Grant PID2020-113341RB-I00 funded by:

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

CMS: Machine Learning

Andrea Trapote Fernández
(On behalf of the CMS Collaboration)

- ICNFP2023 -
10 – 23 July 2023

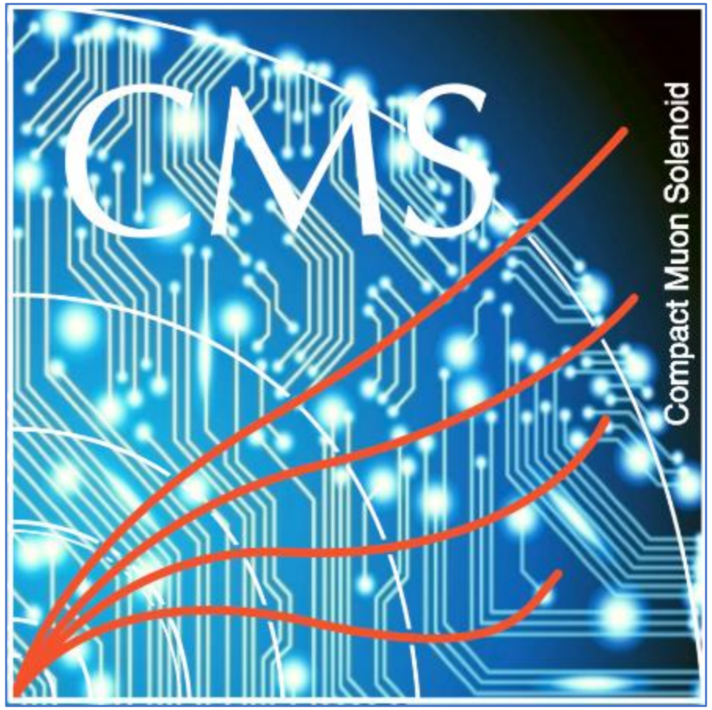
andrea.trapote.fernandez@cern.ch

Why Machine Learning in HEP?

- **Large amount of data** that need to be analyzed **quickly.**
- ML has high **accuracy and sensitivity** in searches for new particles and phenomena by distinguishing signal from background processes.
- **Anomaly detection** to detect rare or unexpected events that deviate from known physics processes.
- ML is **versatile** and can unify different strategies.
- It is in **continuous development** and promising techniques are appearing every day.

INTRODUCTION

- In CMS there is a wide variety of ML techniques used at different levels.
- The **CMS Machine Learning Group** is growing and manages all the ML techniques that are being developed and applied in the different subgroups.
- In this talk, I had to do a selection of relevant studies but… **there is a lot more ongoing!**



- Public Results
  - ML developments in POGs
    - JME
    - EGM / ECAL DPG
    - TRK
  - ML developments in PAGs
  - ML developments in O&C
  - ML developments in DQM/DC
  - ML developments in L1T /HLT
  - ML developments for HGCAL and PF
- Projects with Machine Learning

JME

Topic	Link	Date
Adversarial training for b-tagging algorithms in CMS	CMS-DP-2022-049	Oct 2022
Transformer models for heavy flavor jet identification	CMS-DP-2022-050	Oct 2022
Calibration of the mass-decorrelated ParticleNet tagger for boosted bb and cc jets with Run 2 data	CMS-DP-2022-005	March 2022
Identification of heavy, energetic, hadronically decaying particles using machine-learning techniques	cadi: JME-18-002, arxiv: 2004.08262, paper	June 2020
Neural-network-based displaced jet tagger	cadi: EXO-19-011, arxiv: 1912.12238, paper	May 2020
DNN for b-jet energy corrections and resolution	cadi: HIG-18-027, arxiv: 1912.06046	Dec 2019

EGM / ECAL DPG

Topic	Link	Date
ECAL DeepSC: Optimization of the DeepSC model inference strategy for reconstruction speedup	CMS-DP-2022-058	Nov 2022
ECAL DeepSC Particle ID	CMS-DP-2022-010	May 2022
ECAL SuperClustering with Machine Learning	CMS-DP-2021-032	Nov 2021

ML for JetMET Data Certification of the CMS Detector	CMS-DP-2023-032	May 2023
An AutoEncoder Based Anomaly Detection tool with a per-LS granularity	CMS-DP-2023-010	March 2023
An Autoencoder Based Online Data Quality Monitoring for CMS ECAL (2)	CMS-DP-2023-002	Feb 2023
An Autoencoder Based Online Data Quality Monitoring for CMS ECAL (1)	CMS-DP-2022-043	Oct 2022
Tracker DQM Machine Learning studies for data certification	CMS-DP-2021-034	Dec 2021

Topic	Link	Date
Continual Learning in the CMS Phase-2 Level-1 Trigger	CMS-DP-2023-022, twiki	May 2023
Performance of the ParticleNet tagger on small and large-radius jets at HLT in Run 3	CMS-DP-2021-035	Apr 2023
Performance of DeepJet b-tagging algorithms at HLT using in Run 3	CMS-DP-2023-018	March 2023
NN for the identification of bottom quarks in the CMS Phase-2 Level-1 trigger	CMS-DP-2022-021, twiki	June 2022
NN for Phase-2 Level-1 Trigger PV Reconstruction and Track to Vertex Association (2)	CMS-DP-2022-020, twiki	June 2022
NN for Phase-2 Level-1 Trigger PV Reconstruction and Track to Vertex Association (1)	CMS-DP-2021-035, twiki	Dec 2021

TRK

Topic	Link	Date
Performance of the track selection DNN in Run 3	CMS-DP-2023-009	March 2023

ML developments for HGCAL and PF

Topic	Link	Date
Progress towards an improved particle flow algorithm at CMS with ML	CMS-DP-2022-061	Nov 2022
High Granularity Calorimeter Reconstruction Results using a Graph NN	CMS-DP-2022-004	Jan 2022
Machine Learning for Particle Flow Reconstruction at CMS	CMS-DP-2021-030	Nov 2021

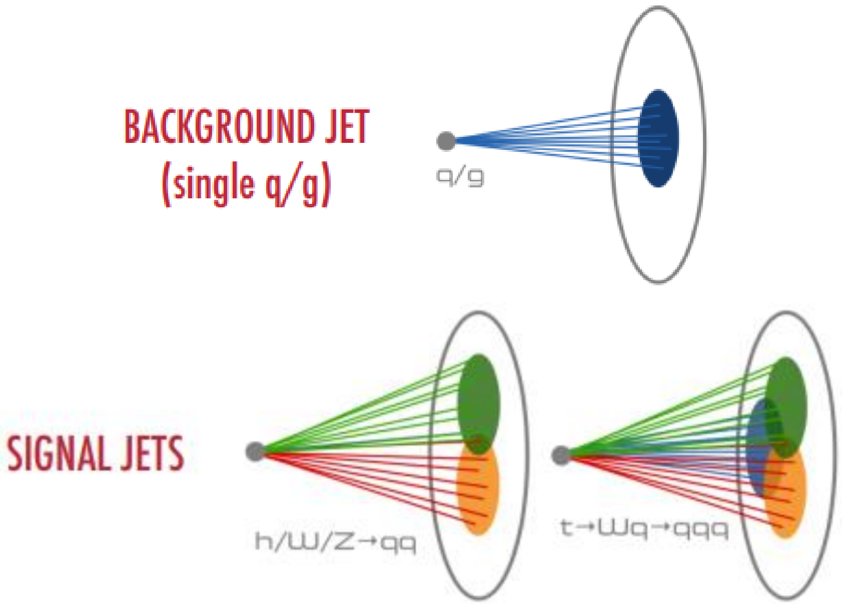
OBJECTS

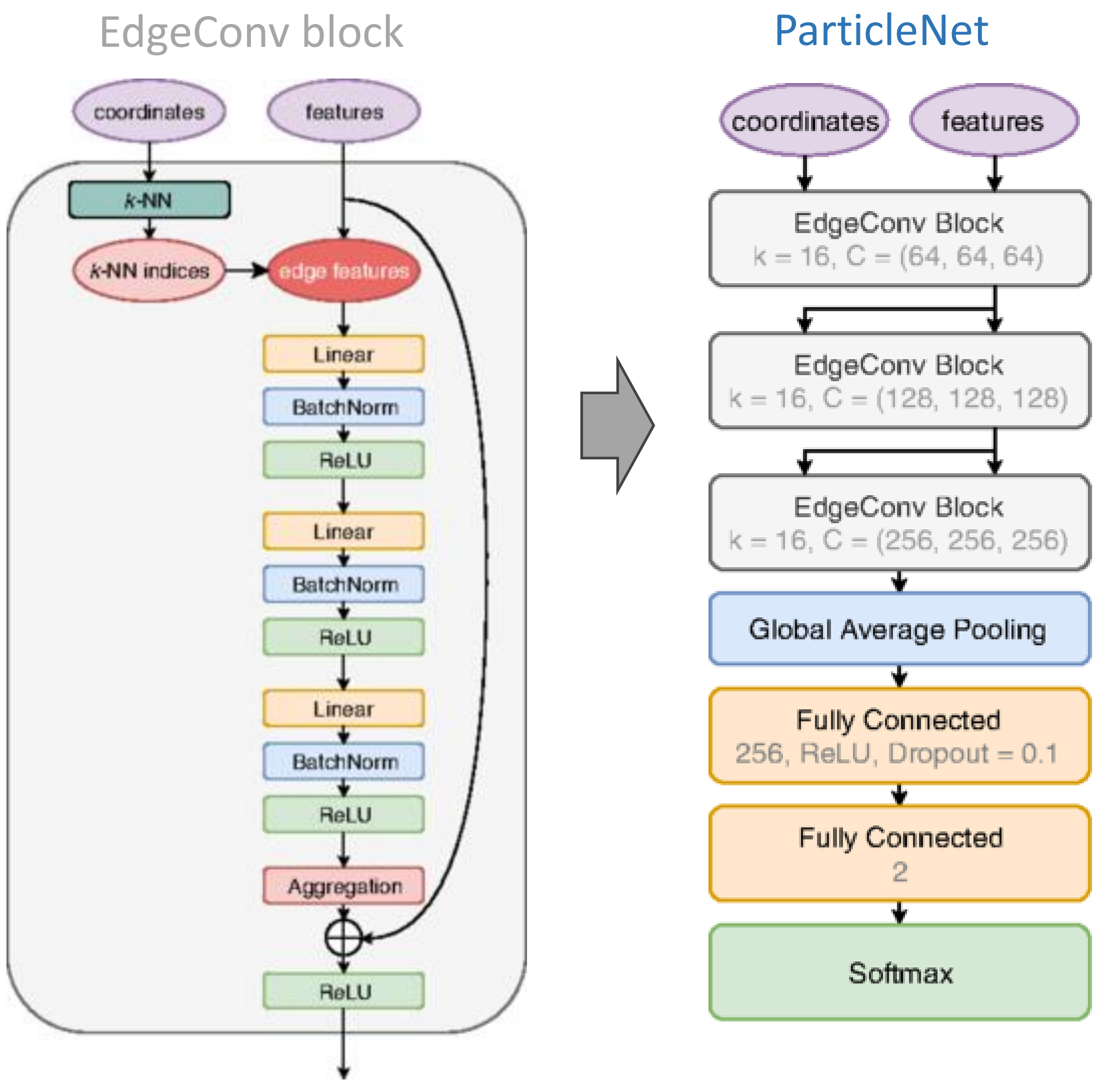
PARTICLENET FOR JET TAGGING

CMS-DP-2020-002 CMS-DP-2022-005

First graph-based tagger at LHC!

- Consider jets **as unordered set of particles** in space and use permutation-invariant **graph neural networks**.
- Jet ParticleFlow constituents and secondary vertices as input nodes, with set of features.
- Connect neighboring nodes to **learn relations** among constituents.
- Sample training jets uniformly in p_T/mass to avoid correlations with network output (MD).



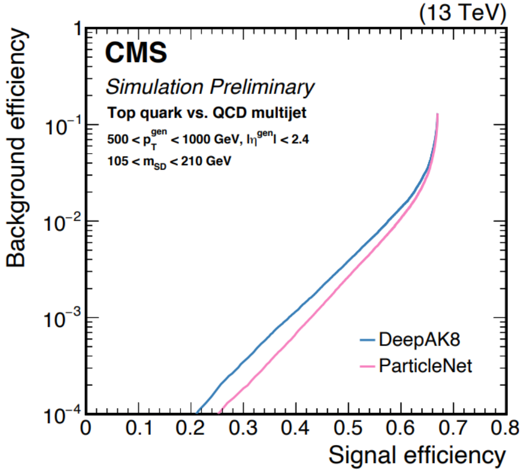


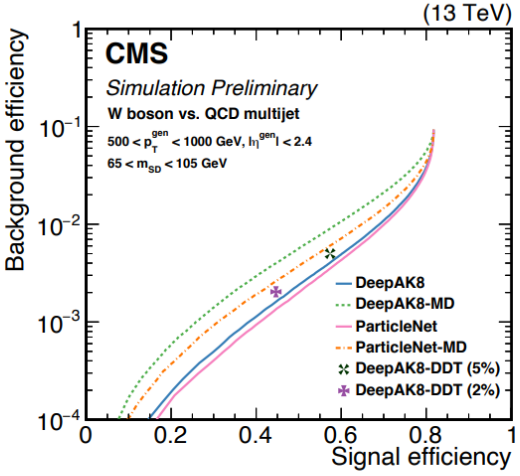
https://journals.aps.org/prd/abstract/10.1103/PhysRevD.101.056019

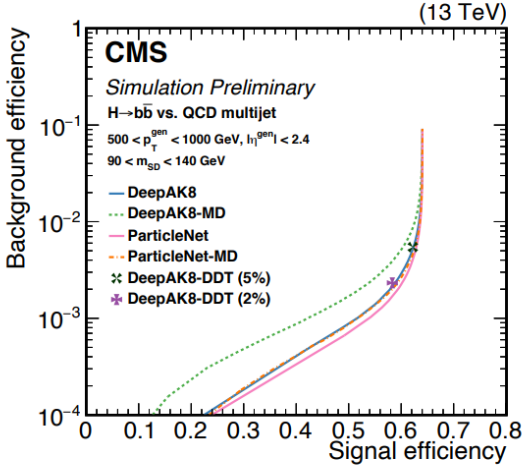
PARTICLENET FOR JET TAGGING

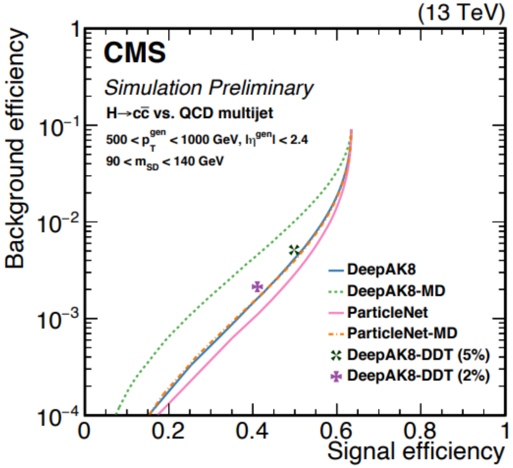
[CMS-DP-2020-002](#) [CMS-DP-2022-005](#)

➢ **Performanced check in different scenarios:**



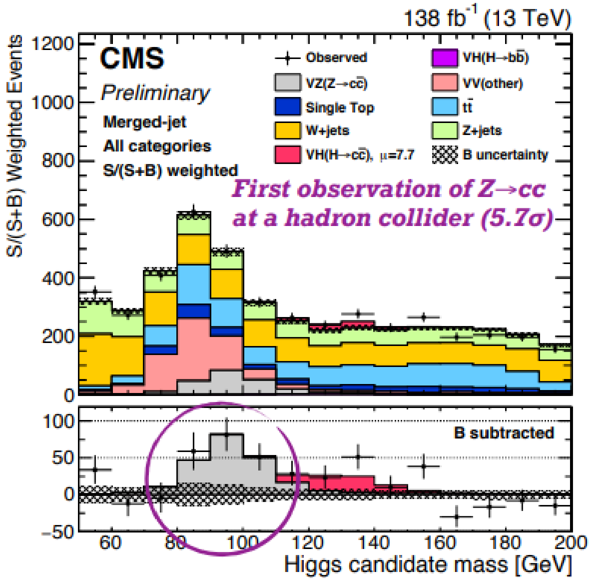


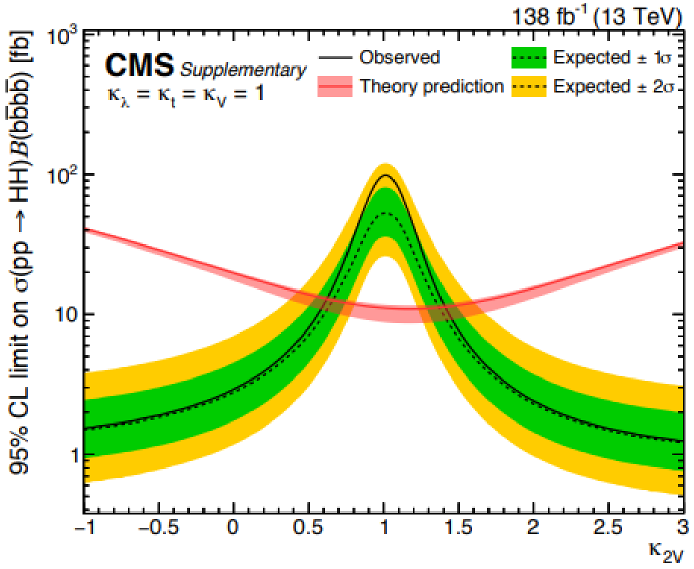




➢ **ParticleNet has been used in many analyses**

- [HIG-21-008](#): VH, H→cc search achieved contraints on y_c comparable to what had recently been expected at end of HL-LHC!
- [B2G-22-003](#): Exclusion of nonzero quartic VVHH coupling, k_{2V}, with significance >5σ.



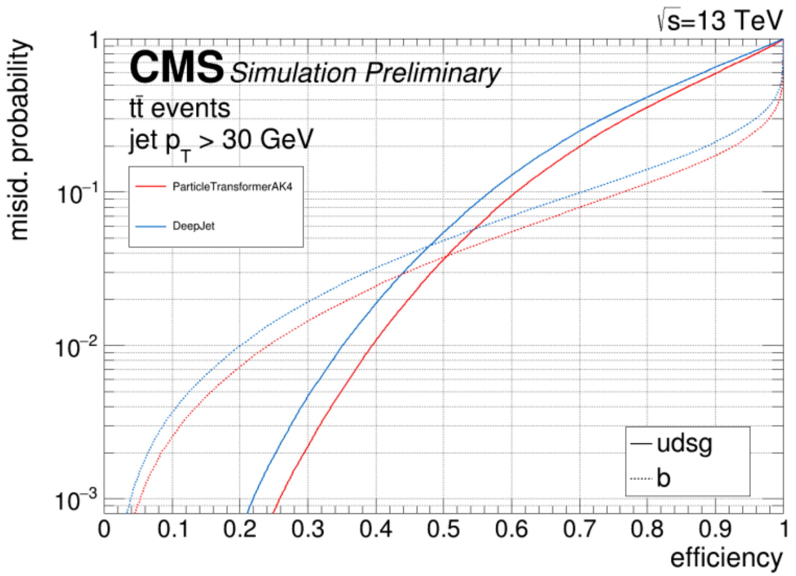


TRANSFORMER FOR HEAVY FLAVOR JET IDENTIFICATION

CMS-DP-2022-050

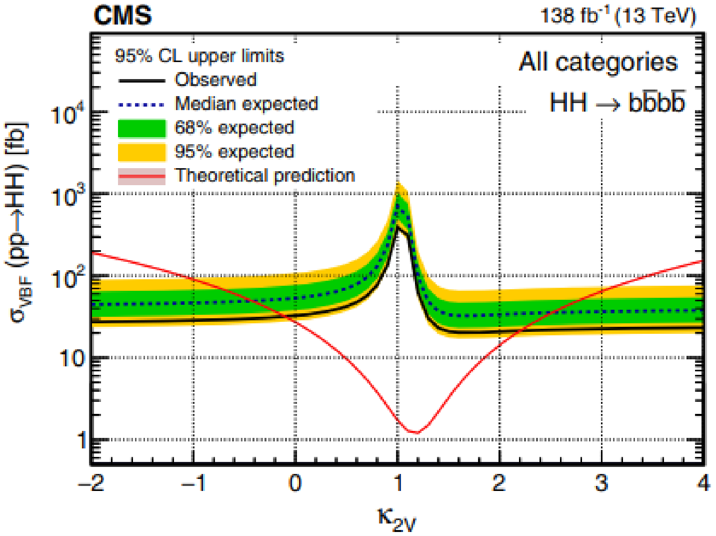
First attention-based tagger at LHC!

- As a step further for ParticleNet, there is a new **Deep Learning** algorithm that incorporates physics-inspired interactions in an augmented attention mechanism: **ParticleTransformerAK4.**
- **Model**: transformer model architecture. It contains a **tailored attention mechanism** involving the introduction of new pairwise features between all the jet constituents and secondary vertices.
- ParticleTransformerAK4 can better learn and understand the **internal structure of a jet** **improving the performance compared to the current state-of-the-art model, DeepJet.**



- **Application of more powerful ML architectures** in heavy flavor identification allowed recently setting the most stringent constraints on HH production!

  - HIG-20-005: Setting the most stringent constraints on HH production in the four b-quark final state.

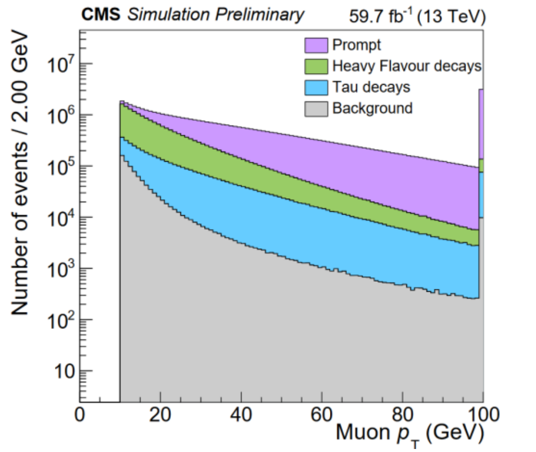


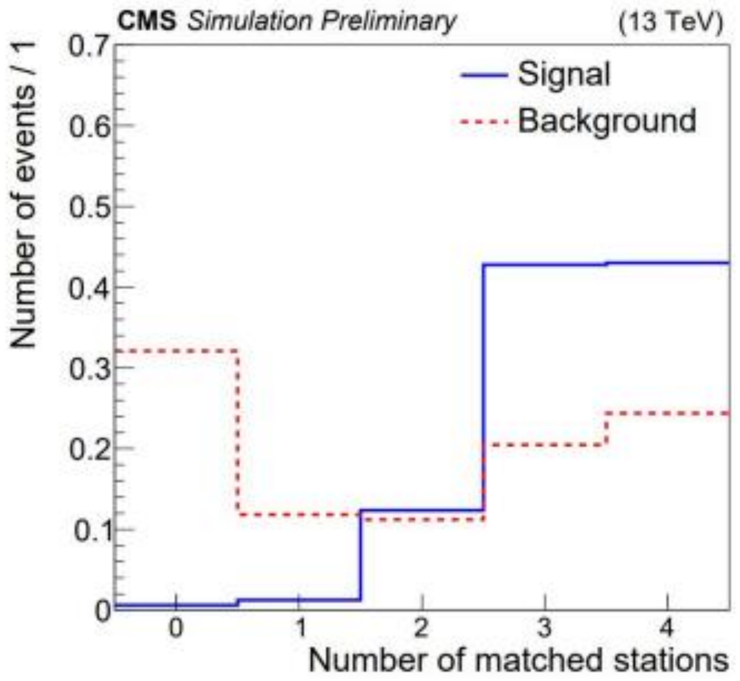
MUON MVA IDENTIFICATION

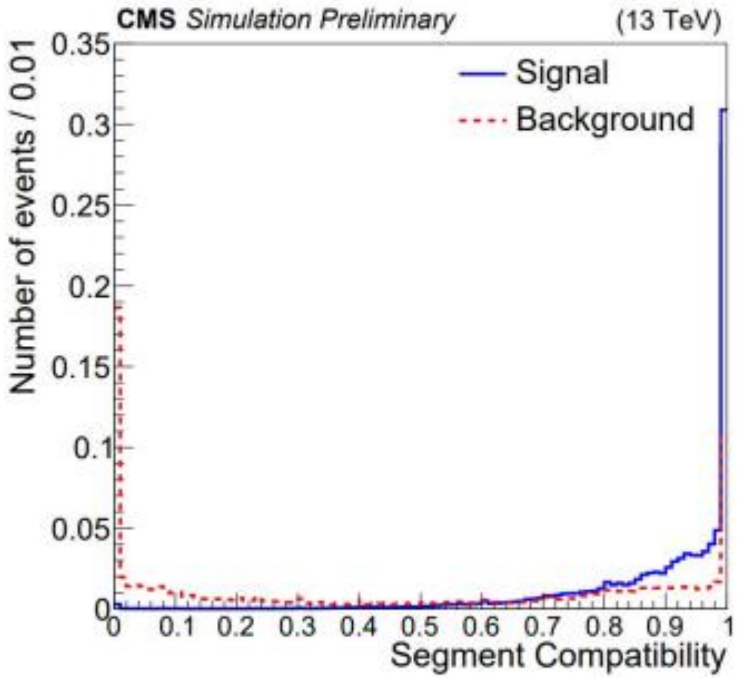
CMS-PAS-MUO-22-001

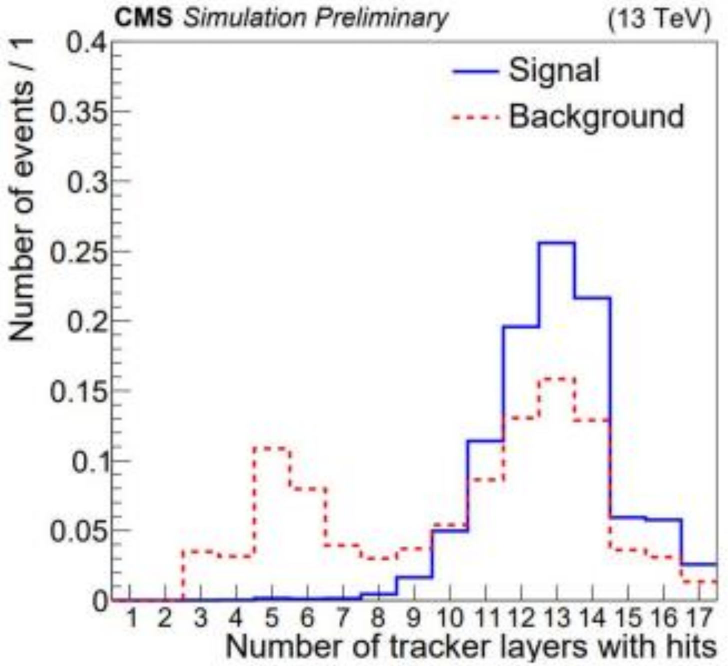
Recently developed and available for Run 3 analyses!

- Aimed at discriminating spurious muons and instrumental backgrounds.
- Should replace standard medium and tight cut-based IDs.
- **Training**: muons from ttbar sample dividing by its origin in ‘signal’ and ‘background’.
- **Model**: random forest.
- **Inputs**: same input variables as standard cut-based IDs.
- **Output**: probability of a muon to be a signal muon.





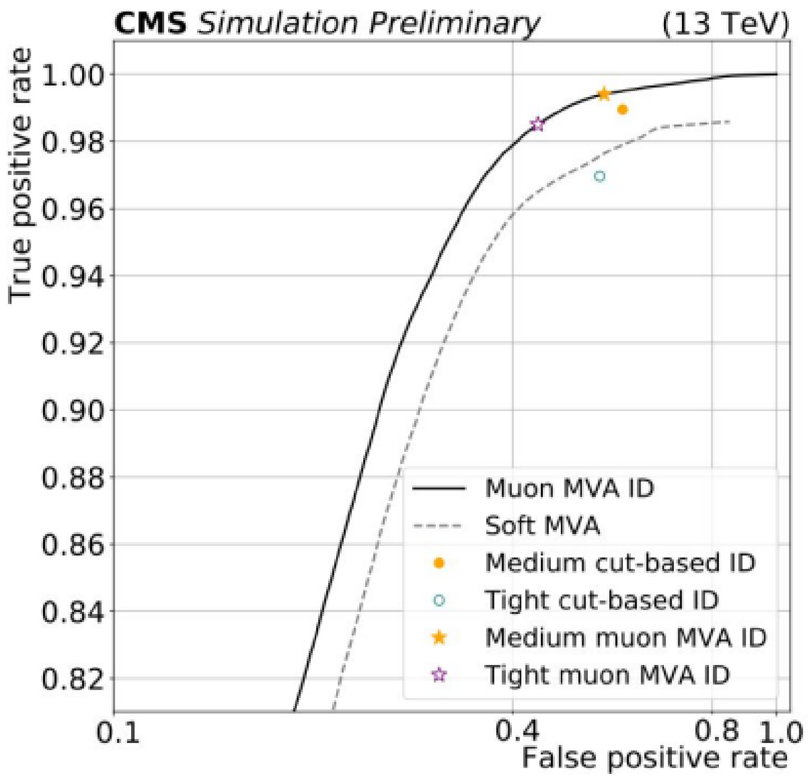


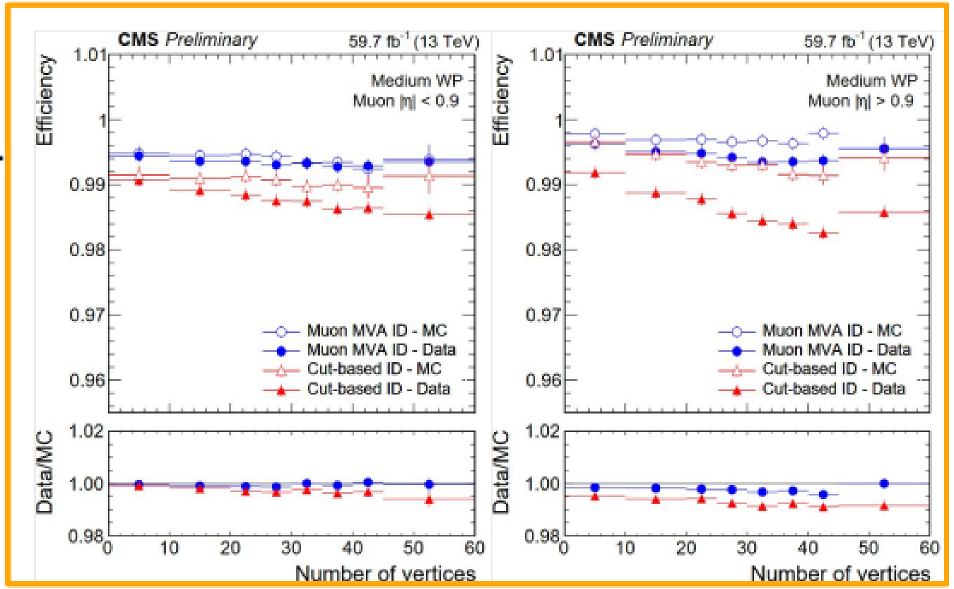


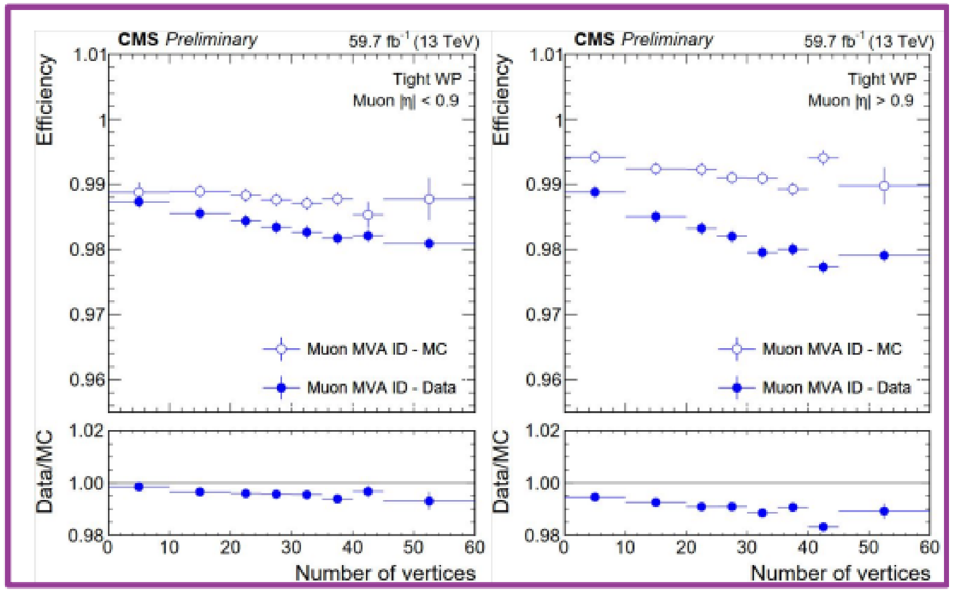
MUON MVA IDENTIFICATION

CMS-PAS-MUO-22-001

➢ **Good performance achieved with the MVA, promising for Run 3!!**





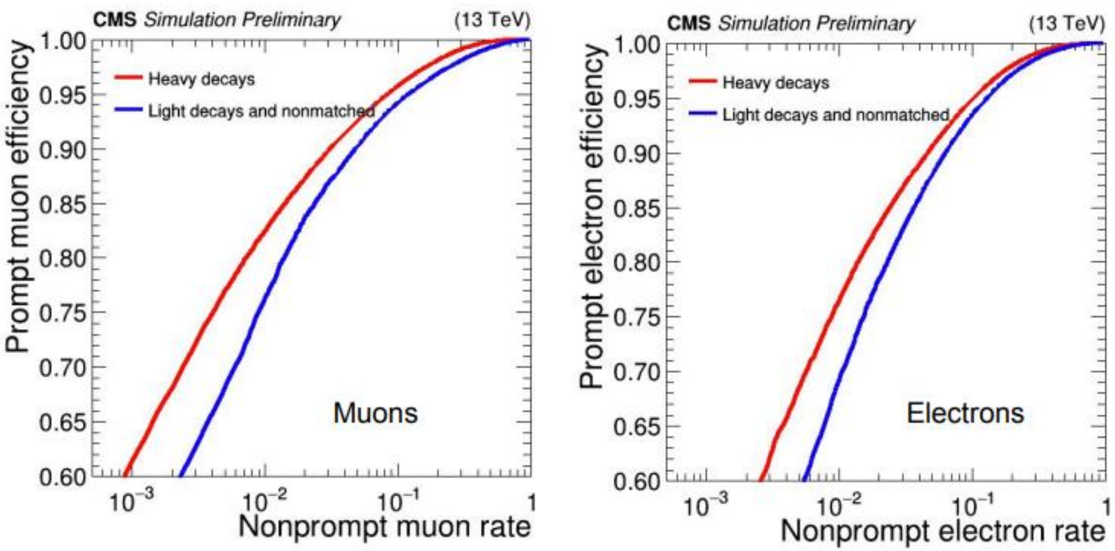


- **Medium MVA WP**: same background contamination as the medium cut-based WP with 0.5-1% higher efficiency.
- **Tight MVA WP**: achieves a 10% smaller background contamination than the medium MVA ID and the efficiency is about 99% .
- MVA ID is **more stable as a function of PU** than the cut-based ID .

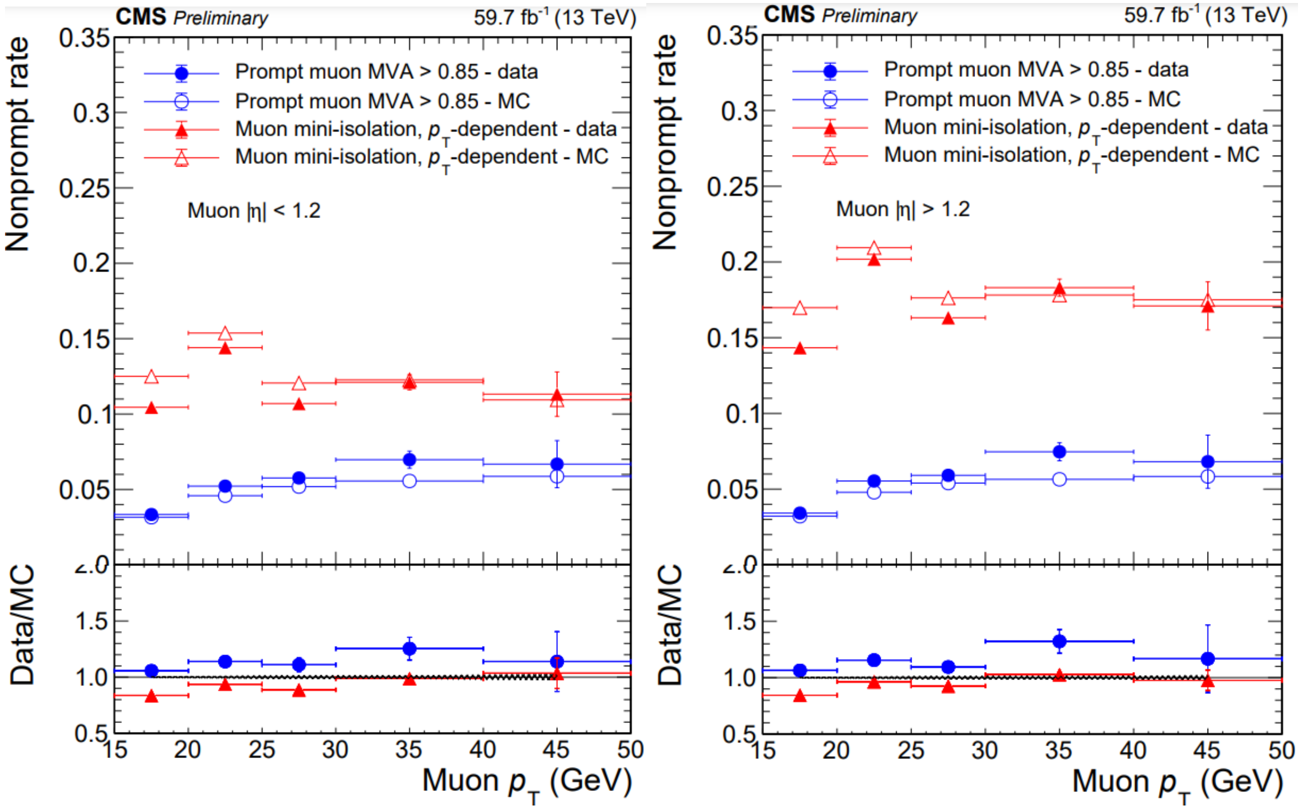
PROMPT MVA (TTH)

[CMS-PAS-MUO-22-001]

- ➢ We have another MVA to select **prompt muons and electrons at analysis level** including isolation variables as input.
- ➢ This MVA aims to improve the selection of prompt muons/electrons, arising from the decay of a W, Z, H boson or τ lepton, and to reduce the contamination of muons from other sources.
- ➢ **Lepton MVA broadly used** in CMS analyses: searches for supersymmetry, standard model precision measurements, studies of the top quark properties and measurements in the Higgs boson sector.



- Much better discrimination power for heavy decays than for light decays.



➢ **This model is the key to reject fakes in many analyses. But there is an effort in CMS to do an adaptation/modification of ParticleNet algorithm used for jet tagging for leptons, which is even more promising!!**

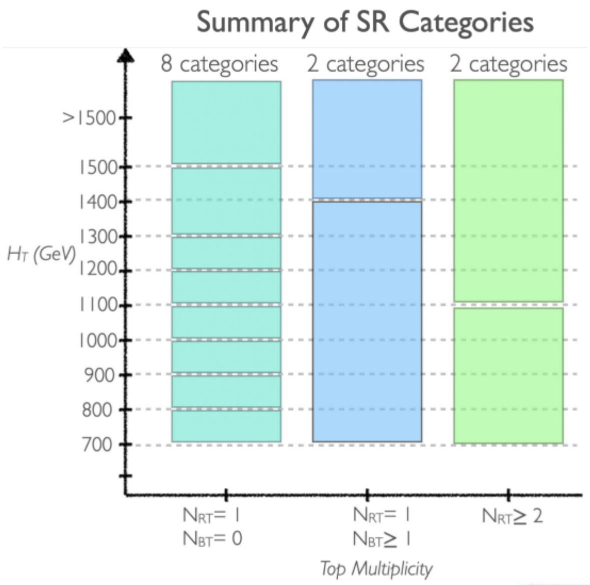
ANALYSES

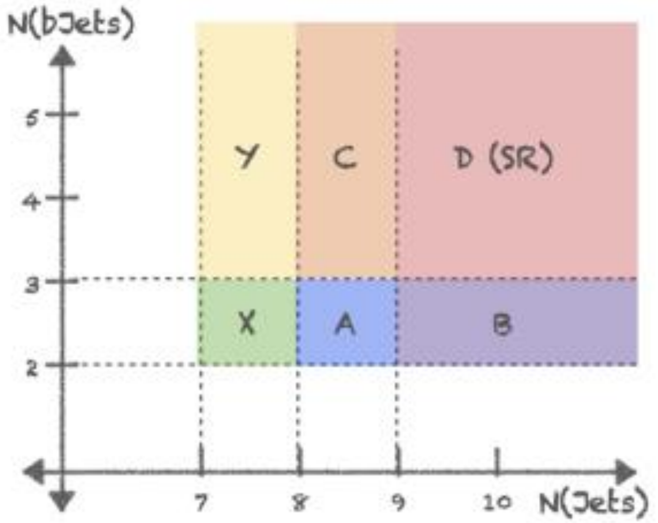
FOUR-TOPS ALL-HADRONIC

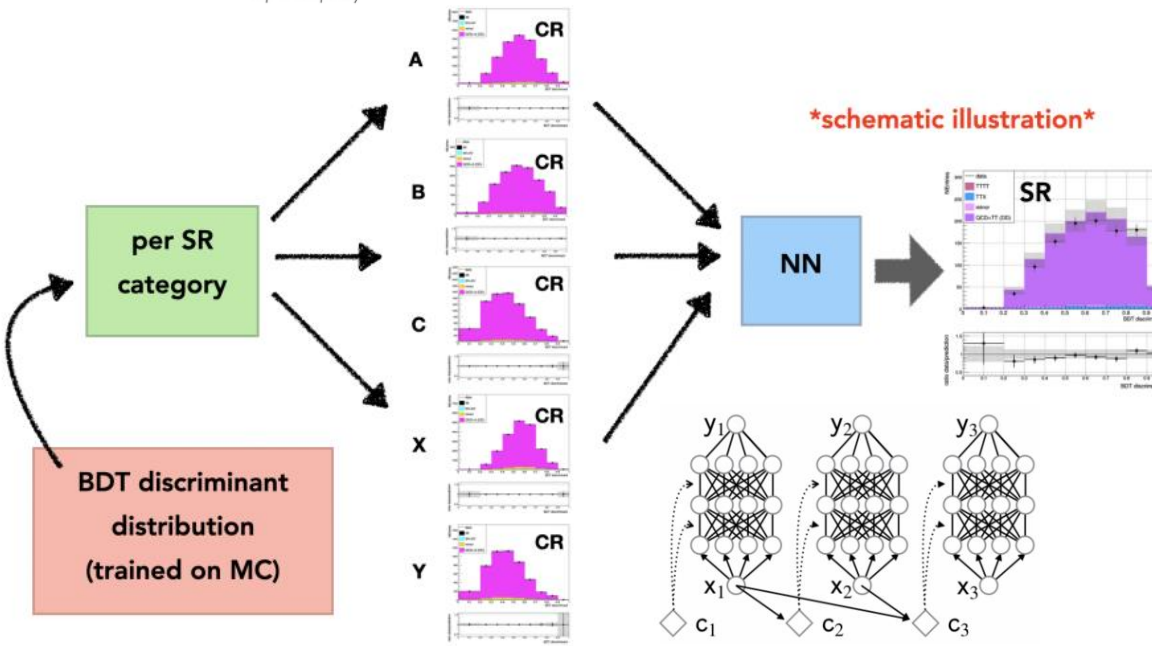
CMS-TOP-21-005 — Submitted to PLB

First all-hadronic tttt search with the main strategy based on ML techniques!

- After multijet preselection a **BDT is trained** to separate signal vs. QCD background → final discriminating observable.
- Signal region (SR) splited by HT and resolved+boosted top multiplicity.
- DD yields estimated from data yields in 5 control regions (CRs) using **extended ABCD method**.
- **Background estimation with Neural Autoregressive Flows (NAF), novel in CMS!!**
  - NAF transforms input BDT histograms (MC) to match target (data) trained on same 5 CRs.



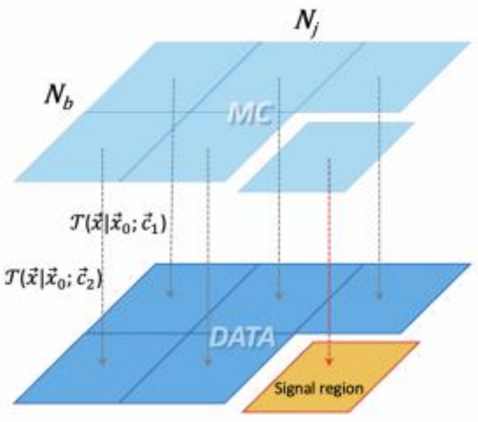




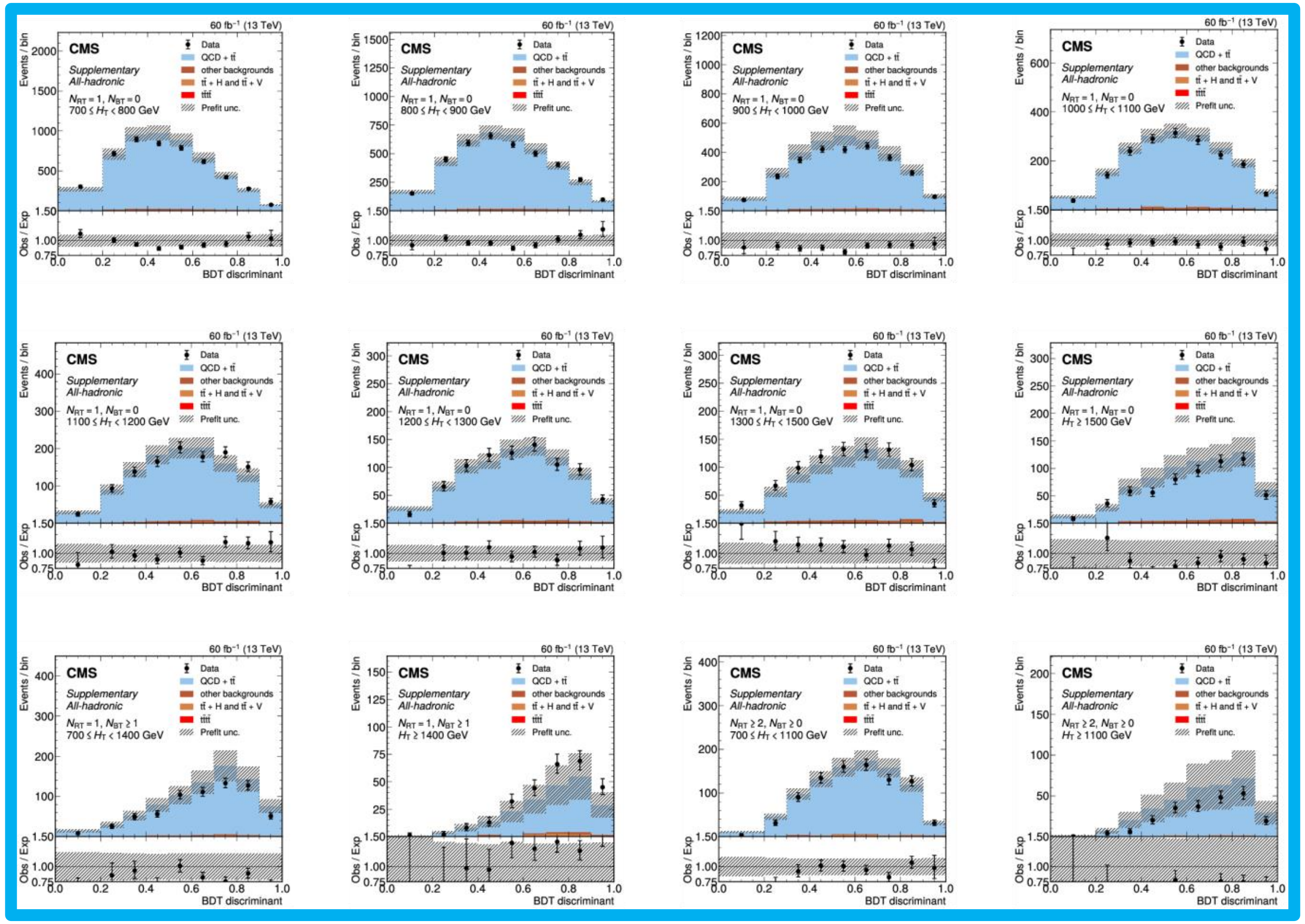
FOUR-TOPS ALL-HADRONIC

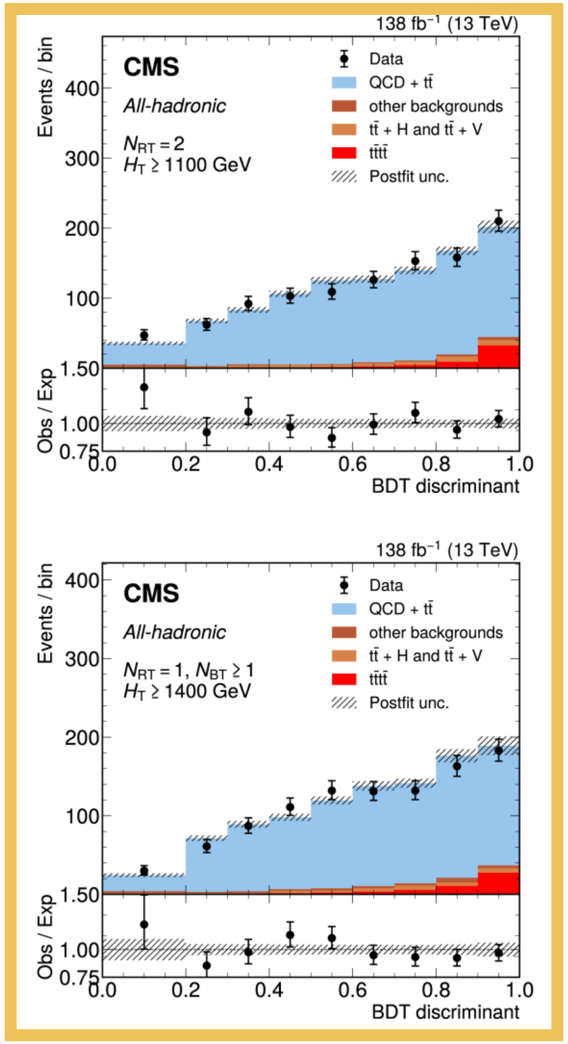
CMS-TOP-21-005 — Submitted to PLB

- After training in the 5 CRs, the transformation between MC and data is applied to simulation in the SR → morphed to predict the shape of the tt+QCDmultijet background in the SR.
- Uncertainties derived from discrepancies in the validation region and applied to corresponding SR.



$$\hat{p}_{data}(\vec{x}|\vec{c}_i) = \mathcal{T}(\vec{x},\vec{x}_0|\,\vec{c}_i) \otimes p_{MC}(\vec{x}_0|\vec{c}_i)$$

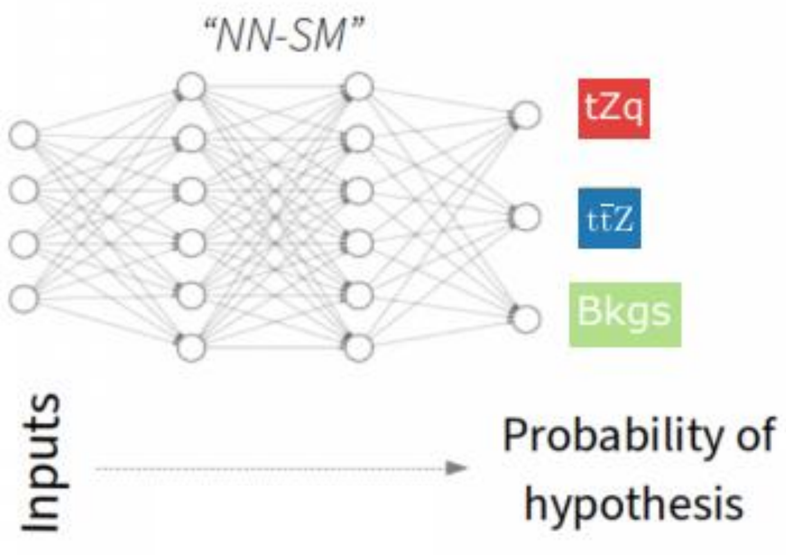


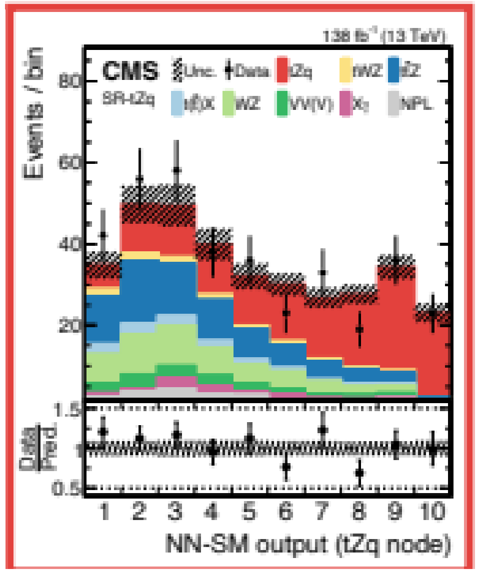


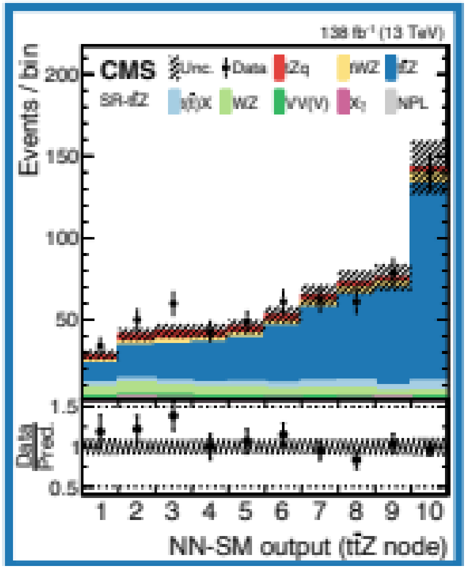
EFT Analysis in associated Z production

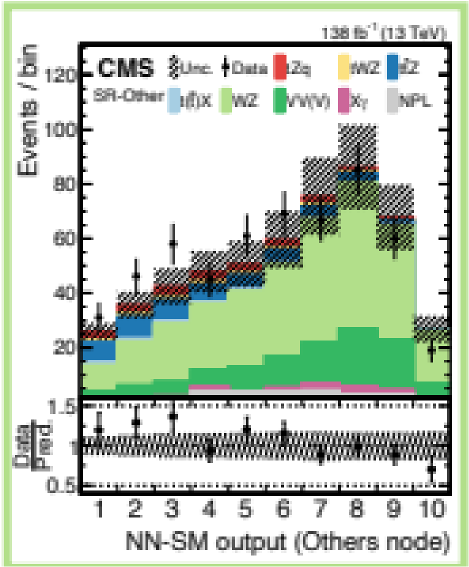
CMS-TOP-21-001

- Search for BSM physics in the scope of Effective Field Theory (EFT) **considering interference effects during ML training!**
- Targeting tt + Z, tZW, tZq processes considering **5 EFT operators**.





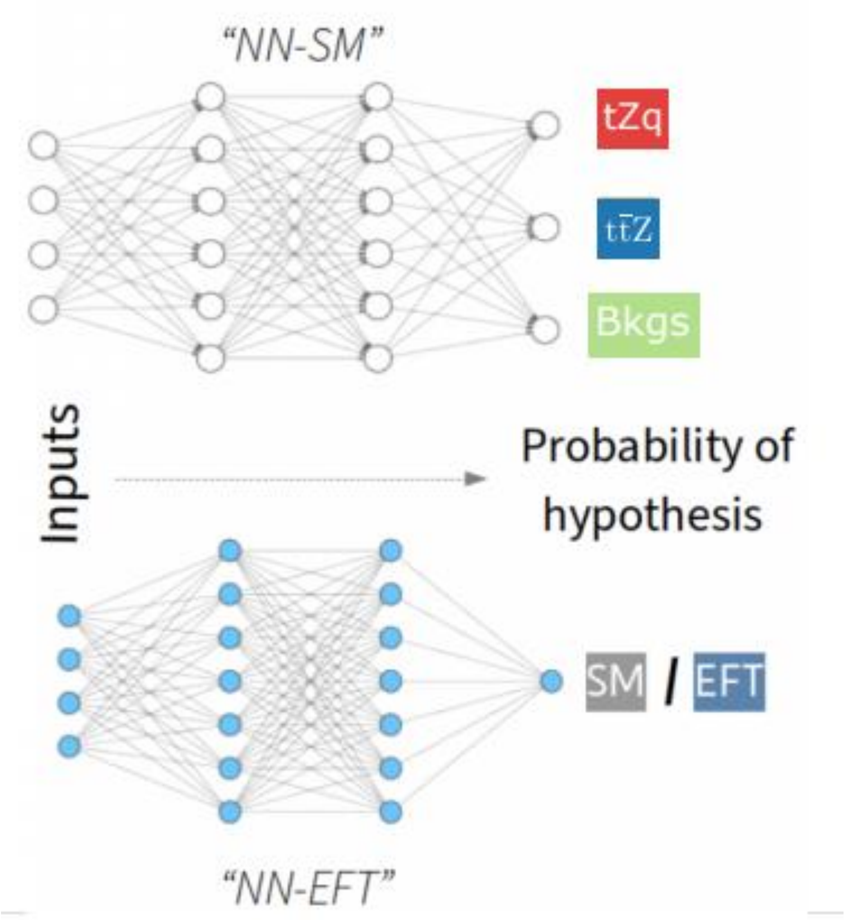




EFT ANALYSIS IN ASSOCIATED Z PRODUCTION

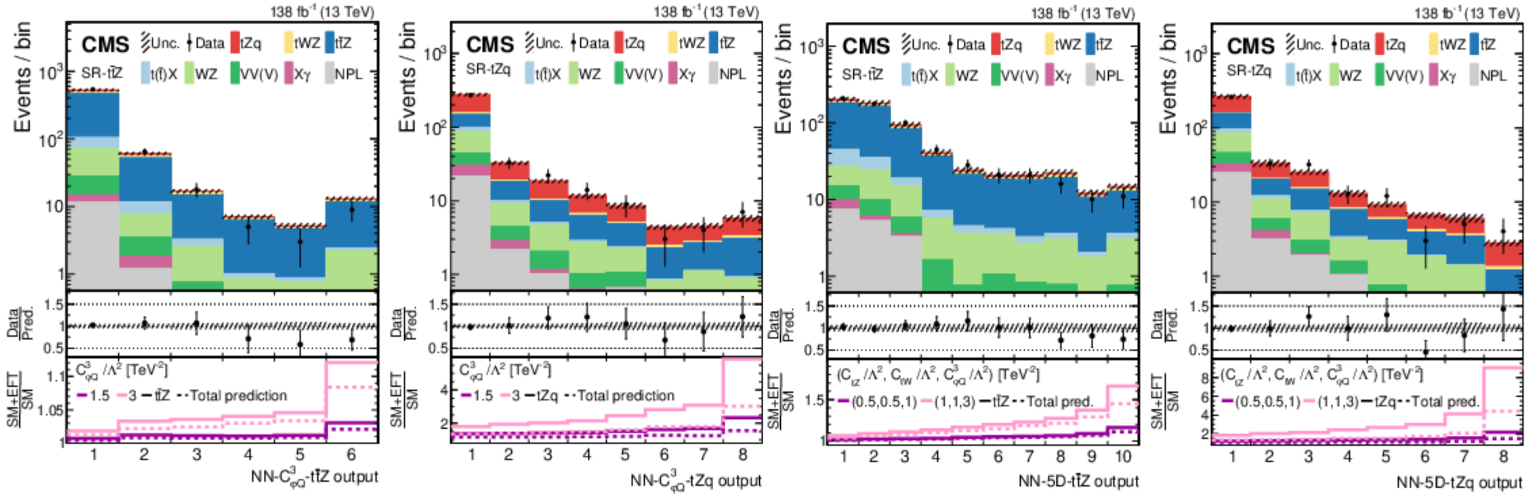
CMS-TOP-21-001

➢ **Increase in sensitivity by usage of ML between 20-70 % → ML crucial for this analysis.**



➢ Further trainings with events classified as tZq and tt+Z → Binary classification of **SM vs. EFT** with following setups:

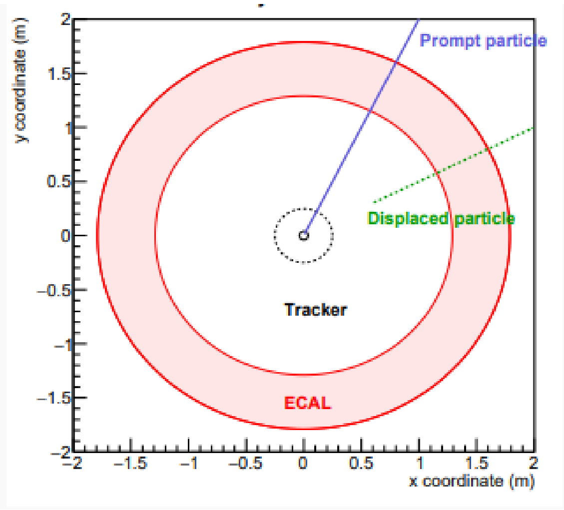
- 1D: Consider only one operator at a time.
- 5D: Consider effects from all operators simultaneously.

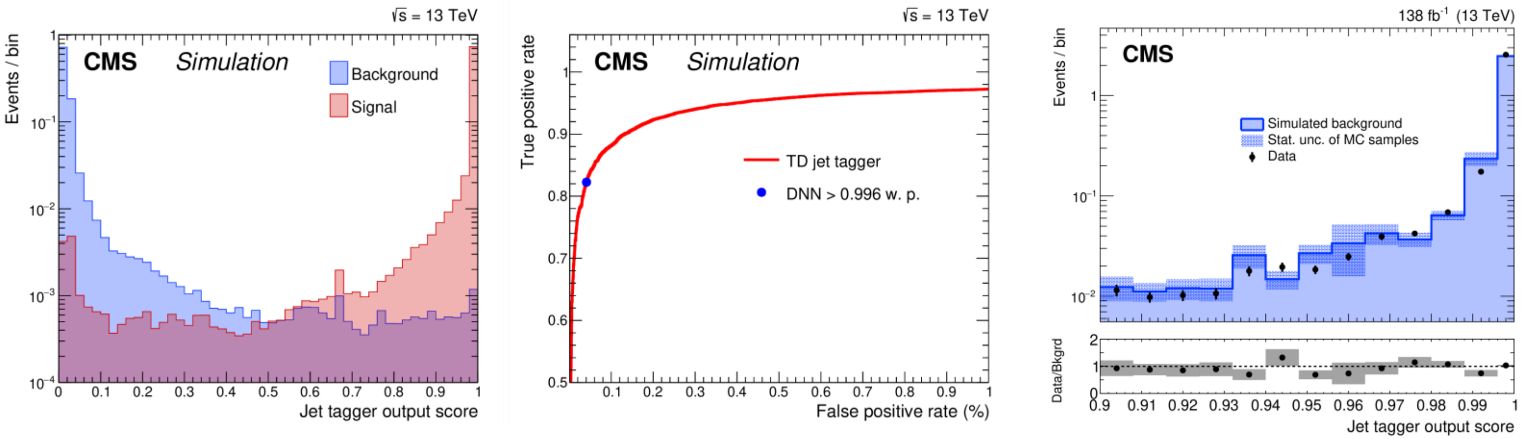


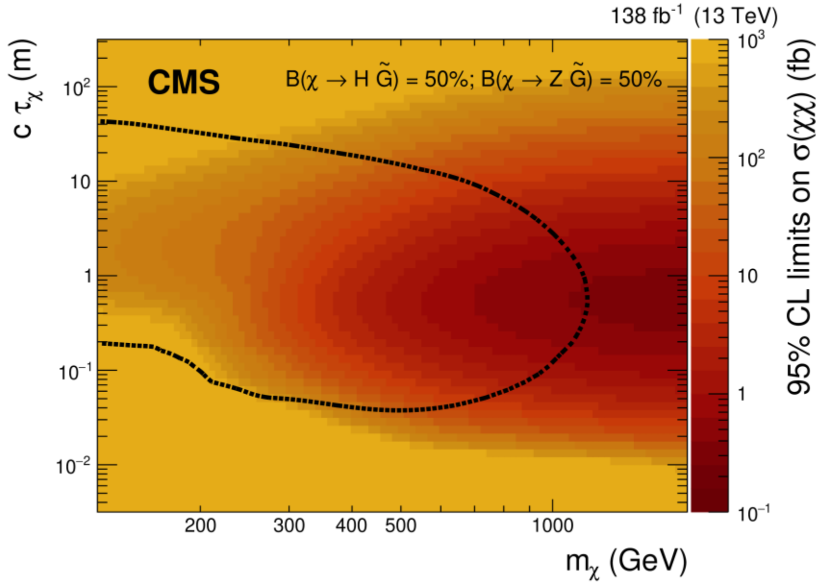
LONG-LIVED USING A TRACKLESS AND DELAYED JET TAGGER

CMS-EXO-21-014

- **Search for long-lived particles (LLPs)** decaying into displaced jets using a trackless and delayed jet tagger.
- Tracking efficiency decreases with displacement and jets appear as **trackless**, mostly consisting of neutral components.
- Slow-moving LLPs and/or path length increase due to displacement (**delay**).
- **Strategy**: increase sensitivity (lower masses) combining ECAL delay with track information in a new **DNN jet tagger.**







- Achieved very strong background suppression by using a DNN tagger.
- Compared to previous searches for promptly decaying χ, **sensitivity 20–10 times better at mχ = 400–600 GeV.**

DQM

MOTIVATION

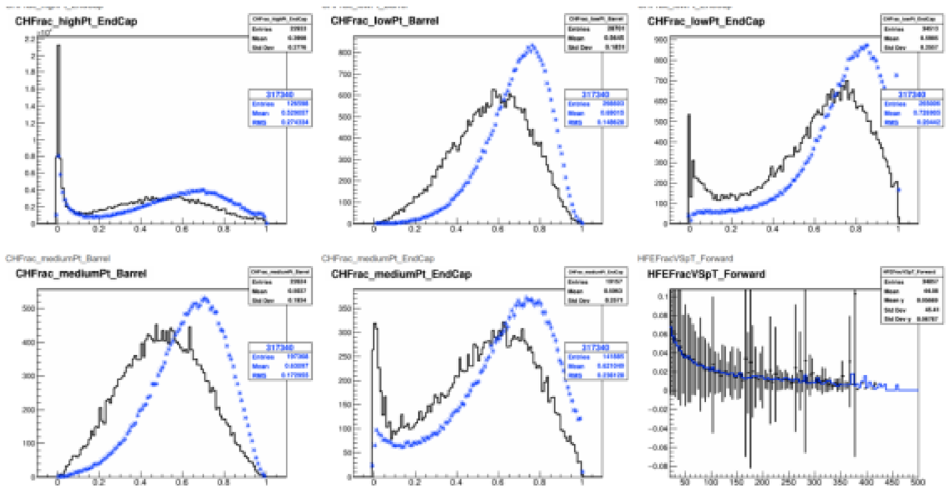
- In CMS, **data quality monitoring (DQM)** and data certification (DC) are crucial components in ensuring reliable data quality suitable for physics analysis.
- The current method for certification of quantities is mostly reliant on **manually monitoring** reference histograms summarizing the status and performance of the detector.
- Given the **large number of distributions** that are mentioned, the process is **time intensive** and prone to **human error** when deviations from the norm are less evident.
- **Solution**: Machine Learning methods for certifying offline/online DQM data!!

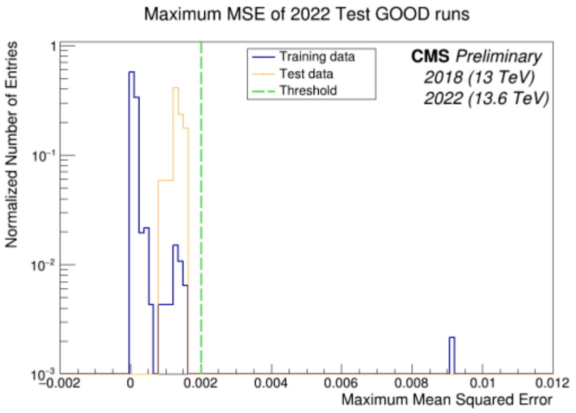
- **Example of JetMET certification, but other efforts on going:**

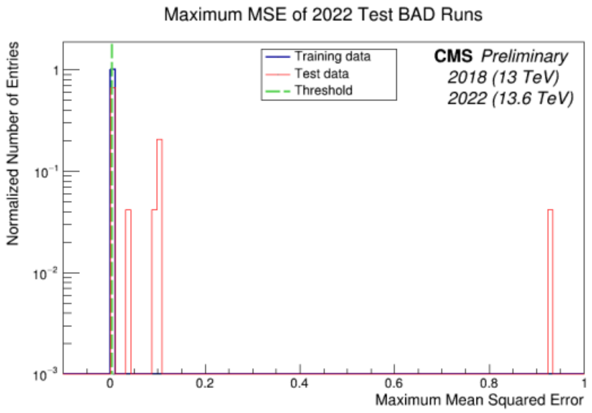
- Resistive Plate Chambers subsystem of muon detectors [ACAT '22].
- Electromagnetic and Hadronic Calorimeters subsystems [CMS-DP-2022-043].
- Pixel Silicon Tracker subsystem [CMS-DP-2022-013].

JetMET certification [CMS-DP-2023-032]

- Variable reduction.
- Data certification with supervised classification.
- Anomaly detection with autoencoders.



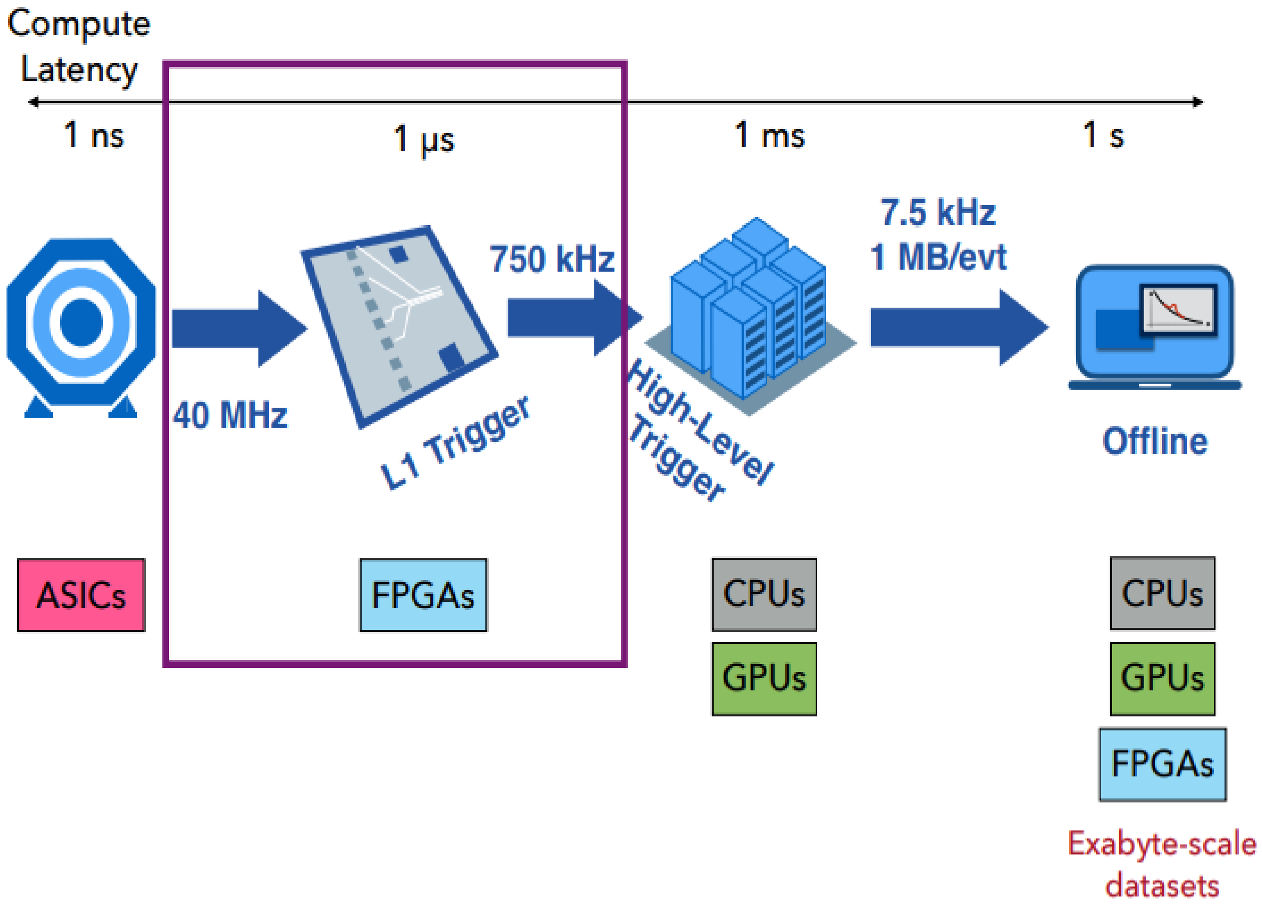




TRIGGER

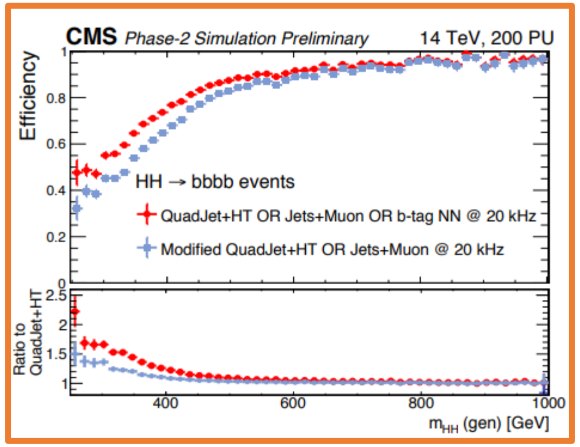
TRIGGER DEVELOPMENTS

- **Challenges for HL-LHC**
  - Computational efficiency.
  - Extreme data rates of (100 TB/s).
  - Development of tool to port ML models to **FPGAs**: CMS first in deploying AI at 40 MHz in Run 3!

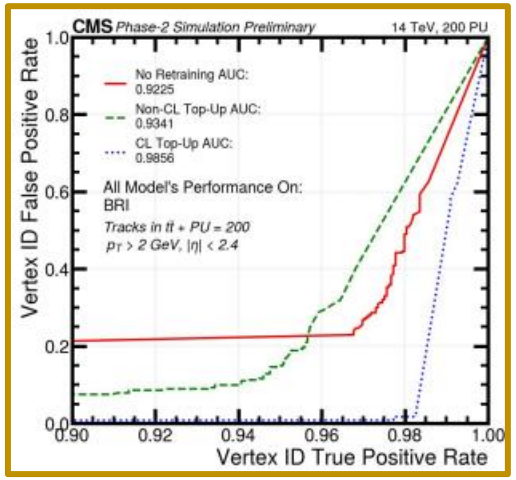


Machine Leaning in Trigger:

- Graph Neural Networks for tracking.
- **1D convolutional neural networks for jets.**
- **Continual learning for top-up trainings.**
- (Variational) autoencoders for anomaly detection.
- Fast ML (quantization, pruning).



CMS-DP-2022-021



CMS-DP-2023-022

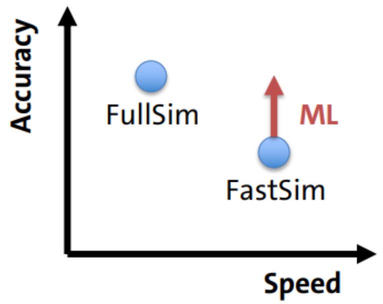
SIMULATION

REFINING FAST SIMULATION USING ML

Talk in CHEP2023

- In CMS, two simulation chains are used that produce output of same dimensionality/structure: FullSim and FastSim.
- In total: FastSim ≈ **10x faster** than FullSim.
- Higher LHC luminosity (= more events) & detector upgrades (= more complex data)-> FastSim techniques needed.

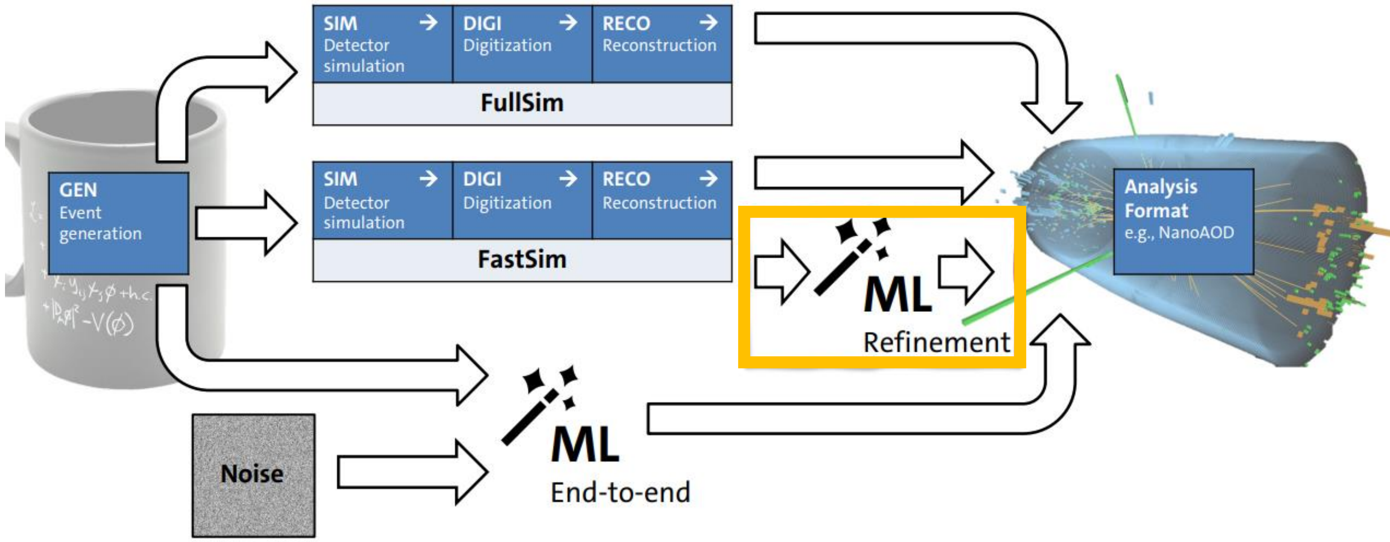
	GEN → Event generation	**SIM** → Detector simulation	**DIGI** → Digitization	**RECO** Reconstruction
FullSim	same e.g., MadGraph	GEANT4	same	analyze as if data
FastSim ≈ 15% of sim. events		parametrized energy loss 100x faster		use GEN info 2.5x faster



Aim: increase FastSim accuracy to promote its wider usage.

Possible FastSim tuning approaches:

- Internal tuning of functions/parameters (within SIM/RECO).
- Post-hoc tuning (after NanoAOD):
  - Reweighting: defining weights for individual events/objects.
  - **Refining:** changing high-level observables.

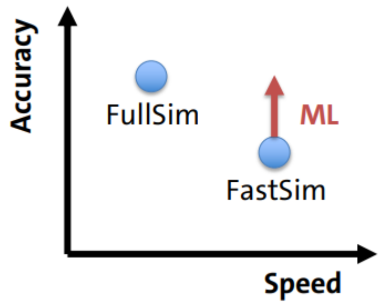


Refining fast simulation using ML

Talk in CHEP2023

- In CMS, two simulation chains are used that produce output of same dimensionality/structure: FullSim and FastSim.
- In total: FastSim ≈ **10x faster** than FullSim.
- Higher LHC luminosity (= more events) & detector upgrades (= more complex data)-> FastSim techniques needed.

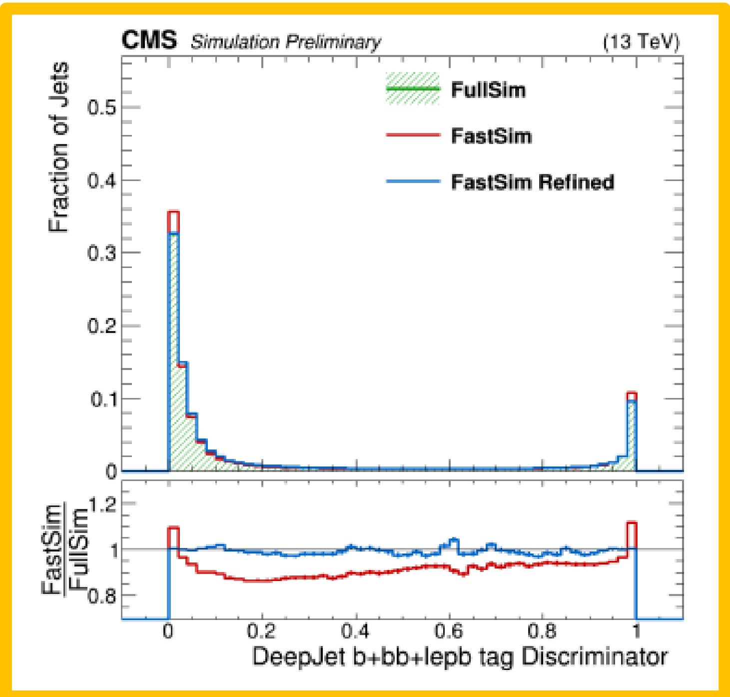
	GEN → Event generation	SIM → Detector simulation	DIGI → Digitization	RECO Reconstruction
FullSim	same e.g., MadGraph	GEANT4	same	analyze as if data
FastSim ≈ 15% of sim. events		parametrized energy loss 100x faster		use GEN info 2.5x faster



Aim: increase FastSim accuracy to promote its wider usage.

Possible FastSim tuning approaches:

- Internal tuning of functions/parameters (within SIM/RECO).
- Post-hoc tuning (after NanoAOD):
  - Reweighting: defining weights for individual events/objects.
  - **Refining:** changing high-level observables.



Considerably improved agreement with FullSim output!!

SUMMARY

- Machine Learning has significant role in HEP.
- Wide array of strategies and applications, very active field of research!!
- ML techniques have helped in many analyses to have more sensitivity and will be very useful for the HL-LHC.
- I had to make a selection, but there are many more promising and innovative efforts in CMS on going, stay tuned!!

Back up

PARTICLENET FOR JET TAGGING

AK8 jets: Jets clustered with the anti-k_T algorithm [1] using a distance parameter of 0.8.

m_{SD}: The groomed jet mass obtained from the "soft drop" algorithm [2] with β = 0 and z_{cut} = 0.1.

DeepAK8: A multi-class particle identification algorithm [3] for identifying hadronic decays of highly Lorentz-boosted top quarks and W, Z, and Higgs bosons and classifying different decay modes (e.g., Z→bb, Z→cc, Z→qq), based on AK8 jets. The DeepAK8 algorithm uses a deep one-dimensional convolutional neural network (CNN) to process particle-flow candidates and secondary vertices associated with the jet.

DeepAK8-MD: An alternative DeepAK8 algorithm [3] designed such that its outputs are decorrelated with the jet mass. The DeepAK8-MD algorithm is developed with the adversarial training technique: A mass prediction network is added to the nominal DeepAK8 network during the training phase with the goal of predicting the jet mass from the learned features, and then the accuracy of the mass prediction is used as a penalty to prevent the algorithm from learning features that are correlated with the jet mass. Jets from different signal and background samples are also reweighted to yield flat distributions in both p_T and m_{SD} to aid the training.

DeepAK8-DDT: An alternative approach to decorrelate the DeepAK8 algorithm with the jet mass based on the Designing Decorrelated Taggers (DDT) method [4]. The output of the nominal DeepAK8 algorithm is transformed as a function of $\rho = \ln(m_{SD}^2/p_T^2)$ and the jet p_T:

$$\text{DeepAK8-DDT}(\rho, p_T) = \text{DeepAK8}(\rho, p_T) - \text{DeepAK8}^{(x\%)}(\rho, p_T)$$

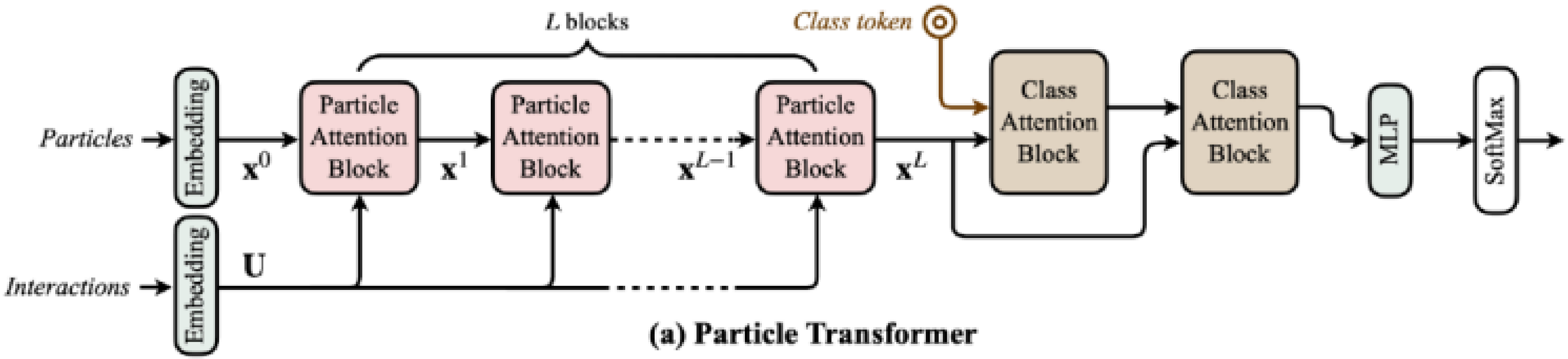
where DeepAK8$^{(x\%)}$ is the (100 – x) percentile of the DeepAK8 output distribution in simulated background (QCD multijet) events. After the transformation, the selection DeepAK8-DDT > 0 yields a constant QCD background efficiency of x% across the m_{SD} and the p_T range. The values x = 5 and x = 2 are investigated in this note, and the corresponding versions are labelled as "DeepAK8-DDT (5%)" and "DeepAK8-DDT (2%)", respectively.

ParticleNet: A graph neural network-based particle identification algorithm for identifying hadronic decays of highly Lorentz-boosted top quarks and W, Z, and Higgs bosons and classifying different decay modes (e.g., Z→bb, Z→cc, Z→qq). The same inputs as the DeepAK8 algorithm, i.e., the particle-flow candidates and secondary vertices associated with the AK8 jet, are also used in this algorithm. The "ParticleNet" neural network architecture [5] is used to process the input particle-flow candidates and secondary vertices in a permutation-invariant way.

ParticleNet-MD: A mass-decorrelated particle identification algorithm designed for identifying two-prong hadronic decays of highly Lorentz-boosted particles (e.g., X→bb, X→cc, X→qq). To achieve mass independence, the training uses a dedicated simulated sample containing Lorentz-boosted spin-0 particles (X) with a flat mass spectrum between 15 to 250 GeV and subsequently decaying to a pair of quarks (X→bb, X→cc, X→qq) as the signal sample, and the QCD multijet sample as the background sample. Jets from the signal and background samples are also reweighted to yield flat distributions in both p_T and m_{SD} for the training. The same inputs and network architecture as the ParticleNet algorithm are used.

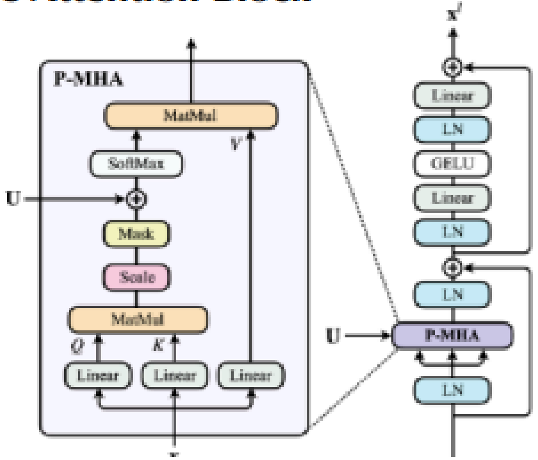
The ParticleNet-MD algorithm outputs four probability-like scores: p(X→bb), p(X→cc), p(X→qq), and p(QCD). An X→bb discriminant can be defined as p(X→bb) / [p(X→bb) + p(QCD)] while an X→cc discriminant can be defined as p(X→cc) / [p(X→cc) + p(QCD)]. A mixed discriminant, defined as [p(X→cc) + p(X→qq)] / [p(X→cc) + p(X→qq) + p(QCD)], can be used for W boson identification.

CMS-DP-2022-050



(a) Particle Transformer

Particle Attention Block



$$\text{P-MHA}(Q,K,V) = \text{SoftMax}(QK^T/\sqrt{d_k} + \mathbf{U})V$$

d_k: dimension of K

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

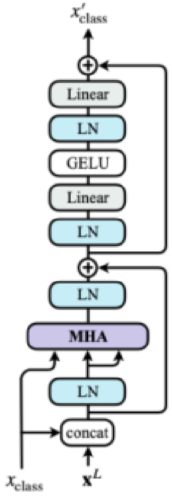
$$\text{MHA}_C(Q_C, K_C, V_C) = \text{SoftMax}(Q_C K_C^T/\sqrt{d_{kC}})V_C$$

$Q_C = W_{qC} x_{\text{class}} + b_{qC}$

$\underbrace{K_C = W_{kC}\mathbf{z} + b_{kC} \quad V_C = W_{vC}\mathbf{z} + b_{vC}}_{\mathbf{z} = [x_{\text{class}}, \mathbf{x}^L]}$

d_{kC}: dimension of K_C

Class Attention Block



(c) Class Attention Block

Transformer for heavy flavor jet identification

CMS-DP-2022-050

AK4 jets: Jets clustered with the anti-k_T algorithm [1] using a distance parameter of 0.4.

DeepJet: A deep neural network algorithm based on charged and neutral particle-flow jet constituents, secondary vertices and jet level variables. The network architecture is based on convolutional layers and recurrent layers (LSTMs) [2]. DeepJet is a sequence-based model: the ordering of the inputs matters, and the performance depends on the ordering choice between these.

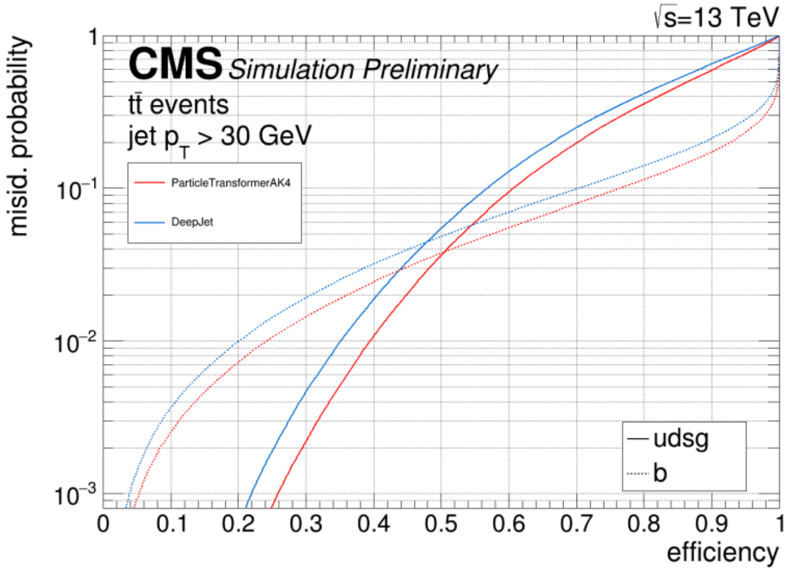
ParticleTransformerAK4: A Transformer neural network [3] tailored for AK4 jet classification task [4]. The network introduces pairwise "interaction" features between all jet constituent particles and secondary vertices as inputs. These additional variables give us a better view of the internal relations of the jet constituents. Thus, the neural network can better learn and understand the internal structure of a jet, and ultimately improving the performance.

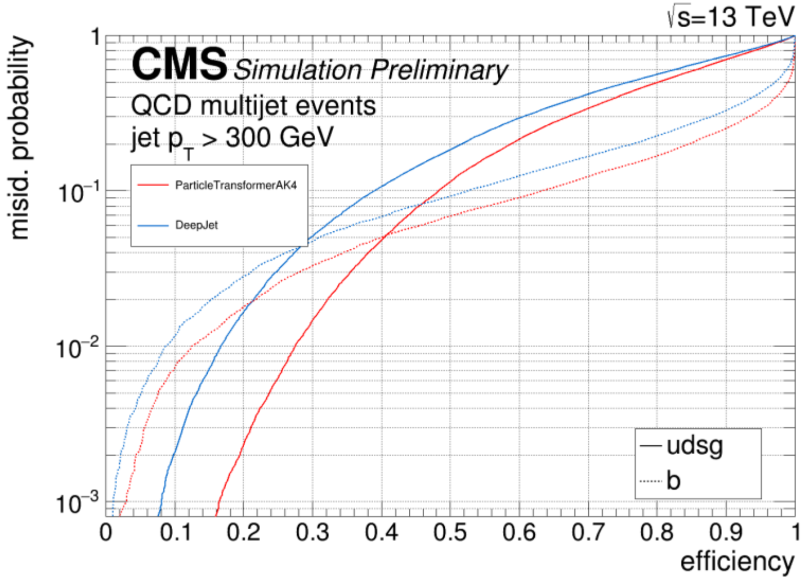
For AK4 jet classification, the network architecture is slightly modified from the original architecture in [3]. The self-interaction terms are removed from the pairwise features to avoid a potential bias in some of the calibration methods. We use three particle attention blocks and one class attention block. The GELU activation function is used through the different attention blocks, while the ReLU activation function is used for the initial embedding MLPs and the final MLP. The number of heads in the attention blocks is 8, and the feature dimension is 128 for all the layers, except for the embedding MLPs which are (128-512-128) for the particle embedding layer, and (64-64-64-8) for the interaction embedding layer.

TRANSFORMER FOR HEAVY FLAVOR JET IDENTIFICATION

CMS-DP-2022-050

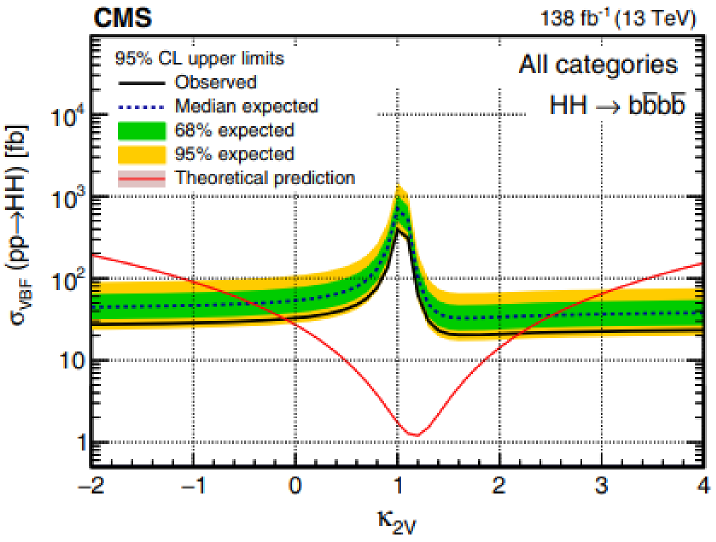
➢ **ParticleTransformerAK4 significantly improves the performance compared to the current state-of-the-art model, DeepJet.**





➢ **Application of more powerful ML architectures** in heavy flavor identification allowed recently setting the most stringent constraints on HH production!

- HIG-20-005: Setting the most stringent constraints on HH production in the four b-quark final state.

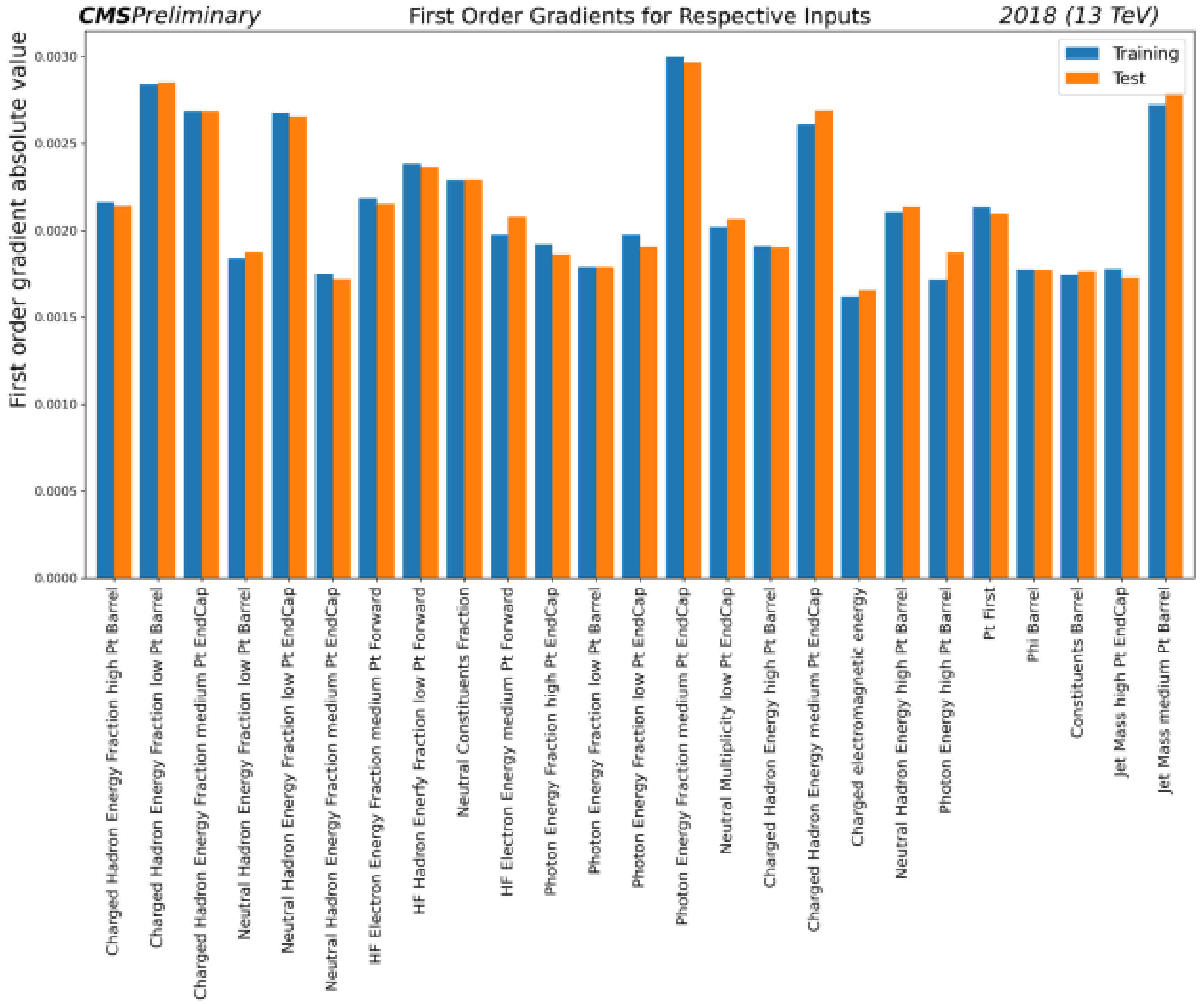


JetMET: Variable reduction

CMS-DP-2023-032

Fist step: select relevant observables.

- In order to determine which observable provides the most power in identifying anomalies, supervised models are trained on a set of already certified Runs.
- **Model**: A fully connected neural network classifier is trained on the mean values of 124 observable distributions of each run, with labels of **good/ bad** are provided for each Run.
- **Loss function** used is binary cross entropy.
- To **optimize** the set of input features, the first order gradient of the loss is calculated with respect to the input features.
- Jet energy fractions are the most important features to use for DC.

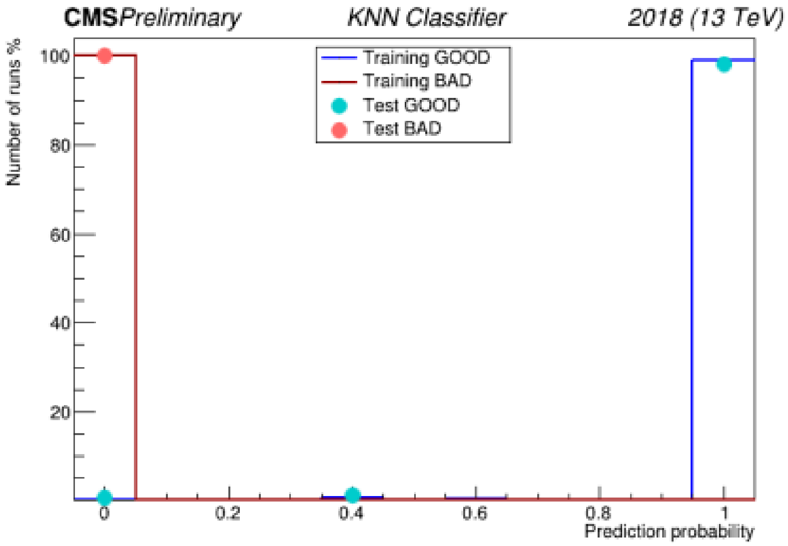


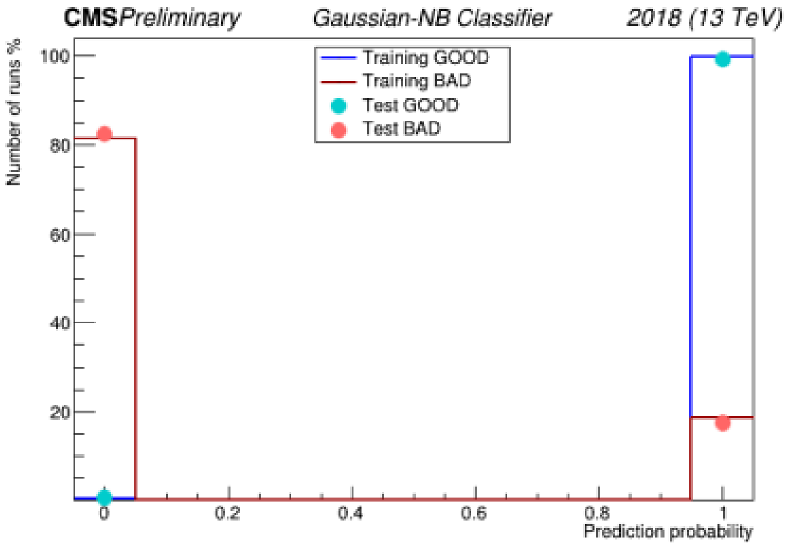
JetMET: Supervised classification

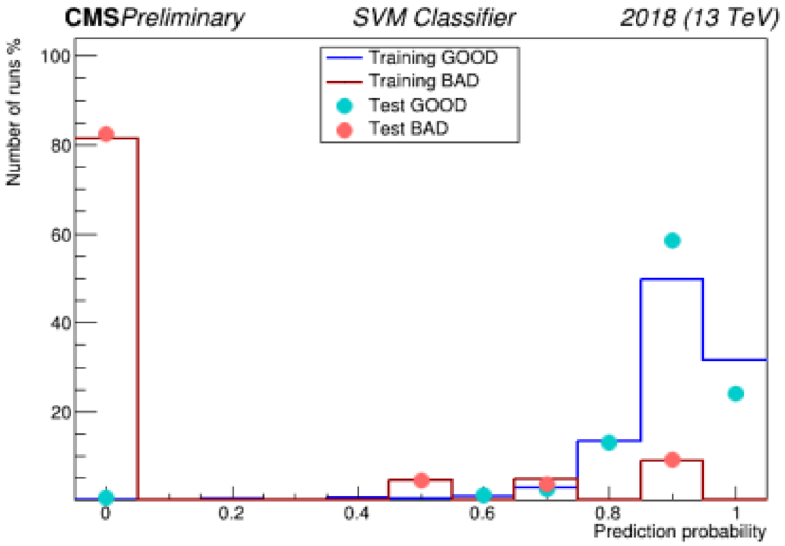
CMS-DP-2023-032

Second step: check the selected observables.

- In order to judge the power of those 25 jet fraction observables in detecting potentially anomalous Runs, we test with **three different classifiers**:
  - K- Nearest Neighbors (KNN).
  - Gaussian Naive Bayes (Gaussian-NB).
  - Support Vector Machine (SVM).
- The three classifiers have shown excellent performance achieving ROC AUC scores above 0.90 for both training and test sets.
- Among all the classifiers, KNN has emerged as the best performing one.





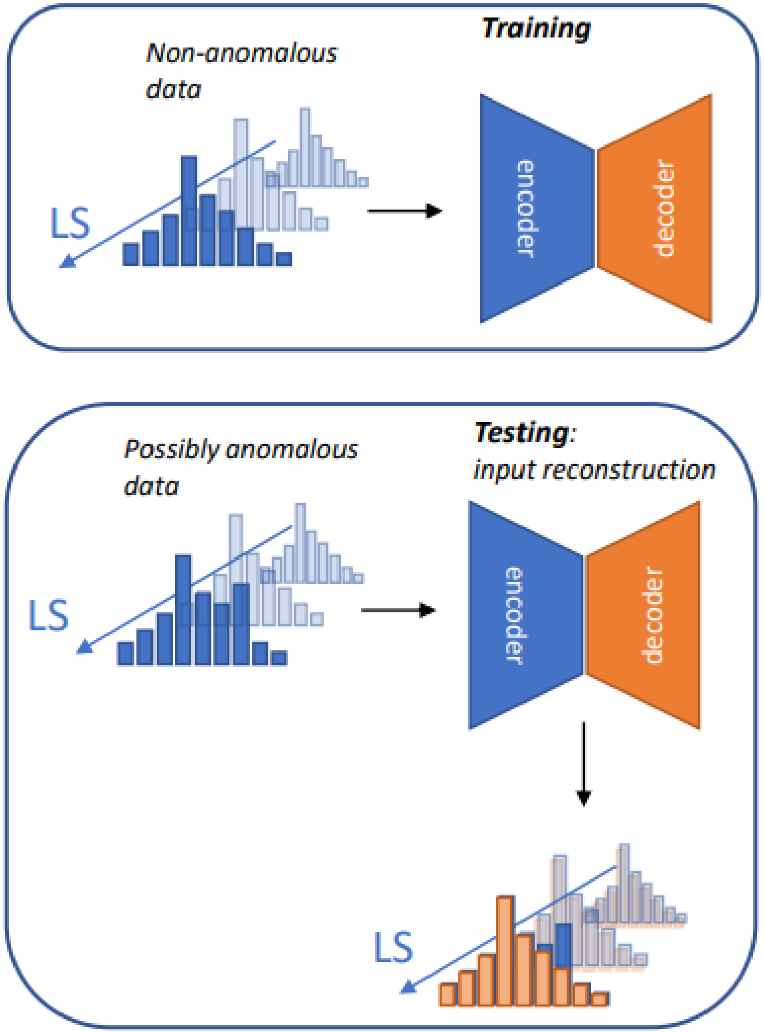


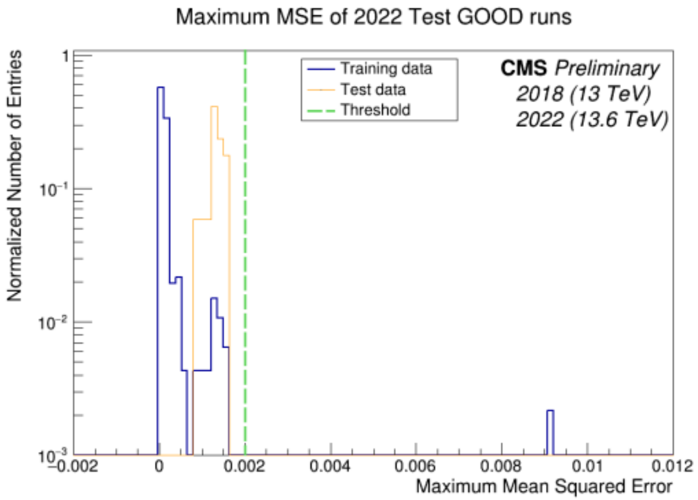
JetMET: Anomaly detection with AE

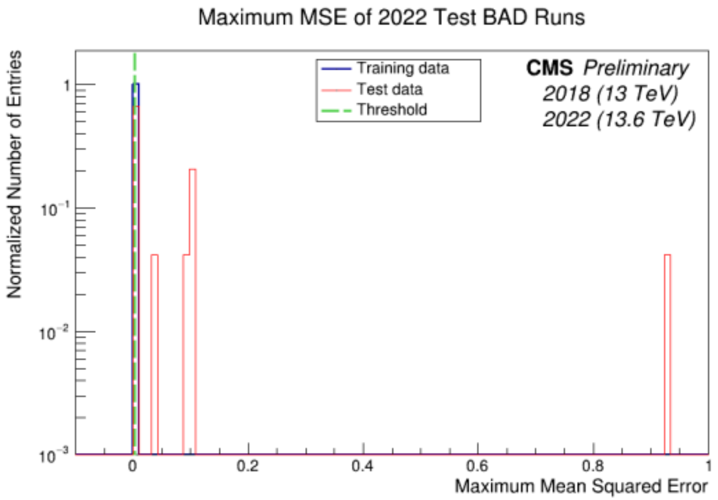
CMS-DP-2023-032

Third step: detect anomalies with unsupervised autoencoders.

- **Model**: fully connected autoencoder model is trained with good Runs only and tested with different good and bad Runs.
- **Inputs**: the mean values of 25 jet energy fractions.
- **Loss function**: the mean square error (MSE) for this model.
- The proposed approach is as follows:
  1. Find the highest MSE value in the training dataset of good Runs and use it as a threshold.
  2. Apply the threshold to the maximum MSE values of the test dataset, creating a cutoff.
  3. Classify the Runs as bad if their max MSE value exceeds the threshold determined; otherwise, classify them as good.

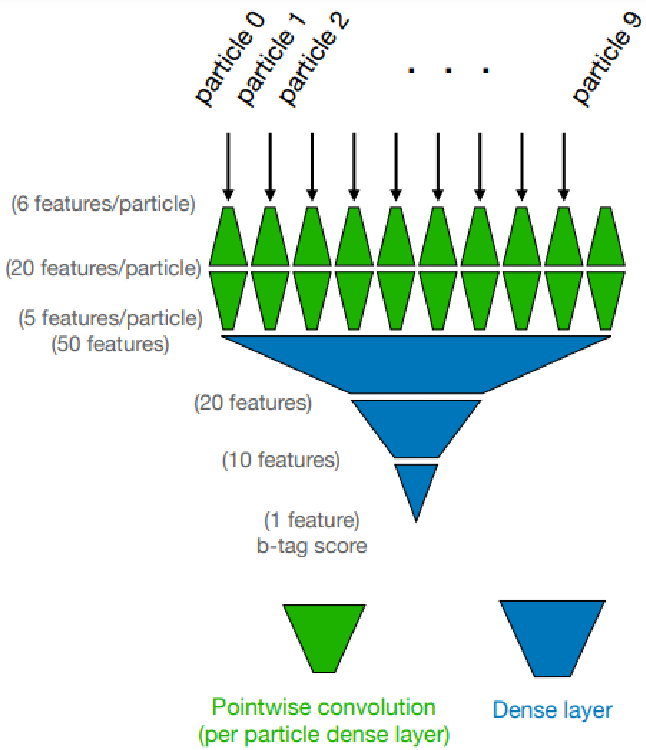




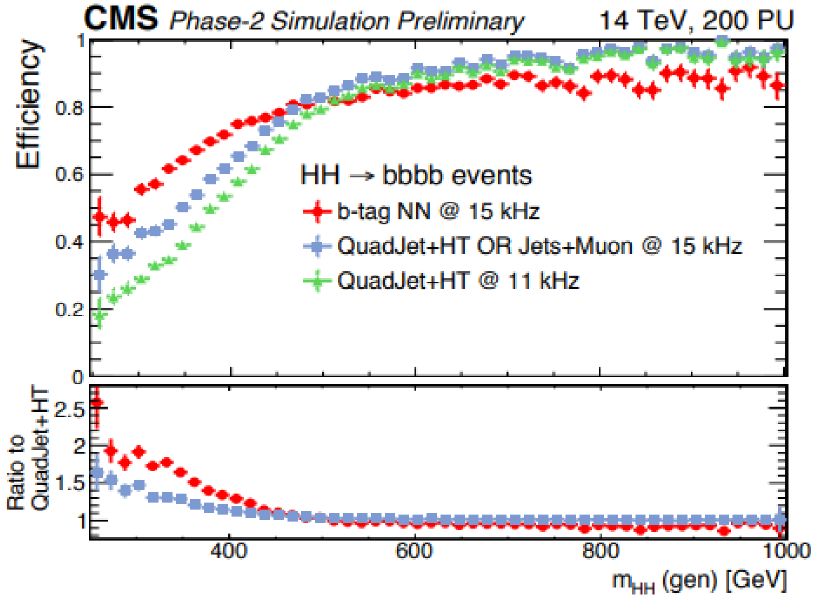


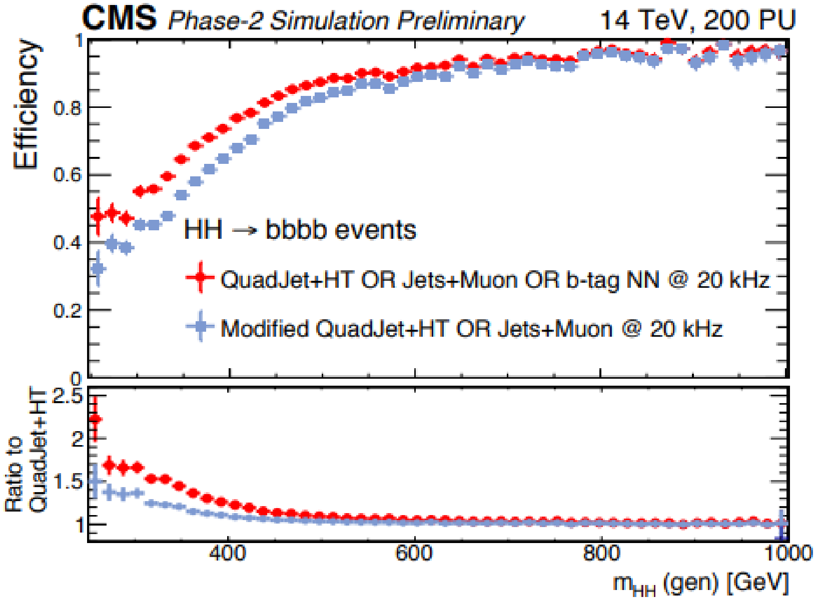
NN FOR THE IDENTIFICATION OF B QUARKS IN THE PHASE-2 LEVEL-1 TRIGGER [CMS-DP-2022-021]

- NN to discriminate between jets originating from bottom quarks and jets originating from light quarks or gluons using.
- It is feasible to be implemented on current trigger hardware where jets are built.
- It is also capable of operating within the budgeted latency requirements of the Level-1 trigger environment.
  - **Training**: Monte Carlo samples with 200 pileup interactions, simulating the conditions of the HL-LHC.
  - **Model**: Neural network with two 1D convolutional layers.
  - **Inputs**: top ten PUPPI candidates within each jet. Variables: particle type, kinematic and vertex information.
  - **Output**: probability of a jet to have been originated from bottom quarks.



- **Performance tested in trigger efficiency in HH→bbbb events**
  - b-tag NN trigger increases the efficiency for events with low mHH.

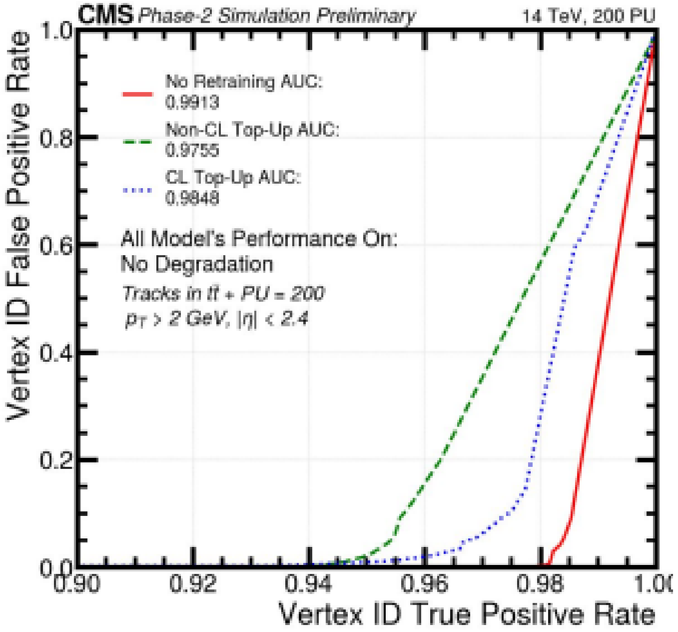


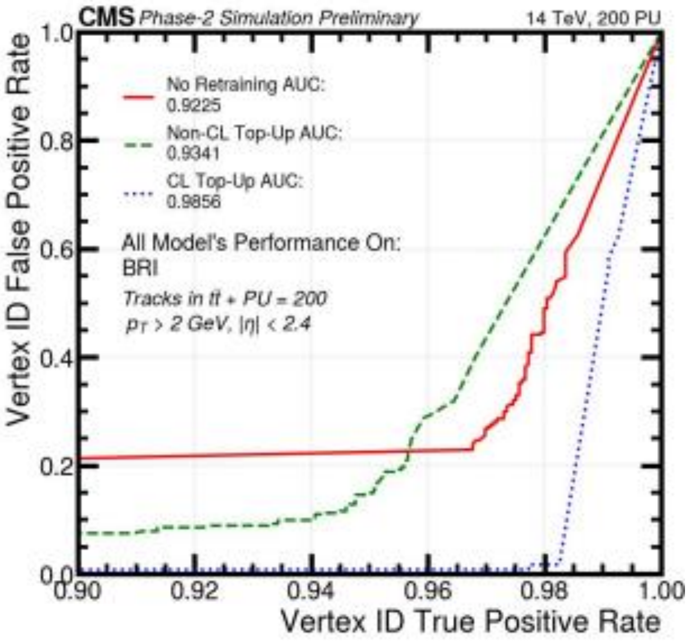


CONTINUAL LEARNING IN THE CMS PHASE-2 LEVEL-1 TRIGGER

CMS-DP-2023-022

- Continual learning (CL) is very useful in online environment and changing conditions to avoid retraining.
- In the Phase-2 Upgrade of the CMS Level-1 trigger many ML algorithms are to be used but have to be small and lightweight so may not be able to deal with large domain shift at inference time.
- This study explores a possible solution to this issue in the form of **top-up trainings of a model on small datasets with degradation.**
- **Performance tested in a simple fake vertex ID task that uses a simple convolutional NN. We have 3 scenarios:**
  - **No retraining:** standard training in large sample.
  - **Non-CL Top-Up:** standard training + top-up trainings in 3 new degraded datasets with a lower learning rate .
  - **CL Top-Up:** standard training + top-up trainings in 3 new degraded datasets but using a CL algorithm.





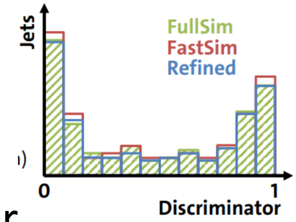
- This study demonstrates how **ML can lose robustness** to changing experimental conditions.
- **CL is shown to be a well-performing solution** to maintaining performance in these kinds of ML models.

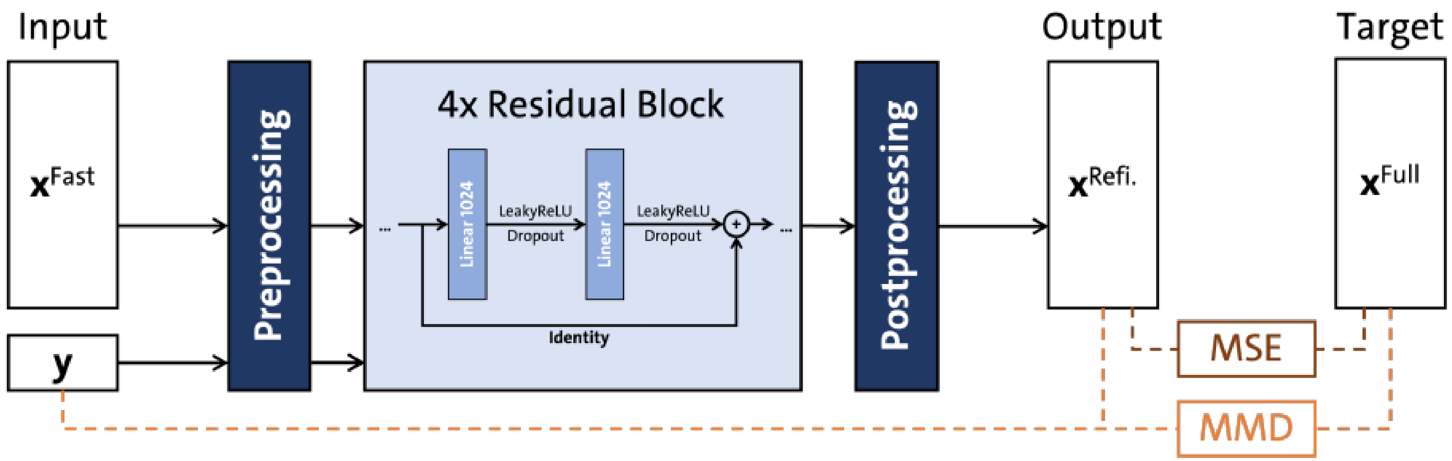
REFINING FAST SIMULATION USING ML

[Talk in CHEP2023]

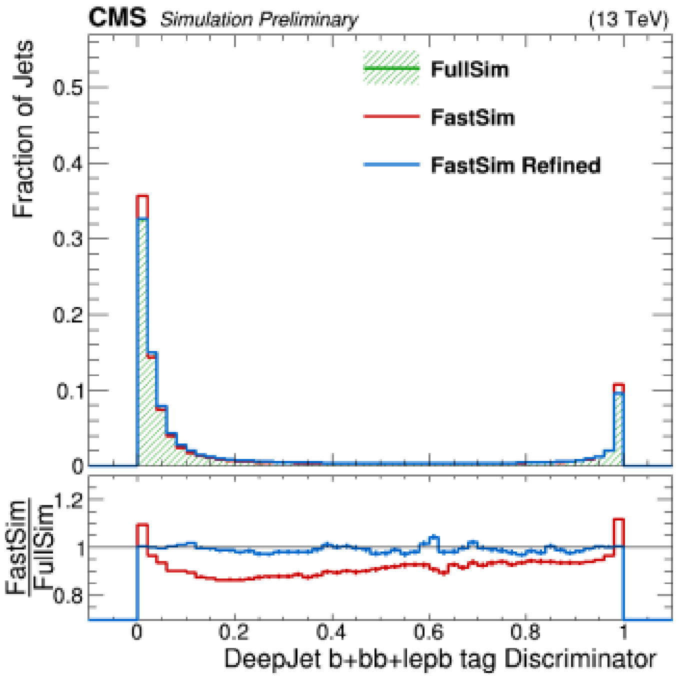
➢ **Example of usage on jet flavour tagging: 4 NanoAOD DeepJet discriminators.**

- **Training:** SUSY simplified model “T1tttt“ simulated with FastSim and FullSim.
- **Model:** regression neural network (ResNet).
- **Inputs**: FastSim variables x^{Fast} = 4 DeepJet discriminators. Parameters y = p_T^{GEN} , η^{GEN}, true hadron flavor.
- **Output:** Refined variables $x^{Refi.}$ = 4 DeepJet discriminators.
- **Target**: FullSim variables x^{Full} = 4 DeepJet discriminators.
- Combine **loss terms** via MDMM (Modified Differential Method of Multipliers) algorithm.





MSE: Mean Squared Error (jet-jet pairs) MMD: Maximum Mean Discrepancy (ensembles)



Considerably improved agreement with FullSim output!!