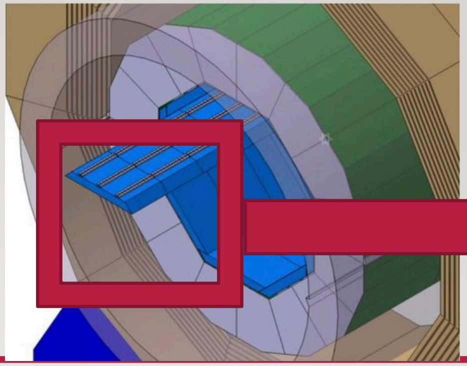
High level reconstruction with DNN for Higgs factories

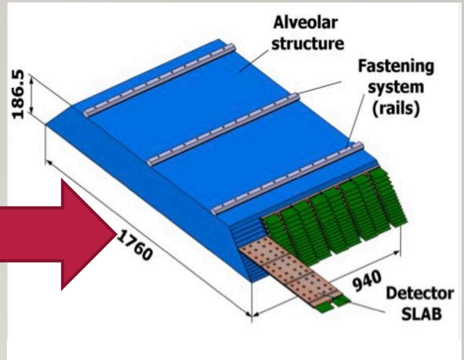
T. Suehara, S. Tsumura, T. Onoe, K. Kawagoe (Kyushu U.)
collaborating with
H. Nagahara, Y. Nakashima (Osaka U.), N. Takemura (Kyushu Tech.),
L. Gray, T. Klijnsma (Fermilab)

Contents

- Calorimeter clustering with GravNet/Object condensation
  - Slides from S. Tsumura
- b/c tagging with Graph Attention Network
  - Slides from T. Onoe

3 ILD / SiW ECAL

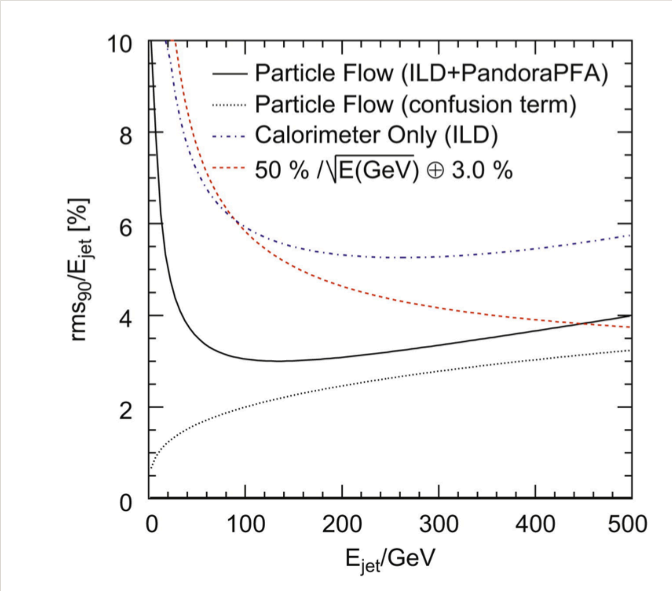


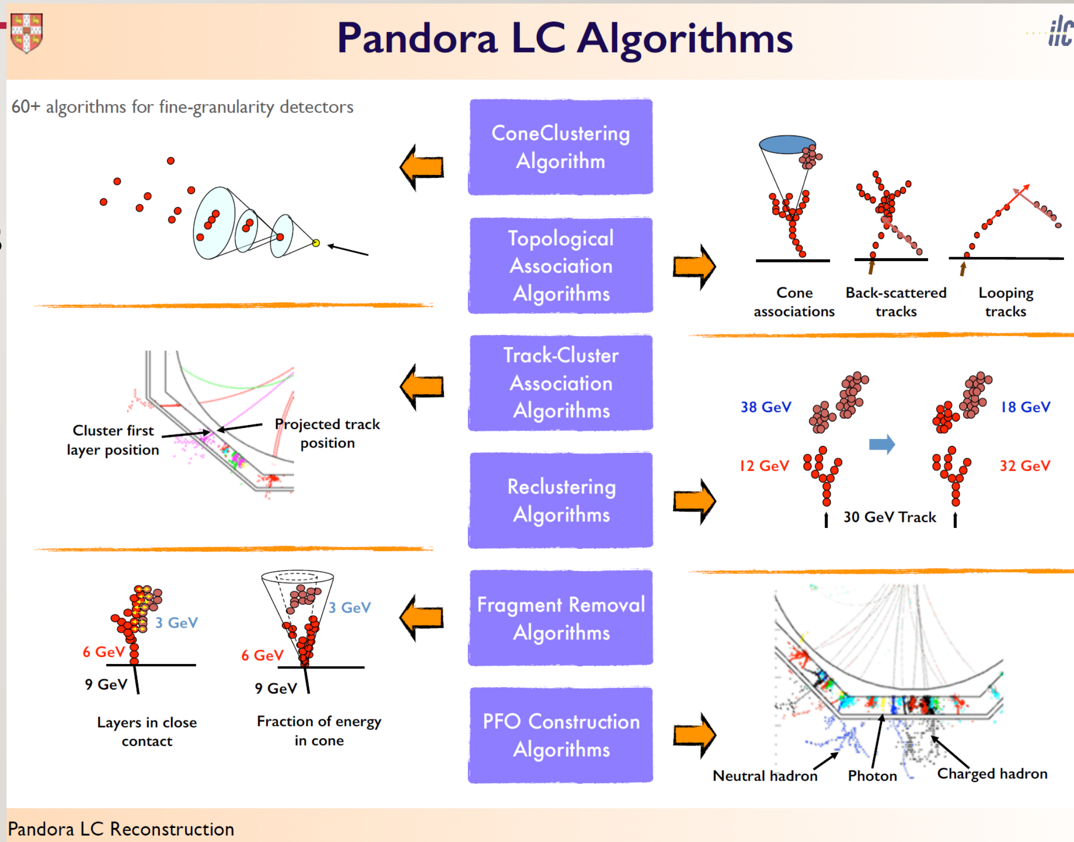


- Electromagnetic calorimeter (ECAL): Detects positions , and energy of gamma rays
  → Higher accuracy of particle identification: PFA
- SiW ECAL equips a lot of channels (~10^8) to identify each particle.
- Sandwich structure with 30 alternating layers of Si detection layer and W absorption layer.
- 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

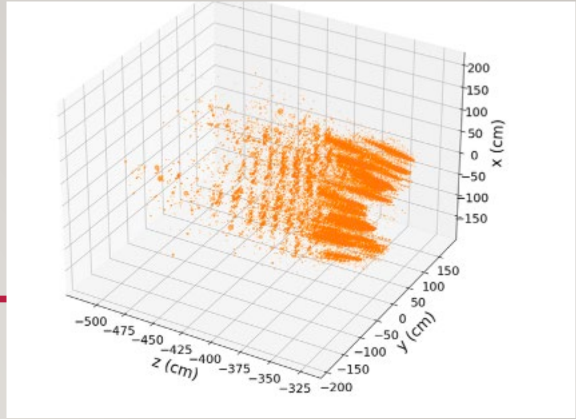
4 Application of Deep Learning to PFA

- Current PFA algorithm : PandoraPFA
  →The pattern recognition based on the human-tuned parameters
- Our targets:
  - Improve performance by reducing confusion term
  - Adding timing information
  - Checking detector effects on
    - Granularity (inc. MAPS?)
    - Timing resolution





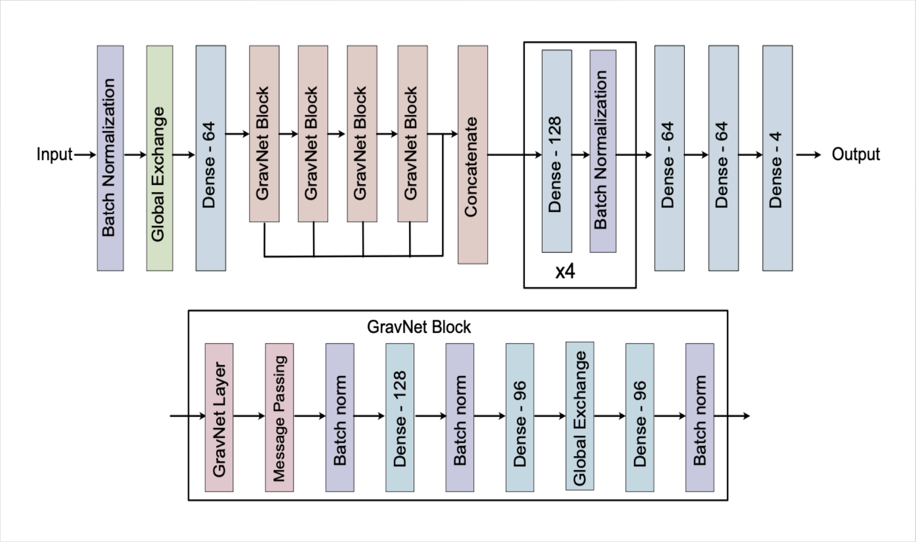
Simple sim with HGCal-like geometry

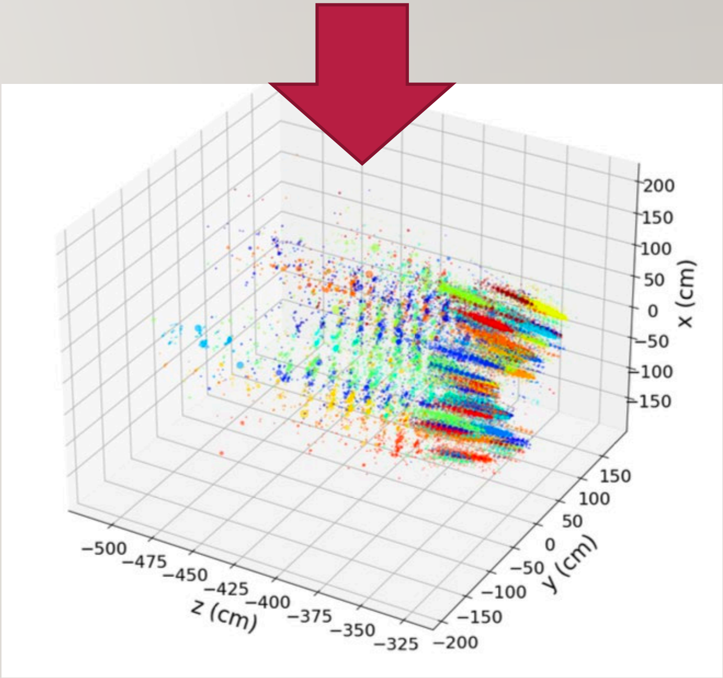


5 Calorimeter Clustering

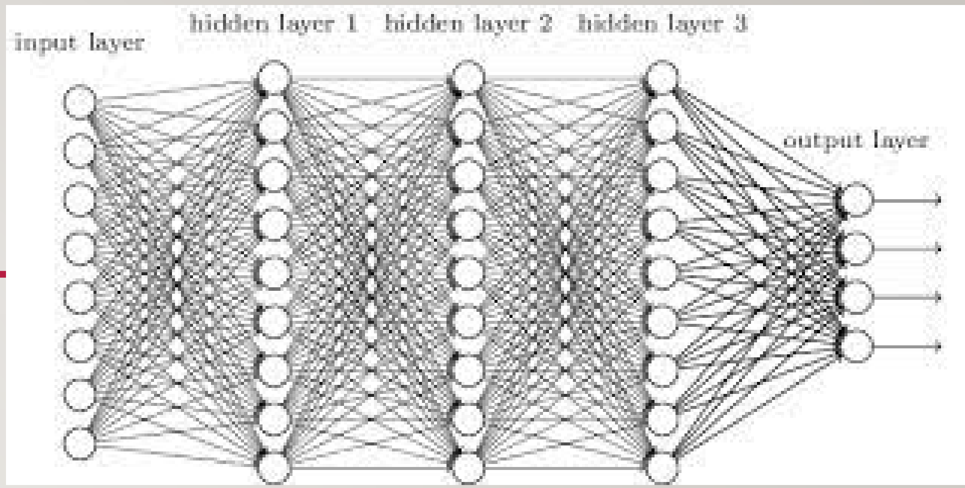
- Input: features of hit in the calorimeter e.g., position, energy, etc.
  → discriminate each cluster
- Deep Learning Architecture
  - Based on Graph Neural Network developed for CMS HGCal

(collaboration with L. Gray et al. (Fermilab))



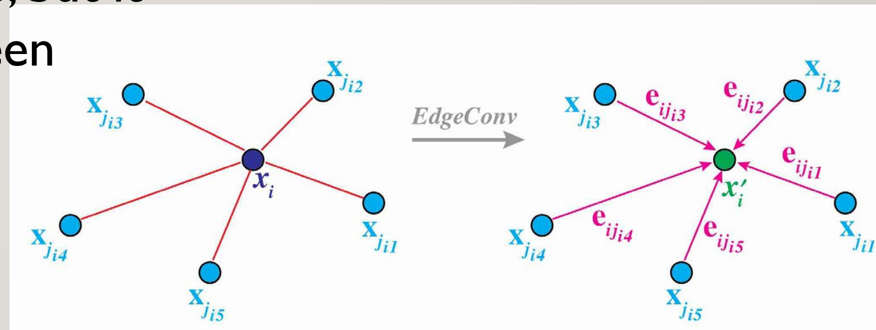


6 Deep Learning



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.

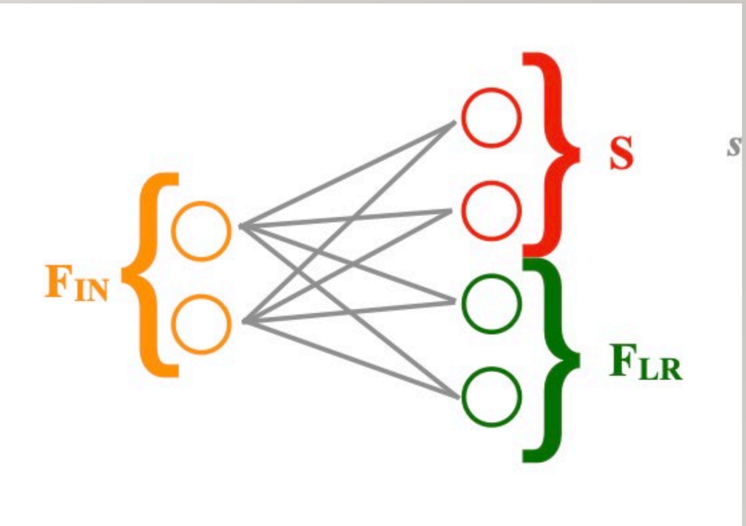


7 GravNet

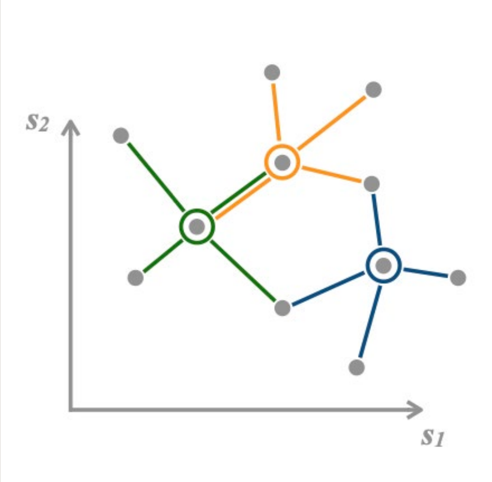
- Input Data : $V \times F_{IN}$

V : Number of hits for each detector

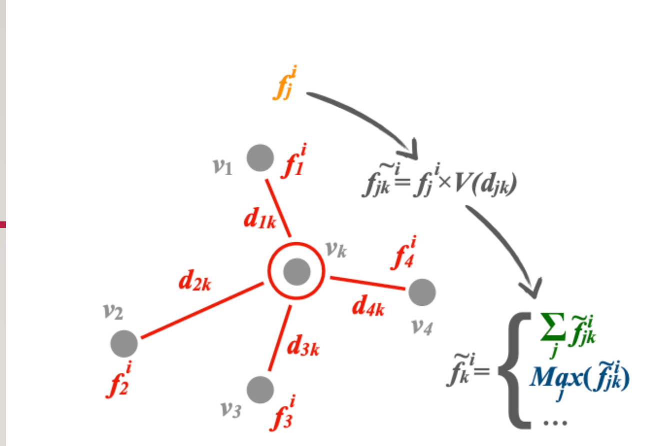
$\mathrm{F_{IN}}$: Number of the features for each hit

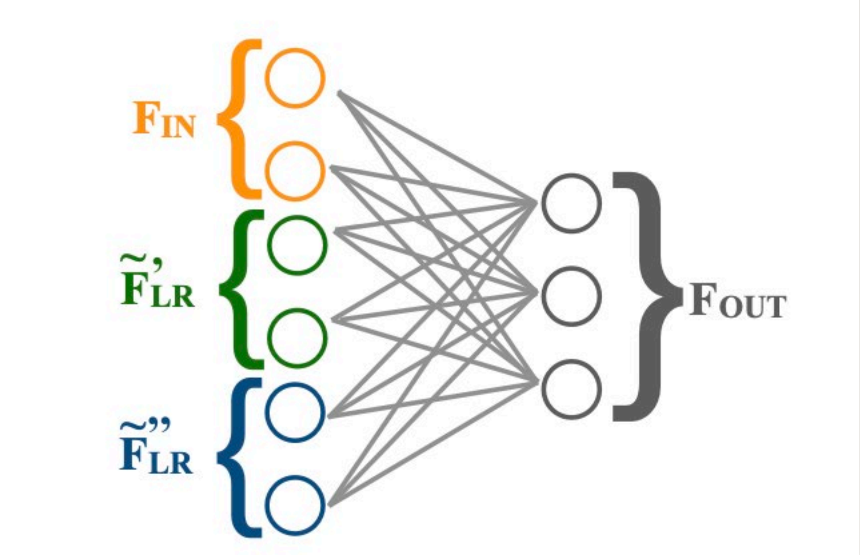


- S : Set of coordinates in some learned representation space
- F_{LR} : learned representation of the vertex features
- Input data of initial dimension $V \times F_{IN}$ is converted into a graph.
- The coordinates of the graph is updated by the learning of the network.



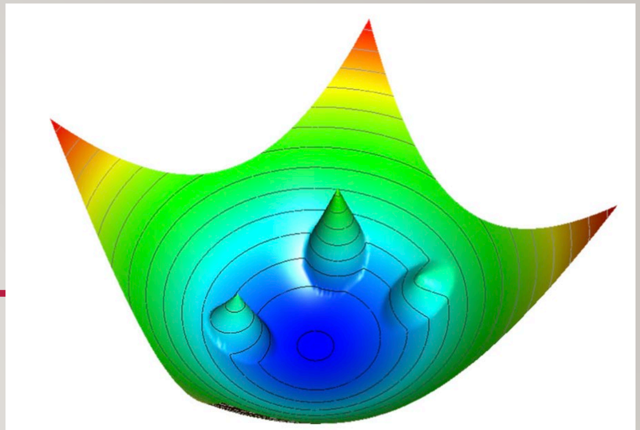
8 GravNet





- 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

$$L = L_V + L_\beta$$

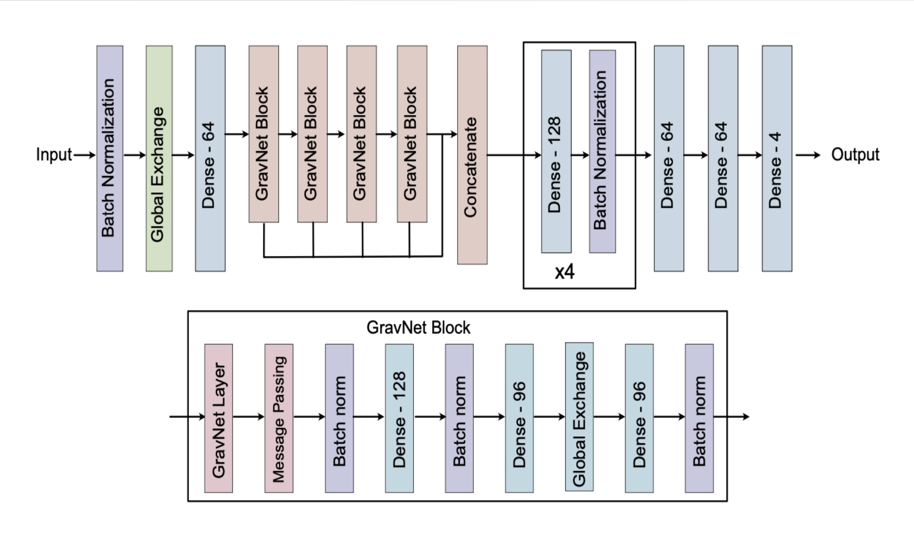
- L_V :The closer the hit is to a particle with high β and belonging to the same particle, the smaller it is, and the more it belongs to a different particle, the larger it is.
  → Equivalent to the attractive and repulsive forces acting on an electric charge
- L_β : Converge β to 1 for only one of each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

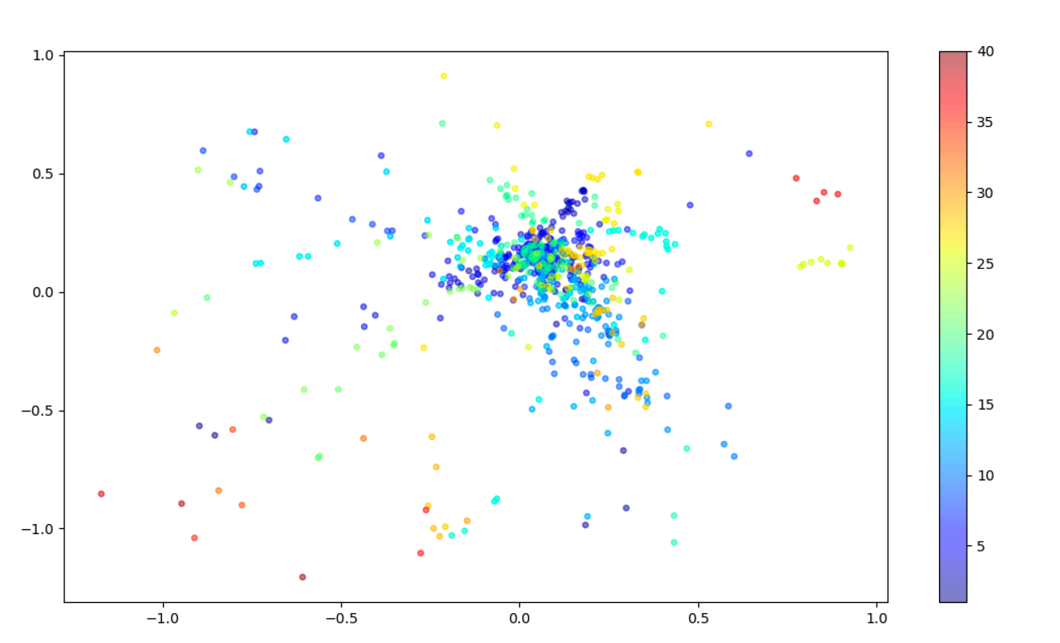
Output of network
- Beta (condensation)
- 2 x coordinate

per hit
Used for clustering

10 Clustering

- Get "condensation point" with hits with beta > threshold
- Cluster other hits to nearest condensation point in the virtual coordinate (of network output)

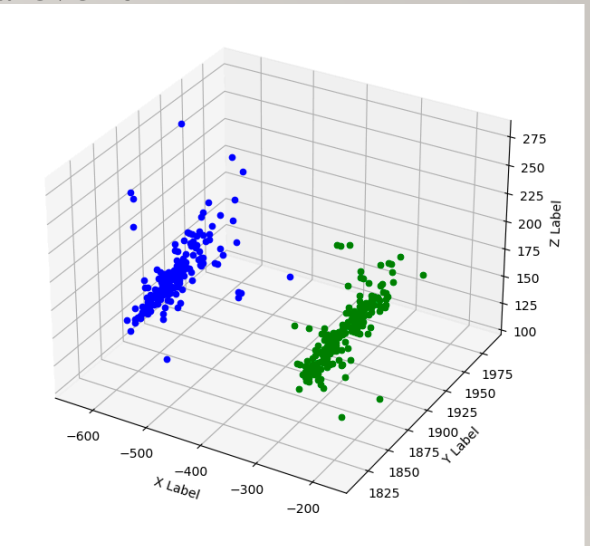




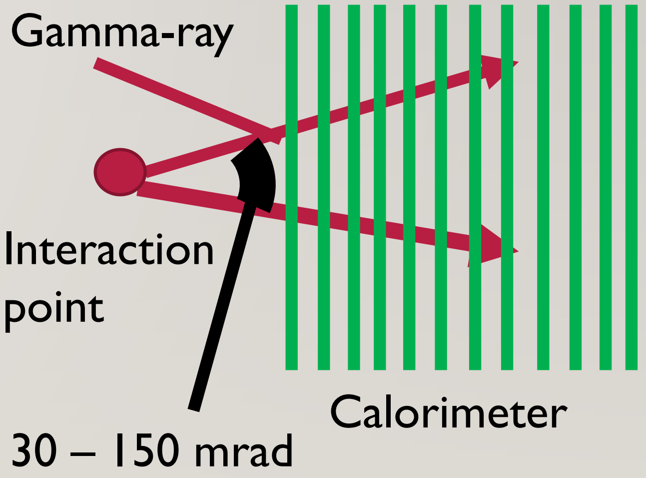
Generation of Input Data

- Two gamma events are generated by ILD detector simulation
- 10000 Events are generated for each of the five data sets from 30 to 150 mrad
- θ: 85/180 π , ϕ: random, momentum: 5.0 GeV

Two gamma event

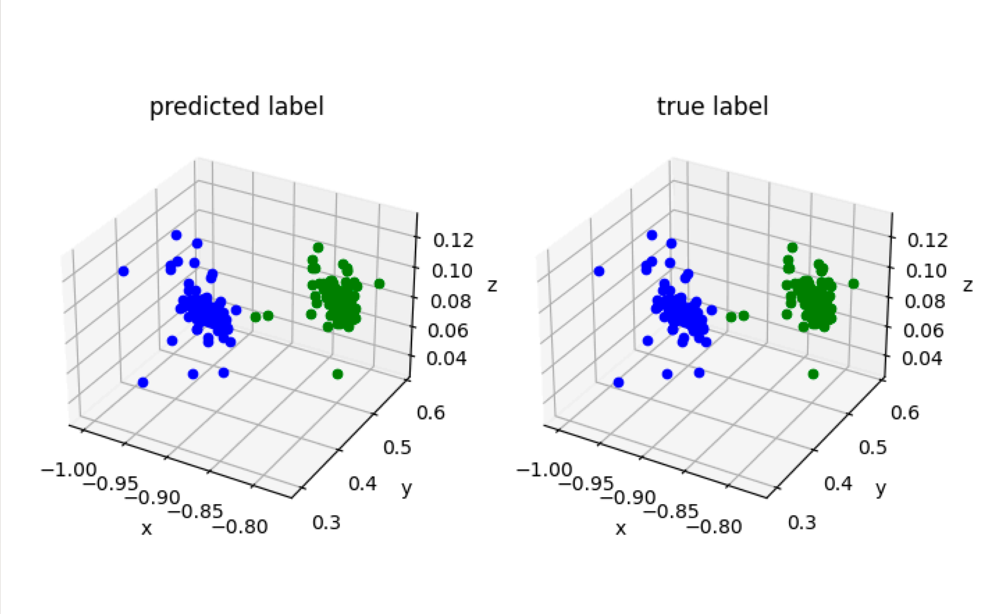


Generation of MC particles → Simulation based on detector geometry by ddsim → Reconstruction of hits in the detector by Marlin

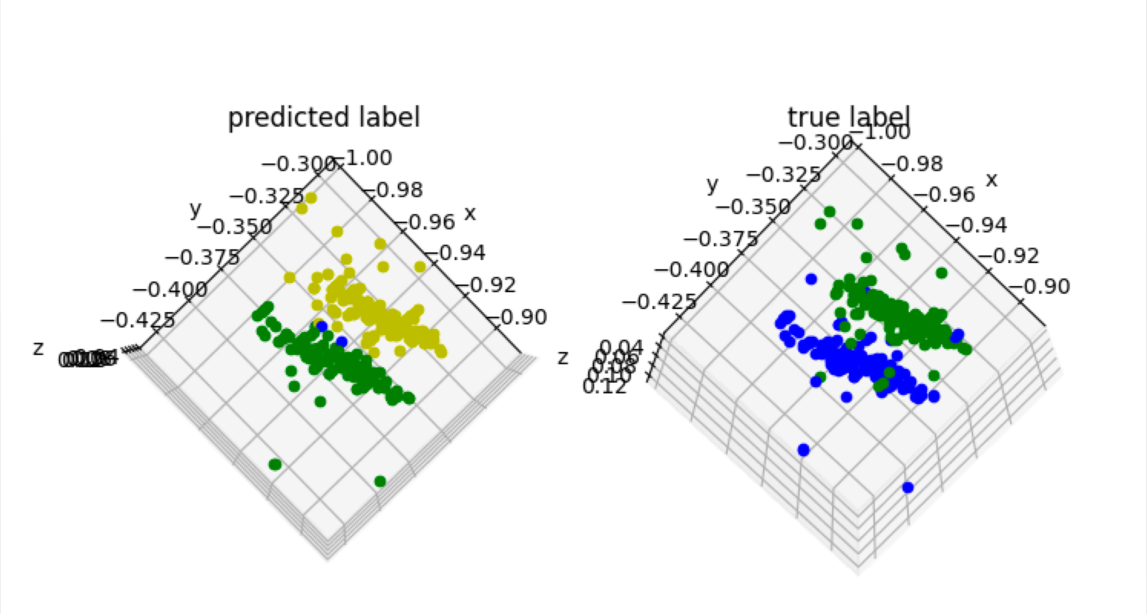


Event Display

Large angle (150 mrad): perfect reconstruction



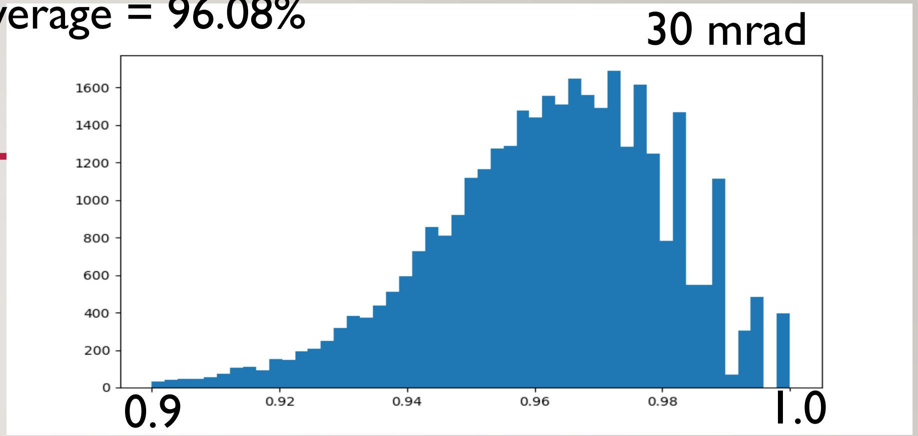
Small angle (30 mrad): a few hits misclustered



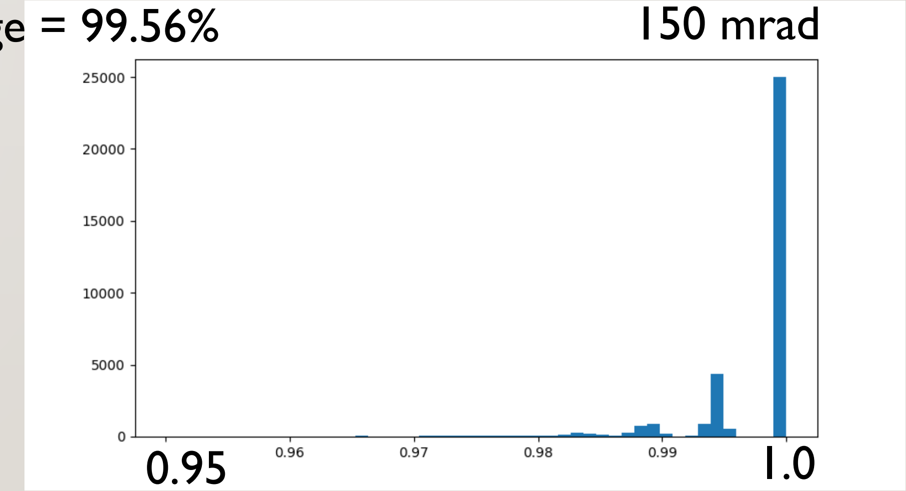
Evaluation of Network

- Accuracy : $\dfrac{\text{Number of hits which is predicted correctly}}{\text{Number of hits with true label of each cluster}}$
- The simulation data includes events where photons are converted into other particles.
- As input data, events with only two clusters are selected

Average = 96.08%



Average = 99.56%



Angle[mrad]	**30**	**60**	**90**	**120**	**150**
Accuracy[%]	96.08	98.64	99.30	99.68	99.56

Plans for PFA in ~this year

- Prepare more complicated data (taus, jets, …)
  - Restructuring data format (npz → awkward arrays)
  - Confirm (or tune) MC truth cluster definition
    - How to treat split clusters
- Track-cluster matching
  - Virtual hit representing a track
    - Position at the entry of calorimeter (with “track” flag)
    - To be forced condensation point – treated by loss function
    - How to integrate momentum (and direction)
      - Additional input to the hit characteristics or add at later stage
- Comparison with PandoraPFA – hoping to be better
  - If better, adapting it to analysis framework (to be used for physics analyses)
- Comparison with timing info included or not included
  - And with different timing resolution

Contents

- Calorimeter clustering with GravNet/Object condensation
  - Slides from S. Tsumura
- b/c tagging with Graph Attention Network
  - Slides from T. Onoe

Jet Flavor Tagging

- Important to identify quarks (b/c/g/uds) of the origin of the jets.
  e.g., Separation of $h \rightarrow b\bar{b}\ /c\bar{c}/q\bar{q}/\ldots$

- Ratio of background can be eliminated determines the limits of analysis cut
- Bottom (b) and charm (c) flavor hadrons have weak interaction
  → b/c hadrons have finite decay lengths
  → Can be identified by finding vertices

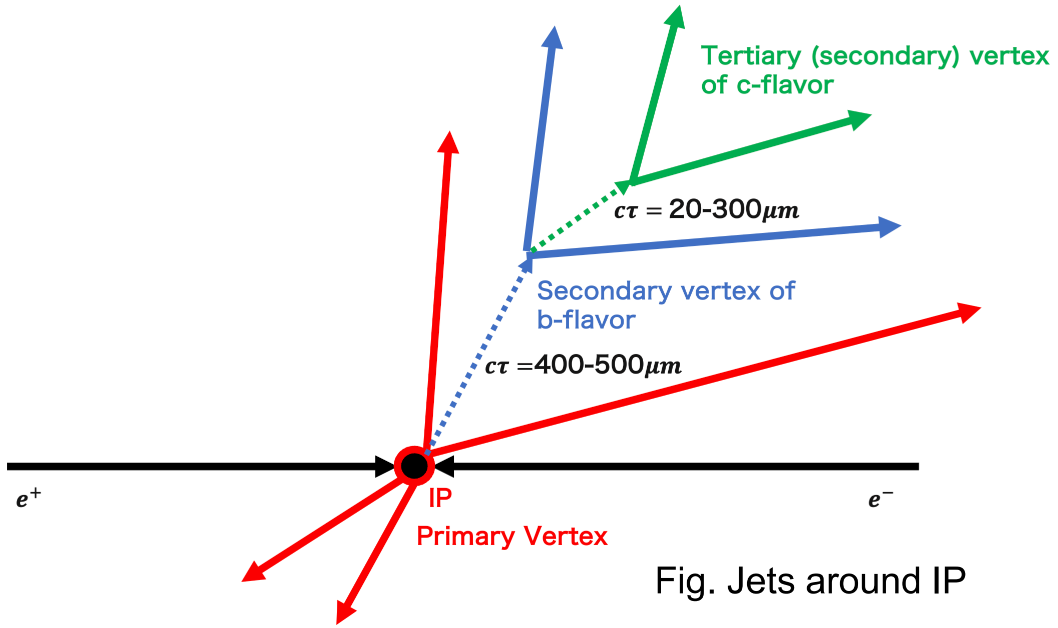


Fig. Jets around IP

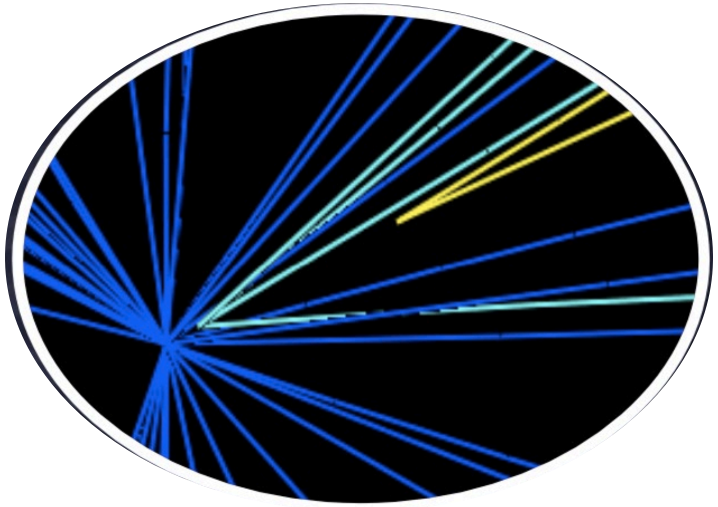


Fig. Monte Carlo simulation of the jet near the IP

Graph Data Approach

Concept

Data is represented as a graph

→ Graph structure data can contain interrelationship by connections

(Fully-connected neural network has no specific relation between nodes)

→ Reduced loss of information when compared to physical phenomena

→ High accuracy of identification is expected

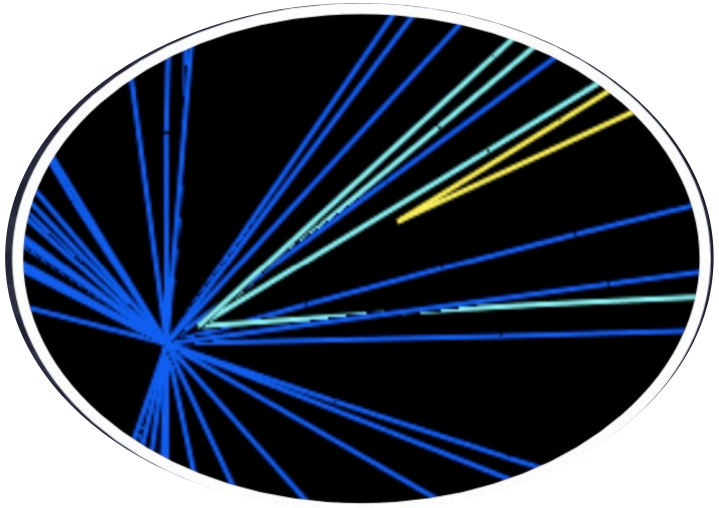


Fig. Event display of Monte-Carlo simulation

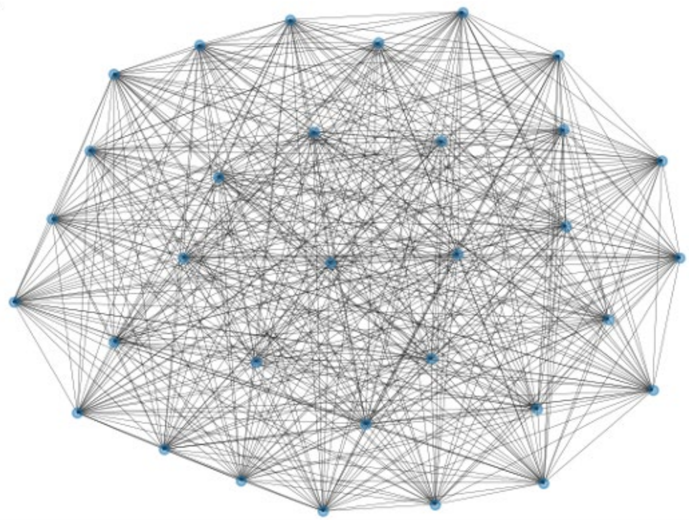


Fig. example of a jet as a graph

Training Data information

Data

- 240,000 jets of 250 GeV ILD full simulation data
  $[\, e^+e^- \rightarrow \nu\bar{\nu}h \rightarrow \nu\bar{\nu}b\bar{b}/c\bar{c}/q\bar{q} \;\; (q = u, d, s)]$

- Build one graph per one jet
- Define the tracks as nodes in the graph
- Edges connect between track pairs

Track Input	
$\mathbf{d_0}$	**Longitudinal distance from track to IP**
$\boldsymbol{\phi}$	**Azimuthal angle of track**
$\boldsymbol{\omega}$	**the curvature of the track**
$\mathbf{z_0}$	**Transverse distance from track to IP**
$\tan\lambda$	**dz/ds in sz plane**
$\sigma(\mathbf{d_0})$	**Uncertainty of** $\mathbf{d_0}$
$\sigma(\mathbf{z_0})$	**Uncertainty of** $\mathbf{z_0}$

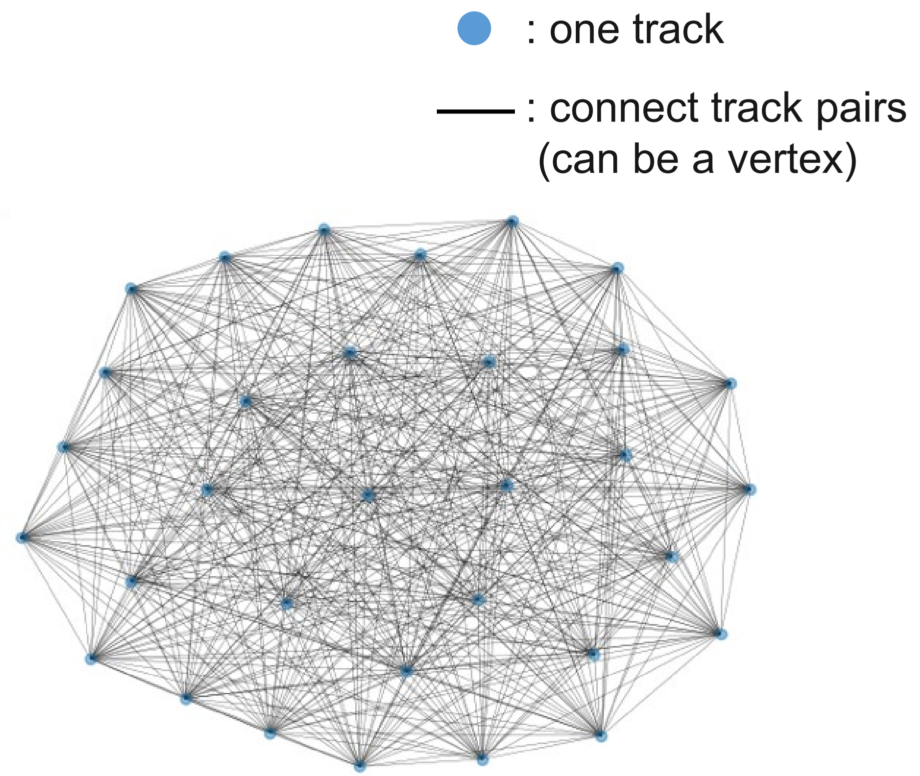


Fig. example of a jet as a graph

Graph Training and GAT

- How to train with graph data (Graph Neural Network; GNN)
  ... Aggregate features from neighboring nodes and update

- We suggest Graph Attention Network (GAT) , a GNN with attention mechanism
- Attention mechanism ... Learn the importance score for each weight
  Take as a coefficient for update parameter.

→ Aimed by attention expressing whether tracks has the same vertex.

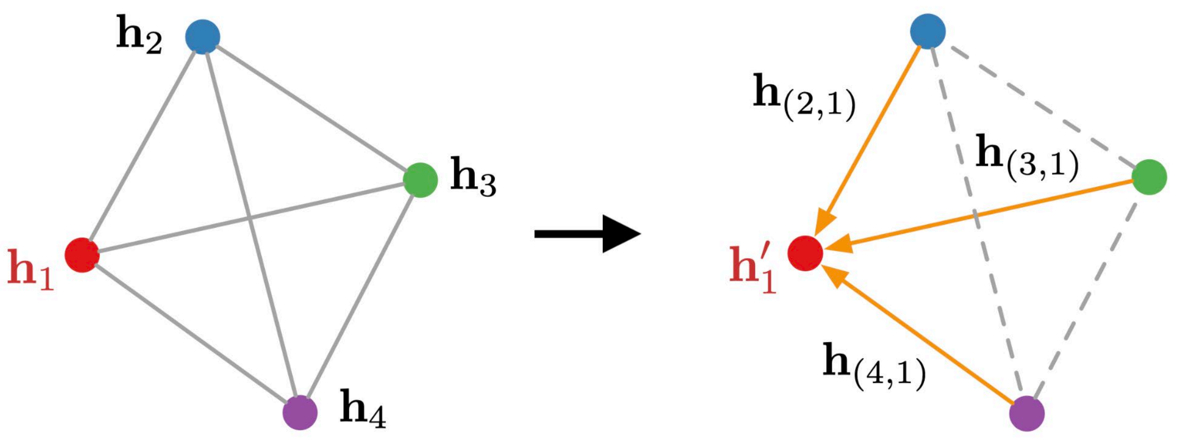


Fig. Graph Training

arXiv [1710.10903]

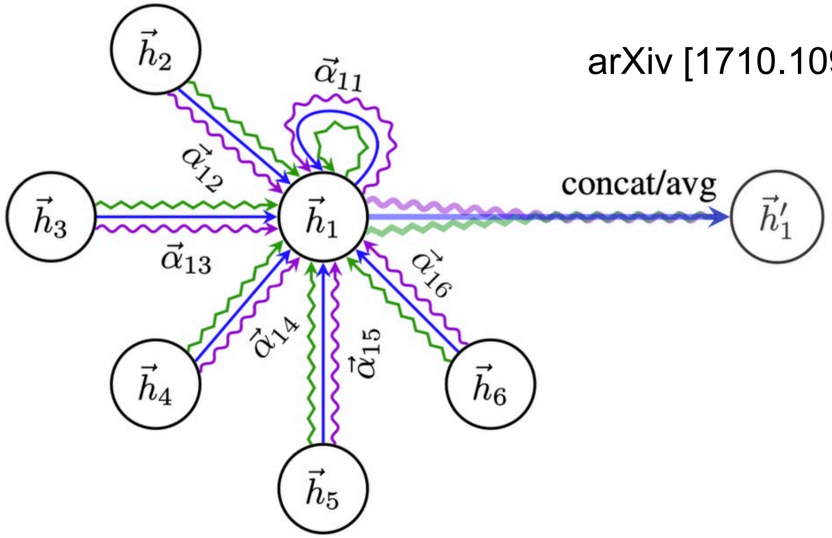


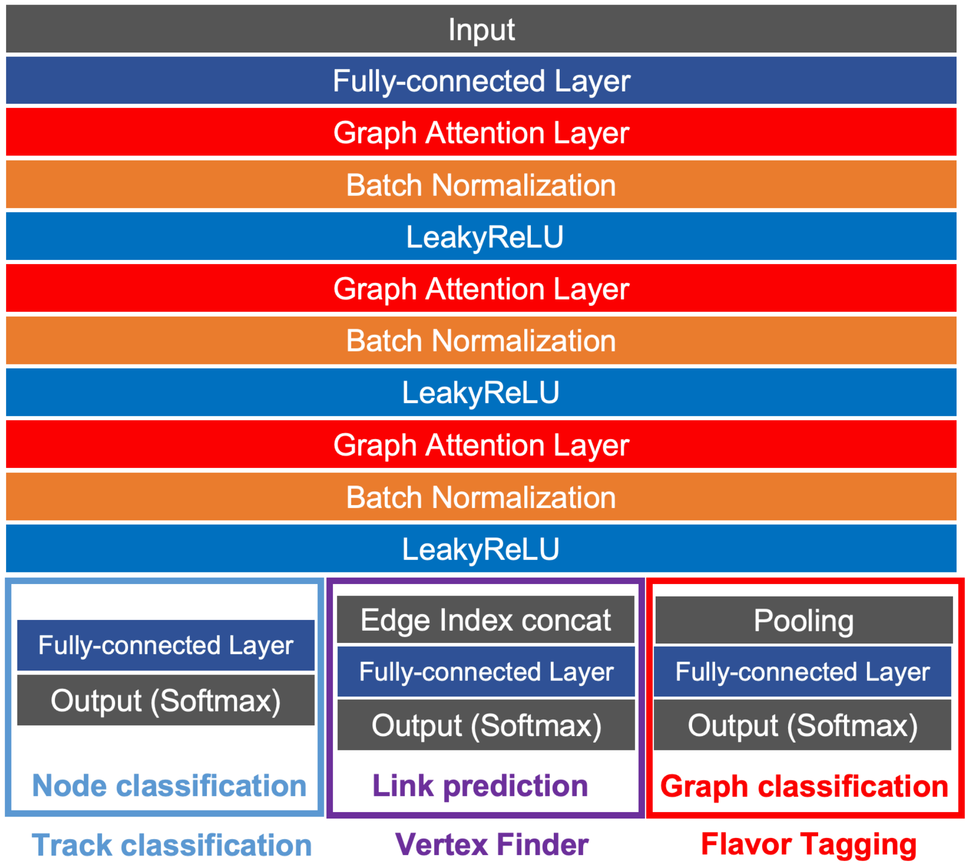
Fig. Graph Attention Network

Training and Network architecture

- Node classification means the origin of tracks as vertices
- Link prediction means whether to form a vertex
- Graph classification means jet flavor tagging
- Loss function

$$L_{total} = L_{Flavor} + \alpha L_{Vertex} + \beta L_{Edge}$$
$$(\alpha \cong 3, \beta \cong 1)$$

Network architecture



Node classification

Label	Description
PV	From primary vertex
SVBB	From secondary vertex of b
SVCC	From secondary vertex of c
TVCC	From tertiary vertex of b
Others	From another particle

Link prediction

Label	Description
Connected	tracks are connected
Not-connected	tracks are not connected

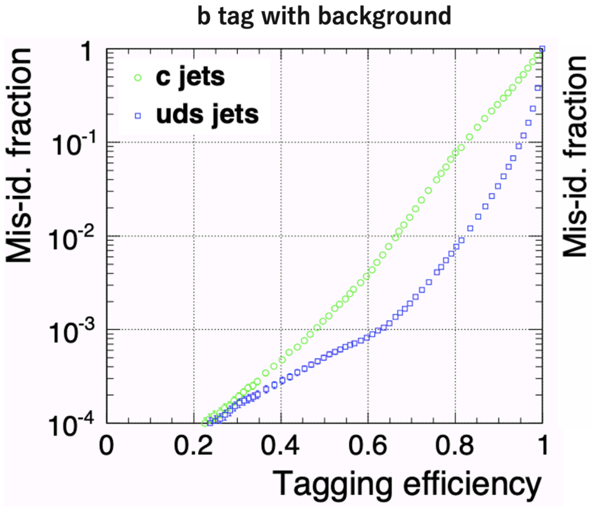
Graph Classification

Label	Description
$b\bar{b}$	the final state of $b\bar{b}$
$c\bar{c}$	the final state of $c\bar{c}$
$q\bar{q}$	the final state of $q\bar{q}$ $(q = u, d, s)$

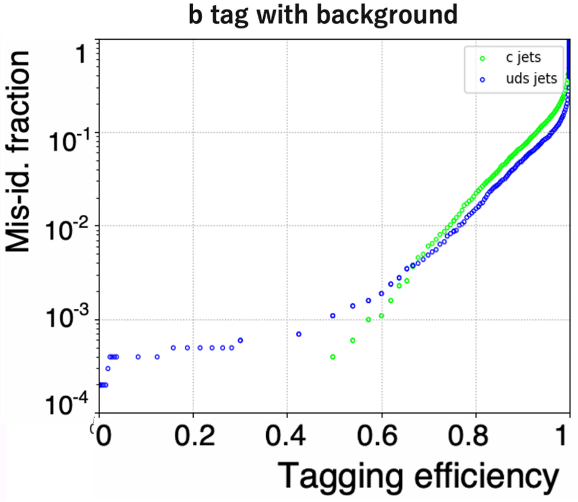
Evaluation of GNN

B tag effciency with background

LCFIPlus

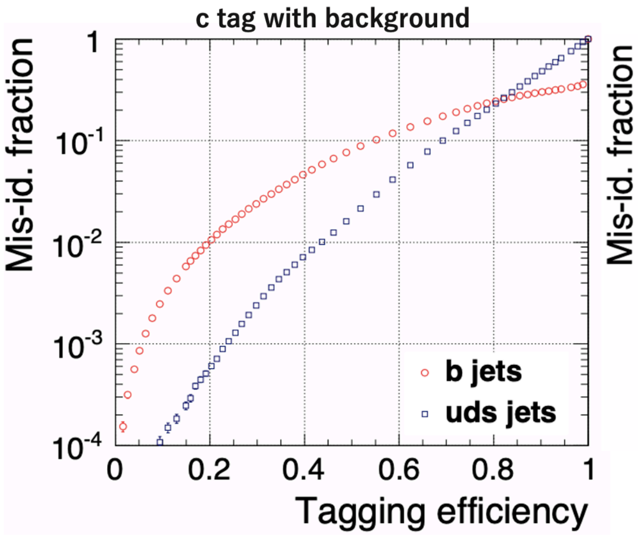


Graph Approach

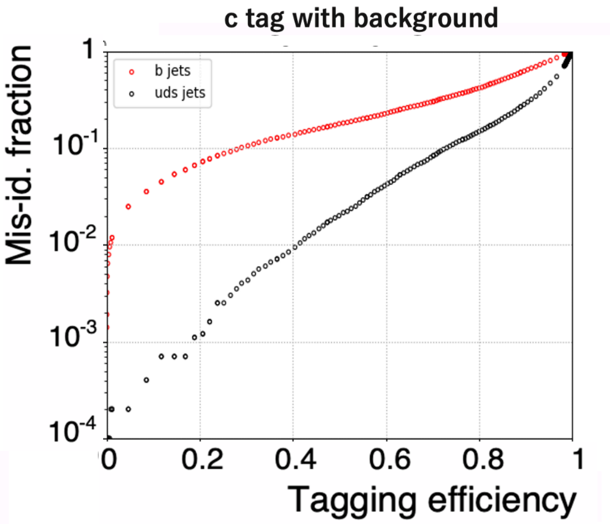


C tag efficiency with background

LCFIPlus



Graph Approach



Tagging efficiency = 0.8	background	Mis-id fraction	
		LCFIPlus	GNN
b jet	c jet	0.073	0.021
	uds jet	0.007	0.015
c jet	b jet	0.22	0.40
	uds jet	0.24	0.14

- For b jet, the ratio of c jet background is reduced.
- For c jet, the ratio of uds jet background is reduced.

- Integrated of Flavor Tagging with Vertex Finder
  → Implementation with low-level of input than LCFIPlus

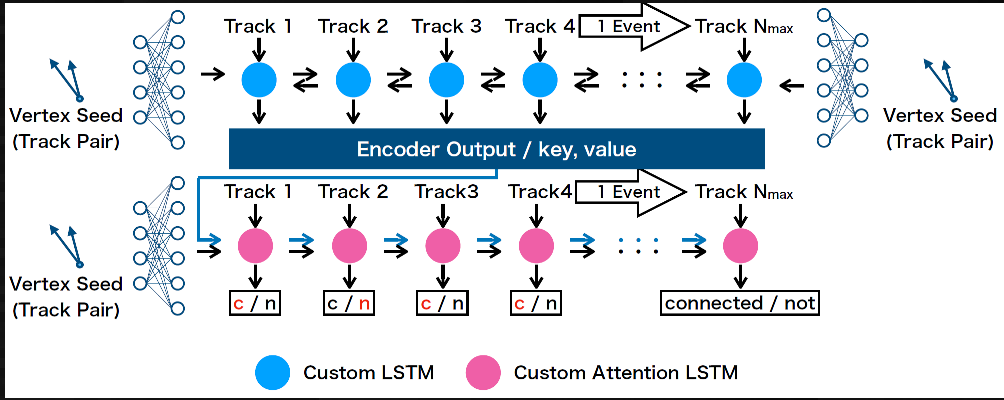
Status and plans in flavor tagging

- Tuning of the GAT-based flavor tagging
  - Investigate reasons for degraded performance on node/edge classification
  - Connecting output (or nearly-output) of node/edge to flavor tagging
- Another methods to be considered
  - Importing LHC method (ParticleNet, LorentzNet etc.)
  - Transformer-like method (graph transformer, set transformer etc.)
- Compare among algorithms as well as LCFIPlus
  - Import it to the analysis framework if better than LCFIPlus
- Considering timing information to be included
  - Dependence of timing resolution also to be seen

Backup slides

Appendix: quark flavor tagging with DNN

- Modified LSTM with attention NIMA 1047 (2023) 167836



Performance of vertex finding in this network

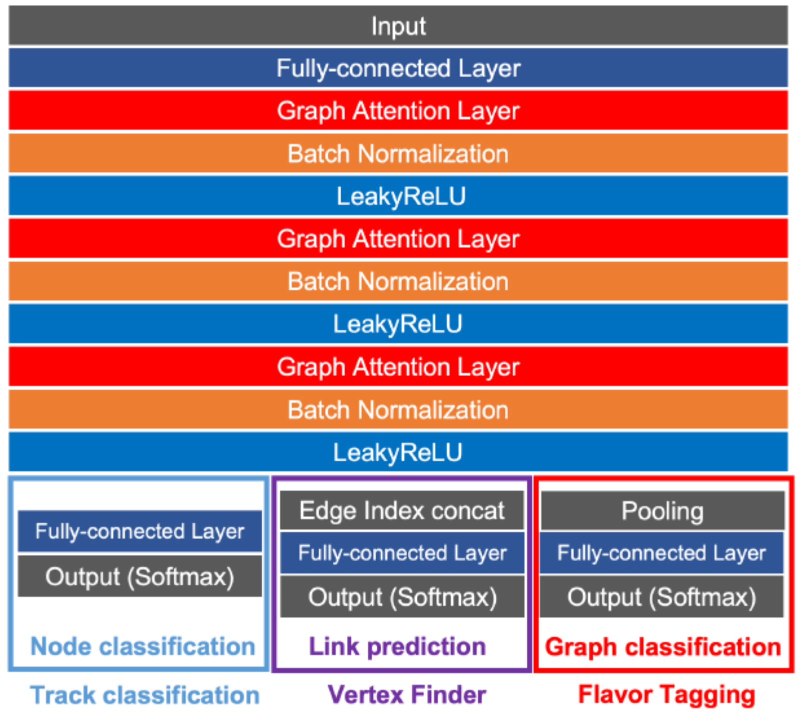
Criteria / True label	Primary	Bottom	Charm	Others
All tracks	307 657	187 283	180 143	42 888
In secondary vertex	2.2%	63.3%	68.4%	9.5%
– of same decay chain		62.3%	67.2%	
– of same parent		38.1%	36.2%	6.4%

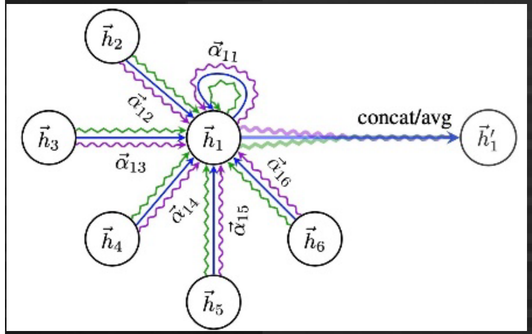
Performance for vertex finding in LCFIPlus

Criteria / True label	Primary	Bottom	Charm	Others
All tracks	307 657	187 283	180 143	42 888
In secondary vertex	0.2%	57.9%	60.3%	0.5%
– of same decay chain		57.5%	59.9%	
– of same parent		34.0%	37.2%	0.3%

Flavor tagging with GNN (ongoing effort)

- Simultaneous classifications of nodes, edges and graphs





Graph Attention Network (GAT)

Tagging efficiency = 0.8	background	Mis-id fraction	
		LCFIPlus	GNN
b jet	*c* jet	0.073	0.021
	uds jet	0.007	0.015
c jet	*b* jet	0.22	0.40
	uds jet	0.24	0.14

Partially better than LCFIPlus

25 GRAVNET - NETWORK -

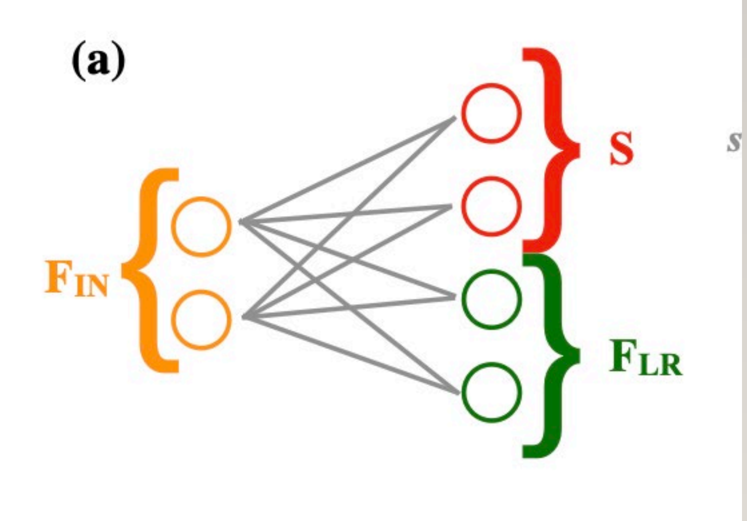
- Input Data : $B \times V \times F_{IN}$

  B ： Number of examples including in a batch

  V ： Number of hits for each detector

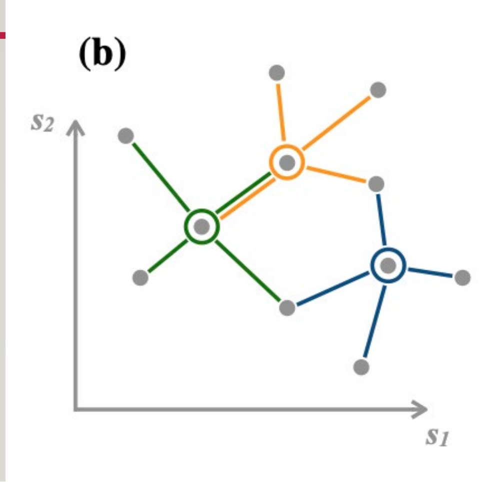
  $\mathrm{F_{IN}}$ ： Number of the features for each hit

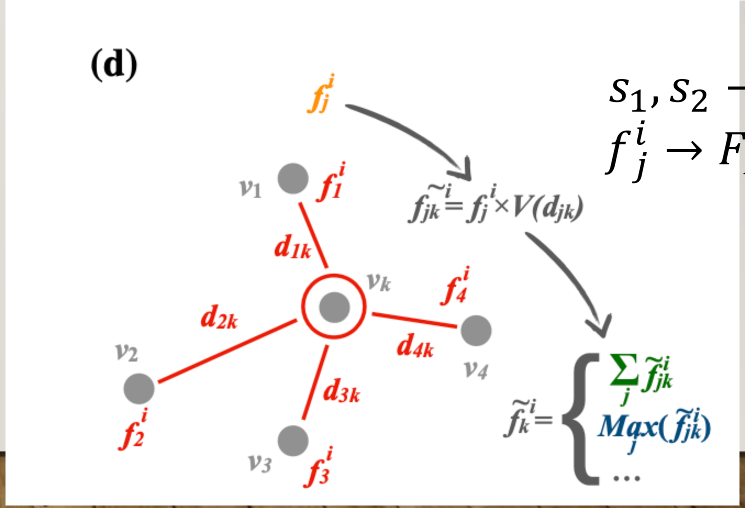
- S : Set of coordinates in some learned representation space
- F_{LR} : learned representation of the vertex features



26 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 $V(d_{jk})$ is introduced to enhance the contribution of close-by vertices.
  Example: $V(d_{jk}) = \exp(-d_{jk}^2)$
- 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

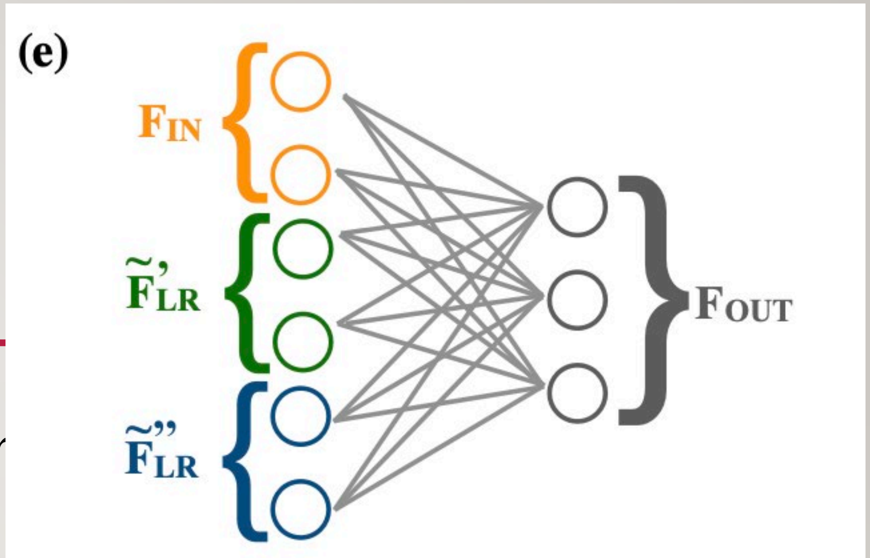




$s_1, s_2 \rightarrow S$

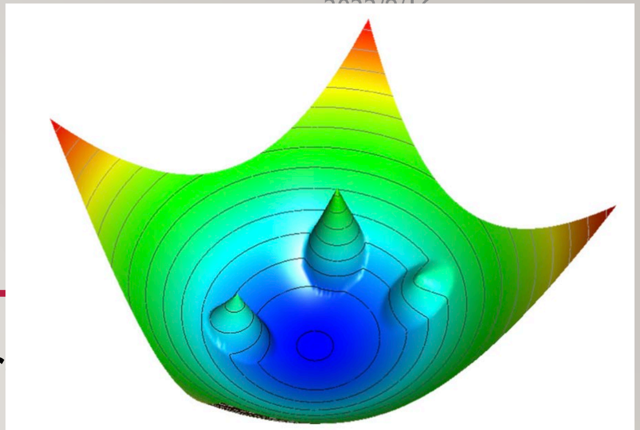
$f_j^i \rightarrow F_{LR}$

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.

28 Object Condensation



- Get the output from GravNet as β and output whether the hit seems to be a r
  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

29 LOSS FUNCTION - NETWORK LEARNING -

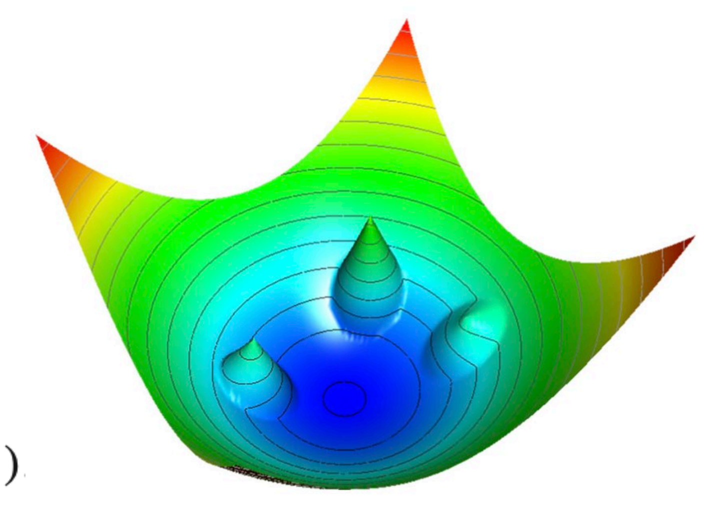
- The value of β_i $(0 < \beta_i < 1)$ is used to define a charge q_i per vertex i

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

- 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

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

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



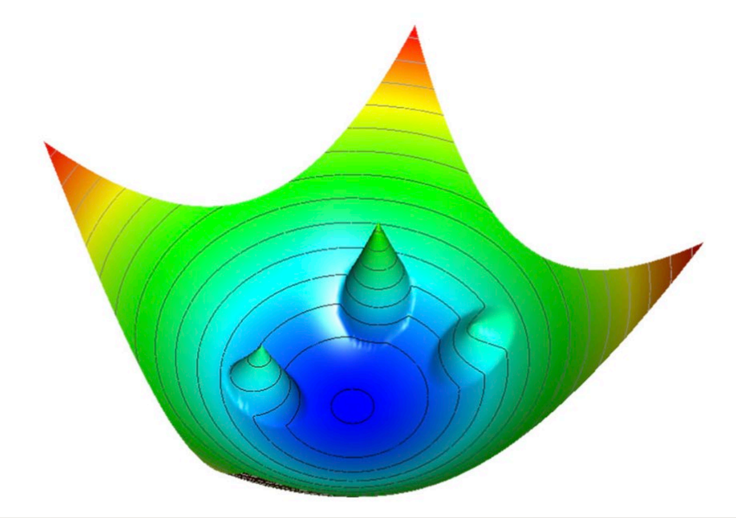
30 LOSS FUNCTION

- 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 :

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

31 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 $\operatorname{arctanh}^2 \beta_i$:

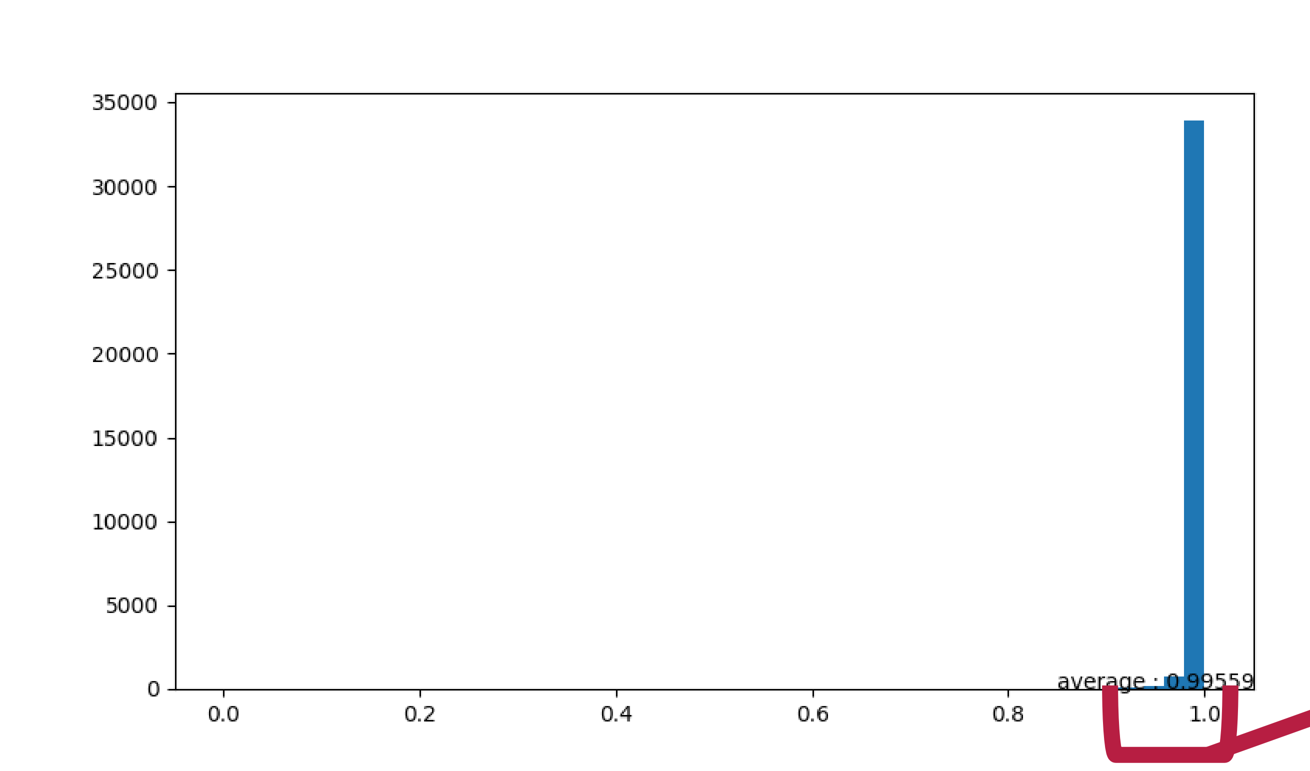
$$L_p = \frac{1}{\sum_{i=1}^N \xi_i} \cdot \sum_{i=1}^N L_i(t_i, p_i)\, \xi_i, \text{ with}$$

$$\xi_i = (1 - n_i) \operatorname{arctanh}^2 \beta_i .$$

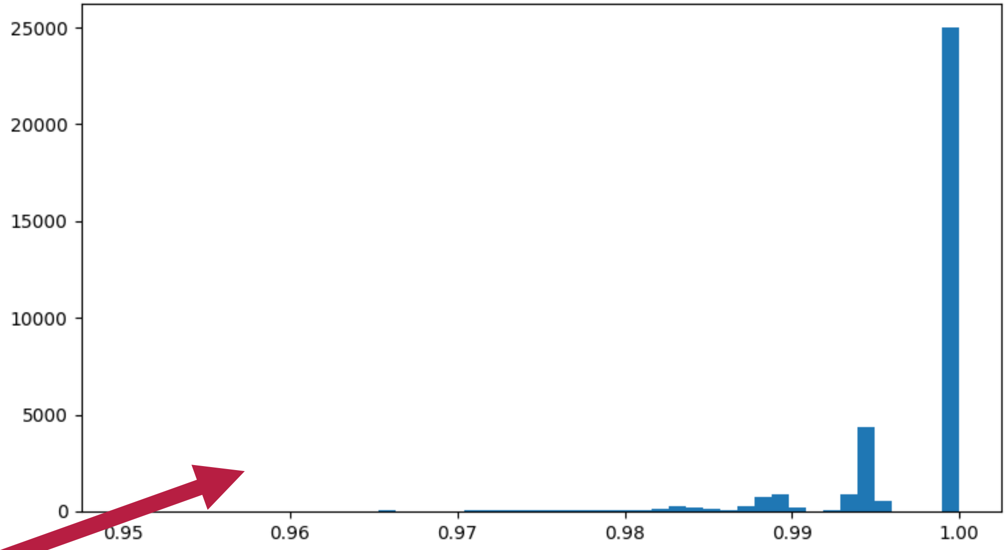
p_i: Featutes
$L_i(t_i, p_i)$: Loss term (Difference between true labels and outputs of network)

EVALUATION

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



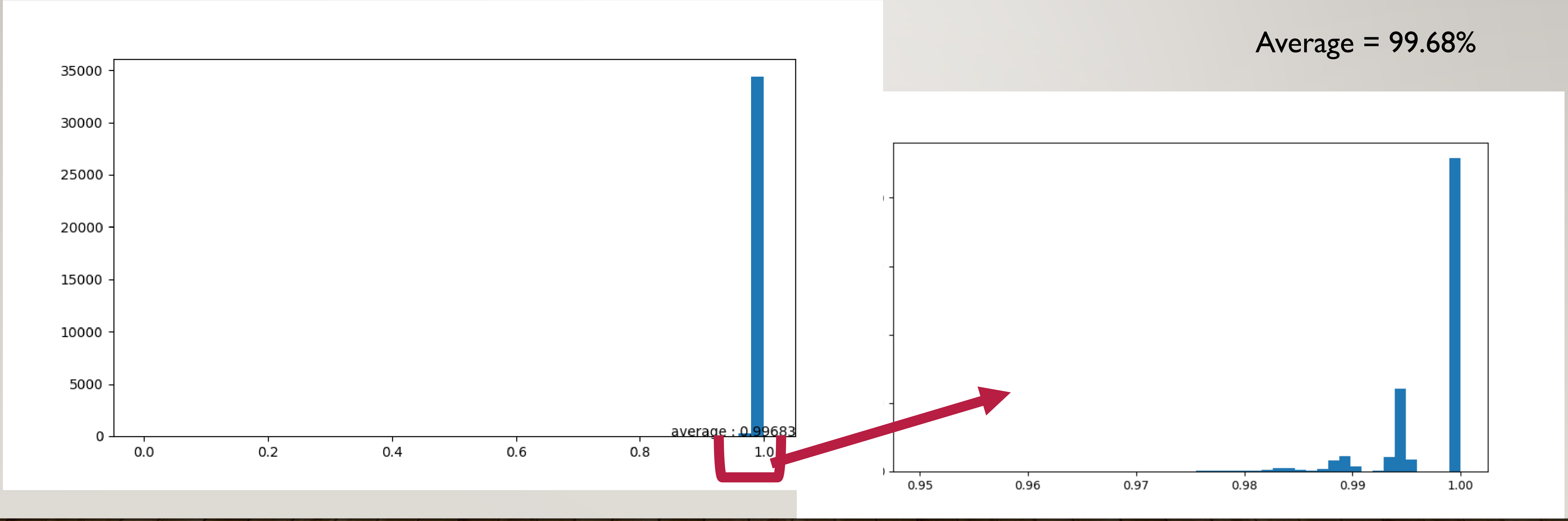
Average = 99.56%



EVALUATION

Opening angle = 0.4 rad

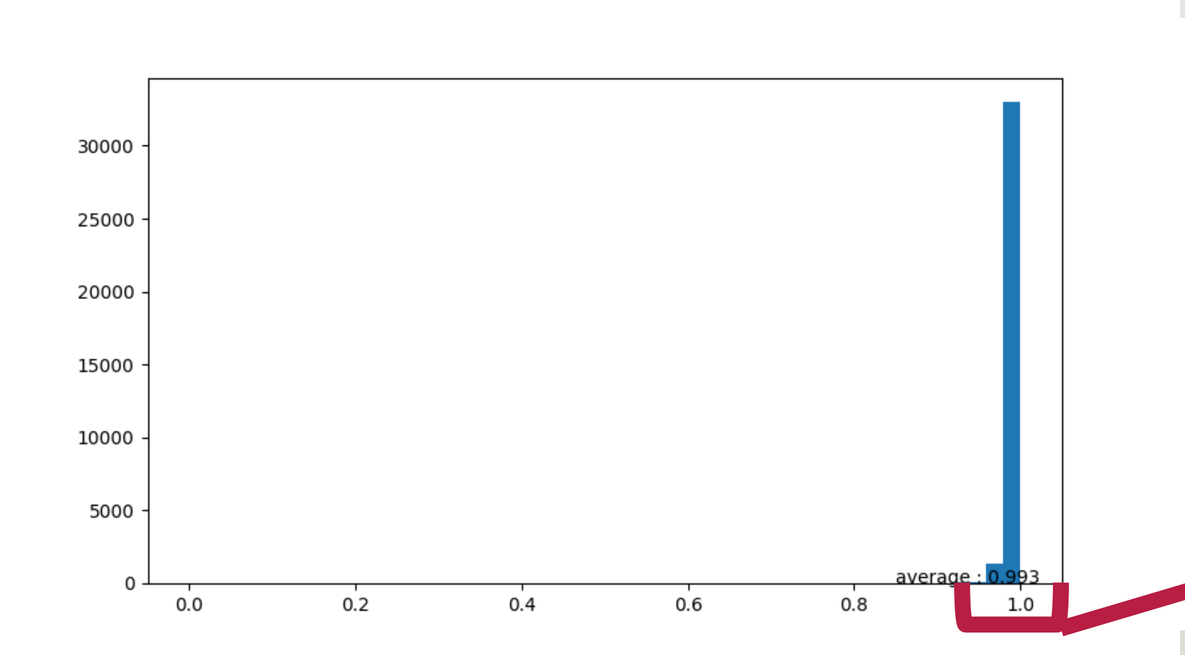
Average = 99.68%

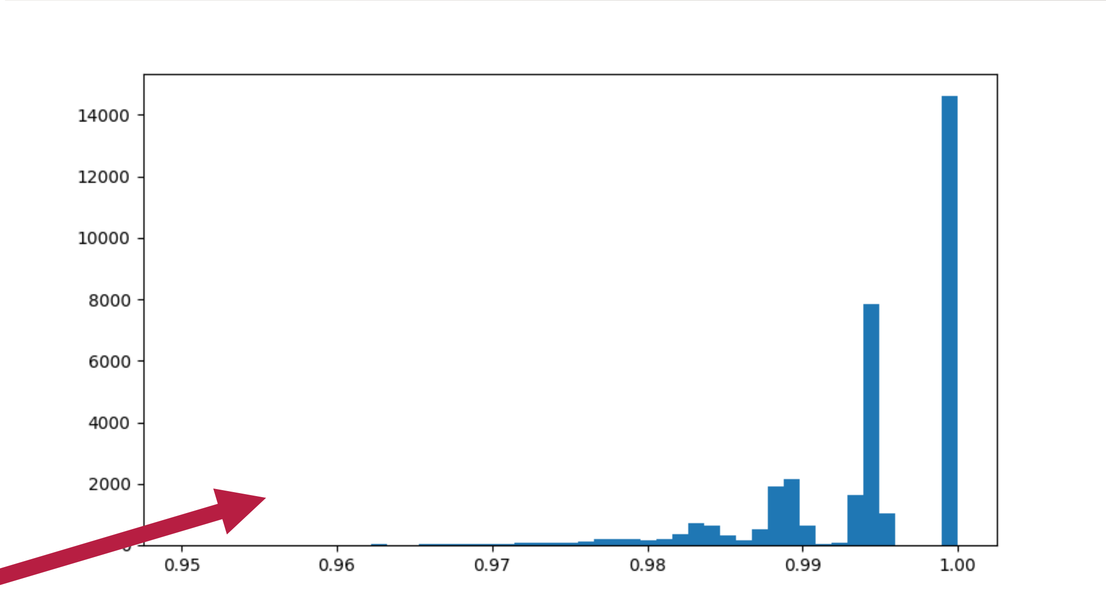


EVALUATION

Opening angle = 0.3 rad

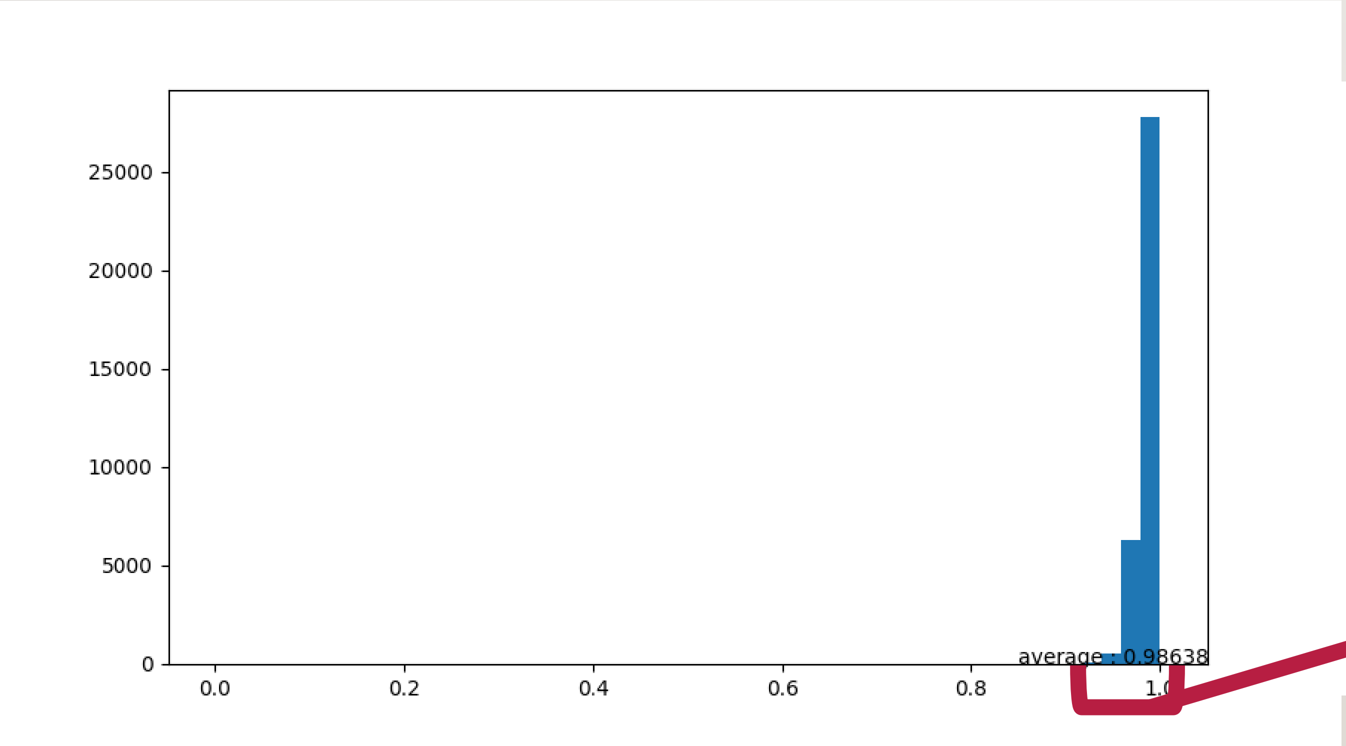
Average = 99.30%



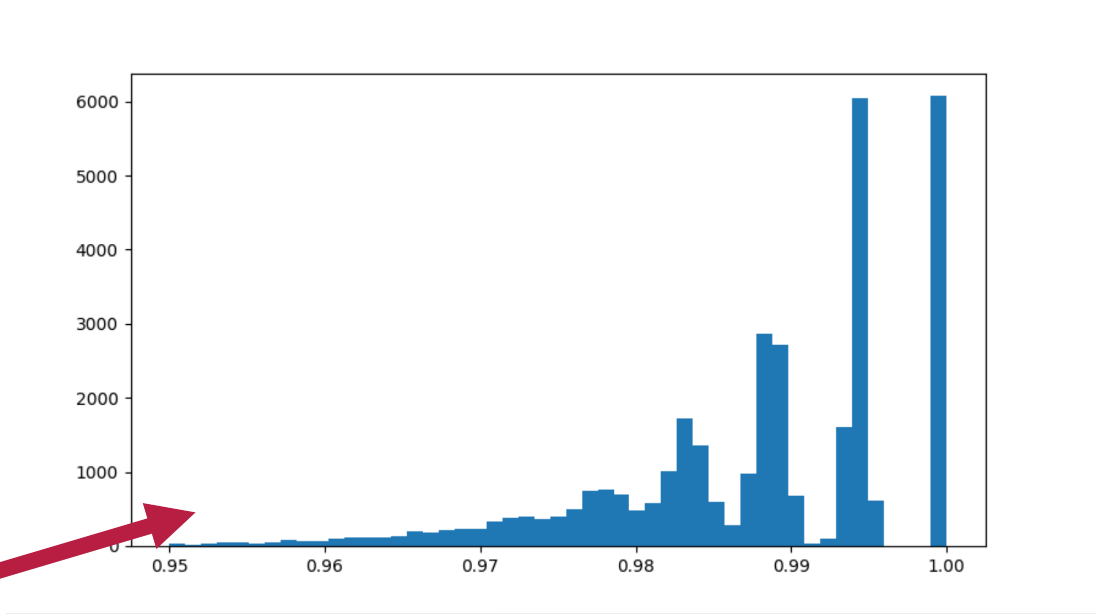


EVALUATION

Opening angle = 0.2 rad



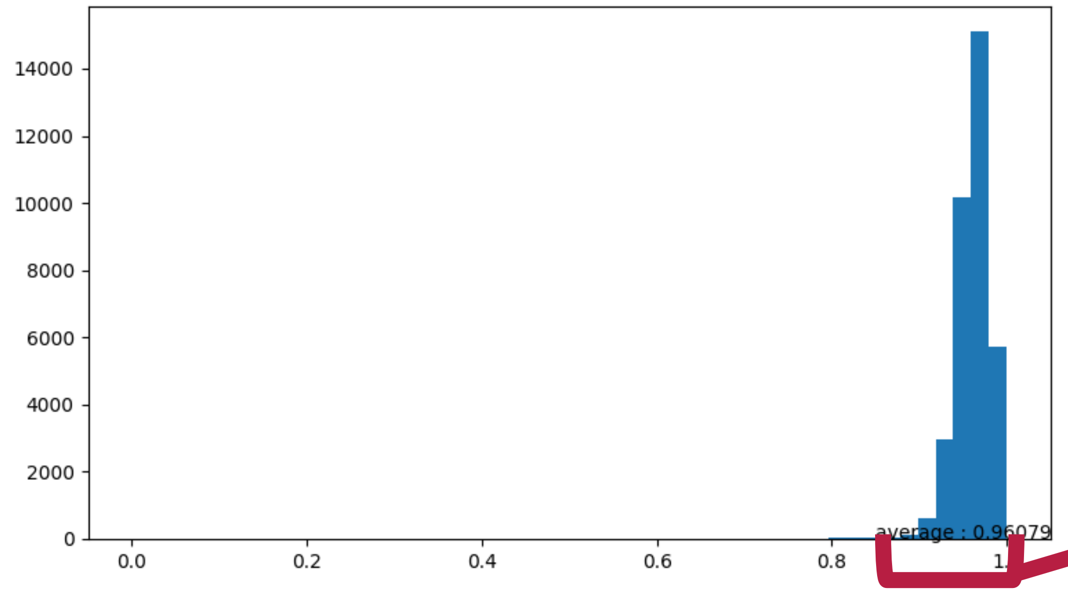
Average = 98.64%

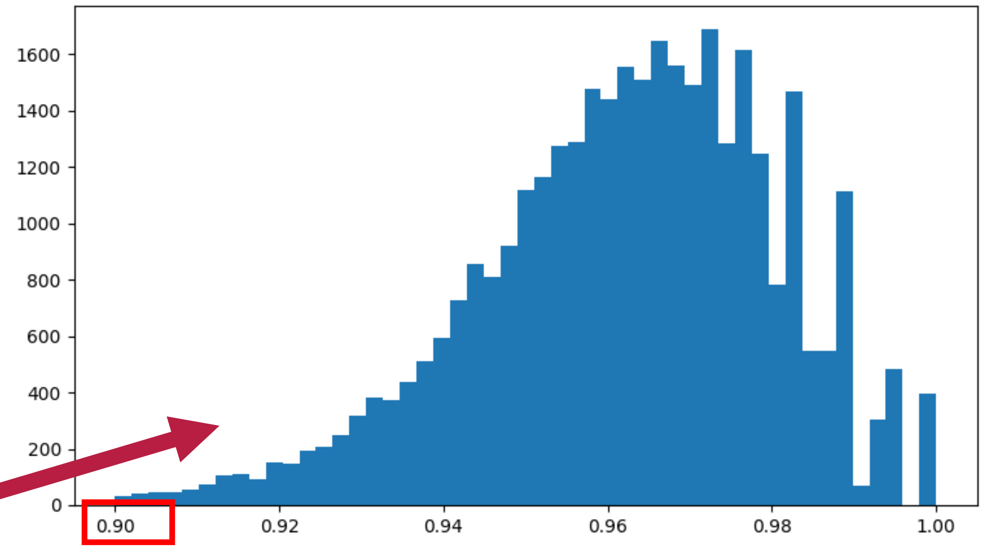


EVALUATION

Opening angle = 0.1 rad (the smallest one)

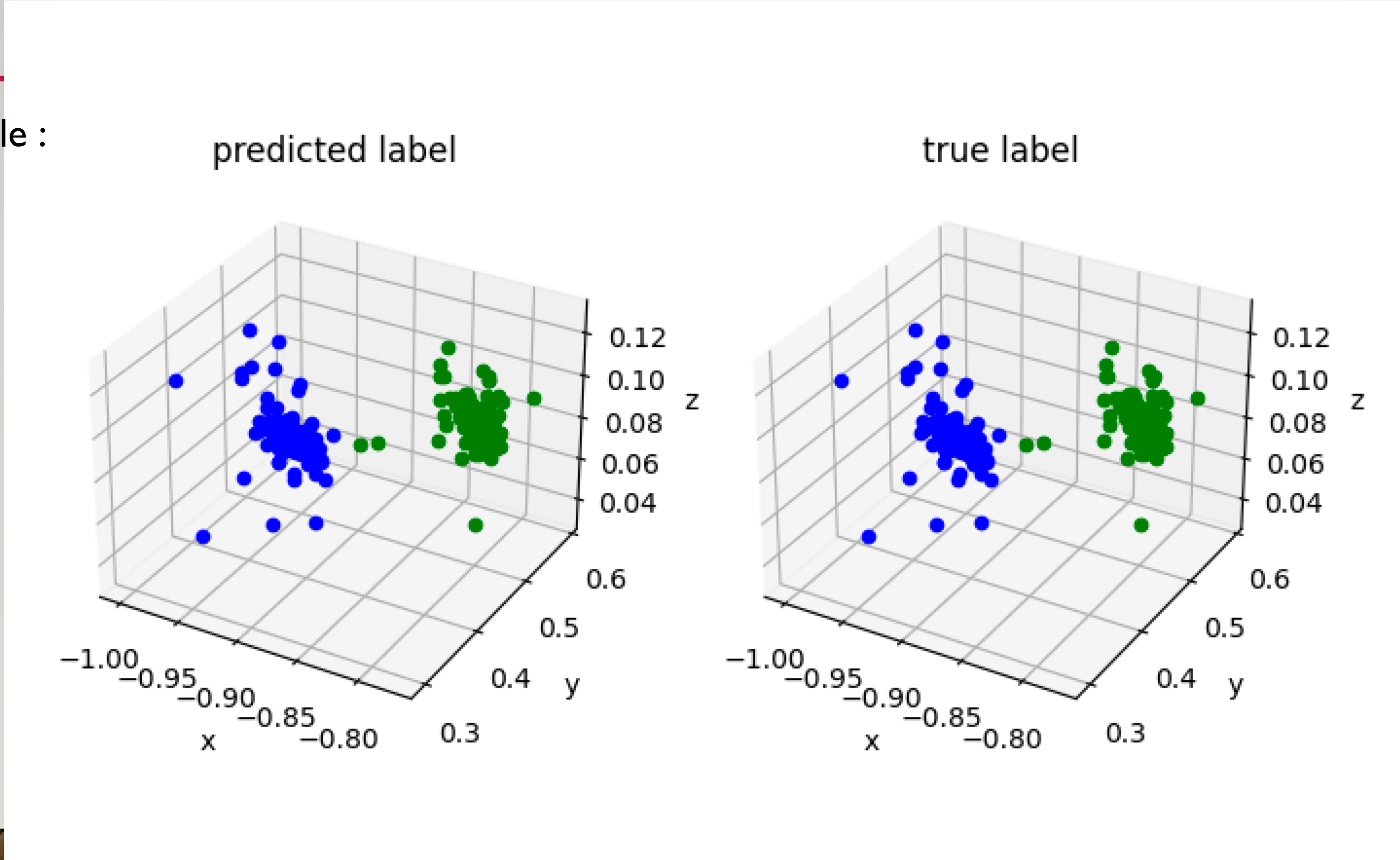
Average = 96.08%





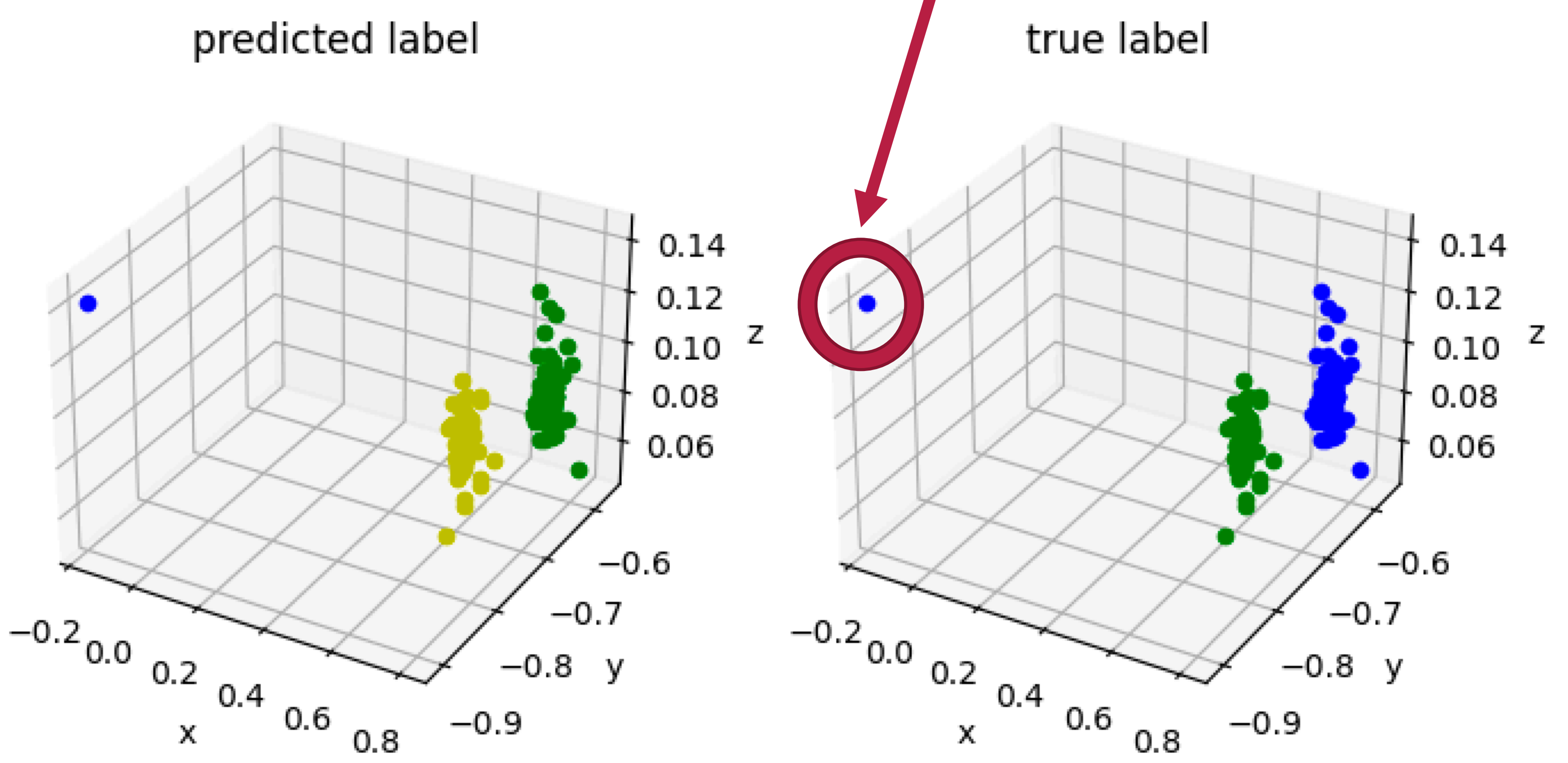
COMPARISON BETWEEN PREDICTION AND TRUE LABEL

Good example :

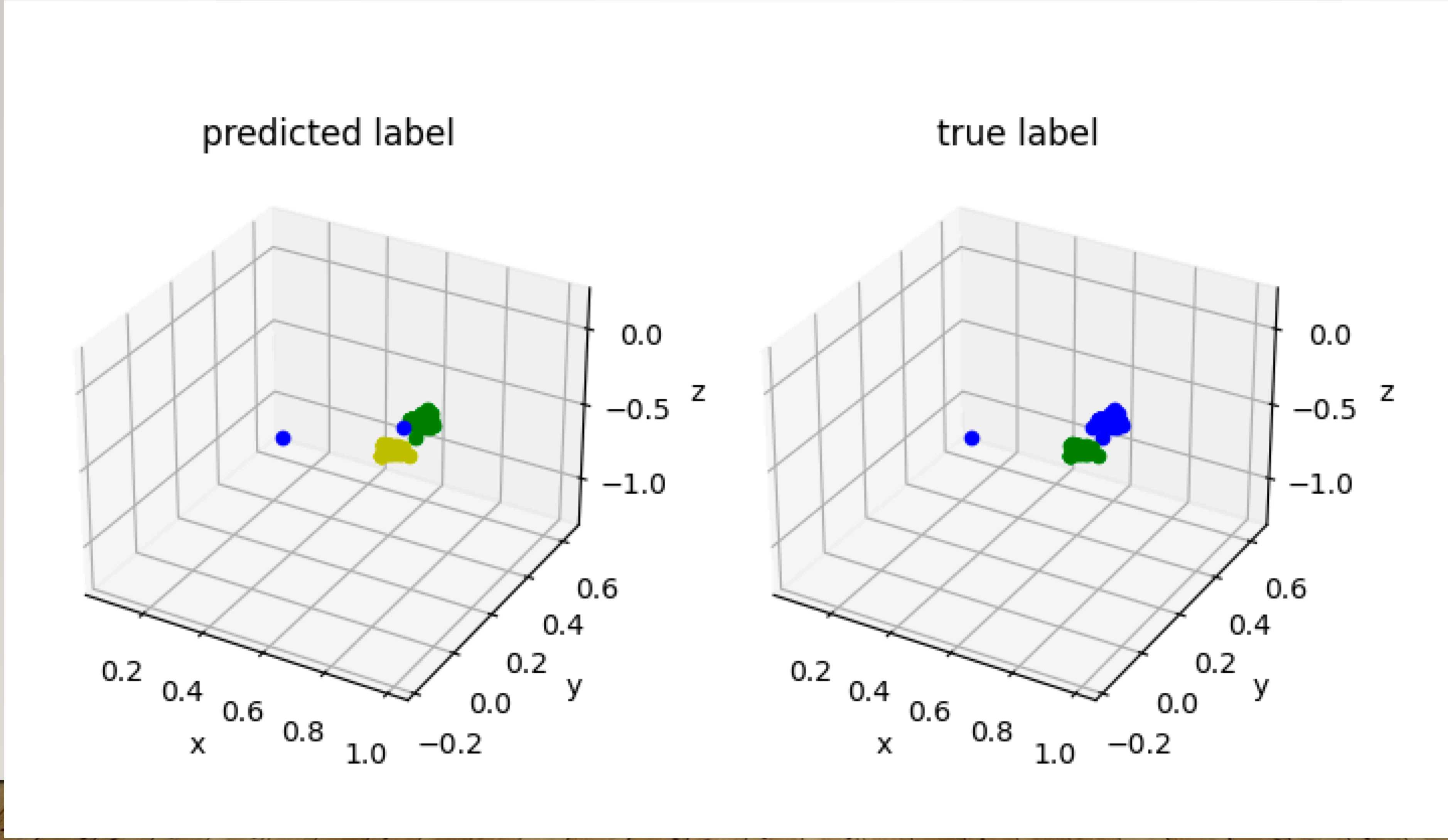


COMPARISON BETWEEN PREDICTION AND TRUE LABEL

The case in which there is a distant hit

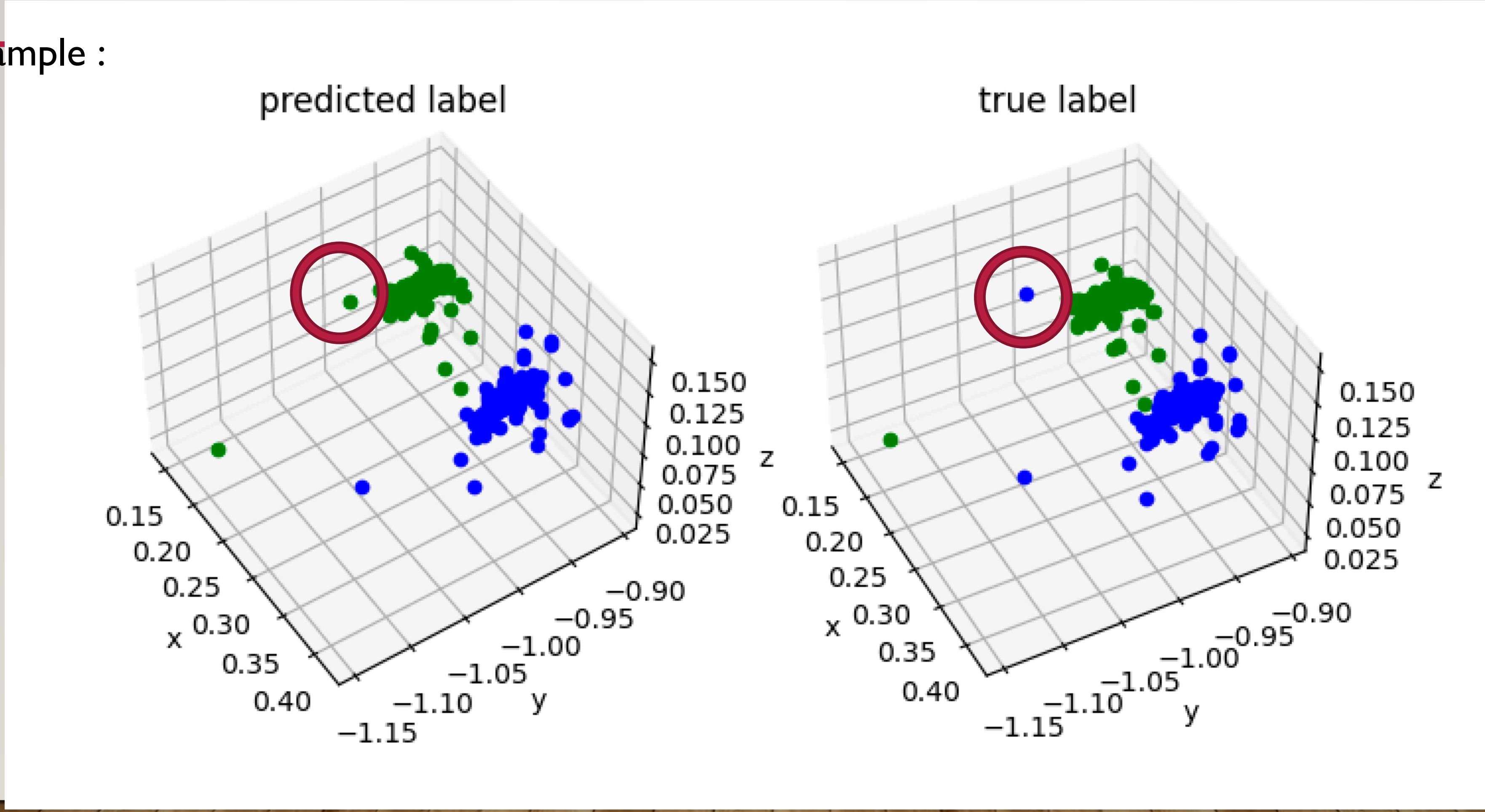


COMPARISON BETWEEN PREDICTION AND TRUE LABEL

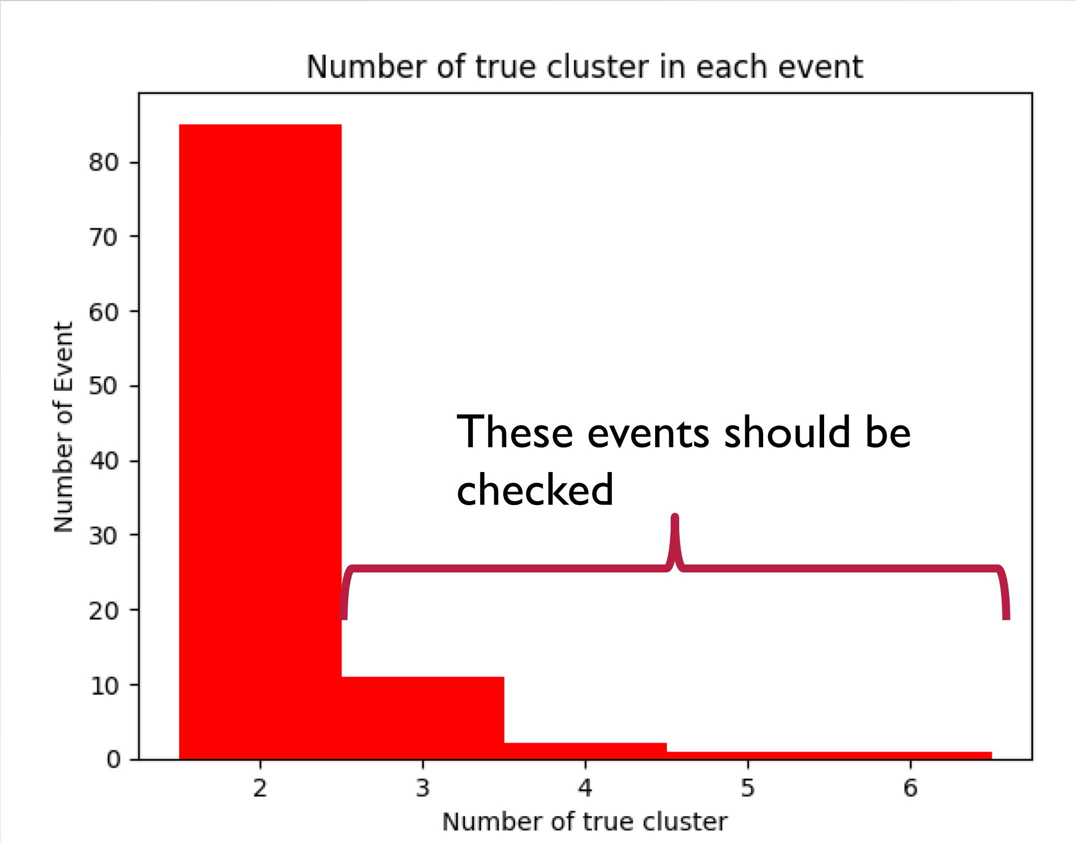


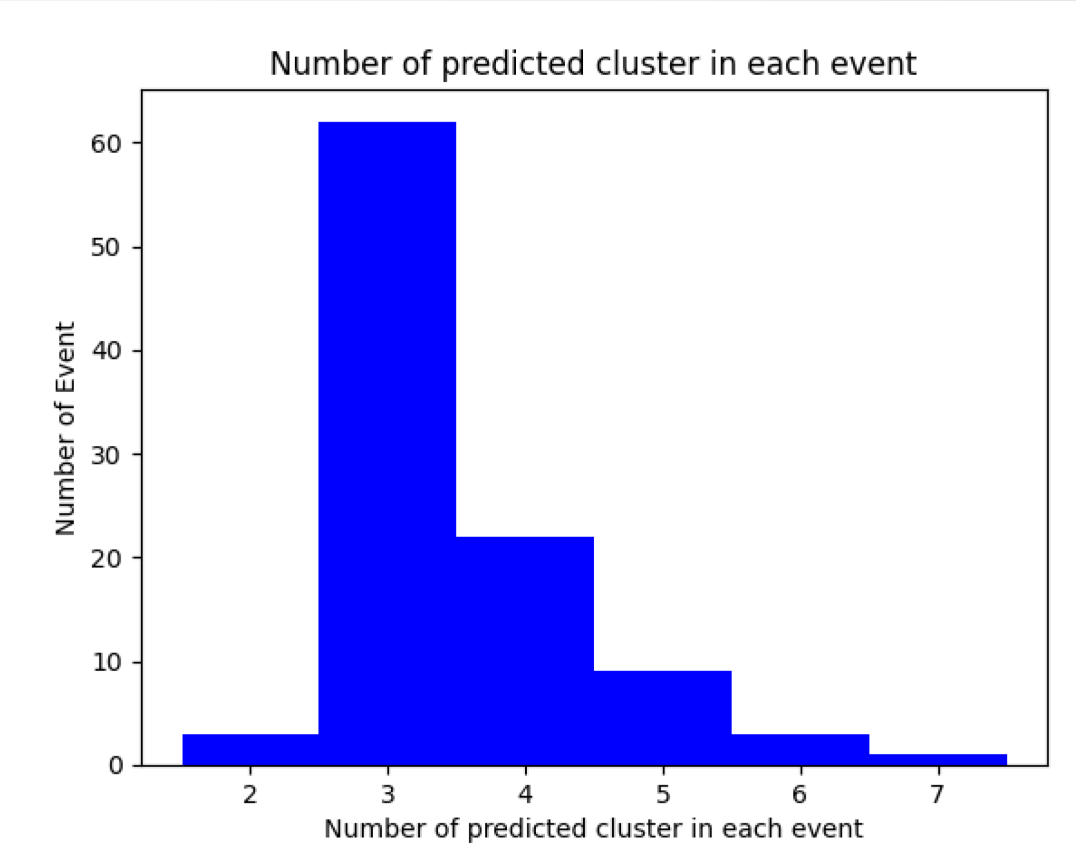
COMPARISON BETWEEN PREDICTION AND TRUE LABEL

Confusion example :



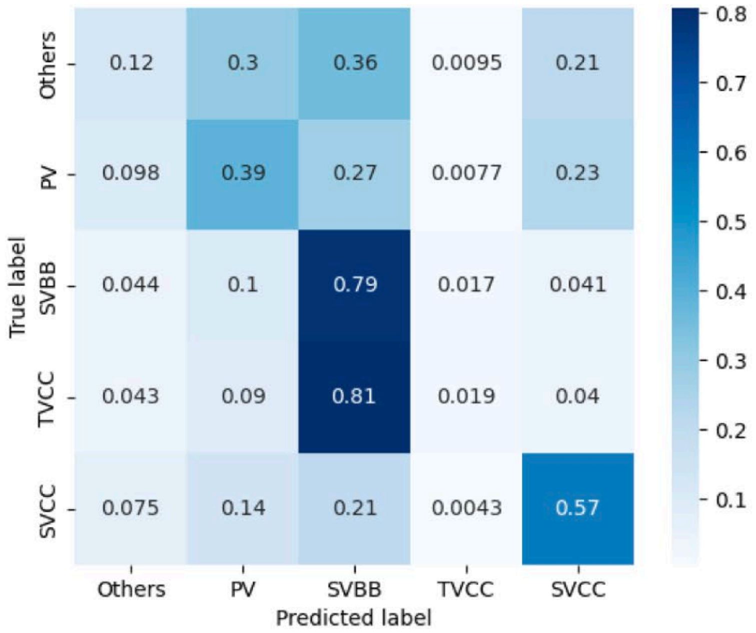
NUMBER OF CLUSTER IN EACH EVENT(JUST 100 EVENTS)



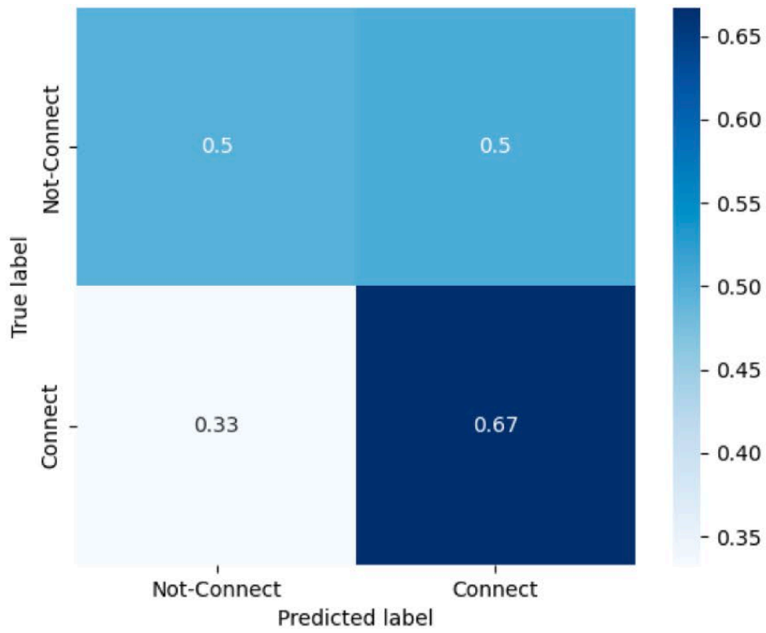


Result of GNN

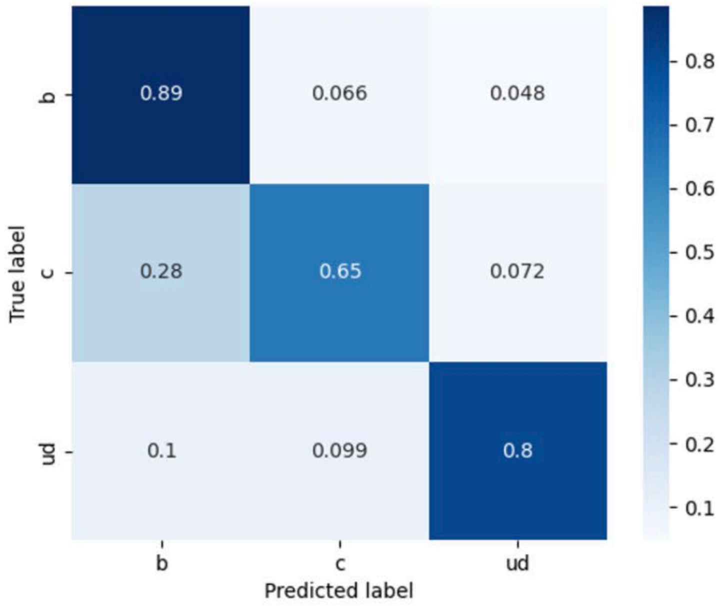
Node classification



Link prediction



Graph classification



- Not much classification of TVCC and SVCC
- Edge connection is not good
- As a graph, we got better accuracy than nodes and edges