# ILC Weekly Meeting

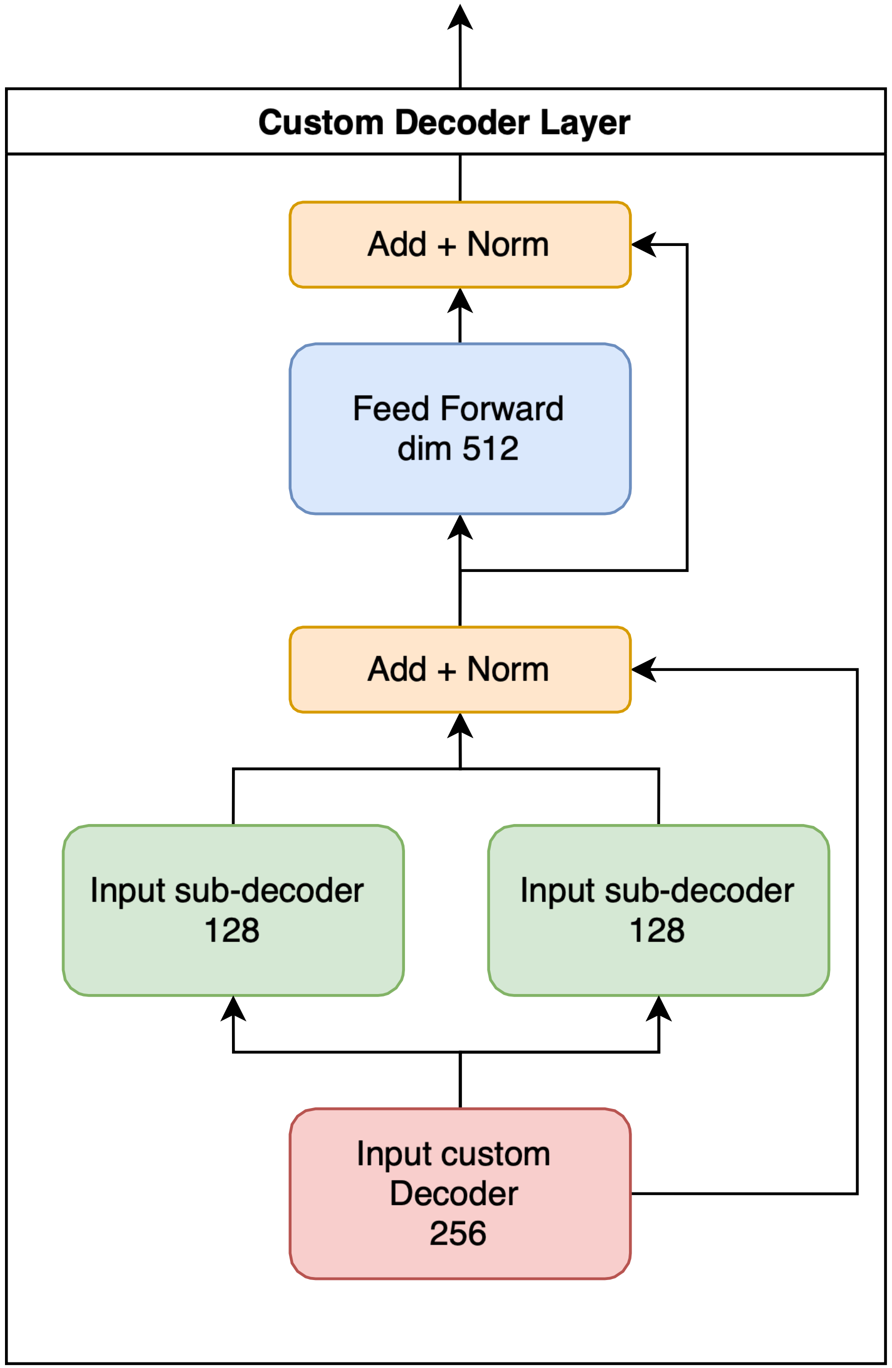
06.20.2024

# Since last week

- Implemented a new architecture
- Currently working on optimizing time efficiency and CPU memory allocation of preprocessing phase before processing the entire dataset

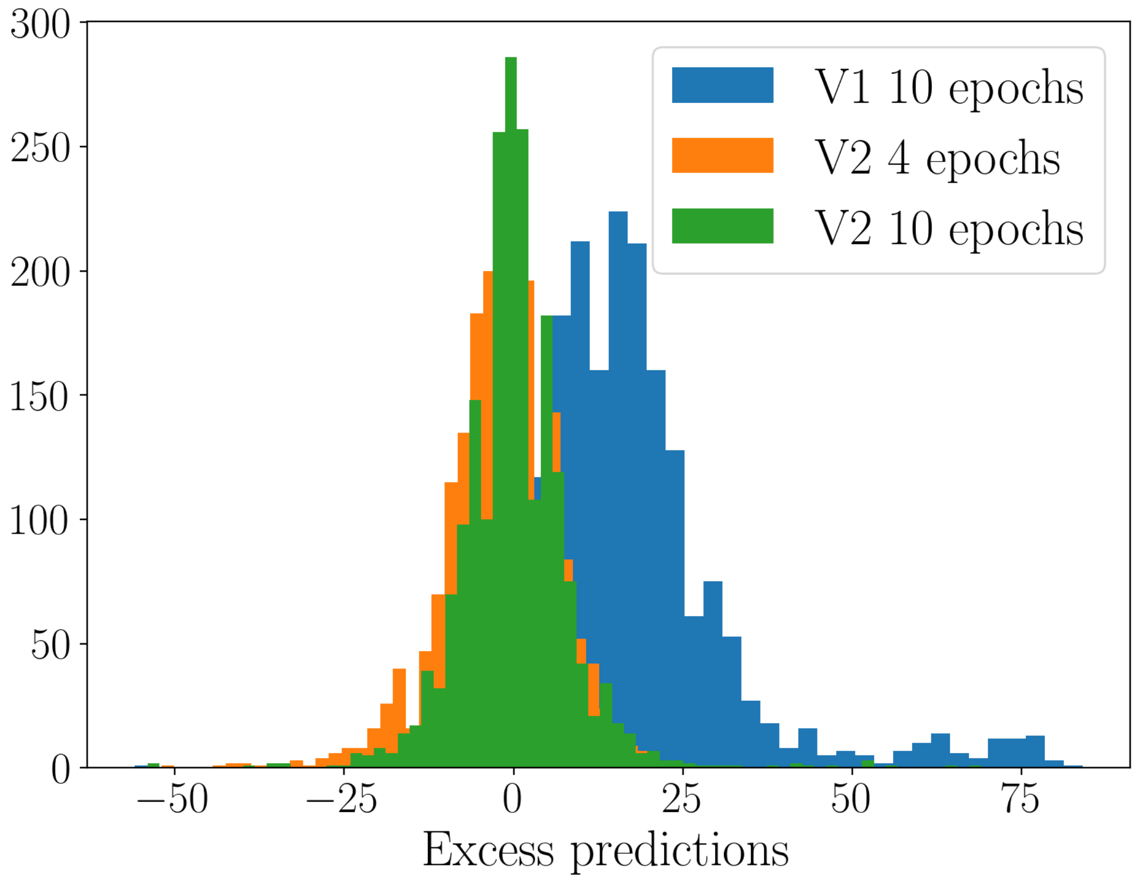
# New architecture

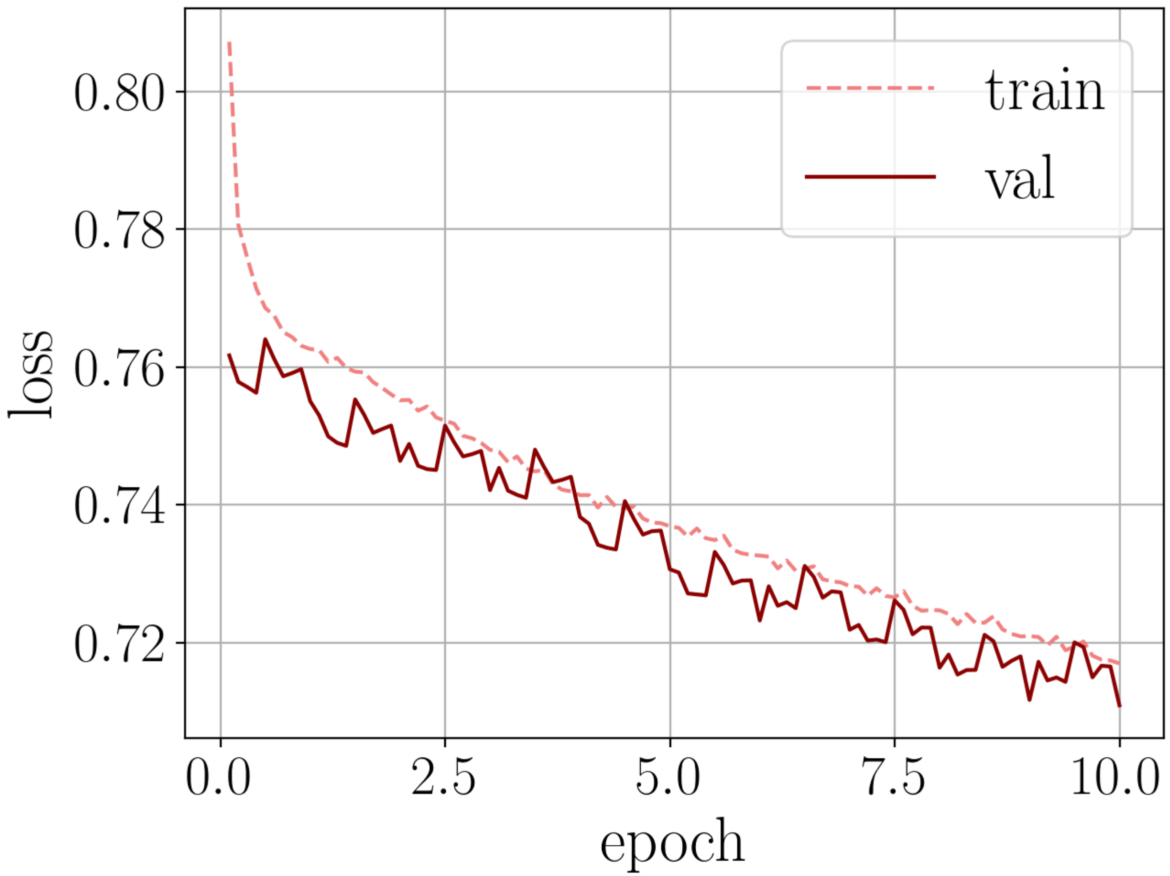
- Tries to address the potential problem of finding common embedding space for source and target to enhance attention relevance
- n decoder sub-units inside an custom decoder layer
- Only 2 sub-decoders on diagram but as many can be added as long as dim input decoder / n = dim output encoder



- 20% of the dataset (160’000 events) for 10 epochs.
- Per epoch: ~6hours

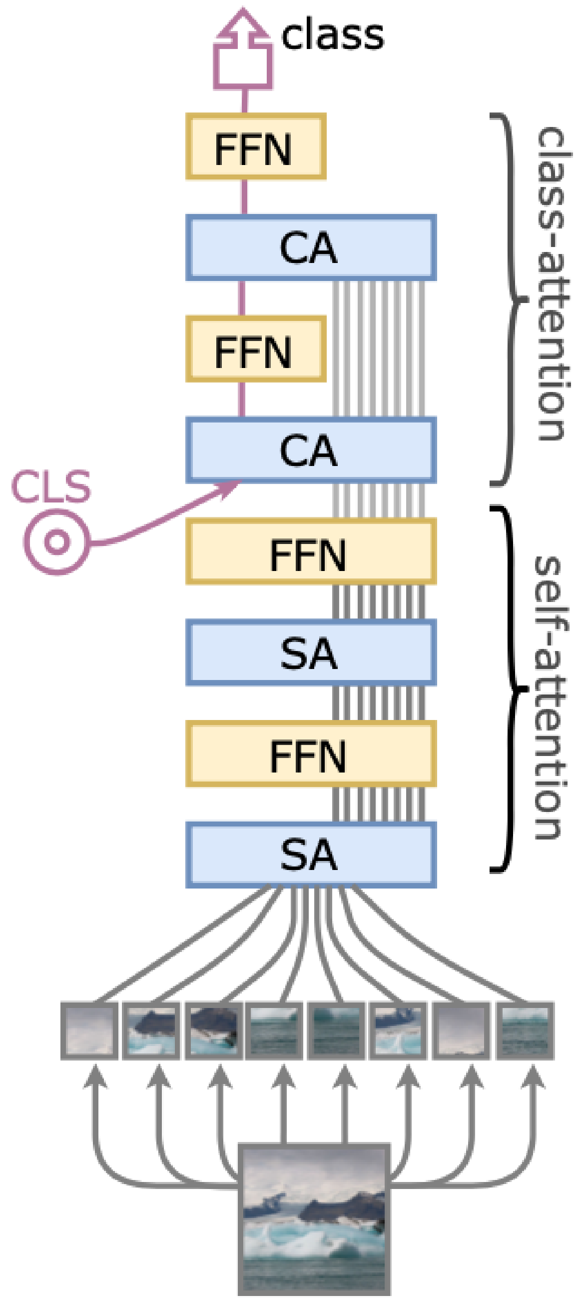
| Version | Charges (%) | PDGs (%) | $E$ (%) | $\theta$ [°] | Excess |
|---|---|---|---|---|---|
| V2 4E | 59.59 | 39.65 | 530 | 71.91 | -2.124 |
| V2 10E | 59.15 | 43.77 | 496 | 69.98 | 0.072 |
| V1 | 62.78 | 45.34 | 2734 | 72.56 | 16.57 |



# Pretraining of source embedding

- Hope: to orientate model embedding towards similar space to labels embedding to achieve better performances and more efficient training.
- Encoder-only transformer to predict the number of clusters from the hits
- First layers are for self attention between tokens
- Last layers are used to extract class information into injected class tokens. Samples tokens are frozen during these class-attention processes.



# To do

- Specific loss function to predict eos tokens
- Training with different fraction of dataset: 2%, 20%, 50%, 100% to compare performances of both models
- Test models with pretrained embedding for the source.
  - Only for the embedding
  - Encoder with frozen weights
  - Encoder with trainable weights