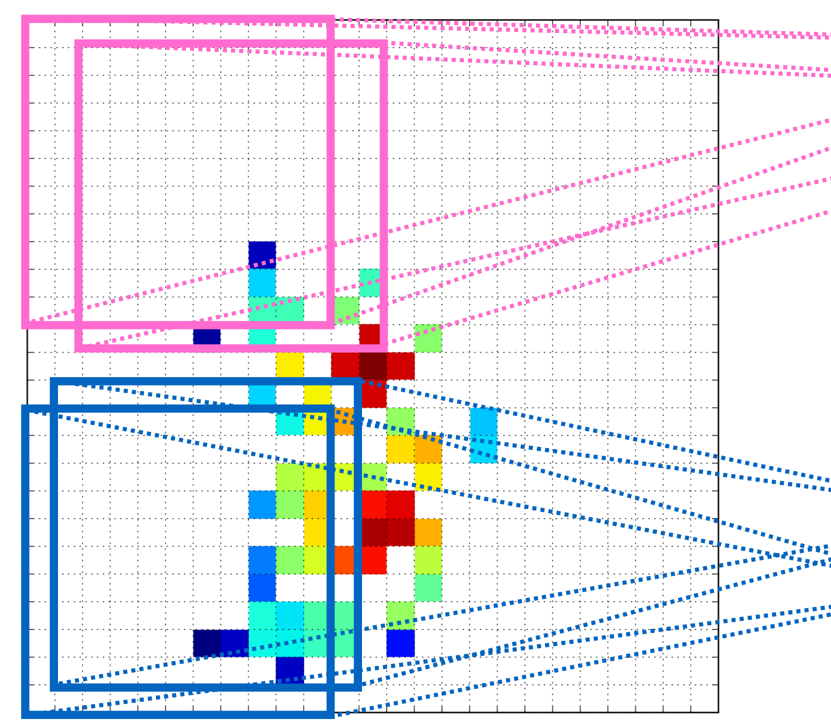
Machine Learning for Precision Physics at the LHC

Benjamin Nachman

Lawrence Berkeley National Laboratory

bpnachman.com
bpnachman@lbl.gov

@bpnachman bnachman



BERKELEY LAB

BIDS
BERKELEY INSTITUTE FOR DATA SCIENCE

LoopFest XXI
June 2023

Precision Physics at the LHC

Theory of everything

↓

Physics simulators

↓

Detector-level observables

↓

Pattern recognition ⟷

Nature

↓

Experiment

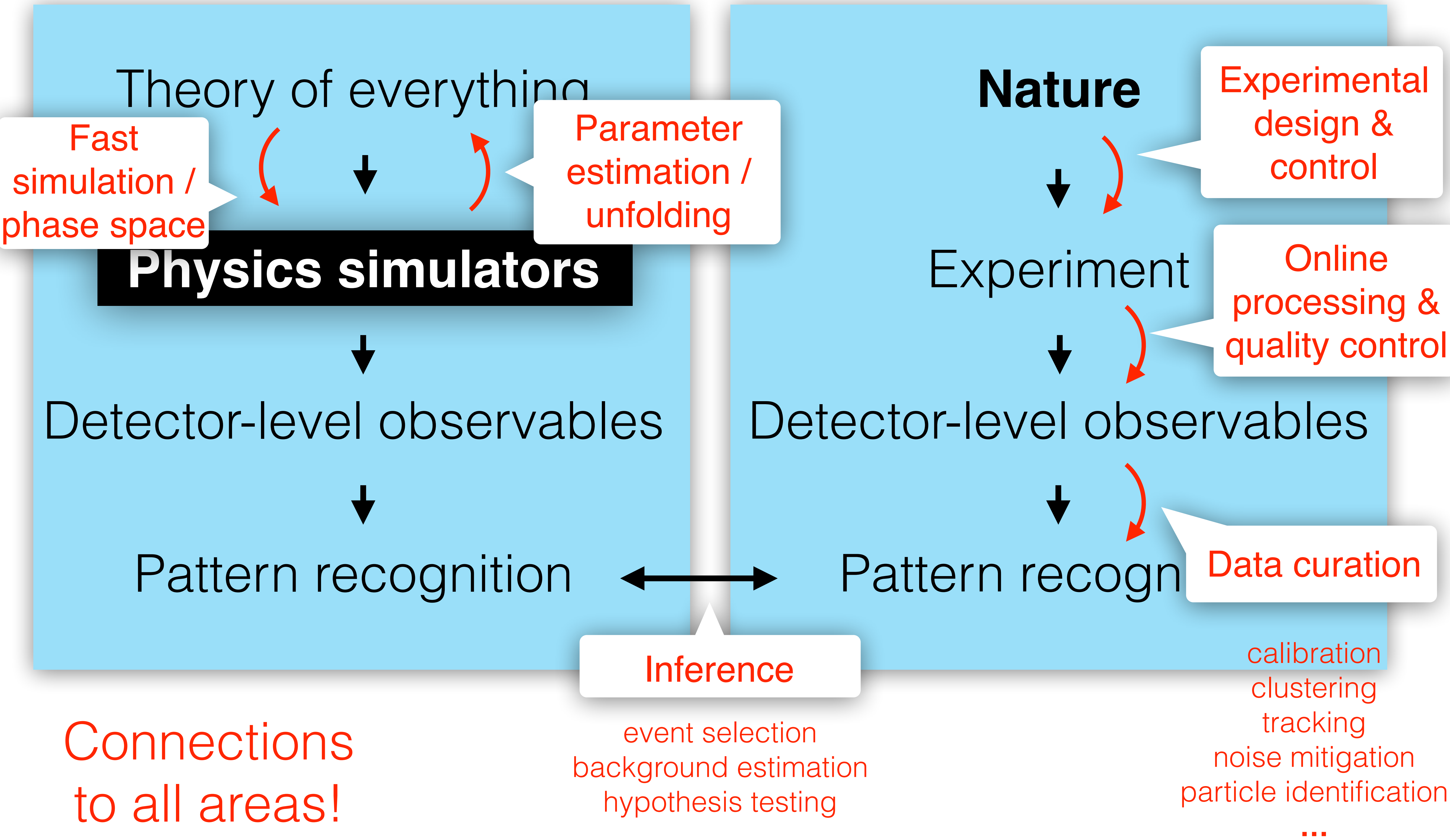
↓

Detector-level observables

↓

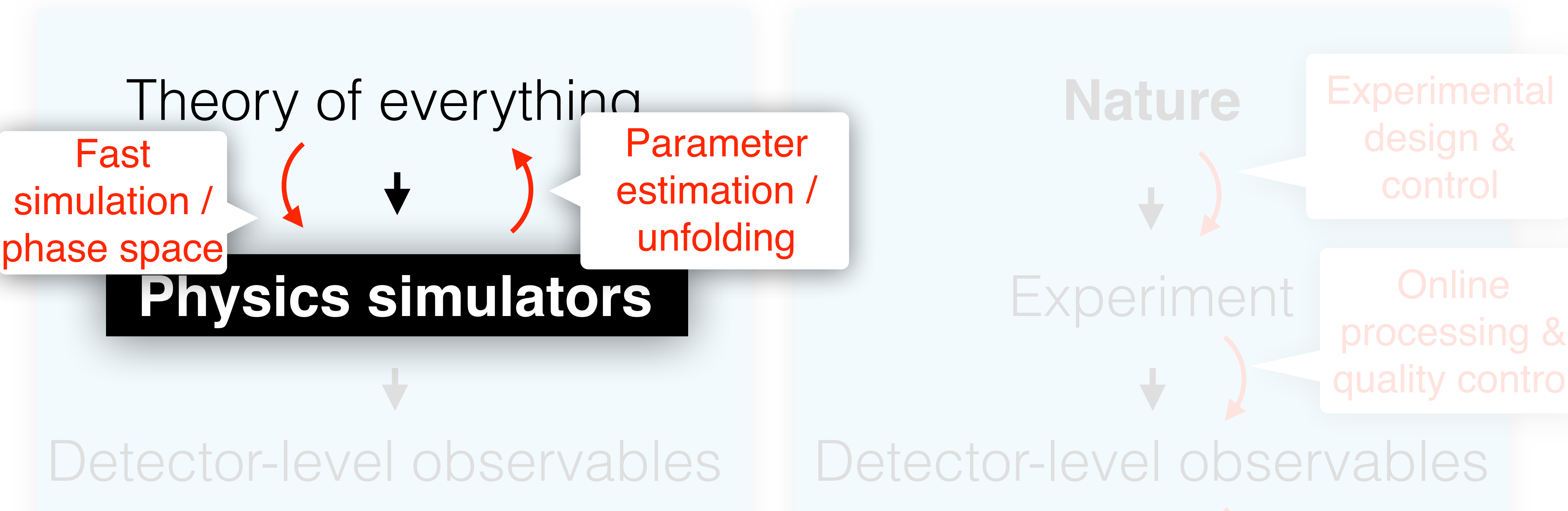
Pattern recognition

Precision Physics at the LHC



Connections to all areas!

Precision Physics at the LHC



Plan for today: give representative examples from each area to illustrate the potential of ML. I'm not going to be comprehensive and I do not claim ML will solve all our problems!

Precision Physics at the LHC

Theory of everything

Forward Model

Inverse Model

Physics simulators

Detector-level observables

Nature

Experimental design & control

Experiment

Online processing & quality control

Detector-level observables

Data curation

calibration
clustering
tracking
noise mitigation
particle identification
...

Inference

event selection
background estimation
hypothesis testing
...

Connections to all areas!

Plan for today: give representative examples from each area to illustrate the potential of ML. I'm not going to be comprehensive and I do not claim ML will solve all our problems!

Precision Physics at the LHC

Theory of everything

Forward Model

Inverse Model

Physics simulators

Detector-level observables

Pattern recognition

Nature

Experimental design & control

Experiment

Online processing & quality control

Detector-level observables

Pattern recognition

Data curation

calibration
clustering
tracking
noise mitigation
particle identification
...

Inference

event selection
background estimation
hypothesis testing
...

Connections to all areas!

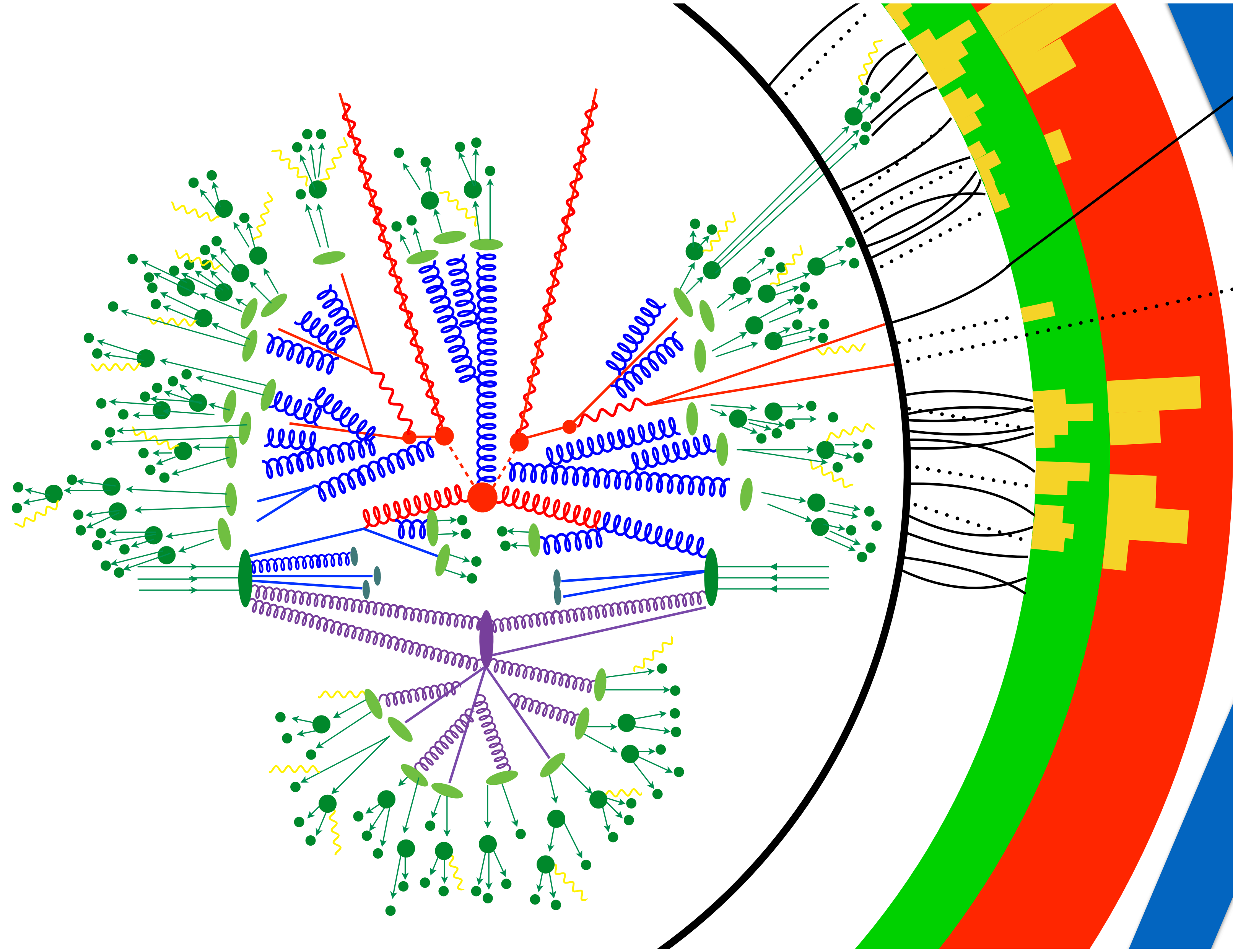
Plan for today: give representative examples from each area to illustrate the potential of ML. I'm not going to be comprehensive and I do not claim ML will solve all our problems!

Enhancing the "forward model"

Complete predictions require composing a number of different components.

There are proposals for ML at all levels:

- PDFs
- Phase space
- Matrix elements
- Parton showers
- Hadronization
- End-to-end



Enhancing the "forward model"

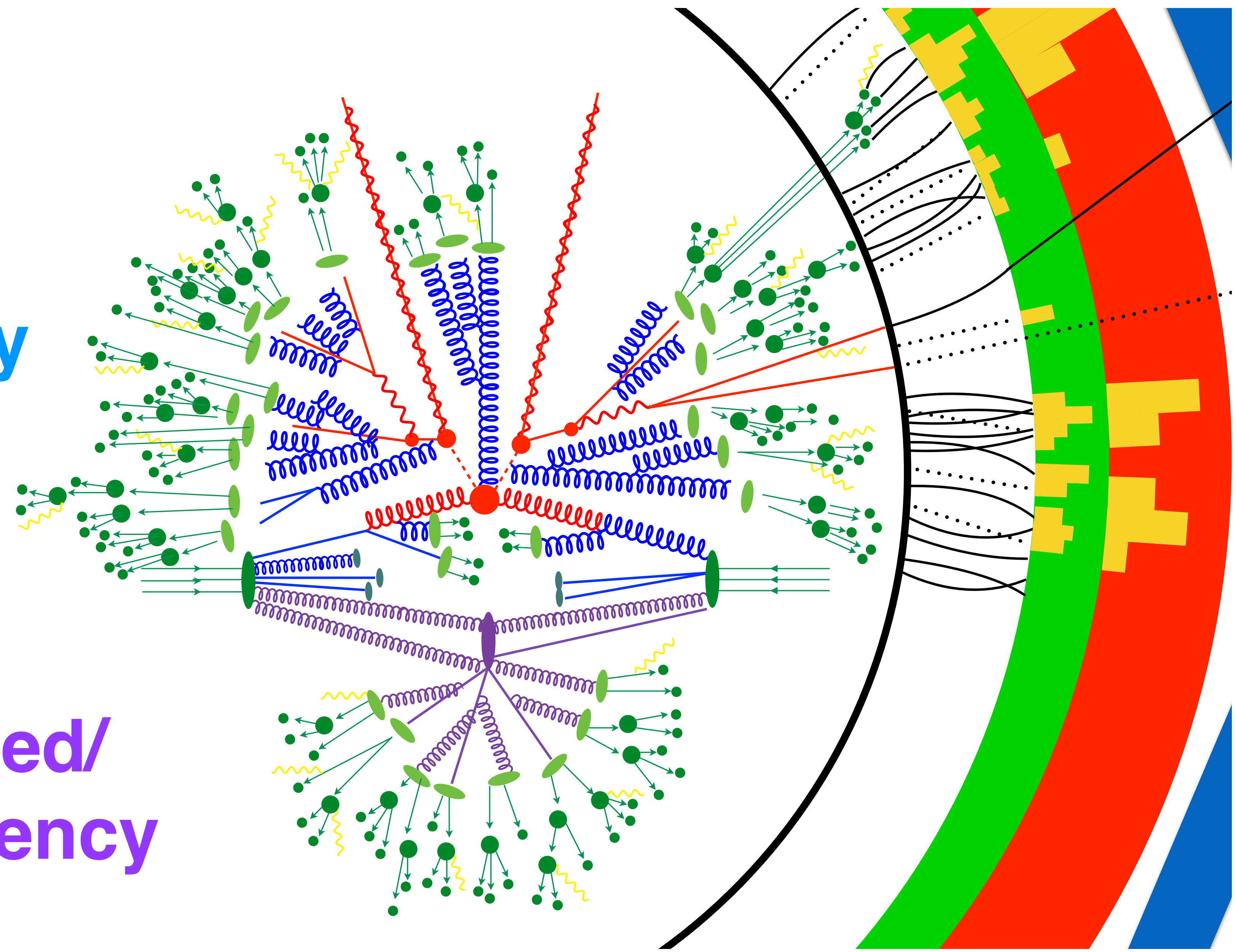
Complete predictions require composing a number of different components.

There are proposals for ML at all levels:

- PDFs
- Phase space
- Matrix elements
- Parton showers
- Hadronization
- End-to-end

accuracy

speed/ efficiency



Enhancing the “forward model”

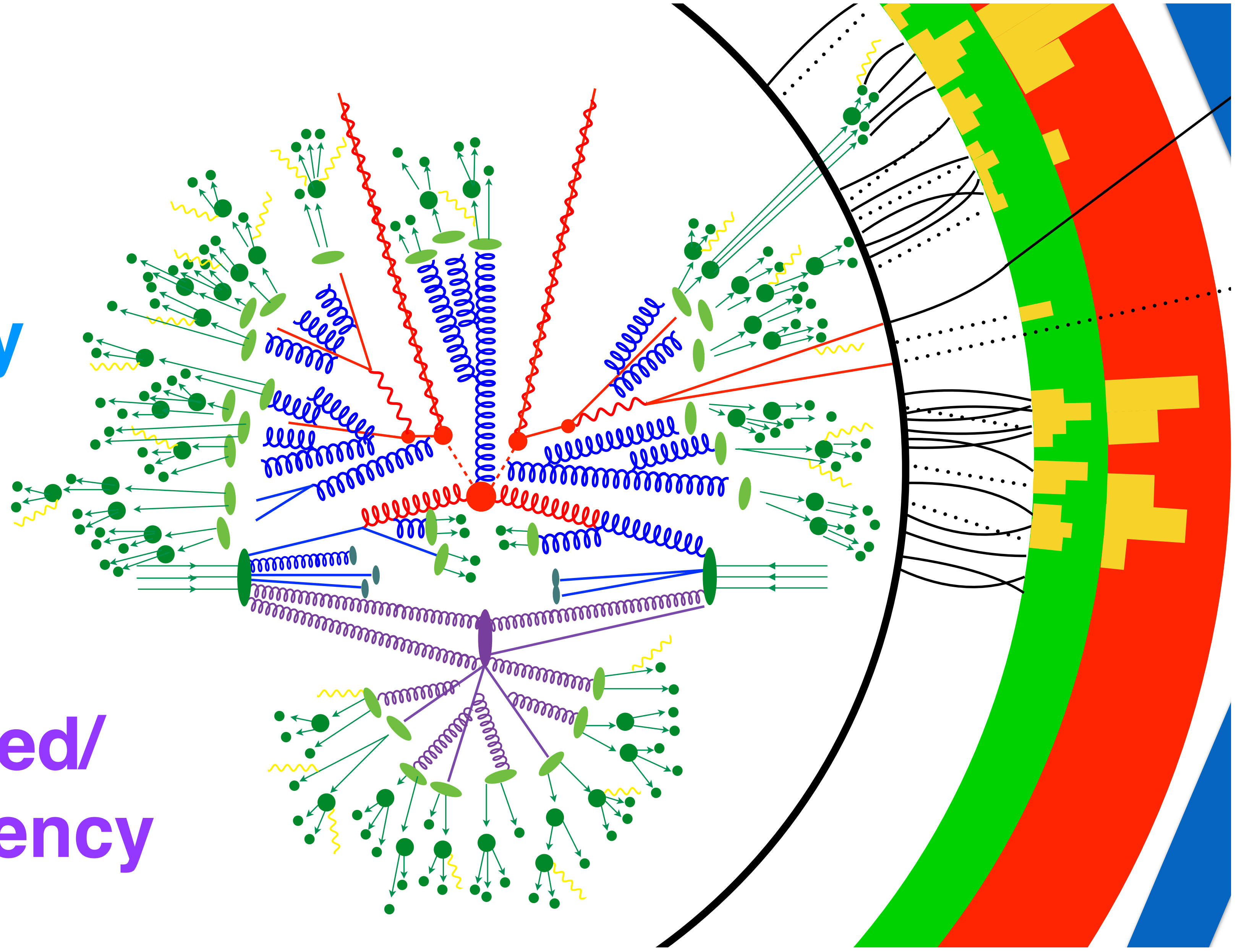
Complete predictions require composing a number of different components.

There are proposals for ML at all levels:

- PDFs
- Phase space
- Matrix elements
- Parton showers
- Hadronization
- End-to-end

accuracy

speed/ efficiency



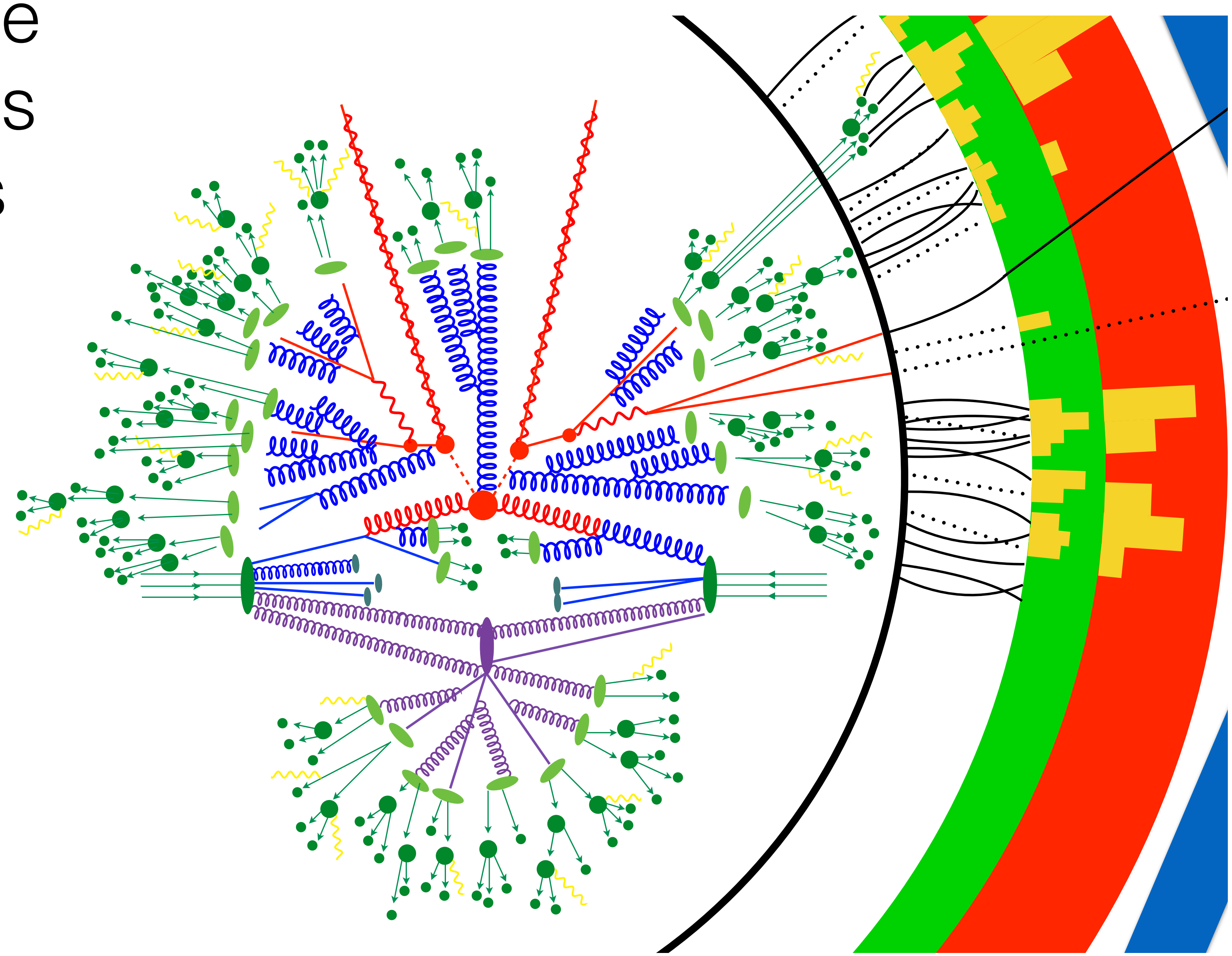
Why hadronization?

We do not understand hadronization from first principles!

Existing approaches use physics-inspired models with lots of parameters that are fit to data.

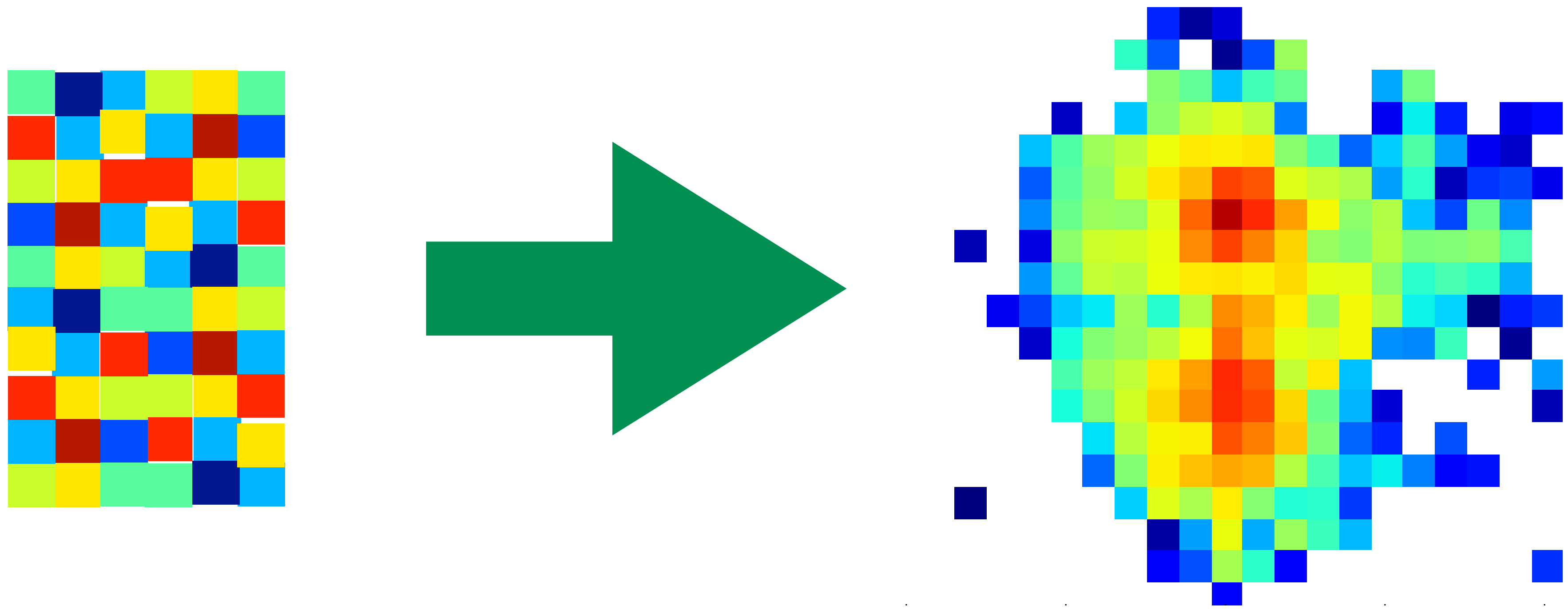
Seems to be calling out for a machine learning solution!

See also PDFs (and the pioneer, NNPDF)



What is a deep generative model?

A **generator** is nothing other than a function that maps random numbers to structure.



Deep generative models: the map is a deep neural network.

What is a deep generative model?

GANs
Generative Adversarial Networks

Score-based

Restricted Botlzmann Machines

Mixture Density Networks

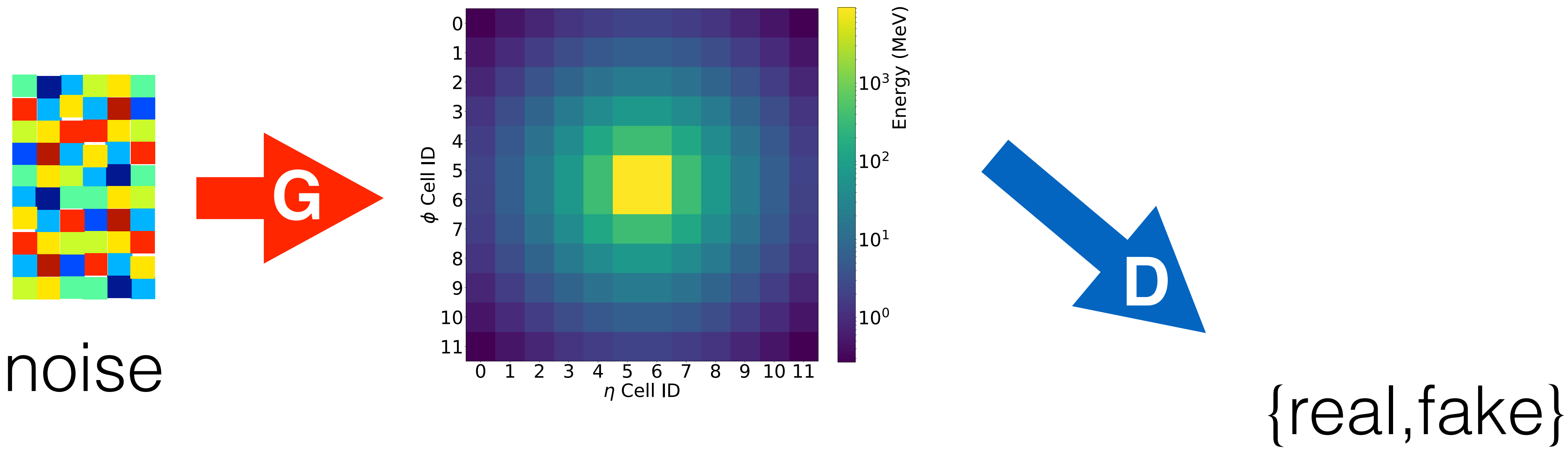
NFs
Normalizing Flows

Energy-based models

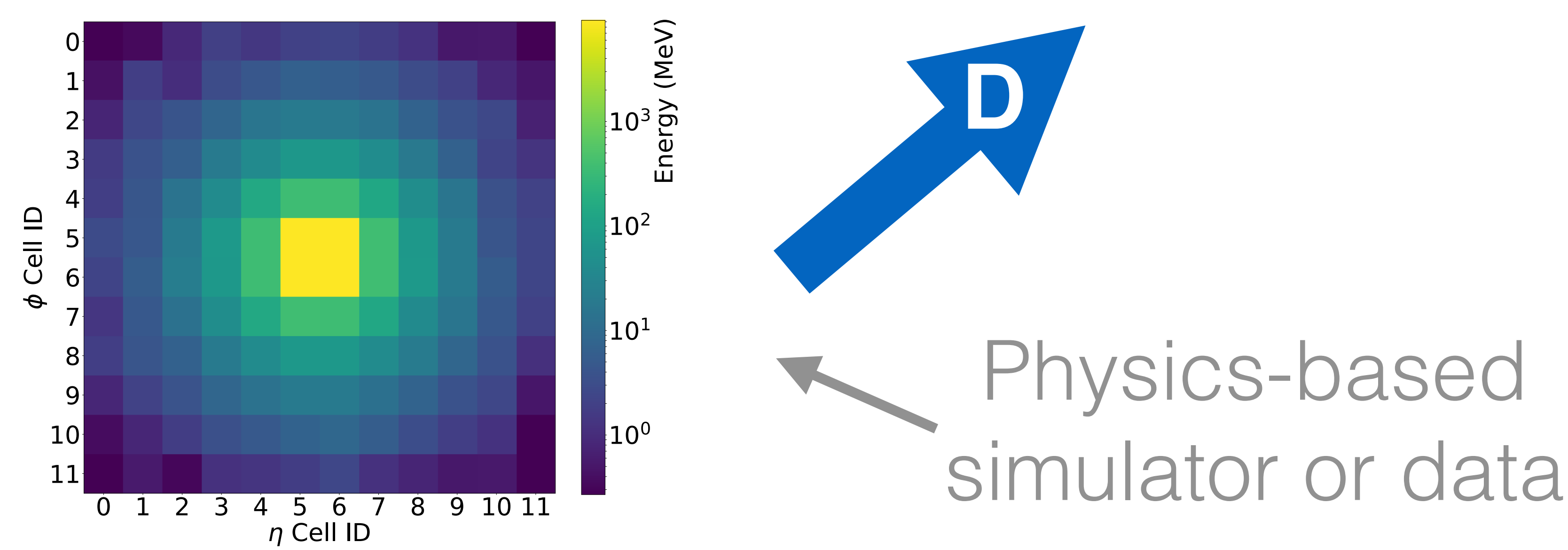
VAEs
Variational Autoencoders

Our tool of choice: GANs

Generative Adversarial Networks (GANs):
A two-network game where one ***maps noise to structure*** *and one* ***classifies images as fake or real****.*



When **D** is maximally confused, **G** will be a good generator



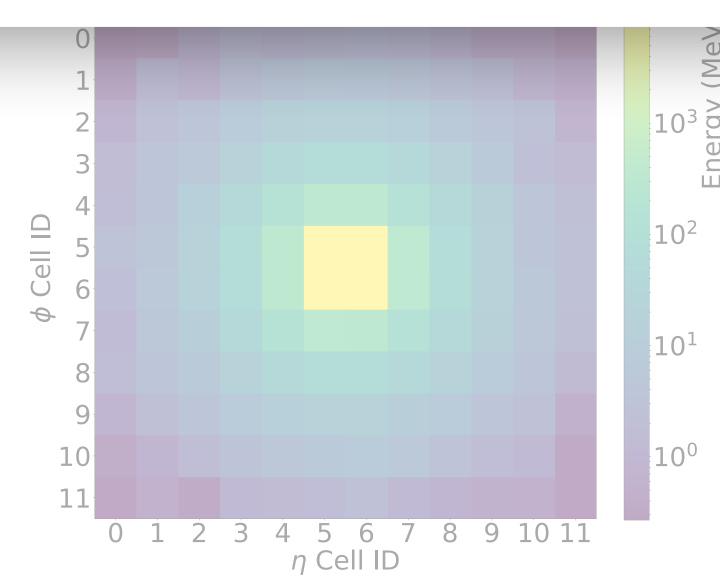
Our tool of choice: GANs

Generative Adversarial Networks (GANs):
A two-network game where one ***maps noise to structure*** *and one* ***classifies images as fake or real****.*

There are many new methods that have superior robustness to GANs, but for reasons I hope will be clear later, we need the flexibility of GANs that no other approach can accommodate (yet).

eal,fake}

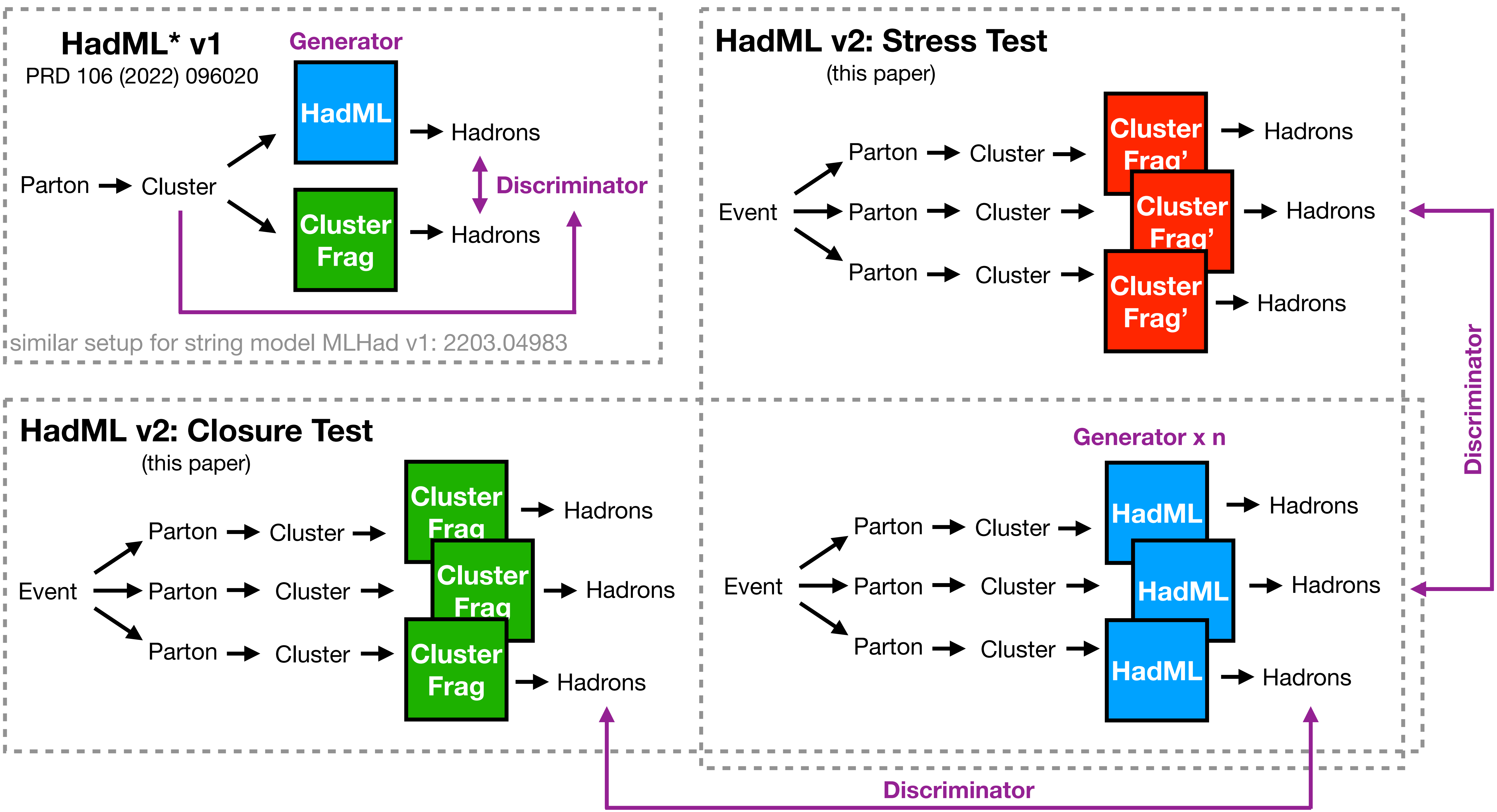
When **D** is maximally confused, **G** will be a good generator



D

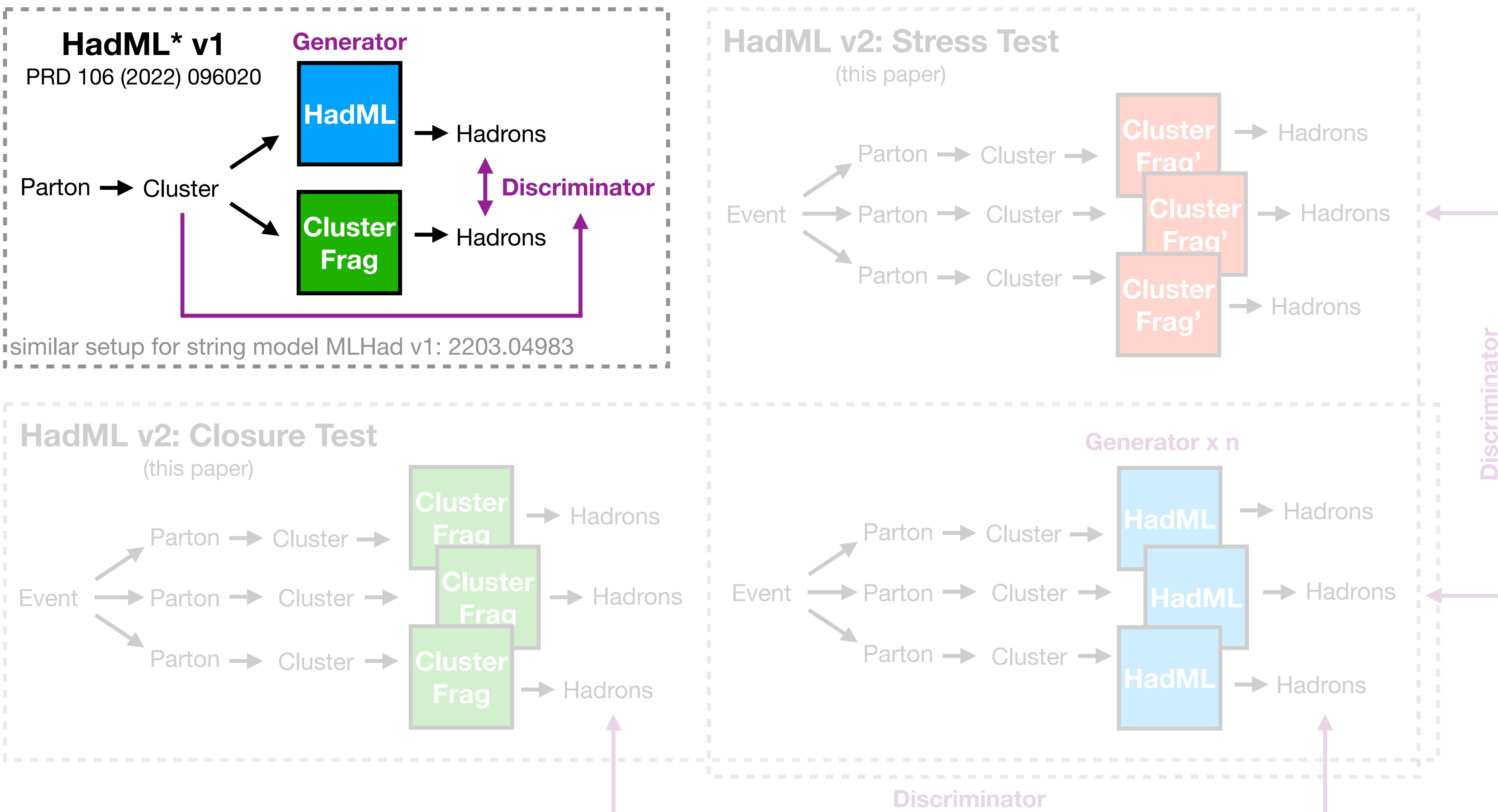
Physics-based simulator or data

ML Hadronization - Overview



Based on 2203.12660 and 2305.17169, set in the cluster model. Similar studies set in the string model found in 2203.04983.

ML Hadronization - Overview

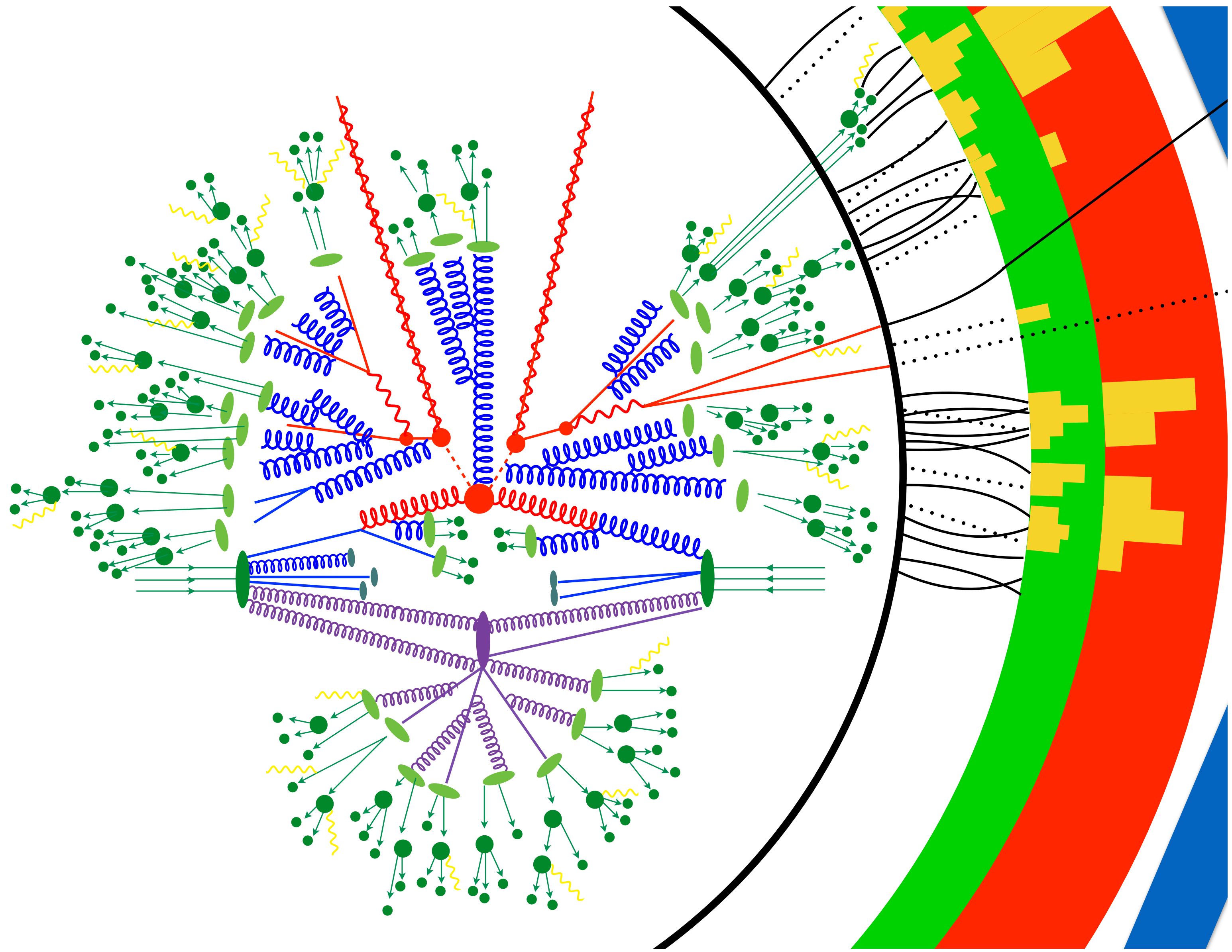


Based on 2203.12660 and 2305.17169, set in the cluster model. Similar studies set in the string model found in 2203.04983.

Cluster hadronization

Physics inspiration: preconfinement. Universal distribution of color singlet objects ('clusters') which decay into hadrons.

Our approach: take preconfinement as a starting point and learn the decay. In the future, we want to be able to go beyond this starting point*



*We actually tried fitting a version of Pythia with all the same simplifying assumptions as our Herwig model and it doesn't work yet for known reasons. Please ask if you want to hear more about this!

Training HADML

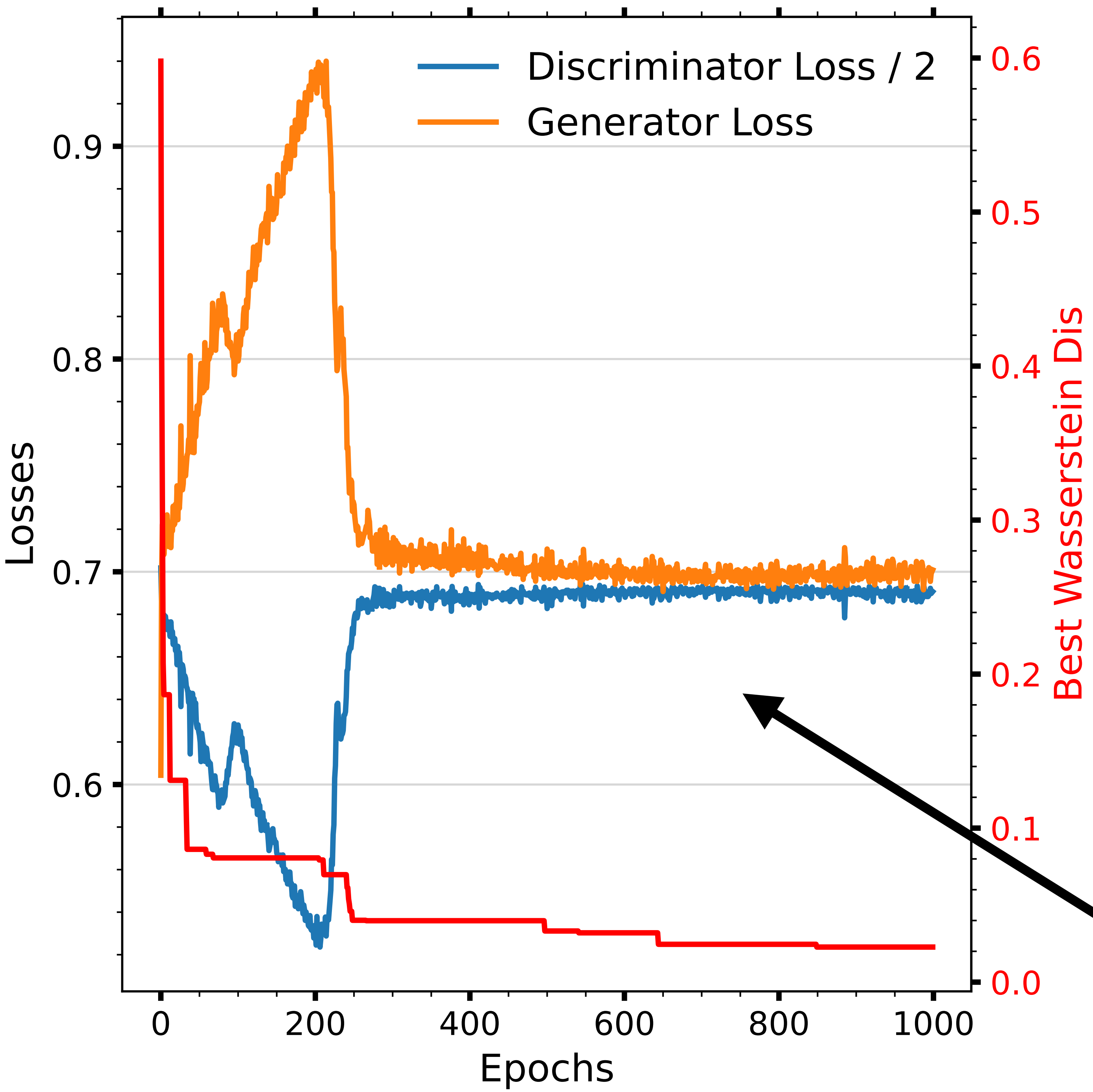
We have a conditional GAN, with cluster 4-vector input and two hadron 4-vector outputs.

Training HADML

We have a conditional GAN, with cluster 4-vector input and two hadron 4-vector outputs.

We simplify this by considering only pions and generating two angles in the cluster rest frame.

Training HADML

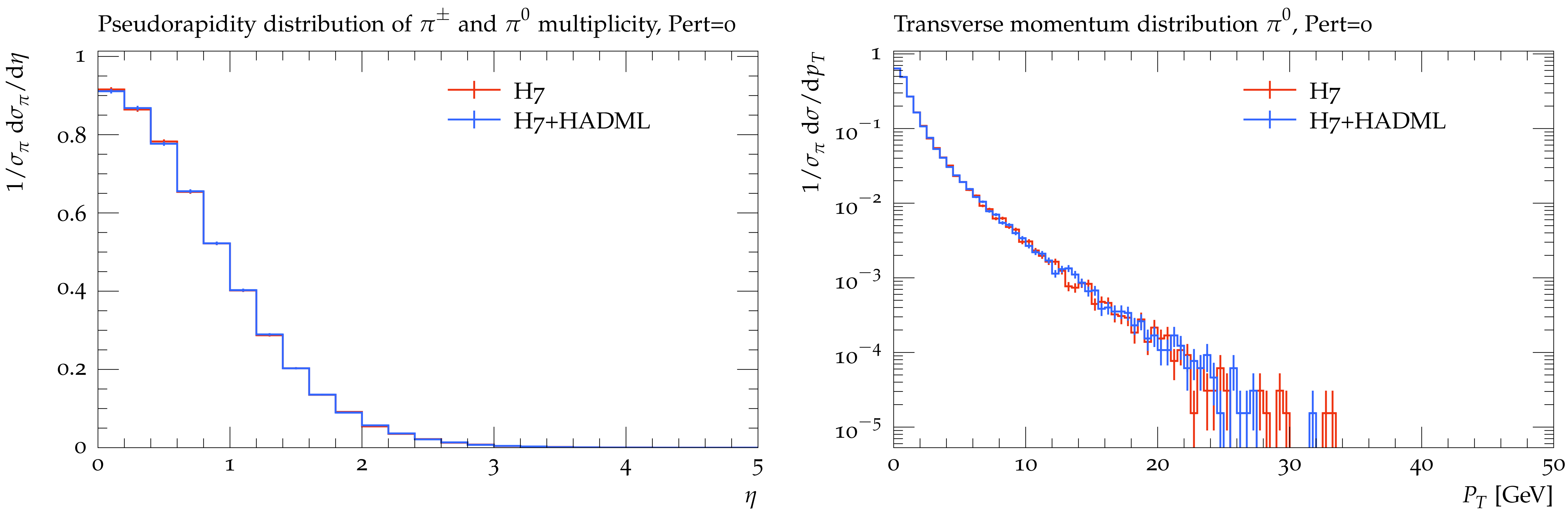


We have a conditional GAN, with cluster 4-vector input and two hadron 4-vector outputs.

We simplify this by considering only pions and generating two angles in the cluster rest frame.

This is a typical learning curve for GAN training.

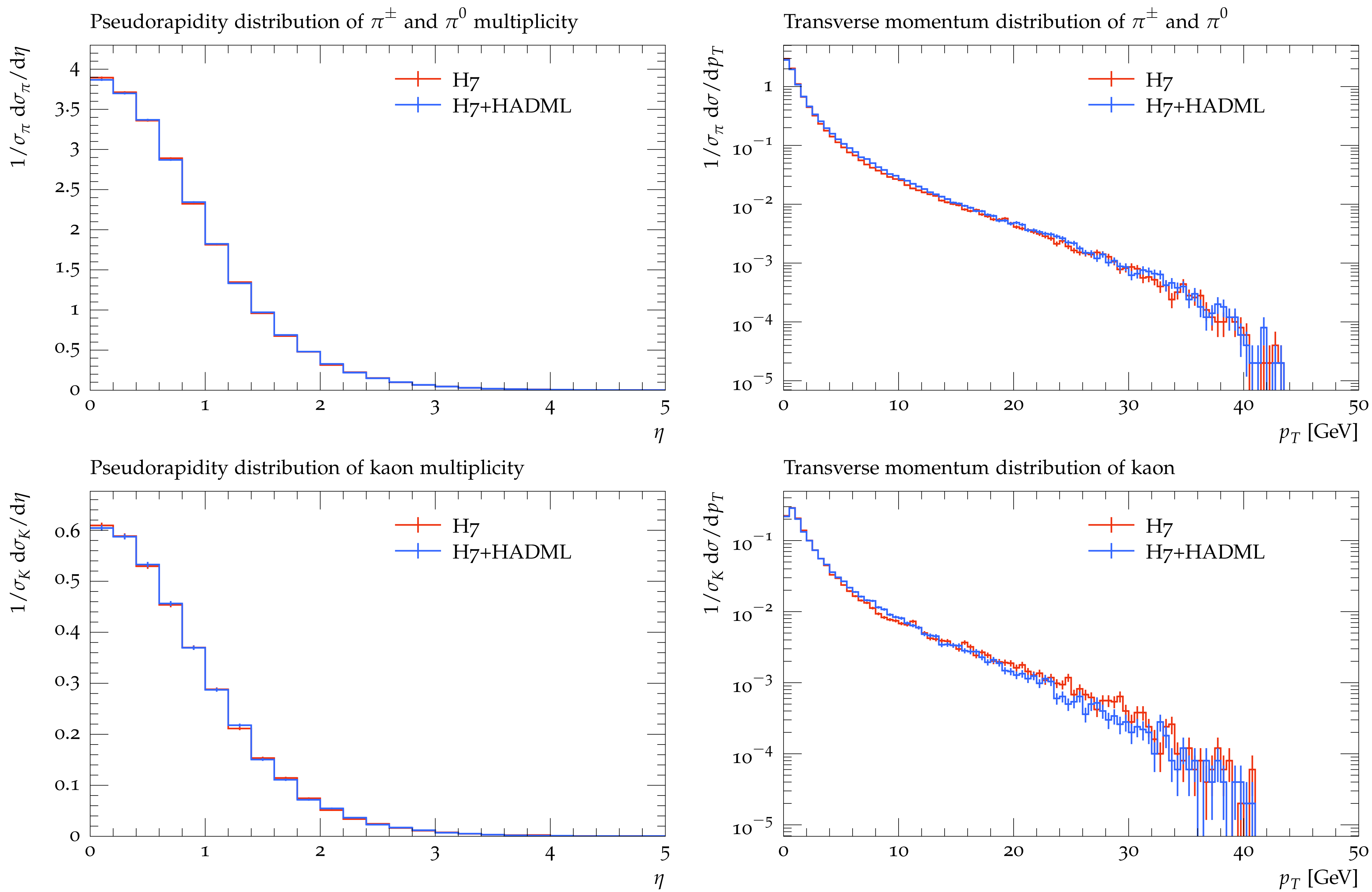
Performance: Pions



We extract clusters + hadrons, train, and then using ONNX, re-insert the model back into Herwig. This then allows us to run a full event generator and produce plots like these!

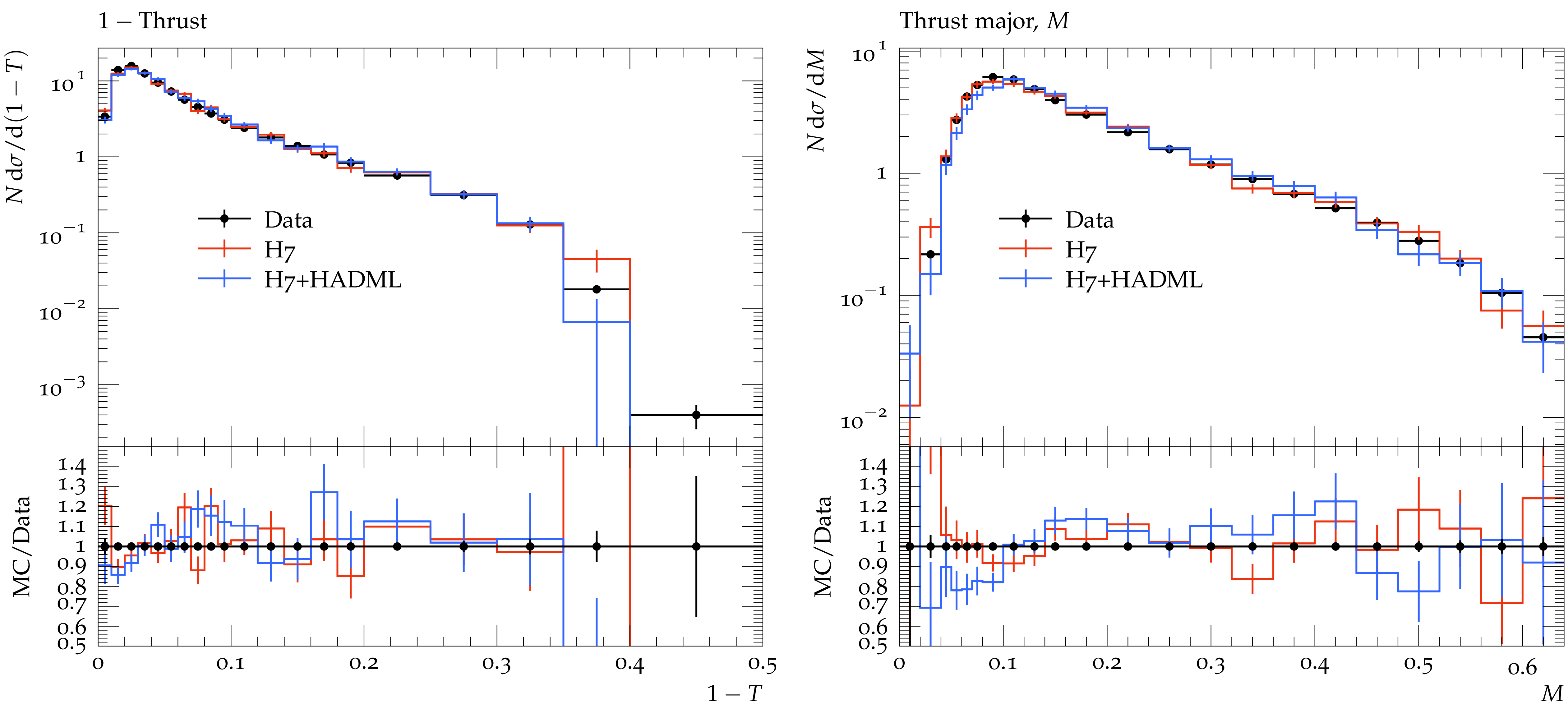
Performance: All Hadrons

As a crude "full" model, we simply take the PIDs from Herwig and the kinematics from the GAN.



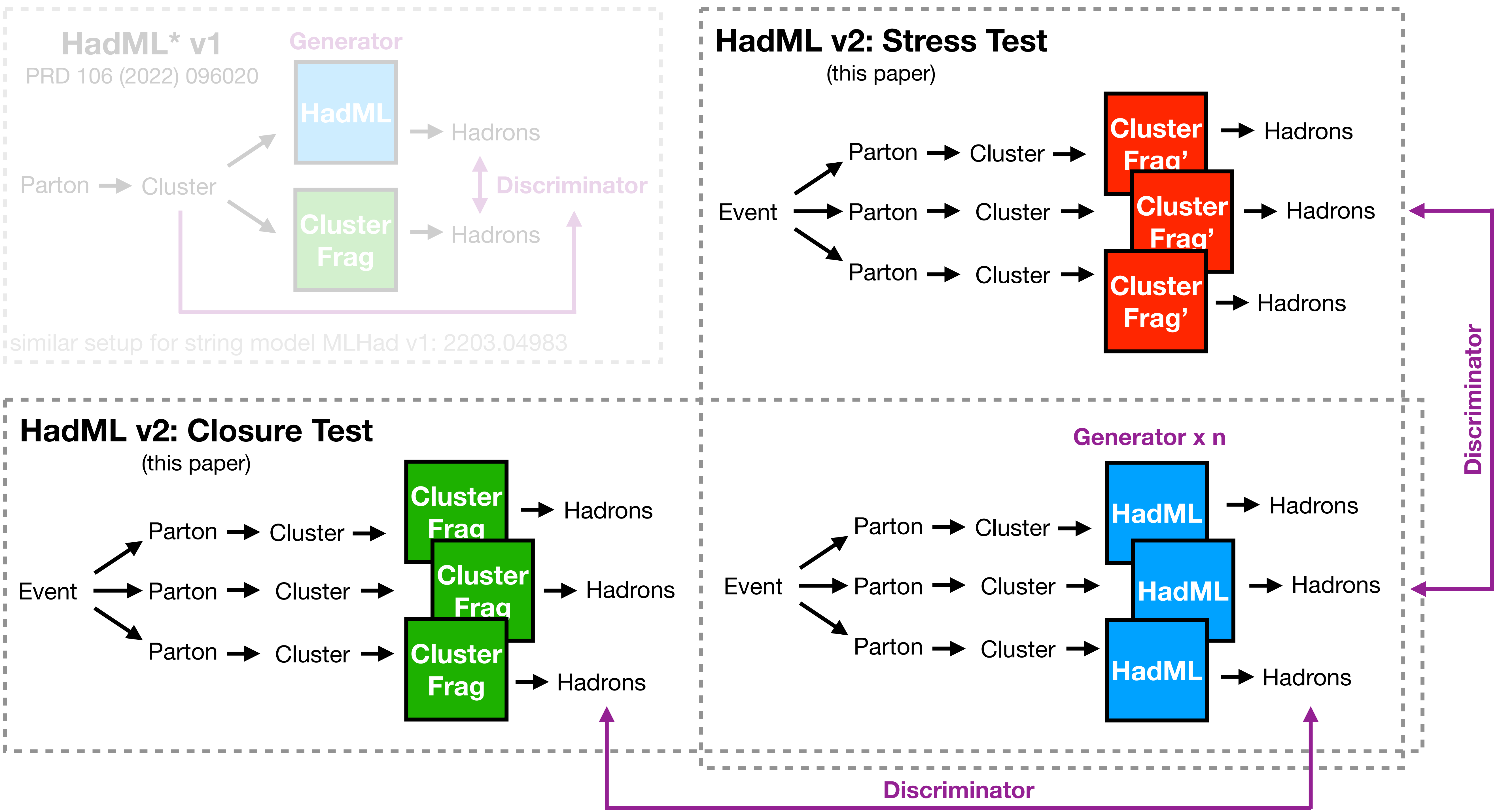
Performance: Data!

With a “full” model, we can compare directly to data!

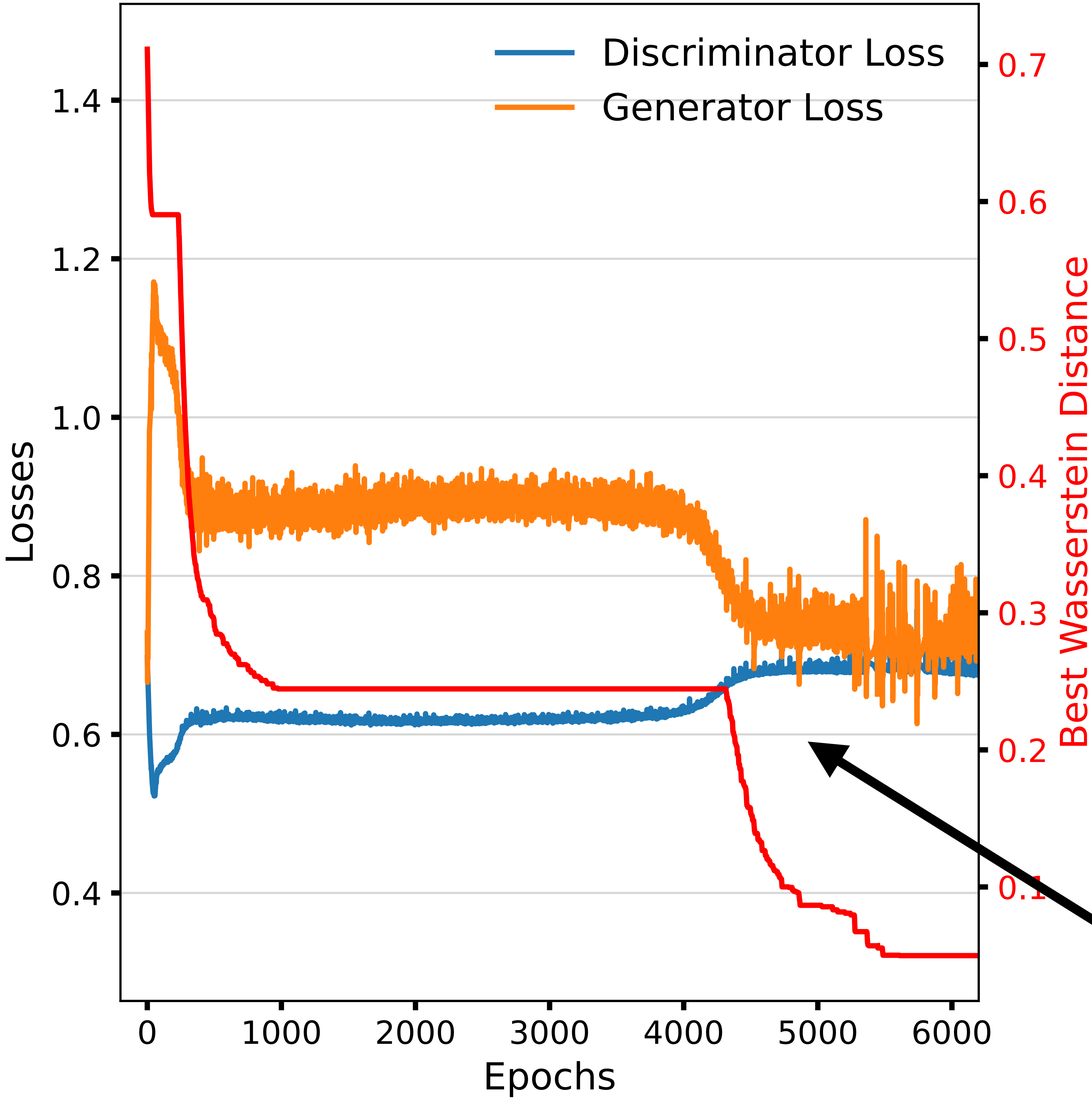


N.B. we have trained on H7, so we don’t expect to be any better than it at modeling the data.

ML Hadronization - Overview



Training HADML v2

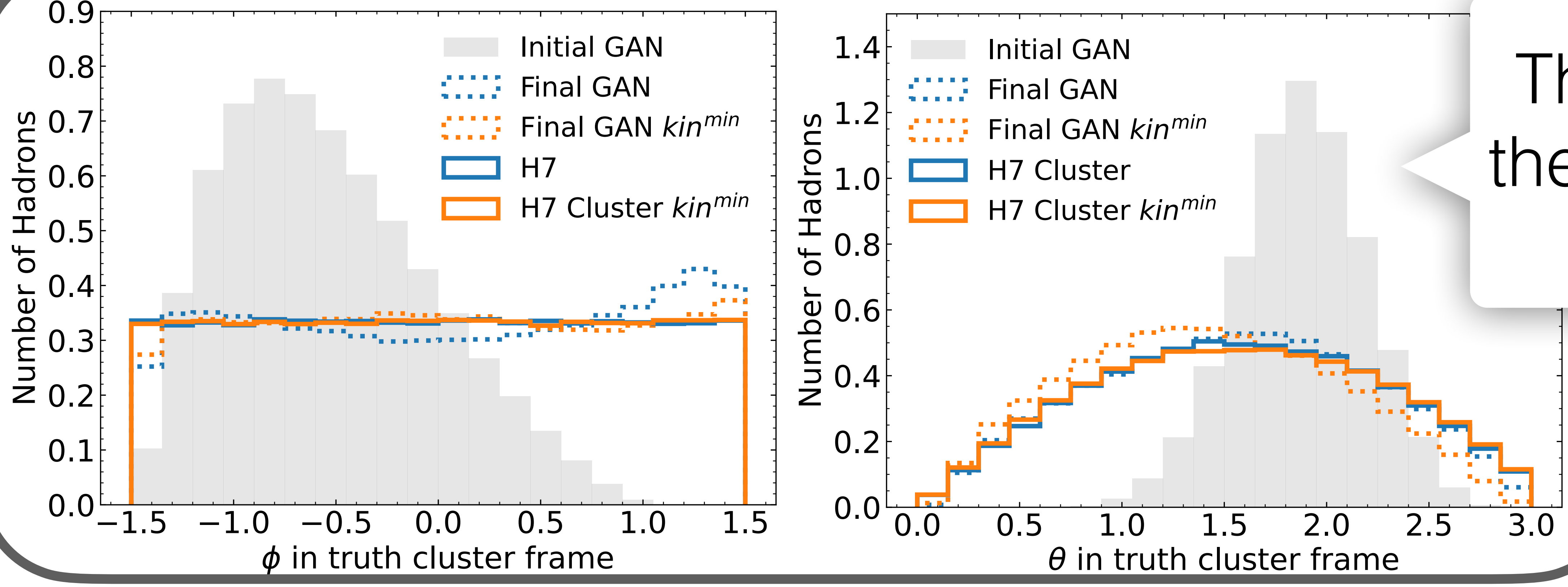


Now, the generator is local (per cluster), but the discriminator is global (whole event).

Discriminator is a permutation-invariant architecture called Deep Sets.

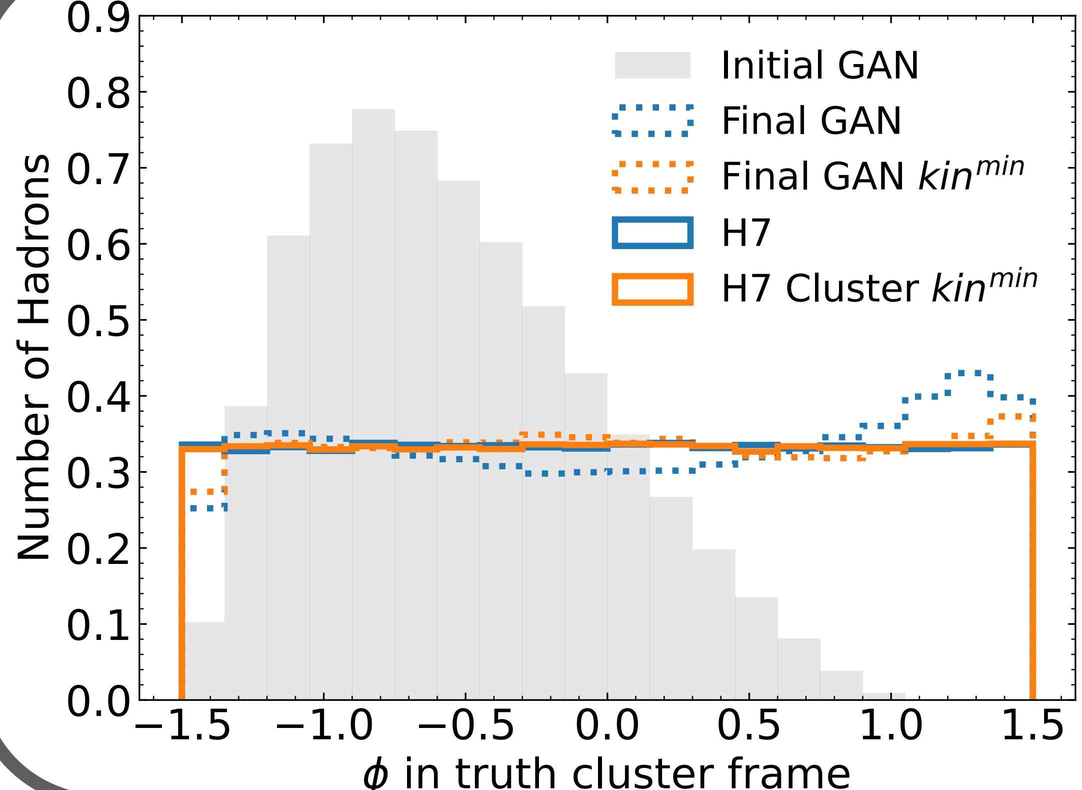
Still works !

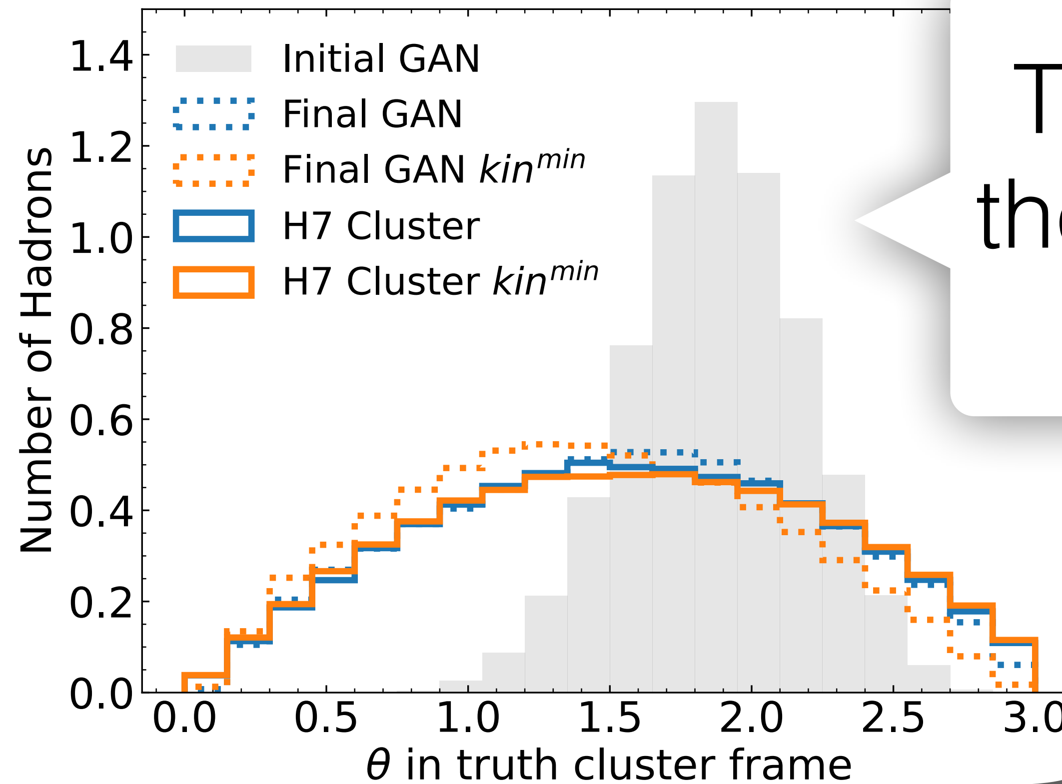
Performance



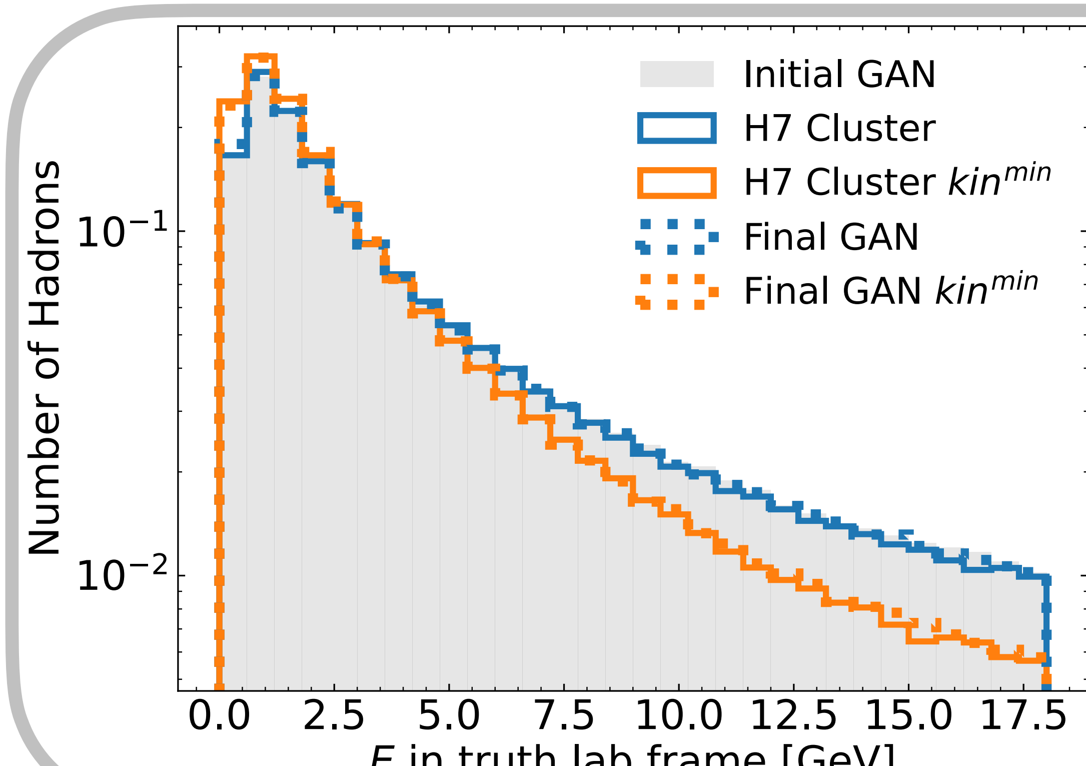
This is what the generator "sees"

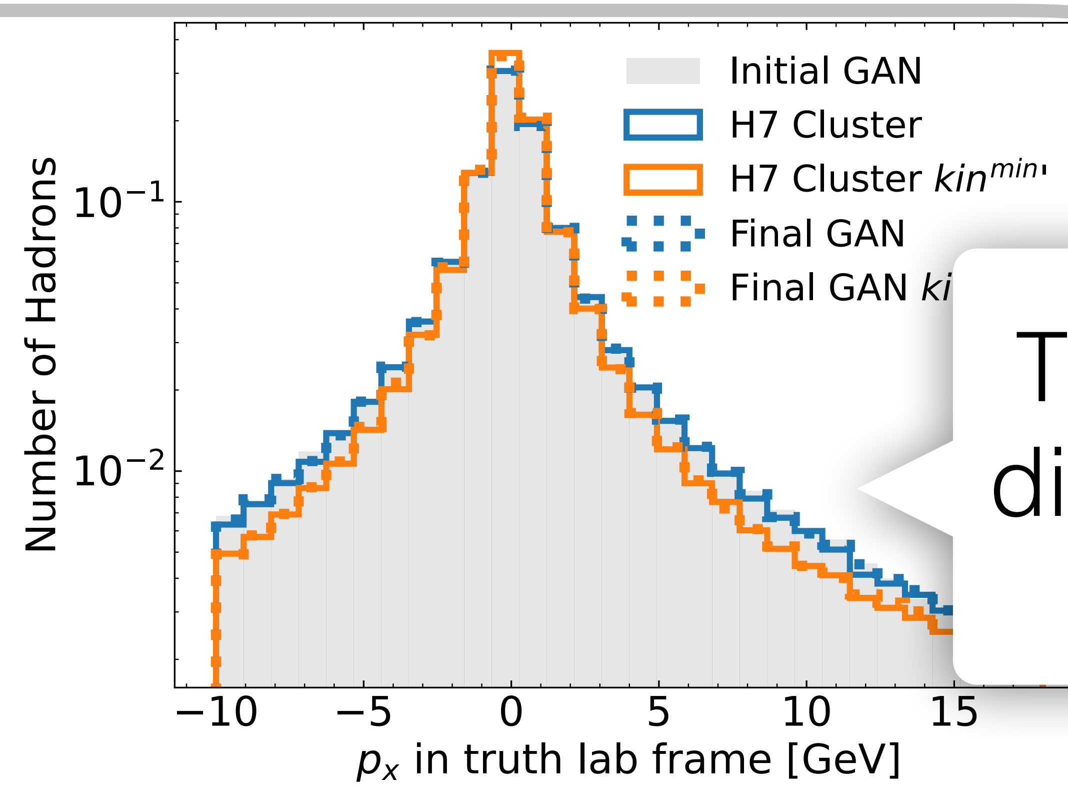
Performance





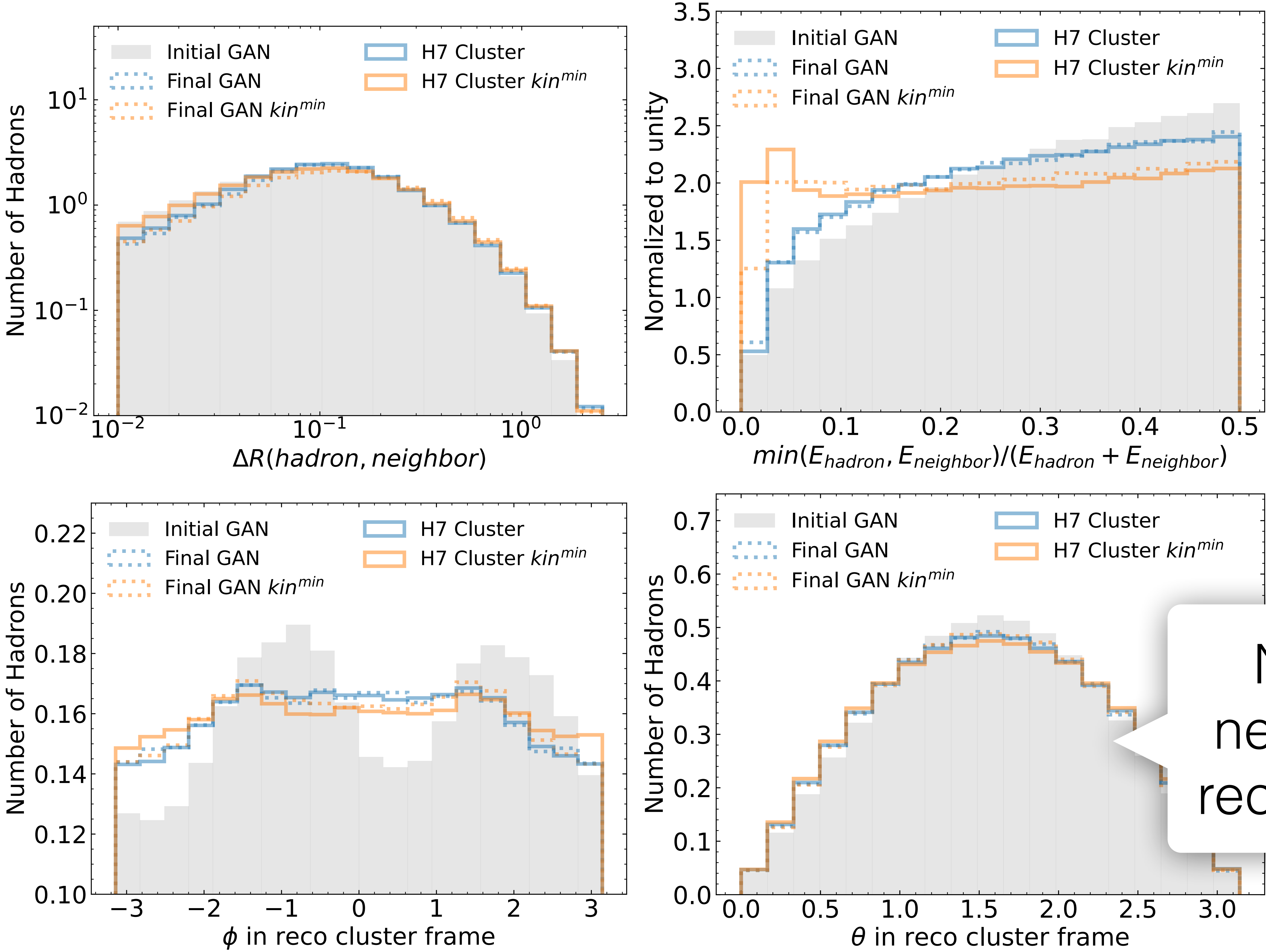
This is what the generator “sees”





This is what discriminator “sees”

Performance: reco. quantities



Nearest neighbor = reco. cluster

What is next?

Non-perturbative functions are natural candidates for ML-modeling to improve accuracy.

Long history (but still very active) for PDFs and now also progress for hadronization. **What about other non-perturbative objects?**

For hadronization, there is still clearly multiple steps before we have a product we can hand off to users, but it is **not science fiction !**

Precision Physics at the LHC

Theory of everything

Forward Model

Inverse Model

Physics simulators

Detector-level observables

Nature

Experimental design & control

Experiment

Online processing & quality control

Detector-level observables

Data curation

Inference

event selection
background estimation
hypothesis testing
…

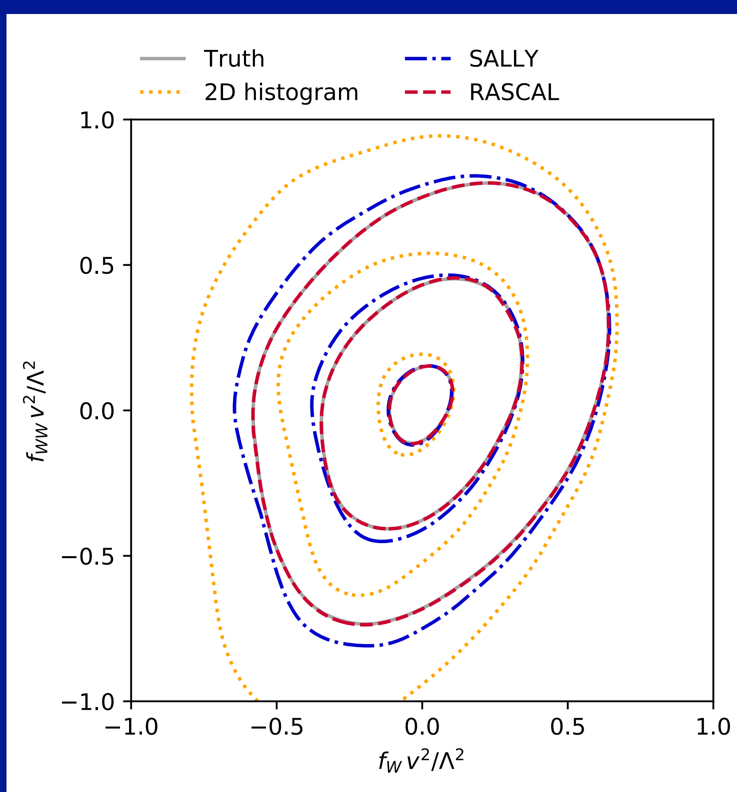
calibration
clustering
tracking
noise mitigation
particle identification
…

Connections to all areas!

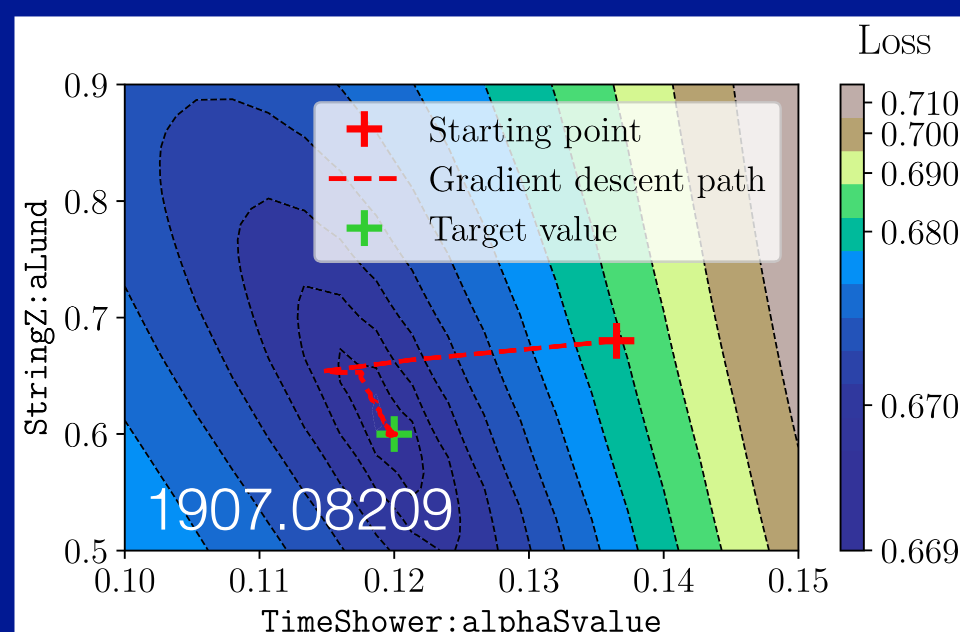
Plan for today: give representative examples from each area to illustrate the potential of ML. I'm not going to be comprehensive and I do not claim ML will solve all our problems!

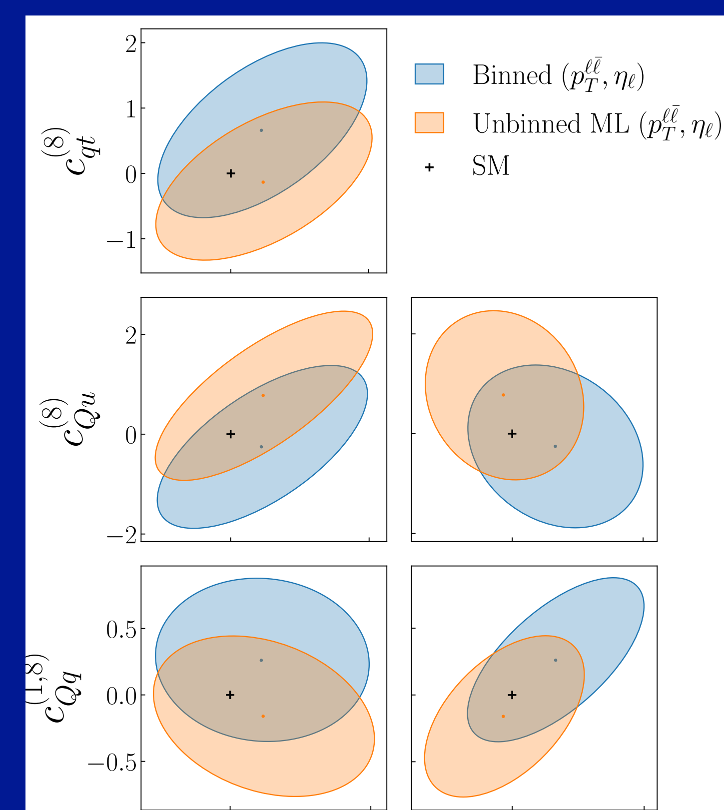
Enhancing the “inverse model”

There are many proposals for using ML to **directly infer model parameters**. This is a very exciting topic, but instead, I want to describe a complementary program on **highly differential cross sections**.



1805.00013

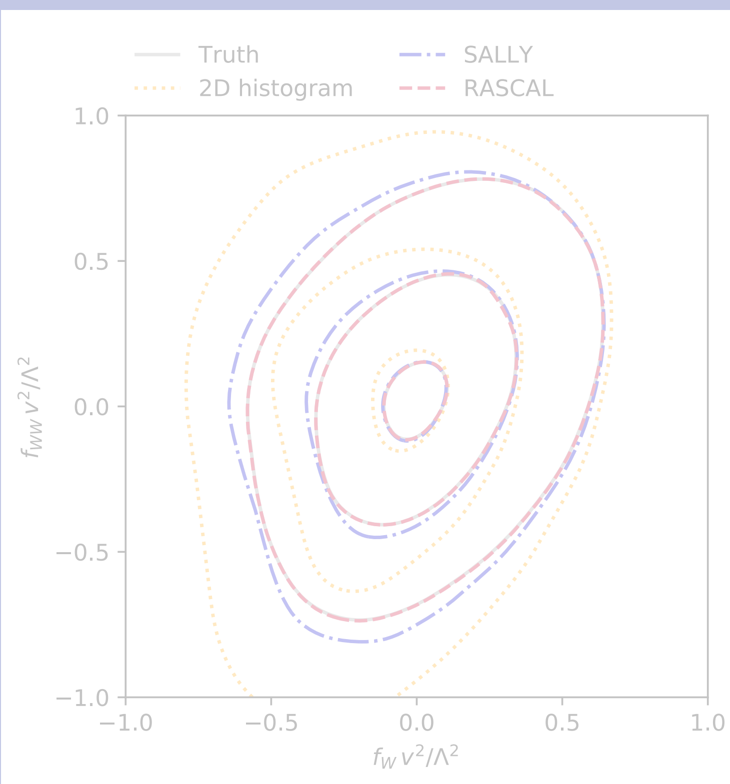




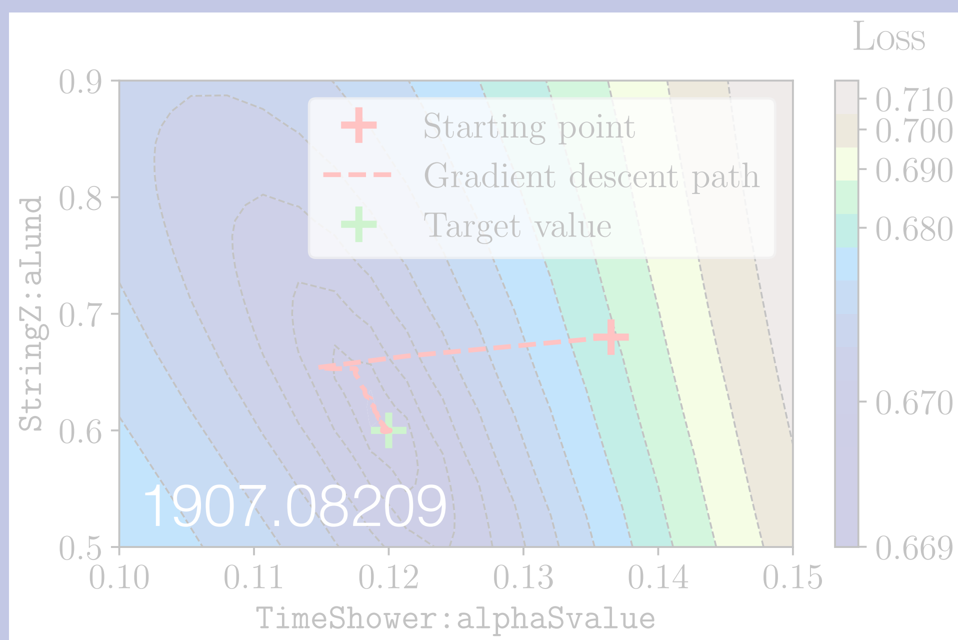
2211.02058

Enhancing the "inverse model"

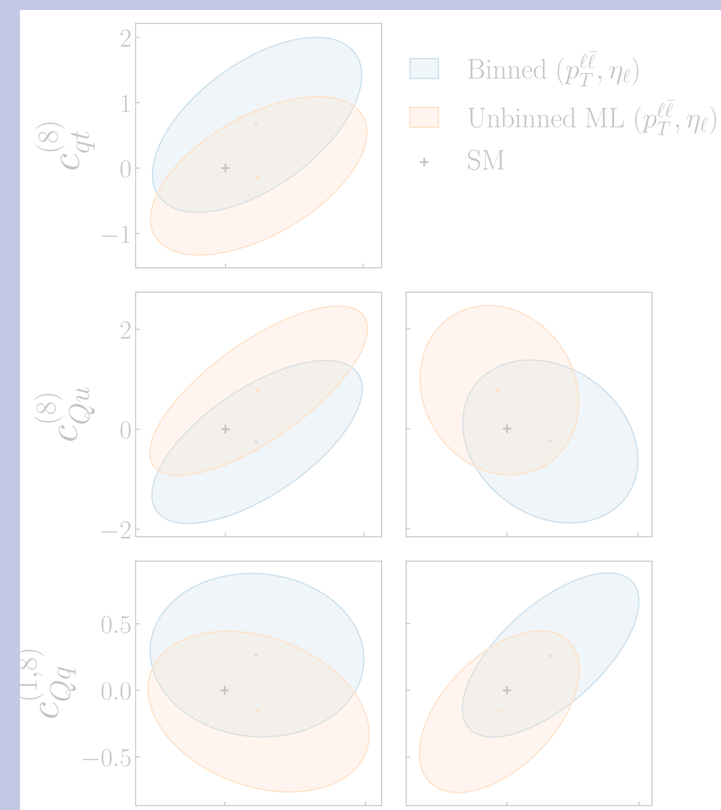
There are many proposals for using ML to **directly infer model parameters**. This is a very exciting topic, but instead, I want to describe a complementary program on **highly differential cross sections**.



1805.00013



1907.08209



2211.02058

Unfolding

Deconvolution ("unfolding"):
correcting for detector effects

Unfolding

Deconvolution ("unfolding"):
correcting for detector effects

Key aspect of all cross section measurements, across particle/nuclear/astro physics (!)

Unfolding

Proton-Proton

Nucleus-Nucleus

Electron-Proton

Neutrino-Nucleus

Electron-Positron

Cosmic Rays

Particle/Nuclear/Astro Physics Experiments

Deconvolution ("unfolding"): correcting for detector effects

Key aspect of **all** cross section measurements, across particle/nuclear/astro physics (!)

Unfolding

Particle/Nuclear/Astro Physics Experiments

Deconvolution ("unfolding"):
correcting for detector effects

Key aspect of **all** cross section measurements, across particle/ nuclear/astro physics (!)

Why "unfold" instead of "fold"?

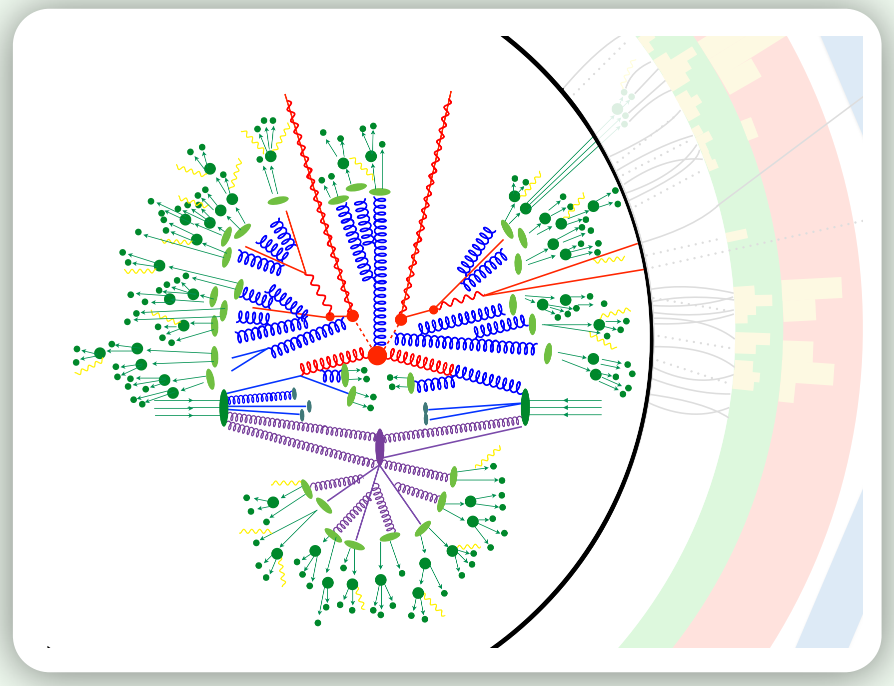
Unfolding is ill-posed, BUT only way to compare different experiments and to compare with non fully exclusive predictions. Data also survive much longer.

The Unfolding Challenge

The Unfolding Challenge

Particle Level

Want this



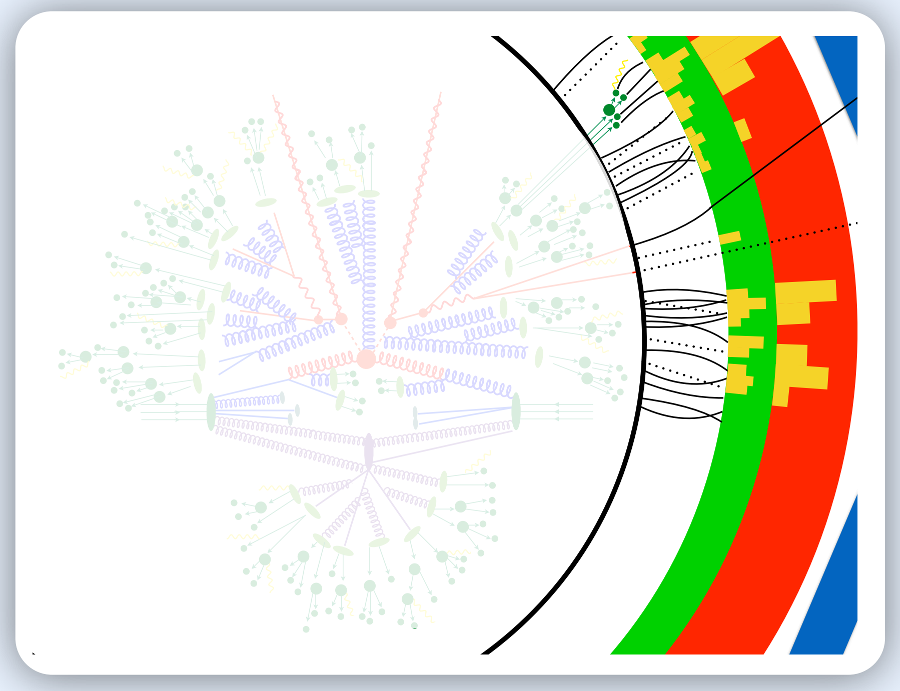
2203.16722

The Unfolding Challenge

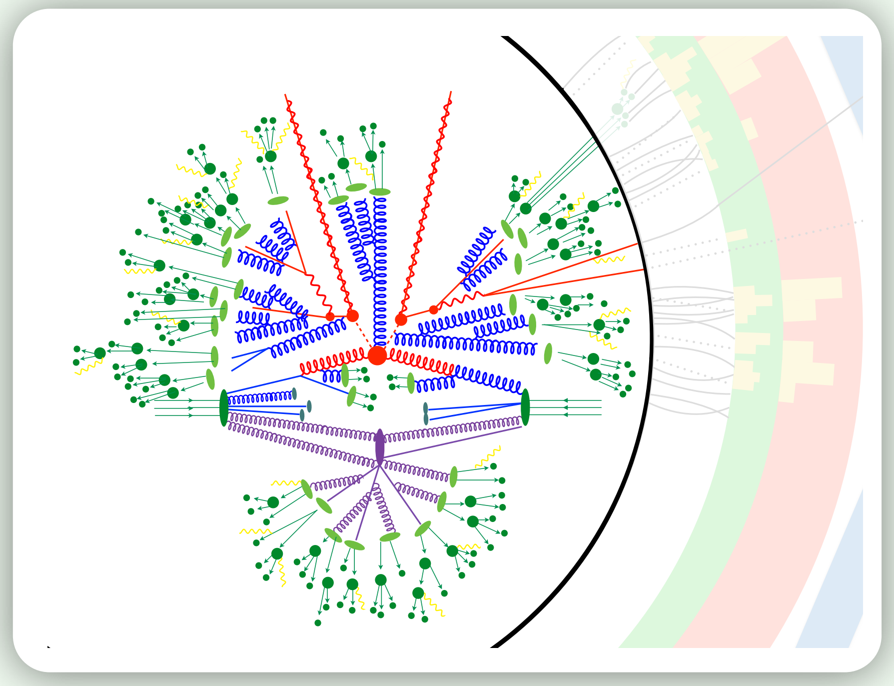
Detector Level

Particle Level

Measure this



Want this



2203.16722

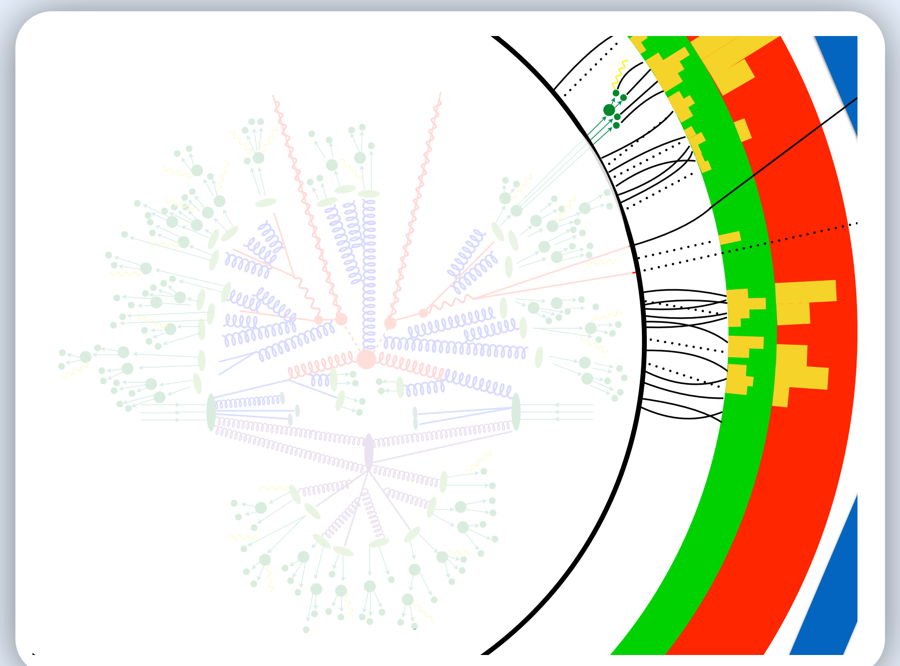
The Unfolding Challenge

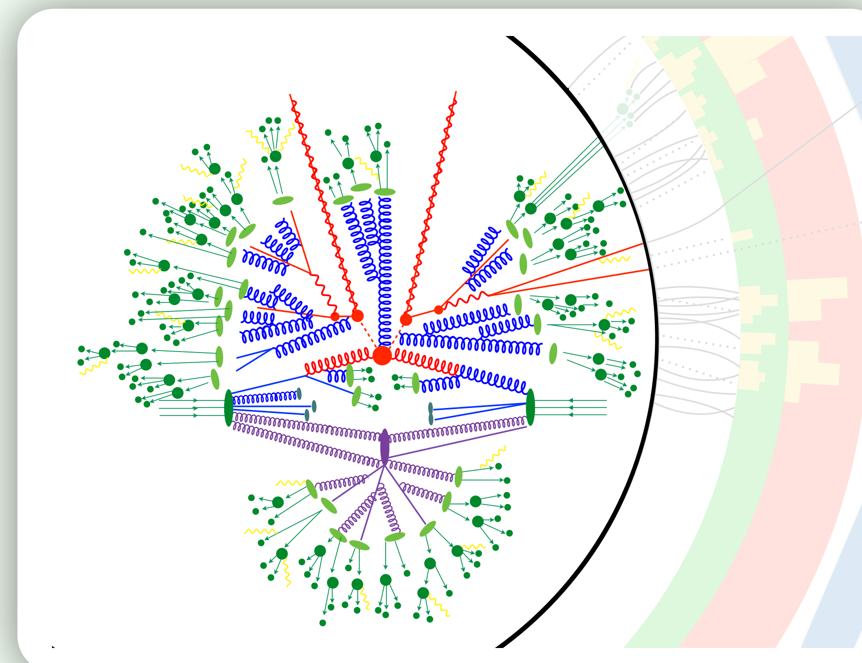
Detector Level

Particle Level

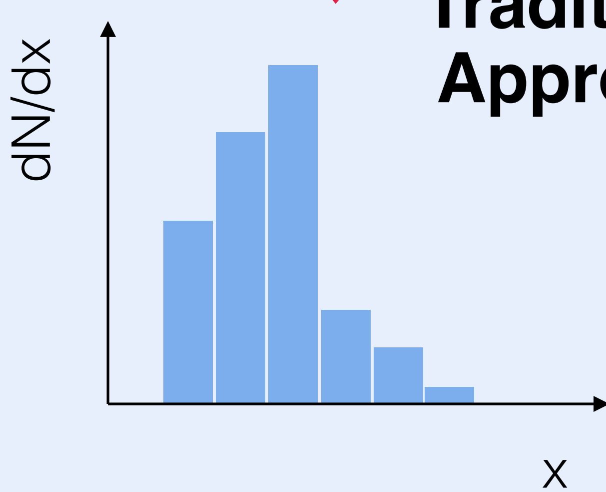
Measure this

Want this



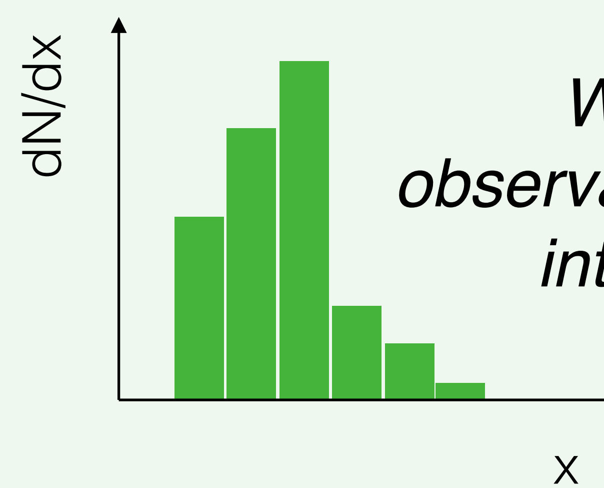


Target observable



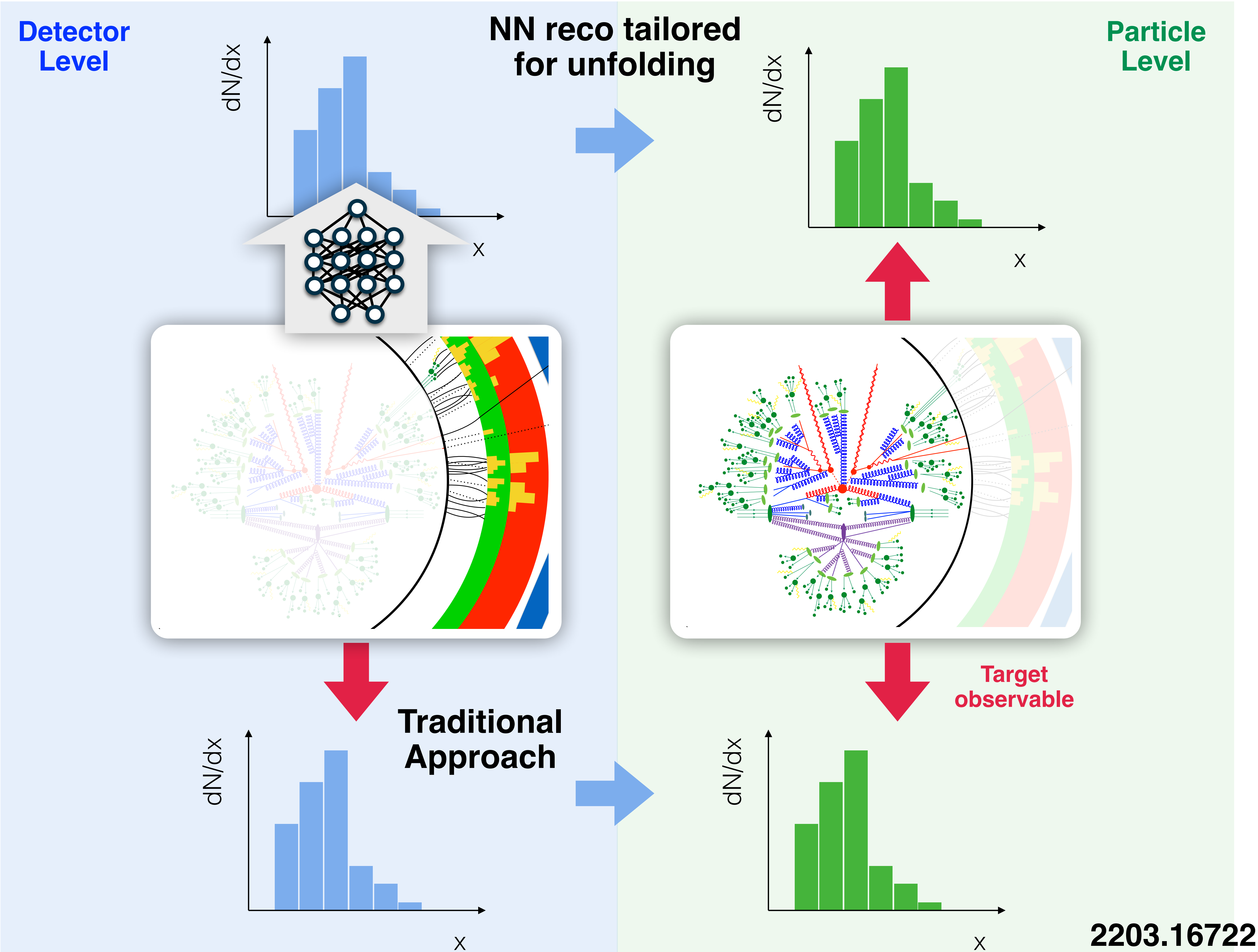
Traditional Approach

$$\hat{T} \approx R^{-1}M$$

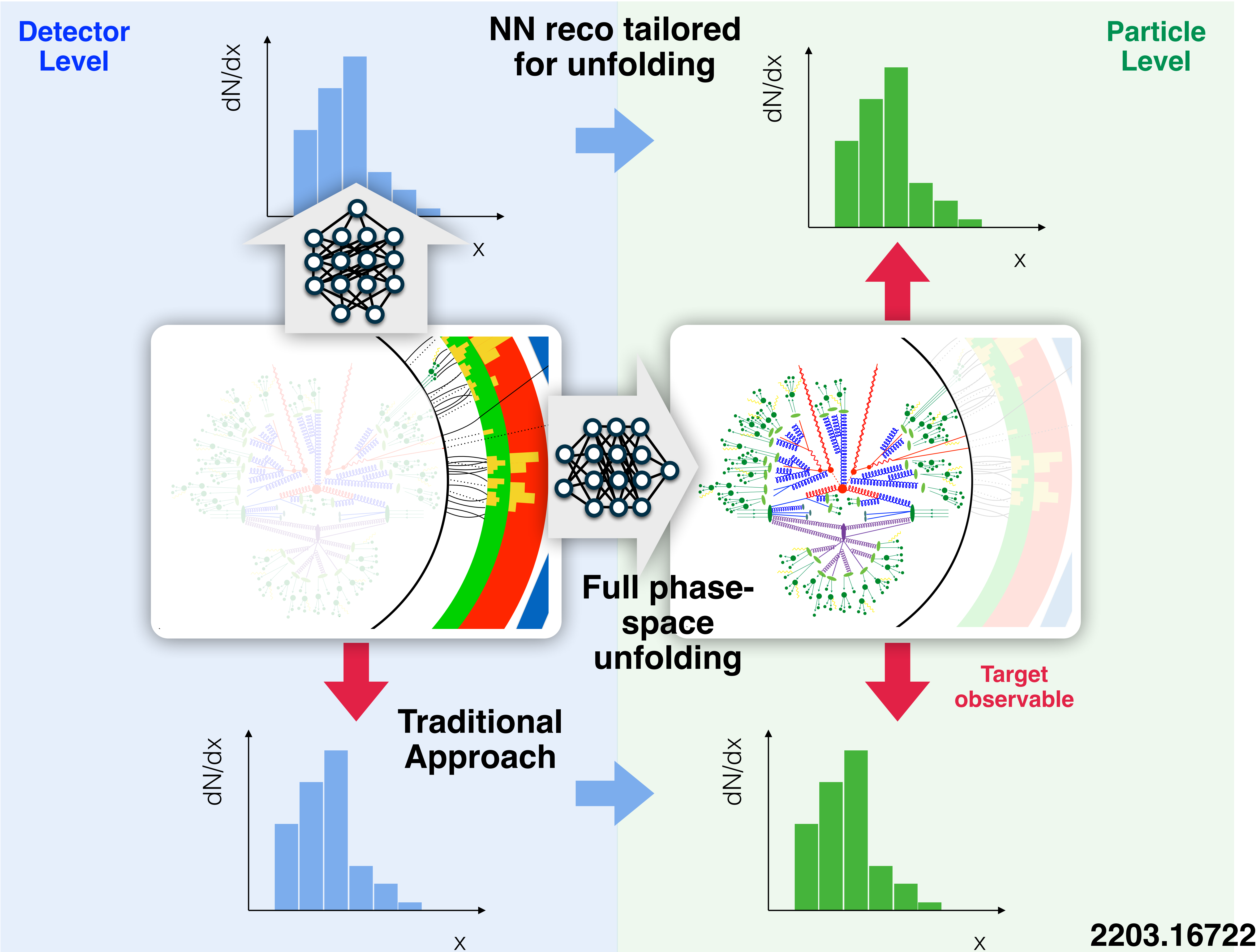


We pick O(1) observables & discretize into O(10) bins

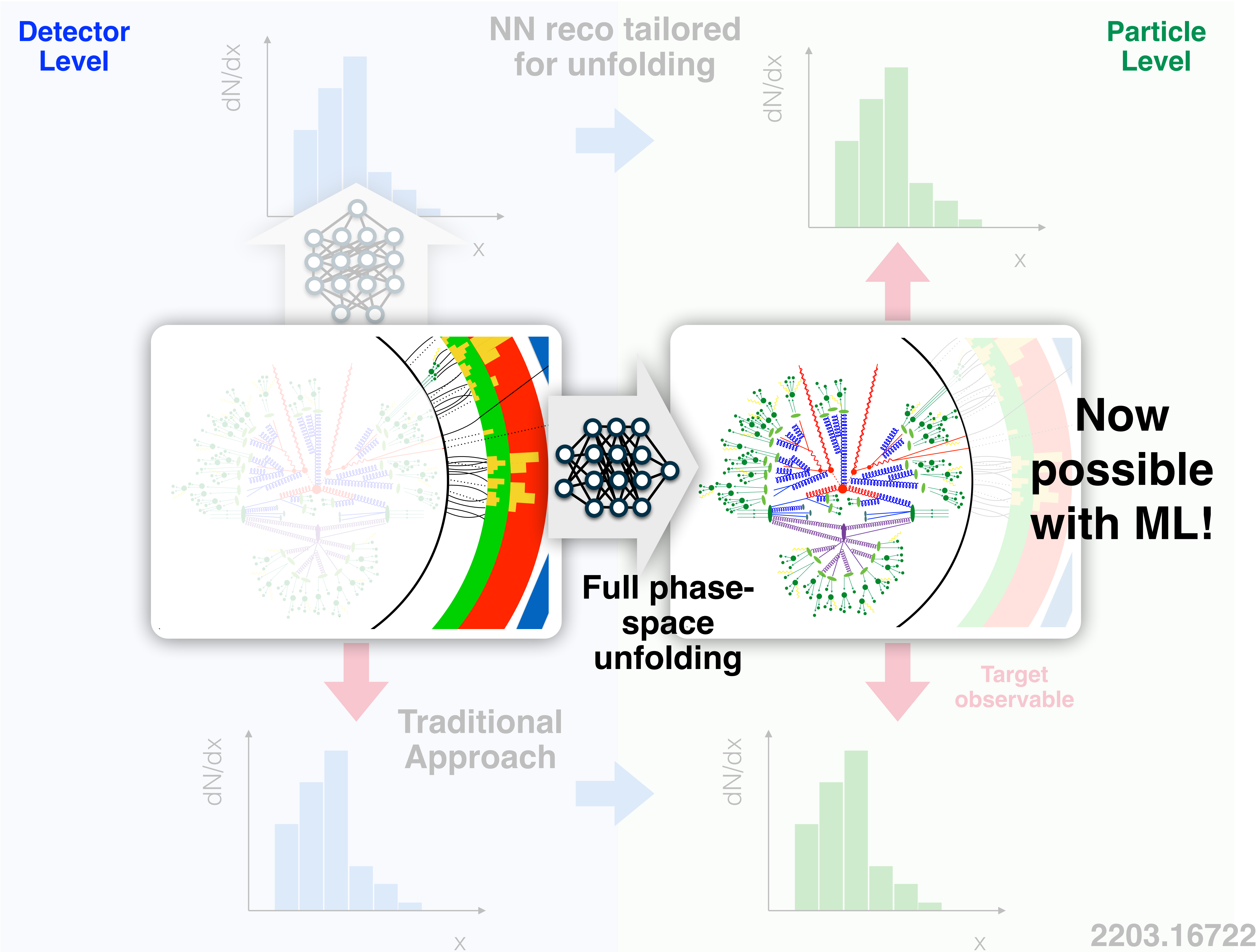
2203.16722



Detector Level
NN reco tailored for unfolding
Particle Level
dN/dx
x
dN/dx
x
Traditional Approach
Target observable
dN/dx
x
dN/dx
x
2203.16722



Detector Level
NN reco tailored for unfolding
Particle Level
dN/dx
x
dN/dx
x
Full phase-space unfolding
Target observable
Traditional Approach
dN/dx
x
dN/dx
x
2203.16722


Detector Level
NN reco tailored for unfolding
Particle Level
dN/dx
x
Now possible with ML!
Full phase-space unfolding
Target observable
Traditional Approach
2203.16722

Why unbinned (+high-dimensional)?

For a community white paper, see JINST 17 (2022) P01024, 2109.13243

Why unbinned (+high-dimensional)?

Inference-Aware Binning

Optimal binning depends on downstream task. Not possible with current setup.

What about moments?
(see e.g. this paper)

For a community white paper, see JINST 17 (2022) P01024, 2109.13243

Why unbinned (+high-dimensional)?

Inference-Aware Binning

Optimal binning depends on downstream task. Not possible with current setup.

What about moments?
(see e.g. this paper)

Derivative Measurements

With binned measurements, essentially impossible to re-use results for a function of the phase space.

For a community white paper, see JINST 17 (2022) P01024, 2109.13243

Why unbinned (+high-dimensional)?

Inference-Aware Binning

Optimal binning depends on downstream task. Not possible with current setup.

What about moments?

(see e.g. this paper)

Derivative Measurements

With binned measurements, essentially impossible to re-use results for a function of the phase space.

Higher Dimensions

Some phenomena can't be probed in a few dimensions.

What about observables that are not per-event?

For a community white paper, see JINST 17 (2022) P01024, 2109.13243

Landscape of Methods

Classifier-Based Methods

Learn (unfolded) data likelihood ratio w.r.t. simulation

For references, see JINST 17 (2022) P01024, 2109.13243 (and papers that cite it!)

Landscape of Methods

Classifier-Based Methods

Learn (unfolded) data likelihood ratio w.r.t. simulation

Density-Based Methods

Learn (unfolded) data probably density implicitly or explicitly.

For references, see JINST 17 (2022) P01024, 2109.13243 (and papers that cite it!)

Landscape of Methods

Classifier-Based Methods

Learn (unfolded) data likelihood ratio w.r.t. simulation

Density-Based Methods

Learn (unfolded) data probably density implicitly or explicitly.

Both methods work with various pros/cons that I won't get into here.

The examples I give are based on the classifier approach.

For references, see JINST 17 (2022) P01024, 2109.13243 (and papers that cite it!)

Likelihood ratios

How do we learn LLRs without binning?

dataset 1: sampled from ***p(x)***
dataset 2: sampled from ***q(x)***

Create weights ***w(x) = q(x)/p(x)*** so that when dataset 1 is weighted by ***w***, it is statistically identical to dataset 2.

What if we don't (and can't easily) know ***q*** and ***p***?

(and don't want to estimate them by binning)

Classification = likelihood ratios

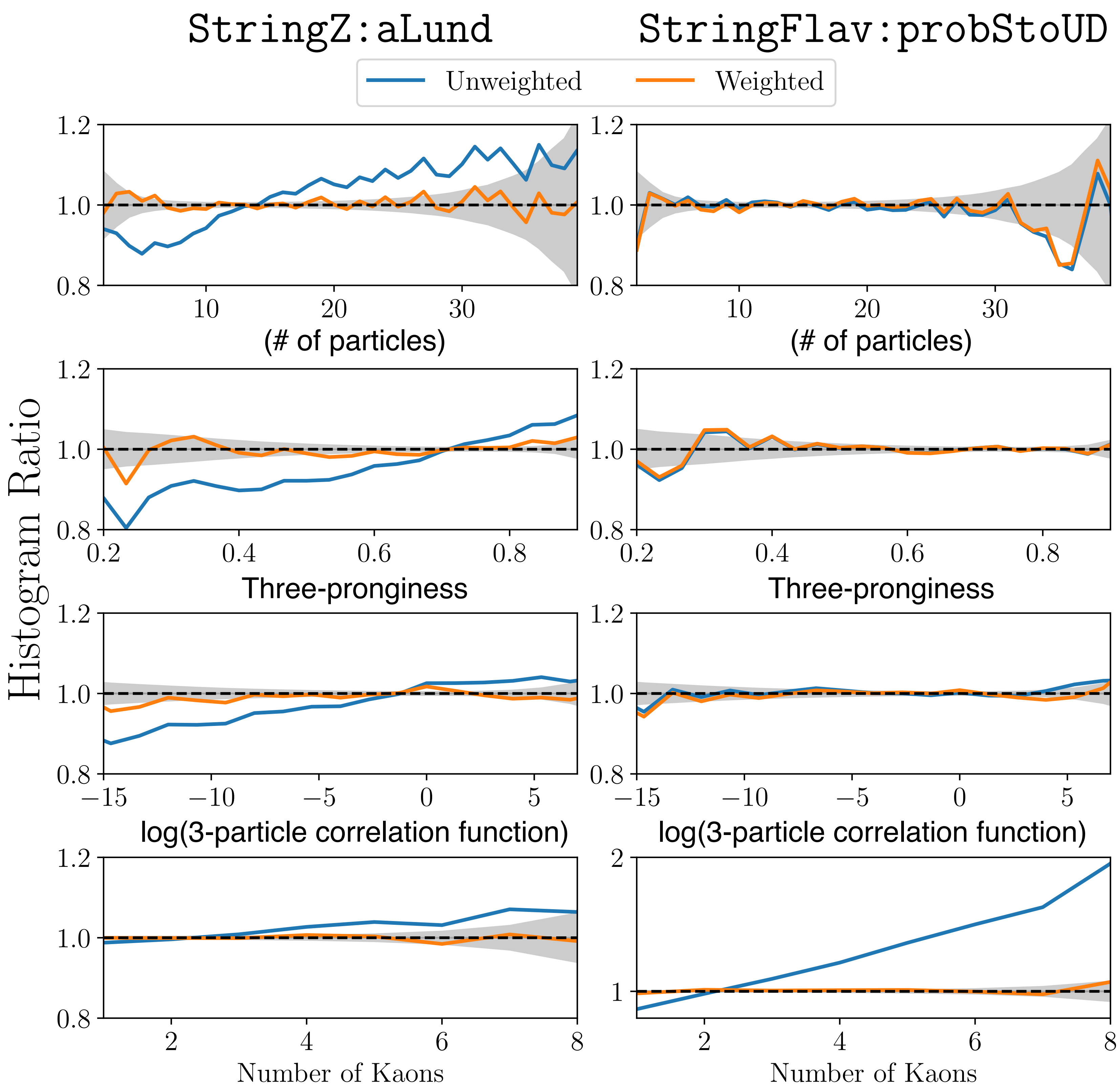
Fact: Neutral networks learn to approximate the likelihood ratio = $q(x)/p(x)$

(or something monotonically related to it in a known way)

Solution: train a neural network to distinguish the two datasets!

This turns the problem of **density estimation** (**hard**) into a problem of **classification** (**easy**)

Neural reweighting: works !



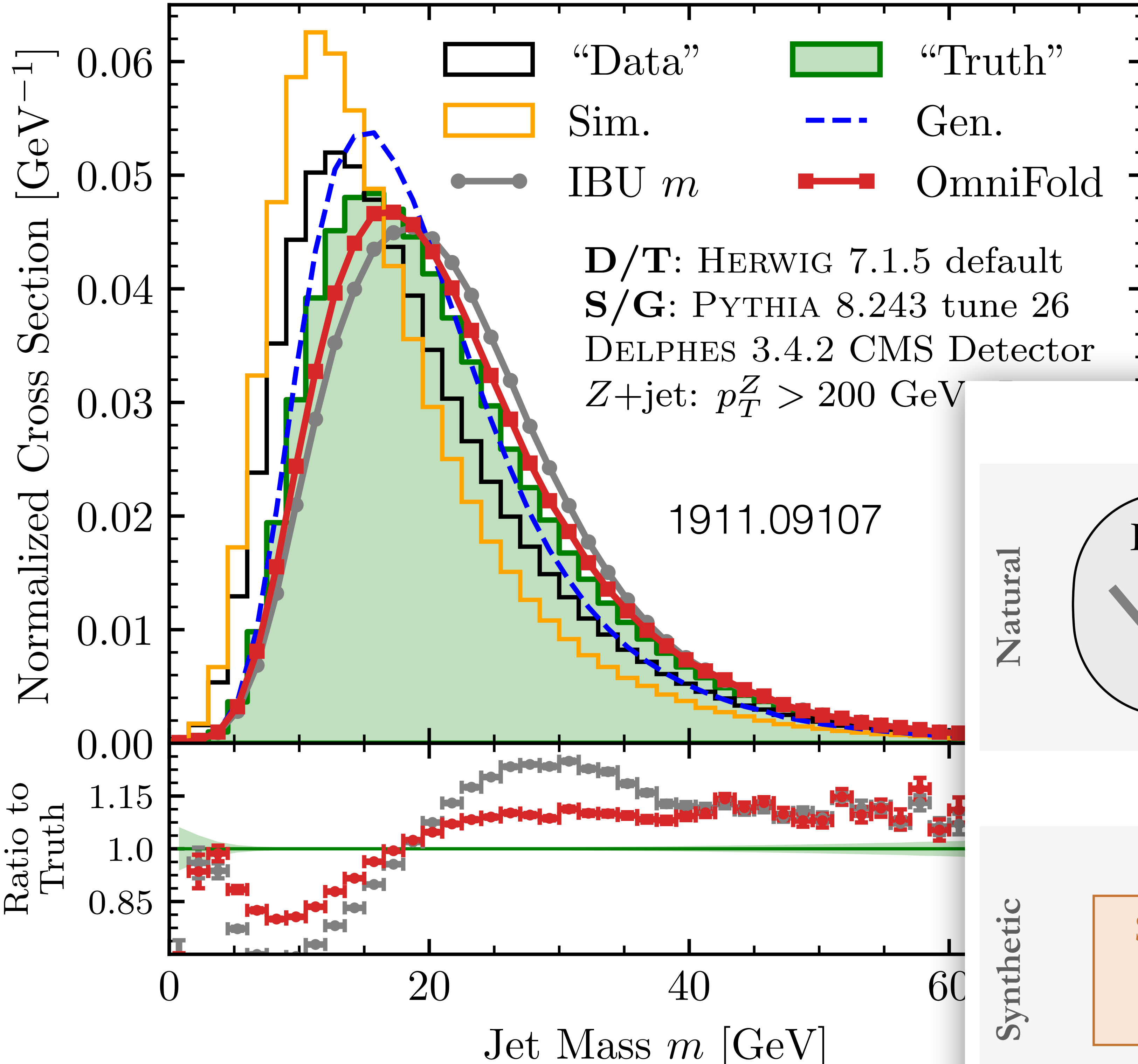
Full phase-space reweighing using simulated e^+e^-

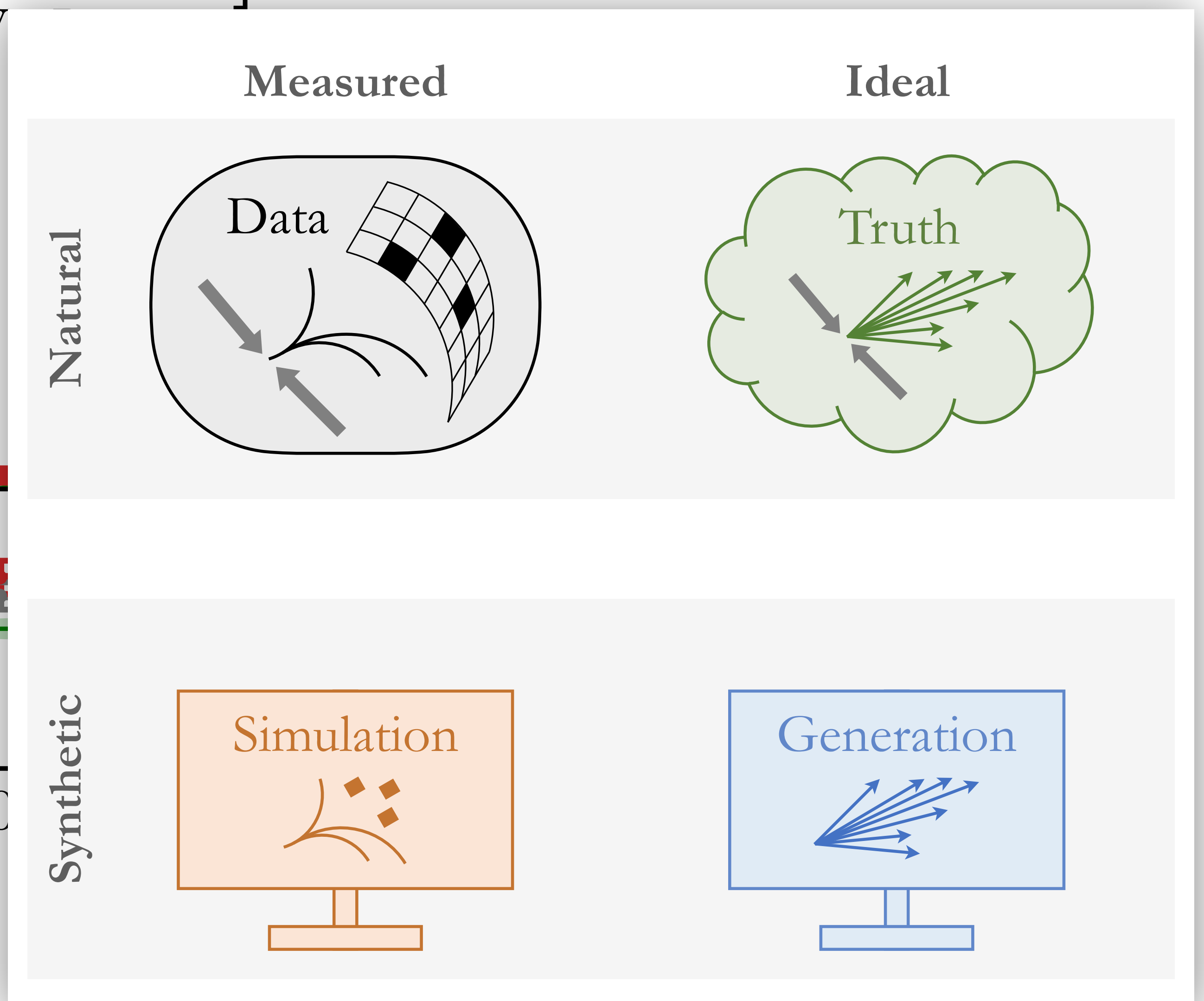
Works even when the differences are **small** (left) or **localized** (right).

These are histogram ratios for a series of one-dimensional observables

1907.08209

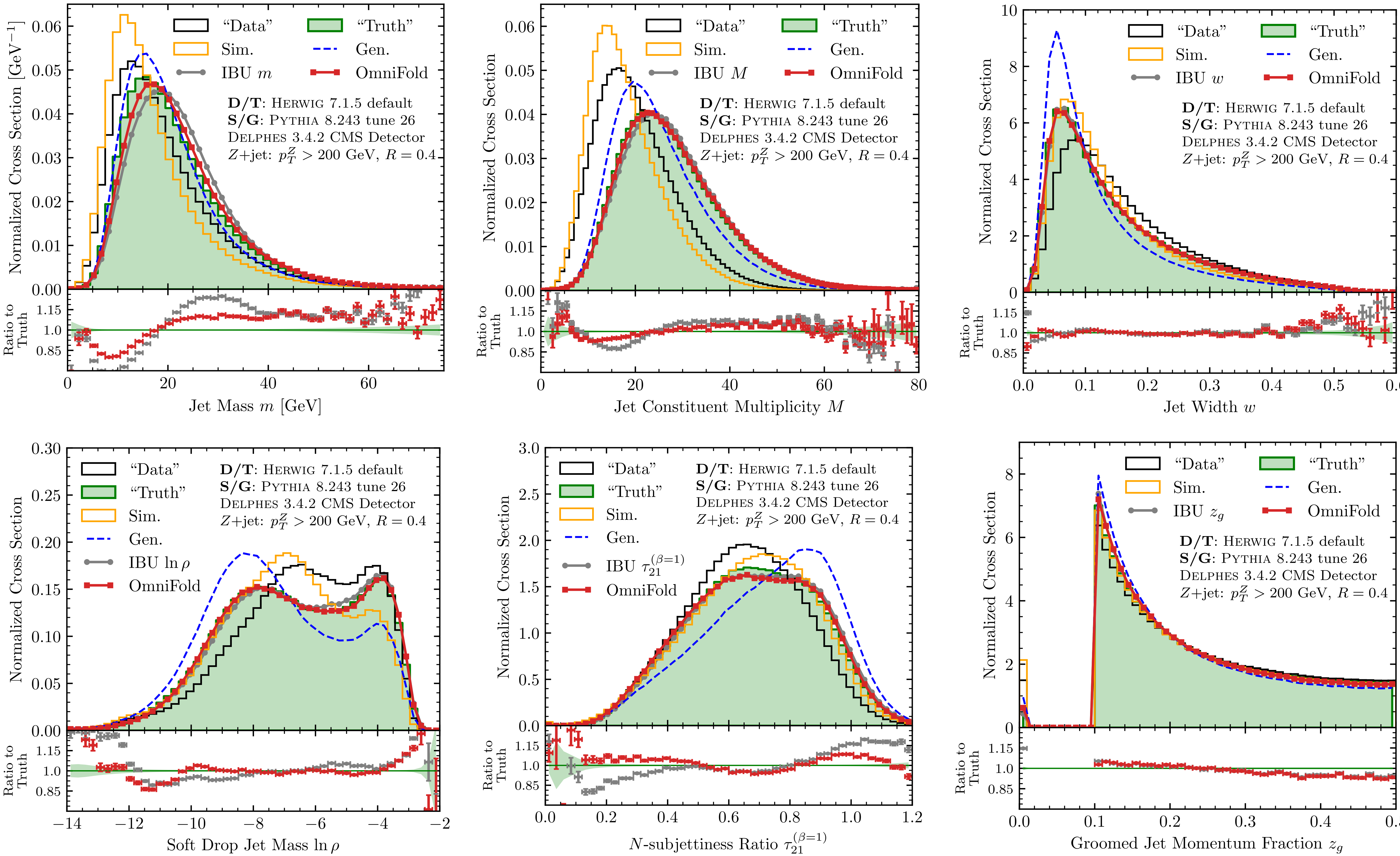
Full phase-space unfolding in action





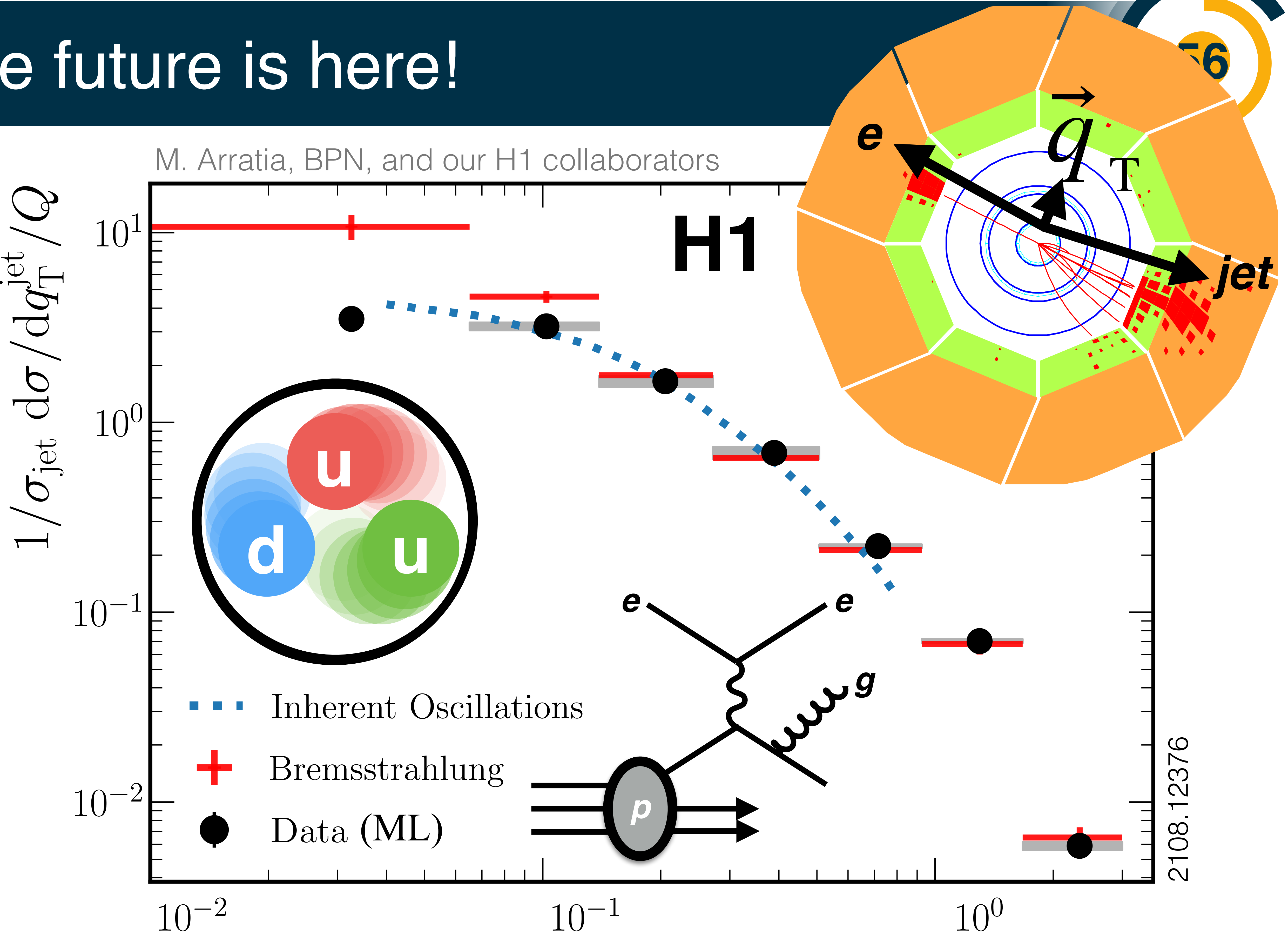
Full phase-space unfolding in action

1911.09107



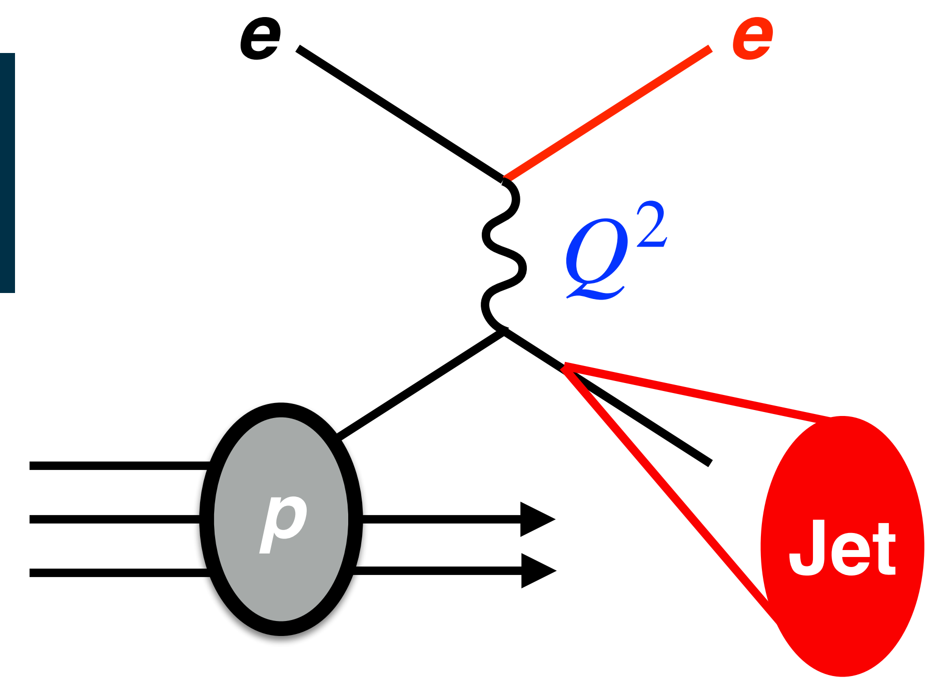
The future is here!

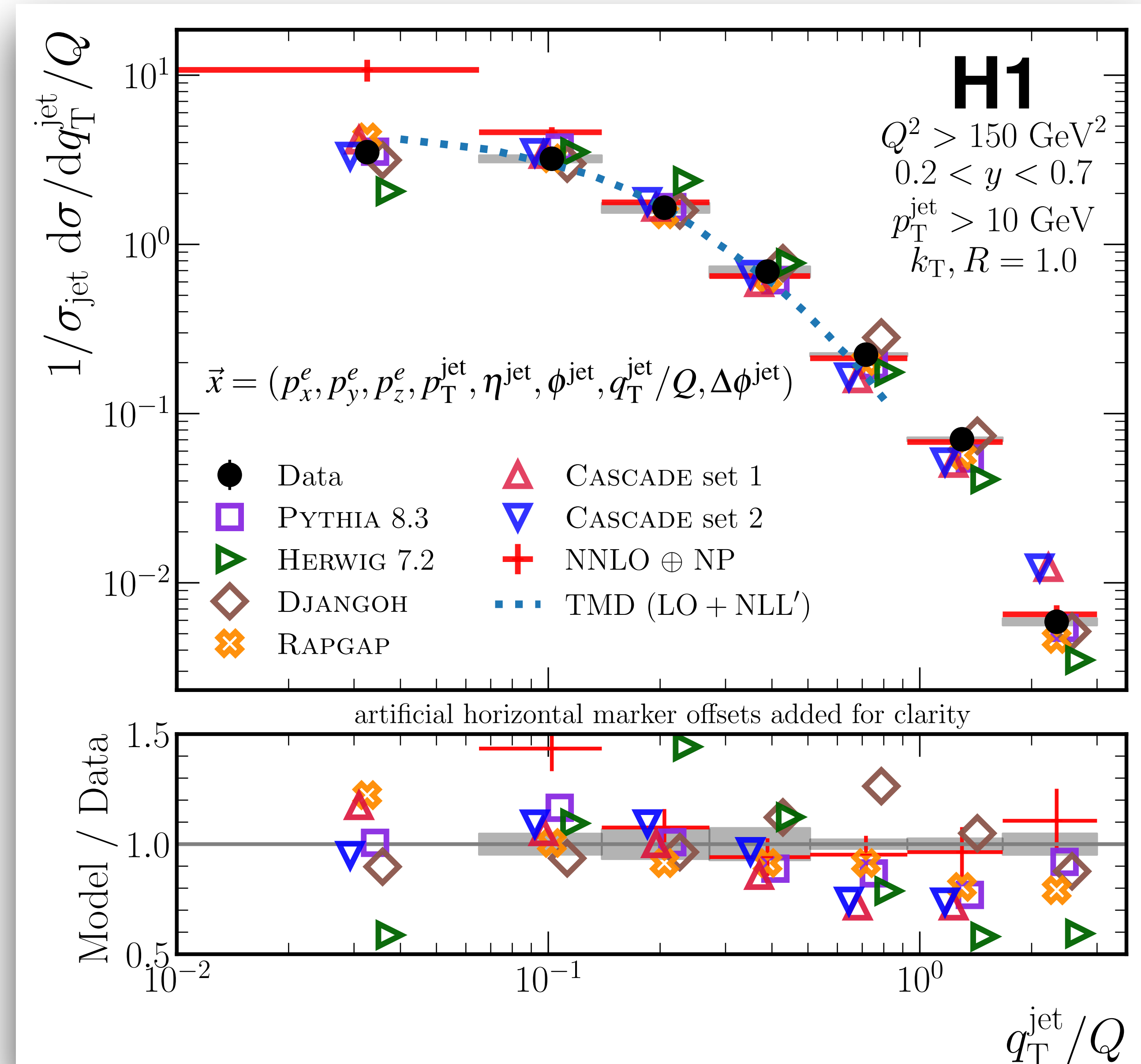
M. Arratia, BPN, and our H1 collaborators



$$\vec{x} = (p_x^e, p_y^e, p_z^e, p_{\mathrm{T}}^{\text{jet}}, \eta^{\text{jet}}, \phi^{\text{jet}}, q_{\mathrm{T}}^{\text{jet}}/Q, \Delta\phi^{\text{jet}})$$

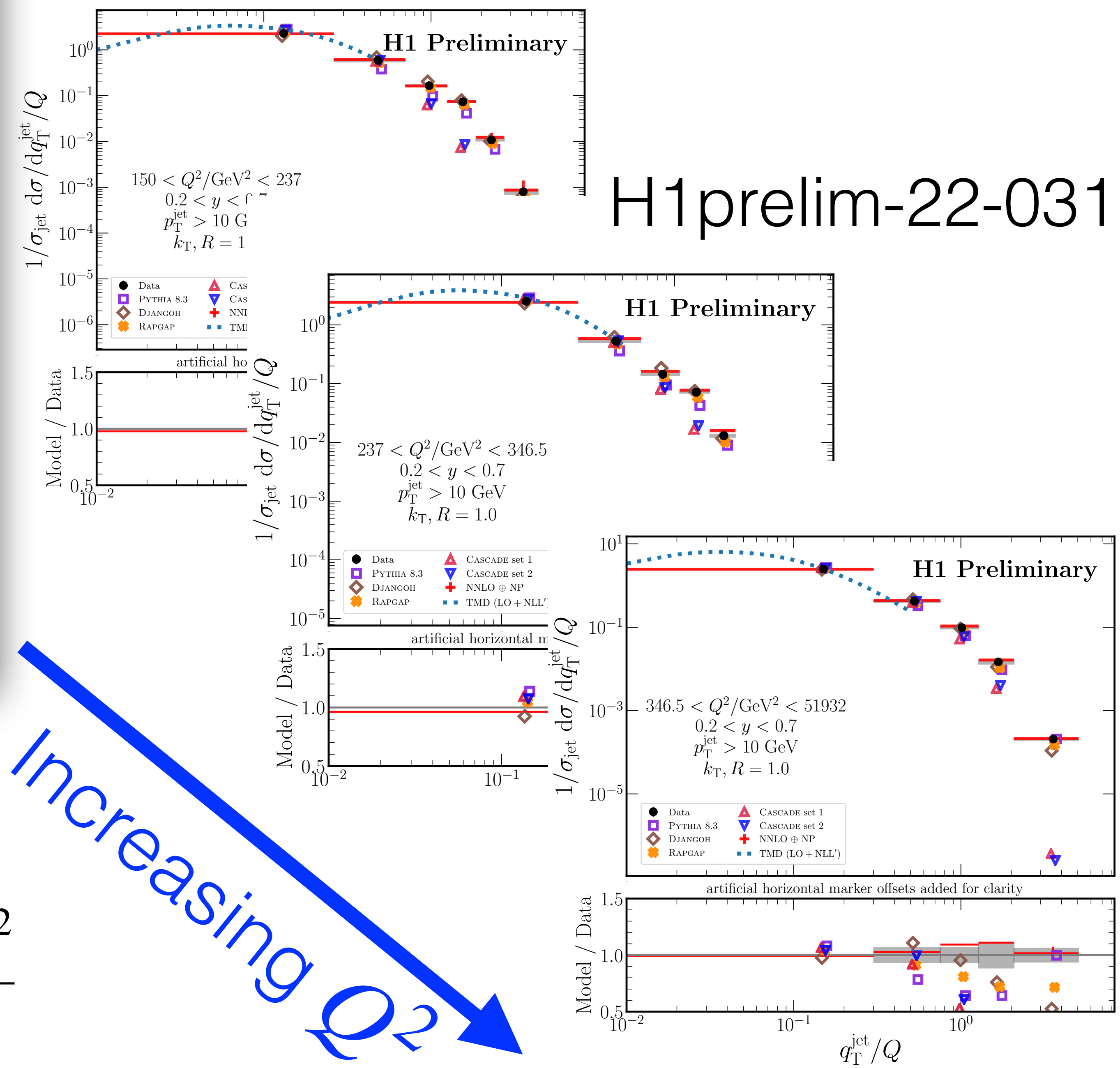
Re-using and extending





$$Q^2 = \frac{(q_T^{jet})^2}{(q_T^{jet}/Q)^2}$$

$$= \frac{(p_x^e + p_T^{jet}\cos(\phi^{jet}))^2 + (p_y^e + p_T^{jet}\sin(\phi^{jet}))^2}{(q_T^{jet}/Q)^2}$$



What is next?

The future is unbinned.
Will your predictions be able to make use of this?

There are still challenges that need to be overcome, but many experiments are starting to use these new methods in parallel.

I just gave one example, but there are also recent/ongoing results from ATLAS, CMS, LHCb, STAR, …

These data will also benefit inference tasks that can handle unbinned, multidimensional data (e.g. ML hadronization!)

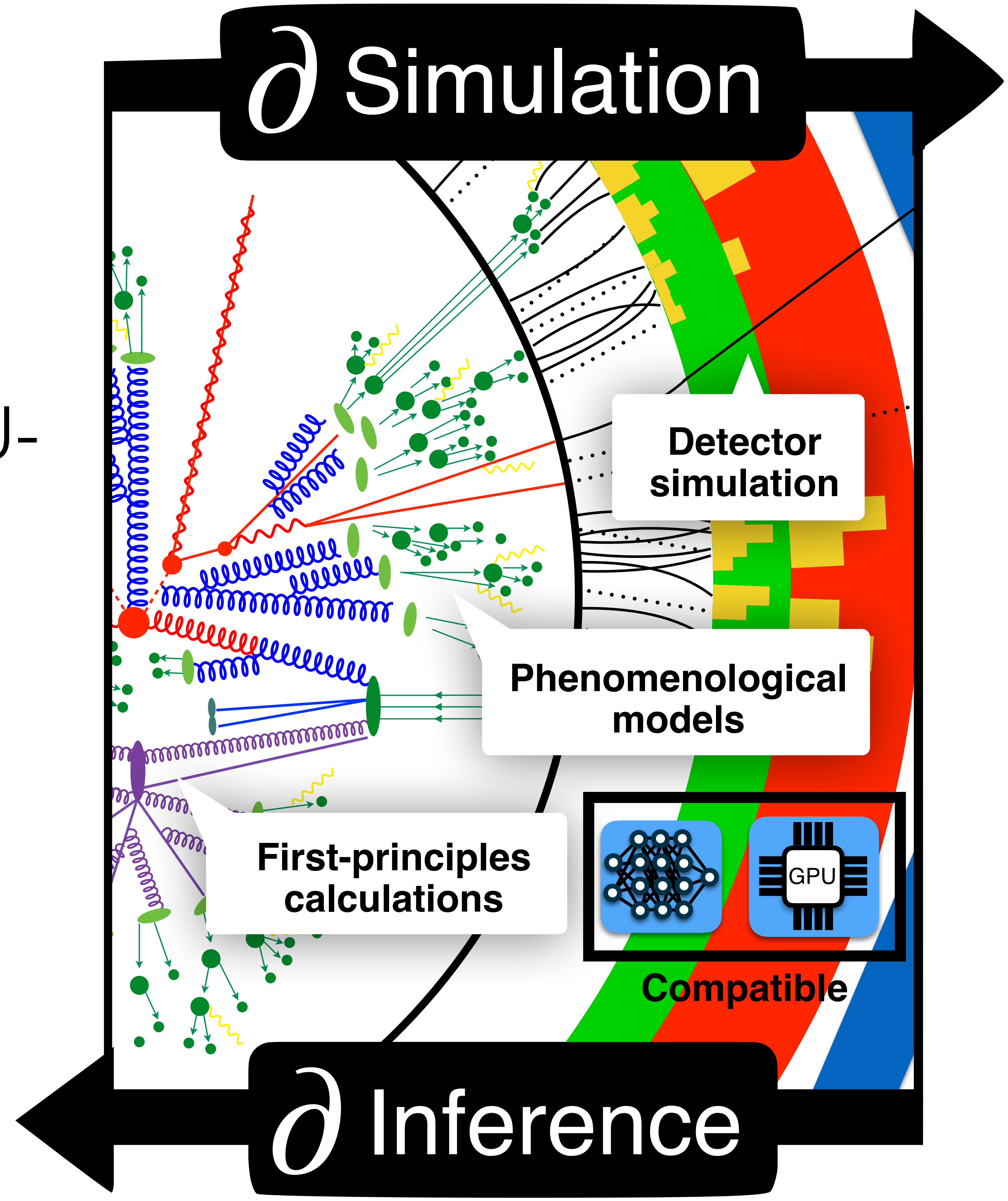
What is next-to-next?

An end-to-end differentiable event generator?

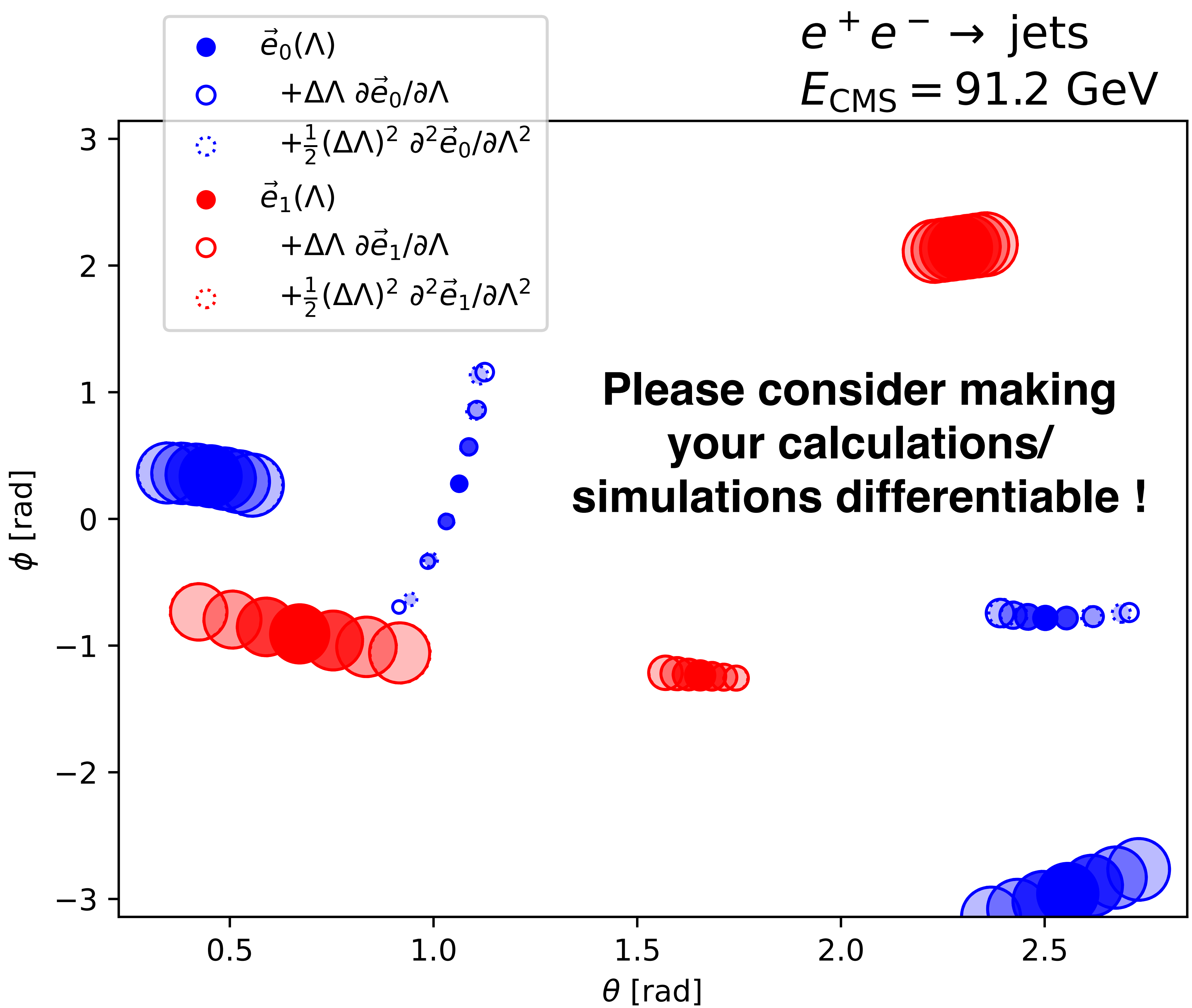
Computing benefits: fast, GPU-compatible, ML-compatible

(Imagine HADML was just one layer of a NN!)

Physics benefits: precise inference + uncertainties



The power of a differentiable generator



Please consider making your calculations/ simulations differentiable !

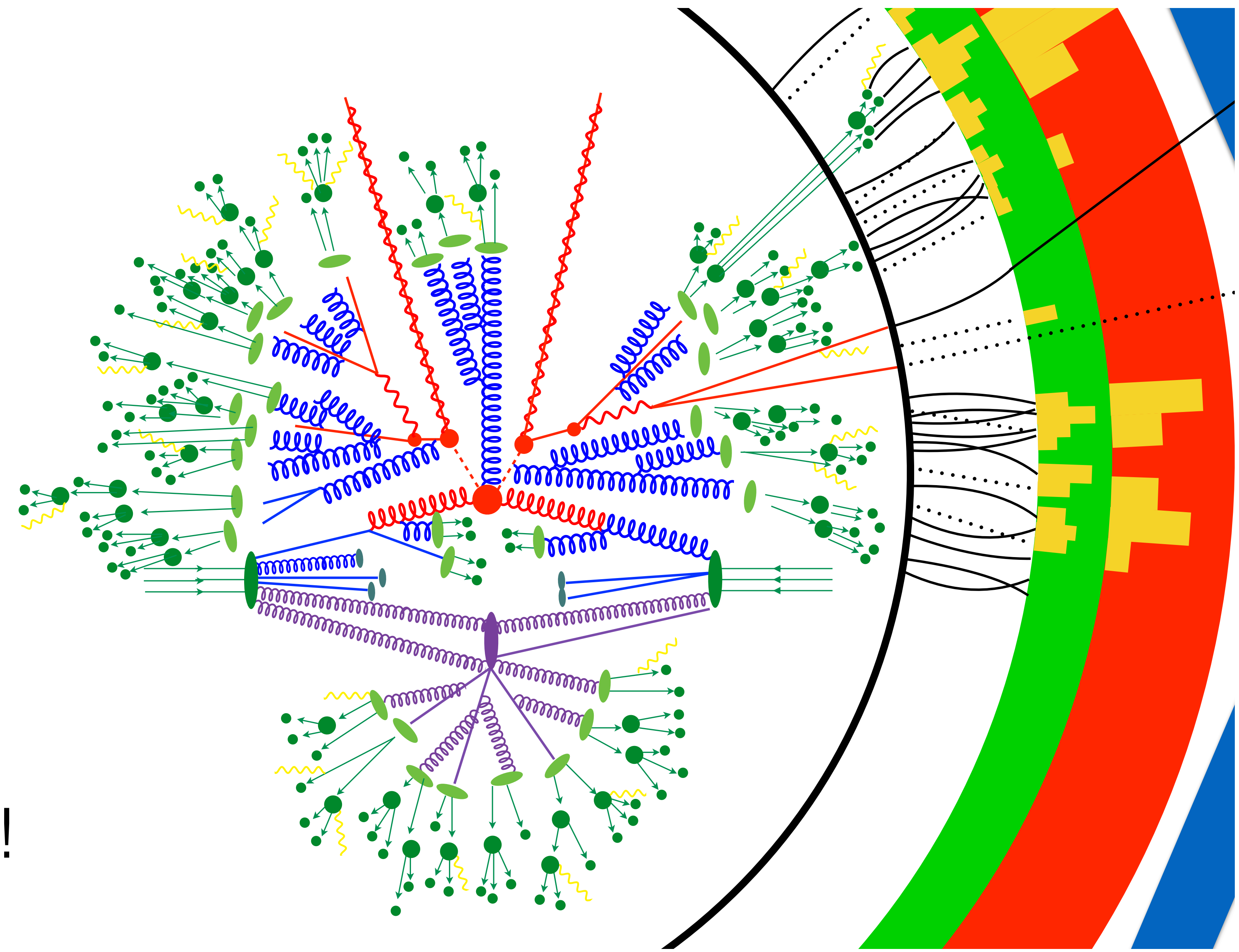
2208.02274

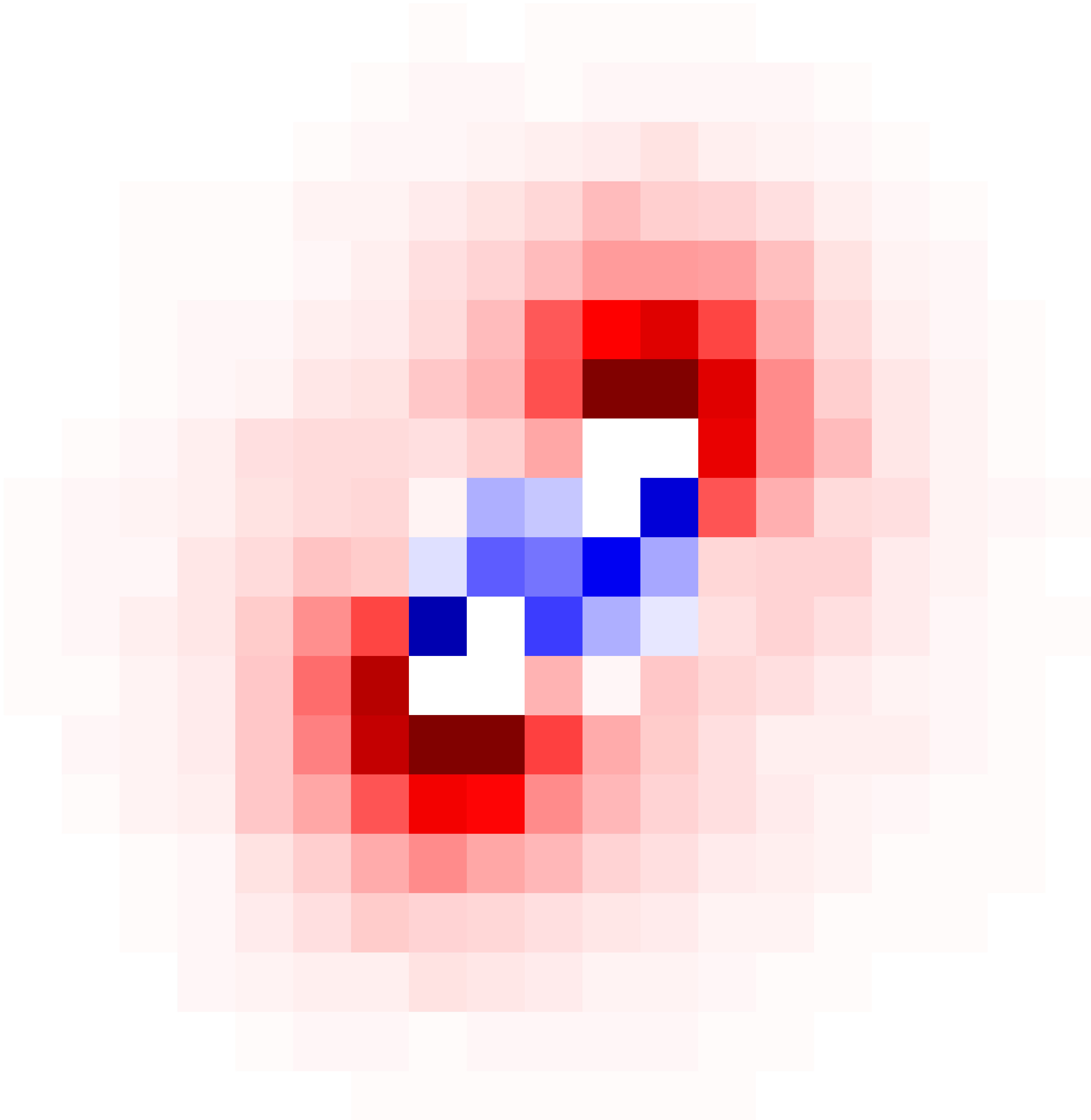
Conclusions and Outlook

ML has many exciting use-cases for precision physics at the LHC. I've only covered a couple of representative examples.

This is not just a tool for experimentalists.

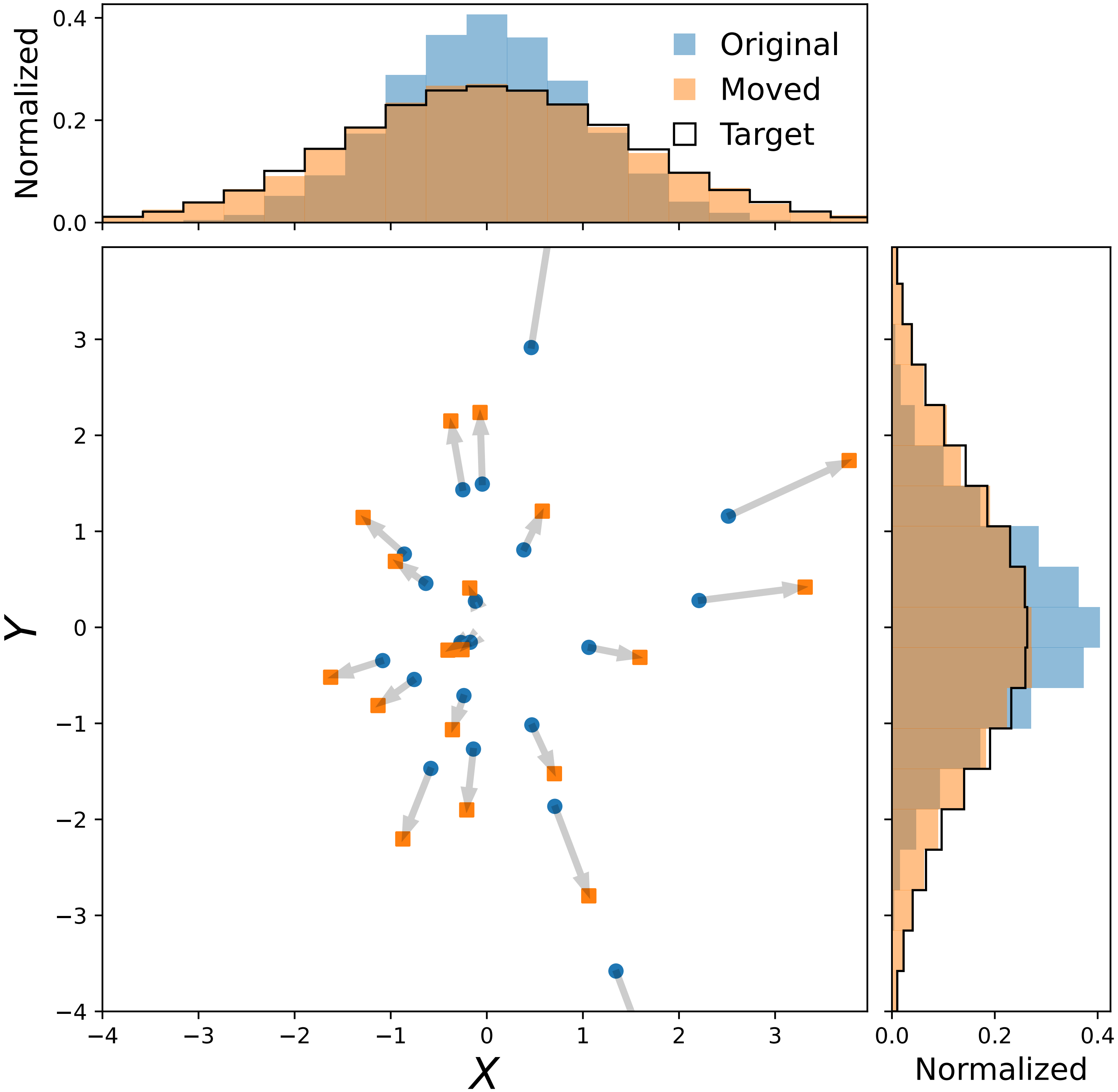
Physicists are also developing new ML methods - we need innovation in order to make the most of past, present, and future data!





Fin.

Differentiable Simulation



$$X \sim \mathcal{N}(\mu, \sigma)$$

$\downarrow$

```
x = np.random.normal(mu,sigma)
```

$\downarrow$

```
z = np.random.uniform(0,1)
x = sigma*Phiinv(z)+mu
```

(**Phiinv** = inverse Gaussian CDF)

Now, can compute $\partial/\partial\mu$ and $\partial/\partial\sigma$

We can then do:

$$\text{sim}(\mu_0 + \epsilon) \approx \text{sim}(\mu_0) + \frac{\partial \text{sim}}{\partial \mu}\epsilon$$

2208.02274