Graph Neural Network Jet Flavor Tagging at ILC

2nd general meeting of ILC-Japan Physics Working Group

2022/02/22

T. Onoe[A], T. Suehara[A], K. Kawagoe[A], T. Yoshioka[A], H. Nagahara[B], Y. Nakashima[B], N. Takemura[C]
(Kyushu Univ.[A], Osaka Univ.[B], Kyushu Institute of Technology[C])

Jet Flavor Tagging

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

- 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

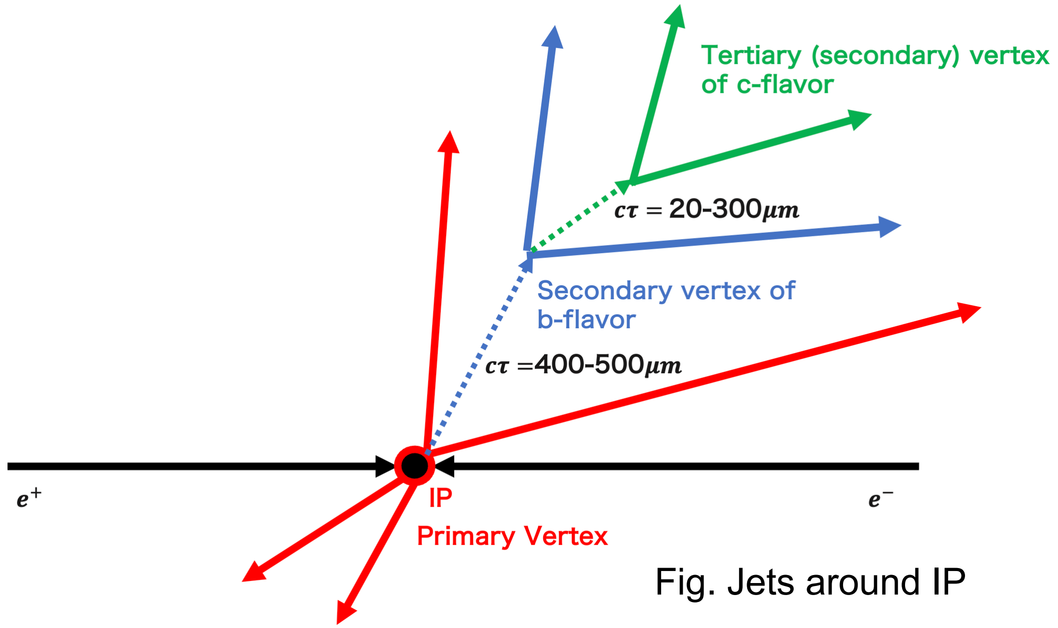


Fig. Jets around IP

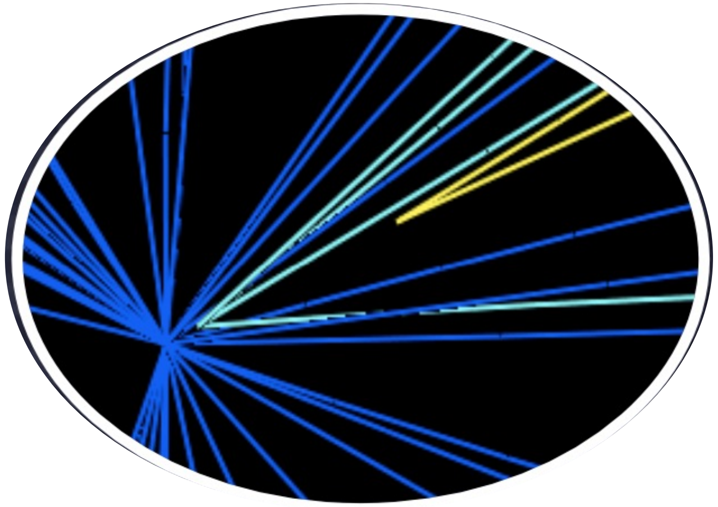
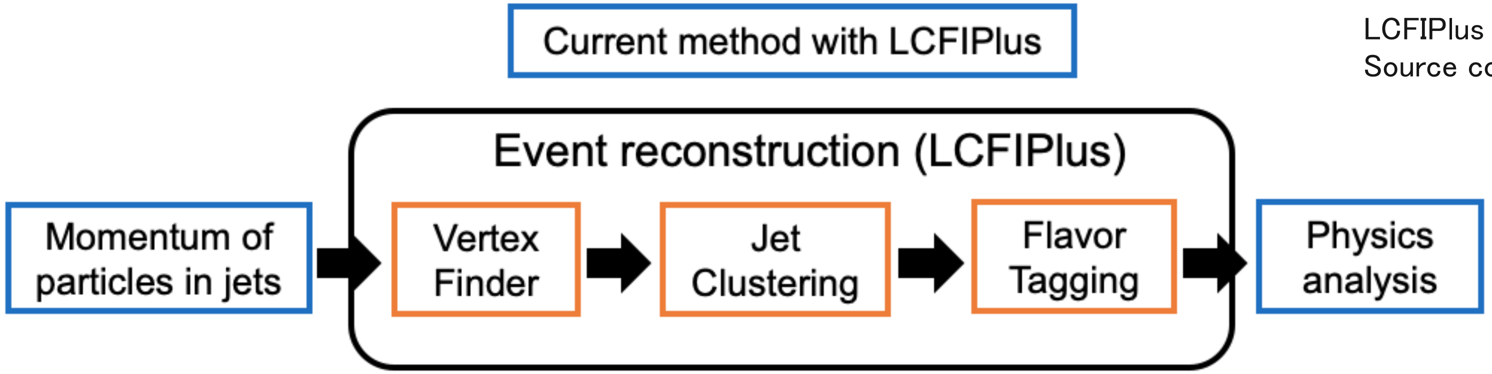


Fig. Monte Carlo simulation of the jet near the IP

Current method of flavor tagging

LCFIPlus paper: NIM A 808 (2016) 109-116
Source codes: https://github.com/lcfiplus/LCFIPlus



1. Vertex Finder : Find vertices by cut-based fitting
2. Jet Clustering : Reconstruct jets by clustering particles
3. Flavor Tagging : Classify jets as b/c/others by <u>Boosted Decision Trees(BDTs)</u>

Purpose of this study

Challenges for LCFIPlus

1. The complexity of trees is limited
   → Cannot express jet representation enough for flavor tagging with low-level input.
2. Fitting calculation of vertex is performed for all track pairs
   → Large amount of time cost for computing

Purpose of this study

1. By implementing Deep Learning, create a model that can represent jets more suitable for better accuracy and efficiency in terms of time
2. From some training approach, implement flavor tagging with low-level input.

Deep Learning (Deep Neural Network; DNN)

- Learning data features → High prediction accuracy for unknown data
- supervised learning on data with answers
- Deep Learning has complex expressions and good scalability

- Various networks and calculation methods exist

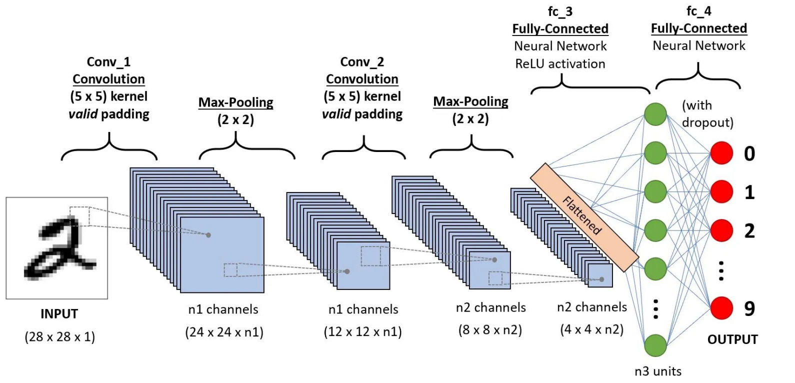
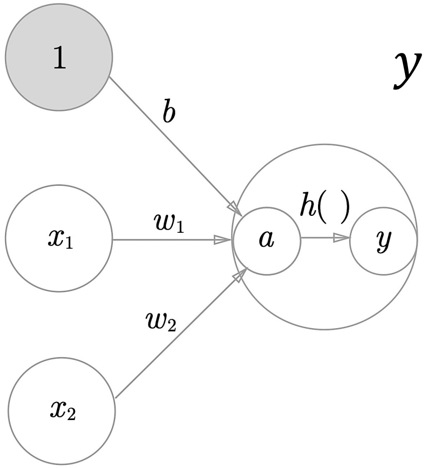


Fig. Convolution neural network

Fig. Overview of computation in a neural network



$$y = h(x_1 w_1 + x_2 w_2 + b)$$

x : input
w : weight
b : bias
a : linear sum
$h()$: non-linear function
y: output

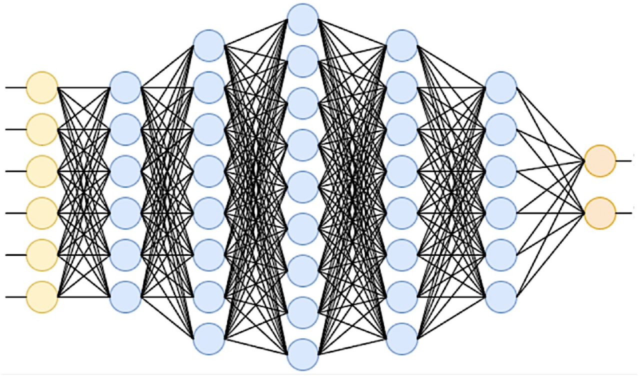


Fig. Fully-connected neural network

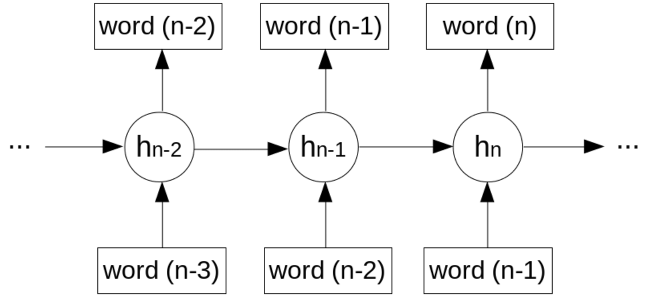


Fig. Recurrent neural network

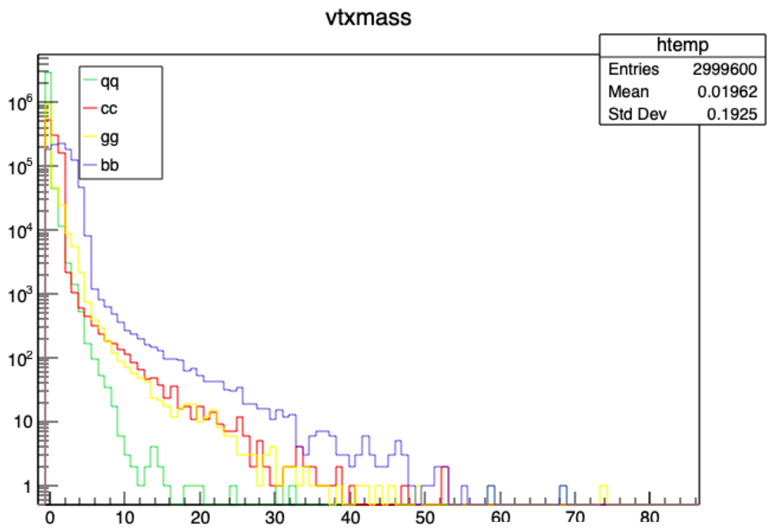
Data information for DNN

Prepare data

- 4 million events data from ILD full simulation (250GeV, bb:cc:qq = 1:1:4)
  $[\, e^{+}e^{-} \rightarrow \nu\bar{\nu}h \rightarrow \nu\bar{\nu}b\bar{b}/c\bar{c}/q\bar{q} \;\; (q = u, d, s)]$
- 42 variables from vertex finder is used for training
  (e.g., number of vertices, position/mass/probability etc.)

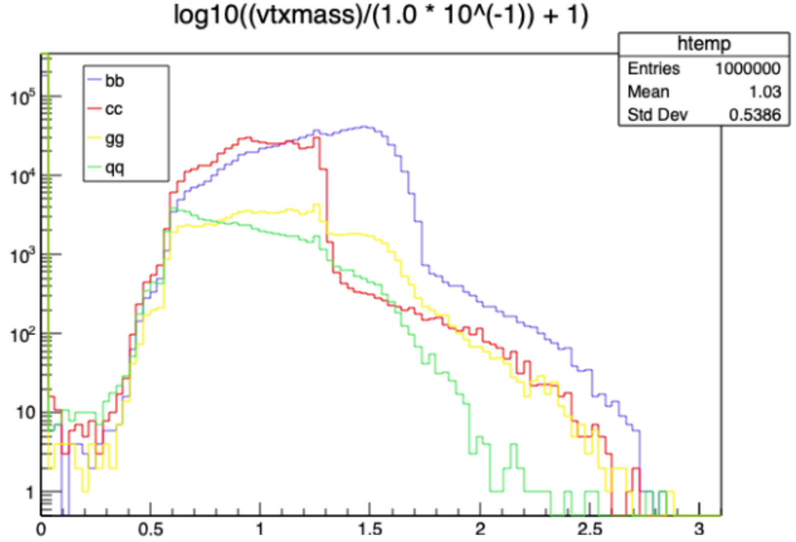
Preprocessing

By transformation of the variable, the distribution of input variables should be flattened and scaled.



Vertex mass

Logarithmic transformation



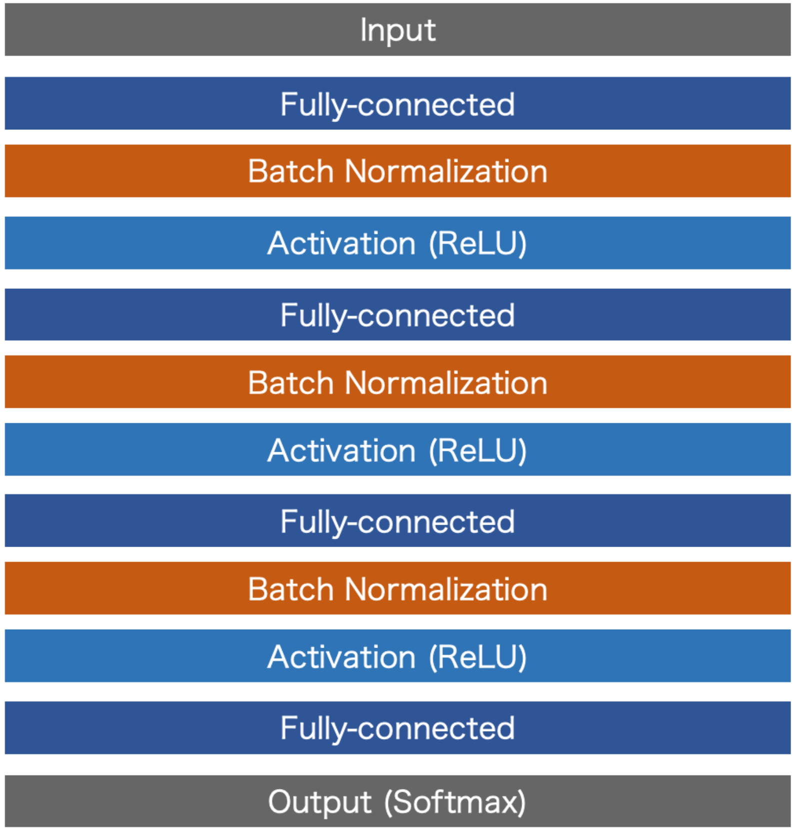
$\log_{10}(\text{Vertex mass} + 1)$

Fully-connected neural network

Training

- Fully-connected neural network (simplest structure)
- Apply batch normalization for each layer
- Output: probability for each label (b/c/uds-likeness)
- Hyperparameters
  - Number of nodes : 512
  - Loss function : Categorical cross entropy
  - Optimization algorithm : RAdam (learning rate 0.001)
  - Number of epochs : 100

Fig. Network architecture



Hyperparameter tuning

- Hyperparameters

  ... Variables to determine the behavior of training

  → Usually tuned manually within reasonable ranges

- We optimize these parameters by Bayesian optimization

- Bayesian optimization in hyperparameter tuning

  ... Define the objective function with hyperparameters as arguments

  Assuming it follows a multivariate Gaussian distribution

  Estimate the objective function by Gaussian process regression

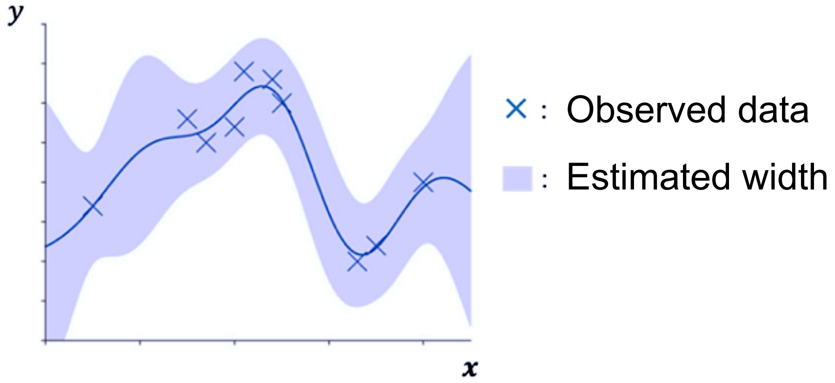
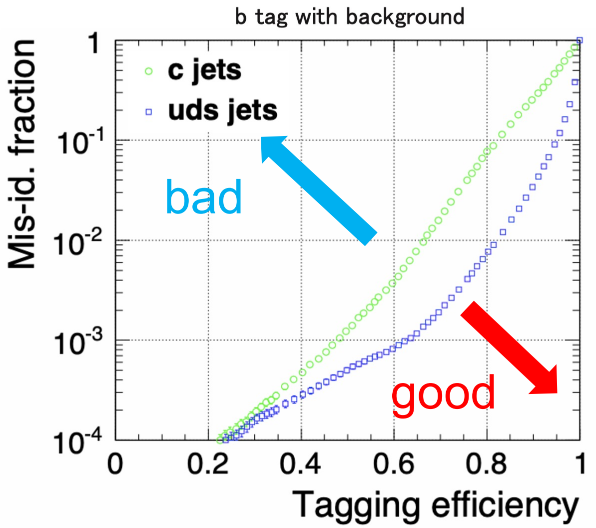


Fig. Search for the objective function (blue line) in two variables

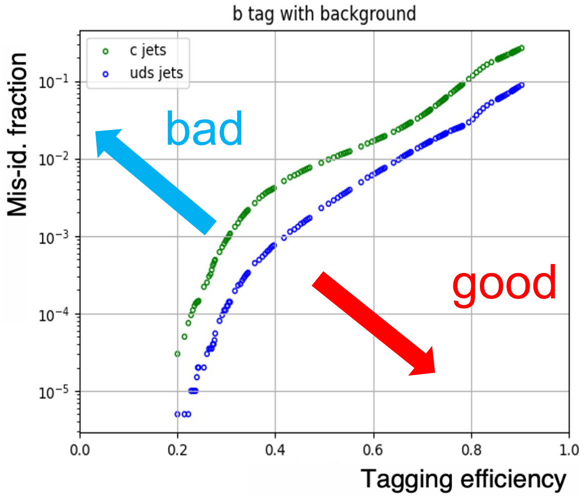
Evaluation of DNN

b tag effciency with background

LCFIPlus

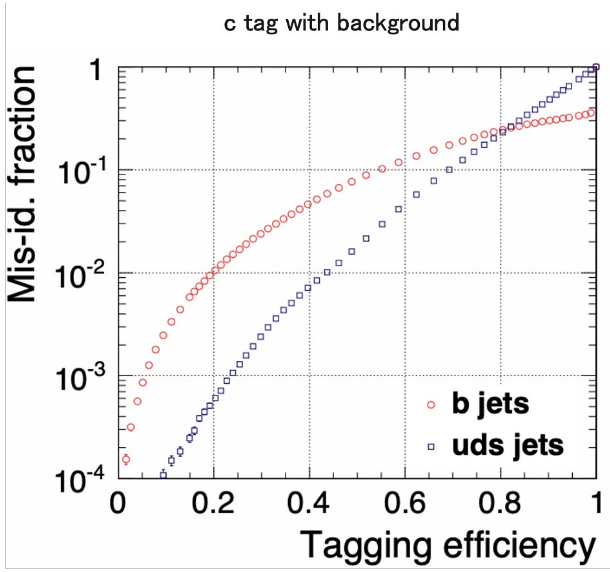


DNN

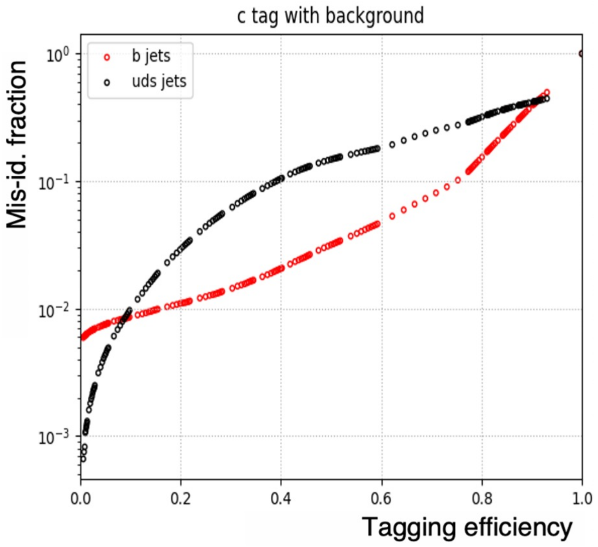


c tag effciency with background

LCFIPlus



DNN



Tagging efficiency = 0.8	background	Mis-id fraction LCFIPlus	Mis-id fraction DNN
b jet	c jet	0.073	0.088
	uds jet	0.007	0.023
c jet	b jet	0.22	0.13
	uds jet	0.24	0.28

- Didn't completely outperform LCFIPlus
- Aiming to improve accuracy by changing method of learning

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

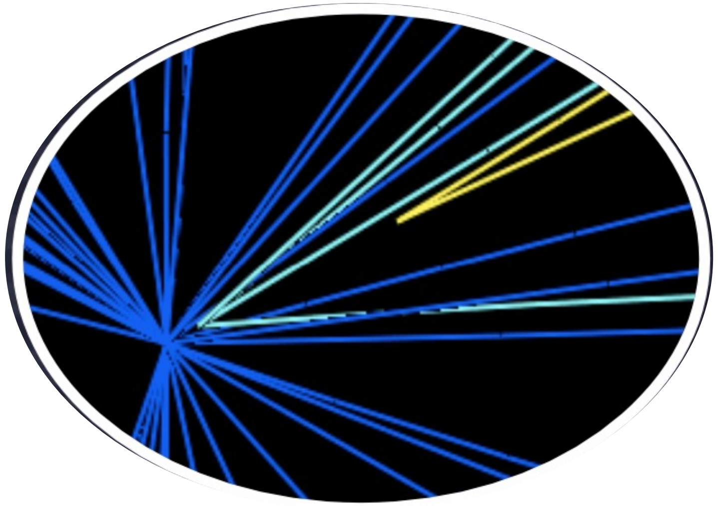


Fig. Event display of Monte-Carlo simulation

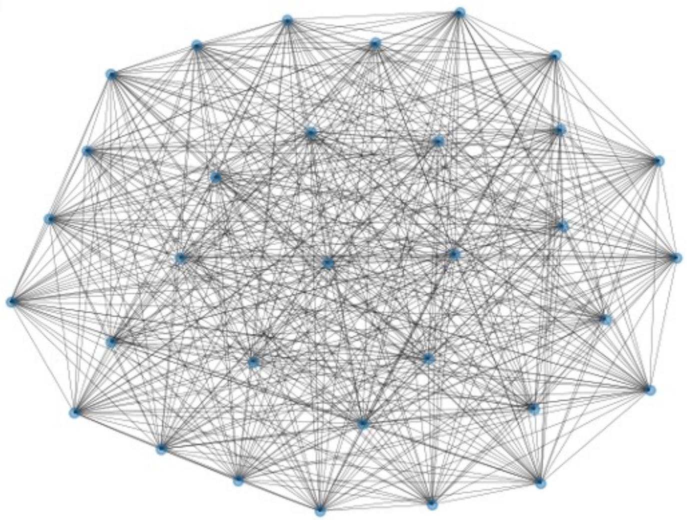


Fig. example of a jet as a graph

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 $\mathbf{z_0}$**

: one track

: connect track pairs
(can be a vertex)

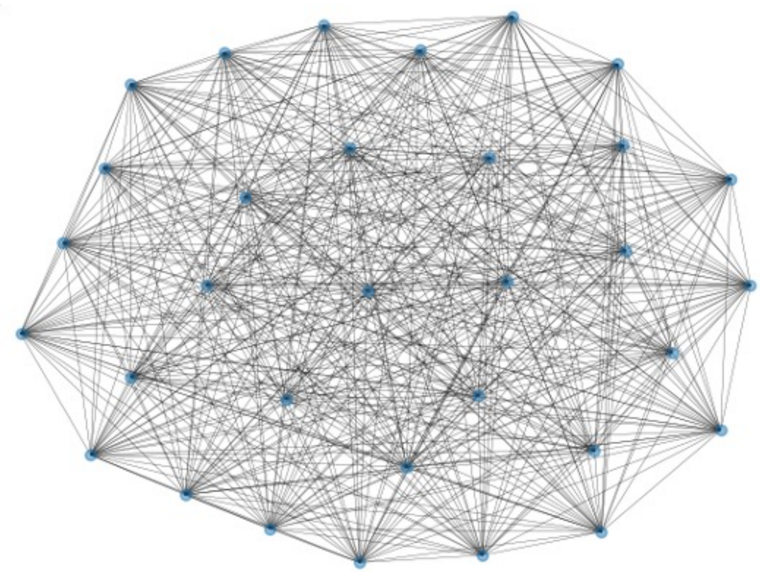


Fig. example of a jet as a graph

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
- <u>Attention mechanism</u> ... Learn the importance score for each weight
  Take as a coefficient for update parameter.

→ <u>Aimed by attention expressing whether tracks has the same vertex</u>.

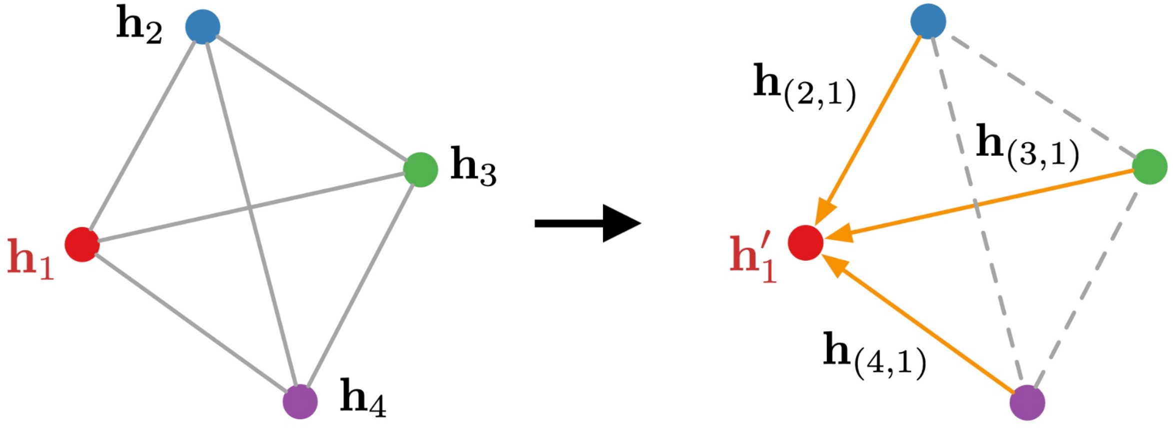


Fig. Graph Training

arXiv [1710.10903]

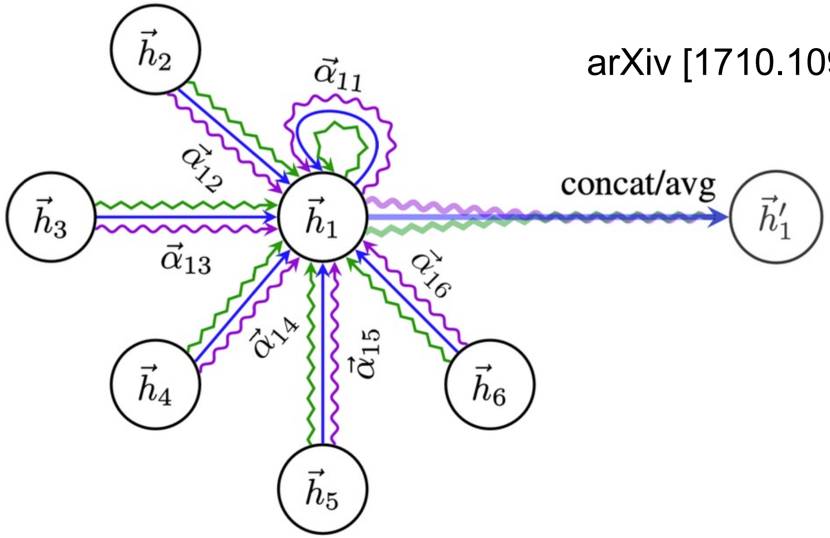


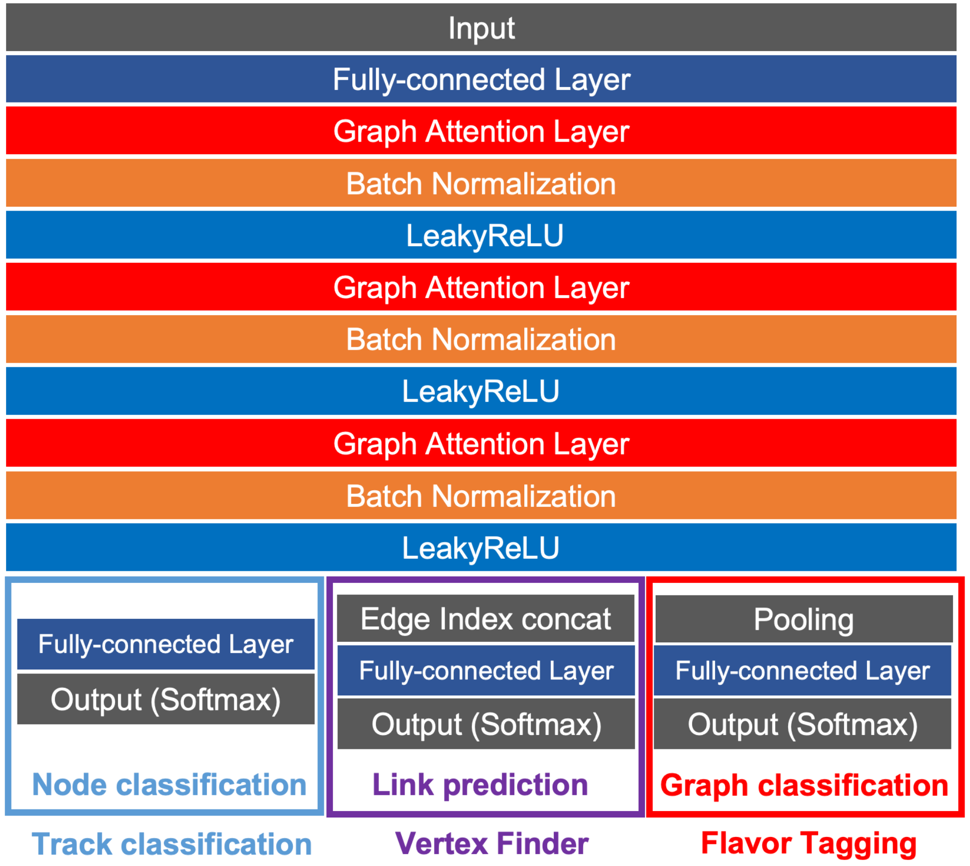
Fig. Graph Attention Network

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

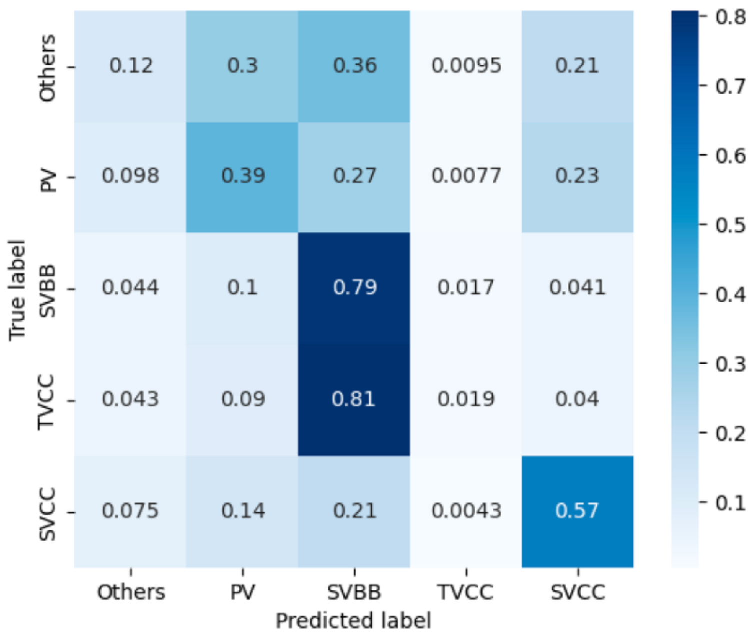
Label	Description
Connected	tracks are connected
Not-connected	tracks are not connected

Graph Classification

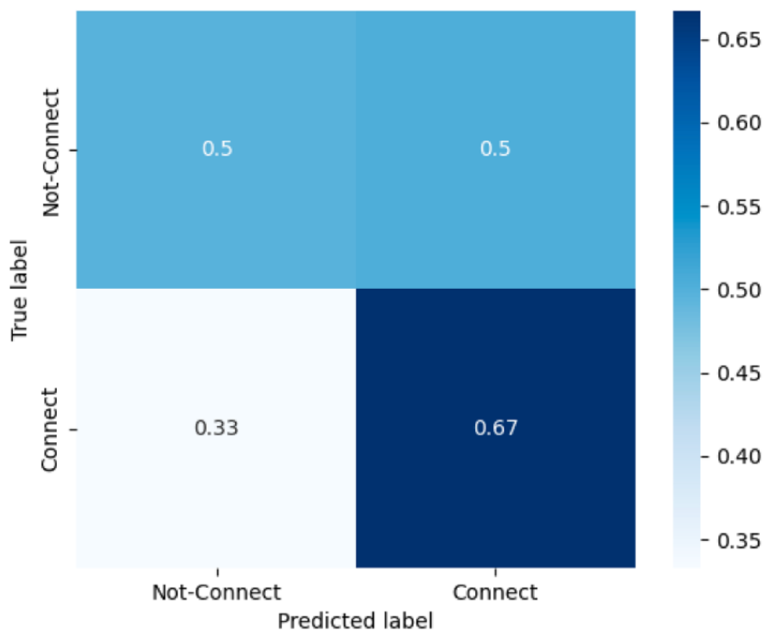
Label	Description
$b\bar{b}$	the final state of $b\bar{b}$
$c\bar{c}$	the final state of $c\bar{c}$
$q\bar{q}$	the final state of $q\bar{q}$ $(q = u, d, s)$

Result of GNN

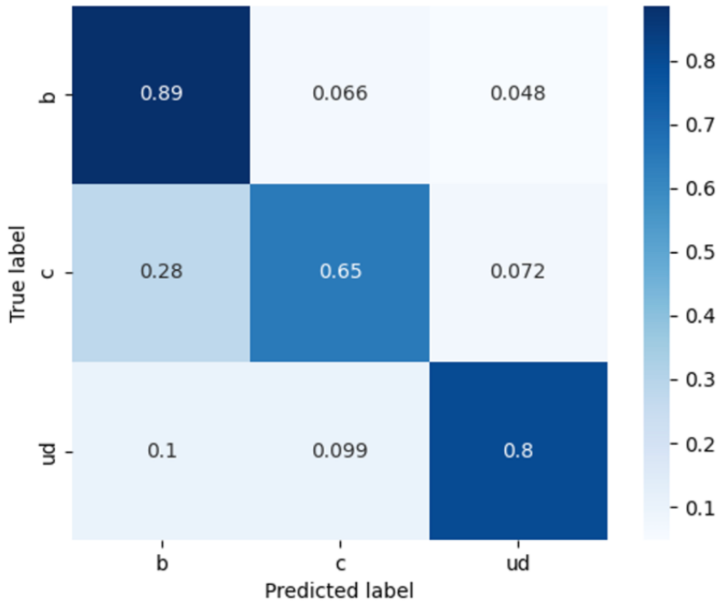
Node classification



Link prediction



Graph classification

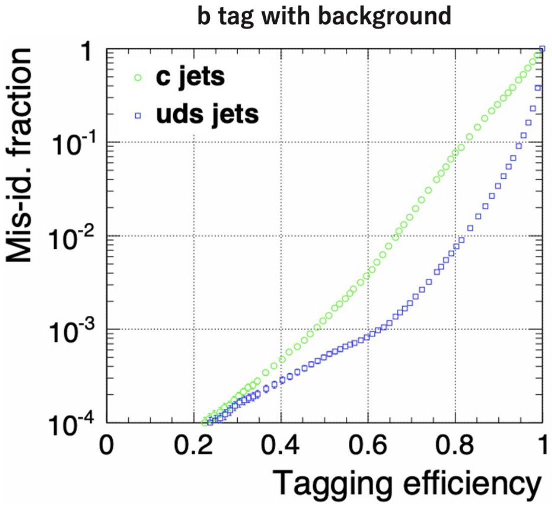


- Not much classification of TVCC and SVCC
- Edge connection is not good
- As a graph, we got better accuracy than nodes and edges

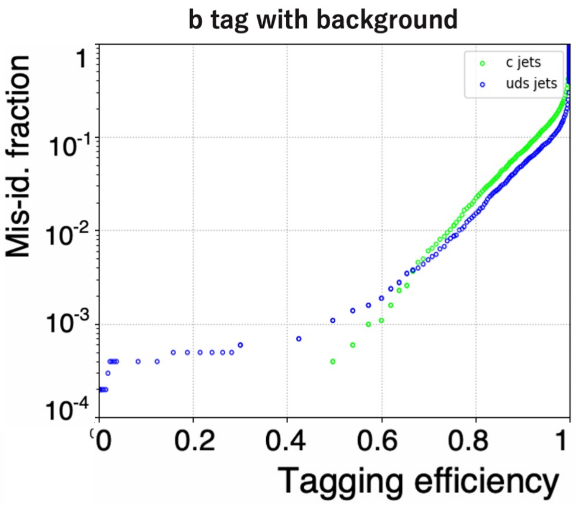
Evaluation of GNN

B tag effciency with background

LCFIPlus

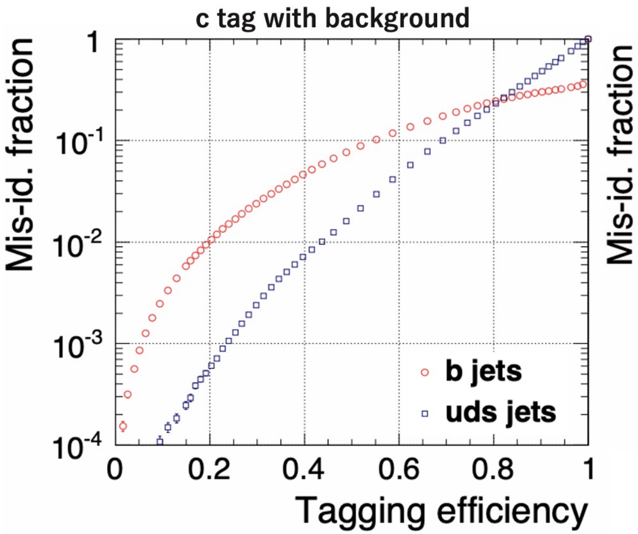


Graph Approach

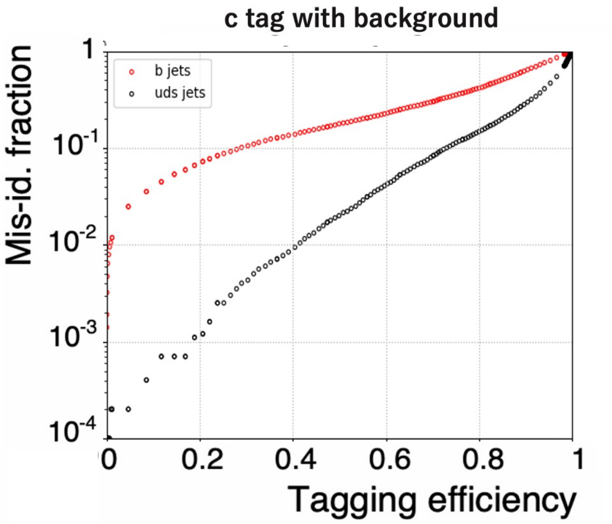


C tag efficiency with background

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

Summary

Conclusion

- Jet flavor tagging is important for Higgs studies on ILC
- We developed the algorithm of jet flavor tagging by Graph Neural Network

✓ Performance improved for some type of jets
✓ Time cost of computation improved
✓ Integrated of Flavor Tagging with Vertex Finder

Discussion

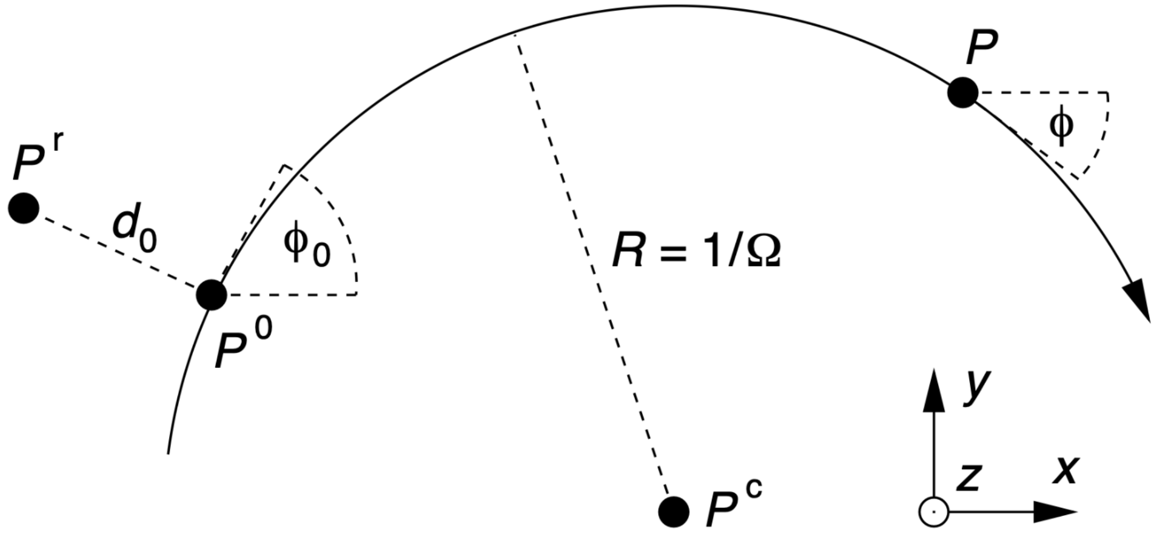
Not sufficiently classification about nodes and edges

➢ Review input data preprocess
➢ Edge features can improve accuracy (from Vertex Fitter)
➢ Understand how out attention is working
➢ Another network model is better?

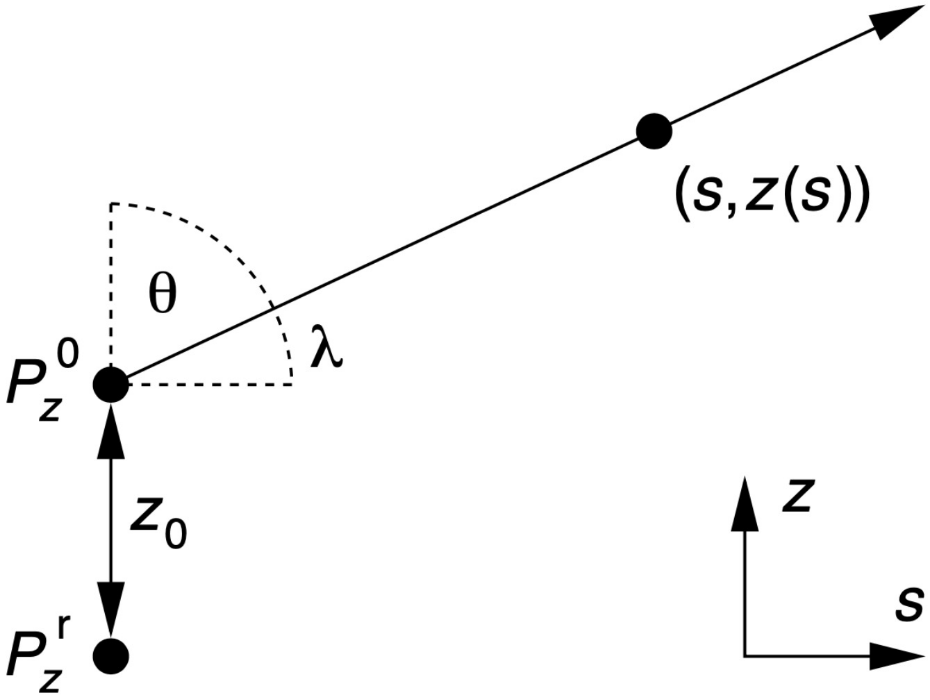
Backup

xy, sz coordinate

xy coordinate



Sz coordinate



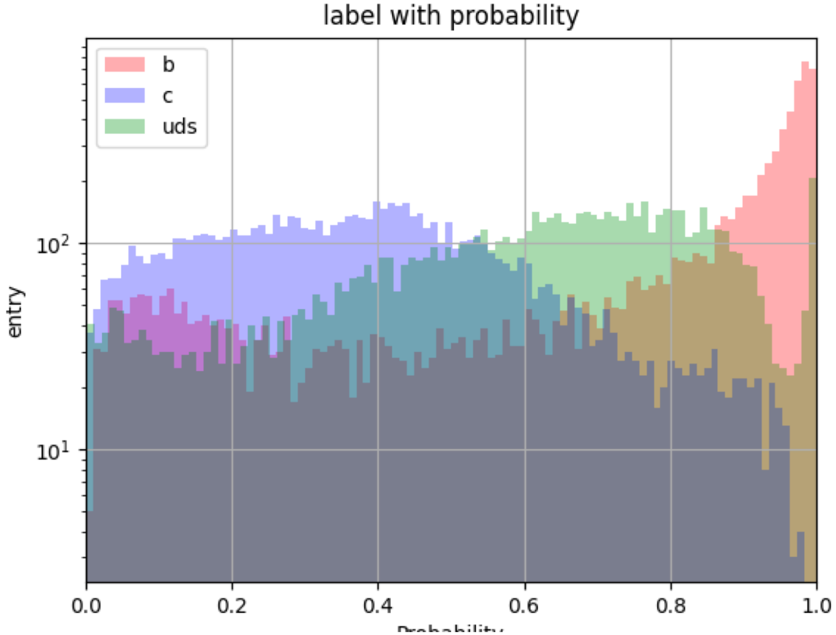
Flavor Tagging Efficiency 補足

- 分類問題における出力 (softmax関数) は、確率値に正規化

  (例) (b probability, c probability, uds probability) = (0.7, 0.2, 0.1)

- Tagging Efficiency は判定の閾値を変え、残る割合を表す

  → b-tag Efficiency = (閾値を超えてbと判断されたジェット) / (全てのbジェット)



ハイパーパラメータ最適化

- 探索を行ったパラメータは右表

RAdamに関する
パラメータ

変数名	最適化範囲
中間層のノード数	$[32, 64, 128, 256, 512, 1024]$
ドロップアウト	$0.1 \sim 0.9$
LeakyReLU	$[0.00001, 0.0001, 0.001, 0.01, 0.1]$
学習率	$[0.00001, 0.0001, 0.001, 0.1]$
β_1	$0.5 \sim 0.9$
β_2	$0.5 \sim 0.9$
eps	$[10^{-9}, 10^{-8}, 10^{-7}]$
weight decay	$[0.000001, 0.0001, 0.001, 0.01, 0.1]$

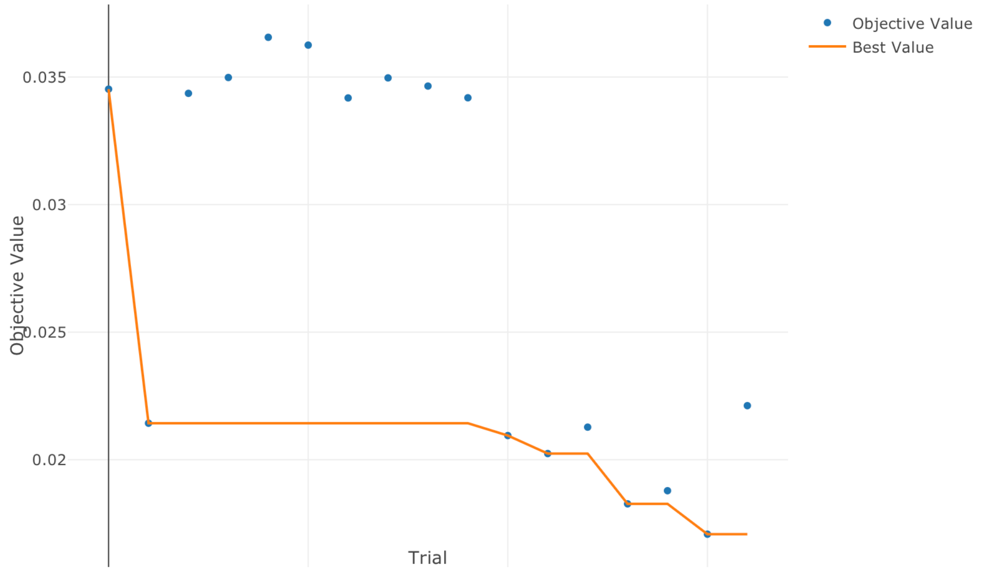


図. 各試行における目的関数(損失関数)の値

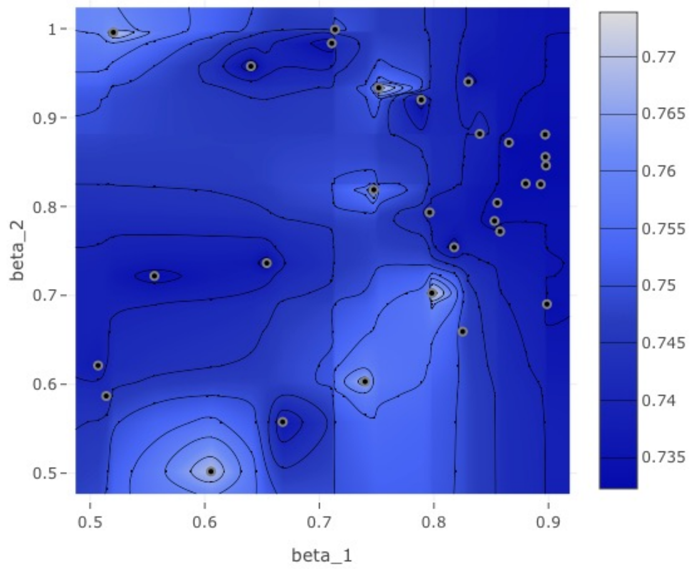


図. β_1, β_2に対する目的関数の値分布