Graph Convolutional Neural Networks for Position Reconstruction in the XENON1T Experiment

Alejandro Oranday

Advisor: Christopher Tunnell

aeo3@rice.edu

March 29, 2021

Contents

Introduction **2**

Background **2**

Dark Matter . . . 2

XENON1T Detector . . . 2

Machine Learning . . . 3

Neural Networks **3**

Convolutional Neural Networks . . . 3

Graph Convolutional Neural Networks . . . 4

Results **6**

During Training . . . 6

Validation Set Performance . . . 7

References **9**

Introduction

Dark matter constitutes 85% of the matter in our Universe and was made evident by observations on galaxy formation, gravitational lensing, and the cosmic microwave background [3]. However, detecting dark matter experimentally is exceedingly difficult with particle physics detectors. These particles are suspected to be weakly interacting [2] such that any detection attempt would need to be sensitive to recoils at keV levels of energy.

The XENON1T detector operated as the most sensitive dark matter detector [1], and the soon-to-be-active XENONnT detector plans to overtake that title [4]. In these detectors, there are two important elements that need to be reconstructed: energy and position. By reconstructing these key elements, we are able to accept or reject numerous observations if the reconstructed position is within the detector's fiducial volume and if the reconstructed energy is within a rejection threshold [2].

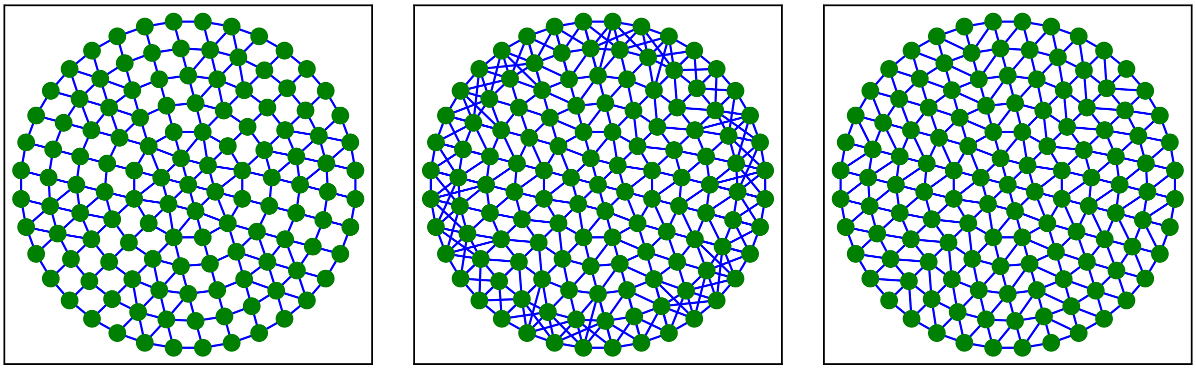
Previous machine learning implementations for position reconstruction perform well enough [6], but still run into issues with reconstruction outside the bounds of the detector as well as having an inward reconstruction bias. Finding the type of machine learning algorithm that is the most appropriate for these problems is essential for the best use of the detector's fiducial volume. This project produced the first application of graph convolutional neural networks (GCNNs) for position reconstruction in the dark-matter field, and one of the first applications of a GCNN for use in regression.

Background

Dark Matter

XENON1T Detector

The XENON1T detector is a dual phase xenon time projection chamber (TPC) located in the *Laboratori Nazionali del Gran Sasso* (LNGS) in central Italy. The detector aimed to observe weakly interacting massive particles (WIMPs) as the primary candidate for dark matter particles. It was a requirement for the detector to be sensitive to keV energy levels in order to observe these particles and was made possible through a combination of the stable xenon 136 isotope, water shielding, and depth within the Gran Sasso massif.



(a) (b) (c)

Figure 1: Three of the considered graph structures: Radius $R = 10$ cm neighbors (a), k-nearest-neighbors $k = 6$ (b), Delaunay triangulation neighbors (c). Each node here is a photomultiplier tube in the top array of the XENON1T detector. The positions of each tube in the detector was used for each of the explored graph structure approaches.

Machine Learning

Neural Networks

Recent experiments have been collecting a significant amount of data. Machine learning has become a very useful tool when processing this data. A neural network is one section of machine learning that is heavily based off the structure of animal brains. Where an animal brain is able to learn by activating specific neurons for thoughts and actions, the neurons in a machine learn in a similar vein through various kernels and learnable parameters. These sets of neurons are divided into layers and generally show up as an input layer, some amount of hidden layers, and an output layer. For our problem, the input layer will be the signal seen by the photomultiplier tubes in the top array of the detector, and the output layer will be the (x, y) position of the interaction. We are mainly concerned with graph convolutional neural networks (GCNNs) which are a result of the success with convolutional neural networks (CNNs).

Convolutional Neural Networks

A convolutional neural network is a specific kind of neural network that features a convolution layer. The input to a CNN typically comes in the form of a matrix where there is a locality between the elements in the matrix. The convolution layer makes use of a kernel that takes information from submatrices of the input matrix and summarizes the values within these submatrices. These summaries maintain the locality of the information for the

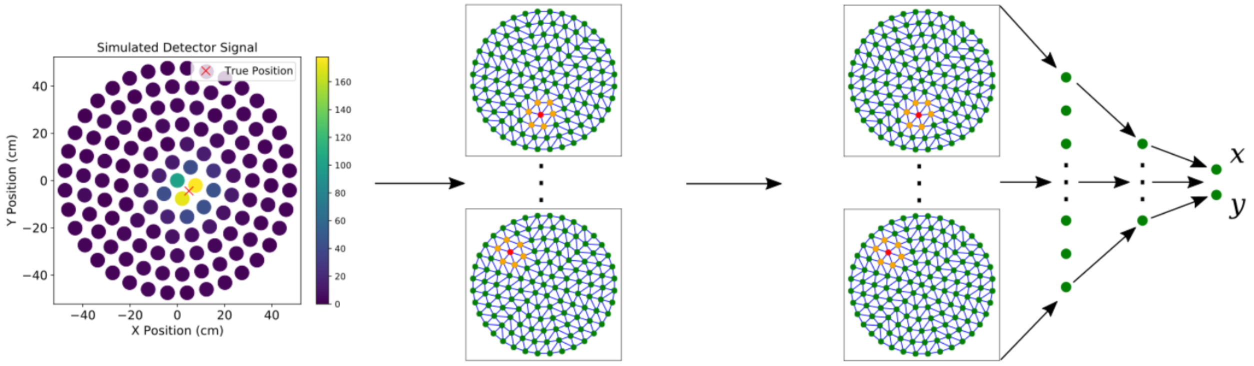


Figure 2: The graph convolutional neural network structure that the results of this paper are based on. The final structure features the signal input layer, two propagation (graph convolution) layers, two fully connected layers, and a final output layer for the (x, y) position of the interaction. Between each layer is a ReLU activation function. There are a total of 57,486 trainable parameters.
TODO: Replace the image with a sharper PDF version with less white space between the first propagation layer and the second.

respective submatrices. However, the structure of our dataset does not have a form that can be easily made into a matrix. Therefore, we would need an algorithm that is capable of making use of datasets with any possible structure.

Graph Convolutional Neural Networks

The design of the graph convolutional neural network (GCNN) came by considering how a convolution could be applied to a graph structured dataset [5]. The graph convolution layer that we use and that was proposed by Kipf and Welling propagates the values of nodes according to the edges [5]. The value of connected nodes will increase according to the values of the connected nodes while nodes that are disconnected will see no change. The exact propagation rule is given by the equation:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right)$$

where $H^{(0)}$ will be our initial input signal, $\tilde{A}$ and $\tilde{D}$ are respectively modified adjacency and modified degree matrices, $W^{(l)}$ is the trainable weights matrix for the l^{th} layer, and $\sigma(\cdot)$ is an activation function [5].

We considered the PMTs of the XENON1T detector as our nodes and their quantity of light collected as their primary value. We also included the (x, y) position of the PMT at the top of the detector as addition values. The network structure we used was based off the success of image classifiers, such as AlexNet. However, convolutional neural networks of this structure and graph convolutional neural networks in general are more often used for classification, while we have a regression problem. Our reasoning to use a GCNN for predicting the position of interactions came from being able to encode the local structure of the XENON1T detector into the dataset. By treating the nodes of our graph as the PMTs at the top of the detector, we understood that the connections or edges that we put in place would maintain the local structure if done carefully. We considered several graph structures shown in Figure 1. We ultimately chose the Delaunay Triangulated graph for it's consistent connection density throughout the graph: only PMTs that are immediately near each other are connected and resulted in most nodes having 6 edges. Only nodes that represent PMTs near the wall of the detector had degrees less than 6. There is potential for finding the graph structure that describes the detector's data best, but for this we chose to go with a heuristic approach.

Results

During Training

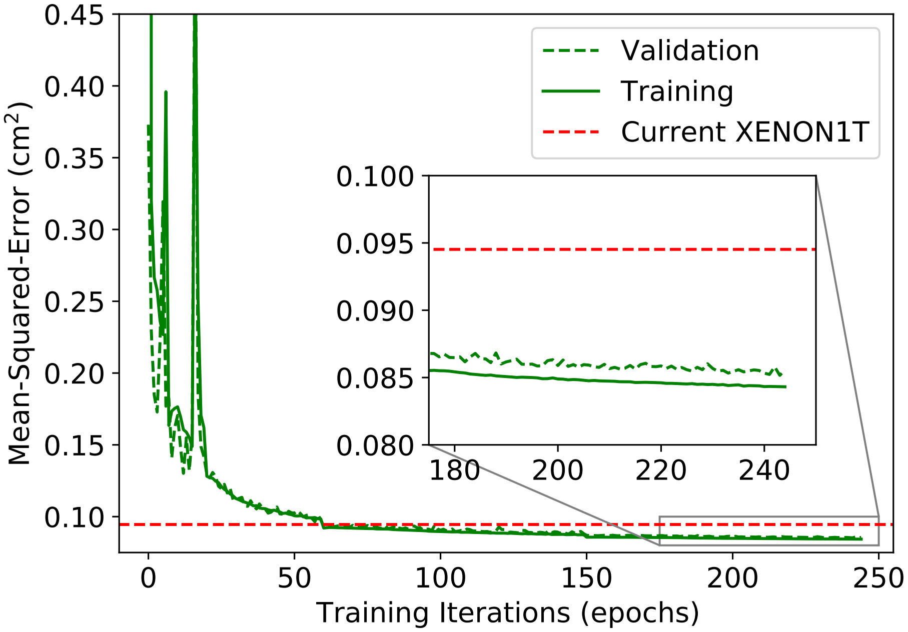


Figure 3: Mean-Squared-Error (MSE) in square centimeters of our algorithm for the training set and validation set during the training process. The minimum MSE for the current state-of-the-art is given in red at 0.0945 cm^2. This is the benchmark that our algorithm had to pass during training. Spikes within the first 20 epochs occur due to large step size during gradient descent. Learning rate was lowered at epochs 20, 60, and 150. The minimum MSE achieved by our GCNN on the validation set is 0.0852 cm^2.

The performance of our algorithm was compared to the current state-of-the-art in XENON1T during training as a benchmark and early warning system. If the GCNN did not approach a comparable performance to that of the state-of-the-art swiftly enough, training would typically stagnate and not surpass this benchmark. By not performing better here, it was generally indicative that the GCNN would also perform worse when we gave attention to our performance metrics. After a few iterations of this, we chose to only look at the performance metrics if the GCNN model produced a lower mean-squared-error during training and would restart training in cases where it was clear that the current iteration would not perform better within a reasonable number of epochs. An example of when it was clear a prospective model would not do better is if the mean-squared-error was not below 0.25 cm^2 within the first 50 epochs.

We used an optical Monte Carlo simulation of 989,875 events for training as an attempt to assume a "perfect" detector. This is to say that no spurious events, such as single electrons, dark counts, or PMT after-pulses, were within our simulation. The observations by the PMTs are as if every part of the detector ran perfectly. By using a simulation like this, we were able to input the data into our model without normalization or standardization.

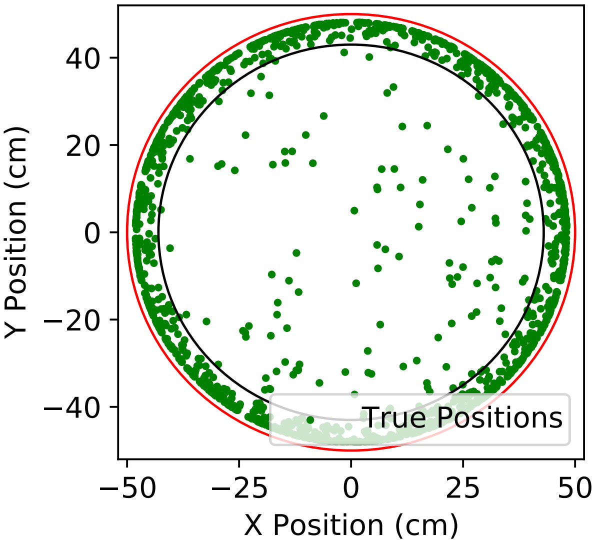


Figure 4: True positions of GCNN mis-reconstructions. 1,680 of the 197,975 simulated events were mis-reconstructed and are shown here. Red circle is the wall of the detector (50 cm); black circle is the largest radius of the fiducial volume (43 cm). 123 of the 1680 mis-reconstructions are within the fiducial volume.

Our algorithm was able to outperform the state-of-the-art in training, which is a good indicator for the overall performance. Much of the work for this stage was in optimizing the learning rate used for gradient descent. Our solution was to lower the learning rate at specific epochs based on the performance of previous results. Specifically, the learning rate was lowered at epochs 20, 60, and 150. This caused notable dips within Figure 3 and resulted in a much smoother curve after epoch 20. However, a better solution would have the learning rate lower based on the model's performance during training instead of milestones set by the attentive user.

Validation Set Performance

As previously stated, the two performance metrics we focused on are to have no reconstructions outside the detector and to minimize the number of reconstructions that are 1 cm away from the true position. For best practice in machine learning, we focused on the results of the validation set which is made of 197,975 simulated events.

Since we are hard set on having no reconstructions outside of the detector, this was the first metric we would check. As it turned out, we counted zero reconstructions outside of the detector for our latest version of the GCNN. At this point, our algorithm has successfully surpassed the state-of-the-art training benchmark and made no exceedingly erroneous reconstructions, a rule that previous implementations had difficulty passing.

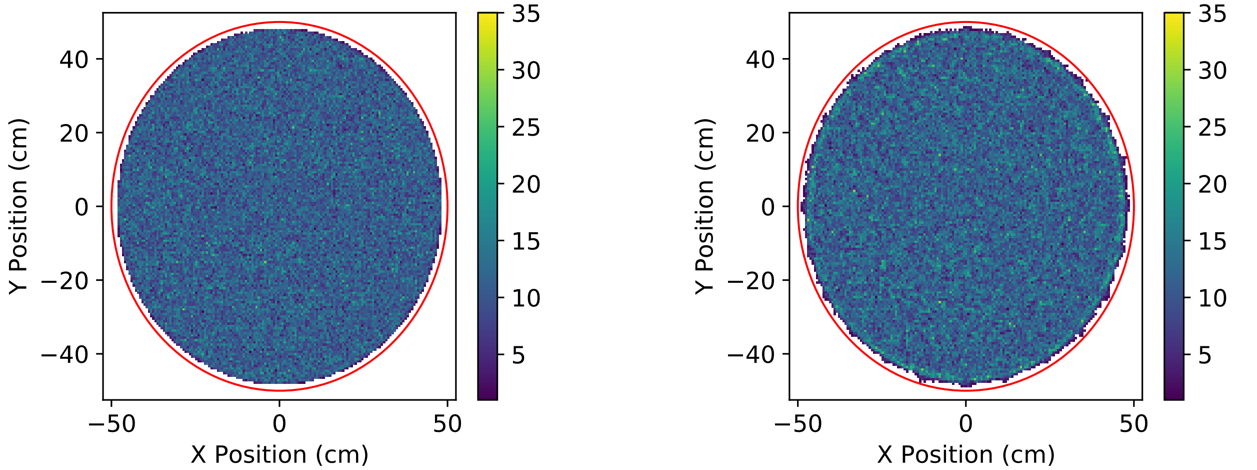


Figure 5: 2D histograms of the true positions (left) and reconstructed (right) positions at 150 bins. Red circle is the wall of the detector (50 cm). The edge of the reconstructed positions is noticeably more jagged.

As for the further than 1 cm reconstructions, these too performed well. Of the 197,975 events, 1,680 were reconstructed at greater than 1 cm away from the true position, about 0.85% of the validation set. As can be seen in Figure 4, many of the mistakes are made along the walls of the detector and explains the jagged edge found in Figure 5. If we reduce the area we count on to the maximum radius of the fiducial volume ($R = 43$ cm), we find only 123 of the 197,975 events mis-reconstructed, 0.06% of the validation set.

The last important performance check, as for any experiment, is to produce the most accurate measurements or reconstructions, in our case. For this we use the resolution metrics ΔX, ΔY, and ΔR:

$$\Delta X \equiv X_{\text{Reconstructed}} - X_{\text{Simulated}}, \quad \Delta Y \equiv Y_{\text{Reconstructed}} - Y_{\text{Simulated}},$$
$$\Delta R \equiv \sqrt{\Delta X^2 + \Delta Y^2}$$

where X and Y are the x and y positions of the reconstructions and the associated simulation. The means and standard deviations produced by our algorithm are shown in Figure 6. This too outperformed the state-of-the-art which had standard deviations greater than 3 cm. From previous observations of the results of our GCNN, the mean and standard deviation of ΔR follows suit with what we expect: most of the reconstructions are within 1 cm of the true, simulated position. At the same time, the approximately-Gaussian curves of ΔX and ΔY further confirms the positive performance of our GCNN.

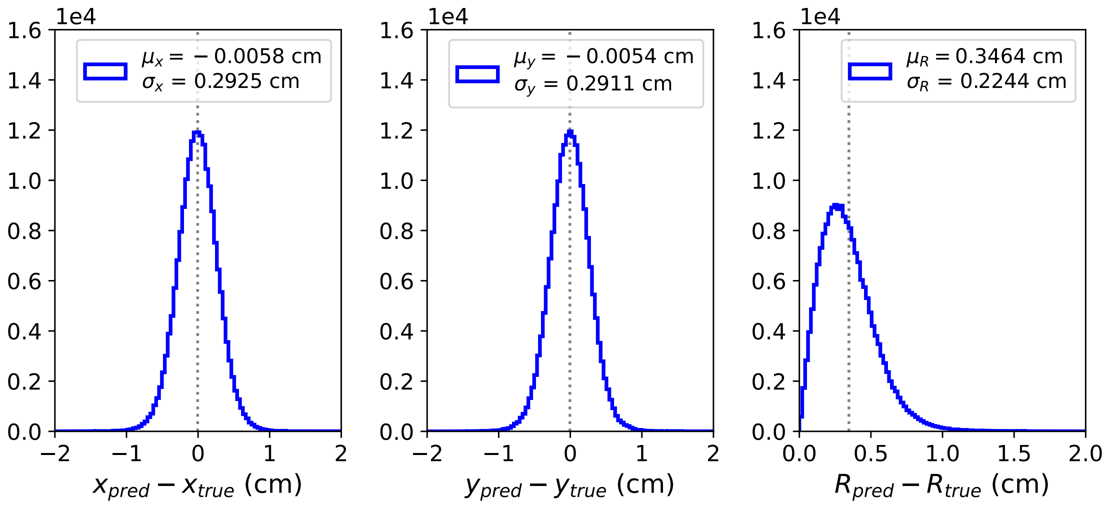


Figure 6: 1D histograms of the reconstructed position minus the true positions. There are 100 bins between -2 cm and 2 cm on $\Delta X, \Delta Y$ and 100 bins between 0 cm and 3 cm on ΔR. Both the ΔX and ΔY histograms are near Gaussian curves of the same statistics.

References

[1] E. Aprile, J. Aalbers, F. Agostini, et al. First dark matter search results from the xenon1t experiment. *Physical Review Letters*, 119(18), Oct 2017. ISSN 1079-7114. doi: 10.1103/physrevlett.119.181301. URL `http://dx.doi.org/10.1103/PhysRevLett.119.181301`.

[2] E. Aprile, J. Aalbers, F. Agostini, et al. Xenon1t dark matter data analysis: Signal reconstruction, calibration, and event selection. *Physical Review D*, 100(5), Sep 2019. ISSN 2470-0029. doi: 10.1103/physrevd.100.052014. URL `http://dx.doi.org/10.1103/PhysRevD.100.052014`.

[3] Gianfranco Bertone and Dan Hooper. History of dark matter. *Reviews of Modern Physics*, 90(4), Oct 2018. ISSN 1539-0756. doi: 10.1103/revmodphys.90.045002. URL `http://dx.doi.org/10.1103/RevModPhys.90.045002`.

[4] The XENON collaboration, E. Aprile, J. Aalbers, et al. Projected wimp sensitivity of the xenonnt dark matter experiment, 2020.

[5] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL `http://arxiv.org/abs/1609.02907`.

[6] B. E. J. Pelssers. Position reconstruction and data quality in xenon. Master's thesis, University of Utrecht, July 2015. URL `https://dspace.library.uu.nl/handle/1874/322783`.