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

The application of ML in simulations

Wenxing Fang (IHEP)

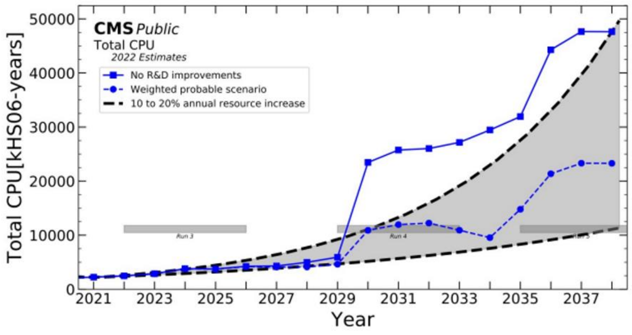
EPD seminar 2024.05.22

Outlines

- Fast simulation
- More accurate simulation
- Summary

Background of fast simulation

- The HL-LHC experiment will take a huge amount of experimental data. Significant computational resources are required for data processing, MC production, and analysis. Without R&D, there will be a shortage of computational resources
- The MC simulation takes most CPU resources. Implementing fast MC simulation is important
  - Traditional method: shower parameterization, frozen shower, Delphes, …
  - ML based: fast calorimeter simulation, Ultra-Fast Simulation (without Geant4), …



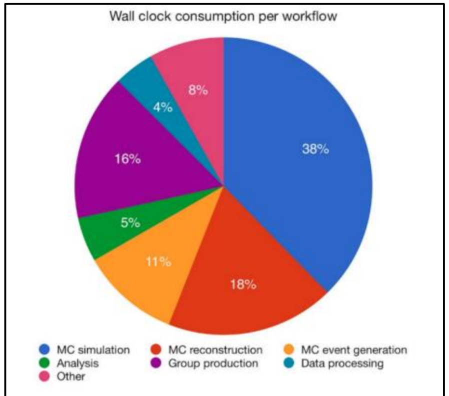
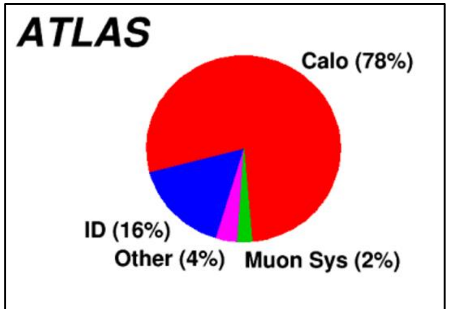


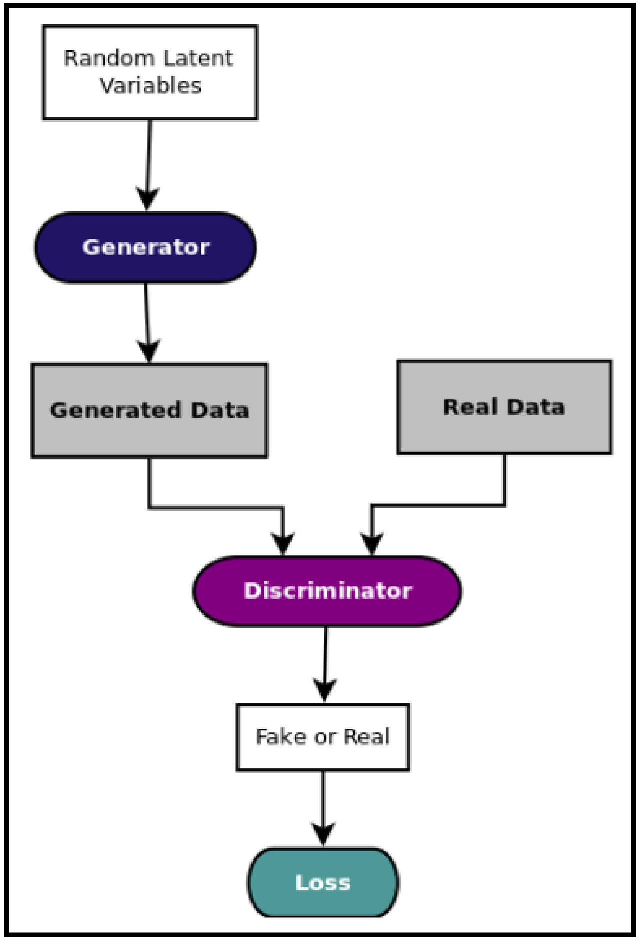
Figure 1: ATLAS CPU hours used by various activities in 2018



Subdetector CPU fraction for 50 ttbar events
MC16 Candidate Release

Generative Adversarial Networks (GAN)

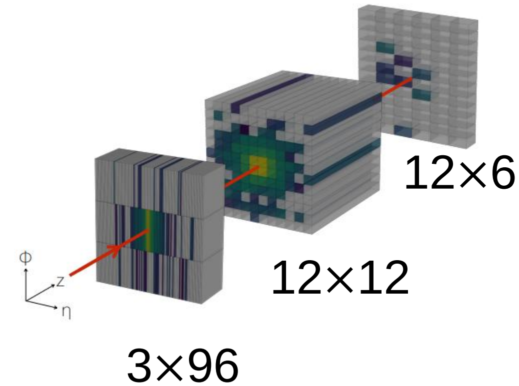
- Discriminator tries to discriminate the real data and generated data
- The generator tries to produce generated data which can confuse the discriminator
- At the end of training, the discriminator can not discriminate the real or generated data. The generator learns the true underlying data distribution

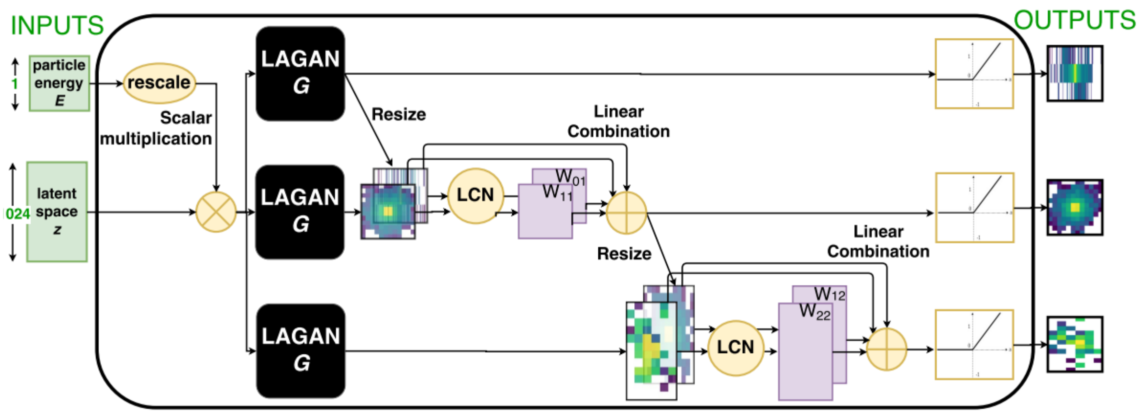


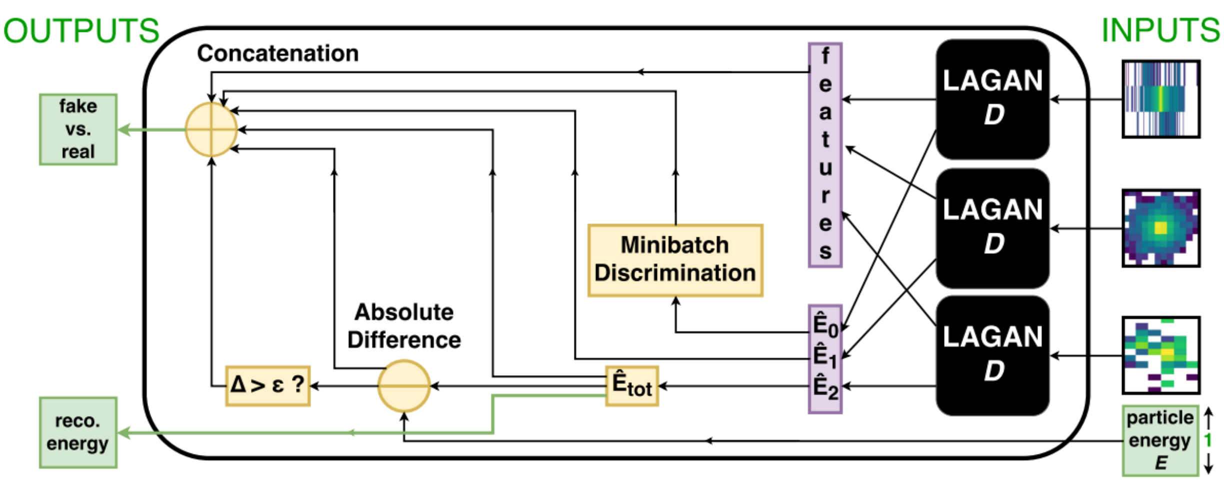
$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[log(1 - D(G(z)))]$$

CaloGAN

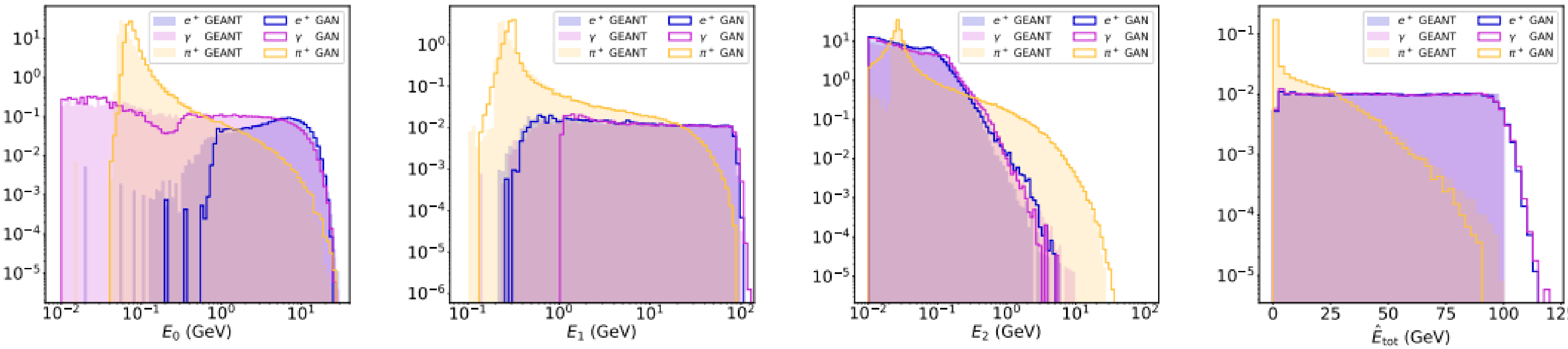
- The CaloGAN (2017) achieved a fast calorimeter simulation based on GAN







CaloGAN performance



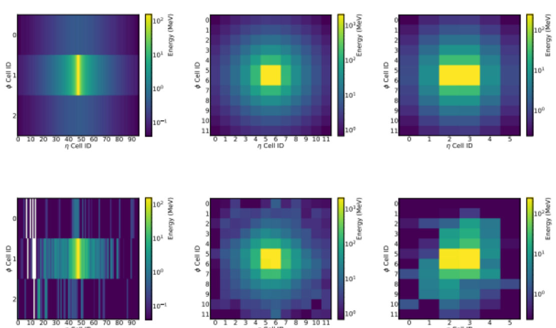


FIG. 8: Average π^+ GEANT4 shower (top), and average π^+ CALOGAN shower (bottom), with progressive calorimeter depth (left to right).

Simulator	**Hardware**	**Batch Size**	**ms/shower**
GEANT4	CPU	N/A	1772
CALOGAN	CPU	1	13.1
		10	5.11
		128	2.19
		1024	2.03
	GPU	1	14.5
		4	3.68
		128	0.021
		512	0.014
		1024	0.012

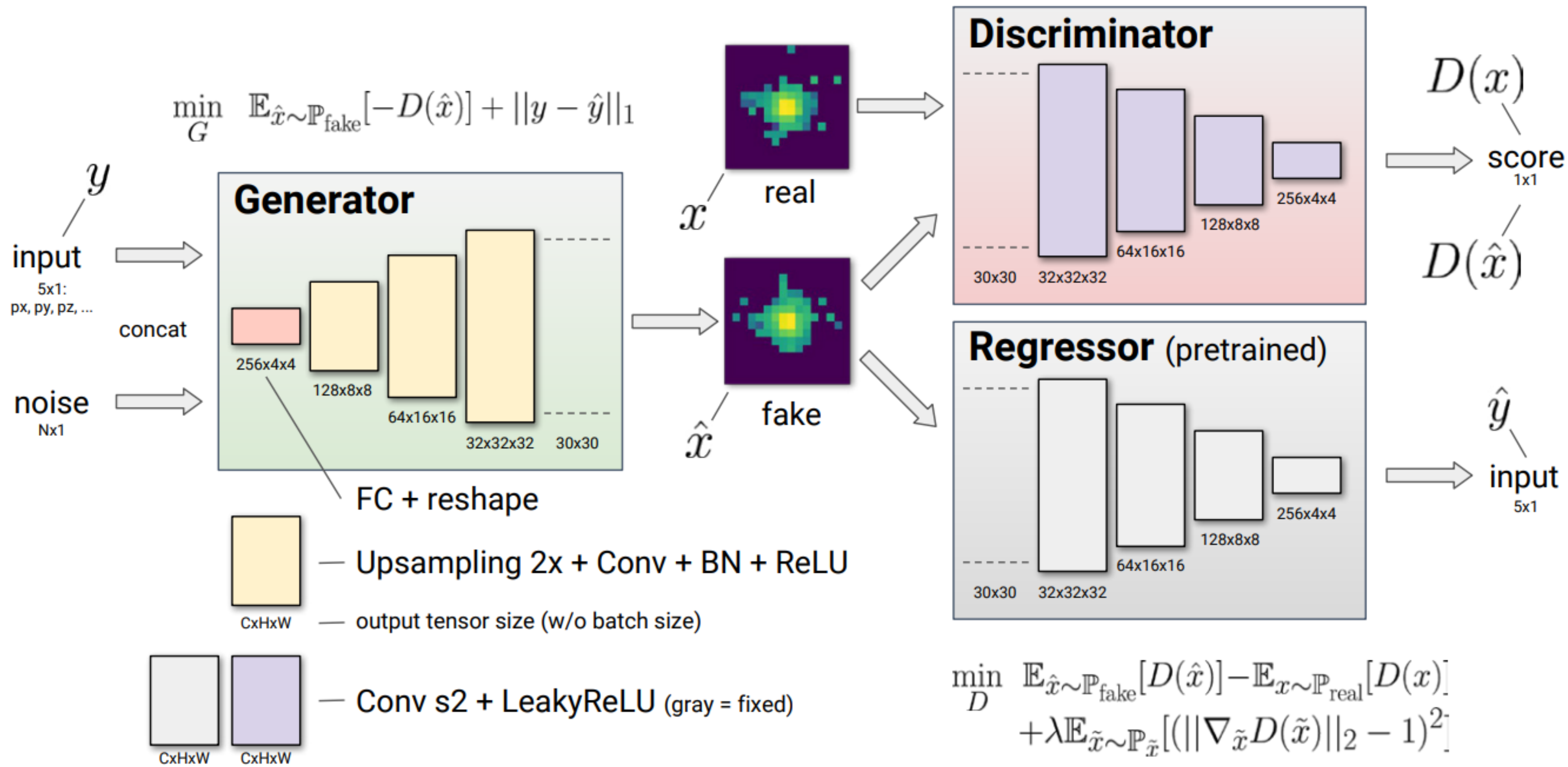
e^+ vs. π^+

		Test on	
		GEANT4	CALOGAN
Train on	GEANT4	99.6% ± 0.1%	96.5% ± 1.1%
	CALOGAN	98.2% ± 0.9%	99.9% ± 0.2%

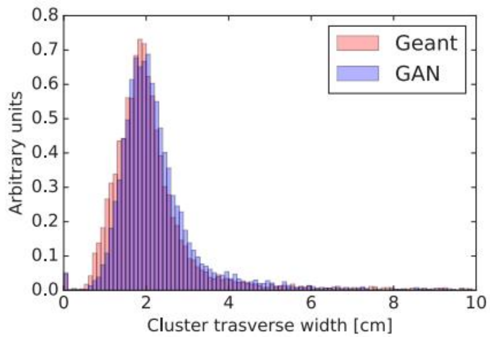
e^+ vs. γ

		Test on	
		GEANT4	CALOGAN
Train on	GEANT4	66.1% ± 1.2%	70.6% ± 2.6%
	CALOGAN	54.3% ± 0.8%	100.0% ± 0.0%

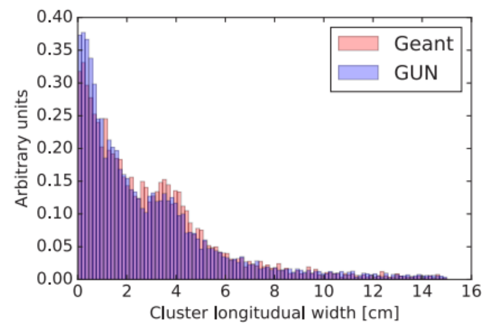
The LHCb case



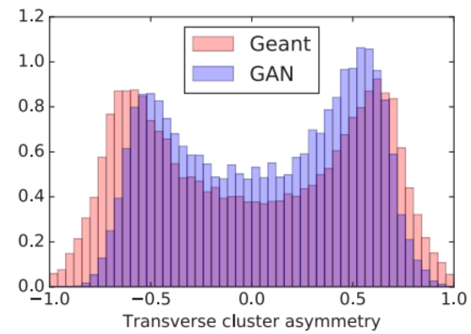
The LHCb case (performance)



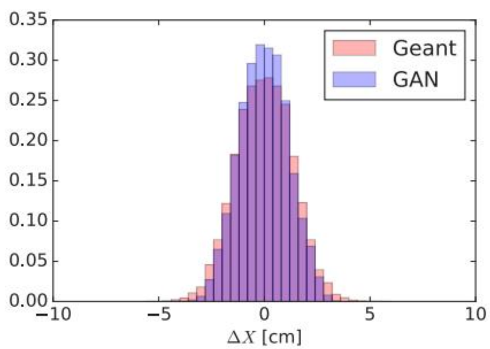
(a) The transverse width of real and generated clusters



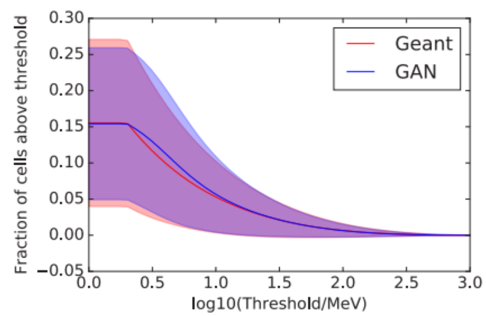
(b) The longitudinal width of real and generated clusters



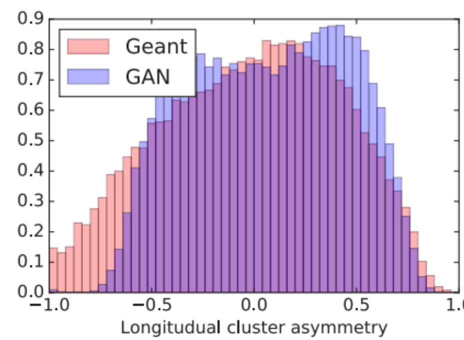
(e) The transverse asymmetry of real and generated clusters



(c) ΔX between cluster center of mass and the true particle coordinate



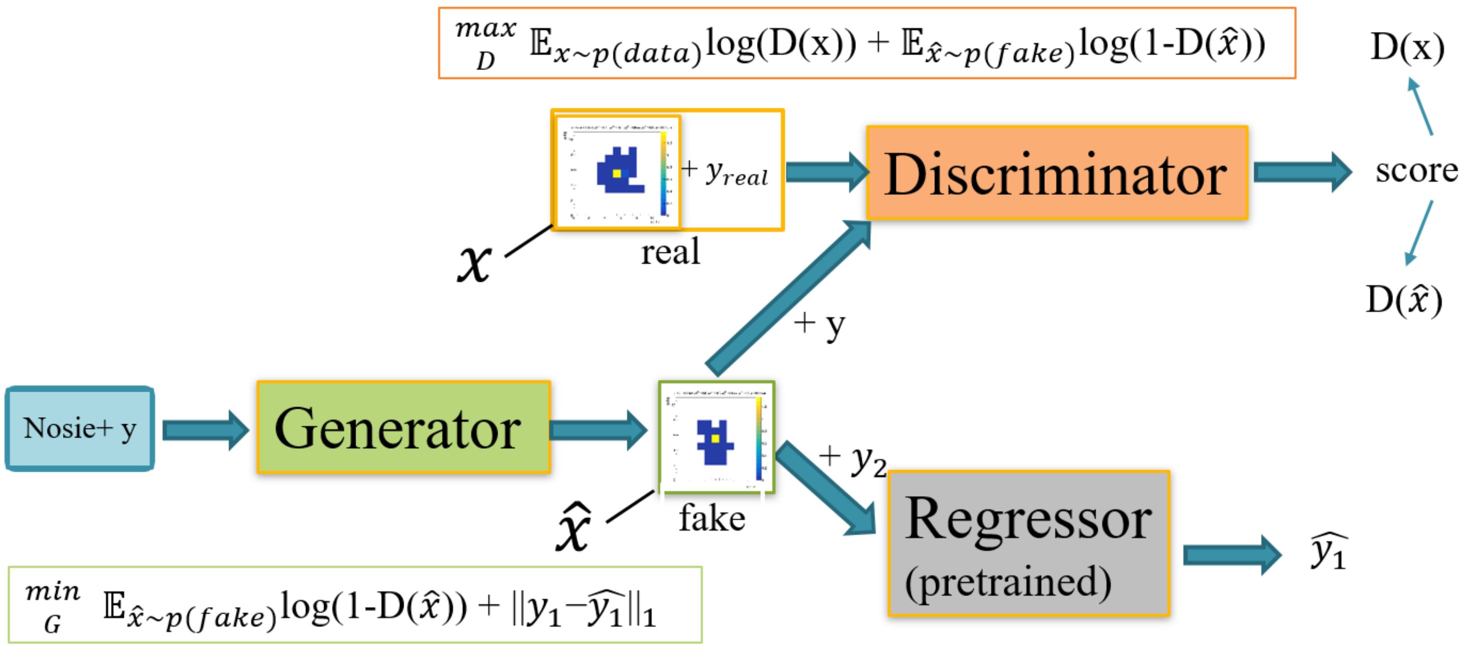
(d) The sparsity of real and generated clusters



(f) The longitudinal asymmetry of real and generated clusters

The BESIII case

- Reference from the LHCb one

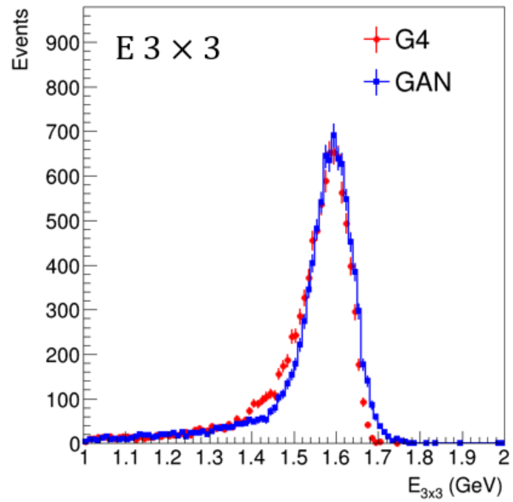


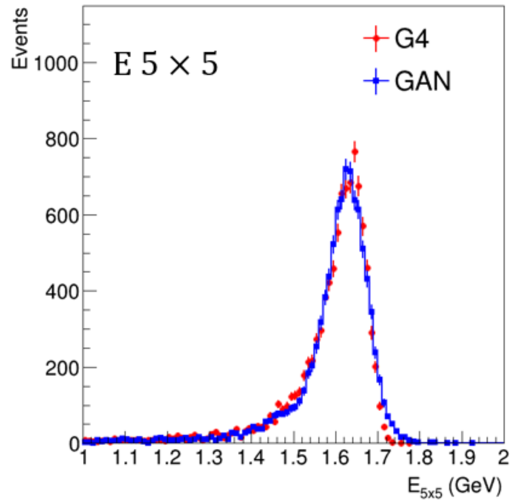
- The y ($y_1 + y_2$) contains the momentum of particle and the relative position and angular between the particle and the calorimeter.
  - y_1
    - Momentum: the momentum of the particle.
    - $\Delta\phi^{Mom}$: the ϕ difference between the momentum of incoming particle and the direction of the crystal.
    - $\Delta\theta^{Mom}$: the θ difference between the momentum of incoming particle and the direction of the crystal.
  - y_2
    - ΔZ^{Pos}: the Z difference between the hit point of incoming particle and the z of front center of the crystal.
    - $\Delta\phi^{Pos}$: the ϕ difference between the hit point of incoming particle and the ϕ of front center of the crystal.
    - Z: the Z value of hit point

The BESIII case (performance)

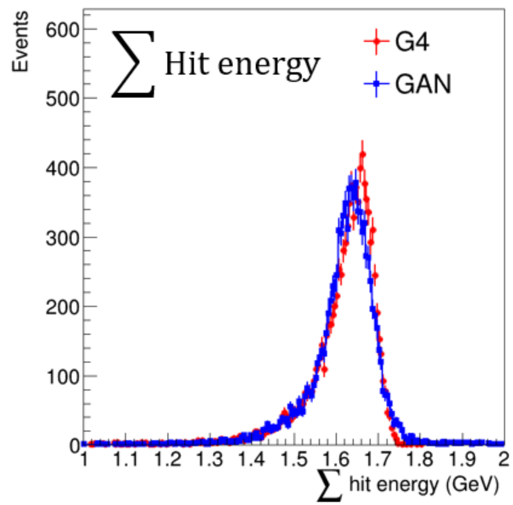
Dataset:

- $e^{\pm}$ showers in ECAL Barrel
- 11x11 voxels

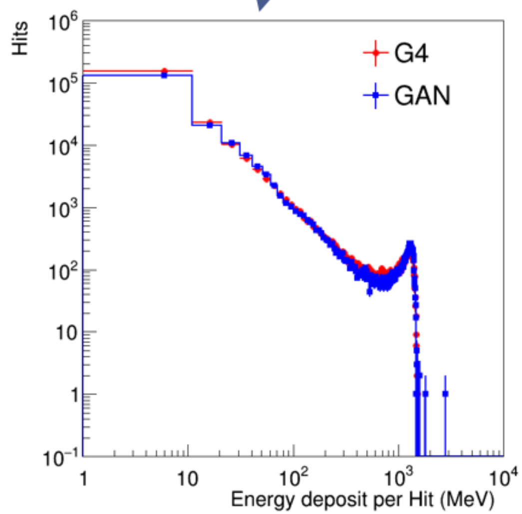




e^-

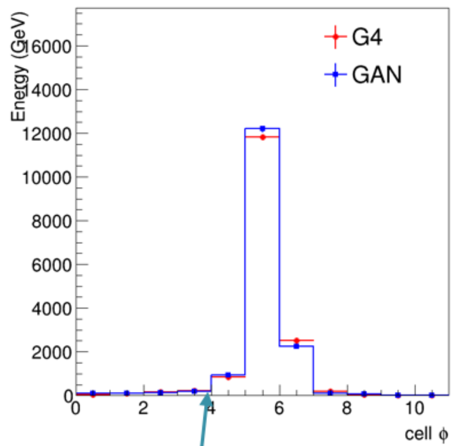


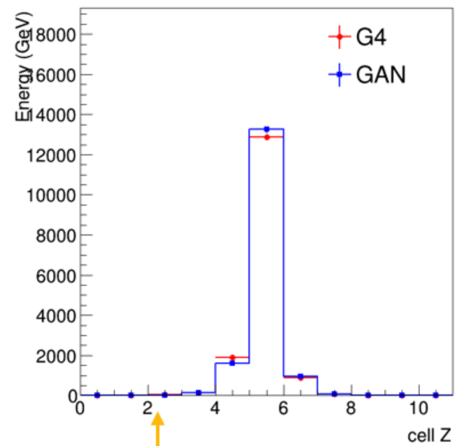
Energy of calorimeter cells



The BESIII case (performance)

e^{-}



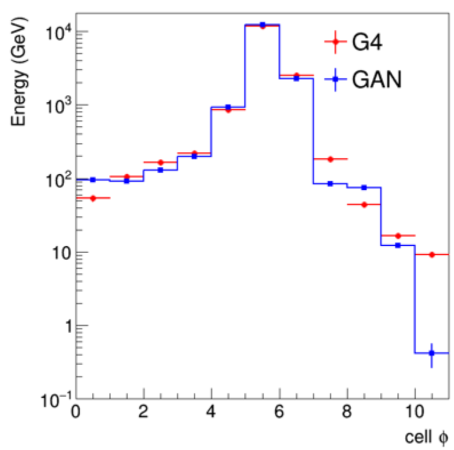


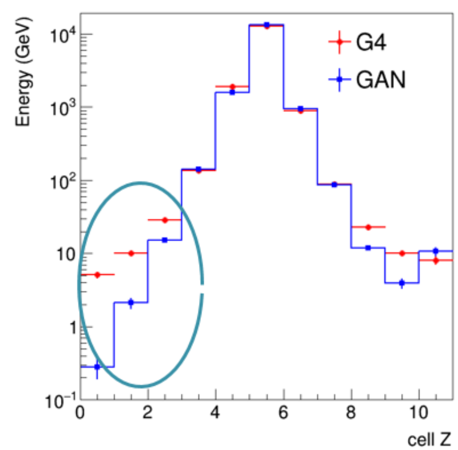
Energy deposited in ϕ direction

log scale

Energy deposited in Z direction

log scale

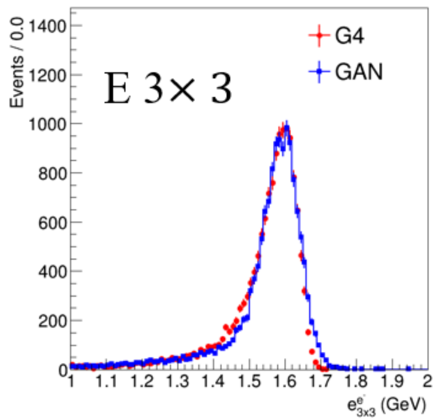


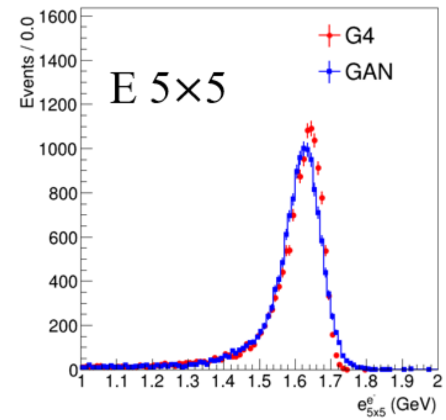


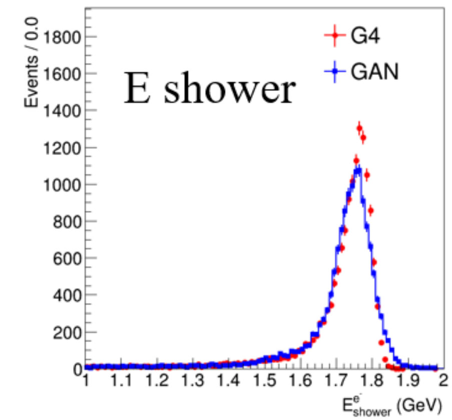
The BESIII case (performance)

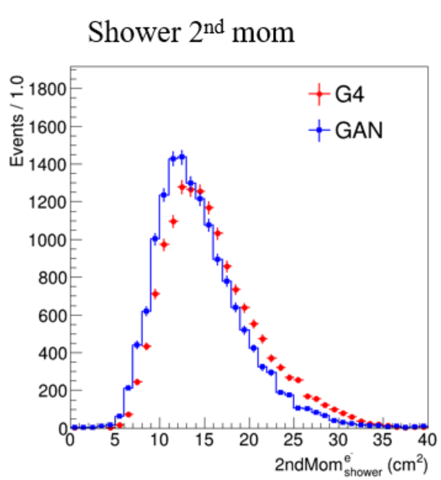
e^{-}

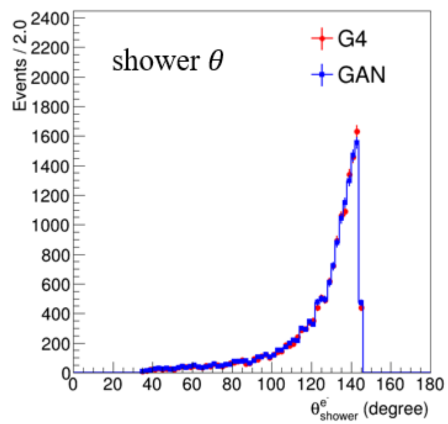
- Apply the GAN simulation in BESIII offline software

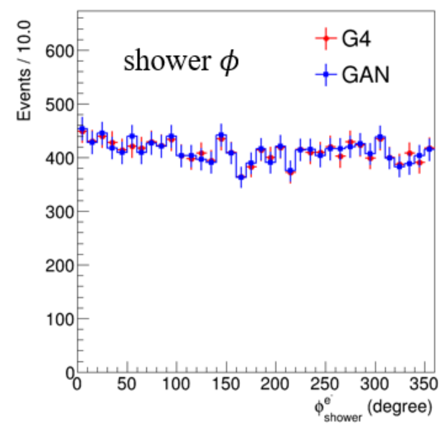






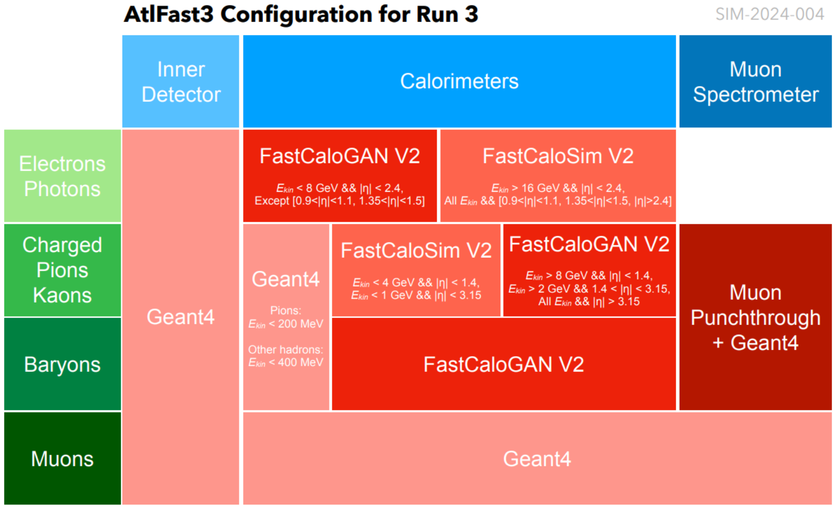


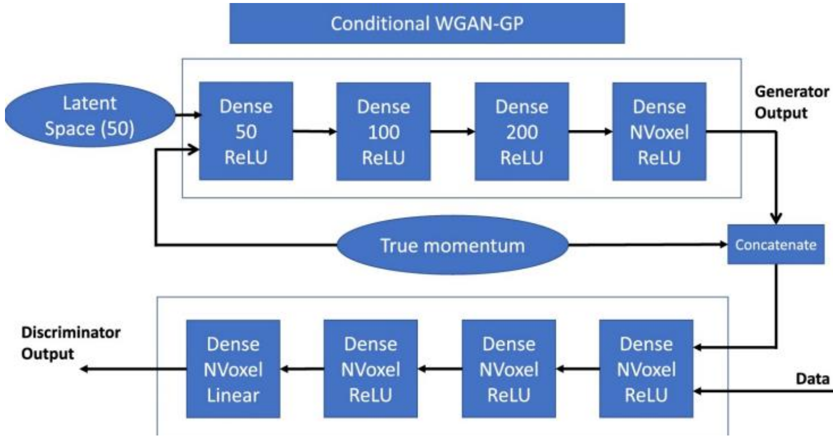




The ATLAS case

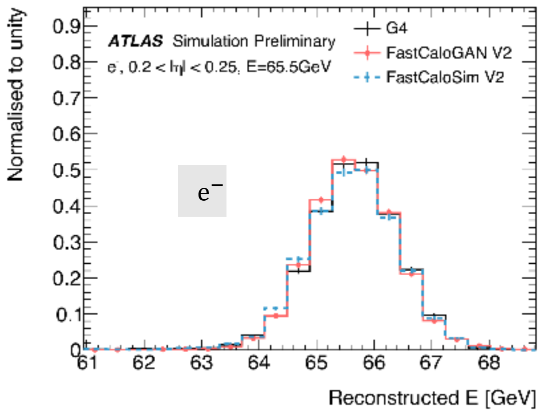
- AltFast3 (a detector response fast simulation system):
  - FastCaloGAN V2 (ML-based)
  - FastCaloSim V2 (parametrization-based)
  - Geant4 (limited to specific cases)
- FastCaloGAN:
  - Simulating calorimeter showers for particles between 256 MeV and 4 TeV over full detector acceptance (protons only at -0.25 ≤ η ≤ 0.25)
  - WGANs trained on each of the 100 bins in | η | and conditioned on truth momentum
  - Total 300 GANs

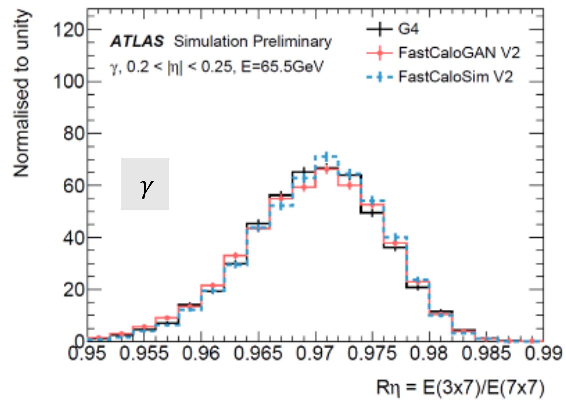


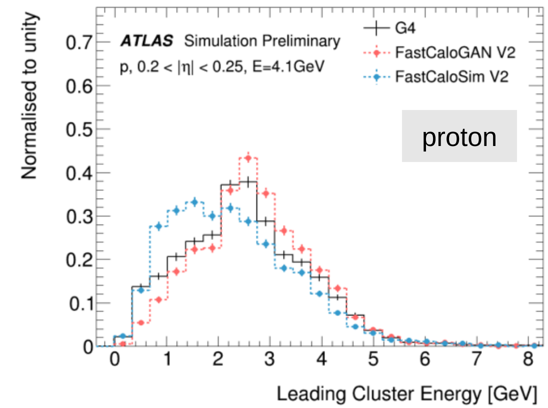


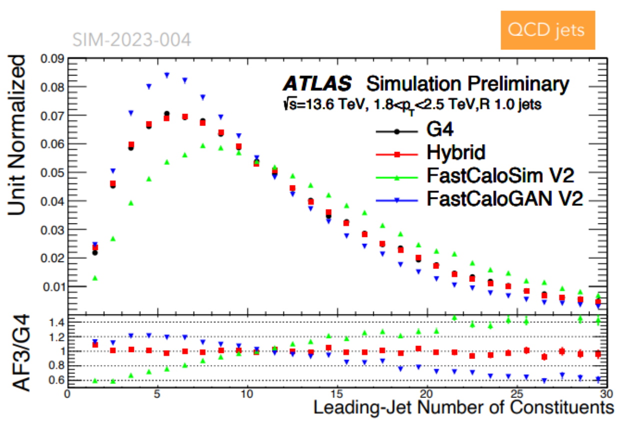
ACAT2024

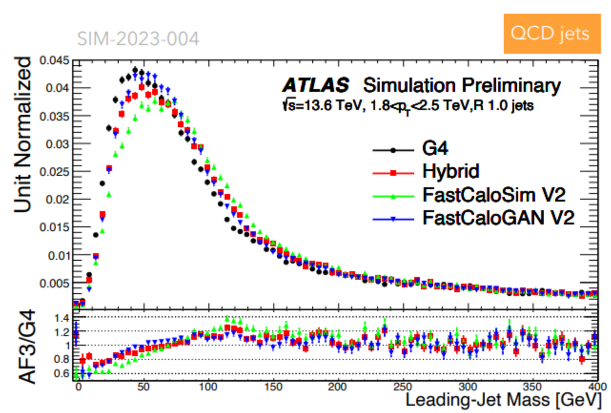
The ATLAS case (performance)

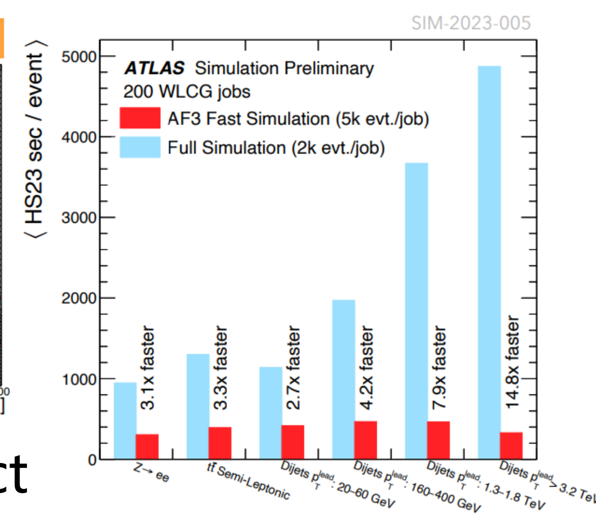








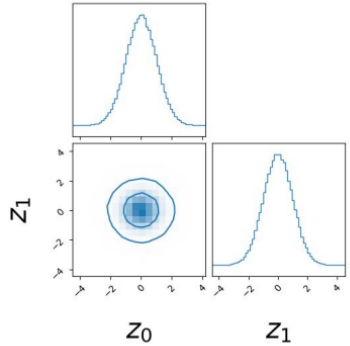




- 3 – 15 speed-up in simulation time with respect to Geant4, depending on the physics process
- Simulation time in AtlFast3 completely dominated by full simulation of the Inner Detector

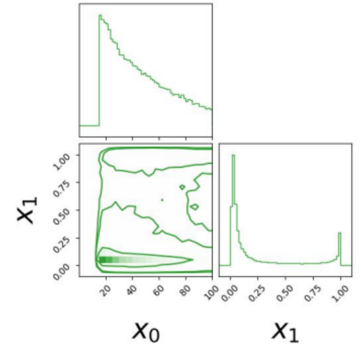
ACAT2024

Normalizing Flows



reversible

$$\longleftarrow \boxed{f^{-1}(\mathbf{x})} \longleftarrow$$

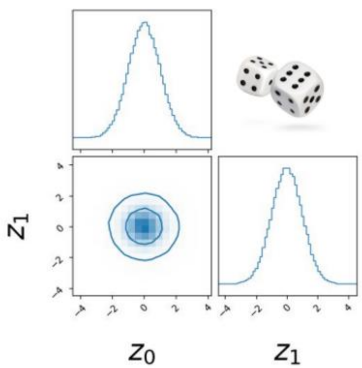


$$\mathbf{x} = f(\mathbf{z})$$

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

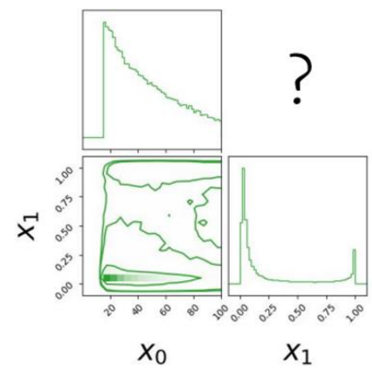
$$\log(p_x(x)) = \boxed{\log(p_z(f^{-1}(\mathbf{x})))} + \boxed{\log\left(\det \mathbb{J}_{f^{-1}}(\mathbf{x})\right)}$$

What we know



$$\longrightarrow \boxed{f(\mathbf{z})} \longrightarrow$$

What we need



$\mathbf{Z}$

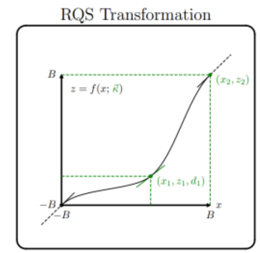
multi-dimensional gaussian

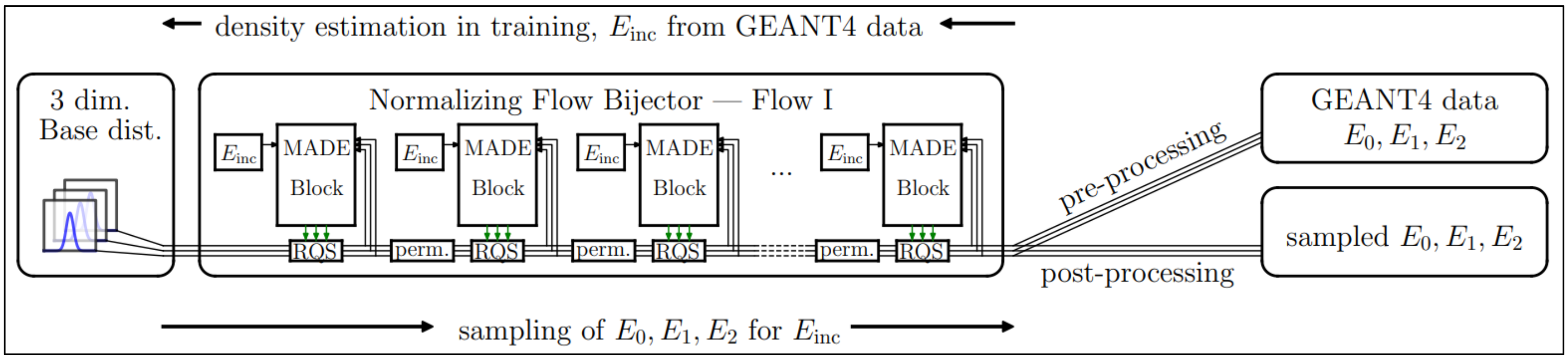
$\mathbf{X}$

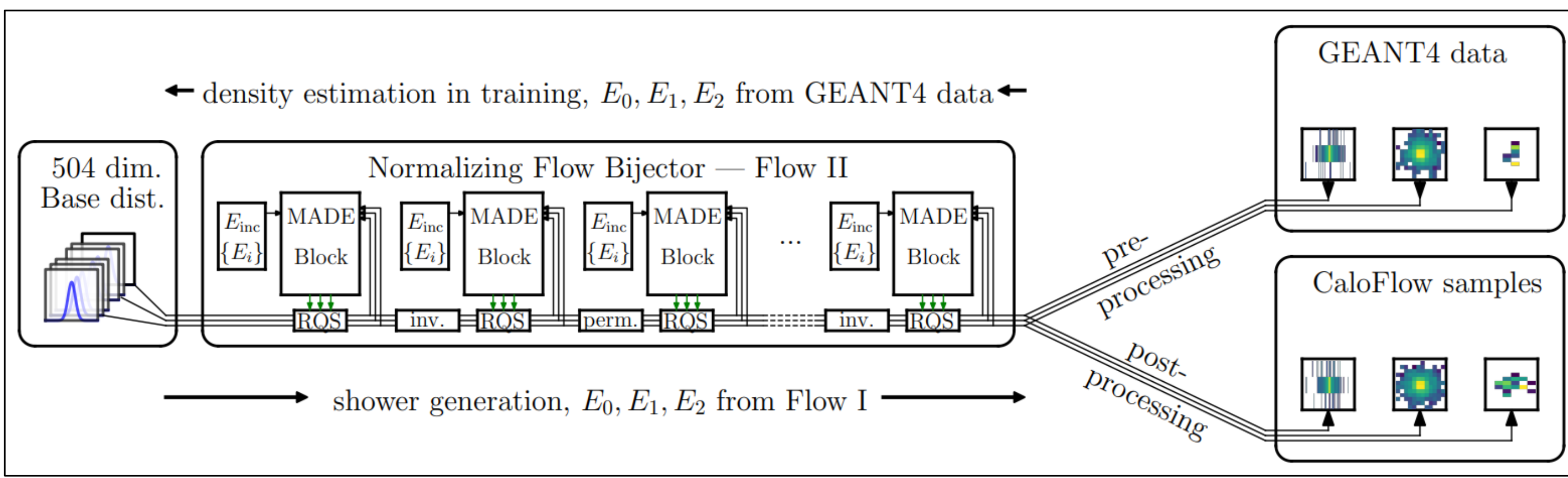
FullSim data, pdf unknown!

CaloFlow

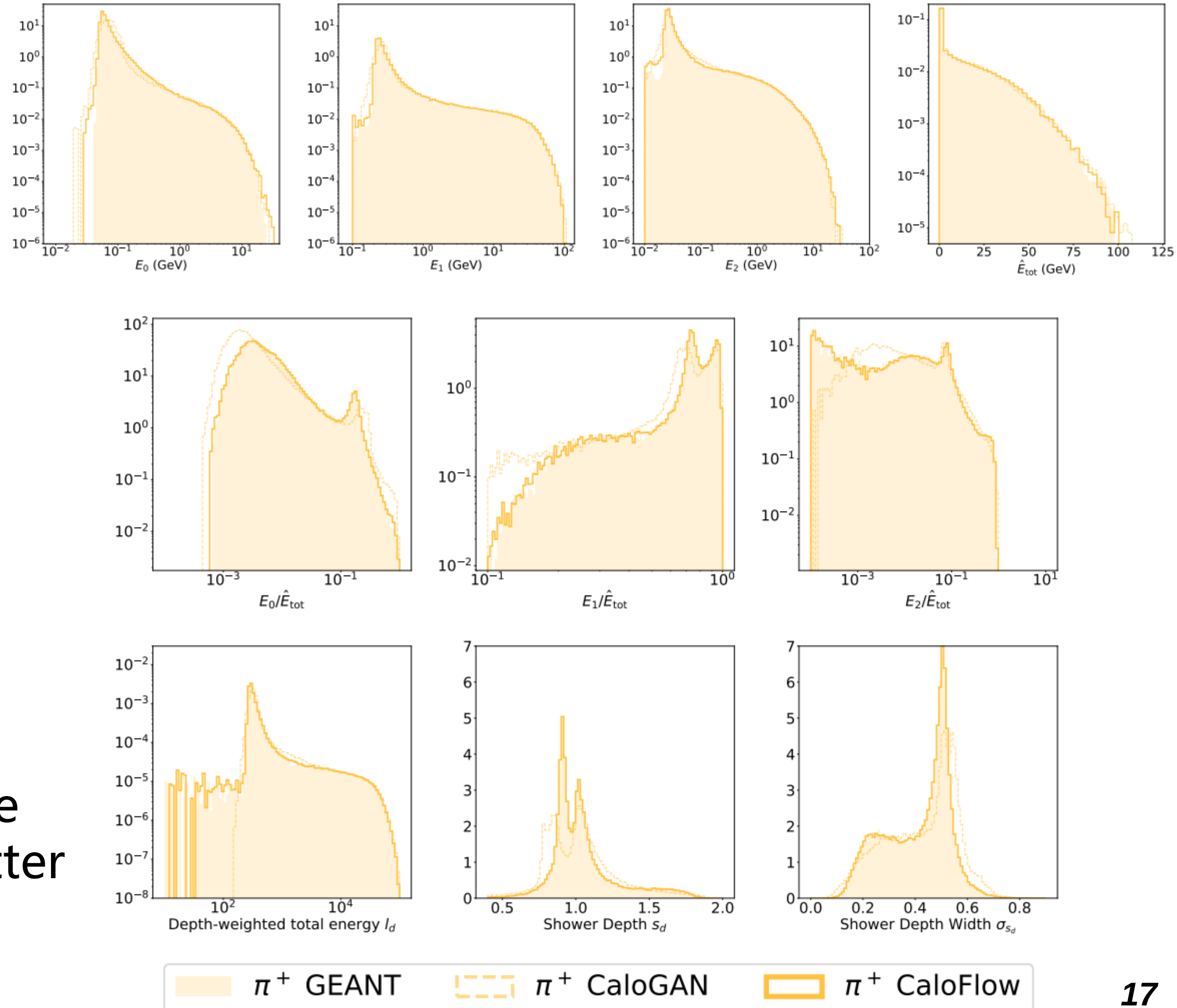
- The CaloFlow (2021) uses the same dataset as CaloGAN and shows much better physics performance







CaloFlow (performance)

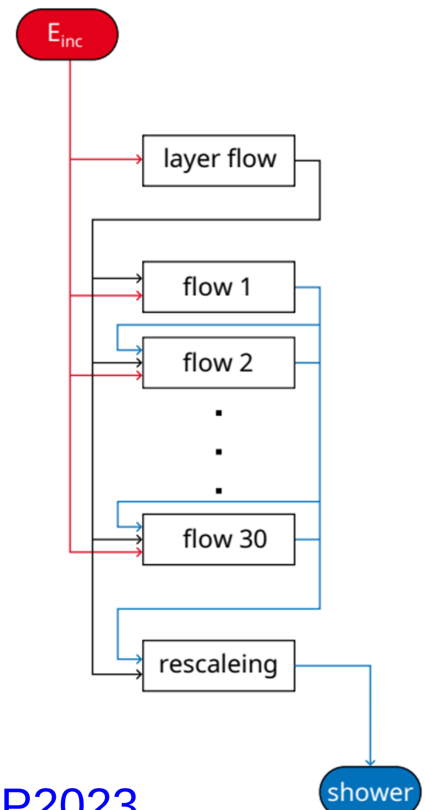


- The performance seems much better than CaloGAN

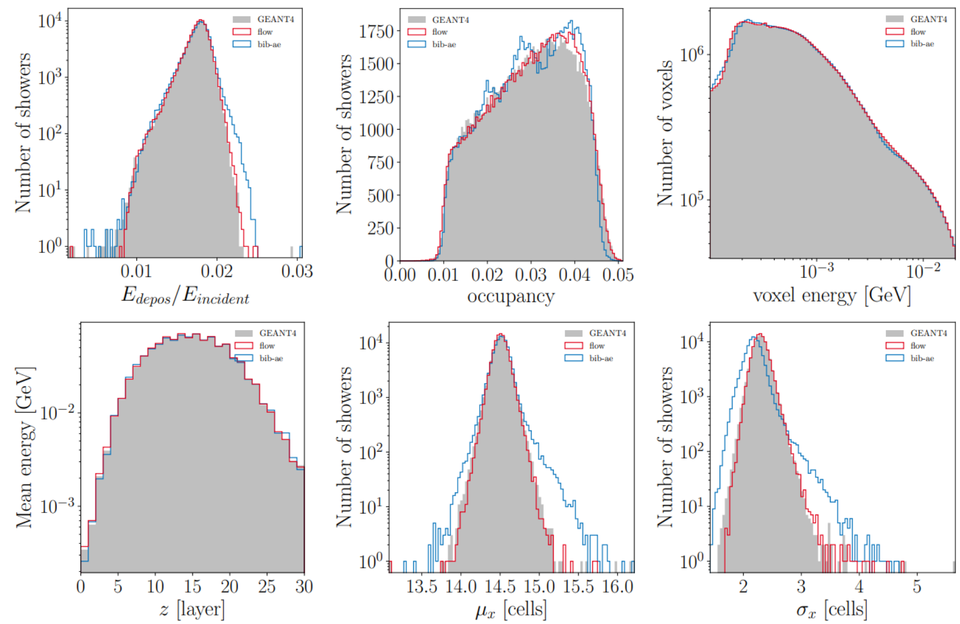
The ILC case

- Dataset:
  - photon showers in ECAL
  - 30x30x30 voxels

Architecture



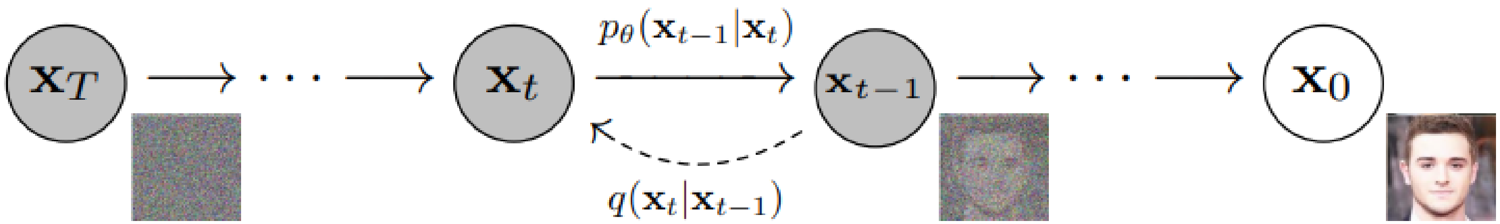
Preliminary Results



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

Diffusion model

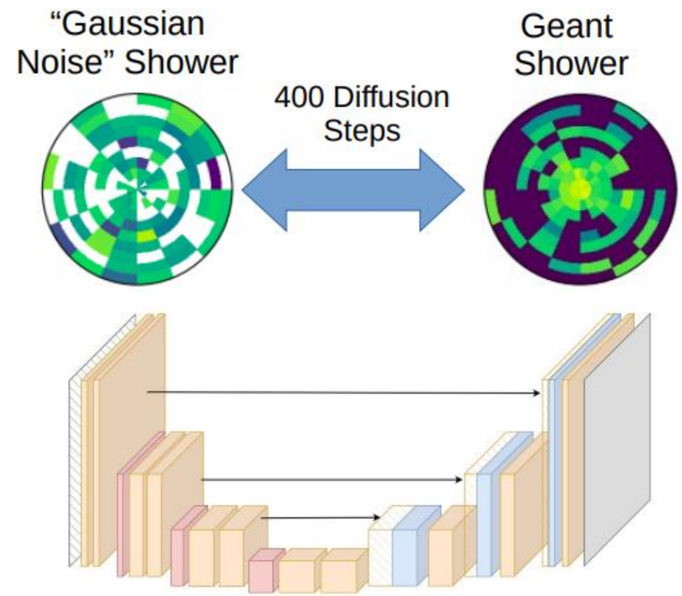
- The diffusion model is proposed in 2020

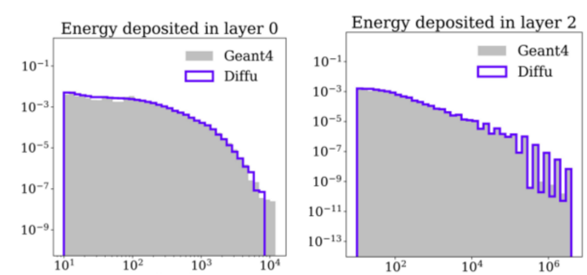


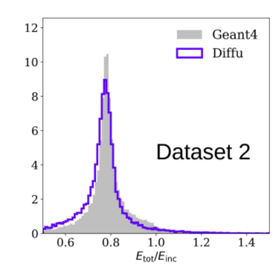
- Diffusion process: $x_0 \rightarrow x_T$
  - Adding noise step by step, making $x_T \sim \mathcal{N}(0, \mathbf{I})$
- Train a model to invert the diffusion process
- When do simulation, start from $\mathcal{N}(0, \mathbf{I})$ and denoise it step by step using the trained model

CaloDiffusion

- CaloDiffusion (a fast calorimeter simulation method based on diffusion model)
- Dataset:
  - ATLAS-like geometry, 5 layer cylinder with irregular binning, 368 voxels
- Denoise model:
  - U-net architecture with 3D convolutions
  - Input: Noisy shower
  - Condition inputs: incident particle energy, diffusion step
  - Output: noise
- Good agreement with Geant4, some properties (e.g. total shower energy), can still be improved
- Generation time is slower than other ML approaches (still faster than Geant4)

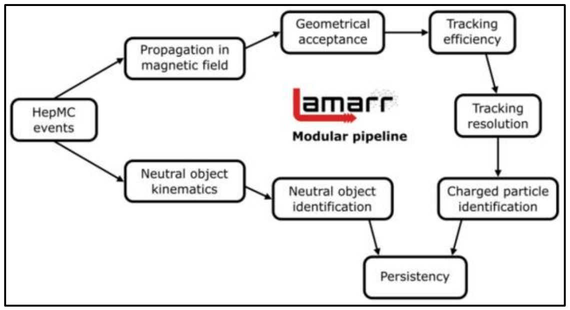


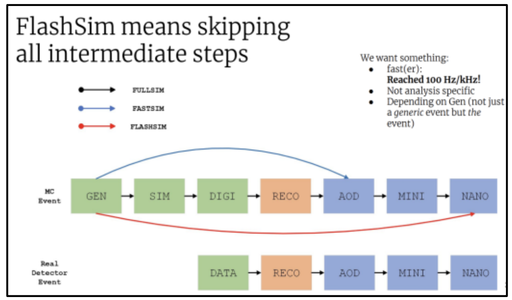


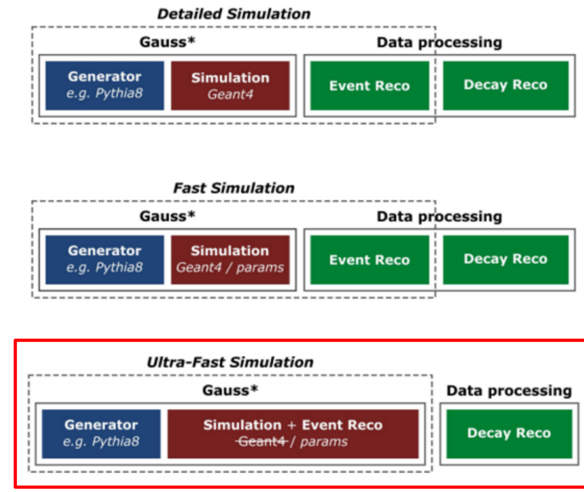


Ultra-Fast Simulation

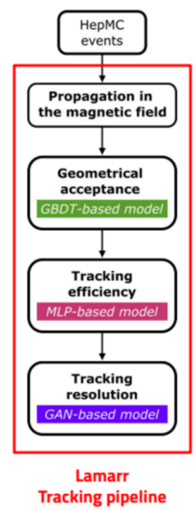
- Without Geant4 simulation, from MC particle to physics analysis object simulation
  - Such as LHCb Lamarr, CMS FlashSim

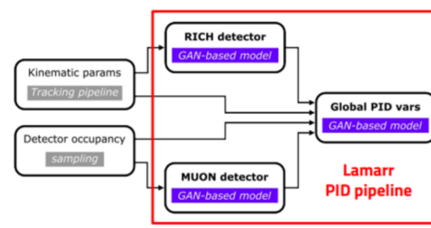


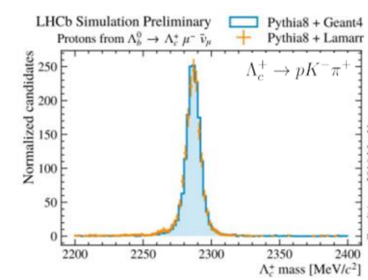




- Most parts are ML-based:
  - GBDT for acceptance
  - MLP for tracking efficiency
  - GAN for tracking resolution and PID



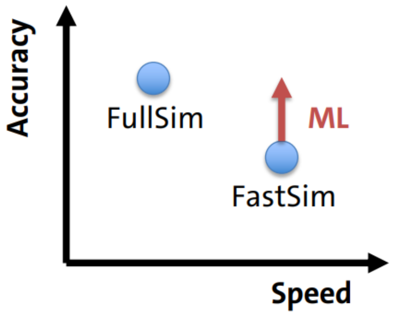




- Simulating ECAL with an ultra-fast approach requires to face the particle-to-particle correlation problem:
  - Sequence of N generated photons → sequence of N′ reconstructed clusters
  - Approached as a language translation problem

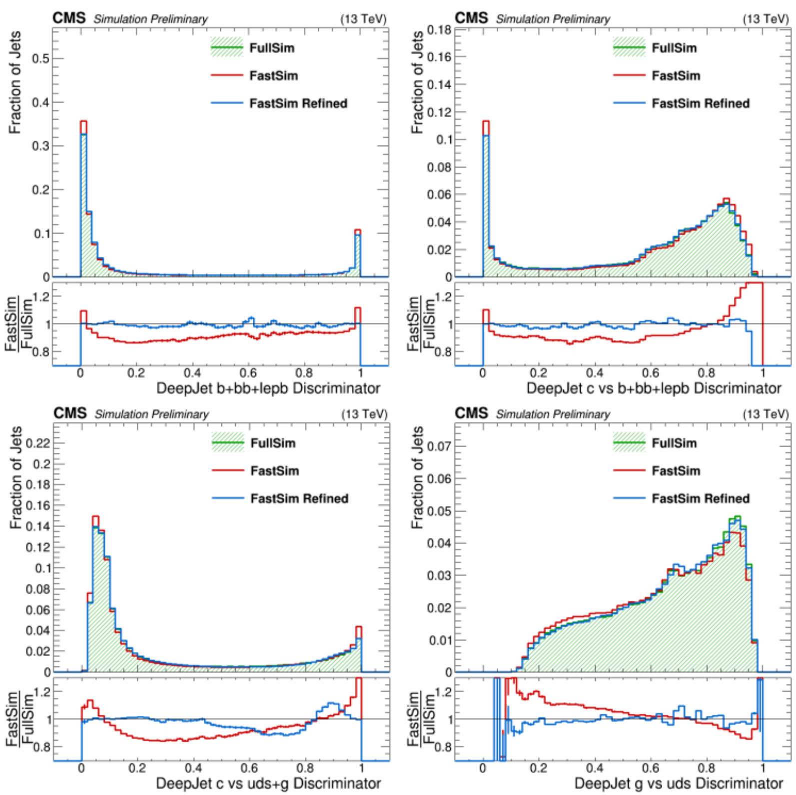
More accurate simulation

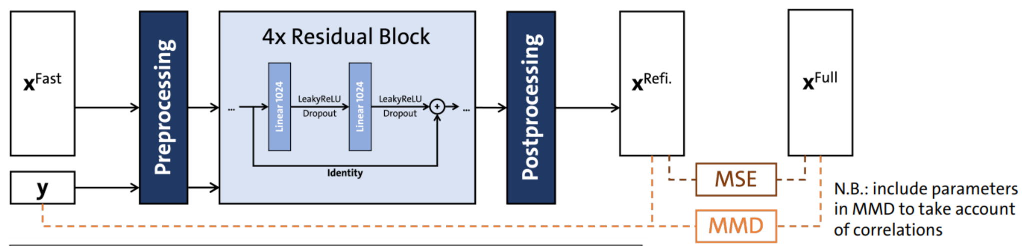
- The reduce systematic uncertainty is important for physics analysis.
- One possible way is to improve the data MC agreement (achieve more accurate simulation)
- Usually, for fast simulation, it has a great speed while the accuracy is lower
- By using ML, one can improve the accuracy of the simulation



Refining

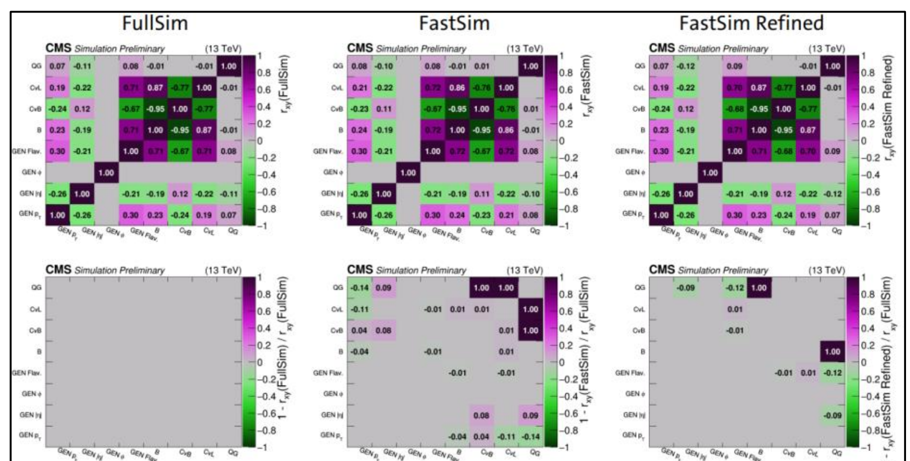
- The interested variables can be refined by ML
- CMS shows how to [refine](#) the jet flavor variable (from fast simulation)
  - Input: x^{Fast} (4 jet flavor discriminators), y (gen jet p_T, η, flavor)
  - Output: the refined $x^{\mathrm{Refi.}}$





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

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

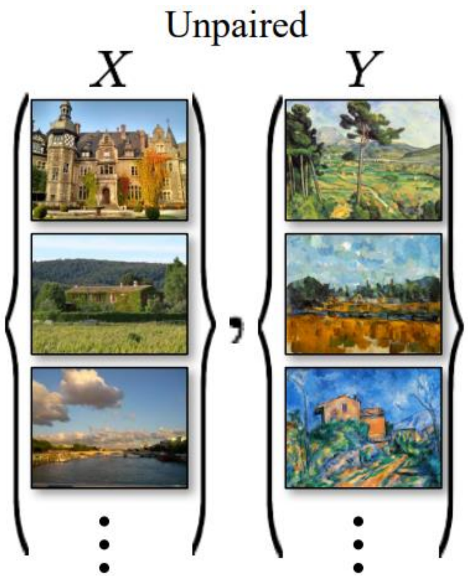


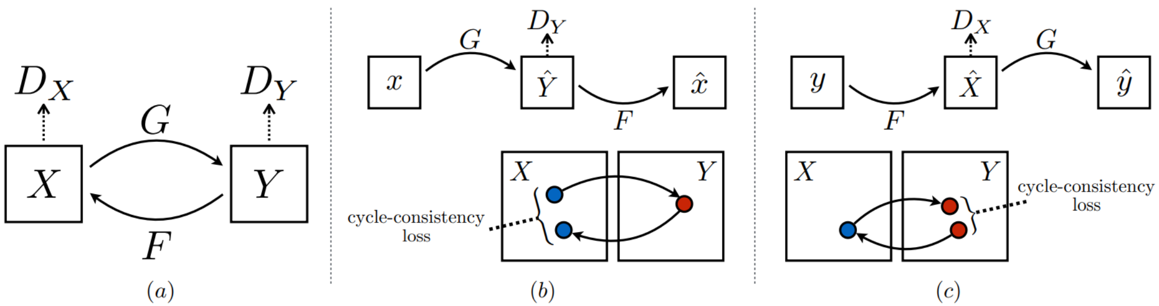
UI2I (Cycle-GAN)

- Unpaired Image-to-Image (UI2I) translation task

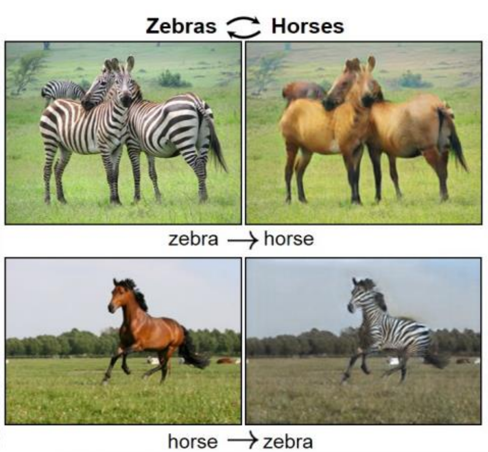
$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F),$$

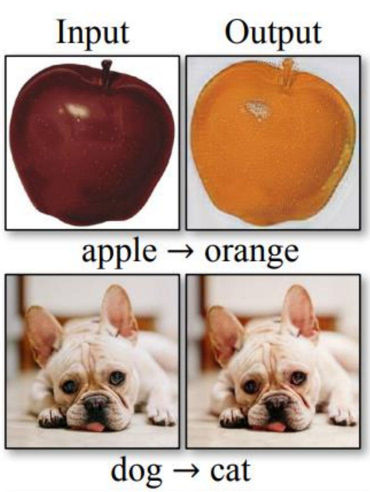
- Cycle-GAN





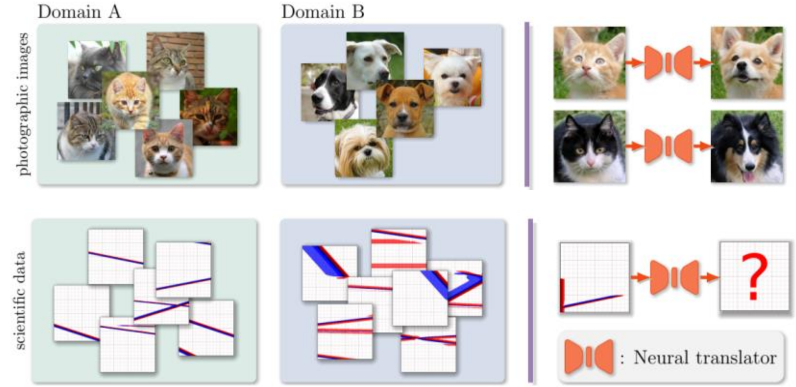
- Often success for color and texture translations
- Little success for geometric changes

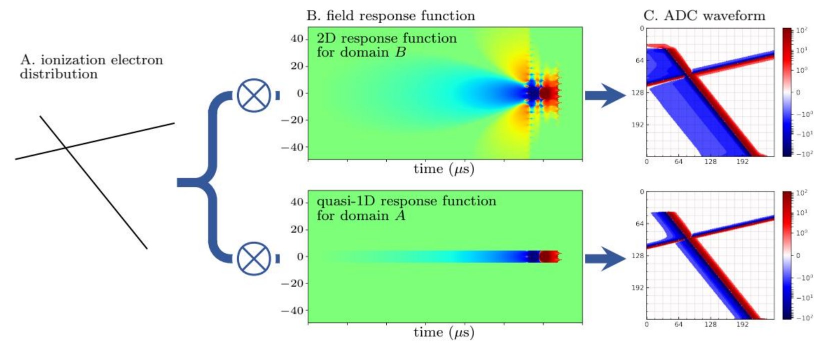


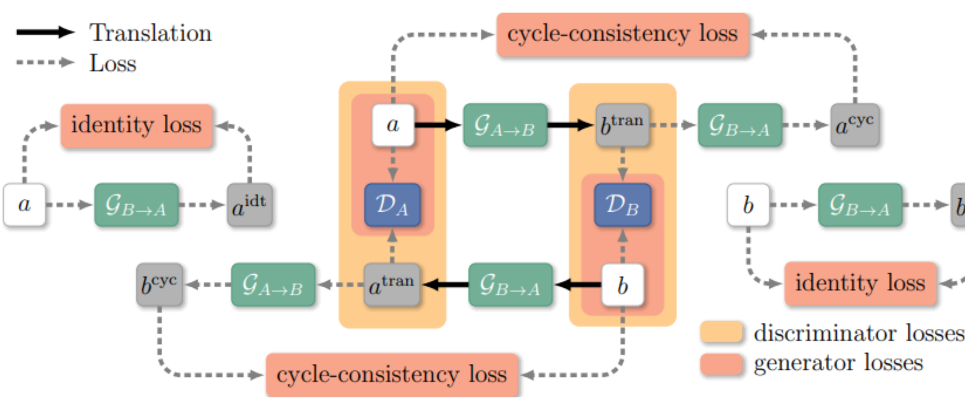


UI2I (LArTPC)

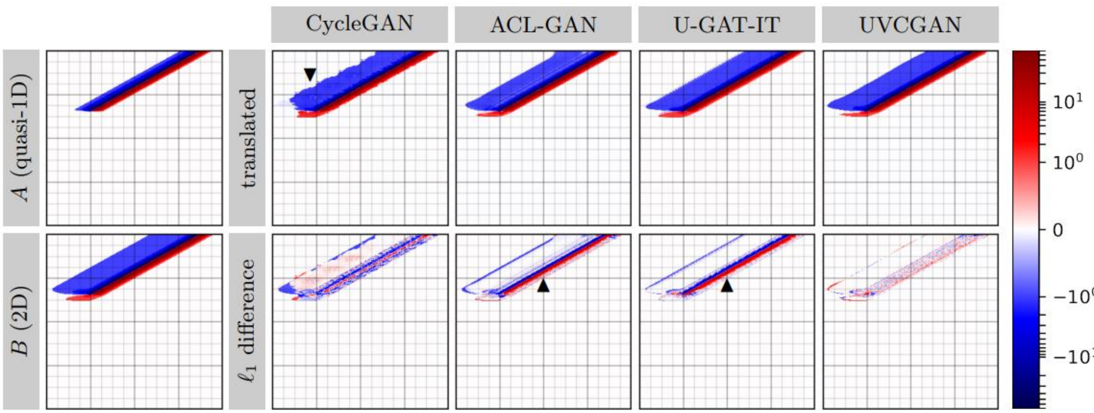
- An example from DUNE LArTPC detector (ACAT2024)
- For a set of simulated simple particle tracks:
  - Domain A: a low fidelity quasi-one dimensional (1D) response function is applied
  - Domain B: a high-fidelity 2D response function is applied





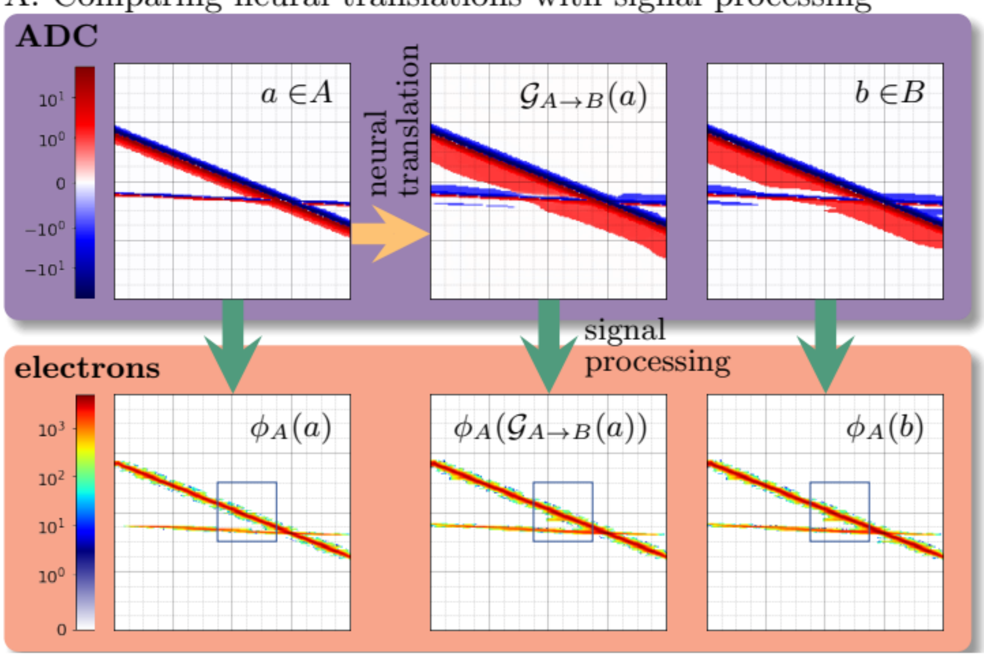


UI2I LArTPC performance

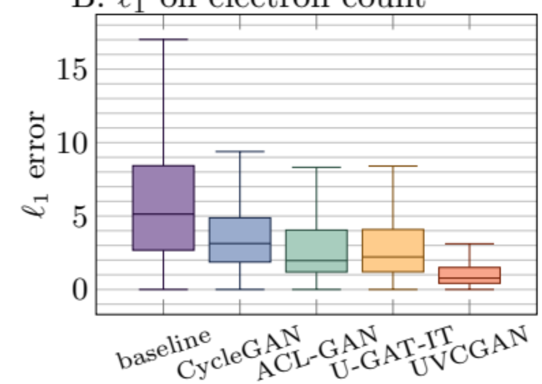


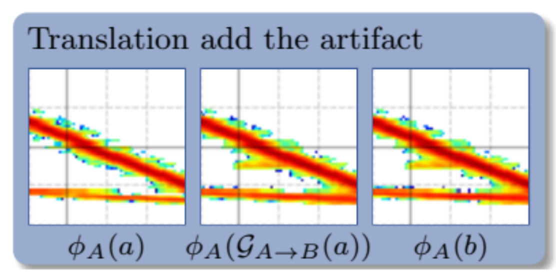
Algorithm	A to B		B to A	
	ℓ_1	ℓ_2	ℓ_1	ℓ_2
CycleGAN	0.074	0.180	0.061	0.159
ACL-GAN	0.083	0.566	0.039	0.121
U-GAT-IT	0.078	1.187	0.073	1.161
UVCGAN	**0.030**	**0.033**	**0.025**	**0.027**

A. Comparing neural translations with signal processing



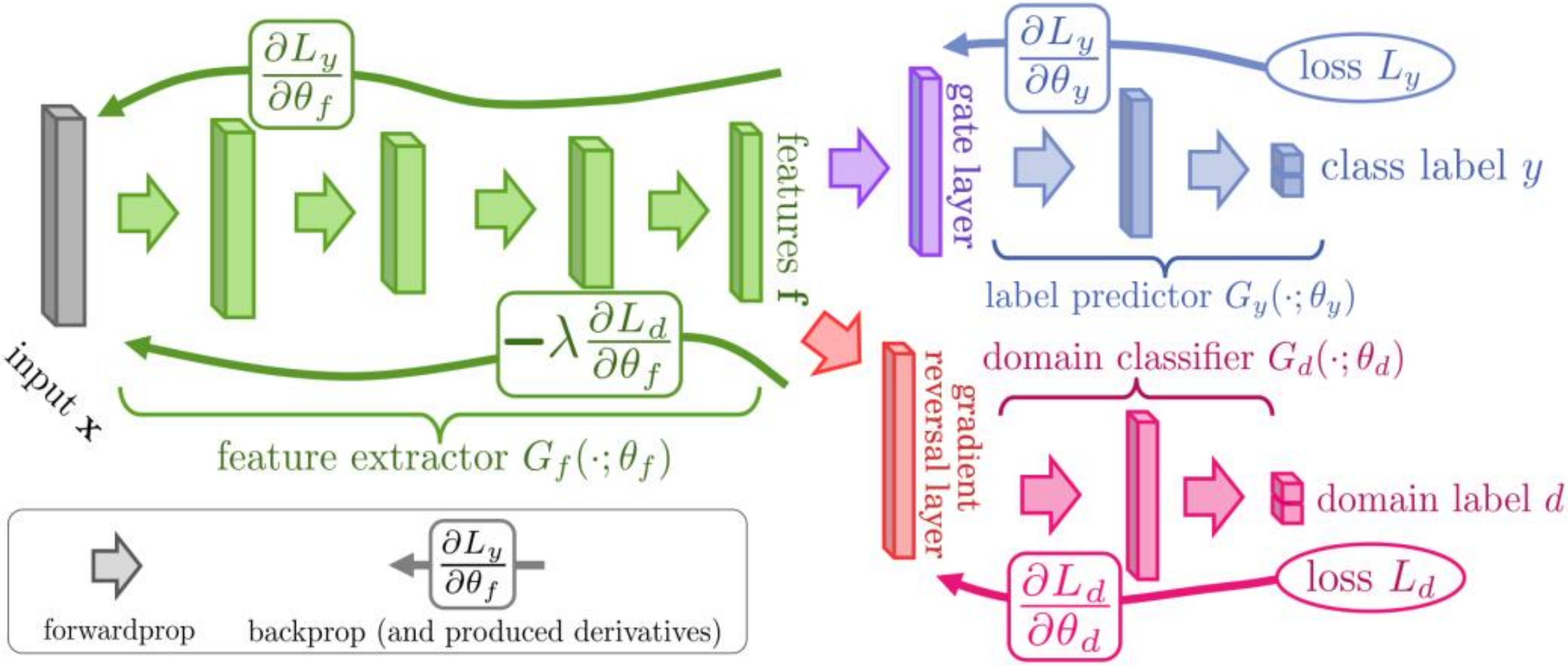
B. ℓ_1 on electron count





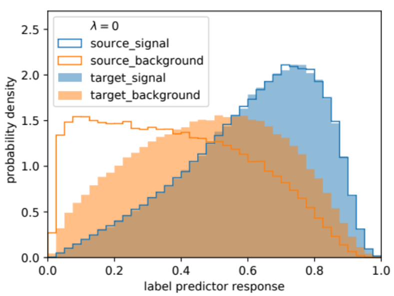
Domain Adversarial Neural Network

- Domain Adversarial Neural Network (DANN)

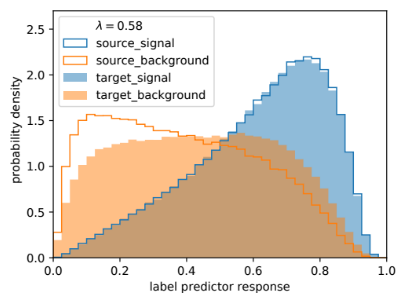


DANN (ATLAS)

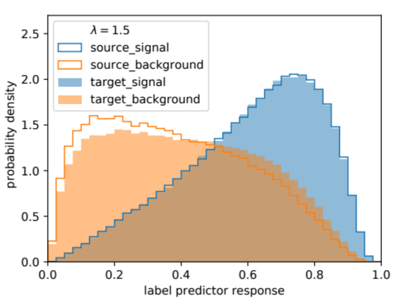
- ATLAS signal background classification
  - Signal 5%: $\mathrm{t\bar{t}H(H \to b\bar{b})}$
  - Background 95%: $\mathrm{t\bar{t} + b\bar{b}}$ (two samples with different generators)
  - Detector simulation: Delphes simulation



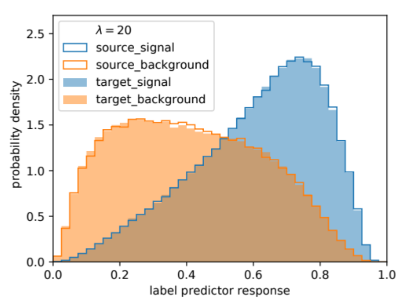
(a)



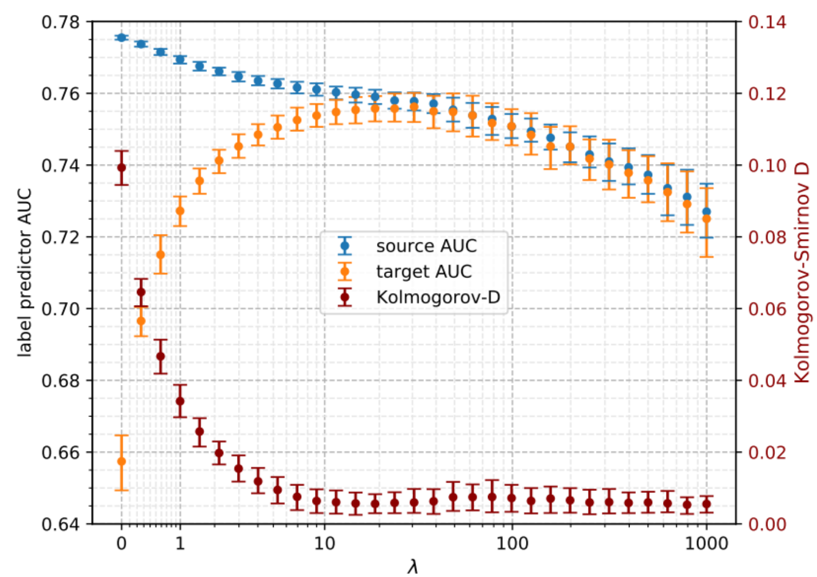
(b)



(c)



(d)

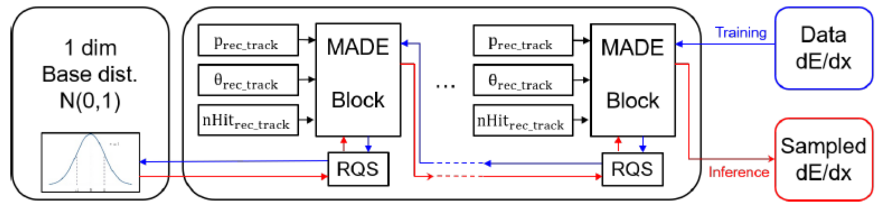


Data-driven simulation

- Learning the distribution of real data and applying it in simulation. For example, the BESIII dE/dx simulation

	$\pi+$	$\pi+$	K^+	K^-	p^+	p^-
Training data	1M	1M	0.5M	0.5M	2M	2M
Testing data	0.4M	0.4M	0.2M	0.2M	0.9M	0.9M

Table 1. The number of training and testing data in million (M).



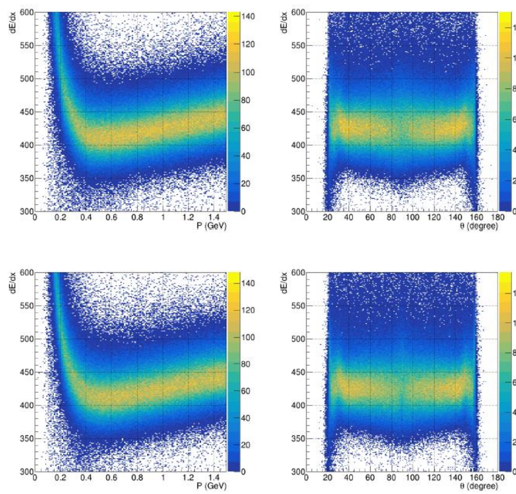
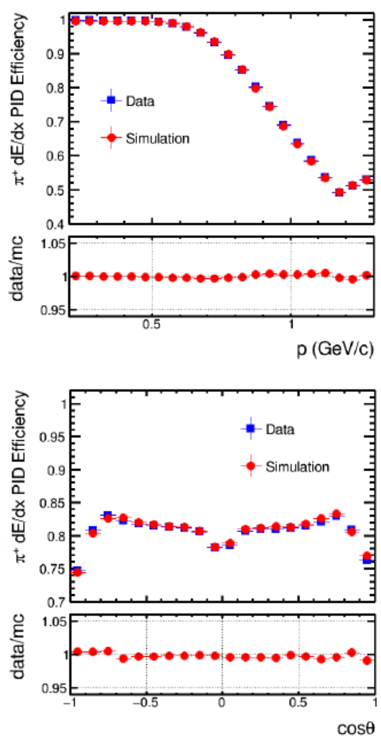
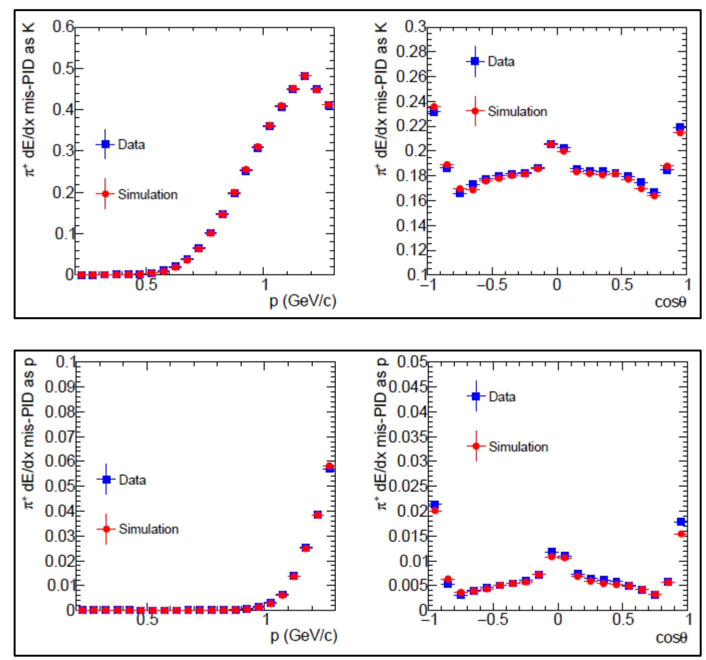


Figure 2. The dE/dx distribution of π^+. The left (right) plots are dE/dx versus momentum (θ). The top (bottom) plots for simulated (data).





Summary

- There are many applications of ML for simulation
- Mainly focus on improving the simulation speed and accuracy
- Many promising results and many challenges
- The field is in a rapid development stage. Please stay tuned
- More in HEPML-LivingReview

Thanks!

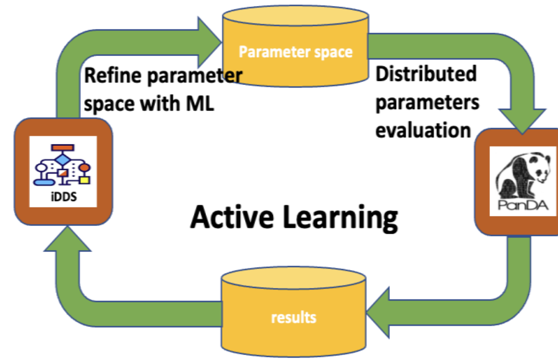
Backup

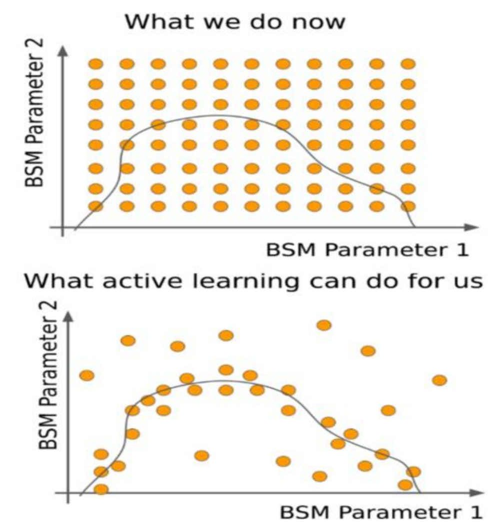
Detector simulation

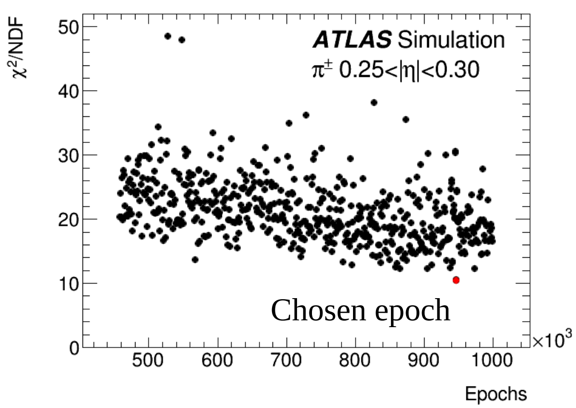
- Calorimeter Fast simulation:
  - FastCaloGAN: a fast simulation of the ATLAS Calorimeter with GANs
  - gaede_chep23_caloml_v01 (jlab.org)
  - Generating Accurate Showers in Highly Granular Calorimeters Using Normalizing Flows
  - Fast and Accurate Calorimeter Simulation with Diffusion Models
  - Transformers 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

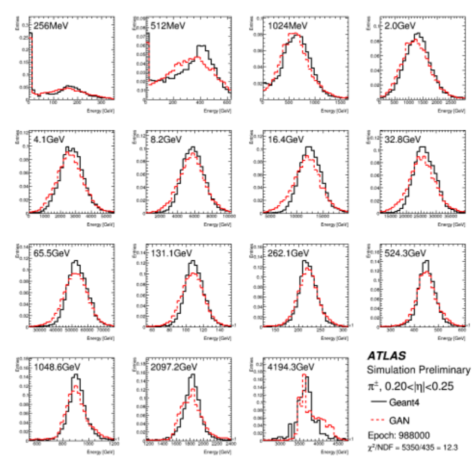
量能器快速模拟（1）

- 由于要训练的 GAN 模型个数多（500 个），且每个 GAN 模型的训练也不容易（训练过程不稳定、需要优化超参数）。因此，需要解决 GAN 模型训练的问题
- 为此 ATLAS 实验利用 Active Learning 的技术，实现网络模型的自动训练和超参数的优化：
  - iDDS（intelligent Data Delivery Service）负责根据当前模型训练的结果产生下一批模型训练作业（如利用贝叶斯算法、GP 等算法缩小超参数范围）
  - PanDA 系统将作业调度到分布式的异构计算资源上进行模型的训练，返回训练结果









The CEPC case (performance)

CEPC 2019

Dataset:

- photon showers in ECAL Barrel
- 31x31x29 voxels