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 in the second semester as a continuation of work done in the first semester [1]. We discuss quantitative comparisons between the prestablished Monte Carlo package known as G4DS and novel deep learning methods with an AutoRegressive Generative Adverserial Network (ARGAN) known as GAN4DS. Furthermore, we present the results of GAN4DS on variables of light intensity S_1, S_2 and f_{200} by implicit learning of their mutual underlying conditional probabilities. Discriminating plots between nuclear and electron recoils of f_{200} vs S_1 and $\log(S_2/S_1)$ vs S_1 are also presented.

1 Introduction

1.1 The search for signal

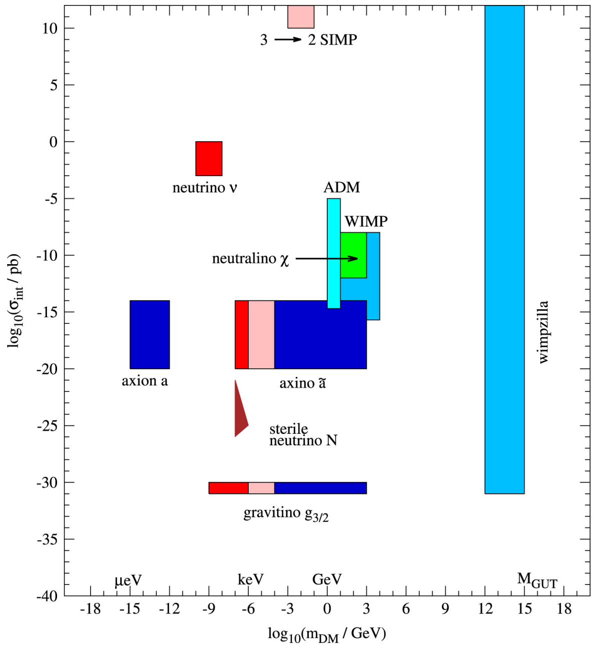


Figure 1: [2] Currently hypothesised masses for dark matter candidates from different theories.

Cosmological findings have been the driving motivation behind dark matter search for the past 90 years [3]. The leap to a Weakly Interacting Massive Particle (WIMP) [4] is not a trivial one and one must take great care in the assumptions [5] it makes and why it makes them, especially in light of increasing ranges of masses being excluded by experiments running today. The first defining feature is thus mass. This is currently under heavy debate in the scientific community as there are supporters of a very low mass dark matter candidate (such as axions [6] or sterile neutrinos [7]) while on the other side of the spectrum most standard direct detection experiments today [8] [9] are for mass ranges running from tens to hundreds of GeV/c^2. Evidence from phenomena such as gravitational lensing [10] and the constant rotational velocities of stars in galaxies with increasing distance to their galactic centres [11] 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 [12], see Figure 1. What many of these theories have in common however is that they all produced these WIMP candidates as a byproduct or as required assumptions to allow their theories to work. This strengthens the theory that such a particle should exist and 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 [13] [14] 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 [15] by observing celestial objects which have a high mass to luminosity disparity. These include but are not limited to galactic centres, dwarf galaxies and galaxy clusters close to ours [16]. 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 byproducts most experiments can reliably measure after accounting for interstellar magnetic fields, other celestial objects and low background limits are neutrinos and specific gamma ray energies.

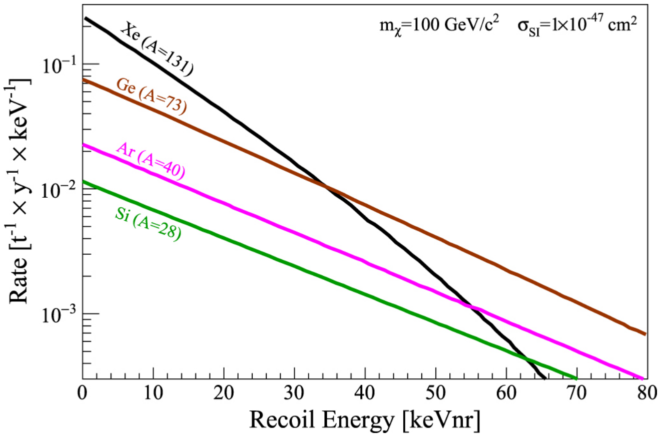


Figure 2: [17] Nuclear recoil spectra for varying noble gas targets highlight the better interaction rate at lower nuclear recoil energies for heavier targets but a lower rate for higher recoil energies.

Finally, the last method of detection is direct detection [18] and the most common one as of today's experiments. 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 cross-sections 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 [19]

$$\frac{\mathrm{d}R}{\mathrm{d}E_{\mathrm{nr}}} = \frac{\rho_0}{m_x m_N} \int_{v_{min}}^{v_{esc}} \frac{\mathrm{d}\sigma_{xN}}{\mathrm{d}E_{\mathrm{nr}}}(v, E_{\mathrm{nr}}) v f(v) dv \tag{1}$$

often approximated to [20]

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

and shown in Figure 2 for increasing mass of the target nucleus. Here m_X is the dark matter mass, m_N is the mass of the noble element, ρ_0 is density of dark matter in the local space, v_{min} is the dark matter velocity and depends on the annual and diurnal modulation [21].

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 Monte Carlo simulators such as G4DS (Geant 4 Darkside) one has to incorporate the nuclear recoil spectrum in the simulation setup and the program will sample from this the ^{40}Ar recoils accordingly. This process is a long one since this program simulates everything from the interaction to detection and can take several days if running over many energies with all the scintillation being captured.

Similarly, our machine learning algorithm is trained uniformly 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.

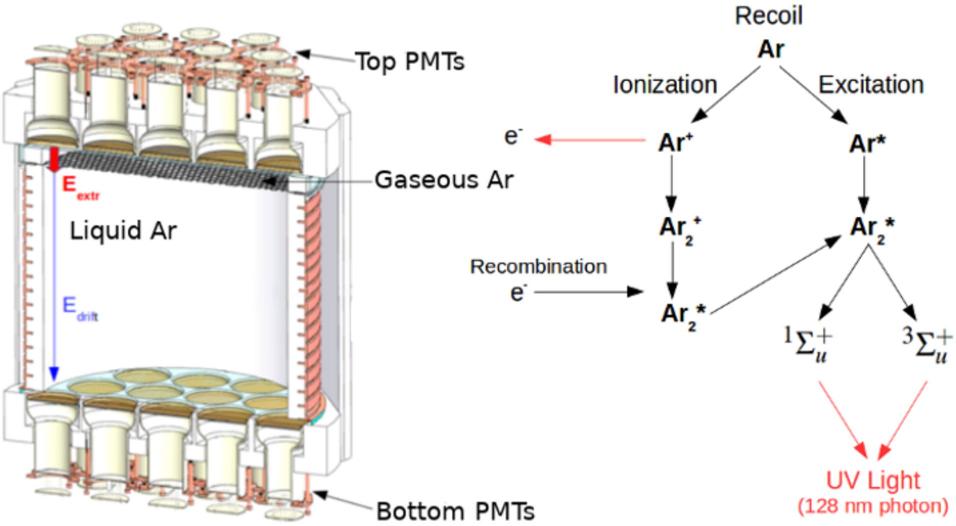


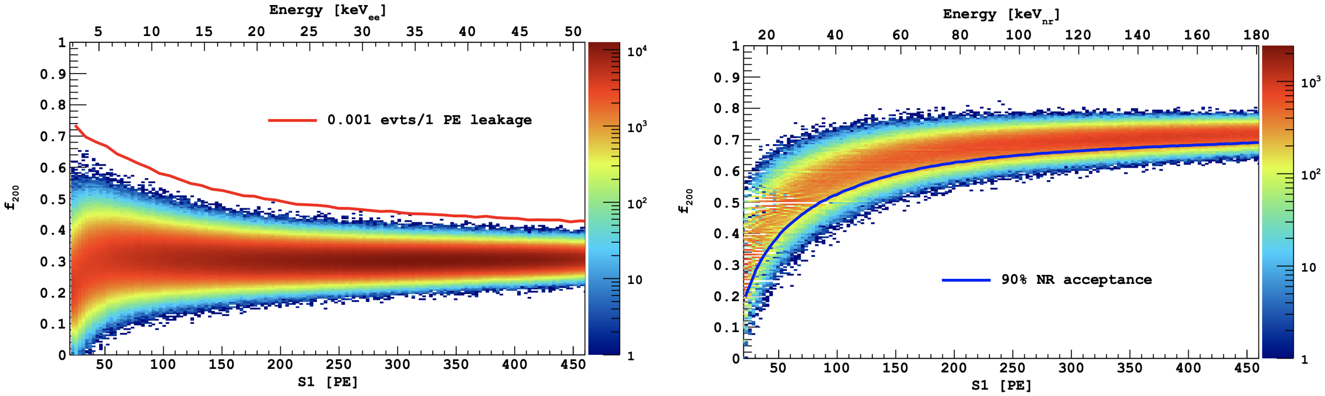
Figure 3: [22] Schematic of a LAr-TPC and processes of VUV photon emission.

Finally, to understand the variables of interest in this study consider Figure 3. A dark matter particle enters the detector fiducial volume where it interacts with an ^{40}Ar nucleus. The nucleus then becomes excited and during de-excitation emits electrons and produces scintillation. This first scintillation is known as S_1 and has a windows of 7μs. After which the electrons drift upwards due to an electric field, within a window of 376μs. The electrons reach a boundary between Argon in the liquid and gaseous phase which produces a secondary, much more intense scintillation. Subsequently these photons are detected by the Silicon photomultiplier tubes (SiPMTs) and are known as S_2; 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 recoils. 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 $^1\Sigma_{\mathrm{u}}{}^+$ and $^3\Sigma_{\mathrm{u}}{}^+$. The former has a lifetime of 7ns while the latter has a lifetime of 1600ns. This difference makes Argon a very competitive candidate as a noble element target since this same difference in lifetimes between excited states in Xenon (the other major competitor for choice of noble element) is only about 25ns. Although Xenon has other benefits and Argon has other sources of background Xenon based dual phase 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 or alpha particles and nuclear recoils with argon ion tracks. Thus a parameter is used called the Pulse Shape Discriminant denoted by f subscripted by the window of time of interest in ns. We shall use f_{200} defined as

$$f_{200} = \frac{\int_0^{200ns} \text{Intensity of photons received}}{\int_0^{7\mu s} \text{Intensity of photons received}}. \tag{3}$$

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 historically the parameter f_{90} has been used for experiments such as Darkside-50, for a much bigger experiment such as this the drift distance is increased substantially so f_{200} is more suitable.



(a) Simulations for Darkside-20k of f_{200} vs S_1 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} vs S_1 for signal nuclear recoils. Blue curve is the 90% NR acceptance region.

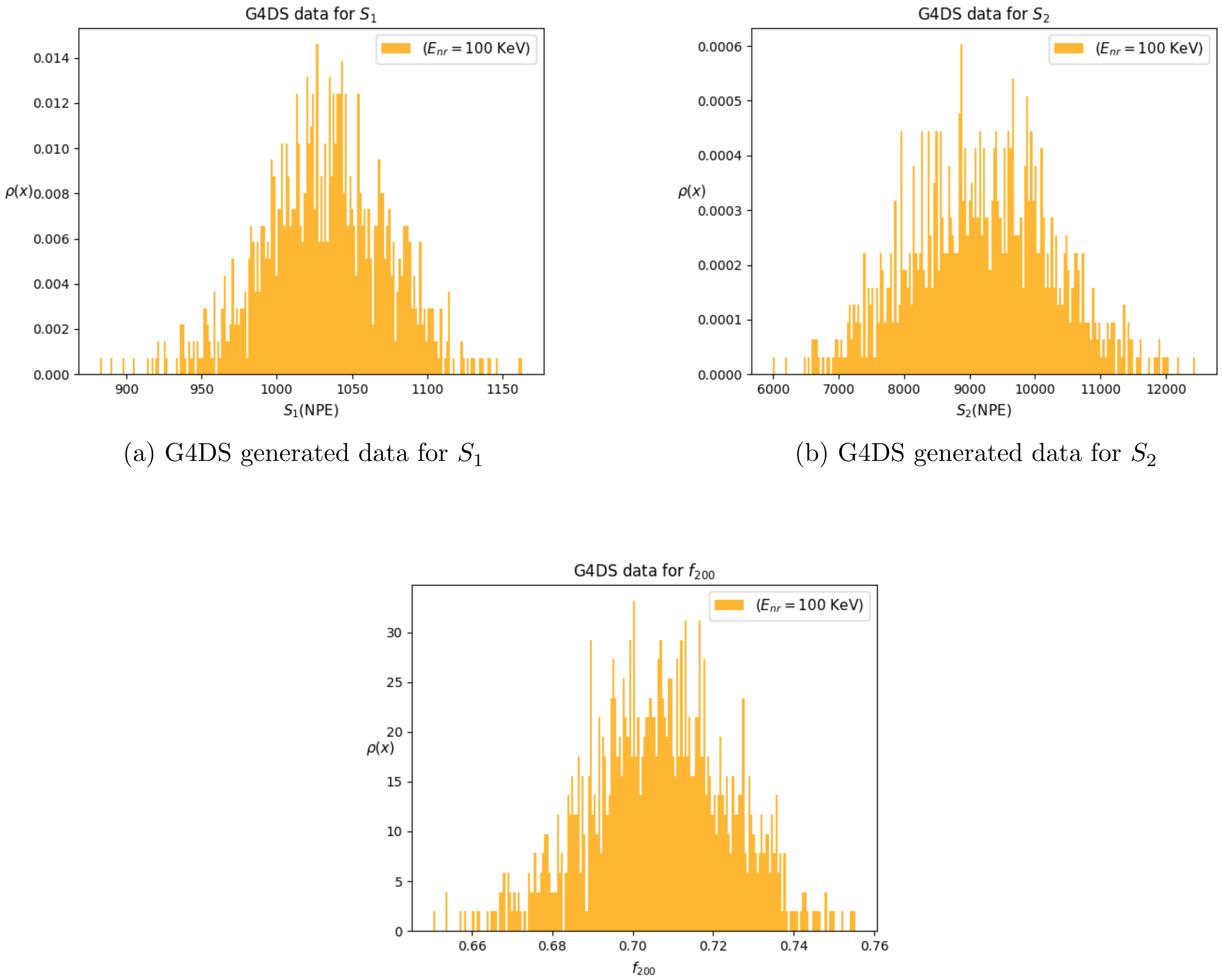
Figure 4: [23] Region of interest using f_{200} pulse shape discriminant against total intensity S_1.

1.2 Current G4DS Results

The current simulation methods used by the Darkside collaboration consist of very sophisticated and proven Monte Carlo methods. These have been programmed in an open-source software package called Geant4 [24] 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 selection 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 events 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 correlation between the three variables has been understood to an acceptable extent.

1.3 Deep learning as an alternative

As discussed before, the results obtained with G4DS are well understood and accepted as correct simulations (backed up by experiments DS50 [25]). 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 that 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 and computational resources needed. 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 scale 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. The following is not an introduction to machine learning techniques, rather for that please refer to my first semester report [1]. A neural network is a set of nodes in different, successively connected 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 optimizer 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 better final weights which will minimize 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 [26] [27]. 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 with, 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 [26]

$$C(G) = \mathbb{E}_{x\sim p_{data}}[\log(D^*_G(\bar{x})] + \mathbb{E}_{x\sim p_g}[\log(1 - D^*_G(\bar{x})] \tag{4}$$

where the optimal discriminator is described by

$$D^*_G(\bar{x}) = \frac{p_{data}(\bar{x})}{p_{data}(\bar{x}) + p_g(\bar{x})}. \tag{5}$$

For the ideal case where the generator is perfect the accuracy of the discriminator is expected to be around 0.5 as it would have no real clue other than a 50-50% chance to tell the difference between real and generated data.

2 Methods

2.1 Problems with previous approach

In the first semester, a qualitative approach was taken to this problem of trying to reproduce G4DS data 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 three main problems with that approach:

i) There was barely any quantitative analysis of the reproducability of training data by the GAN. 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 if such an algorithm would even be possible.

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 created 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. At the end of last semester we identified wGANs, or Wasserstein GANs [28], to be the next architecture we would try. 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 amount of time as the first semester, trying to get essentially to 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/vanilla GAN but this was not the aim of what were doing. We thus decided to focus our efforts on other architectures. Lastly, we are thus 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 even better convergence.

2.2.2 ARGANs

Even though I admit it is hard to believe, the name has nothing to do with the fact we are using this in a liquid Argon TPC. In fact, AutoRegressive GANs [29] [30] 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 generation the method still applies to 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 [31]. The closest to what we have done was done in 2011 [32] 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

$$p(\bar{x}) = \prod_{i=1}^{n} p(x_i|x_0, ..., x_{i-1}) \tag{6}$$

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, $P(S_1|E)$ was taught for s epochs (usually 10,000 in this report) and the best output of that generator was 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 number of variables being taught) and by the end of the process we 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 new variable is wildly different than the three already taught chances are that a very similar neural network can be used.

2.2.3 Moments as a measure of performance

One of the conclusions from last semester was the need for more detailed metrics to compare generator and discriminator performance while training. We used to monitor performance using the discriminator accuracy/loss vs epoch curve. Ideally the accuracy would converge to about 0.5 and loss to about 0.7 however beyond checking this convergence those plots do not convey much information. In fact, because of an effect GANs are known to enter known as a failure mode (where the generator and the discriminator reach a Nash equilibrium [27]) the accuracy/loss plot would show convergence towards expected values but the generated data would be nowhere close to the training data. One can compare this plot in Figure 6 with Figure 7 explained hereunder. While comparing these plots bear in mind that this run ended up not converging for f_{200} and it is fair to assert that Figure 7 does a much better job at showing this than Figure 6.

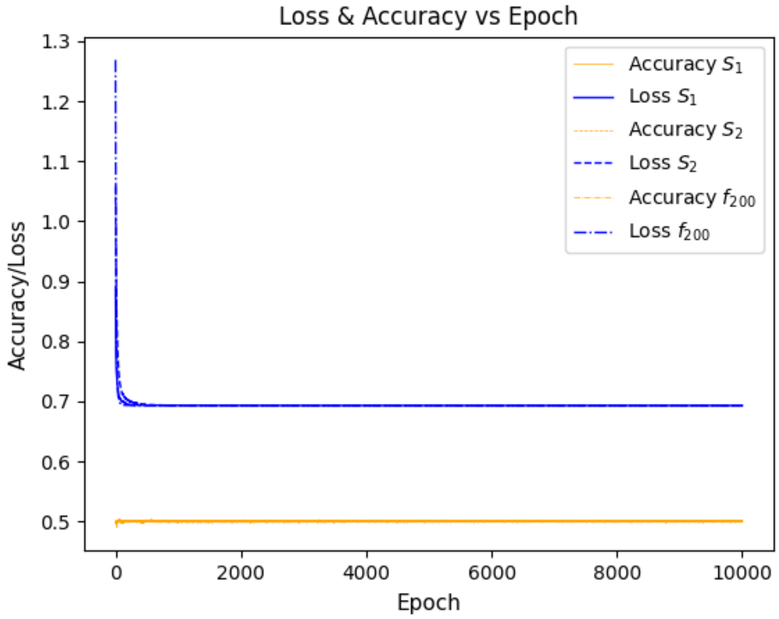
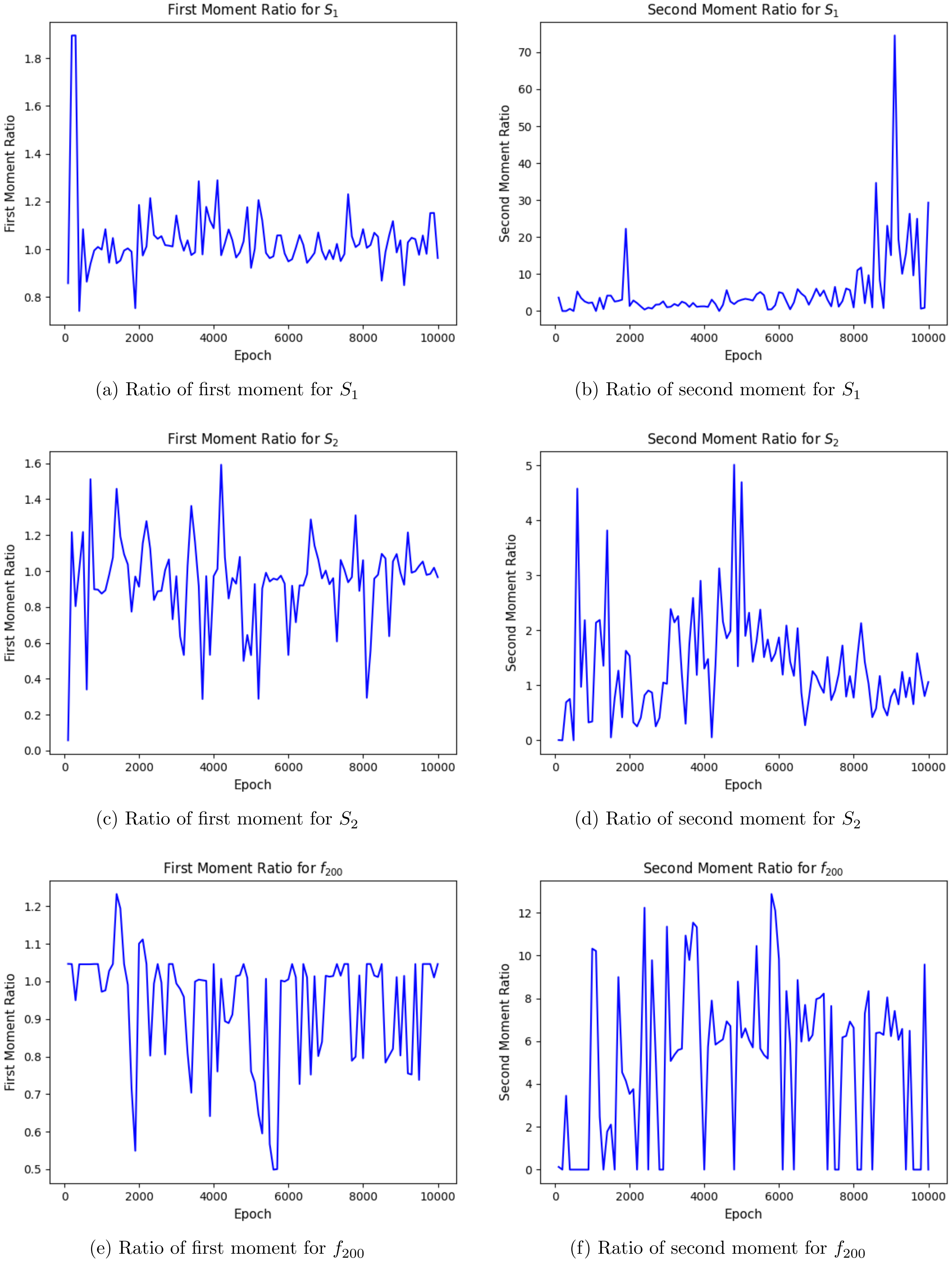


Figure 6: Previously used in semester 1, these accuracy/loss curves clearly do not convey enough information about convergence and performance of the GAN during training.

A new metric would not only allow us to better compare how different GAN configuration affect performance but also in real-time give us the opportunity to gauge if the current configuration being used is likely to converge. Of course we are not the first people to have to compare two distributions together. We needed something fast during training to not hinder the already cumbersome process so, although metrics such as Wasserstein distance are ideal, we carry out these metrics after training in the analysis stage. We also needed something that would not necessarily give us granular detail of performance but more of a global gauge of how the GAN is performing across the whole energy domain, per variable.

Thus we settled on selecting three energies from the whole 230 energy samples being trained on simultaneously. These three energies were specifically selected to be the first, middle and last in the domain. Then, the array of length 3 arrays of length 1000 data points each is flattened and the first and second moments are calculated. The ratio of $\frac{\mathbb{E}(\text{GAN4DS Output})}{\mathbb{E}(\text{G4DS Output})}$ and $\frac{\text{Var(GAN4DS Output)}}{\text{Var(G4DS Output)}}$ is obtained. Figure 7 shows these ratios plotted at every epoch check (which in this run was at every 100 epochs) for a particular run which ended up not converging for f_{200}. One thing to note is that we are suggesting these plots alongside the accuracy/loss curves since a convergence in the latter but non convergence in the former is the best way to observe failure modes, which is an added advantage to having these new plots.

Finally, at every epoch check after logging these ratios we also check if this generator is the best one yet. The way we do this is to calculate Taxicab metric on each energy histogram between GAN4DS and G4DS and sum the distances for all the energies in the domain. Then we check if this is less than the previous sum of distances and if it is we save this particular run. This allows us to save the generator that most closely resembles the training dataset, using this metric. At the same time Tensorflow is trying to minimize the value of the loss. As a potential extension and given the promising results the ratio plots have shown us, we suggest using this metric in the future by including it directly into the Tensorflow network layout as a metric alongside loss minimization.



(a) Ratio of first moment for S_1

(b) Ratio of second moment for S_1

(c) Ratio of first moment for S_2

(d) Ratio of second moment for S_2

(e) Ratio of first moment for f_{200}

(f) Ratio of second moment for f_{200}

Figure 7: Use of ratio of GAN4DS/G4DS moments during training drastically improves the capture of the actual performance of the GAN compared to accuracy-loss more commonly used in machine learning.

2.2.4 Work Pipeline

Before presenting final results, I thought it might be worthwhile briefly mentioning changes to our coding style and the way we completely redefined 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 have. We found however that we easily ended up with tens of notebooks, each with a different architecture and needing to manually save outputs. Consistency 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's Physics Department GPU cluster. The package we produced aims to really focus on being able to be used in the 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, consistent format. Tensorboard is a package provided by Google, the creators of Tensorflow, which allows monitoring machine learning progress to the level where one can see the weights of each individual layer to monitor 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 damaging performance and convergence. We made this codebase in the hope it could serve as a baseline for future teams potentially working on continuing our work and we encourage to expand its capabilities.

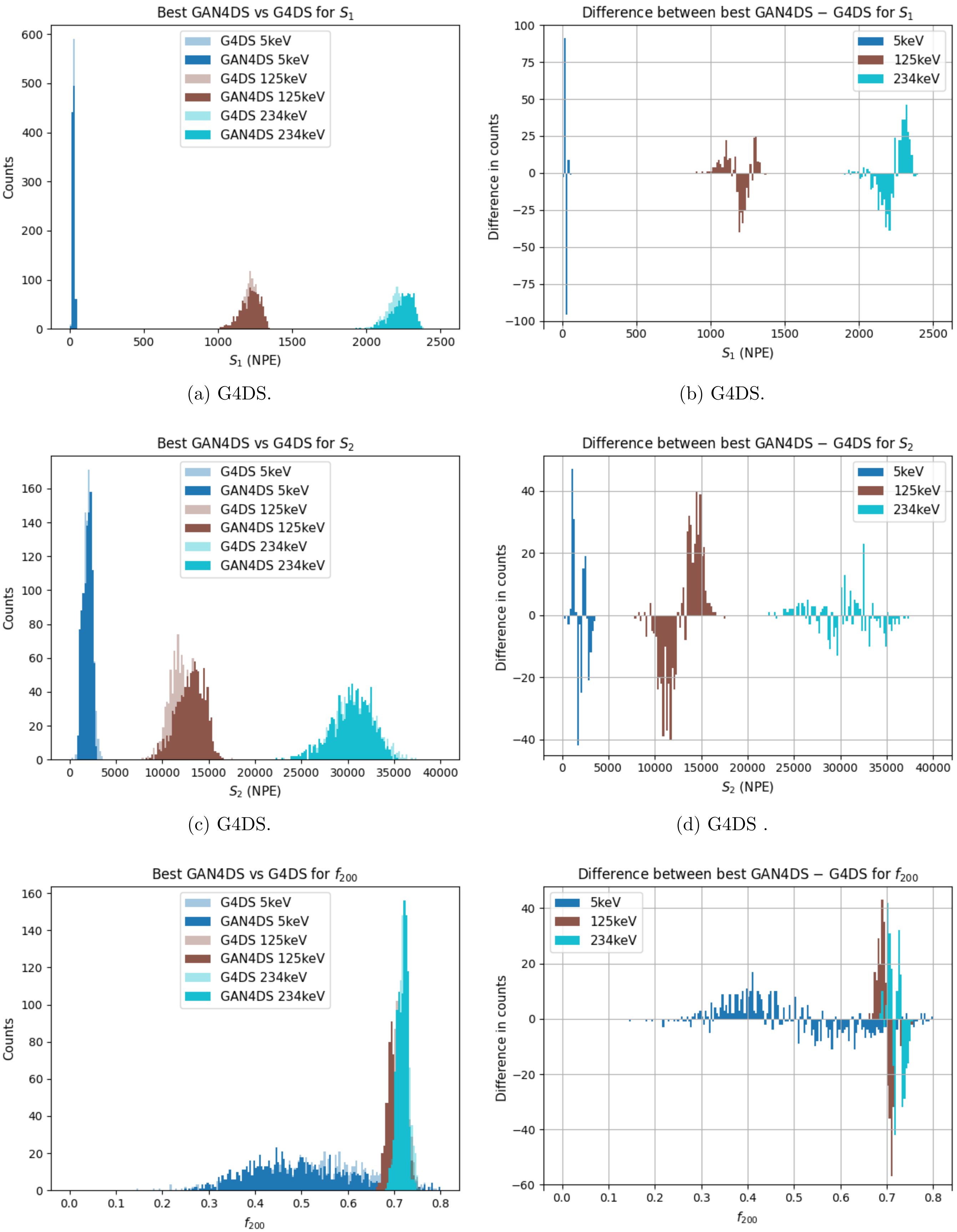
3 Results

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. The order of variables learnt was S_1 then S_2 then f_{200}. All results presented are from the same run with the above configuration.

3.1 Individual Variables

The first results I present are the individual variables. Figure 8 shows (left) the results of 3 nuclear recoil energies (minimum, middle and maximum over the 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. Figure 9 shows the metrics used to monitor performance during training.

The difference plots seem to show a symmetry about the mean of each training dataset distribution. The GAN appears to have produced output which on one side under-estimates the training data and over-estimates it on the other side. The largest difference is seen in the lower energies of the f_{200} distribution. This could be attributed to the much more spread-out shape these lower energies have when compared to the larger energies. GANs are known to suffer in datasets where there is high variance across members.



(a) G4DS.

(b) G4DS.

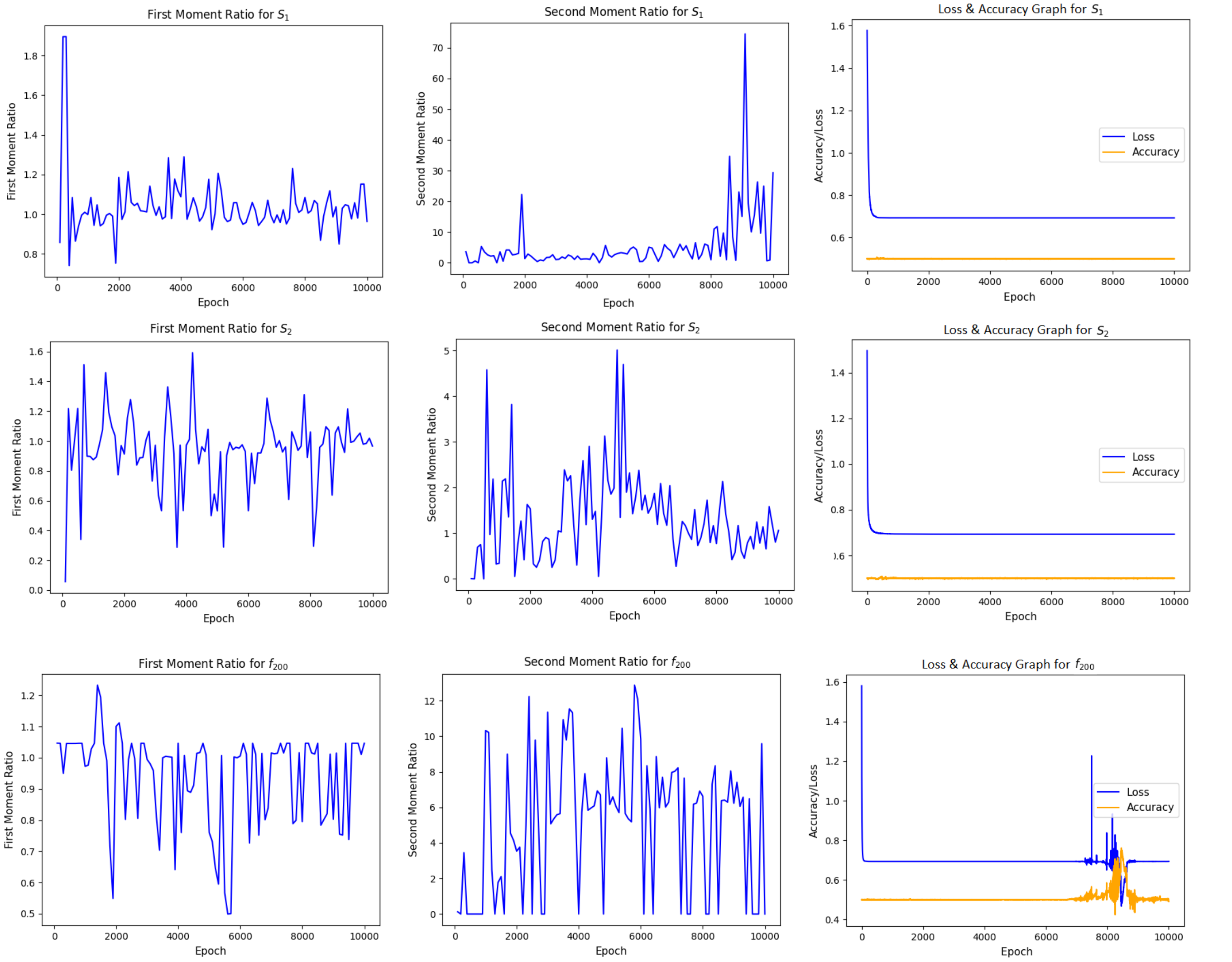
(c) G4DS.

(d) G4DS .

(e) G4DS.

(f) G4DS.

Figure 8: Individual variables learnt successively 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.



First Moment Ratio for S_1

Second Moment Ratio for S_1

Loss & Accuracy Graph for S_1

First Moment Ratio for S_2

Second Moment Ratio for S_2

Loss & Accuracy Graph for S_2

First Moment Ratio for f_{200}

Second Moment Ratio for f_{200}

Loss & Accuracy Graph for f_{200}

Figure 9: The metrics used during training to capture performance and convergence of the GAN on training data. From the accuracy/loss plot and the moments plots it appears the GAN had entered a failure mode for f_{200} but comes out of it to converge on the actual training data at around 8000 epochs.

3.2 Variable Correlations

Although individual variables are very important, what we sought out to teach the GAN were the correlations between these. Figures 11 ($\log(m) = 1.5$) and 12 ($\log(m) = 4$) show the results for correlations between the individual variables shown previously. To produce these correlation plots first a mass for the WIMP is chosen, in this case we take $\log(m) = 1.5$ and $\log(m) = 4$ in a spin-independent model which corresponds to a 31.5 GeV/c^2 mass. From theory the corresponding recoil energy spectrum is chosen, shown by Figure 10 for $\log(m)$ =1.5. After sampling 1000 energies these are used to select from runs of 1000 ^{40}Ar recoils in G4DS and GAN4DS.

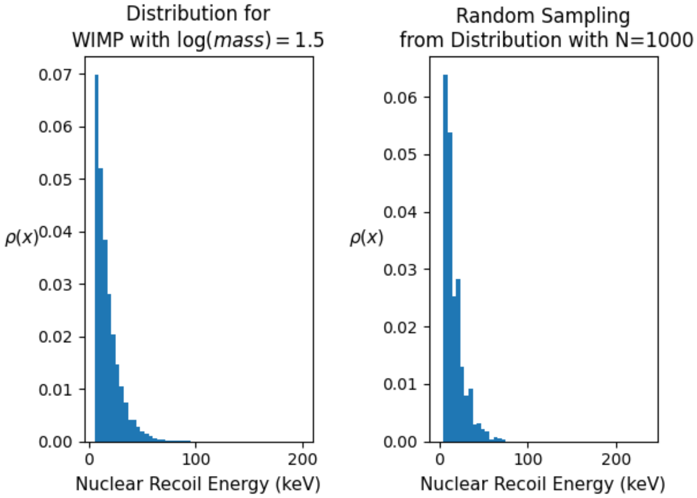
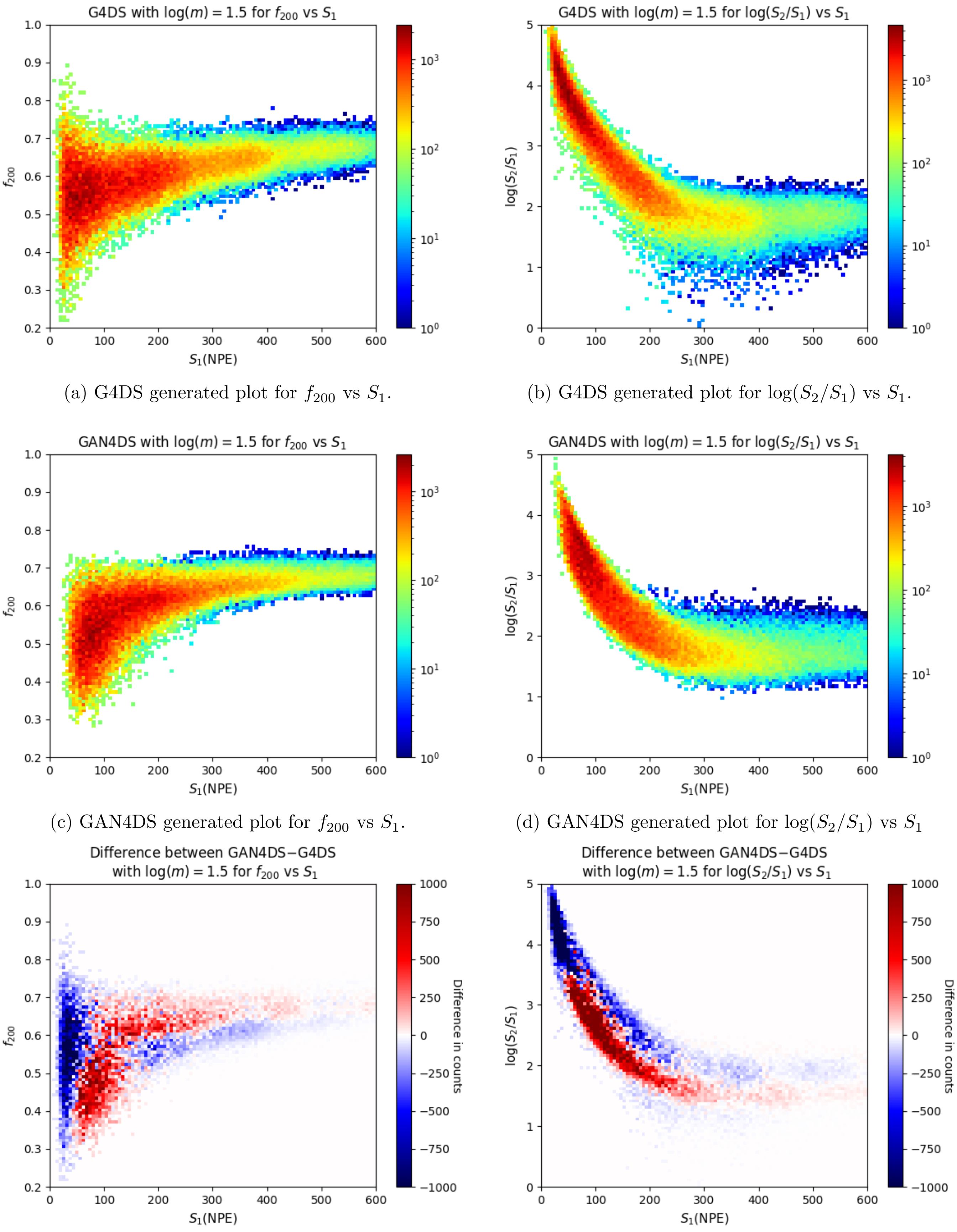


Figure 10: A plot of the theoretical recoil energy spectrum for a WIMP of mass $\log(m) = 1.5$ vs the sampled N=1000 energies spectrum. The same is done for $\log(m) = 4$.

Visually the GAN4DS resembles G4DS very closely both in shape and density. The band in the f_{200} vs S_1 plot in the region of higher energies around the value of $f_{200} \approx 0.7$ is an indication that the correlation was learnt as this is where signal is expected to lie. In both 11c and 11d we see GAN4DS fails to reproduce the low counts of the outlying points with high variance. This could be attributed as before to one of the known failures of GANs to reproduce high variance points in training datasets. One further contribution was the fact that before training, all of the training distributions were normalised by taking the maximum value for that energy for that variable and dividing by it throughout. This allowed the GAN to learn how to draw these distributions which were all similar in shape without being tied to specific magnitudes of mean. We found that without this normalization before training the GAN mostly fails to learn all the variables. So although this method we used is very helpful in learning, it might 'squash' the data into the shapes we see and prevent certain points which have higher variance to be learnt.

Finally, the difference plots are very important since they tell us is visually hard to tell. Again as for the individual variables we notice a kind of symmetry. This time it is difficult to say why it is symmetric about a certain curve but we hypothesise this is due to the behaviour also seen in the individual variables. That is, during training it was observed that the GAN when trying to learn across all the distributions along the energy domain for a single variable, will try to shift all the distributions together by a fixed amount rather than individually moving distributions or transforming them. This was very interesting to see since it could explain a general shift of all the energy distributions as a whole. Further studies of this might indicate an offset that one could apply to the generated dataset after training or a way to penalise the GAN during training when it does this global shifting. One thing to note is that the difference seems to be less for the higher mass plots and since these have a higher sampling of higher recoil energy events (conversely a lower sampling of lower energies) it could mean the problem lies in learning the lower energy distributions for each variable. This is seen later in the analysis subsection.



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

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

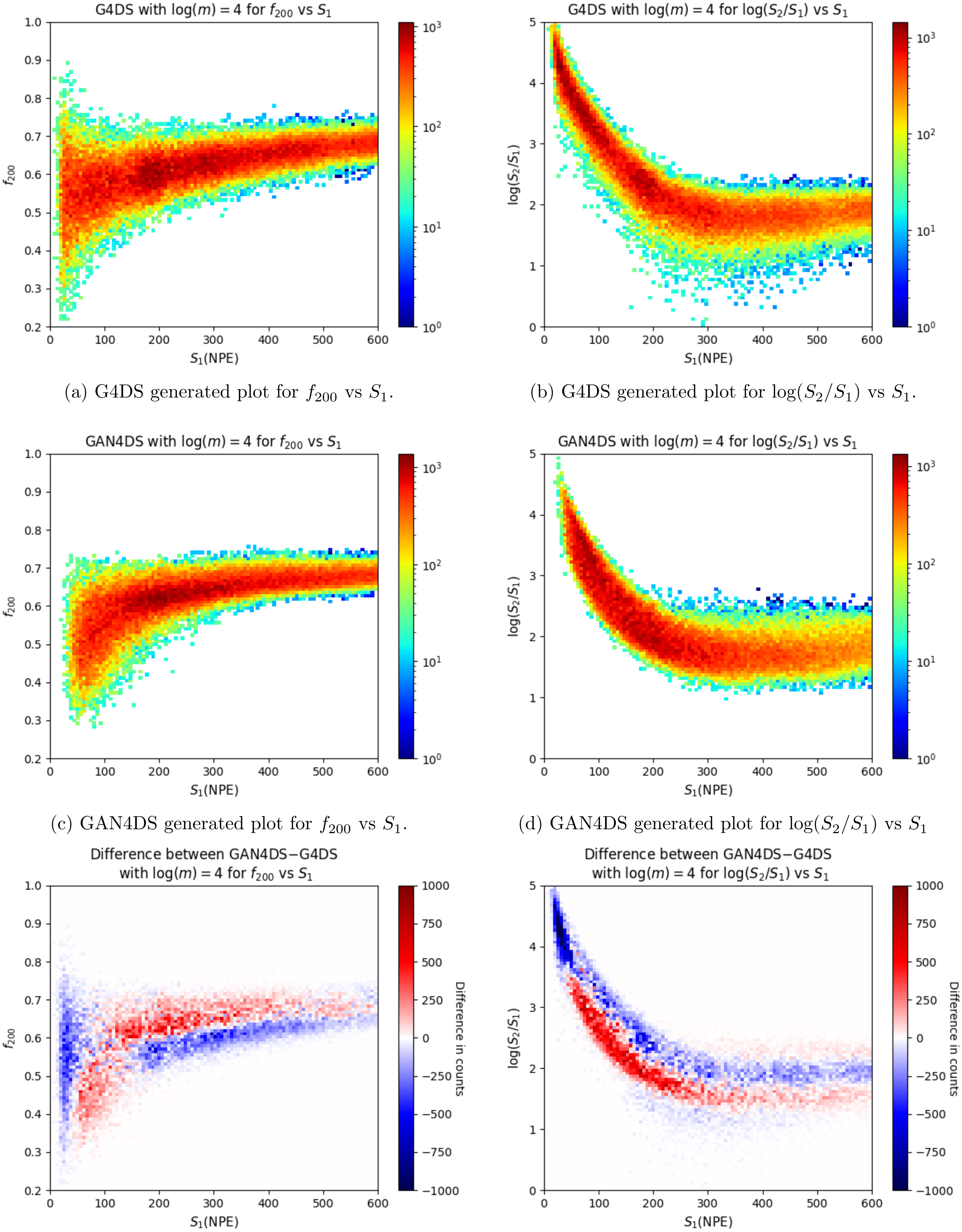
(c) GAN4DS generated plot for f_{200} vs S_1.

(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 11: 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.

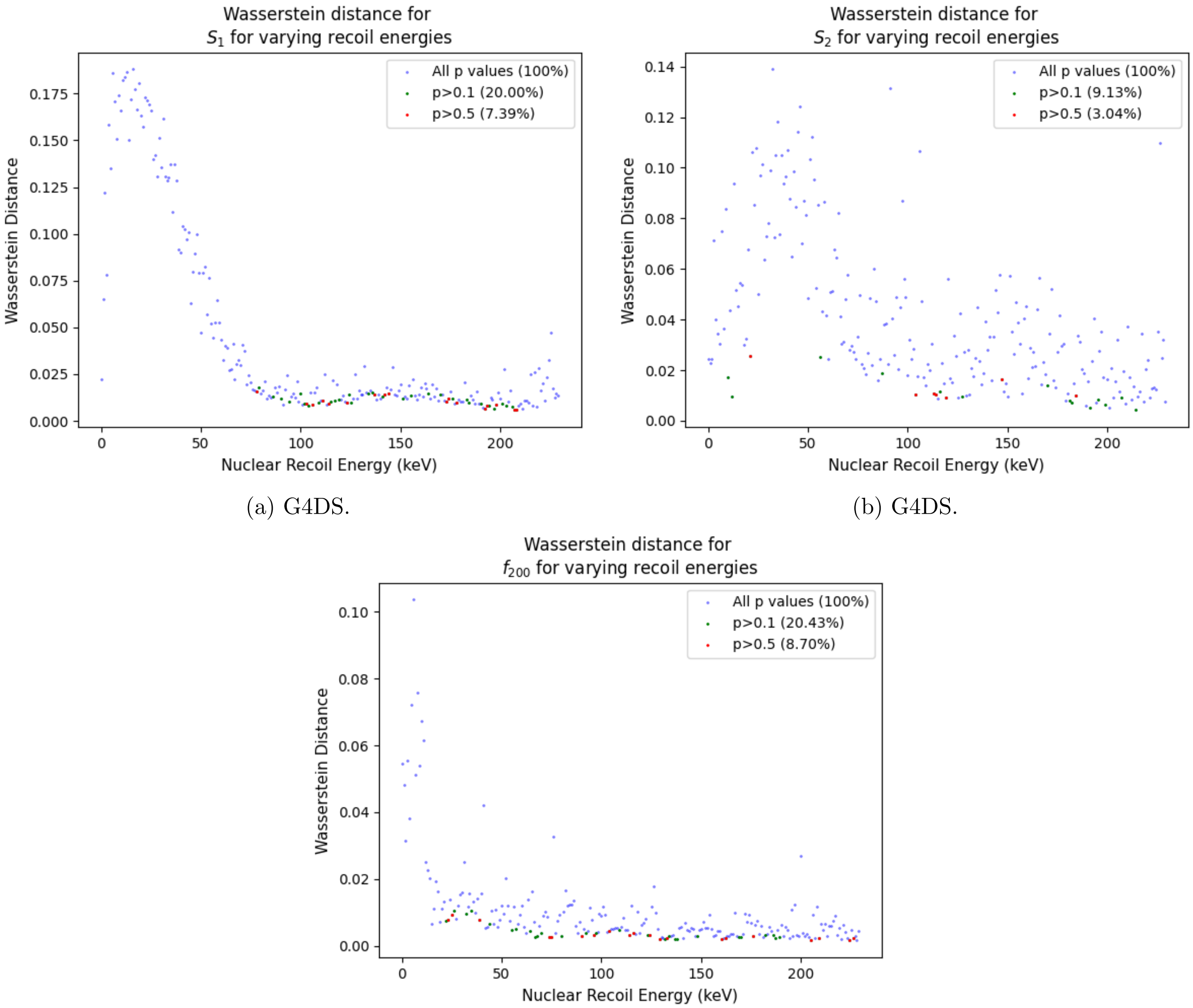
(d) GAN4DS generated plot for $\log(S_2/S_1)$ vs S_1

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

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

Figure 12: 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 13: Wasserstein metric calculated on the GAN4DS and G4DS generated data for each variable for each energy. P values are obtained by using a permutation test on geometric mean. In round brackets is the percentage of points that pass the significance threshold.

Figure 13 shows the results of applying the Wasserstein metric [33] on each variable for each nuclear recoil energy between GAN4DS and G4DS. The Wasserstein metric is often used to compare two distributions together and works on the amount of work/area needed to move around one distribution to make it equal to the other. The colours on the points of these plots are due to another test we carried out. Known as a permutation test [34] [35], we calculate the geometric mean of the each two equal energy, equal variable distributions and take the absolute difference. We then mix all the data points from the two and take split them into two groups of random permutations of all the data points. We then carry out again the difference in geometric mean of the two groups and check if this is at least as extreme as the original geometric mean difference. This is carried out for 1000 permutations and finally we end up with a probability that if we were to observe another value (have another generated distribution by GAN4DS) this would have a geometric mean difference with the real data at least as extreme. Given the Null hypothesis being true, in our case that the two distributions are the same distribution, this value of p that the permutation test returns will be the confidence we have that we cannot reject the Null hypothesis. We can never accept the Null hypothesis with this test but only reject it or fail to reject it.

The first observation out of these plots is that the GAN reproduces training data much better at higher recoil energies than lower ones. This is seen both in the lower Wasserstein distance and the higher amount of points that pass the permutation test's p significance value. The even more interesting observation is that the relative distance between variables in the lower energy region decreases from S_1 to f_{200}. Recall these variables were learnt successively and this behaviour seems to show a self-correcting learning behaviour across variables, not only across epochs. The pronounced spike in Wasserstein distance between G4DS and GAN4DS for lower recoil energies in S_1, seems to disperse for S_2 and for f_{200} again becomes smaller in magnitude and shape. Essentially, if the GAN does not quite learn how to make this Gaussian-like shaped data for S_1, the mistake is not repeated as a whole for S_2 even though the neural network layout used for these two is the same. Rather, it is appears the more variables trained on the more we can actually see the GAN getting better at producing these Gaussian-like shapes. There is one caveat in that Wasserstein distances, as mentioned before, rely on areas so across different variables might not be exactly comparable however we do normalise each distribution before calculating the Wasserstein distance. This is done by dividing both GAN4DS and G4DS variable distributions by the same normalisation value as described previously in Section 3.2. Using this normalization makes the three plots in Figure 13 more comparable.

One final thought is that these results suggest one could benefit in splitting the data around the 75 keV mark into two groups and actually have a GAN for lower energies and one for higher ones since they are more comparable to each other. This result also made us question what would happen if one were to insert S_1 again after f_{200} has been learnt. If the Gaussian-like distribution generation has been perfected with each variable, this would make S_1 even better than it was originally the first time it was learnt. This could potentially be extended to a completely different way of learning. Instead of learning all the energies for one variable, one could shuffle the variables while keeping the number of energies the same and train over 10,000 epochs. We suggest this could be one way of doing this, alternatively going through cycles of S_1, S_2 and f_{200} until the distributions are learnt for all three variables. The only way to know whether these approaches or any other ones would work would be to study them so we hope in future iterations of this project this could be done.

4 Final Remarks

Last semester was focused on getting a proof-of-concept GAN working to learn S_1, S_2 and f_{200} and we managed to show this could be done but not efficiently. What we have done this semester is make these algorithms scalable and stable and analysed not only qualitatively but quantitively performance and accuracy. We have tried multiple architectures and finally have settled on the very new ARGANs however this is by no means the only way to do it nor the best and we definitely encourage new architectures to be tried since this field of research is currently extremely active. More work has to be carried out in the homogenous learning across all the nuclear recoil energy domain, especially for lower energies. We suggest doing this by cycling through the variables being taught multiple times in a successive order since we hypothesise this could really help the GAN converge over the whole energy domain.

Finally future extensions to this project could include noise simulations and background simulations and finally all these events put together in one GAN. An even larger extension would be to generate the individual photon responses by the sensors and PMTs which could then in turn produce the variables studied in this report. This could potentially be used at the same time to produce directionality predictions by making use of a classifier connected to the output of the GAN. This would be a massive help to the current detector setup since as of today the detector does not produce such kind of information. To conclude, we believe this project has produced tantalising results and hope this work will be continued by future iterations of the project.

References

[1] E. Zammit Lonardelli and K. Jethwa, "First semester mphys report," The University of Manchester, Tech. Rep., 2020.

[2] 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`.

[3] G. Bertone and D. Hooper, "History of dark matter," *Reviews of Modern Physics*, vol. 90, no. 4, p. 045 002, 2018.

[4] G. Steigman and M. S. Turner, "Cosmological constraints on the properties of weakly interacting massive particles," Delaware Univ., Tech. Rep., 1984.

[5] R. J. Scherrer and M. S. Turner, "On the relic, cosmic abundance of stable, weakly interacting massive particles," *Physical Review D*, vol. 33, no. 6, p. 1585, 1986.

[6] S. Weinberg, "A new light boson?" *Physical Review Letters*, vol. 40, no. 4, p. 223, 1978.

[7] A. Kusenko, "Sterile neutrinos: The dark side of the light fermions," *Physics Reports*, vol. 481, no. 1-2, pp. 1–28, 2009.

[8] P. Agnes *et al.*, "Direct search for dark matter with DarkSide," *Journal of Physics: Conference Series*, vol. 650, p. 012 006, Nov. 2015. DOI: `10.1088/1742-6596/650/1/012006`. [Online]. Available: `https://doi.org/10.1088%2F1742-6596%2F650%2F1%2F012006`.

[9] D. N. M. and, "The LZ dark matter experiment," *Journal of Physics: Conference Series*, vol. 718, p. 042 039, May 2016. DOI: `10.1088/1742-6596/718/4/042039`. [Online]. Available: `https://doi.org/10.1088%2F1742-6596%2F718%2F4%2F042039`.

[10] D. M. Wittman, J. A. Tyson, D. Kirkman, I. Dell'Antonio, and G. Bernstein, "Detection of weak gravitational lensing distortions of distant galaxies by cosmic dark matter at large scales," *Nature*, vol. 405, no. 6783, pp. 143–148, 2000.

[11] J. H. Oort *et al.*, "The force exerted by the stellar system in the direction perpendicular to the galactic plane and some related problems," *Bulletin of the Astronomical Institutes of the Netherlands*, vol. 6, p. 249, 1932.

[12] L. Roszkowski, "Particle dark matter - a theorist's perspective," *Pramana*, vol. 62, no. 2, pp. 389–401, 2004.

[13] Aad *et al.*, "Search for dark matter candidates and large extra dimensions in events with a jet and missing transverse momentum with the atlas detector," *Journal of High Energy Physics*, vol. 2013, no. 4, p. 75, 2013.

[14] Chatrchyan *et al.*, "Search for dark matter and large extra dimensions in monojet events in pp collisions at sqrt 7 tev," *Journal of High Energy Physics*, vol. 2012, no. 9, p. 94, 2012.

[15] J. M. Gaskins, "A review of indirect searches for particle dark matter," *Contemporary Physics*, vol. 57, no. 4, pp. 496–525, 2016.

[16] D. Harvey, R. Massey, T. Kitching, A. Taylor, and E. Tittley, "The nongravitational interactions of dark matter in colliding galaxy clusters," *Science*, vol. 347, no. 6229, pp. 1462–1465, 2015.

[17] 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`.

[18] R. J. Gaitskell, "Direct detection of dark matter," *Annu. Rev. Nucl. Part. Sci.*, vol. 54, pp. 315–359, 2004.

[19] M. W. Goodman and E. Witten, “Detectability of certain dark-matter candidates,” *Physical Review D*, vol. 31, no. 12, p. 3059, 1985.

[20] 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.

[21] A. K. Drukier, K. Freese, and D. N. Spergel, “Detecting cold dark-matter candidates,” *Physical Review D*, vol. 33, no. 12, p. 3495, 1986.

[22] 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.

[23] 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.

[24] K. Amako, S. Guatelli, V. Ivanchencko, M. Maire, B. Mascialino, K. Murakami, L. Pandola, S. Parlati, M. Pia, M. Piergentili, *et al.*, “Geant4 and its validation,” *Nuclear Physics B-Proceedings Supplements*, vol. 150, pp. 44–49, 2006.

[25] P. Agnes, T. Alexander, A. Alton, K. Arisaka, H. Back, B. Baldin, K. Biery, G. Bonfini, M. Bossa, A. Brigatti, *et al.*, “First results from the darkside-50 dark matter experiment at laboratori nazionali del gran sasso,” *Physics Letters B*, vol. 743, pp. 456–466, 2015.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

[27] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” in *Advances in neural information processing systems*, 2016, pp. 2234–2242.

[28] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein gan,” *arXiv preprint arXiv:1701.07875*, 2017.

[29] Y. Yazici, K.-H. Yap, and S. Winkler, “Autoregressive generative adversarial networks,” 2018.

[30] Z. Zhao, Q. Sun, H. Yang, H. Qiao, Z. Wang, and D. O. Wu, “Compression artifacts reduction by improved generative adversarial networks,” *EURASIP Journal on Image and Video Processing*, vol. 2019, no. 1, pp. 1–7, 2019.

[31] R. Fu, J. Chen, S. Zeng, Y. Zhuang, and A. Sudjianto, “Time series simulation by conditional generative adversarial net,” *arXiv preprint arXiv:1904.11419*, 2019.

[32] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, “Pixel recurrent neural networks,” *arXiv preprint arXiv:1601.06759*, 2016.

[33] A. Ramdas, N. G. Trillos, and M. Cuturi, “On wasserstein two-sample testing and related families of nonparametric tests,” *Entropy*, vol. 19, no. 2, p. 47, 2017.

[34] E. J. Pitman, “Significance tests which may be applied to samples from any populations,” *Supplement to the Journal of the Royal Statistical Society*, vol. 4, no. 1, pp. 119–130, 1937.

[35] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. CRC press, 1994.