

ECCE Simulation

AI/ML charged pion/muon classification

A. Quiroga, W. Phelps, C. Fanelli, and J. Huang

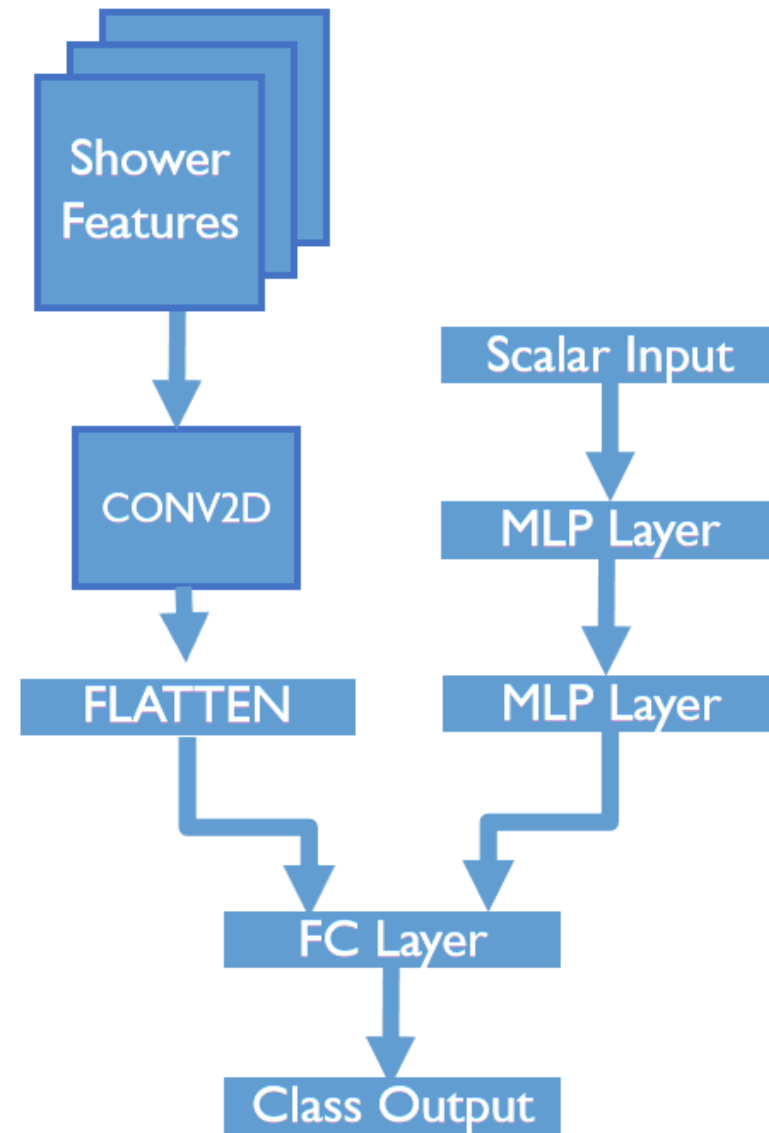
Keras Functional API

```
# build convolution model for tower features
tower_input = layers.Input(shape=(7, 7, 3))
x = layers.Conv2D(16, (3, 3), padding="same")(tower_input)
x = layers.LeakyReLU(alpha=0.2)(x)
# x = layers.Conv2D(64, (3, 3), padding="same")(x)
# x = layers.LeakyReLU(alpha=0.2)(x)
x = layers.Flatten()(x)

# build simple MLP model for all other features
dict_input = layers.Input(shape=(14,))
y = layers.Dense(units=64, activation='relu')(dict_input)
y = layers.Dense(units=64, activation='relu')(y)

# fully connected layer
combined = layers.concatenate((x, y))
z = layers.Dense(32, activation="relu")(combined)
z = layers.Dense(32, activation="relu")(z)

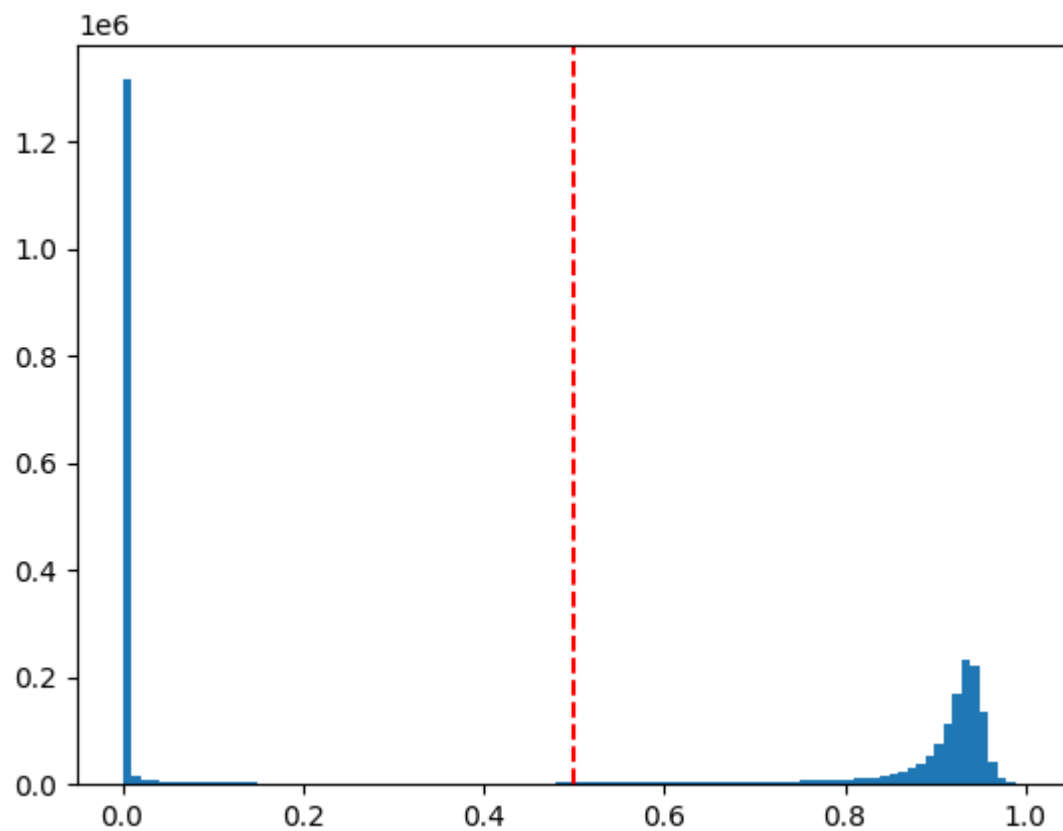
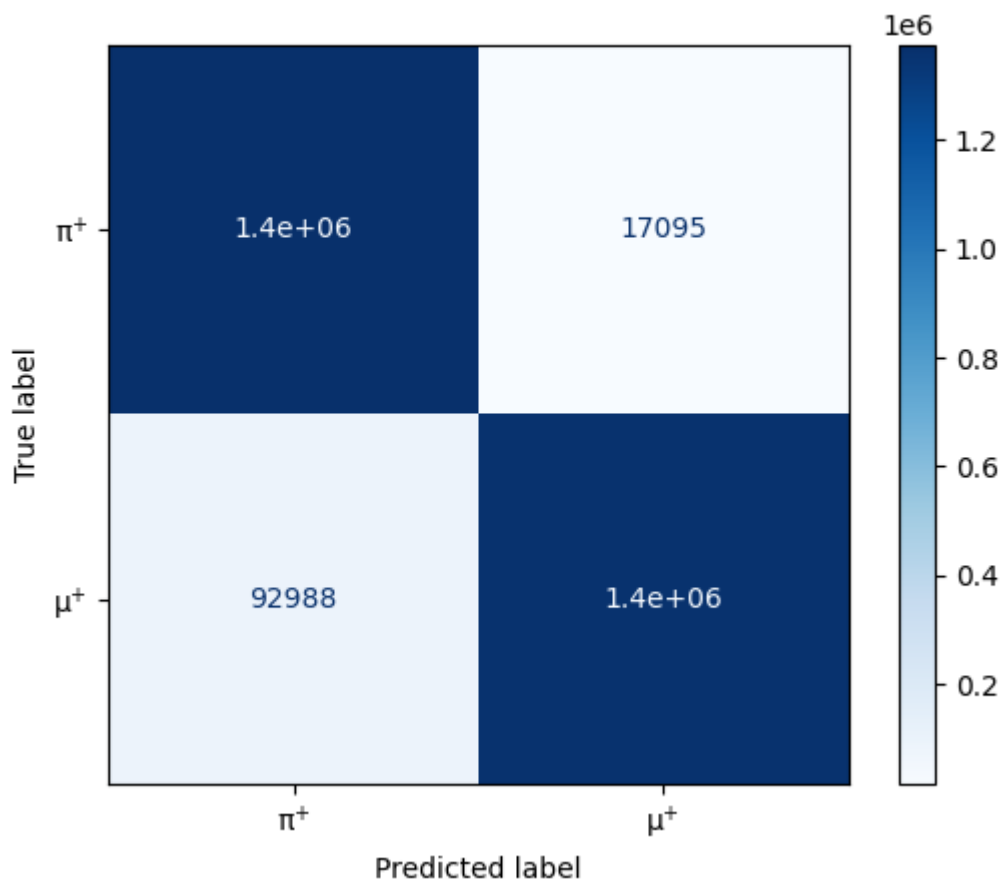
# single output float
z = layers.Dense(1, activation="linear")(z)
```



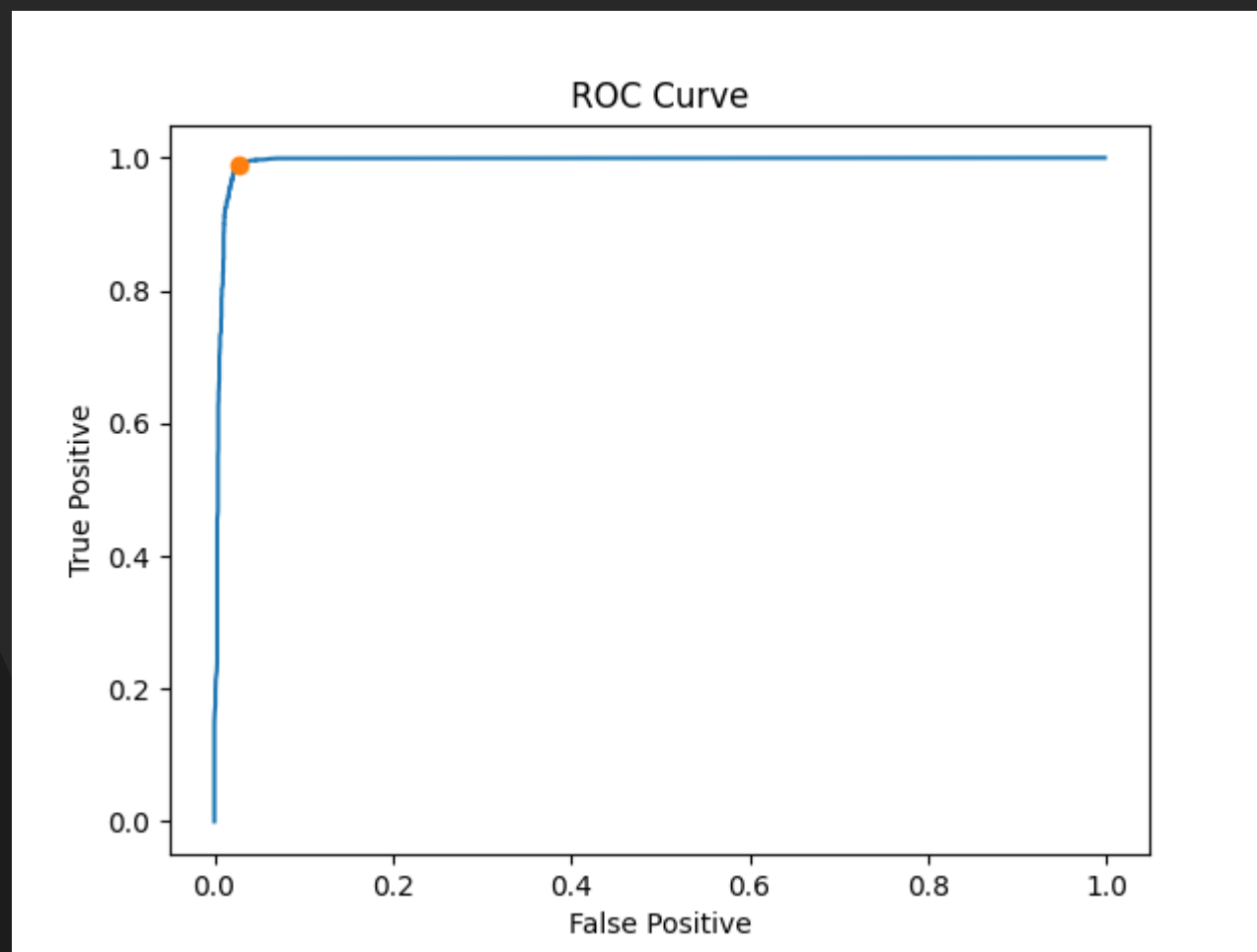
Disclaimer

Keep in mind these are example plots, they are not current!
These were used for sanity checking and clarification purposes.

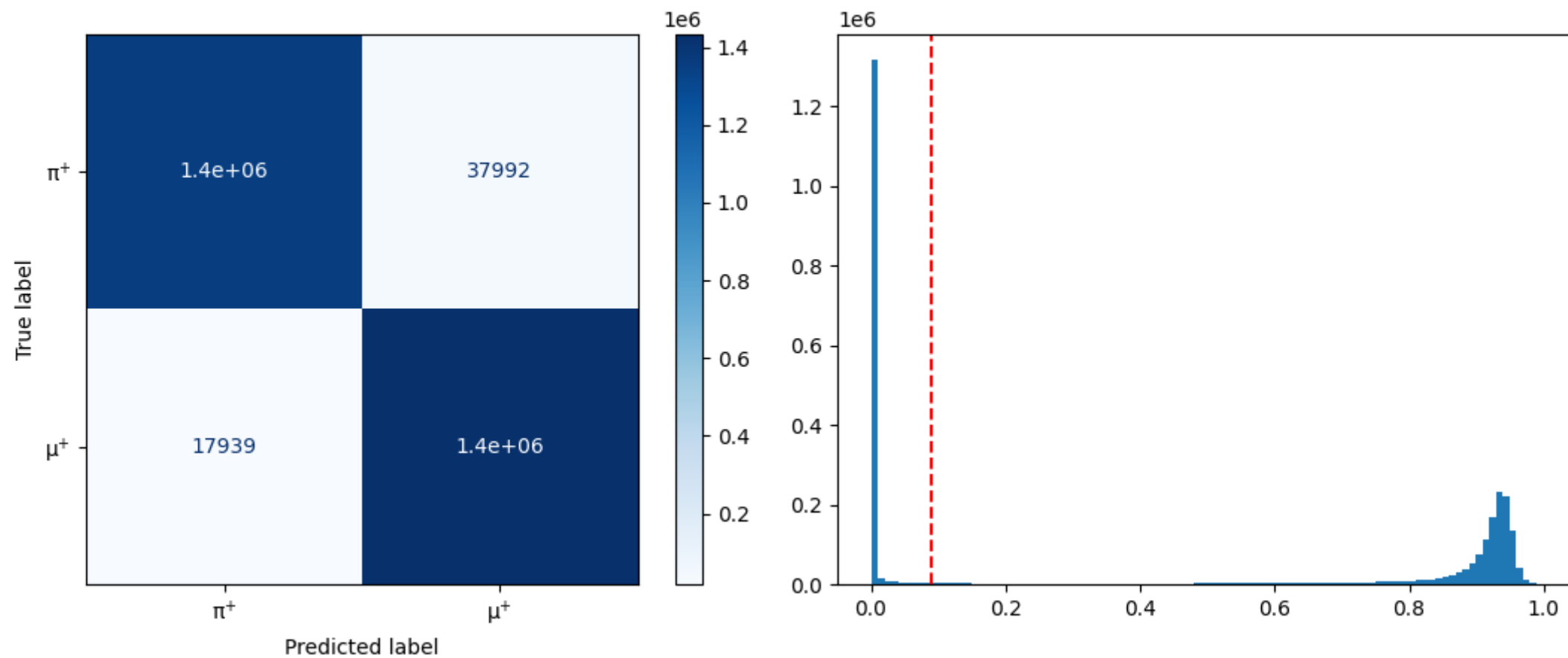
Model Output



Fine tuned

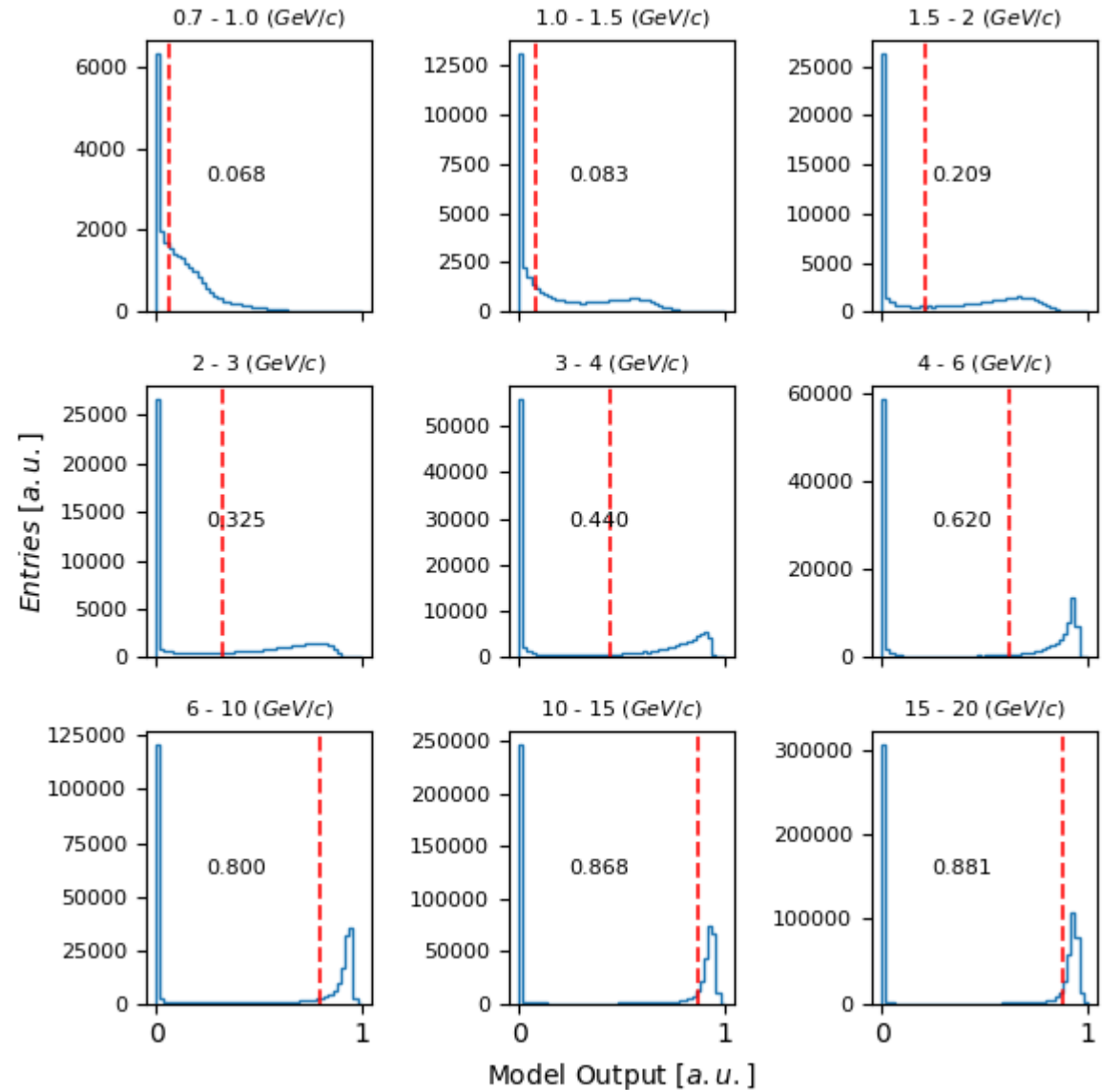


Fine tuned



Fine tuned

- By binning the model's output, I can more deliberately select the cuts per bin. Allowing for more 'fine tuned' classification.



Fine tuned

- These cuts can be adjusted to achieve specific muon efficiencies to compare to the traditional cuts.

