# DeepMET
## Improving Missing Transverse Momentum estimation with a DNN

Yongbin Feng (Texas Tech University)

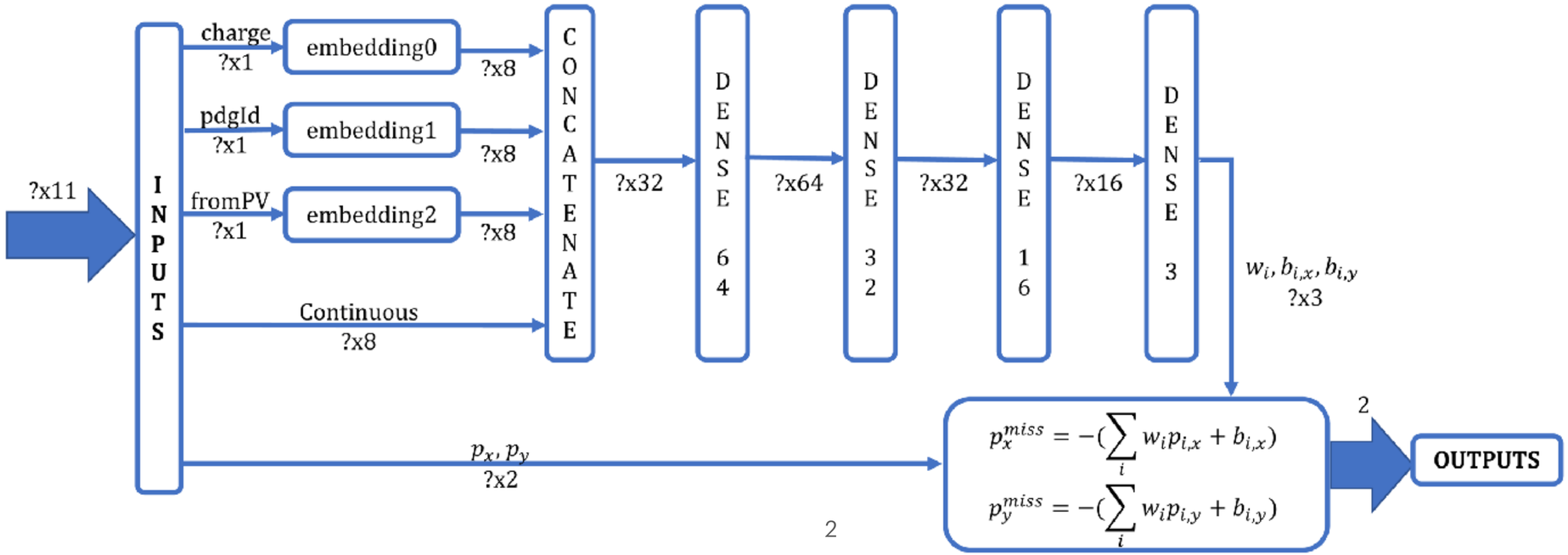
On behalf of CMS Collaboration

BOOST 2025, Providence RI

July 28th, 2025

# Introduction

- Improving MET estimation with ML:
  - Benefiting from the recent fast developments in the ML/DL community to improve on our reconstruction performance, such as MET
  - Inputs are “low” level features - all PF candidates in one event
  - ML model (DNN) assigns individual weights (and biases) to each PF candidate
  - Outputs are the negative vector sum of weighted PF pT, i.e., similar to PUPPI MET

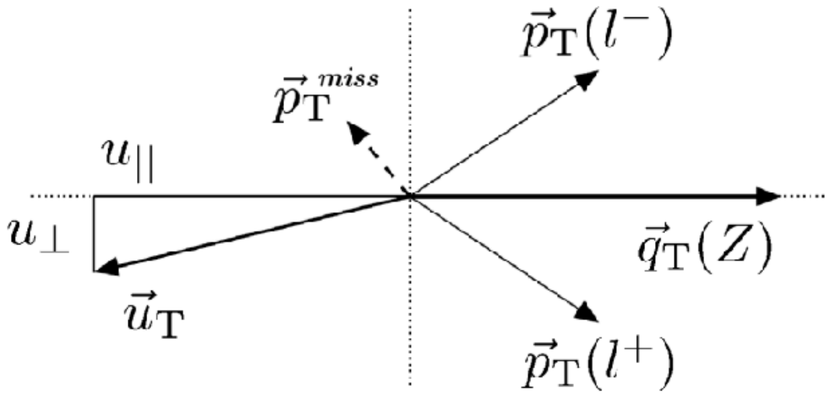


# Main Deliverables

- **Improvements**: With a small DNN (4500 trainable parameter), the performance is found out to be improved (significantly) across different processes
- **Use case**: Improvements of recoil estimations in the DY+jets and W+jets processes, being used in our super high-precision mW analysis
- **Generalization:** despite training only on DY+ttbar events, improvements on other processes over a wide range (high/low pt, with/without true MET, etc) are observed
- **Calibration:** With quantile corrections, the data-MC difference is under control, from which systematic uncertainties are also defined
- "**Auxiliary" studies**:
  - Training without PUPPI weights
  - Computational performance

# Ptmiss Parametrization

- Parametrization:
  - Uparal and Uperp: Uparal focus on scale, Uperp: detector noise, pileup, etc
  - For DY data and MC, take the vector pt sum of reco-ed muons as V pt
  - For W MC, take the vector pt sum of gen-level neutrino and (dressed) gen-level muon



# Algorithm

- Inputs: all PF candidates, each with eleven features
  - Eight continuous features: px, py, pT, eta, dxy, dz, mass, puppiWeights
  - Three categorical features: pdgId, charge, and fromPV (describing the PV association to the PV).

- DNN calculate different weights and biases ($\{w_i, b_i\}$) to all PF candidates. Output are the MET $p_{x,y}^{\text{miss}}$, estimated as

$$p_x^{\text{miss}} = -\sum_i (w_i p_{i,x} + b_{i,x})$$

$$p_y^{\text{miss}} = -\sum_i (w_i p_{i,y} + b_{i,y})$$

  - Bias terms were originally created such that NN has some freedom to handle px and py terms a little bit differently
  - But in reality we found out the effect is very small - the results with and without bias terms are very similar.
  - Here we keep bias term in the paper because the current NN in CMSSW has these terms

# Trainings

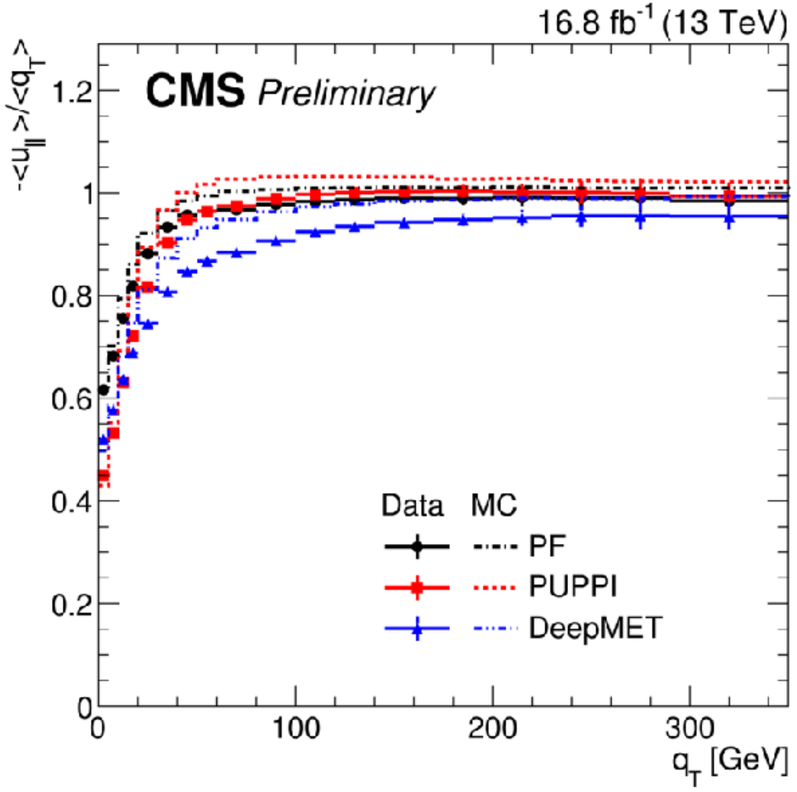
- Training samples are a mixture of DY + ttbar dilepton MC samples
  - Some leptons from the DY and ttbar samples are removed so that the NN can learn different lepton scenarios
  - The fraction of DY and ttbar events are about the same. Varied the ttbar fraction to be 20-50% and the performance change is pretty small

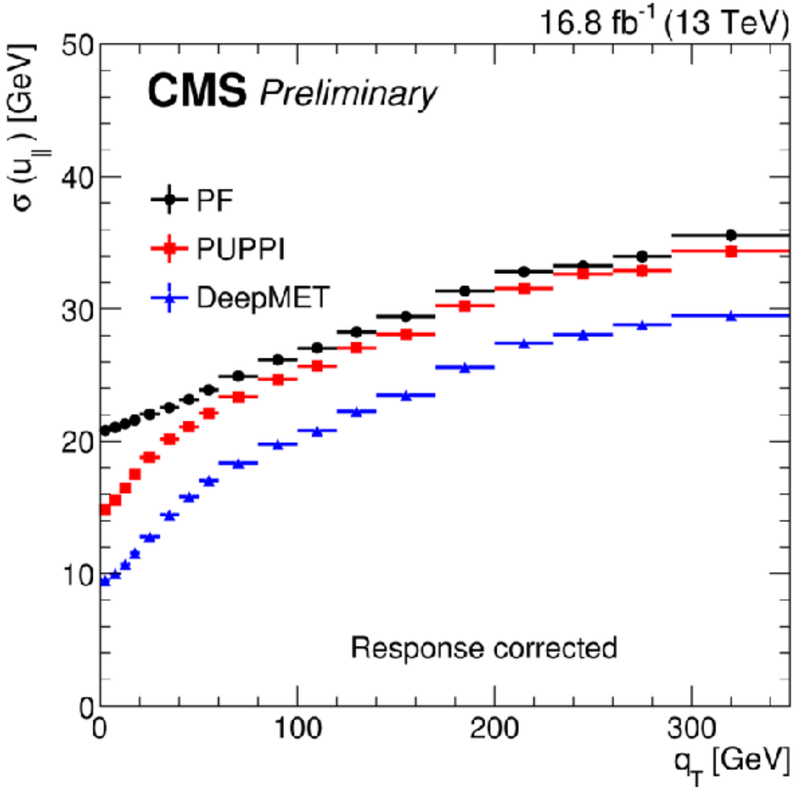
- Loss function

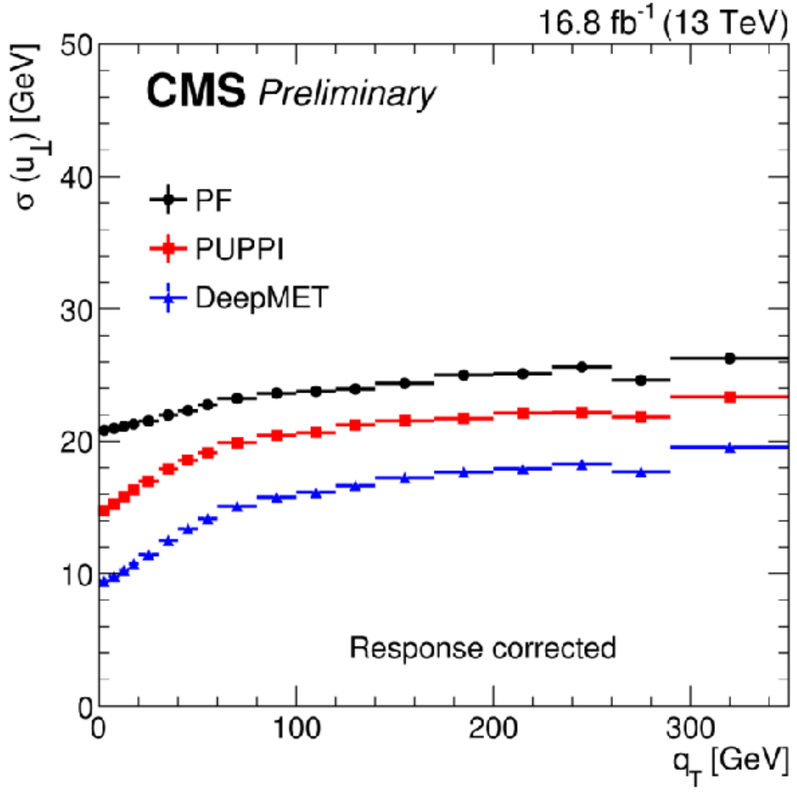
$$L = \frac{1}{2}[(p_x^{\text{miss}} - p_x^{\hat{\text{miss}}})^2 + (p_y^{\text{miss}} - p_y^{\hat{\text{miss}}})^2] + c \cdot |(\sum_{R>1} R - 1) - (\sum_{R<1} 1 - R)|$$

  - Mean Square Error (MSE) (+ some response correction terms)
  - The response correction term can be added, with C as hyper-parameter. Checked c=2, 10, 200, the effect on the final performance is small

# Physics Performance: DY Data

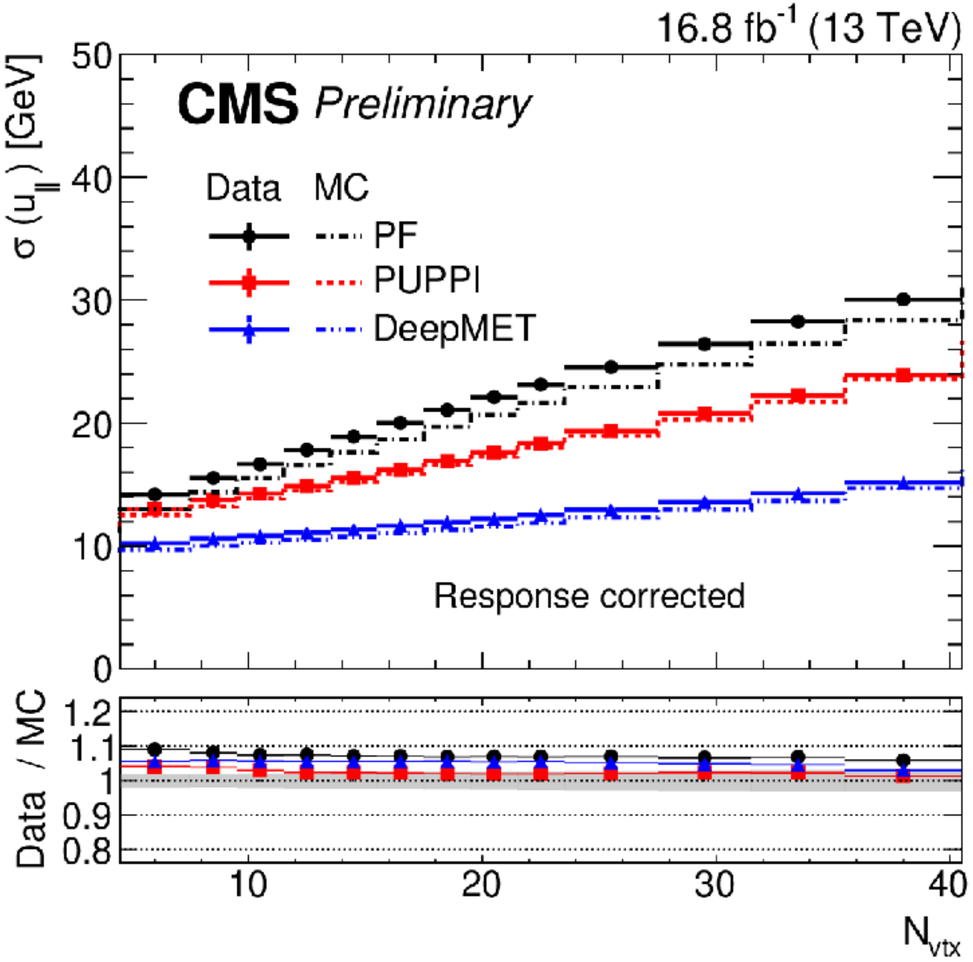


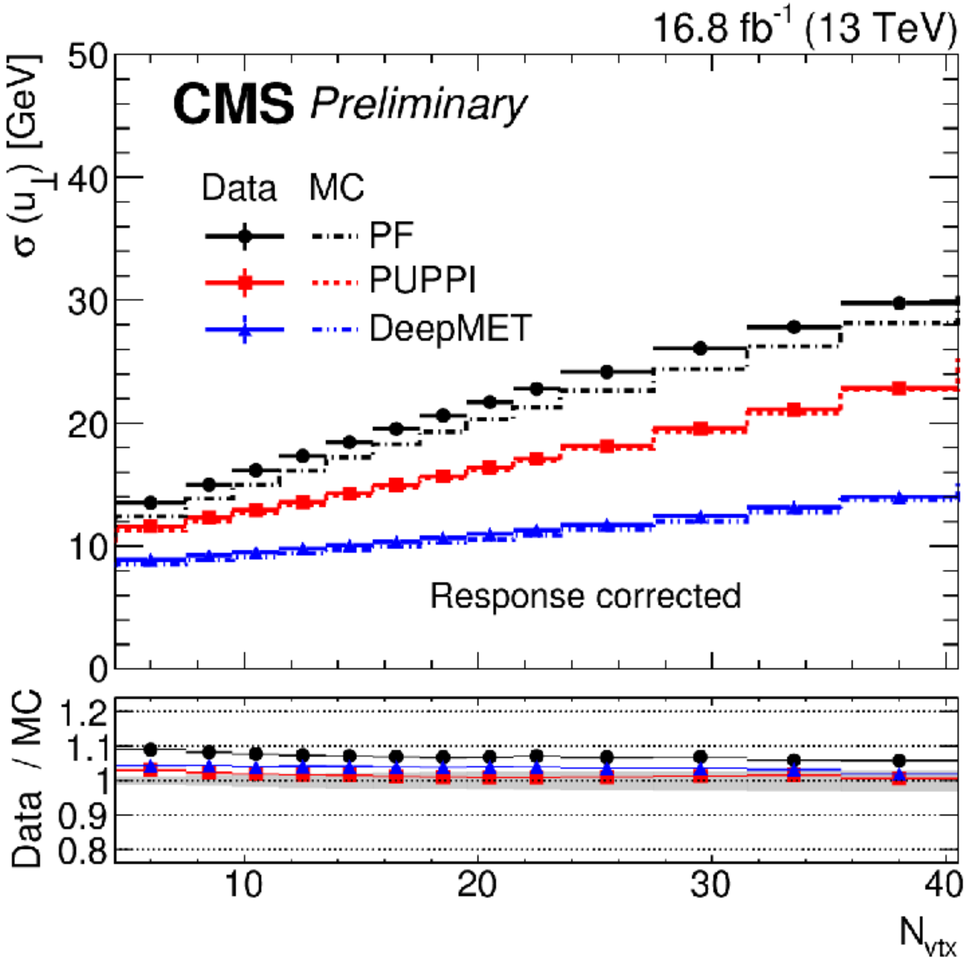




- Response and resolutions of DeepMET, compared with PF and PUPPI
- Resolutions are calculated using RMS (consistent with JME-17-001)
- Resolutions are response-corrected:
  - Calculate average response in the plateau region (qT>150GeV), and use 1/<resp> as the resolution correction

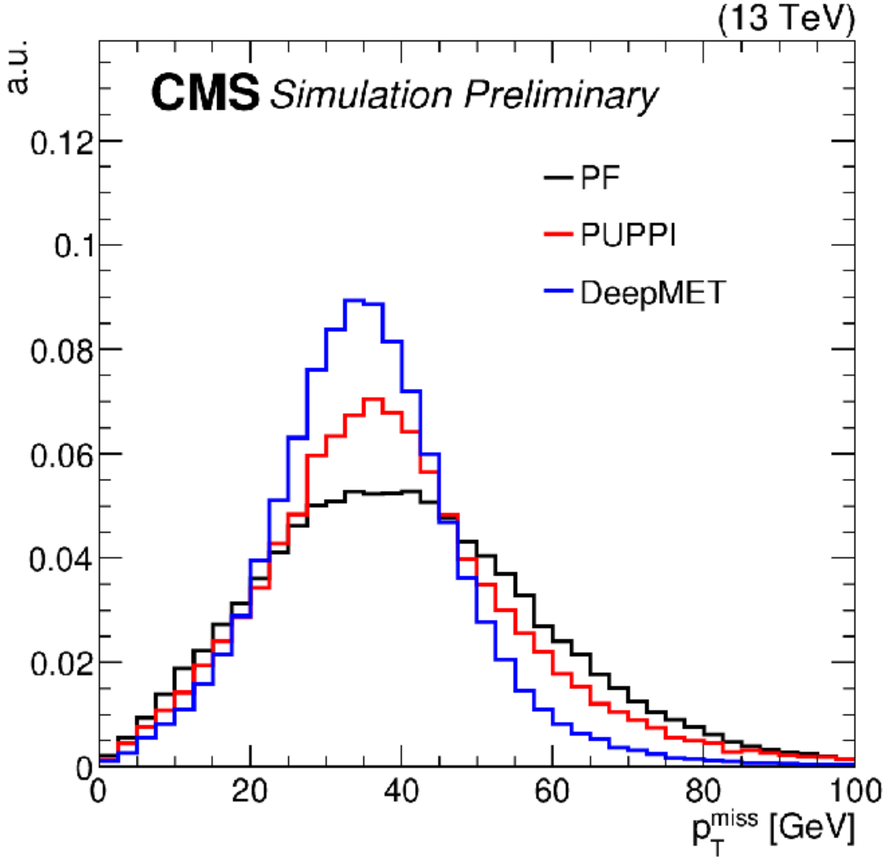
# Physics Performance: DY Data

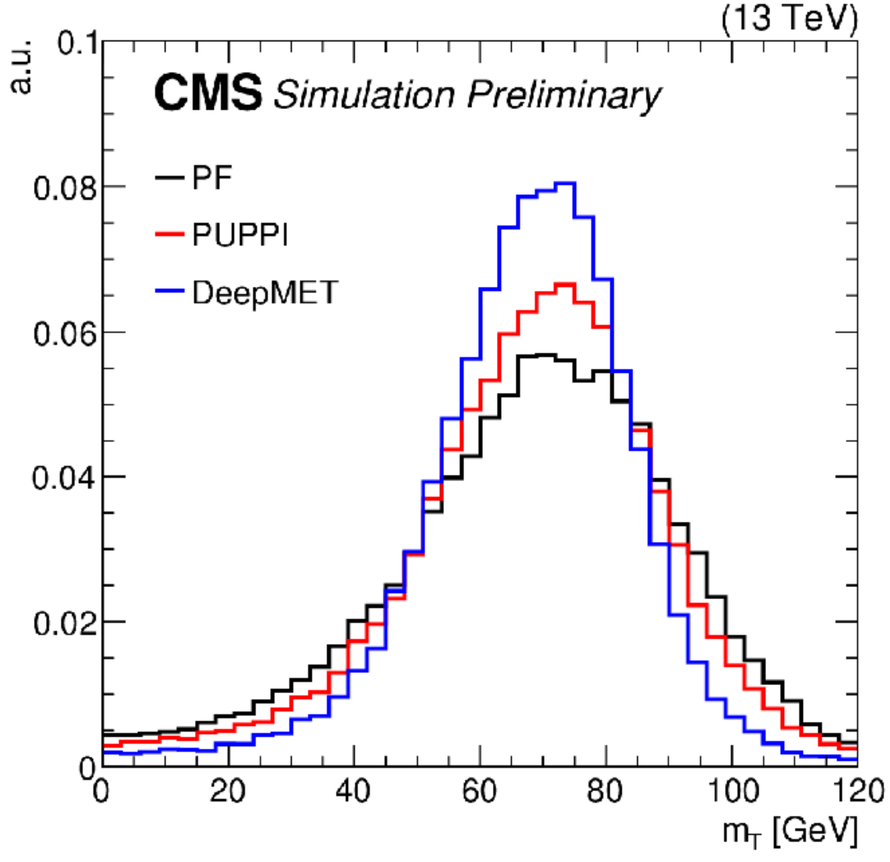



- Resolutions vs nPV, with response corrections
- DeepMET is more robust (flat) with nPV.
- Large potential for Run-2 and Run-3 analyses, more gains for Phase-2 with higher pileup

# Physics Performance: W+jets MC

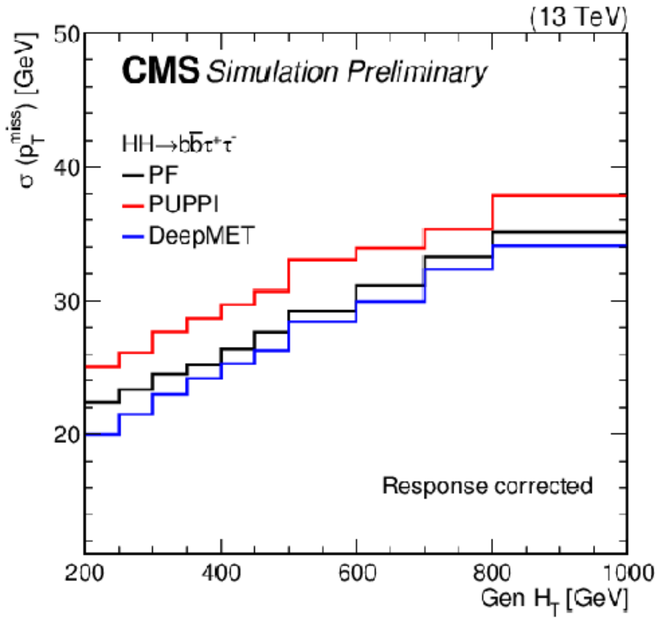


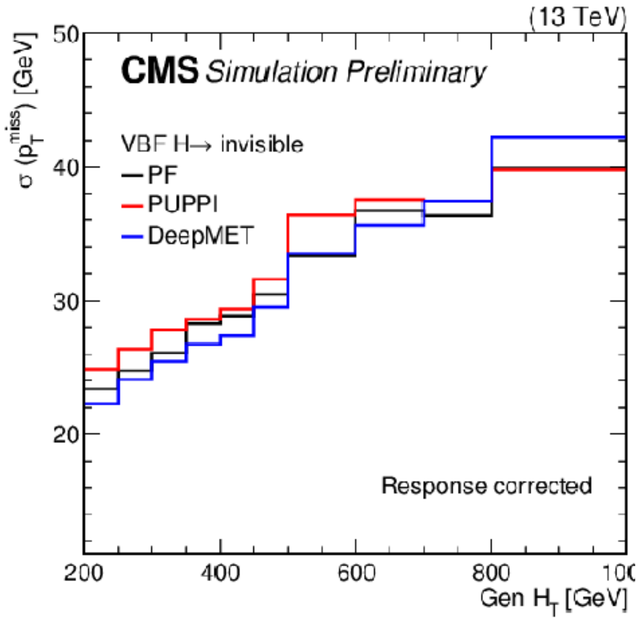



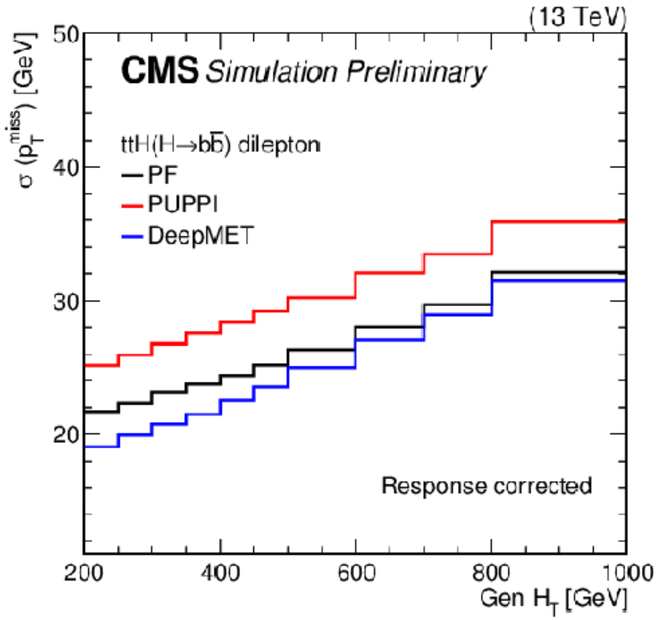

- MET and mT distributions in W+Jets MC.
- In both of these DeepMET shows a nicer Jacobian peak.

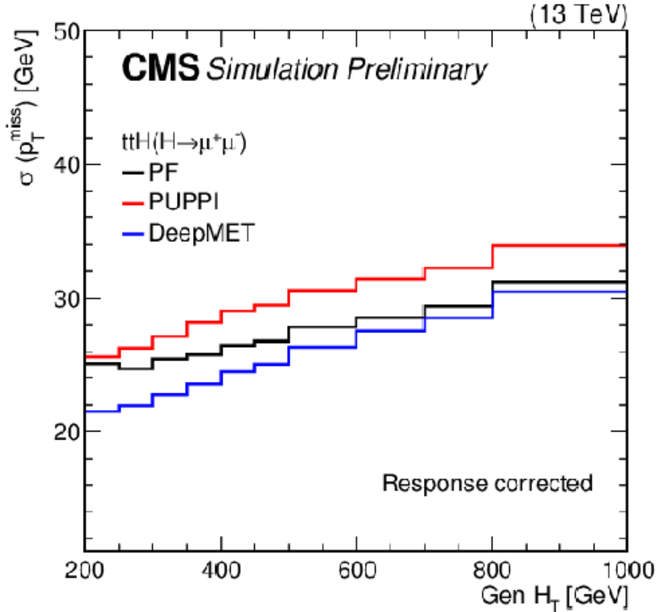
# Performance on other Processes

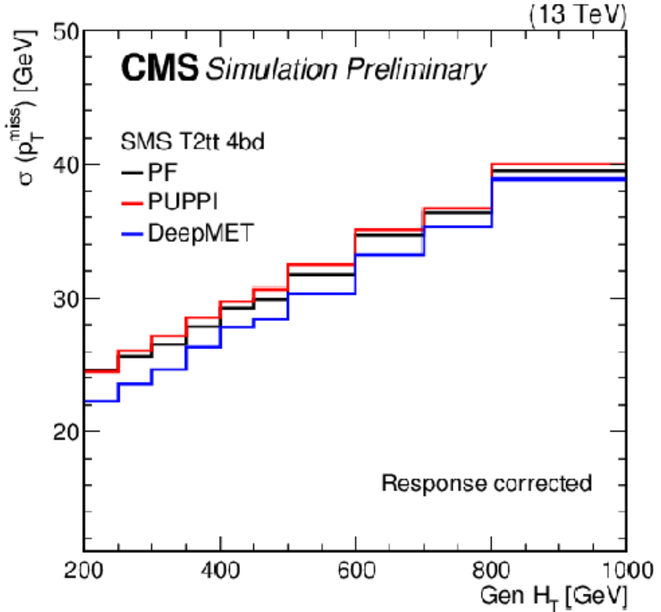
- Generalize well to different processes. Good resolution observed

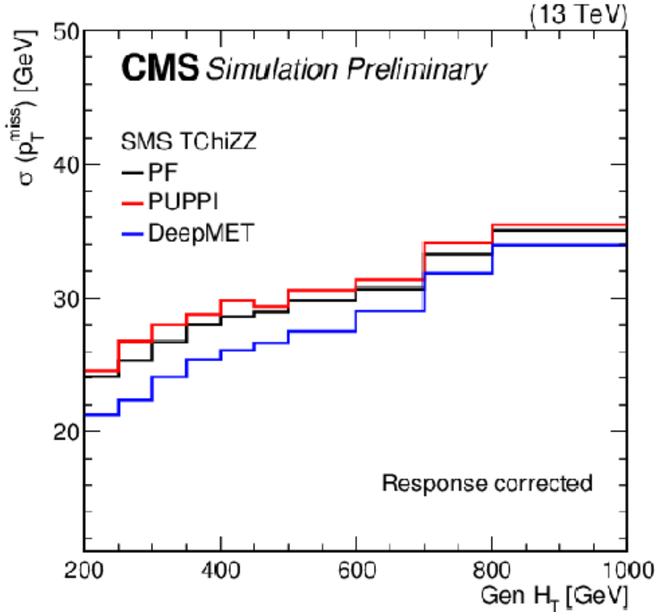


# DeepMET Calibrations

- Quantile correction:
  - Model the uparal and uperp distributions in data and MC with double Gaussian distributions
  - Correct the uparal and uperp distributions in MC back to data
  - Do this in different qT and NJets bins

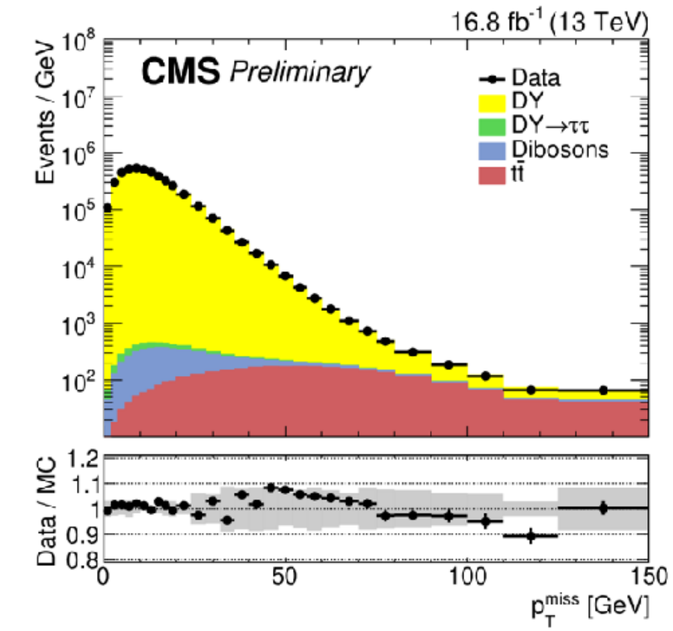
$$p_{\parallel} = CDF_{ibin}^{u_{\parallel},MC}(u_{\parallel}^{MC}),$$
$$p_{\perp} = CDF_{ibin}^{u_{\perp},MC}(u_{\perp}^{MC}).$$

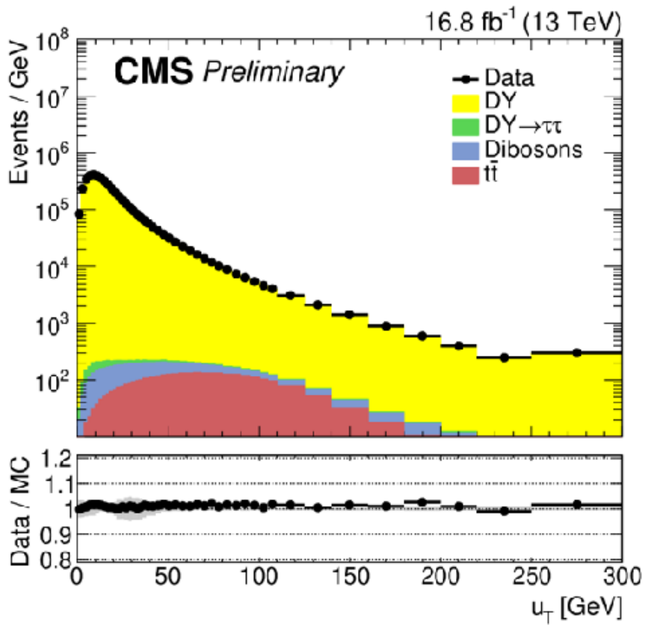
$$u_{\parallel}^{Data} = (CDF_{ibin}^{u_{\parallel},Data})^{-1}(p_{\parallel}),$$
$$u_{\perp}^{Data} = (CDF_{ibin}^{u_{\perp},Data})^{-1}(p_{\perp}).$$

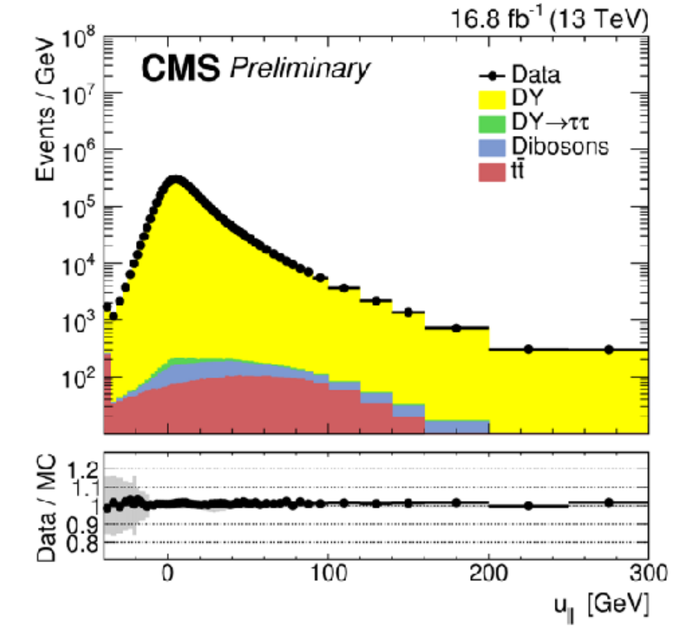
- Uncertainties:
  - Choice of the smoothing function: double Gaussian -> Gaussian smooth functions
  - Background subtraction: varying the background contributions by 20% and re-derive corrections

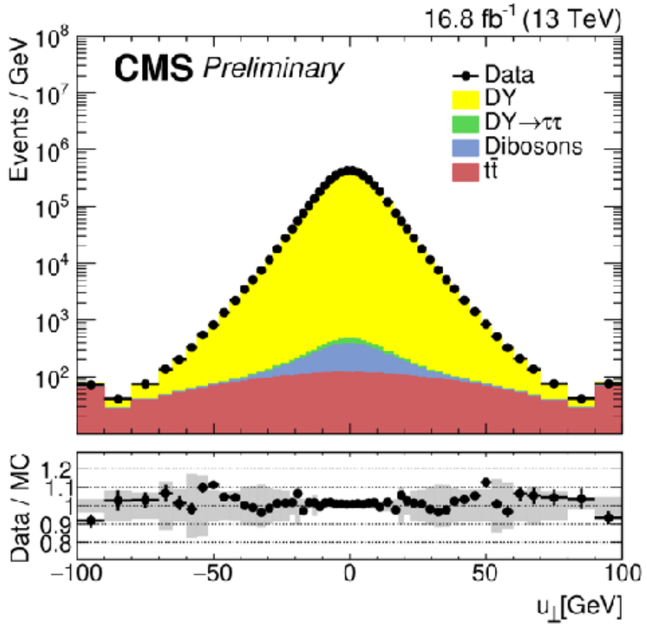
# DeepMET Calibrations

- Good agreement between data and MC after calibrations

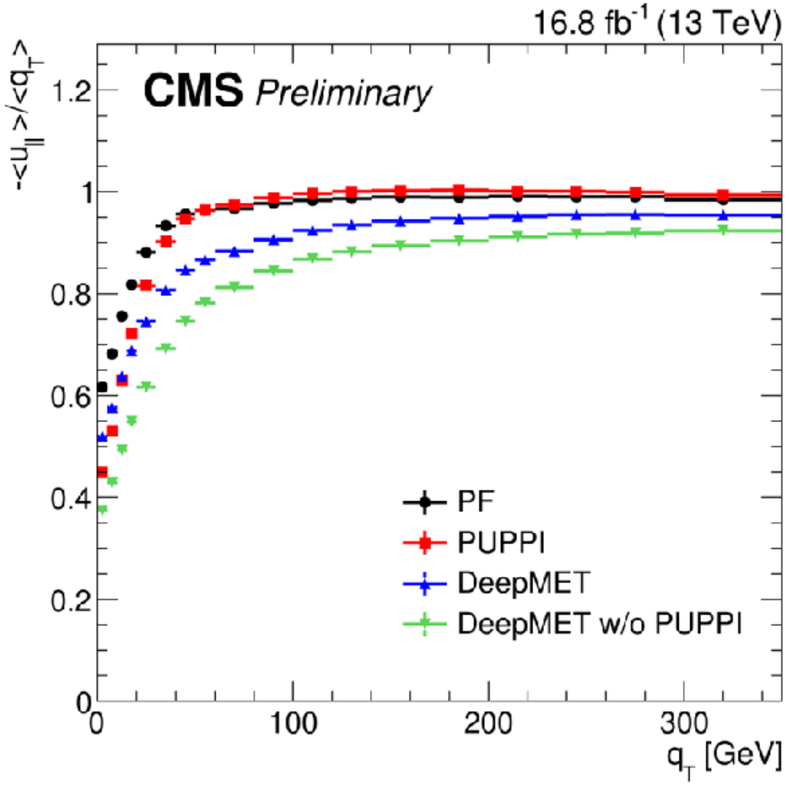


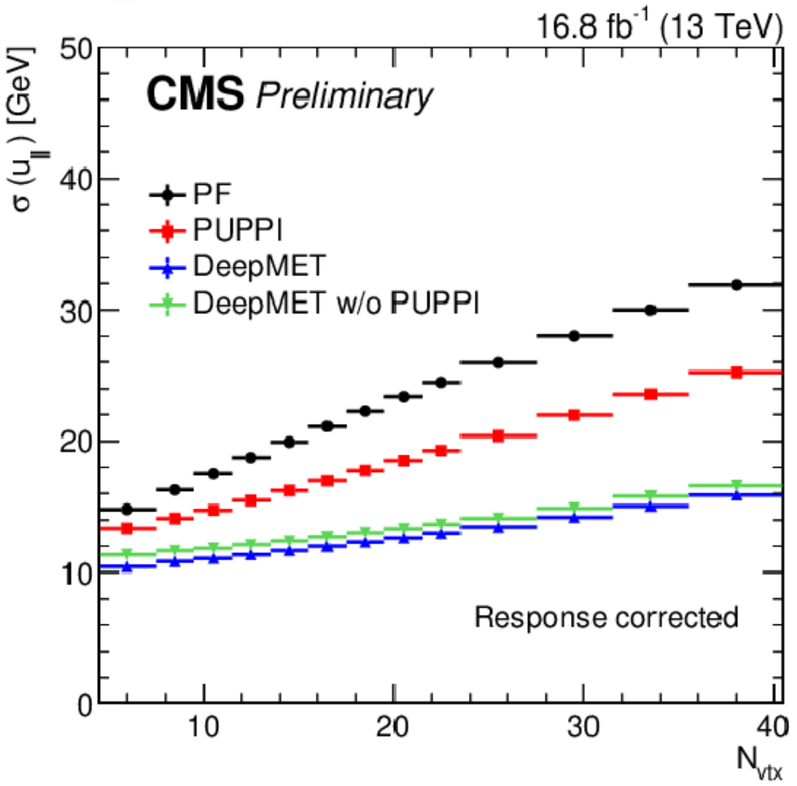



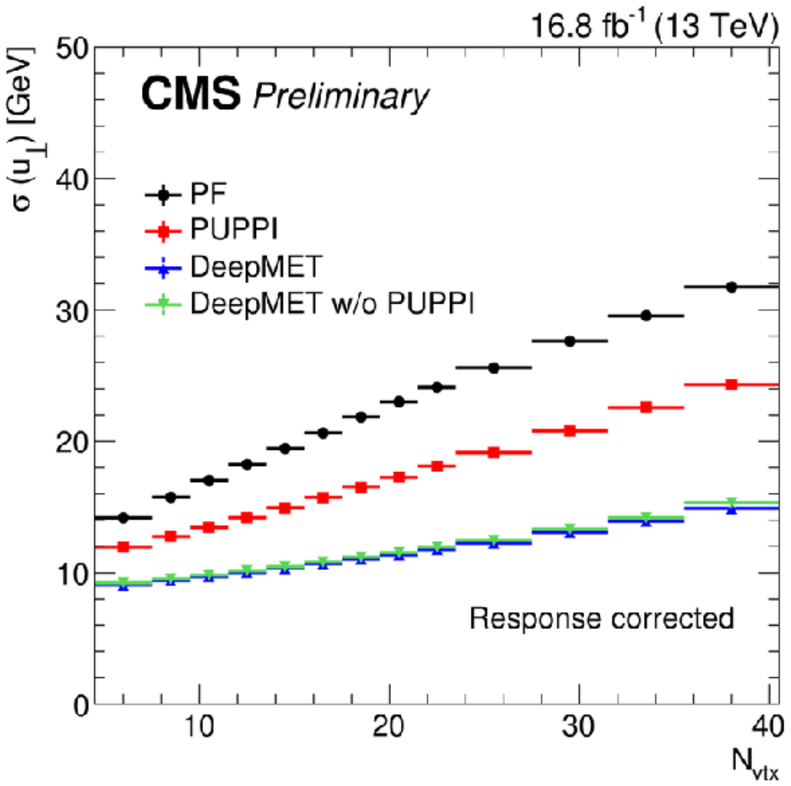



# DeepMET Checks




- Check the performance on DY MC samples of the trainings with PUPPI weights (blue) and without PUPPI weights (green)
- DNN is not able to make use of local neighboring information. So expect some performance drop when training without PUPPI weights
- It seems the effect is mostly on the response

# Computing Performance

- Model:
  - 4541 trainable parameters in total, very small
  - Implemented with Keras + TensorFlow
- Training
  - O(10K) DY events and O(10K) ttbar events
  - Trained on NVIDIA Quadro P6000. Took O(30mins) to converge - very fast; on CPUs with 40 cores it took roughly 3 hours
- Inference
  - CMSSW implementation: TF frozen graph.
  - Number of input PF candidates are fixed to 4500. Inputs are zero-padded if less than 4500
  - Inference time is around 10ms on one CPU single core. For comparison CMS RECO/AOD and MiniAOD production take around 1-10s. So DeepMET inference very fast.

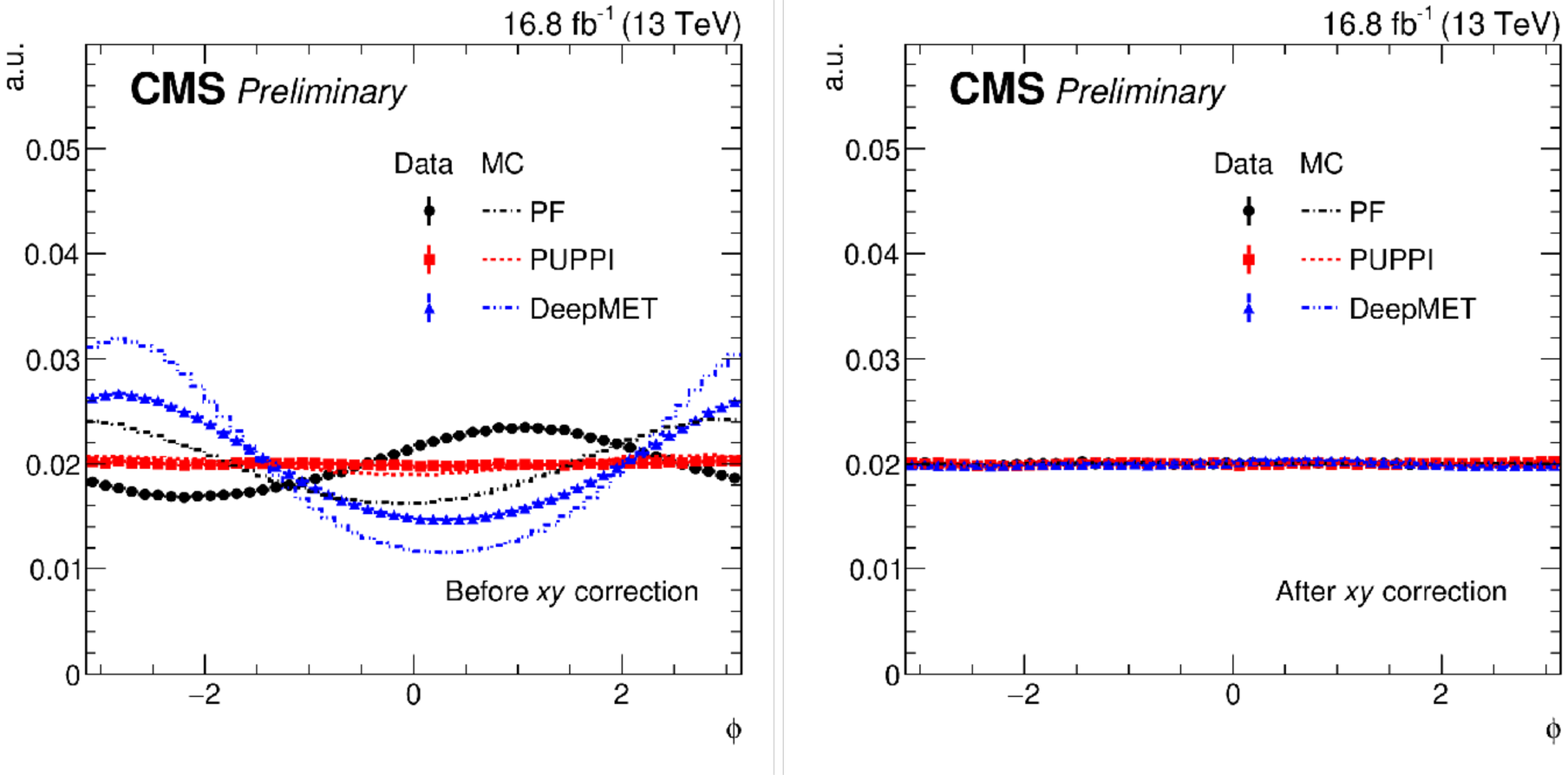
# Summary

- Presented the DeepMET algorithm. Particle-level inputs, event-level outputs, with a “simple” DNN
- It achieves 10-30% improvements compared with other MET estimators.
- The performance is robust to different processes.
- After calibrations good data-MC agreement achieved. Uncertainties derived during the calibration process, similar to other ML algorithms
- Also studied training without PUPPI weights, and integrated into CMSSW. Inference cost is very small

- More information: CMS-PAS-JME-24-001

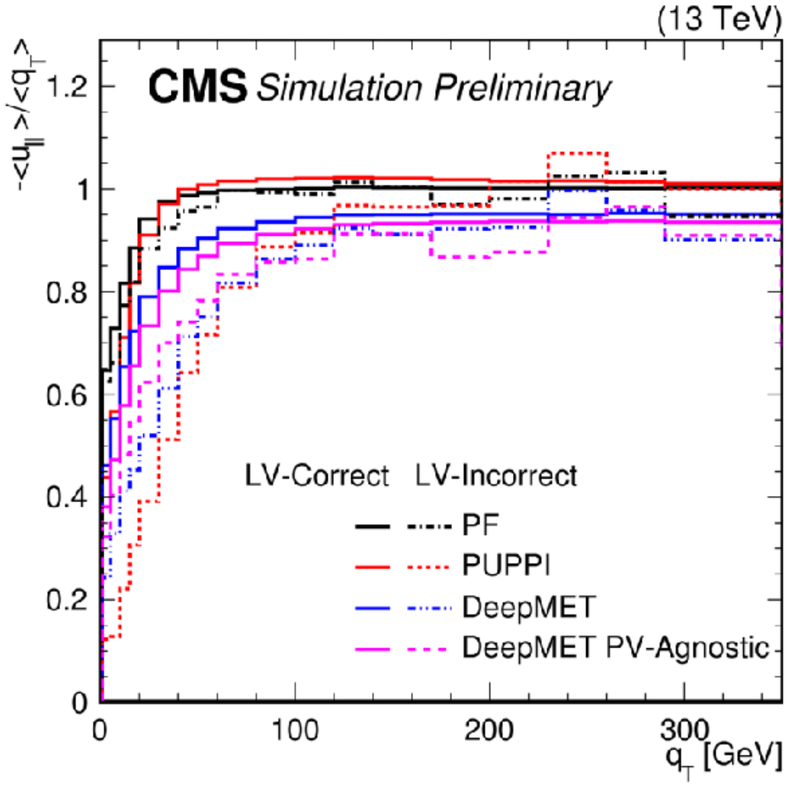
# Back Up

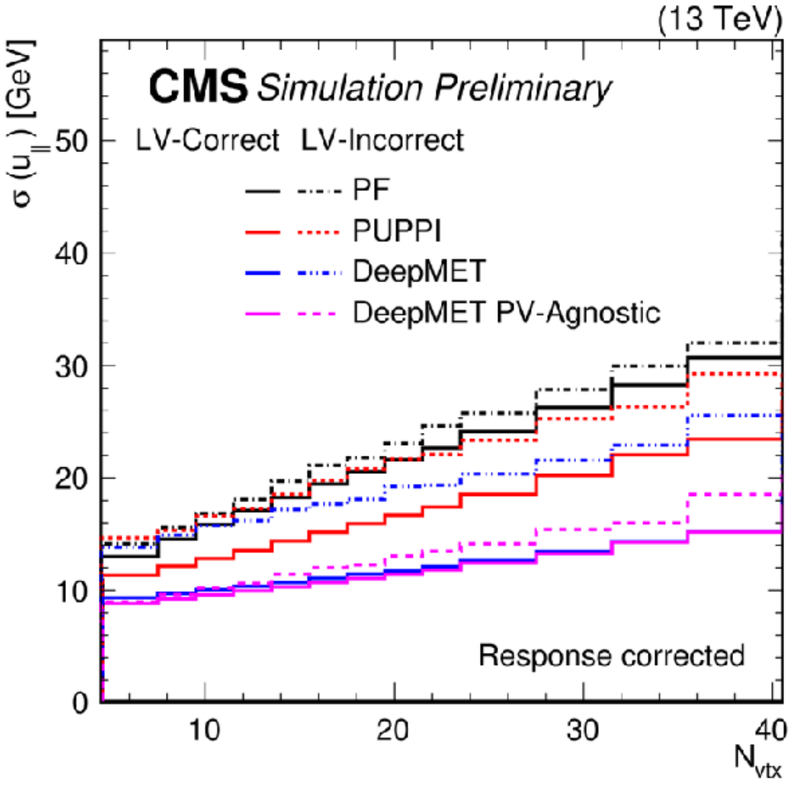
# DeepMET XY Corrections

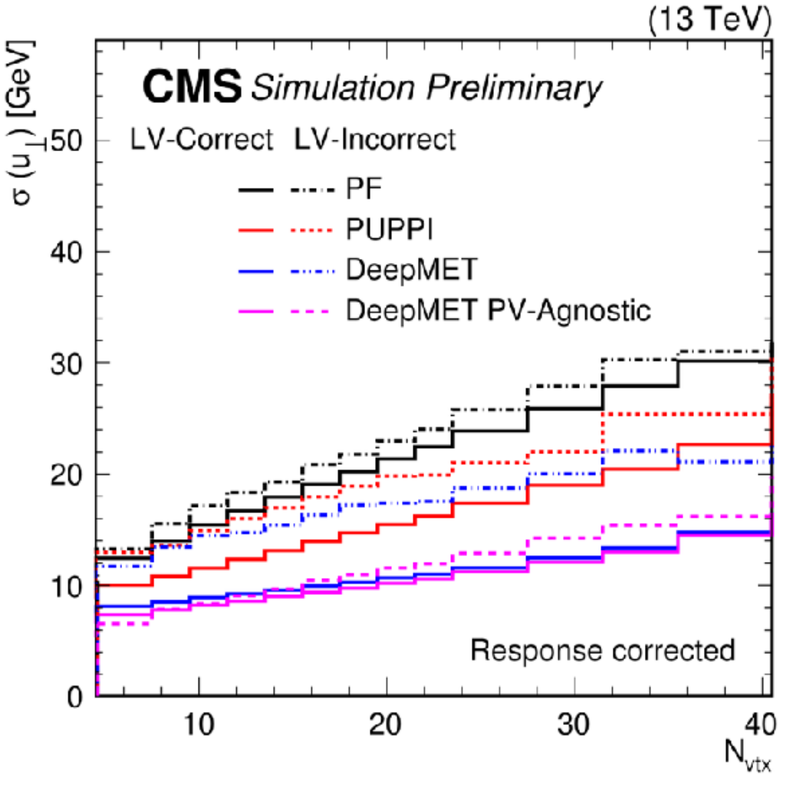



- Derive the standard XY corrections and apply to DeepMET
- After the XY correction, the phi dependence is basically fixed

# PV-Agnostic Version of DeepMET




- For W analyses, in most scenarios one single muon + soft tracks. PV reconstruction can be either unsuccessful, or mis-identified as a pileup track
- To fix this, manually pick the PV with closest z to the muon:
  - If no PV with dz(muon, PV)<0.2cm, use beam spot
- Re-calculate PUPPI weights using the new PV.
- This is the version using in CMS W mass analysis