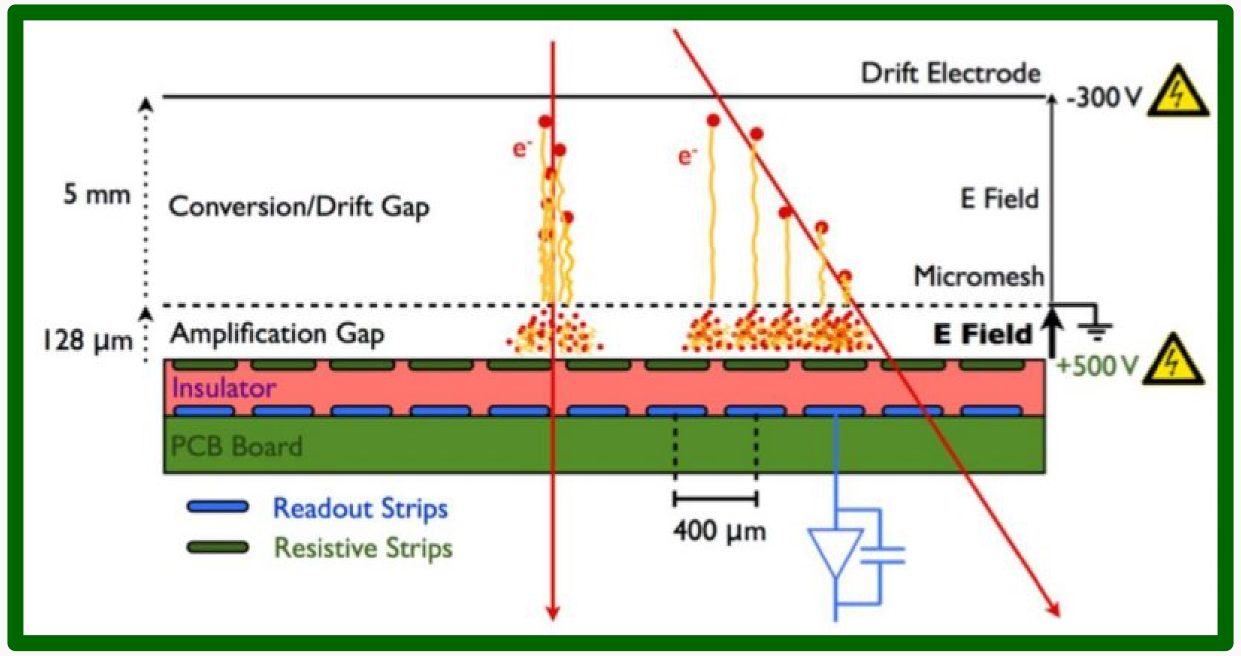
Particle Track Analysis using Neural Networks

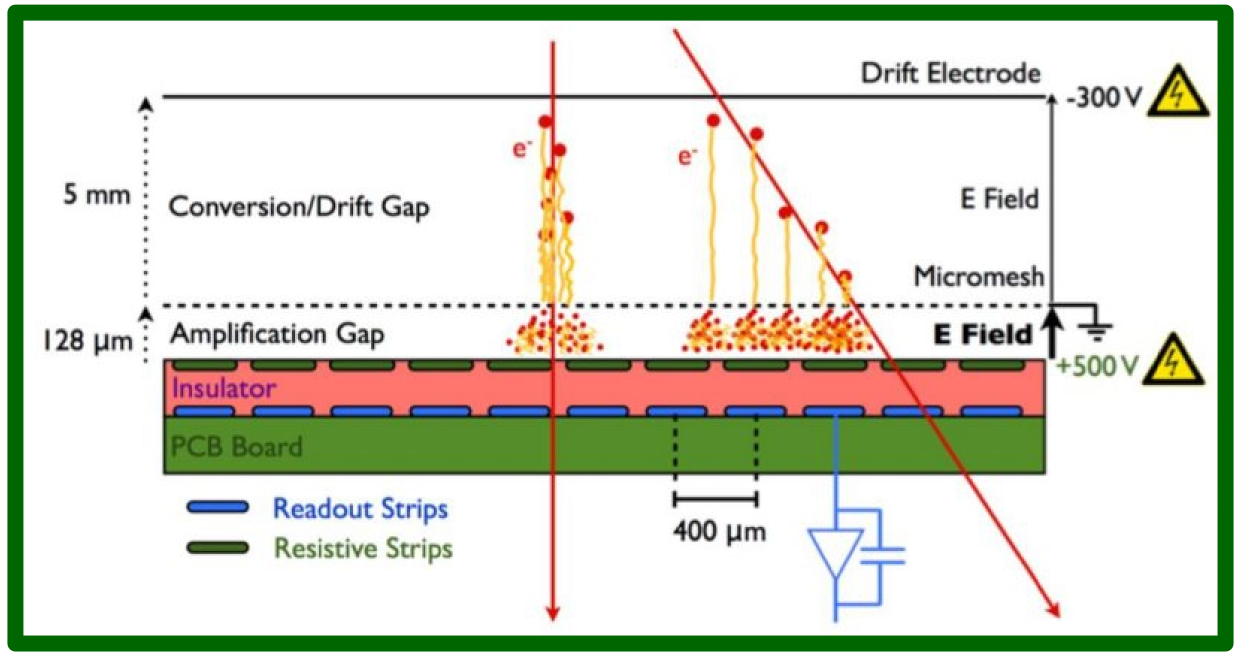


Joint Particle Physics Group Seminar

Master Thesis Roman Lorenz
6 November 2024

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Goal of the thesis :



Use ANNs (artificial neural networks) to improve the results of position reconstruction in Atlas Muon Spectrometer for particle with higher inclination.
Actual results in the group : core resolution for vertical tracks is <100µm while for inclined tracks >100µm

Where I started :
Some python knowledge from courses/projects
Zero machine learning knowledge

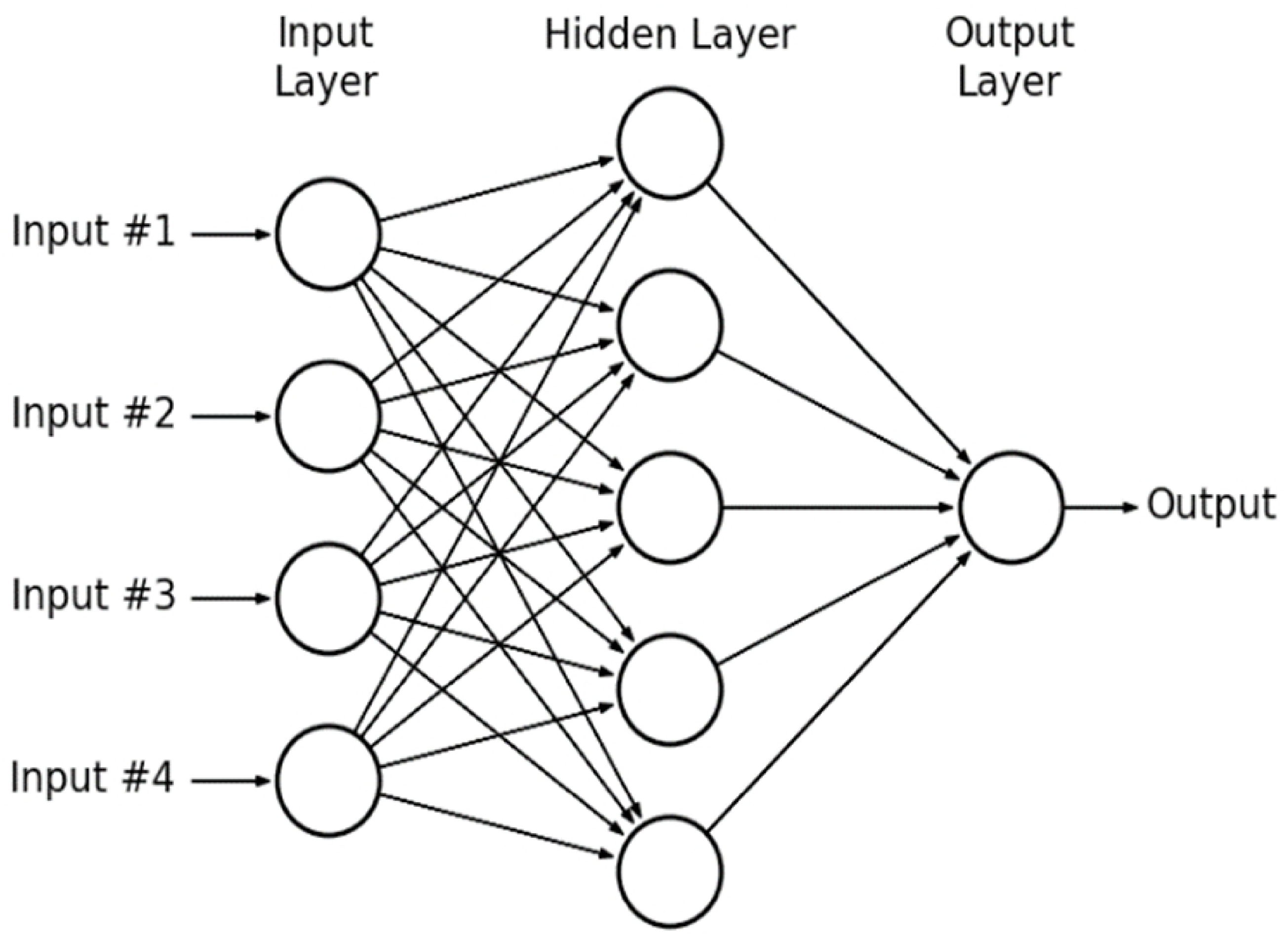
First attempt : Regression with MLP

Row	Instance	out_charg	out_xpos	out_time	out_zPos	out_layer	out_track	out_track	out_non_p
0	0	183	675.0035	57.547012	1939.2	6	4.059e-05	532.63472	839.78904
0	1	457	675.4285	83.191697	1939.2	6	4.059e-05	532.63472	839.78904
0	2	480	675.8535	97.671507	1939.2	6	4.059e-05	532.63472	839.78904
0	3	225	676.2785	105.45331	1939.2	6	4.059e-05	532.63472	839.78904
0	4	96	676.7035	131.59313	1939.2	6	4.059e-05	532.63472	839.78904
0	5	78	784.2285	105.78648	1939.2	6	4.059e-05	532.63472	839.78904
1	0	112	942.7535	19.017442	1939.2	6	-0.000626	759.10063	912.70048
1	1	222	943.1785	13.385013	1939.2	6	-0.000626	759.10063	912.70048
1	2	93	943.6035	38.426794	1939.2	6	-0.000626	759.10063	912.70048
1	3	101	944.0285	37.585999	1939.2	6	-0.000626	759.10063	912.70048
1	4	112	945.3035	113.60661	1939.2	6	-0.000626	759.10063	912.70048
1	5	161	945.7285	109.47327	1939.2	6	-0.000626	759.10063	912.70048
1	6	98	946.1535	139.83771	1939.2	6	-0.000626	759.10063	912.70048

How the data looks like

Testbeam data for inclination of ~29 degrees of an Eta layer with ~150 000 events
Particle track is known (with resolution ~50µm) allowing for supervised learning

First attempt : Regression with MLP



Example of multilayer perceptron with one hidden layer and 5 hidden neurons

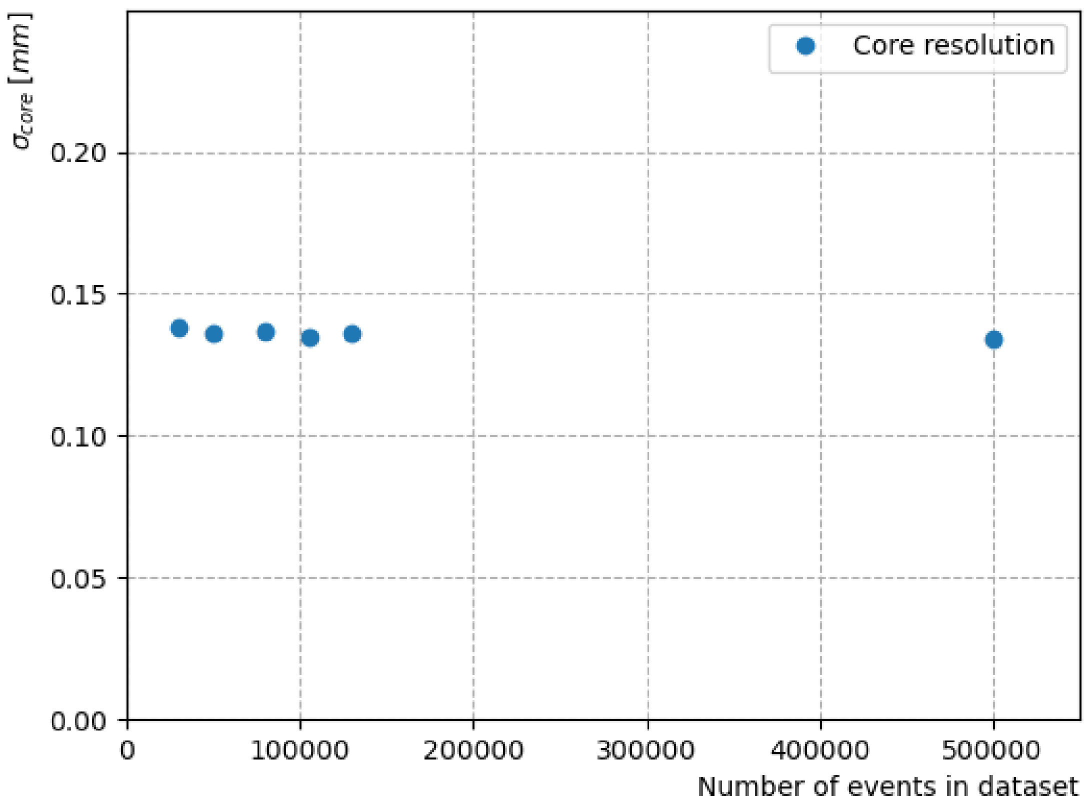
First idea :

Simple NN trained with strips positions/charges/timings as Input and particle position as Output

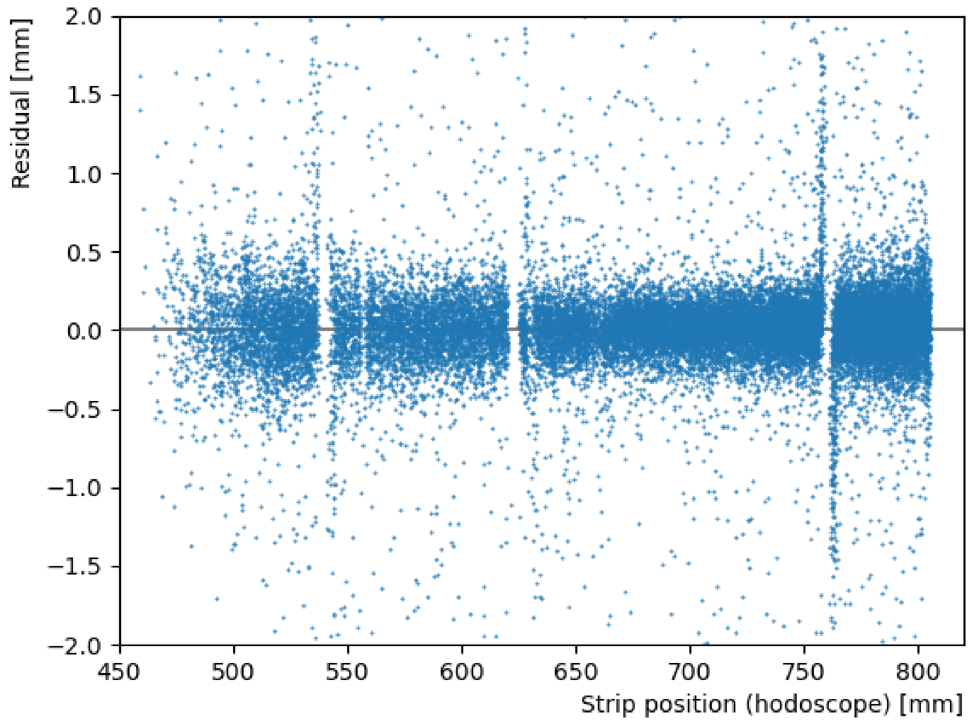
0-padding to deal with varying Input size

Use the full data without performing clustering

First attempt : Results



Core resolution for MLPs trained on datasets with difference sizes

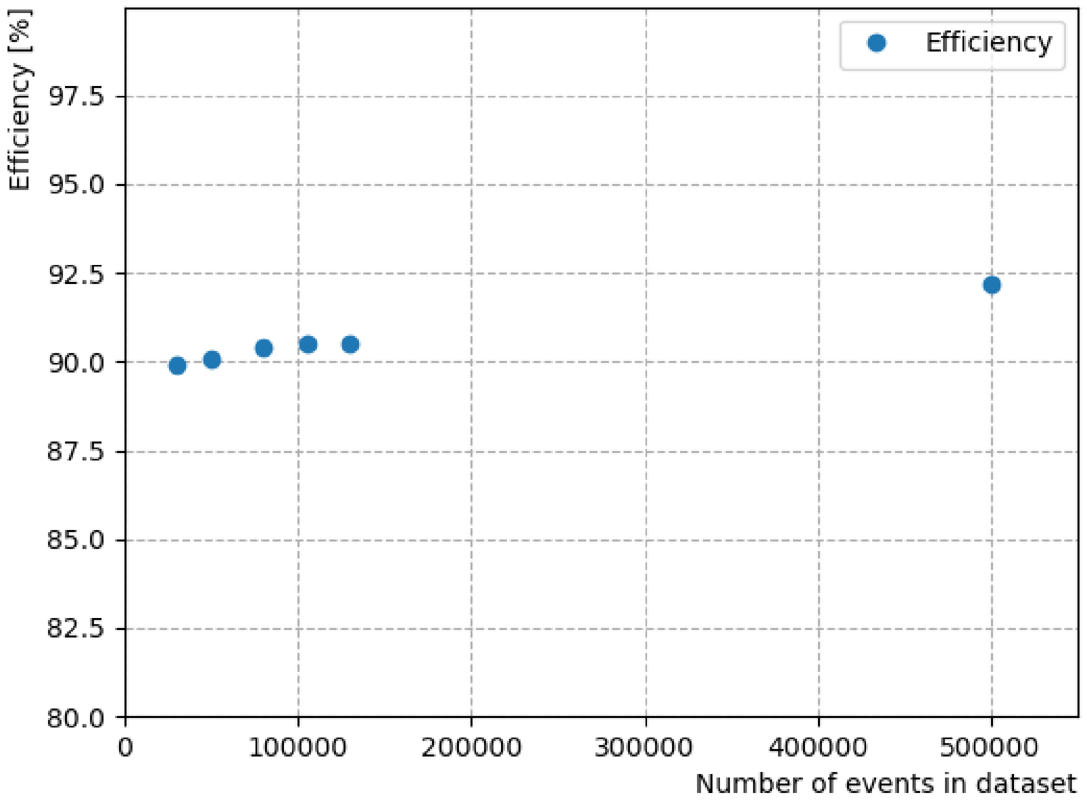


Example of residuals plotted against strip position

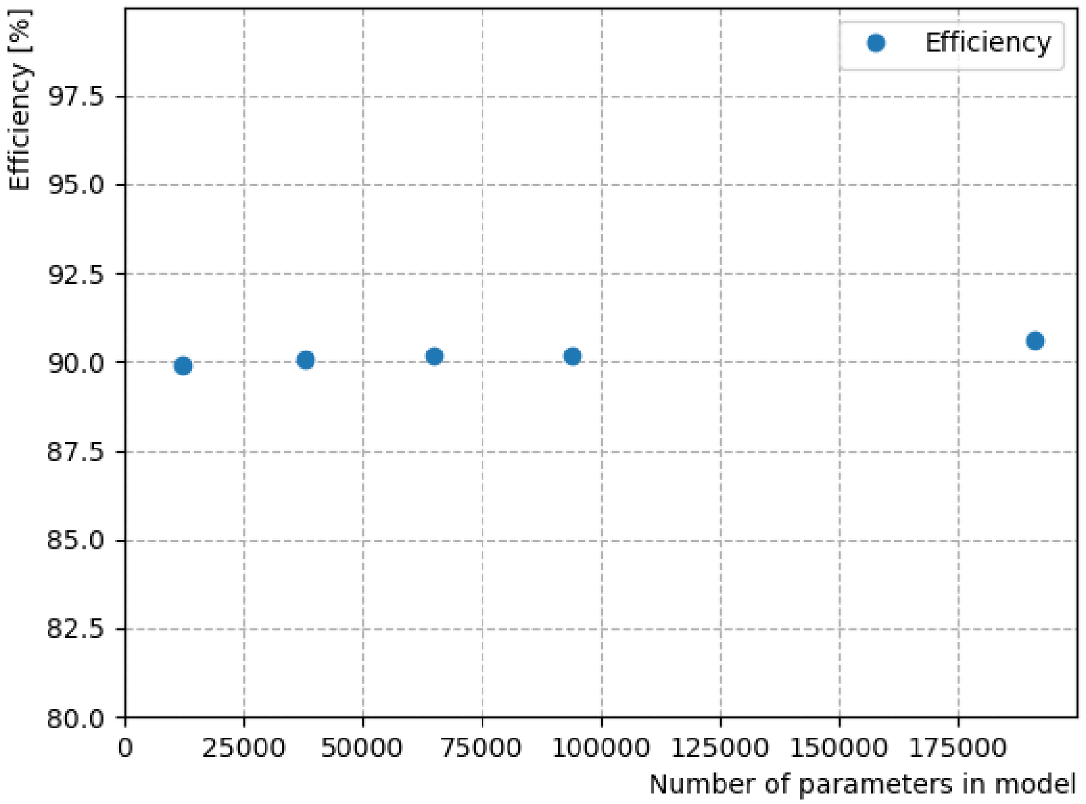
Core resolution around 135µm, similar to previous results in group

Increasing the # of events in the training dataset or the # of parameters in the model doesn't increase accuracy

First attempt : Results



Efficiency for MLPs with 90k trainable parameters on datasets with difference sizes



Efficiency for MLPs with different number of trainable parameters

Efficiency : % of events where the MLP reconstructs position with with an error <2mm

The efficiency gets better with bigger datasets but hits the same wall than previous results in the group

The efficiency gets slightly better with more parameters in the model but it seems to be loosing accuracy

Where does MLP with more parameters gain accuracy ?

	Cluster 1					Cluster 2						Cluster 3							
Strip pos (mm)	728,2	728,6	729,0	729,3	...	741,2	741,6	742,0	742,4	742,7	743,1	...	745,3	745,7	746,1	746,4	746,8	747,2	747,6	747,9
Charge	203	260	102	116	...	252	131	76	92	83	91	...	110	187	274	388	274	283	103	109
Timing (ns)	94	86	54	37	...	25	30	39	43	71	67	...	17	-5	27	23	47	52	66	93

Muon

Example of an event where bigger MLP improved efficiency

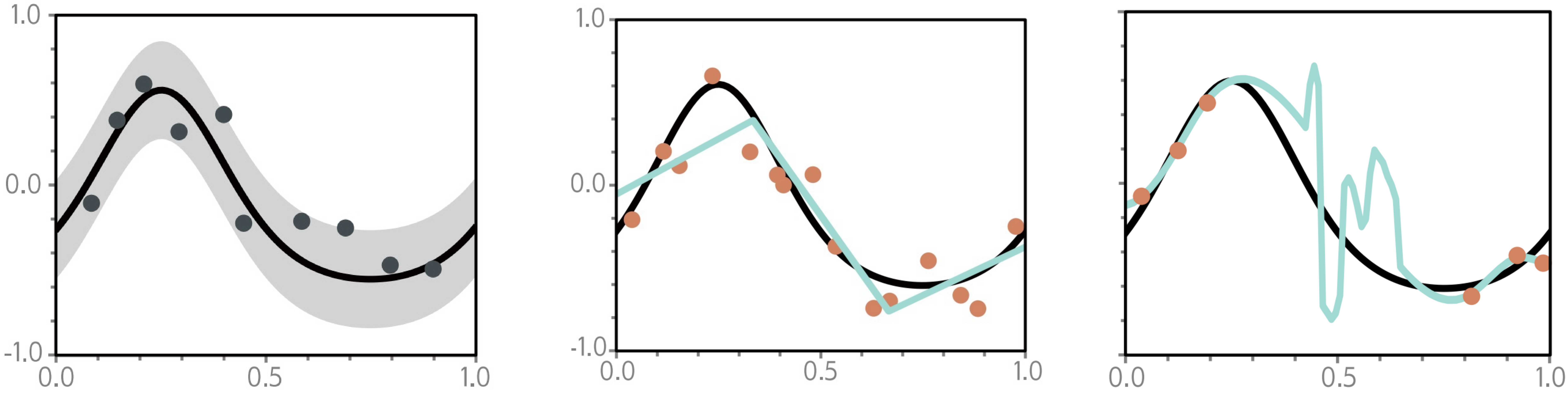
In this event 3 clusters can be build. Cluster 1 and 2 are background and cluster 3 belongs to the particle

MLP manages to completely ignore Cluster 1 and 2 and outputs the same result as if it had only cluster 3 as input

Models with less parameters will use informations from other clusters and output a position >2mm from the truth

Problem : Training models with more parameters increase efficiency but reduce accuracy

Noise, bias and variance



<u>Example of noise :</u>
Black curve is the function we try to measure
Grey dots are the measurements from the detector

Noise error come from the imprecision of the measures

<u>Example of bias :</u>
Cyan curve is the model trying to fit the measurements

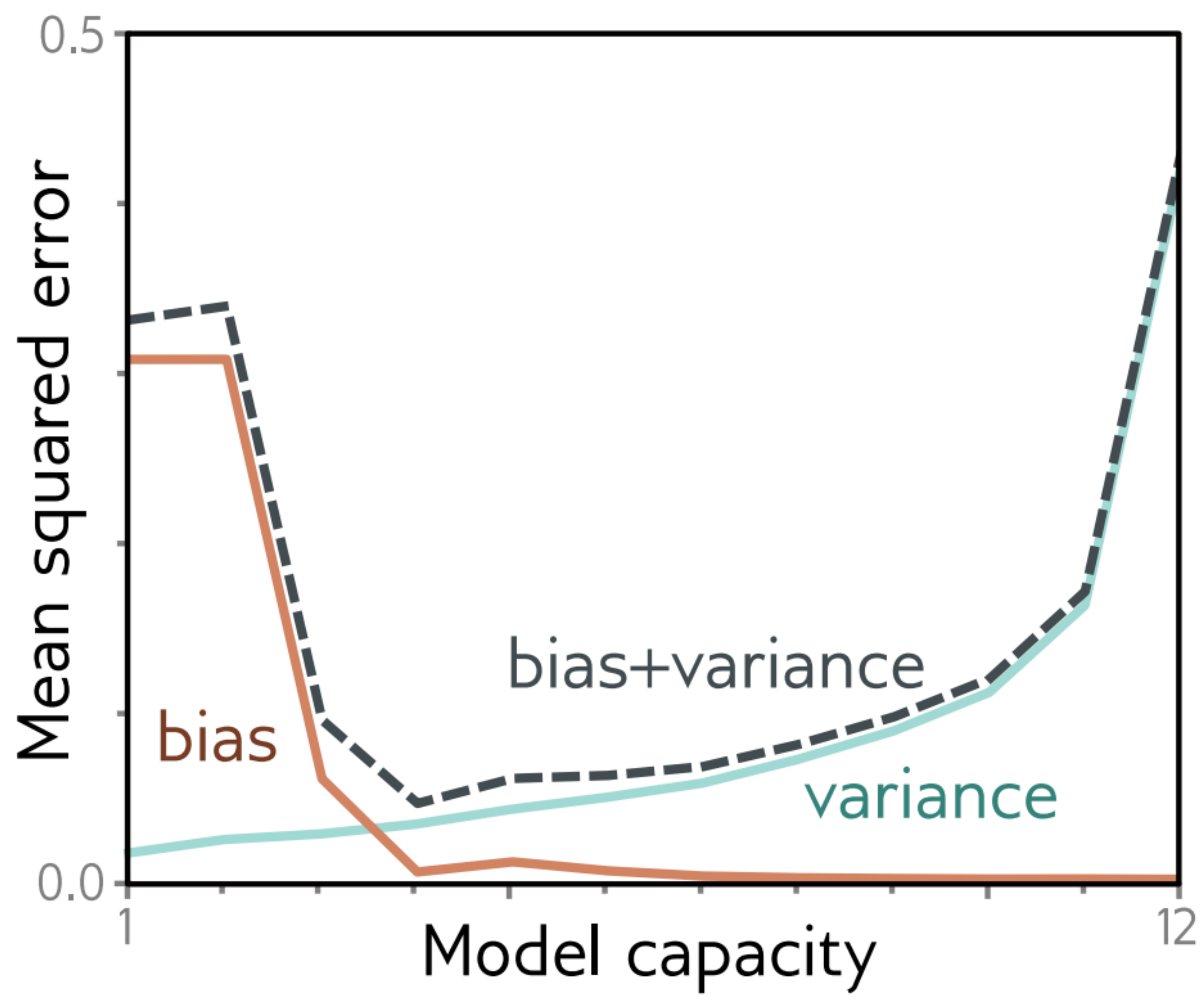
Bias error come from the lack of complexity in the model to fit the data

<u>Example of variance :</u>
Model is complex enough to fit the measurements with its noise

Model overfits the data and its noise leading to big fluctuations in error

Plots from Understanding deep learning , Simon J.D. Prince

Bias variance trade-off



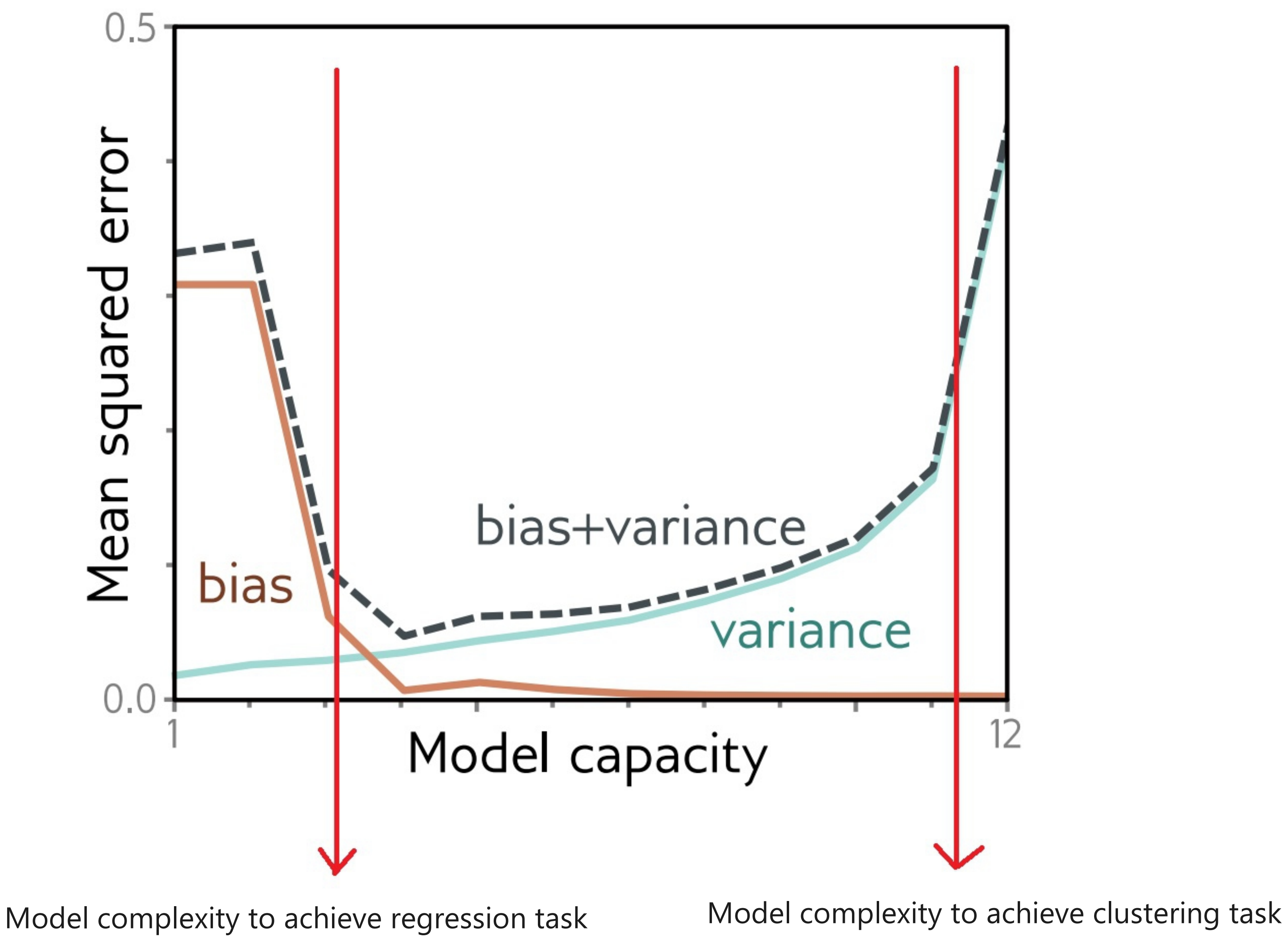
<u>Bias+variance trade-off</u>

For a given problem, upscaling our MLP will first reduce bias and error.

Later, the MLP will become too complex and too keen to overfit the data producing variance error

Plot from Understanding deep learning , Simon J.D. Prince

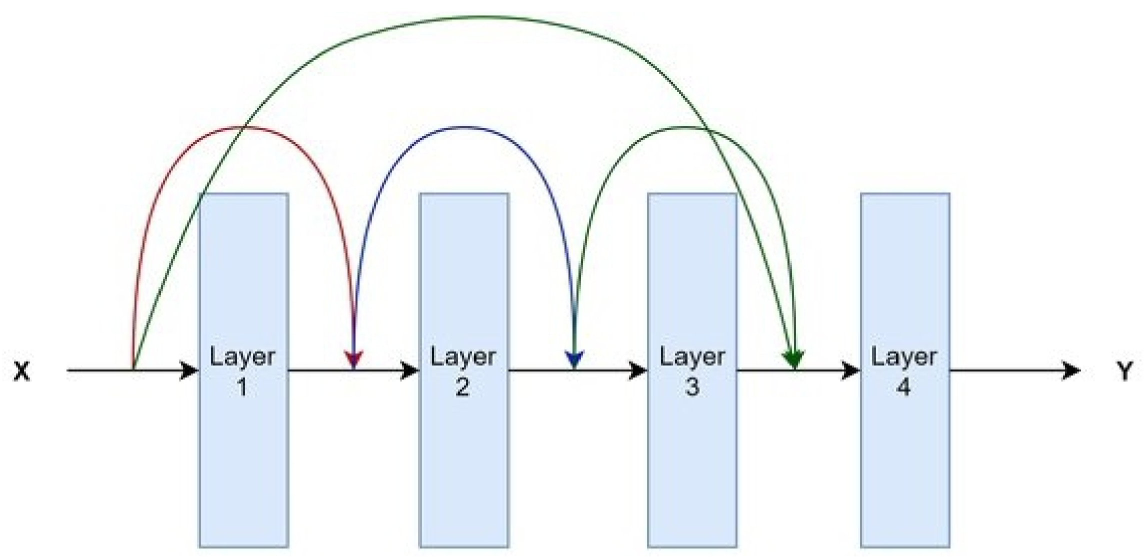
Possible problem of the MLP



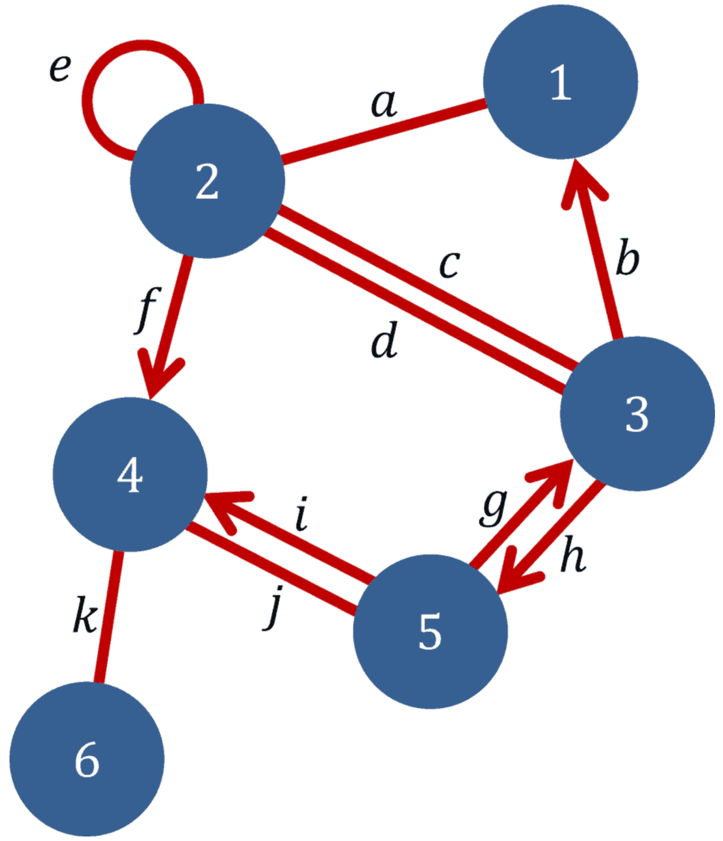
With unclustured events, model has to do at least two tasks :
-The regression task : taking the position/charge/timing of where the particle went through to reconstruct the position
-The clustering task : choosing which measures are relevant to the final computation
Due to a disparity in the complexity of those two tasks, those tasks may need to be done separately

Modified plot from Understanding deep learning , Simon J.D. Prince

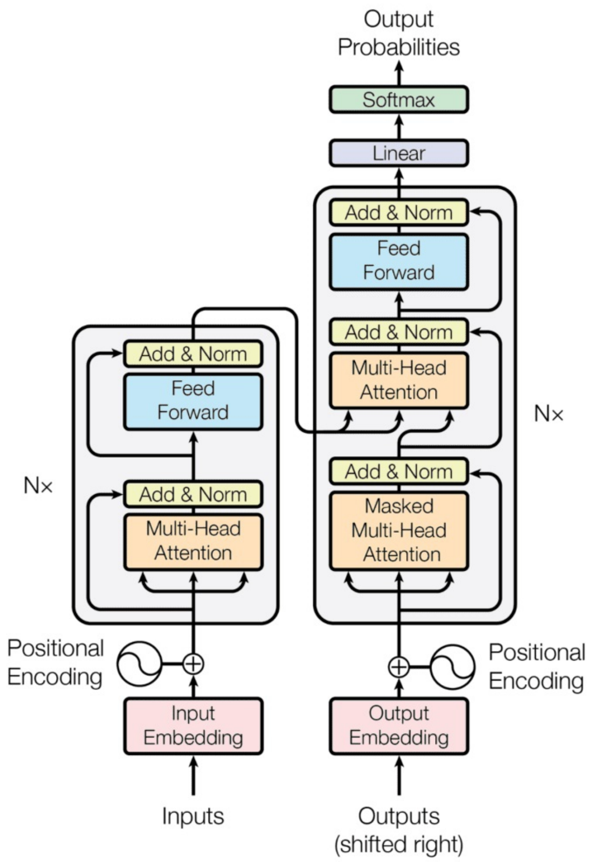
Attempts with more complex models



Deep NN (ResNet) with skip connections



Graph neural network

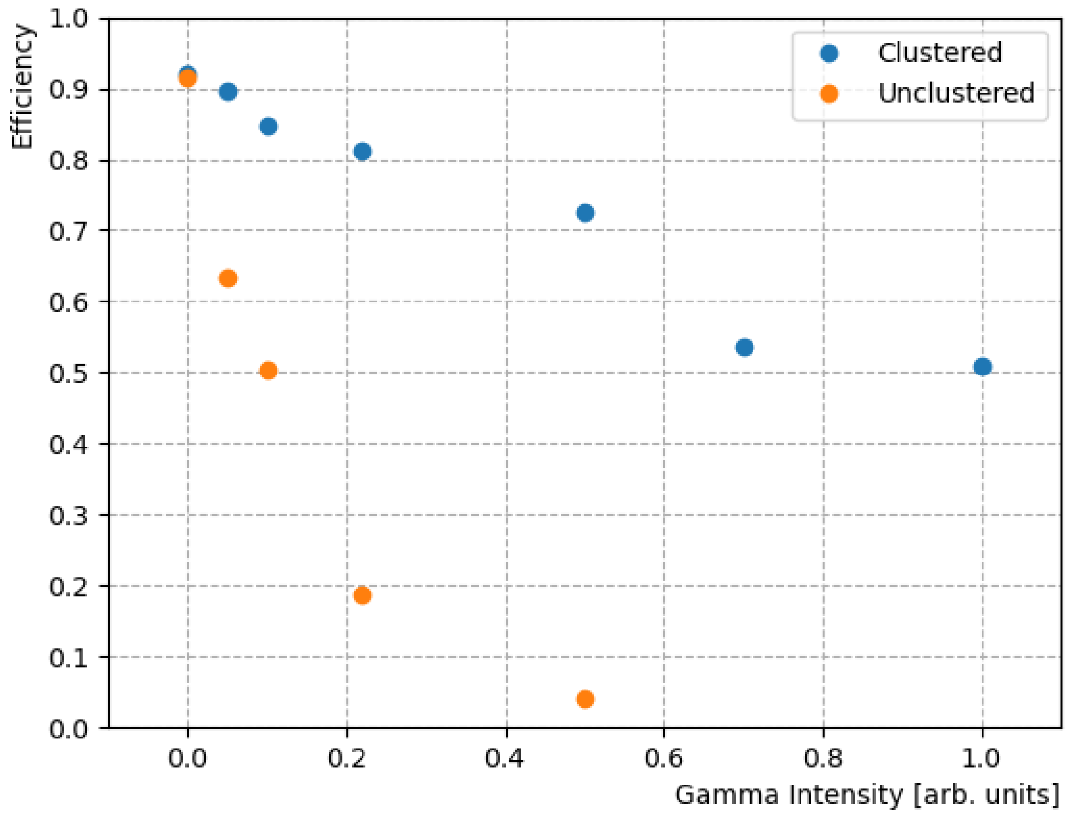


Transformer

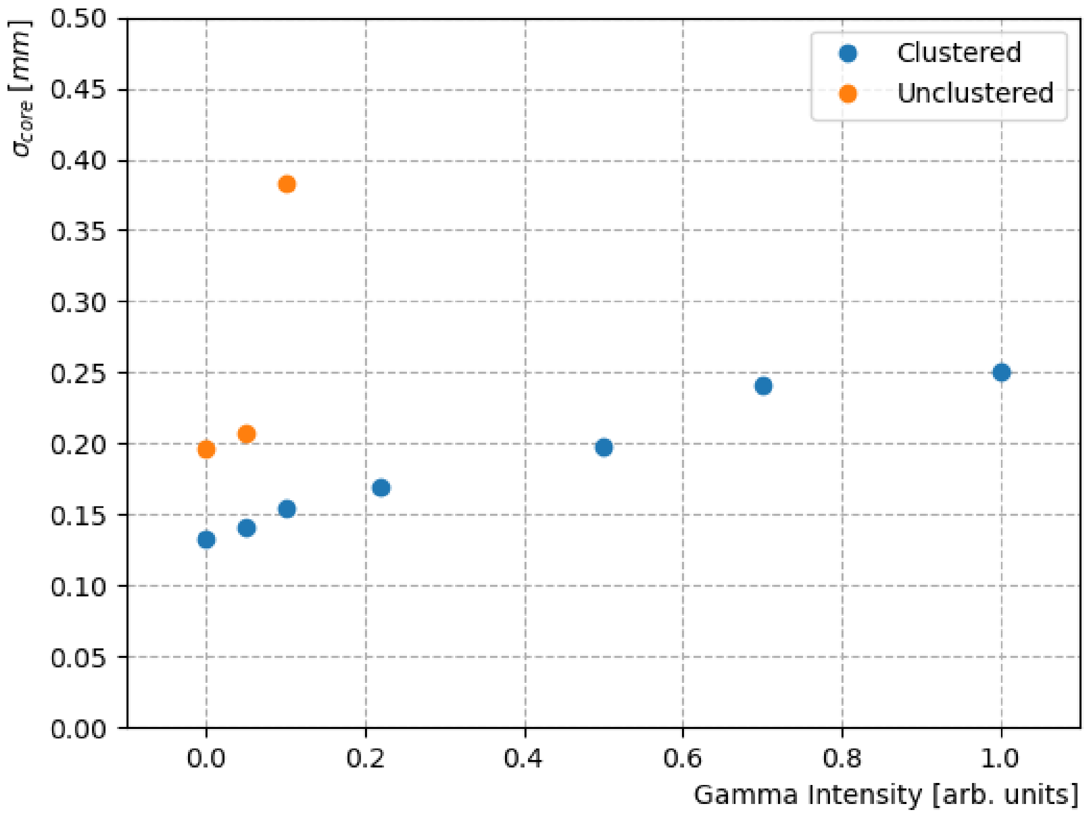
- Deeper neural network with skip connections
- Graph neural network with each node representing a strip connected to neighboring strips through edges
- Transformer model with multihead attention
- Also tried RNNs like LSTM

No clear improvement compared to MLP

Dataset with gamma background radiation



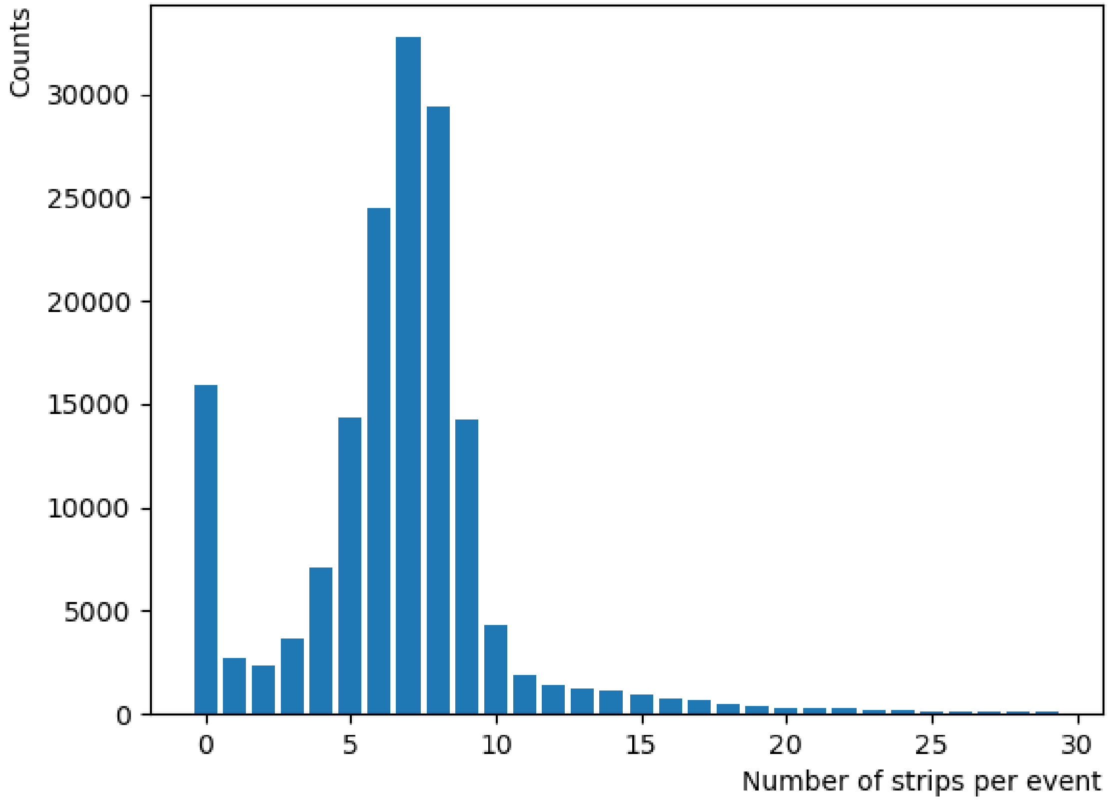
Efficiency

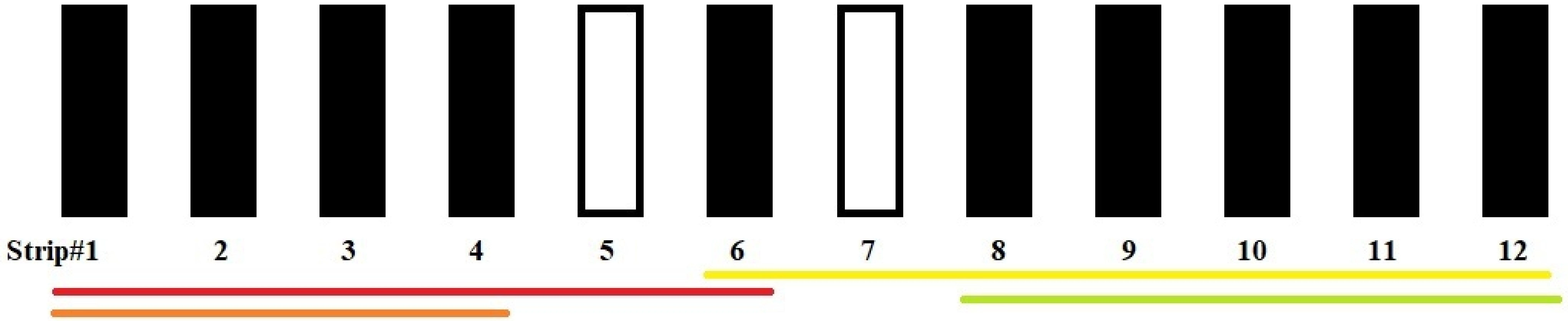


Core resolution

MLP no longer performs well with unclustered data given more background noise
For clustered events, results are similar to previously achieved in group

Clustering task





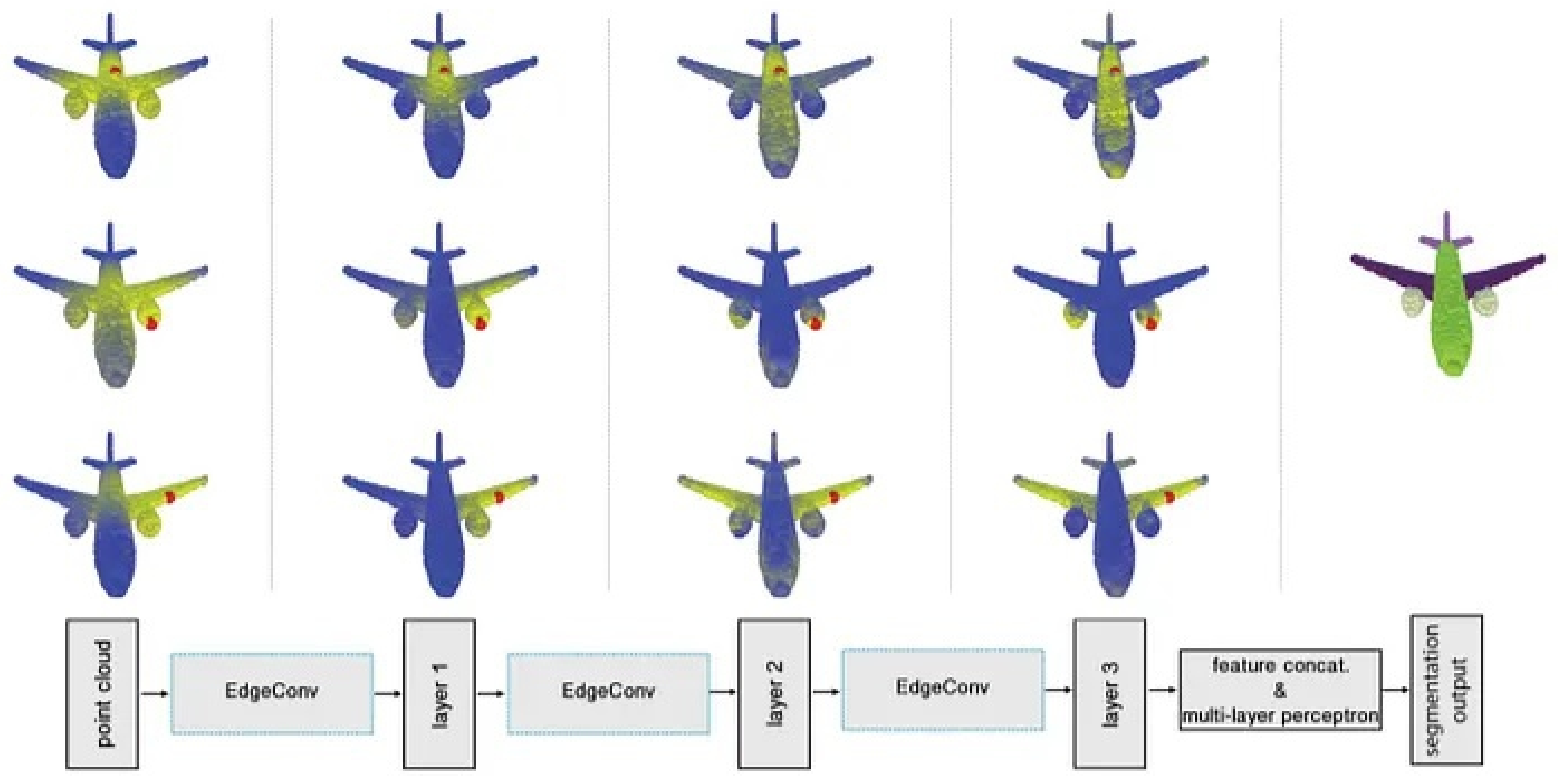
Example of an event with missings strips. Colored lines highlight possible clusters that can be build

For 29 degrees inclination, around 7 strips should be activated

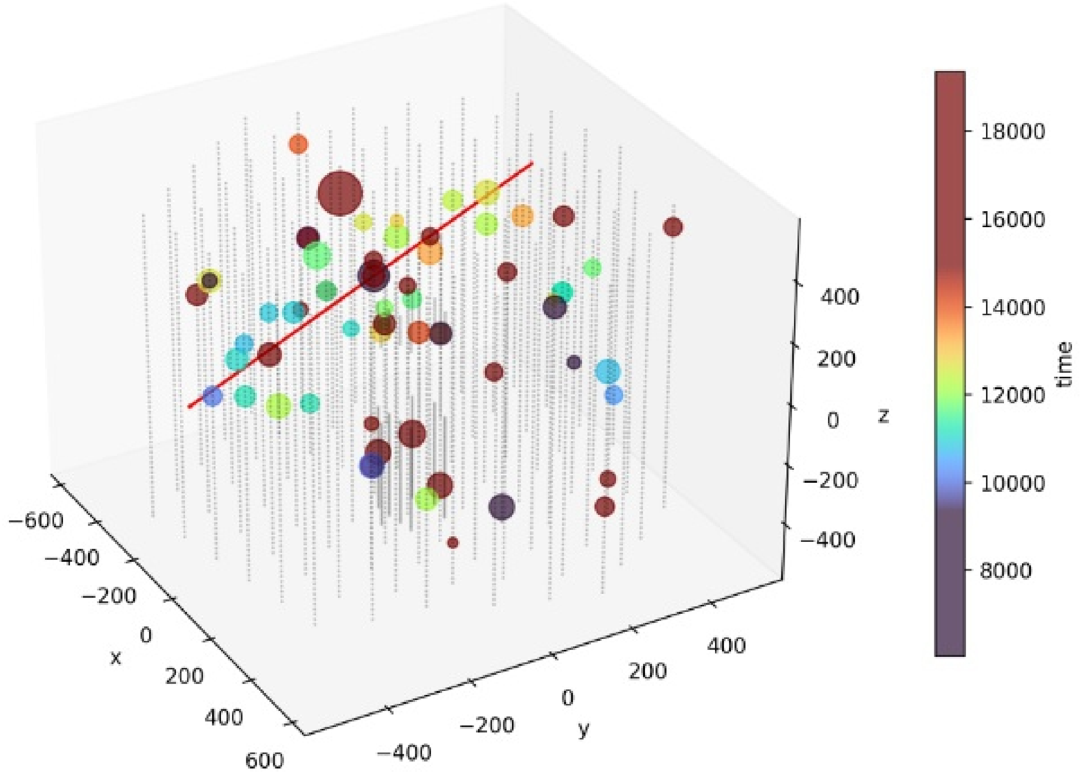
One method is building different clusters for each event and reconstruct position with cluster closest to particle track

Problems : Clustering may discard relevant strips. Big clusters likely have background noise

Work in progress : Clustering with Dynamic Graph CNN ?



Example of Dynamic Graph CNN



Example event IceCube detector

Dynamic Graph CNN was proposed as idea to learn topological features on point clouds

The method is currently used at TUM for analysis of IceCube neutrino detector and shows good results

Other problem :

Neural Network is a black box, we can't know exactly what is going on inside it !
What happens if real data differs from training data ?

Robustness

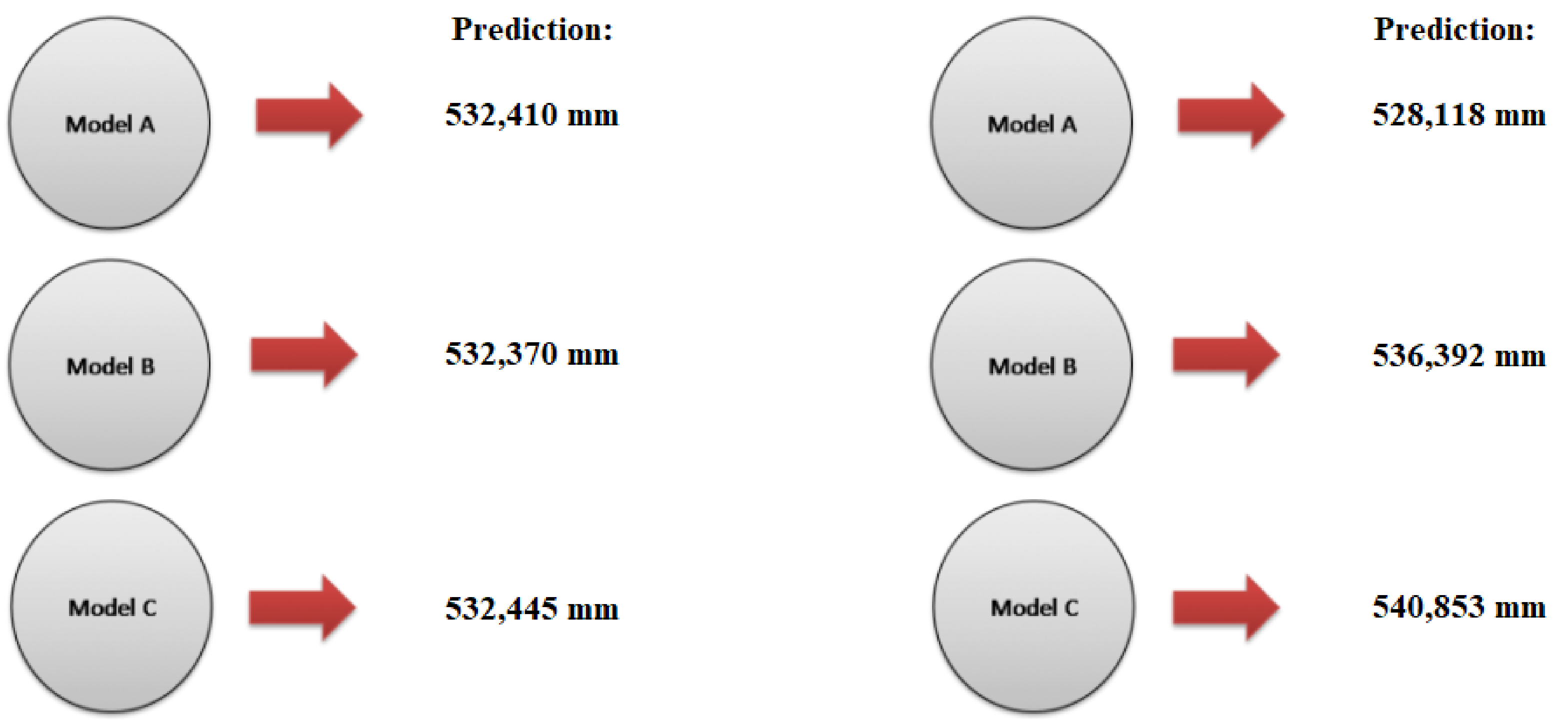
Model : Dataset :	520V 100ns	520V 200ns	530V 100ns	530V 200ns
520V 100ns	Efficiency : 89.0%	Efficiency : 87.6%	Efficiency : 88.1%	Efficiency : 87.1%
	σ_1 = 143 µm	σ_1 = 147 µm	σ_1 = 143 µm	σ_1 = 153 µm
520V 200ns	Efficiency : 90.1%	Efficiency : 90.0%	Efficiency : 90.5%	Efficiency : 89.7%
	σ_1 = 167 µm	σ_1 = 164 µm	σ_1 = 161 µm	σ_1 = 163 µm
530V 100ns	Efficiency : 88.7%	Efficiency : 88.5%	Efficiency : 89,2%	Efficiency : 88.4%
	σ_1 = 139 µm	σ_1 = 142 µm	σ_1 = 136 µm	σ_1 = 143 µm
530V 200ns	Efficiency : 90.4%	Efficiency : 90.5%	Efficiency : 91.1%	Efficiency : 90,2%
	σ_1 = 165 µm	σ_1 = 154 µm	σ_1 = 159 µm	σ_1 = 148 µm

Cross comparison : core resolution and efficiency for varying detector parameter

The MLP is relatively robust dealing with varying detector parameters

A better normalization of the features should increase robustness of the model

Ensemble models



Trustworthy prediction

Unreliable prediction

Robustness of single event reconstruction can be investigated with Ensemble models
We train multiple models with similar architecture and size and compare the results

Other idea : Train a second NN to predict the residuals of the first NN

Conclusion so far

- Results for the regression task are similar to previous results in group
- Improvement may be achieved with a model discriminating background noise from muon signal

Thank you for your attention !