Machine learning studies for improved electron rejection in pixel TPCs

Jeff Schueler*, Majd Ghrear, Sven Vahsen

01/11/2022

*Email: jschuel@hawaii.edu

Overview

1. Motivation and previous electron rejection simulation studies
2. Neural network multivariate study of electron rejection
3. Boosted decision tree multivariate study of electron rejection
4. Deep learning electron rejection study with convolutional neural network
5. Overview of ongoing work with BEAST TPCs

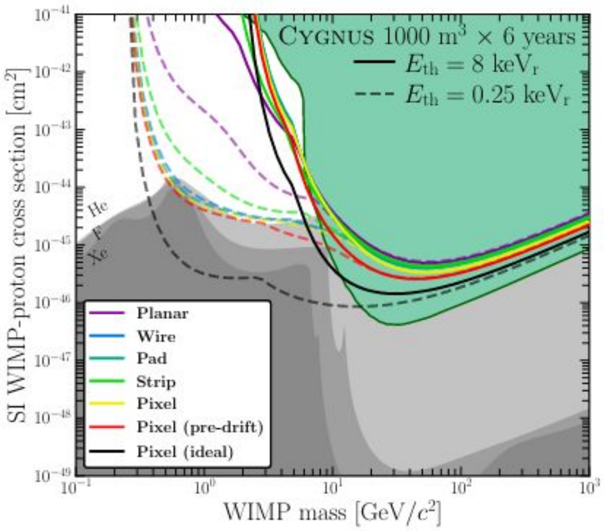
Electron background rejection

From CYGNUS paper: *“**Electron backgrounds** (i.e. gamma-recoils) are a key issue for Cygnus in that they **will effectively determine the energy threshold**.”*

❖ Lowering the energy threshold of a WIMP detector by improving electron rejection performance increases WIMP detection sensitivity

❖ TPCs with high readout segmentation allow for recoil images with many features that can be used for background discrimination -> ideal for deep learning

arXiv:2008.12587
Solid lines = 8.0 keVr threshold
Dashed lines = 0.25 keVr threshold



We aim to improve electron rejection using machine learning.

Simulated samples

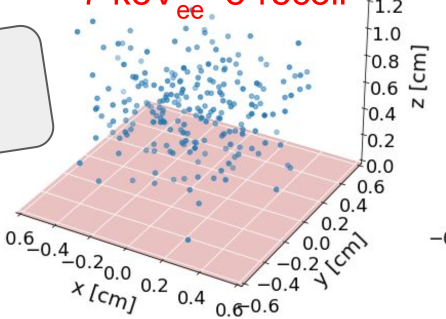
Sample of recoils following the same simulation procedure published in Ghrear et. al. (2021). Some highlights:

- >12 million events
- Optimized 80:10:10 mixture He:CF_4:CHF_3 at 60 torr and 25°C
- 40.6 V/cm drift field -> $(100\mu m)^3$ segmentation
- 25 cm of drift with longitudinal and transverse diffusion of 425μm/√cm and ~400μm/√cm, respectively
- No gain amplification and every electron is recorded

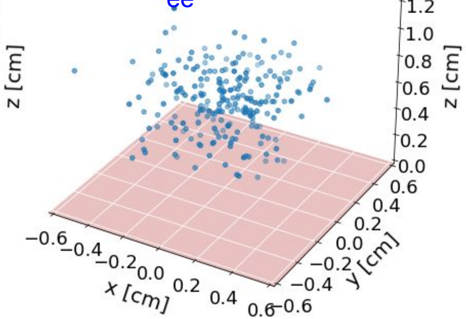
E [keV_{ee}]	# e recoils	# F recoils
0.5-1.5	2.05e6	26,450
1.5-2.5	2.02e6	27,859
2.5-3.5	1.50e6	25,139
3.5-4.5	1.00e6	23,059
4.5-5.5	1.00e6	21,772
5.5-6.5	1.00e6	21,590
6.5-7.5	1.00e6	20,901
7.5-8.5	1.00e6	20,212
8.5-9.5	1.00e6	19,414
9.5-10.5	1.00e6	18,880

Only classifying e and F recoils today!

7 keV_{ee} e-recoil



7 keV_{ee} F-recoil



The recoils events above would be misclassified using length alone, so we would benefit from other variables to reliably classify such events.

Electron rejection discriminants (Ghrear et. al. (2021))

- 7 *unique* discriminants used for distinguishing F recoils versus e recoils
  a. **Length** (along principal axis (PA))
  b. **Standard deviation of charge distribution** (SDCD) – $SDCD = \sqrt{\frac{\sum_{i=1}^{N}(\mathbf{r_i} - \bar{\mathbf{r}})^2}{N}}$
  c. **Number of clusters** (NumClust)
  d. **Clustering threshold** (ClustThres) - minimum number of pixels in most populated cluster
     - Both (c) and (d) use DBSCAN and have free parameters that are optimized for maximizing electron rejection
  e. **Maximum charge density** (MaxDen) in binned charge distribution with bin sizes optimized for e rejection
  f. **Cylindrical thickness** (CylThick) - Sum of each charges squared transverse distance from PA
  g. **Charge uniformity** (ChargeUnif) - Standard deviation of the mean distance between each point and all other points
- ClustThres and MaxDen are each optimized separately for directional and non directional cases → **9 total observables**

We will feed events with computed values of each of these 9 observables into (1) an artificial neural network, and (2) a boosted decision tree and use these as multivariate classifiers of recoil species

Performance of individual observables (Majd Ghrear)

Electron rejection factor, R, is the ratio of total electrons in a given bin to the number of remaining electrons after selections:

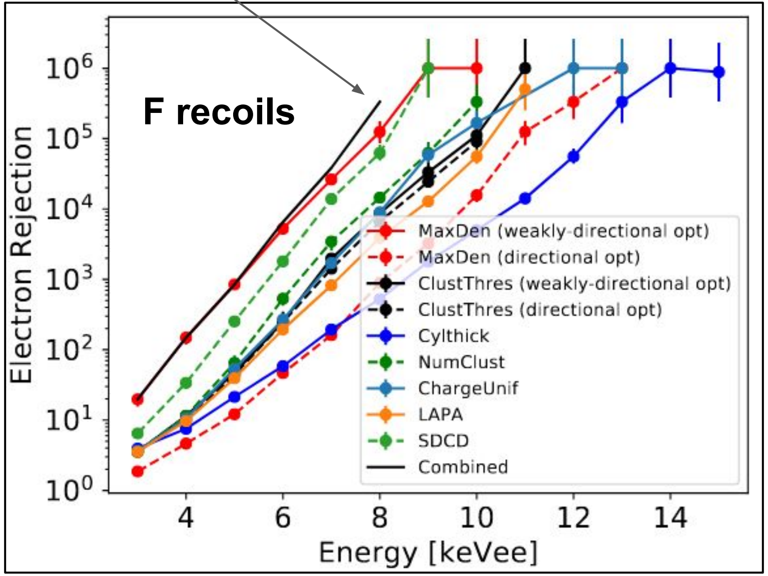
$$R \equiv \frac{N_{\mathrm{e,total}}}{N_{\mathrm{e,kept}}}$$

Nuclear recoil efficiency, ϵ, is the fraction of nuclear recoils remaining after selections:

$$\epsilon \equiv \frac{N_{\mathrm{F,kept}}}{N_{\mathrm{F,total}}}$$

Best individual observables reject all electrons above 8 $\mathrm{keV_{ee}}$

$\epsilon = 0.5$



All but two observables outperform length (LAPA), some by up to two orders of magnitude! The solid black combined observable line will be the baseline performance mark for our multivariate studies

Combining observables with **neural network (NN)**

1. Compute each of the 9 observables for every event
2. Store each event as a (10 x 1) vector containing the 9 observable values and the energy bin of the event
3. Shuffle the data and split into three samples
   a. **Training sample:** 3,046,286 events
   b. **Validation sample:** 507,714 events
   c. **Test sample:** 9,240,402 events
4. **Training loop over entire training set in random order**
5. After each training loop, evaluate the network on every event in the validation sample
   a. If loss is less than previous training epoch, save weight vectors
6. Repeat steps 4 and 5 until model is adequately trained
7. Deploy network on test sample

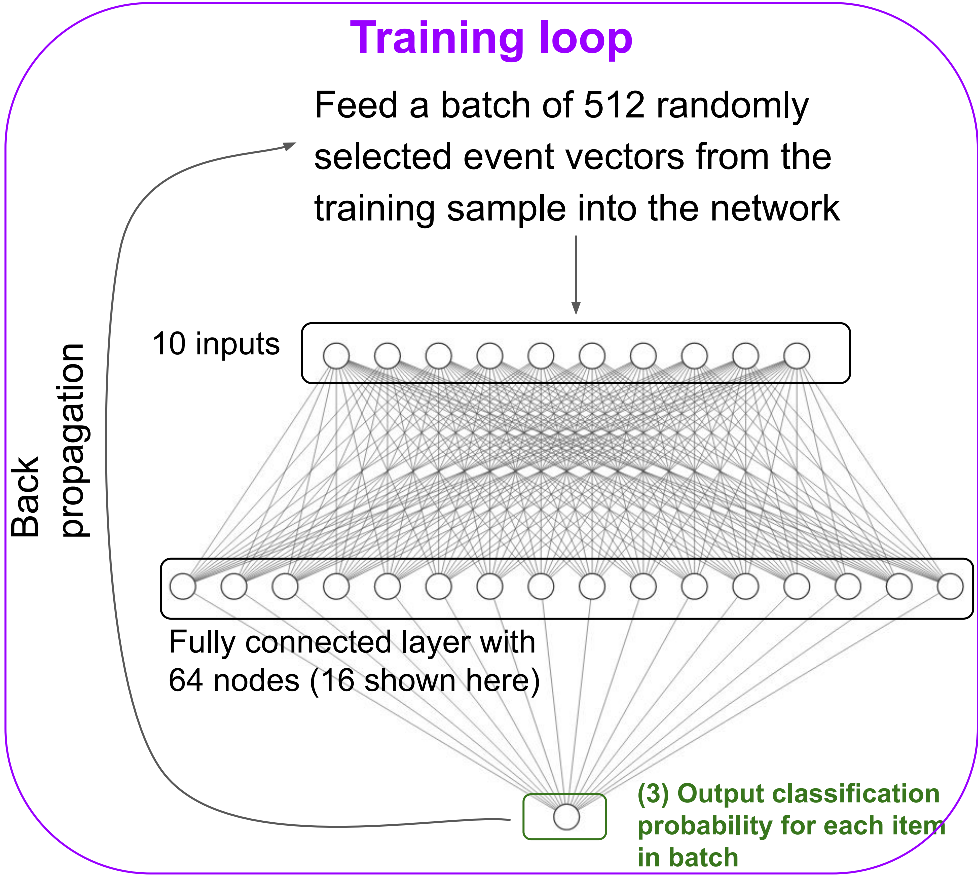


Image generated using: http://alexlenail.me/NN-SVG/index.html

Determining R from NN classifier output

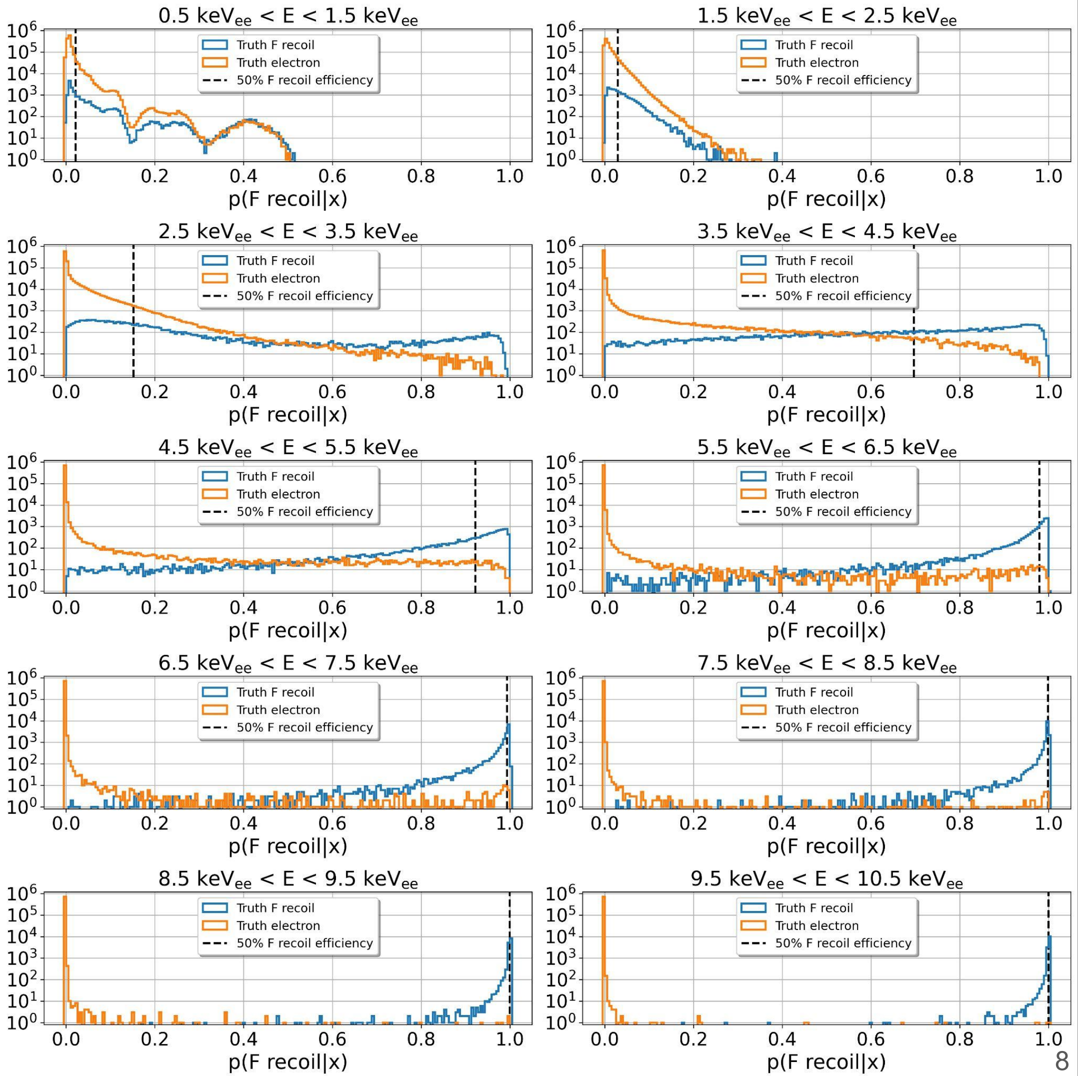
1. Find NN output probability, p_ϵ (black dashed line) corresponding to desired recoil efficiency

$$\epsilon \equiv \frac{N_{\mathrm{F,kept}}}{N_{\mathrm{F,total}}}$$

2. Keep all events where $p(\text{F recoil} \mid x) > p_\epsilon$ and compute

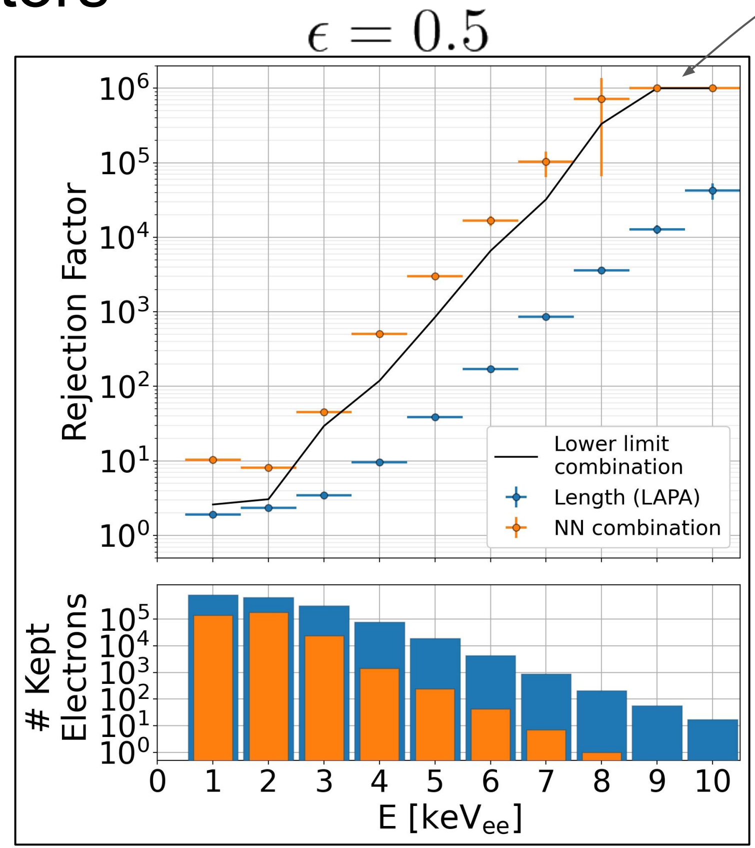
$$R \equiv \frac{N_{\mathrm{e,total}}}{N_{\mathrm{e,kept}}}$$

for this sample of events



NN rejection factors

We set R = 10^6 when all electrons are rejected in an energy bin!



Rejection factor

$$R \equiv \frac{N_{\mathrm{e,total}}}{N_{\mathrm{e,kept}}}$$

- NN achieved best performance after 46 training loops (epochs)
  - Trained through 100 loops
- Further adjustments to network architecture and hyperparameters may lead to improved performance but:
  - High computational cost
  - NN already outperforms other methods so far
  - At the point of diminishing returns?

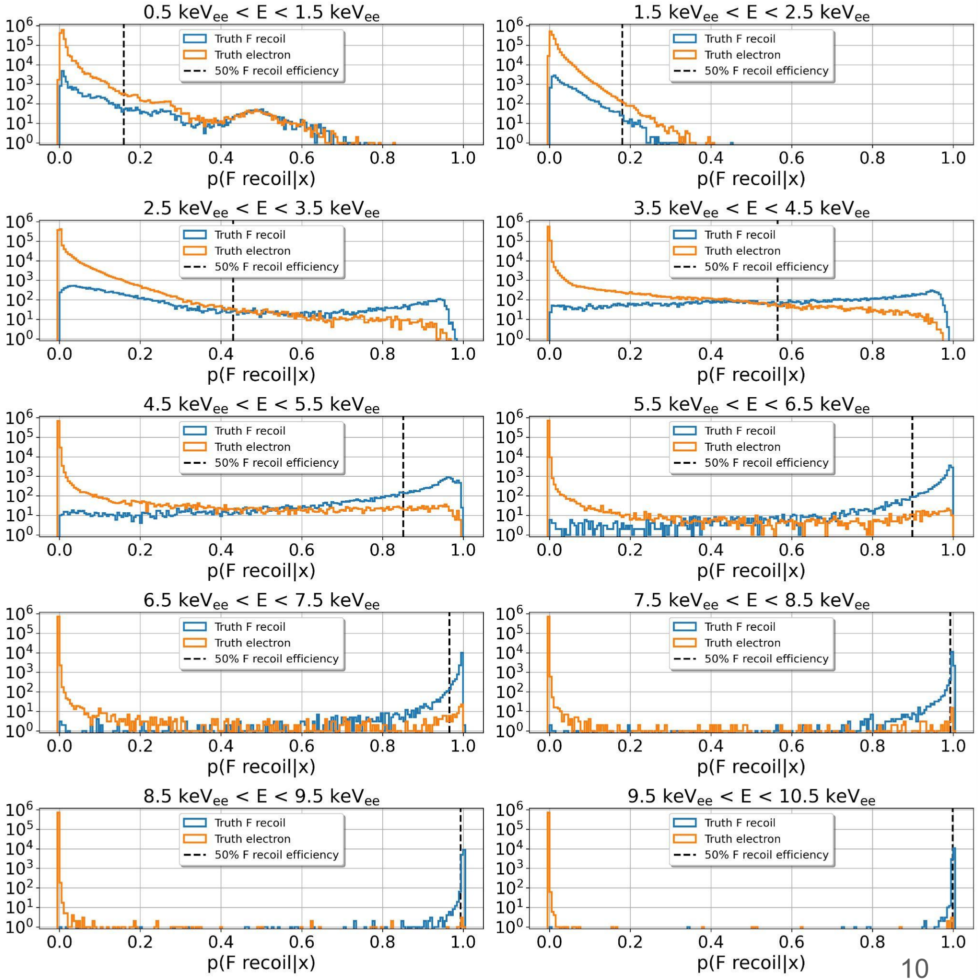
NN classifier outperforms track length (LAPA) and improves electron rejection performance of the 9 physically motivated observables

Combining observables with a **boosted decision tree (BDT)**

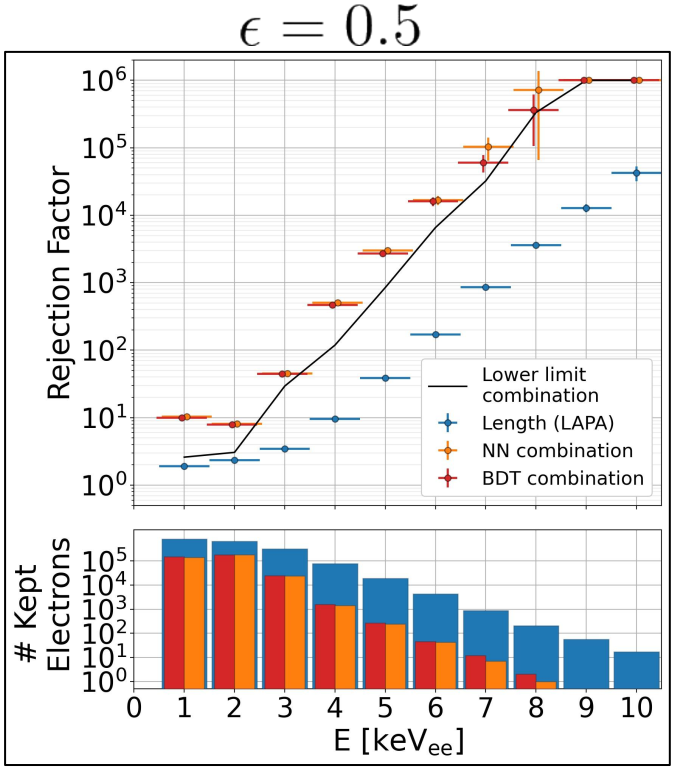
- ❖ Use identical samples to those used in the NN combination

Training sample: 3,046,286 events
Validation sample: 507,714 events
Test sample: 9,240,402 events

- ❖ Feed XGBoost algorithm the same (10 x 1) input vectors as the neural network
- ❖ Some advantages to XGBoost:
  - ➢ Can be trained on all 3e6 events in < 5 mins
  - ➢ Lots of room to optimize parameters
  - ➢ Relative performance of input parameters can be tracked



BDT rejection factors



Rejection factor

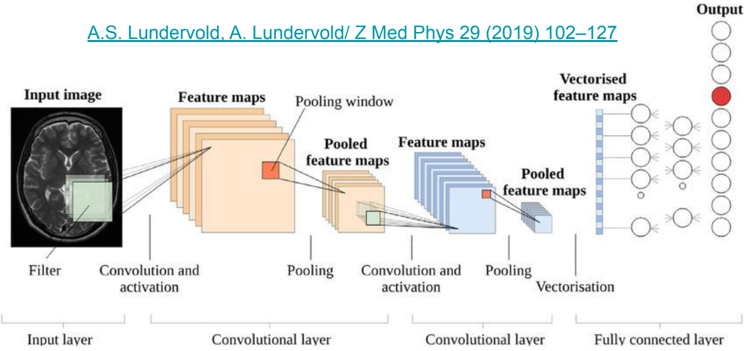
$$R \equiv \frac{N_{\mathrm{e,total}}}{N_{\mathrm{e,kept}}}$$

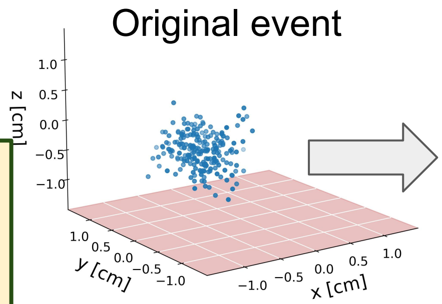
BDT classifier performs comparably but slightly worse than NN classifier

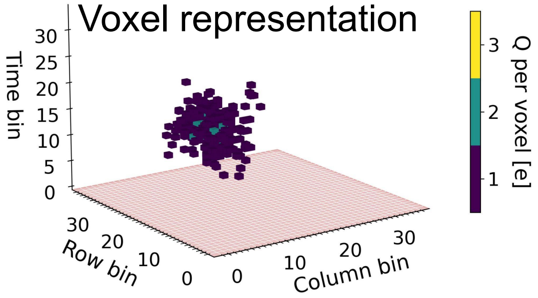
Deep learning using a **convolutional neural network (CNN)**

- Convolutional neural networks (CNNs) are tailor made for feature detection in images and can be used for 3D *segmented* events that would be obtained in a pixel readout detector
- Charge distributions must be binned into voxels
  - Simulated detector can read charge over a $25^3 cm^3$ volume and is "binned" into $100 \mu m^3$ segments => 2500 x 2500 x 2500 voxel grid which is not feasible
    - Each event stored this way would have 15.6GB of mostly 0's
- We instead "fiducialize" within [-1.45, +1.45] cm in (x, y, z), as all charge in every 7 keV_{ee} F recoil satisfies this
- We then bin this region into a 32 x 32 x 32 voxel grid => ~850μm voxel widths
  - ~32kB/event = ~330GB for 1e7 events

A.S. Lundervold, A. Lundervold/ Z Med Phys 29 (2019) 102–127

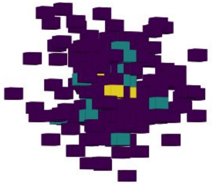






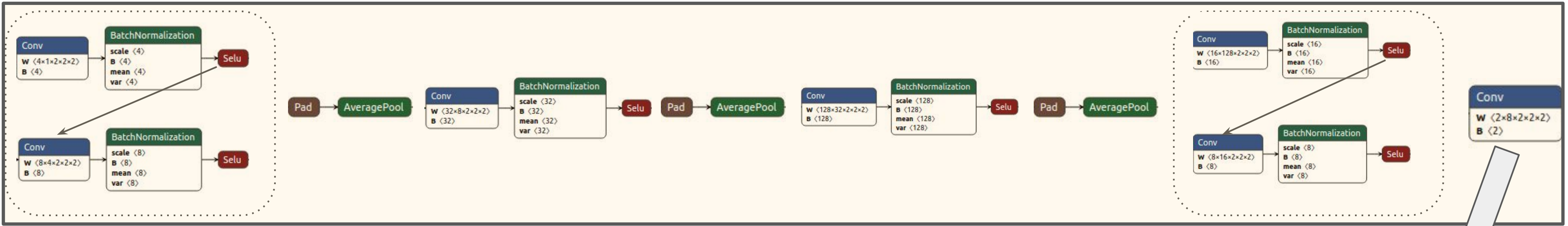
Goal: Instead of using pre-defined observables, directly feed an artificial neural network images of recoils and let it learn which features characterize e-recoils vs F-recoils

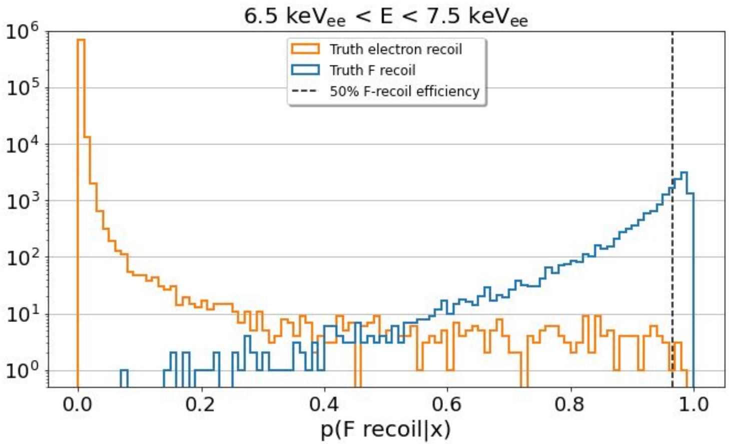
Network architecture



Input is a random batch of 256 full 3D voxel images segmented into (32 x 32 x 32) bins filled with 8-bit unsigned integers representing the charge per voxel. Input tensor dimensions are (batch_size x input_channels x H x W x D) = (256 x 1 x 32 x 32 x 32)

Convolutional neural network





Output

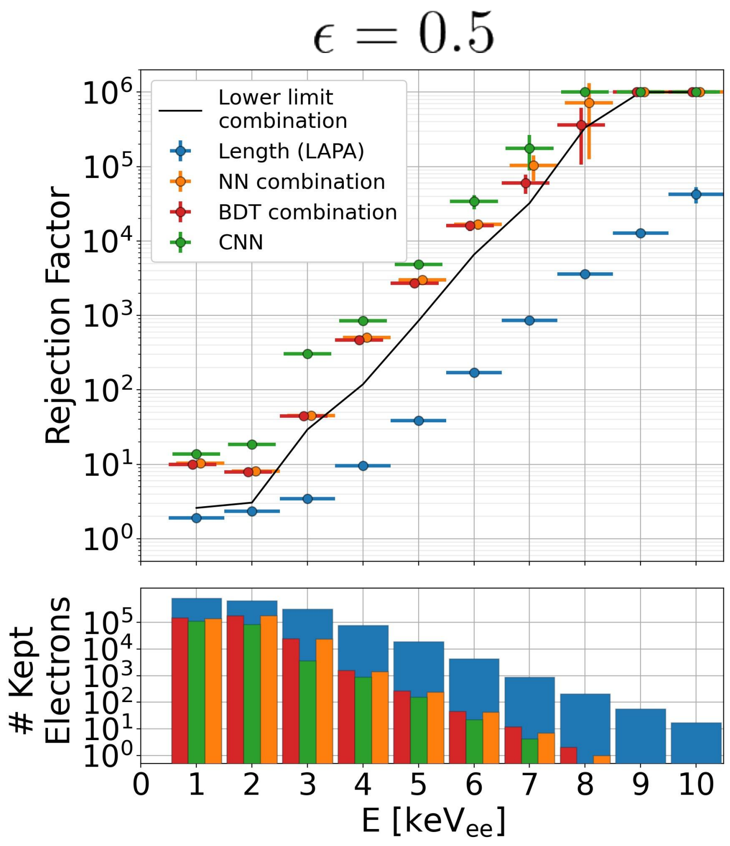
F-recoil classification probability for each event in batch

Note: We train each energy bin separately with the convolutional neural network classifier

Results

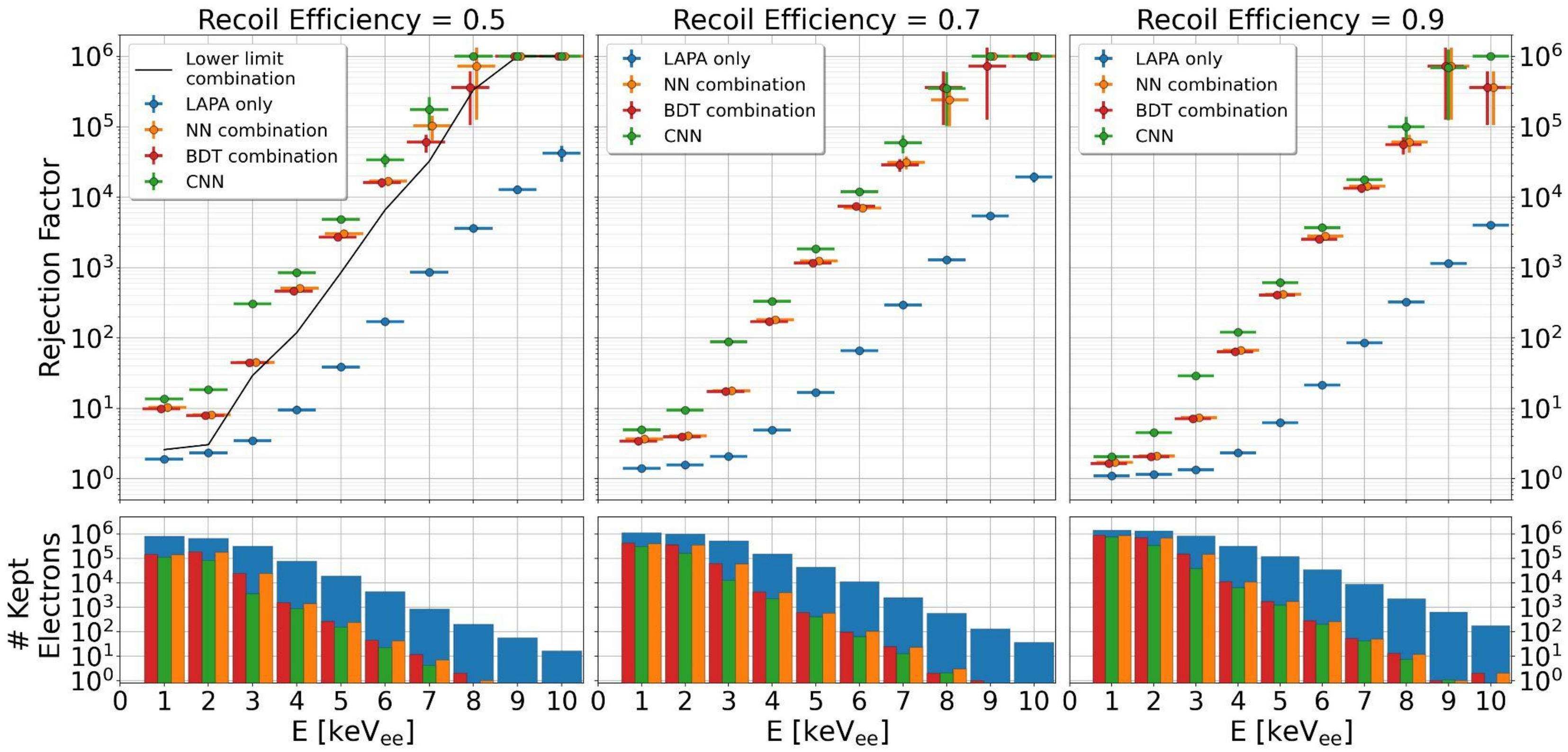
<u>Rejection factor</u>

$$R \equiv \frac{N_{\mathrm{e,total}}}{N_{\mathrm{e,kept}}}$$



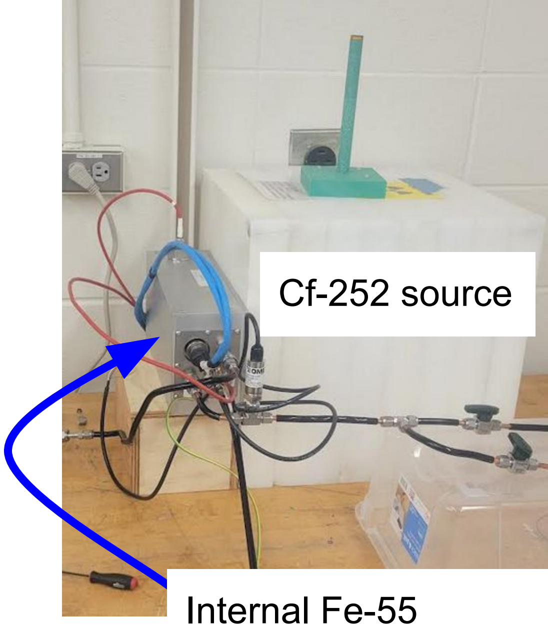
Deep learning outperforms all combinations of predefined observables at 50% F-recoil efficiency

Results summary

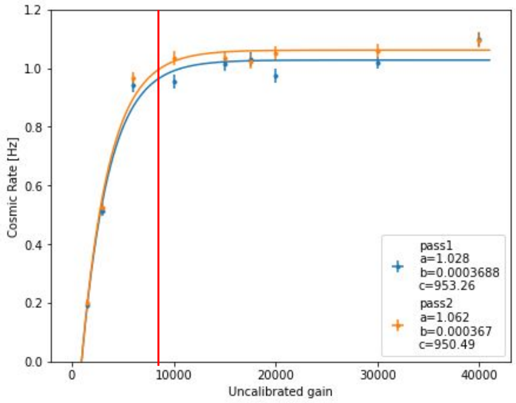


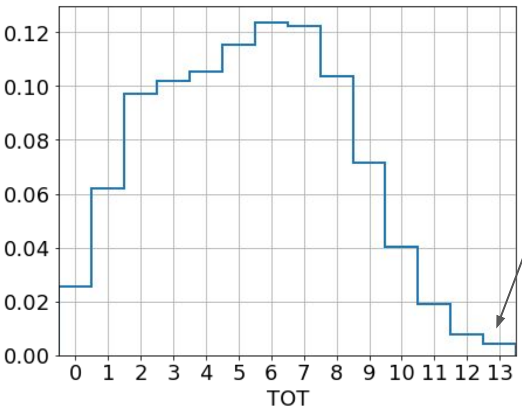
Future studies with BEAST TPCs

- BEAST TPCs use double GEM amplification with a pixel ASIC readout with an 80 x 336 grid of pixels
  - Each pixel is 250μm x 50μm with 2cm x 1.68cm fiducial area
- 70:30 mixture of He:CO_2 gas at 1 atm
- Drift speed ~ 250μm / chip clock cycle
  - 100 cycles read out in event -> max 2.5cm relative z length
- Operating at gain of ~13,500 calibrated with an Fe-55 source
  - Set gain as the highest gain where there is effectively no saturation in Fe-55 X-ray events
  - Energy resolution limited by dynamic range for many recoil events



Internal Fe-55 source with movable shutter



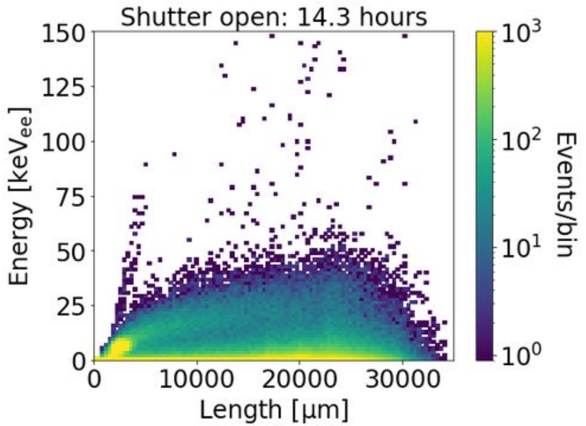


Saturation time over threshold

Studies ongoing with BEAST TPCs operating at near single electron efficiency. Measurements began Dec 2021

Challenges in using deep learning on BEAST data/MC

- Finite dynamic range -> limited energy resolution for recoils
  - Could be an advantage if we're only distinguishing between Fe-55 electrons and neutrons from Cf-252
- If we segment images to feed into a CNN using the (80 x 336 x 100) segmentation of our readout, each event image will take up 2.7 MB
  - 1e6 events will fill up a large hard drive
  - GPU memory will quickly fill up when training as well
  - This segmentation may be doable on an HPC but we may have to consider using coarser voxel bins
- For measurement: obtaining a pure labeled sample of measured nuclear recoils will take some work
  - May have to train on simulation and deploy on data
- For simulation: Digitizing recoil events is computationally expensive



Outlook and summary

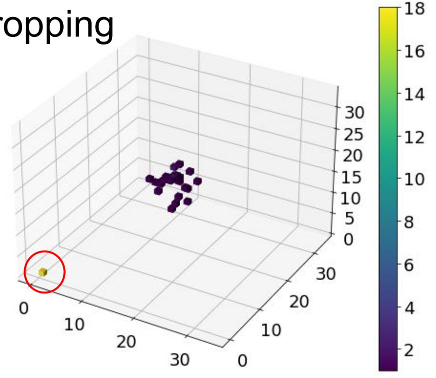
- ❖ Multivariate combinations of observables using machine learning leads to factors of 2-5 increases in rejection factors over previously published results
- ❖ Deep learning seems to be a very fruitful approach for improving electron rejection
  - ➢ Up to a factor of 7 increase in R (in 3 keV_{ee}) bin over ML combinations of observables
- ❖ Began collecting data for e-rejection study on measured data
  - ➢ >1e6 Fe-55 events collected to train on
  - ➢ Still processing neutron data and simulation -> results to come soon!

Backup

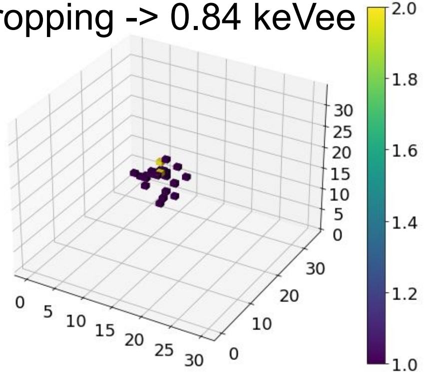
Data processing for neural network

- Electron and F recoils are binned into distinct integer energy bins and combined into one file per bin
  - [(0.5,1.5), (1.5,2.5), (2.5,3.5), … , (9.5,10.5)] keV
- Origin is defined at centroid of charge distribution for each event
- Charge is binned into a 34 x 34 x 34 voxel grid ranging from (-1.45cm,+1.45cm) -> 850μm^3 voxels
- Charge outside of voxel grid is assigned to the nearest *edge* voxel
- Crop out outermost shell of voxel grid when feeding into neural network -> removes any information about charge outside of voxel volume (realistic), leaving a 32 x 32 x 32 grid of voxels
  - No F recoils are cropped (largest extent in any direction is 2cm)
  - Electron recoils are cropped (~15cm largest extent, usually due to multiple charge clusters)

1.4 keVee e recoil before cropping



Same e recoil after cropping -> 0.84 keVee



BDT feature map

- Importance measure is “coverage”