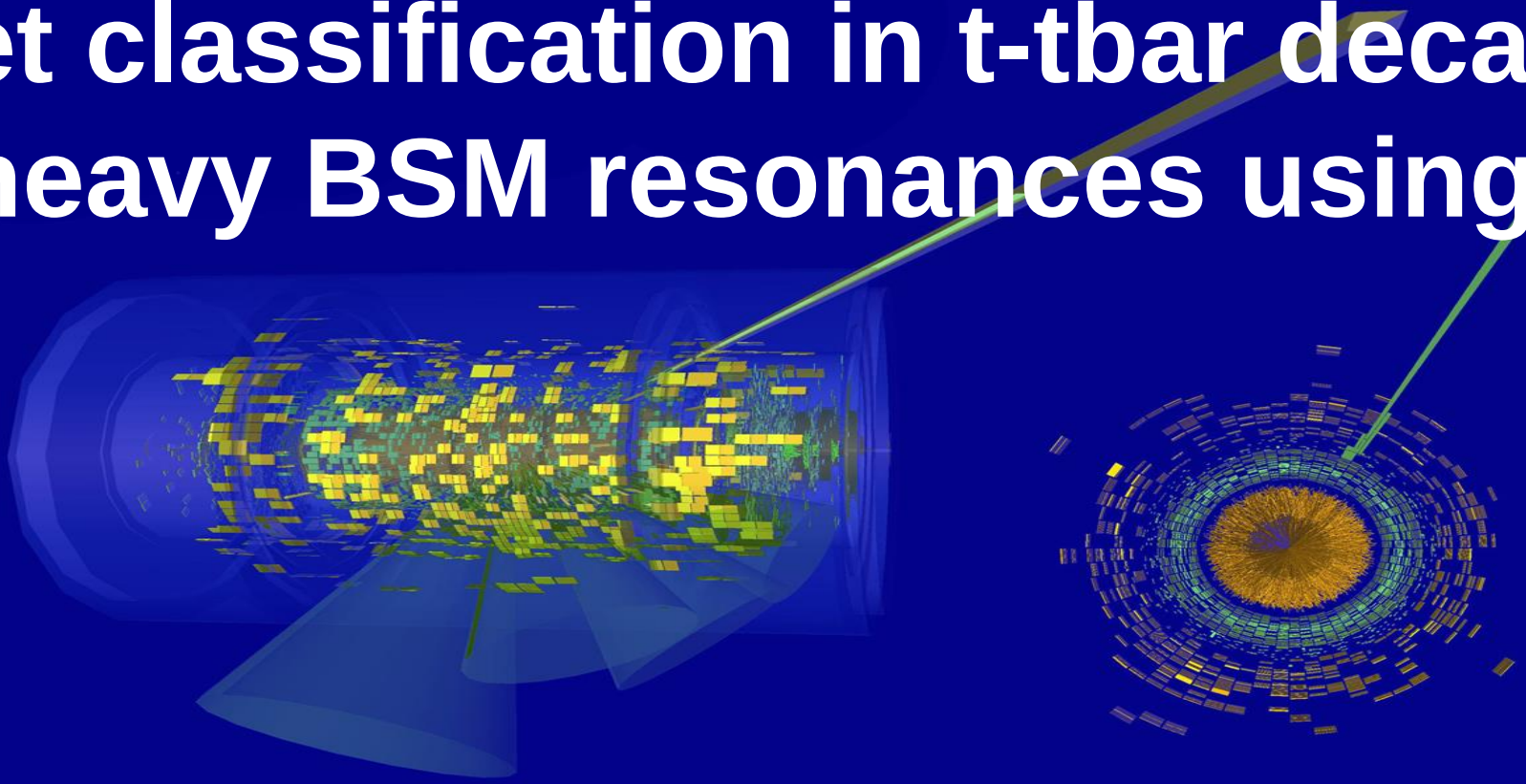
Jet classification in t-tbar decays of heavy BSM resonances using ML



Jorge J. Martínez de Lejarza Samper

Julio Lozano Bahilo

José Salt Cairols

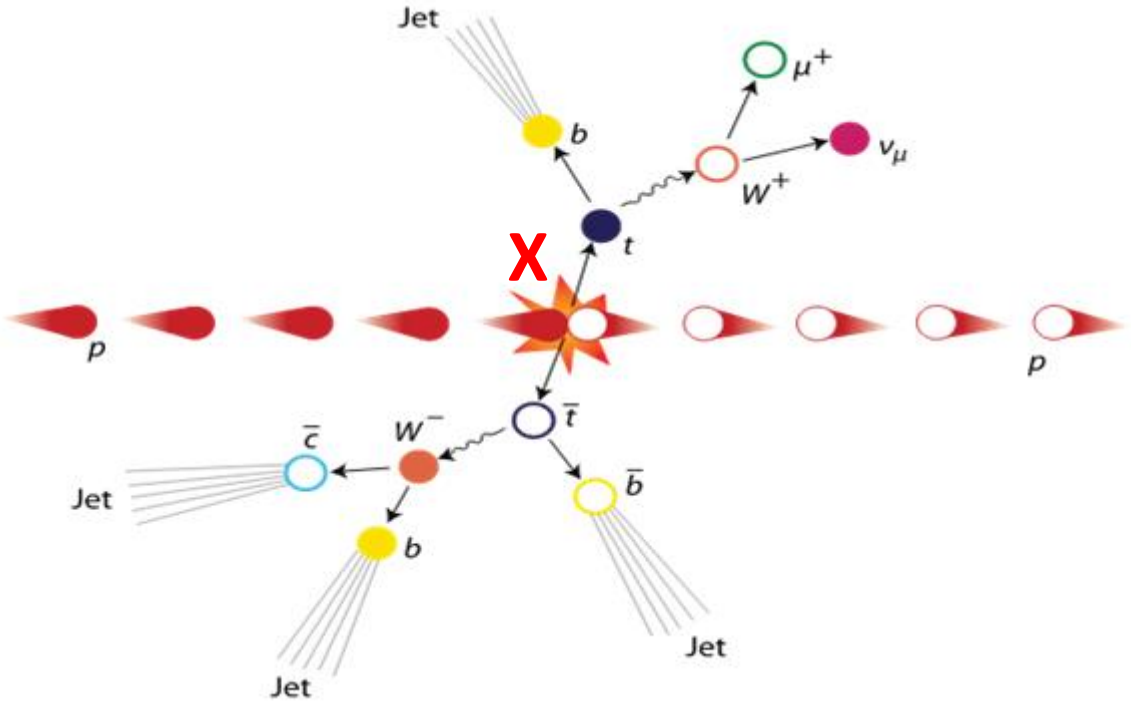
Resonance studies

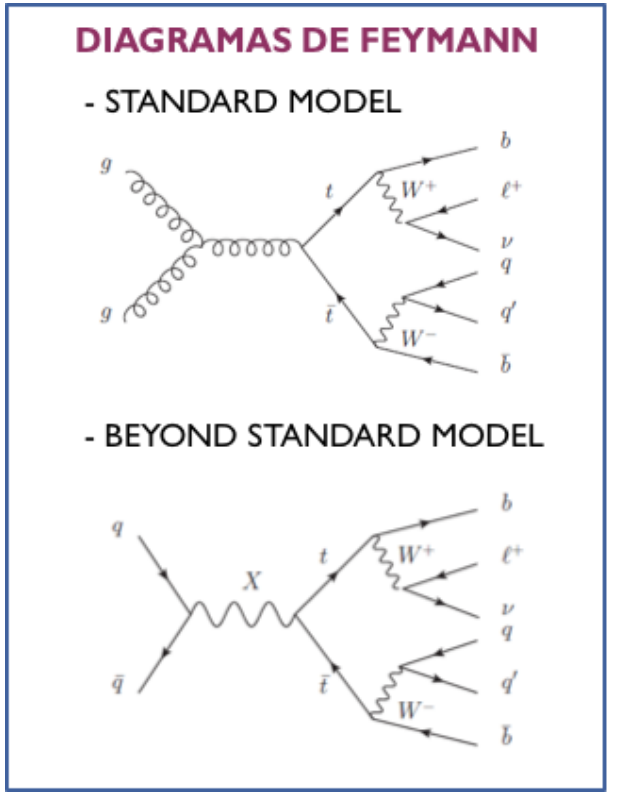
Subject of study

- Events with resonance particles production disintegrated in top quarks pairs
- X could be Z'_{TC2}, G_{KK} o g_{KK} (different masses and widths)

Goal

- Jet classification to get a good resolution in the mass of the resonance





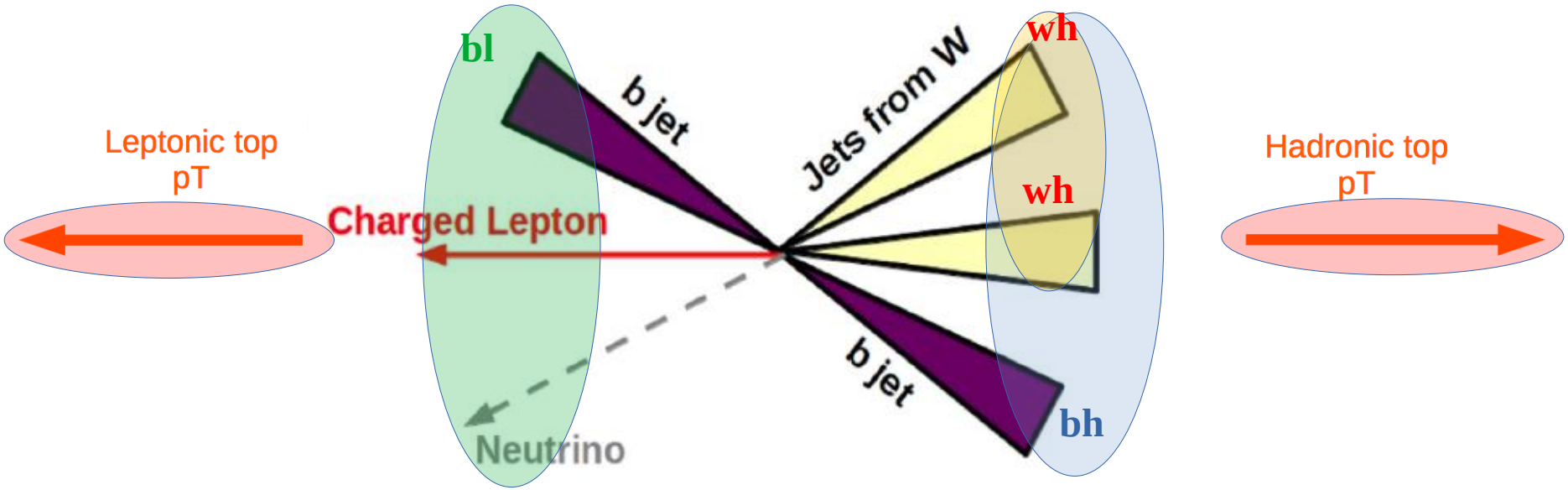
Jet classification

Traditional method: χ^2

- Uses invariant masses and p_T to minimize χ^2
- $\chi^2 = \left[\frac{m_{jj}-m_W}{\sigma_W}\right]^2 + \left[\frac{m_{jjb}-m_{jj}-m_{th-W}}{\sigma_{th-W}}\right]^2 + \left[\frac{m_{jlv}-m_{tl}}{\sigma_{tl}}\right]^2 + \left[\frac{(p_{T,jjb}-p_{T,jlv})-(p_{T,th}-p_{T,tl})}{\sigma_{P_{T,th}-P_{T,tl}}}\right]^2$
- Efficiency ($M_{Z'} = 1\text{TeV}$) ~70%

New methods: ML

- MC events with Z' particles disintegrated in top quarks pairs for ML algorithm training
- Jets classified according to matching variables as :
  - Jet b from the hadronic *decay* of t/tbar quark : **bh**
  - Jet from W *decay* of hadronic *decay* of t/tbar quark : **wh**
  - Jet b from the semileptonic *decay* of ttbar/t quark : **bl**
  - Jet which is not produce in Z' resonance: **oth**



Methodology: *Jet* and event tagging

1. Jet classification :

- A file with relevant variables (27) is created to classify each jet individually
- Using 2/3 of the data for training the ML models
- Multilabel-multiclass problem → *RF,* GB, *XGB* (*eXtreme Gradient Boosting*) *y DNN*

2. Assignment and validation in events:

- Applying the trained algorithm to the test data (1/3) and each piece is assigned. Two different methods:

1) Assigning each jet individually:

- Its label is the highest value given by the ML algorithm
- Possibility of 0/2 bl/bh

2) Each class is assigned to a jet in a orderly way:

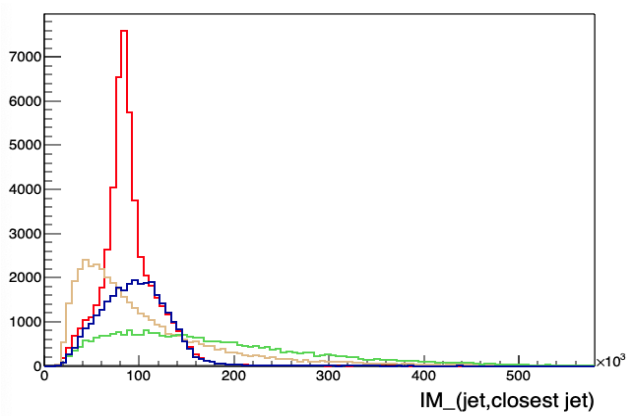
- bl assign to the *jet* with highest probability for this class
- bh assign regarding to the probabilities of the remaining jets
- 2 wh for those with the highest probability among the rest
- All events acquire all their pieces (it would be possible to make cuts in minimum probabilities)

Methodology: Discriminatory variables

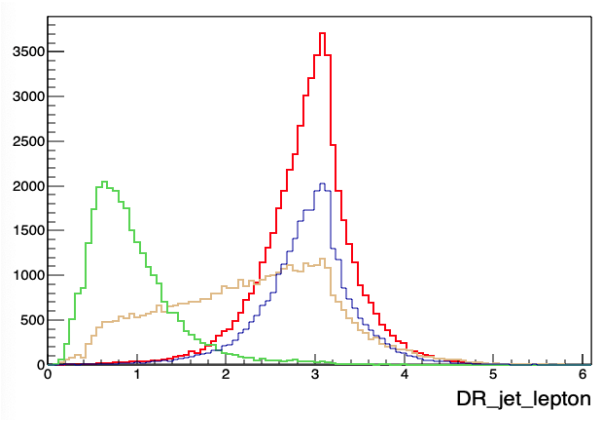
Variables

- Different features that allows us to separate among the distinct types of jets: bh, bl, wh y oth
- Different kind of variables: kinematic, angular and tagging

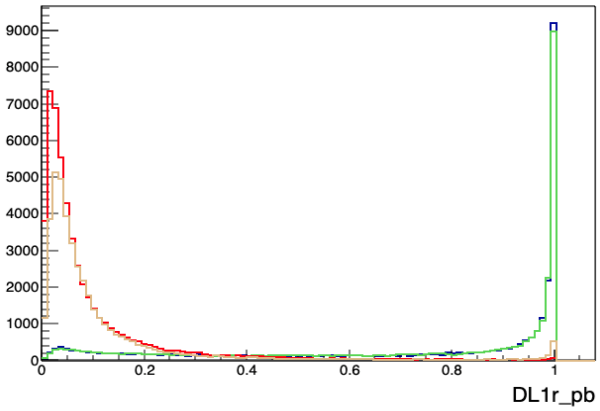
Invariant mass (*jet*, closest *jet*)



ΔR_{jl}



DL1r_pb



Results: Hyperparameters optimization for $M_{Z'}$=1TeV

- Individual optimization of each parameter makes no sense → they are correlated
- Hyperparemeters sweep is performed → to find the best set of hyperparameters

Deep Neural Network (Keras):

- **Best set found:** hidden layers=2, neurons=400, activation='relu'/'sigmoid', dropout=0.4/0.5, loss= 'binary_crossentropy', optimizer= 'adam', metrics ='accuracy', epochs=200, batch_size=256, validation_split=0.0

Random Forest (ScikitLearn):

- **Best set found:** n_estimators= 400 , max_depth=50, min_samples_split=5, min_samples_leaf=6, criterion='mse'

Gradient Boosting (ScikitLearn):

- **Best set found:** n_estimators= 300 , max_depth=15, min_samples_split=600, min_samples_leaf=30, learning_rate=0.05, subsample=0.8

eXtreme Gradient Boosting (XGB):

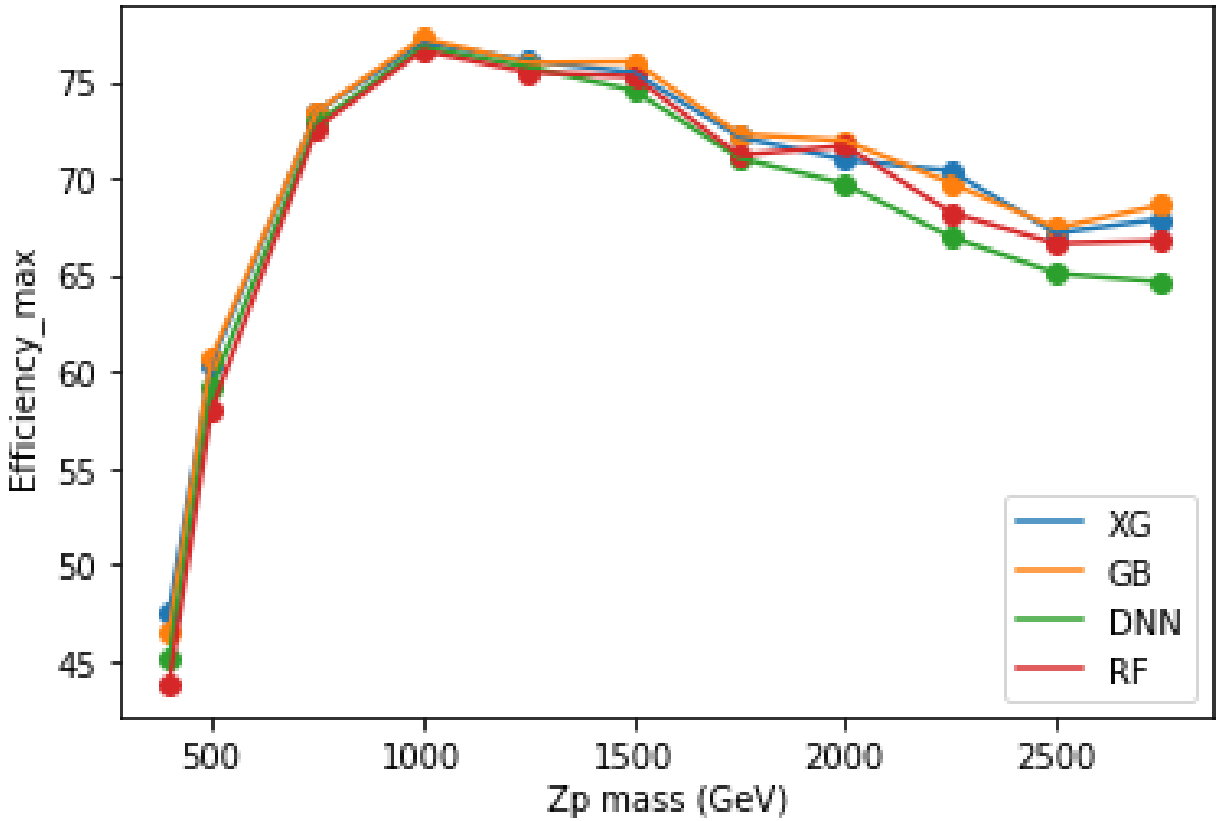
- **Best set found:** n_estimators= 600 , max_depth=6, colsample_bytree=0.9, gamma=0, learning_rate=0.04, subsample=0.7

	DNN	RF	GB	XGB
Default Efficiency (%)	75.1	75.6	74.9	75.4
Optimized Efficiency (%)	76.4	76.6	77.3	77.0
Time execution (s)	1406	184	559	71

Results: Efficiency vs $M_{Z'}$

Reconstruction efficiency vs. $M_{Z'}$:

- Each mass involves its own dataset of training and testing data
- The hyperparameter optimization is performed for each mass individually



χ^2 efficiency ~70% for $M_{Z'}$=1TeV

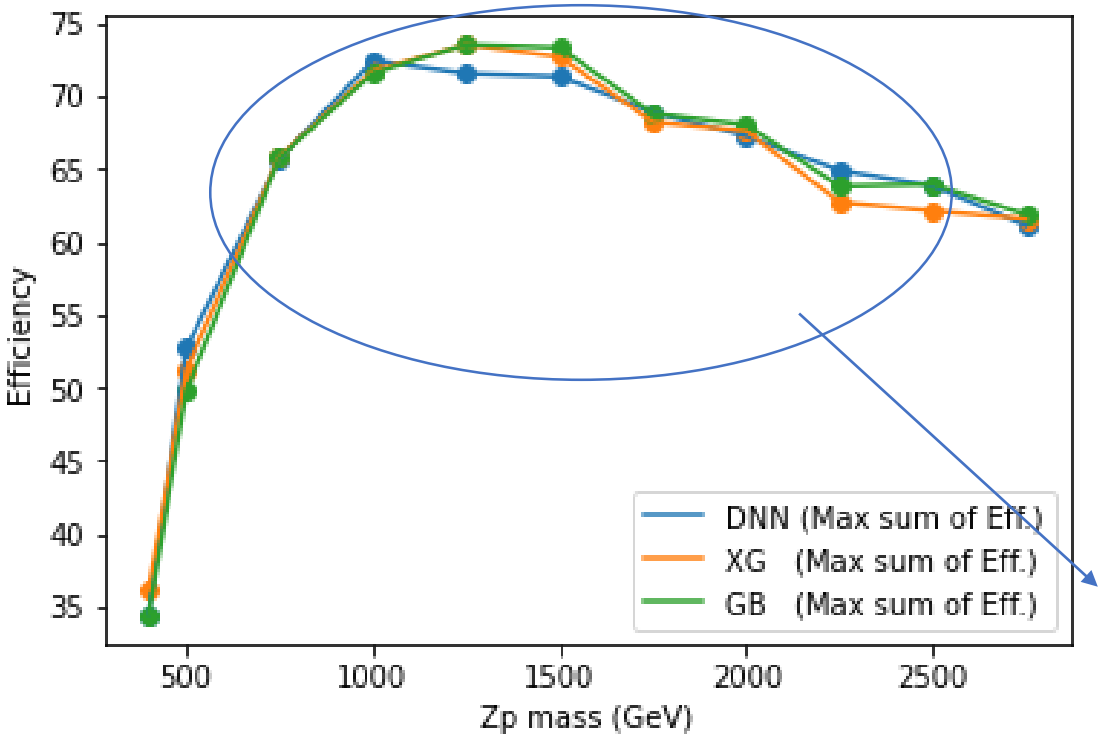
ML efficiency ~77% for $M_{Z'}$=1TeV

Results: Mass parametrization

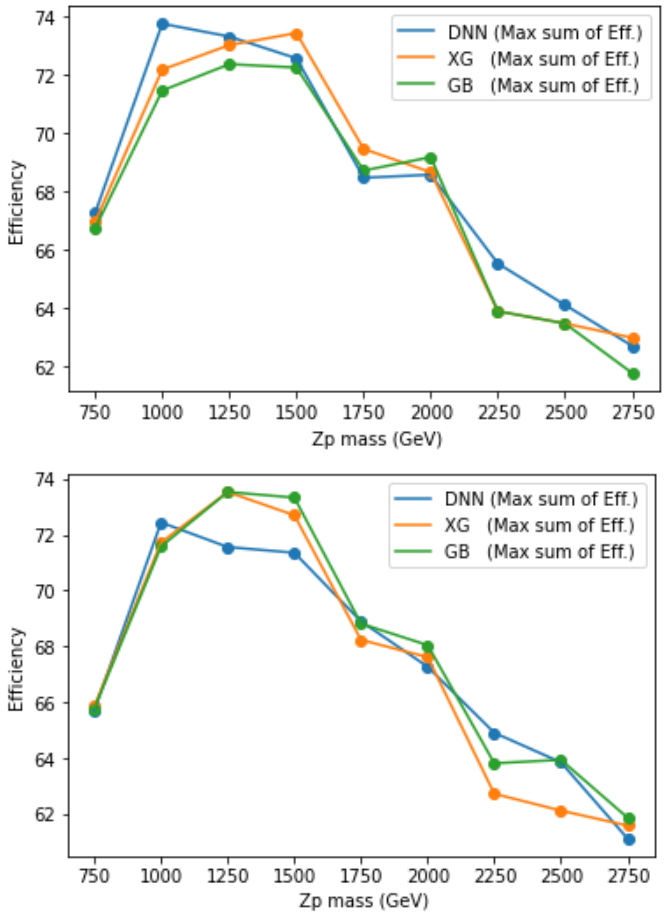
Reconstruction efficiency vs. $M_{Z'}$:

- All the dataset for each mass is joined in a larger dataset, and the mass $M_{Z'}$ is added as an additional variable
- The hyperparameters optimization is performed maximizing the sum over all the efficiencies for each mass

Keeping all the masses



Removing lowest masses

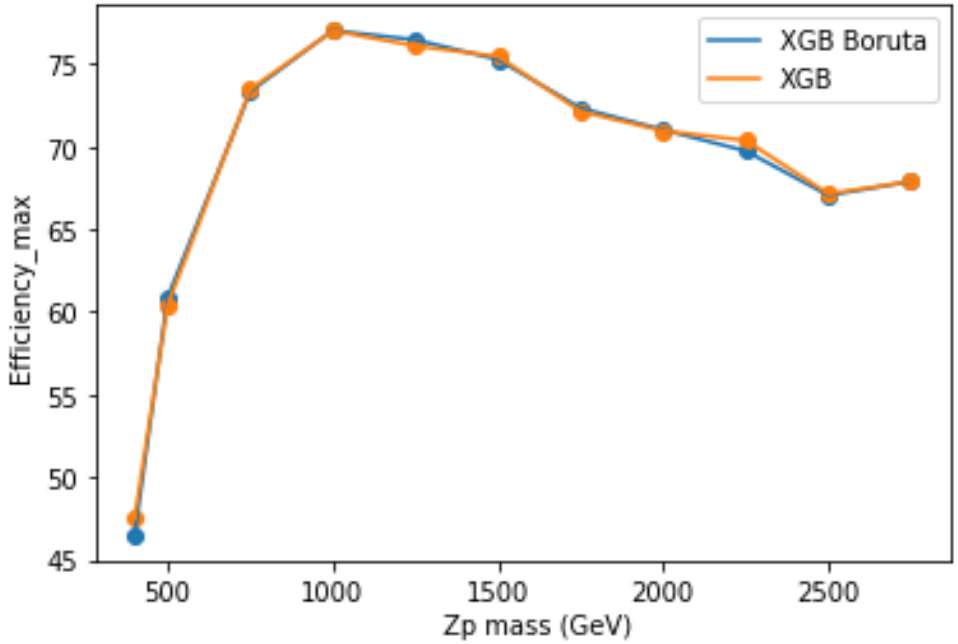


Results: Feature selection

- For DNN → *Permutation Importance*
- For RF → own method in Scikit Learn
- For GB, XGB → Boruta method

Boruta method:

- Performs a selection of the relevant variables for each jet class
- Joining all the selected variables → a reduction from 27 to 22 is achieved



Same efficiency, but lower computational cost!

Further work

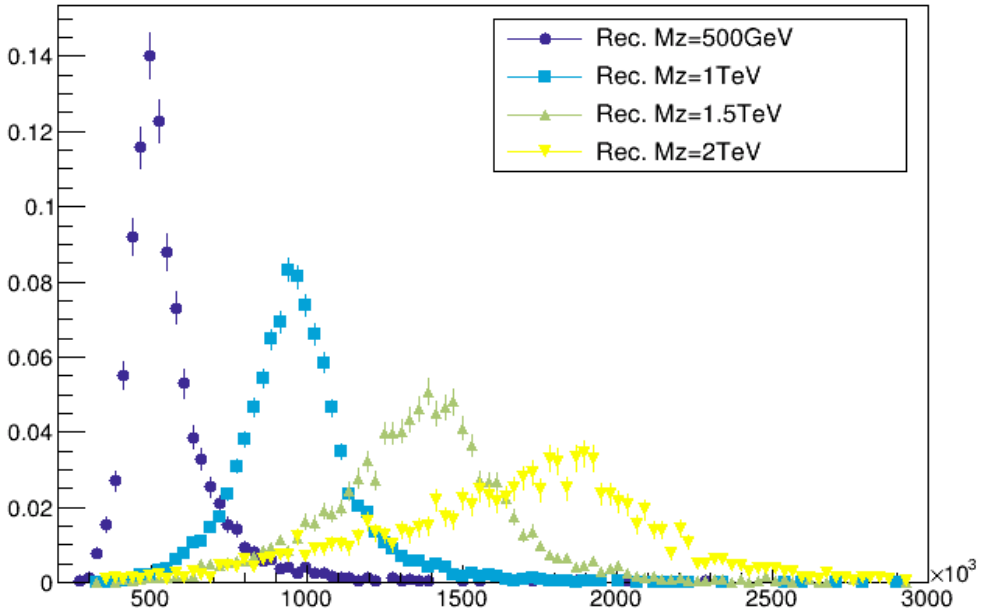
- Improvements in ML techniques
  - *Clustering methods* to the sample to optimize the training of the models (*K-means*, *Expectation-Maximization clustering*, etc …) (Work in progress)
  - *ML explainable:* methods to ease the comprehension of the ML output such as LIME (*Local Interpretable Model-agnostic Explanations*).
- Including χ^2 predictions as variables in our ML models (Work in progress)
- Boosted regime case (high Z' mass): imposible to distinguish among bh and wh → they are joined into a unique *large-R* jet ($\Delta R = 1.0$)
- Study real data and MC background
- Running the ML models on ARTEMISA (GPU)

Thank you for your attention!

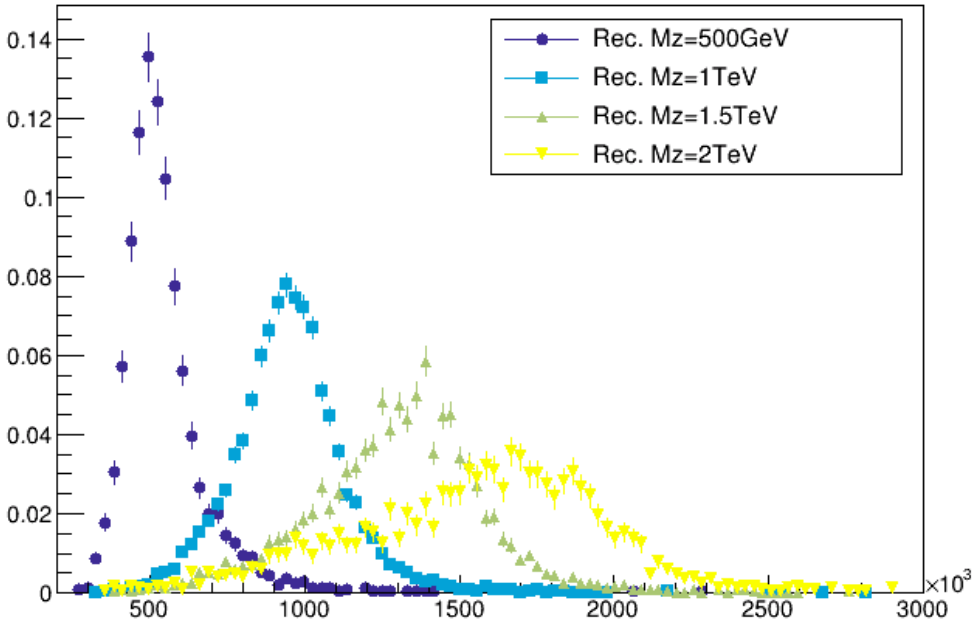
BACKUP

Mass invariant spectrum

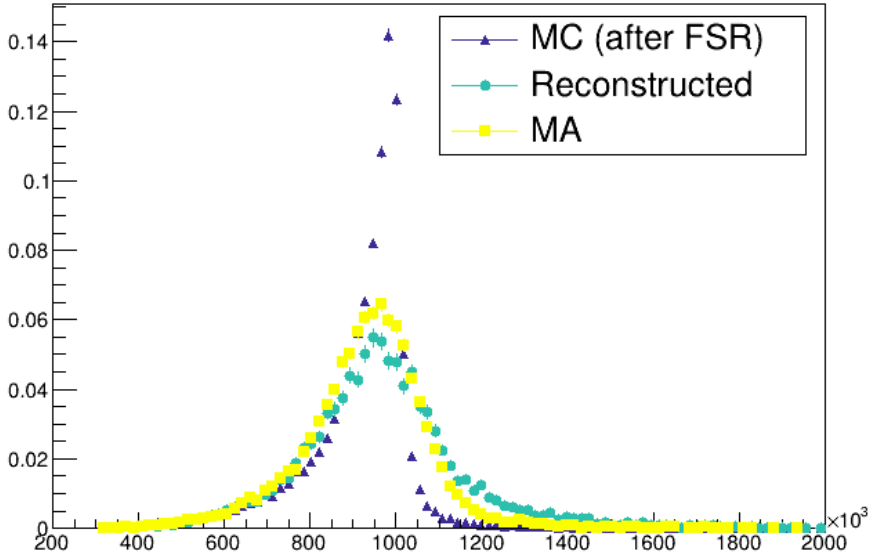
$M_{W\,lep}$ constrain p_ν



Without correction in p_ν



Comparation with MC y *Matching*



Probabilities in each model

	DNN	RF	GB	XGB
bh	76.82	76.92	77.74	77.27
bl	90.28	90.23	91.46	91.03
wh	82.28	83.46	83.80	84.17
oth	79.28	78.54	78.91	78.08

ACADEMIC EXAMPLE:

Method 1:
Method 2:

	bh	bl	wh	oth
jet1	70.82	31.92	77.74	67.27
jet2	90.28	90.23	71.46	61.03
jet3	82.28	83.46	63.80	54.17
jet4	59.28	78.54	68.91	48.08
jet5	39.28	38.54	63.31	78.08