# MLPF - Particle Flow with Machine Learning

Ka Wa, Ho [*]
On behalf of the CMS collaboration

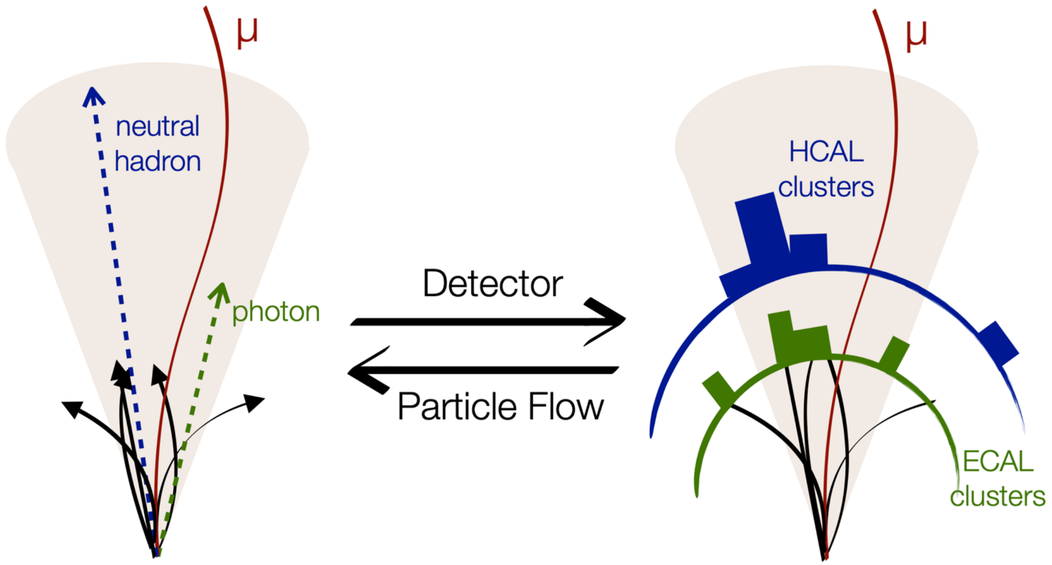
**[*] Brown University**

**BOOST 2025 - 17th International Workshop on Boosted Object Phenomenology, Reconstruction, Measurements, and Searches at Colliders**

# Particle Flow

- **Particle flow (PF)** algorithms reconstruct stable particles by correlating measurements from all sub-detectors in a multi-layered particle detector

- It is a **global** reconstruction strategy in contrast to local reconstruction that only utilize information from individual sub-detector



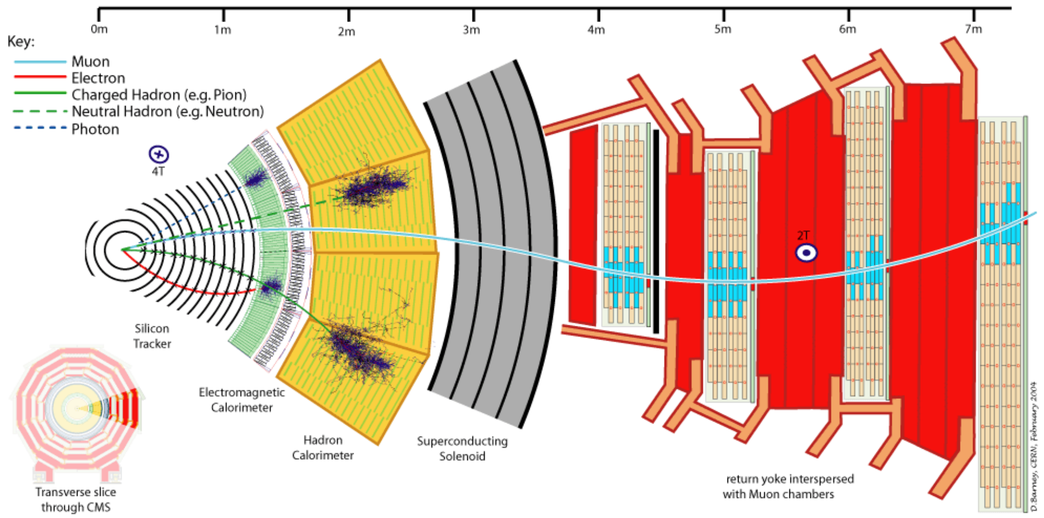

**PF solves the inverse problem of detector signals -> particles**

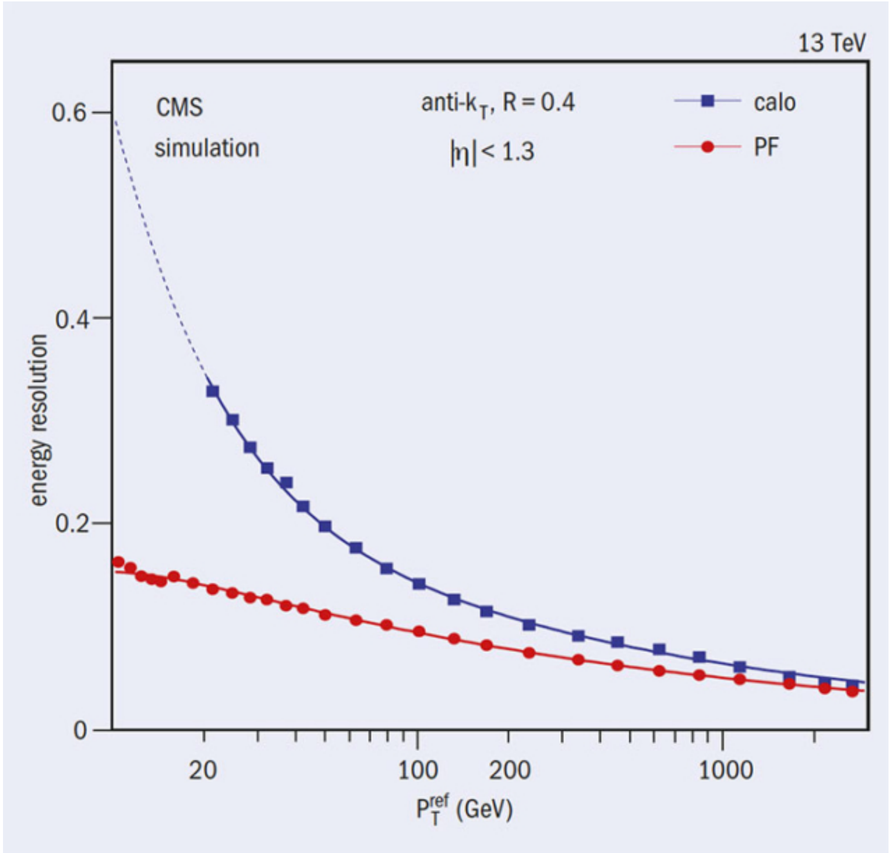
# Particle Flow

- The existing PF algorithm at CMS links **tracks**, reconstructed from the inner tracker and the muon system, to **calorimeter clusters** with **ruled-based criteria**

- PF significantly improves the **jet**, **missing transverse momentum**, and **hadronic tau energy resolution**, as well as **particle identification** in general

- It has been the standard reconstruction algorithm at CMS since Run 1 of the LHC (2009 - 2013)

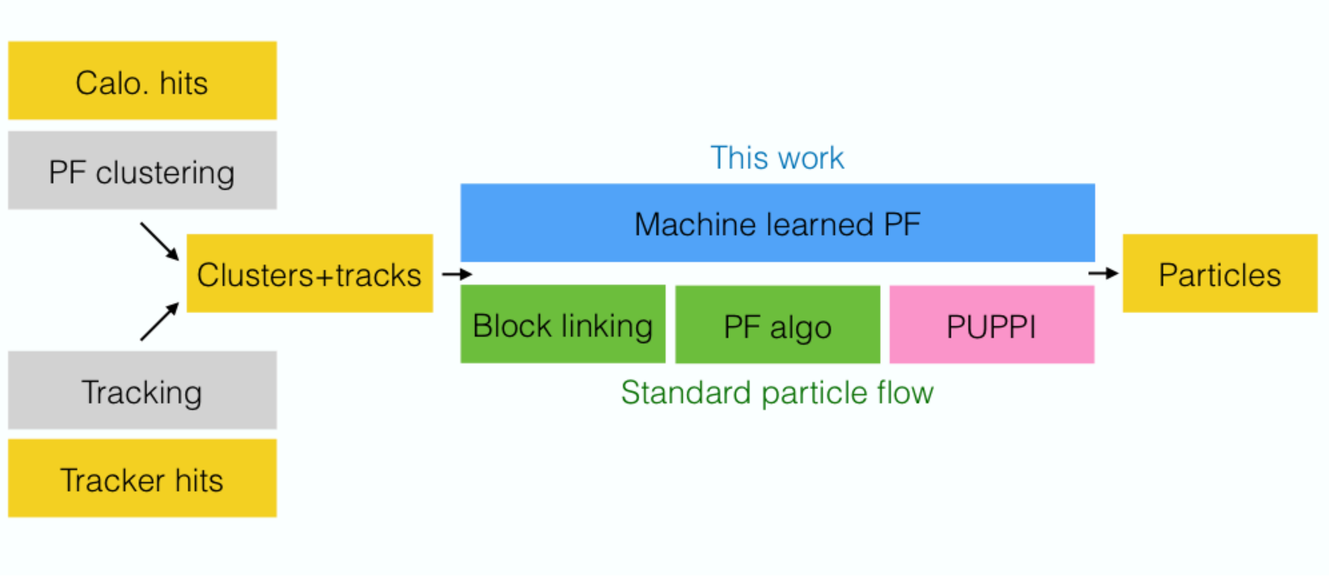



**Energy resolution of PF-based jets compared to calorimeter-based jets**

# Machine Learning Particle Flow

- **Machine learning** (ML) algorithms are known to be excellent in many **pattern recognition**, **classification**, and **regression** tasks
- CMS provides a unique opportunity to study **ML-based PF (MLPF)** in addition to **traditional rule-based PF**
- The MLPF algorithm is tasked with **particle identification, momentum regression** (same as the traditional rule-based PF), and in addition, the possibility to **identify particles from pileup vertices**




**Workflow of PF and MLPF**

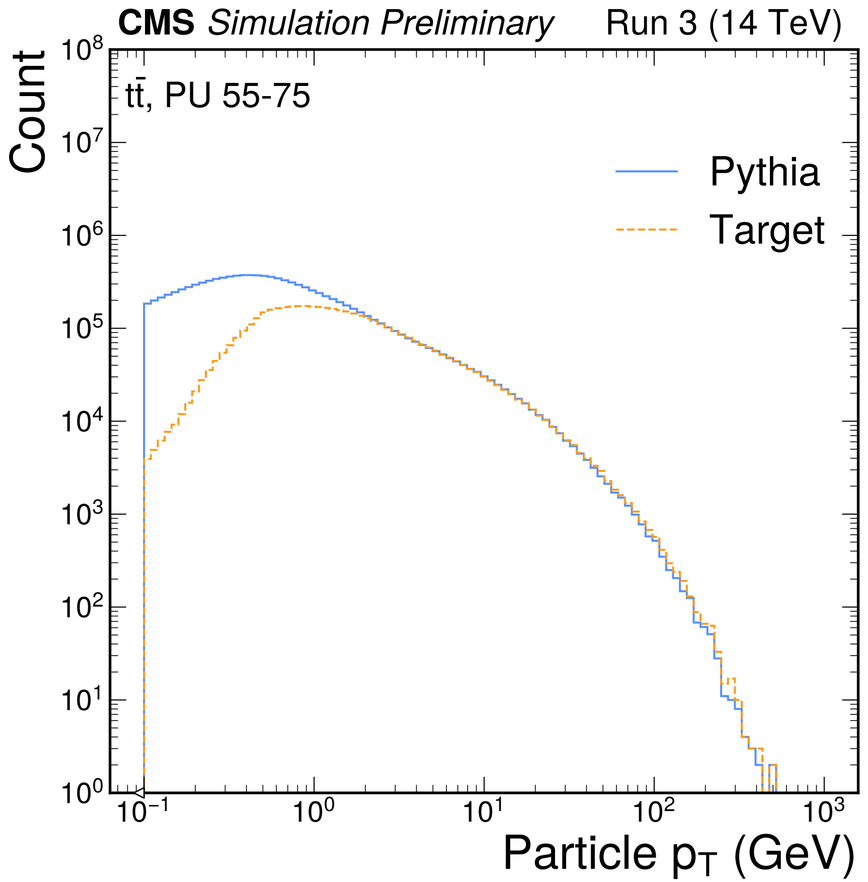
# Target Definition and Training

# Particle Target

- The **MLPF training target** closely approximates the **stable particles** from **Monte Carlo (MC) generator** that the GEANT4-detector simulation receives to leave simulation hits

- The **four momentum** of each target particle and their **fraction of energy from pileup vertices** are recorded

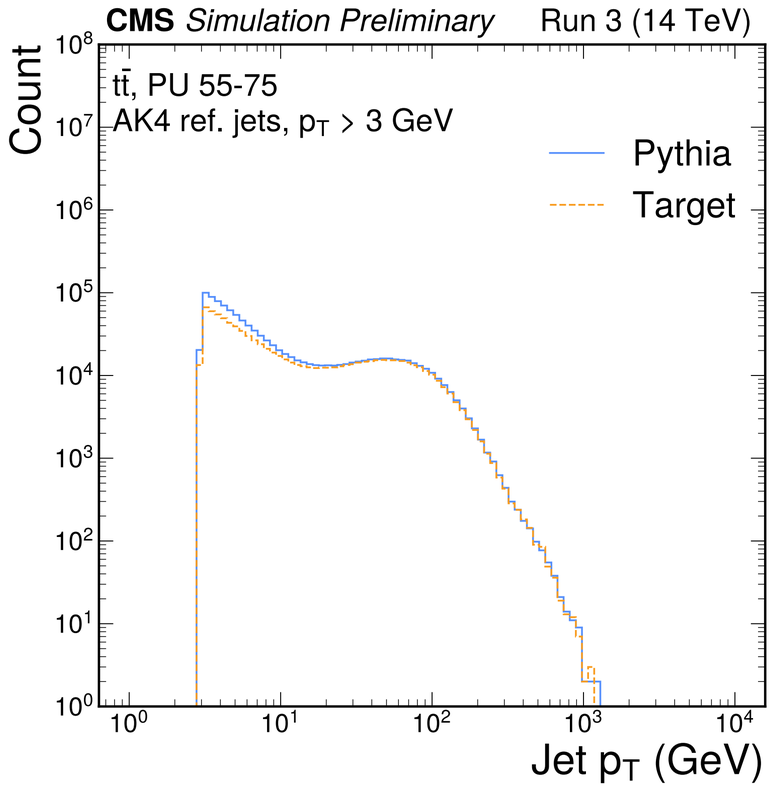
- The defined target is cross-checked against the stable PYTHIA particles

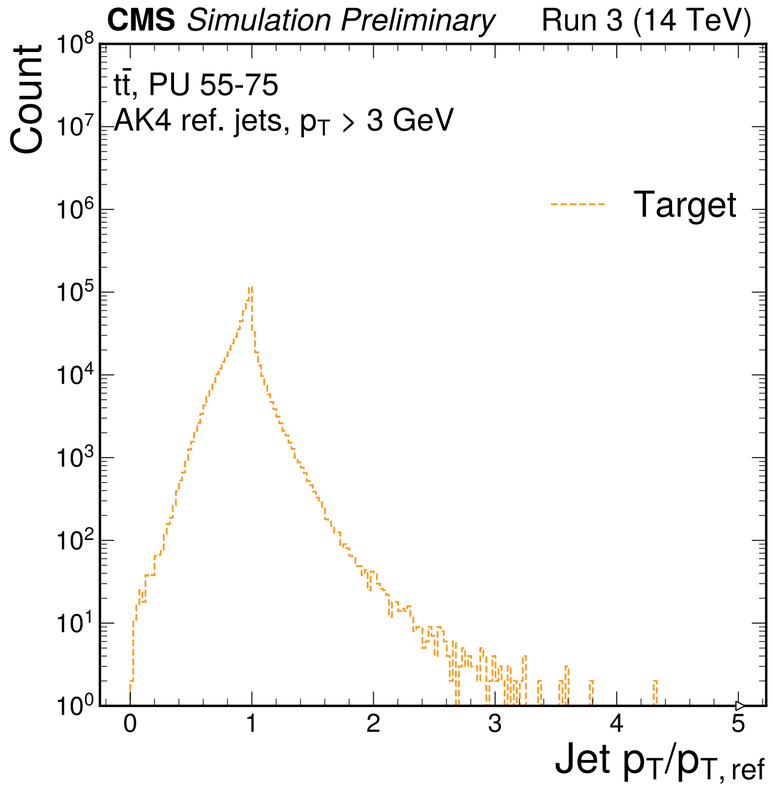



**Disagreements at low $p_T$ come from the limited detector simulation/ reconstruction acceptance**

# Particle Target

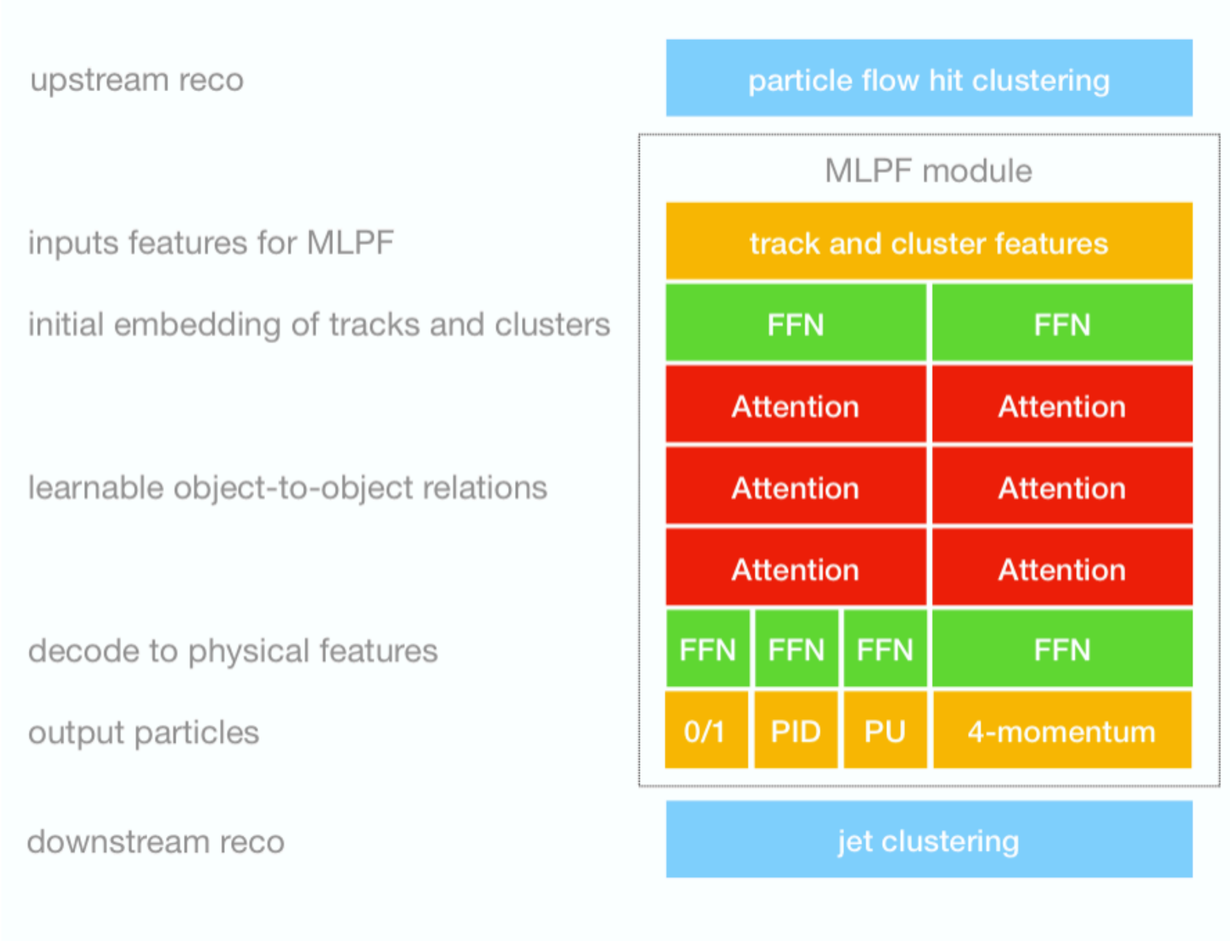
- The **target particle** is also cross-checked against the **stable PYTHIA particles** at jet-level with the **anti-$k_T$ jet algorithm** with a cone size of **0.4** (AK4)



# MLPF Model

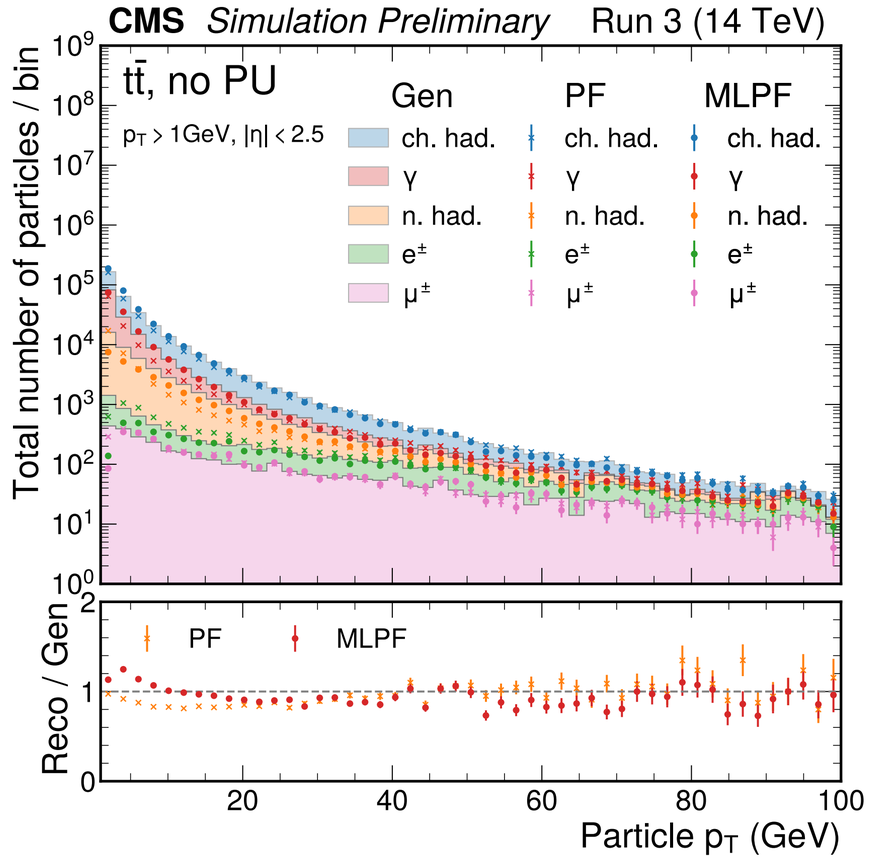
- MLPF employs a **scalable transformer-based model** that takes track and cluster features as input (same as PF)
- The model predicts the **target particle IDs**, their **four-momentum**, and their **probability to be from pileup vertices** (with pileup energy fraction = 1)
- Target particles are associated to either their **primary track** or **cluster** to compute a **per-particle loss**
- The current model has **~4M parameters** and is trained on a mixture of **simulated QCD, $t\bar{t}$, and $Z \to \tau_h\tau_h$ events with and without pileup overlayed**

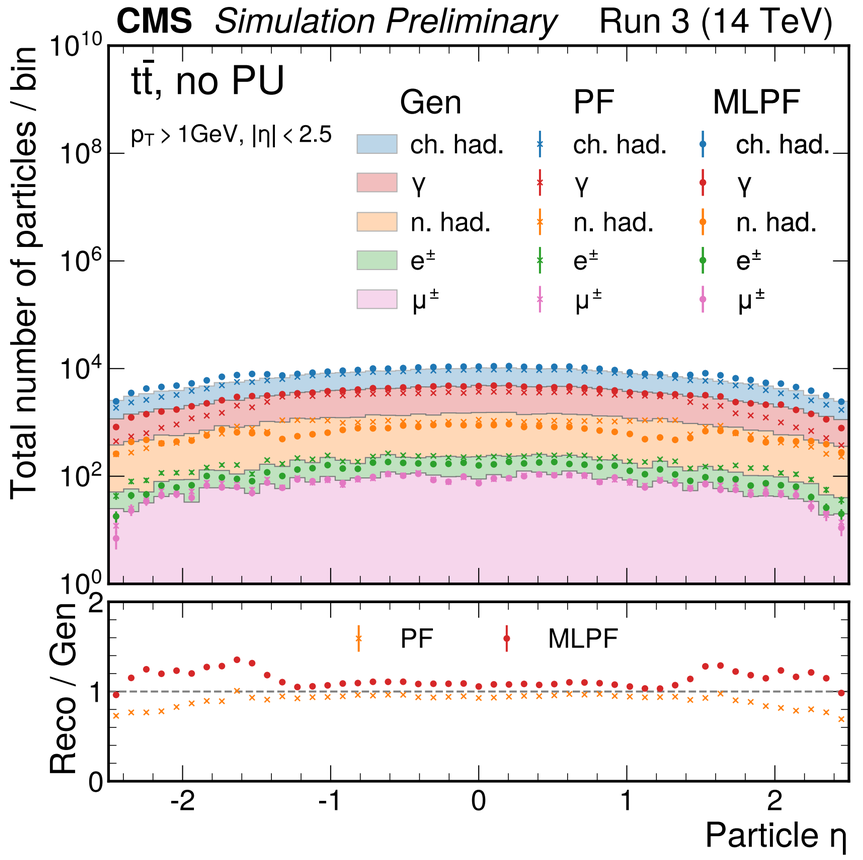


# Performance

# Particle Kinematics

- MLPF shows realistic particle-level kinematics in simulated $t\bar{t}$ events without pileup



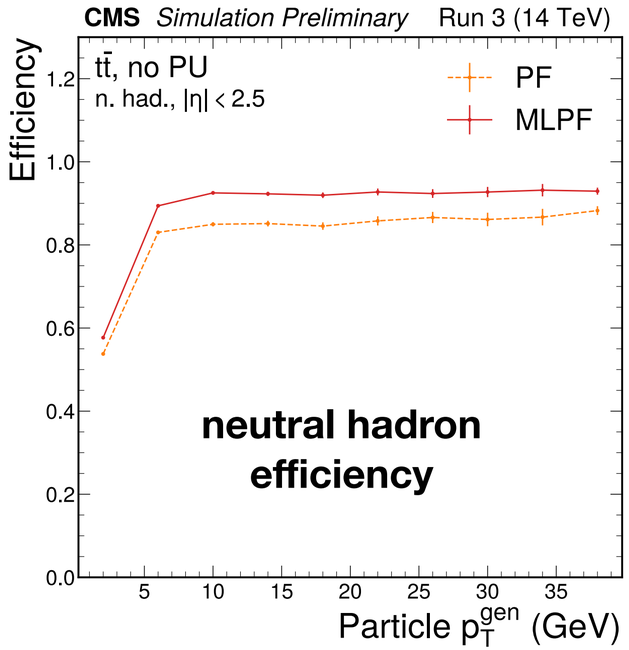
# Particle Level Performance

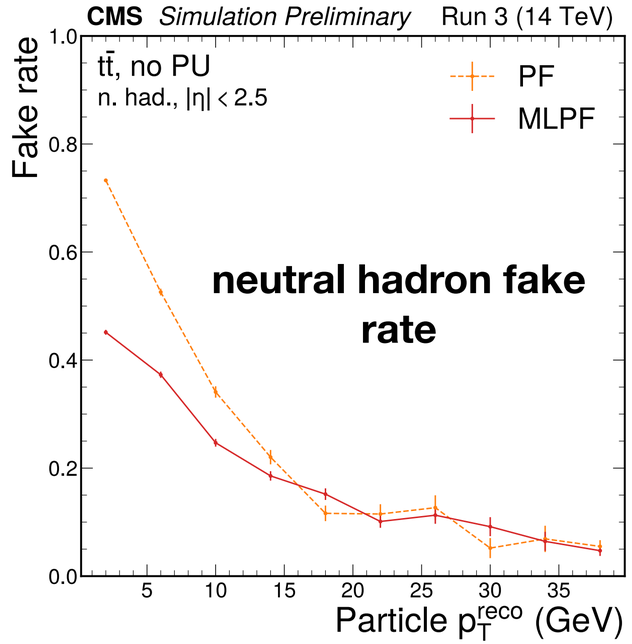
- PF or MLPF reconstructed particles are matched to stable generator particles within

$$\Delta R \equiv \sqrt{\Delta\eta^2 + \Delta\phi^2} < 0.15$$

- We study
  - the **efficiency** of particle reconstruction as the **fraction of stable generator particles reconstructed by PF or MLPF**,

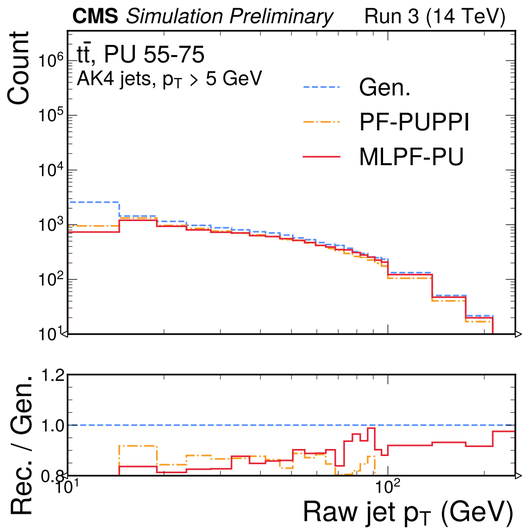
  and
  - the **fake rate** as the **fraction of PF or MLPF candidates without a matched generator particle**

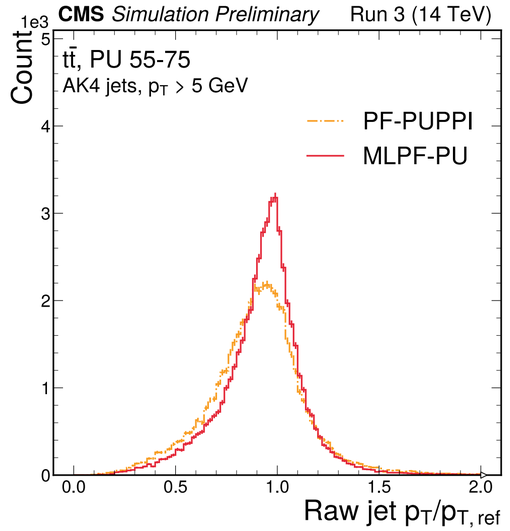


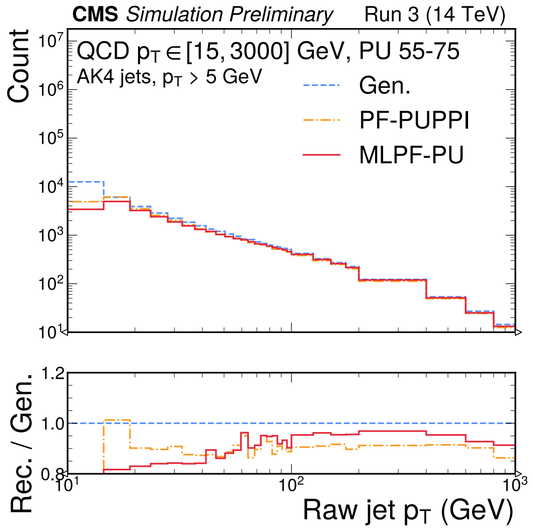



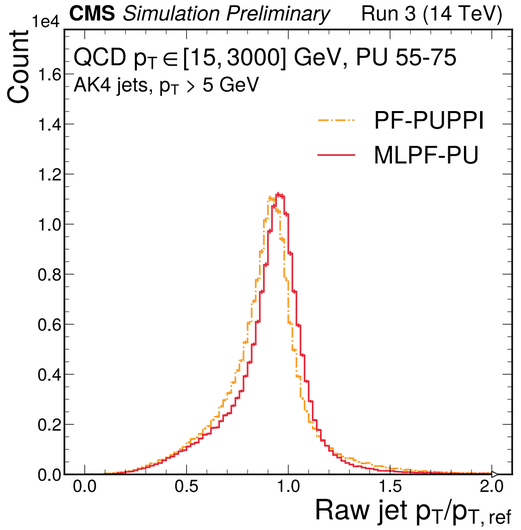
# Jet Performance - AK4

- We study the energy response of AK4 jets (ratio of reconstructed-to-generator level $p_T$)
- For **PF**, jets are clustered from the **PF candidates**, and with the **PUPPI algorithm** applied for pileup mitigation
- For **MLPF**, **MLPF candidates** are clustered after **discarding the identified pileup particles**
- Notice MLPF **ONLY** performs **particle reconstruction** but not jets



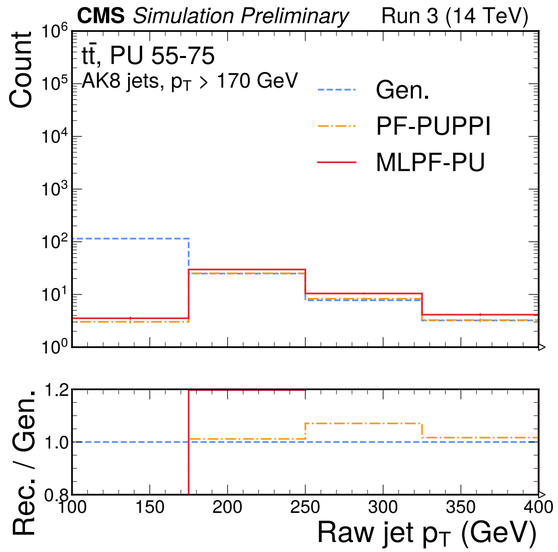



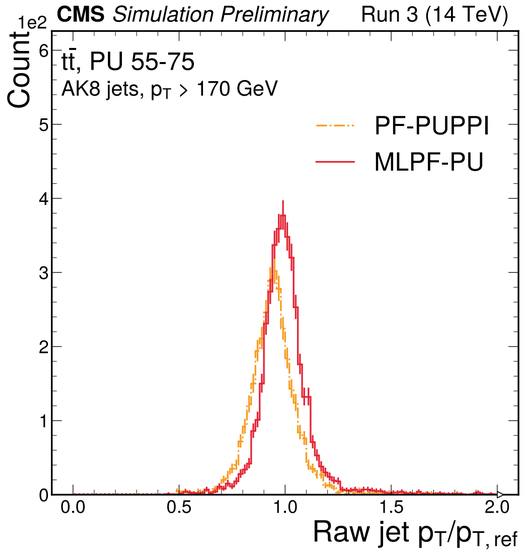



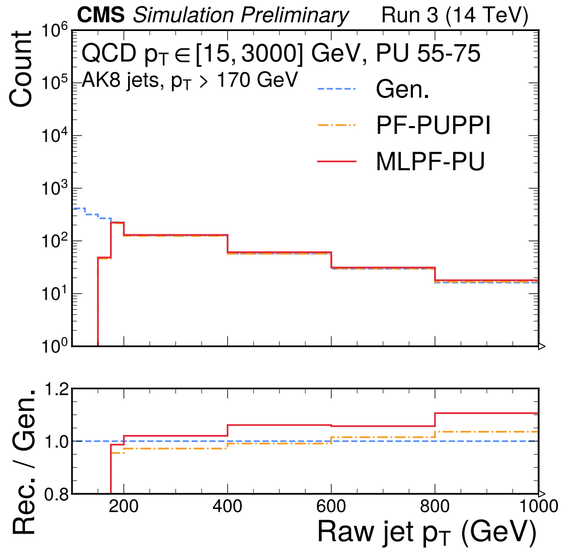



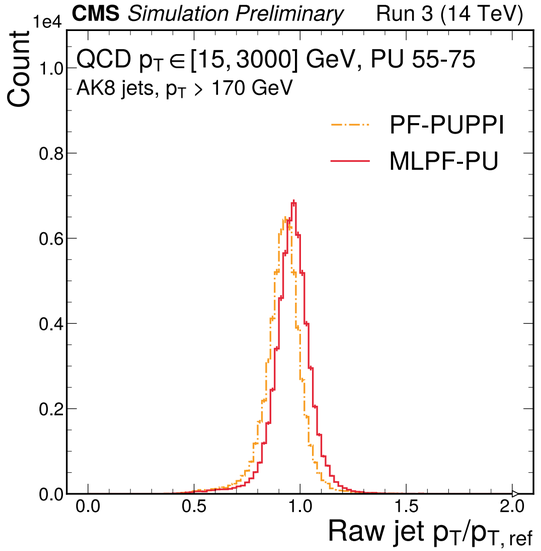
# Jet Performance - AK8

- Similar checks are performed for **anti-$k_T$ jets algorithm** with a cone size of **0.8**



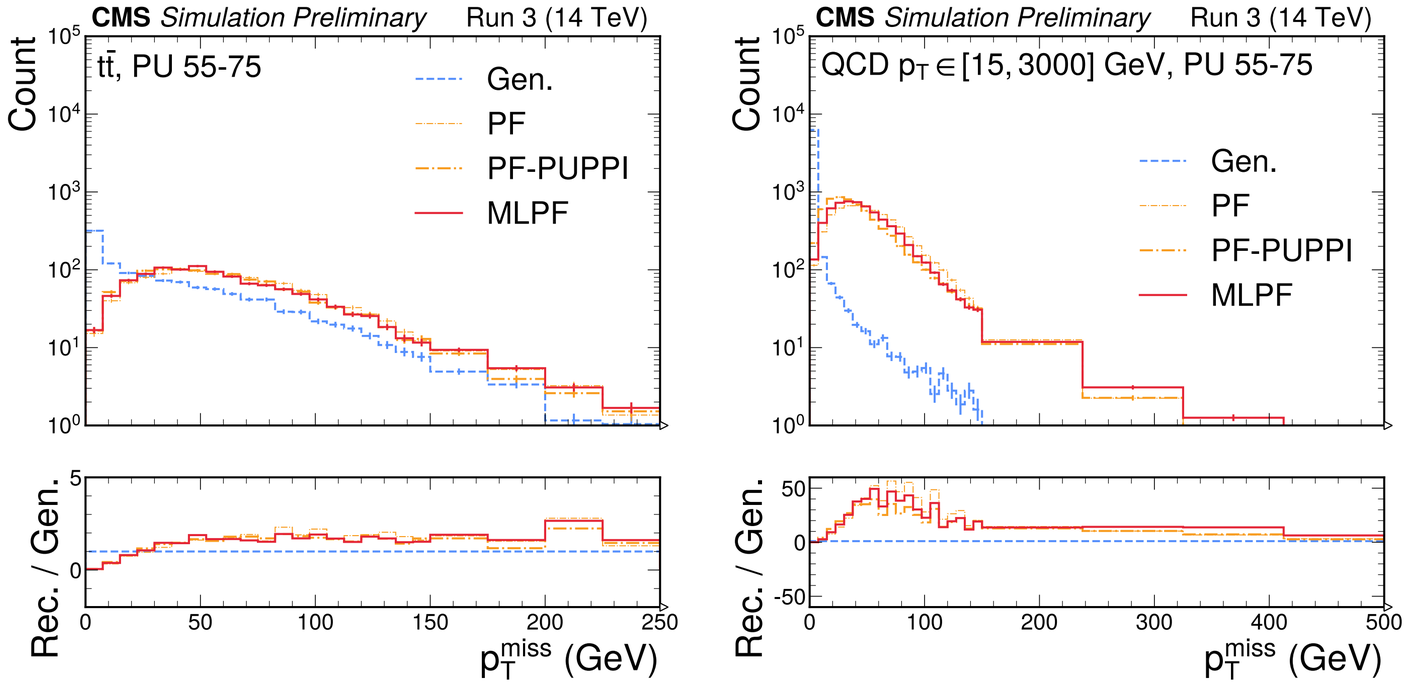



# Missing Transverse Energy

- The **missing transverse energy** (MET or $\vec{p}_T^{\,\text{miss}}$) is defined as the **negative vectorial sum of the** $\vec{p}_T$ of all reconstructed particles

- The MET evaluated from the PF and MLPF candidates are compared to the MET at generator level

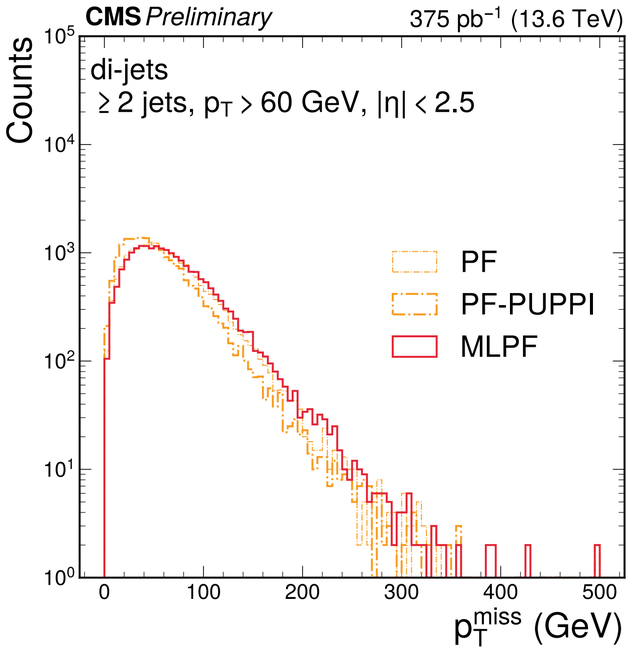
- Notice MLPF **does NOT** do MET regression directly

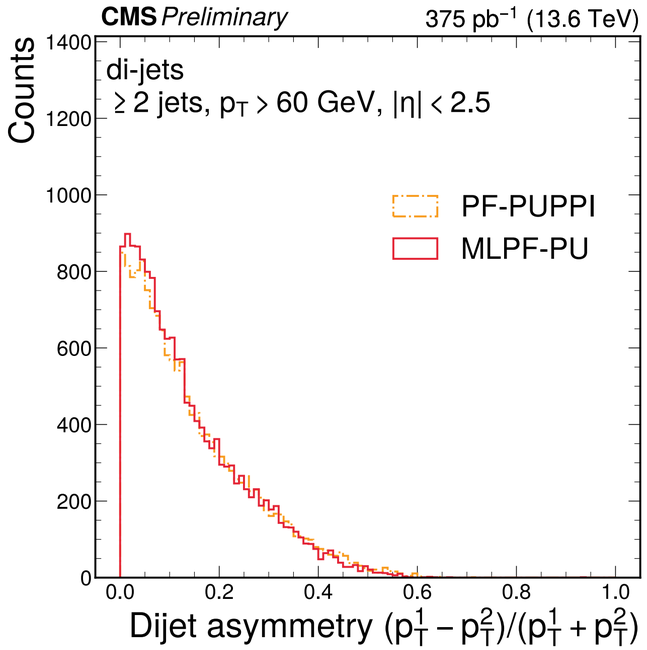



**MLPF MET is consistent with the PF MET; differences in generator and reco MET come from the fiducial cuts in simulation/ reconstruction, and pileup contamination**

# Commissioning on Data

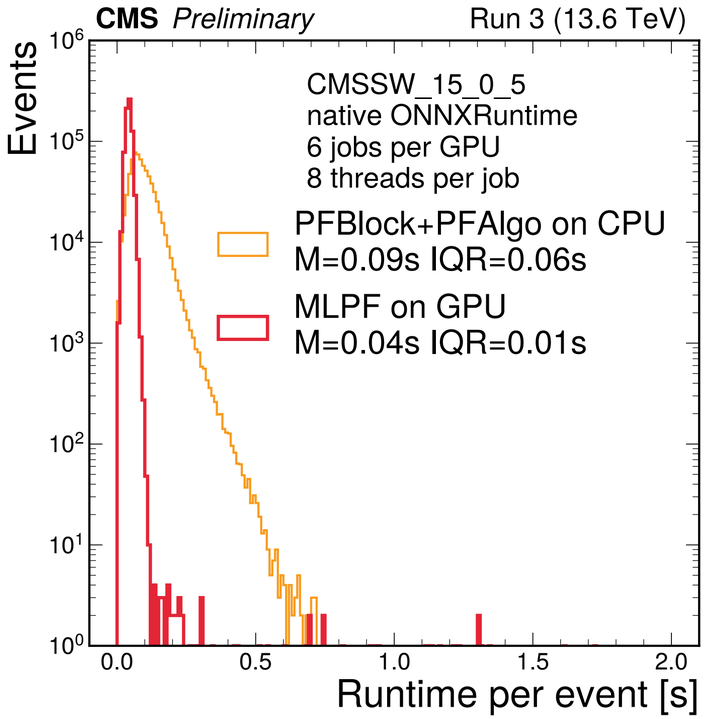
- The MLPF algorithm is **commissioned** on a subset of **proton-proton collision data** collected at CMS in 2024
- The data is collected with a **di-jet trigger** with requirements of
  - $p_T > 125\ (45)$ GeV on the leading (sub-leading) jet, and
  - the invariant mass of the two-leading jets > 1.2 TeV
- Distribution of **MET** and **di-jet asymmetry** in PF and MLPF are consistent

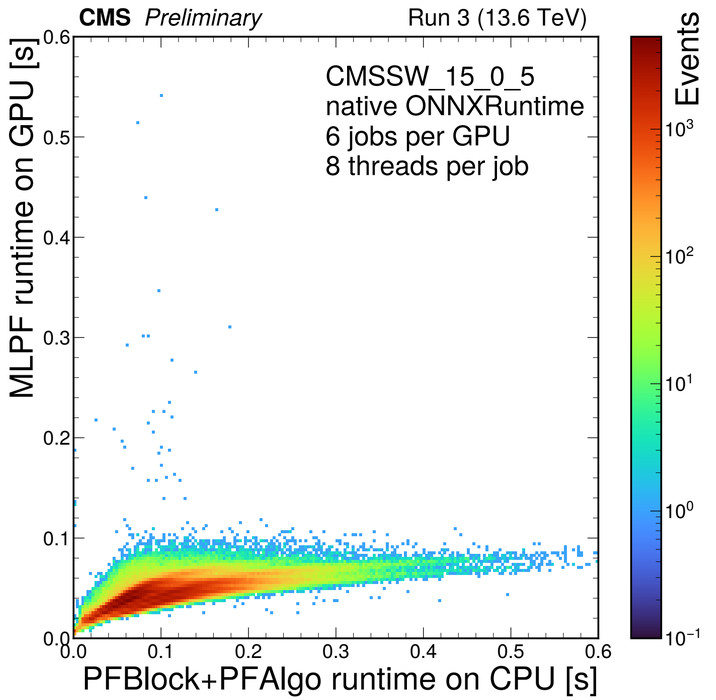


# Runtime Performance

- The MLPF network is exported to a static **ONNX** file and integrated with the **central CMS software** (CMSSW)
- The **runtime performance** is evaluated on a small set of Run 3 data with an average of **37 ± 8 vertices**
- Each reconstruction job is configured to use 8 CPU threads, and 1/7th of a A100 80GB GPU on a dedicated node
- The PF algorithm runs on CPU while the MLPF algorithm runs on GPU



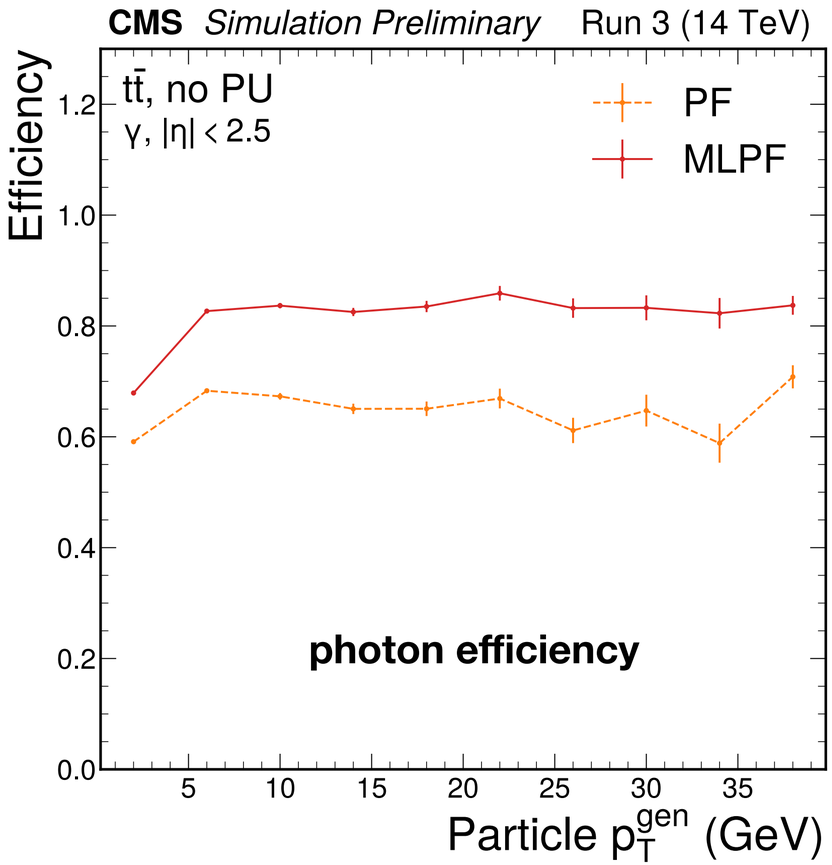



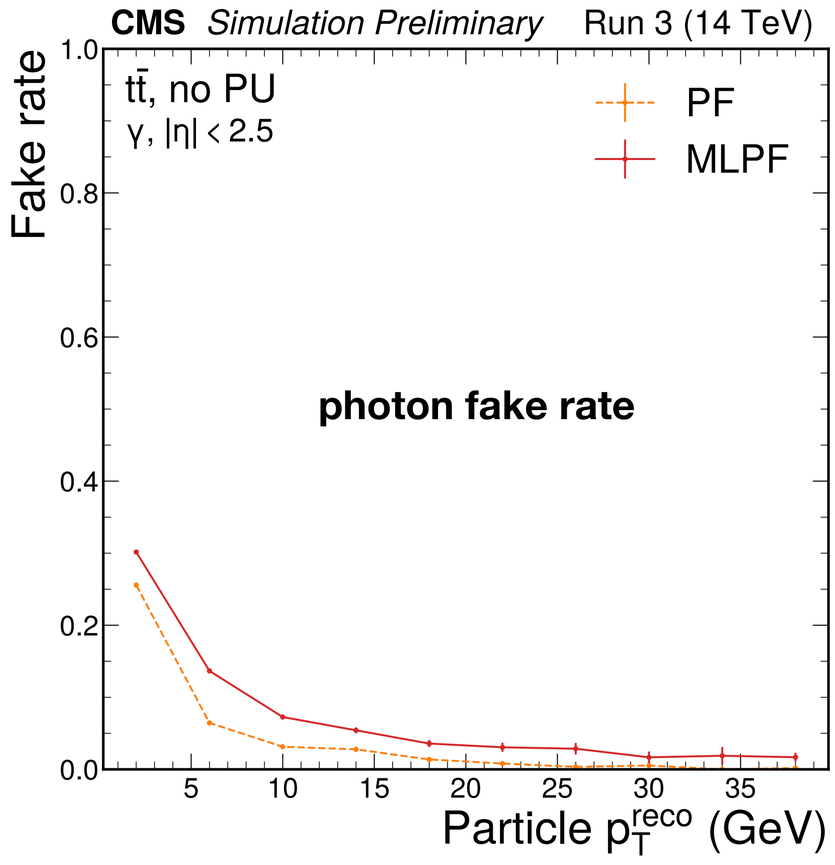
# Summary

- **Particle Flow with Machine Learning** (MLPF) is an alternative to traditional rule-based Particle Flow (PF) with **supervised learning** on Monte Carlo (MC) simulations
- MLPF demonstrates **comparable physics performance** to PF in MC simulations
- Commissioning on **2024 CMS data** shows **consistent event distributions** to PF
- **Integration** to the **central CMS software** is successful and demonstrates **faster runtime and flatter scaling on GPUs** than PF
- Further studies of the MLPF performance and optimizations are ongoing
- Detailed reference is available in the CMS Detector Performance Summary DP-2025-033

# Backup

# Particle Level Performance



# Particle Level Performance