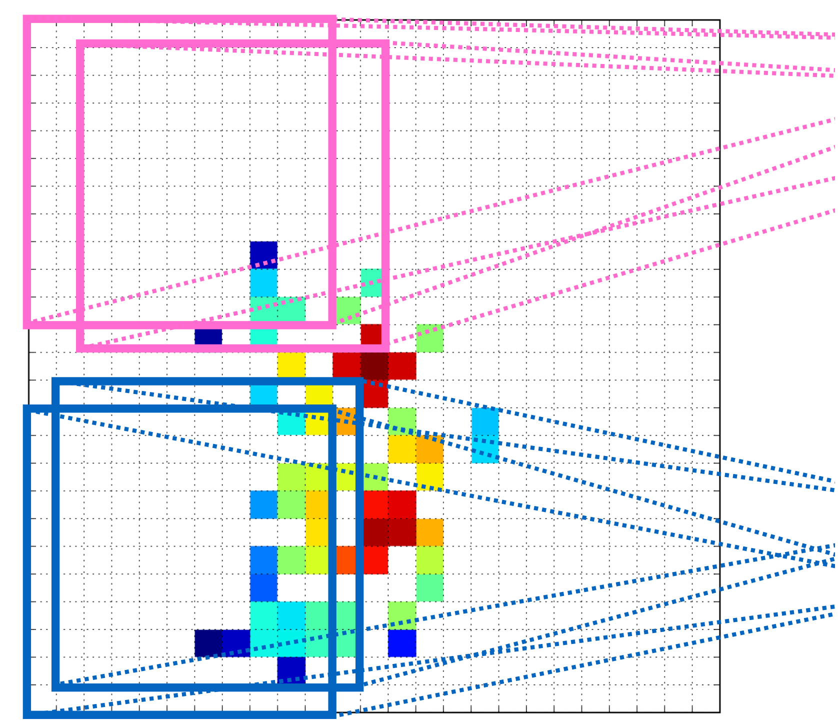
AI-driven detector design for the EIC

Benjamin Nachman

Lawrence Berkeley National Laboratory

bpnachman.com
bpnachman@lbl.gov

@bpnachman bnachman



BERKELEY LAB

BIDS
BERKELEY INSTITUTE FOR DATA SCIENCE

AIWG Special Meeting
July 20, 2022

Overview

Detector Model

parameters of interest θ

Overview

Detector Model

parameters of interest θ

Goal: find best θ given a metric(s).

Overview

Detector Model

parameters of interest θ

Goal: find best θ given a metric(s).

Challenge: detector output is high-dimensional and θ may be high-dimensional.

Overview

Detector Model

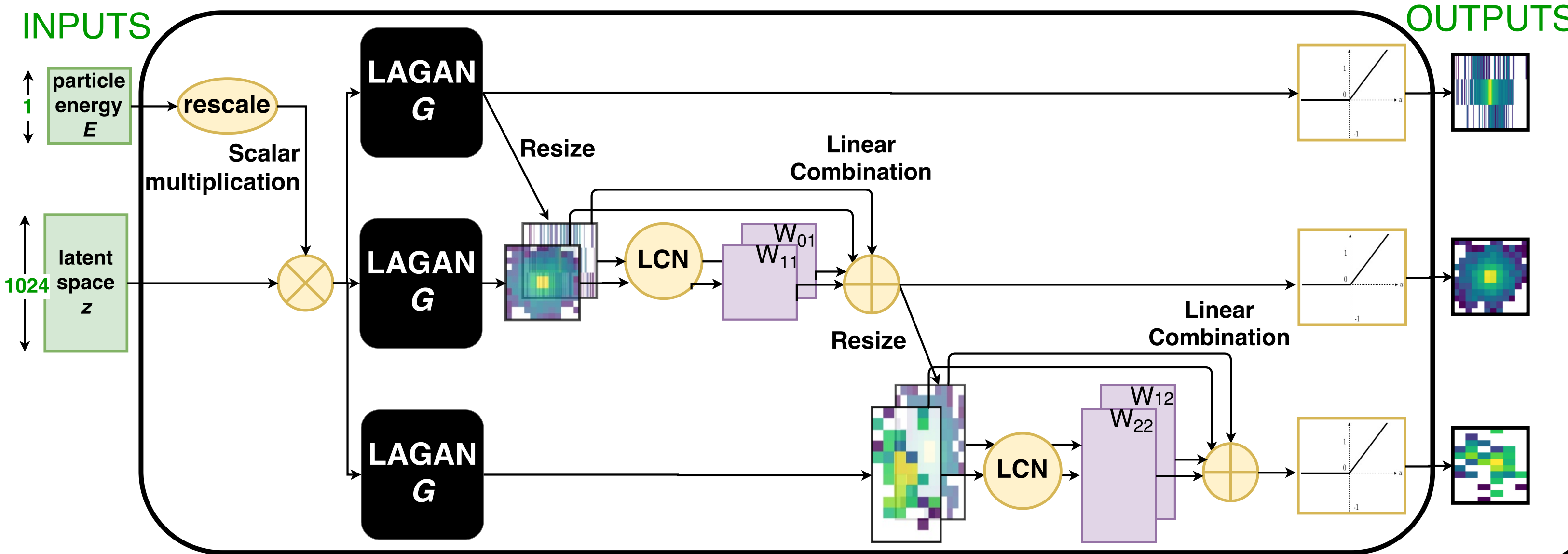
parameters of interest θ

Goal: find best θ given a metric(s).

Challenge: detector output is high-dimensional and θ may be high-dimensional.

Today: can we use ML to (1) interpolate in the high-dimensional space, (2) define optimal metrics, and (3) find the best values of θ.

Outline



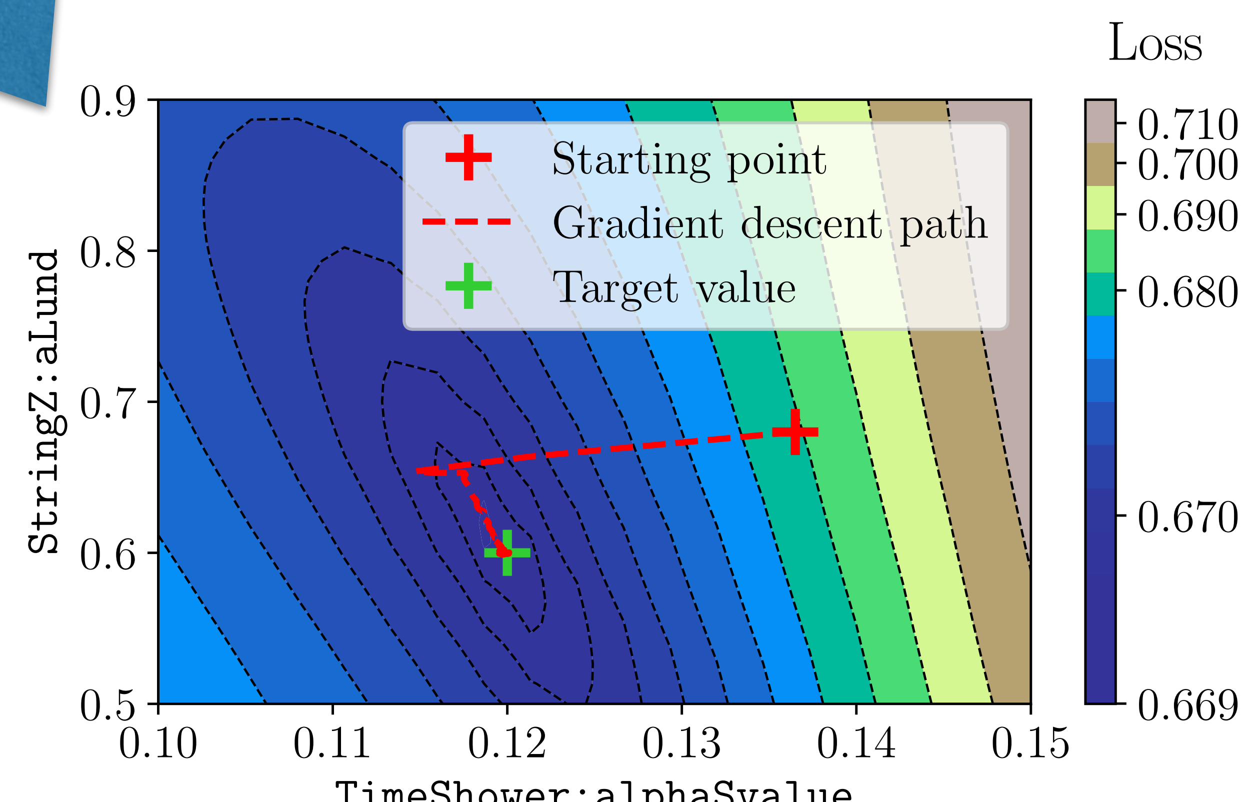
Detector Modeling

Surrogate Model
Differentiable Simulation

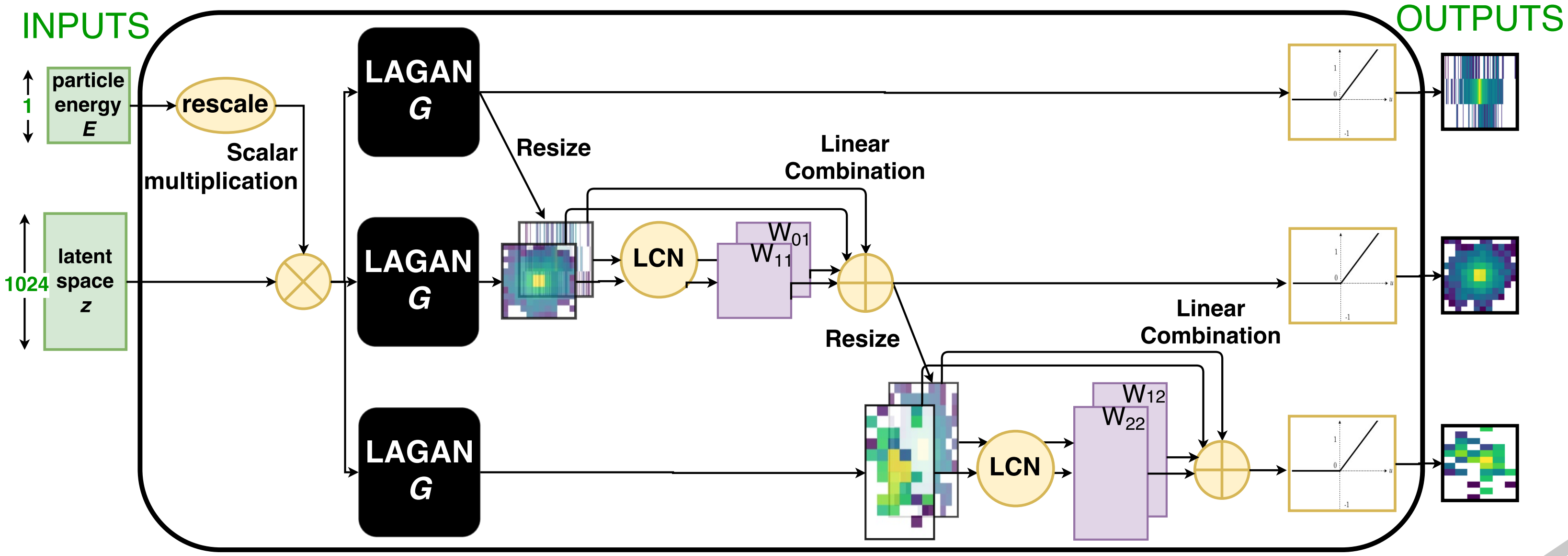
For free: GPU-enabled fast sim.

ML-based Optimization

Gradient-based
Gradient-free



Outline



Detector Modeling

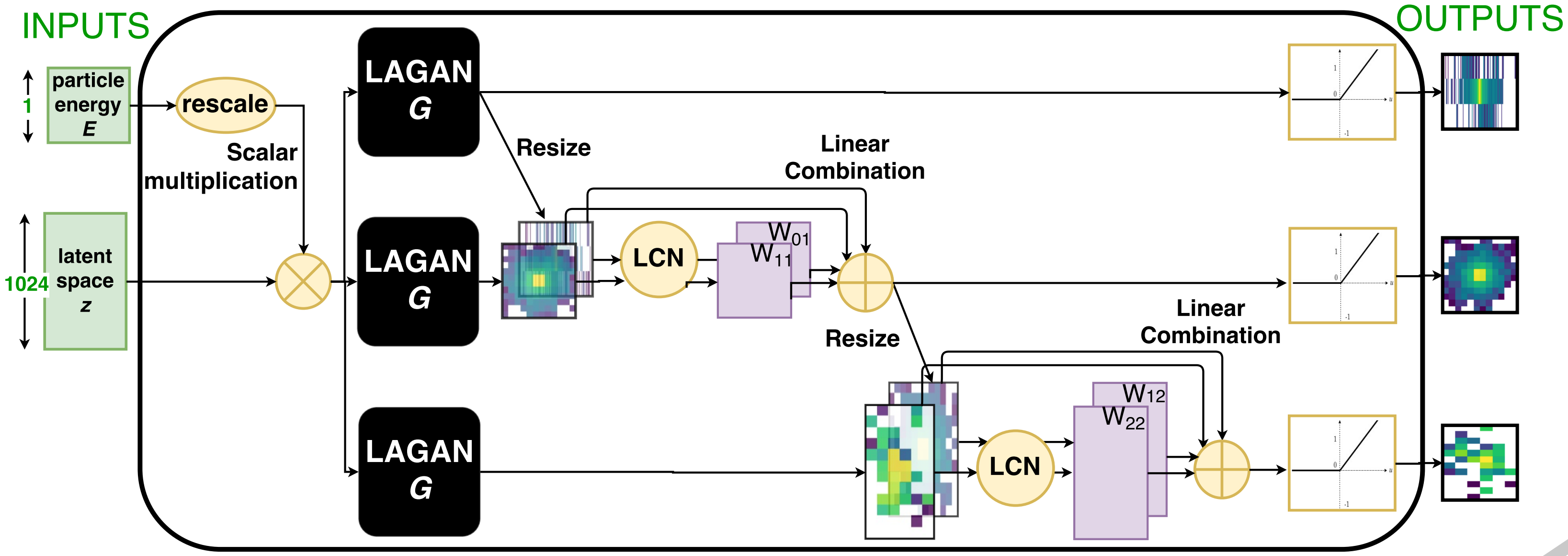
Surrogate Model
Differentiable Simulation

For free: GPU-enabled fast sim.

Gradient-based
Gradient-free

ML-based Optimization

Outline



ML-based Optimization

Detector Modeling

Surrogate Model

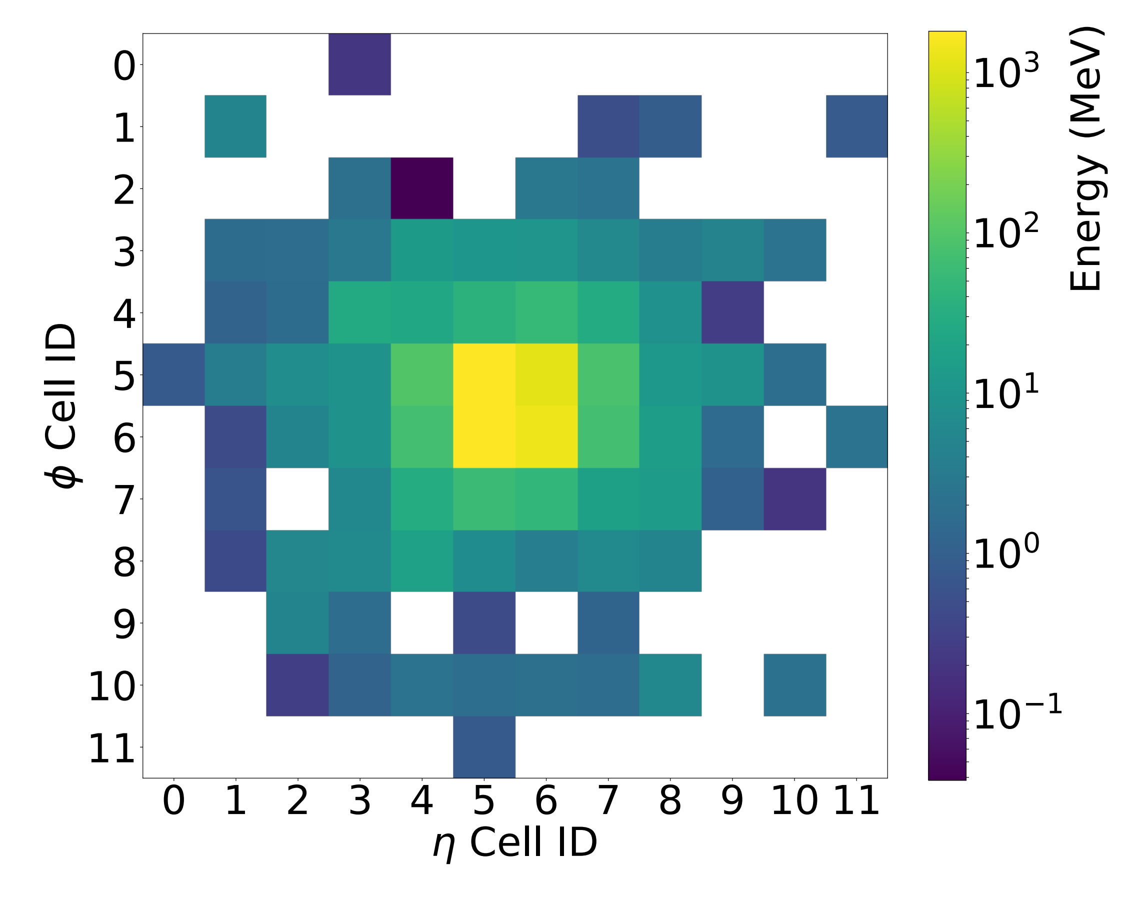
Differentiable Simulation

For free: GPU-enabled fast sim.

Gradient-based

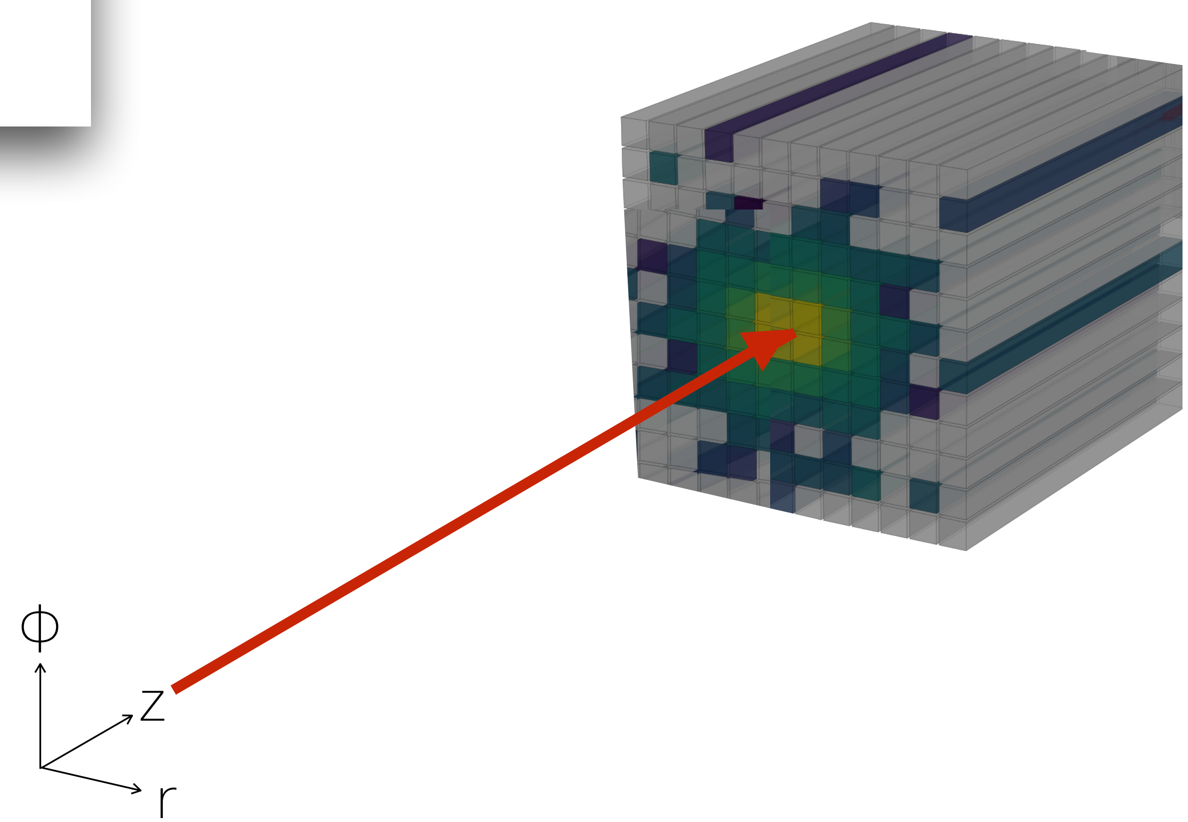
Gradient-free

Surrogate Models with ML



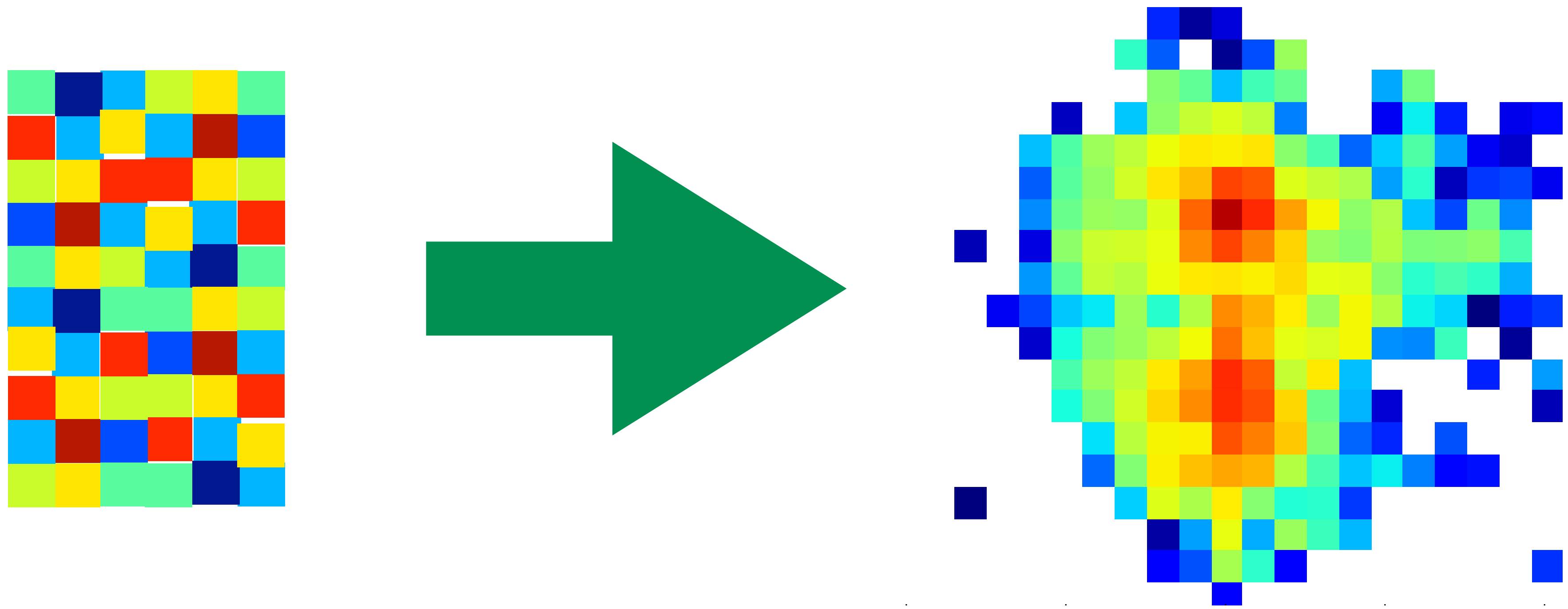
Can we train a neural network to emulate the detector simulation?

Grayscale images: Pixel intensity = energy deposited



Introduction: generative models

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



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

Tools

GANs

Generative Adversarial Networks

Score-based

NFs

Normalizing Flows

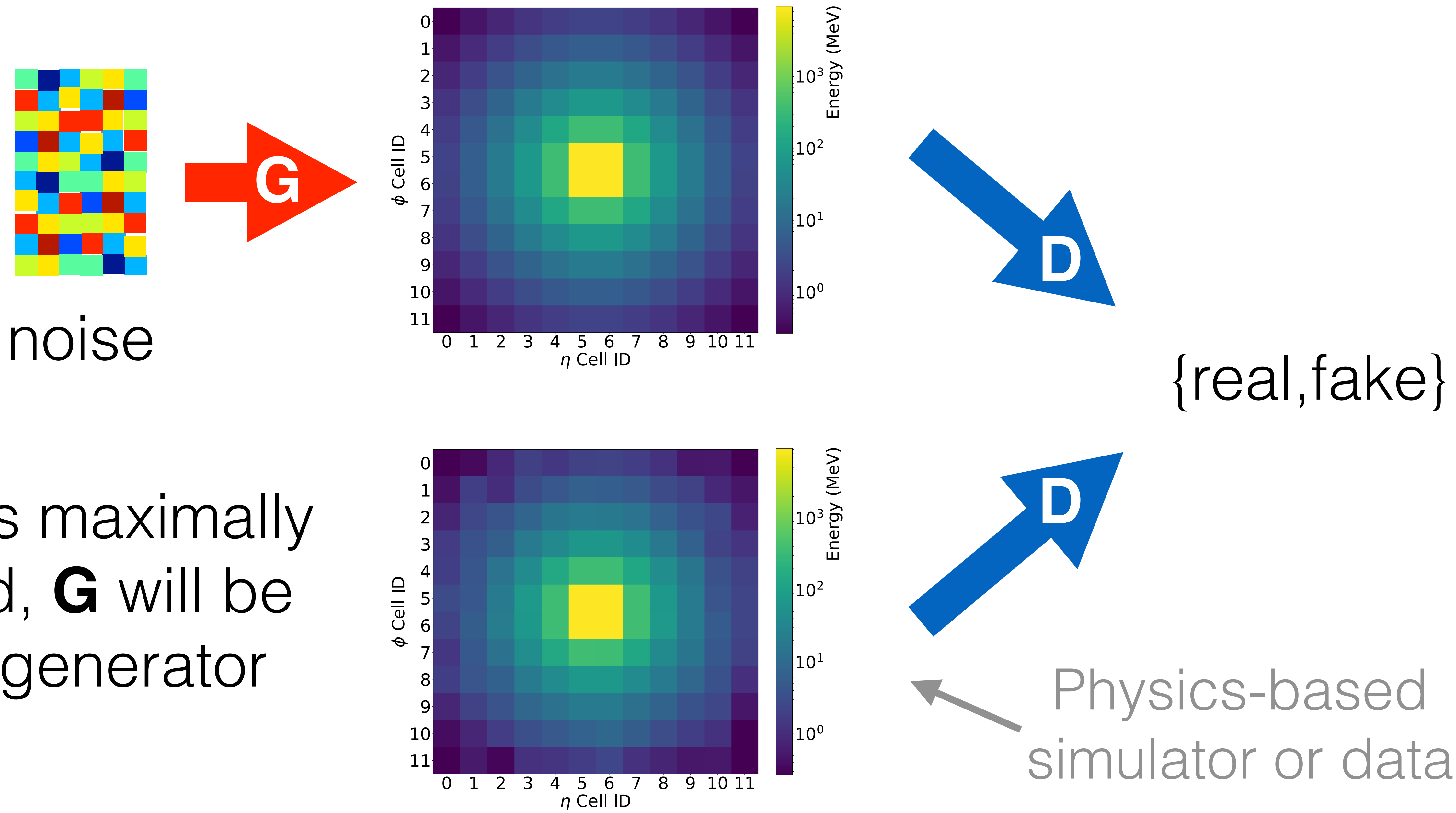
VAEs

Variational Autoencoders

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

Introduction: GANs

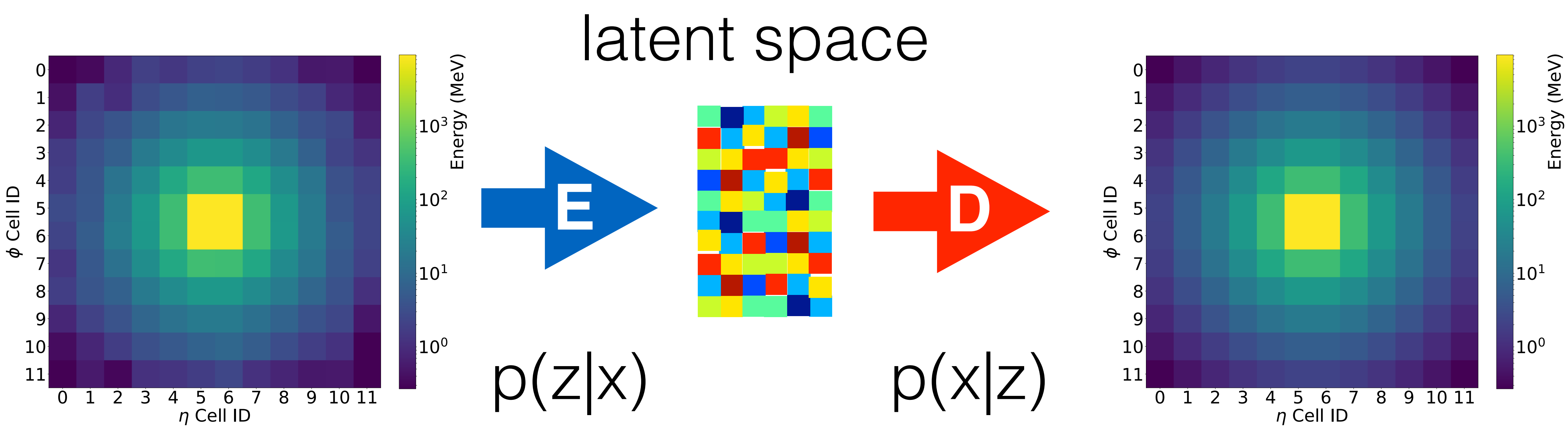
Generative Adversarial Networks (GANs):
A two-network game where one ***maps noise to structure*** *and one* ***classifies images as fake or real****.*



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

Introduction: VAEs

Variational Autoencoders (VAEs):
A pair of networks that embed the data into a latent space with a given prior and decode back to the data space.



Physics-based simulator or data

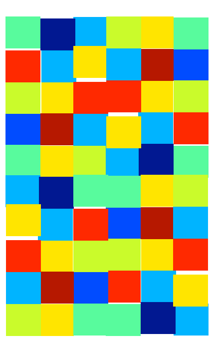
Probabilistic ***encoder***

Probabilistic ***decoder***

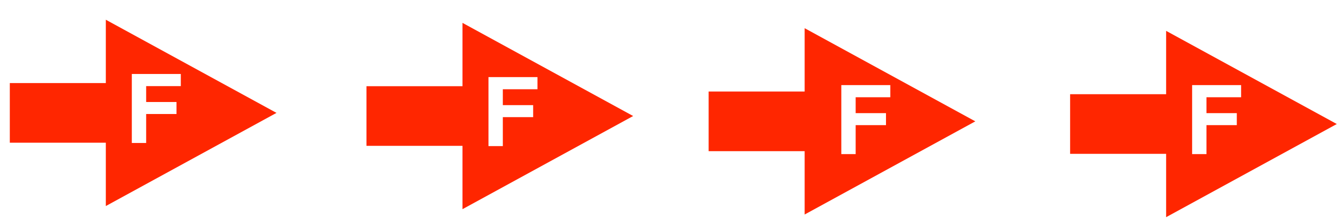
Introduction: NFs

Normalizing Flows (NFs):
A series of invertible transformations mapping a known density into the data density.

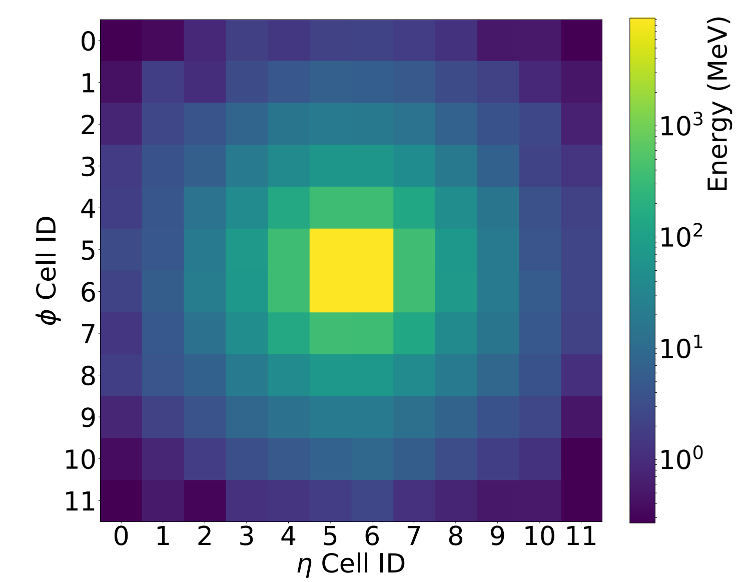
Optimize via maximum likelihood



latent space



Invertible transformations with tractable Jacobians



p(z)

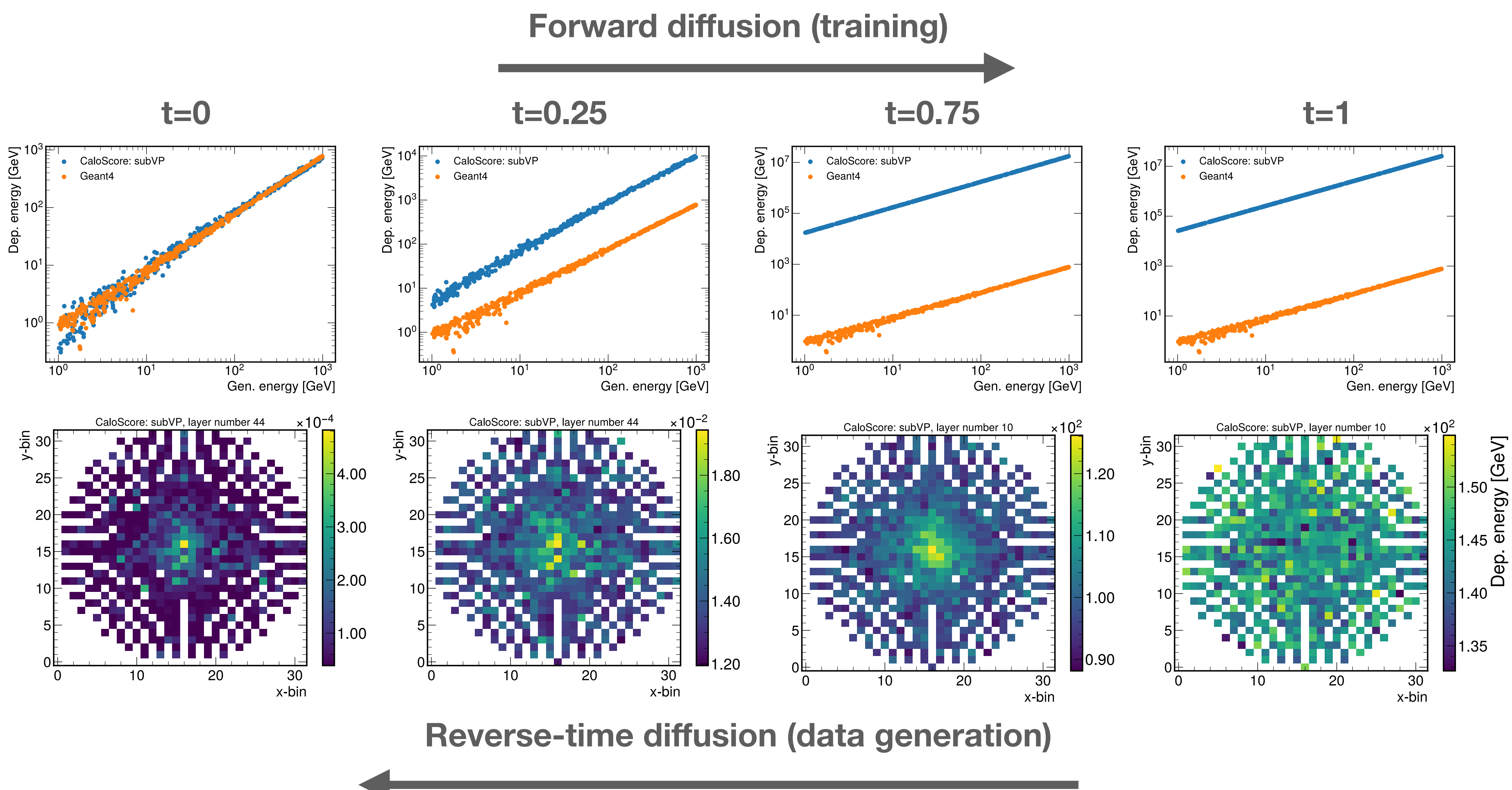
$p(x) = p(z)\, |dF^{-1}/dx|$

p(x)

Introduction: Score-based

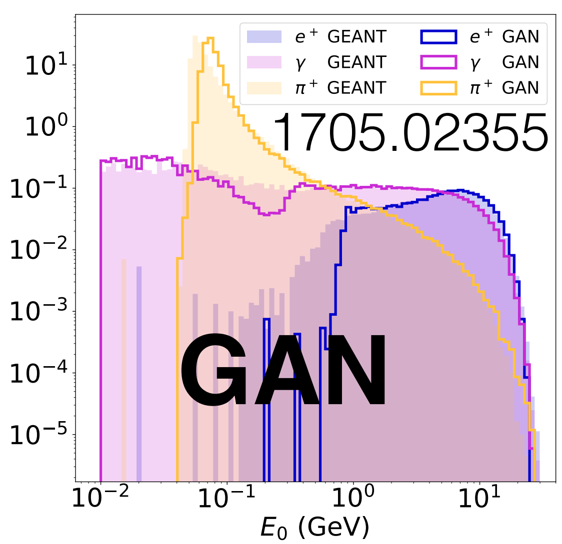
Score-based

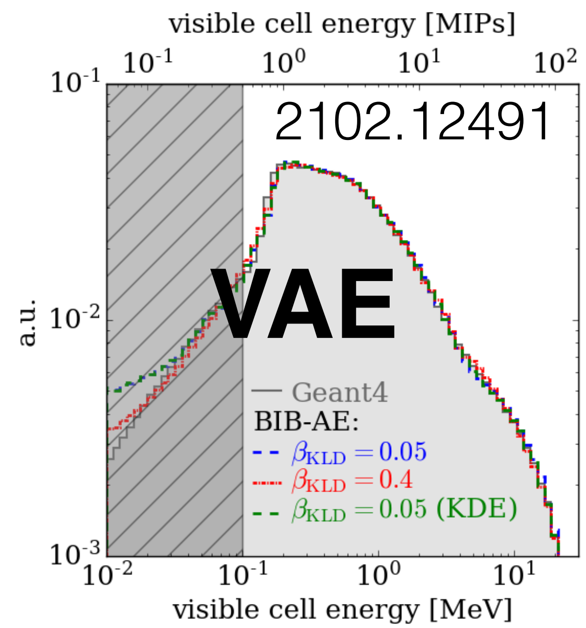
Learn the gradient of the density instead of the probability density itself.

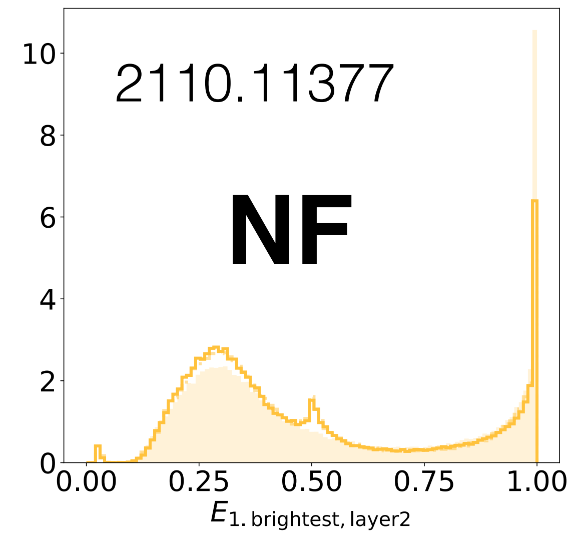


From 2206.11898

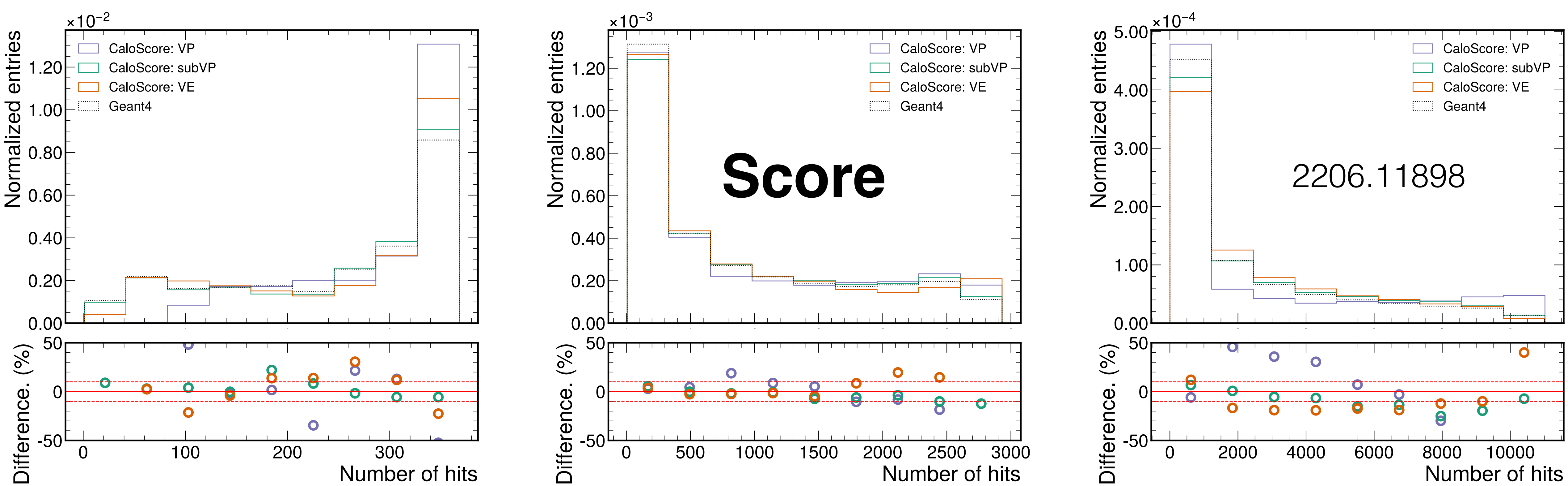
Calorimeter ML Surrogate Models





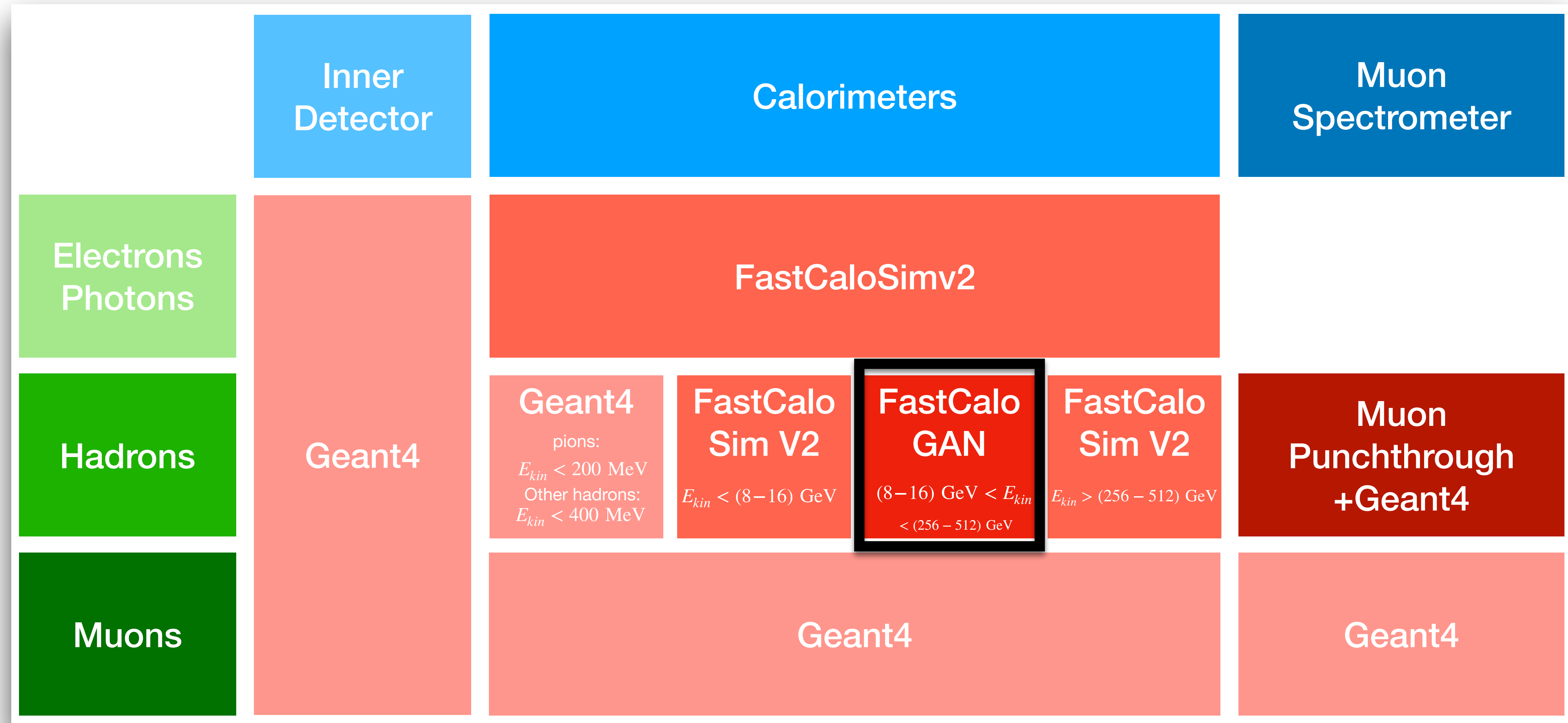


Many papers on this subject - see the living review for all



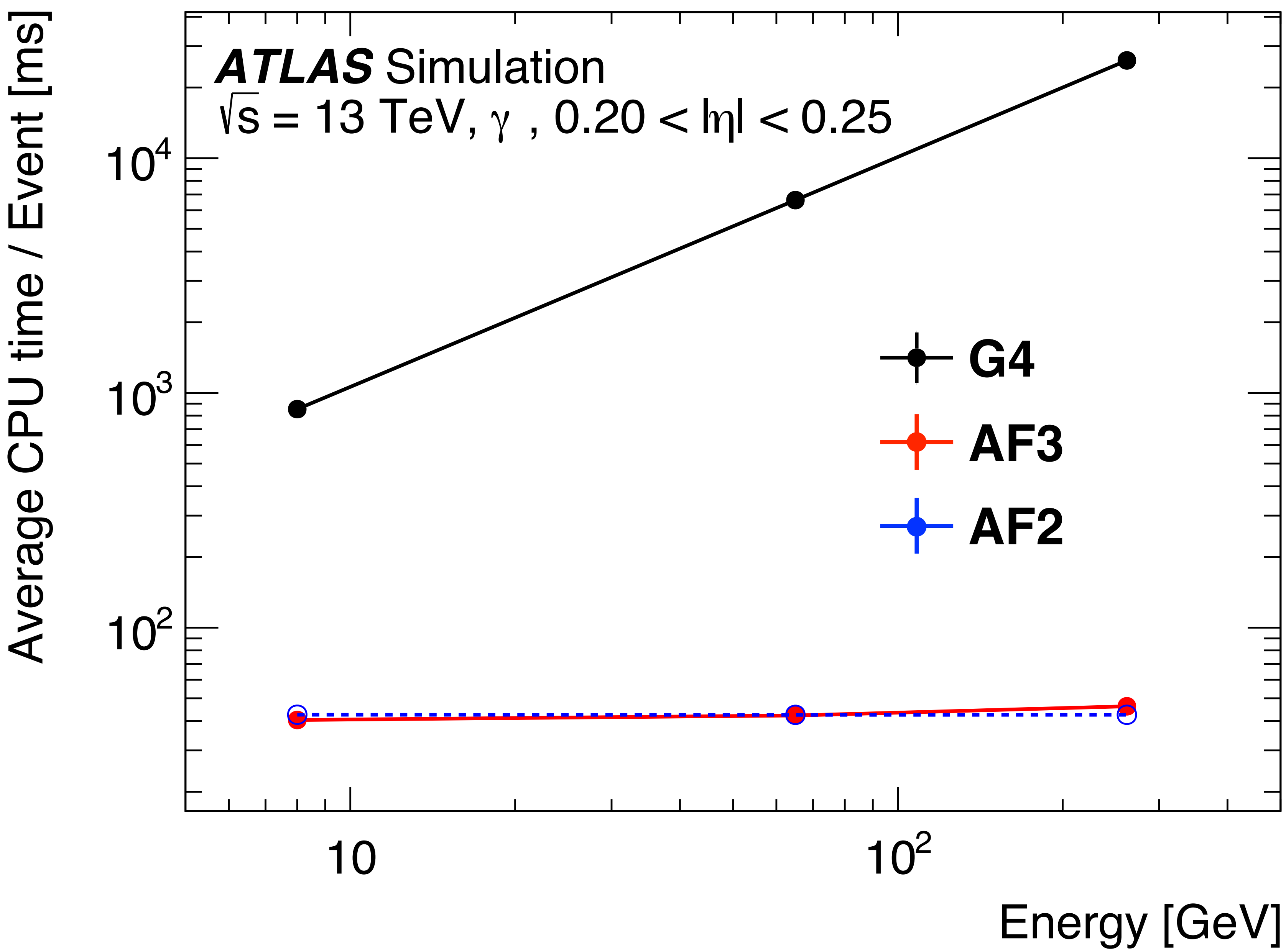
See also https://calochallenge.github.io/homepage/

Integration into real detector sim.



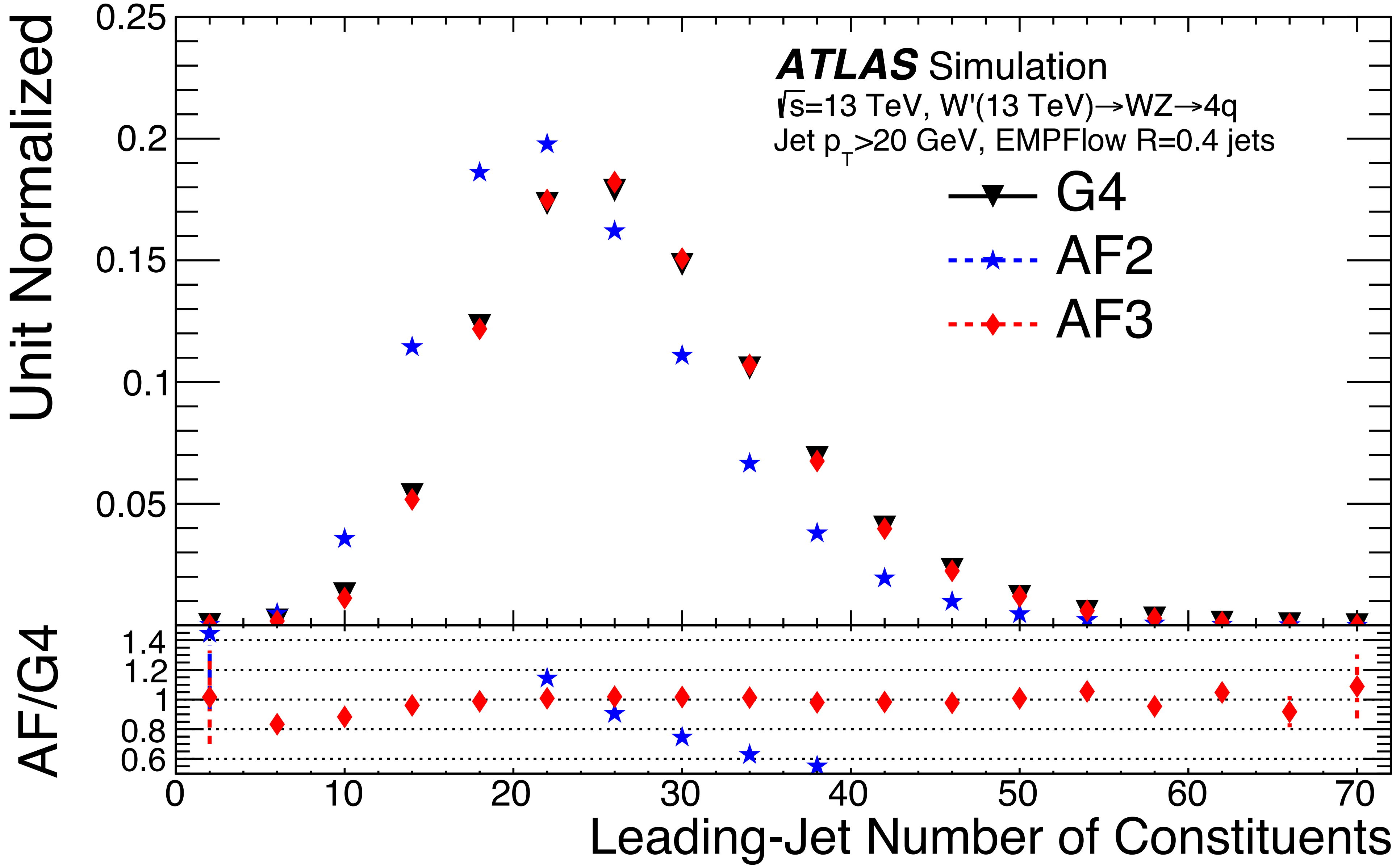
The ATLAS Collaboration fast simulation (AF3) now includes a GAN at intermediate energies for pions

Integration into real detector sim.



As expected, the fast sim. timing is independent of energy, while Geant4 requires more time for higher energy.

Integration into real detector sim.

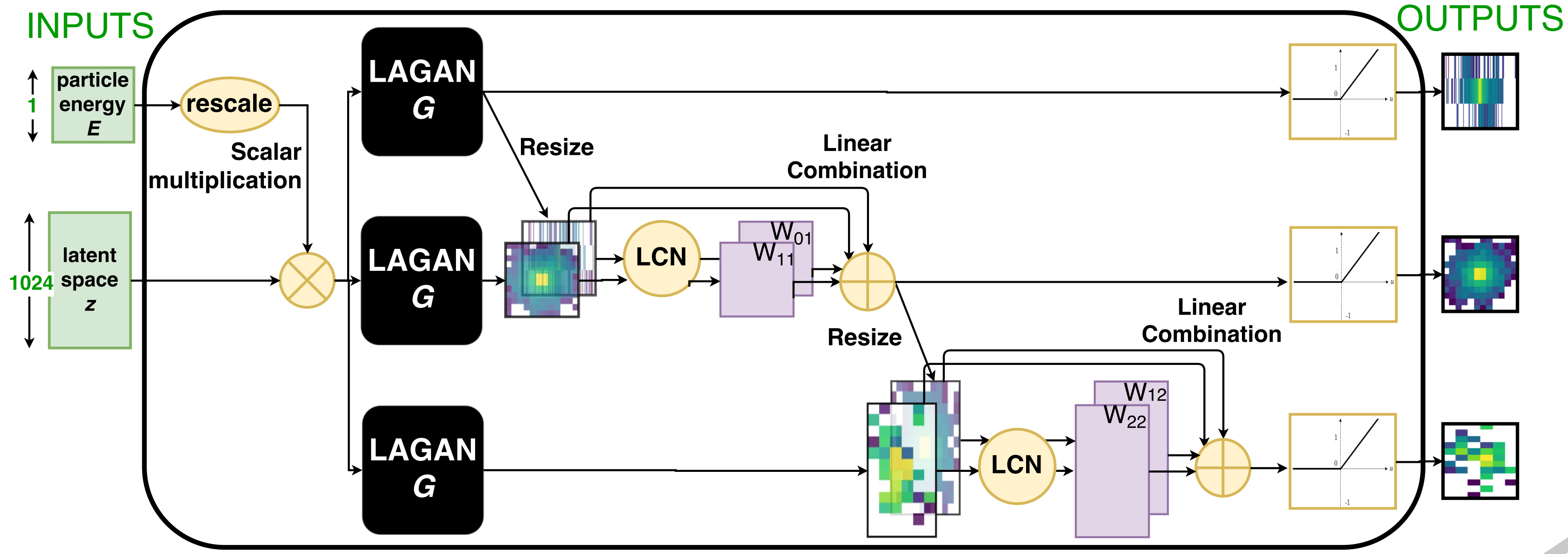


The new fast simulation (**AF3**) significantly improves jet substructure with respect to the older one (**AF2**).

Outline

INPUTS

OUTPUTS



Detector Modeling

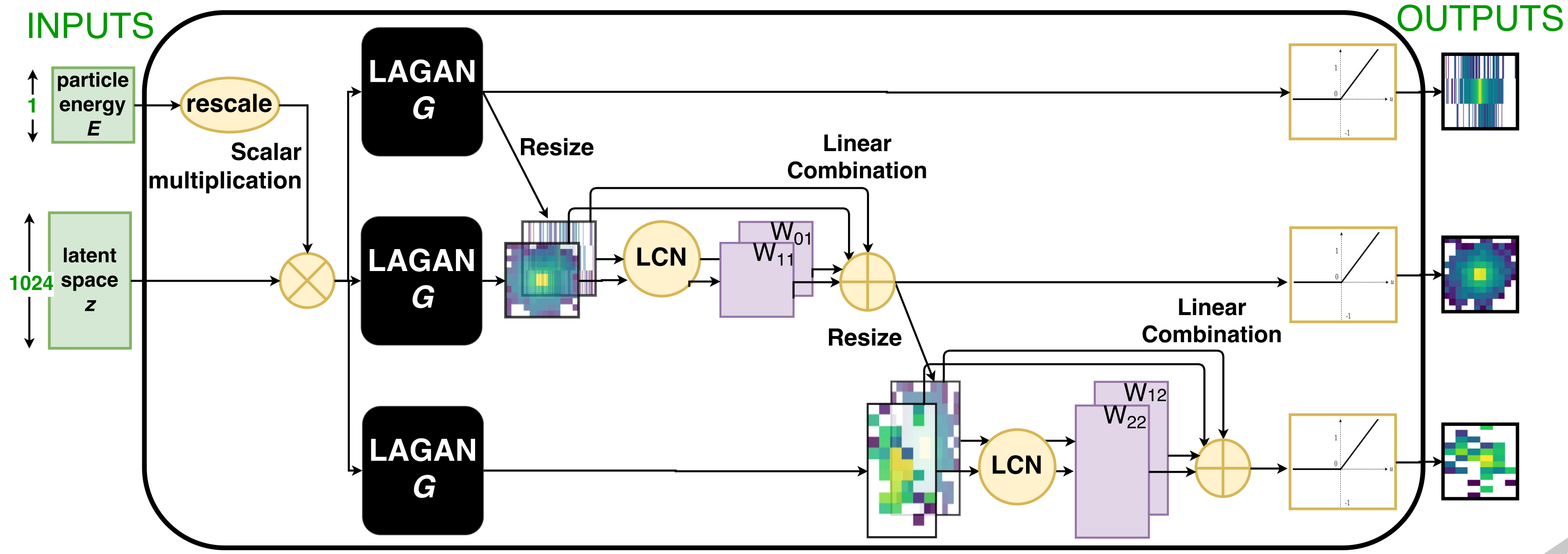
Surrogate Model
Differentiable Simulation

For free: GPU-enabled fast sim.

Gradient-based
Gradient-free

ML-based Optimization

Outline



Detector Modeling

Surrogate Model

Differentiable Simulation

For free: GPU-enabled fast sim.

Gradient-based
Gradient-free

ML-based Optimization

Differentiable Simulation

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

Differentiable Simulation

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

$$\downarrow$$

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

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)

Removed randomness from simulator

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)

Removed randomness from simulator

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

Differentiable Simulation

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

$$\downarrow$$

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

$$\downarrow$$

Removed randomness from simulator

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

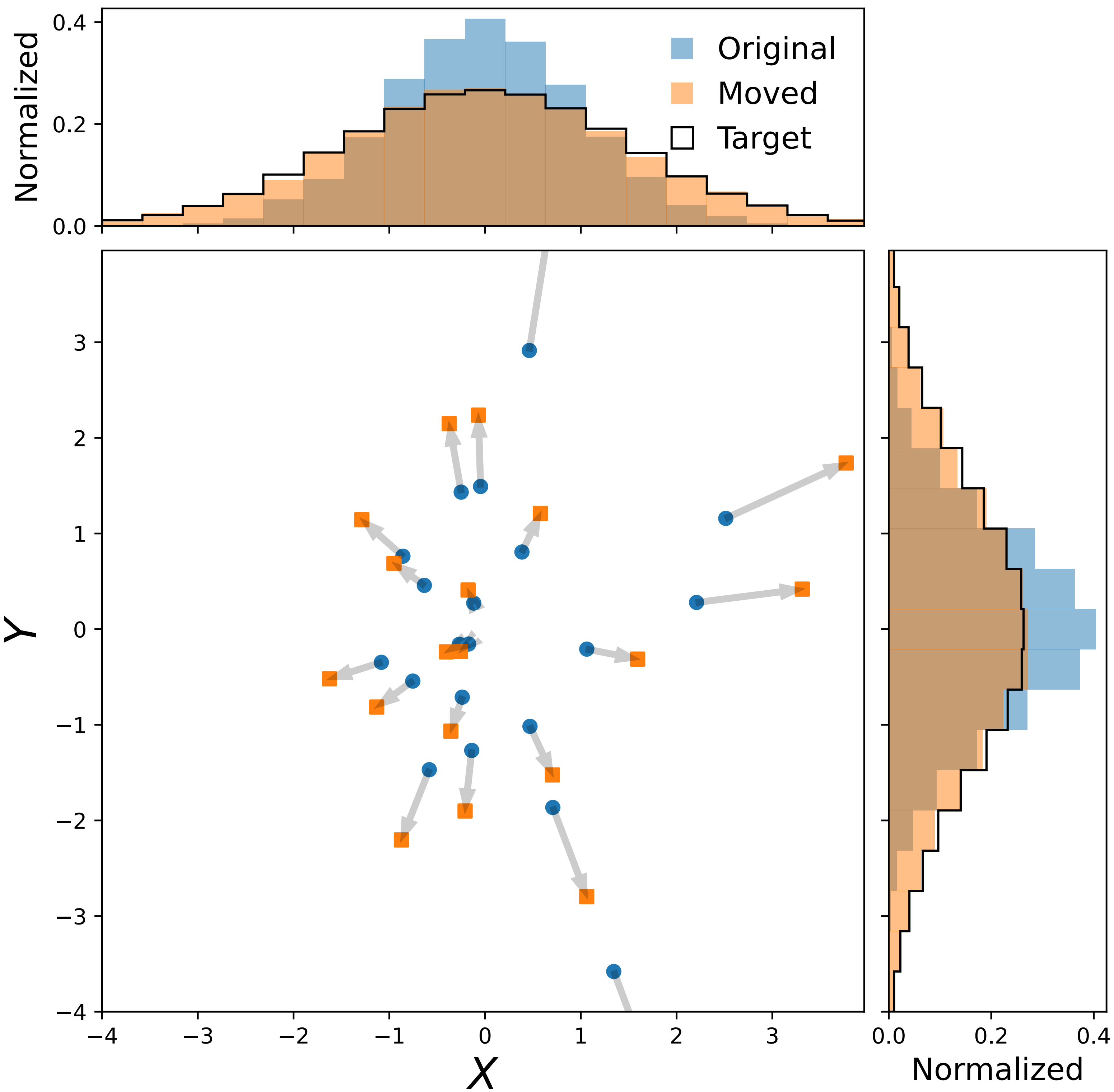
(**Phiinv** = inverse Gaussian CDF)

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

We can then do:

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

Differentiable Simulation



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

↓

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

↓

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

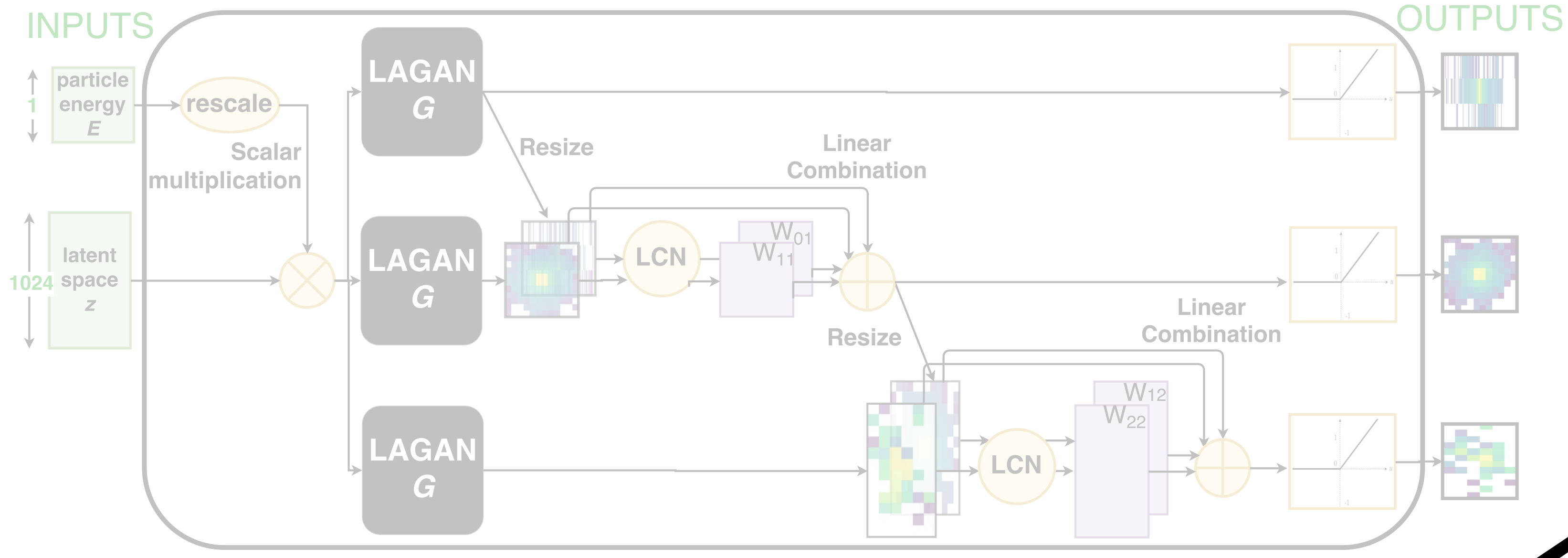
(**`Phiinv`** = inverse Gaussian CDF)

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

We can then do:

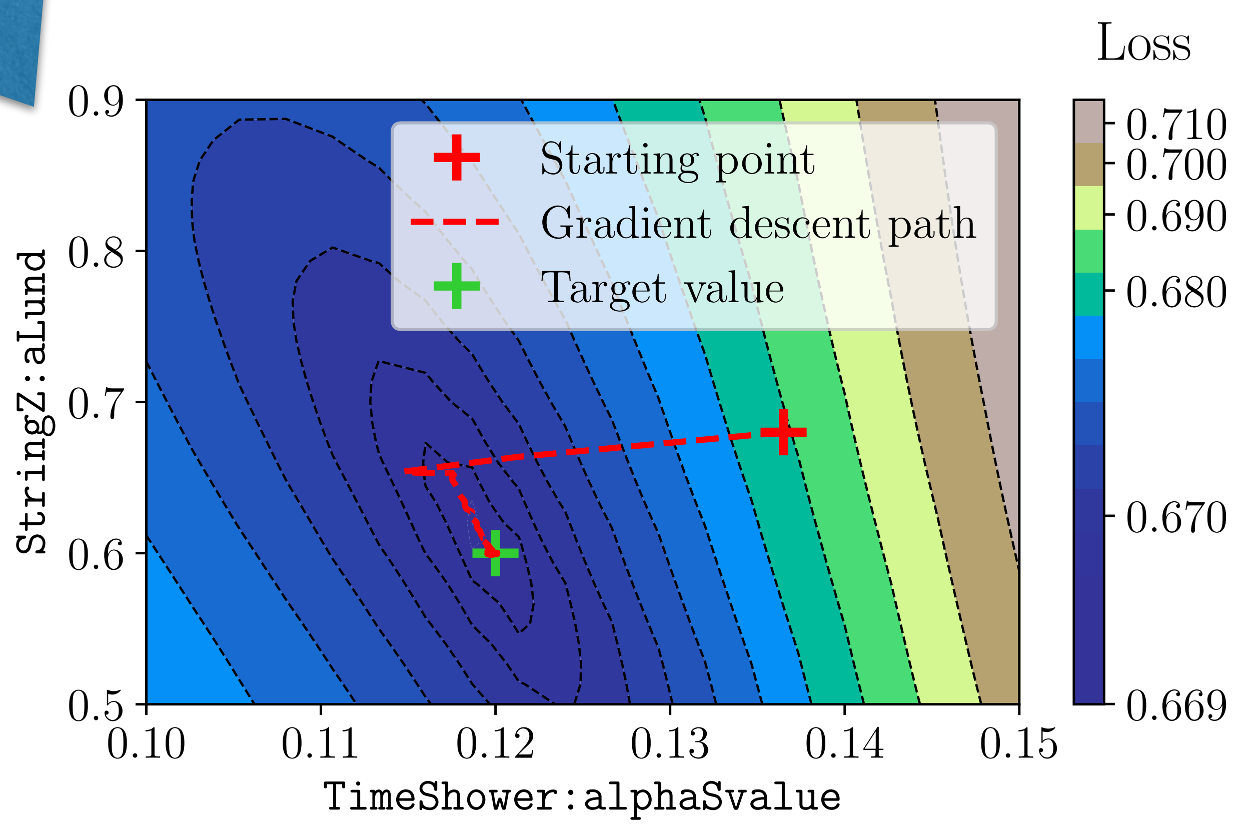
$$\text{sim}(\mu_0 + \epsilon) \approx \text{sim}(\mu_0) + \frac{\partial \text{sim}}{\partial \mu}\epsilon$$

Outline



Detector Modeling

ML-based Optimization



Example

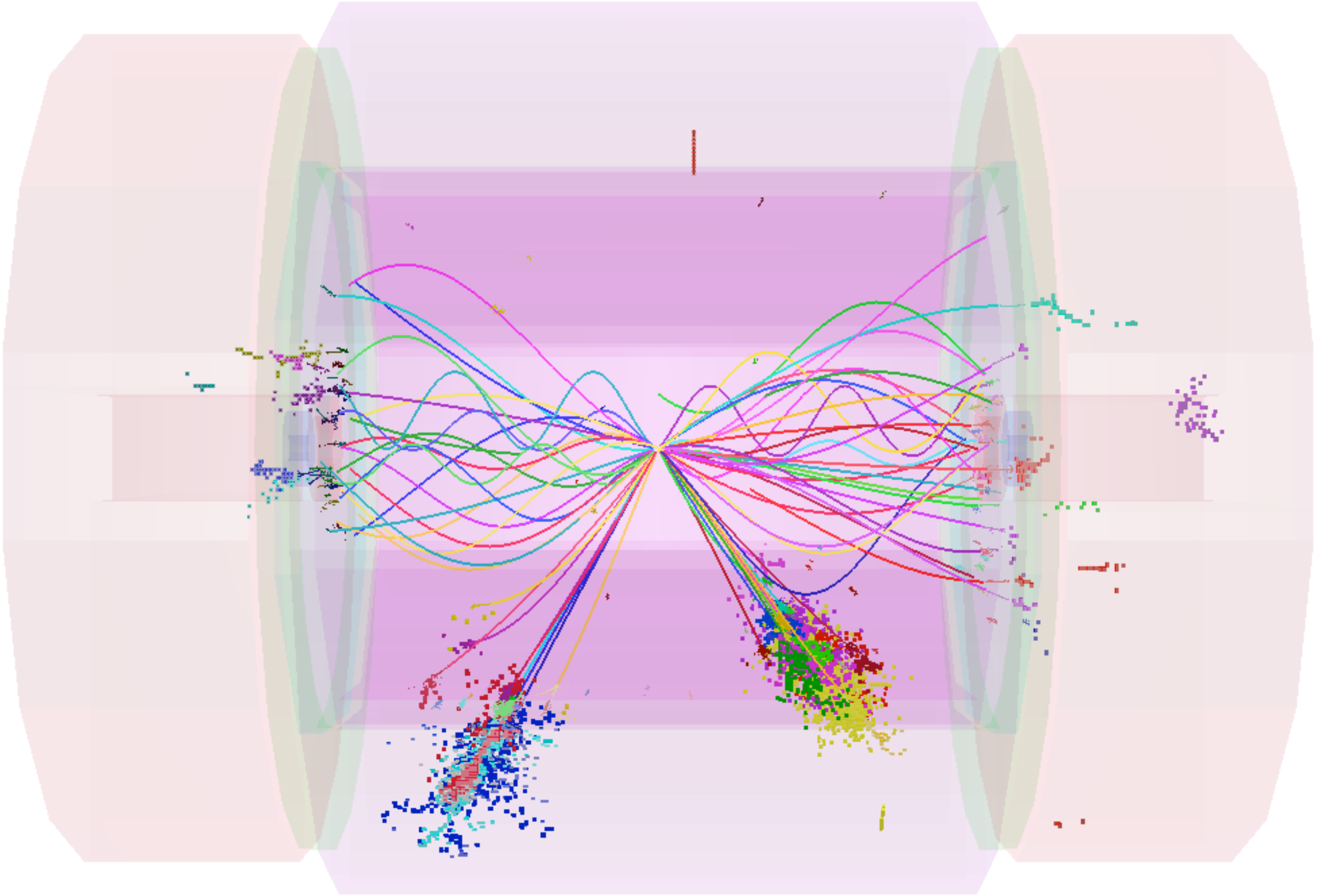
Here, instead of emulating $p(x\,|\,\theta)$ directly, we learn $\dfrac{p(x\,|\,\theta)}{p(x\,|\,\theta_0)}$

(turns the problem of generation into classification)

Example

Here, instead of emulating $p(x\,|\,\theta)$ directly, we learn $\dfrac{p(x\,|\,\theta)}{p(x\,|\,\theta_0)}$

(turns the problem of generation into classification)



Benefit: easy to integrate complex data structure (symmetries, etc.)

Downside: large weights when θ is far from θ_0

Image: Linear Collider Detector Project

Example

Step 1: Differentiable Surrogate Model

$$f(x,\theta) = \underset{f'}{\operatorname{argmax}} \sum_{i\in\boldsymbol{\theta_0}} \log f'(x_i,\theta) + \sum_{i\in\boldsymbol{\theta}} \log(1 - f'(x_i,\theta))$$

See also 1805.00020

Example

Step 1: Differentiable Surrogate Model

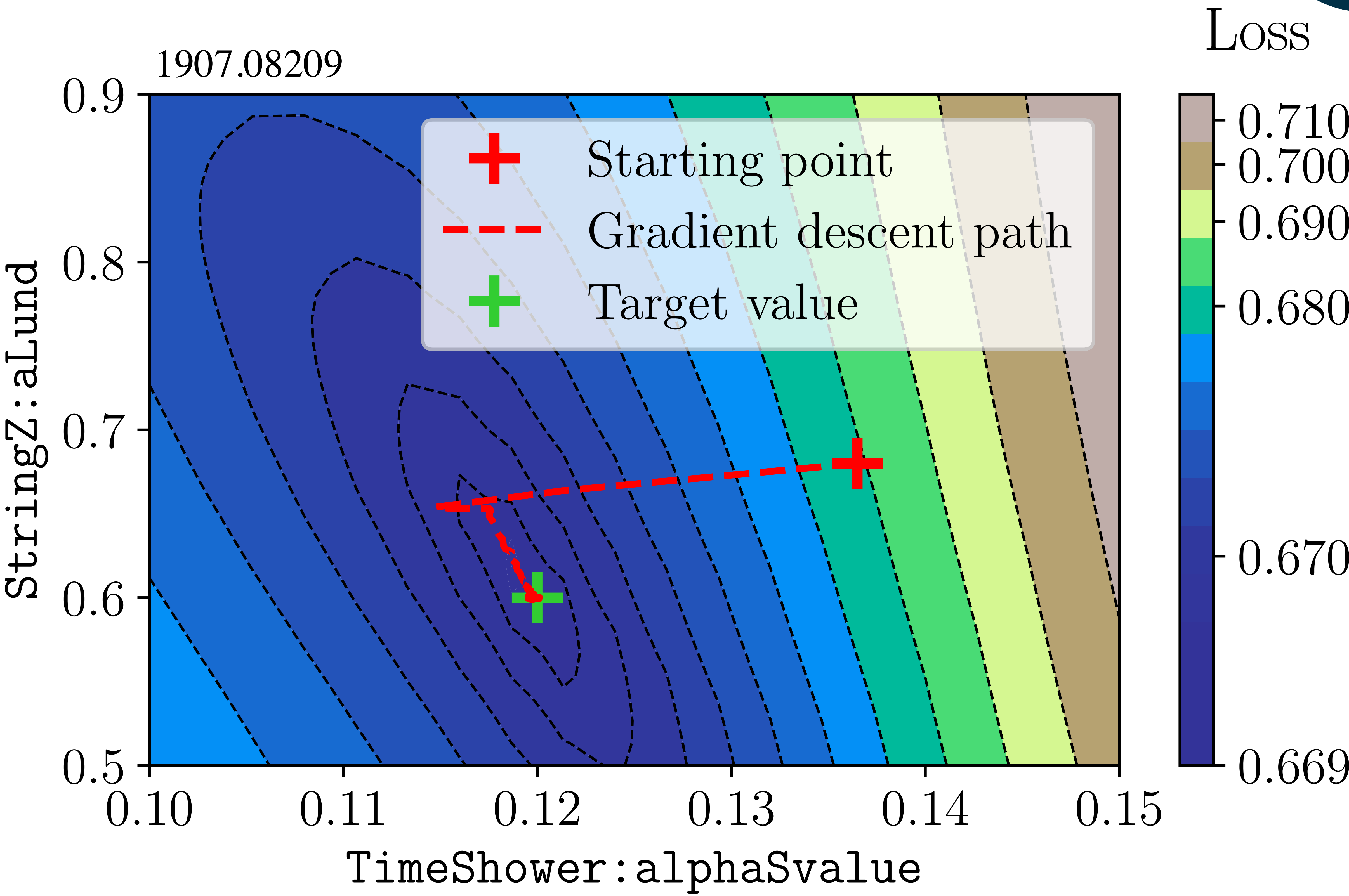
$$f(x,\theta) = \underset{f'}{\operatorname{argmax}} \sum_{i \in \boldsymbol{\theta_0}} \log f'(x_i,\theta) + \sum_{i \in \boldsymbol{\theta}} \log(1 - f'(x_i,\theta))$$

Step 2: Gradient-based optimization

$$\theta^* = \underset{\theta'}{\operatorname{argmax}} \sum_{i \in \boldsymbol{\theta_0}} \log f(x_i,\theta') + \sum_{i \in \boldsymbol{\theta_1}} \log(1 - f(x_i,\theta'))$$

See also 1805.00020

Example

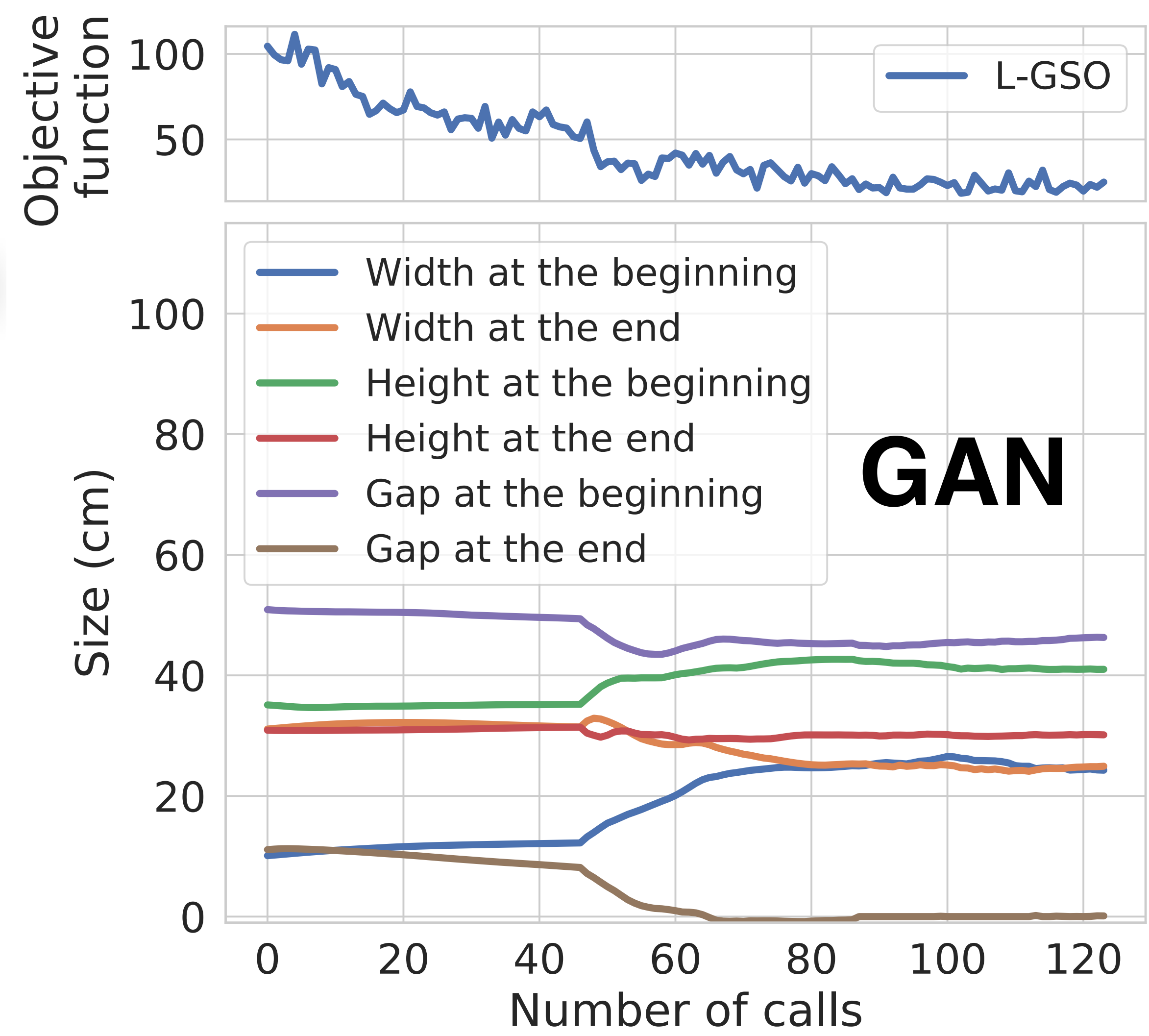


1907.08209
Loss
Starting point
Gradient descent path
Target value
0.9
0.8
0.7
0.6
0.5
StringZ:aLund
0.10
0.11
0.12
0.13
0.14
0.15
TimeShower:alphaSvalue
0.710
0.700
0.690
0.680
0.670
0.669

Other examples

Example: Optimizing the active muon shield of the SHIP experiment (proposed fixed-target @ CERN SPS)

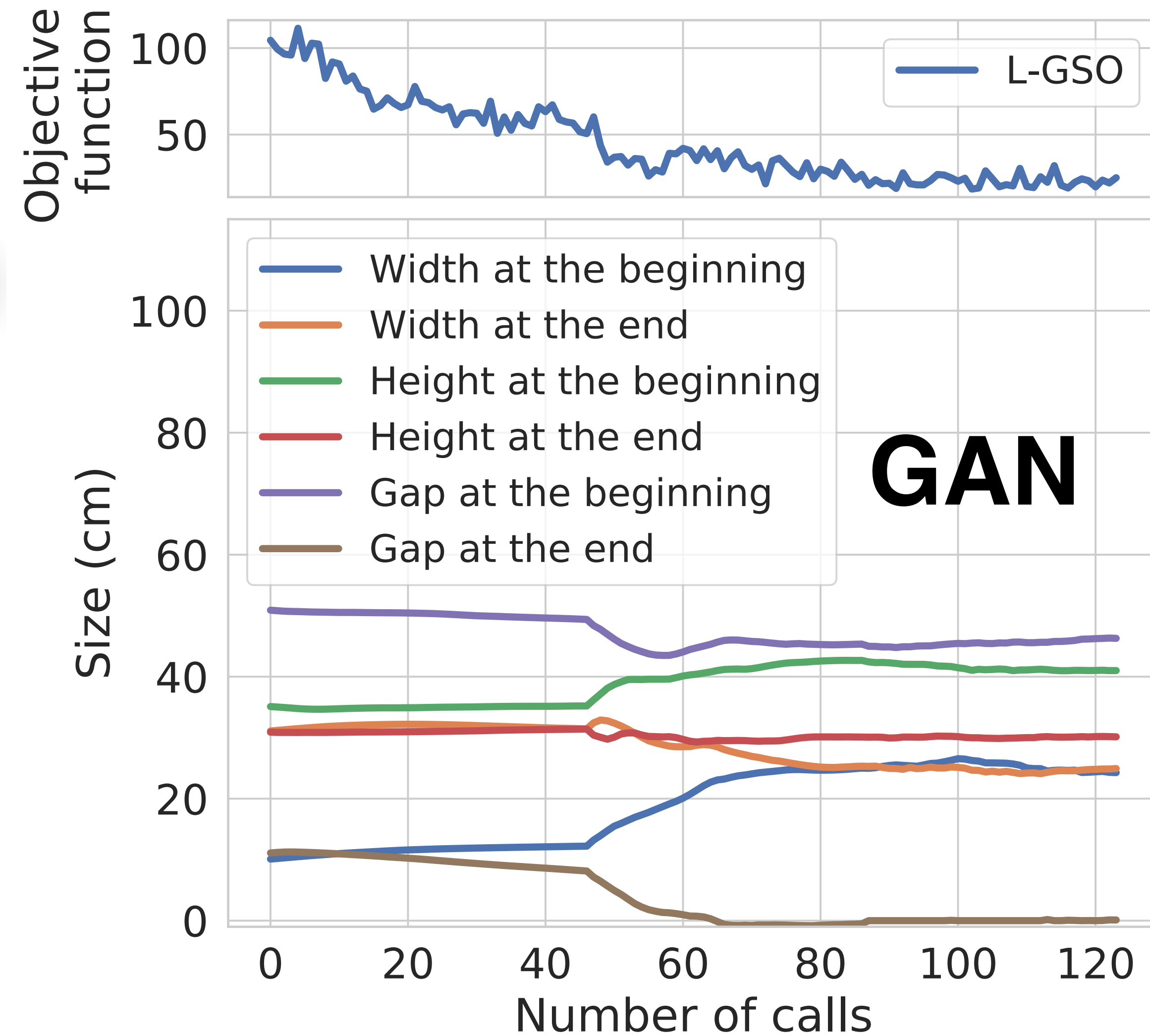
2002.04632

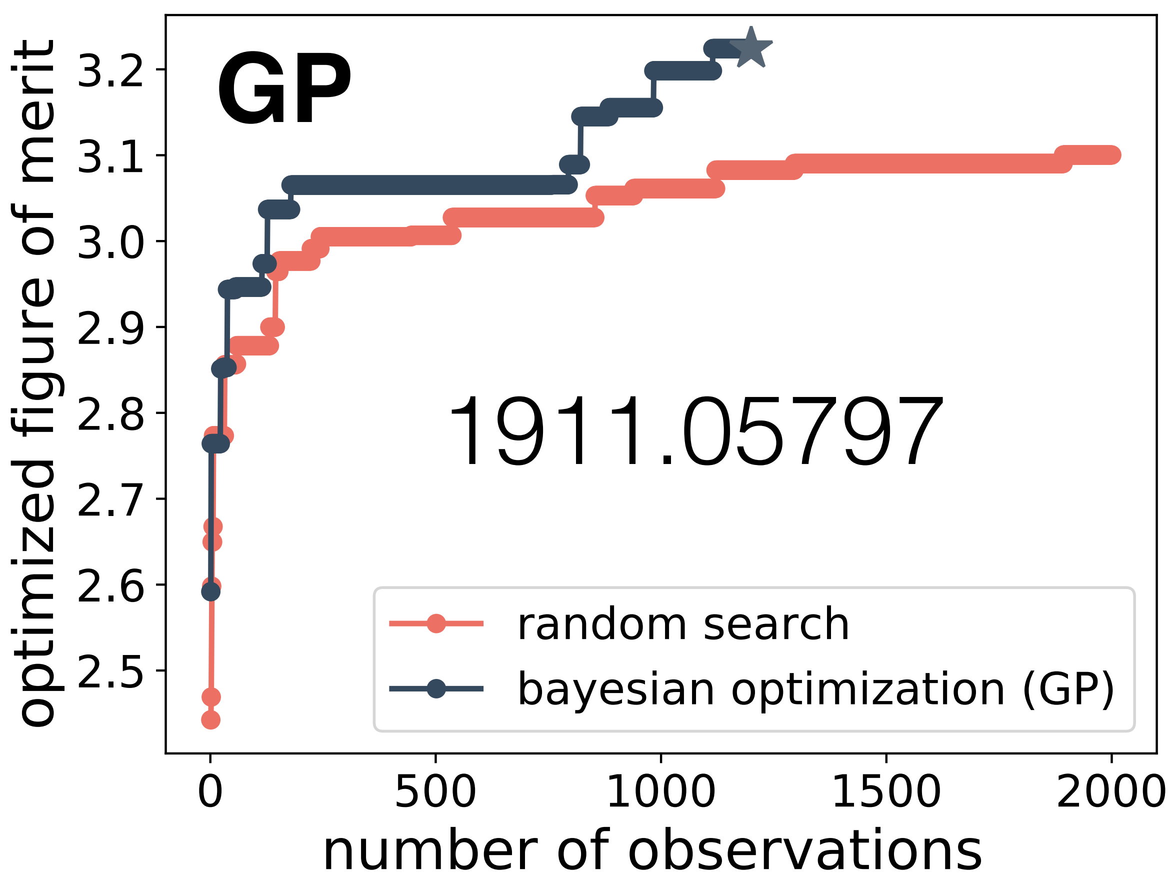


Other examples

Example: Optimizing the active muon shield of the SHIP experiment (proposed fixed-target @ CERN SPS)

2002.04632





RICH detector @ EIC

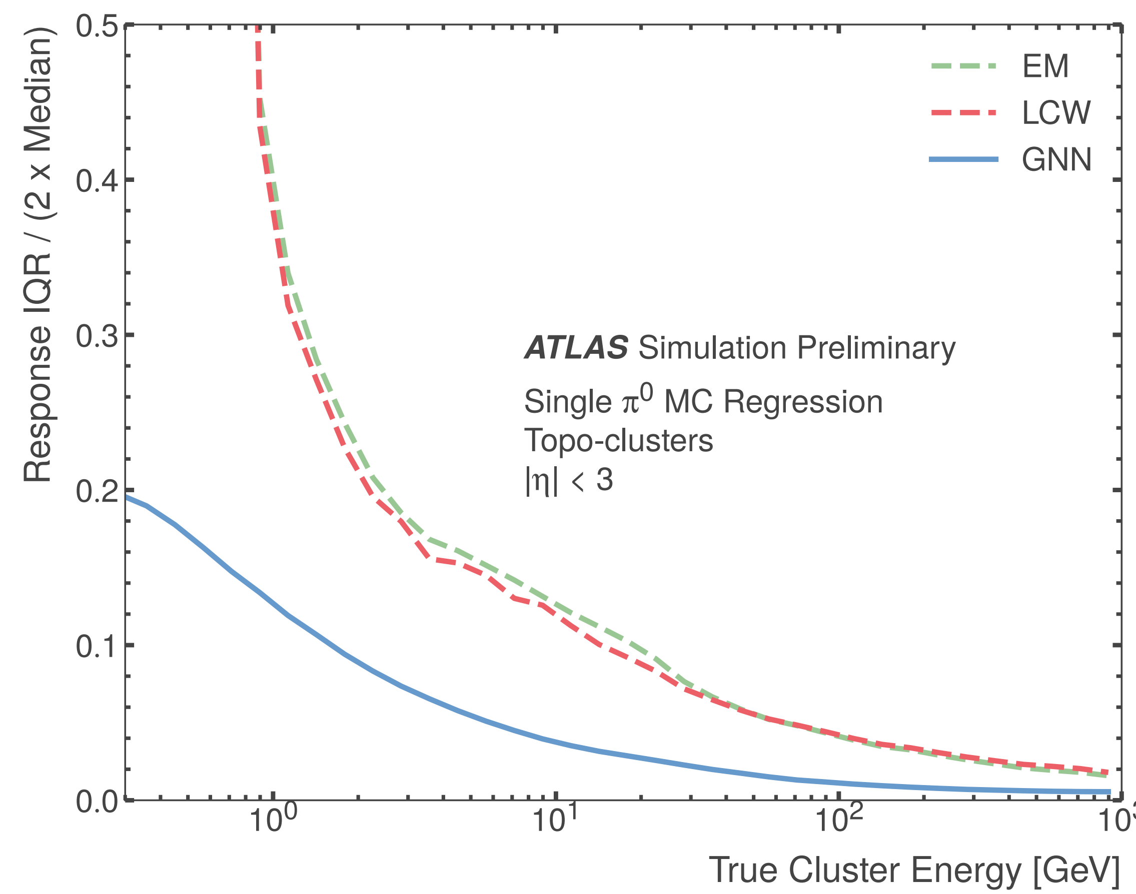
(probably you will hear more about this from Cris!)

Remark about Objective Functions

Today: can we use ML to (1) interpolate in the high-dimensional space, (2) define optimal metrics, and (3) find the best values of θ.

Remark about Objective Functions

If doing gradient-based optimization, the target also needs to be differentiable. For example, target could be resolution of some reconstructed object. This could itself be a neural network!

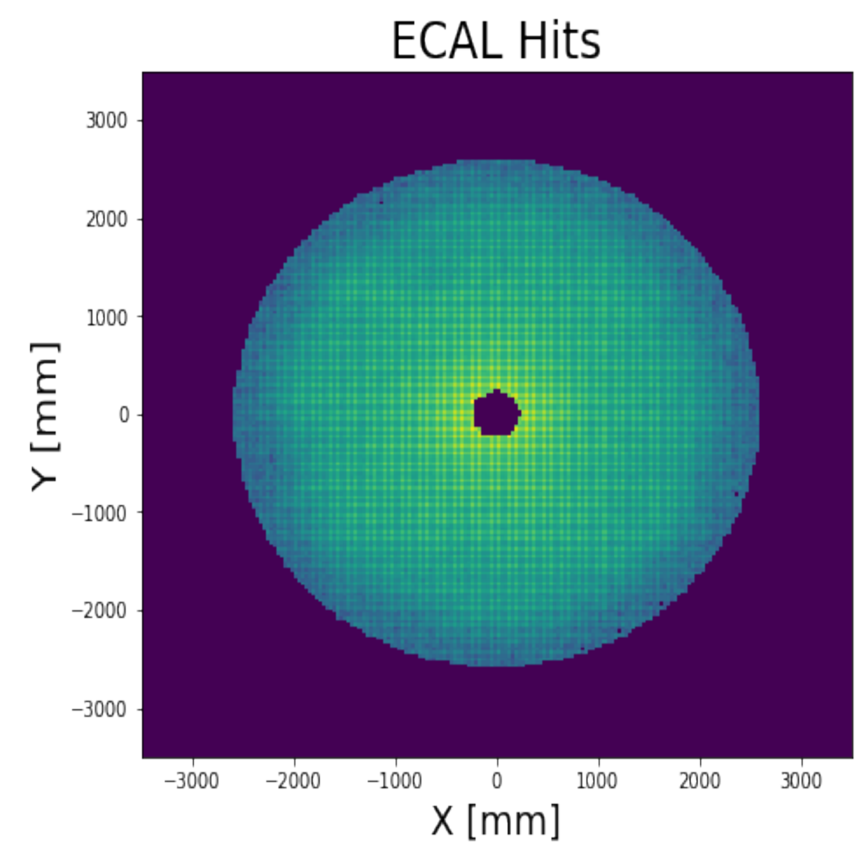


Today: can we use ML to (1) interpolate in the high-dimensional space, (2) define optimal metrics, and (3) find the best values of θ.

Outlook

The EIC detector(s) may the first large-scale detectors optimized with machine learning.

On our side, we are looking into the calorimeter system(s). I am excited to hear about other efforts as well!



Plot from Fernando Torales Acosta

UC RIVERSIDE

Miguel Arratia and co.

BERKELEY LAB

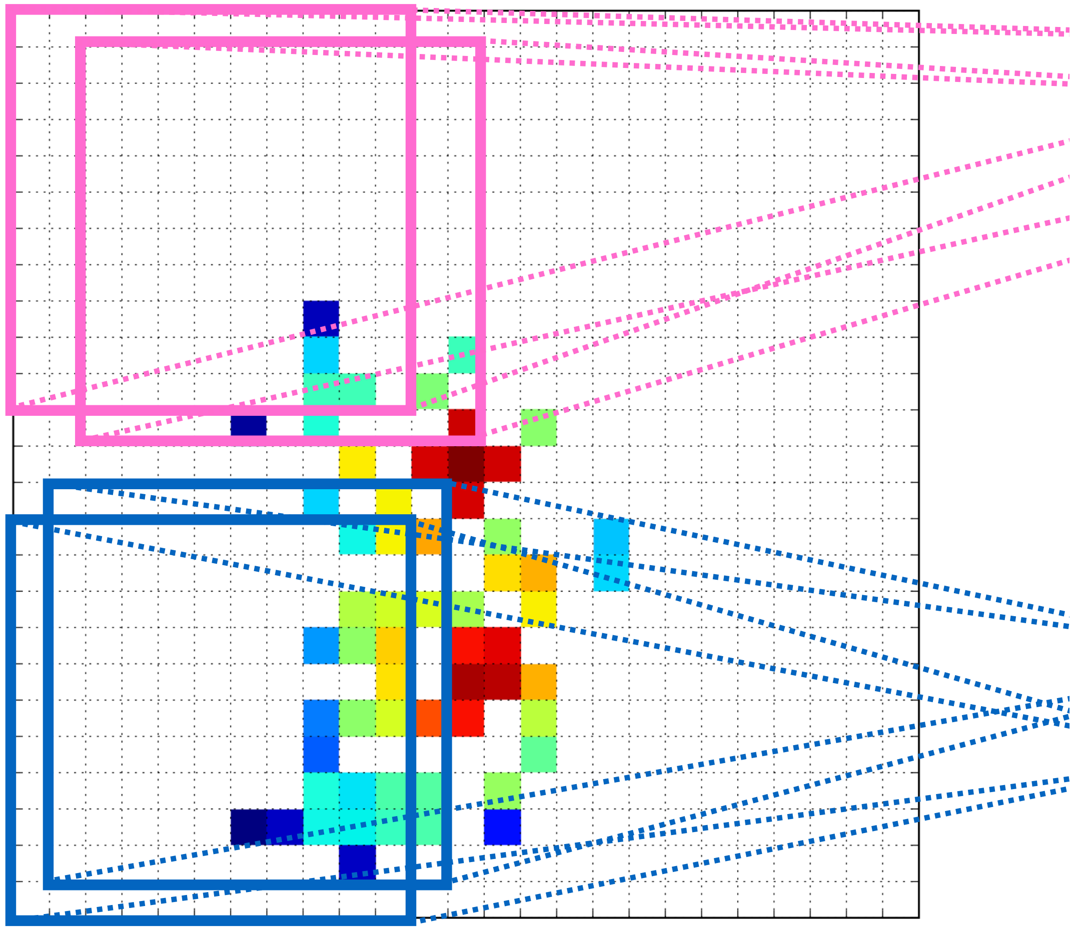
Lawrence Livermore National Laboratory

Aaron Angerami and co.

Conclusions

AI/ML can do more than improve data analysis!

We can use these tools to optimize our detectors - a qualitatively new application of ML!



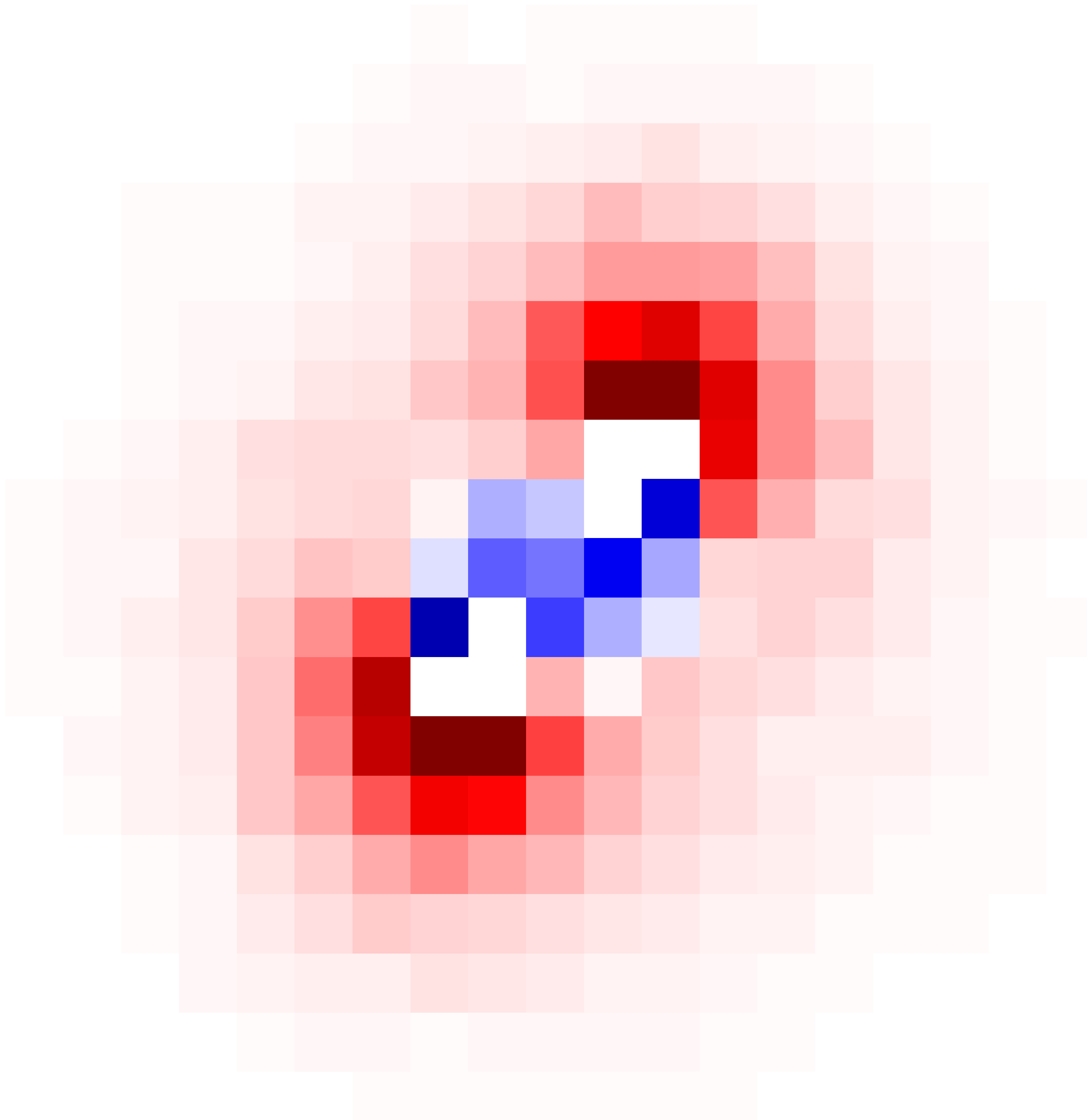
This is an exiting time to be working on this topic - let's use the best tools to get the best physics out of the EIC!

Plug for Snowmass

https://indico.fnal.gov/event/22303/

CompF3 is machine learning and their report has an entire section about detector/accelerator design and control!

Please consider commenting!



Fin.