CALIFA Proton Reconstruction using Neural Networks

Gabriel García Jiménez, Héctor Álvarez Pol

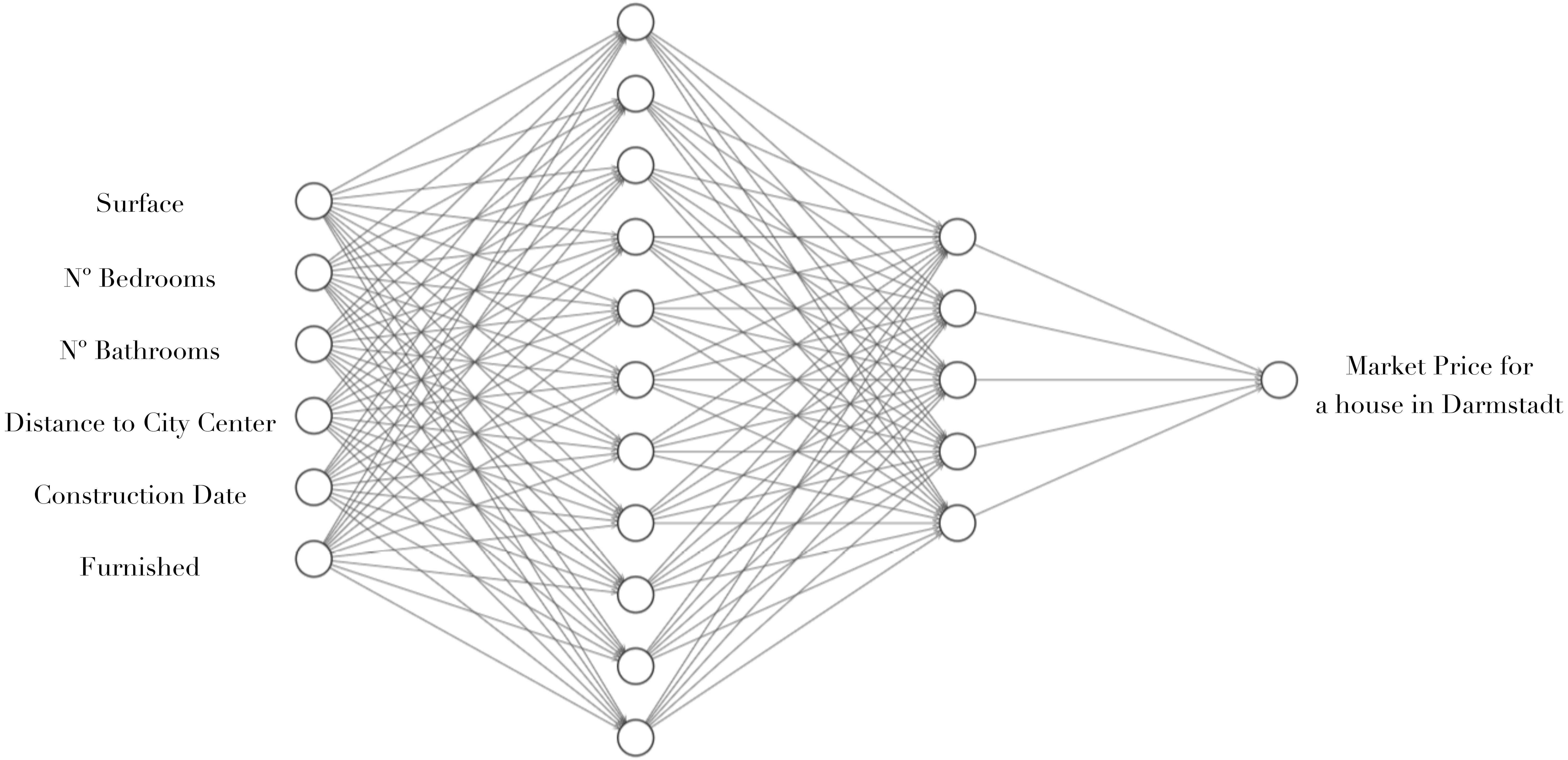
University of Santiago de Compostela (IGFAE)

Introduction

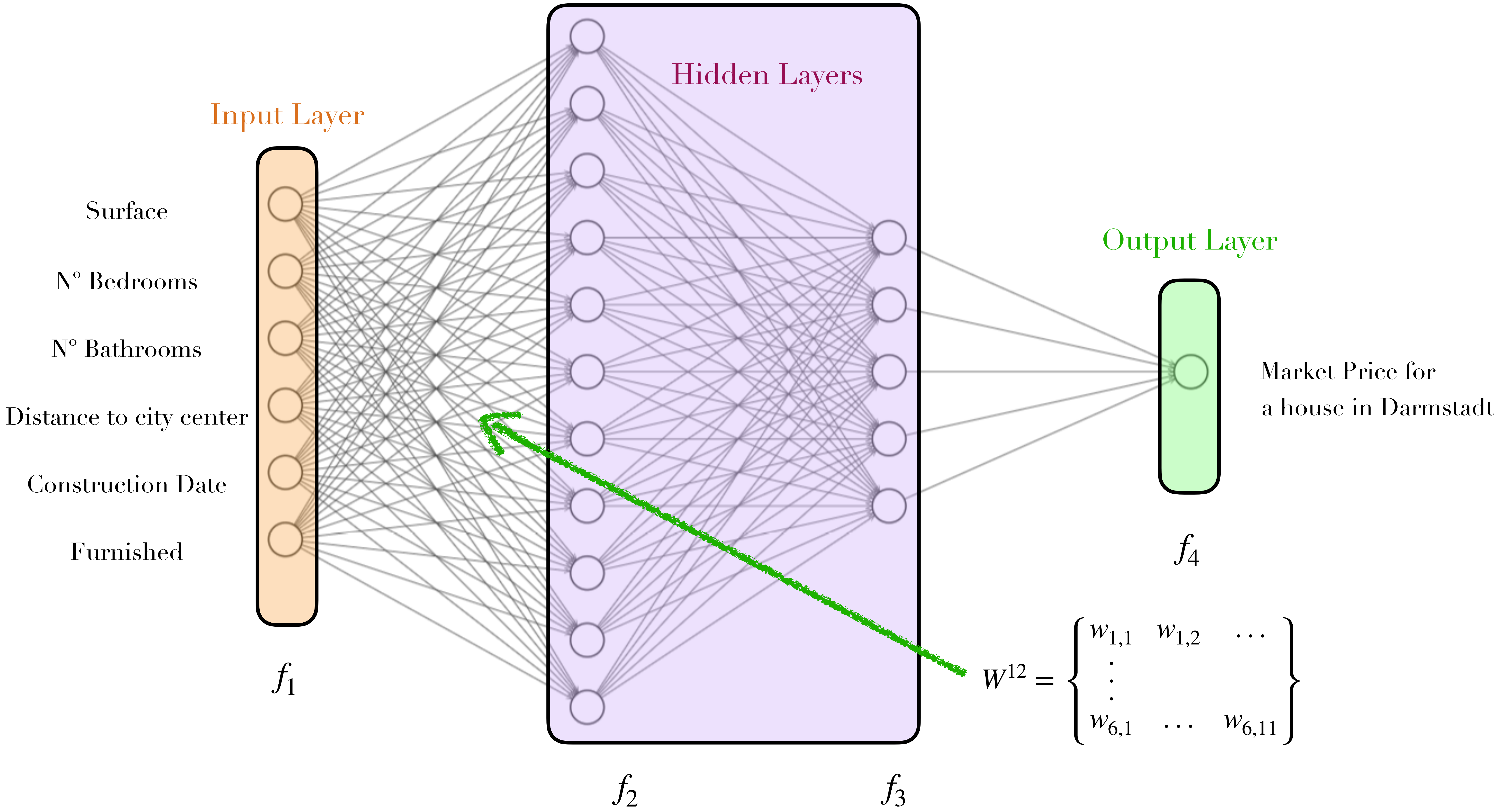
- A neural network is a type of machine learning model inspired by the structure and function of the human brain. It consists of interconnected nodes, called neurons, that process and transmit information to make regressions or classifications.

Introduction

- A neural network is a type of machine learning model inspired by the structure and function of the human brain. It consists of interconnected nodes, called neurons, that process and transmit information to make regressions or classifications.



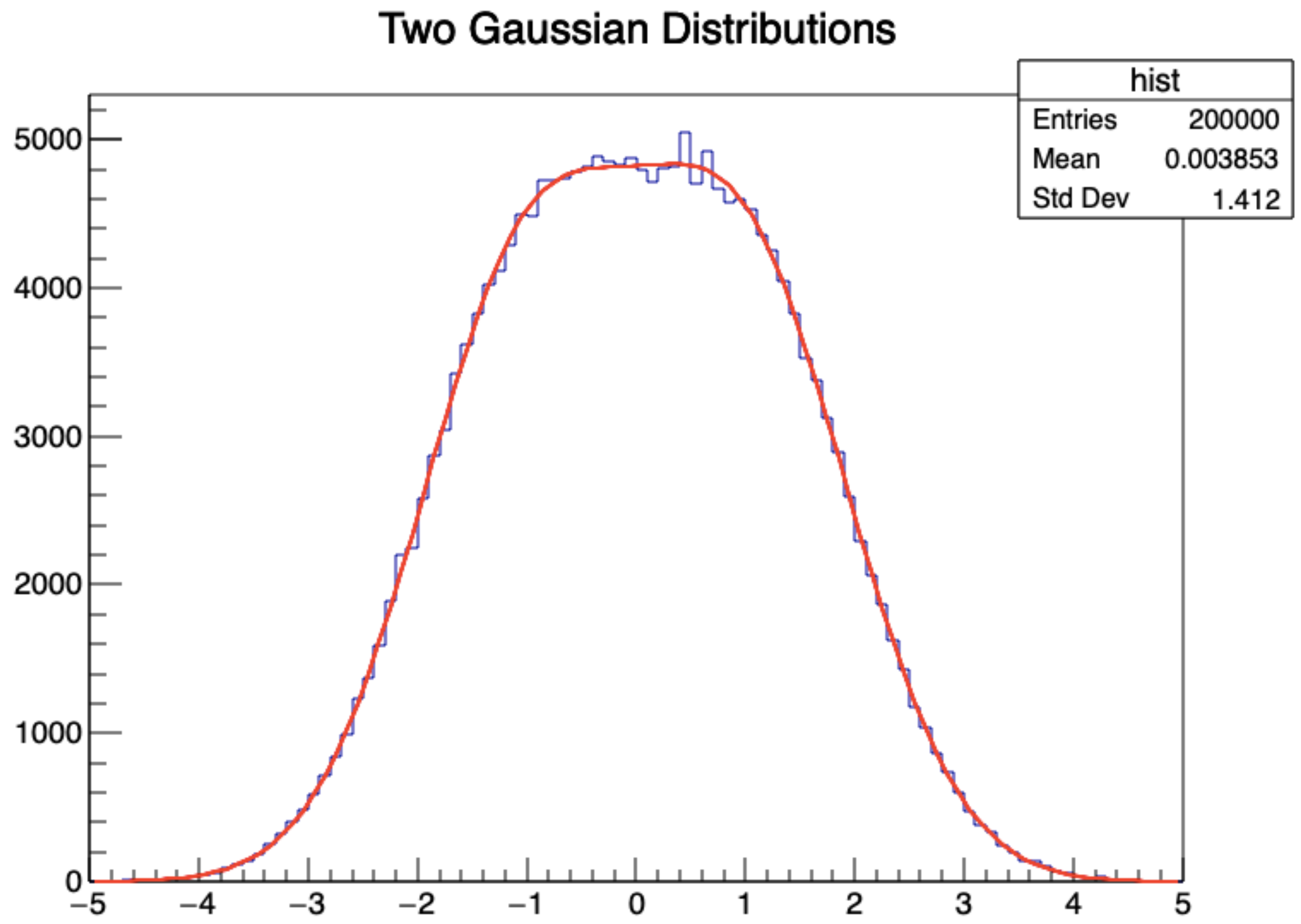
Introduction

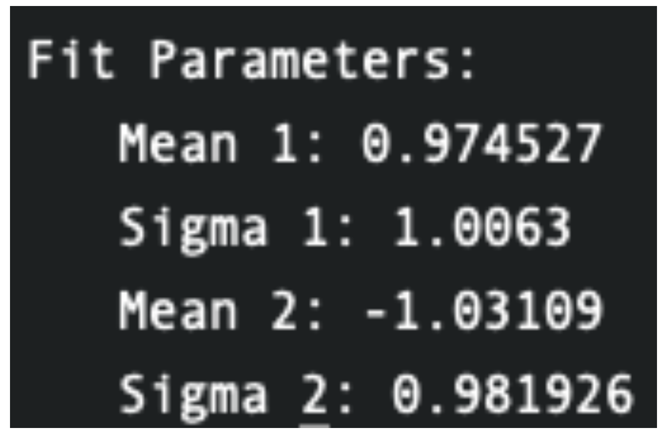


$$W^{12} = \left\{ \begin{matrix} w_{1,1} & w_{1,2} & \cdots \\ \vdots & & \\ w_{6,1} & \ldots & w_{6,11} \end{matrix} \right\}$$

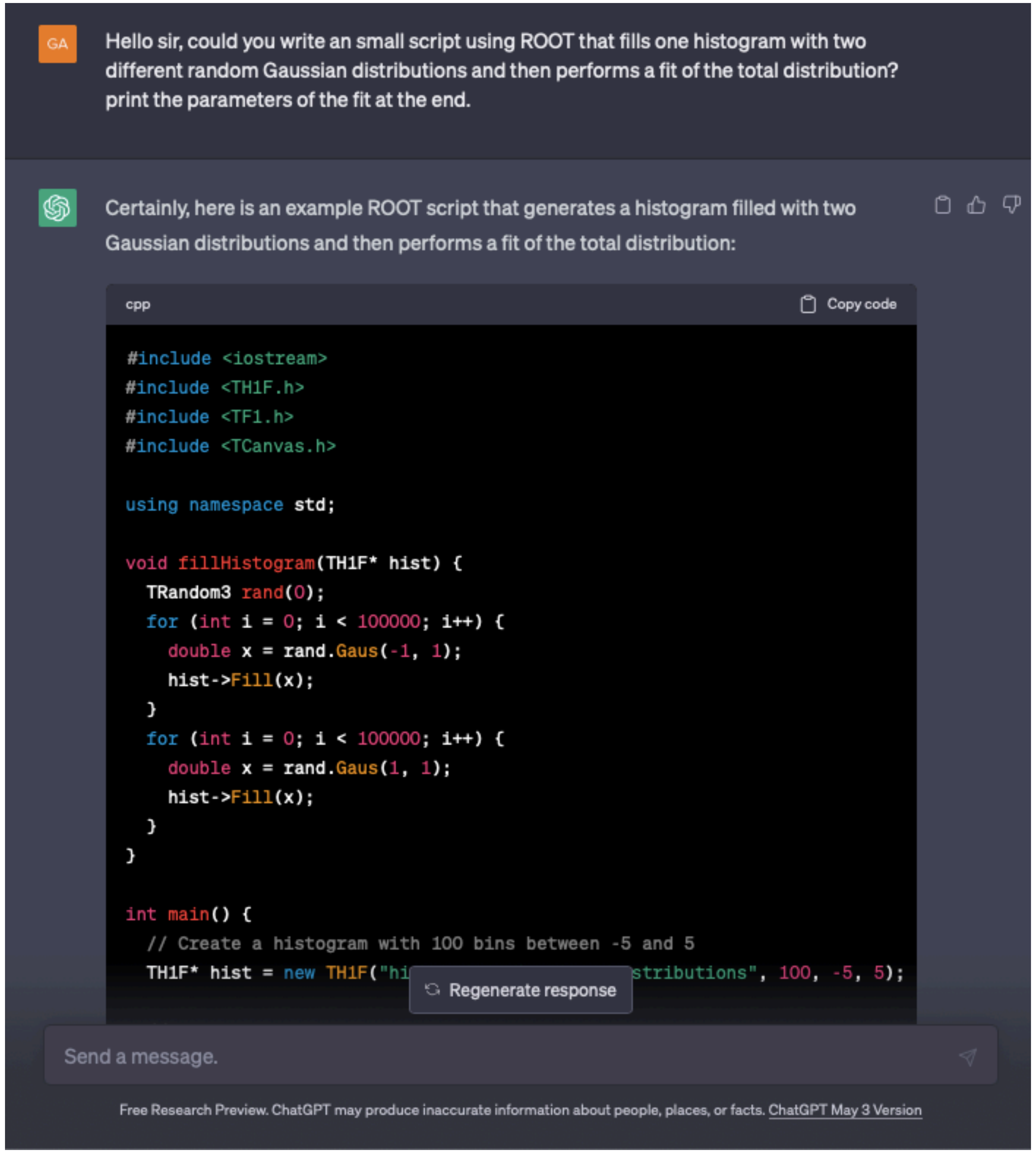
Good prediction? -> $L = L(y, \hat{y})$ ($\hat{y} = Label$) ➡ Backpropagation!

Introduction





```
Fit Parameters:
   Mean 1: 0.974527
   Sigma 1: 1.0063
   Mean 2: -1.03109
   Sigma 2: 0.981926
```



Hello sir, could you write an small script using ROOT that fills one histogram with two different random Gaussian distributions and then performs a fit of the total distribution? print the parameters of the fit at the end.

Certainly, here is an example ROOT script that generates a histogram filled with two Gaussian distributions and then performs a fit of the total distribution:

```cpp
#include <iostream>
#include <TH1F.h>
#include <TF1.h>
#include <TCanvas.h>

using namespace std;

void fillHistogram(TH1F* hist) {
  TRandom3 rand(0);
  for (int i = 0; i < 100000; i++) {
    double x = rand.Gaus(-1, 1);
    hist->Fill(x);
  }
  for (int i = 0; i < 100000; i++) {
    double x = rand.Gaus(1, 1);
    hist->Fill(x);
  }
}

int main() {
  // Create a histogram with 100 bins between -5 and 5
  TH1F* hist = new TH1F("hi                      stributions", 100, -5, 5);
```

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT May 3 Version

Application to CALIFA

1. **Knockout Reactions**
    - **1a. Punch-through classification.**
    - **1b. Punch-through reconstruction.**

2. Gamma and Proton Clustering.

3. Realistic background generation.

4. PID

5. Noise discrimination

6. Cosmic tracking

This started in 2018!

Application to CALIFA

1. **Knockout Reactions**

    1a. Punch-through classification.

    1b. Punch-through reconstruction.

2. Gamma and Proton Clustering.

3. Realistic background generation.

4. PID

5. Noise discrimination

6. Cosmic tracking

SoKAI (Some Kind of Artificial Intelligence)

SoKAI is a neural network framework :

- Written enterily in std C++ (No weird dependencies: CMAKE + GLOG)
- Linked with ROOT (histograms, fits, random generators, etc ...)
- Easy to install : 1) cmake SoKAI 2) make
- Balanced between user friendliness and customization.

This started in 2018!

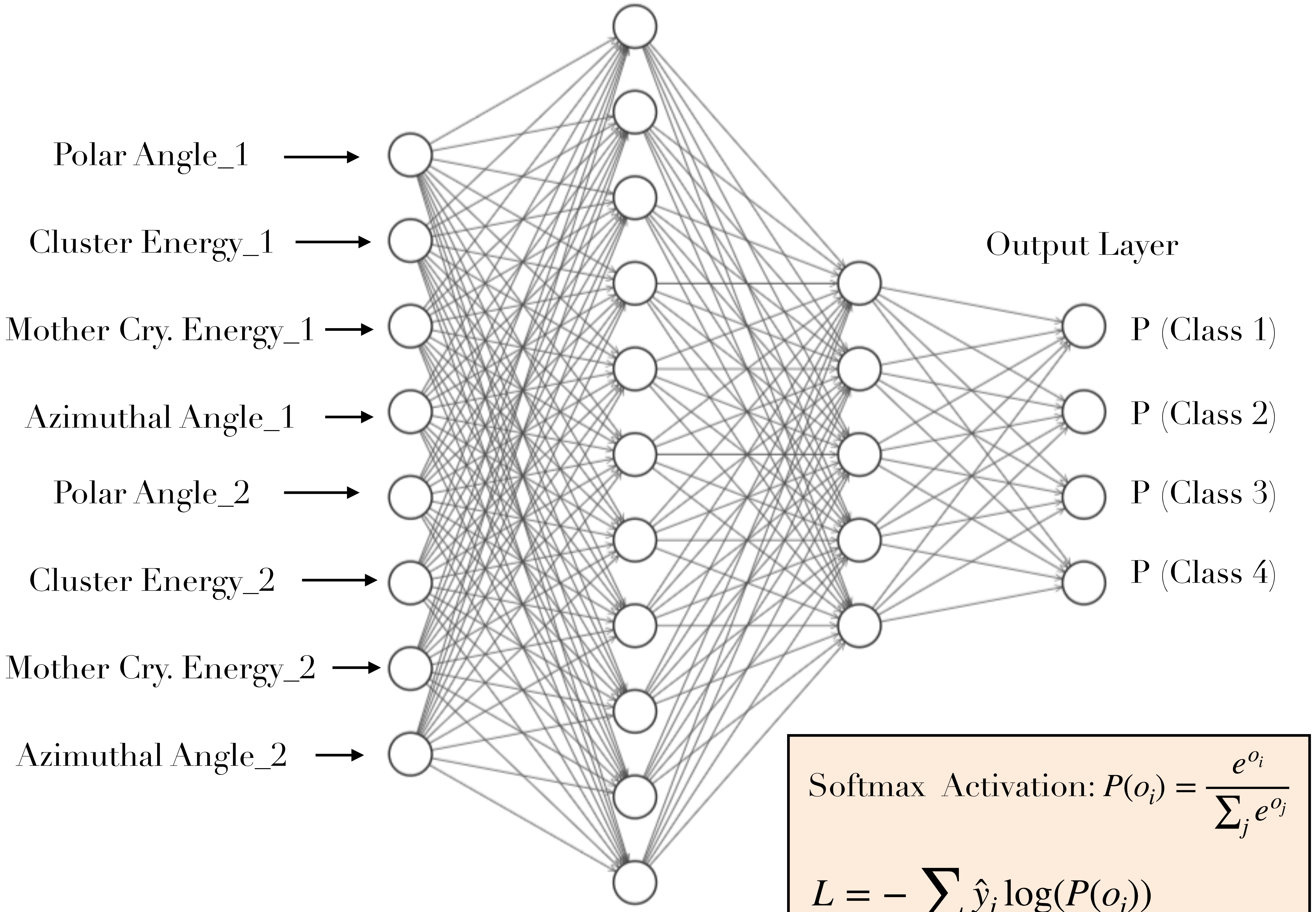
Knockout Classification

- Application to experiment s455.
- We can start by simulating a quasifree channel detected in CALIFA: $^{238}U(p,2p)^{237}Pa$
- With the different observables we can try to train a model for classification: was this a fully stopped proton or a punch through?

Knockout Classification

- We can define Labels for each case and one-hot enconding them:

Class 1, Both Stopped : $(1 \quad 0 \quad 0 \quad 0)$
Class 2, First Stopped, Second Punch : $(0 \quad 1 \quad 0 \quad 0)$
Class 3, First Punch, Second Stopped : $(0 \quad 0 \quad 1 \quad 0)$
Class 4, Both Punch : $(0 \quad 0 \quad 0 \quad 1)$



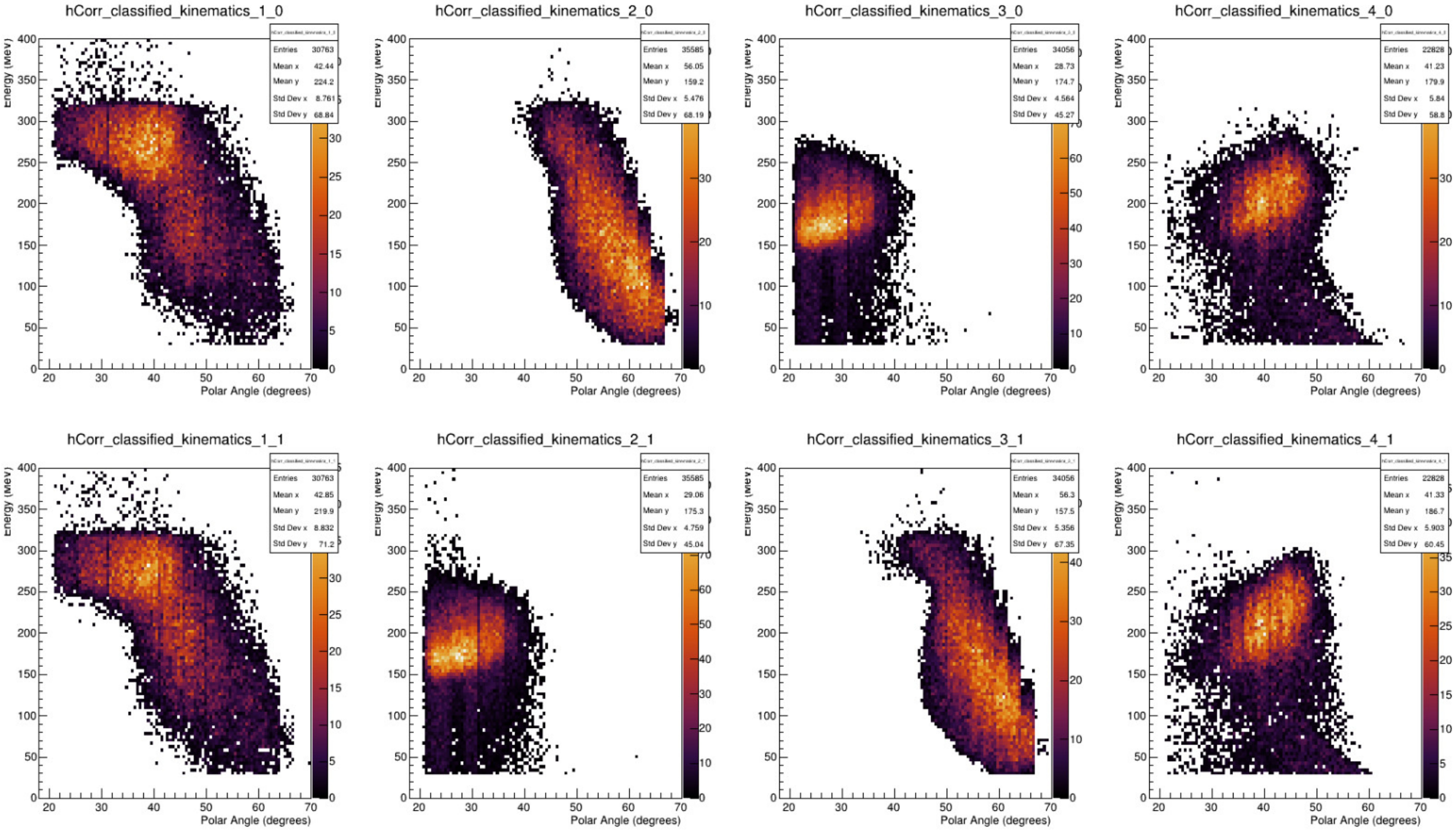
Softmax Activation: $P(o_i) = \dfrac{e^{o_i}}{\sum_j e^{o_j}}$

$$L = -\sum_i \hat{y}_i \log(P(o_i))$$

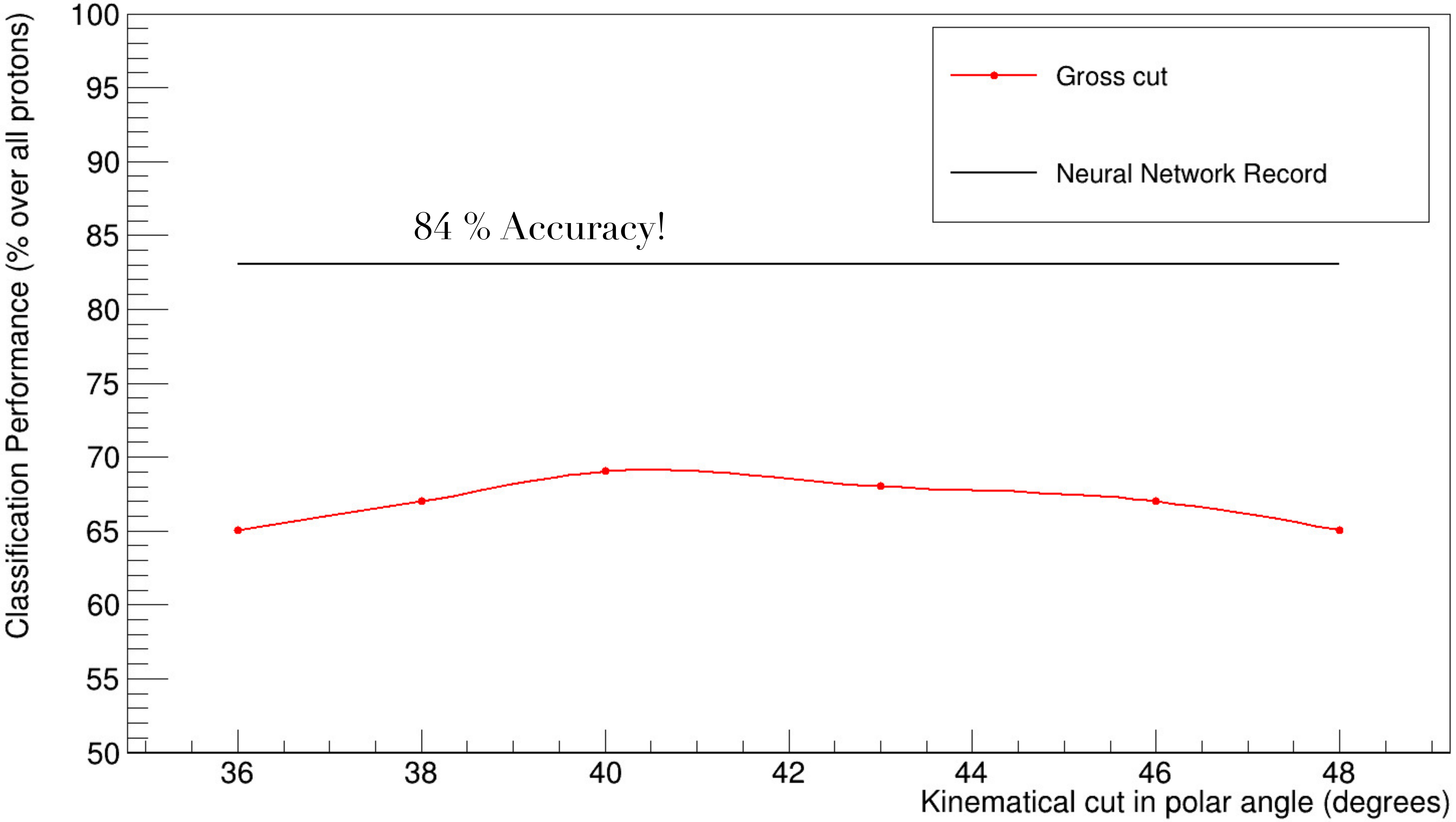
Knockout Classification

Neural Network Architecture:

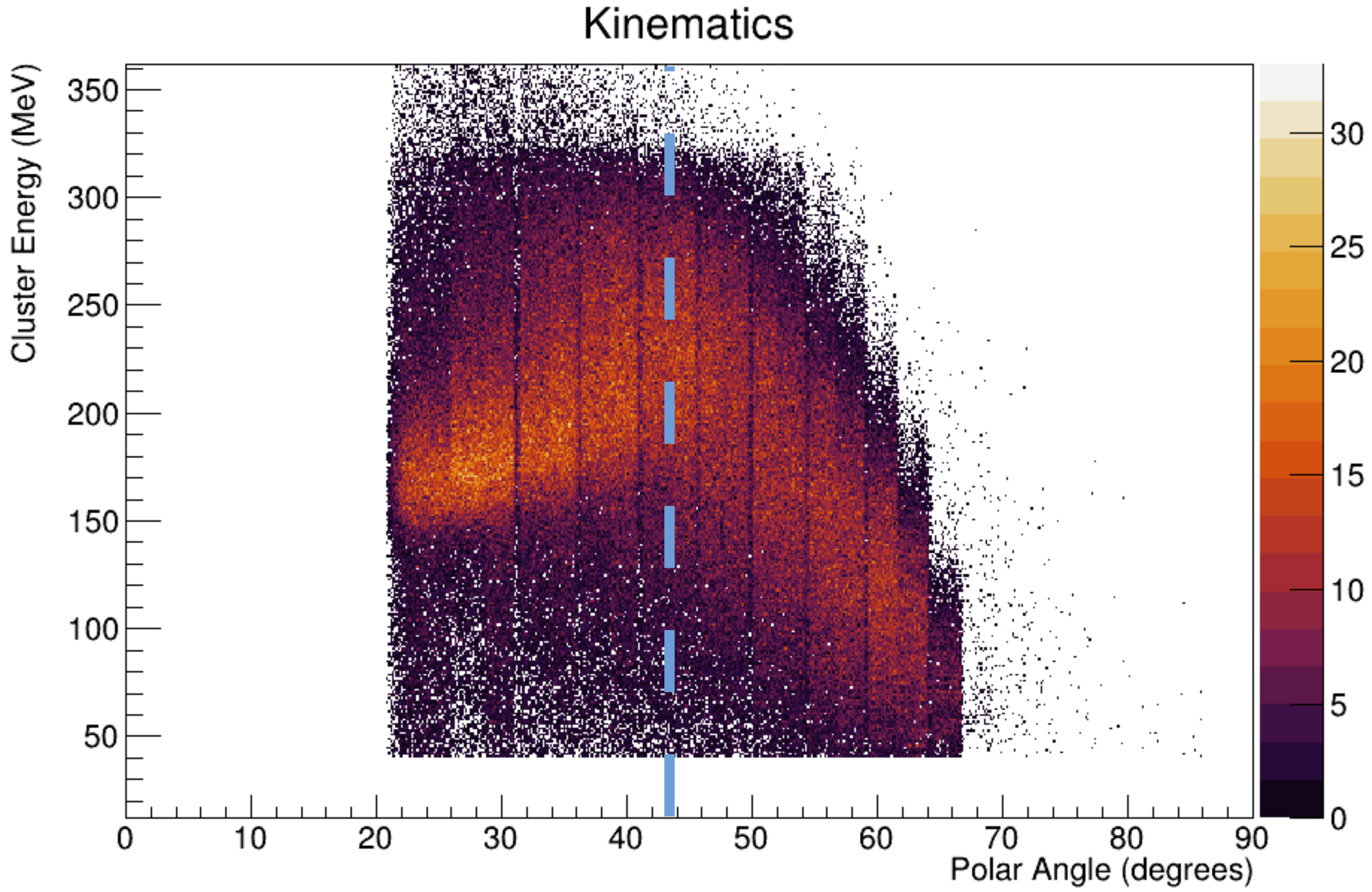
- 4 Layers (L, L, L, L).
- Cross-Entropy Loss.
- Sizes : 8, 10, 20 ,4 = 360 parameters.
- Adam Optimizer ($\alpha_{initial} = 0.001$)
- Batch size: 16



Knockout Classification



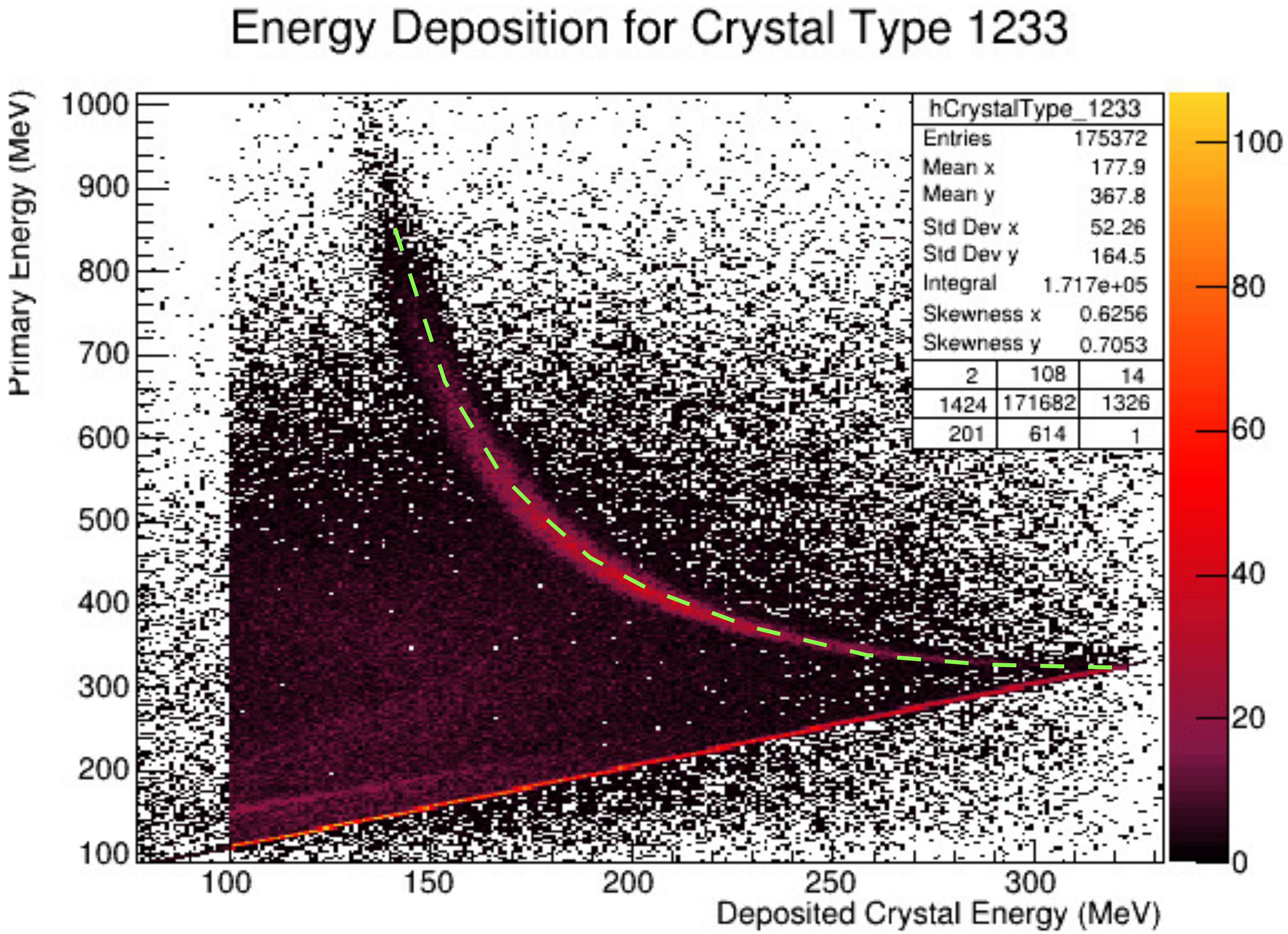
Gross Cut



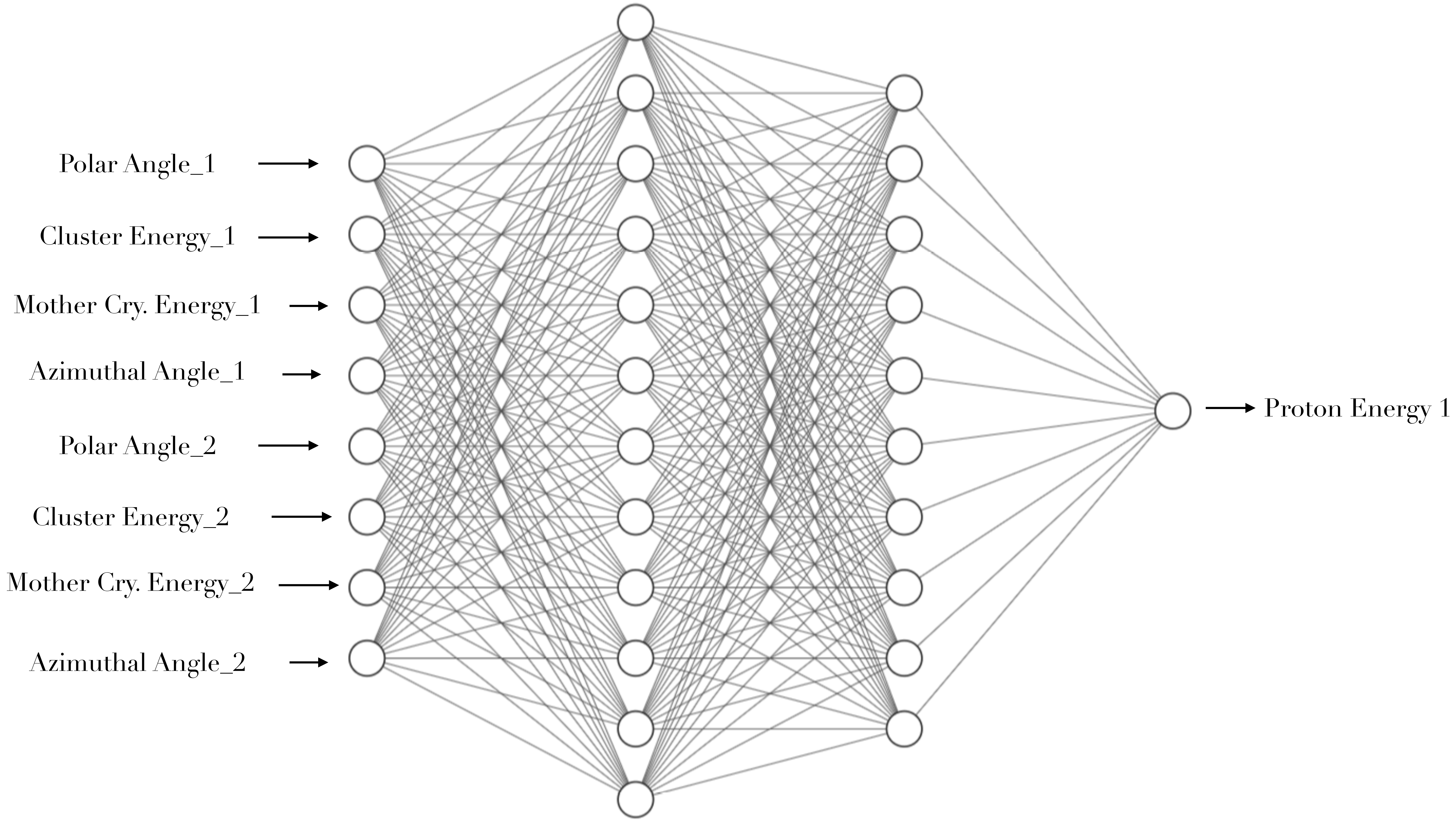
Knockout Reconstruction

Traditional Method for Punch-through reconstruction:

1. Simulate a wide range of energies (INCL , $^{238}U(p,2p)^{X}Pa*$)
2. Represent primary energy vs cluster or single crystal energy.
3. Make different plots for different geometries.
4. Fit the curve.
5. Reconstruct the energy using deposited energy.



Knockout Reconstruction



Polar Angle_1

Cluster Energy_1

Mother Cry. Energy_1

Azimuthal Angle_1

Polar Angle_2

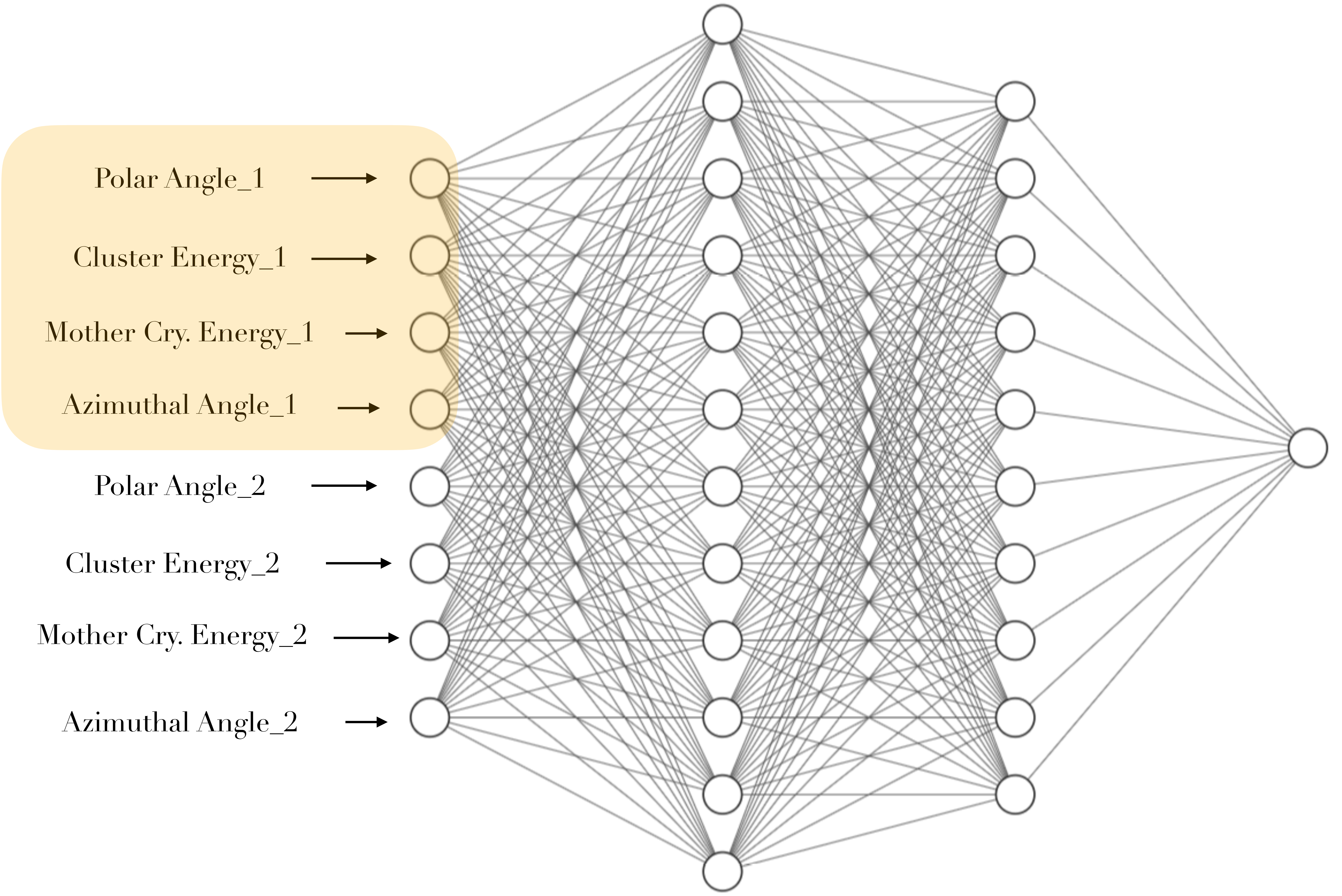
Cluster Energy_2

Mother Cry. Energy_2

Azimuthal Angle_2

Proton Energy 1

Knockout Reconstruction



Polar Angle_1 →
Cluster Energy_1 →
Mother Cry. Energy_1 →
Azimuthal Angle_1 →
Polar Angle_2 →
Cluster Energy_2 →
Mother Cry. Energy_2 →
Azimuthal Angle_2 →

→ Proton Energy 1

$$L = \frac{1}{N} \sum_i \left(y_i - \hat{y}_i\right)^2 \quad \text{(Quadratic)}$$

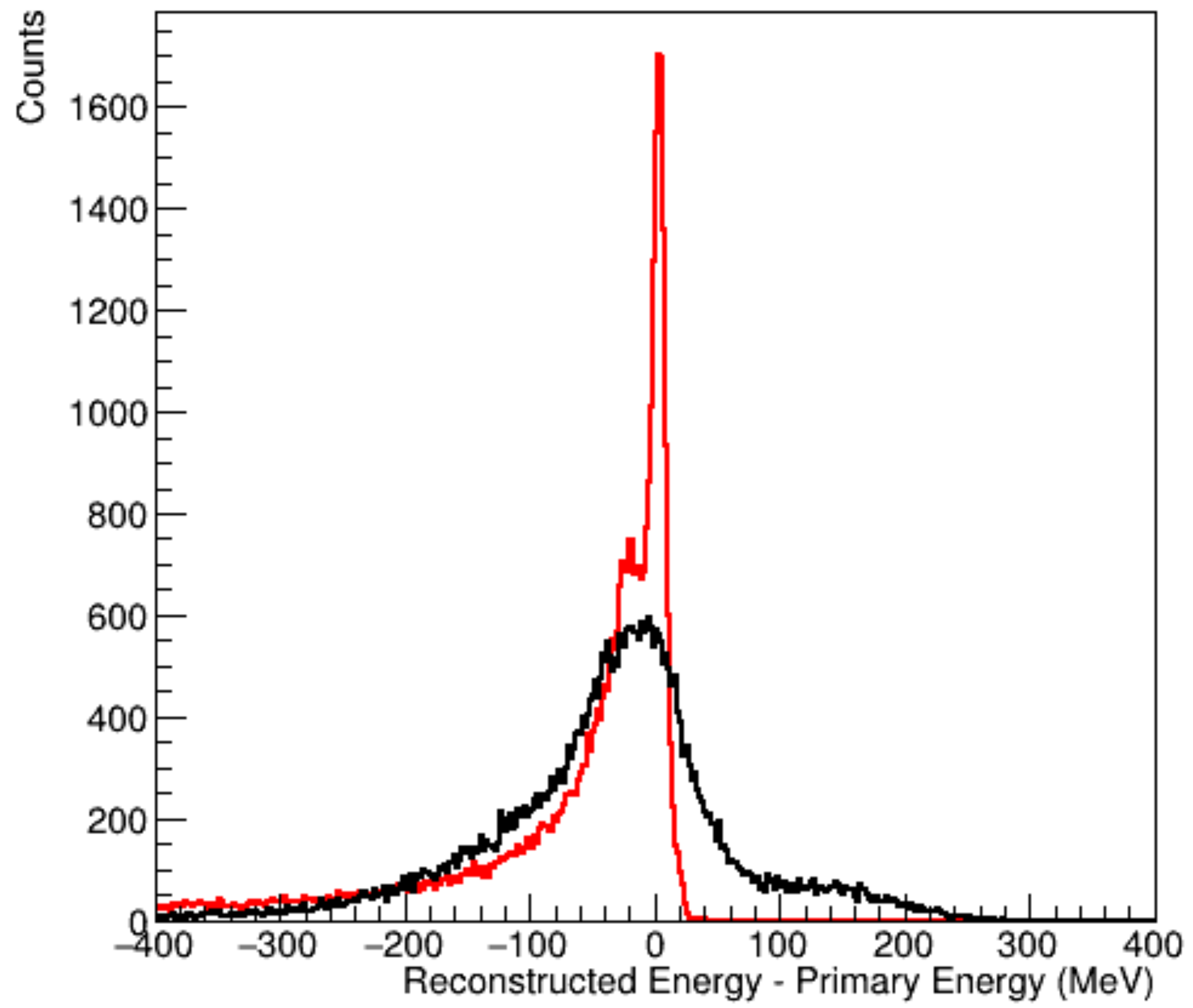
$$L = \frac{1}{N} \sum_i \left|y_i - \hat{y}_i\right| \quad \text{(Absolute)}$$

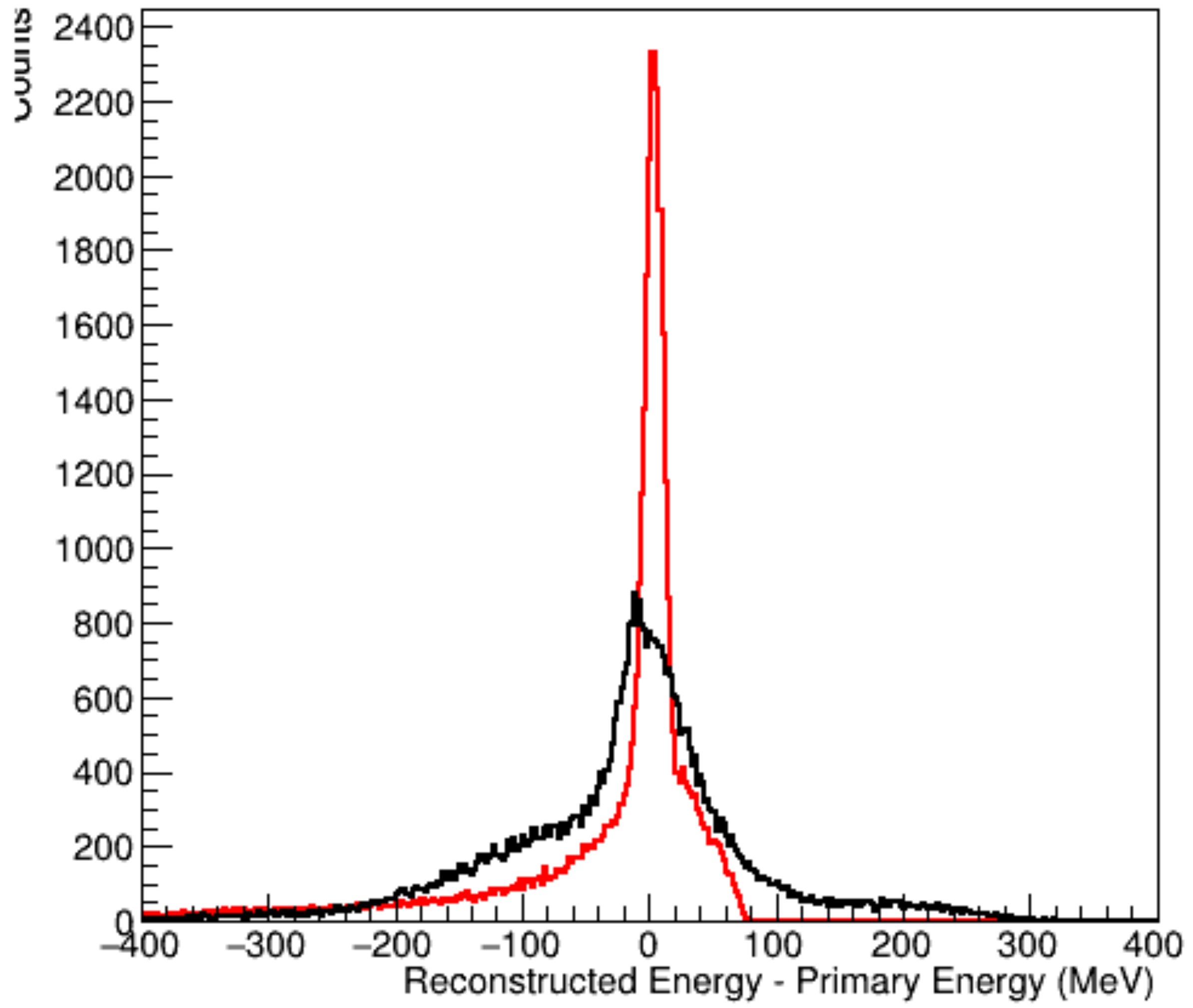
Knockout Reconstruction: Residues

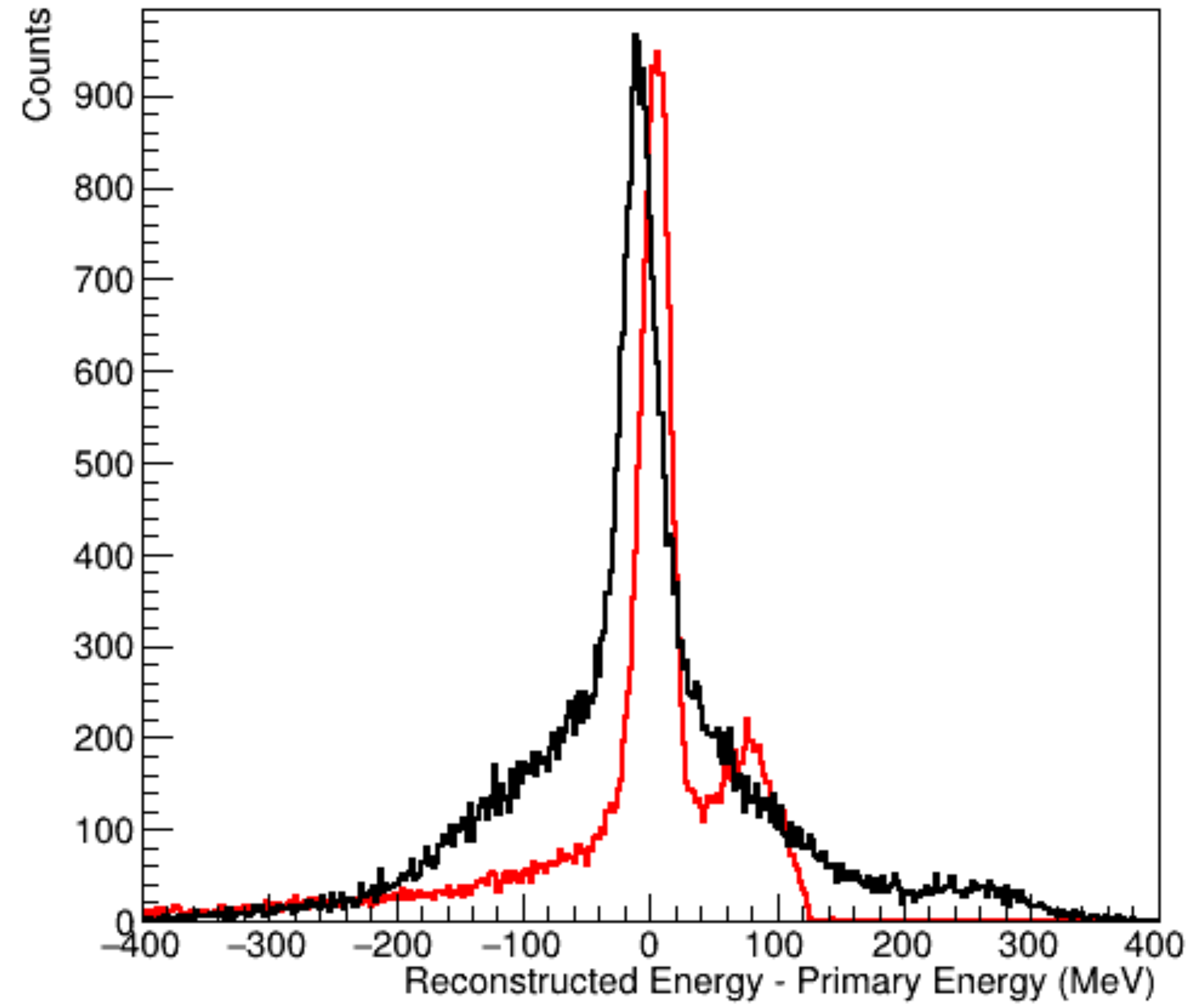
Neural Network
Traditional Fit

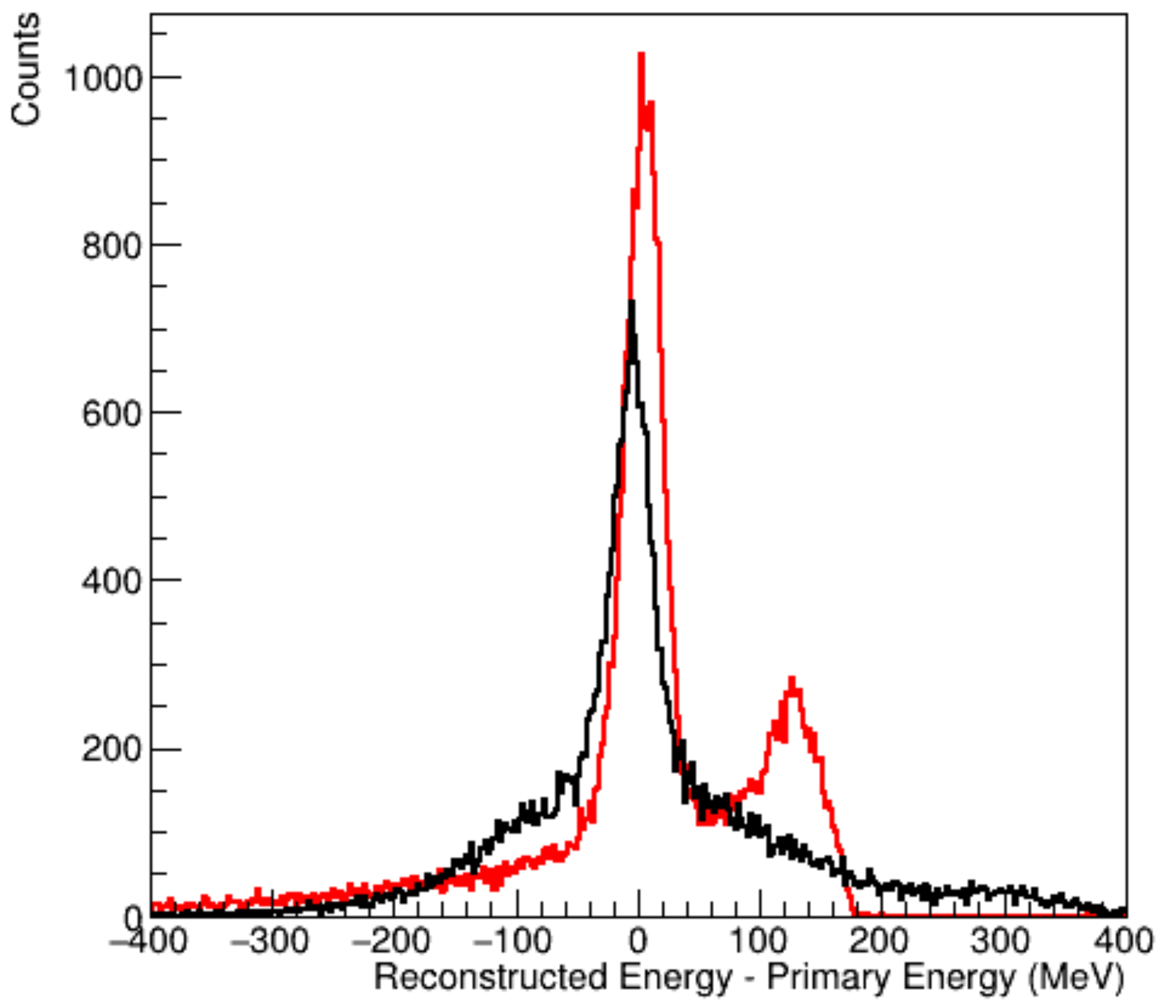
Neural Network Architecture:

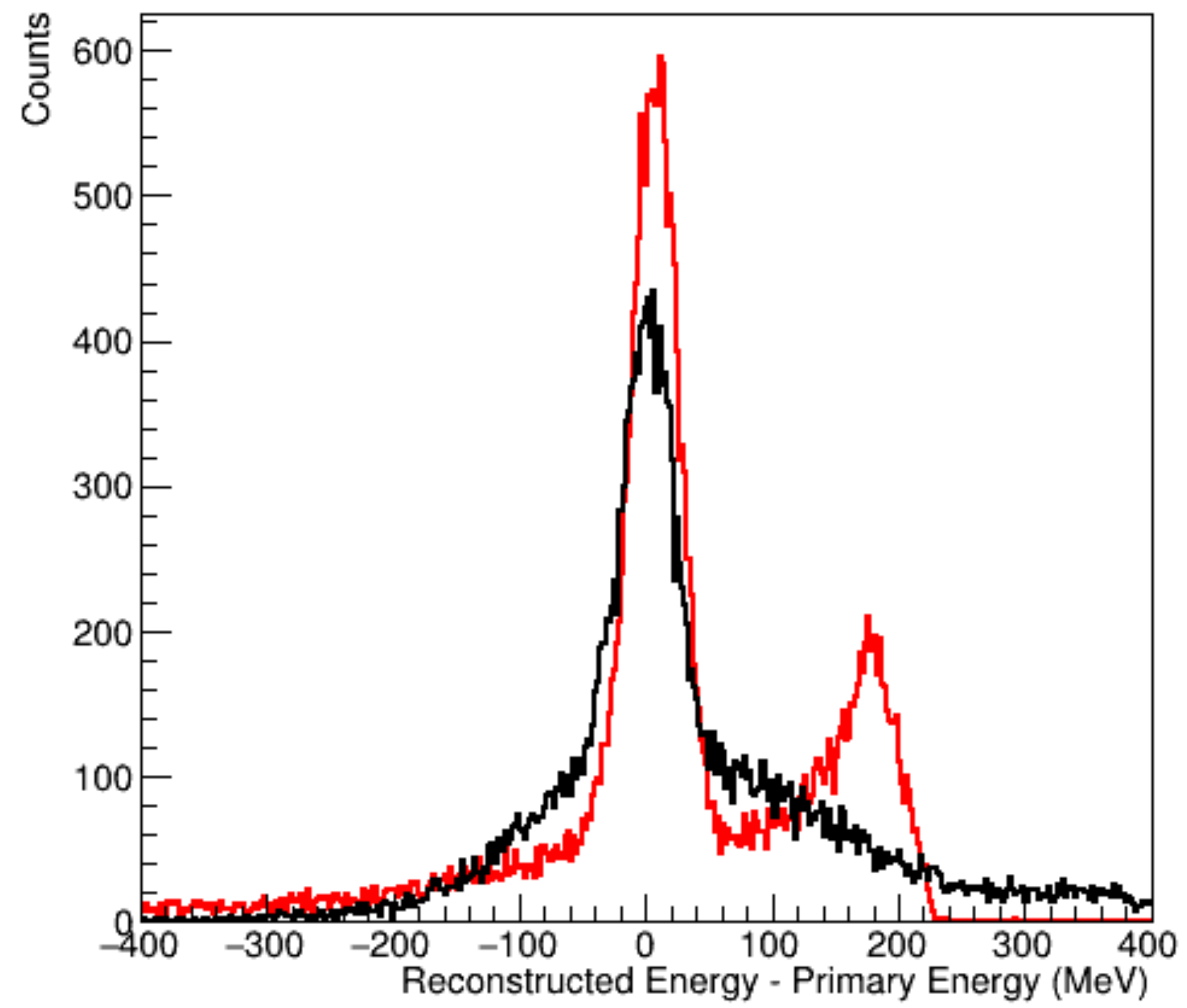
- 5 Layers (L, S, L, L, L).
- Quadratic Loss.
- Sizes : 8, 8,12, 8 and 1 = 264 parameters.
- Adam Optimizer ($\alpha_{initial} = 0.01$)
- Batch size: 8

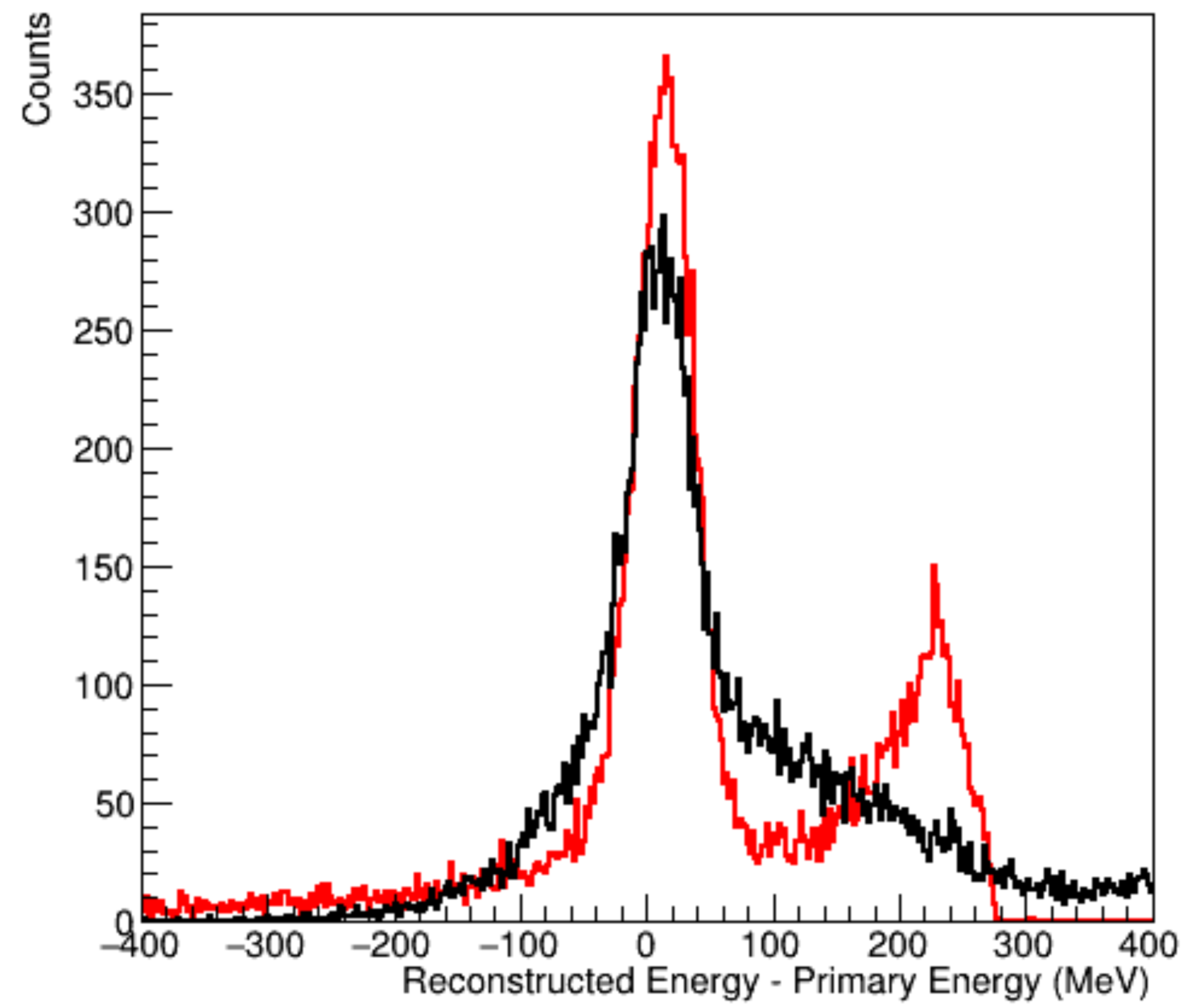

Resolution: 300 - 350 MeV


Resolution: 350 - 400 MeV

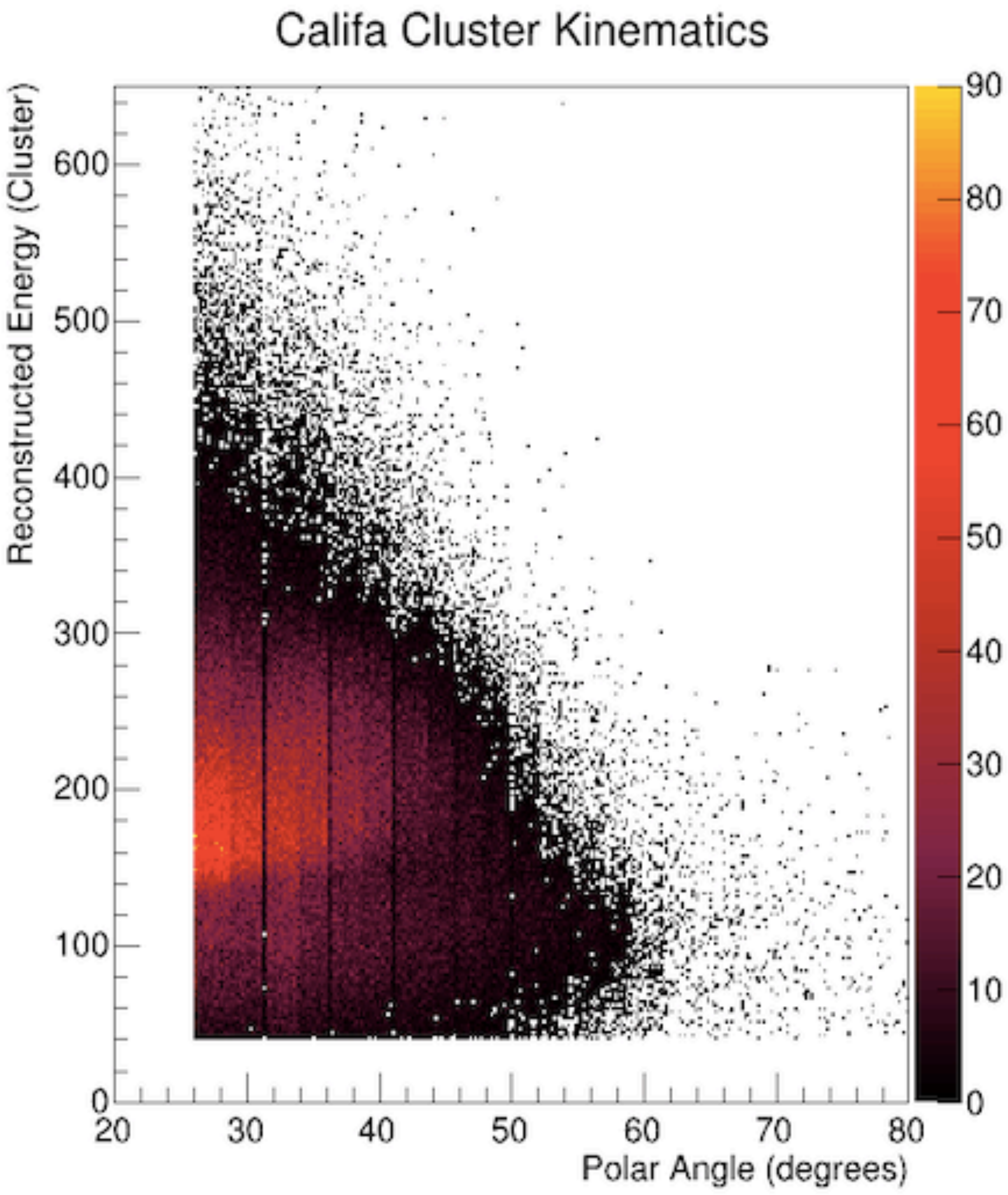

Resolution: 400 - 450 MeV

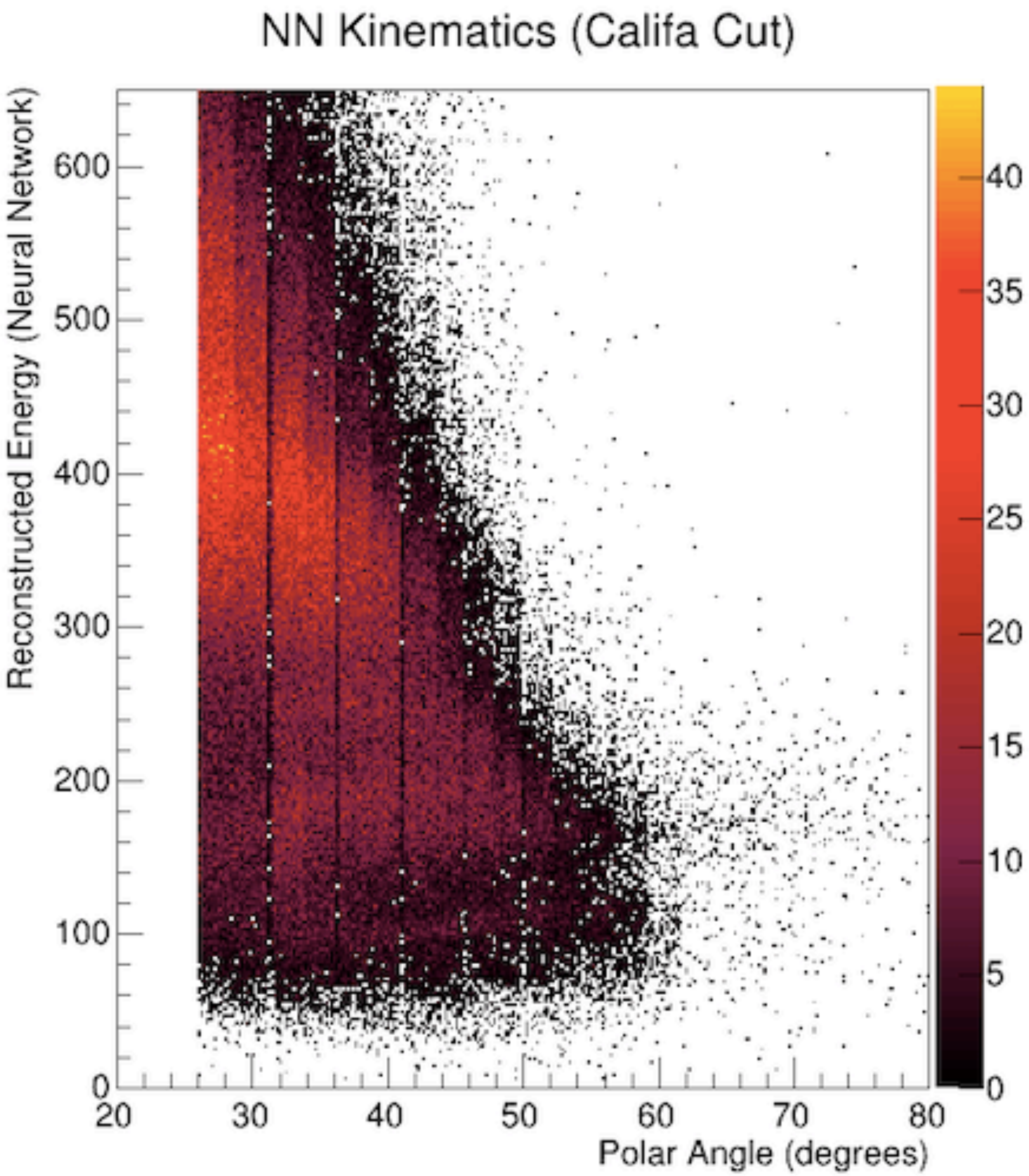

Resolution: 450 - 500 MeV

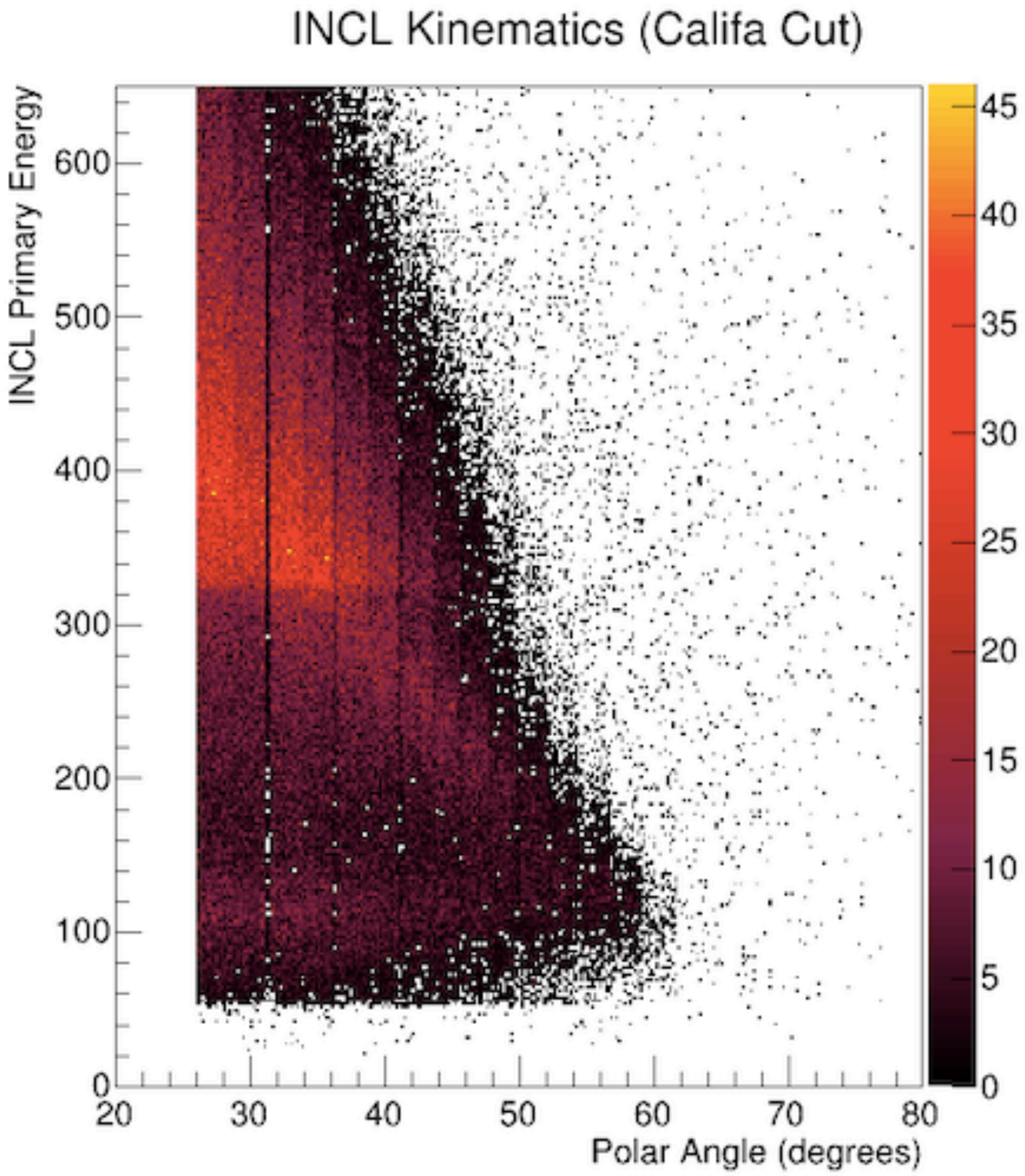

Resolution: 500 - 550 MeV


Resolution: 550 - 600 MeV

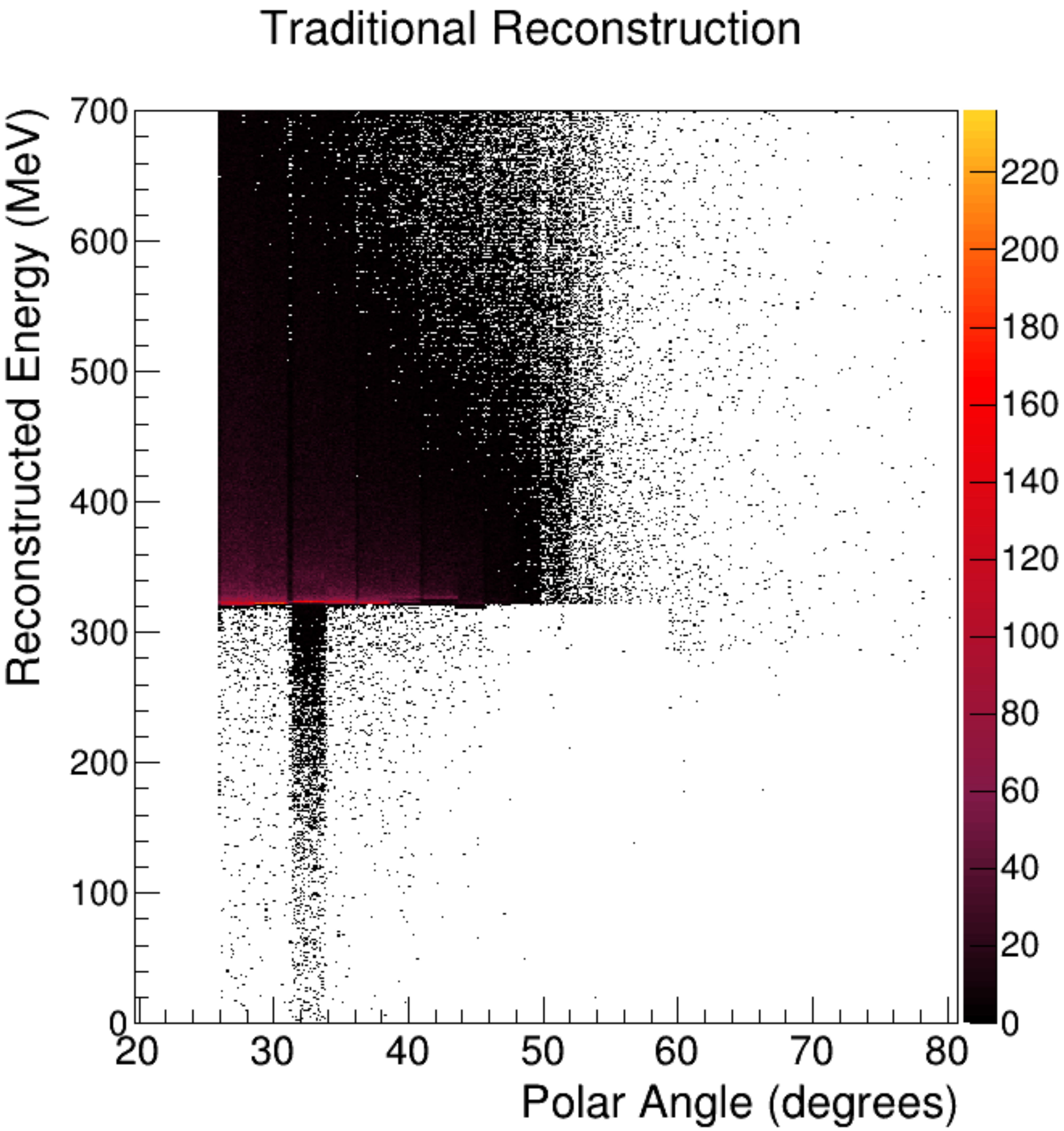
Knockout Reconstruction: Kinematics

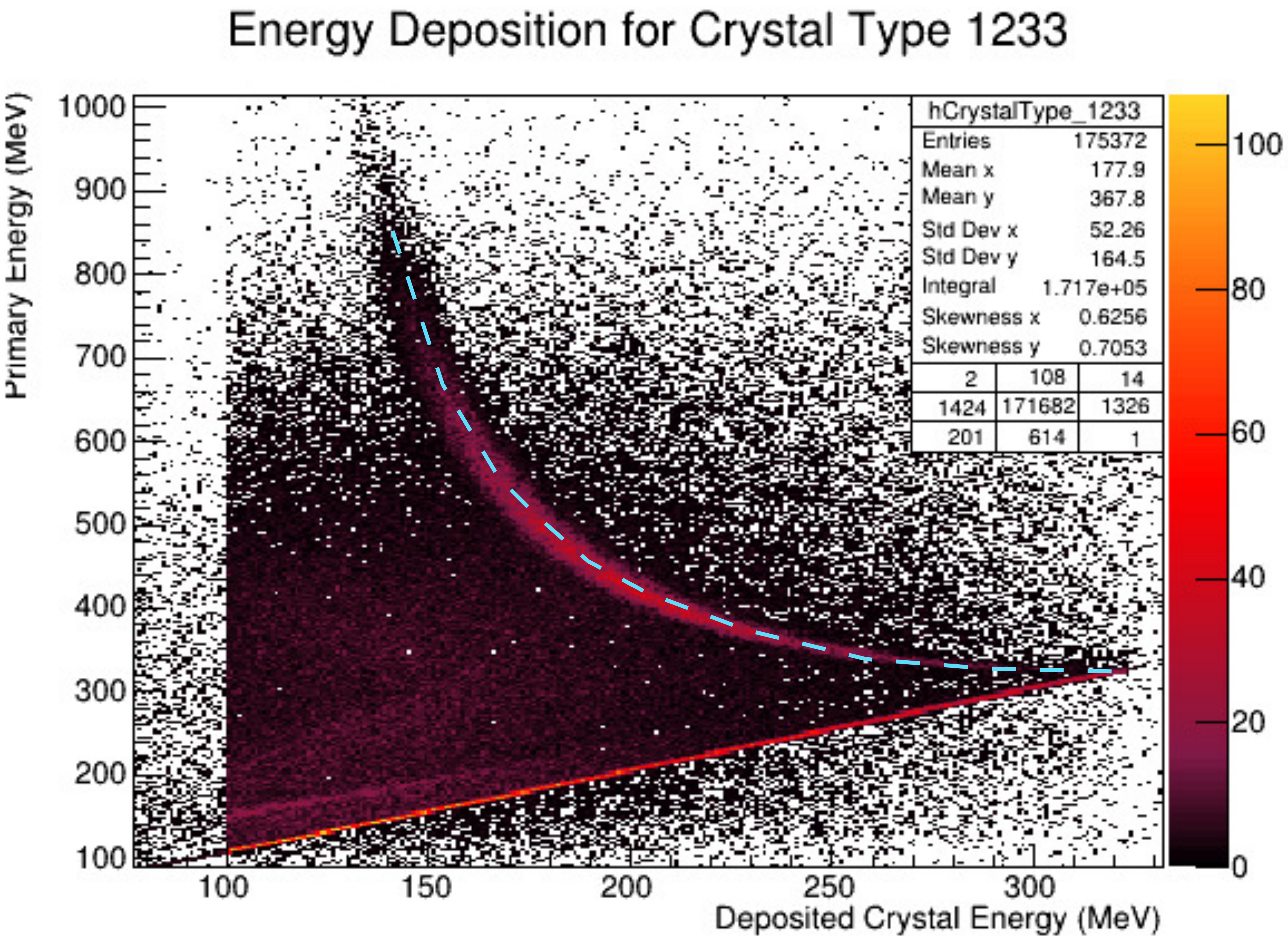




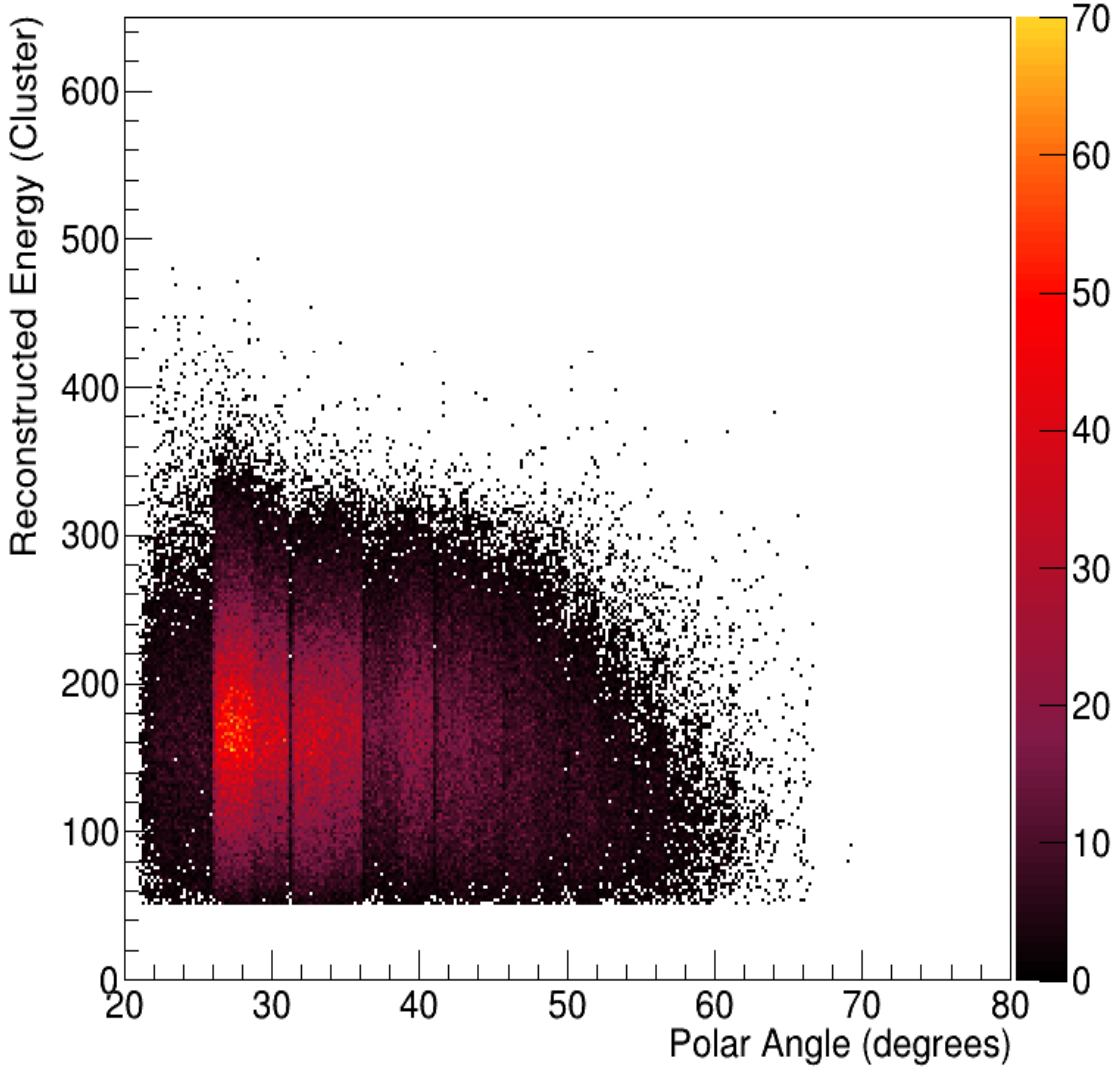


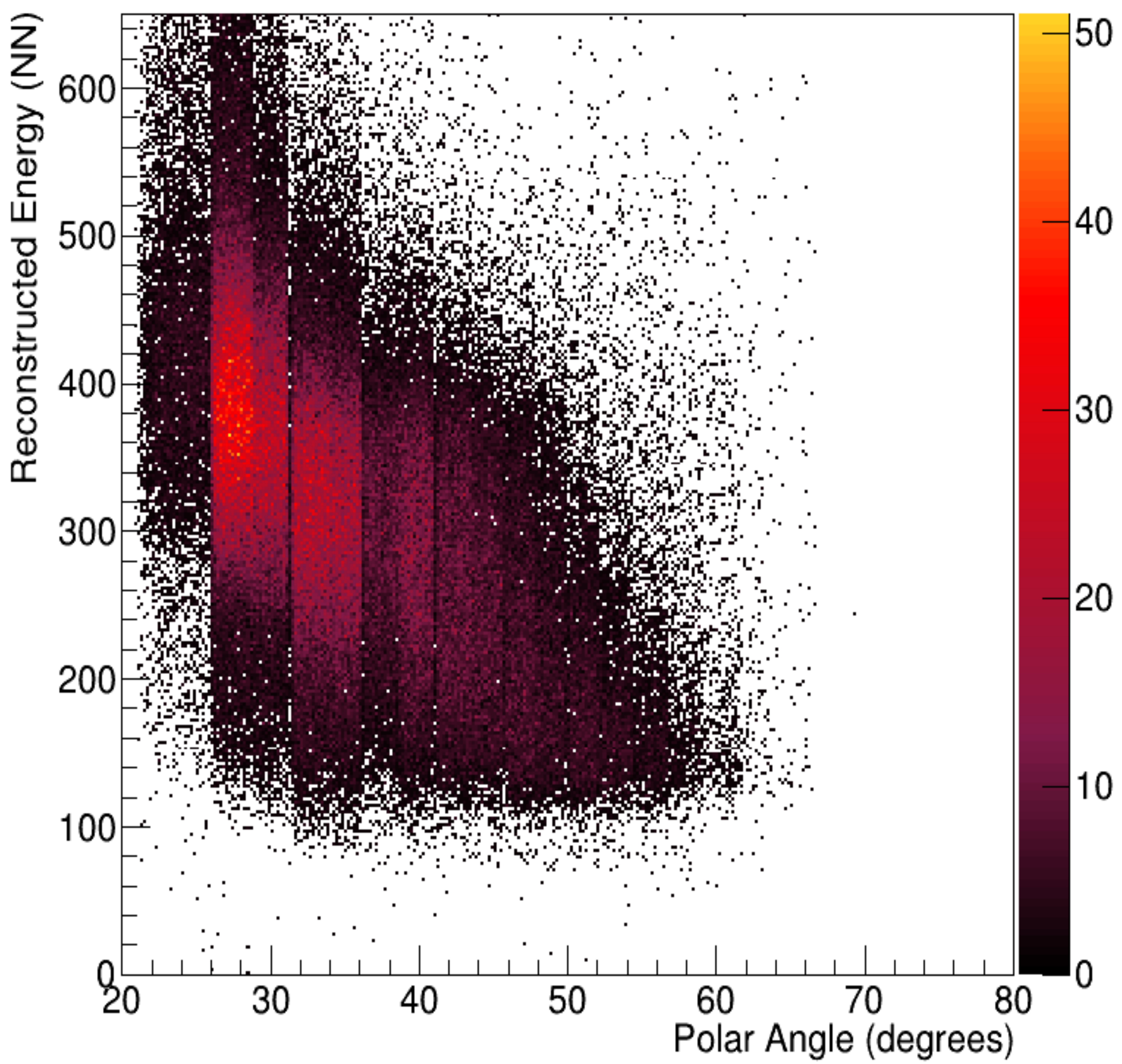
Knockout Reconstruction: Kinematics


Traditional Reconstruction

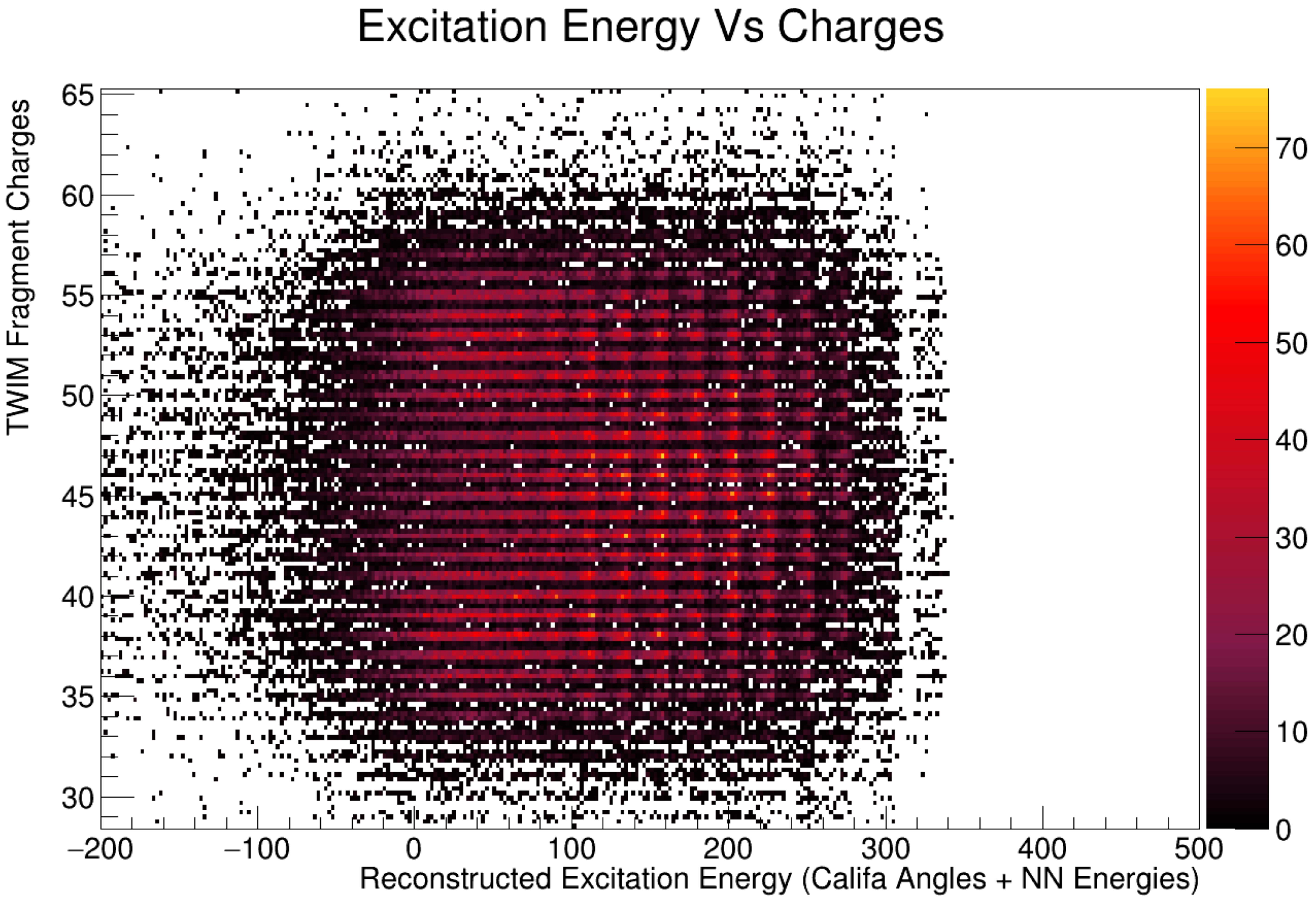

Energy Deposition for Crystal Type 1233

Knockout Reconstruction: Real Data

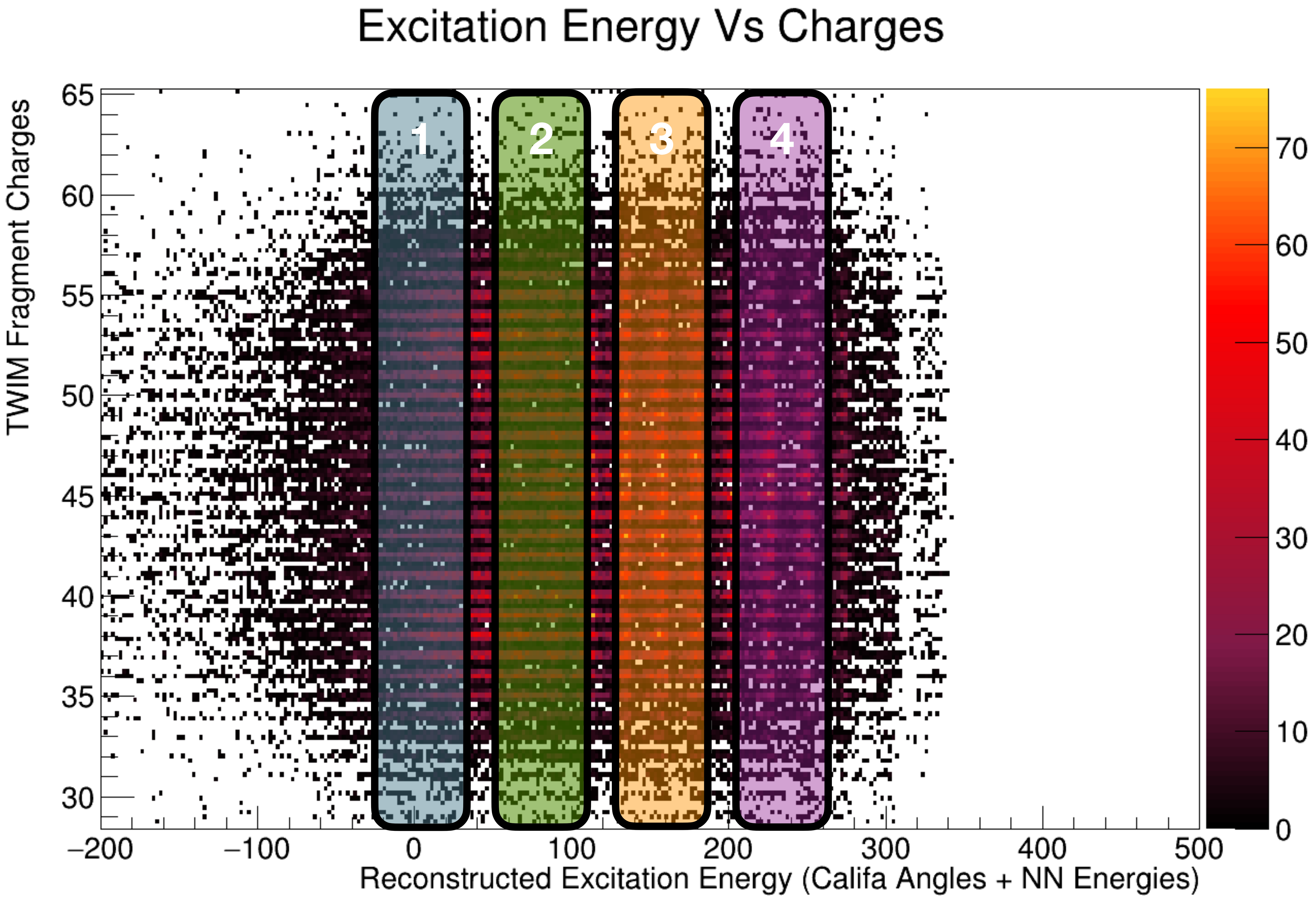

Califa Cluster Kinematics (Real Data)

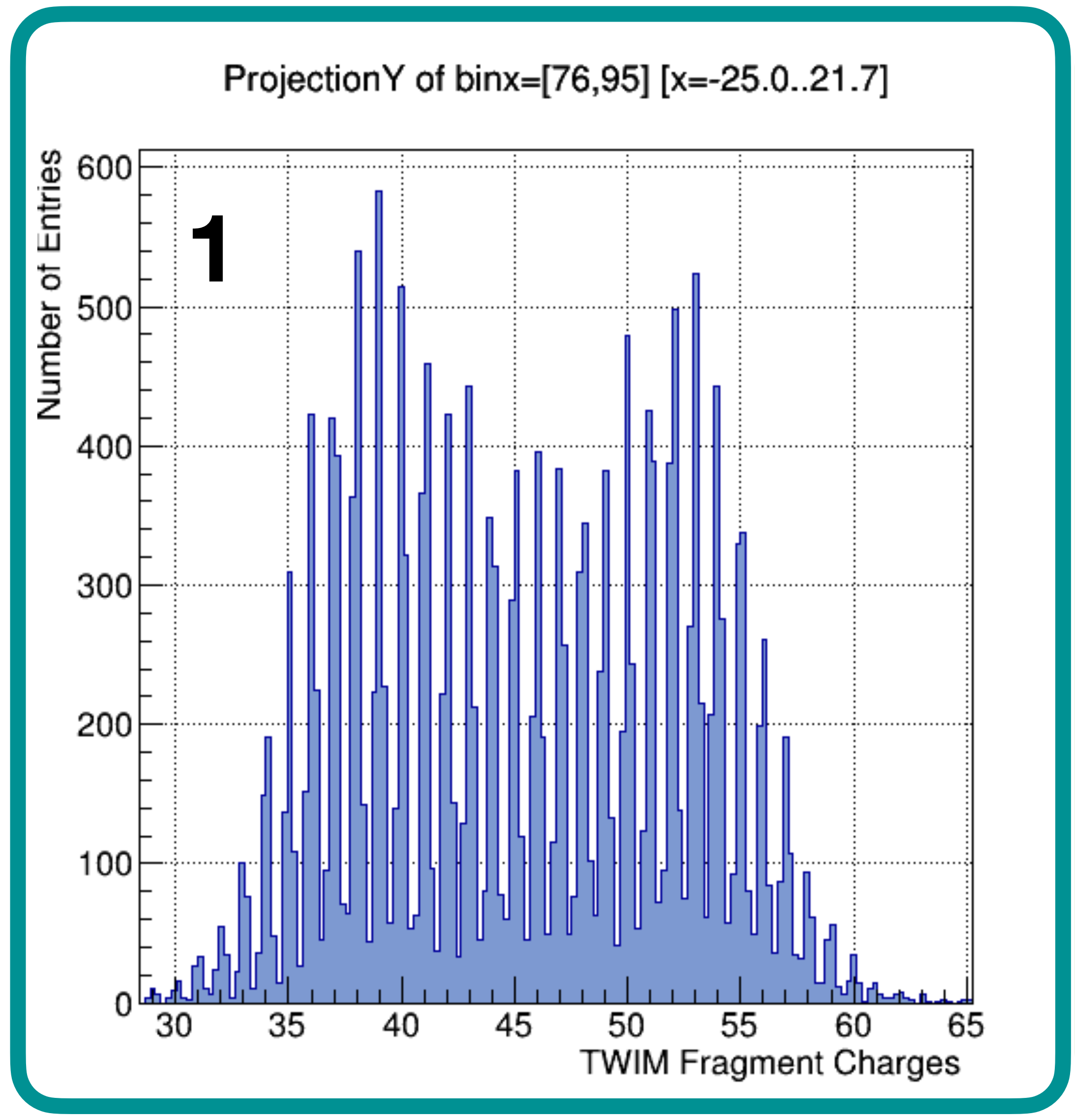

NN Kinematics (Real Data)

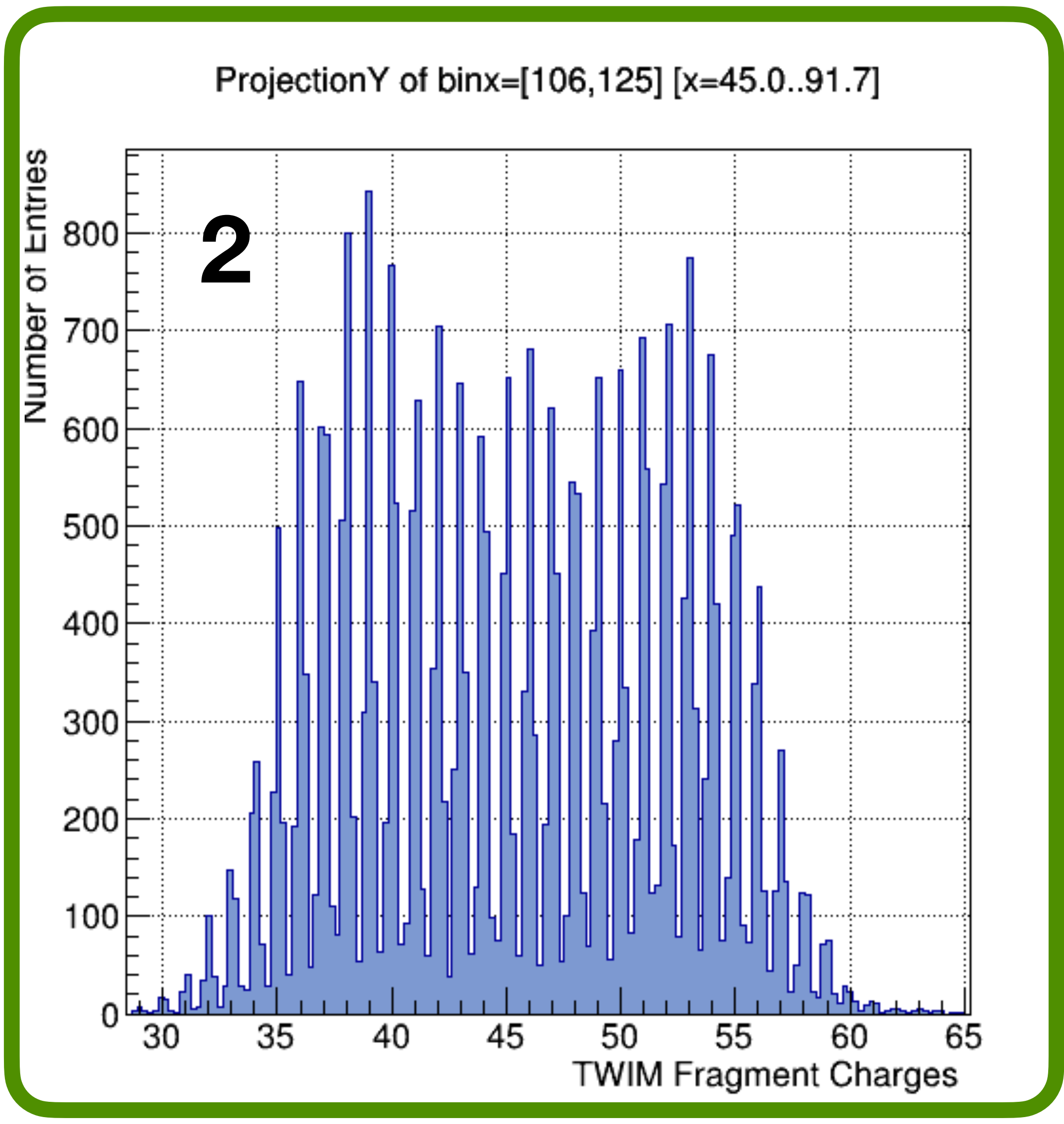
Knockout Reconstruction: Excitation Energies

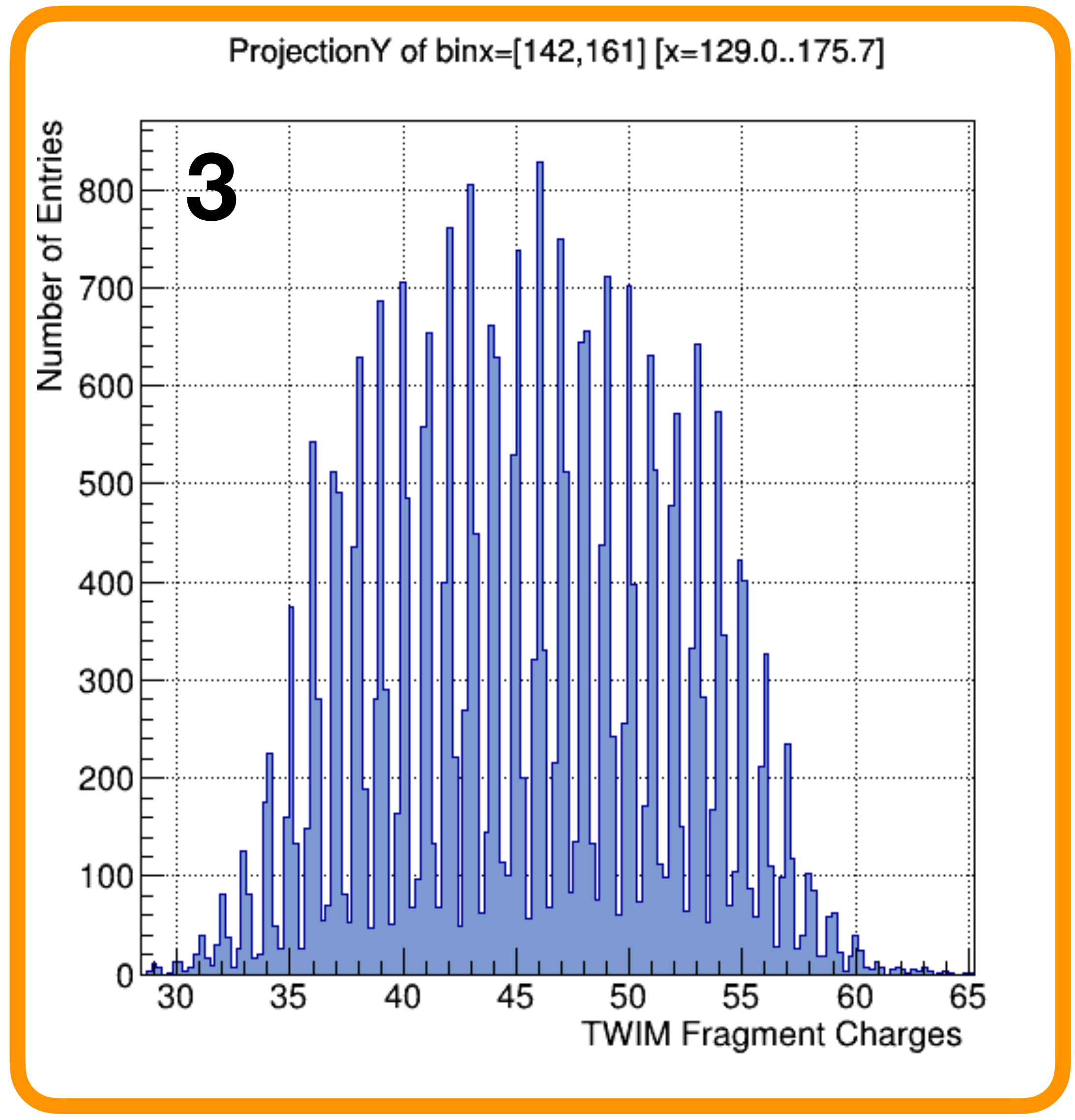


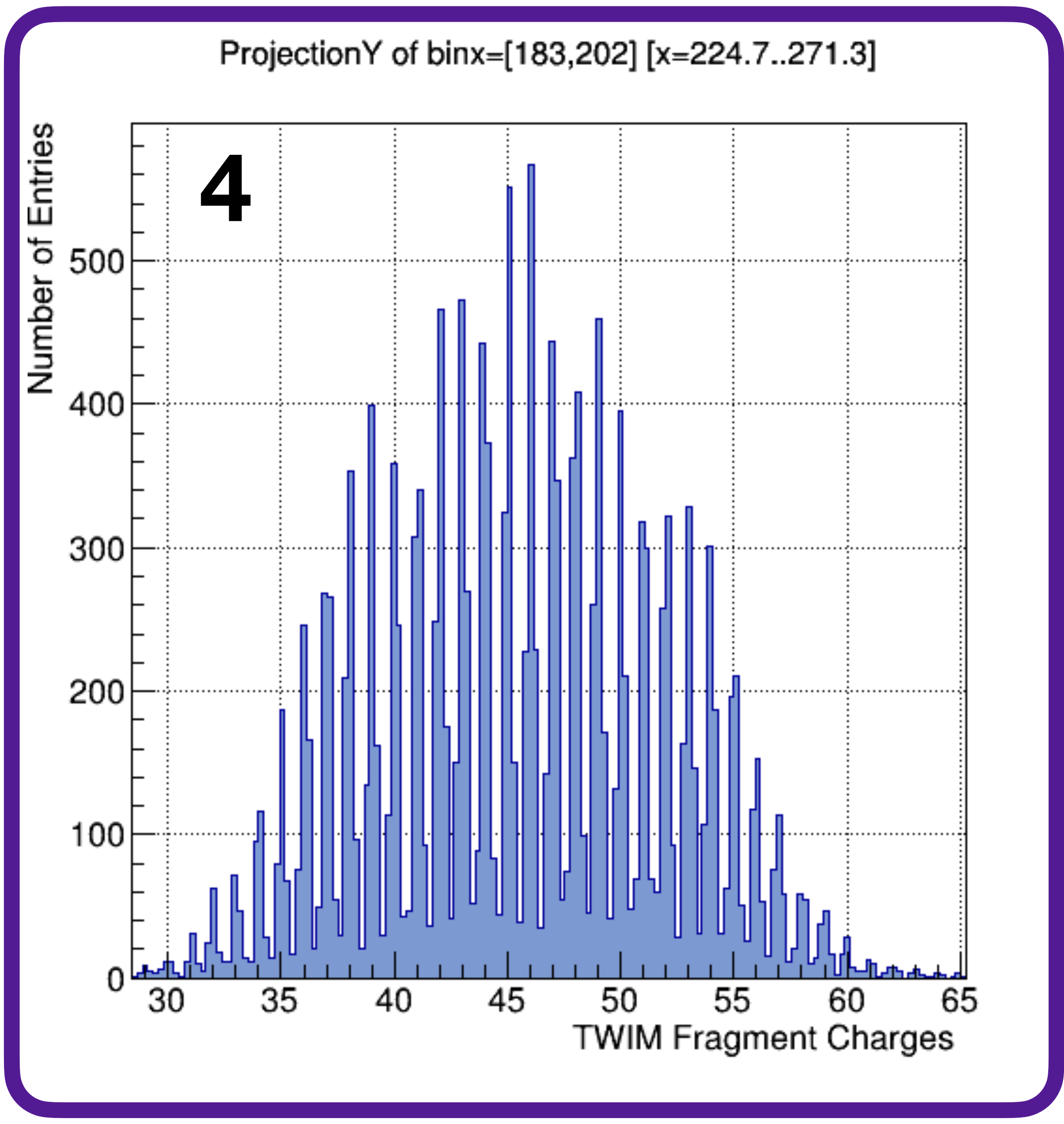
Knockout Reconstruction: Excitation Energies











Future Perspectives

1. Test more channels and compare with real data (outgoing, proton runs 2020)
2. Study dependence on physics model.
3. Compare classification with Nf-Ns (non trivial)
4. Implement convolutions!
5. Autoencoders for simulations.
6. Gammas.