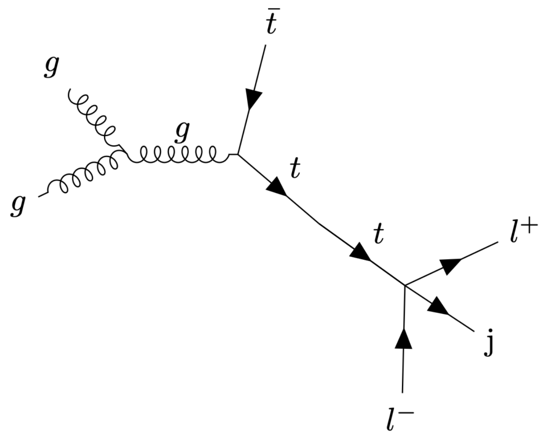
Overview of the Top FC Analysis

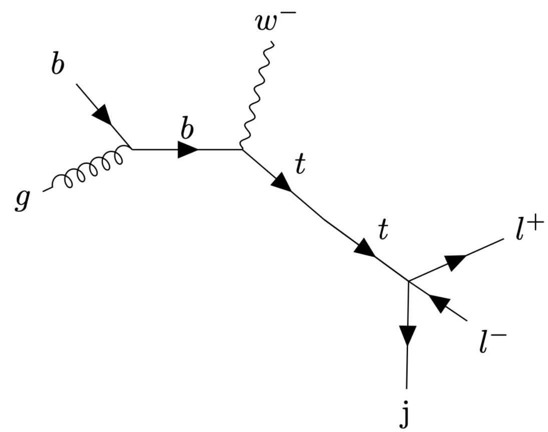
Meisam Ghasemi Bostanabad

Analysis meeting
2023-7-8

Flavor Changing in Top sector

In this analysis we looking for FC ($t \to u$ or $t \to c$) in top sector as the heaviest quark which may be an indicator of new flavor physics.





ttbar

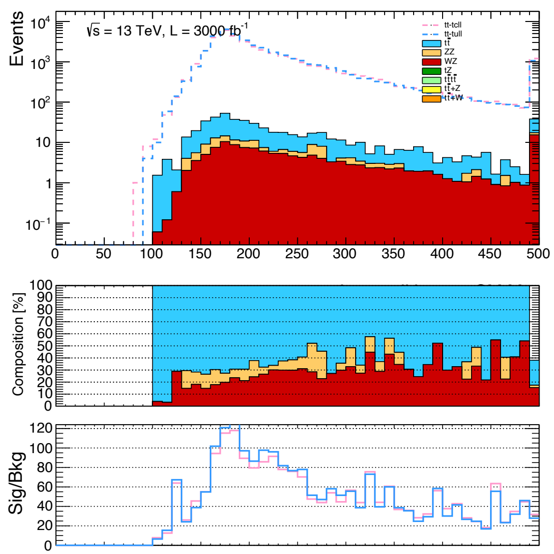
tW (**3M+10M** events generated)

- Starting with **ttbar**, targeting **final states** with three leptons (a pair of OP) and a b-tagged jet (one of the tops decays leptonically via w $\to l\ v_l$)
- The leading potential backgrounds are $t\bar{t}, tZ, WZ, ZZ$

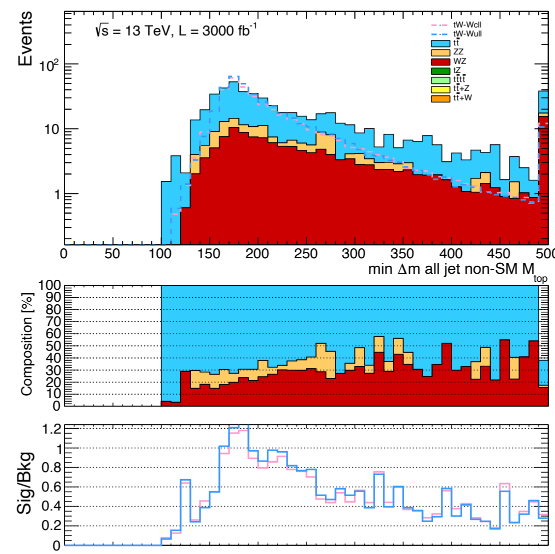
- Next channel **tW**, targeting **final states** with three leptons (a pair of OP) (leptonic decay for W via w $\to l\ v_l$)
- The leading potential backgrounds are tZ, WZ, ZZ

Small bug in weight implementation

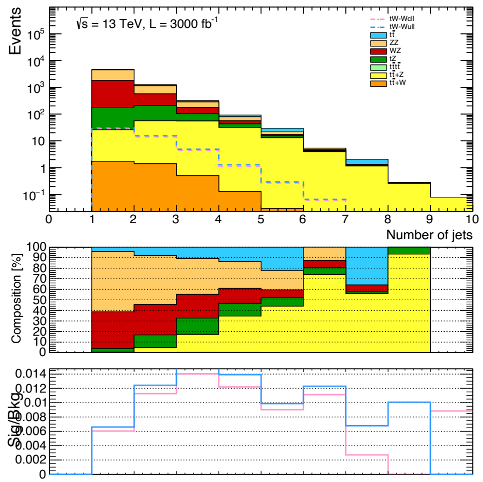
In plotting, cross section weights for signals were not applied (left)! Now fixed (right).

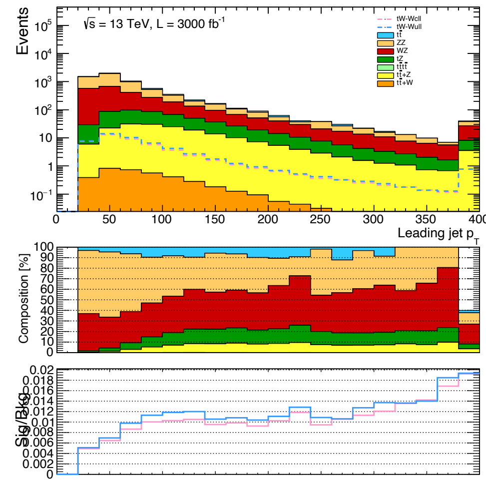


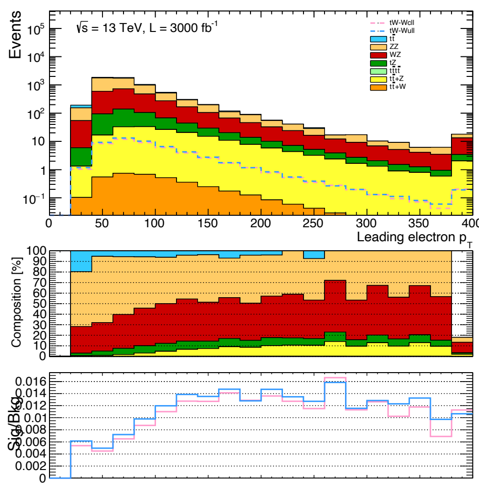
fixed

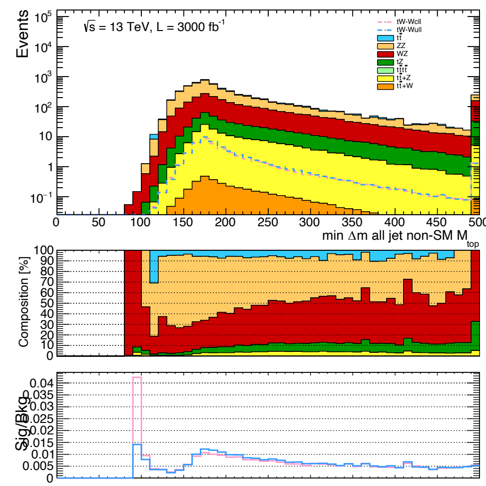


tW channel distributions I

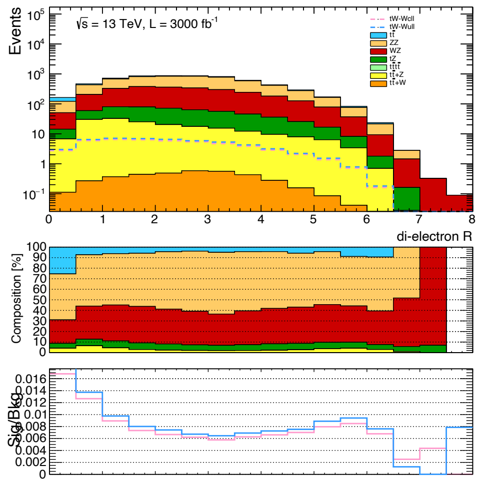


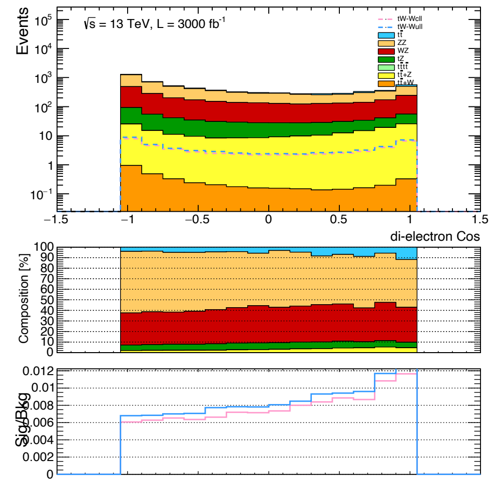


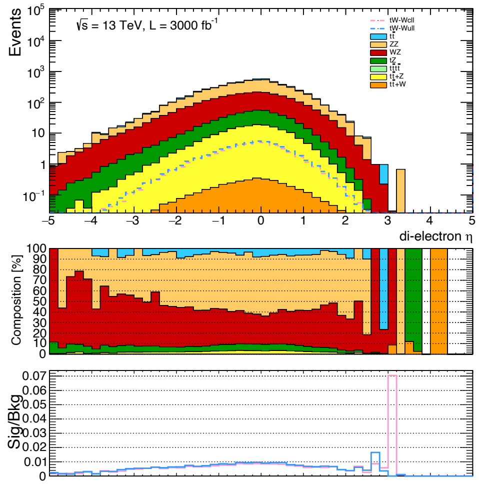


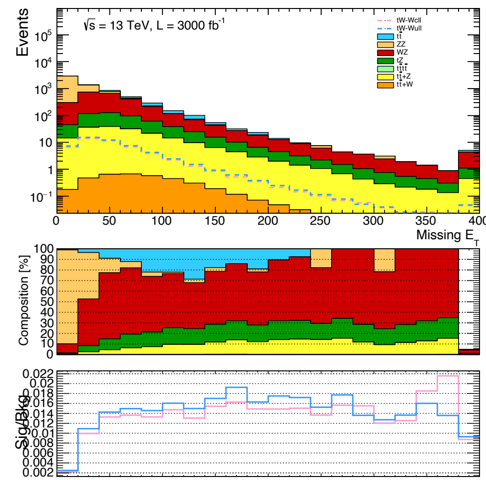


tW channel distributions II

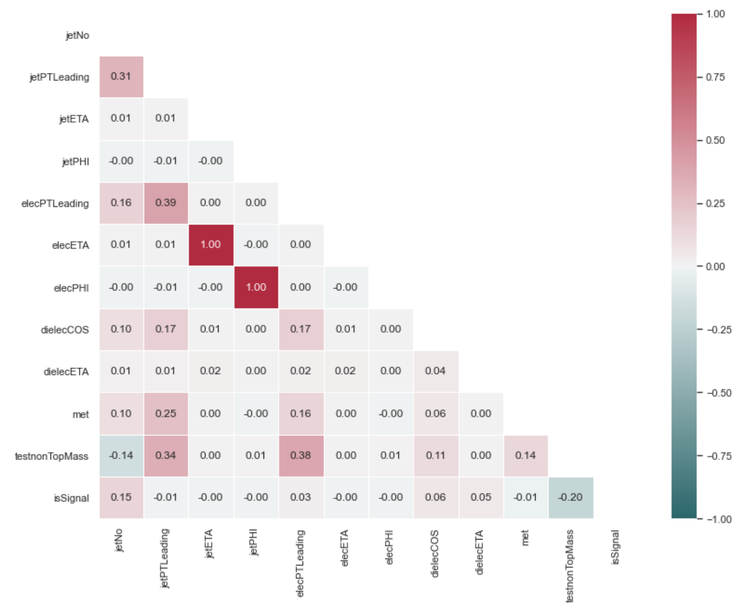






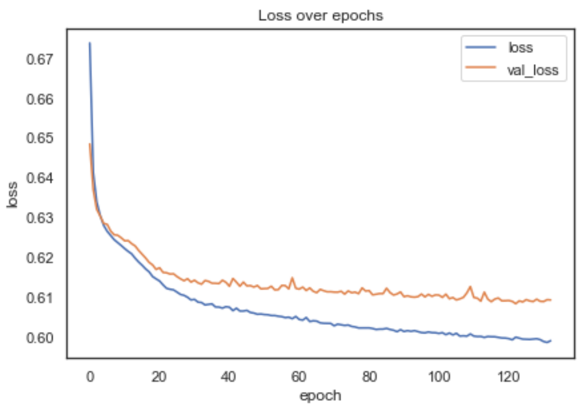


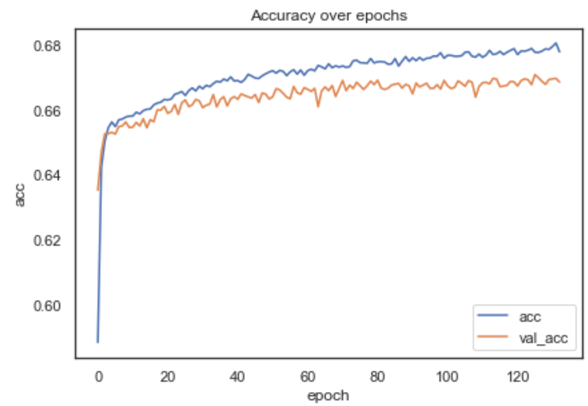
Feature correlations

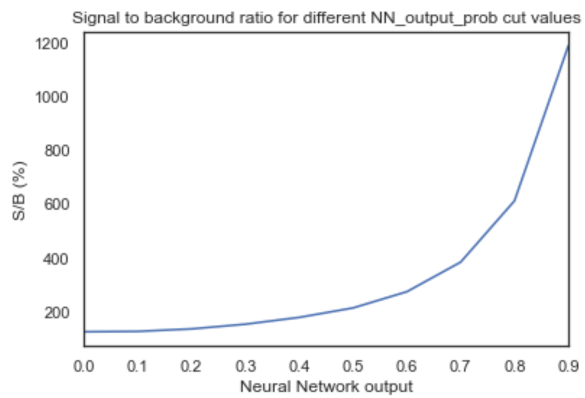


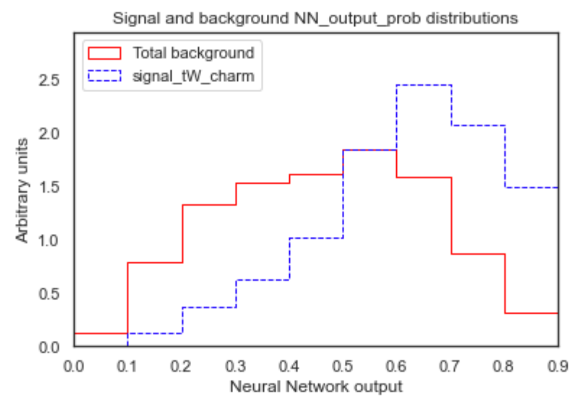
IsSignal is mostly (negatively) correlated to non-SM top mass
JetNo is (positively) correlated – means signal prones to more jets

Simple NN performance



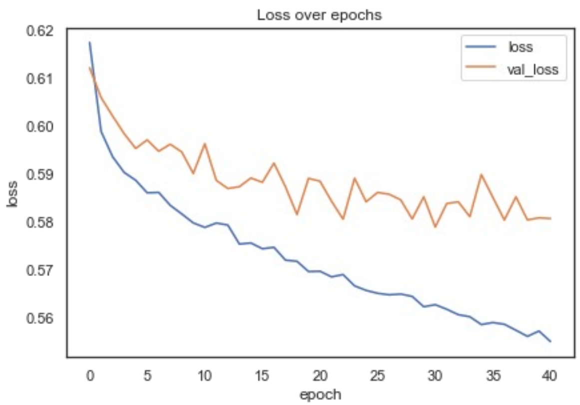


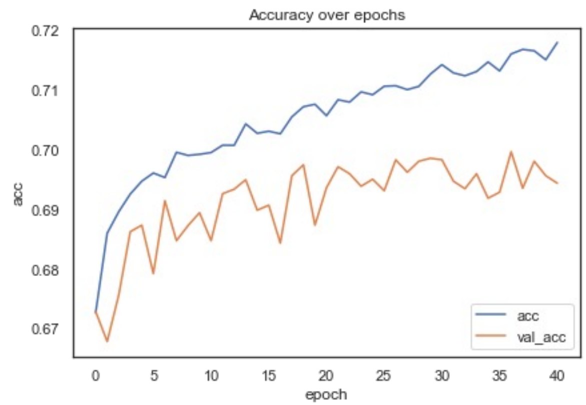


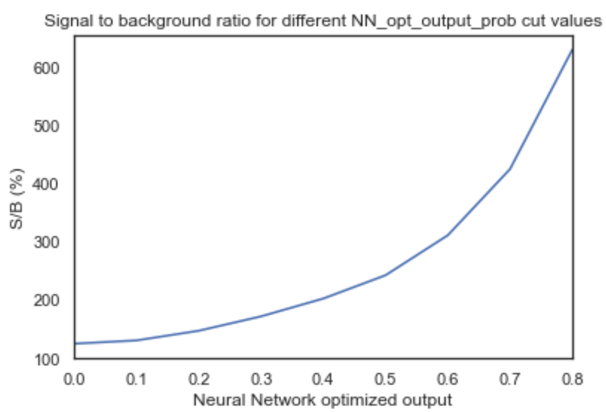


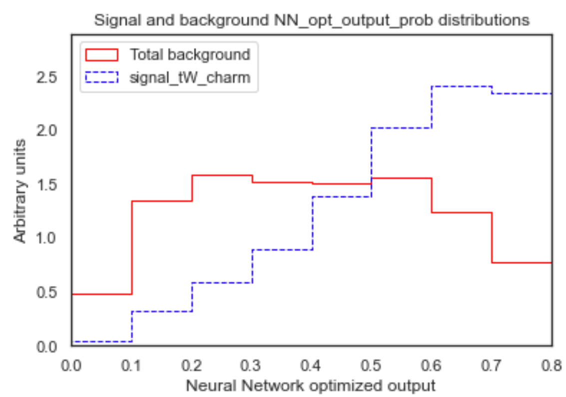
Optimized NN performance

Structure details in the backup

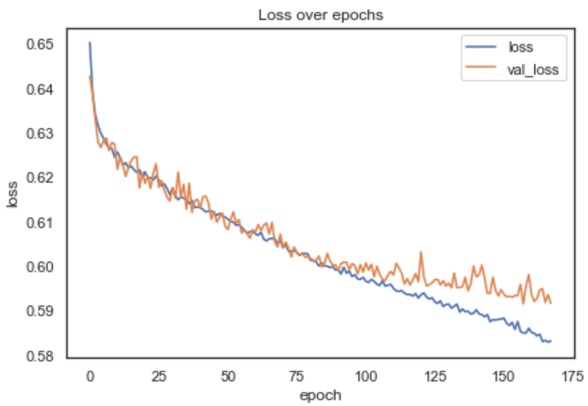


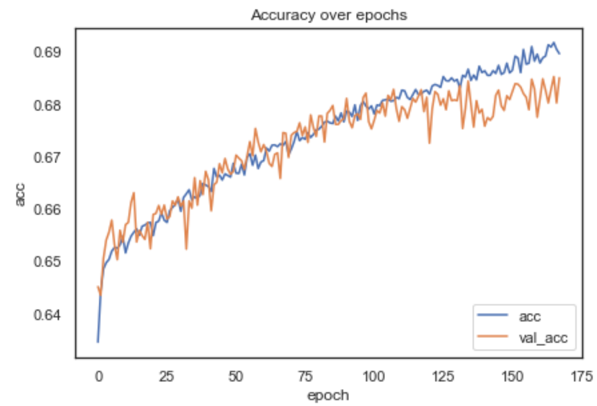


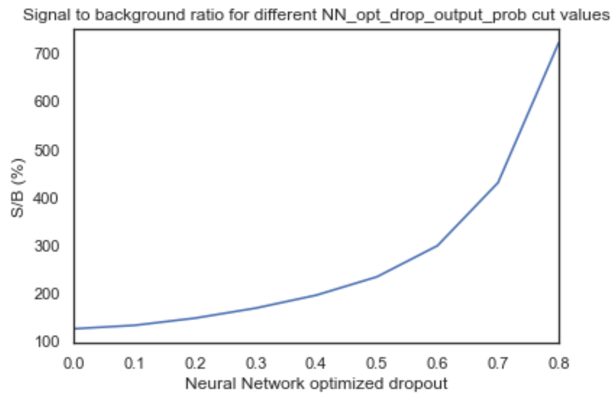


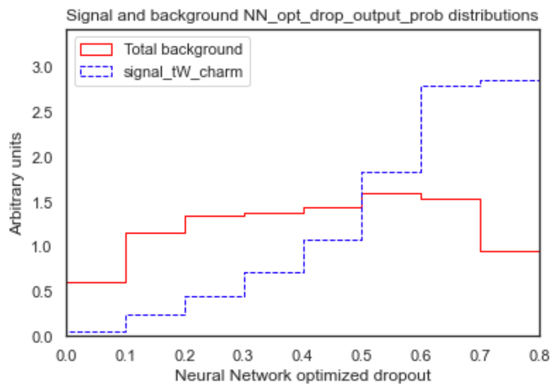


Optimized NN (dropout layer) performance

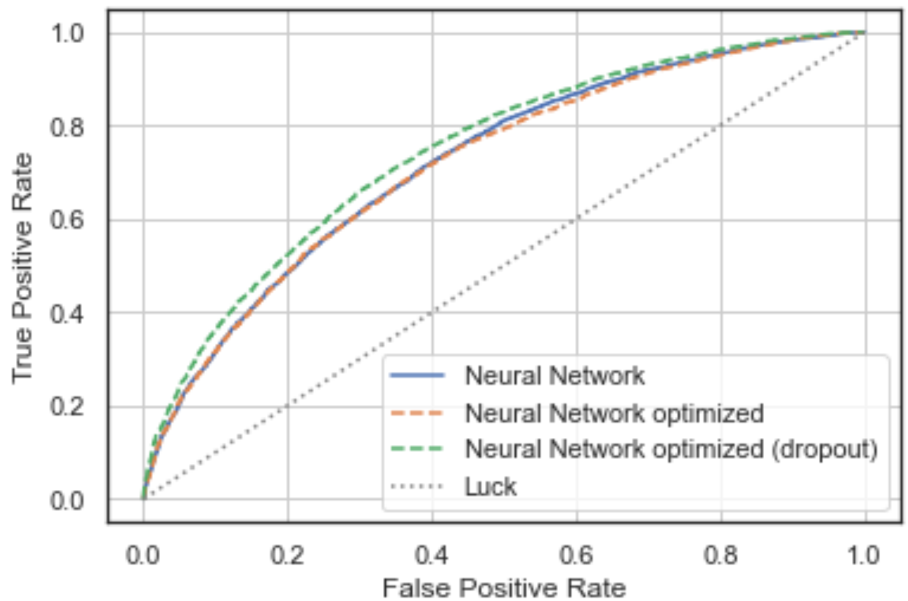






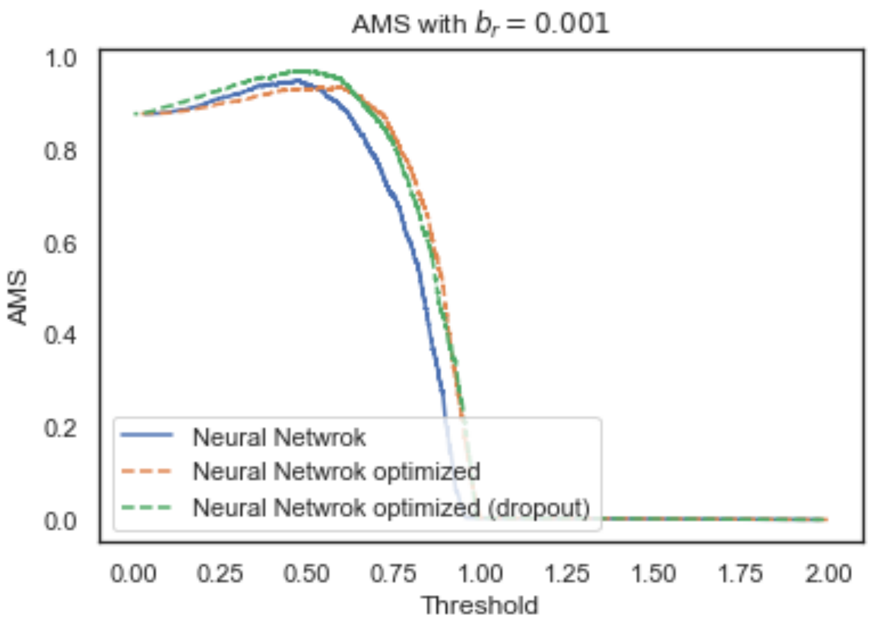


ROC curve



- Receiver Operating Characteristic curve is a graphical representation of the performance of a binary classification model. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at different classification thresholds. A perfect classifier would have a ROC curve that passes through the top-left corner, indicating a high TPR and low FPR.

AMS curve



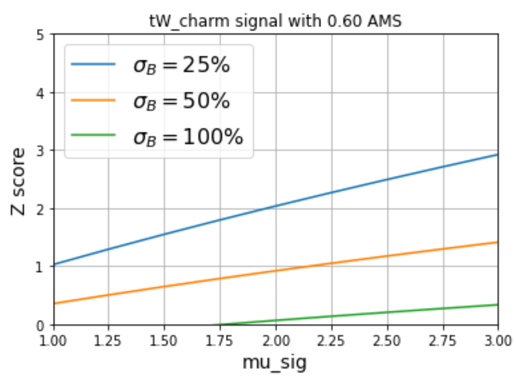
$$\text{AMS} = \sqrt{2\left((TPR + FPR + b_r)\ln\left(1 + \frac{TPR}{FPR + b_r}\right) - TPR\right)}$$

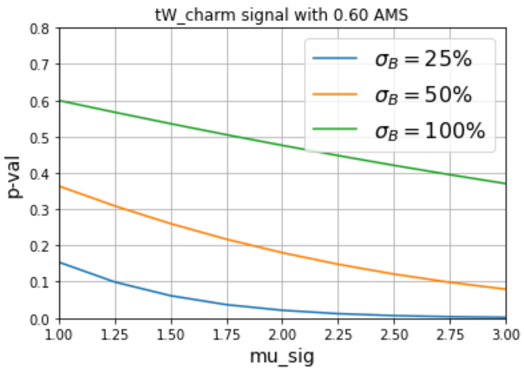
- In classifying signal or background events, the primary goal is optimizing the discovery region for statistical significance. As discussed here, this metric is the approximate median significance (AMS). This metric is used in Higgs Kaggle competition.

tW significance and p-value

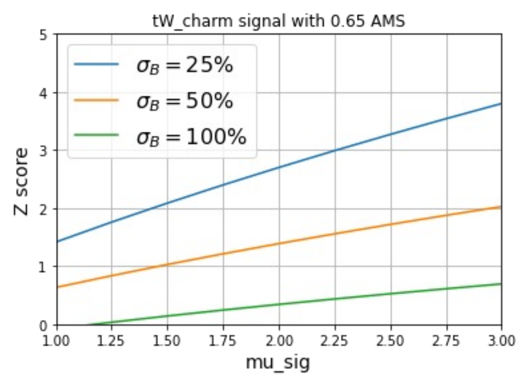
Significance → $p(N_s, N_b, \sigma_b) = \mathcal{I}\left(\frac{1}{1+\frac{1}{N_b\sigma_b^2}}; N_s + N_b; \frac{1}{\sigma_b^2} + 1\right)$

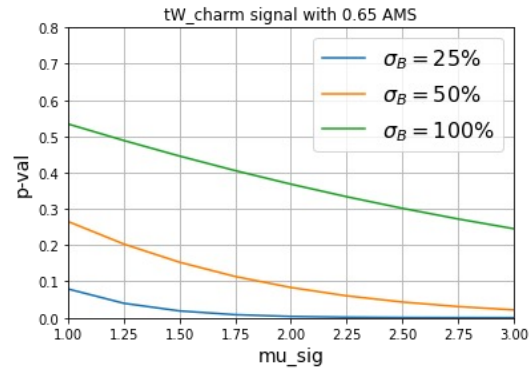
Simple NN



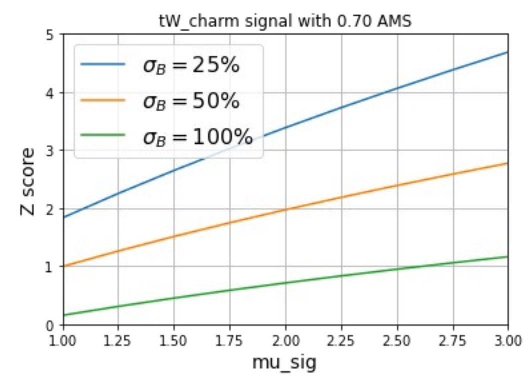


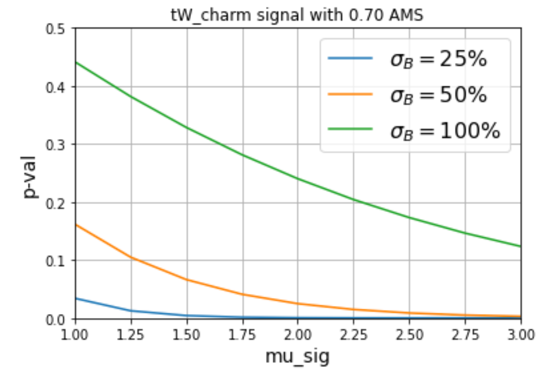
Optimized NN





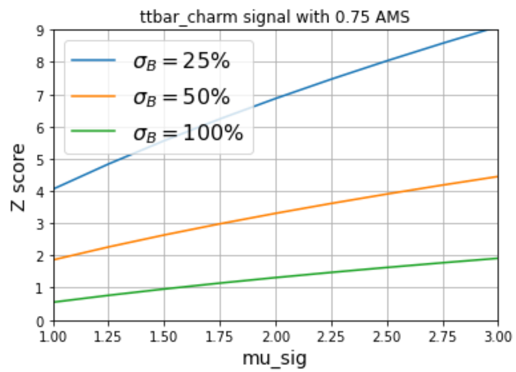
Optimized NN with drop out layers

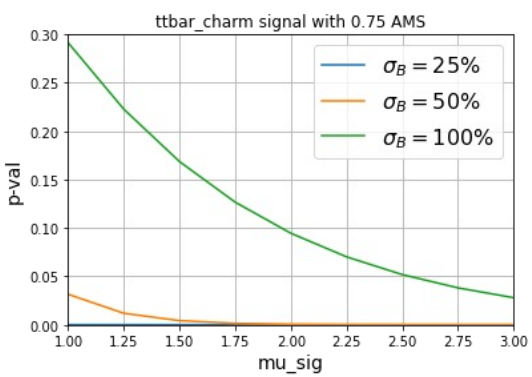




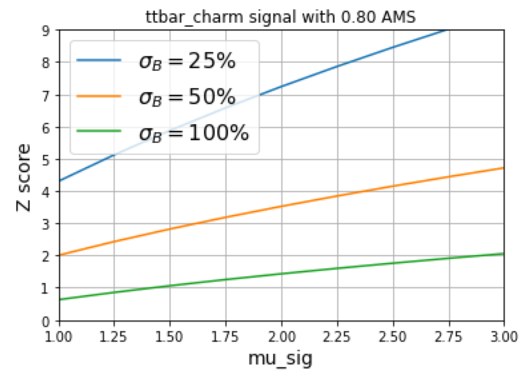
ttbar significance and p-value

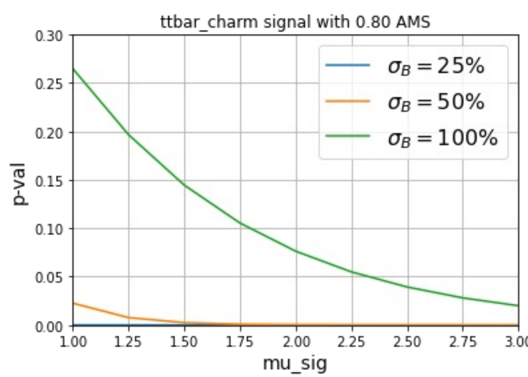
Simple NN



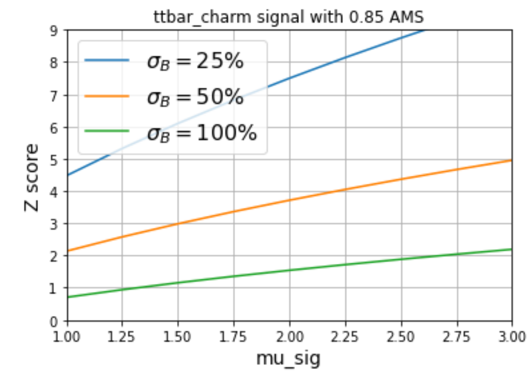


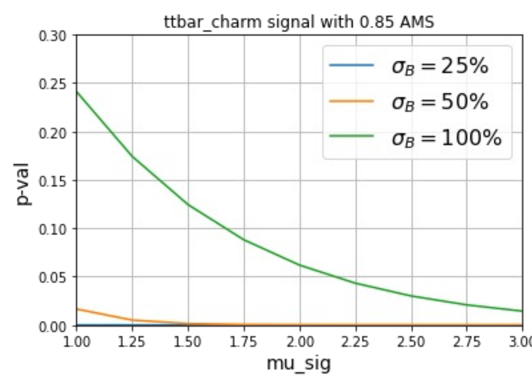
Optimized NN





Optimized NN with drop out layers





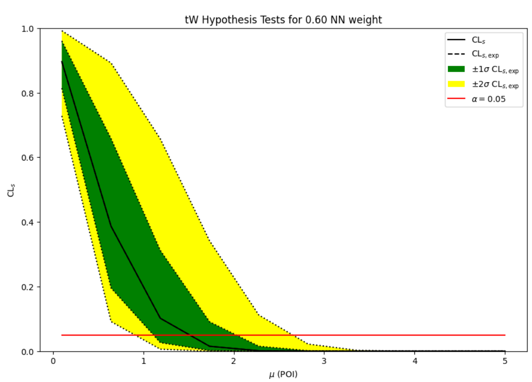
tW upper limits on μ_{sig}

- To get upper limits, we just need to run multiple hypothesis tests for a lot of different null hypotheses of BSM with $\mu_{sig} \in [0, \ldots, 5.0]$ and then find the value of μ_{sig} for which the null hypothesis is rejected (a 95% CLs).

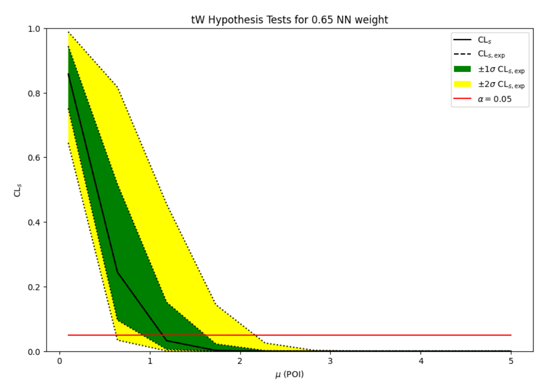
- We can plot the standard "Brazil band" of the observed and expected CLs. The horizontal red line indicates the test size ($\alpha = 0.05$), whose intersection with the CLs lines visually represents the $(1-\alpha)$% CL limit on the μ_{sig}.

- Going to higher AMS threshold, signal background ratio gets bigger and then the 95% CL limit for μ_{sig} becomes smaller.

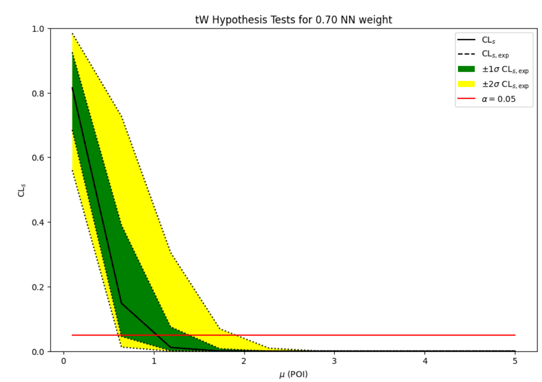
Upper limit (exp): μ = 1.514



Upper limit (exp): μ = 1.143



Upper limit (exp): μ = 1.035



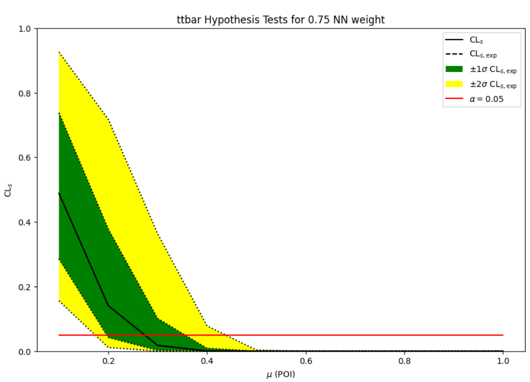
ttbar upper limits on μ_{sig}

- To get upper limits, we just need to run multiple hypothesis tests for a lot of different null hypotheses of BSM with $\mu_{sig} \in [0, \ldots, 5.0]$ and then find the value of μ_{sig} for which the null hypothesis is rejected (a 95% CLs).

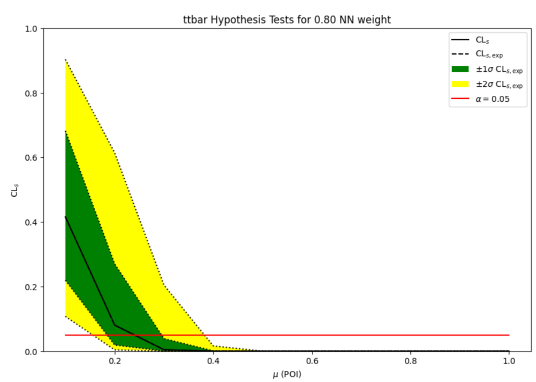
- We can plot the standard “Brazil band” of the observed and expected CLs. The horizontal red line indicates the test size ($\alpha = 0.05$), whose intersection with the CLs lines visually represents the $(1-\alpha)$% CL limit on the μ_{sig}.

- Going to higher AMS threshold, signal background ratio gets bigger and then the 95% CL limit for μ_{sig} becomes smaller.

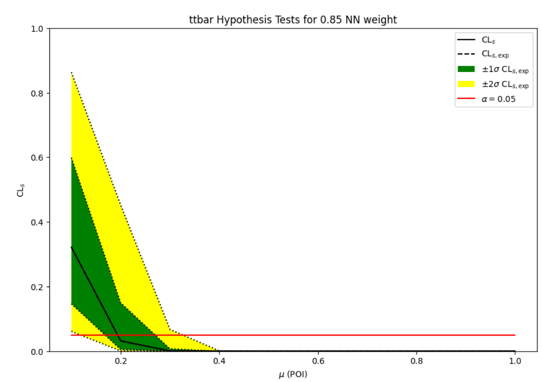
Upper limit (exp): μ = 0.273



Upper limit (exp): μ = 0.240



Upper limit (exp): μ = 0.193



Signal-background yields in tW

Simple NN

	0.75 NN cut	0.80 NN cut	0.85 NN cut
signal	16.5	13.3	10.0
background	157.2	113.7	81.5
S/B	0.105	0.145	0.203

Optimized NN

	0.75 NN cut	0.80 NN cut	0.85 NN cut
signal	15.6	13.0	10.4
background	40.7	22.2	11.4
S/B	0.384	0.704	1.373

Optimized NN with drop out layers

	0.75 NN cut	0.80 NN cut	0.85 NN cut
signal	18.1	15.4	12.3
background	48.8	29.6	12.2
S/B	0.371	0.612	1.484

Signal-background yields in ttbar

Simple NN

	0.75 NN cut	0.80 NN cut	0.85 NN cut
signal	778.1	778.1	778.1
background	732.2	732.2	732.2
S/B	1.1	1.1	1.1

Optimized NN

	0.75 NN cut	0.80 NN cut	0.85 NN cut
signal	567.2	487.0	387.7
background	337.9	246.6	160.1
S/B	1.7	2.3	3.5

Optimized NN with drop out layers

	0.75 NN cut	0.80 NN cut	0.85 NN cut
signal	570.8	492.8	397.1
background	240.7	173.9	116.7
S/B	2.4	3.3	4.9

Summary & ongoing

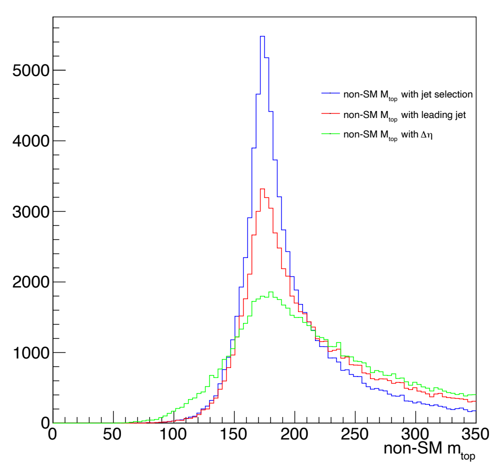
- tW signal has been studied and all variable distributions are made. Like ttbar channel, the non-SM top mass is the best discriminator.
- Several ML classifiers are trained using subset of data and important analysis features.
- After model (NN and RF) optimization, both have good performance. NN gives higher accuracy score, TPR and lower FPR. Overfit observed in RF.
- NN models (simple and optimized) are applied to the whole dataset and the NN weights are saved in a separate tree.
- Roostat and pyhf are used as the main statistical frameworks to compute, significance, p-value and upper limits.
- Analysis tree production with important variables and plotting framework are done (tree framework, plotter framework, ML weights, Statistical fits).
- As the next step, we can generate the signal events for separate coupling (S=1, V=1, T=1) and rerun all the fits to get 95% CL upper limits.
- Your feedback is welcome and appreciated.

Backup

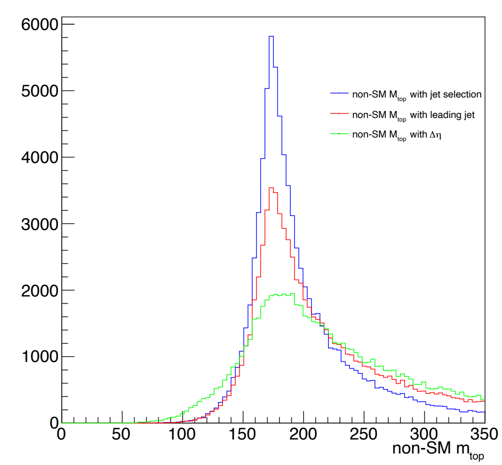
tW channel non-SM top mass reconstruction

- Three algorithms used to reconstruct non-SM top mass:
  1. the min $\Delta\eta$ between electrons is used to select OP electrons and subsequently non-SM top mass reconstruction (green)
  2. the leading non-btagged jet and the 3 electrons are the inputs for $\min(|m_{llq} - m_{top}|)$ to choose the best selection for OS electrons (red)
  3. Loop over all the electrons and jets to get $\min(|m_{llq} - m_{top}|)$. The combination will be used to indicate OS leptons (blue)

Signal charm



Signal up



Signal and background generation

- Signal and background events are generated with MG5 (for ME) + PYTHIA (for PS and HAD) + Delphes (for HLLHC CMS card detection). almost 3M events for both charm and up signals and 2M events for each background.

- Weights look fine (<1) for all signal and background events. Extra 15M $t\bar{t}$ events are being generated to have better ML training (the third lepton in $t\bar{t}$ should be fake btw).

- Here is the weight summary for all analysis processes:

  weights = {'ttbarZ': 0.00431, 'tZ': 0.00375, 'tttt': 2.79520e-05, 'ZZ': 0.67125, 'ttbar’: 0.9485, 'ttbarW': 0.00015, 'WZ': 0.13575,'signal_charm': 0.01376, 'signal_up': 0.01376}

- The preselections applied:
  1. exactly 3 leptons (for now just electrons) with one pair of OS
  2. at least 2-jets with one b-tagged jet
  3. minimum P_T cut and η cut to pass di-lepton trigger

Simple NN structure

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input (InputLayer)          [(None, 13)]              0

 hidden1 (Dense)             (None, 20)                280

 hidden2 (Dense)             (None, 20)                420

 output (Dense)              (None, 1)                 21

=================================================================
Total params: 721
Trainable params: 721
Non-trainable params: 0
```

Keras-tuner to tune NN Hyperparameters

```
RandomizedSearchCV(cv=5,
                   estimator=Pipeline(steps=[('scaler', StandardScaler()),
                                             ('clf',
                                              <keras.wrappers.scikit_learn.KerasClassifier object at 0x7fe642490880>)]),
                   n_iter=5,
                   param_distributions={'clf__activation': ['selu', 'relu',
                                                            'tanh'],
                                        'clf__batch_size': [64, 128, 256, 512],
                                        'clf__dropout_rate': [0.1, 0.01],
                                        'clf__epochs': [5, 10, 15, 50, 100,
                                                        200],
                                        'clf__k_initializer': ['lecun_normal',
                                                               'normal'],
                                        'clf__network_layers': [(32, 32),
                                                                (64, 64),
                                                                (128, 128,
                                                                 128)],
                                        'clf__optimizer': ['Nadam', 'Adam',
                                                           'SGD'],
                                        'clf__verbose': [0]},
                   scoring='accuracy')
```