# Extracting low energy signals from raw LArTPC waveforms using deep learning techniques — A proof of concept

Lorenzo Uboldi[a], David Ruth[b], Michael Andrews[c], Michael H. L. S. Wang[d,*], Hans-Joachim Wenzel[d], Wanwei Wu[d], Tingjun Yang[d]

[a] *CERN, The European Organization for Nuclear Research, 1211 Meyrin, Switzerland*
[b] *Niles North High School, Skokie, IL 60077, USA*
[c] *Carnegie Mellon University, Pittsburgh, PA 15213, USA*
[d] *Fermi National Accelerator Laboratory, Batavia, IL 60510, USA*

**Abstract**

We investigate the feasibility of using deep learning techniques, in the form of a one-dimensional convolutional neural network (1D-CNN), for the extraction of signals from the raw waveforms produced by the individual channels of liquid argon time projection chamber (LArTPC) detectors. A minimal generic LArTPC detector model is developed to generate realistic noise and signal waveforms used to train and test the 1D-CNN, and evaluate its performance on low-level signals. We demonstrate that our approach overcomes the inherent shortcomings of traditional cut-based methods by extending sensitivity to signals with ADC values below their imposed thresholds. This approach exhibits great promise in enhancing the capabilities of future generation neutrino experiments like DUNE to carry out their low-energy neutrino physics programs.

*Keywords:* Low-energy neutrinos, LArTPC, Triggering, Signal processing, Machine learning, Convolutional neural networks

## 1. Introduction

The liquid argon time projection chamber (LArTPC) has been successfully deployed in a number of recent and currently running neutrino experiments and is the technology of choice for massive, next-generation neutrino experiments like the Deep Underground Neutrino Experiment (DUNE) [1]. Born out of combining the novel three-dimensional imaging capabilities of the time projection chamber (TPC) [2] with the unique properties of liquefied noble gases like liquid argon (LAr), it represents the modern, electronic equivalent of the bubble chamber [3].

Unlike the latter, LArTPC detectors are "always-on" devices, continuously detecting and acquiring signals induced by ionization charges on wire planes at the end of the drift path. Furthermore, the electronic readout of these signals from multiple wire planes with different angular orientations that provide 2 spatial coordinates, together with a third determined from drift times, enables the automated reconstruction of detailed topologies, while simultaneously performing calorimetry from the integrated charge.

Alongside the unique benefits provided by the TPC are the excellent characteristics of LAr which include high ionization and scintillation yields. The ability to transport electrons efficiently across long drift distances due to the vanishing electronegativity and low dispersion in LAr permits the high spatial resolution possible with finely segmented wire planes. This allows very large detectors like the DUNE far detectors [1] to be built. Finally, the high density of LAr makes it an ideal neutrino detector due to the low neutrino-nucleon interaction cross sections.

The powerful capabilities of the LArTPC make it an excellent choice for DUNE's long baseline physics program with goals that include determining the neutrino mass hierarchy, observing CP violation in the lepton sector, and making precise measurements of the oscillation parameters using the wide-band beam from Fermilab [4, 5]. Performing these measurements with a LArTPC will not be too challenging due to the relatively high incident neutrino energies, ranging from hundreds of MeVs to a few GeVs, and the reduced background levels from cosmogenic and atmospheric sources made possible by the deep underground location of the far detectors.

Beyond its long-baseline program, DUNE's physics goals also include the detection of neutrinos from core-collapse supernovas, searches for nucleon decay, studies of solar neutrinos, and atmospheric neutrino oscillation studies to supplement the long-baseline measurements [6–8]. Of these, the solar and core-collapse supernova neutrinos involve low energy neutrinos in the 1 MeV (solar) to 10 MeV (supernova) range. Ionization from the products of their interactions in the LAr can induce signals that are close to the noise threshold, making them challenging to detect. This

*Corresponding author
*Email address:* mwang@fnal.gov (Michael H. L. S. Wang)

is further exacerbated by the conventional approach of applying minimum ADC threshold cuts to discriminate signal waveforms from noise which results in poor low-energy efficiency.

In this paper, we take a deep learning (DL) approach to address the drawbacks of conventional threshold-based methods and to optimize the efficiency to low-energy neutrinos. We develop deep learning techniques and apply them to the raw waveforms from individual LArTPC wires, to detect the presence of a signal and narrow down its location in the full waveform in terms of a region of interest (ROI). While DL methods have been applied to LArTPC data, they tend to be applied at later stages in offline reconstruction and on 2D "images" based on wire plane views. To our knowledge, this is the first attempt to apply such methods directly to the raw waveforms associated with single LArTPC wires. This implies potentially promising applications of this method in low-level filtering and triggering in online DAQ systems.

In the discussion that follows, we begin by describing a minimalist but realistic *toy* LArTPC detector simulation that we created and used to develop and test our deep learning models. This is followed by a detailed description of the neural network architecture. The model's performance on training and validation sets is presented. It is then tested on an independent sample to determine various performance metrics including the detection efficiency as functions of energy deposited and ADC counts produced. The main purpose of this paper is to demonstrate a proof-of-principle for using deep learning networks in extracting signals from raw LArTPC waveforms. After successfully accomplishing this, we conclude by summarizing the results and discussing future work and application of the method.

## 2. A minimal toy LArTPC detector

To test the idea of using DL-based methods for extracting signals from raw LArTPC waveforms, we first developed a software simulation of a minimal *toy* LArTPC detector. It consists of a LAr volume defined by a rectangular prism measuring $50 \times 50 \times 180$ cm$^3$, with the long dimension oriented along the $z$ axis and the two shorter ones along the $x$ and $y$ axes, centered at $(x, y) = (25, 0)$. There are two instrumented anode wire planes lying in the $y-z$ plane consisting of an induction plane at $x = 0.075$ cm and a collection plane at $x = -0.075$ cm. Both planes consist of 280 equally spaced parallel wires with a pitch of 0.25 cm, and whose axes are oriented at $+60°$and $-60°$with respect to the $y$ axis for the induction and collection planes, respectively. A uniform electric field is oriented along the $x$ axis with a field strength of 500 V/cm in the main drift region.

The propagation of particles through the LAr volume was simulated using the GEANT4 simulation toolkit [9, 10] with a step limit of 100 microns through the *LArG4* package in the LArSoft framework [11]. If the particle's interaction with the medium in a given step led to energy deposition, the amount of this energy and the position of the interaction was recorded in a *SimEnergyDeposit* object. This information from the GEANT4 simulation was passed on to the next stage which simulated the propagation of electrons to the readout planes and the response of the detector that ultimately produced the digitized raw wire waveforms from the LArTPC electronics.

The detector response was simulated by first calculating the initial number of ionization electrons resulting from the energy deposition, based on the ionization work function in LAr ($\#\,^{\text{electrons}}/_{\text{GeV}} = {}^{1}/_{2.36\times10^{-8}\text{GeV}}$) [12]. Electron-ion recombination effects were taken into account by multiplying this initial number by a factor determined from the *box model* of recombination [13] modified to extend into the low $dE/dx$ region [14]. Diffusion effects were then simulated by first splitting up the total number of electrons after recombination into clusters of 600 electrons [15]. The initial drift time for all clusters was calculated from the drift velocity and the distance between the original energy deposition and the wire plane. Diffusion in the longitudinal direction was simulated by smearing this drift time according to a Gaussian distribution for each cluster. Diffusion in the transverse direction was simulated by smearing the transverse positions of the original energy deposition by a Gaussian distribution for each cluster, providing their spatial coordinates at the wire planes. These smeared coordinates were used to determine the wire closest to a cluster in a plane, on which an induced signal would be simulated. The amount of energy and number of electrons for that cluster were recorded in a *SimChannels* object, and were identified by the channel number corresponding to the wire. Only channels associated with clusters were stored in the object, and the cluster information was saved in the time bin (TDC) corresponding to its drift time. The waveforms from all LArTPC wires were assumed to be digitized at a rate of 5.05 MHz (198 ns/sample), starting from the instant energy was deposited by the GEANT4 simulated tracks (assumed to all occur simultaneously), and lasting for a duration corresponding to 2,048 samples.

To simulate the waveforms produced by the LArTPC wires, we followed a procedure similar to that outlined in References [16–18]. Each waveform is composed of a *pure signal waveform* and *noise waveform* which are generated separately and then added together. To generate a pure signal waveform, we began with a parameterization of the signal induced by a single electron on a wire, as a function of the $j$'th TDC bin, in terms of a quadratic ($\propto j^2$) for the collection plane, and an asymmetric sinusoid, where the amplitude of the negative-going half cycle was 10% larger, for the induction plane. This induced signal was then convoluted with the electronics response which was modeled using the parameterization described in Reference [18], resulting in the *single electron waveforms* shown in Figure 1. For each LArTPC wire, the total number of ionization electrons stored in the *SimChannels* object

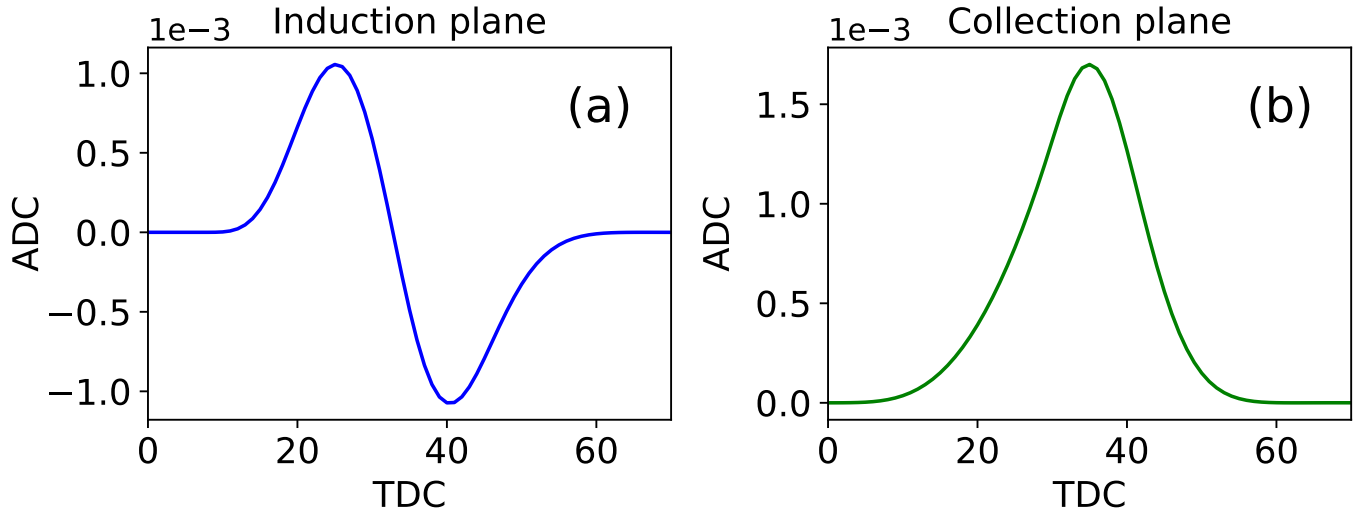



Figure 1: Signal waveforms induced by a single electron on (a) induction and (b) collection plane wires convolved with the electronics response.

for each TDC bin, was convoluted with the single electron waveform for the appropriate wire plane in order to produce the pure signal waveform.

The generation of noise waveforms began in the frequency domain where the complex amplitudes of 1,025 (2,048/2+1) equally spaced noise frequencies from 0-2,525 KHz were generated [17]. The mean of the modulus $r$ of these complex amplitudes was parameterized according to:

$$\begin{aligned} r(f_j) &= |re^{-i\phi}| \\ &= p_0 \times e^{-\frac{1}{2}\left(\frac{f_j - p_1}{p_2}\right)^2} \times e^{-\frac{1}{2}\left(\frac{f_j}{p_3}\right)^{p_4}} + p_5 + e^{-p_6(f_j - p_7)} \end{aligned} \quad (1)$$

where $f_j$ represents the midpoint of the $j$'th frequency bin in kHz and the parameters $p_k$ are given by $p_0 = 4450$, $p_1 = -530$, $p_2 = 280$, $p_3 = 110$, $p_4 = -0.85$, $p_5 = 18$, $p_6 = 0.064$, and $p_7 = 74$. Figure 2 shows a plot $r(f_j)$. The modulus was fluctuated around this mean according to a modified Poisson distribution of the form $P(x) = e^{-\mu}\mu^x/(x-1)!$ with $\mu = 0.28$, while the phase $\phi$ was generated uniformly from $0 \rightarrow 2\pi$. The parameters used in this model were chosen to generate noise very similar to that observed in actual detectors like that described in Reference [19]. Once the complex amplitudes at each frequency bin were generated, an inverse fast Fourier transform was performed to produce a 2,048-bin time sequence representing a noise waveform. These noise waveforms were generated uniquely for each channel and for every event.

A pure signal waveform was added to a noise waveform to produce the simulated LArTPC waveform for a wire if it was associated with a *SimChannels* object; otherwise, only the noise waveform was used. From here on, we will refer to the simulated waveform formed from the sum of a pure signal and a noise waveform simply as a *signal waveform*. Examples of these simulated waveforms are shown in Figure 3 for the generated samples used in training and testing the DL model described below, and in evaluating its performance.

The toy detector simulation described above was not meant to be the most detailed and realistic simulation of a LArTPC detector. Some simplifications and approximations were made for the purpose of demonstrating a proof-of-concept. One main approximation is the assignment of a drifted cluster to the closest wire, simulating an induced signal only on that wire, and ignoring the influence of clusters assigned to neighboring wires. The dependence of the induced signal on the distance of the wire from the cluster that produced it was also not taken into account. These simplifications will have an effect on the size and shape of the simulated signals, but not in a fundamental way that would mask the major features and capabilities of our method, which we are trying to highlight, relative to those of the traditional threshold-based approach. Our primary purpose here is to introduce and provide a detailed description of the technique and demonstrate its effectiveness on a simplified toy simulation. The validity of these assumptions are put to the test in a separate paper [19], where we focus on the application of this technique to actual data from the ArgoNeuT experiment.

## 3. Deep learning approach to LArTPC waveform recognition

### *3.1. 1D-CNN model architecture*

The architecture of the neural network we developed for detecting signal waveforms in a LArTPC is the one-dimensional convolutional neural network (1D-CNN) shown in Figure 4. Starting on the left is the waveform presented to the input of the network, which consists of three one-dimensional convolutional layers (Conv1Ds), all of which use rectified linear unit (ReLU) activation functions, defined by $f(x) = \max(0, x)$. Each Conv1D is immediately followed by a pooling layer that reduces the size of the input feature map. The first two are maximum pooling

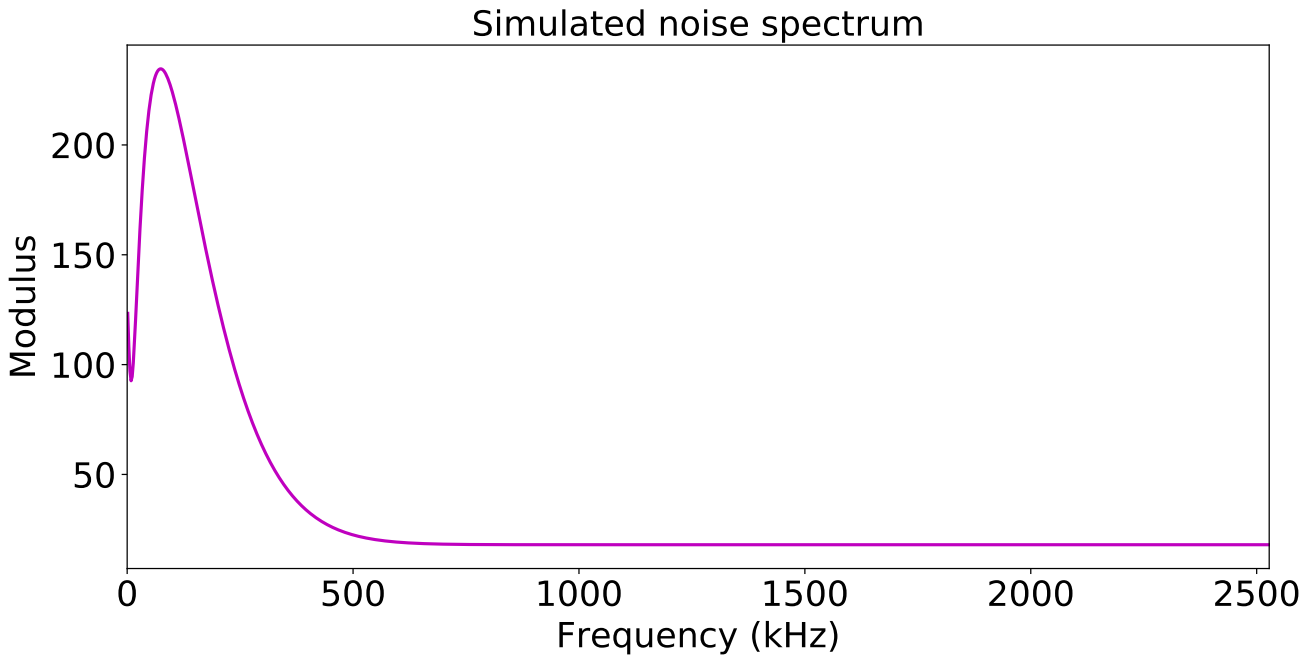



Figure 2: The plot above shows the noise spectrum used to simulate noise waveforms. More precisely, it shows the mean value of the modulus $r$ of the complex amplitude of each noise frequency component given by Equation 1.

layers with pool sizes of 2, and the last is a global pooling layer. There are 16, 32, and 64 filters or kernels in the first, second, and third Conv1D layers, respectively. The kernel sizes in this sequence of three layers are 3, 5, and 9, respectively, with stride lengths of 2 for the first two and 1 for the last. Two dropout layers are used, one with a dropout fraction of 0.1 after the second maximum pooling layer, and the other with a dropout fraction of 0.2 after the global pooling layer. By randomly ignoring a fraction of their inputs, dropout layers help prevent the formation of co-adaptations between layers, which do not generalize well to unseen data and could therefore cause overfitting [20]. The outputs of the global pooling layer terminate into a dense layer with a single node that is activated by a sigmoid function, defined by $f(x) = 1/(1+e^{-x})$. This function yields an output bounded between 0 and 1, which can be conveniently interpreted as the probability that the waveform contains a signal or not. This network has a total of 21,217 trainable parameters.

### *3.2. Training the model*

To create the simulated sample of digitized LArTPC waveforms used to train the model, radiological events from the $\beta$ decay of the $^{39}$Ar nuclide contaminating LAr were first generated. The interactions of the particles produced in these events with the LAr volume were then simulated using the GEANT4 toolkit, followed by the detector response simulation, both of which are described in Section 2. Next, all the particles in an event that deposited energy in the LAr, leading to a detectable signal from at least one wire, were then identified. This was ensured by requiring the maximum ADC value of the pure signal component of the digitized wire waveform to be $> 3$ ADC counts. In addition, the minimum energy of the parent ionizing particle was required to be $\geq$ 50 keV, and the maximum number of ionization electrons associated with the signal was required to be $< 11,000$. For each particle identified above, we randomly selected a single wire channel among all those with signals associated with this particle, satisfying the requirements above. This was done in order to minimize possible correlations between signals from neighboring wires originating from the same particle, which could impact the training process negatively. Since there can be multiple signal contributions from different particles in a given channel, only the largest contribution, based on the energy deposited, was selected. A cutout of the full waveform consisting of 200 time bins, with the selected signal region randomly positioned within it, was then created. The training set used in the procedure described below consisted of such 200-tick waveforms.

The model was trained separately for the induction and collection planes of the LArTPC described in Section 2. The total number of samples used to train the model was ≈2.88 M (≈3.30 M) for the induction (collection) plane. A separate validation sample of ≈721 K (≈824 K) in the induction (collection) plane was not used to train the model directly, but to monitor its performance in the course of training. Both training and validation samples were split roughly equally between signal waveforms and noise waveforms. Prior to feeding the waveforms to the model, the mean $\bar{x}$ and standard deviation $\sigma$ of all ADC values over all waveforms in the sample were first computed. Each

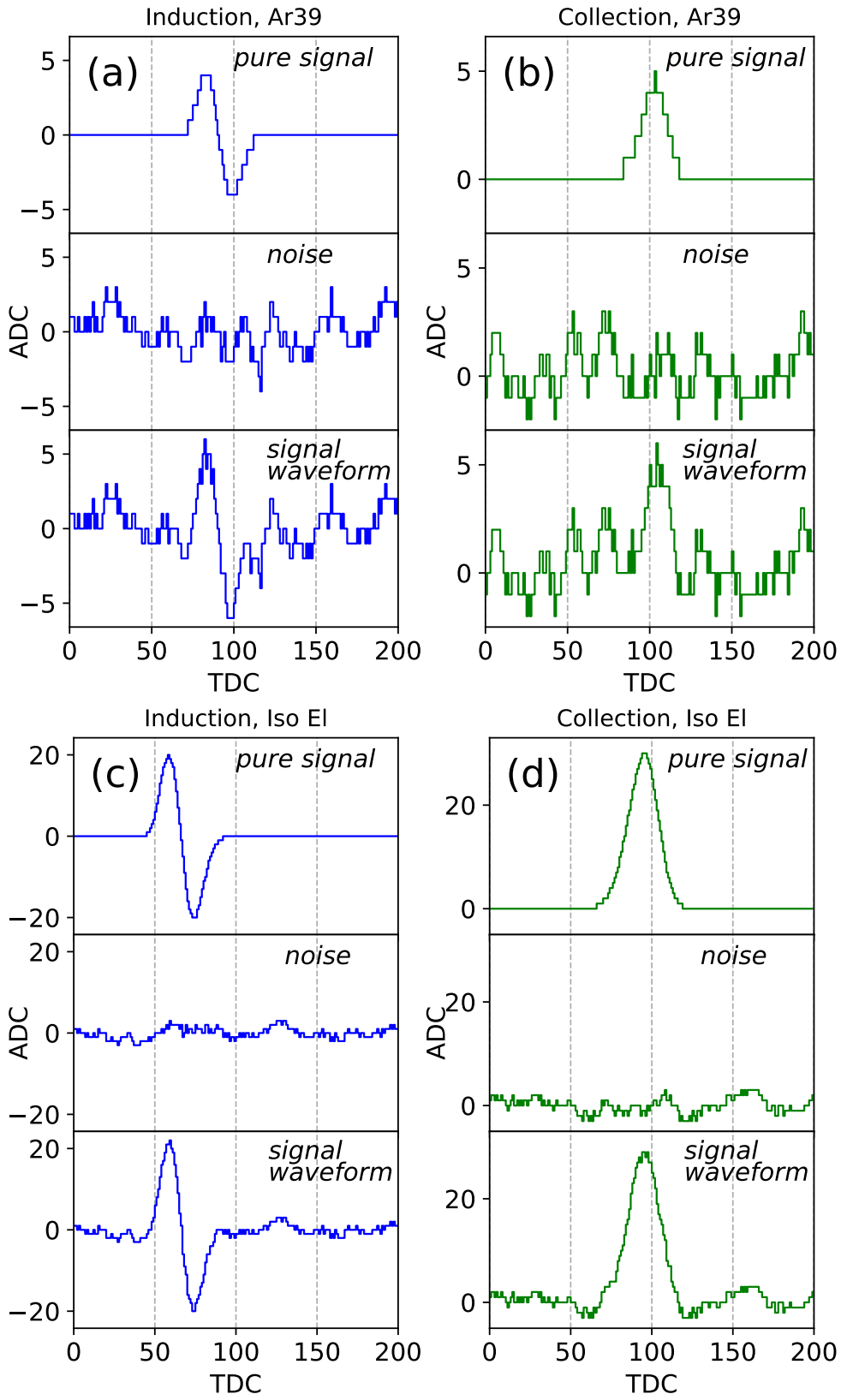



Figure 3: Plots (a) and (b) show simulated wire waveforms for the $^{39}$Ar sample in the induction and collection planes, respectively. Plots (c) and (d) show simulated wire waveforms for the isotropic electron sample in the induction and collection planes, respectively. The first row in each plot shows the pure signal waveform, followed by the noise waveform in the second row, and the sum of these two components in the third row. Details on the simulation of these waveforms is described in Section 2 of the text.

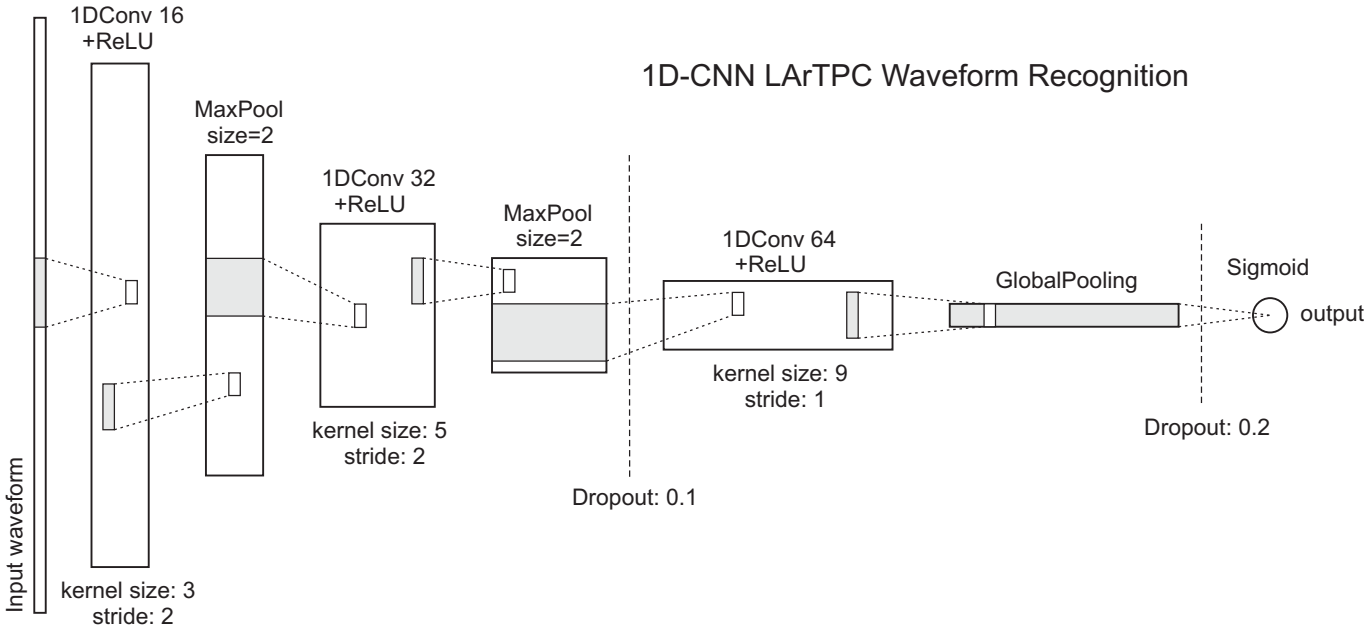



Figure 4: Architecture of the one-dimensional convolutional neural network used to recognize signals in LArTPC waveforms.

waveform, identified by the index $i$, was then *standardized* by scaling its ADC values $x_i$ by a factor $s_i = (x_i - \bar{x})/\sigma$. To fit for the optimal model weights in the training process, we made use of the *Adam* (adaptive moment estimation) optimization algorithm [21]. This is an extension of the mini-batch stochastic gradient method that uses per-parameter learning rates, whose values are adapted based on how quickly the weights have been changing. The weights were determined and updated iteratively using random batches of 2,048 waveforms (*batch size*), in which a full pass over the entire sample was completed in one *epoch*. A total of 8 (12) training epochs were performed for the induction (collection) plane. We used the binary cross-entropy loss function, calculated according to $L = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)$, where the index $i$ runs over the number of observations $N$. $p_i$ is the model output representing the predicted probability for observation $i$ to contain a signal, and $y_i$ is the correct label for that observation (1 for signal and 0 otherwise).

The evolution of the accuracy and the loss as a function of training epoch is shown separately in Figure 5 for the induction and collection planes. We define *accuracy* $= \frac{\mathrm{TP+TN}}{\mathrm{TP+FN+TN+FP}}$, where TP, TN, FP, and FN are the number of true positives, true negatives, false positives, and false negatives, respectively. Loss is as defined above. For each plane, the accuracy and loss curves are shown for both the training (solid) and validation (dashed) samples. In all cases, the validation curves follow the general trend of the training curves, reassuring us that overfitting is not an issue. The values for TN, FP, FN, and TP (elements of the $2 \times 2$ confusion matrix) at the end of the final training epoch are shown for the training and validation samples in the first two rows of Table 1.

### *3.3. Verifying with independent test set*

When the training and validation samples were created, a separate and independent test sample of $^{39}$Ar waveforms was also generated. This served as an unbiased sample that was not used to train the sample and did not influence the hyperparameter choices for the model. The size of this sample was ≈899 K (≈1.29 M) for the induction (collection) plane, consisting of roughly equal portions of signal and noise waveforms. We applied the fully trained model described in Section 3.2 to the test sample after scaling its ADC values with the same standardization parameters used for the training sample. The results, in terms of the elements of the confusion matrix, are shown in the last row of Table 1. From these numbers, we calculate *precision* $= p = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$, *negative predictive value* $= npv = \frac{\mathrm{TN}}{\mathrm{TN+FN}}$, *recall* $= r = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, *specificity* $= s = \frac{\mathrm{TN}}{\mathrm{TN+FP}}$, and accuracy as defined previously. The values for these metrics for the test sample are shown in Table 2. In Figure 6, the true positive rate is plotted against the false positive rate to show the receiver operating characteristic curves (ROCs) from applying the fully trained model on the test sample for each plane. The diagonal dash-dotted line in each plot represents the reference case when a model has absolutely no ability to tell two classes apart, in which case the area under the ROC curve (AUC) is 0.5. In comparison, our model is able to achieve AUC=0.97 on the induction plane and AUC=0.99 on the collection plane, indicating useful discriminating power between signal and noise waveforms.

## 4. Region of interest finding and model performance

### *4.1. Extending the model to localize signals*

The results presented in Section 3.3 demonstrate that our 1D-CNN model can make useful predictions about

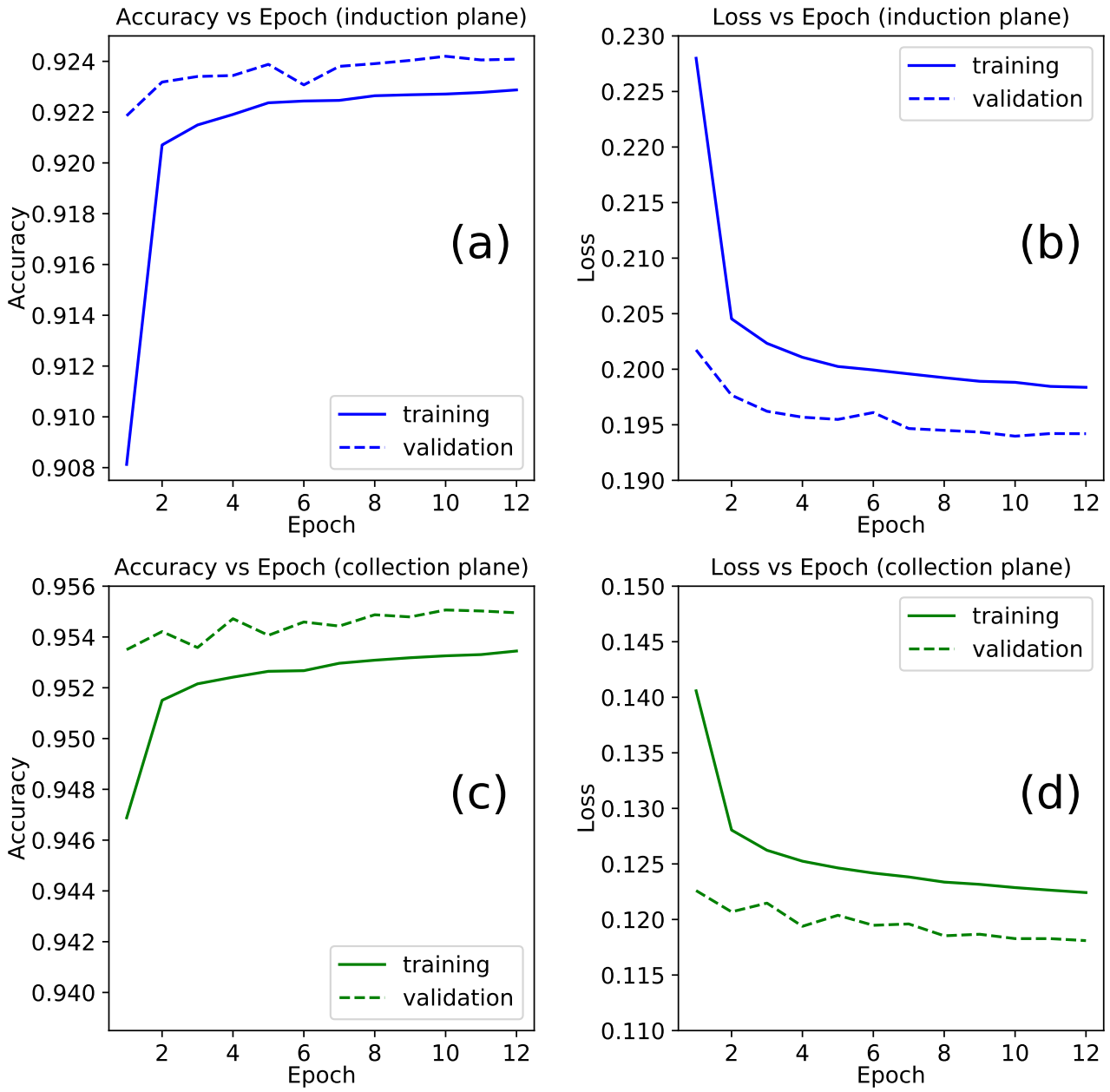



Figure 5: Plots (a) and (b) show how accuracy and loss evolve as a function of training epoch for the induction plane. Plots (c) and (d) shows the evolution of the same quantities as a function of training epoch for the collection plane.

| | Induction Plane | | | | Collection Plane | | | |
|---|---|---|---|---|---|---|---|---|
| Sample | TN | FP | FN | TP | TN | FP | FN | TP |
| training | 0.477 | 0.023 | 0.052 | 0.448 | 0.483 | 0.017 | 0.028 | 0.472 |
| validation | 0.476 | 0.024 | 0.052 | 0.448 | 0.482 | 0.017 | 0.028 | 0.473 |
| testing | 0.476 | 0.024 | 0.052 | 0.448 | 0.483 | 0.017 | 0.028 | 0.472 |

Table 1: The table above shows the elements of the confusion matrix when applying the fully trained 1D-CNN model on the training, validation, and testing samples. Results are shown for both planes.

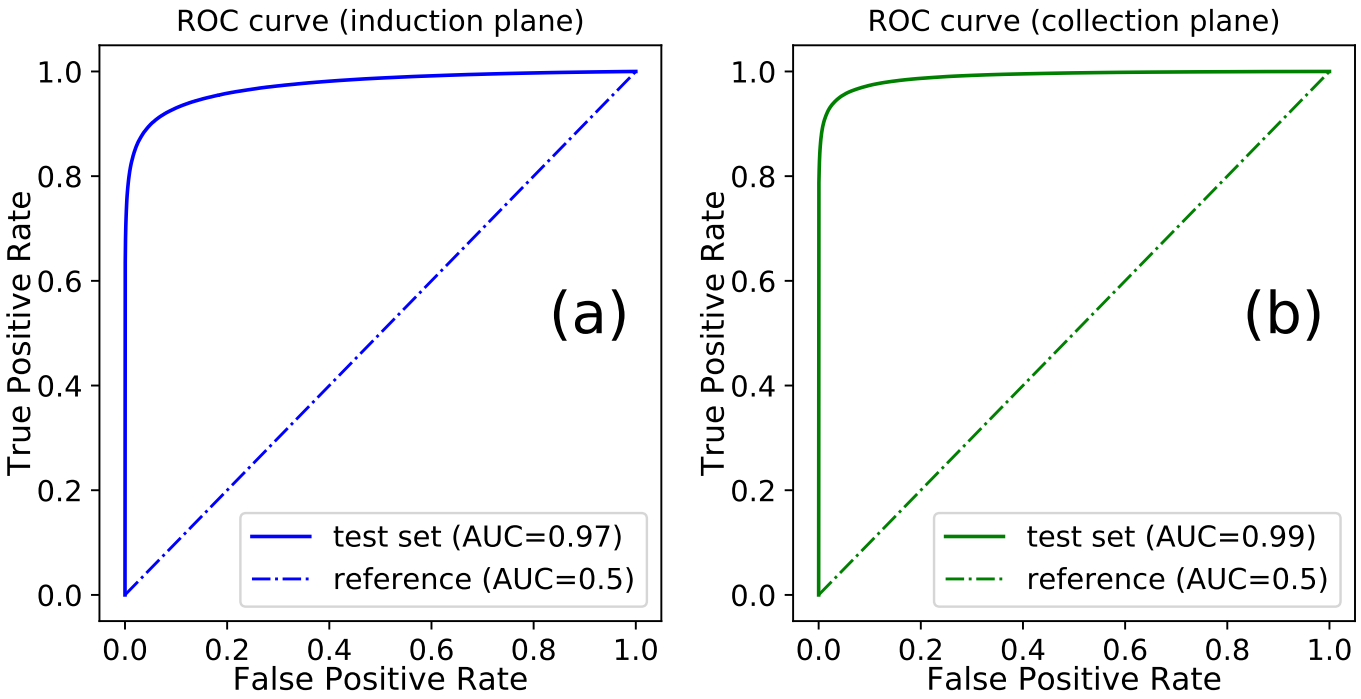



Figure 6: Receiver operating characteristic curves (ROCs) and the associated areas under the ROC curves (AUCs) for the independent test set are shown for the induction plane in (a) and the collection plane in (b). Shown for reference in each plot is the dash-dotted line representing the case when there is absolutely no discriminating power.

whether 200-tick waveform snippets contain signals or not. This section describes how we extended this capability to the localization of signals within the full waveforms in terms of ROIs. This was done in a straightforward manner by simply scanning a window across the entire waveform starting from the leftmost edge and shifting it repeatedly to the right with some finite stride length until it reached the rightmost edge. By performing an inference on the portion of the waveform contained within the window at each step, the signal region could effectively be localized. In our implementation in this paper, the full 2,048-tick simulated LArTPC waveform was subdivided into 14 overlapping 200-tick windows, beginning with the first, whose left edge was aligned with the start of the waveform. Each of the 12 subsequent windows after the first was offset from the previous one by a stride length of 150 ticks, while the last window (13th after the first) was offset from the preceding one by 48 ticks, so that its right edge aligned with the last tick of the full waveform. An inference was then performed on each of these 14 windows to identify the ROI/s within the full waveform. Examples of such ROIs are shown in Figure 7.

### *4.2. Evaluating efficiency with single electrons*

To further quantify the performance and capabilities of our 1D-CNN model, we generated a single electron sample with electron momenta ranging from 1 MeV to 1 GeV. The initial position of each electron was generated uniformly within a rectangular prism that centered on the LAr volume and which measures $30 \times 30 \times 90$ cm$^3$, with the long dimension oriented along the $z$ axis and the two shorter ones along the $x$ and $y$ axes. The initial direction of each electron was generated pseudo-isotropically with angles $\theta_{xz}$ and $\varphi_{yz}$ distributed uniformly within $\pm 180°$ and $\pm 90°$, respectively. After this, the electrons were propagated through the detector using GEANT4 and the detector response simulated as described in Section 2, to produce the raw LArTPC wire waveforms. The fully trained 1D-CNN, described in Section 3.2, was then applied to each waveform to find the signal ROI as detailed in Section 4.1.

| Metric | Induction Plane | Collection Plane |
|---|---|---|
| precision | 0.950 | 0.965 |
| negative predictive value | 0.901 | 0.946 |
| recall | 0.896 | 0.944 |
| specificity | 0.953 | 0.966 |
| accuracy | 0.924 | 0.955 |

Table 2: The table above summarizes the performance of the fully trained 1D-CNN model on the test sample.

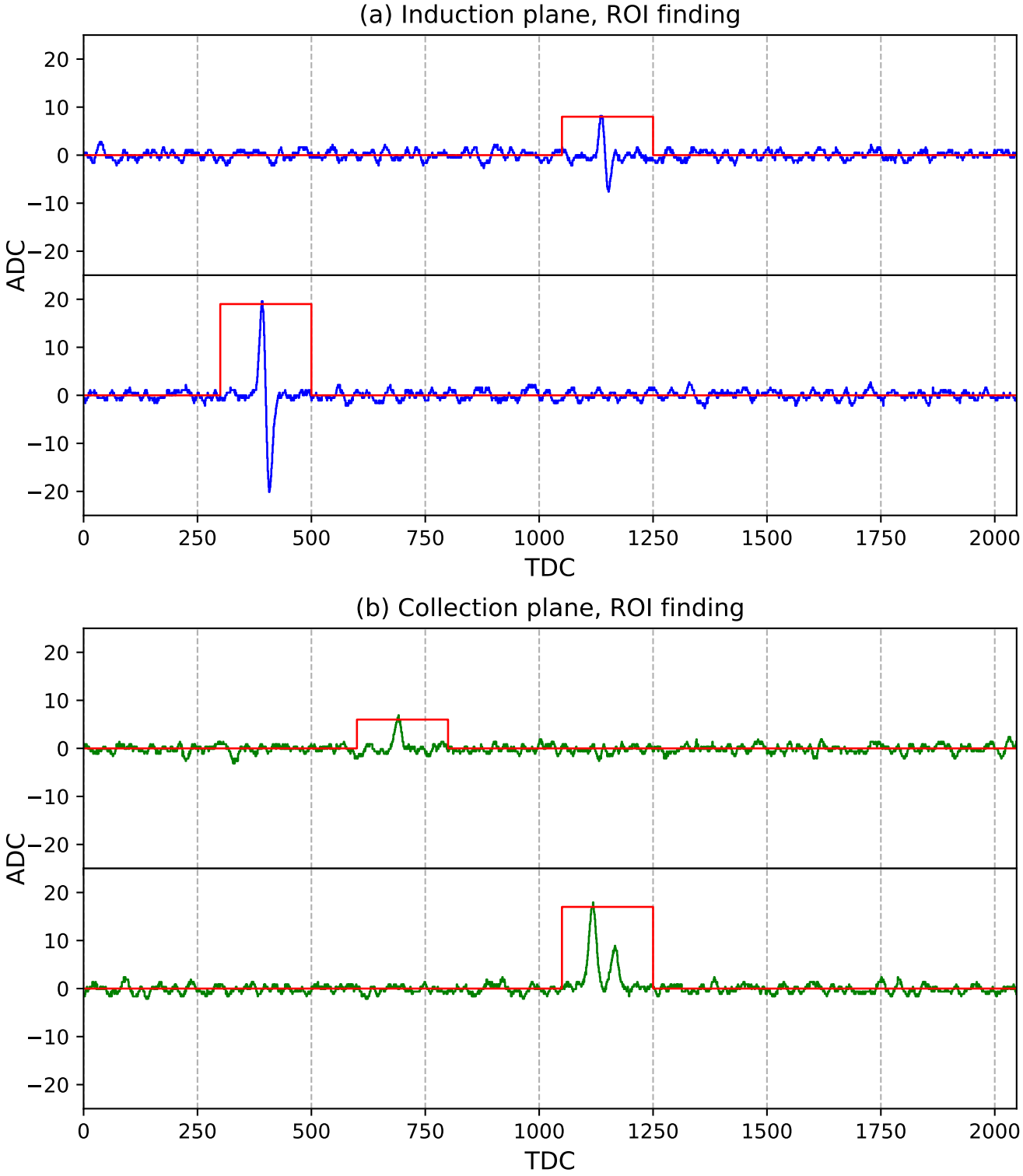



Figure 7: Two examples each of simulated full (2,048-tick) waveforms are shown for the (a) induction and the (b) collection planes. *Regions-of-interest* (ROIs) that localize the signals within the full waveforms, using the method described in Section 4.1, are indicated by the rectangular pulses.

The signal detection efficiency for the isotropic electron sample is presented here as a function of the peak ADC value associated with the pure signal component of a digitized waveform, which shall be referred to hereon as $ADC_{\rm pk}$. We choose to work with ADC values because they are ultimately what the 1D-CNN directly "sees". This avoids the need to convert from some other quantity like energy to ADC values, which would be experiment specific.

To calculate efficiency, two histograms are created to represent the numerator ($H_{\rm N}$) and denominator ($H_{\rm D}$) of the ratio. Each histogram has 12 bins representing $ADC_{\rm pk}$ ranging from 1 to 12. If a signal is present within the full 2,048-tick waveform, an entry is made in the $H_{\rm D}$ bin corresponding to the $ADC_{\rm pk}$ associated with the signal. In case multiple signals are present within the full waveform, only the one with the largest $ADC_{\rm pk}$ is selected for calculating efficiency. Following the procedure described in Section 4.1, the 1D-CNN is used to scan the entire waveform for ROIs. If the signal lies within an ROI identified by the 1D-CNN, an entry is also made in the same bin of $H_{\rm N}$. Once the two histograms are filled, their ratio is taken to yield the efficiency as a function of $ADC_{\rm pk}$. This is done separately for the induction and collection planes, and the results are represented in Figure 8 by the triangular markers. The average energy deposited and average number of ionization electrons associated with the signal, for each bin of $ADC_{\rm pk}$, are also shown below each plot. The CNN output is required to be $> 0.91$ ($> 0.95$) in order to classify a waveform as a signal in the induction (collection) plane.

Also shown for comparison in each plot are the efficiencies for the traditional *over-ADC-threshold* approach, represented by the circular markers. In this case, signal waveforms are discriminated from noise by requiring the measured ADC values in a region containing the signal to be above noise threshold. The mean ADC value of the pure noise waveforms generated as described in Section 2 is 1.77, with a standard deviation of 1.05. In these plots, we require $|\text{ADC}| > 6$ counts for the induction plane and $\text{ADC} > 6$ counts for the collection plane. These correspond to $\approx 4$ standard deviations above the noise average. Such a requirement achieves a background rejection rate of 0.926 (0.962) on the induction (collection) plane, which closely matches the rate of 0.931 (0.966) from the CNN-based method when requiring the output to be $> 0.91$ ($> 0.95$).

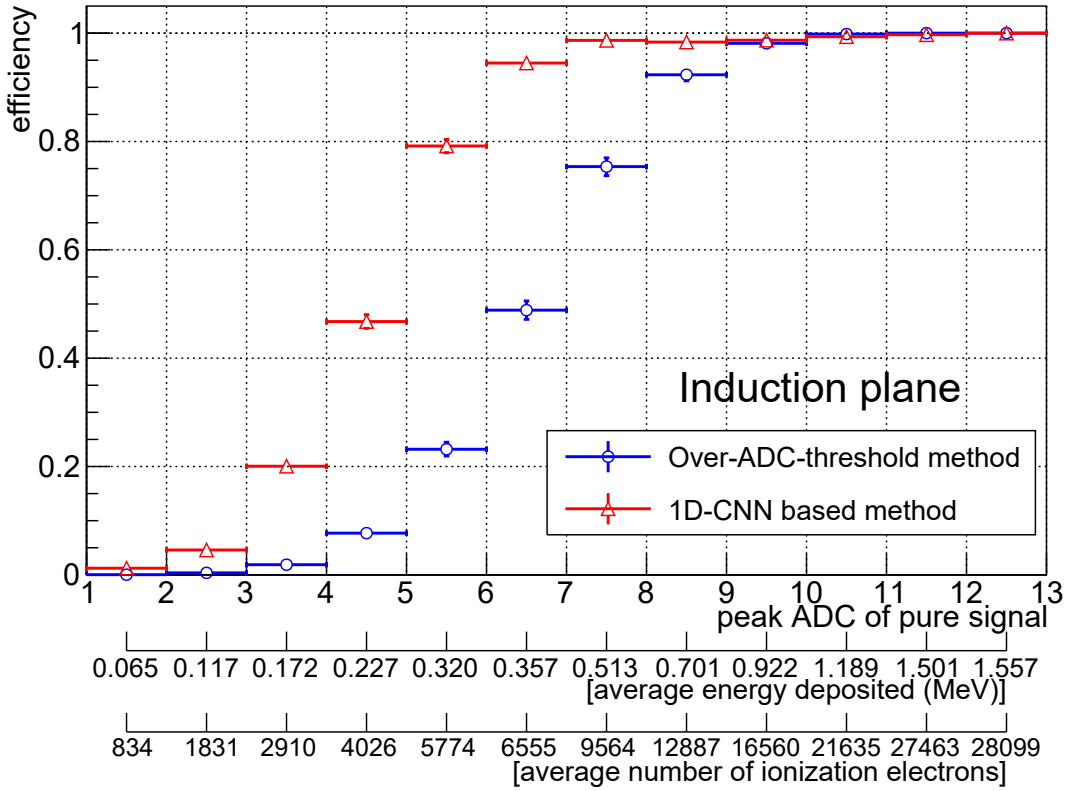
The fact that the efficiency plots for the *over-ADC-threshold* method do not exhibit a sharp cutoff at $ADC_{\rm pk} = 6$ might seem counter-intuitive at first. However, this is to be expected because $ADC_{\rm pk}$ represents the ADC value associated with the pure signal prior to any noise fluctuations. A signal with $ADC_{\rm pk} \leq 6$ could satisfy the requirement of $\text{ADC} > 6$ if it fluctuates above this threshold after the addition of noise. To demonstrate more clearly how the 1D-CNN is able to detect signals in regions inaccessible to the traditional approach, the 1D-CNN efficiencies are shown as a function of the actual ADC value produced by the detector ($ADC_{\rm pk}$ + noise) in Figure 9. The hatched region represents the region above the ADC cut that is accessible to the *over-ADC-threshold* method.

It is not difficult to see from Figure 8 that the CNN-based method outperforms the traditional approach throughout the range from $ADC_{\rm pk} = 2$ to 8, after which both achieve essentially 100% efficiency. The former begins to achieve better than 90% efficiency at $ADC_{\rm pk} = 6$, while the latter only reaches this level at around $ADC_{\rm pk} = 8$.

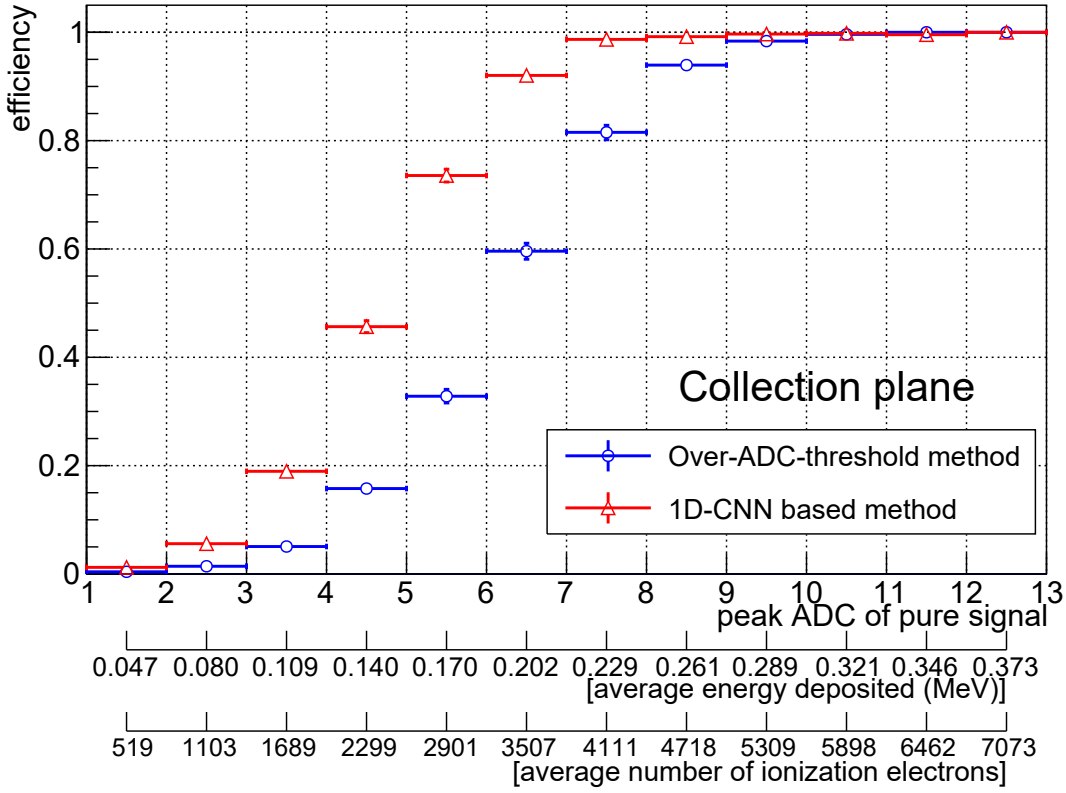
## 5. Conclusion

In this work, we have successfully demonstrated that deep learning methods can be applied directly to the raw wire waveforms produced by the individual channels in a LArTPC detector in order to discriminate signal waveforms from background. This was achieved using a 1D-CNN which was implemented in a way that allowed the ROI of the signal to be identified within the full waveform. The discriminating power of CNNs derives from their ability to learn and detect subtle features that distinguish signal from background such as, but not limited to, shape characteristics. Because of this, they are not constrained by user defined cuts imposed in traditional threshold based signal ROI finders and can maintain signal sensitivity in energy regions inaccessible to such methods. The implications are significant for the rich low-energy physics program of future neutrino experiments like DUNE.

A major advantage of our approach over other ML-based ones, is its use of a simple neural network architecture consisting of a mere ≈20K trainable parameters compared with the millions found on typical designs. To get a rough idea of resource utilization and performance on an FPGA, we did a quick test using `hls4ml` [22] without investing effort in optmization, to translate our 1DCNN model into a synthesized IP core. With 16 bit precision, this quick and dirty implementation consumed less than half of the resources on a mid-range FPGA (e.g. Xilinx KU15P [23] used in the SmartSSD [24]) while achieving maximum latencies of ≈20 $\mu$s per inference on a 200-tick time window, which is half the time to read out the waveform in this window for our toy detector simulation. With further optimizations, it would be worthwhile to investigate the feasibility of implementing this in the online DAQ systems of future LArTPC based experiments such as on DUNE's DAQ front-end computers (FECs) [25] to perform filtering operations like zero suppression on raw wire data from the detector. One idea is to realize an intermediate-level trigger by implementing this on the FPGA of a computational storage drive [24] installed on an FEC to process buffered Supernova candidates awaiting transfer to the surface and back to Fermilab. This approach can help achieve optimal signal efficiency and background rejection that can benefit and complement downstream selection algorithms, including more complex ML-based ones. While the feasibility of such applications is beyond the scope of

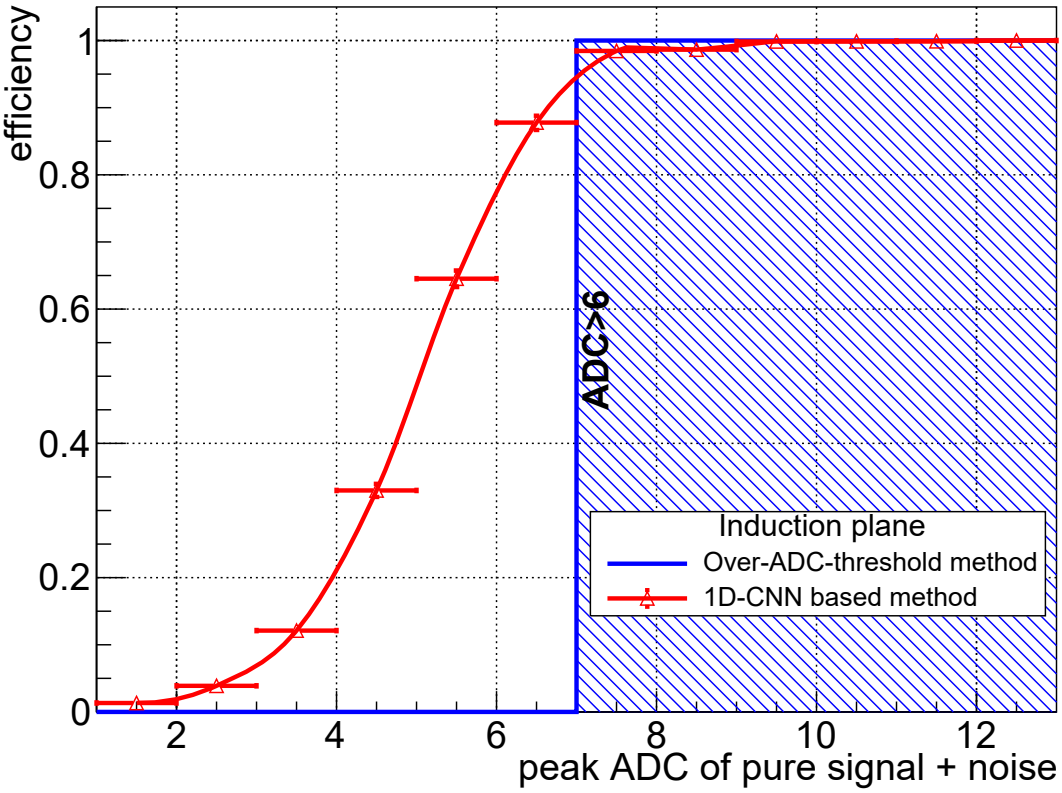



(a) Efficiencies for induction plane

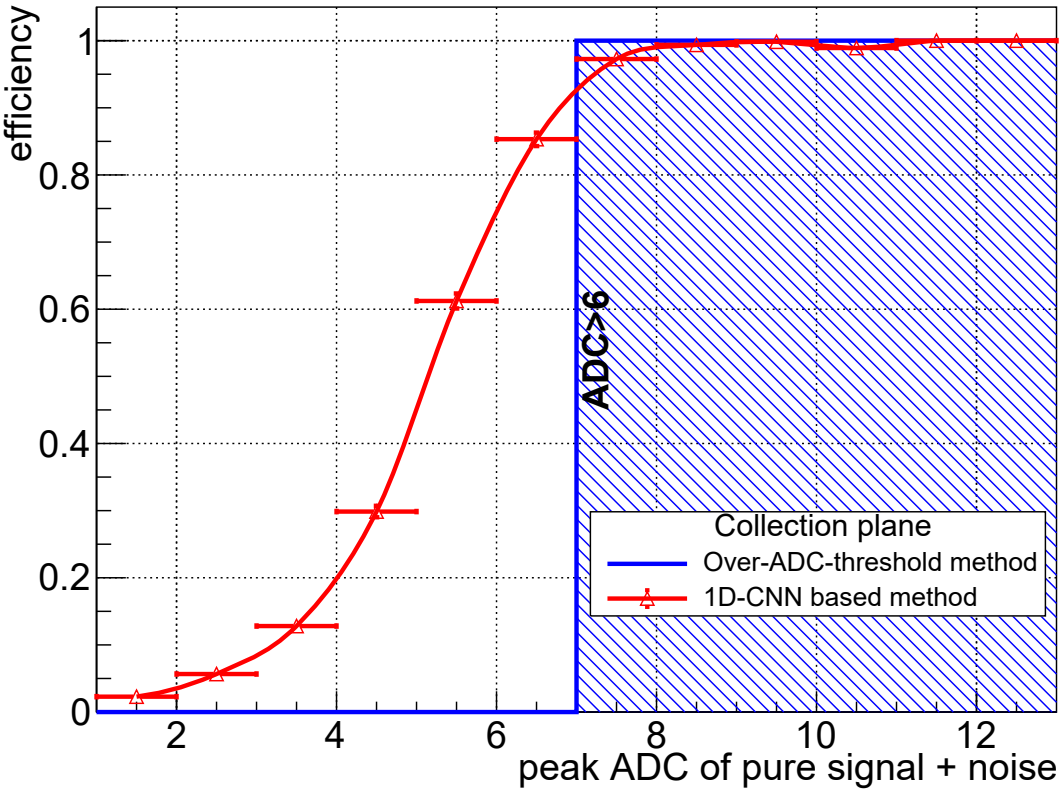



(b) Efficiencies for collection plane

Figure 8: Efficiencies of the 1D-CNN ROI finder as a function of $ADC_{\mathrm{pk}}$ determined using the isotropic electron sample for the (a) induction and (b) collection planes. Results are shown in each case for the traditional *over-ADC-threshold* and the 1D-CNN based approaches. The average energy deposited in MeV and the average number of electrons associated with the signal, for each $ADC_{\mathrm{pk}}$ bin, are shown below each plot.




(a) Efficiencies for induction plane




(b) Efficiencies for collection plane

Figure 9: Efficiencies of the 1D-CNN ROI finder for the (a) induction and (b) collection planes as a function of the actual ADC value produced by the detector, which is the sum $ADC_{\mathrm{pk}}$ + noise. The hatched region represents the efficiency of the *over-ADC-threshold* method.

this paper, we are currently investigating them and hope to report our findings in the future.

This paper set out to establish a proof-of-concept and provide a detailed description of the methodology, and it has succeeded in this task. In subsequent and related papers, this method will be used in actual applications, including those involving real experimental data [19].

## 6. Acknowledgements

We wish to thank Brian Nord and Gabe Perdue of Fermilab for useful discussions on machine learning techniques, and Brian Rebel of the University of Wisconsin for clarifying simulation details in *LArSoft*. We also wish to thank Jovan Mitrevski of Fermilab for his help in estimating the resource usage and performance of the 1DCNN on an FPGA. This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics. This work has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement number 858199, "INTENSE".

## References

[1] S. Jones, et al., Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume 1 Introduction to DUNE (2020). arXiv:2002.02967.

[2] J. N. Marx, D. R. Nygren, The Time Projection Chamber, Phys. Today 31N10 (1978) 46–53. doi:10.1063/1.2994775.

[3] C. Rubbia, The Liquid Argon Time Projection Chamber: A New Concept for Neutrino Detectors (5 1977).

[4] X. Qian, P. Vogel, Neutrino mass hierarchy, Progress in Particle and Nuclear Physics 83 (2015) 1–30. doi:https://doi.org/10.1016/j.ppnp.2015.05.002.
URL https://www.sciencedirect.com/science/article/pii/S0146641015000307

[5] H. Nunokawa, S. Parke, J. W. Valle, CP violation and neutrino oscillations, Progress in Particle and Nuclear Physics 60 (2) (2008) 338–402. doi:https://doi.org/10.1016/j.ppnp.2007.10.001.
URL https://www.sciencedirect.com/science/article/pii/S014664100700083X

[6] B. Abi, et al., Supernova neutrino burst detection with the Deep Underground Neutrino Experiment, The European Physical Journal C 81 (5) (May 2021). doi:10.1140/epjc/s10052-021-09166-w.
URL http://dx.doi.org/10.1140/epjc/s10052-021-09166-w

[7] V. A. Kudryavtsev, et al., Underground physics with DUNE, Journal of Physics: Conference Series 718 (6) (2016) 062032. doi:10.1088/1742-6596/718/6/062032.
URL https://doi.org/10.1088/1742-6596/718/6/062032

[8] F. Capozzi, S. W. Li, G. Zhu, J. F. Beacom, DUNE as the Next-Generation Solar Neutrino Experiment, Phys. Rev. Lett. 123 (2019) 131803. doi:10.1103/PhysRevLett.123.131803.
URL https://link.aps.org/doi/10.1103/PhysRevLett.123.131803

[9] S. Agostinelli, et al., Geant4–a simulation toolkit, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 506 (3) (2003) 250–303. doi:https://doi.org/10.1016/S0168-9002(03)01368-8.
URL https://www.sciencedirect.com/science/article/pii/S0168900203013688

[10] J. Allison, et al., Recent developments in Geant4, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 835 (2016) 186–225. doi:https://doi.org/10.1016/j.nima.2016.06.125.
URL https://www.sciencedirect.com/science/article/pii/S0168900216306957

[11] E. Snider, G. Petrillo, LArSoft: Toolkit for Simulation, Reconstruction and Analysis of Liquid Argon TPC Neutrino Detectors, J. Phys. Conf. Ser. 898 (4) (2017) 042057. doi:10.1088/1742-6596/898/4/042057.

[12] M. Miyajima, T. Takahashi, S. Konno, T. Hamada, S. Kubota, H. Shibamura, T. Doke, Average energy expended per ion pair in liquid argon, Phys. Rev. A 9 (1974) 1438–1443. doi:10.1103/PhysRevA.9.1438.

[13] J. Thomas, D. A. Imel, Recombination of electron-ion pairs in liquid argon and liquid xenon, Phys. Rev. A 36 (1987) 614–616. doi:10.1103/PhysRevA.36.614.
URL https://link.aps.org/doi/10.1103/PhysRevA.36.614

[14] R. Acciarri, A study of electron recombination using highly ionizing particles in the ArgoNeuT liquid argon TPC, Journal of Instrumentation 8 (08) (2013) P08005–P08005. doi:10.1088/1748-0221/8/08/p08005.
URL https://doi.org/10.1088/1748-0221/8/08/p08005

[15] This cluster size corresponds roughly to the number of ionization electrons produced by a minimum ionizing particle in liquid argon in 0.01 cm, which corresponds to the maximum step size used in the GEANT4 simulation. Actual step sizes taken in the simulation are typically an order of magnitude smaller than this maximum, resulting in a single cluster produced per step.

[16] B. Baller, Liquid argon TPC signal formation, signal processing and reconstruction techniques 12 (07) (2017) P07010–P07010. doi:10.1088/1748-0221/12/07/p07010.
URL https://doi.org/10.1088/1748-0221/12/07/p07010

[17] C. Adams, et al., Ionization electron signal processing in single phase LArTPCs. part i. algorithm description and quantitative evaluation with MicroBooNE simulation 13 (07) (2018) P07006–P07006. doi:10.1088/1748-0221/13/07/p07006.
URL https://doi.org/10.1088/1748-0221/13/07/p07006

[18] C. Adams, et al., Ionization electron signal processing in single phase LArTPCs. part II. data/simulation comparison and performance in MicroBooNE, Journal of Instrumentation 13 (07) (2018) P07007–P07007. doi:10.1088/1748-0221/13/07/p07007.
URL https://doi.org/10.1088/1748-0221/13/07/p07007

[19] R. Acciarri, B. Baller, V. Basque, C. Bromberg, F. Cavanna, D. Edmunds, R. Fitzpatrick, B. Fleming, P. Green, C. James, I. Lepetic, X. Luo, O. Palamara, G. Scanavini, M. Soderberg, J. Spitz, A. Szelc, L. Uboldi, M. Wang, W. Wu, T. Yang, A deep-learning based raw waveform region-of-interest finder for the liquid argon time projection chamber, Journal of Instrumentation 17 (01) (2022) P01018. doi:10.1088/1748-0221/17/01/p01018.
URL https://doi.org/10.1088/1748-0221/17/01/p01018

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research 15 (2014) 1929–1958.
URL http://jmlr.org/papers/v15/srivastava14a.html

[21] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.
URL http://arxiv.org/abs/1412.6980

[22] J. Duarte, S. Han, P. Harris, S. Jindariani, E. Kreinar, B. Kreis, J. Ngadiuba, M. Pierini, R. Rivera, N. Tran, Z. Wu, Fast inference of deep neural networks in FPGAs for particle physics 13 (07) (2018) P07027–P07027. doi:10.1088/1748-0221/13/07/p07027.
URL https://doi.org/10.1088/1748-0221/13/07/p07027

[23] Xilinx, UltraScale architecture and product data sheet:

overview (2021).
URL https://www.xilinx.com/support/documentation/data_sheets/ds890-ultrascale-overview.pdf

[24] Xilinx, SmartSSD computational storage drive: installation and user guide (2021).
URL https://www.xilinx.com/support/documentation/boards_and_kits/accelerator-cards/1_2/ug1382-smartssd-csd.pdf

[25] B. Abi, et al., Volume IV. the DUNE far detector single-phase technology 15 (08) (2020) T08010–T08010. doi:10.1088/1748-0221/15/08/t08010.
URL https://doi.org/10.1088/1748-0221/15/08/t08010