Deep Generative Models for High Energy Physics

Benjamin Nachman

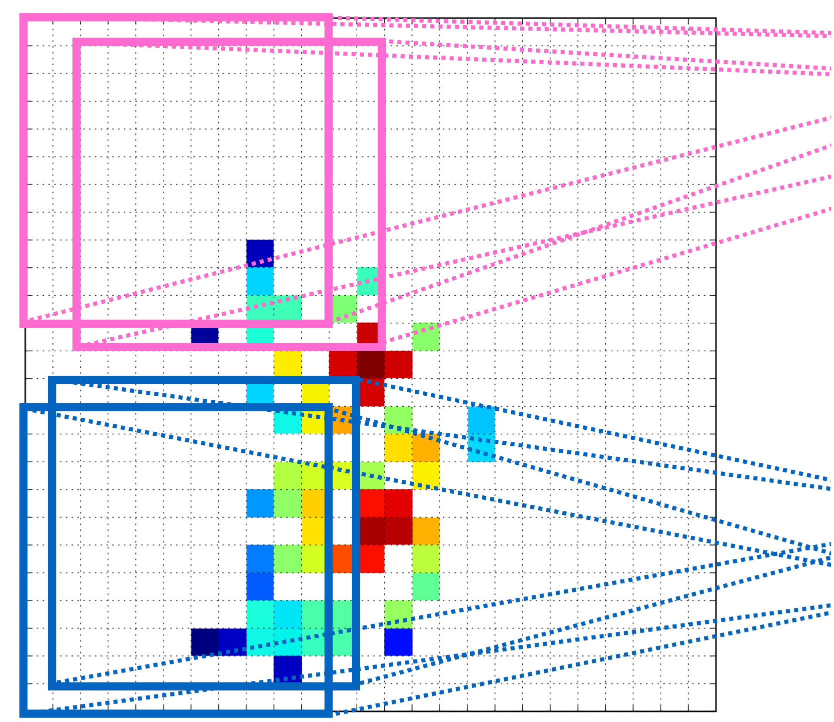
Lawrence Berkeley National Laboratory

bpnachman.com

bpnachman@lbl.gov

@bpnachman

bnachman



BERKELEY LAB

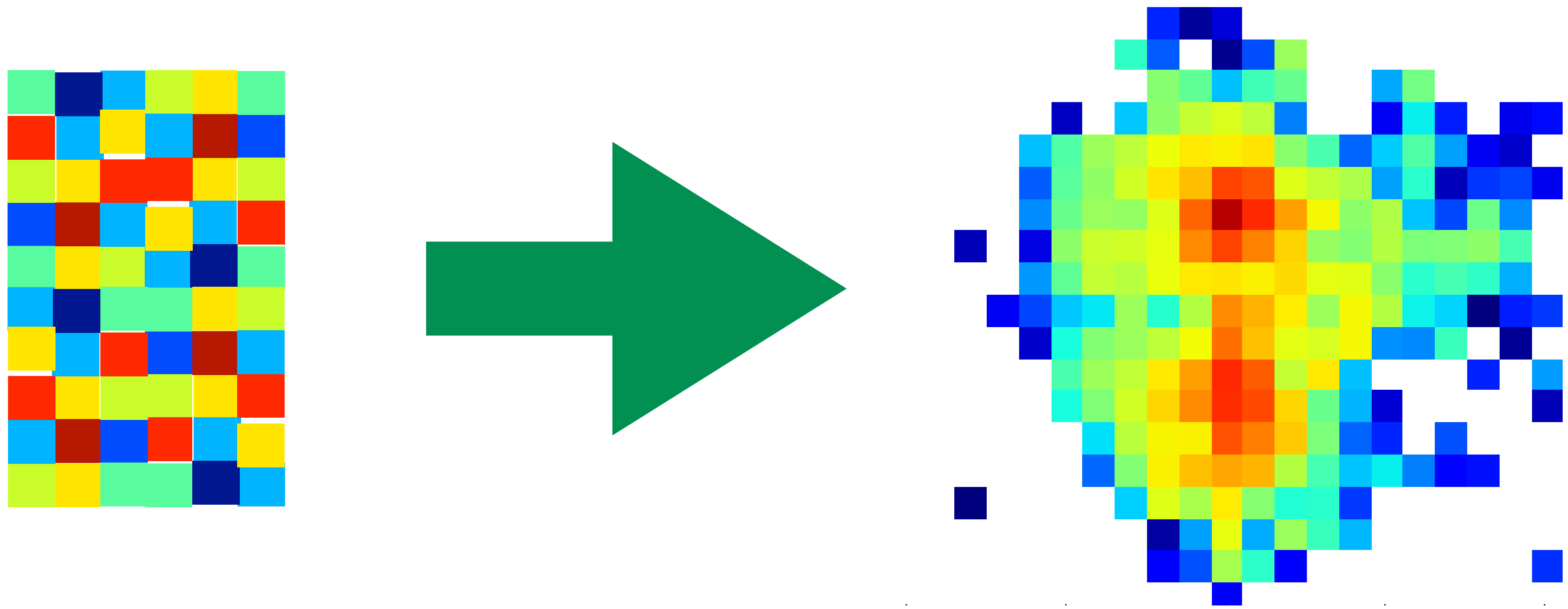
BIDS
BERKELEY INSTITUTE
FOR DATA SCIENCE

UCSD PHYS 139/239

March 7, 2023

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

Restricted Botlzmann Machines

Mixture Density Networks

NFs
Normalizing Flows

Energy-based models

VAEs
Variational Autoencoders

Tools - focus for today

GANs
Generative Adversarial Networks

Score-based

Restricted Botlzmann Machines

Mixture Density Networks

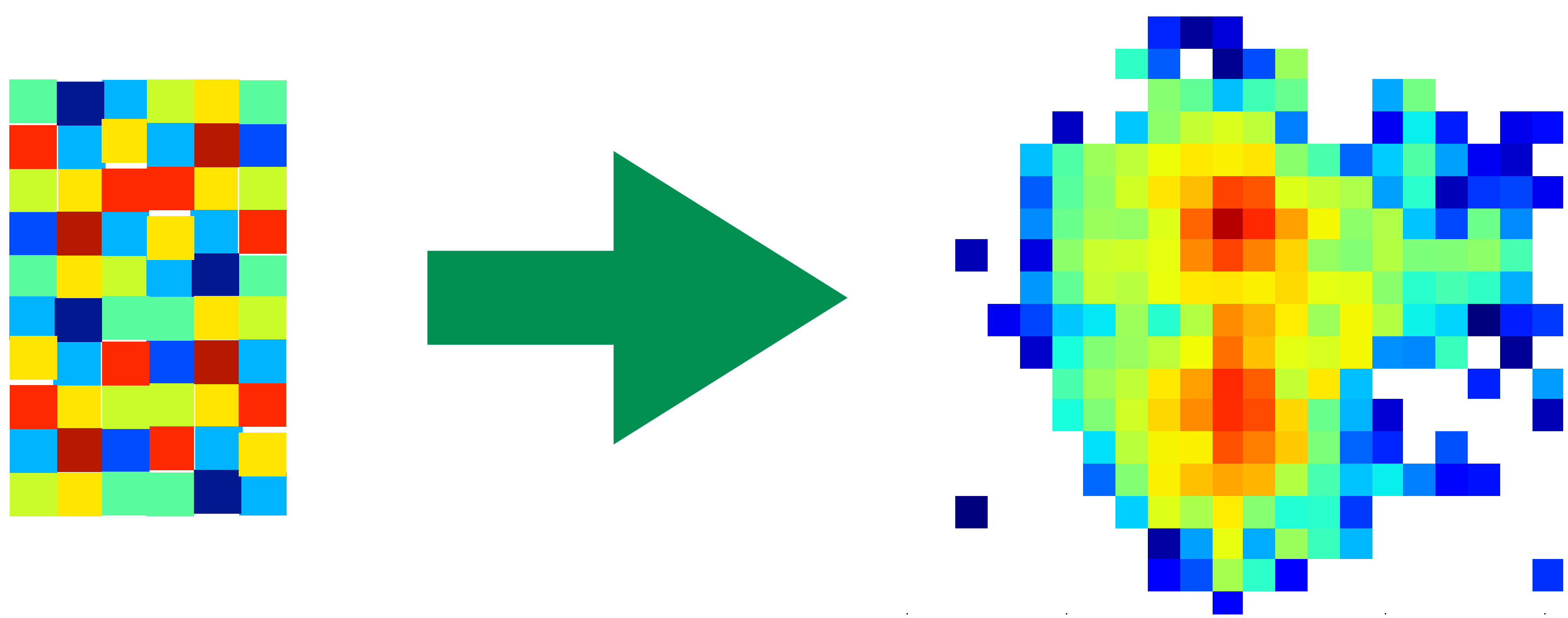
NFs
Normalizing Flows

Energy-based models

VAEs
Variational Autoencoders

Outline

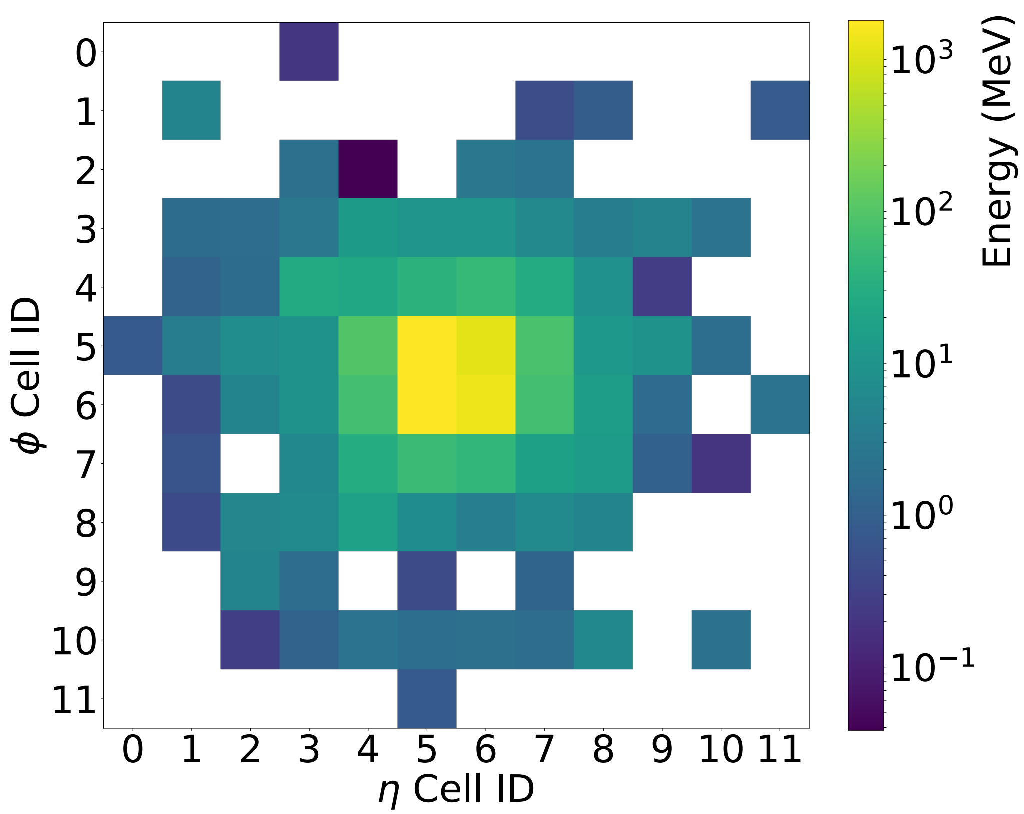
1. Why Generative Models?
2. Warm up: mixture density networks
3. Next: brief overview of GANs, VAEs, NFs
4. Statistical amplification
5. Applications
6. Bonus



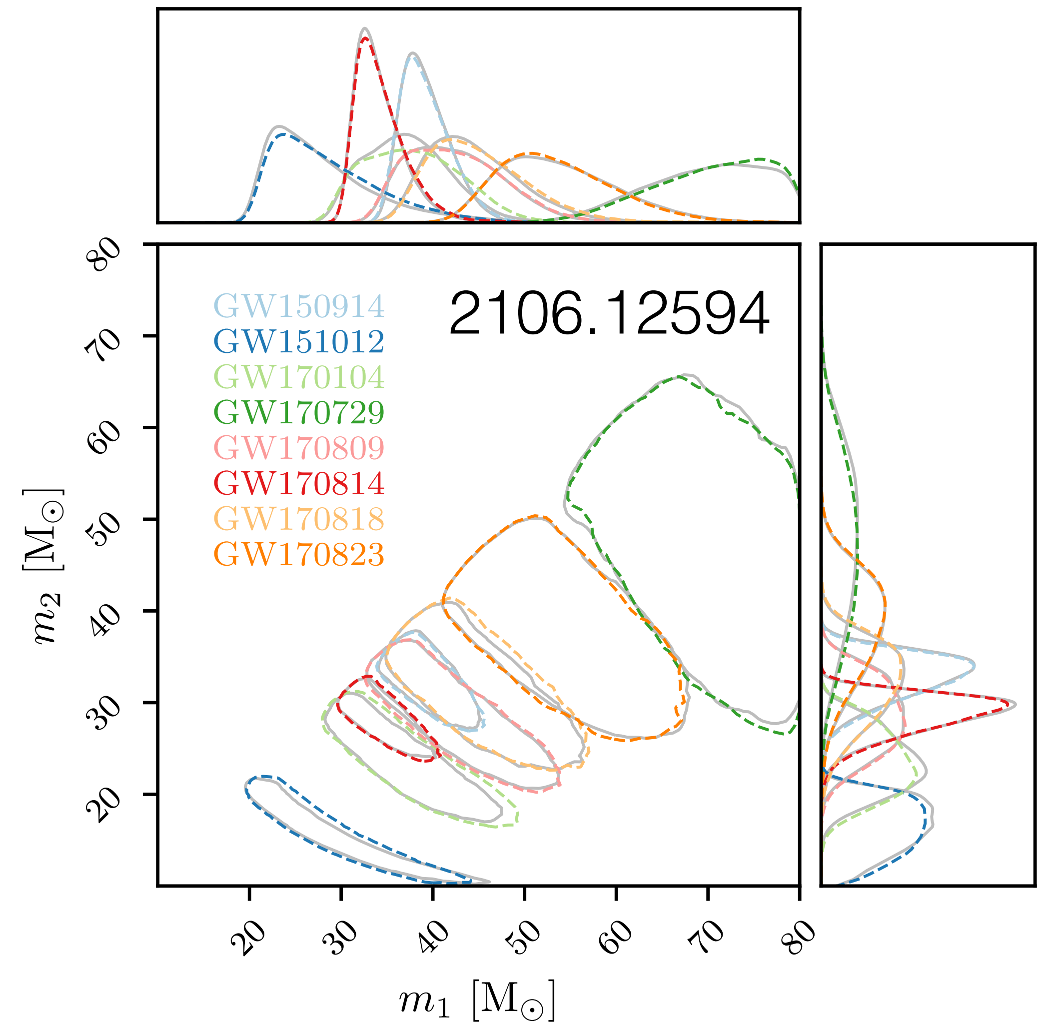
Why generative models?

1. Augment/replace slow physics-based simulations
2. (Fast) Confidence Intervals / Posterior Analysis
3. Background estimation / anomaly detection
4. …

1705.02355



Calorimeter sim. x10^5 faster



GW inference x10^4 faster

Warm-up: MDNs

If you know the family of functions that generated your data, you can't beat fitting that function.

(I'll be more precise later about "beat")

A lot of physics data are nearly Gaussian, or combinations of a small number of Gaussians, possibly with a complex dependence on energy, time, etc.

Classical mixture model:

$$p(x) \propto \sum_i \alpha_i \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

Warm-up: MDNs

In case you have not seen this notation…

$$G(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

In 1D, $\Sigma = \sigma^2$.

Classical mixture model:

$$p(x) \propto \sum_i \alpha_i \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

Warm-up: MDNs

If you know the family of functions that generated your data, you can't beat fitting that function.

(I'll be more precise later about "beat")

A lot of physics data are nearly Gaussian, or combinations of a small number of Gaussians, possibly with a complex dependence on energy, time, etc.

Classical mixture model:

$$p(x) \propto \sum_i \alpha_i \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

Warm-up: MDNs

If you know the family of functions that generated your data, you can't beat fitting that function.

(I'll be more precise later about "beat")

A lot of physics data are nearly Gaussian, or combinations of a small number of Gaussians, possibly with a complex dependence on energy, time, etc.

Classical mixture model:

$$p(x) \propto \sum_i \alpha_i \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

Mixture density network:

$$p(x\,|\,z) \propto \sum_i \alpha_i(z)\ \exp\left(-\frac{1}{2}(x-\mu(z))^T \Sigma(z)^{-1} (x-\mu(z))\right)$$

MDNs: details

Mixture density network:

$$p(x\,|\,z) \propto \sum_i \alpha_i(z)\,\exp\left(-\frac{1}{2}(x-\mu(z))^T\Sigma(z)^{-1}(x-\mu(z))\right)$$

The “parameters” $\alpha(z), \mu(z), \Sigma(z)$ are deep neural networks.

Since Σ is symmetric and positive semi-definite, need to take care when designing its architecture.

Train an MDN by minimizing $-\log(p(x\,|\,z))$.

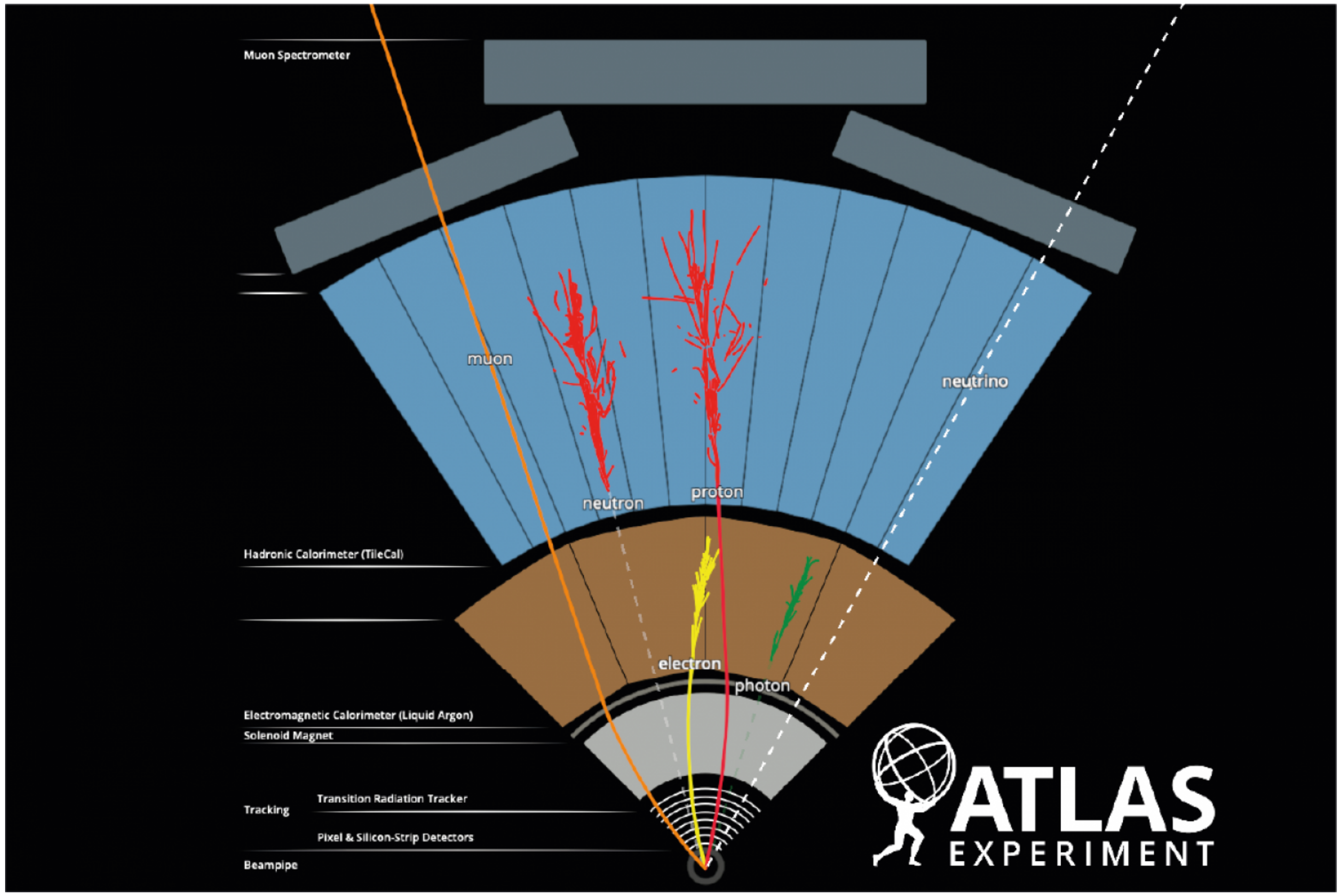
When x is 1D, there is only one component, and Σ is fixed, then this reduces to mean squared error!

MDNs: code


```
from tensorflow import keras
import tensorflow_probability as tfp

tfd = tfp.distributions
tfpl = tfp.layers
tfk = tf.keras
tfkl = tf.keras.layers


n = len(X[:,0:])
z = np.zeros(n)
x = X[:,0:]

# Model the distribution of y given x with a Mixture Density Network.
event_shape = np.shape(X)
num_components = 5
params_size =
tfpl.MixtureSameFamily.params_size(num_components,component_params_size=tfpl.MultivariateNormalTriL.params_size(event_shape[0]))
model = tfk.Sequential([
  tfkl.Dense(32, activation='relu'),
  tfkl.Dense(64, activation='relu'),
  tfkl.Dense(64, activation='relu'),
  tfkl.Dense(params_size, activation=None),
  tfpl.MixtureSameFamily(num_components, tfpl.MultivariateNormalTriL(event_shape[0]))
])

# Fit.
batch_size = 100
model.compile(optimizer='adam',
              loss=lambda x, model: -model.log_prob(x))
myhistory = model.fit(z, x,
          batch_size=batch_size,
          epochs=50,
          steps_per_epoch=n // batch_size)
```

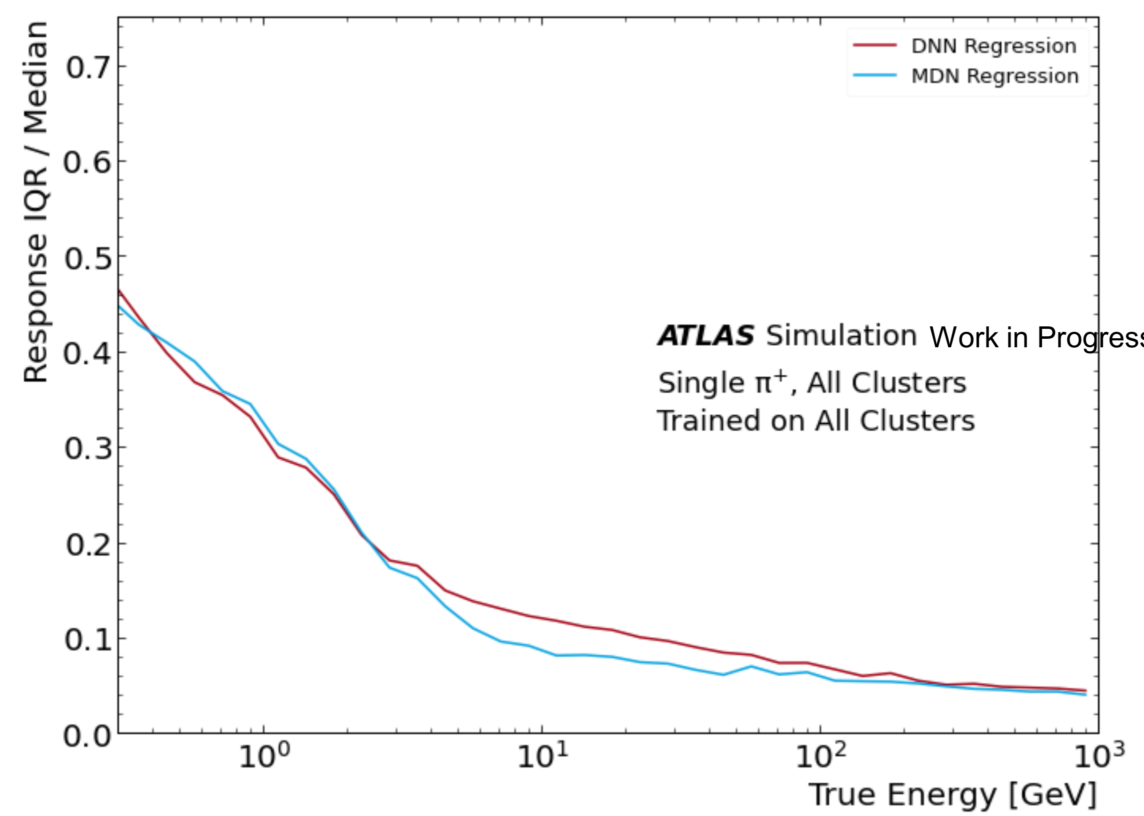
(equivalently simple in PyTorch)

MDNs in action



Example: energy regression for hadrons in the ATLAS calorimeter

(figures without citations are links)



Simultaneously learn confidence interval (“resolution”) and point estimate

MDNs: overview

Pros:

- Access to the density
- Fast to sample
- Easy to specify
- Easy to train

Cons:

- Do not scale well to high dimensions
- Not good when the data are not nearly Gaussian(s)

Up next: deep generative models with more flexibility

Tools

GANs
Generative Adversarial Networks

Score-based

Restricted Botlzmann Machines

Mixture Density Networks

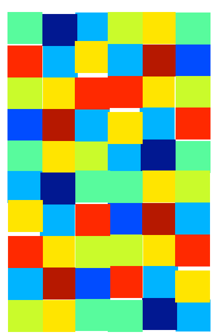
NFs
Normalizing Flows

Energy-based models

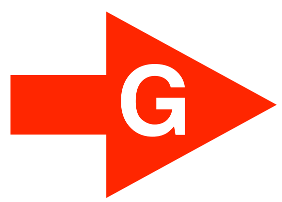
VAEs
Variational Autoencoders

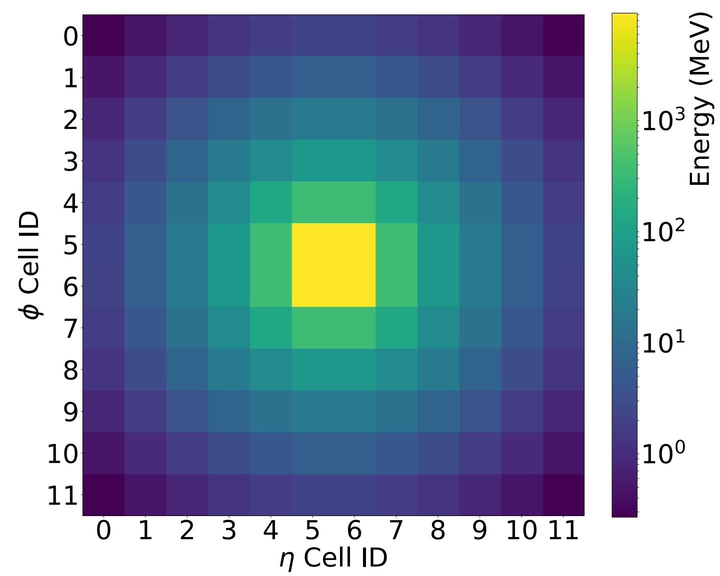
Introduction: GANs

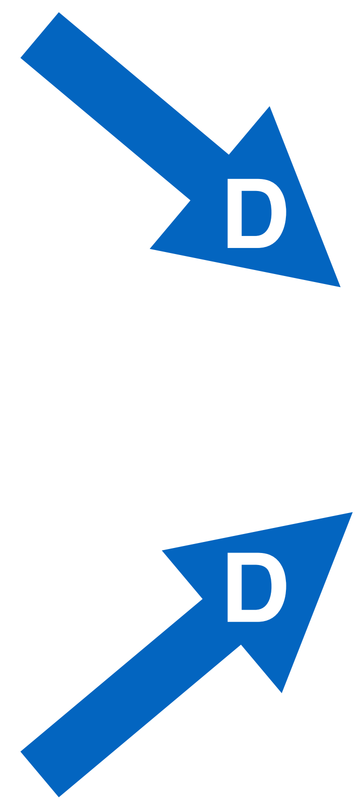
Generative Adversarial Networks (GANs):
*A two-network game where one **maps noise to structure** and one **classifies images as fake or real**.*



noise

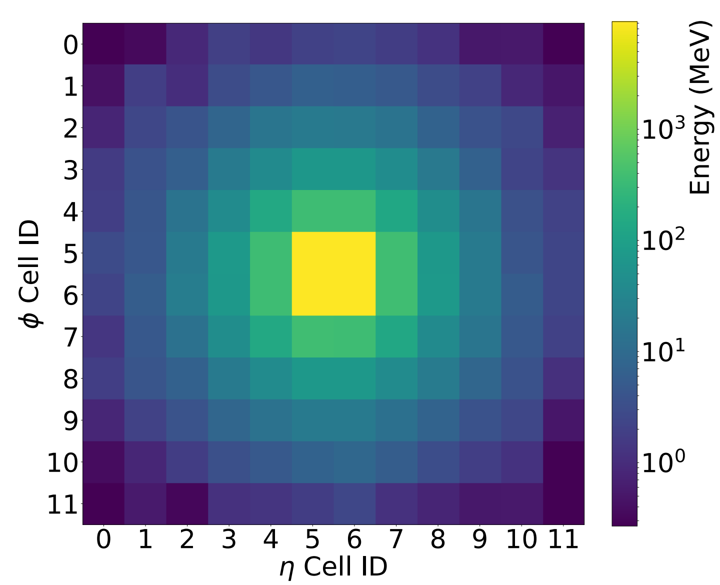






{real,fake}

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

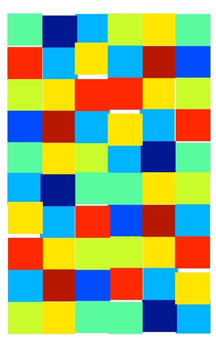


Physics-based simulator or data

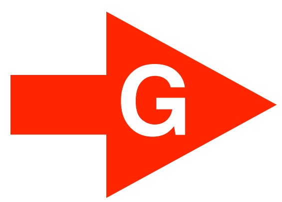
Introduction: GANs

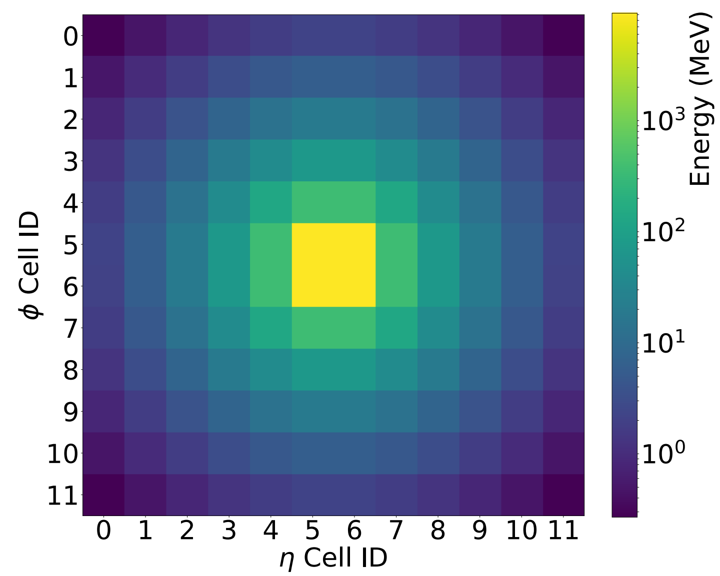
Vanilla GAN loss:

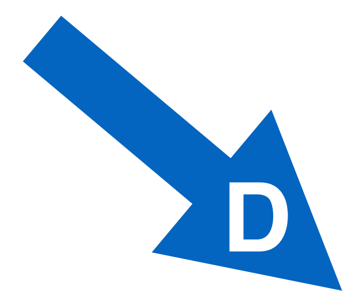
$$-\sum_{\text{X}\sim\text{data}} \log(D(x)) - \sum_{\text{Z}\sim\text{noise}} \log(1 - D(G(z)))$$



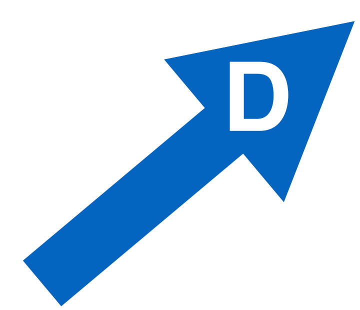
noise



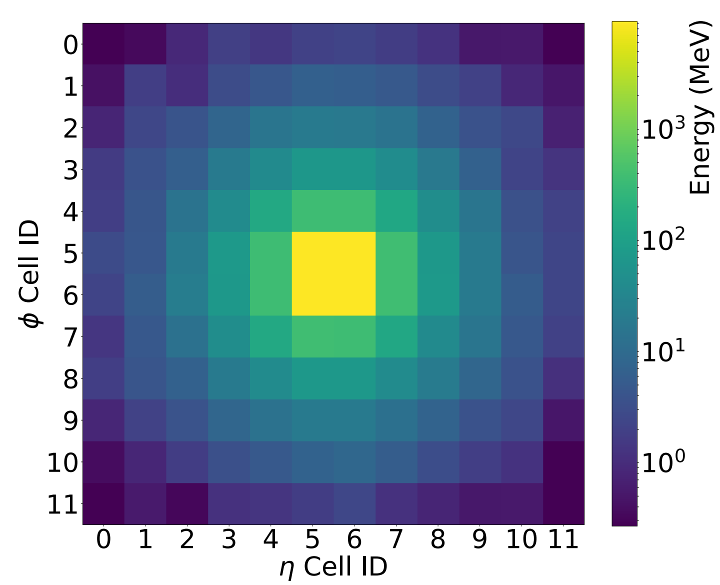




{real,fake}



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



Physics-based simulator or data

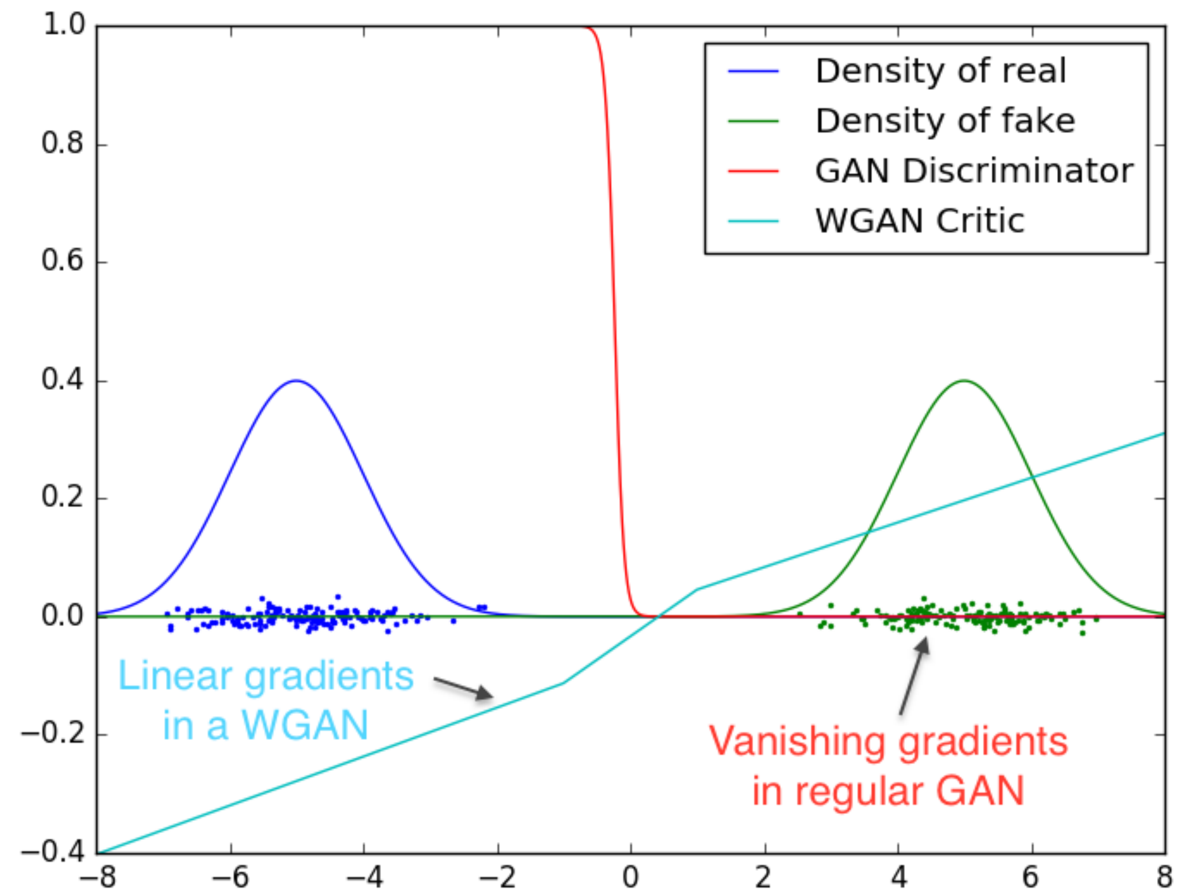
Introduction: GANs

Vanilla GAN loss:

$$-\sum_{\text{X}\sim\text{data}} \log(D(x)) - \sum_{\text{Z}\sim\text{noise}} \log(1 - D(G(z)))$$

There are many variations on this theme*.
One important variation is called the Wasserstein or WGAN.

Schematically, the idea of WGAN is to replace the discriminator with a notion of distance ('earth moving') between the real and fake data



*see e.g. https://github.com/hindupuravinash/the-gan-zoo

GANs: overview

Pros:

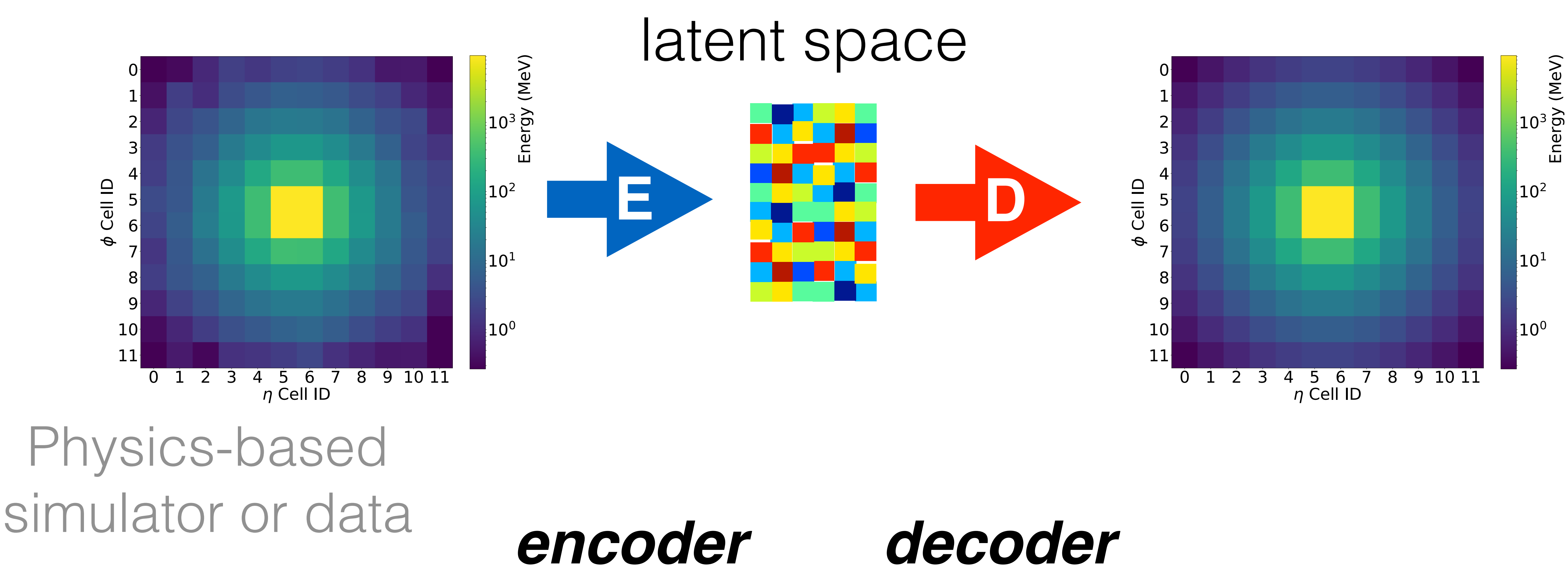
- Fast to sample
- Easy to specify (no restrictions on G)
- Easy to make high dimensional

Cons:

- No access to density
- (Very) Hard to train (minimax)
- Mode collapse

Introduction: AEs

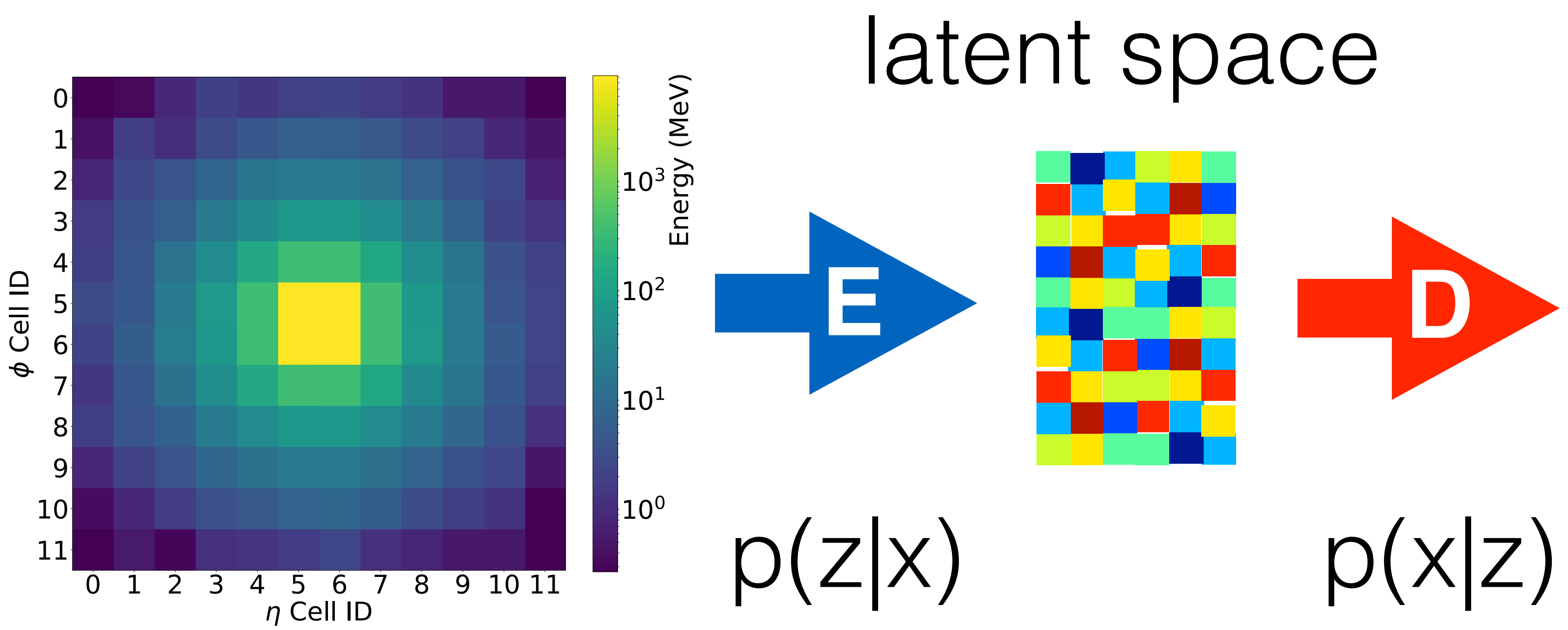
Autoencoders (AEs):
A pair of networks that embed the data into a latent space and decode back to the data space.



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: VAEs

Vanilla VAE loss:

$$\sum_{Z\sim p(z|x)} (x - D(z))^2 + \mathrm{KL}(p(z\,|\,x)\,||\,p_0(z))$$

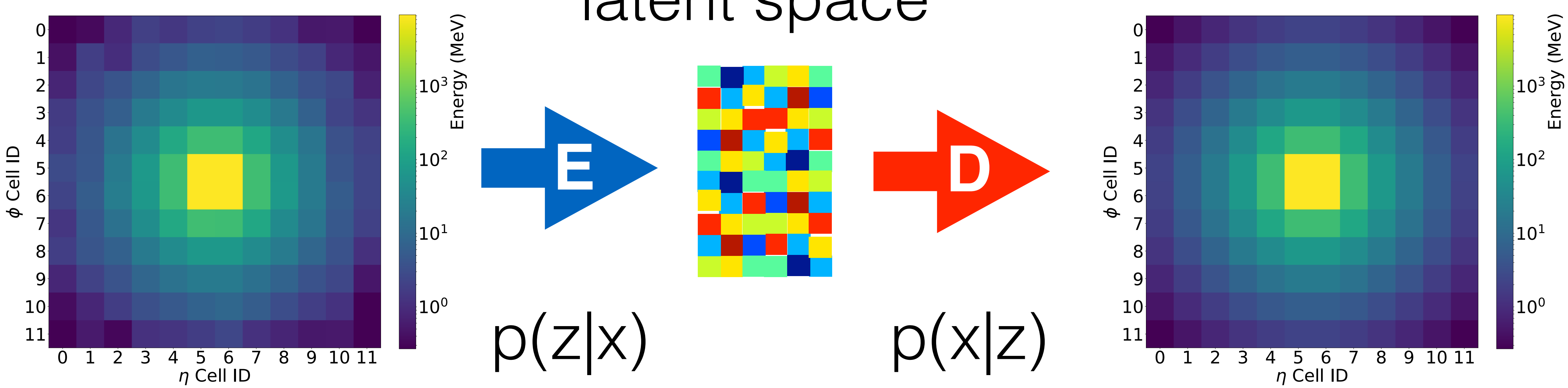
prior

KL-divergence

"reconstruction"

"regularization"

latent space



Physics-based simulator or data

Probabilistic ***encoder***

Probabilistic ***decoder***

Why "variational"? Another way of looking at this is the Evidence Lower Bound (ELBO) and p(z|x) is the variational posterior.

VAEs: overview

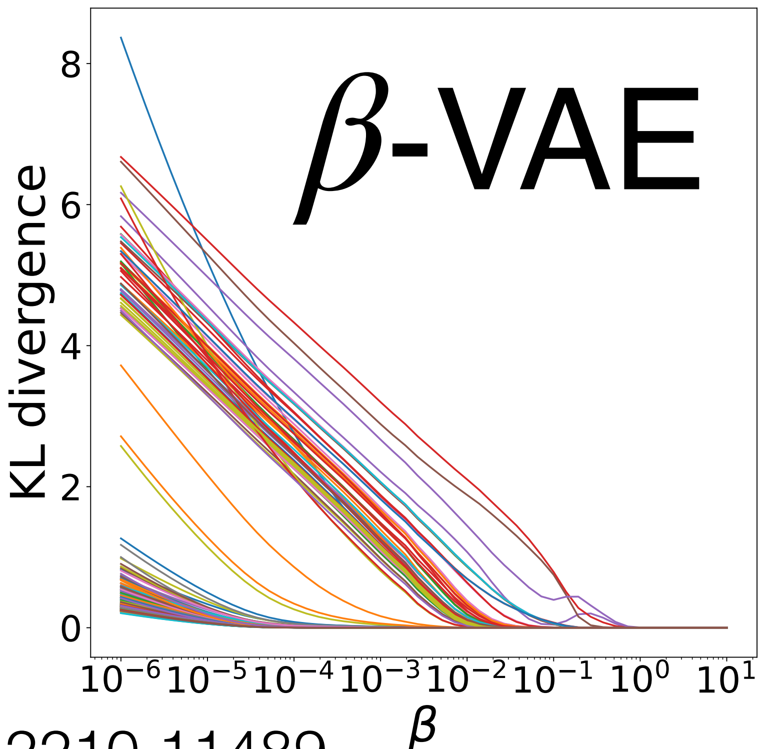
Pros:

- Fast to sample
- Easy to specify (no restrictions on G)
- Easy to make high dimensional
- Easy to train

Cons:

- No access to density
- Tends to over-smooth the density

ex. AE variations: β-VAE, PAE

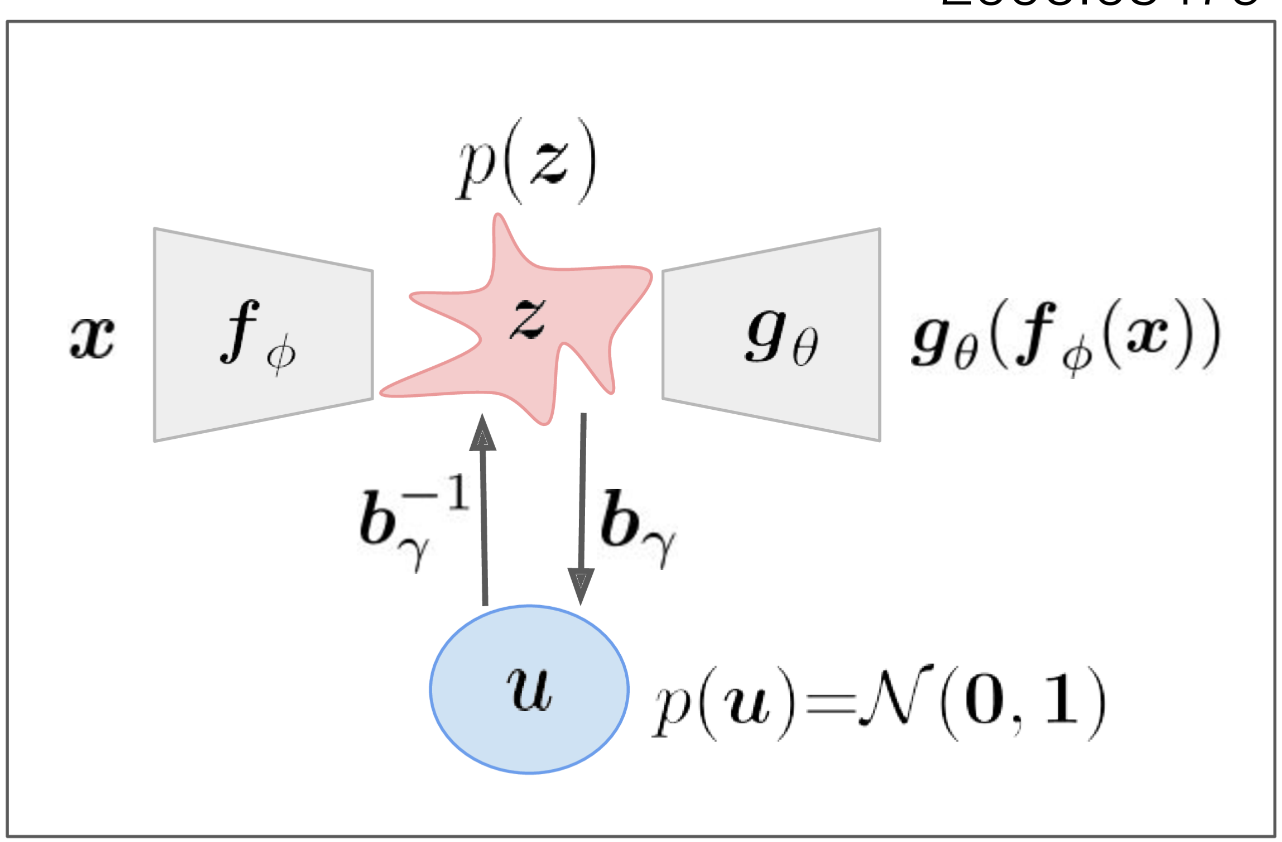


2210.11489

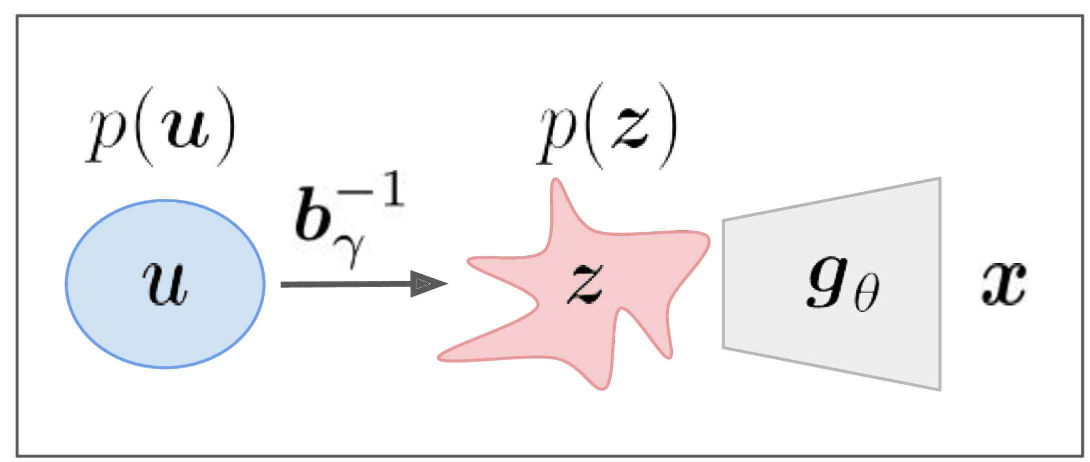
β-VAE loss:

$$\sum_{Z \sim p(z|x)} (x - D(z))^2 + \beta \, \mathrm{KL}(p(z \,|\, x) \,||\, p_0(z))$$

2006.05479



Probabilistic AE

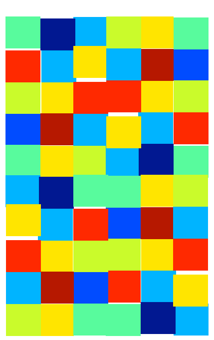


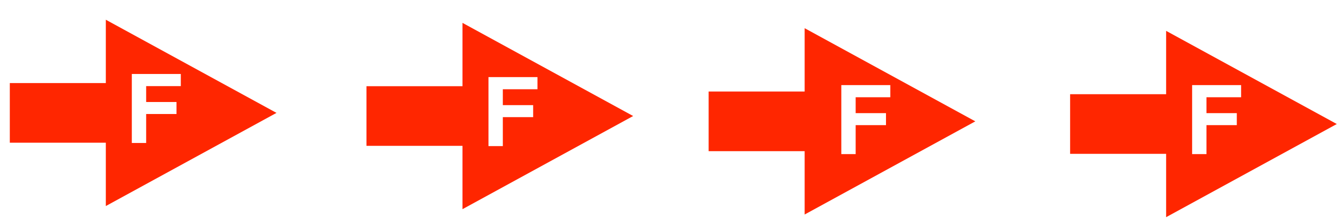
Introduction: NFs

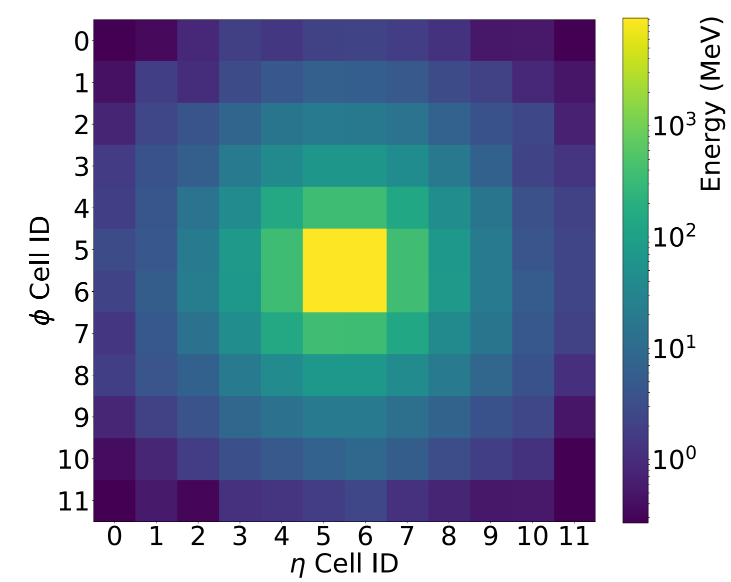
Normalizing Flows (NFs):
A series of invertible transformations mapping a known density into the data density.

Optimize via maximum likelihood

Loss: $-\log(p(x))$







latent space

Invertible transformations with tractable Jacobians

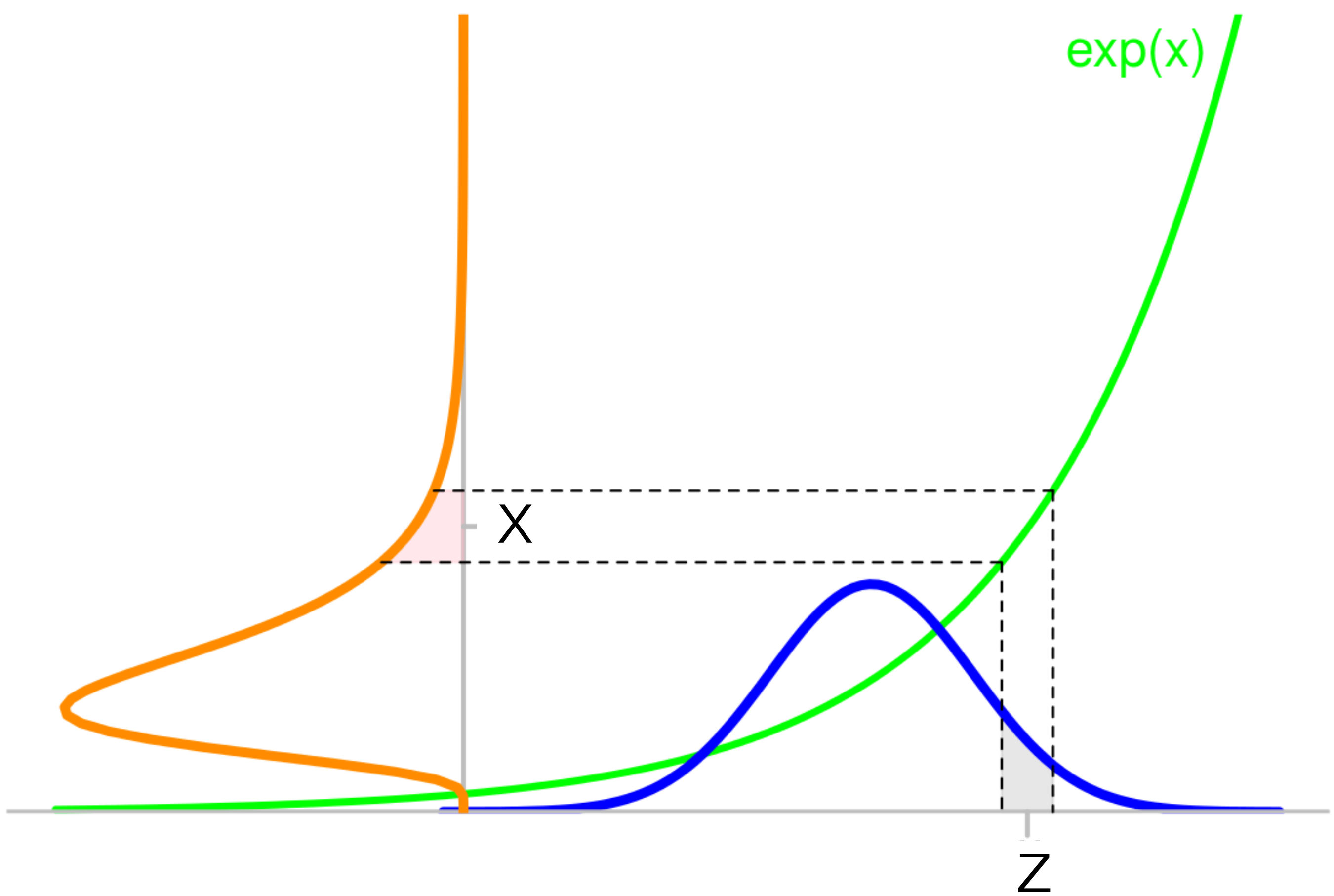
p(z)

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

p(x)

Introduction: NFs

In case you have not seen the change of variables formula…



The Jacobian is a matrix of partial derivatives.

Food for thought: Suppose the PDF of X is f and the CDF is F. What is the density of F(X)?

latent space

with tractable Jacobians

p(z)

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

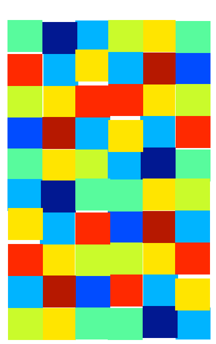
p(x)

Introduction: NFs

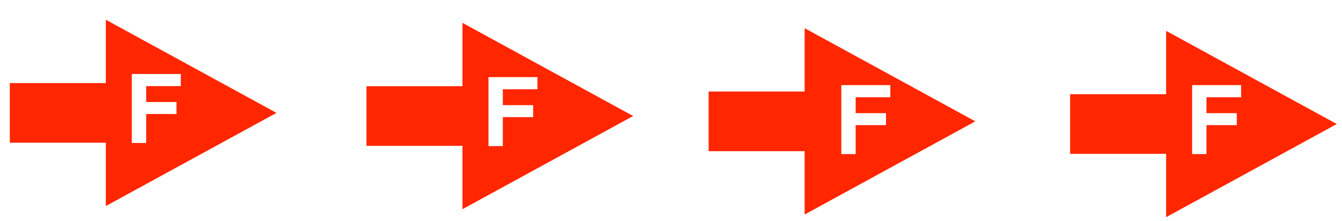
Normalizing Flows (NFs):
A series of invertible transformations mapping a known density into the data density.

Optimize via maximum likelihood

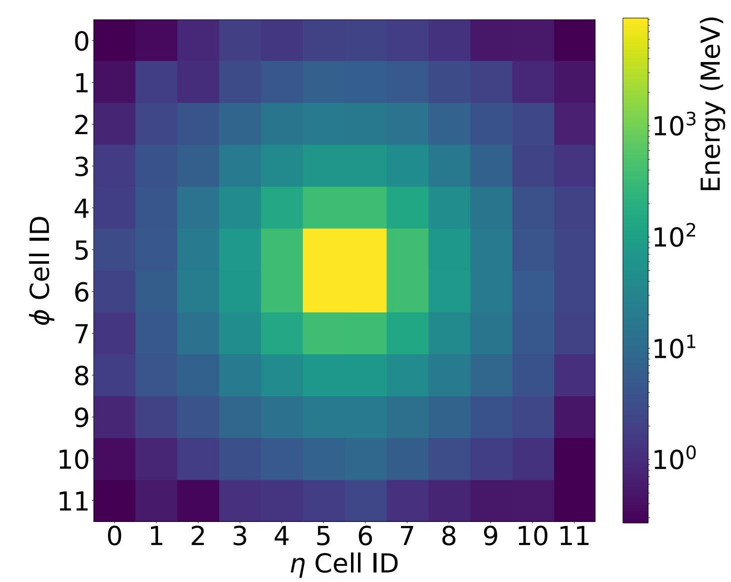
Loss: $-\log(p(x))$



latent space



Invertible transformations with tractable Jacobians



p(z)

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

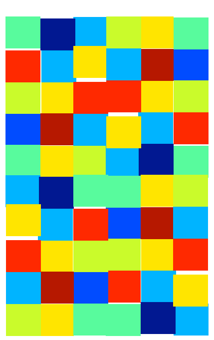
p(x)

Introduction: NFs

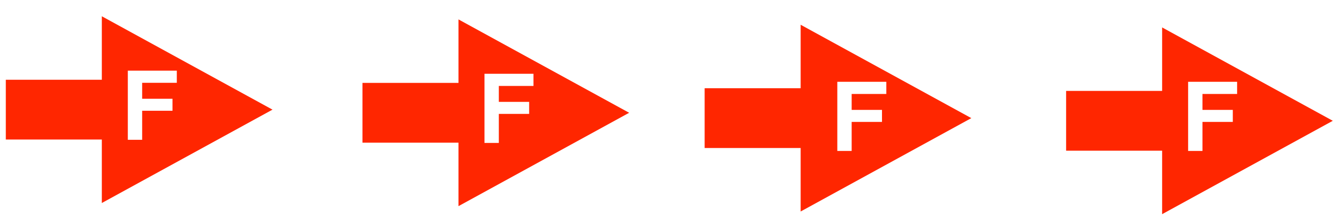
Normalizing Flows (NFs):
A series of invertible transformations mapping a known density into the data density.

Example: $F(x_i) = z_i + \mathrm{NN}(x_{i-1})$
("autoregressive")

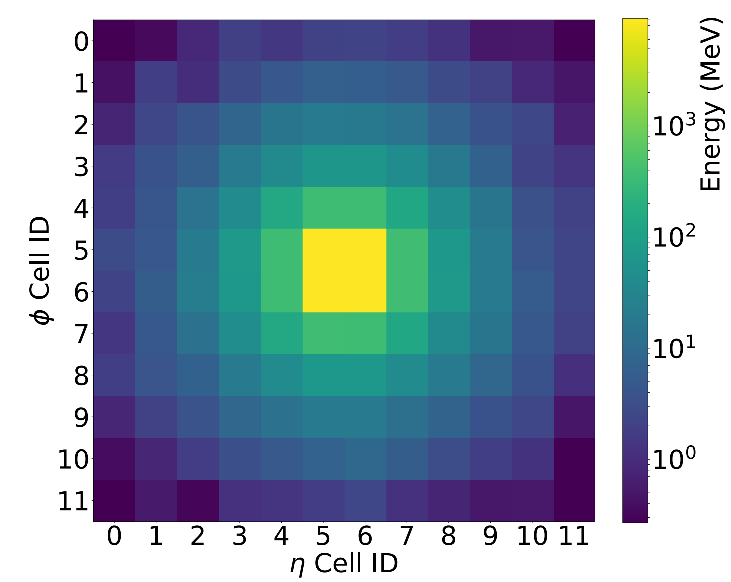
Optimize via maximum likelihood



latent space



Invertible transformations with tractable Jacobians



p(z)

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

p(x)

NFs: overview

Pros:

- Usually only fast to sample OR to estimate the density
- Access to the density
- Easy to train

Cons:

- Sometimes hard to make expressive enough
- “Generator” is highly limited in form
- Cannot learn topology

Introduction: Score-based

Score-based
Learn the gradient of the density instead of the probability density itself.

"score"

Loss: $|f(x) - \nabla p(x)|^2$

…but since we don't know ∇p, we make use of a trick whereby the data are perturbed and it is sufficient to match the score of the perturbing function (!)

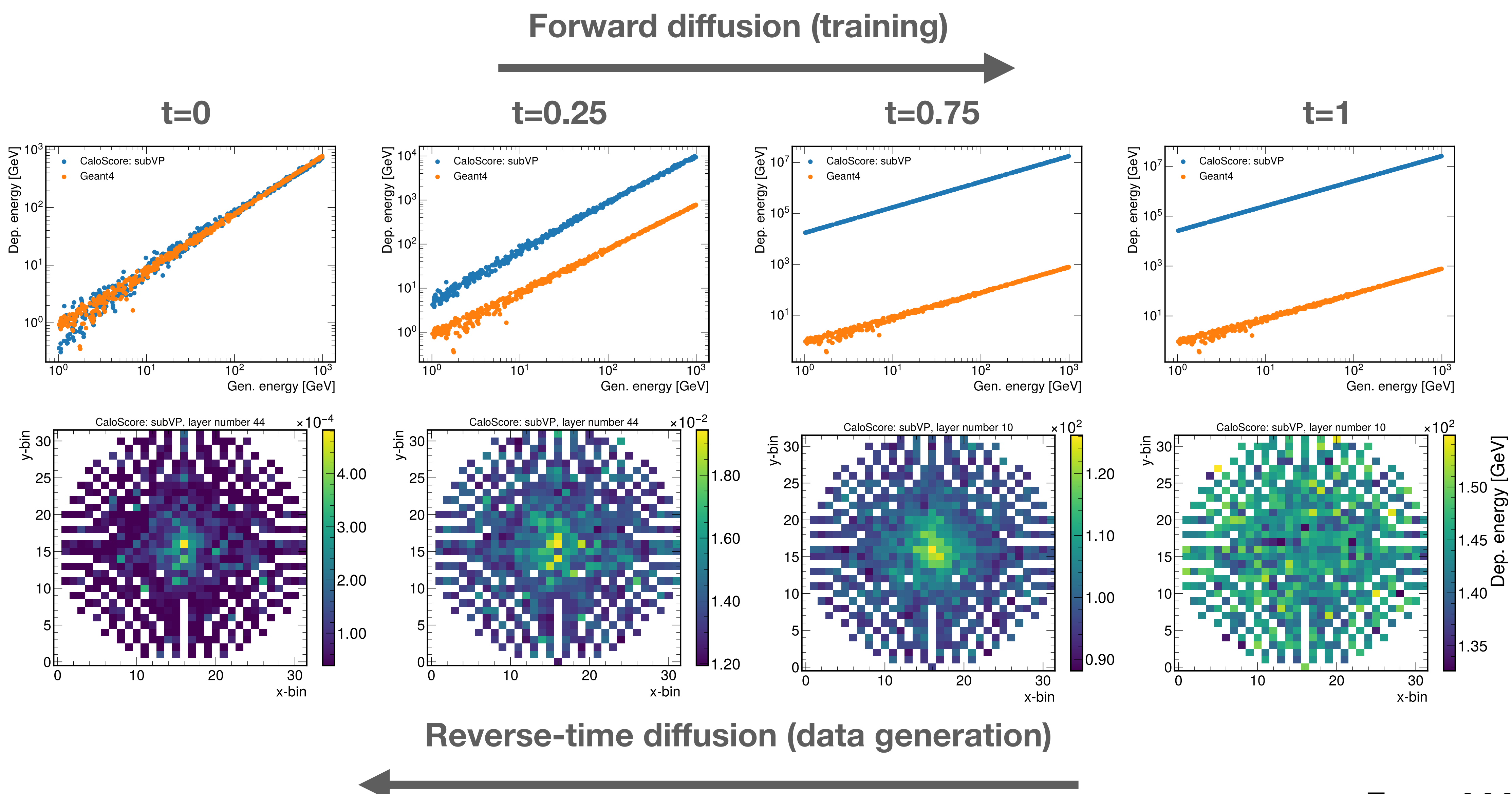
This turns the problem into a stochastic differential equation which is the same as diffusion (sometimes this is called a "diffusion model")

From 2206.11898

Introduction: Score-based

Score-based

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



From 2206.11898

Score-based: overview

Pros:

- Many of the benefits of a NF, but unrestricted functions
- Access to the density with some work
- Easy to train
- Currently the "best" on the market

Cons:

- A bit slow to train / evaluate

Image Generation on CIFAR-10

Rank	Model	FID	Clean-FID-10k	Inception score	bits/dimension	FID-10k-test	Paper	Code	Result	Year	Tags
1	**EDM-G++** (unconditional)	1.77			2.55		Refining Generative Process with Discriminator Guidance in Score-based Diffusion Models			2022	Diffusion, Score-based
2	**StyleGAN-XL**	1.85					StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets			2022	GAN
3	**STF** (unconditional)	1.90					Stable Target Field for Reduced Variance Score Estimation in Diffusion Models			2023	Score-based
4	**LSGM-G++** (FID)	1.94			3.42		Refining Generative Process with Discriminator Guidance in Score-based Diffusion Models			2022	Score-based
5	**PSLD** (ODE)	2.10		9.93			Generative Diffusions in Augmented Spaces: A Complete Recipe			2023	Score-based
6	**LSGM** (FID)	2.10			3.43		Score-based Generative Modeling in Latent Space			2021	Score-based, VAE
7	**Subspace Diffusion** (NSCN++)	2.17		9.99			Subspace Diffusion Generative Models			2022	Score-based
8	**LSGM** (balanced)	2.17			2.95		Score-based Generative Modeling in Latent Space			2021	VAE, Score-based
9	**NCSN++**	2.20		9.73			Score-Based Generative Modeling through Stochastic Differential Equations			2020	Score-based
10	**PSLD** (SDE)	2.21					Generative Diffusions in Augmented Spaces: A Complete Recipe			2023	Score-based

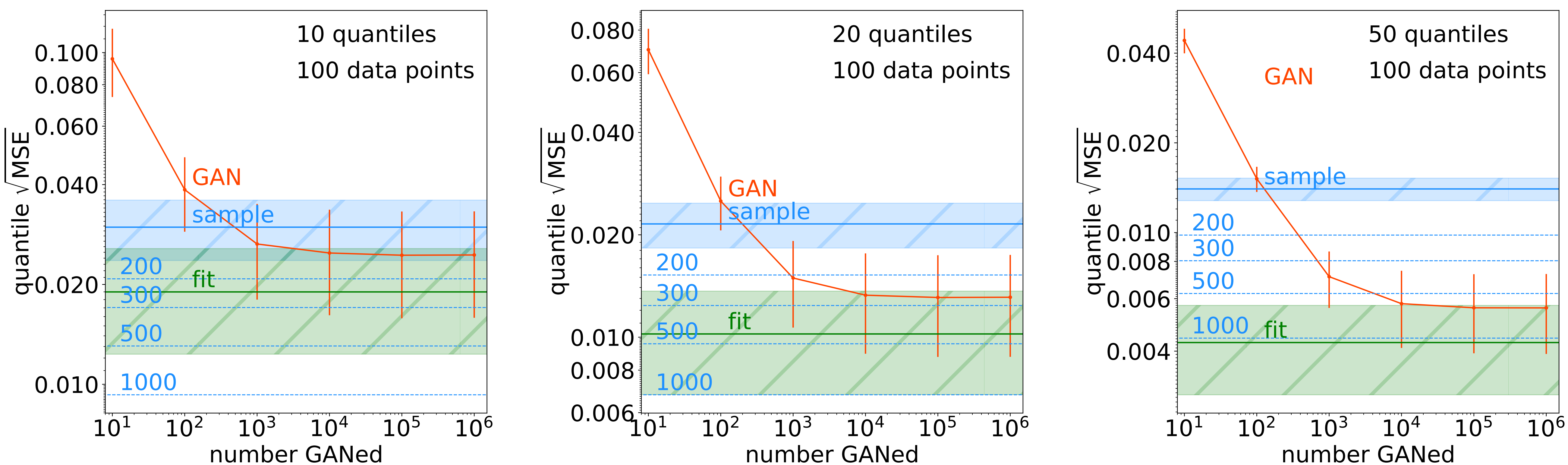
A few words about statistics

2006.06685; 2202.07352

Statistical Amplification

Performance continues to improve on many fronts. As we integrate these tools into our workflows, we need to think about uncertainties.

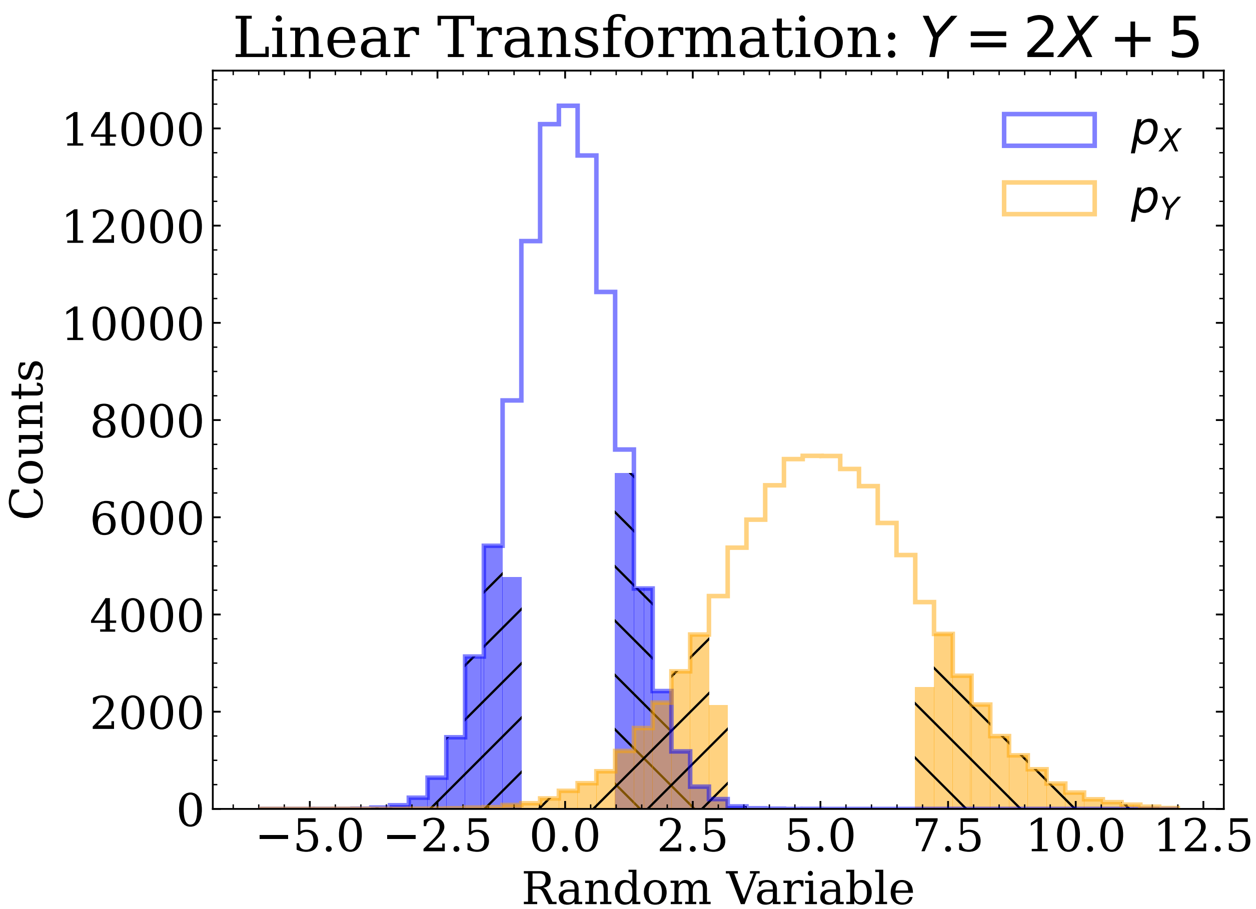
One question is about the **statistical power** of samples from a generative model. This depends on the implicit or explicit information we encode in the networks.

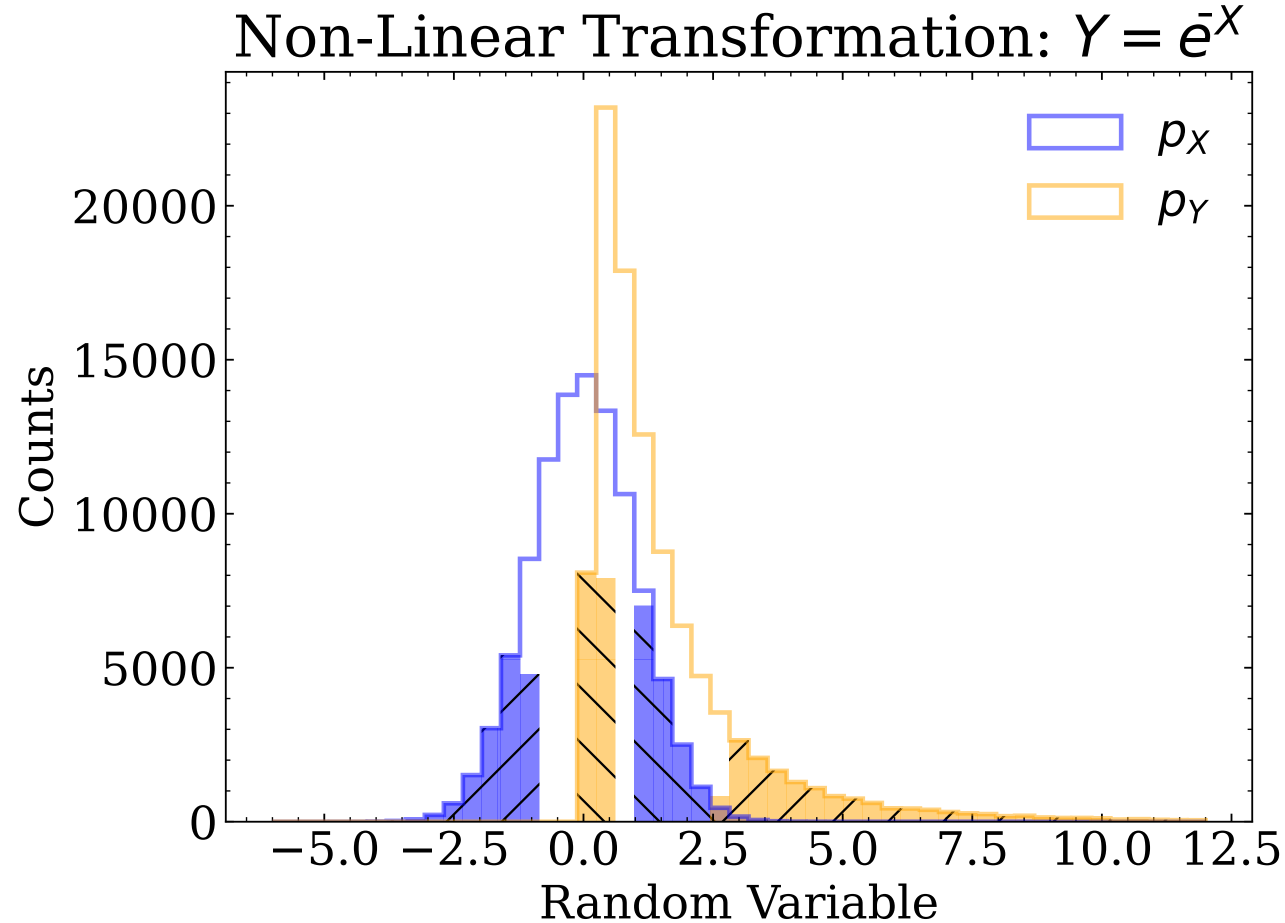


See also 1909.03081, 2002.06307, 2104.04543 (Generative Bayesian NNs), and 2107.08979 (“resampling”)

Impact of (re)parameterization

Generative models are NOT invariant under coordinate transformations. Choose your coordinates wisely!



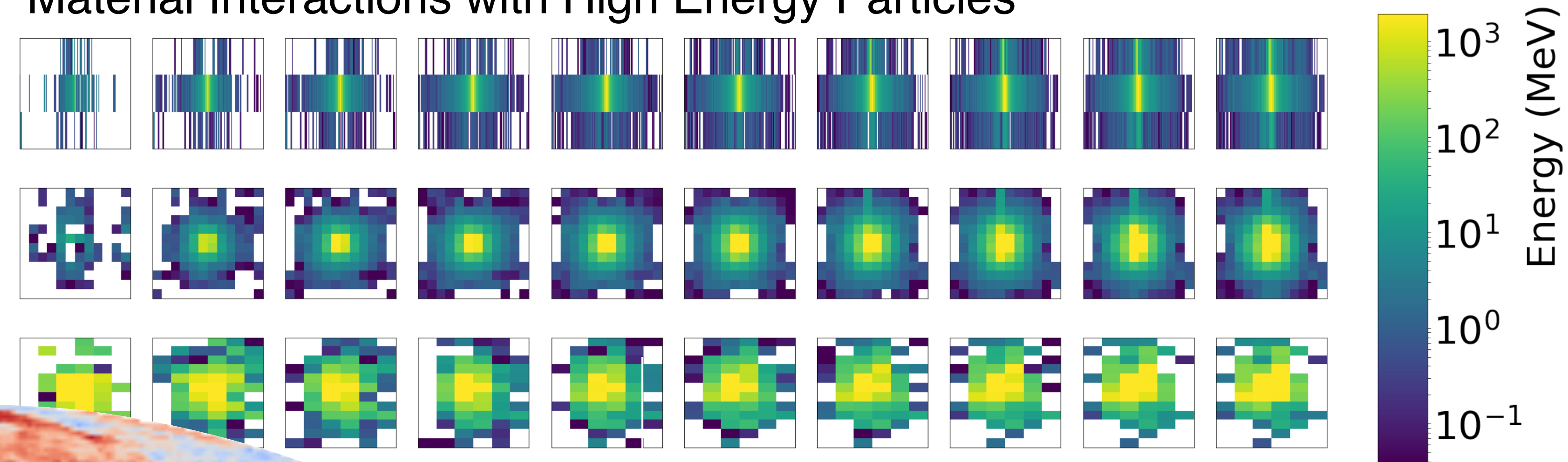


Applications

Generative Models for Particle/Nuclear/Astro

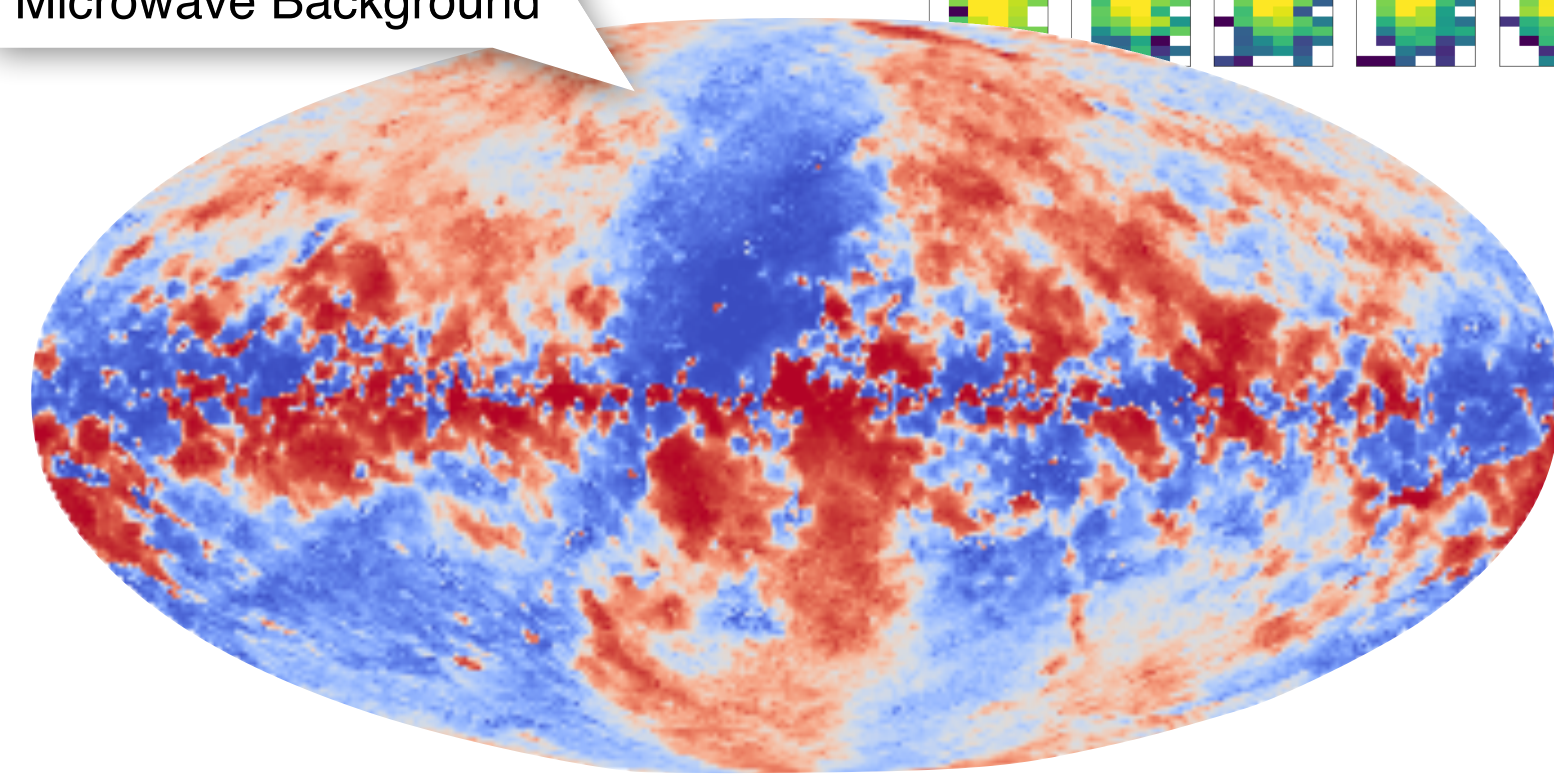
All of these pictures are fake!

Material Interactions with High Energy Particles



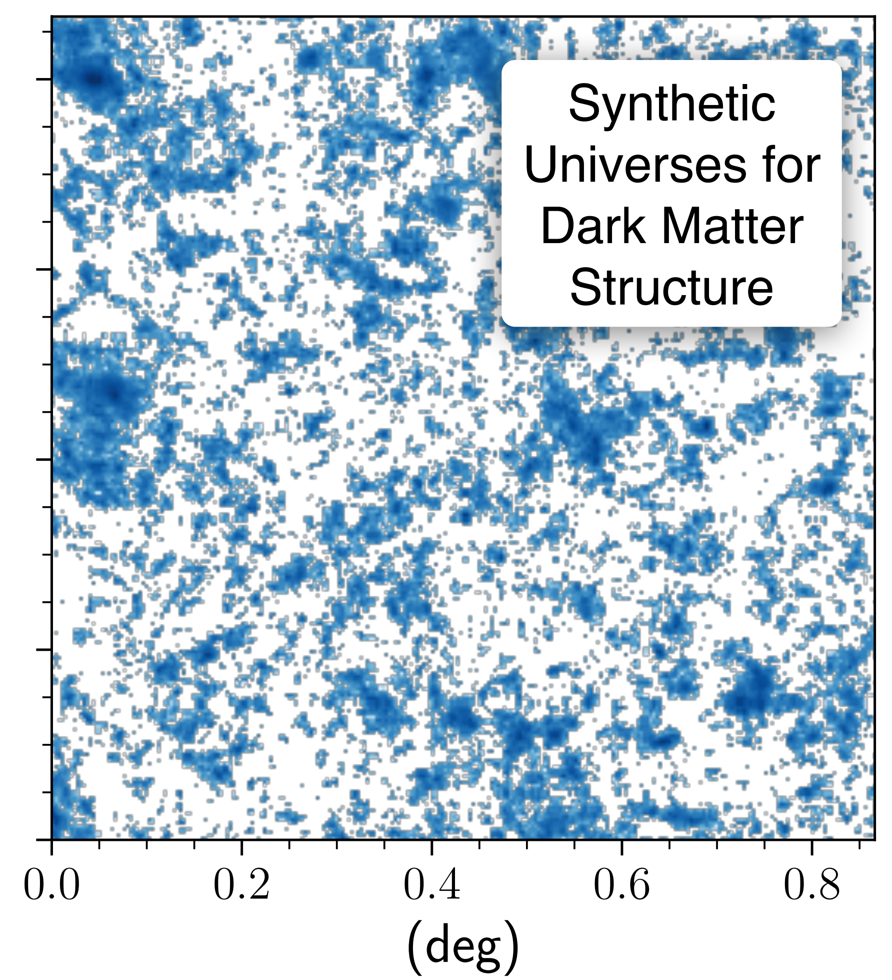
M. Paganini, L. De Oliveira, B. Nachman, Phys. Rev. Lett. 120 (2018) 042003

Synthetic Galactic radiation for Cosmic Microwave Background



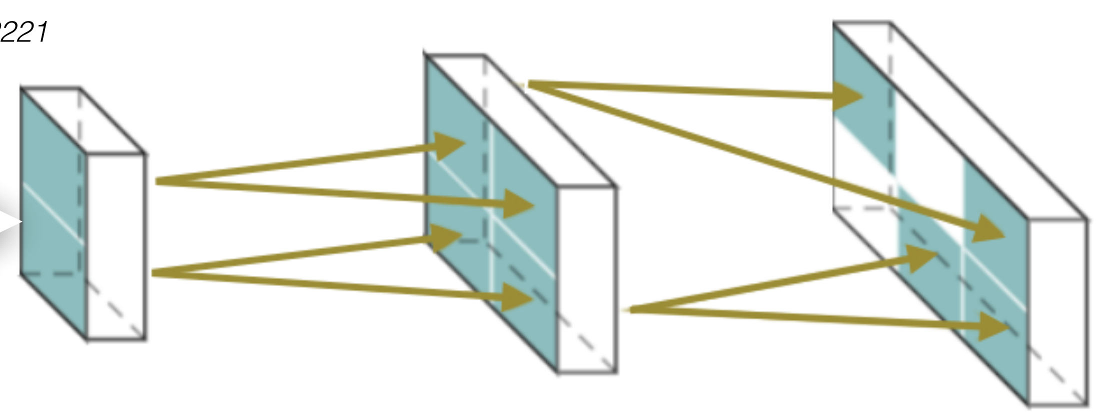
N. Krachmalnicoff and G. Puglisi, arXiv:2011.02221

Synthetic Universes for Dark Matter Structure



M. Mustafa, et al., Comp. Astrophysics and Cosmology 6 (2019)

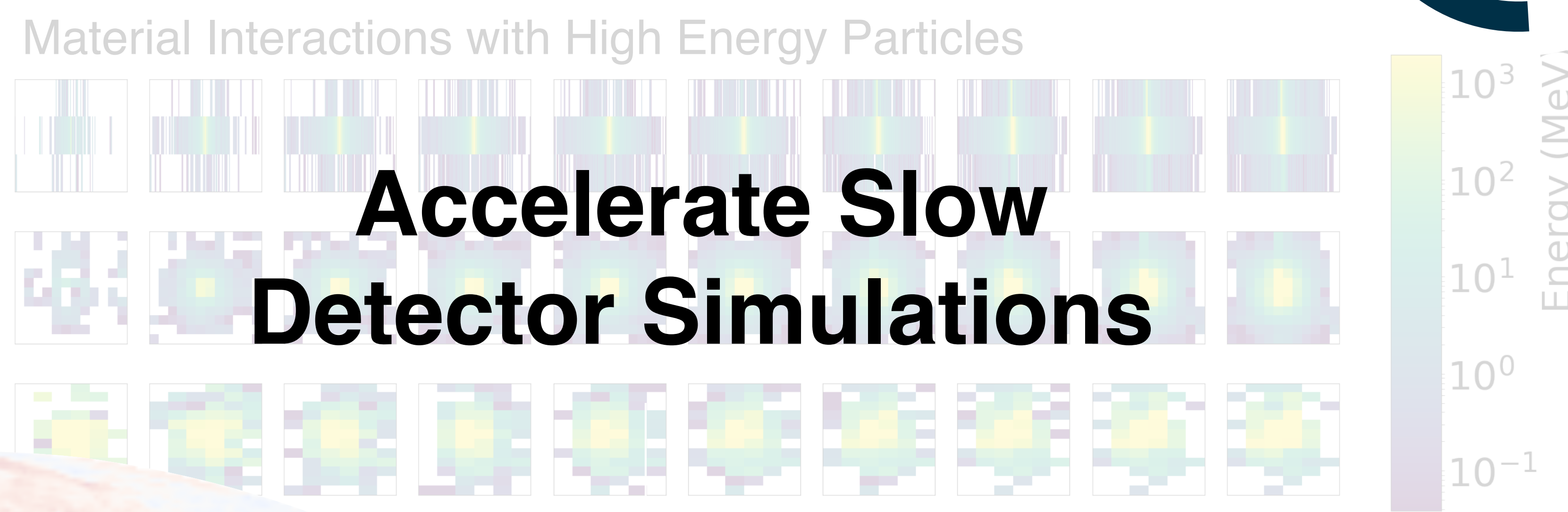
The Structure of Radiation in the Quantum Strong Force



Y. S. Lai, D. Neill, M. Płoskoń, F. Ringer, arXiv:2012.06582

Generative Models for Particle/Nuclear/Astro

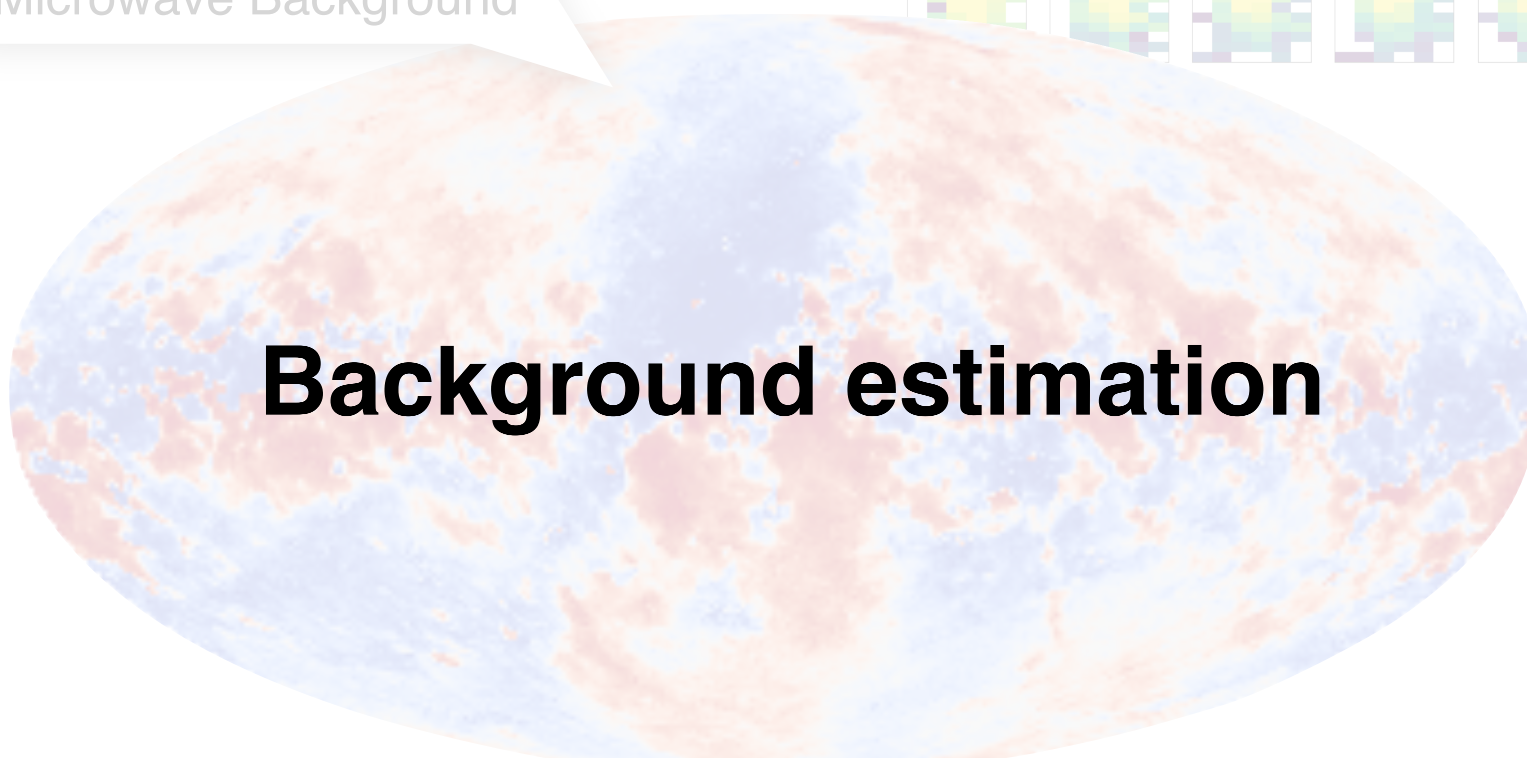
All of these pictures are fake!



Accelerate Slow Detector Simulations

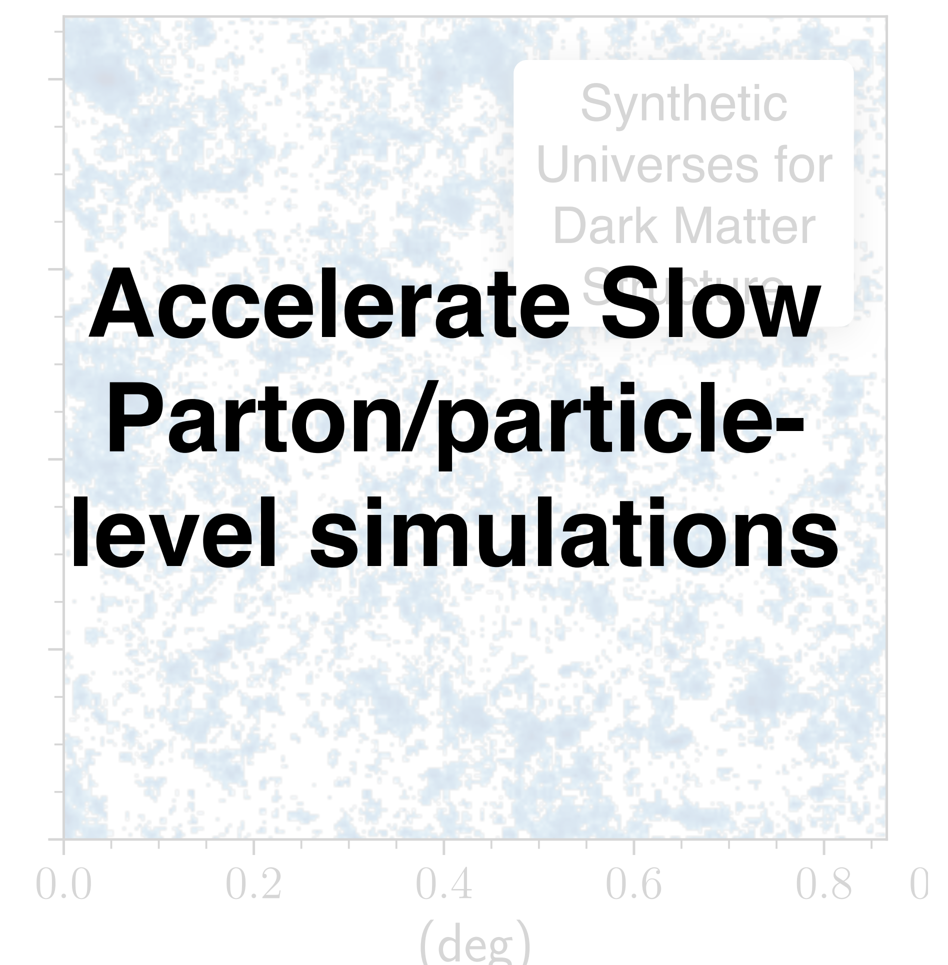
M. Paganini, L. De Oliveira, B. Nachman, Phys. Rev. Lett. 120 (2018) 042003

Synthetic Galactic radiation for Cosmic Microwave Background



Background estimation

N. Krachmalnicoff and G. Puglisi, arXiv:2011.02221

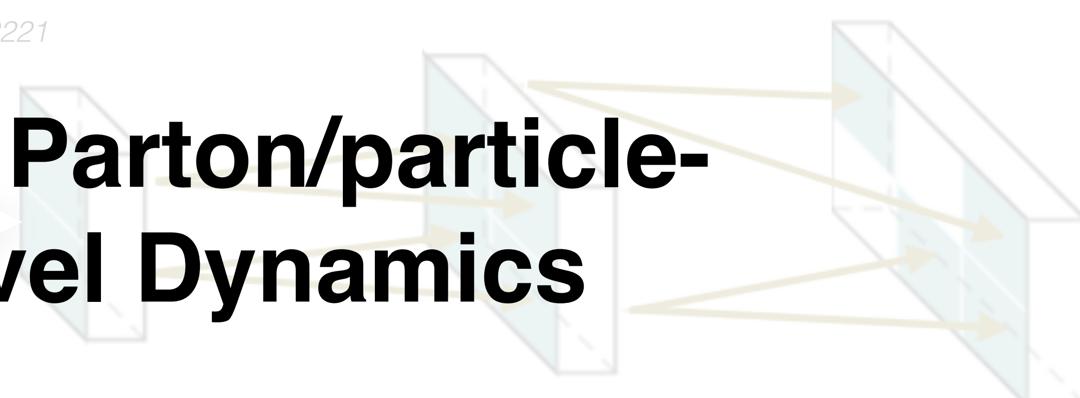


Synthetic Universes for Dark Matter Structure

Accelerate Slow Parton/particle-level simulations

M. Mustafa, et al., Comp. Astrophysics and Cosmology 6 (2019)

The Structure of Radiation in the Quantum Strong Force

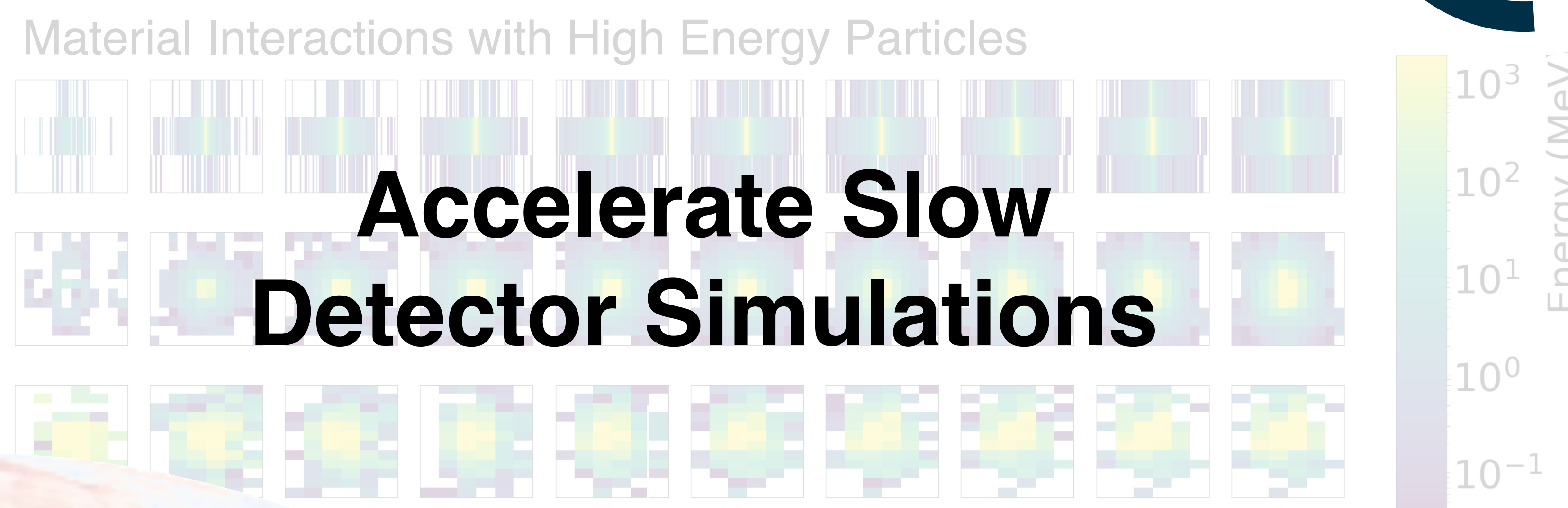


Infer Parton/particle-level Dynamics

Y. S. Lai, D. Neill, M. Płoskoń, F. Ringer, arXiv:2012.06582

Generative Models for Particle/Nuclear/Astro

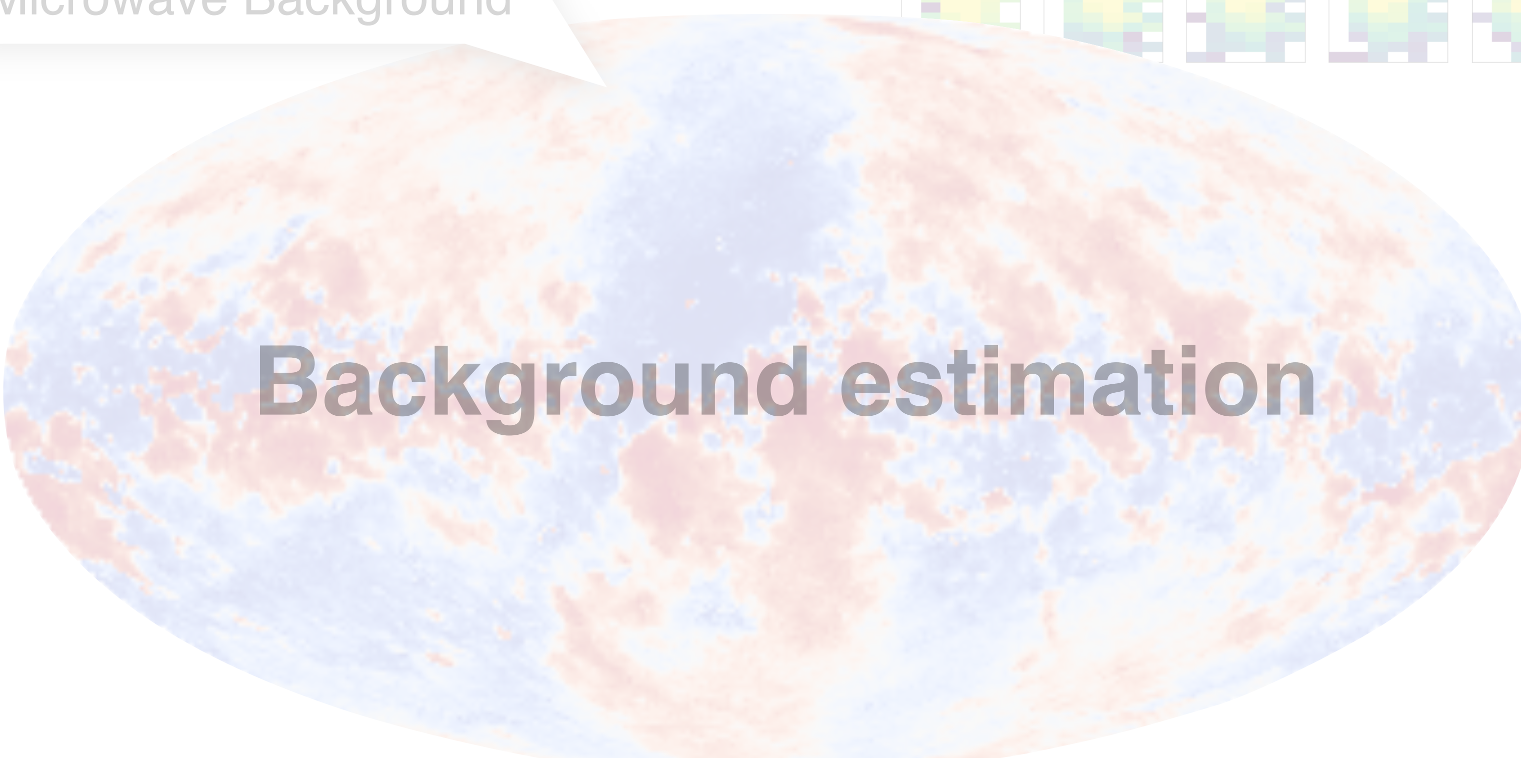
All of these pictures are fake!



Accelerate Slow Detector Simulations

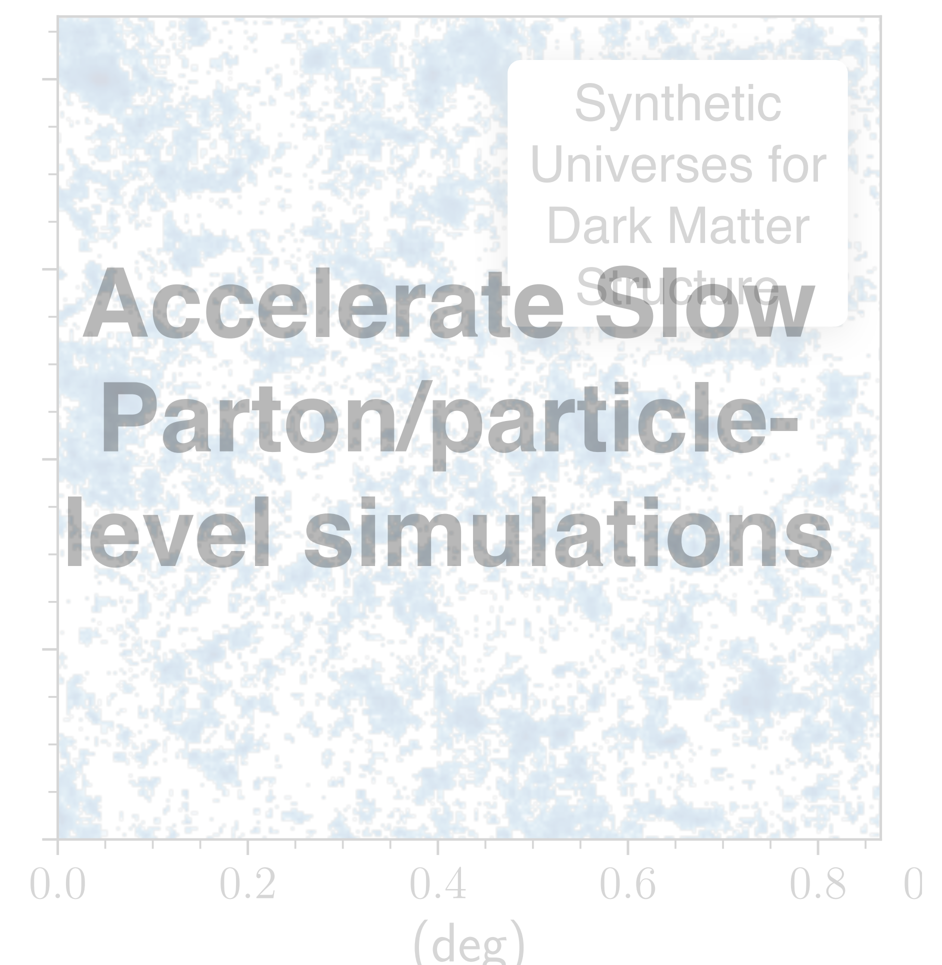
M. Paganini, L. De Oliveira, B. Nachman, Phys. Rev. Lett. 120 (2018) 042003

Synthetic Galactic radiation for Cosmic Microwave Background



Background estimation

N. Krachmalnicoff and G. Puglisi, arXiv:2011.02221

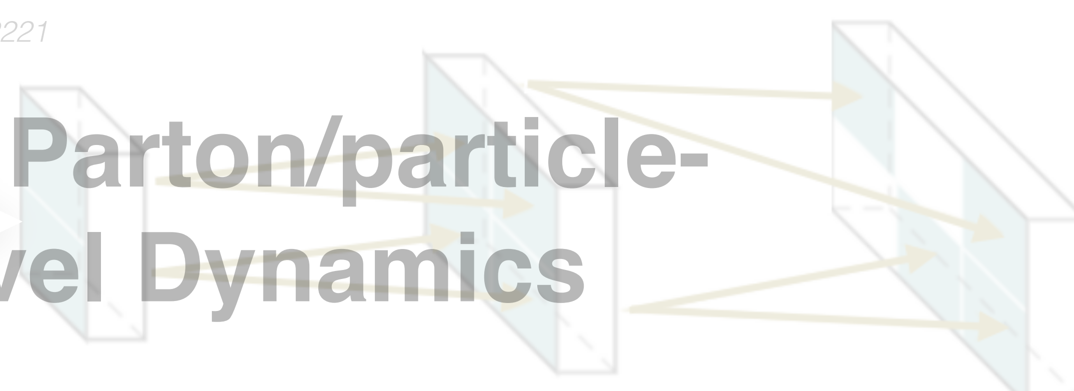


Synthetic Universes for Dark Matter Structure

Accelerate Slow Parton/particle-level simulations

M. Mustafa, et al., Comp. Astrophysics and Cosmology 6 (2019)

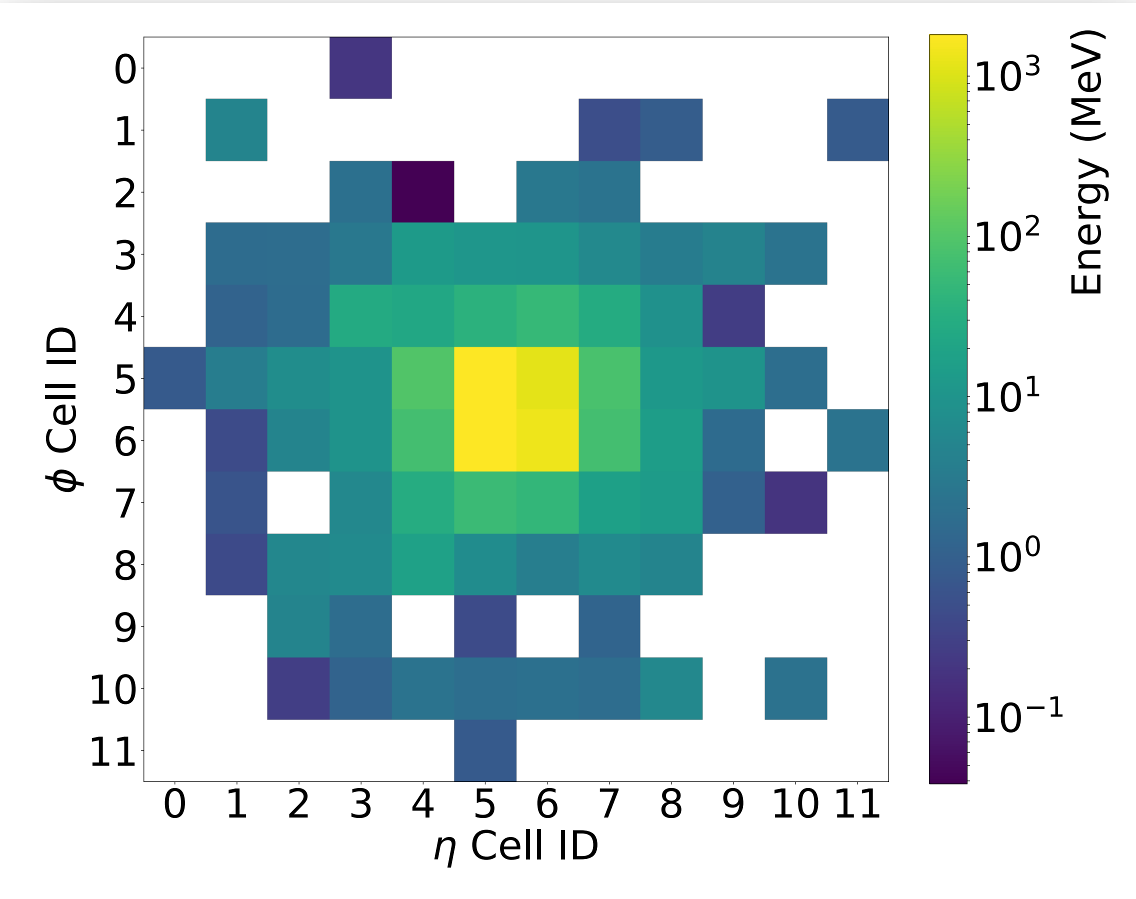
The Structure of Radiation in the Quantum Strong Force



Infer Parton/particle-level Dynamics

Y. S. Lai, D. Neill, M. Płoskoń, F. Ringer, arXiv:2012.06582

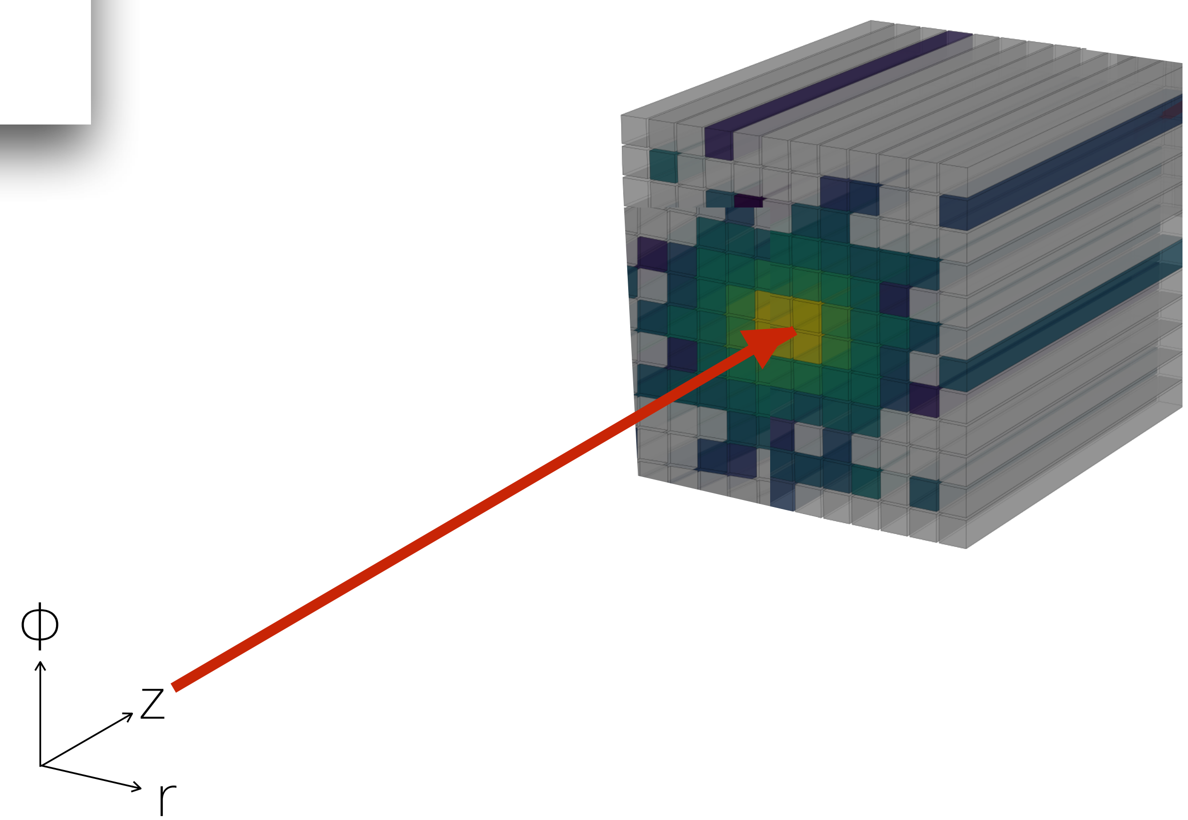
Accelerating Detector Simulations



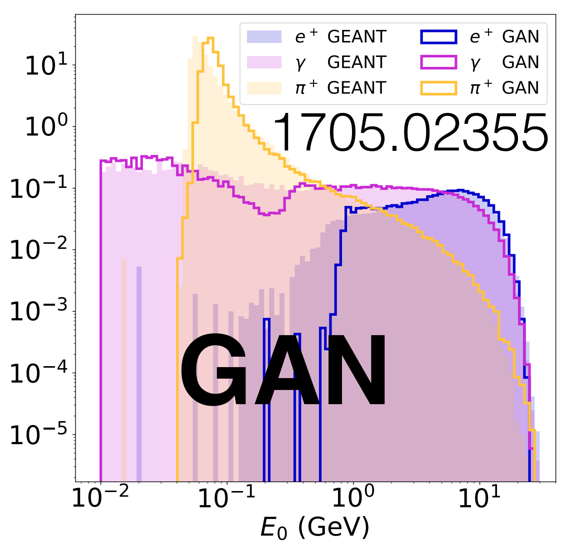
Calorimeters are often the slowest to simulate

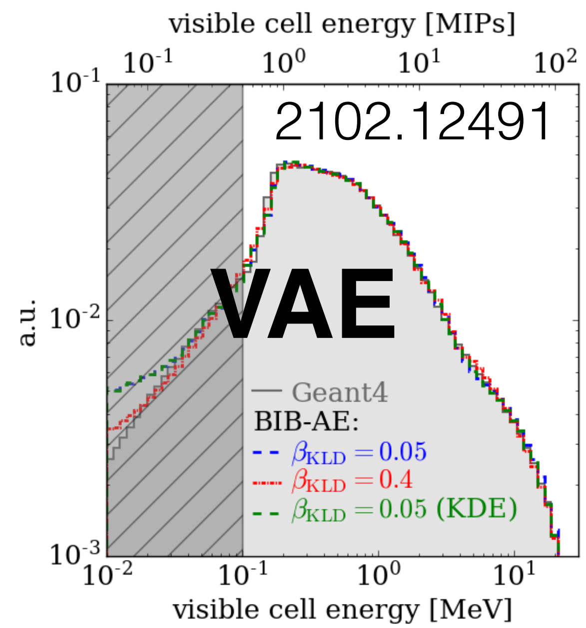
stopping particles requires simulating interactions of all energies

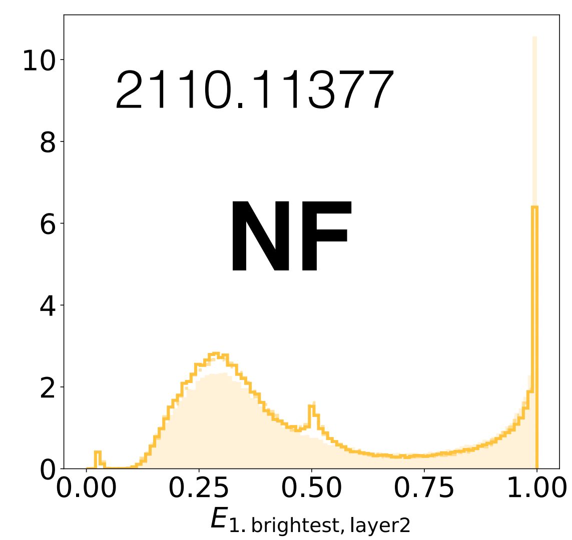
Grayscale images: Pixel intensity = energy deposited



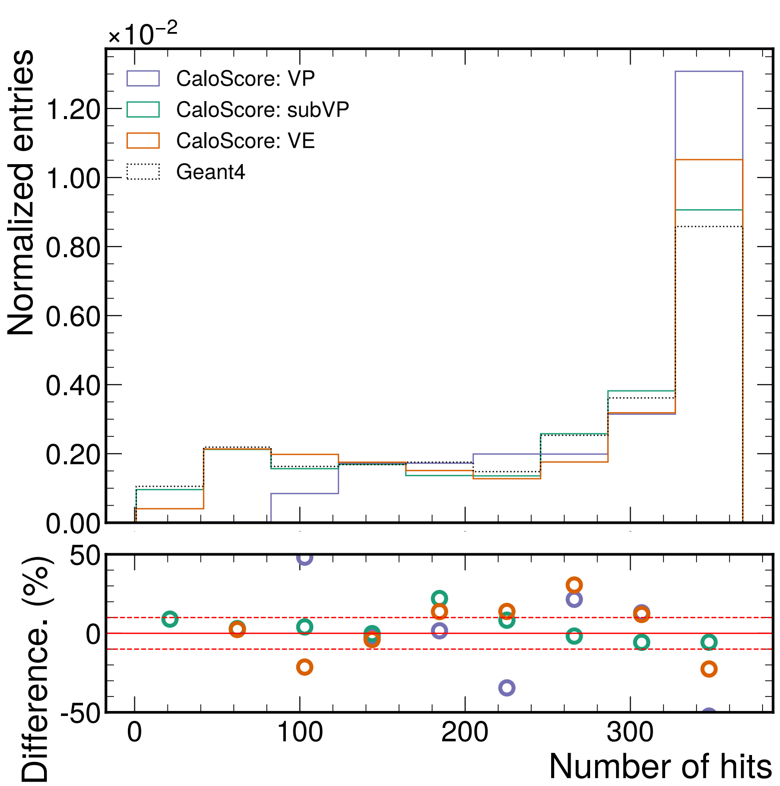
Calorimeter ML Surrogate Models

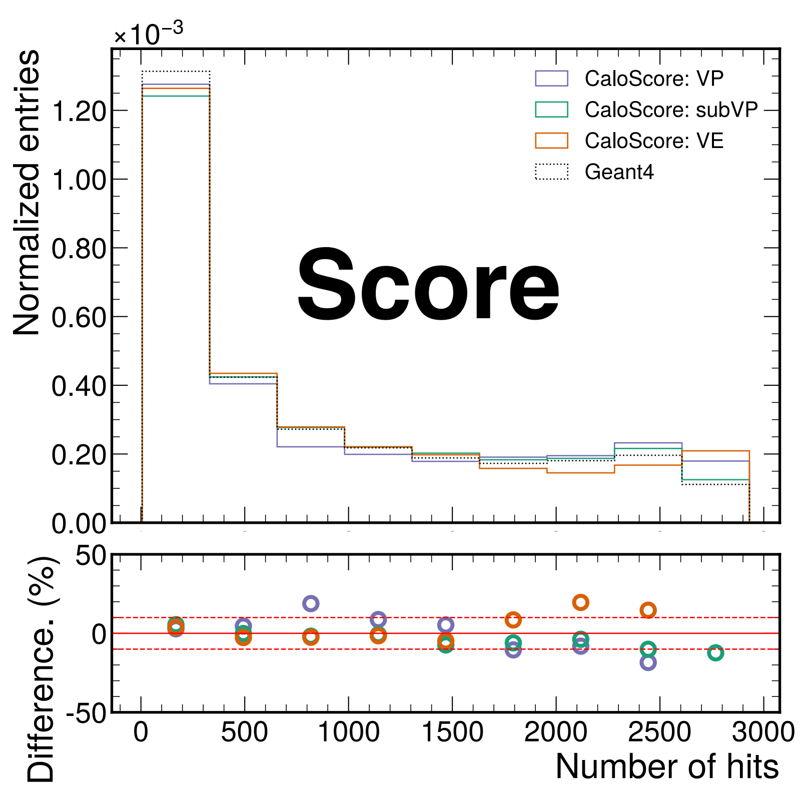


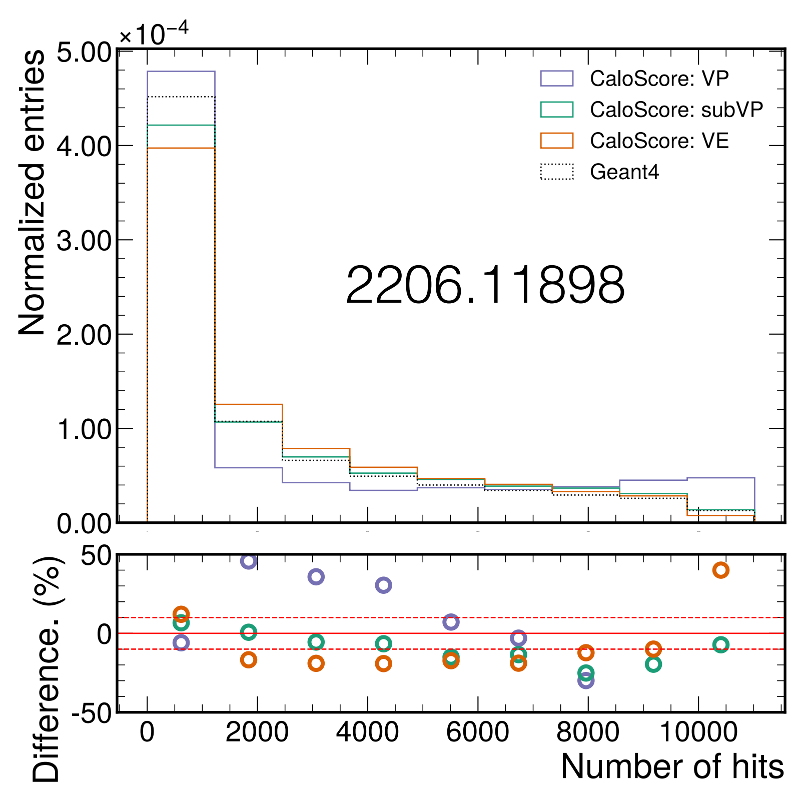




Many papers on this subject - see the living review for all







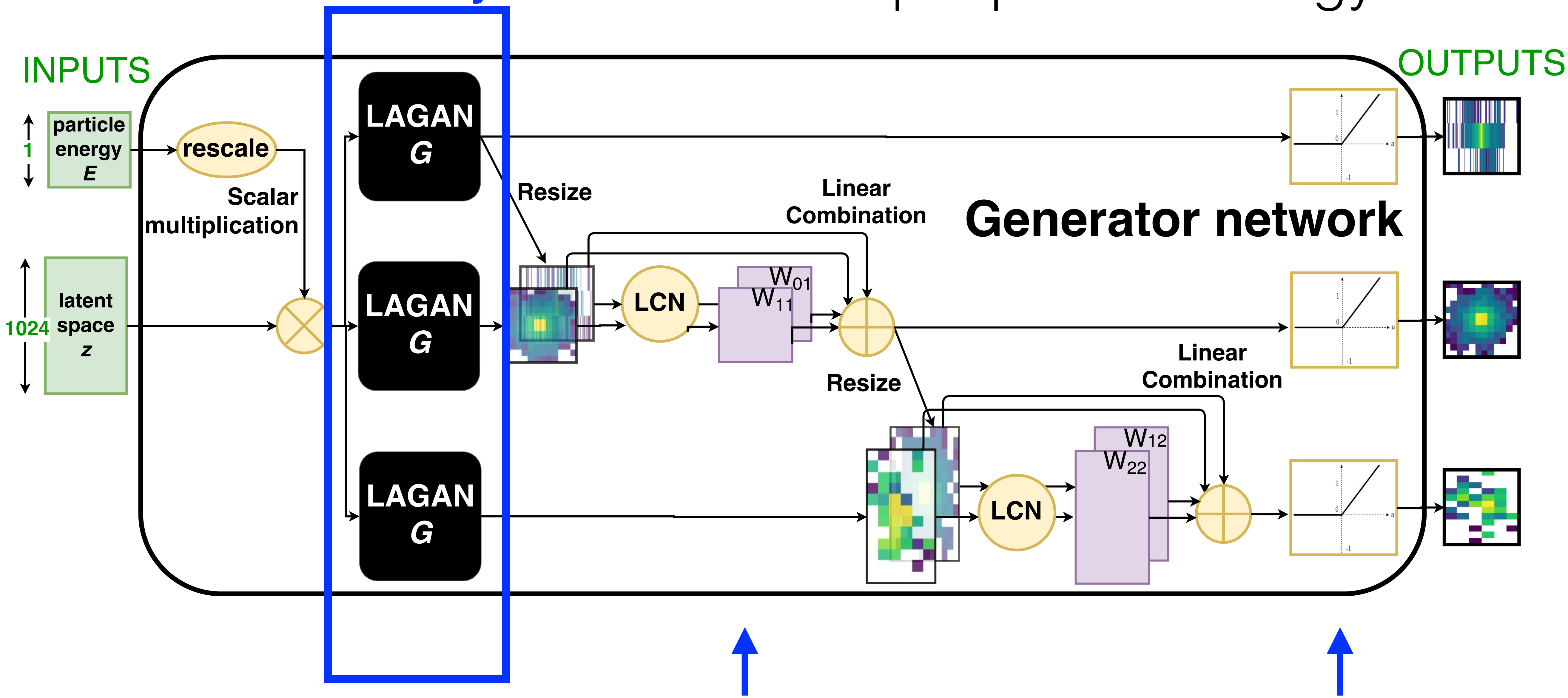
See also https://calochallenge.github.io/homepage/

1705.02355

Introducing CaloGAN

One image per calo layer

One network per particle type; input particle energy



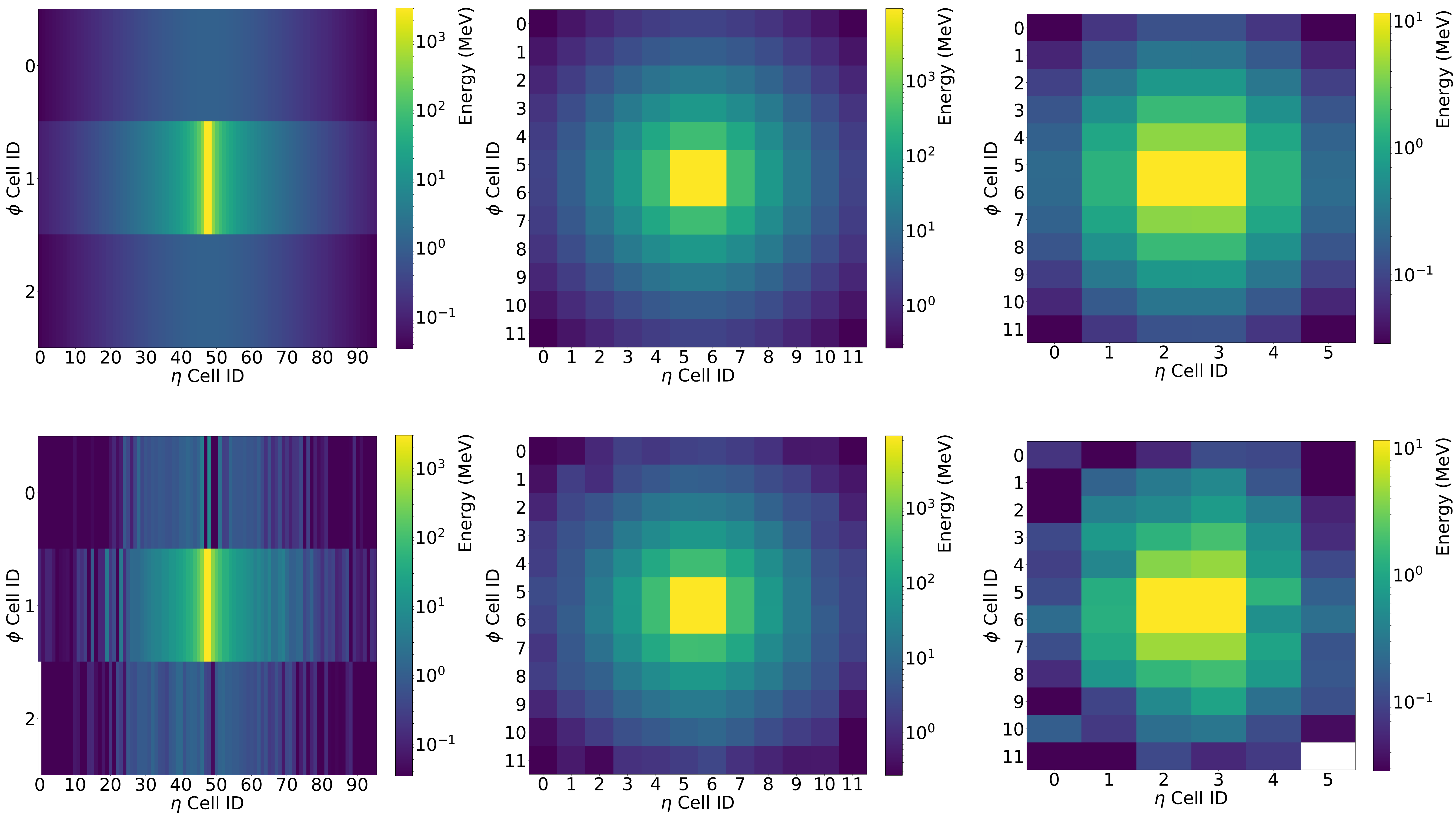
LA = Locally Aware, like a CNN

use layer i as input to layer i+1

ReLU to encourage sparsity

Performance: average images

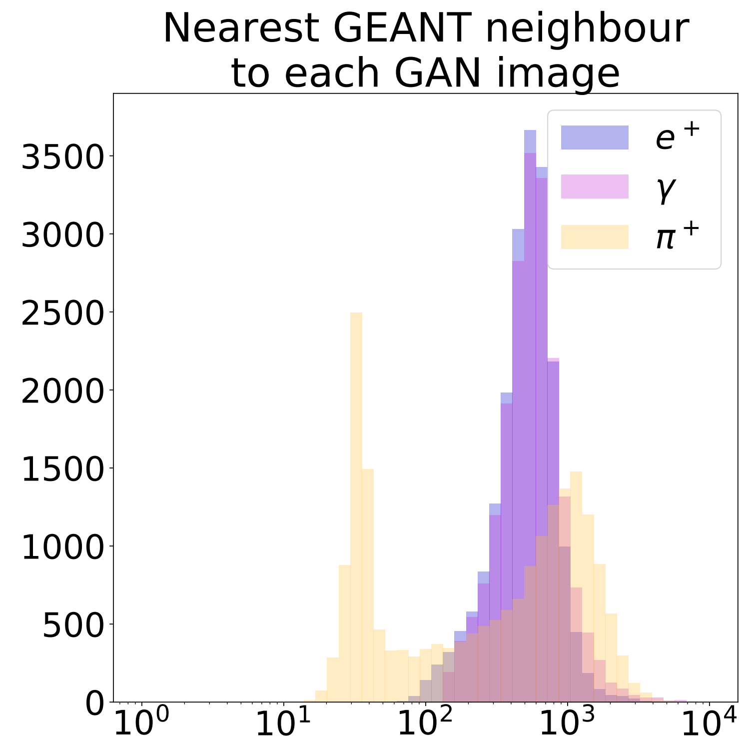
Geant4



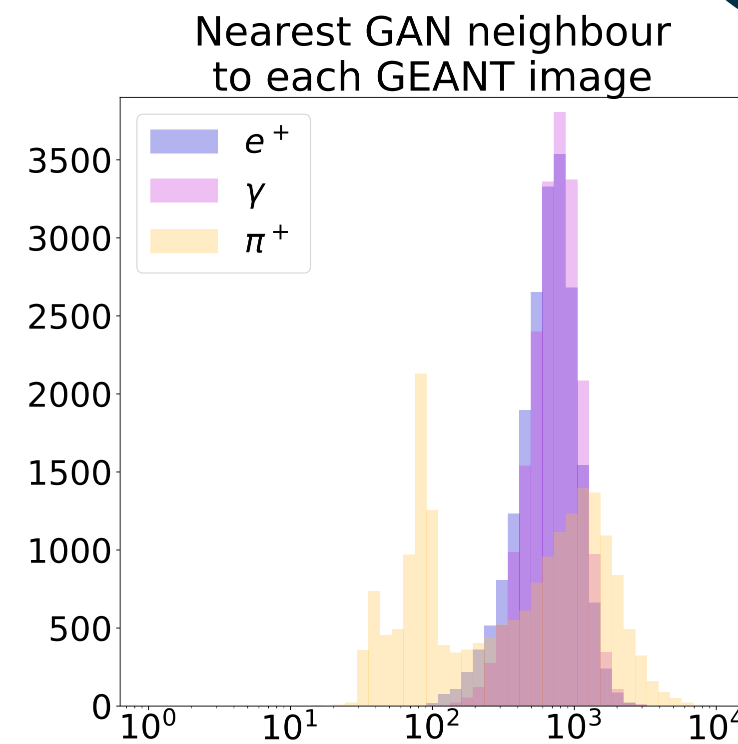
CaloGAN

1705.02355

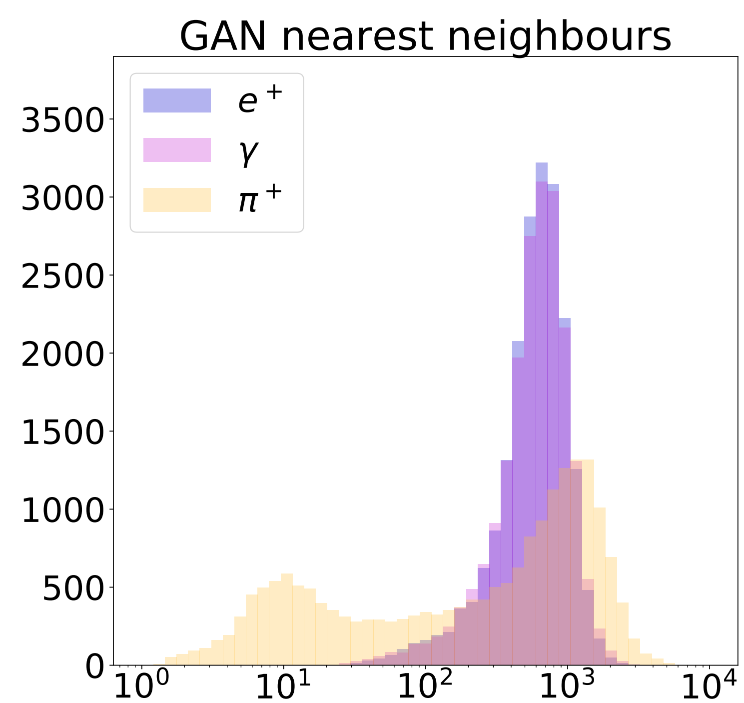
Test for “overtraining”



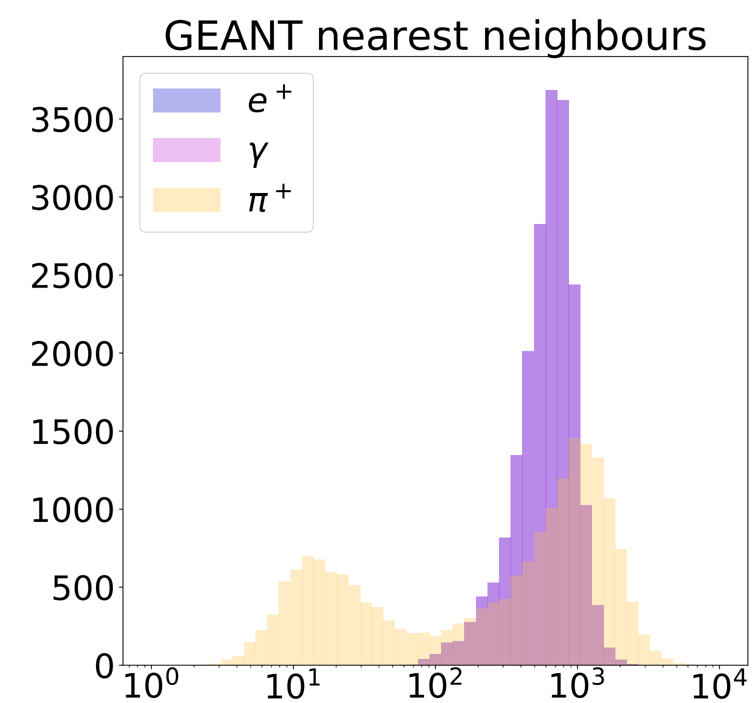
not memorizing



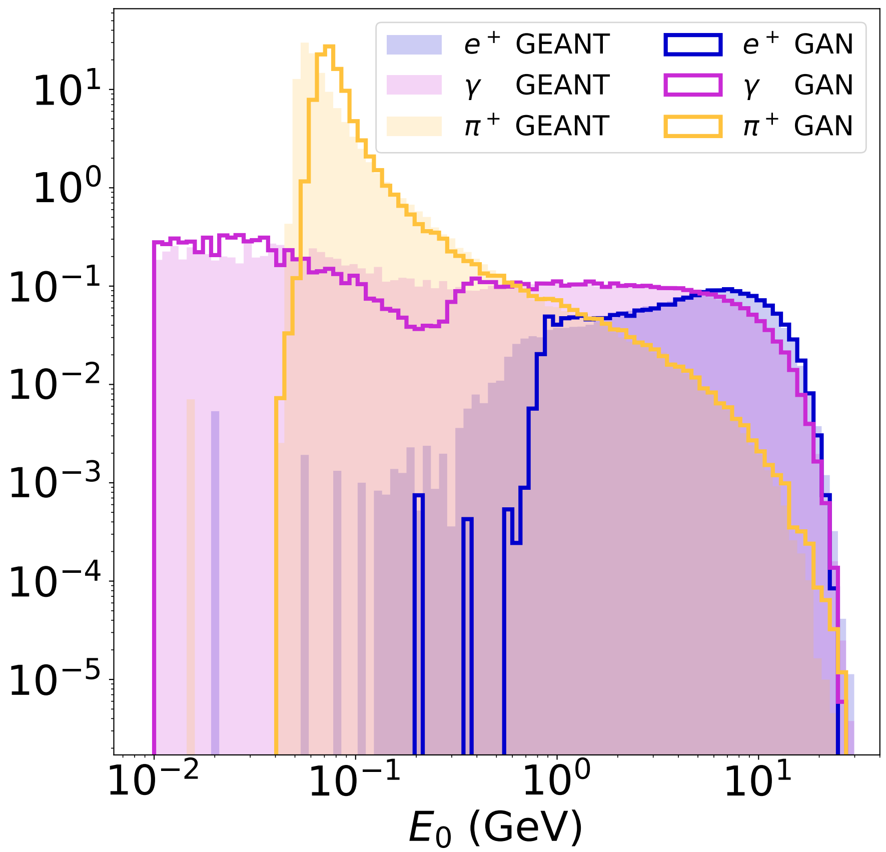
A key challenge in training GANs is the diversity of generated images. This does not seem to be a problem for CaloGAN.



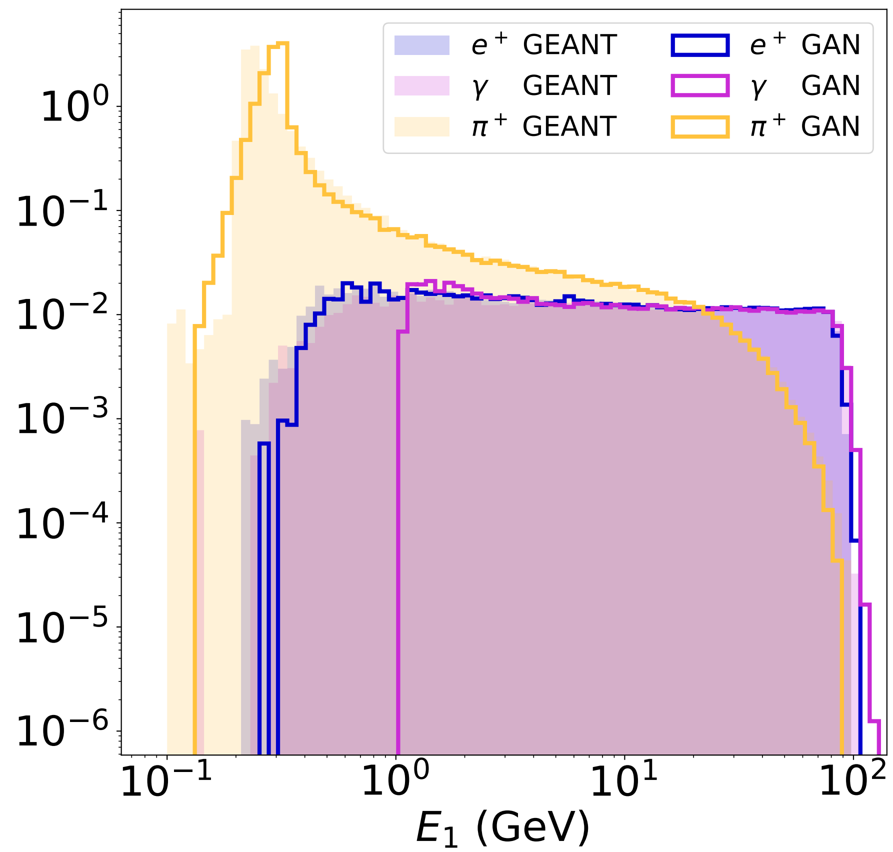
no mode collapse

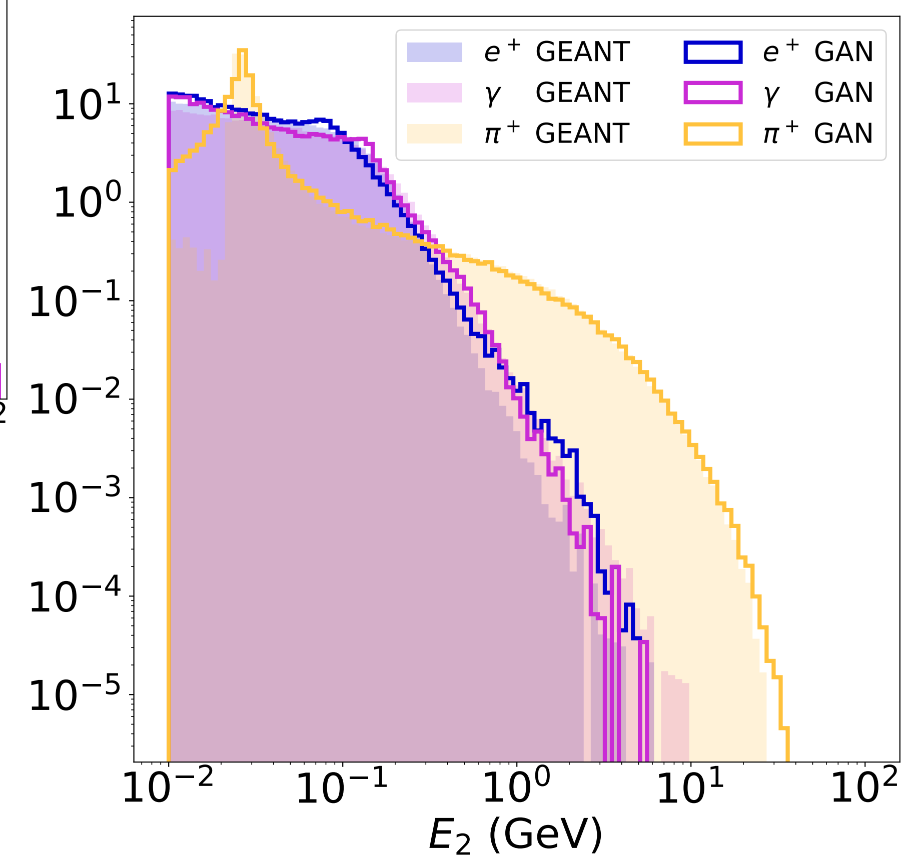


Performance: energy per layer



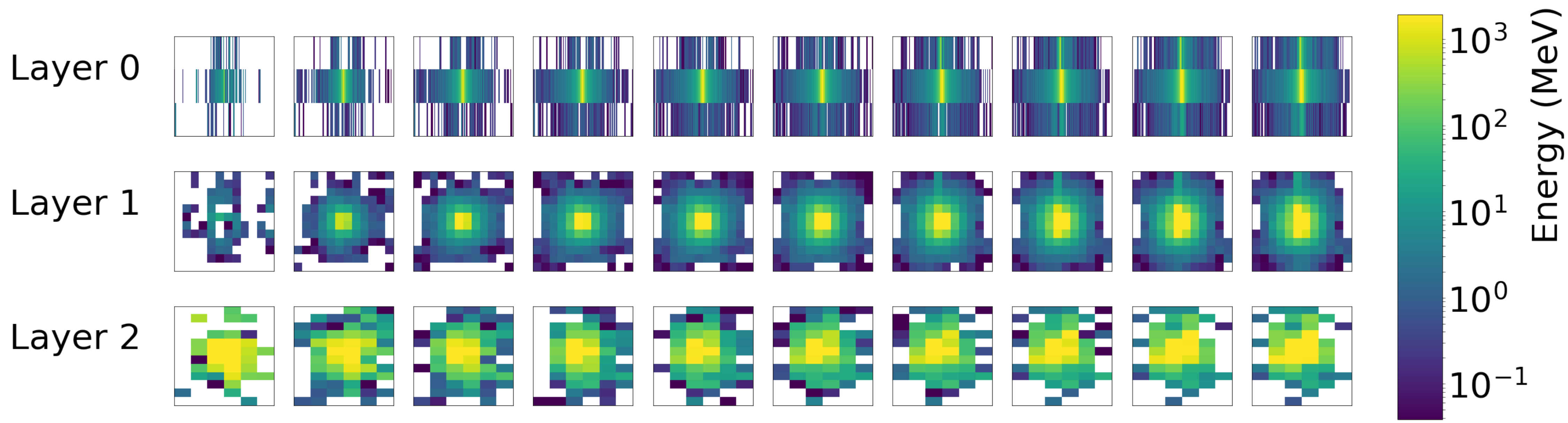
Pions deposit much less energy in the first layers; leave the calorimeter with significant energy



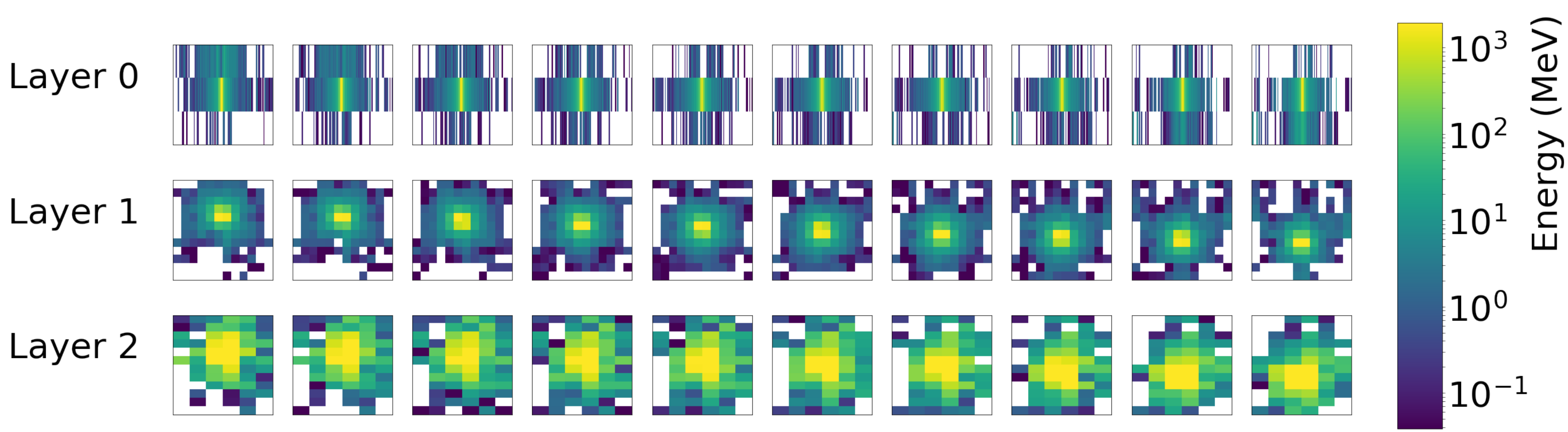


Conditioning

Fix noise, scan latent variable corresponding to energy

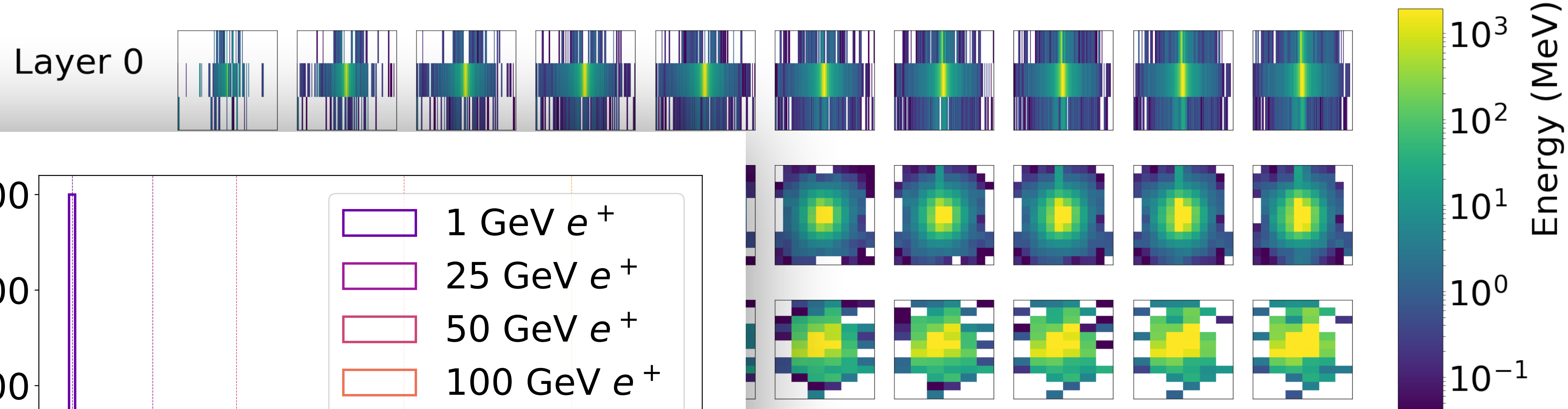


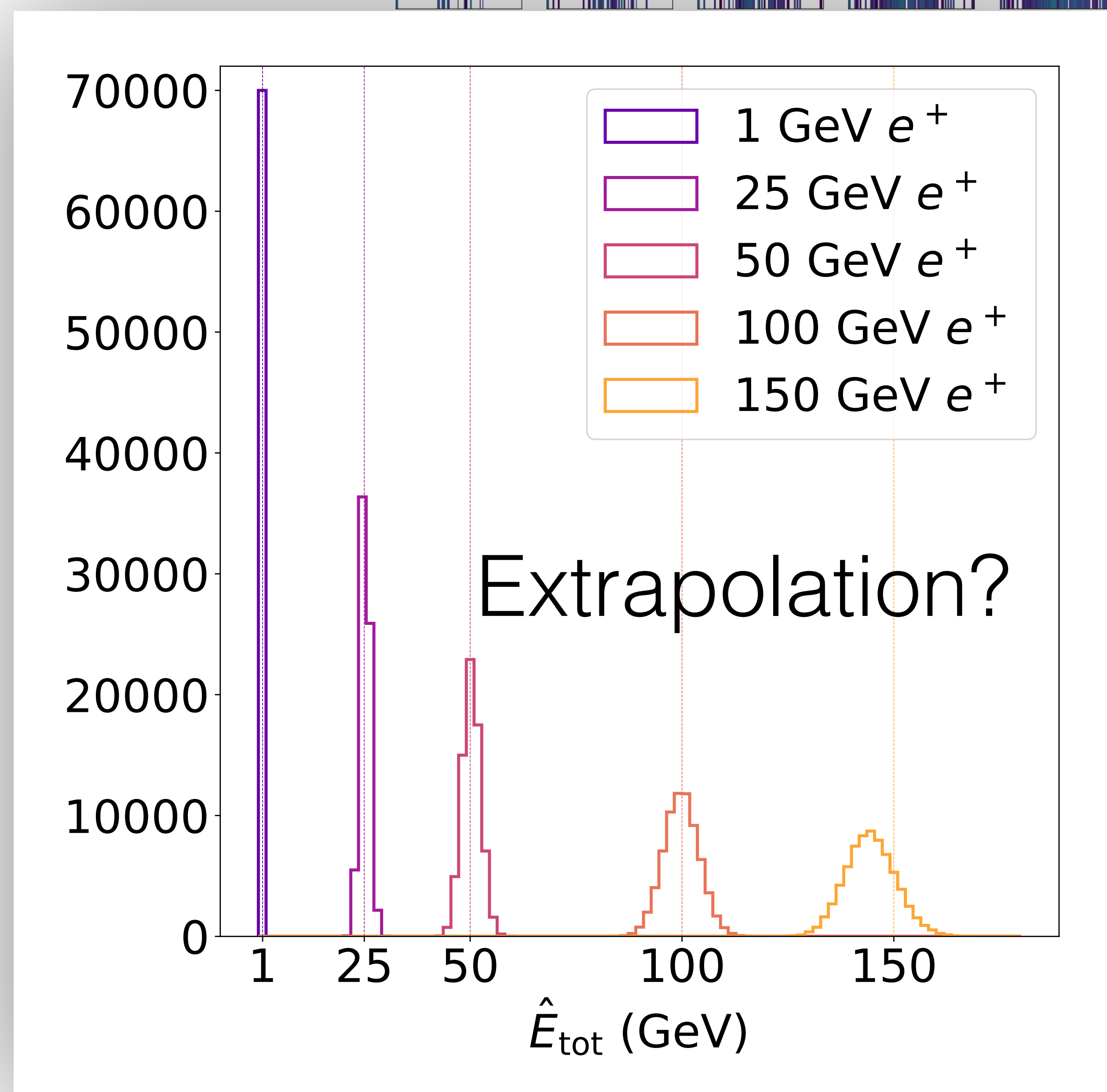
Fix noise, scan latent variable corresponding to x-position



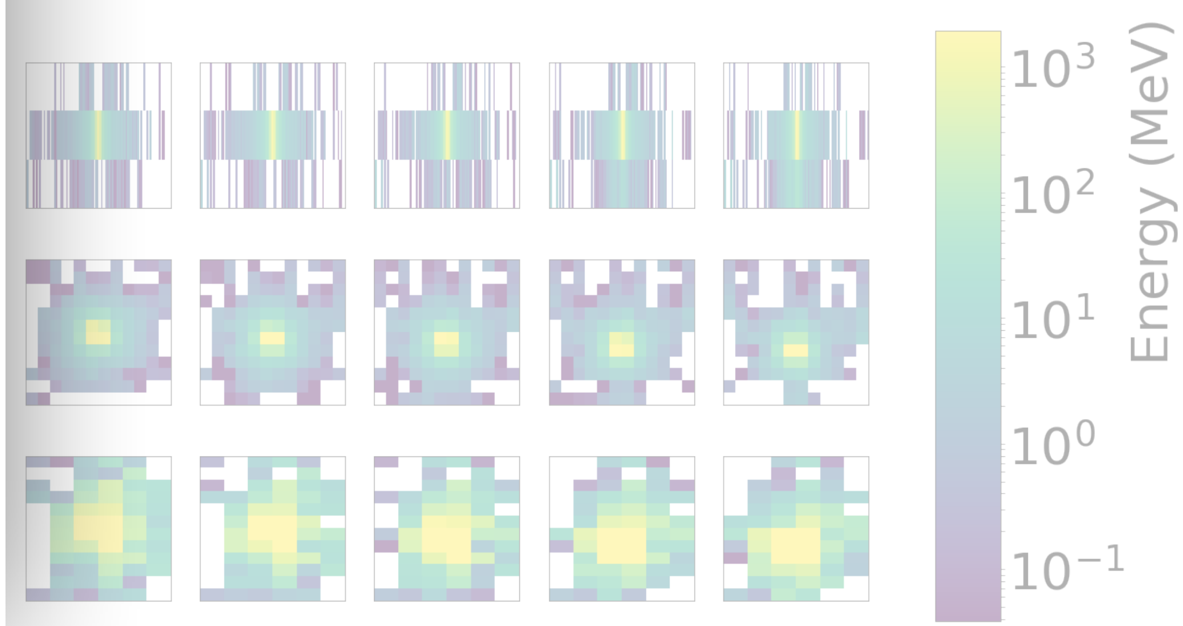
Conditioning

Fix noise, scan latent variable corresponding to energy





ble corresponding to x-position



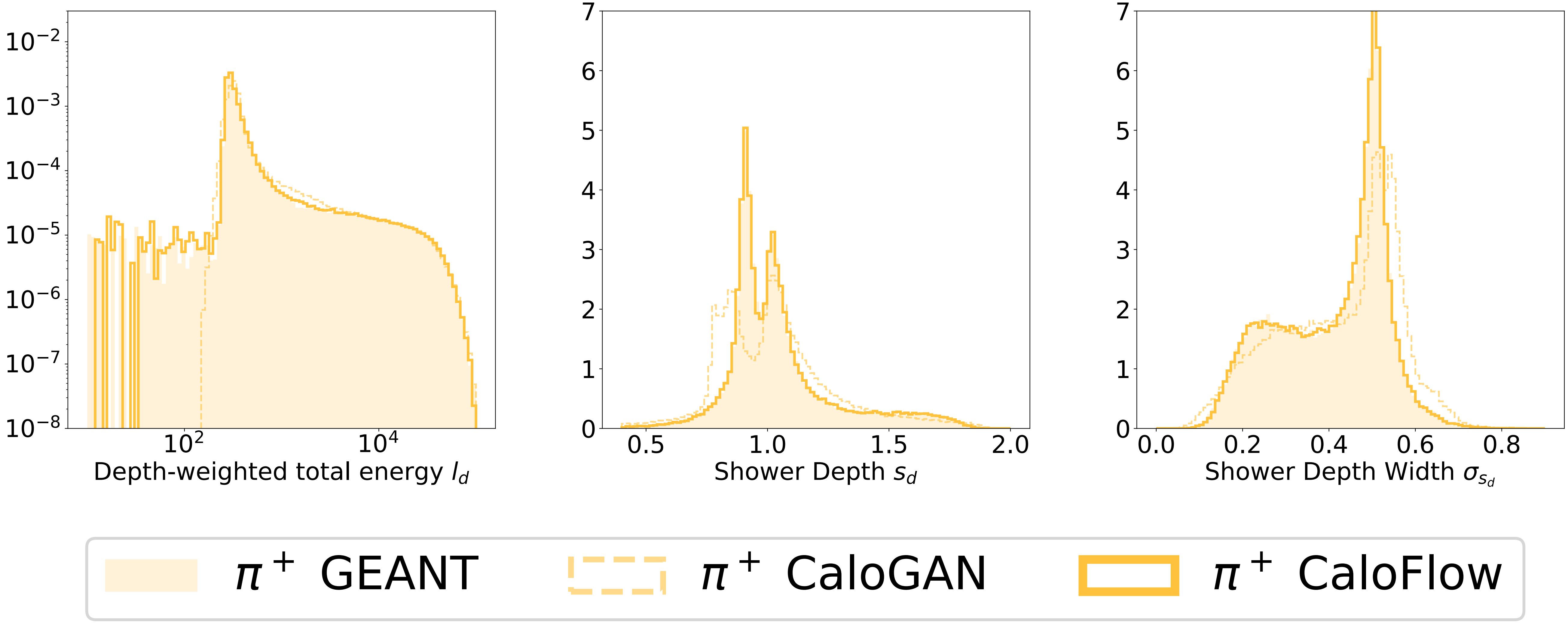
Timing

Generation Method	Hardware	Batch Size	milliseconds/shower
GEANT4	CPU	N/A	1772 ←
CALOGAN	CPU *Intel Xeon E5-2670*	1	13.1
		10	5.11
		128	2.19
		1024	2.03
	GPU *NVIDIA K80*	1	14.5
		4	3.68
		128	0.021
		512	0.014
		1024	0.012 ←

(clearly these numbers have changed as both technologies have improved - this is simply meant to be qualitative & motivating!)

Current State of the art

Generative models have gotten much better; **flow models** are particularly promising. Added bonus: have an explicit density.



See also score-based: 2206.11898

Current State of the art

Generative models have gotten much better: **flow models** are

AUC / JSD		DNN vs. CaloGAN	DNN vs. CaloFlow
e^+	unnormalized	1.000(0) / 0.993(1)	0.847(8) / 0.345(12)
	normalized	1.000(0) / 0.997(0)	0.869(2) / 0.376(4)
γ	unnormalized	1.000(0) / 0.996(1)	0.660(6) / 0.067(4)
	normalized	1.000(0) / 0.994(1)	0.794(4) / 0.213(7)
π^+	unnormalized	1.000(0) / 0.988(1)	0.632(2) / 0.048(1)
	normalized	1.000(0) / 0.997(0)	0.751(4) / 0.148(4)

Output is nearly indistinguishable from Geant4 !

AUC = 1 means easily distinguishable, AUC = 0.5 means not distinguishable

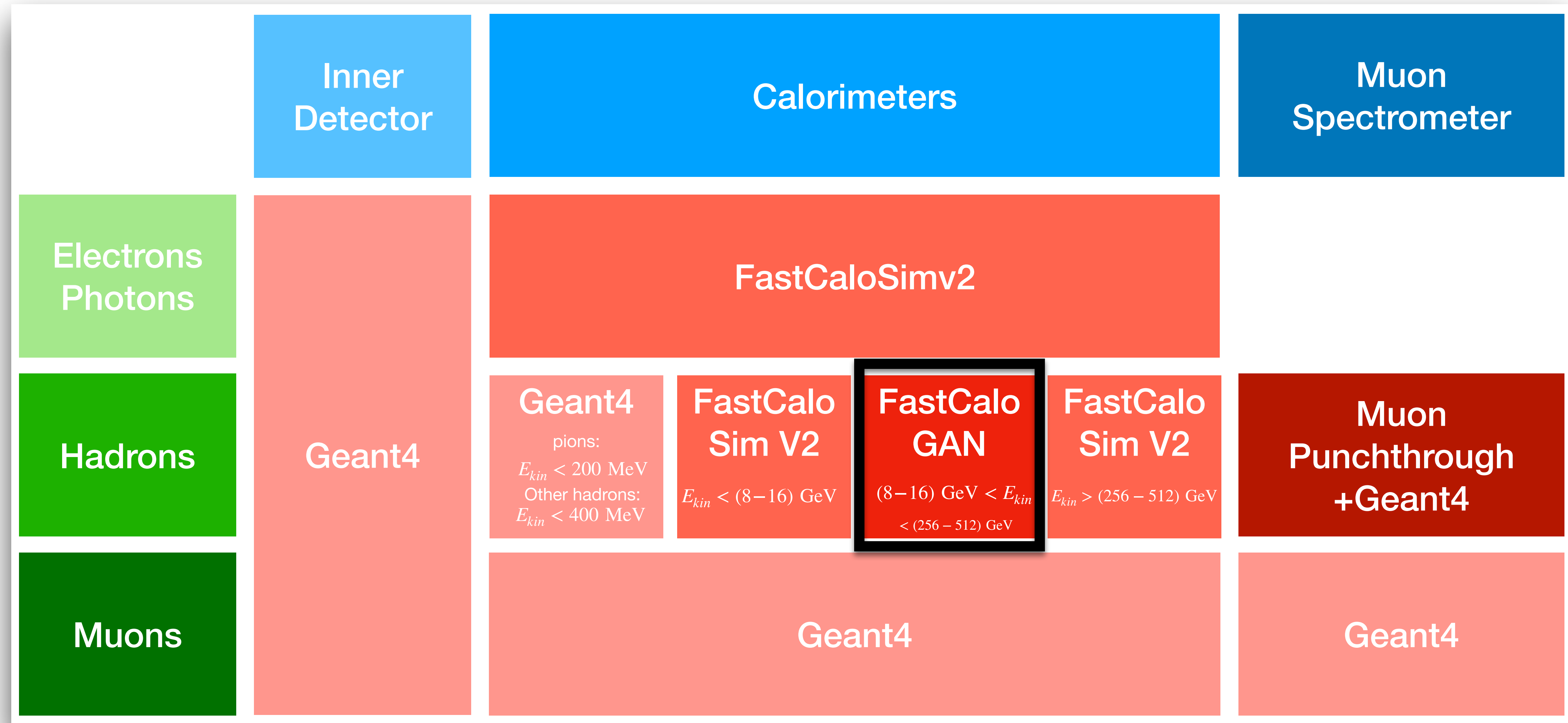
Depth-weighted total energy l_d Shower Depth s_d Shower Depth Width σ_{s_d}

π^+ GEANT π^+ CaloGAN π^+ CaloFlow

How to compare generative models? See e.g. 2211.10295

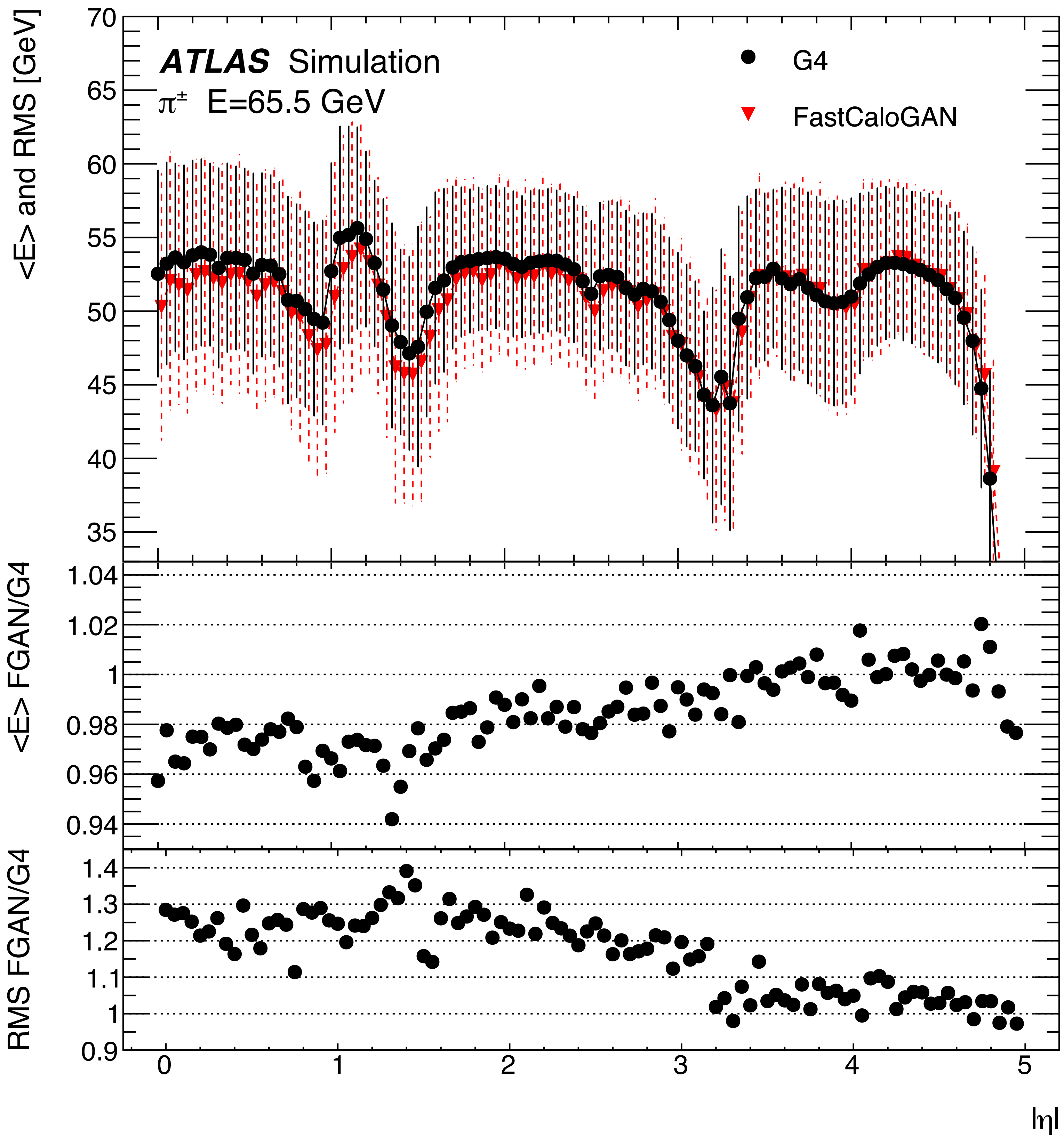
See also score-based: 2206.11898

Integration into real detector sim.



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

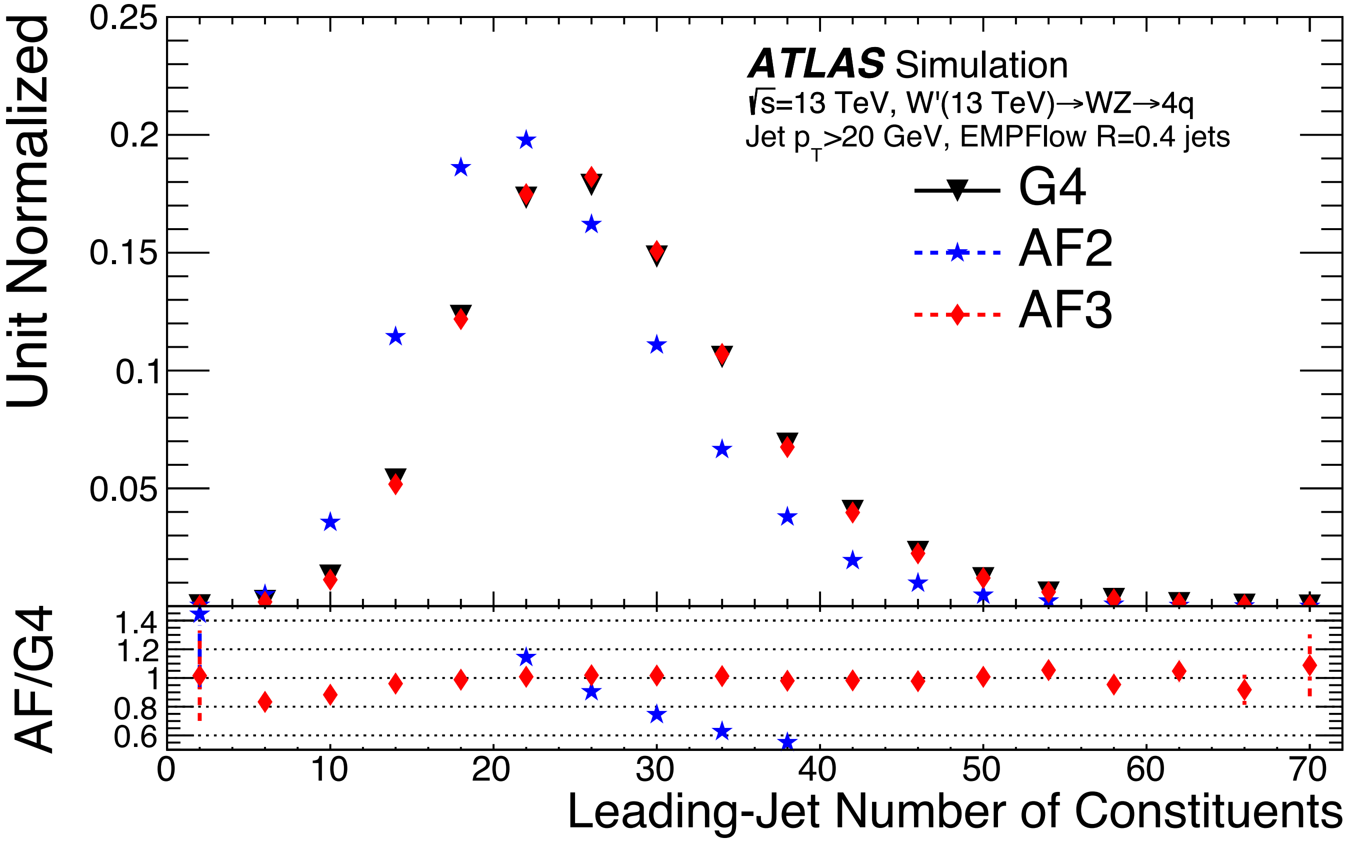
Integration into real detector sim.



The GAN architecture is relatively simple, but it is able to match the energy scale and resolution well.

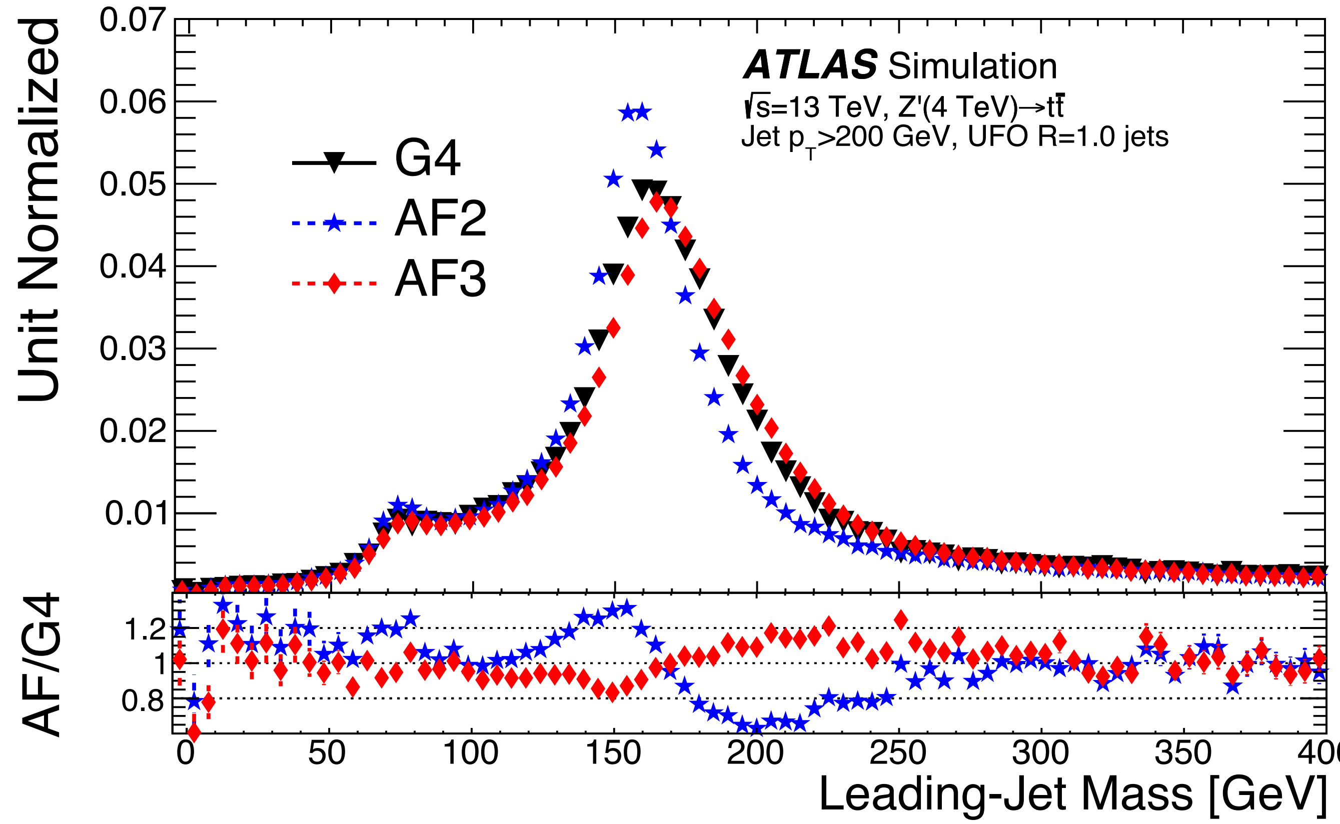
There is one GAN per η slice

Integration into real detector sim.

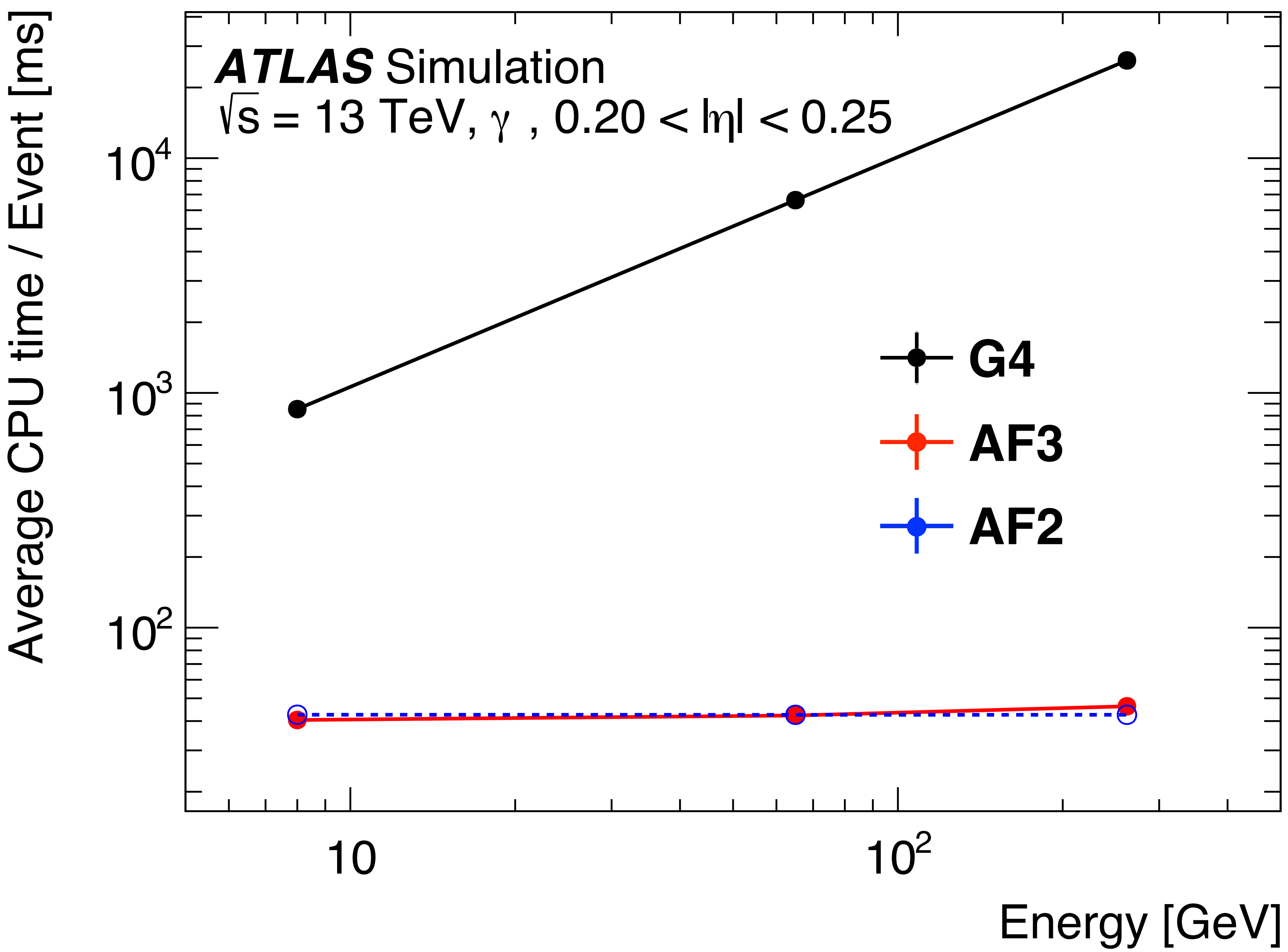


The new fast simulation (**AF3**) significantly improves jet substructure with respect to the older one (**AF2**)

Ideally, the same calibrations derived for full sim. (Geant4-based) can be applied to the fast sim.



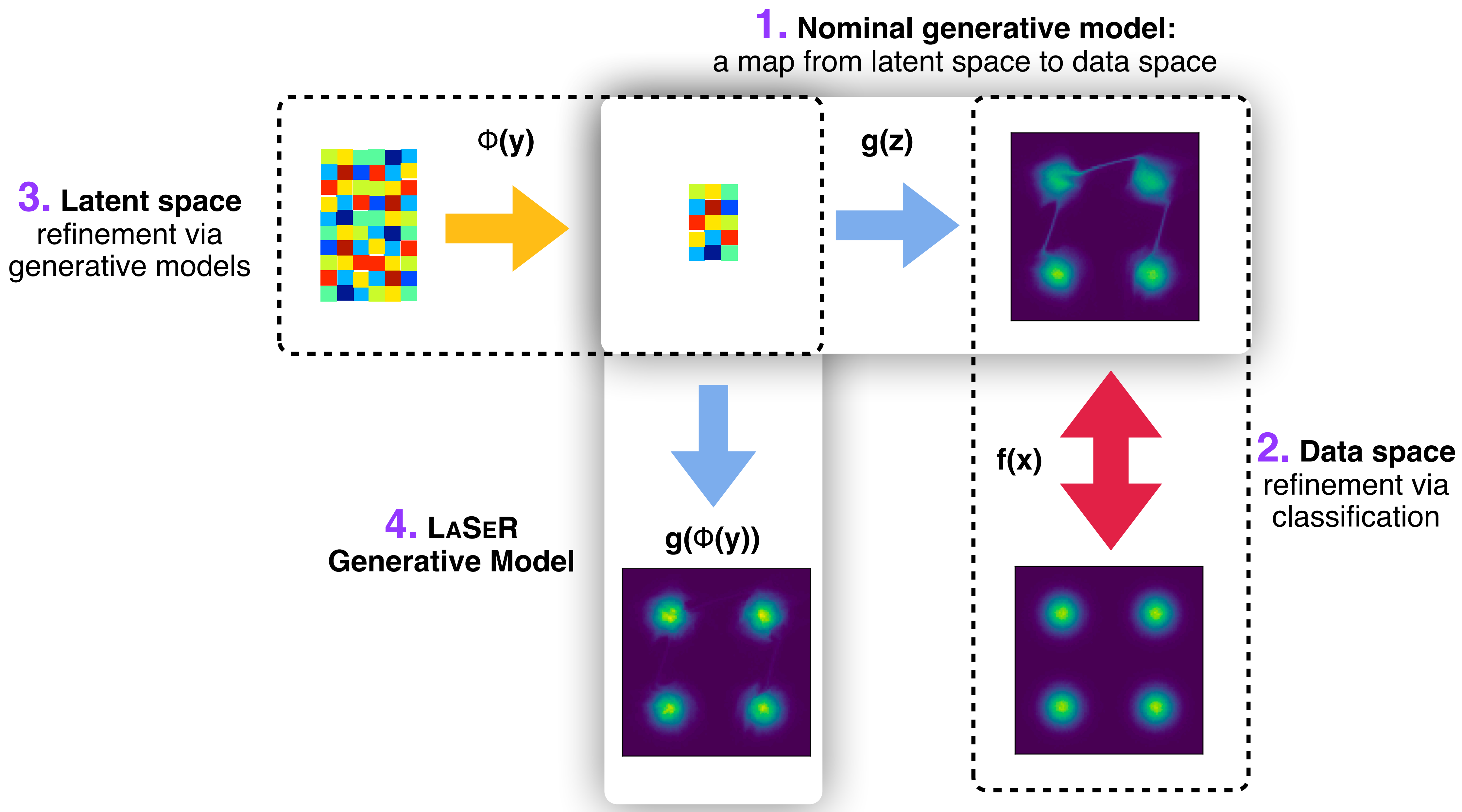
Integration into real detector sim.



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

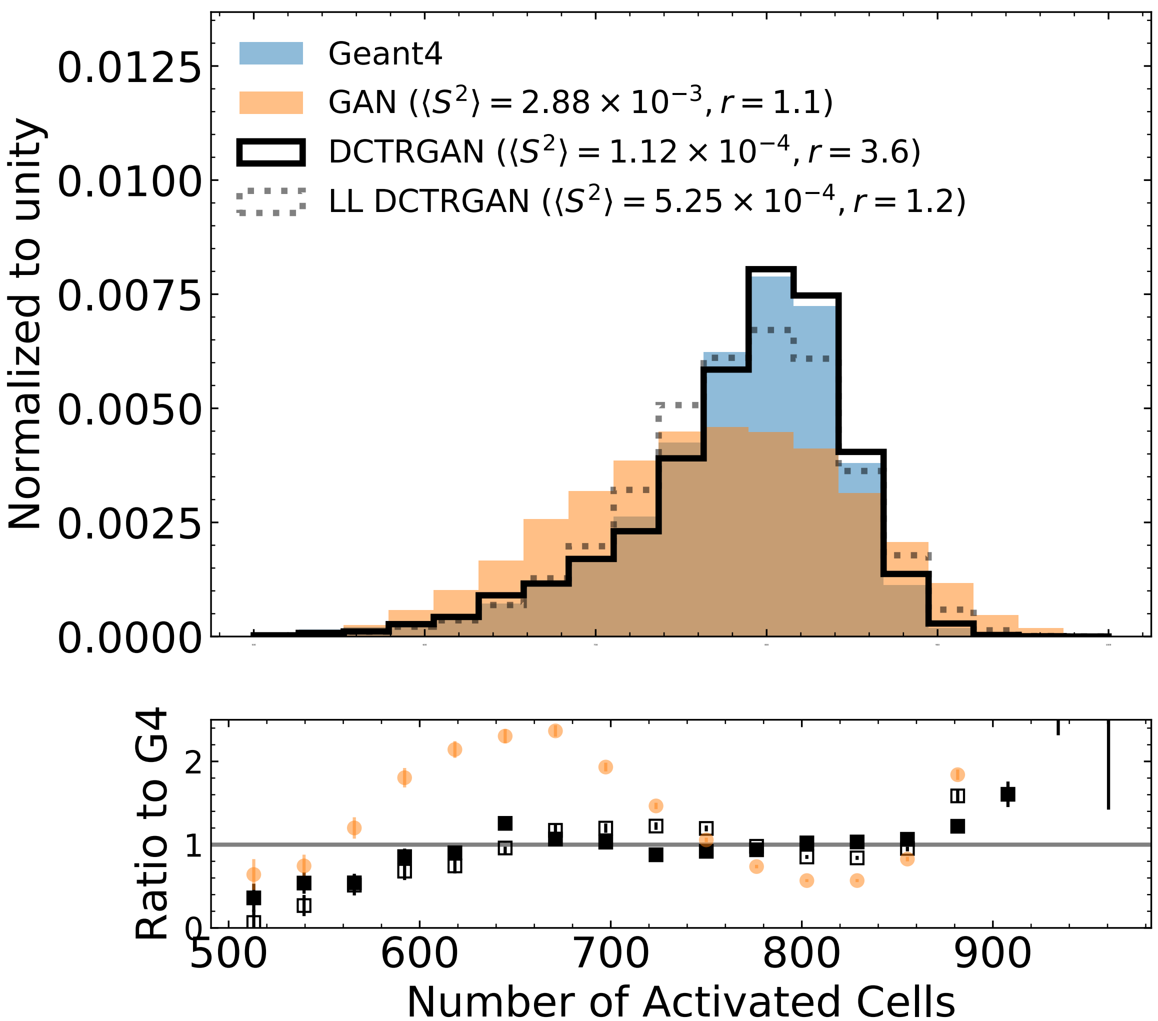
Refining Simulations

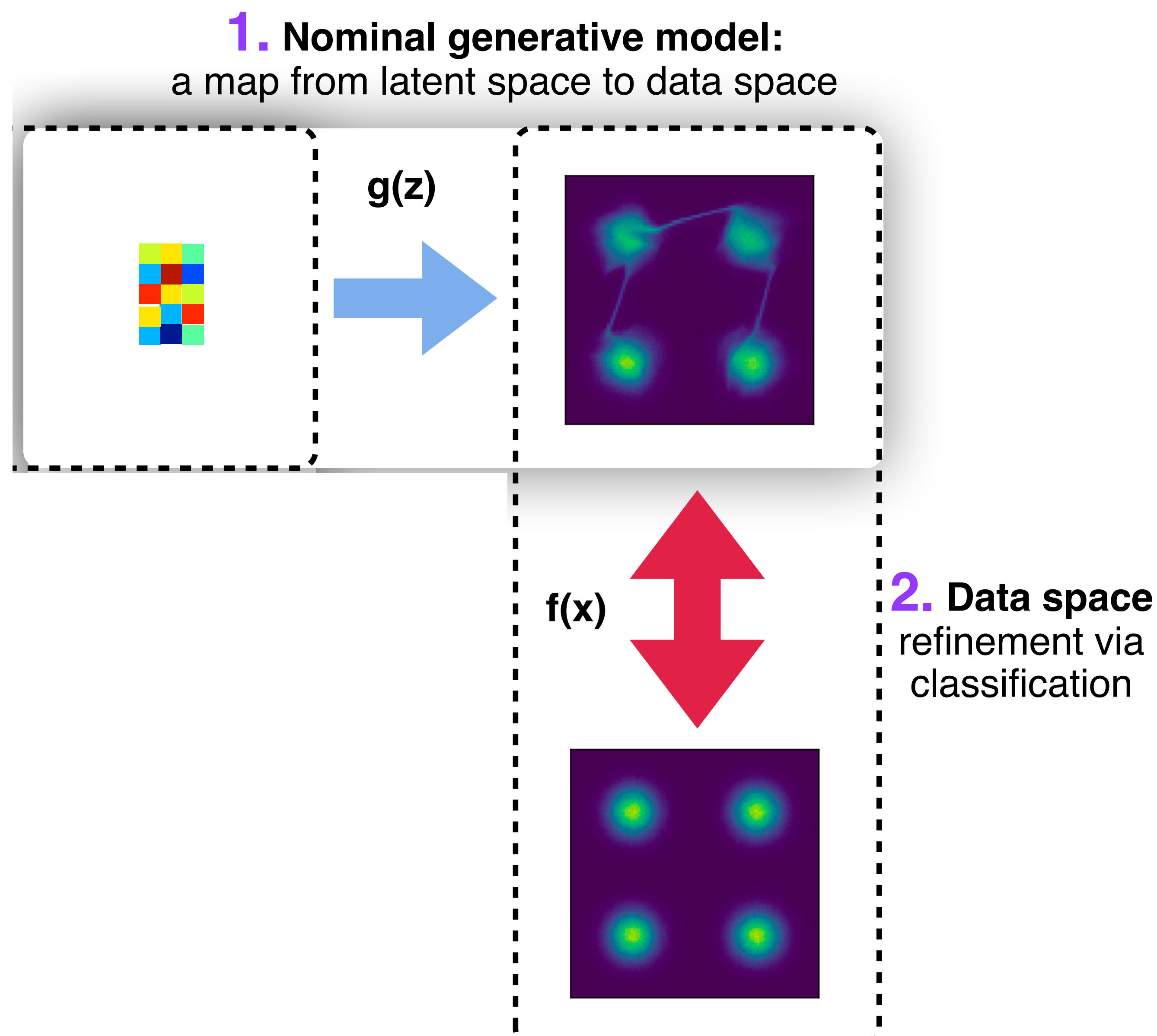
As we move towards precision, we may need to complement primary generative models with post-hoc correction models (e.g. via reweighting)



Refining Simulations

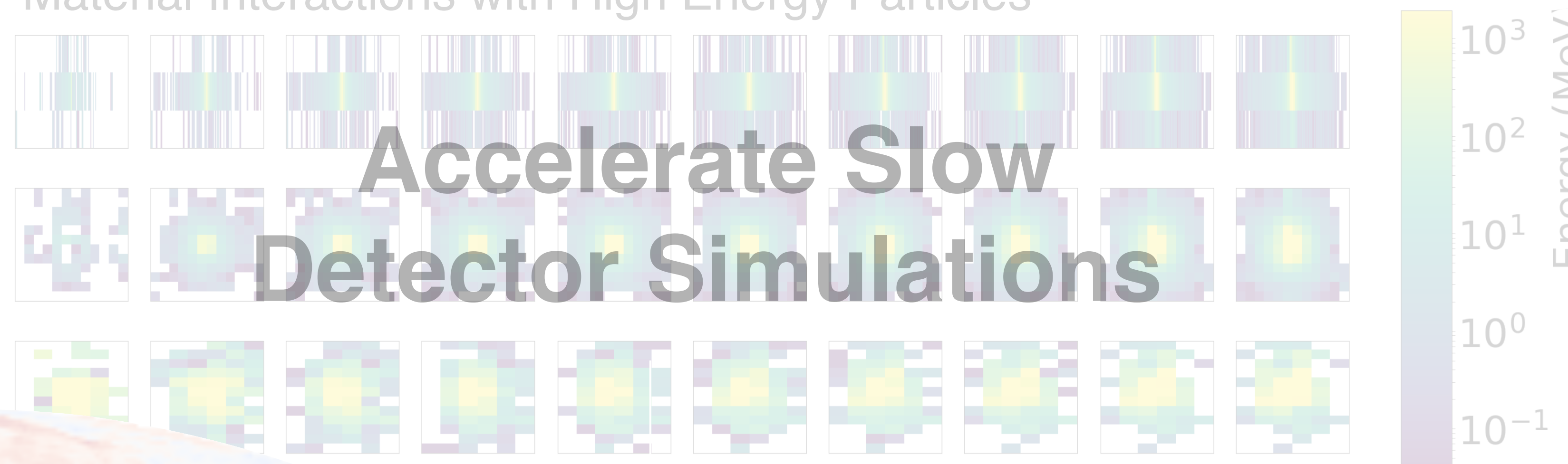
As we move towards precision, we may need to complement primary generative models with post-hoc correction models (e.g. via reweighting)





Generative Models for Particle/Nuclear/Astro

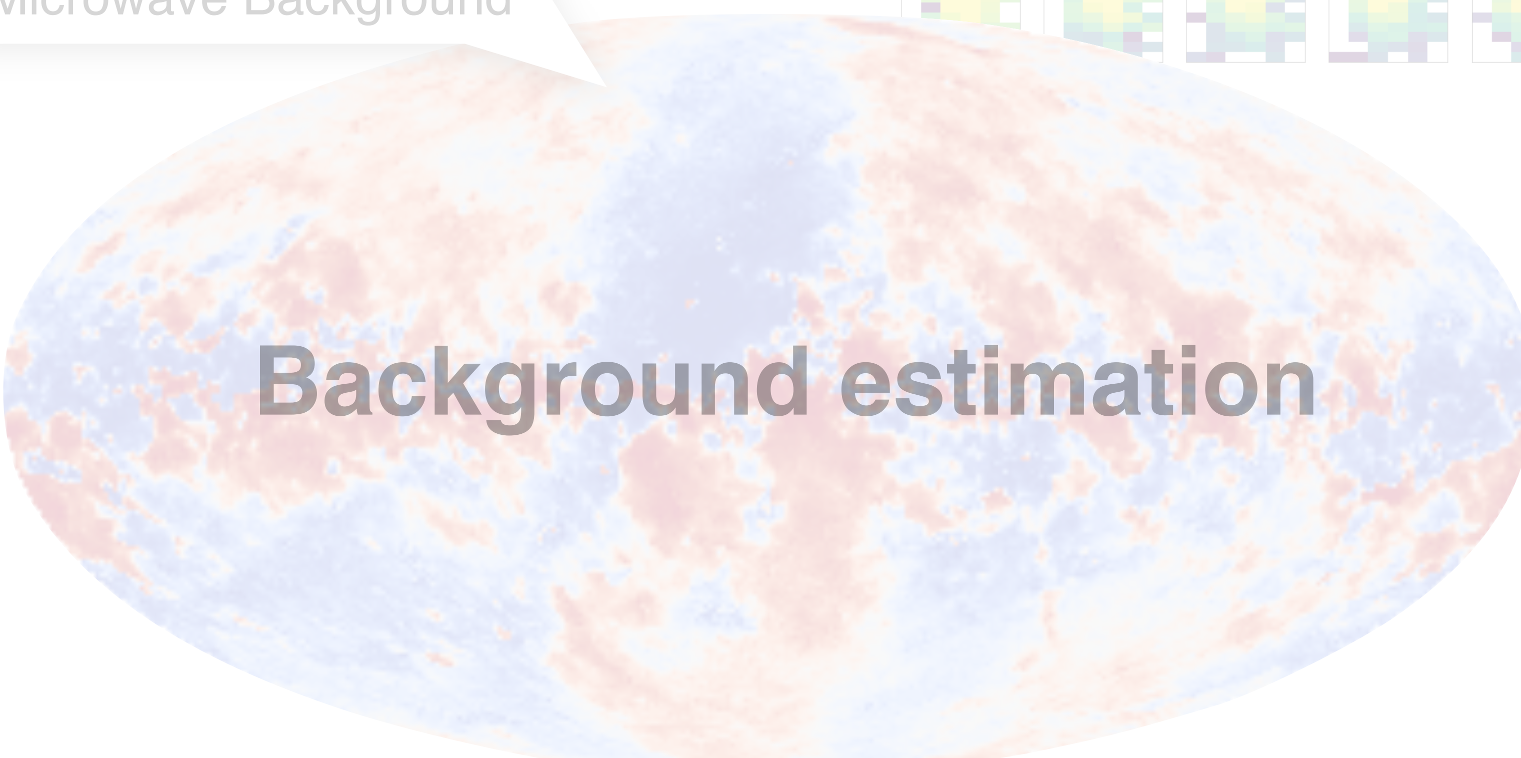
All of these pictures are fake!



Accelerate Slow Detector Simulations

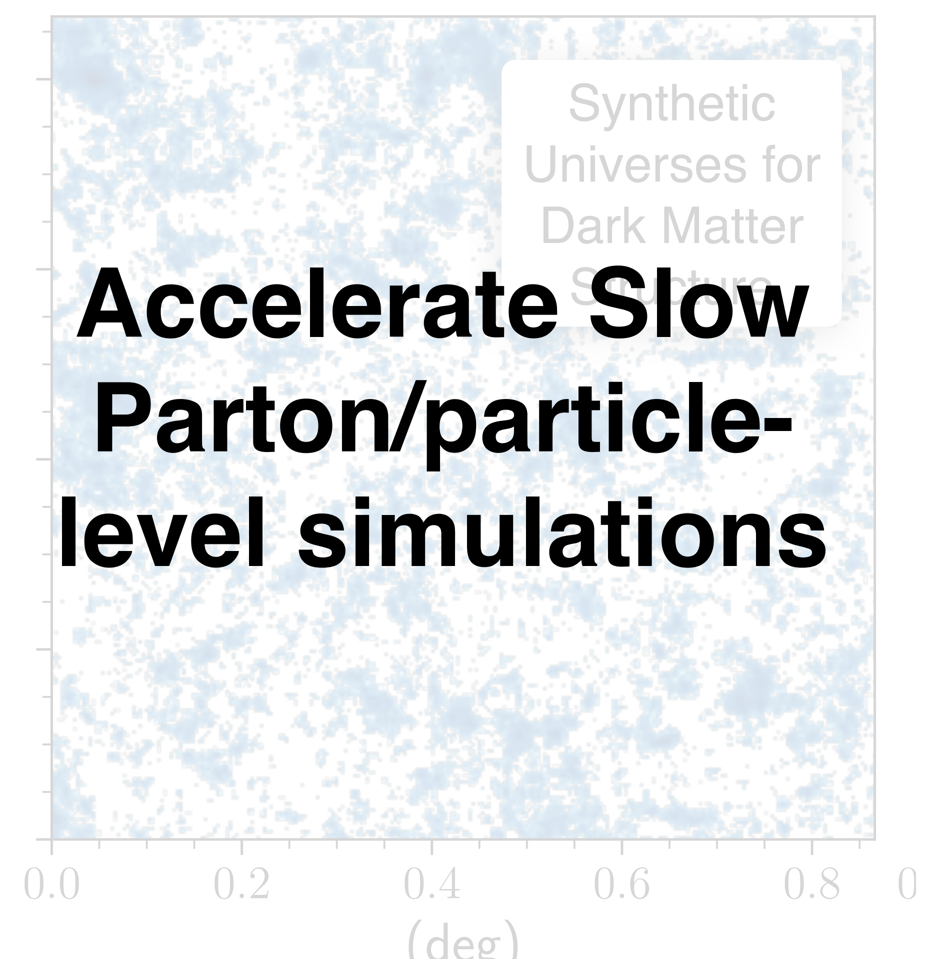
M. Paganini, L. De Oliveira, B. Nachman, Phys. Rev. Lett. 120 (2018) 042003

Synthetic Galactic radiation for Cosmic Microwave Background



Background estimation

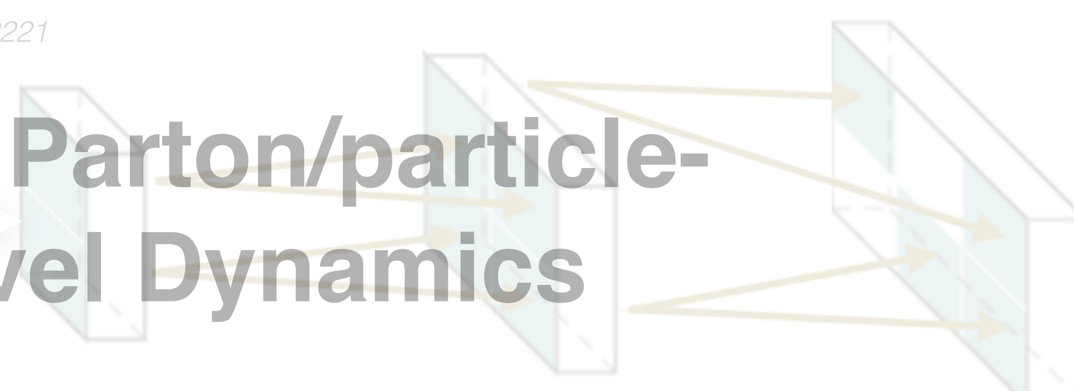
N. Krachmalnicoff and G. Puglisi, arXiv:2011.02221



Accelerate Slow Parton/particle-level simulations

M. Mustafa, et al., Comp. Astrophysics and Cosmology 6 (2019)

The Structure of Radiation in the Quantum Strong Force



Infer Parton/particle-level Dynamics

Y. S. Lai, D. Neill, M. Płoskoń, F. Ringer, arXiv:2012.06582

Accelerating Parton/Particle Sim.*

Flat jet images with GANs

M. Paganini, L. de Oliveira, B. Nachman

1701.05927

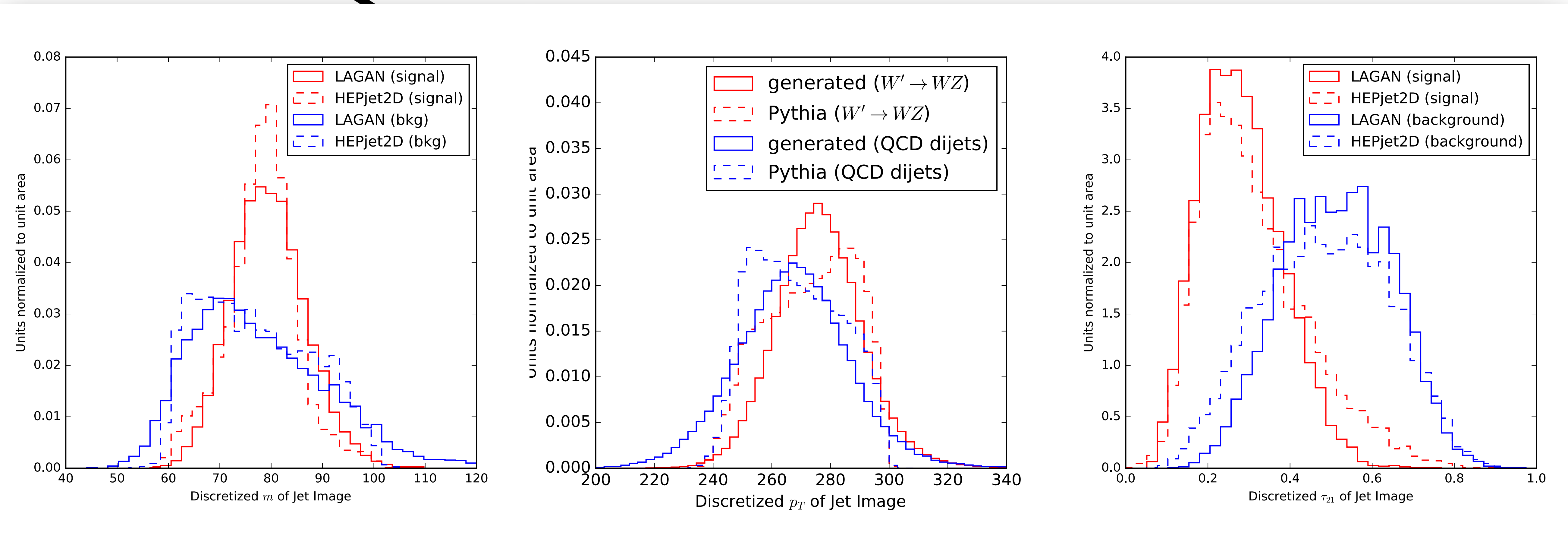
*these are just representative examples - see Living Review, 2102.02770

Accelerating Parton/Particle Sim.*

Flat jet images with GANs

M. Paganini, L. de Oliveira, B. Nachman

1701.05927



LA = Locally aware; somewhere between a DNN and a CNN

Weight sharing across space

*these are just representative examples - see Living Review, 2102.02770

Accelerating Parton/Partic

Flat jet images with GANs

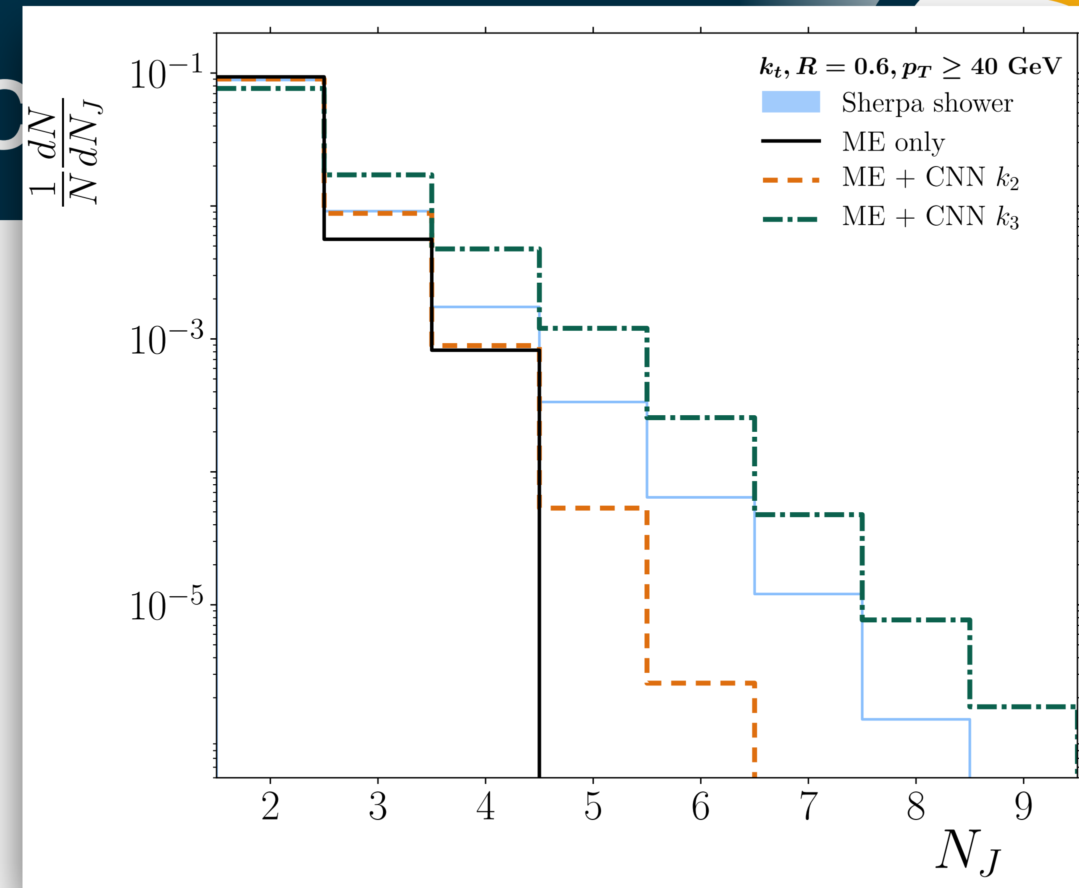
M. Paganini, L. de Oliveira, B. Nachman

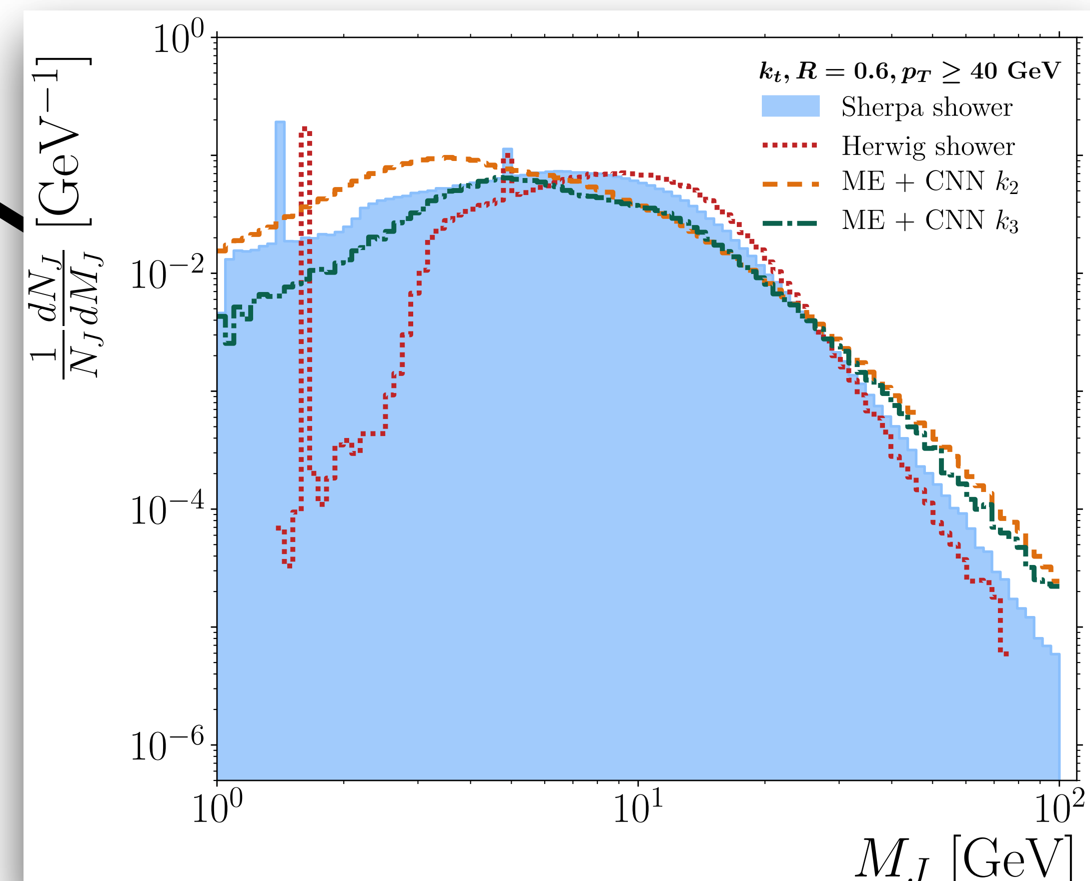
1701.05927

Scale invariant images with AEs

J. Monk

1807.03685





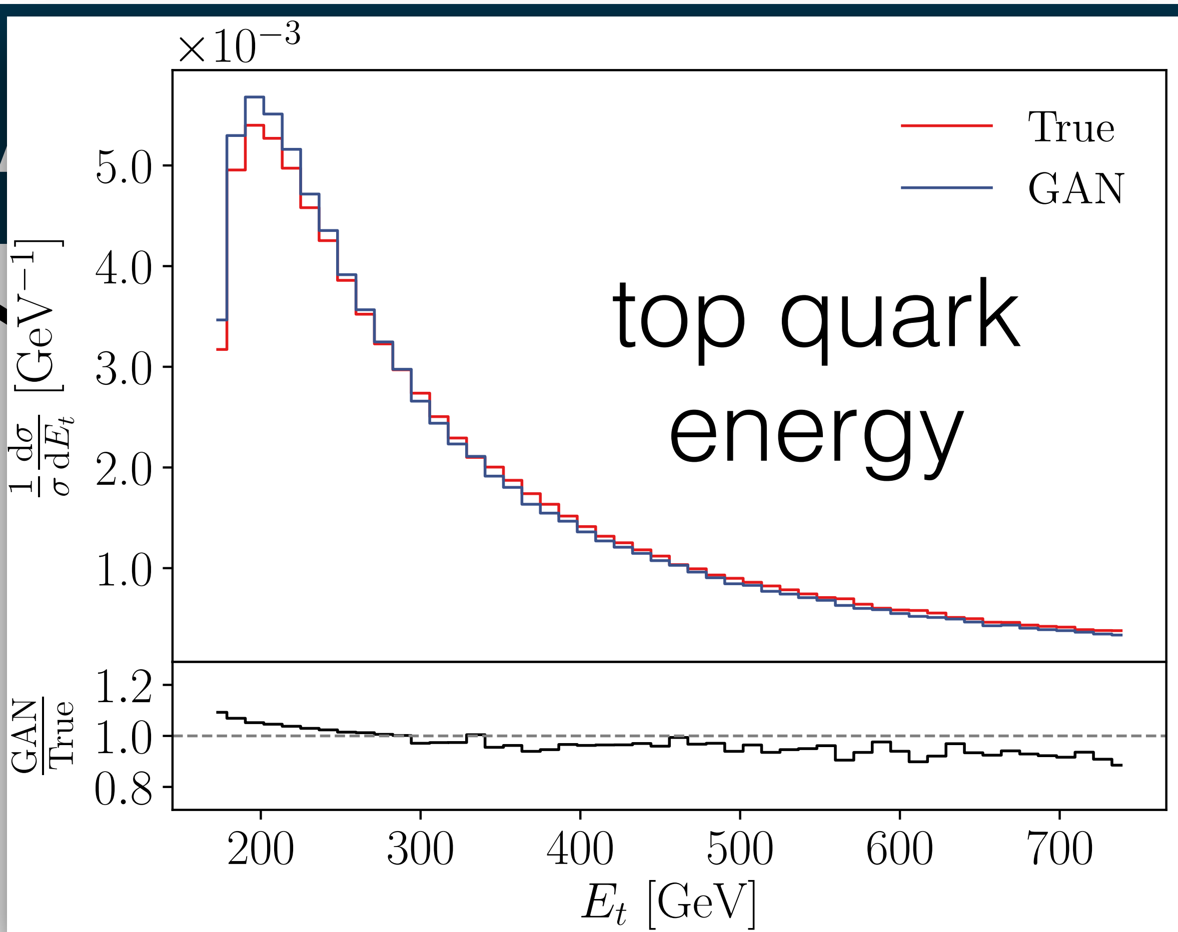
Weight sharing across space + "time"

*these are just representative examples - see Living Review, 2102.02770

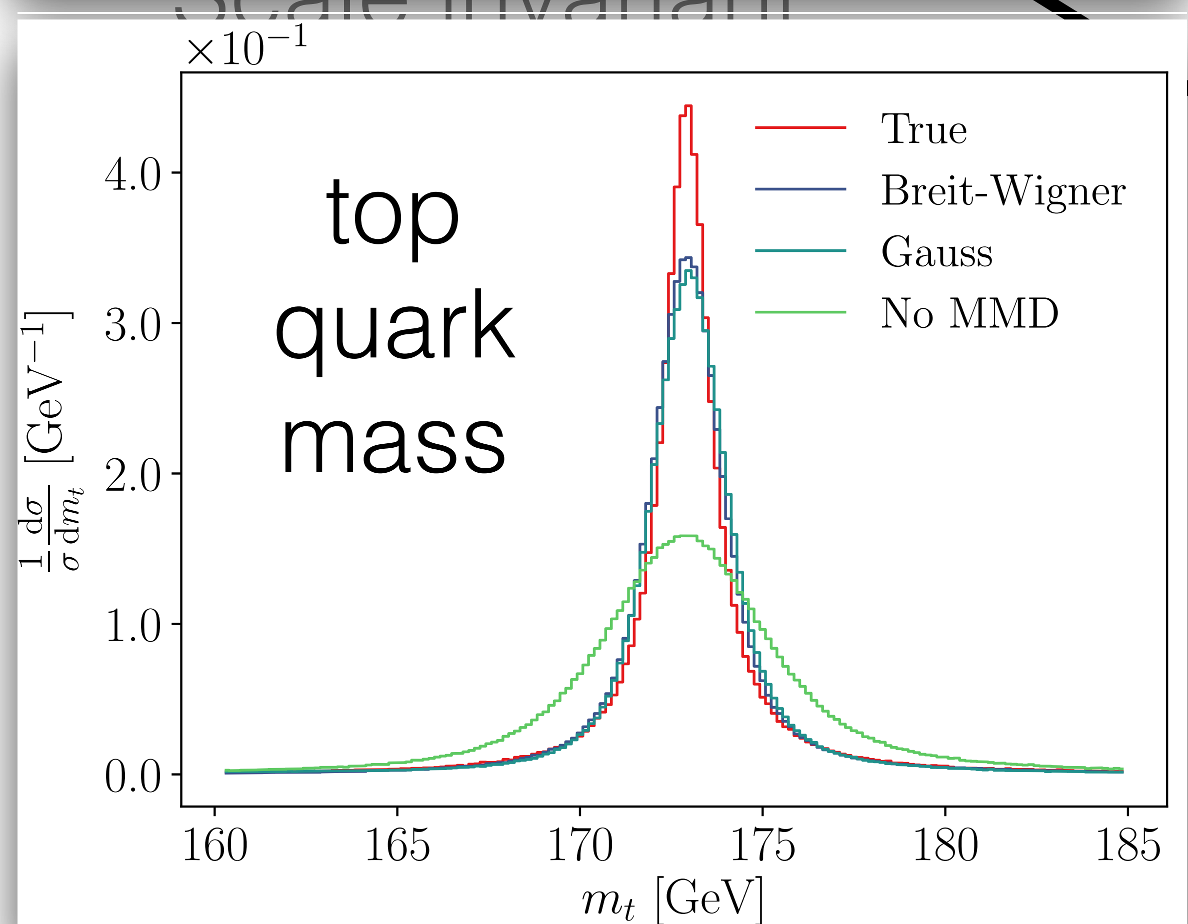
Particle Sim.*

ANs

nman



Scale invariant



MMD = maximum mean discrepancy

Fixed number of 4-vectors, allow for intermediate resonances

A. Butter, T. Plehn, R. Winterhalder

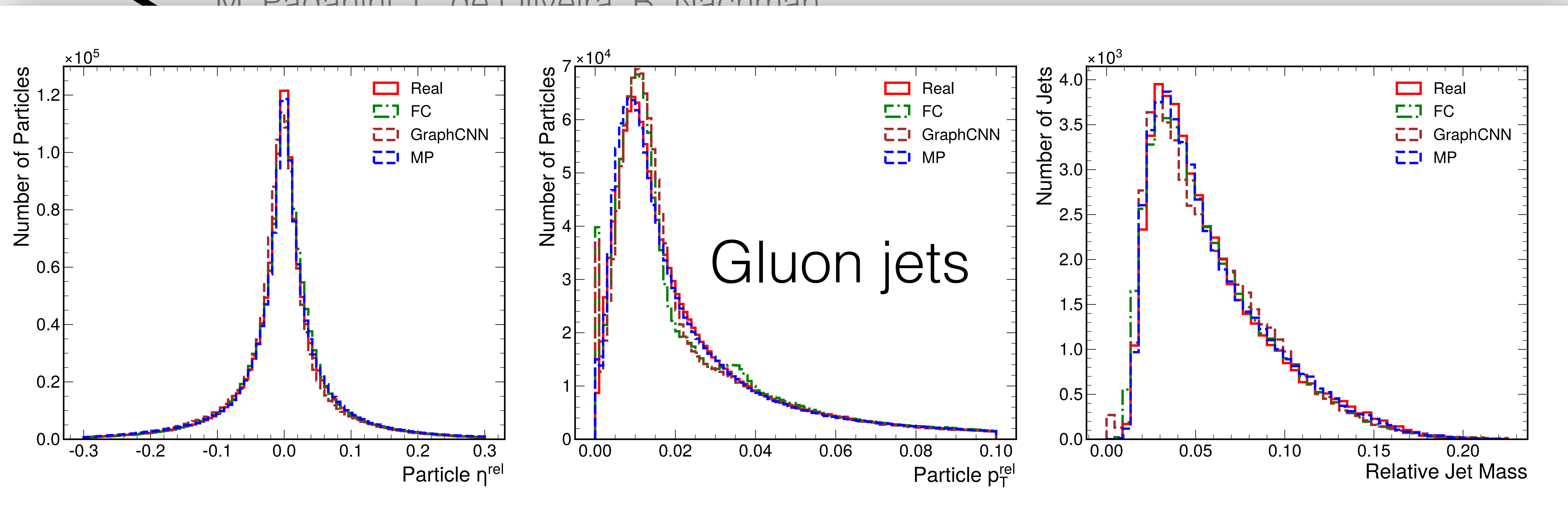
1907.03764

*these are just representative examples - see Living Review, 2102.02770

Accelerating Parton/Particle Sim.*

Flat jet images with GANs

M. Paganini, L. de Oliveira, B. Nachman



Variable-length output with graphs

R. Kansal et al.

2106.11535

*these are just representative examples - see Living Review, 2102.02770

Accelerating Parton/Particle Sim.*

Flat jet images with GANs
M. Paganini, L. de Oliveira, B. Nachman
1701.05927

Scale invariant images with AEs
J. Monk
1807.03685

Fixed number of 4-vectors, allow for intermediate resonances
A. Butter, T. Plehn, R. Winterhalder
1907.03764

Variable-length output with graphs
R. Kansal et al.
2106.11535

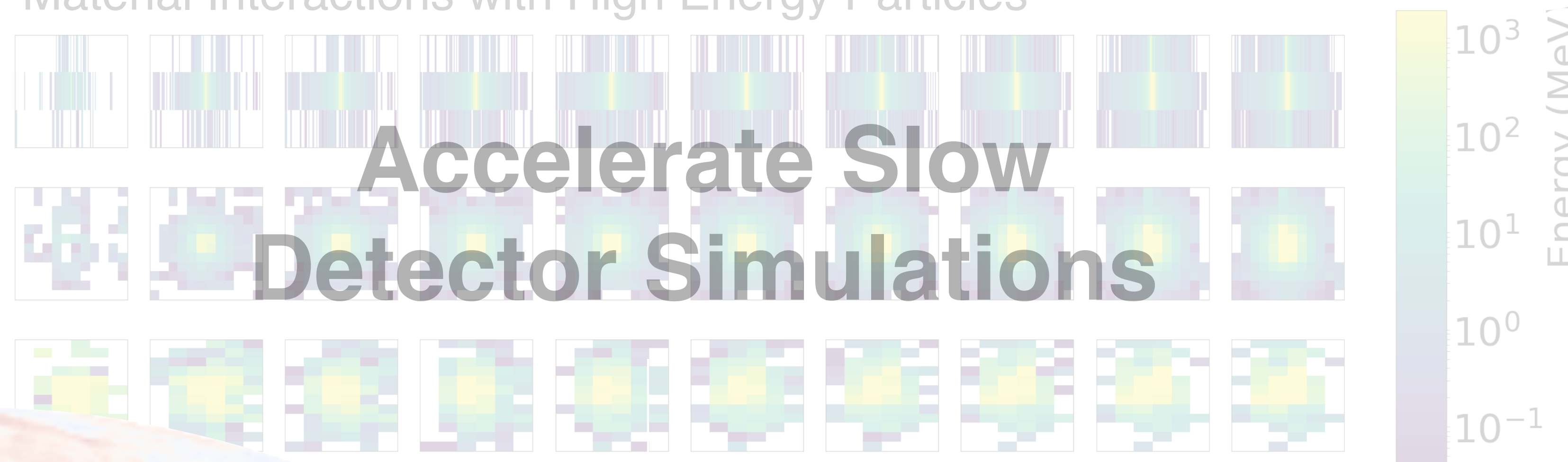
?

*these are just representative examples - see Living Review, 2102.02770

Generative Models for Particle/Nuclear/Astro

All of these pictures are fake!

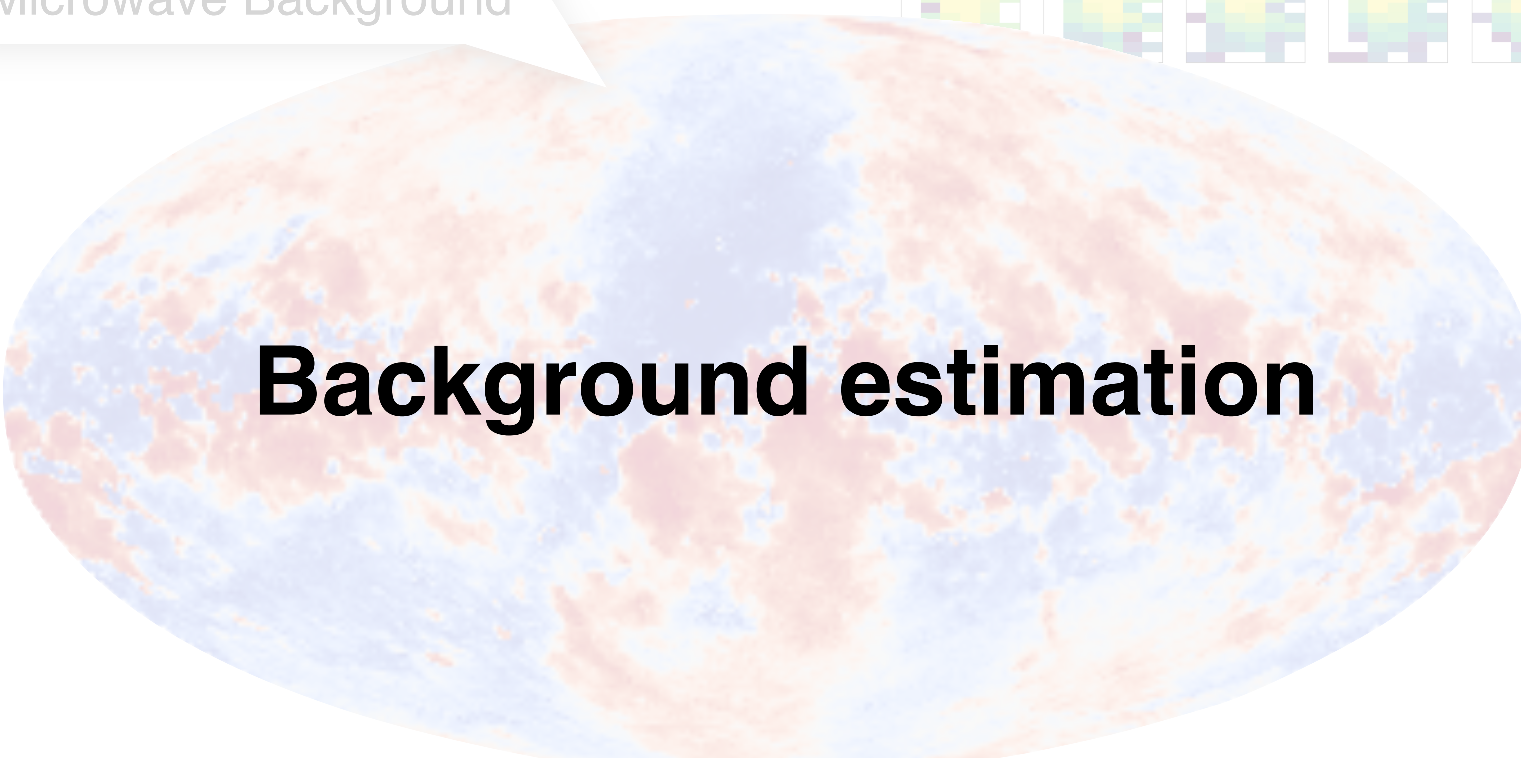
Material Interactions with High Energy Particles



Accelerate Slow Detector Simulations

M. Paganini, L. De Oliveira, B. Nachman, Phys. Rev. Lett. 120 (2018) 042003

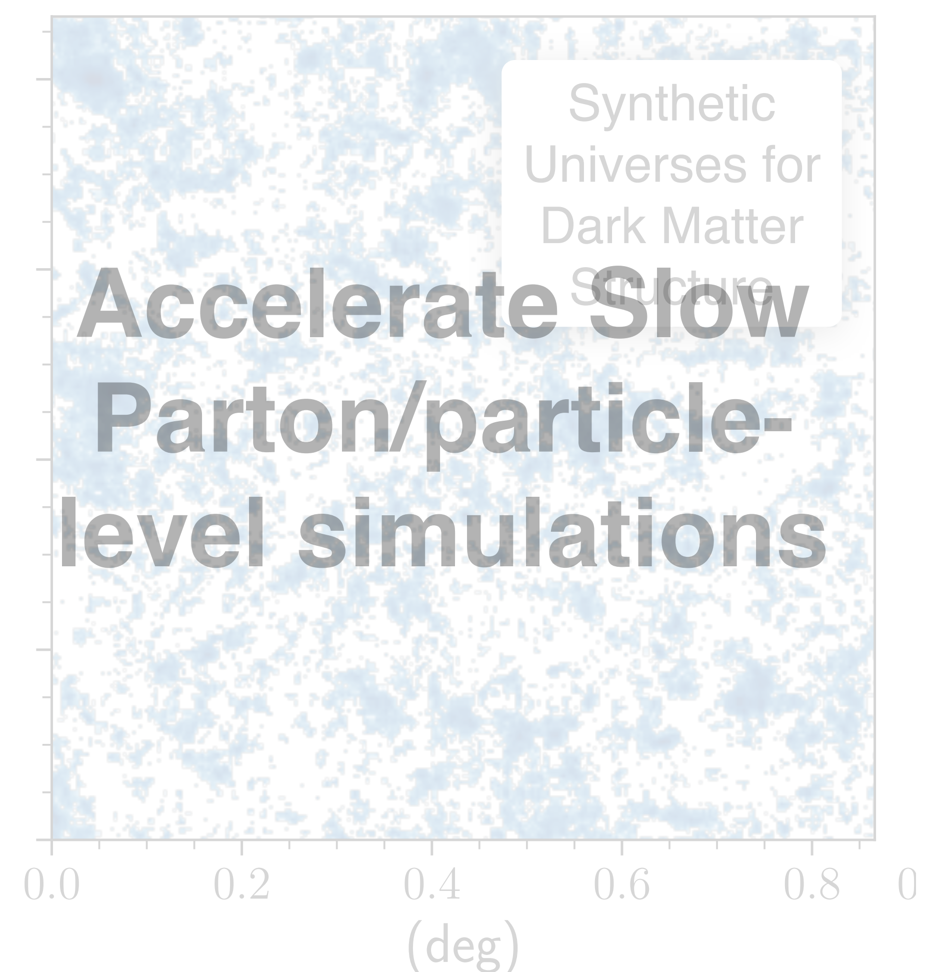
Synthetic Galactic radiation for Cosmic Microwave Background



Background estimation

N. Krachmalnicoff and G. Puglisi, arXiv:2011.02221

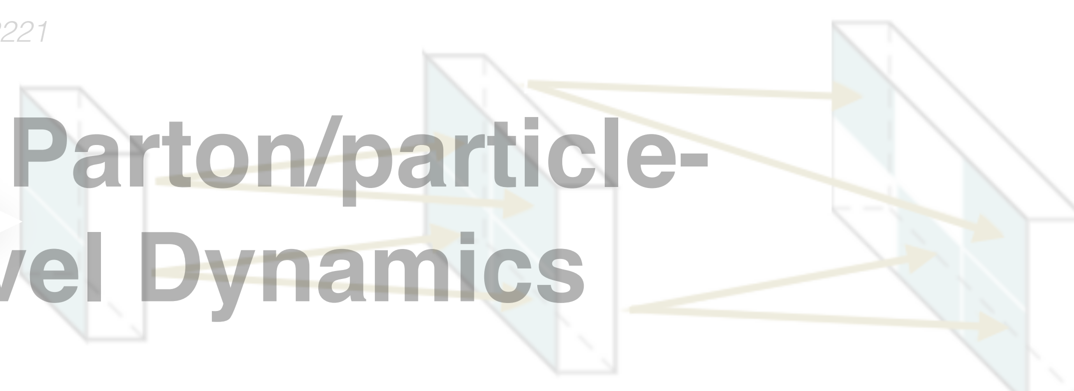
Synthetic Universes for Dark Matter Structure



Accelerate Slow Parton/particle-level simulations

M. Mustafa, et al., Comp. Astrophysics and Cosmology 6 (2019)

The Structure of Radiation in the Quantum Strong Force



Infer Parton/particle-level Dynamics

Y. S. Lai, D. Neill, M. Płoskoń, F. Ringer, arXiv:2012.06582

Background Estimation

Many cases where simulation is **not good enough** - can train generate models to act as **simulation surrogates**

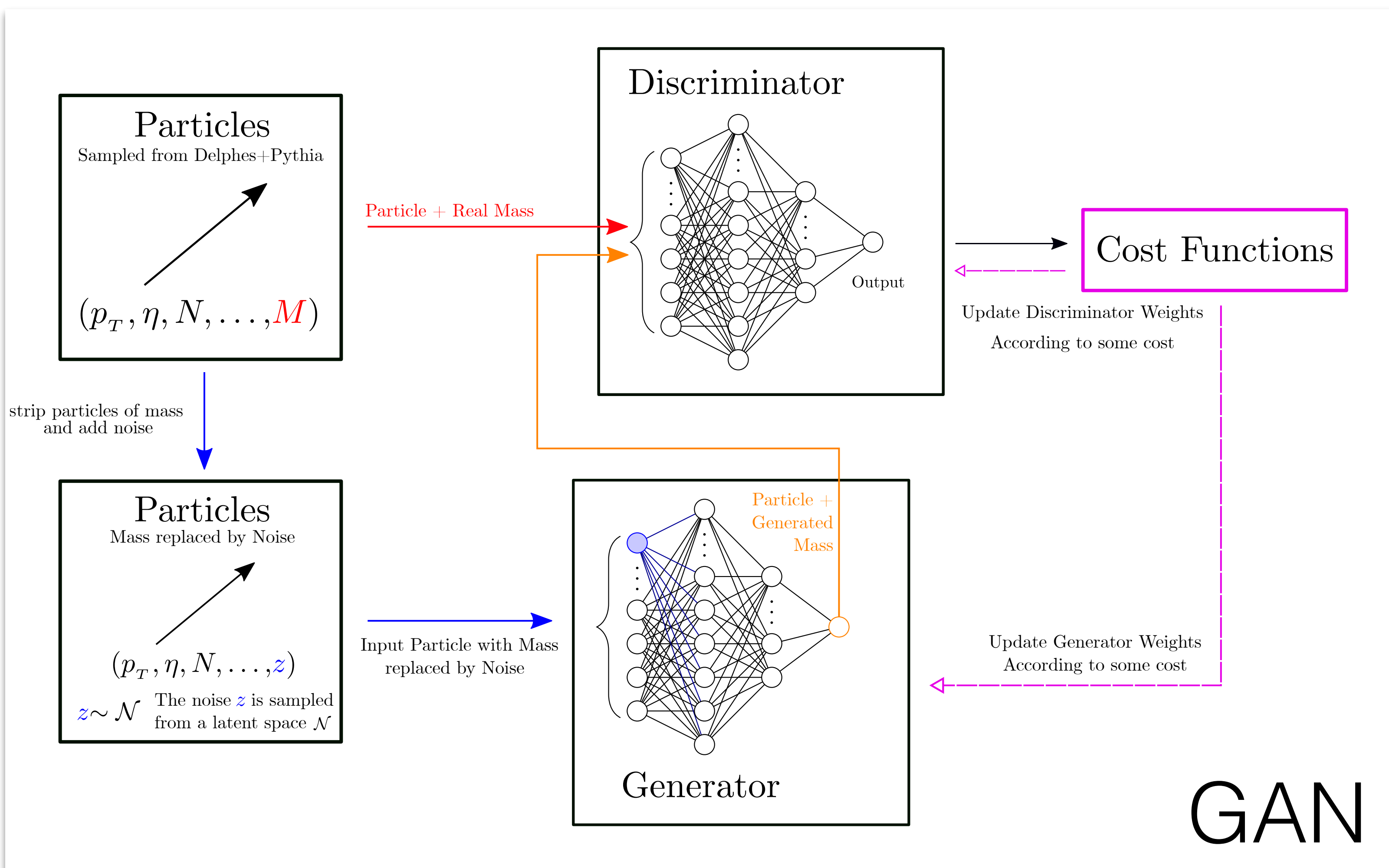
N.B. everything in I've shown before this, we trained on simulation, not on data (!)

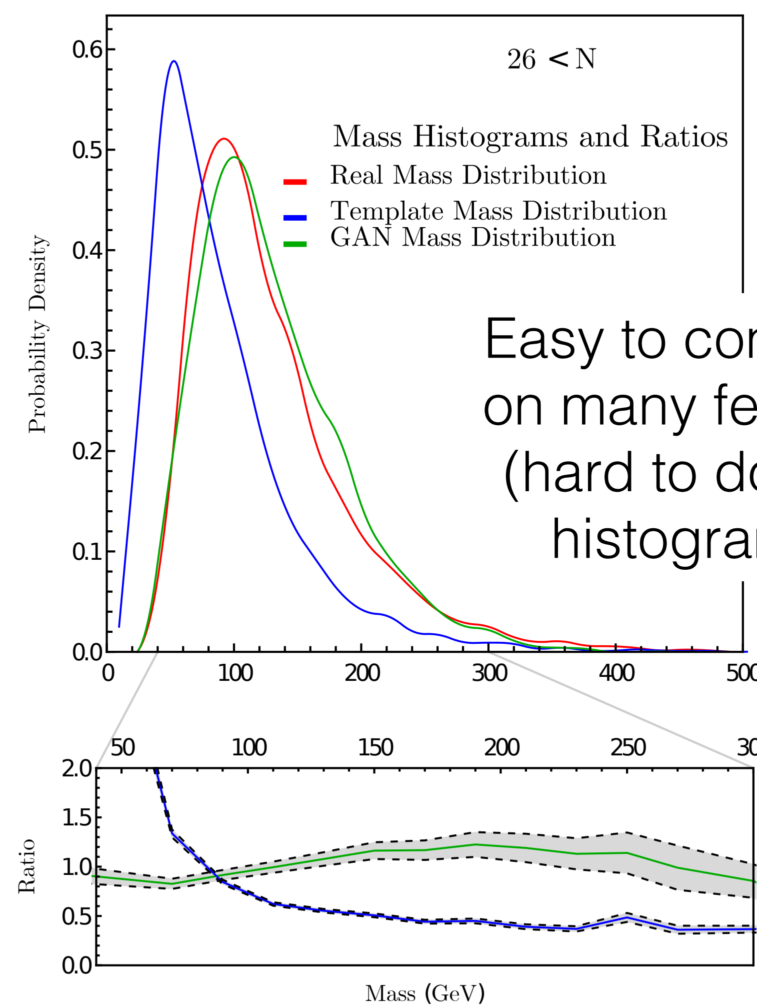
J. Lin, W. Bhimji, B. Nachman 1903.02556

Background Estimation

Many cases where simulation is **not good enough** - can train generate models to act as **simulation surrogates**

Example 1: unbinned templates for QCD jets to extrapolate in jet multiplicity



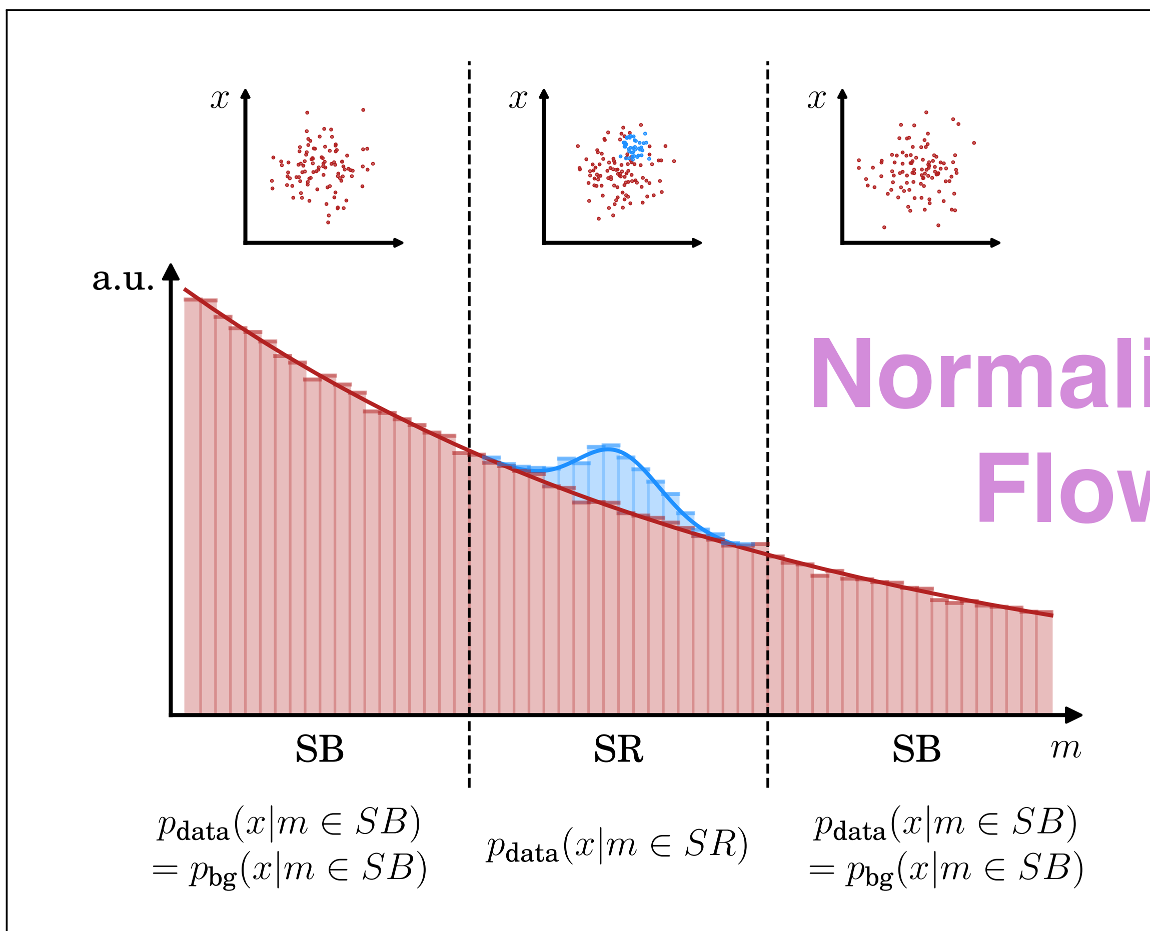


Easy to condition on many features (hard to do with histograms)

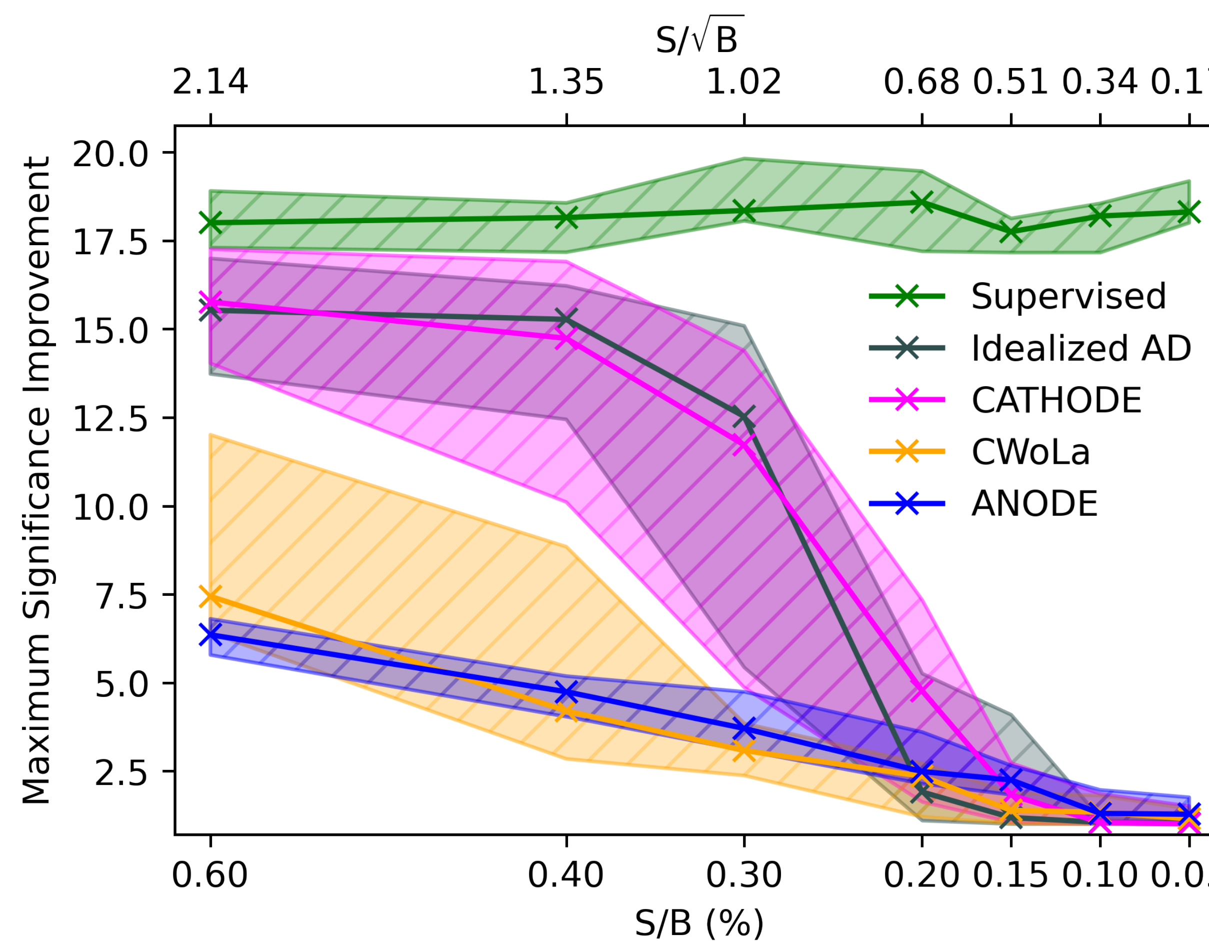
Background Estimation

Many cases where simulation is **not good enough** - can train generate models to act as **simulation surrogates**

Example 2: unbinned templates for QCD jets to extrapolate in dijet mass



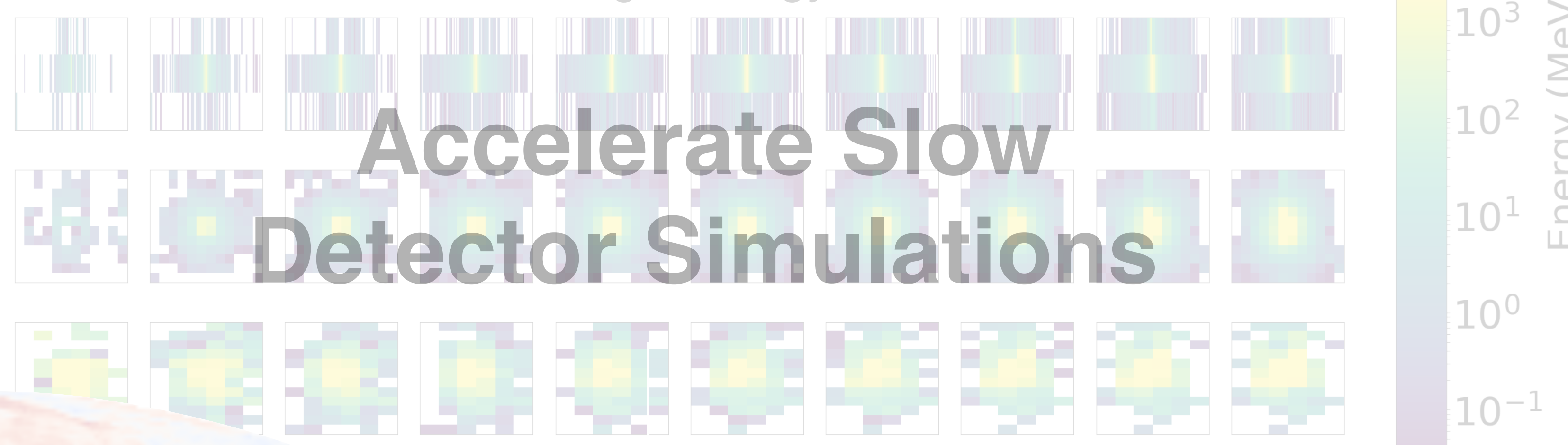
Normalizing Flow



Anomaly detection

Generative Models for Particle/Nuclear/Astro

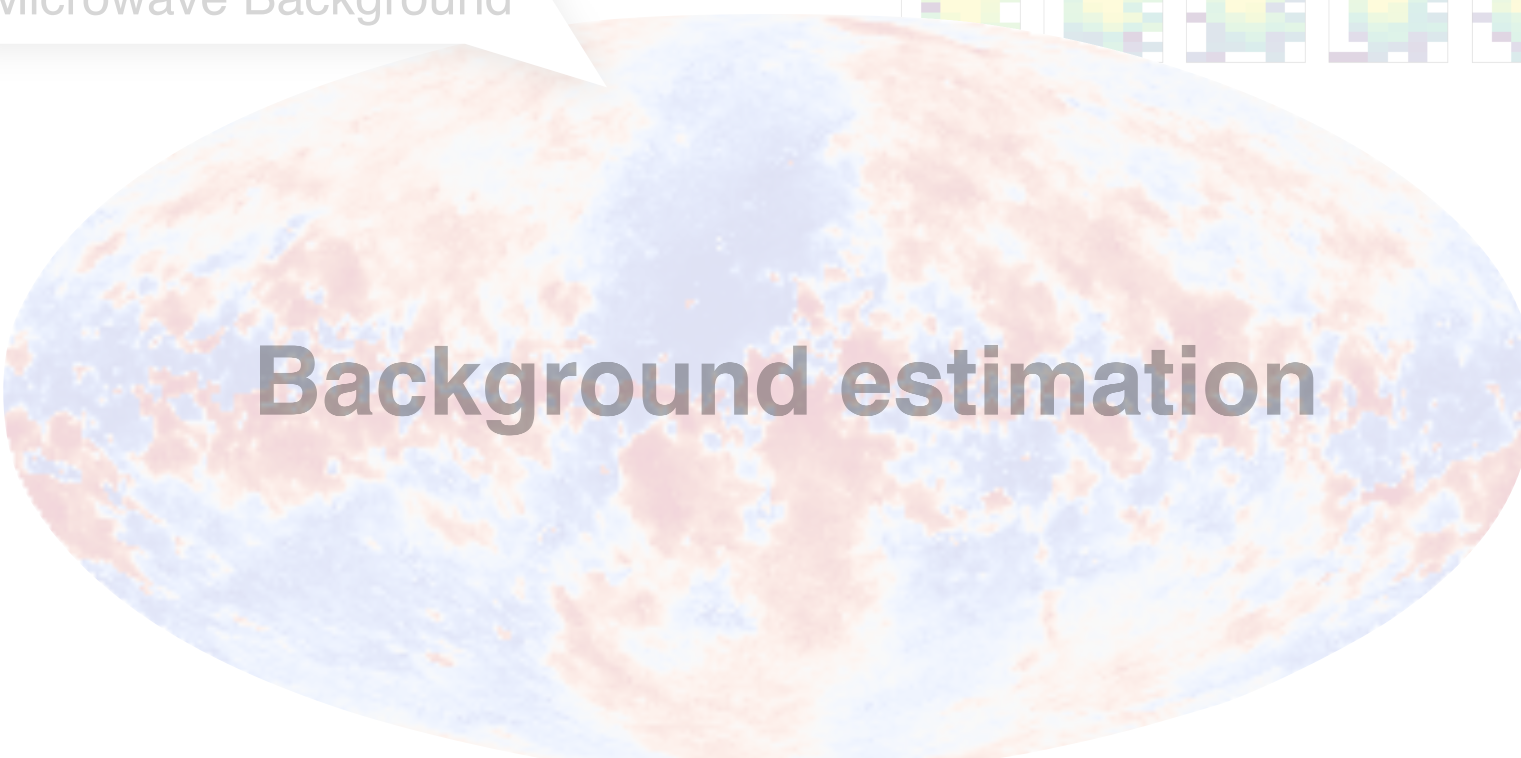
All of these pictures are fake!



Accelerate Slow Detector Simulations

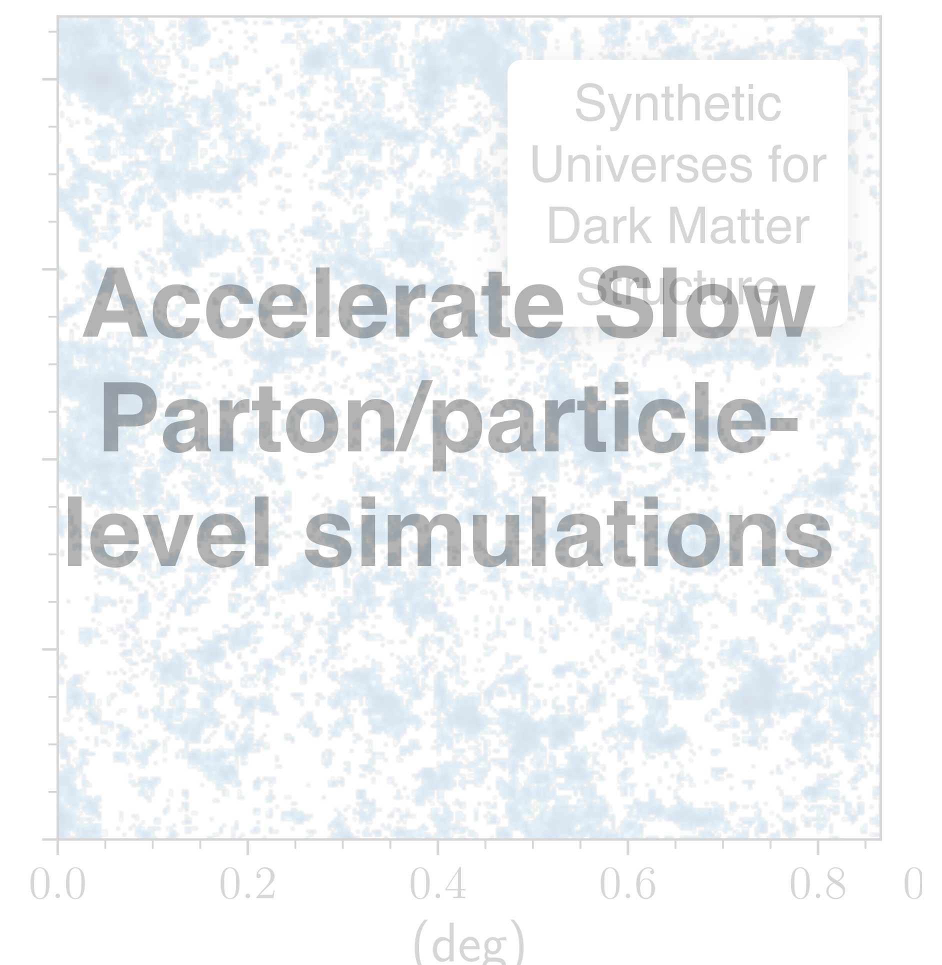
M. Paganini, L. De Oliveira, B. Nachman, Phys. Rev. Lett. 120 (2018) 042003

Synthetic Galactic radiation for Cosmic Microwave Background



Background estimation

N. Krachmalnicoff and G. Puglisi, arXiv:2011.02221

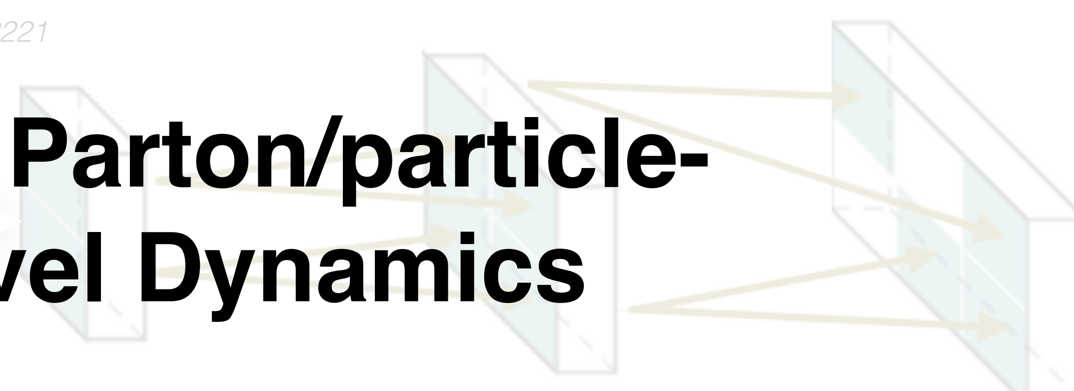


Synthetic Universes for Dark Matter Structure

Accelerate Slow Parton/particle-level simulations

M. Mustafa, et al., Comp. Astrophysics and Cosmology 6 (2019)

The Structure of Radiation in the Quantum Strong Force



Infer Parton/particle-level Dynamics

Y. S. Lai, D. Neill, M. Płoskoń, F. Ringer, arXiv:2012.06582

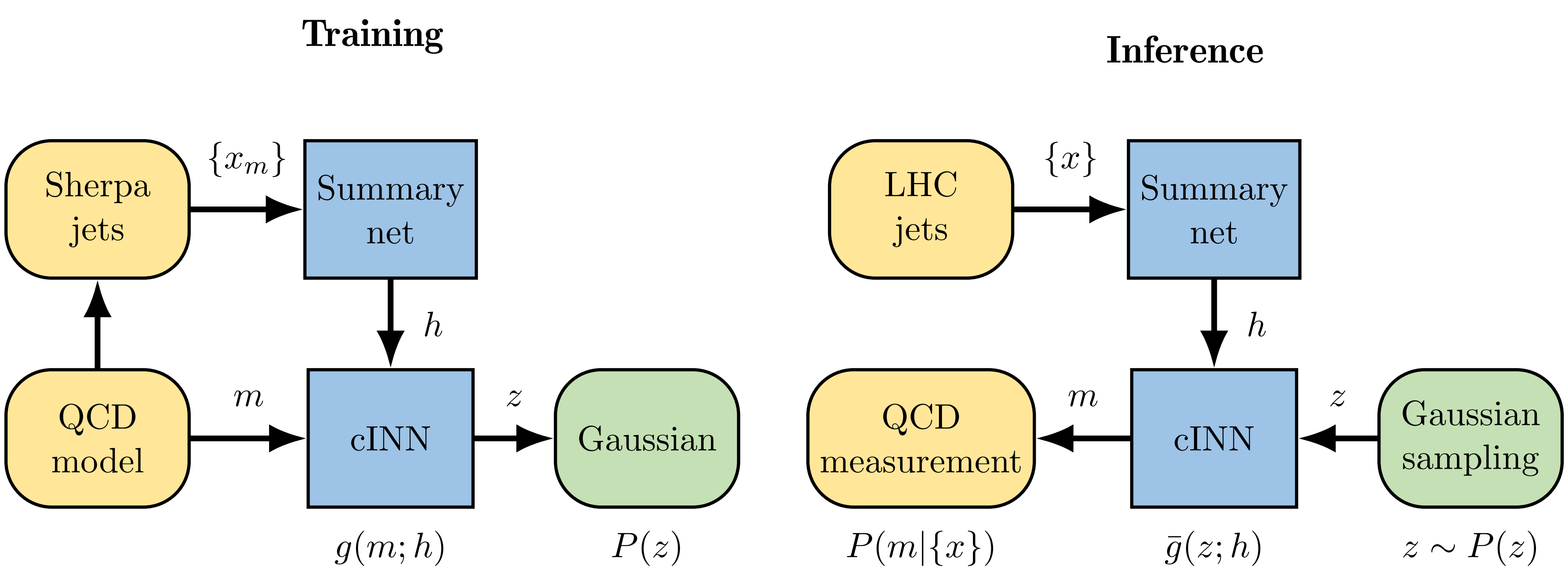
Infering Parton/particle-level Dynamics

Can we use generative models directly for inference?
(and not "just" for augmenting/accelerating simulation)

Infering Parton/particle-level Dynamics

Can we use generative models directly for inference?
(and not "just" for augmenting/accelerating simulation)

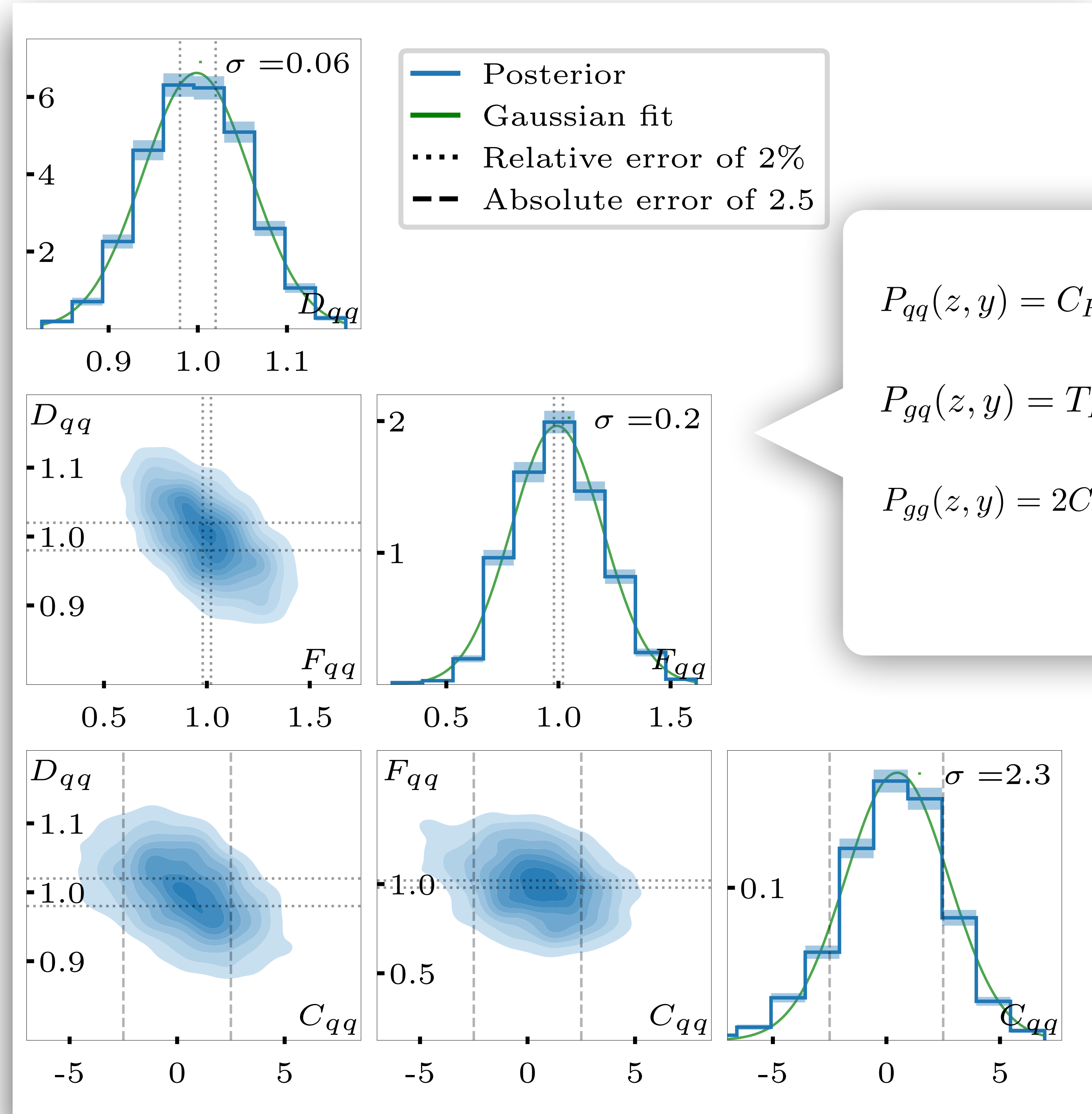
Example 1: Inferring fragmentation functions



See also 1804.09720 ("JUNIPR") and 2012.06582 (GAN-based)

Infering Parton/particle-level Dynamics

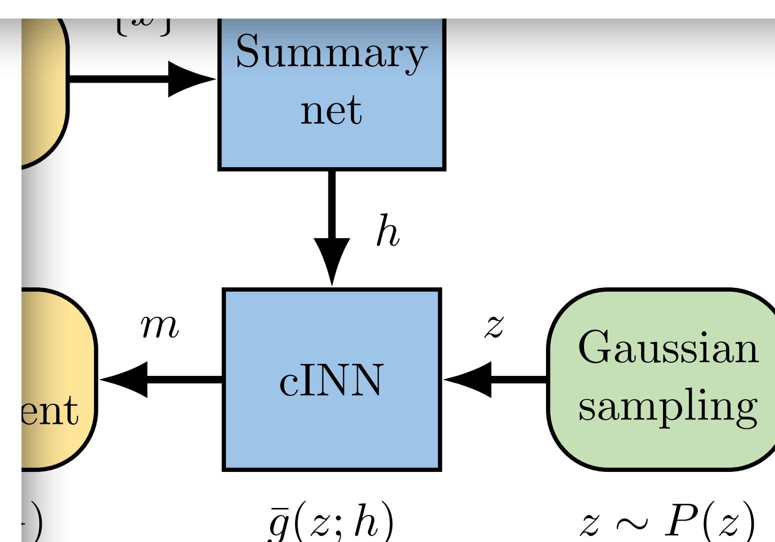
…directly for inference?



$$P_{qq}(z,y) = C_F\left[D_{qq}\frac{2z(1-y)}{1-z(1-y)} + F_{qq}(1-z) + C_{qq}yz(1-z)\right]$$

$$P_{gq}(z,y) = T_R\left[F_{qq}\left(z^2+(1-z)^2\right) + C_{gq}yz(1-z)\right]$$

$$P_{gg}(z,y) = 2C_A\Big[D_{gg}\left(\frac{z(1-y)}{1-z(1-y)} + \frac{(1-z)(1-y)}{1-(1-z)(1-y)}\right) + F_{gg}z(1-z) + C_{gg}yz(1-z)\Big]$$



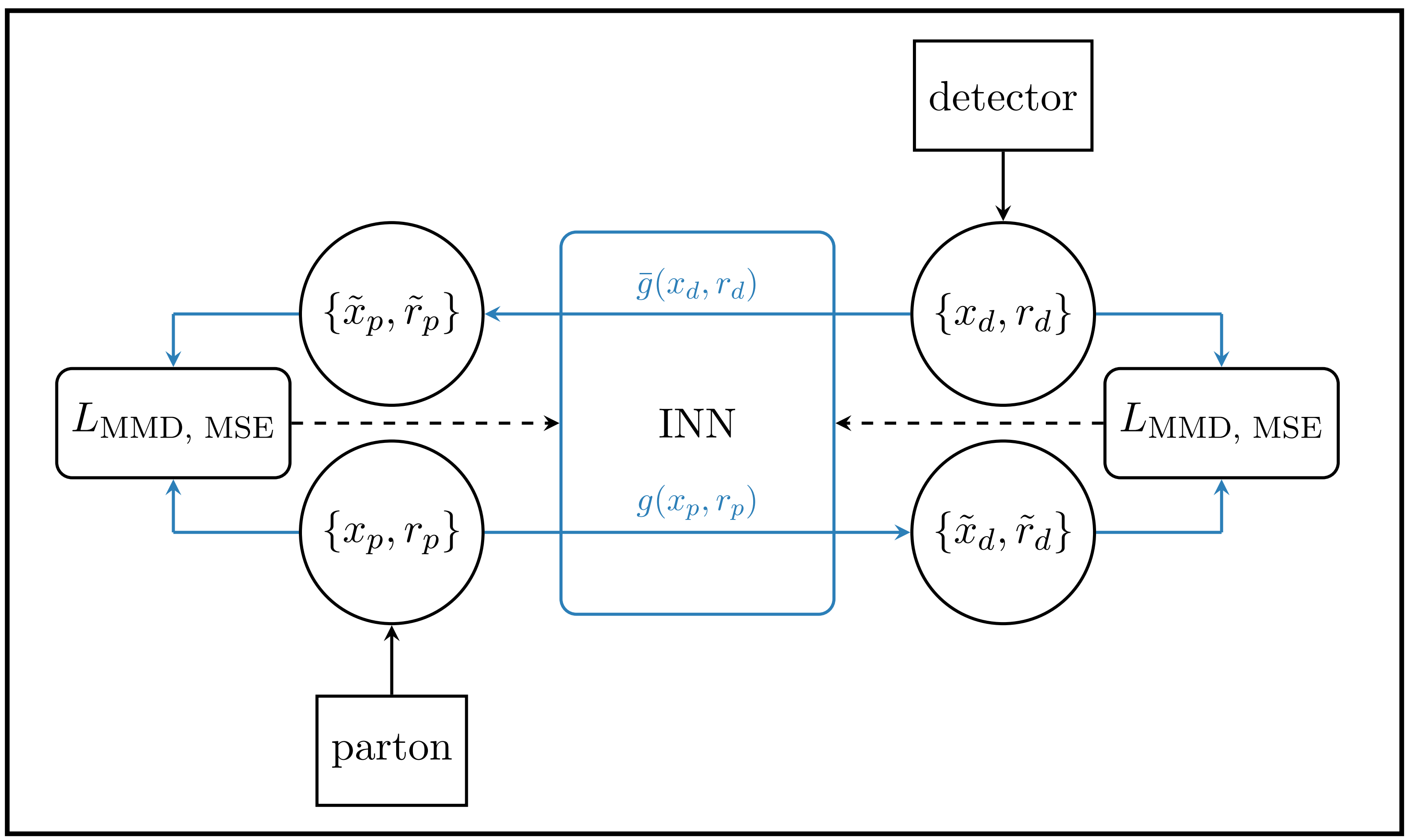
See also 1804.09720 ("JUNIPR") and 2012.06582 (GAN-based)

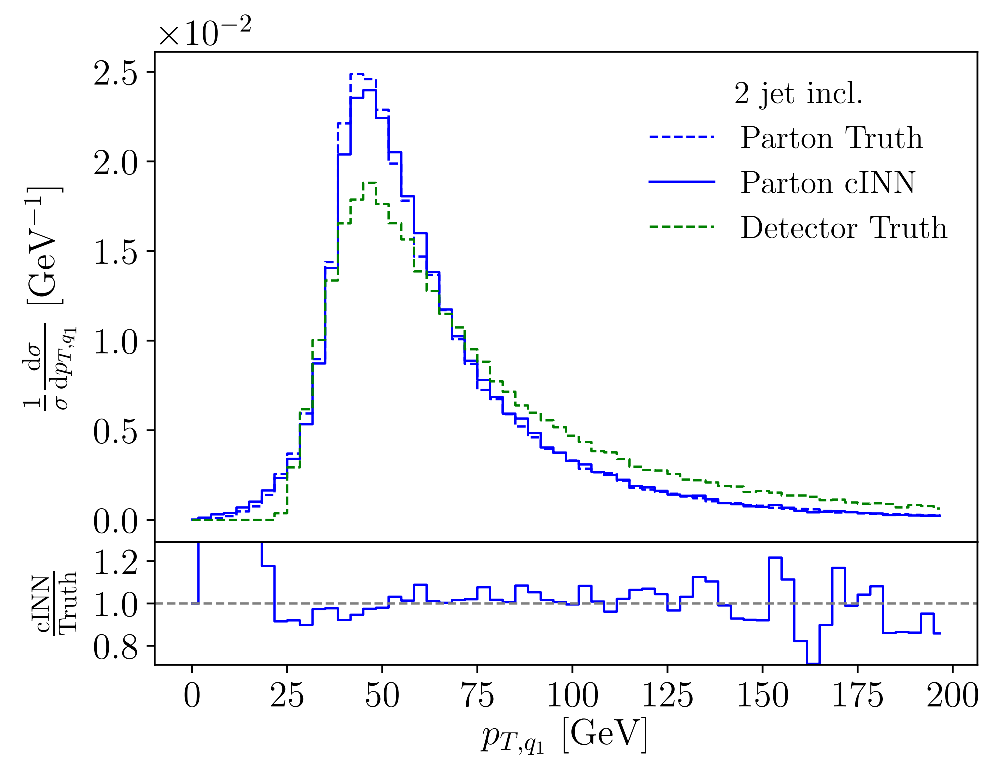
M. Bellagente et al., 2006.06685

Infering Parton/particle-level Dynamics

Can we use generative models directly for inference?
(and not "just" for augmenting/accelerating simulation)

Example 2: Unfolding

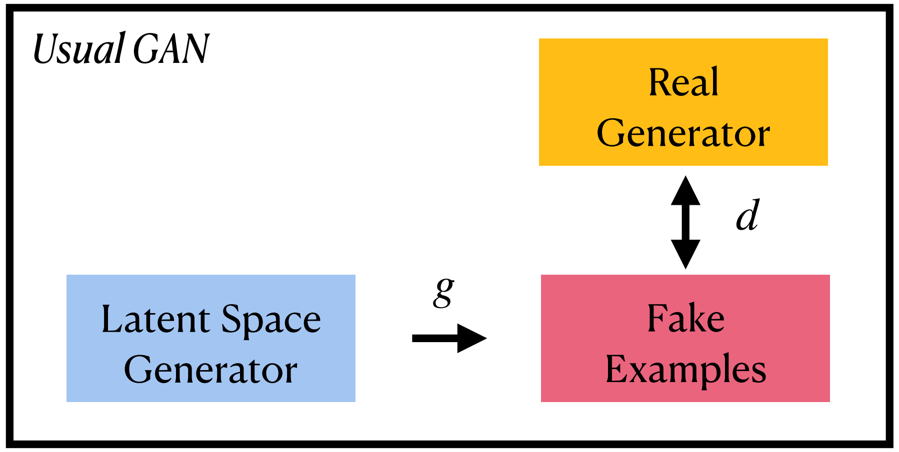


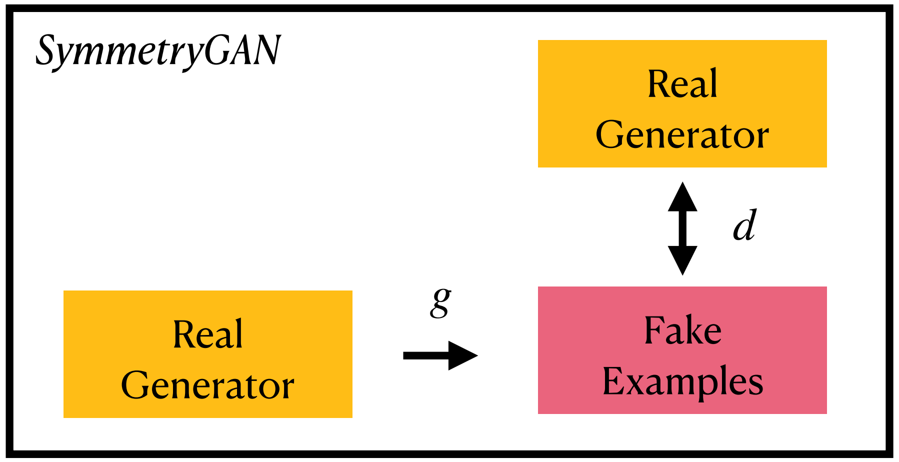


See also 1911.09107 ("OmniFold") and 2101.08944 ("OTUS")

Bonus: What else can we do?

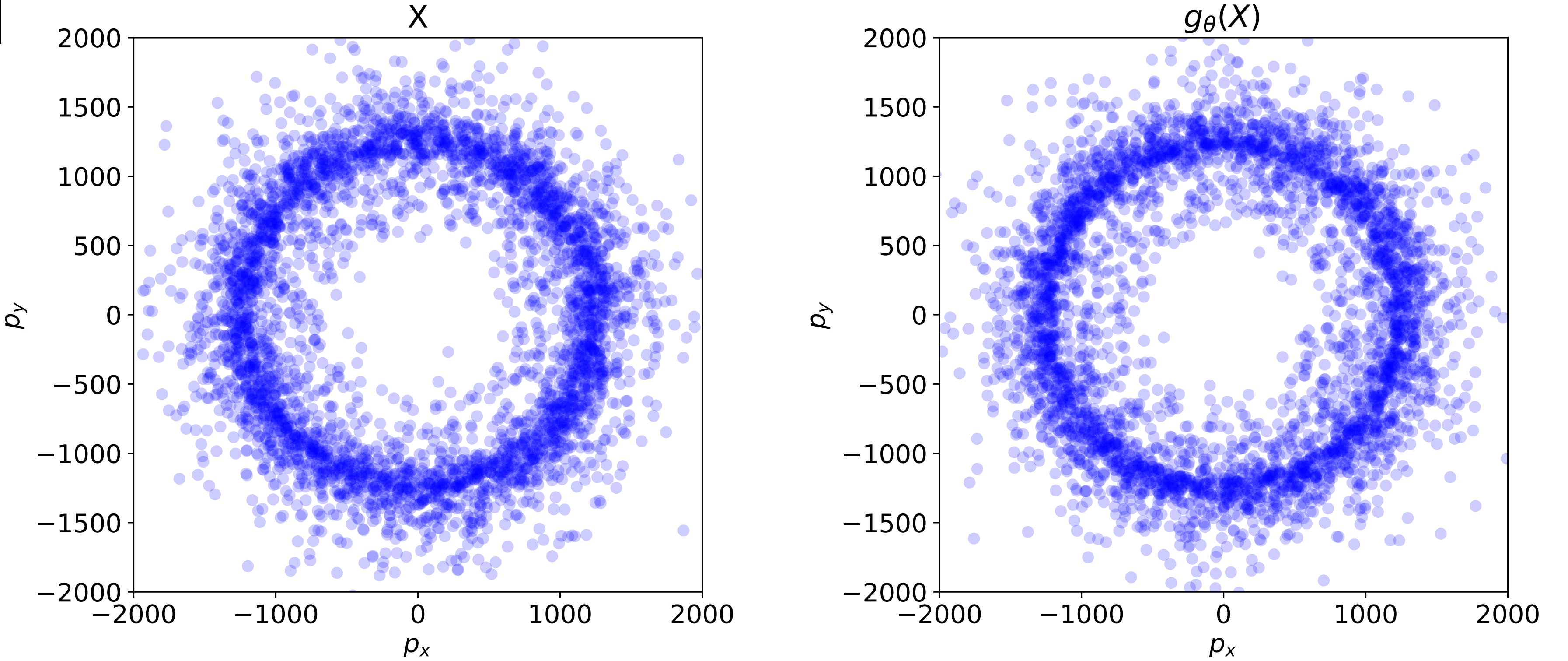
ex. GANs: Symmetry Discovery



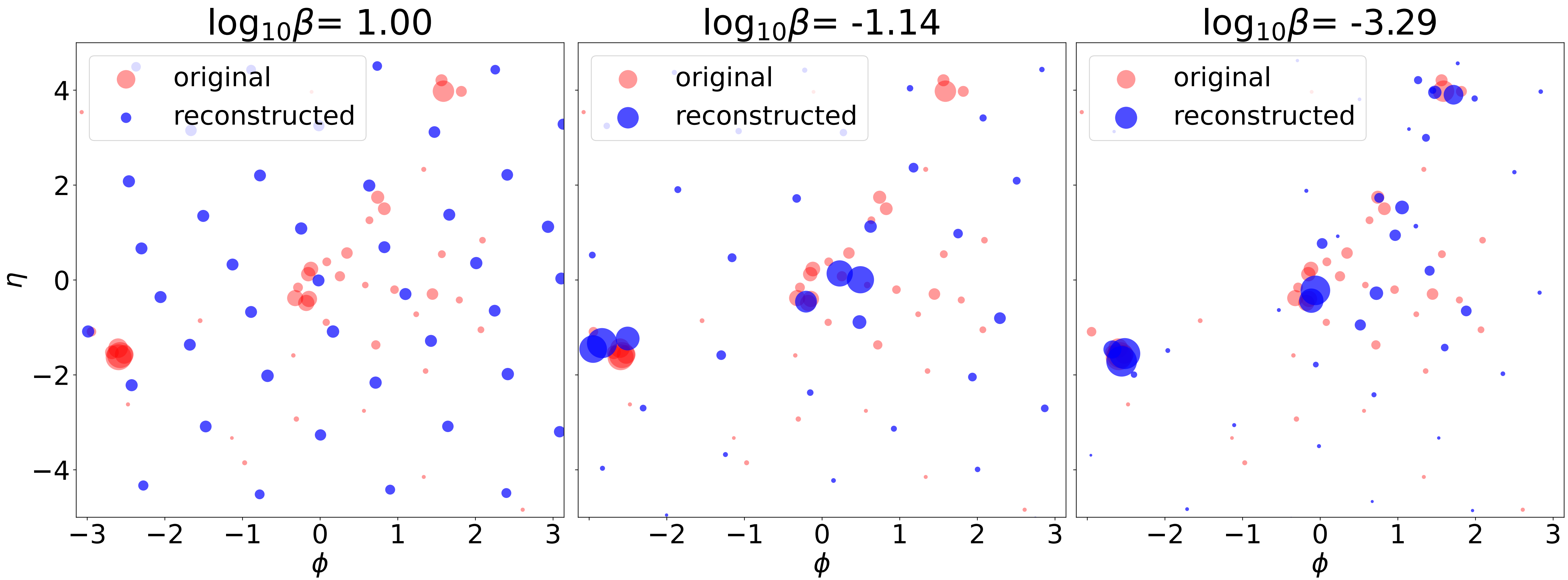


The framework of generative models is quite flexible and we can do more than generate events.

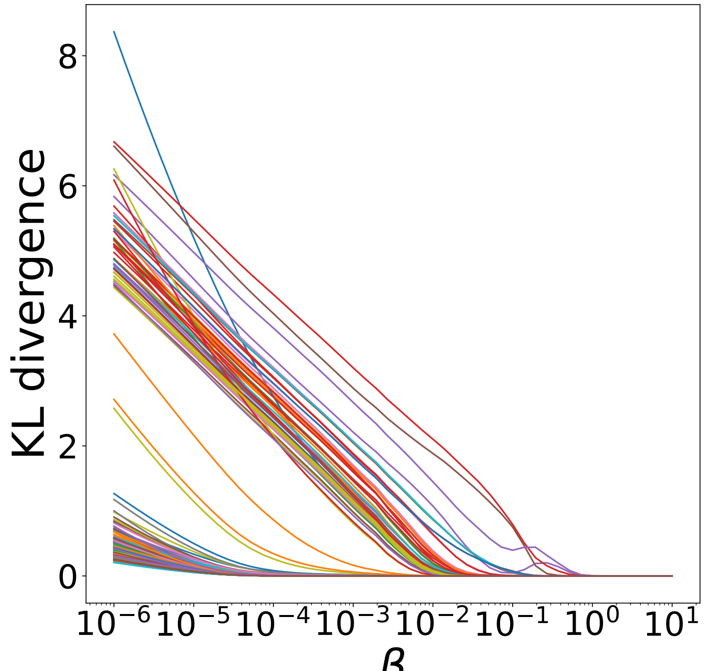
For example, can **discover symmetries** in data!



ex. VAEs: Compression

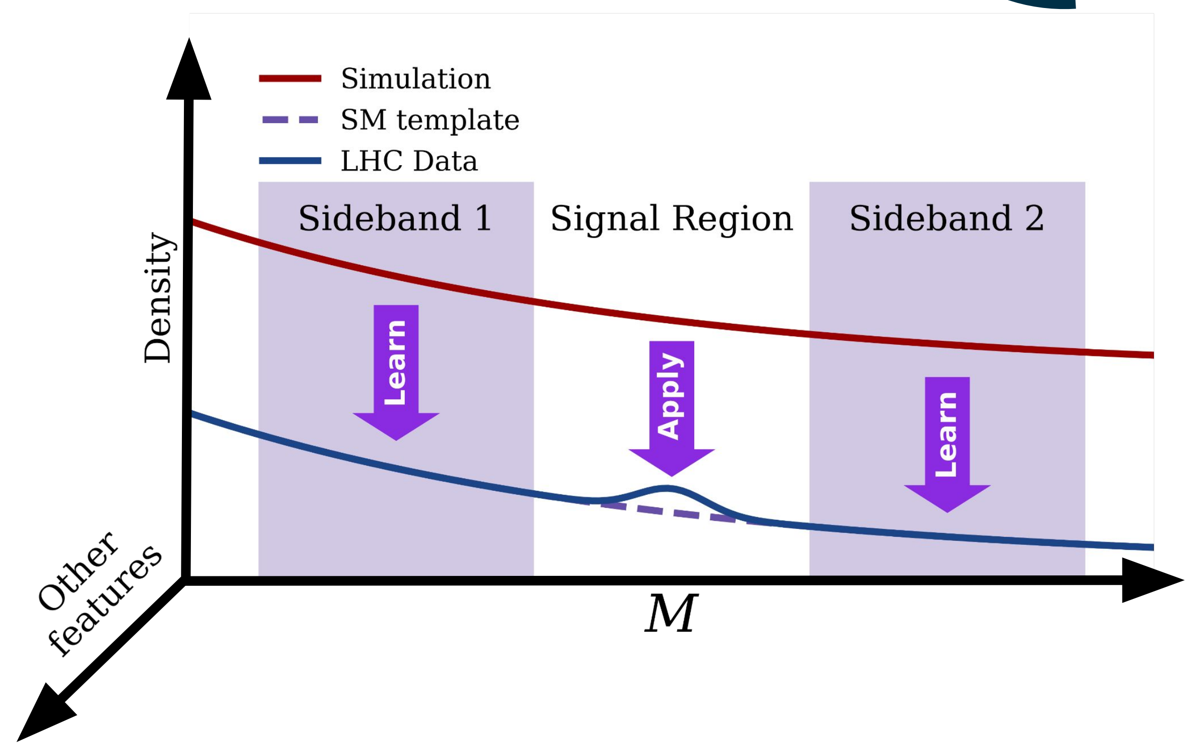


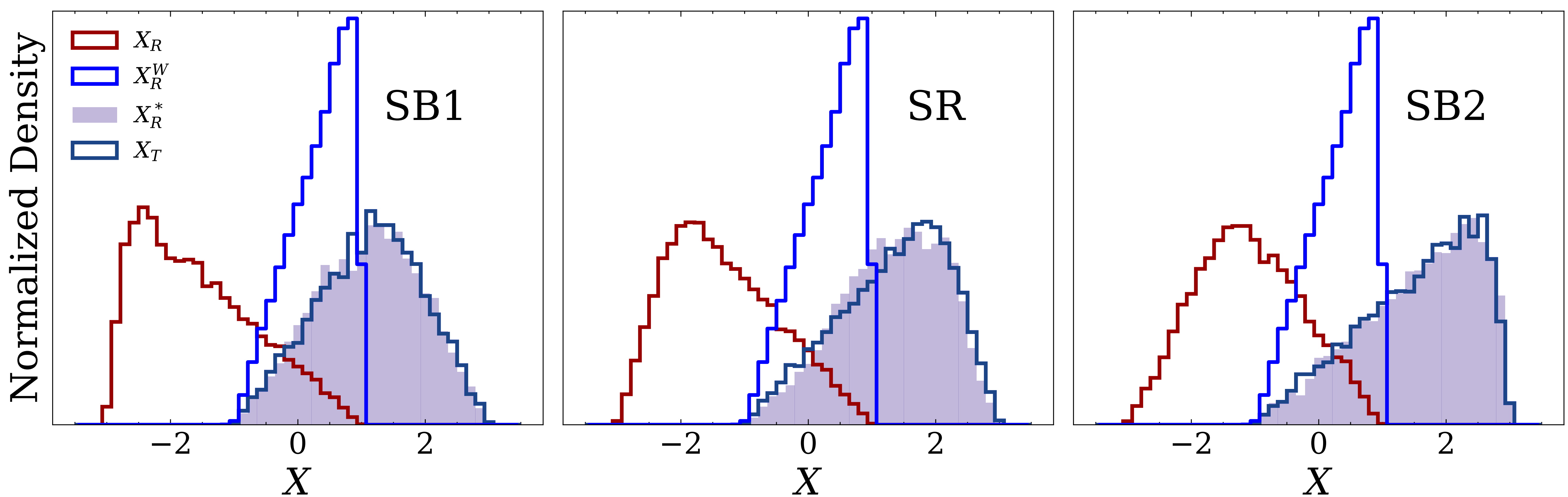
Automatically vary the compression level by varying β



ex. NFs: Data Morphing

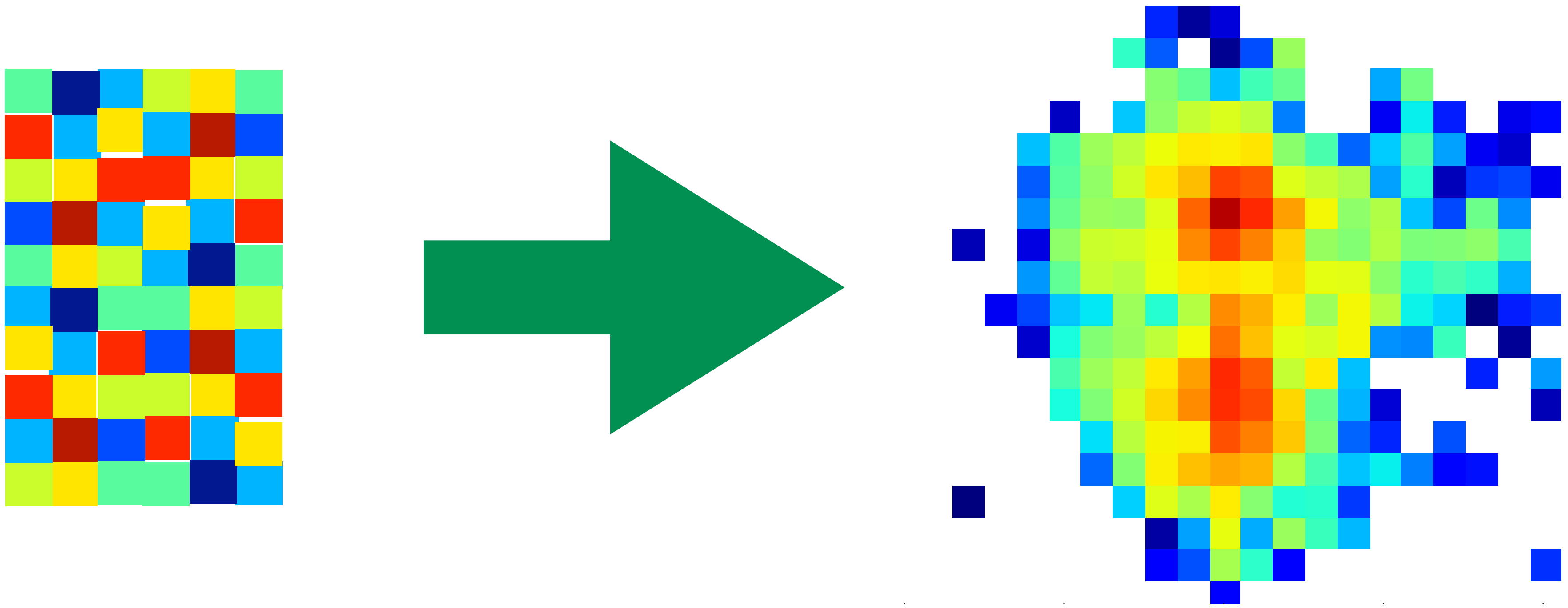
"Move" instead of "Reweight" with normalizing flows





Outline

1. Why Generative Models?
2. Warm up: mixture density networks
3. Next: brief overview of GANs, VAEs, NFs
4. Statistical amplification
5. Applications
6. Bonus



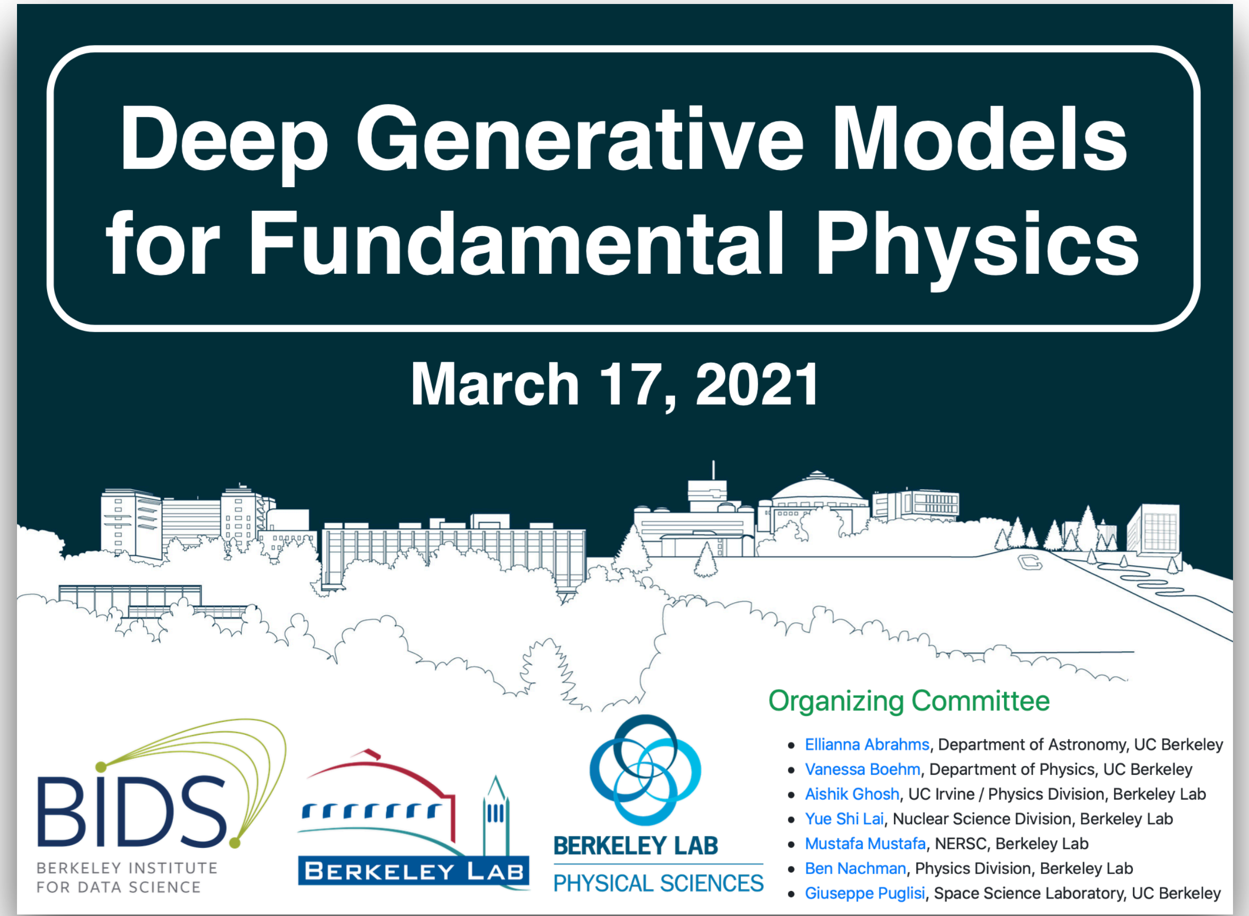
Conclusions and Outlook

Generative models hold great promise for many areas of physics research

This is an area of rapid development at the intersection of theory & experiment and I'm excited for where it will take us!

Examples today were not exhaustive … Prof. Duarte has also been a pioneer in this area!

This is a link to a recent Berkeley workshop dedicated to generative models →



Questions?