ML in CMS: new developments and challenges

CHIPP AI/ML workshop 2024

Davide Valsecchi (ETH Zurich)

19/06/2024

ML is an increasingly important part of **CMS trigger, reconstruction and statistical analysis**

→ **Jet taggers**
- Many different architectures under exploration:
- State-of-art performance with transformer models
- Exploring new techniques to make taggers robust against data/MC differences

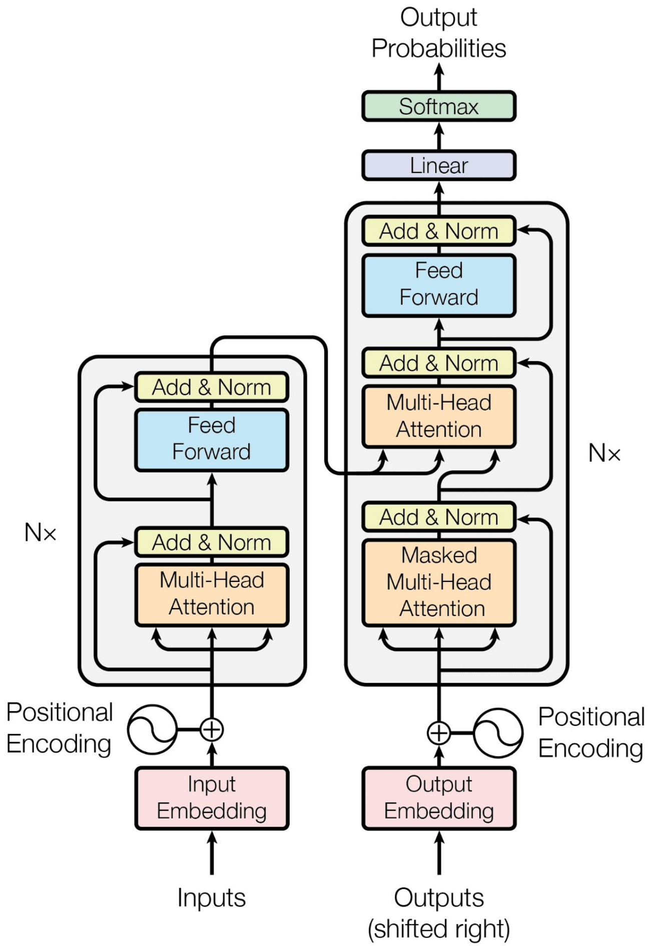
→ **ML particle flow**:
- particle flow based on graph neural network

→ **Reconstruction**
- Complex graph networks for end-to-end detector reconstruction
- Graph networks for object linking and noise/pileup cleaning

→ **Generative models for simulation:**
- Fast simulation based on normalizing flows able to achieve full simulation quality
- Normalizing Flows for data/MC correction and for multidim integrals

→ **Anomaly detection in the trigger:**
- ML algorithms on the FPGAs of the CMS L1 trigger to select anomalous events in new ways

Today I will mention only a biased selection of ML applications in CMS



Transformers and **graph neural networks** are the base architecture for many state of art ML models in CMS:

- handle **variable** set of objects
- no intrinsic **ordering**
- extract information from the **relation** between objects
- possibility to customize the attention mechanism to add physics insight → invariant masses

Natural applications in HEP:

- Particle Cloud tagging (jets)
- Objects linking and particle flow applications
- Full-event analysis → no need to extract high level features

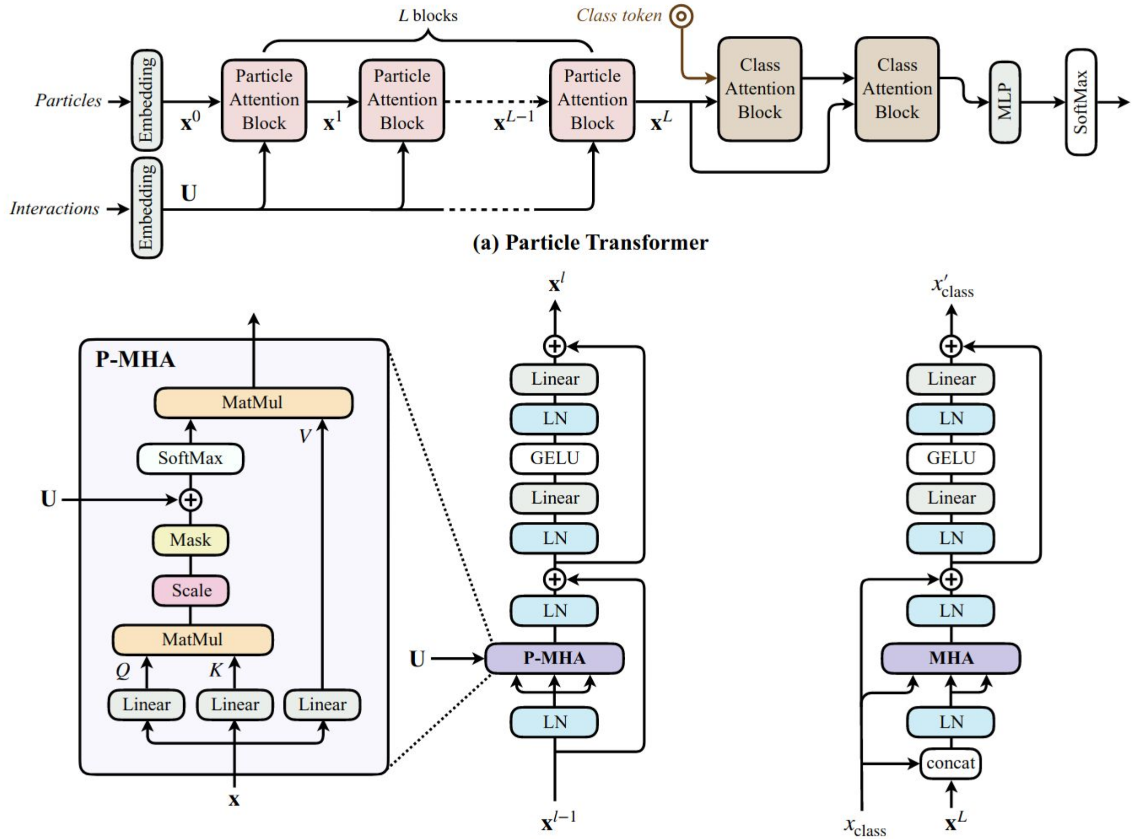
Particle Transformer AK4 (ParT)

Transformer architecture on jet constituents for AK4 jet flavour tagging.

Modified attention layer with custom interaction features

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

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



(a) Particle Transformer

(b) Particle Attention Block

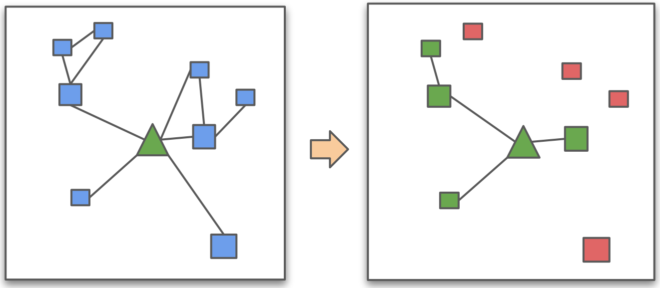
(c) Class Attention Block

Huilin Qu et al 2202.03772

ECAL DeepSuperClustering

SuperClustering in the CMS electromagnetic calorimeter (ECAL):

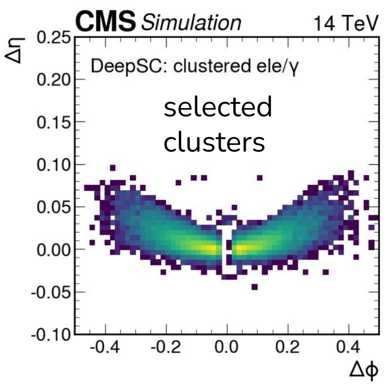
- Linking to recover Bremsstrahlung or photon conversion
- Starting point for ele/gamma reconstruction, ECAL calibration
- Classical algo has high efficiency, but only geometrical + seed energy

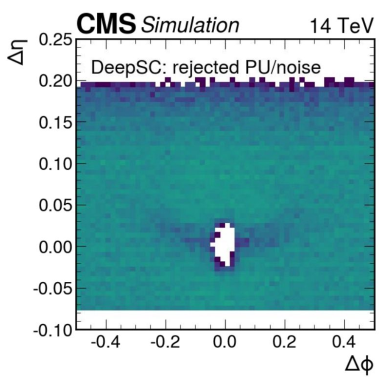


Studied an improvement with a **ML model**:

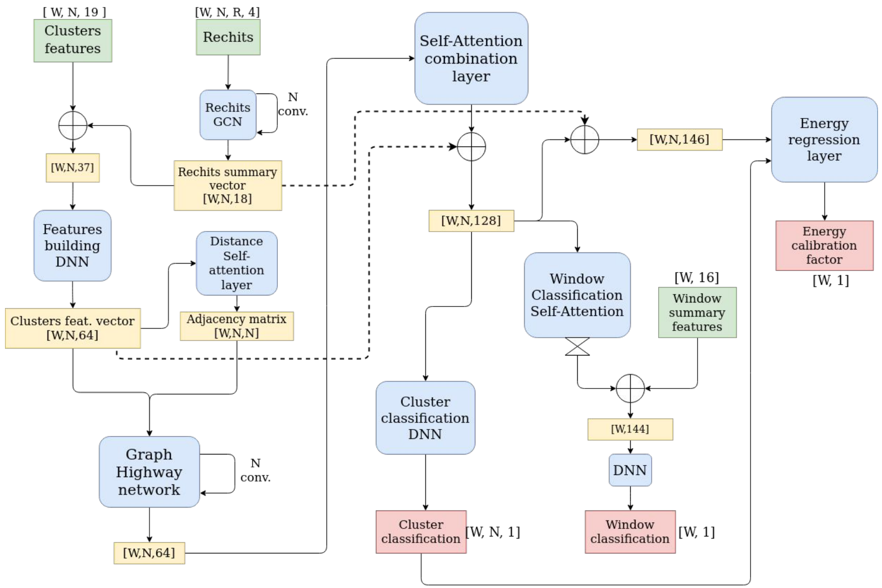
- Graph convolution network + attention layers
- ML model able to analyze the **full info in the detector window** and removes more efficiently pileup and noise
    - Can reach **5-10% resolution improvement** in detector regions with high material budget

[Davide Valsecchi for the CMS Collaboration 2023 *J. Phys.: Conf. Ser.* **2438** 012077]





rejected clusters

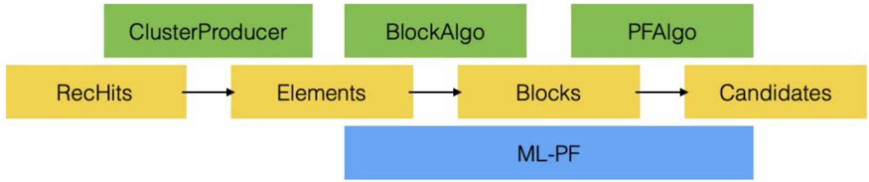


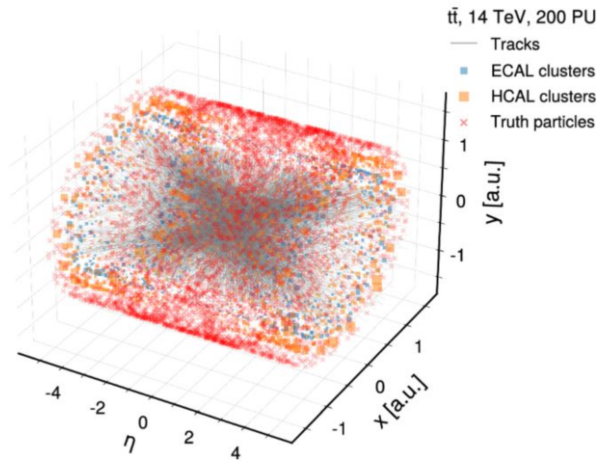
ML for Particle Flow

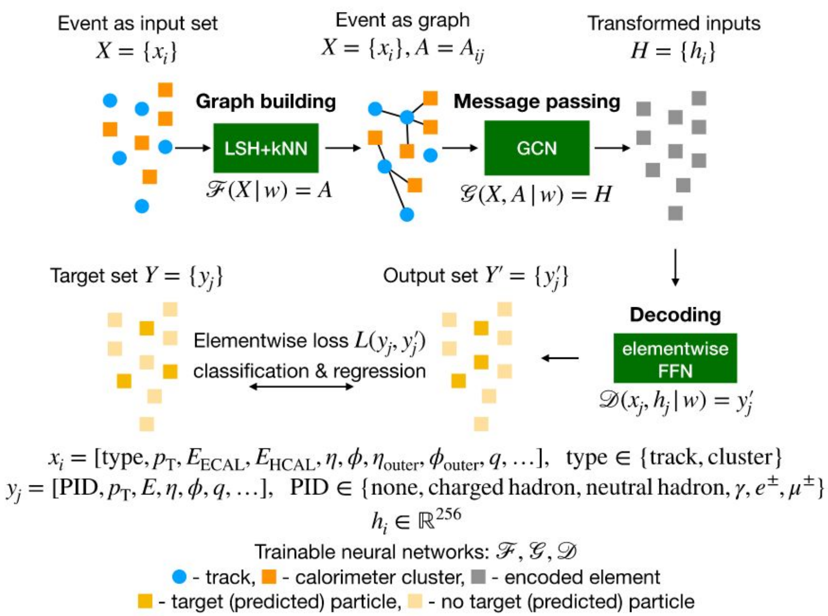
R&D effort in CMS, well advanced and implemented in CMS-SW through ONXX inference.

[Paper]

Sketch of PF algorithm step in CMS







- Starting from all sub-detector ingredients
  - Output directly the list of particle candidates
  - Particle ID and properties regression in one go
- **Dynamic graph building** done in an efficient way:
  - Locality sensitive hashing (LSH) arxiv
- Based on dense operations for portability

Normalizing Flows in CMS

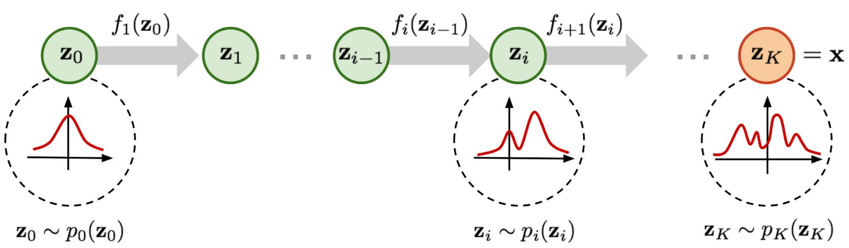
Normalizing Flows are a class of ML models used to learn complex, multimodal **probability density functions:**

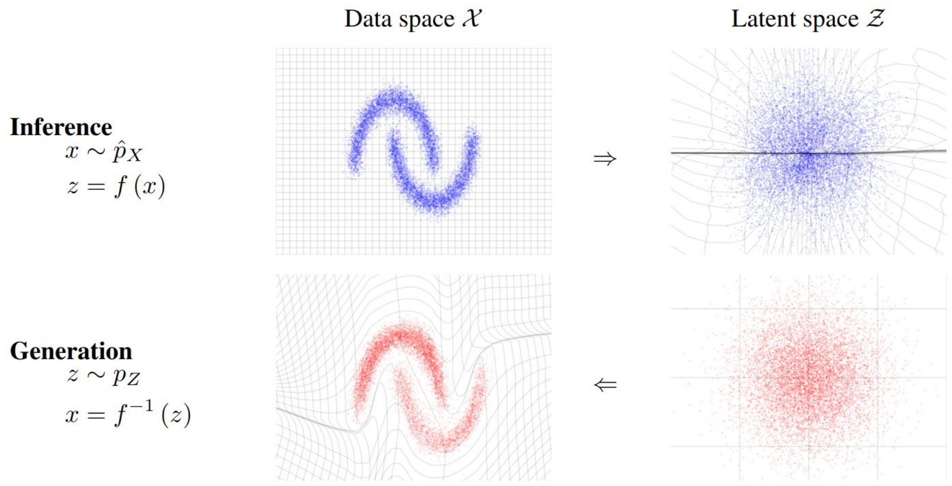
- fast probability density estimation
- fast sampling

In the CMS experiment Normalizing Flows (NFs) are being successfully applied for **MC correction and calibration, fast simulation, and analysis methods using importance sampling**

$$p_X(x) = p_Z(f(x)) \left| \det\left(\frac{\partial f(x)}{\partial x^T}\right) \right|$$

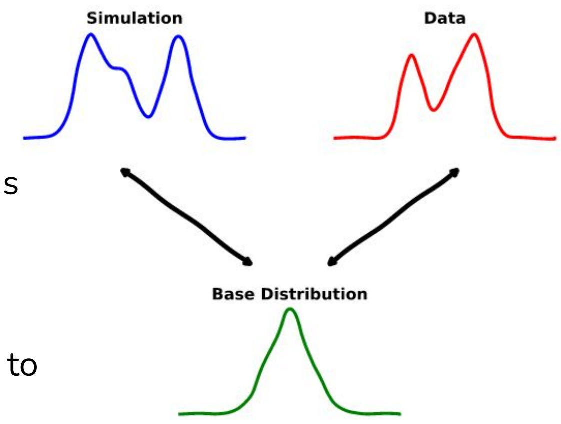
$$\log\left(p_X(x)\right) = \log\left(p_Z(f(x))\right) + \log\left(\left|\det\left(\frac{\partial f(x)}{\partial x^T}\right)\right|\right),$$





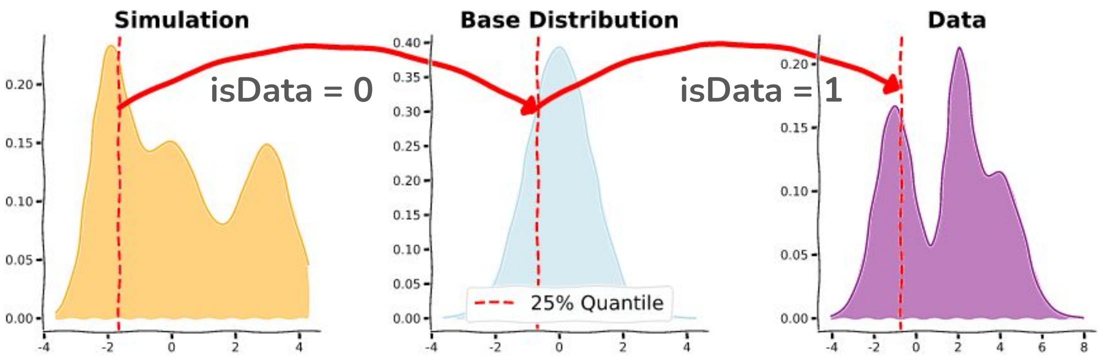
Normalizing Flows (NF) can morph a multivariate distribution in another one:

- Classical use case: input features for a regression/ID are **different between Data and Simulation.** Need to include corrections and related uncertainties, reducing the precision of the result.



The simulation can be calibrated **by morphing the input features** to be distributed as Data:

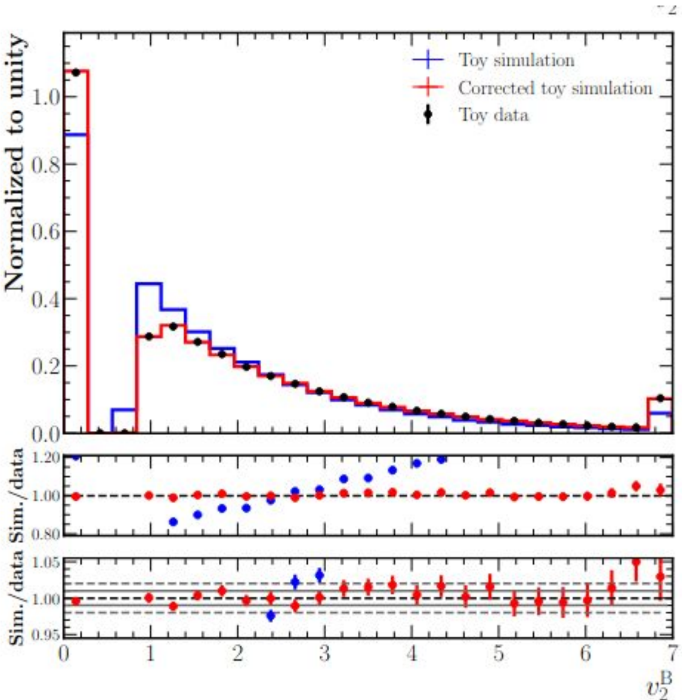
- Train 1 NF, conditioned on a **boolean switch**, on Data and Simulation simultaneously. Then use the reverse NF transformation to go from sim space to the Data space

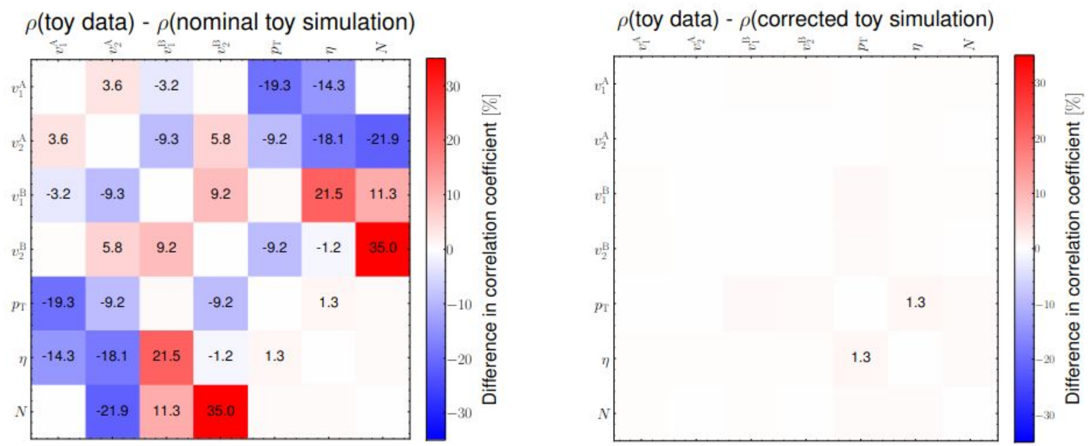


One flow to correct them all: improving simulations in high-energy physics with a single normalising flow and a switch, (C. C. Daumann, J. Erdmann, M. Donega', M. Galli, J.L.Spah, D. Valsecchi,) [2403.18582](2403.18582)

The NF approach works very well also with complicated differences in correlations between Sim and Data and in many dimensions (also conditionally on ancillary values).

→ Successful study on toy data now going to be applied on CMS photon ID inputs.





Effect of morphing of correlation matrix

One flow to correct them all: improving simulations in high-energy physics with a single normalising flow and a switch, (C. C. Daumann, J. Erdmann, M. Donega', M. Galli, J.L.Spah, D. Valsecchi,) [2403.18582](2403.18582)

Normalizing Flows and transformers can be combined to perform faster importance sampling and compute the Matrix Element Method → new application in [CMS-DP-2023/085]

single CMS event

Reconstructed jets and leptons

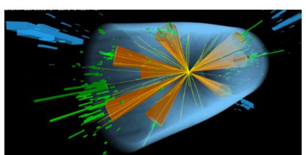
$N \in [0,1]^{10}$

Conditioning Transformer

Sampling Flow

conditioning

Transfer Flow

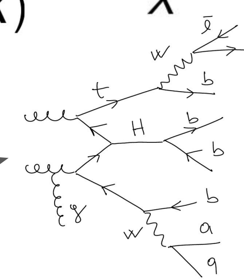


prob. density of each the parton set

N sets of partons X

$\mathcal{W}(\vec{Y}|\vec{X})$

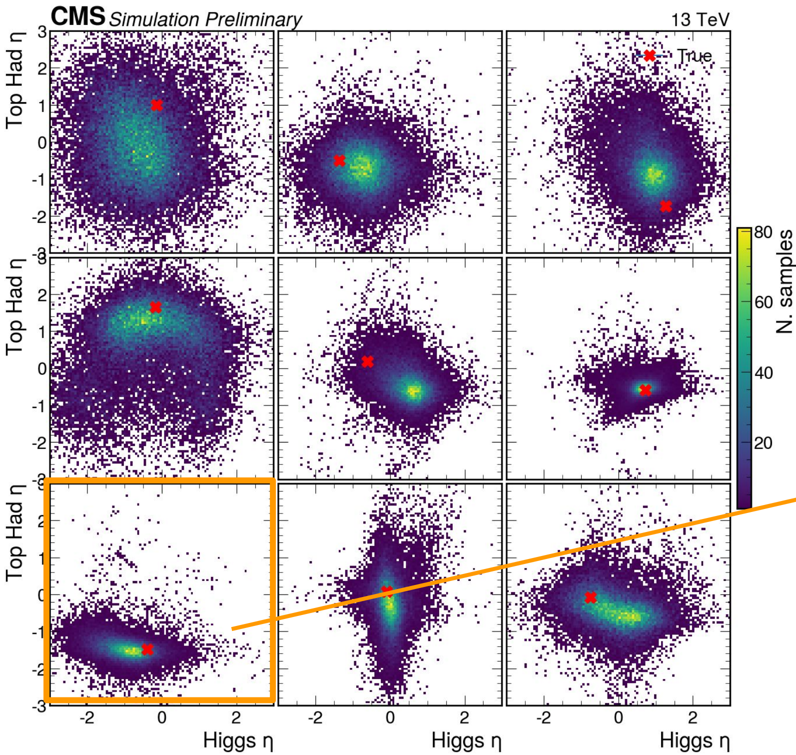
$\vec{X}$

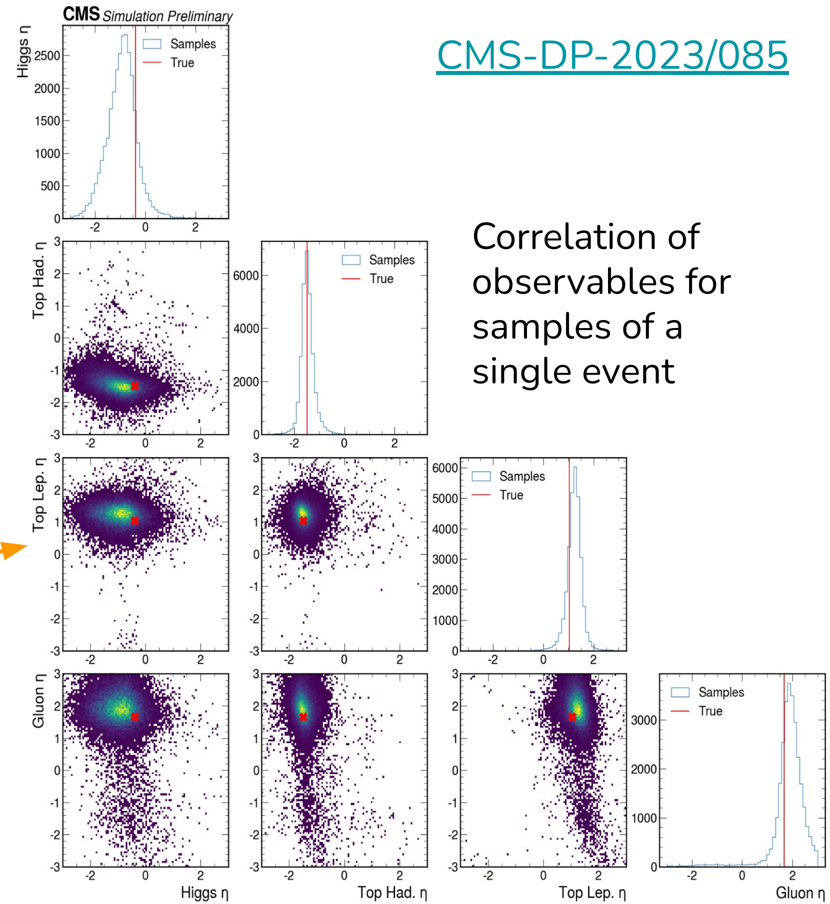


Matrix Element Method computation

$$\mathcal{P}(\vec{Y}|\vec{\theta}) = \int_{\phi} d\vec{X} \cdot |\mathcal{M}(\vec{X}|\vec{\theta})|^2 \cdot Pdf \cdot \mathcal{W}(\vec{Y}|\vec{X})$$

Sampling flow





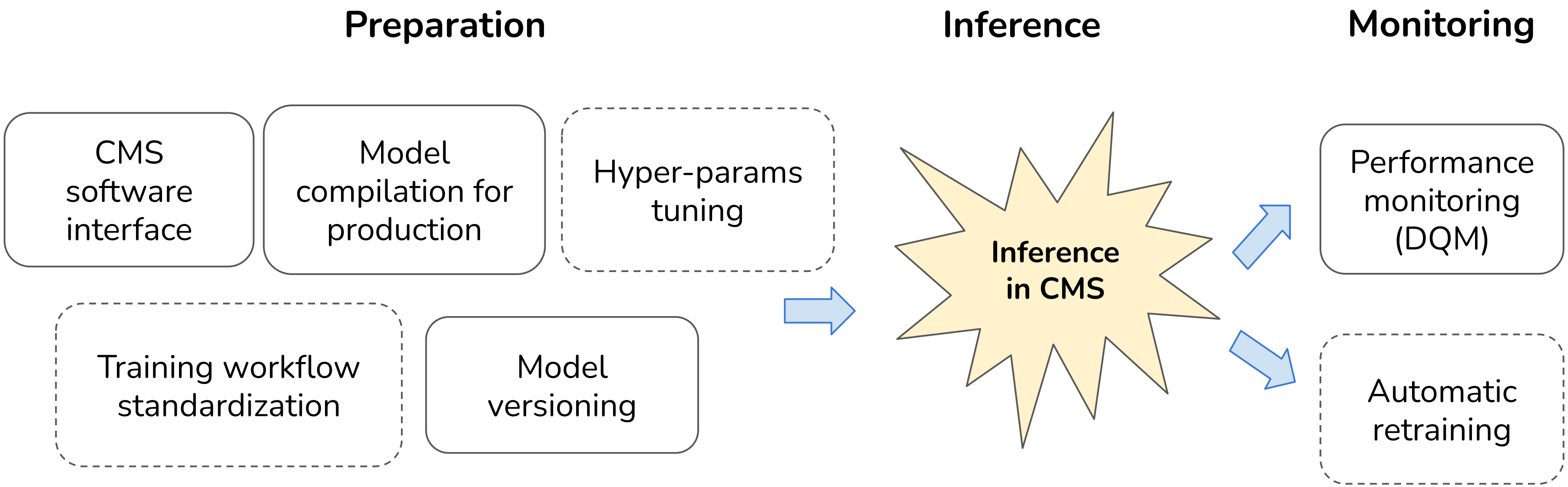
CMS-DP-2023/085

Correlation of observables for samples of a single event

The sampling flow learns the complete conditional probability ***P(partons|reconstructed event)*** and generates partons in the most probable configurations for the MEM integral computation

In the CMS ML group we want to **streamline the integration** of new ML models in the CMS production.

- Defining best practices
- Preparing tools for **profiling, compilation, packaging, versioning, monitoring**
- Documenting all the necessary steps: CMS-ML-docs



Training resources and hyper-param optimization

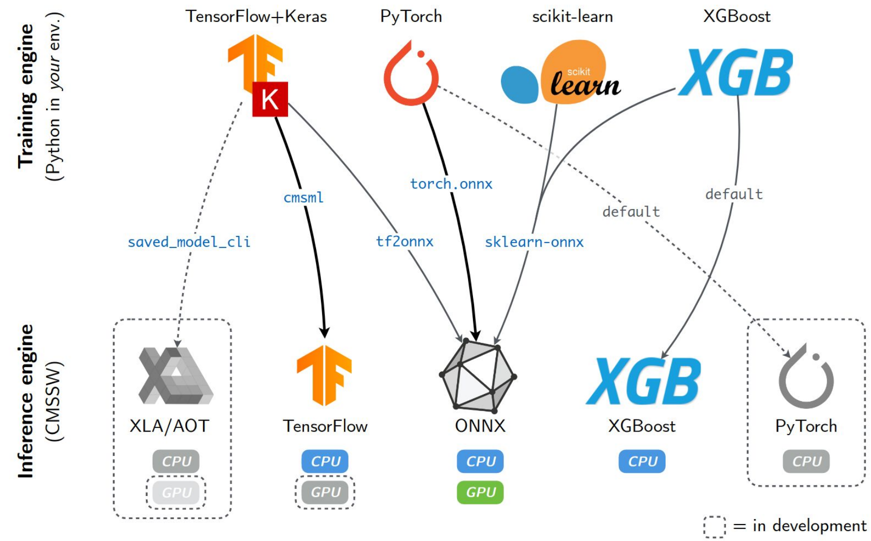
ML development in CMS is carried out **independently by many groups**:

- no central training infrastructure yet in place:
    - Work ongoing on common training frameworks and tools!
- Analysis, reconstruction, trigger, DQM, anomaly detection, simulation → **many different requirements and use cases**
- **GPUs are always used for training:** up to 10 GPUs used for prototyping complex end-to-end reconstruction models
    - Groups relying on university clusters, CMS Tier 2 / 3 resources, CERN resources or seldom HPCs
    - Dedicated **ML training facilities** are emerging as a dedicated solution
- Hyper-parameters optimization is rarely performed due to the lack of time or large training infrastructures and tools

In general the availability of more GPUs resources enables **faster time-to-science** and more hyper-parameters optimization. **Fast storage** well connected to the GPU hardware is also crucial (~TBs training datasets)

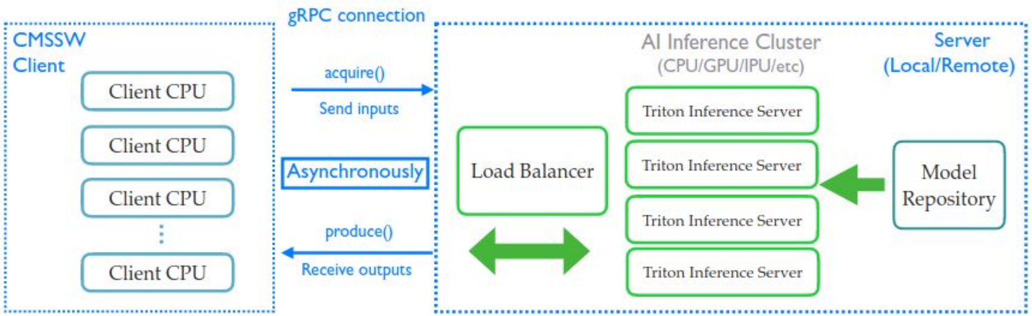
ML models inference in CMS production workflow relies on ML frameworks integrated in the CMS software stack (monolitic):

- TensorFlow, ONNX, XGB, PyTorch available
- Run in single-thread CPU configuration, due to the CMS software nature
- GPU support available but not used for inference in production yet



Exploring an alternative model based on **indirect inference** using Nvidia Triton servers:

- delegate ML models execution to external servers, also with **GPUs**
- Reduce **dependencies complexity** in CMS software
- promising performance study done in [2402.15366]



Machine Learning developments are flourishing in many aspect of the CMS experiment

→ High potential of improving even more the experiment Physics results output!

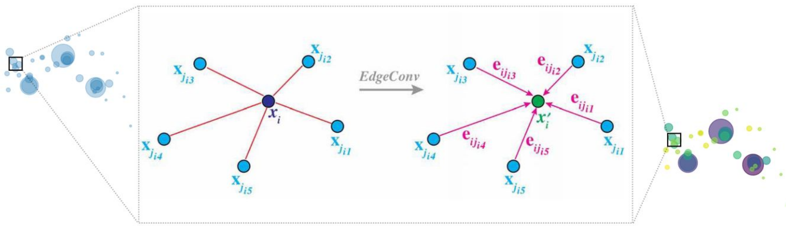
CMS is applying state-of-the-art models, such as transformers and generative AI, to HEP problems with success.

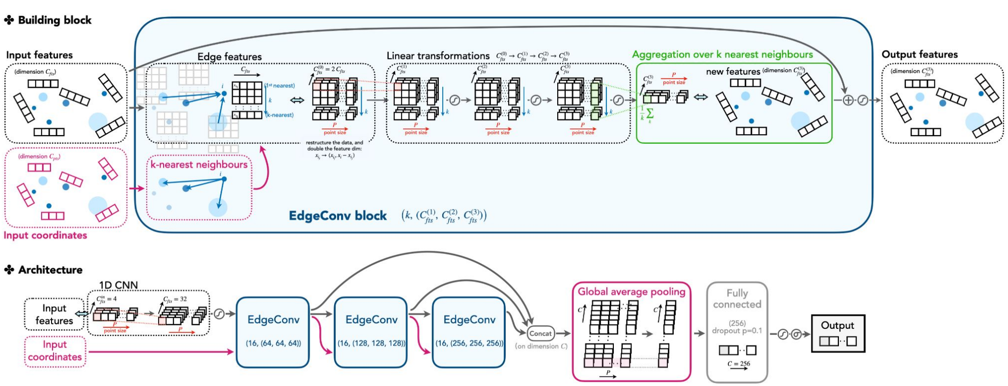
MLOps in CMS are challenging:

- Training hardware needs becoming heavier and hyper-parameter optimization still rare
- Common tools and frameworks are under development
- Maintenance and optimization of models used in production is under way
- R&D ongoing about future models for ML inference at scale

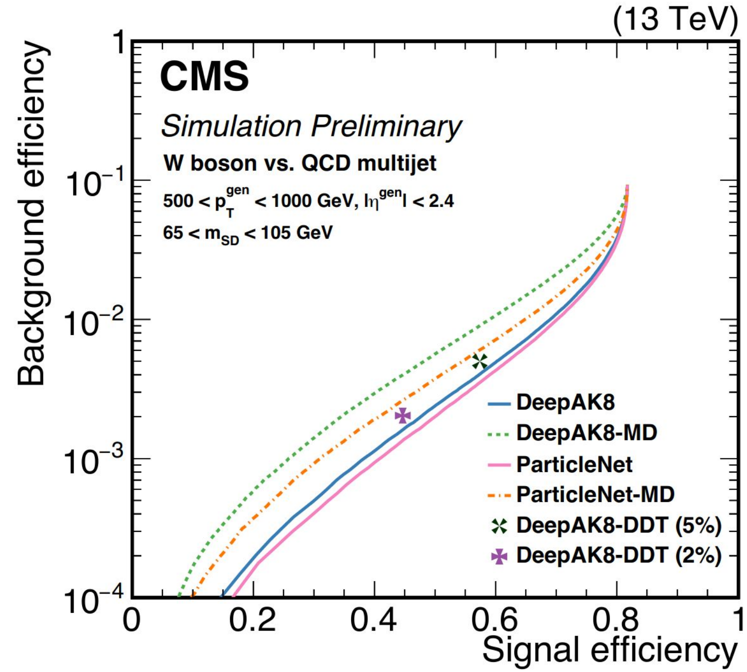
Backup

- EdgeConv GNN based architecture on jet constituents.
- **Success story** of full integration in CMS: similar architecture used for many different tasks. AK4, AK8 tagging, mass regressions..
- Inference in CMSSW from ONXX runtime.
- Full documentation and training framework (Weaver) available



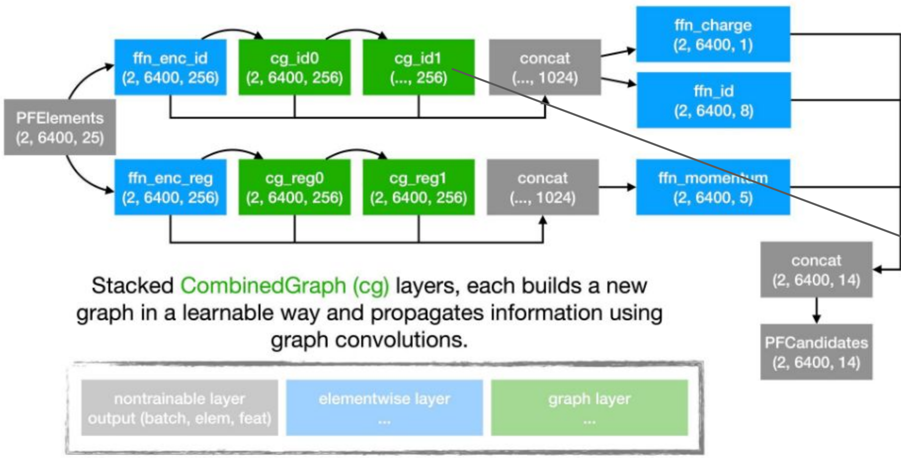


DP note



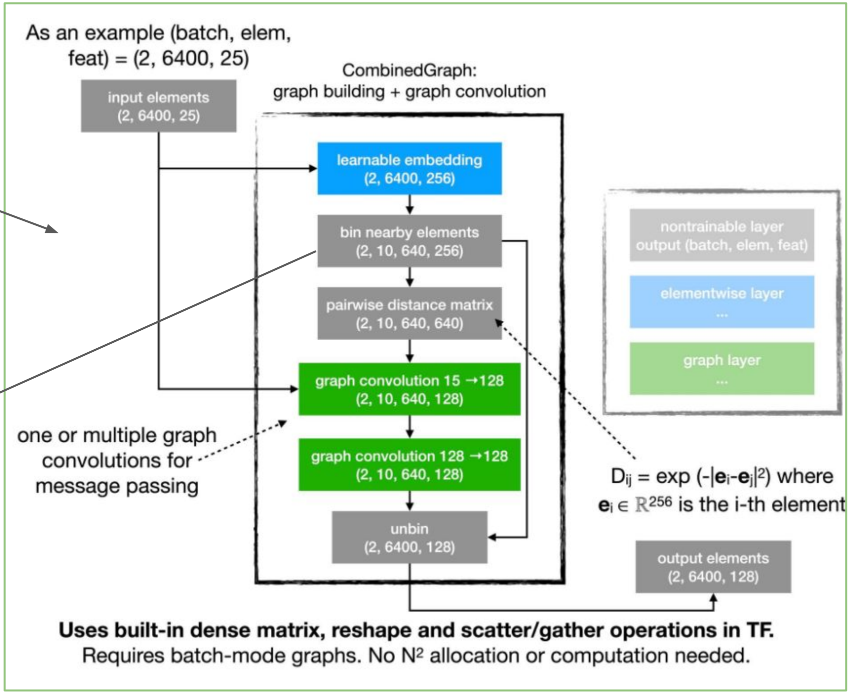
Particle ID an properties are stacked together in the decoder

As an example (batch, elem, feat) = (2, 6400, 25)

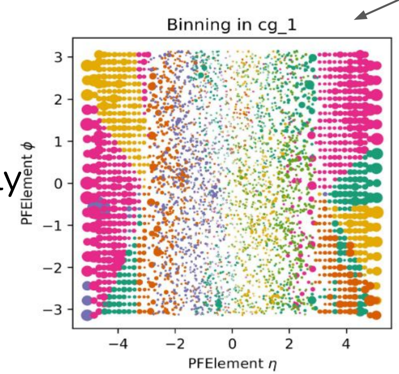


CombinedGraph layer

- Learnable embedding to form sub-graph
- Multiple graph-conv to propagate info.



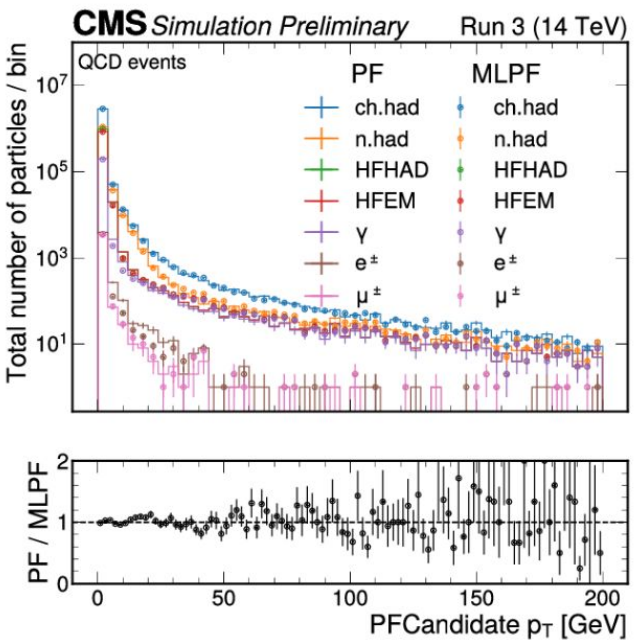
- subgraph binning is fully learnable
- no ground-truth



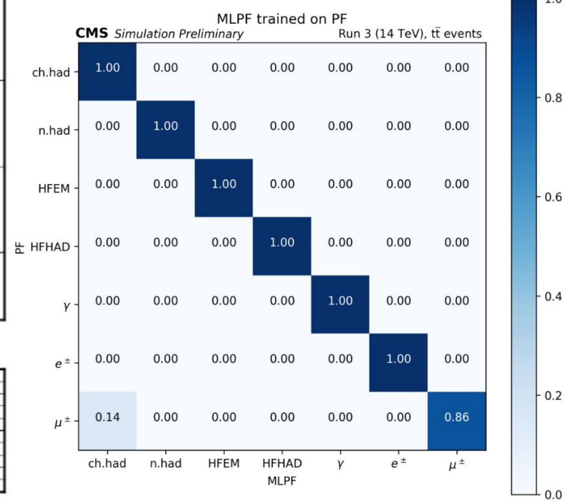
[Slides at CMS ML forum]

MLPF performance

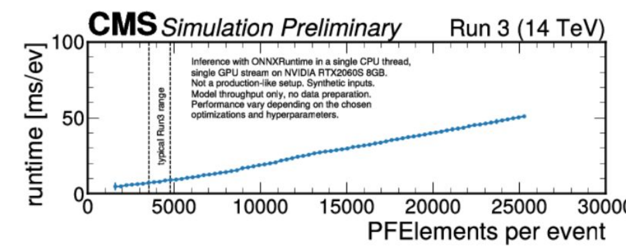
- Hyperparameters optimization is going, but the performance on a realistic environment is very promising.
- Until now trained on PF candidated → work ongoing to define the best possible GEN-level truth

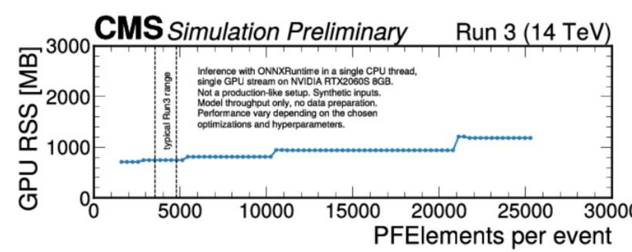


PID confusion matrix



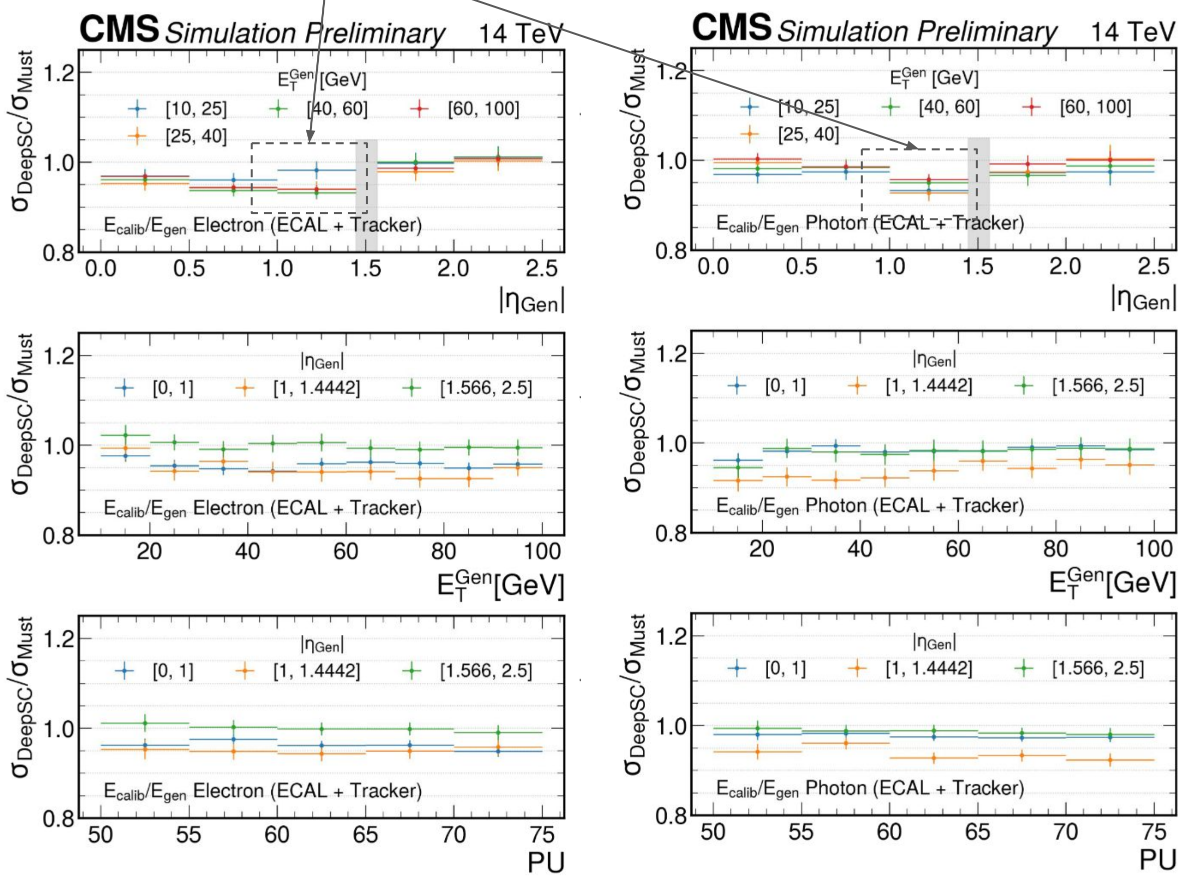
Inference performance is under control





Pata, J. et al. Machine Learning for Particle Flow Reconstruction at CMS. ACAT 2021. https://doi.org/10.48550/arXiv.2203.00330

Improvements in the final resolution (after regression) where the material budget is larger → DeepSC cleans the object, especially at low energy



[CMS DP note](#)

arxiv2001.05486

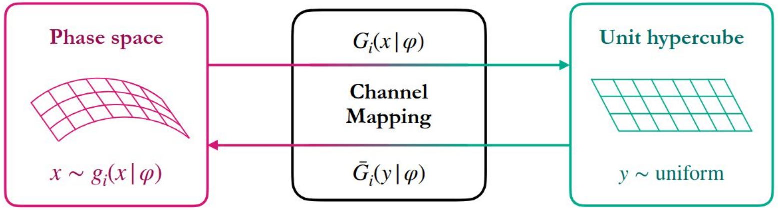
arxiv2011.13445

Flows for integration by importance sampling are gaining a lot of momentum in the theory community:
- general algorithm described as *i-flow* arxiv2001.05486

Large interest to **optimize the phase-sampling for cross-section** calculations

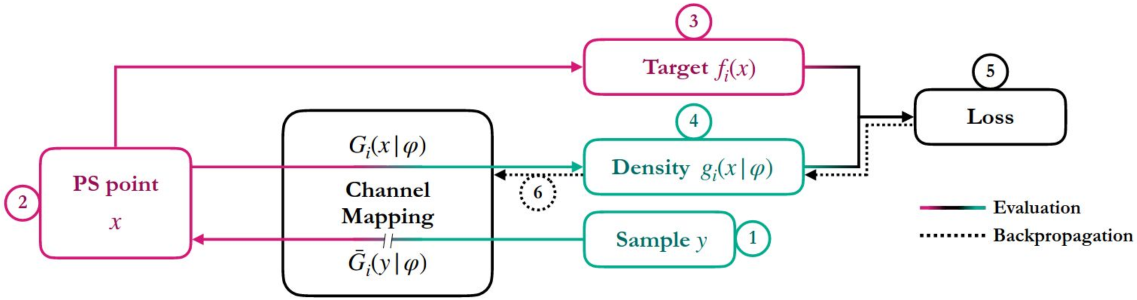
Very recent nice paper about multi-channel integration via normalizing flows to be integrated with MadGraph:

- MadNIS – Neural Multi-Channel Importance Sampling **arxiv2212.06172**



The integrand function is approximated by normalizing flows, (for different integration channels like in Madgraph)

The flow is optimized by sampling and evaluating the target density (hard-scattering probability)

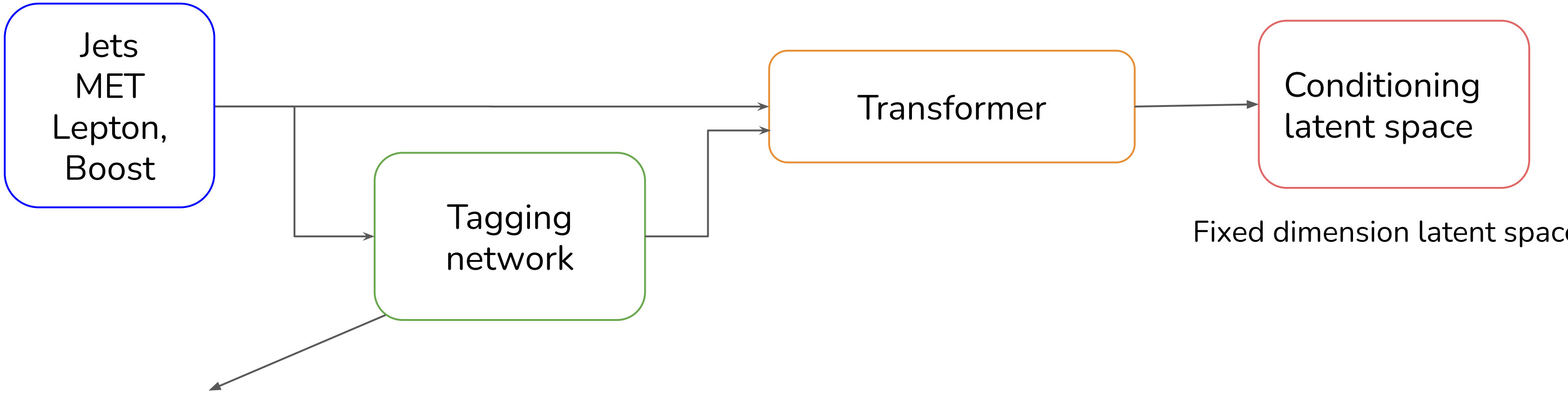


Conditioning on reconstructed events

- Sampled particle sets for the MEM integral computation **strongly depends** on the **reconstructed objects.**

- Use a **transformer** to extract a fixed-size conditioning latent space for the unfolding flow
  → can handle additional radiation and missing objects
  → avoids direct jet-parton combination

- The conditioning latent space should be correlated with the most probable partons



Using a pretrained **SPANET network** and adding to each jet the probability to be generated from H, top hadronic, top leptonic

Conditioner pre-training

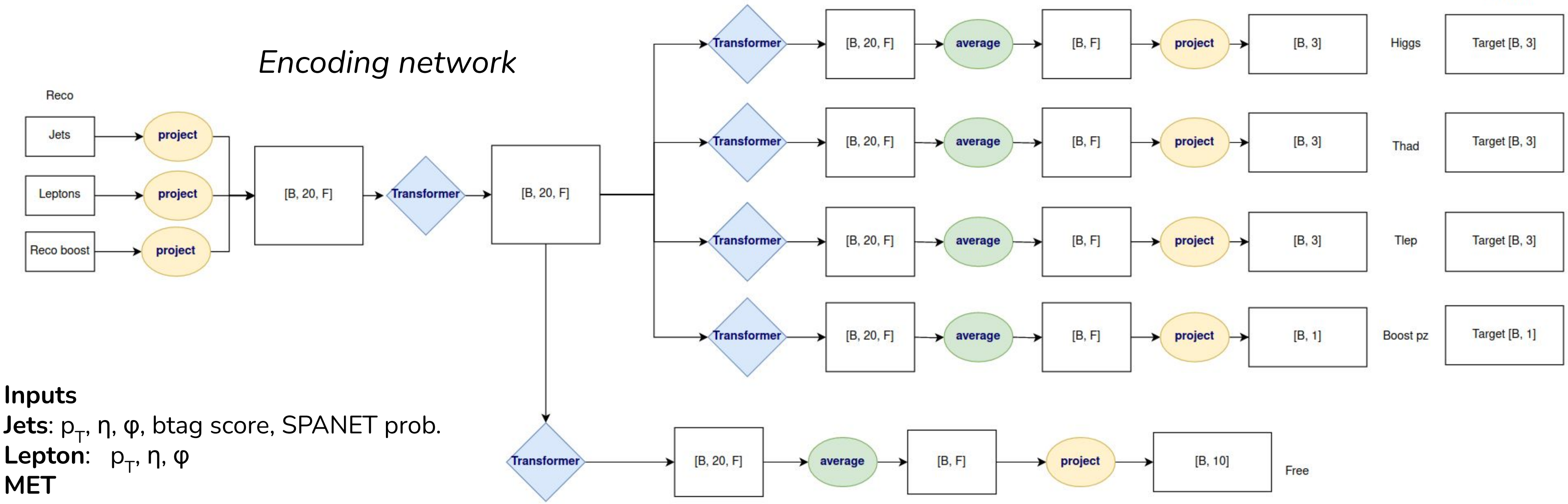
- **Idea**: pretrain the conditioning transformer with a **regression** of the **generator-level particles**:
  higgs, top_{had}, top_{lep} (p_T, η, φ) + total event boost p_Z

  → additional radiation (gluon) computed from momentum balance

  → maximize the correlation with the target of the unfolding flow

Decoding network

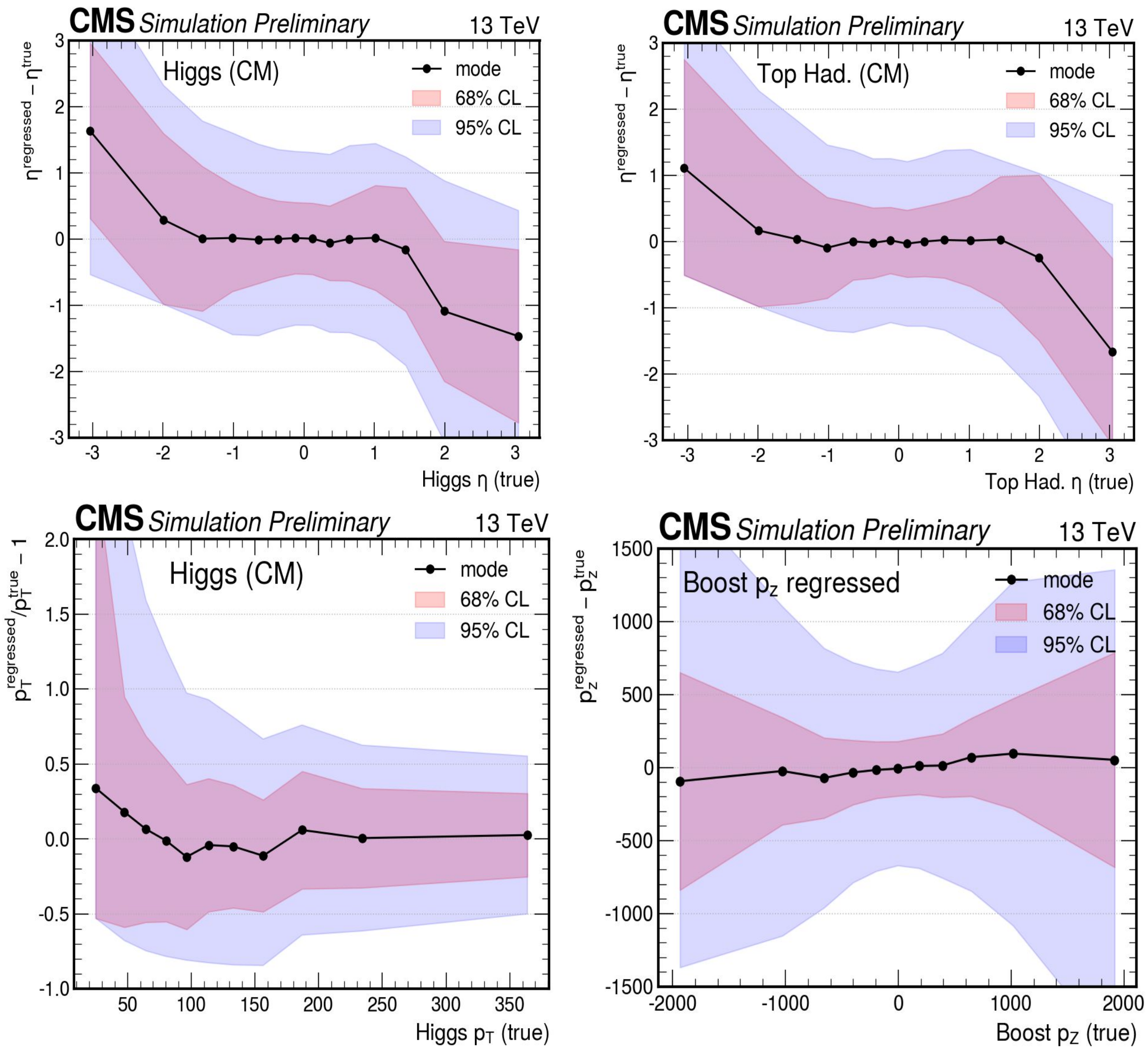
Encoding network



Inputs
Jets: p_T, η, φ, btag score, SPANET prob.
Lepton: p_T, η, φ
MET
Boost: 4-vector sum of all objects $|\eta| < 2.4$

free latent space, not constrained in the pretraining

Parton regression performance



The regression of the generator-level particles is overall unbiased

Also the total p_Z of the event is well regressed
→ the particles can be boosted in the centre-of-mass (CM) correctly.