

Optimization of convolutional neural networks for background suppression in the PandaX-III experiment

Shangning Xia², Suizhi Huang², Kexin Xu², Tao Li⁴, Xun Chen^{*1,3}, Ke Han¹, and Shaobo Wang^{†1,2}

¹INPAC and School of Physics and Astronomy, Shanghai Jiao Tong University, MOE Key Lab for Particle Physics, Astrophysics and Cosmology, Shanghai Key Laboratory for Particle Physics and Cosmology, Shanghai 200240, China

²SJTU-Paris Elite Institute of Technology, Shanghai Jiao Tong University, Shanghai, 200240, China

³Shanghai Jiao Tong University Sichuan Research Institute, Chengdu 610213, China

⁴Sino-French Institute of Nuclear Engineering and Technology, Sun Yat-sen University, Zhuhai 519082, China

October 27, 2023

Abstract

The tracks recorded by a gaseous detector provide a possibility for charged particle identification. For searching the neutrinoless double beta decay events of ^{136}Xe in the PandaX-III experiment, we optimized the convolutional neural network based on the Monte Carlo simulation data to improve the signal-background discrimination power. EfficientNet is chosen as the baseline model and the optimization is performed by tuning the hyperparameters. In particular, the maximum discrimination power is achieved by optimizing the channel number of the top convolutional layer. In comparison with our previous work, the significance of discrimination has been improved by $\sim 70\%$.

1 Introduction

Neutrinoless double beta decay ($0\nu\beta\beta$) is an extremely rare type of nuclear decay process, during which two electrons are emitted from a nucleus with an atomic number of Z , resulting in a nucleus of $Z + 2$, without the emission of neutrinos. The discovery of such a process would imply the neutrinos are likely Majorana particles [1], and the violation of lepton number conservation.

Many experiments have been carried out to search for the $0\nu\beta\beta$ with different isotopes, such as ^{76}Ge [2, 3], ^{130}Te [4], and ^{136}Xe [5, 6, 7], through the spectrum analysis around the Q -value, which is the energy released by the decay and shared by the two emitting electrons. The current lower half-life limits of the three isotopes are 2.3×10^{26} yr [6], 1.8×10^{26} yr [2], and 3.2×10^{25} yr [4] (90% confidence level, or C.L.), established by KamLAND-Zen, GERDA, and CUORE experiments respectively. Its predicted half-life is longer if the mass order of neutrinos is a normal hierarchy. Therefore, necessary new experiments with larger volumes and new technologies are being actively planned and constructed to further improve detection sensitivity. Reducing the background level in the energy region of interest (ROI) is one of the key requirements for the discovery of such a rare process. The gaseous detectors are capable to record event tracks, providing an additional way to discriminate signals

*corresponding author: chenxun@sjtu.edu.cn

†corresponding author: shaobo.wang@sjtu.edu.cn

from backgrounds through extracting track features, thus being actively studied in recent years [8]. In general, the gaseous detector-based experiments utilize the prominent Bragg Blob (BB) feature of event tracks. Each end of a $0\nu\beta\beta$ track has a BB, in which rapid energy loss in unit volume happens because of increased differential energy loss (Bragg peak) and larger scattering angles before an electron stops. However, only one BB is on the ends of the track for the single electron background. The PandaX-III experiment, which utilizes a high-pressure gaseous xenon time projection chamber (TPC) to search for $0\nu\beta\beta$ decay of ^{136}Xe [9], is one of them. The study of this type of detector is pioneered by the Gotthard experiment [10]. Algorithms to extract the different topological features of signal and background tracks within gaseous detectors have been developed with the operation and planning of new experiments [11, 12, 13].

Machine learning techniques, such as deep convolutional neural networks (CNN), have been introduced into experimental particle physics for data analysis and particle identification [14, 15, 16]. Recently, methods of neural networks are applied in $0\nu\beta\beta$ search for background and signal classification. Promising results are obtained based on both simulation data and experimental data in the NEXT experiment [17]. In the PandaX-III experiment, a special type of CNNs is used to discriminate the $0\nu\beta\beta$ signal from the gamma backgrounds within the same energy window based on simulation, leading to an average improvement of 61.8% on the efficiency ratio of $\epsilon_s/\sqrt{\epsilon_b}$ (3.82 to 6.17) in comparison with its design baseline [18].

In this work, we present the optimization of the neural network to improve the discrimination power based on the design of the PandaX-III detector. As a subsequent work of Ref. [18], several improvements are made in the treatment of Monte Carlo (MC) simulation and the optimization of the neural networks. We also aim to understand the process of extracting and enhancing the physical characteristics of tracks in CNN, which serves as the basis for optimizing the model and improving identification ability. The article is organized as follows. A brief introduction to the PandaX-III detector is given in Sec. 2. The data preparation, including the simulation of $0\nu\beta\beta$ and background events, as well as the post-processing of data are discussed in Sec. 3. Detailed information on the selection and optimization of the neural networks and the results are presented in Sec. 4. At last, a summary is given in Sec. 5.

2 The PandaX-III experiment

The PandaX-III experiment under construction aims to search for $0\nu\beta\beta$ with enriched ^{136}Xe using high-pressure gas TPC in the China Jinping underground laboratory (CJPL) [19, 20]. The updated geometry of the PandaX-III detector is shown in Fig. 1, which is different from that in the CDR [9] and the previous work [18]. It has an active volume confined by the field cage, cathode, and readout plane, with a drift length of 120 cm in the vertical direction (Z) and a diameter of 160 cm in the horizontal direction (XY). It can contain ~ 140 kg Xe gas with 90% enriched ^{136}Xe , together with 1% trimethylamine (TMA) at the operating pressure of 10 bar. The charge readout plane consists of 52 Micromegas modules [21], each with a geometry of 20×20 cm² and readout strips at 3 mm pitch. The cathode and the electric field shaping cage are installed inside the copper substrate with a width of 15 cm, which is used to shield the radioactive background outside of the TPC. The materials have been carefully selected to ensure low radioactive background inside the TPC. More detailed information can be found in Ref. [22].

The background for the search of $0\nu\beta\beta$ is from the radioactivities of detector construction materials (^{232}Th , ^{238}U , ^{40}K , and ^{60}Co). Considering the energy distribution of the sources. The majority contribution is gamma from the radioactive decay chains of ^{238}U and ^{232}Th , which may leave energy depositions around the Q-value. The advantage of the gaseous detector is to suppress these backgrounds with the recorded track features.

3 Data preparation

Since there are no recorded data available, we use the simulated data to study the methods of background suppression. The simulated data will be converted to image format so that

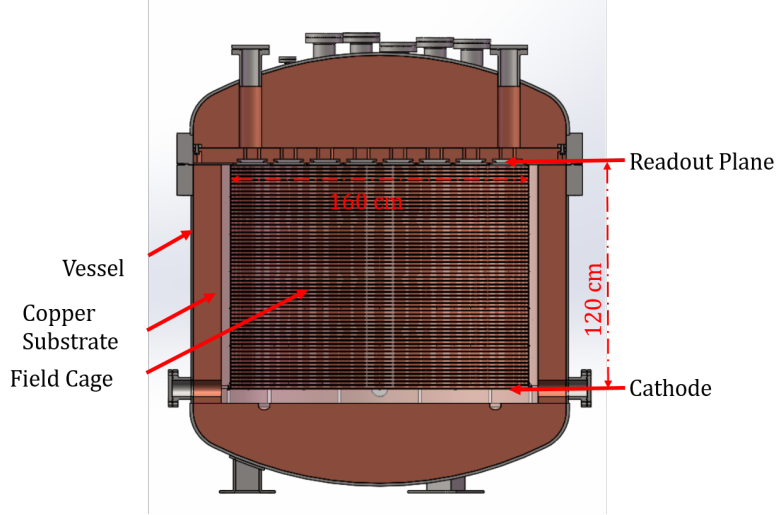


Figure 1: Illustration of the PandaX-III TPC from the cathode at the bottom to the charge readout plane with Micromegas on the upper side. All the main components are annotated.

it could be accepted by the CNNs.

3.1 Simulation and data production

The Monte Carlo simulation is based on the detector geometry shown in Fig. 1. The build of detector geometry in simulation is described in [23]. All the components of the detector, including the vessel, the TPC and the dimension of the readout plane, are simulated. The process of data preparation is shown in Fig. 2. The physics simulation is performed with the Geant4 [24] based package restG4, which is a component of the REST framework [25]. The input signal $0\nu\beta\beta$ events are generated with the DECAY0 package [26]. The background events from the decay chains of ^{232}Th and ^{238}U are sampled directly with restG4. The output energy depositions inside the TPC are processed with the REST framework, generating responses from the detector and data readout by the electronics [27]. The readout data is converted to images for further discrimination with the CNNs.

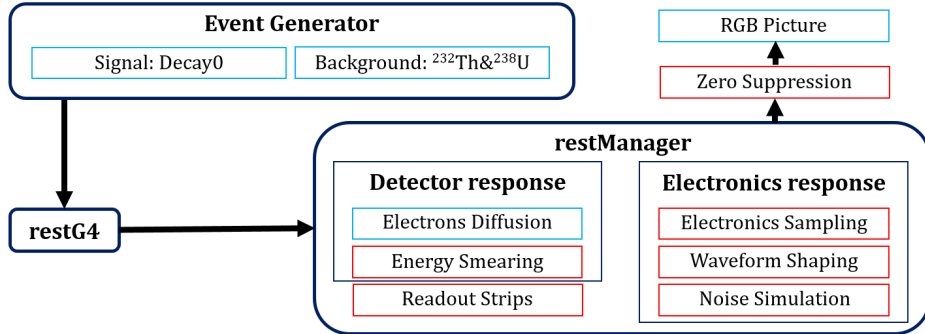


Figure 2: The process of data preparation. In comparison with data processing in Ref. [18], new steps (in red frames) are added.

For each of the signal events, the DECAY0 package generates the momenta of the two primary electrons, which share the same vertex. The vertices are sampled uniformly in the gas region of the detector during the simulation with restG4. The primary particles of the background events are the daughter nucleus ^{208}Tl (^{214}Bi) of the ^{232}Th (^{238}U) decay chain, of which the 2615 keV (2447 keV) gamma events contribute to the background within the ROI. They are sampled within the container, which contributed most of these backgrounds. Since we have not yet determined the background model, we used the same uniform distribution of both signal and background events to study the effect on pure track features in this work.

The restG4 package simulates the transport of the primary particles in the materials as well as the daughter particles which are generated in different physical processes. For each event, the energy loss of all the particles inside the sensitive region of the detector, together with the related positions, are recorded as energy depositions. Fig. 3 Left shows the recorded energy depositions of an example $0\nu\beta\beta$ event in the TPC.

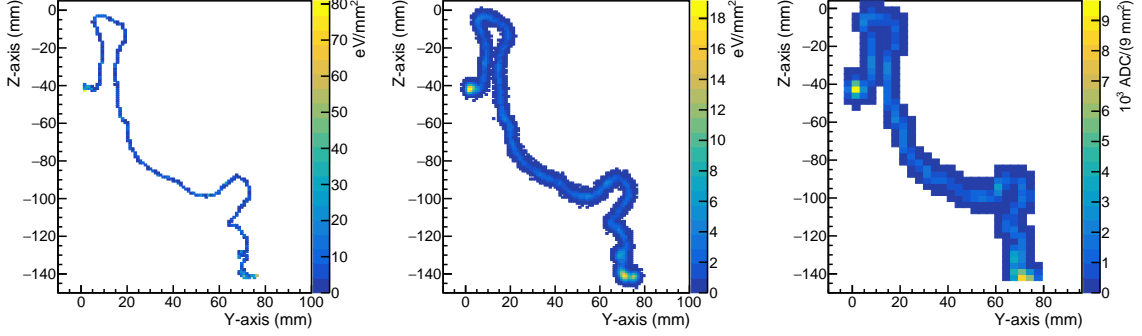


Figure 3: An example simulated $0\nu\beta\beta$ event at different processing stages by the REST framework, the projections on the YZ plane are shown. Left: The recorded energy depositions inside the PandaX-III TPC, simulated with restG4; Middle: The spatial distribution of ionized electrons reaching the readout plane; Right: The final spatial distribution of the readout signals.

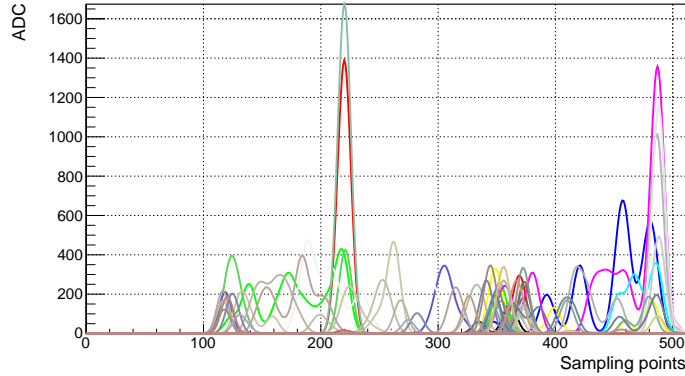


Figure 4: The generated waveforms of the example event, from the simulation of electronics response, noise and shaping by the REST framework, the waveform of the different strip is represented with different color.

The REST framework reads the energy depositions for further data processing. For each of the energy deposition points, ionization electrons are generated. The location distribution of the electrons (XYZ) is given by Geant4. The drift of the electrons to the readout plane and their diffusion [28] are simulated. The parameters of drift velocity and diffusion coefficient are calculated with the Garfield++ package [29] integrated inside REST. With the designed operation drift voltage of 1 kV/cm inside the TPC, the simulated drift velocity is 1.87 mm/ μ s, and the transverse (longitudinal) diffusion coefficients are $1.0(1.5)\times 10^{-2}$ cm^{1/2} [13]. Then the diffusion effect is simulated in both the transverse and longitudinal directions by performing a smear of Gaussian function according to the different locations of ionized electrons. The horizontal position of each smeared electron is the projection on the XY readout plane, and the time arriving at the readout plane is calculated by applying the vertical location (Z) of each smeared electron and the drift velocity. Fig. 3 Middle shows the spatial distribution of the recorded electrons in the same example event on the readout plane by adding the diffusion effect, where the z distribution is concluded from the drift time.

The REST framework simulates the electron collection by the Micromegas and the elec-

tronics response. For the simulation data, the REST framework groups the electrons to XY readout strips according to their position, and creates an electronic response following the same configuration used in the PandaX-III prototype detector [27], including a 5 MHz sampling rate, 5 μ s waveform shaping time, 120 fC dynamic range equivalent to 4096 ADC from electronics, and 0.3 fC electronic noise level obtained from our real data taking of 10 ADC. Therefore, the waveform for each readout strip is built. Fig. 4 shows the waveforms of the example event. The simulated waveforms are digitized with an interval between two sampling of 200 ns. The event window is chosen to be 102.4 μ s. Thus each event has 512 time bins. The effective event window is 412 time bins which can ensure the full collection of $0\nu\beta\beta$ events, the first 100 bins are used for the baseline and noise study as well as in the real data analysis. The generated data has the same format as the data taken in the real experiment. The spatial distribution of the final signals after the zero suppression of baseline and noise. in the example event is shown in Fig. 3 Right. For the energy smearing, we did not simulate respectively the contributions from the avalanche process of Micromegas, gain non-uniformity of the detector, and the electron lifetime effect. The amplitudes of energy depositions are smeared by a Gaussian function to achieve the final design index of the energy resolution of 3% FWHM (Full Width Half Maximum) at the Q-value.

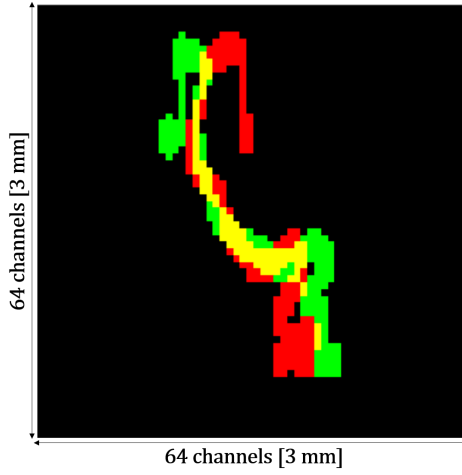


Figure 5: The result RGB image scaled from 64×64 pixels. The track projections in XZ and YZ planes are encoded in the red and green channels. All the pixels take the maximum values for better visualization.

3.2 Data set

The generated events within the $0\nu\beta\beta$ search ROI of PandaX-III, from 2395 keV to 2520 keV, are selected to create the data set used for the CNNs. The energy of tracks on the XZ and YZ plane is normalized to the total energy of the event, and the tracks are then filled into red (XZ) and green (YZ) channels of the input RGB image to the CNN model by aligning through charge centers of each plane. As the strip size of the readout plane is 3 mm, in our study, the image format is 64×64 pixels with a size of 3×3 mm² each. Therefore, the image is produced with the strip size in X or Y direction and is converted to the same size in the Z direction by calculating a drift distance using sampling points. The image size is 19.2 cm in each direction can ensure keeping the whole track of $0\nu\beta\beta$ inside the image area. Fig. 5 shows the resulting image of an example event.

The images then have been linearly scaled by a factor of 2 on each side to 128×128 by nearest neighbor interpolation before being fed into the network. The further discussion of this scaling is in Sec. 4.3. Besides, to prevent over-fitting and improve the diversity of input data, data augmentations are applied, such as the random horizontal and vertical flipping of images. These two methods involve horizontally or vertically flipping an input image with a certain probability during training phase. Consequently, for different iterations, although the actual size of the dataset is unchanged, the same image fed into the network may be

flipped or not. By exposing the model to both the original and flipped versions of an event repeatedly, the model learns to recognize the same patterns from different viewpoints, leading to better generalization ability when presented with new, unseen events.

Our data set contains a total of 320,000 images from the $0\nu\beta\beta$ events and the same number of images from background events, within which half are from ^{232}Th , and another half from ^{238}U . All the events are within the ROI. The images are divided into three sets in this study: 70% for training, 20% for validation, and the remaining 10% for the evaluation of the performance of the networks.

4 Optimization of CNNs

Numerous CNN architectures have been developed in recent years. InceptionV2 [30], DenseNet [31], and EfficientNet [32] have been proven to achieve remarkably higher accuracy and efficiency in the image classification than ResNet [33], which was used for the discrimination of background and signal events in the $0\nu\beta\beta$ search with the PandaX-III detector [18].

ResNet is a convolutional neural network architecture that introduced the concept of “skip connections” to enable the training of very deep neural networks. It allows information to flow directly from one layer to another without passing through all the intermediate layers. These skip connections create shortcuts that enable the network to learn residual mappings or the difference between the input and output feature maps of a layer. InceptionV2 and DenseNet also made some improvements based on this idea. InceptionV2 uses a series of convolutional filters with different sizes in parallel and splits an $n \times n$ convolution into $1 \times n$ and $n \times 1$ convolutions, allowing it to capture features of different scales effectively and to reduce the number of parameters and computations required. DenseNet uses a “dense connectivity” pattern, where each layer is connected to every other layer in a feed-forward fashion. This helps to improve the flow of gradients and information through the network, enabling it to learn more complex features. It also introduces a technique called “bottleneck layers”, which makes the training faster by compressing the input feature maps first and expanding them thereafter. This helps to reduce the overall complexity of the network while still maintaining its representational power. These ideas are then inherited by EfficientNet which we will explain in detail in paragraph 4.3.

Thus, these architectures provide a possibility to improve the capability of background suppression further in the PandaX-III experiment, as well as other experiments with similar technology.

4.1 Comparison of different CNNs

To evaluate a network’s performance for background suppression, the most important metric is the significance Ξ , which is defined as the highest ratio of $\epsilon_s/\sqrt{\epsilon_b}$ at different selected thresholds of the output of the network, where ϵ_s and ϵ_b are the accepting efficiencies of signal and background events, respectively. In the PandaX-III experiment, the $0\nu\beta\beta$ detection sensitivity is linearly and positively correlated with Ξ .

ResNet50, DenseNet169, InceptionV2, and EfficientNetB4 are selected in this study because they have a similar level of trainable parameters and could be trained on our limited computation resources. All the models are implemented with Tensorflow [34] and trained with the same generated data samples on two Nvidia Tesla V100 GPUs. The best weights with the smallest validation losses are saved for the classification of test data. The input shape of all the CNNs is modified to accept images with a resolution of 128×128 and no other optimization is done. The performance reference metrics are calculated with the test data and given in Table 1. ResNet50 gives a comparable Ξ with the result shown in [18], the tiny difference most likely originates from the detector geometry, a TPC with single readout plane is used in this work. Additionally, for the baseline model selection, as the network models show a big difference for the signal-background discrimination, the fluctuation of each model is not further studied. EfficientNetB4 gives the highest value of the significance Ξ . Thus it is used as the baseline model for further optimization.

| Network | Ξ |
|----------------|-------|
| ResNet50 | 6.80 |
| DenseNet169 | 7.92 |
| InceptionV2 | 5.86 |
| EfficientNetB4 | 9.80 |

Table 1: Performance of the selected CNNs with the same test data set.

4.2 Terminology

- Epoch: A complete pass through the entire training set during the training phase.
- Batch size: The number of samples sent into the neural network each iteration for the calculation of gradient descent optimization.
- Feature map: In CNN, a feature map is a 2D array of values that represents the output of a convolution operation between the input and the filter.
- Channel: Besides the width and the height of an input or output, the channel is the third dimension (e.g. a RGB image has 3 channels), which represents the number of feature maps produced by that layer.
- Receptive field: the receptive field refers to the area of the input image that a filter is influenced by. It represents the spatial extent of the input that contributes to the output of a single neuron. The receptive field tends to grow with the number of layers because each layer processes the output of the previous layer, incorporating information from a larger region of the input.
- Convolutional Layer: Like the mathematical definition of convolution, a convolutional layer is a layer of the neural network which has a certain number of filters. Each filter applies directly to the input and produces a feature map. Then, all the feature maps are stacked along the third dimension (channel) and constitute the final output of it.
- Max pooling: Max pooling is an operation that selects the maximum element from the region of the feature map covered by the filter, normally a 3×3 or 2×2 square.

4.3 Optimization of EfficientNetB4

The structure of the baseline EfficientNetB4 model used in the optimization is presented in Table 2. Its basic component is called MBConv (**M**obile inverted **B**ottleneck **C**onvolutional layer) [35, 36], which employs the depth-wise separable convolutional layer along with the SE (**S**queeze-and-**E**xcitation) module [37]. The depth-wise separable convolutional layer separates the normal convolution operation into two consecutive ones. First, it generates the output with the same channel number as its input, then, the output goes through a 1×1 convolutional layer with the desired channel number. By separating the spatial and channel-wise operations, depth-wise separable convolution can significantly reduce the number of parameters and computations required, while still maintaining or even improving the accuracy of the network. SE module is a type of attention mechanism that compresses the input’s spatial dimensions to a single channel and learns the weights for each channel, resulting in the re-calibration of the output feature maps. These structures could improve the performance of CNNs significantly compared to other networks we have mentioned in Sec.4.1

We altered the configuration and hyper-parameters of the chosen baseline model to optimize its ability to extract physical features. Since the original resolution of the images is 64×64 , naturally we want to keep it unchanged when training the model. However, the formal input size of EfficientNetB4 is 380 which is a lot larger and even EfficientNetB0, its smaller version has an input size of 224. Choosing the best input size for the model is in fact reaching the compromise between time/memory cost and performance. Furthermore, in order to not introduce any distortion of the events, the input sizes of 64, 128, and 256 are tested and the nearest neighbor interpolation is used for the resizing. The comparisons is shown in Table. 3. For EfficientNetB0, increasing the resolution from 64×64 to 128×128 can improve the significance of the model by 26%, switching from B0 to B4 can further

| Stage | Operator | Input Shape | Output Shape | Repeat Count* |
|-------|-----------------|-------------|--------------|---------------|
| 1 | Conv3×3 | 128×128×3 | 64×64×48 | 1 |
| 2 | MBConv1, k3×3** | 64×64×48 | 64×64×24 | 2 |
| 3 | MBConv6, k3×3 | 64×64×24 | 32×32×32 | 4 |
| 4 | MBConv6, k5×5 | 32×32×32 | 16×16×56 | 4 |
| 5 | MBConv6, k3×3 | 16×16×56 | 8×8×112 | 6 |
| 6 | MBConv6, k5×5 | 8×8×112 | 8×8×160 | 6 |
| 7 | MBConv6, k5×5 | 8×8×160 | 4×4×272 | 8 |
| 8 | MBConv6, k3×3 | 4×4×272 | 4×4×448 | 2 |
| 9 | TopConv1×1 | 4×4×448 | 4×4×n | 1 |
| 10 | Pooling & FC | 4×4×n | 1 | 1 |

* is the number of repetitions of the module in the network. The output of one module will be the input of another consecutively.

** means this block’s convolution layers have a kernel of size 3.

Table 2: Structure of the EfficientNetB4 baseline model. Each row describes a functional block with input shape and repeat count. The default n of the last convolution is 1792.

| Network scale | Input resolution | Significance | Training time |
|-----------------------|------------------|--------------|---------------------|
| EfficientNetB0 | 64 × 64 | 6.66 | 180s/epoch |
| EfficientNetB0 | 128 × 128 | 8.39 (+26%)* | 350s/epoch (+94%) |
| EfficientNetB4 | 128×128 | 9.80 (+17%) | 800s/epoch (+128%) |
| EfficientNetB4 | 256 × 256 | 10.78 (+10%) | 3000s/epoch (+275%) |

* The percentages of improvement are compared with previous configuration.

Table 3: Cost and benefit of different network scale and image resolution.

improve it by 17%. At this configuration, it takes around 800 seconds for an epoch. However, further increasing the resolution to 256×256 costs 2.75 times more the total training time and double the memory usage with almost unchanged signal-background discrimination power. Thus, 128 is set as the input size in this work. The input and the output of the network have been modified so that it could accept the input image with the resolution 128×128 and generate a single value between 0 (background) and 1 (signal). As shown in Table 2, the default number of channels n in TopConv is 1792. Since average-pooling aims at smoothing the input by taking the average of a region while max-pooling takes only the maximum value, max-pooling will select those that it thinks are more important from a set of features instead of combining them. This is exactly what we want the model to do: to select and only select a couple of features that have physical meanings. Thus, **max-pooling** is chosen as the pooling function before the final fully connected layer. With learning rate set to be 0.001 and batch size set to be 256, our model fully utilizes the GPU’s memory (occupies ~ 32 GB), and takes about 9 hours for a training session. To predict an unseen event, the computation time is only 1.46 ms in average.

The convolutional neural networks are capable to learn the abstract features encoded in the images [38]. The visualization of the feature extraction process in EfficientNetB4 is shown to understand how the model learns the local features and the high level abstract features of the events. Lower layers is supposed to extract local features of the input images as their receptive field is still small. For example, Fig. 6 shows four representative filters from the last layer in first MBConv block (Stage 2). The complete feature maps of all 24 filters in this layer are presented in Appendix. A. The detailed and understandable local features are intuitively extracted, such as the energy deposition, the contour of the tracks, and the shape of projected tracks in YZ and XZ planes. As the number of convolutional layers deepens, the local features are integrated and strengthened layer by layer. In middle

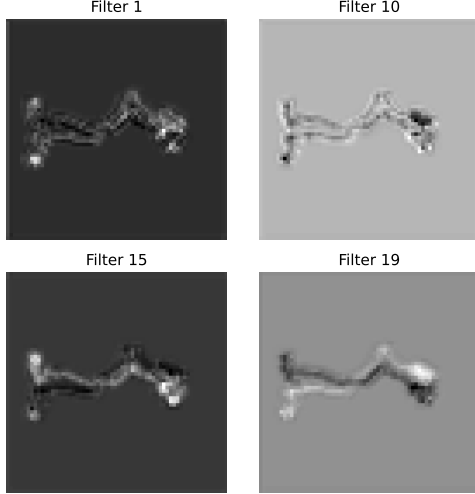


Figure 6: Example of the local feature extraction in Stage 2 (Table 2): Filter 1 detects energy depositions, Filter 10 detects contour of the tracks, Filter 15 and 19 detect the shape of projected tracks in YZ and XZ planes respectively.

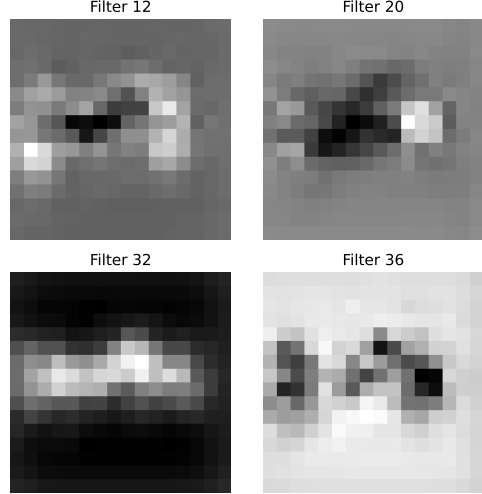


Figure 7: Example of the abstract feature extraction in Stage 4 (Table 2): Filter 12 and 20 detect energy deposition at the ends of the track, Filter 32 and 36 detect the geometric shape of the tracks.

layers, for example in the forth layer (Stage 4) as shown in Fig. 7, the tracks begin to blur as resolution decreases. But at the same time, the filters output enhanced physical features through integration of local features. Some filters try to locate the energy depositions at the end(s) of the track (Brag peak) while others work on the geometric shape of the tracks. The complete feature maps of all 56 filters in this layer are also presented in Appendix. B. These enhanced physical features are further integrated into high-dimensional physical features in subsequent layers. It is worth noting that the function of a certain filter is consistent for different input, regardless a $0\nu\beta\beta$ event or a background event.

The high-level features, which are used to direct determine the class of the input images, are obtained from the last layer of the last MBConv block (Stage 8). The high-level features have become abstract and difficult to recognize directly the specific physical characteristics of the track. However, these features originate from the local feature extraction and enhancement layer by layer in the model as shown previously in Fig. 6 and Fig. 7. Therefore, the distribution of high-level features distribution learned by the model is determined by the physical characteristics of events. In this study, it is a reasonable conjecture that the number of key global high-level features and the physical characteristics for event classification have very close correlation. The input images of tracks in this work are relatively less complex, thus the number of features needed for classification is limited. The grouping of these high-level features is implemented with a widely-used clustering algorithm called DBSCAN [39], which clusters discrete hits based on proximity. All the 448 filters of Stage 8 are automatically clustered into 5 groups (from Cluster 1 to 5) based on the similarity of feature maps as shown in Fig. 8. The feature maps of the same group have extremely high similarity as presented in Fig. 9. Therefore, a network could be then optimized by forcing the model to select a proper number of high-level features, or the number of channels in the last convolutional layers. In addition, n could be interpreted as the number of possible features embedded in the image and the optimization could be achieved by choosing a proper n corresponding to the need of the task.

At last, we analyze the weight distribution of FC (the fully connected layer in Stage 10). The distribution is shown in Fig. 10, only 6 of the weights have an absolute value larger than 0.5, including 2 positive ones and 4 negative ones. It suggests that only a small subset of its input contributes significantly to the final prediction value. To a certain extent, it is also consistent with the visualization of the similarities clustering result of the high-level

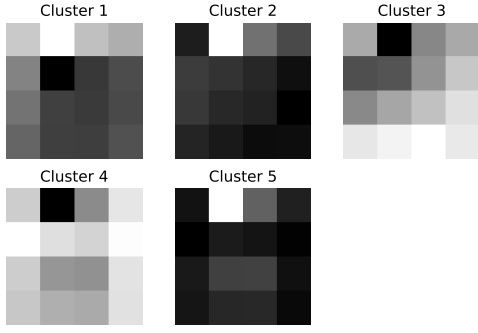


Figure 8: Clustering result of the feature maps from the last MBConv block. One example is selected for each cluster.

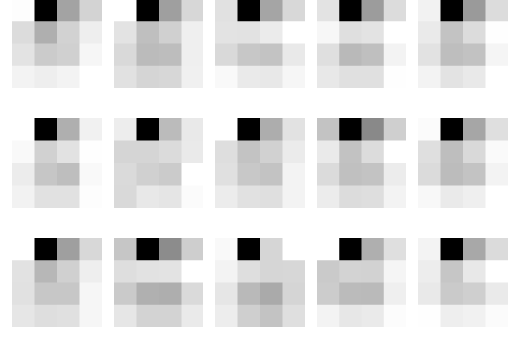


Figure 9: Feature maps of Cluster 4. There are 179 feature maps in Cluster 4, here randomly selected 15 feature maps for better visualization.

features. Thus, the principle optimization work is done by adjusting the channel number n in TopConv (Stage 9). The tested values are 2, 4, 6, 8, and 10.

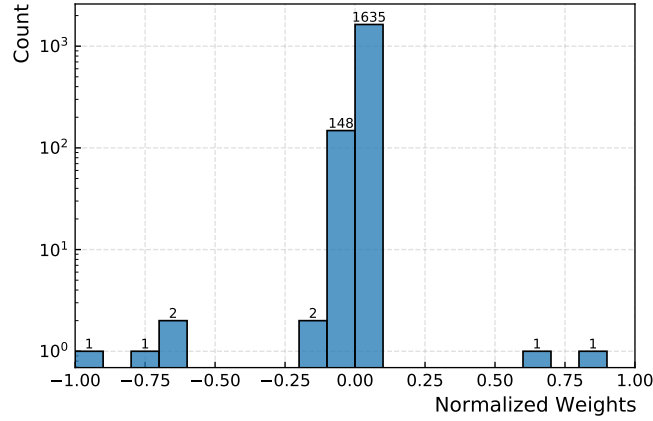


Figure 10: Distribution of the weight of the total 1792 channels in the fully connected layer.

To quantify the model's performance with a small prediction variance, we calculate the final score using the following strategy:

- only the epochs with validation accuracy are close enough ($\delta < 0.001$) to the maximum one are considered;
- for the qualified epochs, choose the weight from the first three epochs with the smallest validation loss values for prediction;
- the significances are calculated with the average prediction score from selected weights.

The network is trained independently for six training sessions to reduce the fluctuation for each value of n . The performance metrics are shown in Fig. 11. For the average significance $\bar{\Xi}$, $n = 1792$ and $n = 2$ give the lowest values as expected, as they either have an excess or insufficient number of channels. With a small but sufficient number of channels, such as 8, the significance is improved further by 10% with an average of 10.32. Such significance is resulted from a highly suppressed background rate by 3 orders, with more than 35% of signals retained.

Therefore, the result suggests that only a limited number of effective physical features are embedded in the tracks of signal and background, and that the matching between the channel number n and the number of the effective features can help to further improve the

model’s generalizability and performance. Based on this assumption, we conclude that too many channels will not help improve the model’s discrimination power. Meanwhile, too few channels will result in a lack of sufficient latent feature space for the model to discriminate between the signal and the background events.

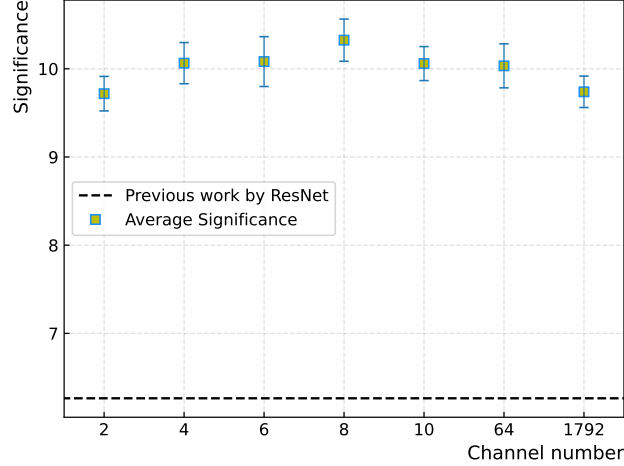


Figure 11: The significances for different models.

5 Summary

We studied the optimization of CNNs to improve the discrimination power of signal and background events in the PandaX-III experiments, based on Monte Carlo simulation data. The detector geometry is modeled according to the latest design, and the responses of the detector and electronics are simulated based on the actual parameters. A more realistic simulation dataset is generated. EfficientNetB4 is selected as a baseline model which shows a better performance among the models studied. We further explored the impact of network structure on the discrimination ability for signal and background events. By checking the feature extraction and enhancement process of our neural network, the clustering of the high level features and the weight distributions of the fully connected layer, we change the number of channels in the last convolutional layer to match the possible physical features of the tracks. The model with a small but sufficient number of channels in the last convolutional layer has a better performance, which may be an indicator for the limited number of the physical features of tracks needed for signal-background discrimination. A relatively high significance of ~ 10 is obtained, which is about 70% higher than previous work [18]. Therefore, the direction of our future work is to establish a connection between the two and improve the interpret-ability of the neural network.

Acknowledgments

This work is supported by the grant from the Ministry of Science and Technology of China (No.2016YFA0400302) and the grants from National Natural Sciences Foundation of China (No.11775142, No.11905127, and No.12175139). We thank the support from the Key Laboratory for Particle Physics, Astrophysics and Cosmology, Ministry of Education. This work is supported in part by the Chinese Academy of Sciences Center for Excellence in Particle Physics (CCEPP).

References

- [1] Matteo Agostini, Giovanni Benato, Jason A. Detwiler, Javier Menéndez, and Francesco Vissani. Toward the discovery of matter creation with neutrinoless $\beta\beta$ decay. *Rev. Mod. Phys.*, 95(2):025002, 2023.
- [2] M. Agostini et al. Final results of GERDA on the search for neutrinoless double- β decay. *Phys. Rev. Lett.*, 125(25):252502, 2020.
- [3] I. J. Arnquist et al. Final Result of the Majorana Demonstrator’s Search for Neutrinoless Double- β Decay in Ge76. *Phys. Rev. Lett.*, 130(6):062501, 2023.
- [4] D. Q. Adams et al. Search for Majorana neutrinos exploiting millikelvin cryogenics with CUORE. *Nature*, 604(7904):53, 2022.
- [5] G. Anton et al. Search for Neutrinoless Double- β Decay with the Complete EXO-200 Dataset. *Phys. Rev. Lett.*, 123(16):161802, 2019.
- [6] S. Abe et al. Search for the Majorana Nature of Neutrinos in the Inverted Mass Ordering Region with KamLAND-Zen. *Phys. Rev. Lett.*, 130(5):051801, 2023.
- [7] P. Novella et al. Measurement of the Xe136 two-neutrino double- β -decay half-life via direct background subtraction in NEXT. *Phys. Rev. C*, 105(5):055501, 2022.
- [8] Juan J. Gomez-Cadenas, Francesc Monrabal Capilla, and Paola Ferrario. High Pressure Gas Xenon TPCs for Double Beta Decay Searches. *Front. in Phys.*, 7:51, 2019.
- [9] Xun Chen et al. PandaX-III: Searching for neutrinoless double beta decay with high pressure ^{136}Xe gas time projection chambers. *Sci. China Phys. Mech. Astron.*, 60(6):061011, 2017.
- [10] R. Luscher et al. Search for beta beta decay in Xe-136: New results from the Gotthard experiment. *Phys. Lett. B*, 434:407–414, 1998.
- [11] P. Ferrario et al. First proof of topological signature in the high pressure xenon gas TPC with electroluminescence amplification for the NEXT experiment. *JHEP*, 01:104, 2016.
- [12] J. Galan et al. Topological background discrimination in the PandaX-III neutrinoless double beta decay experiment. *J. Phys. G*, 47(4):045108, 2020.
- [13] Tao Li, Shaobo Wang, Yu Chen, Ke Han, Heng Lin, Kaixiang Ni, Wei Wang, Yiliu Xu, and An’ni Zou. Signal identification with Kalman Filter towards background-free neutrinoless double beta decay searches in gaseous detectors. *JHEP*, 06:106, 2021.
- [14] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91(4):045002, 2019.
- [15] A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M. D. Messier, E. Niner, G. Pawloski, F. Psihas, A. Sousa, and P. Vahle. A Convolutional Neural Network Neutrino Event Classifier. *JINST*, 11(09):P09001, 2016.
- [16] P. Abratenko et al. Convolutional neural network for multiple particle identification in the MicroBooNE liquid argon time projection chamber. *Phys. Rev. D*, 103(9):092003, 2021.
- [17] M. Kekic et al. Demonstration of background rejection using deep convolutional neural networks in the NEXT experiment. *JHEP*, 01:189, 2021.
- [18] Hao Qiao, Chunyu Lu, Xun Chen, Ke Han, Xiangdong Ji, and Siguang Wang. Signal-background discrimination with convolutional neural networks in the PandaX-III experiment using MC simulation. *Sci. China Phys. Mech. Astron.*, 61(10):101007, 2018.
- [19] K. J. Kang, J. P. Cheng, Y. H. Chen, Y. J. Li, M. B. Shen, S. Y. Wu, and Q. Yue. Status and prospects of a deep underground laboratory in China. *J. Phys. Conf. Ser.*, 203:012028, 2010.
- [20] Jainmin Li, Xiangdong Ji, Wick Haxton, and Joseph S. Y. Wang. The second-phase development of the China JinPing underground Laboratory. *Phys. Procedia*, 61:576–585, 2015.

- [21] S. Andriamonje et al. Development and performance of Microbulk Micromegas detectors. *JINST*, 5:P02001, 2010.
- [22] Shaobo Wang. The TPC detector of PandaX-III Neutrinoless Double Beta Decay experiment. *JINST*, 15(03):C03052, 2020.
- [23] Chen Xie, Kaixiang Ni, Ke Han, and Shaobo Wang. Enhanced search sensitivity to the double beta decay of ^{136}Xe to excited states with topological signatures. *Sci. China Phys. Mech. Astron.*, 64(6):261011, 2021.
- [24] S. Agostinelli et al. GEANT4—a simulation toolkit. *Nucl. Instrum. Meth. A*, 506:250–303, 2003.
- [25] Konrad Altenmüller et al. REST-for-Physics, a ROOT-based framework for event oriented data analysis and combined Monte Carlo response. *Comput. Phys. Commun.*, 273:108281, 2022.
- [26] O. A. Ponkratenko, V. I. Tretyak, and Yu. G. Zdesenko. The Event generator DECAY4 for simulation of double beta processes and decay of radioactive nuclei. *Phys. Atom. Nucl.*, 63:1282–1287, 2000.
- [27] Heng Lin et al. Design and commissioning of a 600 L Time Projection Chamber with Microbulk Micromegas. *JINST*, 13(06):P06012, 2018.
- [28] A. D. McDonald et al. Electron Drift and Longitudinal Diffusion in High Pressure Xenon-Helium Gas Mixtures. *JINST*, 14(08):P08009, 2019.
- [29] H. Schindler and R. Veenhof. Garfield++: simulation of ionisation based tracking detectors. 2020. <https://garfieldpp.web.cern.ch/garfieldpp/>.
- [30] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 37:448–456, 07–09 Jul 2015.
- [31] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. pages 2261–2269, 2017.
- [32] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. 97:6105–6114, 09–15 Jun 2019.
- [33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 2016.
- [34] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [35] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.
- [36] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, and Quoc V. Le. Mnasnet: Platform-aware neural architecture search for mobile. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2815–2823, 2019.
- [37] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.
- [38] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. <https://distill.pub/2017/feature-visualization>.
- [39] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD’96*, page 226–231. AAAI Press, 1996.

A The features of Stage 2

In the baseline model, the lower layers in the network learn local detailed features. For example, Fig. 12 the 24 filters of the first layer. The detailed physical features such as the energy deposition, the tracks in XZ and YZ planes, the geometric shape and edge of the track are extracted.

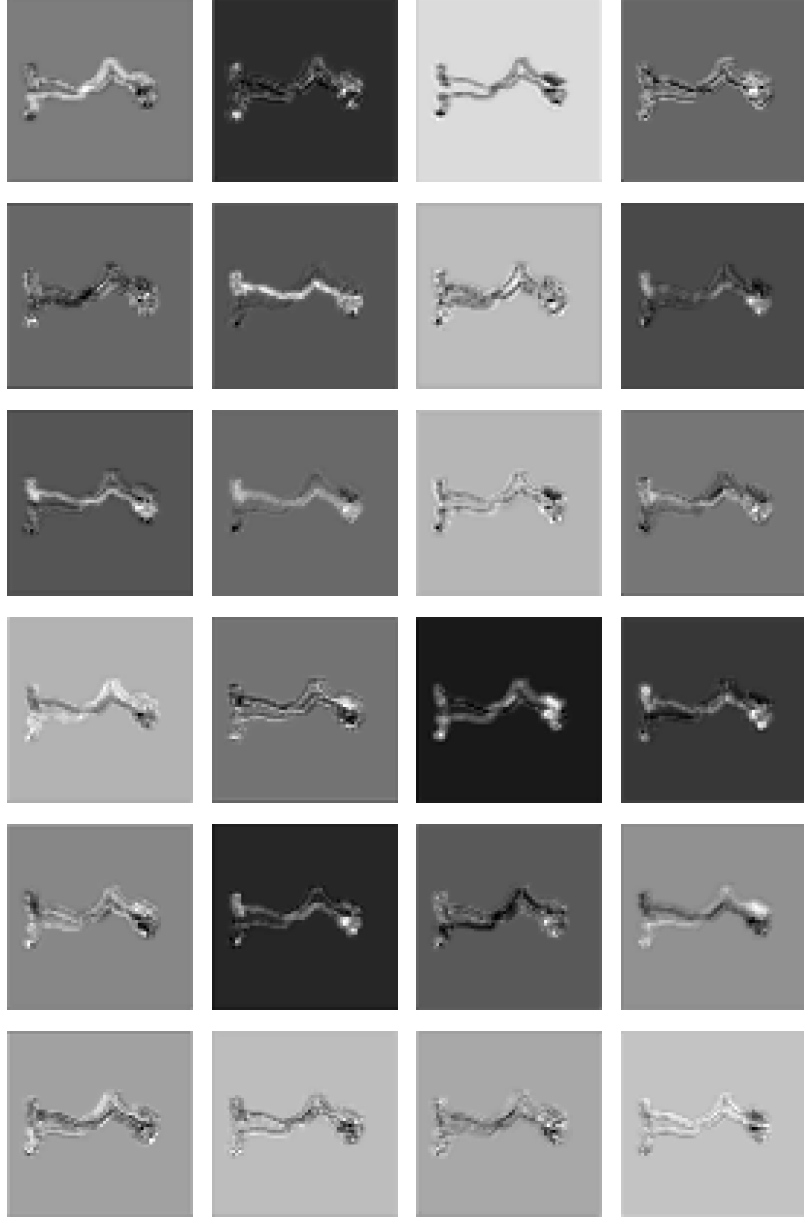


Figure 12: Example of the local features extraction in the first layer. All 24 filters are shown, and each filter extracts different local features of the tracks.

B The features of Stage 4

In the middle layers, the physical features are integrated and strengthened layer by layer. Fig. 13 shows all the 56 filters of the third layer, the tracks begin to blur as resolution decreases. The filters are activated locally on the tracks and the local features such as the energy deposition at the ends of the tracks, the geometric shape of the tracks are enhanced.



Figure 13: Example of the features enhancement in the third layer. All 56 filters are shown, and each filter tries to detect different enhanced local features of the tracks.