Task II: Classical Graph Neural Network (GNN)

For Task II, you will use ParticleNet's data for Quark/Gluon jet classification available here with its corresponding description.

- Choose 2 Graph-based architectures of your choice to classify jets as being quarks or gluons. Provide a description on what considerations you have taken to project this point-cloud dataset to a set of interconnected nodes and edges.
- Discuss the resulting performance of the 2 chosen architectures.

```
!pip install energyflow        #Installing energyflow package to read
data from
!wget https://raw.githubusercontent.com/hqucms/ParticleNet/master/tf-
keras/tf_keras_model.py
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
%cd '/content/drive/MyDrive/GSoC2023'
```

```
/content/drive/MyDrive/GSoC2023
```

```
import logging
logging.basicConfig(level=logging.INFO, format='[%(asctime)s] %
(levelname)s: %(message)s')
```

Dataset

The datasets are contained in twenty files with 100k jets each, and only the required files are downloaded. These are based on the samples used in 1810.05165. Each dataset consists of two components:

- X : a three-dimensional numpy array of the jets with shape (num_data,max_num_particles,4).
- y : a numpy array of quark/gluon jet labels (quark=1 and gluon=0).

The jets are padded with zero-particles in order to make a contiguous array. The particles are given as `(pt,y,phi,pid)` values, where `pid` is the particle's PDG id. Quark jets either include or exclude c and b quarks depending on the `with_bc` argument.

```
from __future__ import absolute_import, division, print_function

# standard numerical library imports
import numpy as np

# energyflow imports
import energyflow as ef
from energyflow.archs import EFN
from energyflow.datasets import qg_jets
```

```
from energyflow.utils import data_split, to_categorical

from sklearn.metrics import roc_auc_score, roc_curve
import matplotlib.pyplot as plt
```

```
train, val, test = 300000, 100000, 100000
```

```
# load data
X, y = qg_jets.load(train + val + test)
X = X[ :, :, :3]  #ignoring pid

# convert labels to categorical
y = to_categorical(y, num_classes = 2)
```

```
Downloading QG_jets.npz from
https://www.dropbox.com/s/fclsl7pukcpobsb/QG_jets.npz?dl=1 to
/root/.energyflow/datasets
Downloading QG_jets_1.npz from
https://www.dropbox.com/s/ztzd1a6lkmgovuy/QG_jets_1.npz?dl=1 to
/root/.energyflow/datasets
Downloading QG_jets_2.npz from
https://www.dropbox.com/s/jzgc9e786tbk1m5/QG_jets_2.npz?dl=1 to
/root/.energyflow/datasets
Downloading QG_jets_3.npz from
https://www.dropbox.com/s/tiwz2ck3wnzvlcr/QG_jets_3.npz?dl=1 to
/root/.energyflow/datasets
Downloading QG_jets_4.npz from
https://www.dropbox.com/s/3miwek1n0brbd2i/QG_jets_4.npz?dl=1 to
/root/.energyflow/datasets
```

```
from sklearn.model_selection import train_test_split

X_train_val, X_test, y_train_val, y_test = train_test_split(X, y,
test_size = 0.2, random_state = 42)
X_train, X_val, y_train, y_val = train_test_split(X_train_val,
y_train_val, test_size = 0.25, random_state = 42)
```

Architectures

For this task, I've considered the following Graph-based architectures,

- ParticleNet
- ParticleNet-Lite
- Energy Flow Networks (EFN)
- Particle Flow Networks (PFN)

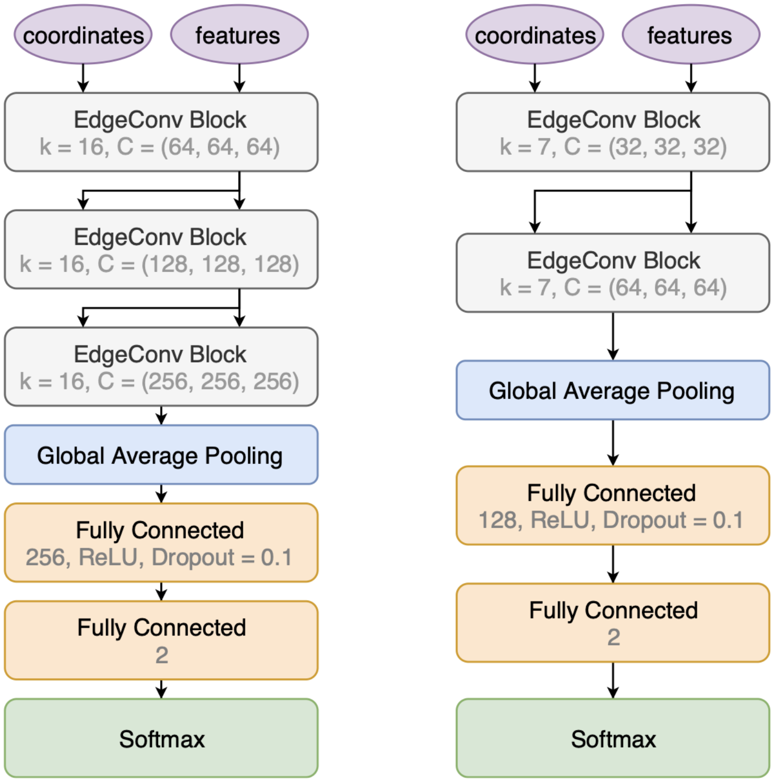
ParticleNet and ParticleNet-Lite have a lot in common when it comes to how they're built. They both use EdgeConv to help them understand the relationships between nearby points in a point-cloud, which makes them more effective than most other methods out there. The ParticleNet-Lite model provides a good balance between speed and performance.

EFN and PFN have a special internal representation for each particle, also known as "latent representation". When you add up the latent representation of all particles, it gives you an overall representation of the entire event.

Given the limited resources available, I have implemented ParticleNet-Lite and EFN. These models were chosen because they are more efficient in terms of resource utilization compared to other options available.

ParticleNet-Lite

The paper [2] introduces a novel deep-learning approach for jet tagging by utilizing a unique way of representing jets. Rather than arranging a jet's constituent particles in a structured format such as a sequence or a tree, the approach treats a jet as an unstructured set of particles. This is similar to the point cloud representation used in computer vision to represent 3D shapes, where a shape is represented as an unordered set of points in space. As such, a jet can be thought of as a "particle cloud".



(a) ParticleNet (b) ParticleNet-Lite

FIG. 2: The architectures of the ParticleNet and the ParticleNet-Lite networks.

```
dict_train = {
    'points'   : X_train[ :, :, 1:3],
    'features' : X_train,
    'mask'     : np.array(np.sum(X_train, axis = 2) != 0,
np.float32).reshape(X_train.shape[0], X_train.shape[1], 1)
}

dict_test = {
    'points'   : X_test[ :, :, 1:3],
    'features' : X_test,
    'mask'     : np.array(np.sum(X_test, axis = 2) != 0,
np.float32).reshape(X_test.shape[0], X_test.shape[1], 1)
}

dict_val = {
    'points'   : X_val[ :, :, 1:3],
    'features' : X_val,
    'mask'     : np.array(np.sum(X_val, axis = 2) != 0,
np.float32).reshape(X_val.shape[0], X_val.shape[1], 1)
}
```

```python
input_shapes = {
    'points'  : X_train[ :, :, 1:3].shape[1:],
    'features': X_train.shape[1:],
    'mask'    : np.array(np.sum(X_train, axis = 2) != 0,
np.float32).reshape(X_train.shape[0], X_train.shape[1], 1).shape[1:]
}

import tensorflow as tf
from tensorflow import keras
from tf_keras_model import get_particle_net, get_particle_net_lite

model_type = 'particle_net_lite' # choose between 'particle_net' and
'particle_net_lite'
num_classes = 2
if 'lite' in model_type:
    model = get_particle_net_lite(num_classes, input_shapes)
else:
    model = get_particle_net(num_classes, input_shapes)

# Training parameters
batch_size = 1024 if 'lite' in model_type else 384
epochs = 60

def lr_schedule(epoch):
    lr = 1e-3
    if epoch > 40:
        lr *= 0.1
    elif epoch > 50:
        lr *= 0.01
    logging.info('Learning rate: %f'%lr)
    return lr

model.compile(loss='categorical_crossentropy',

optimizer=keras.optimizers.Adam(learning_rate=lr_schedule(0)),
              metrics=['accuracy'])
model.summary()
```

```
Model: "ParticleNet"
______________________________________________________________________
____________________________
 Layer (type)                   Output Shape         Param #
Connected to
======================================================================
============================
 mask (InputLayer)              [(None, 139, 1)]     0           []



 tf.math.not_equal (TFOpLambda)  (None, 139, 1)      0
['mask[0][0]']
```

```
 tf.cast (TFOpLambda)           (None, 139, 1)       0
['tf.math.not_equal[0][0]']


 tf.math.equal (TFOpLambda)     (None, 139, 1)       0
['tf.cast[0][0]']


 tf.cast_1 (TFOpLambda)         (None, 139, 1)       0
['tf.math.equal[0][0]']


 tf.math.multiply (TFOpLambda)  (None, 139, 1)       0
['tf.cast_1[0][0]']


 points (InputLayer)            [(None, 139, 2)]     0           []



 tf.math.add (TFOpLambda)       (None, 139, 2)       0
['tf.math.multiply[0][0]',

'points[0][0]']


 features (InputLayer)          [(None, 139, 3)]     0           []



 tf.compat.v1.transpose (TFOpLa  (None, 2, 139)      0
['tf.math.add[0][0]']
 mbda)



 tf.expand_dims (TFOpLambda)    (None, 139, 1, 3)    0
['features[0][0]']


 tf.math.multiply_1 (TFOpLambda  (None, 139, 2)      0
['tf.math.add[0][0]',
 )
'tf.math.add[0][0]']


 tf.linalg.matmul (TFOpLambda)  (None, 139, 139)     0
```

```
['tf.math.add[0][0]',

'tf.compat.v1.transpose[0][0]']


 tf.math.multiply_2 (TFOpLambda  (None, 139, 2)      0
['tf.math.add[0][0]',
 )
'tf.math.add[0][0]']


 ParticleNet_fts_bn (BatchNorma  (None, 139, 1, 3)   12
['tf.expand_dims[0][0]']
 lization)



 tf.math.reduce_sum (TFOpLambda  (None, 139, 1)      0
['tf.math.multiply_1[0][0]']
 )



 tf.math.multiply_3 (TFOpLambda  (None, 139, 139)    0
['tf.linalg.matmul[0][0]']
 )



 tf.math.reduce_sum_1 (TFOpLamb  (None, 139, 1)      0
['tf.math.multiply_2[0][0]']
 da)



 tf.compat.v1.squeeze (TFOpLamb  (None, 139, 3)      0
['ParticleNet_fts_bn[0][0]']
 da)



 tf.math.subtract (TFOpLambda)  (None, 139, 139)     0
['tf.math.reduce_sum[0][0]',

'tf.math.multiply_3[0][0]']


 tf.compat.v1.transpose_1 (TFOp  (None, 1, 139)      0
['tf.math.reduce_sum_1[0][0]']
```

```
 Lambda)



 tf.compat.v1.shape (TFOpLambda  (3,)                 0
['tf.compat.v1.squeeze[0][0]']
 )



 tf.__operators__.add (TFOpLamb  (None, 139, 139)    0
['tf.math.subtract[0][0]',
 da)
'tf.compat.v1.transpose_1[0][0]'
                                                            ]



 tf.__operators__.getitem_1 (Sl  ()                  0
['tf.compat.v1.shape[0][0]']
 icingOpLambda)



 tf.math.negative (TFOpLambda)  (None, 139, 139)     0
['tf.__operators__.add[0][0]']


 tf.range (TFOpLambda)          (None,)              0
['tf.__operators__.getitem_1[0][0
                                                            ]']



 tf.math.top_k (TFOpLambda)     TopKV2(values=(None  0
['tf.math.negative[0][0]']
                                , 139, 8),

                                 indices=(None, 139

                                , 8))



 tf.reshape (TFOpLambda)        (None, 1, 1, 1)      0
['tf.range[0][0]']


 tf.__operators__.getitem (Slic  (None, 139, 7)      0
```

```
['tf.math.top_k[0][1]']
 ingOpLambda)



 tf.tile (TFOpLambda)           (None, 139, 7, 1)    0
['tf.reshape[0][0]']


 tf.expand_dims_1 (TFOpLambda)  (None, 139, 7, 1)    0
['tf.__operators__.getitem[0][0]'
                                                                  ]



 tf.expand_dims_2 (TFOpLambda)  (None, 139, 1, 3)    0
['tf.compat.v1.squeeze[0][0]']


 tf.concat (TFOpLambda)         (None, 139, 7, 2)    0
['tf.tile[0][0]',

'tf.expand_dims_1[0][0]']


 tf.tile_1 (TFOpLambda)         (None, 139, 7, 3)    0
['tf.expand_dims_2[0][0]']


 tf.compat.v1.gather_nd (TFOpLa  (None, 139, 7, 3)   0
['tf.compat.v1.squeeze[0][0]',
 mbda)
'tf.concat[0][0]']


 tf.math.subtract_1 (TFOpLambda  (None, 139, 7, 3)   0
['tf.compat.v1.gather_nd[0][0]',
 )
'tf.tile_1[0][0]']


 tf.concat_1 (TFOpLambda)       (None, 139, 7, 6)    0
['tf.tile_1[0][0]',

'tf.math.subtract_1[0][0]']


 ParticleNet_EdgeConv0_conv0 (C  (None, 139, 7, 32)  192
['tf.concat_1[0][0]']
```

```
 onv2D)



 ParticleNet_EdgeConv0_bn0 (Bat  (None, 139, 7, 32)  128
['ParticleNet_EdgeConv0_conv0[0][
 chNormalization)                                                 0]']



 ParticleNet_EdgeConv0_act0 (Ac  (None, 139, 7, 32)  0
['ParticleNet_EdgeConv0_bn0[0][0]
 tivation)                                                        ']



 ParticleNet_EdgeConv0_conv1 (C  (None, 139, 7, 32)  1024
['ParticleNet_EdgeConv0_act0[0][0
 onv2D)                                                           ]']



 ParticleNet_EdgeConv0_bn1 (Bat  (None, 139, 7, 32)  128
['ParticleNet_EdgeConv0_conv1[0][
 chNormalization)                                                 0]']



 ParticleNet_EdgeConv0_act1 (Ac  (None, 139, 7, 32)  0
['ParticleNet_EdgeConv0_bn1[0][0]
 tivation)                                                        ']



 tf.expand_dims_3 (TFOpLambda)  (None, 139, 1, 3)    0
['tf.compat.v1.squeeze[0][0]']


 ParticleNet_EdgeConv0_conv2 (C  (None, 139, 7, 32)  1024
['ParticleNet_EdgeConv0_act1[0][0
 onv2D)                                                           ]']



 ParticleNet_EdgeConv0_sc_conv   (None, 139, 1, 32)  96
['tf.expand_dims_3[0][0]']
 (Conv2D)
```

```
 ParticleNet_EdgeConv0_bn2 (Bat  (None, 139, 7, 32)  128
['ParticleNet_EdgeConv0_conv2[0][
 chNormalization)                                                0]']


 ParticleNet_EdgeConv0_sc_bn (B  (None, 139, 1, 32)  128
['ParticleNet_EdgeConv0_sc_conv[0
 atchNormalization)                                              ]
[0]']


 ParticleNet_EdgeConv0_act2 (Ac  (None, 139, 7, 32)  0
['ParticleNet_EdgeConv0_bn2[0][0
 tivation)                                                       ']


 tf.compat.v1.squeeze_1 (TFOpLa  (None, 139, 32)     0
['ParticleNet_EdgeConv0_sc_bn[0][
 mbda)                                                           0]']


 tf.math.reduce_mean (TFOpLambd  (None, 139, 32)     0
['ParticleNet_EdgeConv0_act2[0][0
 a)                                                              ]']


 tf.__operators__.add_1 (TFOpLa  (None, 139, 32)     0
['tf.compat.v1.squeeze_1[0][0]',
 mbda)
'tf.math.reduce_mean[0][0]']


 ParticleNet_EdgeConv0_sc_act (  (None, 139, 32)     0
['tf.__operators__.add_1[0][0]']
 Activation)


 tf.math.add_1 (TFOpLambda)     (None, 139, 32)      0
['tf.math.multiply[0][0]',

'ParticleNet_EdgeConv0_sc_act[0]
                                                                 [0]']
```

```
 tf.compat.v1.transpose_2 (TFOp  (None, 32, 139)     0
['tf.math.add_1[0][0]']
 Lambda)



 tf.math.multiply_4 (TFOpLambda  (None, 139, 32)     0
['tf.math.add_1[0][0]',
 )
'tf.math.add_1[0][0]']


 tf.linalg.matmul_1 (TFOpLambda  (None, 139, 139)    0
['tf.math.add_1[0][0]',
 )
'tf.compat.v1.transpose_2[0][0]'
                                                                    ]



 tf.math.multiply_5 (TFOpLambda  (None, 139, 32)     0
['tf.math.add_1[0][0]',
 )
'tf.math.add_1[0][0]']


 tf.math.reduce_sum_2 (TFOpLamb  (None, 139, 1)      0
['tf.math.multiply_4[0][0]']
 da)



 tf.math.multiply_6 (TFOpLambda  (None, 139, 139)    0
['tf.linalg.matmul_1[0][0]']
 )



 tf.math.reduce_sum_3 (TFOpLamb  (None, 139, 1)      0
['tf.math.multiply_5[0][0]']
 da)



 tf.math.subtract_2 (TFOpLambda  (None, 139, 139)    0
['tf.math.reduce_sum_2[0][0]',
 )
```

```
'tf.math.multiply_6[0][0]']


 tf.compat.v1.transpose_3 (TFOp  (None, 1, 139)      0
['tf.math.reduce_sum_3[0][0]']
 Lambda)



 tf.compat.v1.shape_1 (TFOpLamb  (3,)                0
['ParticleNet_EdgeConv0_sc_act[0]
 da)                                                                 [0]']



 tf.__operators__.add_2 (TFOpLa  (None, 139, 139)    0
['tf.math.subtract_2[0][0]',
 mbda)
'tf.compat.v1.transpose_3[0][0]'
                                                                     ]



 tf.__operators__.getitem_3 (Sl  ()                  0
['tf.compat.v1.shape_1[0][0]']
 icingOpLambda)



 tf.math.negative_1 (TFOpLambda  (None, 139, 139)    0
['tf.__operators__.add_2[0][0]']
 )



 tf.range_1 (TFOpLambda)        (None,)              0
['tf.__operators__.getitem_3[0][0
                                                                     ]']



 tf.math.top_k_1 (TFOpLambda)   TopKV2(values=(None  0
['tf.math.negative_1[0][0]']
                                , 139, 8),

                                 indices=(None, 139

                                , 8))
```

```
 tf.reshape_1 (TFOpLambda)      (None, 1, 1, 1)      0
['tf.range_1[0][0]']


 tf.__operators__.getitem_2 (Sl  (None, 139, 7)      0
['tf.math.top_k_1[0][1]']
 icingOpLambda)



 tf.tile_2 (TFOpLambda)         (None, 139, 7, 1)    0
['tf.reshape_1[0][0]']


 tf.expand_dims_4 (TFOpLambda)  (None, 139, 7, 1)    0
['tf.__operators__.getitem_2[0][0
                                                                 ]']



 tf.expand_dims_5 (TFOpLambda)  (None, 139, 1, 32)   0
['ParticleNet_EdgeConv0_sc_act[0]
                                                                 [0]']



 tf.concat_2 (TFOpLambda)       (None, 139, 7, 2)    0
['tf.tile_2[0][0]',

'tf.expand_dims_4[0][0]']


 tf.tile_3 (TFOpLambda)         (None, 139, 7, 32)   0
['tf.expand_dims_5[0][0]']


 tf.compat.v1.gather_nd_1 (TFOp  (None, 139, 7, 32)  0
['ParticleNet_EdgeConv0_sc_act[0]
 Lambda)                                                         [0]',


'tf.concat_2[0][0]']


 tf.math.subtract_3 (TFOpLambda  (None, 139, 7, 32)  0
['tf.compat.v1.gather_nd_1[0][0]'
 )                                                               ,
```

```
'tf.tile_3[0][0]']


 tf.concat_3 (TFOpLambda)       (None, 139, 7, 64)   0
['tf.tile_3[0][0]',

'tf.math.subtract_3[0][0]']


 ParticleNet_EdgeConv1_conv0 (C  (None, 139, 7, 64)  4096
['tf.concat_3[0][0]']
 onv2D)



 ParticleNet_EdgeConv1_bn0 (Bat  (None, 139, 7, 64)  256
['ParticleNet_EdgeConv1_conv0[0][
 chNormalization)                                                0]']



 ParticleNet_EdgeConv1_act0 (Ac  (None, 139, 7, 64)  0
['ParticleNet_EdgeConv1_bn0[0][0]
 tivation)                                                       ']



 ParticleNet_EdgeConv1_conv1 (C  (None, 139, 7, 64)  4096
['ParticleNet_EdgeConv1_act0[0][0
 onv2D)                                                          ]']



 ParticleNet_EdgeConv1_bn1 (Bat  (None, 139, 7, 64)  256
['ParticleNet_EdgeConv1_conv1[0][
 chNormalization)                                                0]']



 ParticleNet_EdgeConv1_act1 (Ac  (None, 139, 7, 64)  0
['ParticleNet_EdgeConv1_bn1[0][0]
 tivation)                                                       ']



 tf.expand_dims_6 (TFOpLambda)  (None, 139, 1, 32)   0
['ParticleNet_EdgeConv0_sc_act[0]
                                                                 [0]']
```

```
 ParticleNet_EdgeConv1_conv2 (C  (None, 139, 7, 64)  4096
['ParticleNet_EdgeConv1_act1[0][0
 onv2D)                                                          ]']


 ParticleNet_EdgeConv1_sc_conv   (None, 139, 1, 64)  2048
['tf.expand_dims_6[0][0]']
 (Conv2D)


 ParticleNet_EdgeConv1_bn2 (Bat  (None, 139, 7, 64)  256
['ParticleNet_EdgeConv1_conv2[0][
 chNormalization)                                                0]']


 ParticleNet_EdgeConv1_sc_bn (B  (None, 139, 1, 64)  256
['ParticleNet_EdgeConv1_sc_conv[0
 atchNormalization)                                              ]
[0]']

 ParticleNet_EdgeConv1_act2 (Ac  (None, 139, 7, 64)  0
['ParticleNet_EdgeConv1_bn2[0][0]
 tivation)                                                       ']


 tf.compat.v1.squeeze_2 (TFOpLa  (None, 139, 64)     0
['ParticleNet_EdgeConv1_sc_bn[0][
 mbda)                                                           0]']


 tf.math.reduce_mean_1 (TFOpLam  (None, 139, 64)     0
['ParticleNet_EdgeConv1_act2[0][0
 bda)                                                            ]']


 tf.__operators__.add_3 (TFOpLa  (None, 139, 64)     0
['tf.compat.v1.squeeze_2[0][0]',
 mbda)
'tf.math.reduce_mean_1[0][0]']
```

```
 ParticleNet_EdgeConv1_sc_act (  (None, 139, 64)     0
['tf.__operators__.add_3[0][0]']
 Activation)



 tf.math.multiply_7 (TFOpLambda  (None, 139, 64)     0
['ParticleNet_EdgeConv1_sc_act[0]
 )                                                            [0]',


'tf.cast[0][0]']


 tf.math.reduce_mean_2 (TFOpLam  (None, 64)          0
['tf.math.multiply_7[0][0]']
 bda)



 dense (Dense)                  (None, 128)          8320
['tf.math.reduce_mean_2[0][0]']


 dropout (Dropout)              (None, 128)          0
['dense[0][0]']


 dense_1 (Dense)                (None, 2)            258
['dropout[0][0]']


======================================================================
============================
Total params: 26,798
Trainable params: 26,024
Non-trainable params: 774
______________________________________________________________________
____________________________
```

```python
# Prepare model model saving directory.
import os
save_dir = 'model_checkpoints'
model_name = '%s_model.{epoch:03d}.h5' % model_type
if not os.path.isdir(save_dir):
    os.makedirs(save_dir)
filepath = os.path.join(save_dir, model_name)
```

```
# Prepare callbacks for model saving and for learning rate adjustment.
checkpoint = keras.callbacks.ModelCheckpoint(filepath = filepath,
                             monitor = 'val_accuracy',
                             verbose = 1,
                             save_best_only = True)

lr_scheduler = keras.callbacks.LearningRateScheduler(lr_schedule)
progress_bar = keras.callbacks.ProgbarLogger()
callbacks = [checkpoint, lr_scheduler, progress_bar]

model.fit(dict_train, y_train,
          batch_size = batch_size,
          epochs = epochs,
          # epochs=1, # --- train only for 1 epoch here for
demonstration ---
          validation_data = (dict_val, y_val),
          shuffle = True,
          callbacks = callbacks)
```

```
Epoch 1/60
      0/Unknown - 182s 0s/sample - loss: 0.4965 - accuracy: 0.7661
Epoch 1: val_accuracy improved from -inf to 0.77371, saving model to
model_checkpoints/particle_net_lite_model.001.h5
293/293 [==============================] - 266s 909ms/sample - loss:
0.4965 - accuracy: 0.7661 - val_loss: 0.4867 - val_accuracy: 0.7737 -
lr: 0.0010
Epoch 2/60
  0/293 [..............................] - ETA: 0s - loss: 0.4702 -
accuracy: 0.7849
Epoch 2: val_accuracy did not improve from 0.77371
293/293 [==============================] - 247s 843ms/sample - loss:
0.4702 - accuracy: 0.7849 - val_loss: 0.4876 - val_accuracy: 0.7726 -
lr: 0.0010
Epoch 3/60
  0/293 [..............................] - ETA: 0s - loss: 0.4603 -
accuracy: 0.7902
Epoch 3: val_accuracy improved from 0.77371 to 0.78893, saving model
to model_checkpoints/particle_net_lite_model.003.h5
293/293 [==============================] - 248s 846ms/sample - loss:
0.4603 - accuracy: 0.7902 - val_loss: 0.4618 - val_accuracy: 0.7889 -
lr: 0.0010
Epoch 4/60
  0/293 [..............................] - ETA: 0s - loss: 0.4560 -
accuracy: 0.7928
Epoch 4: val_accuracy improved from 0.78893 to 0.79155, saving model
to model_checkpoints/particle_net_lite_model.004.h5
293/293 [==============================] - 248s 845ms/sample - loss:
0.4560 - accuracy: 0.7928 - val_loss: 0.4563 - val_accuracy: 0.7915 -
lr: 0.0010
Epoch 5/60
```

```
  0/293 [..............................] - ETA: 0s - loss: 0.4520 -
accuracy: 0.7944
Epoch 5: val_accuracy did not improve from 0.79155
293/293 [==============================] - 247s 842ms/sample - loss:
0.4520 - accuracy: 0.7944 - val_loss: 0.4637 - val_accuracy: 0.7836 -
lr: 0.0010
Epoch 6/60
  0/293 [..............................] - ETA: 0s - loss: 0.4477 -
accuracy: 0.7973
Epoch 6: val_accuracy did not improve from 0.79155
293/293 [==============================] - 247s 843ms/sample - loss:
0.4477 - accuracy: 0.7973 - val_loss: 0.4873 - val_accuracy: 0.7705 -
lr: 0.0010
Epoch 7/60
  0/293 [..............................] - ETA: 0s - loss: 0.4451 -
accuracy: 0.7983
Epoch 7: val_accuracy improved from 0.79155 to 0.79410, saving model
to model_checkpoints/particle_net_lite_model.007.h5
293/293 [==============================] - 248s 846ms/sample - loss:
0.4451 - accuracy: 0.7983 - val_loss: 0.4535 - val_accuracy: 0.7941 -
lr: 0.0010
Epoch 8/60
  0/293 [..............................] - ETA: 0s - loss: 0.4426 -
accuracy: 0.7996
Epoch 8: val_accuracy improved from 0.79410 to 0.79765, saving model
to model_checkpoints/particle_net_lite_model.008.h5
293/293 [==============================] - 248s 846ms/sample - loss:
0.4426 - accuracy: 0.7996 - val_loss: 0.4444 - val_accuracy: 0.7976 -
lr: 0.0010
Epoch 9/60
  0/293 [..............................] - ETA: 0s - loss: 0.4409 -
accuracy: 0.8004
Epoch 9: val_accuracy did not improve from 0.79765
293/293 [==============================] - 206s 704ms/sample - loss:
0.4409 - accuracy: 0.8004 - val_loss: 0.4463 - val_accuracy: 0.7971 -
lr: 0.0010
Epoch 10/60
  0/293 [..............................] - ETA: 0s - loss: 0.4387 -
accuracy: 0.8024
Epoch 10: val_accuracy did not improve from 0.79765
293/293 [==============================] - 206s 704ms/sample - loss:
0.4387 - accuracy: 0.8024 - val_loss: 0.4461 - val_accuracy: 0.7962 -
lr: 0.0010
Epoch 11/60
  0/293 [..............................] - ETA: 0s - loss: 0.4374 -
accuracy: 0.8030
Epoch 11: val_accuracy improved from 0.79765 to 0.79945, saving model
to model_checkpoints/particle_net_lite_model.011.h5
293/293 [==============================] - 207s 708ms/sample - loss:
0.4374 - accuracy: 0.8030 - val_loss: 0.4421 - val_accuracy: 0.7994 -
```

```
lr: 0.0010
Epoch 12/60
  0/293 [..............................] - ETA: 0s - loss: 0.4363 -
accuracy: 0.8032
Epoch 12: val_accuracy improved from 0.79945 to 0.80232, saving model
to model_checkpoints/particle_net_lite_model.012.h5
293/293 [==============================] - 247s 843ms/sample - loss:
0.4363 - accuracy: 0.8032 - val_loss: 0.4398 - val_accuracy: 0.8023 -
lr: 0.0010
Epoch 13/60
  0/293 [..............................] - ETA: 0s - loss: 0.4358 -
accuracy: 0.8040
Epoch 13: val_accuracy did not improve from 0.80232
293/293 [==============================] - 207s 705ms/sample - loss:
0.4358 - accuracy: 0.8040 - val_loss: 0.4389 - val_accuracy: 0.8019 -
lr: 0.0010
Epoch 14/60
  0/293 [..............................] - ETA: 0s - loss: 0.4349 -
accuracy: 0.8048
Epoch 14: val_accuracy did not improve from 0.80232
293/293 [==============================] - 246s 841ms/sample - loss:
0.4349 - accuracy: 0.8048 - val_loss: 0.4447 - val_accuracy: 0.7987 -
lr: 0.0010
Epoch 15/60
  0/293 [..............................] - ETA: 0s - loss: 0.4343 -
accuracy: 0.8051
Epoch 15: val_accuracy did not improve from 0.80232
293/293 [==============================] - 247s 842ms/sample - loss:
0.4343 - accuracy: 0.8051 - val_loss: 0.4437 - val_accuracy: 0.8017 -
lr: 0.0010
Epoch 16/60
  0/293 [..............................] - ETA: 0s - loss: 0.4334 -
accuracy: 0.8052
Epoch 16: val_accuracy did not improve from 0.80232
293/293 [==============================] - 247s 843ms/sample - loss:
0.4334 - accuracy: 0.8052 - val_loss: 0.4488 - val_accuracy: 0.7956 -
lr: 0.0010
Epoch 17/60
  0/293 [..............................] - ETA: 0s - loss: 0.4326 -
accuracy: 0.8058
Epoch 17: val_accuracy did not improve from 0.80232
293/293 [==============================] - 247s 843ms/sample - loss:
0.4326 - accuracy: 0.8058 - val_loss: 0.4376 - val_accuracy: 0.8018 -
lr: 0.0010
Epoch 18/60
  0/293 [..............................] - ETA: 0s - loss: 0.4318 -
accuracy: 0.8063
Epoch 18: val_accuracy improved from 0.80232 to 0.80422, saving model
to model_checkpoints/particle_net_lite_model.018.h5
293/293 [==============================] - 247s 843ms/sample - loss:
```

```
0.4318 - accuracy: 0.8063 - val_loss: 0.4357 - val_accuracy: 0.8042 -
lr: 0.0010
Epoch 19/60
  0/293 [..............................] - ETA: 0s - loss: 0.4316 -
accuracy: 0.8067
Epoch 19: val_accuracy did not improve from 0.80422
293/293 [==============================] - 247s 842ms/sample - loss:
0.4316 - accuracy: 0.8067 - val_loss: 0.4366 - val_accuracy: 0.8030 -
lr: 0.0010
Epoch 20/60
  0/293 [..............................] - ETA: 0s - loss: 0.4310 -
accuracy: 0.8067
Epoch 20: val_accuracy did not improve from 0.80422
293/293 [==============================] - 247s 842ms/sample - loss:
0.4310 - accuracy: 0.8067 - val_loss: 0.4391 - val_accuracy: 0.8023 -
lr: 0.0010
Epoch 21/60
  0/293 [..............................] - ETA: 0s - loss: 0.4306 -
accuracy: 0.8072
Epoch 21: val_accuracy did not improve from 0.80422
293/293 [==============================] - 247s 842ms/sample - loss:
0.4306 - accuracy: 0.8072 - val_loss: 0.4378 - val_accuracy: 0.8027 -
lr: 0.0010
Epoch 22/60
  0/293 [..............................] - ETA: 0s - loss: 0.4311 -
accuracy: 0.8071
Epoch 22: val_accuracy did not improve from 0.80422
293/293 [==============================] - 247s 843ms/sample - loss:
0.4311 - accuracy: 0.8071 - val_loss: 0.4397 - val_accuracy: 0.8003 -
lr: 0.0010
Epoch 23/60
  0/293 [..............................] - ETA: 0s - loss: 0.4293 -
accuracy: 0.8081
Epoch 23: val_accuracy did not improve from 0.80422
293/293 [==============================] - 247s 843ms/sample - loss:
0.4293 - accuracy: 0.8081 - val_loss: 0.4724 - val_accuracy: 0.7775 -
lr: 0.0010
Epoch 24/60
  0/293 [..............................] - ETA: 0s - loss: 0.4296 -
accuracy: 0.8082
Epoch 24: val_accuracy did not improve from 0.80422
293/293 [==============================] - 206s 704ms/sample - loss:
0.4296 - accuracy: 0.8082 - val_loss: 0.4662 - val_accuracy: 0.7852 -
lr: 0.0010
Epoch 25/60
  0/293 [..............................] - ETA: 0s - loss: 0.4292 -
accuracy: 0.8084
Epoch 25: val_accuracy did not improve from 0.80422
293/293 [==============================] - 247s 841ms/sample - loss:
0.4292 - accuracy: 0.8084 - val_loss: 0.4430 - val_accuracy: 0.7994 -
```

```
lr: 0.0010
Epoch 26/60
  0/293 [..............................] - ETA: 0s - loss: 0.4280 -
accuracy: 0.8090
Epoch 26: val_accuracy did not improve from 0.80422
293/293 [==============================] - 247s 842ms/sample - loss:
0.4280 - accuracy: 0.8090 - val_loss: 0.4407 - val_accuracy: 0.7996 -
lr: 0.0010
Epoch 27/60
  0/293 [..............................] - ETA: 0s - loss: 0.4287 -
accuracy: 0.8080
Epoch 27: val_accuracy did not improve from 0.80422
293/293 [==============================] - 207s 705ms/sample - loss:
0.4287 - accuracy: 0.8080 - val_loss: 0.4384 - val_accuracy: 0.8013 -
lr: 0.0010
Epoch 28/60
  0/293 [..............................] - ETA: 0s - loss: 0.4274 -
accuracy: 0.8089
Epoch 28: val_accuracy improved from 0.80422 to 0.80508, saving model
to model_checkpoints/particle_net_lite_model.028.h5
293/293 [==============================] - 206s 704ms/sample - loss:
0.4274 - accuracy: 0.8089 - val_loss: 0.4332 - val_accuracy: 0.8051 -
lr: 0.0010
Epoch 29/60
  0/293 [..............................] - ETA: 0s - loss: 0.4274 -
accuracy: 0.8091
Epoch 29: val_accuracy did not improve from 0.80508
293/293 [==============================] - 247s 842ms/sample - loss:
0.4274 - accuracy: 0.8091 - val_loss: 0.4384 - val_accuracy: 0.8040 -
lr: 0.0010
Epoch 30/60
  0/293 [..............................] - ETA: 0s - loss: 0.4272 -
accuracy: 0.8092
Epoch 30: val_accuracy did not improve from 0.80508
293/293 [==============================] - 207s 705ms/sample - loss:
0.4272 - accuracy: 0.8092 - val_loss: 0.4361 - val_accuracy: 0.8048 -
lr: 0.0010
Epoch 31/60
  0/293 [..............................] - ETA: 0s - loss: 0.4270 -
accuracy: 0.8093
Epoch 31: val_accuracy improved from 0.80508 to 0.80588, saving model
to model_checkpoints/particle_net_lite_model.031.h5
293/293 [==============================] - 207s 705ms/sample - loss:
0.4270 - accuracy: 0.8093 - val_loss: 0.4322 - val_accuracy: 0.8059 -
lr: 0.0010
Epoch 32/60
  0/293 [..............................] - ETA: 0s - loss: 0.4266 -
accuracy: 0.8096
Epoch 32: val_accuracy improved from 0.80588 to 0.80607, saving model
to model_checkpoints/particle_net_lite_model.032.h5
```

```
293/293 [==============================] - 247s 843ms/sample - loss:
0.4266 - accuracy: 0.8096 - val_loss: 0.4316 - val_accuracy: 0.8061 -
lr: 0.0010
Epoch 33/60
  0/293 [..............................] - ETA: 0s - loss: 0.4263 -
accuracy: 0.8097
Epoch 33: val_accuracy did not improve from 0.80607
293/293 [==============================] - 247s 843ms/sample - loss:
0.4263 - accuracy: 0.8097 - val_loss: 0.4427 - val_accuracy: 0.8007 -
lr: 0.0010
Epoch 34/60
  0/293 [..............................] - ETA: 0s - loss: 0.4261 -
accuracy: 0.8096
Epoch 34: val_accuracy did not improve from 0.80607
293/293 [==============================] - 247s 843ms/sample - loss:
0.4261 - accuracy: 0.8096 - val_loss: 0.4426 - val_accuracy: 0.8010 -
lr: 0.0010
Epoch 35/60
  0/293 [..............................] - ETA: 0s - loss: 0.4255 -
accuracy: 0.8103
Epoch 35: val_accuracy improved from 0.80607 to 0.80661, saving model
to model_checkpoints/particle_net_lite_model.035.h5
293/293 [==============================] - 247s 843ms/sample - loss:
0.4255 - accuracy: 0.8103 - val_loss: 0.4309 - val_accuracy: 0.8066 -
lr: 0.0010
Epoch 36/60
  0/293 [..............................] - ETA: 0s - loss: 0.4252 -
accuracy: 0.8105
Epoch 36: val_accuracy did not improve from 0.80661
293/293 [==============================] - 247s 843ms/sample - loss:
0.4252 - accuracy: 0.8105 - val_loss: 0.4305 - val_accuracy: 0.8063 -
lr: 0.0010
Epoch 37/60
  0/293 [..............................] - ETA: 0s - loss: 0.4247 -
accuracy: 0.8104
Epoch 37: val_accuracy did not improve from 0.80661
293/293 [==============================] - 247s 843ms/sample - loss:
0.4247 - accuracy: 0.8104 - val_loss: 0.4316 - val_accuracy: 0.8062 -
lr: 0.0010
Epoch 38/60
  0/293 [..............................] - ETA: 0s - loss: 0.4243 -
accuracy: 0.8111
Epoch 38: val_accuracy did not improve from 0.80661
293/293 [==============================] - 247s 843ms/sample - loss:
0.4243 - accuracy: 0.8111 - val_loss: 0.4329 - val_accuracy: 0.8061 -
lr: 0.0010
Epoch 39/60
  0/293 [..............................] - ETA: 0s - loss: 0.4241 -
accuracy: 0.8109
Epoch 39: val_accuracy did not improve from 0.80661
```

```
293/293 [==============================] - 247s 844ms/sample - loss:
0.4241 - accuracy: 0.8109 - val_loss: 0.4321 - val_accuracy: 0.8058 -
lr: 0.0010
Epoch 40/60
  0/293 [..............................] - ETA: 0s - loss: 0.4234 -
accuracy: 0.8112
Epoch 40: val_accuracy did not improve from 0.80661
293/293 [==============================] - 247s 843ms/sample - loss:
0.4234 - accuracy: 0.8112 - val_loss: 0.4341 - val_accuracy: 0.8050 -
lr: 0.0010
Epoch 41/60
  0/293 [..............................] - ETA: 0s - loss: 0.4232 -
accuracy: 0.8121
Epoch 41: val_accuracy improved from 0.80661 to 0.80790, saving model
to model_checkpoints/particle_net_lite_model.041.h5
293/293 [==============================] - 247s 844ms/sample - loss:
0.4232 - accuracy: 0.8121 - val_loss: 0.4295 - val_accuracy: 0.8079 -
lr: 0.0010
Epoch 42/60
  0/293 [..............................] - ETA: 0s - loss: 0.4195 -
accuracy: 0.8136
Epoch 42: val_accuracy improved from 0.80790 to 0.80825, saving model
to model_checkpoints/particle_net_lite_model.042.h5
293/293 [==============================] - 247s 844ms/sample - loss:
0.4195 - accuracy: 0.8136 - val_loss: 0.4279 - val_accuracy: 0.8083 -
lr: 1.0000e-04
Epoch 43/60
  0/293 [..............................] - ETA: 0s - loss: 0.4189 -
accuracy: 0.8140
Epoch 43: val_accuracy did not improve from 0.80825
293/293 [==============================] - 247s 843ms/sample - loss:
0.4189 - accuracy: 0.8140 - val_loss: 0.4279 - val_accuracy: 0.8079 -
lr: 1.0000e-04
Epoch 44/60
  0/293 [..............................] - ETA: 0s - loss: 0.4186 -
accuracy: 0.8144
Epoch 44: val_accuracy did not improve from 0.80825
293/293 [==============================] - 247s 842ms/sample - loss:
0.4186 - accuracy: 0.8144 - val_loss: 0.4280 - val_accuracy: 0.8081 -
lr: 1.0000e-04
Epoch 45/60
  0/293 [..............................] - ETA: 0s - loss: 0.4181 -
accuracy: 0.8145
Epoch 45: val_accuracy did not improve from 0.80825
293/293 [==============================] - 247s 843ms/sample - loss:
0.4181 - accuracy: 0.8145 - val_loss: 0.4282 - val_accuracy: 0.8080 -
lr: 1.0000e-04
Epoch 46/60
  0/293 [..............................] - ETA: 0s - loss: 0.4182 -
accuracy: 0.8146
```

```
Epoch 46: val_accuracy did not improve from 0.80825
293/293 [==============================] - 207s 706ms/sample - loss:
0.4182 - accuracy: 0.8146 - val_loss: 0.4286 - val_accuracy: 0.8075 -
lr: 1.0000e-04
Epoch 47/60
  0/293 [..............................] - ETA: 0s - loss: 0.4180 -
accuracy: 0.8146
Epoch 47: val_accuracy did not improve from 0.80825
293/293 [==============================] - 247s 842ms/sample - loss:
0.4180 - accuracy: 0.8146 - val_loss: 0.4280 - val_accuracy: 0.8081 -
lr: 1.0000e-04
Epoch 48/60
  0/293 [..............................] - ETA: 0s - loss: 0.4181 -
accuracy: 0.8141
Epoch 48: val_accuracy improved from 0.80825 to 0.80873, saving model
to model_checkpoints/particle_net_lite_model.048.h5
293/293 [==============================] - 247s 844ms/sample - loss:
0.4181 - accuracy: 0.8141 - val_loss: 0.4275 - val_accuracy: 0.8087 -
lr: 1.0000e-04
Epoch 49/60
  0/293 [..............................] - ETA: 0s - loss: 0.4179 -
accuracy: 0.8146
Epoch 49: val_accuracy did not improve from 0.80873
293/293 [==============================] - 247s 843ms/sample - loss:
0.4179 - accuracy: 0.8146 - val_loss: 0.4274 - val_accuracy: 0.8087 -
lr: 1.0000e-04
Epoch 50/60
  0/293 [..............................] - ETA: 0s - loss: 0.4178 -
accuracy: 0.8146
Epoch 50: val_accuracy did not improve from 0.80873
293/293 [==============================] - 206s 704ms/sample - loss:
0.4178 - accuracy: 0.8146 - val_loss: 0.4276 - val_accuracy: 0.8085 -
lr: 1.0000e-04
Epoch 51/60
  0/293 [..............................] - ETA: 0s - loss: 0.4177 -
accuracy: 0.8151
Epoch 51: val_accuracy did not improve from 0.80873
293/293 [==============================] - 206s 705ms/sample - loss:
0.4177 - accuracy: 0.8151 - val_loss: 0.4285 - val_accuracy: 0.8080 -
lr: 1.0000e-04
Epoch 52/60
  0/293 [..............................] - ETA: 0s - loss: 0.4176 -
accuracy: 0.8149
Epoch 52: val_accuracy did not improve from 0.80873
293/293 [==============================] - 247s 842ms/sample - loss:
0.4176 - accuracy: 0.8149 - val_loss: 0.4279 - val_accuracy: 0.8085 -
lr: 1.0000e-04
Epoch 53/60
  0/293 [..............................] - ETA: 0s - loss: 0.4176 -
accuracy: 0.8150
```

```
Epoch 53: val_accuracy did not improve from 0.80873
293/293 [==============================] - 247s 843ms/sample - loss:
0.4176 - accuracy: 0.8150 - val_loss: 0.4277 - val_accuracy: 0.8081 -
lr: 1.0000e-04
Epoch 54/60
  0/293 [..............................] - ETA: 0s - loss: 0.4174 -
accuracy: 0.8150
Epoch 54: val_accuracy did not improve from 0.80873
293/293 [==============================] - 247s 842ms/sample - loss:
0.4174 - accuracy: 0.8150 - val_loss: 0.4276 - val_accuracy: 0.8085 -
lr: 1.0000e-04
Epoch 55/60
  0/293 [..............................] - ETA: 0s - loss: 0.4173 -
accuracy: 0.8149
Epoch 55: val_accuracy did not improve from 0.80873
293/293 [==============================] - 206s 705ms/sample - loss:
0.4173 - accuracy: 0.8149 - val_loss: 0.4278 - val_accuracy: 0.8087 -
lr: 1.0000e-04
Epoch 56/60
  0/293 [..............................] - ETA: 0s - loss: 0.4171 -
accuracy: 0.8154
Epoch 56: val_accuracy did not improve from 0.80873
293/293 [==============================] - 247s 841ms/sample - loss:
0.4171 - accuracy: 0.8154 - val_loss: 0.4278 - val_accuracy: 0.8087 -
lr: 1.0000e-04
Epoch 57/60
  0/293 [..............................] - ETA: 0s - loss: 0.4172 -
accuracy: 0.8150
Epoch 57: val_accuracy did not improve from 0.80873
293/293 [==============================] - 247s 843ms/sample - loss:
0.4172 - accuracy: 0.8150 - val_loss: 0.4273 - val_accuracy: 0.8087 -
lr: 1.0000e-04
Epoch 58/60
  0/293 [..............................] - ETA: 0s - loss: 0.4172 -
accuracy: 0.8149
Epoch 58: val_accuracy did not improve from 0.80873
293/293 [==============================] - 247s 843ms/sample - loss:
0.4172 - accuracy: 0.8149 - val_loss: 0.4281 - val_accuracy: 0.8084 -
lr: 1.0000e-04
Epoch 59/60
  0/293 [..............................] - ETA: 0s - loss: 0.4169 -
accuracy: 0.8152
Epoch 59: val_accuracy did not improve from 0.80873
293/293 [==============================] - 247s 843ms/sample - loss:
0.4169 - accuracy: 0.8152 - val_loss: 0.4278 - val_accuracy: 0.8086 -
lr: 1.0000e-04
Epoch 60/60
  0/293 [..............................] - ETA: 0s - loss: 0.4168 -
accuracy: 0.8155
Epoch 60: val_accuracy did not improve from 0.80873
```

```
293/293 [==============================] - 207s 705ms/sample - loss:
0.4168 - accuracy: 0.8155 - val_loss: 0.4277 - val_accuracy: 0.8086 -
lr: 1.0000e-04
```

```
<keras.callbacks.History at 0x7f70c03d1fd0>
```

```python
from keras.models import load_model

model = load_model('model_checkpoints/particle_net_lite_model.048.h5')

test_loss, test_accuracy = model.evaluate(dict_test, y_test)

print("Loss: ", test_loss)
print("Accuracy:", test_accuracy)
```

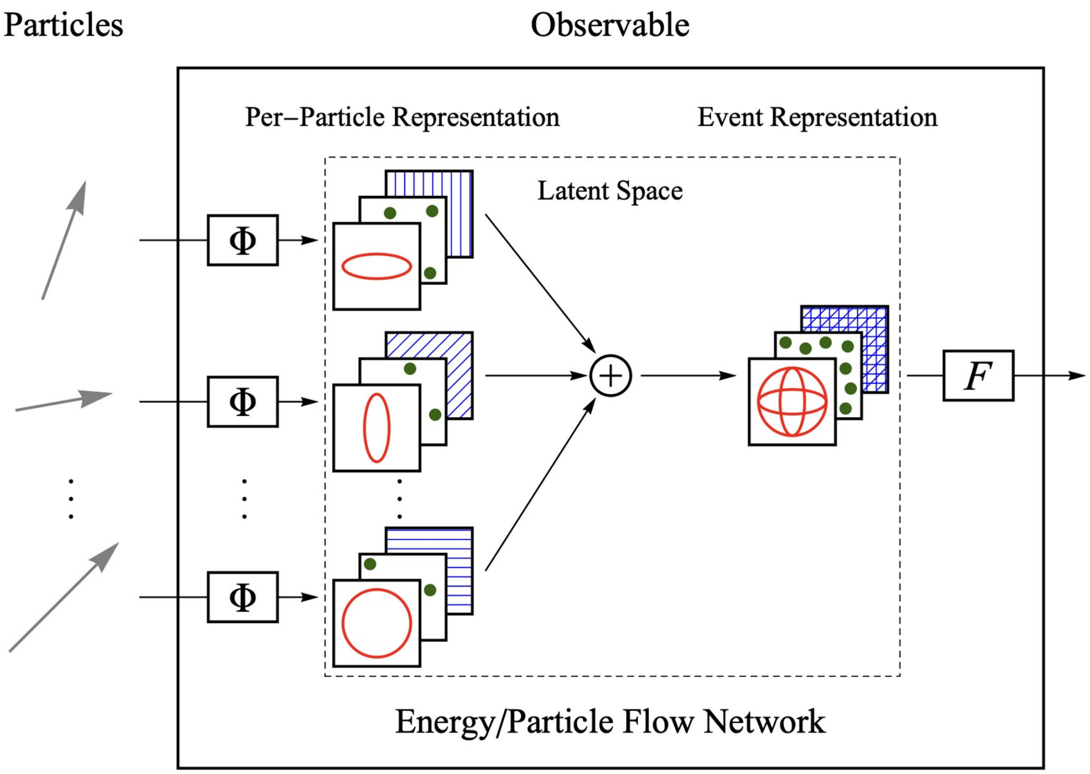
```
3125/3125 [==============================] - 138s 44ms/step - loss:
0.4268 - accuracy: 0.8094
Loss:  0.4267784059047699
Accuracy: 0.8093799948692322
```

```python
train_loss, train_accuracy = model.evaluate(dict_train, y_train)

print("Loss: ", train_loss)
print("Accuracy: ", train_accuracy)
```

```
9375/9375 [==============================] - 423s 45ms/step - loss:
0.4165 - accuracy: 0.8154
Loss:  0.4165427088737488
Accuracy:  0.8154066801071167
```

```python
val_loss, val_accuracy = model.evaluate(dict_val, y_val)

print("Loss: ", val_loss)
print("Accuracy: ", val_accuracy)
```

```
3125/3125 [==============================] - 139s 44ms/step - loss:
0.4275 - accuracy: 0.8087
Loss:  0.4274614453315735
Accuracy:  0.8087300062179565
```



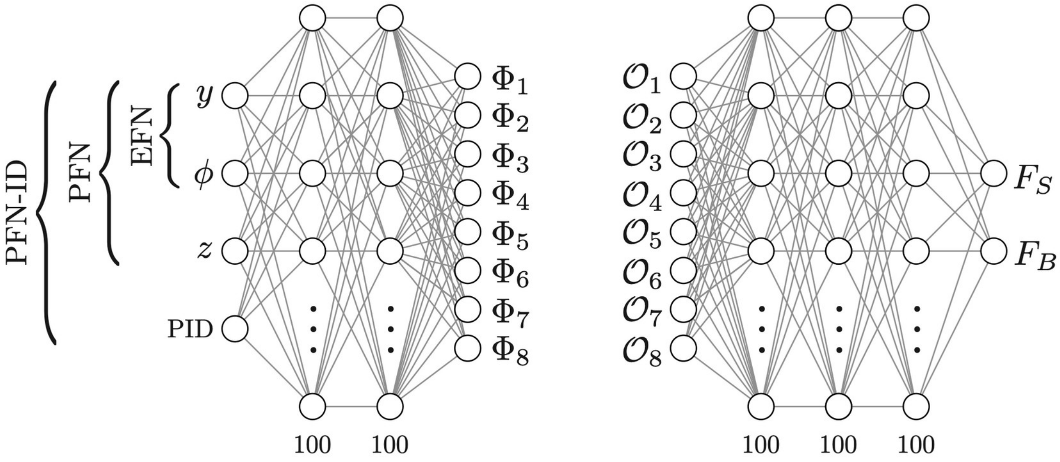
EFN

```
# network architecture parameters
Phi_sizes, F_sizes = (100, 100, 128), (100, 100, 100)
# Phi_sizes, F_sizes = (100, 100, 256), (100, 100, 100)

# network training parameters
num_epoch = 100
batch_size = 256

print('Model summary:')

# build architecture
efn = EFN(input_dim = 2, Phi_sizes = Phi_sizes, F_sizes = F_sizes)
```

```
Model summary:
Model: "model"
______________________________________________________________________
____________________________
 Layer (type)                   Output Shape         Param #
Connected to
======================================================================
============================
 phats_input (InputLayer)       [(None, None, 2)]    0           []



 tdist_0 (TimeDistributed)      (None, None, 100)    300
```

```
['phats_input[0][0]']


 activation (Activation)        (None, None, 100)    0
['tdist_0[0][0]']


 tdist_1 (TimeDistributed)      (None, None, 100)    10100
['activation[0][0]']


 activation_1 (Activation)      (None, None, 100)    0
['tdist_1[0][0]']


 zs_input (InputLayer)          [(None, None)]       0           []



 tdist_2 (TimeDistributed)      (None, None, 128)    12928
['activation_1[0][0]']


 mask (Lambda)                  (None, None)         0
['zs_input[0][0]']


 activation_2 (Activation)      (None, None, 128)    0
['tdist_2[0][0]']


 sum (Dot)                      (None, 128)          0
['mask[0][0]',

'activation_2[0][0]']


 dense_0 (Dense)                (None, 100)          12900
['sum[0][0]']


 activation_3 (Activation)      (None, 100)          0
['dense_0[0][0]']


 dense_1 (Dense)                (None, 100)          10100
['activation_3[0][0]']
```

```
 activation_4 (Activation)      (None, 100)          0
['dense_1[0][0]']


 dense_2 (Dense)                (None, 100)          10100
['activation_4[0][0]']


 activation_5 (Activation)      (None, 100)          0
['dense_2[0][0]']


 output (Dense)                 (None, 2)            202
['activation_5[0][0]']


 activation_6 (Activation)      (None, 2)            0
['output[0][0]']


======================================================================
============================
Total params: 56,630
Trainable params: 56,630
Non-trainable params: 0
______________________________________________________________________
____________________________
```



(a) (b)

Figure 4: The particular dense networks used here to parametrize (a) the per-particle mapping Φ and (b) the function F, shown for the case of a latent space of dimension $\ell = 8$. For the EFN, the latent observable is $\mathcal{O}_a = \sum_i z_i \, \Phi_a(y_i, \phi_i)$. For the PFN family, the latent observable is $\mathcal{O}_a = \sum_i \Phi_a(y_i, \phi_i, z_i, \text{PID}_i)$, with different levels of particle-ID (PID) information. The output of F is a softmaxed signal (S) versus background (B) discriminant.

```
# preprocess by centering jets and normalizing pts
for x in X:
    mask = x[:,0] > 0
    yphi_avg = np.average(x[mask,1:3], weights=x[mask,0], axis=0)
    x[mask,1:3] -= yphi_avg
    x[mask,0] /= x[:,0].sum()

print('Finished preprocessing')

# do train/val/test split
(z_train, z_val, z_test,
 p_train, p_val, p_test,
 Y_train, Y_val, Y_test) = data_split(X[:,:,0], X[:,:,1:], y, val=val,
test=test)

print('Done train/val/test split')
print('Model summary:')

# build architecture
efn = EFN(input_dim=2, Phi_sizes=Phi_sizes, F_sizes=F_sizes)
```

```
Finished preprocessing
Done train/val/test split
Model summary:
Model: "model_1"
```

```
_________________________________________________________________________________
__________________________
 Layer (type)                   Output Shape         Param #
Connected to
=================================================================================
==========================
 phats_input (InputLayer)       [(None, None, 2)]    0            []


 tdist_0 (TimeDistributed)      (None, None, 100)    300
['phats_input[0][0]']


 activation_7 (Activation)      (None, None, 100)    0
['tdist_0[0][0]']


 tdist_1 (TimeDistributed)      (None, None, 100)    10100
['activation_7[0][0]']


 activation_8 (Activation)      (None, None, 100)    0
['tdist_1[0][0]']


 zs_input (InputLayer)          [(None, None)]       0            []


 tdist_2 (TimeDistributed)      (None, None, 128)    12928
['activation_8[0][0]']


 mask (Lambda)                  (None, None)         0
['zs_input[0][0]']


 activation_9 (Activation)      (None, None, 128)    0
['tdist_2[0][0]']


 sum (Dot)                      (None, 128)          0
['mask[0][0]',

'activation_9[0][0]']


 dense_0 (Dense)                (None, 100)          12900
```

```
['sum[0][0]']


 activation_10 (Activation)     (None, 100)          0
['dense_0[0][0]']


 dense_1 (Dense)                (None, 100)          10100
['activation_10[0][0]']


 activation_11 (Activation)     (None, 100)          0
['dense_1[0][0]']


 dense_2 (Dense)                (None, 100)          10100
['activation_11[0][0]']


 activation_12 (Activation)     (None, 100)          0
['dense_2[0][0]']


 output (Dense)                 (None, 2)            202
['activation_12[0][0]']


 activation_13 (Activation)     (None, 2)            0
['output[0][0]']


======================================================================
============================
Total params: 56,630
Trainable params: 56,630
Non-trainable params: 0
______________________________________________________________________
____________________________
```

```
# train model
efn.fit([z_train, p_train], Y_train,
        epochs=num_epoch,
        batch_size=batch_size,
        validation_data=([z_val, p_val], Y_val),
        verbose=1)
```

```
Epoch 1/100
1172/1172 [==============================] - 19s 10ms/step - loss:
0.5214 - acc: 0.7414 - val_loss: 0.4888 - val_acc: 0.7682
Epoch 2/100
```

```
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4862 - acc: 0.7698 - val_loss: 0.4829 - val_acc: 0.7719
Epoch 3/100
1172/1172 [==============================] - 10s 8ms/step - loss:
0.4792 - acc: 0.7747 - val_loss: 0.4813 - val_acc: 0.7728
Epoch 4/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4741 - acc: 0.7785 - val_loss: 0.4742 - val_acc: 0.7779
Epoch 5/100
1172/1172 [==============================] - 10s 8ms/step - loss:
0.4680 - acc: 0.7820 - val_loss: 0.4678 - val_acc: 0.7828
Epoch 6/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4623 - acc: 0.7861 - val_loss: 0.4650 - val_acc: 0.7847
Epoch 7/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4582 - acc: 0.7890 - val_loss: 0.4606 - val_acc: 0.7869
Epoch 8/100
1172/1172 [==============================] - 10s 8ms/step - loss:
0.4552 - acc: 0.7911 - val_loss: 0.4577 - val_acc: 0.7899
Epoch 9/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4530 - acc: 0.7920 - val_loss: 0.4592 - val_acc: 0.7879
Epoch 10/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4507 - acc: 0.7940 - val_loss: 0.4500 - val_acc: 0.7926
Epoch 11/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4487 - acc: 0.7952 - val_loss: 0.4541 - val_acc: 0.7907
Epoch 12/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4474 - acc: 0.7961 - val_loss: 0.4482 - val_acc: 0.7954
Epoch 13/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4453 - acc: 0.7979 - val_loss: 0.4508 - val_acc: 0.7928
Epoch 14/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4446 - acc: 0.7977 - val_loss: 0.4492 - val_acc: 0.7945
Epoch 15/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4439 - acc: 0.7988 - val_loss: 0.4478 - val_acc: 0.7950
Epoch 16/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4427 - acc: 0.7994 - val_loss: 0.4458 - val_acc: 0.7967
Epoch 17/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4415 - acc: 0.8002 - val_loss: 0.4464 - val_acc: 0.7955
Epoch 18/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4399 - acc: 0.8004 - val_loss: 0.4493 - val_acc: 0.7950
```

```
Epoch 19/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4399 - acc: 0.8013 - val_loss: 0.4450 - val_acc: 0.7980
Epoch 20/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4387 - acc: 0.8016 - val_loss: 0.4444 - val_acc: 0.7980
Epoch 21/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4377 - acc: 0.8021 - val_loss: 0.4441 - val_acc: 0.7983
Epoch 22/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4373 - acc: 0.8025 - val_loss: 0.4459 - val_acc: 0.7965
Epoch 23/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4366 - acc: 0.8035 - val_loss: 0.4471 - val_acc: 0.7973
Epoch 24/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4354 - acc: 0.8035 - val_loss: 0.4451 - val_acc: 0.7971
Epoch 25/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4353 - acc: 0.8041 - val_loss: 0.4423 - val_acc: 0.7998
Epoch 26/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4340 - acc: 0.8047 - val_loss: 0.4433 - val_acc: 0.7992
Epoch 27/100
1172/1172 [==============================] - 10s 8ms/step - loss:
0.4333 - acc: 0.8053 - val_loss: 0.4422 - val_acc: 0.7983
Epoch 28/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4328 - acc: 0.8054 - val_loss: 0.4424 - val_acc: 0.7996
Epoch 29/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4319 - acc: 0.8062 - val_loss: 0.4398 - val_acc: 0.8017
Epoch 30/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4312 - acc: 0.8067 - val_loss: 0.4424 - val_acc: 0.8001
Epoch 31/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4306 - acc: 0.8070 - val_loss: 0.4446 - val_acc: 0.7987
Epoch 32/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4298 - acc: 0.8075 - val_loss: 0.4410 - val_acc: 0.8008
Epoch 33/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4293 - acc: 0.8072 - val_loss: 0.4408 - val_acc: 0.7997
Epoch 34/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4290 - acc: 0.8072 - val_loss: 0.4411 - val_acc: 0.7997
Epoch 35/100
1172/1172 [==============================] - 10s 8ms/step - loss:
```

```
0.4279 - acc: 0.8086 - val_loss: 0.4390 - val_acc: 0.8023
Epoch 36/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4275 - acc: 0.8088 - val_loss: 0.4424 - val_acc: 0.8003
Epoch 37/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4269 - acc: 0.8091 - val_loss: 0.4414 - val_acc: 0.8003
Epoch 38/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4263 - acc: 0.8094 - val_loss: 0.4409 - val_acc: 0.8009
Epoch 39/100
1172/1172 [==============================] - 10s 8ms/step - loss:
0.4258 - acc: 0.8094 - val_loss: 0.4401 - val_acc: 0.8008
Epoch 40/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4254 - acc: 0.8096 - val_loss: 0.4403 - val_acc: 0.8015
Epoch 41/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4244 - acc: 0.8104 - val_loss: 0.4416 - val_acc: 0.8009
Epoch 42/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4243 - acc: 0.8102 - val_loss: 0.4459 - val_acc: 0.7992
Epoch 43/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4236 - acc: 0.8111 - val_loss: 0.4427 - val_acc: 0.7994
Epoch 44/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4228 - acc: 0.8114 - val_loss: 0.4396 - val_acc: 0.8030
Epoch 45/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4221 - acc: 0.8120 - val_loss: 0.4386 - val_acc: 0.8019
Epoch 46/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4216 - acc: 0.8122 - val_loss: 0.4418 - val_acc: 0.8008
Epoch 47/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4214 - acc: 0.8123 - val_loss: 0.4394 - val_acc: 0.8026
Epoch 48/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4208 - acc: 0.8130 - val_loss: 0.4415 - val_acc: 0.8005
Epoch 49/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4205 - acc: 0.8128 - val_loss: 0.4412 - val_acc: 0.8009
Epoch 50/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4196 - acc: 0.8138 - val_loss: 0.4432 - val_acc: 0.8006
Epoch 51/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4193 - acc: 0.8136 - val_loss: 0.4421 - val_acc: 0.8020
Epoch 52/100
```

```
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4188 - acc: 0.8142 - val_loss: 0.4423 - val_acc: 0.8001
Epoch 53/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4184 - acc: 0.8141 - val_loss: 0.4408 - val_acc: 0.8025
Epoch 54/100
1172/1172 [==============================] - 10s 8ms/step - loss:
0.4175 - acc: 0.8145 - val_loss: 0.4443 - val_acc: 0.8018
Epoch 55/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4172 - acc: 0.8153 - val_loss: 0.4416 - val_acc: 0.8017
Epoch 56/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4168 - acc: 0.8149 - val_loss: 0.4427 - val_acc: 0.8010
Epoch 57/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4162 - acc: 0.8156 - val_loss: 0.4415 - val_acc: 0.8011
Epoch 58/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4159 - acc: 0.8154 - val_loss: 0.4423 - val_acc: 0.8008
Epoch 59/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4152 - acc: 0.8153 - val_loss: 0.4419 - val_acc: 0.8016
Epoch 60/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4145 - acc: 0.8167 - val_loss: 0.4427 - val_acc: 0.8013
Epoch 61/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4144 - acc: 0.8166 - val_loss: 0.4422 - val_acc: 0.8016
Epoch 62/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4135 - acc: 0.8179 - val_loss: 0.4423 - val_acc: 0.8020
Epoch 63/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4135 - acc: 0.8169 - val_loss: 0.4460 - val_acc: 0.8007
Epoch 64/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4126 - acc: 0.8180 - val_loss: 0.4461 - val_acc: 0.8003
Epoch 65/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4119 - acc: 0.8179 - val_loss: 0.4424 - val_acc: 0.8010
Epoch 66/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4117 - acc: 0.8186 - val_loss: 0.4437 - val_acc: 0.8008
Epoch 67/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4111 - acc: 0.8185 - val_loss: 0.4462 - val_acc: 0.8000
Epoch 68/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4108 - acc: 0.8186 - val_loss: 0.4468 - val_acc: 0.8009
```

```
Epoch 69/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4099 - acc: 0.8197 - val_loss: 0.4476 - val_acc: 0.7990
Epoch 70/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4097 - acc: 0.8195 - val_loss: 0.4468 - val_acc: 0.7991
Epoch 71/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4093 - acc: 0.8199 - val_loss: 0.4473 - val_acc: 0.7997
Epoch 72/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4086 - acc: 0.8200 - val_loss: 0.4458 - val_acc: 0.8003
Epoch 73/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4083 - acc: 0.8203 - val_loss: 0.4473 - val_acc: 0.7999
Epoch 74/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4074 - acc: 0.8207 - val_loss: 0.4506 - val_acc: 0.7992
Epoch 75/100
1172/1172 [==============================] - 10s 8ms/step - loss:
0.4073 - acc: 0.8206 - val_loss: 0.4491 - val_acc: 0.7998
Epoch 76/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4066 - acc: 0.8207 - val_loss: 0.4479 - val_acc: 0.8002
Epoch 77/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4061 - acc: 0.8214 - val_loss: 0.4502 - val_acc: 0.8008
Epoch 78/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4059 - acc: 0.8216 - val_loss: 0.4484 - val_acc: 0.7991
Epoch 79/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4056 - acc: 0.8218 - val_loss: 0.4497 - val_acc: 0.7982
Epoch 80/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4051 - acc: 0.8221 - val_loss: 0.4501 - val_acc: 0.7977
Epoch 81/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4047 - acc: 0.8219 - val_loss: 0.4503 - val_acc: 0.7987
Epoch 82/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4049 - acc: 0.8221 - val_loss: 0.4495 - val_acc: 0.8000
Epoch 83/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4036 - acc: 0.8229 - val_loss: 0.4505 - val_acc: 0.8007
Epoch 84/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4033 - acc: 0.8233 - val_loss: 0.4495 - val_acc: 0.7996
Epoch 85/100
1172/1172 [==============================] - 9s 8ms/step - loss:
```

```
0.4032 - acc: 0.8230 - val_loss: 0.4514 - val_acc: 0.7981
Epoch 86/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4025 - acc: 0.8237 - val_loss: 0.4506 - val_acc: 0.7985
Epoch 87/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4022 - acc: 0.8236 - val_loss: 0.4497 - val_acc: 0.7993
Epoch 88/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4019 - acc: 0.8235 - val_loss: 0.4526 - val_acc: 0.7983
Epoch 89/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.4013 - acc: 0.8241 - val_loss: 0.4509 - val_acc: 0.7997
Epoch 90/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.4003 - acc: 0.8249 - val_loss: 0.4551 - val_acc: 0.7997
Epoch 91/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.4006 - acc: 0.8246 - val_loss: 0.4542 - val_acc: 0.7999
Epoch 92/100
1172/1172 [==============================] - 10s 8ms/step - loss:
0.4000 - acc: 0.8249 - val_loss: 0.4554 - val_acc: 0.7983
Epoch 93/100
1172/1172 [==============================] - 11s 9ms/step - loss:
0.3996 - acc: 0.8249 - val_loss: 0.4565 - val_acc: 0.7977
Epoch 94/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.3993 - acc: 0.8254 - val_loss: 0.4525 - val_acc: 0.7984
Epoch 95/100
1172/1172 [==============================] - 10s 9ms/step - loss:
0.3991 - acc: 0.8250 - val_loss: 0.4562 - val_acc: 0.7961
Epoch 96/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.3986 - acc: 0.8257 - val_loss: 0.4559 - val_acc: 0.7993
Epoch 97/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.3981 - acc: 0.8261 - val_loss: 0.4561 - val_acc: 0.7980
Epoch 98/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.3977 - acc: 0.8261 - val_loss: 0.4584 - val_acc: 0.7970
Epoch 99/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.3976 - acc: 0.8260 - val_loss: 0.4585 - val_acc: 0.7985
Epoch 100/100
1172/1172 [==============================] - 9s 8ms/step - loss:
0.3970 - acc: 0.8262 - val_loss: 0.4586 - val_acc: 0.7974

<keras.callbacks.History at 0x7fcb506ed790>
```

```
res = efn.evaluate([z_test, p_test], y_test)

print("Test loss: ", res[0])
print("Test accuracy: ", res[1])
```

```
3125/3125 [==============================] - 10s 3ms/step - loss:
1.1717 - acc: 0.5013
Test loss:  1.1716506481170654
Test accuracy:  0.5013099908828735
```

Comparison of Results

Model	Train		Test		Validation	
	Acc	Loss	Acc	Loss	Acc	Loss
ParticleNet-Lite	0.8154	0.4165	0.8093	0.4268	0.8087	0.4275
EFN	0.8262	0.3970	0.5013	1.1717	0.7974	0.4586

Conclusion:

- The "ParticleNet-Lite" model performed better than the "EFN" model on all three metrics: Train Accuracy, Test Accuracy, and Validation Accuracy.
- The "EFN" model achieved a lower Train Loss than the "ParticleNet-Lite" model, but its Test Loss is significantly higher than the "ParticleNet-Lite" model. This suggests that the "EFN" model is overfitting on the training data and does not generalize well to the test data.
- The "ParticleNet-Lite" model achieved a similar Validation Loss to the "EFN" model, indicating that it is not overfitting on the training data and can generalize well to unseen data.

Overall, these results suggest that the "ParticleNet-Lite" model is the better performing model between the two, as it achieved higher accuracy on all three metrics while maintaining a relatively low test loss. However, further analysis would be required to determine the significance of the differences in the accuracy and loss between the two models.

References

[1] Energy Flow Networks: Deep Sets for Particle Jets

[2] ParticleNet: Jet Tagging via Particle Clouds