulated samples, our machine learning-based algorithm surpasses traditional methods. Specifically, it achieves a remarkable 10% improvement in K/pi separation for PID performance, which meets the necessary PID requirements for CEPC.

Full Text

Cluster Counting Algorithm for the CEPC Drift Chamber using LSTM and DGCNN

Zhe-Fei Tian,[1] Guang Zhao,[2,†] Ling-Hui Wu,[2] Zhen-Yu Zhang,[1,‡] Xiang Zhou,[1] Shui-Ting Xin,[2] Shuai-Yi Liu,[2] Gang Li,[2] Ming-Yi Dong,[2,3] and Sheng-Sen Sun[2,3]

[1]Hubei Nuclear Solid Physics Key Laboratory, School of Physics and Technology, Wuhan University, Wuhan 430072, China
[2]Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China
[3]University of Chinese Academy of Sciences, Beijing 100049, China

Particle identification (PID) of hadrons plays a crucial role in particle physics experiments, especially in flavor physics and jet tagging. The cluster-counting

method, which measures the number of primary ionizations in gaseous detectors, represents a promising breakthrough in PID. However, developing an effective reconstruction algorithm for cluster counting remains challenging. To address this challenge, we propose a cluster-counting algorithm based on long short-term memory (LSTM) and dynamic graph convolutional neural networks (DGCNN) for the CEPC drift chamber. Experiments on Monte Carlo simulated samples demonstrate that our machine-learning-based algorithm surpasses traditional methods, improving K/π separation by 10% and meeting the PID requirements of CEPC.

Keywords: Particle identification, Cluster counting, Machine learning, Drift chamber

INTRODUCTION

The Circular Electron Positron Collider (CEPC) [1, 2] is a large-scale collider facility proposed in 2012 following the discovery of the Higgs boson. With a circumference of 100 km and two interaction points, it can operate at multiple center-of-mass energies: as a Higgs factory at 240 GeV [3–6], for W^+W^- threshold scans at 160 GeV, and as a Z factory at 91 GeV [7, 8]. Furthermore, it can be upgraded to 360 GeV for $t\bar{t}$ threshold scans, and in the future, to a proton-proton collider enabling direct exploration of new physics at approximately 100 TeV [9, 10]. The primary scientific objective of CEPC is to precisely measure Higgs properties, particularly their coupling strengths. Additionally, trillions of $Z \rightarrow q\bar{q}$ events produced by CEPC offer excellent opportunities to study flavor physics [11, 12].

Particle identification (PID) of hadrons is crucial in high-energy physics experiments, especially for flavor physics and jet tagging [13]. PID helps suppress combinatorial backgrounds, distinguish final states with identical topology, and provide valuable additional information for jet flavor tagging. Future experiments such as CEPC require advanced detector techniques with PID performance surpassing current capabilities.

The drift chamber is a key detector component in high-energy physics experiments. Beyond charged particle tracking, it can provide excellent PID with minimal additional detector budget. PID in drift chambers relies on the ionization behavior of charged particles traversing the working gas. A well-established technique measures the average ionization energy loss per unit length (dE/dx) of charged particles [14]. In a drift chamber cell, charged particles ionize the gas, creating electron cascades detectable as primary signals. This primary ionization follows a Poisson process, while secondary ionization produces a Landau distribution in dE/dx. The Landau distribution's infinitely long tail and large fluctuations limit dE/dx resolution [15]. [Figure 1: see original paper] shows an example signal waveform in a drift-chamber cell.

Alternatively, the cluster-counting technique directly measures the average number of primary ionizations per unit length in waveforms processed by fast electronics, rather than dE/dx, thereby reducing secondary ionization impact [16] and significantly improving PID performance. Cluster counting potentially improves resolution by a factor of two, making it the most promising PID breakthrough for future high-energy frontier colliders such as CEPC and the Future Circular Collider (FCC) [17]. A previous BESIII upgrade study demonstrated that cluster counting achieved superior PID performance compared to dE/dx, enhancing π/K separation power by approximately 1.7 times [18].

Reconstruction poses a significant challenge for cluster counting. An effective algorithm must efficiently and accurately determine the number of primary ionizations in a waveform. However, the stochastic nature of ionization processes and signal complexity create substantial obstacles. Traditional methods typically divide cluster counting into two stages: peak finding (detecting all peaks from primary and secondary ionizations) and clustering (determining the number of primary ionizations among detected peaks). Derivative-based peak finding computes first and second derivatives, identifying signals via threshold crossings, but often fails to achieve state-of-the-art performance, especially under high pile-up and noise. Time-based clusterization exploits the fact that average time differences between signals from different clusters tend to exceed those within the same cluster, but significant overlap in time difference distributions leads to low accuracy.

Machine learning (ML) offers a rapidly advancing approach using algorithms and statistical models to improve performance through data-driven learning. Neural networks, the most common ML technique, are computational models inspired by biological neural systems. Recurrent neural networks (RNNs) [19] and graph neural networks (GNNs) [20] are particularly popular. ML has been widely applied in high-energy and nuclear physics, such as the GNN-based ParticleNet algorithm for jet tagging [21, 22], and studies of QCD phase transitions [23, 24] and heavy-ion collisions [25–27]. For cluster counting, ML can leverage full waveform information to uncover hidden features in signal peaks, modeling the problem as a classification task amenable to mature tools like PyTorch [28] and PyTorch Geometric [29].

This paper presents an ML-based cluster-counting algorithm optimized for CEPC drift chambers. The remainder is organized as follows: Section II introduces the fast simulation method and samples used for training and testing. Section III details the ML-based algorithm. Section IV evaluates performance and compares it with traditional methods. Section V concludes.

II. DETECTOR, SIMULATION AND DATA SETS

A. The CEPC Drift Chamber

In the CEPC 4th conceptual detector design, a drift chamber is proposed between the silicon inner tracker (SIT) and silicon external tracker (SET). This chamber primarily provides PID capability while enhancing tracking and momentum measurements. Based on preliminary design, the chamber length is approximately 5800 mm, with radial extent from 600 to 1800 mm. The inner wall uses a carbon fiber cylinder, while outer support features a carbon fiber frame with eight longitudinal hollow beams and eight rings, sealed with a gas envelope. Aluminum endplates have a multistepped, tilted design to minimize wire tension deformation. [Figure 2: see original paper] shows a schematic of the drift chamber.

The entire chamber comprises approximately 67 layers. To meet PID and momentum measurement requirements, a cell size of 18 mm × 18 mm was adopted. Each cell contains a sense wire surrounded by eight field wires in a square configuration. Sense wires are 20 µm gold-plated tungsten, while field wires are 80 µm gold-plated aluminum. A gas mixture of 90% He and 10% iC_4H_{10} provides suitable primary ionization density.

B. Simulation and Data Sets

A sophisticated first-principles simulation package was developed for cluster counting. The package precisely simulates particle interactions and detector responses, creating realistic waveforms labeled with MC truth timing for supervised training. The package consists of simulation and digitization components. The drift-chamber cell geometry was constructed for simulation. Ionizations were generated using the Heed package. To reduce computational expense, electron transportation, amplification, and signal creation were parameterized according to Garfield++ simulation results, outputting analog waveforms [30]. Digitization incorporated data-driven electronic responses and noise. The preamplifier impulse response was measured experimentally and convolved with waveforms. Noise was extracted from experimental data using fast Fourier transform and added via inverse FFT. Digitization outputs realistic waveforms showing good agreement with experimental data in peak rise times and noise levels. [Figure 3: see original paper] shows the simulation flowchart.

Based on the CEPC 4th conceptual detector design and test beam experiments [31], waveforms exhibit approximately 4 ns single-pulse rise time, 5% noise level, and 1.5 GHz sampling rate. Using this package, MC samples with varying momenta were generated to train and test the neural network. summarizes the datasets.

. Summary of data sets used for training and testing ML-based cluster-counting algorithms.

Purpose	Algorithm	Particle	Number of Events	Momentum (GeV/c)
Training peak-finding	-	K±	5×10^5	$0.2 - 20.0$
Testing peak-finding	-	$\pi\pm$	5×10^5	$0.2 - 20.0$

Purpose	Algorithm	Particle	Number of Events	Momentum (GeV/c)
Training Clusterization	-	K±	5×10^5	$0.2 - 20.0$
Testing Clusterization	-	$\pi\pm$	$1 \times 10^5 \times 7$	5.0/7.5/10.0/12.5/15.0/17.5/20.0
Testing Clusterization	-	K±	$1 \times 10^5 \times 7$	5.0/7.5/10.0/12.5/15.0/17.5/20.0

III. METHODOLOGY

A. Algorithm Overview

An effective cluster-counting reconstruction algorithm must efficiently and accurately determine the number of primary ionizations in a waveform. As introduced in Section I, cluster counting typically decomposes into peak finding and clusterization. Peak finding detects signals from both primary and secondary ionizations, while clusterization discriminates primary ionizations among detected peaks. Traditional peak-finding uses waveform derivatives [32]. Ionization electron pulses, characterized by swift rise (nanoseconds) and prolonged decay (tens of nanoseconds), yield pronounced derivative values facilitating peak identification. Higher-order derivatives enhance hidden peak detection but increase noise susceptibility, requiring low-pass filtering beforehand. For clusterization, peak-merging algorithms exploit the proximity of electrons from single primary clusters in waveform time structure. However, overlap between inter-cluster and intra-cluster time differences necessitates precise merging criteria.

Traditional rule-based algorithms depend on incomplete raw hit information and human expertise, often failing to achieve state-of-the-art performance. In contrast, ML-based algorithms leverage abundant labeled samples for supervised learning, directly extracting intricate data features. Our approach uses an LSTM network for peak finding to discriminate signals from noise, detecting both primary and secondary ionization signals. The second step, clusterization, employs a dynamic graph neural network (DGCNN) to classify whether detected peaks originate from primary ionization.

B. Peak-finding

The peak-finding algorithm identifies all ionization peaks from waveforms. To reduce complexity, waveforms are divided into sliding windows of 15 data points. Each window receives a label based on MC truth information, identifying signal or noise candidates and defining peak finding as binary classification.

An LSTM-based network processes this time-series data. LSTM, a recurrent neural network variant, successfully handles sequential data across applications [33]. RNNs excel at sequence modeling tasks like prediction and labeling through dynamic contextual windows capturing entire sequence history. However, standard RNNs struggle with long sequences and suffer from vanishing/exploding gradients [34, 35].

LSTMs address these limitations through memory blocks containing cells and gates. Memory cells store temporal states via self-connections, while gates regulate information flow: input gates manage activations, output gates control output flow, and forget gates adaptively reset cell memory [35, 36].

The LSTM-based peak-finding architecture comprises: - **LSTM layer**: Processes sequential data, capturing long-term dependencies. Input features: 1; hidden state features: 32. - **Two linear layers**: Fully connected layers mapping 32$→32*and*32→$1, with sigmoid activation producing final classification.

[Figure 4: see original paper] illustrates this network structure. The model was trained on 5×10^5 π meson waveform events with momenta from 0.2–20 GeV/c, using batch size 64 for 50 epochs. Binary cross-entropy loss guided optimization, with Adam optimizer at initial learning rate 10^{-4}, reduced by factor 0.5 every 10 epochs. Optuna [39] optimized hyperparameters including learning rate and network size.

C. Clusterization

After LSTM-based peak finding detects all ionization peaks (primary and secondary), the clusterization algorithm determines the number of primary ionization peaks. Secondary ionization occurs locally relative to primary electrons with sufficient energy, causing electrons from single clusters to appear proximate in waveform time structure. This property enables algorithms distinguishing primary from secondary electrons. Traditional methods combine adjacent peaks based on timing.

GNNs, operating on graph-structured data, handle such complex information through pairwise message passing where nodes iteratively update representations by exchanging information with neighbors [40]. For cluster counting, peak timing serves as node features, with edges initially connected based on temporal similarity, enabling GNNs to learn complex temporal structures.

A DGCNN, specialized for learning local point cloud structure, was applied. The edge convolution layer dynamically computes graphs at each network layer, being differentiable and integrable into existing architectures. Peak timings from the first step form a graph: each peak's timing encodes node features, edge distances define temporal similarity, and nodes connect to k-nearest neighbors (k-NN) [41]. Through message passing, the network captures hidden local relationships for improved primary/secondary classification.

The clusterization architecture comprises: - **Three dynamic edge convolution layers**: Process graph-structured data by dynamically creating edges between nodes and neighbors, capturing local information via k-NN [42]. MLPs map input to output channels, with concatenated features from three layers producing 128-dimensional outputs (32+32+64). - **4-layer MLP**: Feedforward network with three hidden layers (256 neurons each) and one output layer (2 channels). Dropout rate 0.5 prevents overfitting. Log-softmax activation yields classification probabilities.

[Figure 5: see original paper] illustrates this architecture. The model trained on 5×10^5 pion waveform events (0.2–20 GeV/c) using batch size 128 for 100 epochs. Negative log-likelihood loss with Adam optimizer (initial learning rate 10^{-3}, reduced by 0.5 every 10 epochs) was used. Optuna tuned hyperparameters including MLP sizes and k-NN value (optimized to k=4).

IV. PERFORMANCE

The two-step model was trained via supervised learning on extensive waveform samples. Generalization performance was evaluated on test samples.

For peak finding, both LSTM-based and traditional second-derivative (D2) algorithms served as classifiers, evaluated using precision (purity) and recall (efficiency) metrics defined in Eq. (1):

$$\text{Purity} = \frac{TP}{TP + FP}, \quad \text{Efficiency} = \frac{TP}{TP + FN}$$

where TP is correctly detected peaks, (TP+FP) is total detected peaks, and (TP+FN) is total MC truth peaks. The LSTM algorithm was tested on 5×10^5 π sample events (0.2–20 GeV/c), evaluating purity and efficiency across probability thresholds.

[Figure 6: see original paper] shows purity and efficiency versus threshold. At threshold 0.95, the LSTM algorithm achieved purity 0.8986 and efficiency 0.8820. The D2 algorithm threshold was adjusted to match this purity, yielding efficiency 0.6827 (), demonstrating significantly higher LSTM efficiency, particularly for pile-up recovery ([Figure 7: see original paper]).

. Purity and efficiency comparison between LSTM-based and traditional D2 algorithms for peak-finding.

Algorithm	Purity	Efficiency
LSTM algorithm	0.8986	0.8820
D2 algorithm	0.8986	0.6827

After peak finding, clusterization determines primary cluster numbers. Implementing both LSTM peak-finding and DGCNN clusterization yields charged particle cluster-number distributions, enabling separation power calculations. Clusterization performs node classification in DGCNN. The classifier threshold was tuned to maximize K/π separation power, defined as:

$$\text{Separation Power} = \frac{\left| \left.\frac{dN}{dx}\right|_{\pi} - \left.\frac{dN}{dx}\right|_{K} \right|}{(\sigma_\pi + \sigma_K)/2}$$

where dN/dx and σ represent measured primary ionization density per unit length and uncertainties for π/K.

Optimization used K/π samples at fixed momenta (5.0, 7.5, 10.0, 12.5, 15.0, 17.5, 20.0 GeV/c). [Figure 9: see original paper] shows separation power versus threshold, with optimal overall performance at threshold 0.26.

[Figure 8: see original paper] compares cluster-number distributions from MC truth, traditional algorithm, and ML-based algorithm for 10 GeV/c pions. The ML-based distribution's mean closely matches MC truth, demonstrating higher efficiency. [Figure 9: see original paper] presents K/π separation power versus momentum for 1 m track length. The ML algorithm shows ~10% improvement across all momenta compared to traditional methods. Since separation power scales with √(track length), this corresponds to ~20% effective detector radius increase, significantly reducing cost. lists detailed results. Extrapolating 20 GeV/c K/π separation to various track lengths ([Figure 10: see original paper]) shows the CEPC design (600–1800 mm radius) meets the required 3σ K/π separation up to 20 GeV/c using ML reconstruction.

. Efficiency and separation power for charged K and π at various momenta for different algorithms. ML thresholds: 0.95 (LSTM peak-finding), 0.26 (DGCNN clusterization). Efficiency = reconstructed clusters / MC truth clusters.

Momentum (GeV/c)	Metric	ML-based algorithm	Traditional algorithm
5.0	$\pi\pm$ efficiency	0.882	0.683
	K$\pm$ efficiency	0.882	0.683
	K/π separation power	4.203	3.888
7.5	$\pi\pm$ efficiency	0.882	0.683
	K$\pm$ efficiency	0.882	0.683
	K/π separation power	4.279	3.954
10.0	$\pi\pm$ efficiency	0.882	0.683
	K$\pm$ efficiency	0.882	0.683
	K/π separation power	4.081	3.765
12.5	$\pi\pm$ efficiency	0.882	0.683
	K$\pm$ efficiency	0.882	0.683
	K/π separation power	3.832	3.550
15.0	$\pi\pm$ efficiency	0.882	0.683
	K$\pm$ efficiency	0.882	0.683
	K/π separation power	3.509	3.277
17.5	$\pi\pm$ efficiency	0.882	0.683
	K$\pm$ efficiency	0.882	0.683

Momentum (GeV/c)	Metric	ML-based algorithm	Traditional algorithm
	K/π separation power	3.216	3.054
20.0	$\pi\pm$ efficiency	0.882	0.683
	K$\pm$ efficiency	0.882	0.683
	K/π separation power	2.921	2.697

V. CONCLUSION

We developed a cluster-counting algorithm incorporating ML-based peak-finding and clusterization. Our peak-finding algorithm demonstrates superior efficiency over derivative-based methods. The clusterization algorithm produces Gaussian-distributed cluster numbers with efficiency approaching MC truth. The complete ML-based algorithm outperforms traditional methods by 10% in K/π separation power, equivalent to ~20% larger detector radius. This performance enables the current CEPC drift chamber design to meet PID requirements. The critical role of ML in cluster counting suggests strong potential for future high-energy physics experiments.

Data Availability

The data supporting this study are openly available in Science Data Bank at https://cstr.cn/31253.11.sciencedb.16322 and https://doi.org/10.57760/sciencedb.16322.

References

[1] The CEPC Study Group, CEPC Technical Design Report - Accelerator. (2023). doi: 10.48550/arXiv.2312.14363.
[2] The CEPC Study Group, CEPC conceptual design report: volume 2 - physics & detector. (2018). doi: 10.48550/arXiv.1811.10545.
[3] F.F. An, Y. Bai, C.H. Chen et al., Precision Higgs physics at the CEPC. Chin. Phys. C 43, 043002 (2019). doi: 10.1088/1674-1137/43/4/043002.
[4] D. Yu, M.Q. Ruan, V. Boudry et al., Higgs to $\tau\tau$ analysis in the future e^+e^- Higgs factories. (2019). doi: 10.48550/arXiv.1903.12327.
[5] Y. Bai, C.H. Chen, Y.Q. Fang et al., Measurements of decay branching fractions of $H \to b\bar{b}/c\bar{c}/gg$ in associated $(e^+e^-/\,^{+}\,^{-})$H production at the CEPC. Chin. Phys. C 44, 013001 (2020). doi: 10.1088/1674-1137/44/1/013001.
[6] Y.H. Tan, X. Shi, R. Kiuchi et al., Search for invisible decays of the Higgs boson produced at the CEPC. Chin. Phys. C 44, 123001 (2020). doi: 10.1088/1674-1137/abb4d8.
[7] P.X. Shen, P. Azzurri, C.X. Yu, M. Boonekamp et al., Data-taking strategy for the precise measurement of the W boson mass with a threshold scan at circular electron positron colliders. Eur. Phys. J. B 80, 66 (2020). doi: 10.1140/epjc/s10052-019-7602-x.
[8] Z.J. Liang, Electroweak physics at CEPC. Int. J. Mod. Phys. A 34, 1940013 (2019). doi: 10.1142/S0217751X1940013X.
[9] J. Gao, CEPC-SPPC accelerator status towards CDR. (2017). Int. J. Mod. Phys. A 32, 10.1142/S0217751X17460034.

[10] J. Gao, CEPC and SppC Status—From the completion of CDR towards TDR. Int. J. Mod. Phys. A 36, 2142005 (2021). doi: 10.1142/S0217751X21420057.
[11] T.F. Zheng, J. Xu, L. Cao et al., Analysis of Bc $\rightarrow \tau$ _τ at CEPC. Chin. Phys. C 45, 023001 (2021). doi: 10.1088/1674-1137/abcf1f.
[12] L.F. Li, M.Q. Ruan, Y.D. Wang et al., Analysis of Bs $\rightarrow \phi$ $\bar{\nu}$ at CEPC. Phys. Rev. D 105, 114036 (2022). doi: 10.1103/PhysRevD.105.114036.
[13] Y.F. Zhu, S.Z. Chen, H.H. Cui et al., Requirement analysis for dE/dx measurement and PID performance at the CEPC baseline detector. Nucl. Instrum. Meth. A 1047, 167835 (2023). doi: 10.1016/j.nima.2022.167835.
[14] G. Charpak, R. Bouclier, T. Bressani et al., The use of multiwire proportional counters to select and localize charged particles. Nucl. Instr. and Meth. 62, 262-268 (1968). doi: 10.1016/0029-554X(68)90371-6.
[15] W. Blum, W. Riegler, L. Rolandi, Particle Detection with Drift Chambers. (2008). doi: 10.1007/978-3-540-76684-1.
[16] A. H. Walenta, The Time Expansion Chamber and Single Ionization Cluster Measurement. IEEE Trans. Nucl. Sci. 26, 73-80 (1979). doi: 10.1109/TNS.1979.4329616.
[17] A. Abada, M. Abbrescia, S.S. AbdusSalam et al., FCC-ee: The Lepton Collider. Eur. Phys. J. Spec. Top. 228, 261-623 (2019). doi: 10.1140/epjst/e2019-900045-4.
[18] S.T. Xin, G. Zhao, L.H. Wu et al., Simulation study of particle identification using cluster counting technique for the BESIII drift chamber. J. Instrum. 18, T01006 (2023). doi: 10.1088/1748-0221/18/01/T01006.
[19] A. Sherstinsky, Fundamentals of recurrent neural network (RNN) and long short-term Memory (LSTM) network. Phys. D: Nonlinear Phenom. 404, 132306 (2020). doi: 10.1016/j.physd.2019.132306.
[20] J. Zhou, G. Cui, S.D. Hu et al., Graph neural networks: A review of methods and applications. AI Open 1, 57-81 (2020). doi: 10.1016/j.aiopen.2021.01.001.
[21] H.L. Qu, L. Gouskos, Jet tagging via particle clouds. Phys. Rev. D 101, 056019 (2020). doi: 10.1103/PhysRevD.101.056019.
[22] Y.F. Zhu, H. Liang, Y.X. Wang et al., ParticleNet and its application on CEPC jet flavor tagging. Eur. Phys. J. C 84, 152 (2024). doi: 10.1140/epjc/s10052-024-12475-5.
[23] Y.G. Ma, L.G. Pang, R. Wang et al., Phase Transition Study Meets Machine Learning. Chin. Phys. Lett. 40, 122101 (2023). doi: 10.1088/0256-307X/40/12/122101.
[24] F.P. Li, L.G. Pang, R. Wang et al., Application of machine learning to the study of QCD transition in heavy ion collisions. Nucl. Tech. 46, 040014 (2023). doi: 10.11889/j.0253-3219.2023.hjs.46.040014.
[25] W.B. He, Y.G. Ma, L.G. Pang et al., High energy nuclear physics meets machine learning. Nucl. Sci. Tech. 34, 88 (2023). doi: 10.1007/s41365-023-01233-z.
[26] W.B. He, Q.F. Li, Y.G. Ma et al., Machine learning in nuclear physics at low and intermediate energies. Sci. China-Phys. Mech. Astron. 66, 282001 (2023). doi: 10.1007/s11433-023-2158-3.

[27] Z.P. Gao, Q.F. Li, Studies on several problems in nuclear physics by using machine learning. Nucl. Tech. 46, 080009 (2023). doi: 10.11889/j.0253-3219.2023.hjs.46.080009.
[28] A. Paszke, S. Gross, F. Massa et al., PyTorch: An Imperative Style, High-Performance Deep Learning Library. Adv. Neural Inf. Process Syst. 32, 8024-8035 (2019). doi: 10.48550/arXiv.1912.01703.
[29] M. Fey, J.E. Lenssen, Fast graph representation learning with PyTorch Geometric. ICLR Workshop on Representation Learning on Graphs and Manifolds (2019). doi: 10.48550/arXiv.1903.02428.
[30] D. Pfeiffer, L. De Keukeleere, C. Azevedo et al., Interfacing Geant4, Garfield++ and Degrad for the simulation of gaseous detectors. Nucl. Instr. and Meth. A 935, 121-134 (2019). doi: 10.1016/j.nima.2019.04.110.
[31] C. Caputo, G. Chiarello, A. Corvaglia et al., Particle identification with the cluster counting technique for the IDEA drift chamber. Nucl. Instr. and Meth. A 1048, 167969 (2023). doi: 10.1016/j.nima.2022.167969.
[32] G. Zhao, L.H. Wu, F. Grancagnolo et al., Peak finding algorithm for cluster counting with domain adaptation. Comput. Phys. Commun. 300, 109208 (2024). doi: 10.1016/j.cpc.2024.109208.
[33] S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Comput. 9, 1735-1780 (1997). doi: 10.1162/neco.1997.9.8.1735.
[34] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. 5, 157-166 (1994). doi: 10.1109/72.279181.
[35] Y. Yu, X.S. Si, C.H. Hu et al., A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. 31, 1235-1270 (2019). doi: 10.1162/neco_a_{01199}.
[36] F.A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with LSTM. Neural Comput. 12, 2451-2471 (2000). doi: 10.1162/089976600300015015.
[37] J. Han, C. Moraga, The influence of the sigmoid function parameters on the speed of backpropagation learning. From Natural to Artificial Neural Computation (1995). doi: 10.1007/3-540-59497-3_{175}.
[38] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization. Proceedings of the 3rd International Conference on Learning Representations (2015). doi: 10.48550/arXiv.1412.6980.
[39] T. Akiba, S. Sano, T. Yanase et al., Optuna: A Next-generation Hyperparameter Optimization Framework. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2623-2631 (2019). doi: 10.48550/arXiv.1907.10902.
[40] J. Gilmer, S.S. Schoenholz, P.F. Riley, et al., Neural message passing for quantum chemistry. Proceedings of the 34th International Conference on Machine Learning 70, 1263-1272 (2017). doi: 10.48550/arXiv.1704.01212.
[41] Y. Wang, Y.B. Sun, Z.W. Liu et al., Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. 38 (2019). doi: 10.1145/3326362.
[42] Z.H. Zhang, Introduction to machine learning: k-nearest neighbors. Ann. Transl. Med. 4 (2016). doi: 10.21037/atm.2016.03.37.

[43] A. Pinkus et al., Approximation theory of the MLP model in neural networks. Acta Numer. 8, 143-195 (1999). doi: 10.1017/S0962492900002919.
[44] M. Hossin, M.N. Sulaiman, A review on evaluation metrics for data classification evaluations. Int. J. Data Min. Knowl. Manag. Process 5, 1 (2015). doi: 10.5121/ijdkp.2015.5201.

Note: Figure translations are in progress. See original paper for figures.