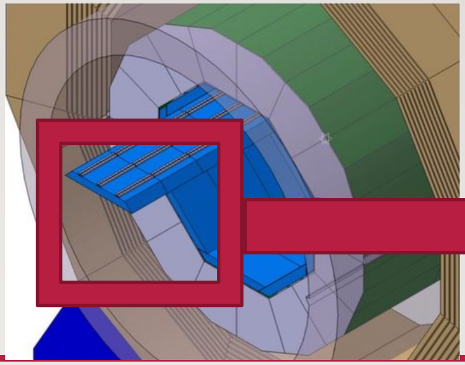
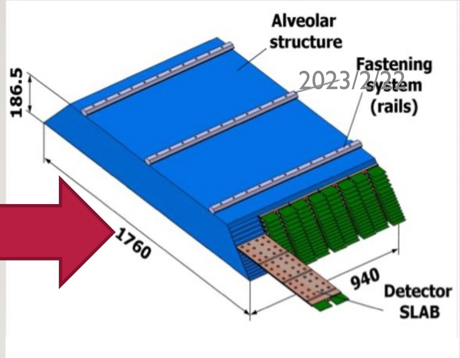
Clustering of Calorimeter Hits with GravNet

Kyushu Univ.[A], Osaka Univ. RCNP [B], Osaka Univ. IDS[C], Kyushu Institute of Technology [D]
Shusaku Tsumura [A], Taikan Suehara [A], Kiyotomo Kawagoe [A],
Hajime Nagahara [B,C], Yuta Nakashima [B,C], Noriko Takemura[c,d]

2 ILD / SiW ECAL





- Electromagnetic calorimeter (ECAL): Detects positions , and energy of gamma rays
  → Higher accuracy of particle identification: PFA
- ECAL equips a lot of channels (~10^8) to identify each particle.
- Sandwich structure with 30 alternating layers of Si detection layer and W absorption layer.
- W-absorbing layer: Electromagnetic shower is induced when electrons and gamma rays are incident.
  → ~ 24 X_0 in total
- Feature: Moliere radius is small enough to separate each particle

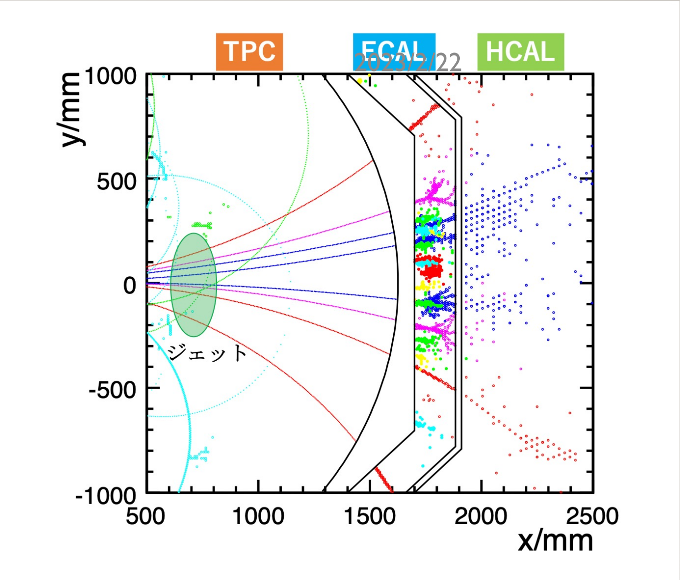
3 Particle Flow Algorithm (PFA)

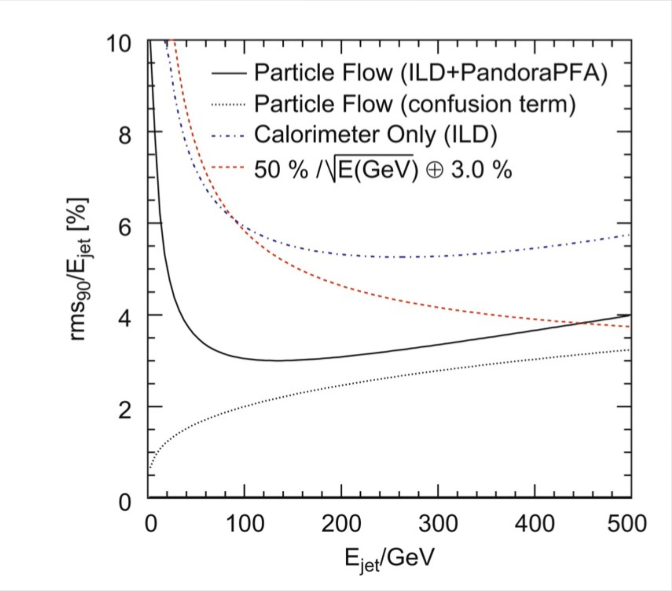
- A method to obtain higher jet energy resolution by reconstructing the particle trajectory for each type of particle in the jet.
- Charged particles: Tracker
- Photons : ECAL
- Neutral hadrons : HCAL
- Resolution of a calorimeter for a single particle :

Perfect PFA: $\sim 20\%/\sqrt{E(GeV)}$

PandoraPFA : $\sim 30\%/\sqrt{E(GeV)}$ (< 100 GeV)

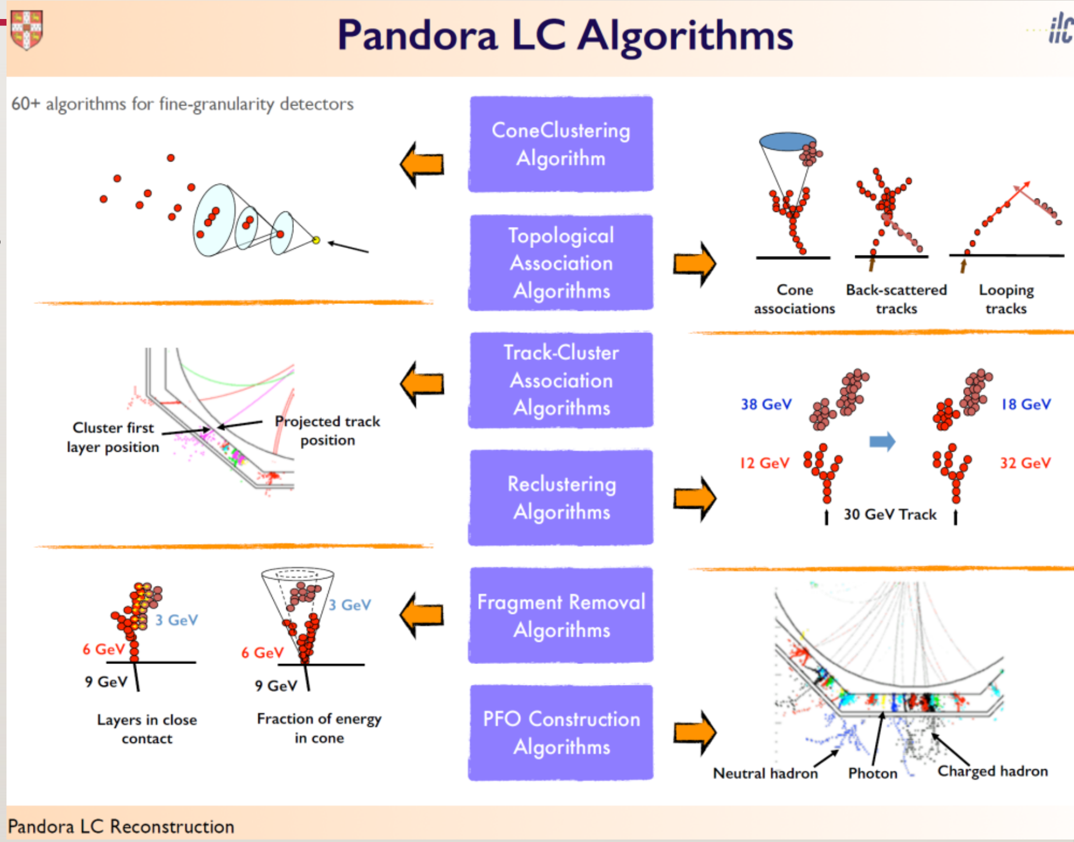
w/o PFA : $50 - 60\%/\sqrt{E(GeV)}$





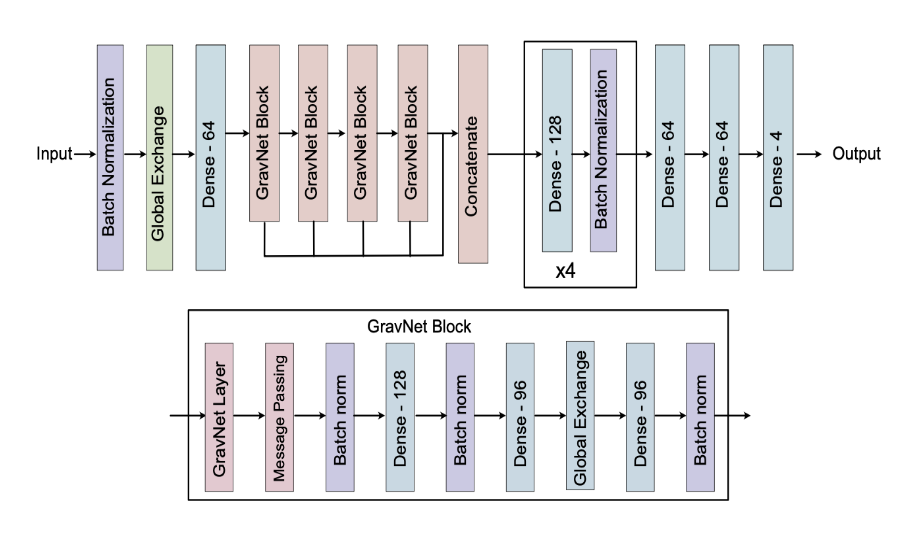
4 Application of Deep Learning to PFA

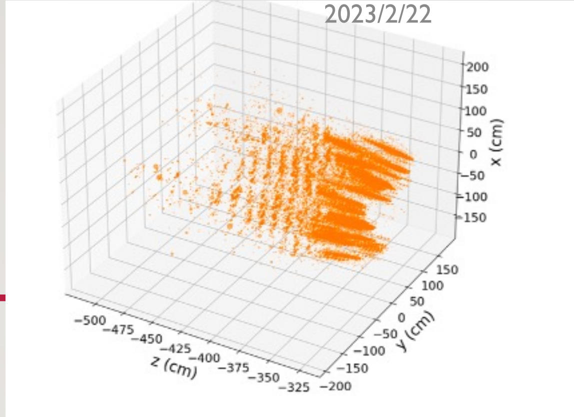
- Current PFA algorithm : PandoraPFA
  →The pattern recognition based on the human-tuned parameters
- We may achieve better accuracy by considering the hidden and complicated relationships among the hit information
- Aim to further improve performance by using deep learning technique.

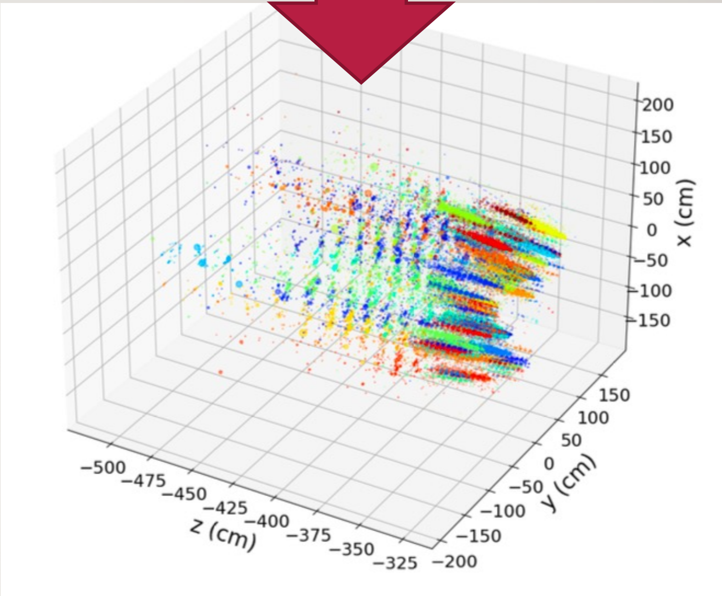


5 Calorimeter Clustering

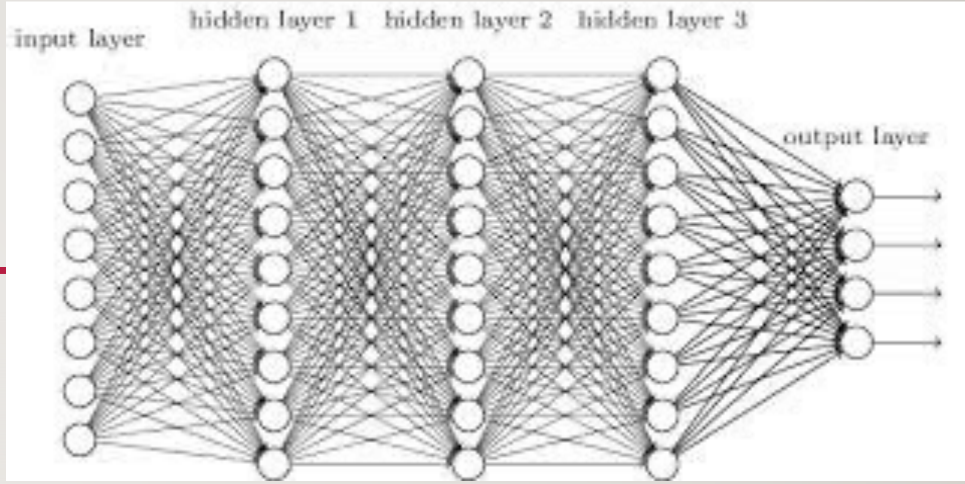
- Input: feature values of hits in the calorimeter e.g., position, energy, etc.
  → discriminate each cluster
- Deep Learning Architecture
  - Developed for a CMS detector that has a lot of separated channels for PFA







6 Deep Learning

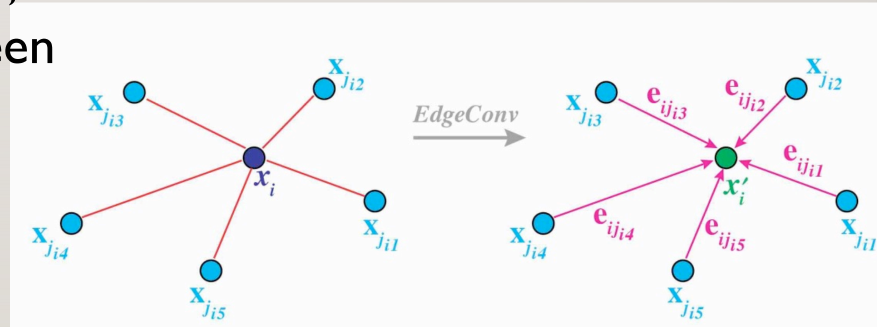


Fully Connected Layer

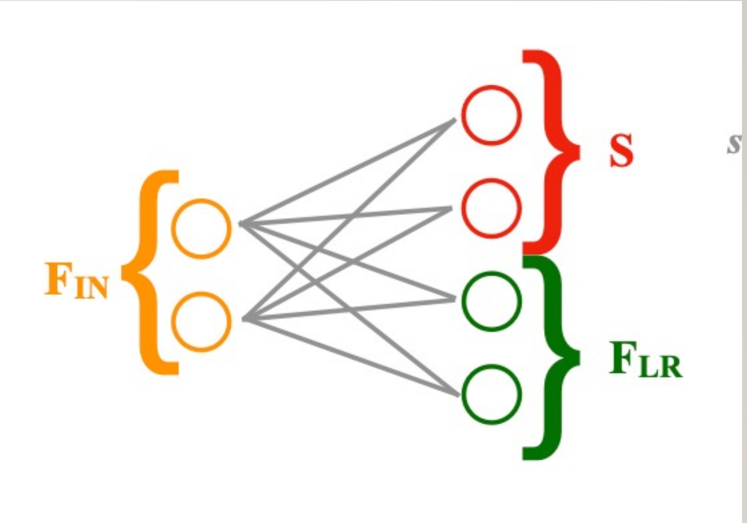
- One of the most basic structures in deep learning
- Consists of an input layer, a hidden layer, and an output layer
- A more expressive network can be built by increasing the number of layers

Graph Neural Network

- A network is constructed as a graph consisting of nodes (points) and edges (lines)
- Not only can it learn the features of materials with a graph-like structure, but it can also be used in many ways, such as expressing the relationship between features as a graph.



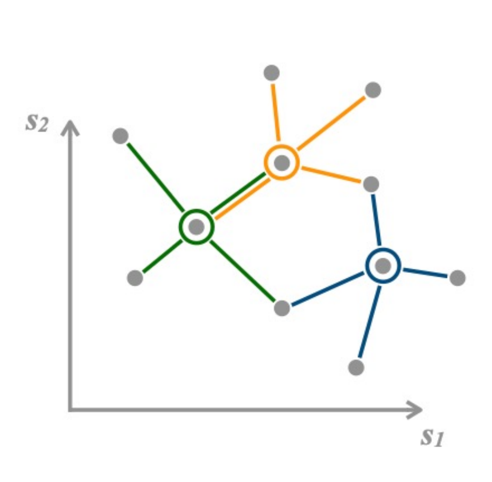
7 GravNet



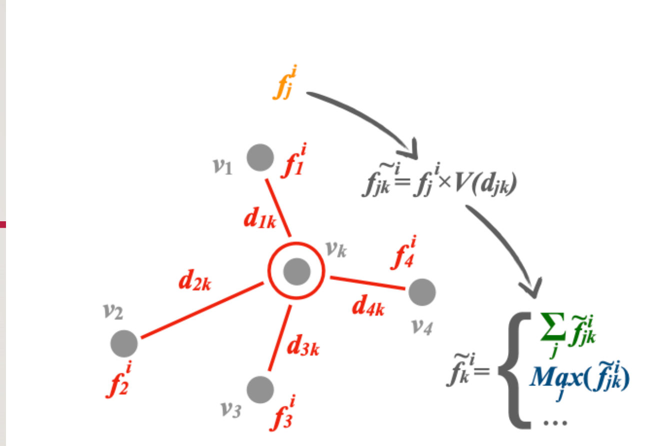
- Input Data : $V \times F_{IN}$

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

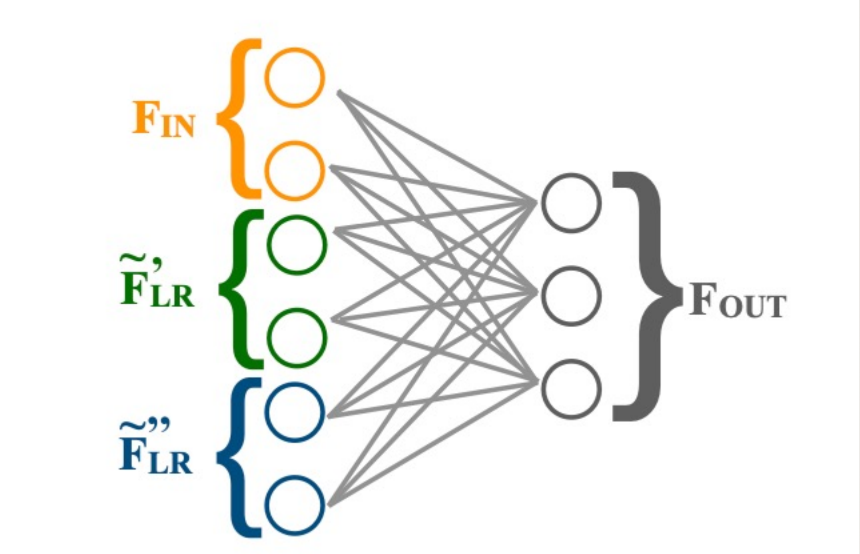
- S : Set of coordinates in some learned representation space
- F_{LR} : learned representation of the vertex features
- Input data of initial dimension $V \times F_{IN}$ is converted into a graph.
- The coordinates of the graph is updated by the learning of the ne

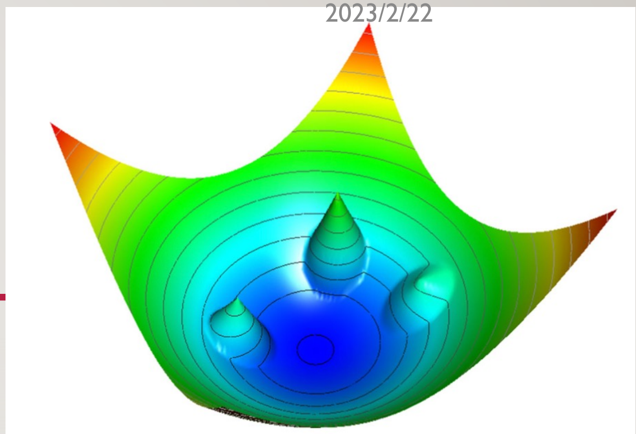


8 GravNet



- The contribution of each point is bigger depending on the distance between the points
- The output is calculated for each point based on the contribution
- At last, the outputs ($\widetilde{F_{LR}}$) are concatenated with the initial inputs and previous outputs and pass the FC layer.
- The F_{OUT} output carries collective information from each vertex and its surrounding.





9 Object Condensation

- A loss function technique to recognition for multi-object
- 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

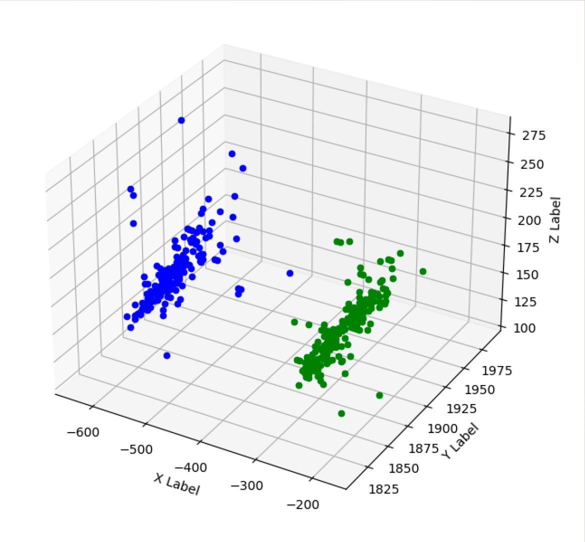
$$L = L_V + L_\beta$$

- 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

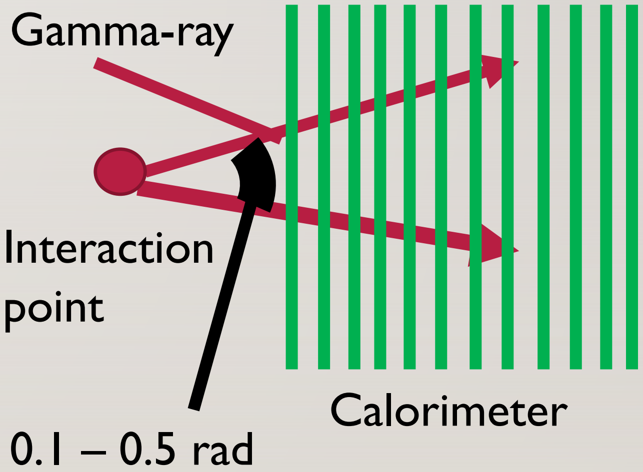
Generation of Input Data

- Two gamma events are generated by the simulation software
- 10000 Events are generated for each of the five data sets every 0.1 rad from 0.1 to 0.5 rad
- θ : 85/180 π , ϕ : random

Two gamma event

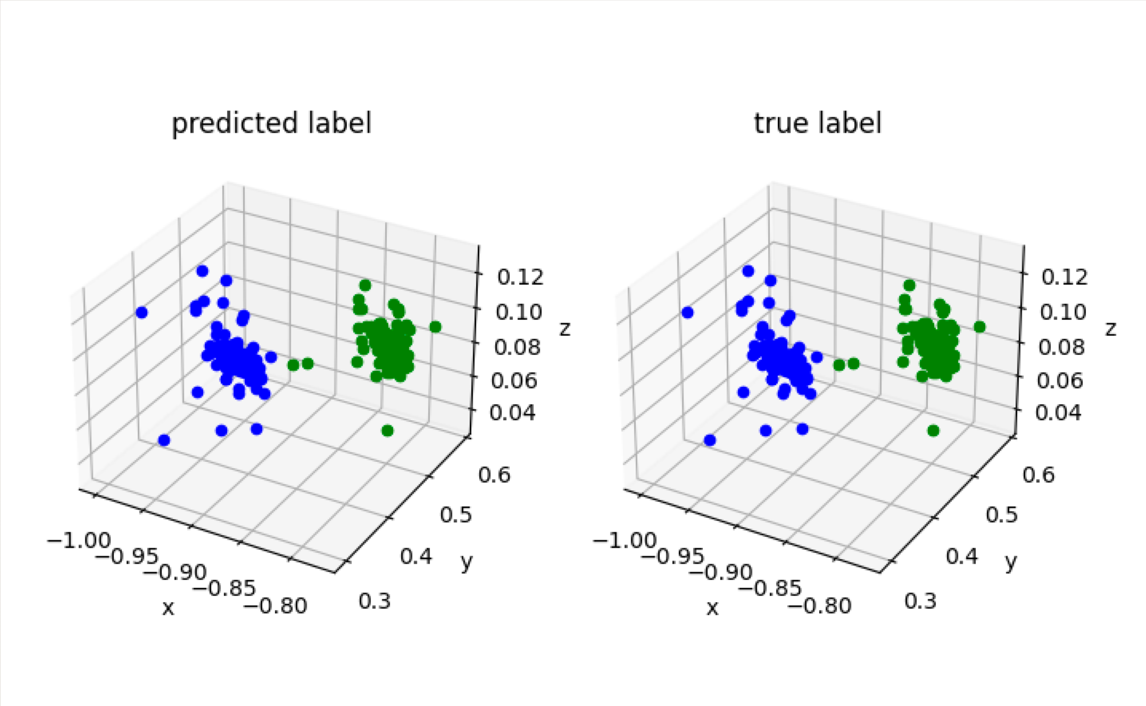


Generation of MC particles → Simulation based on detector geometry by ddsim → Reconstruction of hits in the detector by Marlin



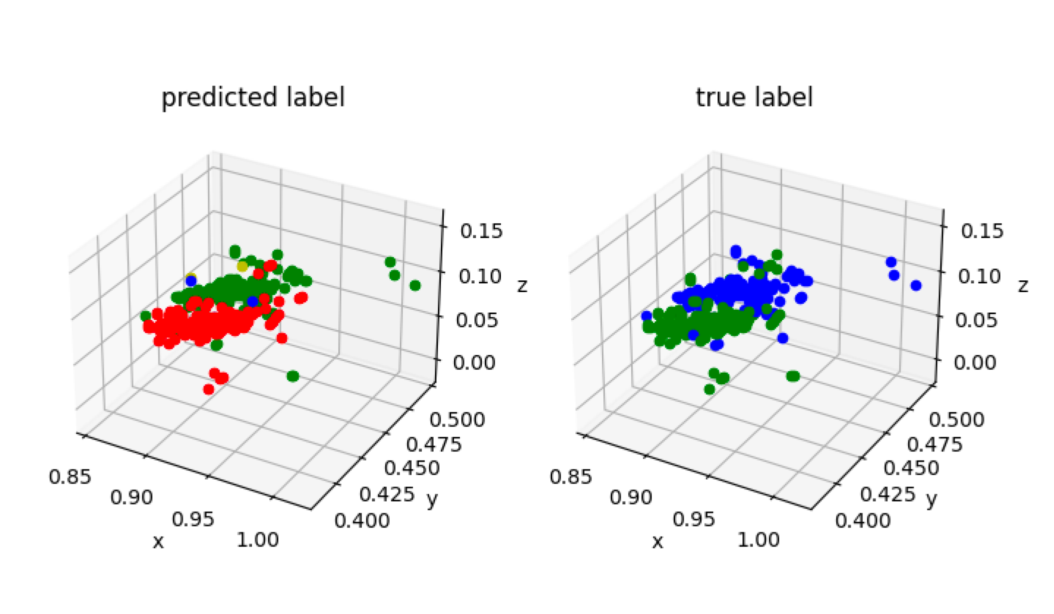
Event Display

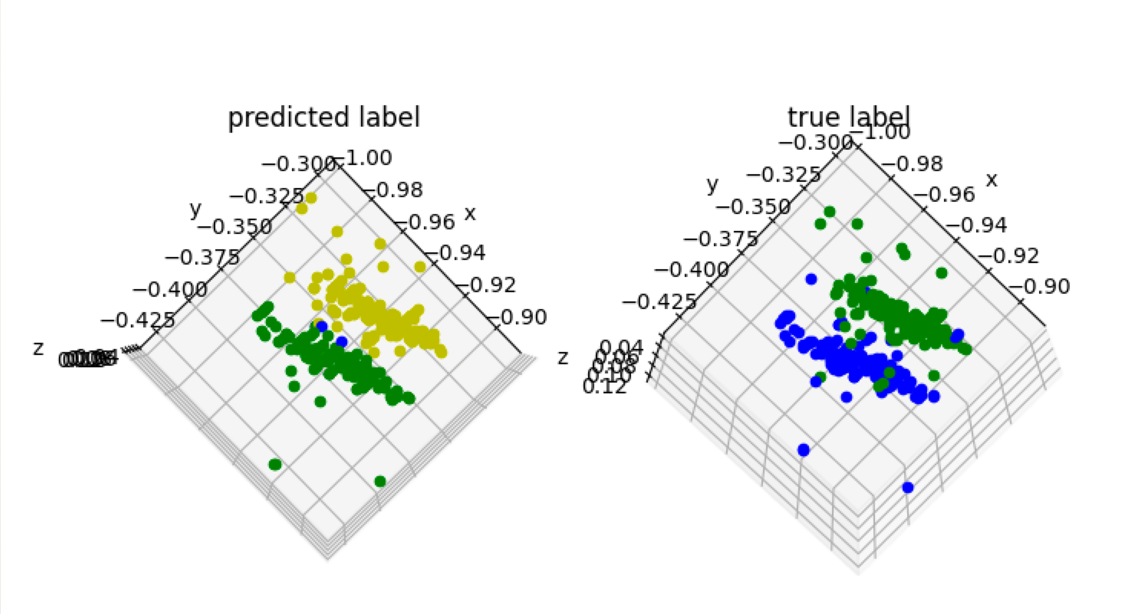
- Cluster identification resulting from learning (test data):



12 Event Display

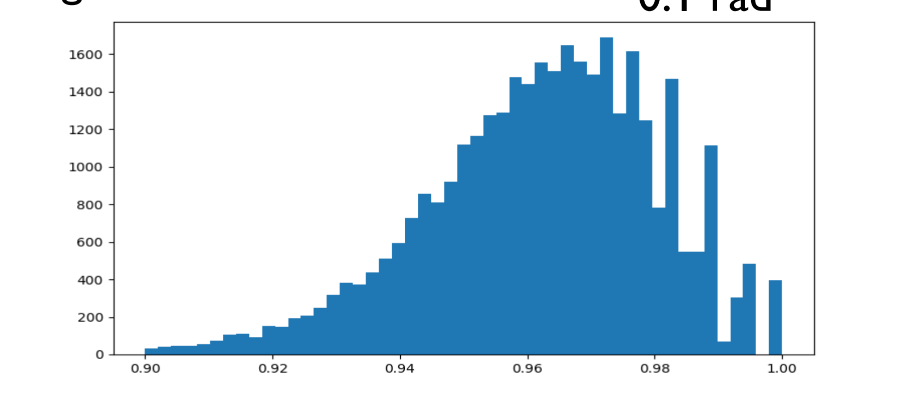
- Cluster identification resulting from learning for small opening angles(test data)：





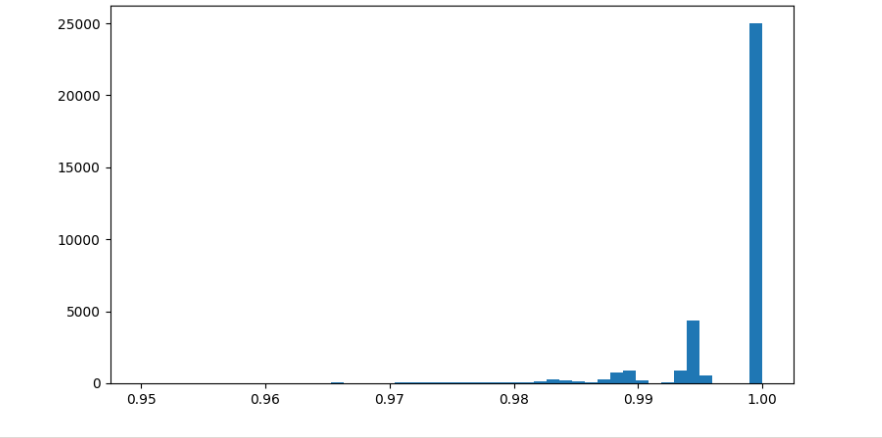
Evaluation of Network

Average = 96.08%



- Accuracy : $\dfrac{\text{Number of hits which is predicted correctly}}{\text{Number of hits with true label of each cluster}}$

Average = 99.56%



- The simulation data includes events where photons are converted into other particles.
- As input data, events with only two clusters are selected

Angle[rad]	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
Accuracy[%]	96.08	98.64	99.30	99.68	99.56

14 Summary

- Graph Neural Networks are applied to the PFA and shower clustering algorithms in the ILC analysis framework.
- The two gamma events are generated and the GravNet architecture is applied.
- The training results show an accuracy of more than 90% for each angle.

Plan :

- This architecture will be examined with more realistic events (jets etc.) for performance comparison with the current PFA algorithm

BACKUP

16 GRAVNET - NETWORK -

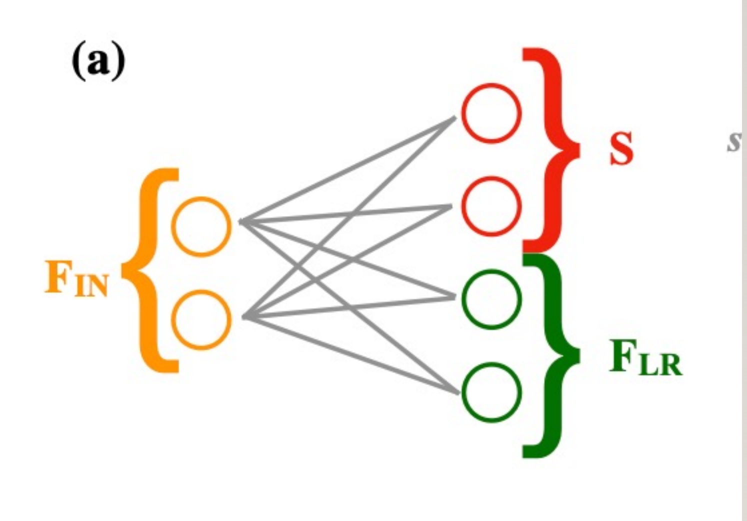
- Input Data : $B \times V \times F_{IN}$

  B : Number of examples including in a batch

  V : Number of hits for each detector

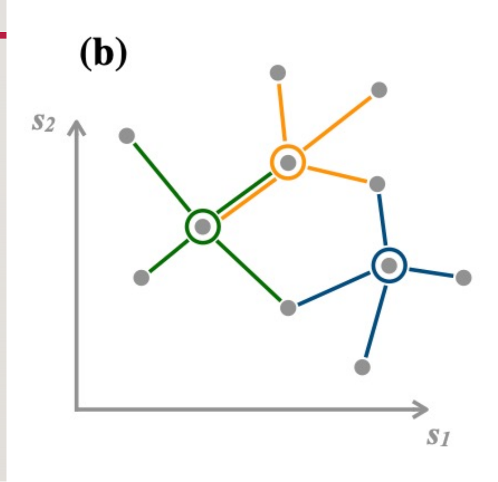
  $\mathrm{F_{IN}}$: Number of the features for each hit

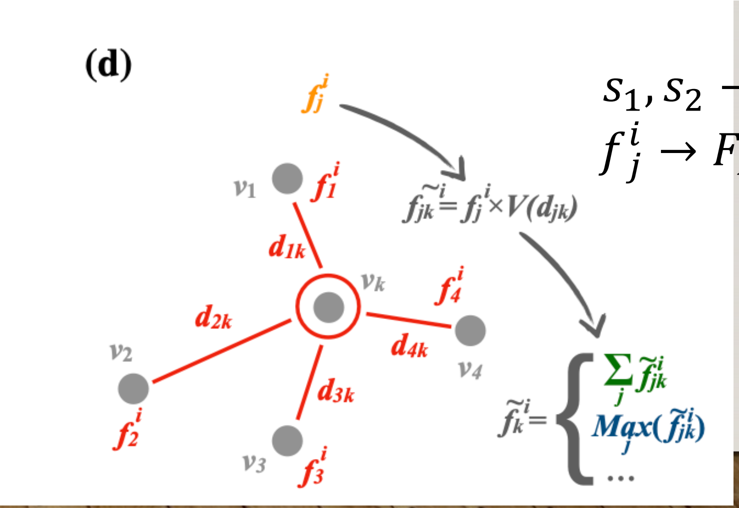
- S : Set of coordinates in some learned representation space
- F_{LR} : learned representation of the vertex features



17 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

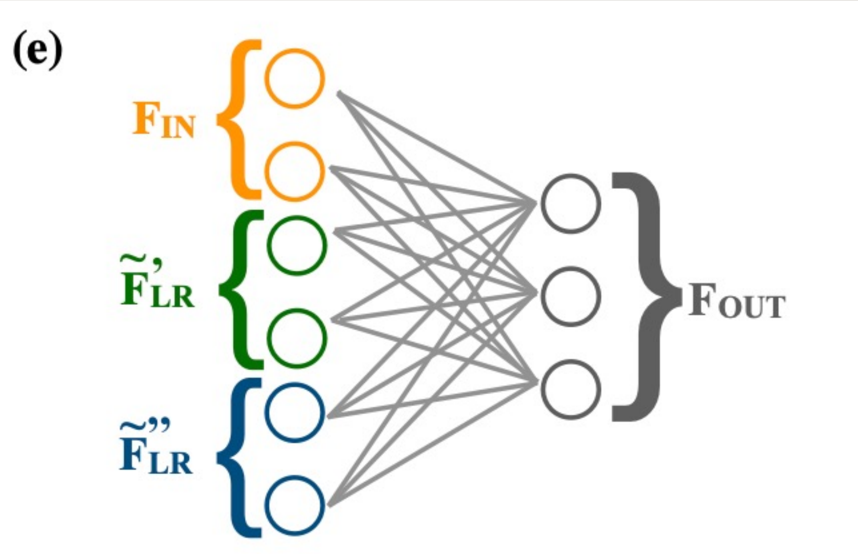




$s_1, s_2 \rightarrow S$

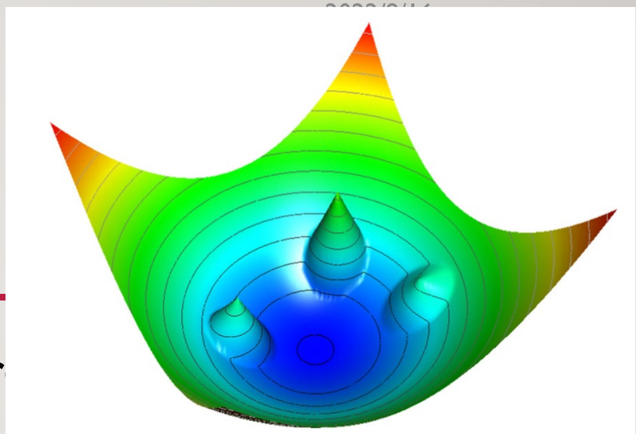
$f_j^i \rightarrow F_{LR}$

GRAVNET



- For each choice of gathering function, a new set of featur
- 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 r
  point of the particle (0 < β < 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

20 LOSS FUNCTION - NETWORK LEARNING -

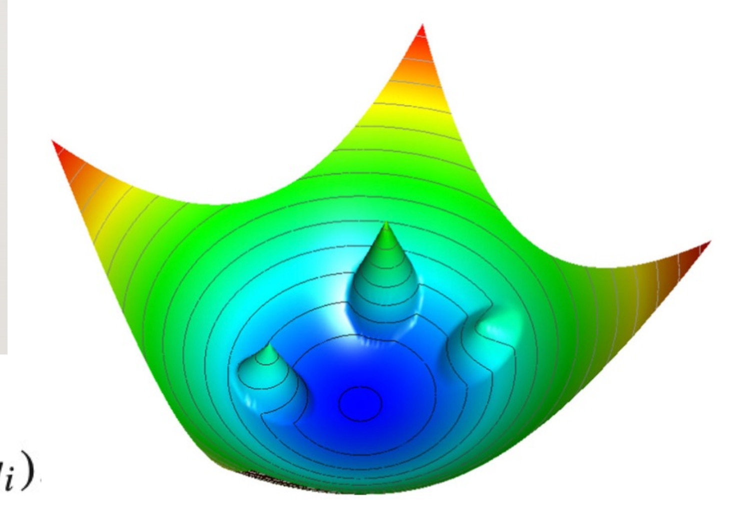
- The value of β_i $(0 < \beta_i < 1)$ is used to define a charge q_i per vertex i

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

- 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

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

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



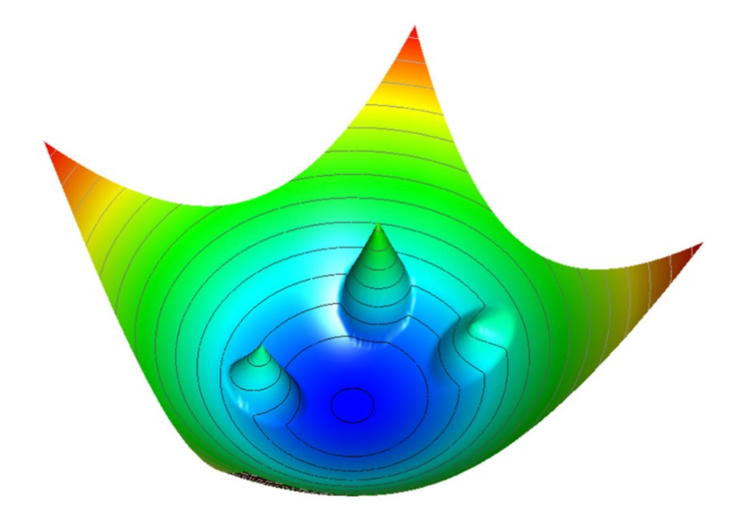
21 LOSS FUNCTION

- 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 :

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

22 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.)

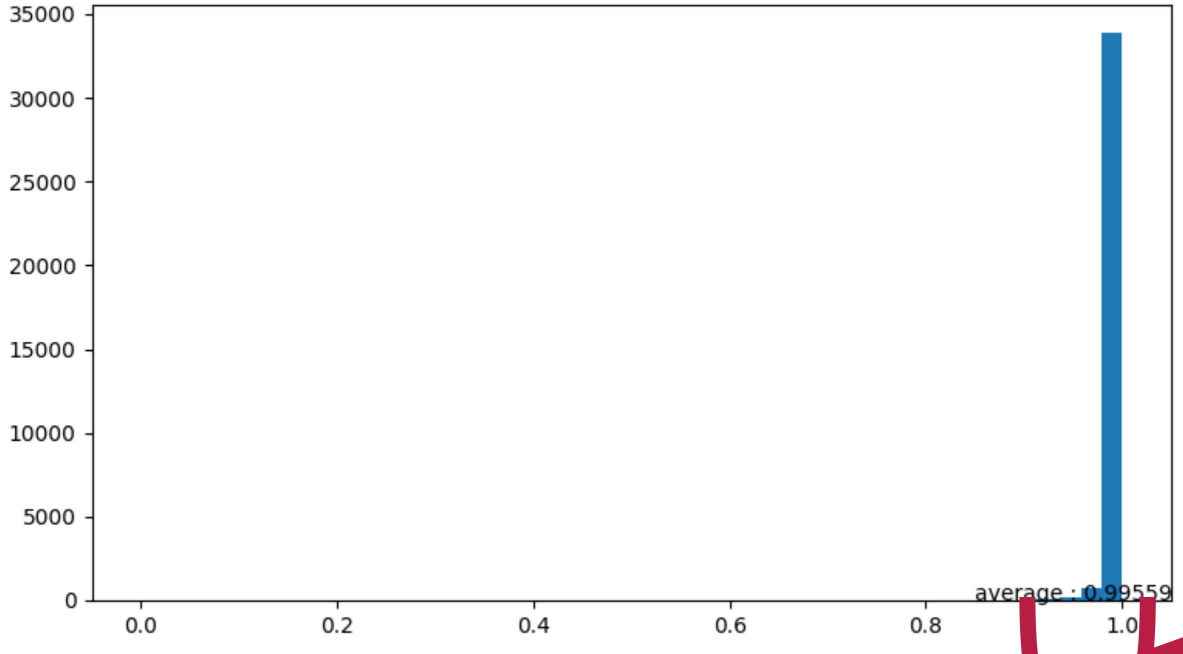
- The loss terms are also weighted by $\operatorname{arctanh}^2 \beta_i$:

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

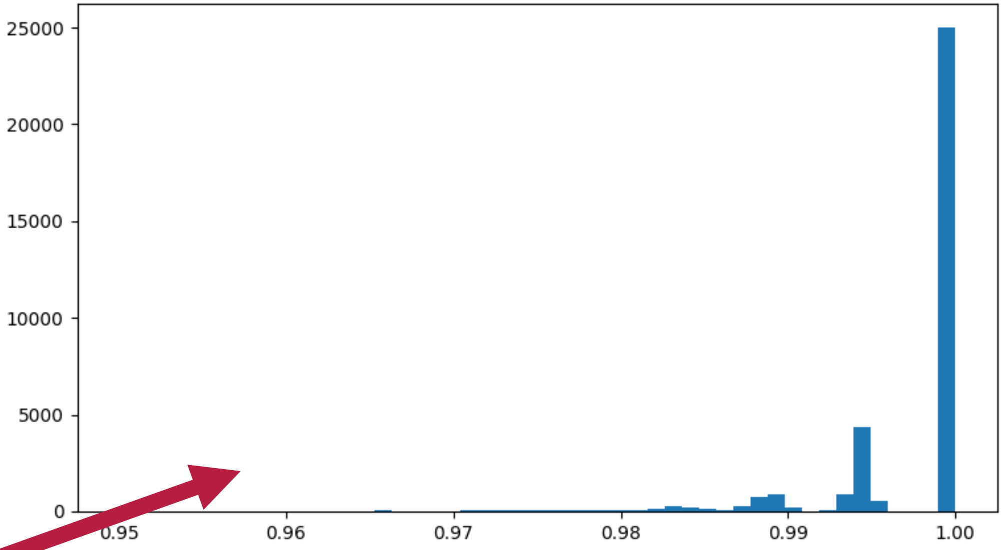
p_i: Featutes
$L_i(t_i, p_i)$: Loss term (Difference between true labels and outputs of network)

EVALUATION

- Accuracy = $\dfrac{Number\ of\ hits\ with\ predicted\ label\ correctly}{Number\ of\ \ hits\ with\ true\ label}$
- Opening angle = 0.5 rad (the largest one)
- Event selection : events which include 2 clusters



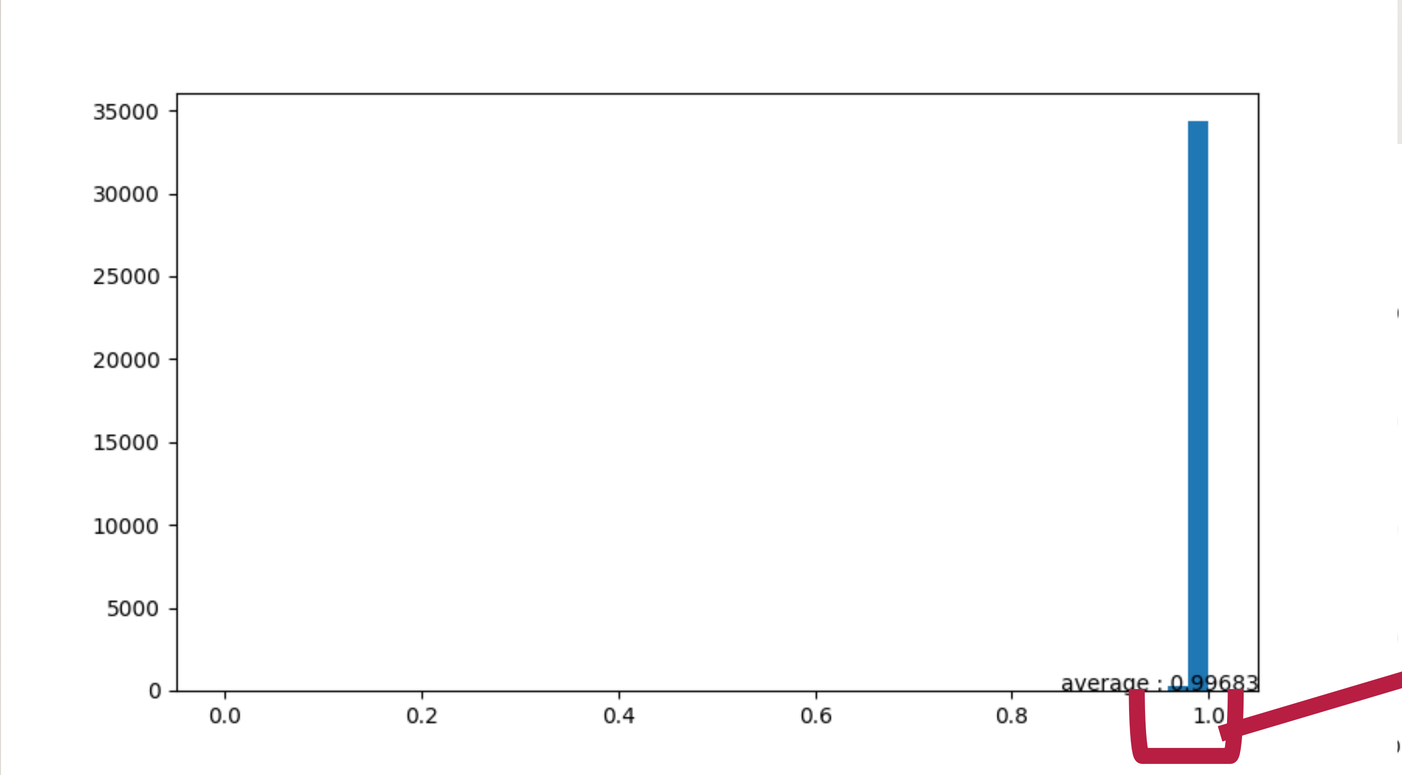
Average = 99.56%

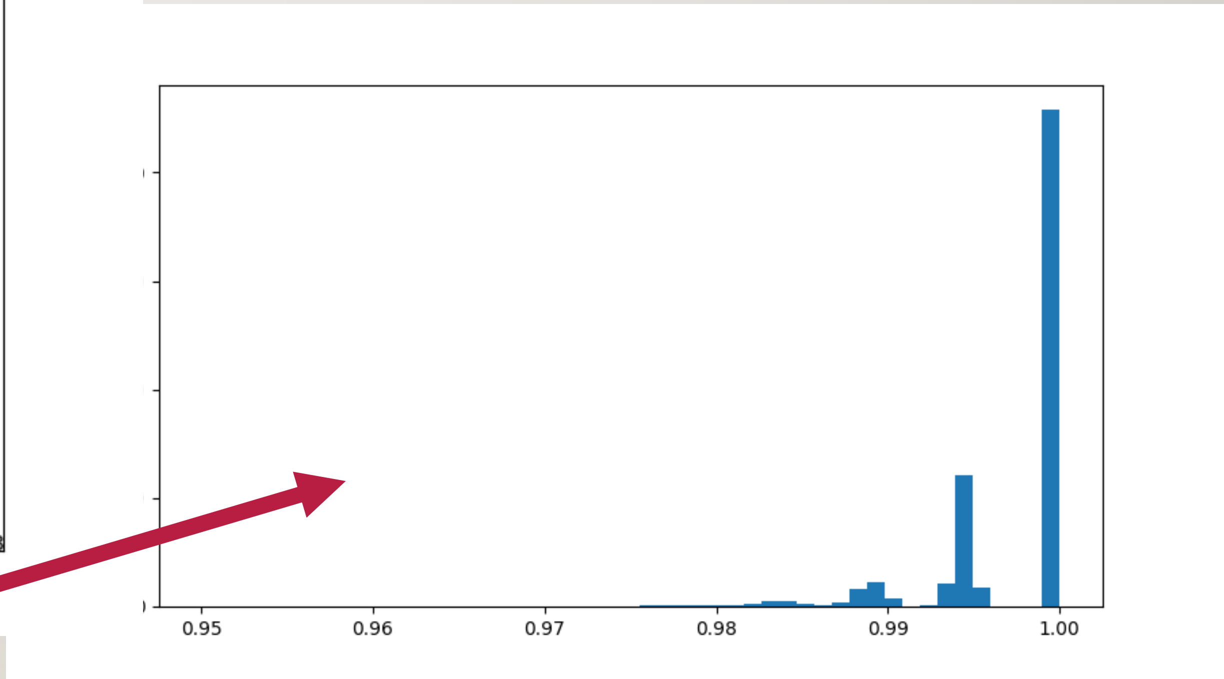


EVALUATION

Opening angle = 0.4 rad

Average = 99.68%

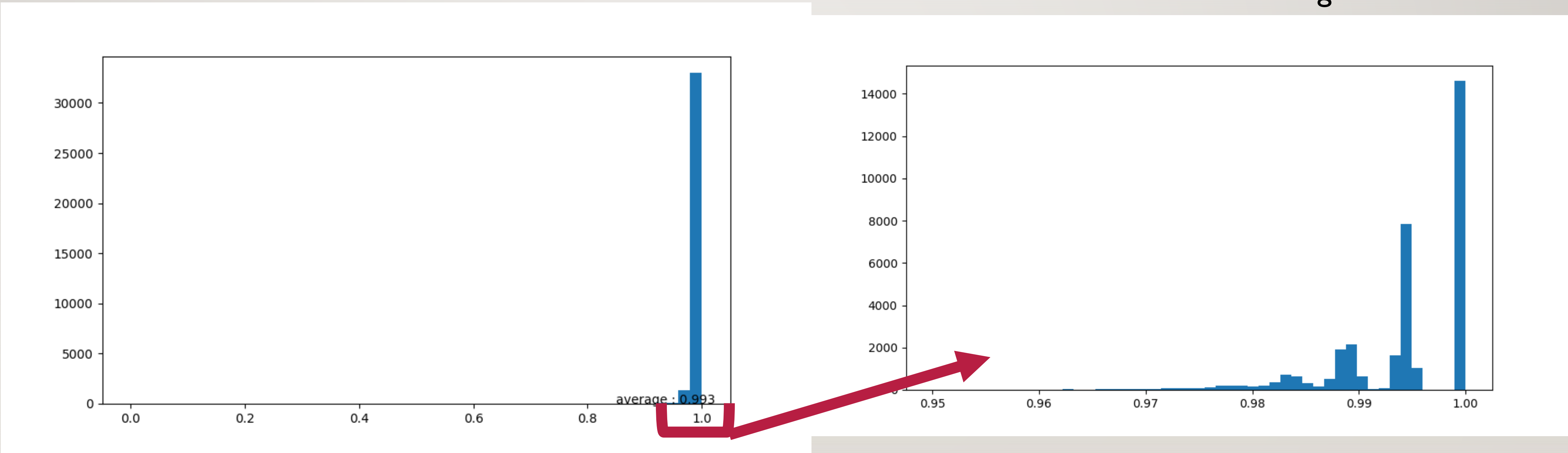




EVALUATION

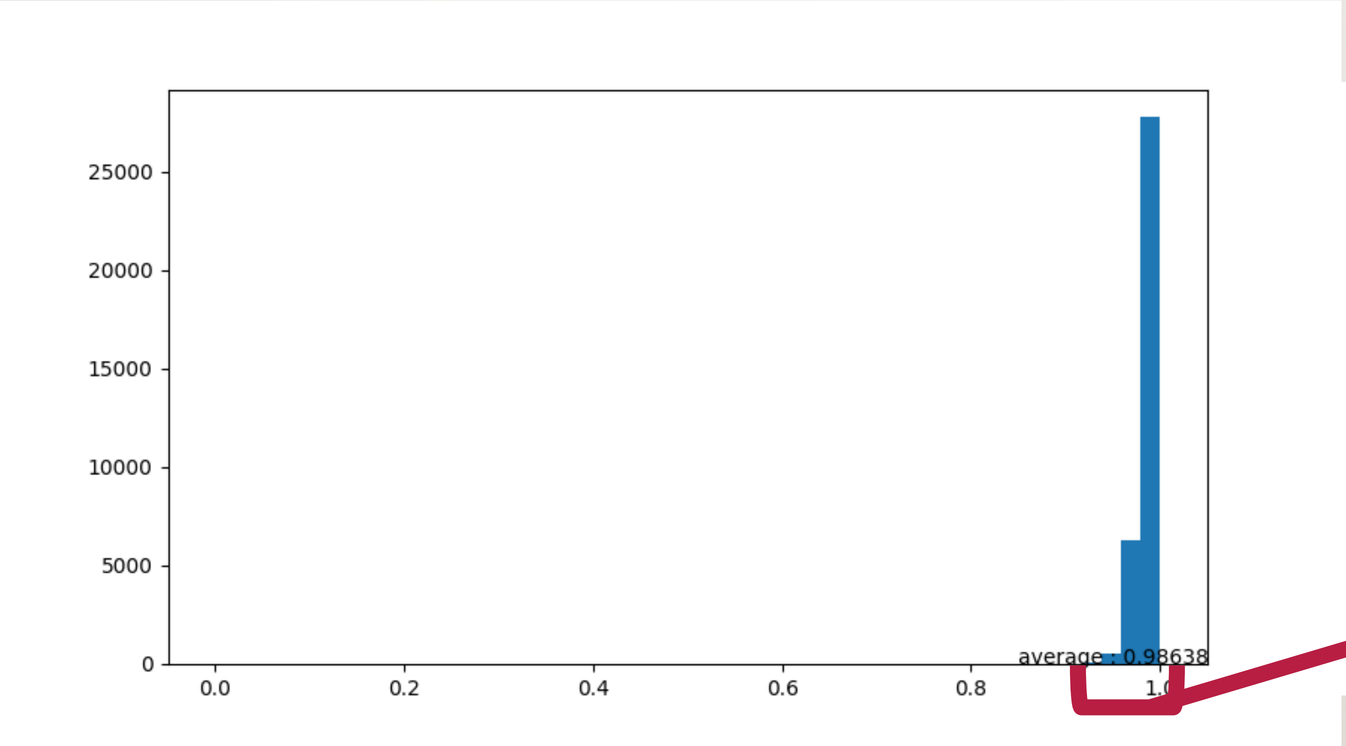
Opening angle = 0.3 rad

Average = 99.30%

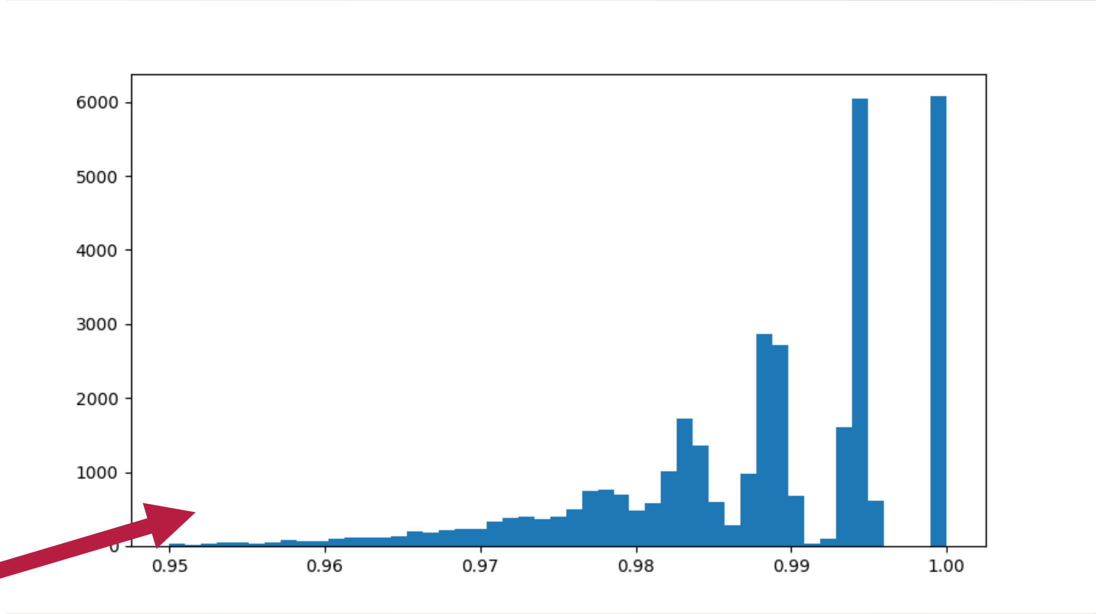


EVALUATION

Opening angle = 0.2 rad



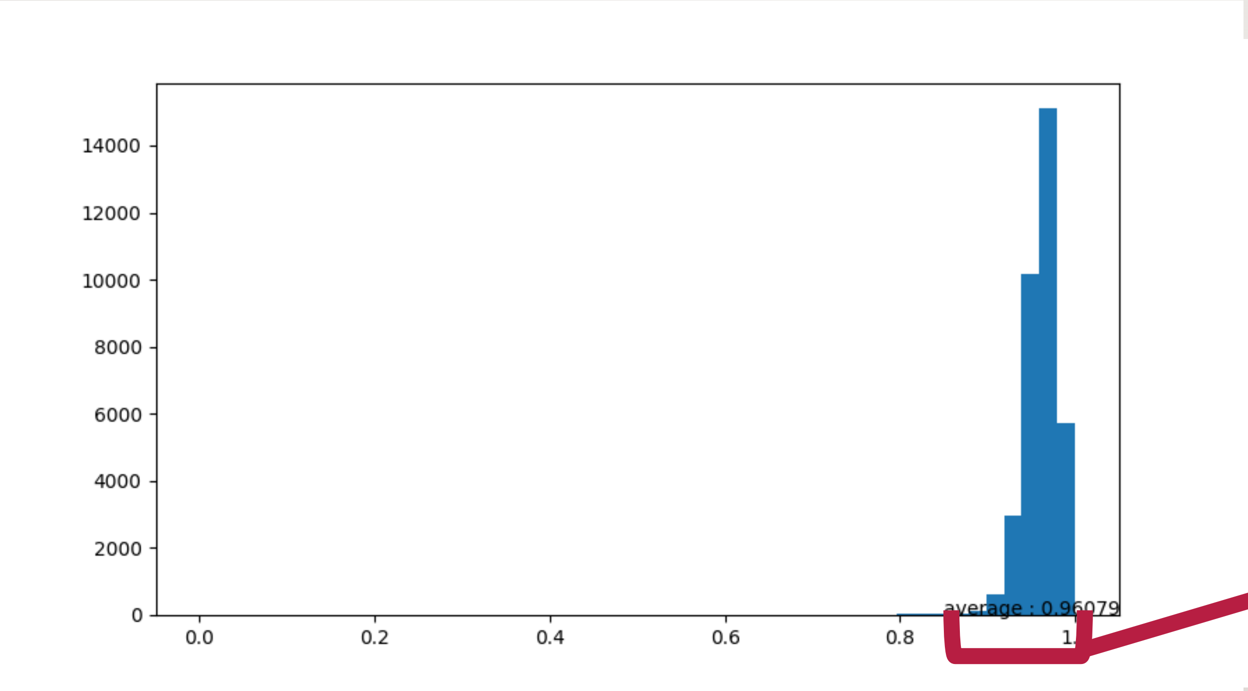
Average = 98.64%

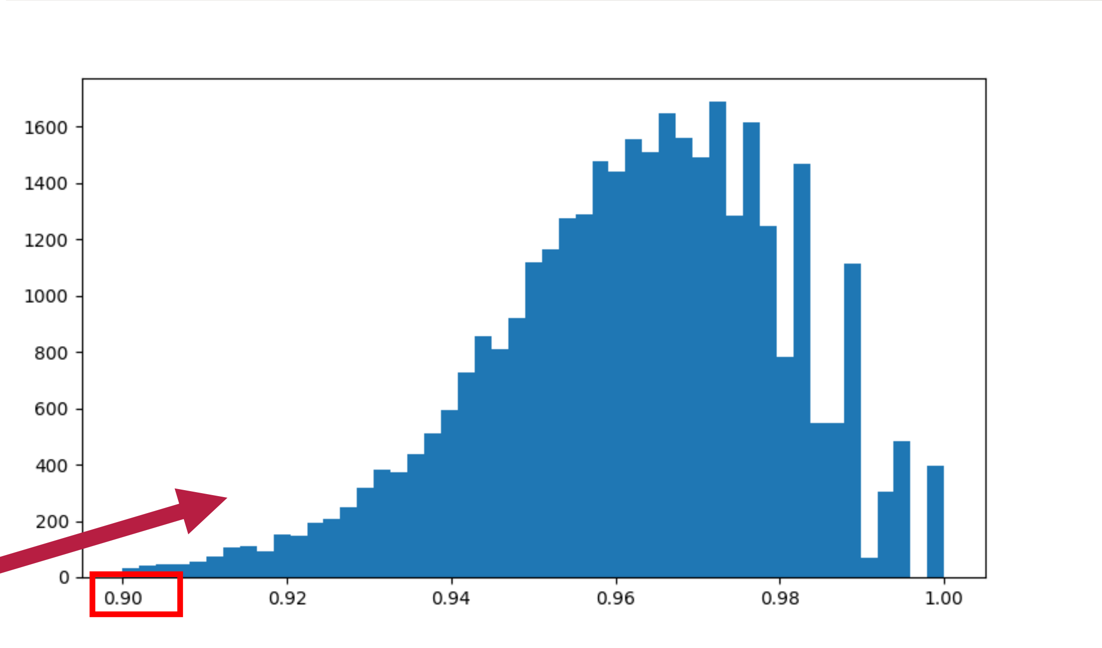


EVALUATION

Opening angle = 0.1 rad (the smallest one)

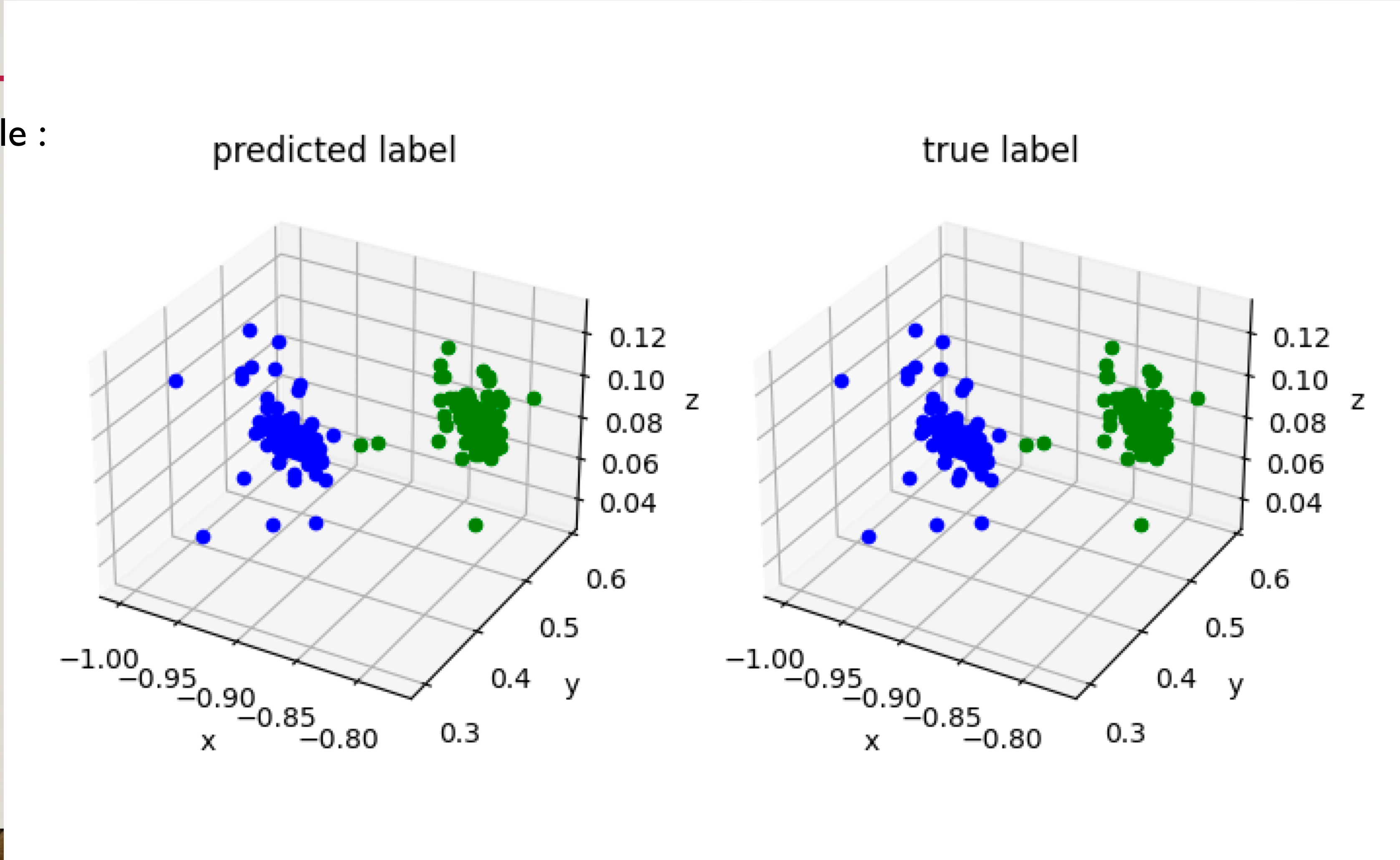
Average = 96.08%



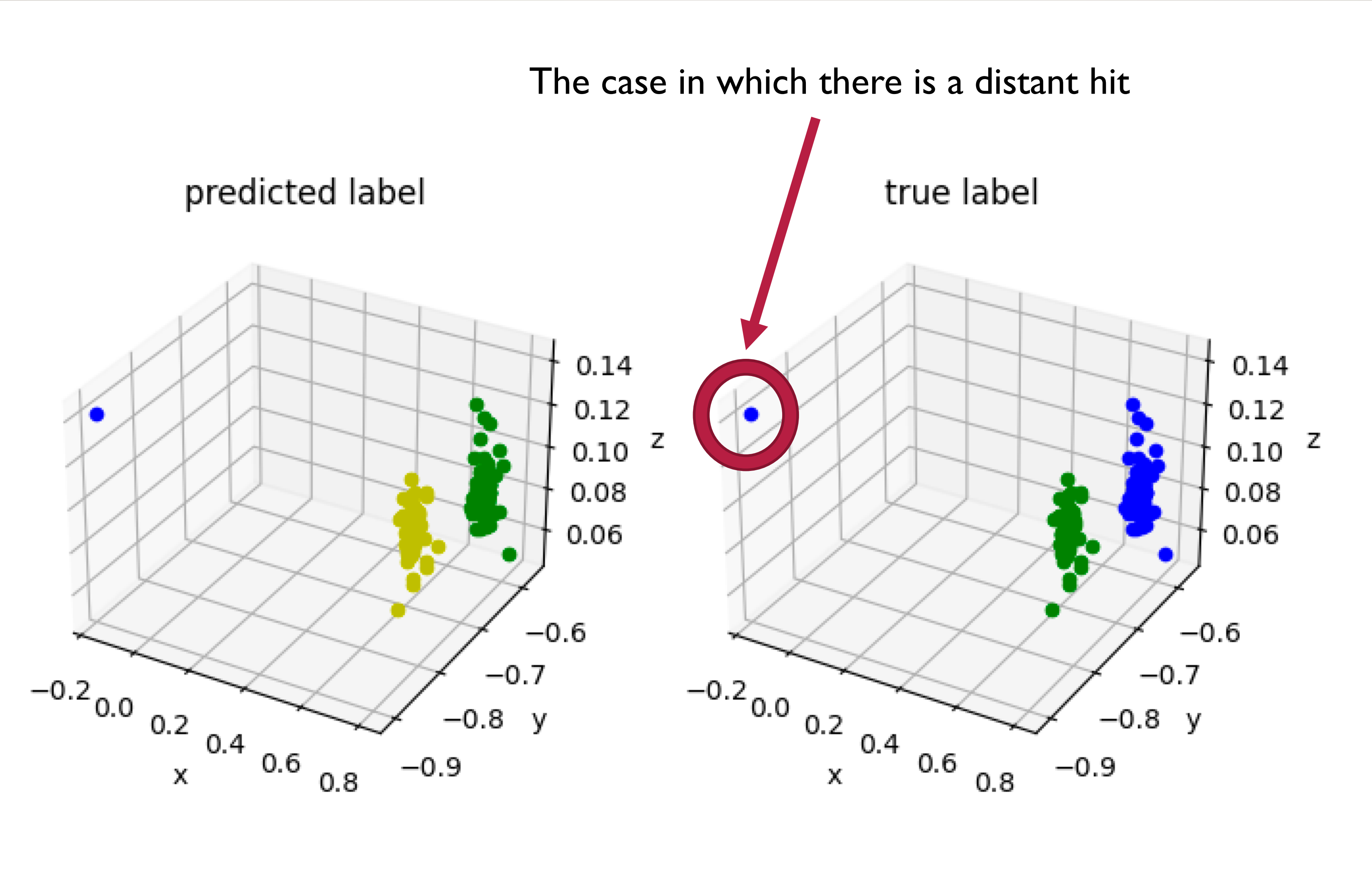


COMPARISON BETWEEN PREDICTION AND TRUE LABEL

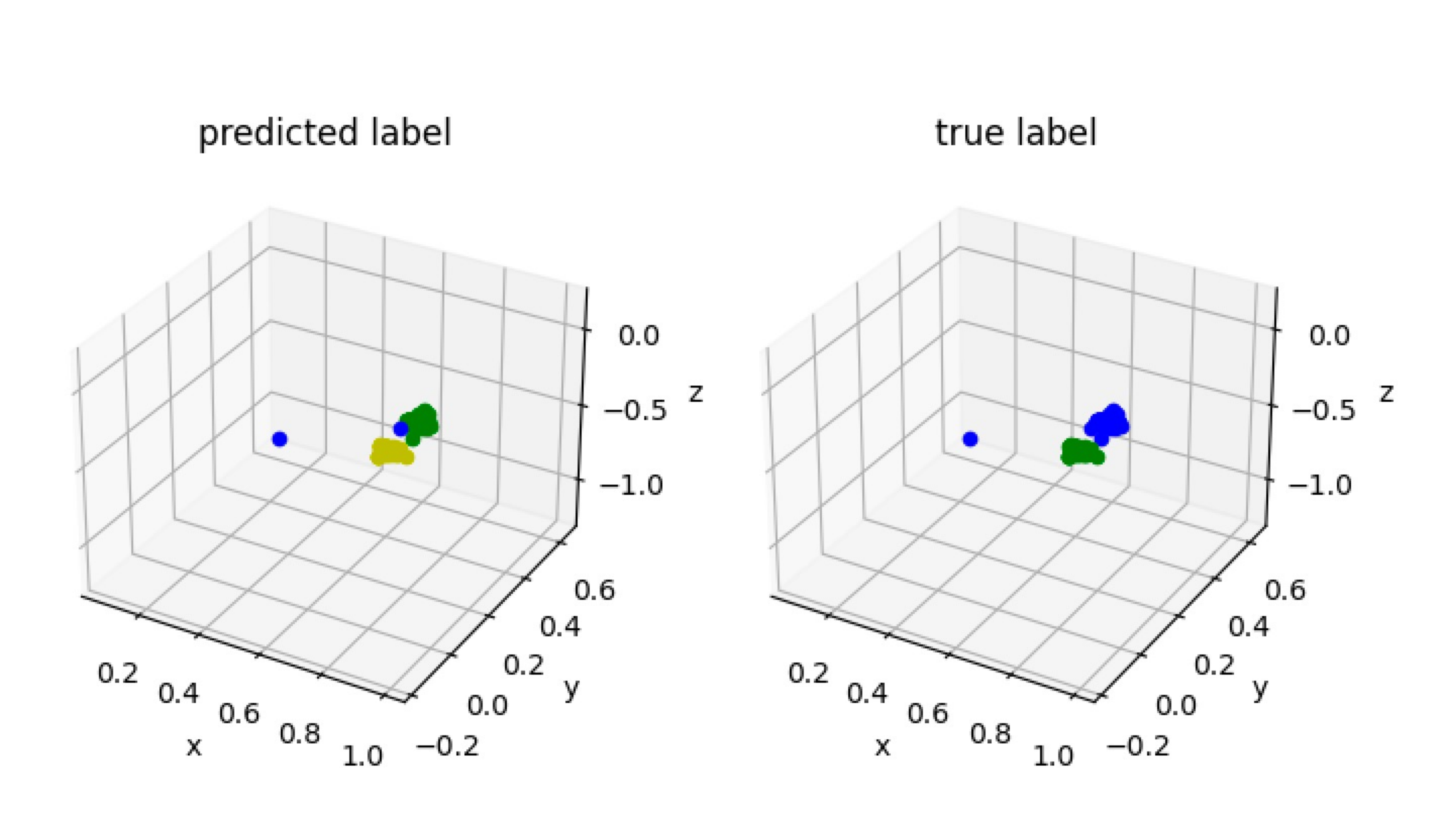
Good example :



COMPARISON BETWEEN PREDICTION AND TRUE LABEL

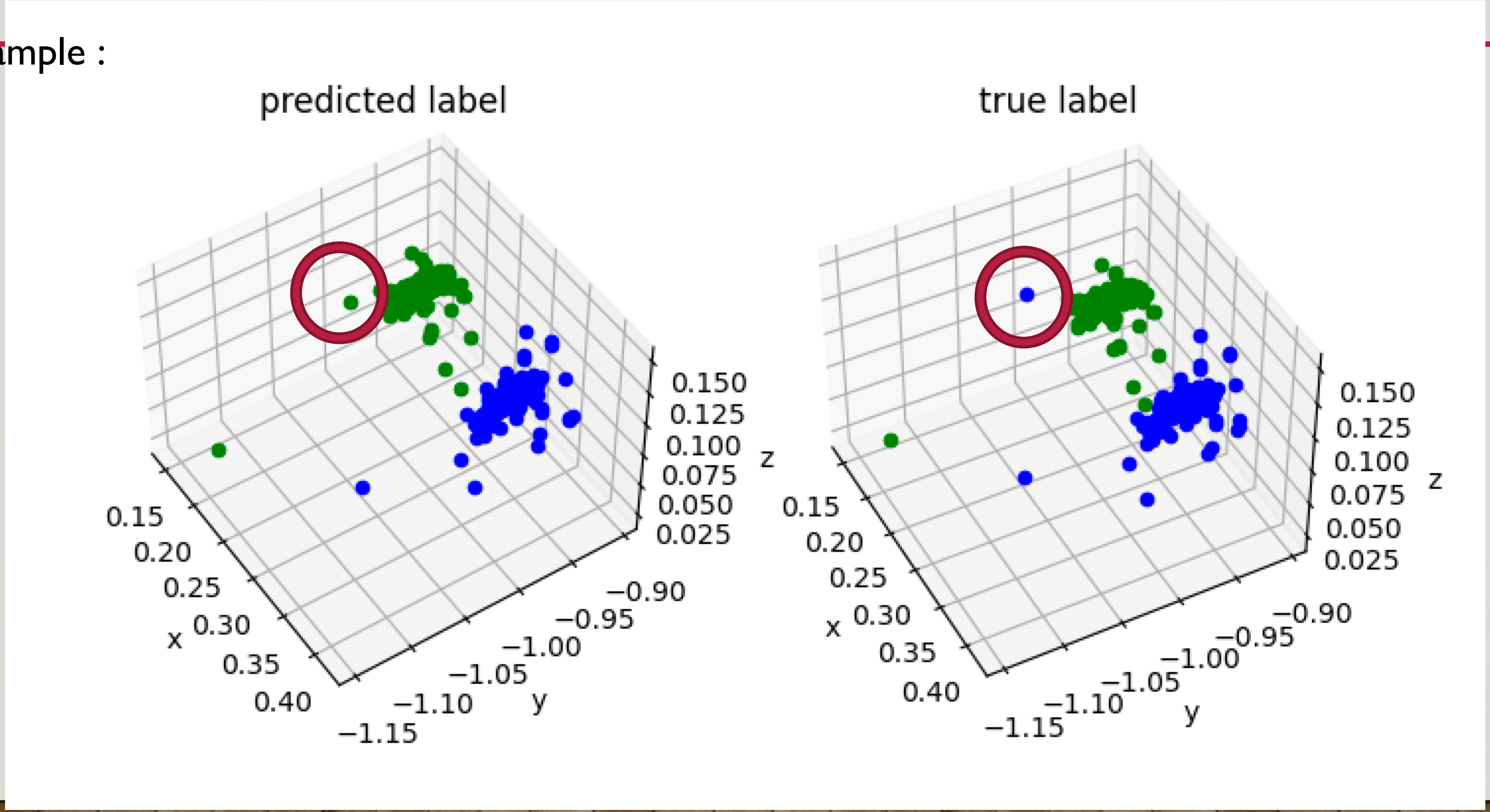


COMPARISON BETWEEN PREDICTION AND TRUE LABEL



COMPARISON BETWEEN PREDICTION AND TRUE LABEL

Confusion example :



NUMBER OF CLUSTER IN EACH EVENT(JUST 100 EVENTS)