NMSU Update

Dinupa

March 31, 2023

Binned Data

- We binned the data in the following ranges;

```
mass = [4., 5.), [5., 6.)
pT = [0., 0.4), [0.4, 0.8), [0.8, 1.2), [1.2, 1.6)
xF = [0.1, 0.3), [0.3, 0.5), [0.5, 0.7), [0.7, 0.9)
```

- We filled the test histograms with 50K DY reco. events with `occuD1 < 200.`.
- When training the neural network, we need to train the network in the bins that we are interested. For example, if we need to extract the angular coef. in the $x_F = [0.3, 0.5)$ bin we have to train the network in the same bin.

$\{\lambda, \mu, \nu\} = \{0.4, 0.4, 0.4\}$ in $mass = [4., 5.)$

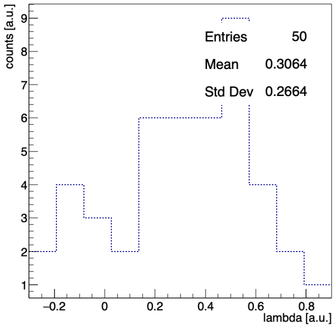


Figure 1: Extracted lambda.

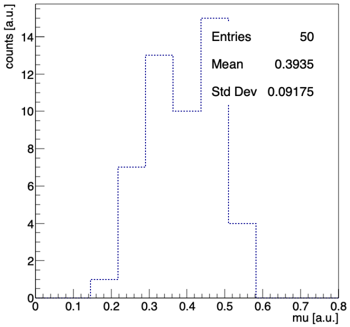


Figure 2: Extracted mu.

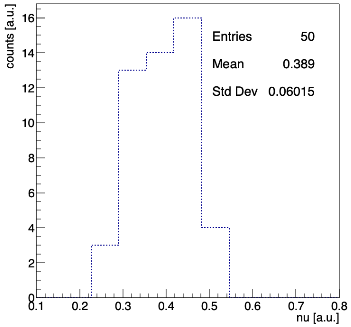


Figure 3: Extracted nu.

$\{\lambda, \mu, \nu\} = \{0.4, 0.4, 0.4\}$ in $p_T = [0.4, 0.8)$

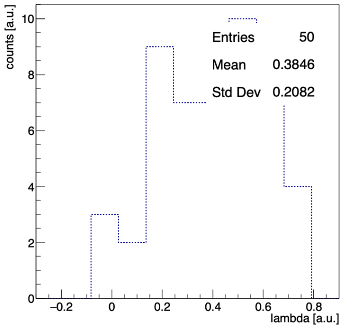


Figure 4: Extracted lambda.

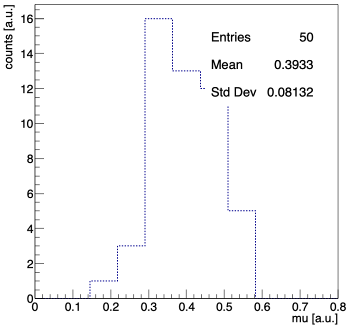


Figure 5: Extracted mu.

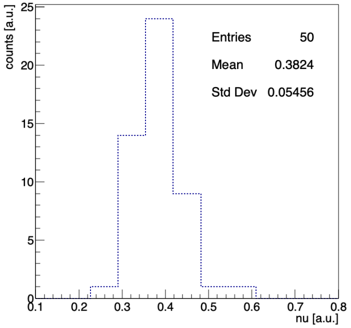


Figure 6: Extracted nu.

$\{\lambda, \mu, \nu\} = \{0.4, 0.4, 0.4\}$ in $x_F = [0.3, 0.5)$

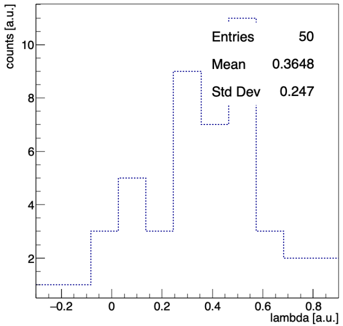


Figure 7: Extracted lambda.

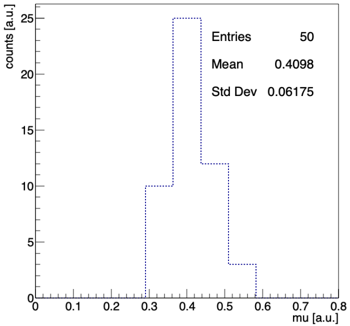


Figure 8: Extracted mu.

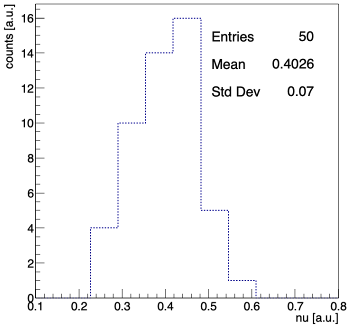


Figure 9: Extracted nu.

$\{\lambda, \mu, \nu\} = \{0.4, 0.4, 0.4\}$ in $x_F = [0.5, 0.7)$

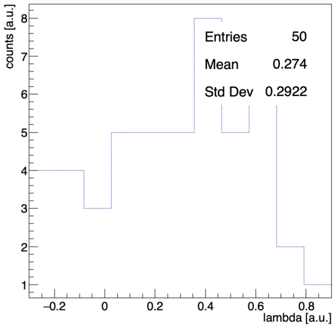


Figure 10: Extracted lambda.

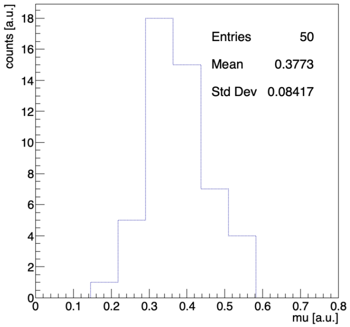


Figure 11: Extracted mu.

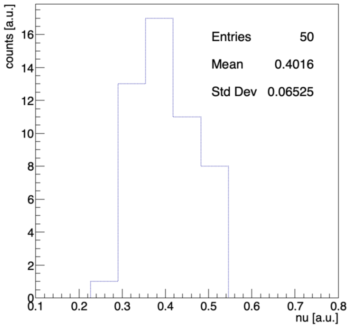
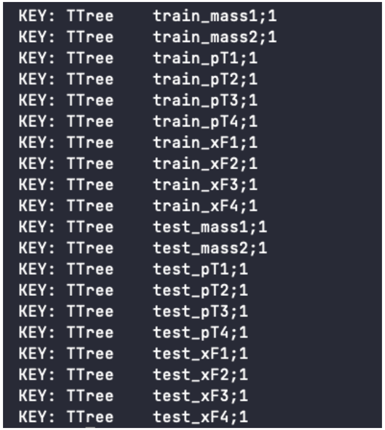


Figure 12: Extracted nu.

Go Parallel

- We can use singularity image with necessary python packages to run the network in grid (OSG). I was able to create singularity image with ubuntu base. But could not install python packages with pip/conda. Plan to discuss this with Abi.
- We can use one tree to train on one node.



```
KEY: TTree    train_mass1;1
KEY: TTree    train_mass2;1
KEY: TTree    train_pT1;1
KEY: TTree    train_pT2;1
KEY: TTree    train_pT3;1
KEY: TTree    train_pT4;1
KEY: TTree    train_xF1;1
KEY: TTree    train_xF2;1
KEY: TTree    train_xF3;1
KEY: TTree    train_xF4;1
KEY: TTree    test_mass1;1
KEY: TTree    test_mass2;1
KEY: TTree    test_pT1;1
KEY: TTree    test_pT2;1
KEY: TTree    test_pT3;1
KEY: TTree    test_pT4;1
KEY: TTree    test_xF1;1
KEY: TTree    test_xF2;1
KEY: TTree    test_xF3;1
KEY: TTree    test_xF4;1
```

Figure 13: Data structure.