The role of Machine Learning algorithms for object identification & reconstruction in CMS

Alessandro Calandri, Davide Valsecchi
ETH Zürich

IPA workshop on Machine Learning for Particle Physics and Astrophysics

Outline of the talk

➡ **Machine Learning techniques are presently being developed for several aspects of the CMS reconstruction chain**

- object identification and reconstruction (tracking, clustering, jet tagging, lepton identifications)
- global event features (Particle Flow, MET reconstruction)
- pile-up mitigation and definition of detector geometry for HL-LHC (HGCAL)

➡ **Will present a selection of ML efforts from several perspective underlining impact on physics measurements and ETH contributions to developments**

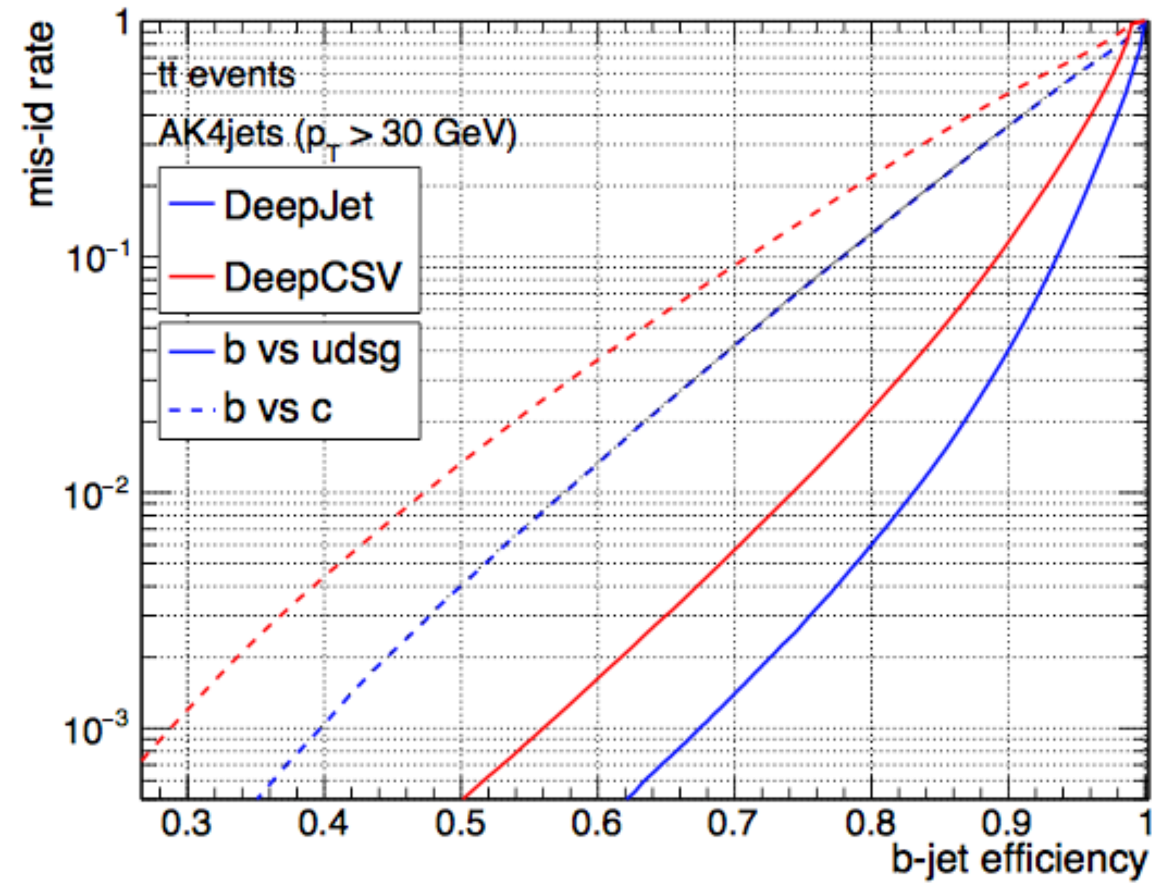
- jets and flavour tagging (earlier developments, ParticleNet, ParticleTransformer); jet efficiency determination
- super-clustering for electron/photon reconstruction
- ML-based regressions - the example of the b-jet energy regression
- ML developments tackling pile-up mitigation
- a quick glimpse on ML used for ParticleFlow reconstruction
- examples of ML beneficial usage in physics analyses (jet flavour tagging and regressions)

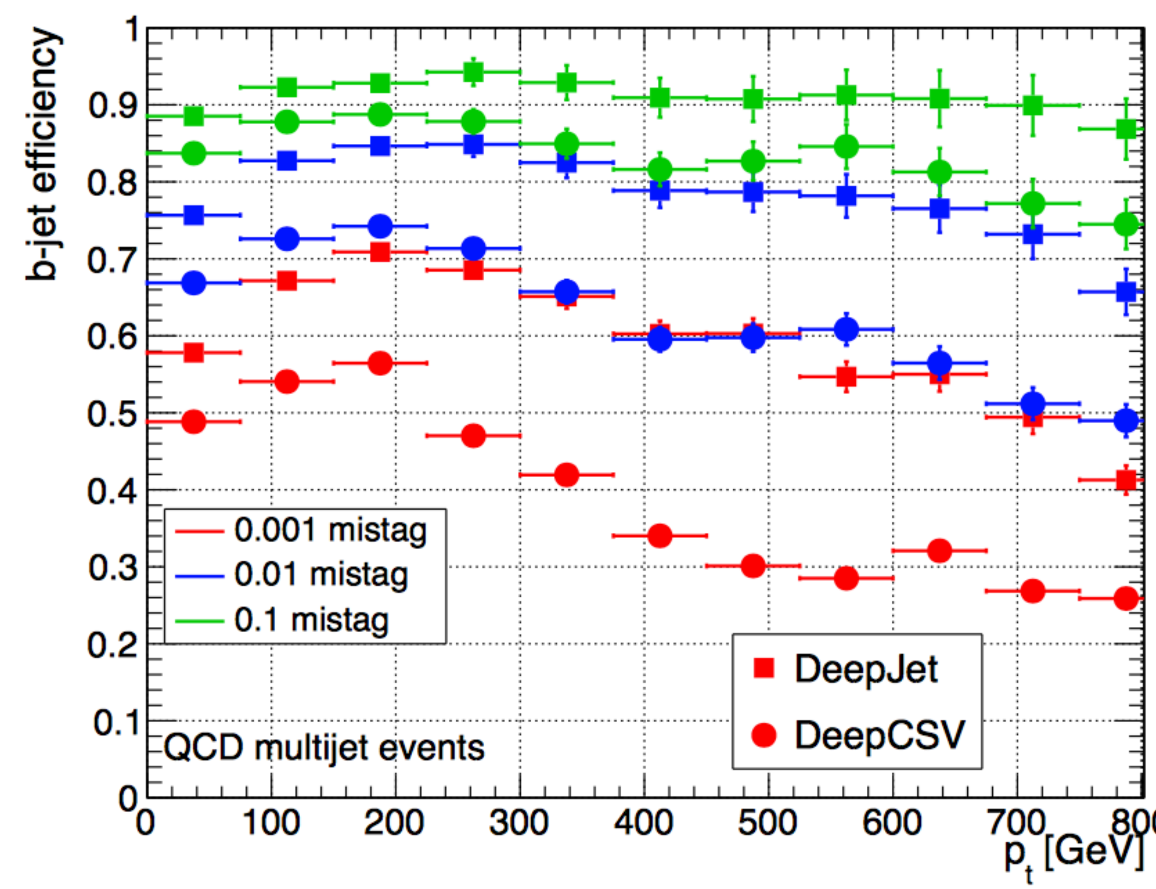
ML for Jet Tagging

DeepJet (JINST 15 (2020) P12012)

➡ Most active area of research for ML developments: several applications, e.g. resolved and boosted jet tagging, quark/gluon discrimination, …

- exploited several architectures from simple feed-forward DNN to CNN, RNN, point-cloud and Transformer models
- developed techniques to make use of low-level features (PF Candidates) to improve algorithm performance w/o degradation of data/MC modelling
- steady improvement in performance especially for b-tagging and jet-identification
  - DeepCSV (DNN architecture), DeepJet (RNN + PF Candidates), DeepAK8 (CNN), ParticleNet (GNN)
  - DeepCSV/DeepJet, DeepAK8/PNet widely used in several CMS analyses for AK4 and AK8 jet
    - ✷ largest improvement especially observed in high momentum phase-space (large track multiplicity)



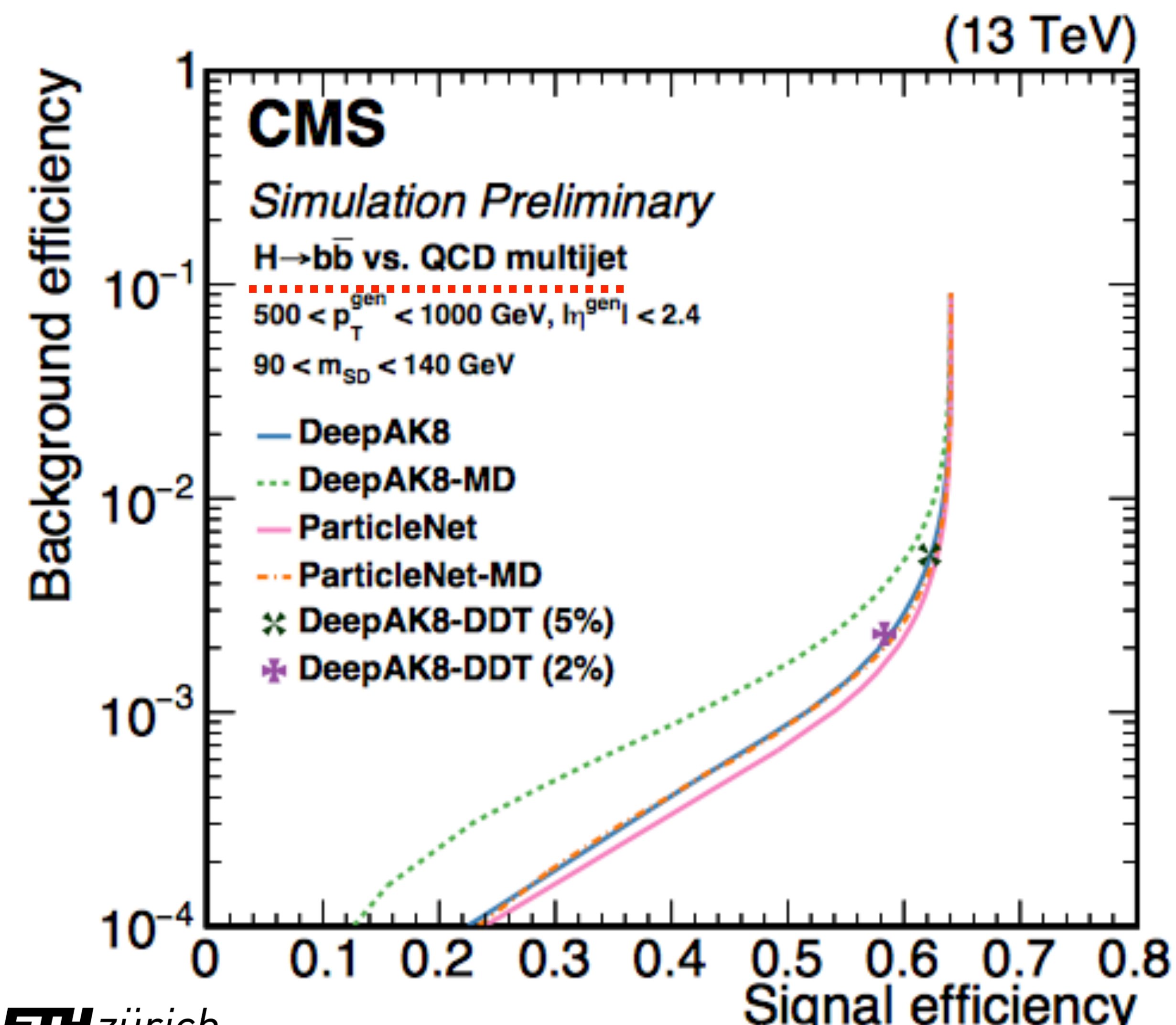


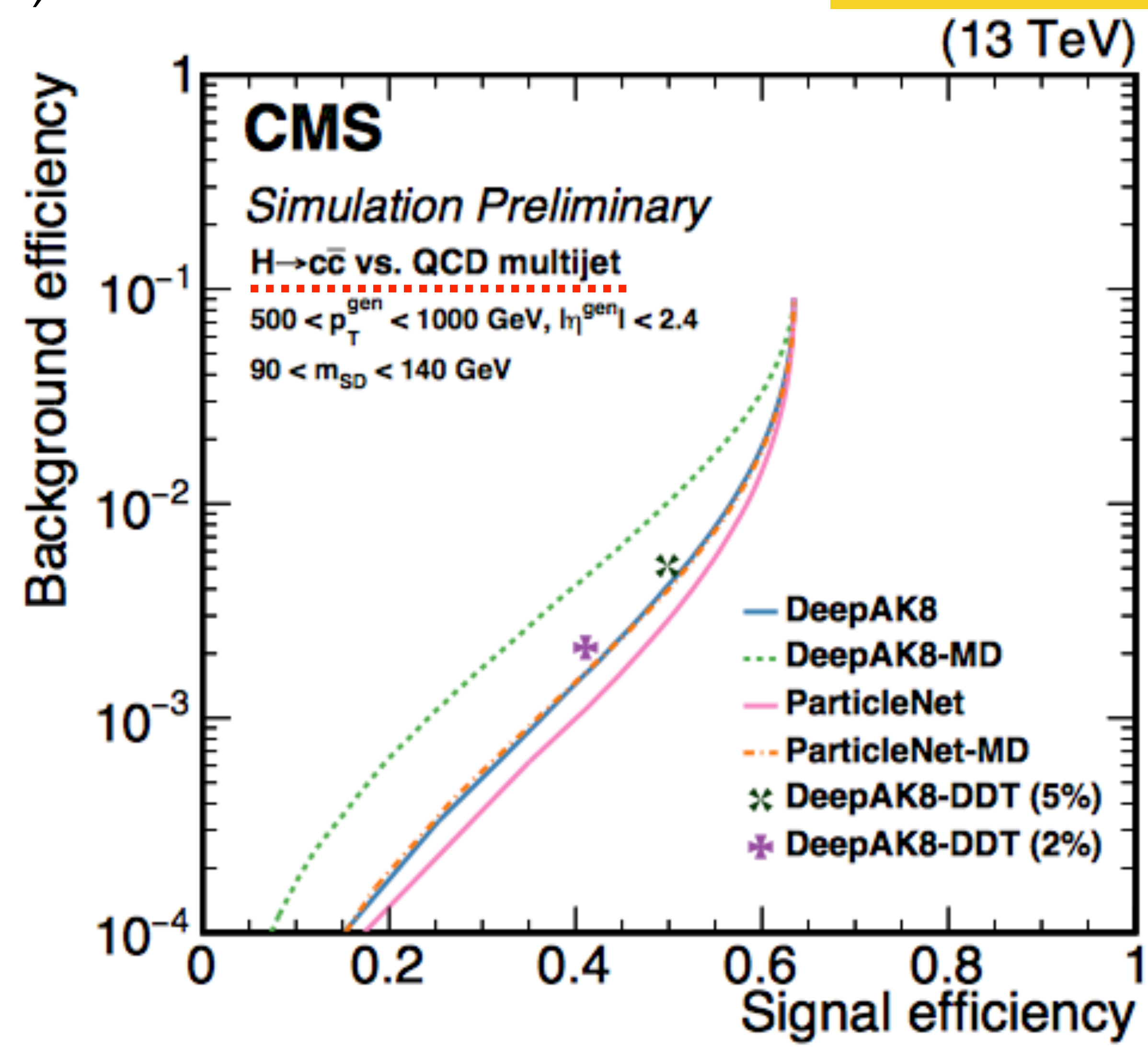
ML for Jet Tagging – ParticleNet

➡ EdgeConv GNN based algorithm using unordered jet constituents and ParticleFlow candidates

- using same GNN architecture for AK8 and AK4 tagging
- several output nodes for bb, cc, LF and QCD jets, using two-prong hadronic decays of highly boosted objects as signal and QCD as background training samples
- achieved excellent mass-decorrelation of PN tagger output at the price of low drop in performance (backup material)
- large improvement over previous boosted taggers (DeepAK8, DeepDoubleX) making use of high-level features and different architectures (CNN)

CMS-DP 2020-002





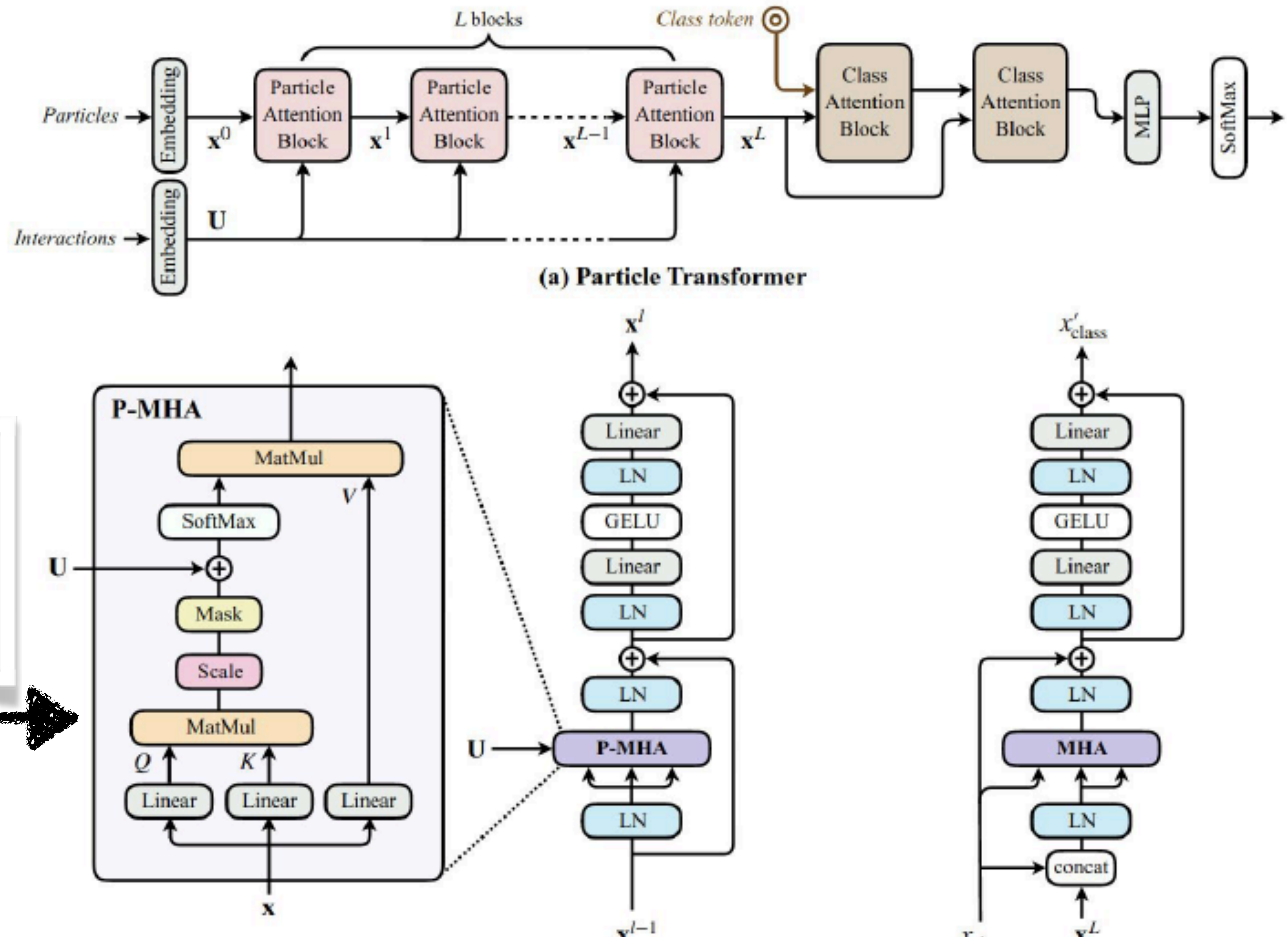
ML for Jet Tagging - Particle Transformer for AK4 jets

➡ Developing studies towards replacement of DeepJet with ParticleTransformer

arXiv:2202.03772

- pairwise interaction features between all jet constituents and secondary vertices → exploit internal correlation of jet constituents

$$
\begin{aligned}
\Delta &= \sqrt{(y_a - y_b)^2 + (\phi_a - \phi_b)^2}, \\
k_{\mathrm{T}} &= \min(p_{\mathrm{T},a}, p_{\mathrm{T},b})\Delta, \\
z &= \min(p_{\mathrm{T},a}, p_{\mathrm{T},b})/(p_{\mathrm{T},a} + p_{\mathrm{T},b}), \\
m^2 &= (E_a + E_b)^2 - \|\mathbf{p}_a + \mathbf{p}_b\|^2,
\end{aligned}
$$



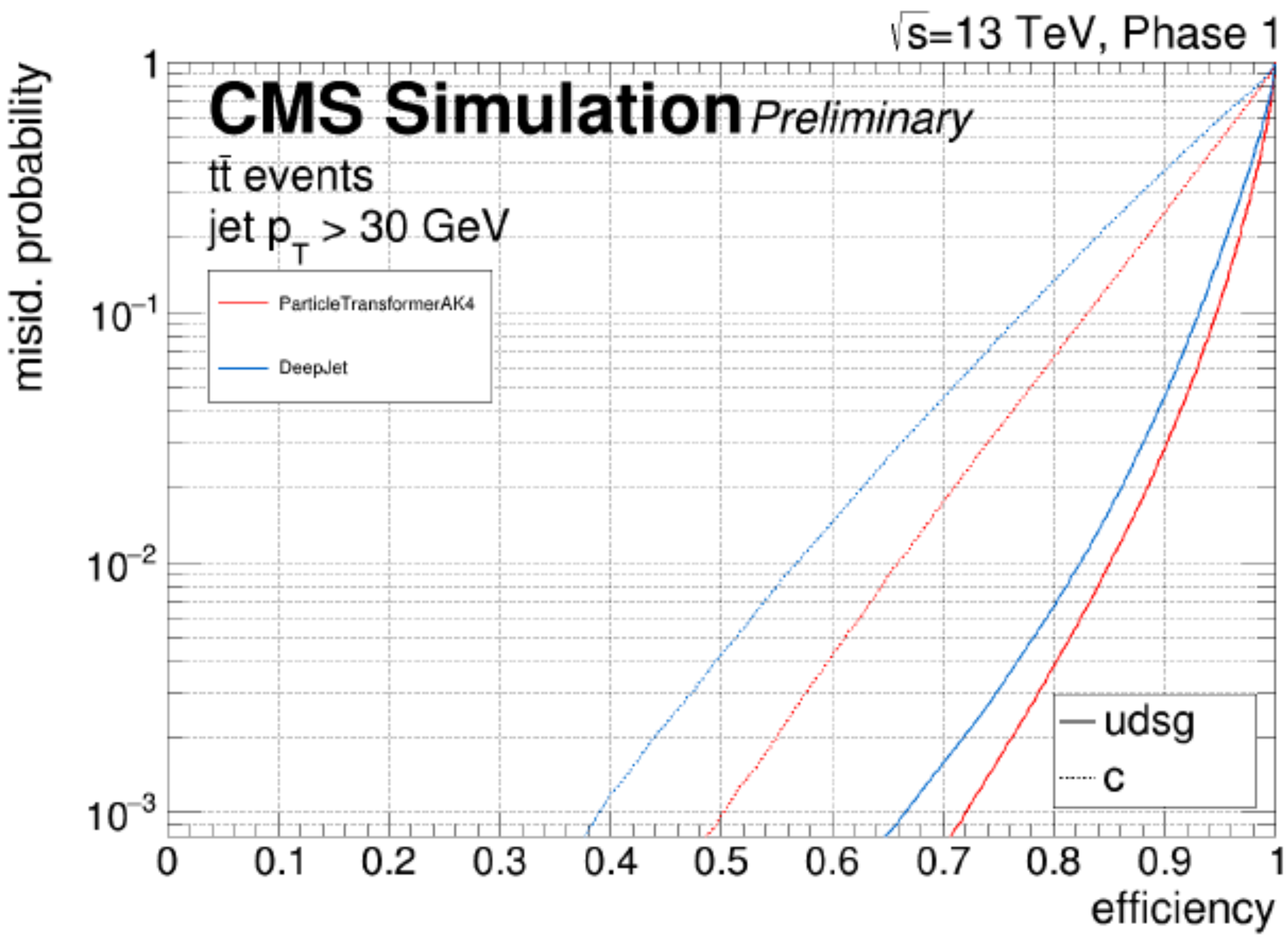
(a) Particle Transformer

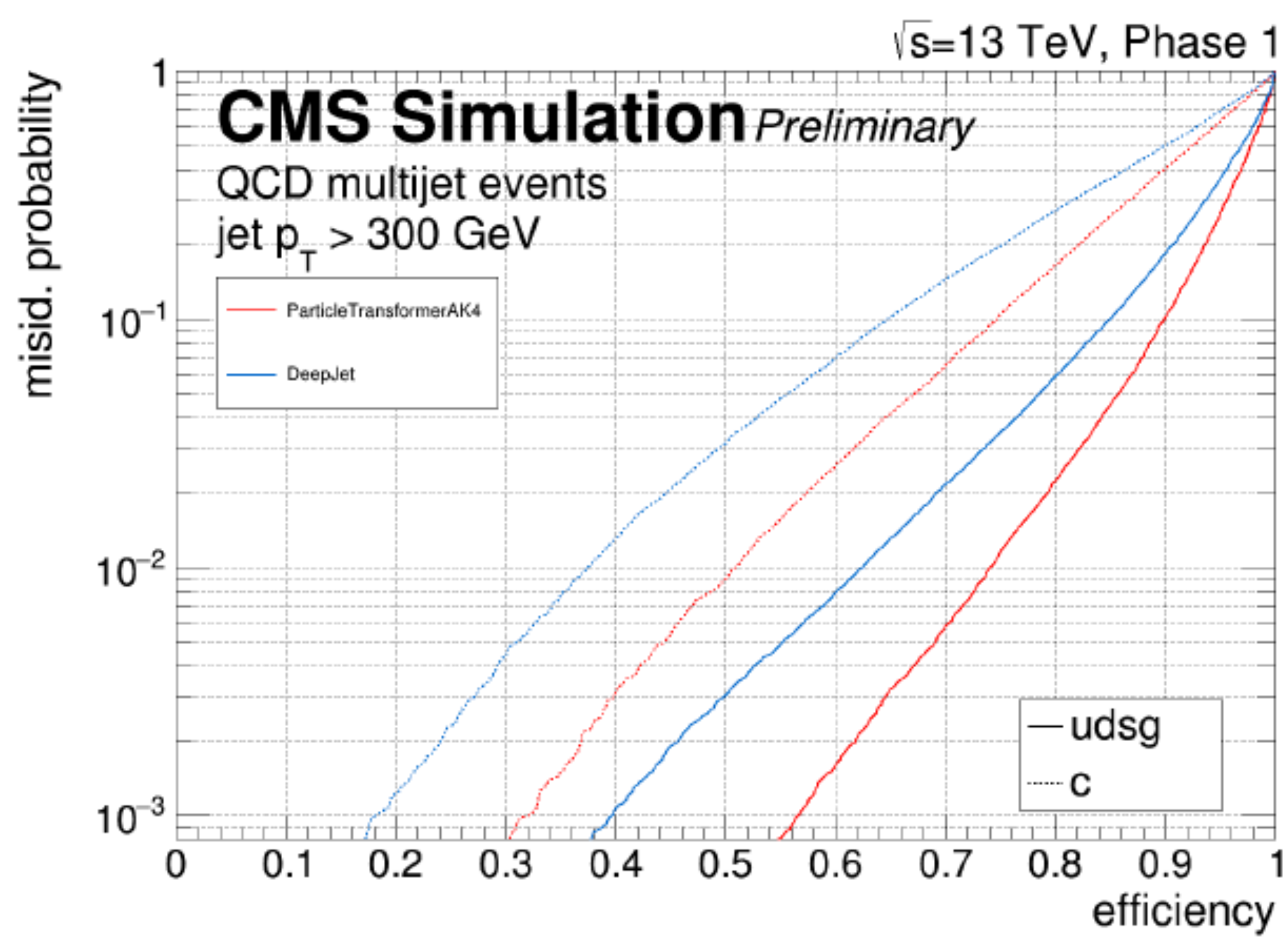
(b) Particle Attention Block

(c) Class Attention Block

ML for Jet Tagging - Particle Transformer for AK4 jets

CMS-DP-2022-050





Promising improvement in LF and c-jet rejection using Transform model over state-of-the-art DeepJet

ML for Jet Tagging - efficiency parameterisation

GNN approach to parametrise efficiency weights for each jet flavour/ each of the standard b-tag WP's

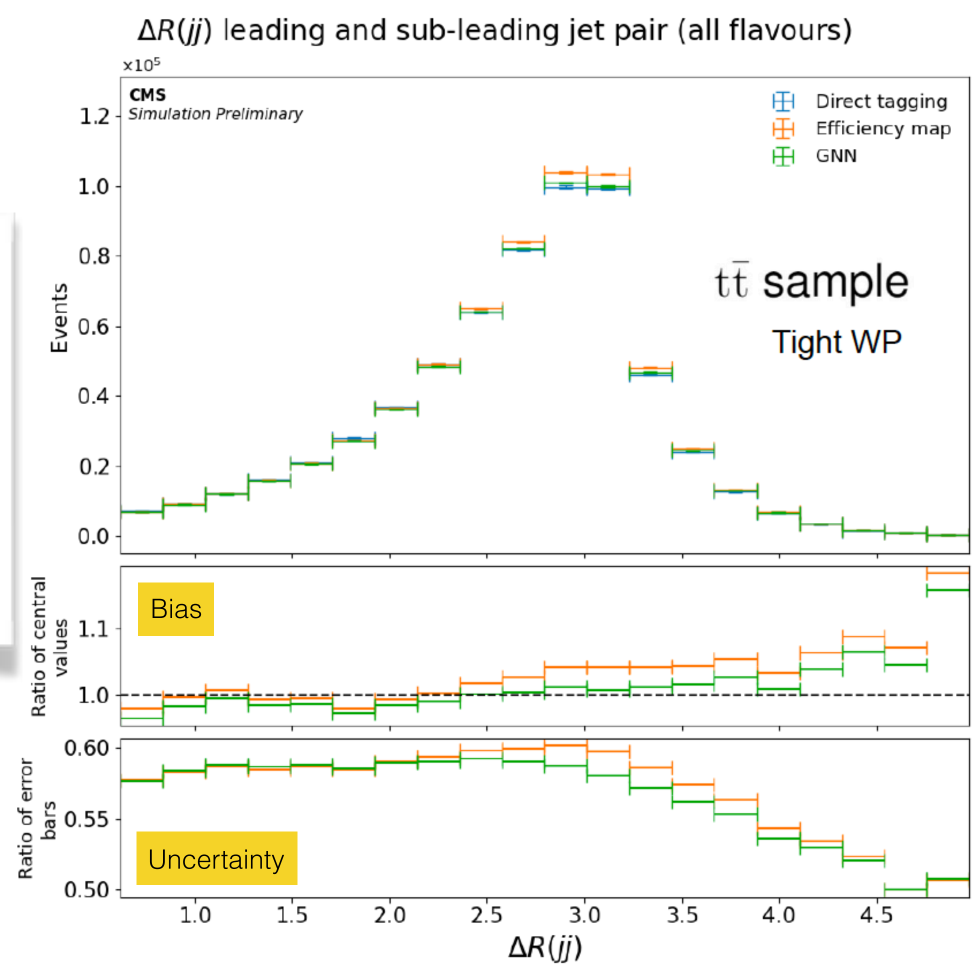
- takes full event as input and provides simultaneous efficiency weights for each flavour/WP - training performed on tt and QCD multiet simulation events
- approach captures higher-order correlations among jets associated to events and environment-related effects - using GNN with b-tagging observables as input features and ΔR between jets as edge feature
  - GNN training intrinsic uncertainty evaluated using bagging technique (bootstrap)

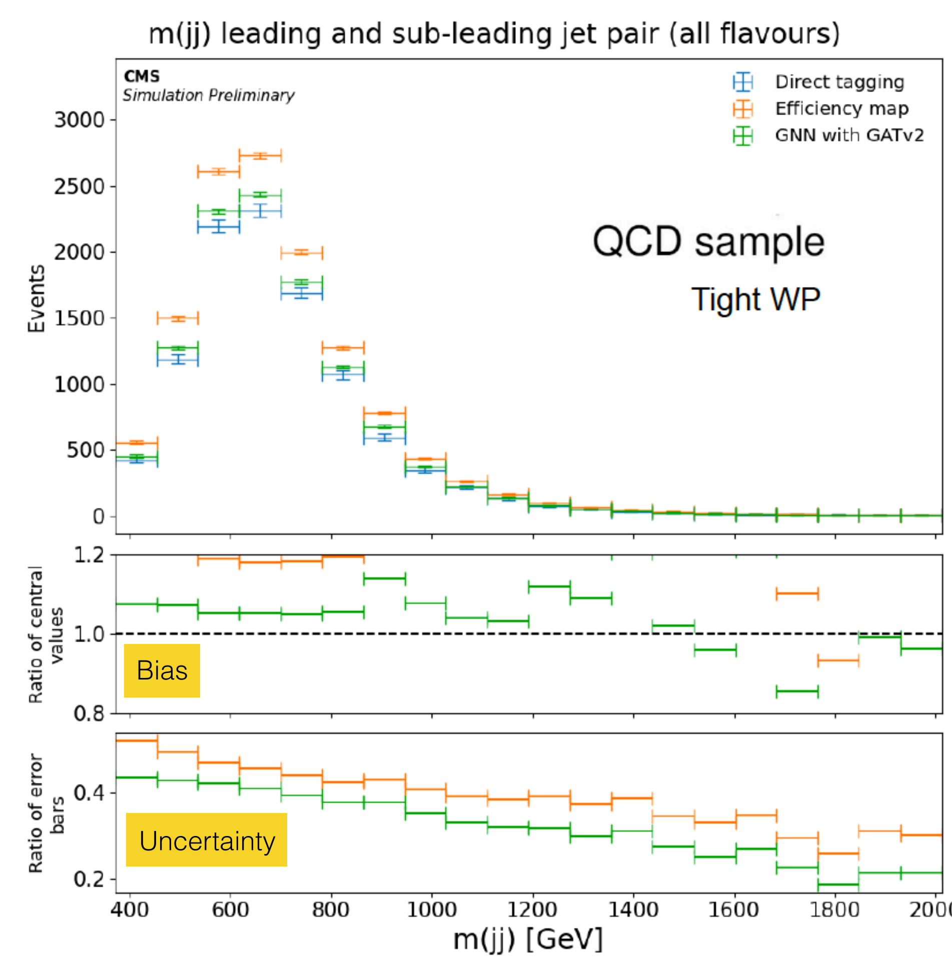
Performance evaluation using closure to direct tagging results and improvements in statistical uncertainty of efficiency parametrisation as figure of merits

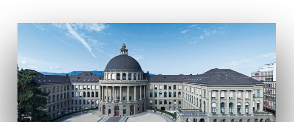
- using jet and multi-jet constructed observables, e.g mjj, ΔR(jj)

CMS-DP-2022-051

Successfully tackling simulation statistical uncertainty and limited size of simulated samples!

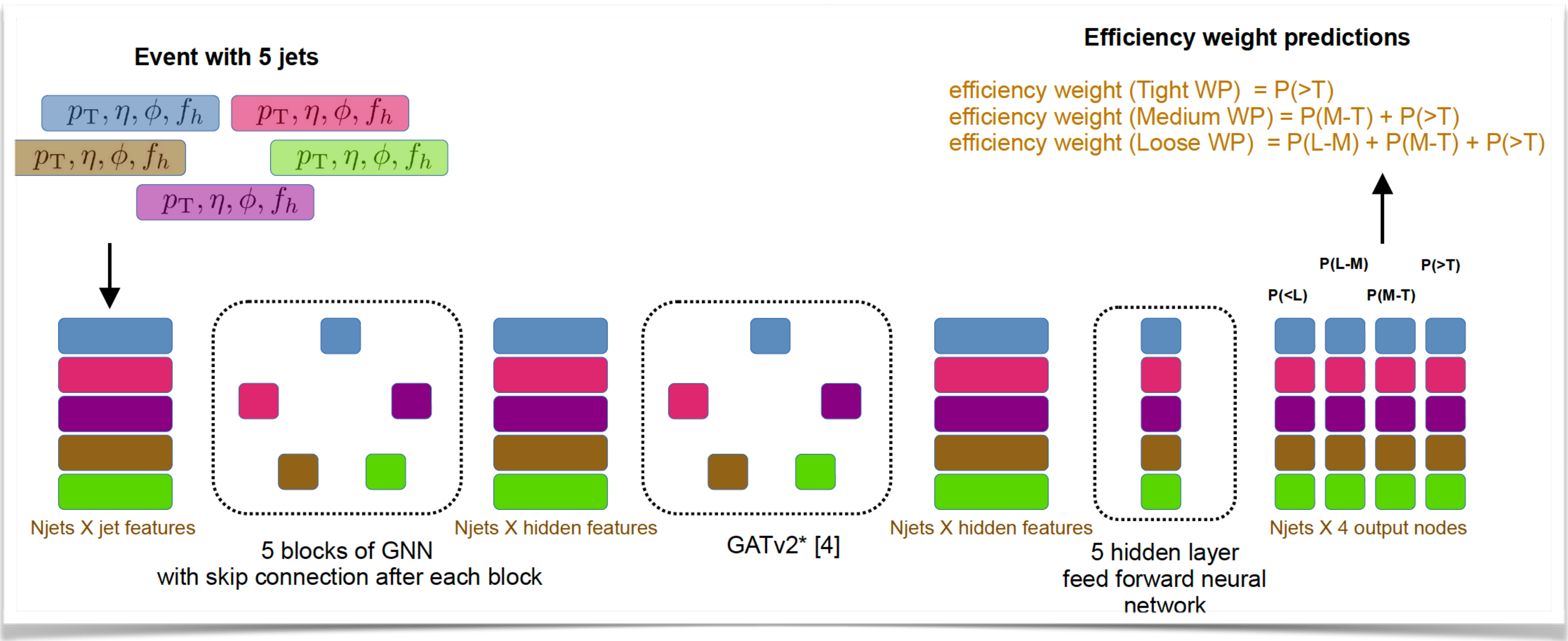






ETH zürich

ML for Jet Tagging - efficiency parameterisation (2)

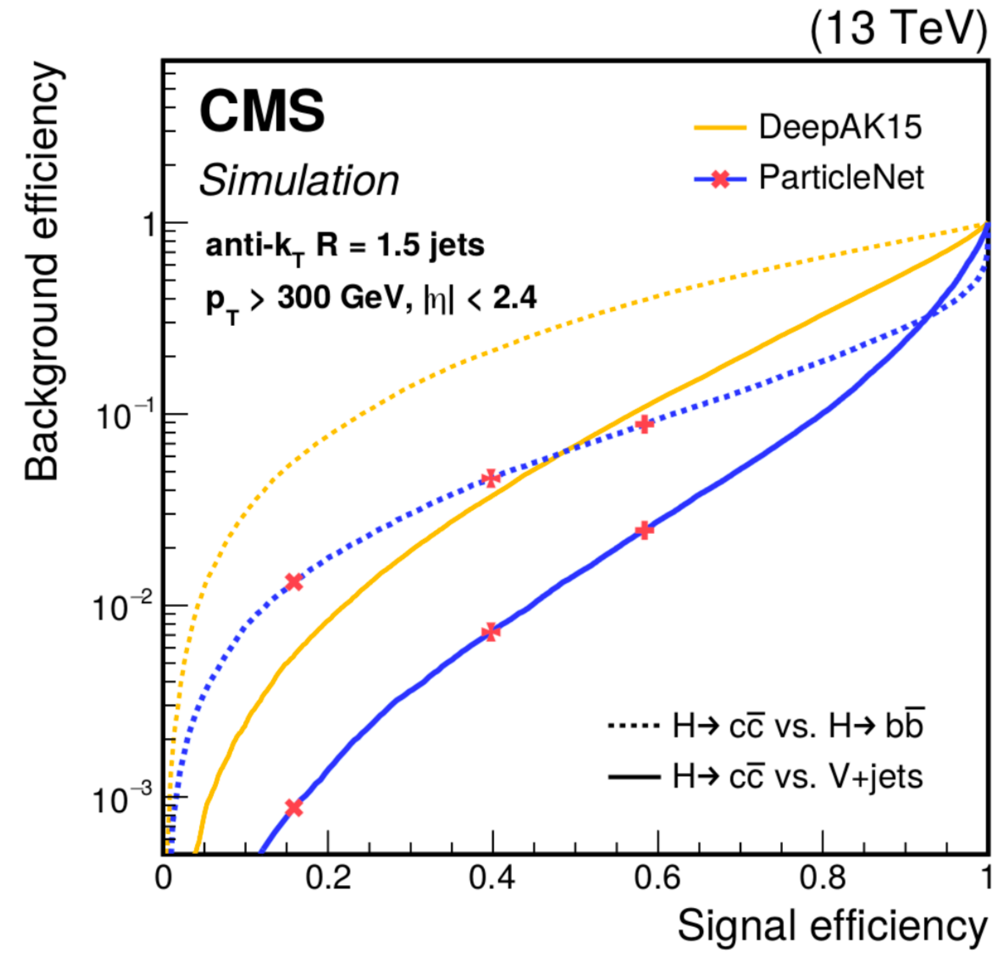


CMS-DP-2022-051

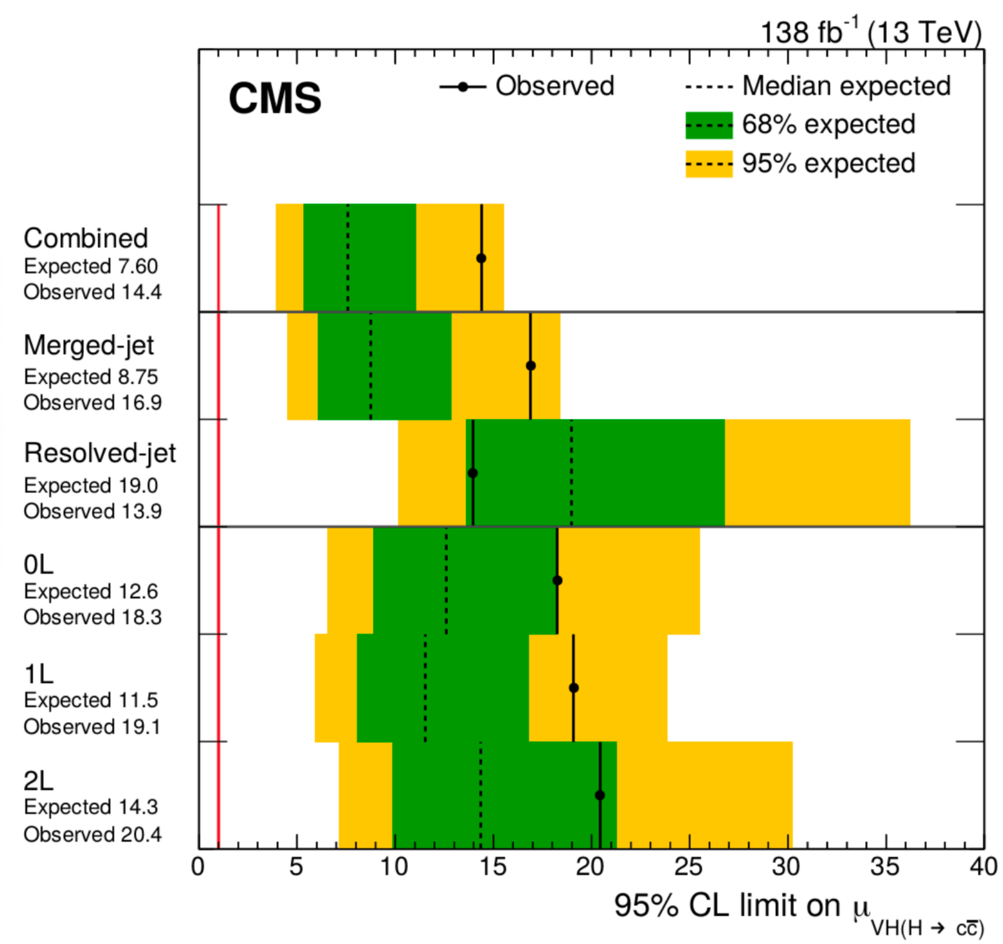
ETH zürich

ML for Jet tagging: the example of Run 2 CMS VH(->cc)

- ➡ Charm-jet identification algorithm (ParticleNet) largely enhances signal efficiency in VH(cc) boosted regime
- ➡ Resolved (AK4) and merged jet (AK15) topologies using DeepJet and ParticleNet taggers
- ➡ Cross-check VZ, Z→cc analysis: μ=1.01±0.22 → first observation of VZ, Z→cc at hadron collider



- ➡ Most stringent constraints on Higgs-charm Yukawa coupling at the LHC
- ➡ Novel flavour tagging ML for boosted topology has paid off - reaching sensitivity expected with larger dataset!

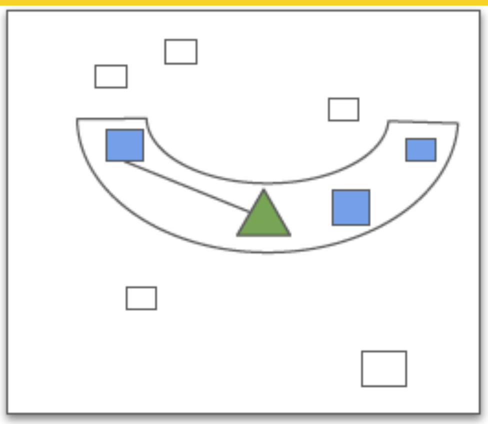


ML for ECAL - superclustering GNN for RECO

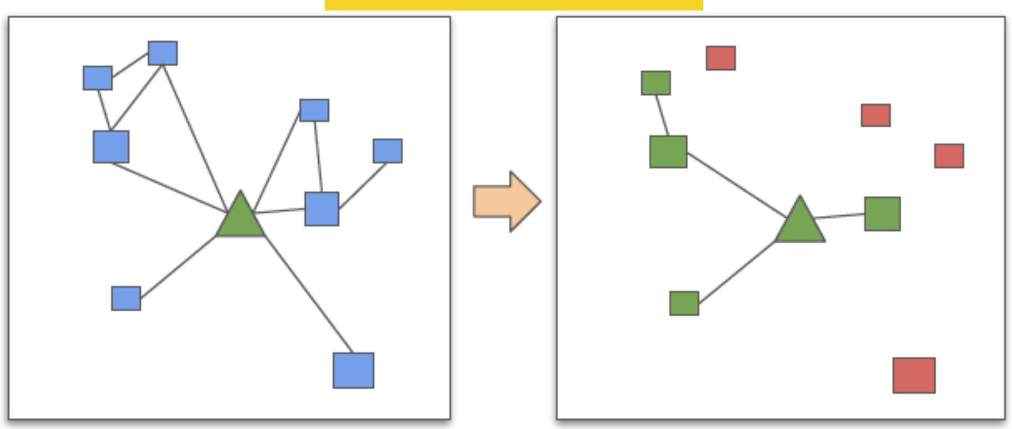
- Effort to improve the ECAL SuperClustering step (DeepSC): targeting replacement of current CMS ele/gamma reconstruction workflow: work on all the clusters in a region around the seed together
  - uses for ele/gamma reconstruction, ECAL calibration and input to PF algorithm
  - recover Bremsstrahlung or photon conversion and filter noise/PU on cluster-by-cluster basis
  - Architecture: graph-convolution network + attention layers using cluster and reconstructed hits as input features. Network target: cluster selection and window classification

CMS DP note

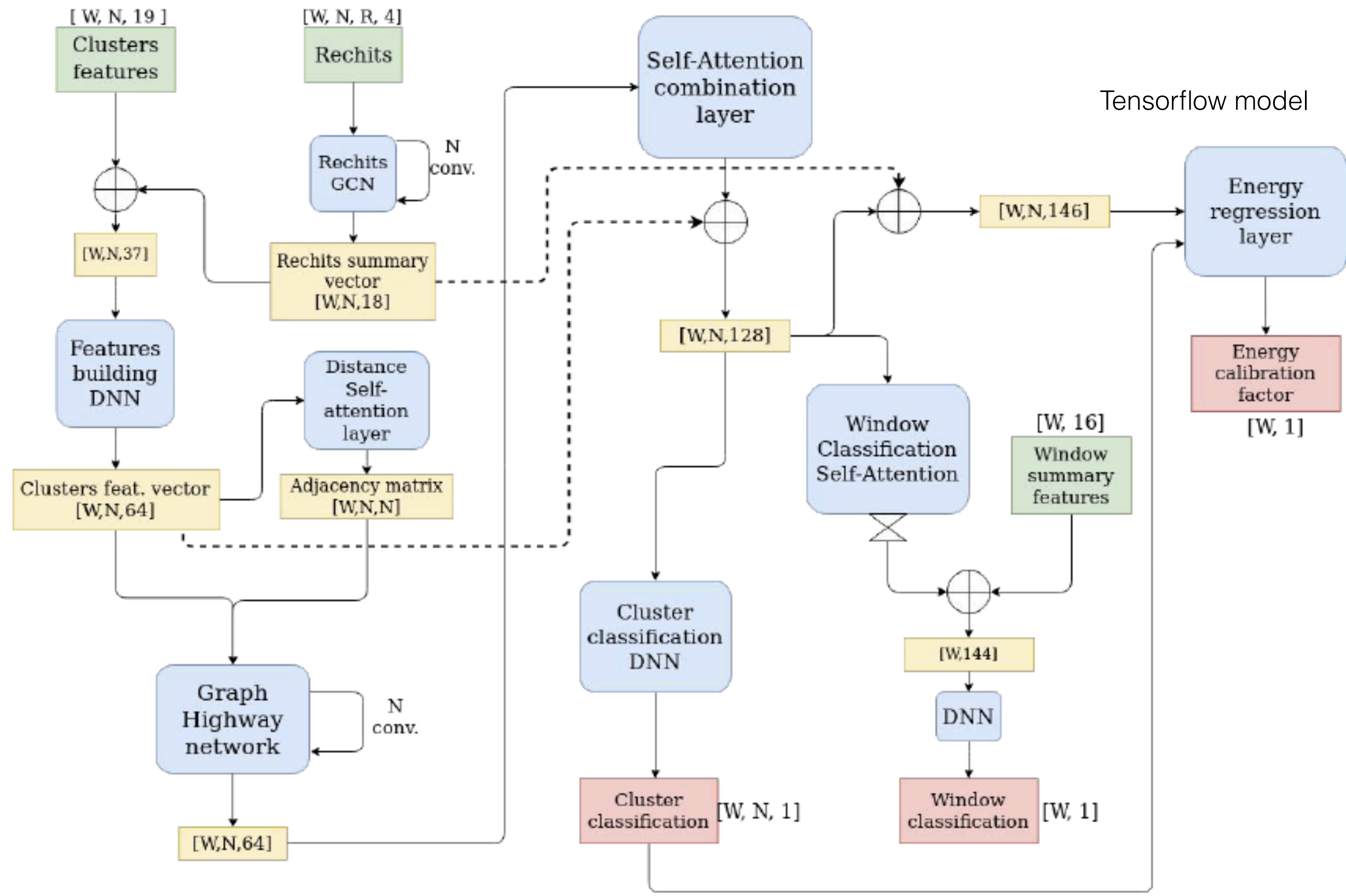
Old model (purely geometrical)



DeepSC model

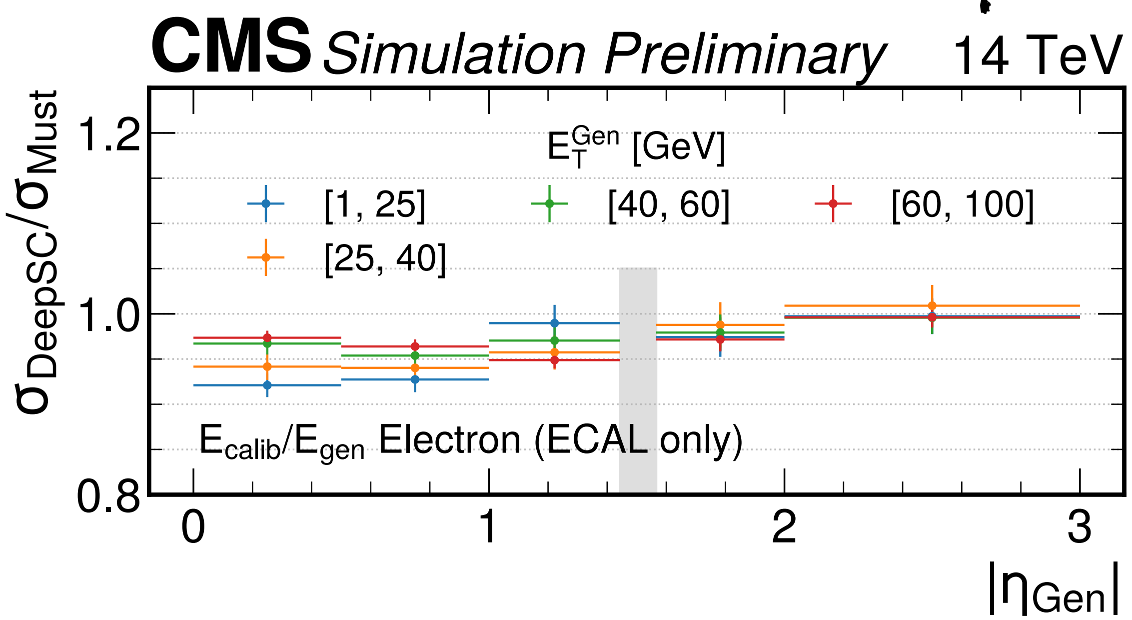


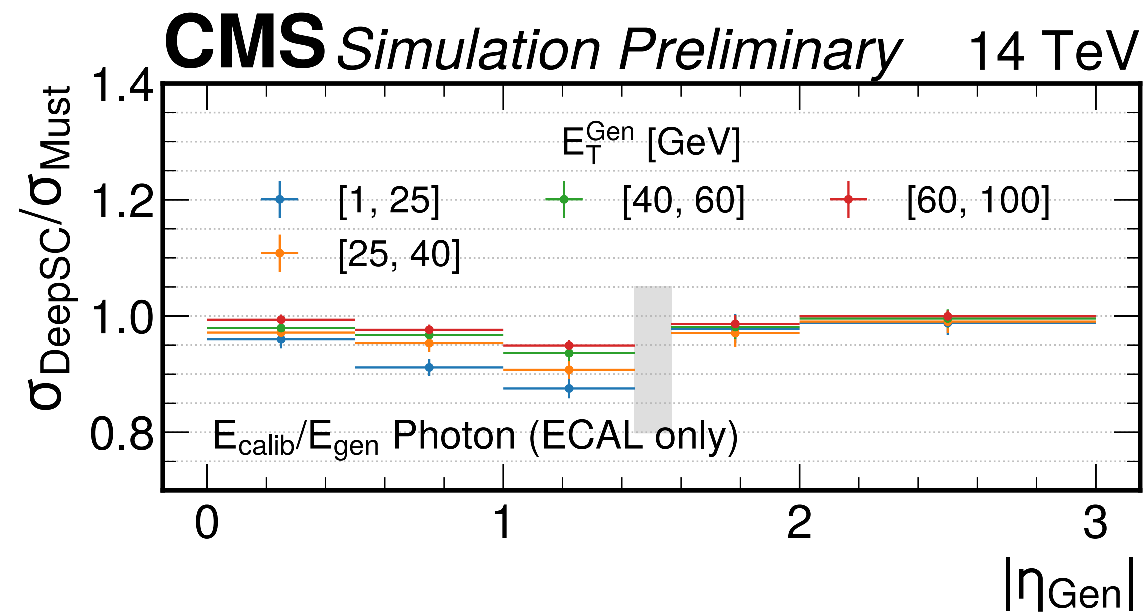
Tensorflow model

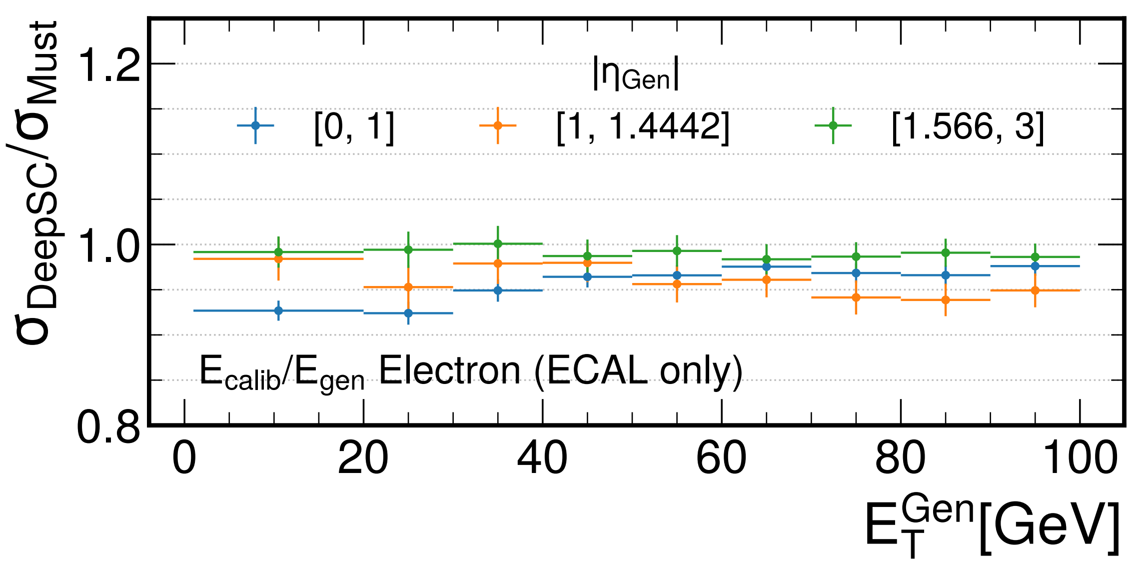


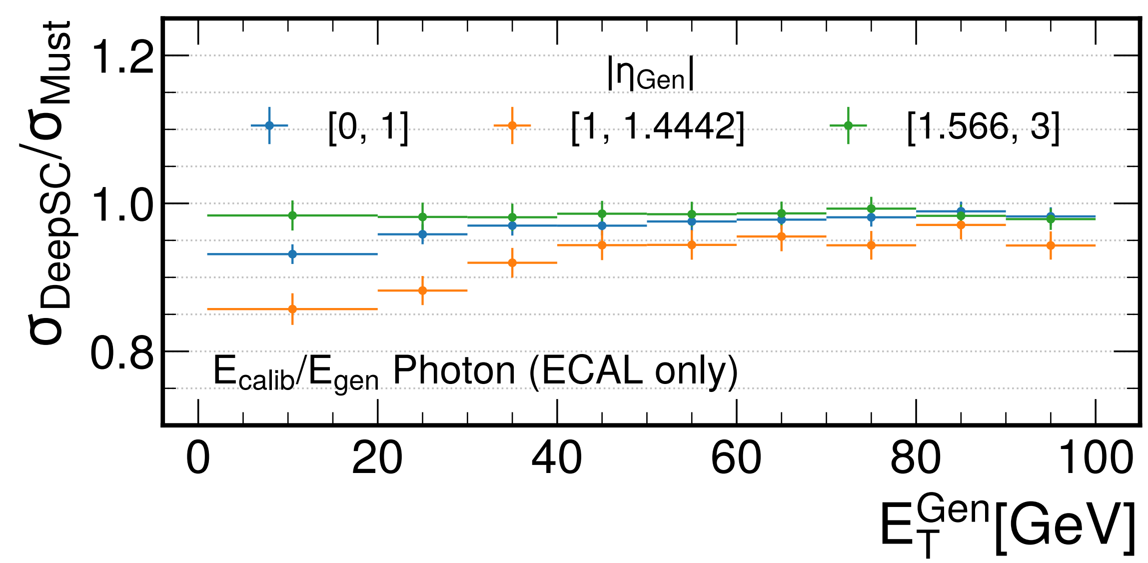
ETH zürich

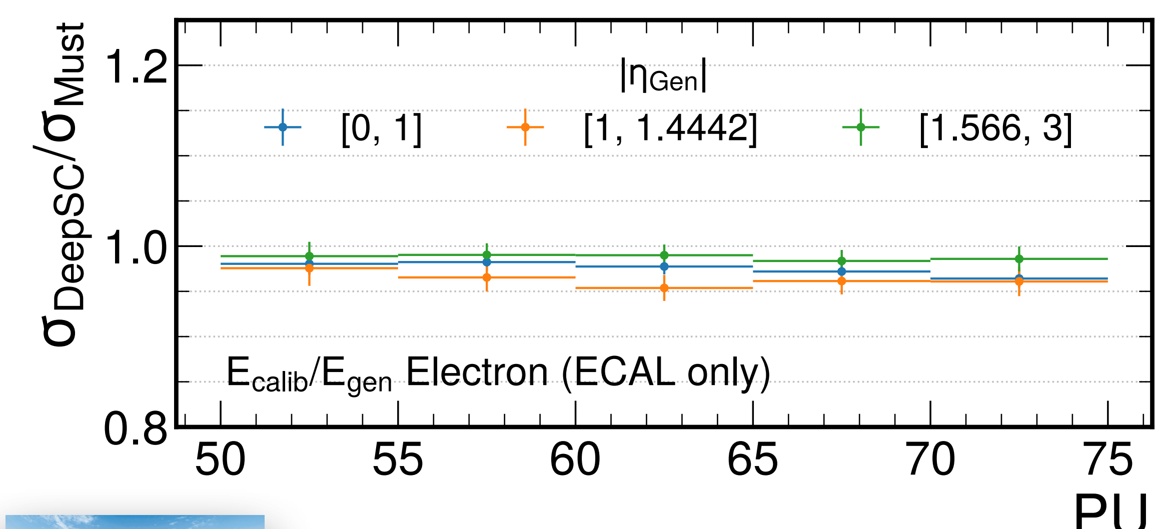
ML for ECAL - superclustering GNN for RECO (2)

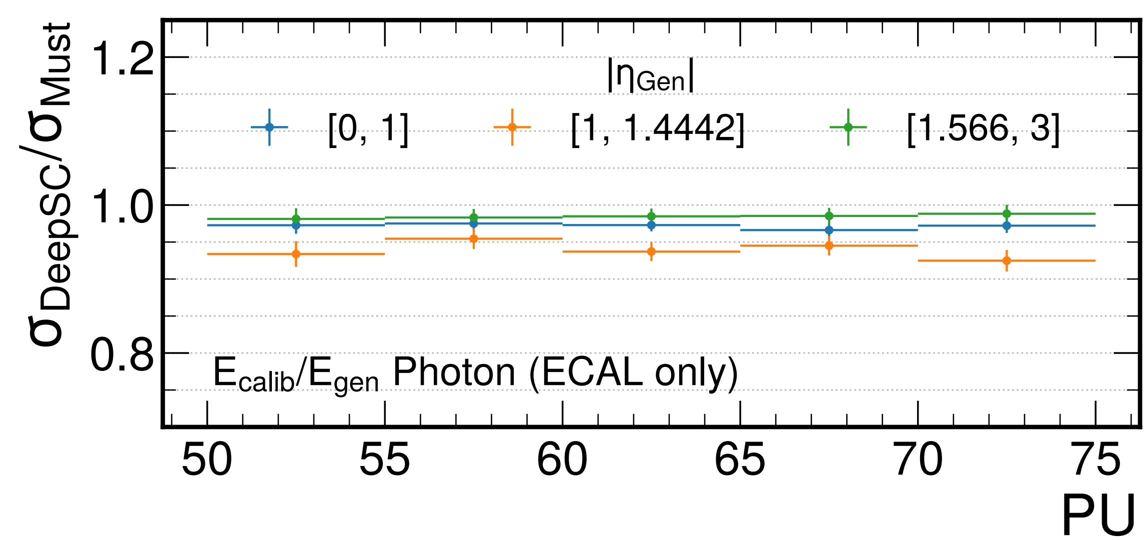


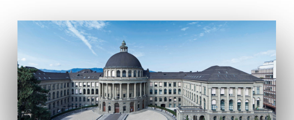












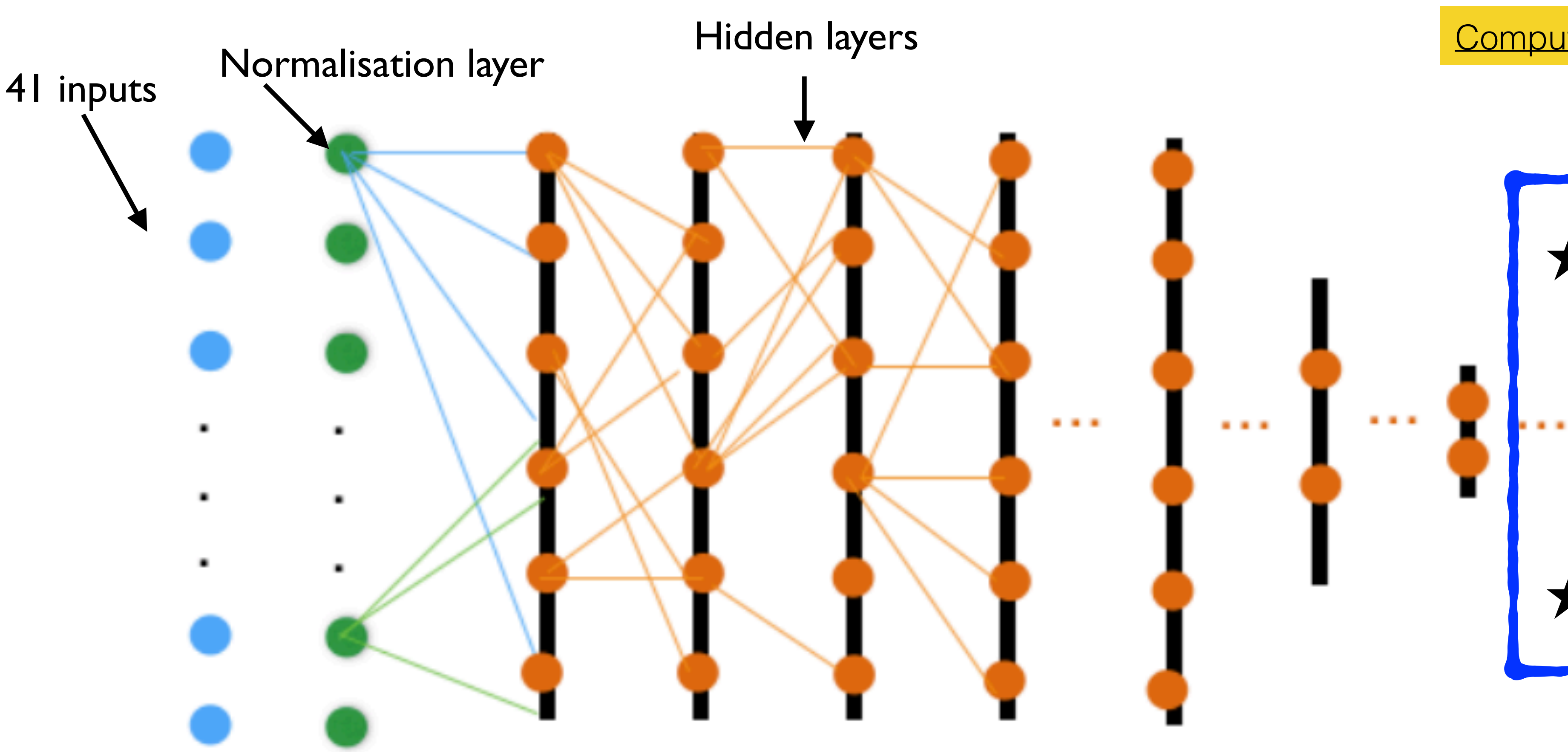
➡ Good improvement in the ele/gamma object resolution where the material budget is significant and in the low ele/gamma momentum region compared to geometrical clustering

ML regressions: b-jet energy regression

➡ b-jet energy regression corrects the b-jet energy scale and accounts for escaping neutrinos (semileptonic b-decays) - Feed Forward Fully Connected NN

- training on TT using as target ratio of truth (including neutrinos) over reconstructed jet momenta (jet energy corrections applied to training jets)
- basic inputs: jet kinematics, jet and soft-lepton tracks and secondary vertices, jet energy fractions, PU
- two outputs from regression: energy correction (Huber loss function) and resolution estimator (quantile loss function for 25% and 75% quantile)

Computing and Software for Big Science 4 (2020) 10



★ Hidden layers with Dense Unit, Batch normalisation (helps network convergence) and Drop-out Unit (for regularisation)

★ ReLu activation function used

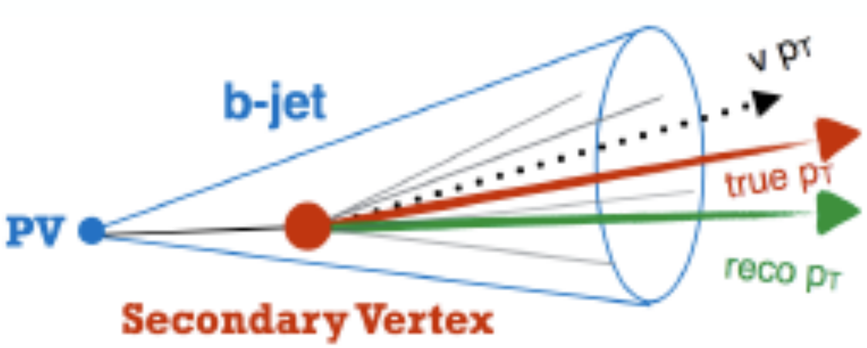
✓ Results on electron/photon regressions discussed later in the workshop agenda

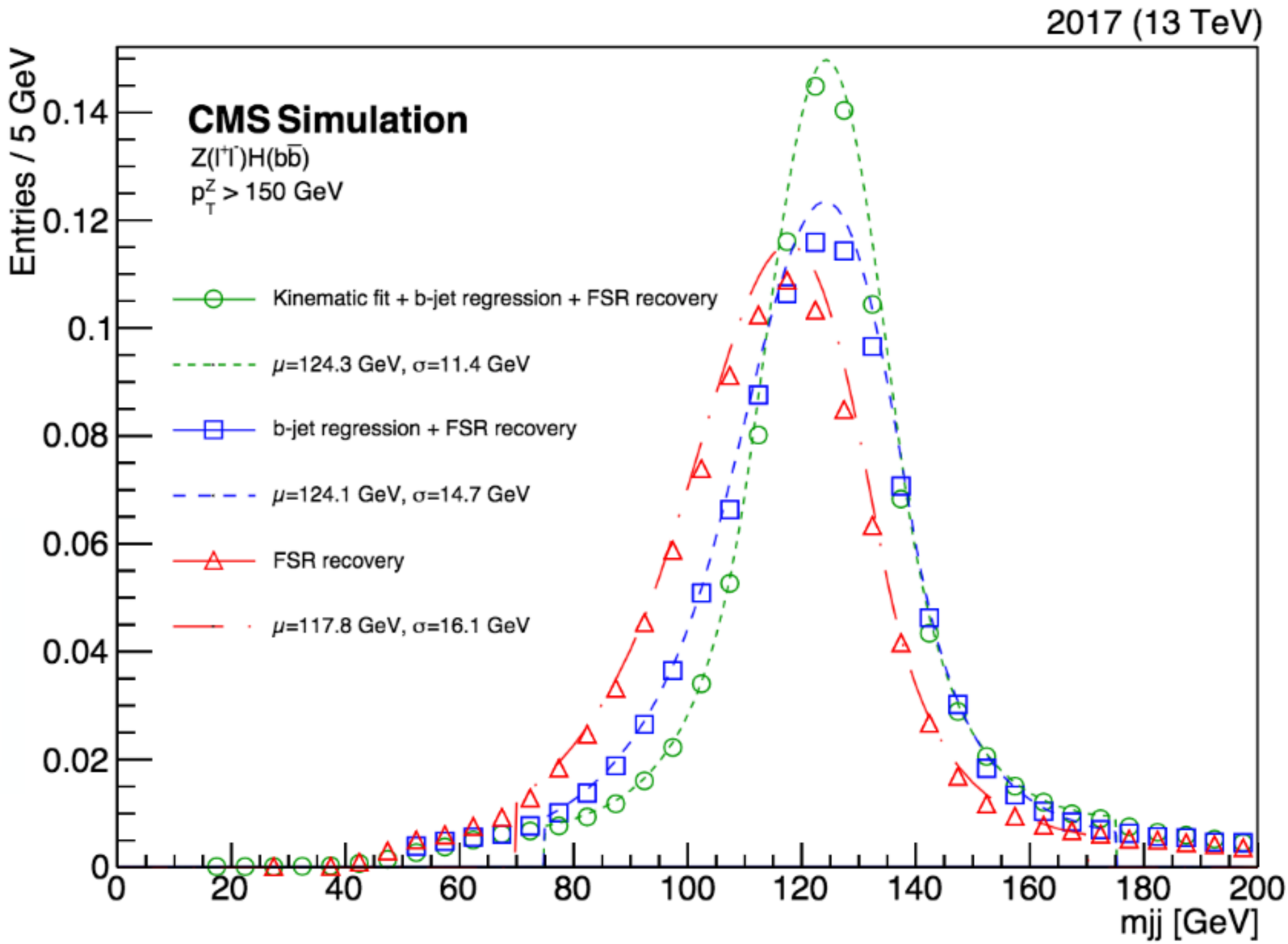
ML regressions: the example of Run 2 CMS VH(->bb)

➡ b-jet energy regression proved to be beneficial in several measurements with b-jets in the final state

- VH(→bb) measurement clear example of scale/resolution improvement due to b-jet energy regression
- training is agnostic of data/MC modelling - dedicated scale/resolution systematics needed

CMS-PAS-HIG-20-001

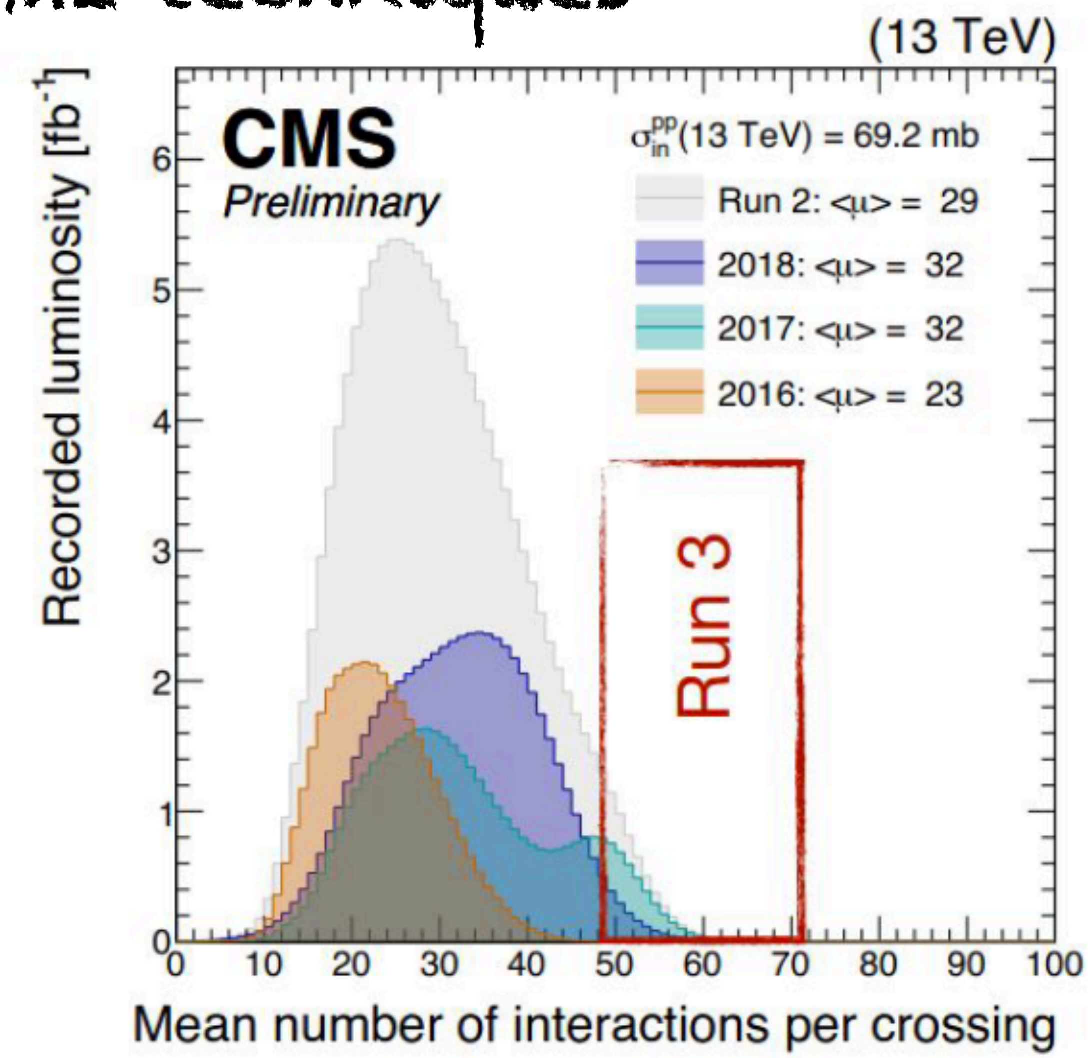


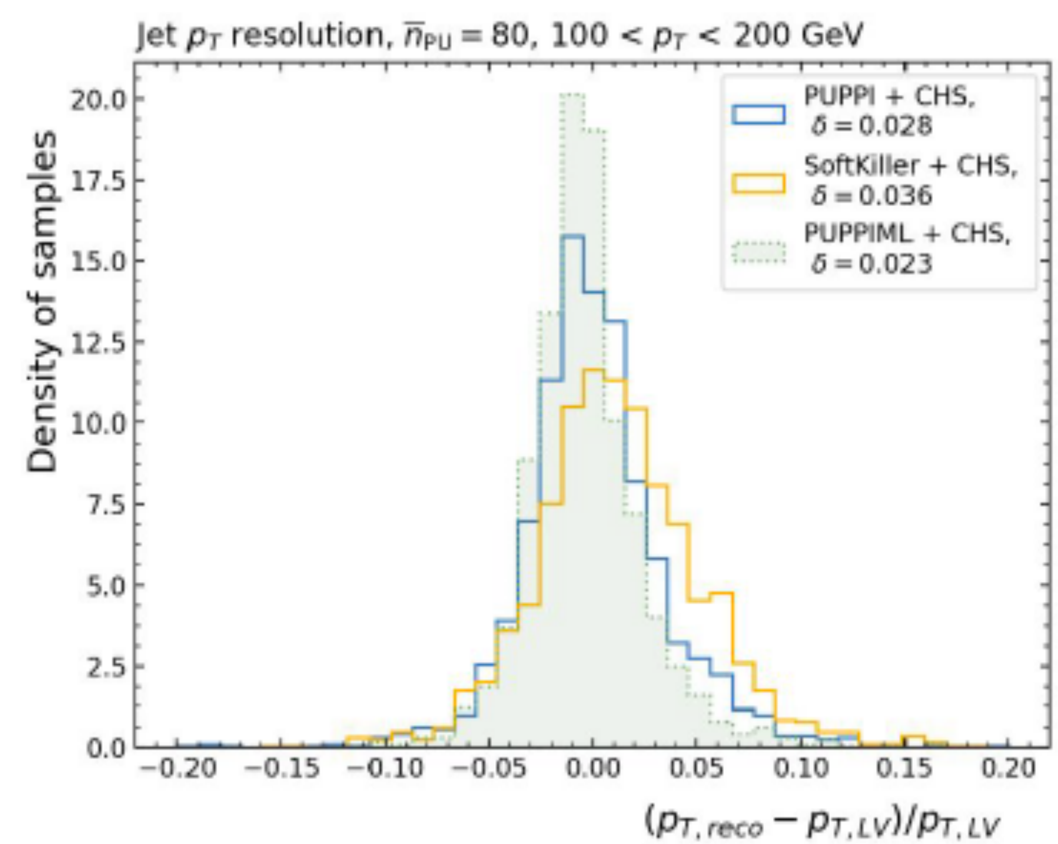


ETH zürich

Pile-up mitigation using ML techniques

- ➡ Tackling and mitigating pile-up is an obvious need of LHC experiments towards Run 3
  - additional soft momentum collision produced along with hard scattering degrades object resolutions and overall capability of physics measurements
- ➡ CMS techniques for pile-up mitigation based on selection on optimal observables or simple BDT's so far
  - ML approaches beyond selection-based PU removal
  - very promising improvements in observable resolutions event for high PU scenarios





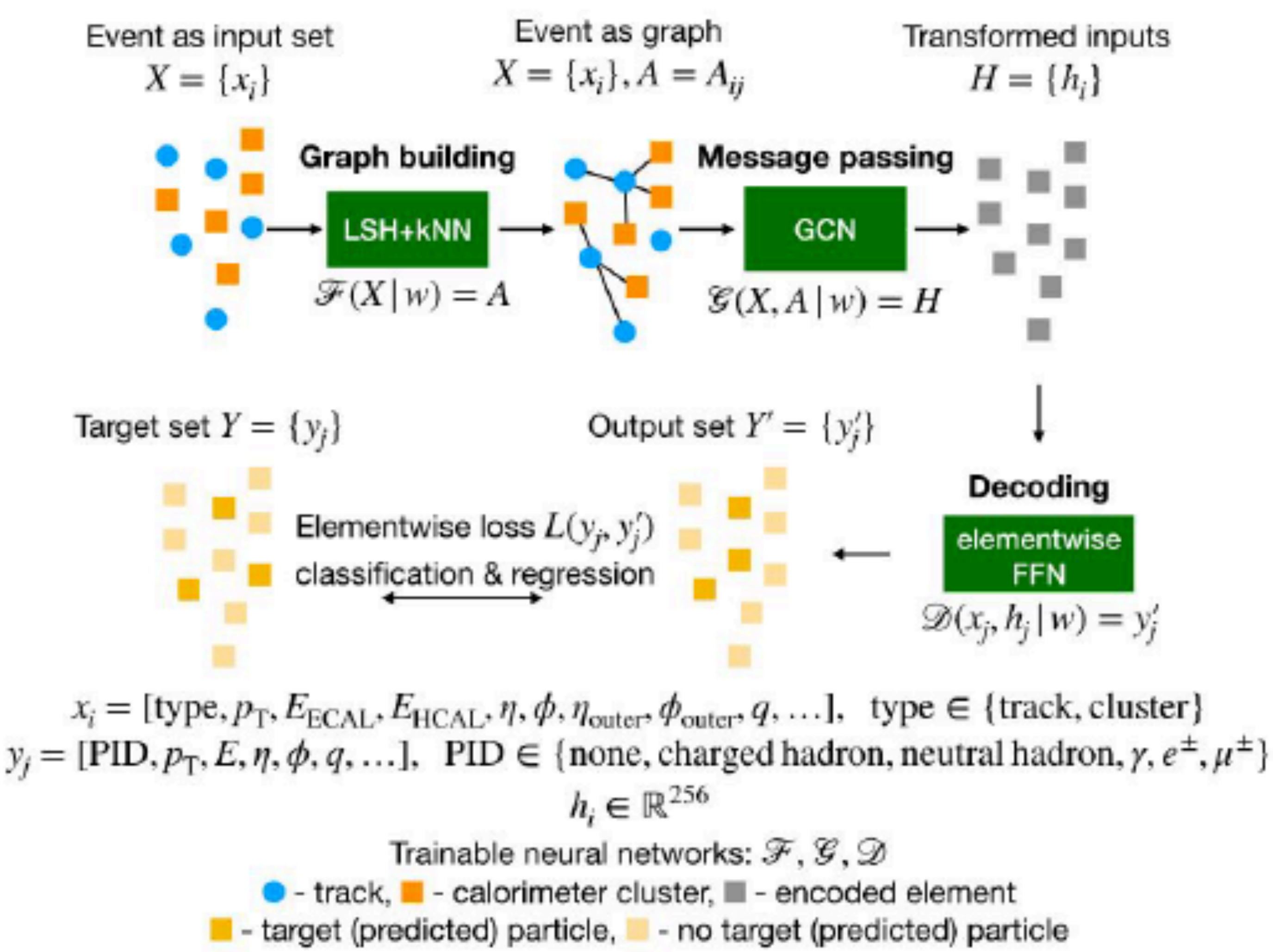
⟨PU⟩	20	80	140
SoftKiller	92.3%	92.3%	92.5%
PUPPI	94.1%	93.9%	94.4%
PUPPIML	**96.1%**	**96.1%**	**96.0%**

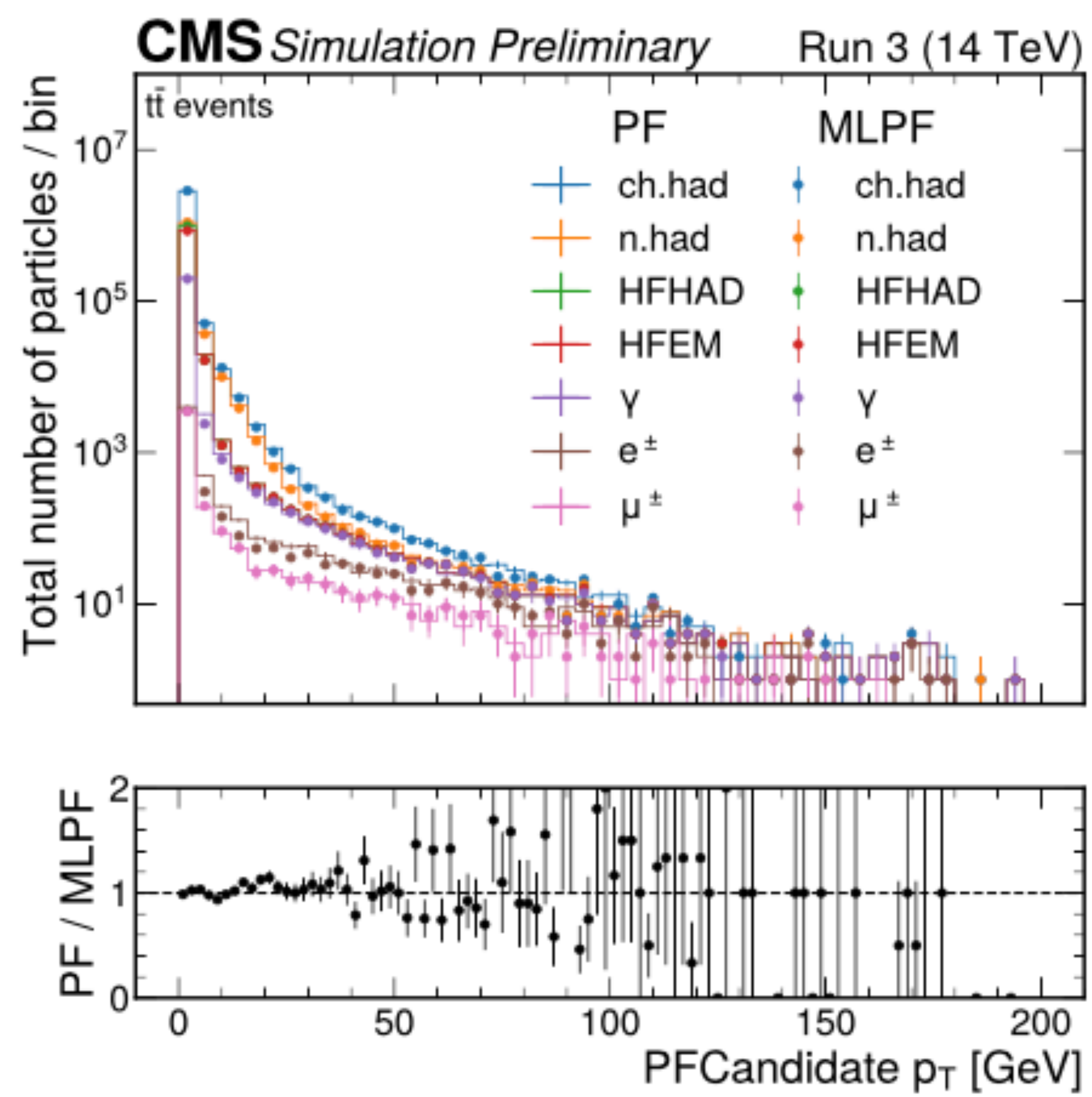
ML @ ParticleFlow

➡ Topical R&D effort in CMS, currently being implemented in CMS reconstruction workflow

- using all sub-detector, output list of PF candidates - running also particle ID and regressions
- using CombinedGraph layer (Transformer) technology: learnable embedding to form sub-graph; multiple graph-convolutions to propagate the information
- ongoing activities to define hyper-parameter optimisation and to assess performance on realistic CMS environment

J. Phys.: Conf. Ser. 2438, 012100 (2023)





Wrapping-up and conclusions

➡ **Several ongoing efforts using ML techniques for CMS object identification and reconstruction**

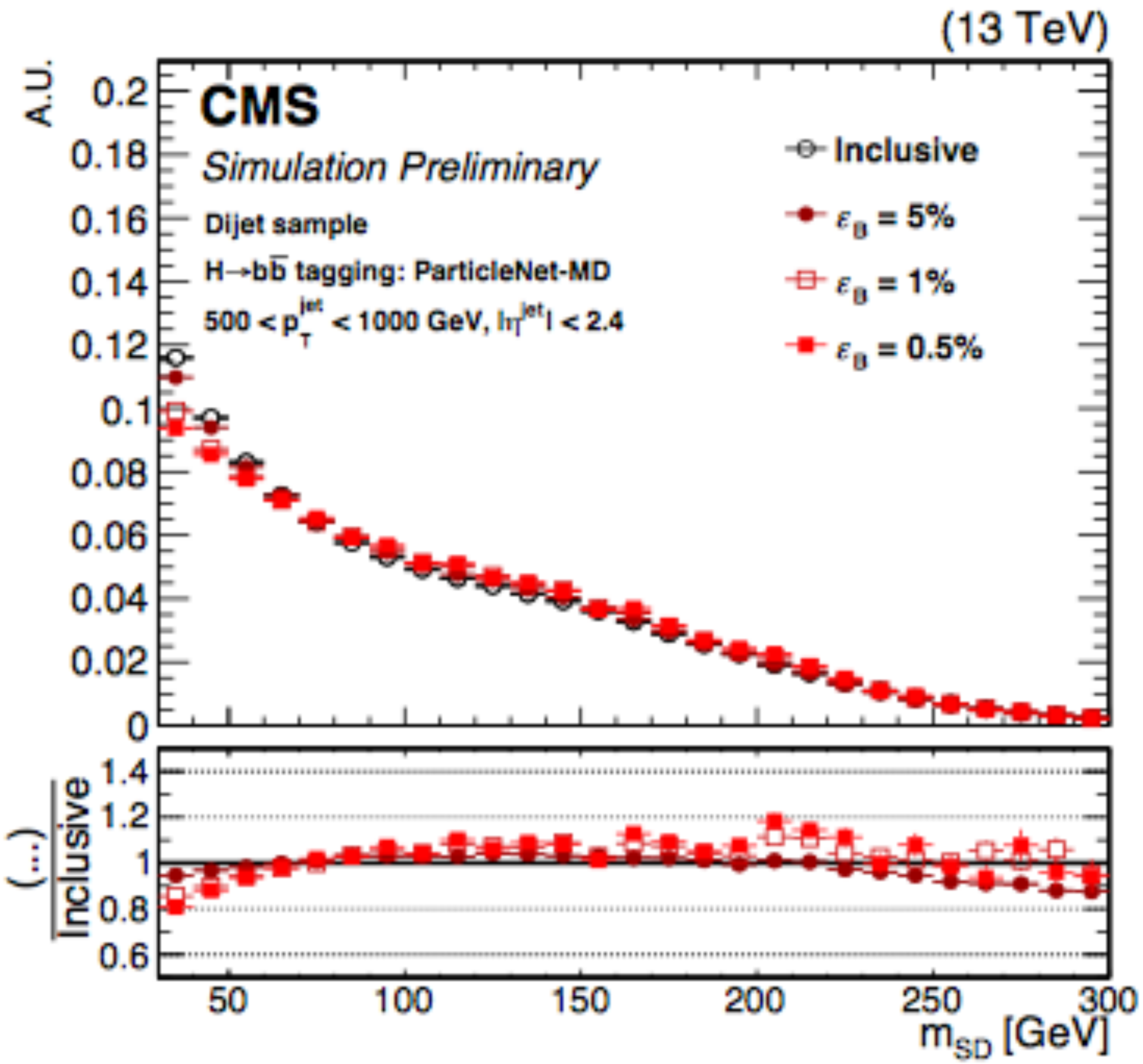
- most of these efforts have converged and have been successfully included in the CMS object reconstruction workflow and significantly impacted physics analyses
  - ML developments have definitely paid off: significant improvements on object ID/reco and subsequent S/B discrimination
- other efforts are in the development phase and will be fully commissioned for Run 3 analyses
- didn't cover additional ML-based efforts for tracking reconstruction, HGCAL reconstruction for HL-LHC environment

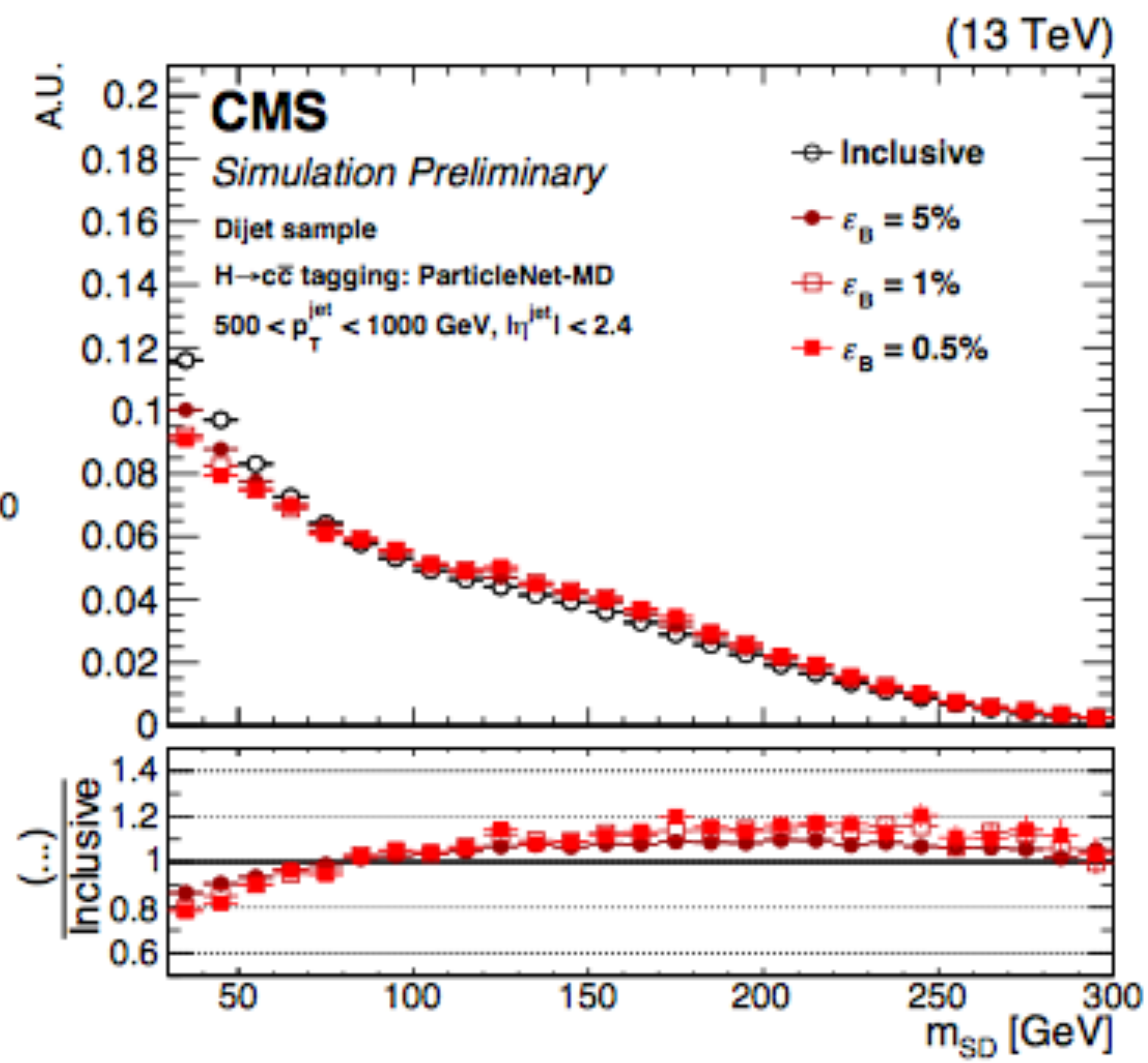
➡ **Open points to be addressed for further ML developments**

- <u>robustness</u>
  - performance of ML techniques vs data/MC modelling (SF determination); detector failure schemes, …
- <u>maintainability</u>
  - dedicated ML model retrainings, automatisation of ML retraining/evaluation steps and output validations, ….

Additional slides

ML for Jet Tagging - ParticleNet (mass decorrelation)





Efficiency measurements using GNN - results

➡ Performance evaluation using closure to direct tagging results and improvements in statistical uncertainty of efficiency parametrisation as figure of merits

- using jet and multi-jet constructed observables which are expected to carry environment effects parametrised by GNN, e.g mjj, ΔR(jj)

Tight WP

$t\bar{t}$ sample

χ^2	Efficiency map	GNN
p_T(j)	203.86	113.78
η(j)	350.01	103.53
ϕ(j)	145.34	61.81
m(j)	232.11	186.12
area(j)	105.30	85.79
m(jj)	22.16	11.71
ΔR(jj)	25.85	11.00

QCD sample

χ^2	Efficiency map	GNN with GATv2
p_T(j)	20.02	14.66
η(j)	50.11	13.31
ϕ(j)	24.67	18.12
m(j)	25.84	14.96
area(j)	15.67	7.15
m(jj)	22.55	6.81
ΔR(jj)	23.83	8.18

Medium WP

$t\bar{t}$ sample

χ^2	Efficiency map	GNN
p_T(j)	388.09	303.01
η(j)	5997.29	2441.21
ϕ(j)	192.82	153.69
m(j)	358.32	314.00
area(j)	199.80	174.52
m(jj)	48.52	26.17
ΔR(jj)	48.02	26.89

QCD sample

χ^2	Efficiency map	GNN with GATv2
p_T(j)	71.18	37.04
η(j)	731.08	67.73
ϕ(j)	34.24	20.29
m(j)	81.17	44.84
area(j)	36.52	17.12
m(jj)	24.72	6.56
ΔR(jj)	24.81	7.33

GNN architecture

$$
\begin{aligned}
\text{Node Input features } (\mathbf{v}_f) &= p_{\mathrm{T}}, \eta, \phi \text{ (azimuthal angle)}, f_h \text{ (jet flavour)} \\
\text{Embedding vector dimension for } f_h &= 2 \\
\text{Edge Input features } (\mathbf{e}_f) &= \Delta R \\
\text{Output classes} &= \text{4 DeepCSV WP categories (<Loose, Loose-Medium, Medium-Tight, >Tight)} \\
\text{train : val : test} &= 0.95 * 0.75 : 0.05 * 0.75 : 0.25 \\
\mathrm{GNN} &= \left(\text{five blocks with } d_{\mathbf{e}'_h} = 256 \text{ and } d_{\mathbf{v}'_h} = 512 \right), \\
\mathrm{GATv2} &= (\text{ eight heads with } d_{\mathbf{head}} = 64 \text{ and total output features per node} = 512), \\
\text{feed forward hidden layers} &= \{512, 256, 128, 50\}. \\
p^{\text{edge}}_{\text{dropout}}, p^{\text{node}}_{\text{dropout}}, p^{\text{GATv2}}_{\text{dropout}}, p^{\text{ffNN}}_{\text{dropout}} &= 30\%, 30\%, 10\%, 30\%
\end{aligned}
$$

GNN architecture

Node Input features $(\mathbf{v}_f) = p_{\mathrm{T}}, \eta, \phi$ (azimuthal angle), f_h (jet flavour)

Embedding vector dimension for $f_h = 2$

Edge Input features $(\mathbf{e}_f) = \Delta R$

Output classes = 4 DeepCSV WP categories ($<$Loose, Loose-Medium, Medium-Tight, $>$Tight)

train : val : test $= 0.95 * 0.75 : 0.05 * 0.75 : 0.25$

$\mathrm{GNN} = \Big($ five blocks with $d_{\mathbf{e}'_h} = 256$ and $d_{\mathbf{v}'_h} = 512\Big)$,

$\mathrm{GATv2} = ($ eight heads with $d_{\mathbf{head}} = 64$ and total output features per node $= 512)$,

feed forward hidden layers $= \{512, 256, 128, 50\}$.

$p^{\text{edge}}_{\text{dropout}}, p^{\text{node}}_{\text{dropout}}, p^{\text{GATv2}}_{\text{dropout}}, p^{\text{ffNN}}_{\text{dropout}} = 30\%, 30\%, 10\%, 30\%$