Jet flavor tagging in pp collisions using GNN for the ALICE experiment

Changhwan Choi

Nuclear Physics Laboratory, Department of Physics, Pusan National University

Contents

Google Colab Notebook here!

https://colab.research.google.com/drive/1pr1tb-qPGV6TnneKSxjyvq07IwkvNe2a#scrollTo=2-D-0cy9Txf_&forceEdit=true&sandboxMode=true

Lecture notes about GNN from Stanford Univ.
https://web.stanford.edu/class/cs224w/

0. PyTorch & GNN

- torch.nn.Module

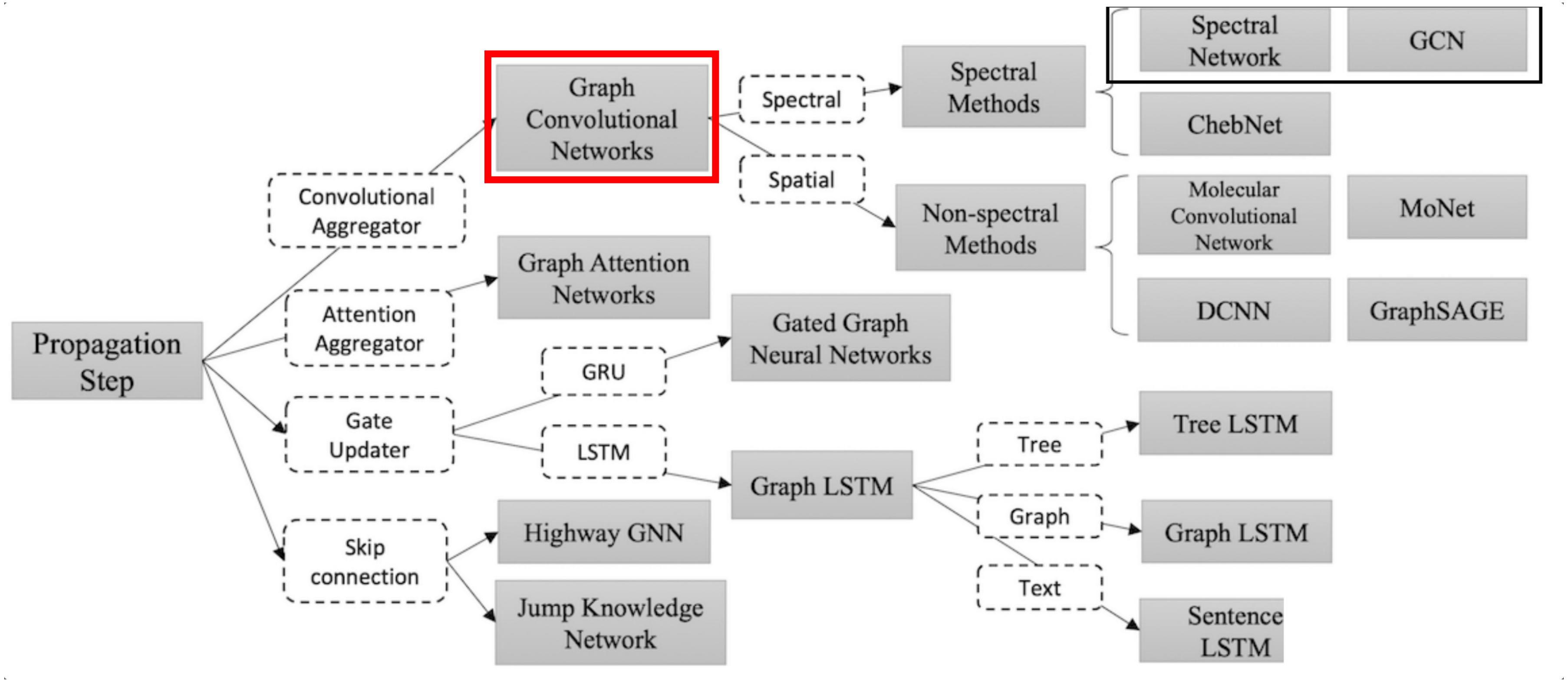
```
class MyModule(nn.Module):
    def __init__(self):
        ”””
         Define layer structure
        ”””
        pass
    def forward(self, x):
        ”””
         Return layer output
        ”””
        pass
```

```python
import torch
import torch.nn as nn
import torch.nn.functional as F

class GraphConvolution(nn.Module):
    def __init__(self, in_features, out_features):
        super(GraphConvolution, self).__init__()
        self.linear = nn.Linear(in_features, out_features)

    def forward(self, x, adj_matrix):
        x = self.linear(x)
        x = torch.matmul(adj_matrix, x)  # 곱셈 대신에 인접 행렬과의 행렬 곱을
        return x

class GCN(nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        super(GCN, self).__init__()
        self.gc1 = GraphConvolution(input_size, hidden_size)
        self.gc2 = GraphConvolution(hidden_size, output_size)

    def forward(self, x, adj_matrix):
        x = F.relu(self.gc1(x, adj_matrix))
        x = self.gc2(x, adj_matrix)
        return F.log_softmax(x, dim=1)
```

0. PyTorch & GNN

- GNN

Many kinds of GNNs there are…



1. Introduction

Jet flavor tagging

Traditional methods

Neural networks

1. Introduction

- Jet flavor tagging

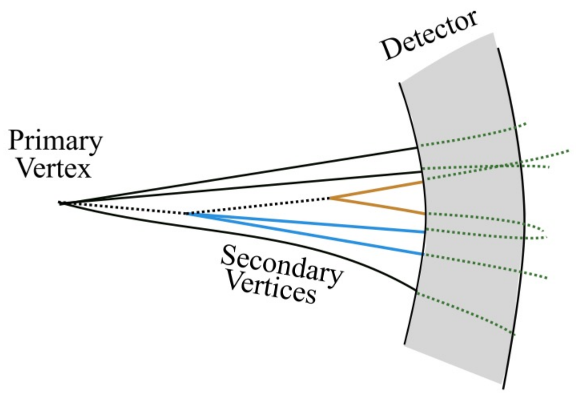
: Identifying which flavor(bottom, charm or light flavor) of parton is responsible for the jet production.

(light = u, d, s quark & gluon)

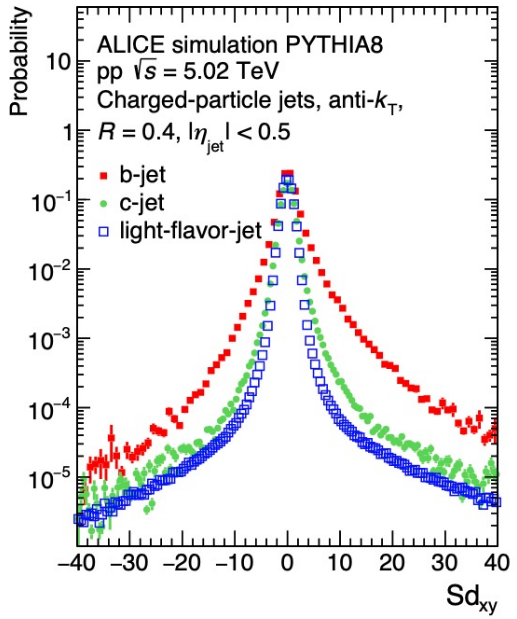
Physical phenomena vary depending on the flavor of quark.

e.g. dead cone effect, longer lifetime of heavy-flavor hadrons

These can be studied through observables such as DCA(track impact parameter), secondary vertices, momenta, the number of jet constituents, etc..



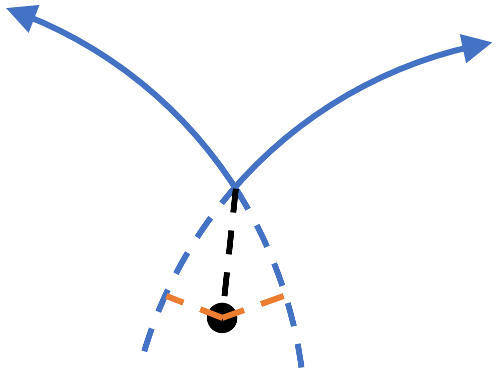
J. Shlomi et al., *Eur.Phys.J.C* (2021) 81:540



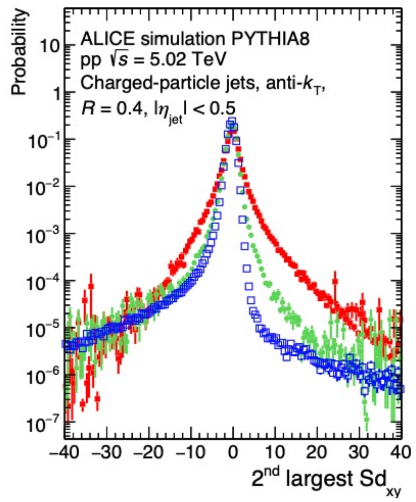
ALICE Collaboration, *JHEP* 01 (2022) 178

1. Introduction

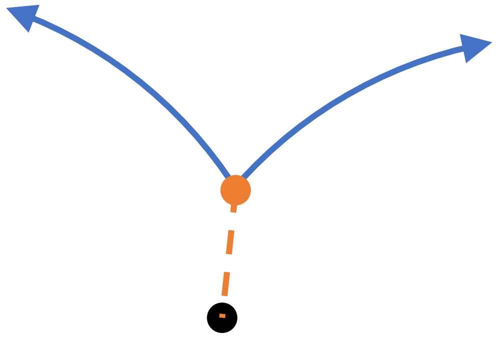
- Traditional methods
  - IP (Impact parameter)



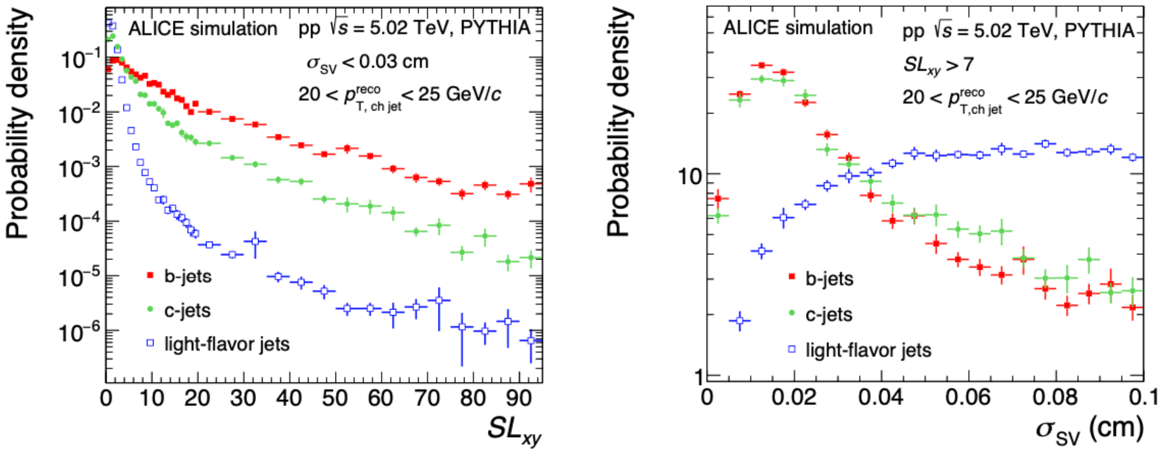
$Sd_{xy} > Sd_{xy}^{\min}$
→ b-jet candidate



  - SV (Secondary vertex)



$SL_{xy} > SL_{xy}^{\min}$
$\sigma_{SV} < \sigma_{SV}^{\max}$
→ b-jet candidate



ALICE Collaboration, op. cit.

1. Introduction

- Neural network

  : Many recent studies using NN for jet flavor tagging are ongoing,
  and they show improved performances compared to previous methods.

  Many different types of neural networks
  : DNN, GNN, RNN, CNN(image), …

  In this research…
  → Secondary vertex finding using Set2Graph NN,
  and jet flavor tagging using Graph Neural Networks (GNN)

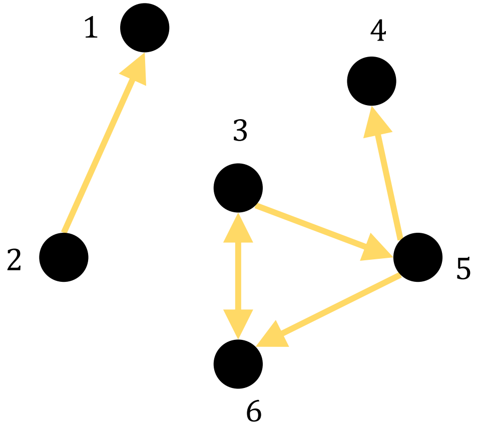
  (Reference: J. Shlomi et al., *Eur.Phys.J.C* (2021) 81:540)

1. Introduction

- Graph (discrete mathematics)

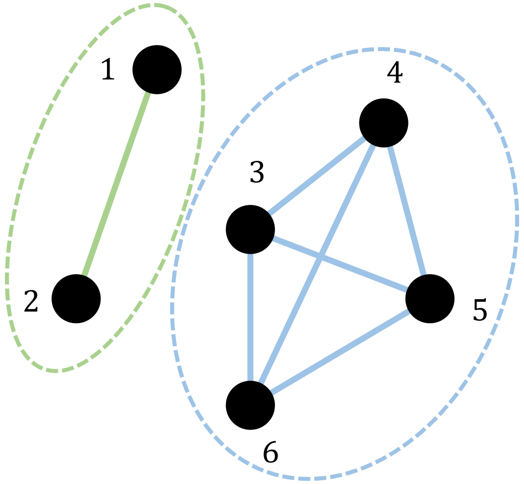
: Sets of Nodes connected by Edges.

In this research, **A Graph** represents A single Jet, **Nodes** correspond to Tracks(jet constituents), and **Edges** correspond to Connections between tracks originating from same vertex.



	1	2	3	4	5	6
1	0	0	0	0	0	0
2	1	0	0	0	0	0
3	0	0	0	0	1	1
4	0	0	0	0	0	0
5	0	0	0	1	0	1
6	0	0	1	0	0	0

▲ directed graph representation



	1	2	3	4	5	6
1	1	1	0	0	0	0
2	1	1	0	0	0	0
3	0	0	1	1	1	1
4	0	0	1	1	1	1
5	0	0	1	1	1	1
6	0	0	1	1	1	1

▲ cluster graph representation

2. Neural network & dataset

Neural network structure

Vertex finding (Set2Graph NN)

Jet flavor tagging (GNN)

Dataset specification

2. Neural network & dataset

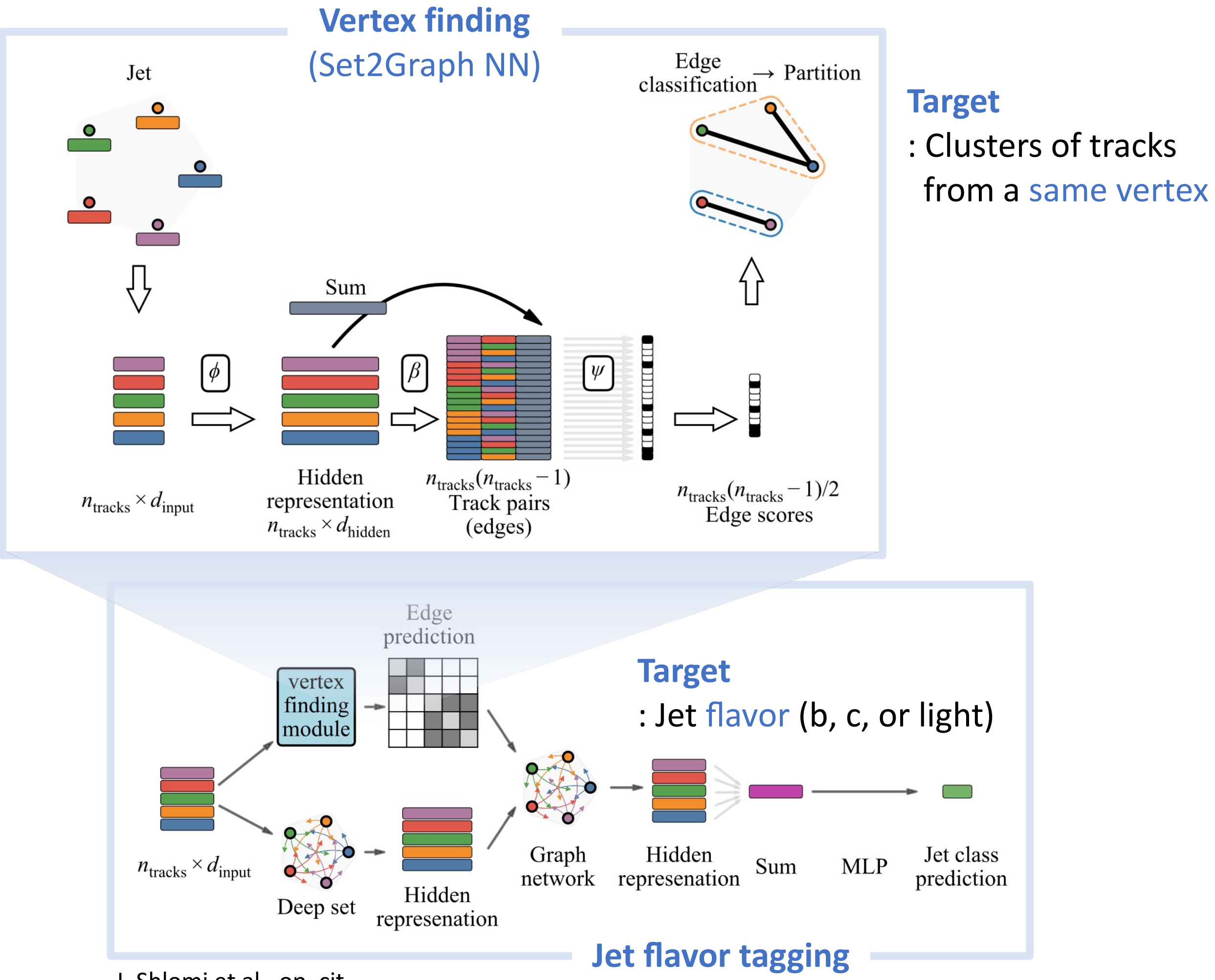
- Neural network structure

  - Vertex finding
    : Set2Graph NN

  → Grouping of tracks originating from a common (primary or secondary) vertex

  - Jet flavor tagging
    : GNN that takes hidden representations of tracks and vertex prediction by vertex finding module as input.

  → Jet flavor (b, c, or light jet)



J. Shlomi et al., op. cit.

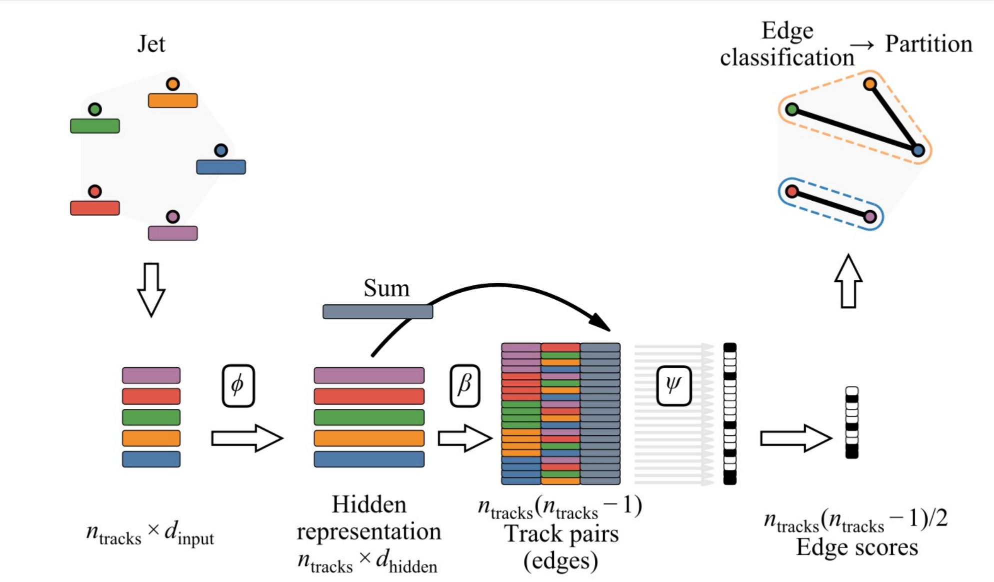
2. Neural network & dataset

- Vertex finding

  : Set2Graph NN

  Input: Set of constituent tracks

  → Output: Graph connecting tracks originating from a common vertex.



ϕ: set-to-set component

→ Deep sets network

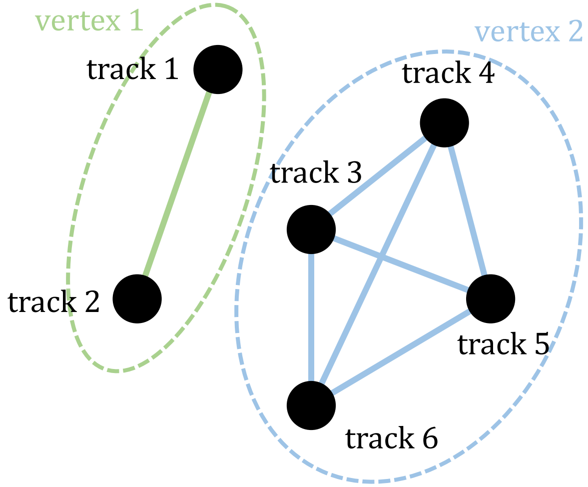
β: broadcasting layer

→ Node representations to edge representations

(Pairs of track i and track j)

ψ: final edge classifier

→ Edge prediction (MLP)



$$\begin{matrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{matrix}$$

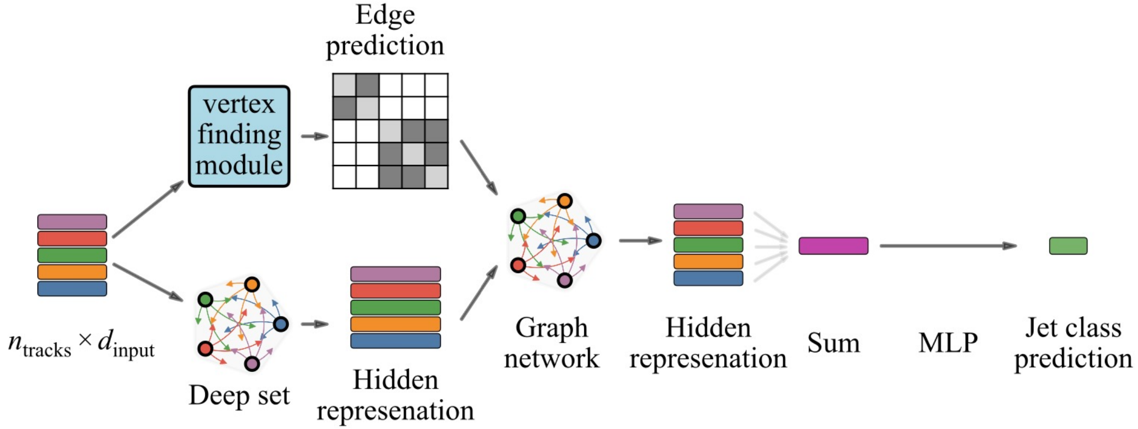
▲ Matrix representation of an example output

2. Neural network & dataset

- Jet flavor tagging

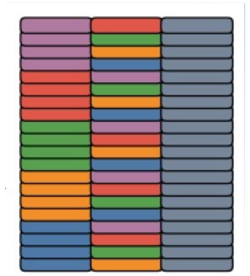
: GNN that takes vertex prediction result of vertex finding module as input.

GNN (Graph Neural Networks)



Input

: Graph consisting of Nodes(features of tracks and a jet) and Edge prediction by the vertex finding module

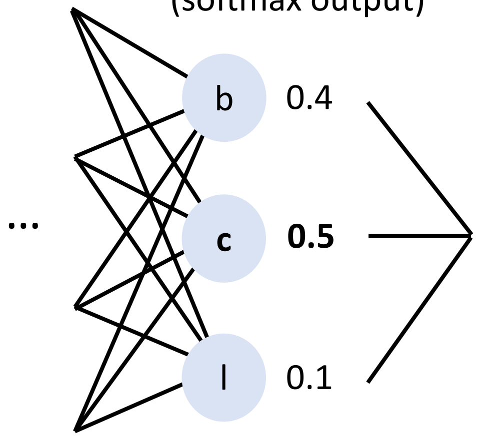


$$+ \quad \begin{matrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{matrix}$$

Output

: The flavor of the jet (b, c, or light jet)

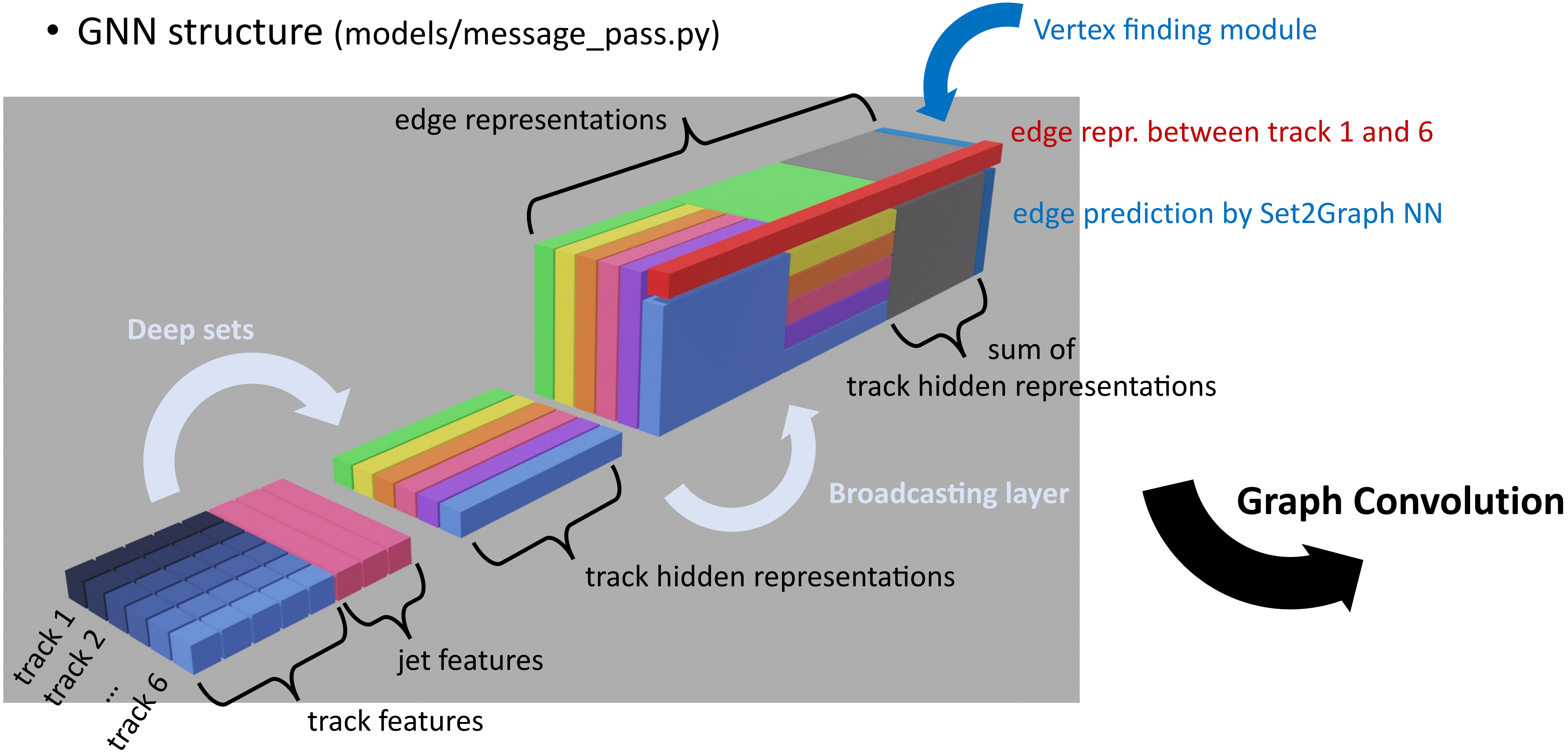
(softmax output)



c-jet

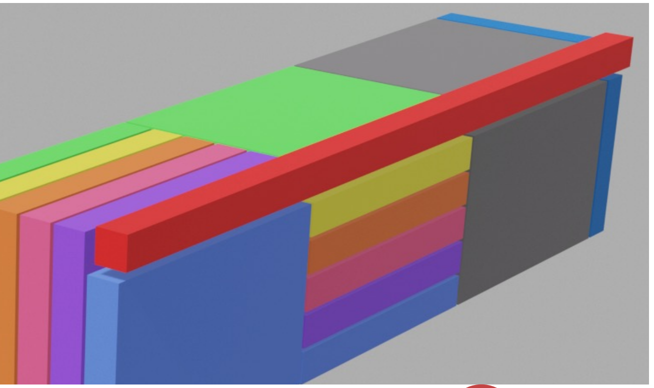
2. Neural network & dataset

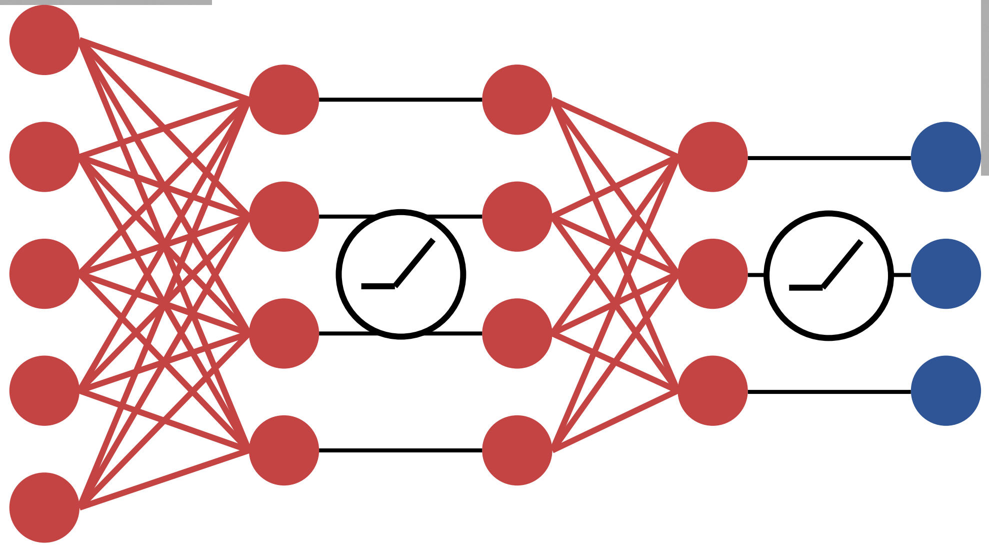
- GNN structure (models/message_pass.py)

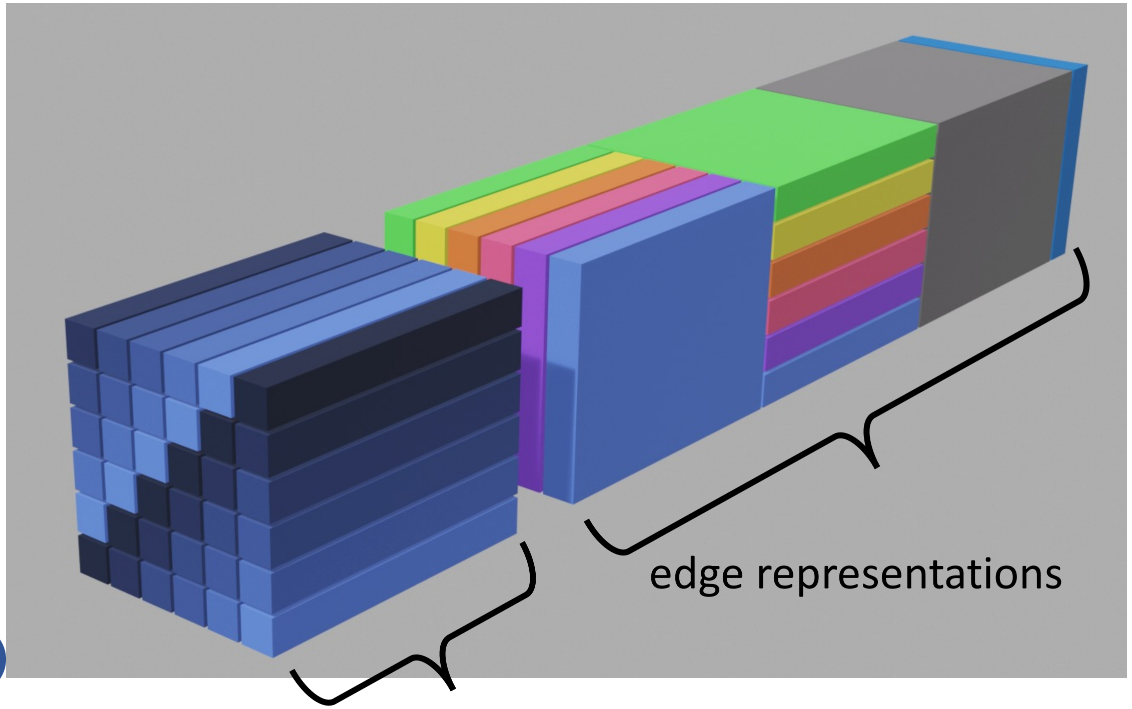


2. Neural network & dataset

- GNN structure (models/message_pass.py)

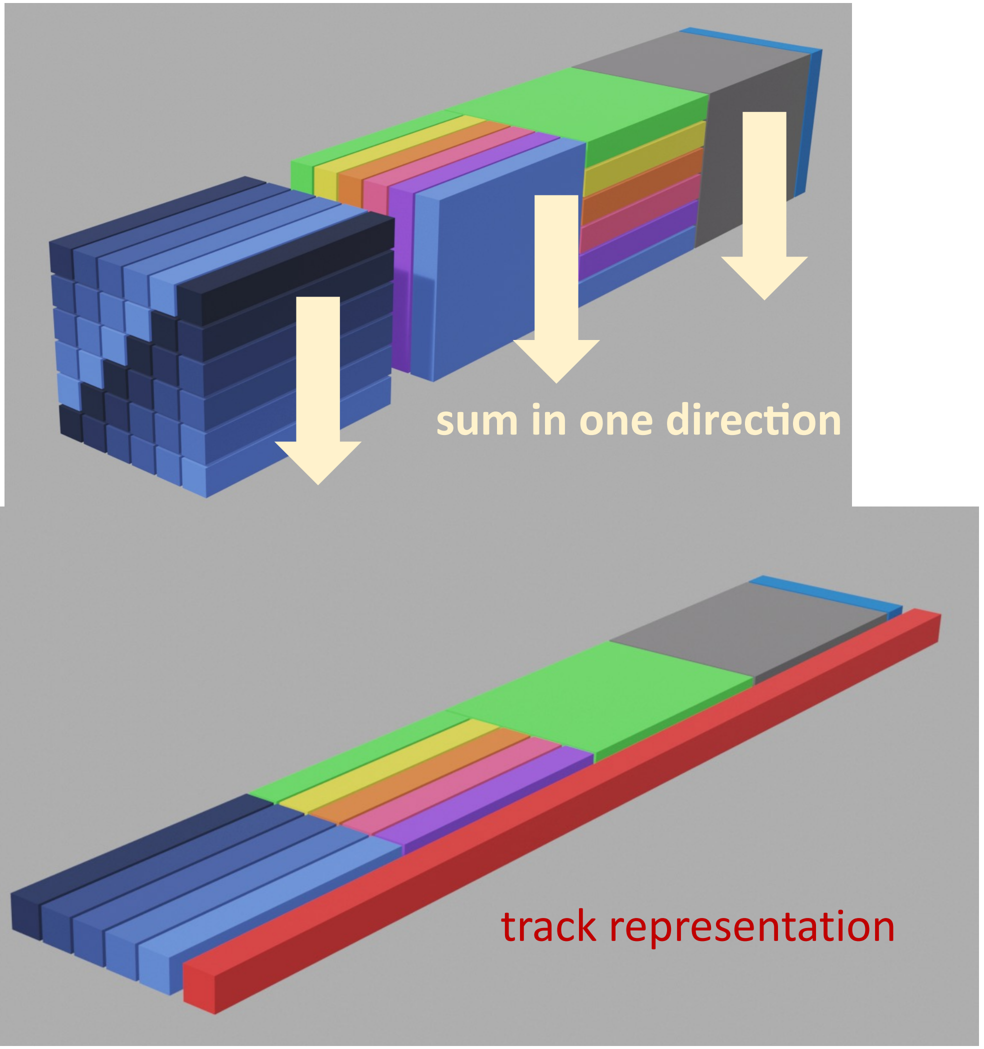


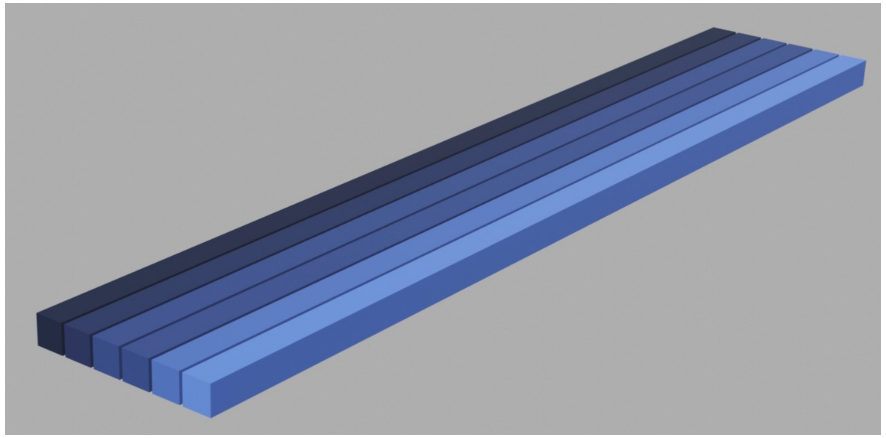




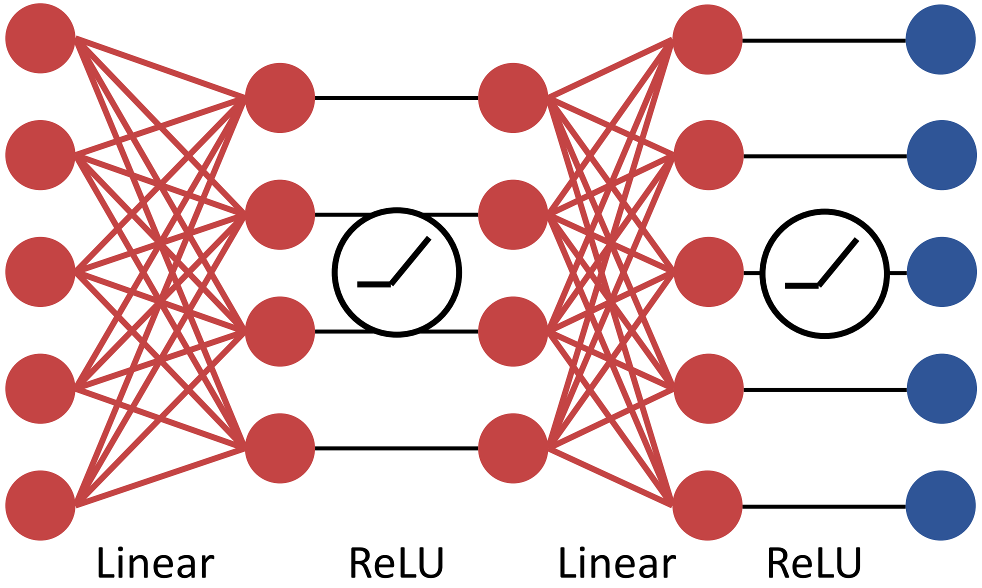
2. Neural network & dataset

- GNN structure (models/message_pass.py)



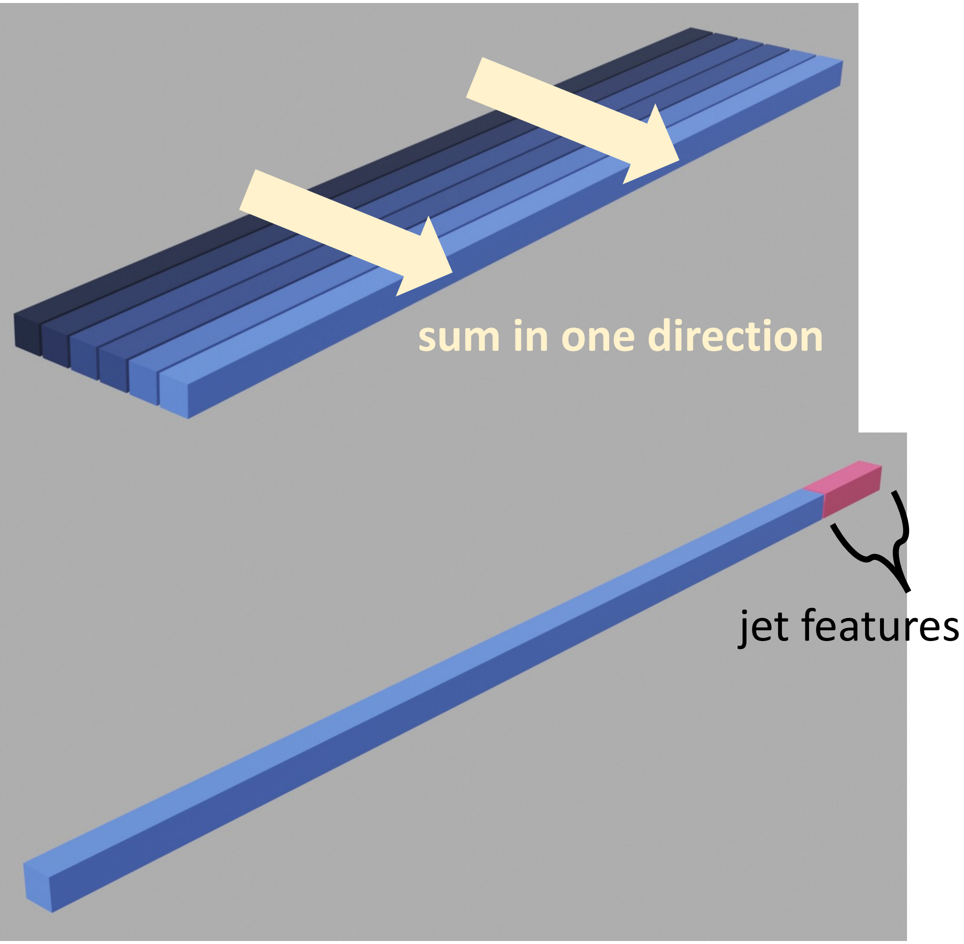


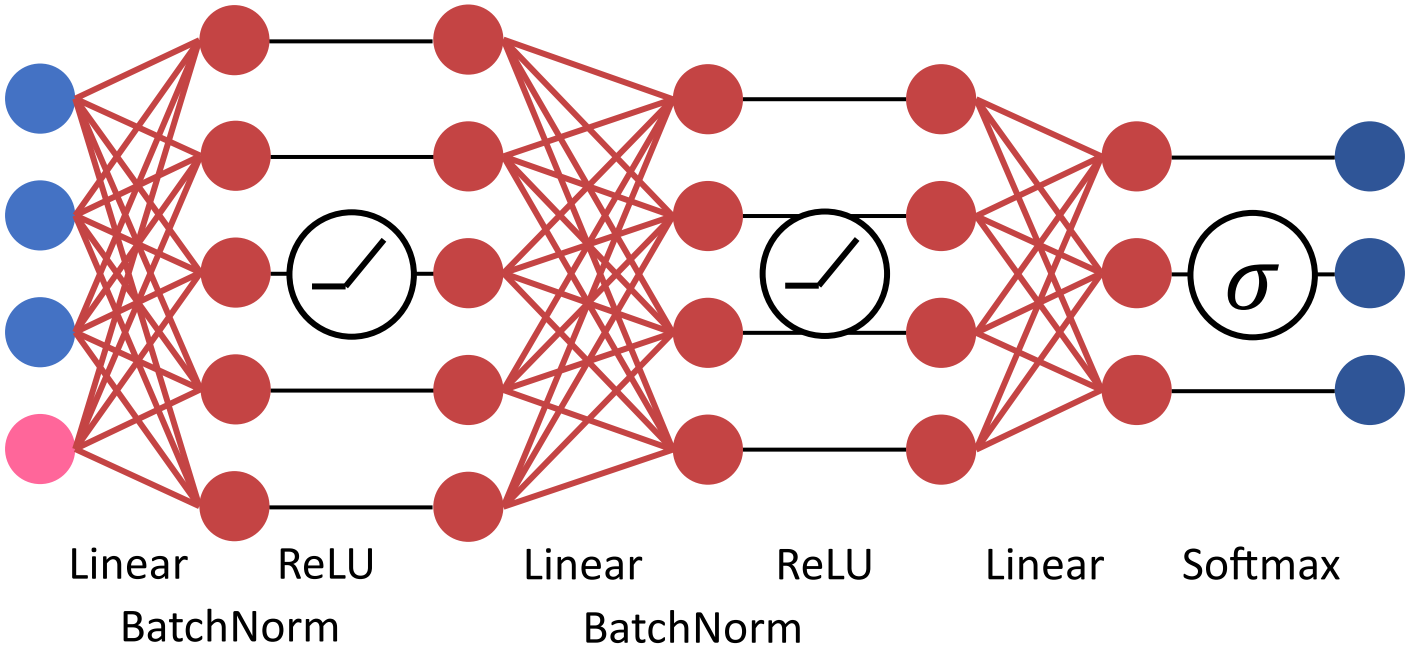
node network



2. Neural network & dataset

- GNN structure (models/message_pass.py)





softmax → argmax → jet flavor!

2. Neural network & dataset

- Training procedure

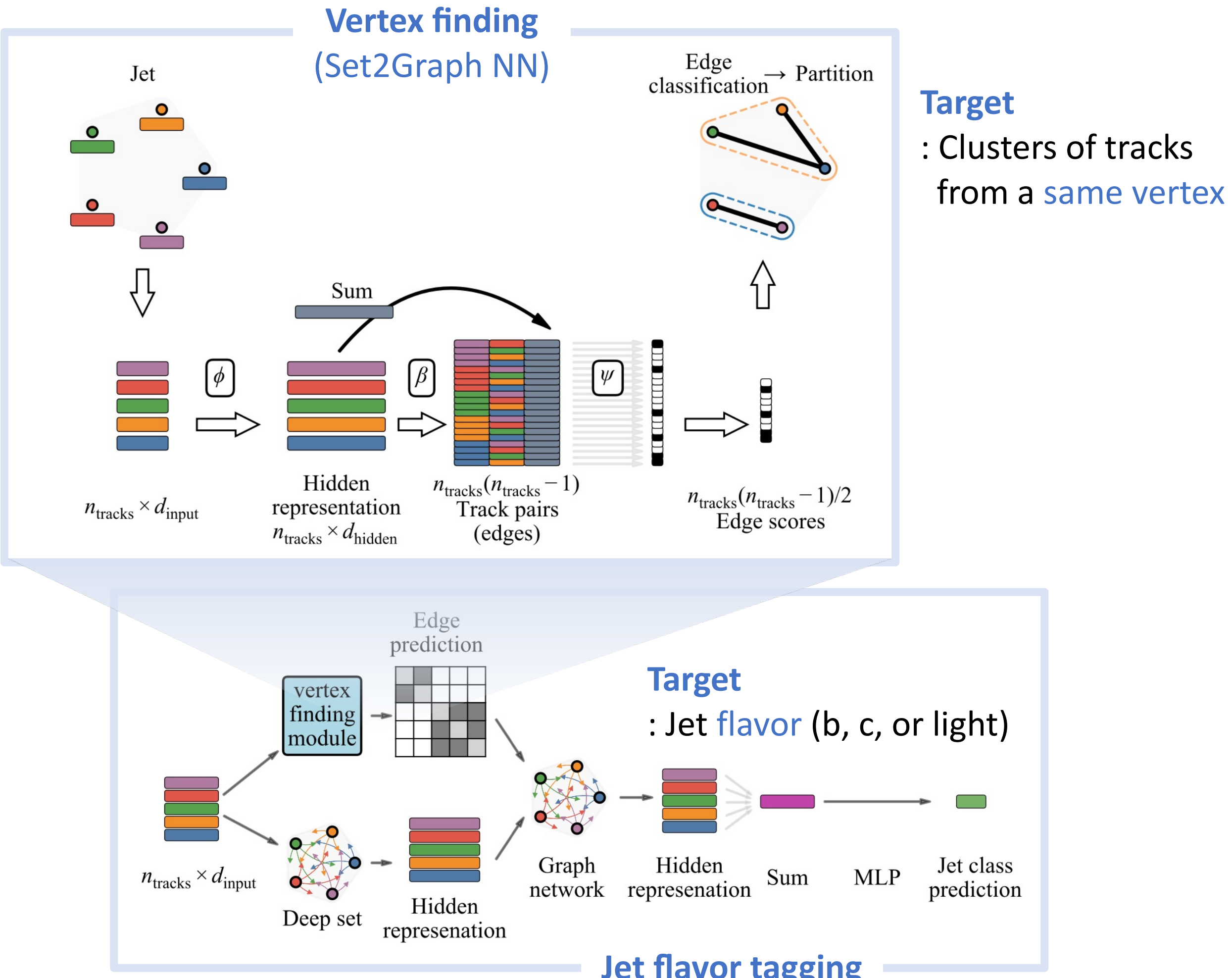
(1) Training (supervised learning) the vertex finding module with MC truth vertex information.

Batch size: 2048
Optimizer: Adam ($\mathrm{lr} = 10^{-3}$)
Loss function: BCE and F_β^*
Early stopping: 20 epochs

(2) Training jet flavor tagging neural networks (including trained vertex finding module inside).

Batch size: 1000
Optimizer: Adam ($\mathrm{lr} = 5\times10^{-4}$)
Loss function: Cross entropy
Early stopping: 20 epochs

* It is also possible to omit procedure (1).



J. Shlomi et al., op. cit.

2. Neural network & dataset

- Dataset specification

ALICE Run2 MC data

- PYTHIA pp collision, $\sqrt{s} = 5.02\ \mathrm{TeV}$, $b\bar{b}$ (LHC18k6a), c$\bar{c}$ (LHC18k6b), jet-jet (LHC18b8) events
- ALICE (Run2) full simulation

Jets

- Anti-k_{T} $(R = 0.4)$, charged particle jets
- $10 < p_{\mathrm{T,\,jet}} < 100\ \mathrm{GeV}/c$
- $|\eta_{\mathrm{jet}}| < 0.5$
- $2 \leq n_{\mathrm{tracks}} (\leq 16)$

Dataset size

- Training $500\ \mathrm{k\ jets}$, validation $100\ \mathrm{k\ jets}$, test $100\ \mathrm{k\ jets}$ (smaller dataset will be used for hands-on.)
- Dataset contains almost same numbers of b/c/light jets.

2. Neural network & dataset

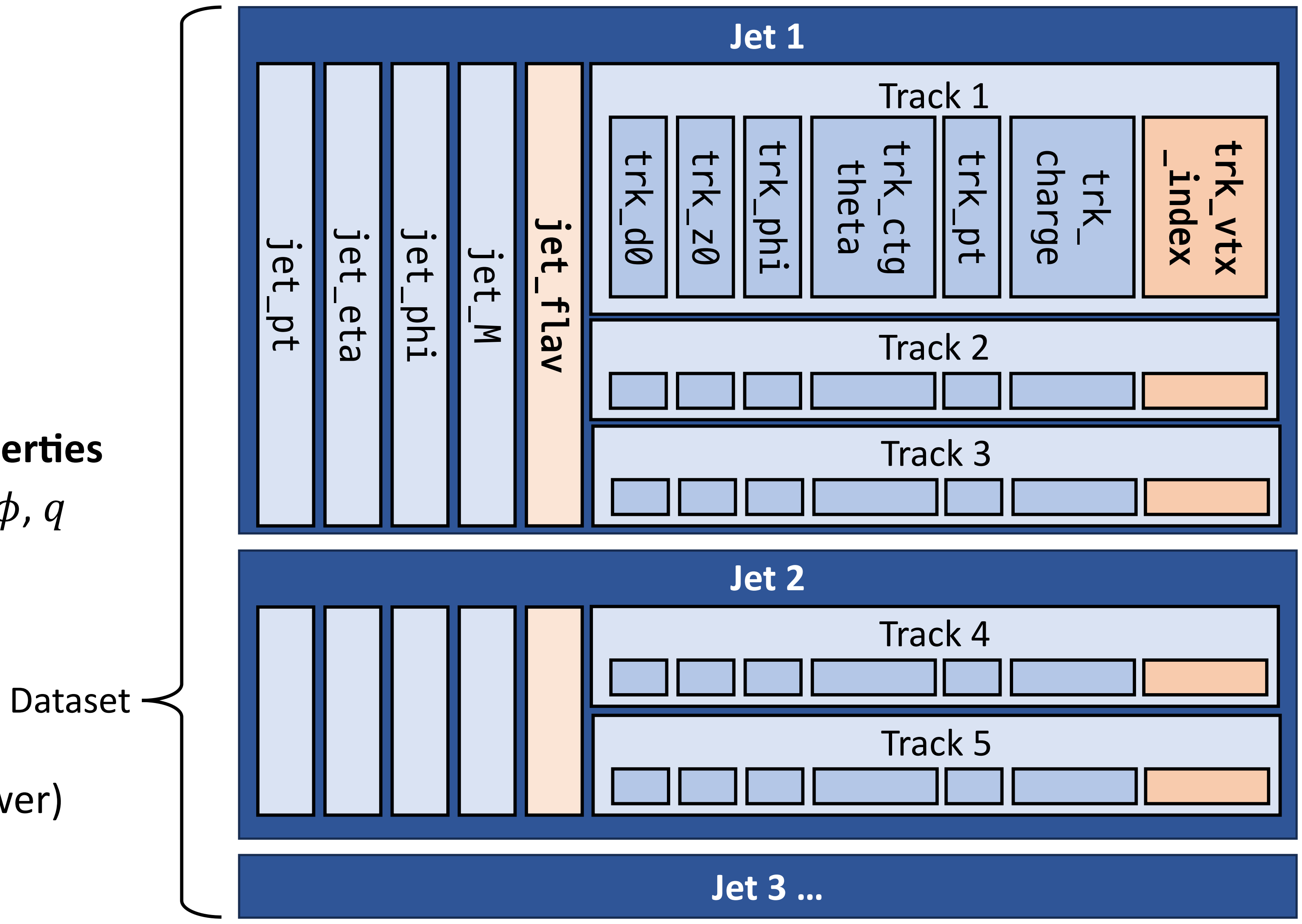
- Input properties

Jet properties

: $p_{\mathrm{T,\,jet}}, \eta_{\mathrm{jet}}, \phi_{\mathrm{jet}}, m_{\mathrm{jet}}$

(reconstructed) **Track properties**

: $\mathrm{DCA}_{xy}, \mathrm{DCA}_{z}, p_{\mathrm{T}}, \cot\theta, \phi, q$



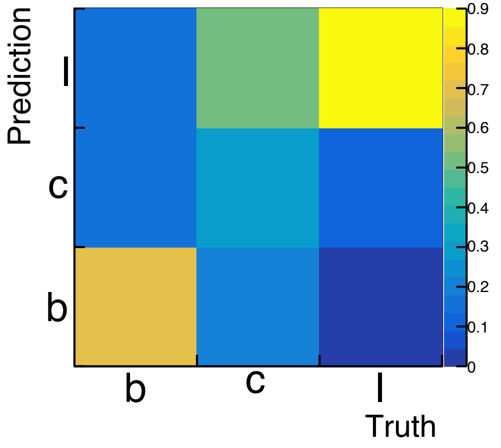
■ Input data

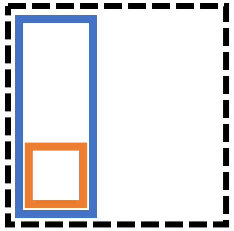
■ Label data (=correct answer)

3. Training result

3. Training result

- Performance metrics

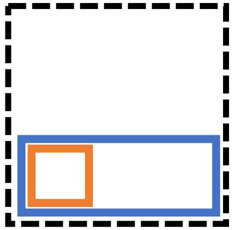




- $\text{Efficiency}(x) = \dfrac{(\text{truth} = x \wedge \text{pred} = x)}{(\text{truth} = x)}$

→ How many truth x are found?

(Independent to the numbers of truth b/c/light jets)

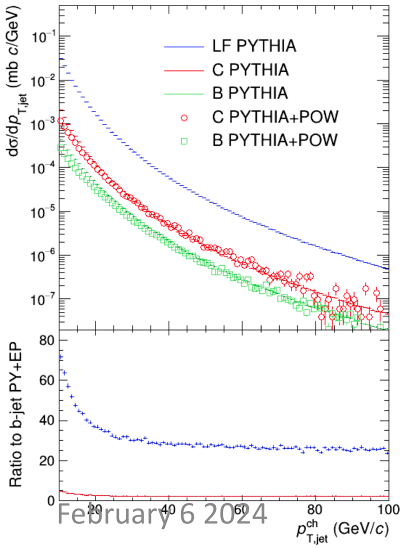


- $\text{Purity}(x) = \dfrac{(\text{truth} = x \wedge \text{pred} = x)}{(\text{pred} = x)} = \dfrac{\varepsilon_b N_b}{\varepsilon_b N_b + \varepsilon_{c\to b} N_c + \varepsilon_{l\to b} N_l}$

(ε_b: b-jet efficiency, $\varepsilon_{x\to b}$: fraction of mis-tagged truth x-jets among b-jet candidates)

→ How many of predicted x are true?

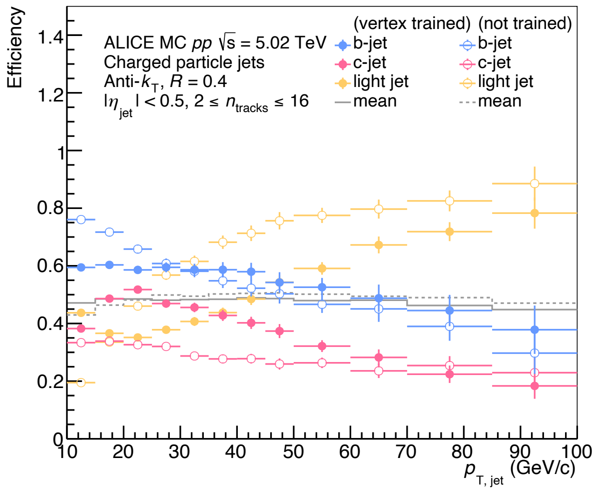
(Dependent to the numbers of truth b/c/light jets)

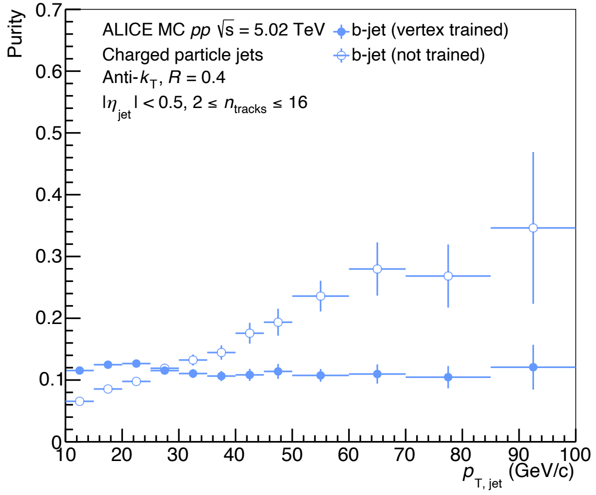


*Purity is calculated on the assumption that jet cross-section ratio as constant b : c : l = 1 : 2 : 27. (in temporary)

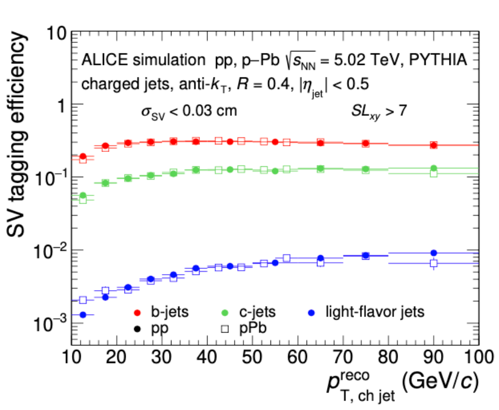
3. Training result

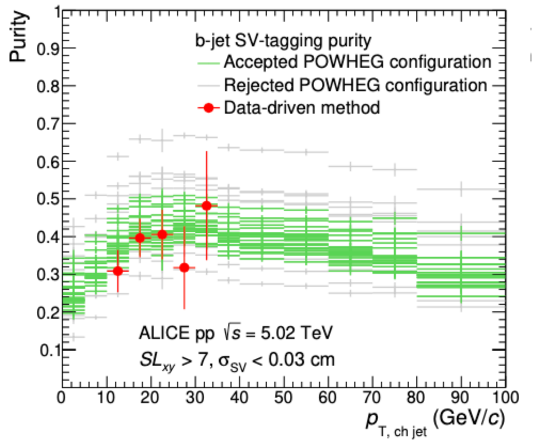
- Performance





SV method (ALICE)





- B-jet efficiency is higher, purity is lower than previous SV method.

- Efficiency and purity are complementary, so both should be considered at the same time.

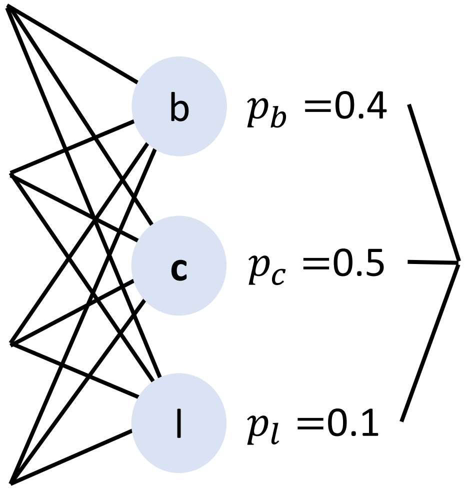
→ The optimization of working point is needed to get higher purity b-jet result.

ALICE Collaboration, op.cit.

4. B-jet selection optimization

4. B-jet selection optimization

- B-jet tagging discriminant

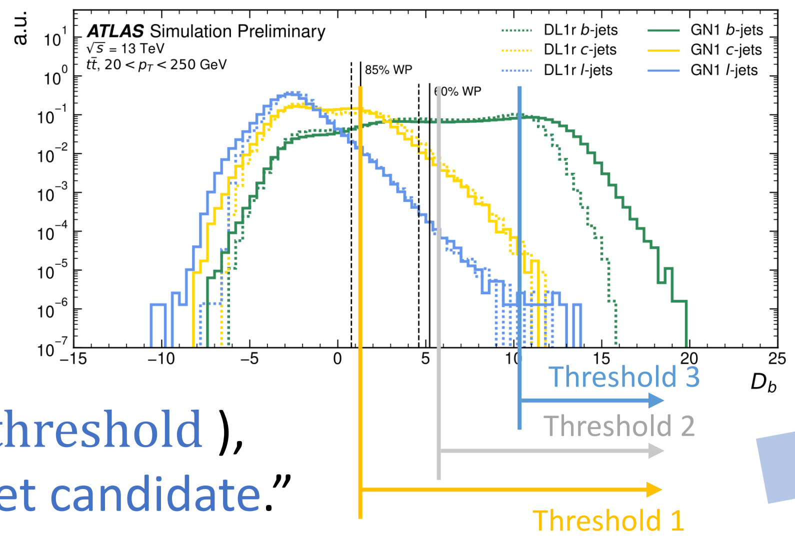


(softmax outputs)

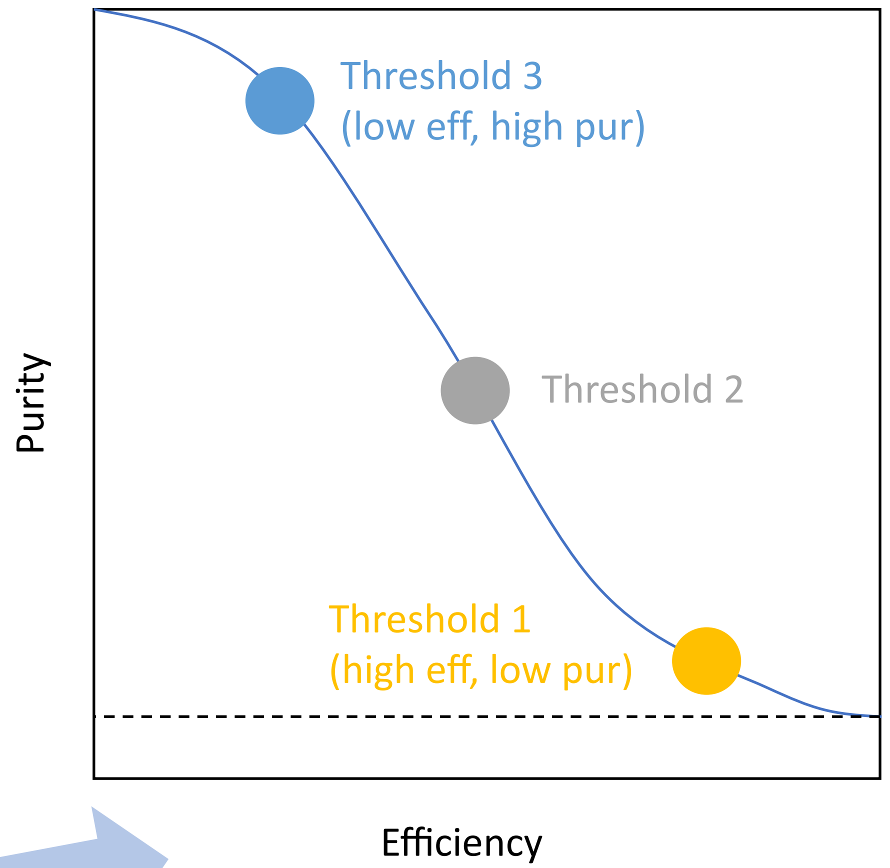
$$D_b = \log \frac{p_b}{(1 - f_c)p_l + f_c p_c}$$

(f_c = 0.018, optimized parameter)

The ATLAS Collaboration, *ATL-PHYS-PUB-2022-027*



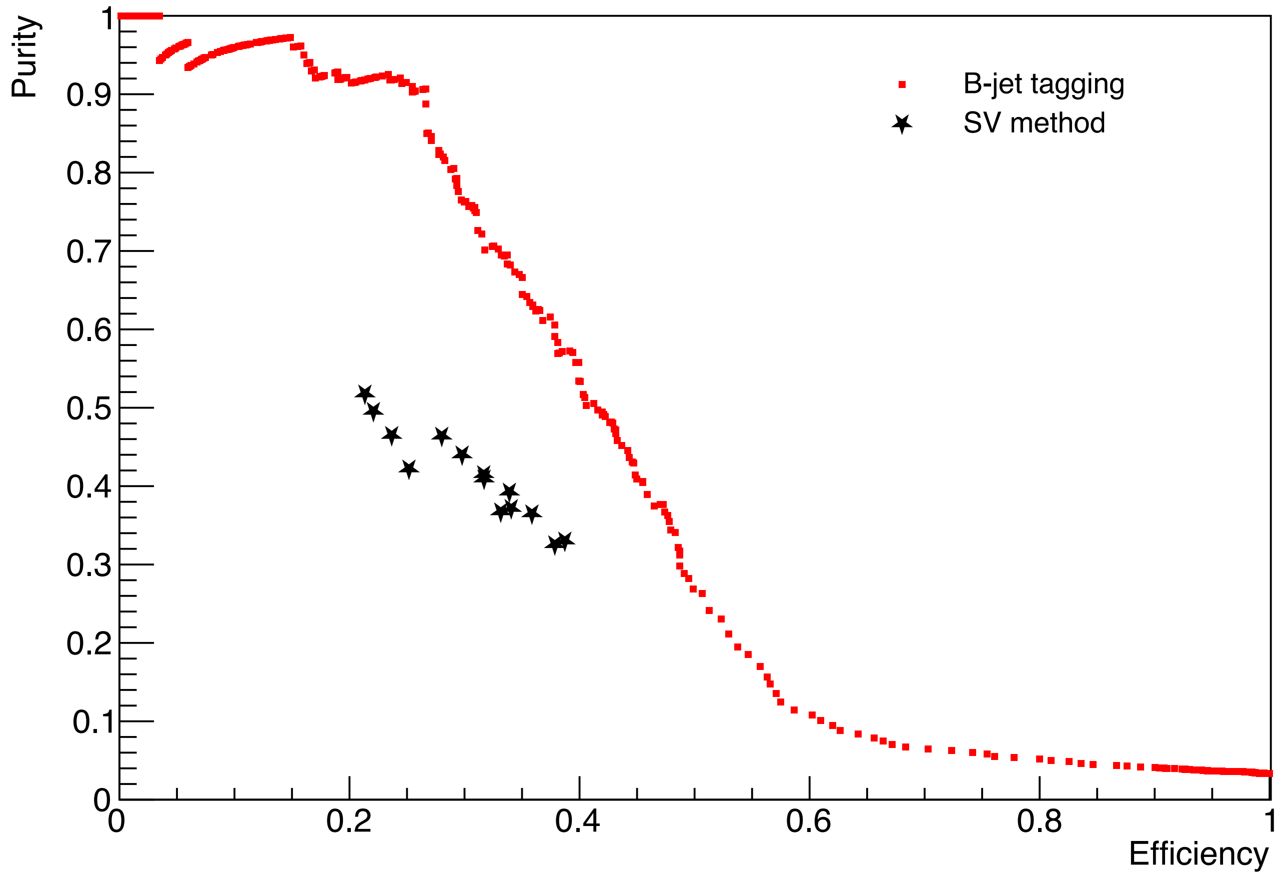
"if (D_b > threshold),
then it is a b-jet candidate."

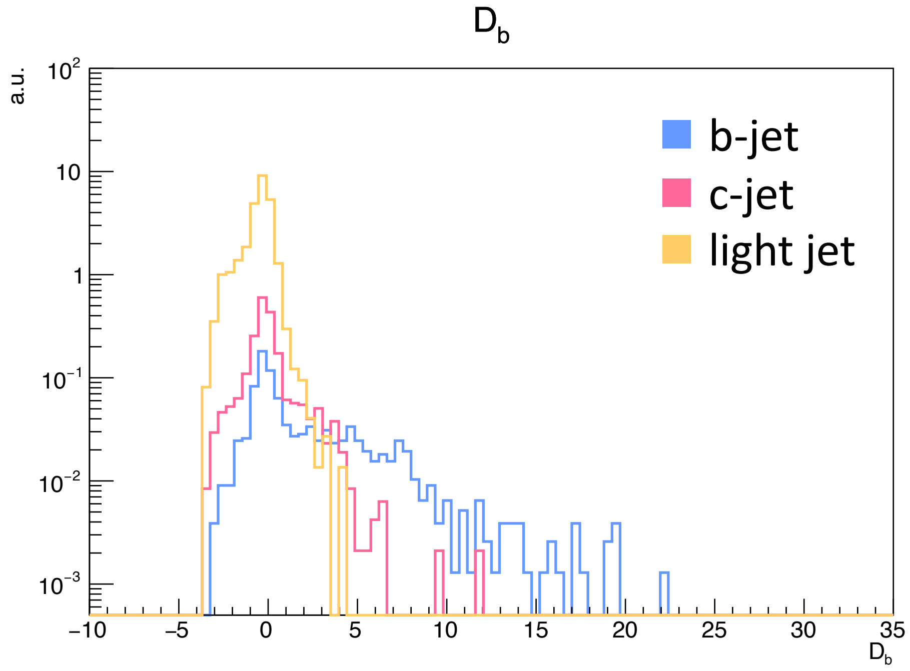


4. B-jet selection optimization

- Efficiency and purity

(for the range $p_{\mathrm{T,jet}} = 50\sim60\ \mathrm{GeV}/c$)



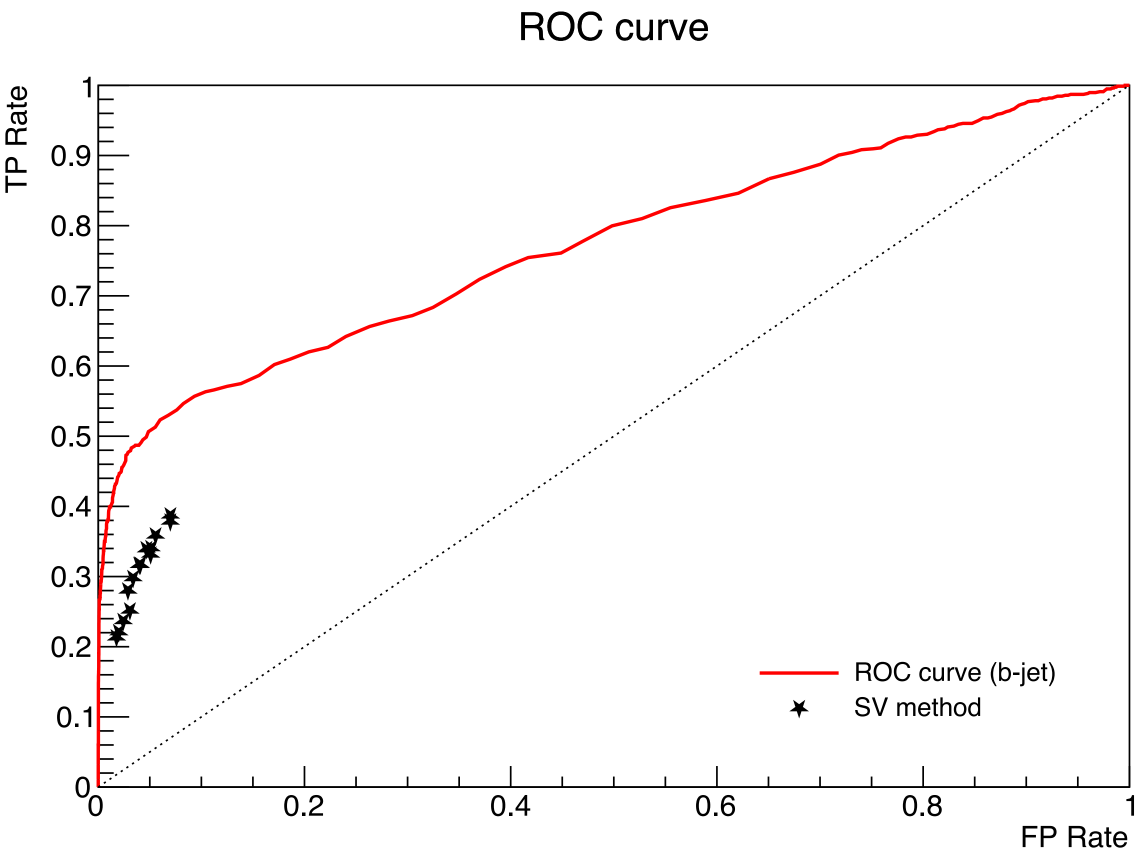


- Both efficiency and purity are higher than SV method at high purity region.

4. B-jet selection optimization

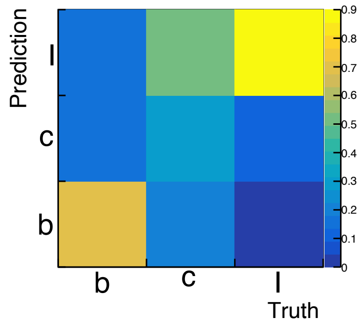
- ROC (Receiver Operating Characteristic) curve

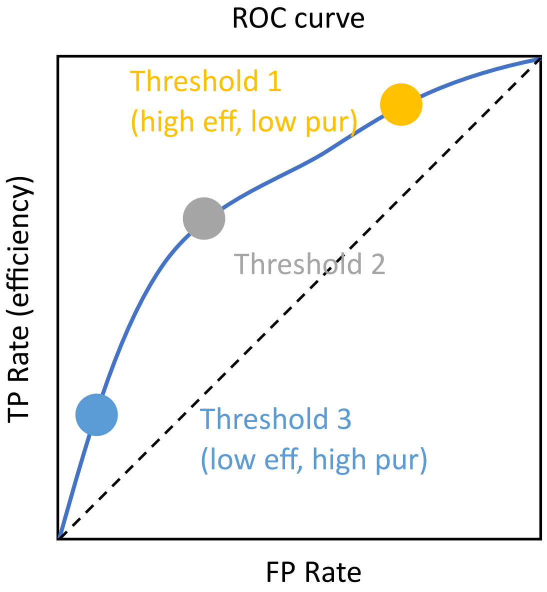
(for the range $p_{\mathrm{T,\,jet}} = 50{\sim}60\ \mathrm{GeV}/c$)

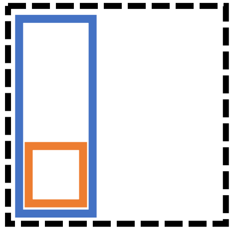


- AUC (Area Under Curve)

: 0.773

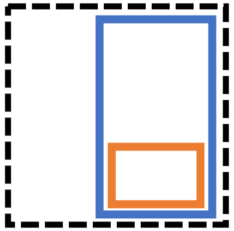






TP rate

$$= \frac{(\mathrm{truth} = x \wedge \mathrm{pred} = x)}{(\mathrm{truth} = x)}$$



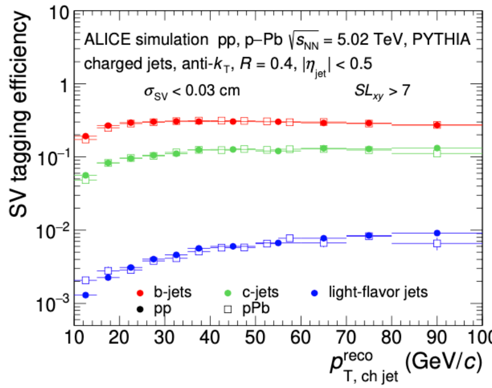
FP rate

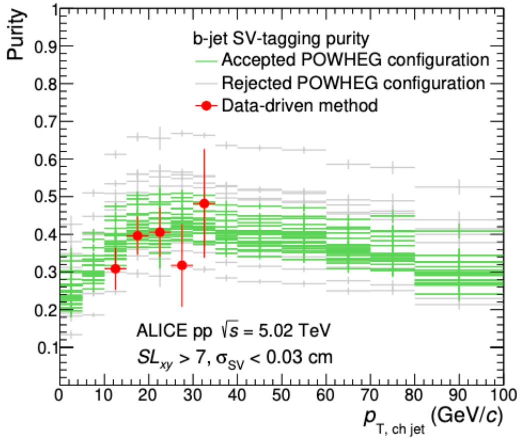
$$= \frac{(\mathrm{truth} \neq x \wedge \mathrm{pred} = x)}{(\mathrm{truth} \neq x)}$$

4. B-jet selection optimization

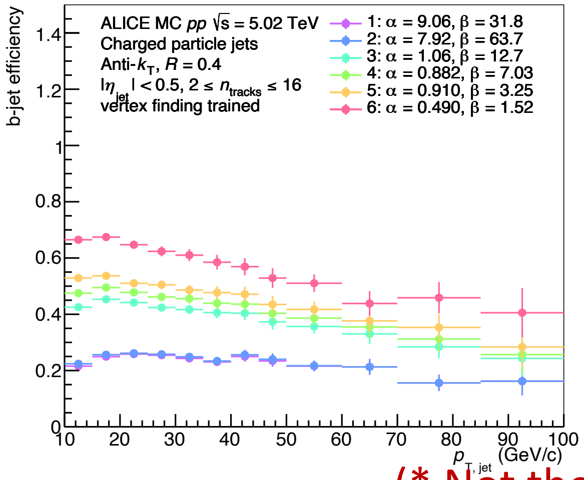
- High purity working points

SV method (ALICE)

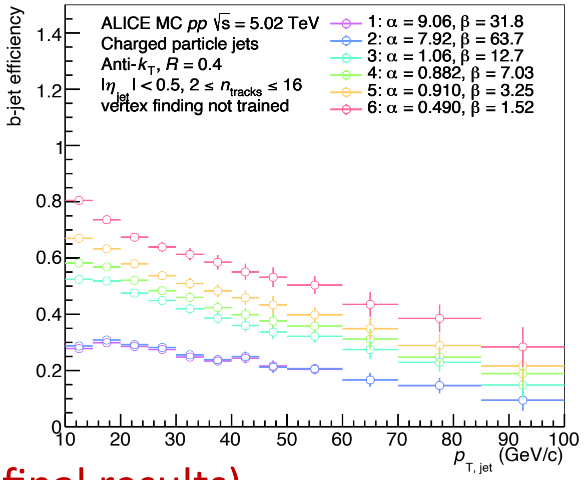




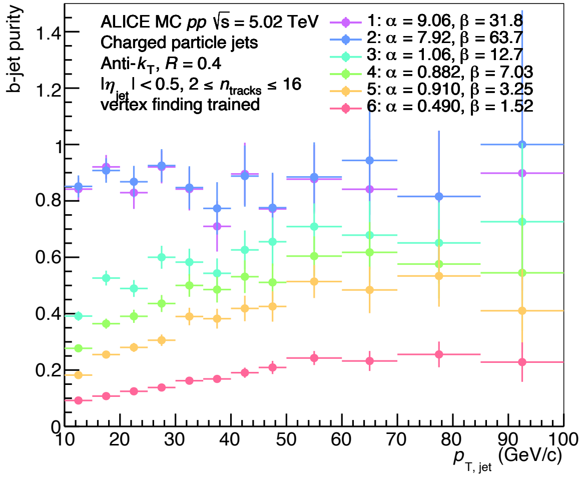
Vertex finding trained

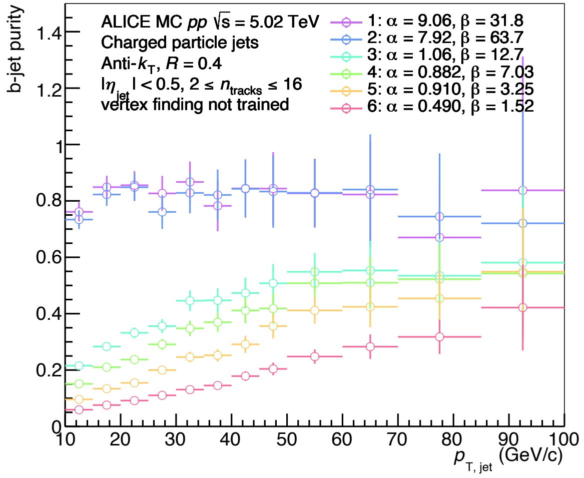


Vertex finding not trained



(* Not the final results)





Thank you

References

[1] J. Shlomi et al, *Eur.Phys.J.C* (2021) 81:540.

[2] ALICE Collaboration, *JHEP* 01 (2022) 178.

[3] The ATLAS Collaboration, ATL-PHYS-PUB-2022-027.