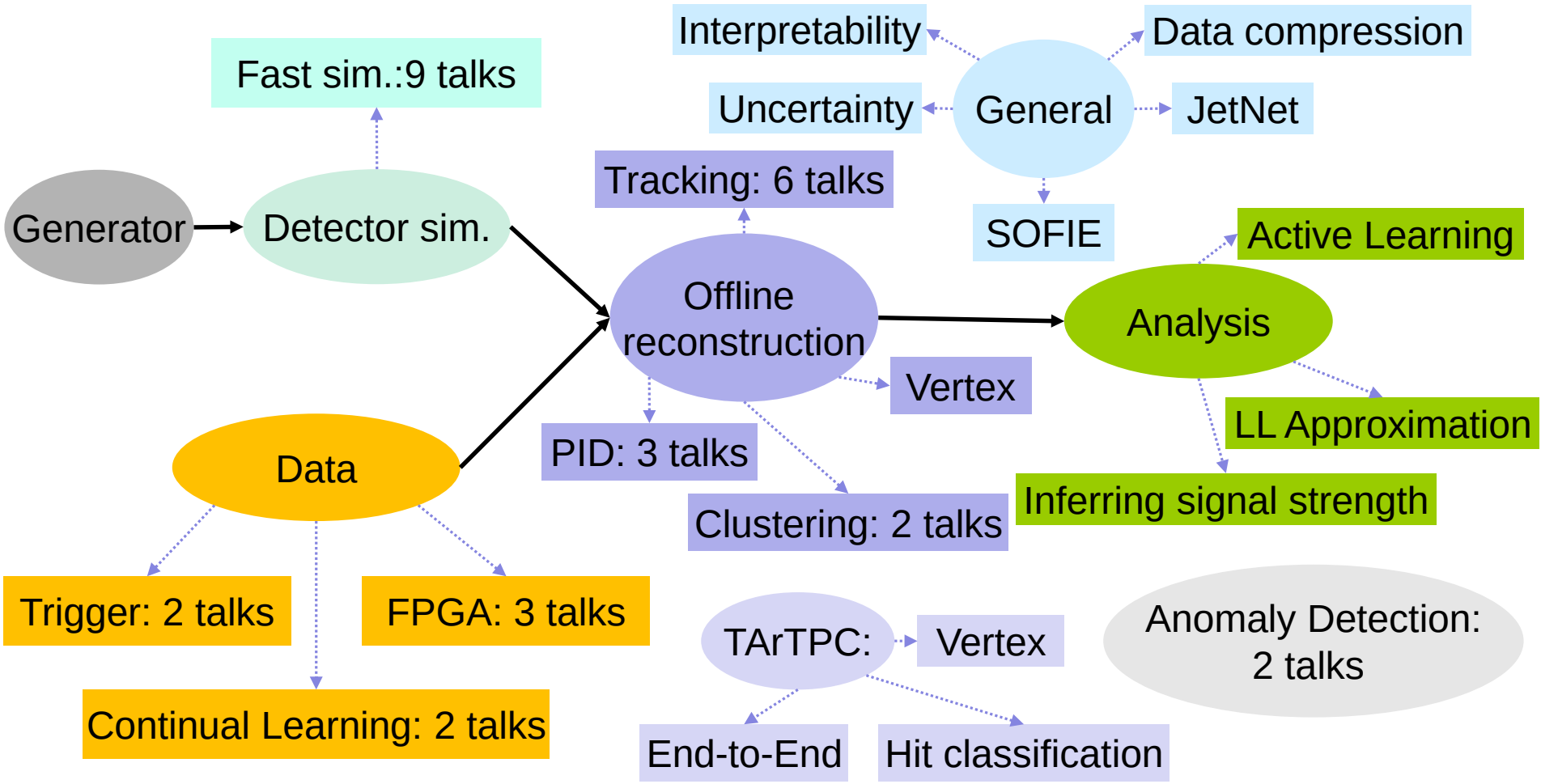
中国科学院高能物理研究所
Institute of High Energy Physics
Chinese Academy of Sciences

News from CHEP2023 for AI in HEP

Wenxing Fang (IHEP)

Ihep-ml-innovation] weekly meeting (2023.06.01)

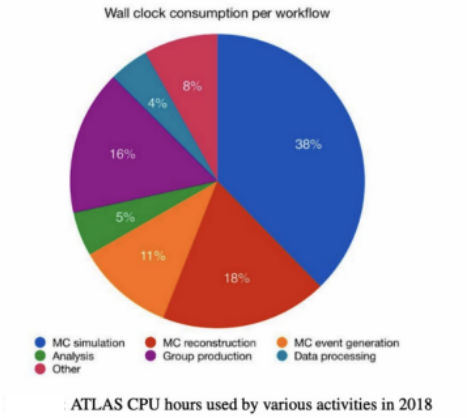
Overview of the ML talks



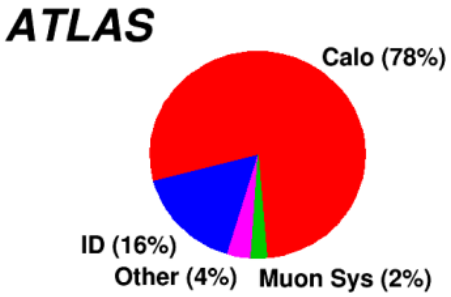
Fast simulation

Fast simulation

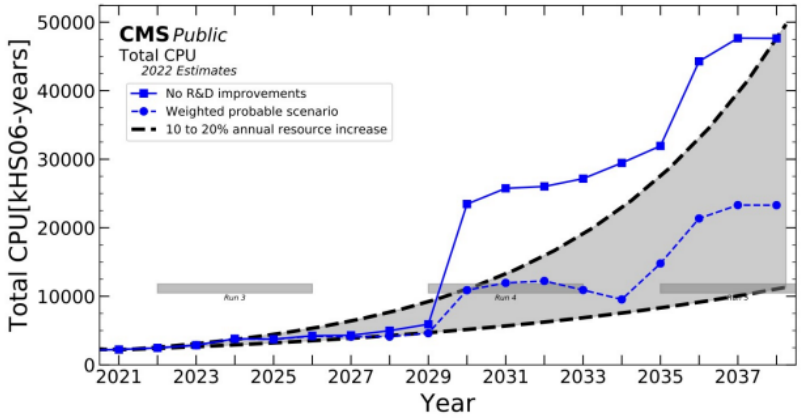
The Need for Fast Simulation



ATLAS CPU hours used by various activities in 2018



Subdetector CPU fraction for 50 ttbar events
MC16 Candidate Release

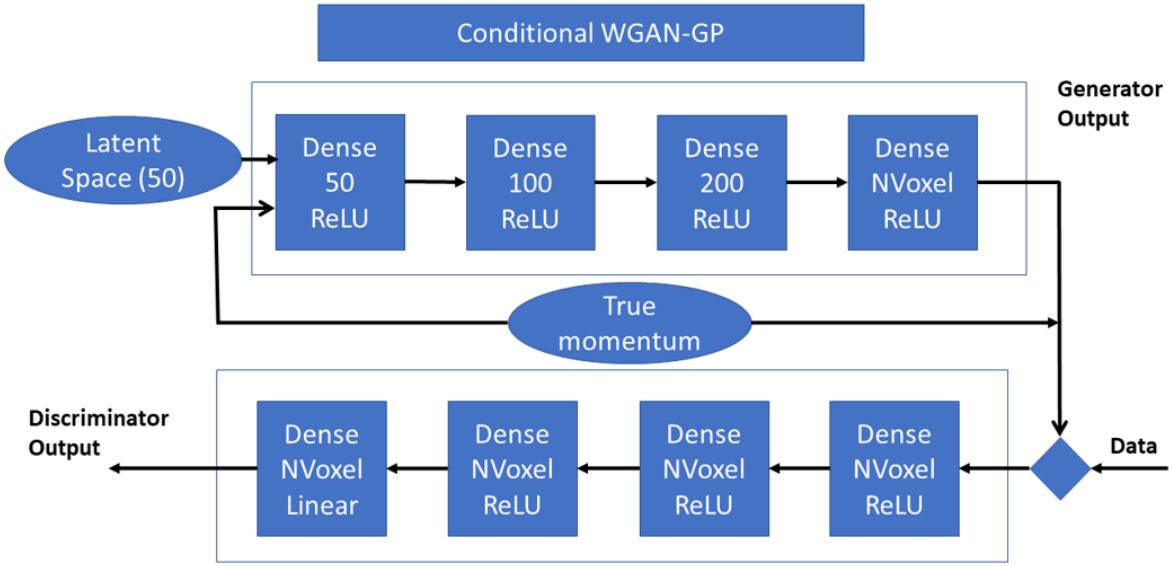


- Geant4 calo simulation is a significant part of ATLAS computing budget
  - CMS will face similar needs with HGCAL in HL-LHC
- For HL-LHC, computing simulation more crunched
  - Reconstruction usage will scale ~linearly with pileup → less resources for sim.

Oz Amram
In collaboration with Kevin Pedro

Fast calorimeter simulation (GAN)

The GAN

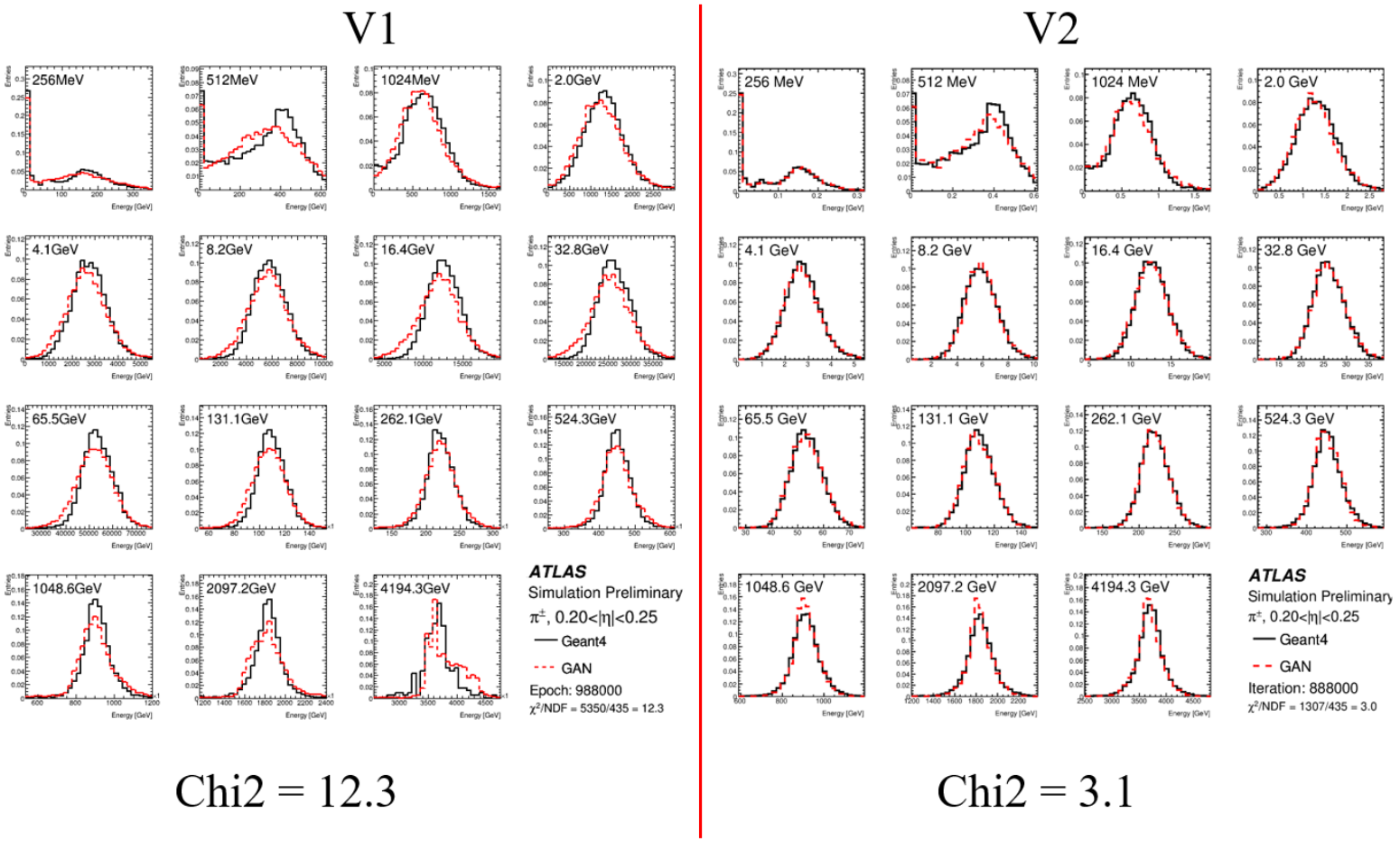


G	50 (Input latent Space), 50, 100, 200, NVoxel (pid and η dependent)
D	NVoxel, NVoxel, NVoxel, NVoxel, 1
Activation function	ReLU (in all layers)
Optimiser	Adam [21]
Learning Rate	10^{-4}
β	0.5
Batchsize	128
Training ratio (D/G)	5
Gradient penalty λ	10

NVoxel depends on PID and η

Fast calorimeter simulation (GAN)

Best GAN for pions: v1 vs v2

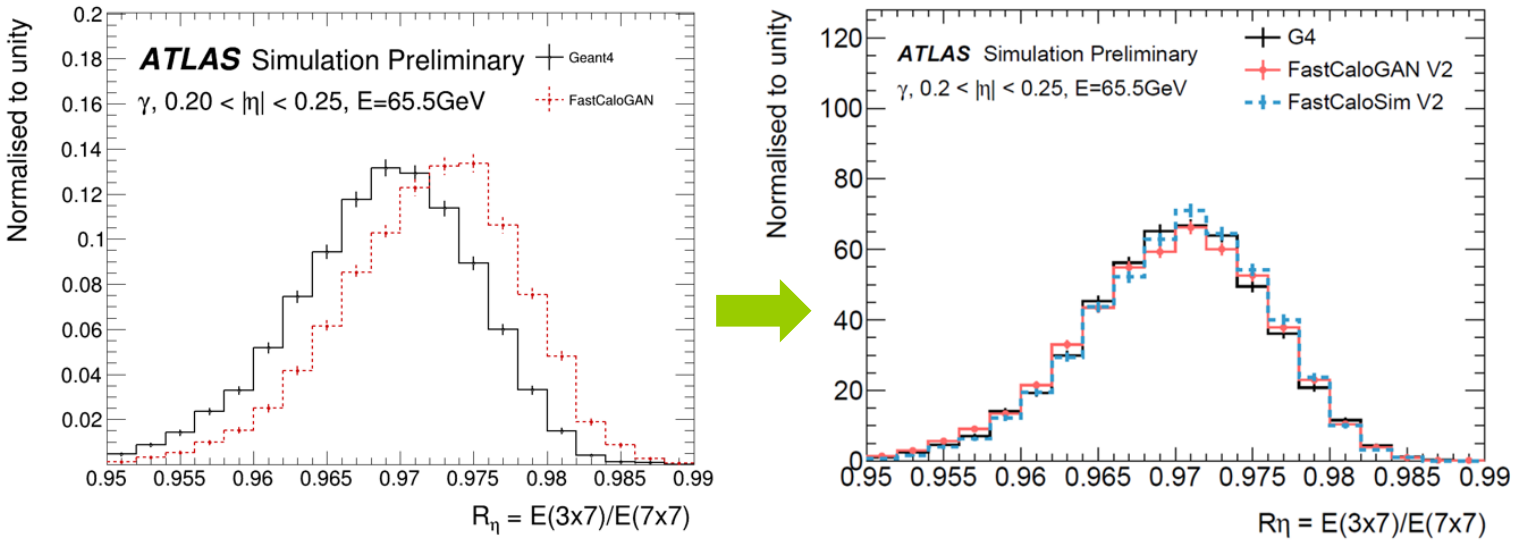


Chi2 = 12.3

Chi2 = 3.1

Fast calorimeter simulation (GAN)

Photon shape



- FastCaloGAN was born as a prototype and even at that early stage was able to have great performance for pions, so much as to be used in AF3
- FastCaloGAN V2 is a mature tool that is now very accurate for all particles and will be extensively used in Run3 fast simulation

Fast calorimeter simulation (Diffusion)

Diffusion Models

- Diffusion has become the dominant paradigm for ML image generation
  - Dalle-2, Midjourney, Stable Diffusion, etc
- Easy training, high quality results, reasonable computation times

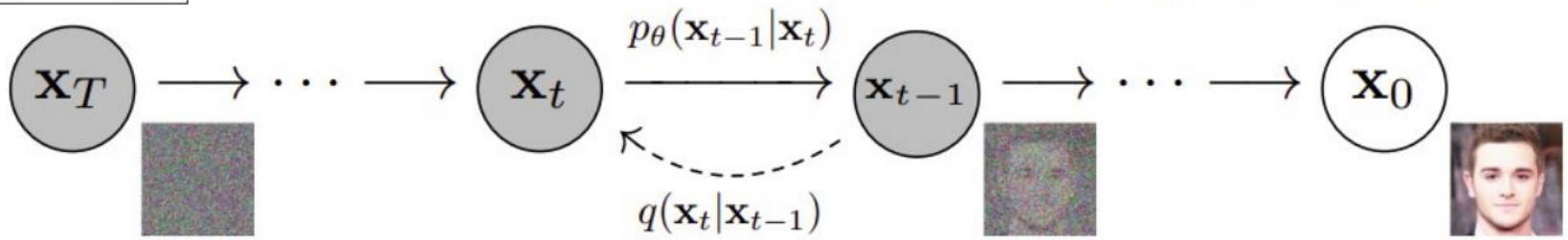
"AI aiding physicists at LHC to analyze data and discover new particles"

Fast calorimeter simulation (Diffusion)

Diffusion Models : Technical Details

2006.11239

NSteps typically ~few hundred



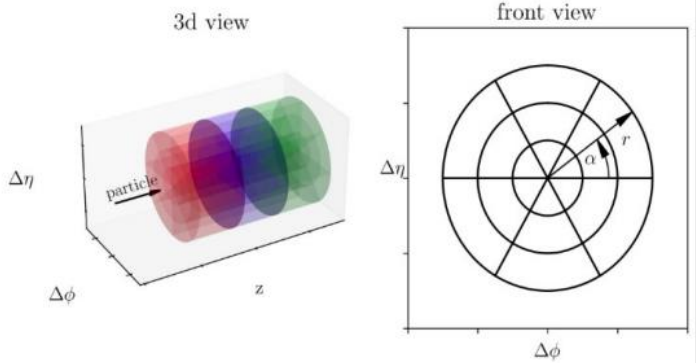
- Diffusion process: Starting with some image, **iteratively add Gaussian noise**, eventually reaching pure noise
- Train a model to **invert the diffusion process**
- Generate by starting from noise image, **iteratively denoise** using trained model
- Can condition on additional input information
  - Eg. text prompt or incident particle energy

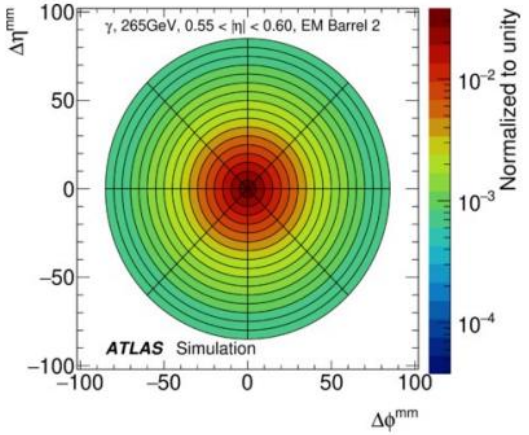
5

Fast calorimeter simulation (Diffusion)

Dataset: Calo Challenge

- Community challenge to compare generative models for Calorimeter simulation
- Standard datasets to allow comparison
  - Dataset1: ATLAS-like geometry, 5 layer cylinder with **irregular binning**, 368 voxels
  - Dataset2: 45 layers, 6480 total voxels
  - Dataset3: 45 layers, 40,500 total voxels

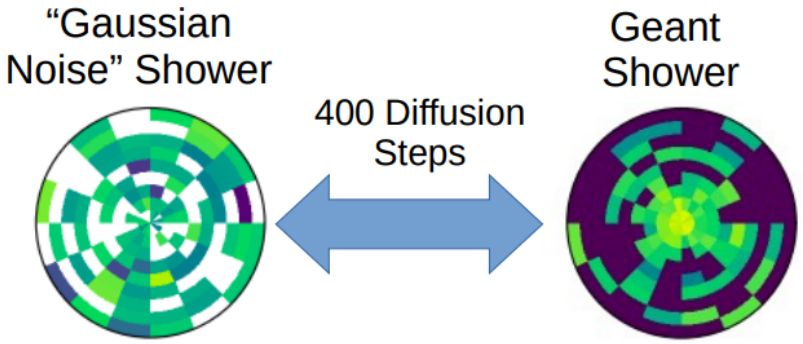


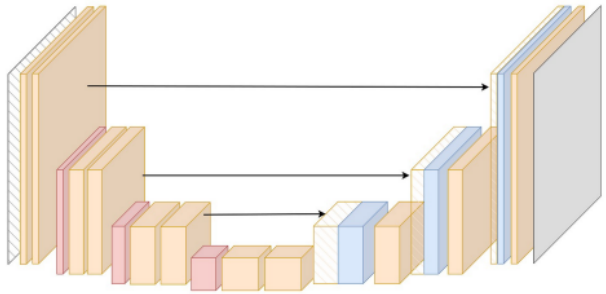


Fast calorimeter simulation (Diffusion)

‘CaloDiffusion’

- We train diffusion models to generate synthetic calorimeter showers based on Geant simulations
- We use **400 steps** to interpolate from real shower to Gaussian noise
- Denoising network is has ‘U-net’ architecture based on 3D convolutions
  - Primary input: Noisy shower
  - Conditioning inputs: incident particle energy & diffusion step
- Training objective normalized noise component of the shower
  - Denoising → subtracting noise off
- Several novel optimizations utilized

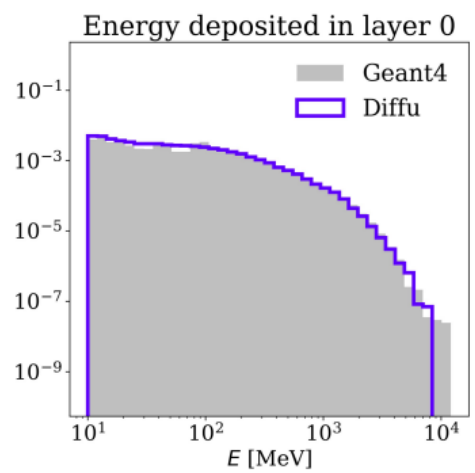


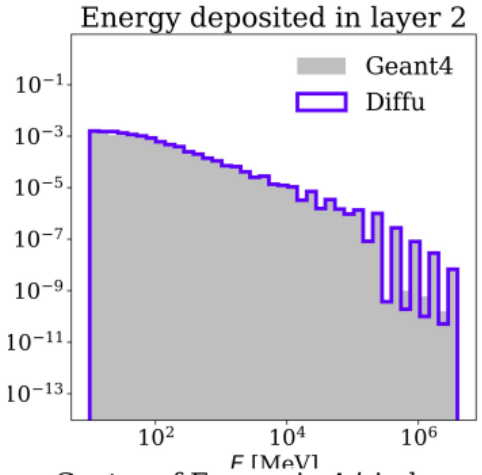


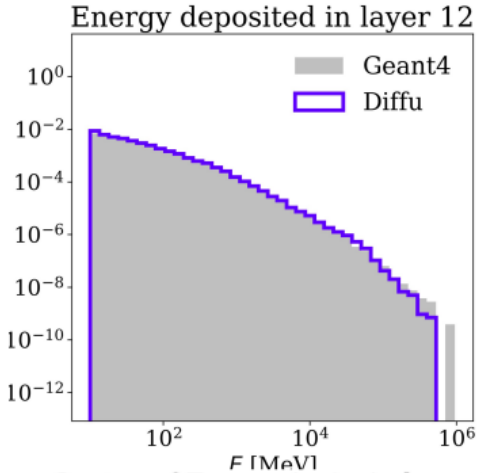
U-nets compress to a smaller dim space but also include skip connections

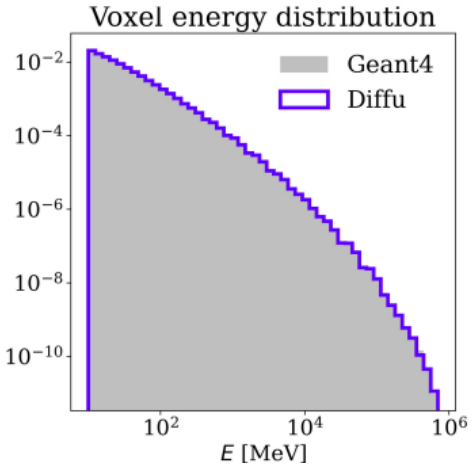
Fast calorimeter simulation (Diffusion)

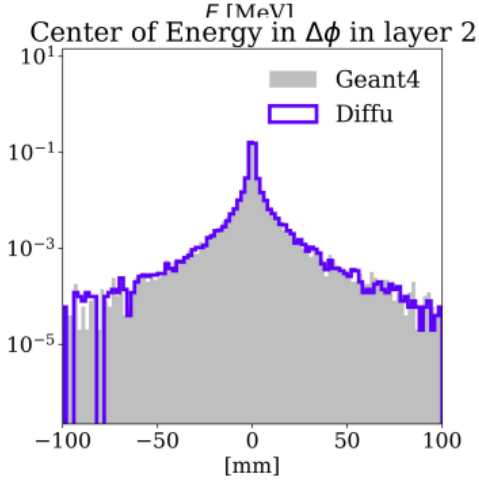
Dataset 1 Results

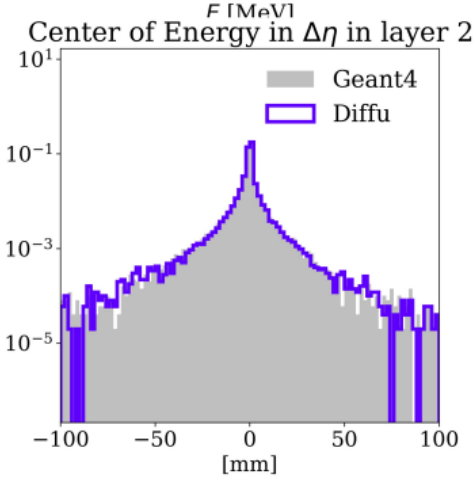








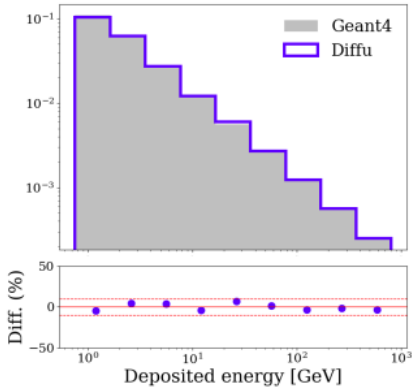


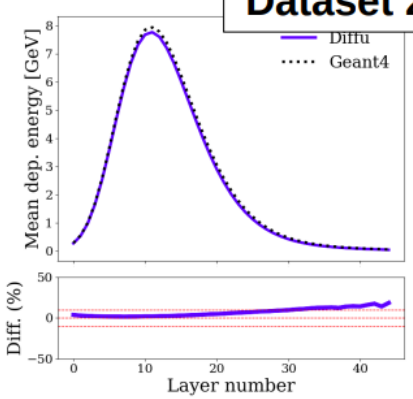


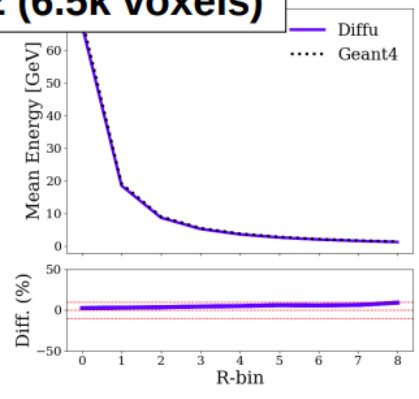
Fast calorimeter simulation (Diffusion)

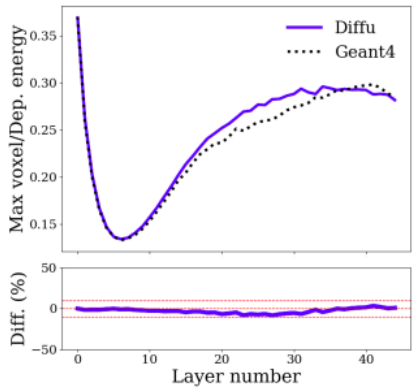
Results: Datasets 2 & 3

Dataset 2 (6.5k voxels)

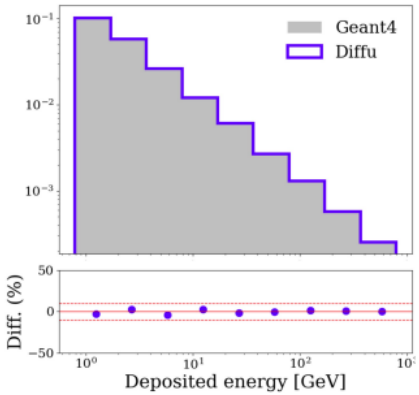


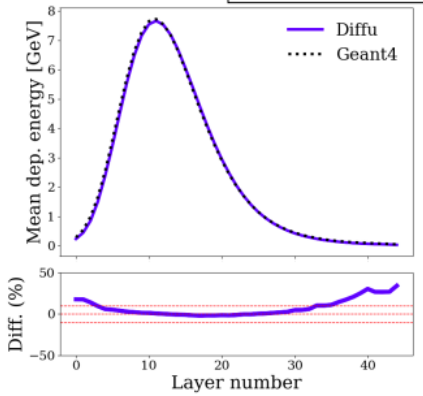


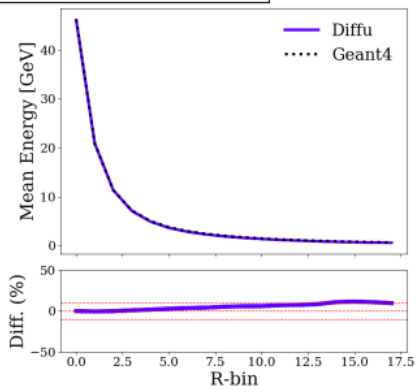


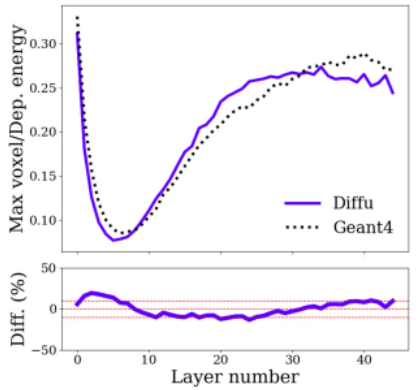


Dataset 3 (40k voxels)









Fast calorimeter simulation (Diffusion)

Quantifying Performance

- Train a NN classifier to distinguish between Geant showers and CaloDiffusion showers
- Quantify sample quality based on AUC on holdout set)

	Dataset 1 (ATLAS-like)	Dataset 2	Dataset 3
Classifier AUC *	**~0.65**	**~0.6**	**~0.7**

AUC much less than 1 → Very similar showers!

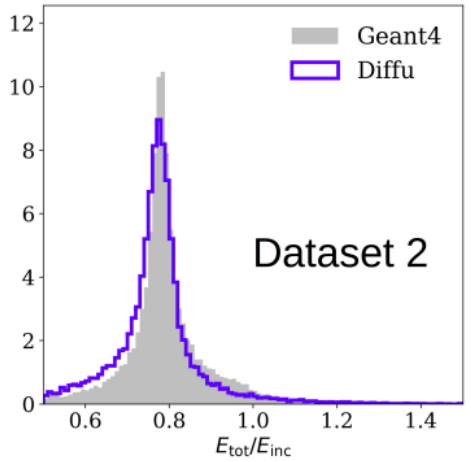
Additional metrics in backup

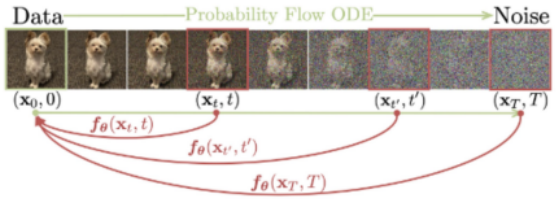
*Preliminary numbers, somewhat dependent on exact classifier training setup

Fast calorimeter simulation (Diffusion)

Future Work

- Some “global” properties (ie total shower energy), can still be improved
  - Hard to specifically optimize in diffusion training
  - Will try batch-level MMD loss
- Generation time is slower than other ML approaches b/c of iterated generation (still faster than Geant)
  - Can be improved with different sampling algos, compression, or **distillation methods**
  - Or start generation from **approximate shower** instead of pure noise (“Cold Diffusion”, 2208.09392)
- Extend to more complicated geometries e.g. **CMS HGCal**





“Consistency Models” distill diffusion model to allow ~few step generation

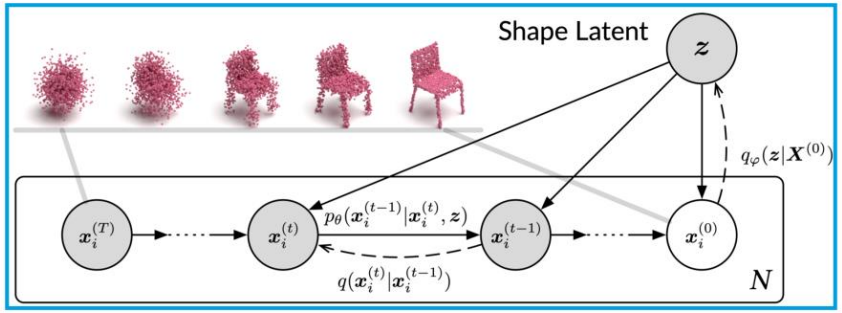
Fast calorimeter simulation (Diffusion)

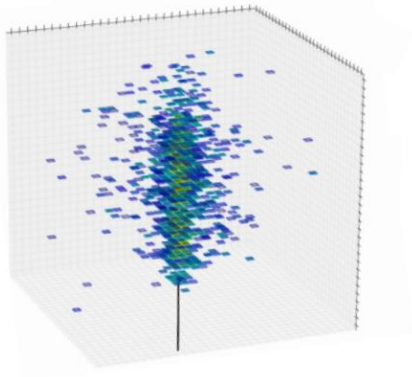
CaloCloud diffusion model

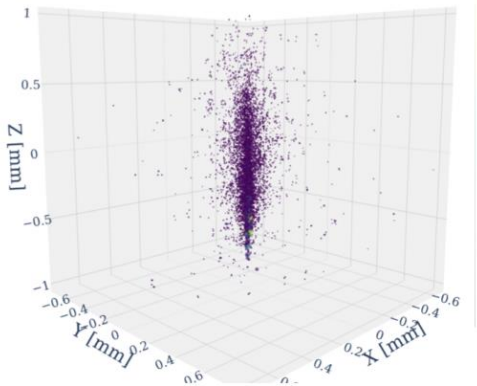
from regular grids to point clouds

- regular grid models like WGAN or BIB-AE show very high physics fidelity - yet they have two problems:
  - low occupancy -> lots of superfluous compute
  - projecting energy back into realistic detector cells causes artefacts
- are point clouds a “way out” ?

S.Luo, W.Hu: Diffusion Probabilistic Models for 3D Point Cloud Generation, arXiv:2103.01458





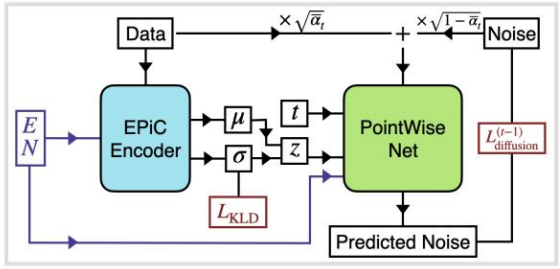


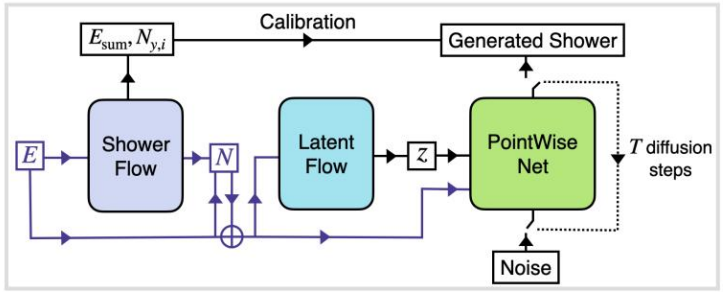
- recently many publications using point clouds and diffusion models
- can we adapt this to our HEP calorimeter use case - using much higher granularity provided by individual Geant4 steps ?

Fast calorimeter simulation (Diffusion)

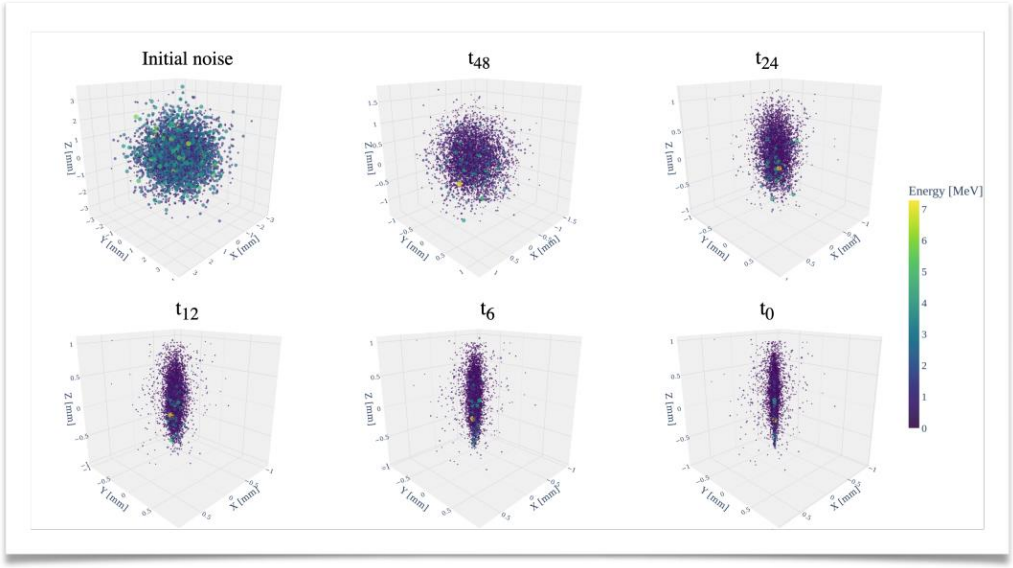
CaloCloud diffusion model

model architecture





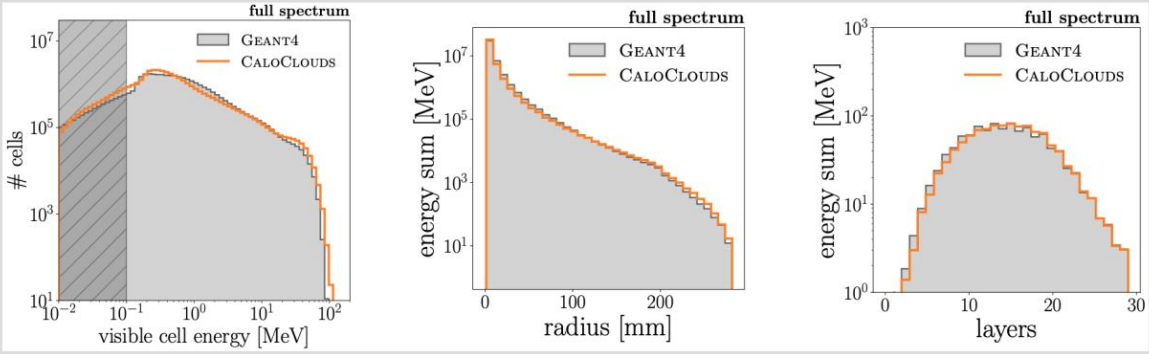
- training of PointWise Net with EPiC Encoder (e-Print: 2301.08128)
- inference uses two additional flows for number of space points, calibration and latent space



Fast calorimeter simulation (Diffusion)

CaloCloud diffusion model

performance

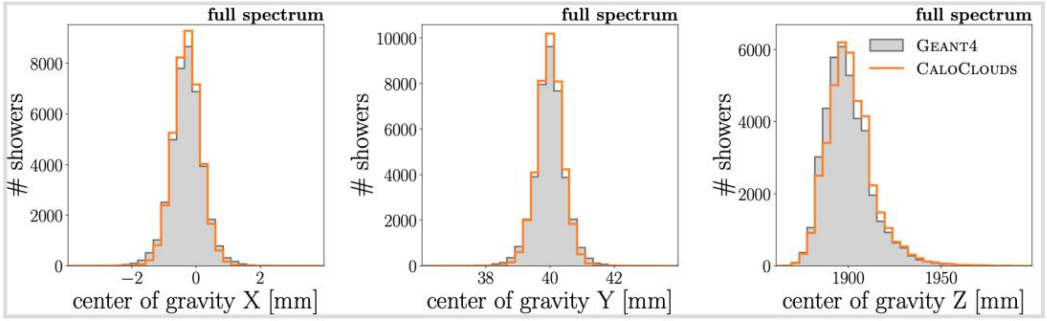


- observe overall very good physics fidelity on photon sample
- first successful application of diffusion models to (high granular) calorimeter simulation using point clouds

preprint will be available soon…

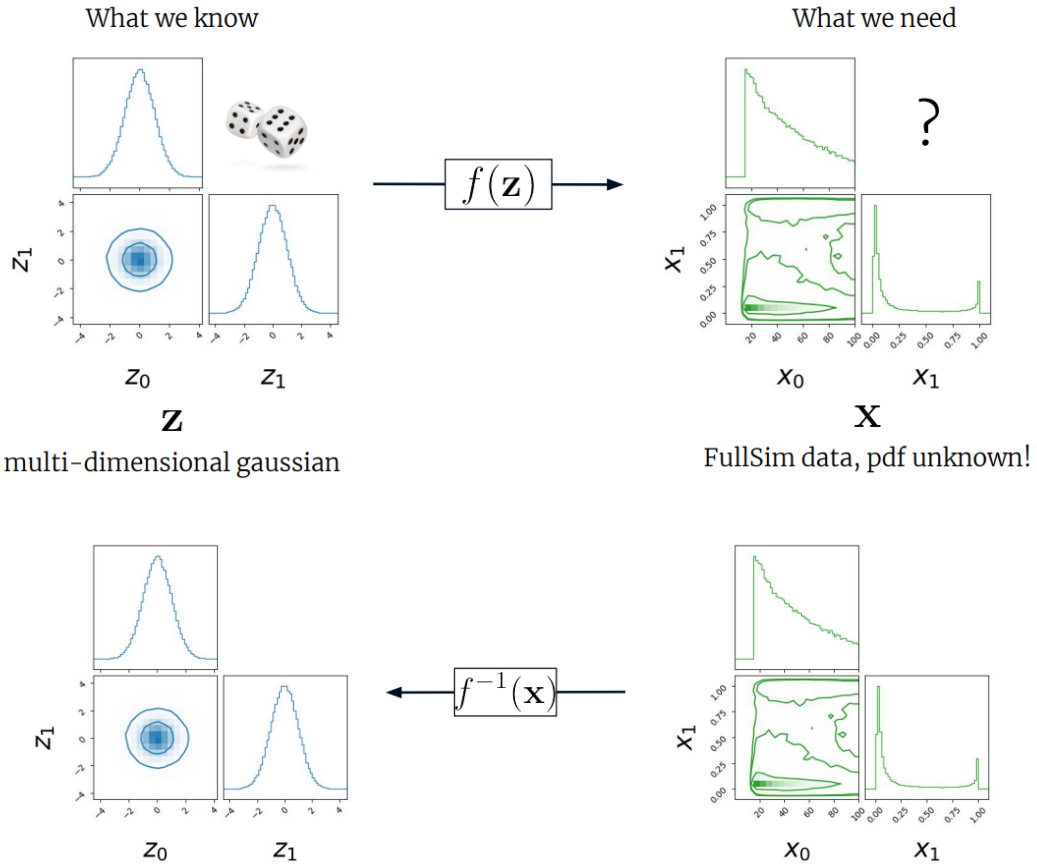
Hardware	Simulator	Time / Shower [ms]	Speed-up
CPU	GEANT4	4082 ± 170	×1
	CALOCLOUDS	3509 ± 220	×1.2
GPU	CALOCLOUDS	38 ± 3	×107

improvement of timing
focus of future work



Fast calorimeter simulation (Normalizing Flows)

Normalizing Flows:
generative model for *pdf*s!



$$\mathbf{x} = f(\mathbf{z})$$

$$p_x(\mathbf{x}) = p_z(\mathbf{z}) \det \left| \frac{d\mathbf{z}}{d\mathbf{x}} \right|$$

$$\log(p_x(x)) = \log(p_z(f^{-1}(\mathbf{x}))) + \log\left(\det \mathbb{J}_{f^{-1}}(\mathbf{x})\right)$$

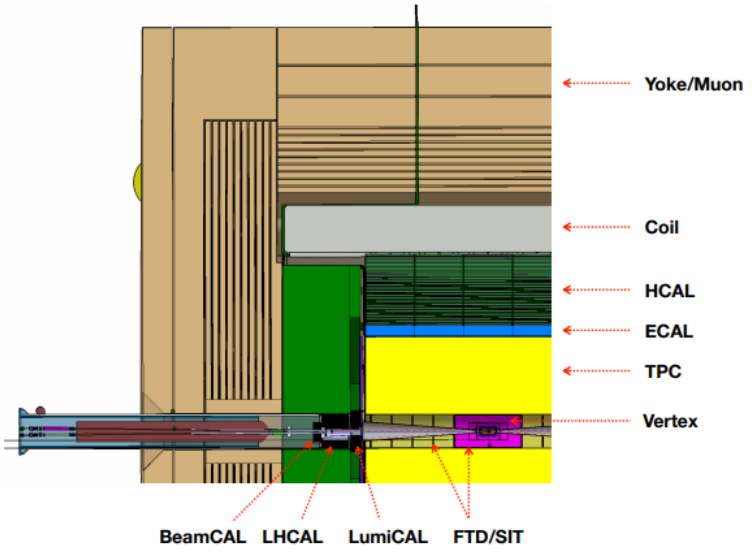
Fast calorimeter simulation (Normalizing Flows)

International Large Detector (ILD)

- proposed detector for the ILC
- has two sampling calorimeters
- electromagnetic calorimeter
  - 30 layers, 5mm x 5mm cells
- hadronic calorimeter
  - 48 layers, 30mm x 30mm cells

dataset[1]:

- photon showers in ECAL
- 30x30x30 voxels



graphics taken from "ILD: Design Report"[2]

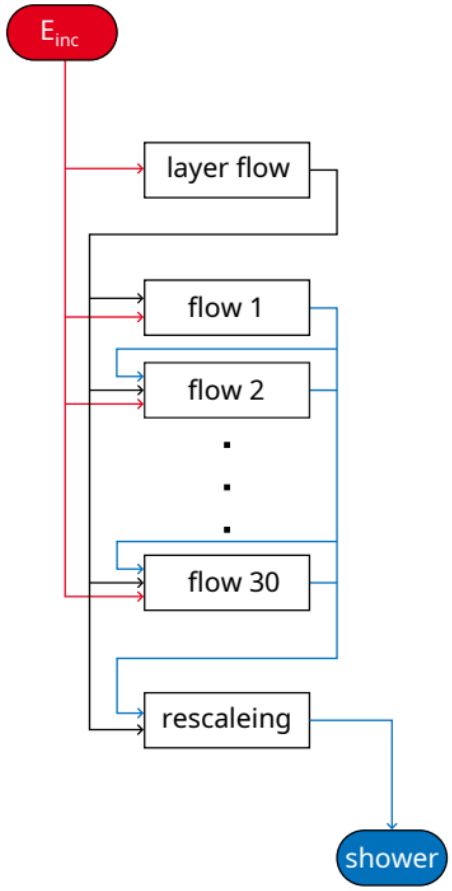
[1] Erik Buhmann et al. *Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed*. 2021. arXiv: 2005.05334.

[2] ILD Concept Group. *International Large Detector: Interim Design Report*. 2020. arXiv: 2003.01116.

Fast calorimeter simulation (Normalizing Flows)

- based on CaloFlow[3] and L2L Flow[4]
- one layer flow
  - learns distribution of layer energies
  - conditioned on incident energy
- 30 multiple flows
  - learn shower shape in layer
  - conditioned on
    - incident energy
    - layer energy
    - privies layers
  - summary network reduces cardinality
- generation
  - sample layer energies using layer flow
  - sample shower shape using multiple flows
  - rescale voxel energies

Architecture

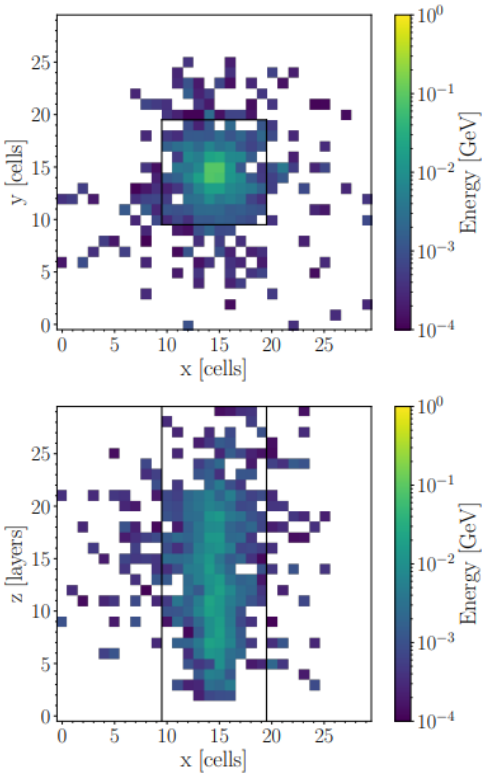


[3] Claudius Krause and David Shih. *CaloFlow: Fast and Accurate Generation of Calorimeter Showers with Normalizing Flows*. 2021. arXiv: 2106.05285.

[4] Sascha Diefenbacher et al. *L2LFlows: Generating High-Fidelity 3D Calorimeter Images*. 2023. arXiv: 2302.11594.

Fast calorimeter simulation (Normalizing Flows)

Upscaling



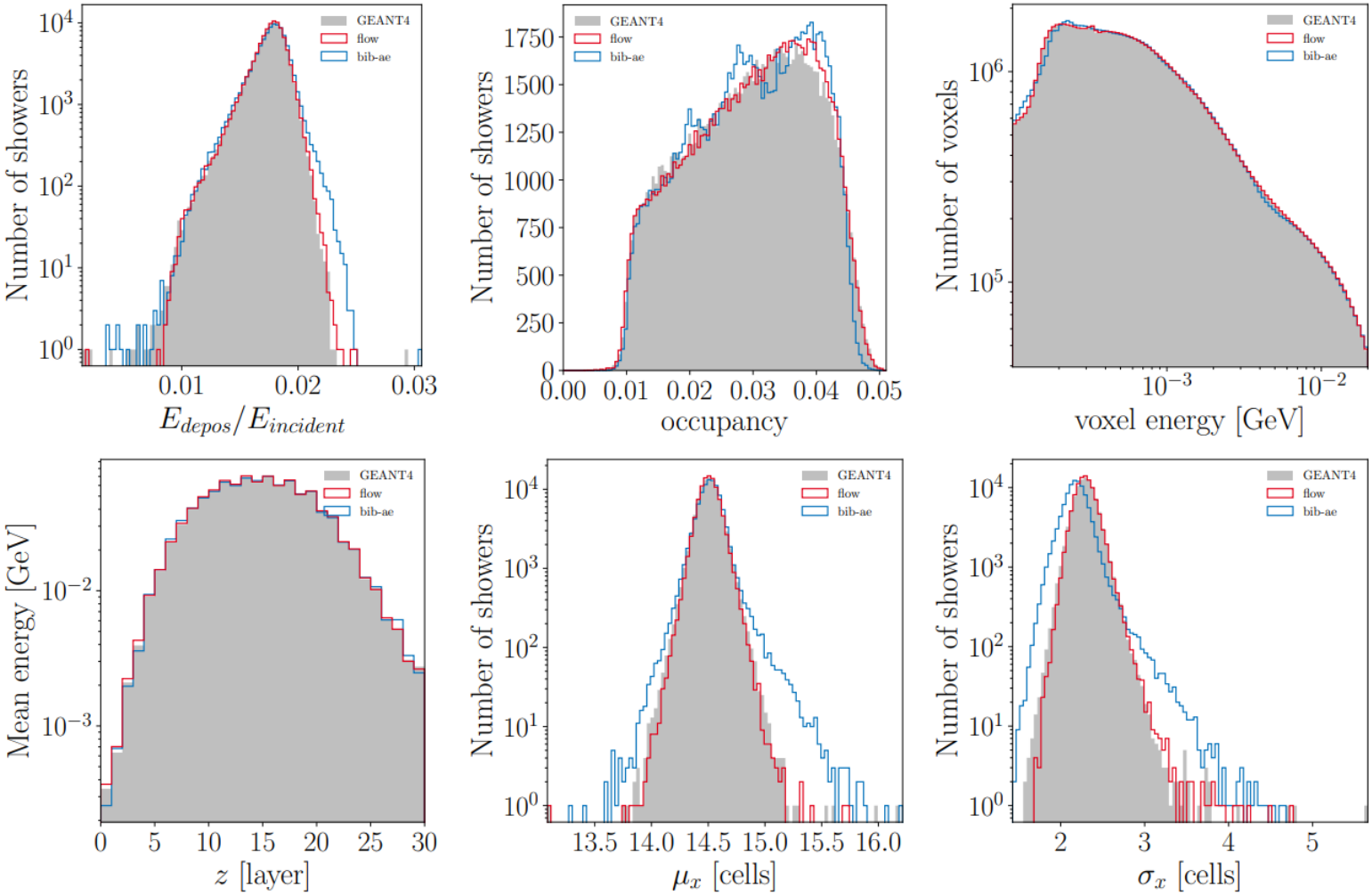
photon shower

- scaling up to 30x30x30 voxels
- switch to convolution flows
  - using multi scale architecture like in Glow[5]
  - faster generation
  - less weights
  - more accurate showers
- improve training
  - apply gradient clipping
  - add LR scheduler
- features in energy spectrum are smeared out
  - → apply element with function to get them back

[5] Diederik P. Kingma and Prafulla Dhariwal. *Glow: Generative Flow with Invertible 1x1 Convolutions*. 2018. arXiv: 1807.03039.

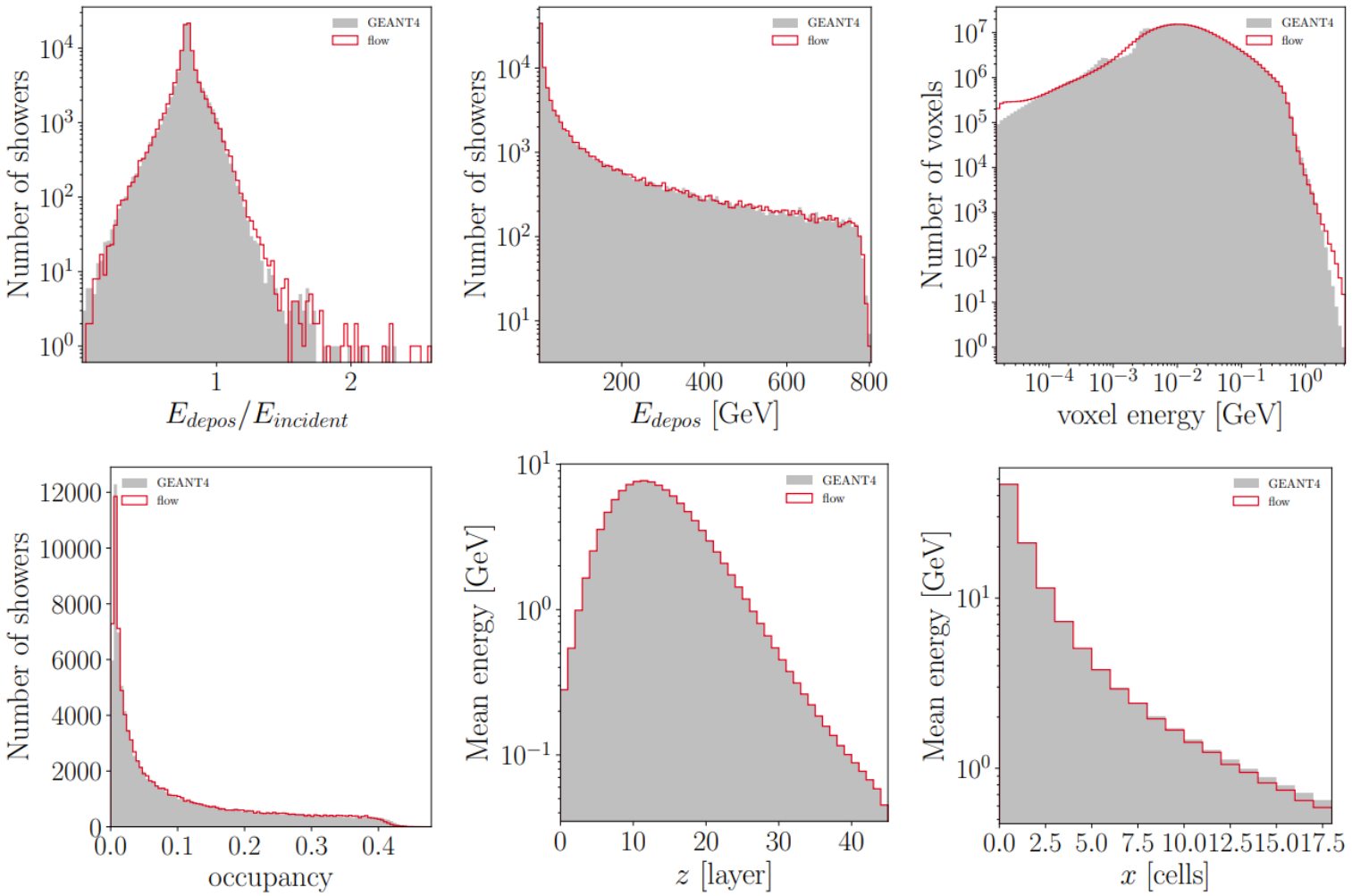
Fast calorimeter simulation (Normalizing Flows)

Preliminary Results



Fast calorimeter simulation (Normalizing Flows)

Calo Challenge



Fast calorimeter simulation (Normalizing Flows)

Timing

Simulator	Hardware	Batch size	time [ms]	Speedup
GEANT4	CPU	1	4081.53 ± 169.92	×1.0
BIB-AE	CPU	1	102.25 ± 0.64	×40.0
		10	37.81 ± 0.13	×108.0
		100	48.51 ± 0.01	×84.1
		1000	48.19 ± 0.01	×84.7
Flow	CPU	1	1746.61 ± 64.50	x2.3
		10	392.61 ± 0.34	x10.4
		100	228.86 ± 7.09	x17.8
		1000	275.55 ± 3.01	x14.8
BIB-AE	GPU	1	74.22 ± 3.18	×42.5
		1000	0.249 ± 0.002	×16326.1
Flow	GPU	1	2471.07 ± 70.20	x1.7
		1000	3.39 ± 0.09	×1202.3

LHCb ultra-fast simulation with Lamarr

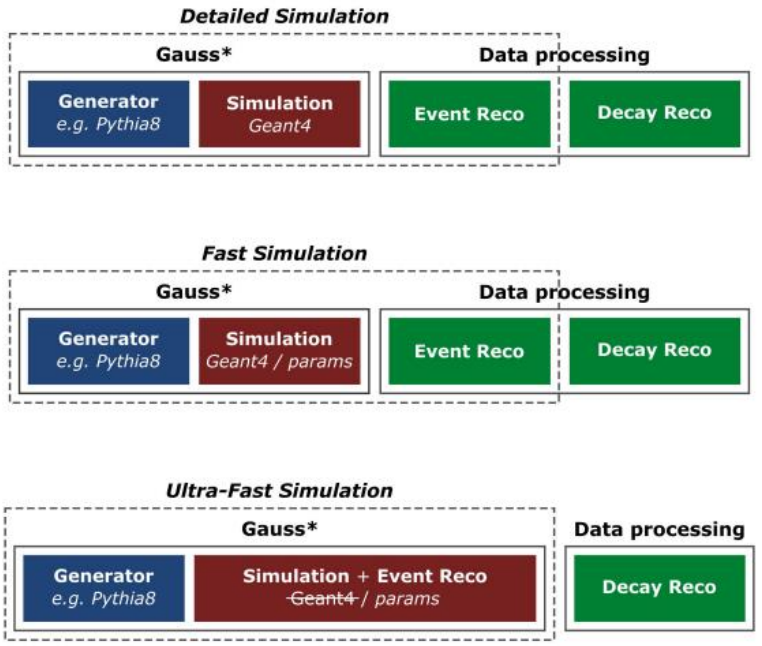
Fast simulation VS. ultra-fast simulation

Fast Simulation techniques aim to speed up the Geant4-based simulation production:

- Simulation framework upgrade (see first Michal's talk)
- Reducing the detector geometry (*e.g.*, track-only sim)
- Reuse of the underlying events, **ReDecay** [2]
- Parameterizing **energy deposits** instead of relying on Geant4 (*e.g.*, shower libraries [3] or CaloGAN [4])

Ultra-Fast Simulation strategies replace Geant4 with parameterizations able to transform generator-level particles into analysis-level reconstructed objects [5].

* **Gauss** is the LHCb simulation framework based on Gaudi [1]



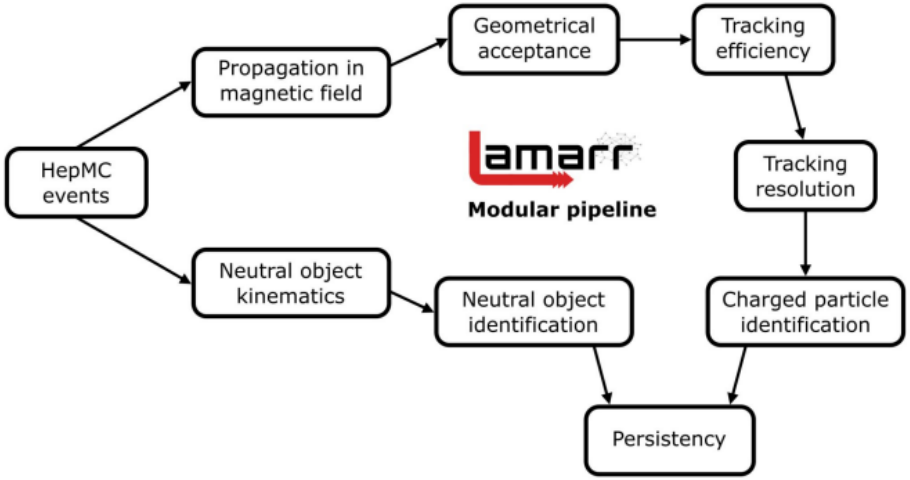
LHCb ultra-fast simulation with Lamarr

Lamarr: the LHCb ultra-fast simulation option

Lamarr is the novel ultra-fast simulation framework of LHCb, able to offer the fastest options for simulation. Lamarr consists of a **pipeline of** (ML-based) **modular parameterizations** designed to replace both the simulation and reconstruction steps [6, 7].

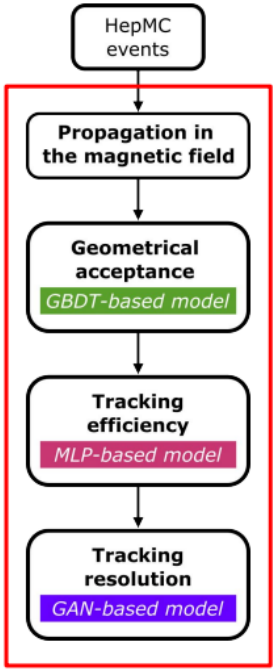
Lamarr is integrated with the LHCb simulation framework:

- compatibility with all the **LHCb-tuned generators**
- compatible with the **distributed computing** middleware (LHCbDirac) and production environment
- able to provide datasets in the same format used for analysis



LHCb ultra-fast simulation with Lamarr

Tracking system models

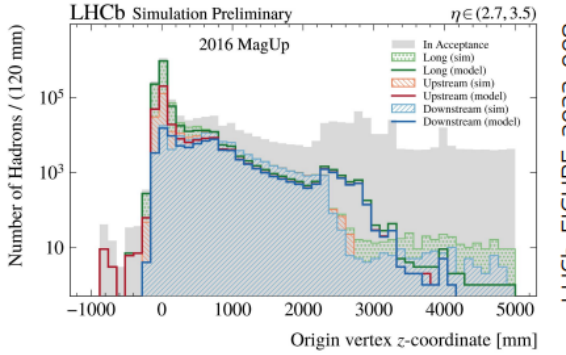


Lamarr Tracking pipeline

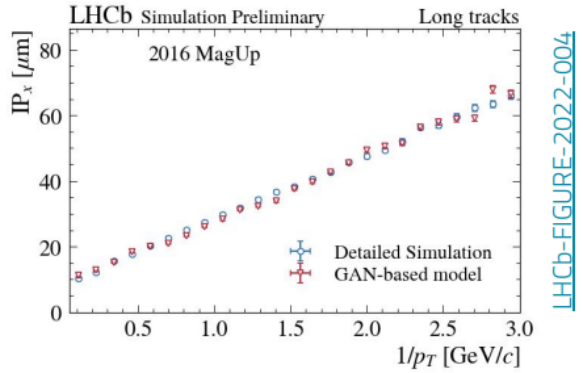
Lamarr parameterizes the **LHCb Tracking system** mostly relying on a set of (ML-based) modules:

- **acceptance** → predict which of the generated tracks fall in the geometrical acceptance of the experiment
- **efficiency** → predict which of the generated tracks in acceptance are properly reconstructed by the detector
- **resolution** → convert the generator-level parameters of the reconstructed tracks into analysis-level ones, including track-quality features

A major effort is ongoing to model correctly the Tracking system in function of the **type of tracks**.



LHCb-FIGURE-2023-008



LHCb-FIGURE-2022-004

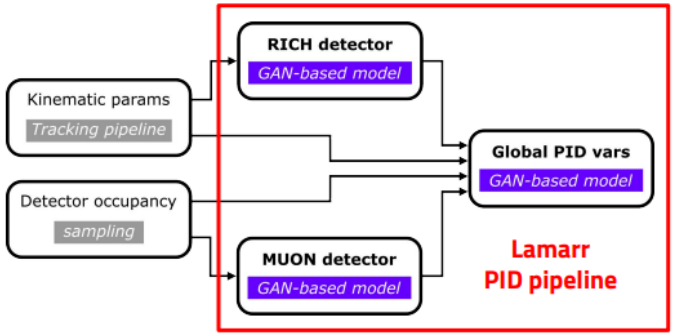
LHCb ultra-fast simulation with Lamarr

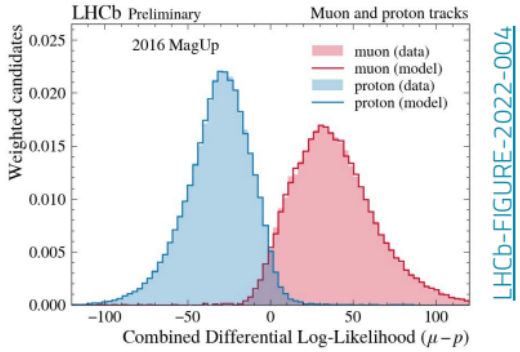
Particle identification system models

The high-level response of the LHCb PID system relies on ***Generative Adversarial Networks*** (GAN) [8, 9] trained on either detailed simulated samples or real data.

Lamarr provides RICH and MUON models for for **muon**, **pion**, **kaon** and **proton** tracks based on the kinematics of the reconstructed tracks and a description of the detector occupancy.

This information alone aren't enough to parameterize the **Global PID variables**, that also need the response of the RICH and MUON systems. Hence, Lamarr provides the higher-level response of the PID system relying on a **stack of GAN-based models**.





LHCb ultra-fast simulation with Lamarr

Electromagnetic calorimeter model

Parameterization of the **LHCb calorimeter** available in Lamarr designed for detector studies → not suitable for simulation production

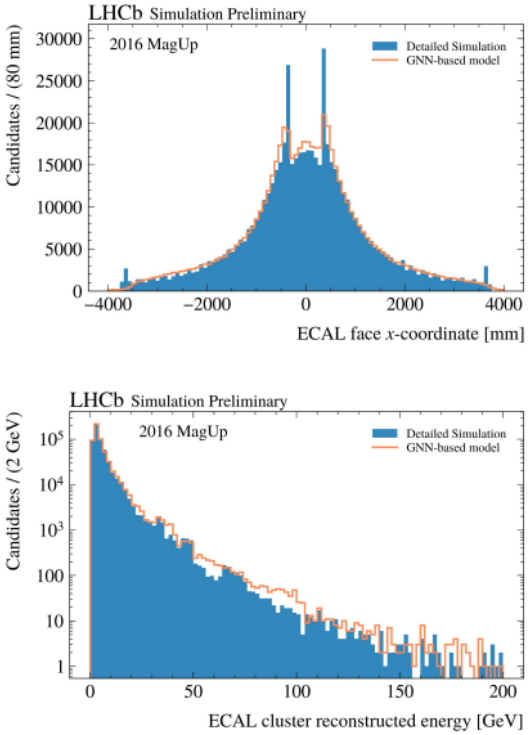
Improving the ECAL models is a necessary step if we want that Lamarr provides reliable parameterizations also for **photons** and **electrons**.

Simulating ECAL with an ultra-fast approach requires to face the **particle-to-particle correlation problem**:

- sequence of *n* generated photons → sequence of *m* reconstructed clusters (in general, with $n \neq m$)
- approached as a **translation problem**

Two strategies are currently under investigation:

- ***Graph Neural Networks*** (GNN) [10, 11]
- ***Transformer*** [12, 13]



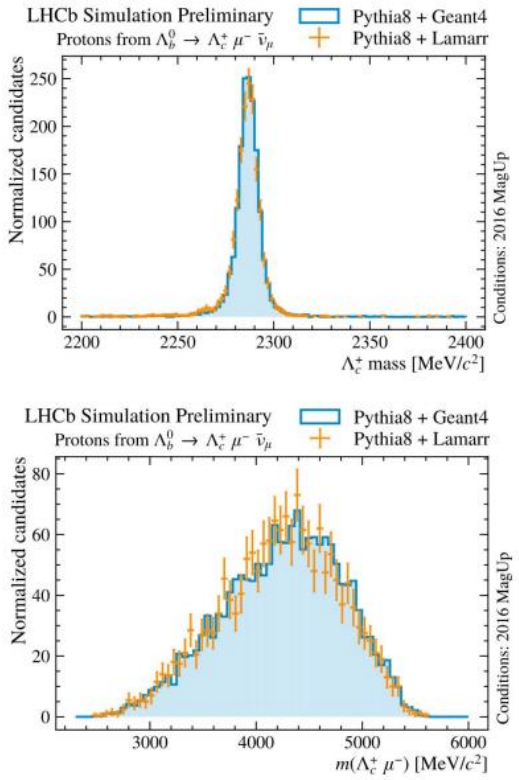
LHCb-FIGURE-2023-008

LHCb ultra-fast simulation with Lamarr

Lamarr validation campaign

Lamarr is currently under validation, comparing the distributions of the **analysis-level reconstructed quantities** parameterized with what obtained from Detailed Simulation.

- semileptonic decay mode: $\Lambda_b^0 \to \Lambda_c^+ \mu^- X$ with $\Lambda_c^+ \to p K^- \pi^+$
  - crucial the interface with LHCb-tuned generators
- **muons**, **pions**, **kaons** and **protons** in a single decay
  - all particle species for which Lamarr provides parameterizations
- Lamarr-based samples, detailed simulated samples and plots obtained from the **LHCb analysis software**
  - testing the integration with the current version of Gauss



LHCb-FIGURE-2022-014

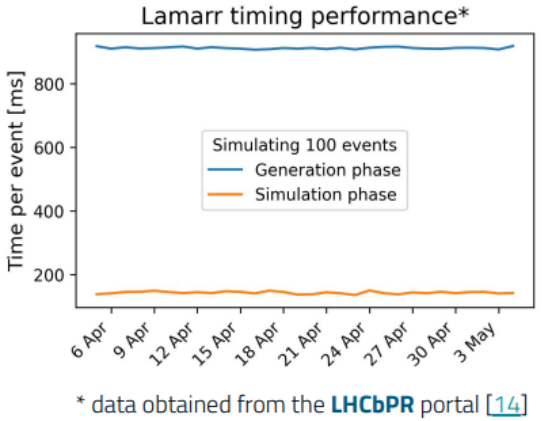
LHCb ultra-fast simulation with Lamarr

Preliminary timing studies

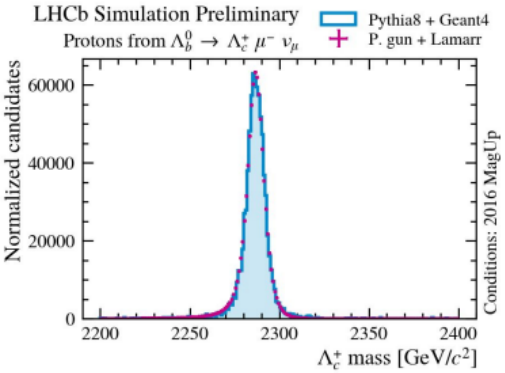
Comparing the normalized CPU spent for Geant4-based and Lamarr simulations of $\Lambda_b^0 \to \Lambda_c^+ \mu^- X$ decays we estimate a CPU reduction of **two-order-of-magnitude** for the Simulation phase.

Since the generation of *b*-baryons is exceptionally expensive, Pythia8 becomes the **major consumer of CPU** for simulation in the ultra-fast paradigm.

A **further speed-up** can be reached reducing the cost for generation, for example simulating only the signal particles (*i.e.* with ***Particle Guns***) and avoiding at all the simulation of the *pp* collisions, not needed since Lamarr models the detector occupancy.



* data obtained from the **LHCbPR** portal [14]



LHCb-FIGURE-2022-014

Lamarr will never replace the Geant4-based simulation, but may provide soon a precious tool to reduce the pressure on CPU of Detailed Simulation. Lamarr is designed to meet most of the needs for simulation of physics groups, from **designing selection** strategies, **training multivariate classifiers**, to **studying systematics** or **correlation effects**.

Refining fast simulation using ML

INTRODUCTION

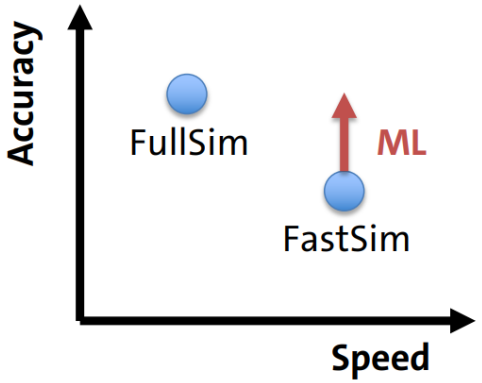
- FastSim **advantage in speed** comes at the price of **decreased accuracy** in some of the final observables

  (N.B.: standard analyses use further processed data formats e.g., AOD or NanoAOD)

GEN → Event generation	SIM → Detector simulation	DIGI → Digitization	RECO → Reconstruction	AOD → "Analysis Object Data"	NanoAOD further processed analysis format

➢ **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/… e.g., DCTR introduced in arXiv:1907.08209
    - **Refining** = changing (high-level) observables e.g., Wasserstein-GAN for air showers in arXiv:1802.03325



Refining fast simulation using ML

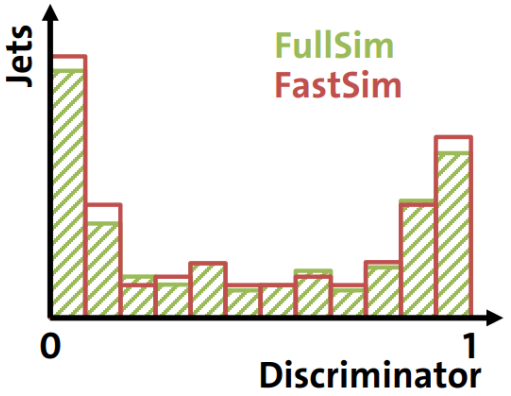
INTRODUCTION

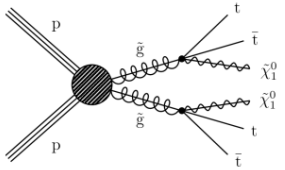
- Focus on jet flavour tagging: 4 NanoAOD **DeepJet discriminators**: b+bb+lepb, c vs b+bb+lepb, c vs uds+g, g vs uds

  „B“ „CvB“ „CvL“ „QG“

  (DeepJet NN softmax output nodes: b, bb, lepb, c, uds, g; arXiv:2008.10519)

  ➢ FastSim/FullSim discrepancies O(10 %)



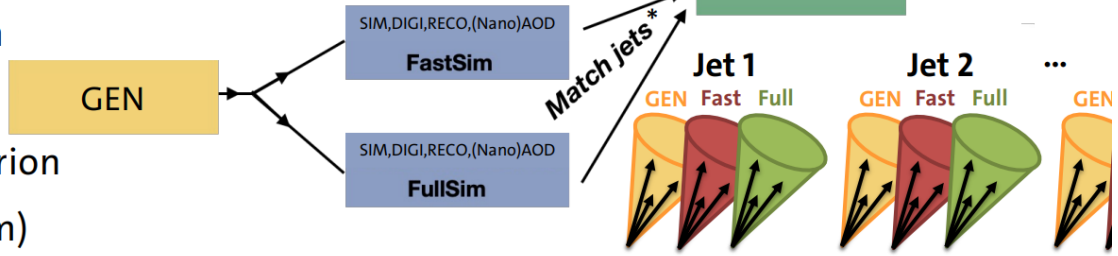


- **Training sample**: SUSY simplified model „T1tttt“ simulated with FastSim and FullSim

  (same GEN events, no pile-up)

  ➢ Match jets using ΔR angular criterion

  ➢ **Jet triplets**: (GEN, FastSim, FullSim)



$*\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < 0.2$

Refining fast simulation using ML

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

REFINING (REGRESSION) – LOSS TERMS

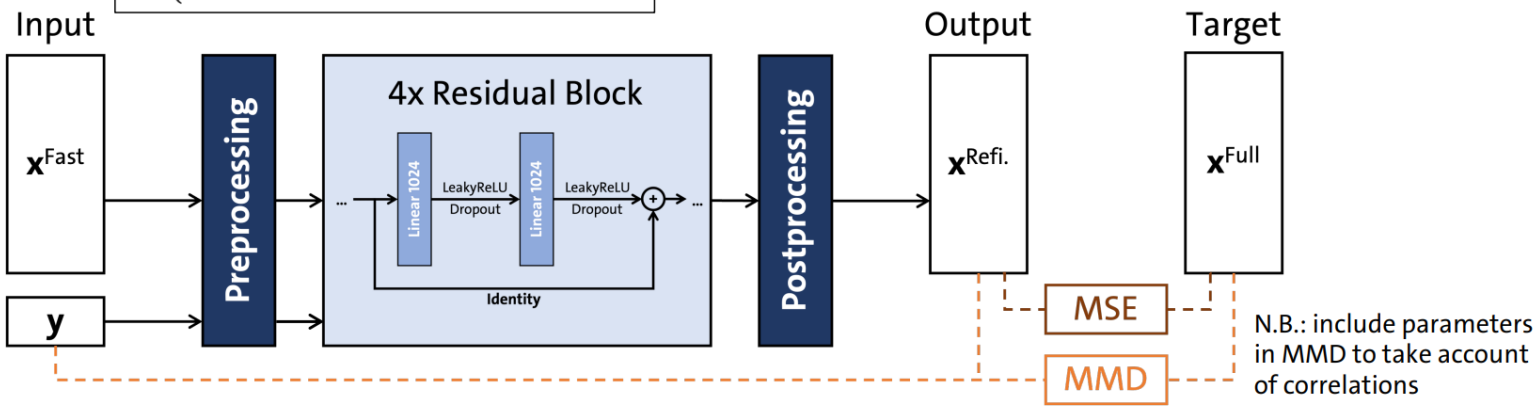
- **Primary loss: MMD** (distribution-based)
  - Comparing ensembles of jets not jet-jet pairs
  - To cope with independent stochasticity in both simulation chains

given two samples from P(X) and Q(Y):

$$\widehat{\mathrm{MMD}}(P,Q) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}k(x_i,x_j) + \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}k(y_i,y_j) - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}k(x_i,y_j)$$

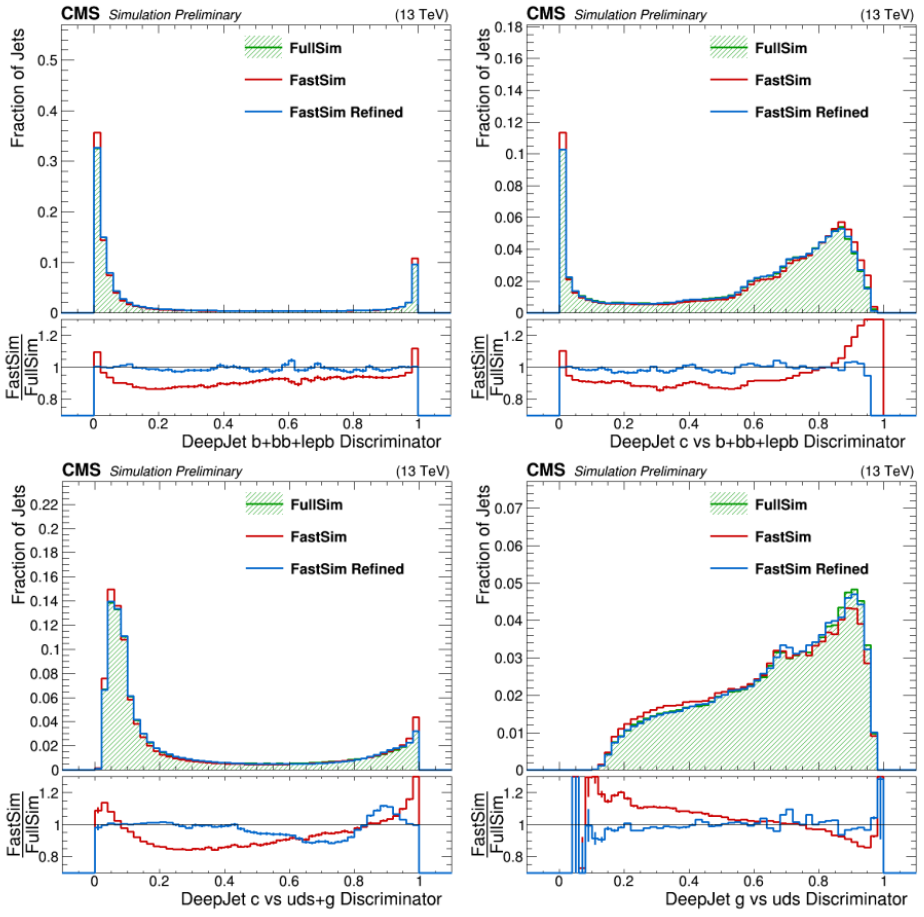
n = m = batch size = 4096
k: Gaussian kernel (adaptive σ)

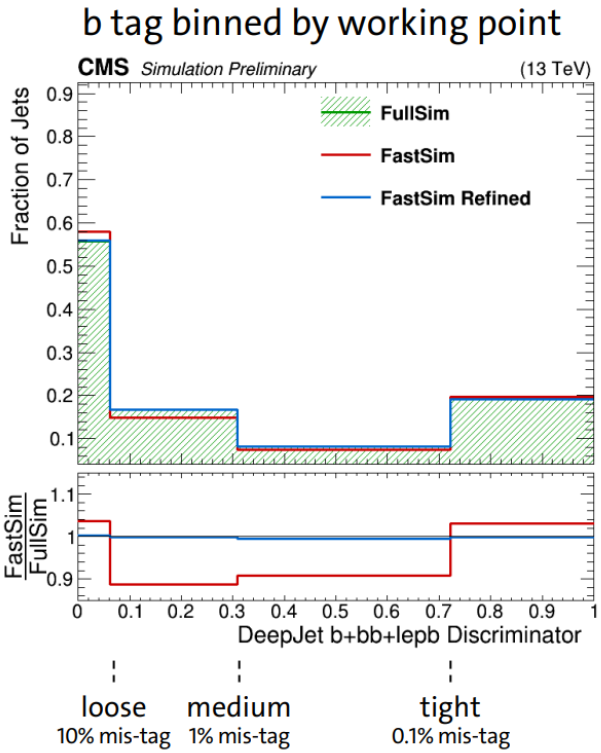
- **Additional loss: MSE/Huber** (output-target pair-based) → correct for deterministic FastSim biases
  - Use Huber loss: $l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$ (combination of MSE/MAE, less sensitive to outliers)



Refining fast simulation using ML

REFINING (REGRESSION) – RESULTS





Refining fast simulation using ML

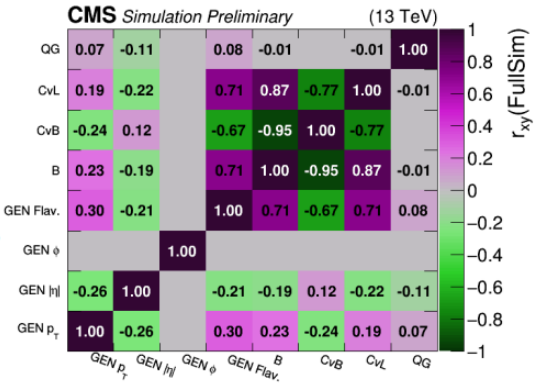
REFINING (REGRESSION) – RESULTS

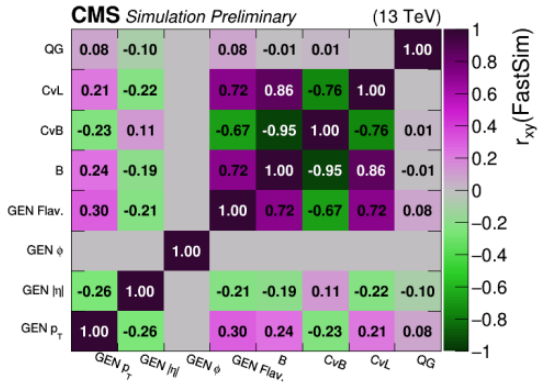
FullSim

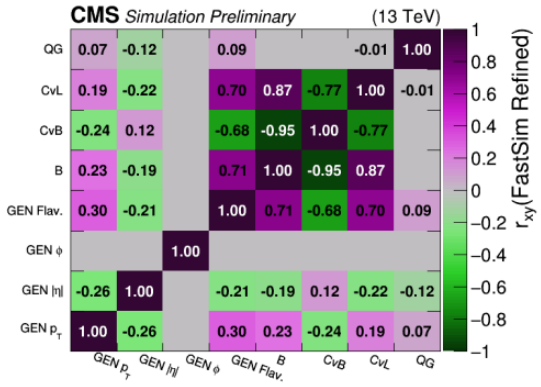
FastSim

FastSim Refined

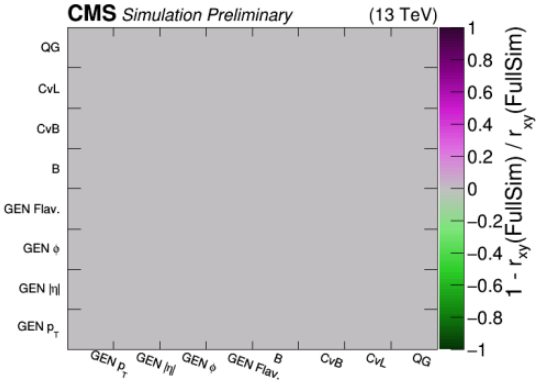
Pearson correlation coefficients

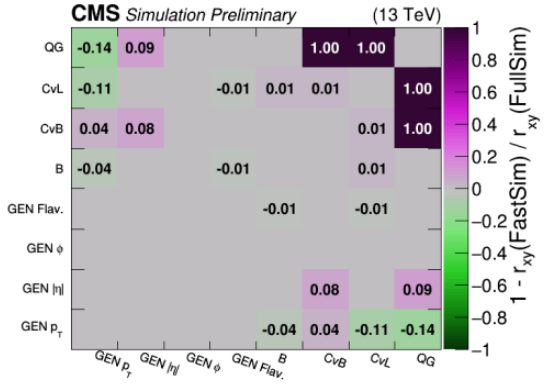


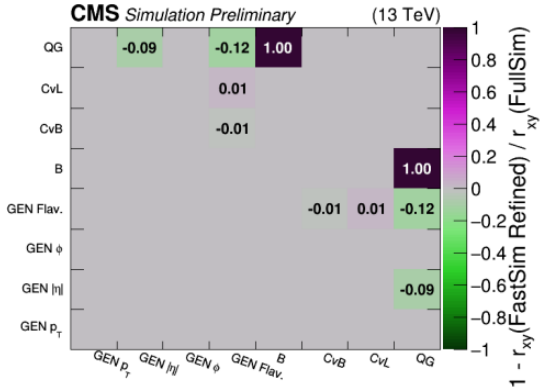




relative difference to FullSim



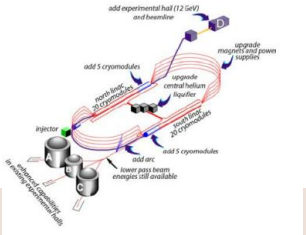


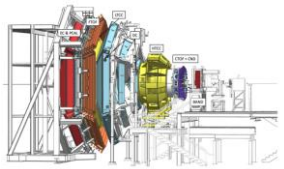


Reconstruction

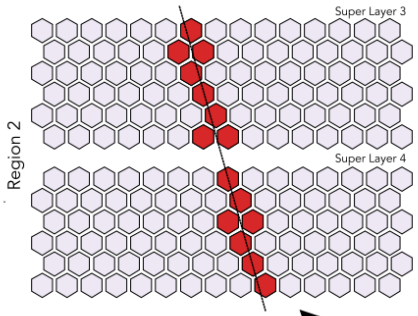
Track Identification for CLAS12 using AI

- CEBAF Large Acceptance Spectrometer (CLAS12) Located in Hall-B at Jefferson Lab

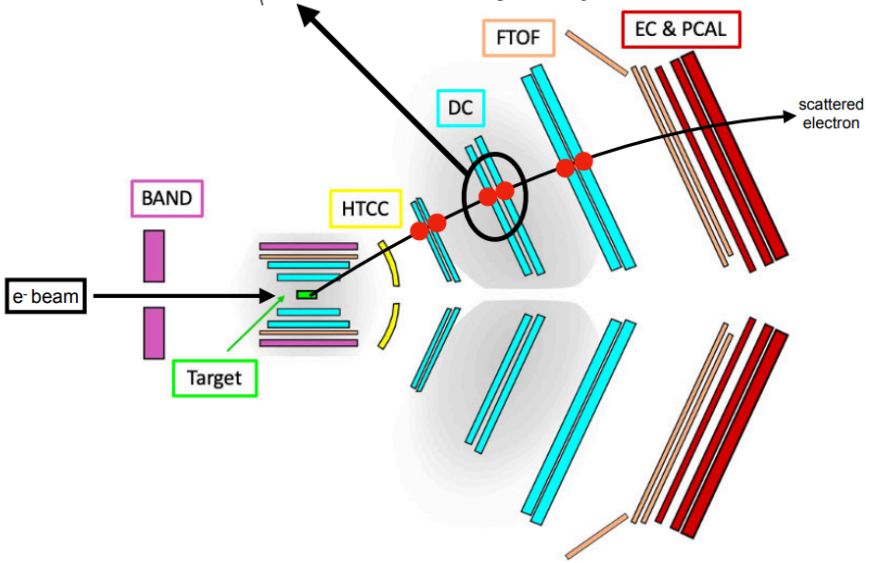




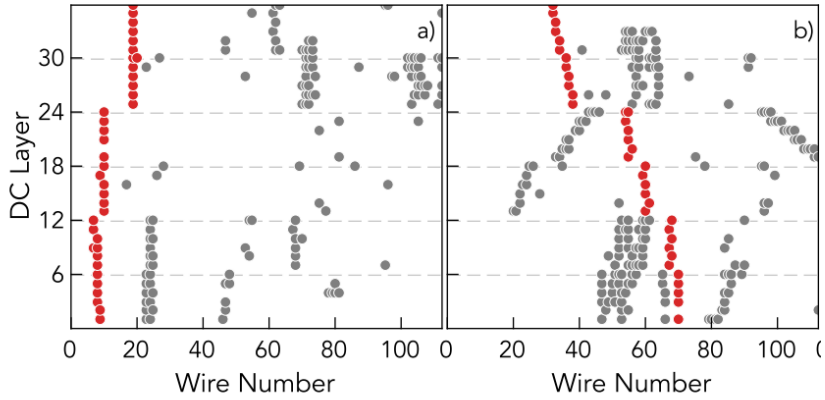
CLAS12 Detector



- 2 super layers in each region
- 6 wire planes in each super layer with 6-degree tilt relative to each other, (112 wires in each plane)
- Clusters in each super layer are considered part of the track trajectory

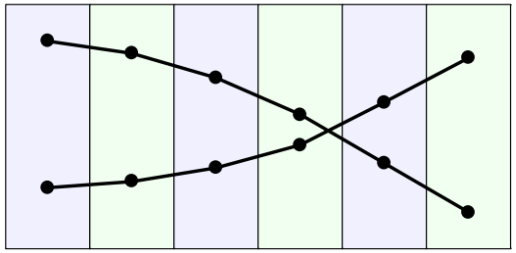


- Charged particle tracking is computationally extensive (about 80% of data processing time)
- The multi-particle final states produce numerous clusters in each sector which have to be analyzed to find the right combinations of clusters that form a track
- Identifying correct cluster combinations can speed up the tracking process and improve efficiency



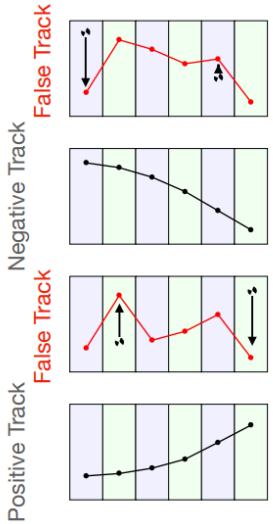
Track Identification for CLAS12 using AI

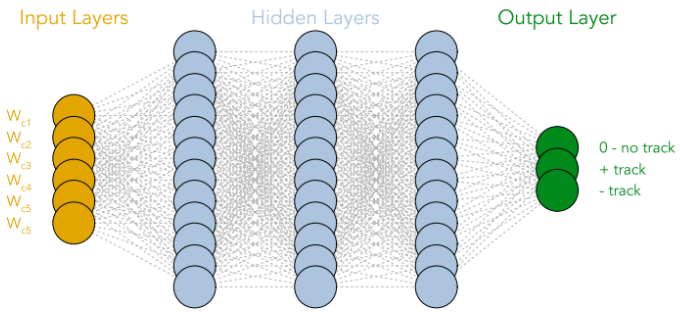
Physics Results



- True tracks are identified by conventional algorithms from real data.
- One negative and one positive track (different curvature due to magnetic field)
- False tracks are constructed by interchanging randomly one or two clusters with the clusters from the other track in the event

- The average wire position in each super layer is used as an input to Multi-Layer Perceptron (MLP)
- The network is trained on 6 inputs and produces three outputs:
  - False track
  - Negative Track
  - Positive Track

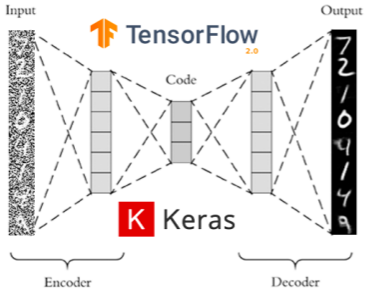




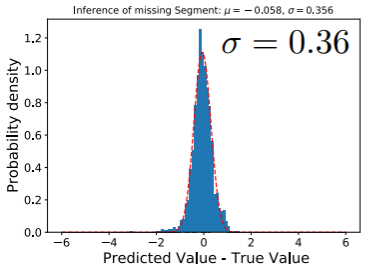
Track Identification for CLAS12 using AI

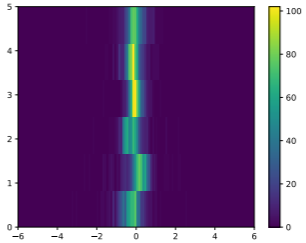
Corruption Auto-Encoder

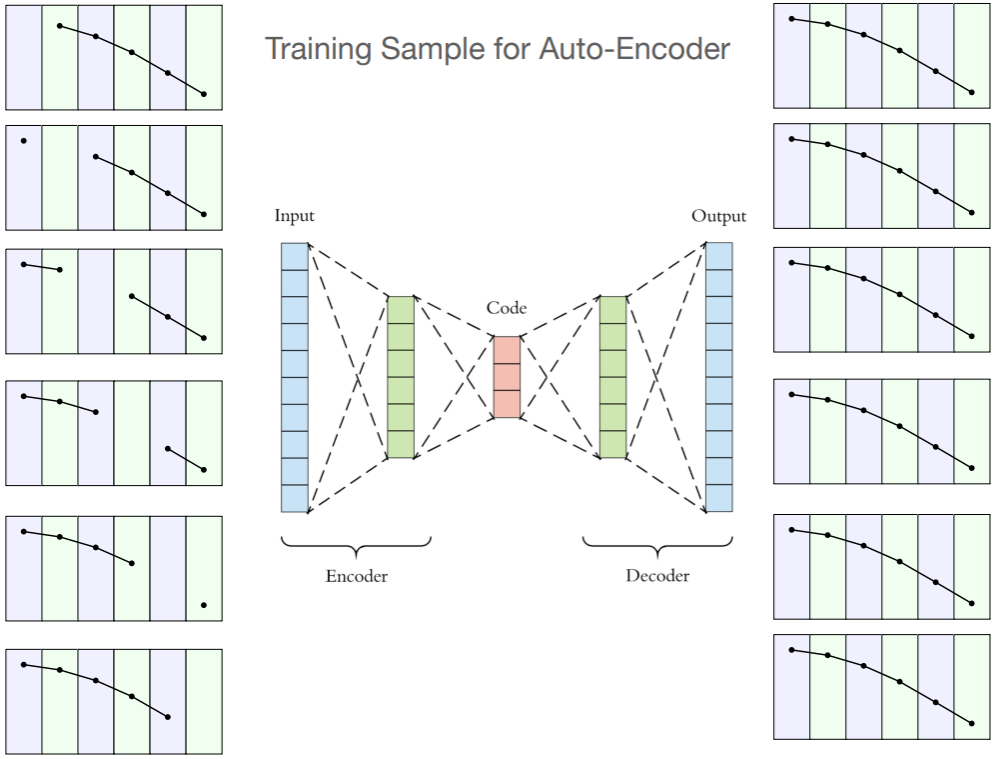
- An auto-encoder is composed of an encoder and a decoder sub-models. The encoder compresses the input and the decoder attempts to recreate the input from the compressed version provided by the encoder.
- **Typically used for de-noising, but can be used for fixing glitches (our case).**



- The network Predicts the missing cluster position with a precision of 0.36 Wire



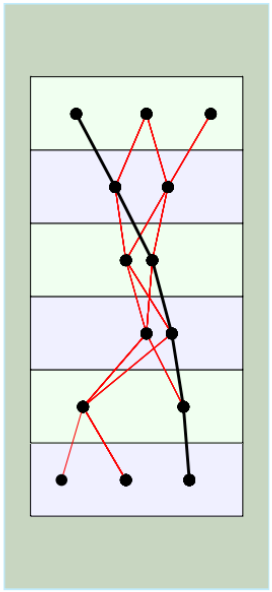




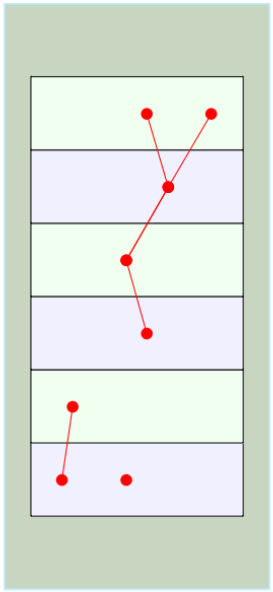
- Use Auto-Encoders to fix the missing cluster (provide a position)
- Good reconstructed tracks are used to generate training samples by removing one cluster from each super layer

Track Identification for CLAS12 using AI

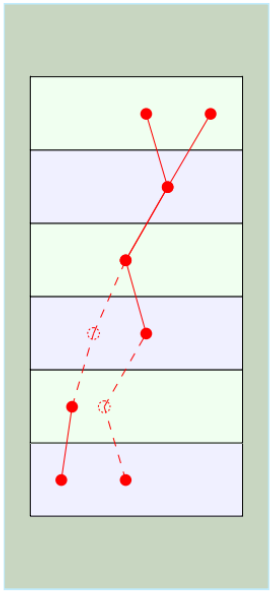
Putting things together



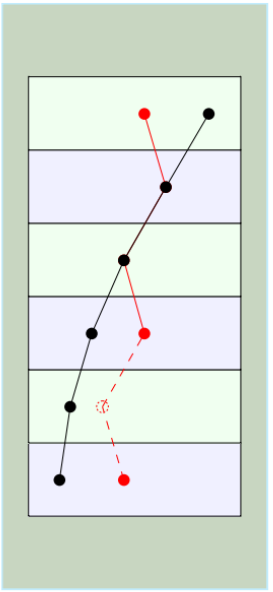
Classifier picks the correct track from 6 super-layer combinations



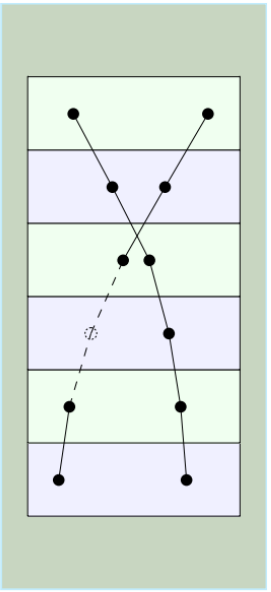
Remove all clusters belonging to identified track



Construct pseudo-clusters for all 5 super layer combinations using Corruption Auto-Encoder



Identify tracks using 6 super layer candidates with pseudo-clusters

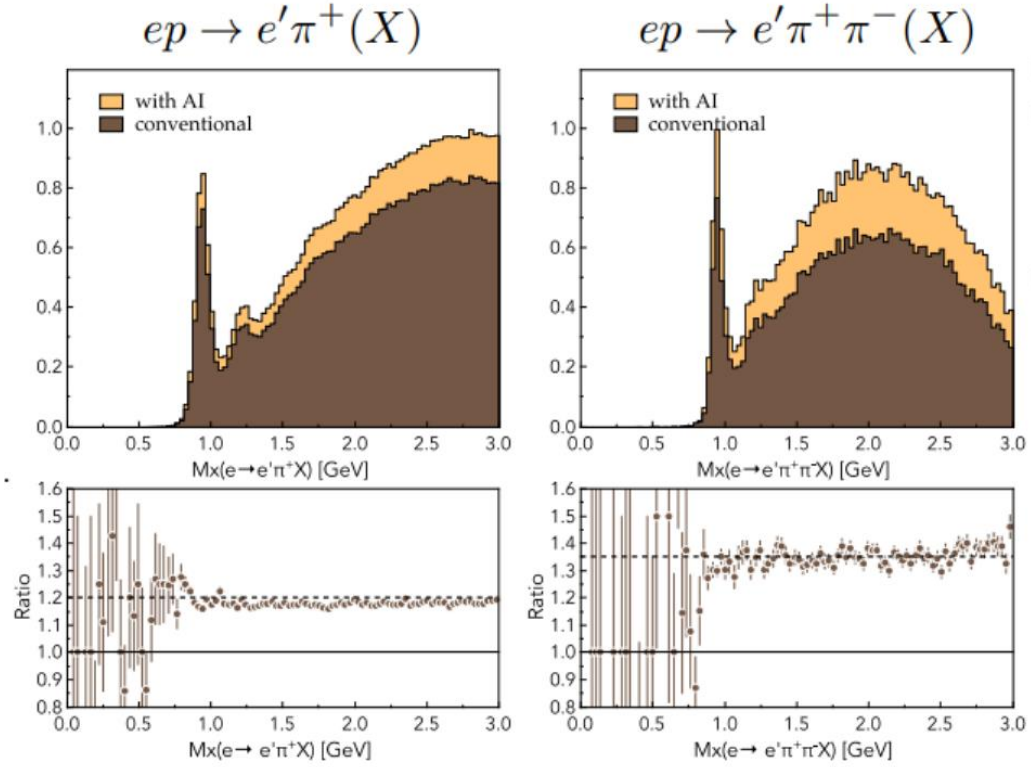


Voila!

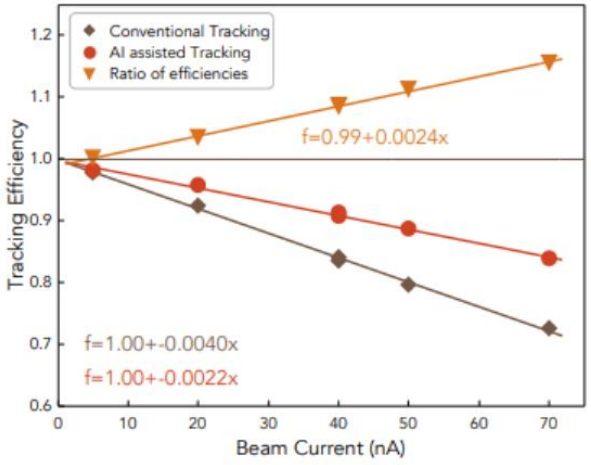
Track Identification for CLAS12 using AI

Physics Results

AI-assisted track candidate classification and Inefficiency Reduction Auto-Encoder



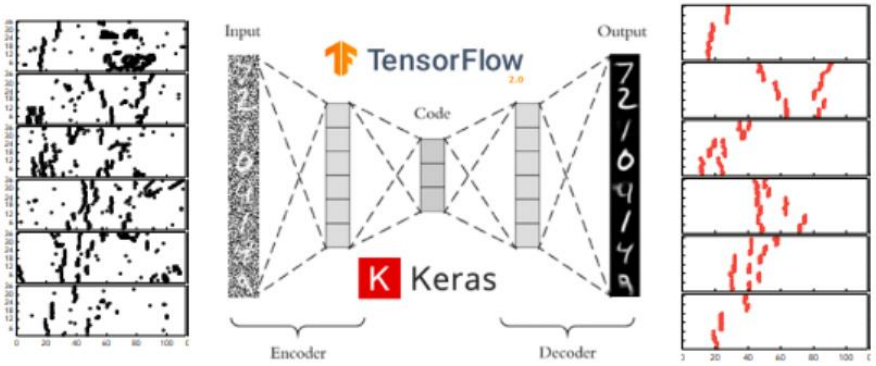
- Single particle efficiency increases by ~10%.
- The impact on physics for a multi-particle final state is dramatic (20% for the two-particle final state and ~35% for the three-particle final state)
- The tracking code speedup is ~30%.

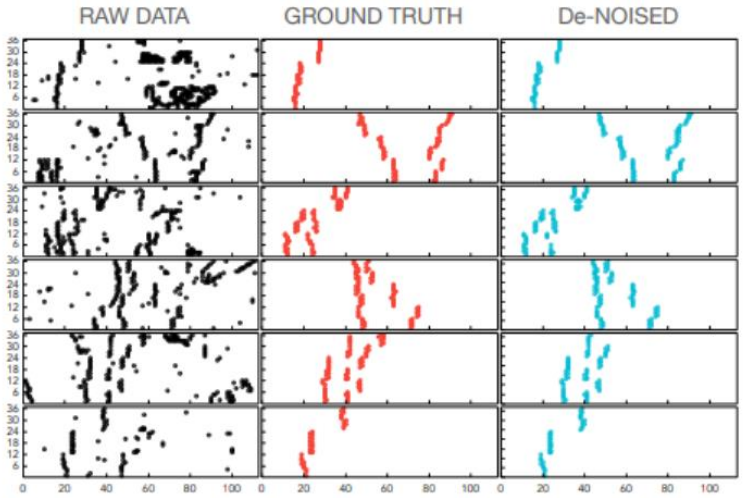


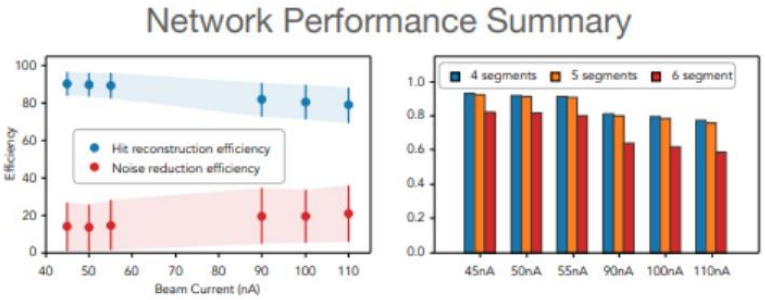
Track Identification for CLAS12 using AI

De-Noising

- Convolutional Auto-Encoder is used to de-noise raw data from drift chambers.
- The network is trained on reconstructed data with track hits isolated from raw DC hits.
- The network is able to isolate hits that potentially belong to a valid track through drift chambers



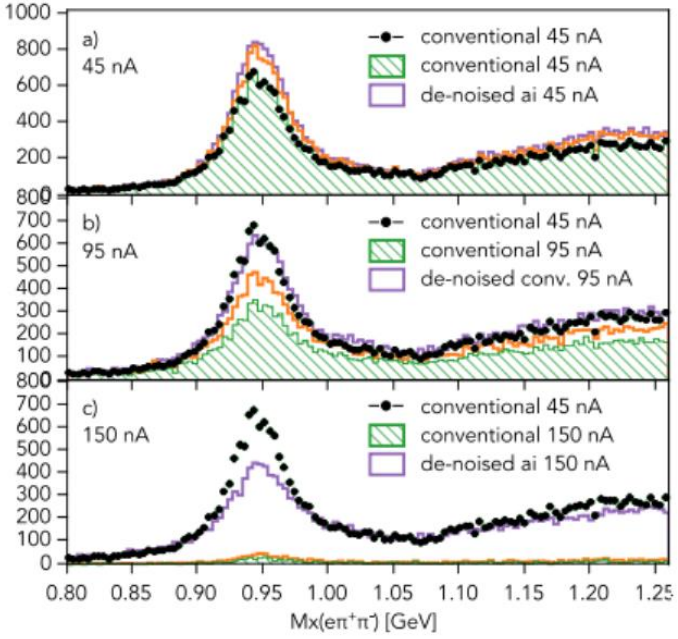


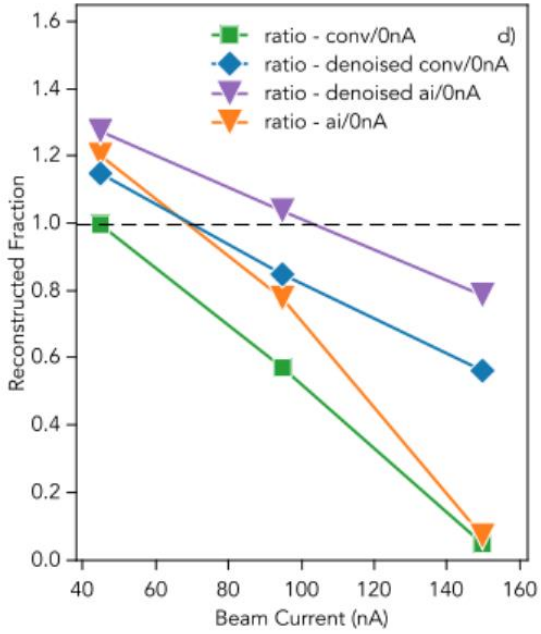


Track Identification for CLAS12 using AI

De-Noising Results (simulation)

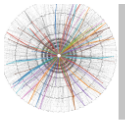
- The reconstruction is run on simulated data with a merging background for different incident beam currents (luminosity)
- The simulated three-particle final state is analyzed to measure yield for de-noised data and for conventional





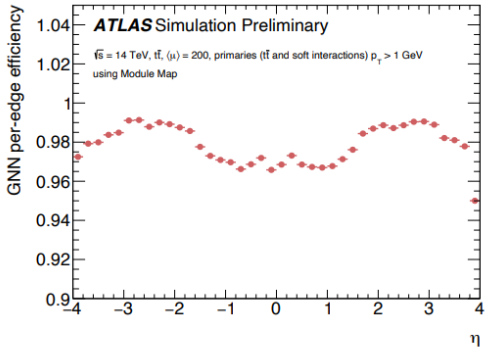
- At standard running luminosity, the de-noising slightly increases the yield compared to AI-assisted tracking.
- With increased luminosity, the de-noising helps to increase the yield significantly compared to conventional and AI-assisted tracking.
- Simulation underestimates the gain in yield significantly. In data the gain is much larger.

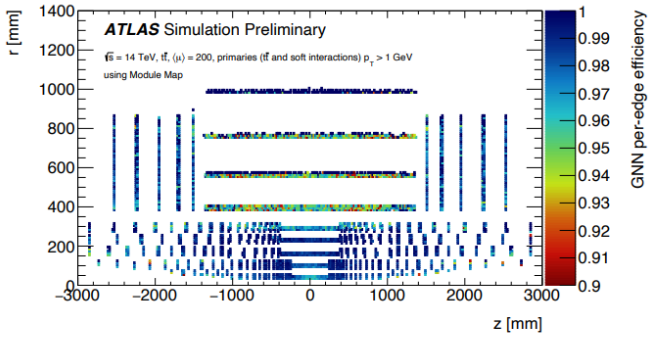
Novel fully-heterogeneous GNN designs for track reconstruction at the HL-LHC



CTD2022 GNN performance on ITk simulated data

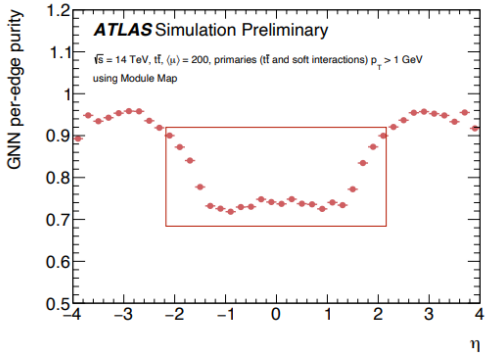
CTD2022

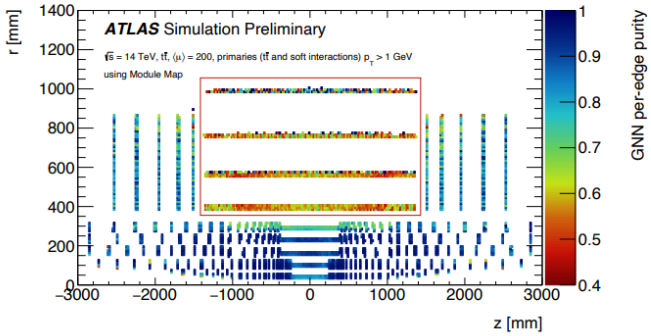




$$efficiency = \frac{n_{true>treshold}}{n_{true}}$$

- High efficiency: ~98%
- Slight drop of efficiency in the STRIP BARREL region

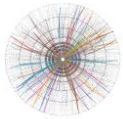




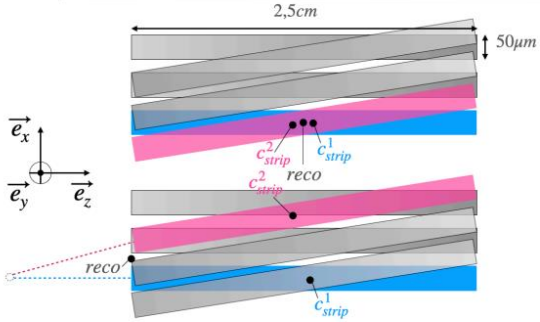
$$purity = \frac{n_{true>treshold}}{n_{true>treshold} + n_{fake>treshold}}$$

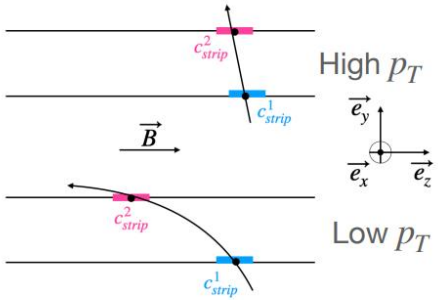
- Global purity ~85%
- Important drop of purity in the STRIP BARREL region

Novel fully-heterogeneous GNN designs for track reconstruction at the HL-LHC



STRIP low spatial resolution and Heterogenous Data

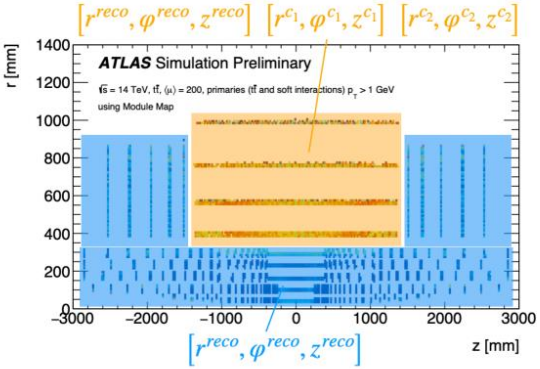




$$r^{reco}, \varphi^{reco}, z^{reco} = f_{strip}(c^1_{strip}, c^2_{strip})$$

The mechanism that led to the poor purity in CTD2022 results is understood: straight line approximation used in the ATLAS space point reconstruction leads to poor z resolution ($O(cm)$!) at low p_T

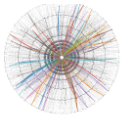
Impossible to exactly reconstruct space point position without knowing curvature of the track !



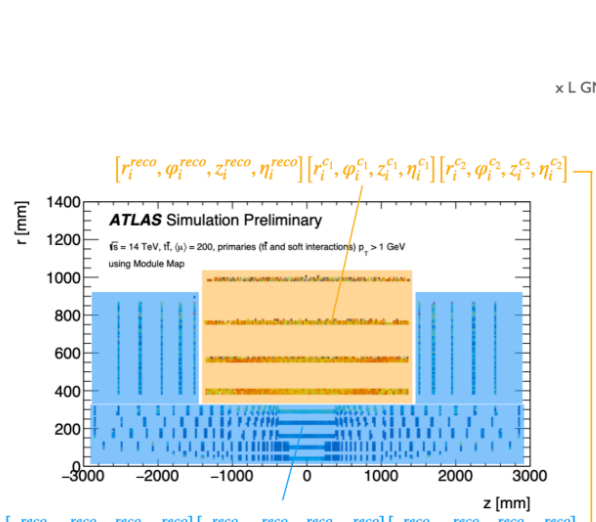
Key Idea: Give as node features in STRIP BARREL the **STRIP clusters data**
Hopefully the GNN will be able to learn a **better representation of the Space Point**

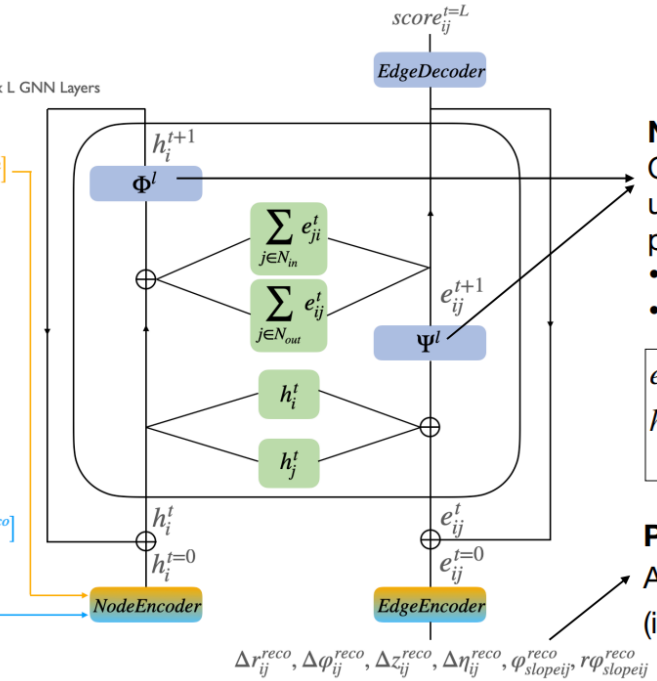
GNN combines space points into tracks, i.e. has access to curvature !

Novel fully-heterogeneous GNN designs for track reconstruction at the HL-LHC



Additional GNN improvements





Non recurrent GNN layers

One instance of edge updater and node updater per GNN layer (No shared parameters)

- Better function estimation
- Avoid vanishing gradients

$$e_{ij}^{t+1} \leftarrow \psi^l(e_{ij}^t | h_i^t | h_j^t)$$
$$h_i^{t+1} \leftarrow \phi^l(h_i^t | \sum_{j \in N_{in}} e_{ji}^t | \sum_{j \in N_{out}} e_{ij}^t)$$

Preprocessed edge features

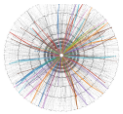
Add φ_{slope} which is closely related to p_T (informative for the curvature of the tracks)

Global architecture

Move to 3 layers per MLP (vs 2 layers per MLP in the CTD2022 GNN model)

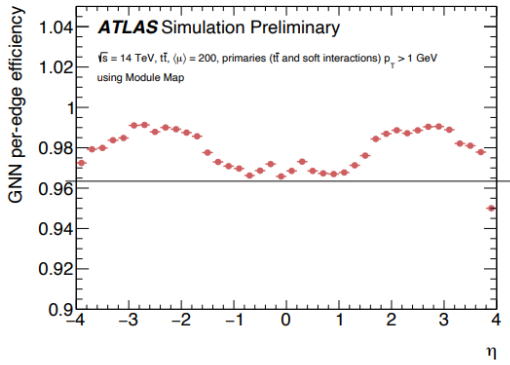
Add BatchNorm layers to make the training faster and more stable

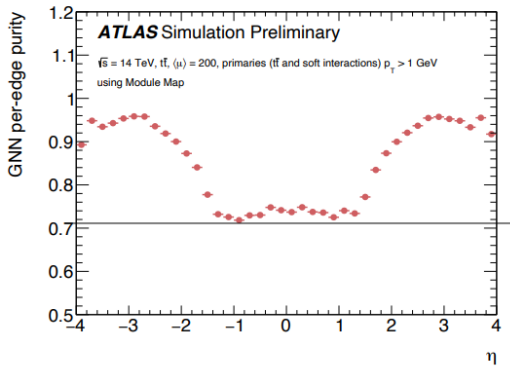
Novel fully-heterogeneous GNN designs for track reconstruction at the HL-LHC



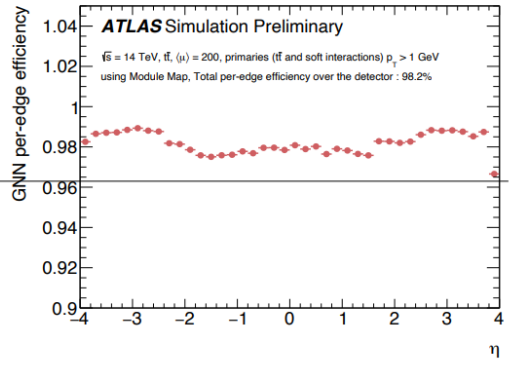
Efficiency and purity vs η

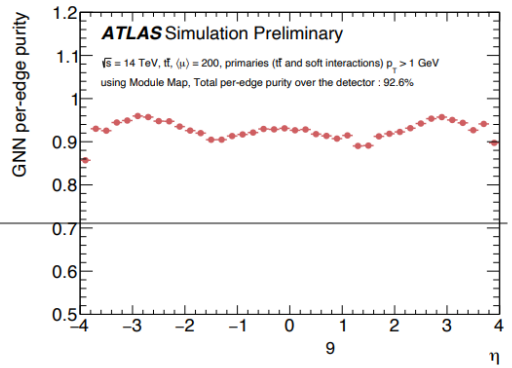
CTD2022





Heterogeneous Data + Extended GNN





$$efficiency = \frac{n_{true>treshold}}{n_{true}}$$

Similar efficiency for the same threshold (0.5)

$$purity = \frac{n_{true>treshold}}{n_{true>treshold} + n_{fake>treshold}}$$

Significant improvement of purity in the CENTRAL region

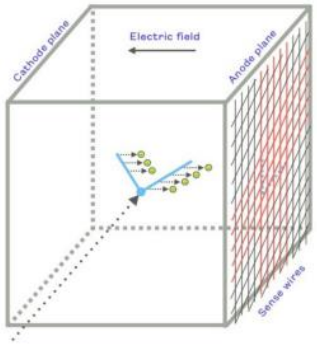
Scalable, End-to-End, Machine-Learning-Based Data Reconstruction Chain for Particle Imaging Detectors

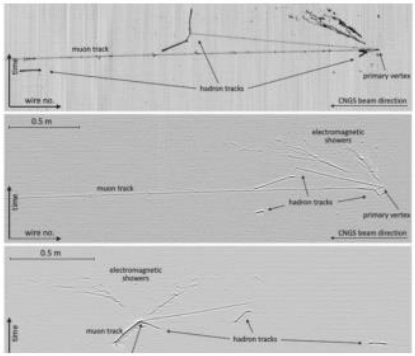
Liquid Argon Time-Projection Chambers

SLAC NATIONAL ACCELERATOR LABORATORY

The modern Particle Imaging Detector

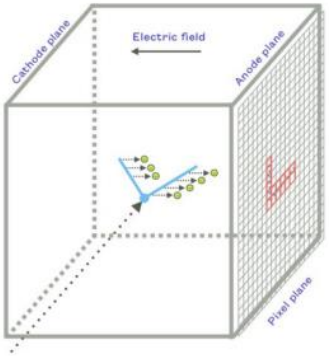
Wire TPC (2D)

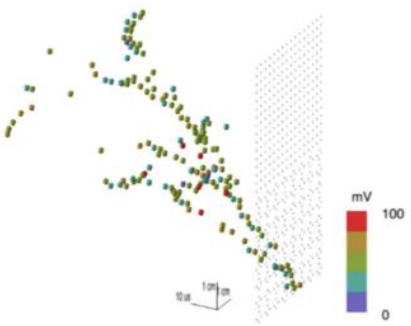




ICARUS, arXiv:1210.5089

Pixel TPC (3D)





LArPIX, arXiv:1808.02969

LArTPC are at the center stage of **beam ν physics** in the US

Short Baseline Neutrino program

- μBooNE, **ICARUS**, SBND

DUNE long-baseline experiment

- **Wire**: DUNE FD
- **Pixel**: DUNE ND-LAr

Advantages:

- **Detailed:** O(1) mm resolution, precise calorimetry
- **Scalable**: Up to tens of kt

Scalable, End-to-End, Machine-Learning-Based Data Reconstruction Chain for Particle Imaging Detectors

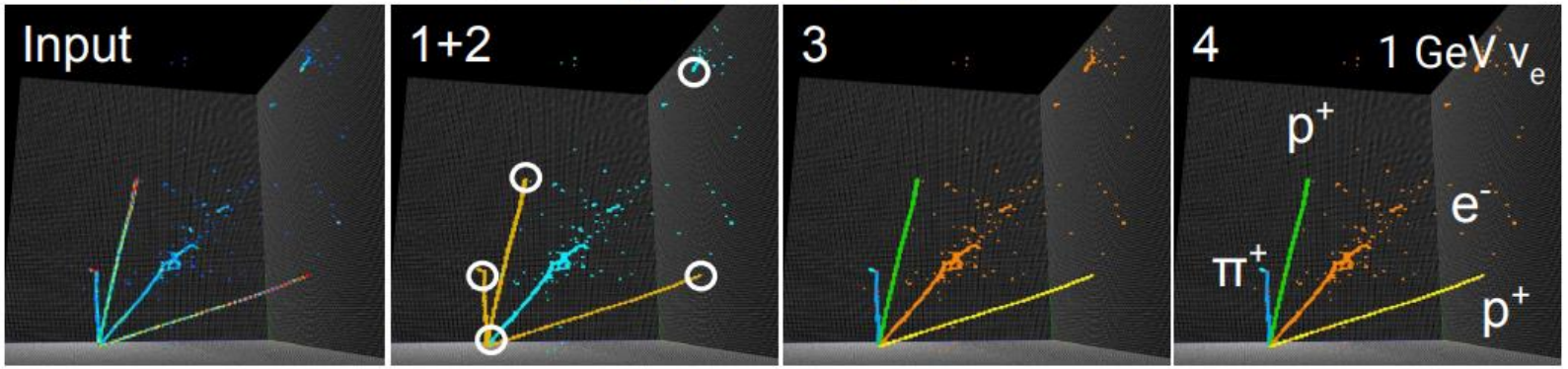
Reconstruction in LArTPCs

SLAC NATIONAL ACCELERATOR LABORATORY

Hierarchical feature extraction

What is relevant to pattern recognition in a detailed interaction image?

1. Separate topologically distinguishable **types of activity** → **Pixel-level**
2. Identify **important points** (vertex, start points, end points)
3. Cluster individual **particles** (tracks and full showers) → **Cluster-level**
4. Cluster **interactions**, identify **particle properties** in context



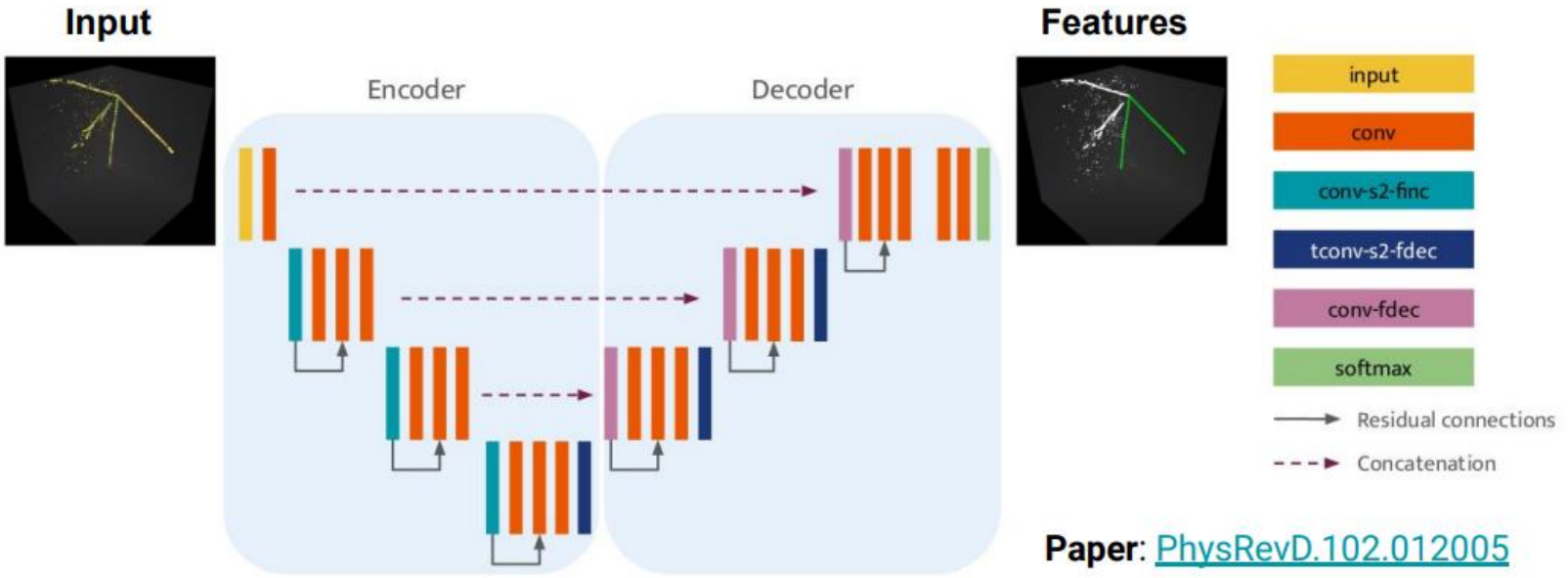
Scalable, End-to-End, Machine-Learning-Based Data Reconstruction Chain for Particle Imaging Detectors

Pixel-Level Feature Extraction

SLAC NATIONAL ACCELERATOR LABORATORY

Backbone

UResNet (UNet + ResNet + Sparse Conv.) as the **backbone feature extractor**



Paper: PhysRevD.102.012005

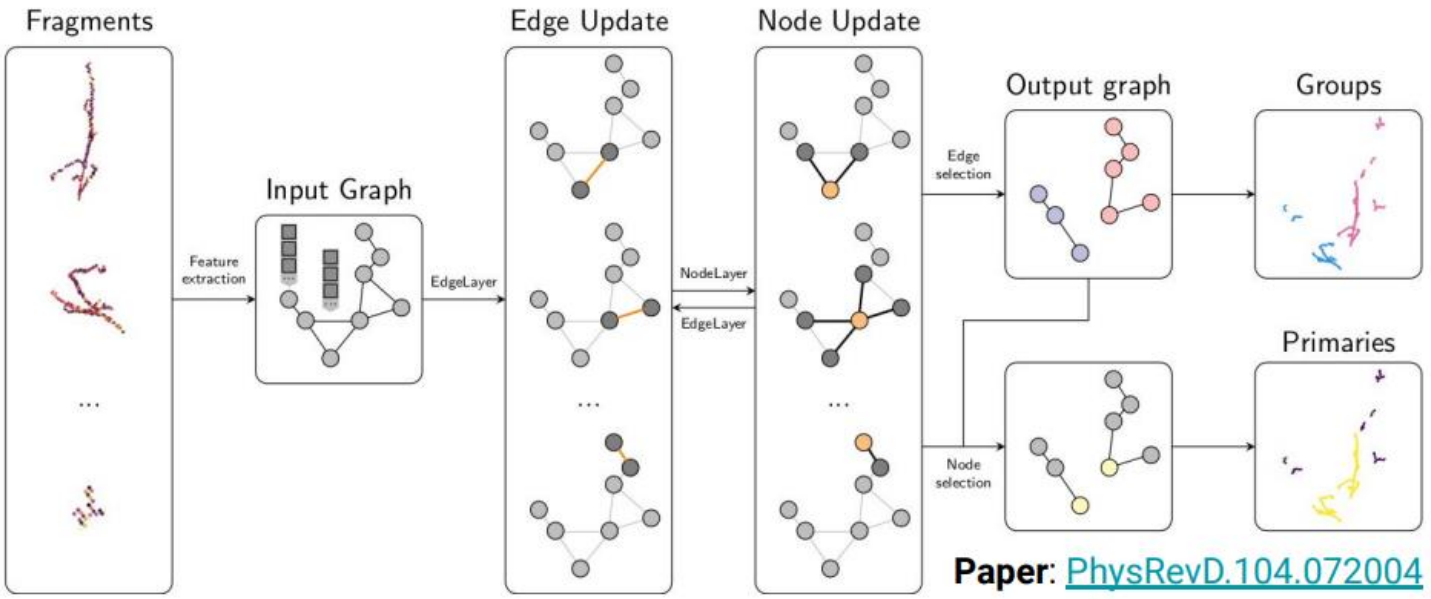
Scalable, End-to-End, Machine-Learning-Based Data Reconstruction Chain for Particle Imaging Detectors

Particle-Level Aggregation

SLAC NATIONAL ACCELERATOR LABORATORY

Graph Particle Aggregator (GrapPA)

Graph Neural Network: fragments/particles (**nodes**), correlations (**edges**)



Paper: PhysRevD.104.072004

Scalable, End-to-End, Machine-Learning-Based Data Reconstruction Chain for Particle Imaging Detectors

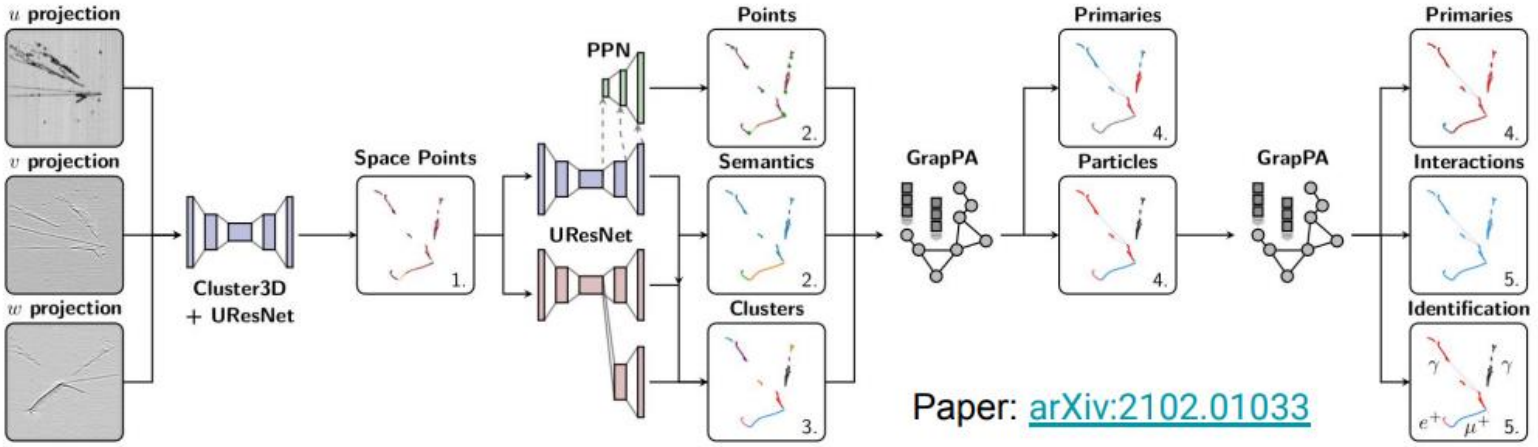
Conclusions

SLAC NATIONAL ACCELERATOR LABORATORY

Takeaways

End-to-end ML-based reconstruction chain mature and functional

- **UResNet** for pixel feature extraction, **GrapPA** for superstructure formation
- Used on **ICARUS** sim./data and **DUNE-ND** (high neutrino pileup) sim. **today**
- Check out this ICARUS **interactive reconstructed event** !



其它

- 异常检测：基于 VAE 网络实现对 SNS 加速器相关模块异常的提前预测，减少加速器维修时间
- 数据压缩：为了解决未来 HL-LHC 实验存储空间不足的问题，研究了基于 Auto-Encoder 的数据压缩方法，结果 promising
- 模型部署：ROOT/SOFIE 可以根据用户提供的网络模型（onnx, Pytorch, Keras）产生相应的 C++ 代码，将模型部署在 C++ 环境中，实现高性能的推理（性能和ONNXRuntime 相当）
- JetNet: 提供公共数据集、数据分析工具、模型性能评价指标，让用户将精力集中在模型的构建和训练

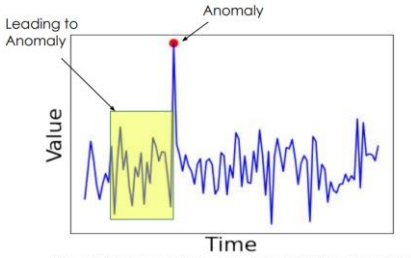
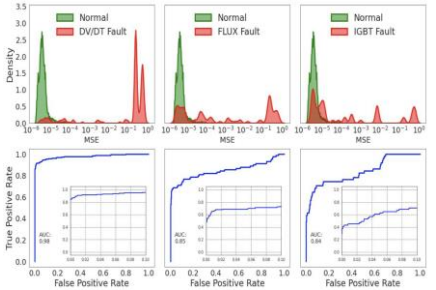
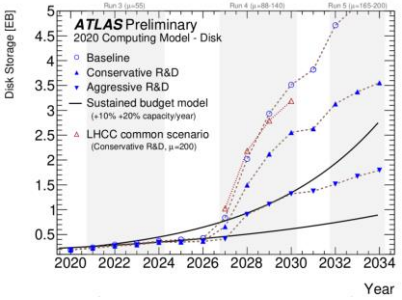
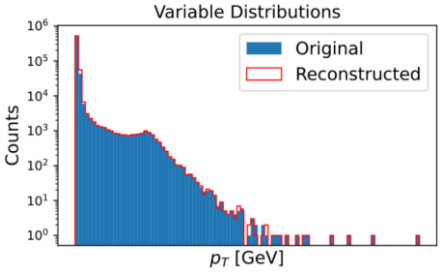
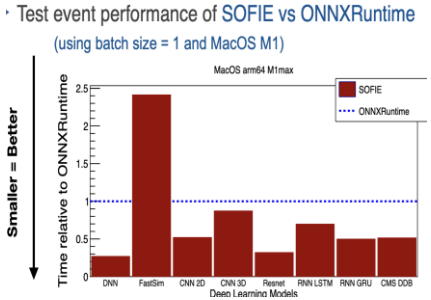


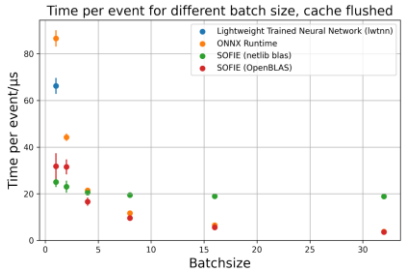
Figure 2: Toy example shows a visual representation of anomaly (red dot)) and data leading to anomaly (yellow shaded area).

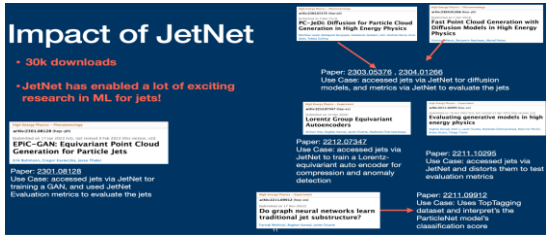












backup

Simulation

- Calorimeter Fast simulation:
  - FastCaloGAN: a fast simulation of the ATLAS Calorimeter with GANs
  - gaede_chep23_caloml_v01 (jlab.org)
  - Generating Accurate Showers in Highly Granular Calorimeters Using Normalizing Flows
  - Fast and Accurate Calorimeter Simulation with Diffusion Models
  - Transformers for Generalized Fast Shower Simulation
- Ultra-fast simulation
  - THE LHCB ULTRA-FAST SIMULATION OPTION, LAMARR
  - Flashsim: an ML simulation framework
- Refining fast simulation using machine learning
- Hadronic Simulation with conditional Masked Autoregressive Flow

Reconstruction

- PID:
  - Particle identification with machine learning in ALICE Run 3
  - Pion/Kaon Identification at STCF DTOF Based on Classical/Quantum Convolutional Neural Network
  - Fast Inclusive Flavor Tagging at LHCb
- Vertex:
  - Advances in developing deep neural networks for finding primary vertices in proton-proton collisions at LHC
- Tracking:
  - An Object Condensation Pipeline for Charged Particle Tracking
  - End-to-End Geometric Representation Learning for Track Reconstruction
  - BESIII track reconstruction algorithm based on machine learning
  - Track Identification for CLAS12 using Artificial Intelligence
  - Novel fully-heterogeneous GNN designs for track reconstruction at the HL-LHC
  - HyperTrack: neural combinatorics for high energy physics
- Cluster:
  - Improved Clustering in the Belle II Electromagnetic Calorimeter with Graph Neural Networks
  - Development of particle flow algorithms based on Neural Network techniques for the IDEA calorimeter at future colliders
- LArTPC:
  - Neutrino interaction vertex-finding in a DUNE far-detector using Pandora deep-learning
  - Scalable, End-to-End, Machine-Learning-Based Data Reconstruction Chain for Particle Imaging Detectors
  - Graph Neural Network for 3D Reconstruction in Liquid Argon Time Projection Chambers

Online

- Trigger:
  - Applications of Lipschitz neural networks to the Run 3 LHCb trigger system
  - Development of a Deep-learning based Full Event Interpretation (DFEI) algorithm for the future LHCb trigger
- Continual Learning：
  - The Deployment of Realtime ML in Changing Environments
  - Embedded Continual Learning for HEP
- FPGA:
  - Symbolic Regression on FPGAs for Fast Machine Learning Inference
  - Machine Learning for Real-Time Processing of ATLAS Liquid Argon Calorimeter Signals with FPGAs
  - Acceleration of a CMS DNN based Algorithm

Analysis

- Efficient search for new physics using Active Learning in the ATLAS Experiment
- Using a Neural Network to Approximate the Negative Log Likelihood Distribution
- A method for inferring signal strength modifiers by conditional invertible neural networks

Anomaly Detection

- Multi-Module based VAE to predict HVCM faults in the SNS accelerator
- Resilient Variational Autencoder for Unsupervised Anomaly Detection at the SLAC Linac Cohrerent Light Source

General

- Uncertainty Aware Machine Learning Models for Particle Physics Applications
- Exploring Interpretability of Deep Neural Networks in Top Tagging
- JetNet library for machine learning in high energy physics
- New developments of TMVA/SOFIE: Code Generation and Fast Inference for Graph Neural Networks
- Machine learning based compression for scientific data