

Quantum Invertible Neural Networks for Fast Calorimetry as Quantum Generative Model with high interpretability at Future Colliders

A preliminary study on quantum-INN in generative ML in ATLAS for future colliders

---

Corchia F. , Franchini M.



# Why interested in generative models?

Given a dataset, a model is trained to generate new datapoints from the same distribution.

<https://thispersondoesnotexist.com/>

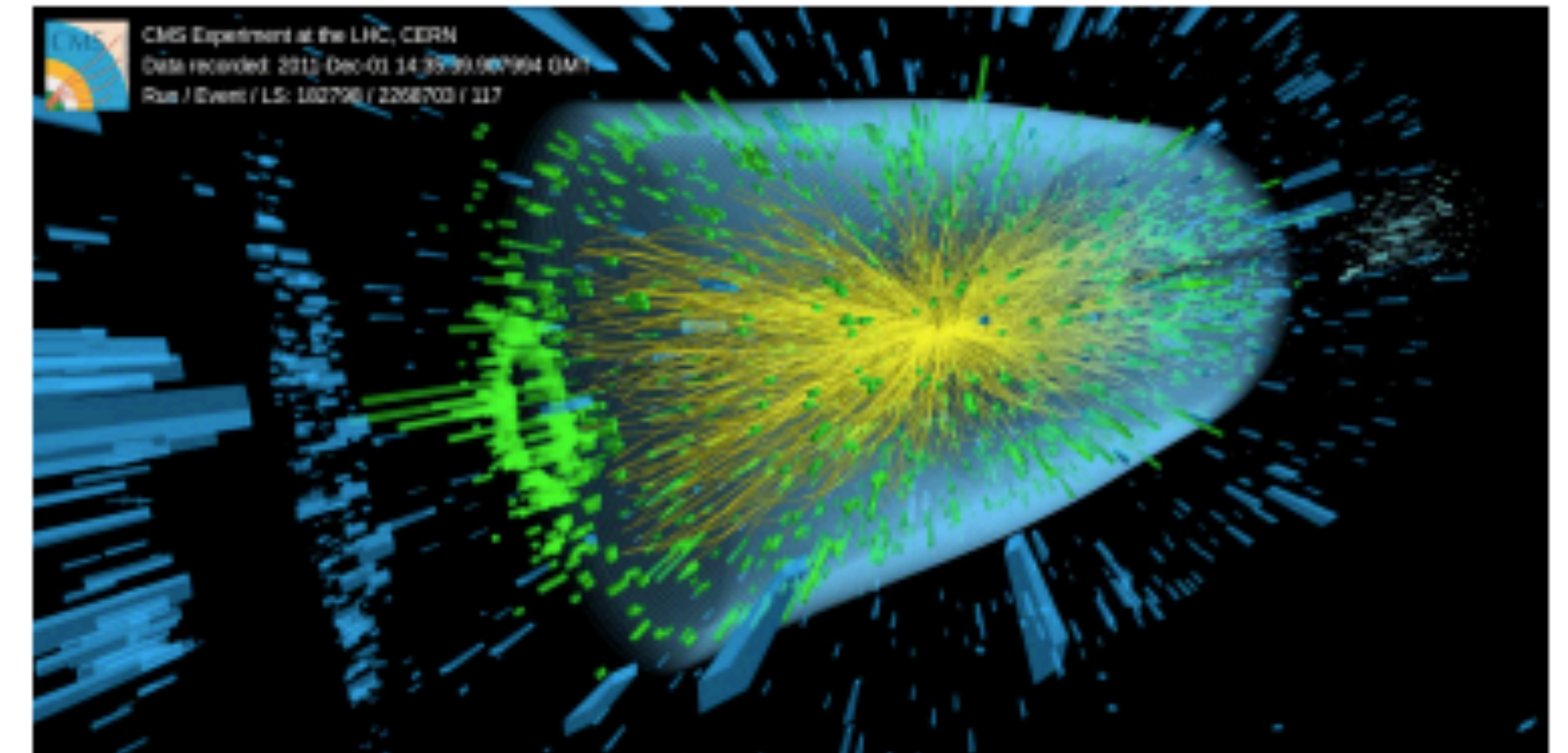


Images

**You**  
Hi, if quantum skiing was a sport what would be the goal ?  
**ChatGPT**  
In quantum skiing, the goal would be to simultaneously reach both the top and the bottom of the mountain, all while navigating every possible slope at once. But watch out for the uncertainty principle – you might end up skiing uphill and downhill at the same time!

Natural Language

<https://home.cern/>

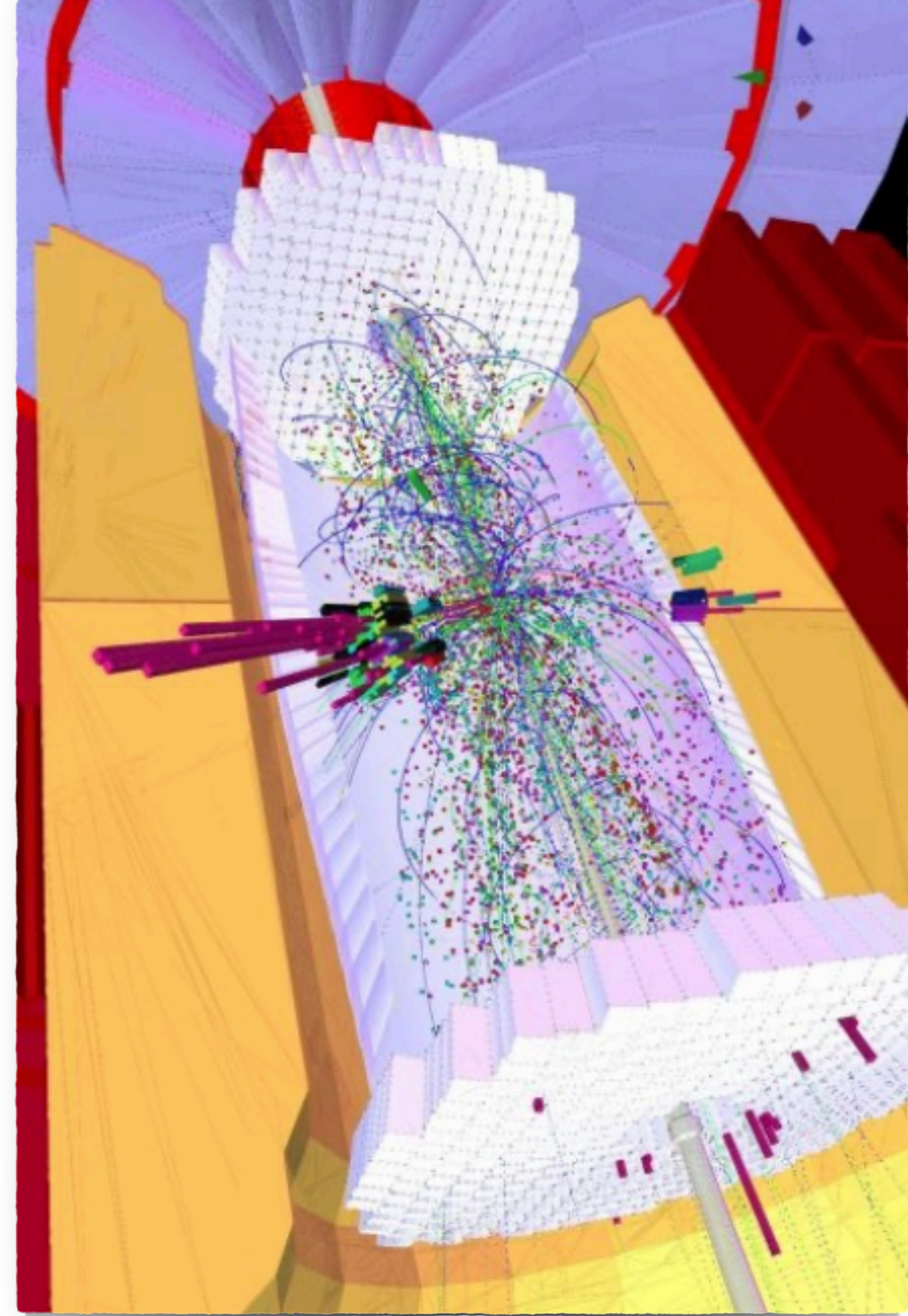


Physics experiments

- \* Focus on HEP, relies heavily for: **fast detector simulation, calorimeter shower generation**, background modelling, anomaly detection, sampling of complex physics distributions
- \* At HL-LHC (and more for **Future Colliders and Detectors**) scale, simulation and data-processing costs will become even more critical.

# Quantum Generative NN

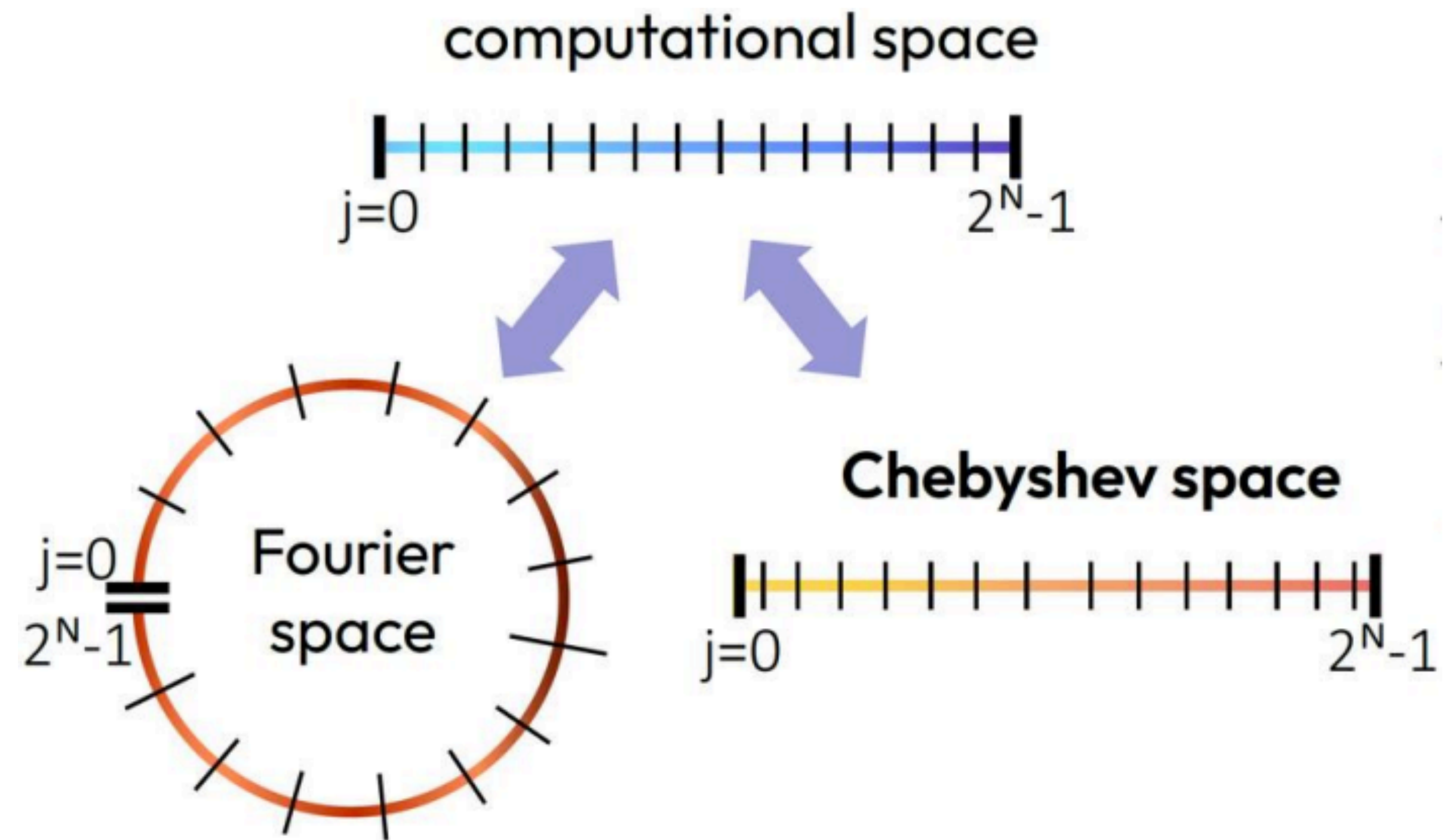
- \* The size of classic NN architectures reaching the boundary of computational feasibility.
- \* Gen-QNN are **naturally sampling machines** and may represent complex correlations compactly:
  - Superposition enables qubits to represent **multiple states simultaneously**
  - entanglement allows qubits to be **correlated** in ways that classical bits cannot.
- \* The field has produced encouraging demonstrations on:
  - low-dimensional distributions
  - **reduced calorimeter images**
  - ...
- \* Can also become more expressive, more efficient, or more scalable than strong classical baselines?



# Recent CERN IQT study

How do we develop embeddings that are hard to sample classically?  
How to build a high-performing quantum G model?

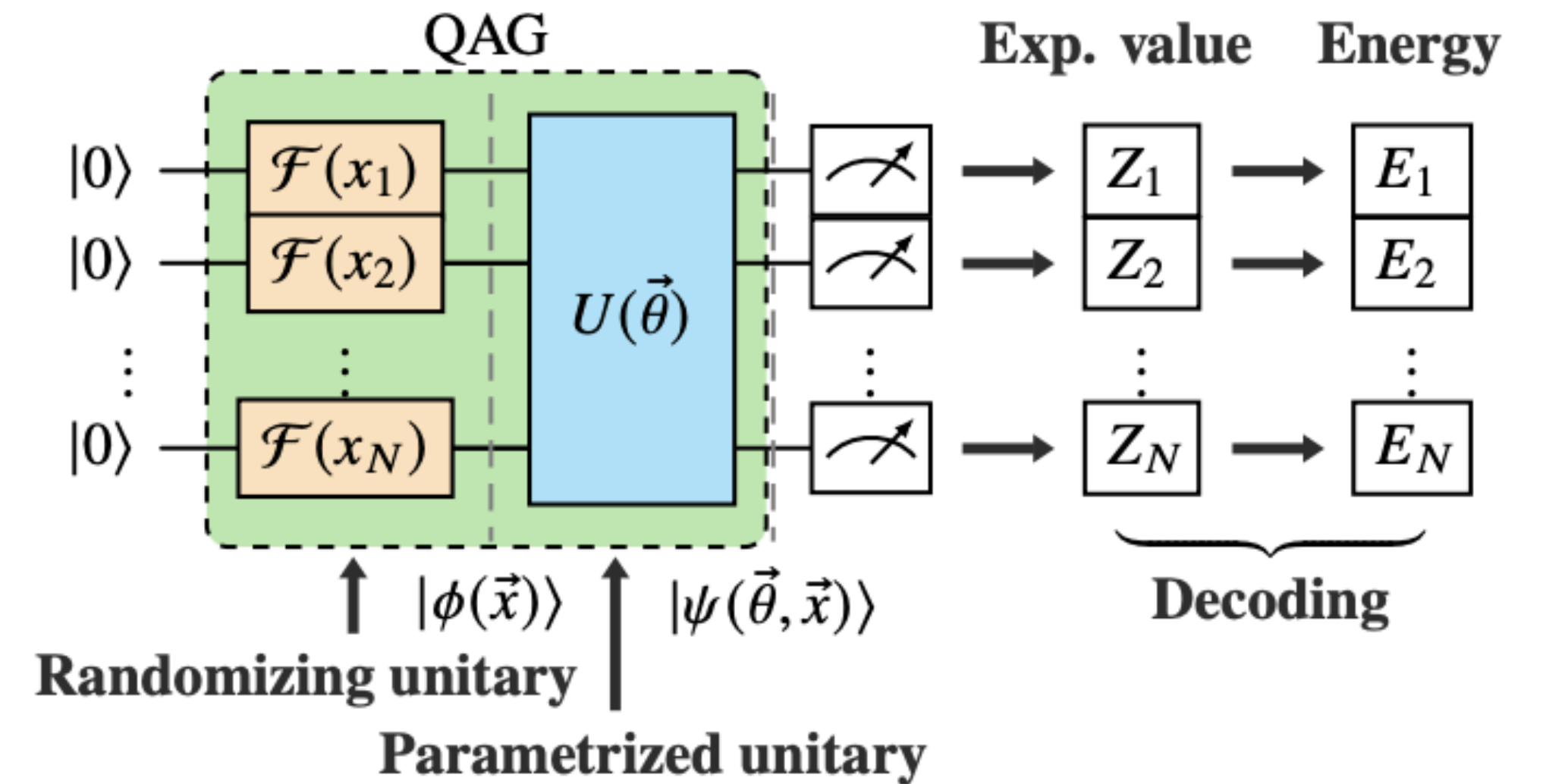
- \* Define a feature maps that can enable sampling (projective measurements) in the computational basis
- \* Model capacity depends on model expressivity that depend on ansatz structure
- \* Need to define a transformation between 2 bases that can enable efficient generative modelling (QFT)



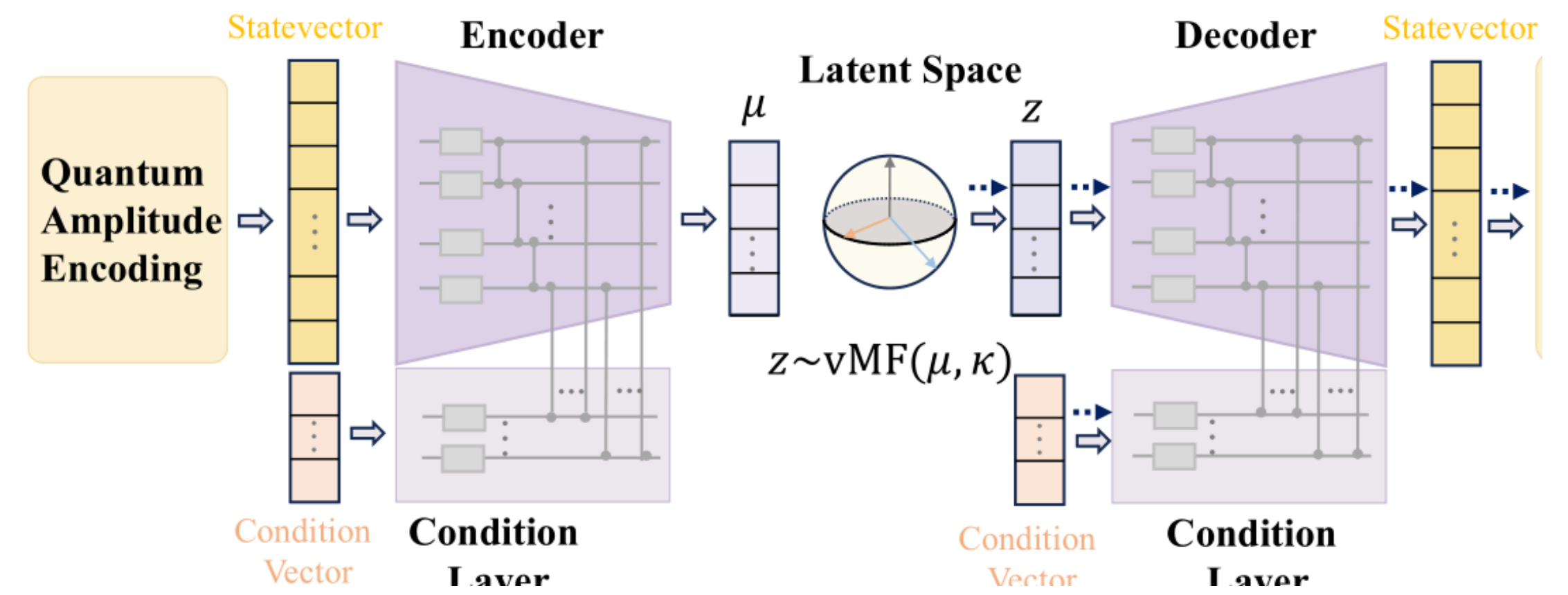
[https://agenda.infn.it/event/48148/contributions/283939/attachments/145676/221556/Grossi\\_CERN\\_QCINFN26.pdf](https://agenda.infn.it/event/48148/contributions/283939/attachments/145676/221556/Grossi_CERN_QCINFN26.pdf)

# Explored gen-QINN so far

- \* **Quantum Circuit Born Machines (QCBMs)** - Learn a probability distribution directly from measurement outcomes of a parameterized quantum circuit.
- \* **Quantum GANs / hybrid GANs** - Use a quantum generator, often with a classical discriminator.
- \* **Quantum VAEs / hybrid latent-variable models** - Quantum circuit used in latent representation or decoder/generator block.
- \* **Quantum-enhanced normalizing flows** - Hybrid designs where quantum components enrich transformations inside a classical flow.

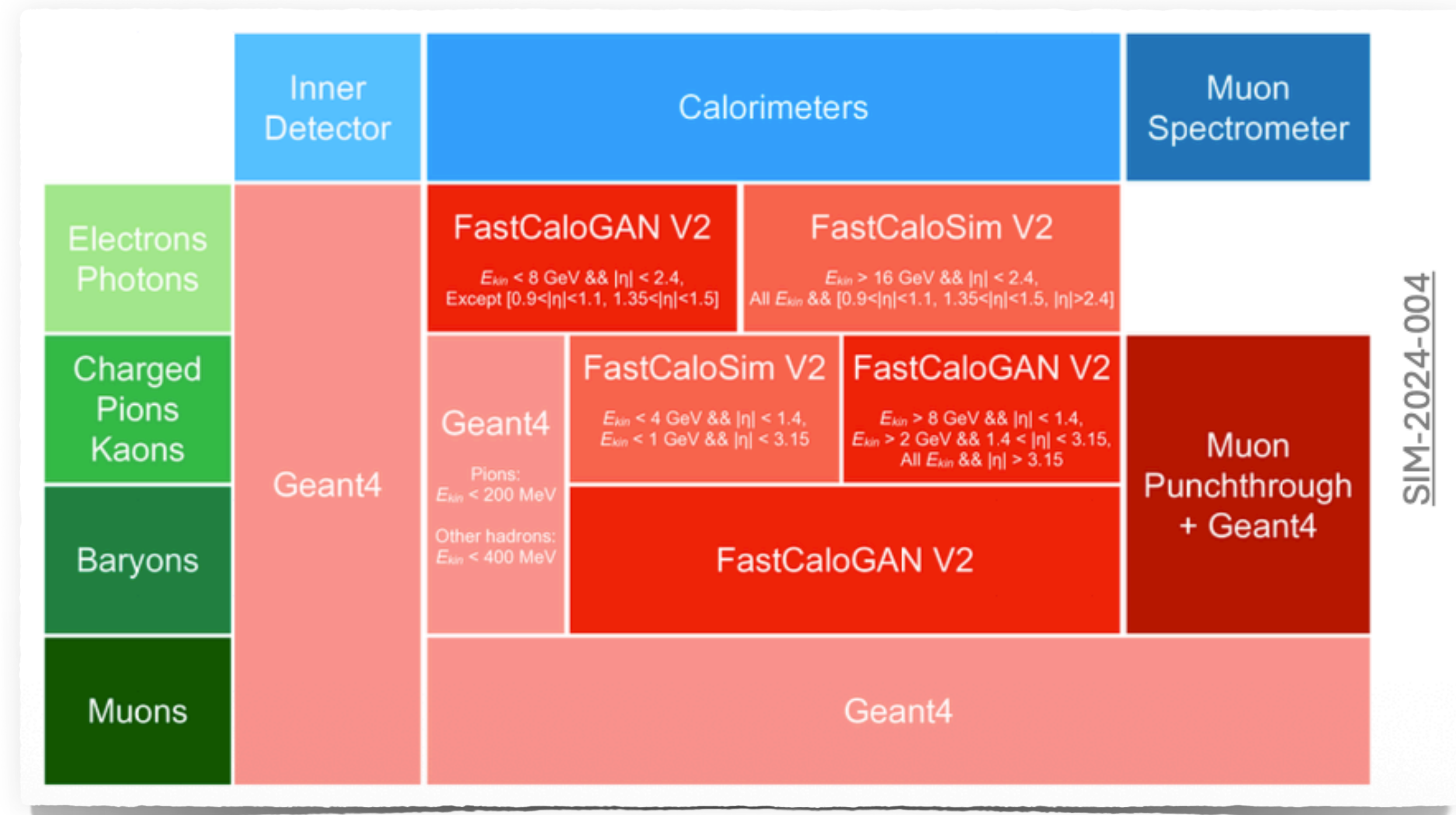


PoS(EPS-HEP2025)639



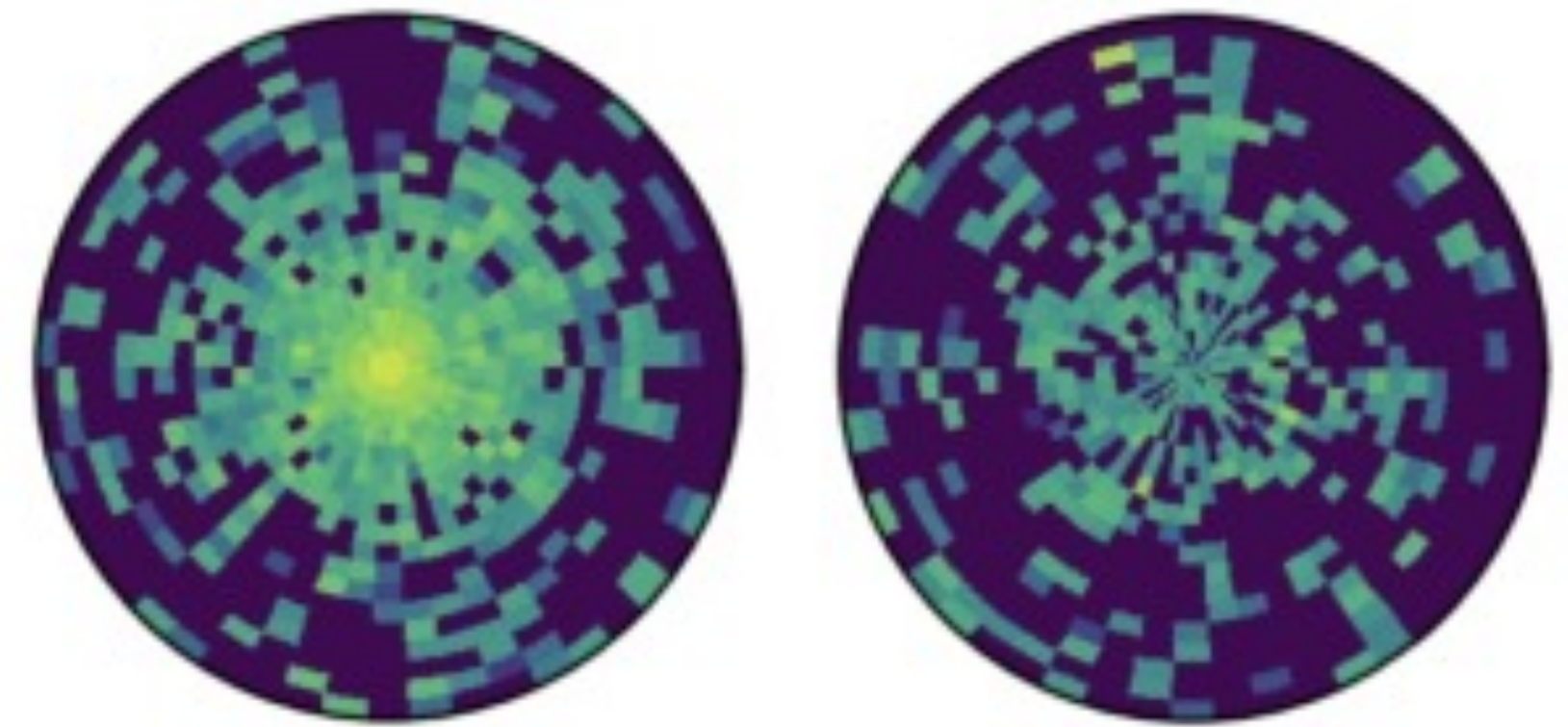
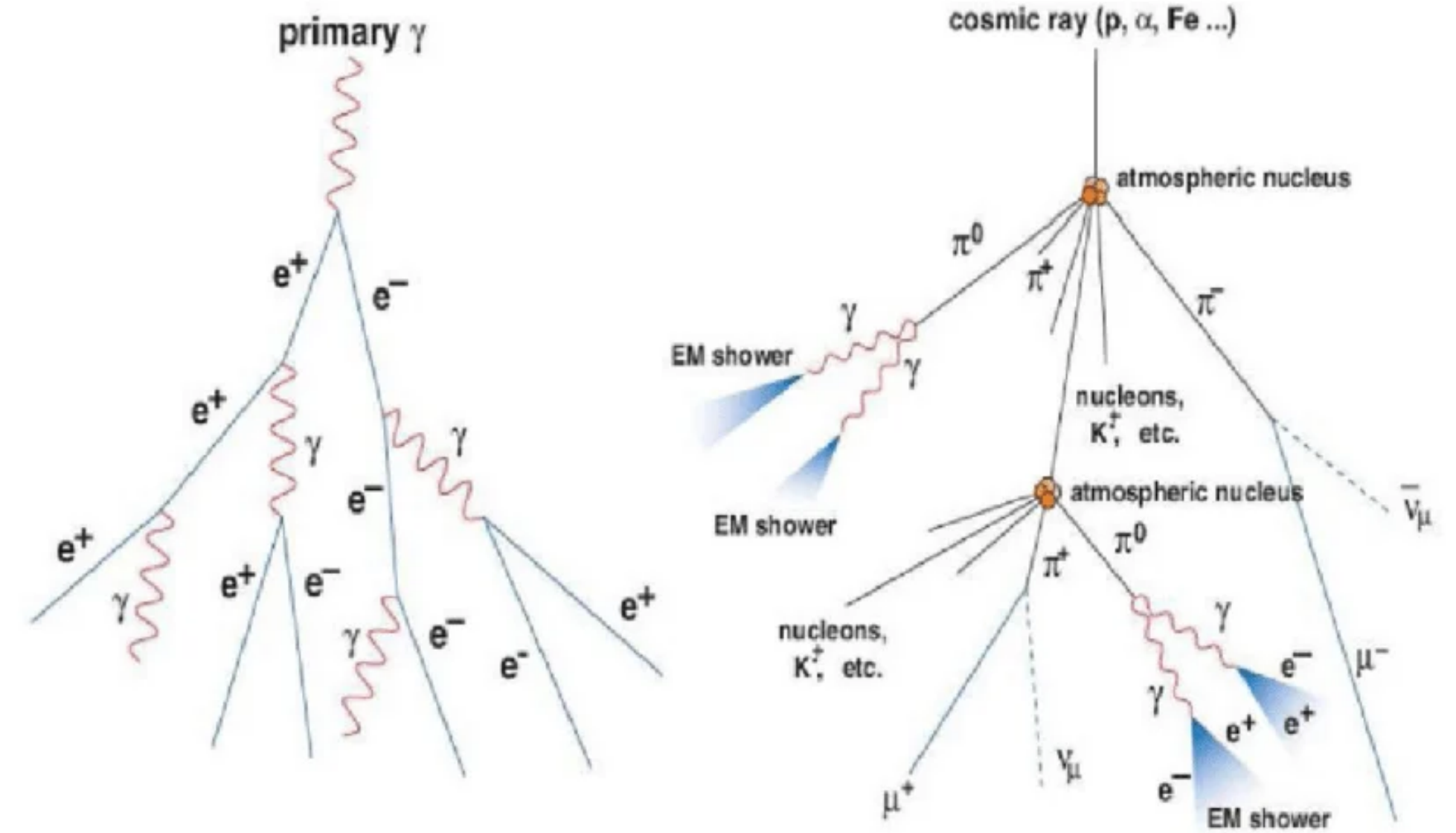
# ATLAS - Fast Calo Simulation

- \* Current model (rely on GAN or no-NN) selection issue: **ignoring the actual shower shape**, FastCaloGANV2 relies on total simulated energy
- \* Currently a big efforts to solved this issue!
- \* Implemented new normalising flows models.
  - Validated and works nicely
- \* Waiting for code integration and internal deployment for next round generation



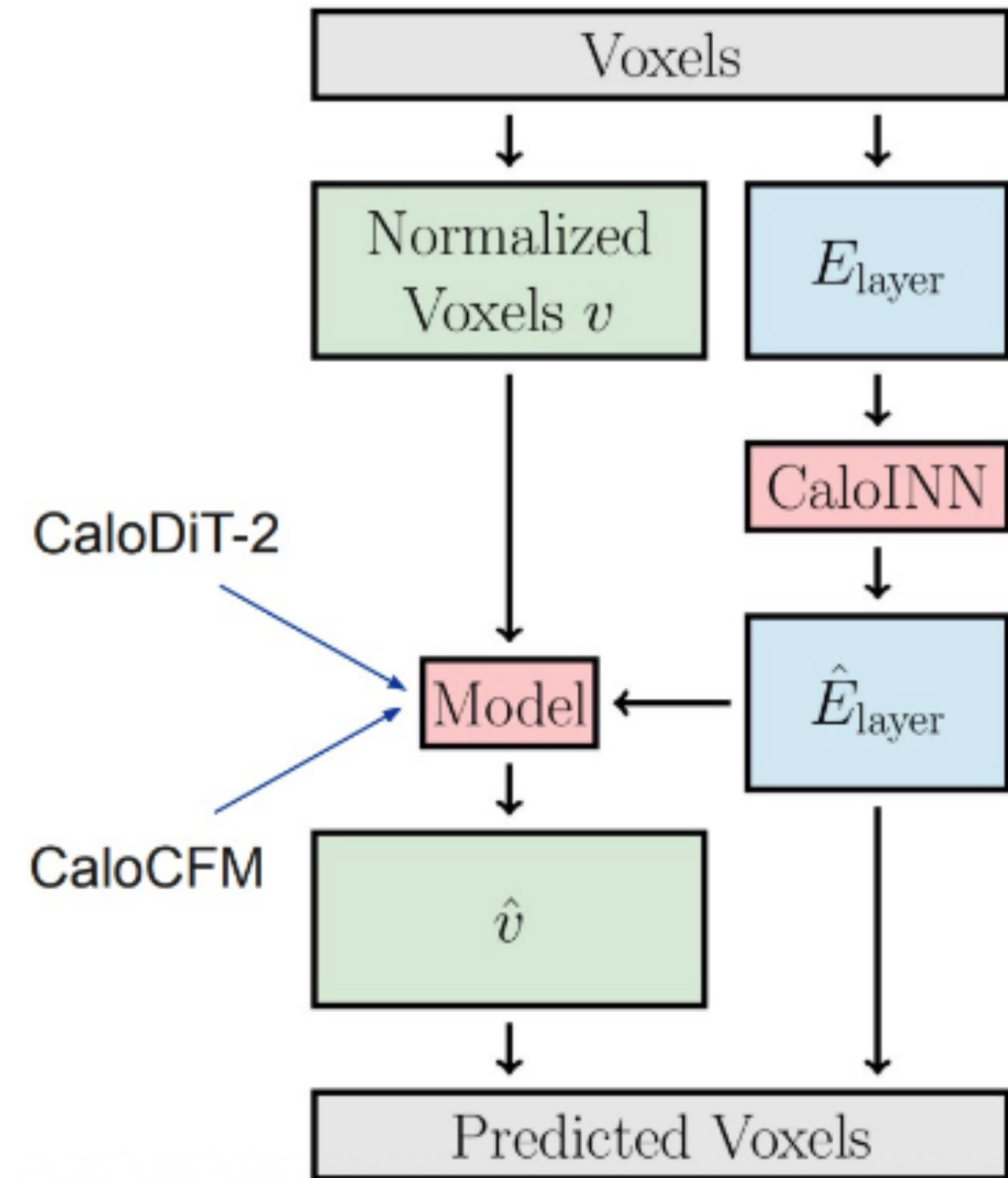
# ATLAS - Fast Calo Simulation

- \* Current model (rely on GAN or no-NN) selection issue: **ignoring the actual shower shape**, FastCaloGANV2 relies on total simulated energy
- \* Currently a big efforts to solved this issue!
- \* Implemented new normalising flows models.
  - Validated and works nicely
- \* Waiting for code integration and internal deployment for next round generation

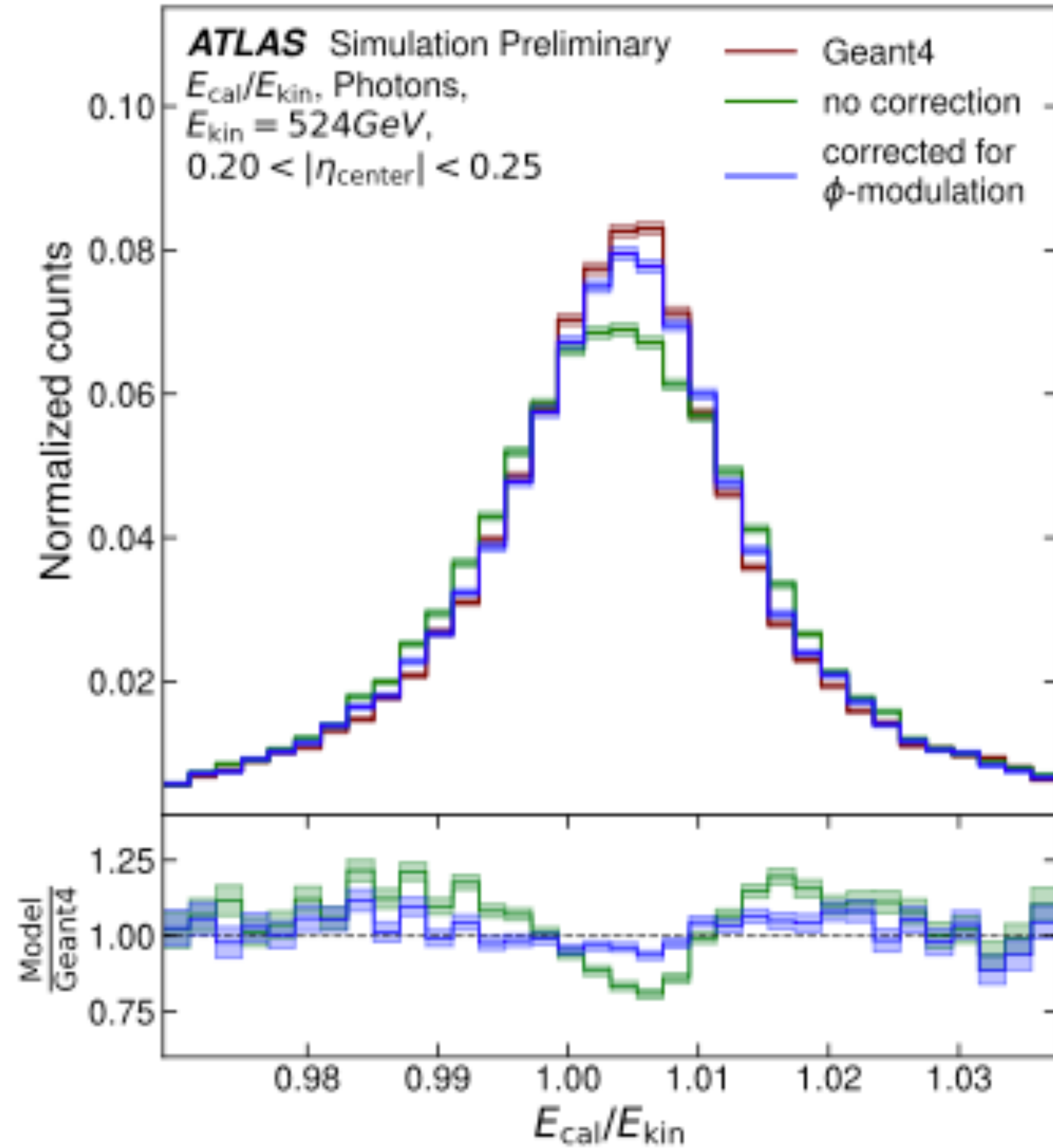


# INN in FastCalo-Sim

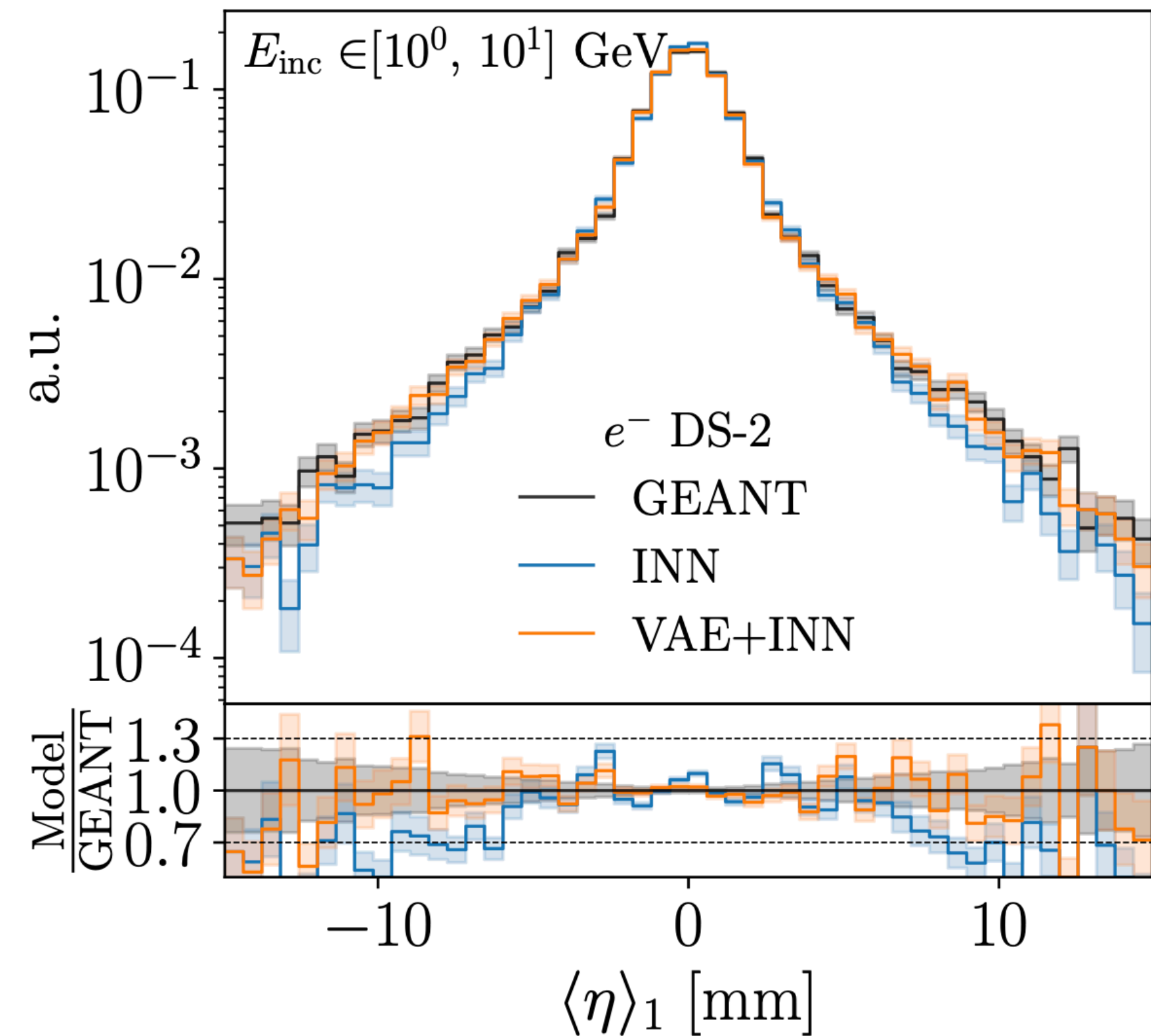
- \* Both energy and shower shape evaluated with normalizing flow models
- \* First layer energy deposits are predicted by **CaloINN**
- \* Other shape models to predict energy distribution within layers. Two separate studies:
  - **Continuous Normalizing Flows (CaloCFM)**- time “continuous” invertible transformation
  - **Diffusion Transformer**, stochastic inversion problem
- \* Conditioning models structure (and possibly more)



# Shower Shape generation



\* Great results both in energy interval and distribution, for both INN based strategies



# Normalising Flows & Invertible Neural Networks

- \* INN generative models are built on **bijective** and **invertible** transformations

$$y = f(x)$$

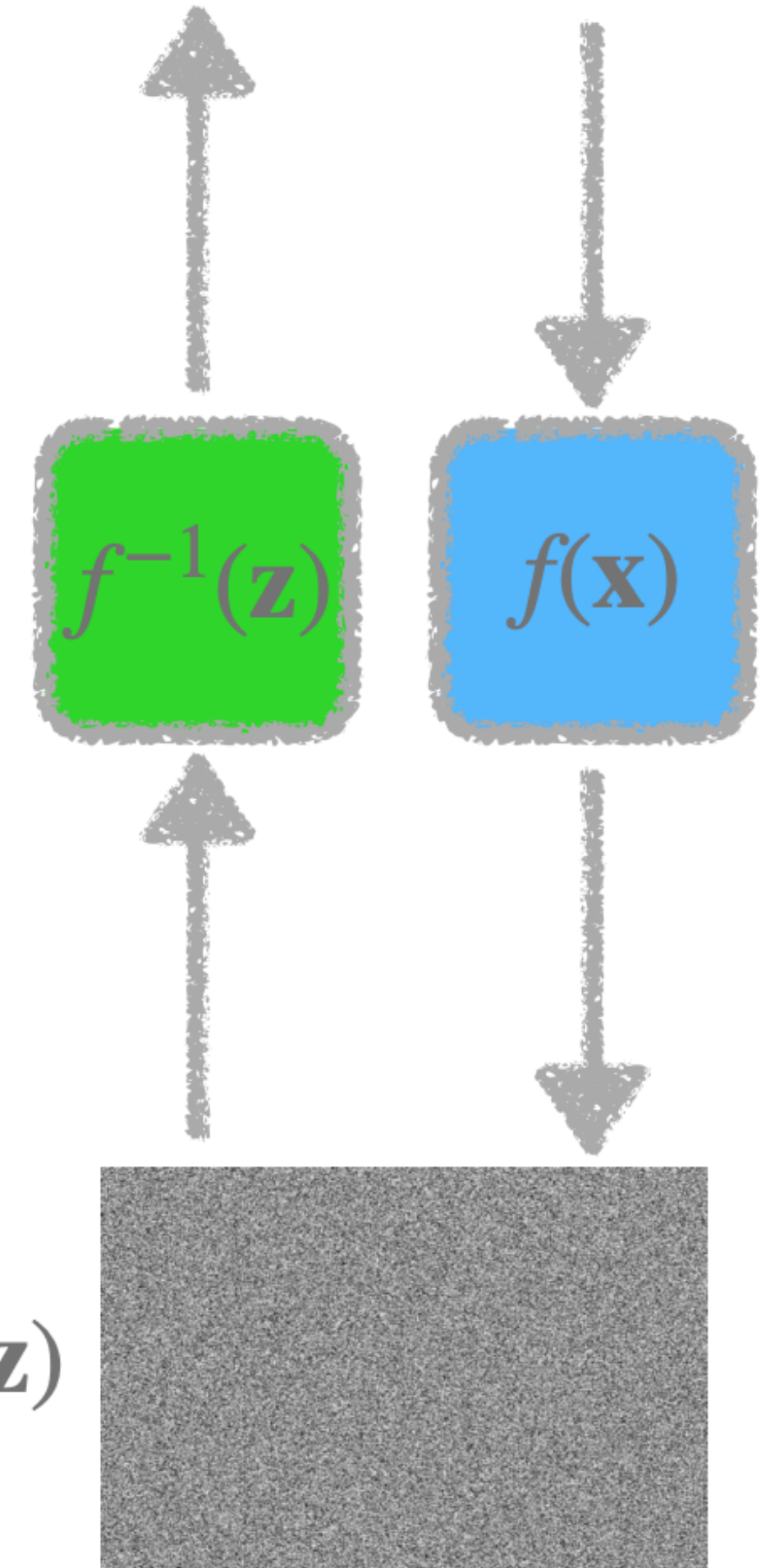
$$x = f^{-1}(y)$$

- \* **Works great for:** physics inverse problems, data generation, bi-directional simulation
- \* **Not so good for:** noisy data, discrimination, high dimension (huge Jacobian)
- \* Normalizing Flows are the more general version of INN, where the goal learning an invertible transformation from a **simple distribution to a complex one**.

## General characteristics

- Efficient sampling
- Efficient evaluation
- Highly expressive
- Useful latent representation
- Straightforward training
- Generally more interpretable

$p_{\mathbf{X}}(\mathbf{x})$



$p_{\mathbf{Z}}(\mathbf{z})$

# Invertible Neural Networks

\* Idea is to divide sample in 2  $x = (x_1, x_2)$  and define the transformation:

- where  $s(\cdot)$  and  $t(\cdot)$  are **small NN**
- $\odot$  is element by element product

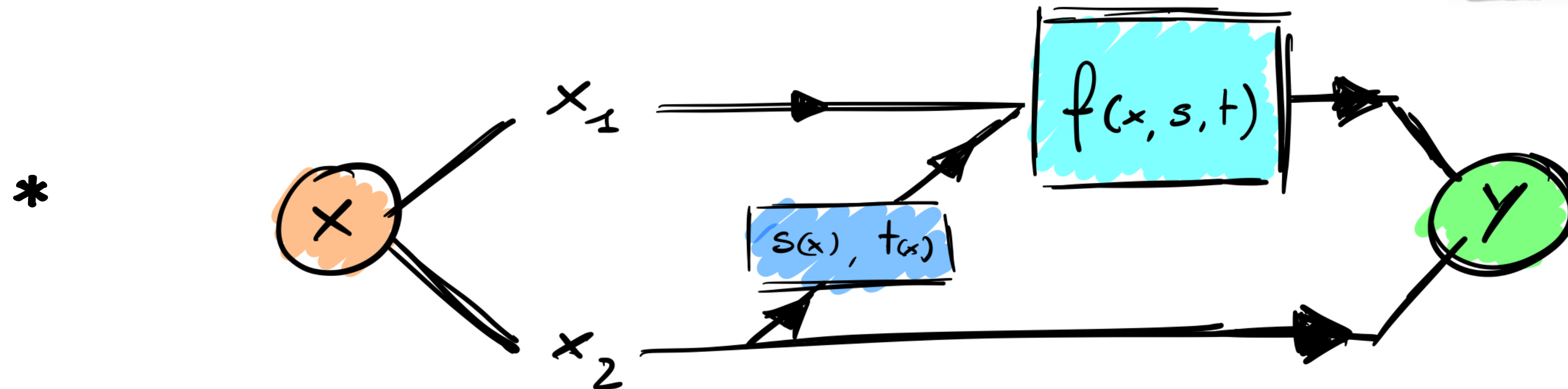
\* So the **inverse transformation** is straightforward

$$y_1 = x_1$$

$$y_2 = x_2 \odot \exp(s(x_1)) + t(x_1)$$

$$x_1 = y_1$$

$$x_2 = (y_2 - t(y_1)) \odot \exp(-s(y_1))$$



# Invertible Neural Networks

- \* **Key idea** is the density transformation (changing variable th.), so we can map simple distributions (i.e. gauss) to a generally complex one  $p(X)$ .

$$p_Y(y) = p_X(f^{-1}(y)) \cdot \left| \det \left( \frac{\partial f^{-1}(y)}{\partial y} \right) \right|$$

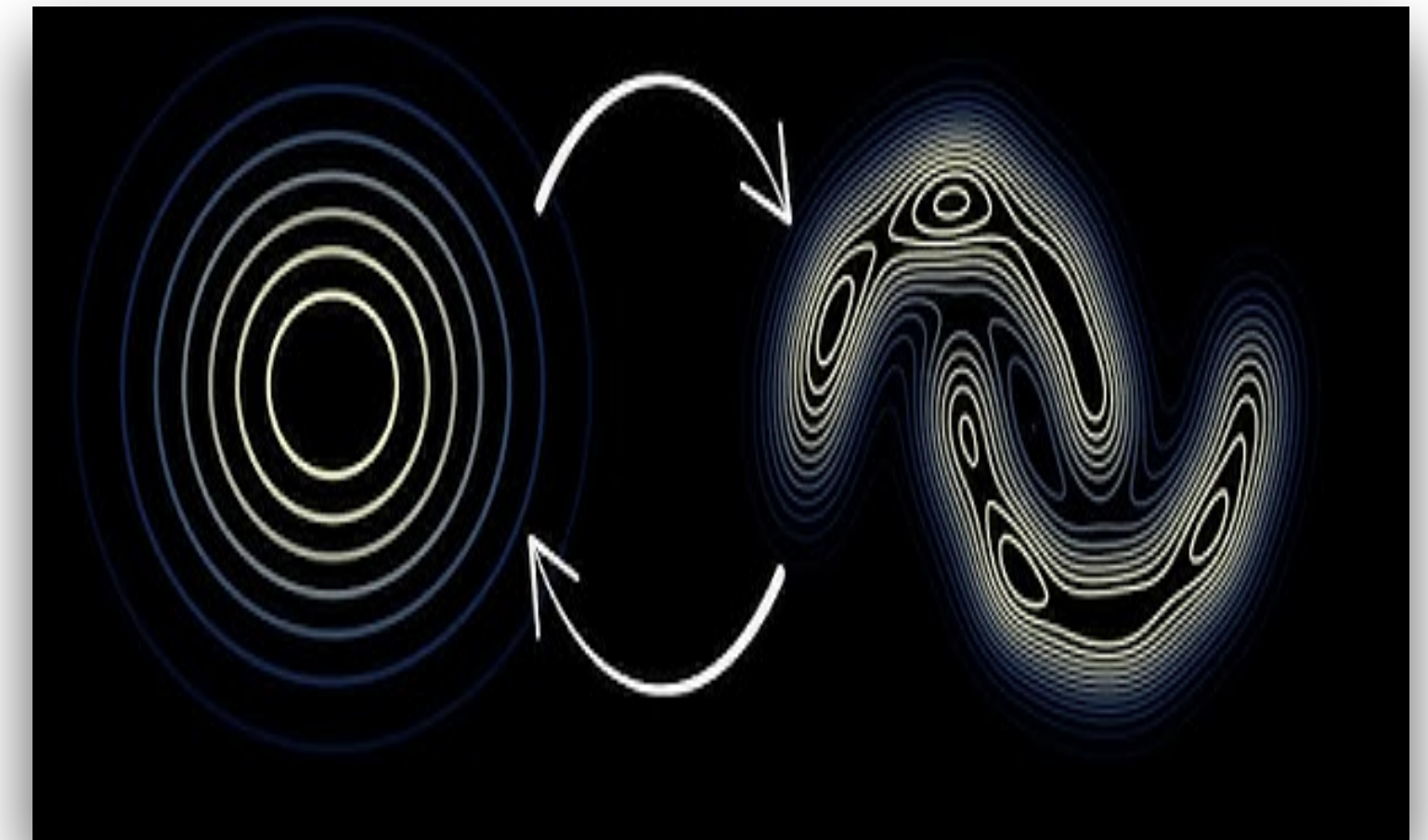
- \* Training doing **maximum likelihood** on data

$$\mathcal{L} = - \sum_i \log p_Y(y_i)$$

- \* and by changing variables

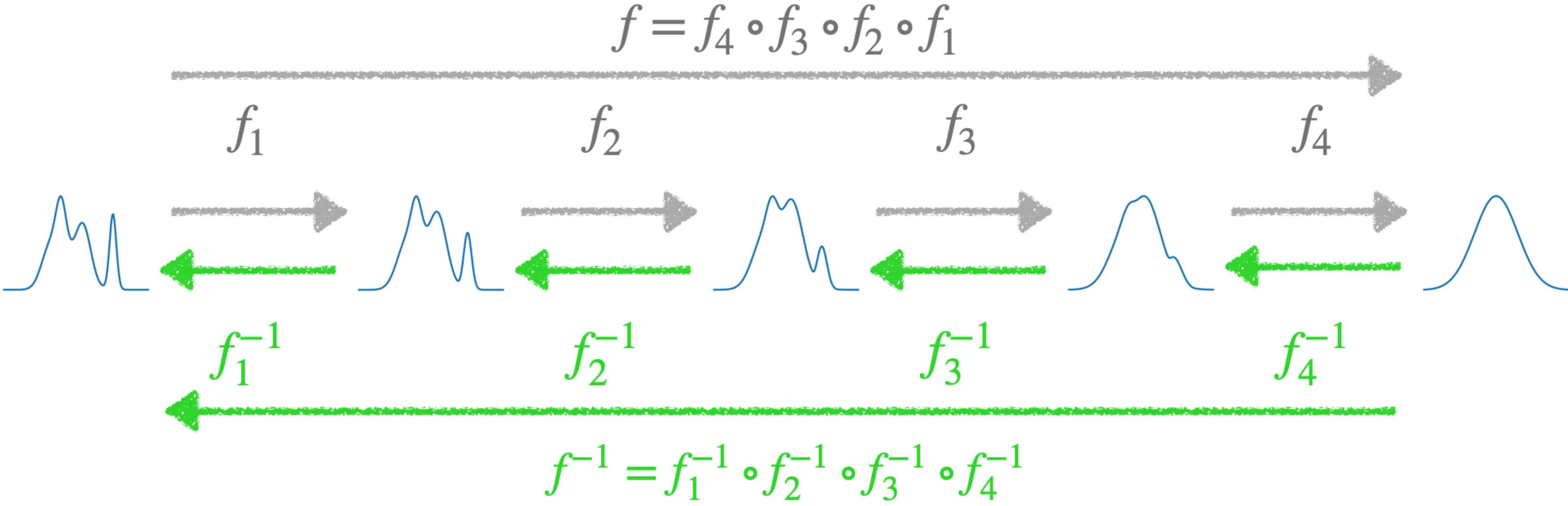
$$\mathcal{L} = - \sum_i \left[ \log p_X(f^{-1}(y_i)) + \log \left| \det \frac{\partial f^{-1}(y_i)}{\partial y_i} \right| \right]$$

- \* Good because inverse is **known** and “**easy**” Jacobian determinant



# Concatenating INN

## Composition of Flows



# Quantum Invertible Neural Networks

- \* Quantum circuits are described by Unitary Operations  $\rightarrow$  invertible by definition!
  - Natural connection to INN...

$$|\psi_{\text{out}}\rangle = U(\theta) |\psi_{\text{in}}\rangle$$

$$U(\theta) = U_1(\theta_1) U_2(\theta_2) \cdots U_N(\theta_N)$$

- \* First simple structure to try seems straightforward, starting with a simple 2 layer superposition

$$z \leftrightarrow x, \quad x = f_{\theta}(z),$$

$$|\psi_{\theta}\rangle = U(\theta)|z\rangle.$$

- \* Measuring and comparing them (generated) to data

$$p_{\theta}(x) = |\langle x|\psi_{\theta}\rangle|^2.$$

- \* Done minimising a log-likelihood function

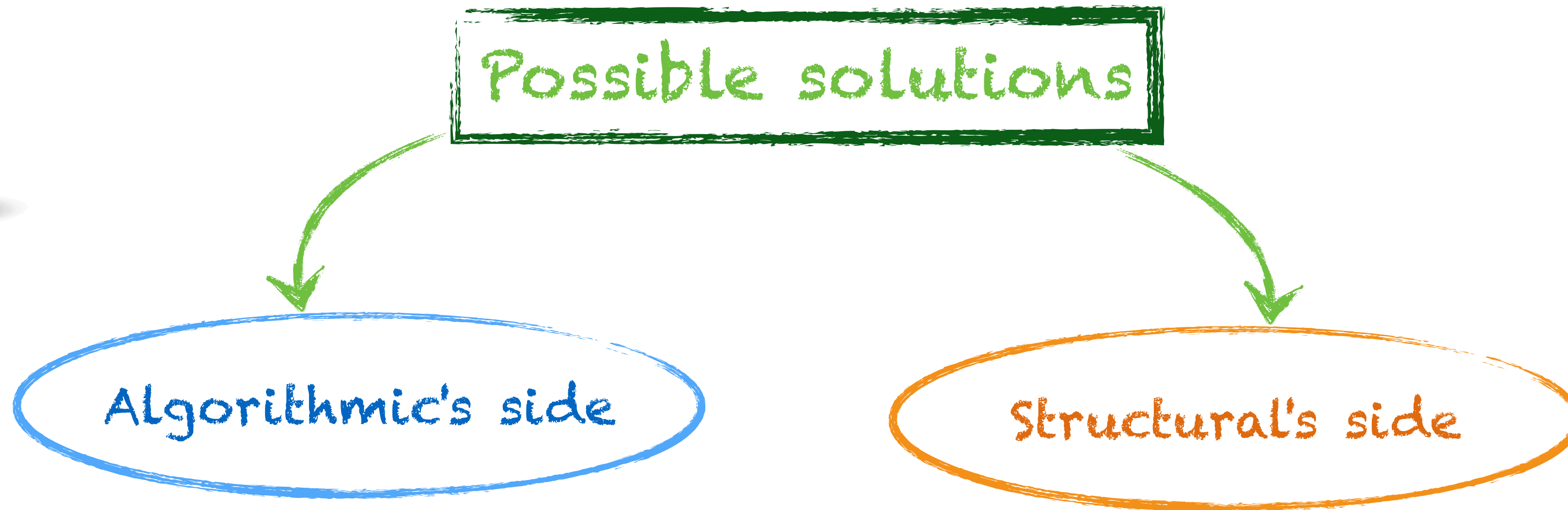
\*

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^N \log p_{\theta}(x_i).$$

# First qINN implementation

\* This is a common **reversible generative circuit**...

What is **not invertible** is the measurement process, so no good learning...



# Quantum Invertible Neural Networks

## Algorithmic's side

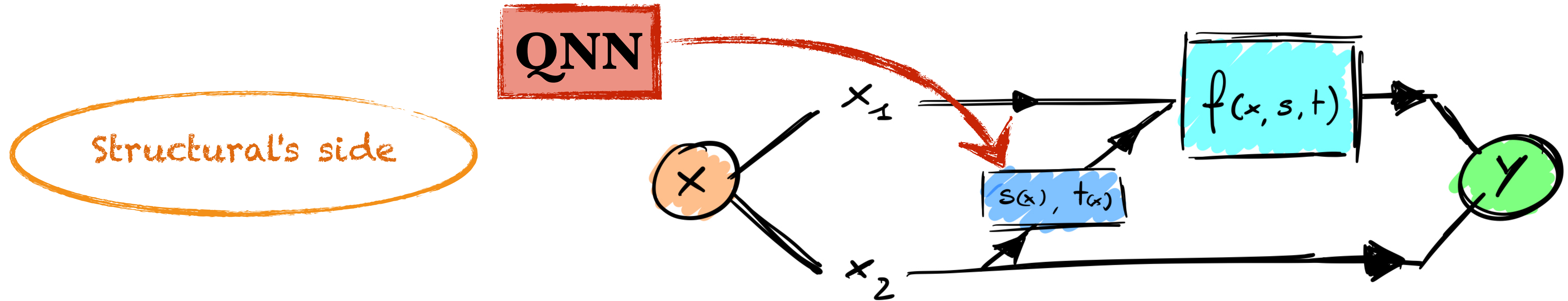
### \* 1) Saving full state vector

- **PRO:** Possible to invert the process exactly, maintain all correlation
- **CONS:** ok only for few qubits, not scalable. In HEP we need complex and high feature space...
  - ☆ Computationally huge to simulate
  - ☆ Possible on quantum hardware, but full tomography require many measurements (time)
- In HEP we need complex and high feature space...
- Possible with future quantum memories

### \* 2) Finding smart projection space

- Try to make projection univocal, SWAP-test. Complex and a bit problematic...
- by Spannowsky et al. <https://doi.org/10.21468/SciPostPhys.16.6.146>

# Quantum Invertible Neural Networks



- \* **Changing the paradigm**, using a classical INN structure and doing **quantum conditioning** looking to improve expressibility in difficult phase space region
- \* **PRO:** inversion classically granted, scalable, may improve correlation modelling and specific variable conditioning
- \* **CONS:** not guarantee improvement in expressibility, classic core with same classical limits (Jacobian Inversion)

# Implementation strategy



- \* Starting first implementation from Fast Calo Challenge framework

<https://calochallenge.github.io/homepage/>

- \* Moving from Keras to PyTorch with **Pannylane** interface
- \* Using **ATLAS OpenData** sample of pions (photons) generated at 65 GeV using Geant4
- \* Energy distribution generation starting with a dataset of 1M event. Possible to make shape distributions too.

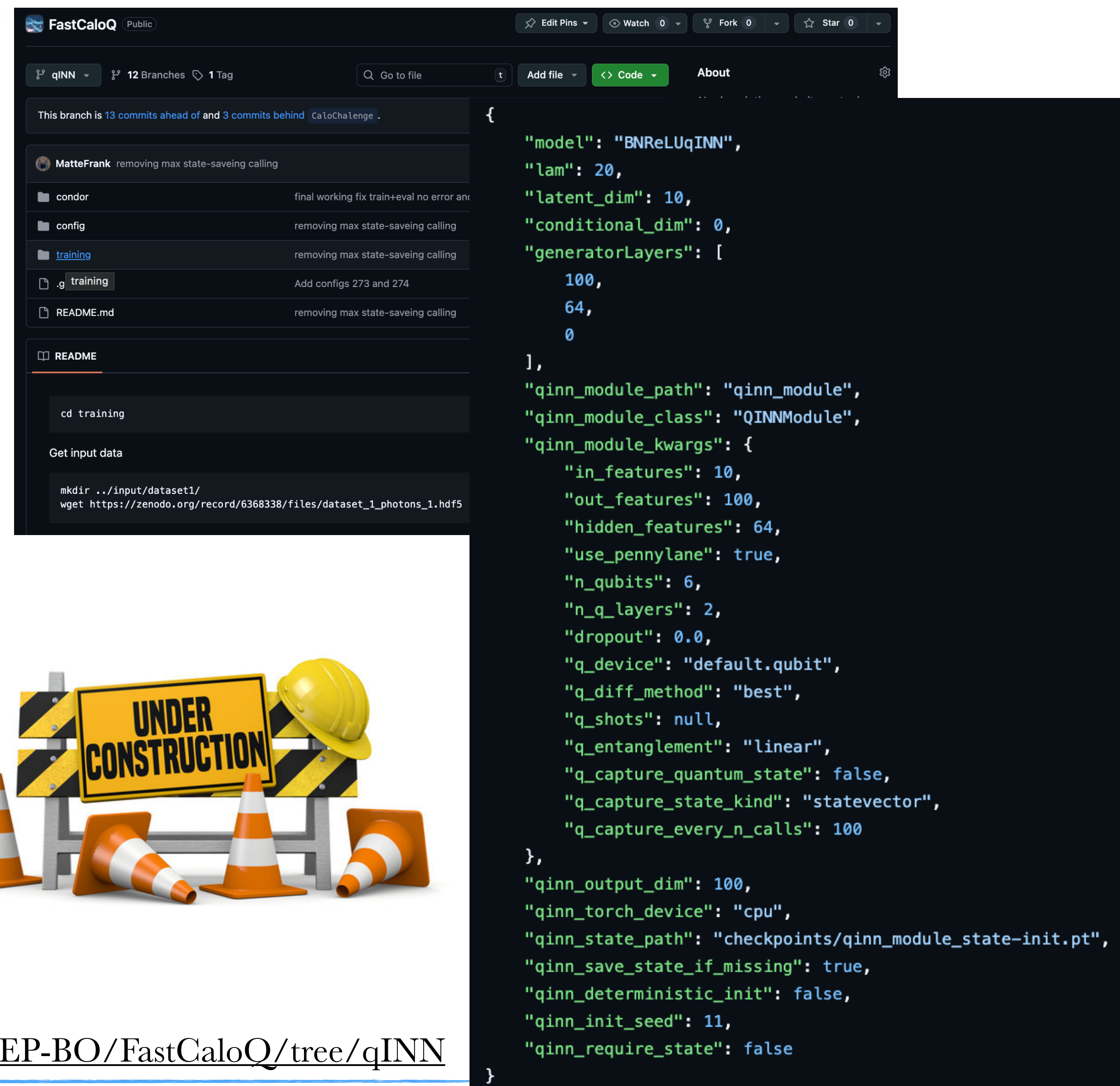


Instrumentation and Detectors (physics.ins-det)

<https://doi.org/10.1088/1361-6633/ae1304>

# Quantum Condition implementation (Work in progress)

- \* **Preliminary** implementation and test on **full energy distribution generation**.
- \* Small  $f$  to start (100,64), 1000 ev.
- \* Latent space of dimension 10,
  - 10 input features per 10 batches
- \* Only one conditioning function with 6 qubits in 2 layers



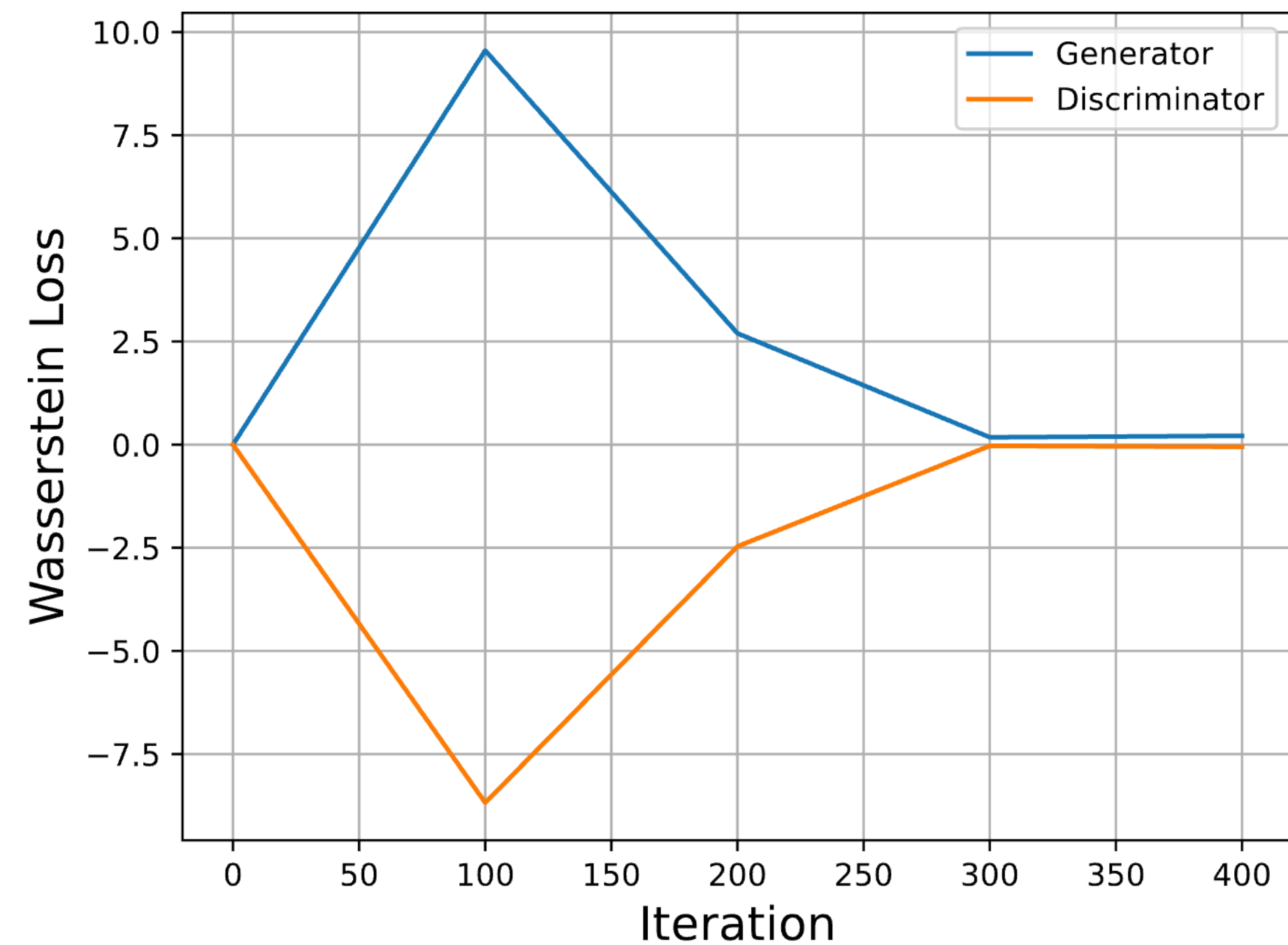
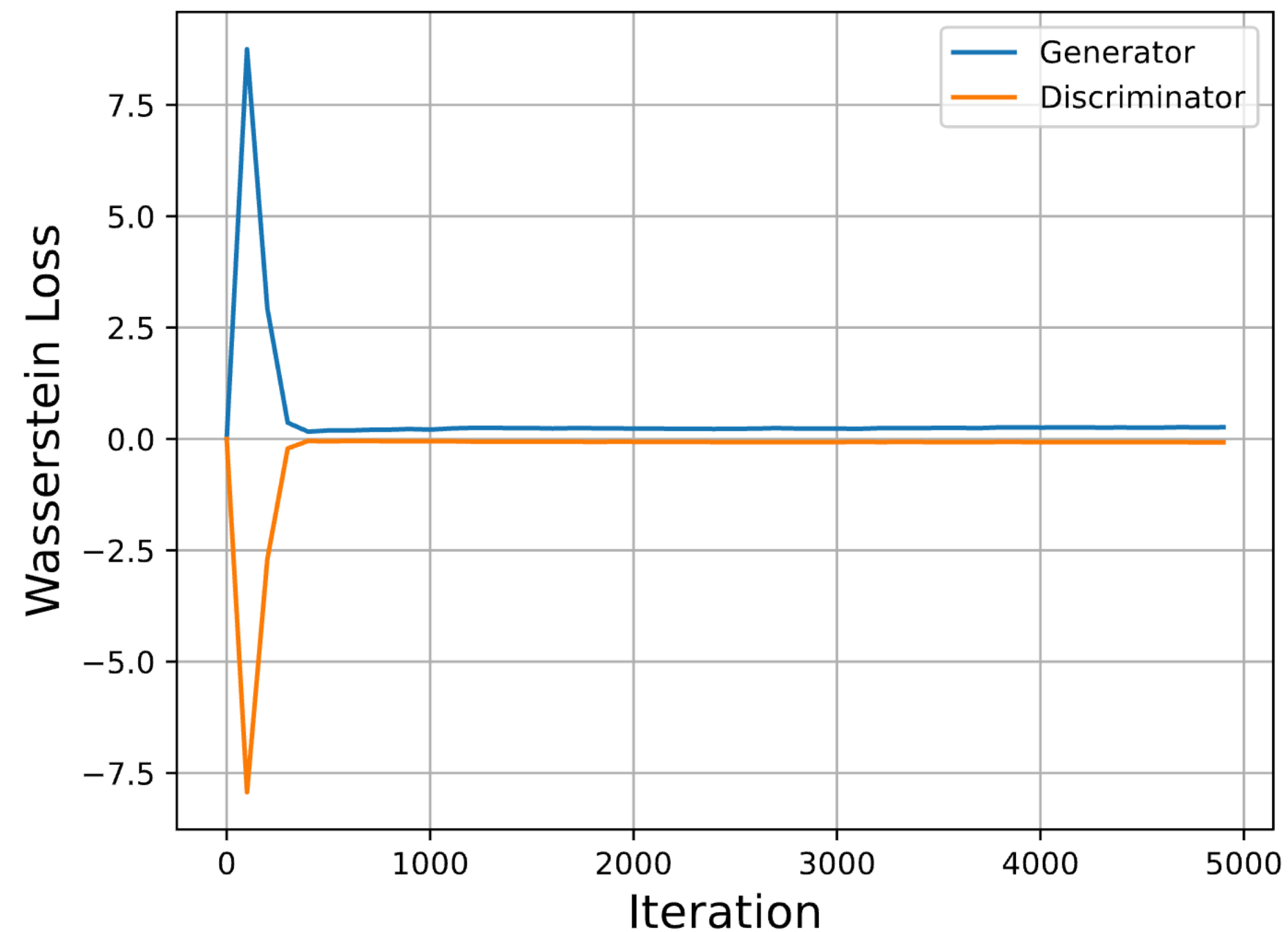
The screenshot shows a GitHub repository for 'FastCaloQ' with a public license. The repository has 12 branches and 1 tag. The current branch is 'qINN', which is 13 commits ahead of and 3 commits behind the 'CaloChallenge' branch. A commit by 'MatteFrank' is shown, titled 'removing max state-saving calling'. The repository structure includes folders for 'condor', 'config', and 'training', along with files for '.g training', 'README.md', and 'README'. The 'README' file contains instructions to run 'cd training' and 'Get input data' using 'mkdir' and 'wget'. A yellow 'UNDER CONSTRUCTION' sign with a hard hat and orange traffic cones is overlaid on the bottom part of the screenshot. To the right, a configuration file is shown with the following JSON content:

```
{
  "model": "BNReLUqINN",
  "lam": 20,
  "latent_dim": 10,
  "conditional_dim": 0,
  "generatorLayers": [
    100,
    64,
    0
  ],
  "qinn_module_path": "qinn_module",
  "qinn_module_class": "QINNModule",
  "qinn_module_kwargs": {
    "in_features": 10,
    "out_features": 100,
    "hidden_features": 64,
    "use_pennylane": true,
    "n_qubits": 6,
    "n_q_layers": 2,
    "dropout": 0.0,
    "q_device": "default.qubit",
    "q_diff_method": "best",
    "q_shots": null,
    "q_entanglement": "linear",
    "q_capture_quantum_state": false,
    "q_capture_state_kind": "statevector",
    "q_capture_every_n_calls": 100
  },
  "qinn_output_dim": 100,
  "qinn_torch_device": "cpu",
  "qinn_state_path": "checkpoints/qinn_module_state-init.pt",
  "qinn_save_state_if_missing": true,
  "qinn_deterministic_init": false,
  "qinn_init_seed": 11,
  "qinn_require_state": false
}
```

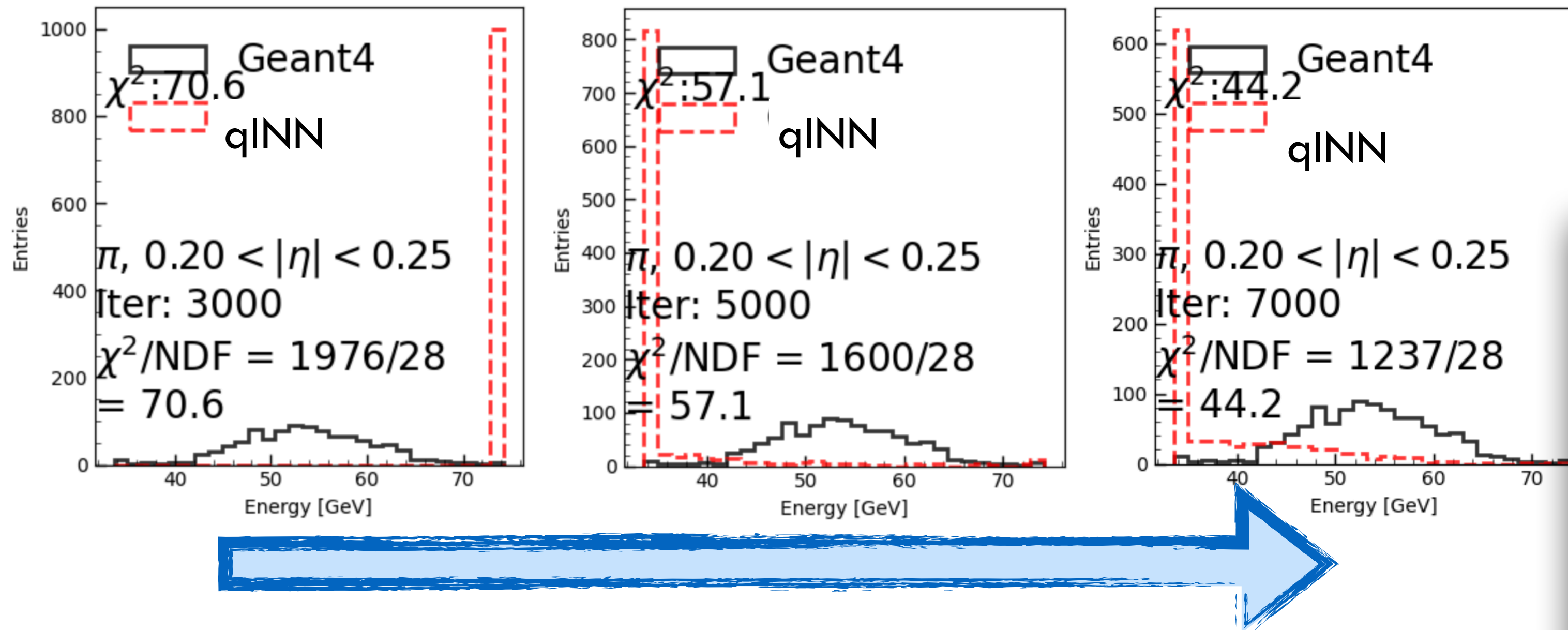
<https://github.com/QC4HEP-BO/FastCaloQ/tree/qINN>

Very, VERY, Preliminary!!

- \* Currently using Wasserstein Loss. Start **converging quite fast**, but do not reflect correct learning. Maybe change to specific likelihood.
- \* Saving the latent at different checkpoints for **studying evolution**, different batch size



Very, VERY, Preliminary!!

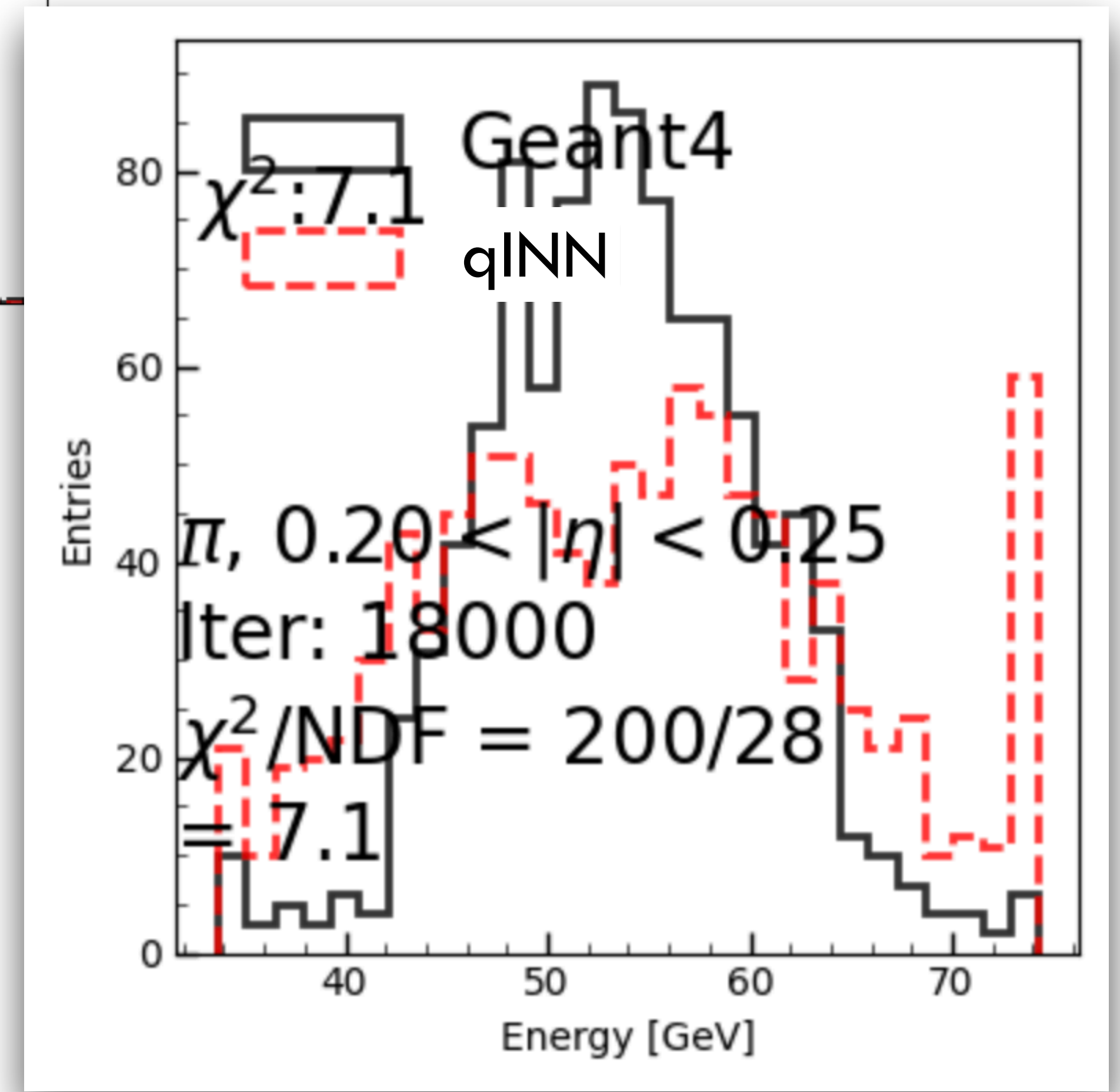


\* **First feasibility result**, not final implementation!

\* Iter = epoch\*batch

\* **Encouraging learning** after 2000 epochs

• fast training ~120 min



# Next Steps

---

- \* **Idea feasibility confirmed**, now need to investigate if could give any possible advantage
- \* qNN for both  $s(\cdot)$  and  $t(\cdot)$  NN components and optimise the loss.
- \* **Tease the network and tune parameters**: more depth classical part, features, events...
- \* Extend to generate also **shower shapes!**
- \* Extensive tests and try integration as conditioning in ATLAS FastCaloSim INN part
- \* Already implementing also the **reversible generative circuit** version



# Conclusion

- \* Implementation of a **first quantum INN version in generative HEP** application in FastCaloChallenge ecosystem (public dataset with ATLAS openData)
- \* Idea seem to work with (**very very**) **preliminary** encouraging results
- \* Quantum generative NN may become a **key feature in HEP** with increasing data and more complex models, HighLumi - LHC in 2030
- \* **Future colliders and detectors** will gives even more computational challenges!
- \* Already looking at next FastCaloSim with them.
- \* **Better be prepared and start already!!!**

