LHCb Jet Flavour Classification with Graph Neural Networks

Gabriella Pesticci

July 25, 2024

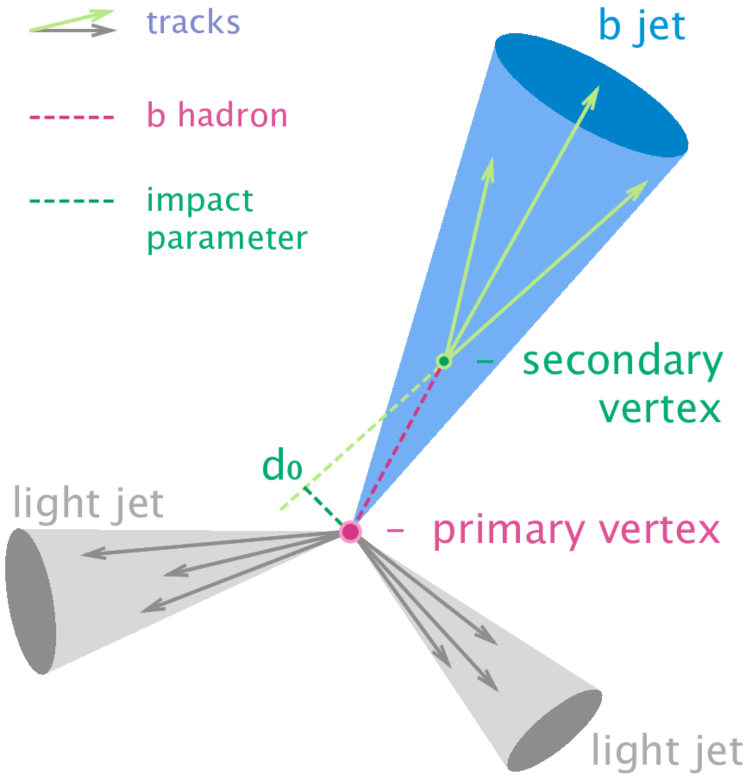
Motivation

Jet Hadronization

- p-p collisions at LHC produce quarks and gluons which hadronize due to QCD confinement
- b-quarks are heavy quarks with longer lifetimes, so there is a characteristic delay before hadronization
- We need to reconstruct the event and identify the original particle

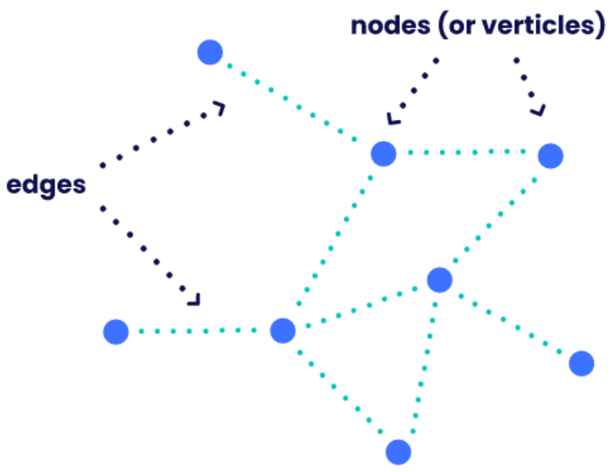
Goal:

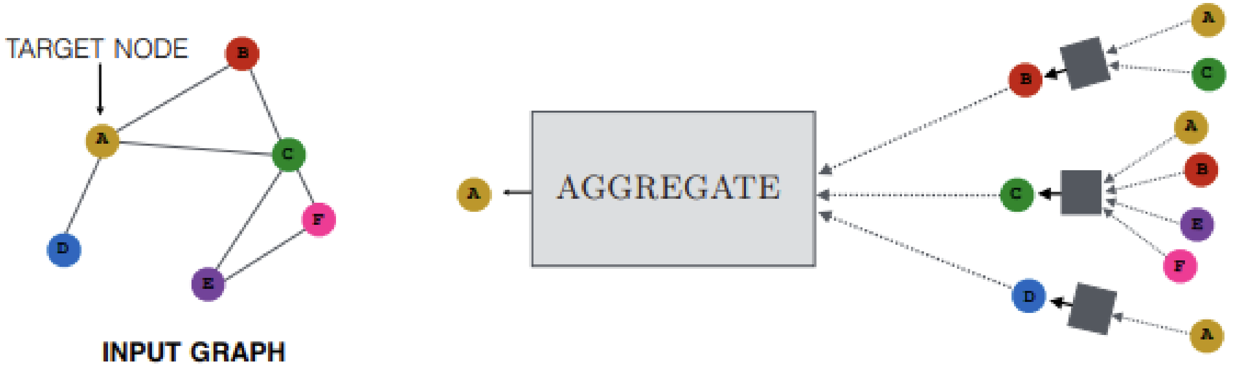
- Binary classification: identify if jets are produced by b-quarks or not
- Utilize deep learning to accurately classify the jets in our data
- Apply this to classifier to identify c-jets and fat-jets



Deep Learning with Graph Neural Networks

- **Deep learning**
  - Multiple hidden layers
- **GNN**
  - Represent data as graphical structures
  - Able to handle complex datasets and capture relationships
  - Message passing allows for information to be passed between nodes

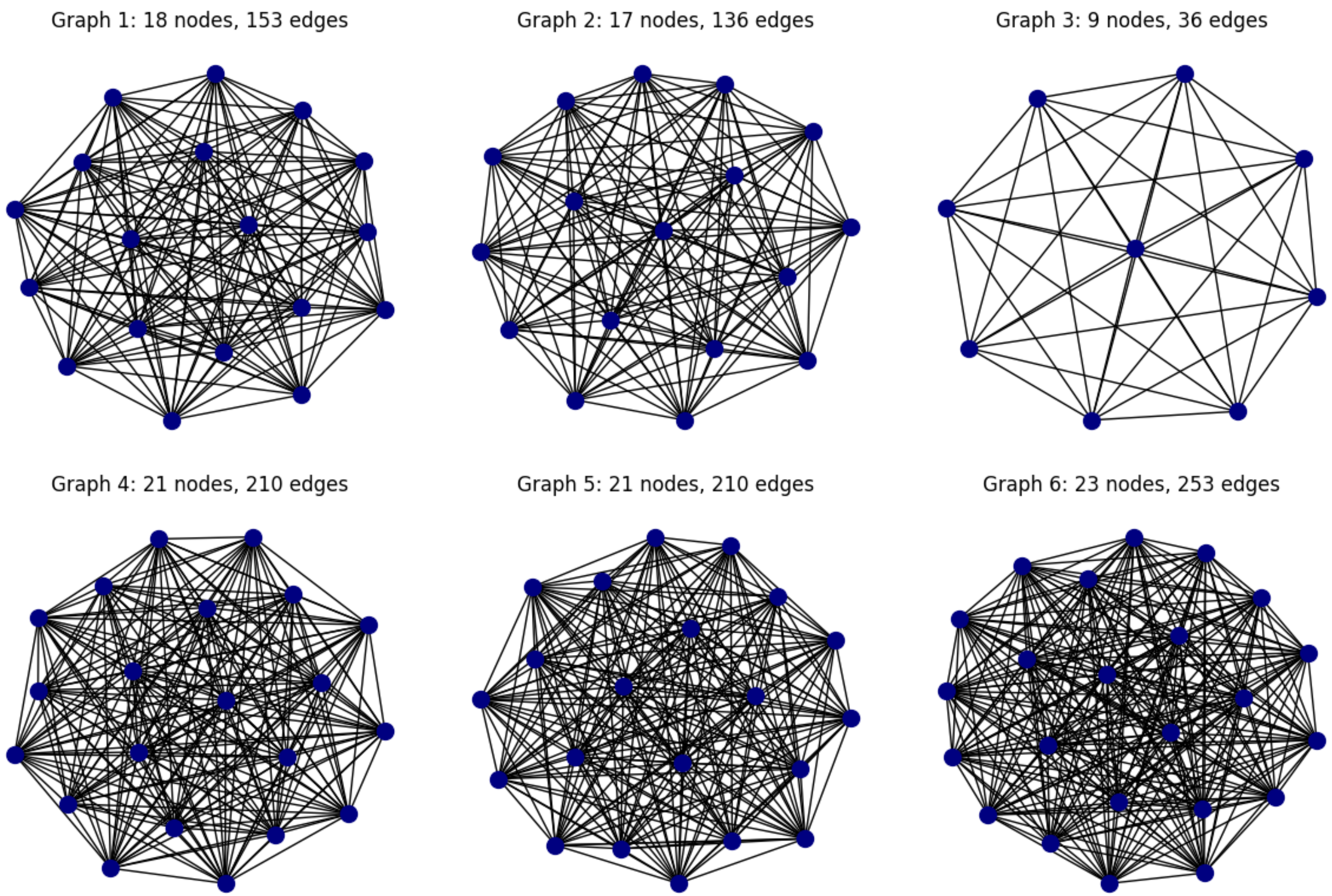




Graphical Representations of Data

Dijet Sample:

- Leading jet
- Each graph represents one jet
- Each node = one daughter
- Fully connected edges
- Features: jet-level and daughter-level
- Assign truth labels for each jet after truth matching and applying cuts to jet features



Data Preparation

Truth Matching

- For all data:
  - MC Match = 1
- Signal data:
  - MC Jet EfB > 0.6
- Background data:
  - MC Jet EfB < 0.6
  - MC Jet EfD < 0.6

Jet Selection

- pT > 20 GeV
- 2.2 < η < 4.4

Jet Features

- Describes overall jet kinematics
- Shared by all daughters in the jet

Daughter Features

- Unique kinematics for each daughter in the jet

SV Features

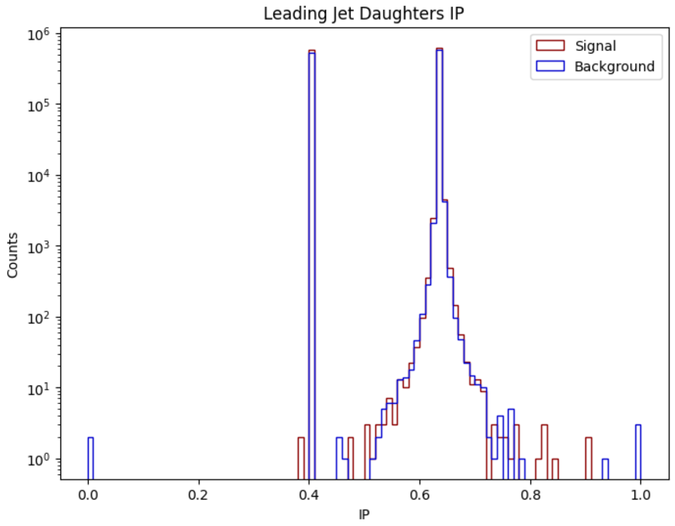
- SV tagging
- Jet-level features

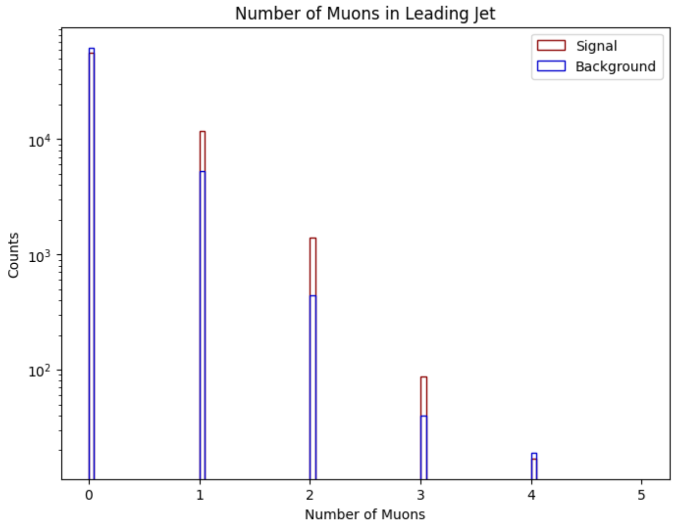
Jet Features
η
p_T

Daughter Features
E
p_T
ID
p (x,y,z)
η
ϕ
Q
IP
IP χ^2
IP raw
NNe
NNk
NNp
NNπ
NNμ
χ^2
Q/P
Track (x, y, z)
TrackV (x, y, z)
CaloNeutralEcal
CaloNeutralHcal2Ecal
CaloNeutralE49
CaloNeutralPrs

SV Features
fdrMin ptSvrJet
nTrk
nTrkJet
drSvrJet
absQSum
m
mCor
fdChi2
ipChi2Sum
bdt0
bdt1
pass
tau
z
pt
backwards

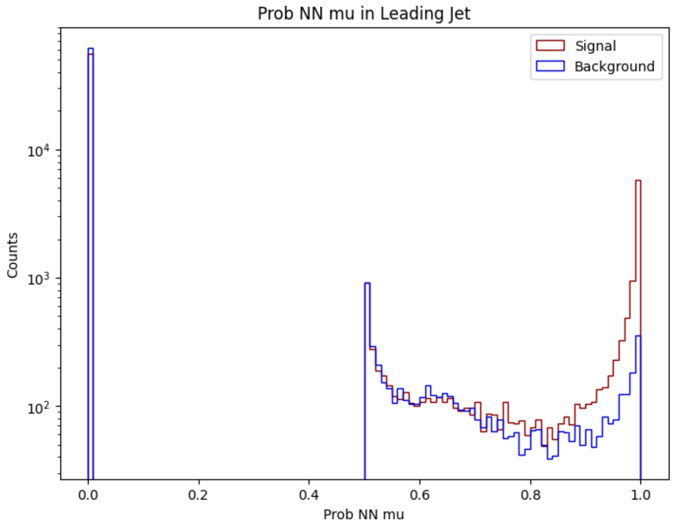
Graph Features

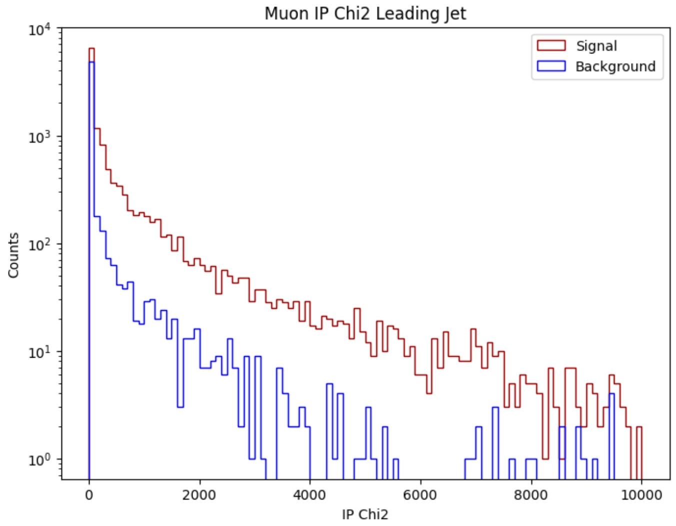


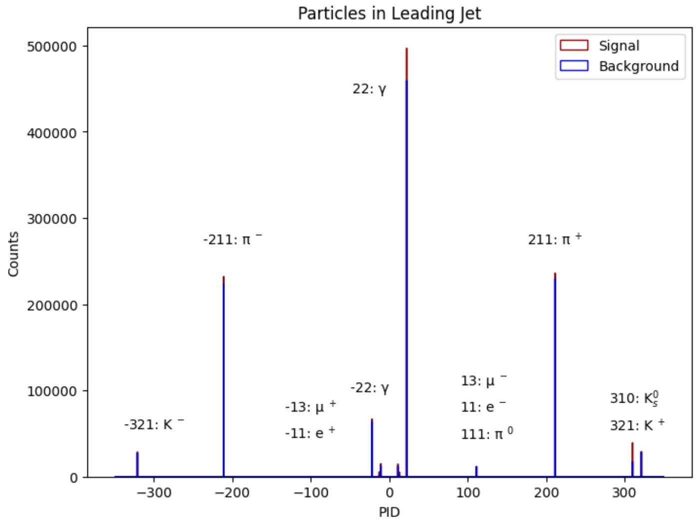


Particles Present in Leading Jet

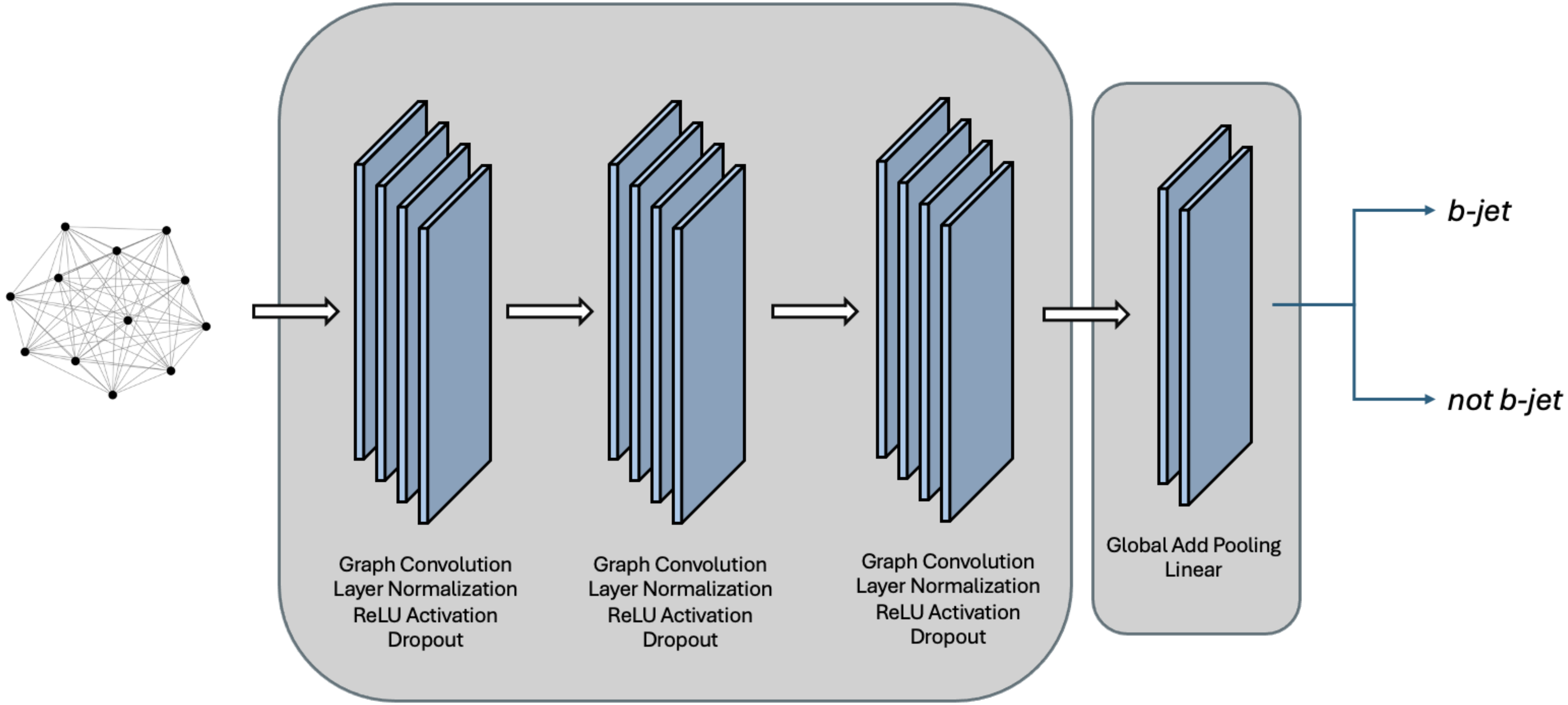
Particle ID	Signal Counts	Background Counts
-3122: Λ	4861	2501
-2212: p^-	9640	16193
-321: K^-	28599	27363
-211: π^-	232082	223814
-22: γ	66737	63545
-13: μ^+	5456	1114
-11: e^+	14963	12551
11: e^-	14550	12104
13: μ^-	5055	945
22: γ	496754	459141
111: π^0	11397	11790
211: π^+	235952	229267
310: K_s^0	39285	17327
321: K^+	28977	28592
2212: p^+	10255	17210
3122: Λ	4722	2409







GNN Architecture



GNN Layers Functions

SAGEConv

- Aggregates information from neighbors - mean

$$\boldsymbol{x}_i' = \boldsymbol{W}_1 \boldsymbol{x}_1 + \boldsymbol{W}_2 \cdot \mathrm{mean}_{j \in \mathcal{N}(i)} \boldsymbol{x}_j$$

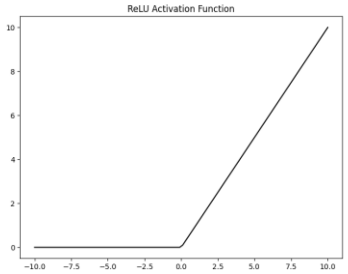
LayerNorm

- Normalize inputs across all features independently

$$y = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} * \gamma + \beta$$

ReLU

$$R(z) = \max(0, z)$$



Dropout

- Zero elements with probability, *p*
- Scale by factor of $\frac{1}{1-p}$

Global Add Pooling

$$\boldsymbol{r}_i = \sum_{n=1}^{N_i} \boldsymbol{x}_n$$

Linear

$$y = xA^T + b$$

Loss and Optimization

Binary Cross Entropy Loss

$$\ell(x,y) = L = \{l_1, \dots, l_N\}^\top, l_n$$
$$= -w_n[y_n \cdot \log \sigma(x_n) + (1 - y_n) \cdot \log(1 - \sigma(x_n))]$$

AdamW

- Stochastic gradient descent method
- Separates weight decay from gradients
- Better regularization than Adam

---

input : γ(lr), β_1, β_2(betas), θ_0(params), $f(\theta)$(objective), ϵ (epsilon)
λ(weight decay), *amsgrad*, *maximize*
initialize : $m_0 \leftarrow 0$ (first moment), $v_0 \leftarrow 0$ (second moment), $\widehat{v_0}^{max} \leftarrow 0$

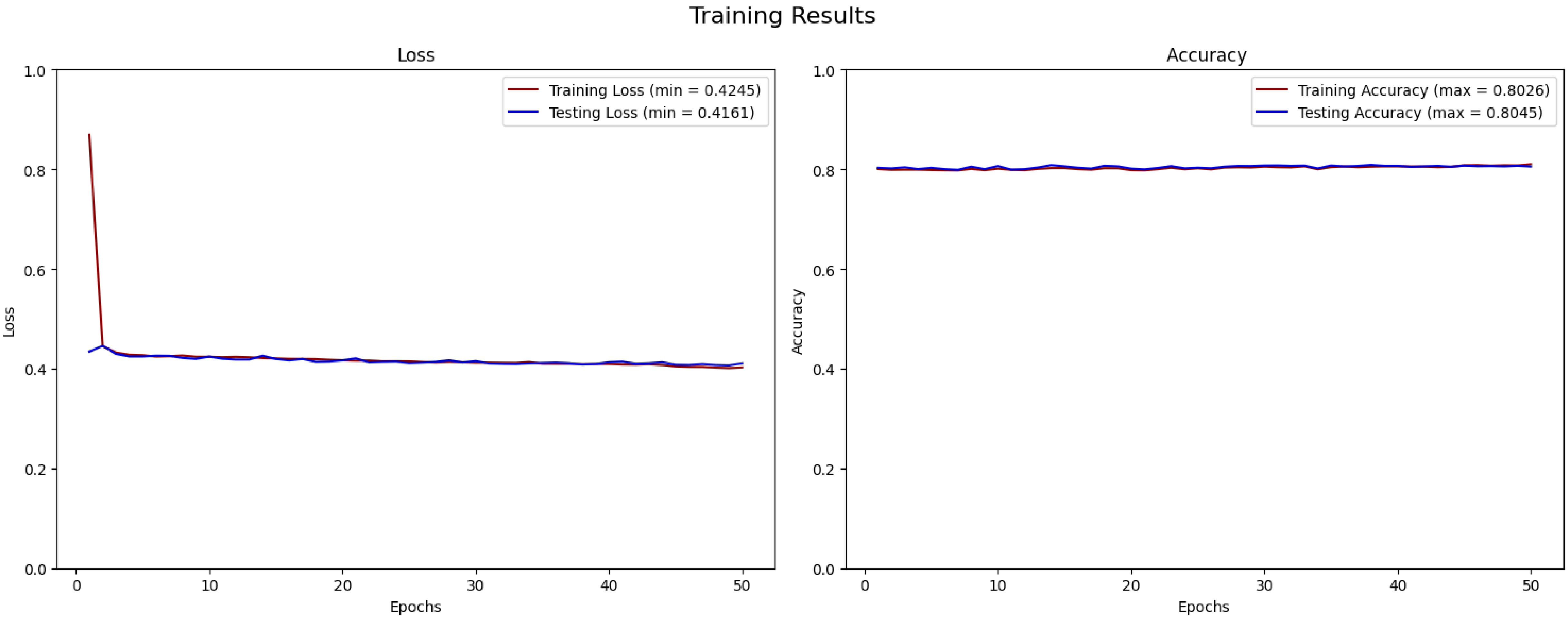
---

for $t = 1$ **to** ... **do**
  if *maximize* :
    $g_t \leftarrow -\nabla_\theta f_t(\theta_{t-1})$
  else
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
  $\theta_t \leftarrow \theta_{t-1} - \gamma\lambda\theta_{t-1}$
  $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$
  $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
  $\widehat{m_t} \leftarrow m_t / \big(1 - \beta_1^t\big)$
  $\widehat{v_t} \leftarrow v_t / \big(1 - \beta_2^t\big)$
  if *amsgrad*
    $\widehat{v_t}^{max} \leftarrow \max(\widehat{v_t}^{max}, \widehat{v_t})$
    $\theta_t \leftarrow \theta_t - \gamma \widehat{m_t} / \big(\sqrt{\widehat{v_t}^{max}} + \epsilon\big)$
  else
    $\theta_t \leftarrow \theta_t - \gamma \widehat{m_t} / \big(\sqrt{\widehat{v_t}} + \epsilon\big)$

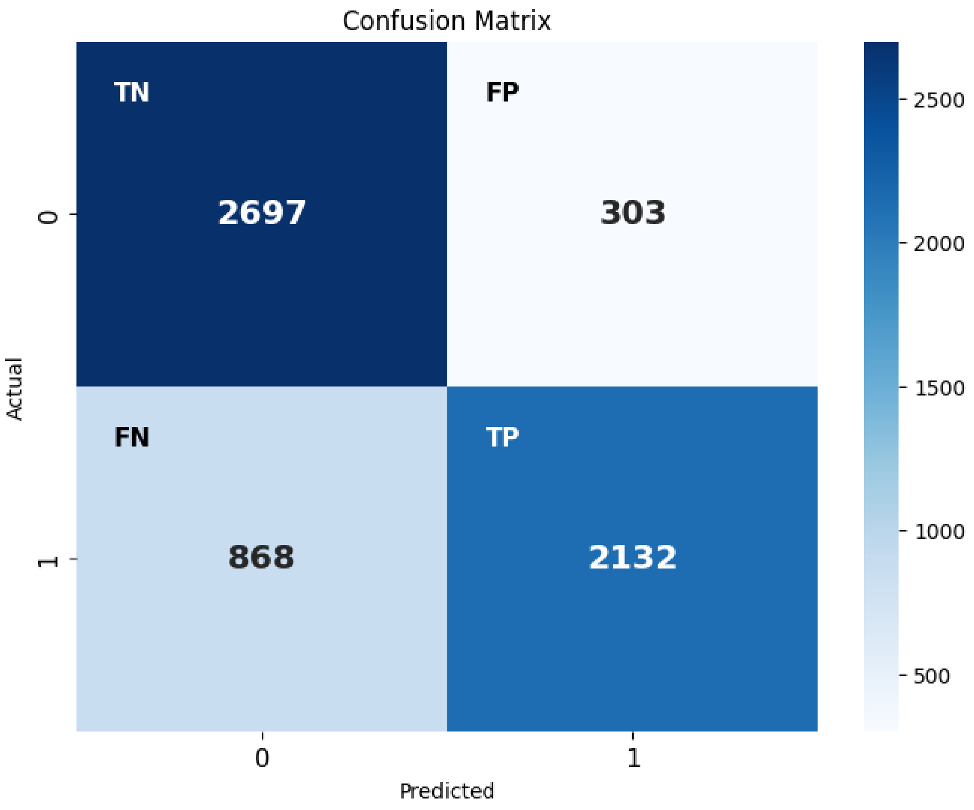
---

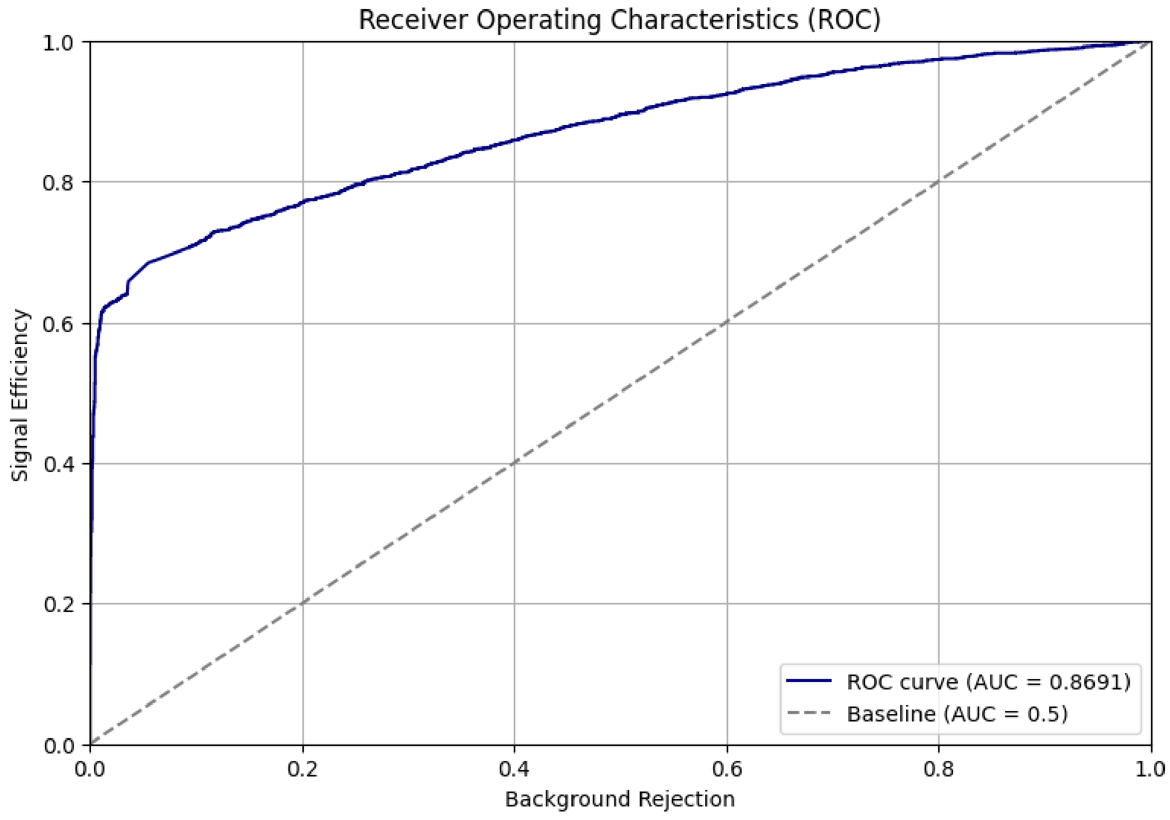
return θ_t

Results


Training Results
Loss
Training Loss (min = 0.4245)
Testing Loss (min = 0.4161)
Loss
Epochs
Accuracy
Training Accuracy (max = 0.8026)
Testing Accuracy (max = 0.8045)
Accuracy
Epochs
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
50

Results

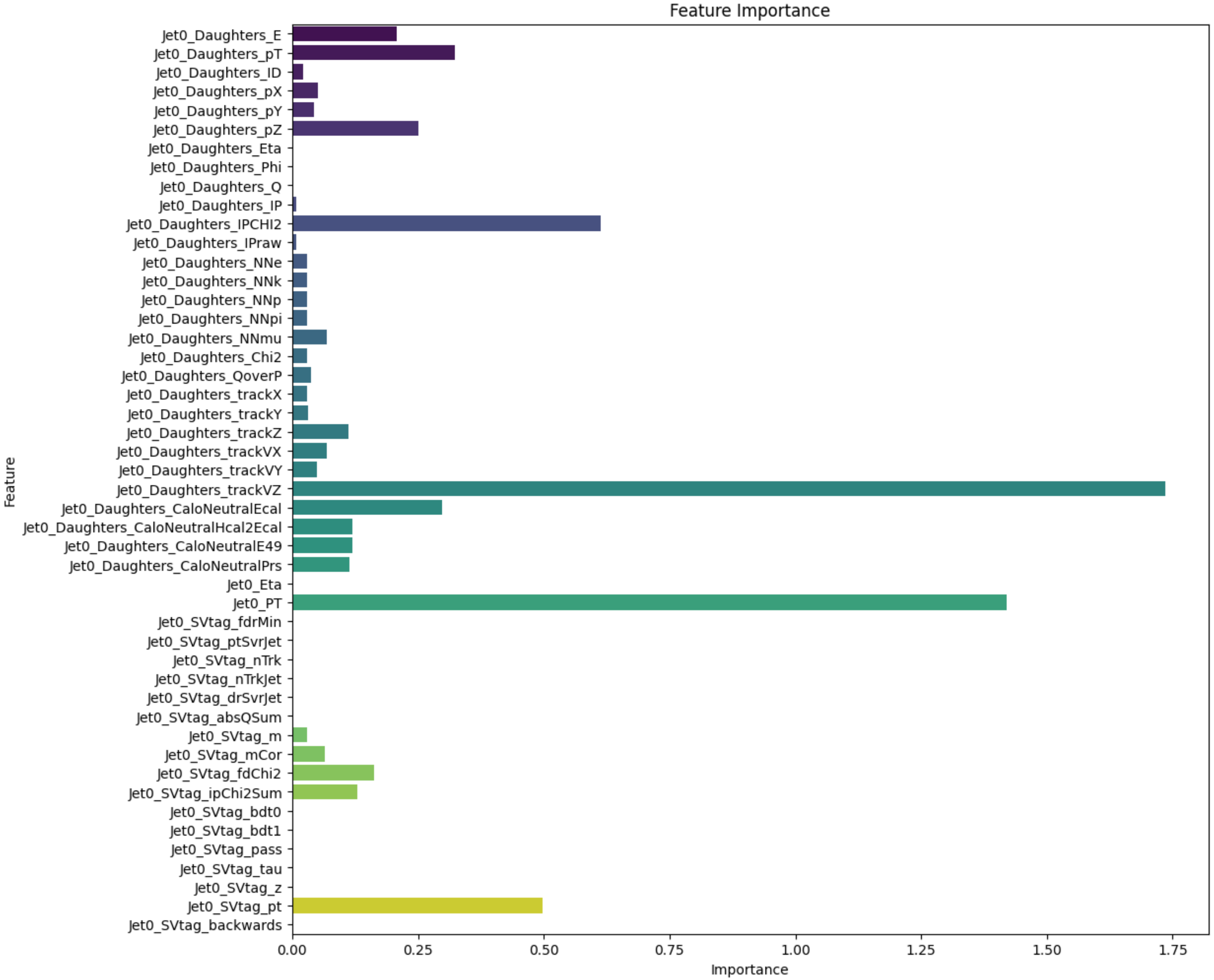

Confusion Matrix
TN
2697
FP
303
FN
868
TP
2132
0
1
Actual
0
1
Predicted
2500
2000
1500
1000
500


Receiver Operating Characteristics (ROC)
Signal Efficiency
Background Rejection
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (AUC = 0.8691)
Baseline (AUC = 0.5)

Results

Feature Ablation:

- Remove one feature at a time and compare predictions



Next steps:

- Hyperparameter tuning and optimization
- Complete documentation
- Apply for c-jet and fat jet classification