

ML for Polarized VBS

*

Starting data and cuts

In this analysis we focused on SSWW VBS and we studied the polarization of the W vector bosons; in particular, we applied the following cuts to the observed features

- $m_{jj} > 500 \text{ GeV}$
- $m_{ll} > 20 \text{ GeV}$
- $p_T^{\text{MET}} > 30 \text{ GeV}$
- $d\eta_{jj} > 2.5$
- $p_T^{j1} > 50 \text{ GeV}$
- $p_T^{j2} > 50 \text{ GeV}$
- $p_T^{l1} > 25 \text{ GeV}$
- $p_T^{l2} > 20 \text{ GeV}$

The data we are working on are labeled with the polarization states of the W bosons in the LAB reference frame, with either an L for longitudinal polarization and T for transverse

- $W_L^\pm W_L^\pm$
- $W_T^\pm W_L^\pm$
- $W_T^\pm W_T^\pm$

Our goal is to build a ML model that can separate LL events from TX (i.e. TL or TT)

Features

Of the various physical quantities available we focus on the subset that we consider relevant in characterizing the events (still not a small set – we train the model and then, in the case of useless variables, we can drop them):

- Leptons data:
 - transverse momenta $p_{l1} p_{l2}$
 - pseudorapidities $\eta_{l1} \eta_{l2}$
 - azimuthal angles $\phi_{l1} \phi_{l2}$
- VBS Jets data (Jet data comes from anti-kt algorithm with radius=0.4; we call VBS Jets the first two jets in order of transverse momentum):
 - $p_{j1} p_{j2}, \eta_{j1} \eta_{j2}, \phi_{j1} \phi_{j2}$
 - invariant masses $M_{j1} M_{j2}$
- non-VBS Jets data (the following two jets, if any):
 - $p^+_{j1} p^+_{j2}, \eta^+_{j1} \eta^+_{j2}, \phi^+_{j1} \phi^+_{j2}, M^+_{j1} M^+_{j2}$
- Other data:
 - $m_{jj} d\eta_{jj}$ of the VBS jets
 - missing transverse energy $p_T^{MET} \phi^{MET}$
- Neutrino data (from GEN level):
 - $p_{j1} p_{j2}, \eta_{j1} \eta_{j2}, \phi_{j1} \phi_{j2}$

For a total of $6+8+8+4+6=32$ features

New features

We decided to include other variables that could help in extracting even more information from the event data:

- $\cos\theta^*$ – the lepton decay angle in the W frame (the axis are rotated before applying the boost), one per lepton (Pelliccioli [1710.09339](#))
- $\cos\theta_{\text{CS}}$ – Collins-Soper frame angle ([1605.05450](#))
- $\cos\theta_{\text{2D}}$ – transverse helicity angle, one per lepton ([1203.2165](#))
- R_{pT} – ratio of leptons and highest- p_T -jets transverse momenta ([1201.2768](#))

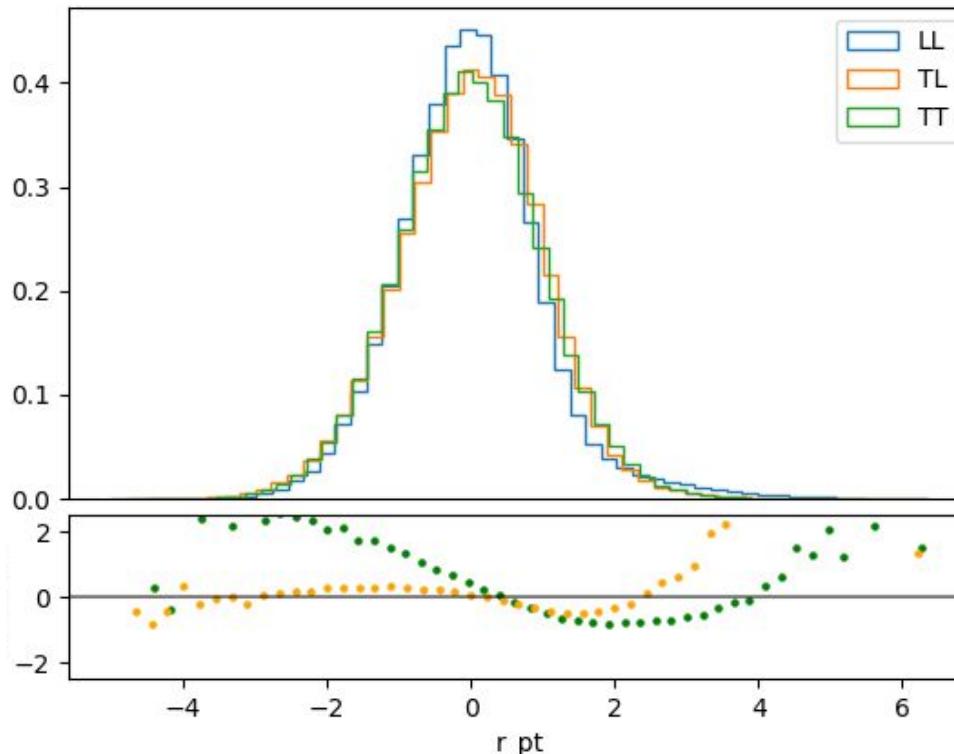
So $2+1+2+1=6$ new features.

Each feature is scaled if (and as) needed and then normalized, so that the ML models can fruitfully understand the distributions

Some examples in the following

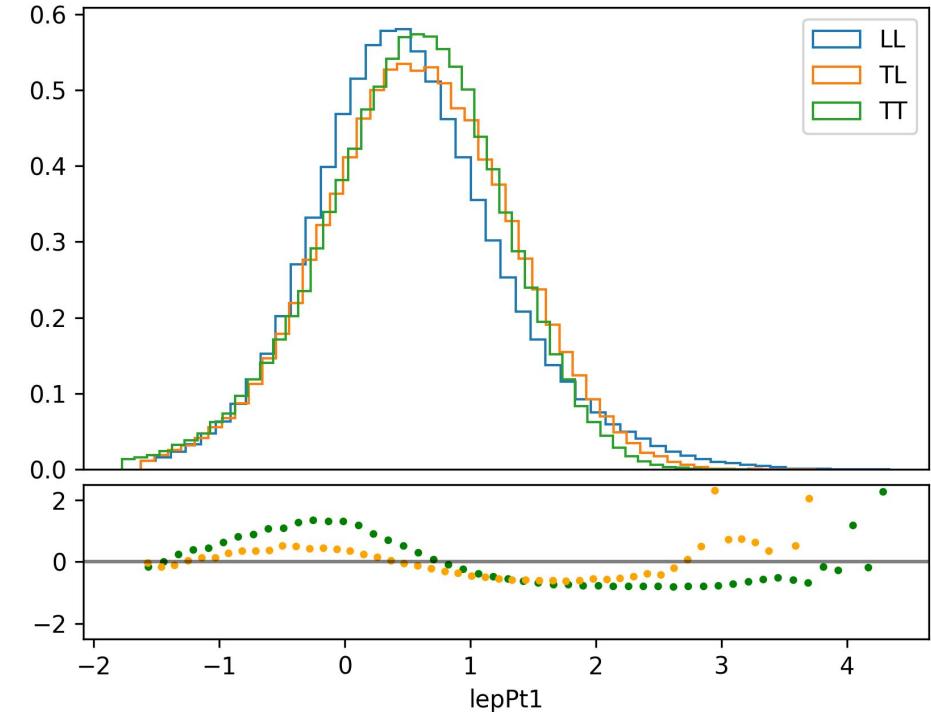
Most important features visualization

Scaled and normalized r_{pt} histogram

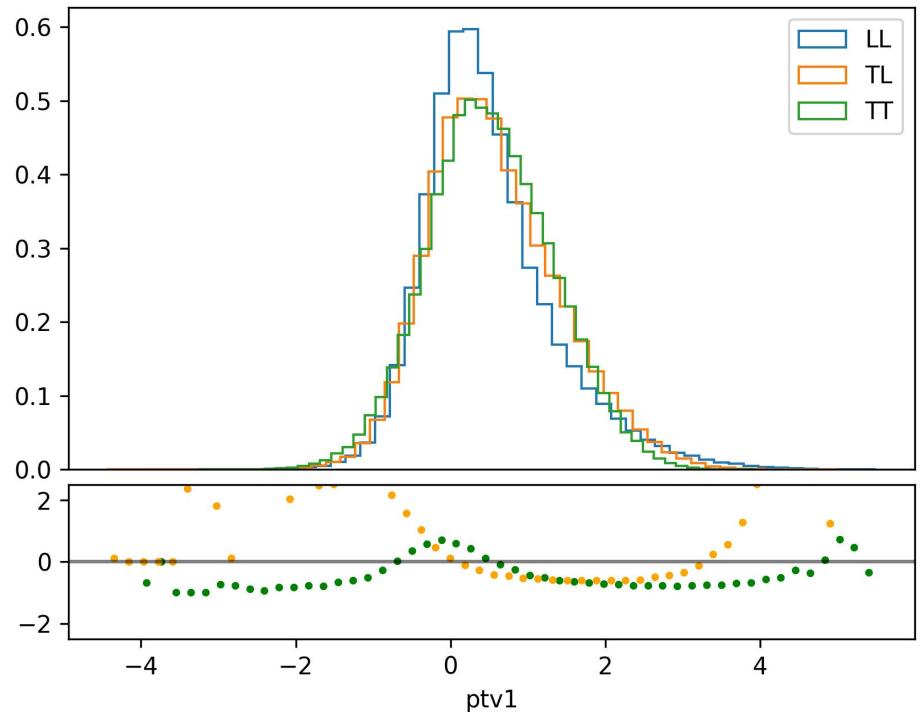


Most important features visualization

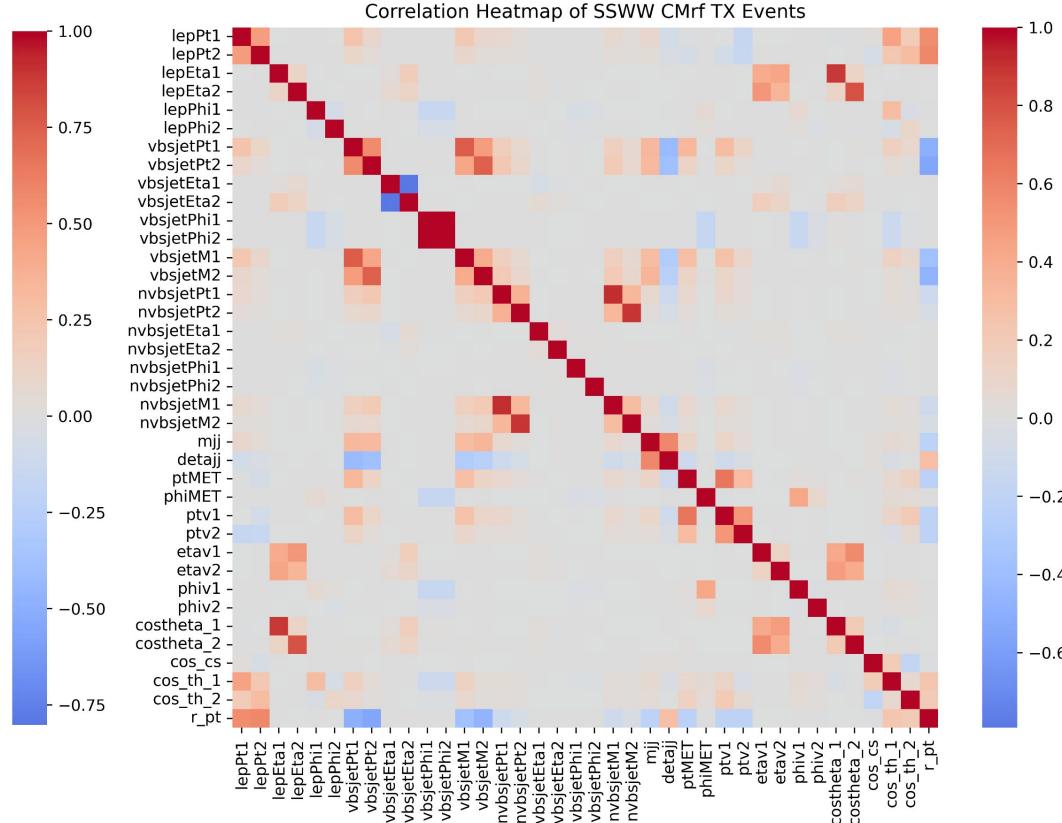
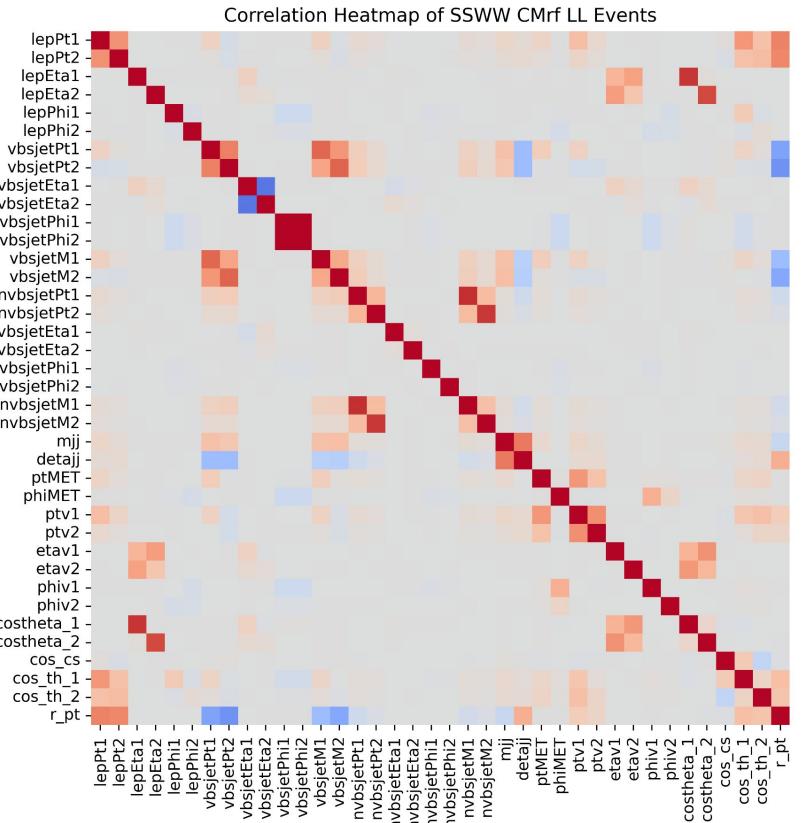
Scaled and normalized lepPt1 histogram



Scaled and normalized ptv1 histogram

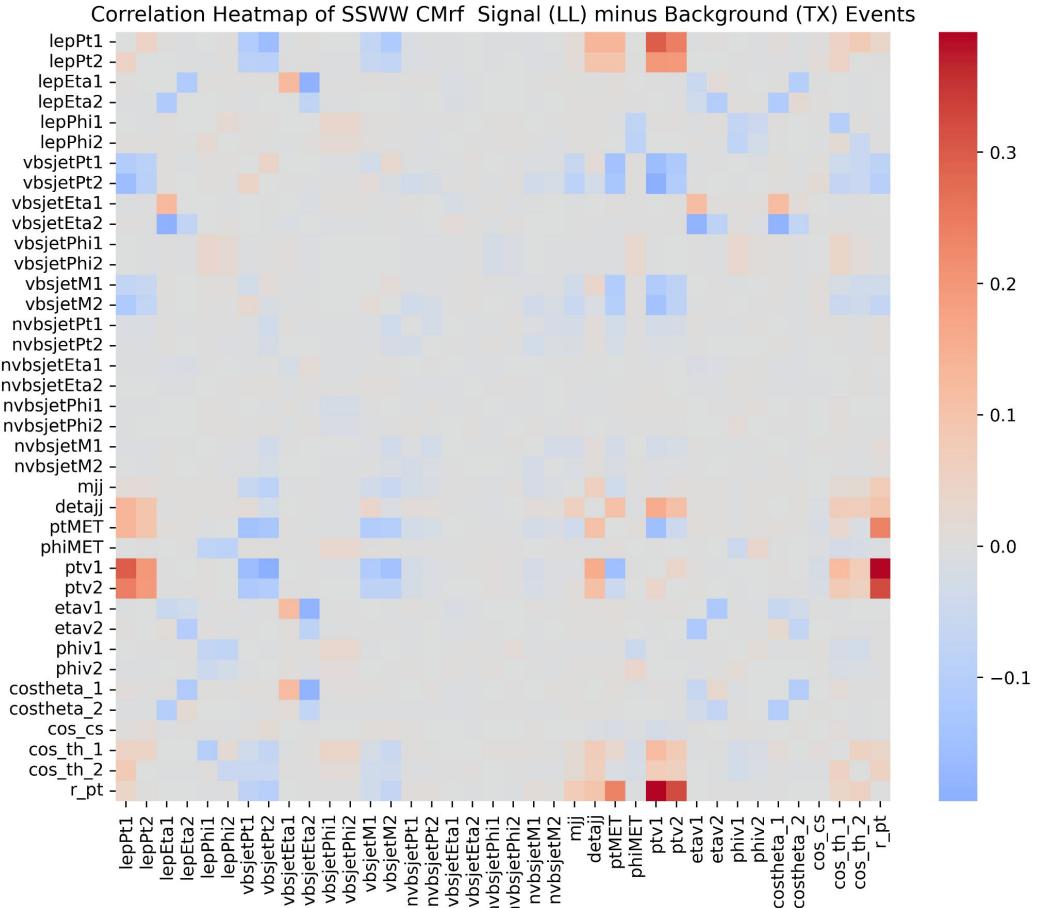


Correlation (1/3)



Feature correlation is apparently similar in signal events and background ones

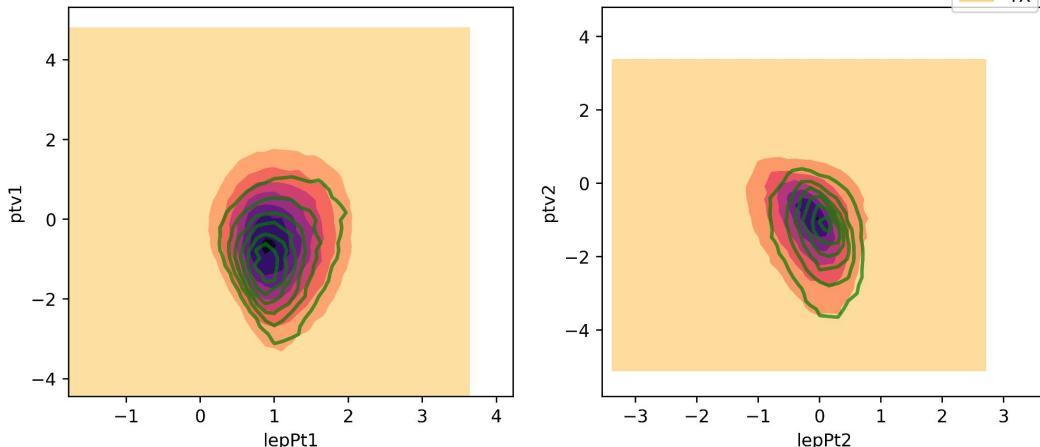
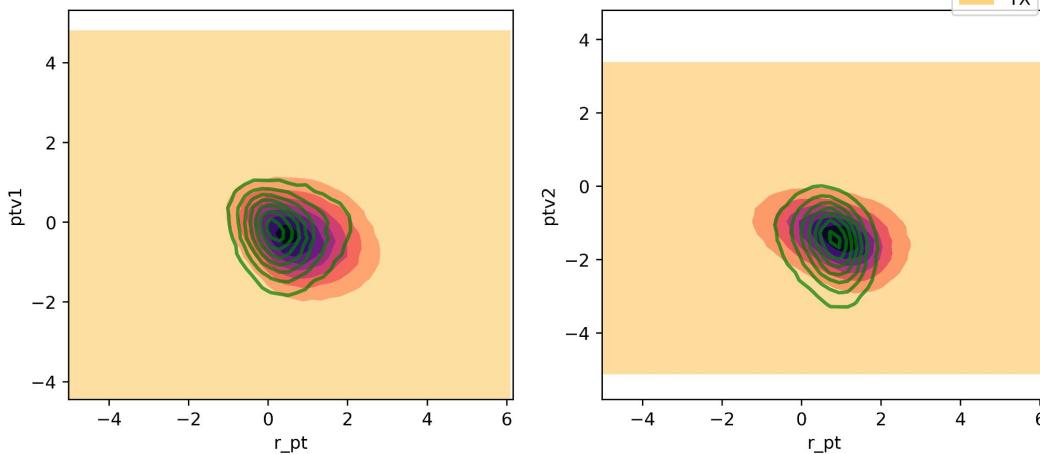
Correlation (2/3)



But if we plot the difference of the correlation matrices, we can observe that certain features present higher correlation amongst the signal (LL) events compared to the background (TX) ones, or viceversa.

Correlation (3/3)

In particular, we see that the main difference in correlation is between leptons and the correspondent neutrinos transverse momenta, and between R_{pT} and neutrino momenta

lep- p_T vs. $\nu-p_T$ feature plot of the LL and TX samplesRpT vs. $\nu-p_T$ feature plot of the LL and TX samples

Model (1/3)

The ML model we built is a DNN in pytorch: we included multiple hidden layers with Dropout and Batch Normalization

Number of hidden layers	3
Hidden layers depth	64, 48, 32 respectively
Dropout	0.1 for each layer
Batch Normalization	in each layer
Output function	Sigmoid
Optimizer	Adam
Learning rate	5e-4

Model (2/3)

Dataset shuffling and splitting	train_test_split scikitlearn function, with a train/test ratio of 75/25
Training batches size	100

We implemented Early Stopping in order to prevent overfitting: specifically, the validation loss is monitored, with a patience of 5 epochs and a delta of 0.01 (i.e. the training stops if the validation loss does not decrease more than 0.01 in 5 epochs)

Value monitored	Validation loss
Patience	5 epochs
Delta	0.01

Model (3/3)

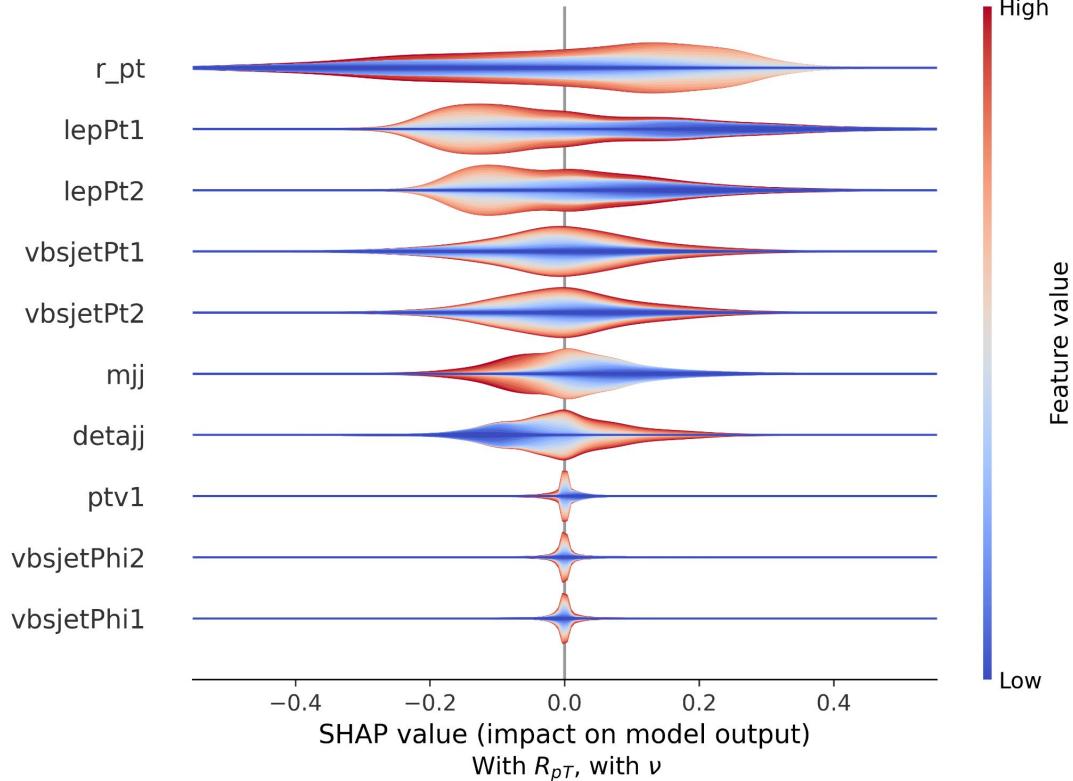
We present the AUC resulting from distinct training runs, differing by the inclusion or exclusion of certain features in the training dataset

AUC	without R_{pT}	with R_{pT}
without neutrinos	0.921	0.994
with neutrinos	0.939	0.996

One of the hypotheses we made before starting the work is that the GEN neutrino features could play an important role in the training, but we didn't observe such a strong dependence. Instead, we note the importance of the R_{pT} feature

SHAP (1/2)

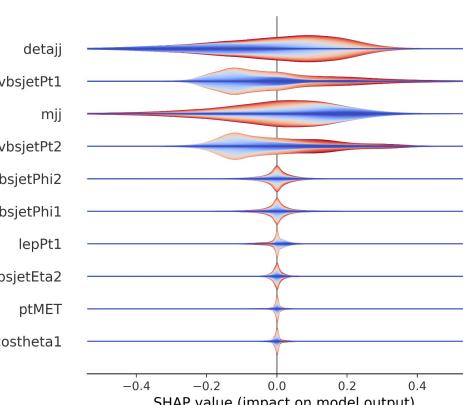
After the training we used the SHAP library (in particular the KernelExplainer) to better understand what are the variables that influence the model the most



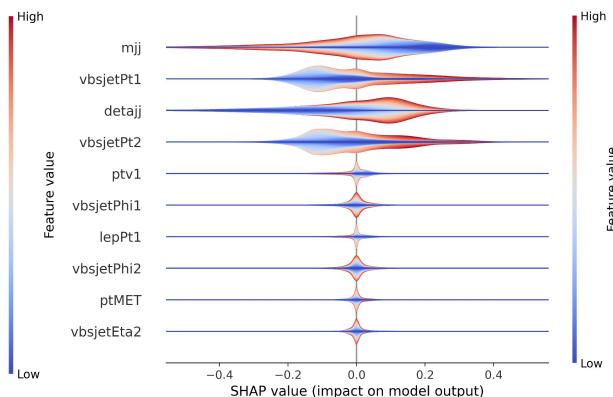
Other features, in order of decreasing importance: ptMET, vbsjetEta2, etav1, etav2, vbsjetM2, costheta1, vbsjetEta1, cos_th_1, lepEta1, costheta2, CScostheta, nvbsjetPt1, lepPhi1, phiMET, ptv2, nvbsjetM1, nvbsjetEta2, phiv1, lepPhi2, vbsjetM1, phiv2, nvbsjetEta1, nvbsjetPt2, nvbsjetPhi2, nvbsjetPhi1, cos_th_2, nvbsjetM2, lepEta2

SHAP (2/2)

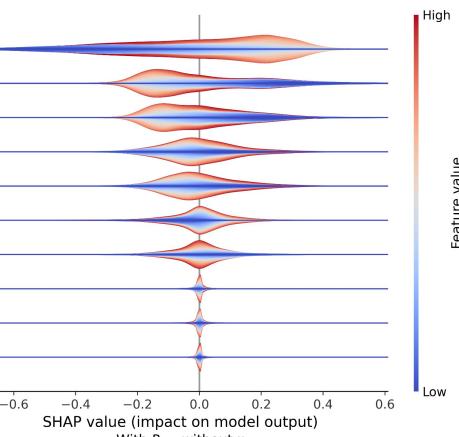
As stressed above, we see that the R_{pT} variable is strong in helping the model in this classification task. It is interesting to note how the importance of the other variables changes when we include R_{pT}



Other features, in order of decreasing importance: lepEta1, phiMET, lepPhi1, CScostheta1, vbsjetEta1, lepPt2, lepPhi2, cos_th_1, lepEta2, costhetaa2, cos_th_2, nvbsjetPt1, vbsjetM2, vbsjetM1, nvbsjetEta1, nvbsjetPt2, nvbsjetM1, nvbsjetPhi1, nvbsjetPhi2, nvbsjetEta2



Other features, in order of decreasing importance: etav1, etav2, ptv2, vbsjetEta1, costhetaa1, costhetaa2, lepEta1, CScostheta, cos_th_1, lepPhi1, lepPhi2, nvbsjetPt1, phiMET, nvbsjetM1, phiV2, vbsjetM1, phiV1, nvbsjetPt2, cos_th_2, lepPhi2, nvbsjetPhi1, vbsjetM2, nvbsjetEta1, nvbsjetEta2, nvbsjetPhi2, nvbsjetM2



Other features, in order of decreasing importance: costhetaa1, lepEta1, ptMET, vbsjetM2, cos_th_1, cos_th_2, lepPhi1, vbsjetEta1, nvbsjetM1, nvbsjetPt1, phiMET, nvbsjetPt2, lepPhi2, nvbsjetEta1, vbsjetM1, nvbsjetPhi1, nvbsjetEta2, costhetaa2, nvbsjetPhi2, lepEta2, nvbsjetM2, CScostheta

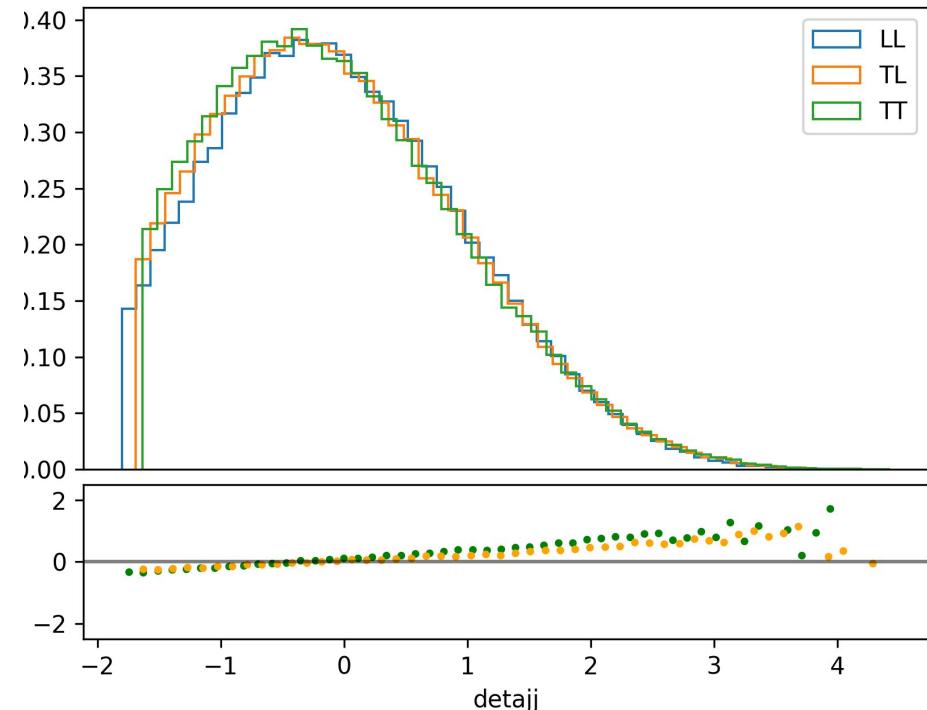
For the code see the repo

<https://github.com/marcortinovis/MLforPolarizedVBS>

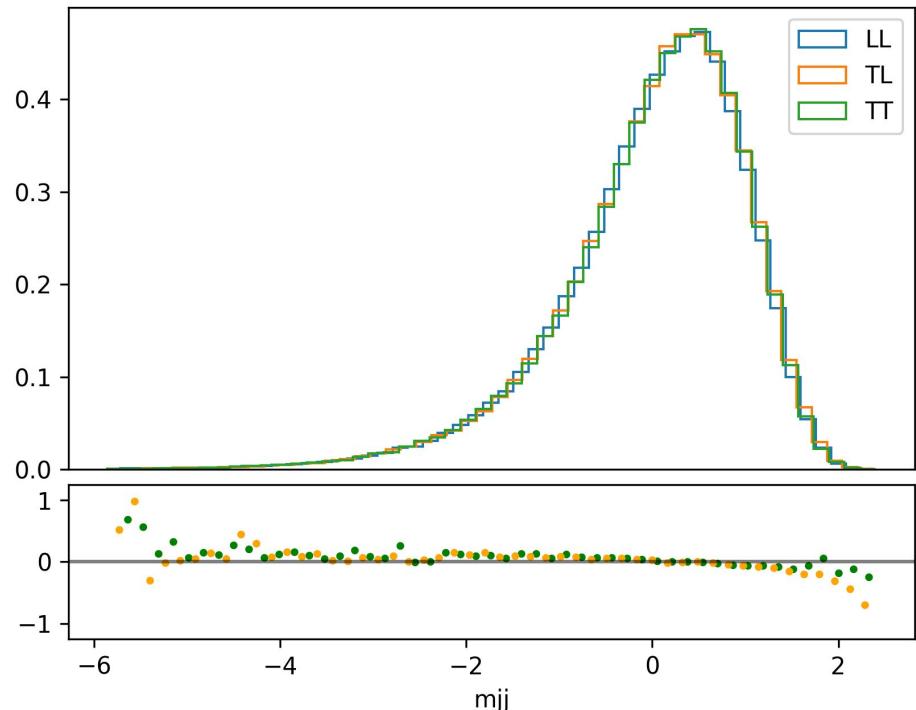
Backup

Most important features visualization

Scaled and normalized detajj histogram

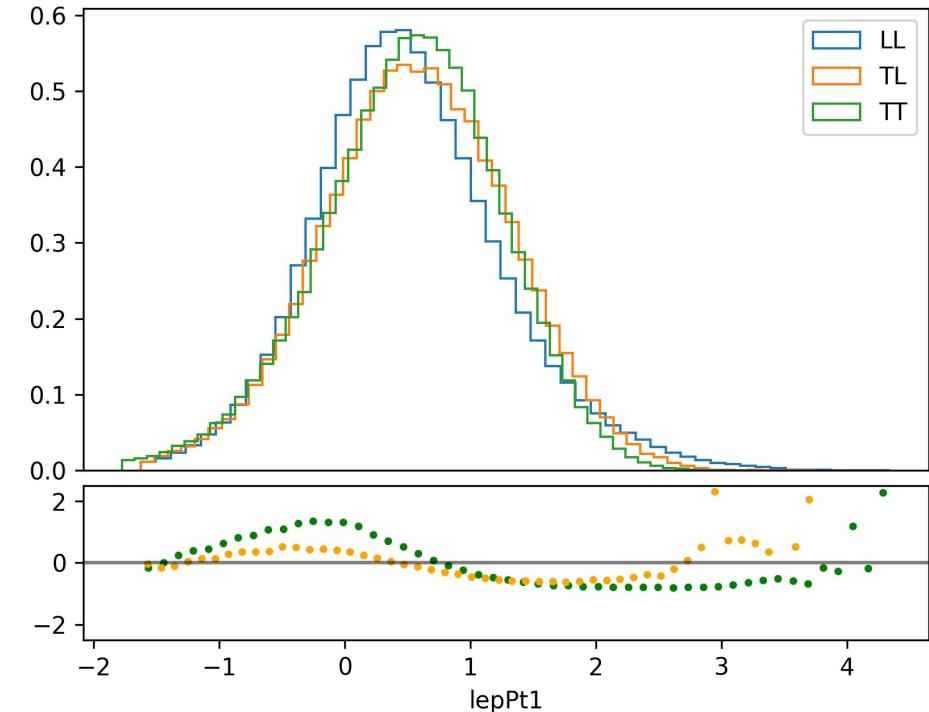


Scaled and normalized mjj histogram

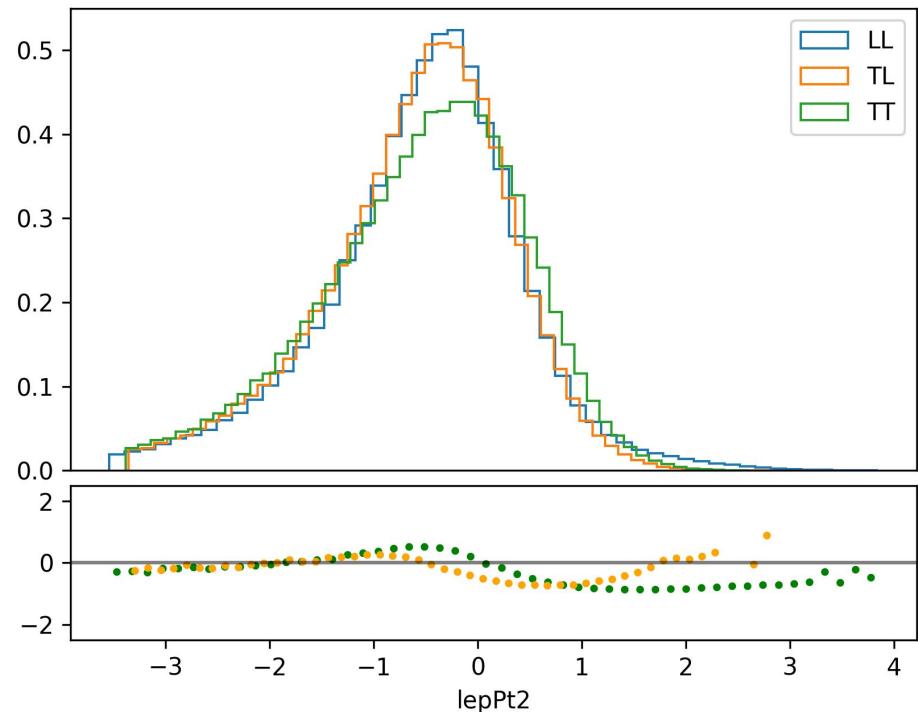


Most important features visualization

Scaled and normalized lepPt1 histogram

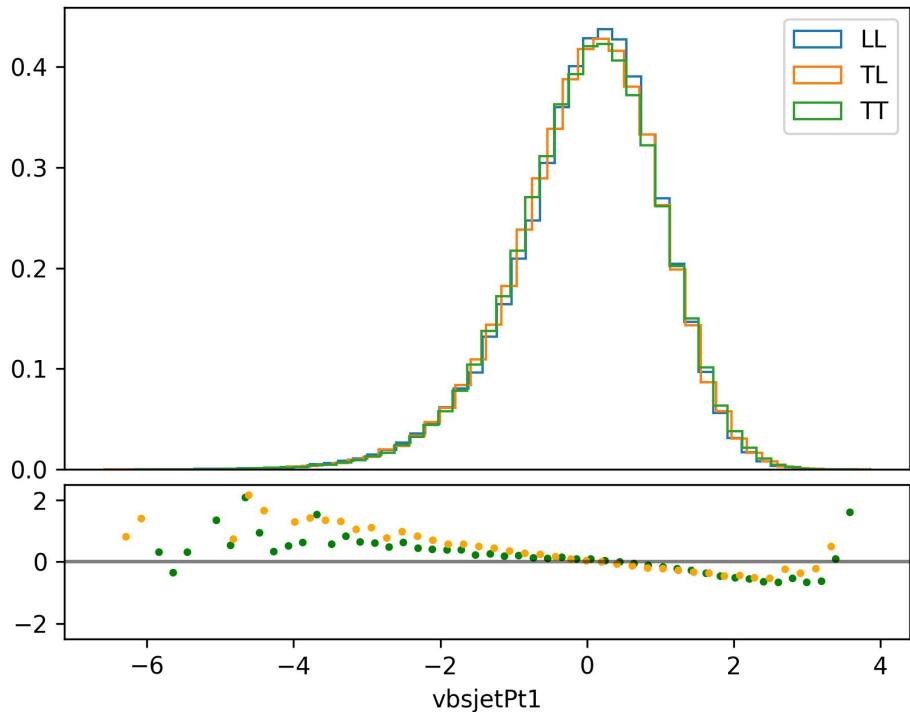


Scaled and normalized lepPt2 histogram

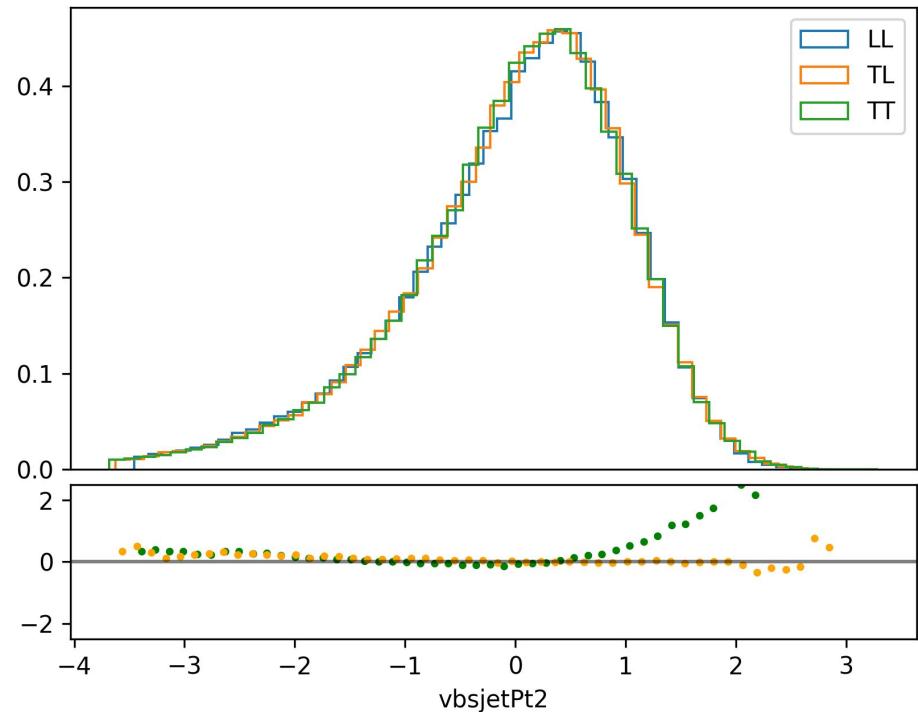


Most important features visualization

Scaled and normalized vbsjetPt1 histogram

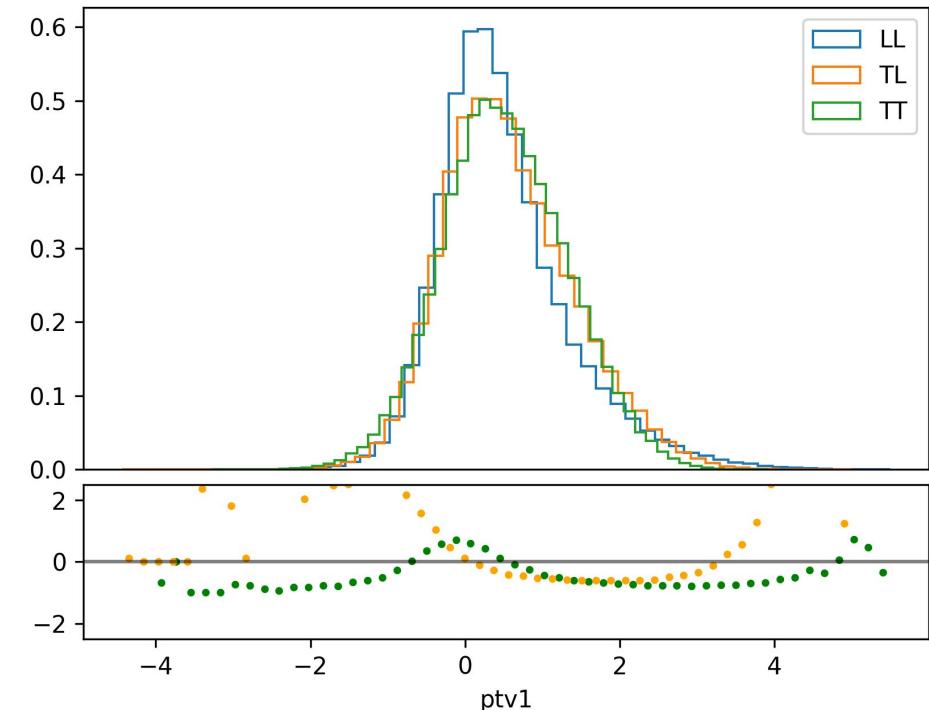


Scaled and normalized vbsjetPt2 histogram

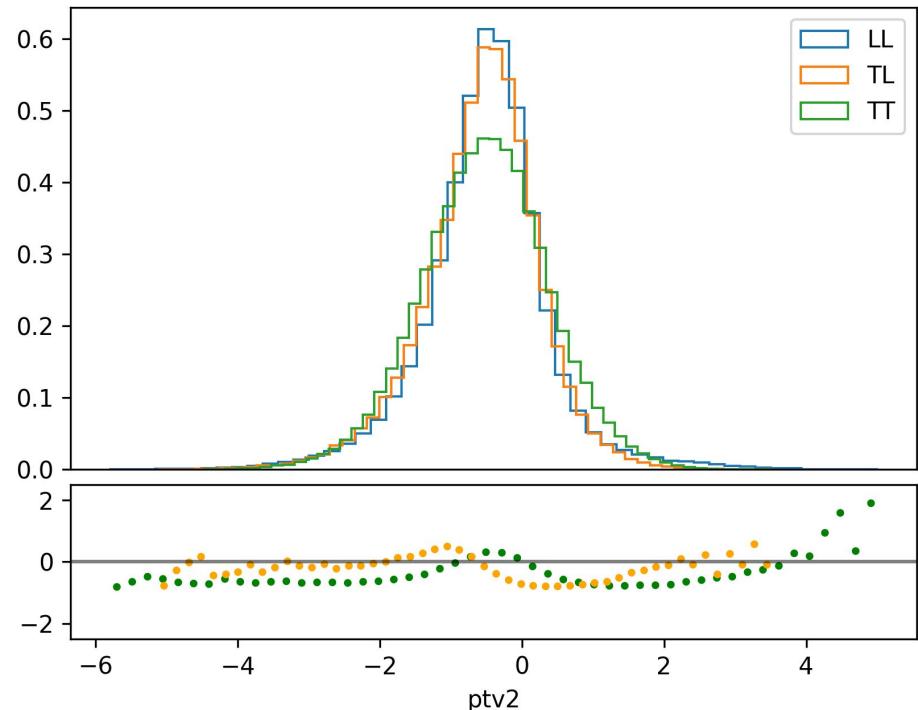


Most important features visualization

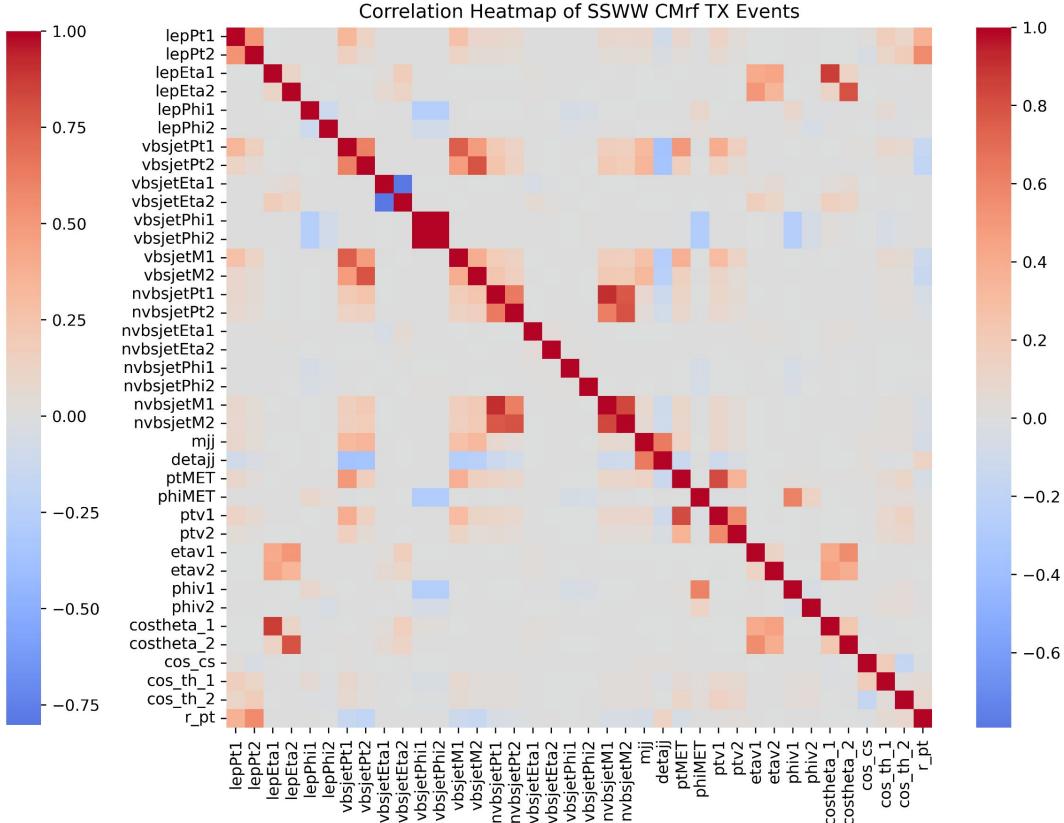
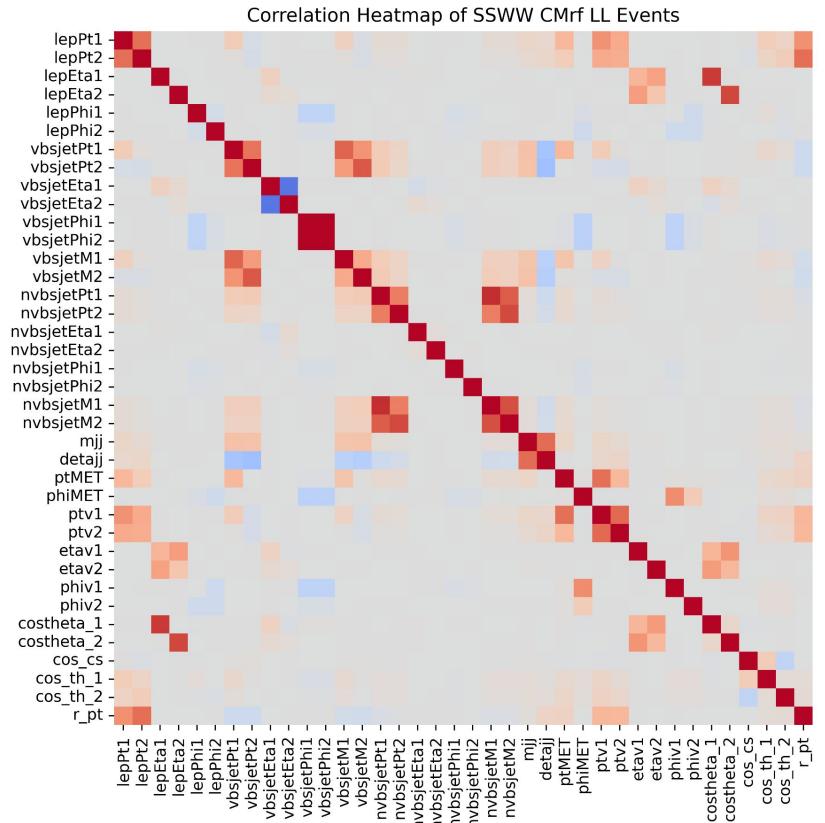
Scaled and normalized ptv1 histogram



Scaled and normalized ptv2 histogram

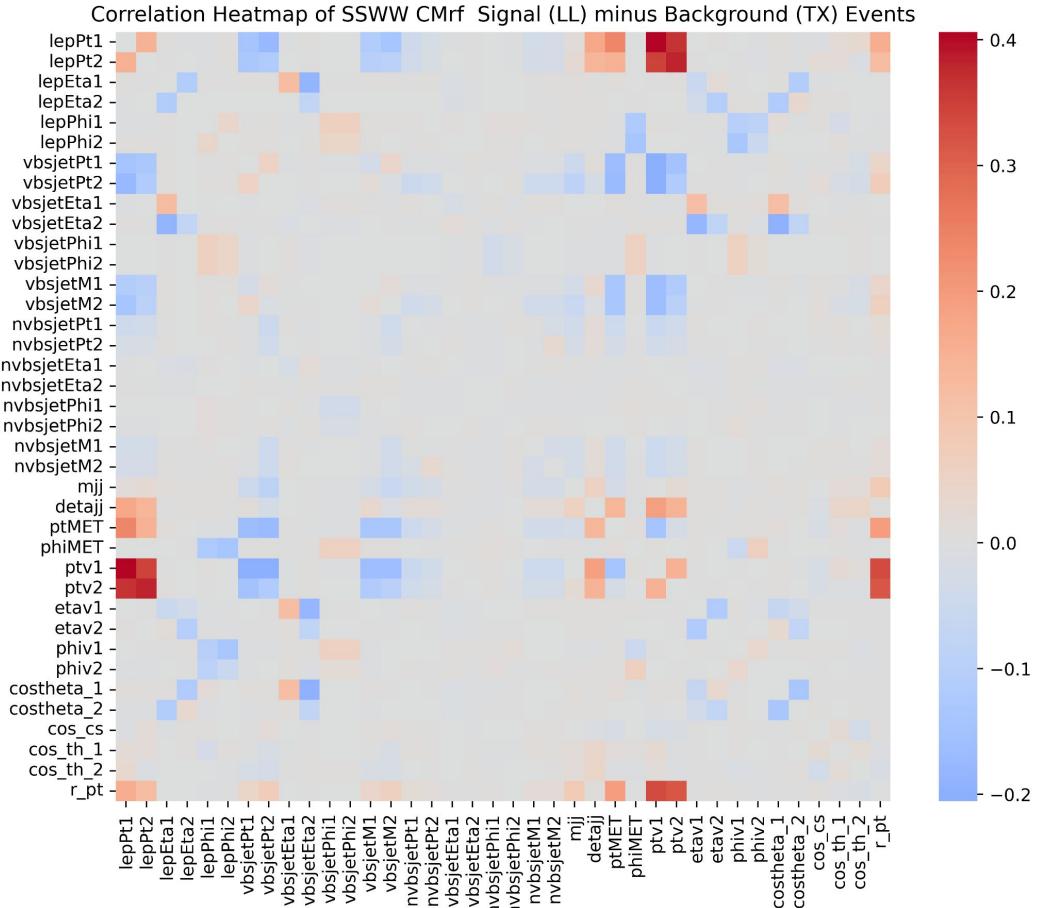


Correlation, source data



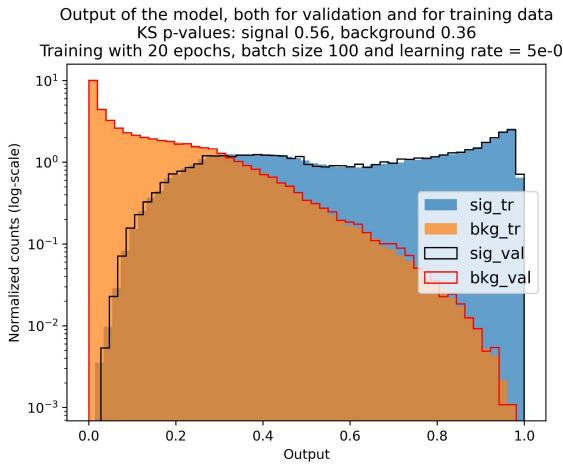
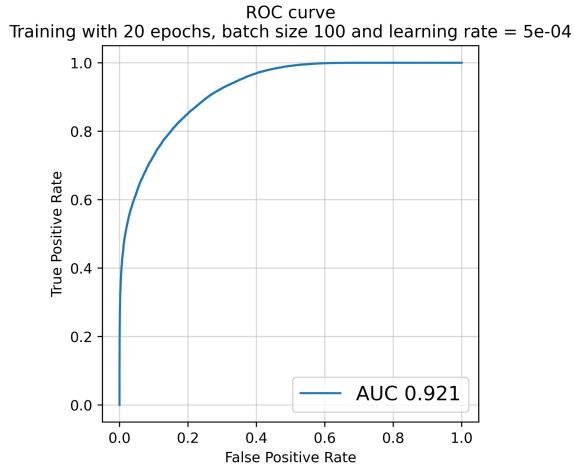
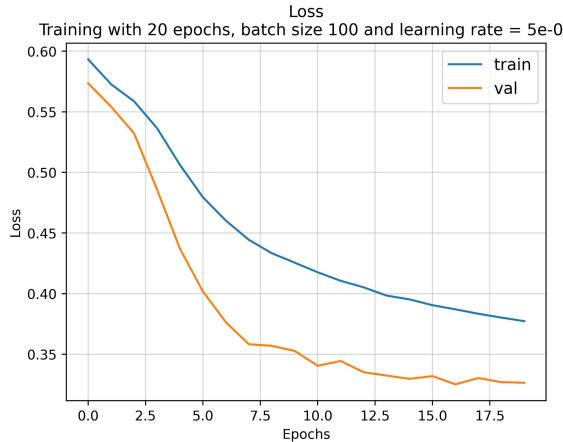
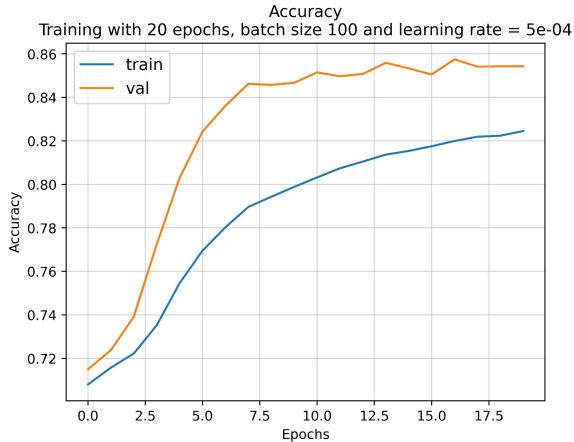
We see no difference with the correlation of scaled and normalized data.

Correlation, source data



We see no difference with the correlation of scaled and normalized data.

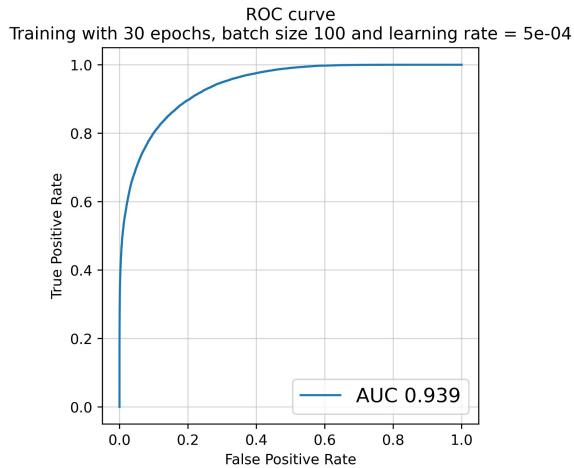
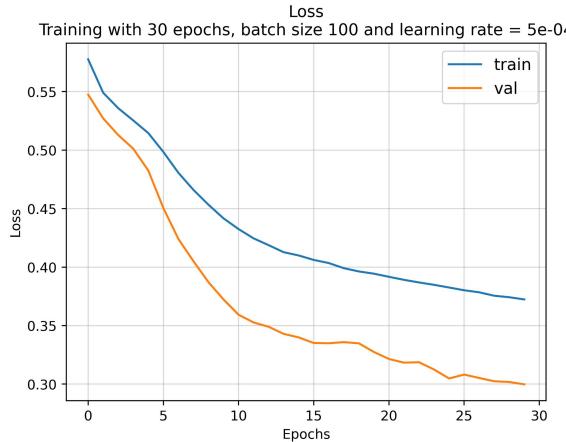
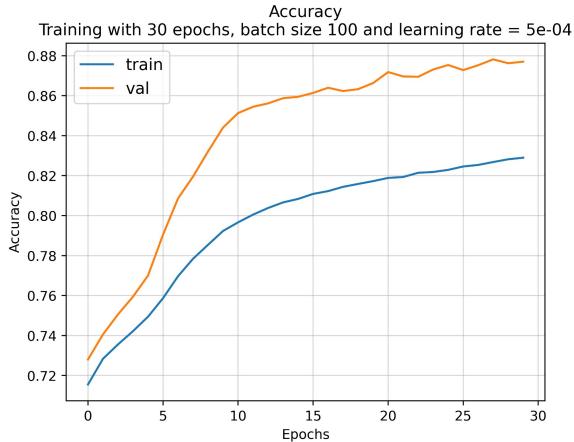
Training results: without R_{pT} and without neutrinos



Note: the validation loss is systematically lower than the training one because of the dropout

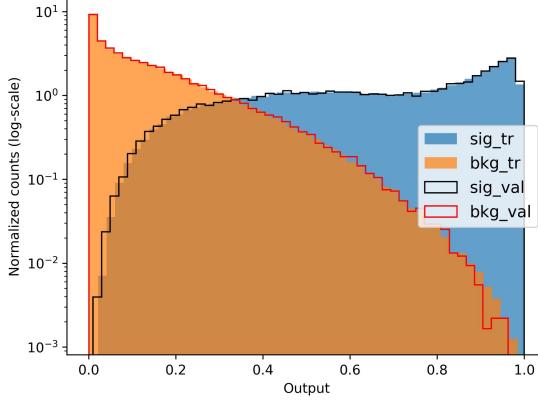
The double secondary peak at ~0.3 still has no clear explanation, but with a better choice of features it decreases

Training results: without R_{pT} and with neutrinos

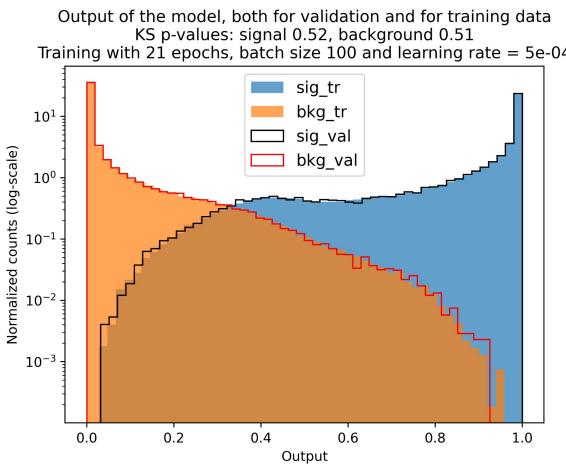
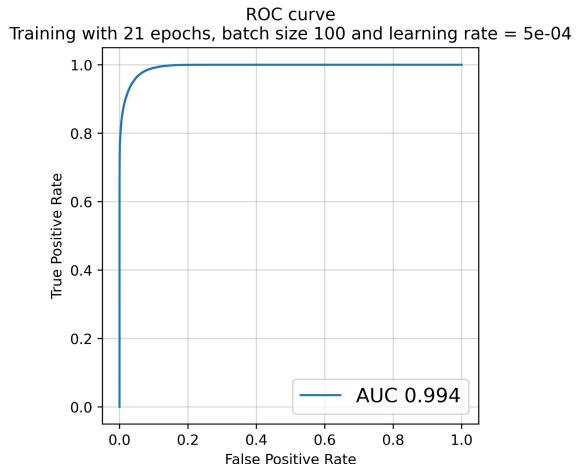
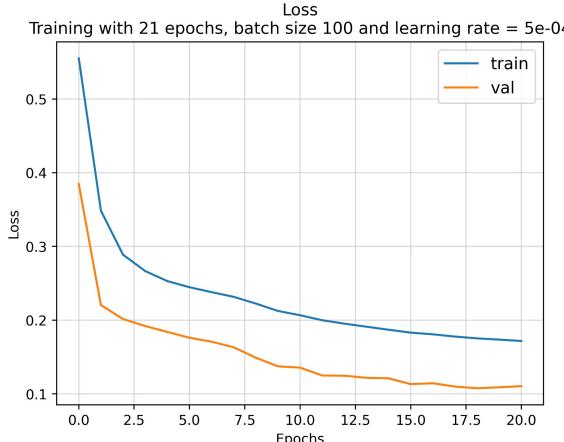
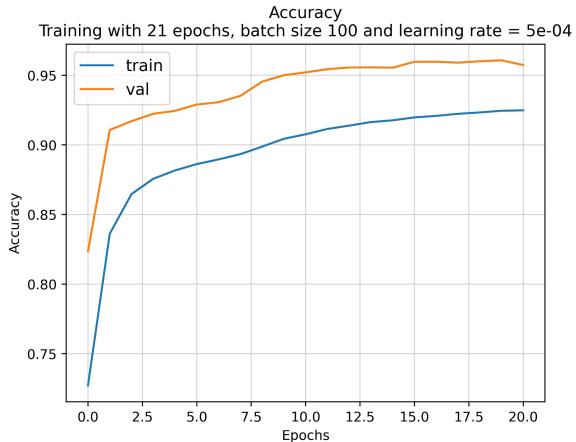


Output of the model, both for validation and for training data
 KS p-values: signal 0.52, background 0.57

Training with 30 epochs, batch size 100 and learning rate = 5e-04



Training results: with R_{pT} and without neutrinos



Training results: with R_{pT} and with neutrinos

