Quark and Gluon Discrimination in QCD Jets and Deep Neural Network (DNN)

İdil Turan
Yeditepe University – Physics

Quarks
Matter Particles

- Quarks are the smalleset fundamental building particles of matter; they form protons and neutrons
- carry a color charge in QCD: Red, Green or Blue
- Quarks have smaller color factor (Cf = 4/3) compared to Gluons Ca = 3
- Because of color factor, quarks radiate fewer soft gluons

Gluons
Force Carriers

- Massless spin-1 gauge bosons that carry the strong nuclear force.
- Unlike photons in QED, gluons carry color charge themselves -specifically a color and an anti-color combination
- This self-coupling allows gluons to radiate other gluons

Quantum Chromodynamics (QCD) ?

The Theory of Strong Force

- QCD is the fundamental quantum theory of strong interaction between quarks and gluons.
- It explains how quarks and gluons interact with each other

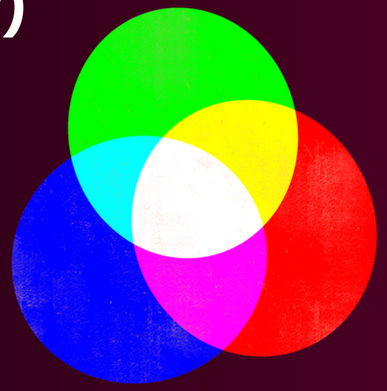
Color Charge

- Like electromagnetism uses electric charge (+, -), QCD uses color charge: Red, Green and Blue
- Quarks and gluons carry this color charge. But other particles is neutral.

Two Main Rules of QCD

Asymptotic Freedom (At high energy)

- At very high energies, the strong force becomes weak
- Quarks and Gluons act almost like free particles



Color Confinement (At low energy)

- Quarks and Gluons cannot exist alone in nature
- They must always combine to form color - neutral particles (hadrons)

From Partons to Jets

1. Hard Scattering

- Parton–parton interaction
- High-energy outgoing q/g

2. QCD radiation

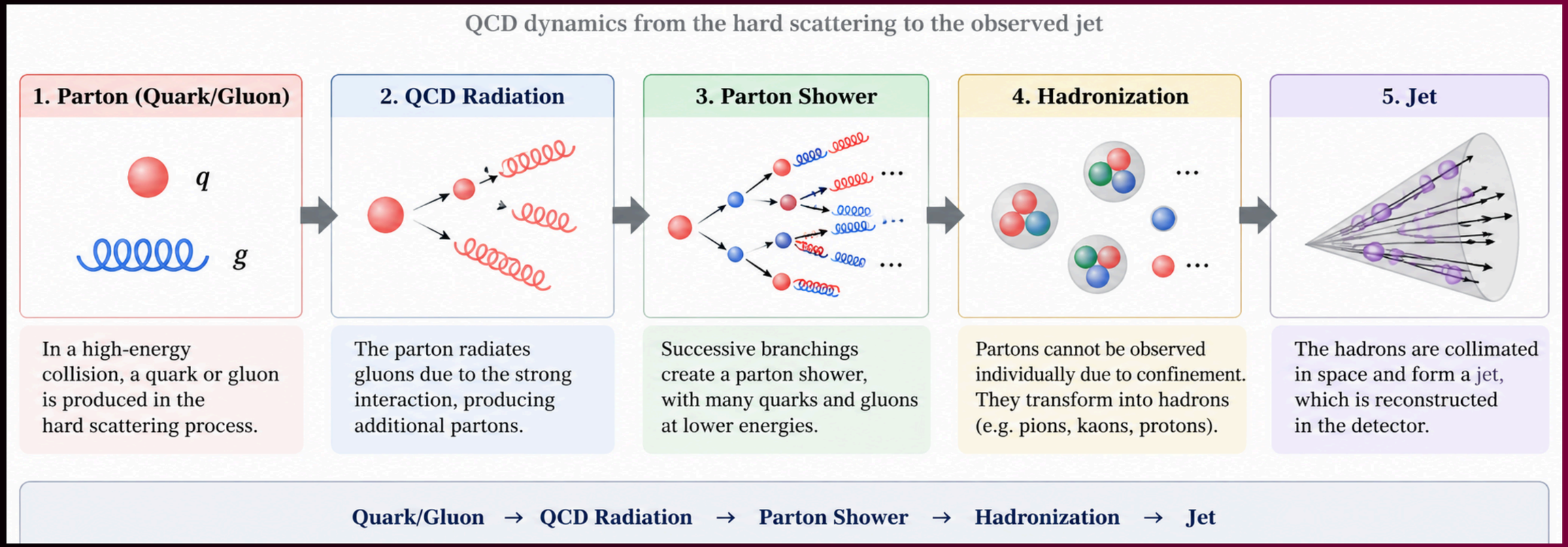
- $q \rightarrow q+g$
- $g \rightarrow gg, q\bar{q}$
- Parton shower

3. Hadronization

- Color confinement
- Partons → color-neutral hadrons

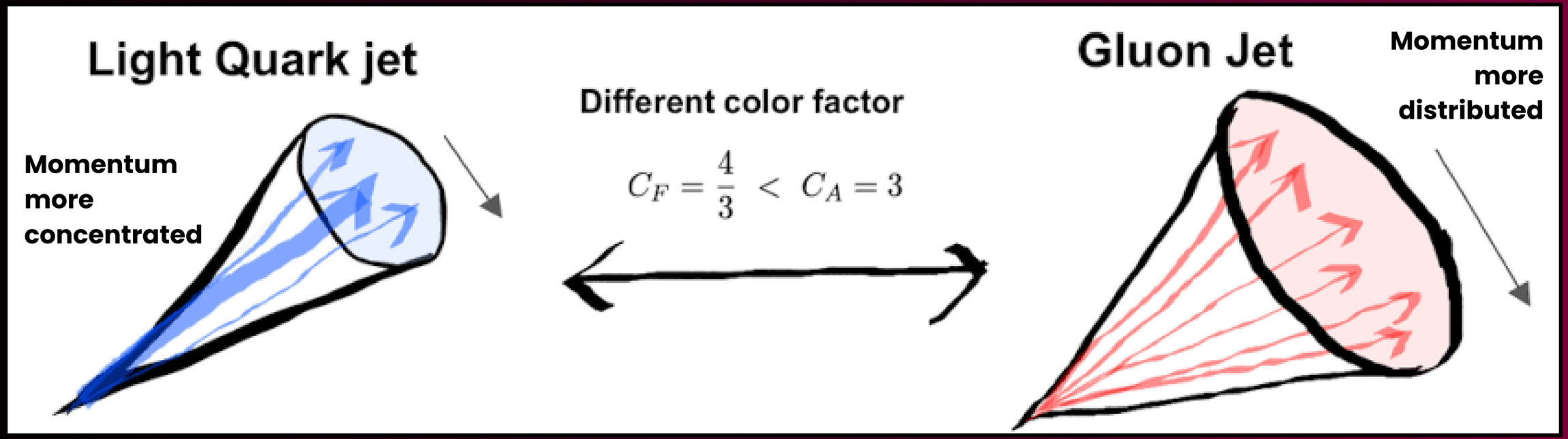
4. Jet

- Hadrons form a collimated spray



Quark Jets vs Gluon Jets

While both jets generally go through the same processes, the initial part might be different.



Different color factors → different radiation patterns → different jet structures

Jet Variables

How can we measure the differences between quark and gluon jets?

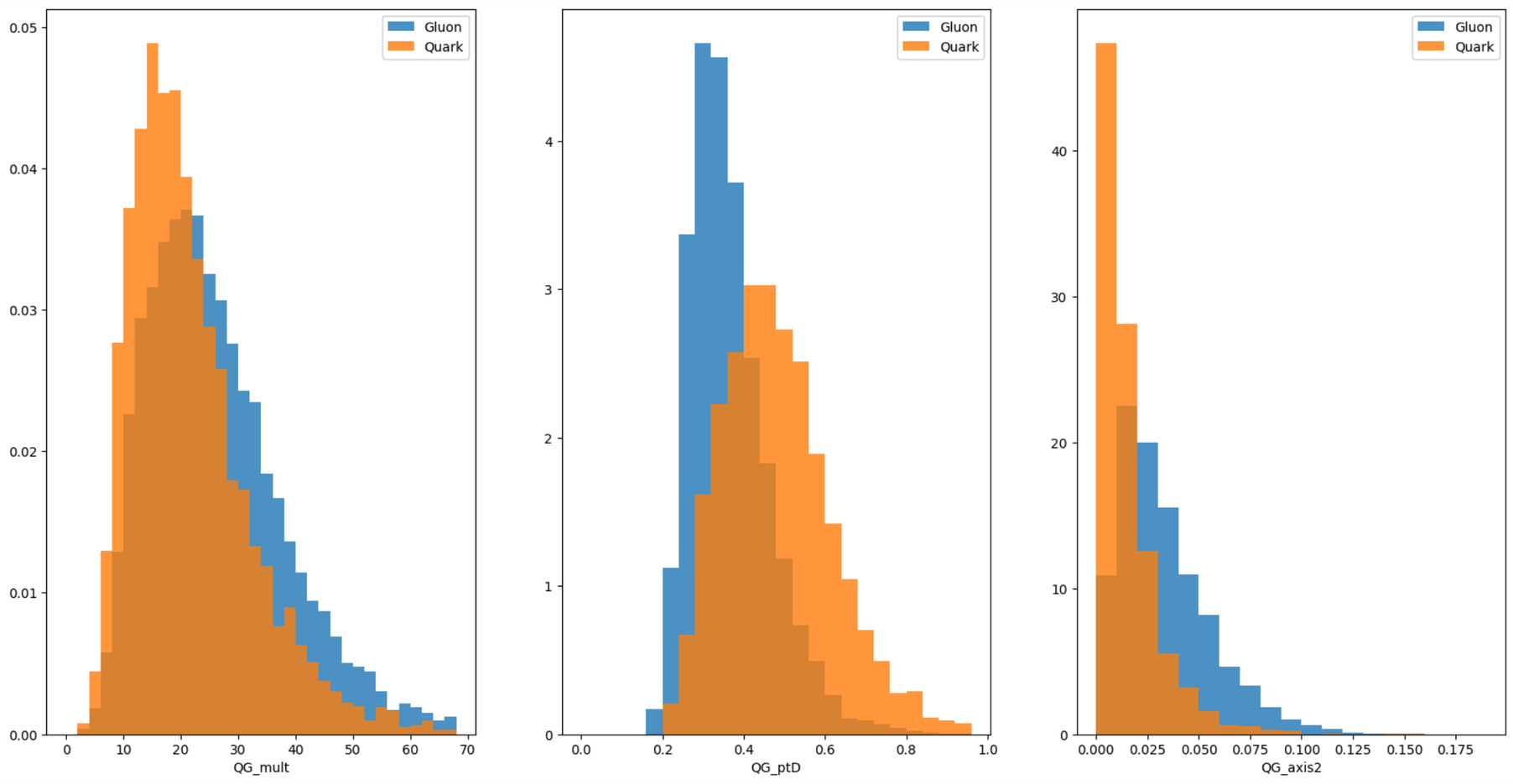
QG_mult = Number of constituents in a jet

QG_ptD = How the jet's transverse momentum is distributed among its constituents.

QG_axis2 = The width of the jet / its minor axis.

DataSet & Analysis

Dataset:
CMS OpenData QCD Jets
ROOT file:
JetNtuple_RunIISummer16_13TeV_MC_1.root



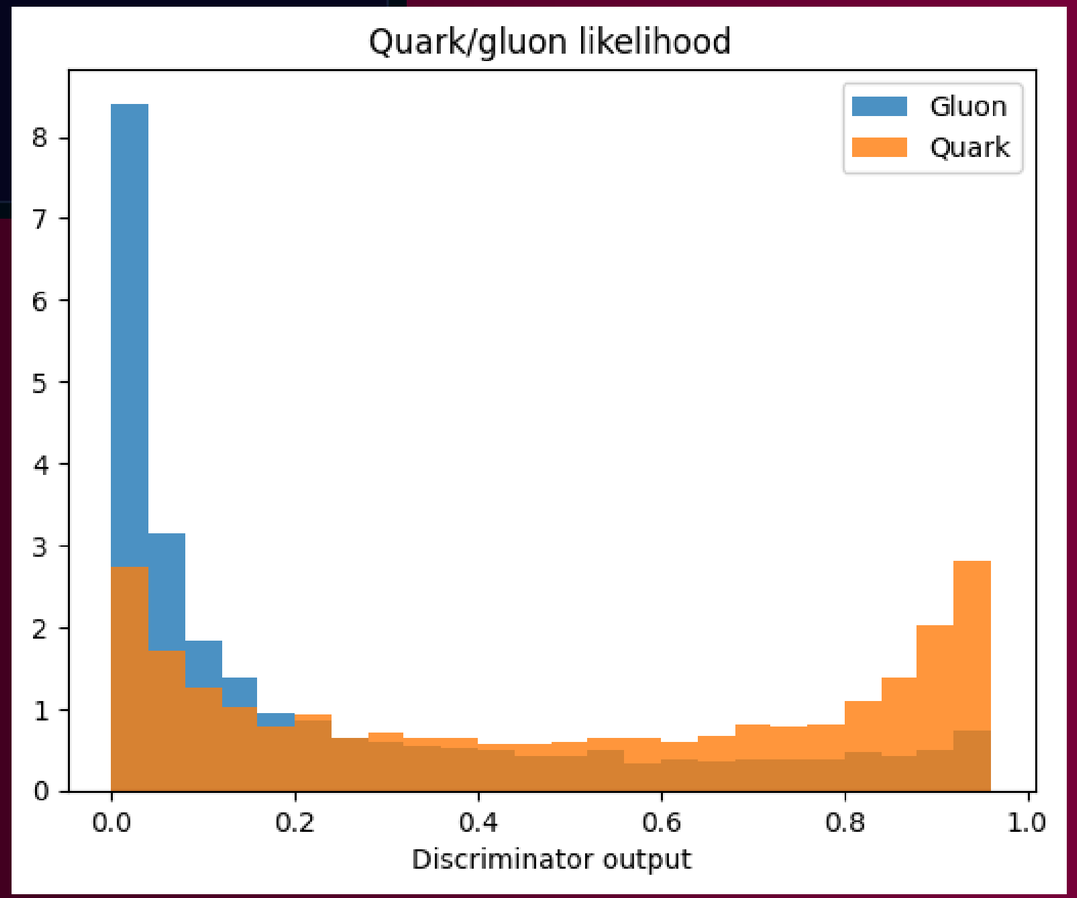
Tree:
AK4jets/jetTree

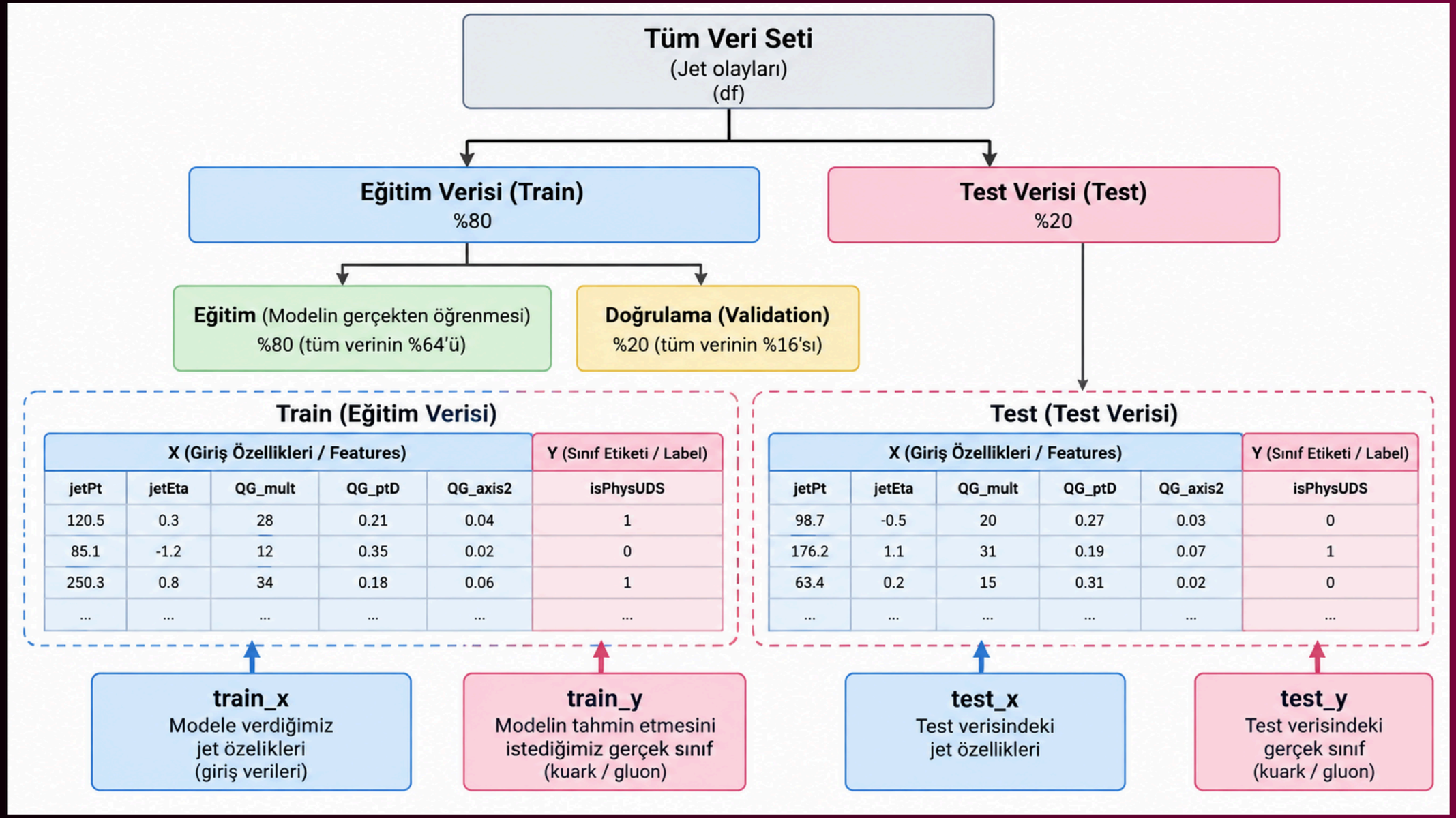
Selected jets:
Gluon (isPhysG = 1)
L. Quark / UDS (isPhysUDS = 1)

Variables:
QG_mult
QG_ptD
QG_axis2

Quark - Gluon Distributions

```
plt.clf()
binning = np.arange(0.0, 1.0, 0.04)
plt.hist( df[df.isPhysG==1]['jetQGl'], bins=binning, alpha=0.8, label='Gluon', density=1 )
plt.hist( df[df.isPhysUDS==1]['jetQGl'], bins=binning, alpha=0.8, label='Quark', density=1 )
plt.legend()
plt.xlabel('Discriminator output')
plt.title('Quark/gluon likelihood');
```




Tüm Veri Seti
(Jet olayları)
(df)
Eğitim Verisi (Train)
%80
Test Verisi (Test)
%20
Eğitim (Modelin gerçekten öğrenmesi)
%80 (tüm verinin %64'ü)
Doğrulama (Validation)
%20 (tüm verinin %16'sı)
Train (Eğitim Verisi)
X (Giriş Özellikleri / Features)
Y (Sınıf Etiketi / Label)
jetPt jetEta QG_mult QG_ptD QG_axis2 isPhysUDS
120.5 0.3 28 0.21 0.04 1
85.1 -1.2 12 0.35 0.02 0
250.3 0.8 34 0.18 0.06 1
... ...
Test (Test Verisi)
X (Giriş Özellikleri / Features)
Y (Sınıf Etiketi / Label)
jetPt jetEta QG_mult QG_ptD QG_axis2 isPhysUDS
98.7 -0.5 20 0.27 0.03 0
176.2 1.1 31 0.19 0.07 1
63.4 0.2 15 0.31 0.02 0
... ...
train_x
Modele verdiğimiz
jet özelikleri
(giriş verileri)
train_y
Modelin tahmin etmesini
istediğimiz gerçek sınıf
(kuark / gluon)
test_x
Test verisindeki
jet özellikleri
test_y
Test verisindeki
gerçek sınıf
(kuark / gluon)

From Likelihood to Deep Neural Network (DNN)

```python
from sklearn.model_selection import train_test_split

train, test = train_test_split(df, test_size=0.2, random_state=42)

train_y = train.isPhysUDS
test_y = test.isPhysUDS
```

Train_y = is like the answer key of an exam. You give the model the questions with train_x and train_y tells the answer.

```python
input_features = ['jetPt', 'jetEta', 'QG_mult', 'QG_ptD', 'QG_axis2']

train_x = train[input_features]
test_x = test[input_features]

# Normalize the features
train_x_mean = train_x.mean()
train_x_std = train_x.std()

train_x = (train_x - train_x_mean) / train_x_std
test_x = (test_x - train_x_mean) / train_x_std
```

Train_x = represents the input feature containing given variables of jets,

Likelihood
QG_mult + QG_ptD + QG_axis2
↓
Limitation: variables are not uncorrelated
↓
DNN
jetPt + jetEta + QG_mult + QG_ptD + QG_axis2
↓
DNN Architecture
100 → 100 → 50 → 1
↓
Quark / Gluon classification

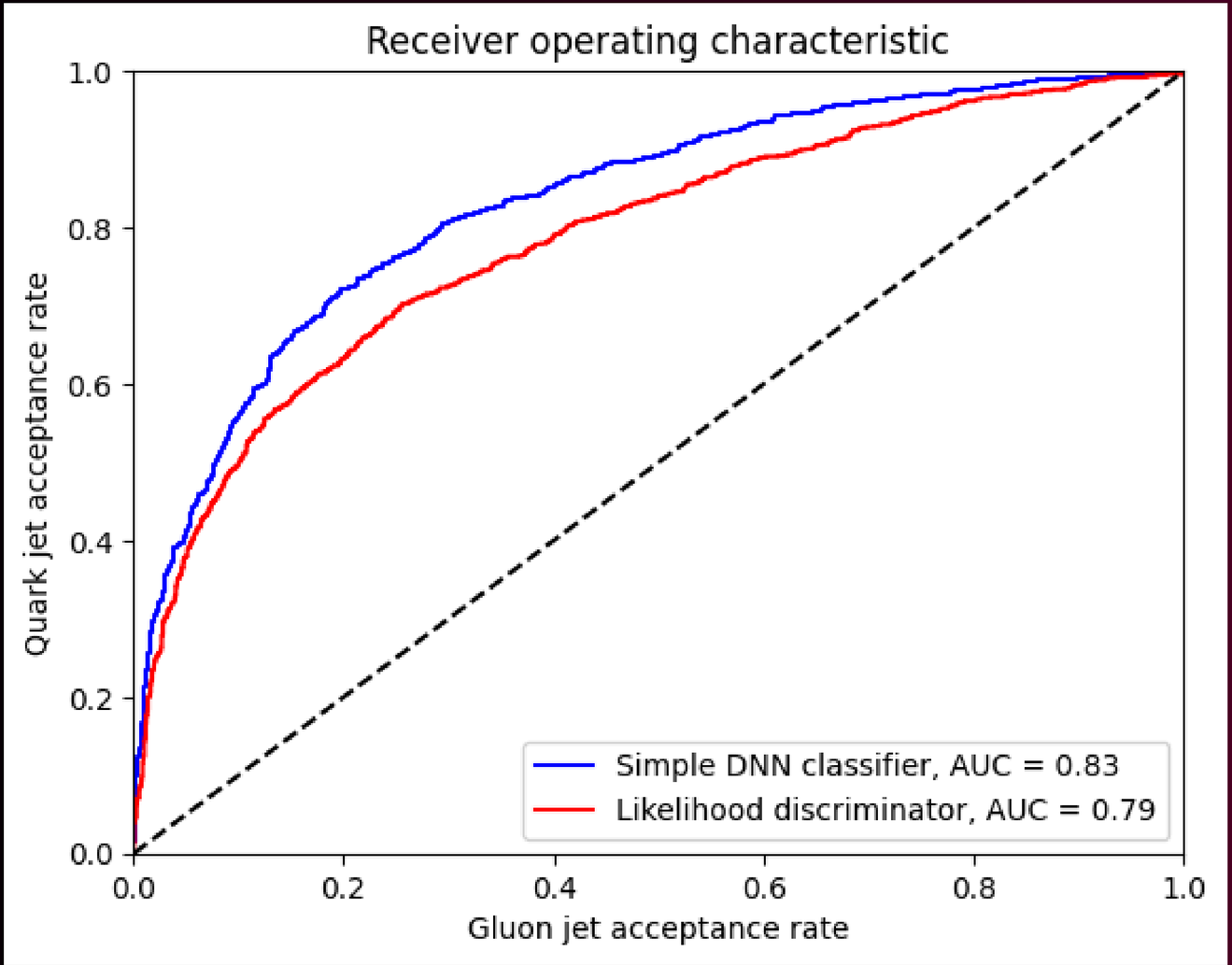
```
# Define the DNN architecture

model = Sequential()
model.add(Dense(100, kernel_initializer='normal', activation='relu', input_dim=train_x.shape[1]))
model.add(Dropout(0.2))
model.add(Dense(100, kernel_initializer='normal', activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(50, kernel_initializer='normal', activation='relu'))
model.add(Dense(1, kernel_initializer='normal', activation='sigmoid'))
model.compile(optimizer='Nadam', loss='binary_crossentropy', metrics=['accuracy'])

# Weight the training samples so that there is equal weight on gluon and quark jets
# even if there are different amount of them in the training set
classes = np.unique(train_y)
weights = class_weight.compute_class_weight('balanced', classes=classes,y=train_y)
class_weights_dict = dict(zip(classes, weights))

# Train the model
model.fit(train_x,
          train_y,
          epochs=20,
          batch_size=128,
          class_weight=class_weights_dict,
          validation_split=0.2,
          shuffle=True,
          verbose=1);
```

Results



- DNN classifier: AUC = 0.83
- Likelihood discriminator: AUC = 0.79

Higher AUC → better discrimination

Future Plans

- Explore additional jet observables and physical features
- Optimize the DNN architecture
- Compare different machine-learning models

Thank you for listening