ANU-FRT Summary

PREETHAM .

May-July 2025

1 Introduction

This is a summary of the work done during the ANU-FRT program 2025. Most of the project focused on writing Python codes that analysed the data obtained in the experiment. A summary of each code is given, and where relevant, the results obtained from the code are provided too.

2 Analyzing the .bin files

As the experiment is taking place, the software installed in the lab shows real-time readings of the PMT and many other signals. One of the problems was that there wasn't a quick and easy way to assess these signals. For example, if you wanted to know if the signals received right now are more intense than the previous batch (i.e., say, for a specified GEM setting), the only way was to load the whole data and analyse it through some code or guess it on a rough estimate of the naked eye. The codes given in this section ease this process - they can be directly called from the terminal with specific arguments, and they analyse the data and give you the option to save the results. One of the main quantities of interest is the "charge", which is the sum of the area under the histogram in the .bin file. There are three codes in the section -

- Height.py, which gives the height of the max value in a single row of the .bin file and the index of the maximum value.
- Compare.py, which overlays the histograms of the charge (normalised) of two bin file inputs.
- round_window.py analyses the given .bin file and returns the histogram of charge (normalised) with the option to save the plot and the result data as a .csv file.

A small tip is that the syntax and a short explanation of each of these codes can be obtained by the "–help" command.

3 Analysing the Ion-backfire

Here, the goal was to analyse the intensity of the images captured in 3 separate regions - where the track is (i.e., where the particle passed through), the region inside the camera viewport but not the track region and outside the camera's viewport. The code used was "Can you see the pixels.ipynb", which is a Jupyter notebook file and is pretty self-explanatory. A very basic noise reduction method was employed, ignoring all pixel values below 10. The 3 regions selected for analysis are shown in the code. An interesting result was that the plot of the track to dark intensity ratio v/s the intensifier gate length showed a prominent peak, as shown below. This might be evidence of the ion-backfire process, in which the negative ions produced in the experiment travel to the GEM, where they undergo an avalanche, after which they ionize the gas around them, which produces a positive ion that travels towards the cathode and ionizes the gas on its path again. These ions travel back towards the GEM and cause a secondary avalanche process and thus a peak in intensity.

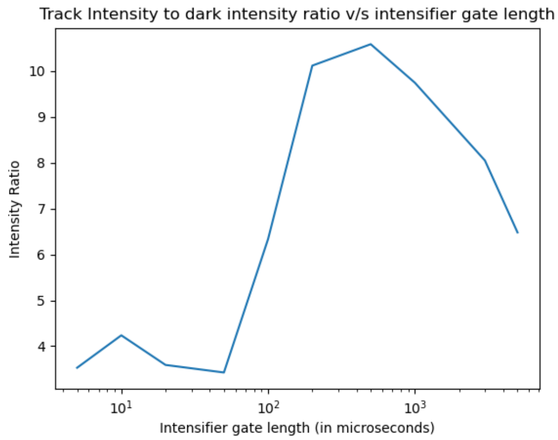

Figure 1: Ion-backfire

4 Techniques to measure Intensity

To measure the intensity of a track, there are several methods. One way is to just simply sum the pixel values in a track. Second is to count the number of pixels that fired in the track. Third is to count the number of small clusters in the track (which in turn gives an estimate of the number of electrons produced in the event). Each method has its pros and cons - on paper, we would expect the accuracy of estimated intensity to increase along the methods in the given order. However, due to problems in finding the correct parameters for the clustering algorithm and probably also due to the fact that there wasn't a rigorous noise reduction technique, the latter 2 methods weren't significantly better than the summed intensity method. The code for this is the "Track.ipynb". A direct comparison for the case of Katie's data was done in the "Final.ipynb" file.

5 Miscellanous

In this section, I have put various images and codes that were used/made in the project, but don't quite fit in other sections. Open at your own risk :).