

# ***Energy Distribution of Particle Flow in PP Collider: Evidenced by BP and MLP Neural Networks Prediction***

**Xiaoxi Zhou<sup>1\*</sup>, Ao Qiu<sup>2</sup>, Ziyang Jiang<sup>2</sup>, Hua Yong<sup>3</sup>, Kubota Rikako<sup>4</sup>, Keyu Zhang<sup>5</sup>**

<sup>1</sup>*Nanjing Foreign Languages School, Nanjing, China*

<sup>2</sup>*ShenYuan College, Beihang University, Beijing, China*

<sup>3</sup>*Chengdu Shishi High School, Chengdu, China*

<sup>4</sup>*Beijing Chaoyang Kaiwen Academy, Beijing, China*

<sup>5</sup>*University College London, London, United Kingdom*

*\*Corresponding Author. Email: xiaoxizhou27@gmail.com*

**Abstract:** Particle flow analysis is pivotal in high-energy physics experiments, especially for the examination of particle collisions. By combining calorimeter data from neutral and charged particles, this paper explores energy distributions within proton-proton (PP) collisions, employing machine learning techniques. Specifically, data from the Inner Tracker, Electromagnetic Calorimeter, and Hadronic Calorimeter are utilized to generate predictive graphs, which are then compared to reconstructed truth-level data. This approach establishes a benchmark for assessing the accuracy of particle flow analysis techniques. Furthermore, we demonstrate the application of backpropagation and multilayer perceptron (MLP) neural networks to enhance particle identification, incorporating magnetic field-induced deflection for improved prediction accuracy.

**Keywords:** Particle Flow Algorithm, Backpropagation, Multilayer Perceptron Neural Networks

## **1. Introduction**

In high energy physics, particle flow algorithms are employed to reconstruct events by integrating measurements from sub-detectors such as tracker, the Electromagnetic Calorimeter (EMCal) and Hadronic Calorimeter (HCal). This approach facilitates the reconstruction of charged particle trajectories in magnetic fields and energy deposition in calorimeters, providing vital data for measuring the energy of particles. [1] Particle flow is to optimally combine these measurements to reconstruct the particle content and energy response of the entire event, thereby enhancing energy resolution in collision experiments.

The particle detector captures data from particles emerging after high-energy collisions, with each sub-detector designed to track, stop, or measure particle energy. In this setup, particles first encounter the tracker, which records their positions to determine their trajectory and momentum. [2] The momentum of a particle is inferred by observing its trajectory in a magnetic field; a more curved path

indicates lower momentum. The detector generates a uniform 3.8 Tesla magnetic field over a 12.5-meter length and a 3.15-meter radius, encompassing the tracker, EMCal, and HCal. [3] The tracking system, composed of silicon pixel and microstrip detectors, records the paths of charged particles by generating small electric signals that are amplified and detected. Challenges in particle flow include accurately matching reconstructed data with actual events and managing large datasets, requiring substantial computational power. The performance is limited by both algorithm accuracy and hardware capabilities [4].

This paper investigates different statistical methods and neural networks for data-driven graph construction: backpropagation (BP) and multilayer perceptrons (MLP). [5] Model performance is evaluated using three error metrics: 1D error histograms, 2D histograms of truth vs. predicted data, and 2D energy distributions. These methods enable effective determination of the optimal model for accurate event reconstruction.

## 2. Methodology

### 2.1. Data and Methods

The truth data for this study is generated using Monte Carlo simulations combined with seed cluster reconstruction, a technique widely employed in high-energy physics for simulating particle events. [6] In each simulated event, multiple particles are assigned random properties, such as energy, charge, and type (e.g., neutral pions or hadrons). These particles are randomly distributed within the eta-phi space, mimicking the complexities encountered in real experimental data. Detector responses are then generated for each event, capturing the interactions of particles with various components of the detector, including the tracker, electromagnetic calorimeter (EMCal), and hadronic calorimeter (HCal). The analysis reveals that all four detectors show a positive correlation with the Truth values, suggesting that they can effectively approximate the actual measurements. The ultimate goal is to develop a predictive model that accurately estimates the truth data based on these detector responses, leveraging the observed relationships.

### 2.2. Backpropagation Algorithm

#### 2.2.1. Research Design

This study aims to process TIFF images using Python and extract information from three sub-images to train a Back Propagation Neural Network (BPNN) for predicting truth values. The dataset consists of 50,000 TIFF images, each with a resolution of 56×56 pixels. The images are categorized into tracker p, tracker n, emcal, hcal, and truth, with 10,000 images in each category. The data is divided into a training set and a testing set, each containing 5,000 images.

#### 2.2.2. Data Collection and Preprocessing

First, we used the `skimage` library in Python to read each TIFF image and extract the pixel values row by row. These pixel values were then converted into CSV files for further analysis and processing.

#### 2.2.3. Model Design and Training

Next, the preprocessed data was fed into the BPNN model in MATLAB for training. The BPNN operates through two main phases: forward propagation and backpropagation. During forward propagation, input data passes through each layer of the network until it reaches the output layer, where

each layer's output serves as the input for the next layer. In the backpropagation phase, the network adjusts its weights and thresholds based on the error between the predicted output and the expected output, gradually reducing the error and improving the model's accuracy. [7]

The key steps include:

1. Forward propagation to compute the predicted results;
2. Calculation of the loss function (e.g., Mean Squared Error, MSE);
3. Gradient calculation using backpropagation;
4. Updating weights and biases using an optimizer.

BPNN is a classical supervised learning algorithm widely used in pattern recognition, function approximation, and data classification. Its core mechanism lies in continuously optimizing the network's weights and biases through gradient descent to minimize the error between the predicted and actual values. This enables BPNN to effectively learn and optimize complex nonlinear patterns, demonstrating excellent performance across various practical problems. [8]

The BPNN model designed for this study comprises 4 input nodes, 1 output node, and 10 hidden layers.

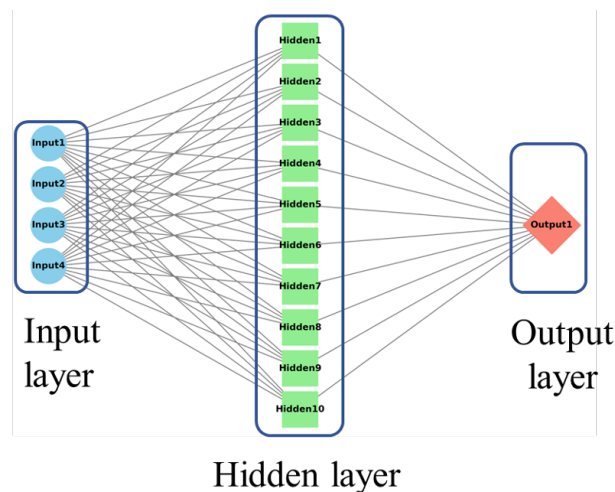


Figure 1: Layers of BPNN , which consists of an input layer, a hidden layer, and an output layer.

By adjusting the network's weights and biases, the BPNN model gradually learns and establishes a mapping relationship between the input features and the truth values.

## 2.3. Multi-Layer Perceptron (MLP)

### 2.3.1. Research Design

The Multi-Layer Perceptron (MLP) method is employed in this study as an alternative approach to model the non-linear relationships inherent in the dataset. The MLP is a type of neural network that, like BPNN, consists of multiple layers of neurons. Each neuron in the network processes input data through a set of learned weights and biases. The purpose of this design is to capture more complex dependencies that may arise in the data, particularly those resulting from non-linear interactions.

### 2.3.2. Data Collection and Preprocessing

Similar to the BPNN approach, the data collection and preprocessing for the MLP were conducted using the `skimage` library in Python. The TIFF images were read and the pixel values were extracted row by row. These pixel values were then organized into CSV files for further analysis. This preprocessing step is crucial for ensuring that the input data is in a suitable format for the MLP model.

### 2.3.3. Model Design and Training

The preprocessed data was then input into the MLP model. The MLP, much like the BPNN, operates through forward propagation and backpropagation. [9] During forward propagation, the input data passes through each layer of neurons until it reaches the output layer, where the final prediction is made. In the backpropagation phase, the model adjusts its weights and biases based on the error between the predicted and actual truth values, thereby improving its accuracy.

The key steps in the MLP training process include:

1. Forward propagation to compute the output predictions;
2. Calculation of the loss function (e.g., Cross-Entropy Loss or MSE);
3. Backpropagation to compute the gradient of the loss function;
4. Updating of weights and biases through an optimization algorithm.

The MLP framework is a robust method for capturing non-linear patterns in data, but it may face challenges when dealing with complex dependencies, such as those introduced by energy distribution deflections in charged particles like electrons and photons. [10] These deflections often create non-linearities and spatial dependencies that an MLP's fixed architecture might struggle to fully capture.

Despite these challenges, the MLP showed an improvement by effectively capturing some of the non-linear relationships within the data. However, it still requires further optimization to fully address the intricate dependencies present in the dataset.

## 3. Results

### 3.1. 1D Histogram of the Error

For the assessment of the model's performance, 1D histograms of the prediction error is constructed, where the error is defined as the discrepancy between the actual and the predicted values. To prevent bias in the results, instances where the actual value equals zero, along with their corresponding predicted values, are excluded. This exclusion is necessary because predicting zeros is relatively trivial, and the true data contains a large number of zeros, which are also matched by zeros in the Tracker, EMCal, and HCal. Including these would result in a high peak at zero in the histogram, which does not provide a meaningful assessment of our model's predictive capability. The ideal error distribution is anticipated to align with a Gaussian distribution, as depicted in Figure 2. The efficacy of the model is quantified through the calculation of the standard deviation and the Root Mean Square (RMS) of the errors. A reduction in these values suggests that the errors are more closely clustered around zero, indicative of enhanced prediction accuracy. The aspiration is to minimize these metrics, thereby achieving a model with superior predictive precision. The magnetic field's magnitude serves as a measure of particle deflection, with the Monte Carlo simulations configured to include magnetic field strengths of 0.00 and 0.50. It is hypothesized that a stronger magnetic field leads to a greater degree of particle deflection.

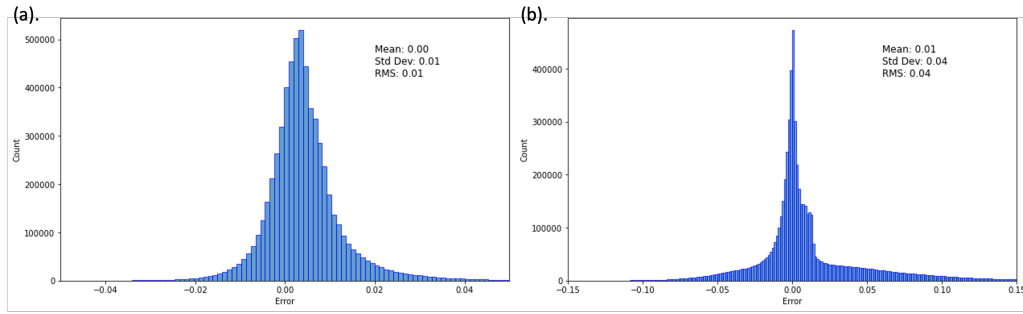


Figure 2: The diagrams present 1D histograms for MLP-BP, where the x-axis represents the error, defined as the difference between the predicted value and the true value. Diagram (a) corresponds to the condition without a magnetic field, while diagram (b) depicts the condition with a 0.50T magnetic field.

### 3.2. Two-dimensional histogram of real and predicted data

To further evaluate the models, two-dimensional histogram is constructed with the true data on the horizontal axis and the predicted data on the vertical axis. The scatter plot is colored to reflect the frequency of occurrences, with the color bar enhancing the visualization. The closer a point is to yellow, the more frequently that data point occurs; conversely, purple indicates fewer occurrences. A green line representing  $y = x$  is included in the plot, serving as a reference line—the closer the data points are to this line, the better the model's prediction accuracy. Figure 3 displays the 2D histogram results for the MLP model using BP algorithm.

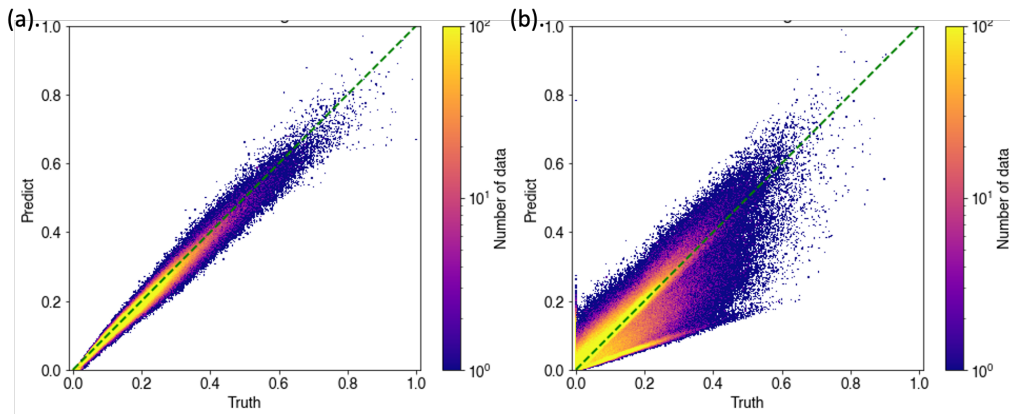


Figure 3: The diagrams present 2D histograms for MLP-BP, where the x-axis represents energy value in truth diagram and y-axis stands for energy value in predicted truth diagram. Diagram (a) corresponds to the condition without a magnetic field, while diagram (b) depicts the condition with a 0.50T magnetic field.

### 3.3. Two-dimensional Distribution of Energy

The spatial distribution of energy was analyzed by segmenting the rapidity and azimuthal angle of the entire space into 56 equal divisions, which served as the horizontal and vertical axes, respectively.

Each cell on the resulting plot represents a specific range of azimuths and rapidity within the space. The color intensity within each cell indicates the magnitude of the energy. This method facilitates the visualization of the spatial energy distribution through a two-dimensional graph.

To enhance the interpretability of the results, two types of visualizations were employed:

- **Color Map:** This visualization technique is utilized to juxtapose the predicted energy distribution with the actual data. The color map accentuates the distribution of energy, where darker shades are indicative of regions with higher energy concentrations. This approach aids in the direct comparison of predicted and actual energy distributions, highlighting any discrepancies visually.
- **Black and White Map:** This visualization is designed to illustrate the absolute difference between the predicted and actual data, focusing on the error distribution. In this map, darker shades denote larger errors, thereby simplifying the assessment of the prediction accuracy. The more prominent the dark areas, the greater the divergence between the predicted and true values.

Figure 4 displays the two-dimensional energy distribution outcomes for the MLP model, providing a comprehensive overview of the spatial energy distribution and the accuracy of the predictive model.

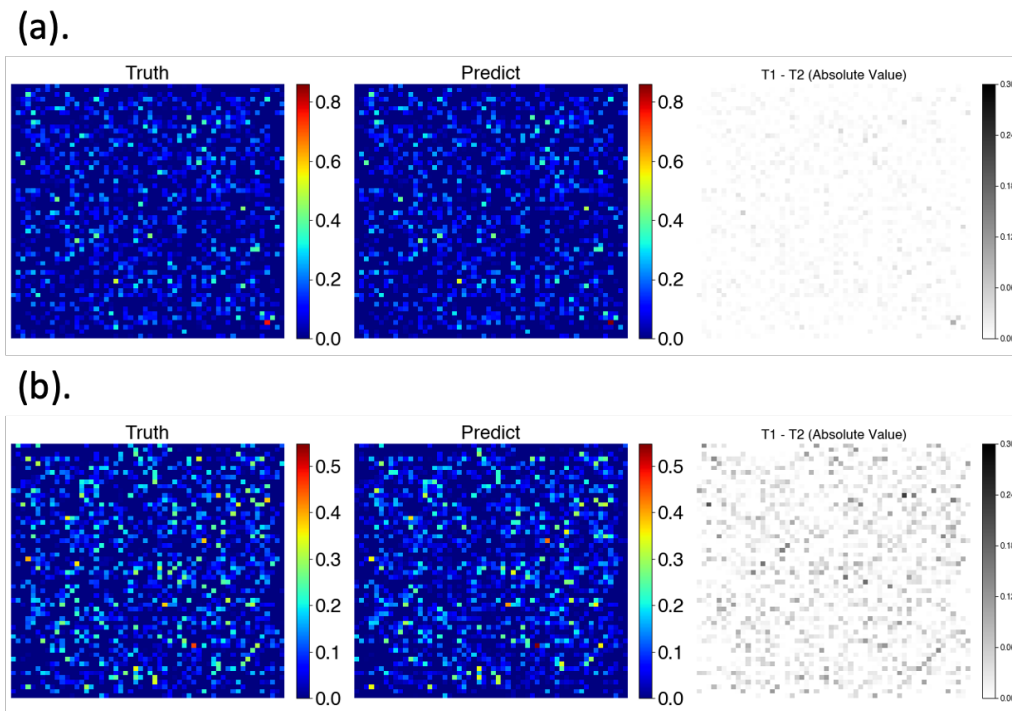


Figure 4: The diagrams show 2D energy distribution maps, where each pixel represents energy. The first diagram illustrates the true energy distribution, the second shows the predicted energy distribution, and the third represents the error, with each pixel showing the difference between the predicted and true values. Diagram (a) corresponds to the condition without a magnetic field, while diagram (b) represents the condition with a 0.50T magnetic field.

## 4. Discussion

### 4.1. Further Analysis

Figure 1 indicates that when there is no deflection in the data, the prediction results align well with the actual data. In cases where the data exhibits deflection, the prediction results are less ideal. However, the distribution of errors across the dataset still conforms to a Gaussian distribution. And the standard deviation and root mean square (RMS) values are relatively low, which is indicative of the model maintains a level of performance that is considered to be within the bounds of acceptability for the intended application. The analysis of Figure 2 reveals several key insights into the model's performance. Firstly, the lighter color intensity near the green line signifies a higher density of data points in proximity to the actual values, suggesting that the majority of predictions are reasonably accurate. Secondly, the model demonstrates increased accuracy for high-energy data, which is expected as these particles typically undergo smaller deflections, facilitating more precise predictions. In contrast, the accuracy diminishes for low-energy data due to larger deflections, underscoring a current limitation in the prediction method and highlighting the necessity for enhanced accuracy in this domain.

### 4.2. Comparison with Existing Literature

The machine-learned particle-flow (MLPF) algorithm has been widely applied in particle physics to reconstruct particle collisions and their energy distributions. [11] [12] Previous studies have primarily focused on using machine learning techniques such as graph neural network(GNN) structures and LRP algorithms for reconstructing events like electron-positron and other collisions. [13] [14] In contrast, this paper primarily utilizes MLP structures along with the BP algorithm to focus on energy reconstruction. Furthermore, this study has taken into account the effects induced by magnetic deflection, a critical factor that has been shown to significantly influence the energy distribution of particles. The reconstruction targets are pp collisions and heavy-ion collisions. This is because we aim to establish a foundation for further jet reconstruction and the search for missing transverse energy by obtaining the energy distribution after collisions, which is crucial for exploring the properties of quark gluon plasma(QGP).

### 4.3. Limitations of the Study

There are several factors that may influence the results, which mainly includes the size of the dataset, the complexity of the model and the number of training iterations. In this study, The dataset size is sufficiently large to support robust model training. Instead of one hidden layer, multiple layers are applied in this study to enhances the flexibility and efficiency of our model. And increasing the number of training iterations can further improve the performance of machine learning models. Machine learning enables us to bypass the direct computation of intricate physical processes, such as the deflection of particles and their interactions within Tracker, EMCal, and HCal—processes that are often too complex or infeasible to calculate explicitly(In Monte Carlo simulations, the random generation of particle types and energies allows for the derivation of energy distributions within detectors. However, during actual collision processes, a single event may encompass a multitude of particles, complicating the direct calculation of their deflections based solely on energy measurements from the detectors). Instead, these interactions are embedded within the machine learning framework, allowing us to directly facilitate the energy distribution in space after the collisions from the experimental energy loss data of Tracker, EMCal, and HCal. However, a limitation of this method is its reliance on available training and validation datasets. In this study, we utilized data from proton-proton collision simula-

tions for model training. If similar data cannot be simulated or acquired for other types of collisions, the effectiveness of the machine learning approach may be compromised.

## 5. Conclusion

Our analysis of the predictive accuracy, using MLP structure and BP to train, involved three distinct methodologies: one-dimensional error histograms, two-dimensional histograms contrasting real data with predictions, and two-dimensional energy distribution. Each approach provided a unique perspective on the same set of predictions, highlighting different aspects of the correlation between the predicted outcomes and the actual data. Nevertheless, the overarching conclusion is consistent across all methods: predictions are most accurate in the absence of deflection. Although the accuracy decreases in the presence of a magnetic field, the overall results still demonstrate strong performance.

Looking ahead, our primary objectives are to improve both the accuracy and efficiency of our algorithms. On the hardware side, we plan to incorporate GPU capabilities into our computational infrastructure to enhance processing power. On the software side, adopting advanced machine learning frameworks such as TensorFlow and PyTorch will be crucial for refining our models. Additionally, we aim to address the issue of energy loss in jet event reconstruction by identifying where this energy dissipation occurs. These advancements are essential for refining our models and achieving more accurate and reliable results in future studies.

## Acknowledgement

We would like to express our deepest gratitude to Professor Gunther Roland for his invaluable guidance and support throughout the course of this research. His insights and encouragement have been instrumental in the successful completion of this work.

## Data Availability

The datasets generated and analyzed during the current study are available from the corresponding author upon reasonable request. The data is also publicly accessible through the GitHub repository at the following URL: <https://github.com/lexie2749/Energy-Distribution-Prediction-of-Particle-Flow-Using-BP-and-MLP-Neural-Networks.git>.

Researchers interested in utilizing the data for their own studies are encouraged to appropriately reference this work. Detailed instructions on how to access and use the data are provided on the repository page.

## References

- [1] Deepak Kar. Transverse energy flow and charged particle event shapes with atlas. In *Proceedings of XXI International Workshop on Deep-Inelastic Scattering and Related Subjects – PoS(DIS 2013)*, 2013.
- [2] CERN Accelerating science. Energy of electrons and photons (hcal). <https://cms.cern/detector/measuring-energy/energy-electrons-and-photons-ecal>. n.d.
- [3] A. Sirunyan, A. Tumasyan, W. Adam, E. Asilar, T. Bergauer, J. Brandstetter, E. Brondolin, M. Dragicevic, J. Erö, M. Flechl, M. Friedl, R. Frühwirth, V. Ghete, C. Hartl, N. Hörmann, J. Hrubec, M. Jeitler, A. König, I. Krätschmer, and A. Caudron. Particle-flow reconstruction and global event description with the cms detector. *Journal of Instrumentation*, 12(10):P10003, 2017.

- [4] C. D. Saunter. Quantifying subpixel accuracy: An experimental method for measuring accuracy in image-correlation-based, single-particle tracking. *Biophysical Journal*, 98(8):1566–1570, 2010.
- [5] A. Saleh, M. Sheaves, D. Jerry, and M. R. Azghadi. Applications of deep learning in fish habitat monitoring: A tutorial and survey. *Expert Systems With Applications*, 238:121841, 2024.
- [6] S. Deb. Reconstruction of cluster masses using particle based lensing. <https://doi.org/10.17918/etd-3284>, n.d. Accessed: 2024-09-03.
- [7] *Forward and Backpropagation*, pages 9–34. Cambridge University Press, 2022. Accessed: 2024-09-03.
- [8] T. K. Hua. Supervised learning algorithm. <https://doi.org/10.31219/osf.io/tuk56>, 2022. Accessed: 2024-09-03.
- [9] S. Stalin and T. V. Sreenivas. Vectorized backpropagation and automatic pruning for mlp network optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, 1993. n.d.
- [10] Z. Sroubek and G. Falcone. Energy distribution of electrons excited by slow particles in solids. In *NATO ASI Series*, pages 593–598. 1991.
- [11] F. Mokhtar, J. Pata, J. Duarte, E. Wulff, M. Pierini, and J.-R. Vlimant. Progress towards an improved particle flow algorithm at cms with machine learning. *arXiv.org*, 2023. Accessed: Sep. 02, 2024.
- [12] J. Pata et al. Machine learning for particle flow reconstruction at cms. *Journal of Physics: Conference Series*, 2438(1):012100, Feb 2023.
- [13] F. Mokhtar et al. Explaining machine-learned particle-flow reconstruction. *arXiv.org*, 2021. Accessed: Sep. 02, 2024.
- [14] J. Pata et al. Improved particle-flow event reconstruction with scalable neural networks for current and future particle detectors. *Communications Physics*, 7(1), Apr 2024.