MLPF status (PFT-25-001)

Joosep Pata[1], Eric Wulff[2], David Southwick[2], Shivam Raj[6], Maria Girone[2], Farouk Mokhtar[3], Mengke Zhang[3], Javier Duarte[3], Ethan Colbert[4], Yao Yao[4], Mia Liu[4], Yibo Zhong[4], Dylan Ponman[5], Ka Wa Ho[5], Jennifer Roloff[5]

[1]NICPB [2]CERN [3]UCSD [4]Purdue [5]Brown [6]CUA

April 4th, 2025

PF meeting

Reminder of the summary and goals

CMS is in a unique position to test full event reconstruction using ML-based methods on real data.

- ✅ Run natively on computational accelerators (AMD, Nvidia, Intel Habana tested)
- ✅ Extensible with new inputs or under new conditions (CMS, CLIC, FCCee-CLD tested)
- ✅ Demonstrate realistic event-level performance of ML-based PF reconstruction
- ⏳ Integrate per-particle PU rejection in MLPF
- ⏳ Test full event reconstruction on data

AN-2024/253 status

Revision history

https://cms.cern.ch/iCMS/user/noteinfo?cmsnoteid=CMS%20AN-2024/253

- **Version 1-2 in iCMS**
  - Empty placeholders in iCMS
- **Version 3 (previous v1), 2024-12-16**
  - First draft of the introduction.
  - First draft of the ML simulated-based target definition.
  - Provided `isPU` flag information.
  - Some performance plots in simulation obtained directly from CMSSW.
- **Version 4, 2025-01-17**
  - Updated result plots with a new training: includes $\sqrt{p_T}$ weight term in the regression loss, improves jet scale and resolution over the previous version.
  - The model size has been reduced from 100M to 5M parameters.
  - The CMSSW validation is now based fully on standard JME and BTV NANOAODs that are available to the rest of the collaboration for suggestions.
  - Added AK8 jet performance plots from CMSSW.
  - Removed several outdated .png plots from 2022 and replaced with .pdf from current results.
  - Added input feature distributions.
  - Added an explanatory figure on the model structure.
  - Added particle resolution plots from CMSSW NANOAOD on $t\bar{t}$ +PU.
  - Rewrote the abstract to clarify the purpose of the paper.

Received feedback from Andre Govinda Staahl on 25/02/25

Received feedback from Swagata Mukherjee on 25/02/25

Responses in progress

https://twiki.cern.ch/twiki/bin/view/CMS/PFT25001Review

PubTalk discussion: https://cms-pub-talk.web.cern.ch/c/pft/pft-25-001/745

Progress update

Progress since the last meeting

- Ported recipe to CMSSW_15_0_1, readiness for data validation and fixed ONNXRuntime+CUDA version
- Physics validation done directly on GPU in CMSSW
- New timing study directly in CMSSW
- Progress on per-particle PU rejection

Work in progress

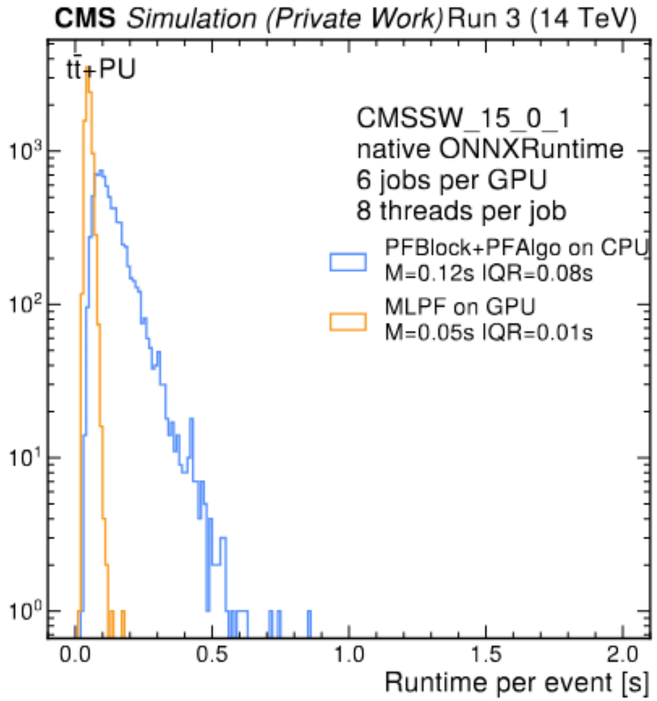
- Address comments: study and explain performance in detector subregions
- Finalize and integrate PU prediction model
- Add option to run inference on SONIC+Triton in CMSSW
- Test on data
- Compile results to paper

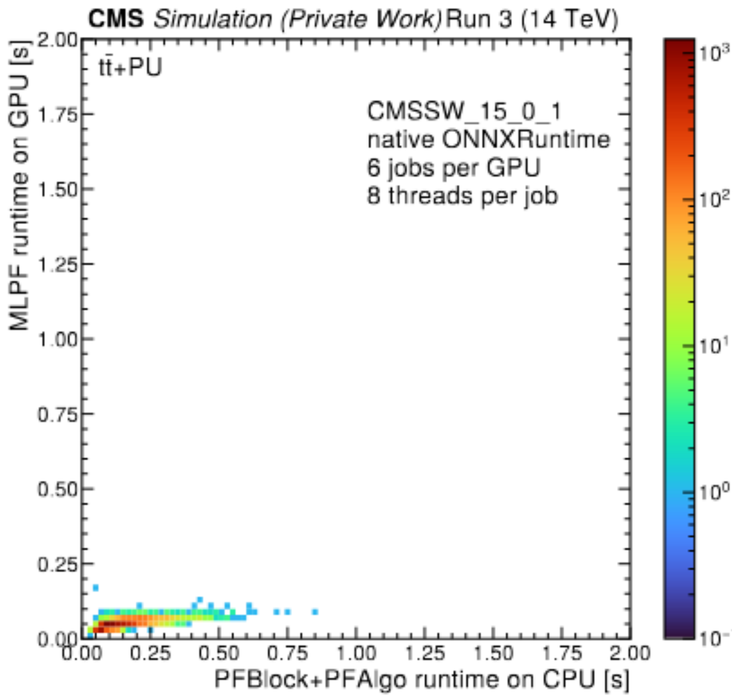
Timing evaluation

new

- We replaced the extrapolated timing plots with a direct measurement

- We measure the performance directly in CMSSW workflows using native ONNXRuntime on GPU (not intended as a full and exhaustive test, rather as an indication of scalability and feasibility)





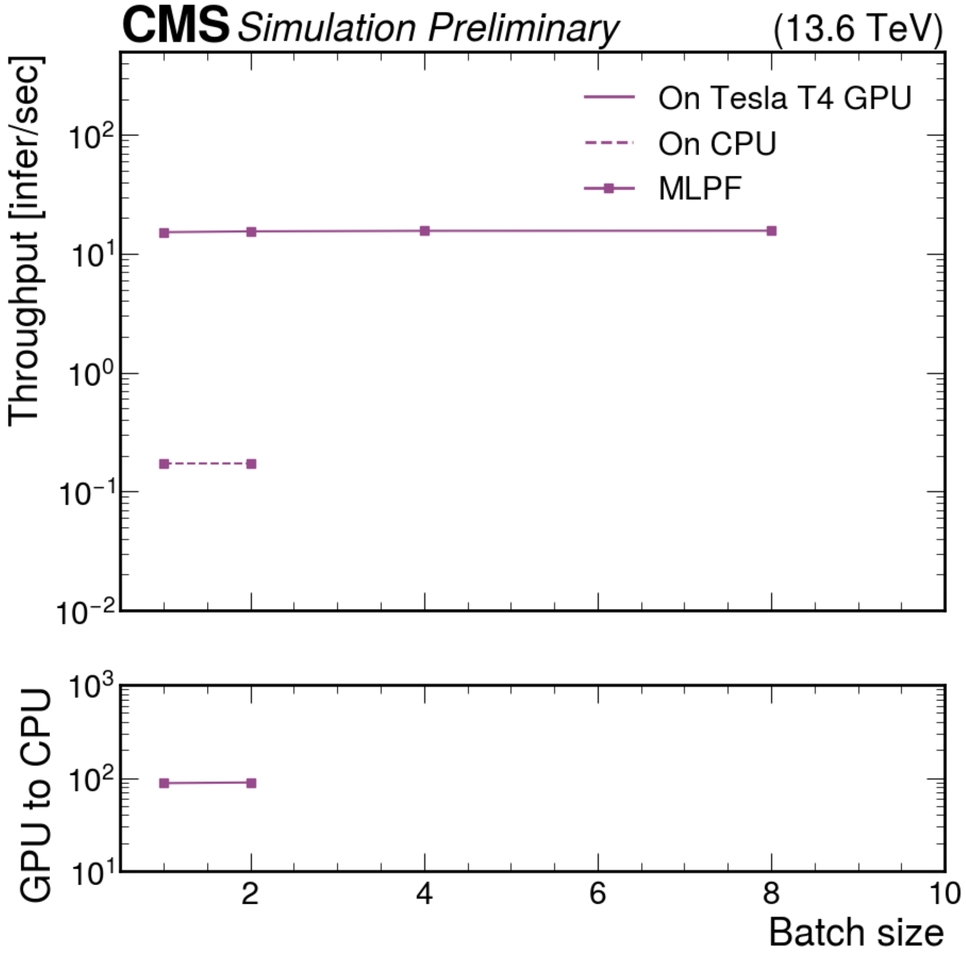
Work ongoing by the Purdue team to evaluate the performance using SONIC/ Triton (next slide)

Figure 17: The runtime of the baseline PF block algorithm (PFBlock) and particle reconstruction (PFAlgo) compared to the MLPF inference, directly in CMSSW using the native ONNX runtime on GPU. We process QCD and $t\bar{t}$ events with a flat pileup profile between 55–75 vertices. A dedicated machine is loaded with 6 parallel jobs, each job using 8 CPU threads via the CMSSW multi-threading framework. For MLPF, each job uses one 1.10g slice of an A100 80GB GPU, corresponding to 1/7th of the GPU capacity. We track the runtime per event of the corresponding reconstruction algorithms using the TimeReport functionality. We find that the median runtime of the MLPF algorithm using the GPU is around 0.05s per event, compared to around 0.12–0.13s for the baseline PF. By correlating the event identifiers, we also find that the scaling of the MLPF inference is approximately flat compared to the PF runtime.

SONIC inference pathway

- SONIC Producer ready: Link
- One inference per event in CMSSW. (Still understanding the batching effect)
- Test with Triton server release 25.01
- Inference result validation: CPU vs SONIC GPU, all output scores are identical, tested with RelValTTbar_14TeV/GEN-SIM-DIGI-RAW/PU_140X_mcRun3_2024_realistic_v8_STD_2024_PU-v2.
- Performance measured on 1 Tesla T4 GPU vs 1 CPU
  - GPU, max throughput ~ 17 infer/sec
  - CPU, max throughput ~ 0.17 infer/sec

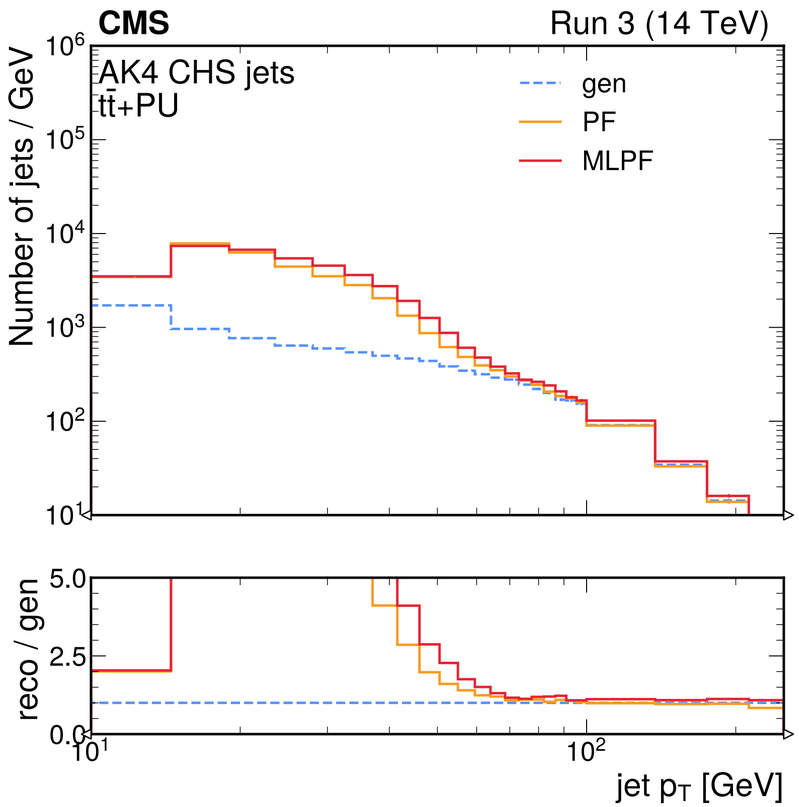
2025/04/02 – D. Kondratyev, Y. Yao

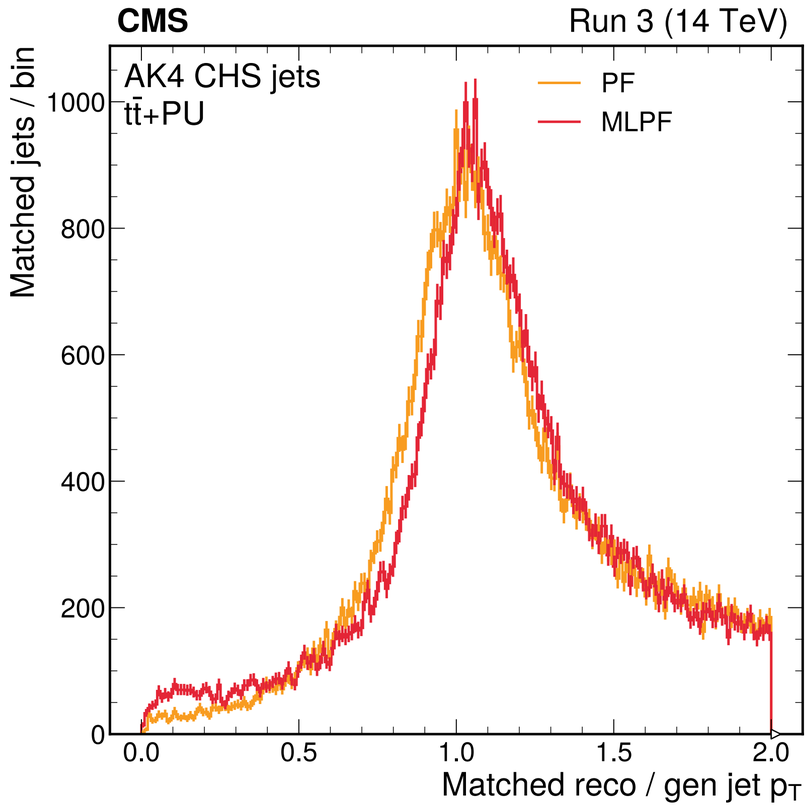


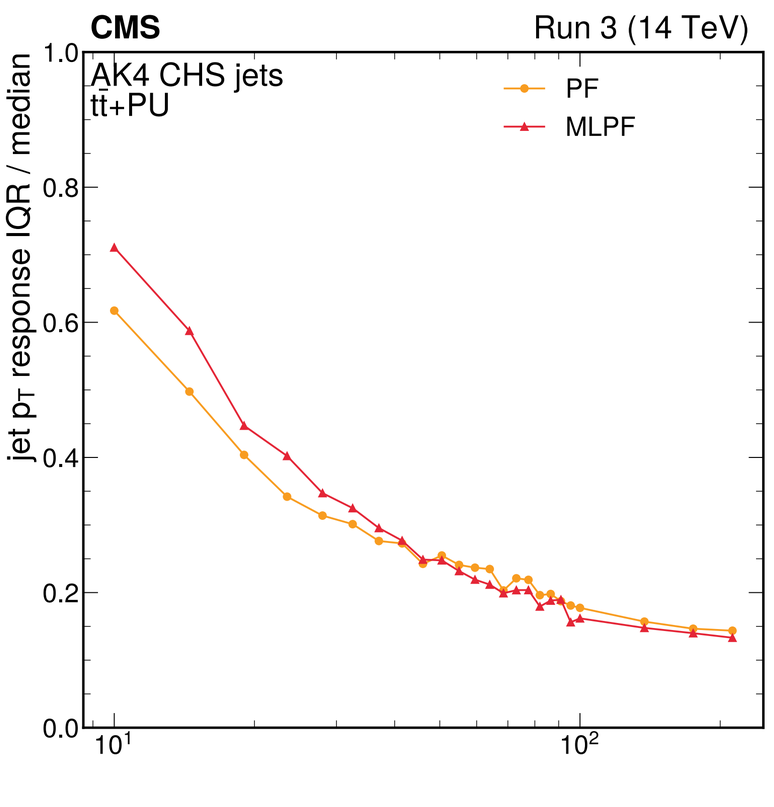
AK4 jets with CHS

updated

Evaluation redone in CMSSW_15_0_1, using ONNXRuntime+GPU





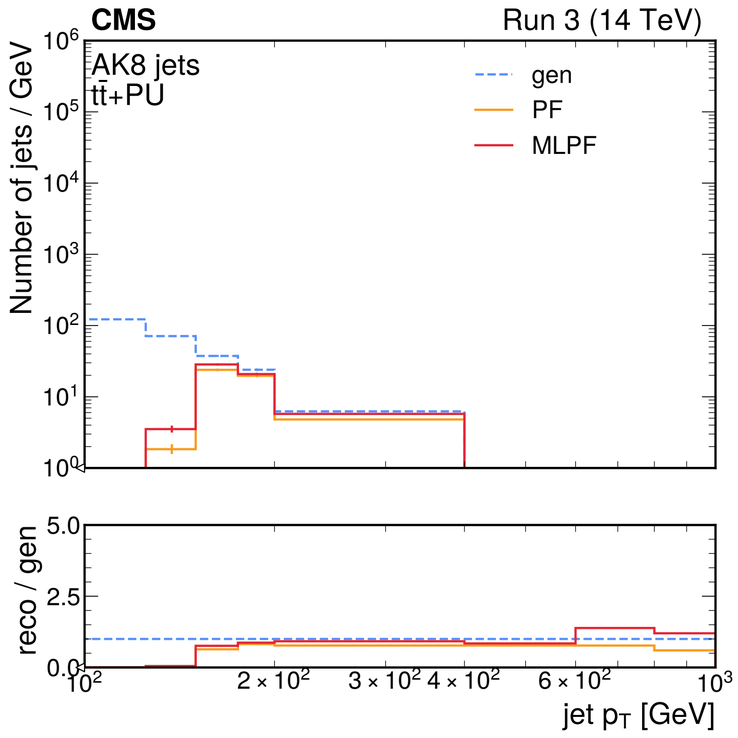


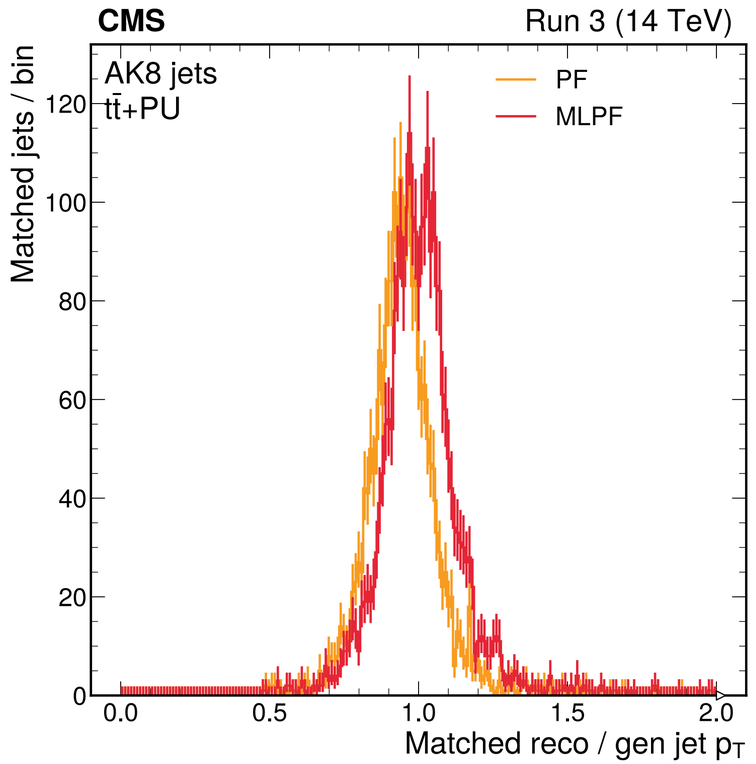
Gen does not include PU, PF and MLPF reconstruct PU + CHS subtraction.
Jet performance in CMSSW, on samples with PU, is largely compatible between PF and MLPF.

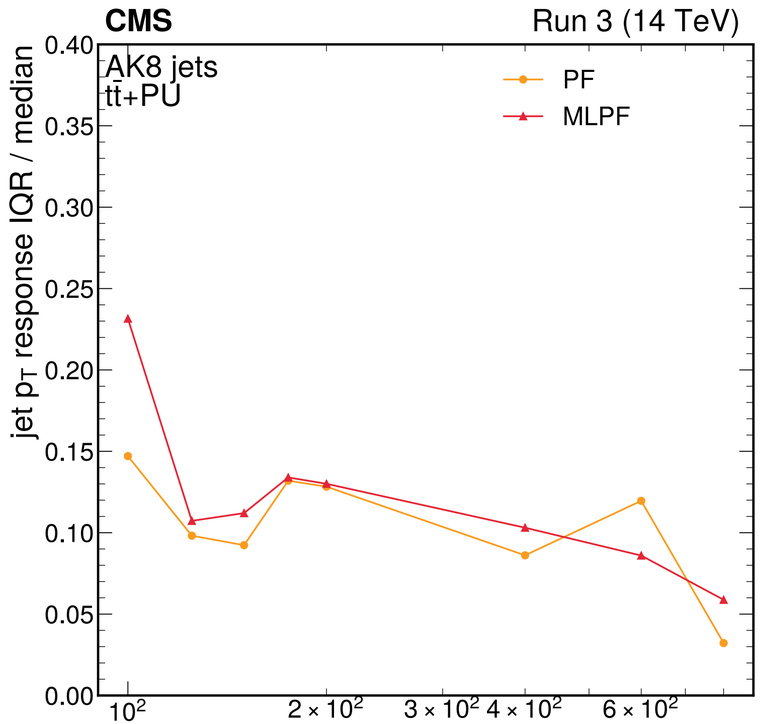
AK8 jets with CHS

updated

AK8 (nor AK4) jets never explicitly trained for, an additional validation of the model







By construction, gen does not include PU, PF and MLPF reconstruct PU + CHS subtraction. Compatible performance

Uncalibrated MET

updated

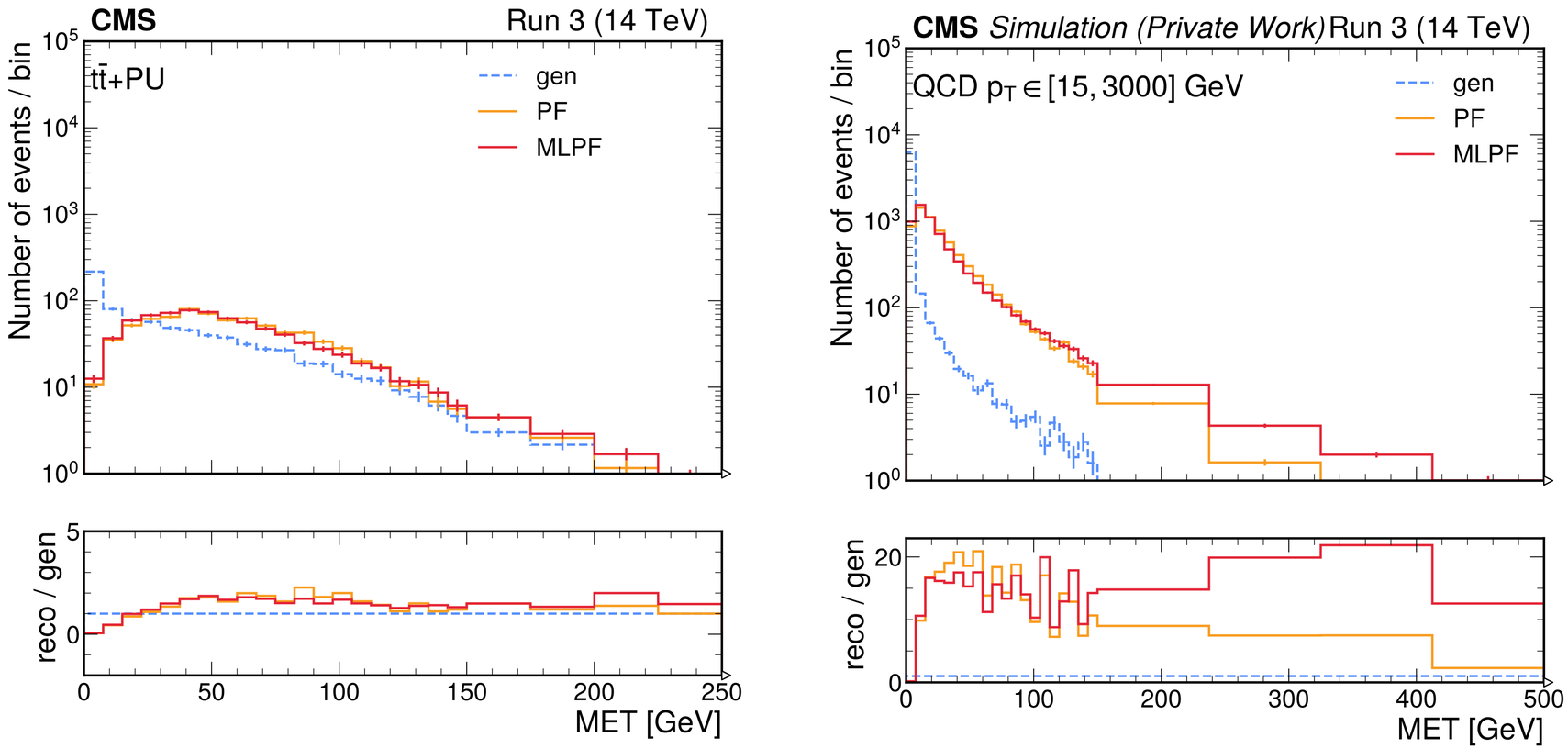


Figure 31: For paper. Uncalibrated MET in samples with pileup, using CMSSW +ONNX. The differences with respect to generator-level MET are due to generator-level not including PU, while PF and MLPF do include PU. We find the MET response to be consistent between PF and MLPF in $t\bar{t}$, while in QCD, the response in MLPF is somewhat worse due to limited training data for high-p_T particles

PU rejection

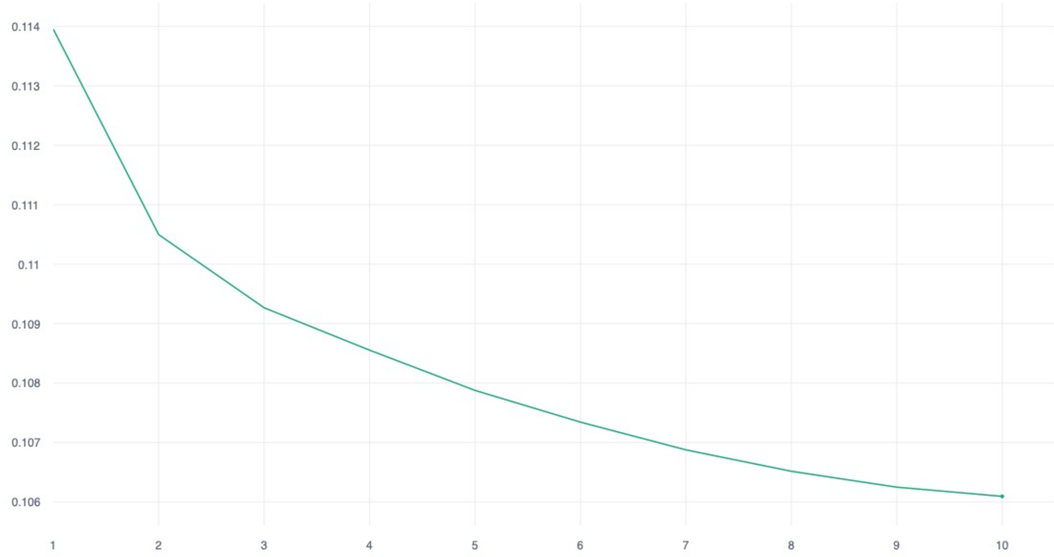
- Target pileup label is added for each particles, defined as the fraction of energy contributed by pileup particles, ranging from 0 (prompt) to 1 (pileup)

- A FFN is added to “classification side” of MLPF for binary classification node of PU (whether the PU energy fraction is 1 or not)

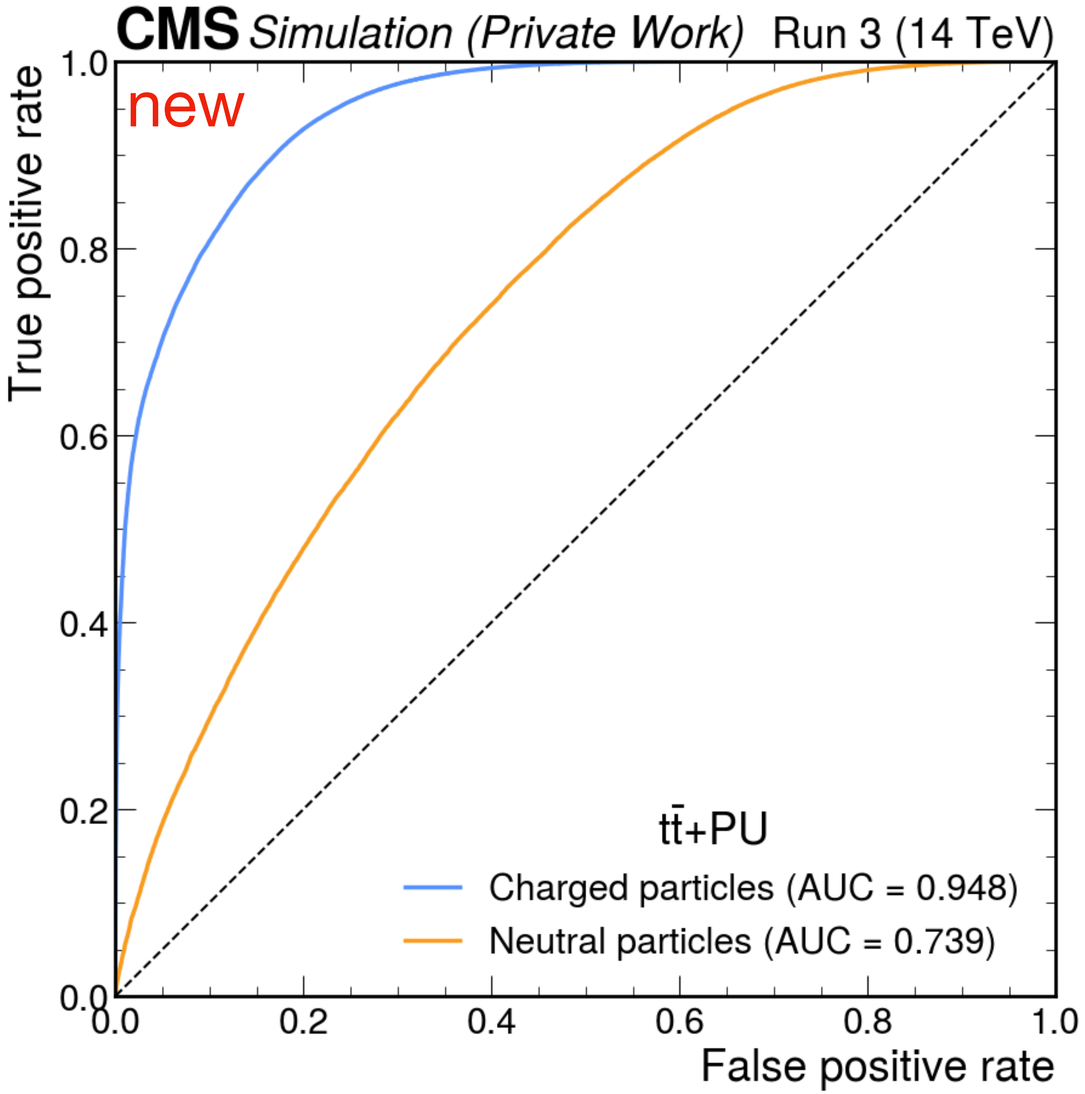
- A binary cross entropy loss is utilized

- At the moment, we load pre-trained MLPF weights without the PU FFN to train the full MLPF with a mix of 10k events of QCD [15, 3000] GeV and $t\bar{t}$ events for 10 epochs

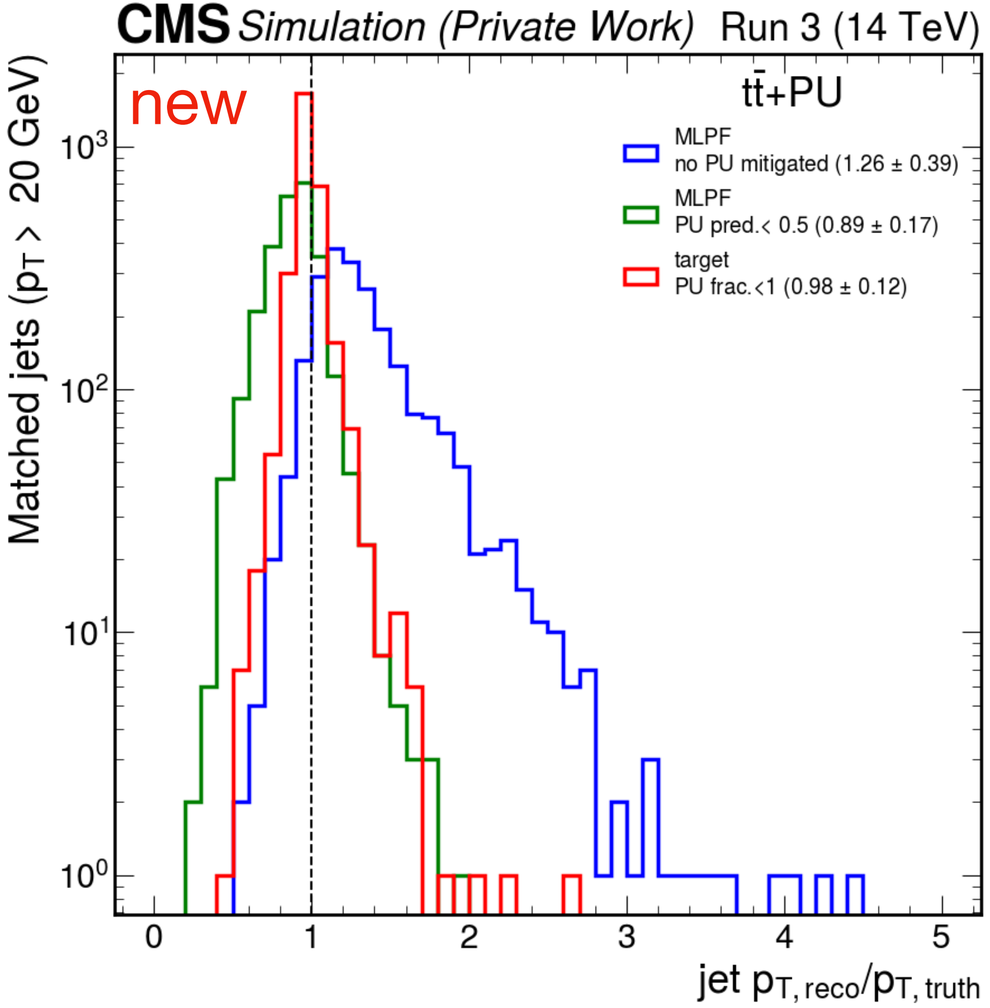
PU classification loss new



Work by Ka Wa, Dylan, Jennifer from Brown



Expected performance of MLPF-based PU rejection for charged and neutral hadrons

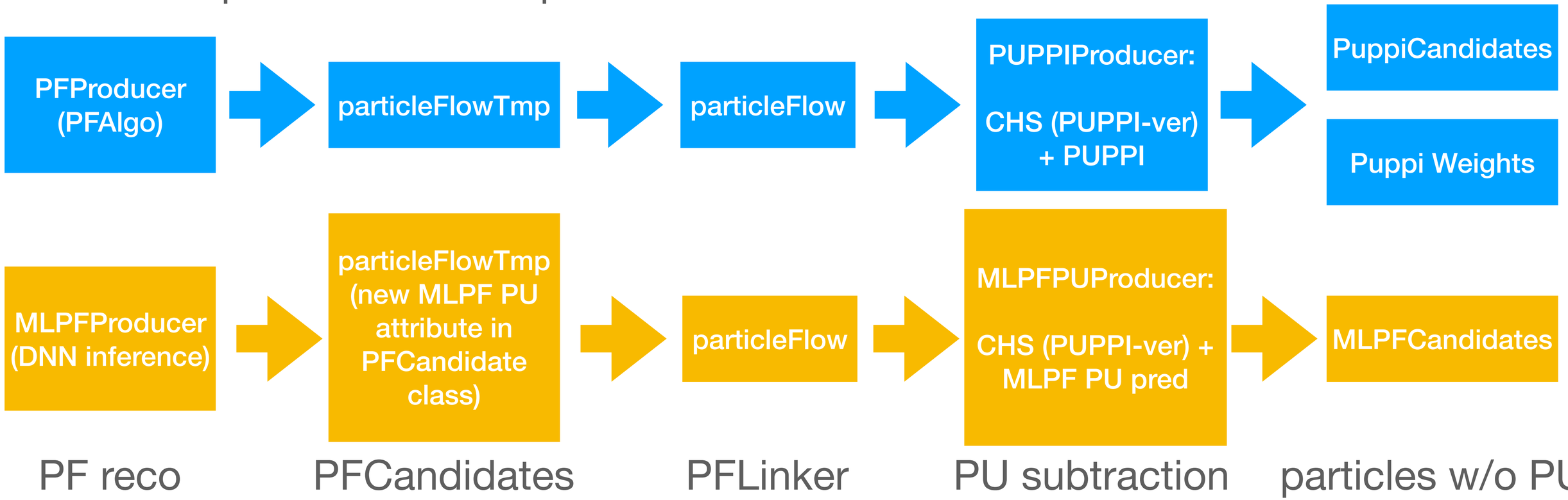


Applying the MLPF-based PU rejection model (green), we can significantly improve jet performance, nearly matching ideal performance (red)

PU model: next steps

new

- Study and validation of the performance ongoing
- Integration of the new per-particle PU flag from MLPF with CMSSW ongoing]
- Plan to compare MLPF + PU prediction with PF + PUPPI



Summary

- Model structure, training setup and datasets are generally frozen
- Updated CMSSW recipe to 15_0_1, general readiness to test on data
- New timing and physics validation directly on GPU in CMSSW using ONNXRuntime
- Progress on integrated PU rejection
- Progress on an optional SONIC/Triton inference pathway
- Work ongoing on comments by Swagata & Andre for AN-2024/253-v4

Pre-training helps to generalize to new detectors

Separately from the CMS developments, we have recently tested the model across detector designs (CLIC → CLD). Shared model/training codebase with CMS.

Fine-tuning machine-learned particle-flow reconstruction for new detector geometries in future colliders

Farouk Mokhtar [1,*] Joosep Pata [2,†] Dolores Garcia [3] Eric Wulff [3] Mengke Zhang [1] Michael Kagan [4] and Javier Duarte [1]

[1] *University of California San Diego, La Jolla, USA*
[2] *National Institute of Chemical Physics and Biophysics, Tallinn, Estonia*
[3] *European Center for Nuclear Research (CERN), Geneva, Switzerland*
[4] *SLAC National Accelerator Laboratory, Stanford, USA*

(Dated: March 25, 2025)

We demonstrate transfer learning capabilities in a machine-learned algorithm trained for particle-flow reconstruction in high energy particle colliders. This paper presents a cross-detector fine-tuning study, where we initially pre-train the model on a large full simulation dataset from one detector design, and subsequently fine-tune the model on a sample with a different collider and detector design. Specifically, we use the Compact Linear Collider detector (CLICdet) model for the initial training set, and demonstrate successful knowledge transfer to the CLIC-like detector (CLD) proposed for the Future Circular Collider in electron-positron mode (FCC-ee). We show that with an order of magnitude less samples from the second dataset, we can achieve the same performance as a costly training from scratch, across particle-level and event-level performance metrics, including jet and missing transverse momentum resolution. Furthermore, we find that the fine-tuned model achieves comparable performance to the traditional rule-based particle-flow approach on event-level metrics after training on 100,000 CLD events, whereas a model trained from scratch requires at least 1 million CLD events to achieve similar reconstruction performance. To our knowledge, this represents the first full-simulation cross-detector transfer learning study for particle-flow reconstruction. These findings offer valuable insights towards building large foundation models that can be fine-tuned across different detector designs and geometries, helping to accelerate the development cycle for new detectors and opening the door to rapid detector design and optimization using machine learning.

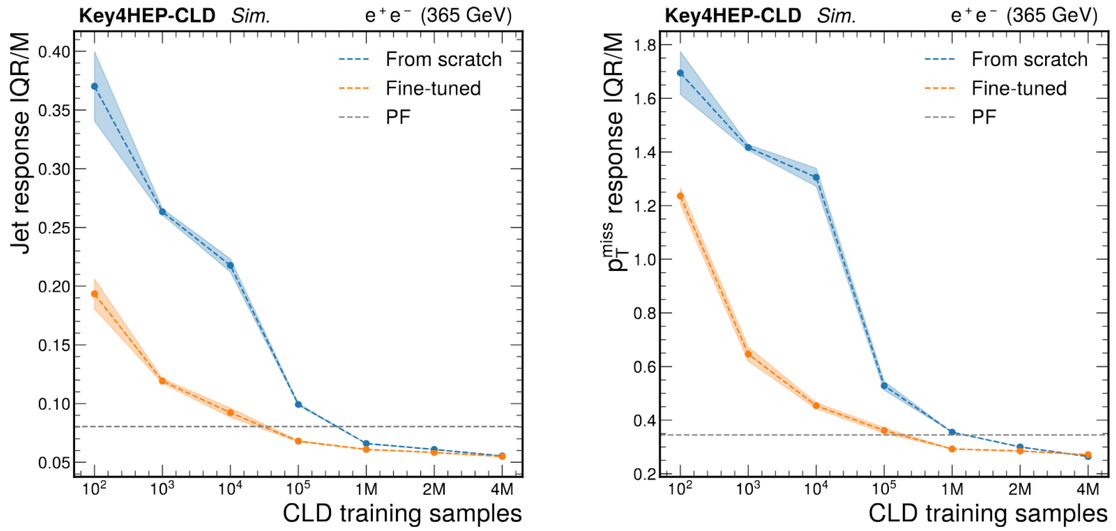
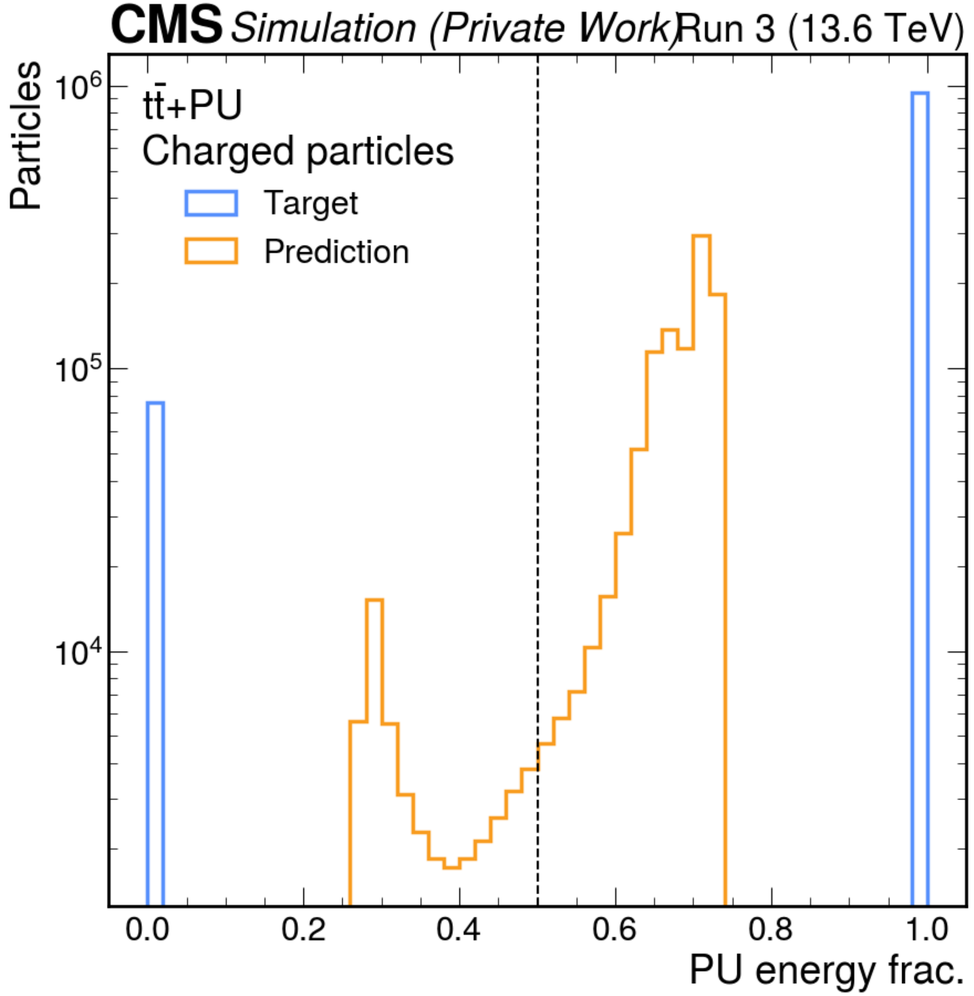
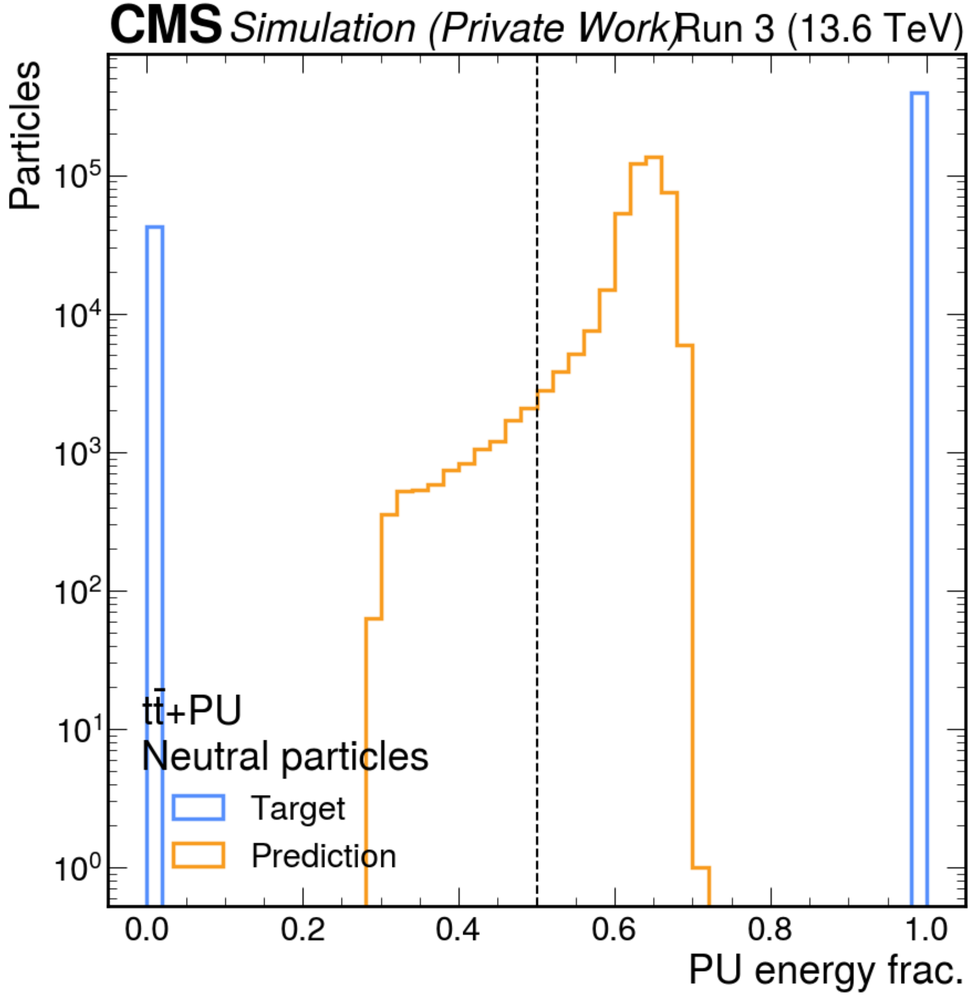


FIG. 7. The jet and $p_{\mathrm{T}}^{\mathrm{miss}}$ performance as a function of the CLD dataset size, for the FINE-TUNED model (orange), FROM SCRATCH (blue), and the traditional PF algorithm (grey). Experiments on < 1M CLD training samples, are repeated three times with different random seeds, and the shaded uncertainty band covers the RMS uncertainty of the three runs, while the dotted line represents the mean performance.

https://arxiv.org/pdf/2503.00131

Backup

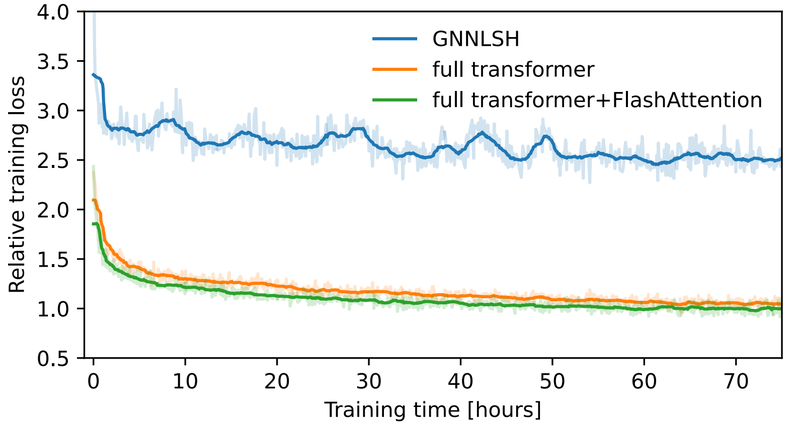




Binary cross-entropy applied on a per-particle basis, continuous predictions matched to binary targets

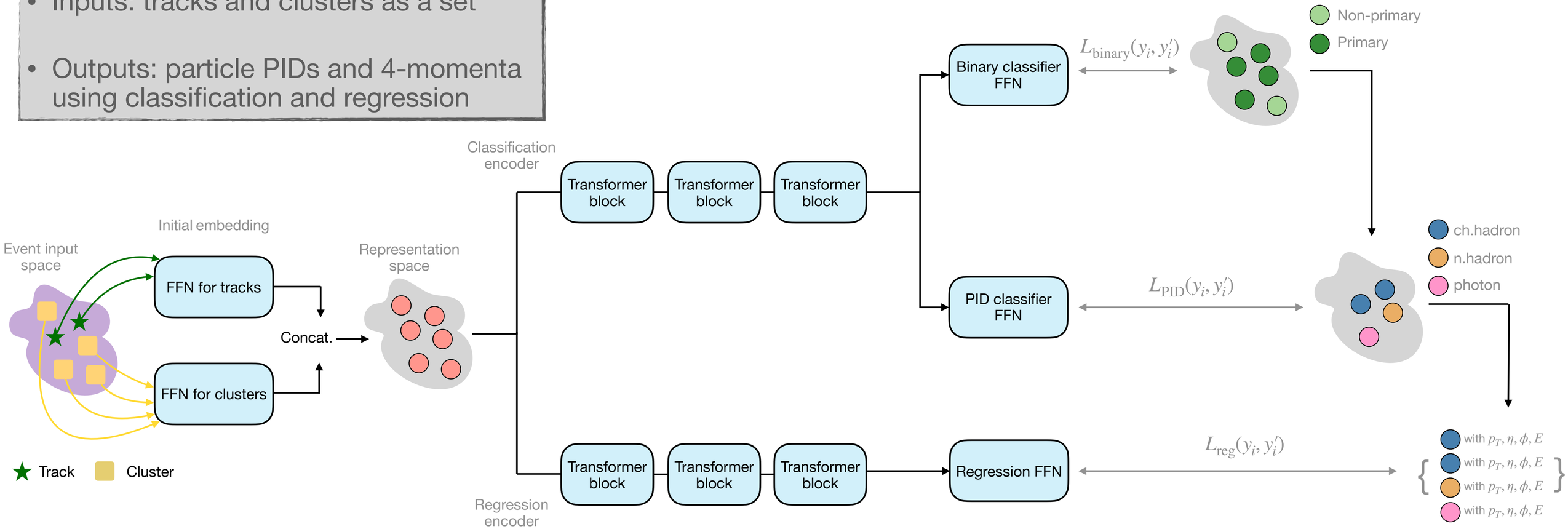
Transformer model

- Transformers can be fast on full events using specialized kernels, i.e. Flash Attention
- Transformer significantly outperforms our previous, GNN-based model, both in training time and absolute loss
- Since 2024, **MLPF is based on training a transformer** with a per-particle classification and regression loss



Rundown of the algorithm

- Inputs: tracks and clusters as a set
- Outputs: particle PIDs and 4-momenta using classification and regression



(a) Encoding (b) Linking (c) Particle Reconstruction with simplified per-particle loss