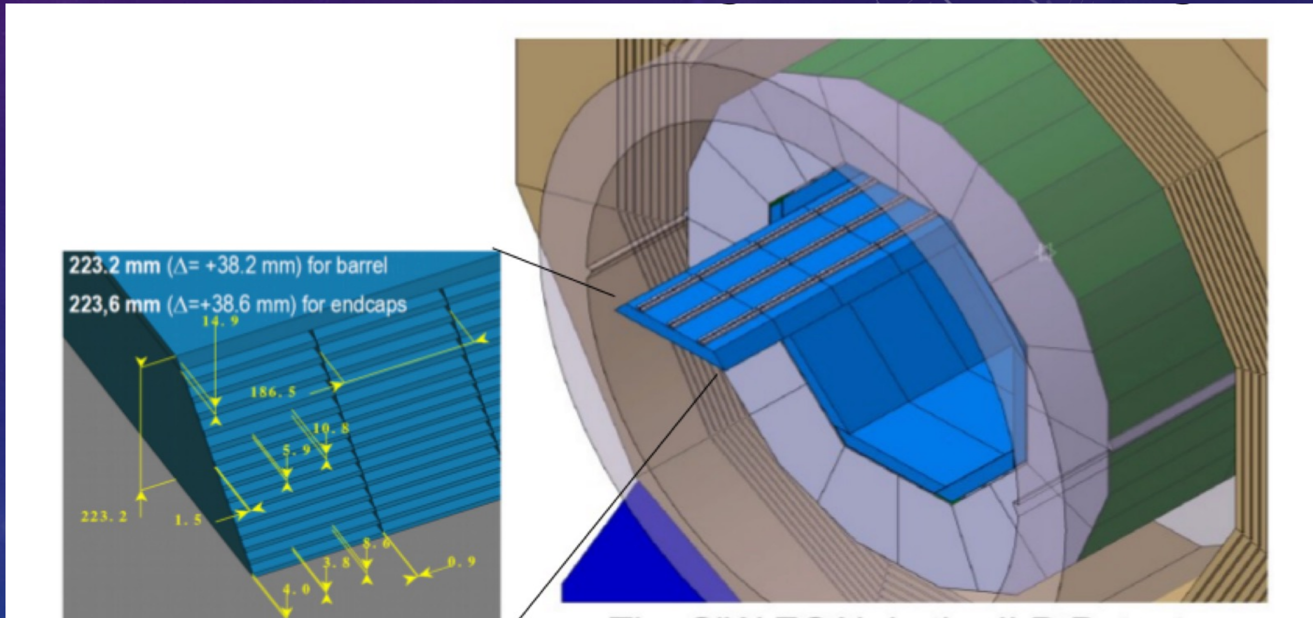
DEVELOPMENT OF ILC SHOWER CLUSTERING ALGORITHM USING DEEP NEURAL NETWORK

SHUSAKU TSUMURA

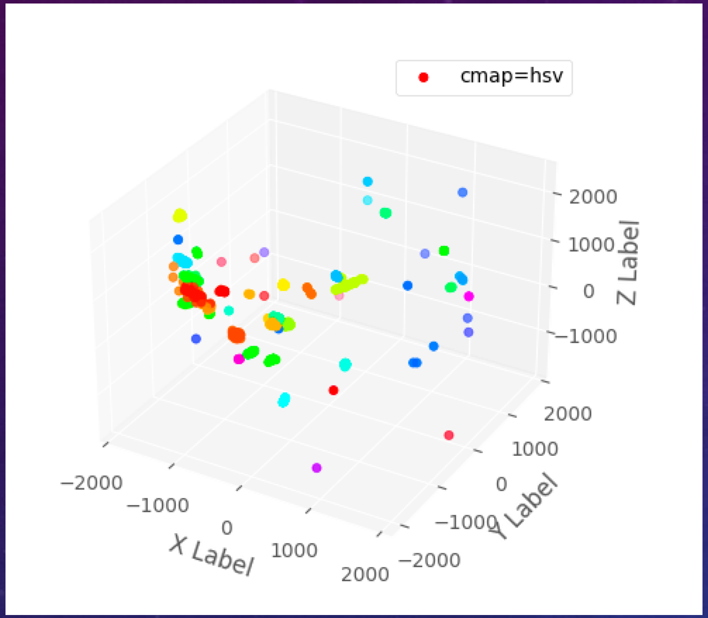
SIMULATION DATA

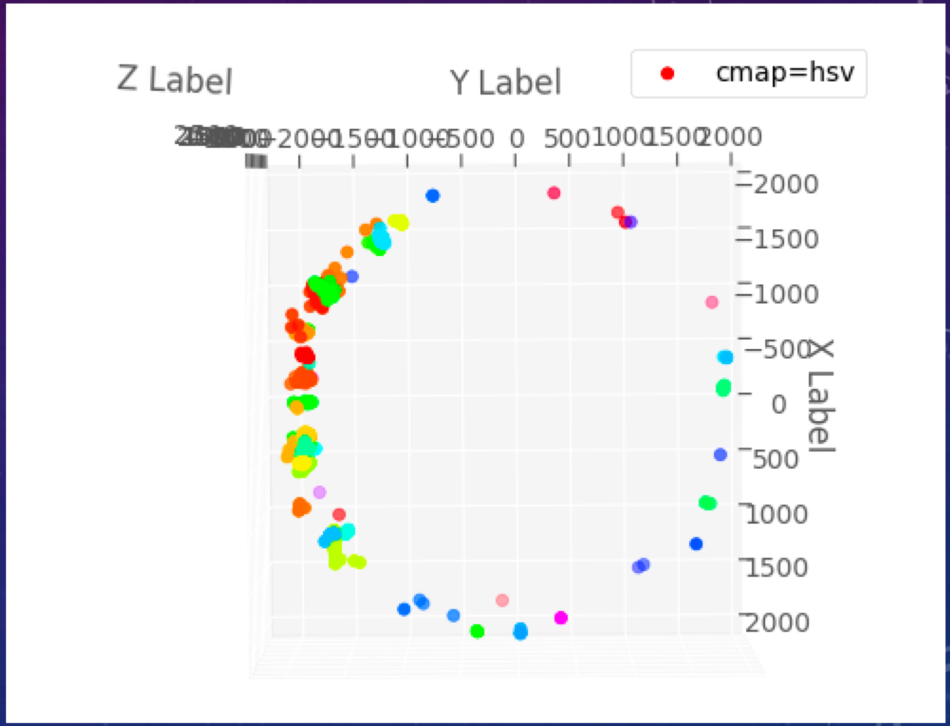
- Utilizing ILC and 500 GeV Simulation Data
- Z-> 2q events
- Clustering showers from hit information (Energy , x, y, z) measured in Ecal Barrel section



The SiW ECAL in the ILD Detector

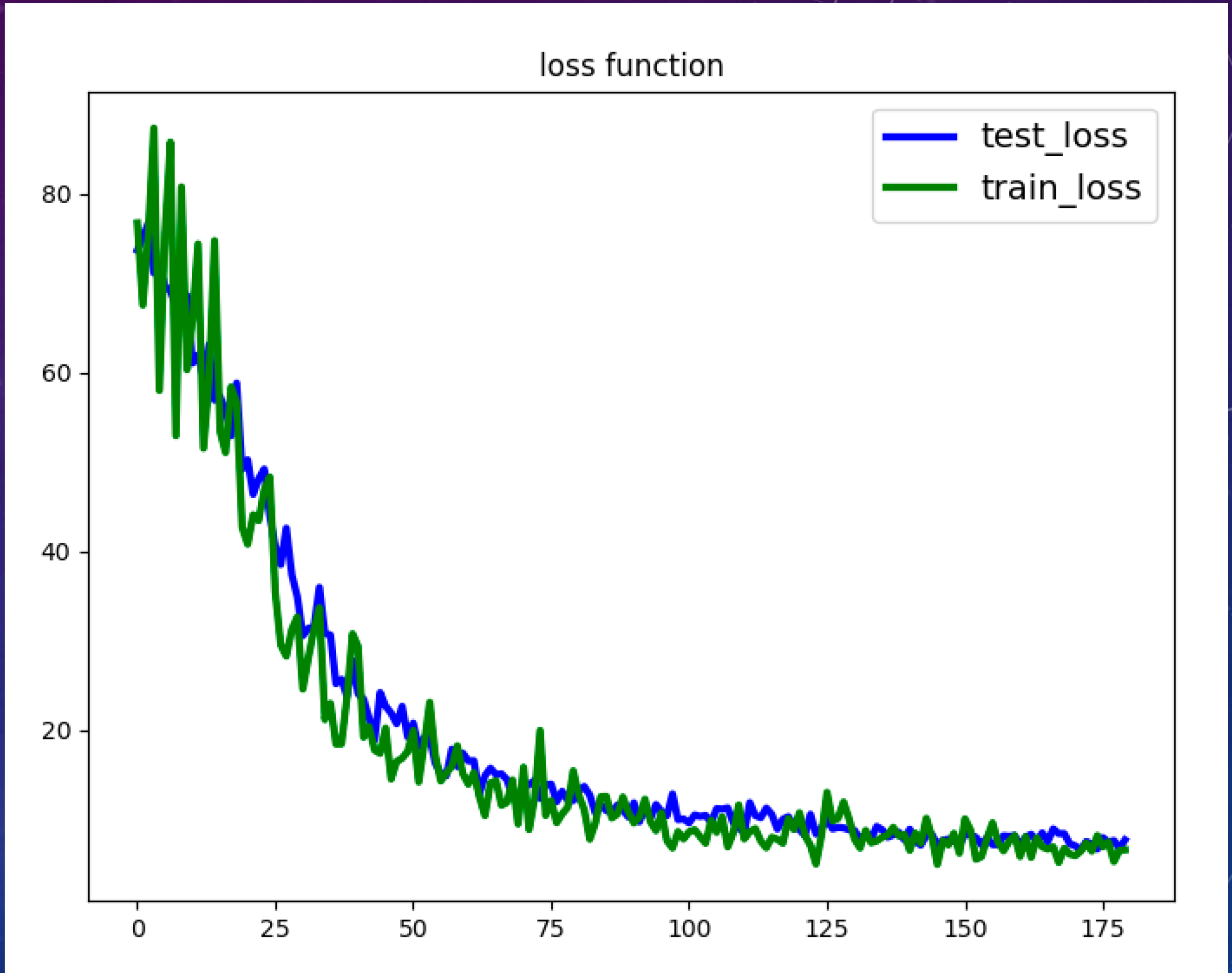
HIT DISTRIBUTION





- Events: 200, 80% as training data, 20% as evaluation data
- Each parameter is converted to the range of [-1, 1] by diving by 2000

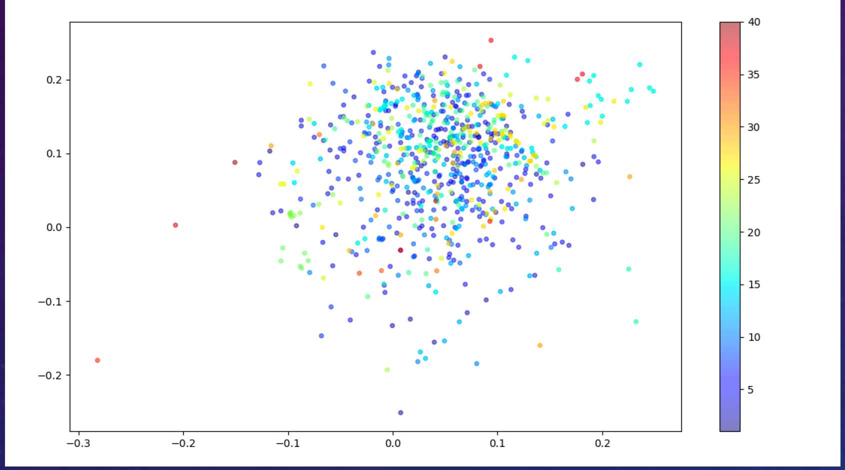
RESULT – LOSS FUNCTION



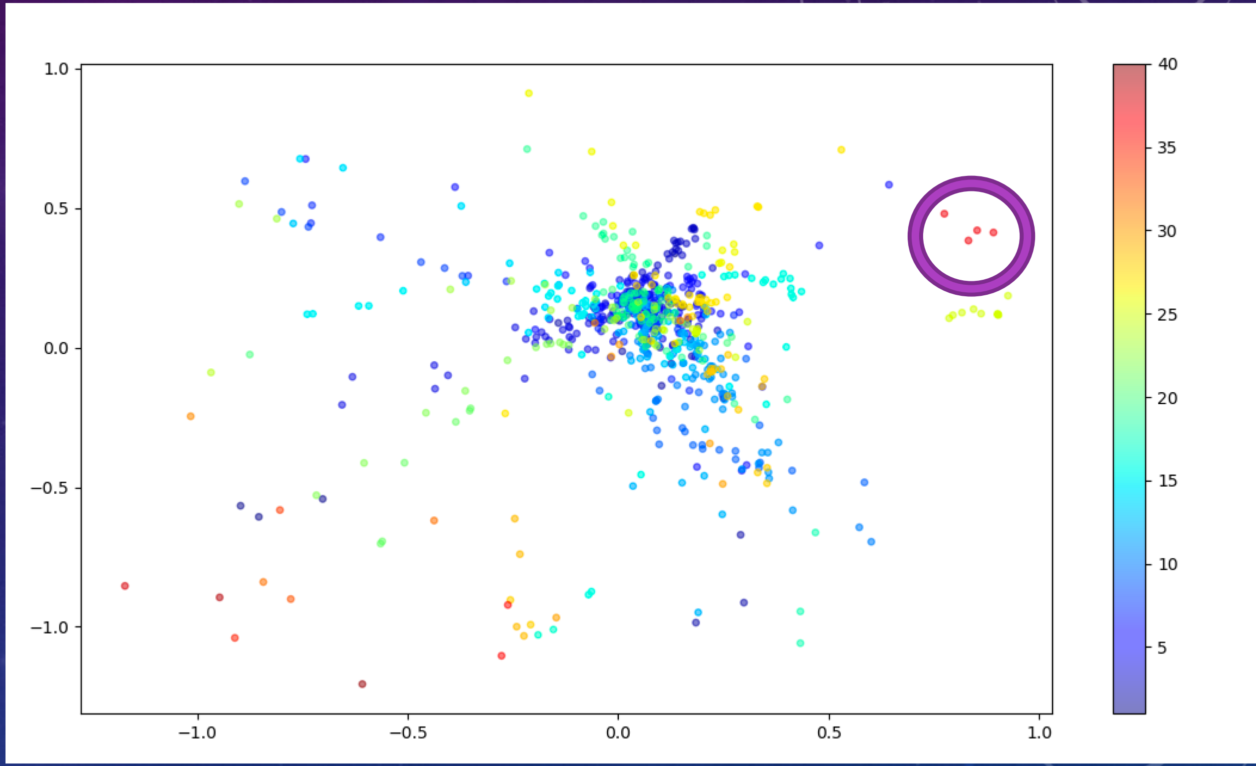
- Loss functions of both training and training data are decreasing
  → Learning works correctly.
- I have to evaluate accuracy also.

The Change of Representation Space

- Learning in representation space color-coded by cluster ID
- The dimension of the representation space is one of the hyperparameters, and is plotted here in two dimensions.



1 epoch

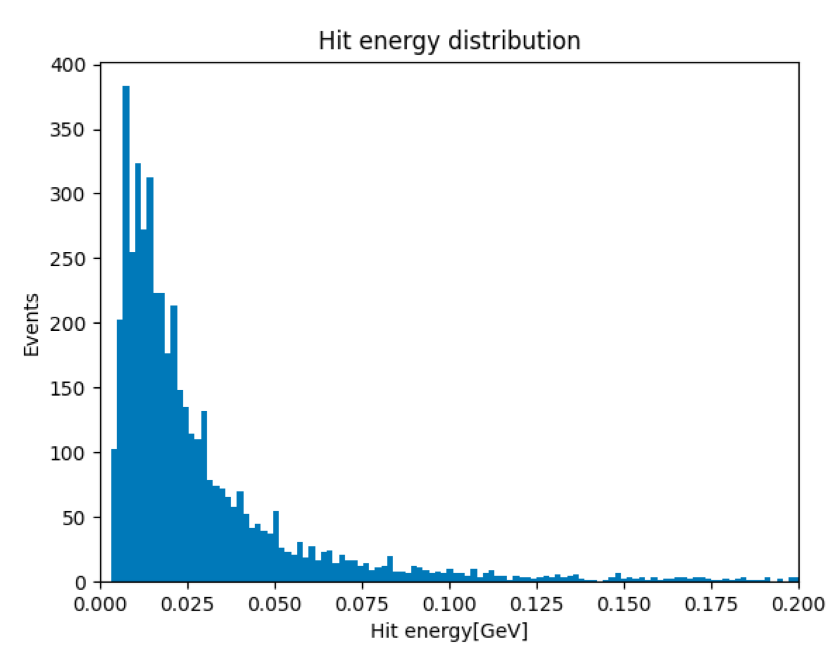


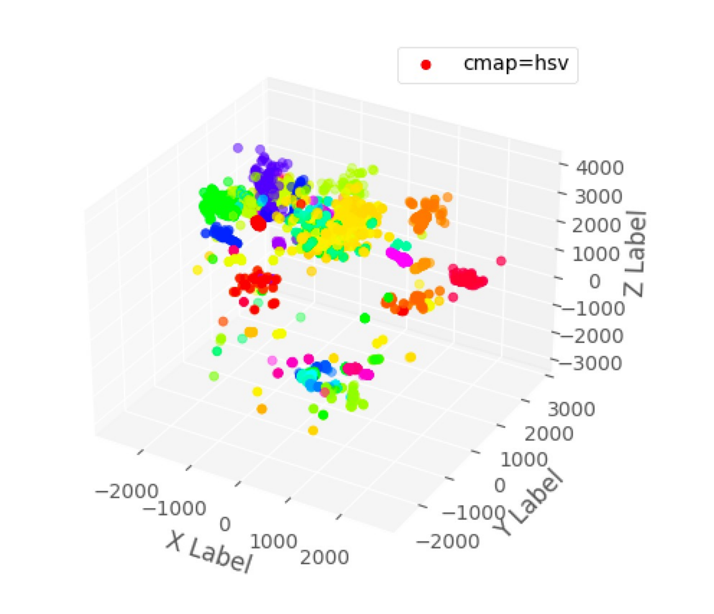
70 epoch

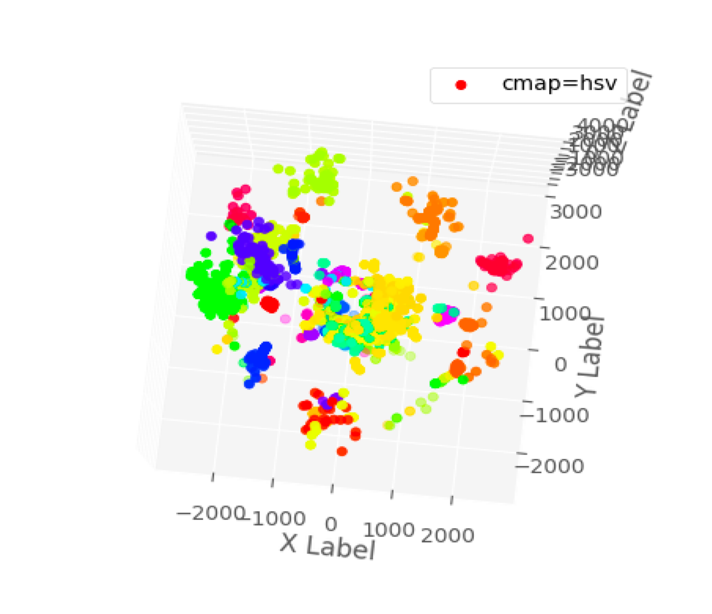
As the study progresses, each cluster is being collected by ID.
Easily identifiable IDs are separated, but some clusters are mixed

Adding the Data

- Added data for Endcap Rings and Endcap sections as well as Ecal Barrels and Hcal hits.
- Number of events added from 200 to 1600



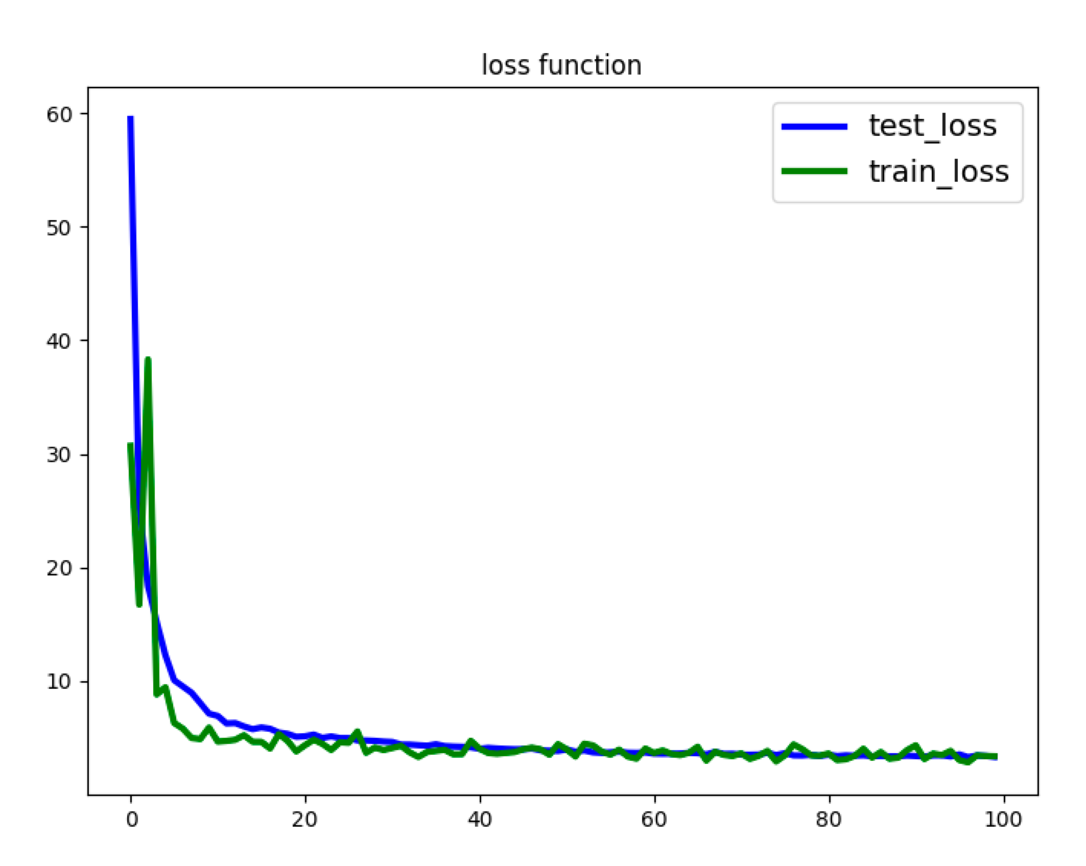




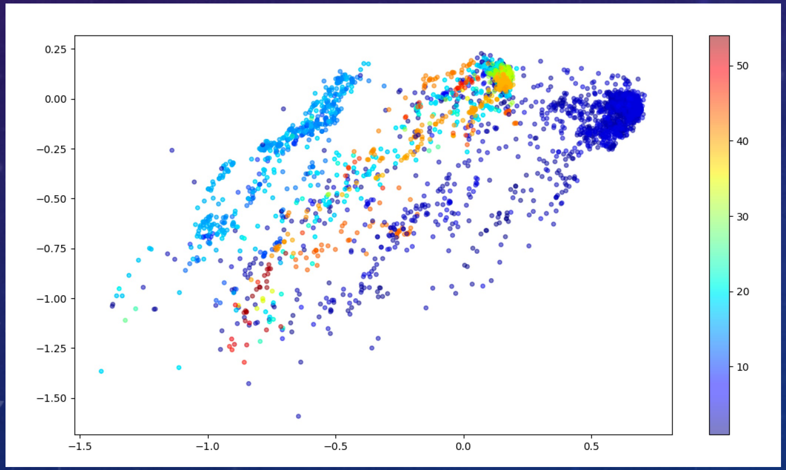
Results

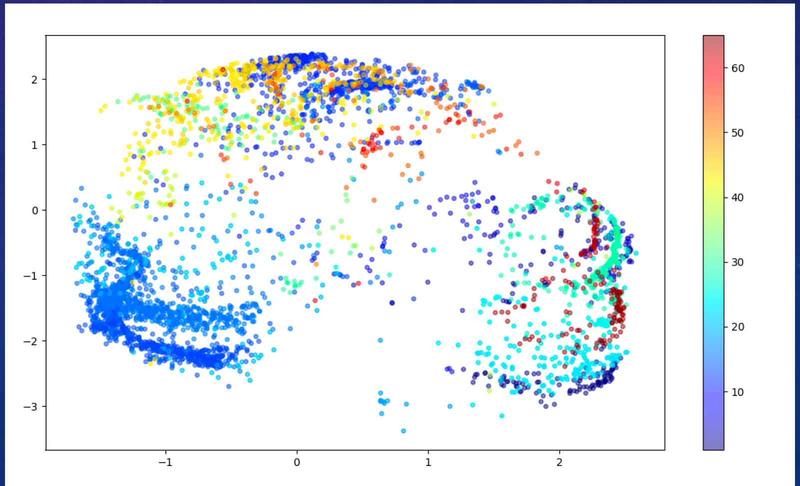
- Loss function is falling faster and to a greater extent than with less data

Loss



Epoch





Summary and Future Plan

- Graph Neural Networks are applied to the PFA and shower clustering algorithms in the ILC analysis framework.
- Two hundred events of Hit data measured with Ecal are used as simulation data.
- The training results showed a decrease in the loss function for both the training and evaluation data.

Future :

- Accuracy of the network
- Hyperparameter tuning and performance evaluation by comparison with PandoraPFA

BACKUP

GRAVNET - NETWORK -

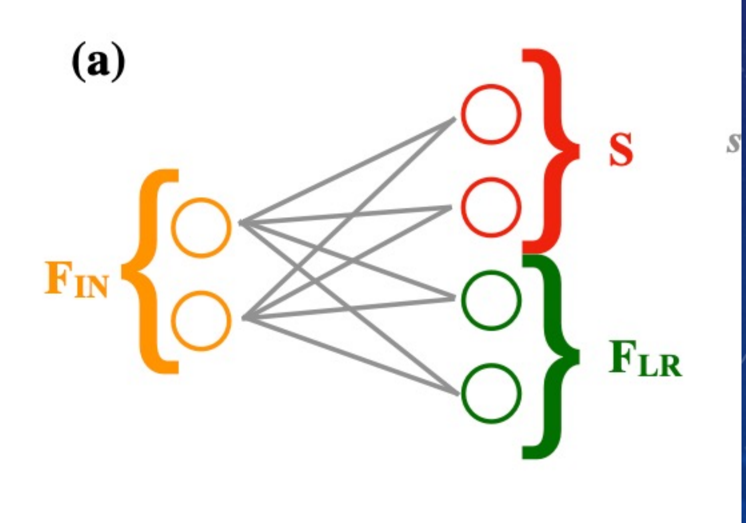
- Input Data : $B \times V \times F_{IN}$

  B : Number of examples including in a batch

  V : Number of hits for each detector

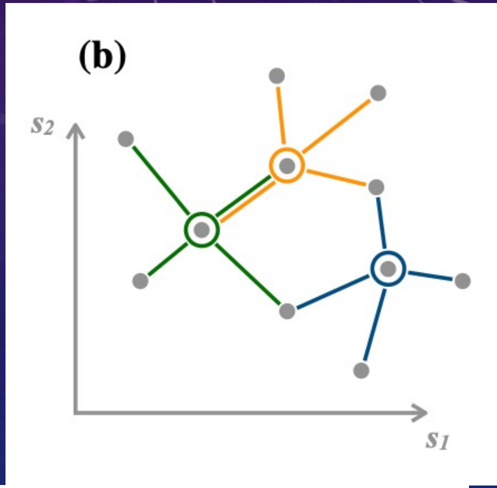
  F_{IN} : Number of the features for each hit

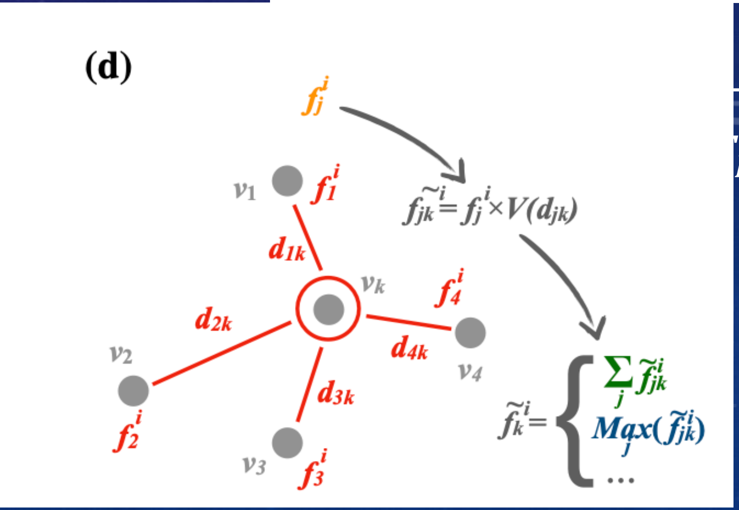
- S : Set of coordinates in some learned representation space
- F_{LR} : learned representation of the vertex features



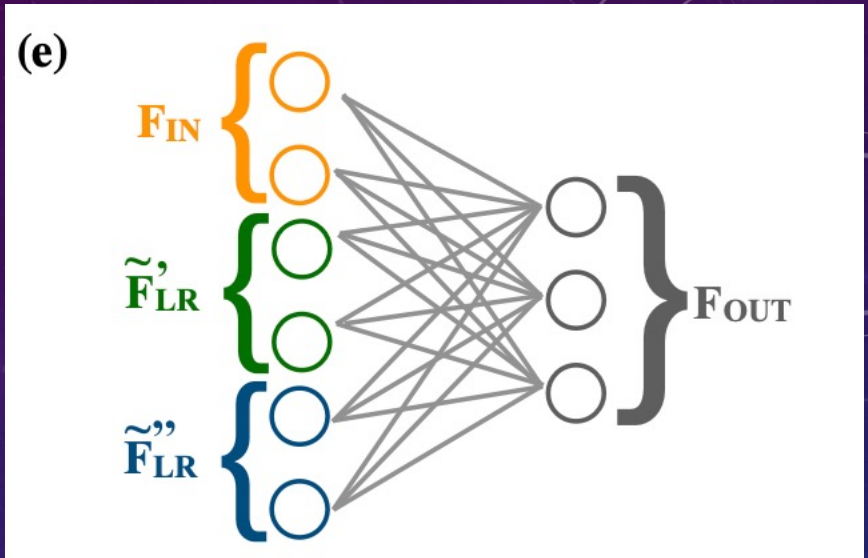
GRAVNET

- Input example of initial dimension $V \times F_{IN}$ is converted into a graph.
- the f_j^i features of the v_j vertices connected to a given vertex or aggregator v_k are converted into the $\widetilde{f_{jk}}^i$ quantities, through a potential (function of euclidean distance d_{jk}).
- The potential function $V\left(d_{jk}\right)$ is introduced to enhance the contribution of close-by vertices.
  Example: $V\left(d_{jk}\right) = \exp(-d_{jk}^2)$
- The $\widetilde{f_{jk}}^i$ functions computed from all the edges associated to a vertex of aggregator v_k are combined, generating a new feature $\widetilde{f_k}^i$ of v_k.
  Example : the average of the $\widetilde{f_{jk}}^i$ across the j edges / their maximum



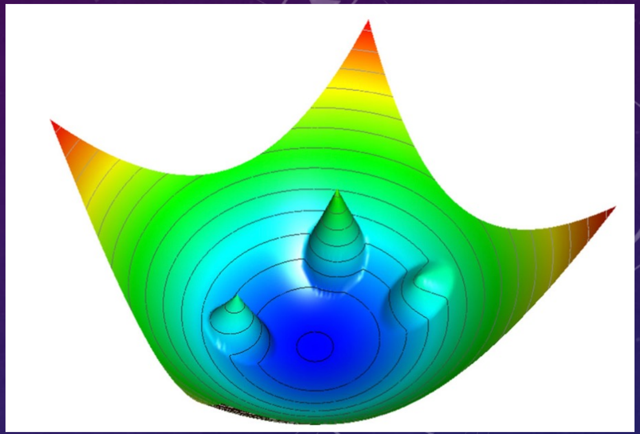


GRAVNET



- For each choice of gathering function, a new set of features $\widetilde{f_k}^i \in \widetilde{F_{LR}}$ is generated.
- The $\widetilde{F_{LR}}$ vector is concatenated to the initial vector.
- Activation function : tanh
- The F_{OUT} output carries collective information from each vertex and its surrounding.

Object Condensation



- Get the output from GravNet as β and output whether the hit seems to be a representative point of the particle ($0 < \beta < 1$)
- Employs two terms as Loss terms to improve cluster and background identification

$$L = L_V + L_\beta$$

- L_V : The closer the hit is to a particle with high β and belonging to the same particle, the smaller it is, and the more it belongs to a different particle, the larger it is.
  → Equivalent to the attractive and repulsive forces acting on an electric charge
- L_β : Converge β to 1 for only one of each particle corresponding to a true cluster
  The remaining β works its way closer to 0

LOSS FUNCTION - NETWORK LEARNING -

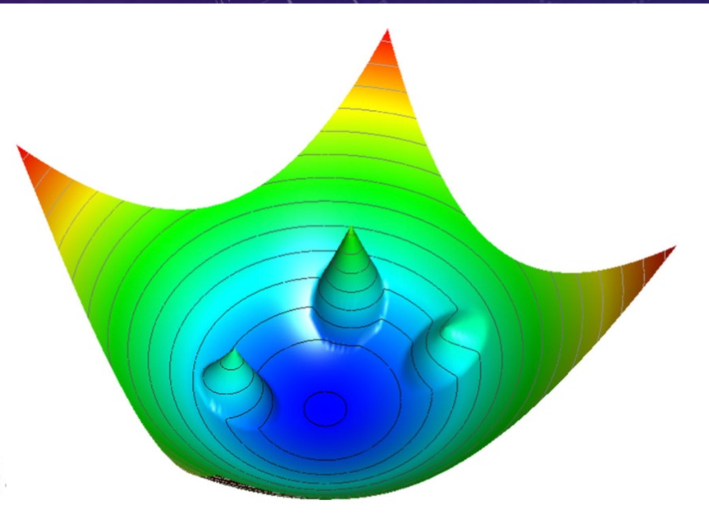
- The value of β_i $(0 < \beta_i < 1)$ is used to define a charge q_i per vertex i

$$q_i = \operatorname{arctanh}^2 \beta_i + q_{\min} \quad (\beta_i \to 1 : q_i \to +\infty)$$

- The charge q_i of each vertex belonging to an object k defines a potential $V_{ik}(x) \propto q_i$
- The force affecting vertex j can be described by

$$M_{ik} = \begin{cases} 1 \ (vertex\ i\ belonging\ to\ object\ k) \\ 0 \ (otherwise) \end{cases}$$

$$q_j \cdot \nabla V_k(x_j) = q_j \nabla \sum_{i=1}^{N} M_{ik} V_{ik}(x_j, q_i)$$



LOSS FUNCTION

- The potential of object k can be approximated :

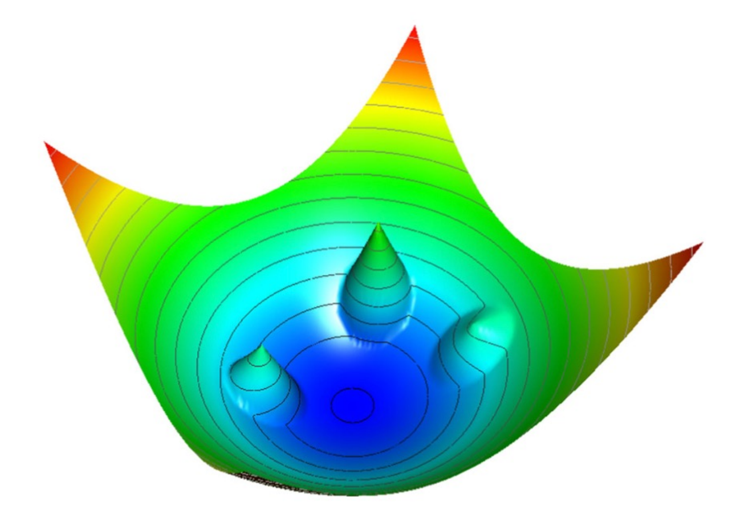
$$V_k(x) \approx V_{\alpha k}(x, q_{\alpha k}), \quad \text{with } q_{\alpha k} = \max_i q_i M_{ik}.$$

- An attractive and repulsive potential are defined as :

$$\check{V}_k(x) = \|x - x_\alpha\|^2 q_{\alpha k}, \text{ and}$$
$$\hat{V}_k(x) = \max(0, 1 - \|x - x_\alpha\|) q_{\alpha k}.$$

- The total potential loss L_V :

$$L_V = \frac{1}{N} \sum_{j=1}^{N} q_j \sum_{k=1}^{K} \left(M_{jk} \check{V}_k(x_j) + (1 - M_{jk}) \hat{V}_k(x_j) \right)$$



LOSS FUNCTION

- The L_V has the minimum value for $q_i = q_{\min} + \epsilon \; \forall i$
- To enforce one condensation point per object, and none for background or noise vertices, the following additional loss term L_β is introduced :

$$L_\beta = \frac{1}{K}\sum_k (1 - \beta_{\alpha k}) + s_B \frac{1}{N_B}\sum_i^N n_i \beta_i,$$

s_B : hyperparameter describing the background suppression strength
K : Maximum value of objects
N_B : Number of background
n_i : Noise tag (if noise, it equals 1.)

- The loss terms are also weighted by $\operatorname{arctanh}^2 \beta_i$:

$$L_p = \frac{1}{\sum_{i=1}^N \xi_i} \cdot \sum_{i=1}^N L_i(t_i, p_i)\, \xi_i, \text{ with}$$
$$\xi_i = (1 - n_i) \operatorname{arctanh}^2 \beta_i .$$

p_i: Featutes
$L_i(t_i, p_i)$: Loss term (Difference between true labels and outputs of network)