π^0/γ identification with the FCC-ee LAr ECAL

Pavlo Panasiuk[2,1] **, Brieuc Francois**[1]

[1] *CERN*

[2] *Taras Shevchenko National University of Kyiv, Ukraine*

Abstract: The FCC-ee is a lepton collider that will operate in a 100 km circular tunnel and will consist in the first phase of the FCC integrated program proposed as the post-LHC CERN flagship experiment. Several aspects of the FCC-ee physics program rely on strong particle identification capabilities among which single photon and neutral pion originated electromagnetic shower separation is of utmost importance. This document reports on the π^0/γ separation performance of the Noble Liquid electromagnetic calorimeter proposed for FCC-ee. Different detector geometries and machine learning approaches are investigated. A neutral pion mis-identification probability of 10% for a 95% photon efficiency working point is achieved with a regular geometry and a Hybrid Neural Network approach.

Key words: Particle identification, FCC-ee, Z factory, machine learning, computer vision, dense/convolutional neural networks

Contents

1 Monte Carlo setup **5**
- 1.1 Generated Samples 5
- 1.2 Training and validation datasets 5

2 High-level variables with separation power **7**
- 2.1 List 7
- 2.2 Visuals 8

3 Fully connected neural network **13**
- 3.1 Introduction 13
- 3.2 Layout and setup 15
- 3.3 DNN's performance 16
- 3.4 Rejection comparison: FCC-hh DNN 20

4 Advanced networks **21**
- 4.1 Introduction 21
- 4.2 Layout and setup 22
- 4.3 Performance 24

5 Summary **28**

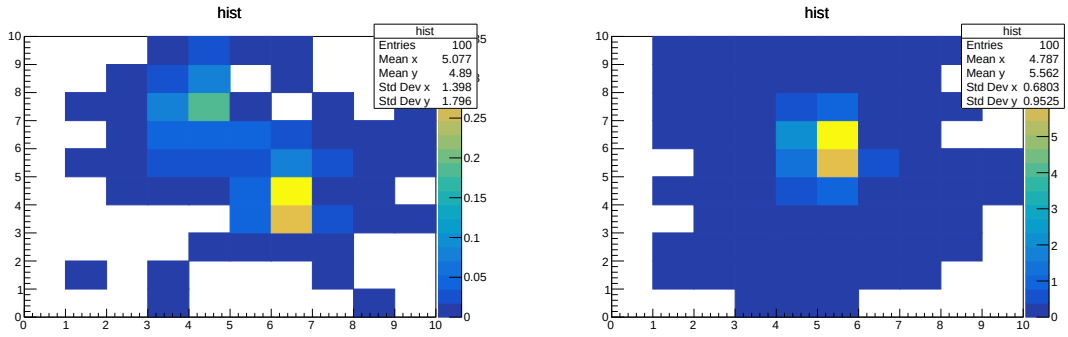
Introduction

Event reconstruction

When it comes to detecting accelerator outputs, one must always bear in mind it is only possible to detect few (semi-)stable particle species, therefore leading to a loss of intermediate processes information. If one possesses information about possible processes (and it is usually the case), it is only natural to strive against this loss by building *event reconstruction tools*, capable of regaining some intermediate information.

While many processes can be easily distinguished by charges, simple variable cuts, etc, some lead to detector signatures that are tricky to identify and which require advanced algorithms.

An example of such hard case is the distinction between photon pairs originating from a particle decay (e.g. $\pi^0 \to \gamma\gamma$) and single photon production: at high energy, the 2γ from the π^0 decay are emitted with little angular separation, what leads to their showers overlapping, replicating the single photon detector signature. However, even on direct inspection, it is sometimes possible to identify "easy" signatures (see, e.g., Fig. 1, with double-peaked structure).



(a) π^0 event

(b) γ event

Figure 1: Comparison of π^0/γ unrelated events in FCC-ee LAr ECal simulation's 5^{th} layer. In this easy case one can easily separate one particle from another by considering two photon peaks on figure (1a).

Motivation behind $\gamma\gamma$ vs γ separation

Tools able to "count" final-state photons generally speaking have a broad application within within various precision measurements and BSM searches.

As for one example, one could benefit from such tool in searches for ALP [1] (see Fig. 2), where the ability to count separate photons is essential.

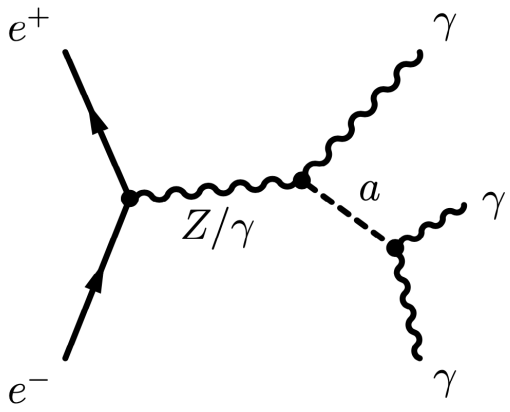


Figure 2: $e^+e^- \to Z(\gamma) \to Z(\gamma)a \to Z(\gamma)\gamma\gamma$ from MG5 analysis tool.

Such a reconstruction algorithm will also help to reduce background events with jets leading to isolated π^0s in analyses involving isolated photons signature.

Another example of analysis benefiting from high π^0/γ separation power is τ polarization measurement, where sensitivity is greatly enhanced by categorization of final τ states [2]. Indeed, as shown on Fig. 3, many non negligible tau decay modes feature one or more π^0.

Decay modes	Branching fraction [%]
$e^- \bar{\nu}_e \nu_\tau$	17.82 ± 0.04
$\mu^- \bar{\nu}_\mu \nu_\tau$	17.39 ± 0.04
$h^- \nu_\tau$	11.51 ± 0.05
$h^- \pi^0 \nu_\tau$	25.93 ± 0.09
$h^- 2\pi^0 \nu_\tau$	9.48 ± 0.10
$h^- 3\pi^0 \nu_\tau$	1.18 ± 0.07
$h^- 4\pi^0 \nu_\tau$	0.16 ± 0.04
3 prongs	15.20 ± 0.06

Figure 3: τ branching ratios predicted from Standard Model. Ability to count separate π^0s is essential here. Credits to [2].

As for this project we specifically aim to reach high quality separation between π^0 and γ events in the simulation framework of planned FCC-ee Liquid Argon (LAr) electromagnetic calorimeter.

Current report is organized as follows. In chapter 1 we discuss samples, generated for training and testing purposes. In chapter 2 we introduce variables which proved useful in the context of particle separation. In chapter 3 we comment on performance

of fully connected neural net, trained with use of high-level variables from 2. Similar discussion is conducted in chapter 4, but for more complex (and, at the same time, more problem-specific) neural net layouts. Summary is given in chapter 5.

1 Monte Carlo setup

1.1 Generated Samples

All samples were generated based on a LAr ECAL geometry similar to the one used in FCC-hh studies described in [3] but adapted for FCC-ee. Differences include number of radial layers (12 for our case), shortened calorimeter radius and others, insignificant for current discussion. For simplicity, all the events are produced with the help of **GEANT4** particle gun (emitting π^0/γ's), uniform in initial particle energy $\in [1, 100]$ GeV.

1.2 Training and validation datasets

For us to "train" the separation tool to actually distinguish particles, we are first required to produce "training" datasets, i.e. datasets that are used for tuning weights in the tool. More details on this can be found in chapter 3.

As for now, important details can be confined to the following:

- ◇ 50k (out of total 100k; see further the reason) events per particle
- ◇ No noise included
- ◇ Geometries with/without 2^{nd} layer geometries (explanation follows)

As a separate development, we consider two rather similar geometries with, however, crucial difference. While a geometry w/o strip layer (interchangeably referred as to uniform celling geometry) denotes celling with uniform size in η and ϕ ($\Delta\eta = 0.01$ and $\Delta\phi \approx 8$ mrad), a geometry w/ strip 2^{nd} layer uses finer (4x) resolution in η along the 2^{nd} radial layer (see Fig. 4). This distinction is very useful for particle separation, since it allows to very precisely render particle's shower in the beginning. This of course adds complexity to the detector (4x readout channels from 2^{nd} layer), but at the same time it is less severe than introducing these fine granularity cells everywhere in the detector.

In order to check overall performance, exclude possible issues like overtraining and fine-tune hyperparameters, we apply the trained model to a "validation" dataset, i.e., remaining fraction of events that we excluded from training. This allows to make sure the model did not acquire any features that are specific to training set exclusively, and therefore is capable to do meaningful predictions. It is also useful for adjusting model's layout.

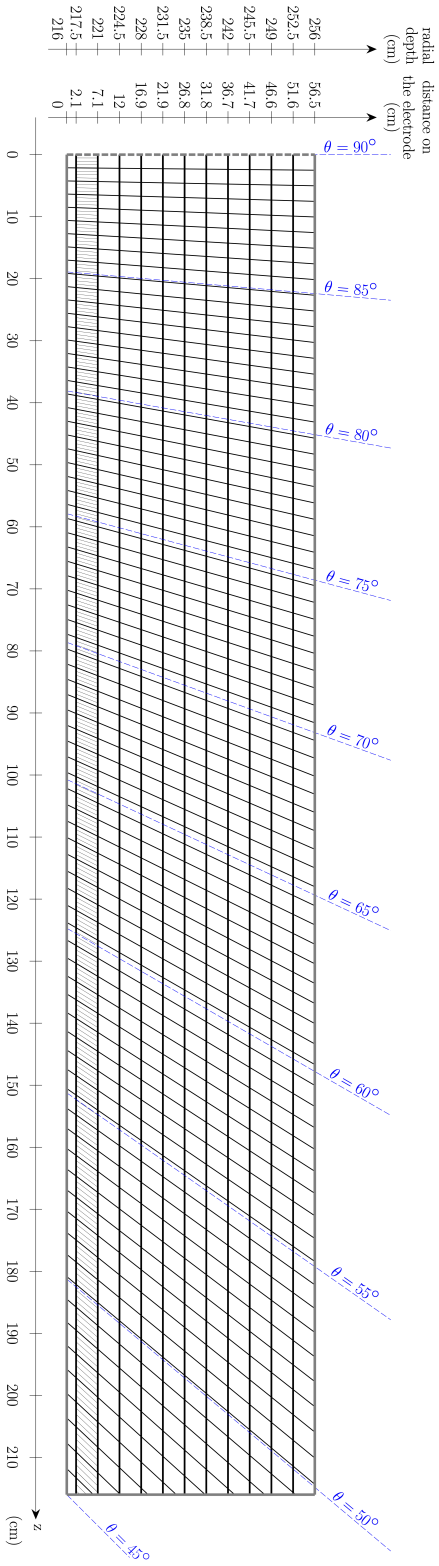


Figure 4: Readout scheme for geometry with strip layer.

2 High-level variables with separation power

A common approach (apart from "trivial variable cuts", which does not fit best for π^0/γ separation purposes) is to convolute detector's information into a few distinctive variables.

In this section we discuss and construct kinematic variables bringing separation power between single photon and double photon events.

2.1 List

Analysis concerning search and discriminativity check was thoroughly performed in [3] regarding FCC-hh. Most of them are satisfactory for our purposes.

Here follows the **full** list of variables we probed. A single asterisk denotes "*of the most energetic cluster*":

- ◇ Cluster count : Number of clusters above 400 MeV in an event. Resolving at very low energies, when π^0 is not that much boosted and therefore may leave energy deposit in two separate clusters.
- ◇ Energy (*) : Overall energy deposited in the cluster. Useful for distinguishing easy cases from hard cases (we discuss it later in a sub-chapter 2.2)
- ◇ Maximum cell's energy in a given layer (*) : A little discriminating at the whole energy range. For our purposes, we consider 2^{nd} layer.
- ◇ $E^{2nd-max} - E^{min}$ in 2^{nd} layer (*), where E^{min} is searched within a box enclosed by 1st and 2nd energy maxima : Telling variable when there's a double-peaked structure of roughly same height. Mid-to-high energy resolving.
- ◇ Fraction of energy in a given layer (*) : Self-explanatory. We take into account $2^{nd}, 4^{th}, 10^{th}$ layers (even though last ones may be similar, they may grant slightly different information). Whole-range resolving.
- ◇ Angular eccentricities in given layer (*) : Eccentricity of a shower calculated via variance matrix with energy weighting (uniform weighting proves to work worse). Low-to-top energy resolving.
- ◇ Shower core-relative variance in a given layer (*) at a given offset from core : $E_{ocore} = 1 - \dfrac{\text{sum of deposits}(\pm\,\text{offset})}{\text{sum of deposits}(\pm\,1)}$, where sum of deposits signify that we sum all the energy around cell with maximum energy at specified ϕ, η bin offset on squared trajectory. We consider (layer, offset) pairs of (2,2) and (2,3), since they prove to be the most discriminative. Slightly resolving at mid-to-top energies.

◇ Shower bin variance in a given layer (*) at a given offset from core in η : Energy averaged quantity of $(\phi-\phi_0+\eta-\eta_0)^2$ among cells within box $[\phi-\phi_0, \eta-\eta_0] \in [(-1,1),(-\text{offset},\text{offset})]$. (2,2) and (6,2) combinations are used.

Variables from above mostly intersect with FCC-hh document (or they may be correlated with them). Here follow additional entries to this list which arose from trial and error process (mostly for better discrimination at low-mid energies)

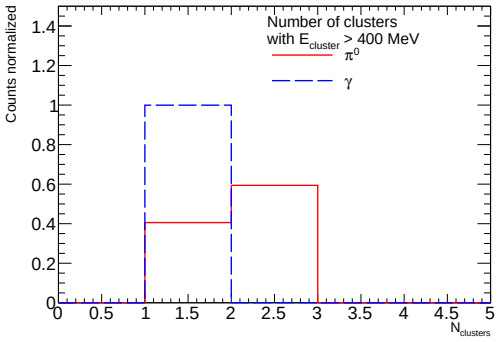
◇ Invariant mass cell-/cluster-based: Calculated on the basis of 4-momenta of all cells/clusters, extrapolated from the interaction point. Due to finite detector granularity, this variable gets worse with energy grow; cell-based version, though we believe it does not carry independent physical sense, is able to separate particles up to mid-energies, and at the same time very capable at very low energies.

◇ Amount of cells (*) : For this variable we take all layers into account. Slightly distinctive at whole energy range.

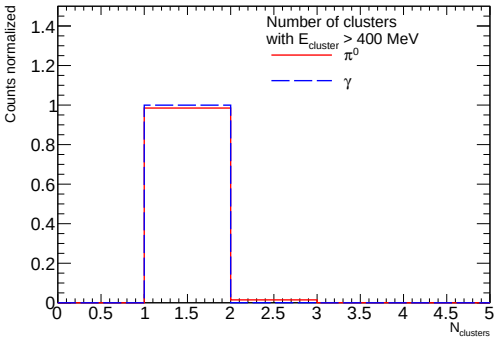
2.2 Visuals

In this sub-chapter we discuss some plots of variables, discussed in sec. 2.1, at selected energy ranges (of generated particle). This should grant some insight about capabilities of the separation tool we have assembled. Unless else specified, figures are presented for the geometry w/o strip layer.

It makes sense to start off with low-energy range, since it is the simplest case for us. It should be mentioned here that we benefit from two different clustering algorithms: one based on the localisation of local minima and a fixed size window (*Sliding Window Algorithm*) and one dynamically building clusters based on high energy seed cells and using only cells exceeding a certain energy threshold (*Topological Clustering Algorithm*). For different variables different algorithms can be used, based on their relative discriminative power. Figures (5, 6, 7) reflect our general low-energy capabilities.

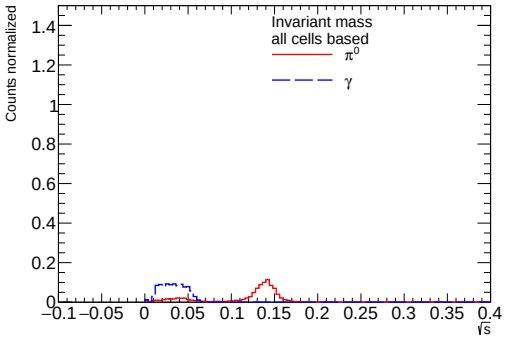


(a) $\{1, 5\}$ GeV

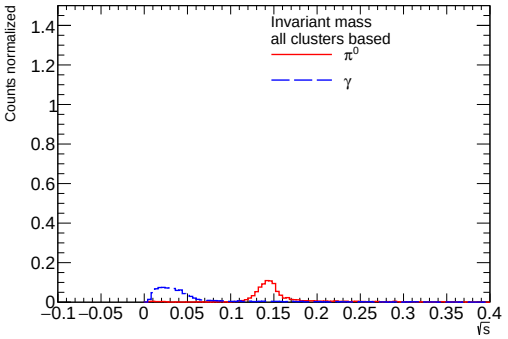


(b) $\{5, 10\}$ GeV

Figure 5: Number of clusters in topological clustering scheme for specified ranges. As we observe, this variable loses its discriminative ability already at 5 GeV because both photons from a π^0 decay end up in the same cluster.



(a) cells based

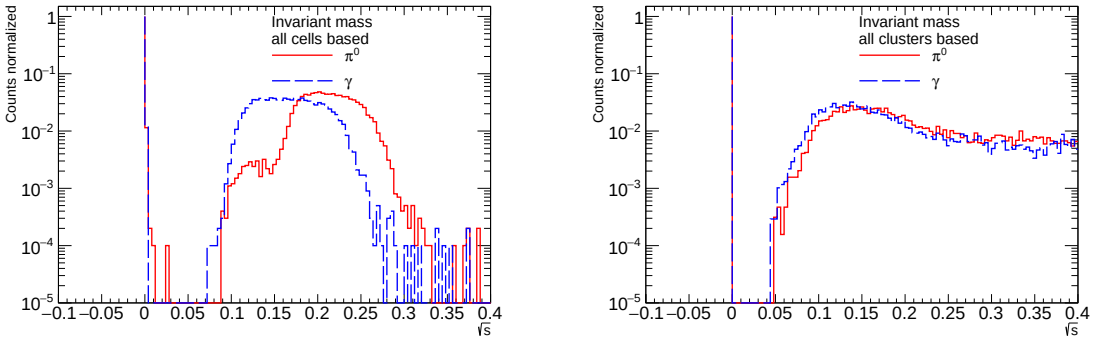


(b) clusters based

Figure 6: Invariant mass, calculated with use of different methods, in $\{1, 5\}$ GeV range. We note very similar (strong) separation power.

Further analysis gets more complex due to π^0 being boosted enough to almost certainly land both decay photons into a single cluster (exactly what we observe at Fig. 5b). We still, though, are able to do some separation (less performant, of course, in comparison to low energies), but this requires many different variables within most energetic cluster, including various variances, layer energy fractions, two-peaks-valley variables etc. Figures 8 and 9 represent our abilities to separate π^0 and γ at mid-energy region, when it is barely possible to do so without advanced algorithms.

Hardest cases occur at highest available generated energy, where it is sometimes impossible to find any structural difference, like double peak. Even in such cases, some variable are still partially able to perform at least low-quality separation (see Fig. 10, 11, 12)



(a) cells based (b) clusters based

Figure 7: Invariant mass, calculated with use of different methods for strip layer geometry, in $\{10, 20\}$ GeV range. Cells based invariant mass proves to be way more superior up to mid-energies when compared against cluster based version.

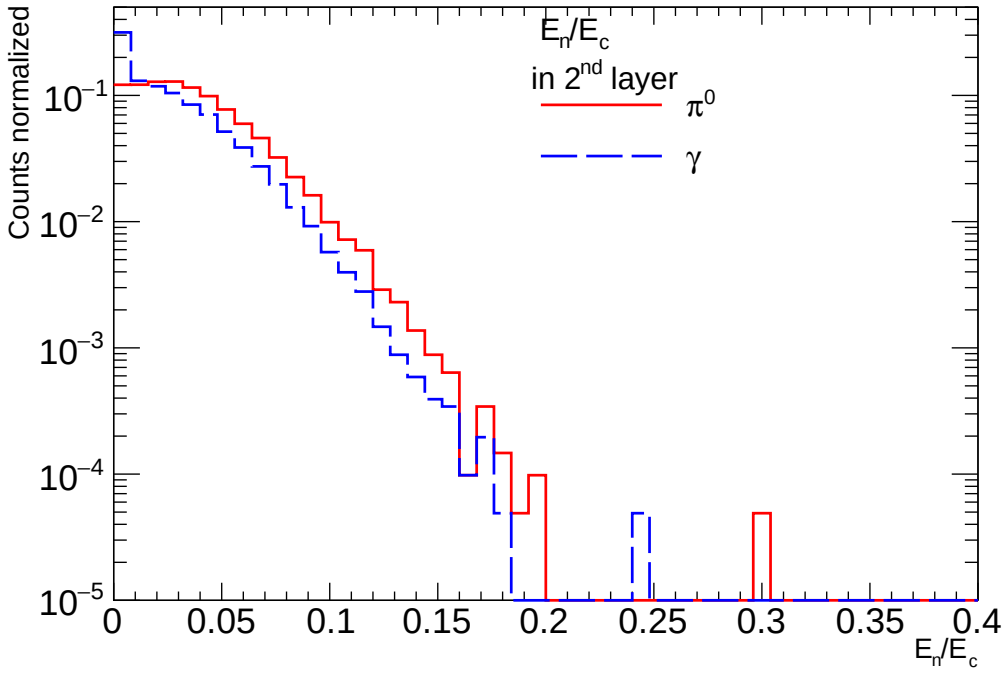


Figure 8: 2^{nd} layer energy fraction for $\{20, 40\}$ GeV.

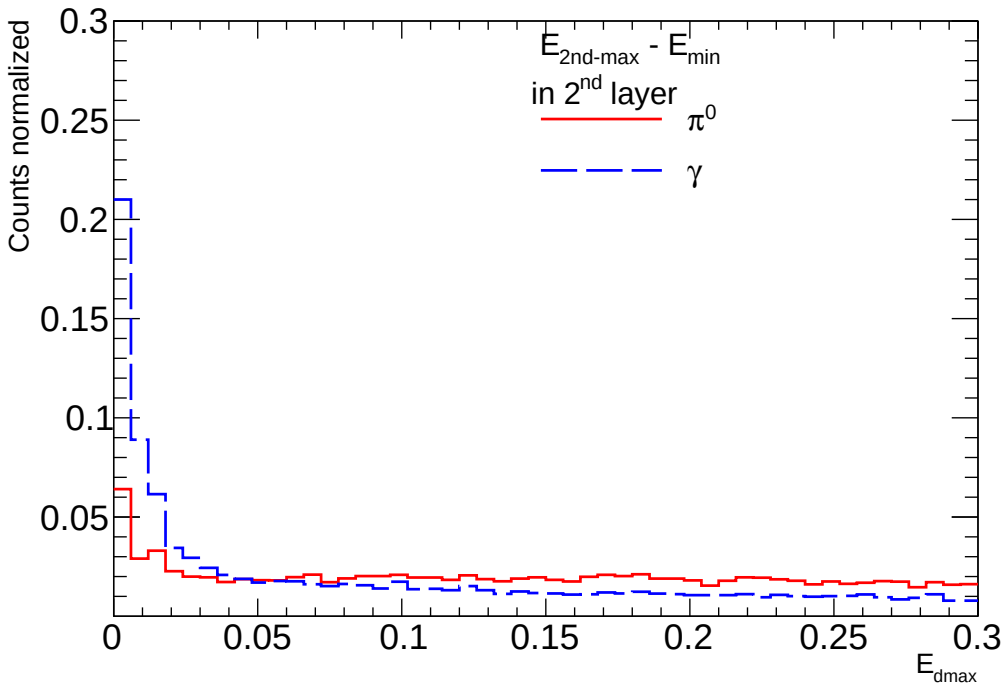


Figure 9: Valley energy difference for $\{40, 60\}$ GeV.

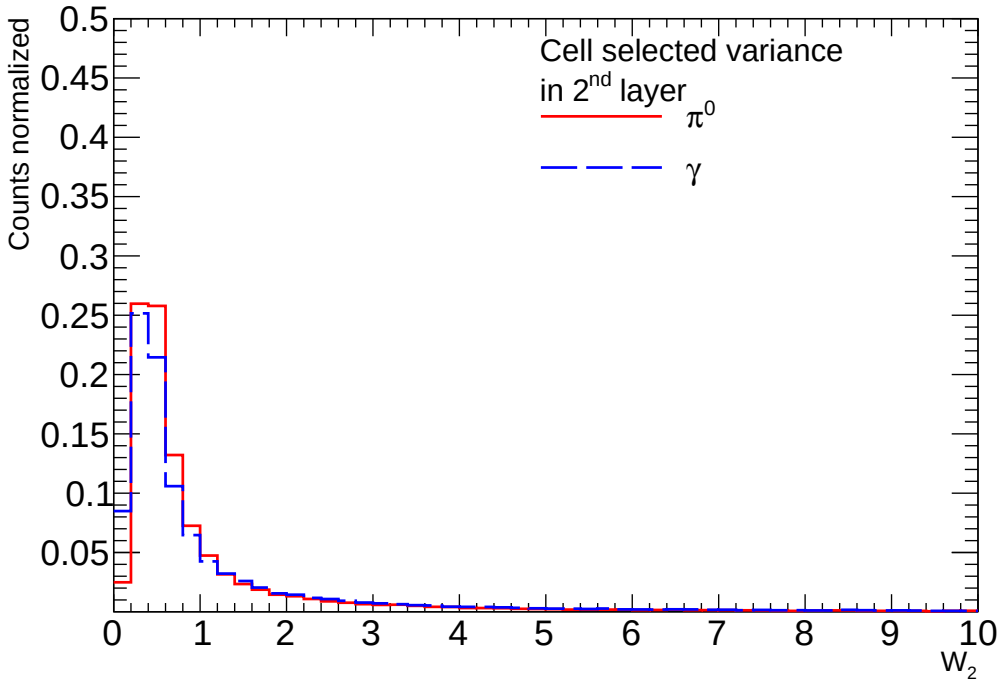


Figure 10: Shower width variance for $\{60, 100\}$ GeV.

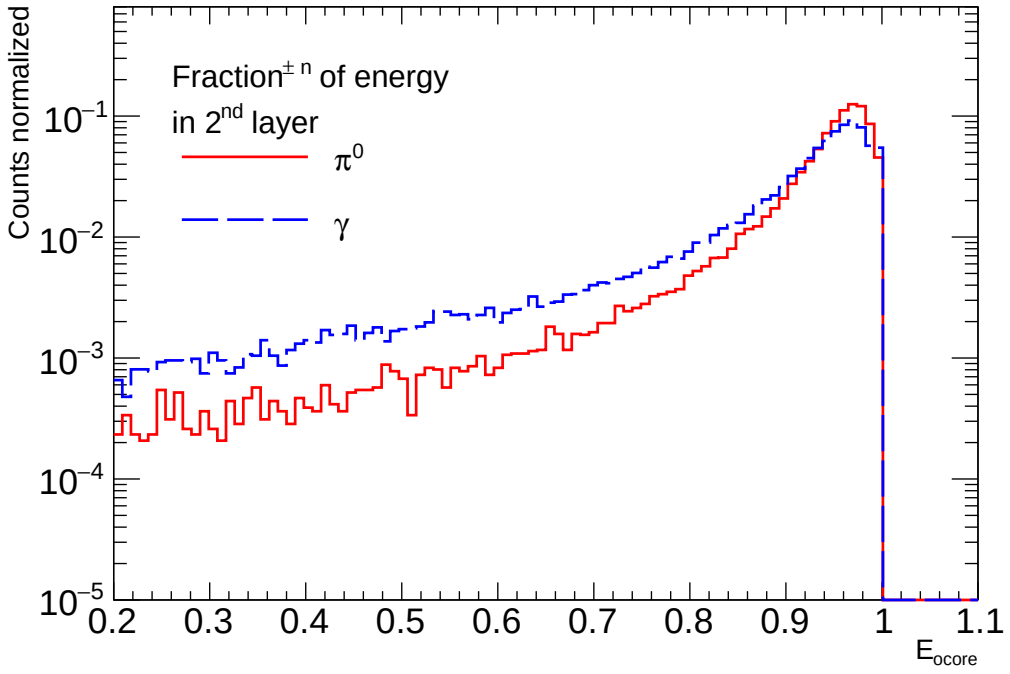


Figure 11: Shower core-relative variance for $\{60, 100\}$ GeV.

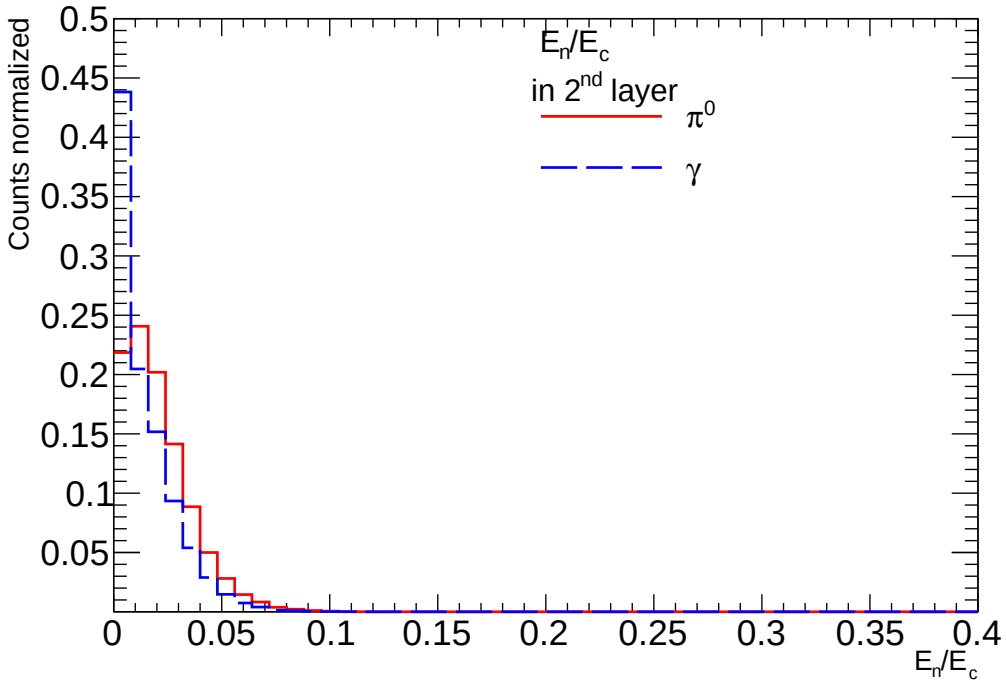


Figure 12: 2^{nd} layer energy fraction for $\{60, 100\}$ GeV.

3 Fully connected neural network

With the insights brought from Chapter 2, one may start wondering, how exactly those variables are supposed to work in a single mechanism. While some, like invariant mass, could be used as a sort of variable cut, we usually struggle to cross a definitive line between particles signatures. Therefore we can safely assume that in this case we are unable to build a human-understandable model. Instead, we see grounds for application of neural network.

3.1 Introduction

Fully connected neural network (a.k.a. dense neural network, FCNN/DNN for short) is a type of neural network working with single-valued data types in a specific way.

Some terminology simplified:

- ◇ Neural Network – complex (usually nonlinear) regression/classification fit model with a substantial amount of weights, ruled by a loss function (e.g. χ^2)
- ◇ Layer – intermediate placeholder for data. It directly affects further layers content
- ◇ Neuron – a single entry within a layer. Every neuron has weights associated to it and previous layer, what defines layer to neuron transformation law.

◇ Fully connected neural network – neural network built on the basis that every single neuron in current layer depends on every single neuron from previous layer (therefore the name "dense", since we have a lot of links; see Fig. 13)

◇ Activation function – a nonlinearity source in the fit model. Since every single layer to neuron transformation law is actually a simple dot product, stack of dense layers could be interchanged with a single dense layer, what limits the models capabilities a lot. To strive against it, usually, this transformation law gets composed with a nonlinear so-called activation function. I.e.

$$x_j^{\text{layer}=i} = A(\vec{W}_j \cdot \vec{x}^{\text{layer}=i-1}) \tag{3.1}$$

◇ Epoch – a single iteration over training set. Mainly includes forward propagation (calculating model's output based on weights and inputs) and backward propagation (updating model's weights with help of minimizer).

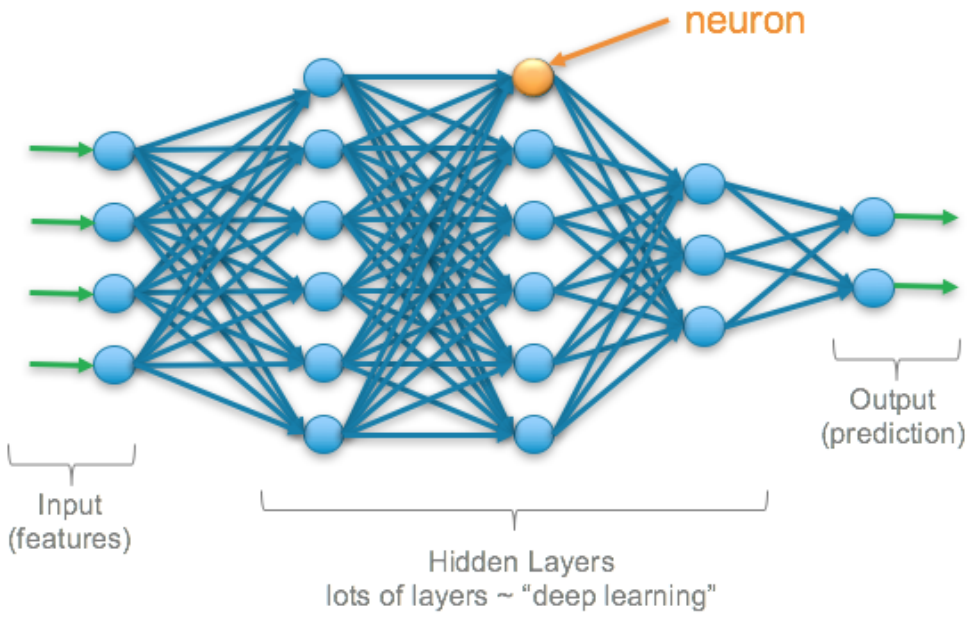


Figure 13: Usual FCNN

With this in mind, in this chapter we present our achievements with DNN, based on variables from chapter 2.1.

3.2 Layout and setup

The used layout itself can be found on Fig. 14; it simply consists of several sequential dense layers (without dropout). This layout proves to perform in stable fashion and does not accumulate overtraining (i.e. when a net learns features specific to training dataset only, like noise) for our case, what makes it very convenient to stick to. Model was compiled with

- ◇ Adam optimizer
- ◇ Binary crossentropy loss function
- ◇ Accuracy metrics

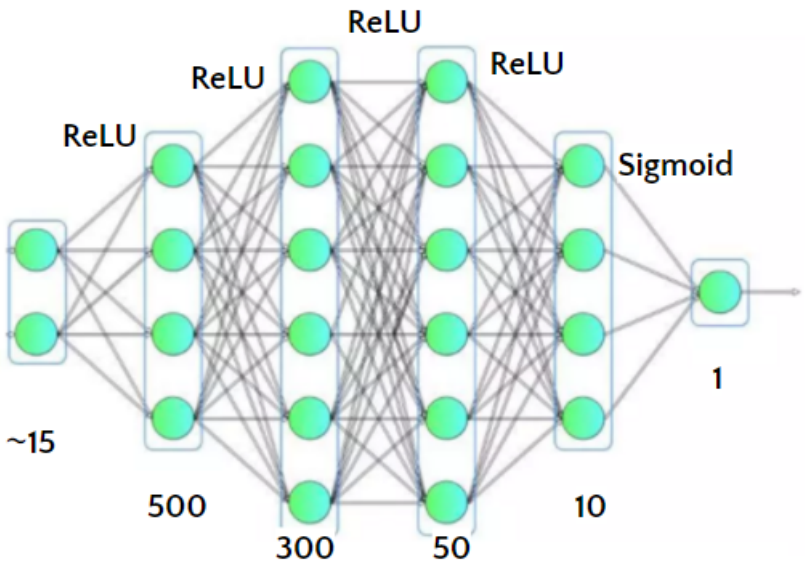


Figure 14: DNN layout used for all tries. Changes to it did not prove to change prediction quality drastically.

All the 2×100k events each containing columns with discriminative variables were split in half for training and validation purposes (later we usually refer to them as to train/test sets). We set the so-called "pionity" variable as an output label. It ideally equals 1 for a pion, 0 for a photon, and it is something in between when model makes prediction.

It is important to discuss how we rate model's efficiency. Each model has its averaged precision estimate, i.e. fraction of properly recognized events, which could serve a purpose. It is, however, not so informative: it does not capture the energy dependence of the separation power, and it cuts events at predefined pionity = 0.5. With a bit more of work, we are able to choose our own pionity cut for different-category events.

In short, our success metrics is as follows

- ◇ Pionity distributions : Density of events recognized at given pionity $\frac{dN_{\text{ev}}}{d\pi^0_{\text{ty}}}$. Useful for visualising overtraining, when train and test sets outputs are superimposed. The more different particle distributions are separated the better model is.
- ◇ Receiver operating characteristic (ROC) curve : efficiency/efficiency curve, parameterized with pionity. The ultimate metric in a sense that if one model has higher ROC than the other one in every single point then the first one is ultimately better.
- ◇ π^0 rejection factor (at 90% γ efficiency): $\frac{1}{1-\pi^0_{\text{eff}}}$. More compact in comparison to ROC, therefore allows for multiple sub-ranges analysis simultaneously.

The information these metrics carry is similar to some extent, however they all are useful for better understanding and visualisation.

In the following sub-chapter we discuss our successes and failures with DNN applied to different geometries.

3.3 DNN's performance

Variable selection (17 total), described in Sec. 2.1, proved to be the most optimal for our case. They therefore form input layer for both geometries.

In order to recognize our sub-range capabilities (specifically, in initial energy), it is fruitful to compare ROC curves for several energy ranges.

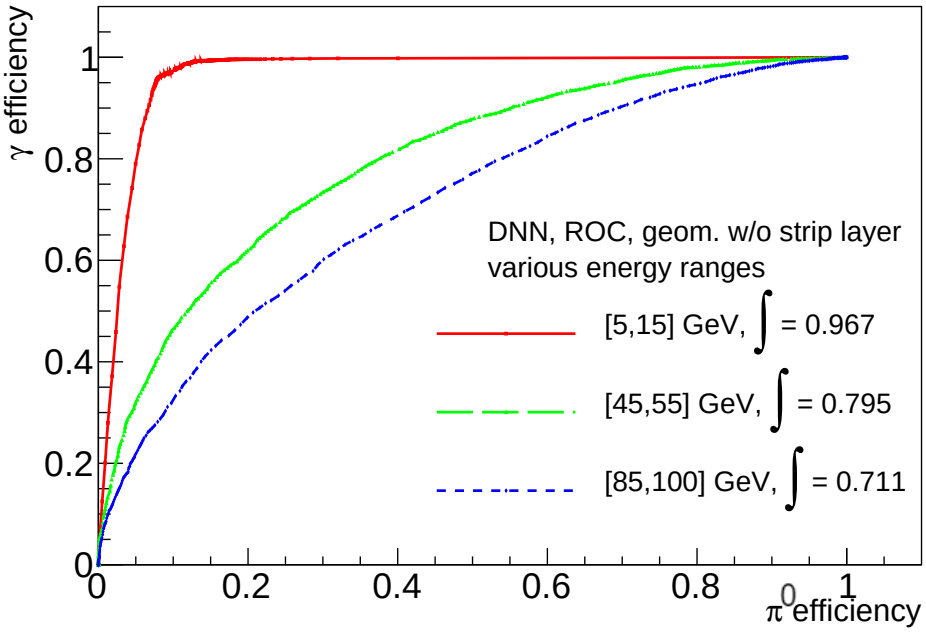
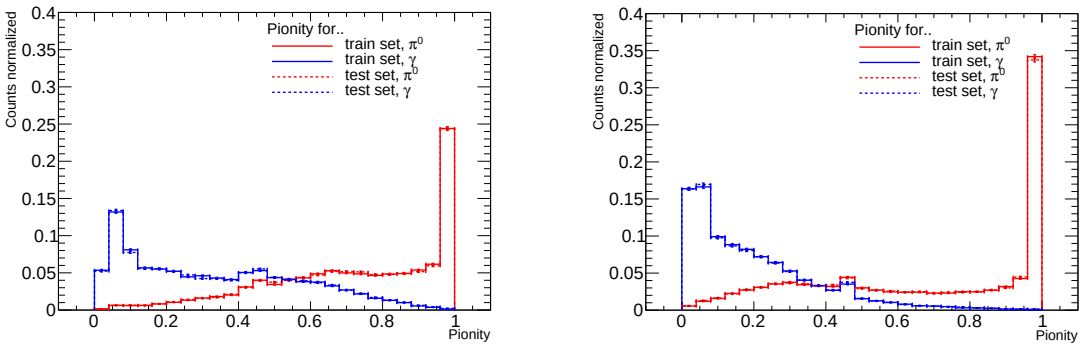


Figure 15: Geometry w/o strip layer DNN's ROC for low, mid and high energy ranges.



(a) uniform celling

(b) strip 2^{nd} layer

Figure 16: Inclusive pionity distribution comparison for different geometries. Absense of overtraining is notable (same curve behaviour for train and test sets within error).

Fig. 15 shows clear degrading of separation quality with increase of energy, which is exactly expected. It is still remarkable, though, that the DNN is able to reach 71% of ROC integral value in the worst case. In upcoming chapters we will discuss how to achieve an even better separation at high energies.

At this point we also probed strip layer geometry on top of these variables for comparison. On average, 20 epochs training grant us with 76% and 80% recognition without

overtraining accumulation for geometries with and without strip layer geometry respectively (see Fig. 16 and specifically Fig. 17 for inclusive comparison).

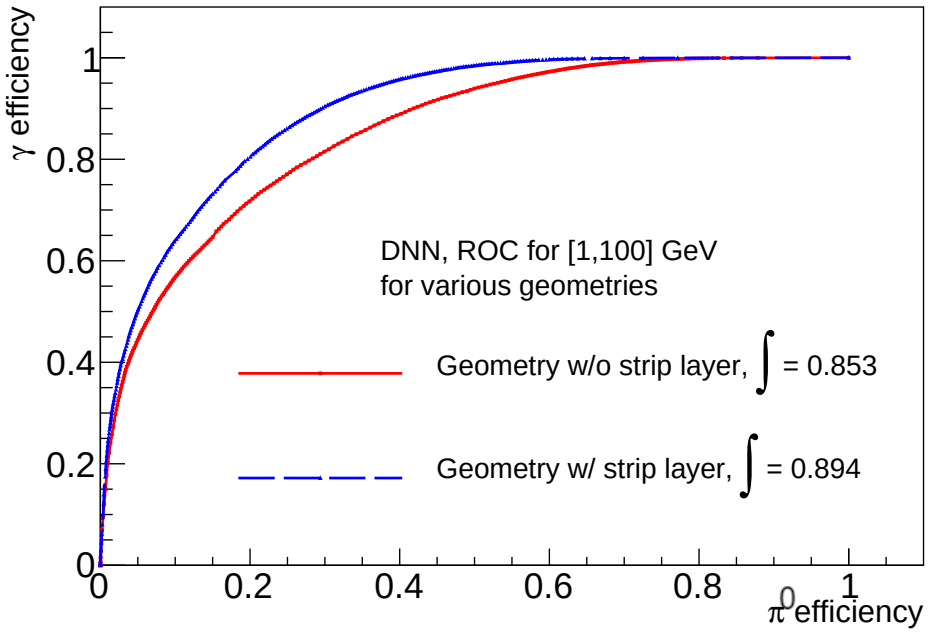
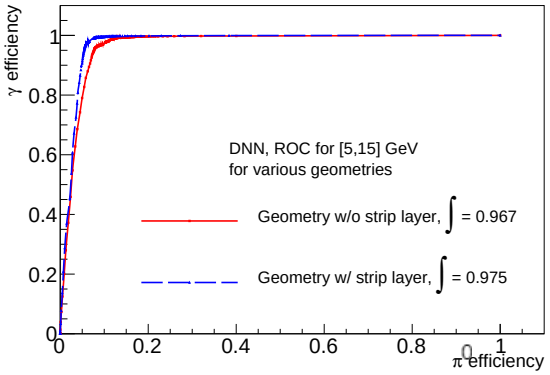
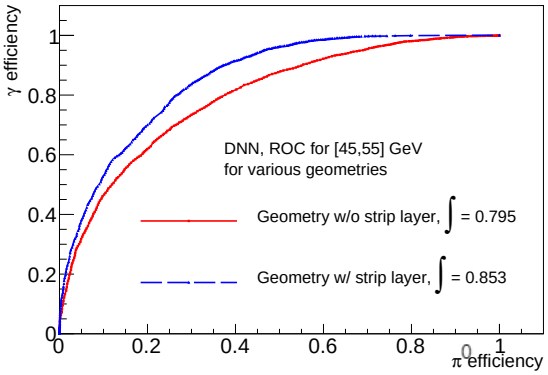


Figure 17: Inclusive ROC curve comparison. Strip layer version perform notably better in general. We observe ROC integral value increase of about 4%, with room for hypothetical improvement of only 15%! Acknowledging that this is only due to "minimal" change in geometry, such gain is very remarkable.

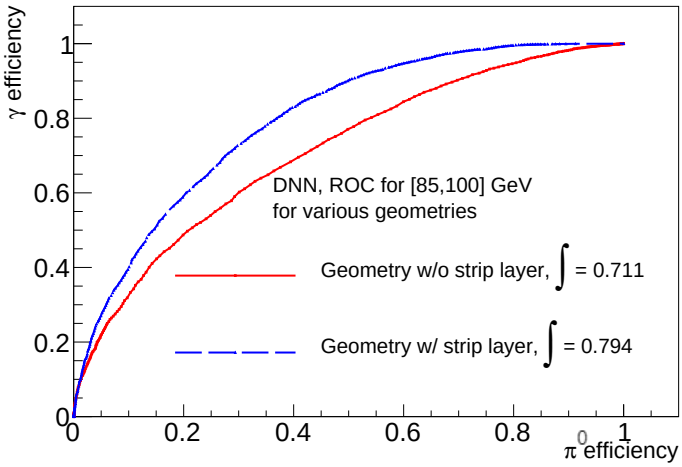
Figures 17 and 18 allow us to observe strong benefits for the DNN from introducing a strip layer. We immediately highlight a better averaged event recognition level and, what is also very important, relative ROC integral growth with energy, i.e. strong enhancement of recognition at high energy. These qualities should be considered as indications in favour of the strip layer geometry.



(a) $\{5, 15\}$ GeV



(b) $\{45, 55\}$ GeV

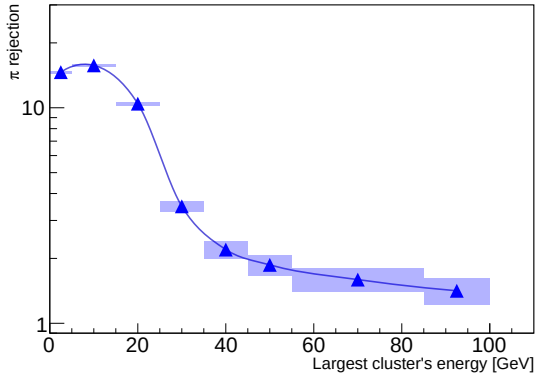


(c) $\{85, 100\}$ GeV

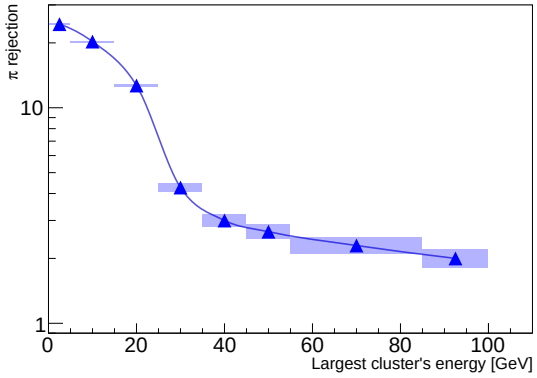
Figure 18: Different geometries DNN's ROC for different initial energy ranges. Even though strip layer geometry outperform uniform version at every single range, the rate of outperformance apparently rises with energy. This makes this geometry a perfect candidate for striving against model's high energy degrading.

3.4 Rejection comparison: FCC-hh DNN

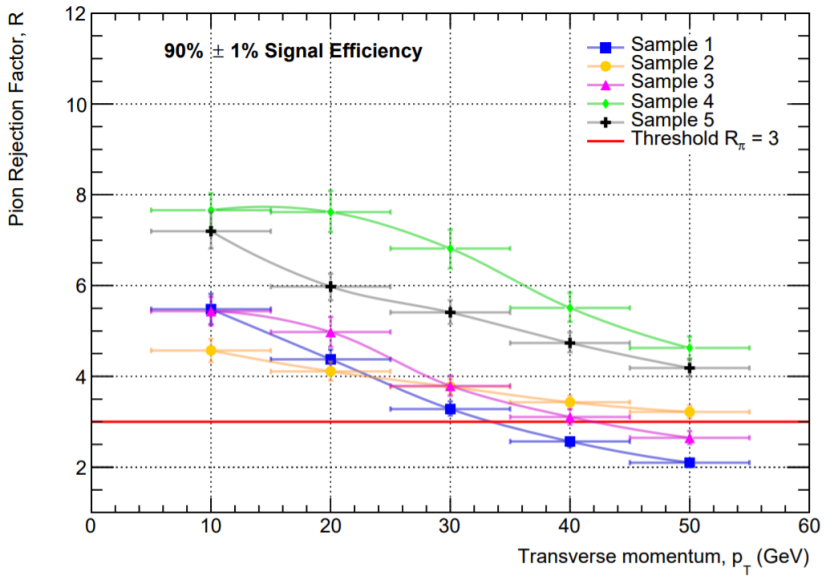
So that to ensure a reader these results are satisfactory, in this subsection we present a comparison with actual FCC-hh fully connected network, built on top of variables, discussed in Sec. 2.1. We present both DNN's rejection factors at 90% γ efficiency and compare them against similar calculation from [3]. This comparison is also useful for validating that the variable list is sufficient, since the FCC-hh document feature same geometries as we do.



(a) Sample 1(geometry w/o strip layer)



(b) Sample 2(geometry w/ strip layer)



(c) FCC-hh rejection plot for various geometries

Figure 19: Rejection factor comparison between FCC-hh DNNs [3] and our best DNNs for Sample 1(i.e. geometry w/o strip layer) and 2(i.e. strip layer geometry).

On Fig. 19 it is apparent that we greatly benefit from additional variables. The most apparent benefit is observed at low energies, where we added e.g. different invariant masses, what proves to be a powerful separator.

4 Advanced networks

Even though we achieved the notable recognition level in Sec 3, turns out there are ways to progress. As it was previously mentioned, with a more appropriate layout we should be able to access better performance.

4.1 Introduction

Convolutional neural network (CNN for short) is a type of neural network generally suited for image recognition. While FCNN would work with a few high-level variables, CNN requires a whole picture (pixel array) to work with, what makes it way more complicated right away (~ 10 input layer size against ~ 1000). CNN also conceptually replaces dense fit layout with a convolutional one, i.e. only nearest **literal** neighbours contribute to a following neuron. This *convolution* process is nicely visualised on Fig. 20.

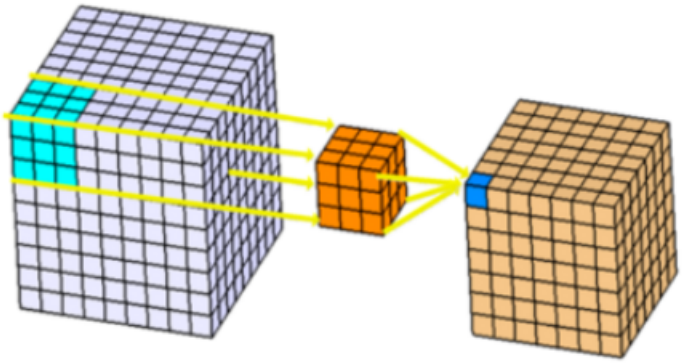


Figure 20: Visualisation of a layer $\rightarrow$ neuron transformation in 3D convolution. It is mostly defined by a sliding window size (3x3x3 in this case, i.e. 27 weights and neighboring neurons associated to the following neuron), number of CNN filters ("big" neurons, made out of whole pictures) and dilatation rate.

As it will be discussed in the following sub-chapter, it is notably easier to leave a low-energy resolution task for a DNN, while a CNN should be way better suited for single cluster cases. This does imply, however, that somehow we are supposed to merge these two models efficiencies. Instead of having two separate models, we employ a *hybrid* layout.

Hybrid neural network (we also use HNN for short) is, in general, a multi-input model, capable of handling non-sequential models (see Fig. 21). Among all models we present, these ones are the most complicated ones (in terms of training complexity, memory/CPU consumption), but also the most promising ones. We can employ a hybrid layout for at least two following use cases:

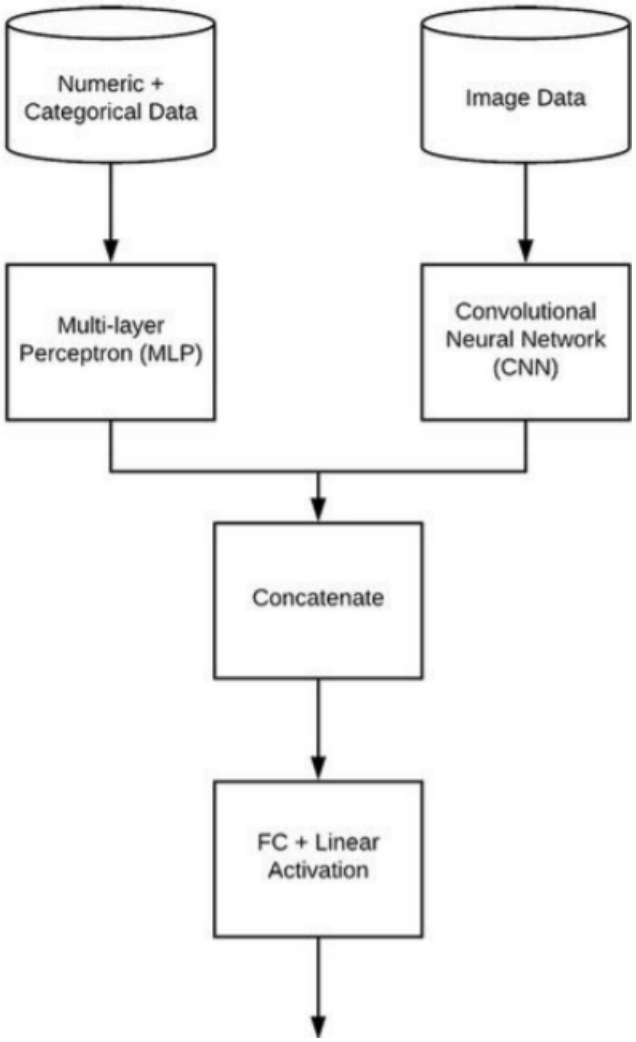


Figure 21: Simplified view of a hybrid model used in the project: image data processor (CNN) gets concatenated with high-level variable processor (DNN), resulting in their mutual output.

- ◇ Best CNN + cluster's energy as DNN : has the same performance, as a pure CNN, but also allows to analyse energy sub-ranged capabilities of it
- ◇ Best CNN + best DNN : if done properly, it grants us with the best of two worlds, effectively becoming the "smartest" model

4.2 Layout and setup

It should at first cause wondering, how exactly can image recognition net be applied for our case. There are actually a few different approaches, which we have only partially exploited by now, but the basis is the same.

Since we may safely delegate low-energy recognition to DNN (e.g. more than one

significant cluster, recall Fig. 5a), we may feed CNN with only the most energetic cluster information, which is both more concise in terms of space usage and useful information percentage.

With that in mind, we shall now more strictly define, what do we mean under "cluster information" above. Since a CNN expects an image, it is natural to feed it with a cell-by-cell energy map of the cluster. Such a map can be represented as an 3D image with a single color channel, namely cell energy. It is also natural to center this map along ϕ and η in a way that center of a picture coincides with an energy barycenter.

Now that we have defined in details, what variables pool we should use, it is time to discuss possible layouts.

Here we see a few possibilities:

- ◇ A simplest one would be to apply a 3D convolution on top of extracted 3D image. This approach, while works, leaves an open question regarding how one is supposed to work with strip layer geometry, in case of which we do not have a cubic matrix:
  - Inner zero padding: Extend the whole cluster's image so that it at least contains strip layer, and fill previously nonexistent cells with zeros, i.e. only every single 4th cell out of strip layer is non-zero. This way, we neither gain nor remove any information from cluster. It also seems to be favourable due to preserving neighbouring scheme.
  - Outer zero padding: Instead of stretching cluster's information, we fill outer cells with zero. This approach is computationally less expensive, however it breaks cells neighbouring scheme.
- ◇ One could also apply a regular FCNN on top of each pixel from the extracted 3D image. This approach is suitable for both geometries without any intermediate reshaping of the image, however it has a significant flaw. During training, the model immediately reaches sub-100% recognition level at training set, with only 83% at validation set. Such an overtraining level makes this approach unfavorable.
- ◇ Several 2D convolutions over a few layers combined: such a layout is easy to set up for both geometries and it is less expensive in terms of memory and computations. The reason we did not prioritize this one is that we expect a less powerful interaction of neurons within different detector's layers than in the case of 3D convolution.

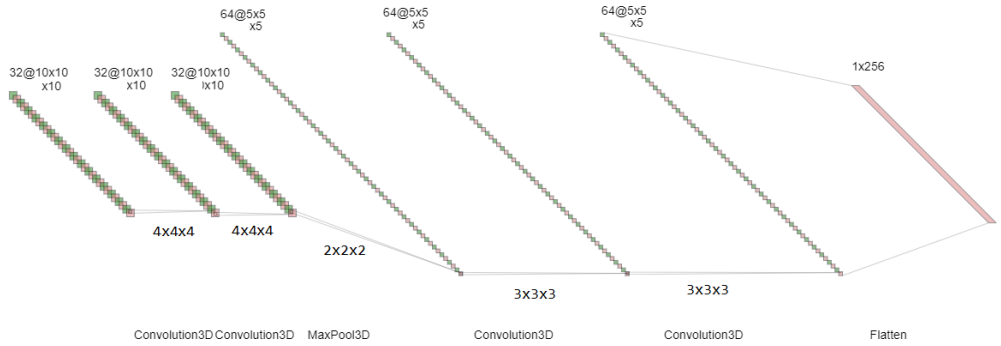


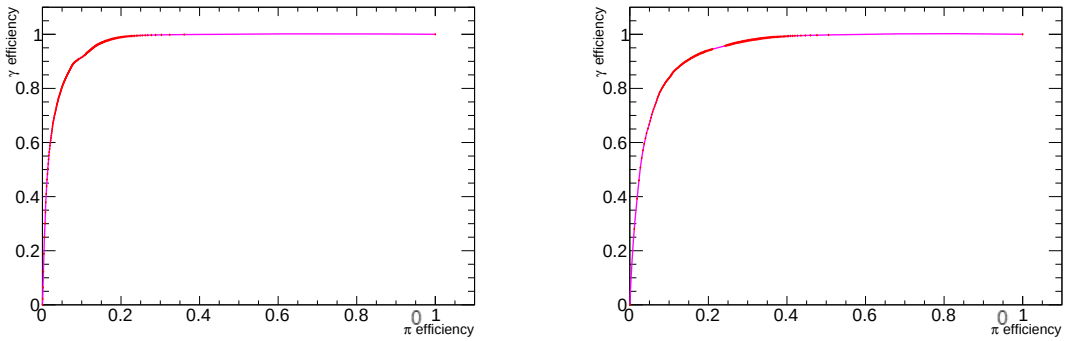
Figure 22: CNN part of a neural net for the geometry w/o strip layer. For strip layer, the layout is inflated along ϕ axis by 4.

For CNN part of a model we use the LeNet-5 2D image recognition net adapted for our 3D window, see Fig. 22.

In the following sub-chapters we employ a CNN part with inner zero padding specifically. As for now, outer zero padding does not prove to be more powerful, while the mix of 2D convolutions is left for further researches. This same CNN part we reuse for a HNN, and comment on differences.

4.3 Performance

First we discuss pure CNN's performance on geometries with and without strip layer. On Fig. 23 we see comparison of inclusive ROCs for these pure CNNs. What is immediately apparent is that with current CNN architecture we are unable to improve separation dramatically, on the contrary to DNN part. The separation levels are, at the same time, way more impressive, than the ones presented on Fig. 17.



(a) uniform celling , $\int = 0.968$

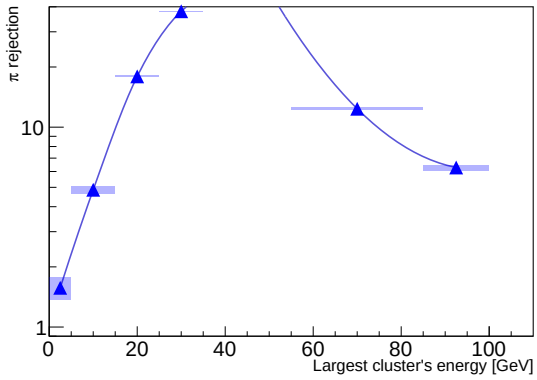
(b) strip 2^{nd} layer , $\int = 0.944$

Figure 23: Inclusive ROC curve for CNNs built on top of various geometries.

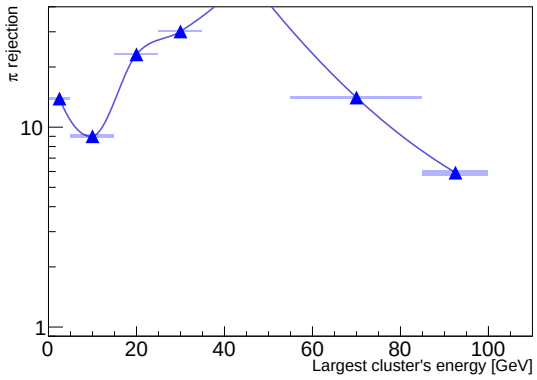
It is now fruitful to compare rejection factors of different HNNs mentioned in Sec. 4.1, namely

- ◇ best CNN + energy on top of geometry w/o strip layer, i.e. "pure CNN" with energy ranges
- ◇ best CNN + best DNN on top of geometry w/o strip layer, i.e. "HNN w/o strip layer"
- ◇ best CNN + best DNN on top of strip layer geometry, i.e. "strip layer HNN"

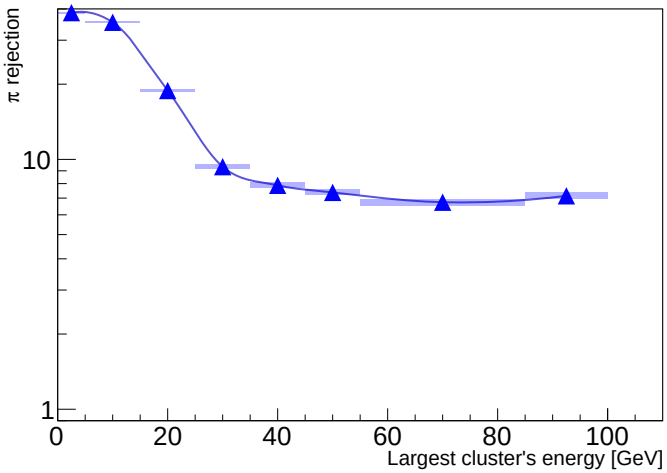
These are presented on Fig. 24. What is immediately notable across Figs. 24a, 24b, 24c



(a) pure CNN w/o strip layer



(b) HNN w/o strip layer



(c) Strip layer HNN

Figure 24: Rejection factors comparison for "pure" CNN with HNNs for different geometries.

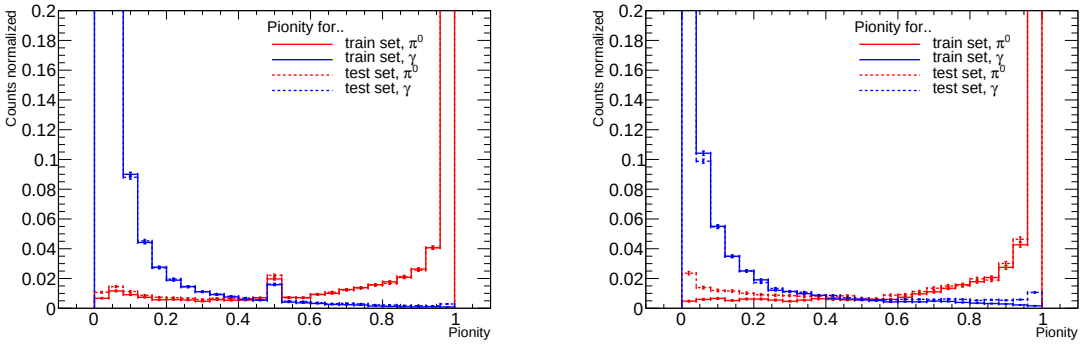
is that DNN part mostly contribute to the low-energy separation power, while hav-

ing basically no effect at high energy. At low energy the strip layer HNN beats the uniform version by a factor of 2-3 and it beats the pure CNN by an order.

At mid energy we observe very strong dominance of the CNN part (in comparison to pure DNN, like Fig. 19). Different behaviour between Figs. 24a, 24b in that energy region is to be investigated. We expect that this difference in behaviour signifies possible further fine-tuning of the model, resulting in even better recognition of uniform HNN. At the same time, Fig. 24c performs way worse at the CNN dominance region (even though it beats the pure DNN by about an order). This gets us to the point, that inner zero padding layout is either not appropriate or not fine-tuned for strip layer geometry. We leave this technical question opened for an upcoming research. Overtraining can also pose an issue, see Figs. 25.

Finally, at highest available energy all three models perform in similar fashion, with rejection of order 10 (note that strip layer HNN is the best at the last point, what is promising).

ROC curves for HNNs with different geometries are presented on Figs. 26, while energy subranged ROC comparison is presented on Fig. 27.

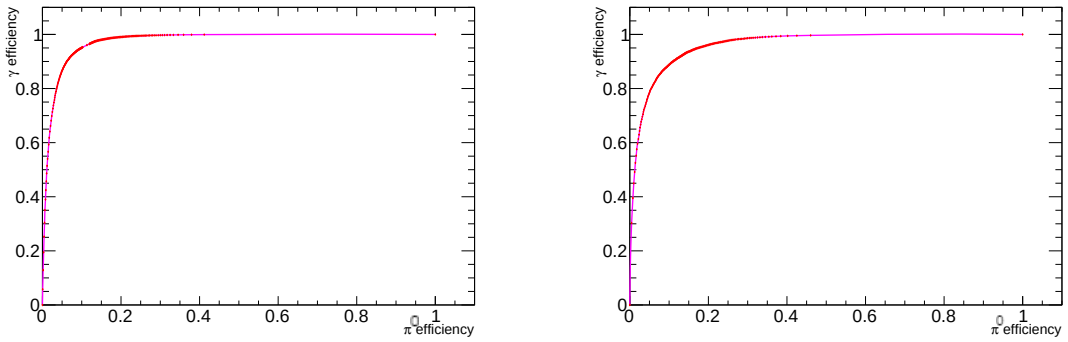


(a) Uniform celling

(b) Strip 2^{nd} layer

Figure 25: Inclusive pionity distribution for HNNs built on top of various geometries. We note a slightly more overtraining for strip layer geometry, what can play an important role on these recognition levels.

We can draw a general conclusion out of it that HNN is way better suited for the separation task than pure DNN and it is definitely worth implementing it instead.



(a) Uniform celling , $\int = 0.974$ (b) Strip 2^{nd} layer , $\int = 0.960$

Figure 26: Inclusive ROC curve for HNNs built on top of various geometries. Due to technical issues with the strip layer layout, as for now, we did not manage to achieve a better separation in comparison to the geometry w/o strip layer.

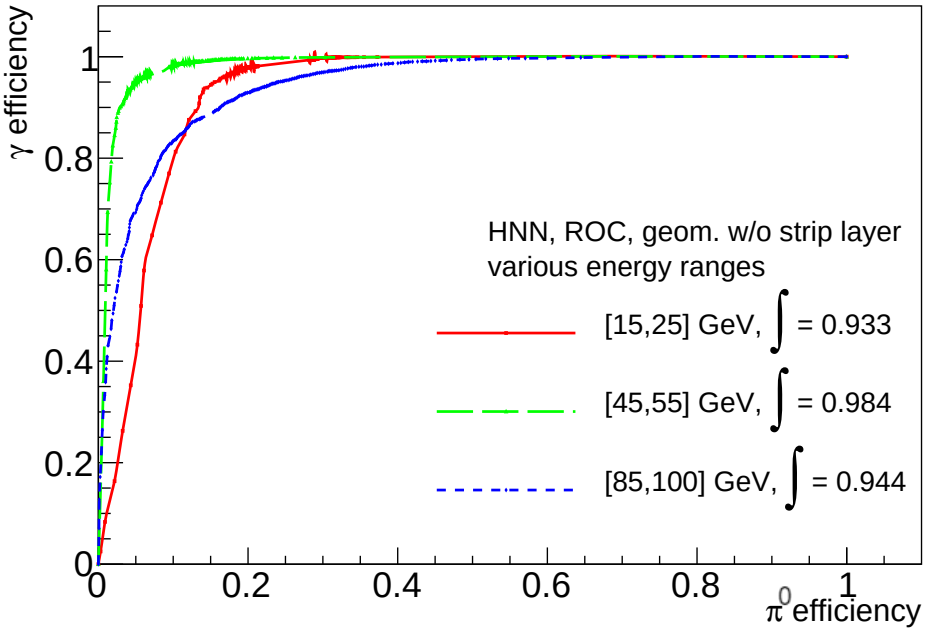


Figure 27: Energy subranged ROC curves for HNN with geometry w/o strip layer. Here we observe an interesting spike in ROC integral value at mid-energies, what will be studied (and hopefully improved) in further research.

5 Summary

We have investigated an impact of various promising layouts and geometries on the performance of a $\gamma\gamma - \gamma$ separation neural network. The strip layer geometry was probed for the task and immediately notable improvements were observed for low-energy and high-energy separation. Average recognition of 94.5% was achieved with the help of hybrid neural network based on geometry without strip layer. Layout difficulties were found for the convolutional part of a neural network built on top of strip layer geometry, what shall be tackled in upcoming researches. We also plan to implement a software managing multiple separation nets for more optimal search for the most promising layout.

References

[1] M. Bauer, M. Heiles, M. Neubert and A. Thamm, *Axion-like particles at future colliders*, *The European Physical Journal C* **79** (2019) .

[2] K. Wandall-Christensen, τ *Decay Mode Identification in a Liquid Argon Electromagnetic Calorimeter at the FCC-ee*. dec, 2021.

[3] M. Aleksa, P. Allport, R. Bosley, J. Faltova, J. Gentil, R. Goncalo et al., *Calorimeters for the fcc-hh*, 2019. 10.48550/ARXIV.1912.09962.