

Reconstruction of di- τ mass using ML based techniques

IN2P3/IRFU Machine Learning Workshop

Janik ANDREJKOVIC, Ece AŞILAR, Janek BECHTEL, Colin BERNET, Sebastian BROMMER, Maximilian BURKART, Artur GOTTMANN, Mareike MEYER, Günter QUAST, Aliaksei RASPIAREZA, Lucas TORTEROTOT, Roger WOLF, Sebastian WOZNIOWSKI, Stefan WUNSCH

IP2I (Lyon) & MSSM $H \rightarrow \tau\tau$ analysis group

March 16th 2021

- 1 Few words on di- τ events
- 2 Building the ML model
 - Lucas TORTEROTOT, Ece AŞILAR, Colin BERNET (IP2I Lyon), Sebastian WOZNIEWSKI (KIT)
- 3 Comparison against SVFIT
 - Lucas TORTEROTOT & the $H \rightarrow \tau\tau$ MSSM analysis group

1 Few words on di- τ events

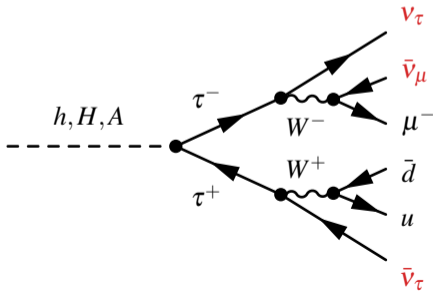
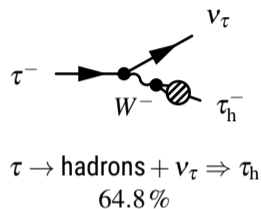
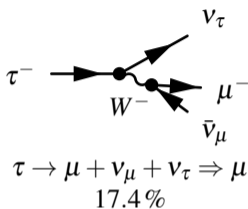
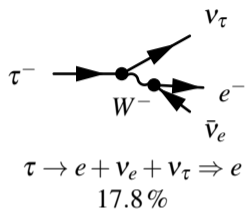
2 Building the ML model

- Lucas TORTEROTOT, Ece AŞILAR, Colin BERNET (IP2I Lyon), Sebastian WOZNIEWSKI (KIT)

3 Comparison against SVFIT

- Lucas TORTEROTOT & the $H \rightarrow \tau\tau$ MSSM analysis group

$X \rightarrow \tau\tau$ events: 2 to 4 neutrinos in final state!



- ▶ 6 different channels:
 - ▷ $\tau_h \tau_h$: 2 neutrinos,
 - ▷ $\mu \tau_h, e \tau_h$: 3 neutrinos,
 - ▷ $\mu\mu, e\mu, ee$: 4 neutrinos.
- ▶ Neutrinos = invisible in CMS!
- ▶ Impossible to get the full invariant mass.

1 Few words on di- τ events

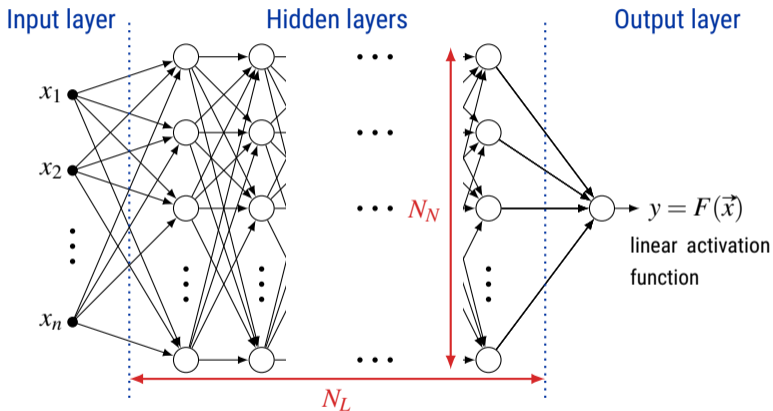
2 **Building the ML model**

- Lucas TORTEROTOT, Ece AŞILAR, Colin BERNET (IP2I Lyon), Sebastian WOZNIEWSKI (KIT)

3 Comparison against SVFIT

- Lucas TORTEROTOT & the $H \rightarrow \tau\tau$ MSSM analysis group

Build a neural network: deep but simple structure!



► Here $N_L = 3$ and $N_N = 1000$

Build a neural network: chosen hyperparameters

- ▶ NN hyperparameters (and tested values):
 - ▷ **Adam** optimizer (Adam, Adadelta, SGD),
 - ▷ Weight initialized with **Glorot uniform** (Glorot normal, Glorot uniform, normal, uniform),
 - ▷ **MAPE** loss (MAPE, MSE),

$$MAPE(y_{true}, y_{pred}) = \frac{100}{N} \sum_{i=1}^N \left| \frac{y_{pred,i} - y_{true,i}}{y_{true,i}} \right|$$

- ▷ **Softplus** activation function (ReLU, ELU, SELU, Exponential, Softplus),

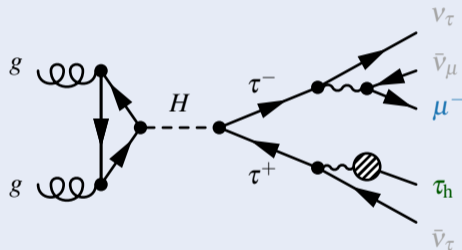
$$softplus(x) = \ln(1 + e^x)$$

- ▷ 3 hidden layers (2 to 5),
- ▷ 1000 neurons per hidden layer (200, 600, 1000, 1400 for all other hyperparameters + 200 to 2000 per steps of 100 after focus),

Build a neural network: target and inputs?

- ▶ Model target: generated Higgs mass.
- ▶ Model inputs:
 - ▷ τ_1 (here = μ^-) and τ_2 (here = τ_h) p_T, η, ϕ ;
 - ▷ PuppiMET p_T, ϕ ;
 - ▷ METcov xx, xy and yy ;
 - ▷ $m_T^{(1, MET)}, m_T^{(2, MET)}, m_T^{(1,2)}, m_T^{tot}$ (Puppi);
 - ▷ jet 1, jet 2 p_T, η, ϕ ;
 - ▷ remaining hadronic activity p_T, η, ϕ ;
 - ▷ npvsGood \rightarrow how much PU;
 - ▷ Number of neutrinos from tau decays.

$$gg \rightarrow H \rightarrow \tau\tau \rightarrow \mu\tau_h$$



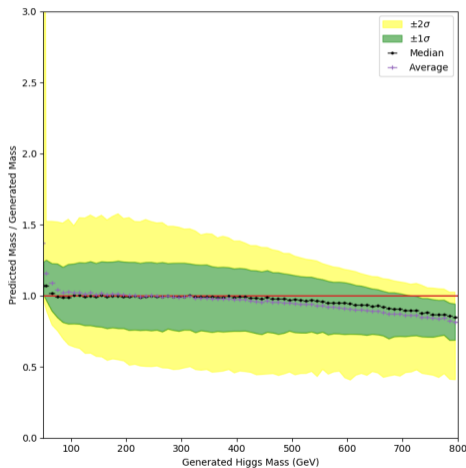
$$m_T^{tot} = \sqrt{m_T^2(\tau_1, E_T^{miss}) + m_T^2(\tau_2, E_T^{miss}) + m_T^2(\tau_1, \tau_2)}, \quad m_T(1,2) = \sqrt{2p_T^{(1)} p_T^{(2)} (1 - \cos \Delta\phi)}$$

Model trained on $ggH \rightarrow \tau\tau$ samples with floating m_h

Model response

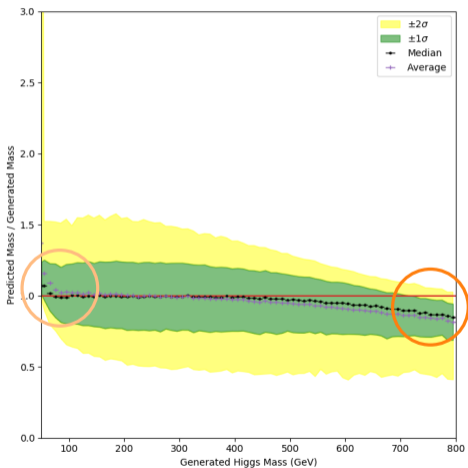
Model response as function of target \rightarrow

- ▶ Reconstruction achieved from 70 to ~ 500 GeV!
 - ▷ Below 70 GeV: small overestimation.
 - ▷ Above 500 GeV: small underestimation.
 - ▷ Due to the dataset boundaries!

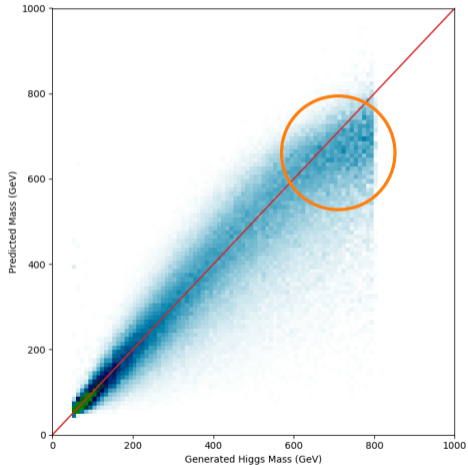


Upgrade of the model: training sample boundaries effects

Model response

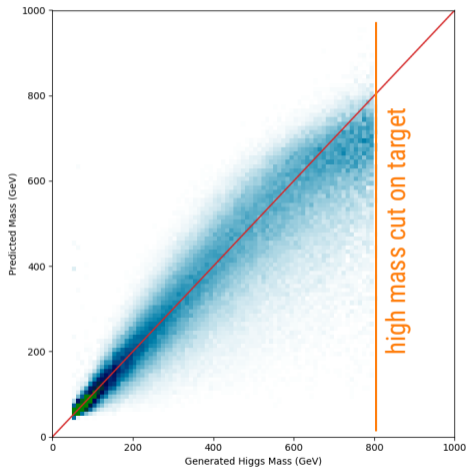


Predictions vs truth



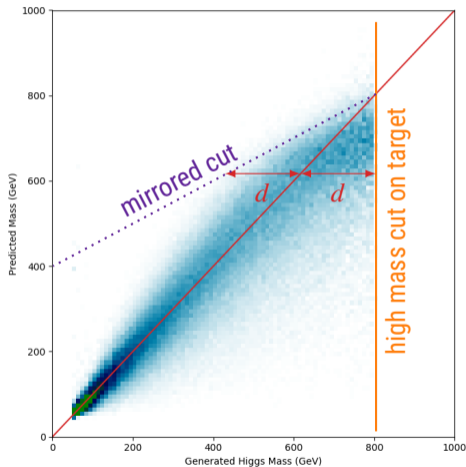
Upgrade of the model – origin of the over/under-estimations

- ▶ Due to the 800 GeV cut on the target:
 - ▷ The NN never sees events above 800 GeV;
 - ▷ The NN is not likely to predict above 800 GeV.
- ▶ Same (reversed) effect at low mass!
- ▶ **How to cope with the boundaries?**
 - ▷ Bias to be balanced,
 - ▷ Extend the mass range?
 - ▷ Would be nice!
 - ▷ Not always feasible...



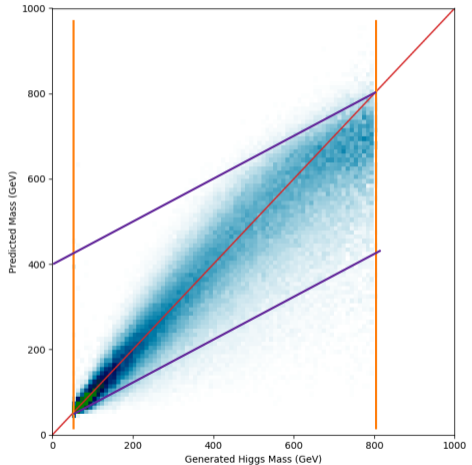
Upgrade of the model – mirror the boundaries!

- ▶ The hard cut at 800 GeV is equivalent to having events but not counting them in the loss.
- ▶ We test the following:
Take events into account for the loss only if they are within the mirrored boundary *i.e.* if they are not further away from the red line than the red line to the boundary.



Upgrade of the model – mirror the boundaries at low mass too

Take events into account for the loss only if they are within the mirrored boundary *i.e.* if they are not further away from the red line than the red line to the boundary.



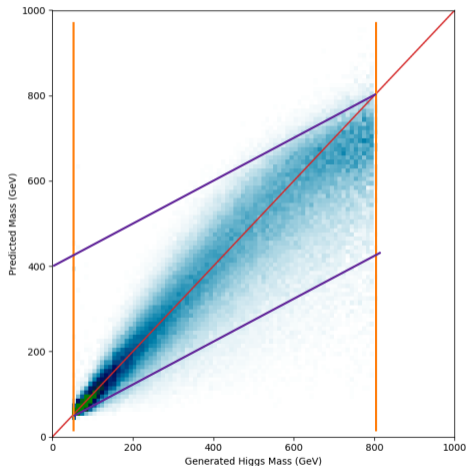
Upgrade of the model – tuning the loss in the central region

- ▶ MAPE loss is

$$MAPE(y_{\text{true}}, y_{\text{pred}}) = \frac{100}{N} \sum_{i=1}^N \left| \frac{y_{\text{pred},i} - y_{\text{true},i}}{y_{\text{true},i}} \right|$$

- ▶ We found out that better constrains at high masses is obtained with $MAPE \times \sqrt{y_{\text{true}}}$

$$MAPE_{\text{sqrt}}(y_{\text{true}}, y_{\text{pred}}) = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_{\text{pred},i} - y_{\text{true},i}}{\sqrt{y_{\text{true},i}}} \right|$$



Upgrade of the model – custom loss

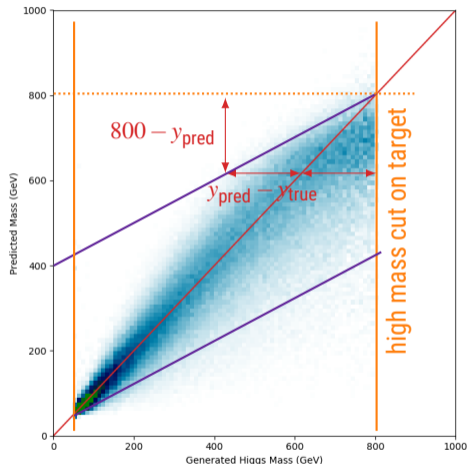
► custom loss = $MAPE_{sqrt} \times$ boundary cuts:

```

1 def custom_loss(y_true, y_pred):
2     loss = tf.abs(
3         (y_true - y_pred)/(y_true**.5) * where(
4             kb.greater_equal(
5                 y_pred - y_true, 800 - y_pred
6             ),
7             0.0,
8             where(
9                 kb.greater_equal(
10                    y_true - y_pred, y_pred - 50
11                ),
12                0.1,
13                1.0,
14            )))
15     return tf.reduce_mean(loss, axis=-1)

```

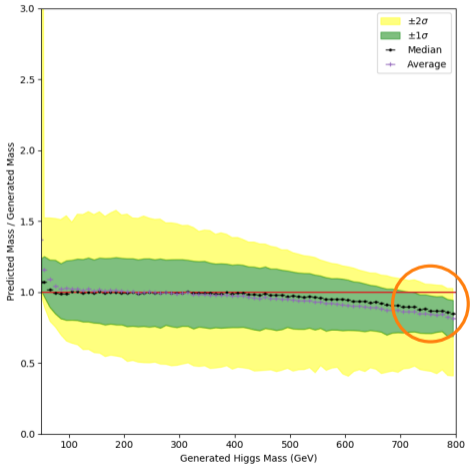
- line 5: upper boundary condition.
- line 10: lower boundary condition.



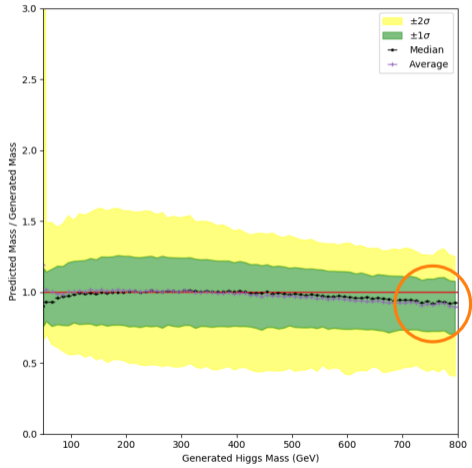
► Convergence problems can appear \Rightarrow the 0.1 factor for low mass is applied because of that (instead of 0.0).

Custom loss reduces the boundaries effect by a factor of 2!

Native MAPE loss



Custom loss



Notes on the loss

- ▶ The loss function obtained does **not** respect the condition:

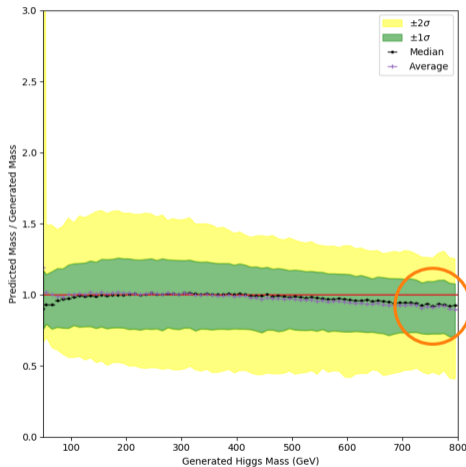
$$Loss = 0 \Leftrightarrow y_{\text{pred}} = y_{\text{true}}$$

i.e. loss can be 0 even with predictions different from truth ...

- ▶ Yet the boundary effect is reduced with this loss!
 - ▷ This confirms our interpretation.
- ▶ This loss also provides a good model.
- ▶ Other possibility:
 - ▷ For each y_{pred} value, instead of setting a mirrored boundary hard cut:
 - ▷ tune weights to have equal weighted-quantity of events above and below y_{true}
- ▶ Did anyone encounter the same boundary issue?
 - ▷ Regardless of our case: what if generating other mass points (lower and higher) can't be done?

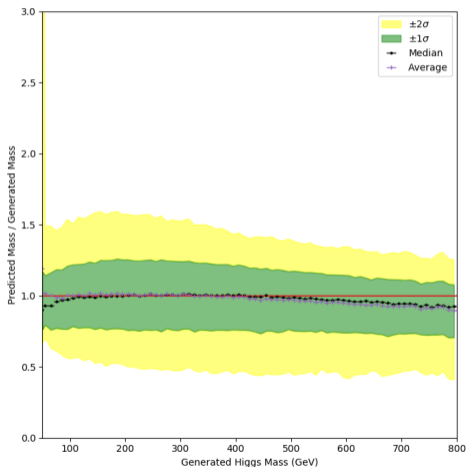
Native MAPE vs custom loss: custom reduces the boundaries effect!

- ▶ We still have a small underestimation at high mass;
- ▶ Take advantage of the generated Higgs width:
 - ▷ Setting $m_H = 800\text{ GeV}$ we end up with events at 850 GeV , 1 TeV , ...

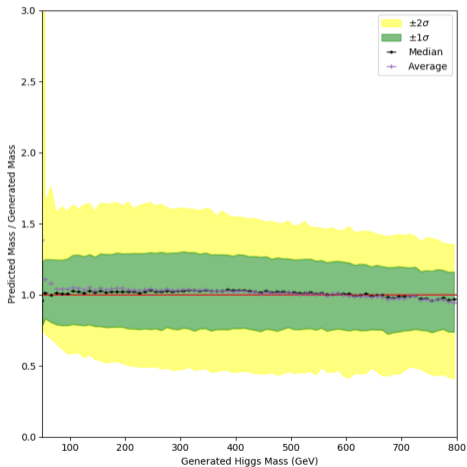


Upgrade of the model – go even up to 1 TeV using the target distribution tails

Custom loss (up to 800 GeV)



Custom loss + up to 1 TeV



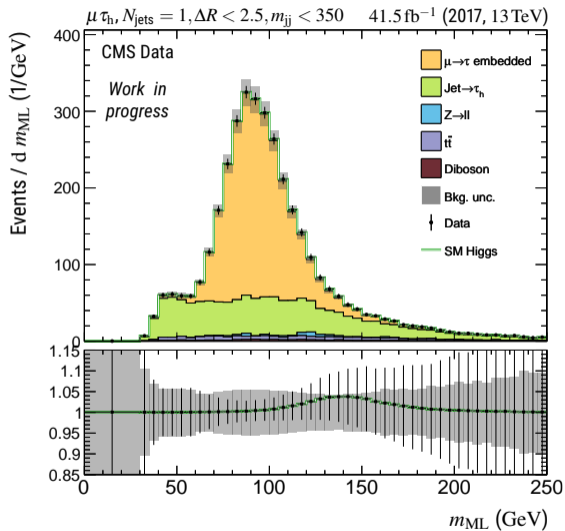
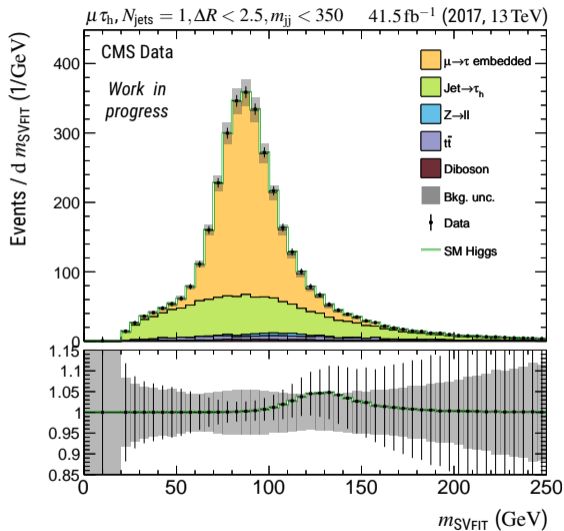
1 Few words on di- τ events

2 Building the ML model

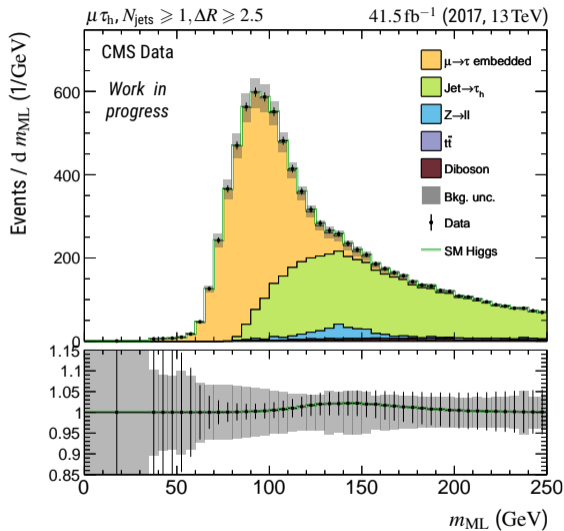
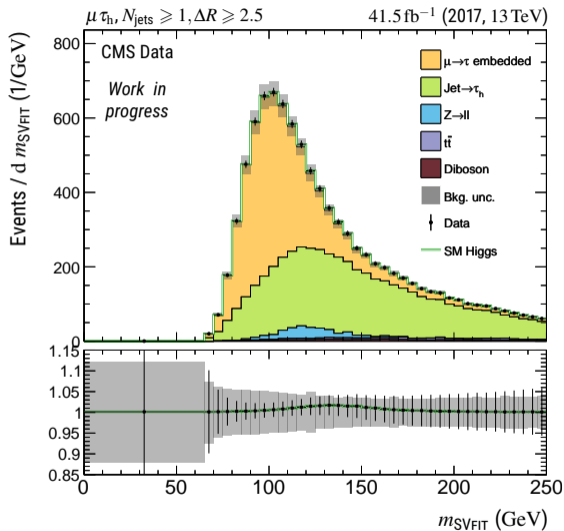
- Lucas TORTEROTOT, Ece AŞILAR, Colin BERNET (IP2I Lyon), Sebastian WOZNIEWSKI (KIT)

3 Comparison against SVFIT

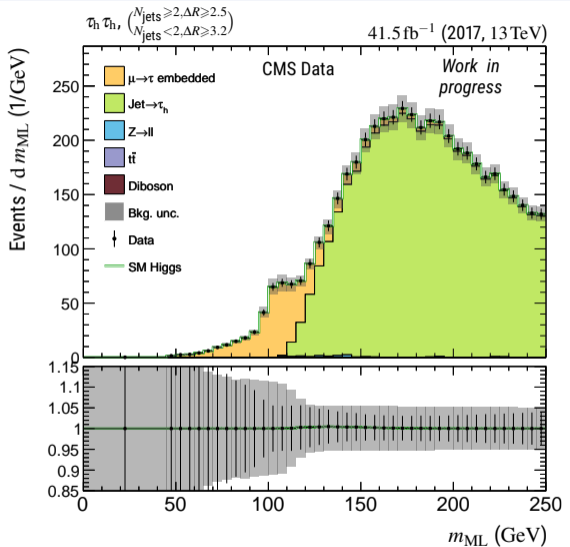
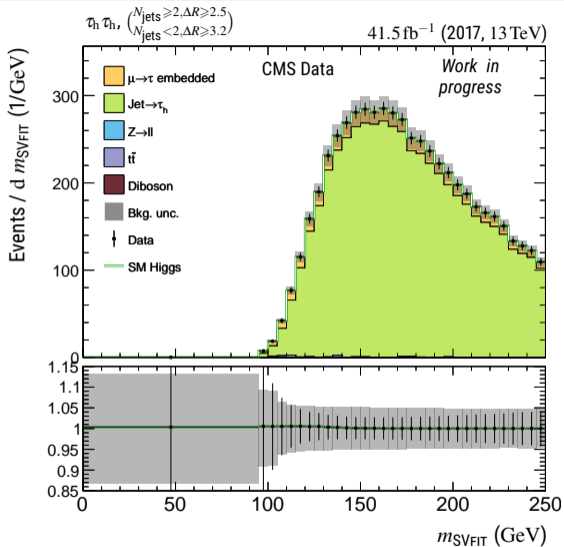
- Lucas TORTEROTOT & the $H \rightarrow \tau\tau$ MSSM analysis group



▶ **Similar SM Higgs signal sensitivity**, small (expected) overestimation from our model.



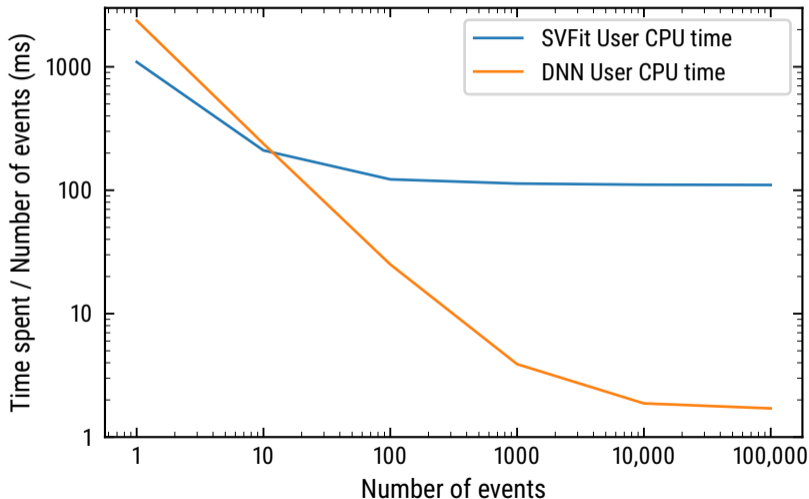
▶ **Better DY estimations** (peak at 100 GeV for m_{SVFIT} , 92 GeV for m_{ML}) and fakes at higher masses!



► Our model finds $Z \rightarrow \tau\tau$ events when SVFIT does not!

Computing time: DNN (Python) is $\sim 60\times$ faster than SVFIT (C++)!

- ▶ SVFIT:
 - ▷ fit to find the best mass
 - ▷ for each event
- ▶ DNN:
 - ▷ fit done once (training)
 - ▷ apply the DNN formula



Conclusion

- ▶ Successful m_H reconstruction in di- τ events!
 - ▷ Not only $H \rightarrow \tau\tau$ but any $X \rightarrow \tau\tau$ analysis could benefit!
- ▶ SVFIT comparison:
 - ▷ Similar Higgs sensitivity!
 - ▷ Better Z estimation observed (the model has been trained on $h \rightarrow \tau\tau$ with various masses only);
 - ▷ Faster (about 60 times)!

If you want to go deeper in:

- ▶ The DNN GitHub repository:

https://github.com/lucastorterotot/DL_for_HTTP_mass

- ▶ The training events generation:

https://github.com/easilar/cmssw/blob/from-CMSSW_10_2_22/README

- ▶ The DNN/SVFIT derivation scripts used for the benchmark:

DNN: shorturl.at/gmsN8

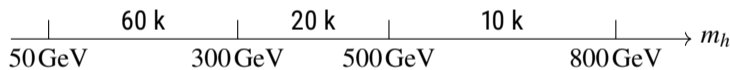
SVFIT: shorturl.at/agyN0

Thanks for your attention!

`l.torterotot@ipnl.in2p3.fr`

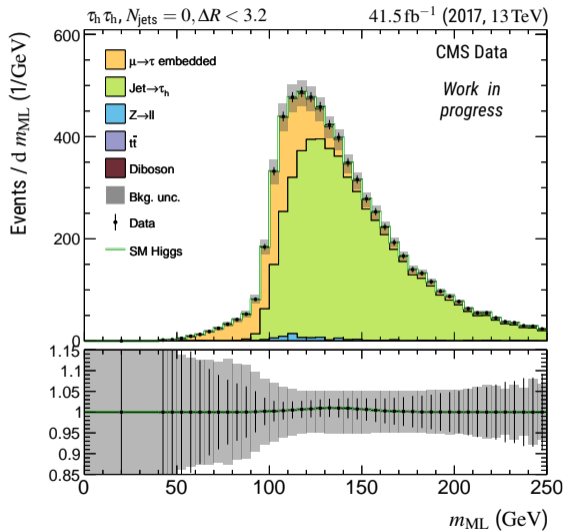
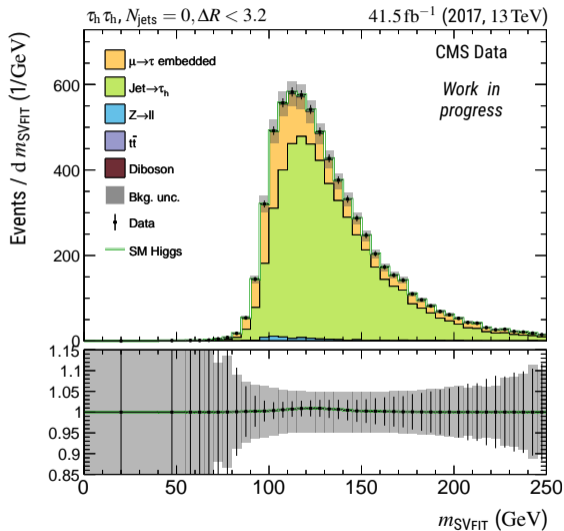
Obtaining datasets

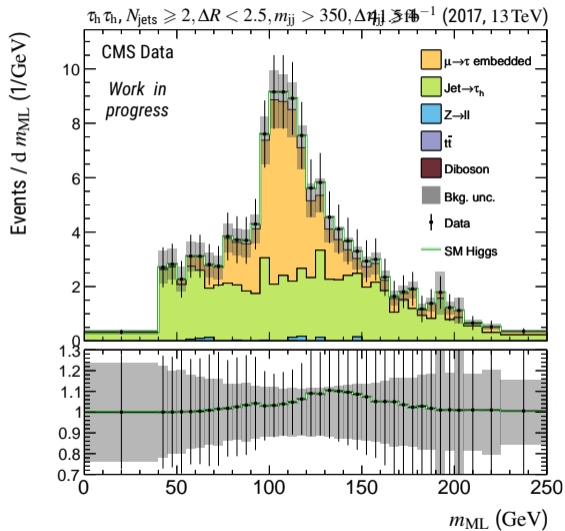
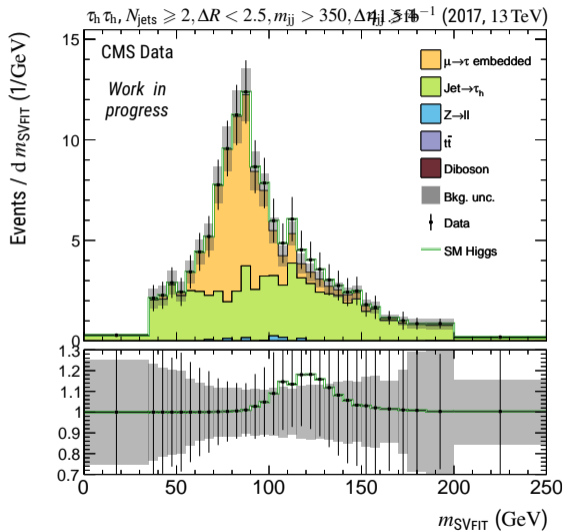
- ▶ Trained on SM $h \rightarrow \tau\tau$ with $m_h \in \llbracket 50, 800 \rrbracket \text{GeV}$.
 - ▷ We generated events with FASTSIM, see Github repository [here](#).
 - ▷ Pile-Up added with the 2017 PU profile with MinBias events.
 - ▷ Event selection similar to the current MSSM $H \rightarrow \tau\tau$ analysis (p_T and η cuts, DEEPTAU, lepton vetoes, ...).
 - ▷ See [CMS AN 2020/218](#).
 - ▷ More events at low mass to account for the selection efficiency and have statistics:

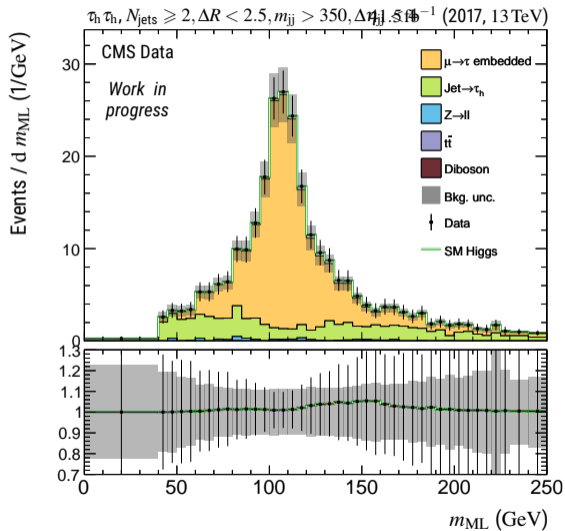
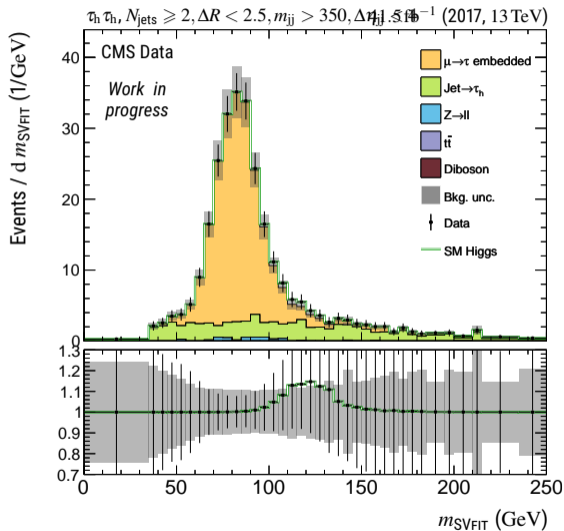


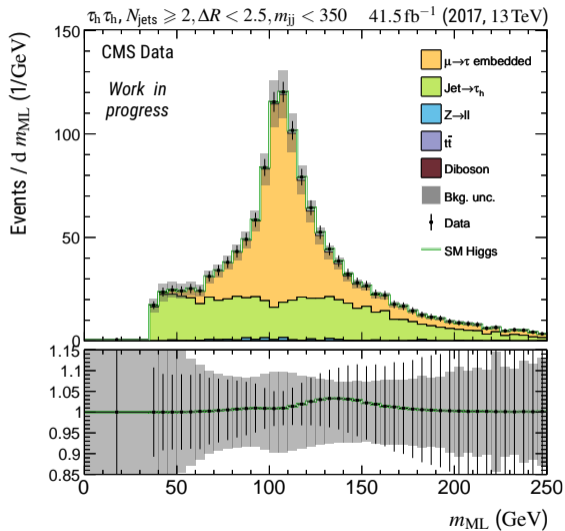
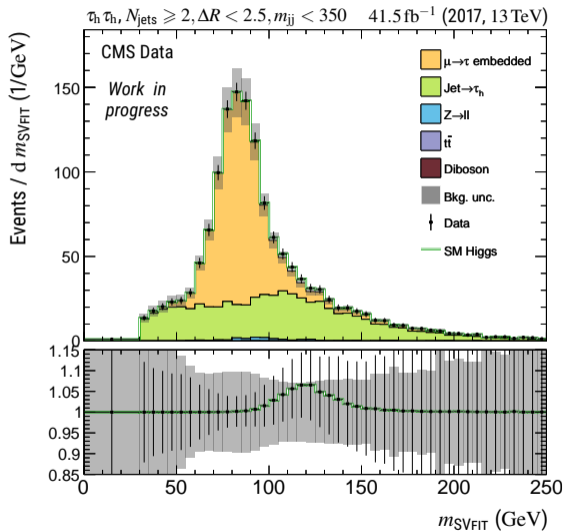
- ▶ 22 M events, 3 M (14%) selected in the mass range (due to the mass width).
- ▶ Split into 3 subsamples, use weights to flatten the target distribution:

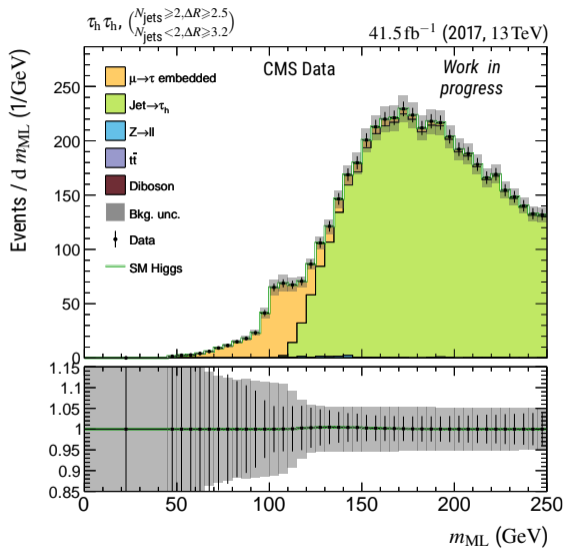
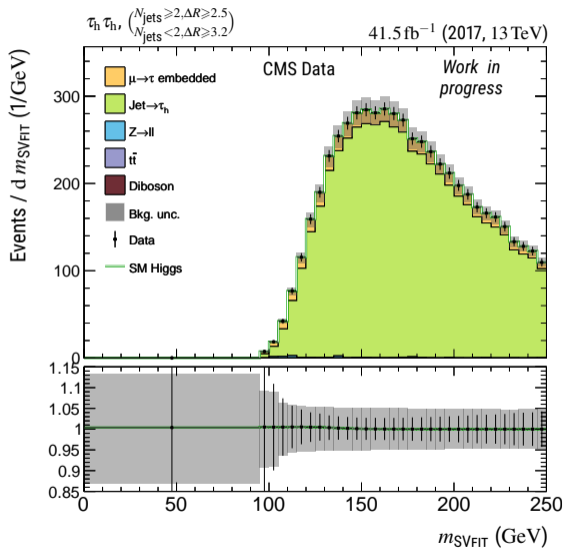
Purpose	training	validation	tests
Quantity	70%	20%	10%

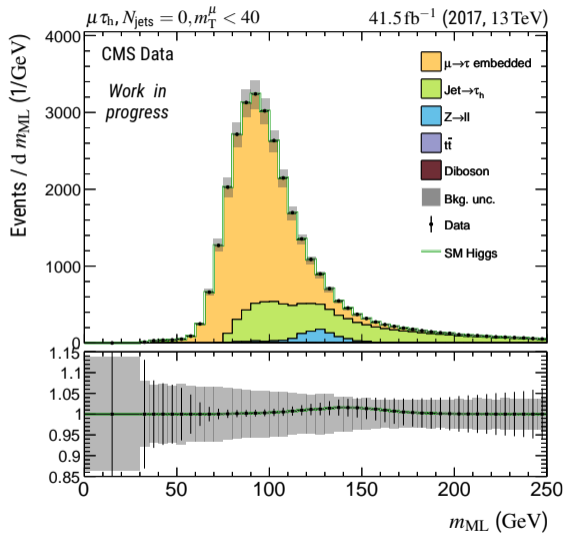
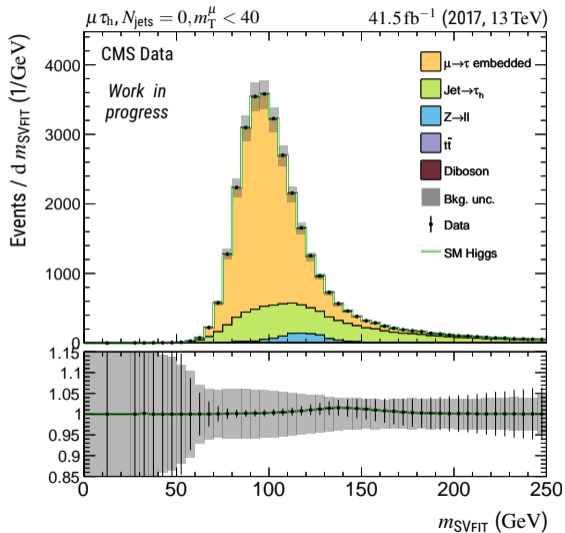


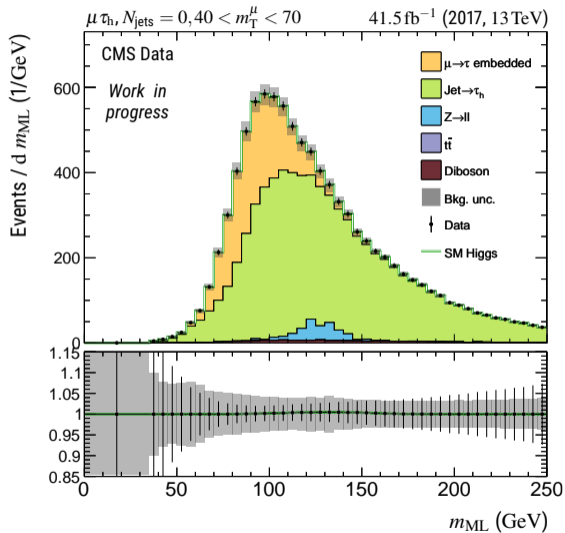
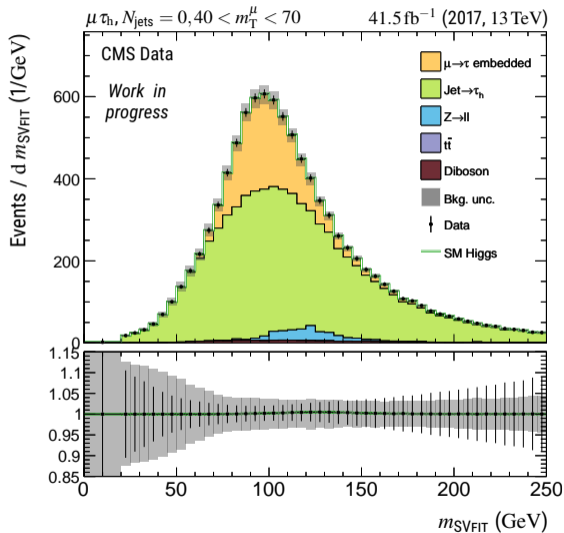


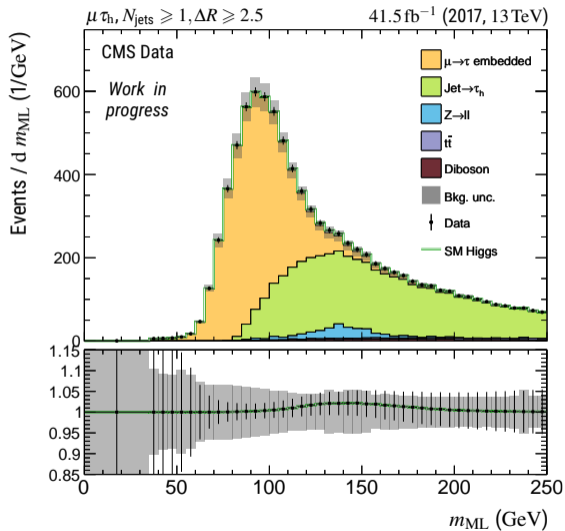
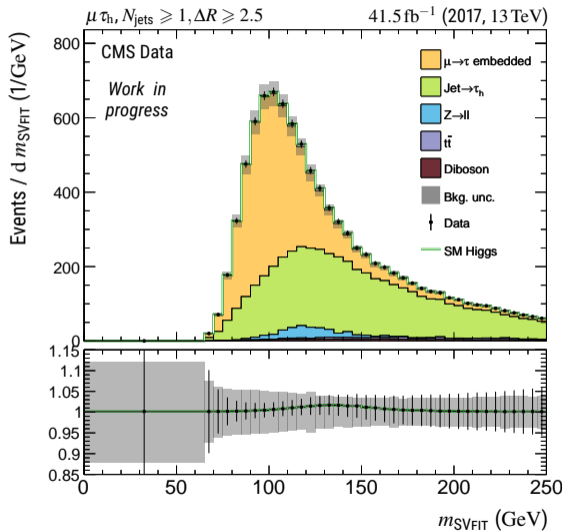


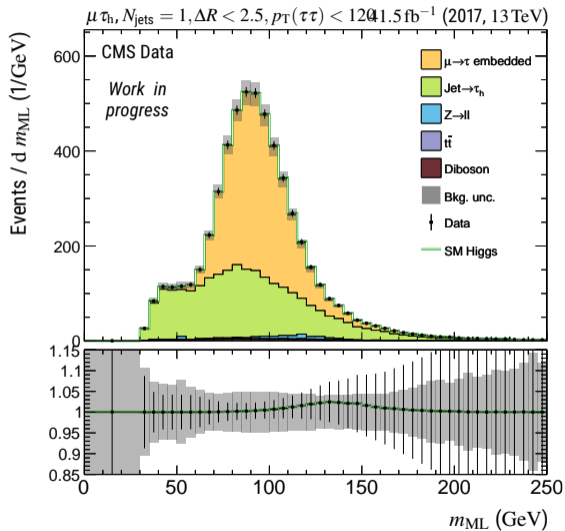
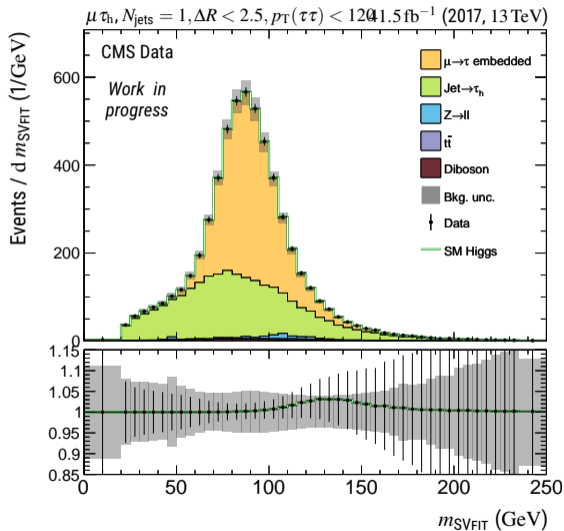


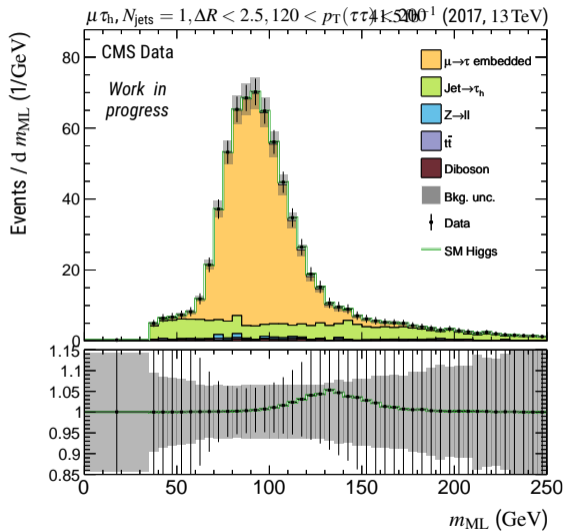
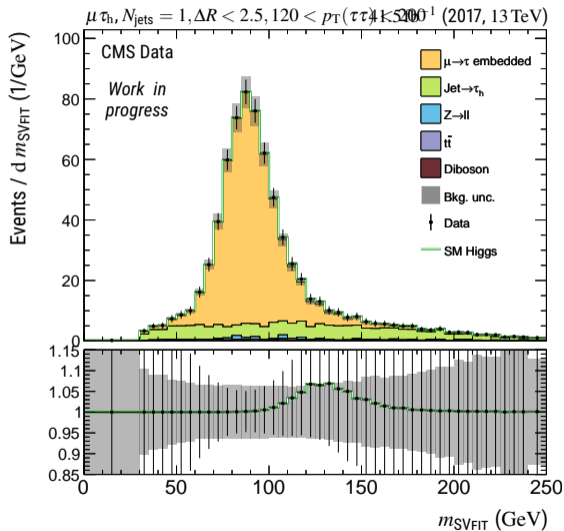


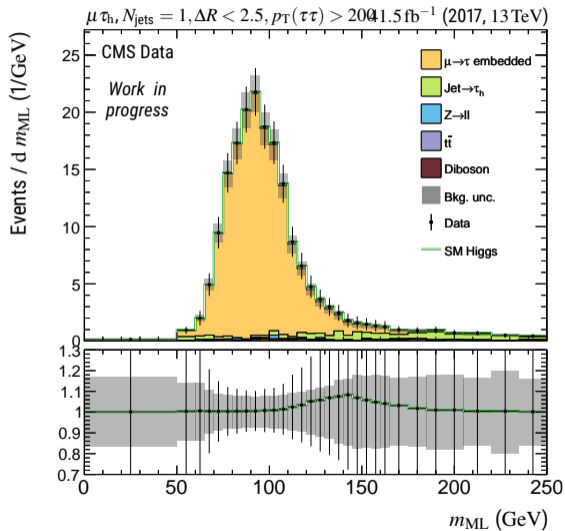
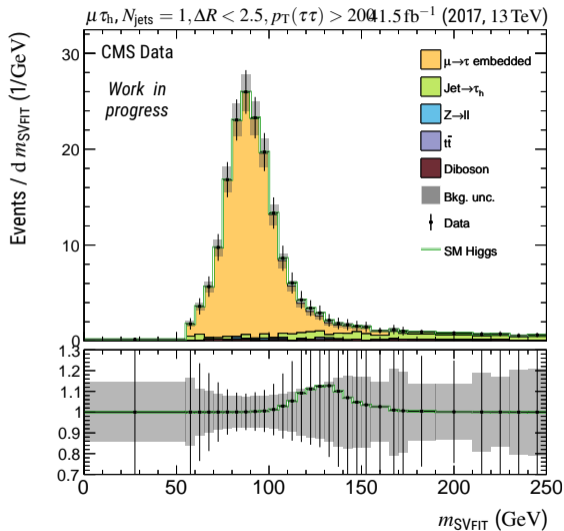


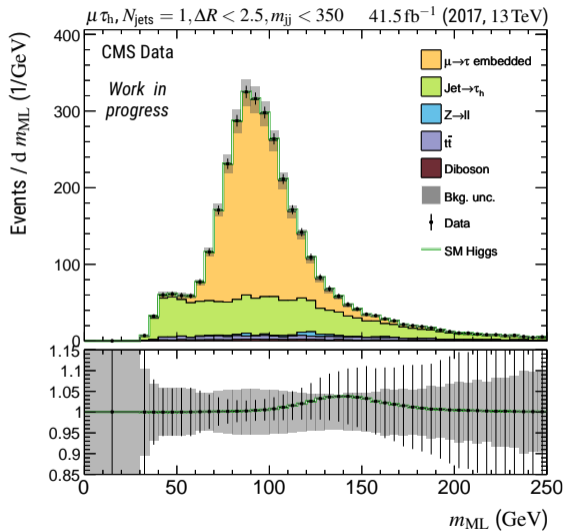
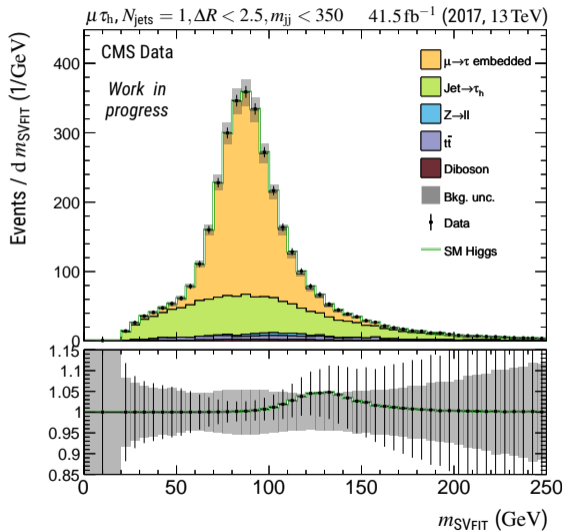


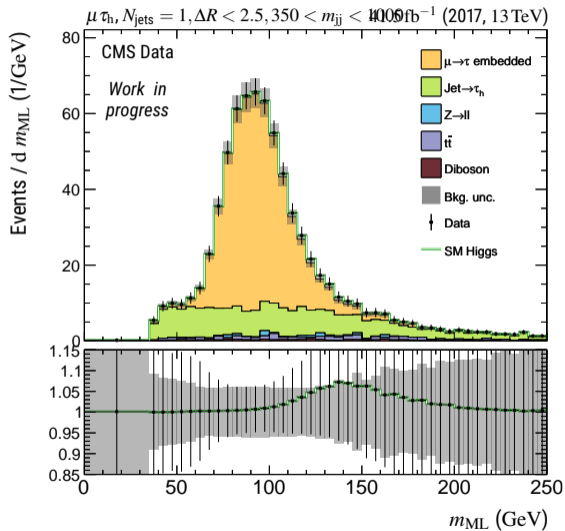
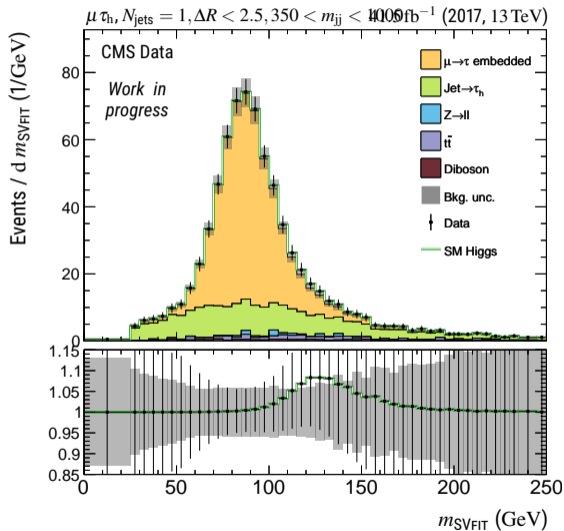


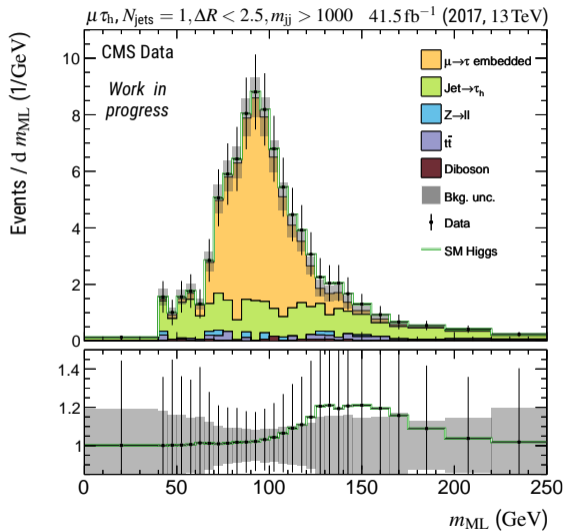
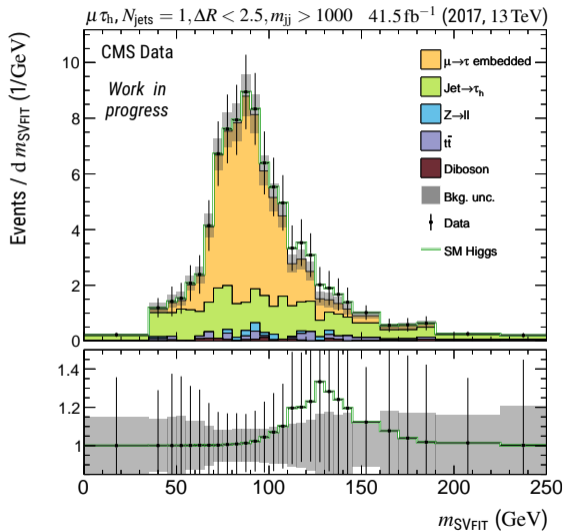


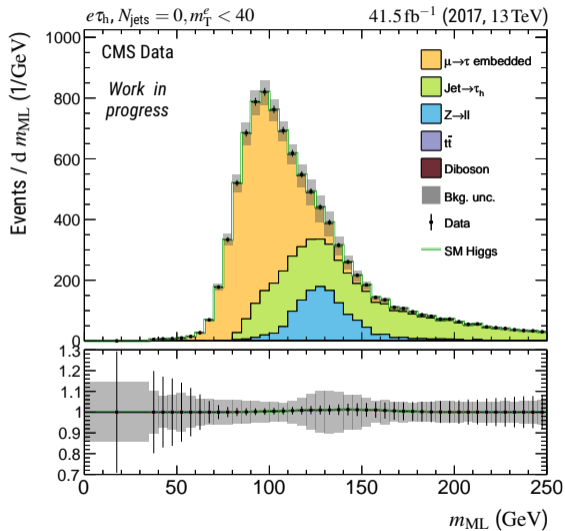
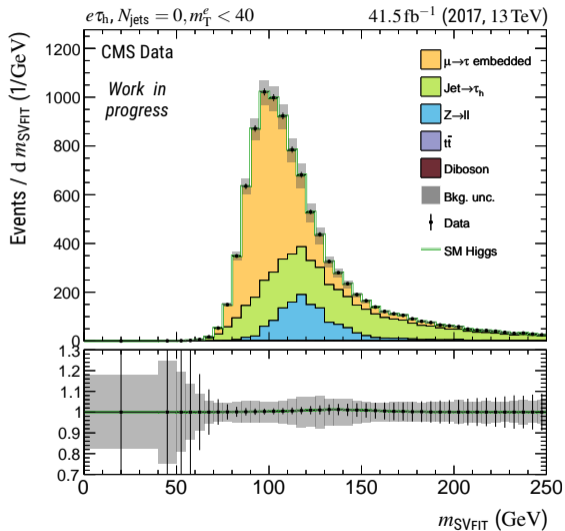


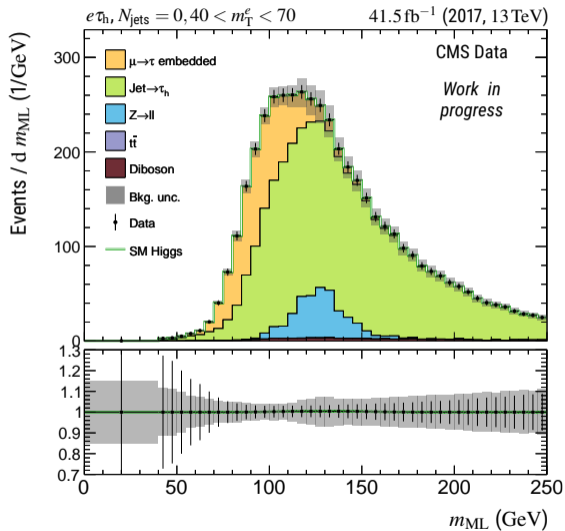
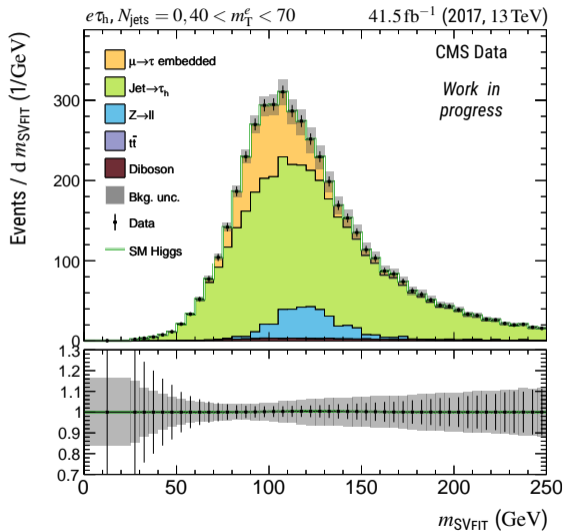


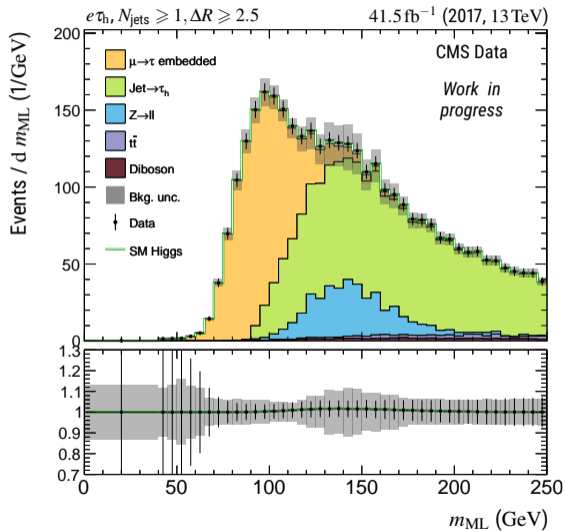
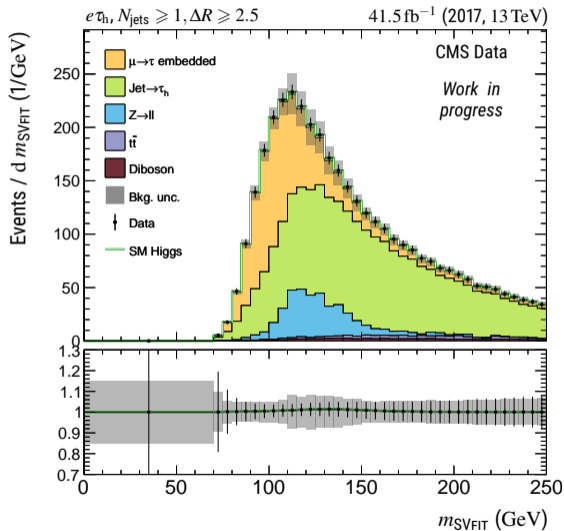


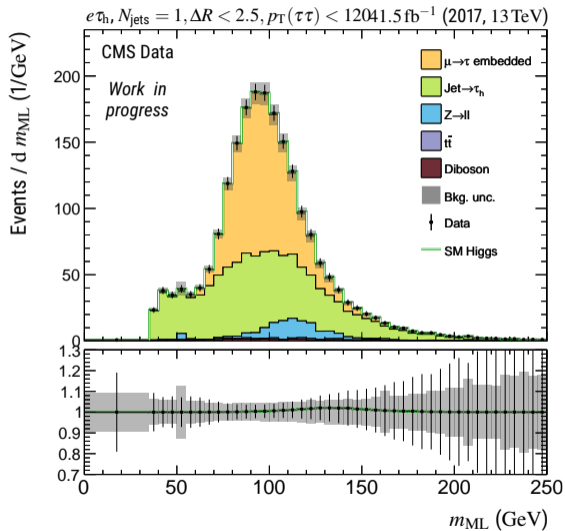
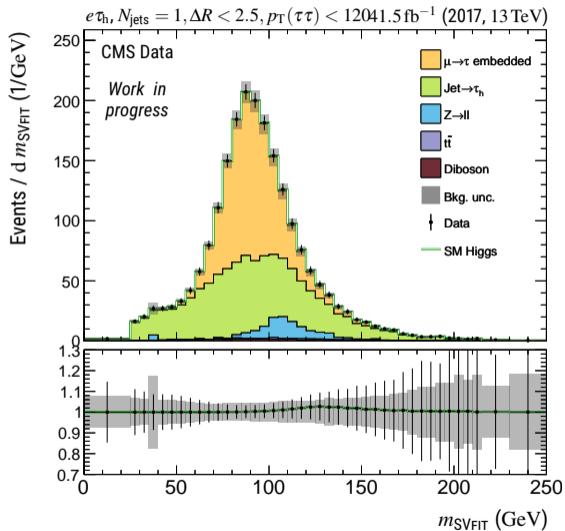


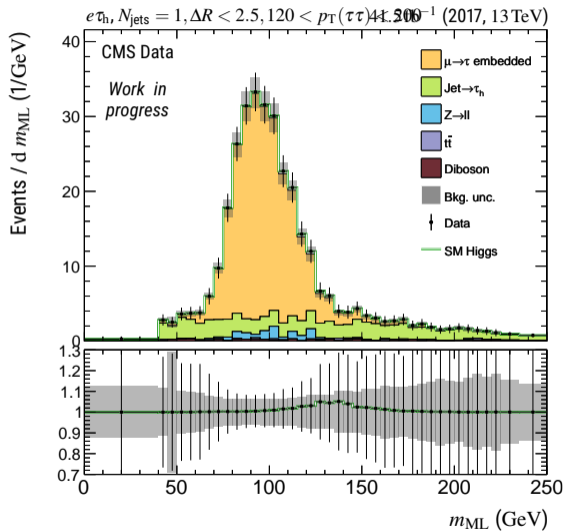
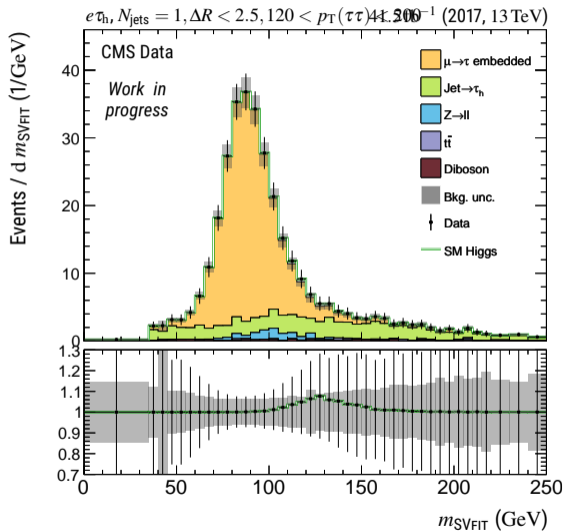


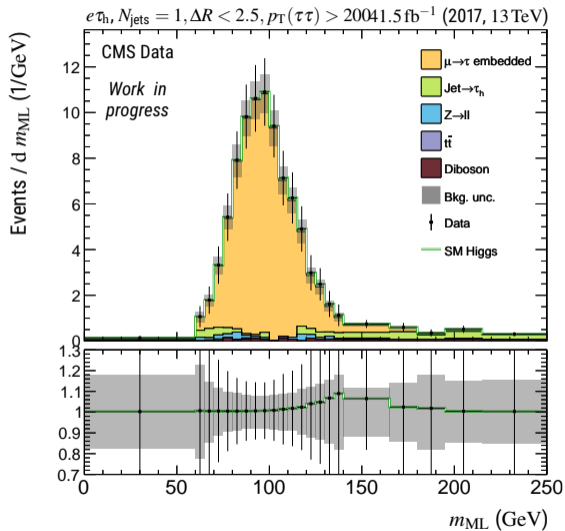
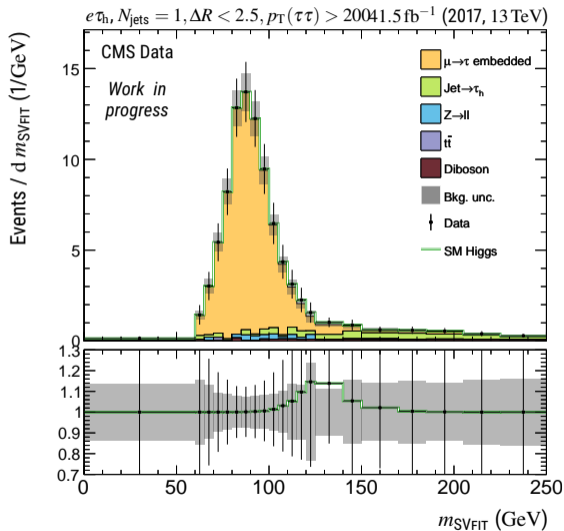


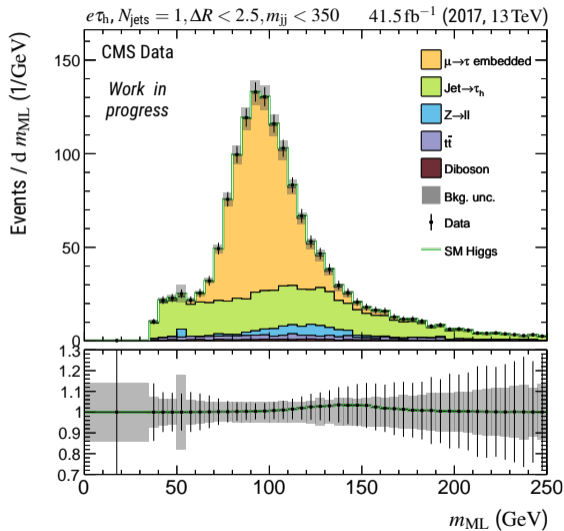
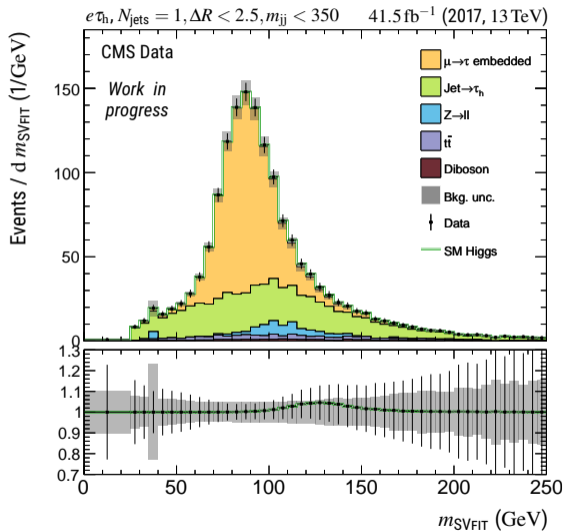


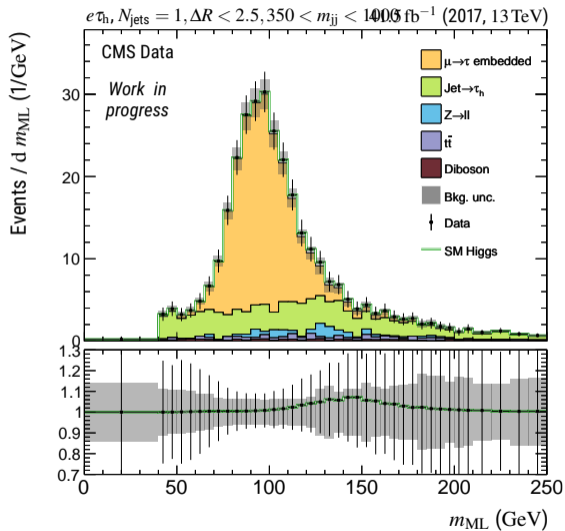
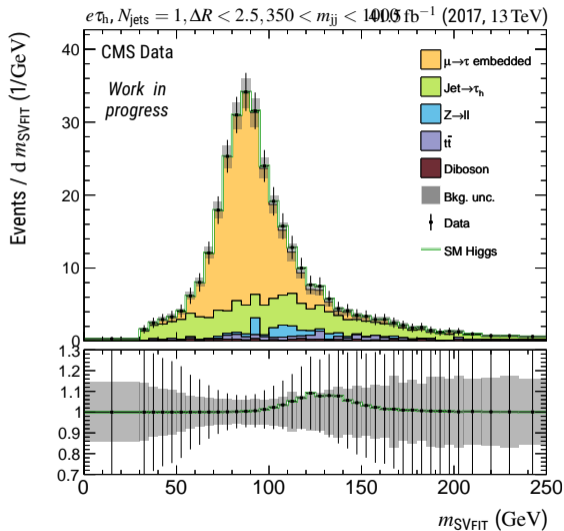


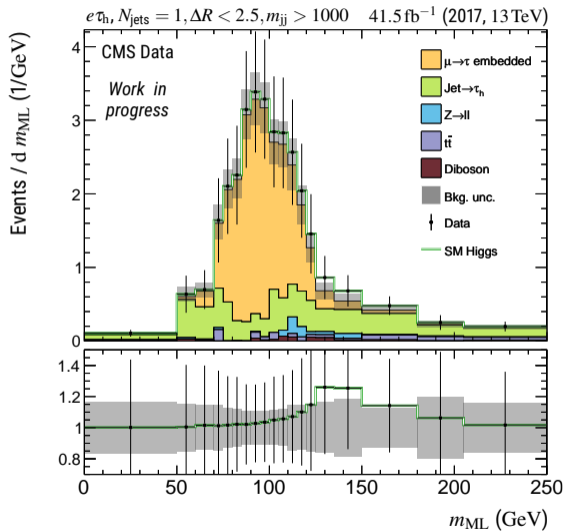
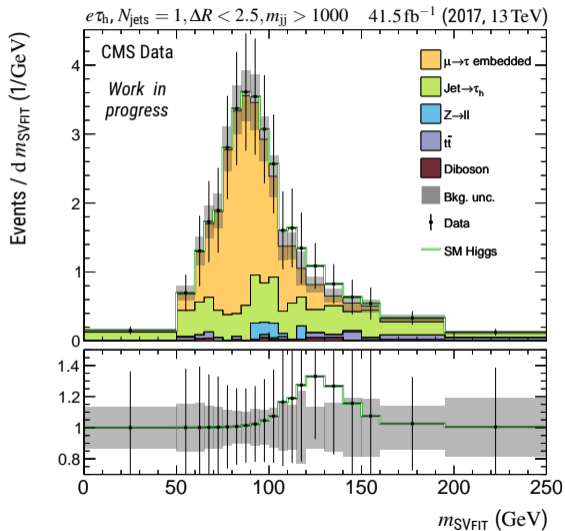


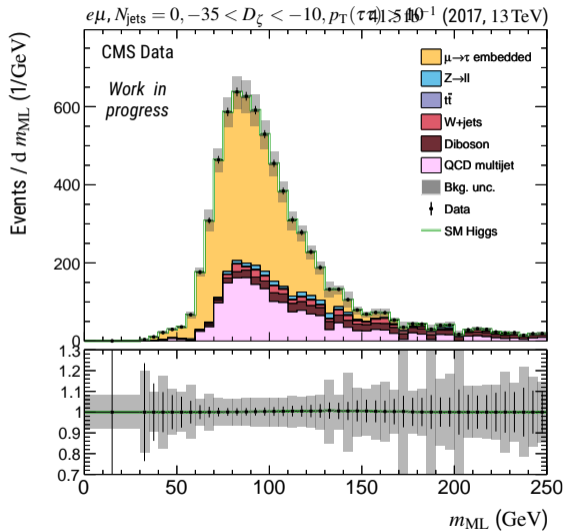
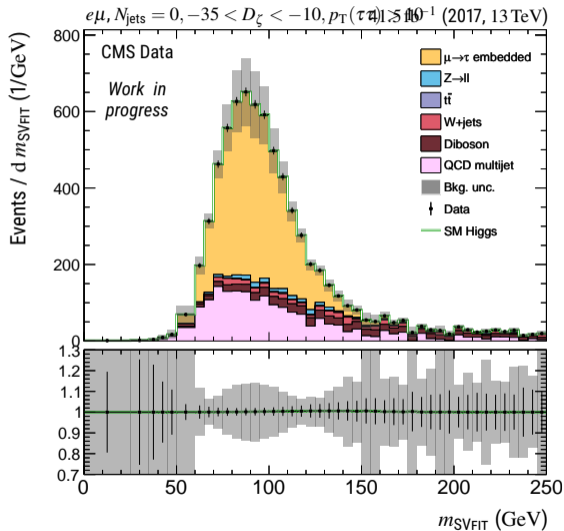


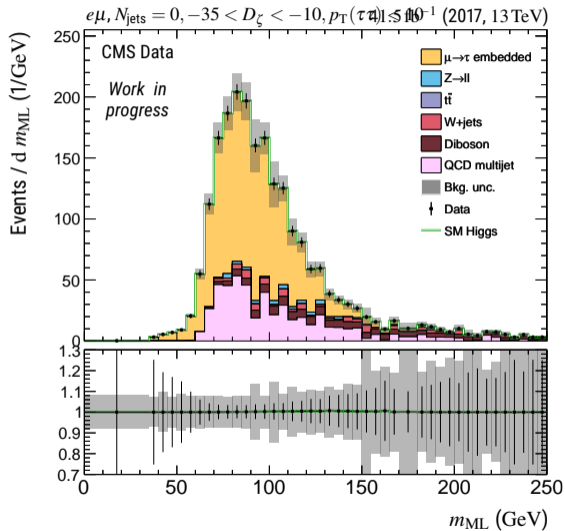
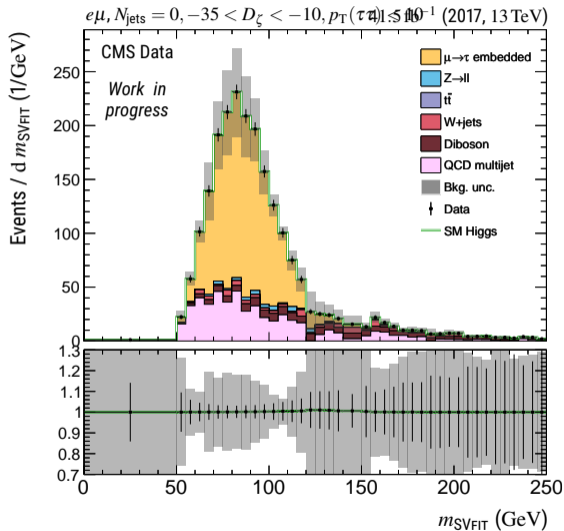


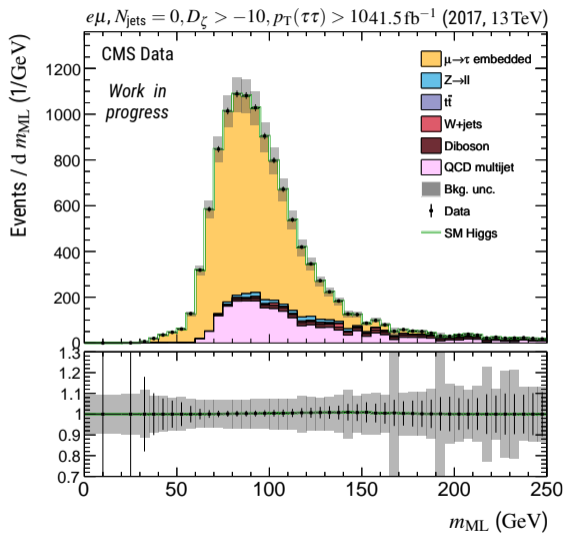
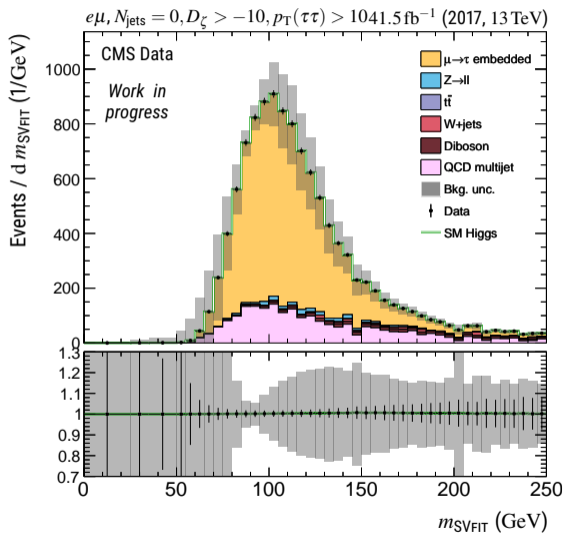


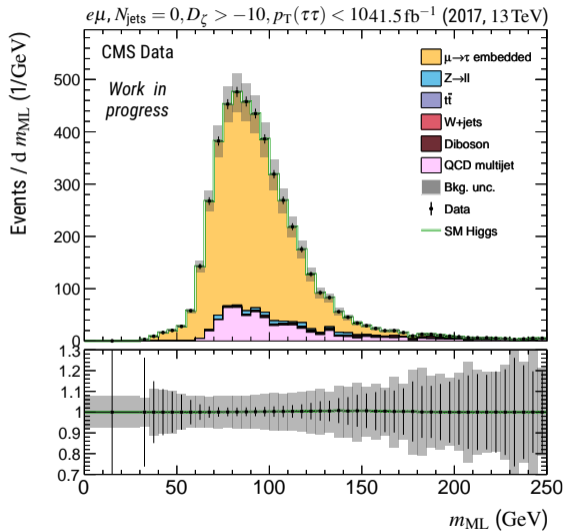
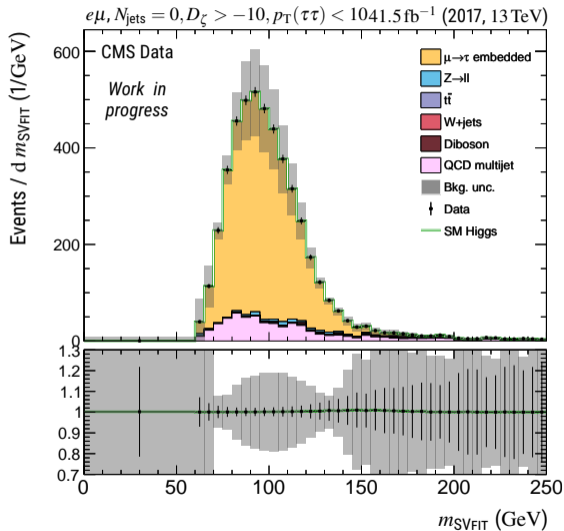


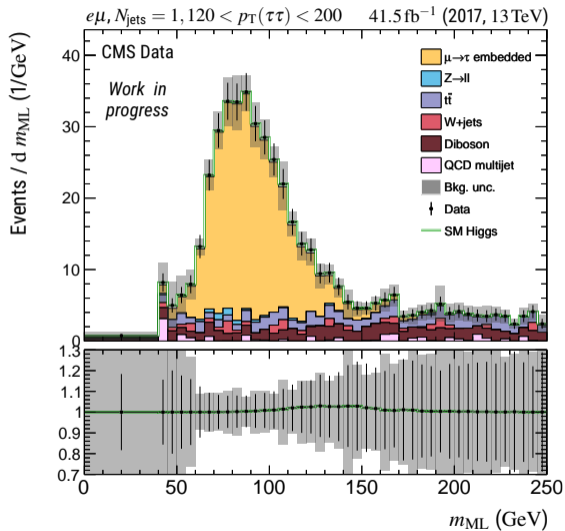
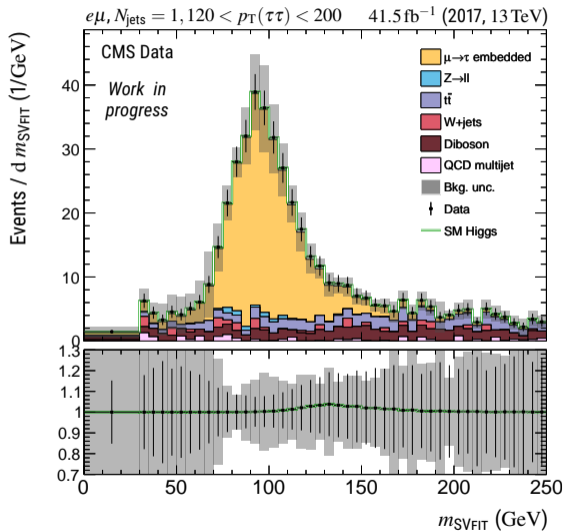


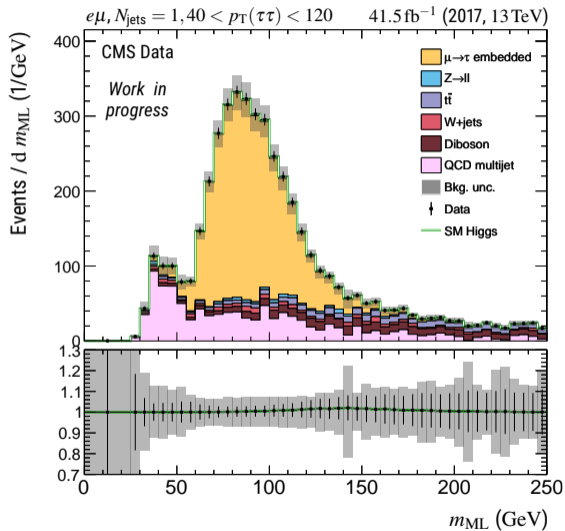
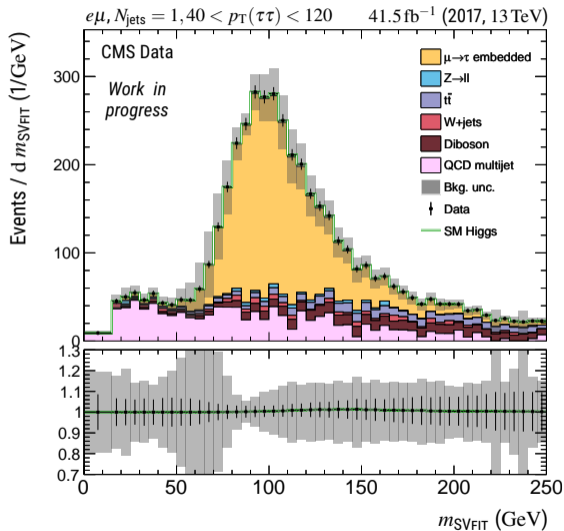


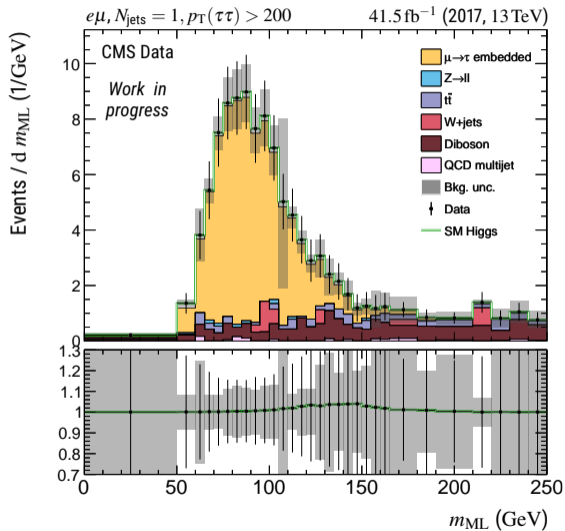
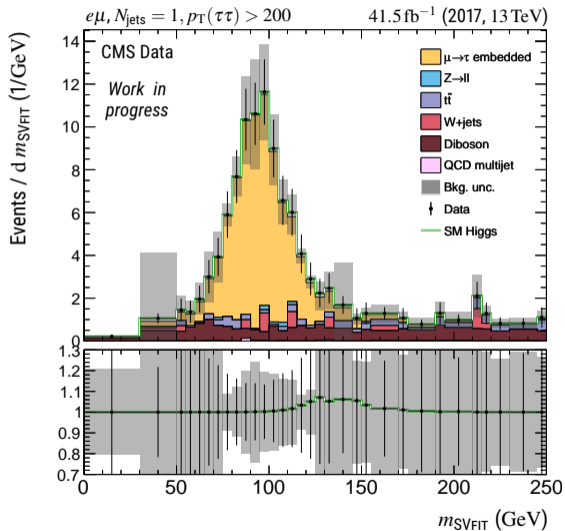


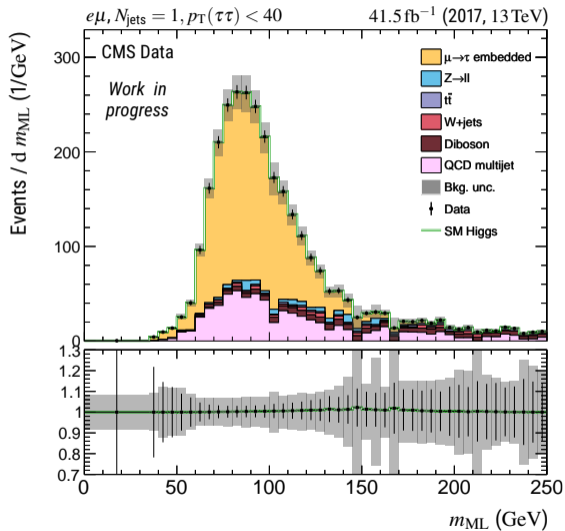
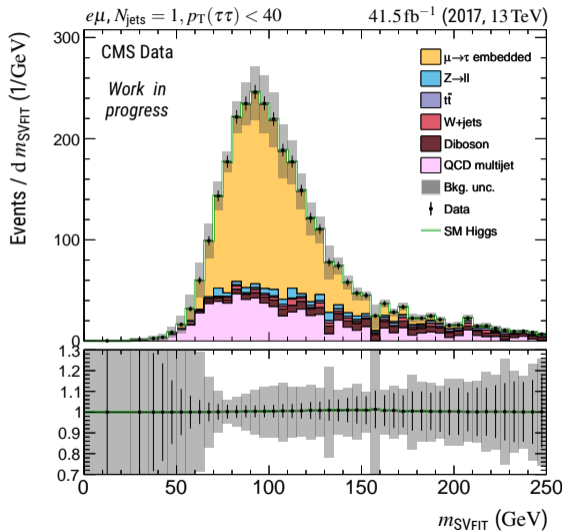


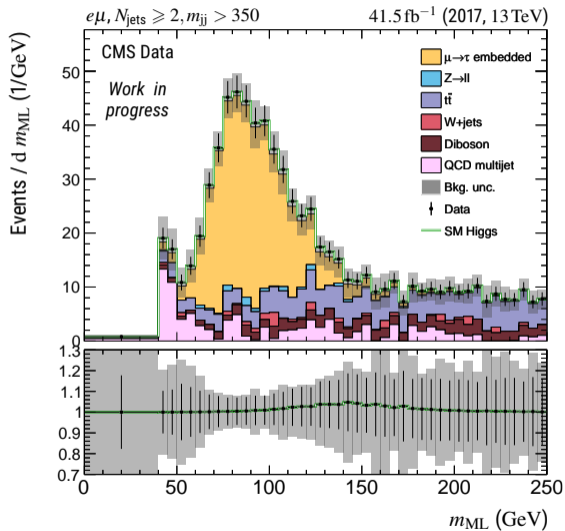
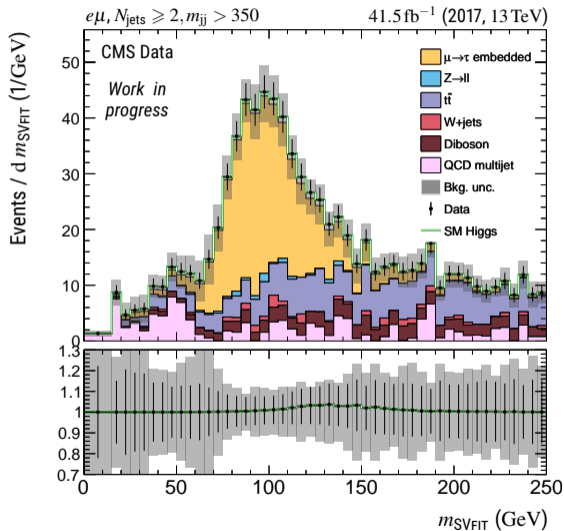


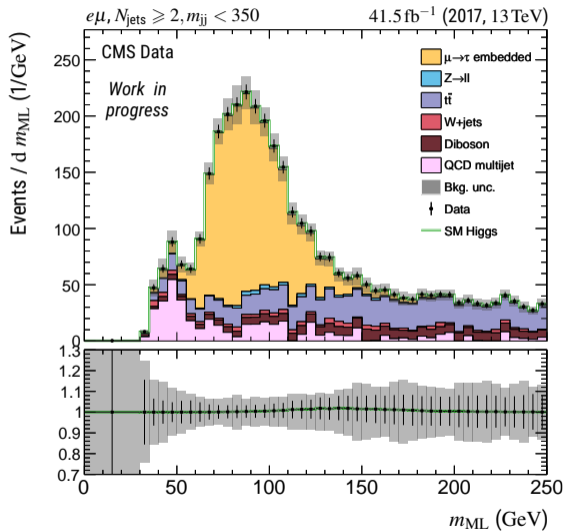
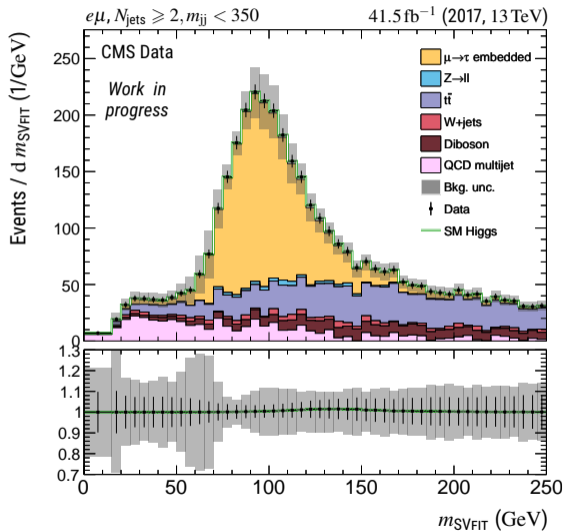


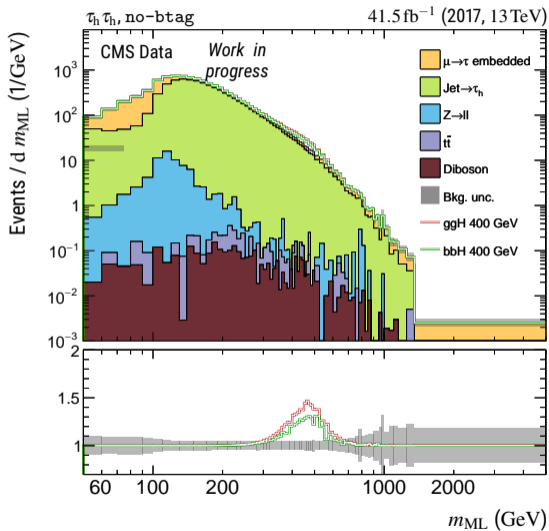
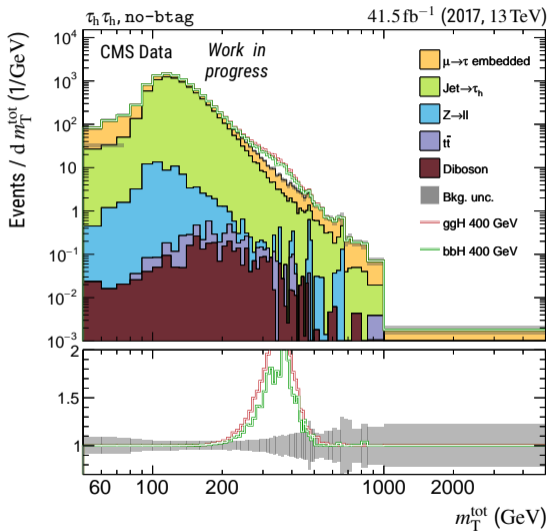


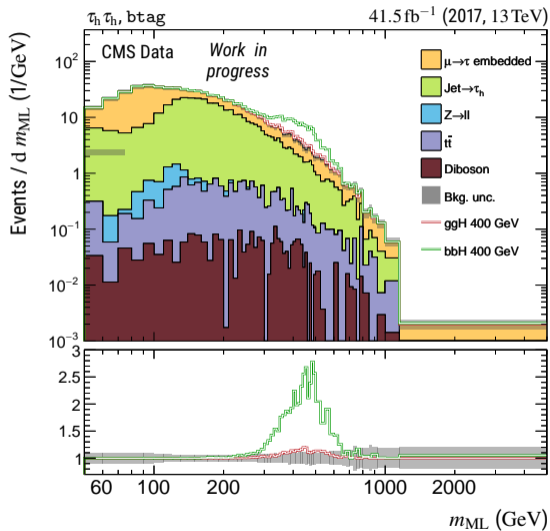
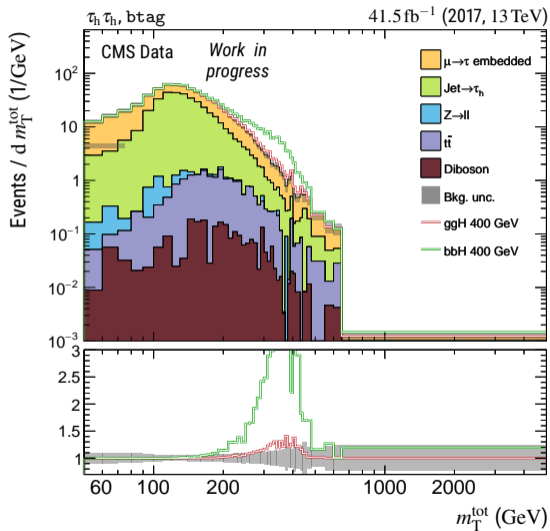


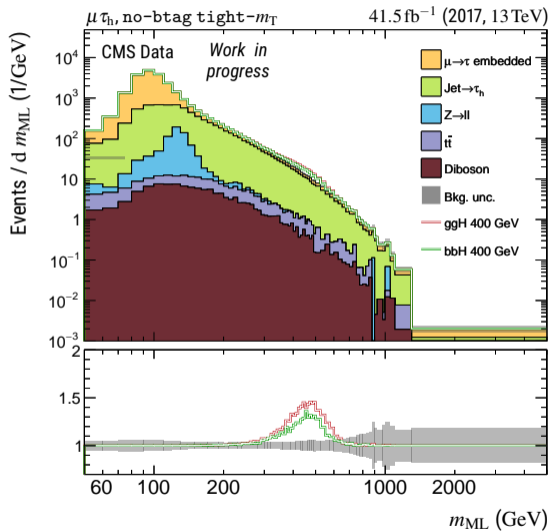
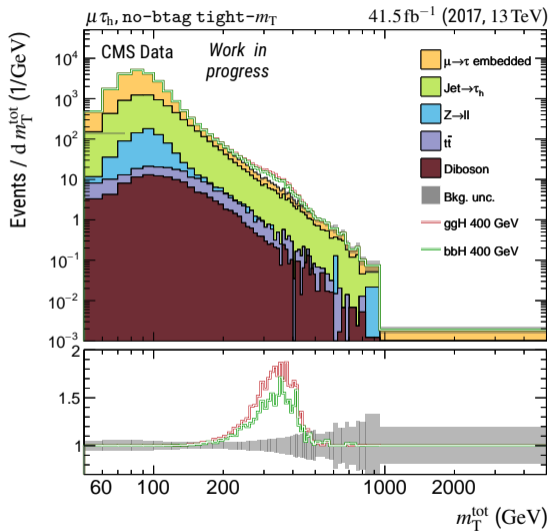


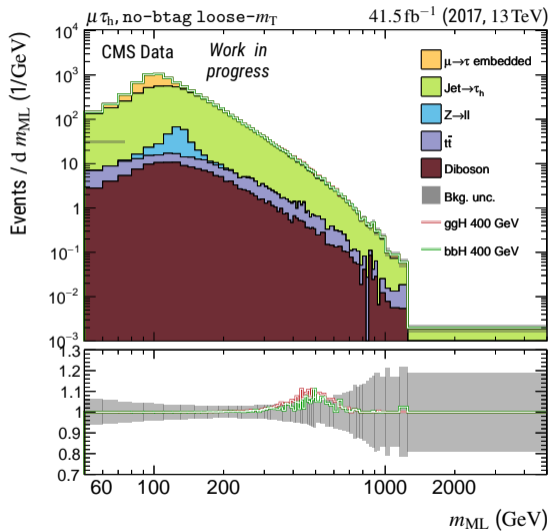
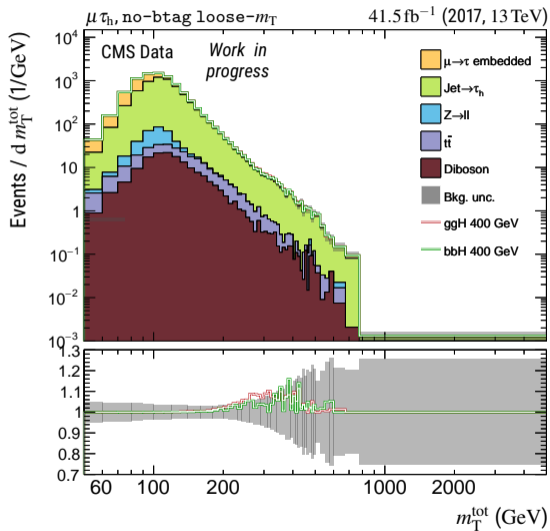


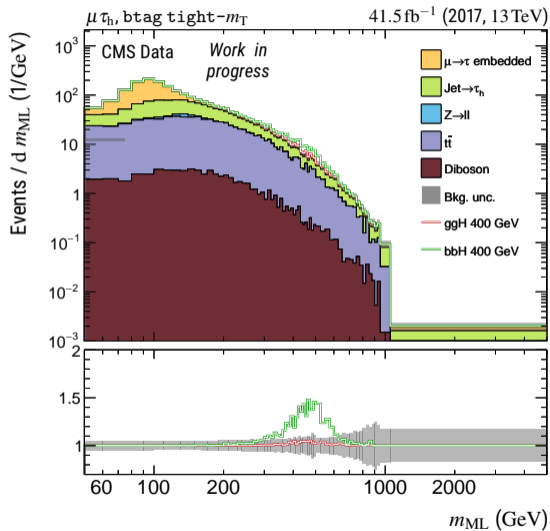
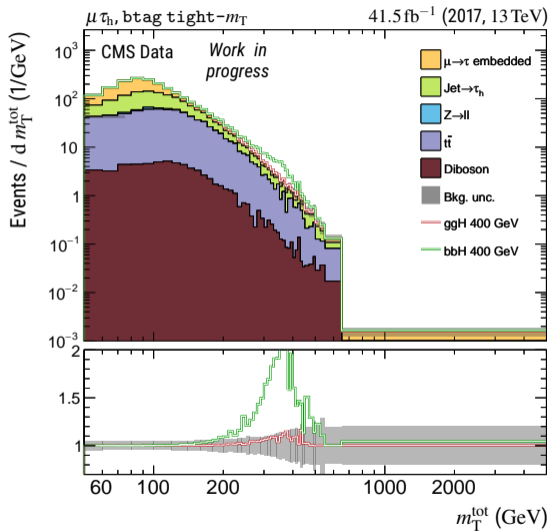


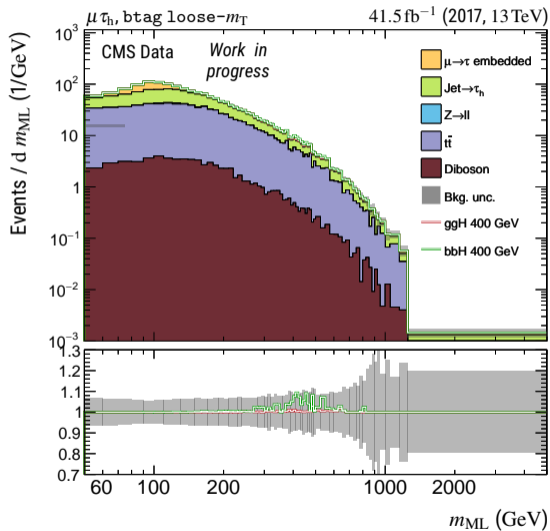
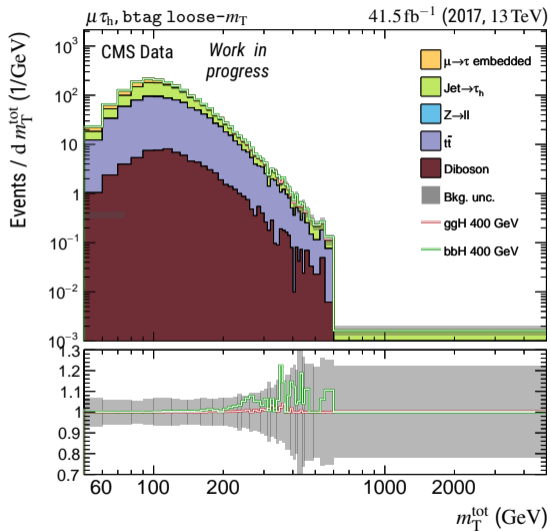


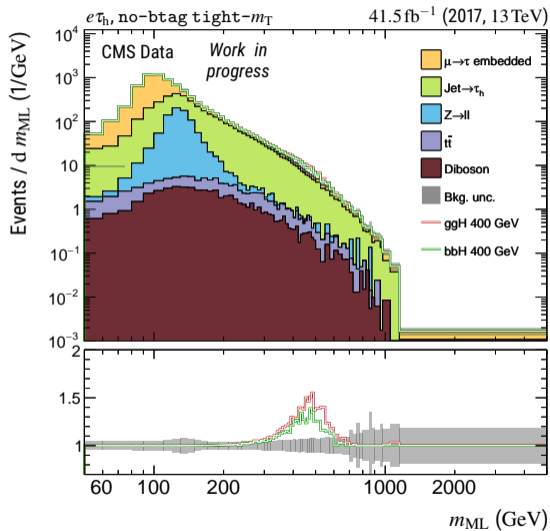
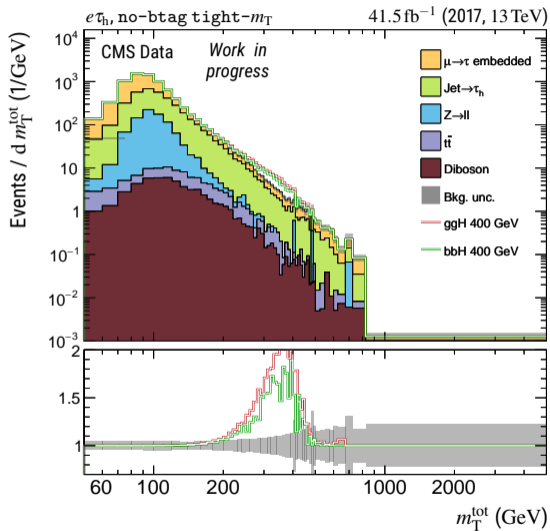


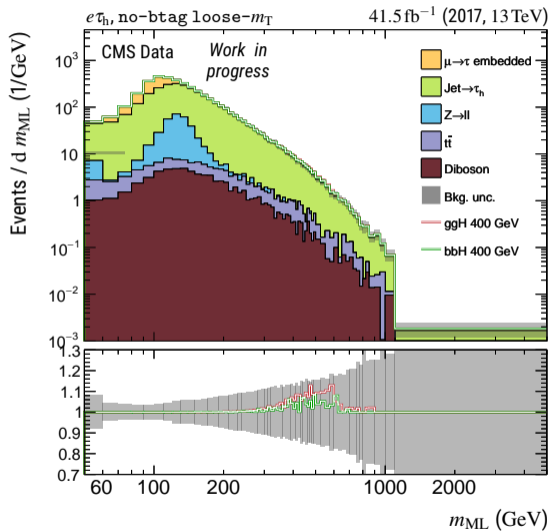
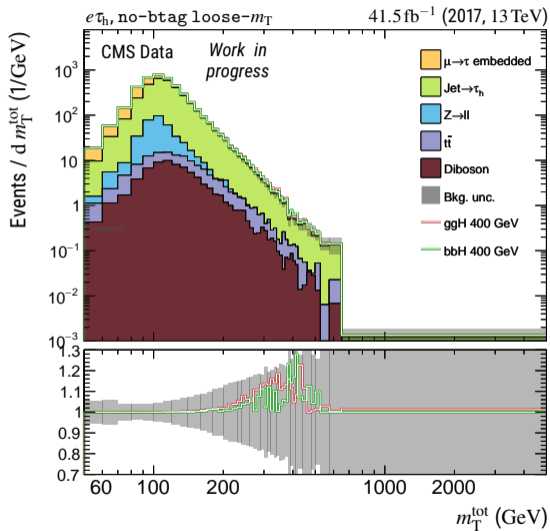


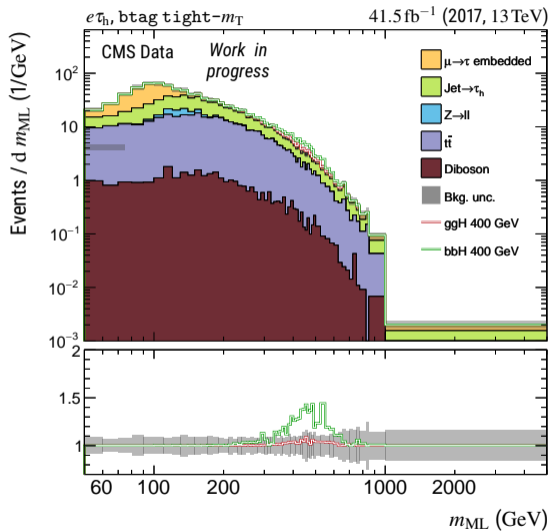
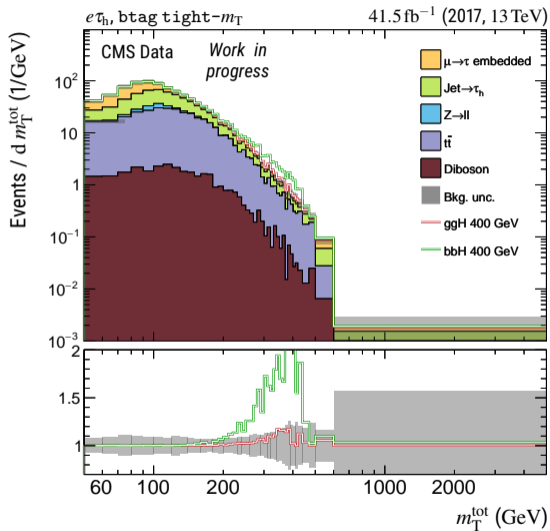


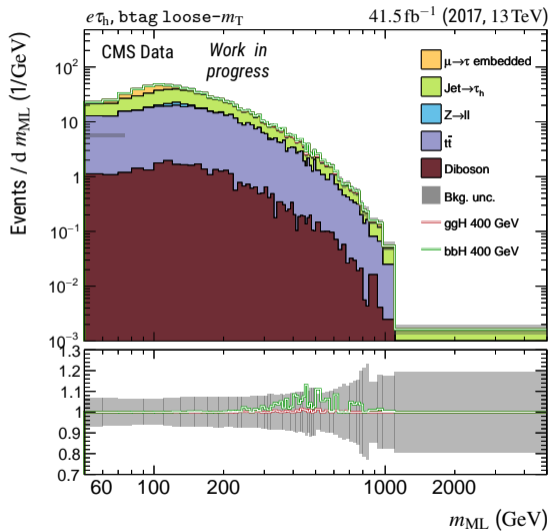
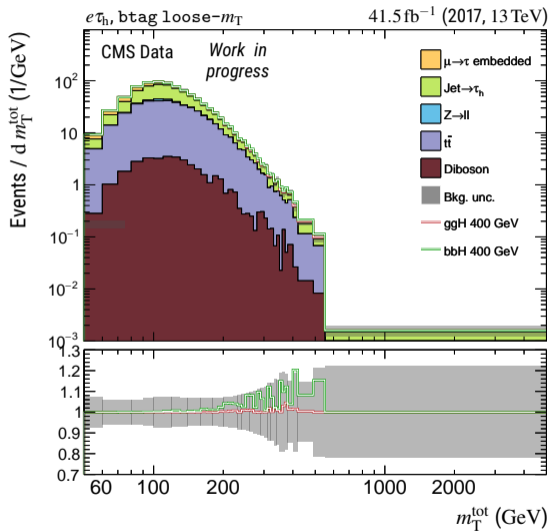


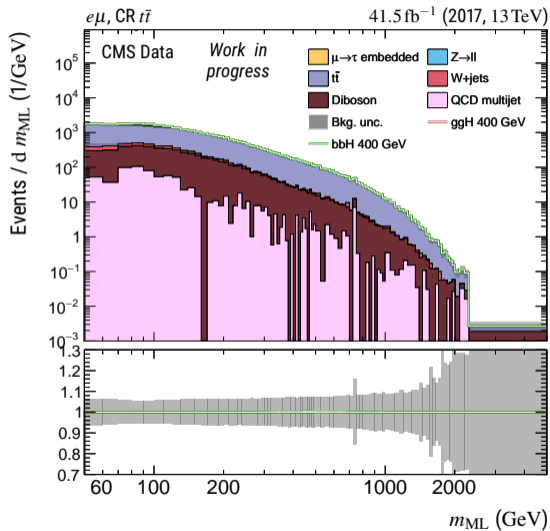
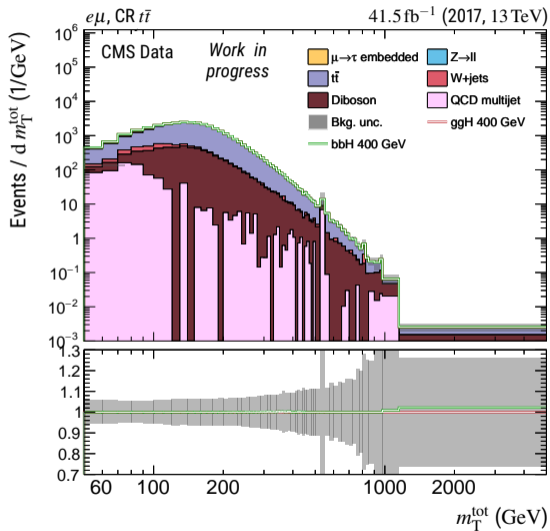


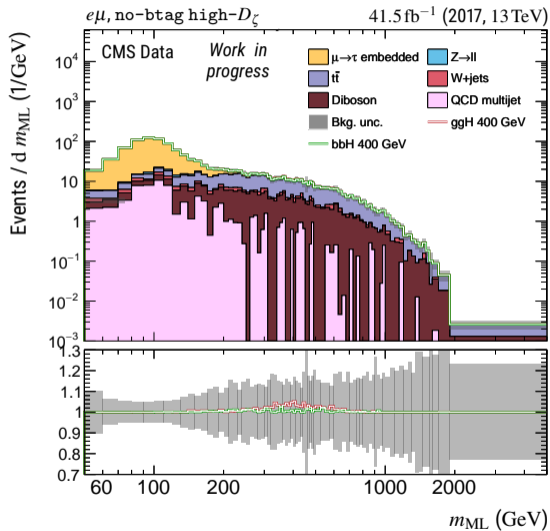
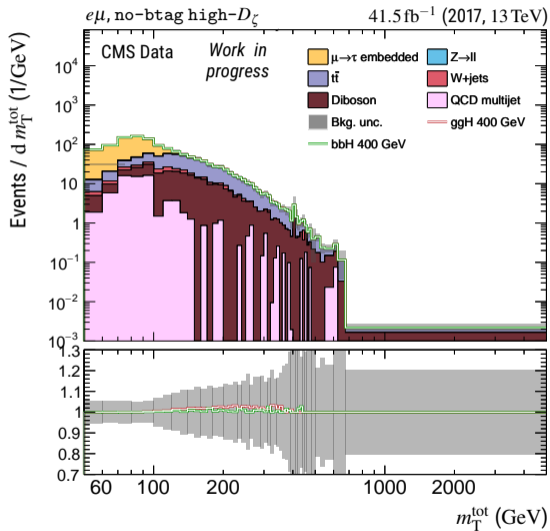


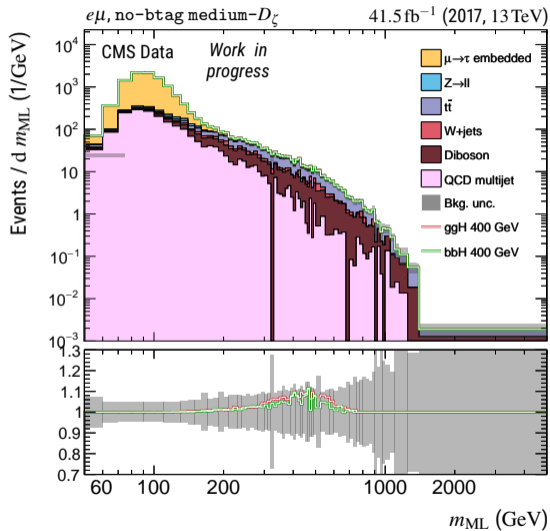
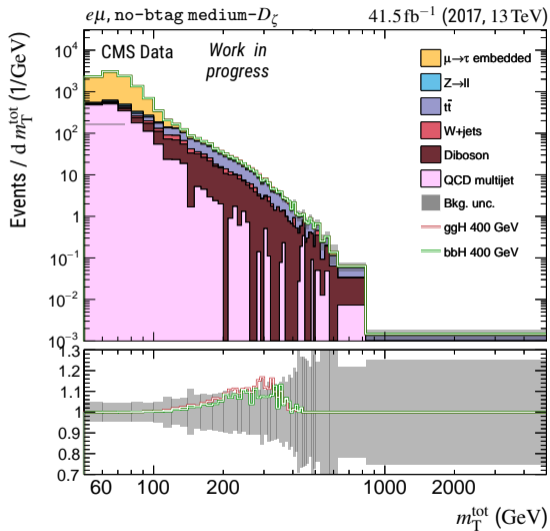


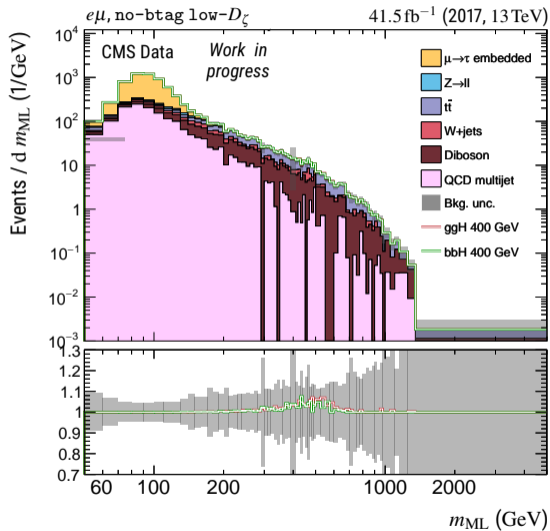
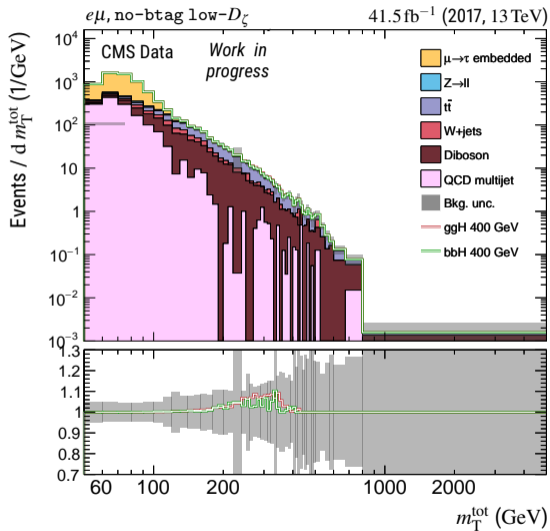


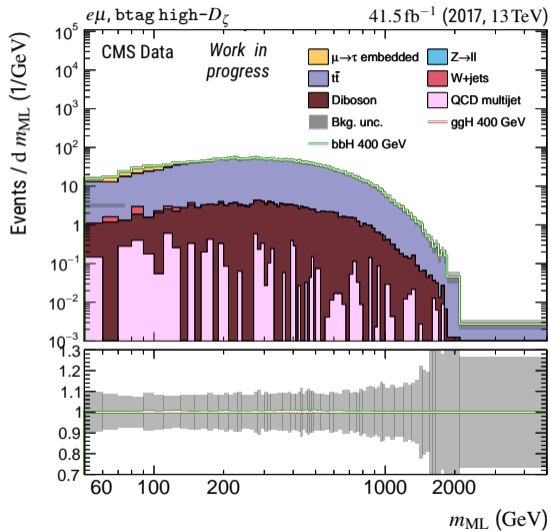
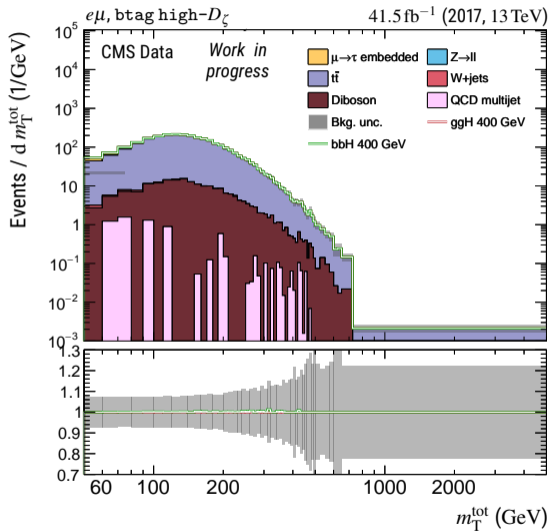


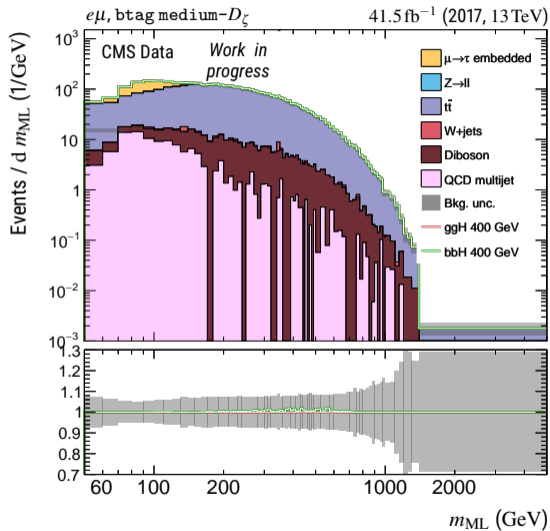
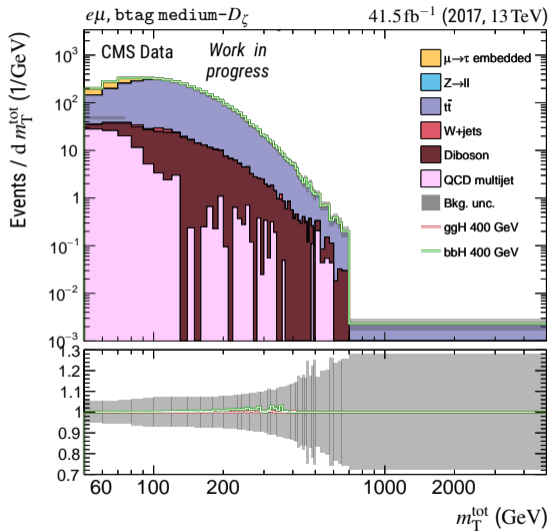


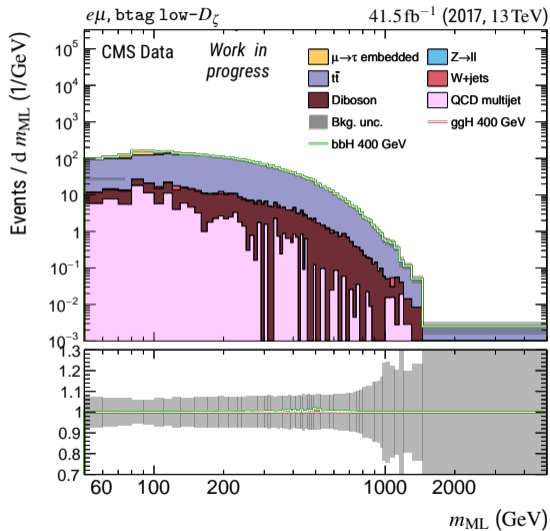
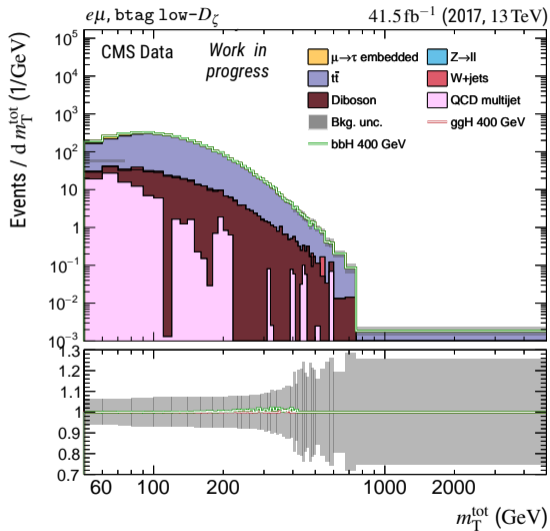


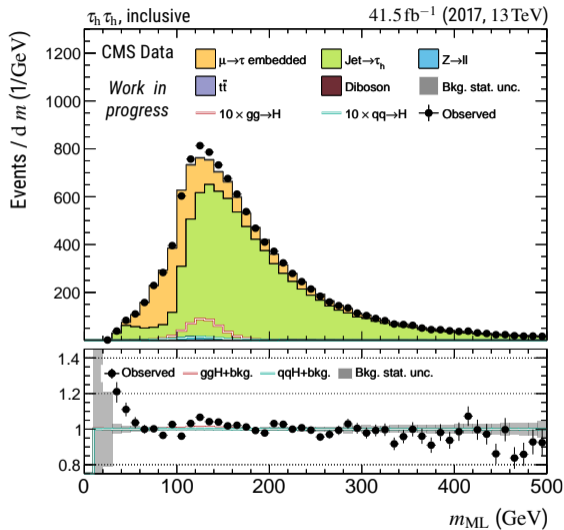
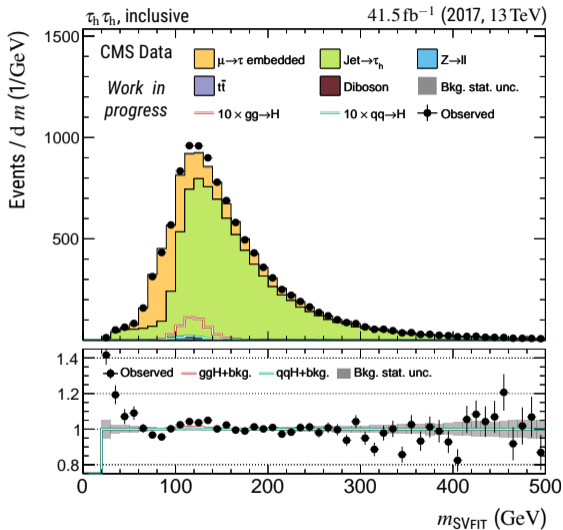




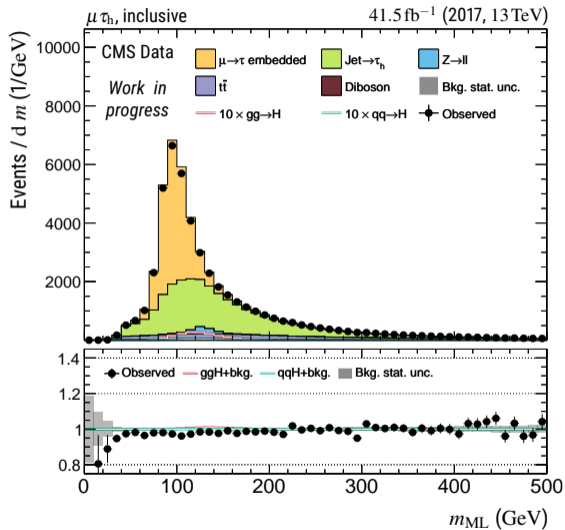
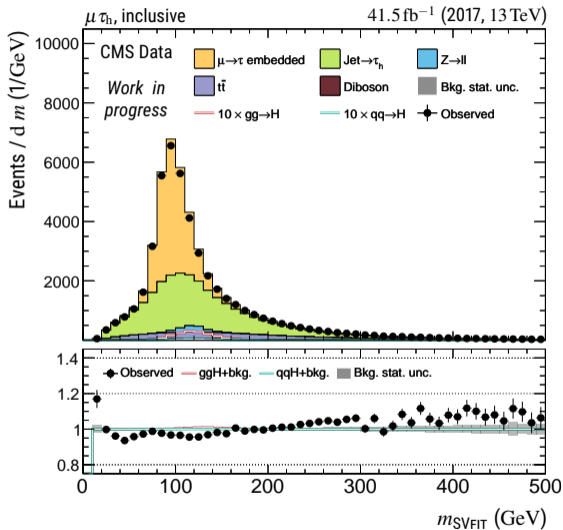




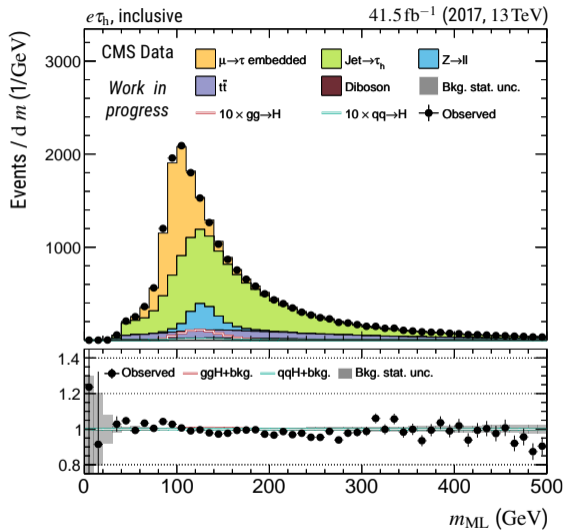
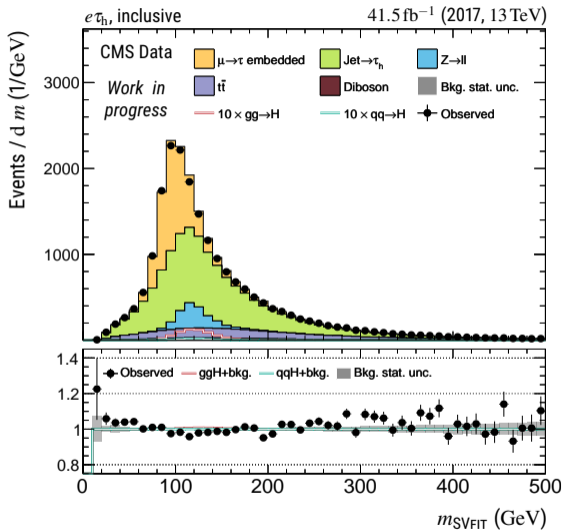




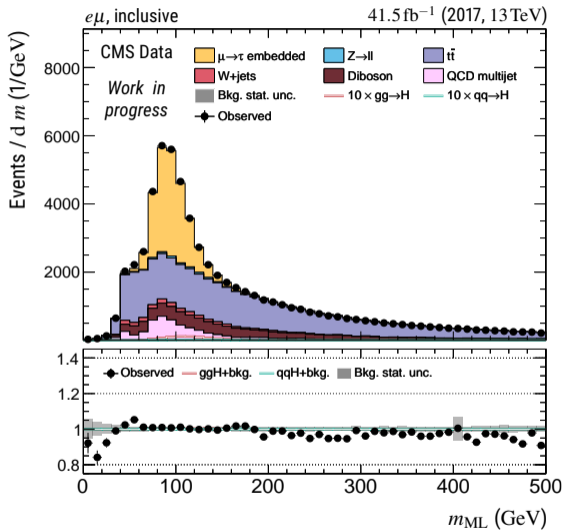
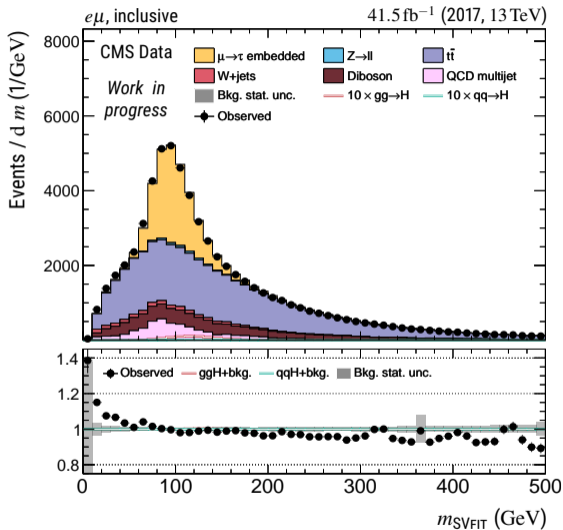
► **vs m_{SVFIT}** : Same looking shapes, expected.



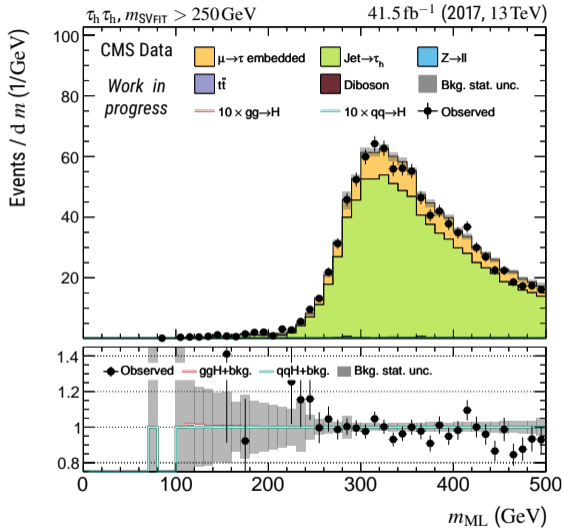
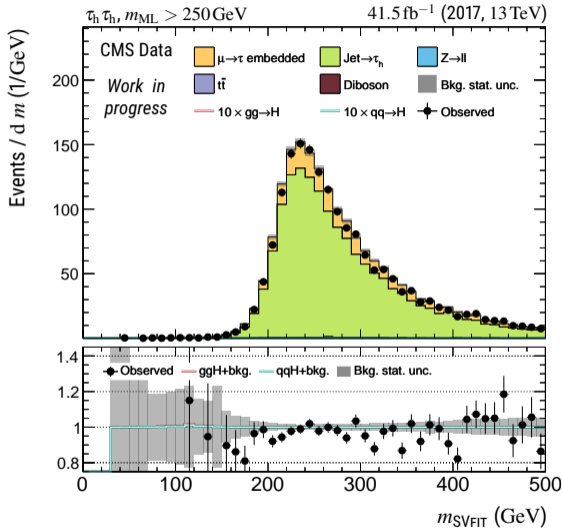
▶ **vs m_{SVFIT} : Better Data-Bkg agreement with m_{ML} !**



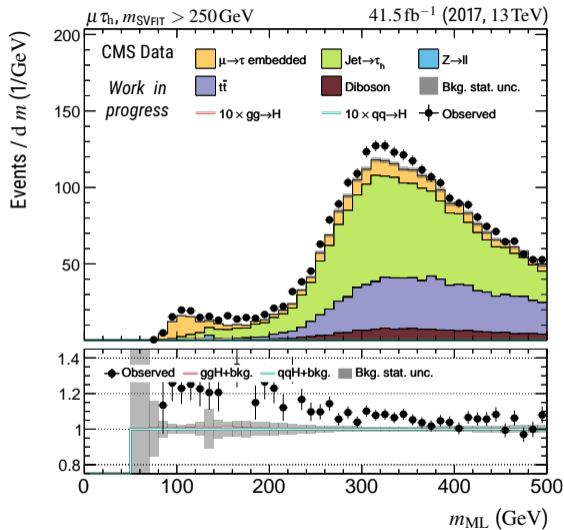
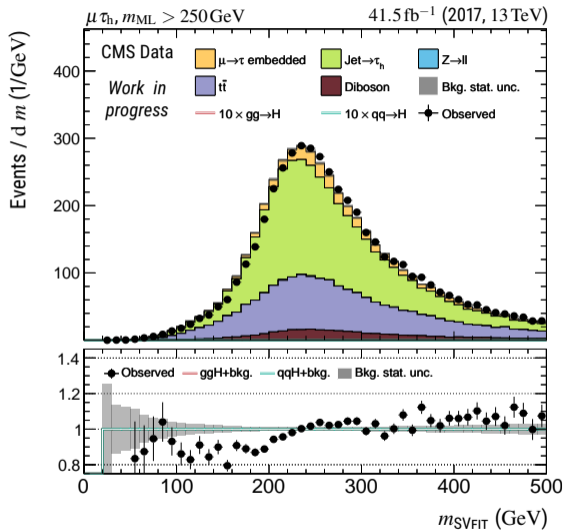
► **vs m_{SVFIT}** : Slightly better Data-Bkg agreement with m_{ML} .



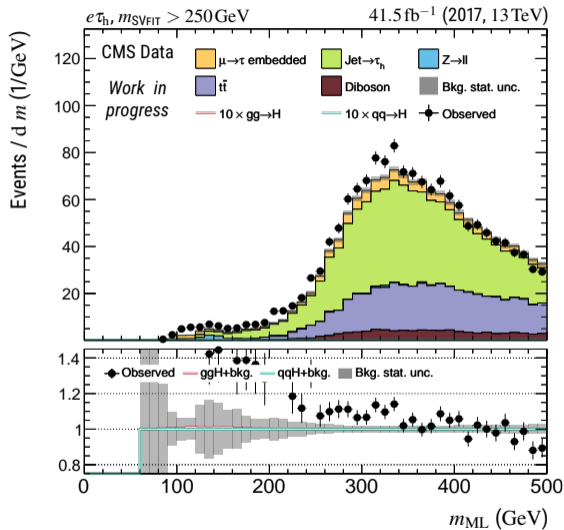
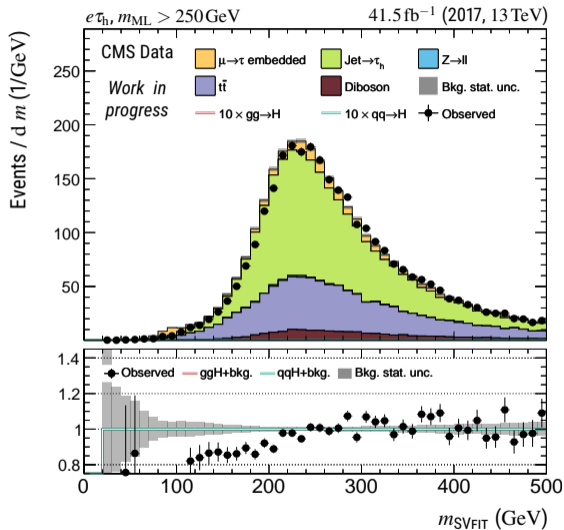
▶ **vs m_{SVFIT}** : Slightly better Data-Bkg agreement with m_{ML} .



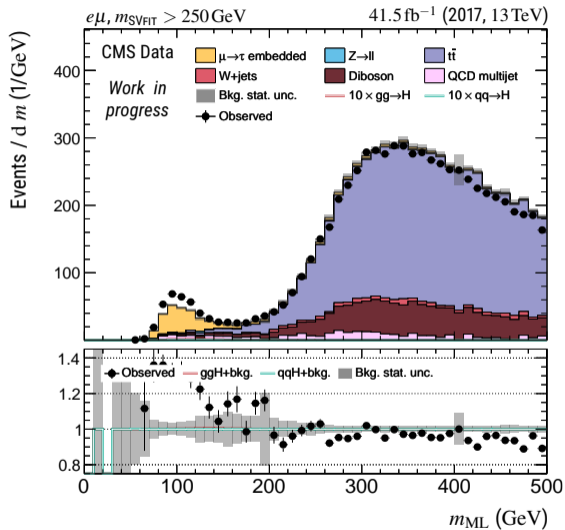
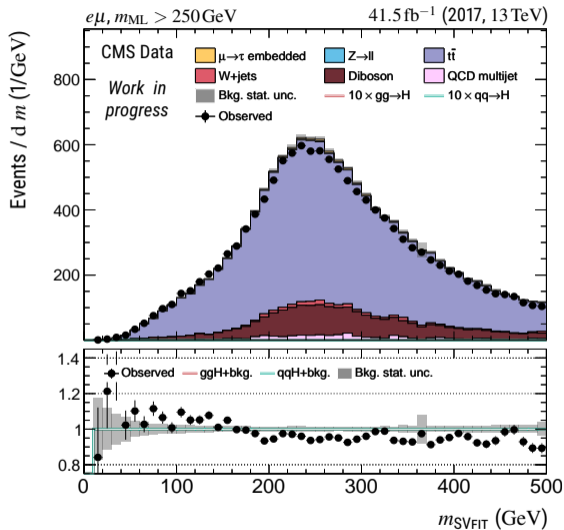
► Distributions start to populate at 200 or 300 GeV. **Yield is different**



► Our model finds $Z \rightarrow \tau\tau$ events at $\sim 100 \text{ GeV}$ in the $m_{SVFIT} > 250 \text{ GeV}$ region!



▶ Same kind of effect as seen in $\mu\tau_h$.



▶ Same kind of effect as seen in $\mu\tau_h$.

