Simulating light detection in liquid argon time projection chambers for neutrino and dark matter experiments with deep learning techniques

Enrico Zammit Lonardelli

9910821

School of Physics and Astronomy

The University of Manchester

Masters Project

May 2020

This experiment was performed in collaboration with *Krishan Jethwa*

Abstract

This report details the work done as part of our Masters project, as a continuiation of work done in the first semester. We discuss quantitative comparisons between the prestablished Montecarlo and novel deep learning methods. Furthermore, we present the results of our GAN architecture to learn variables of light intensity S_1, S_2 and f_{200} by implicit learning of their mutual underlying conditional probabilities. We conclude that the results are INSERT HERE QUANTITATIVE MEASURE OF CONCLUSION.

1 Introduction

1.1 The search for signal

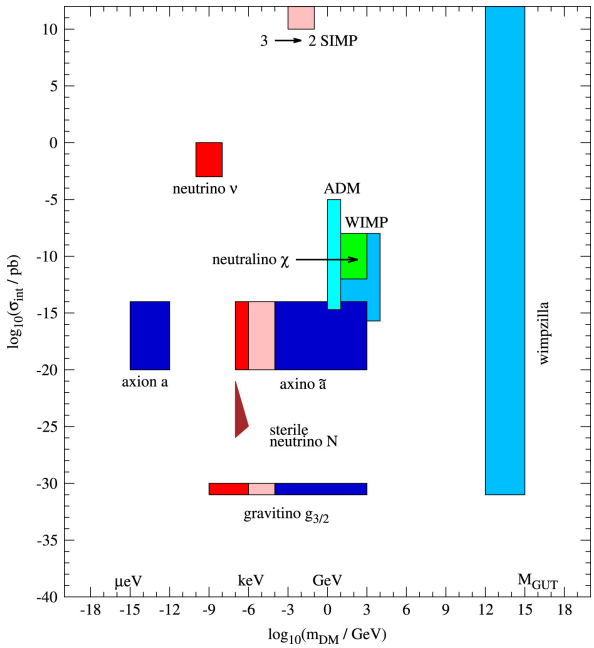


Figure 1: [1] Map of the different masses hypthesised for dark matter candidates amongst many theories.

Cosmological findings have been the driving force for dark matter search for the past X years. The leap to a Weakly Interacting Massive Particle (WIMP) is not a trivial one and must take great care in assumptions it makes and why it makes them, especially in light of increasing ranges of masses excluded by experiments running today. The first feature of importance is thus mass. This is currently under heavy debate in the scientific community as there are supporters of a MILLI mass while on the other spectrum most standard direct detection experiments today are for ranges running from MASSIVE. Evidence from phenomena such as gravitational lensing and the constant rotational velocities of stars in galaxies with increasing distance to their galactic centres suggest a candidate of dark matter halos around these celestial objects.

From supersymmetrical neutralinos to superheavy dark matter candidates we are looking at a range from GeV/c^2 to several TeV/c^2 and even higher in certain theories, see Figure 1. What many of these theories have in common however is that they all produced these WIMP candidates as a bi-product or as required assumptions to allow their theories to work. This then strengthens the theory that such a particle should exist and also what regions of mass, energy and interaction type to look for. These WIMPs are hypothesised to have been in thermal equilibrium with thermal plasma in the early universe. As the universe expanded and WIMP annihilation rate was less than the Hubble expansion rate, relic density for dark matter was reached. This brings us to the cross-sections expected for such WIMPs. Although this varies from theory to theory, we are expecting orders of the weak interaction scale.

This incredibly low interaction rate with regular matter makes it a challenge to detect such

WIMPs. There have been efforts at the Large Hadron Collider to detect missing energies and transverse momenta which could be explained only through a missing new particle in the mass range of a dark matter candidate. Although, to date, these efforts have translated into constraints of cross sections and mass the search is still active.

Another method of search is through indirect detection by observing celestial objects which have a high mass to luminosity disparity. These include but are not limited to galactic centres, dwarf galaxies and close galaxy clusters. This method relies on closely monitoring the particle flux coming from these places waiting for self-interactions or decays into measurable standard model particles to occur. Searches via these methods are made even harder by the fact that the only biproducts most experiments can reliably measure after accounting for interstellar magnetic fields, other celestial objects and low background limists are neutrinos and specifi gamma ray energies.

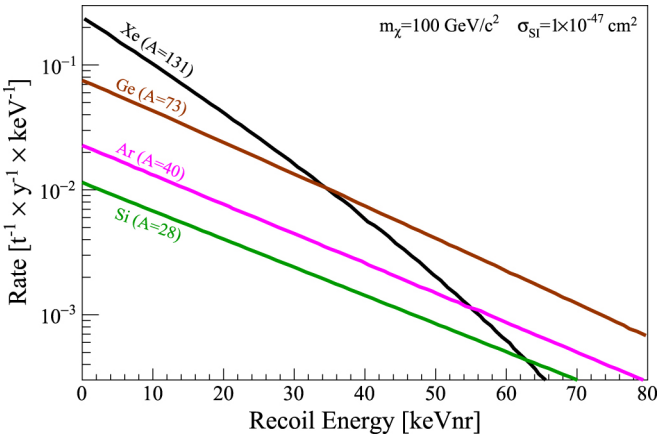


Figure 2: [2] Nuclear recoil spctra for varying noble gas targets highlight the better interaction rate at lower nuclear recoil energies for heavier targets but a lower rate for higher recoil energies.

Finally, the last method of detection is direct detection. Large detector chambers are set up, often many kilometers under the earth, essentially waiting for a WIMP candidate to produce an elastic nuclear recoil with a noble element atom and produce measurable scintillation. For a WIMP mass ranging between 1 GeV/c^2 and 1000 GeV/c^2 the recoil energies are in the range 1-100 keV after which the crossections become way too small for modern detectors. The choice of noble gas element to use is also non trivial since the rate for spin-indipendent interactions increases with nucleon number however decreases at high energies due to form factor suppression, as expressed by
often approximated to [3]

$$\frac{\mathrm{d}R}{\mathrm{d}E_{\mathrm{nr}}} \propto \exp(-\frac{E_{\mathrm{nr}}}{E_0}\frac{4m_\chi m_N}{(m_\chi + m_N)^2}) \tag{1}$$

and shown in Figure 2 for increasing mass of the target nucleus.

The higher interaction rate for lower recoil energies makes it more probable to detect a WIMP candidate interaction however these energies produce a lower intensity of scintillation which results in larger errors (from sources such as photomultiplier calibration, photon efficiency, dark currents) so there is a compromise to be made. These interaction rates are directly relatable

to our study in teaching an algorithm the photon efficiency maps of the detector with varying recoil energies. With the use of Montecarlo simulators such as G4DS one has to incorporate the nuclear recoil spectrum in the simulation setup and the program will sample from this the Ar40 recoils accordingly. This process is a long one since this program simulates everything from the interaction to detection. This process can take several days if running over many energies with all the scintillation being captured.

Similarly, our machine learning algorithm is trained uniformally across the different energies but the choice of dark matter regime to be studied can be changed after training directly through the interaction rate distribution by choosing a suitable nuclear recoil spectrum. This is where the real advantage presented by this deep learning approach comes into play since the training is done once and changing sampling distribution can be done virtually instantly and does not require retraining.

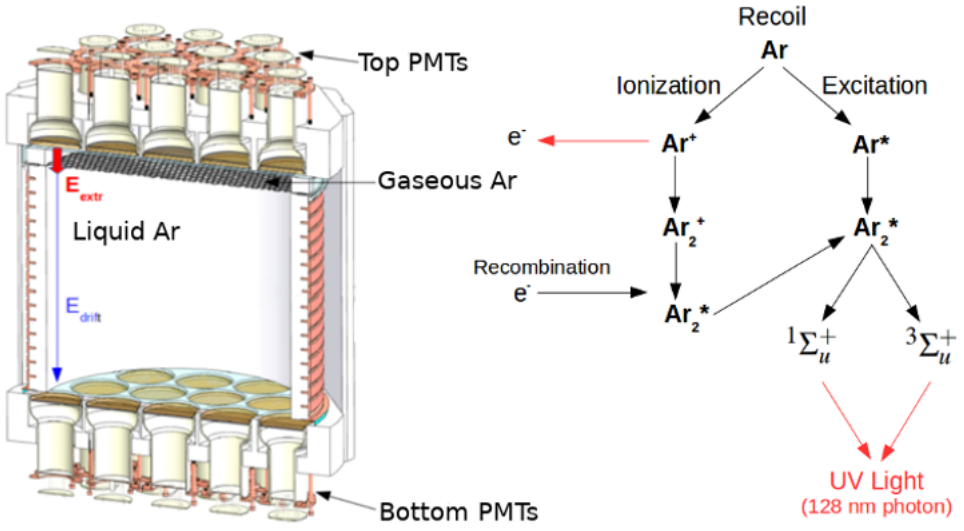


Figure 3: [4] Schematic of a LAr-TPC and processes of VUV photon emmission.

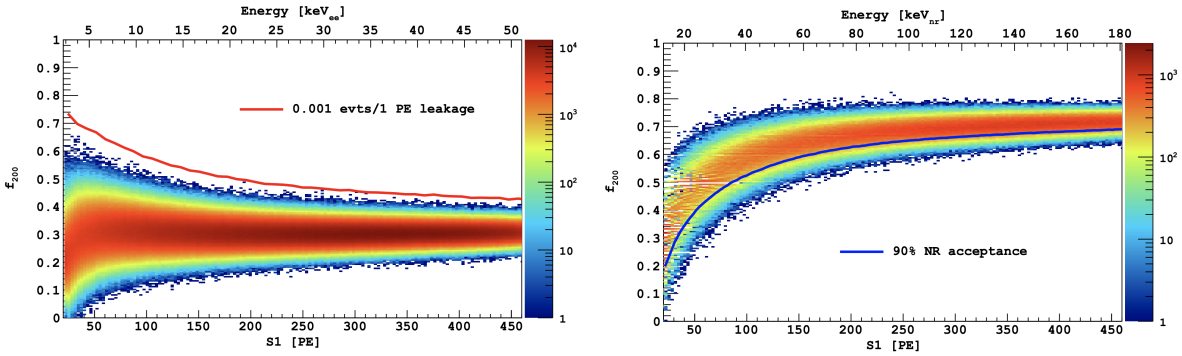
Finally, to understand the variables of interest in this study consider Figure 3. A dark matter particle would enter the detector fiducial volume where it would interact with an Argon nucleus. The nucleus would then become excited and during de-excitation would emit electrons and produce scintillation. This first scintillation is known as S_1 and has a windows of 7μs. After which the electrons would drift upwards due to an electric field within a window of 376μs. The electrons would reach a boundary between Argon in the liquid and gaseous phase which would produce a secondary, much more intense scintillation. Subsequently these photons would be detected by the Silicon PMTs and known as S_2 and this window of scintillation is about 30μs. The ratio of ionization to scintillation is lower for nuclear recoils than for electron recoils and therefore can be used to place selection cuts to increase sensitivity of the detector.

There is a further variable used in the discrimination between background, electron recoils, and nuclear recoil. This is known as the pulse discrimination shape and relates to the de-excitation modes of the Argon nucleus post-recoil. As illustrated by Figure 3 there are two excited states $^1\Sigma_u{}^+$ and $^3\Sigma_u{}^+$. The former has a lifetime of 7ns while the latter has a lifetime of 1600ns.

This difference makes Argon a very competitve candidate as a nuclear target since this same difference in lifetimes between excited states in Xenon is only about 25ns. Although Xenon has other benefits and Argon has other sources of background Xenon based TPCs do not have, for LAr-TPC based detectors this feature is a very good discriminant. This is due to the fact that the ratio of these excited state lifetimes is related to the stopping power or deposited energy per unit path length $\frac{dE}{dx}$ and this differs between electron recoils such as gamma photons, alpha particles and nuclear recoils with argon ion tracks. Thus a parameter called the Pulse Shape Discriminant denoted by f subscripted by the window of interest of time in ns is used. We shall use f_{200} defined as

$$f_{200} = \frac{\int_0^{200ns} \text{Intensity of photons received}}{\int_0^{7\mu s} \text{Intensity of photons received}}. \tag{2}$$

Illustrated in Figure 4 is the simulated difference in Darkside-20k between nuclear and electron recoils for f_{200} against total S_1 intensity. Although normally the parameter f_{90} has been used for experiments such as Darkside-50, for such a much bigger experiment the drift distance is increased substaintially so this is compensated by using f_{200}.



(a) Simulations for Darkside-20k of f_{200} for background data using ^{39}Ar β's. Red line is a leakage curve for a 5-PE requirement on β's.

(b) Simulations for Darkside-20k of f_{200} for signal nuclear recoils. Blue curve is the 90% NR acceptance region.

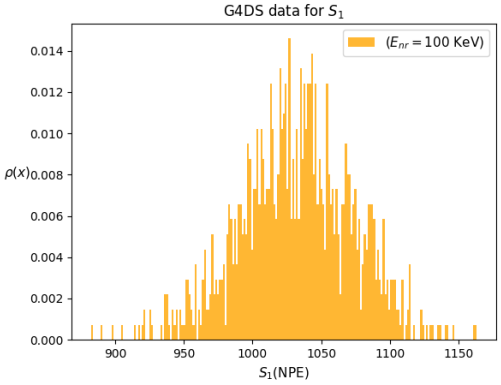
Figure 4: [5] Region of interest using f_{200} pulse shape discriminant against total intensity S_1.

1.2 Current G4DS Results

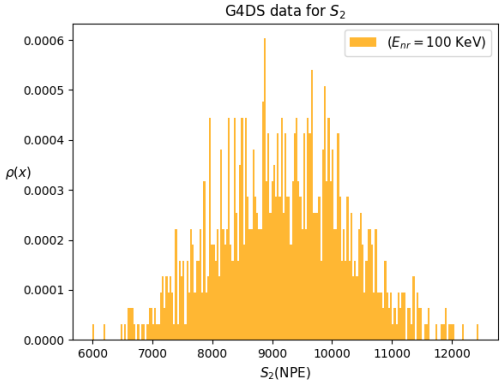
The current simulation methods used by the Darkside collaboration consist of very sophisticated and proven Montecarlo methods. These have been programmed in an opensource program called Geant4 and the complete set of detector macros and routines is called G4DS. For the purposes of this report and our study we ran G4DS with the following configuration detailed in Table 1. Although the default configuration was used, no cuts were made on any of the data for the purpose of simply studying the reproducing power of the deep learning technique. We simulated 1000 uniformly distributed events per ^{40}Ar recoil in the range 5-235 keV in steps of 1 keV. An example for 1000, 100 keV nuclear recoil event is shown in Figure 5 for each of the three variables S_1, S_2 and f_{200}. The data for each variable is similar in shape but the ranges are different and the separation in arithmetic means for the 200 different energies is quite small for f_{200} when compared to S_1 and S_2. What this means is that there might be difficulty in training a neural network to produce such f_{200} distributions on condition of the energy, since they are not very distinguishable from each other and do overlap.

Drift Field	200V
TPC Height	262cm
TPC Width	150cm
Thickness Acrylic Walls	5cm
Thickness LArBuffers	40cm
Thickness Veto Shell	10cm
Thickness TPB	0.1 mm

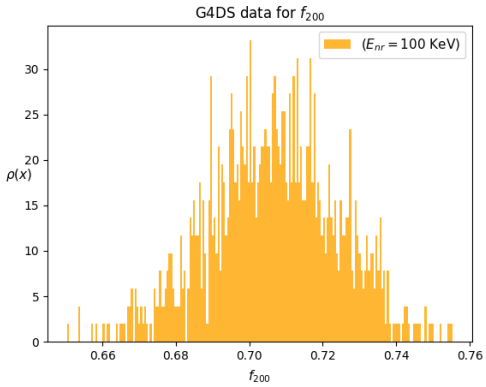
Table 1: Table detailing the major features of the detector setup used in G4DS for the purposes of this study.



(a) G4DS generated data for S_1



(b) G4DS generated data for S_2



(c) G4DS generated data for f_{200}

Figure 5: Example of the generated data for a run of 1000, 100 keV ^{40}Ar recoils in G4DS.

The neural networks described in the next section will therefore be trained on these types of distributions and finally we will want to reproduce plots such as Figure 4 to check that the connection between the three variables has been understood to an extent. To produce a plot for f_{200} vs S_1 first a mass for the WIMP is chosen, in this case we take $\log(m) = 1.5$ which corresponds to a 31.5 GeV/c^2 mass. From theory the corresponding recoil energy spectrum is chosen, shown by Figure 6 for $\log(m)$ =1.5. After sampling 1000 energies these are used to select from runs of 1000 ^{40}Ar recoils in G4DS to then produce the Figures seen later in Section 3.

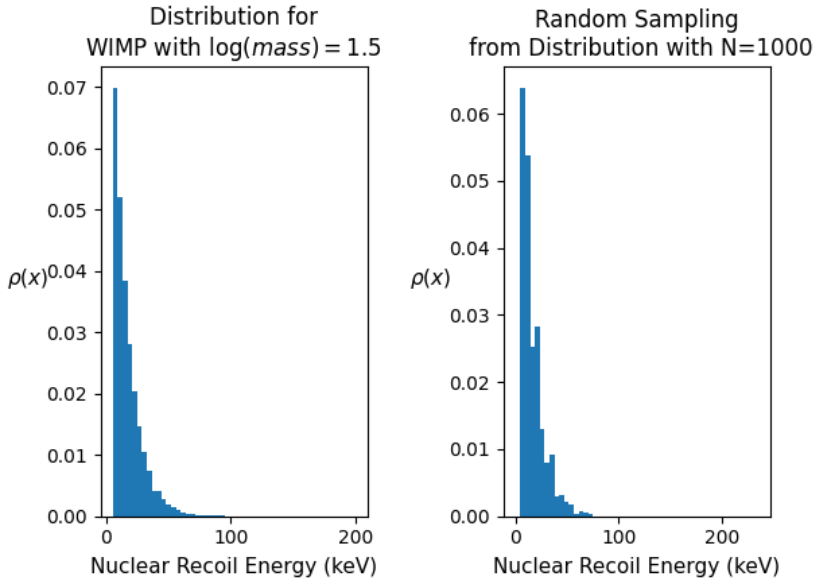


Figure 6: A plot of the recoil energy spectrum from theory vs the sampled N=1000 energies spectrum used to work for mass of WIMP $\log(m) = 1.5$.

1.3 Deep learning as an alternative

As discussed before, the results obtained with G4DS are well understood and accepted as good simulations (backed up by experiments such as SCENE, DS50). However, with great detail come large amounts of time waiting for the simulations to complete. Moreover anytime the nuclear recoil spectrum is changed to simulate different masses of WIMPs, the simulations must be run again. This hinders progress thatg can be made to an extent since instead of being able to test multiple theories, one must take great care to compromise with time spent waiting. Lastly, a solution to this could be increasing the number of processors, GPUs and computing capabilities but these cost a lot of money and do not necessarily sale linearly. It would also be improbable for an institution or a department with a fixed budget it must stick to, to assign more and more resources to one research group which means this is not a sustainable solution.

This is where neural networks appear as a possible solution to this problem. This will not be an introduction to machine learning techniques, rather for that please refer to INSERT HERE. A neural network is a set of connected nodes in different layers. Each layer contains many nodes connected to other nodes from successive and previous layers. Each of these nodes carry intermediary weights to certain functions of output. In a forward run, the input layers (containing the data to be trained) are connected to intermediary layers whom carry out some transformation or apply a so called 'activation function' to give the intermediary weights some values. This is repeated until the final layer is reached which will usually have an activation function which is dependent on the type of problem at hand. For example, a classification problem will have weights representing each category which will be largest for the category the algorithm classifies the input as and lower or 0 for the others. These final weights are then compared to what they should be from the known labelled training data and a quantity to measure 'goodness' of the algorithm is set. The function doing this measure is called the loss function. Acting on this

number will be an optimzer which then changes the intermediary weights accordingly so that ideally on the next run the new weights and activation functions will guide the algorithm towards a better final weights which will minimze loss and maximise accuracy.

This is actually only one type of machine learning known as supervised learning. The other kind, unsupervised learning, is what we carry out in the algorithms detailed in this report where we essentially implement dimensionality reduction. Particularly, we make use of a rather new method of machine learning known as Generative Adverserial Networks. In our case we have two neural networks, one known as the classifier and the other as the generator. The aim of the generator is to reproduce training data as close as possible while the job of the classifier is to spot at each iteration of training (known as an epoch) which of the two inputs it is being presented by, real or generated. Their loss functions, for a fixed generator G and the optimal discriminator $D^*_G(\bar{x})$ is given by minimizing the function

$$C(G) = \mathbb{E}_{x\sim p_{data}}[\log(D^*_G(\bar{x})] + \mathbb{E}_{x\sim p_g}[\log(1 - D^*_G(\bar{x})] \tag{3}$$

where the optimal discriminator is described by

$$D^*_G(\bar{x}) = \frac{p_{data}(\bar{x})}{p_{data}(\bar{x}) + p_g(\bar{x})}. \tag{4}$$

For the ideal case where the generator is perfect the accuracy of the discriminator results in around 0.5 as it would have no real clue other than a 50-50% chance to tell the difference.

2 Methods

2.1 Improvements over previous work

In the first semester, a qualitative approach was taken to this problem of trying to reproduce G4DS dat through a GAN. Moreover, we required a cGAN, or conditional GAN, which would accept as a condition the nuclear recoil energy E and produce the corresponding S_1, S_2, f_200. The results of that report showed promising reproductions of the real data but there were two main problems with that approach:

i) There was barely any quantitave analysis of the reproducability of the GAN with respect to training data. This was done mainly by visually looking at any two same energy plots for a particular variable since the main aim was to check that we had an algorithm that worked to some extent.

ii) As we trained the GAN to learn $P(S_1|E)$ and then $P(S_1 \cap S_2|E)$ and so on the network needed was exponentially larger and more complicated. It was not only a matter of adding more layers or nodes but required for each variable a substantial rethinking of the networks (generator and discriminator) as a whole every time.

iii) The 3D, 1 conditional GAN was not able to learn the 3 variables at once and we had to abandon this path quickly since it did not produce any returns on effort put in.

2.2 Novel techniques

2.2.1 Wasserstein GANs

As a result of this we expanded our search for a different architecture of GANs. We found wGANs, or Wasserstein GANs, to be the next architecture we would try in this last semester. The main feature of this GAN is that it uses a different loss function, or way of quantifying how bad the generator output is from the real data. The problem with this is that although initially we thought this difference was only about modifying the loss function and some minor tweaks in the code, initial results quickly highlighted this was fundamentally a different architecture we were dealing with. This would have meant spending close to the same time as the first semester, trying to get essentially the same point we already had got. This architecture promised better convergence in less epochs and is proven to be much better in terms of performance than the minmax approach taken by the underlying loss function of the traditional GAN but this was not the aim of what were doing. We thus decided to set focus our efforts on other architectures. Finally, here we are not saying wGANs are not worth implementing, rather from what we observed during proof of concept training they are very efficient but quickly end up in failure modes if not configured well. So we do suggest this architecture be studied in the future, perhaps in the next iteration of this project since it could potentially save many epochs worth of work and achieve evene better convergence.

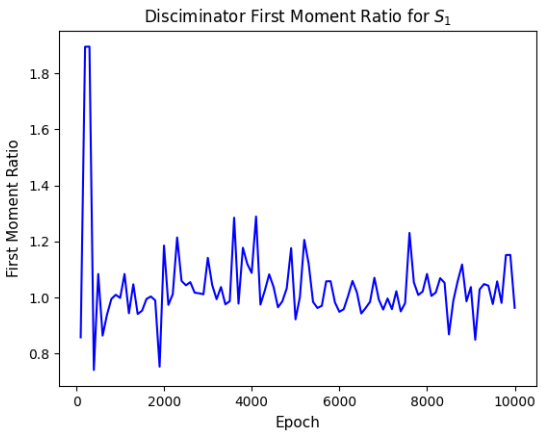
2.2.2 ARGANs

AutoRegressive GANs have been first called so in 2018 but the underlying theory has been used in the context of machine learning for decades now. Although mostly used in the ambit of image recognition the method still applies for our problem. In reality, the ARGAN proposed in 2018 actually made use of modeling data in the latent (feature) space rather than data space like the older employers of this method have. The closest to what we have done was done in 2011 when a GAN was used by training it in an image generation context. Each pixel was trained individually with the condition of all its preceding pixels learnt. Similarly the probability we are trying to teach our generator in this method is given by

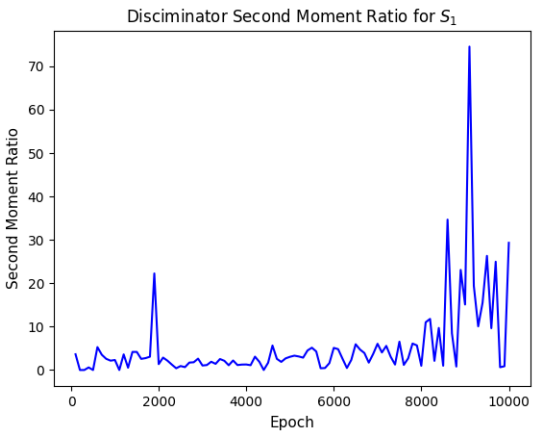
$$p(\bar{x}) = \prod_{i=1}^{n} p(x_i|x_0, ..., x_{i-1}) \tag{5}$$

where in our case i runs from 1 to 3 for S_1, S_2, f_200 and $i = 0$ is the recoil energy E. More explicitly, $P(S_1|E)$ will be taught for s epochs (usually 10,000 in this report) and then the best output of that generator will be passed to $P(S_2|S_1, E)$ to be trained for a further 10,000 epochs and again to obtain $P(f_{200}|S_2, S_1, E)$. The massive advantage of this is that essentially we are training 1DcGANs (where the dimension is the variable being taught) and by the end of the process we will have trained the three variables with their joint porbabilities, all from a single energy input. Moreover, the complexity of this architecture is comparable to the 1D vanilla cGAN rather than the 3D cGAN. Lastly, in my personal opinion as an advocate for scalability and future-proofing, we are making sure that if the collaboration's needs are ever changed so as to require the need to train a further variable, this can be easily done. Unless this variable is wildly different than the three already taught chances are that a very similar neural network can be used.

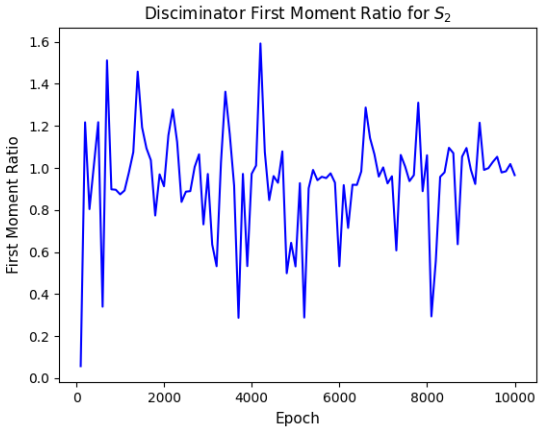
2.2.3 New Metric: Moments



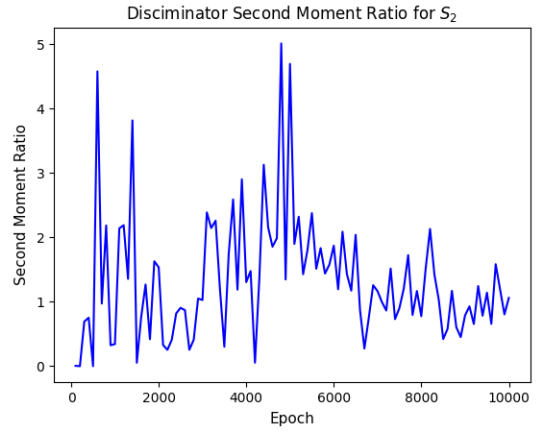
(a) G4DS.



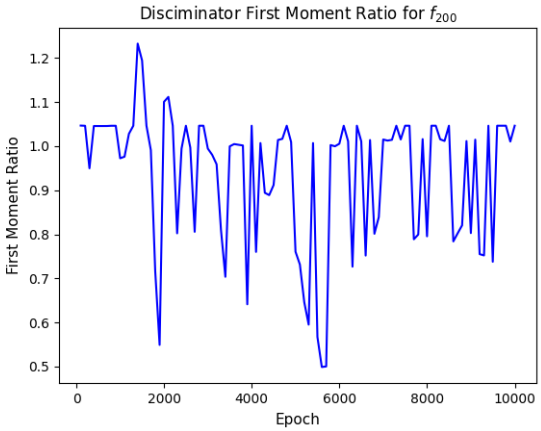
(b) G4DS.



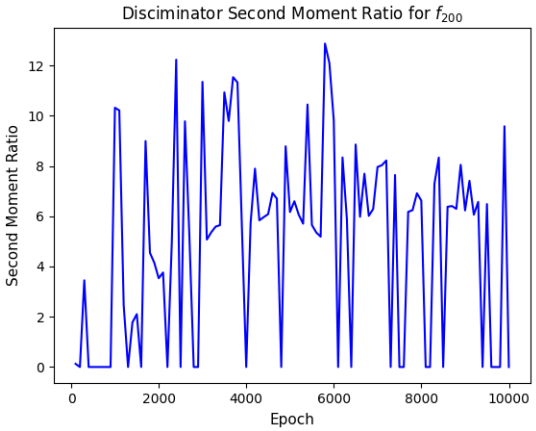
(c) G4DS.



(d) G4DS .



(e) G4DS.



(f) G4DS.

Figure 7: Use of moments during training drasticlly captures the actual performance of the GAN rather than accuracy-loss.

2.2.4 Work Pipeline

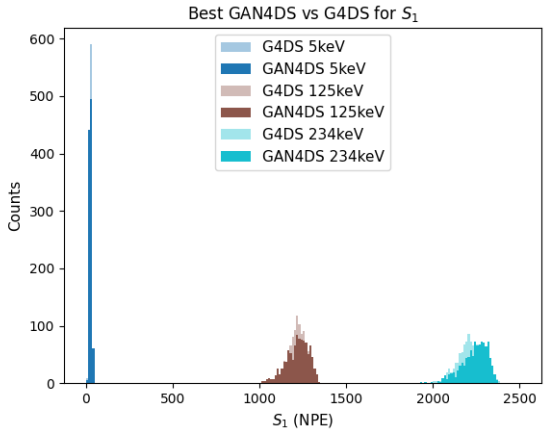
Before presenting results, I thought it might be worthwhile briefly mentioning changes to our coding style and the way we completely redfined our work pipeline. Last semester we found working on an online, free platform provided by Google called Colab helped us work together thus sharing the same results and make use of the Python Jupyter notebooks with NVIDIA T4 GPUs which have a 16GB memory needed for the large dataset of training we had with 3 variable, 1000 events per energy with 230 energies. We found however that we easily ended up with tens of notebooks, each with a different architecture and needing to manually save outputs. Consitency between different notebooks was lacking at best and the biggest drawback was that the session would timeout after a while so the network could not be let to train for long periods of time reliably.

We therefore took the opportunity to convert all of the code to a single Python project to be run on the University of Manchester Physics Department GPU cluster. The package we produced aims to really focus on being able to be used i nthe future by people that might not be aware of the inner workings of machine learning and instead changes the focus on what the researcher wants to train and what they would like to see it output. The program takes in the variables and the training dataset (after self-extracting from ROOT files of $\approx$ 1GB per energy to $\approx$ 20kB) and a layout specified in a markup language independent from the Python code. The program then automatically splits up the data in as many batches as needed to serve memory requirements, creates the ARGAN structure, opens a monitorable Tensorboard session and saves all plots and data logs in a single, consitent format. Tensorboard is a package provided by Google, the creators of Tensorflow, which allows monitoring machine learning progress to the level where one can monitor the weights of each individual layer to see how they change with epochs. We actually made use of this to remove any layers or nodes which were essentially useless or redundant or in certain cases were damagin performance and convergability. We made this codebase in the hope it could serve as a baseline for future teams potnetially working on continuing our work and encourage to expand its capabilities.

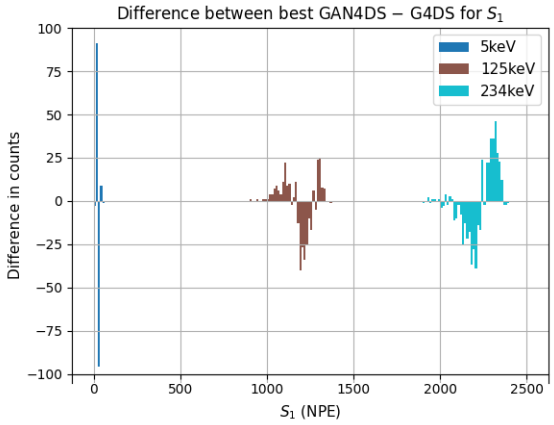
3 Results

3.1 Individual Variables

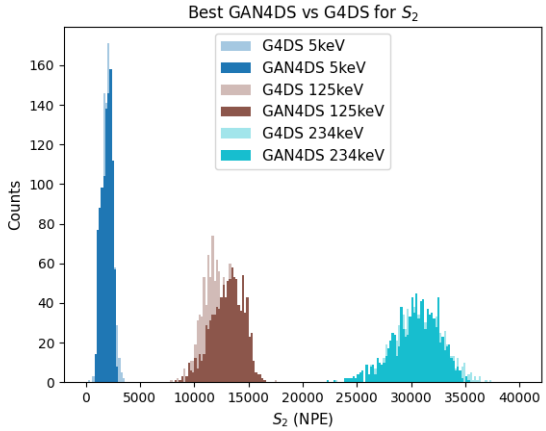
All results unless specified were obtained by using an ARGAN with 10,000 epochs per variable. The dataset is comprised of 1000 ^{40}Ar recoils per nuclear recoil energy in the range [5,235] keV in steps of 1 keV. Figure 8 shows (left) the results of 3 samples (min,middle and max in recoil energy domain) of training dataset overlayed with the best GAN output per variable and (right) the differences for each two equal energy, equal variable counts.



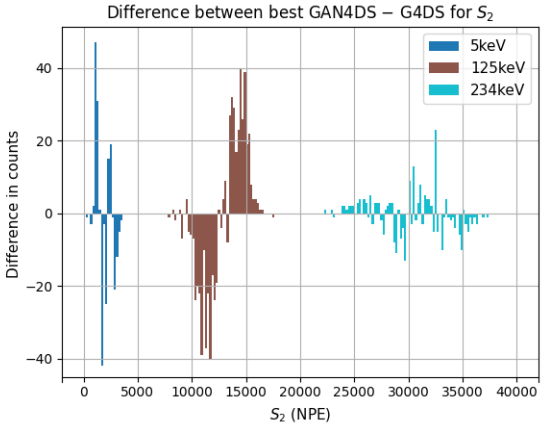
(a) G4DS.



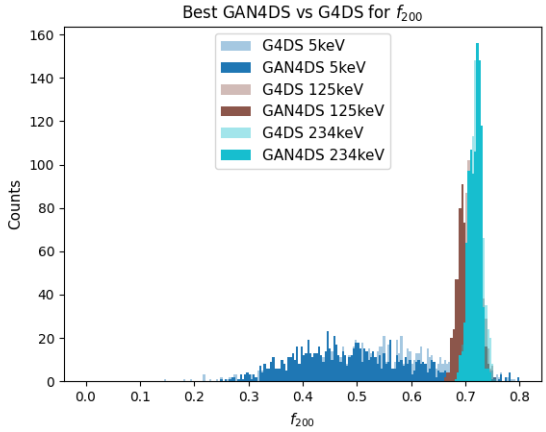
(b) G4DS.



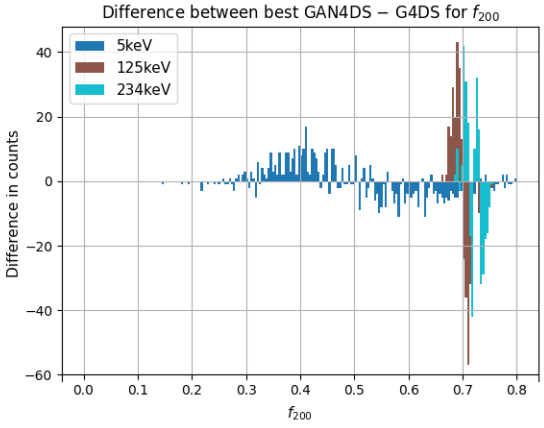
(c) G4DS.



(d) G4DS .



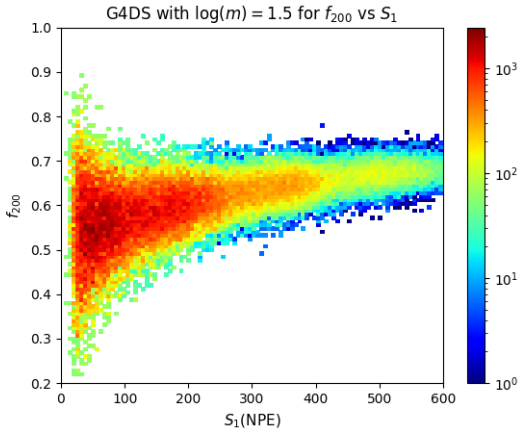
(e) G4DS.



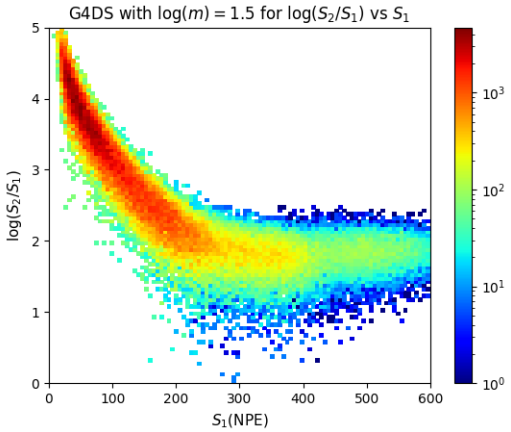
(f) G4DS.

Figure 8: Individual variables learnt successivly with each variable adding itself as an input condition to the next variable being trained. Direct comparisons between generated and trained data to the left, differences per bin on the right.

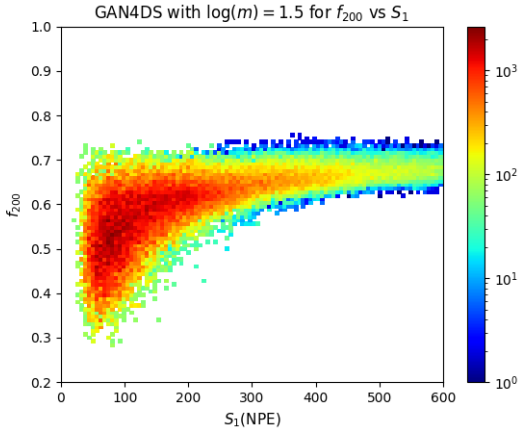
3.2 Variable Correlations



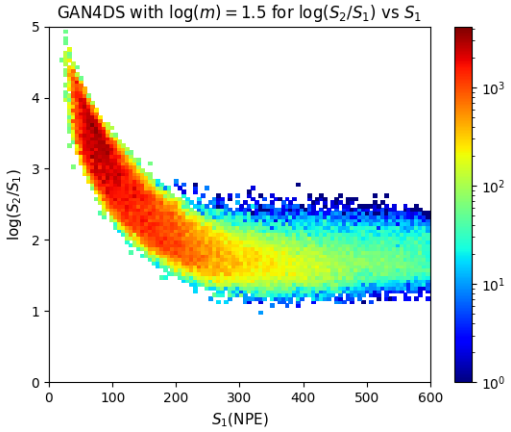
(a) G4DS generated plot for f_{200} vs S_1. As in Figure 4 there is a band of mean value close to $f_{200} = 0.7$.



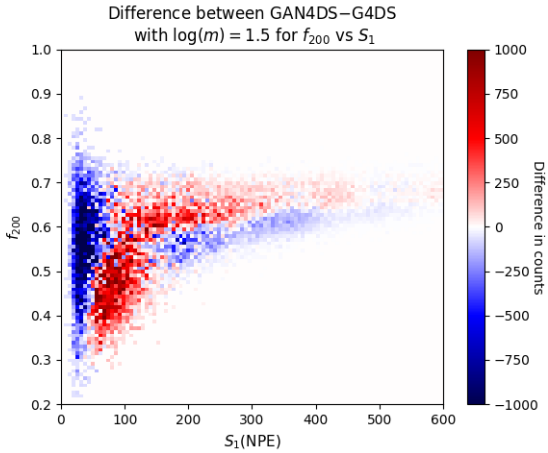
(b) G4DS generated plot for $\log(S_2/S_1)$ vs S_1 as another discriminant between signal and background.



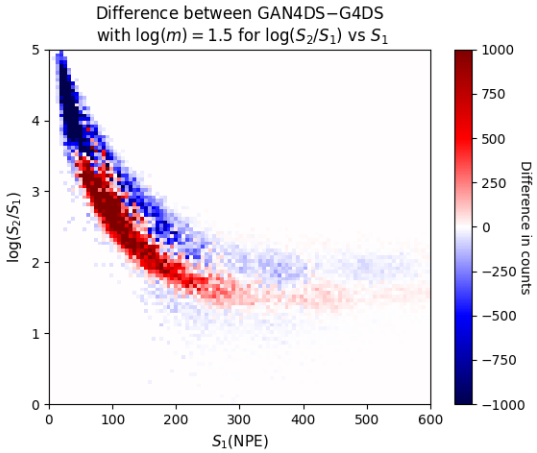
(c) Reproduced plot by GAN4DS through learning individual variables



(d) Reproduced plot by GAN4DS through learning individual variables

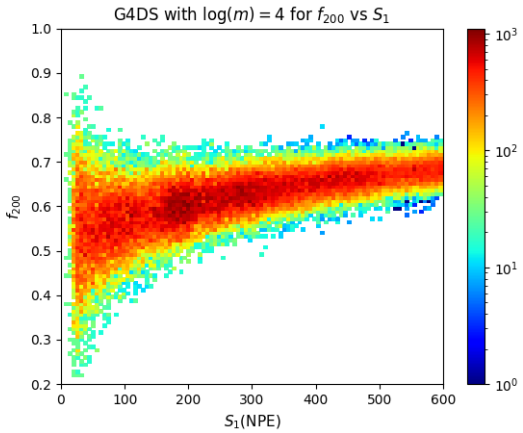


(e) Difference between per bin GAN4DS and G4DS

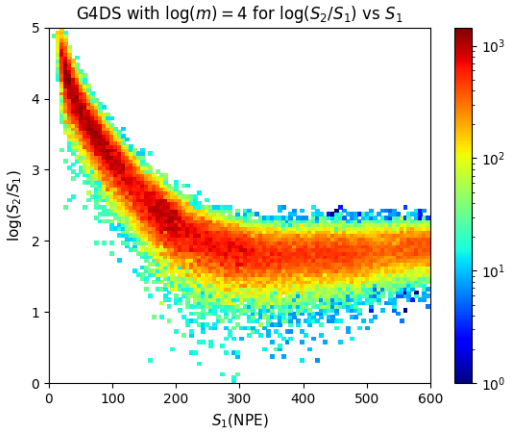


(f) Difference per bin between GAN4DS and G4DS

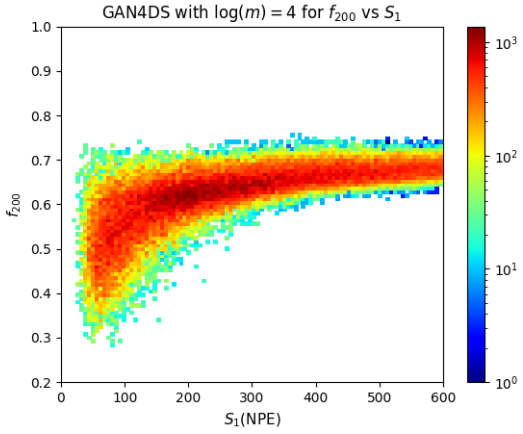
Figure 9: Results showing correlation learnt between the different variables by GAN4DS for $\log(m) = 1.5$.



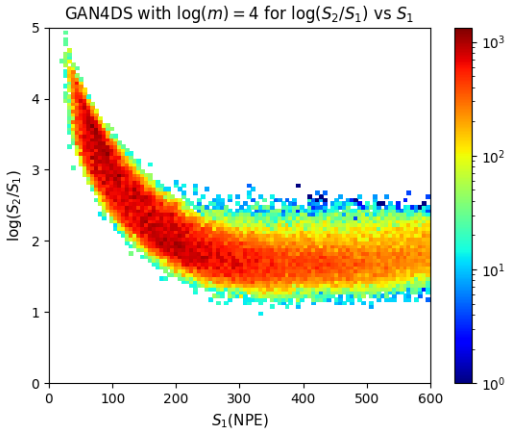
(a) G4DS generated plot for f_{200} vs S_1.



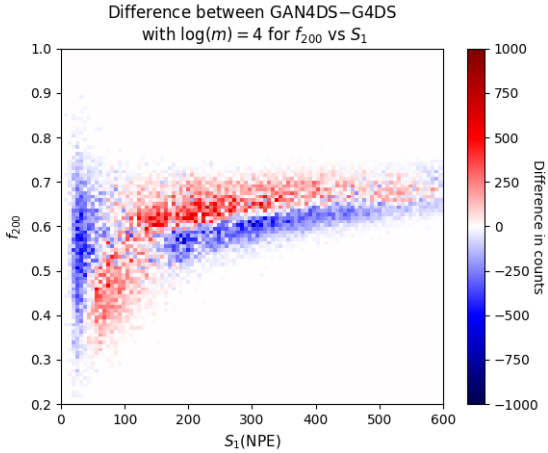
(b) G4DS generated plot for $\log(S_2/S_1)$ vs S_1.



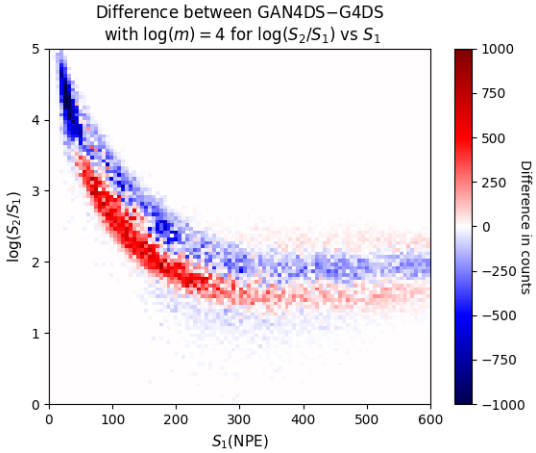
(c) GAN4DS generated plot for f_{200} vs S_1.



(d) GAN4DS generated plot for $\log(S_2/S_1)$ vs S_1



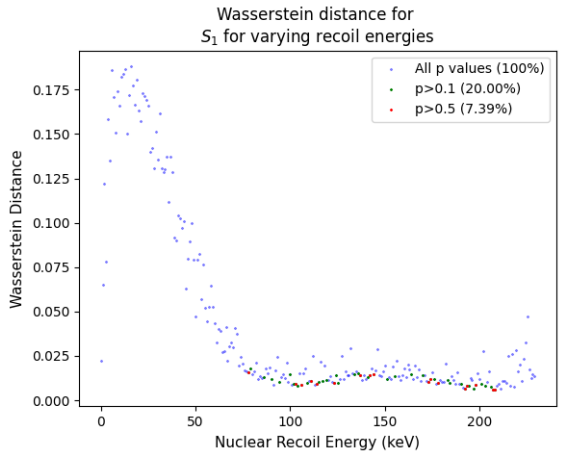
(e) Difference per bin between GAN4DS − G4DS generated data.



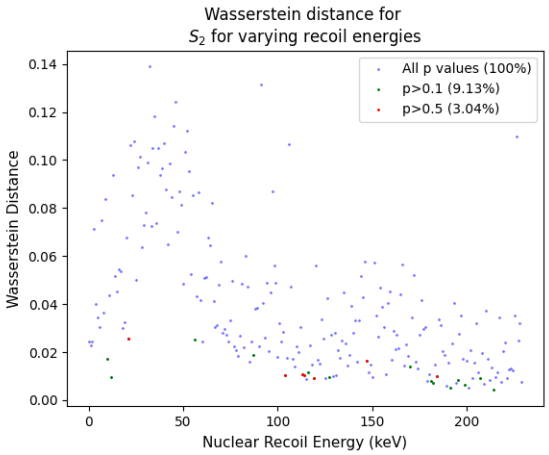
(f) Difference per bin between GAN4DS − G4DS generated data.

Figure 10: Results showing correlation learnt between the different variables by GAN4DS for $\log(m) = 4$.

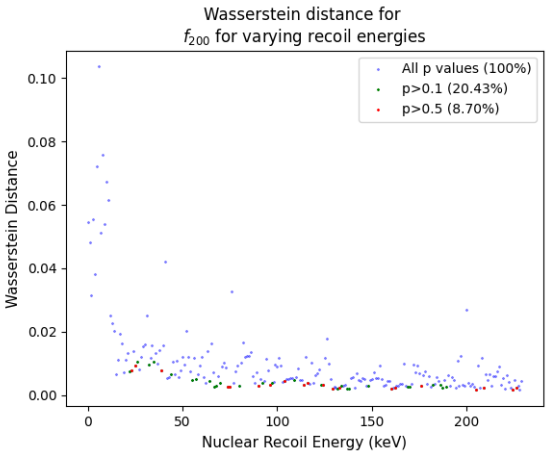
3.3 Accuracy Analysis



(a) G4DS.



(b) G4DS.



(c) G4DS.

Figure 11: Individual variables learnt successivly with each variable adding itself as an input condition to the next variable being trained. Direct comparisons between generated and trained data to the left, differences per bin on the right.

4 Final Remarks

References

[1] H. Baer *et al.*, "Dark matter production in the early universe: Beyond the thermal wimp paradigm," *Physics Reports*, vol. 555, pp. 1–60, 2015, Dark matter production in the early Universe: Beyond the thermal WIMP paradigm, ISSN: 0370-1573. DOI: https://doi.org/10.1016/j.physrep.2014.10.002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0370157314003925.

[2] M. Schumann, "Direct detection of WIMP dark matter: Concepts and status," *Journal of Physics G: Nuclear and Particle Physics*, vol. 46, no. 10, p. 103 003, Aug. 2019. DOI: 10.1088/1361-6471/ab2ea5. [Online]. Available: https://doi.org/10.1088%2F1361-6471%2Fab2ea5.

[3] J. Lewin and P. Smith, "Review of mathematics, numerical factors, and corrections for dark matter experiments based on elastic nuclear recoil," SCAN-9603159, Tech. Rep., 1996.

[4] E. E. Edkins, "Detailed characterization of nuclear recoil pulse shape discrimination in the darkside-50 direct dark matter experiment.," PhD thesis, University of Hawaii at Manoa, 2017.

[5] Aalseth *et al.*, "Darkside-20k: A 20 tonne two-phase lar tpc for direct dark matter detection at lngs," *The European Physical Journal Plus*, vol. 133, no. 3, p. 131, 2018.