

Novel XENON1T Event Localization Using Graph Neural Networks

Alejandro Oranday

Advisor: Christopher Tunnell

2021-05-02

1 Introduction

Dark matter constitutes 85% of the matter in our Universe and is visible through observations on galaxy formation, gravitational lensing, and the cosmic microwave background [5]. Images such as the Bullet Cluster in Figure 1 show the effects of gravitational lensing and presence of dark matter by visualizing the separation of baryonic matter and high mass concentrations. Such an image is only made possible through the presence of dark matter and discussed in further detail in Sec. 2.1.

However, detecting dark matter experimentally is exceedingly difficult with particle physics detectors. They clearly interact with gravitational force, but do not interact with electromagnetic force. One candidate, the Weakly Interacting Massive Particle (WIMP), would interact through the electroweak force [5]. For us to detect WIMPs directly, one would need a detector that is sensitive to electroweak forces at keV levels of energy [5].

The XENON1T detector operated at this level of sensitivity as the most sensitive dark matter detector during its commission [2], while the newly operating XENONnT detector is currently the most sensitive dark matter detector [7]. In these detectors, there are two important elements that need to be reconstructed: the energy and the position of particle interactions. 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 [4].

This is made possible by a defined fiducial volumes and energy thresholds in XENON1T [4]. The placement of the fiducial volume is to reject extraneous interactions that take place near the walls of the detector [4]. The shape of the fiducial volume is shown in Figure 4 as the purple line [3]. This paper will focus on the use of the fiducial volume and position reconstruction in XENON1T.

The problem of position reconstruction, or *event localization*, is an inverse problem. Generally, the data we collect is the result of an event. Using this data, we aim to reconstruct the position of the event. This reconstructed position can then be used with the fiducial

volume to filter the extraneous events. Details on the inverse problem for the XENON1T detector are given in Sec. 2.2.

However, we know the limitations of this inverse problem while machines do not. All events must take place within the detector and, for a uniform position distribution of events, the reconstructed positions should be uniform as well. This is not the case for previous position reconstruction implementations, such as multilayer perceptrons, weighted sum position, and maximum photomultiplier tube (PMT) [13]. Events are reconstructed outside of the detector, shown by the grey points outside the black line in Figure 4 and tend to have an inward reconstruction bias [13].

In this paper, we show the novel application of a graph neural network for position reconstruction in XENON1T. Specifically, we will make use of a graph *convolutional* neural network (GCNN). These networks have primarily been used for classification, while the inverse problem we aim to solve is instead regression [9]. The results of this paper will show the acceptability of GCNNs for this class of inverse problems and for its application within particle physics.

2 Background

2.1 Dark Matter

The effects of dark matter are visible through various observations, such as the gravitational lensing within the Bullet Cluster [12]. This cluster is the result of the collision between two galaxy clusters. The baryonic matter is present as hot gases, visible through X-ray images, and acts as we’d expect: there is a drag between the two clusters that resulted in the Bullet Cluster’s distinct shape [12]. However, the massive portions of these clusters continue to move past each other as shown by the gravitational lensing map in Figure 1. These results are indicative of both the “dark” and weakly interacting nature of dark matter. The particles are not visible through electromagnetism, thus receiving the name dark matter. The particles

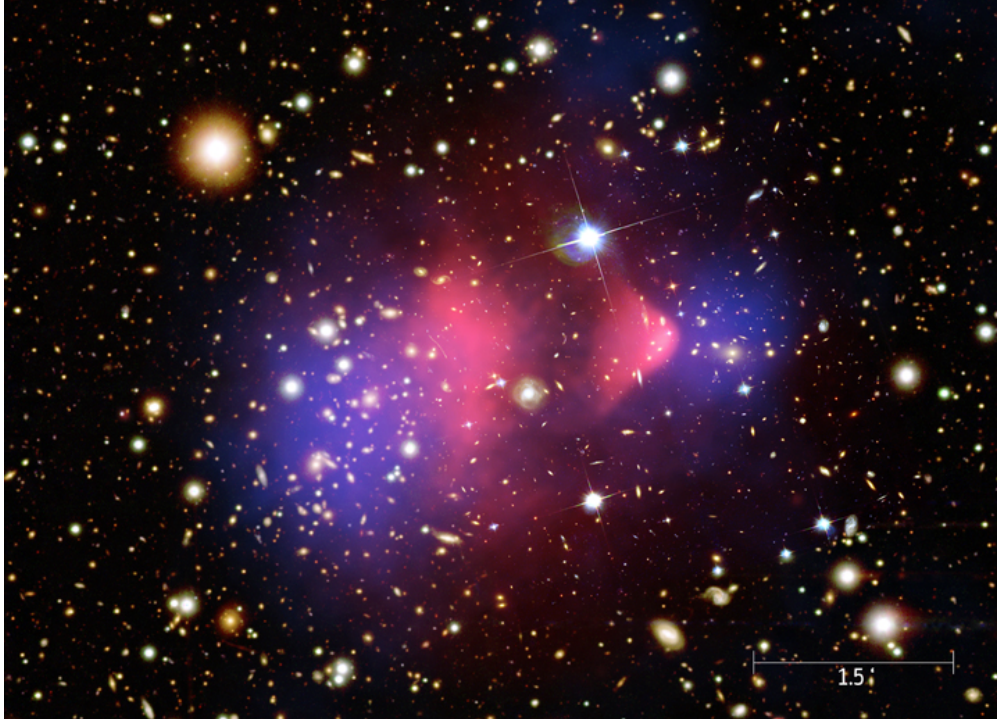


Figure 1: Composite image of the Bullet Cluster that shows the presence of dark matter through gravitational lensing. Pink regions are the X-rays of baryonic matter. Blue regions are from the gravitational lensing map. The scale on the bottom right is in arcminutes. [14]

have a non-negligible amount of mass that causes the gravitational lensing. This upper limit on the related mass and collision cross section is given by Markevitch et al. [12] as:

$$\frac{\sigma}{m} < 1 \text{ cm}^2 \text{g}^{-1} \quad (1)$$

where σ is the collision cross section and m is the particle mass. A more detailed explanation of how this limit is derived can be viewed in [12].

One candidate for dark matter particles is the weakly interacting massive particle (WIMP) [5]. This particle interacts by the electroweak force and has a lower bound of mass at the keV level [5]. Thereby, a direct detection experiment must be sensitive to nuclear recoils at the keV level. The XENON Collaboration produced the XENON1T detector for this exact purpose. The collaboration has since built and started commissioning the successor XENONnT, but the topics discussed in this paper will be with regard to XENON1T.

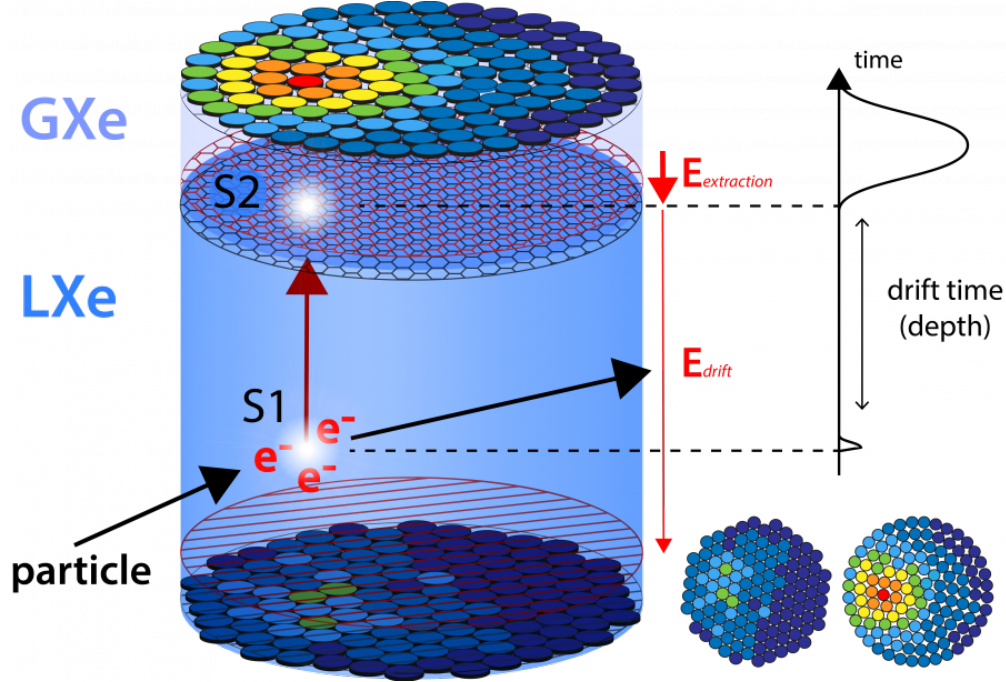


Figure 2: Diagram of the XENON1T detector and event scintillation and ionization. The first signal is the S1 where electrons and photons are emitted by the particle interaction. The electrons from the S1 are drifted upward by a constant electric field and extracted from the liquid phase to the gas phase using a stronger electric field. This produces the second signal S2 where more photons are emitted. The plane of circles at the bottom and top of the detector represent the photomultiplier tubes in the detector. The color of these circles indicate the photoelectrons observed for an S1 or S2 and is shown more clearly in Figure 3. [1]

2.2 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. As stated in Sec. 2.1, it was a requirement for the detector to be sensitive to keV energy levels in order to detect WIMPs. To meet these requirements, the detector's placement in LNGS, use of xenon, and presence of water shielding all serve to reduce the presence of extraneous signals that would appear from muons and more radioactive media [2]. It contains a target of ~ 1.3 tons of liquid xenon and a fiducial volume with a maximum radius of 42.84 cm [4]. The detector features 258 photomultiplier tubes (PMTs): 127 on the top array and 121 on the bottom array. Over the course of the experiment, these PMTs will break and the signal they would see can no longer be used.

There is a constant electric field present within the detector. In the liquid xenon phase, the field strength is 116.7 V/cm and guides the electrons upward [2]. At the liquid-gas interface starts a stronger electric field, $E_{\text{gas}} > 10$ kV/cm, that is used to extract these electrons and produce a second source of photons [2].

An event in the detector is defined by a particle scattering off xenon atoms. The recoiling nuclei of the xenon atoms results in a scintillation and ionization. The scintillation is observed by the PMTs and regarded as the S1 signal. The signal produced by the extracted electron at the liquid-gas interface is regarded as the S2 signal. An example of an S2 signal is shown in Figure 3. A diagram of the detector, S1, and S2 are shown in Figure 2.

The resulting light pattern from the S2 signal, or *hit pattern*, is the observation that is used for our inverse problem. Our inverse problem is to use this light pattern from the top array of PMTs to reconstruct the (x, y) position of the event in reference to the plane of the PMTs. It is unnecessary to include the depth, or z position, of the event as that is achieved using the time separating the S1 and S2 and the known applied electric field. This is the problem for position reconstruction.

Using the reconstructed positions, the data from the experiment can be filtered according to the position relative to the fiducial volume. As shown in Figure 4, many of these events are reconstructed near and beyond the wall of the detector. This can be caused by background events and broken PMTs that will reduce the accuracy of position reconstruction. Many of

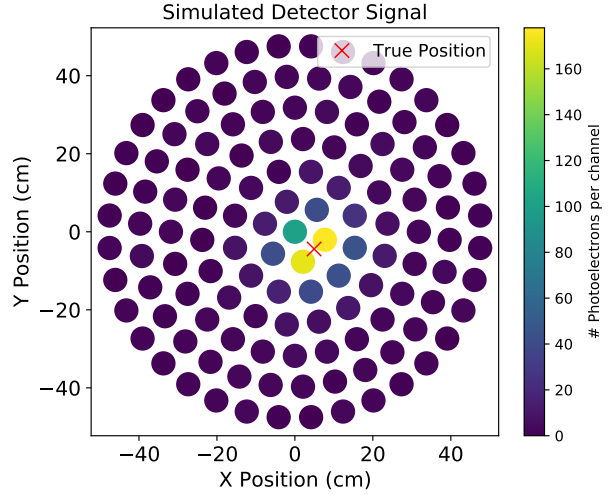


Figure 3: Example simulated hit pattern. Each circle represents a PMT in the top array of the detector. The position of each circle is the real position of the PMT in the detector. The simulation has a true position marked by the red x and created the shown hit pattern.

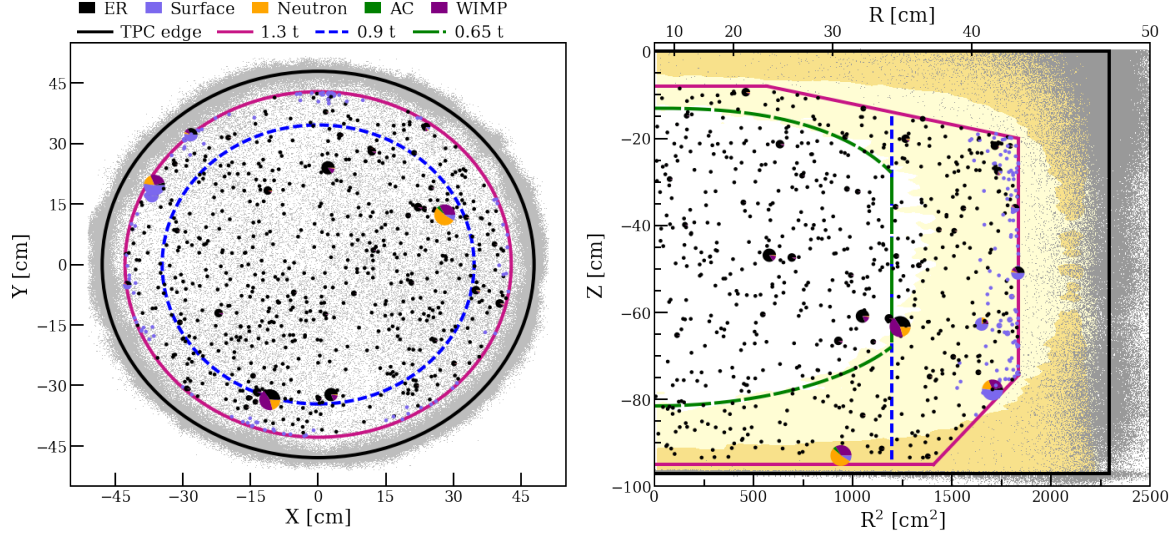


Figure 4: Spatial distribution of dark matter search data. The fiducial volume is shown in the magenta and the TPC edge is shown in black. Grey points are reconstructed events that are outside the fiducial volume. Many of these points are reconstructed outside the bounds of the detector. [3]

these background events are present along the walls due to the radioactivity in the detector's materials [3]. However, reconstruction of events outside the detector is clearly not possible for this experiment.

We aim to learn the detector by using a GCNN to minimize these outside reconstructions and minimize the number of reconstructions that are greater than 1 cm away from the true position when applied to simulation. At the same time, this will be the novel implementation of a GCNN for regression and in particle physics.

3 Neural Networks

Neural networks have become an increasingly relevant part to data analysis, and is also true for the XENON1T experiment [13]. A neural network is one section of machine learning that is heavily based off the structure of animal brains [11]. 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 by adjusting weights according to how wrong the machine has so far

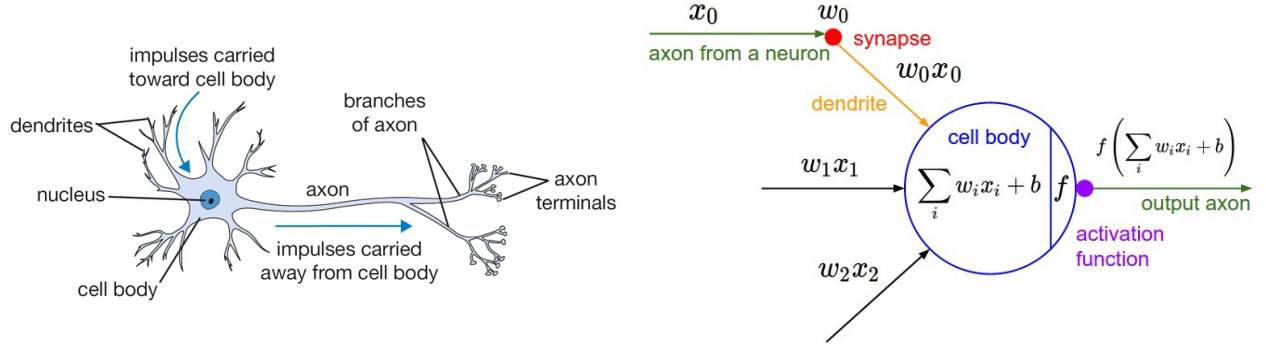


Figure 5: Analogy between animal neurons and machine neurons. [8]

been performing [11]. The exact manner for a neural network to change these weights is done through backpropagation [11]. Backpropagation is influenced by the structure of the neural network, such as the number of neurons, how they are connected between layers, and the activation functions that are used [11]. For this reason, it is relevant for us to try alternative neural networks that have not yet been applied.

An example of a single neuron, for both an animal and a machine, is shown in Figure 5. For the mathematical model of neuron, x_i represents the values of inputs, or a perceived stimulation in the case of an animal. These inputs are adjusted according to the weights w_i which are learned by the model. A final weight that is additive instead of multiplicative is included as b . Finally, this is passed through an activation function. There are various potential activation functions that can be used, of which two are shown in Figure 8. For a more detailed explanation of neural networks see [8].

For our problem, the input layer will be the signal seen by the PMTs 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 were motivated by the success of convolutional neural networks (CNNs) [9].

3.1 Convolutional Neural Networks

A convolutional neural network is a specific kind of neural network that features a convolution layer and a pooling layer [11]. The input to a CNN frequently comes in the form

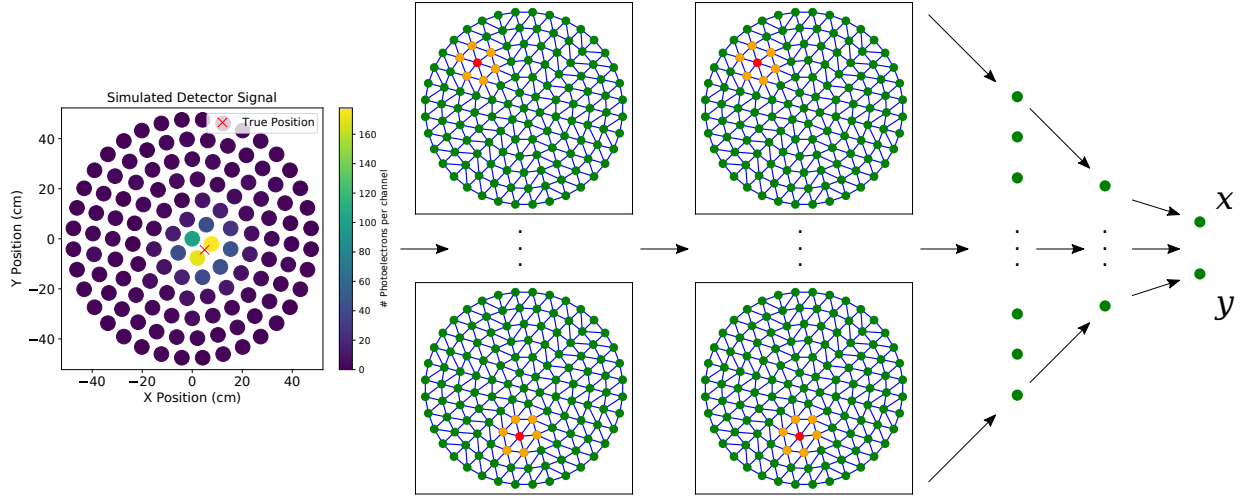


Figure 6: The graph convolutional neural network structure that the results of this paper are based on. The final structure features the signal input layer, two propagation (graph convolution) layers, two fully connected layers, and a final output layer for the (x, y) position of the interaction. Between each layer is a ReLU activation function. There are a total of 57,486 trainable parameters.

of a matrix where there is a locality between the elements in the matrix [11]. Many CNNs are applied to images, where pixels are the elements of the matrix. The convolution layer makes use of a kernel that does element multiplication of weights to a submatrix of the input [11]. This is repeated as a scan over the entire matrix. In the case of an image, smaller sections of the image pass through this kernel until all of the image is scanned. This results in the network learning the important features of the matrix, or image. In an image, these features can come as lines separating objects, textures, or the presence of certain colors.

However, the structure of our dataset does not have a form that can be easily made into a matrix. The top array of PMTs has a radial formation and symmetry. A transformation into a rectangular matrix would not preserve the locality between nearby PMTs. Therefore, we would need an algorithm that is capable of making use of datasets with any possible structure.

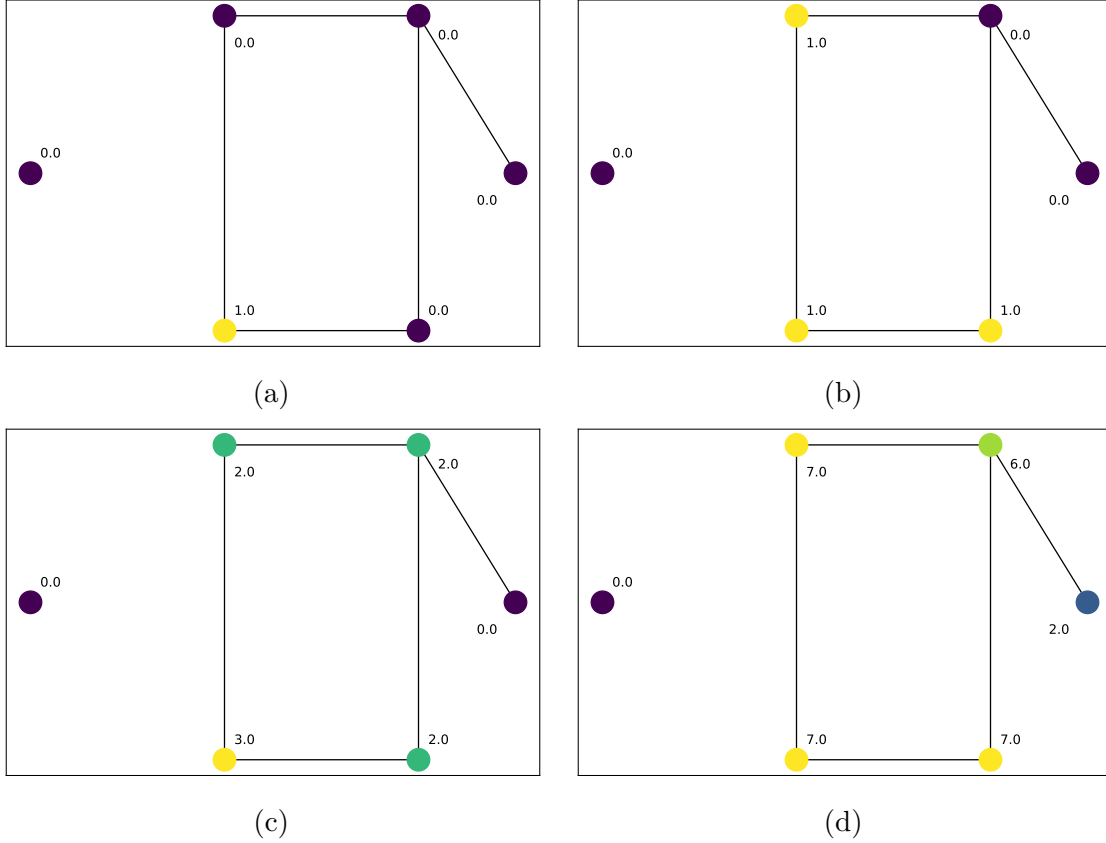


Figure 7: Example propagation from graph convolutions. The value immediately next to each node is an attribute of that node and will be passed during propagation. These are passed according to edges that connect each of the nodes. The initial signal is shown in (a), the result after the first propagation in (b), after the second propagation in (c), and after the third propagation in (d).

3.2 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 [9]. The graph convolution layer that we use and that was proposed by Kipf and Welling propagates the values of nodes according to the edges [9]. The value of a node will increase according to the value of itself and the values of the nodes connected to it, or *neighbors*. This is explained in further detail in Appendix A.

Figure 7 is a toy example of how a graph convolution layer propagates values. The initial value for every node is given in Figure 7a. The node will propagate the value of 1 to its

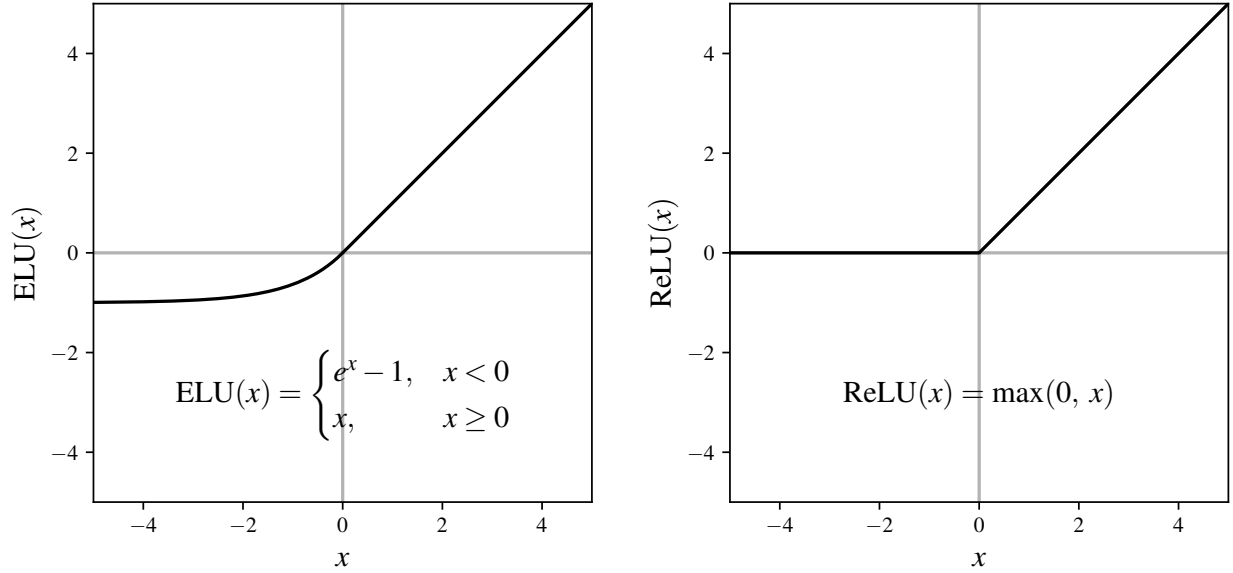


Figure 8: Plots of an Exponential Linear Unit (ELU) and the Rectified Linear Unit (ReLU). We use ReLU activation functions in our network structure.

neighbors and back to the node itself. Since all other nodes have values of 0, only this node will have an effect. We would expect to see three nodes with values of 1, as is shown in Figure 7b. Continuing the propagation, results in graph in Figure 7c and one step further results in 7d. The nodes nearest the initially high value node will also become high in value, while nodes that are further away will have a lesser value. Nodes that are completely disconnected will see no changes.

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 additional values. These positions are intended to provide additional weighting when learning, but the event hit pattern will be the main factor in reconstructing the positions of events. We chose to use the rectified linear unit (ReLU) as our activation function for all layers. This function is defined as:

$$\text{ReLU}(x) \equiv \max(0, x). \quad (2)$$

This choice was to create a strict cut off for the wall of the detector. The use of the exponential

linear unit (ELU) would not provide this strict cut off. This activation function is defined as:

$$\text{ELU}(x) = \begin{cases} \alpha(e^x - 1), & x < 0 \\ x, & x \geq 0 \end{cases}. \quad (3)$$

With regard to the network structure, we were influenced by the success of image classifiers, such as AlexNet [10]. 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 graph neural network (GNN) for reconstructing 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 thought the use of these graph convolution layers would aid in maintaining a contrast between high value and low value regions, as was seen in the toy example in Figure 7

We considered several graph structures shown in Figure 9. We ultimately chose the Delaunay triangulated graph Figure 9c. Delaunay triangulation is a specific triangulation method that makes use of the circumcircles of the triangles it creates. These circumcircles are required to not contain the vertex of any other triangle in the set [6]. Therefore, the connected triangles should occur between nearby points and created the triangulation that we hoped for and chose to use [6].

This triangulation created a 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 a degree less than 6. There is potential for finding the graph structure that describes the detector’s data best on certain metrics, but for this paper we chose to use the Delaunay triangulated graph as a heuristic.

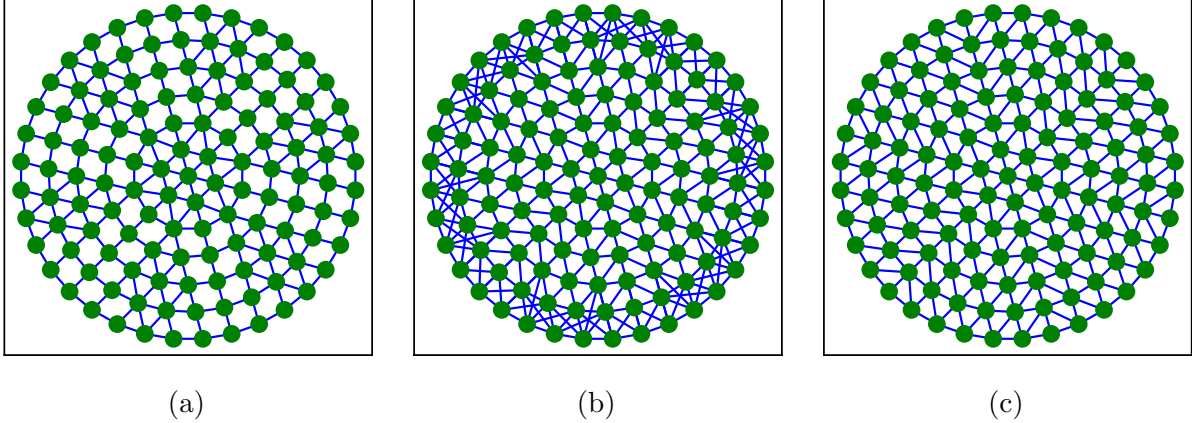


Figure 9: Three of the considered graph structures: Radius $R = 10$ cm neighbors (a), k -nearest-neighbors $k = 6$ (b), Delaunay triangulation neighbors (c). Each node here is a photomultiplier tube in the top array of the XENON1T detector. The positions of each tube in the detector was used for each of the explored graph structure approaches.

4 Results

4.1 During Training

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 – approximately within the first 50 epochs – training would 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 (MSE) 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 MSE 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 were present within the

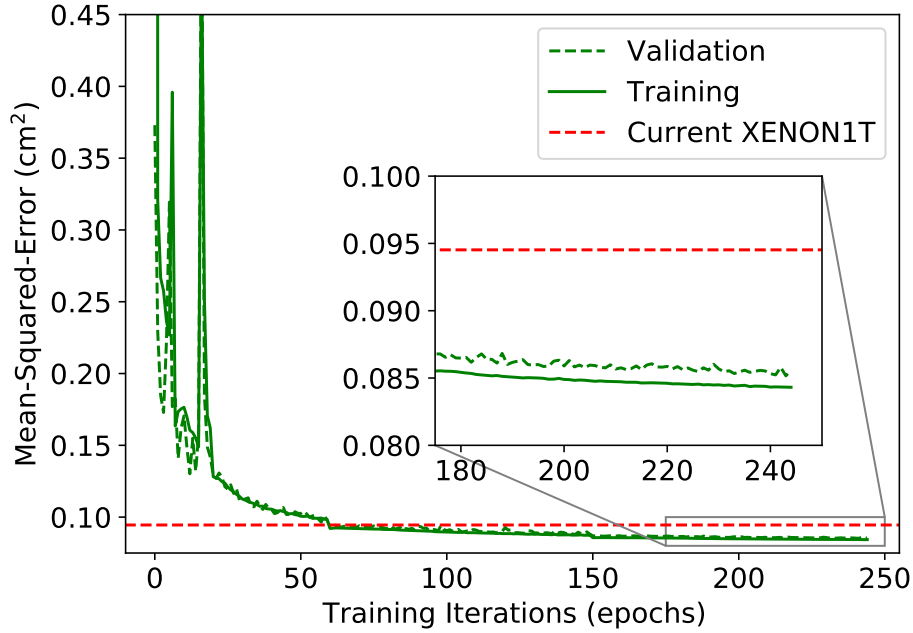


Figure 10: Mean-Squared-Error (MSE) in square centimeters of our algorithm for the training set and validation set during the training process. The minimum MSE for the current state-of-the-art is given in red at 0.0945 cm^2 . This is the benchmark that our algorithm had to pass during training. Spikes 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 .

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.

Our algorithm was able to outperform the state-of-the-art in training, which was a good indicator for the overall performance. Much of the work for this stage was in optimizing the learning rate used during training. 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 10 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 researcher.

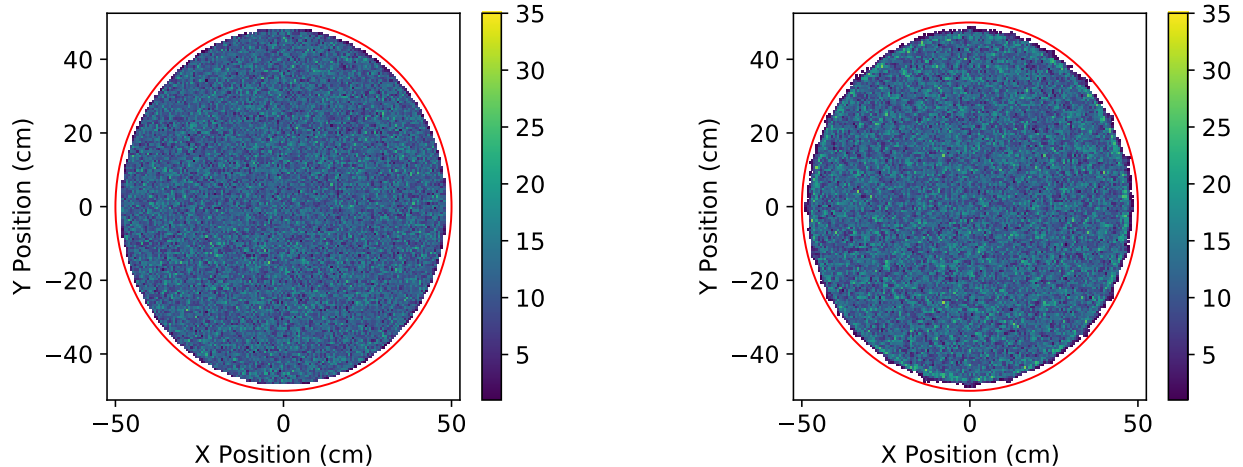


Figure 11: 2D histograms of the true positions (left) and reconstructed positions (right) at 150 bins. The red circle is the wall of the detector (50 cm). The edge of the reconstructed positions is noticeably more jagged.

4.2 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 [16], we focused on the results of the validation set which is made of 197,975 simulated events.

First to check is if any reconstructions occurred outside of the detector. 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.

Second to check is the number of events that are reconstructed further than 1 cm from the true position. Of the 197,975 events in the validation set, 1,680 were reconstructed at greater than 1 cm away from the true position, or about 0.85% of the set. As can be seen in Figure 12, many of the mistakes are made along the walls of the detector and explains the jagged edge found in Figure 11. If we reduce the region we count on to within the fiducial volume ($R = 43$ cm), we find 123 of the 197,975 events are mis-reconstructed, or about 0.06%

of the validation set.

The last important metric we look at is the resolution of our reconstructions. For this we define 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 13. These results are competitive with the reconstructions of the state-of-the-art. 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 , ΔY , and radius difference further confirms the positive performance of our GCNN.

The visual uniformity of the 2D histogram in Figure 11 had us question the true uniformity of both distributions. We made use of Ripley's \hat{K} and \hat{L} functions [15]. These

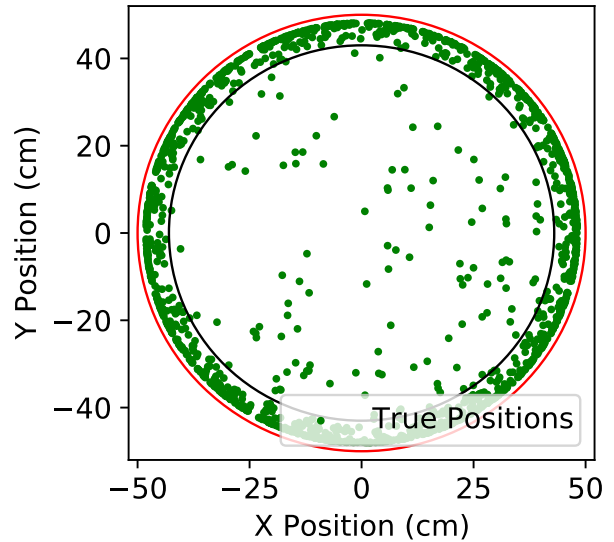


Figure 12: Scatter plot of the true positions of our algorithm's mis-reconstructions. The red circle is the wall of the detector (50 cm); the black circle is the largest radius of the fiducial volume (43 cm). Of the 197,975 simulated events, there were 1,680 mis-reconstructions which are shown here. Within the fiducial volume, there were 123 out of 1,680.

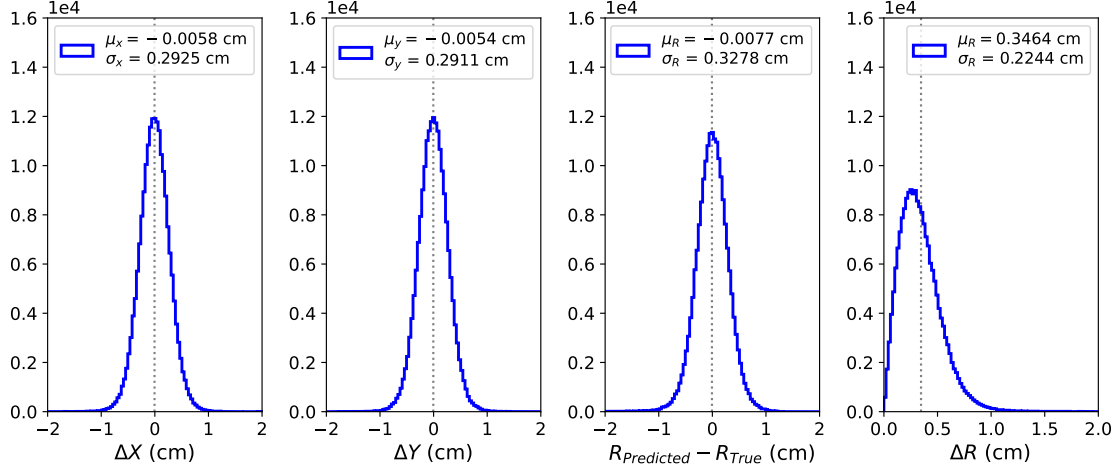


Figure 13: 1D histograms of the reconstructed position minus the true positions. There are 100 bins between -2 cm and 2 cm on ΔX , ΔY , $(R_{\text{Predicted}} - R_{\text{True}})$ and 100 bins between 0 cm and 3 cm on ΔR . The left three histograms are near Gaussian curves of the same statistics.

functions are used for describing spatial uniformity and are defined as:

$$\hat{K}(r) \equiv \frac{1}{n\lambda} \sum_{i \neq j} I(d_{ij} < r)$$

$$\hat{L}(r) \equiv \left(\frac{\hat{K}(r)}{\pi} \right)^{1/2}.$$

Here n is the number of points, λ is the number of points per unit area, $I(\cdot)$ is an indicator function, d_{ij} is the distance between data points i and j . In the ideal case for perfect, spatial uniformity, $\hat{K}(r) = \pi r^2$ and $\hat{L}(r) = r$.

The results of applying Ripley's \hat{L} function are shown in Figure 14 and indicate that the reconstructed positions are more spatially uniform than the true positions. Both the predicted and true positions are more dispersed than the ideal case, however, these values are not significantly large enough to merit further discussion.

4.3 Experimental Data

We applied this trained model to $\text{Kr}^{83\text{m}}$ data. However, the detector at that time featured multiple broken PMTs. Nodes that represent these broken PMTs receive values of 0 instead.

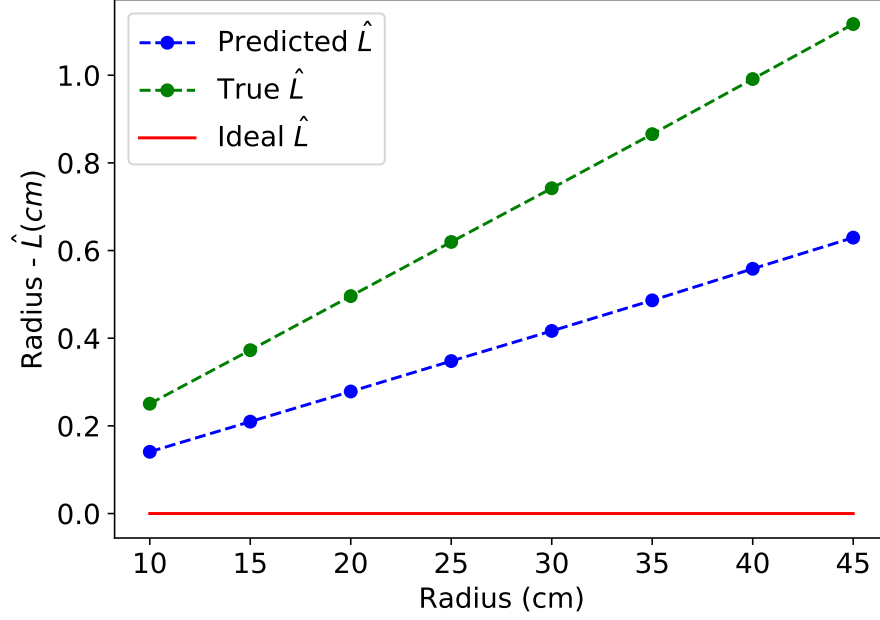


Figure 14: Ripley's \hat{L} function shifted according to radius. The closer to the ideal case is better. These values are indicative of a more dispersed set of points.

In addition to this, the data is no longer free of background events. It's expected for the model to do poorly. One of the first steps we had to do was normalize the data to a similar level that was used during training. This was done according to:

$$\hat{x} \propto \frac{x}{\sum_{i=1}^{127} x_i} \quad (4)$$

where x is the event signal and \hat{x} is the normalized signal.

The result of the model on normalized, experimental data is better than we expected. The low density areas and inward curves along the edge of the reconstructions in Figure 15 occurs in the same positions as the broken PMTs. The effect of 0 values at the associated nodes indicates that events would not take place under those nodes and should therefore be reconstructed elsewhere. Although this result is not useful for position reconstruction, this does indicate the model is able to learn the detector. Another notable result is the limited number of events that are reconstructed outside of the detector. For a model that was trained

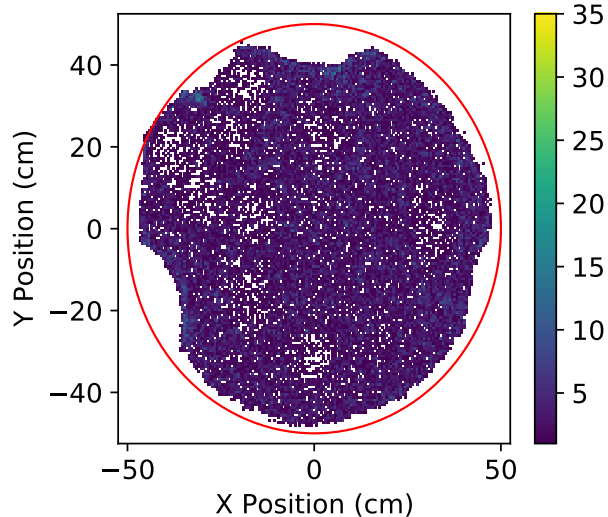


Figure 15: 2D histogram of the reconstructed positions of $\text{Kr}^{83\text{m}}$ experimental data. The red circle is the wall of the detector (50 cm). The low density areas and inward curves along the edge the histogram are the same positions as the broken PMTs.

on a “perfect” detector, these results are much better and more meaningful than expected.

5 Conclusion & Future Work

We presented the novel application of a graph neural network – more specifically a graph *convolutional* neural network – in particle physics as well as one of the first regression applications of such a network. Our algorithm performed at a standard that is competitive with the state-of-the-art position reconstruction algorithm for XENON1T. However, we do not think that this algorithm is the best approach to take in the future. We suspect that a broader graph neural network would work better for position reconstruction for the upcoming XENONnT experiment. The use of a graph neural network has shown the ability to maintain locality between PMTs, our graph’s nodes, but is not attributed to the use of this specific graph convolution. We believe an alternative graph convolution layer would perform better, but also suspect that a graph convolution layer may not be necessary for the model to perform well. The effects of broken PMTs clearly produce a negative influence on the position reconstruction and indicate their importance during training.

We have shown that graph neural networks are applicable to particle physics detectors. They are able to preserve the locality of sensors in the detector when necessary for the application. Constructing these graphs will be unique for the problem to be solved and the detector itself, and creates a potential area of research to find the most optimal graph structure. We have shown that graph neural networks can be applied as both classifiers and regressors with significant effectiveness.

6 Acknowledgments

I would like to thank my advisor Professor Christopher Tunnell for providing guidance throughout my early research career. I would like to thank Dr. Aaron Higuera and Rice University graduate student Shixiao Liang for the assistance they gave on the simulations and graph network structure during this research. I would like to thank the Rice Astroparticle and the Data-Intensive Discovery Accelerated by Computational Techniques for Science (DIDACTS) groups for motivating me during this research and beyond it.

Appendices

A GCNN Propagation

The exact propagation rule is given by the equation:

$$H^{(l+1)} = \sigma \left(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W^{(l)} \right) \quad (5)$$

where $H^{(0)}$ will be our initial input signal, \tilde{A} and \tilde{D} are respectively modified, unweighted adjacency and modified degree matrices, $W^{(l)}$ is the trainable weights matrix for the l^{th} layer, and $\sigma(\cdot)$ is an activation function which we chose as ReLU [9]. This activation function is defined in equation 2 and shown in Figure 8.

The adjacency and degree matrices will come directly from the Delaunay triangulated graph shown in 9c. The modified adjacency and degree matrices are to include self-loops. Thus, nodes will connect to themselves and propagate values to themselves when passed through the graph convolution layers. These effects are shown in the toy example in Figure 7. The modified adjacency and degree matrices take the form:

$$\tilde{A} = A + I_N$$

$$\tilde{D} = D + I_N$$

where I_N is a $N \times N$ identity matrix where N is the number of nodes. When applied to our problem, the matrices take the form:

$$\tilde{A}, \tilde{D} \in \mathbb{R}^{127 \times 127}$$

$$H^{(0)} \in \mathbb{R}^{127 \times 3}$$

$$H^{(1)}, W^{(0)} \in \mathbb{R}^{127 \times w}$$

where w will depend on our choice of network structure.

References

- [1] L. Althüser. Light collection efficiency simulations of the xenon1t experiment and comparison to data. Master's thesis, WWU Münster, 2017.
- [2] 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>.
- [3] E. Aprile, J. Aalbers, F. Agostini, et al. Dark matter search results from a one ton-year exposure of xenon1t. *Phys. Rev. Lett.*, 121:111302, Sep 2018. doi: 10.1103/PhysRevLett.121.111302. URL <https://link.aps.org/doi/10.1103/PhysRevLett.121.111302>.
- [4] 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>.
- [5] 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>.
- [6] J. Blazek. Chapter 11 - principles of grid generation. In J. Blazek, editor, *Computational Fluid Dynamics: Principles and Applications (Second Edition)*, pages 373–413. Elsevier Science, Oxford, second edition edition, 2005. ISBN 978-0-08-044506-9. doi: <https://doi.org/10.1016/B978-008044506-9/50013-8>. URL <https://www.sciencedirect.com/science/article/pii/B9780080445069500138>.
- [7] The XENON collaboration, E. Aprile, J. Aalbers, et al. Projected wimp sensitivity of the xenonnt dark matter experiment, 2020.

- [8] Yosefa Gilon, Fei-Fei Li, Ranjay Krishna, and Danfei Xu. Cs231n convolutional neural networks for visual recognition, 2021. URL <https://cs231n.github.io/neural-networks-1/>.
- [9] 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>.
- [10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012. doi: 10.1145/3065386.
- [11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, May 2015. ISSN 1476-4687. doi: 10.1038/nature14539. URL <https://doi.org/10.1038/nature14539>.
- [12] M. Markevitch, A. H. Gonzalez, D. Clowe, et al. Direct constraints on the dark matter self-interaction cross section from the merging galaxy cluster 1e 0657-56. *The Astrophysical Journal*, 606(2):819–824, May 2004. ISSN 1538-4357. doi: 10.1086/383178. URL <http://dx.doi.org/10.1086/383178>.
- [13] 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>.
- [14] X ray: NASA/CXC/CfA/M.Markevitch et al.; Optical: NASA/ST-ScI; Magellan/U.Arizona/D.Clowe et al.; Lensing Map: NASA/STScI; ESO WFI; Magellan/U.Arizona/D.Clowe et al. Bullet cluster, 2006. URL <https://chandra.harvard.edu/photo/2006/1e0657/more.html>.
- [15] B. D. Ripley. The second-order analysis of stationary point processes. *Journal of Applied Probability*, 13(2):255–266, 1976. doi: 10.2307/3212829.

- [16] Igor V. Tetko, David J. Livingstone, and Alexander I. Luik. Neural network studies.
1. comparison of overfitting and overtraining. *Journal of chemical information and computer sciences.*, 35(5), 1995. ISSN 0095-2338.