LUND
UNIVERSITY

Trigger-Level Electron Counting in the Light Dark Matter eXperiment using Artificial Neural Networks.

Pavel Oshchepkov

May 2024

Supervised by Lene Kristian Bryngemark and Ruth Pöttgen
Thesis submitted for the degree of Bachelor of Science
Project duration: 2 months
Department of Physics
Division of Particle and Nuclear Physics

Contents

Abstract

The Light Dark Matter eXperiment is a fixed target missing-momentum experiment that searches for light dark matter production via the process of Dark Bremsstrahlung by analysing the energy of beam electrons after they hit a tungsten target. Electron counting is an important part of the experiment as this forms one of the two components of the missing energy trigger. The thesis looks into viability of Artificial Neural Network models in the electron counting procedure, as electron counting is a classification problem, a type of problem in which ANNs have seen large success.
The project looks into two ANN architectures, Convolutional Neural Networks and Recurring Neural Networks. The ultimate aim is an accuracy of 1 misclassification in 100000 events. We analyze the difference between the performance of various architectures. The best results is a CNN network with 0.9944 accuracy and an RNN network with 0.9598 accuracy. RNN architecture misclassified events mainly based on the number of readout hits present in an event, while the CNN relies on a combination of total energy and event geometry when classifying an event. In particular we show that performance is not degraded when using a CNN with data pooled in pre-processing to match the "Trigger Cells" which will provide the calorimetric input to the trigger decision. In contrast, both shorter training time and an improved accuracy can be obtained.

Popular Summary

Dark matter is estimated to make up about a quarter of our universe. This is about five times more than the scientific community has been able to describe using the largely successful Standard Model. Dark matter does not interact electromagnetically and thus is invisible for the majority of techniques used by scientists to explore matter. We know little about the nature of dark matter and its properties. The reason we know of dark matter's existence is because we can infer it from the gravitational effects it has on the visible matter around it.
The Light Dark Matter eXperiment aims to explore dark matter production through a process called dark photon bremsstrahlung. In this process a dark photon is released from an electron close to a tungsten nucleus. This dark photon is theorised to decay into a dark particle anti-particle pair. In the experiment a beam of electrons is fired at a thin tungsten target with the energy of electrons being recorded in a calorimeter. A large amount of missing energy could indicate dark matter production.
Dark photon bremsstrahlung is expected to be a rare event. The electron beam produces 37 million events every second, however the majority of the events produced have no value to the research. Only 5 thousand events are to be read out by the the detector for analysis. In order to find the possible dark bremsstrahlung events, a system of triggers detects if the total deposited energy is lower than a cutoff energy. The cutoff energy varies based on the electron multiplicity of the event. Thus for the appropriate cutoff energy to be applied one needs to know the electron multiplicity of the event.
Artificial Neural Networks have been historically strong in solving classification problems. The electron counting is a classification problem based on the inputs from the electromagnetic calorimeter and the trigger scintillator of the LDMX detector. It was shown in previous research[1] that Artificial Neural Networks are capable of achieving better than 95% classification accuracy. However the "black box" nature of these models means that we have little knowledge of what the classification is based on. This project analyzes the effectiveness of Artificial Neural Networks in counting electrons in the LDMX detector, compares the performance of various Artificial Neural Network architectures to understand what event qualities affect the classification process.

1 Introduction

Dark Matter constitutes about 85% of mass in the universe that we live in[2]. The existence of Dark Matter is supported by a large body of research from the last 100 years. One piece of evidence in support of Dark Matters existence being the astronomical observation of rotation curves of galaxies which are not consistent with results given by applying Newtonian mechanics[3]. This result indicates an existence of a large mass, that is invisible to us, throughout the galaxies. Dark matter is electromagnetically neutral and thus is hard to detect using majority of detection tools available to us[2]. Some methods used in Dark matter detection involve using Direct Detection experiments in which research is done on Dark Matter candidates via scattering off known particles, or indirect detection experiments in which researchers observe the particles produced in Dark Matter annihilation[3].
The Light Dark Matter eXperiment (LDMX) aims to look into Dark Matter Production via a process of Dark Bremsstrahlung, a process in which an electron scatters off a heavy nucleus and releases a dark matter particle antiparticle pair either directly or through a mediator particle[4]. The subject of study is the energy missing after the Dark Bremsstrahlung event[4][5]. As Dark Bremsstrahlung events are expected to be rare, LDMX is expected to fire up to 10^{16} electrons on a tungsten target, with the majority of the electrons producing no Dark Bremsstrahlung events in the process. This constitutes a large amount of data that holds little research value and thus should be discarded as early as possible. At trigger level, LDMX uses a system based on information from two of its detector components, that is meant to discard events of no interest[6]. The beam has a rate of 37 million events per second, majority of the events will be discarded and only 5 thousand events per second will be saved for further analysis. The decision is made using a set of triggers. A single electron event passes the trigger if the electron energy is below 1.5 GeV, for a 4 GeV electron beam, which corresponds to at least 2.5 GeV of missing energy[4]. When an event involves more than one electron a different energy cutoff is applied. Because of this counting the electrons in an event becomes an important task that needs to be accomplished at a trigger level. This project looks into a possibility of using Artificial Neural Networks in order to count the electrons in the detector at a trigger level. This would allow for a consistent application of appropriate triggering for events of various electron multiplicities.

2 Theory

2.1 Dark Matter

The Standard model is a significant scientific achievement that was developed over the past century. It is a theory that describes all of the known fundamental forces, with gravity being a notable exception, along with all of the elementary particles known to humanity. This however accounts for only about 5% of the known universe energy content. The rest is constituted of Dark Matter and Dark Energy, both of which are not visible to us.
The idea that more mass exists in the universe than is accounted for by the Standard model comes from a multitude of astronomical observations with one being the observations of rotation curves of galaxies[3]. The basic expectation is that the rotational speed of the galaxies would drop off as $v(r) \propto \frac{1}{\sqrt{r}}$ with increased distance from the galactic center as evident from Newtonian Mechanics $v(r) = \sqrt{\frac{GM(r)}{r}}$ where $M(r)$ is the mass as a function of the radius[3]. This however is not observed. The velocity plateaus in contradiction to the results expected from Newtonian mechanics. An explanation for such behaviour would be existence of invisible mass that increases with radius as $M(r) \propto r$[3]. Research in Dark Matter is still developing and there exists a variety of proposed candidates for Dark Matter particles[2]. While the exact properties of Dark Matter are not known there are a number of qualities that a Dark Matter candidate should exhibit. Some of these qualities are the electromagnetic neutrality of the Dark Matter particles, the particle's abundance, and limited self interaction of the candidate[2].
One of the explanations for Dark Matter is it being a thermal relic of the early universe. The thermal relic explanation of the origin of Dark Matter allows for Dark Matter masses of sub-MeV to a 100 TeV range[4]. An important feature of thermal relic Dark Matter is that it ends up stopping the production and annihilation process at a comoving density based on the particles annihilation cross-section[2].
One of the possible candidates that fulfills these requirements is known as a Weakly Interacting Massive Particle, or WIMP for short. As suggested by its name the particle is expected to interact primarily via the Weak Interaction and has a large mass between $m = 100$ GeV and 1 TeV. A feature of WIMPs is that they could constitute all of the Dark Matter in the universe[2]. While WIMPs cover the multi-GeV there is still room for weakly-coupled sub-GeV Dark Matter known as light dark matter. The basic qualities of light dark matter are the sub-GeV mass and neutrality in electromagnetic, weak and strong interactions[7]. Dark Photon is a proposed intermediary for Dark Matter production which is capable of feebly interacting with an electrically charged particle through the process of kinetic mixing[7]. This would allow for a light dark matter interaction with electrons and open a path for Dark Matter production at accelerators[4].

2.2 Dark Matter Production Mechanisms

The thermal relic origin of dark matter implies a small non gravitational interaction with standard model particles which allows for a possibility of Dark Matter being produced in an accelerator[4]. LDMX is an experiment designed to search for Dark Matter candidates in a sub-GeV range by firing an electron beam with energy of either 4 or 8 GeV at a fixed tungsten target[4]. Multiple Dark Matter production mechanisms are suggested.

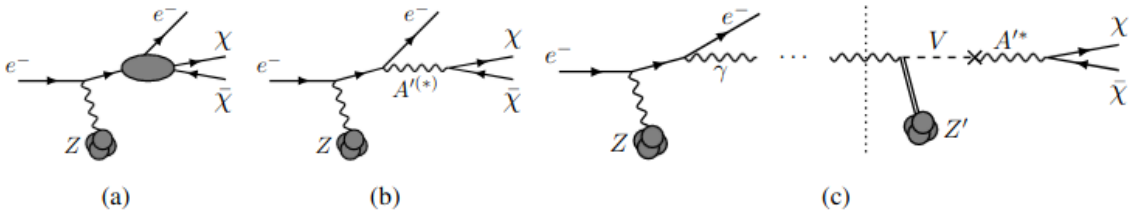


Figure 1: Dark matter production mechanism schematic. (a) direct particle-antiparticle production. (b) dark photon bremsstrahlung. (c) invisible vector mason decay[6]

The Dark Bremsstrahlung results in a production of either a dark particle-antiparticle pair or a mediator particle that then decays into a particle anti-particle pair. A second mechanism proposed for dark matter production in the LDMX experiment is the photo-production of vector mesons. This mechanism involves a hard Bremsstrahlung of a photon which later decays into a particle-antiparticle pair via mixing with an intermediate particle[6]. All of these mechanisms involve a beam electron loosing energy in a bremsstrahlung interaction, which is the main handle for triggering on potential dark-matter production events in LDMX.

2.3 Particle Detection

The LDMX detector consists of various sub-systems that allow to analyze the trajectory, energy and momentum of the electron before and after it hits the target. The two detector parts of particular interest for this project are the trigger scintillator and the electromagnetic calorimeter. Before one can understand how these detectors work it is first important to understand the mechanism through which an electron can transfer its energy. Electron detection largely depends on electromagnetic interaction with matter. The main modes of interaction with matter for electron are ionization and bremsstrahlung. Ionization is the main mechanism for electron energy transfer at low energies. In the interaction with the electron cloud of an atom, energy of the electron is transferred to the atom which can result in an electron being ejected from the cloud if the transferred energy was higher than the ionization energy of the atom. Bremsstrahlung overtakes ionization as the dominant mechanism for electron matter interaction at the electron energy of about 10 MeV and thus is the more relevant process in this project. The process involves an energetic electron slowing down in the Coulomb field of a nucleus and loosing energy through photon radiation[8].
For photons interacting with matter proceeds through photoelectric effect, Compton scattering and pair production. These effects are prevalent at different energies[8]. Photoelectric effect is the main mechanism of energy transfer at low energies. In photo-electric effect a photon is completely absorbed by an atom[8]. This leads to the expulsion of the electron if the energy of the photon was larger than the binding energy of the electron. After the expulsion the electron obtains a kinetic energy equivalent to the energy difference between the energy of the photon and the electron binding energy. The mechanism most prevalent between the energies of about 10 keV and 10 MeV is Compton scattering[8]. In Compton scattering the gamma ray interacts with the electron transferring a portion of its energy and is subsequently scattered off the electron with its remaining energy. The last mode of photon interaction with matter is pair production. In pair production a photon with substantial amount of energy produces an electron-positron pair. The production of the electron-positron pair sets a lowest energy cut off for this interaction at 1.022 MeV which corresponds to the mass of produced particles. The remaining energy of the

photon is divided among the produced particles as their kinetic energy[8].
The fact that high energy electrons are capable of producing photon and high energy photons are capable of producing electrons and positrons results in the phenomena of electromagnetic showers. A high energy electron upon interaction with matter ends up producing a cascade of photons and electrons that dissipate energy inside of the material. This is used as a basis for the electromagnetic calorimeter.
The trigger scintillator is used to provide the information on the number of electrons that have hit the detector. The electromagnetic calorimeter designed for the experiment is a multilayer Silicon-Tungsten (Si-W) sampling calorimeter. The main concept in this calorimeter utilizes electromagnetic showers in order to detect the hits and read out the energy of the incoming electron. When the electron hits the calorimeter it creates an electromagnetic shower. At high energies the primary interactions seen from electrons and photon are bremstrahlung and pair-production. Thus the electromagnetic shower continues until the produced electrons and photons no longer have the energy to interact through those mechanisms. At low energies the main modes of interaction become ionization and the photoelectric effect which allow for a analogue signal readout from the silicon pads inside of the calorimeter. The second part of the calorimeter is the tungsten plates that are layered between the silicon pads. The tungsten is used as the absorption material which provides a medium for the electron shower to develop more compactly due to high density.The use of Si-W sampling calorimeter allows for a high spatial resolution of analyzed events. However a drawback of the calorimeter is the lower energy resolution. The tungsten layers provide no energy readout resulting in the energy missing from the detector. Energy deposited to tungsten and shower leakage result in the recorded energy being generally lower than the energy of the beam. This is partially mitigated by calibrating the calorimeter, however introduces an energy spread where the energy distribution of the events forms a Gaussian about the beam energy. Figure (14a) shows the energy distribution of events in test sample, the events are mostly grouped into Gaussians around the beam energy multiplied by the event multiplicity.

2.4 The Light Dark Matter eXperiment

LDMX is a fixed target missing momentum experiment[6][9]. The detector is comprised of trigger scintillators, tagger , recoil tracker, electromagnetic and hadronic calorimeter[6].

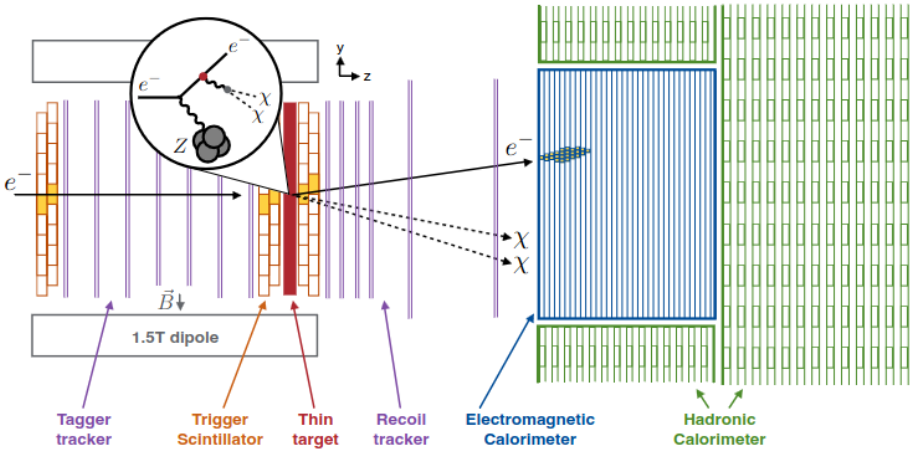


Figure 2: Schematic representation of the LDMX detector[6]

The individual part of interest for this research are the trigger scintillator and the electromagnetic calorimeter as data from these two components will be used to perform the electron counting. During the experiment a beam of electrons is pointed at a tungsten target. The electrons first pass through two trigger scintillators and hit the tungsten target, after which the electrons pass through another trigger scintillator and hit the electromagnetic calorimeter where their energy is recorded. In the experiment the electromagnetic calorimeter serves a role of recording the energy of the electrons that hit it, while the trigger scintillator tracks the trajectory of the incoming electrons.
The trigger scintillator used in the experiment is a polyvinyltoluene scintillator, It consists of bars mounted in an array read out by silicon photo-multipliers (SiPMs) in order to analyze the trajectory of the electron. One module consists of 48 bars in the y plane as indicated by sketch in Figure (2). The project also studies the horizontal segmentation of 8 bars in the x plane.
The electromagnetic calorimeter is made up of 7 hexagonal modules arranged in a "flower" shape as seen in the Figure (3):

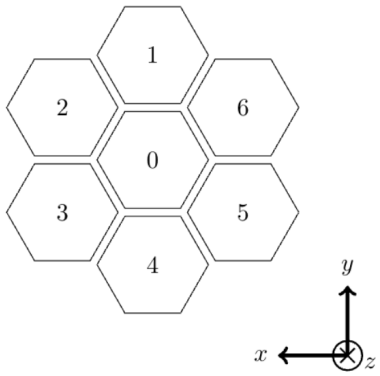
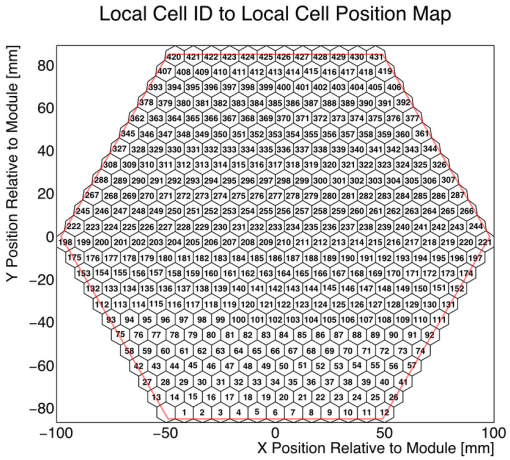
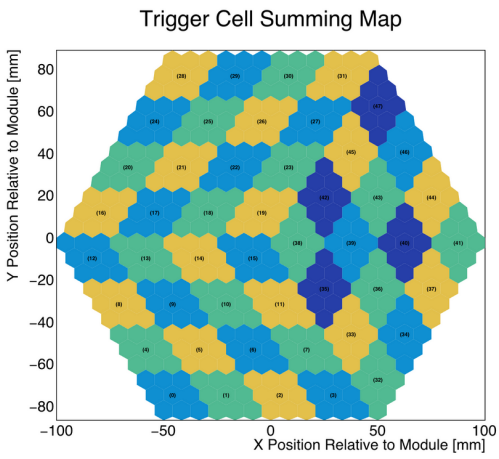


Figure 3: Electromagnetic calorimeter module flower layout[10]

There are 34 layers with each layer consisting of 7 modules and 432 individual cells for each module. This arrangement gives the electromagnetic calorimeter high spatial resolution which allows for accurate spatial analysis of multi-electron events and showers. At the trigger level, the information is grouped into 48 trigger cells that combine 9 cells each as seen in Figure (4b).



(a) Electromagnetic calorimeter module cell layout



(b) Electromagnetic calorimeter module trigger cell layout

Figure 4: Electromagnetic calorimeter module division[10]

2.5 Current Electron Counting Algorithm

The current electron counting strategy in the LDMX detector utilizes information from three trigger scintillator pads that can be seen on Figure (2). The trigger scintillator pads are divided into 48 cells vertically. An energy cutoff limits background from electronics noise. An electron passing through a trigger scintillator pad deposits energy which is read out as a hit. Hits are detected and grouped together with nearest neighbour into electron hit clusters. An algorithm then fits a straight line from the initial pad through the two following pads. If there exists a cluster within a Chi-squared based cutoff distance from the straight line on all scintillators then a track is recorded. The number of straight tracks that are recorded by the system corresponds to the number of electrons counted in an event[11].

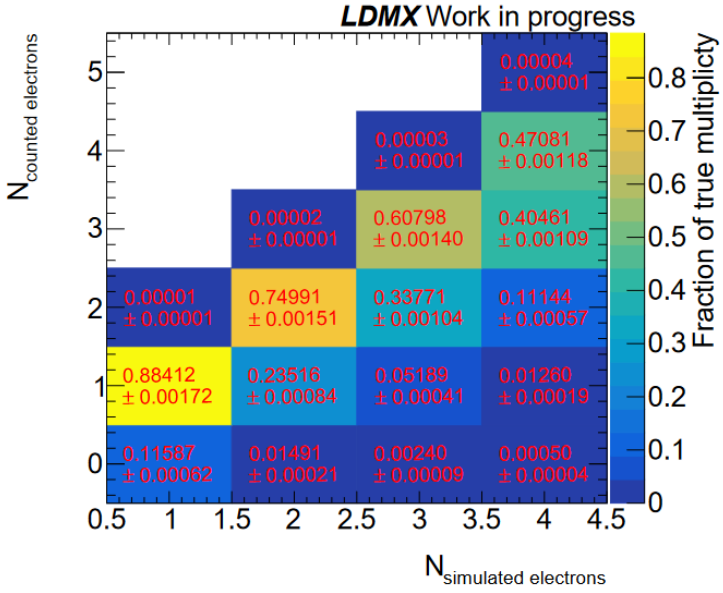


Figure 5: Confusion matrix for current electron counting algorithm[11]

Figure (5) shows the so-called confusion matrix for the current electron counting algorithm. The matrix shows the classification accuracy of the current electron counting algorithm for events of various electron multiplicity. Two qualities that can be seen directly from the confusion matrix is the strongly decreasing accuracy with an increase in the electron multiplicity, and as the algorithm's tendency to undercount the electrons in an event. The undercounting bias in the algorithm is preferable as the detector relies on missing momentum when classifying an event as a dark bremsstrahlung events[9]. Thus in case of electron overcounting the energy expected would be higher than the one measured leading to a false positive classification.

2.6 Artificial Neural Networks

The task of trigger level electron counting in LDMX constitutes a classification problem. In this classification problem a relation can be established between the multiplicity of electrons hitting the detectors and the hits recorded in the trigger scintillator, the energy and distribution of the electromagnetic shower inside of the calorimeter. Machine learning and artificial neural networks in particular are valuable tools in performing classification tasks on large data samples. The name artificial neural network refers to a set of algorithms whose design was inspired by the neurons in human brain. Artificial neural networks can be described as a large set of massively parallelized small computing nodes that can learn from provided data and store the knowledge[12][13]. The small computing nodes perform basic mathematical processes based on received input and provide an output based on the activation function. A basic model of an ANN node is shown in the figure below:

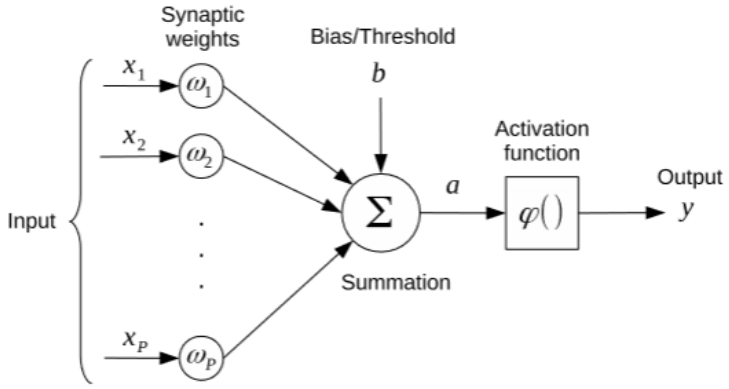


Figure 6: Artificial Neural Network node representation[13]

An artificial neural network node receives inputs from either input nodes or a previous layer of compute nodes, the input values are then multiplied by synaptic weights which are then summed up and passed to an activation function[13].

$$y = \phi(\sum_{k=1}^{K} \omega_k x_k + b)$$

Artificial Neural Networks consist either of a single or multiple layers that are comprised of these nodes. The input data provided to the network passes through the series of node layers to provide an output. An artificial neural network learns information from a data set through a process known as training. There exist supervised and unsupervised learning of artificial

neural networks. The process of supervised learning provides the network with a set of true labels to benchmark the output while unsupervised learning does not[13][14]. For this project supervised learning will be used with all of the events having a true label attached to them. In the process of supervised learning a network checks the produced output against the true label. The synaptic weights of the network are responsible for increasing the performance of the network during training as well as saving the training state of the network. The weights are initialised to random values and are then tuned in the process of training[13]. In order to train a network one needs a numerical representation of the network accuracy. This representation comes in a form of loss function $E(\omega)$ which is suitable for a type of problem that the artificial neural network aims to solve. The loss function represents a difference between the network output y and the target value. Thus the problem becomes an optimisation problem in which $E(\omega)$ needs to be minimised with respect to weights ω[13]. In order to do that a learning rate η is defined and the weights are modified based on η and the derivative of the loss function $E(\omega)$ to ω[13].

$$\frac{\delta E}{\delta \omega_k} = \sum_n \frac{\delta E}{\delta y_n} \cdot \frac{\delta y_n}{\delta a_n} \cdot \frac{\delta a_n}{\delta \omega_k}$$

Then the value of weights is changed by $\Delta\omega = -\eta \frac{\delta E}{\delta \omega}$. The process of weight tuning is then repeated throughout the training until the loss function $E(\omega)$ is minimised[13].
The two types of artificial networks of interest for this project are the Recurrent Neural Network (RNN) and the Convolution Neural Network (CNN). Recurrent Neural Network is a type of neural network that specialises specifically on the processing of sequential data. The unique part of the recurrent neural network approach is its ability to feed the layer output into the previous layer instead of being a strictly feed forward network[14][13].
The Convolutional Neural Network is designed to specifically work with grid arranged values. Convolution is a mathematical operation on two functions that end up combining the information to produce a third function. The convolution between two functions x and ω is given as

$$s(t) = \int x(a)\omega(t-a)da.$$

Where $x(t)$ represents a single input and $\omega(t)$ represents a corresponding kernel. The act of convolution is denoted using the $*$ symbol as follows[14].

$$s(t) = (x * \omega)(t) = \int x(a)\omega(t-a)da.$$

The above represents convolution over a set of continuous values in a single dimension, however it can be expanded to any amount of dimensions. When working with a tensor like in case of convolution on the 3-dimensional position data from the electromagnetic calorimeter the data needs to be discretized. For a tensor the convolution process will look as follows[13][14][1]:

$$s(i,j,k) = (x * \omega)(i,j,k) = \sum_{m,n,l} x(i+m, j+n, k+l)\omega(m,n,l).$$

The process of training a CNN uses a number of convolution layers that search for important features in the provided data[13]. The unique behaviour of the CNN is the use of local kernels. That means that instead of each node in the layer being connected to each node in the following layer the connections are instead localized connecting nodes to only a few closest nodes of the next layer. This allows for increased performance of a CNN on data samples that involve a large amount of input data[14][13].

3 Method

3.1 Data Pre-processing

This project builds upon the code[15] and findings by Jacob Lindahl[1], using two of the four ANN architectures and the pre-processing methodology developed in the previous work[1]. This project studies RNNs and CNNs for electron counting at trigger level. In order to train the Artificial Neural Networks a set of events simulated by ldmx-sw package was used[16]. The simulated ROOT[17] files contained information about up to 10000 electron hit events, and each ROOT[17] file contained events of only one multiplicity. For the project events of electron multiplicities 1-4 were used in training.
Two different Artificial Neural Network architectures were explored during the project. For each of these, three different data sets were used for training. The first set of data included only information from the electromagnetic calorimeter and will from now on be referred to as just Ecal. The second set also included y position information from the trigger scintillator, from now on referred to as ETS. The last set of data included the information from electromagnetic calorimeter and both x and y information from the trigger scintillator, from now referred to as ETSX. The goal was to see the effect additional trigger scintillator information would have on the ANN performance. This makes six different types of data pre-processing for various architectures and data models. In order to achieve that a two step process is used. First the relevant information is extracted from a set of ROOT to a set of csv formatted files. The saved data consists of electromagnetic calorimeter hit coordinates, the energy for each hit read out from the calorimeter and coordinates of the trigger scintillator hits. This data is then used by one of the 6 pre-processing methods to convert it to architecture appropriate representation. Afterwards each event is saved in its unique compressed numpy[18] file.

3.1.1 RNN

For Recurrent Neural Networks the data is combined into a single one-dimensional array. The coordinates of a hit in electromagnetic calorimeter are processed in layer, module and cell coordinates. The values of coordinates and energy are combined into a single value for each individual hit as follows:

$$HE = L \cdot 10^7 + M \cdot 10^6 + C \cdot 10^3 + E \tag{1}$$

where HE is the single value representation of a hit used in the RNN, L is the layer coordinate, M is the Module coordinate,as seen in Figure (3), C is the cell coordinate as seen in Figure (4a) and E is the recorded energy value[1][15]. For trigger scintillator hits a similar procedure is done. For ETS RNN a trigger scintillator hit is calculated as follows:

$$HTS = Y \cdot 10^3 + 1 \tag{2}$$

where HTS is a single value representation of a trigger scintillator hit, Y is the y coordinate of the trigger scintillator hit. For ETSX RNN the trigger scintillator hits were calculated as follows:

$$HTS = X \cdot 10^6 + Y \cdot 10^3 + 1 \tag{3}$$

with X being the x coordinate for trigger scintillator hit[15]. This information is then combined into a single array. For ETS and ETSX the trigger scintillator hits come in first, followed by electromagnetic calorimeter hits and padded by zeros to achieve a uniform shape. Ecal events

do not include the trigger scintillator information. An example ETSX event with 4 tracks on the trigger scintillator and 300 hits on electromagnetic calorimeter would look as follows.

$$[HTS_0, HTS_1, HTS_2, HTS_3, HE_0, HE_1 ... HE_{298}, HE_{299}, 0, 0..., 0, 0]$$

The total length of the array is not important, but should stay constant for a single training, for this project an array length of 500 was used as it accommodated all of the events used during the training and testing process. The value is based on the largest number of hits seen in an event from the test sample.

3.1.2 CNN

For CNN the same data is reorganised in a 3 dimensional tensor. The three dimensions are the module, layer and cell coordinates of the hit and the value at those coordinates is the energy readout by the electromagnetic calorimeter. For the Ecal CNN a 3 dimensional tensor of empty values is created with dimensions of 34 layers, 7 modules and 450 cell values. The energy for each of the electromagnetic calorimeter hits is then recorded into the tensor at the appropriate coordinates. The data for ETS and ETSX CNNs is prepared in a similar way with one extra layer added resulting in a shape of 35 layers, 8 modules and 450 cells. Then the first layer of coordinates is used to store the trigger scintillator hit coordinate information. The values corresponding to trigger scintillator y coordinate on the 0th axis of the $L = 0$ and $M = 0$ are turned to 1 for each hit for ETS CNN. For ETSX both the x and y coordinates are used to find a position on the $L = 0$ layer and are subsequently turned from 0 to 1 for each hit. An example of a 4 electron multiplicity ETSX CNN event is given in Figure (7).

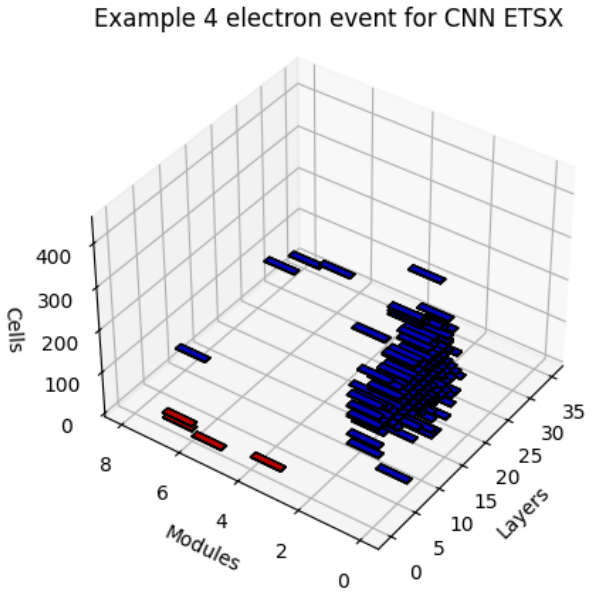


Figure 7: A representation of an ETSX CNN event.

The trigger scintillator hits are represented in red while the electromagnetic calorimeter hits are shown in blue. Figure (7) has a number of interesting features that are common for a majority of the events. It can be seen that most of electromagnetic calorimeter hits are located

in module 0, which is the center of the flower arrangement and thus the module at which the beam is targeted, see Figure (3). Similarly majority of cell readings are between cells 100 and 300, at the center of the module, see Figure (4a). The number of affected cells increases with an increase in layer values due to the increasing spread of the electron shower through the layers and then the readings disappear as majority of the energy is deposited

3.2 Artificial Neural Network Training

For the training of the artificial neural networks a set of data is first pre-processed and located into a specific directory. Arrays of file names and true electron multiplicity labels are then created and shuffled to achieve an even distribution of events of various multiplicity during the training. The shuffled arrays are then split into a training and validation set with 70% going for training and 30% validation. The processed file paths are then combined with labels to produce a batch of file-label pairs. The model is subsequently trained using this data. For the training of the model a sample of 2 million events was used to verify replication of results in previous work [1], 8 million newly processed events were used for training of the scaled up models, 400 thousand events were used for doing the pooling research and a set of 40 thousand events was used as a test sample. In each of this data samples the events are evenly split by their multiplicity. For each type of architecture and data model a different set of hyper parameters, parameters that control the learning process, was required in training.

3.2.1 RNN

The RNN architecture consisted of four consecutive GRU layers, followed by two Dense Layers and a Dropout layer ending with a last Dense layer. The GRU layers are responsible for learning about dependencies between sequential data and are a defining characteristic of RNNs; the Dense layers are node layers in which every node receives input from each node of the previous layer; lastly the dropout layer is responsible for dropping out training data which helps avoiding over training in the network[19]. The input expected by the network is a single dimensional array of a user specified length. The output is an array of 4 probabilities where the index of the largest probability determines the predicted electron multiplicity. The parameters used in training various RNN networks can be seen in Table (1). The batch size used in all training is 32. The input shape is a one dimensional array of given length.

RNN Type	Input shape	Learning rate	Layer sizes
Ecal	454	10^{-4}	[128,32,32,32,64,32,16,4]
ETS	500	3×10^{-4}	[128,32,32,64,64,32,32,4]
ETSX	500	3×10^{-4}	[128,32,32,128,64,32,32,4]

Table 1: Hyper-parameters used when training RNN networks

3.2.2 CNN

The CNN architecture uses a set of 4 layer blocks, each layer block consists of Convolution 3D, Batch Normalization and MaxPooling3D layers, afterwards it uses a series of Flatten, Dense, Dropout and Dense layers.The Convolution3D are responsible for performing the convolution

on the data; Batch Normalization layer normalizes the output of the previous layer by subtracting the mean value of the batch and dividing every value by the standard deviation; MaxPooling3D combines the information of multiple cell into a single cell, thus lowering the data resolution; and Flatten layer transforms the tensor shaped data into a vector[19]. The parameters used when training CNN networks are seen in Table (2). The batch size stays the same for all networks at 32. The input is a tensor of given input shape.

CNN Type	Input shape	Learning rate	Layer sizes
Ecal	(34,7,450)	5×10^{-3}	[64,64,64,64,64,64]
ETS	(35,8,450)	5×10^{-3}	[64,64,64,64,64,128]
ETSX	(35,8,450)	5×10^{-3}	[64,64,64,64,64,128]

Table 2: Hyper-parameters used when training CNN networks

3.3 CNN Training optimization

Training the CNN networks on larger data sets takes a long time. In particular the training of Ecal CNN on 8 million events took over 400 hours for a single epoch. While a single epoch seems to be enough to achieve high accuracy, the large time investment prevented testing out multi epoch training results on a large data sets. One solution that was implemented is two types of data pooling during the data pre-processing.

The first pooling method used was "naive pooling": the energy readout of a set of cells were summed up into a single energy readout, combining the cells sequentially based on their cell ID. As seen in Figure (4a), for a pooling rate of 9 cells, 0 to 8 would be combined into super-cell 0, and cells 9 to 17 would be combined in super-cell 1. This allowed for a simple pooling mechanism where the pooling rate can be set to any divisor of 450. However this smeared spatial information from the detector, as an example combining readout information from cells 12 and 13 almost surely results in combining energy deposited by two different electrons.

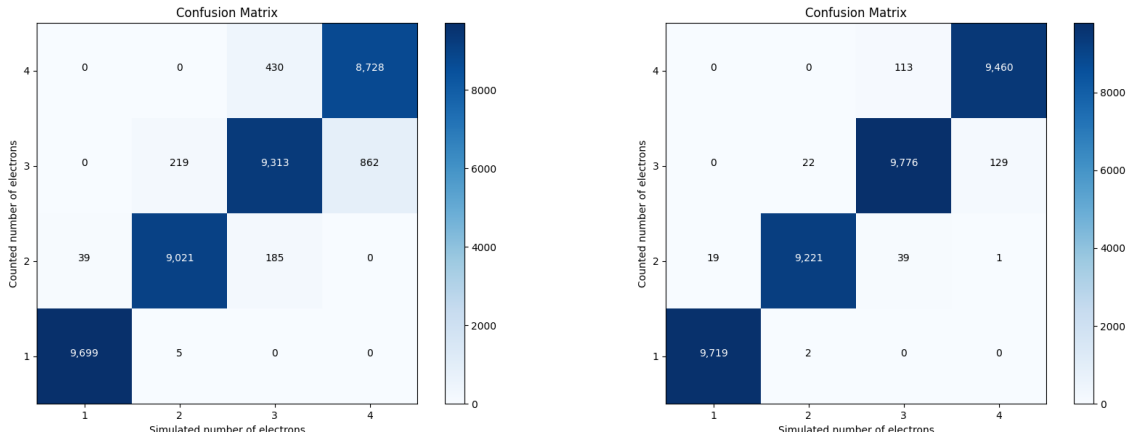
The second type of pooling used was based on the proposed trigger cell mapping from the LDMX detector. As seen on Figure (4b) the trigger cells are a combination of 9 adjacent cells from the detector. This approach combines the ability to pool data and decrease training time with the ability to see if the information available from the LDMX detector at trigger time is enough to make accurate electron multiplicity predictions.

For the training of naive pooled networks, two different variations of the CNN architecture were used. One of them had max pooling set to 1 for cells on all layers, the second one used the same max pooling layer values as the non-pooled networks. The naively pooled networks were trained on a 400 thousand event dataset with pooling rates corresponding to all divisors of 450. These values were used to keep pooled cells at a consistent size, as otherwise some of the pooled cells would agglomerate more cells than others. The Trigger Pooled CNN was trained on the 400 thousand event sample and subsequently scaled up to the 8 million event data set used for other networks.

4 Results

4.1 Reproducing previous results, 2 million events data sample

First set of results was obtained when trying to replicate previously obtained results in Jacob Lindahl's thesis [1]. While trying to obtain the results two bugs were found in the code. One of the bugs was with the processing of trigger scintillator information. Instead of obtaining the data for separate electron hits on the trigger scintillator the hit information from the same electron at different time points was used. This led to all trigger scintillator information being degenerate. This unfortunately meant that the results for the ETS and ETSX networks were unreliable. The second bug was related to data pre-processing. Only a small portion of pre-processed events were saved. This led to the networks being trained on a smaller data sample than expected. Due to this some of the networks in Jacobs thesis take numerous epochs to achieve the same accuracy that networks in this work achieved in one epoch. After uncovering and fixing the two bugs Ecal RNN and CNN networks have been trained on the same 2 million event data set used by Jacob Lindhal. The two trained networks were tested on the test sample of 40 thousand events and provided the following results.



(a) Ecal RNN.

(b) Ecal CNN.

Figure 8: Confusion Matrices for networks trained on 2 million electron events.

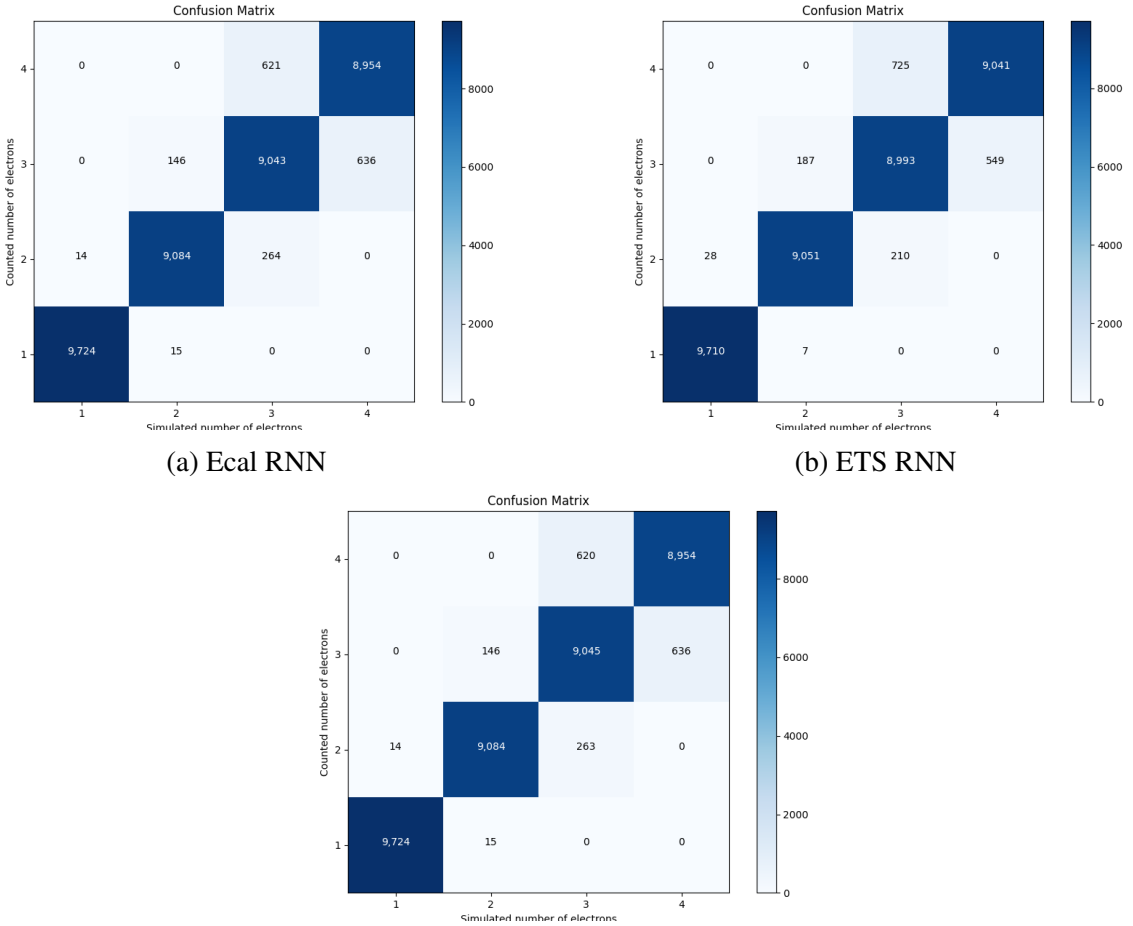
The results for the RNN Ecal network are shown in Figure (8a). The overall accuracy of the network is 0.9545. The network has a tendency to over count electrons and the amount of misclassified events increases with electron multiplicity. This training shows a small overall improvement over the previous result of 0.9470 accuracy [1]. The performance of CNN Ecal network can be seen in Figure (8b). The CNN has an overall accuracy of 0.9911 and a similar tendency to over count the electrons in an event. The CNN showed a significant improvement in accuracy over the previous result at 0.9072 [1].

4.2 Scaling the network, 8 million events data sample

After reproducing the results of of the 2 million event run and fixing the bug related to the trigger scintillator information the data samples were regenerated to include new trigger scintillator information and avoid previously existing corrupted data files. This allowed for ETS and ETSX networks to be trained and tested.

4.2.1 RNN

The training of 8 million RNN networks was done using only a single epoch. A single epoch training for the RNN provides consistent results with accuracy slightly higher than seen in 2 million event training. The performance of Ecal, ETS and ETSX networks on the test dataset is shown in Figures (9a), (9b) and (9c) respectively.



(a) Ecal RNN

(b) ETS RNN

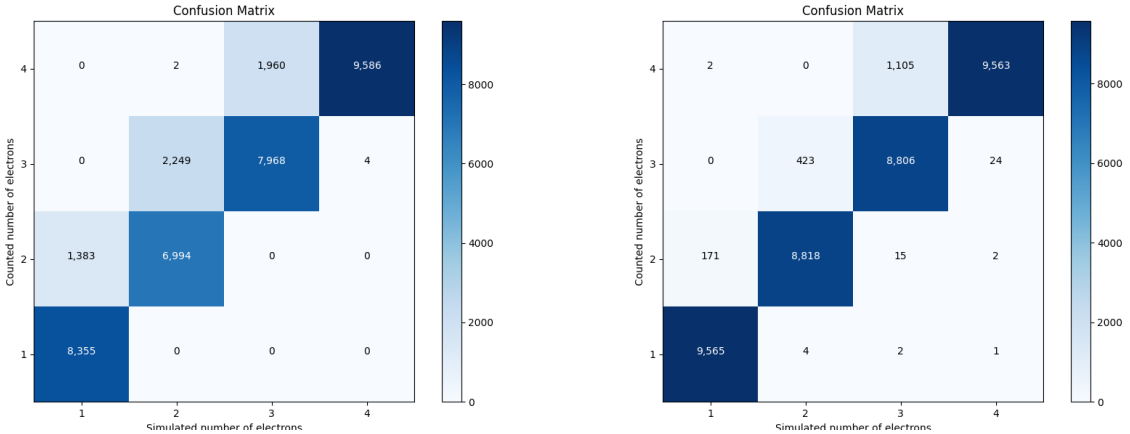
(c) ETSX RNN

Figure 9: Confusion Matrices for RNN trained on 8 million electron events.

Overall the networks performance stayed consistent over the different types of data with accuracy of 0.9598 for Ecal and 0.9557 and 0.9560 for ETS and ETSX respectively. The Networks performance did not differ significantly from the performance of the RNN Ecal trained on 2 million samples which suggests an existence of accuracy plateau after which the performance of the network no longer drastically improves. The use of Trigger Scintillator data did not provide an increase in network accuracy, but on the contrary resulted in slight performance decrease.

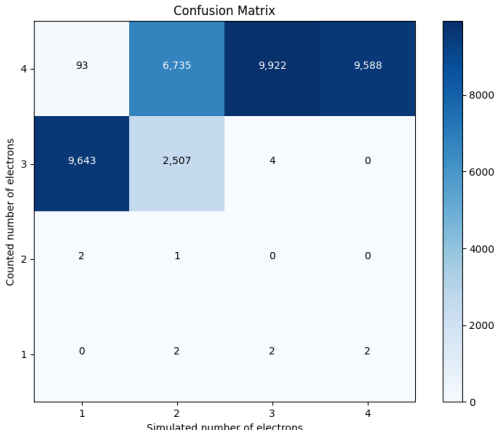
4.2.2 CNN

The various CNN networks trained on the 8 million events sample show a significant disparity in the accuracy of their predictions. The results can be seen in Figures (10a), (10b) and (10c) for Ecal, ETS and ETSX CNNs respectivly.



(a) Ecal CNN

(b) ETS CNN



(c) ETSX CNN

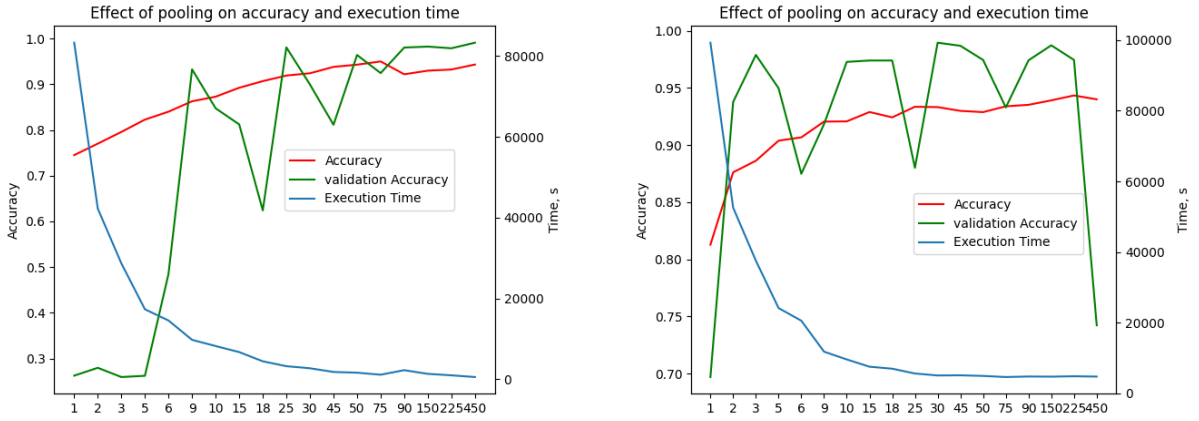
Figure 10: Confusion Matrices for CNN trained on 8 million electron events.

The Ecal CNN results show an overall accuracy of 0.8547 with a strong tendency to over count the electrons in the events. The ETS architecture showed the best performance out of CNN architectures with an accuracy of 0.9547.The results for ETSX clearly indicate a problem with the architecture or training process and will not be included in further analysis. The result is surprising considering the minimal difference of the architecture and training data from the ETS CNNs. Overall the results show that CNN training procedure is inconsistent and the architectures used would require more tuning to achieve better performance. A result of interest is the accuracy decrease that can be seen between the Ecal CNN trained on 2 million and 8 million events, from 0.9911 to 0.8547.

4.3 Pooling results

In the CNN architecture there are 5 MaxPooling3D layers that pool the cells during the training process. The pooling factors on the layers were chosen specifically for the cell dimension being 450. When doing pooling during pre-processing this creates a problem. Either layer pooling factors could be adapted for each pre-processing pooling factor, which would result in an extra variable contributing to possible result inconsistency between the pooled and unpooled network. Alternativly the MaxPooling3D factors can be kept the same. This results in the number of cells being pooled to weird numbers due to rounding after division that results in a non integer value. The third option is to set the MaxPooling3D factors to 1, disabling the pooling on cells during

the training. The last two approaches were used and compared when testing out naively pooled CNNs.



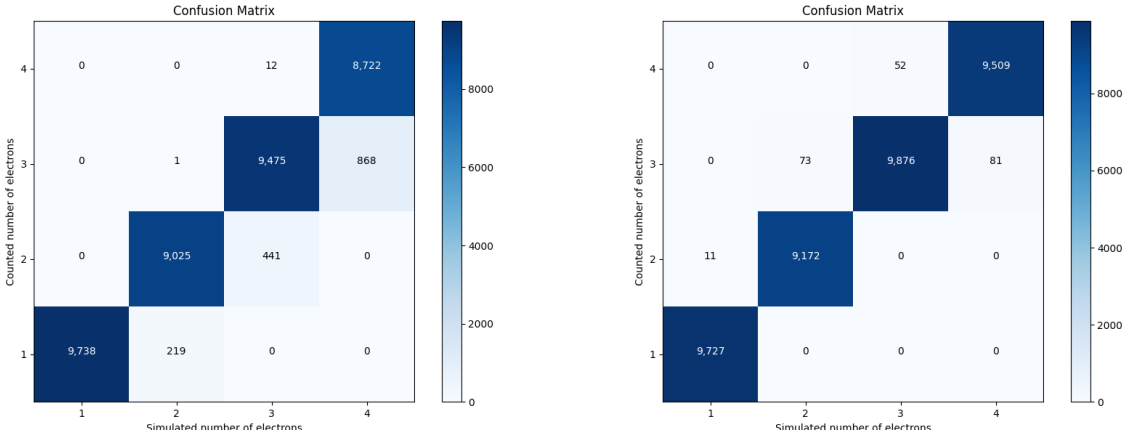
(a) Max Pooling on Cells set to 1

(b) Max Pooling on cells set to [2, 2, 3, 5, 3]

Figure 11: Effects of naive pooling on cells at pre-processing level on accuracy and training time

Figures (11a) and (11b) show the effect of increased pooling on the execution time and accuracy of the networks. The final training (red) and validation accuracy (green) are displayed on the left axis, and the total time needed to train the network over a single epoch (blue) is displayed on the right axis. The results give certain indication on what data is important for CNNs. Figure (11a) shows the performance of various CNNs with MaxPooling3D layers not pooling the cell data. Surprising result is the fact that the network is incapable of making reasonable prediction when the data is not pooled. The validation accuracy stays around 0.25 for pooling rates of 1, 2, 3, 5 and sharply increases to 0.8-0.9 after the data is pooled by a factor of 10 and larger. After this the validation accuracy varies drastically until it stabilizes at pooling rates larger than 90. When MaxPooling3D layer factors are set to [2, 2, 3, 5, 3] the network's performance on data with low pooling factors increases significantly, but drops when the pooling is too large as seen on Figure (11b). An unexpected result of this part of the project was that the networks were able to make good predictions even when data was pooled by factors of 90 and above. This means that the geometric information present is not as important in the training of CNNs and a large amount of training time can be saved when training new networks. Another observation was that the CNNs are not able to train on the data that has high resolution with no pooling layers, which is indicated by bad performance of the first architecture on data with low pooling rate.

After seeing promising results with naively pooled networks a network was trained with pooling based on Trigger Cells as seen in Figure (4b). The trigger pooled data sample mimics the data sample that can be expected at trigger level of the LDMX detector.



(a) Trained on 400 thousand events.

(b) Trained on 8 million events.

Figure 12: Confusion Matrices for Trigger Pooled Ecal CNN

The first attempt at training an Ecal CNN on Trigger Pooled data set gave results seen in Figure (12a). The overall accuracy on the test data set came out to 0.9600. Uniquely, the resulting network showed a tendency to under count electrons in an event, which is the preferred behaviour. Since the results looked promising on a 400 thousand event data set a trial was done on a scaled up 8 million event data set. The models performance came out to 0.9944 with details shown on Figure (12b). The overall accuracy of this model turned out to be the best among all of trained models. The training time of the model also ended up being 60 hours which is a significant improvement over the 400 hour training time needed for CNNs that were not pooled during pre-processing.

4.4 Analyzing Results

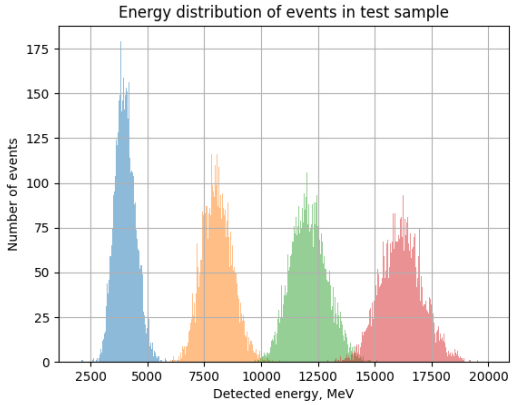
Having a large amount of network with different architecture and trained on differently processed data allows the analysis of misclassified events. Events that are misclassified by multiple networks should have common attributes that lead to the misclassification. In order to explore misclassified events the sets of event misclassified by networks using different information were compared between each other.

(a) RNN on 8 million events sample

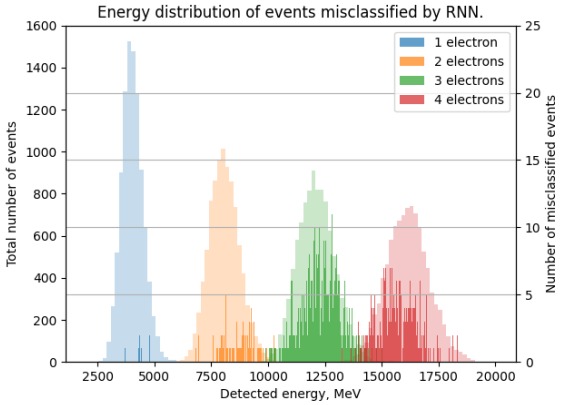
(b) CNN on 8 million event sample

Figure 13: Intersections in misclassified events

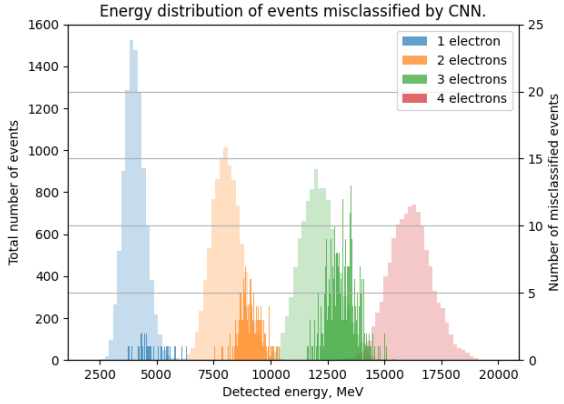
Figures (13a) and (13b) represent the number of events misclassified by the networks based on the model used. When comparing the sets of misclassified events for the RNN networks a significant overlap is found between the three trained networks. As seen on Figure (13a) for each of the networks over 90% of misclassified events are in common. This sends a clear signal that the misclassified events share common qualities that lead to the misclassification by the networks. A surprising result is that 99.87% of the events misclassified by the Ecal RNN network belong to the intersection and including trigger scintillator information in RNN training tends to introduce more misclassified events that are uniquely misclassified by the network type.
The results for CNN misclassified events are seen on Figure (13b). These results show that the misclassifications are not as consistent in CNN, which is in part due to unfortunately low accuracy performance of the Ecal and ETSX CNN models. Interestingly the intersection of misclassified between the networks consists of 1040 events, which is smaller than the intersection of RNNs. This shows that the results are far less consistent between different inputs. Another significant piece of information is that since the Ecal and ETSX networks have a significant over-counting bias, this results in 100% classification for 4 electron events. This creates a boundary problem where it is not known if the 4 electron events would be misclassified if more classification groups existed.



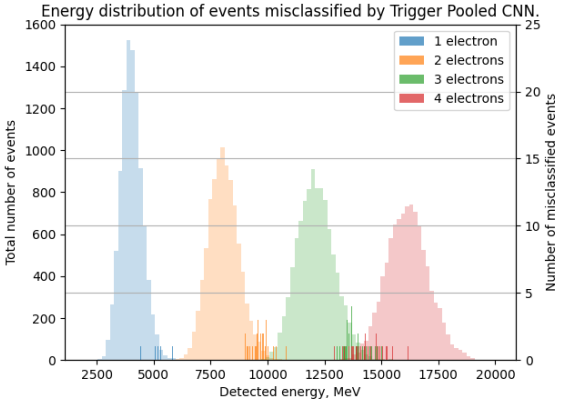
(a) All events in the test sample.



(b) Events misclassified by all 3 RNN models.



(c) Events misclassified by all 3 CNN models



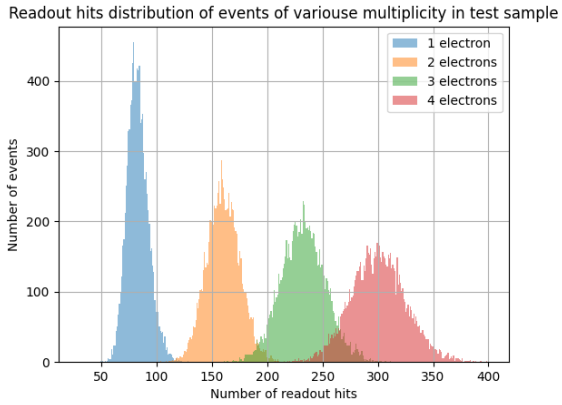
(d) Events misclassified by pooled CNN model.

Figure 14: Energy distributions of misclassified events for electron multiplicity 1-4. Distribution for all events overlaid with semi-transparent colors. Misclassified event count on right axis.

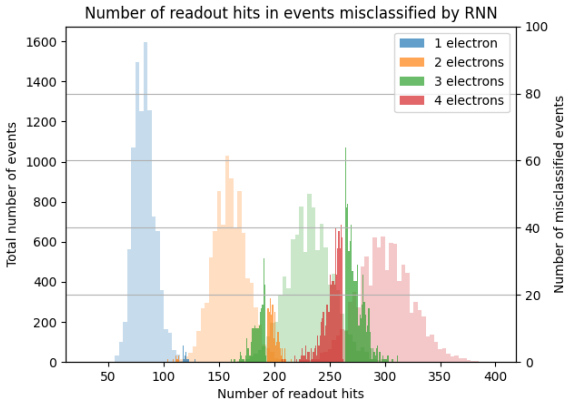
After seeing the result of naive pooling runs, a hypothesis was made that the performance

increased with an increase in pooling rate due to the fact that the geometry of an event was not important and that classification was largely determined by the total energy deposited into the electromagnetic calorimeter. This would mean that the events most commonly misclassified by the network would be found in the energy regions where the energy distributions of various electron multiplicities overlap. A general energy distribution of events in the test sample can be seen in Figure (14a). The plot shows that the energy distribution of various electron multiplicities have a Gaussian shape with the standard deviation increasing with increasing electron multiplicity. The overlap region between the events is almost non-existent for 1 and and 2 electron events however it increases significantly with the increase in electron multiplicity. The plot of events misclassified by the set of RNN networks can be seen in Figure (14b). The histogram shows that event misclassification in RNN are largely independent of energy readout. The histogram largely follows the energy distribution of events of multiplicity 3 and 4. The results for CNN can be seen on figure (14c). The results have a few interesting qualities. Due to over-counting tendency of the CNN network the distribution of 4 electron misclassified events is missing. The three misclassification peaks seen in the histogram all tend to be shifted to the right of test sample energy distribution and lead into the overlap region, meaning that the networks have a tendency to misclassify higher energy events as higher electron multiplicity. Figure (14d) shows the results of the Trigger Pooled CNN network that was trained on 8 million events. It shows that on a successful training run the majority of misclassified events do end up in the energy overlap regions of the distribution. This supports the previously proposed hypothesis of total energy being one of the determining factors in CNN training.

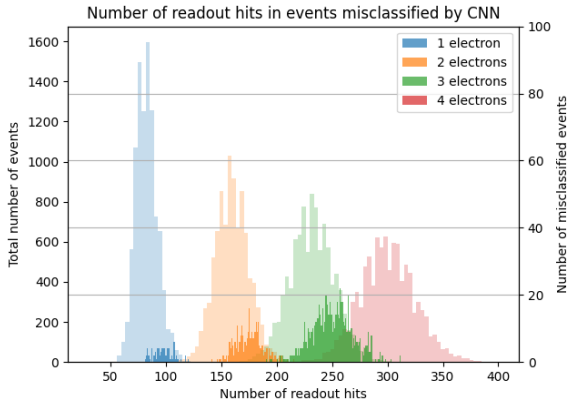
The second event quality of interest is the total number of readout hits. The value represents the number of non-zero energy readouts by the electromagnetic calorimeter in a single event. A large amount of hits could correspond either to a larger electron multiplicity of an event or a large electromagntic shower. A hypothesis for this was that in case of a large electromagnetic shower, the networks would classify an event as having a higher multiplicity. Then the misclassification peaks would be found in the overlap region of the general readout hit amount distribution.



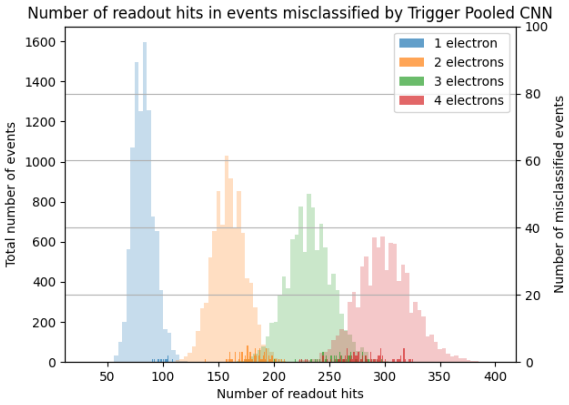
(a) All events in the test sample.



(b) Events misclassified by RNN.



(c) Events misclassified by CNN.



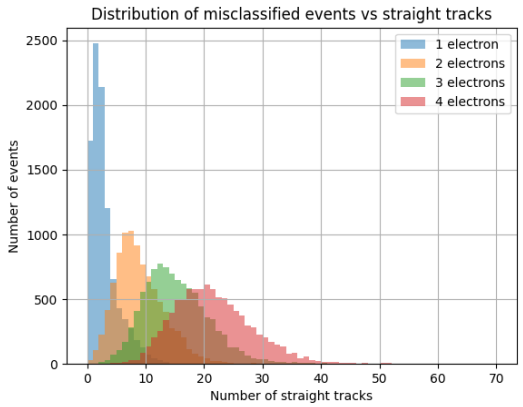
(d) Events misclassified by pooled CNN.

Figure 15: Distribution for number of readout hits in misclassified events for electron multiplicity 1-4. Distribution for all events overlaid with semi-transparent colors. Misclassified event count on right axis.

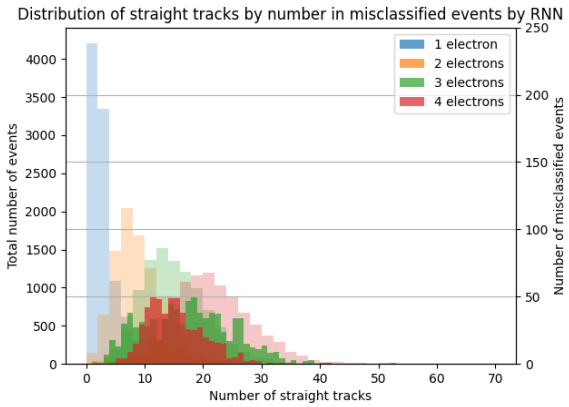
The general readout hit distribution can be seen in Figure (15a) and it in general follows the same pattern as the energy distribution, consisting of Gaussians for each of the electron multiplicities with the standard deviation and overlap region increasing along with electron multiplicity. The overlap regions in this case were larger than in the case of event energy. The misclassified event distribution for RNN networks can be seen on Figure (15b). The histogram shows that the misclassified peaks are clearly aligned with the boundaries of the overlap region of the general distribution, meaning that in case of RNN networks the amount of readout hits is the dominant contributor to the event classification. The CNN results can be found on the histogram in Figure (15c). Similarly to the energy case, the misclassification peaks in the CNN case are shifted slightly to the right of the general distribution and do not include misclassified 4 electron events due to the over-counting bias seen in the models. This shows that the amount of readout hits has an effect on the misclassification in CNN however the correlation is not as strong as in the RNN case. Figure (15d) gives further support to the notion that amount of readout hits is important in the CNN training, but is not the defining factor, the misclassified events are more commonly seen in the overlap regions, but also exist outside of them. The number of hits information is largely lost in the pooling process, it could be that the misclassified events appear more commonly in the overlap region due to a correlation between the number of hits and the total energy, which as we have seen before has an effect on classification in trigger

pooled CNN.

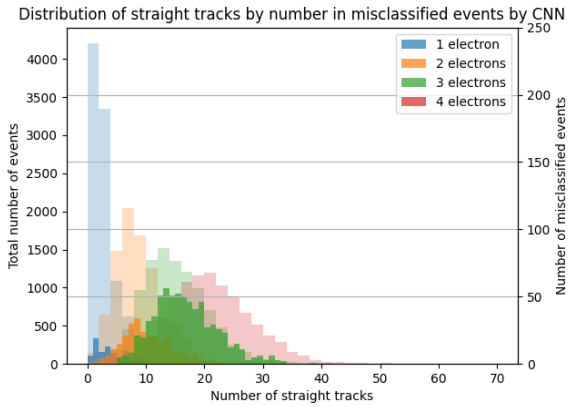
The next explored quantity is of interest as it has not been a part of the dataset directly. During the veto condition processing, LDMX uses algorithms to construct straight tracks through the electromagnetic calorimeter, effectively tracking the path taken by the elements of the electron shower. The number of straight tracks is not used directly as a part of the training data and can be used to represent the geometry of the event. Similarly to the amount of the readout hits, a larger number of straight tracks could correspond to a more spread out electron shower, which can serve as an indicator of a larger electron multiplicity in the event. The reason this quality was chosen is to see if the geometry of the event has a significant effect on the misclassification.



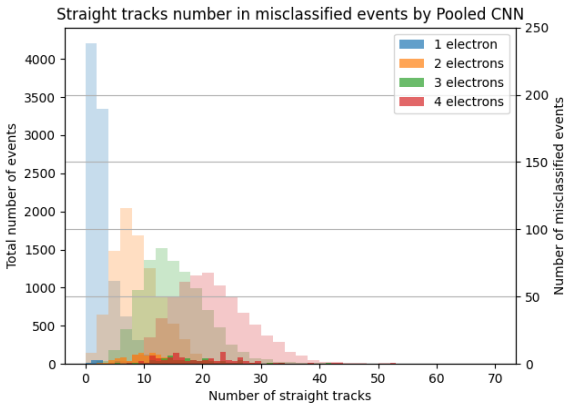
(a) All events in the test sample.



(b) Events misclassified by RNN.



(c) Events misclassified by CNN.



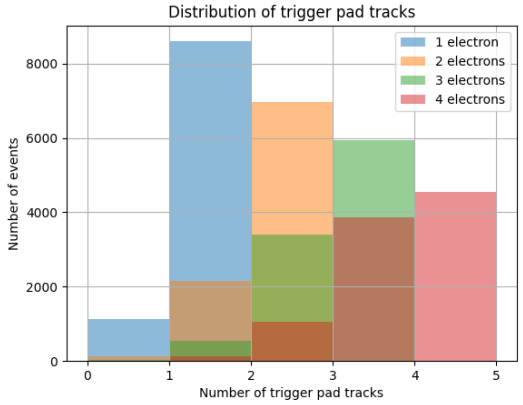
(d) Events misclassified by pooled CNN.

Figure 16: Distribution for number of straight tracks in misclassified events for electron multiplicity 1-4. Distribution for all events overlaid with semi-transparent colors. Misclassified event count on right axis.

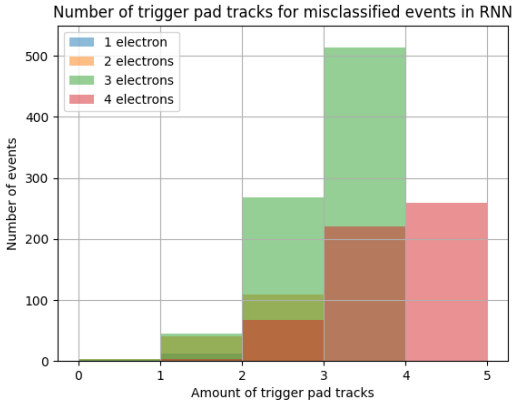
The general distribution of straight track multiplicity in the test sample can be seen in Figure (16a). The histogram shows that there exists a large overlap region between events of different electron multiplicity. The results of RNN model can be seen in Figure (16b). The majority of misclassifications happen in the overlap zone of the general distribution. The two highest peaks correspond to the overlap regions of events with 3 different electron multiplicities. The results for CNN can be found in Figure (16c) and show a similar behaviour to RNN with the peak shifted slightly to the right, likely due to the over-counting issue. The CNN distribution is less broad and has a stronger fall off outside of the overlap region. This could likely be attributed

to the fact that the CNN model learns more about the event geometry compared to the RNN model. Similarly the trigger pooled CNN model, the results of which can be found in Figure (16d), shows that the vast majority of misclassified events are found in the heavy overlap region of various event multiplicities. This does suggest that CNN models are able to pick out some of the event geometry information during training.

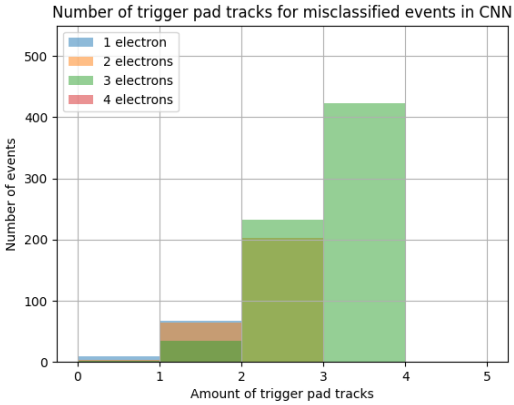
The last part of the analysis revolves around the correlation between current triggering strategy and the results obtained from the trained models. The current strategy uses a number of straight lines that can successfully be fitted to the coordinated of electron hits the electron hits on three trigger scintillator pads. In the simulated data this information is known as trigger pad tracks. The general distribution of the trigger pad tracks in the test sample can be seen in Figure (17a).



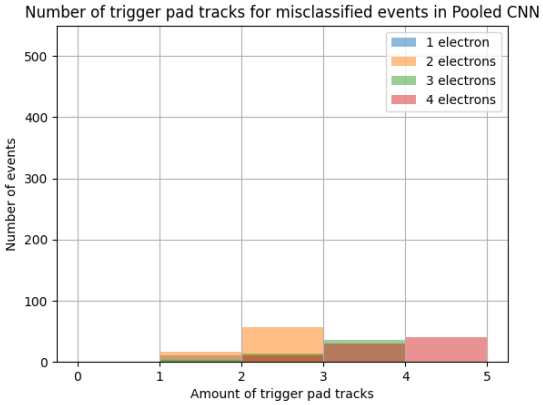
(a) All events in the test sample.



(b) Events misclassified by RNN.



(c) Events misclassified by CNN.



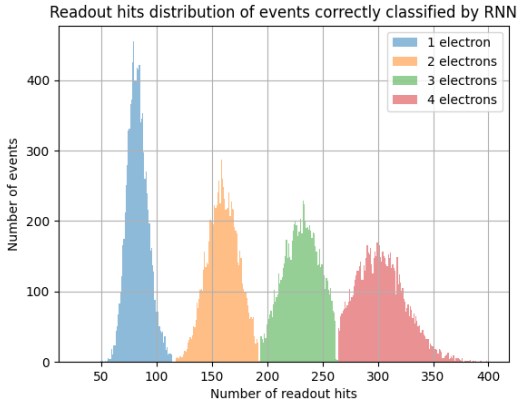
(d) Events misclassified by pooled CNN.

Figure 17: Distribution for number of trigger scintillator tracks in misclassified events for electron multiplicity 1-4. Distribution for all events overlaid with semi-transparent colors. Misclassified event count on right axis.

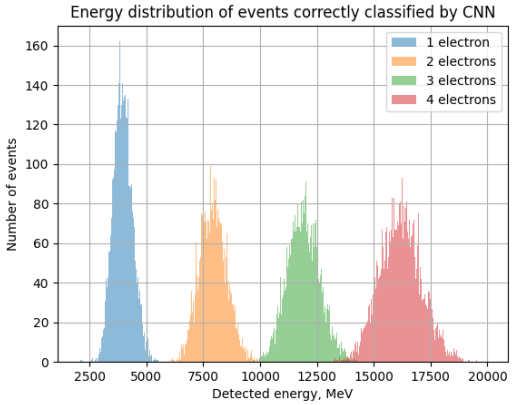
A trend that can be observed in these results is that the number of trigger scintillator tracks tends to be equivalent to or less than the event multiplicity as the current electron counting system is designed to strongly prefer under-counting of the electrons. This however is not the case with the trained models. What can be observed from the Figures (17b), (17c) and (17d) is that there is no real correlation between number of tracks and event misclassification. Majority of misclassifications for a specific event multiplicity are done when the amount of trigger

pad tracks corresponds to the given multiplicity. The plots mostly follow the distribution of misclassified event multiplicities seen in the confusion matrices for the corresponding models. This, again, suggests that the trigger scintillator info does not meaningfully contribute to the classification process.

Throughout the results section the results indicated a strong reliance of the RNN on the number of readout hits and the reliance of CNN on the total energy. In order to look deeper into this reliance plots were made for the correctly classified events.



(a) Distribution of the number of readout hits in events correctly classified by the RNN.



(b) Distribution of total deposited energy in events correctly classified by CNN.

Figure 18: Plots of correctly classified events.

The results seen in Figure (18a) solidify the idea that the RNN is strictly dependant on the number of readout hits during its classification process as there is no overlap in this distribution for events correctly classified by the network. Similarly Figure (18b) shows the CNNs dependence on the total deposited energy, which however is not as strong as in the case of RNN and readout hits. Here it can be seen that the majority of events in the overlap region ended up misclassified with the only exception being 4 electron events bleeding into the 3 electron event energy distribution.

5 Discussion

After training, testing and comparing various ANN architectures trained on differently processed data, a number of conclusions can be made about their performance. The models following RNN architecture gave a consistently good performance with the accuracy averaging at 95%. However a significant problem that can be observed with the RNN architecture was the lack of improvement with an increased training sample. The RNN performance hits a plateau and does not show a significant increase with a a larger data sample. After analyzing the events misclassified by the RNN networks it can be confidently stated that the major determiner in an events misclassification is the number of readout hits in the electromagnetic calorimeter. This is the most likely source of misclassifications in the RNN network and will most likely be hard to overcome without changing its structure. The positives of the RNN network are its consistency in training and shorter execution time. The RNN architectures do not seem to need hyperparameter tuning and multiple epochs of training can be done on RNN architecture while a single epoch is being trained in a non pooled CNN. Additionally it can be said that in the case of RNN networks the inclusion of trigger scintillator data does not contribute to an increase in training performance, but the opposite. The RNN models trained with trigger scintillator information introduce uniquely misclassified events. This is likely a consequence of the architecture's reliance on number of readout hits: the trigger scintillator information could be falsely interpreted as additional hits resulting in an extra set of misclassified events.

For future work with the RNN network a multi-epoch trained model would be interesting to look at. Possibly training on the same data sets multiple times would allow the network to pick up more information about energy and coordinates and thus overcome the misclassifications based on the readout hits. An important step to take in order to make the RNN models more viable is to overcome their reliance on the amount of readout hits in order to classify events. Looking at Figure (14b) it can be clearly said that the RNN does not base its classification on information about energy in the electromagnetic calorimeter. This is likely due to the fact that the Energy component gets dominated by the coordinate components when constructing an RNN hit as seen in Equation (1). In order to overcome this problem the Energy could be given a higher level of importance. In a similar way RNN does not pick up on the provided trigger scintillator information, a possible reason for that is that the way trigger scintillator information is processed currently makes included trigger hits seem too similar to hits in the electromagnetic calorimter. A possible solution to that would be to reserve layer and module 0 for trigger scintillator information like it is done in the CNN pre-processing. A strength of RNN is its ability to process sequential data and analyze time evolution's of provided data, this quality is currently under utilized. A possible way to use this quality is the sort the data by the layer information of the event allowing the network to pick up on energy evolution of the event.

Drawing conclusions about the performance of CNN models is harder than with RNN models. The CNN models trained on the 8 million events data set showed a disappointing result especially considering the performance of the model trained on 2 million events and the trigger pooled model. There is a clear indication that the ETSX model needs hyperparameter tuning as it seems to be unable to make consistent predictions. In a similar manner the Ecal model needs to be tuned to make it perform more consistently. On the positive note, when training is successful, the CNN Ecal models seem to have the most potential of all the models trained during the project. The 2 million event trained model and various pooled models show a vastly improved performance of above 99% correct classification. It looks like the CNN networks depend primarily on the total deposited energy, although a slight correlation can be seen with number of readout hits. The training time is a significant downside of using the CNN models,

however it has been shown that there exist ways to improve the training time without loosing much of performance. The naively pooled models showed an increase in performance with decrease of available data, while the Trigger Pooled model showed better performance than its non pooled counterpart while needing an eighth of training time. A possible explanation of this behaviour is that the non-pooled variant contains information that is too granular to be able to consistently classify events after a single epoch of training. A different explanation is that trigger pooling provides better geometrical information to the system, and because of that, the network becomes more capable despite the smaller amount of information provided to it. A possible improvement to the data being fed into the CNN is to make the current coordinate mappings correspond better to the geometry of the event in Cartesian coordinates. In discussion about Figure (7) it was mentioned that the majority of the energy deposited by the event ends up in module 0, which is the central module of the flower arrangement. This however means that if the electromagnetic shower leaks to a neighbouring module in the Layer-Module-Cell coordinates it will look like a completely unrelated event or even noise. In a similar way, it is likely that the trigger scintillator coordinates are too far away from the event majority in module 0 to be interpreted by the CNN. Thus a suggestion is to change the mapping used by the Layer-Cell-Module coordinates to create events that are more batched together which would allow the CNN to better interpret event geometry and rely less on the total energy and amount of readout hits.

The pooling at the data pre-processing level was a success with multiple future implications for the project. The faster training time and increased performance of pooled models means that it is more efficient to trained pooled models in the future. Similar to the RNN case only a single epoch of training was done on the CNN models. Thus it would be interesting to see performance changes in models trained on multiple epochs. It seems natural to do the multiepoch training on the trigger pooled data set as it is faster, better performing and more indicative of the performance that can be seen in the physical detector.

Overall although the accuracy target of 1 misclassified event in 100000 was not hit, a number of large steps were made towards it. The Ecal CNN network seems to be the most likely to achieve the high accuracy value. The pooling experiment and specifically trigger pooling were largely a success in optimising the training process and model performance. More training could be done to check the viability of various architecture when working with multi epoch training. The input data could be changed to make the architectures less reliant on the number of readout hits and total deposited energy. Lastly a possible prospect for future work is doing a similar study on the performance of Graph Neural Networks and Combined Convolution and Recurrent Networks that were introduced in Jacob Lindahl's thesis[1].

6 Bibliography

References

[1] Lindahl, Jacob. Trigger-Level Multiple Electron Event Classification with LDMX using Artificial Neural Networks, 2023. Lund University, Student Paper, Master Thesis.

[2] Annika H. G. Peter. Dark matter: A brief review, 2012. arXiv:1201.3942[astro-ph.CO].

[3] Gianfranco Bertone, Dan Hooper, and Joseph Silk. Particle dark matter: evidence, candidates and constraints. *Physics Reports*, 405(5–6):279–390, January 2005.

[4] Torsten Åkesson et al. A High Efficiency Photon Veto for the LDMX, 2019. arXiv:1912.05535[physics.ins-det].

[5] Torsten Åkesson et al. Photon-rejection Power of the LDMX in an 8 GeV Beam, 2023. arXiv:2308.15173[hep-th].

[6] Torsten Åkesson et al. Current Status and Future Prospects for the LDMX, 2023. arXiv:2203.08192[hep-ph].

[7] R. Essig et al. Dark sectors and new, light, weakly-coupled particles, 2013.

[8] John Lilley. *Nuclear Physics: Principles and Applications*. Wiley India Pvt. Limited, 2006.

[9] Torsten Åkesson et al. Light Dark Matter eXperiment (LDMX), 2018. arXiv:1808.05219[hep-ex].

[10] Thomas Eichlersmith. Private communication, 2024.

[11] Lene Kristian Bryngemark. Private communication, 2024.

[12] S.S. Haykin. *Neural Networks: A Comprehensive Foundation*. International edition. Prentice Hall, 1999.

[13] Mattias Ohlsson and Patrik Eden. *Introduction to Artificial Neural Networks and Deep Learning*. Lund University, 2023.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[15] Jacob Lindahl. LMDX Machine Learning Project . `https://github.com/Udcstb99/LDMXML`. Accessed: 28-01-2024.

[16] LDMX-Software. ldmx-sw. `https://github.com/LDMX-Software/ldmx-sw`, 2016. Accessed: 13-05-2024.

[17] Brun and F. Rademakers. ROOT - An Object Oriented Data Analysis Framework. In *AIHENP'96 Workshop, Lausane*, volume 389, pages 81–86, 1996. `http://root.cern.ch/`.

[18] Charles R. Harris, K. Jarrod Millman, and Stéfan J. van der Walt et al. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[19] François Chollet et al. Keras. `https://keras.io`, 2015.