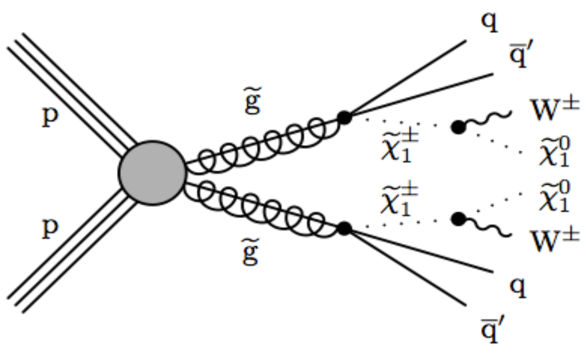
1L $\Delta\phi$ ANALYSIS

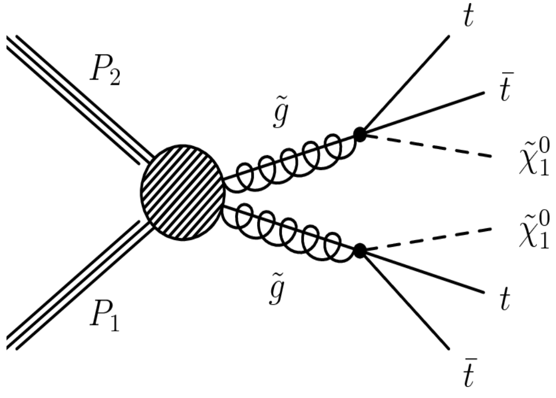
Ashraf Mohamed

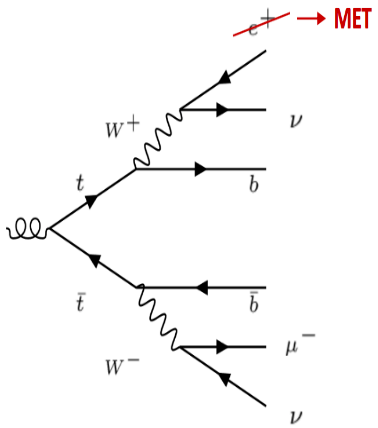
DESY, RWTH AACHEN III A

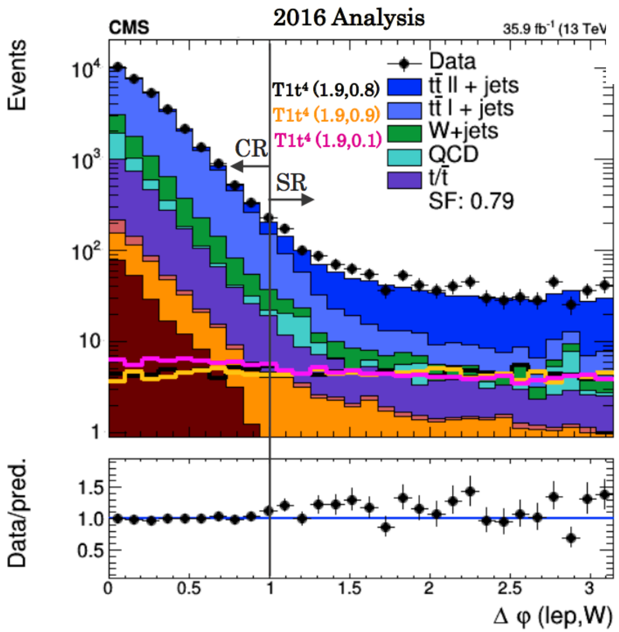
DESY : Dirk Krücker, Isabell Melzer-Pellmann

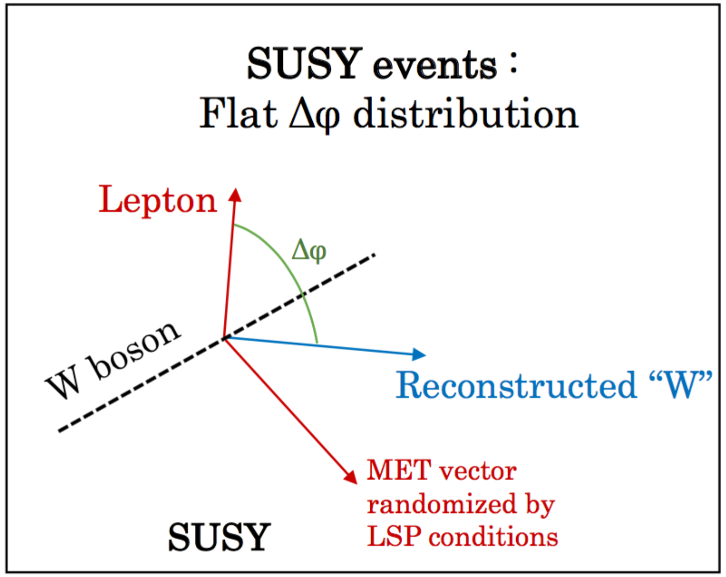
SIGNAL MODEL AND MAIN BACKGROUND

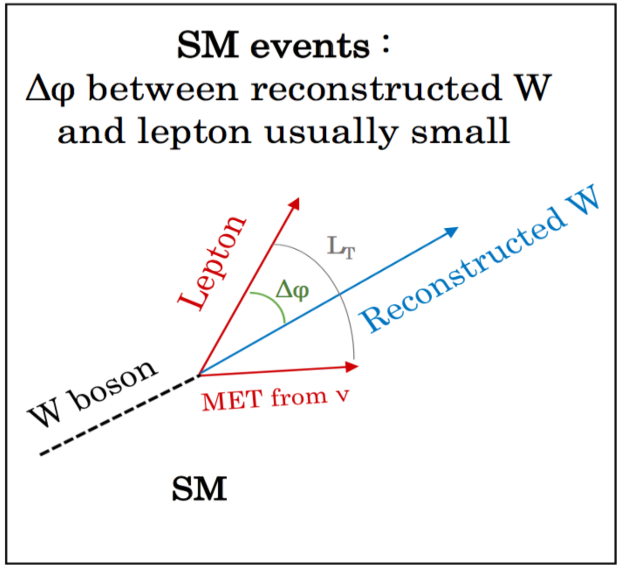












Search regions with large Δϕ binned in:

L_T =|Lep p_T| + |MET|

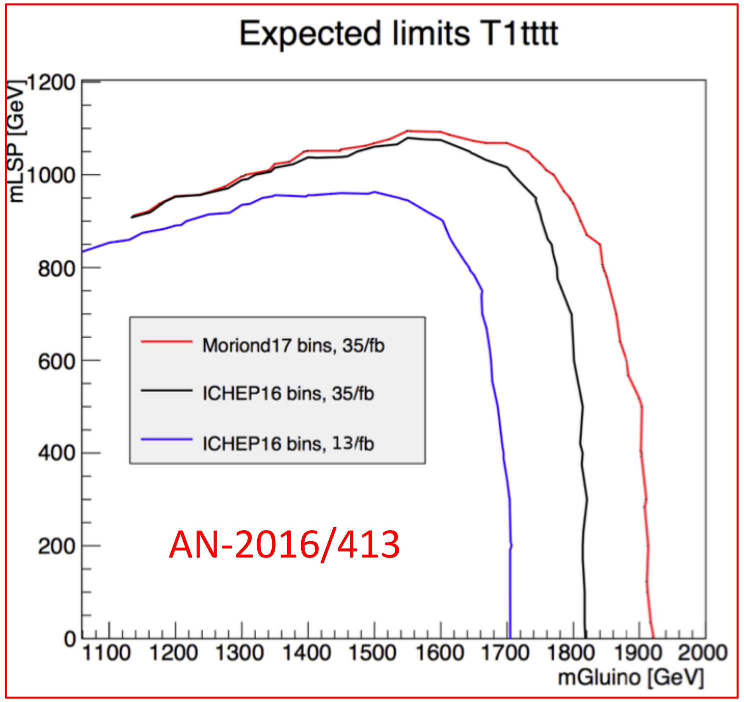
H_T= Sum(Jet p_T)

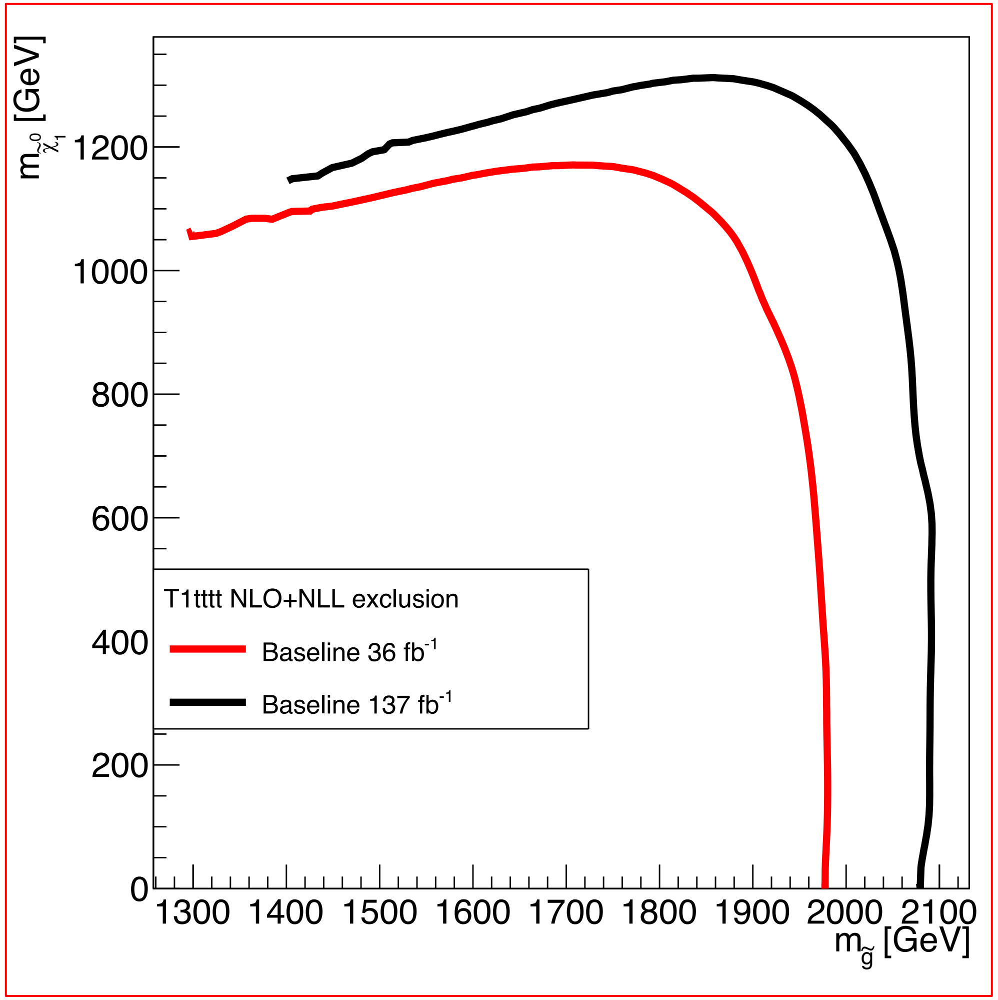
nJets

nBjets

BASELINE LIMITS RESULTS 137 fb^{-1}

- Plot has full RunII Limits
- Red is baseline using the new scan at 36 fb^{-1}
- Black is baseline using the new scan at 137 fb^{-1} using the full statistics of 16+17+18





DNN Multi-Class

Peter Fackeldey, RWTH Aachen 2018, M.Sc.
Marcel Rieger, RWTH Aachen

DNN DETAILS

- All classes are weighted to the same level (exactly as TMVA) only for training, physics weight Is used for evaluations
- Independent sample is used for training and testing
- **SoftMax activation function in the output layer transforms the output score into probability with sum = 1 for all classes**

256 nodes

Input Features

$MET, MT, Jet_{p_T}^{1/2}$

$Lep_{p_T}, L_T, H_T, n_{bJets}$

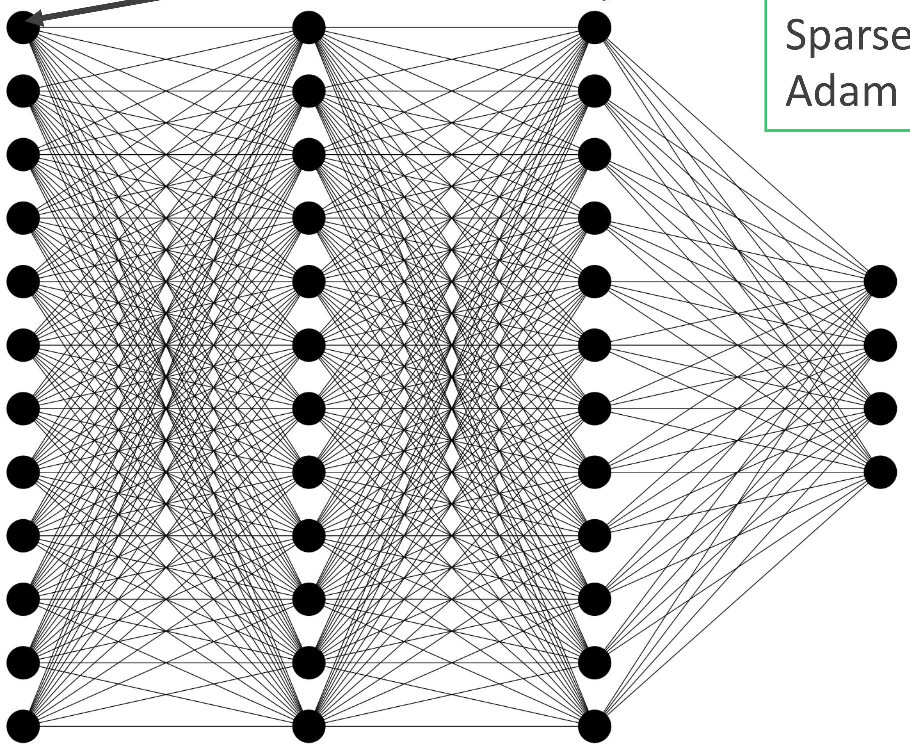
$n_{Top}, n_{jets}, \Delta\phi, Lep_{relIso}$

$Lep_{MiniIso}, IsoTrack_{p_T}$

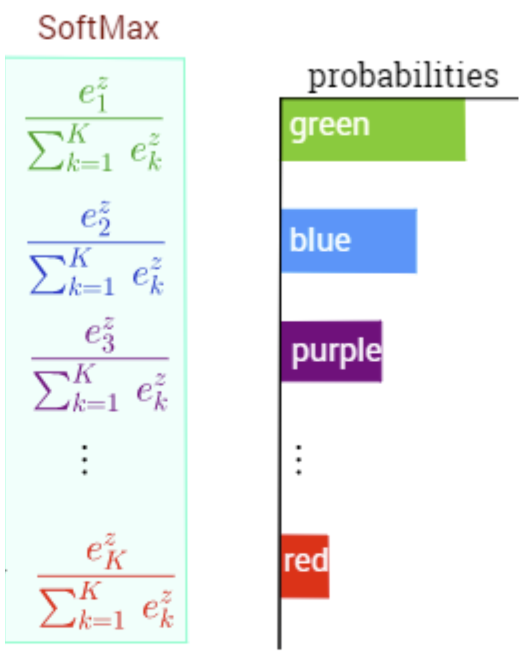
$IsoTrack_{MT_2}$

$m_{\tilde{g}}, m_{\chi_1^0}$

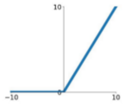
17 input variable



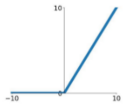
Sparse categorical crossentropy loss with Adam Optimizer (learning rate 0.01)



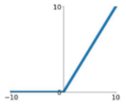
ReLU $\max(0, x)$



ReLU $\max(0, x)$



ReLU $\max(0, x)$

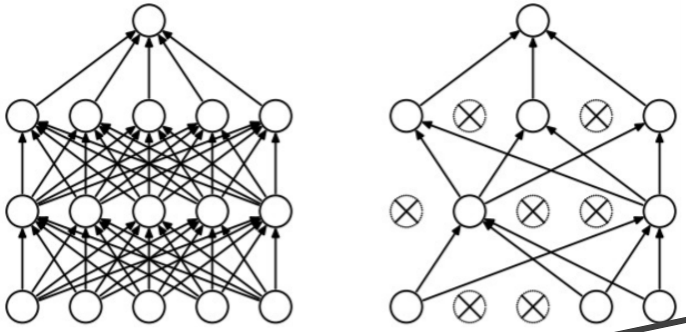


DNN DETAILS

100 epochs, batch size = 2048
Dropout with rate of 0.01

Early stopping to stop training once the model performance stops improving on a hold out validation dataset with patience of 10 epochs.

Preselect events :
- I Lep , no Veto, LT > 250 , HT > 500, nJ > 3 nbJ >= 1, iso_track veto

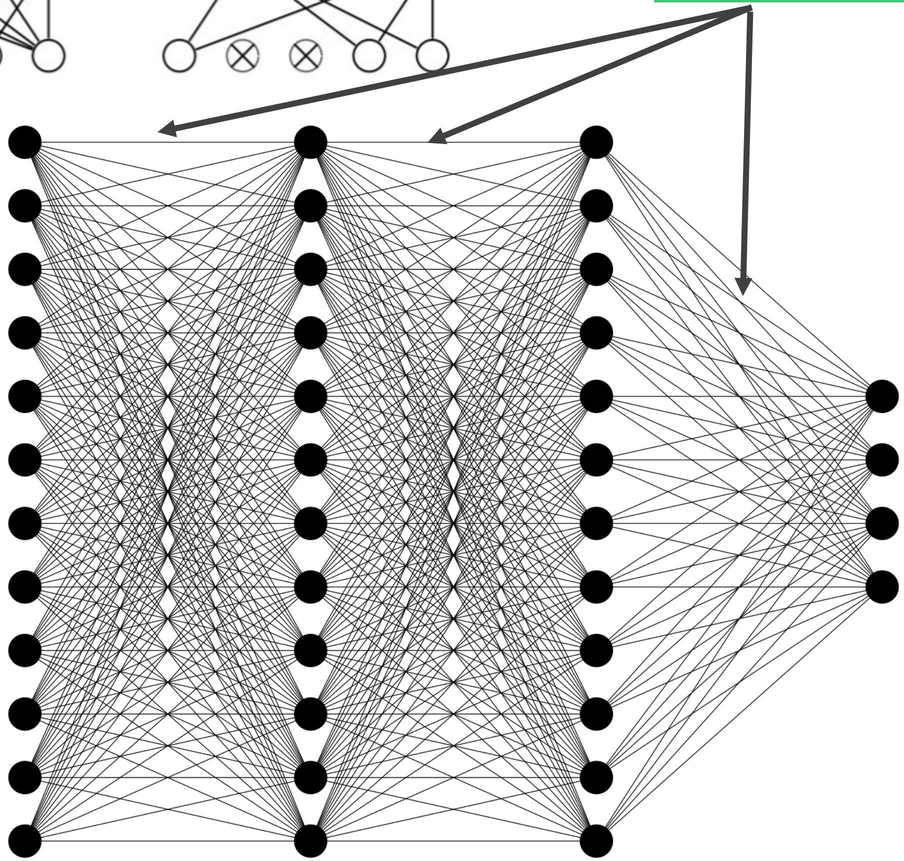


Dropout layer

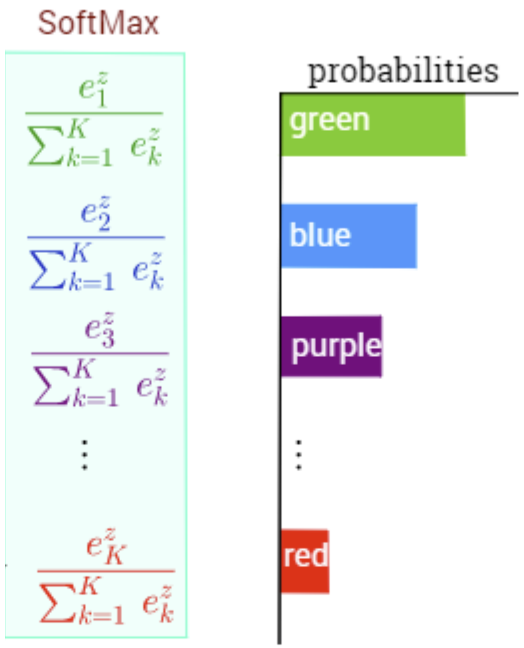
Input Features

$$MET, MT, Jet_{p_T}^{1/2}$$
$$Lep_{p_T}, L_T, H_T, n_{bJets}$$
$$n_{Top}, n_{jets}, \Delta\phi, Lep_{relIso}$$
$$Lep_{MiniIIso}, IsoTrack_{p_T}$$
$$IsoTrack_{MT_2}$$
$$m_{\tilde{g}}, m_{\chi_1^0}$$

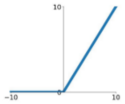
17 input variable



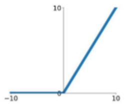
Class 0 → DiLep Ttbar
Class 1 → SemiLep Ttbar
Class 2 → Wjets + others
Class 4 → Signal



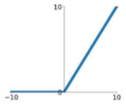
ReLU $\max(0, x)$



ReLU $\max(0, x)$



ReLU $\max(0, x)$



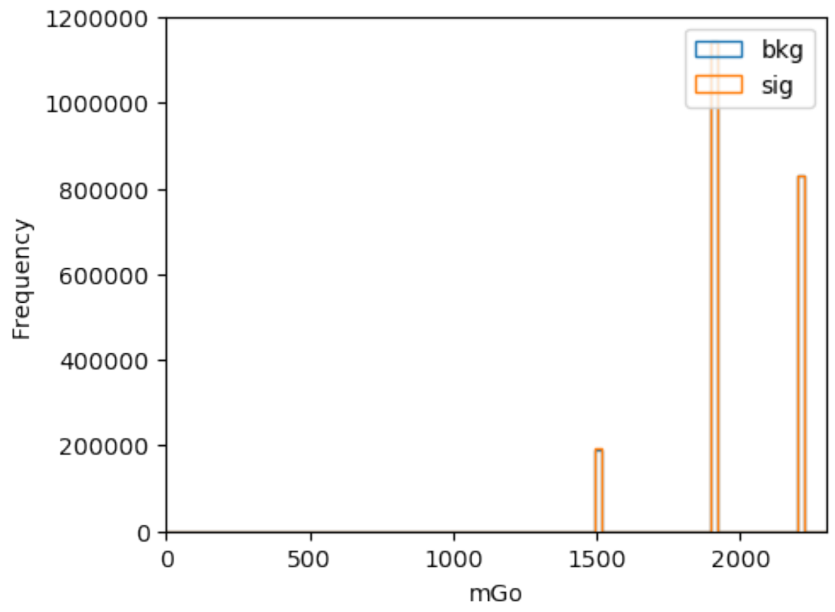
PARAMETRIZED DNN

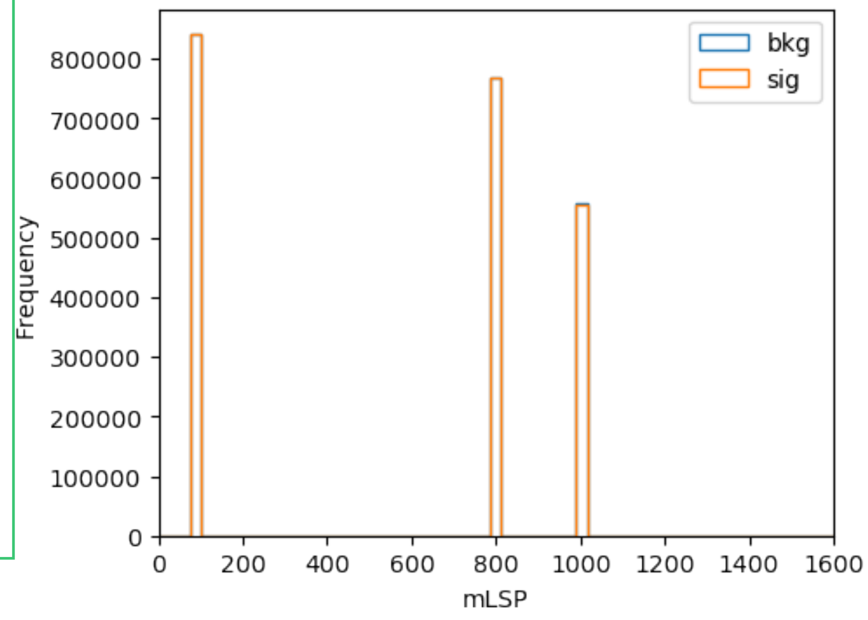
- List of mass points used for this training
  (1.9/0.1 – 2.2/0.1 – 2.2/0.8 – 1.9/0.8 – 1.9/1.0)
- Need to define signal mass for both backgrounds and data as well.
- Detailed study on the implementation of parametrized training can be founded in this talk param. BDT
- For the backgrounds two options were considered:

1. Randomization: Assign mGo & mLSP randomly (out of list of signal masses) during DNN training to every background event. **(has a problem that each signal mass will be trained with part of background phase space)**

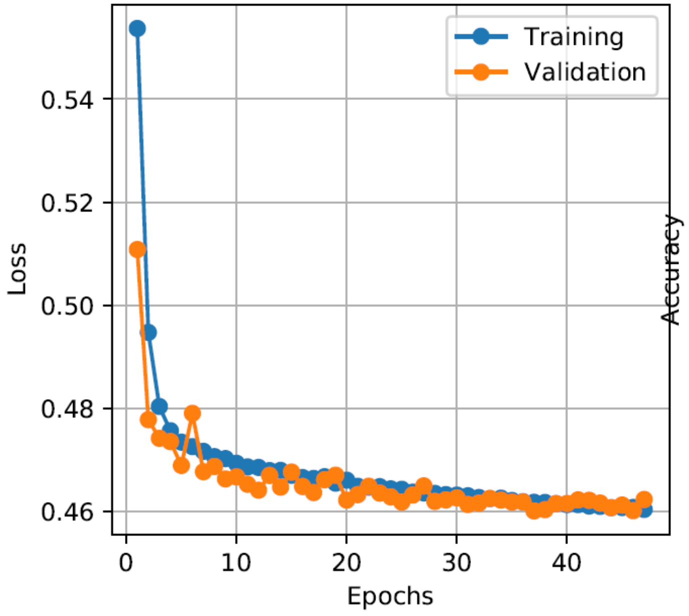
2. Over-sampling: Create multiple copies of a given MC background event. Each copy having same values of all reconstructed variables but a different mGo & mLSP value

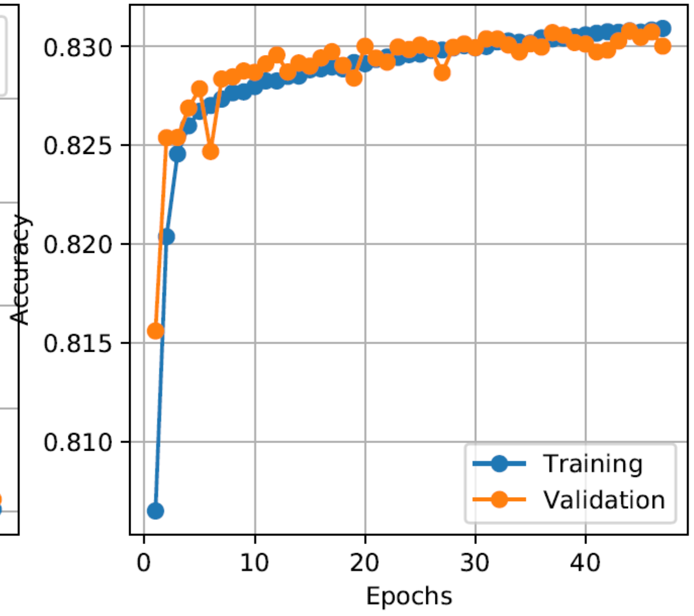
Full background mGo_1,mLSP_1 —·—· Full background mGo_n,mLSP_n

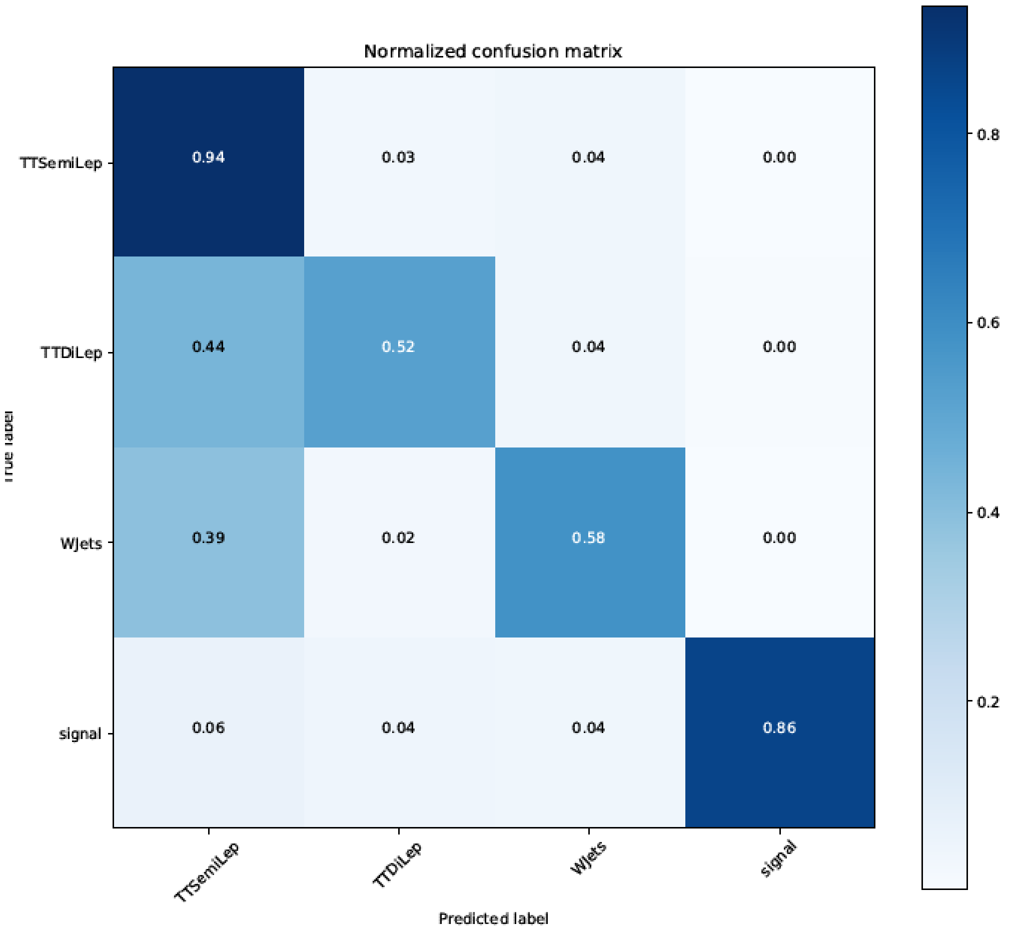




DNN PERFORMANCE

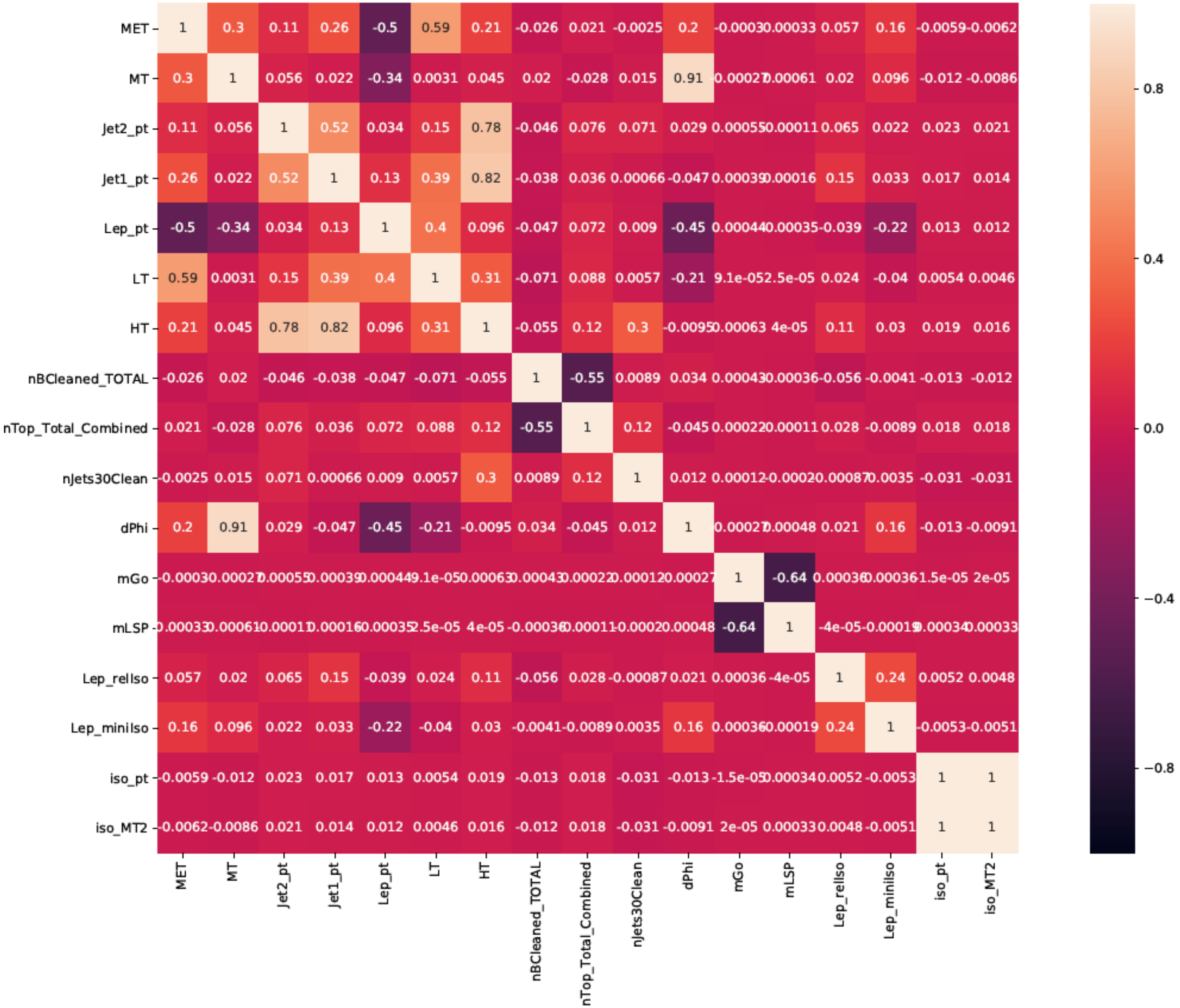






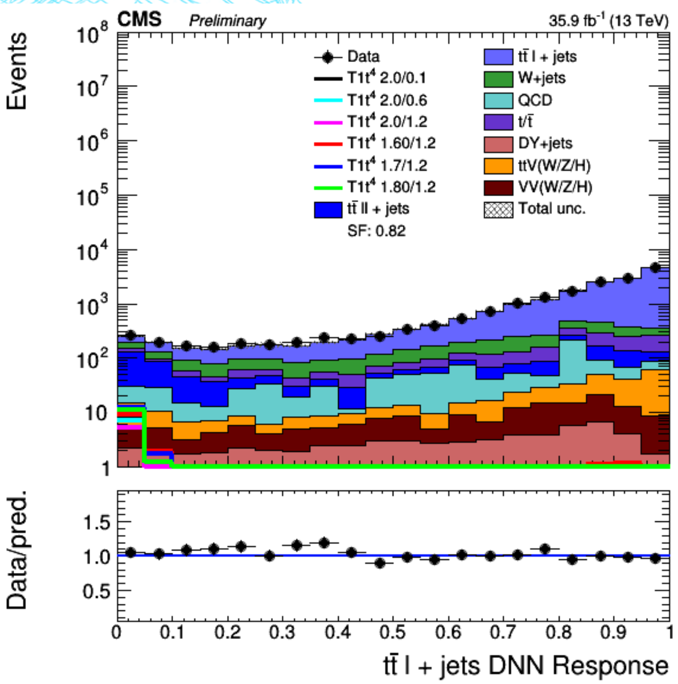
No over fitting at all and early stopping stops the training after ~ 45 epochs (nothing suspicious)

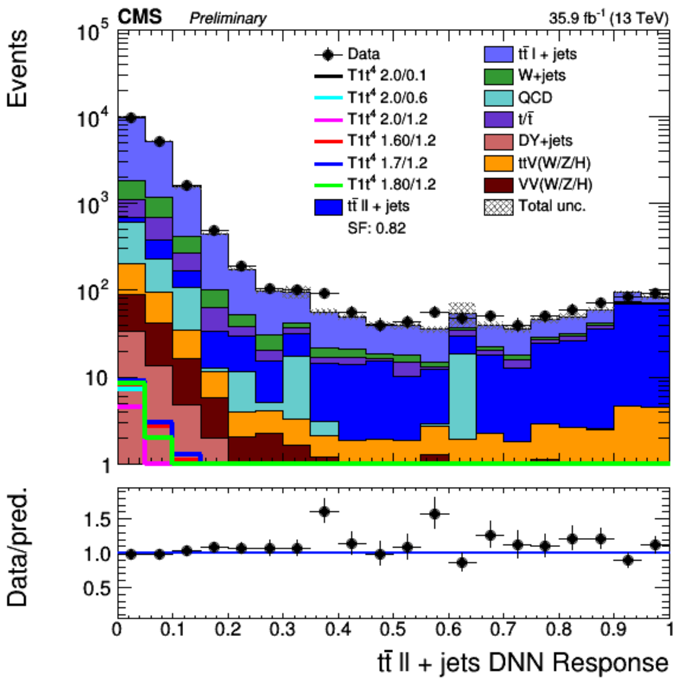
Correlation matrix

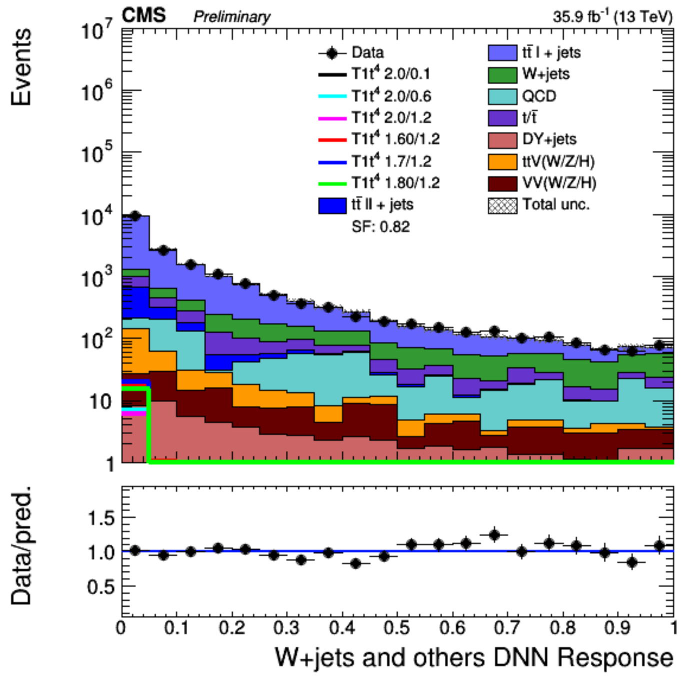


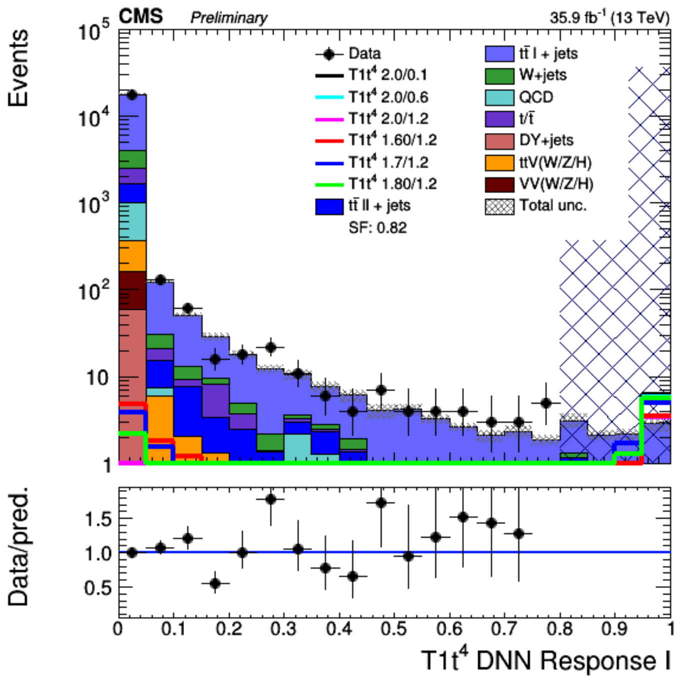
Class 0
(semiLep TTbar)

DNN PERFORMANCE





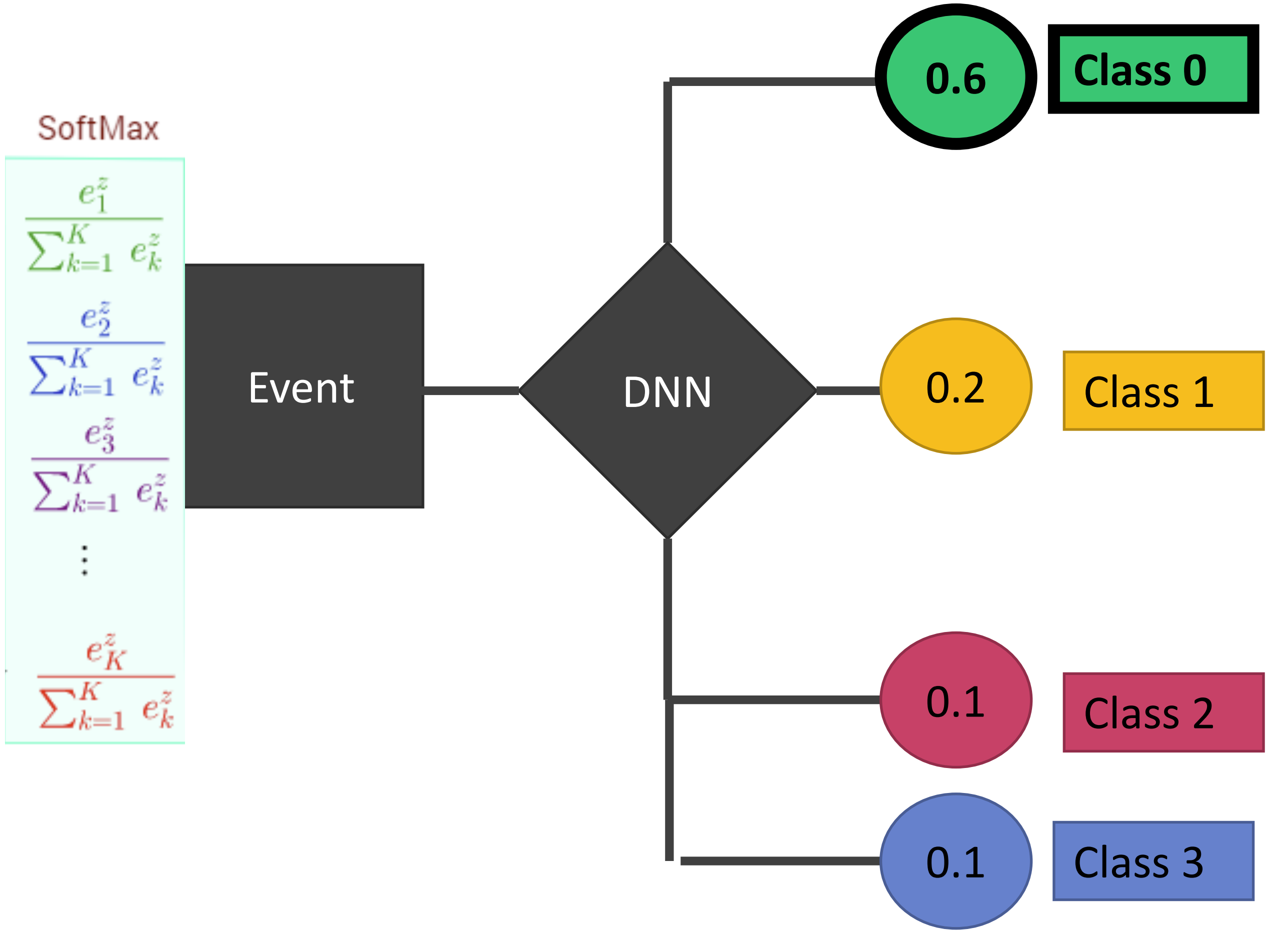




High Purity for each class, nice agreement, very useful for the background modeling
Nice Idea to get the 4 classes orthogonal (see next!)

DNN MULTICLASS FOR BACKGROUND

- The choice of the last layer activation function (SoftMax) makes all classes mutual exclusive by definition
- Each event will have a certain probability to belong to a certain physical process
- A single event can not land in two classes at the same time
- Exclusive class (category) will be defined as a result to the above definitions



Background normalization

- After doing the event-wise categorization each category we can have and exclusive background since all the classes are mutually exclusive
- Data/MC SFs are applied per event [LepSFs, btagSFs, ……] * Xsec * lumi * 1000 / total weight
- Each category is defining a control region (SemiLep TT – Dilep TT – WJ+others)
- Each control region will be normalized (fitted) to data independently with specific SFs
- SFs are written in every control plot (see next slides)
- Background in SR will be corrected by solving the following system of equations :

$$Y_i = A_{ij} X_i$$

$$Y_i \rightarrow \text{ data counts}$$

$$A_{ij} \rightarrow \text{MC counts}$$

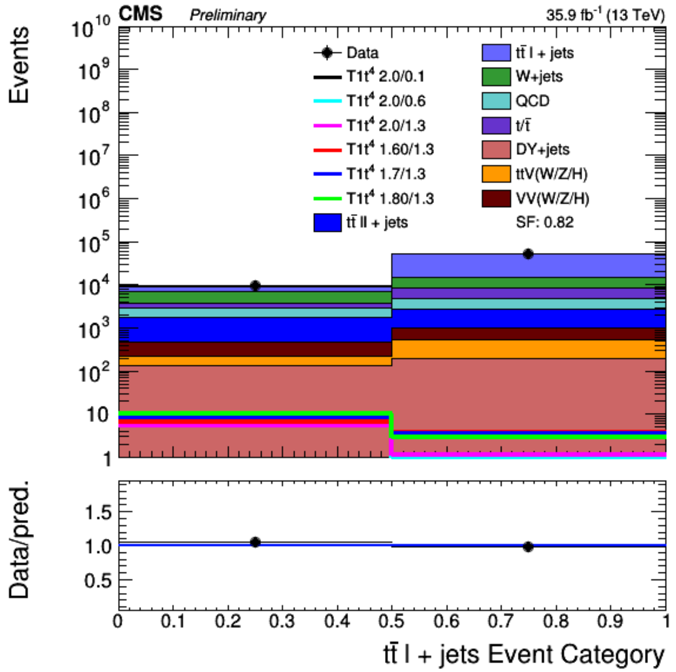
$$X_i \rightarrow \text{scale factor for each background}$$

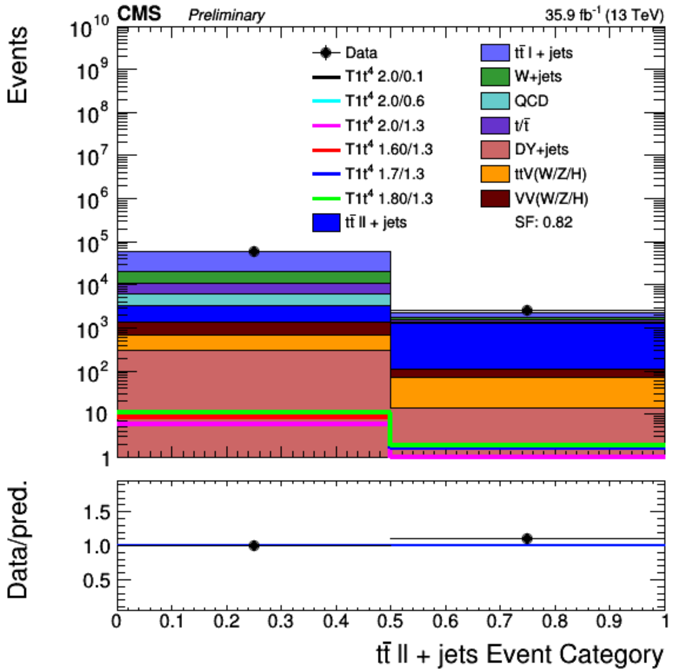
$$i \rightarrow \text{number of CRs}$$

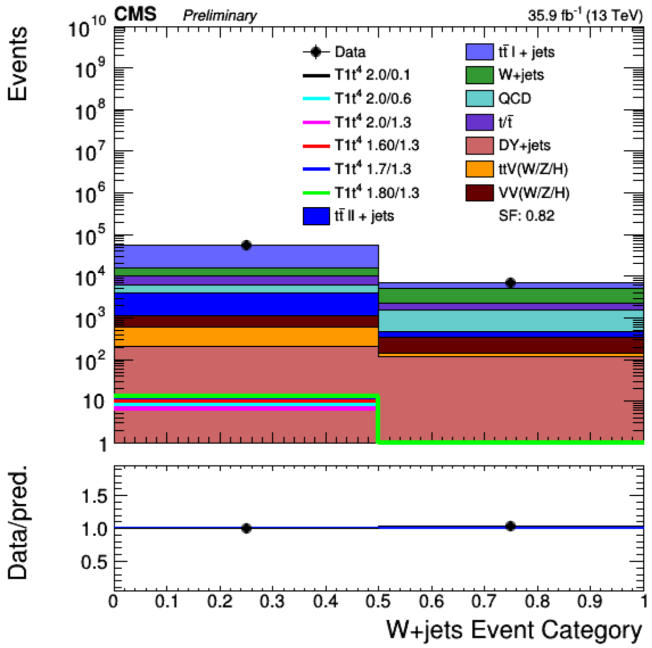
$$j \rightarrow \text{background number}$$

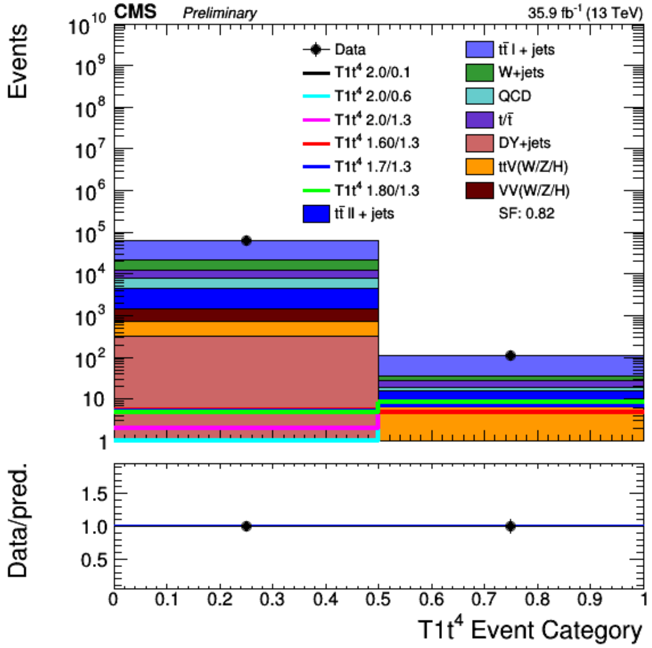
- We can account for the 2nd or 3rd order backgrounds if they have non-negligible contributions

EVENT WISE CATEGORIES







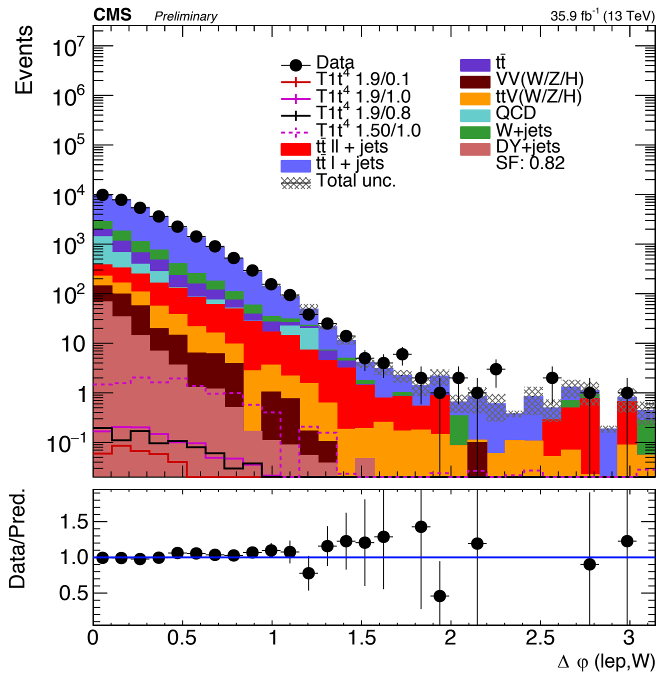


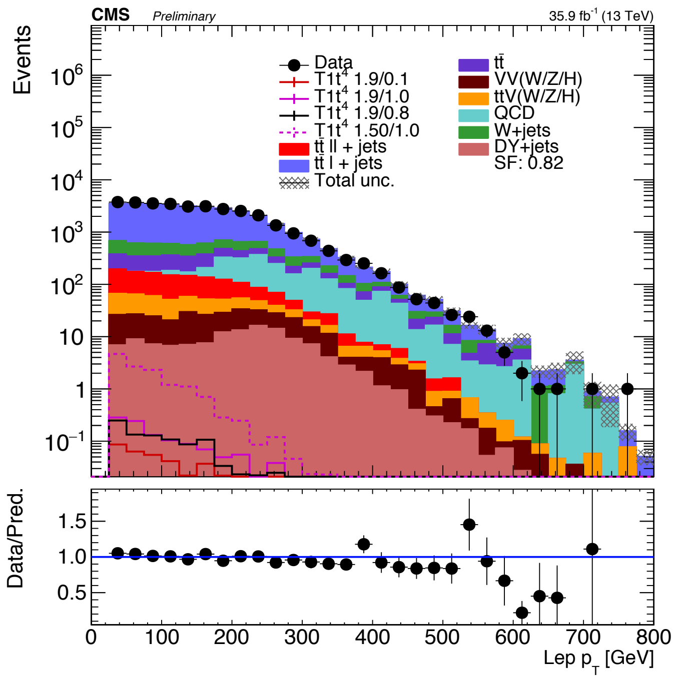
Preselect events :
- I Lep , no Veto, LT > 250 , HT > 500, nJ > 3 nbJ >= 1, iso_track veto

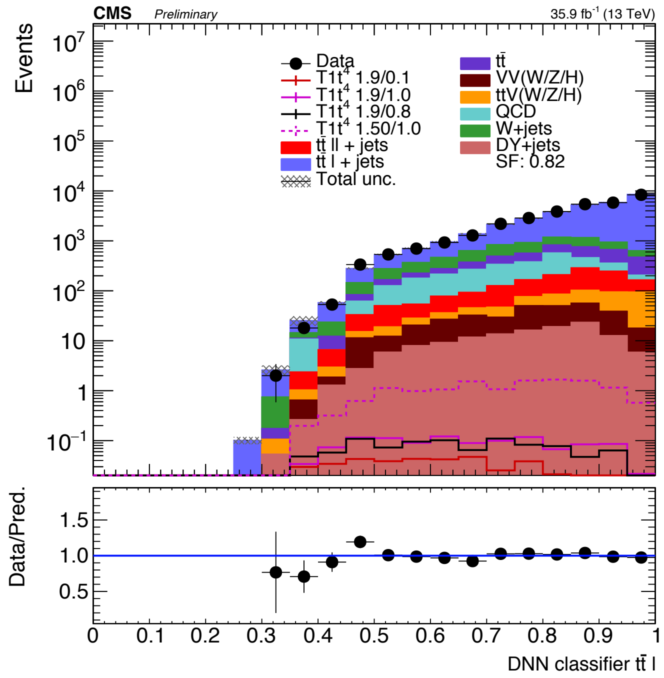
1900/100

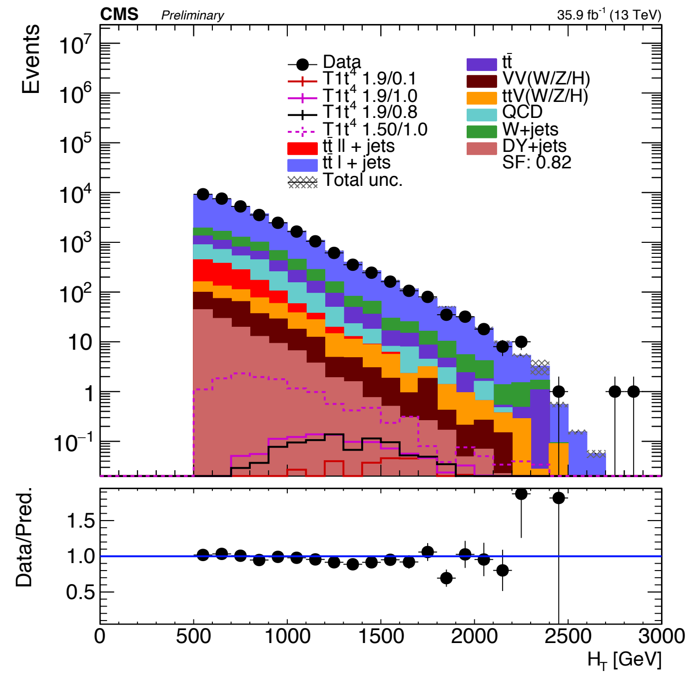
SEMILEP TT CAT

1900/100

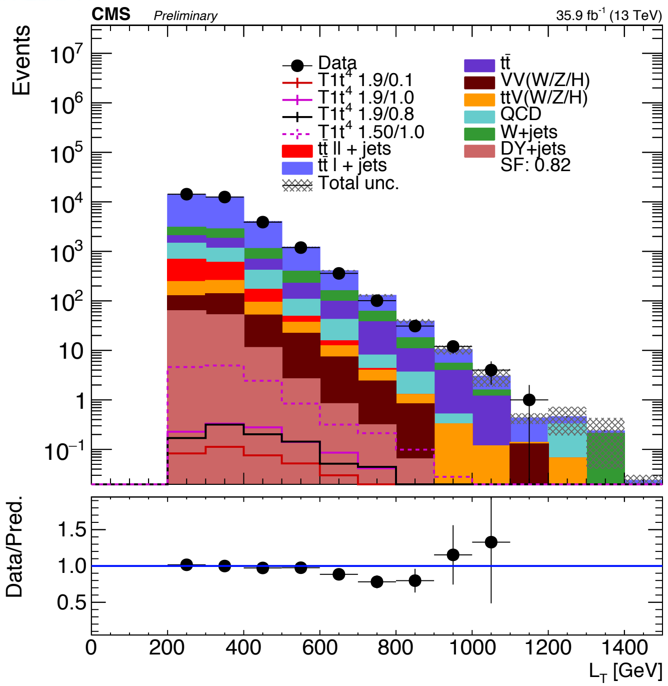


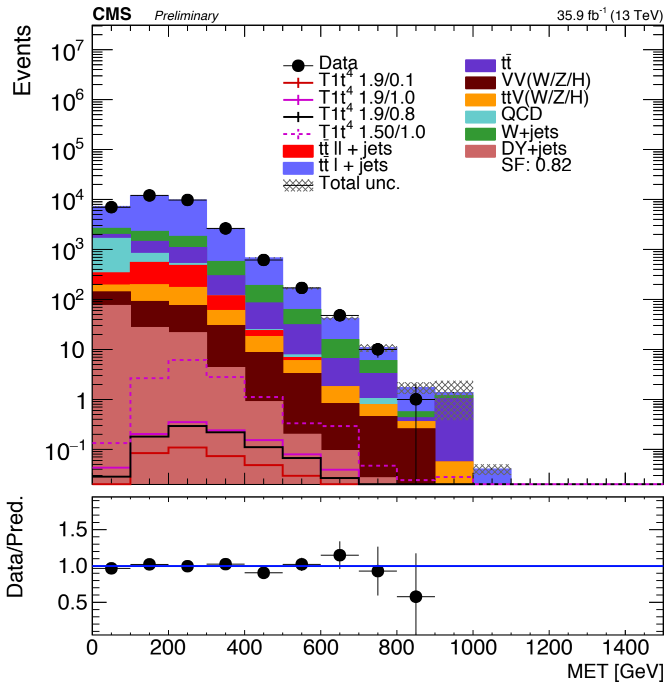


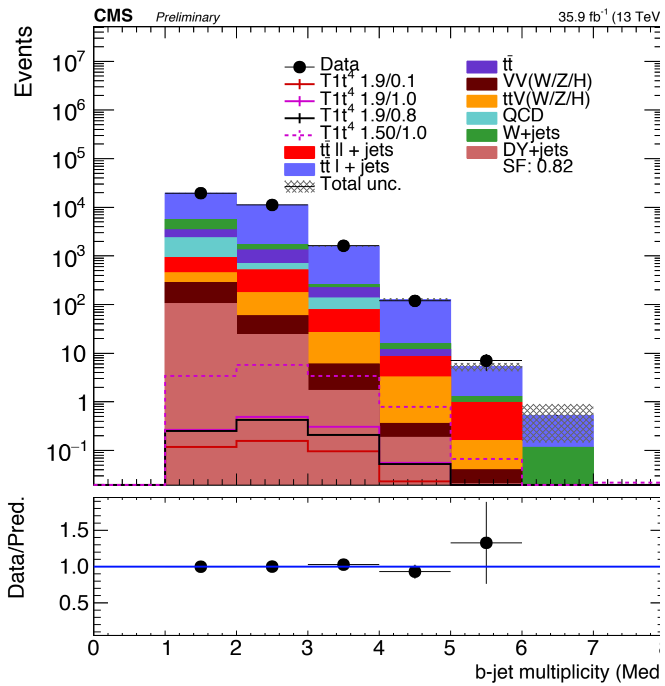


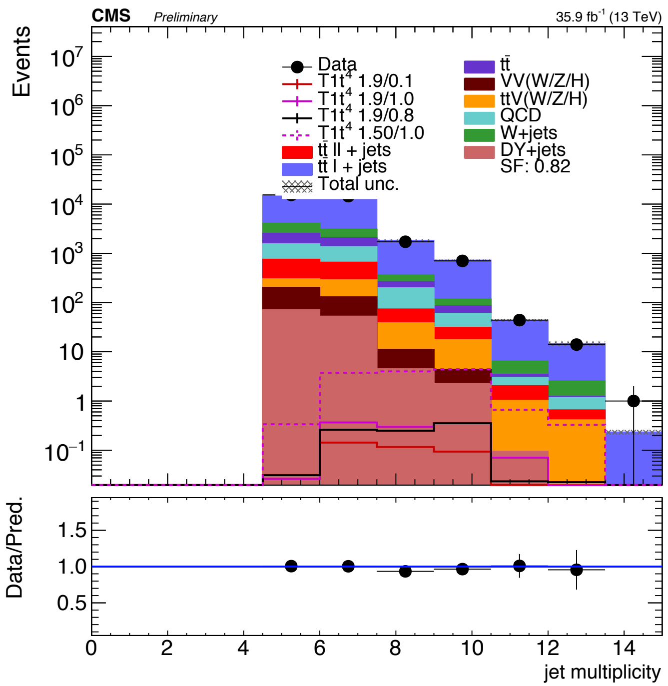


SEMILEP TT





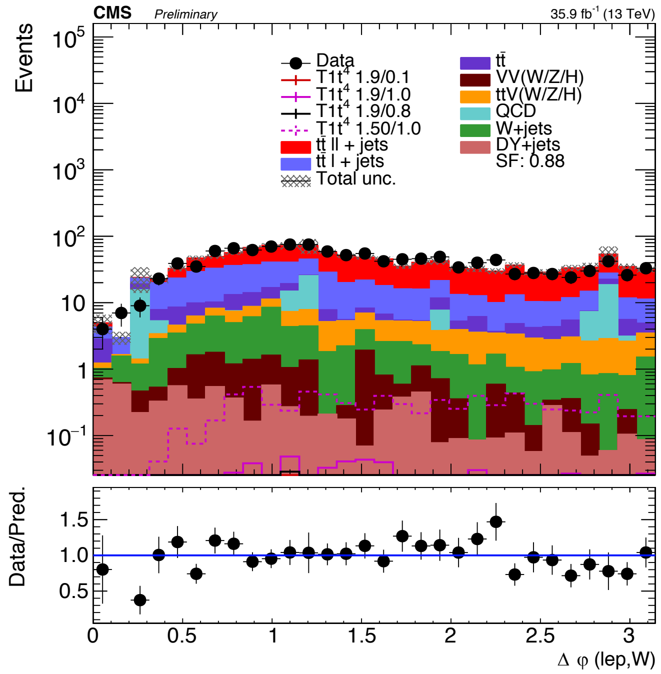


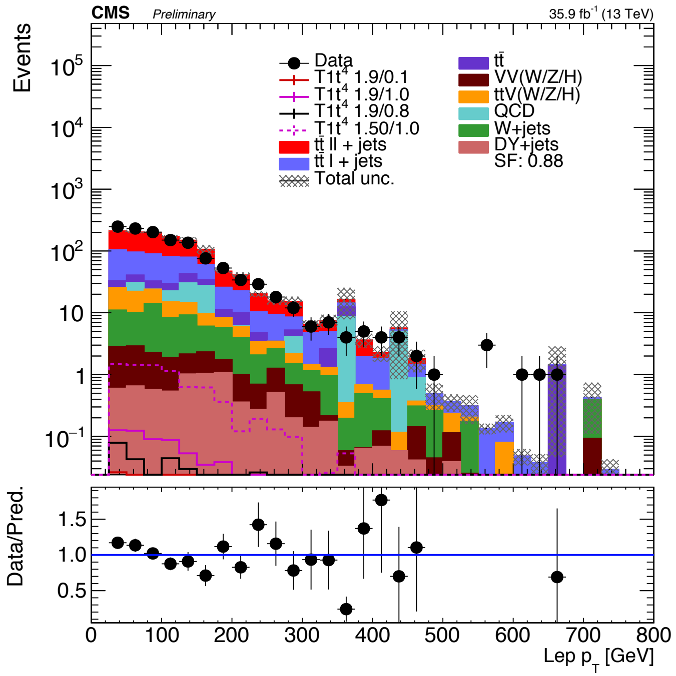


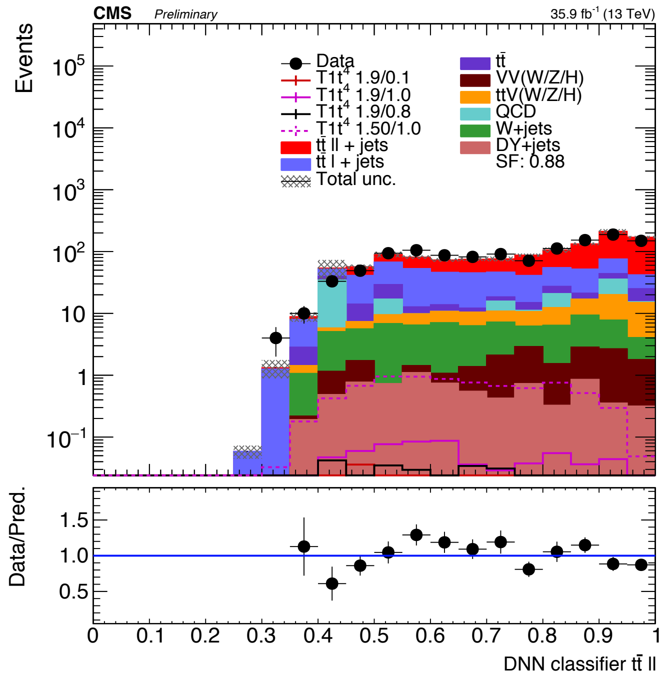
1900/100

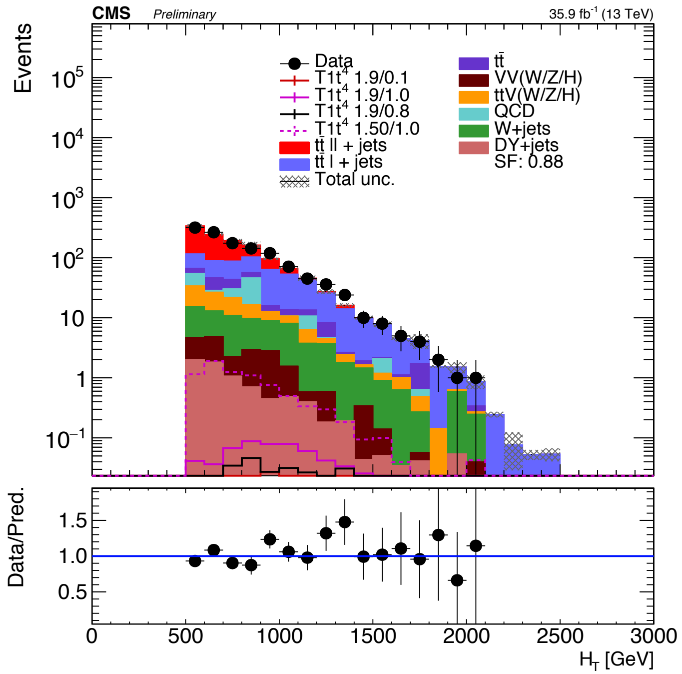
DI-LEP TT CAT

1900/100



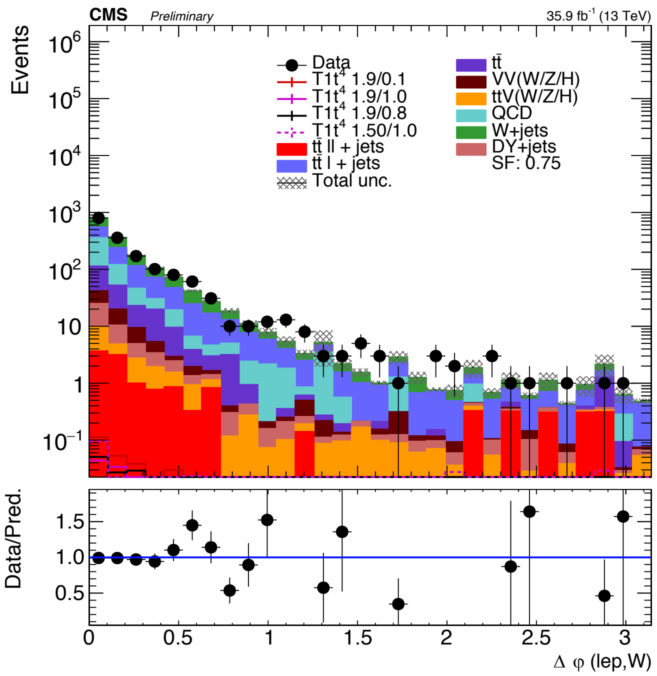


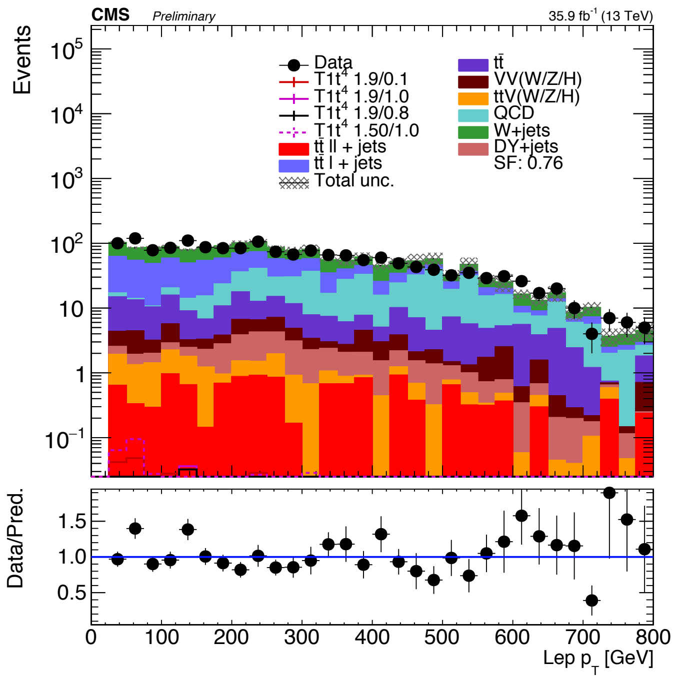


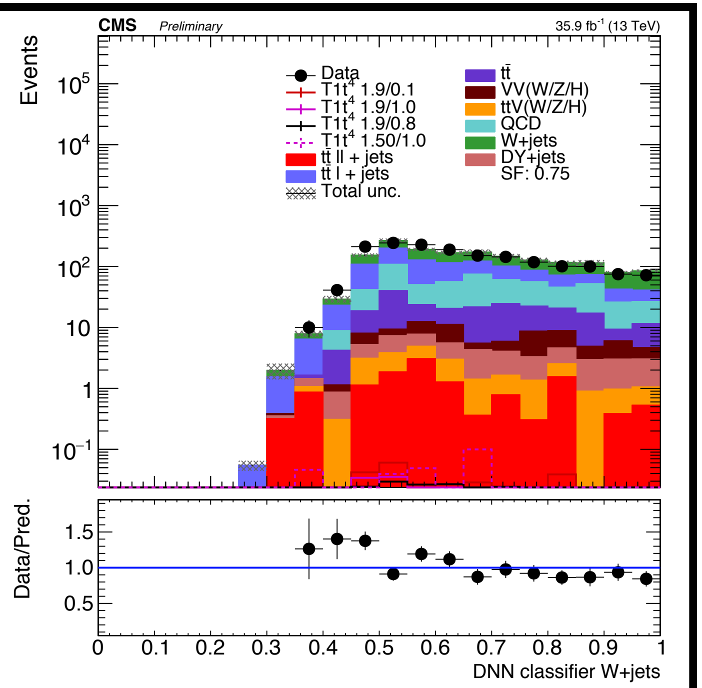


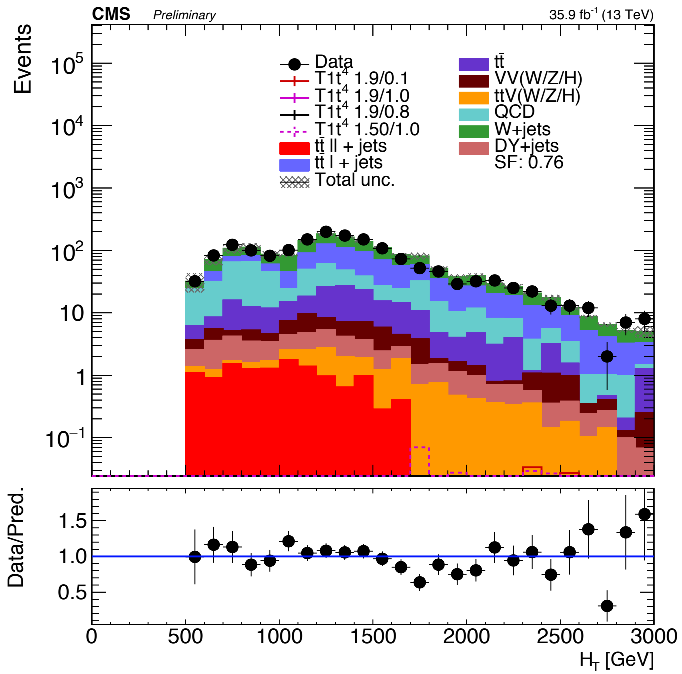
W+JETS CATE

1900/100



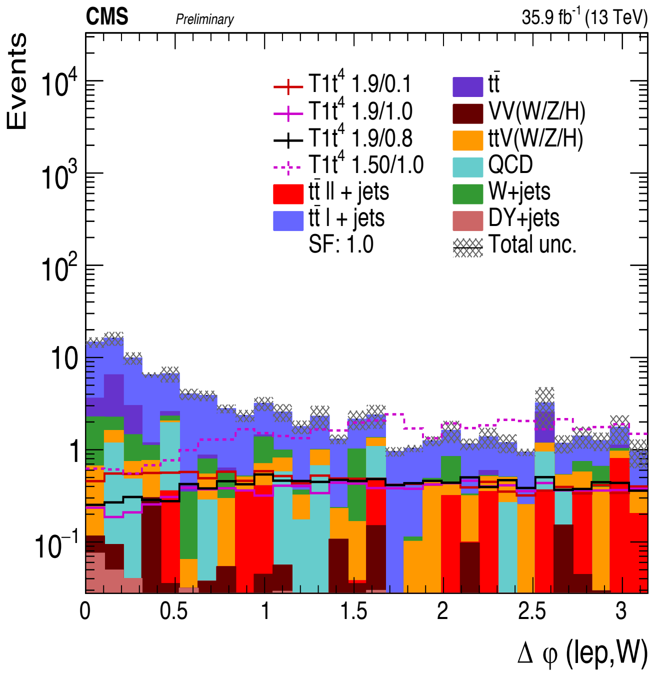


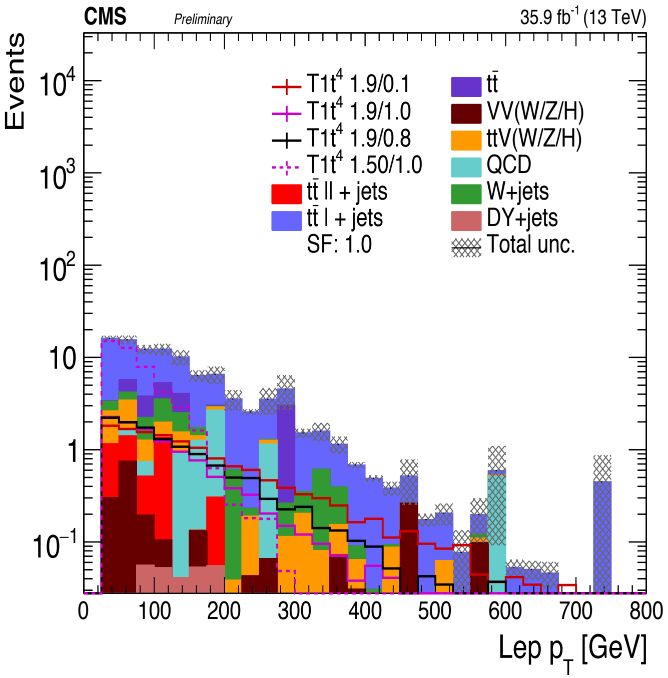


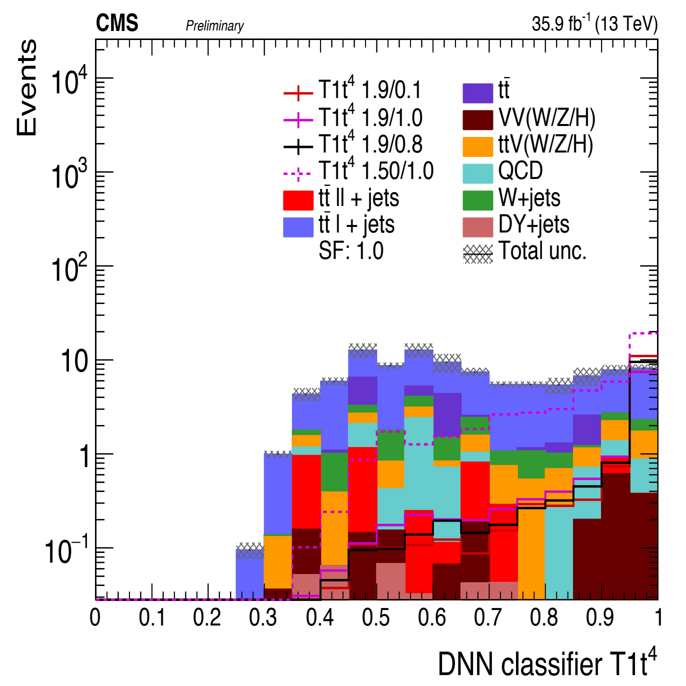


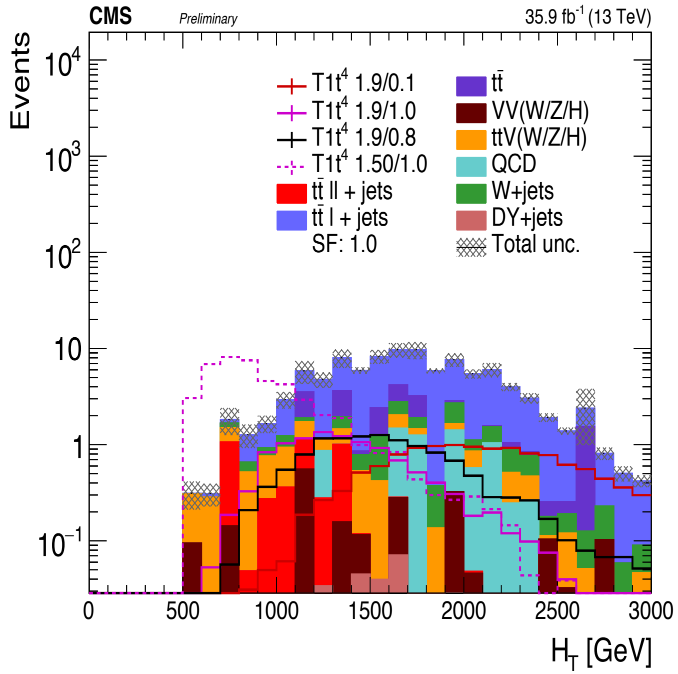
SIGNAL CAT

1900/100









BACKGROUND

$$\begin{bmatrix}\alpha\\ \beta\\ \gamma\end{bmatrix} \times \begin{bmatrix}A_{11} & A_{12} & A_{13}\\ A_{21} & A_{22} & A_{23}\\ A_{31} & A_{32} & A_{33}\end{bmatrix} = \begin{bmatrix}data_1\\ data_2\\ data_3\end{bmatrix}$$

For background 1900/100

	SemiLepTT	DiLepTT	others	Data
SemiLepTTCat	30183.96	1064.66	8270.63	32391.0
DiLepTTCat	401.55	686.69	314.39	1228.0
othersCat	57.59	0.31	76.91	93.0

==

alpha	beta	gamma
0.89	1.02	0.54

For background 1500/1000

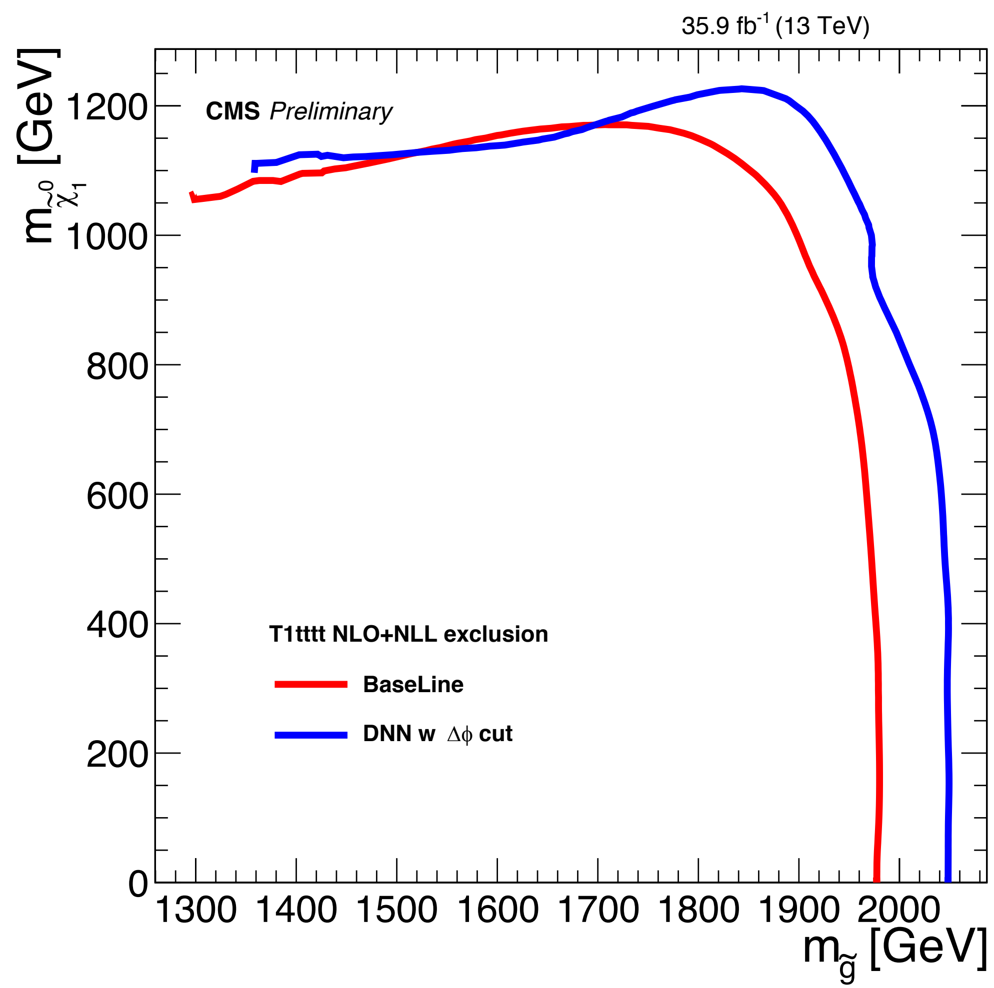
	SemiLepTT	DiLepTT	others	Data
SemiLepTTCat	30116.42	1030.97	8201.25	32222.0
DiLepTTCat	439.06	687.06	319.14	1285.0
othersCat	7.59	0.0	10.5	16.0

==

alpha	beta	gamma
0.78	0.93	0.96

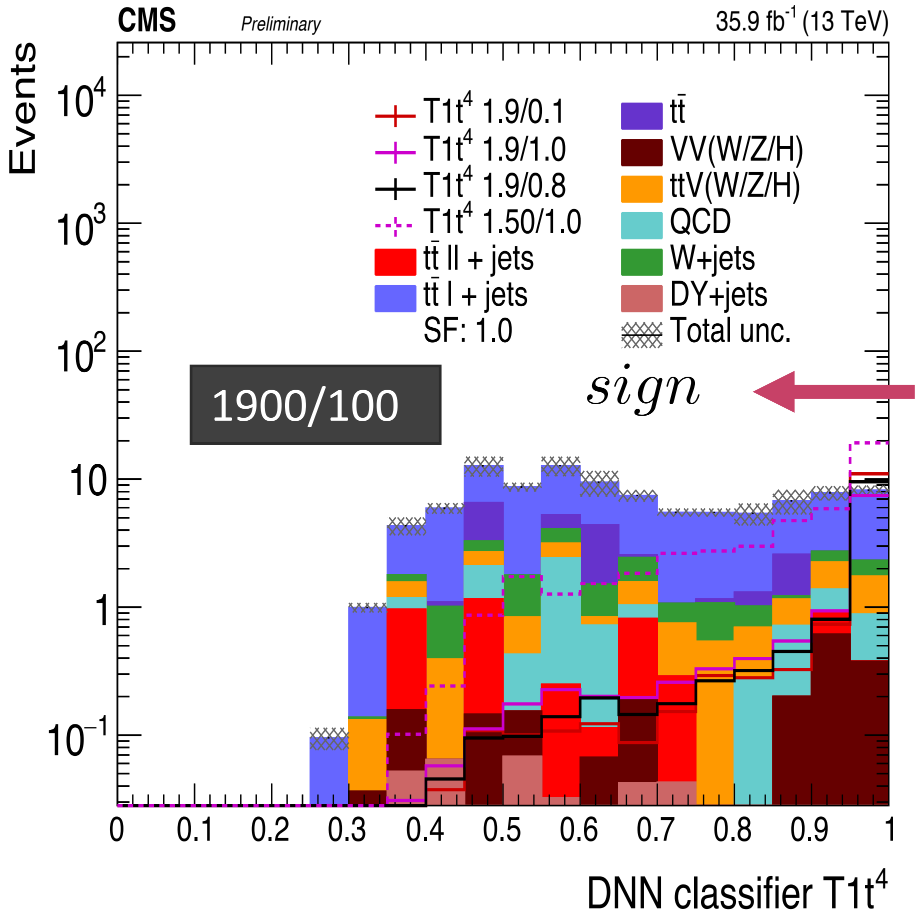
For every signal I choose the closest background

DNN LIMIT PERFORMANCE



$$sign = \frac{s}{\sqrt{s+b+(0.2b)}}$$

dPhi cut not visible here



Calculate the significance starting from the last bin and merge bins when getting better significance

A. Mohamed, SUSY 1Lep ,DESY, RWTH

SUMMARY

- BDT has a good performance but no simple background model
- Binary classification with DNN has been investigated but no simple background model
- Parametric DNN training is used with oversampling background method
- MultiClass DNN can be a good candidate using the categorization method
- Nice performance on the Multicalss DNN shape analysis and proper background model is very possible
- ttH(bb) (SM) observation & tt resonance (BSM) search have already tested the background categorization methods with MulitClass DNN successfully [HIG-17-026,AN-2017/063 and more references in our CERNBox area]
- May be we can test having 2 signal classes instead of one (compressed class + non compressed class) or have 2 different multiclass networks for each of them (to be tested with enough statistics)
- **0b analysis can be straight forward to the multiclass DNN (already skimmed) (different training will be done for that)**

TO DO LIST

- Analysis items:

  1- limit improvements

  2- Background estimation → ongoing

  3- prefire estimate

  4- 2017 EEMETFix v2

  5- ISR study

  6- systematics

  7- check the SFs

  8- full expected limit setting

  9- documentation

  10- 2018 data?

  11- 0-b Analysis

Danke

شكرا

Thanks

ROCs

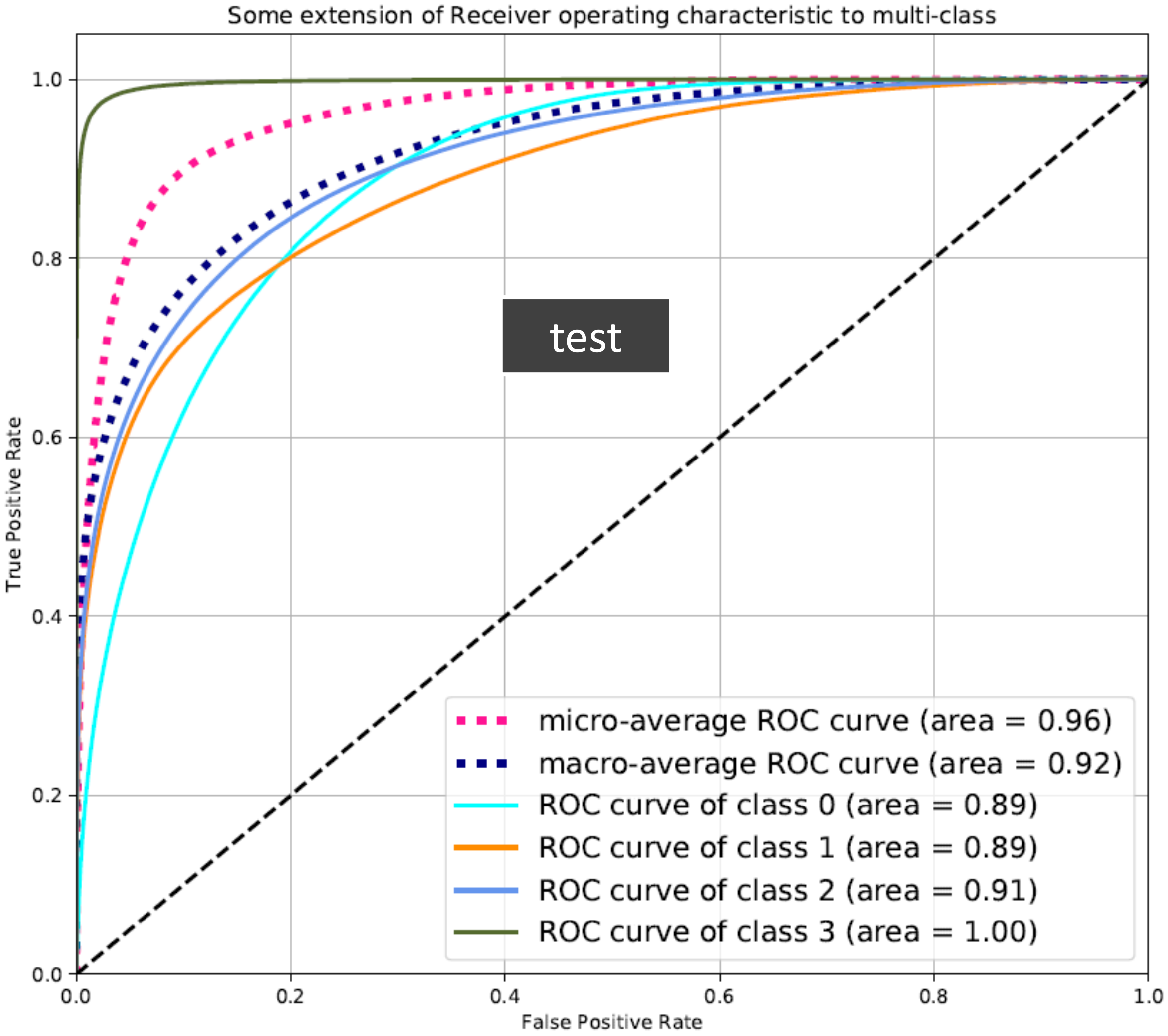
	Predicted Class	
Actual Class	**True Positive (TP)**	**False negative (FN)**
	False Positive (FP)	**True negative (TP)**

- "micro-average": AUC from the individual true positives, true negatives, false positives, and false negatives of the the k-class

$$Pred_{micro} = \frac{TP_1 + ... + TP_k}{TP_1 + ... + TP_k + FP_1 + ... + FP_k}$$

- "macro-averaging": average the performances of each individual class independently and calculate the average

$$Pred_{macro} = \frac{PRE_1 + ... + PRE_k}{k}$$



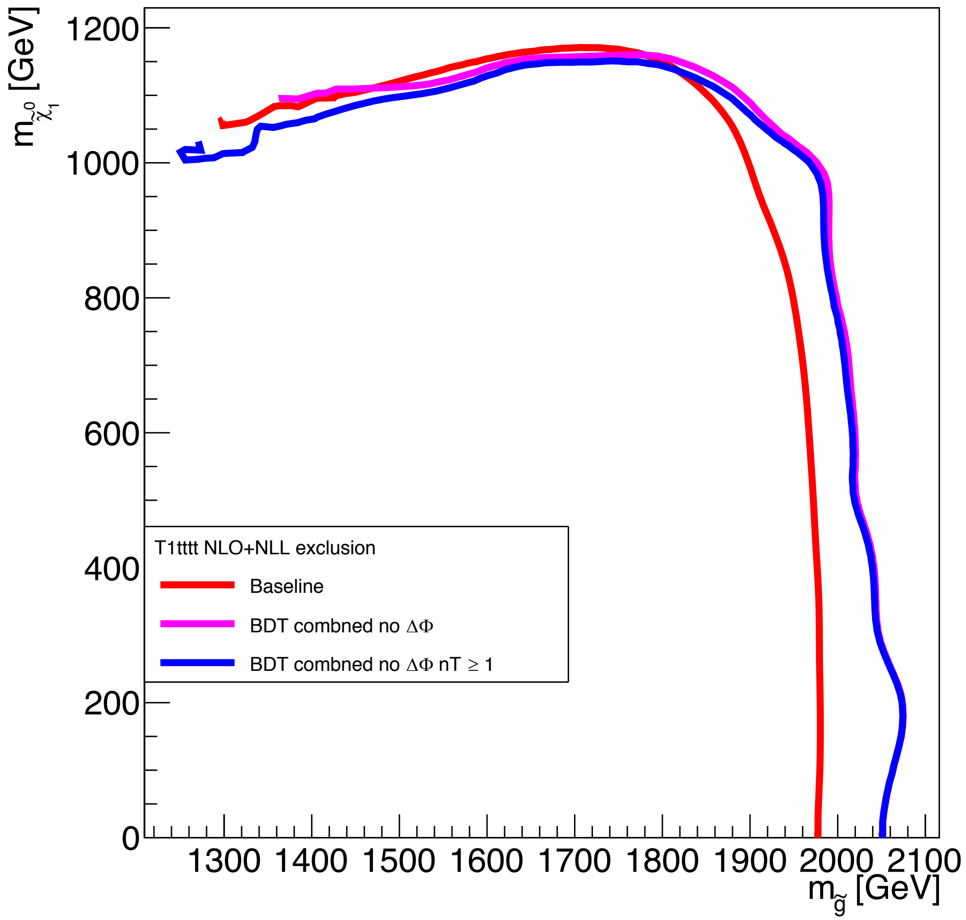
BDT

BDT DETAILS

Samples used in training	TTJets_GenMET, TTJets_Dilep_ext, TTJets_Semilep_ext, QCD_HT*_ext, DY_HT*_ext, Wjets_HT*_ext (excluded from evaluation and limit setting)
Signals used in training (privately produced)	SMS_T1tttt (1.5/1.0,1.9/0.1,1.9/08,1.9/1.0,2.2/0.1,2.2/0.8) Single classifier for combined mass points
Variable used in training	"Lep_pt", "Jet1_pt","Jet2_pt","MET","LT","HT","nJets30Clean", "nBCleaned_TOTAL","nTop_Total_Combined","nResolvedTop" Removed dPhi and MT from the training
Preselection	[nLep == 1 && Lep_pt > 25 && nVeto == 0 && nJets30Clean >= 5 && Jet2_pt > 80 && HT > 500 && LT > 250 && nbJets >= 1]
Number of sig/BKG passed the preselection	Background -- training and testing events: 3021536 Signal -- training and testing events: 473265
Weight applied during training	Xsec * all scalefactors

BDT LIMITS RESULTS 2016

- Plot has 2016 Limits
- Red is baseline using the new scan
- Pink line is the limit when used the new strategy of combined mass points
- Blue line is the limit when used the new strategy of combined mass points and apply nTop >= 1



BACKGROUND TEST

- When Δϕ is removed from the training and Used nTop >= 1 as a cut and defined the ABCD regions in the Δϕ /BDT space

A	BDT < 0.0 Δϕ < X	15046.77 ±94.29
D	BDT < 0.0 Δϕ > X	304.58 ± 20.75
B	BDT > 0.0 Δϕ < X	752.61 ± 15.62
C	BDT > 0.0 Δϕ > X	36.33 ± 2.87

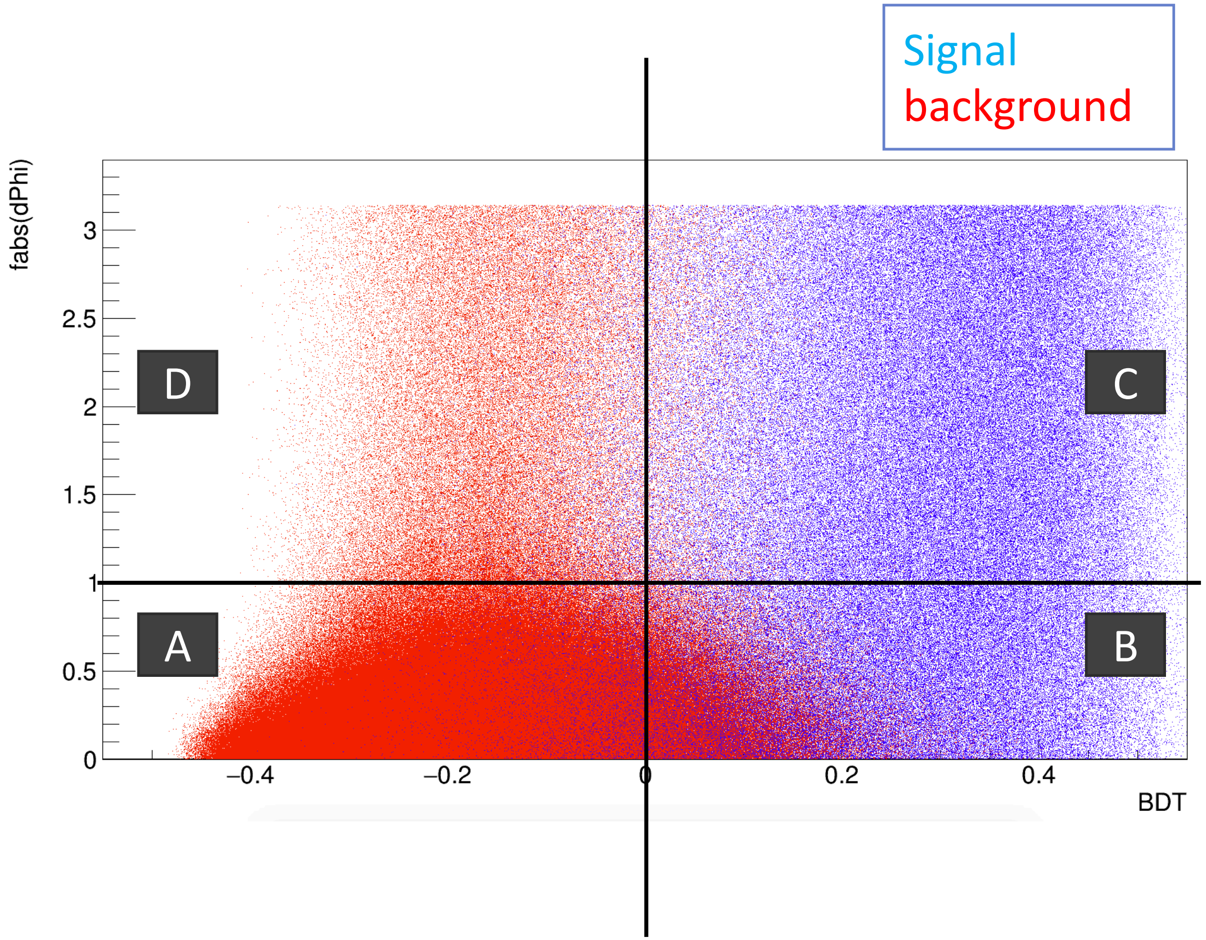
Where X is sliding Δϕ cut based on LT

Rcs_{BC} = B/C = 752.61/36.33 = 20.7

Rcs_{AD} = A/D = 15046.77/304.58 = 49.4

κ = Rcs_{AD} / Rcs_{BC} = 2.3 !!!

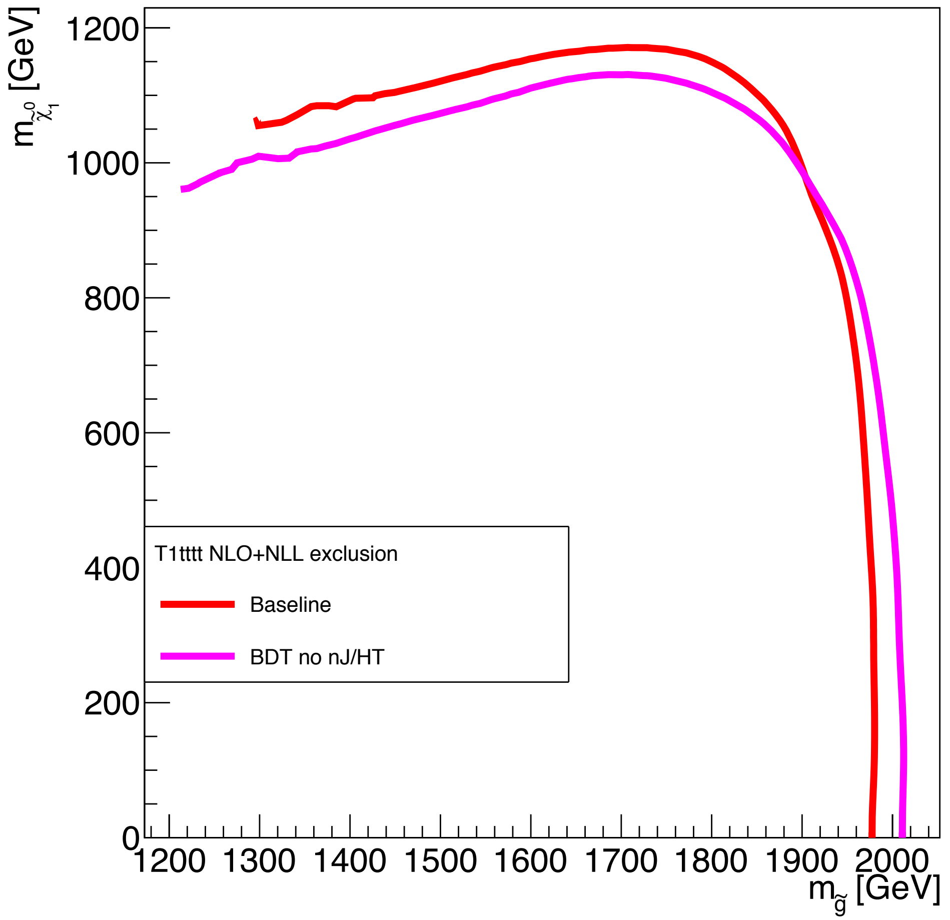
Cutting away BDT < -0.2 | -0.25 |-0.3 | -0.35 did not help much kappa could be of order 1.8 or more



A. Mohamed, SUSY 1Lep ,DESY, RWTH

BDT LIMITS RESULTS 2016

- Plot has 2016 Limits when removing nJet/HT from the training and use $\Delta\phi$ instead
- Red is baseline using the new scan
- Pink line is the limit when used the strategy of combined mass points removing nJ/HT from the training



BACKGROUND TEST

- When nJ/HT are removed from the training and Used nTop >= 1 as a cut and defined the ABCD regions in the nJ /BDT space

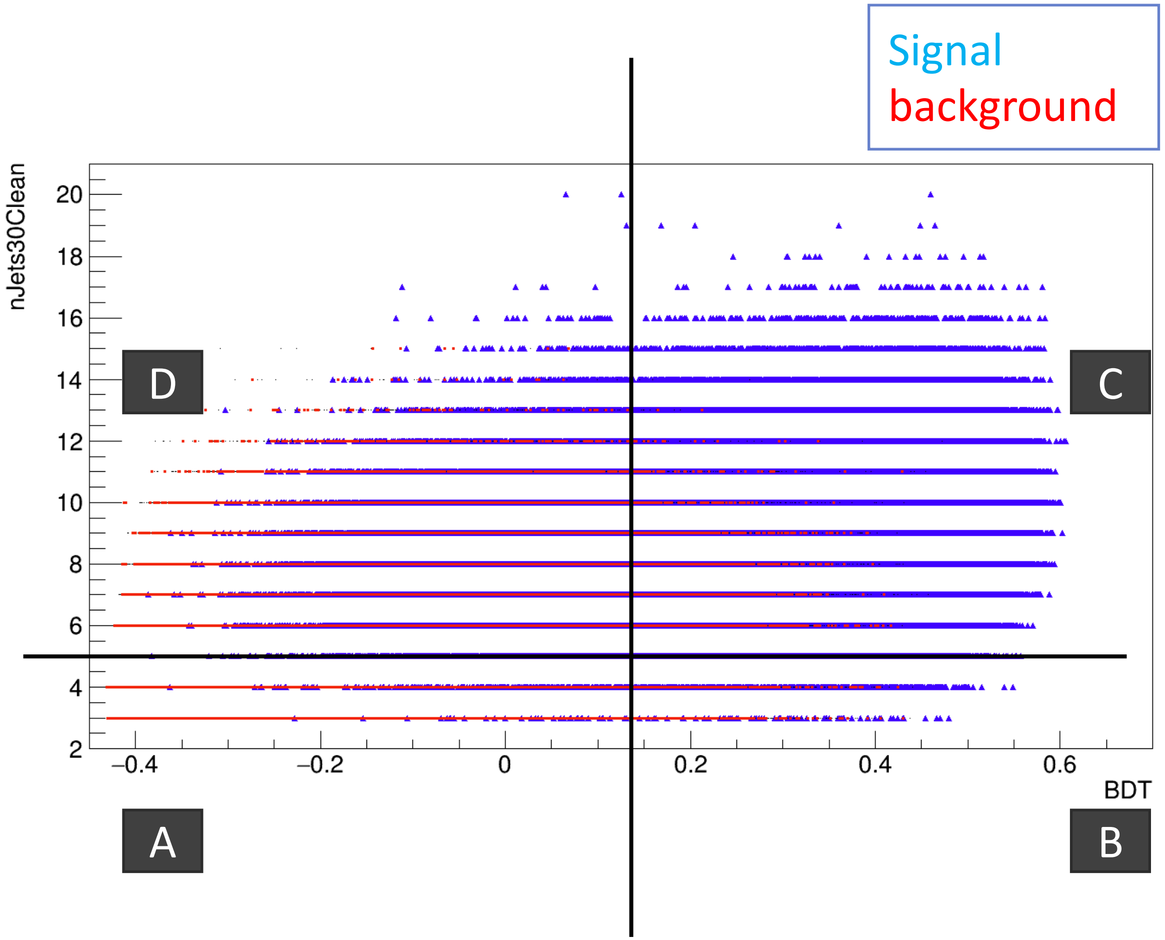
A BDT < 0.0 nJ in [4,5]	70.65 ± 6.1
D BDT < 0.0 nJ >= 6	48.77 ± 4.17
B BDT > 0.0 nJ in [4,5]	152.0 ± 17.09
C BDT > 0.0 nJ >=6.	150.58 ± 11.4

Rcs_{BC} = B/C = 152.0 /150.58 = 1.00

Rcs_{AD} = A/D = 70.65 /48.77 = 1.44

κ = Rcs_{AD} / Rcs_{BC} = 1.4 !!!

Cutting away BDT < -0.2 | -0.25 |-0.3 | -0.35 did not help much **κ** is constant at 1.4 all the time



A. Mohamed, SUSY 1Lep ,DESY, RWTH

DNN Binary classification

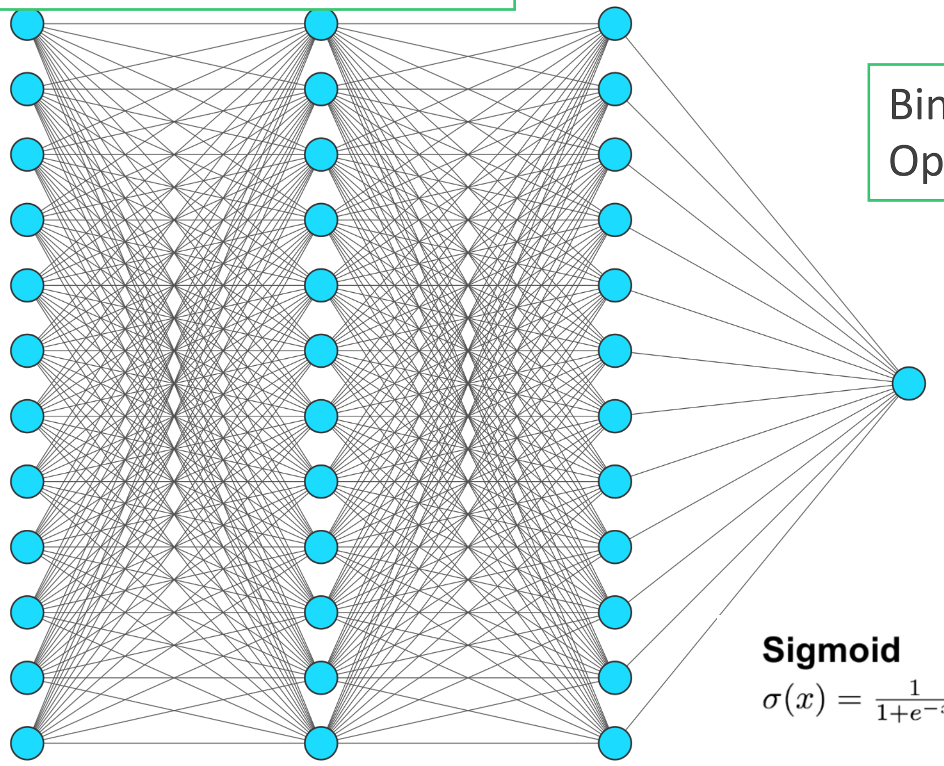
DNN DETAILS

Signal and background are weighted to the same level (exactly as TMVA) only for training, physics weight Is used for evaluations
Independent sample is used for training and testing

Input Features

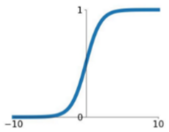
$$MET, MT, Jet_{p_T}^{1/2}$$
$$Lep_{p_T}, L_T, H_T, n_{bJets}$$
$$n_{Top}, n_{jets}, \Delta\phi, Lep_{relIso}$$
$$Lep_{MinilIso}, IsoTrack_{p_T}$$
$$IsoTrack_{MT_2}$$
$$m_{\tilde{g}}, m_{\chi_1^0}$$

Preselect events :
- I Lep , no Veto, LT > 250 , HT > 500, nJ > 3 nbJ >= 1, iso_track veto

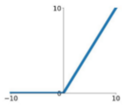


Binary crossentropy loss with Adam Optimizer (learning rate 0.001)

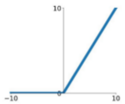
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$



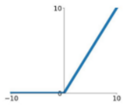
ReLU
$\max(0, x)$



ReLU
$\max(0, x)$



ReLU
$\max(0, x)$



DNN DETAILS

Early stopping to stop training once the model performance stops improving on a hold out validation dataset with patience of 10 epochs.

100 epochs, batch size = 512
Dropout with rate of 0.01 [Dropout]

Input Features

$MET, MT, Jet_{p_T}^{1/2}$

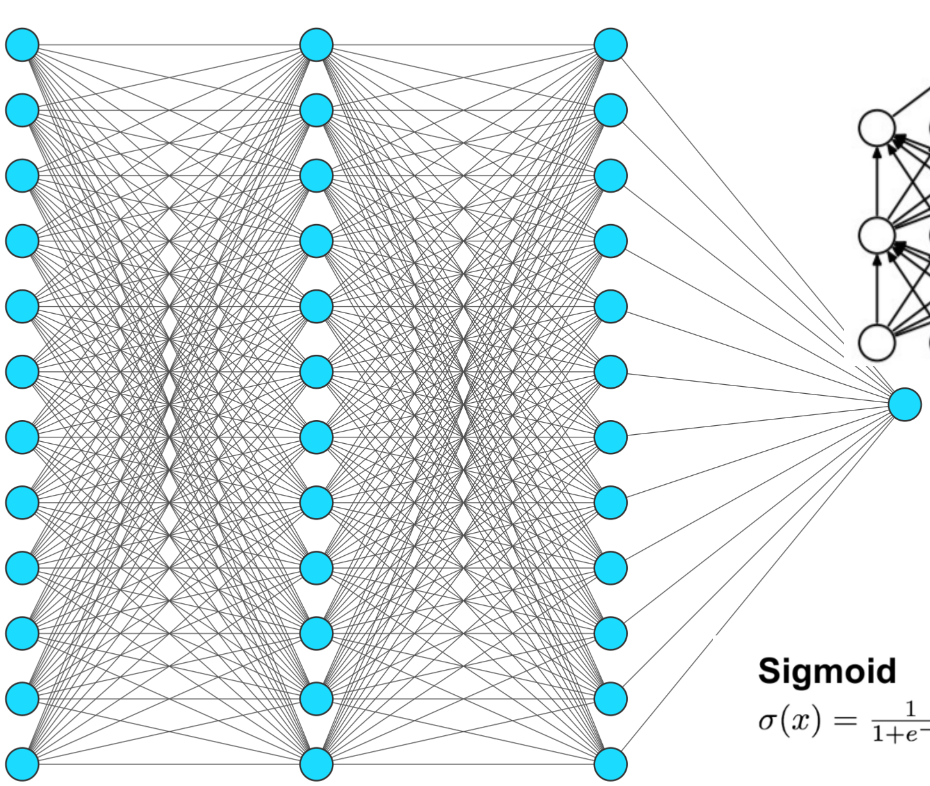
$Lep_{p_T}, L_T, H_T, n_{bJets}$

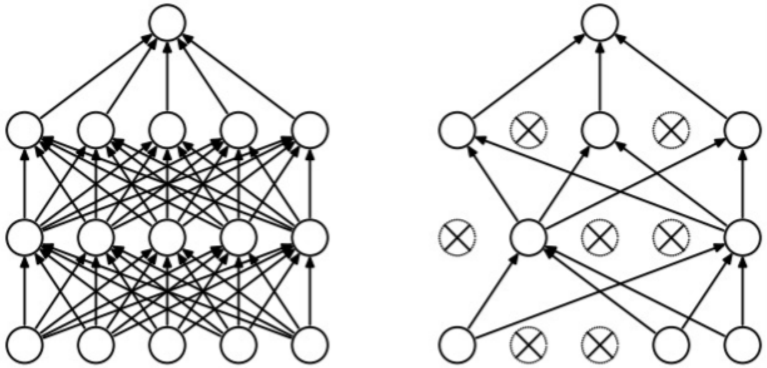
$n_{Top}, n_{jets}, \Delta\phi, Lep_{relIso}$

$Lep_{MinilIso}, IsoTrack_{p_T}$

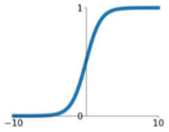
$IsoTrack_{MT_2}$

$m_{\tilde{g}}, m_{\chi_1^0}$

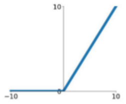




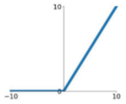
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$



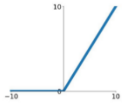
ReLU
$\max(0, x)$



ReLU
$\max(0, x)$



ReLU
$\max(0, x)$



PARAMETRIZED DNN

ML_param_BDT

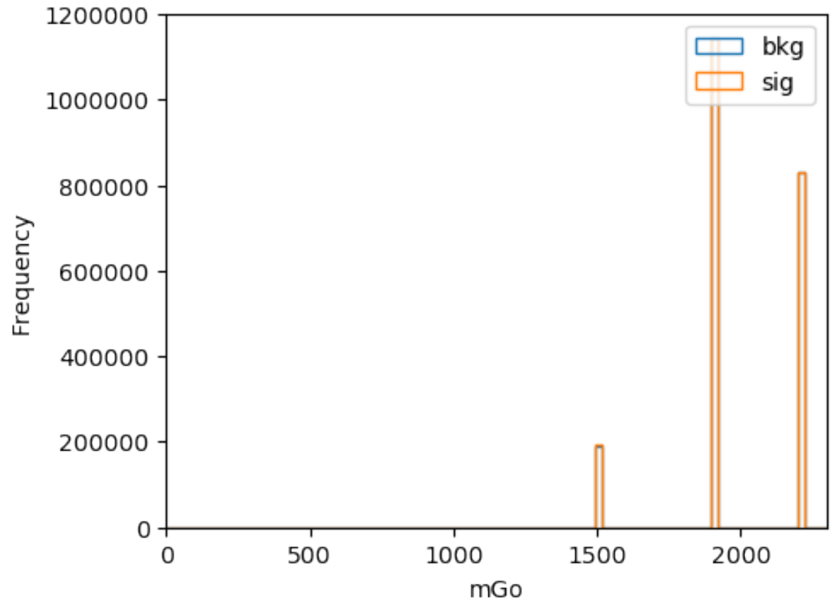
- The same 6 mass points used for this training (1.5/1.0,1.9/0.1,1.9/08,1.9/1.0,2.2/0.1,2.2/0.8)
- Need to define signal mass for both backgrounds and data as well.
- For the backgrounds two options were considered:

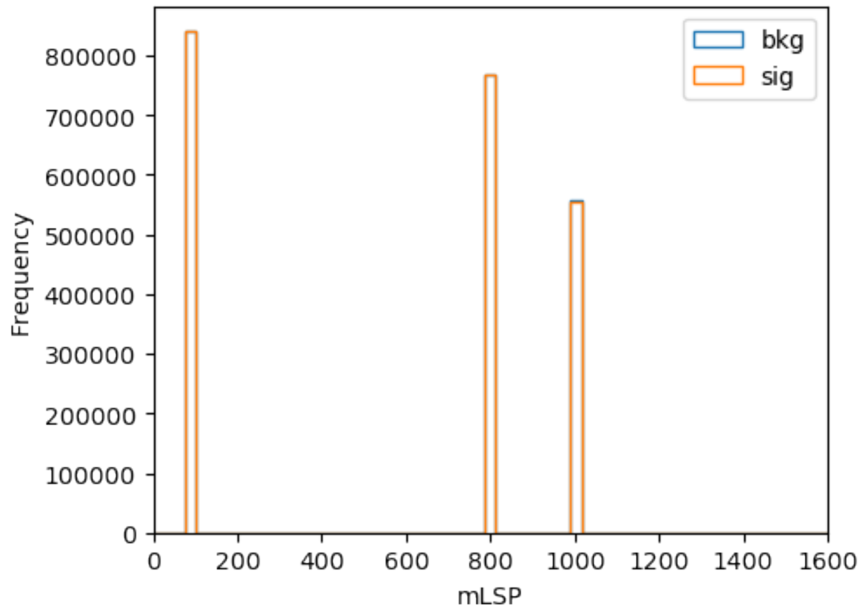
1. Randomization: Assign mGo & mLSP randomly (out of list of signal masses) during DNN training to every background event. **(has a problem that each signal mass will be trained with part of background phase space)**

2. Over-sampling: Create multiple copies of a given MC background event. Each copy having same values of all reconstructed variables but a different mGo & mLSP value

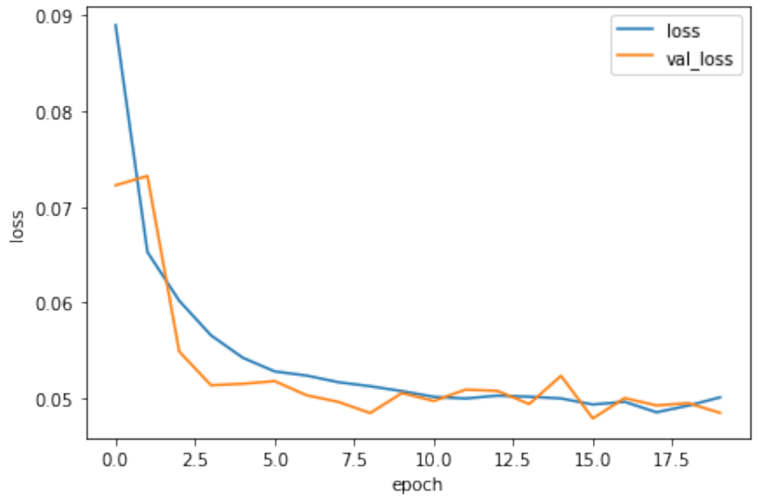
Full background
mGo_1,mLSP_1

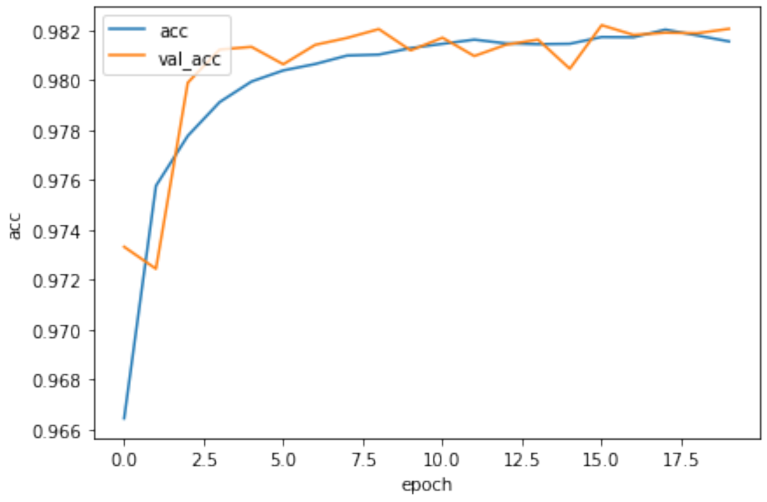
Full background
mGo_n,mLSP_n

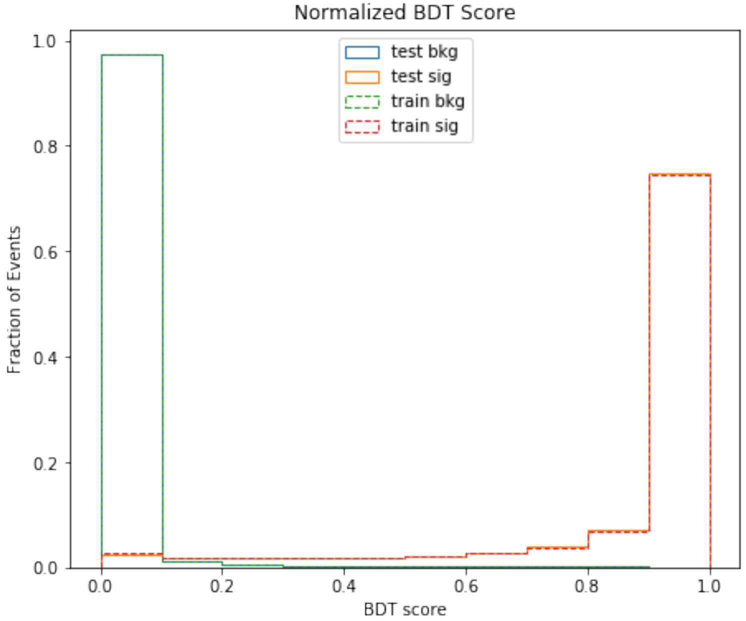


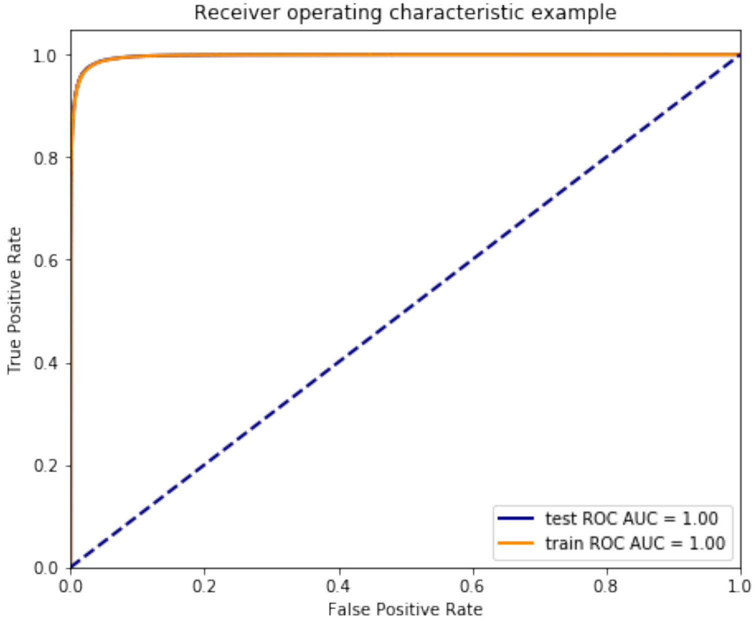


DNN PERFORMANCE

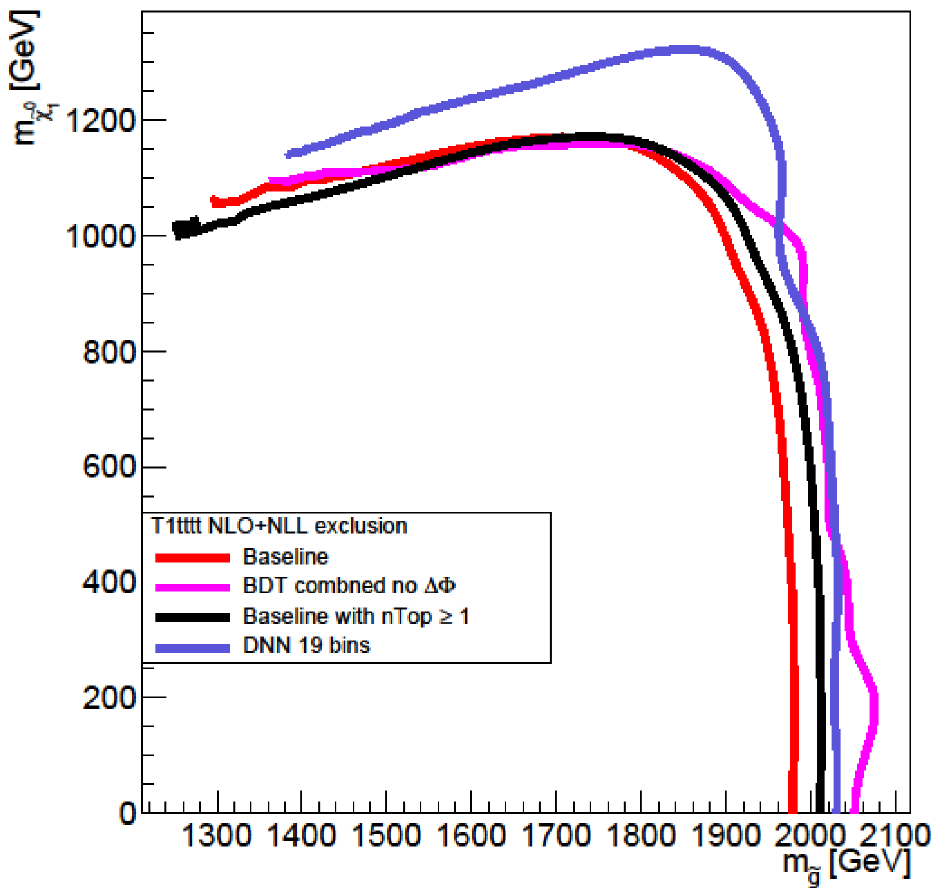


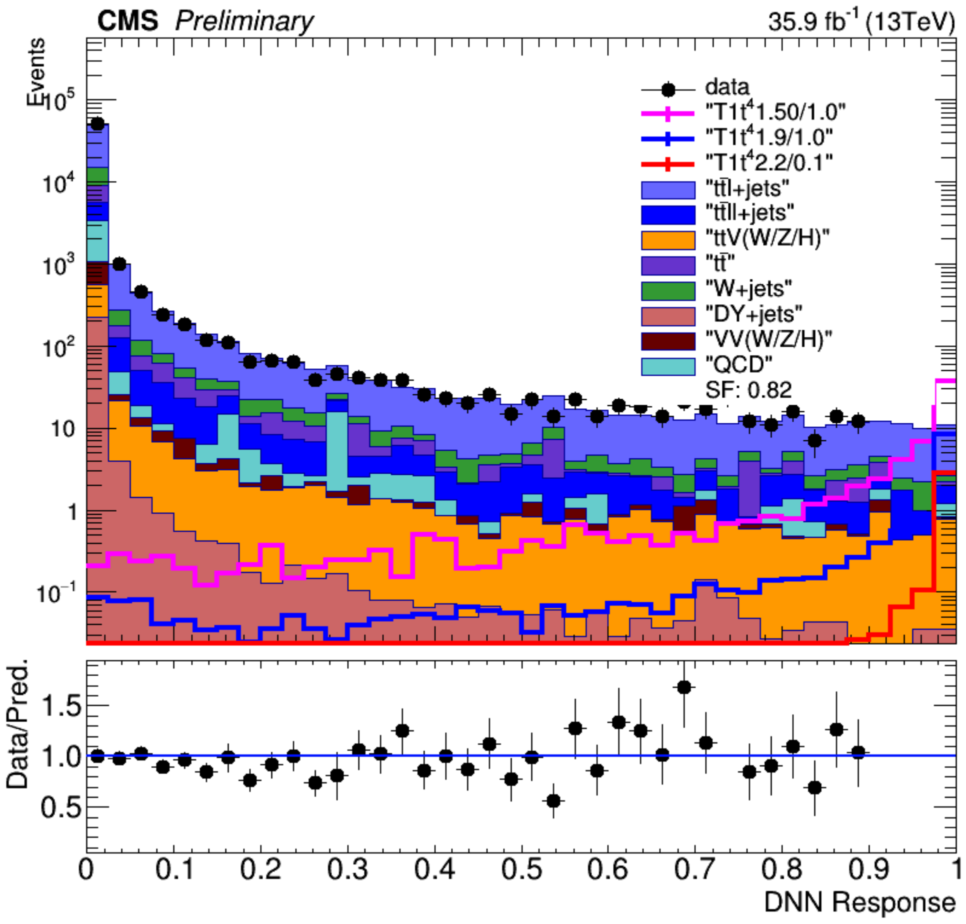






DNN PERFORMANCE



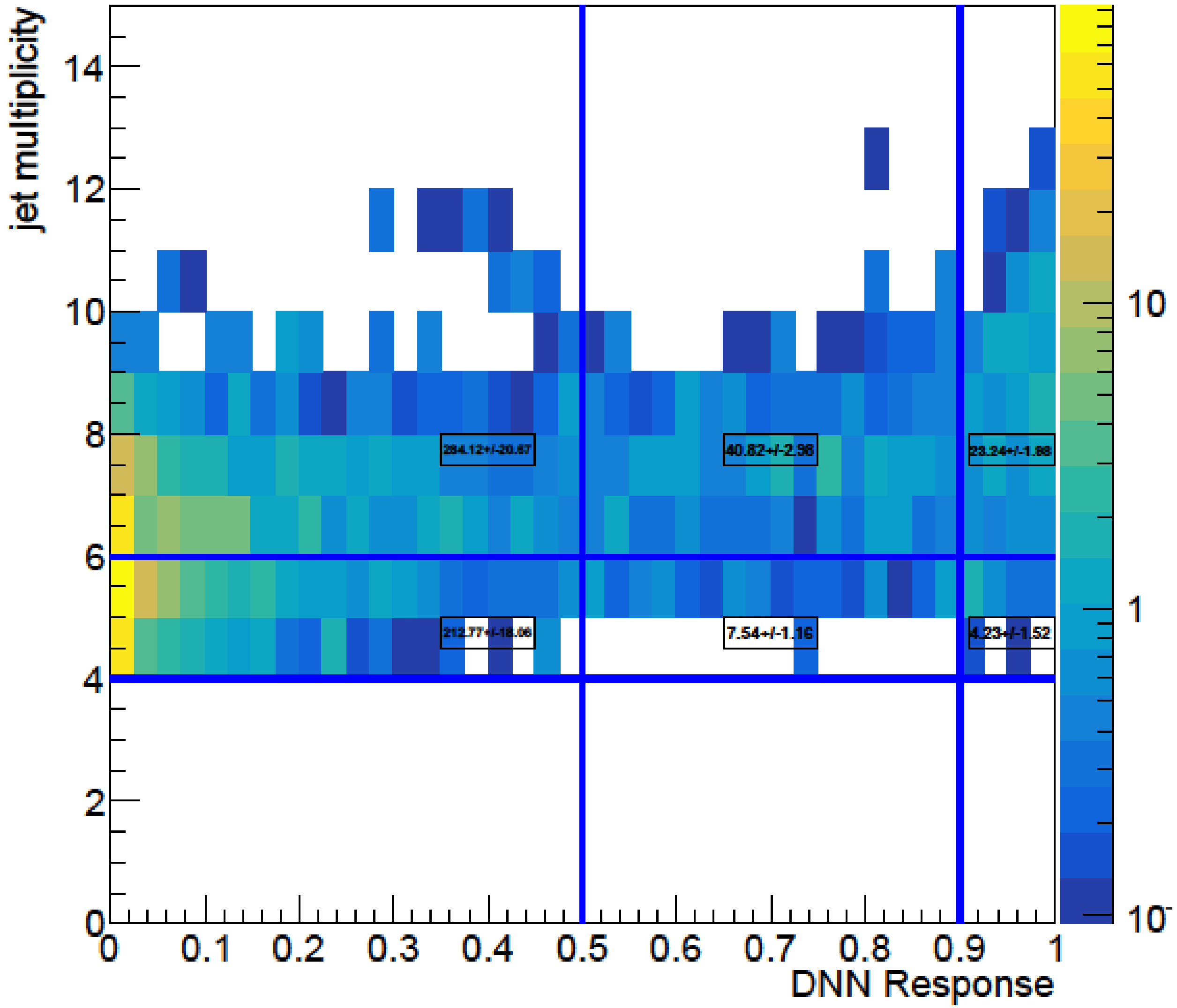


A. Mohamed, SUSY 1Lep ,DESY, RWTH

07.08.19

DNN FOR BACKGROUND

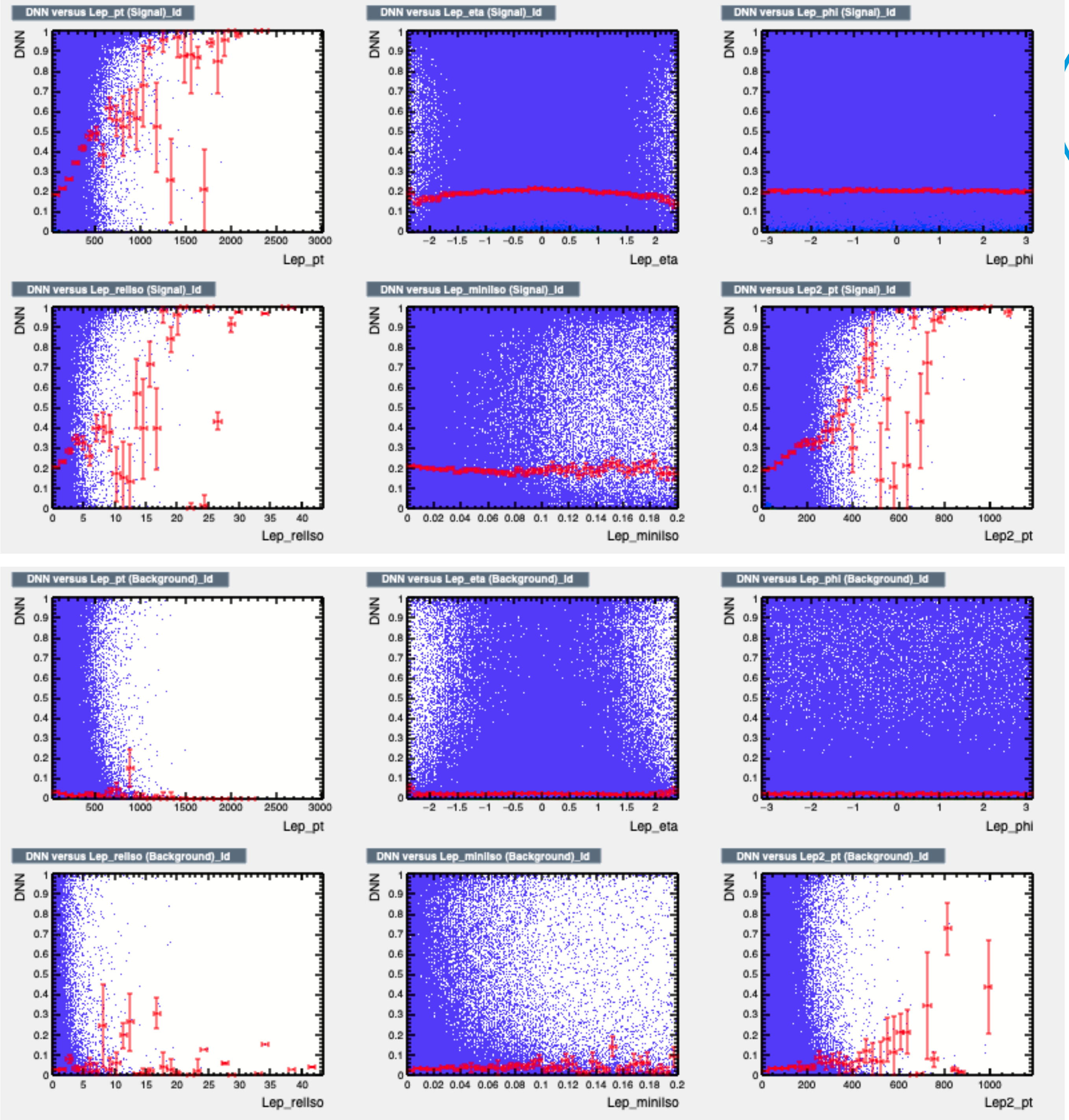
- Cut away the DNN < 0.5 and take the rest define ABCD as in the plot without DNN < 0.5
- Add nTop-dPhi cuts
- Rcs_$_{BC}$ = B/C = 4.23/23.24 = 0.18
- Rcs_$_{AD}$ = A/D = 7.54/40.82 = 0.18
- **κ** = Rcs_$_{AD}$ / Rcs_$_{BC}$ = 1.0 !!!
- But the problem is backgound in A & B is << C & D



DNN FOR BACKGROUND

- Studied the correlations between DNN and other 25 variables
- Non of them seems to work
- Only Lep_phi seems to be flat (as expected)

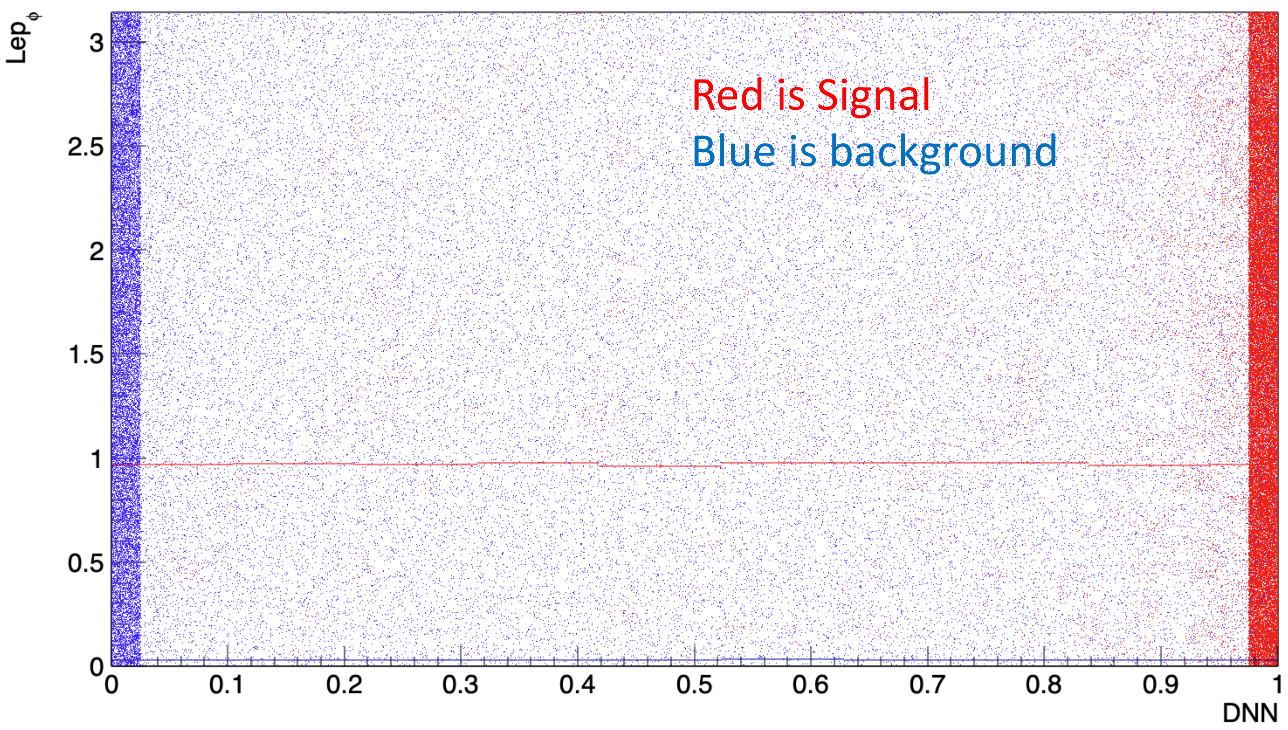
More plots in the backup



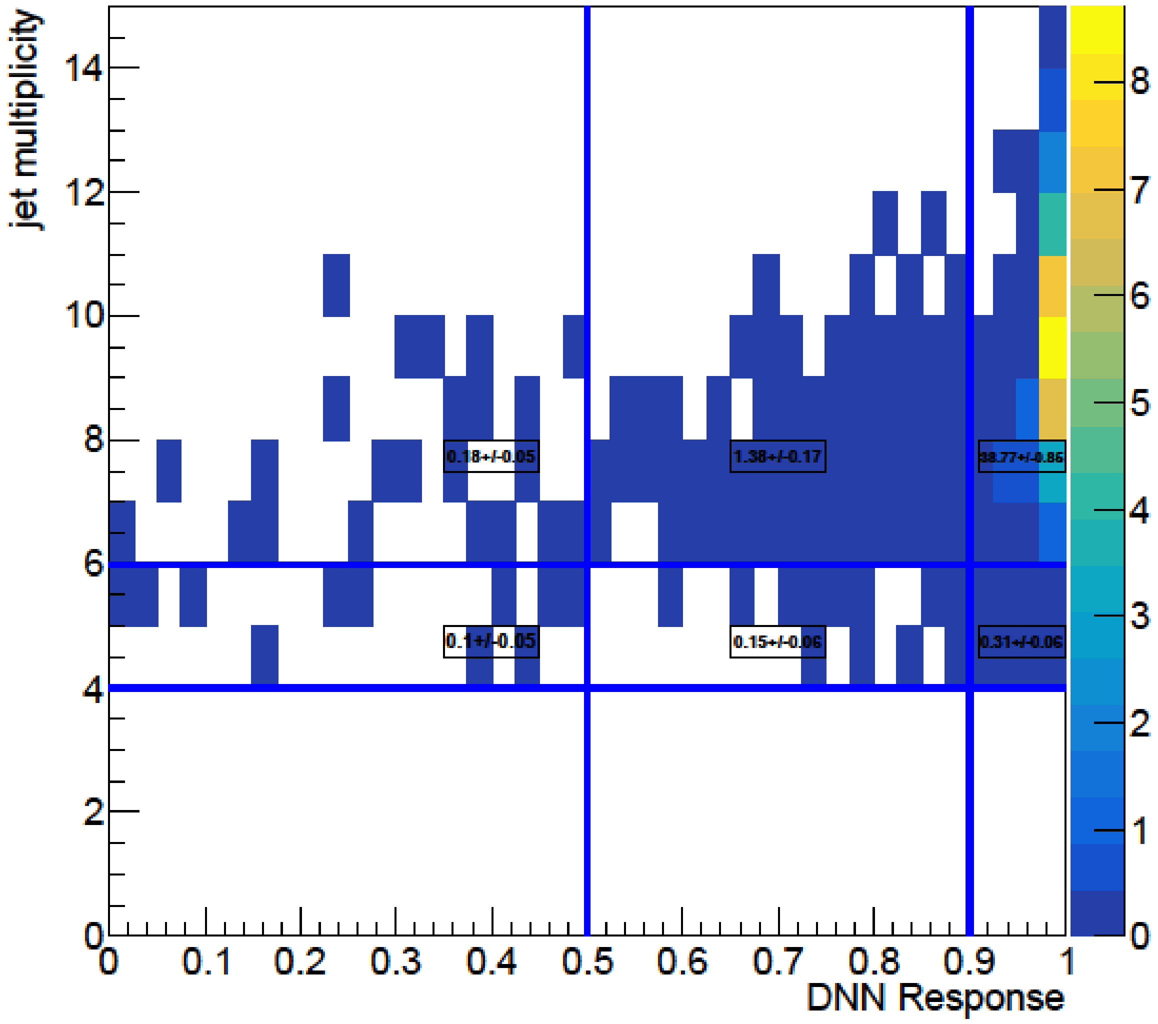
A. Mohamed, SUSY 1Lep ,DESY, RWTH

DNN FOR BACKGROUND

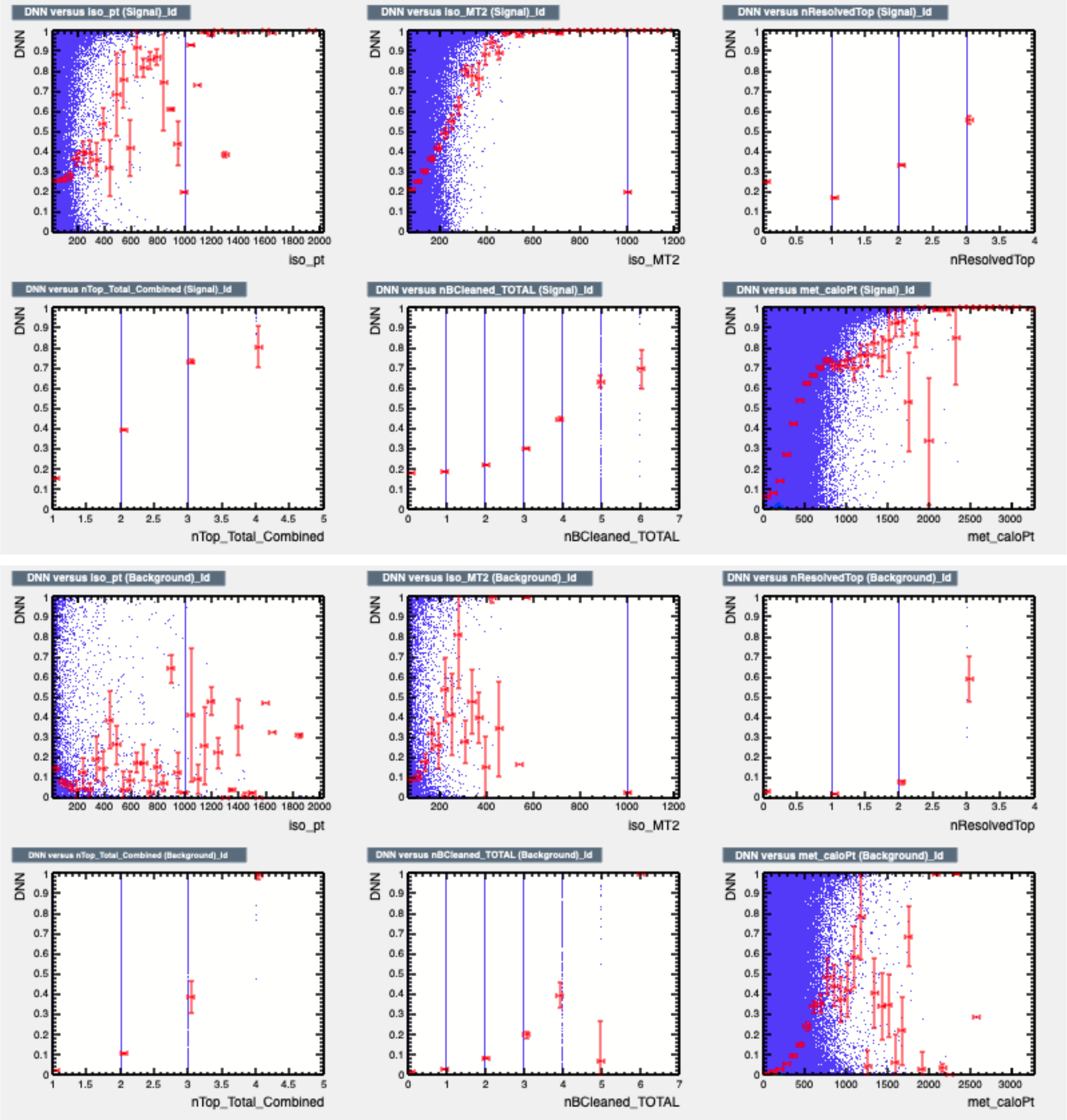
- Although Lep_phi is not correlated, it's not working as it look very the same in signal and background w.r.t DNN



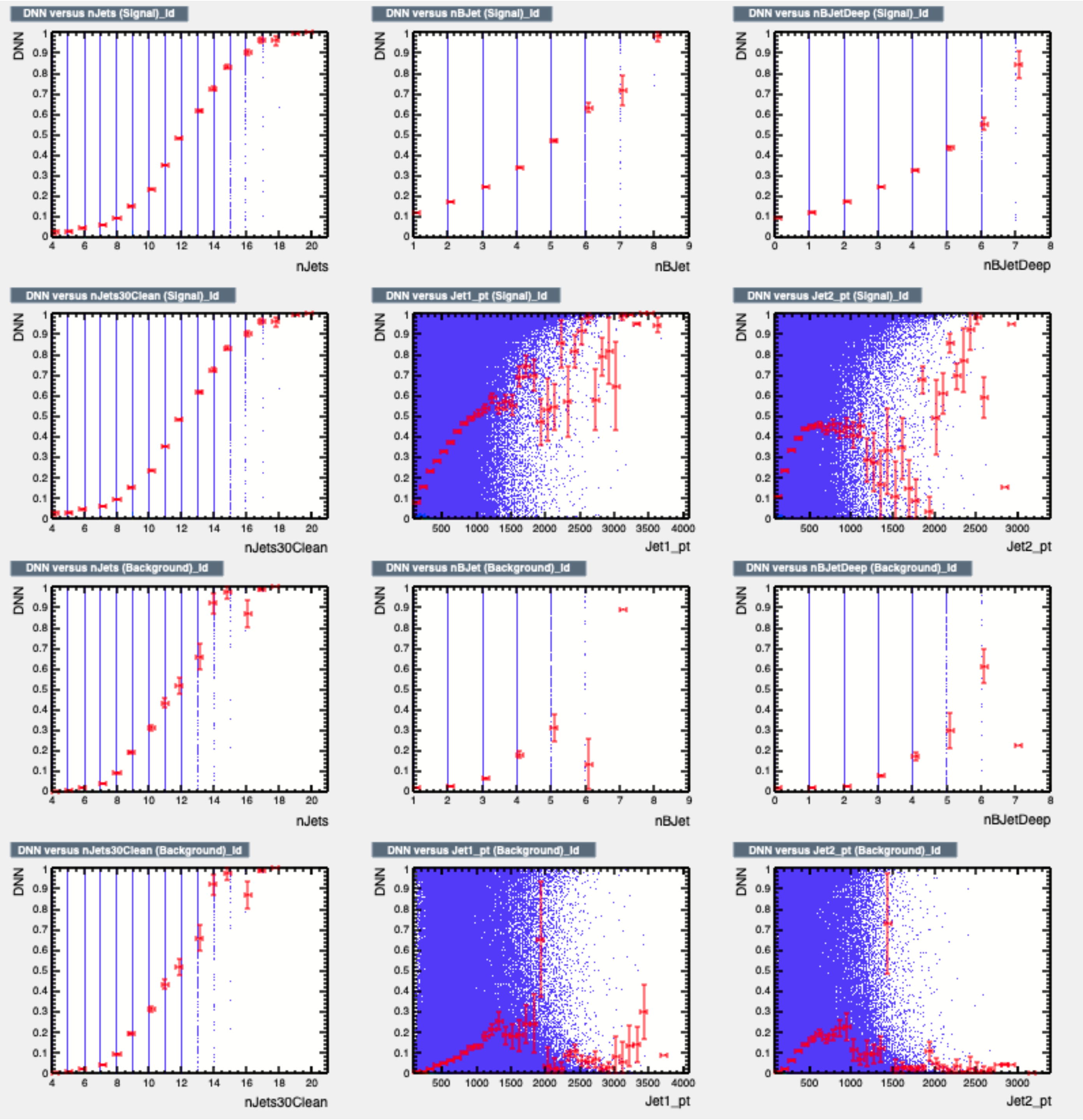
SIGNAL DNN VS NJ



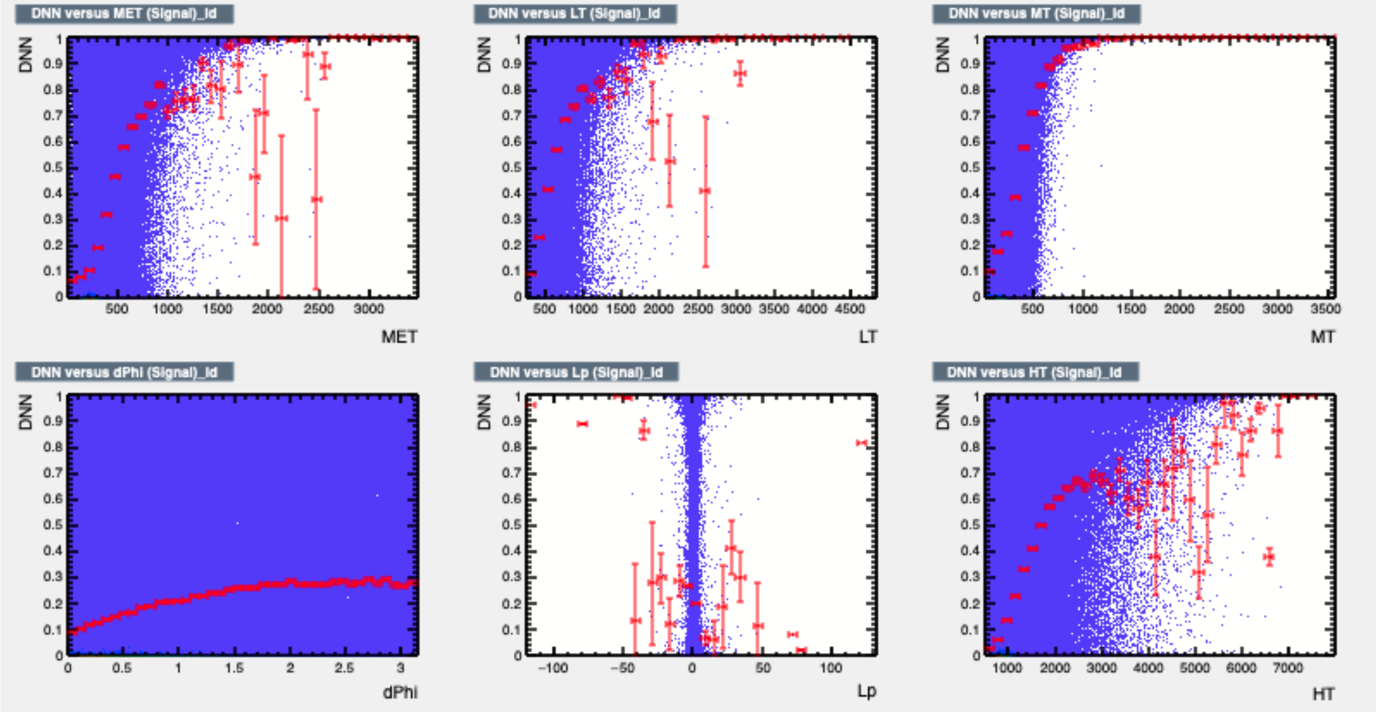
DNN FOR BACKGROUND

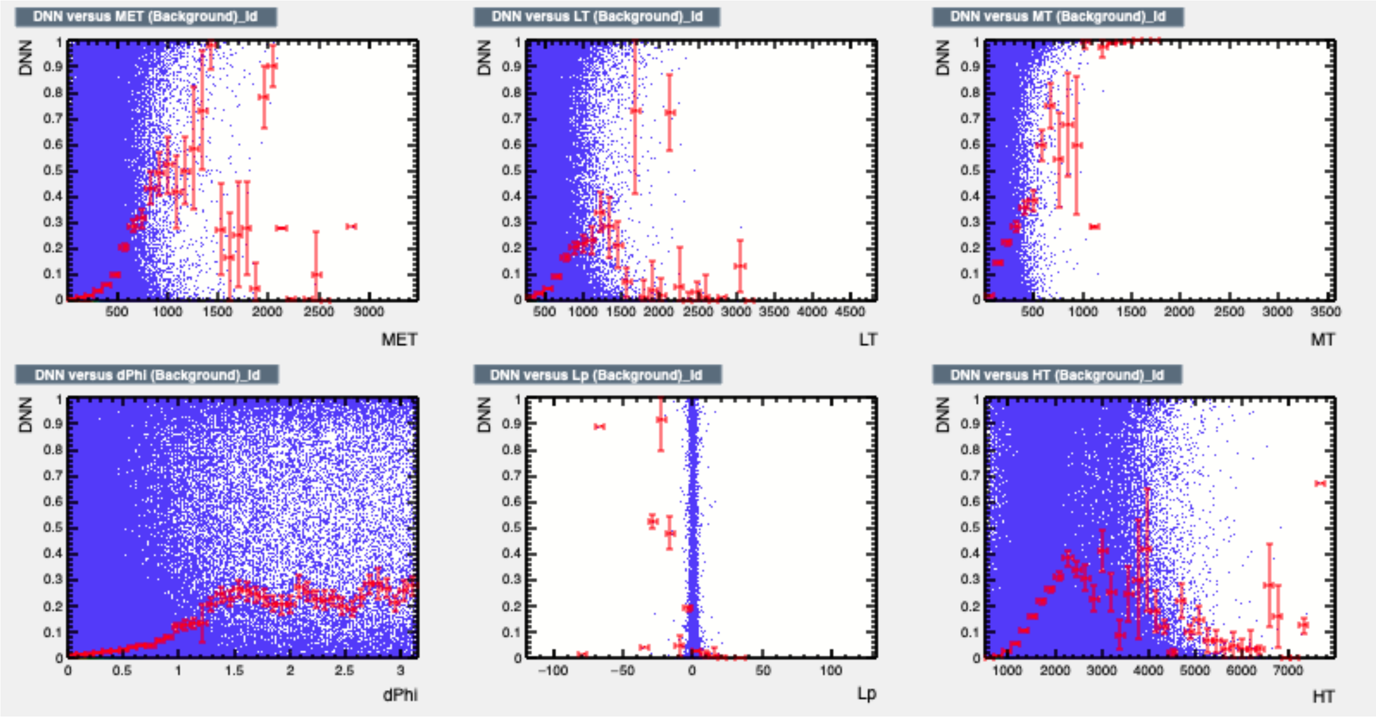


DNN FOR BACKGROUND



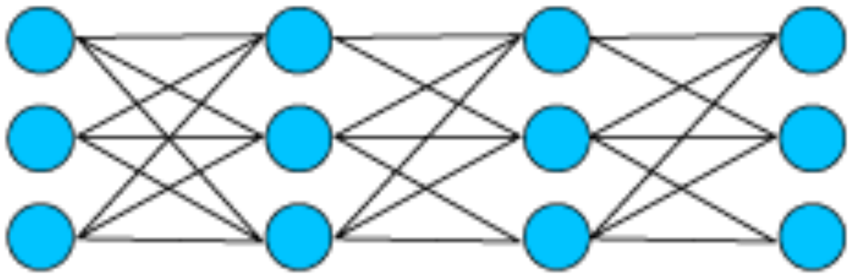
DNN FOR BACKGROUND





DNN DETAILS

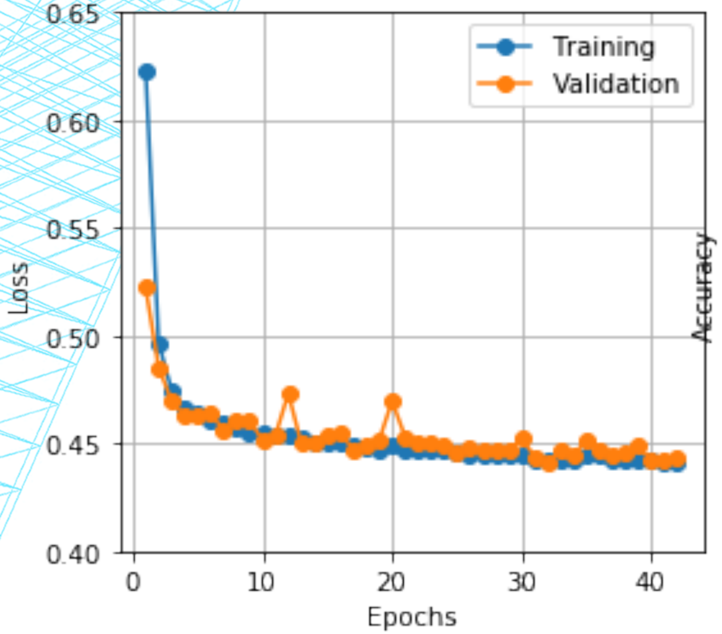
- Fully connected 2 hidden layer multi-class network
- var_list = ['MET', 'MT', 'Jet2_pt', 'Lep_pt', 'LT', 'HT','nBCleaned_TOTAL','nTop_Total_Combined', 'nJets30Clean', 'dPhi’]
- 'mGo', 'mLSP’ are passed to the network as input parameters (random number generated for the background from signal shape)
- loss='sparse_categorical_crossentropy’
- optimizer=Adam
- All classes are weighted to the same level (exactly as TMVA) only for training and physics weight Is used for evaluations
- 200 epochs, batch_size = 1024
- Early stopping is used
- Independent sample is used for training and testing only

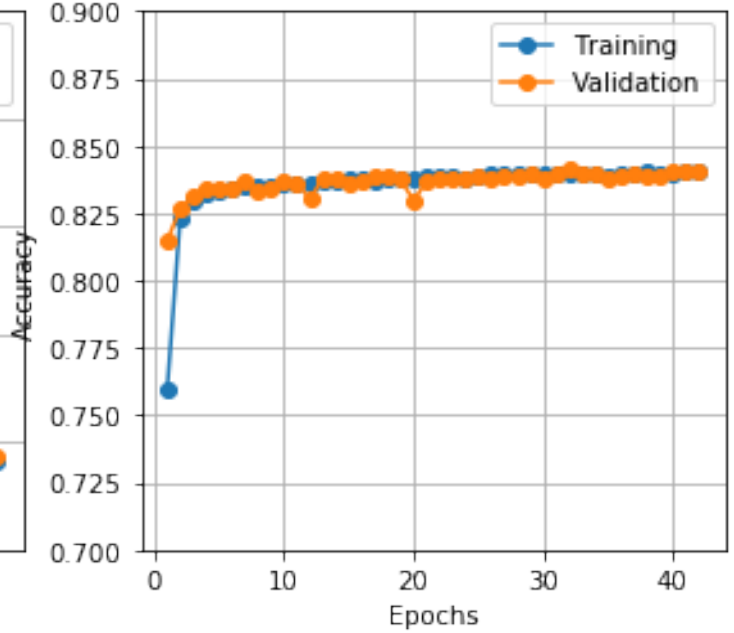


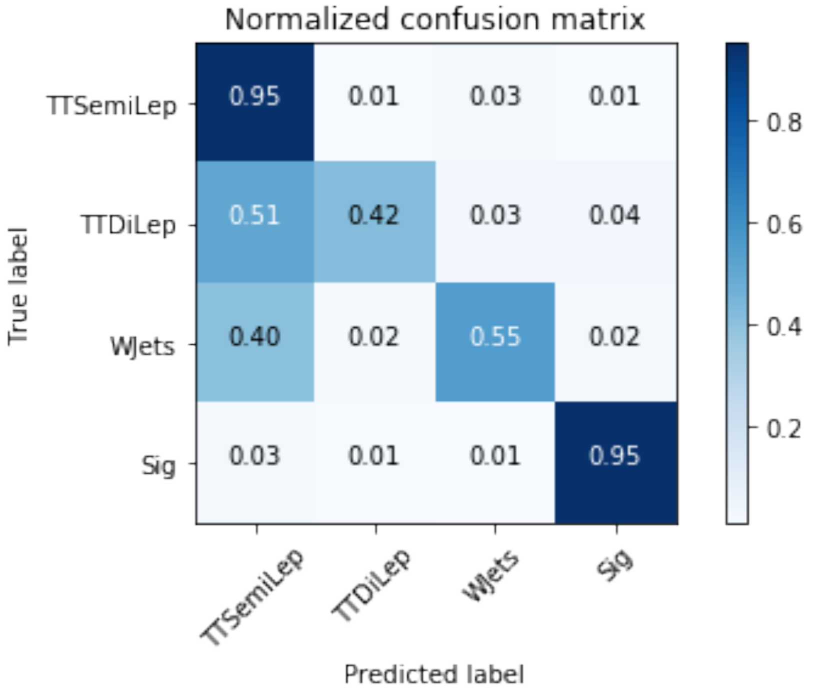
relu | Relu 256 | Relu 256 | sigmoid

Class 0 → DiLep Ttbar
Class 1 → SemiLep Ttbar
Class 2 → Wjets + others
Class 4 → Signal

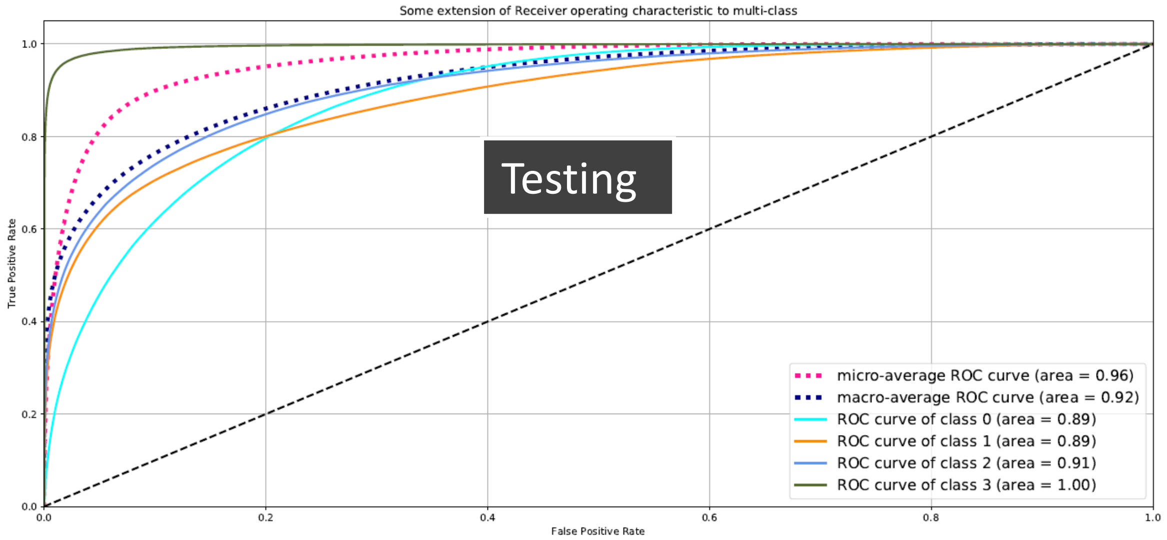
DNN PERFORMANCE

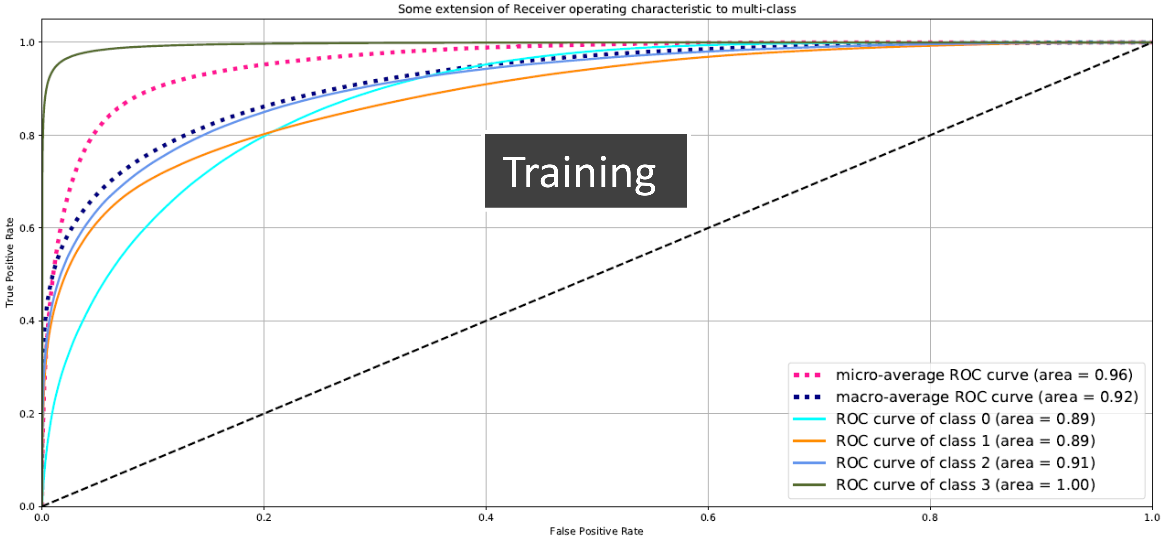






DNN PERFORMANCE





A macro-average will compute the metric independently for each class and then take the average (hence treating all classes equally), whereas a micro-average will aggregate the contributions of all classes to compute the average metric.

https://datascience.stackexchange.com/questions/15989/micro-average-vs-macro-average-performance-in-a-multiclass-classification-settin

DNN PERFORMANCE

For all of the classes the map looks the similar
Interesting to mention that mGo & mLSP have no power at all