Nuclear Science and Techniques (2025) 36:113
https://doi.org/10.1007/s41365-025-01670-y

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] · Sheng-Sen Sun[2,3]

Received: 22 March 2024 / Revised: 4 June 2024 / Accepted: 10 June 2024 / Published online: 7 May 2025

Abstract
The 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, is 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 and dynamic graph convolutional neural networks for the CEPC drift chamber. Experiments on Monte Carlo simulated samples demonstrate that our machine learning-based algorithm surpasses traditional methods. It improves the K/π separation of PID by 10%, meeting the PID requirements of CEPC.

Keywords Particle identification · Cluster counting · Machine learning · Drift chamber

1 Introduction

The circular electron positron collider (CEPC) [1, 2] is a large-scale collider facility proposed in 2012 after the discovery of the Higgs boson. It has a circumference of 100 km and two interaction points, which allows it to operate at multiple center-of-mass energies. Specifically, it serves as a Higgs factory at 240 GeV [3–6], facilitates W^+W^- threshold scans at 160 GeV, and functions as a Z factory at 91 GeV [7, 8]. Furthermore, it can be upgraded to 360 GeV for a $t\bar{t}$ threshold scan. In the future, the CEPC can be upgraded to a proton-proton collider, enabling the direct exploration of new physics at a center-of-mass energy of approximately 100 TeV [9, 10]. The primary scientific objective of the CEPC is to precisely measure the Higgs properties, particularly their coupling properties. Additionally, trillions of $Z \to q\bar{q}$ events produced by the CEPC offer an excellent opportunity to study flavor physics [11, 12].

The particle identification (PID) of hadrons is crucial in high-energy physics experiments, especially in flavor physics and jet tagging [13]. Particle identification can help suppress combinatorial backgrounds, distinguish between the final states of the same topology, and provide valuable additional information for jet flavor tagging. Future particle physics experiments, such as CEPC, require advanced detector techniques with PID performances that surpass current techniques.

The drift chamber is a key detector in high-energy physical experiments. In addition to charged particle tracking, the drift chamber can also provide excellent PID while requiring almost no additional detector budget. In a drift chamber, PID is based on the ionization behavior of charged particles

This work was supported by National Natural Science Foundation of China (NSFC) (Nos. 12475200 and 12275296), Joint Fund of Research utilizing Large-Scale Scientific Facility of the NSFC and CAS (No. U2032114), Institute of High Energy Physics (Chinese Academy of Sciences) Innovative Project on Sciences and Technologies (Nos. E3545BU210 and E25456U210).

✉ Guang Zhao
zhaog@ihep.ac.cn

✉ Zhen-Yu Zhang
zhenyuzhang@whu.edu.cn

[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

traversing the working gas. A well-established technique for identifying particles is the measurement of average ionization energy loss per unit length (dE/dx) of charged particles [14]. In a drift chamber cell, charged particles ionize the gas, creating a cascade of electrons that can be detected as primary signals. This type of ionization is called primary ionization and is a Poisson process. Moreover, some of these electrons can cause secondary ionization, leading to a Landau distribution dE/dx. The Landau distribution has an infinitely long tail and large fluctuations that limit the dE/dx-resolution [15]. Figure 1 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 the waveforms processed by fast electronics, rather than dE/dx, which reduces the impact of secondary ionization [16] and significantly improves PID performance. The cluster counting technique has the potential to improve the resolution by a factor of two. Therefore, the cluster counting technique, which is the most promising breakthrough in PID, has been proposed for future colliders for high-energy frontiers, such as the CEPC and the Future Circular Collider (FCC) [17]. A previous study on cluster counting for the BESIII upgrade demonstrated that the cluster counting method exhibited superior PID performance compared with the dE/dx method. This significantly enhanced PID performance for π/K, achieving a separation power that is approximately 1.7 times that of the dE/dx-method [18].

Reconstruction poses a significant challenge in cluster counting. An effective reconstruction algorithm must efficiently and accurately determine the number of primary ionizations in a waveform. However, the stochastic nature of ionization processes and the complexity of signals present substantial obstacles to developing reliable cluster counting algorithms. In traditional methods, cluster counting algorithms are typically divided into two stages: peak finding (detecting all peaks from both primary and secondary ionizations) and clustering (determining the number of primary ionizations among the detected peaks in the previous step). For derivative-based peak finding, the first and second derivatives of the waveform are computed, and signals are identified via threshold crossings. Unfortunately, derivative-based algorithms often fail to achieve state-of-the-art performance, especially in scenarios with high pile-up and noise levels. In time-based clusterization, the average time differences between signals from different clusters tend to be larger than those within the same cluster. This information can be exploited to design peak-merging algorithms. However, due to the significant overlap in time difference distributions between inter-cluster and intra-cluster signals, these algorithms often suffer from low accuracy. Machine learning (ML) is a rapidly advancing field in computer science that uses algorithms and statistical models to enable systems to improve their performance by learning from data. Neural networks, the most commonly used ML techniques, are computational models loosely inspired by the human brain and consist of interconnected layers. Recurrent neural networks (RNNs) [19] and graph neural networks (GNNs) [20] are particularly popular types of neural networks. ML techniques have been widely applied in high-energy and nuclear physics. For instance, in high-energy physics, the GNN-based ParticleNet algorithm was developed for jet tagging [21], with applications to CEPC jet tagging [22]. In nuclear physics, ML techniques have been used to study phase transitions in nuclear matter governed by quantum chromodynamics (QCD) [23, 24], and to analyze heavy-ion collisions across various energy scales [25–27]. Machine learning has shown preliminary promise for handling large-scale data in high-energy physics. For cluster counting algorithms, ML can leverage full waveform

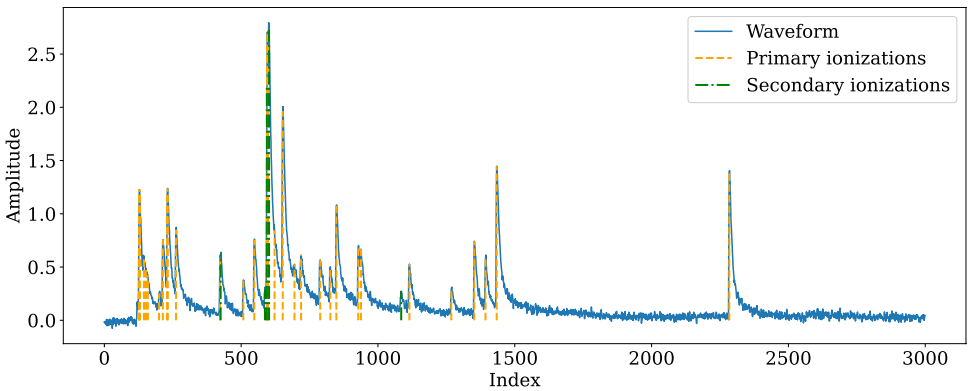

Fig. 1 A waveform example of induced current on a sense wire of a drift chamber. The x-axis represents the index of the waveform, which is sampled over a time window of 2000 ns at a sampling rate of 1.5 GHz. Both primary and secondary ionizations contribute to the waveform. The orange lines indicate peaks from primary ionizations. The green lines indicate peaks from secondary ionizations. An effective reconstruction algorithm needs to efficiently and accurately count the number of primary ionizations in the waveform. (Color figure online)

information and potentially uncover hidden features within the signal peaks. This problem can be modeled as a classification task, making it amenable to mature ML tools such as PyTorch [28] and PyTorch Geometric [29].

This paper presents an ML-based algorithm for cluster counting, optimized for a CEPC drift chamber. The remainder of the paper is organized as follows: Sect. 2.2 introduces the fast simulation method and the simulated samples used to train and test the ML-based algorithm. Section 3 details the ML-based cluster counting algorithm. Section 4 evaluates the performance of the ML-based algorithm and compares it with traditional methods. Section 5 concludes the paper.

2 Detector, simulation and datasets

2.1 The CEPC drift chamber

In the design of the CEPC 4th conceptual detector, a drift chamber is proposed to be inserted between the silicon inner tracker (SIT) and the silicon external tracker (SET). This chamber primarily provides PID capability and enhances tracking and momentum measurements.

Based on the preliminary design, the chamber length was approximately 5800 mm, with a radial extent ranging from 600 to 1800 mm. The inner wall consisted of a carbon fiber cylinder, while the outer support featured a carbon fiber frame structure comprising eight longitudinal hollow beams and eight rings. These components were sealed with a gas envelope. The aluminum endplates were designed with a multistepped and tilted shape to minimize deformation caused by wire tension. A schematic of the drift chamber is shown in Fig. 2.

The entire chamber comprises approximately 67 layers. To meet the requirements for PID capability and momentum measurements, a cell size of 18 mm × 18 mm was adopted. Each cell consists of a sense wire surrounded by eight field wires, forming a square configuration. The sense wires were 20 µm gold-plated tungsten wires, while the field wires were 80 µm gold-plated aluminum wires. To achieve a suitable primary ionization density, a gas mixture of 90% He and 10% iC_4H_{10} was proposed.

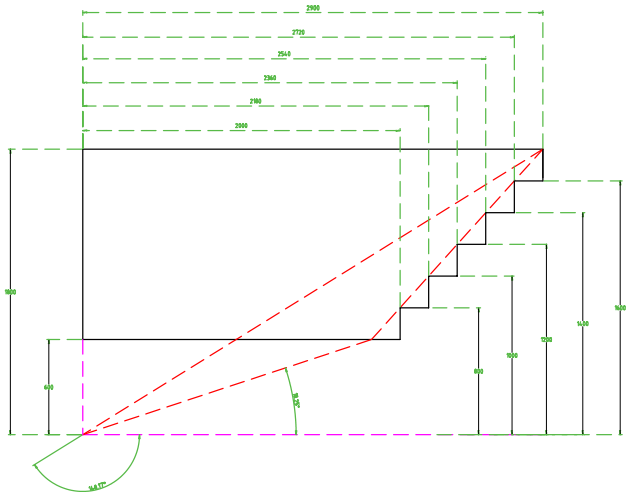

Fig. 2 Schematic layout of one-fourth of the CEPC drift chamber. The black lines show the boundaries of the drift chamber

2.2 Simulation and datasets

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, which enables supervised training. The simulation package consisted of two components: simulation and digitization. The geometry of the drift chamber cells was constructed for the simulation. Ionizations of charged particles were generated using the Heed package. To reduce computational expense, the transportation, amplification, and signal creation processes for each electron were parameterized according to the Garfield++ simulation results, which output analog waveforms for drift chamber cells [30]. Data-driven electronic responses and noise were considered in digitization. The impulse response of the preamplifier was measured experimentally and convoluted with the waveform. Noise was extracted from experimental data using the fast Fourier transform and added to the signal via the inverse fast Fourier transform. The digitization outputs realistic digitized waveforms that exhibit good agreement with experimental data in terms of peak rise times and noise levels. A flowchart of the simulation is presented in Fig. 3.

The simulation geometry is based on the design of the CEPC 4th conceptual detector. According to test beam experiments [31], the waveform exhibits a single-pulse rise time of approximately 4 ns, a noise level of 5%, and a sampling rate of 1.5 GHz. Using the simulation package, MC samples with varying momenta were generated to train and test the neural network algorithm. Detailed information about the samples is presented in Table 1.

3 Methodology

3.1 Algorithm overview

An effective reconstruction algorithm for cluster counting must efficiently and accurately determine the number of primary ionizations in a waveform. As introduced in Sect. 1, the cluster counting algorithm is typically decomposed into two steps: peak finding and clusterization. Peaks from both

Table 1 Summary of datasets used for training and testing ML-based cluster counting algorithms

Purpose	Algorithm	Particle	Number of events	Momentum (GeV/c)
Training	Peak finding	$\pi^{\pm}$	5×10^5	0.2–20.0
Testing	Peak finding	$\pi^{\pm}$	5×10^5	0.2–20.0
Training	Clusterization	$\pi^{\pm}$	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	$K^{\pm}$	$1 \times 10^5 \times 7$	5.0/7.5/10.0/12.5/15.0/17.5/20.0

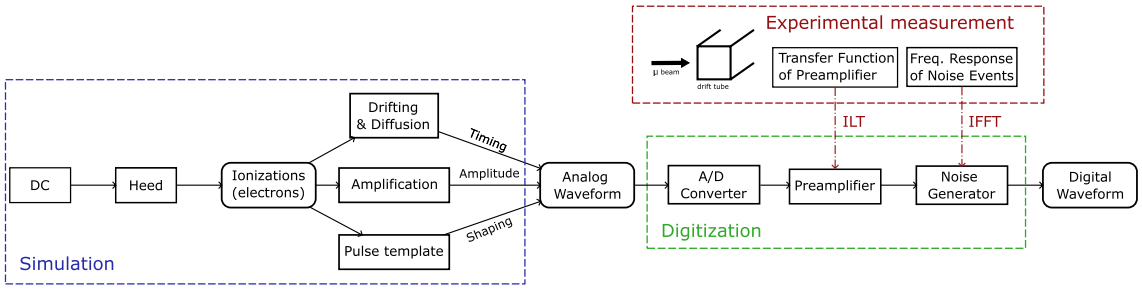

Fig. 3 Simulation package for cluster counting study. The package consists of simulation and digitization. The digitization takes input from the experimental measurement. (Color figure online)

primary and secondary ionizations were detected, while clusterization discriminates primary ionizations from the peaks detected in the previous step. The traditional peak finding algorithm uses the first and second derivatives of a waveform [32]. Ionization electron pulses, characterized by a swift rise (mere 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. To mitigate high noise levels, preprocessing with low-pass filters, such as moving averages, is recommended before applying derivatives. For clusterization, a peak-merging algorithm was used. Electrons from a single primary cluster, which are typically spatially localized, exhibit proximate arrival times at the sensing wire, forming discernible clusters in the waveform. Timing information from peak detection aids in distinguishing primary and secondary electron signals. Nonetheless, due to potential overlap between electrons from distinct primary clusters, a precise peak-merging requirement is crucial for the clusterization algorithm.

The aforementioned traditional rule-based algorithms, which depend on incomplete raw hit information and human expertise, often fail to achieve state-of-the-art performance. In contrast, ML-based algorithms harness an abundance of labeled samples for supervised learning, directly extracting intricate data features. In the first step of cluster counting, a long short-term memory (LSTM) network is employed to discriminate between signals and noise. Both the primary and secondary ionization signals are detected during this step. The second step of the algorithm, clusterization, is achieved using a dynamic graph neural network (DGCNN). The DGCNN is used to classify whether a detected peak in the first step originates from primary ionization.

3.2 Peak finding

The peak finding algorithm identifies all ionization peaks from a waveform. To reduce complexity, waveforms are divided into sliding windows with a window size of 15 data points. For each sliding window, a label is added based on MC truth information. Labels can identify a signal candidate or a noise candidate, defining peak finding as a binary classification.

To process time-series data in sliding windows, an LSTM-based network is explored for the peak finding algorithm. LSTM, a type of recurrent neural network (RNN), can process sequential data and has been successfully used in a range of applications [33]. RNNs are particularly effective for sequence modeling tasks, such as sequence prediction and labeling, because they utilize a dynamic contextual window that captures the entire history of the sequence. However, RNNs face limitations in processing long sequences effectively and are susceptible to issues related to vanishing and exploding gradients [34, 35].

The LSTMs have a unique architecture that includes memory blocks within a recurrent hidden layer. These memory blocks consist of memory cells and forget gates. Memory cells store the temporal state of the network through self-connections, while special multiplicative units, known as gates, regulate information flow. Each memory block includes an input gate to manage input activations in the memory cell, an output gate to control the output flow of

cell activations, and a forget gate to scale the internal state of the cell before adding it as an input through self-recurrent connections, thereby adaptively forgetting or resetting the cell's memory [35, 36].

The architecture of the LSTM-based peak finding algorithm is summarized as follows:

- An LSTM layer
  The LSTM layer is used for processing sequential data and capturing long-term dependencies between data points. This LSTM layer has one feature in the input data and 32 features in the hidden state.
- Two linear layers
  The neural network model consists of two linear layers that serve as fully connected layers. The first layer has an input size of 32 and an output size of 32. The second layer has an input size of 32 and an output size of 1. A sigmoid activation function [37] is applied to the output of the second layer to produce the final classification result.

Figure 4 illustrates the network structure of the LSTM-based model used to train the peak finding algorithm. The model was trained using a simulated sample of π mesons, consisting of 5×10^5 waveform events with momenta ranging from 0.2 to 20 GeV/c. After preprocessing, the data were divided into multiple batches, each with a batch size of 64, and the training process spanned 50 epochs.

Binary cross-entropy loss, a pivotal function for binary classification tasks, quantifies the discrepancy between true labels and predicted probabilities. This function effectively guides the model toward accurate predictions by handling cases where the output is a probability value between zero and one, making it particularly suited for our binary classification task. The Adam optimizer [38] was adopted, with an initial learning rate of 10^{-4}, which was reduced by a factor of 0.5 every 10 epochs. To further enhance algorithm performance, Optuna [39], a hyperparameter optimization framework, was employed to tune parameters such as the learning rate and network size.

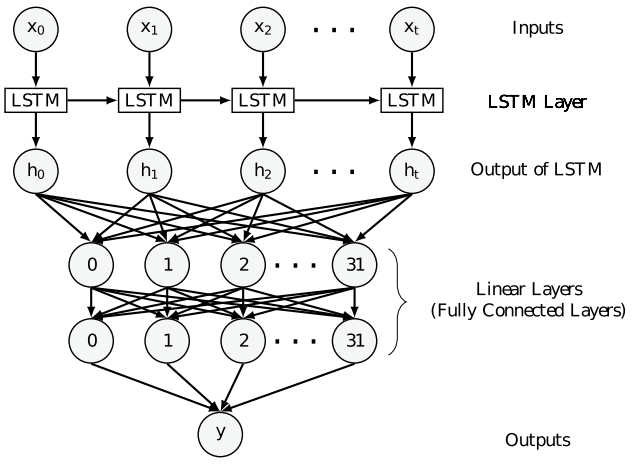

Fig. 4 The neural network structure of the LSTM-based model for the peak finding algorithm. (Color figure online)

3.3 Clusterization

After applying the LSTM-based peak finding algorithm, all ionization signal peaks, including both primary and secondary peaks, were detected. A second algorithm, termed the clusterization algorithm, was then developed to determine the number of primary ionization peaks.

In principle, secondary ionization occurs locally with respect to primary electrons if the primary electrons possess sufficient energy. This proximity causes electrons from a single cluster to appear close together in the waveform, a property that can be exploited to design algorithms for distinguishing between primary and secondary electrons. As mentioned in Sect. 1, traditional algorithms for this purpose rely on combining adjacent peaks.

GNNs, which operate on graph-structured data, are well-suited for handling complex information. The key feature of GNNs is pairwise message passing, where graph nodes iteratively update their representations by exchanging information with their neighbors [40]. For cluster counting, peak timing information is set as the node feature, while edges are initially connected based on timing similarities. GNNs can effectively learn the complex temporal structure of primary and secondary electrons through this message-passing mechanism.

A DGCNN, a specialized type of GNN, was applied to the clusterization algorithm. DGCNNs are designed to learn from the local structure of point clouds, enabling high-level tasks such as classification and segmentation. The edge convolution layer, a critical component of DGCNNs, dynamically computes graphs at each network layer. This layer is differentiable and can be integrated into existing architectures. In this study, the timings of peaks detected during peak finding were represented as a graph. Each peak's timing was encoded as a node feature, while edge distances were defined by the temporal similarity between nodes. Nodes were connected to their kth nearest neighbors (k-NN) [41]. During training, nodes updated their features and connections through message passing, enabling the network to capture hidden local relationships between peaks and achieve better performance in classifying primary and secondary ionizations.

The neural network architecture of the clusterization algorithm is summarized as follows:

- Three dynamic edge convolution layers

Three dynamic edge convolution layers process graph-structured data by dynamically creating edges between each node and its neighboring nodes, thereby capturing local information. A new graph is generated at each layer of the GNN based on the k-NN approach [42]. The multilayer perceptrons within the dynamic edge convolution layers map the number of input channels to the number of output channels. The features from three dynamic edge convolution layers were concatenated to get outputs with $32 + 32 + 64 = 128$ dimensions.

- A 4-layer multilayer perceptron (MLP)

  Multilayer perceptron (MLP) is a type of feedforward neural network that consists of multiple layers of neurons connected in a sequential manner [43]. This 4-layer MLP takes the concatenated output of the dynamic edge convolution layers as input. It has three hidden layers each with 256 neurons and 1 output layer with 2 channels. The dropout rate is set to 0.5, indicating that during training, each neuron in the network will have a 50% probability of being randomly dropped in order to prevent overfitting and encourage the network to learn more robust features. Finally, the model applies a log-softmax activation function to the output of the MLP and returns the classification probabilities.

Figure 5 illustrates the neural network architecture for clusterization. The model was trained using a pion sample containing 5×10^5 waveform events with momenta ranging from 0.2 to 20 GeV/c. After preprocessing, the data were divided into multiple batches, each with a batch size of 128, and training was conducted over 100 epochs.

For this binary classification model, the negative log-likelihood loss function and the Adam optimizer were adopted, with an initial learning rate of 10^{-3}, which was reduced by a factor of 0.5 every 10 epochs. Hyperparameters, including the sizes of the three MLPs in the dynamic edge convolution layers and the MLP serving as a fully connected layer, were tuned using Optuna. The value of k in k-NN, which determines how dynamic edge convolution layers establish relationships between nodes and their k nearest neighbors, was optimized to 4.

4 Performance

The two-step model was trained using supervised learning on a large number of waveform samples. To evaluate the model's generalization performance, it was applied to test samples.

For the peak finding algorithm, both the LSTM-based algorithm and a traditional second derivative-based (D2) algorithm served as classifiers. Their performance was evaluated using standard classifier metrics, including precision (purity) and recall (efficiency). Purity and efficiency are defined in terms of true positives (TP), false positives (FP), and false negatives (FN) [44], as shown in Eq. (1):

$$\begin{aligned} \text{Purity} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{Efficiency} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \end{aligned} \tag{1}$$

where TP is the number of correctly detected peaks, (TP+FP) is the total number of detected peaks, and (TP+FN) is the total number of peaks in MC truth of the waveform. The LSTM-based peak finding algorithm was tested on a π sample with momenta ranging from 0.2 to 20.0 GeV/c, consisting of 5×10^5 waveform events. Classifier purity and efficiency were evaluated by applying various probability thresholds. Figure 6 shows the purity and efficiency of the LSTM-based peak finding algorithm and the traditional D2 algorithm as functions of the threshold. For the LSTM-based algorithm, a threshold of 0.95 yielded a purity of 0.8986 and an efficiency of 0.8820. For the D2 algorithm, the threshold was adjusted to match the LSTM-based algorithm's purity, yielding an efficiency of 0.6827 (Table 2).This demonstrates that the LSTM-based algorithm is significantly more efficient than the D2 algorithm, particularly in recovering pile-up events (Fig. 7).

The clusterization algorithm was applied after peak finding to determine the number of primary clusters from the detected peaks. After implementing both the LSTM-based peak finding and DGCNN-based clusterization algorithms, the number of cluster distribution for a charged particle was obtained, enabling the calculation of separation power for different types of charged particles. In this study, clusterization was achieved by performing node classification in the DGCNN. To achieve optimal performance, the classifier threshold was tuned to

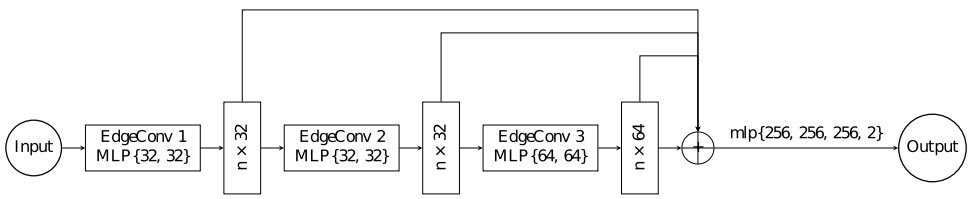

Fig. 5 The neural network structure of the DGCNN-based algorithm for clusterization. (Color figure online)

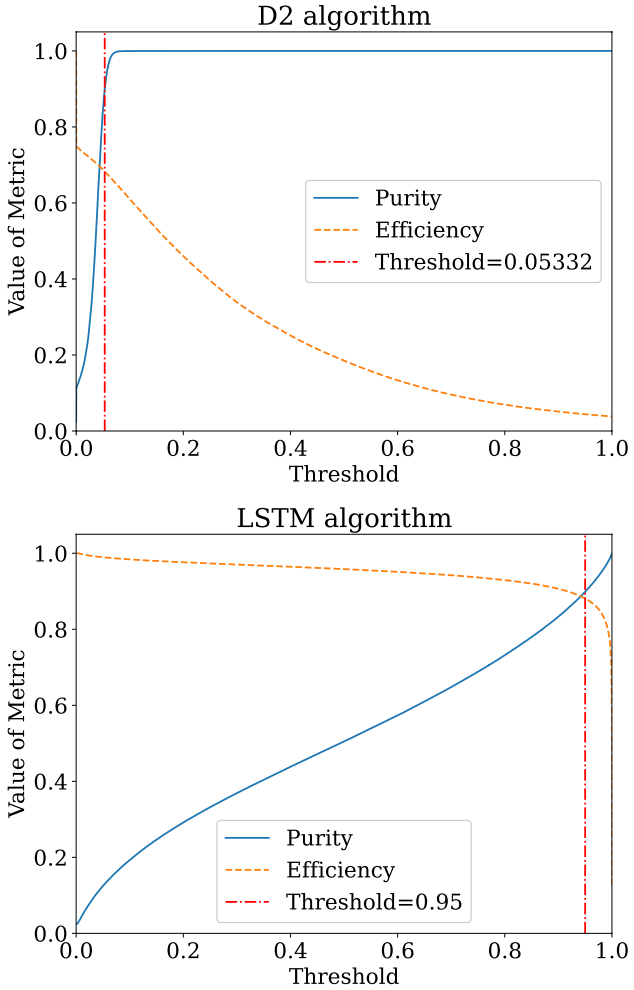

Fig. 6 Purity and efficiency as a function of the threshold for derivative-based D2 and LSTM-based algorithm, respectively. The blue solid line is the purity curve, the orange dashed line is the efficiency curve, and the red dash dotted line is the optimized threshold. The threshold for the D2 algorithm acts on the second derivative. While the threshold for the LSTM algorithm applies to the predicted probability of the neural network, with a range of [0, 1]. Any candidate that surpasses this threshold, either from D2 or LSTM algorithm, is considered as an ionization peak. (Color figure online)

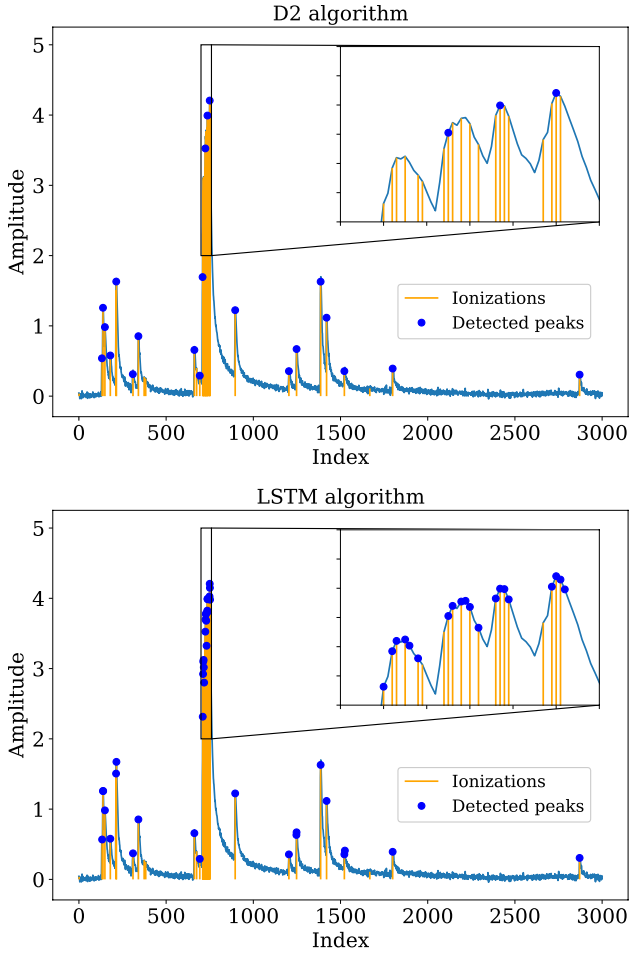

Fig. 7 Applying the derivative-based D2 and LSTM-based peak finding algorithms on a simulated waveform. The x-axis represents the index of the waveform, which is sampled over a time window of 2000 ns at a sampling rate of 1.5 GHz. The blue points are the detected peaks. The orange lines are the peaks from the MC truth. The zoomed figure shows that the LSTM-based algorithm detects the pile-up peaks more accurately and more efficiently than the D2 algorithm. (Color figure online)

Table 2 The purity and efficiency comparison between LSTM-based algorithm and traditional D2 algorithm for peak finding

	Purity	Efficiency
LSTM algorithm	0.8986	0.8820
D2 algorithm	0.8986	0.6827

maximize the K/π-separation power. The K/π-separation power is defined as

$$S = \frac{\left|\left(\frac{\mathrm{d}N}{\mathrm{d}x}\right)_{\pi} - \left(\frac{\mathrm{d}N}{\mathrm{d}x}\right)_{K}\right|}{(\sigma_{\pi} + \sigma_{K})/2}, \tag{2}$$

where $\mathrm{d}N/\mathrm{d}x_{\pi(K)}$ and $\sigma_{\pi(K)}$ represent the measured values and uncertainties in the number of primary ionizations per unit length for π (K). Optimization was performed using K/π samples with fixed momenta of $p = 5.0$ GeV/c, 7.5 GeV/c, 10.0 GeV/c, 12.5 GeV/c, 15.0 GeV/c, 17.5 GeV/c, and 20.0 GeV/c. The solid blue, dashed violet, and dashed cyan lines in Fig. 9 illustrate the K/π separation power for different thresholds. According to the optimization, the model achieves the best overall performance with a threshold of 0.26.

Using the optimized threshold, Fig. 8 compares the number of cluster distributions derived from the MC truth, the traditional algorithm, and the DGCNN-based algorithm. The mean value of the number of cluster distribution obtained

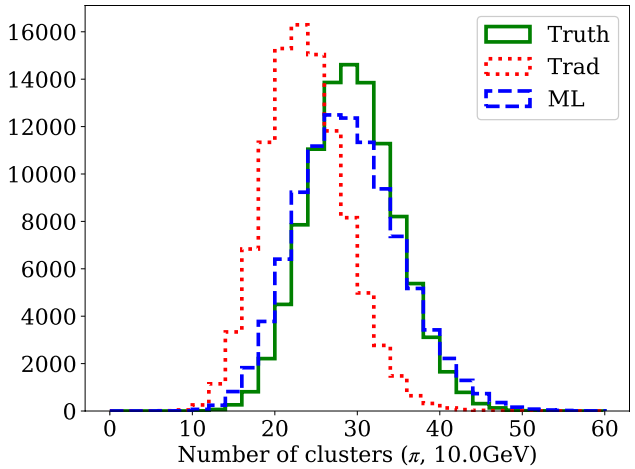

Fig. 8 The number of cluster distribution from MC truth (solid green), reconstruction by a traditional algorithm (dotted red) and reconstruction by an ML-based algorithm (dashed blue) for a 10 GeV/c pion sample. (Color figure online)

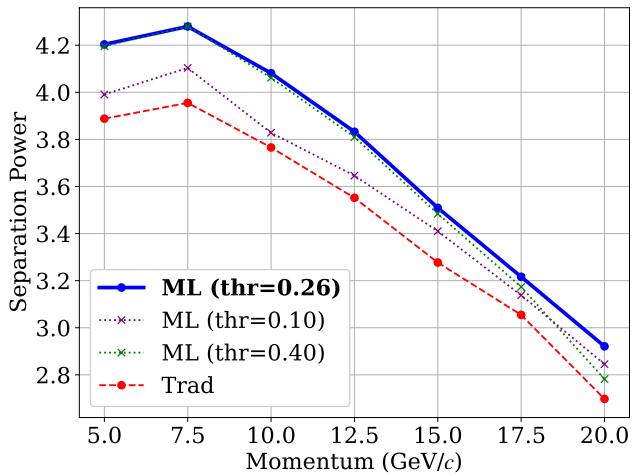

Fig. 9 The K/π separation power as a function of track momentum for a track length of 1 m. The red dashed line is from the traditional algorithm. The blue solid, violet dotted and green dotted lines are from the ML-based algorithm with thresholds of 0.26, 0.10, and 0.40. The blue solid line with a threshold of 0.26 achieves the overall best performance, which has a K/π separation that is approximately 10% better than that of the traditional algorithm. (Color figure online)

from the ML-based algorithm closely aligns with the MC truth, demonstrating that the ML-based algorithm achieves higher efficiency than traditional approaches. Figure 9 presents the K/π separation powers at various momenta for a track length of 1 m using different algorithms. The ML-based cluster counting algorithm shows approximately a 10% improvement in separation power across all momenta compared to traditional methods. Since separation power scales with the square root of the track length, this performance improvement corresponds to an effective increase of about 20% in the detector radius when using the traditional algorithm. This enhancement could significantly reduce the overall cost of the detector. Detailed numerical results are listed in Table 3. Additionally, K/π separation power for a track with $p = 20$ GeV/c was extrapolated to different track lengths starting at 1 m. Figure 10 shows the K/π separation power as a function of track length. The CEPC design requires a 3σ K/π separation for momenta up to 20 GeV/c. Using the ML-based reconstruction algorithm, the current drift chamber design, with a radius ranging from 600 to 1800 mm, meets the necessary PID requirements.

5 Conclusion

In this study, we developed a cluster counting algorithm that incorporates both peak finding and clusterization algorithms based on ML. Our approach offers several advantages over traditional cluster counting methods. In particular, our peak finding algorithm demonstrated better efficiency than the derivative-based algorithm. The clusterization algorithm provides a Gaussian-distributed number of clusters and achieves an efficiency close to that of the ground truth (MC truth). The entire cluster counting algorithm outperformed traditional methods, showing a 10 % improvement in the K/π separation power. This level of PID performance with ML-based algorithms is approximately equivalent to having a 20% larger detector size than traditional algorithms. With such performance, the current design of the CEPC drift chamber meets the necessary PID requirements.

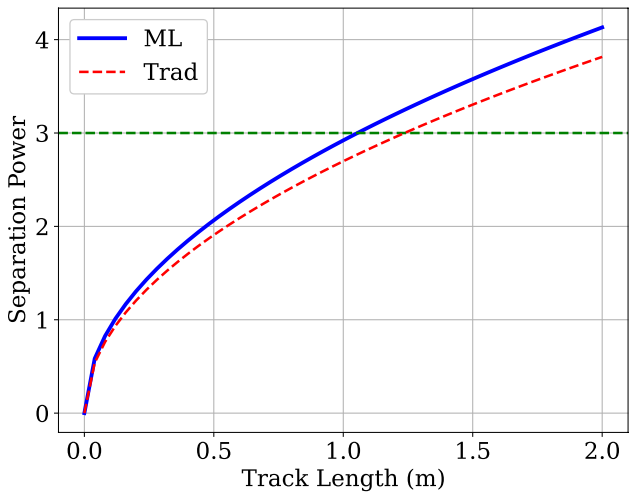

Fig. 10 The K/π separation power as a function of track length (L) at 20 GeV/c. The curve is extrapolated from the value at $L = 1$ m by $\sqrt{L}$. The red dashed line is from the traditional algorithm. The blue solid line is from our ML-based algorithm with two steps. The green dashed line shows the target of 3σ separation power. (Color figure online)

Table 3 Efficiency and separation power for charged K and π at various momenta and for different algorithms

Algorithm	Metric	Momentum (GeV/c)						
		5.0	7.5	10.0	12.5	15.0	17.5	20.0
ML-based algorithm	$\pi^{\pm}$ efficiency	1.003	1.001	0.999	0.999	0.998	0.998	0.999
	$K^{\pm}$ efficiency	1.014	1.011	1.010	1.008	1.006	1.004	1.003
	K/π separation power	4.203	4.279	4.081	3.832	3.509	3.216	2.921
Traditional algorithm	$\pi^{\pm}$ efficiency	0.814	0.808	0.803	0.801	0.801	0.800	0.800
	$K^{\pm}$ efficiency	0.837	0.830	0.824	0.820	0.817	0.814	0.812
	K/π separation power	3.888	3.954	3.765	3.550	3.277	3.054	2.697

The threshold of the ML-based algorithm is optimized as 0.95 for the LSTM-based peak finding algorithm and 0.26 for the DGCNN-based clusterization algorithm. The efficiency is defined as the ratio of the number of reconstructed clusters to the number of MC truth clusters

Furthermore, the critical role of ML-based algorithms in cluster counting suggests their potential applications in future high-energy physics experiments.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Zhe-Fei Tian and Guang Zhao. The first draft of the manuscript was written by Zhe-Fei Tian and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Data availability The data that support the findings of 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.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. The CEPC Study Group, CEPC Technical Design Report - Accelerator (2023). https://doi.org/10.48550/arXiv.2312.14363
2. The CEPC Study Group, CEPC conceptual design report: volume 2 - physics & detector (2018). https://doi.org/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). https://doi.org/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). https://doi.org/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 $(e^+e^-/\mu^+\mu^-)H$ production at the CEPC. Chin. Phys. C **44**, 013001 (2020). https://doi.org/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). https://doi.org/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). https://doi.org/10.1140/epjc/s10052-019-7602-x
8. Z.J. Liang, Electroweak physics at CEPC. Int. J. Mod. Phys. A **34**, 1940013 (2019). https://doi.org/10.1142/S0217751X1940013X
9. J. Gao, CEPC-SPPC accelerator status towards CDR. Int. J. Mod. Phys. A **32**, 1746003 (2017). https://doi.org/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). https://doi.org/10.1142/S0217751X21420057
11. T.F. Zheng, J. Xu, L. Cao et al., Analysis of $B_c \to \tau\nu_\tau$ at CEPC. Chin. Phys. C **45**, 023001 (2021). https://doi.org/10.1088/1674-1137/abcf1f
12. L.F. Li, M.Q. Ruan, Y.D. Wang et al., Analysis of $B_s \to \phi\nu\bar{\nu}$ at CEPC. Phys. Rev. D **105**, 114036 (2022). https://doi.org/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. Methods A **1047**, 167835 (2023). https://doi.org/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. Instrum. Methods **62**, 262–268 (1968). https://doi.org/10.1016/0029-554X(68)90371-6
15. W. Blum, W. Riegler, L. Rolandi, *Particle Detection with Drift Chambers* (Springer, Berlin, 2008). https://doi.org/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). https://doi.org/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). https://doi.org/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). https://doi.org/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). https://doi.org/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). https://doi.org/10.1016/j.aiopen.2021.01.001
21. H.L. Qu, L. Gouskos, Jet tagging via particle clouds. Phys. Rev. D **101**, 056019 (2020). https://doi.org/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). https://doi.org/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). https://doi.org/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. (in Chinese) **46**, 040014 (2023). https://doi.org/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). https://doi.org/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). https://doi.org/10.1007/s11433-023-2116-0
27. Z.P. Gao, Q.F. Li, Studies on several problems in nuclear physics by using machine learning. Nucl. Tech. (in Chinese) **46**, 080009 (2023). https://doi.org/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). https://doi.org/10.48550/arXiv.1912.01703
29. M. Fey, J.E. Lenssen, Fast graph representation learning with PyTorch geometric, in *ICLR Workshop on Representation Learning on Graphs and Manifolds* (2019). https://doi.org/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. Methods A **935**, 121–134 (2019). https://doi.org/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. Methods A **1048**, 167969 (2023). https://doi.org/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). https://doi.org/10.1016/j.cpc.2024.109208
33. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Comput. **9**, 1735–1780 (1997). https://doi.org/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). https://doi.org/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). https://doi.org/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). https://doi.org/10.1162/089976600300015015
37. J. Han, C. Moraga, The influence of the sigmoid function parameters on the speed of backpropagation learning, in *From Natural to Artificial Neural Computation* (1995). https://doi.org/10.1007/3-540-59497-3_175
38. D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in *Proceedings of the 3rd International Conference on Learning Representations* (2015). https://doi.org/10.48550/arXiv.1412.6980
39. T. Akiba, S. Sano, T. Yanase et al., Optuna: a next-generation hyperparameter optimization framework, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019), pp. 2623–2631. https://doi.org/10.48550/arXiv.1907.10902
40. J. Gilmer, S.S. Schoenholz, P.F. Riley et al., Neural message passing for quantum chemistry, in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 (2017), pp. 1263–1272. https://doi.org/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**, 146 (2019). https://doi.org/10.1145/3326362
42. Z.H. Zhang, Introduction to machine learning: k-nearest neighbors. Ann. Transl. Med. **4**, 218 (2016). https://doi.org/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). https://doi.org/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). https://doi.org/10.5121/ijdkp.2015.5201