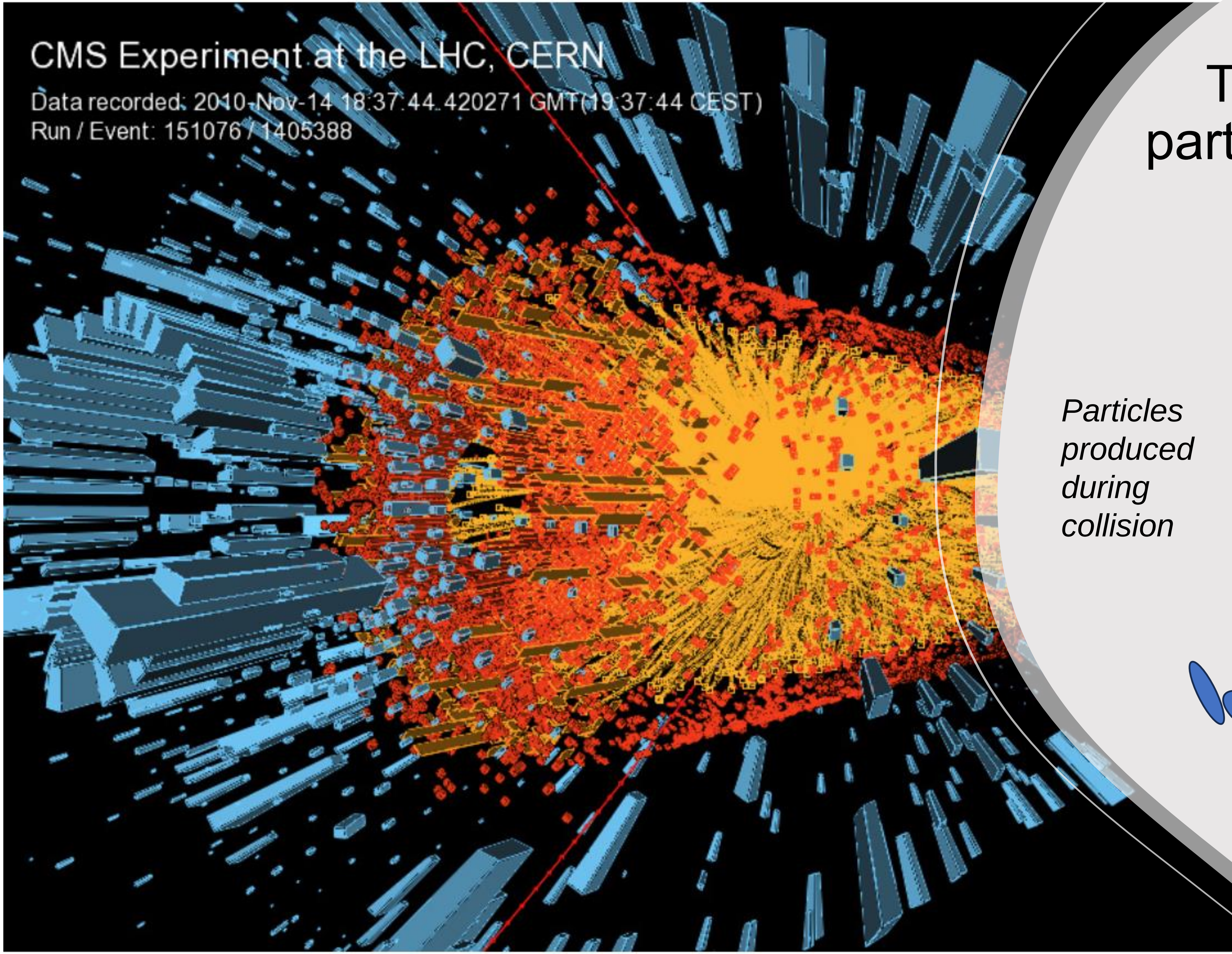
Implenting a Transformer as a Particle Flow Algorithm

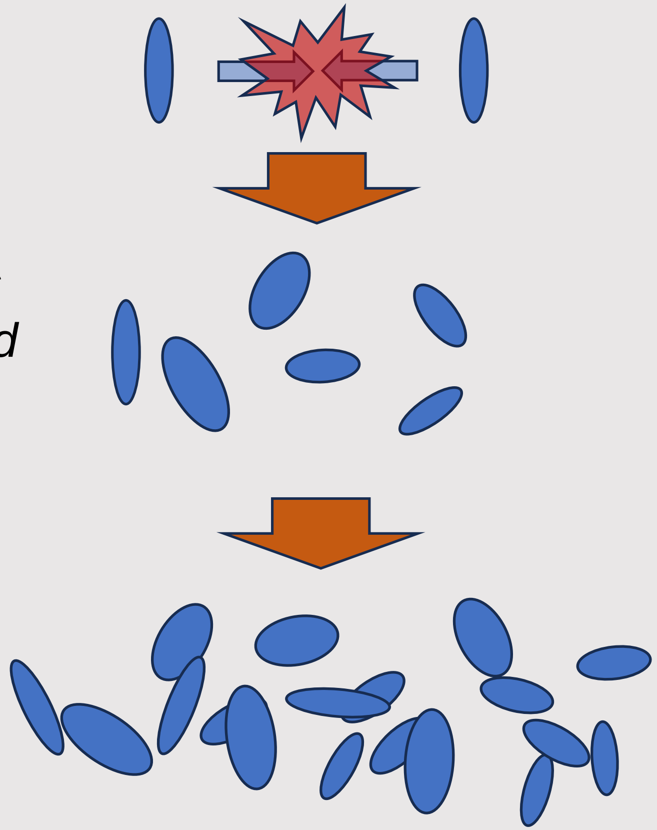
03.07.2024

Paul Wahlen,

Supervised by Taikan Suehara & Junping Tian

The problem with particle accelerators…

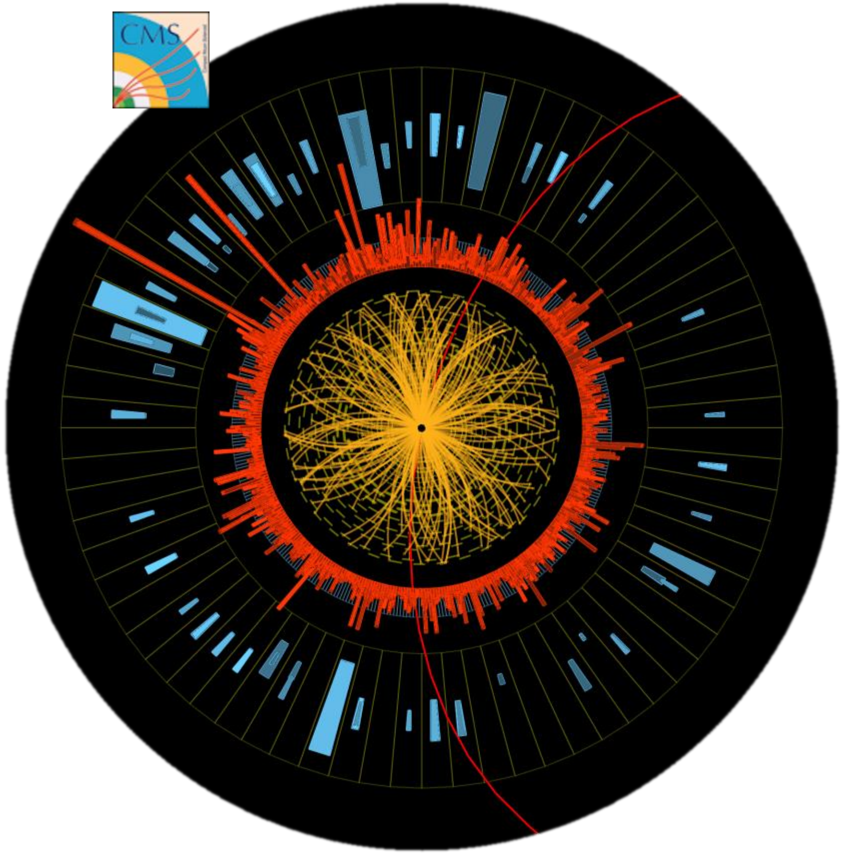




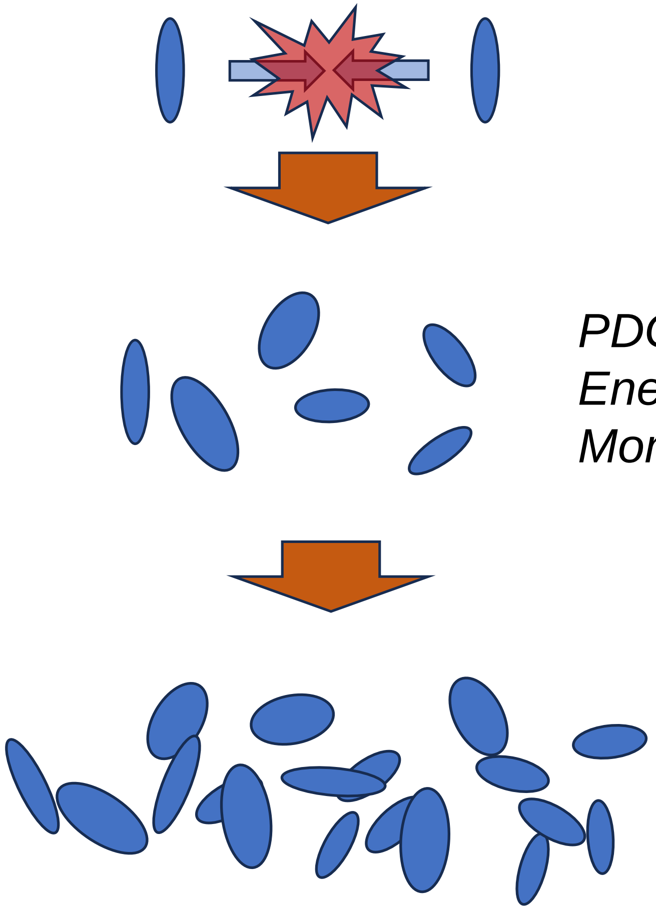
Particles produced during collision

Going through the detectors…

The problem with particle accelerators…



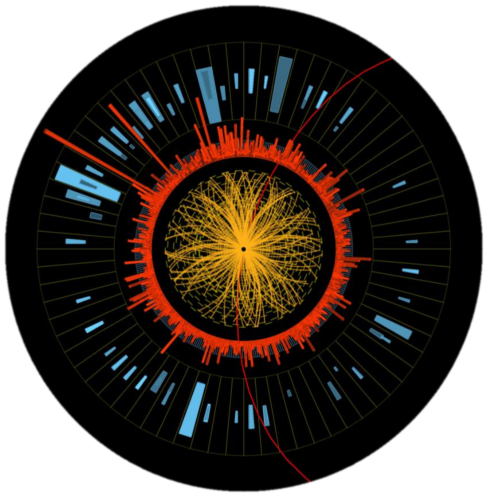
From the detectors, how do we retrieve the particles initially produced?



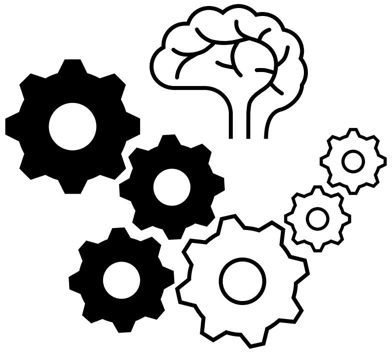
PDG?
Energy?
Momentum?

Only energy deposits and tracks are left behind

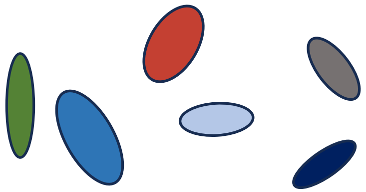
A possible solution (hopefully)



Energy deposit
Position

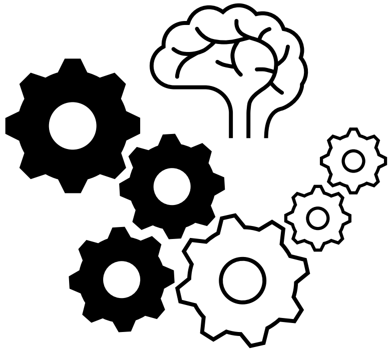


Charge
PDG
Energy
Momentum



Looks somewhat similar to a well known problem: machine translation

My name is Paul



私はポールです

Nowadays, achieved using a Neural Network called Transformer

Project concept

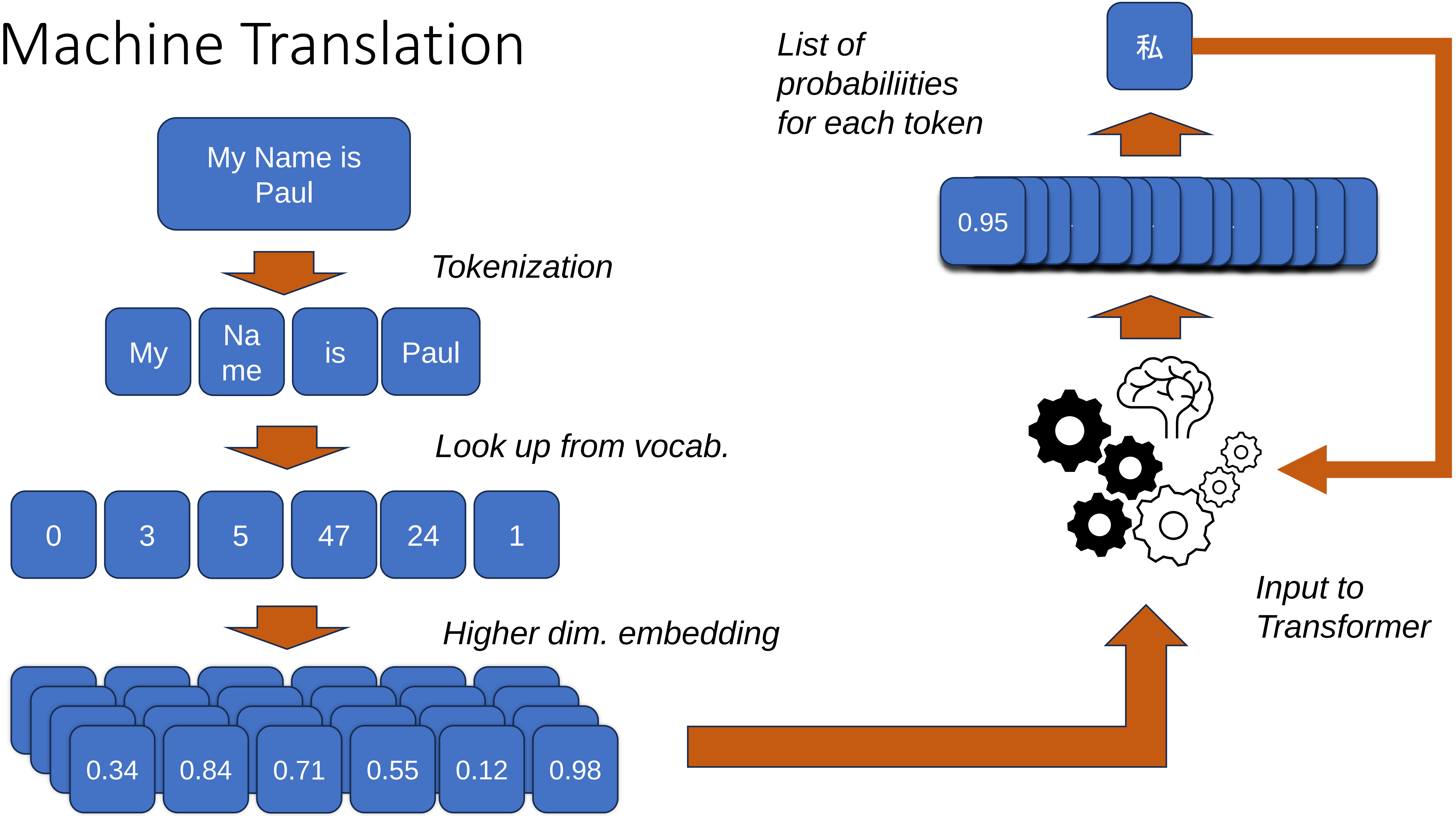
Using a Transformer to predict the particles generating the clusters

	Sequence to Sequence	**Physics**
Input	Sentence	List of hits from 1 event
Output	Machine translation of Seq	List of clusters to which belongs each hit
token	Depends, words/ few char.	1 hit
Special tokens	bos, eos, unkwn, pad	bos, eos, sample, pad

$$\begin{pmatrix} \text{bos} & \text{hit} & \text{hit} & \text{eos} & \text{pad} & \text{pad} \\ \text{bos} & \text{hit} & \text{hit} & \text{hit} & \text{eos} & \text{pad} \\ \text{bos} & \text{hit} & \text{eos} & \text{pad} & \text{pad} & \text{pad} \end{pmatrix}$$

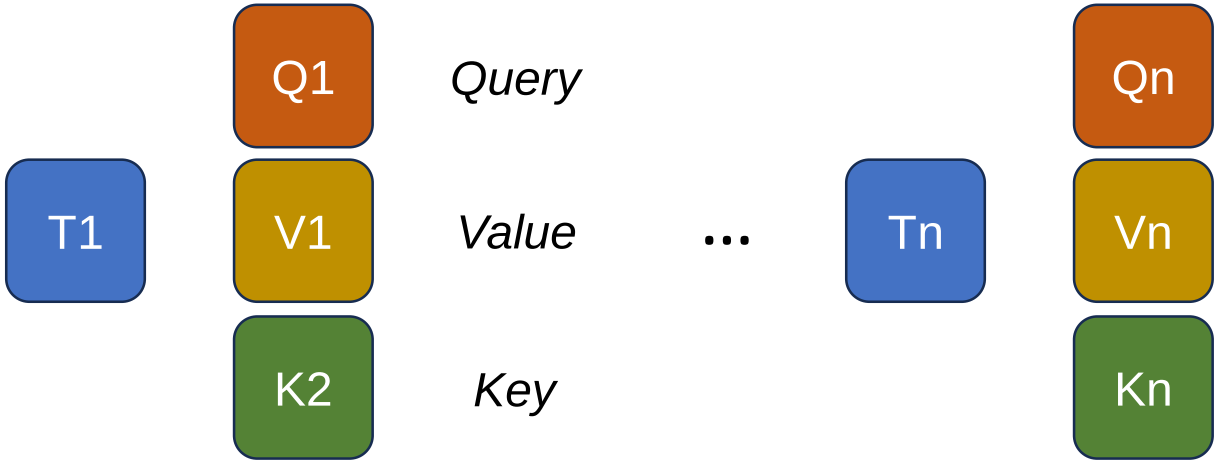
- Special symbols and general formatting of the raw dataset is done by a custom Pytorch's Dataset

Machine Translation

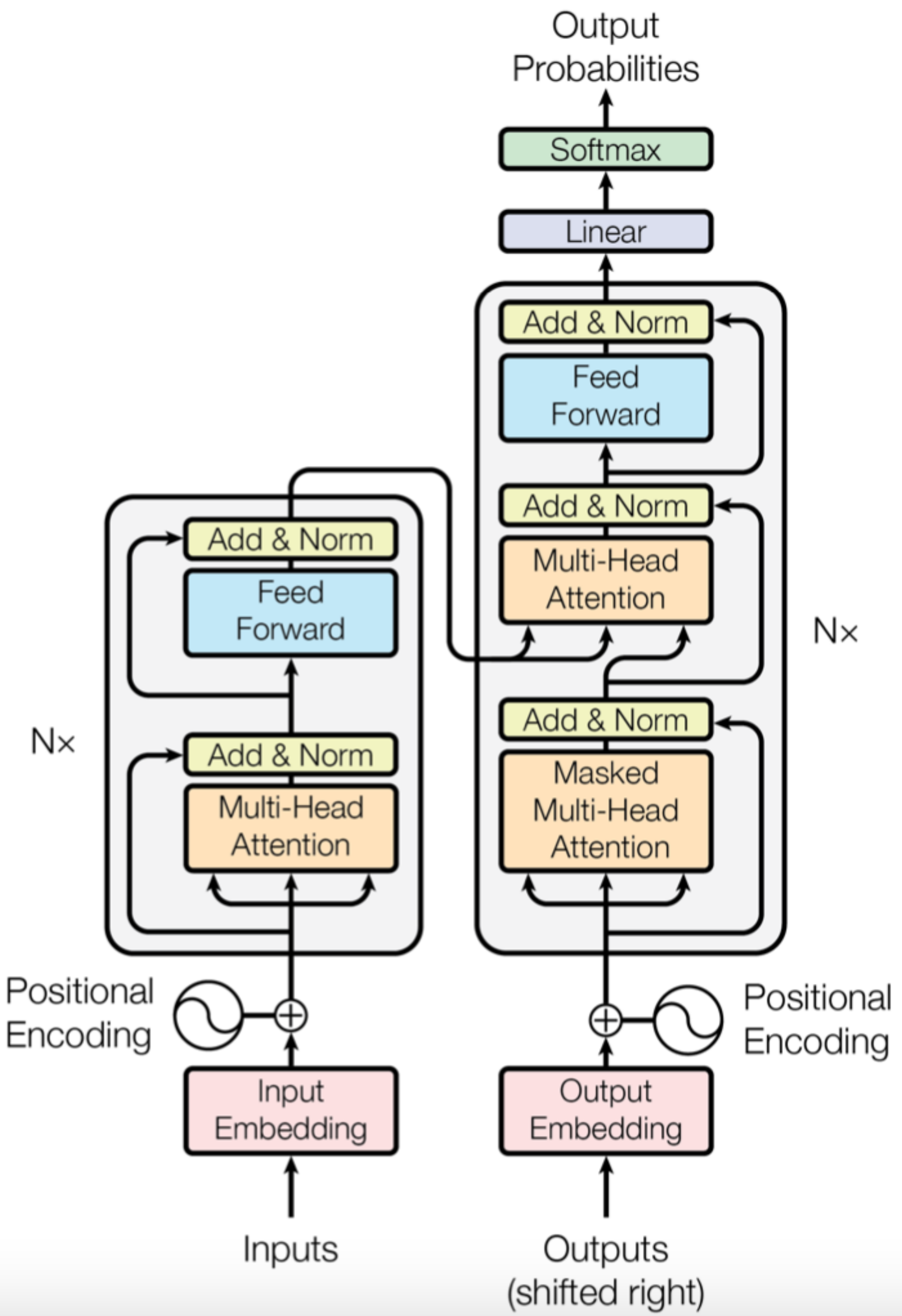


Inside a transformer

Multi-head Attention: Allows tokens to communicate with each other to better understand the context in which they are used.

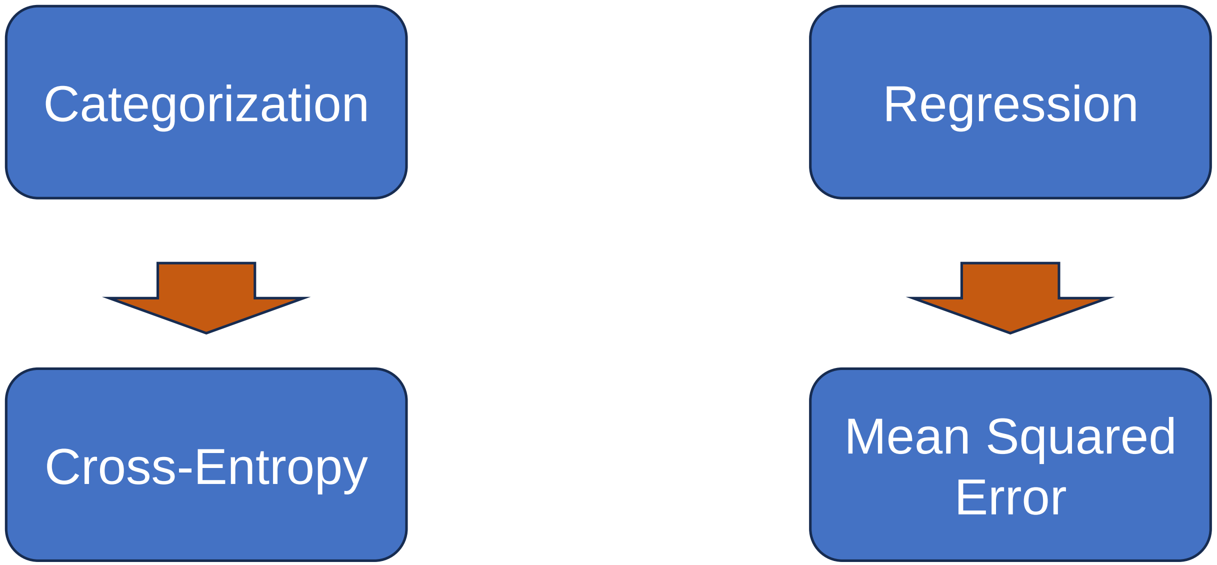


$$T_i' = \text{Attention}(Q_i, K_j, V_j)$$
$$= \sum_{j=1}^{n} \text{softmax}\left(\frac{Q_i K_j^T}{\sqrt{d_k}}\right) V_j$$



Shaping the output we want

- Determined during training, when the model adjusts its parameters to minimize *the loss function*
- Good prediction means that the ”distance” between model output and truth is small.



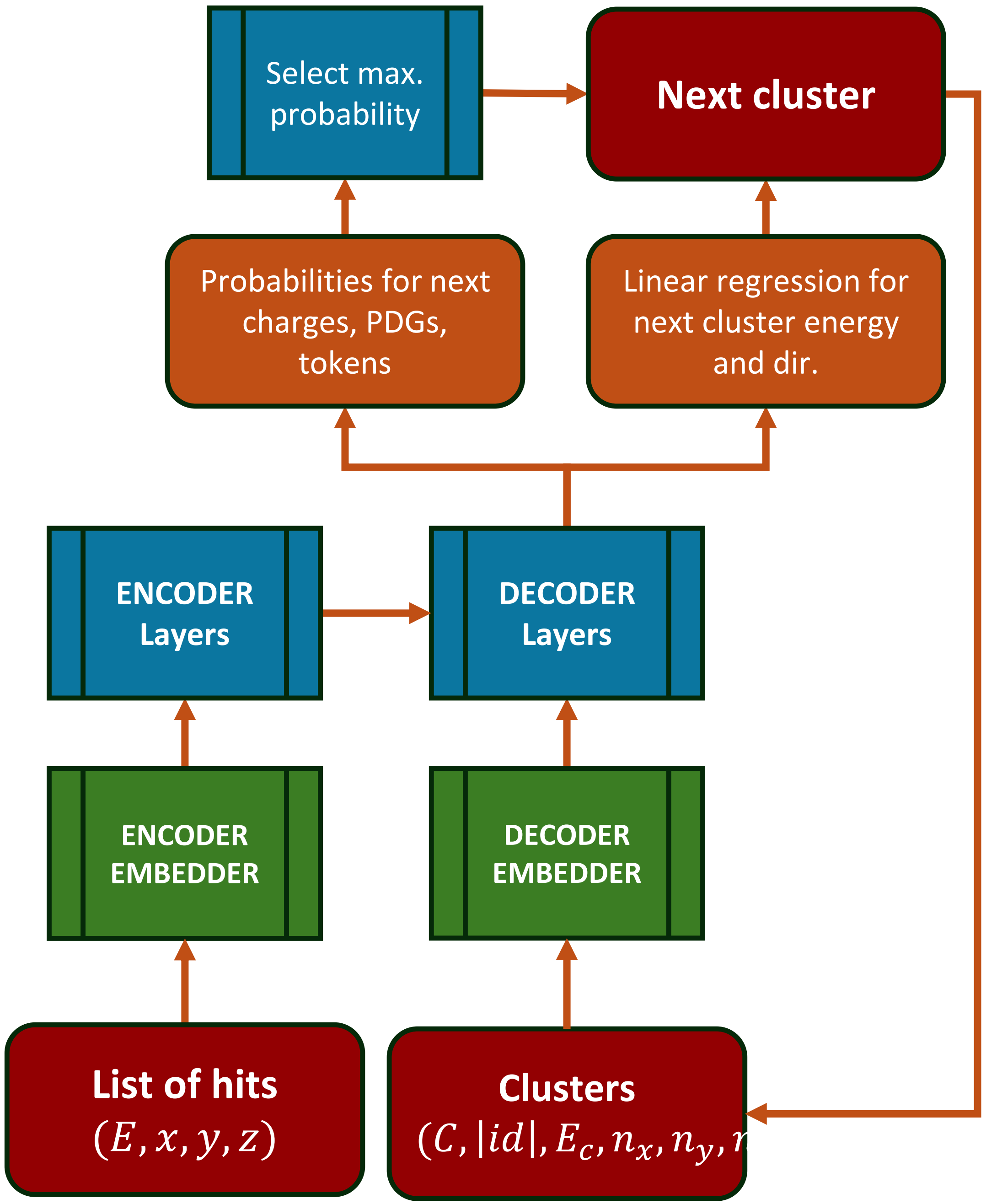
In both cases, the further away from the truth, the larger the value of the loss function

General Architecture

Cluster information are obtained from MC Particle truth information.

3 loss functions, weighted by hyperparameters:

- Most common particle ids form vocabulary: $\gamma, K_s, K_L, K^+, \mu^-, p, n, \pi^\pm, e^-$ CrossEntropyLoss
- Charges form other vocabulary. -1, 0 ,1. Also CrossEntropyLoss
- Continuous variables are obtained by regression. MSE for the loss function.



Photons

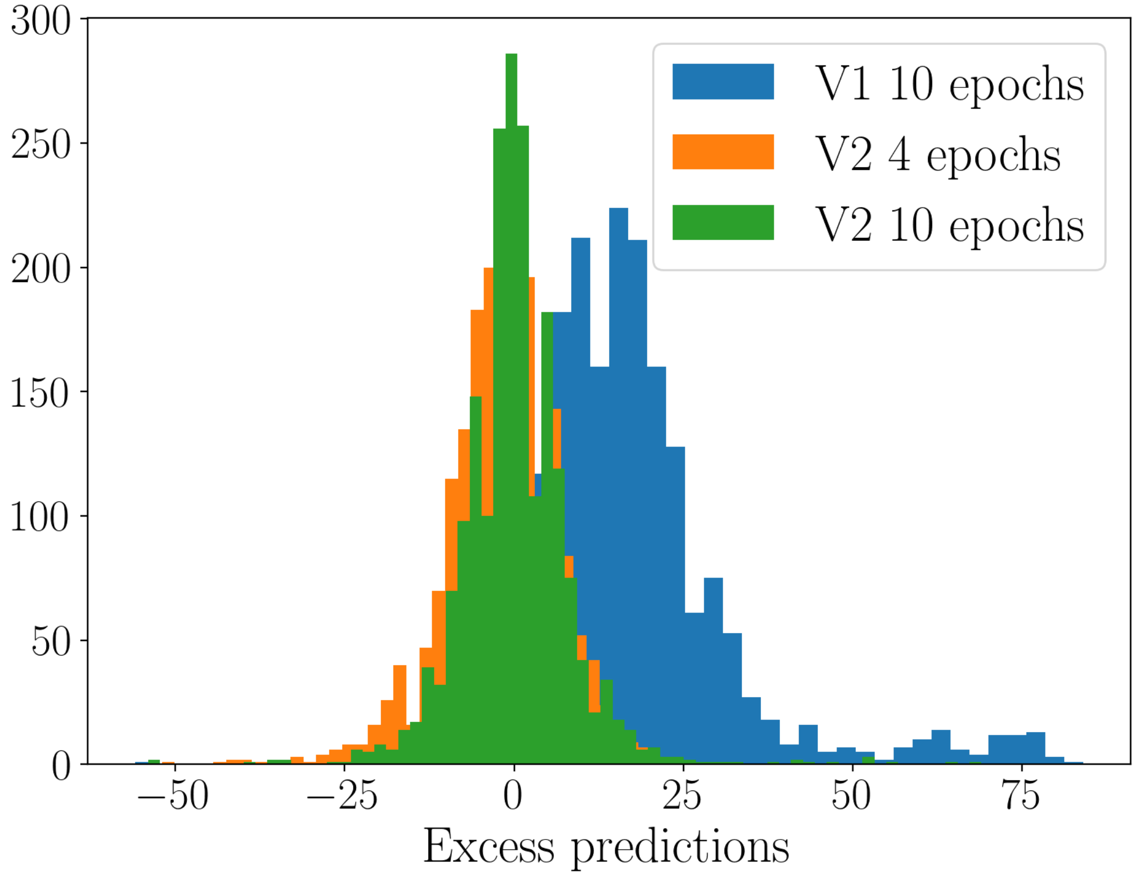
- Model was tested against clusters generated by either single photons or 2 photons
- Maximum accuracy is not achieved since photons can split into particles/antiparticles, etc...

Dataset	Charges	PDGs	direction	Energy	Excess
1 photons	88%	88%	5 degrees	6.1%	-0.27
2 photons	93%	96%	7 degrees	1.1%	-0.079

Further work

Version	Charges (%)	PDGs (%)	E (%)	θ [°]	Excess
V2 4E	59.59	39.65	530	71.91	-2.124
V2 10E	59.15	43.77	496	69.98	0.072
V1	62.78	45.34	2734	72.56	16.57

- Increasing the complexity of the dataset using 10 taus to form the clusters
- Focusing on predicting the correct numbers of clusters first



Conclusion

- Implementing a transformer to cluster hits in calorimeters for particle flow with an architecture analog to what is used in Language Model
- High accuracy achieved for most simple datasets
- Currently trying to generalize to more complex dataset