Jet Classification – A comparison Between BDT, CNN, Transformers

Group – 5

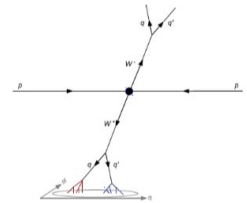
Jaideep Kalani, Arkodip Biswas, Ankan Roy, Abhishek MS, Rajesh Pramanik, Sarthak Tripathy

Guide - Saranya Samik Ghosh

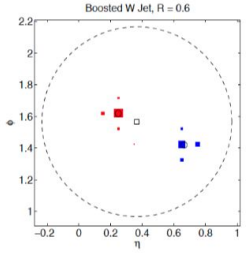
टाटा मूलभूत अनुसंधान संस्थान

Date - 11/07/2026

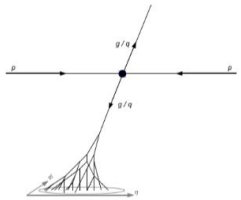
Jets As Images

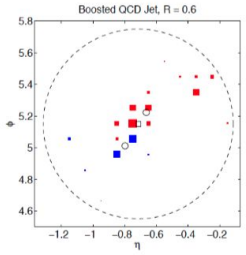


(a)

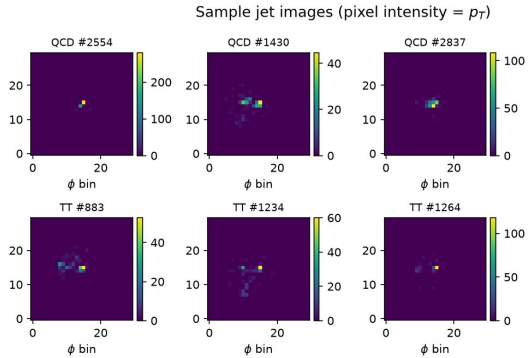


(b)

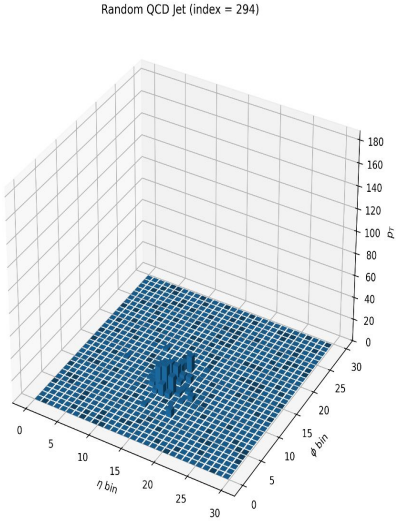


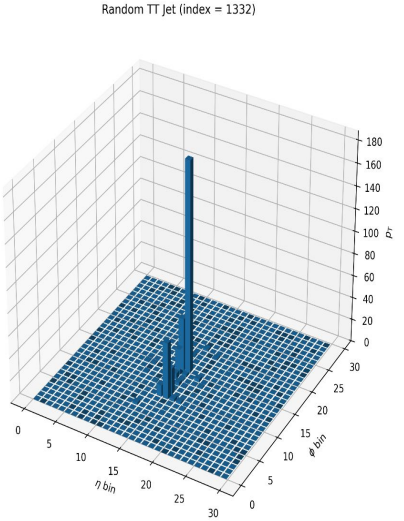


https://cs229.stanford.edu/proj2013/Anenberg-WvsQCDJetTaggingatLHC.pdf

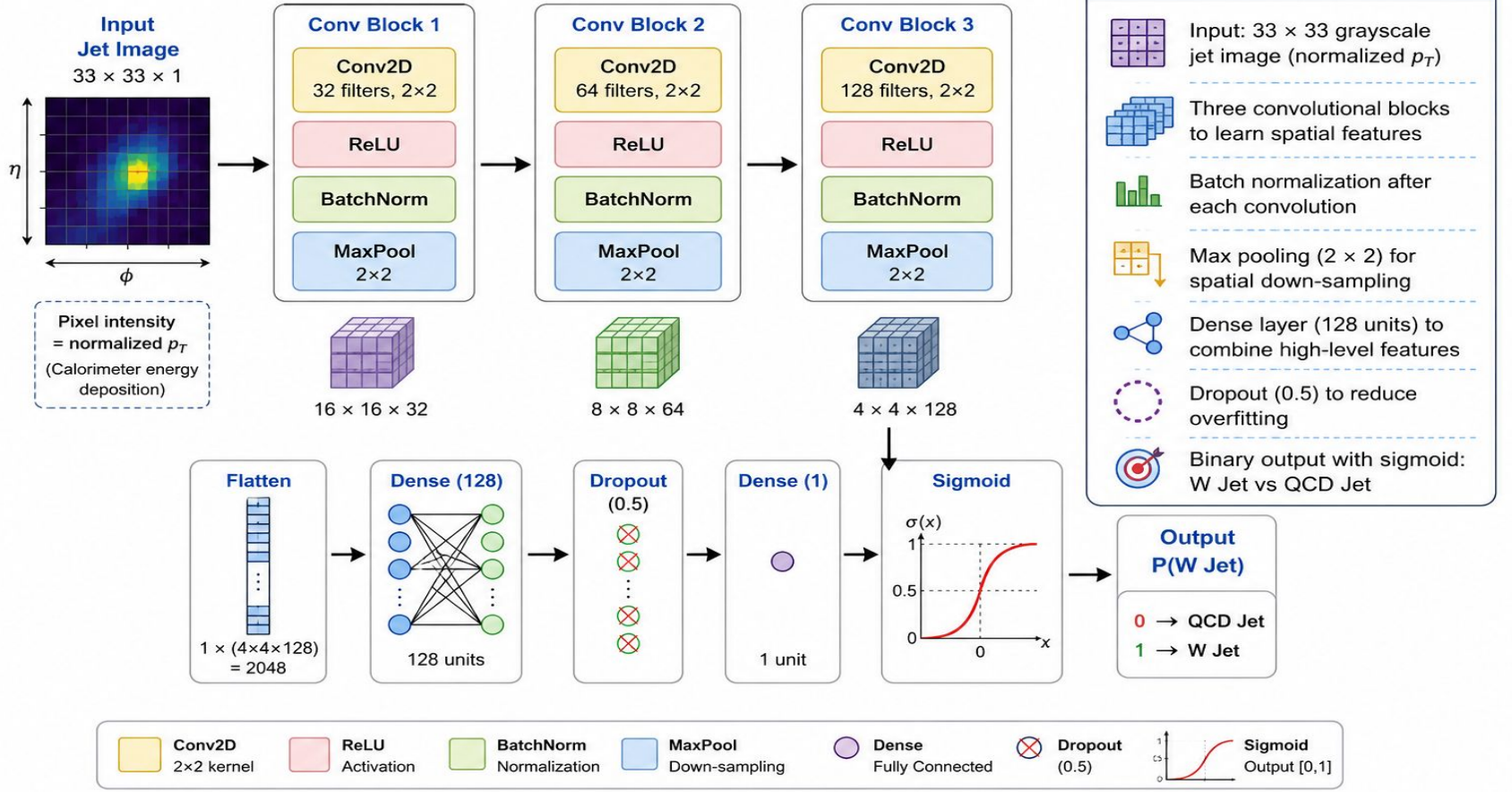


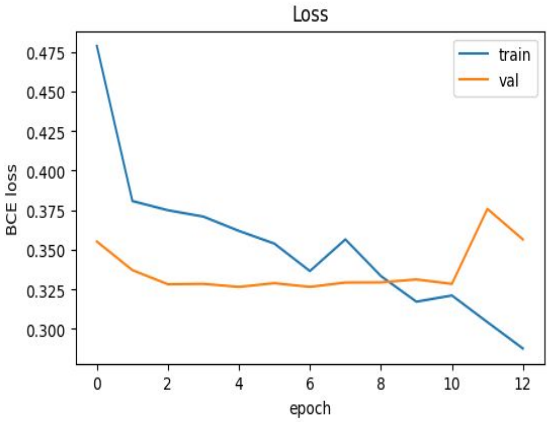
➢ The goal of the project is to discriminate between jets that originate from boosted W-boson and QCD

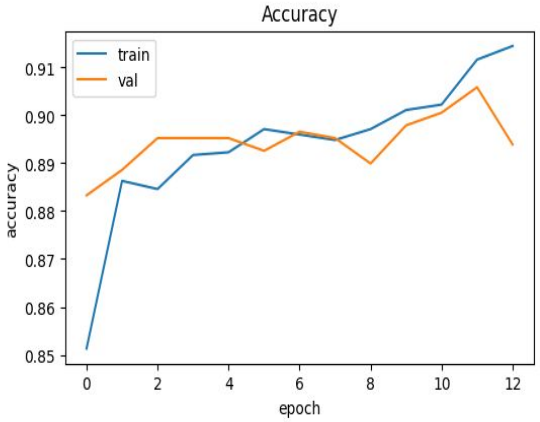


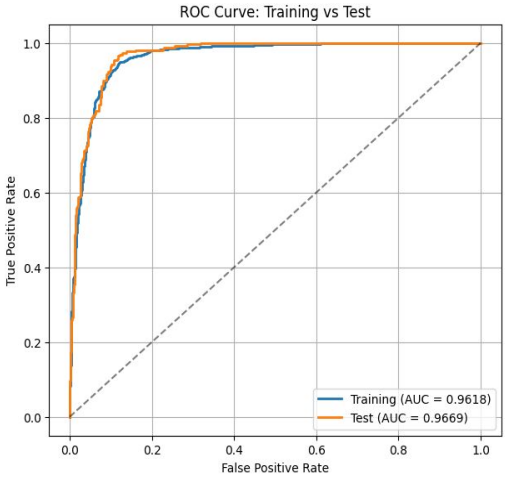


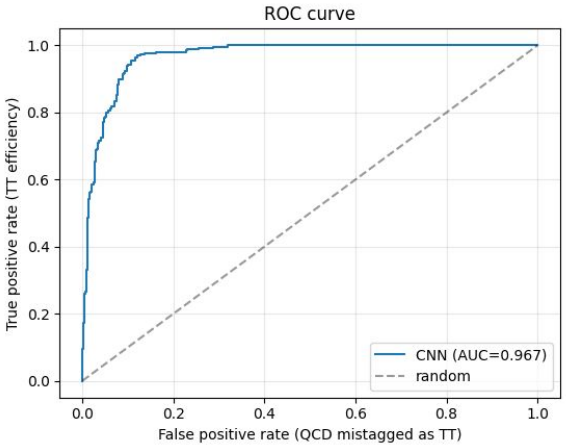
CNN - Architecture

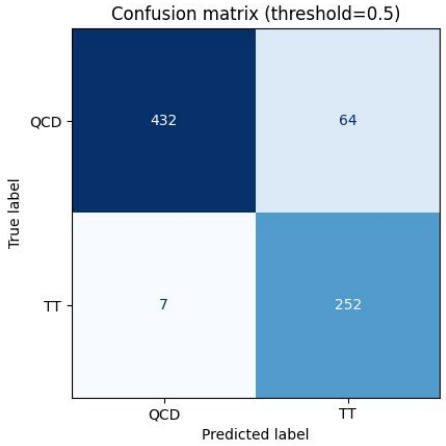


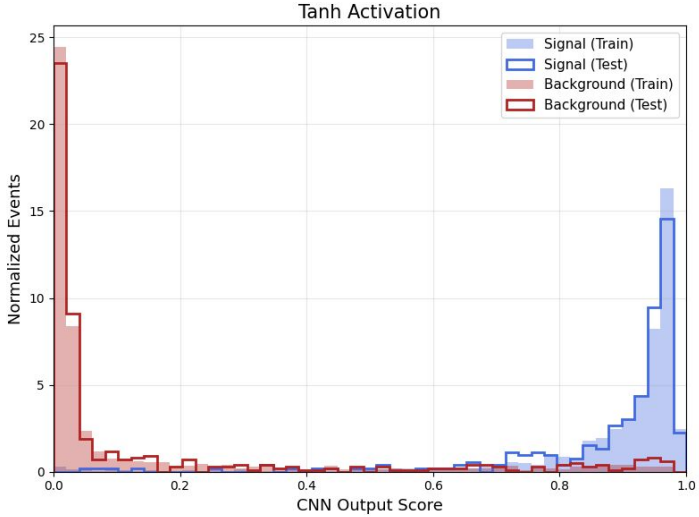

Loss
train
val
BCE loss
epoch


Accuracy
train
val
accuracy
epoch

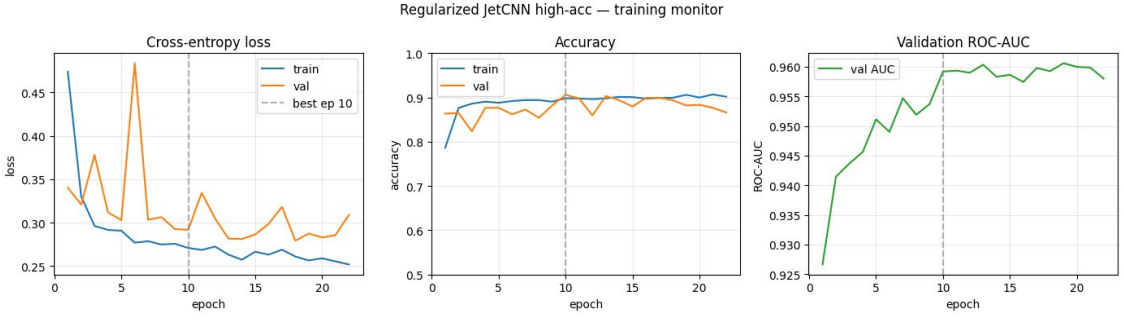

ROC Curve: Training vs Test
True Positive Rate
False Positive Rate
Training (AUC = 0.9618)
Test (AUC = 0.9669)

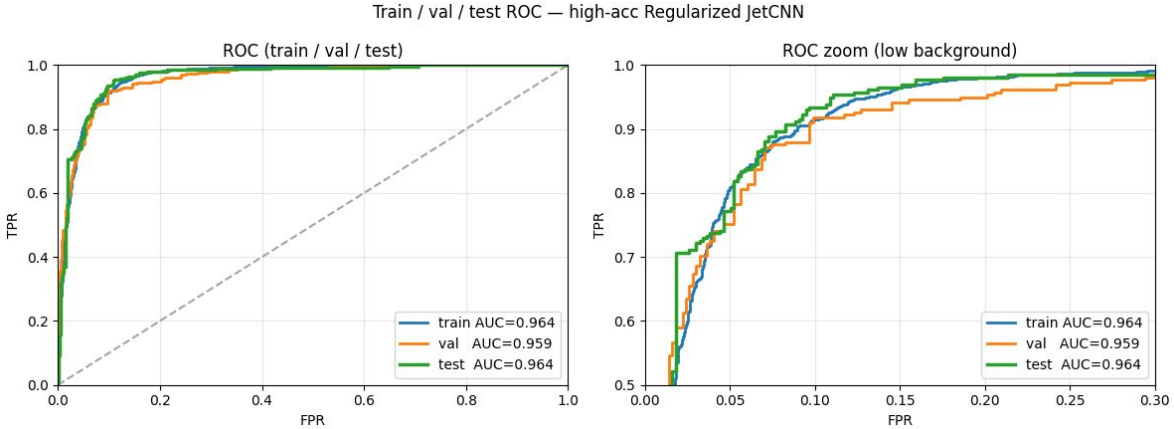

ROC curve
True positive rate (TT efficiency)
False positive rate (QCD mistagged as TT)
CNN (AUC=0.967)
random

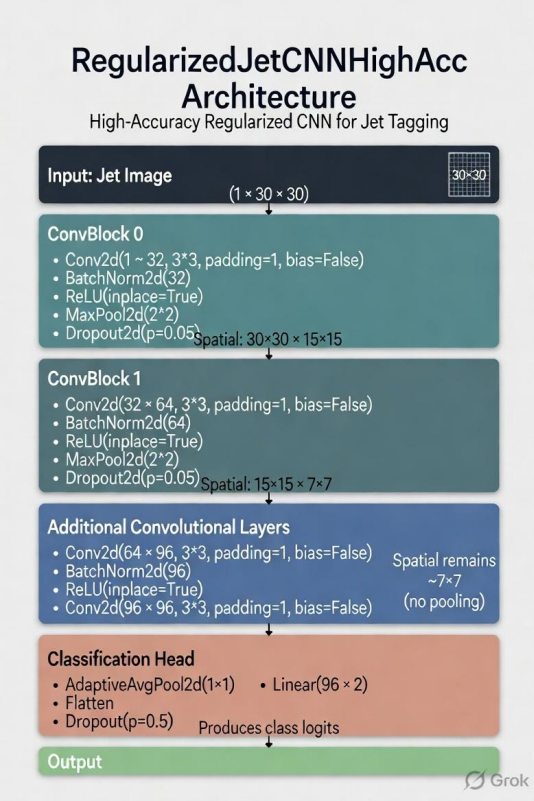

Confusion matrix (threshold=0.5)
QCD
TT
True label
Predicted label
432
64
7
252


Tanh Activation
Signal (Train)
Signal (Test)
Background (Train)
Background (Test)
Normalized Events
CNN Output Score

Architecture 2

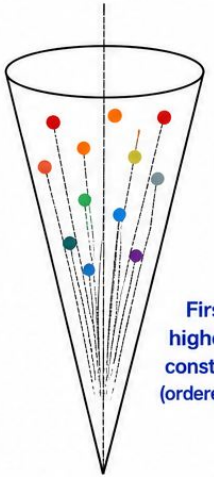






Jets as Feature Vector

Variable	Definition
$\Delta\eta$	difference in pseudorapidity between the particle and the jet axis
$\Delta\phi$	difference in azimuthal angle between the particle and the jet axis
$\log p_T$	logarithm of the particle's p_T
$\log E$	logarithm of the particle's energy
$\log \frac{p_T}{p_T(\text{jet})}$	logarithm of the particle's p_T relative to the jet p_T
$\log \frac{E}{E(\text{jet})}$	logarithm of the particle's energy relative to the jet energy
ΔR	angular separation between the particle and the jet axis ($\sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$)



A jet is represented using the first 10 highest-p_T constituents with 7 kinematic features per constituent

Constituent-level Features (7 per constituent)

Feature Name	η_{rel} (eta relative to jet axis)	ϕ_{rel} (phi relative to jet axis)	p_T (transverse momentum)	E (energy)	p_{Trel} (relative p_T to jet)	E_{rel} (relative energy to jet)	ΔR (distance from jet axis)
Column Index	8	11	5	3	6	4	13
Constituent Index (order)	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1 (highest p_T)	η_{rel}^1	ϕ_{rel}^1	p_T^1	E^1	p_{Trel}^1	$Erel^1$	ΔR^1
2	η_{rel}^2	ϕ_{rel}^2	p_T^2	E^2	p_{Trel}^2	$Erel^2$	ΔR^2
3	η_{rel}^3	ϕ_{rel}^3	p_T^3	E^3	p_{Trel}^3	$Erel^3$	ΔR^3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
10 (10th highest p_T)	η_{rel}^{10}	ϕ_{rel}^{10}	p_T^{10}	E^{10}	p_{Trel}^{10}	$Erel^{10}$	ΔR^{10}

First 10 highest-p_T constituents (ordered by p_T) →

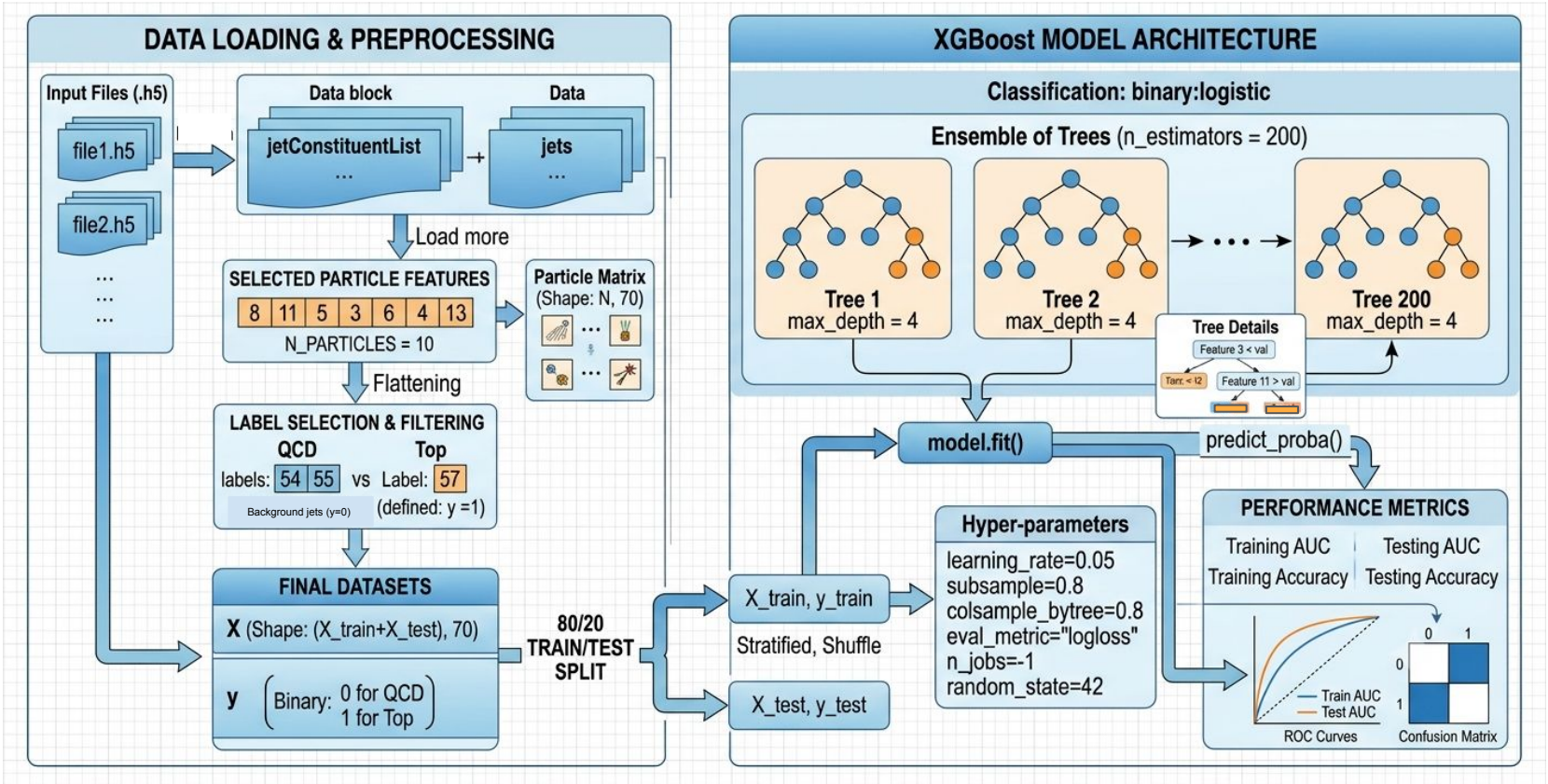
10 constituents × 7 features = 70 values

Flatten (10 × 7) row-wise → 70-dimensional Feature Vector

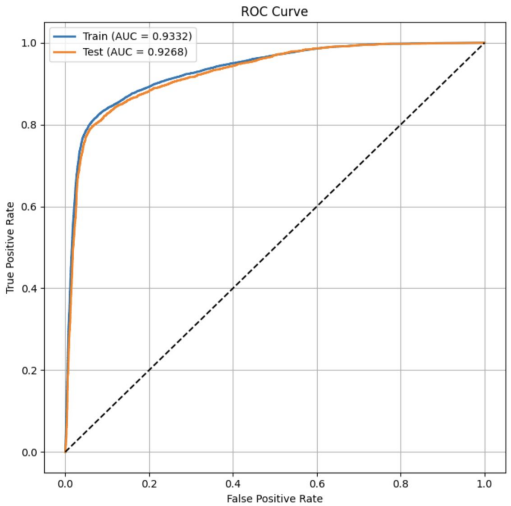
Feature (Col. Index)	8	11	5	3	6	4	13	8	11	5	3	6	4	13	…	8	11	5	3	6	4	13
Position in Vector →	η_{rel}^1	ϕ_{rel}^1	p_T^1	E^1	p_{Trel}^1	$Erel^1$	ΔR^1	η_{rel}^2	ϕ_{rel}^2	p_T^2	E^2	p_{Trel}^2	$Erel^2$	ΔR^2	…	η_{rel}^{10}	ϕ_{rel}^{10}	p_T^{10}	E^{10}	p_{Trel}^{10}	$Erel^{10}$	ΔR^{10}
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	…	64	65	66	67	68	69	70

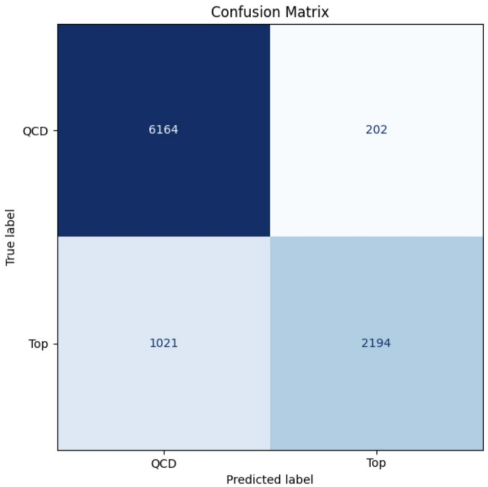
70 values

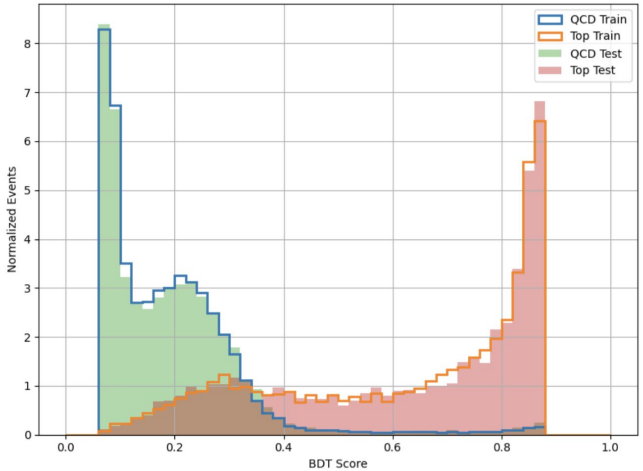
BDT Architecture



BDT Results


ROC Curve
Train (AUC = 0.9332)
Test (AUC = 0.9268)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


Confusion Matrix
QCD
Top
6164
202
1021
2194
True label
Predicted label
QCD
Top


QCD Train
Top Train
QCD Test
Top Test
Normalized Events
BDT Score
0
1
2
3
4
5
6
7
8
0.0
0.2
0.4
0.6
0.8
1.0

Jets as Sets

```
Jet features:
  [0] j_ptfrac
  [1] j_pt
  [2] j_eta
  [3] j_mass
  [4] j_tau1_b1
  [5] j_tau2_b1
  [6] j_tau3_b1
  [7] j_tau1_b2
  [8] j_tau2_b2
  [9] j_tau3_b2
  [10] j_tau32_b1
  [11] j_tau32_b2
  [12] j_zlogz
  [13] j_c1_b0
  [14] j_c1_b1
  [15] j_c1_b2
  [16] j_c2_b1
  [17] j_c2_b2
```

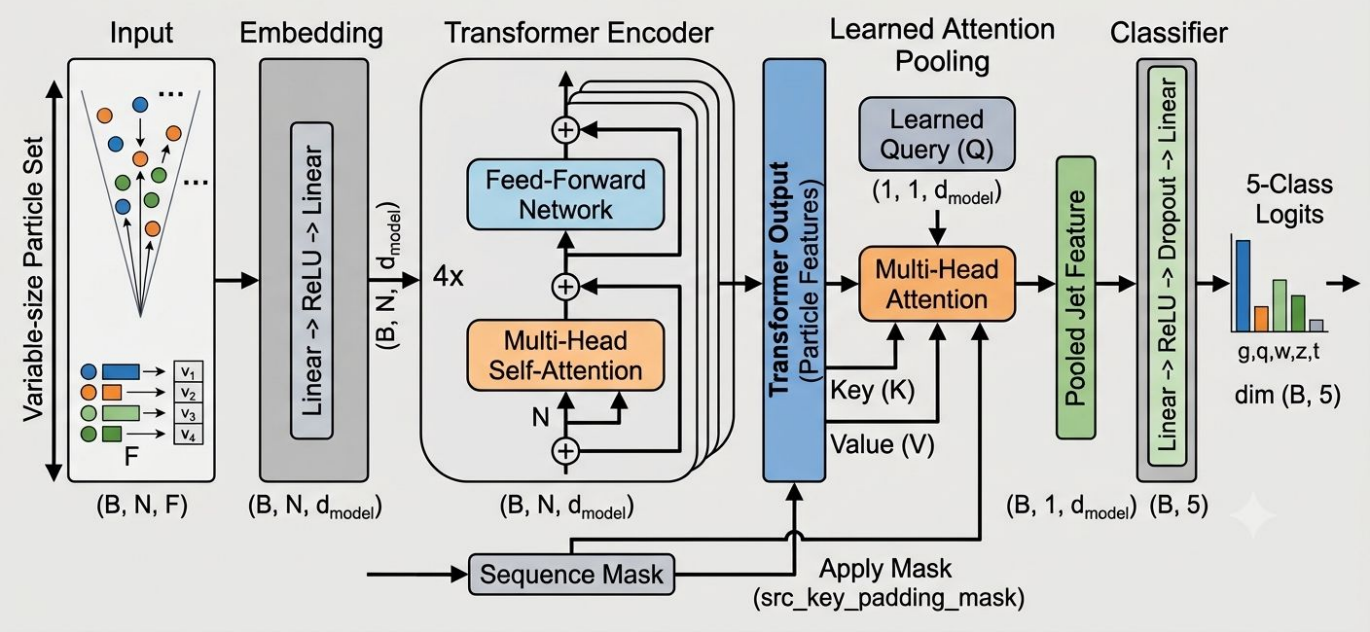
```
Particle features:
  [0] j1_px
  [1] j1_py
  [2] j1_pz
  [3] j1_e
  [4] j1_erel
  [5] j1_pt
  [6] j1_ptrel
  [7] j1_eta
  [8] j1_etarel
  [9] j1_etarot
  [10] j1_phi
  [11] j1_phirel
  [12] j1_phirot
  [13] j1_deltaR
  [14] j1_costheta
  [15] j1_costhetarel
  [16] j1_pdgid
```

```
jetConstituentList (10000, 100, 16) float64
jetFeatureNames (59,) object
jetImage (10000, 100, 100) float64
jetImageECAL (10000, 100, 100) float64
jetImageHCAL (10000, 100, 100) float64
jets (10000, 59) float64
particleFeatureNames (17,) object
```

Variable	Definition
$\Delta\eta$	difference in pseudorapidity between the particle and the jet axis
$\Delta\phi$	difference in azimuthal angle between the particle and the jet axis
$\log p_T$	logarithm of the particle's p_T
$\log E$	logarithm of the particle's energy
$\log \frac{p_T}{p_T(\text{jet})}$	logarithm of the particle's p_T relative to the jet p_T
$\log \frac{E}{E(\text{jet})}$	logarithm of the particle's energy relative to the jet energy
ΔR	angular separation between the particle and the jet axis ($\sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$)

https://arxiv.org/abs/1902.08570

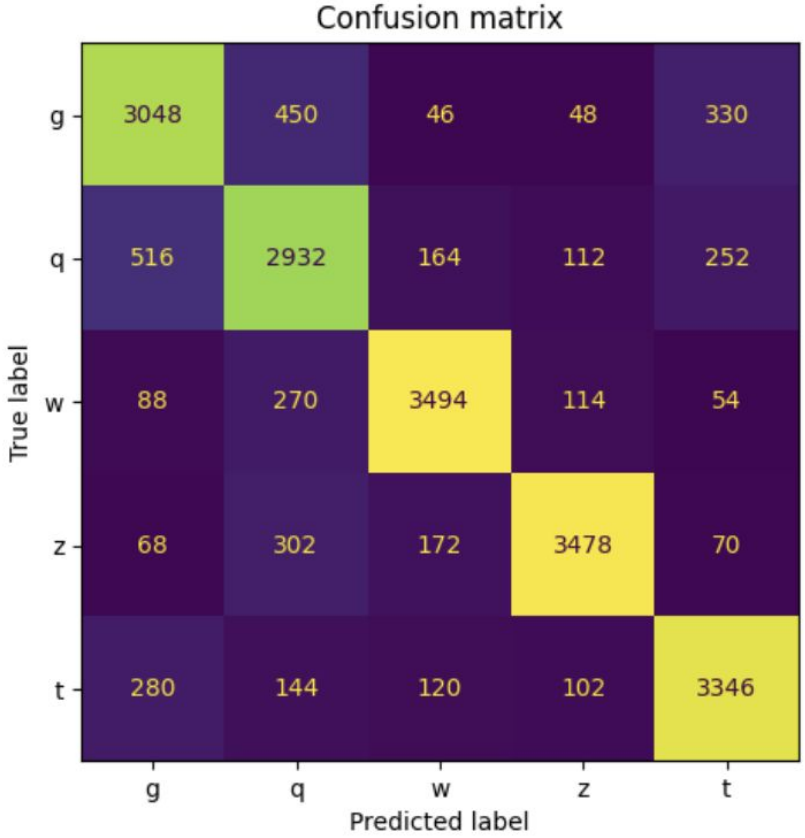
Transformer Architecture

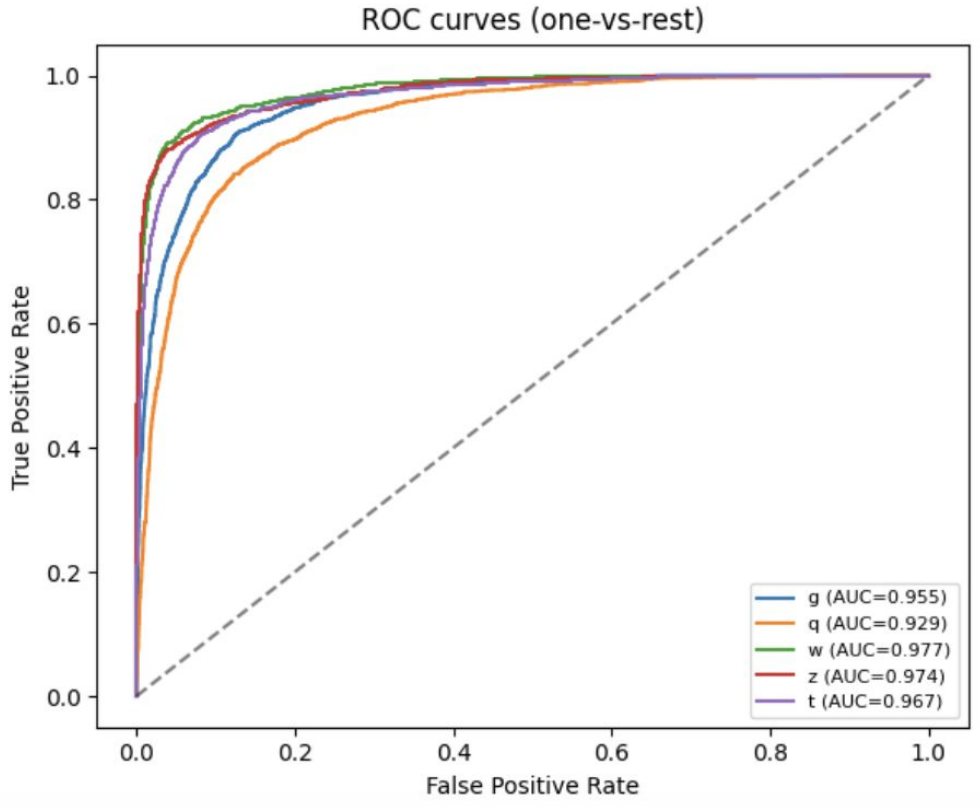



```
N_EPOCHS = 200
PATIENCE = 10
d_model = 64
```

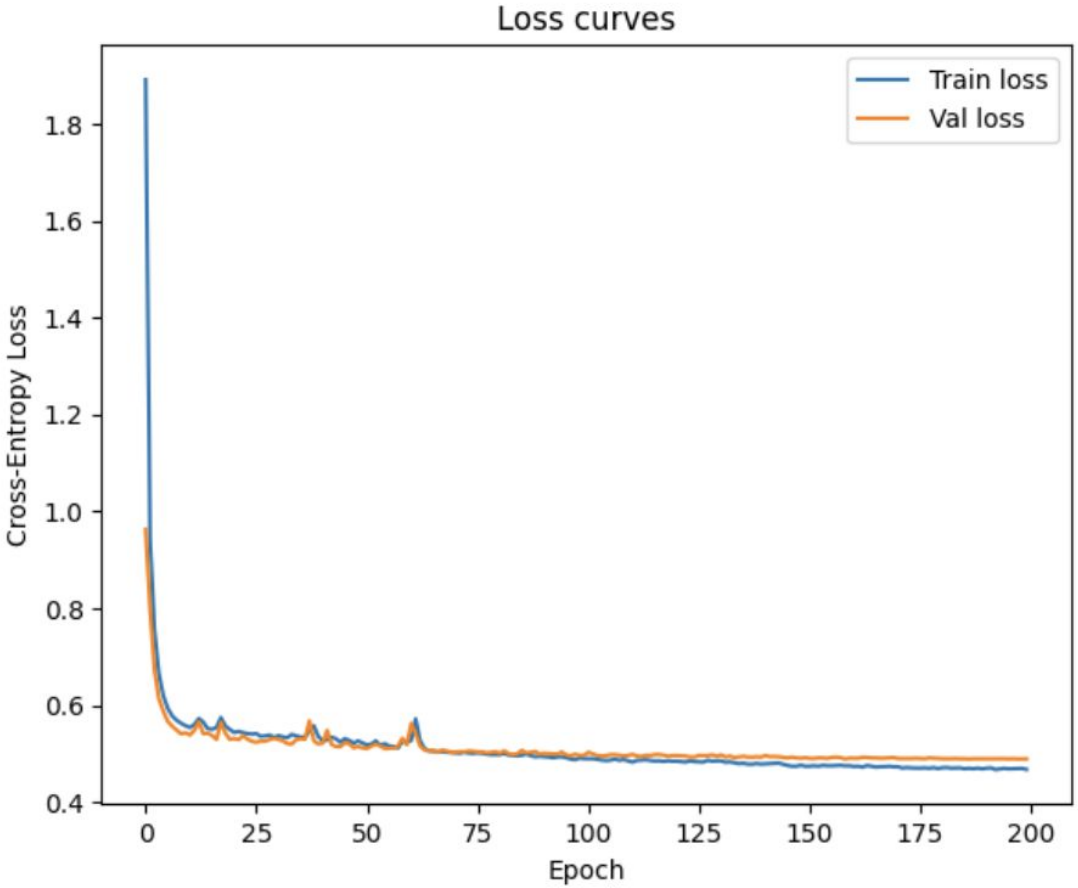
$$\mathcal{L} = -\log\left[\frac{e^{z_{true}}}{\sum_{i=1}^{5} e^{z_i}}\right]$$

GNN-Transformer Results





No Overtraining!

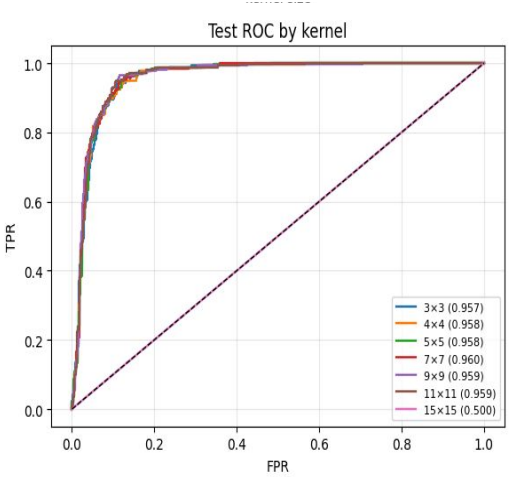


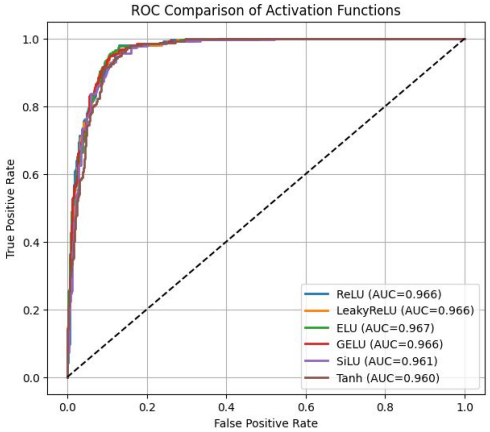
Summary

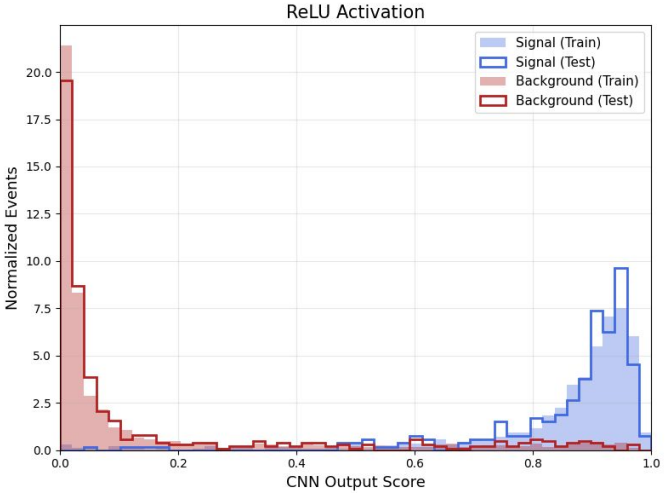
- Classify jets as per their sources using simulated datasets
- CNN - treat the jet as an image and classify the images into different labels
- BDT - treat the jet as a feature vector, and correct the decision boundaries on particle features to evaluate the class
- Transformer - treat the particles in a list as unordered sets and classify them into finer classes

Thank You!!

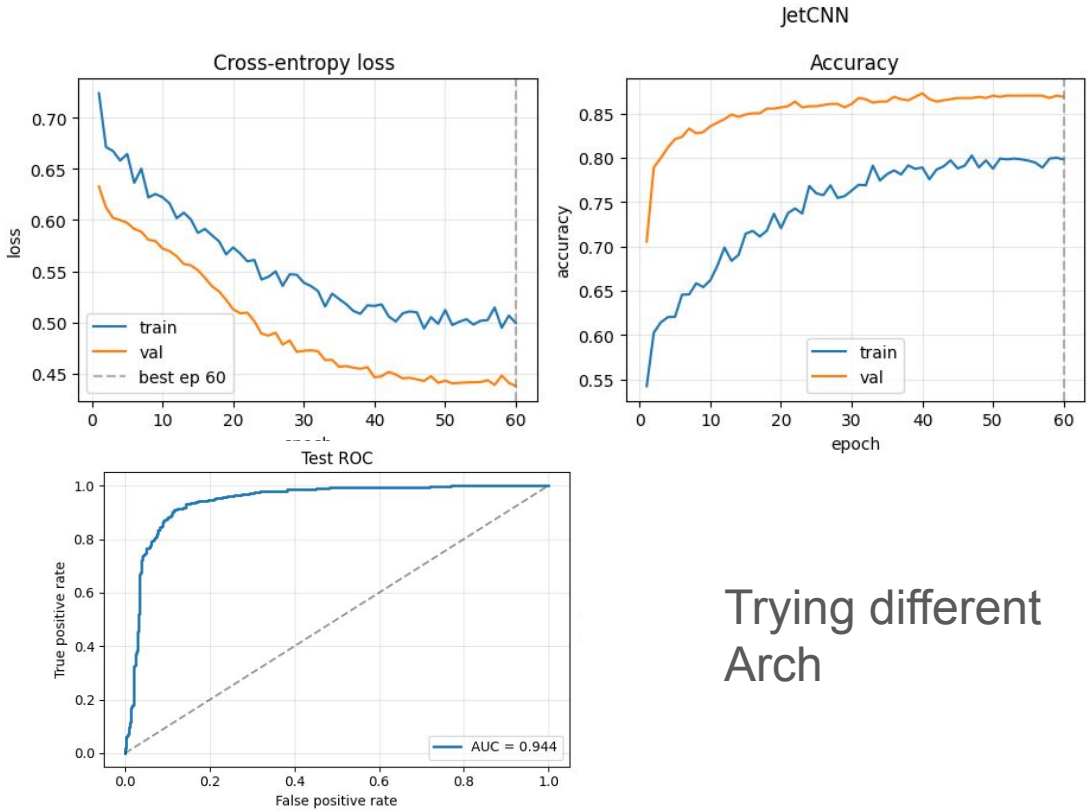
Sweep over architecture



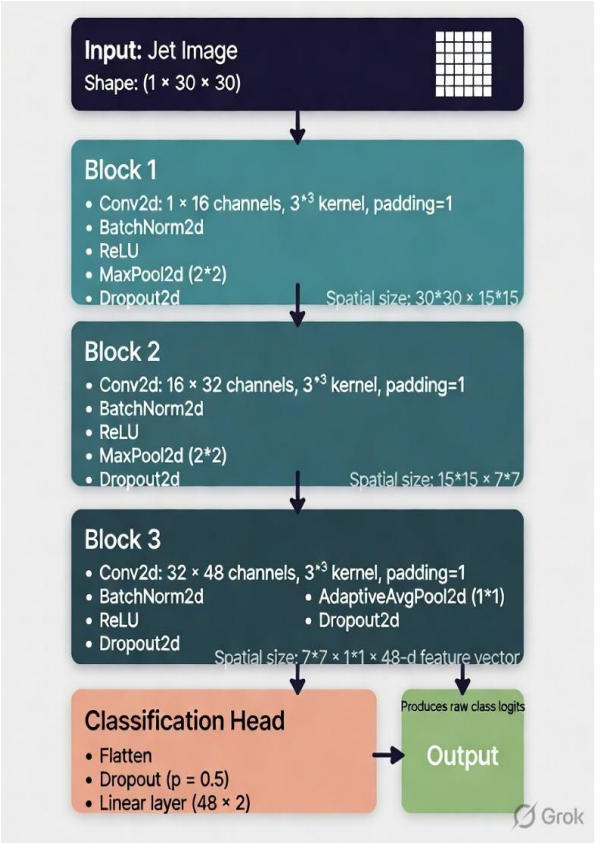


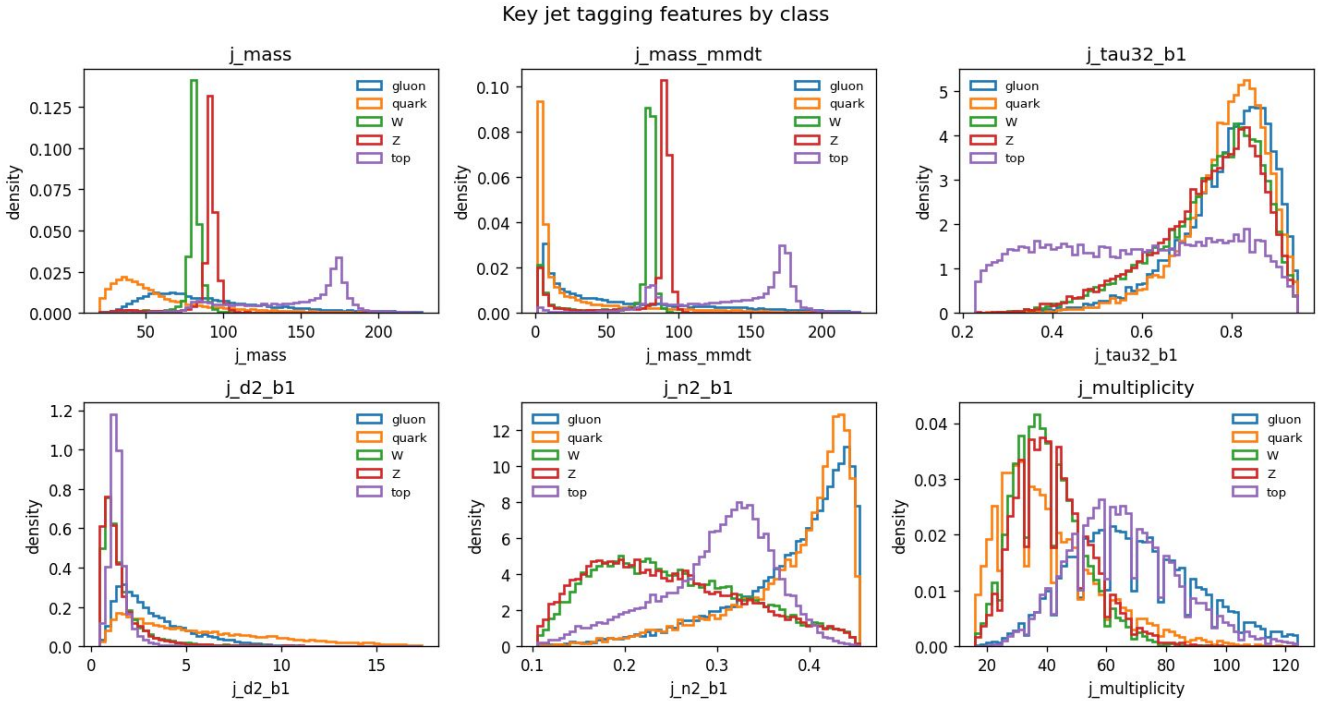


Architecture 2

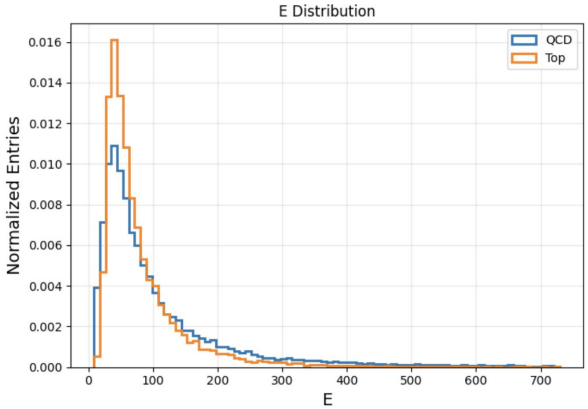


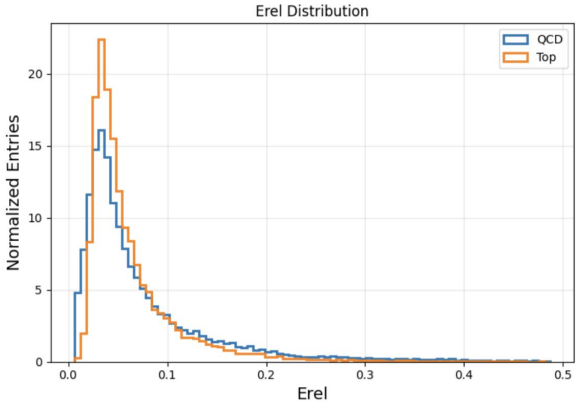
Trying different Arch

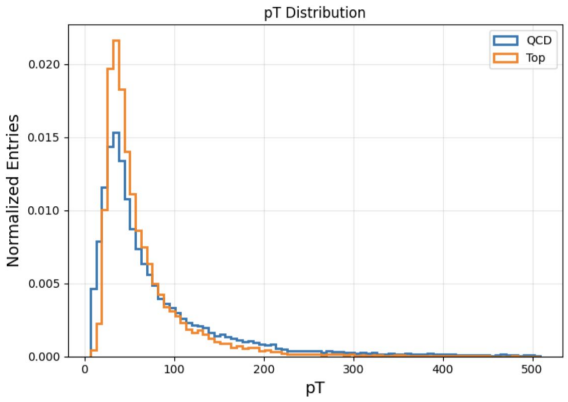


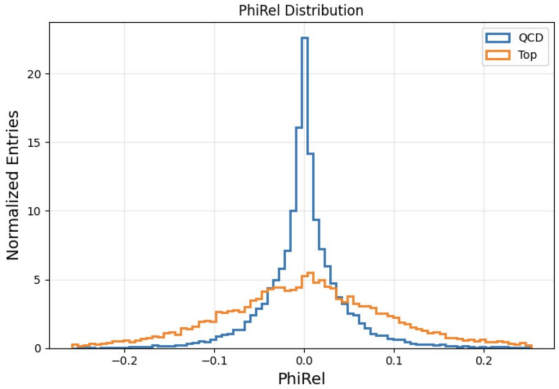

Key jet tagging features by class
j_mass
j_mass_mmdt
j_tau32_b1
j_d2_b1
j_n2_b1
j_multiplicity
gluon
quark
W
Z
top
density

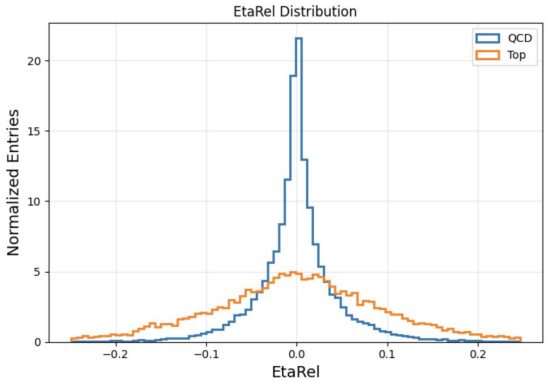
BDT

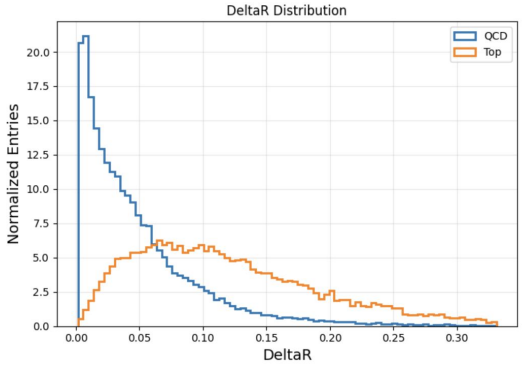

E Distribution
QCD
Top
Normalized Entries
E


Erel Distribution
QCD
Top
Normalized Entries
Erel

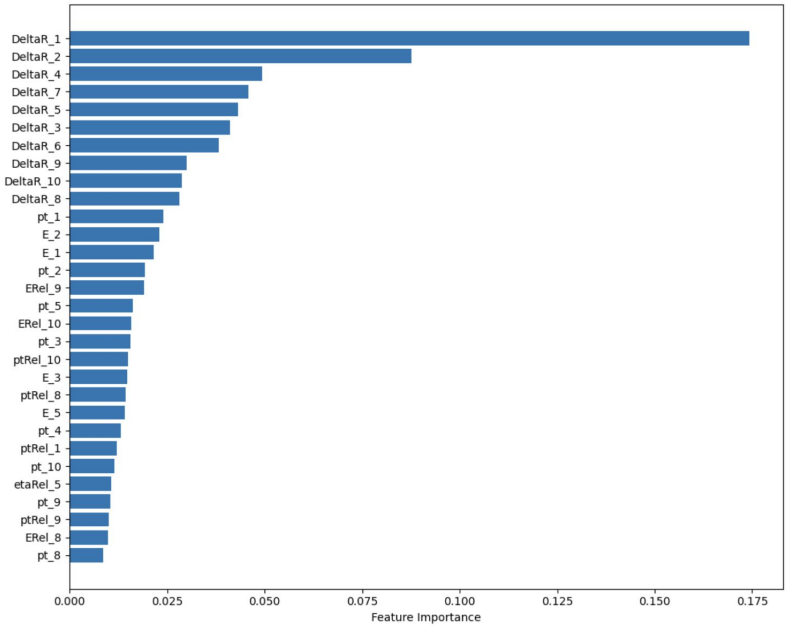

pT Distribution
QCD
Top
Normalized Entries
pT

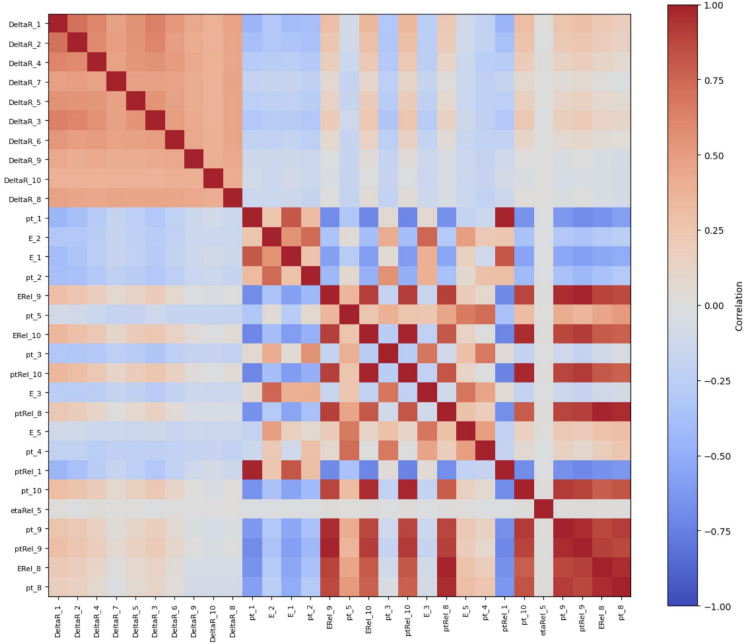

PhiRel Distribution
QCD
Top
Normalized Entries
PhiRel


EtaRel Distribution
QCD
Top
Normalized Entries
EtaRel

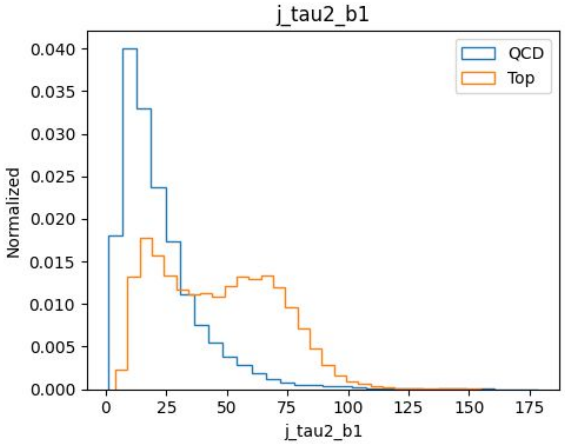

DeltaR Distribution
QCD
Top
Normalized Entries
DeltaR

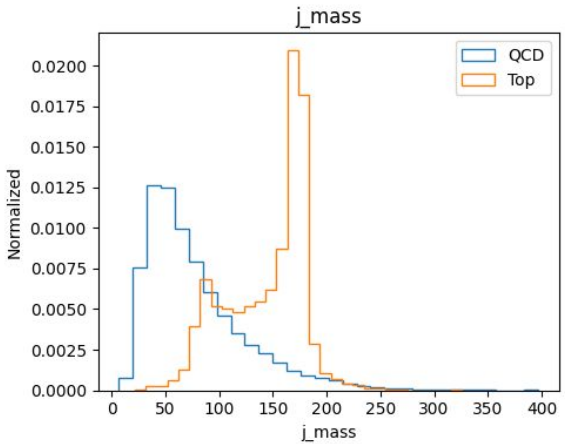
BDT

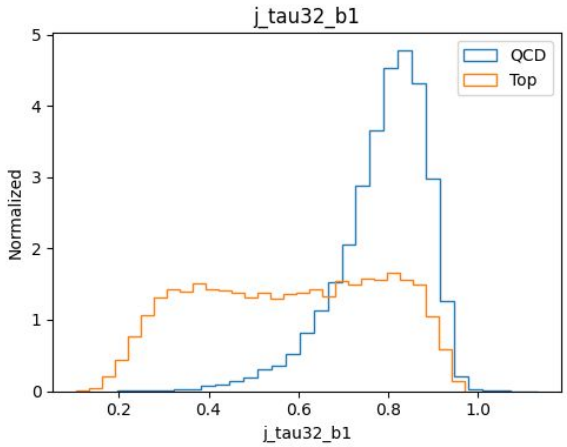

DeltaR_1
DeltaR_2
DeltaR_4
DeltaR_7
DeltaR_5
DeltaR_3
DeltaR_6
DeltaR_9
DeltaR_10
DeltaR_8
pt_1
E_2
E_1
pt_2
ERel_9
pt_5
ERel_10
pt_3
ptRel_10
E_3
ptRel_8
E_5
pt_4
ptRel_1
pt_10
etaRel_5
pt_9
ptRel_9
ERel_8
pt_8
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
Feature Importance

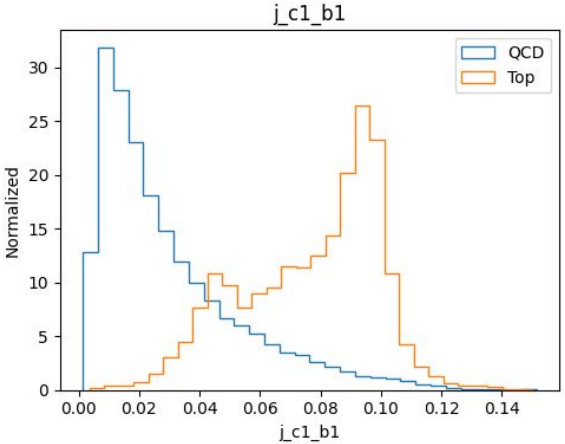

Correlation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

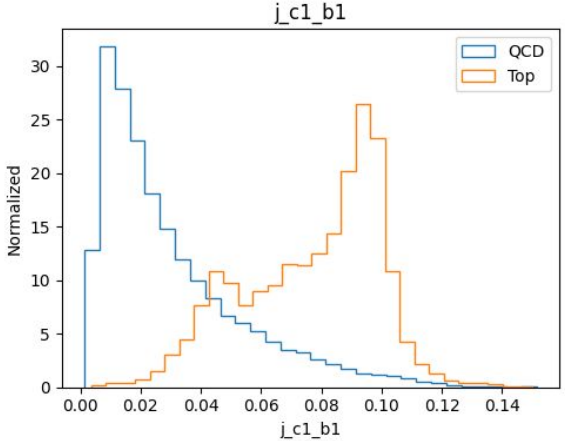
BDT using high level jet features

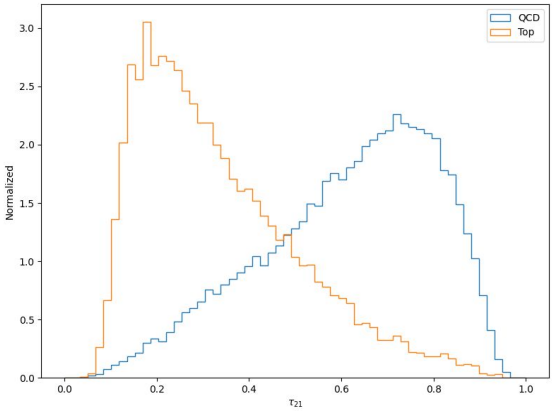




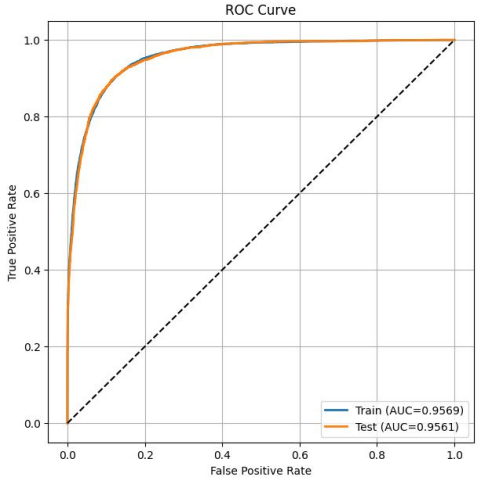


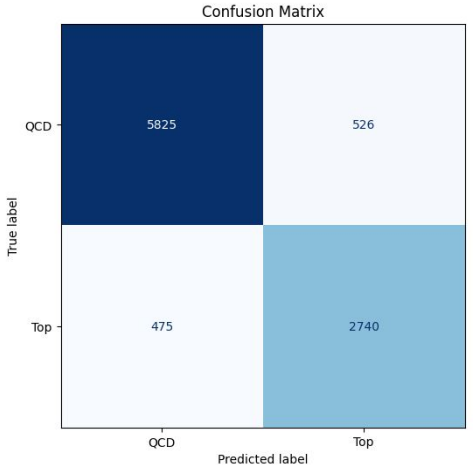


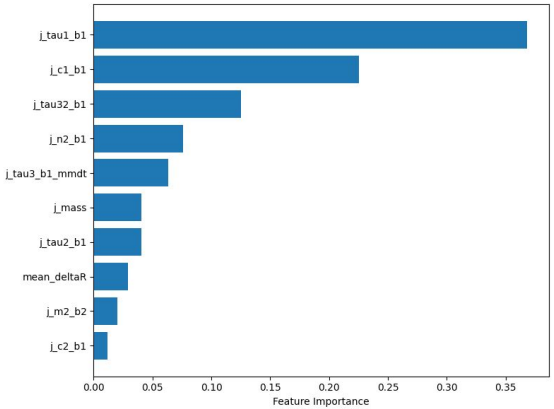


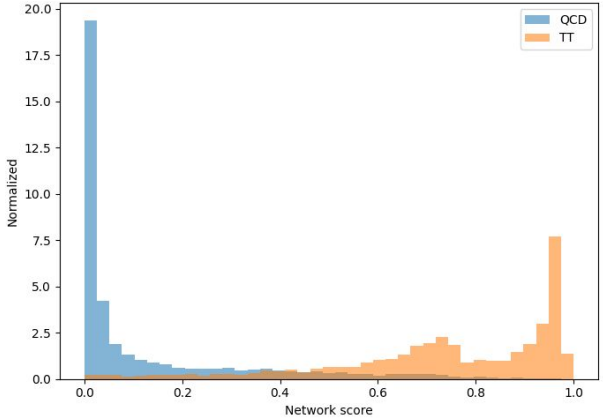


BDT using high level jet features









Query, Key, Value Architecture

1 · From hard k-NN edges to soft attention

Write a GNN layer and an attention layer side by side:

$$\textbf{GNN:}\ h_i' = \phi\Big(h_i,\ \bigoplus_{j\in\mathcal{N}(i)} \psi(h_i, h_j)\Big) \qquad \textbf{Attention:}\ h_i' = \sum_j \alpha_{ij} v_j,\quad \alpha_{ij} = \frac{\exp(q_i\cdot k_j/\sqrt{d})}{\sum_{j'}\exp(q_i\cdot k_{j'}/\sqrt{d})}$$

The only differences: the neighbourhood is **all** particles, and the "edge weight" α_{ij} is **learned and soft** (a probability) rather than a 0/1 k-NN membership. So a Transformer is a GNN on a **fully-connected graph with a learned, normalized, per-layer adjacency**. Everything you know from Module 2 transfers.

Because the sum over j is permutation-invariant, attention on a *set* needs **no positional encoding** — and that is exactly what we want for particles, which have no canonical order.

2 · Scaled dot-product attention

Given queries Q, keys K, values V (each a matrix with one row per particle):

$$\text{Attention}(Q,K,V) = \text{softmax}\Big(\frac{QK^\top}{\sqrt{d_k}}\Big)V.$$

- $QK^\top$ scores how much particle i (query) cares about particle j (key);
- $\sqrt{d_k}$ keeps the scores from blowing up with dimension;
- softmax turns each row into attention weights $\alpha_{i\cdot}$ that sum to 1;
- multiplying by V returns a weighted average of the value vectors.

We must **mask padded particles**: their key columns get $-\infty$ before the softmax so nobody attends to them.