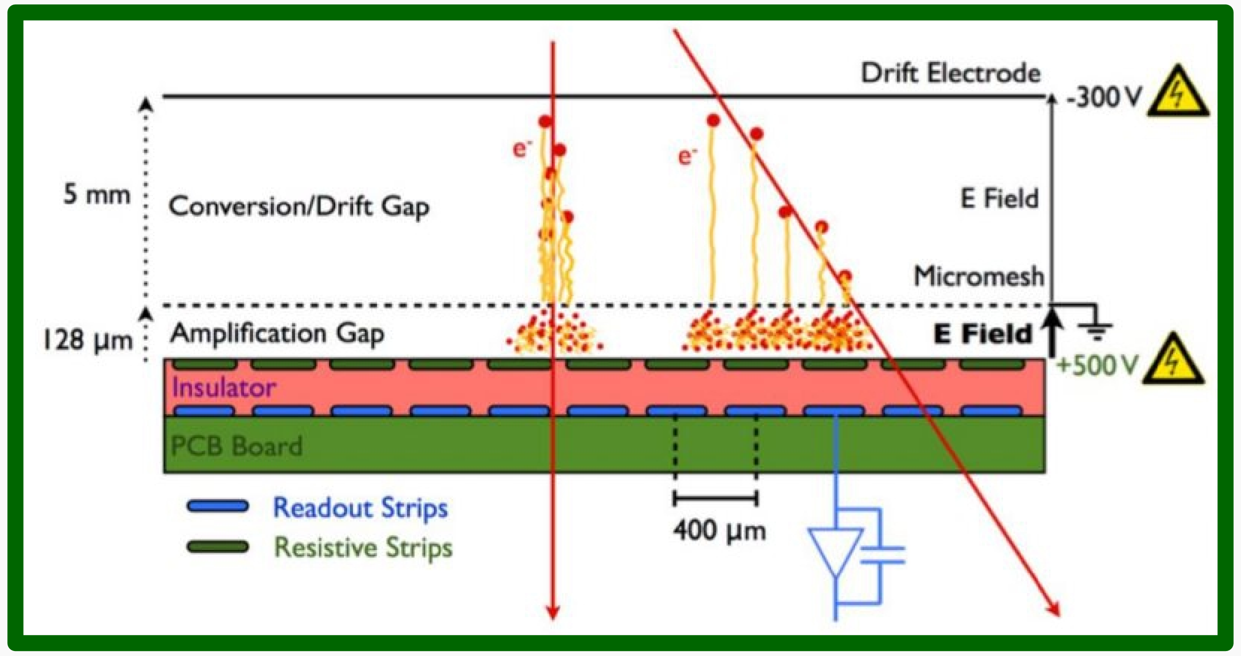
Particle Track Analysis using Neural Networks



WEEKLY HARDWARE MEETING

Roman Lorenz

19 September 2024

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

The datasets and neural network

4 datasets for track inclination of 29° :

- Amplification voltage 520V/530V and peaking time 100ns/200ns.
- ~150 000 events per dataset
- Split in 80% for training and 20% for validation

Events with >20 strips (1.9% of data) were discarded in this try to avoid too much 0-padding

Neural network with 3 hidden layers (300+200+100) :

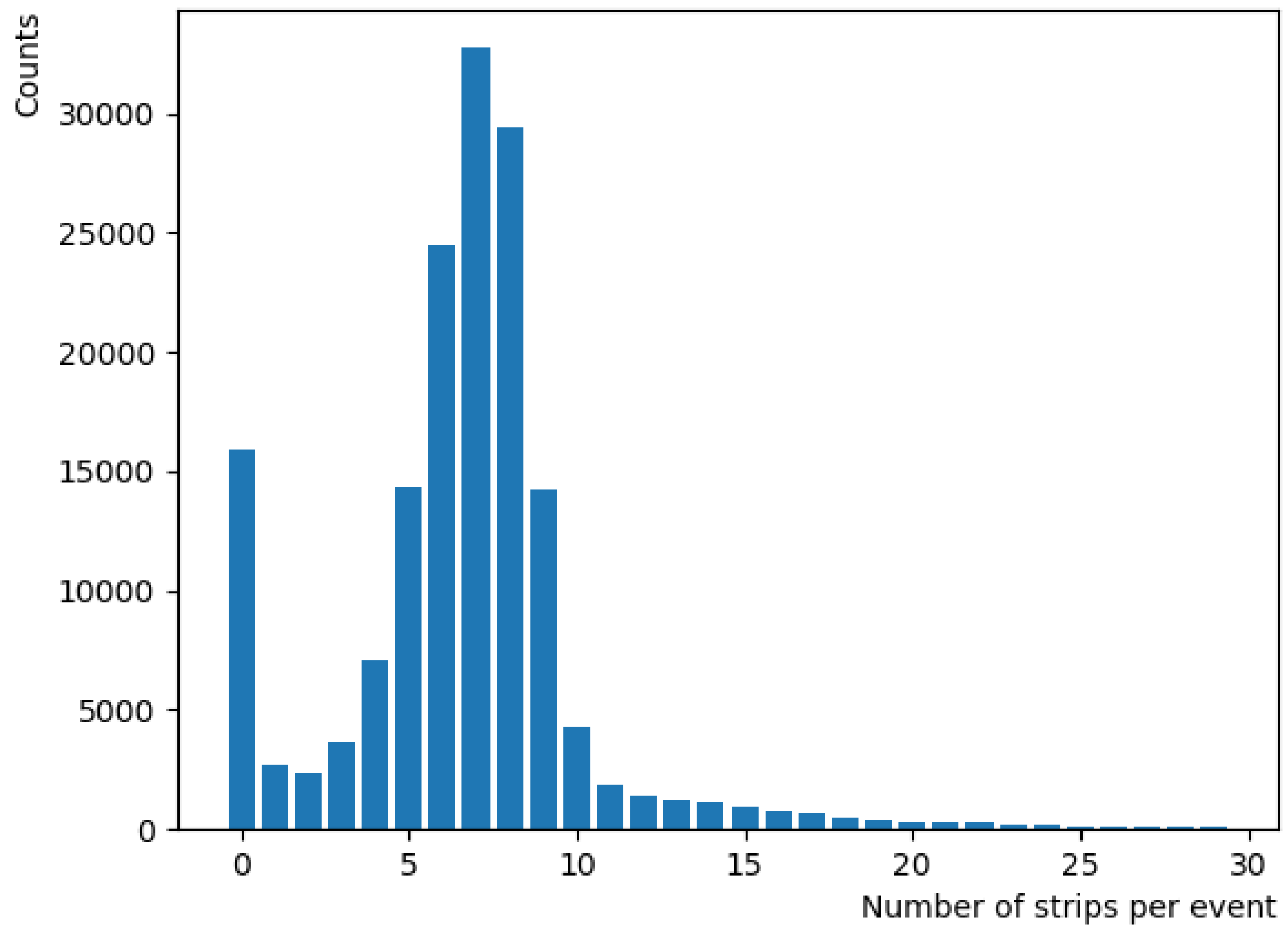
Epochs : 500-1000

Optimizer : Adam

Learning rate : 0.01 to 0.0001

Plan :

1) Train a single NN for each dataset and compare how each NN reconstructs each dataset
2) Train a NN with the whole data and compare how it reconstructs each dataset.



Distribution of the number of strips per event.

4 different neural networks :

Model : Dataset :	520V 100ns		520V 200ns		530V 100ns		530V 200ns	
	Single Gaussian :	Double Gaussian :	Single Gaussian :	Double Gaussian :	Single Gaussian :	Double Gaussian :	Single Gaussian :	Double Gaussian :
520V 100ns	Efficiency : 89.0%		Efficiency : 87.6%		Efficiency : 88.1%		Efficiency : 87.1%	
	σ = 175 μm	σ_1 = 143 μm σ_2 = 348 μm	σ = 171 μm	σ_1 = 147 μm σ_2 = 392 μm	σ = 173 μm	σ_1 = 143 μm σ_2 = 362 μm	σ = 185 μm	σ_1 = 153 μm σ_2 = 368 μm
520V 200ns	Efficiency : 90.1%		Efficiency : 90.0%		Efficiency : 90.5%		Efficiency : 89.7%	
	σ = 200 μm	σ_1 = 167 μm σ_2 = 431 μm	σ = 186 μm	σ_1 = 164 μm σ_2 = 506 μm	σ = 196 μm	σ_1 = 161 μm σ_2 = 404 μm	σ = 198 μm	σ_1 = 163 μm σ_2 = 409 μm
530V 100ns	Efficiency : 88.7%		Efficiency : 88.5%		Efficiency : 89,2%		Efficiency : 88.4%	
	σ = 170 μm	σ_1 = 139 μm σ_2 = 309 μm	σ = 163 μm	σ_1 = 142 μm σ_2 = 378 μm	σ = 164 μm	σ_1 = 136 μm σ_2 = 332 μm	σ = 170 μm	σ_1 = 143 μm σ_2 = 345 μm
530V 200ns	Efficiency : 90.4%		Efficiency : 90.5%		Efficiency : 91.1%		Efficiency : 90,2%	
	σ = 189 μm	σ_1 = 165 μm σ_2 = 467 μm	σ = 173 μm	σ_1 = 154 μm σ_2 = 463 μm	σ = 183 μm	σ_1 = 159 μm σ_2 = 450 μm	σ = 180 μm	σ_1 = 148 μm σ_2 = 361 μm

Notes : -Efficiency is here the percentage of event positions reconstructed with the residual in range of -2mm to 2mm.
-When model and dataset match, only 20% of the dataset is used because of the train/validation split.

Higher voltage : better accuracy and often better efficiency
Higher peaking time : better efficiency but worse accuracy

Neural network trained on all datasets

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Datasets :	Efficiency :	Single Gaussian fit	Double Gaussian fit
All 4	92.44%	σ = 179 µm	σ_1 = 149 µm σ_2 = 383 µm
520V 100ns	90.7%	σ = 178 µm	σ_1 = 151 µm σ_2 = 386 µm
520V 200ns	93.1%	σ = 191 µm	σ_1 = 145 µm σ_2 = 329 µm
530V 100ns	92.2%	σ = 167 µm	σ_1 = 138 µm σ_2 = 383 µm
530V 200ns	94.1%	σ = 181 µm	σ_1 = 156 µm σ_2 = 438 µm

Same result than before concerning voltage and peaking time.

Efficiency is higher, this could hint that the neural network would benefit from bigger datasets.

Resolution for each dataset is slightly worse than in the case of individually trained NNs.

Clustering

			Cluster 1						Cluster 2		
Strip position	673,73	674,15	674,58	675	675,43	...	982,28	982,7	983,13	983,55	983,98
Strip charge	100	108	124	183	128	...	73	122	106	87	74
Strip timing	36,8	53,4	51,2	77,5	67,6	...	11,3	23,9	29,3	36,3	62,1

Example of an event that requires clustering

Attempts with conventional algorithms :

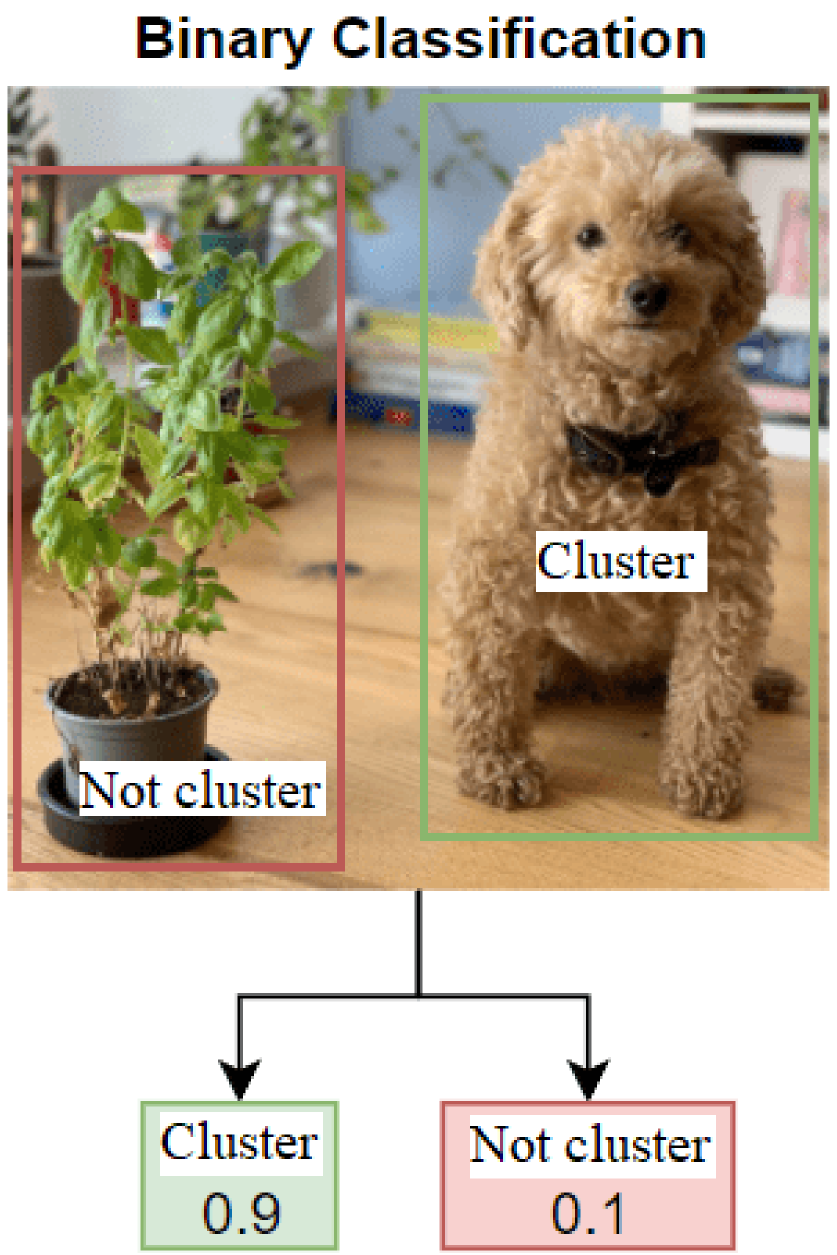
Choose cluster with most strips, with 0 or 1 gap (missing strip) allowed : ❌

Choose cluster with highest total or mean charge : ❌

Discard clusters with -100ns < timing < 200ns : ❌

Other desperates attempts : ❌

Clustering with classification neural network ?



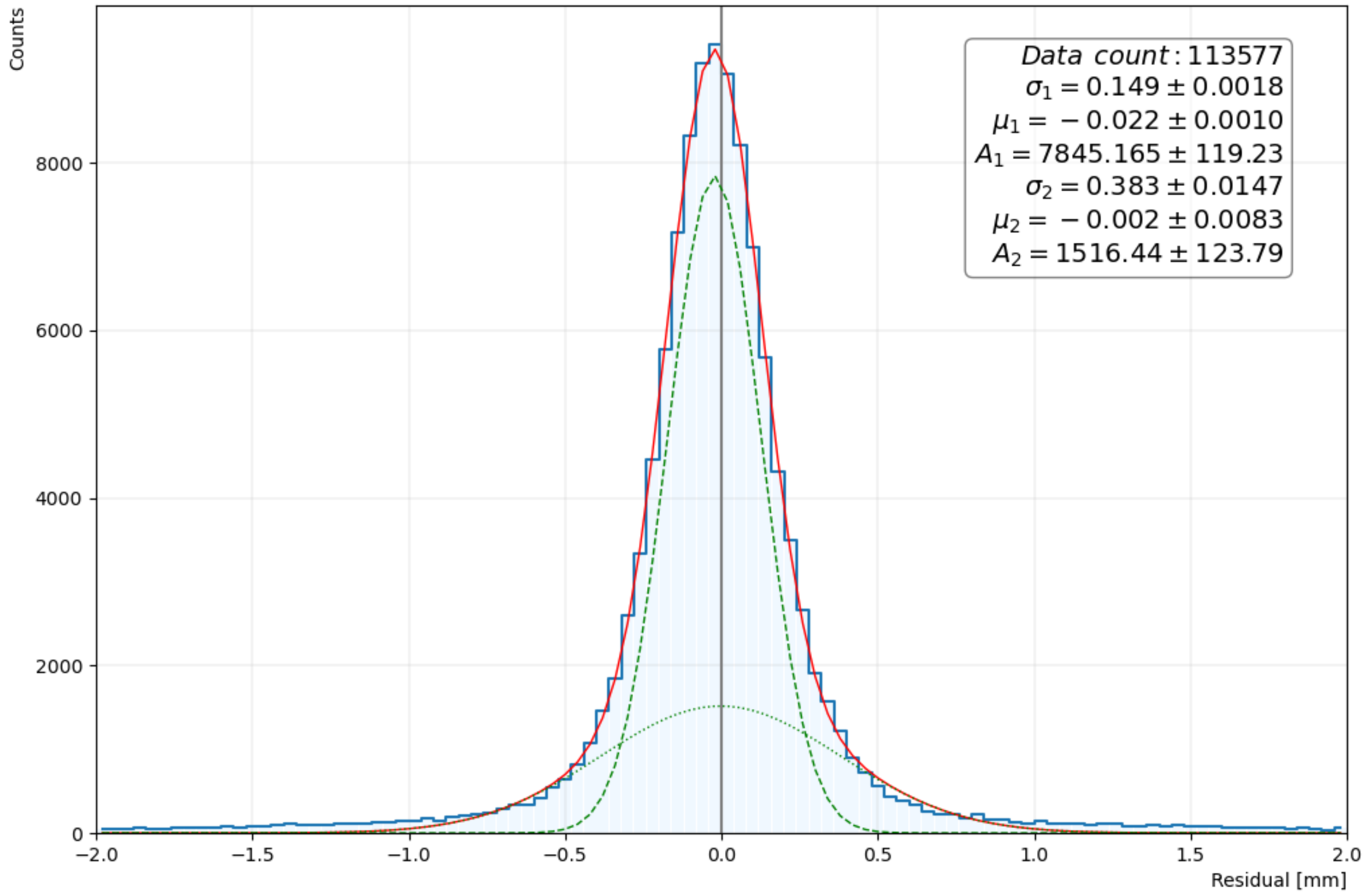
Modified image from mathworks.com (Source original file)

Train a classification neural network :

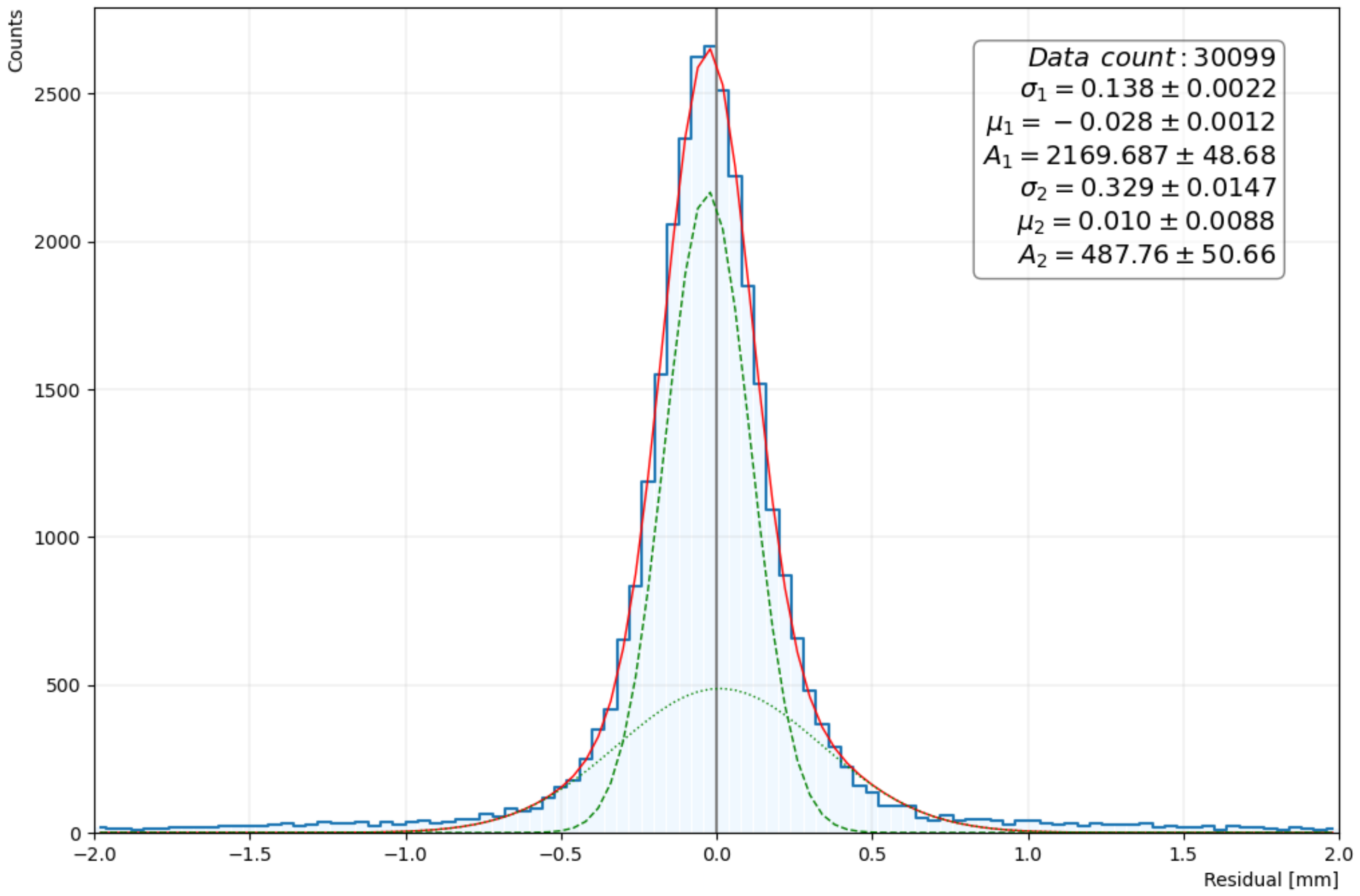
First idea :
Similar to image recognition, it could be trained to form clusters of varying size with a likelihood value between 0 and 1.
But : data may lack features compared to images with millions of pixels.

Second idea :
Classification that gives a likelihood for each strip to be in the correct cluster.
Followed by a conventional algorithm to form the cluster with the most likelihood.

Backup : Plots on NN trained on all datasets

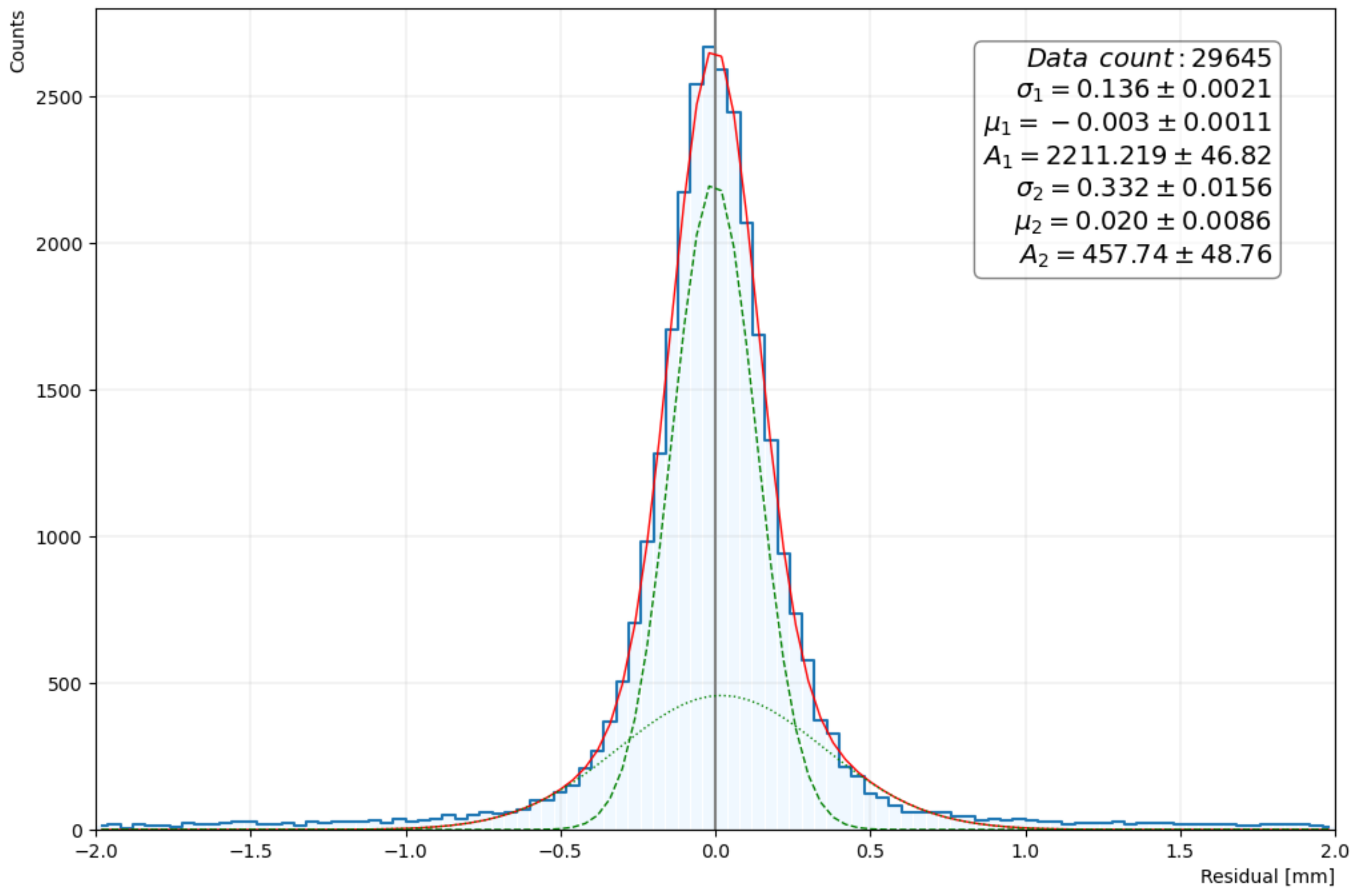


Reconstructed position residuals for the 4 datasets in the NN trained on the 4 datasets
Chi^2 / Ndf (-0.8 to 0.8mm) = 217.5284 / 34
Efficiency (-2mm to 2mm) : 92.4%

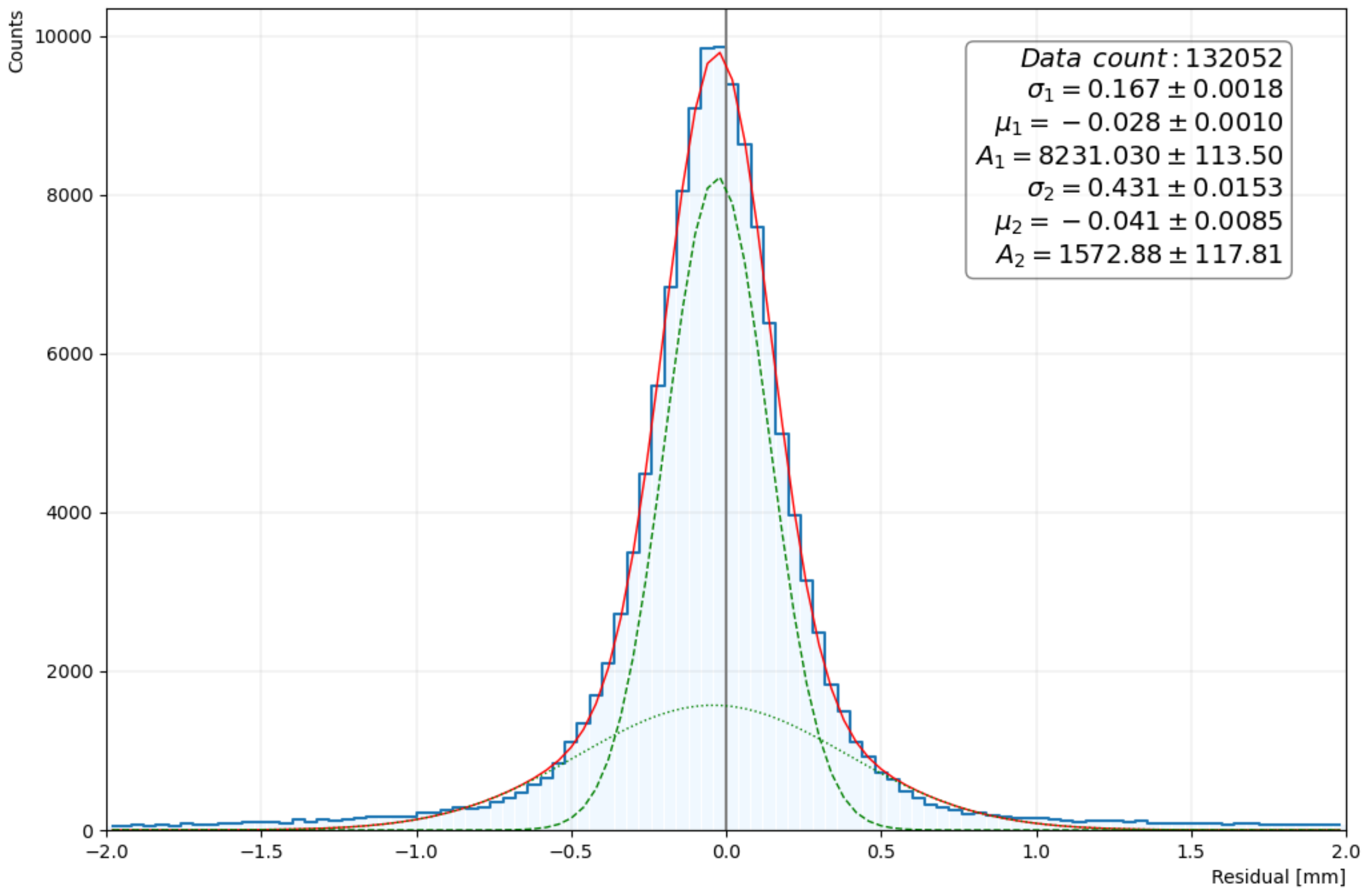


Reconstructed position residuals for the dataset 530V100ns in the NN trained on the 4 datasets
Chi^2 / Ndf (-0.8 to 0.8mm) = 112.2575 / 34
Efficiency (-2mm to 2mm) : 92.2%

Backup : Plots on NN trained on single dataset



Reconstructed position residuals for the dataset 530V100ns in the NN trained on the same dataset
Best result in term of σ
Chi^2 / Ndf (-0.8 to 0.8mm) = 130.4580 / 34
Efficiency (-2mm to 2mm) : 89,2%



Reconstructed position residuals for the dataset 520V200ns in the NN trained on the dataset 520V100ns
Worst result in term of σ
Chi^2 / Ndf (-0.8 to 0.8mm) = 228.9782 / 34
Efficiency (-2mm to 2mm) : 90,1%

Backup : Question about Chi² goodness of fit

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

Oi = an observed count for bin i
Ei = an expected count for bin i