Currently:

Formerly:

Normalizing Flows for high-dimensional HEP.

Humberto Reyes-González
RWTH Aachen

ML coffee.
LPSC Grenoble.
22/03/24

Motivation

- Machine Learning is becoming a part of our society. Is being used everywhere from finance and health to climate research and astrophysics.

- High Energy Physics is not the exception. Machine learning is making great strides towards revolutionizing the field.

- Large amounts of more complex data is being collected by the Large Hadron collider, as well as by a number of other under on and above ground astrophysical experiments etc.

- To be able to draw insights from this data about the fundamental laws of Nature, we must rely on modern data science, ML included.

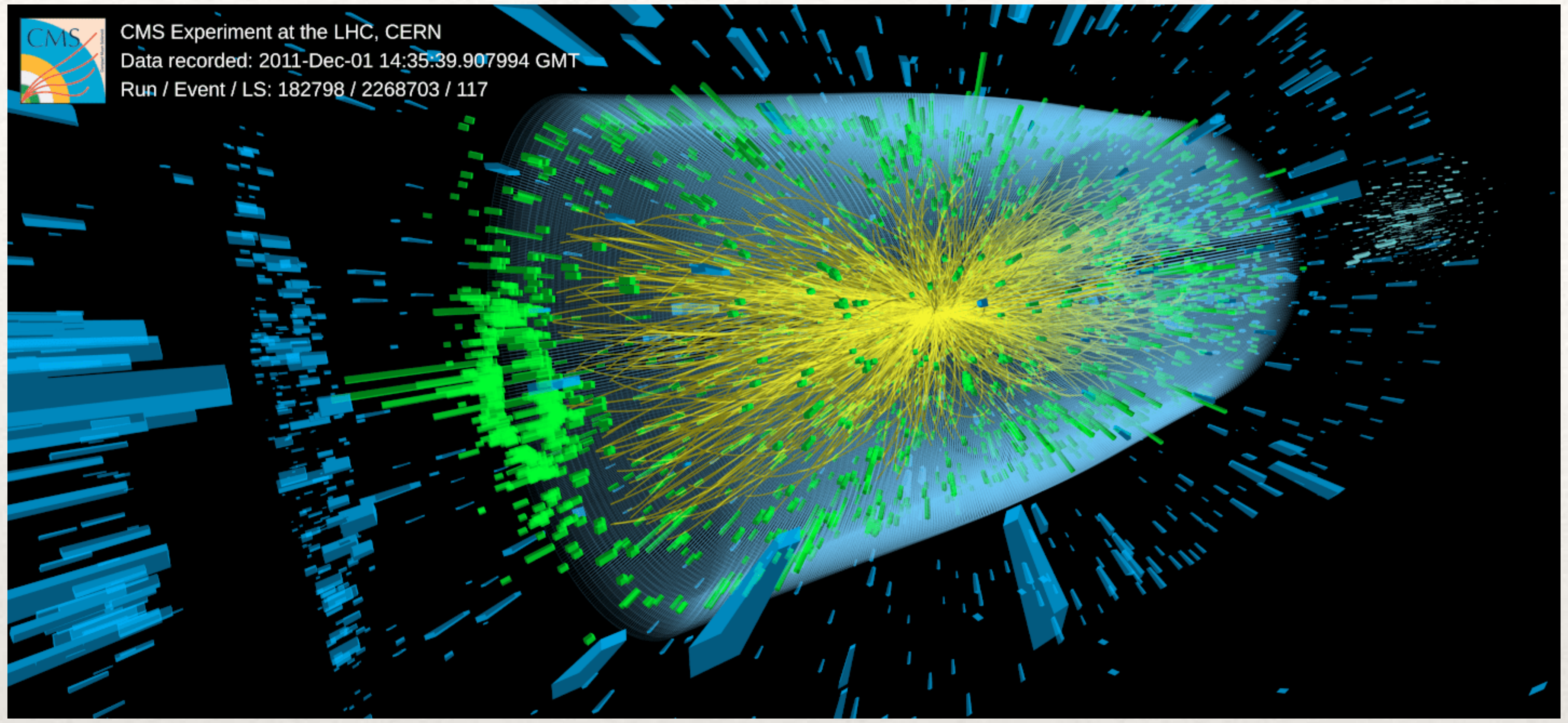
- To ensure the systematic usage of ML methods, we should be aware of their capabilities and limitations.

- A particularly interesting brand of ML are the so-called Normalizing Flows. Generative networks with explicit density estimation.

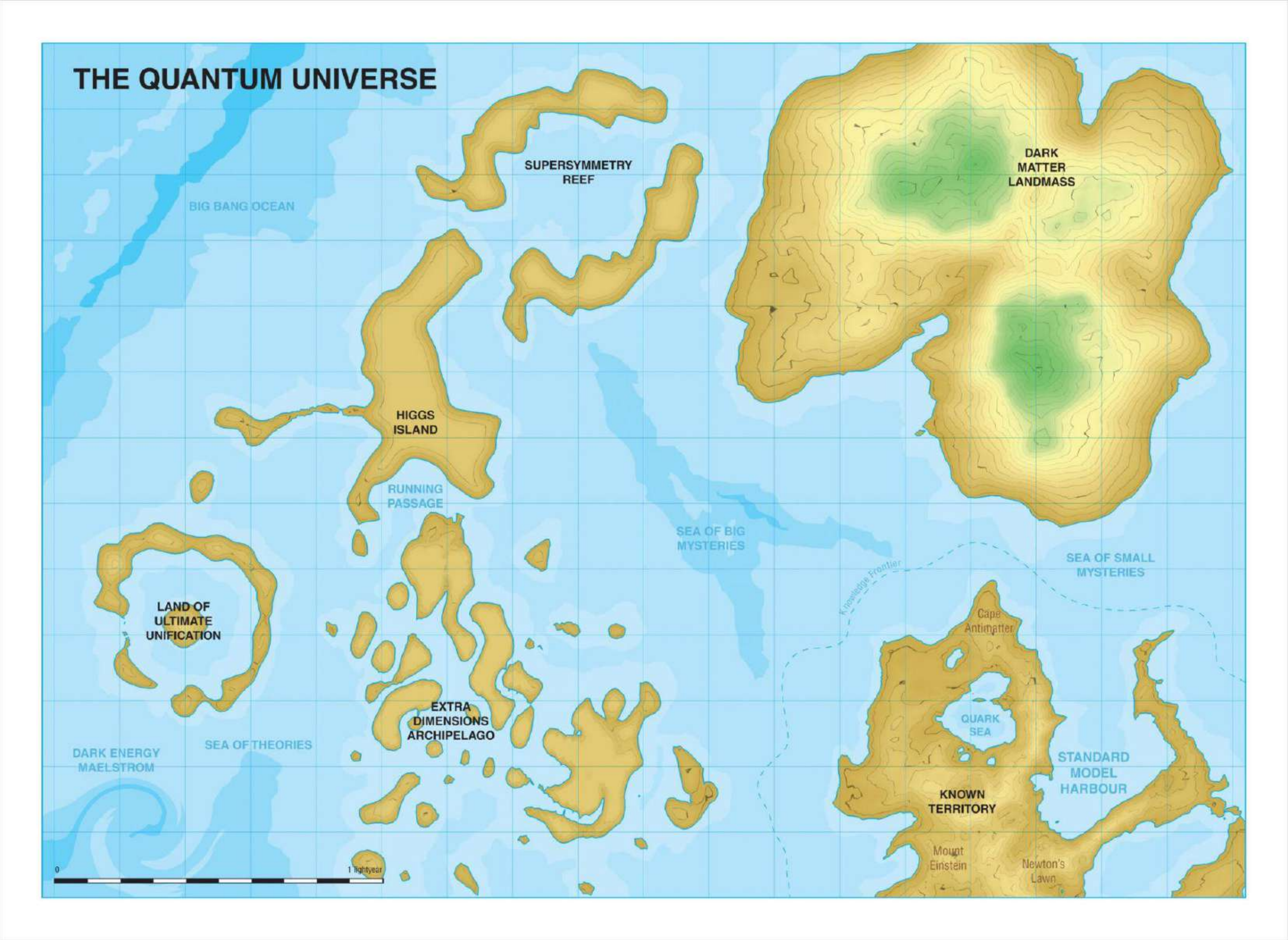
Outline

1. Machine Learning (ML) and High Energy Physics (HEP).
2. Normalizing Flows (NFs).
3. NFs 4 HD-HEP.
4. Conclusions.

Challenges in HEP



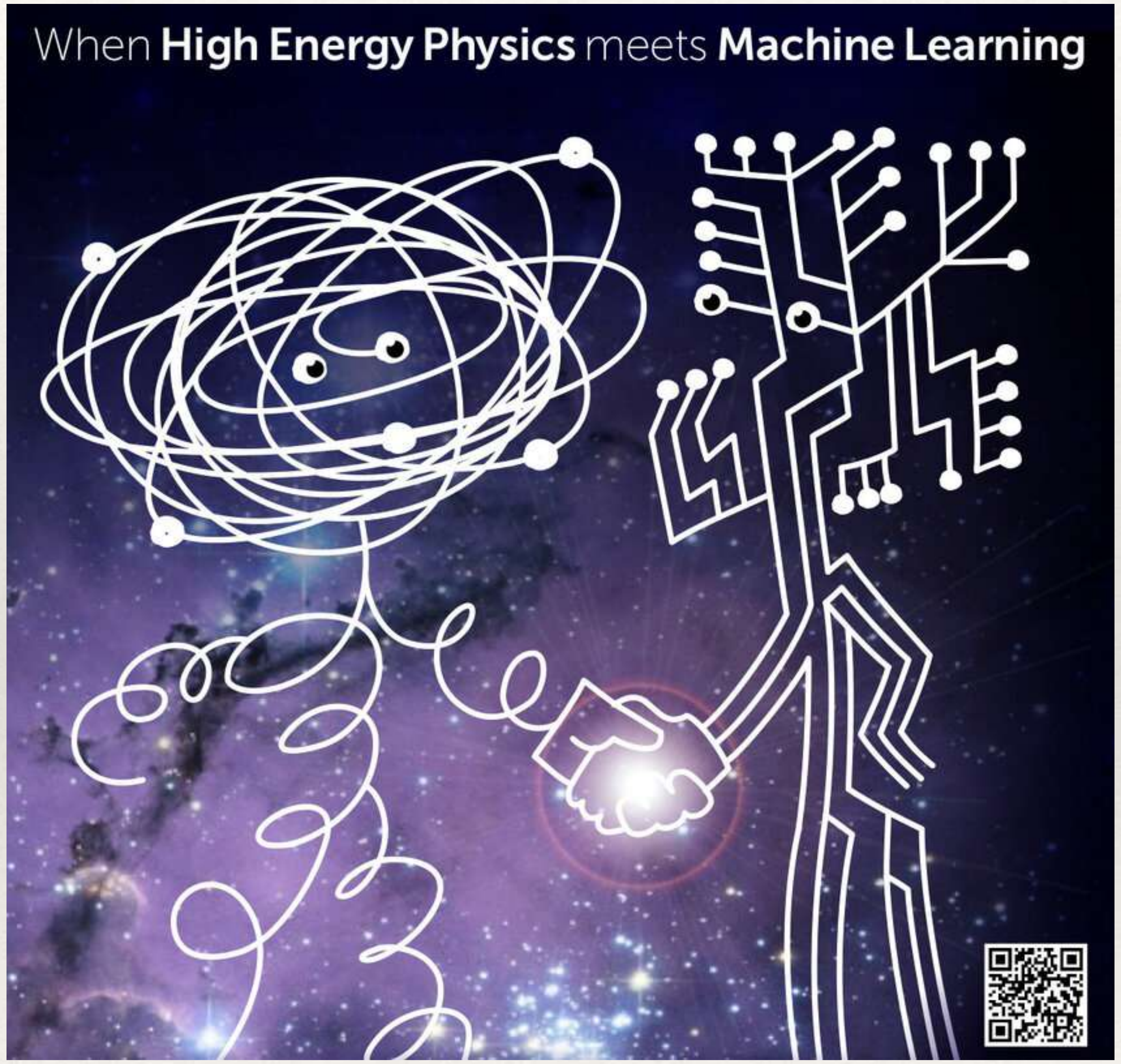
https://images.app.goo.gl/hfdgqBWqNngoakQD8



Challenges in HEP

Machine learning (ML) has the potential to address several challenges in high-energy physics (HEP). For:

- Enabling faster and more accurate data analysis,
- Improving signal processing and background rejection,
- Aiding in anomaly detection,
- Enhancing simulations and modeling,
- Optimizing detector design, and
- Facilitating efficient data compression and storage



The HiggsML poster advertising the challenge.- Clara Nellist, https://www.researchgate.net/

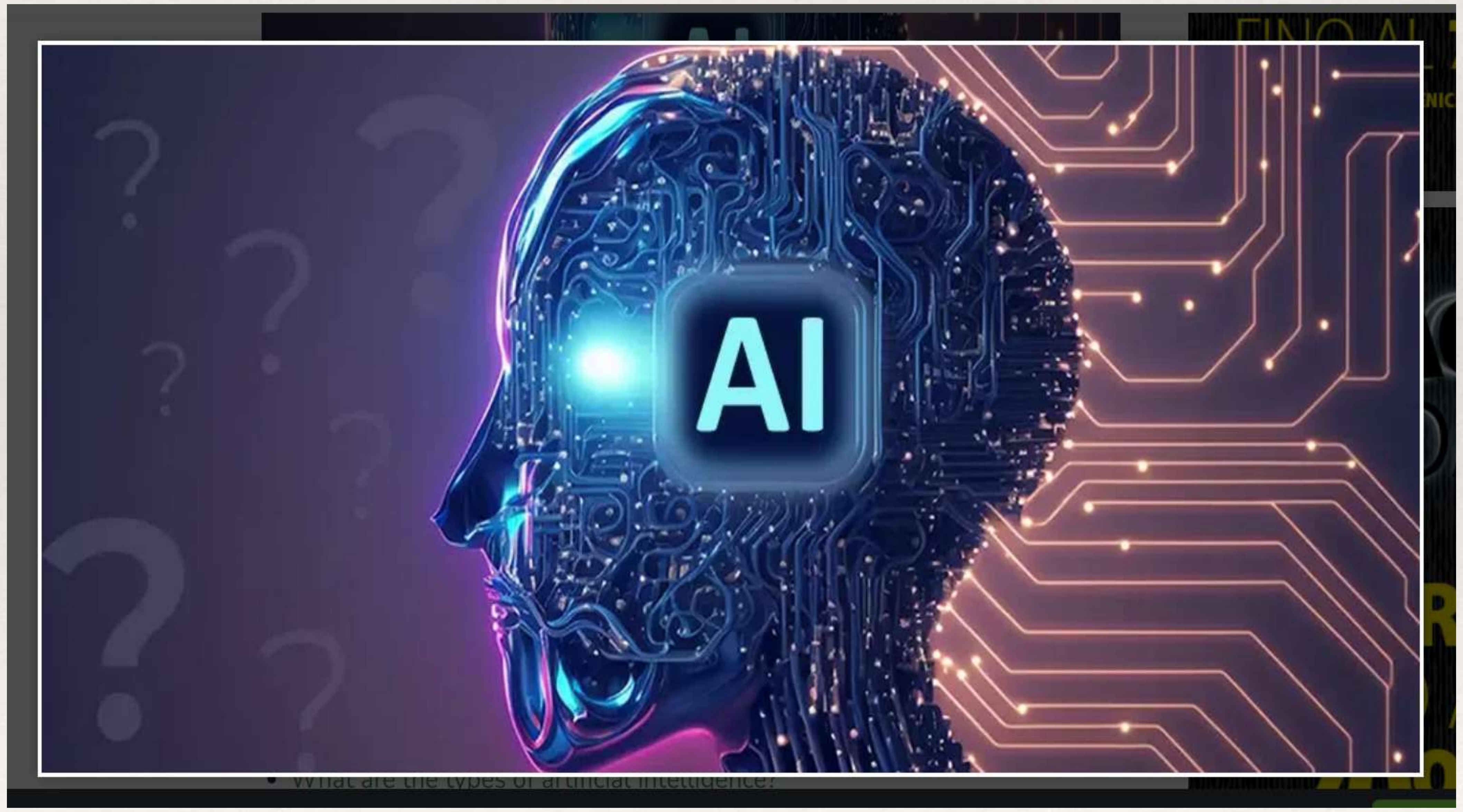
ML in a nutshell

Machine learning is a branch of artificial intelligence (AI) that focuses on the development of algorithms and statistical models that allow computers to learn and make predictions or decisions without being explicitly programmed Levering from large amounts of data it allows to extract patterns, learn from them, and make informed decisions or predictions.

ML is revolusinind many fileds:

1. Healthcare
2. Finance
3. Transportation
4. E-commerce and Marketing
5. Natural Language Processing (NLP)
6. Manufacturing and Quality Control
7. Energy and Sustainability

AND SCIENCE!!!

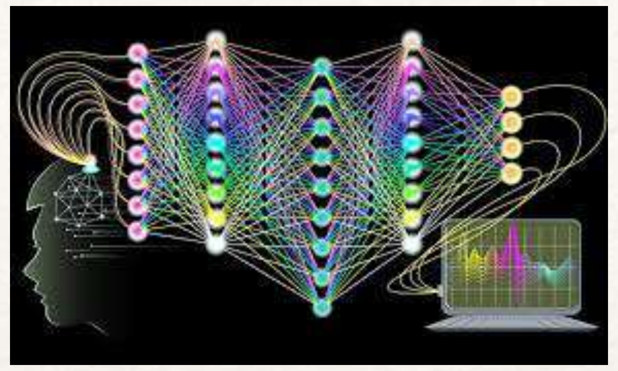


https://www.geeksforgeeks.org/what-is-artificial-intelligence/

The ML landscape

Supervised Learning:

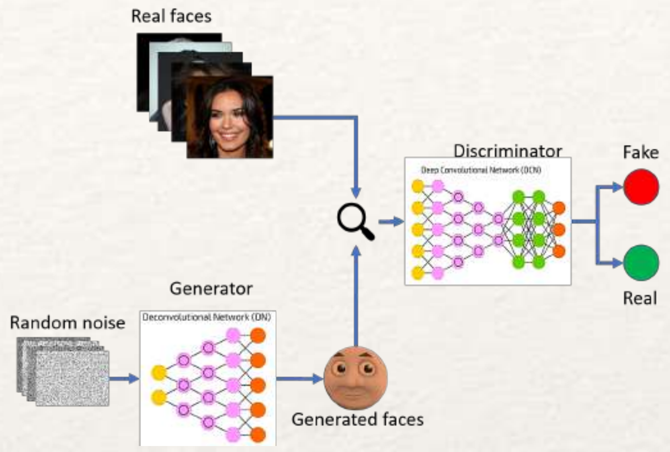
1. Random Forests
2. Boosted decision trees
3. Neural Networks
4. Graph neural Networks
5. …



Unite Ai

Unsupervised Learning

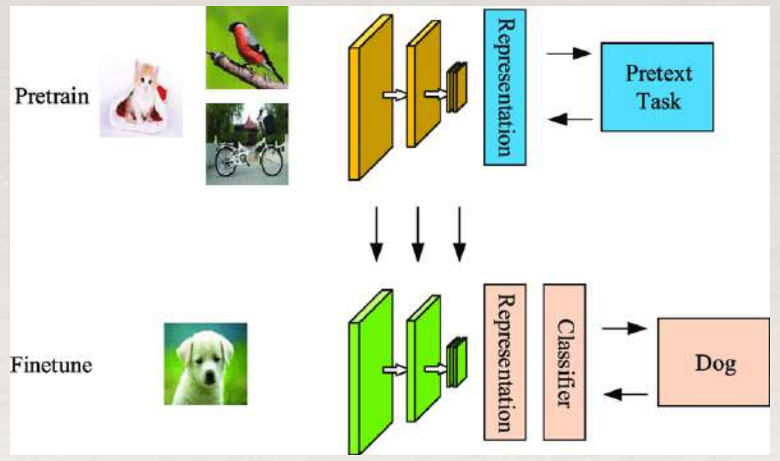
1. Variational Autoencoders
2. Generative Adversarial Networks
3. **Normalizing Flows**
4. Diffusion Models
5. …



.spindox.it

Self-Supervised.

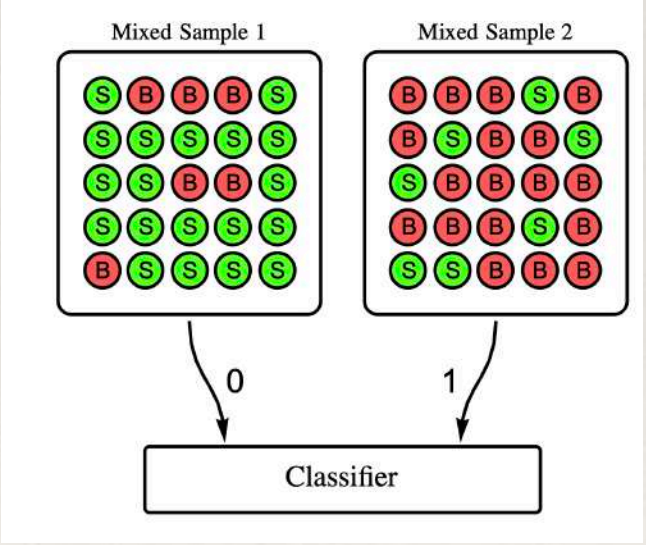
1. Large Language Models
2. Vision Transformers.
3. Contrastive Learning.
4. …



https://www.v7labs.com/blog/self-supervised-learning-guide

Weakly-supervised:

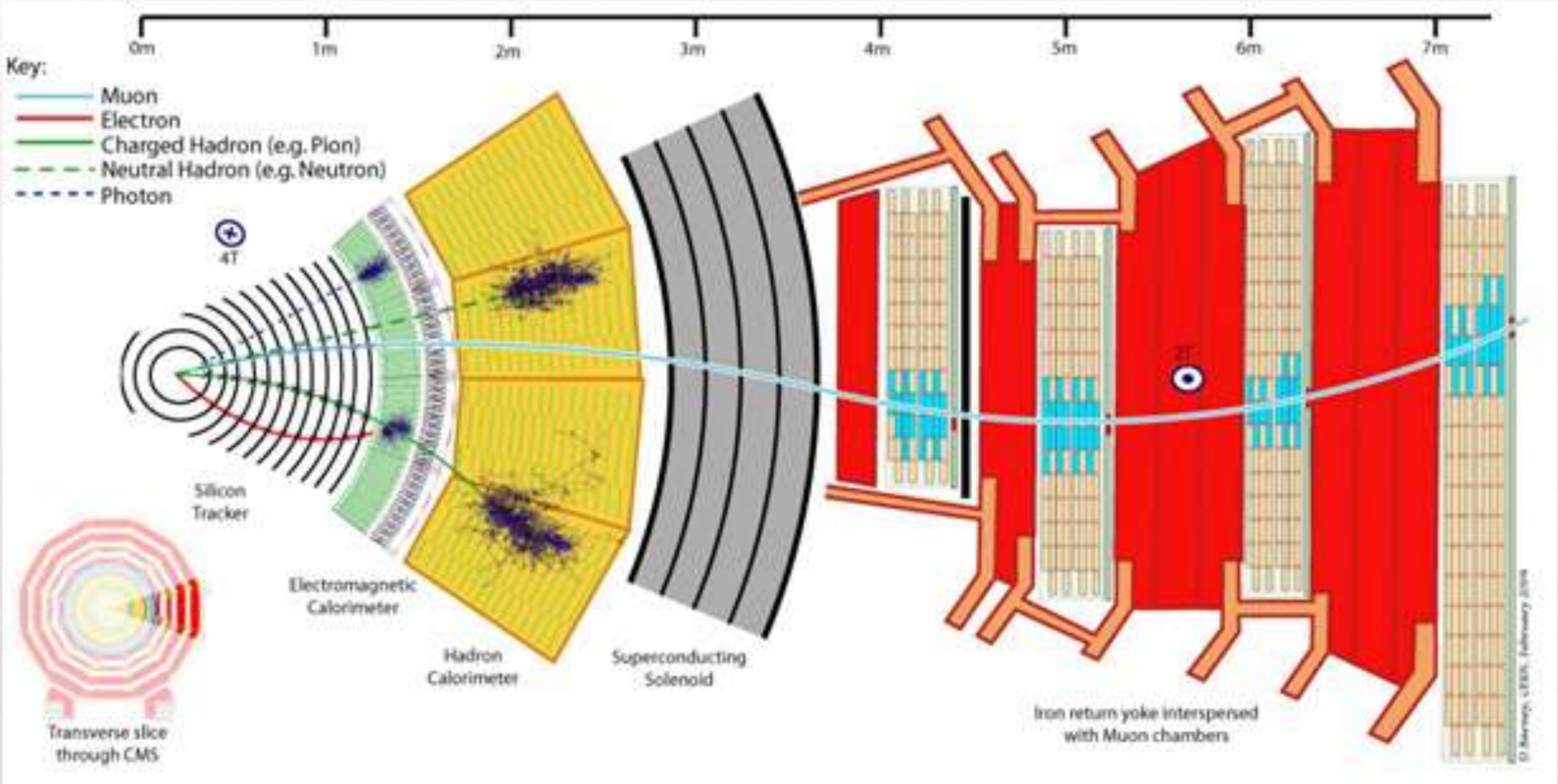
1. Anomaly detection Classifiers
2. Active Learning.
3. Transfer Learning.
4. …



arXiv:1708.02949

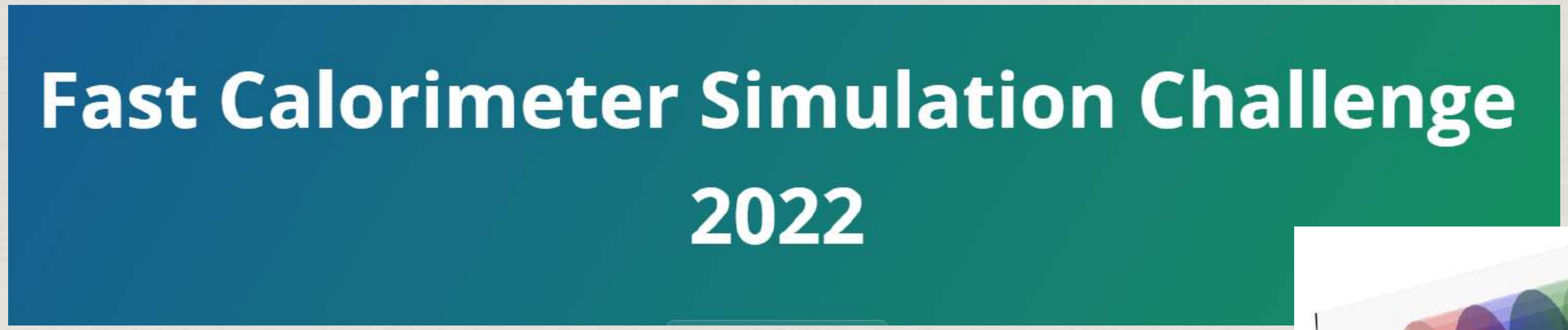
Applications of ML in HEP

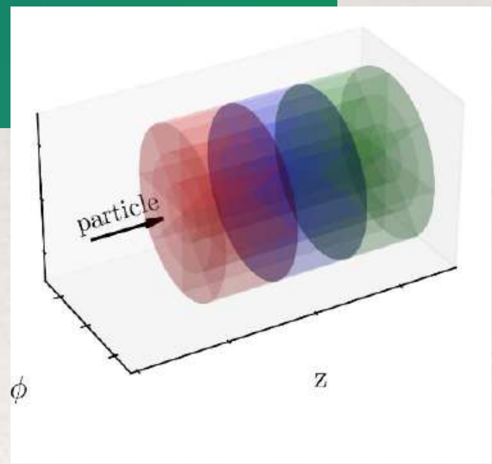
1. Particle identification
2. Event Reconstruction.
3. Anomaly Detection
4. Unfolding.
5. Calibration and Corrections
6. Event Simulations
7. Analysis design.
8. Likelihood learning.
9. …



HEP data has particular challenges as compared to 'real world data' !

Open challenges ML4HEP





and more to come… stay tuned and participate!

Normalizing Flows.

Introduction

- In HEP we find complex Probability Distribution Functions (PDFs) EVERYWHERE!
- What do we want to do with them? -> (Re)-interpret, preserve, sample, combine, invert, …
- Can Normalizing Flows (NFs) help us on these endeavours?…

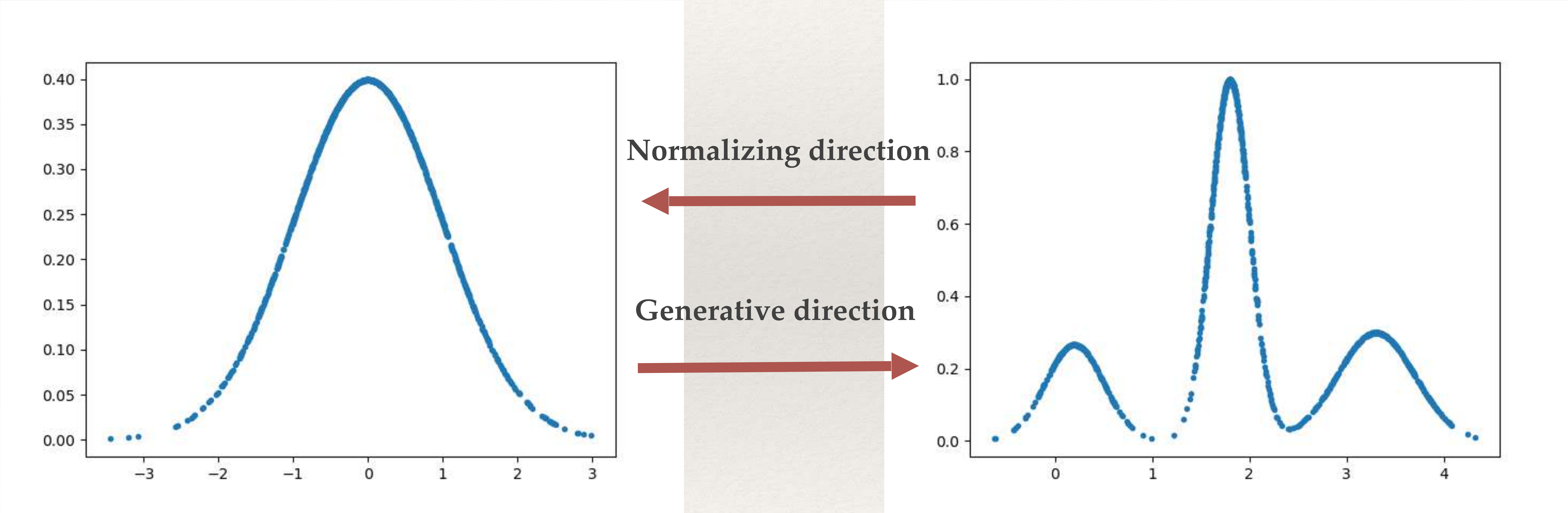
- Normalizing flows are a powerful brand of generative models.
- They map simple to complex distributions.
- They allow for efficient sampling of complex PDFs…
- … and include density estimation by construction!

Some applications:

- Learning LHC likelihoods (arxiv:2309.09743)
- Unfolding (arXiv:2006.06685)
- Calorimeter shower simulation (arXiv:2106.05285)
- Event generation and numerical integration (arXiv:2001.10028, arXiv:2001.05486 ,arXiv:2110.13632)

Basic principle

Following the change of variables formula, perform a series of **bijective, continuous, invertible** transformations on a *simple* probability density function (pdf) to obtain a *complex* one.



See review: Ivan K. et. al.
arXiv:1908.09257

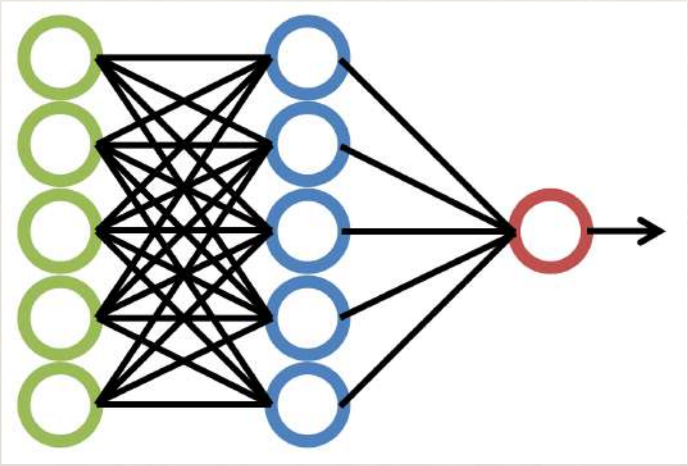
Choosing the transformations

THE OBJECTIVE:

To perform the right transformations to accurately estimate the complex underlying distribution of some observed data.

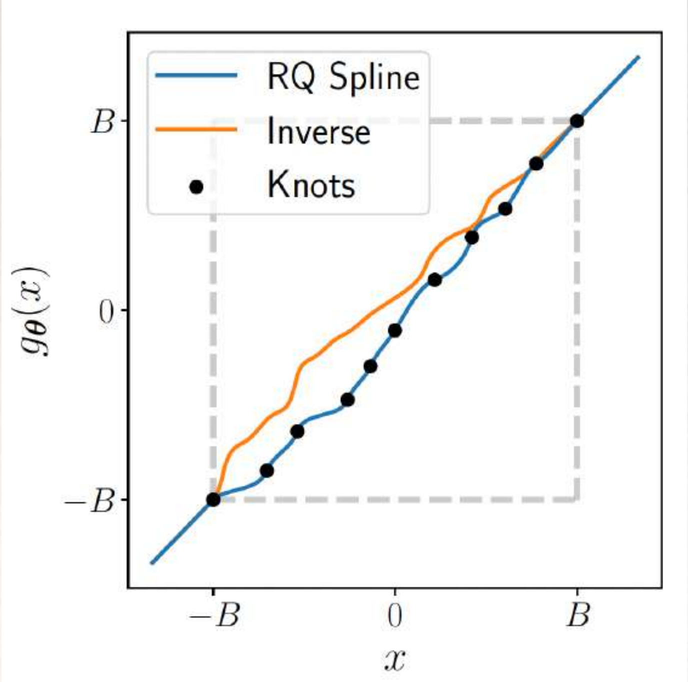
THE RULES OF THE GAME:

- The transformations (bijectors) must be invertible
- They should be sufficiently expressive
- And computationally efficient (including Jacobian)



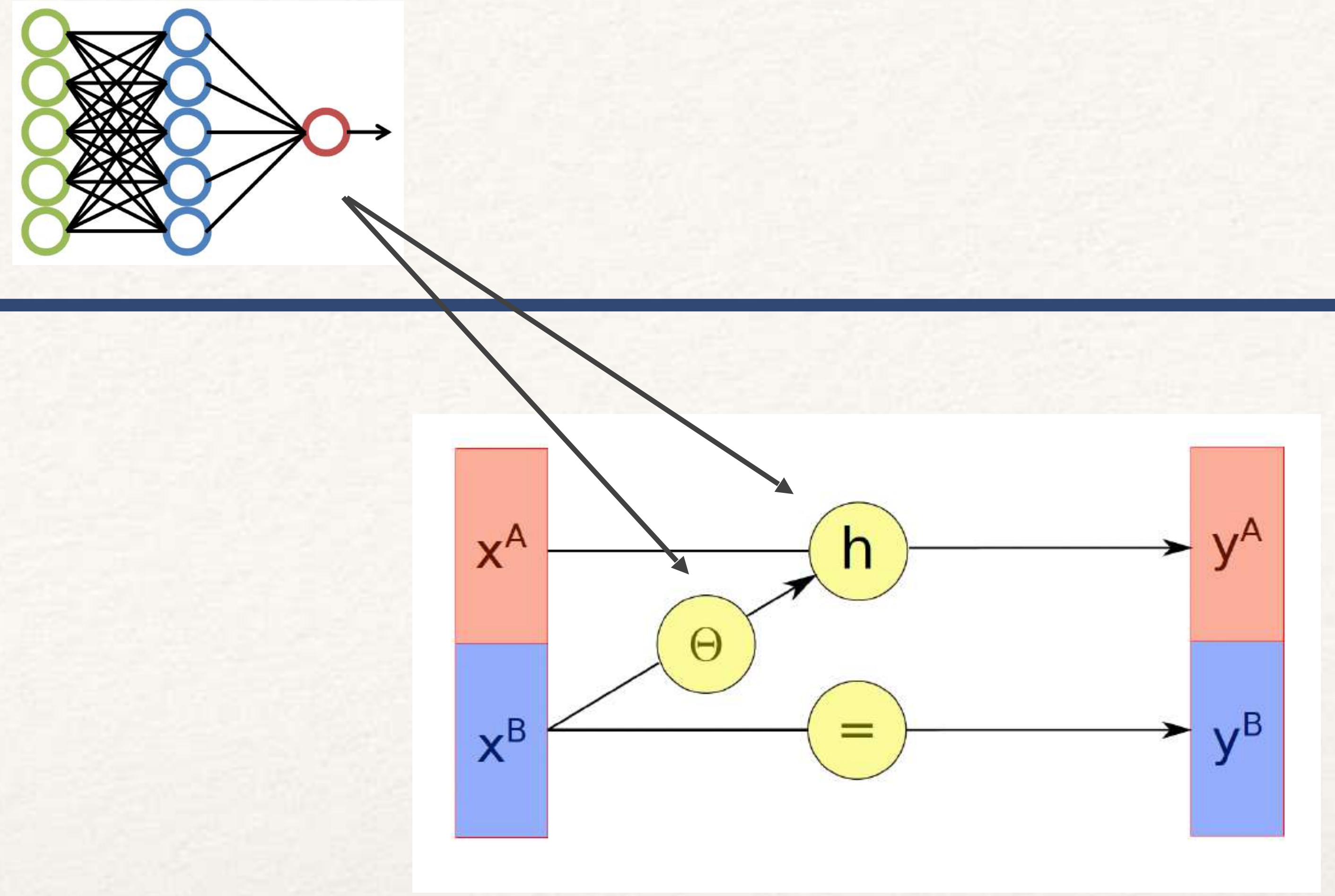
THE STRATEGY:

Let ***Neural Networks*** learn the parameters of ***Autoregressive* Normalizing Flows.***



*One among many types of NFs

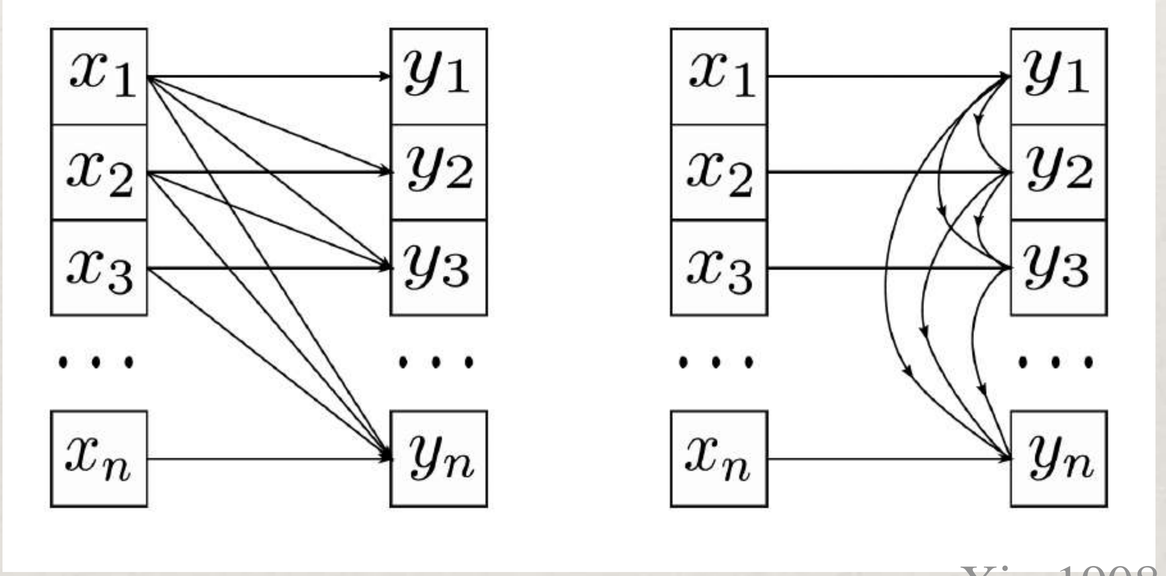
Autoregressive Flows



arXiv:1908.09257

Coupling Flows:

- Dimensions are divided in two sets: and
- We transform with bijectors trained with .
- The bijector parameters are functions of a NN.
- The Jacobian J is triangular ->
- **Jacobian is easily computed!**
- **Direct sampling AND density estimation.**
- **Less expressive.**



arXiv:1908.09257

Autoregressive Flows :

- Dimension is transformed with bijectors trained with
- Bijector parameters are trained with Autoregressive NNs.
- The Jacobian J is also triangular thus…
- **Jacobian is easily computed!**
- **Direct sampling OR density estimation.**
- **More expressive.**

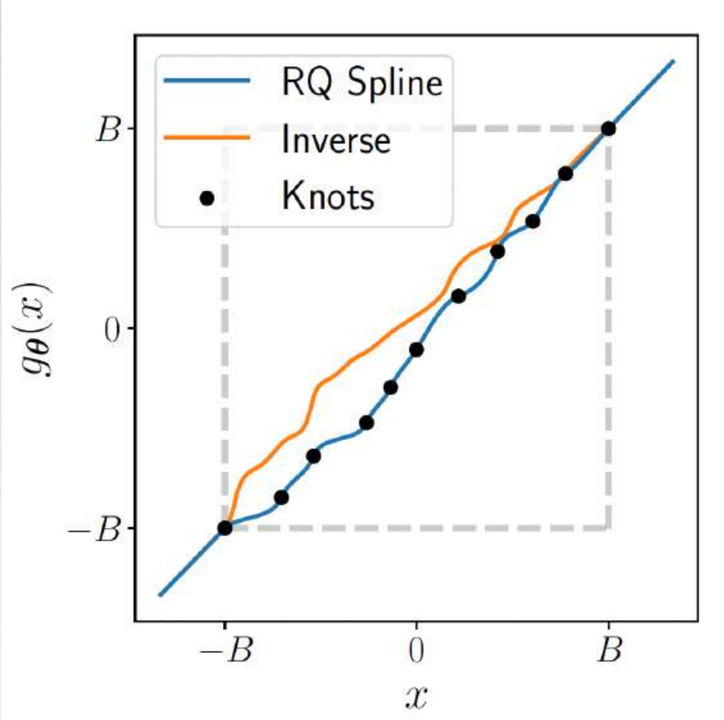
The loss function:

$-\log(p_{AF}(target_{dist}))$

Autoregressive Flows

Bijector / NF type	Affine	Rational Quadratic Spline
Coupling	**RealNVP**	**C-RQS**
Autoregressive	**MAF**	**A-RQS**

$$y = \Theta x + h$$



NFs for high-dimensional HEP

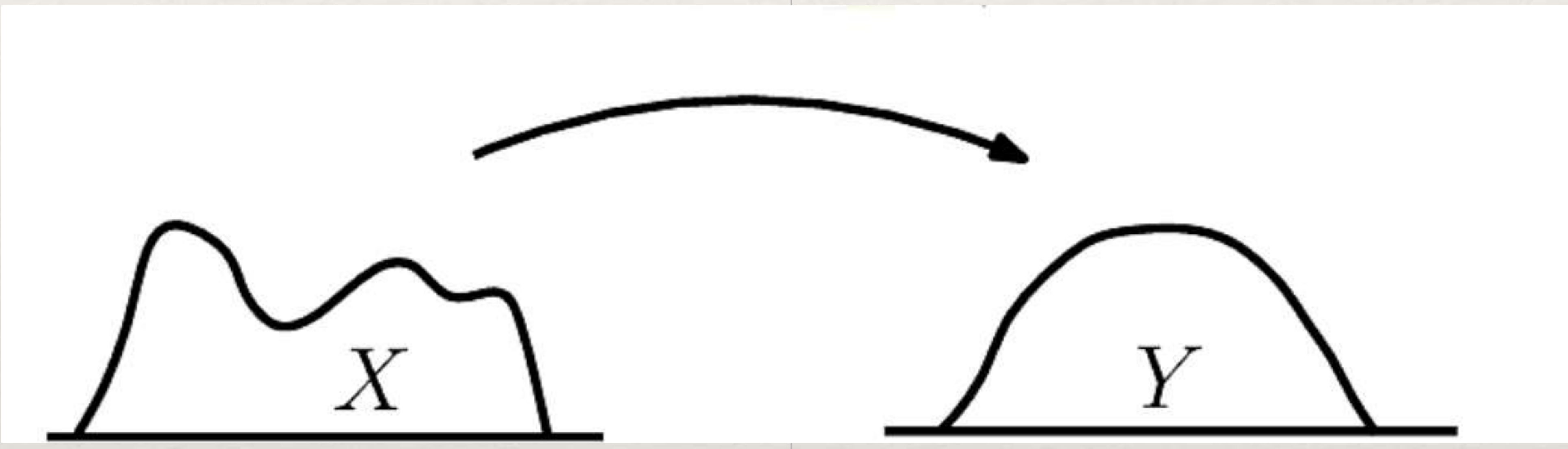
Testing ML methods

Non-parametric methods

- Non-parametric methods are statistical techniques that do not make any assumptions about the underlying probability distribution or data structure.
- They are used to evaluate the quality of the generated samples without relying on predefined probability distributions.
- Their applicability for high dimensional distributions is an active topic of research..

Examples:

- Kolgomorov-Smirnov test
- Anderson-Darling test.
- (Sliced) Wasserstein Distance
- Frechet physics distance
- Kernel physics distance
- Clasifier-based tests.



- R, Kansal et. al. Evaluating generative models in high energy physics. (arXiv:2211.10295)
- R. Torre, et. Al. Performances of non-paratemtric two sample tests for high dimensional samples. *To appear*.

Non-parametric methods

- Two-sample 1D Kolgomonov - Smirnov test (ks test):

- Computes the p-value for two sets of 1D samples coming from the same *unknown* distribution.
- We average over ks test estimations and compute the median over dimensions.
- **Optimal value 0.5**

$$D_{y,z} = \mathbf{sup}_x \mid F_y(x) - F_z(x) \mid ,$$

- The sliced Wasserstein distance:

- The one-dimensional Wasserstein distance between two empirical distributions is formulated as:

$$W_{y,z} = \int_{\mathbb{R}} dx \mid F_y(x) - F_z(x) \mid .$$

- In our sliced approach, we randomly select Nd = 2D directions, with D the dimensionality of the sample, uniformly distributed over the 4π solid angle.
- We then project all samples on such directions and compute the one-dimensional Wasserstein distance and finally take the mean over the directions.

NFs in High Dimensions

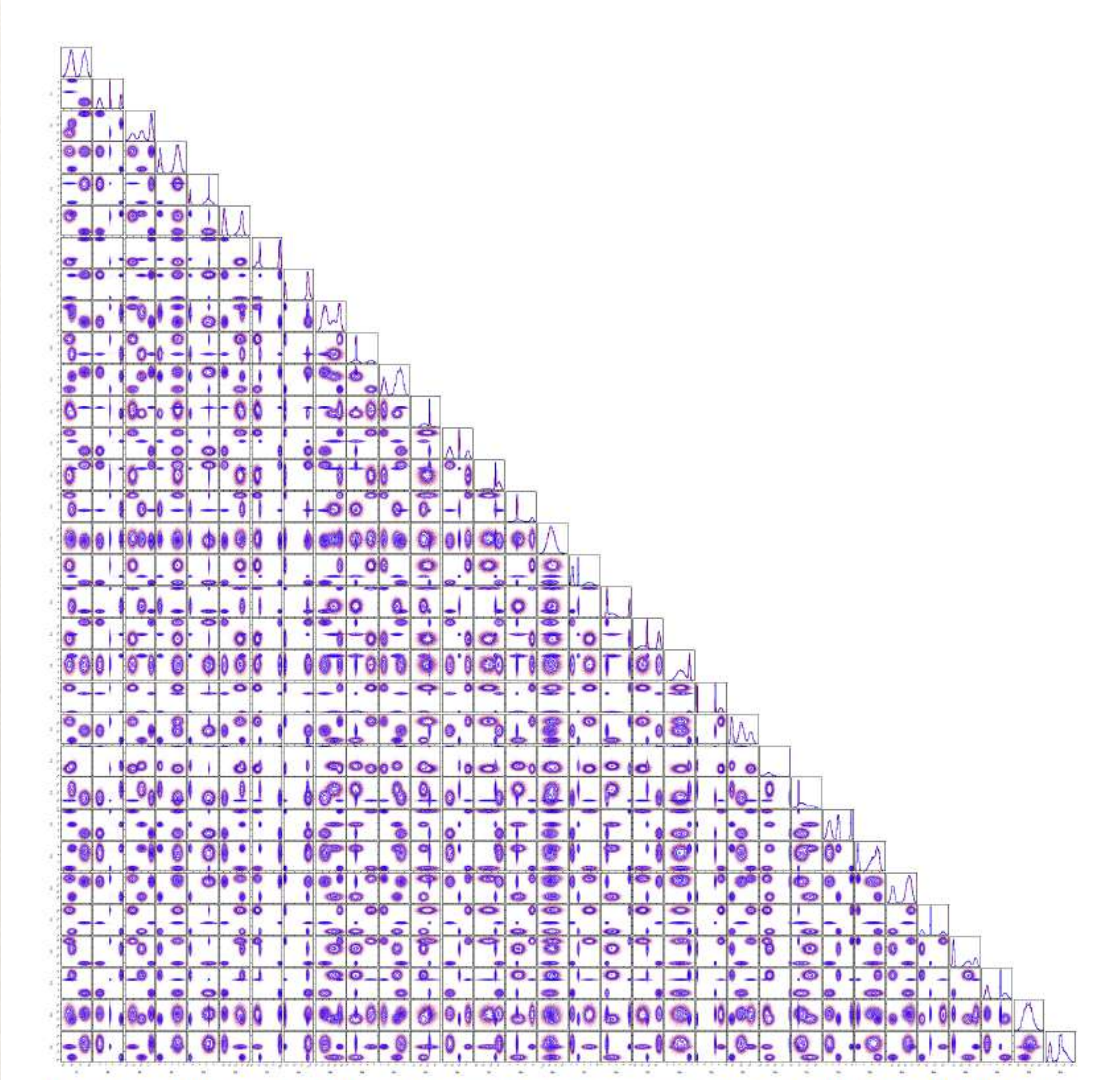
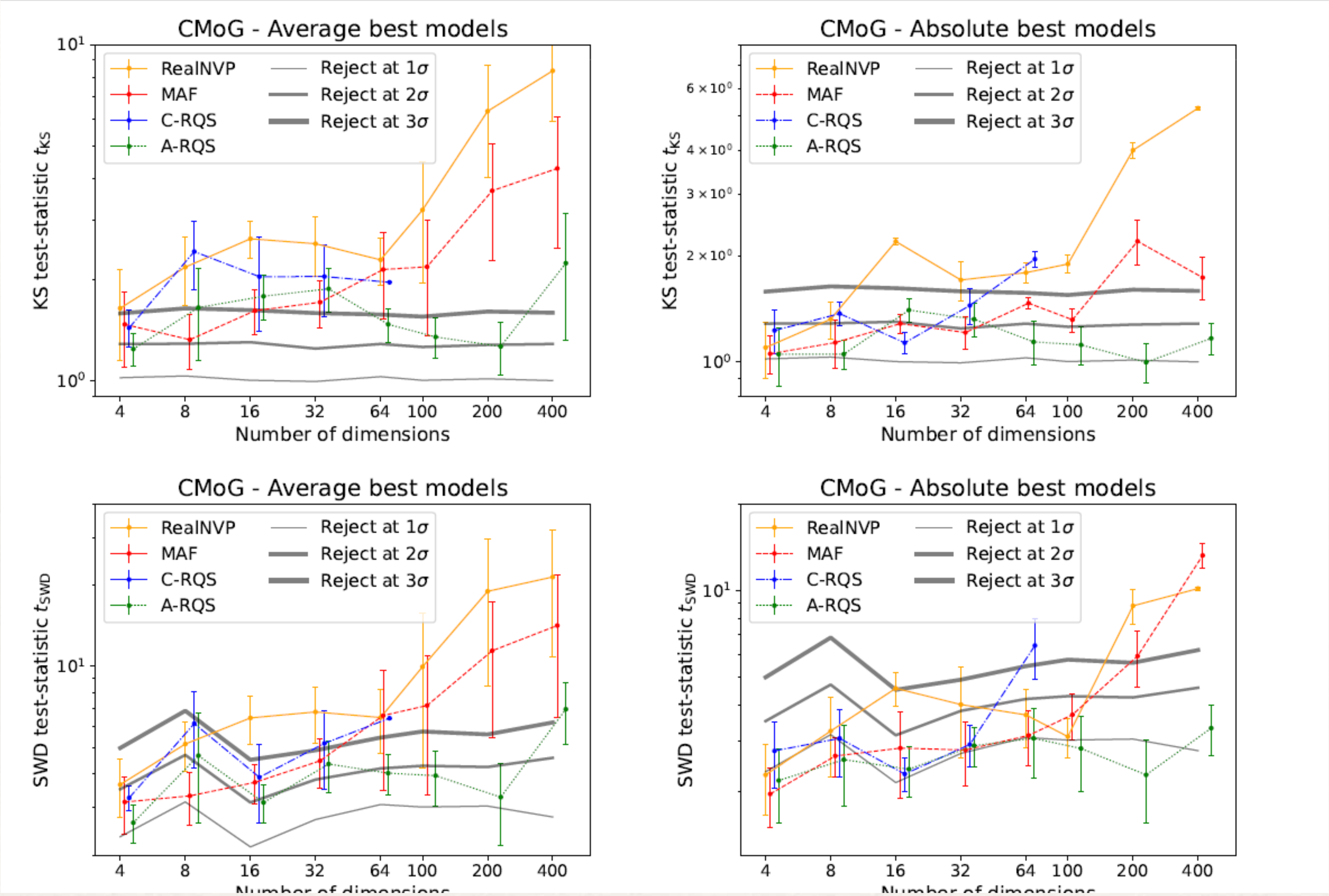
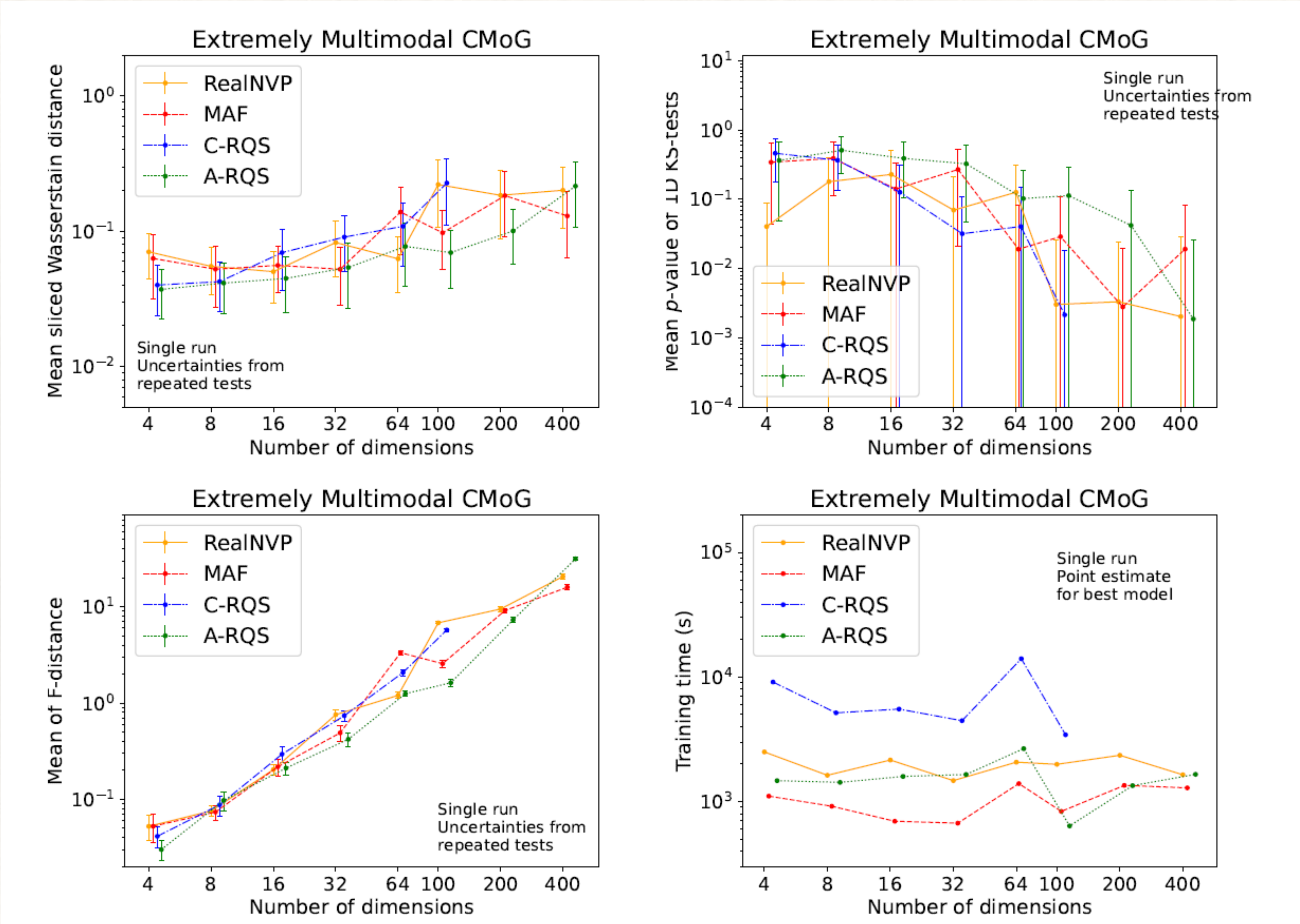


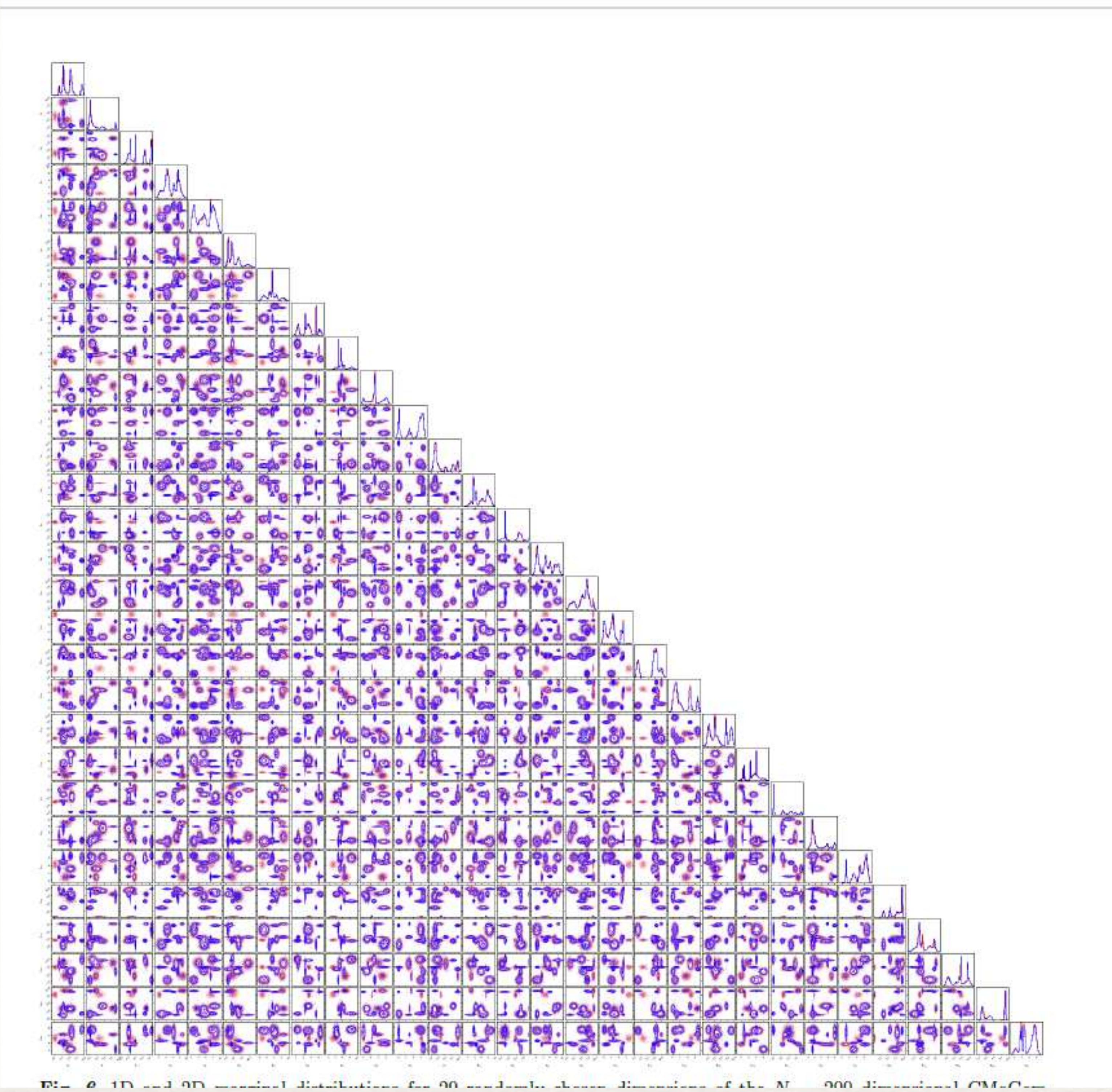
Fig. 5 1D and 2D marginal distributions for 32 randomly chosen dimensions of the $N = 1000$ dimensional CMoG distribu-



A. Coccaro, M. Letizia, H.R.G, R. Torre. arxiv:2302.12024

NFs in High Dimensions





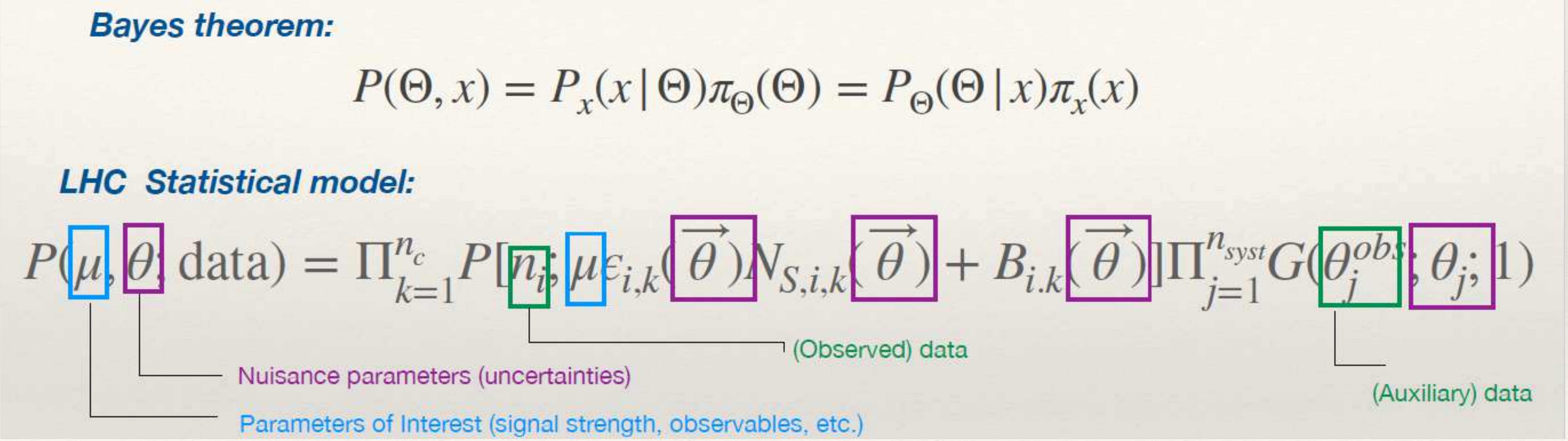
A. Coccaro, M. Letizia, H.R.G, R. Torre. arxiv:2302.12024

NFs 4 HD-HEP: LHC Likelihoods.

LHC- Likelihoods.

Likelihood functions (full statistical models) parametrise the full information of an LHC analysis; whether it is New Physics (NP) search or an SM measurement.

- Their **preservation** is a key part of the **LHC legacy.**



Example likelihoods

$$P_\Theta(\Theta \,|\, x = \text{obs})$$

LHC-like toy likelihood.

- Simplified likelihood (Multivariate-Gaussian)
- 1 parameter of interest (signal strength)
- 89 nuisance parameters.
- Ref. arXiv:1809.05548

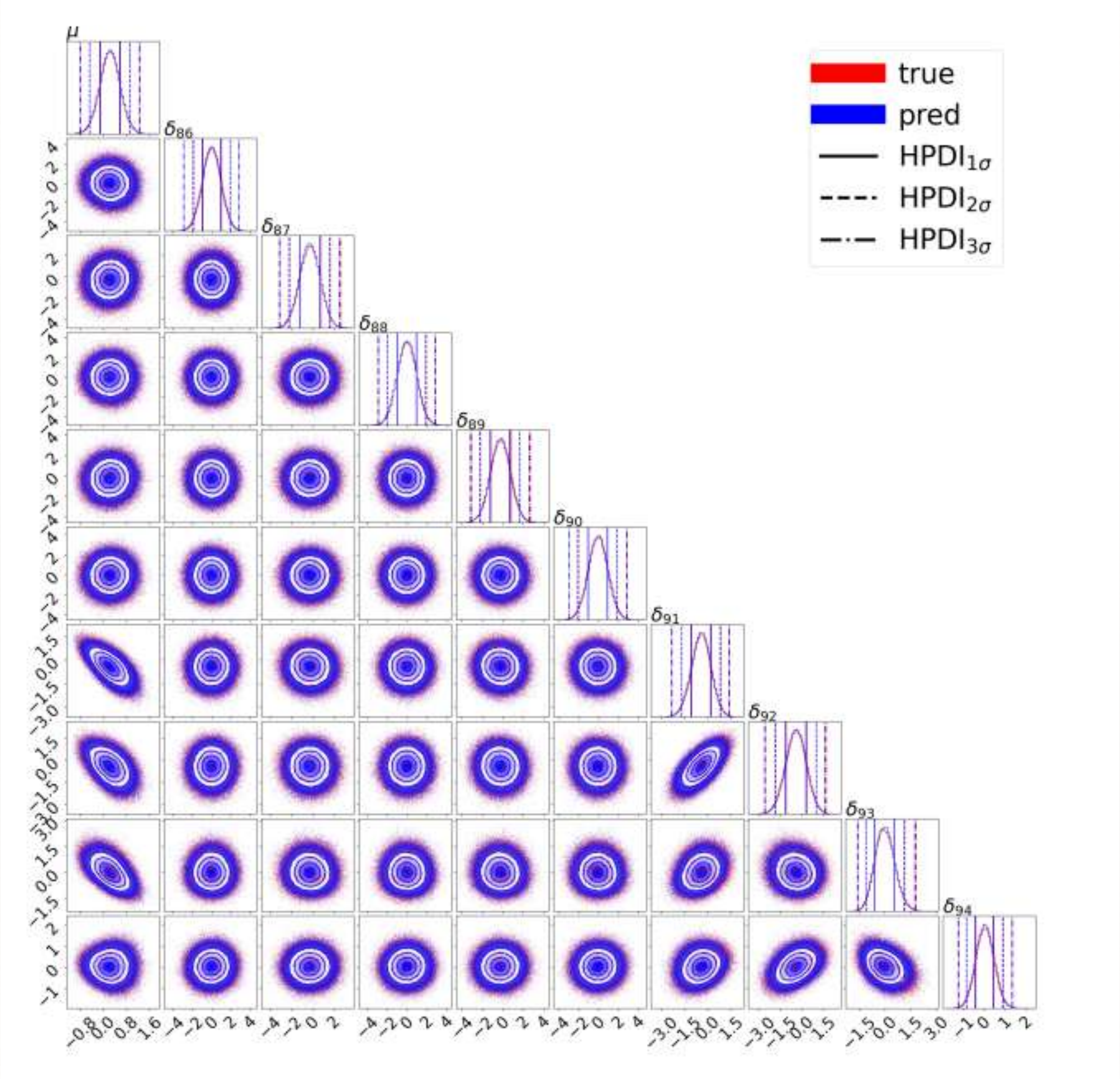
Flavor fit likelihood

- Flavor observables related to
- 12 parameters of interest (Wilson coefficients)
- 77 nuisance parameters.
- Ref. arXiv:1809.05548

ElectroWeak fit Likelihood

- EW observables.
- Including recent measurements of top mass (CMS) and W mass (CDF).
- 8 parameters of interest (Wilson coefficients of SMEFT operators)
- 32 nuisance parameters.
- Ref. arXiv:2204.04204

LHC-like toy likelihood



Hyperparameters for Toy Likelihood

# of samples	hidden layers	algorithm	# of bijec.	spline knots	range	L1 factor	patience	max # of epochs
$\mathbf{2 \cdot 10^5}$	3×64	MAF	2	-	-	0	20	200

Table 1: Hyperparameters leading to the best determination of the Toy Likelihood.

Results for Toy Likelihood

# of samples	Mean KS-test	Mean SWD	HPDIe$_{1\sigma}$	HPDIe$_{2\sigma}$	HPDIe$_{3\sigma}$	time (s)
$\mathbf{2 \cdot 10^5}$	$0.4893 \pm .0292$	$0.03947 \pm .0019$	0.02073	0.01207	0.01623	133

Table 2: Best results obtained for the Toy Likelihood.

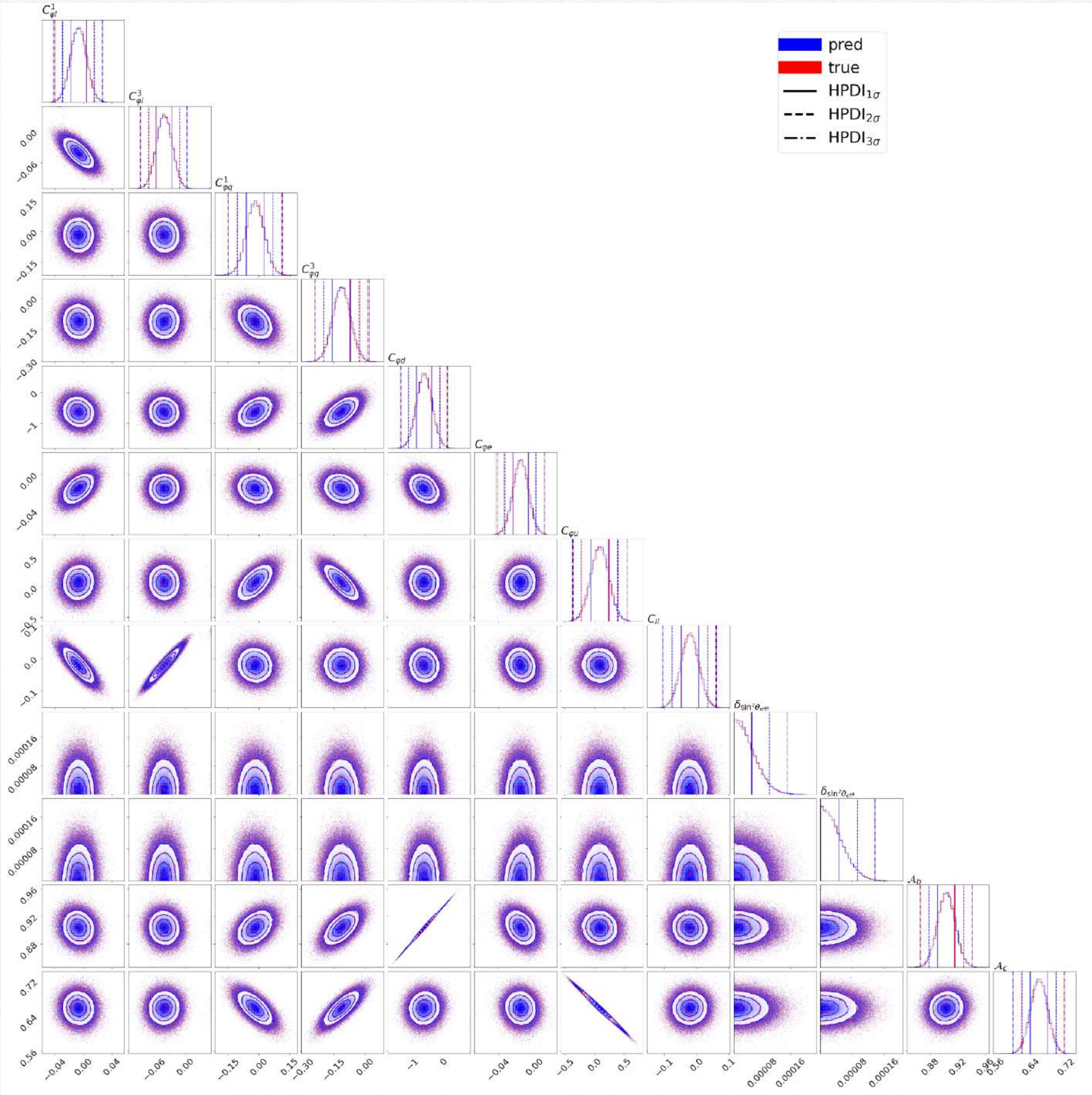
Results for Toy Likelihood POI

POI	KS-test	HPDIe$_{1\sigma}$	HPDIe$_{2\sigma}$	HPDIe$_{3\sigma}$
μ	0.54	0.02742	0.01359	0.01786

Table 3: Results for the POI in the Toy Likelihood.

R. Torre, HRG. arxiv:2309.09743

EW-fit likelihood



Hyperparameters for the EW Likelihood								
# of samples	hidden layers	# of bijec.	algorithm	spline knots	range	L1 factor	patience	# of epochs
$2 \cdot 10^5$	2	3×128	A-RQS	4	-6	0	20	800

Table 4: Hyperparameters leading to the best determination of the EW Likelihood.

Results for the EW Likelihood						
# of samples	Mean KS-test	Mean SWD	HPDIe$_{1\sigma}$	HPDIe$_{2\sigma}$	HPDIe$_{3\sigma}$	time (s)
$2 \cdot 10^5$	0.4307 ± 0.06848	0.003131 ± 0.00053	0.000339	0.0008664	0.006973	7255

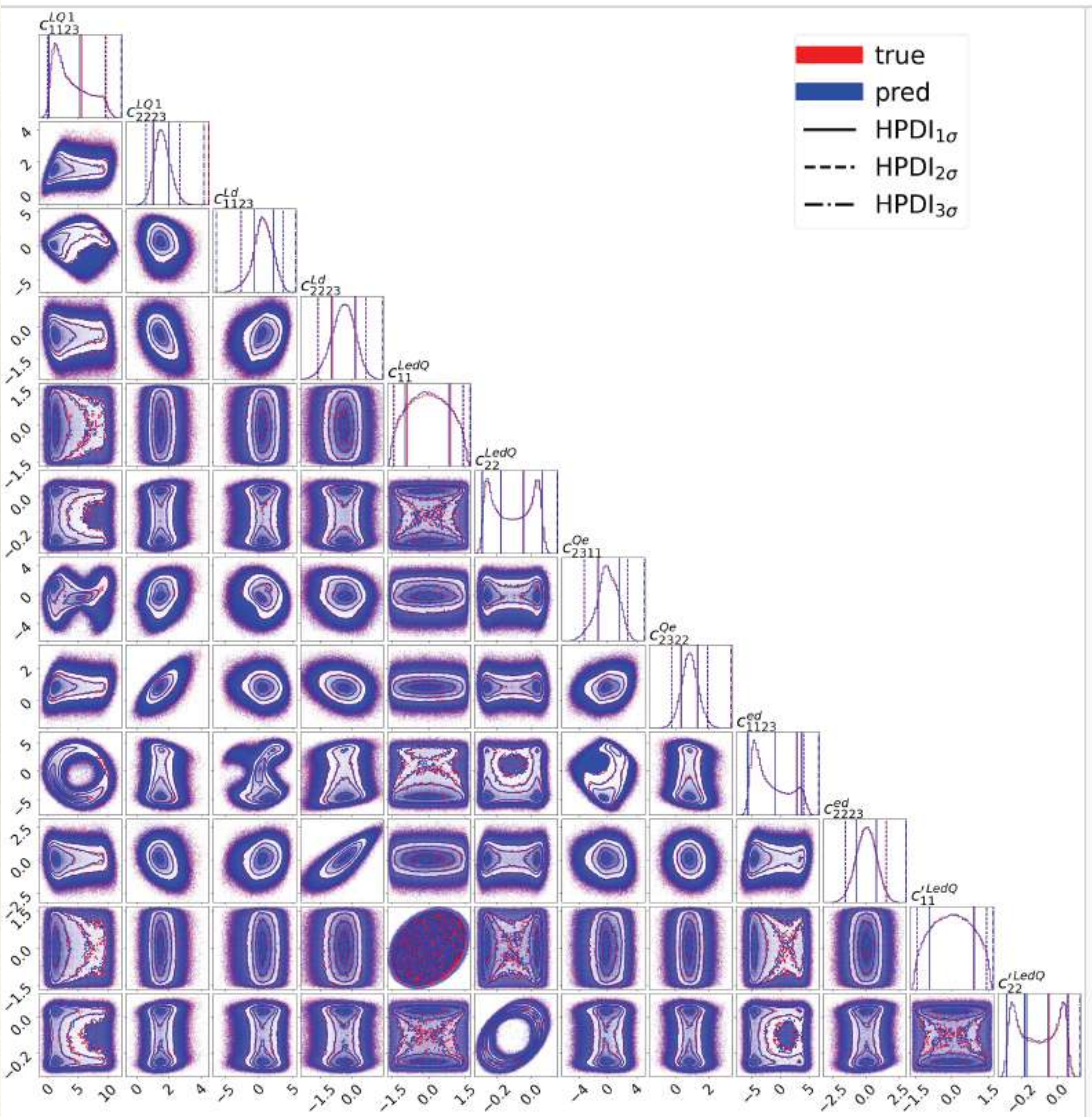
Table 5: Best results obtained on the EW Likelihood.

Results for EW Likelihood				
POI	KS-test	HPDIe$_{1\sigma}$	HPDIe$_{2\sigma}$	HPDIe$_{3\sigma}$
$c^1_{\varphi l}$	0.1901	0.08384	0.09787	0.437
$c^3_{\varphi l}$	0.2078	0.0346	0.1039	0.4967
$c^1_{\varphi q}$	0.4581	0.02279	0.01131	0.04866
$c^3_{\varphi q}$	0.4989	0.01219	0.01439	4.1017
$c_{\varphi d}$	0.5221	0.01713	0.03808	0.09952
$c_{\varphi e}$	0.4885	0.01453	0.2146	0.1401
$c_{\varphi u}$	0.5259	0.005409	0.005082	0.341
c_{ll}	0.2193	0.1667	0.08047	0.0713

Table 6: Results for the Wilson coefficients in the EW Likelihood.

R. Torre, HRG. arxiv:2309.09743

Flavor likelihood



Hyperparameters for the Flavor Likelihood								
# of samples	hidden layers	# of bijec.	algorithm	spline knots	range	L1 factor	patience	max # of epochs
10^6	3×1024	2	A-RQS	8	-5	1e-4	50	12000

Table 7: Hyperparameters leading to the best determination of the Flavor Likelihood.

Results for the Flavor Likelihood						
# of samples	Mean KS-test	Mean SWD	HPDIe$_{1\sigma}$	HPDIe$_{2\sigma}$	HPDIe$_{3\sigma}$	time (s)
$5 \cdot 10^5$	0.4237 ± 0.03405	0.02717 ± 0.002374	0.00867	0.007346	1.419e-07	9550

Table 8: Best results obtained for the Flavor Likelihood.

Results for Flavor Likelihood POIs				
POI	KS-test	HPDIe$_{1\sigma}$	HPDIe$_{2\sigma}$	HPDIe$_{3\sigma}$
$c^{LQ\,1}_{1123}$	0.4346	0.007251	1.83e-05	4.731e-08
$c^{LQ\,1}_{2223}$	0.4736	0.01249	0.00162	0.03575
c^{Ld}_{1123}	0.486	0.01466	0.006628	0.002338
c^{Ld}_{2223}	0.4138	0.0513	0.02446	2.398e-08
c^{LedQ}_{11}	0.5362	0.00738	0.004683	5.387e-08
c^{LedQ}_{22}	0.5161	0.02799	0.001639	2.155e-09
c^{Qe}_{2311}	0.4476	0.01389	0.007458	1.419e-07
c^{Qe}_{2322}	0.382	0.02132	0.02496	0.0004609
c^{ed}_{1123}	0.4789	0.04076	0.00333	5.602e-08
c^{ed}_{2223}	0.4436	0.008685	0.016	1.502e-08
$c'^{\,LedQ}_{11}$	0.3203	0.09194	0.007041	8.011e-08
$c'^{\,LedQ}_{22}$	0.4157	0.03001	0.008749	4.374e-08

Table 9: Results for the Wilson coefficients in the Flavor Likelihood.

R. Torre, HRG. arxiv:2309.09743

NFs: LHC- Likelihoods.

R. Torre, HRG. arxiv:2309.09743

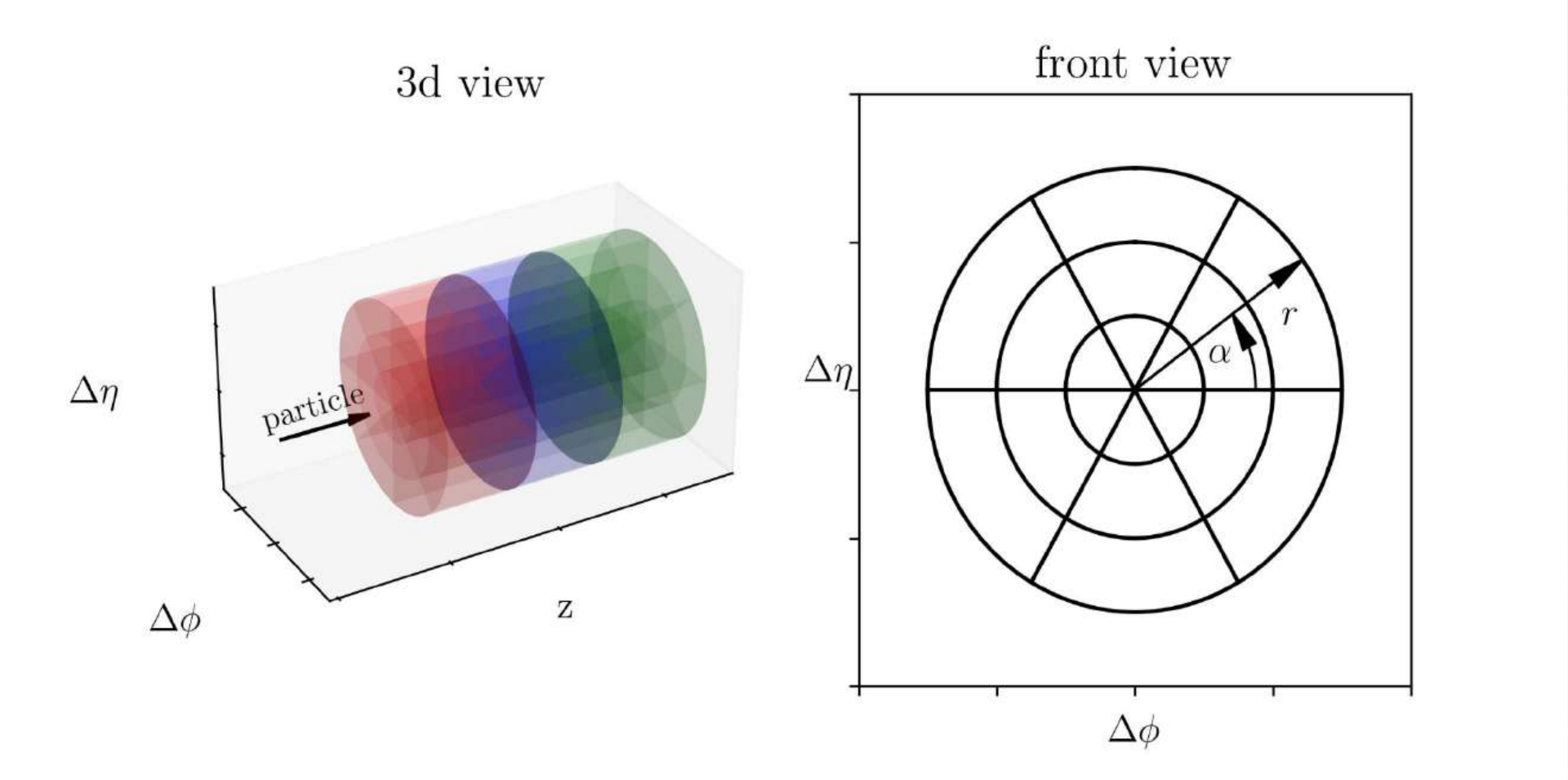
NFs 4 HD-HEP: Calorimeter Showers.

Normalizing Flows Calorimeter shower simulation

Calorimeters are detectors used to measure the energy of particles that pass through them. In order to validate and optimize the performance of calorimeters, simulation studies are conducted.

- **The Fast Calorimeter simulation challenge***

"The purpose of this challenge is to spur the development and benchmarking of fast and high-fidelity calorimeter shower generation using deep learning methods. Currently, generating calorimeter showers of interacting particles (electrons, photons, pions, ...) using GEANT4 is a major computational bottleneck at the LHC, and it is forecast to overwhelm the computing budget of the LHC experiments in the near future."



*To appear on ArXiV soon

The datasets

Each dataset has the same general format. The detector geometry consists of concentric cylinders with particles propagating along the z-axis. The detector is segmented along the z-axis into discrete layers. Each layer has bins along the radial direction and some of them have bins in the angle α. The coordinates Δϕ and Δη correspond to the x- and y axis of the cylindrical coordinates. The events are conditioned by the incident energy.

Dataset 1

- Separated in two parts: Photon and Pion showers.
- 15 incident energies from 256 MeV up to 4 TeV
- 368 voxels (in 5 layers) for photons and 533 (in 7 layers) for pions.

Dataset 2

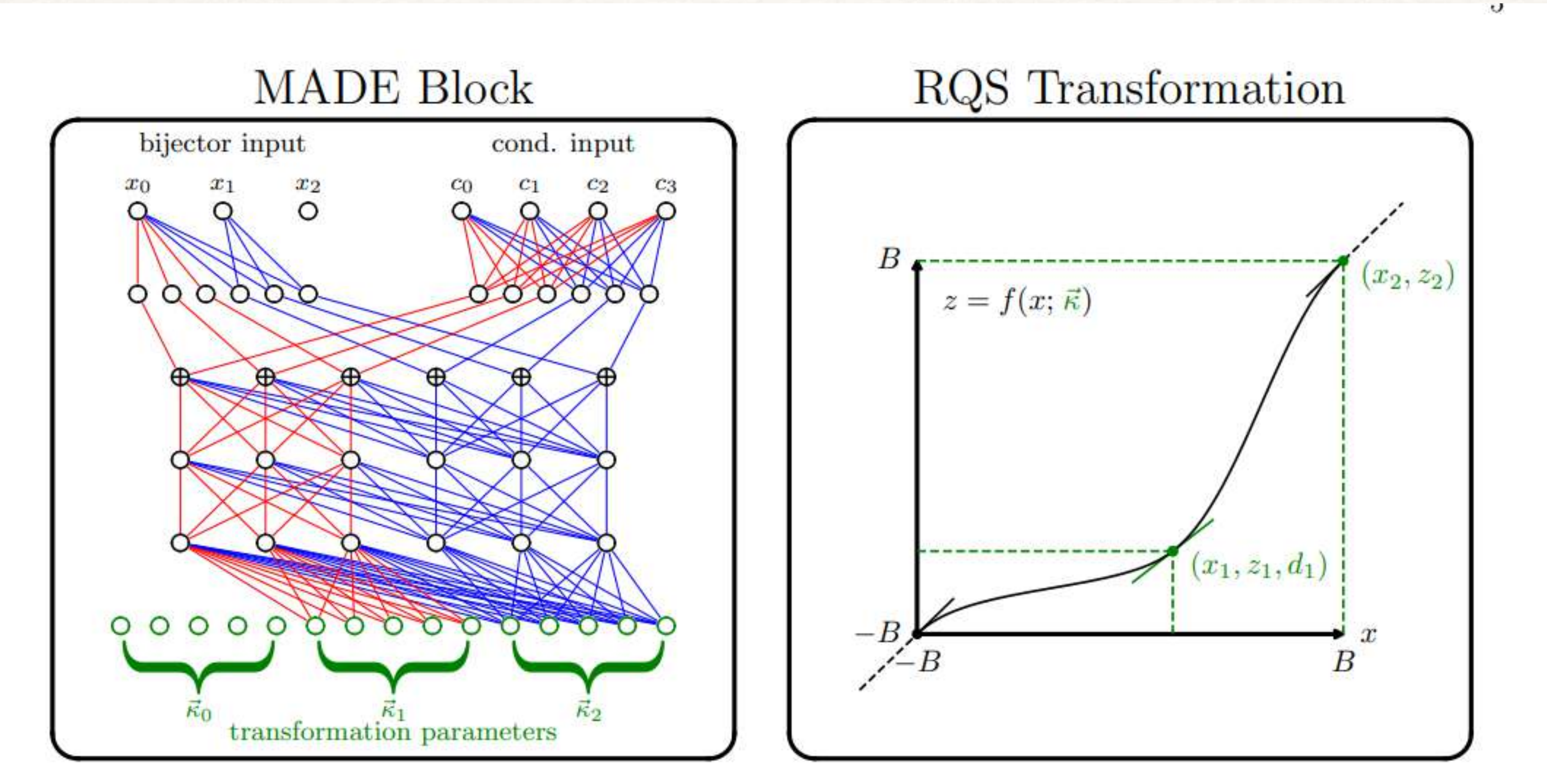
- Electron showers.
- Energies sampled from a log-uniform from 1 GeV to 1 TeV.
- 45x16x9 =6480 uniform voxels.

Dataset 3

- Electron showers.
- Energies sampled from a log-uniform from 1 GeV to 1 TeV.
- 45x50x18=40500 uniform voxels.

Normalizing Flows Calorimeter shower simulation

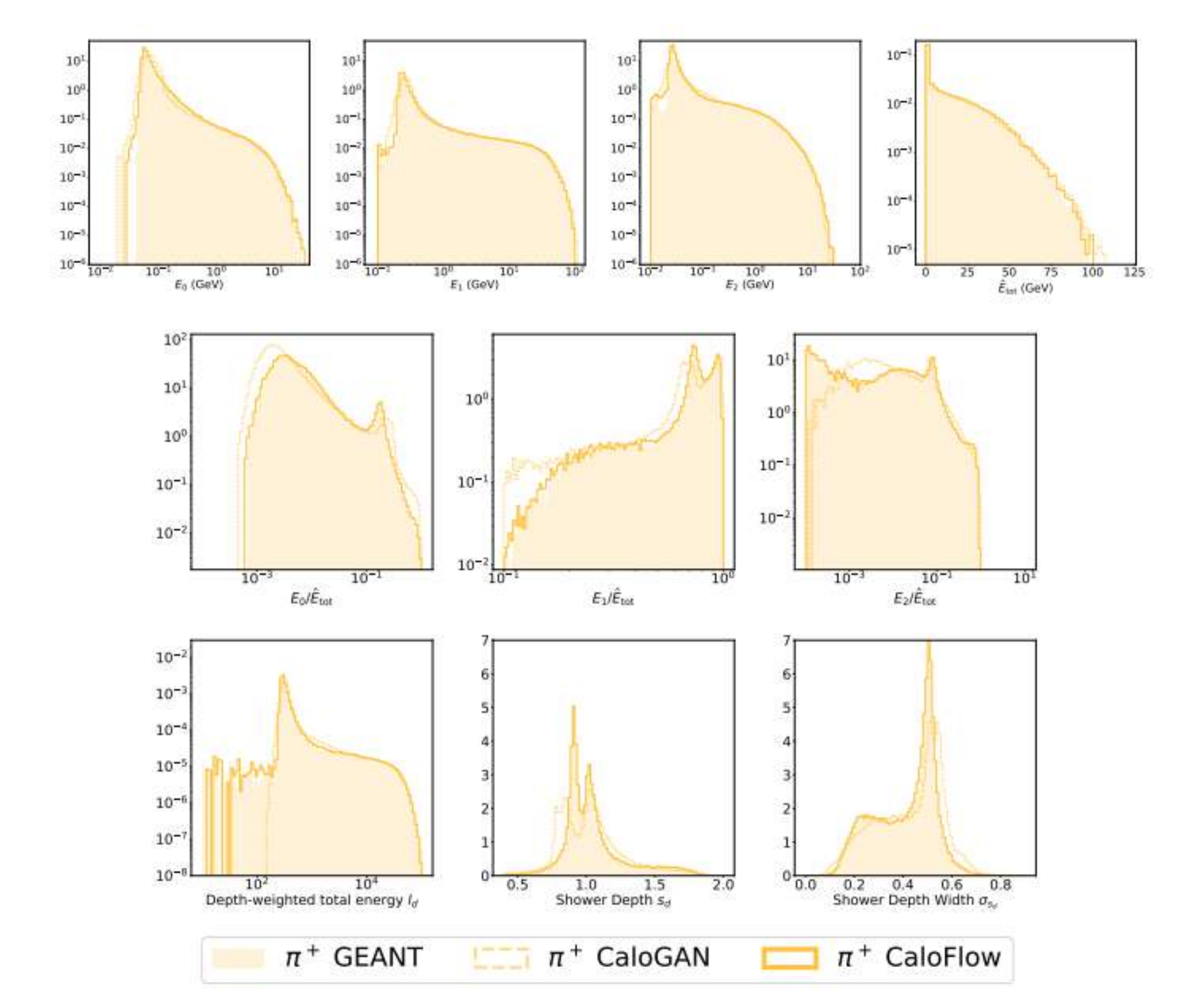
CaloFlow



C. Krause, D. Shih arxiv:2106.05285, arxiv:2110.11377

Normalizing Flows Calorimeter shower simulation

CaloFlow

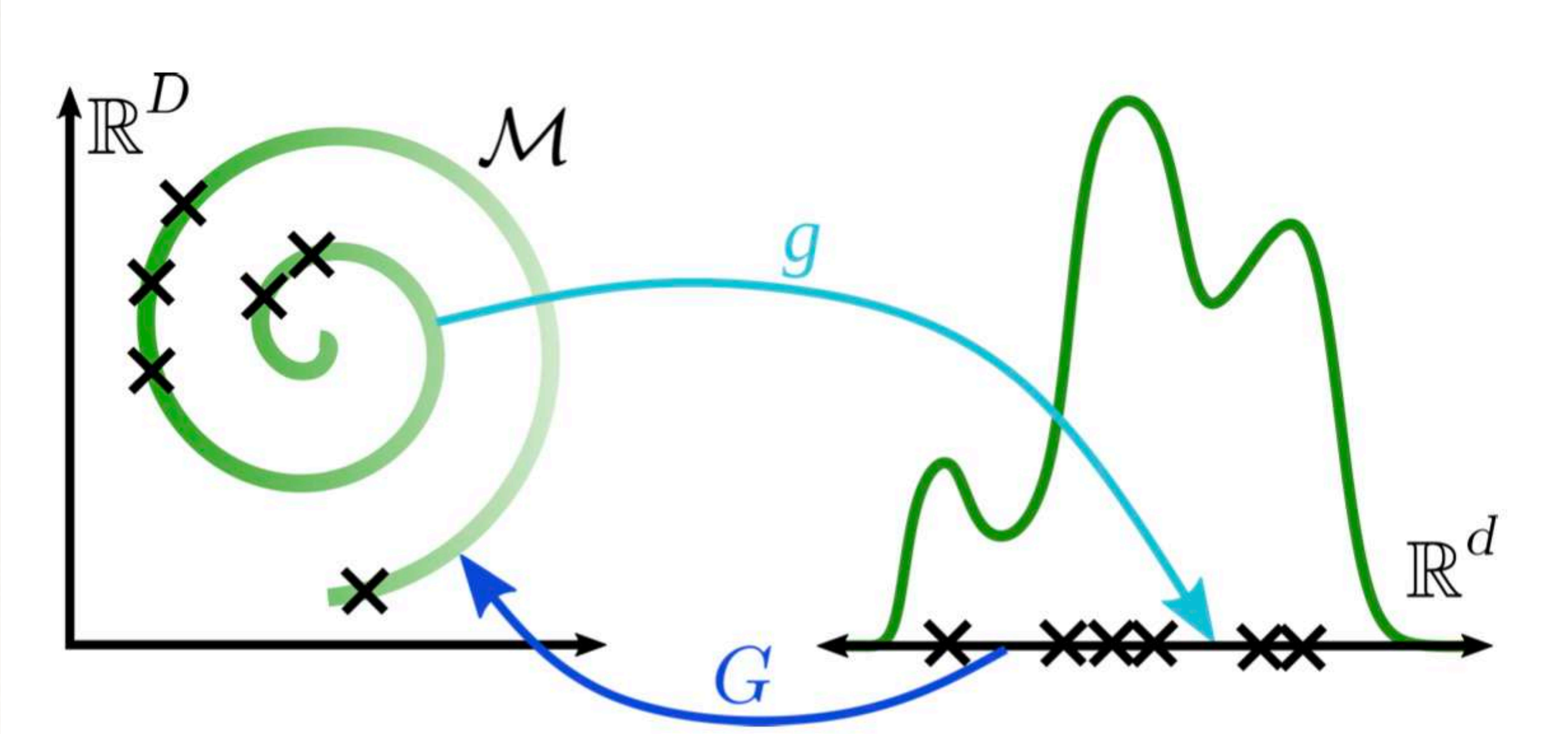


Dataset 1

C. Krause, D. Shih arxiv:2106.05285, arxiv:2110.11377

Normalizing Flows in the latent space.

- NFs are very expressive generative networks even in HD.
- However, they are bijective functions: The size of the model scales with the dimensionality of data. How can we solve this? -> Mapping data to lower dimensional manifolds.
- In Machine Learning, this is known as the *manifold hypothesis* from machine learning, which states that high- dimensional data is supported on low-dimensional manifolds.
- Our assumption is that the seemingly high-dimensional structure of calorimeter showers, can be described by simpler physical laws.
- We propose to model calorimeter showers in two steps: To first learn learn a mapping to a lower dimensional manifold to then perform density on the manifold.



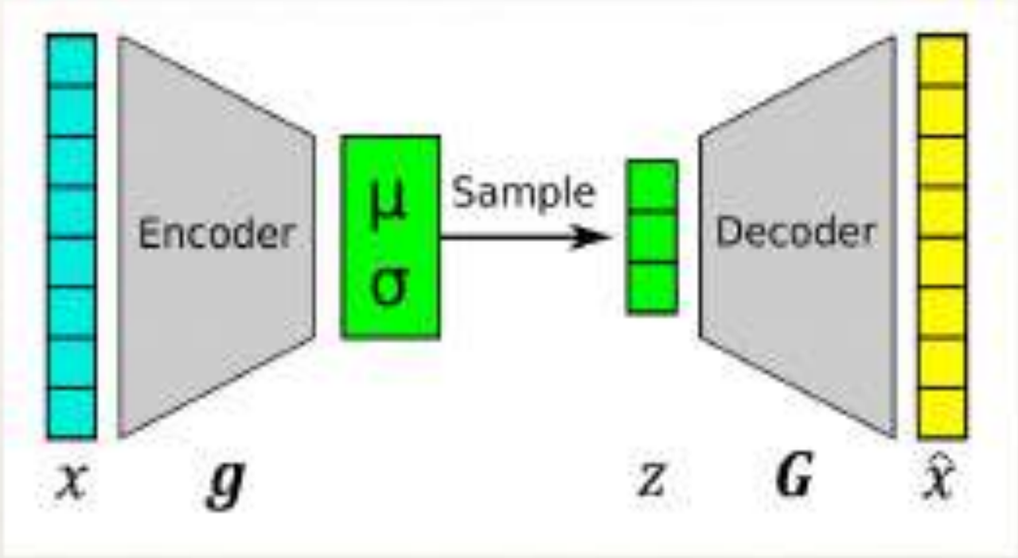
arXiv:1708.02949

Normalizing Flows calorimeter shower simulation

CaloMan

STEP 1: Learn $\mathscr{M}$ with a generalized autoencoder.
This may be an **AE**, **VAE**, GAN, Wasserstein AE, bi-GAN, etc.

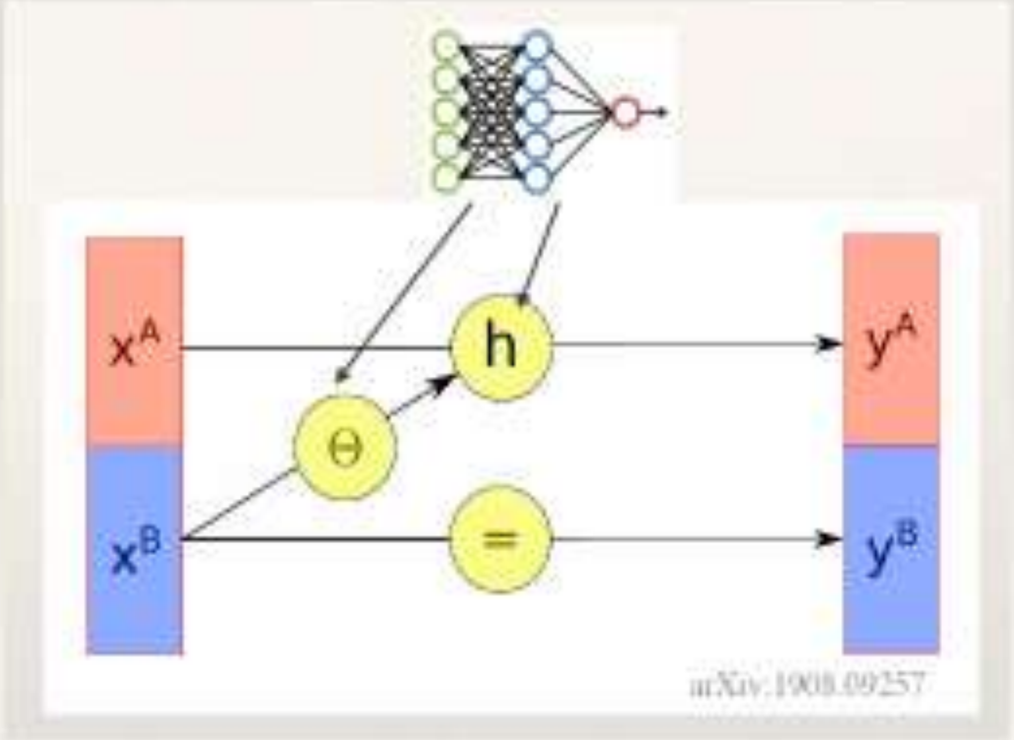
Here we use



+

STEP 2: Perform density estimation on the manifold,
with **NFs**, autoregressive, score-based, diffusion models

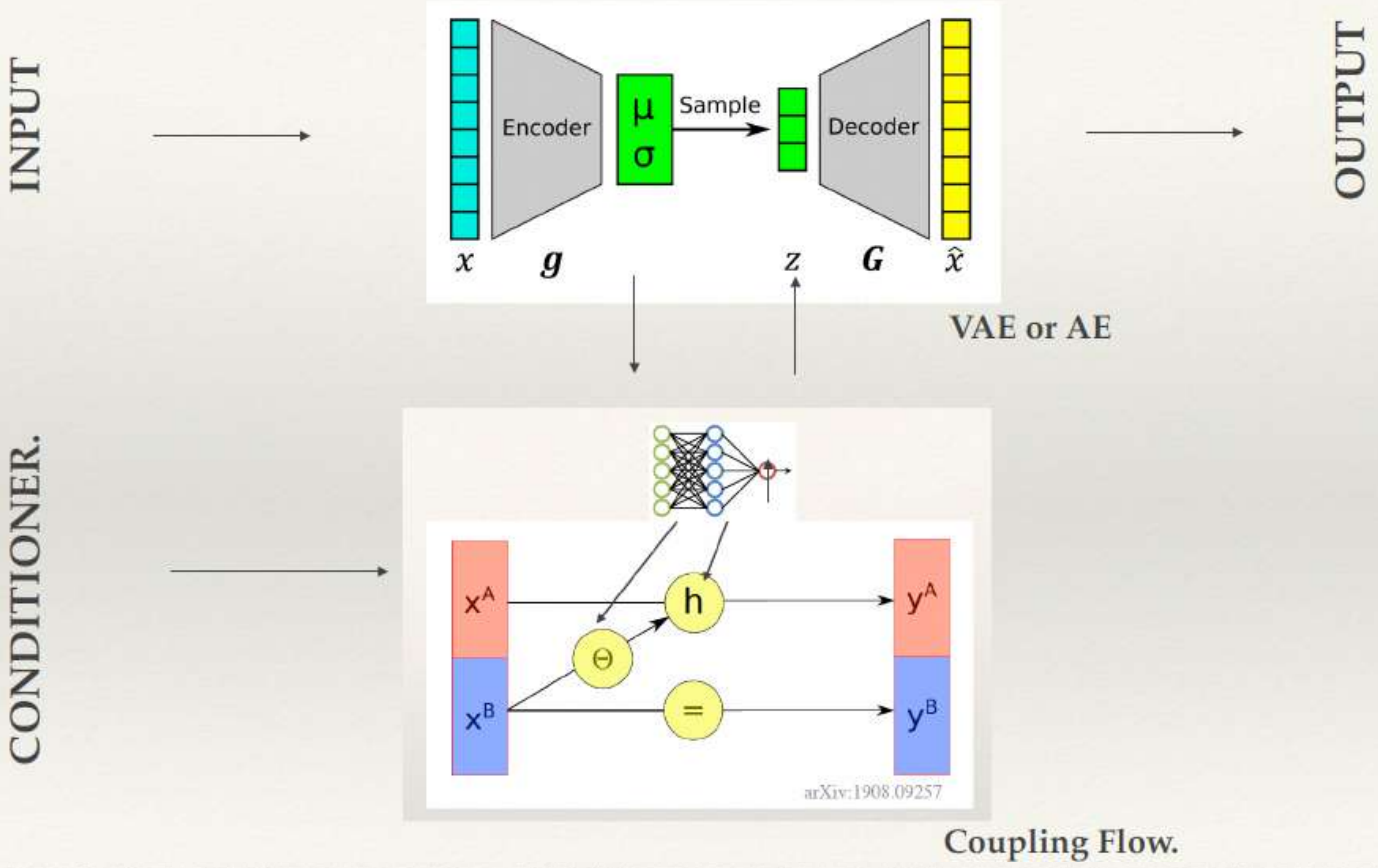
Here we use (Coupling) NFs



J. Cresswell, B. Leigh-Ross, G. Loaiza-Ganem, H.R.G., M. Letizia, A. Caterini. arxiv:2211.15380

Normalizing Flows calorimeter shower simulation

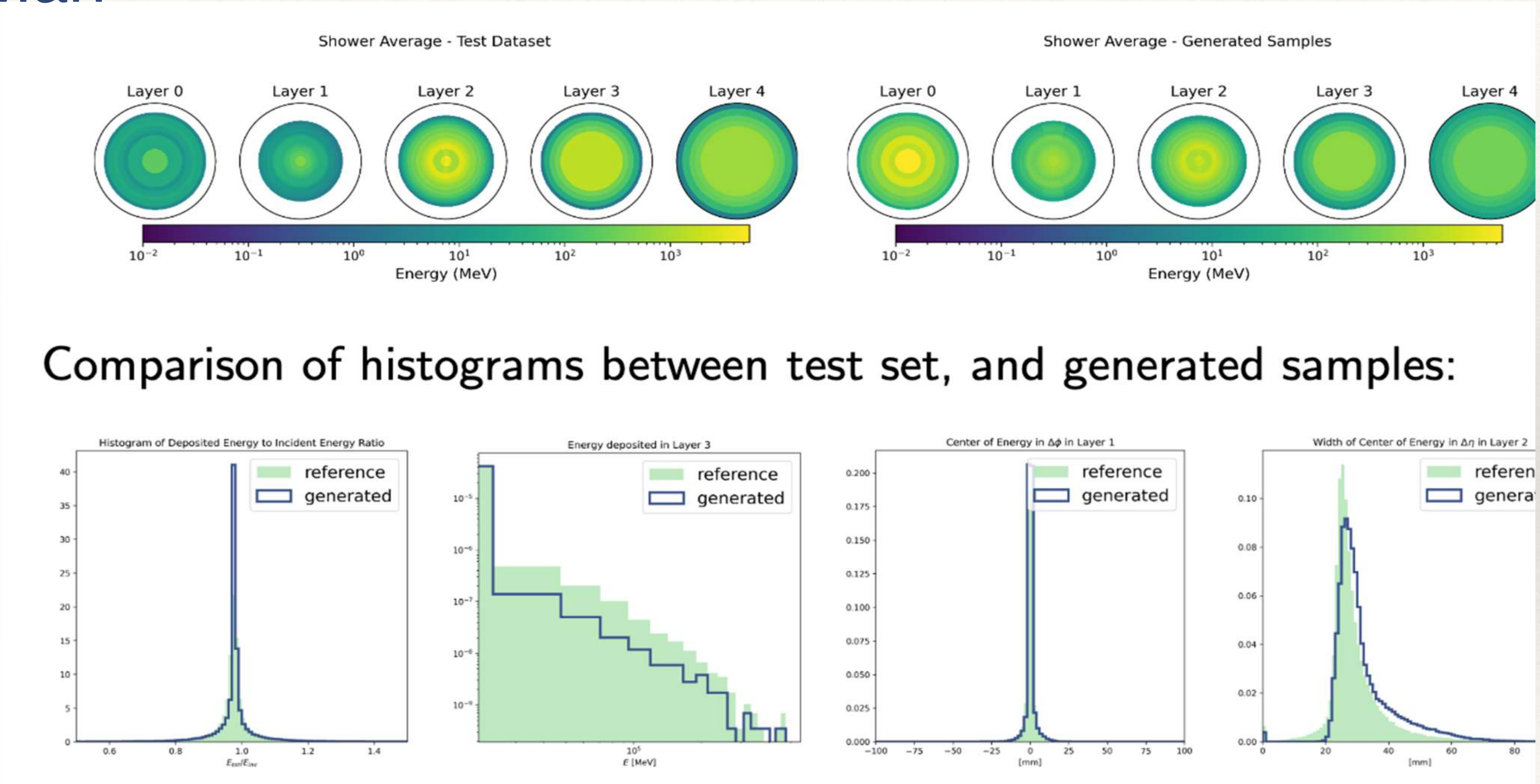
CaloMan



J. Cresswell, B. Leigh-Ross, G. Loaiza-Ganem, H.R.G., M. Letizia, A. Caterini. arxiv:2211.15380

Normalizing Flows calorimeter shower simulation

CaloMan



J. Cresswell, B. Leigh-Ross, G. Loaiza-Ganem, H.R.G., M. Letizia, A. Caterini. arxiv:2211.15380

Dataset 1

Requirements of ML methods for HEP.

- **Robustness:** The limits of the method in regards of complexity, range or type of data of should be clear as much as possible.

- **Scalability:** Understanding how well ML methods work as the dimensionality of the problem increases is crucial to ensure their systematic usage beyond vanilla proof of concepts.

- **Accuracy determination:** Reliable, statistically robust assessment of the level of accuracy of trained ML models is essential. This is not easy for Unsupervised learning approaches.

- **Uncertainty estimation:** ML methods should provide an estimation of the uncertainty in the prediction. For most applications in HEP, this is necessary to derive statistical sound conclusions.

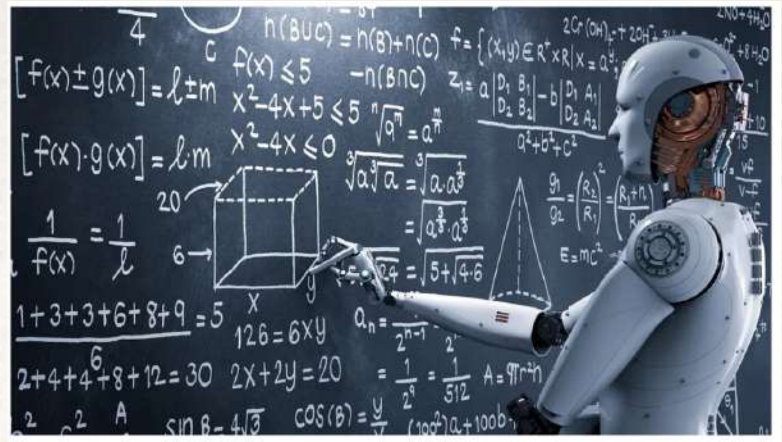
- **Reproducibility:** In science, any result and computational tool should be reproducible and openly available. This ensures cross-checking, reusability and the overall lasting legacy of the research.

- **Deployment:** Collider experiments are extremely sophisticated. Integration of ML, requires careful planning.

Conclusions

- Machine Learning is revolutionizing society and is strongly making its way into High Energy Physics.
- It will be crucial to enlarge our understanding of fundamental physics trough data intensive approaches.
- We find that Normalizing Flows are a particularly interesting example of ML methods, given its expressibility, scalability and their explicit density estimation.
- To ensure their usage ML methods should be thoroughly tested and must fulfill a number of conditions.
- The effort will require interdisciplinary research that will be fruitful for Physics and Data Science.
- The emerge of Data Physicists is imminent.

  (see: https://www.aps.org/publications/apsnews/202311/backpage.cfm)



https://towardsdatascience.com/from-theoretical-physicist-to-data-scientist-d8b781462c9

THANK YOU!

Contact information:

Email: humberto.reyes@rwth-aachen.de

Skype: humberto.reyes32

I look forward to discussion/collaboration!