Overview of Classical Machine Learning in HEP

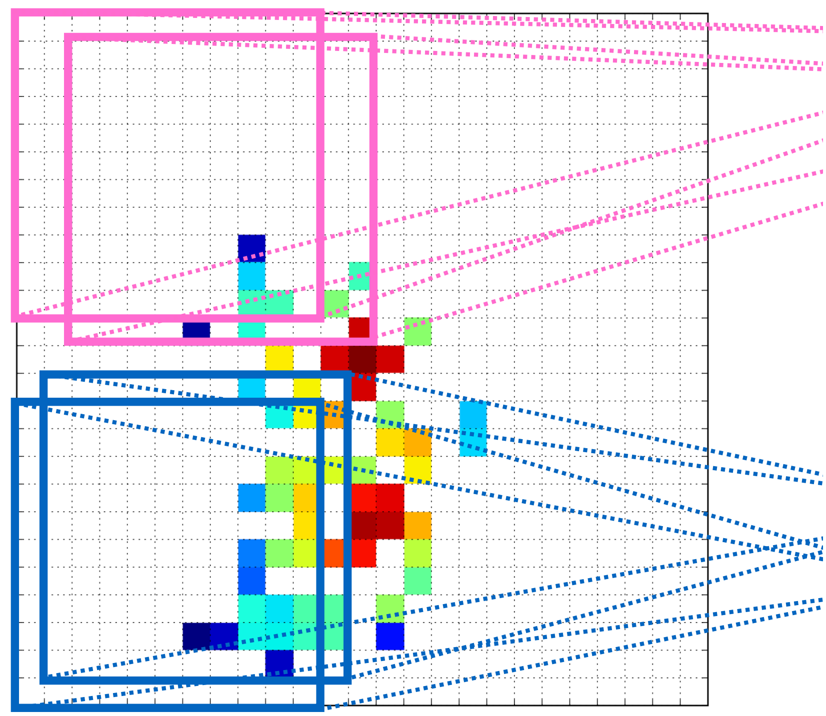
Benjamin Nachman

Lawrence Berkeley National Laboratory

bpnachman.com

bpnachman@lbl.gov

@bpnachman bnachman



BERKELEY LAB

BIDS BERKELEY INSTITUTE FOR DATA SCIENCE

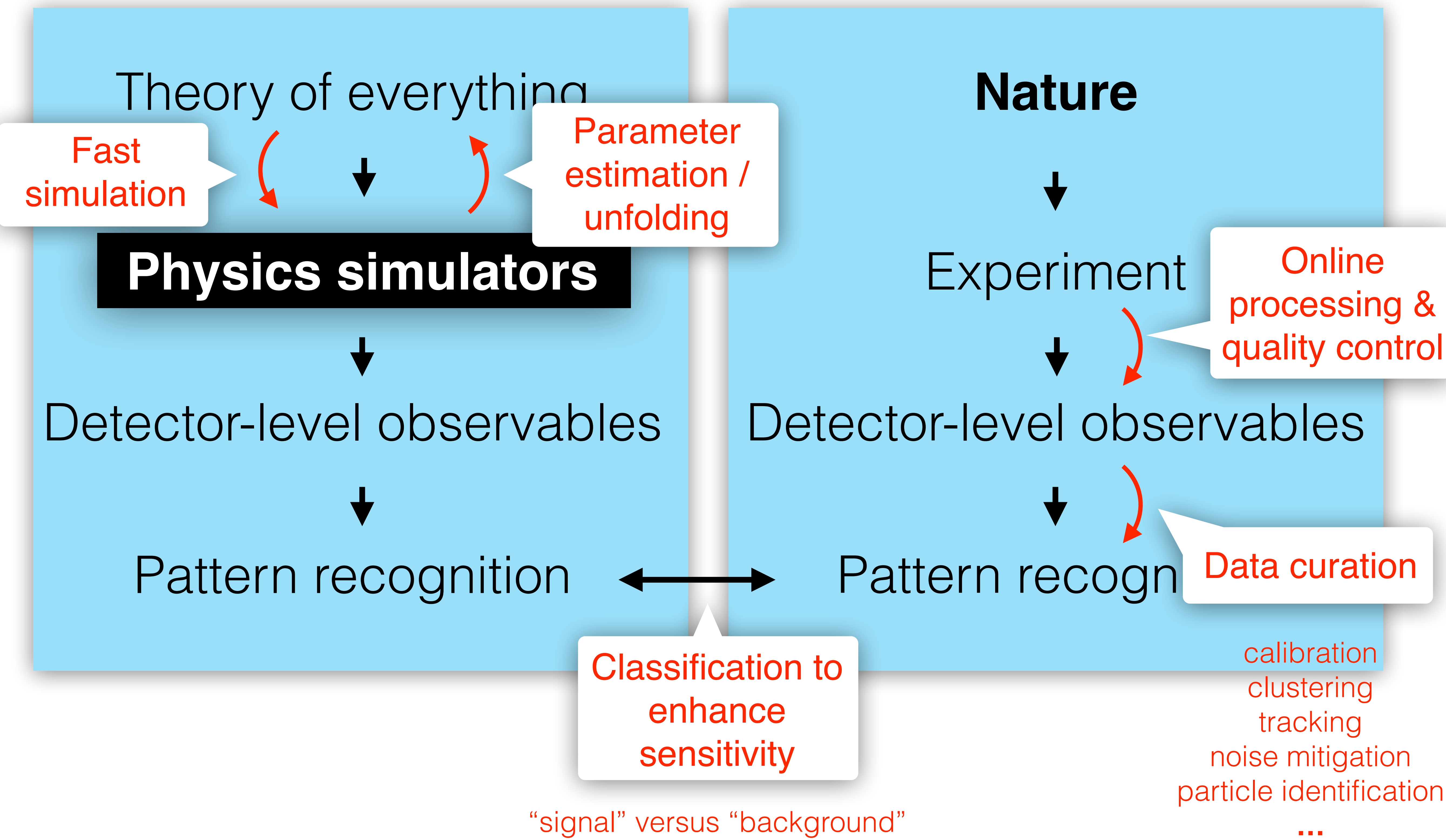
QC/QML for HEP

June 2022

Data analysis in HEP

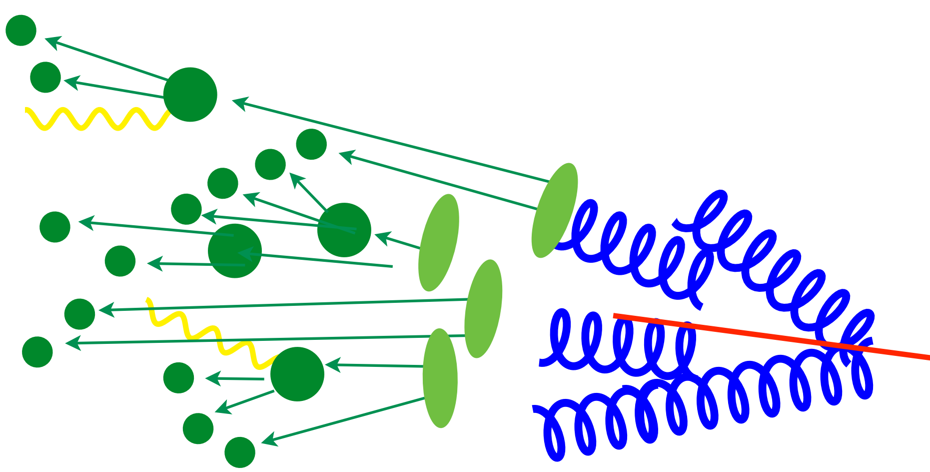
Theory of everything	**Nature**
↓	↓
Physics simulators	Experiment
↓	↓
Detector-level observables	Detector-level observables
↓	↓
Pattern recognition ⟷	Pattern recognition

Data analysis in HEP + Machine Learning



Outline

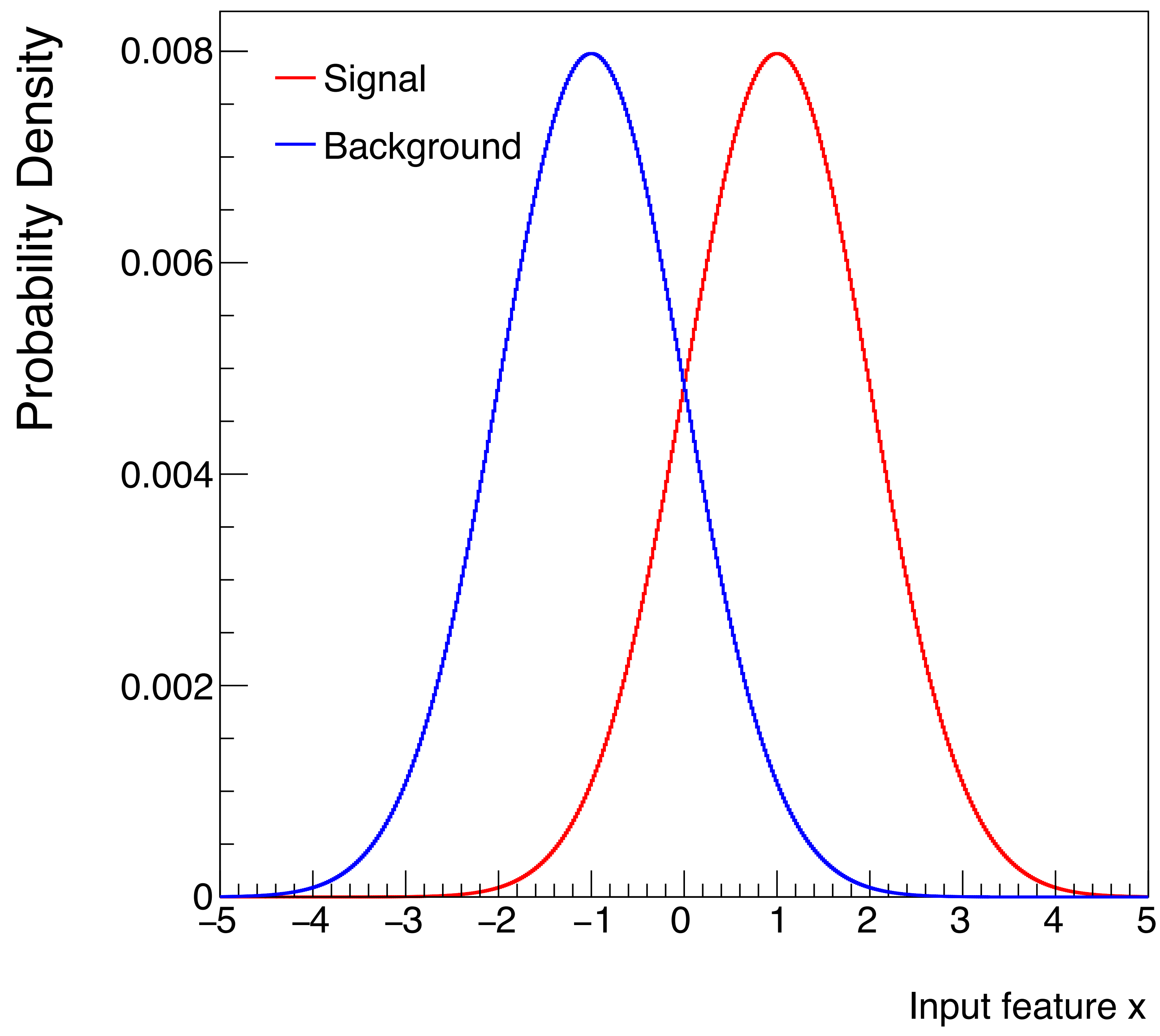
- Classification
  - Machine Learning and Optimality
  - HEP images
  - Other architectures for HEP



- Regression, Generative Models / likelihood-free approaches, Anomaly Detection

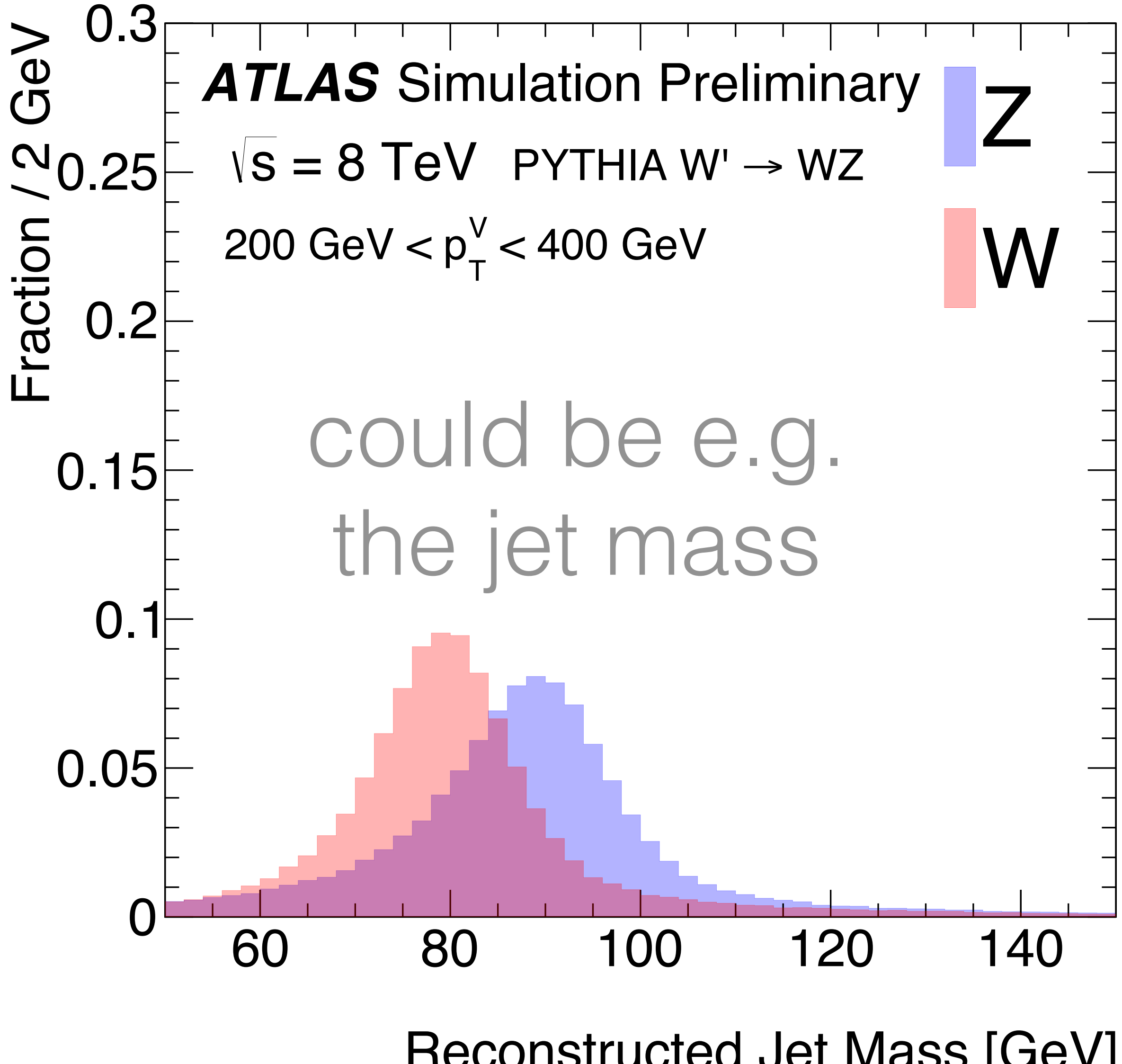
Machine learning and optimality

Let's consider an important special case:
binary classification in 1D

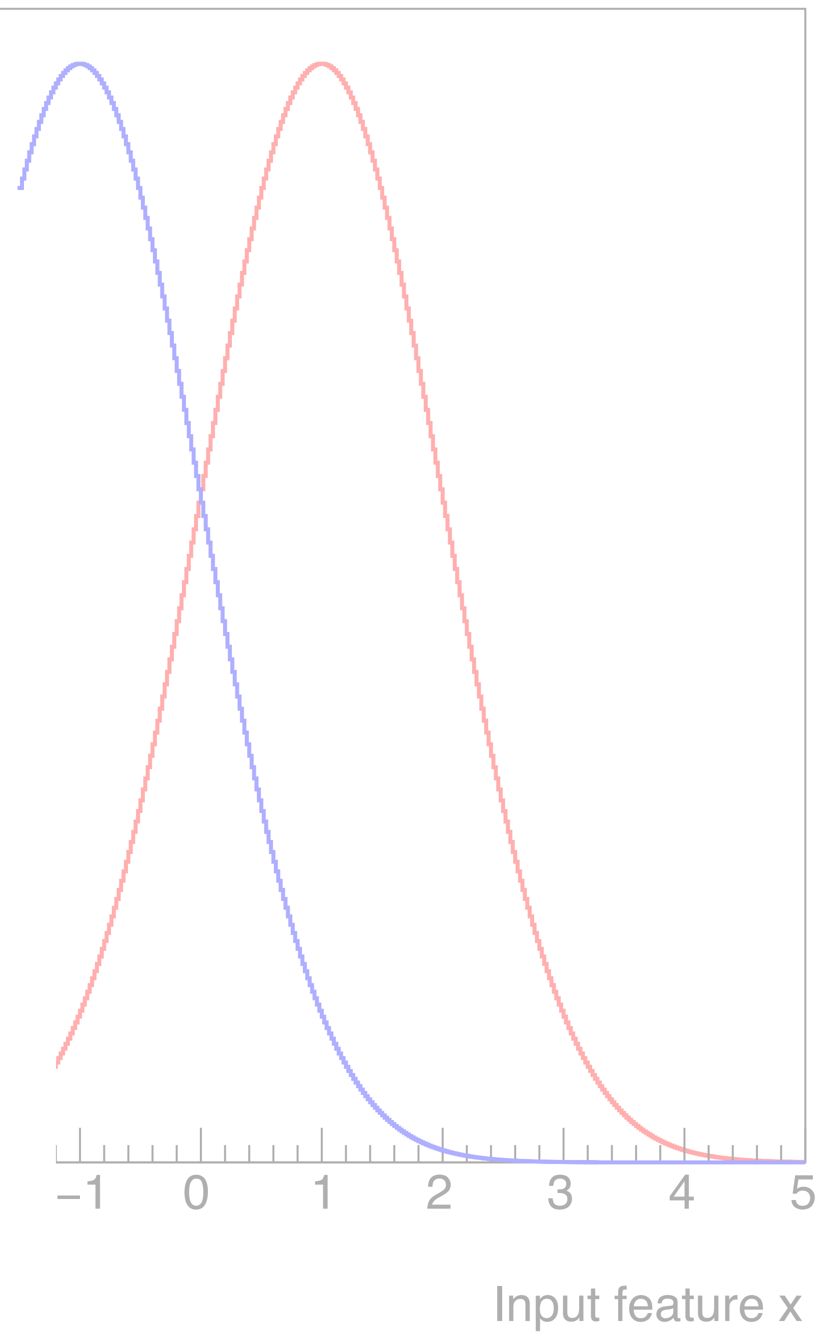


Machine learning and optimality

Let's consider an important special case:
binary classification in 1D

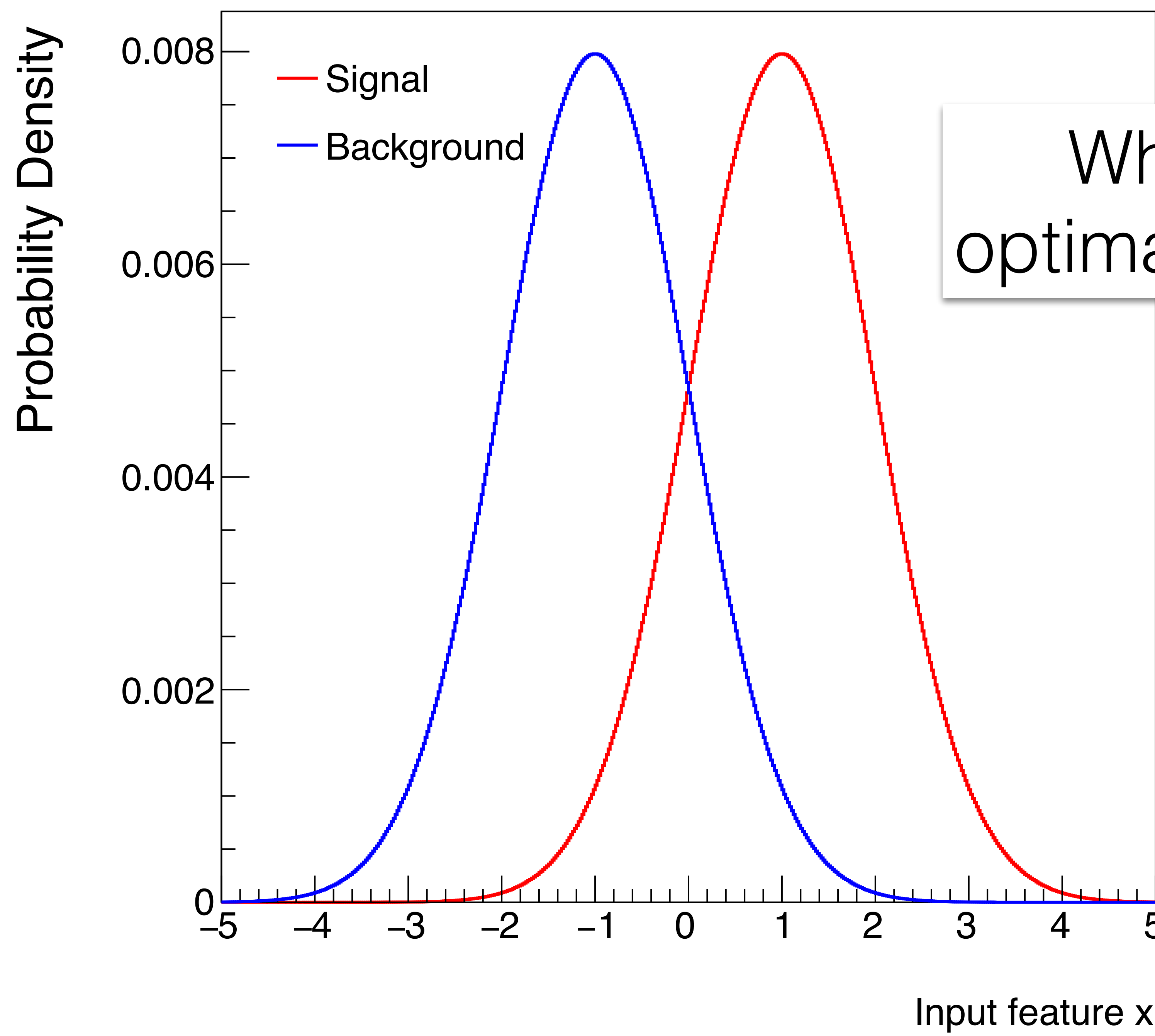


could be e.g. the jet mass



Machine learning and optimality

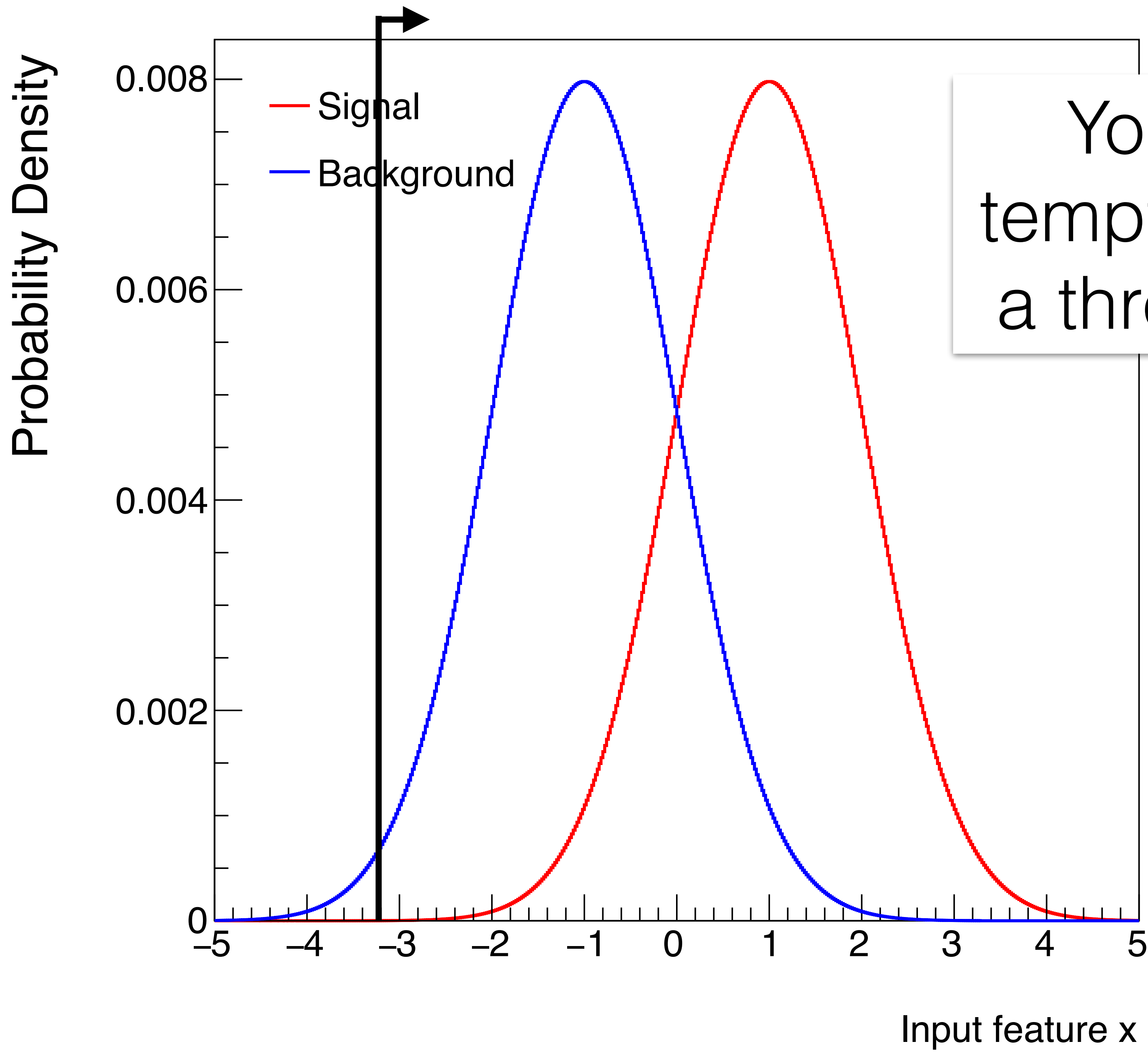
Let's consider an important special case:
binary classification in 1D



What is the optimal classifier?

Machine learning and optimality

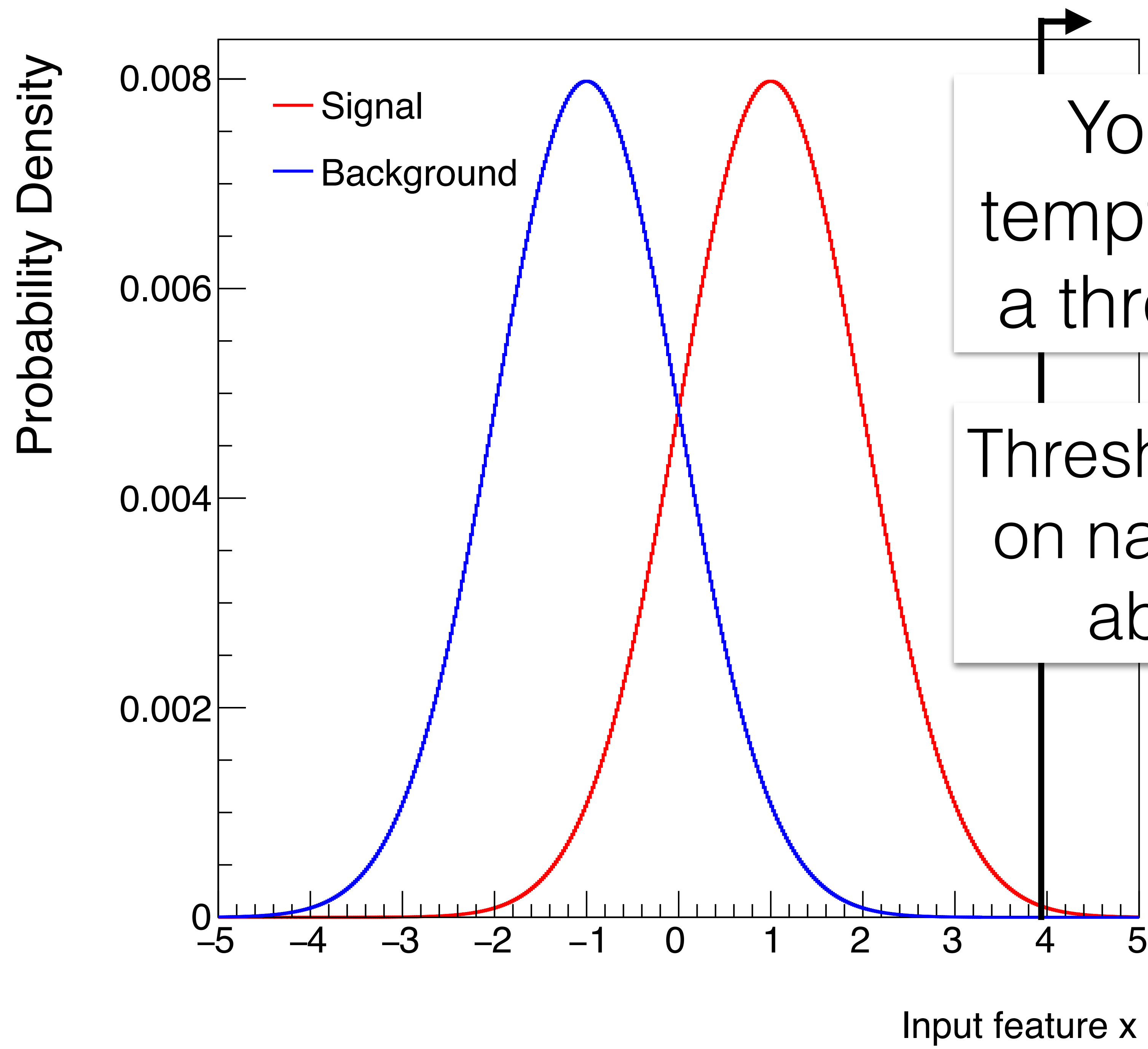
Let's consider an important special case:
binary classification in 1D



You may be tempted to place a threshold on x

Machine learning and optimality

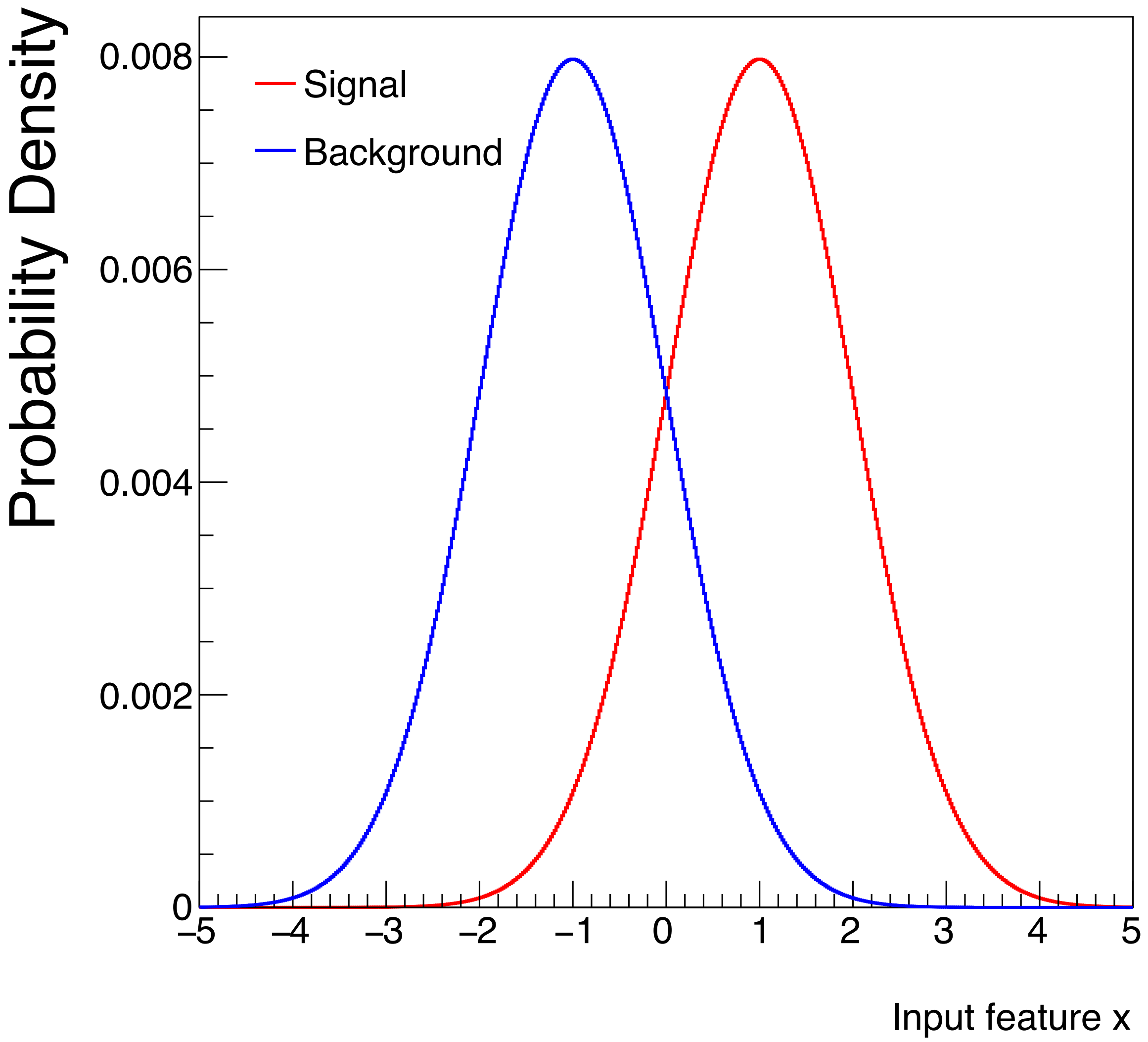
Let's consider an important special case: binary classification in 1D



You may be tempted to place a threshold on x

Threshold depends on natural relative abundance

Machine learning and optimality



Is the simple threshold cut **optimal**?

TPR = true positive rate or "signal efficiency"

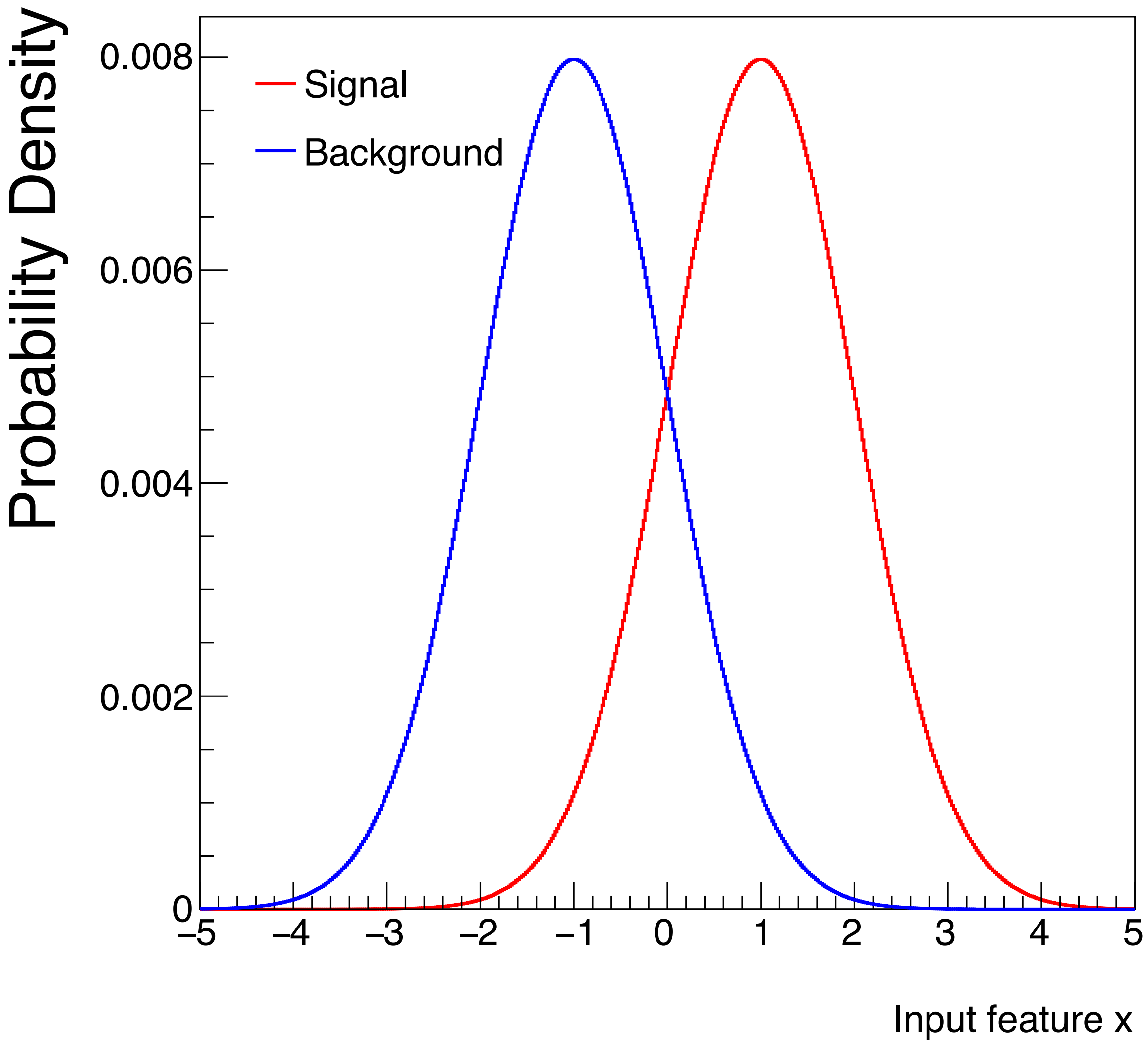
FPR = false positive rate or 1 - "background rejection"

See Neyman-Pearson lemma

Fact 1: The classifier that results in the lowest FPR for a given TPR is a cut on the **likelihood ratio (LR)**.

$$LR(x) > c,\ LR(x) = p(x|\text{signal}) \,/\, p(x|\text{background})$$

Machine learning and optimality



Is the simple threshold cut **optimal**?

TPR = true positive rate or "signal efficiency"

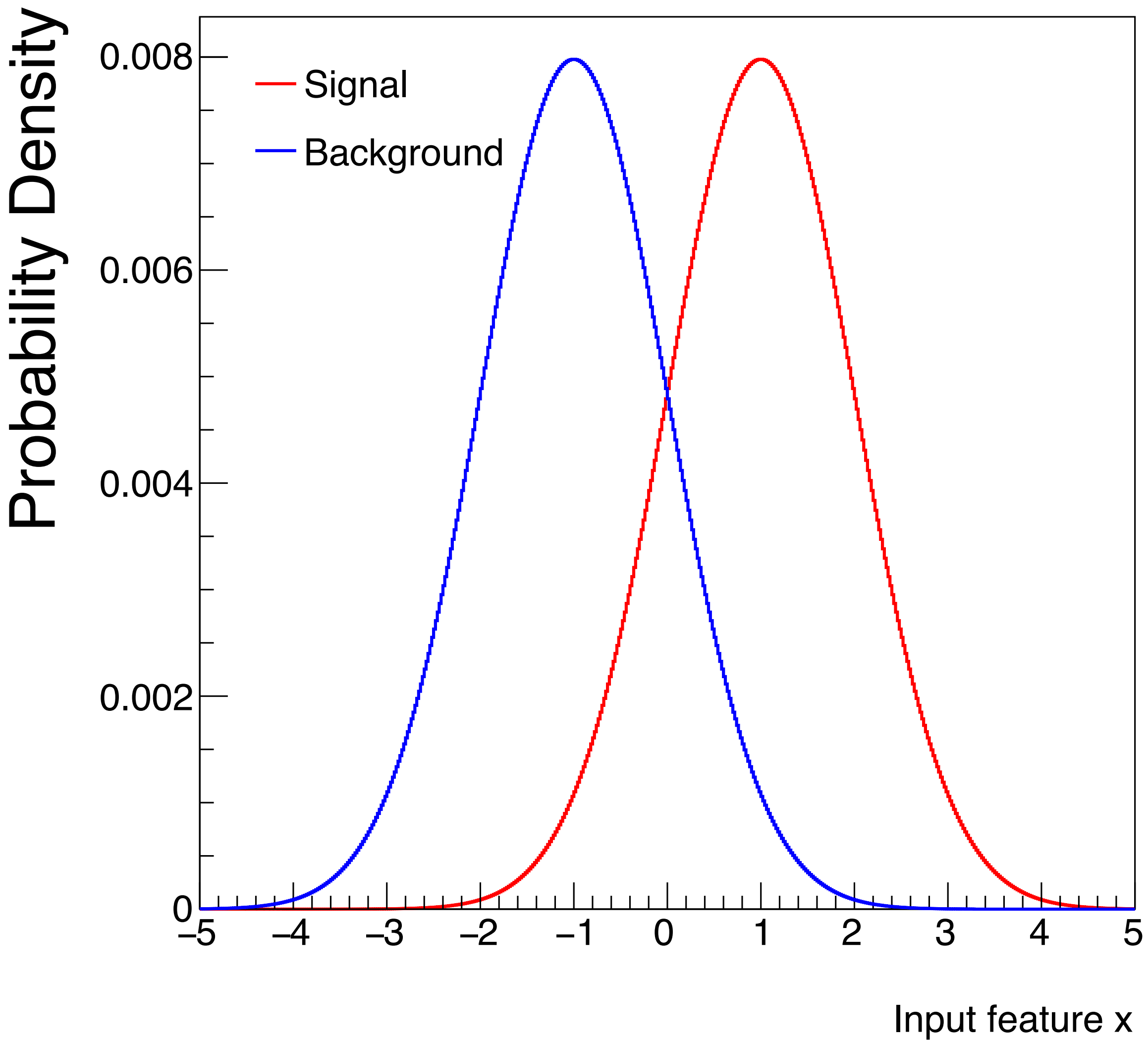
FPR = false positive rate or 1 - "background rejection"

See Neyman-Pearson lemma

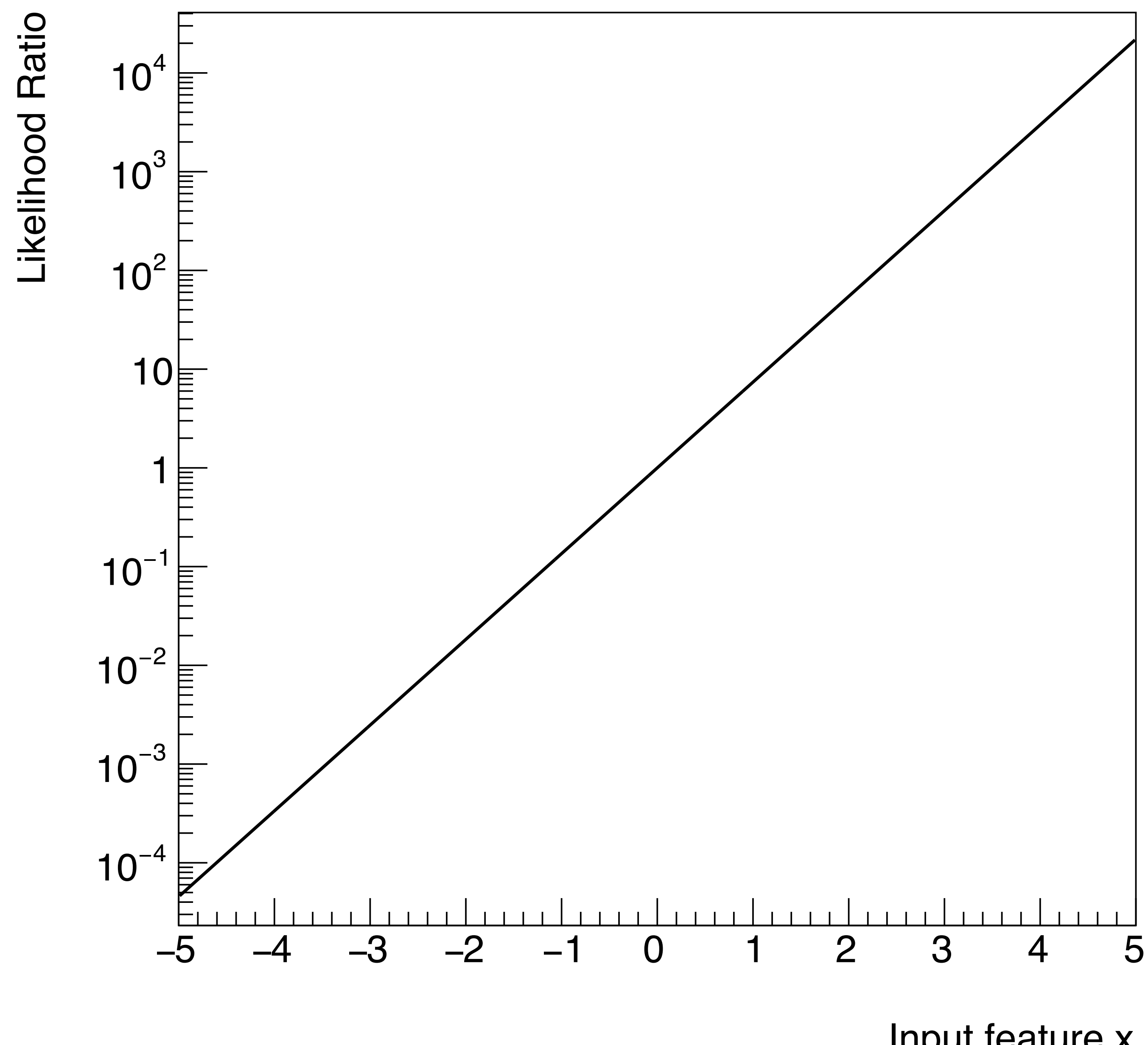
Fact 1: The classifier that results in the lowest FPR for a given TPR is a cut on the **likelihood ratio (LR)**.

Fact 2: Two classifiers that are related by a **monotonic transformation** result in the same performance.

Machine learning and optimality



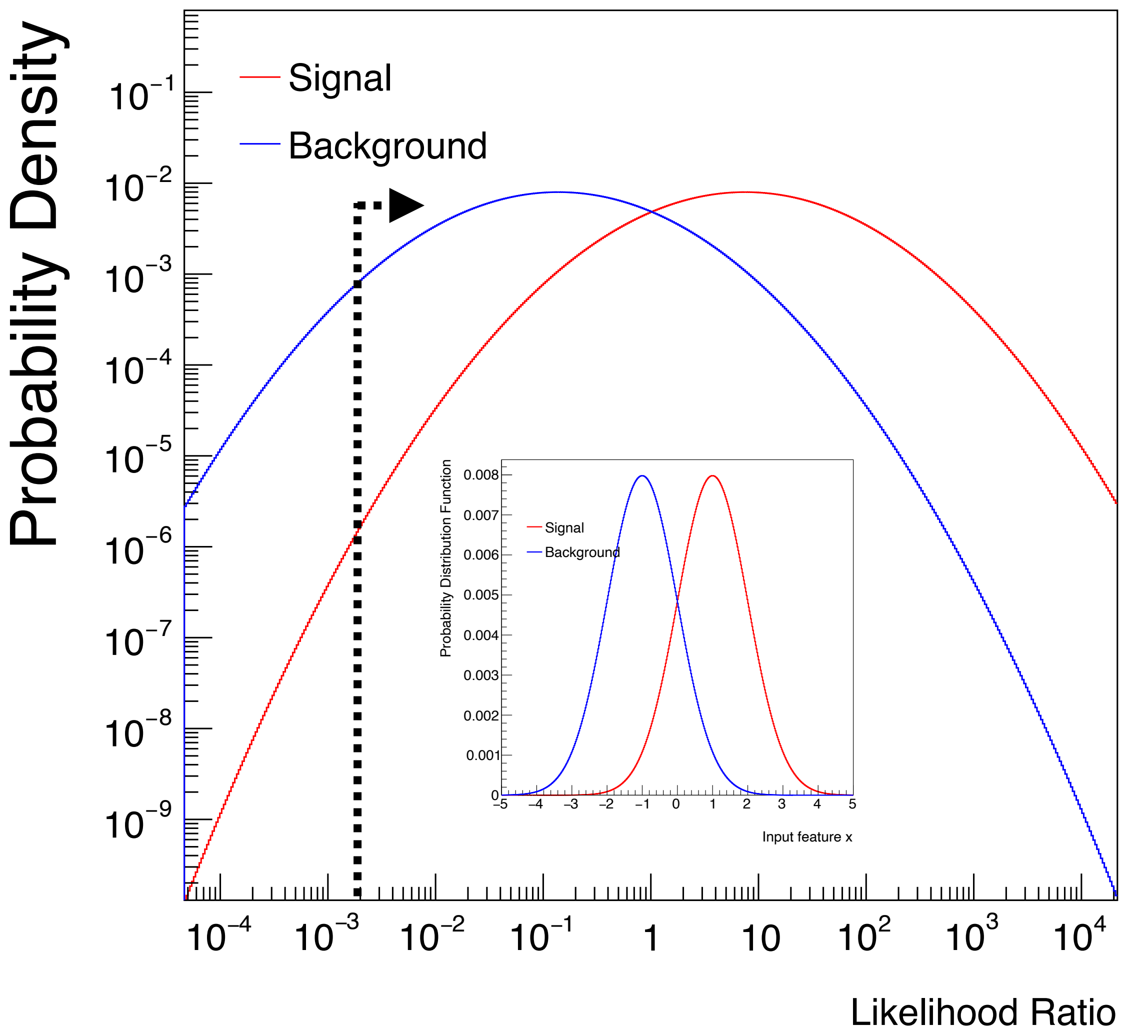
Is the simple threshold cut **optimal**?



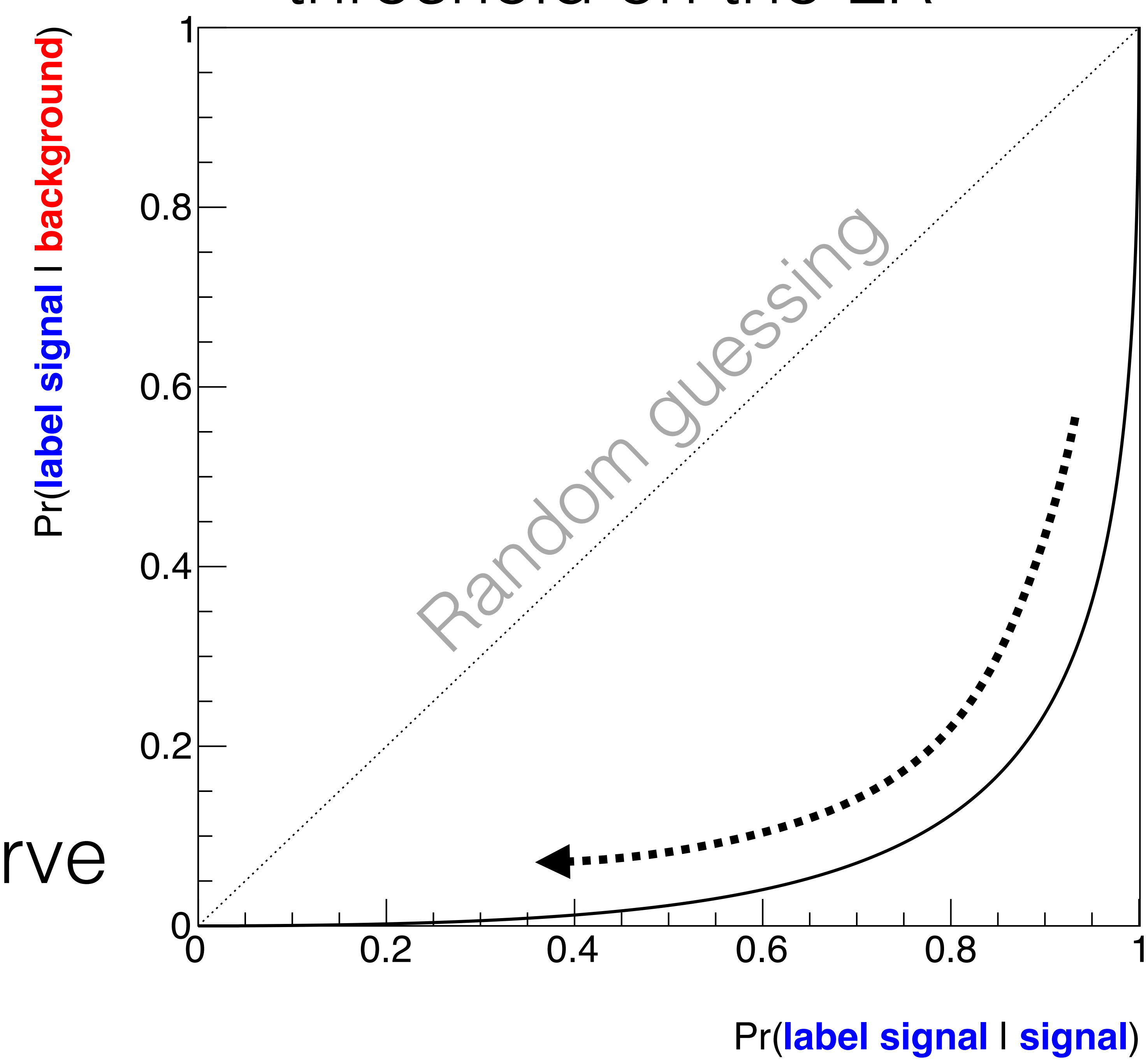
In this simple case, the log LR is proportional to x: **no need for non-linearities**!

Threshold cut is optimal

Machine learning and optimality



The optimal procedure is a threshold on the LR

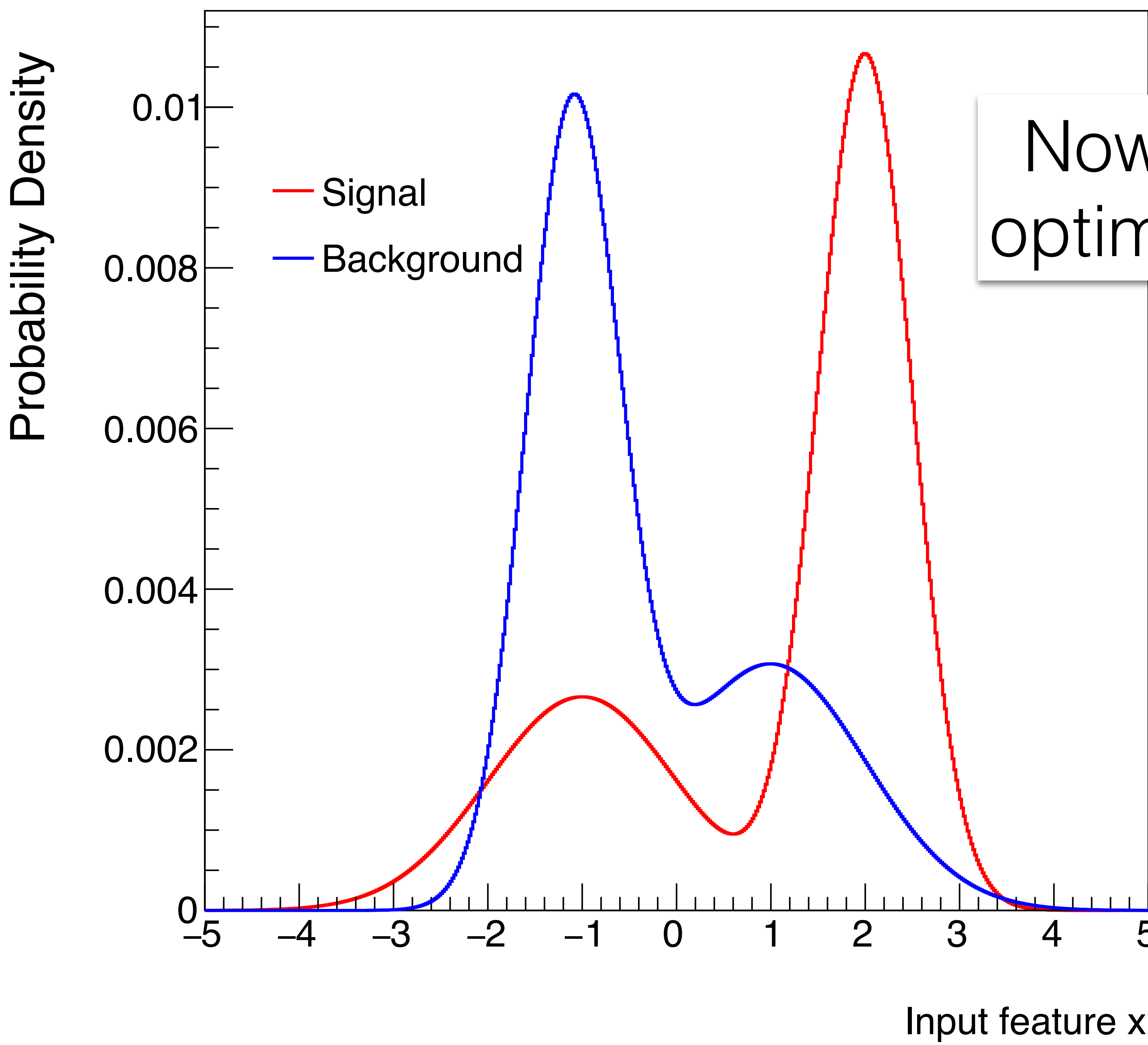


"Receiver Operating Characteristic" (**ROC**) Curve

Machine learning and optimality

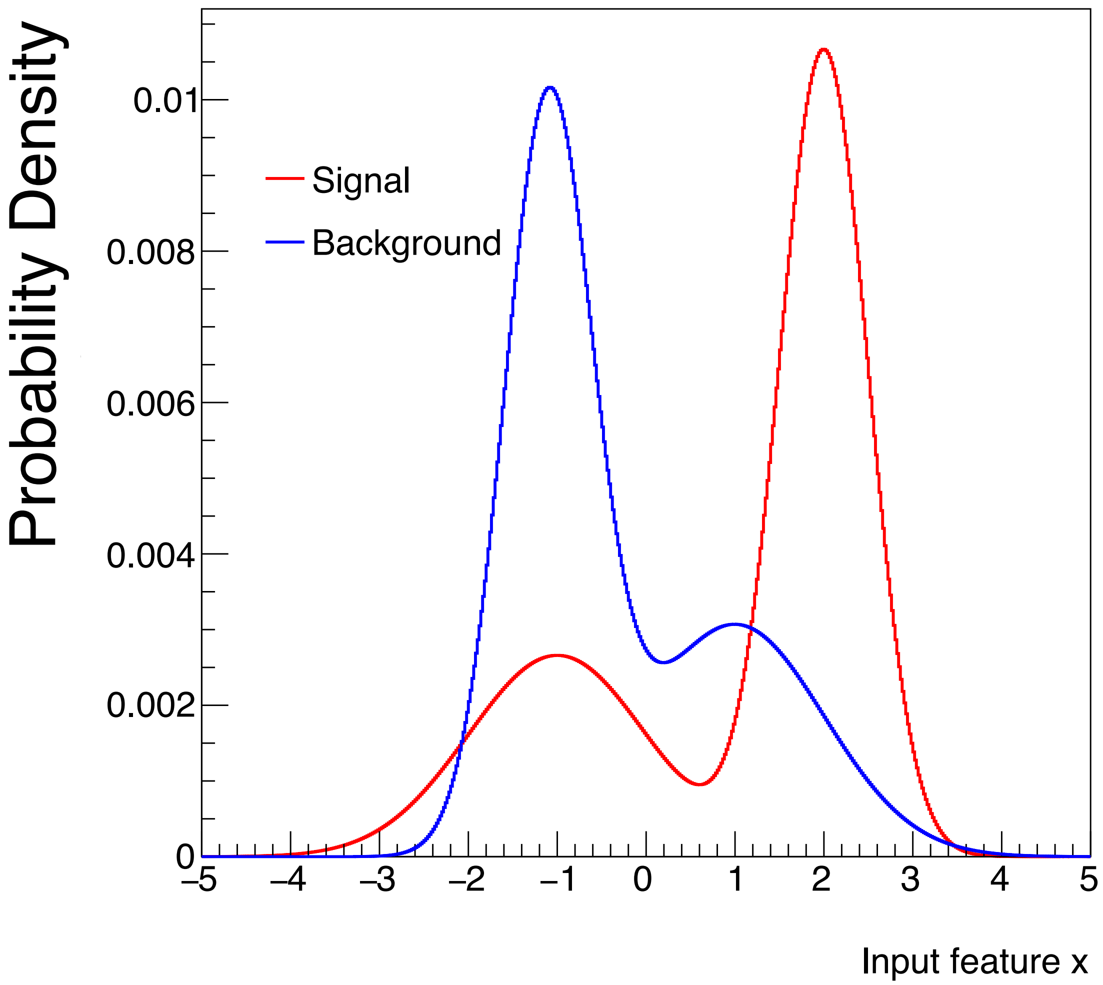
What if the distribution of x is complicated?

Real life is complicated!

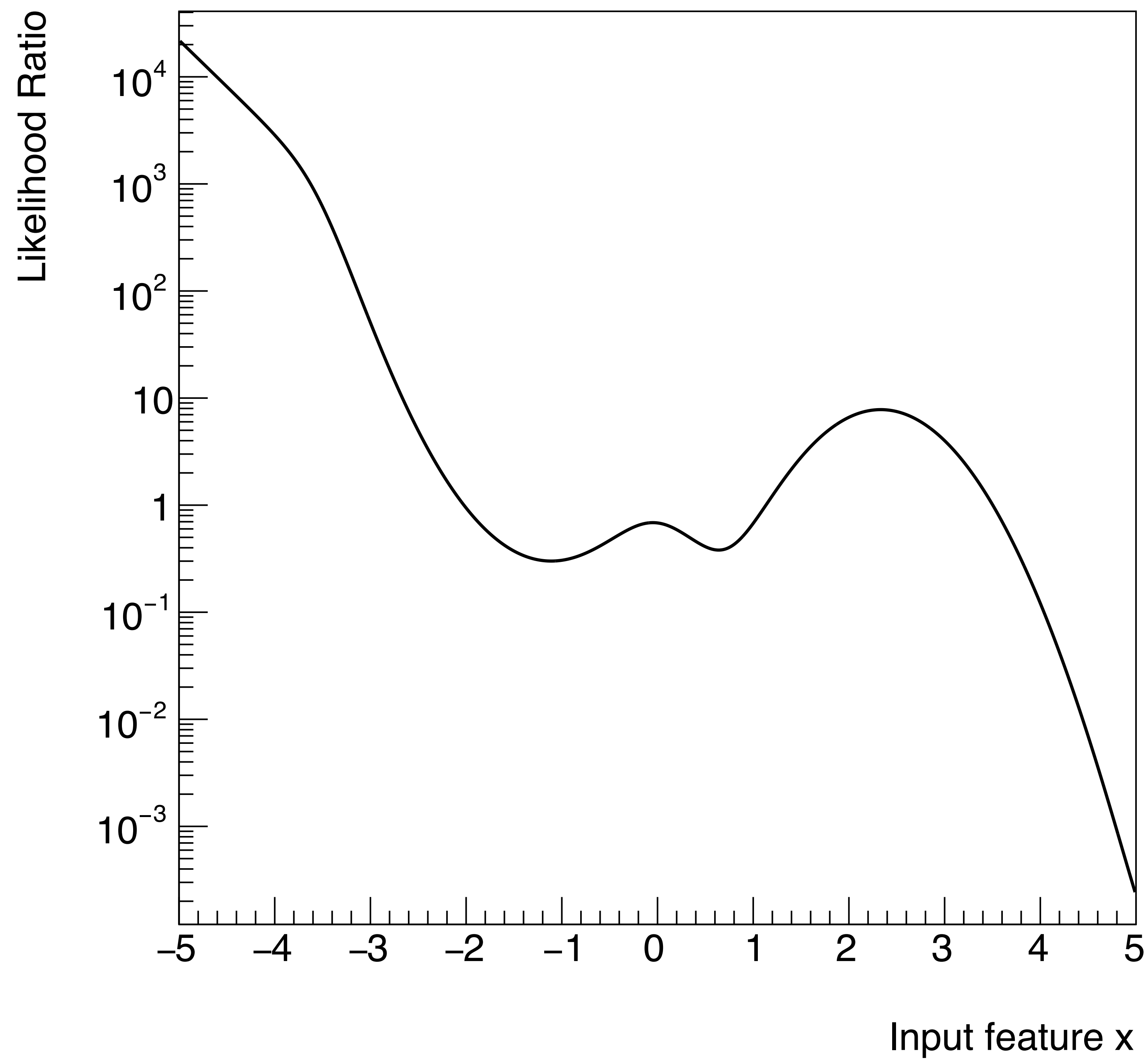


Now what is the optimal classifier?

Machine learning and optimality

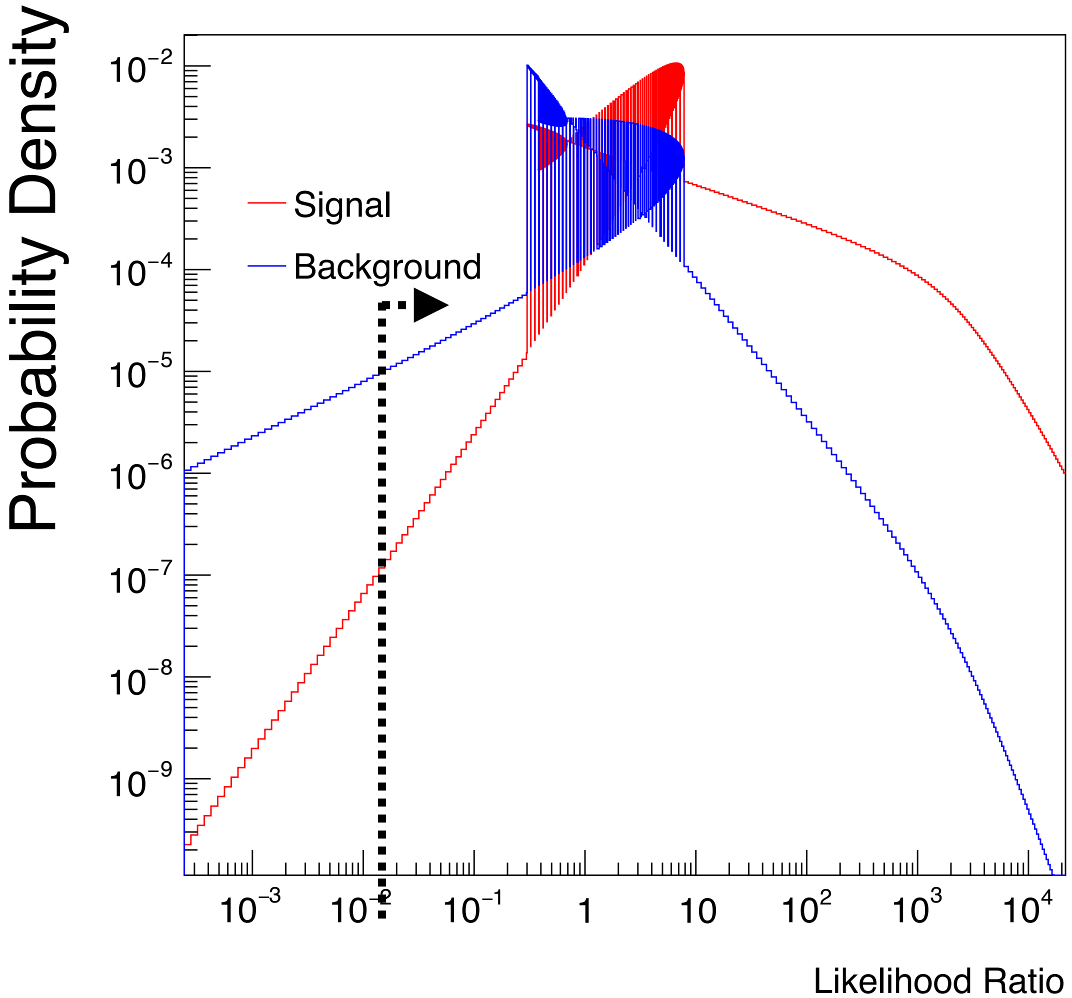


In this case, LR is highly non-linear (**non-monotonic**) function of x

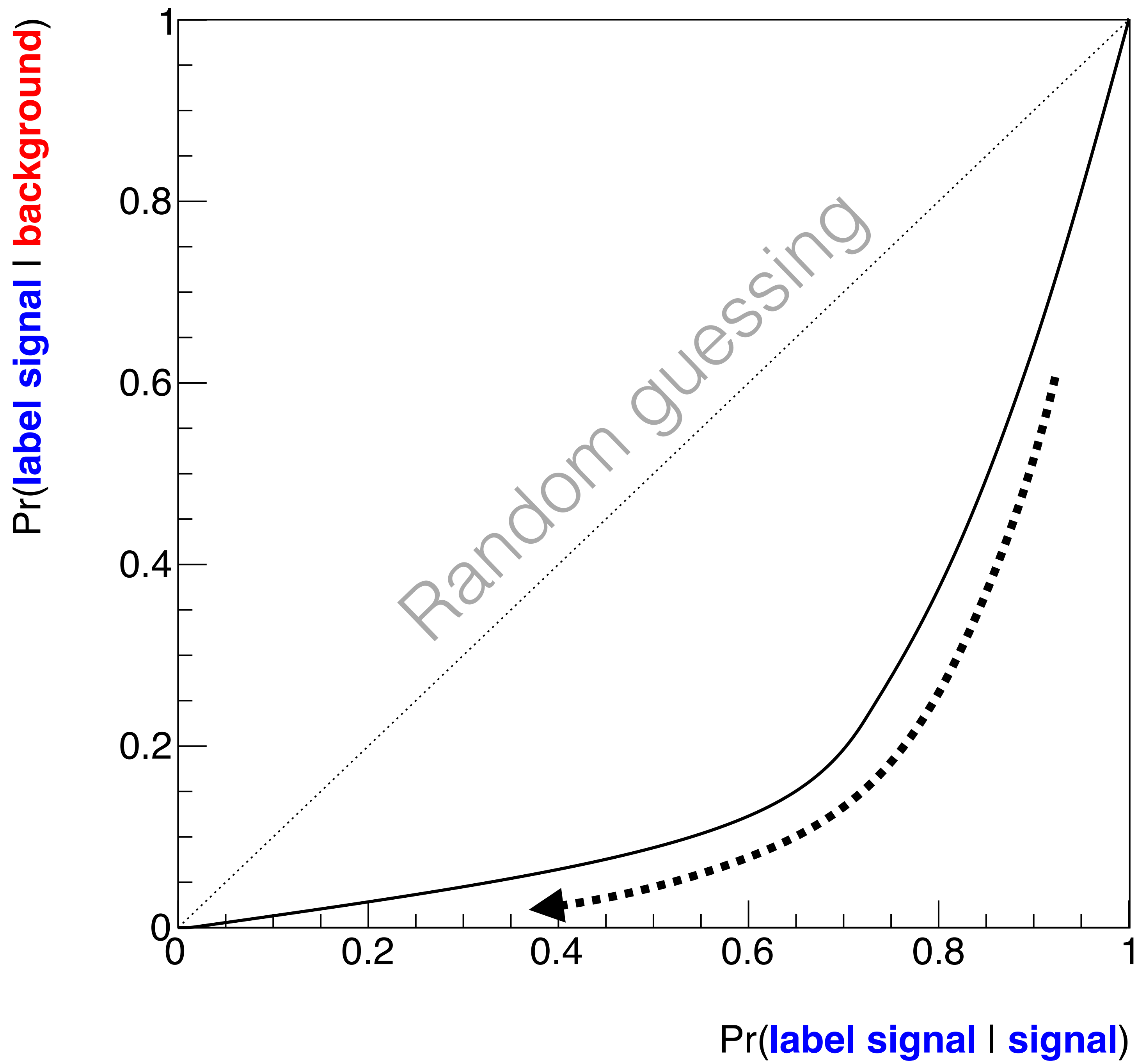


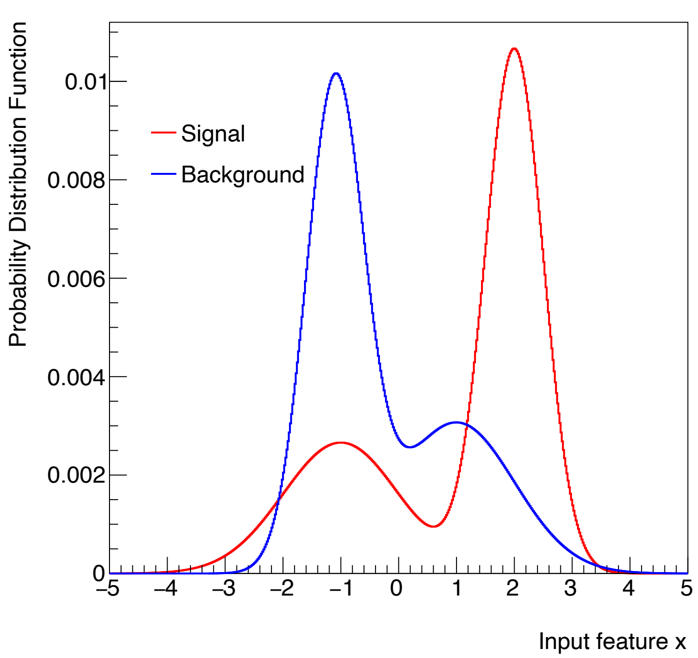
A threshold on x would be sub-optimal

Machine learning and optimality



ROC worse than the Gaussians, as expected since more PDF overlap





Tools for Classification

Why don't we always just compute the optimal classifier?

In the last slides, we had to estimate the likelihood ratio - this required binning the PDF

binning works in 1D, but intractable as feature dimension >> 1 ("curse of dimensionality")

machine learning for classification is simply **the art of estimating the likelihood ratio with limited training examples**

HEP Tools for (Classical) Classification

= tools for likelihood ratio estimation

- "Histograming"
- Nearest Neighbors
- Support Vector Machines (SVM)
- (Boosted) Decision Trees
- (Deep) Neural Networks
- …

Not widely used; only useful if decision boundary is 'simple'

has most things and ROOT-compatible but the community base is **much** smaller than the other ones

Software: TMVA, scikit-learn, XGBoost, tensorflow, pytorch…

does "everything" except DNNs

Data formats: .root, .npy, .hdf5

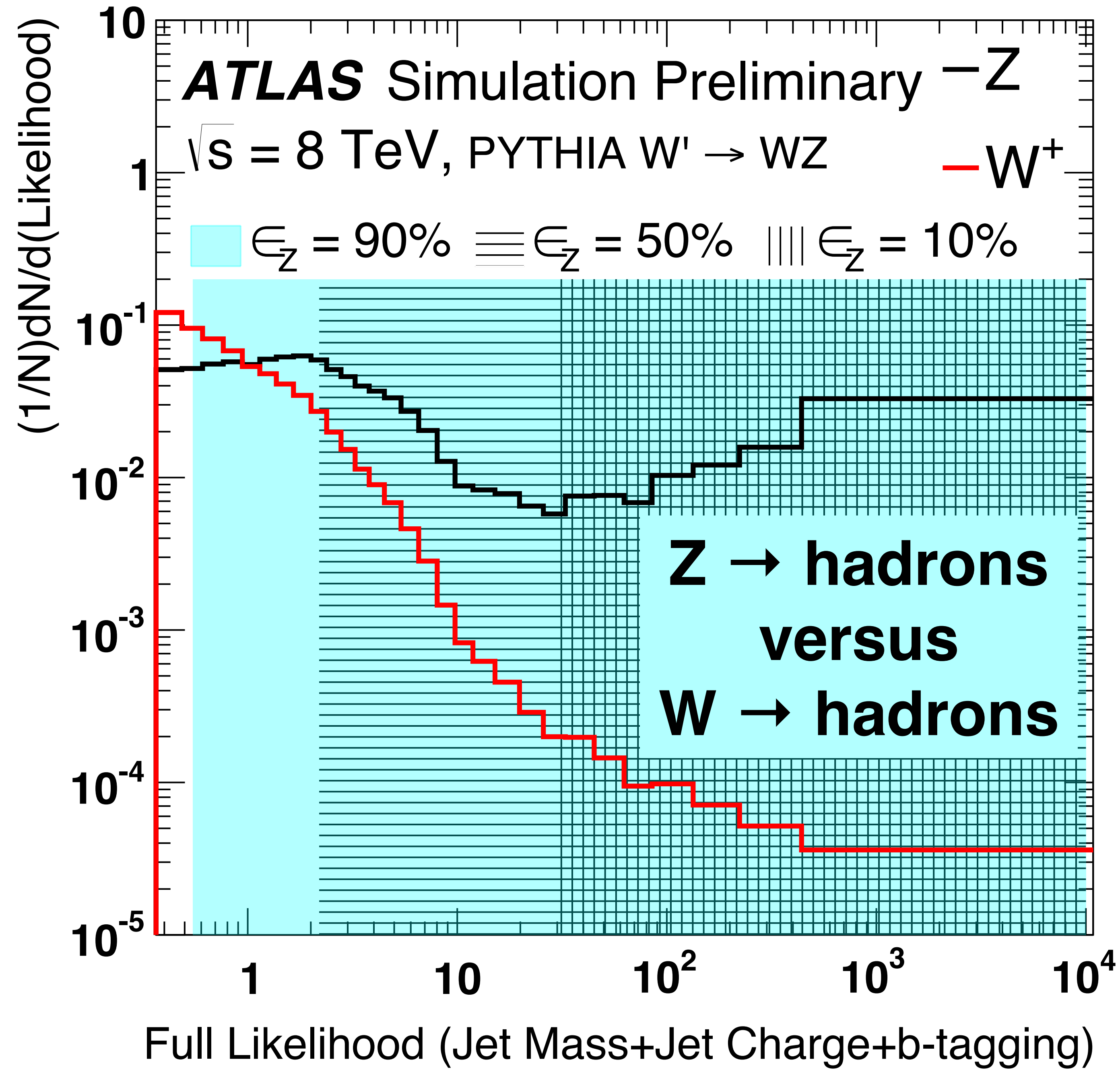
Histogramming

Eur. Phys. J. C76 (2016) 238

If you have a 1D problem, look no further!

If your problem can be decomposed into a product/sum of 1D problems…look no further!

If these do not apply… look elsewhere.

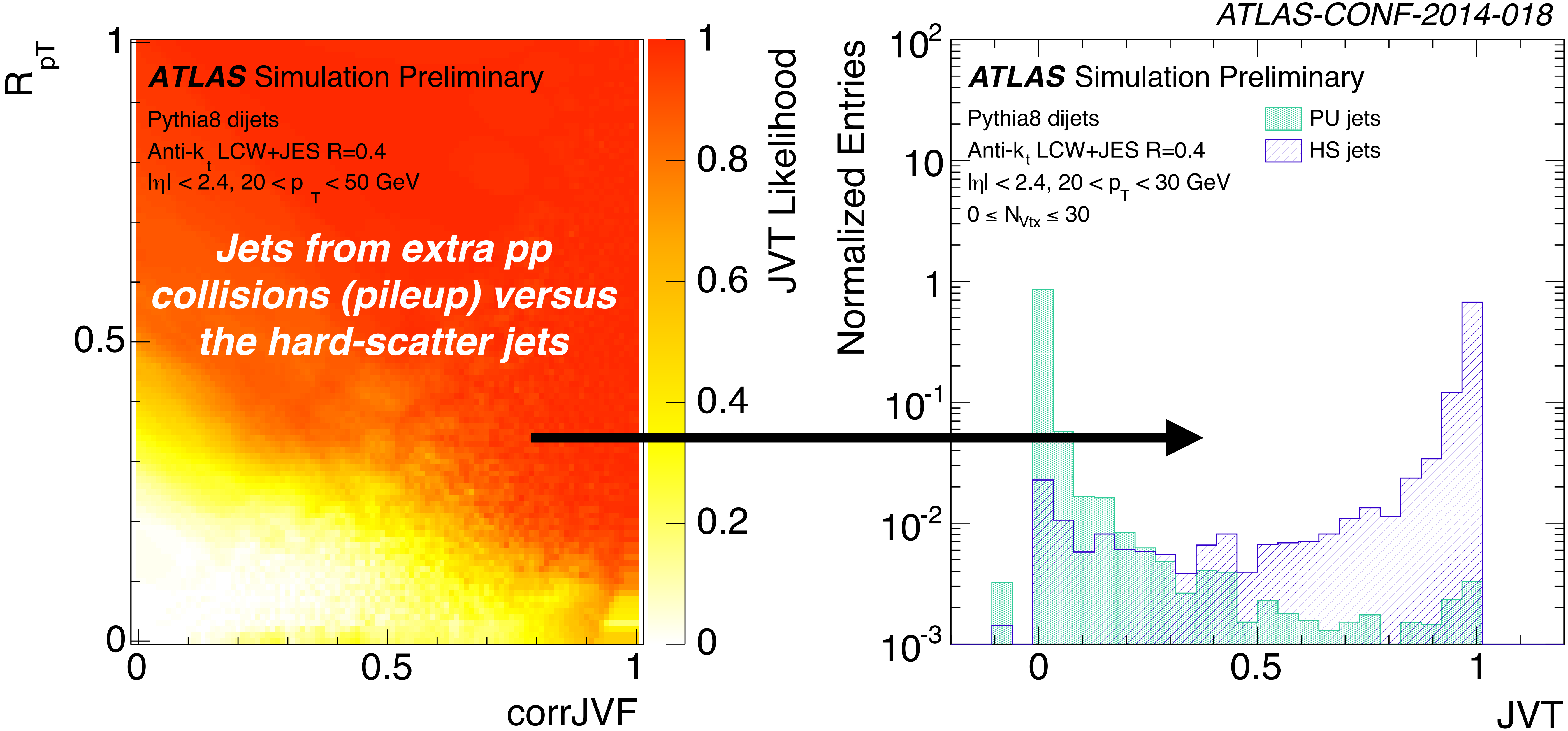


$$p(M, Q, B|V) = \sum_{\mathcal{F}} \Pr(\mathcal{F}|V)p(M|\mathcal{F}, V)p(Q|\mathcal{F}, V)\Pr(B|\mathcal{F}, V),$$

Nearest Neighbors

In 2D, a nice extension of histogramming is to estimate the likelihood ratio based on the number of S and B points nearby.

ATLAS-CONF-2014-018



Boosted Decision Trees (BDTs)

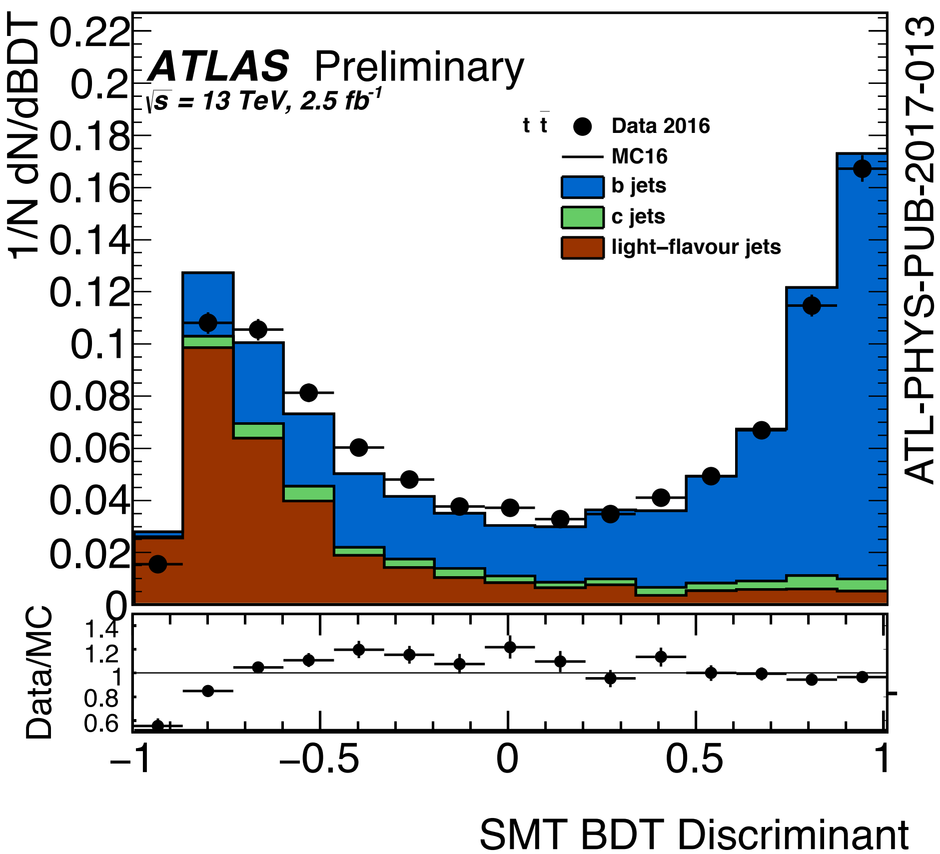
A decision tree is a partition of the feature space. One tree is a set of binary “cuts”.

Boosting makes an ensemble classifier. For example, a community favorite AdaBoost, applies weights to the misclassified events.

XGBoost is becoming more the de facto standard. Actually, this method was popularized because of the Higgs Kaggle Challenge!

N.B. BDTs are not differentiable

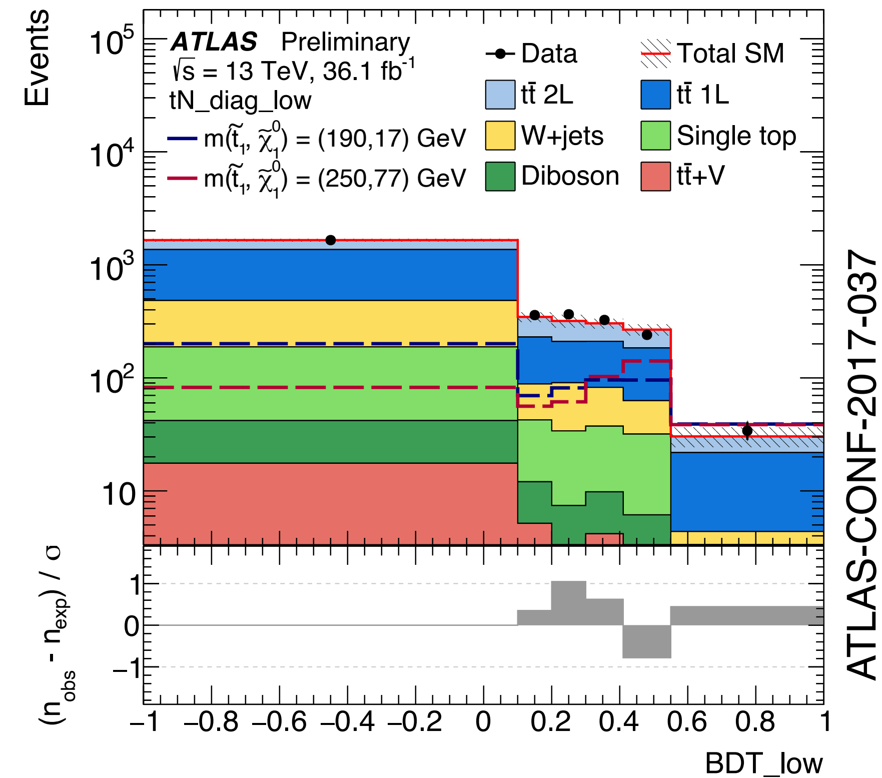
Boosted Decision Trees (BDTs)

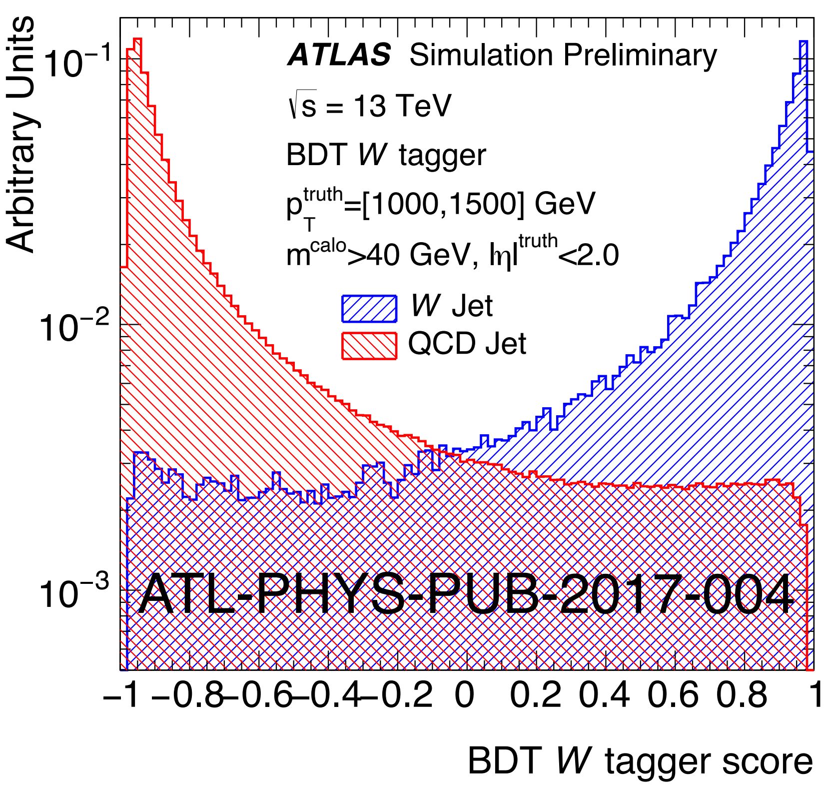


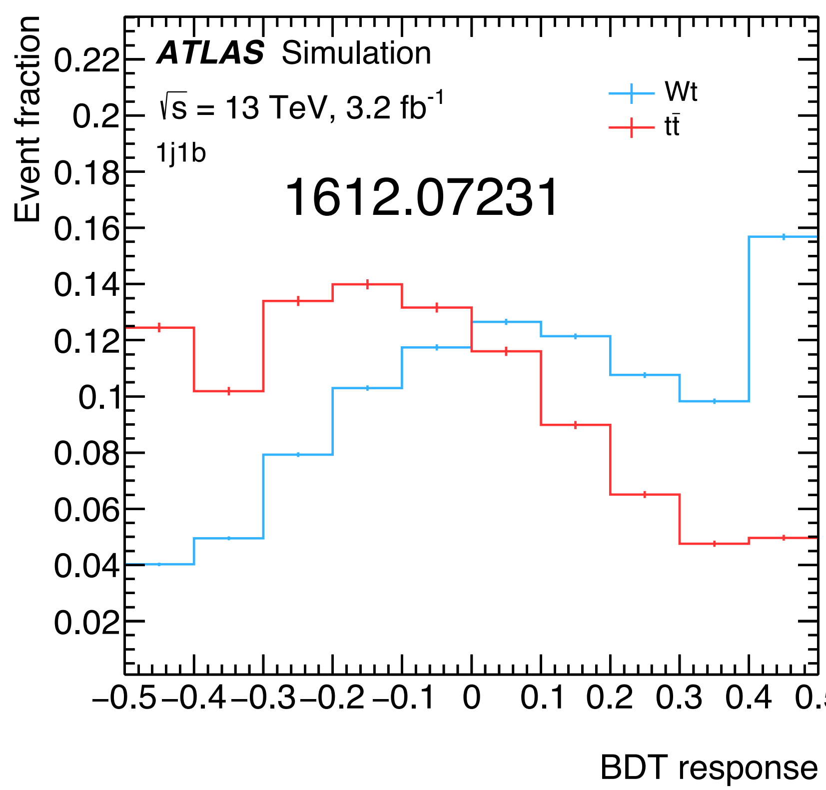
We love BDTs.

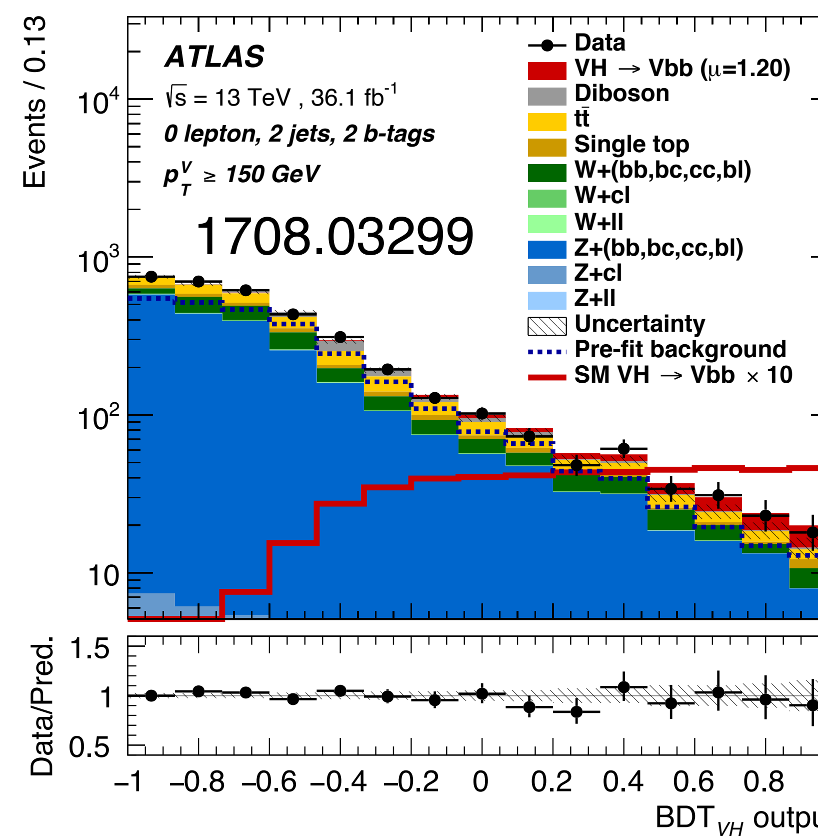
If 3 < dim(feature vector) < O(10)

this is probably right for you!



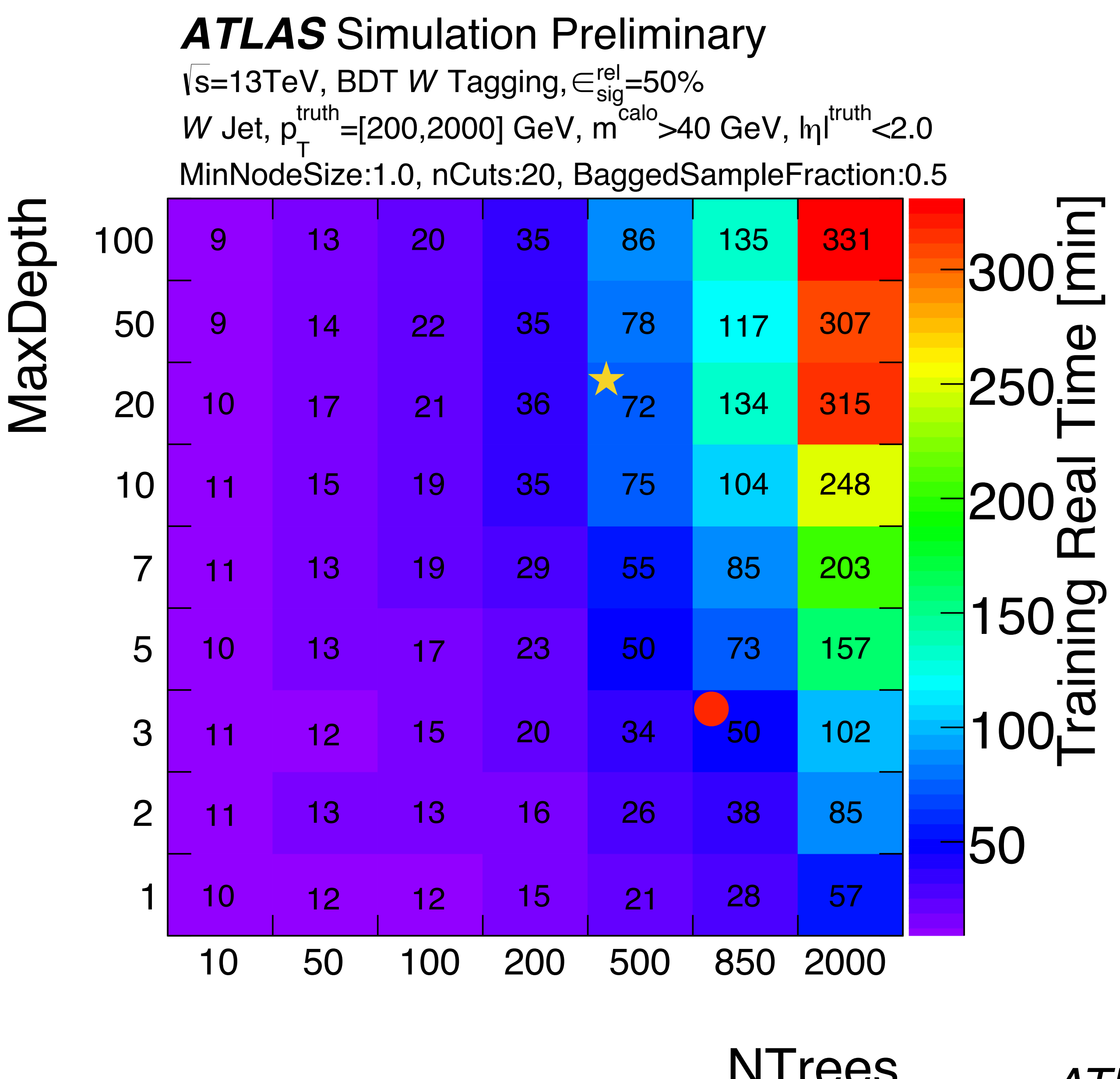




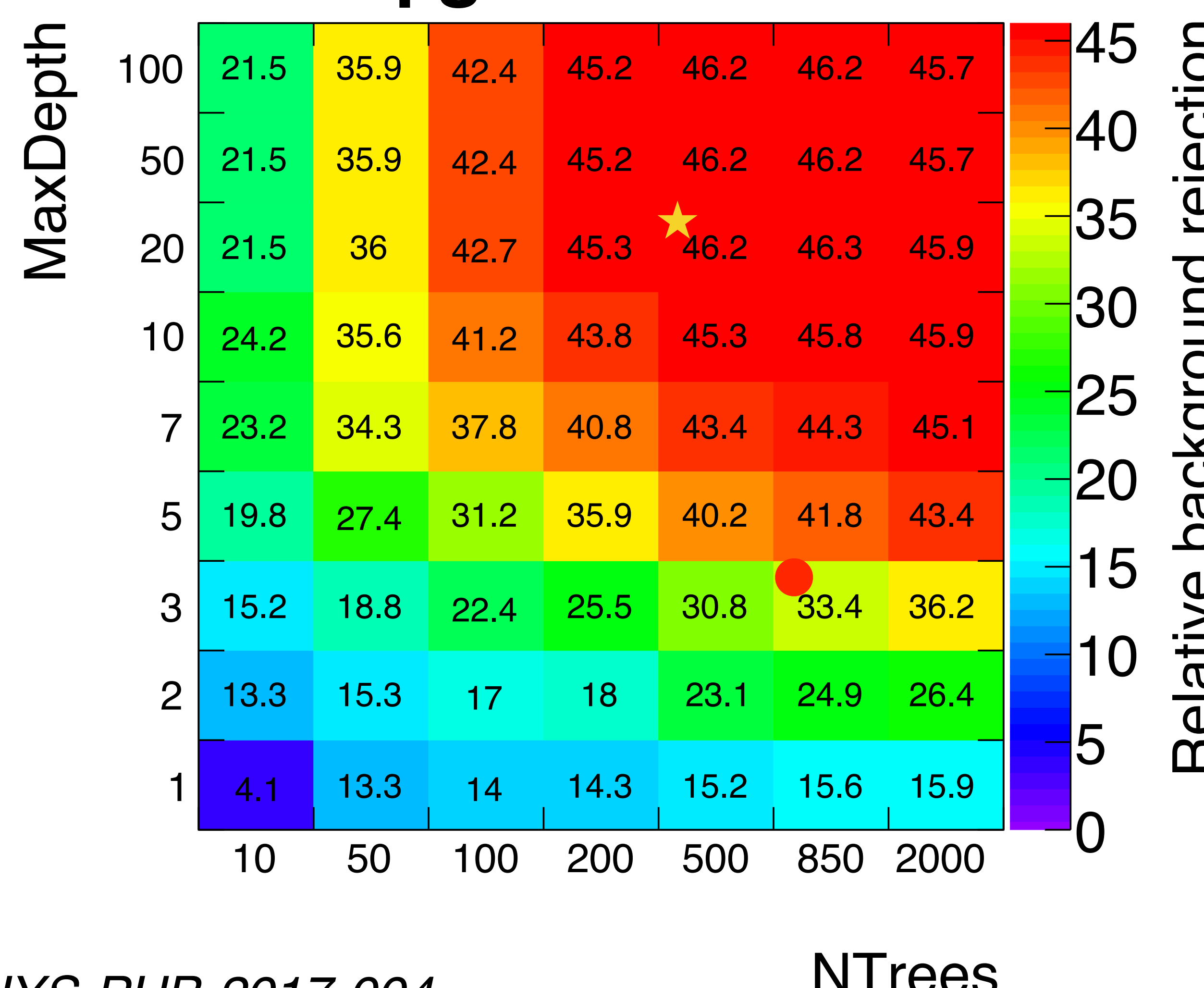


Boosted Decision Trees (BDTs)

We love BDTs because they are fast to train, are close to “cuts”, and do not have very many parameters. They are also rather robust to *overtraining*.



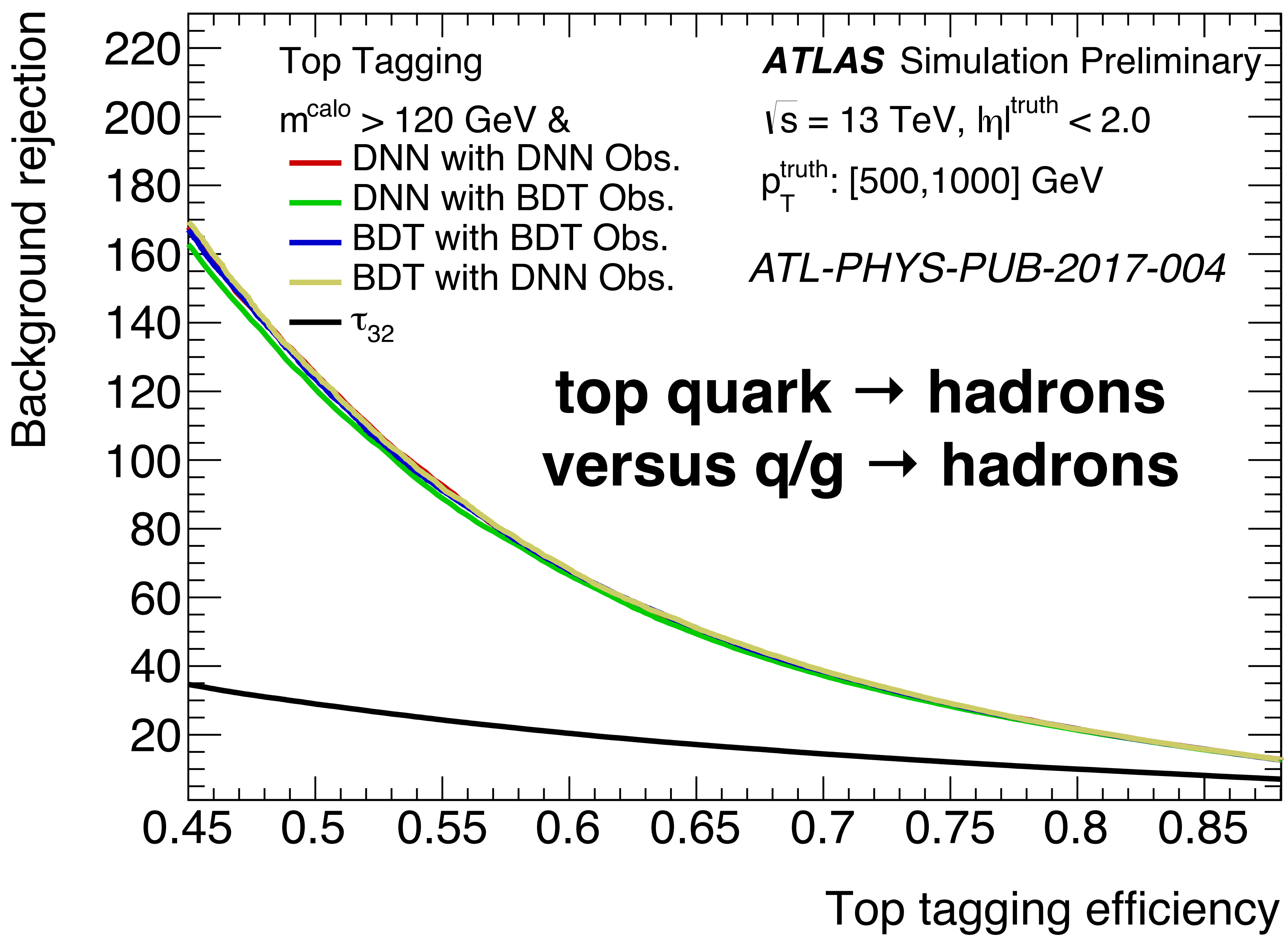
W → hadrons versus q/g → hadrons



ATL-PHYS-PUB-2017-004

Boosted Decision Trees (BDTs)

There is really not a good reason to use a (D)NN with << O(100) dimensions.



However, they are becoming increasingly easy to train …

Neural Networks

Neural Networks were popular at LEP and then mostly fell out of favor until the deep learning revolution.

Finding Gluon Jets with a Neural Trigger

Leif Lönnblad[1], Carsten Peterson[2] and Thorsteinn Rögnvaldsson[3]

Department of Theoretical Physics, University of Lund
Sölvegatan 14A, S-22362 Lund, Sweden

Phys. Rev. Lett. 65 (1990) 1321

Neural Networks

Neural Networks were popular at LEP and then mostly fell out of favor until the deep learning revolution.

The NN's of the 90s are rather different than those of today! With advanced in hardware (GPUs), architectures (dropout, ReLU), etc. the DNNs of today are qualitatively different and more powerful.

What is a Neural Network learning?

Consider the popular binary cross entropy:

$$\text{loss}(f(x)) = -\sum_{i \in S} \log f(x_i) - \sum_{i \in B} \log(1 - f(x_i))$$

What is a Neural Network learning?

Consider the popular binary cross entropy:

$$\text{loss}(f(x)) = -\sum_{i\in S}\log f(x_i) - \sum_{i\in B}\log(1 - f(x_i))$$

If f is optimal, what will it learn?

One can show that **asymptotically**,

$$f(x) \approx \Pr(S \,|\, X = x)$$

(this is monotonic with the likelihood ratio)

What is a Neural Network le

Consider the popular bin

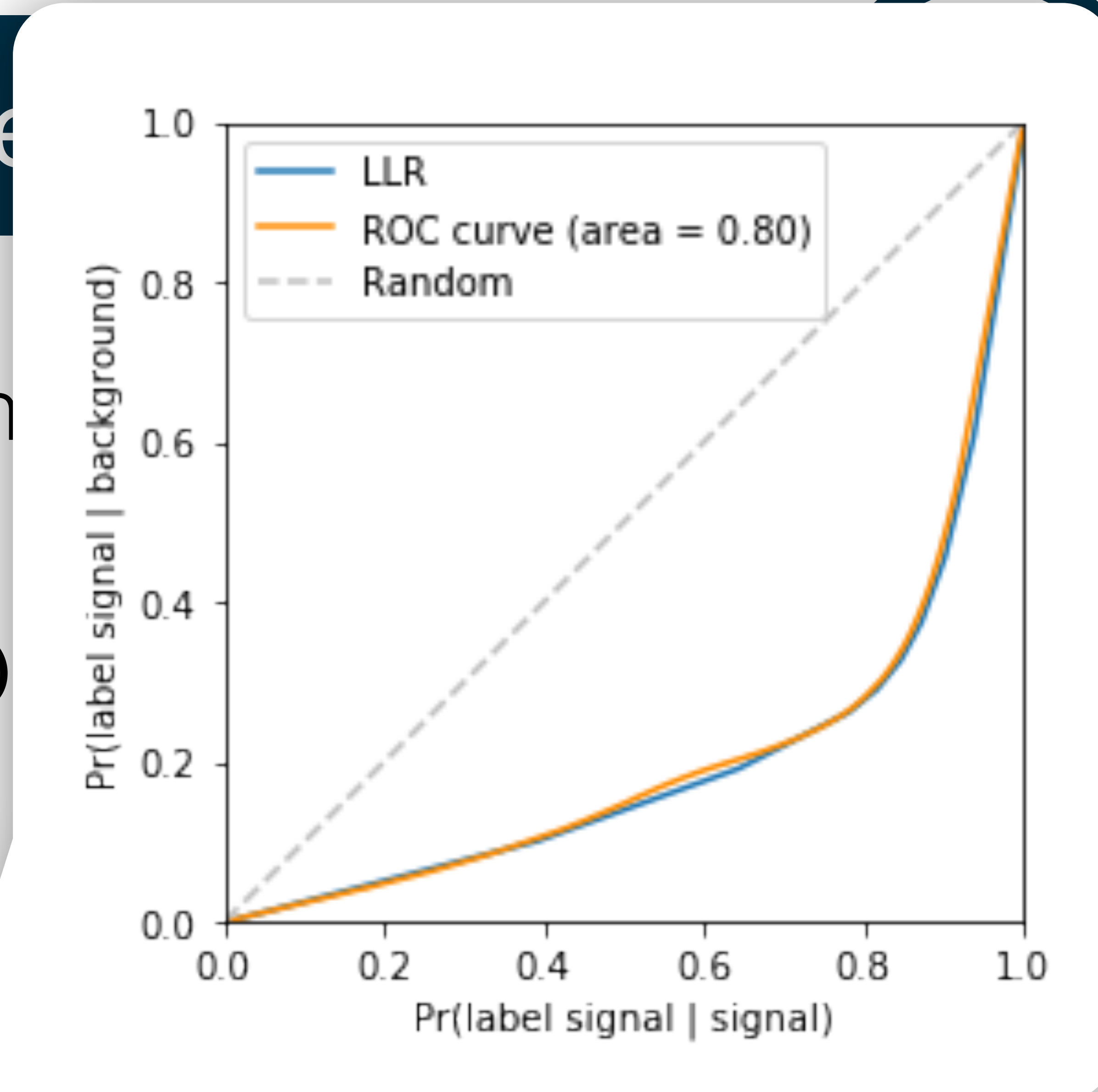
$$\text{loss}(f(x)) = -\sum_{i\in S} \log f(x_i)$$

If f is optimal, wha

One can show tha **ymptotically**,

$$f(x) \approx \Pr(S \mid X = x)$$

(this is monotonic with the likelihood ratio)



See 1709.04464 for image refs.

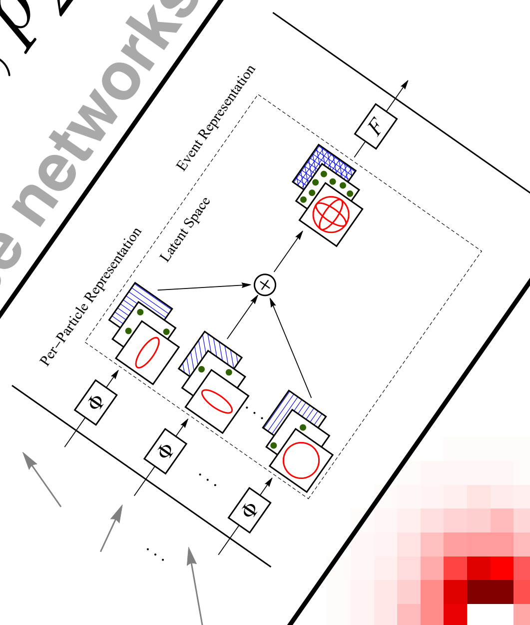
Deep neural networks for HEP classification

Fixed sets

$$J = \{p_1^\mu, p_2^\mu, \ldots, p_n^\mu\}$$

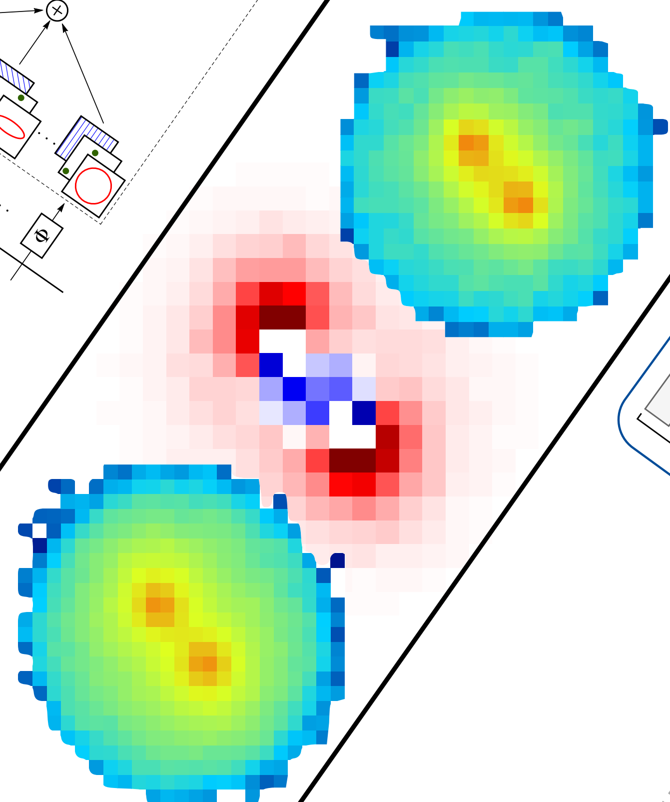
Dense networks

Variable sets



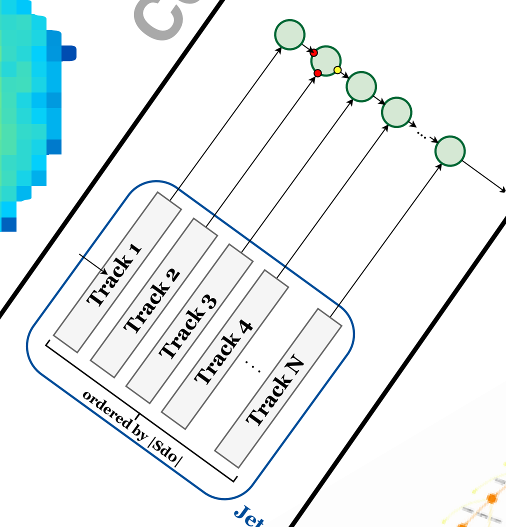
Deep sets

Images



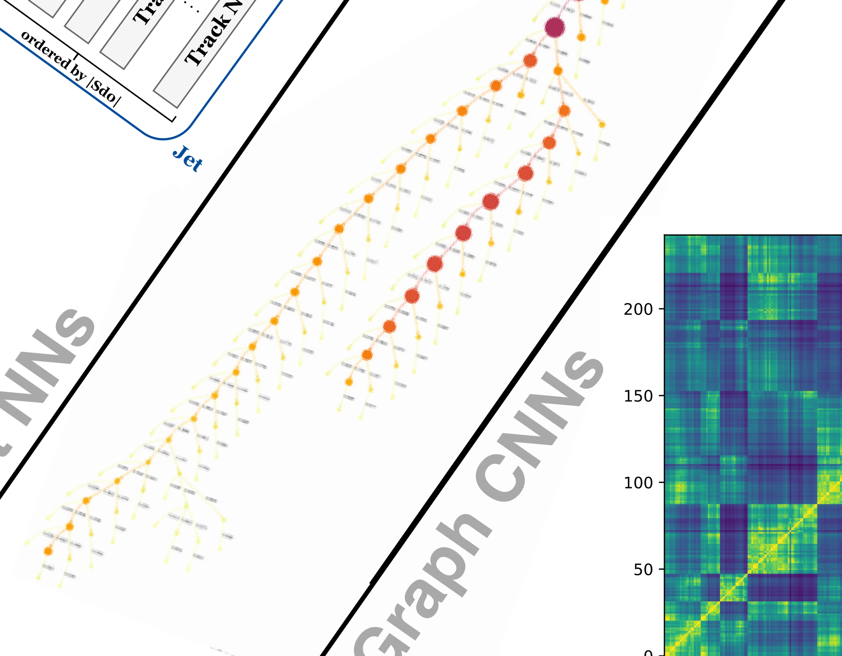
Convolutional NNs (CNNs)

Sequences



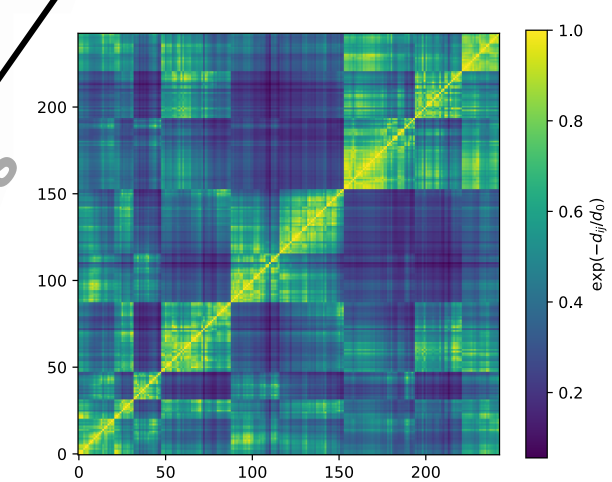
Recurrent NNs

Trees



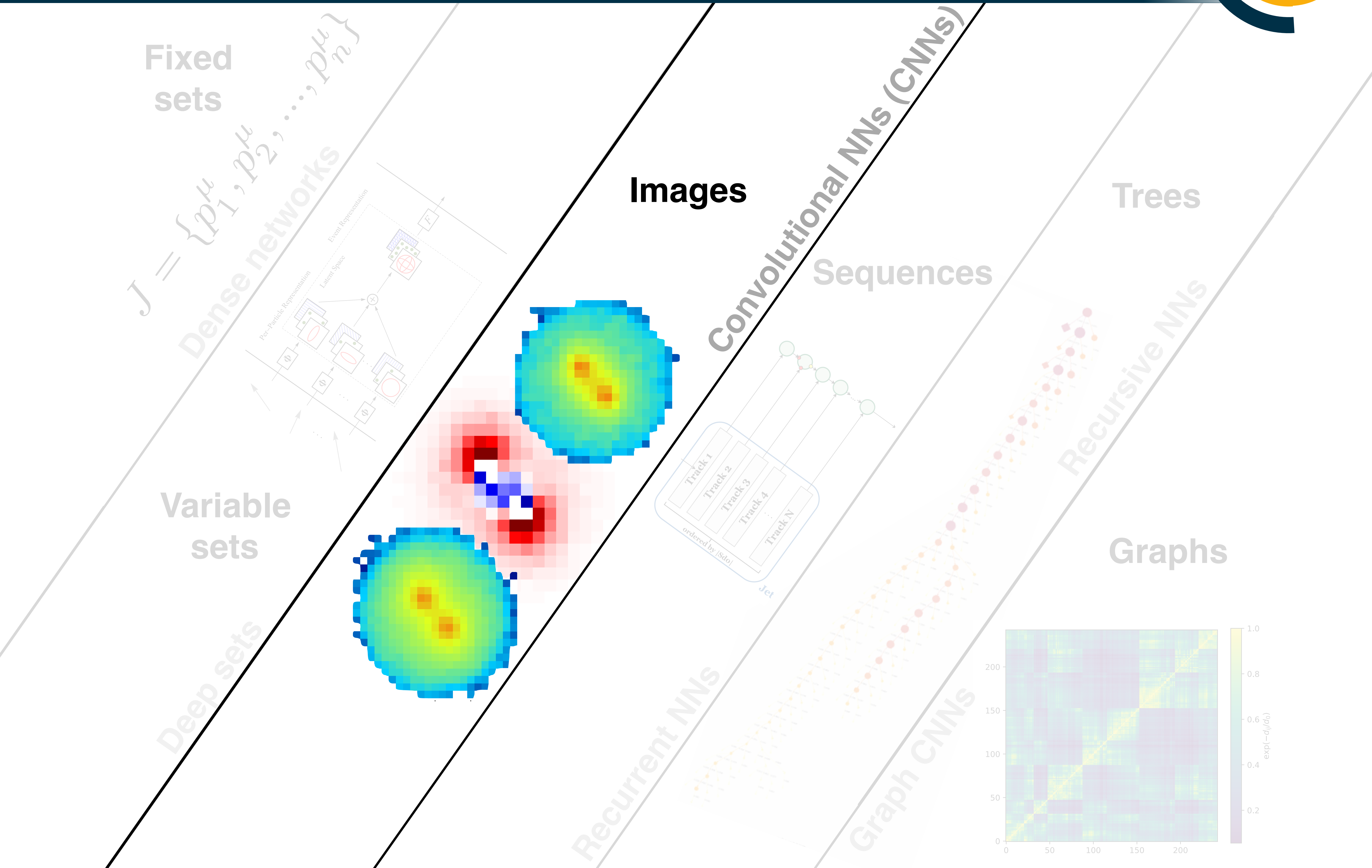
Recursive NNs

Graphs

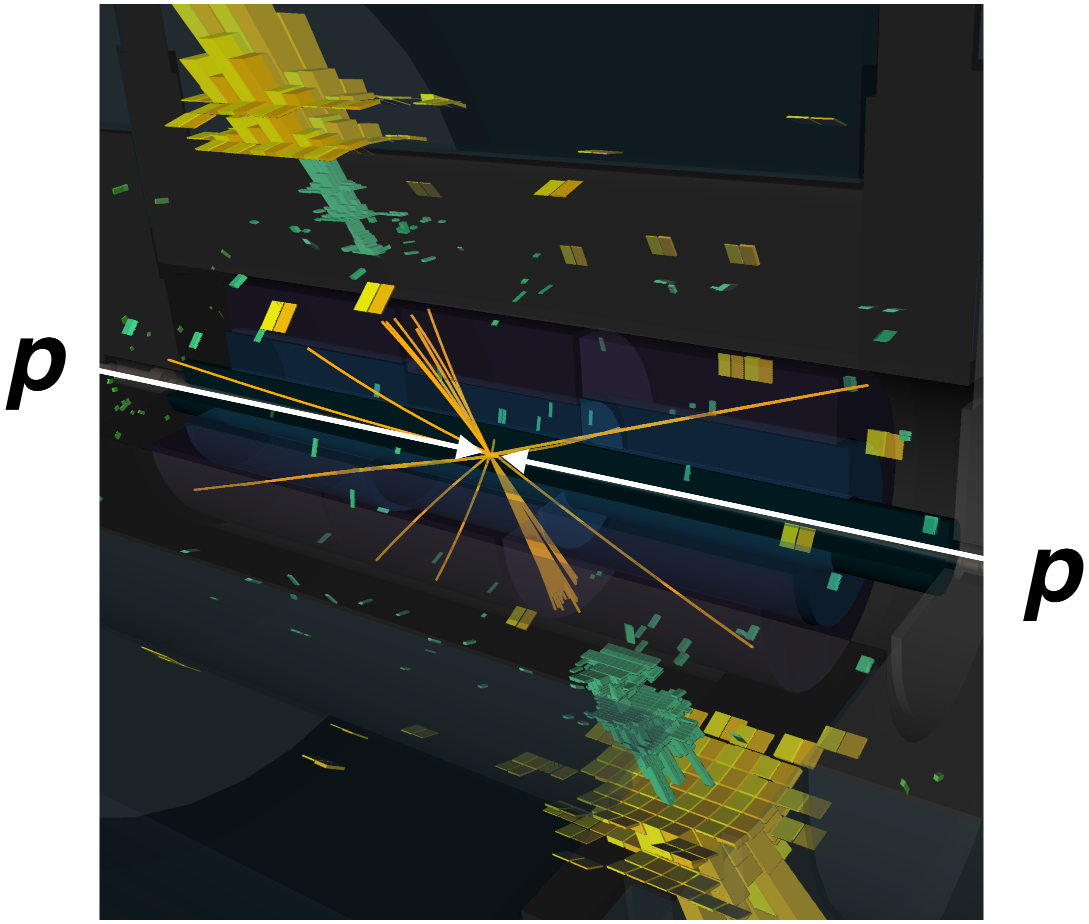


Graph CNNs

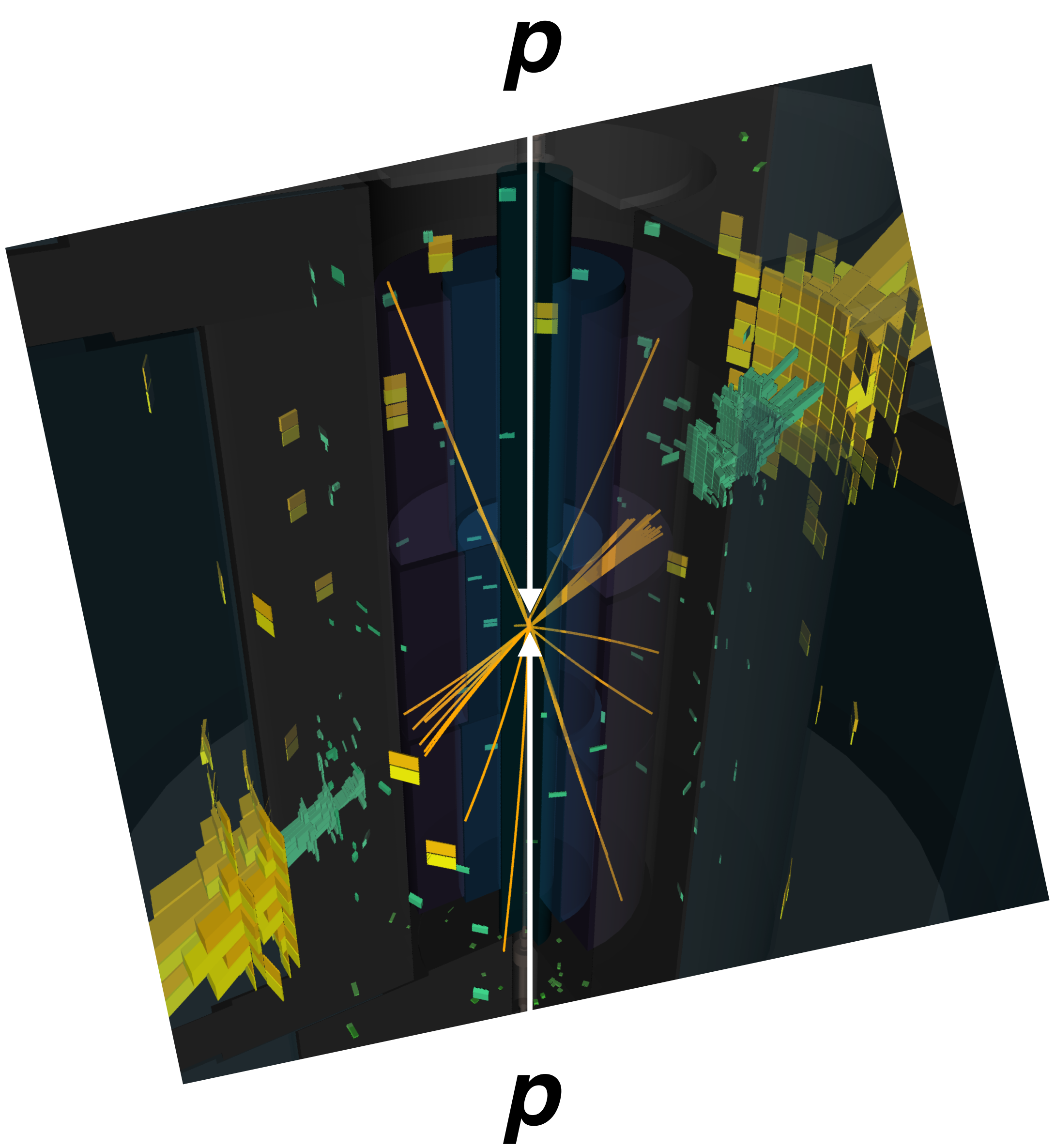
Deep neural networks for HEP classification



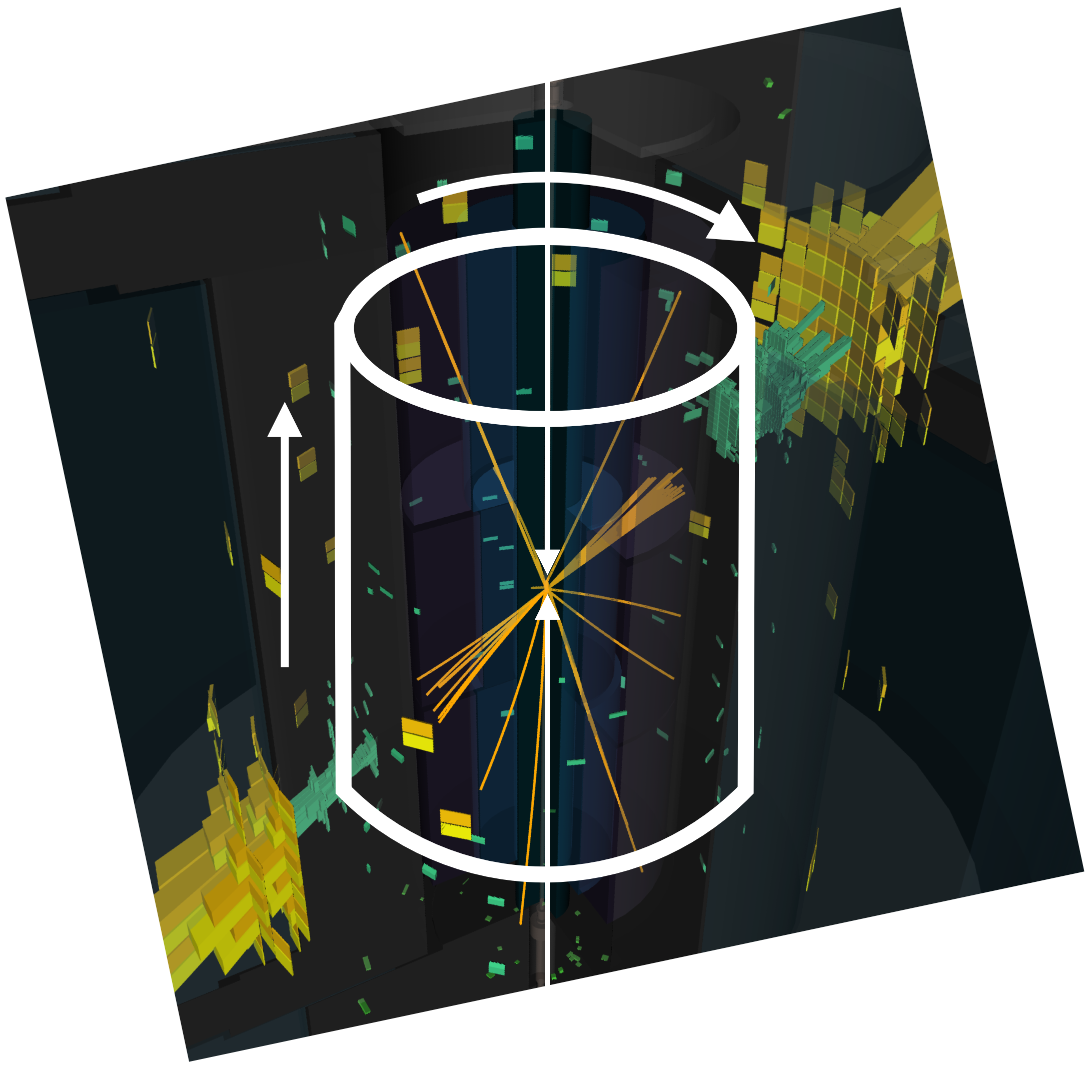
HEP data as an image



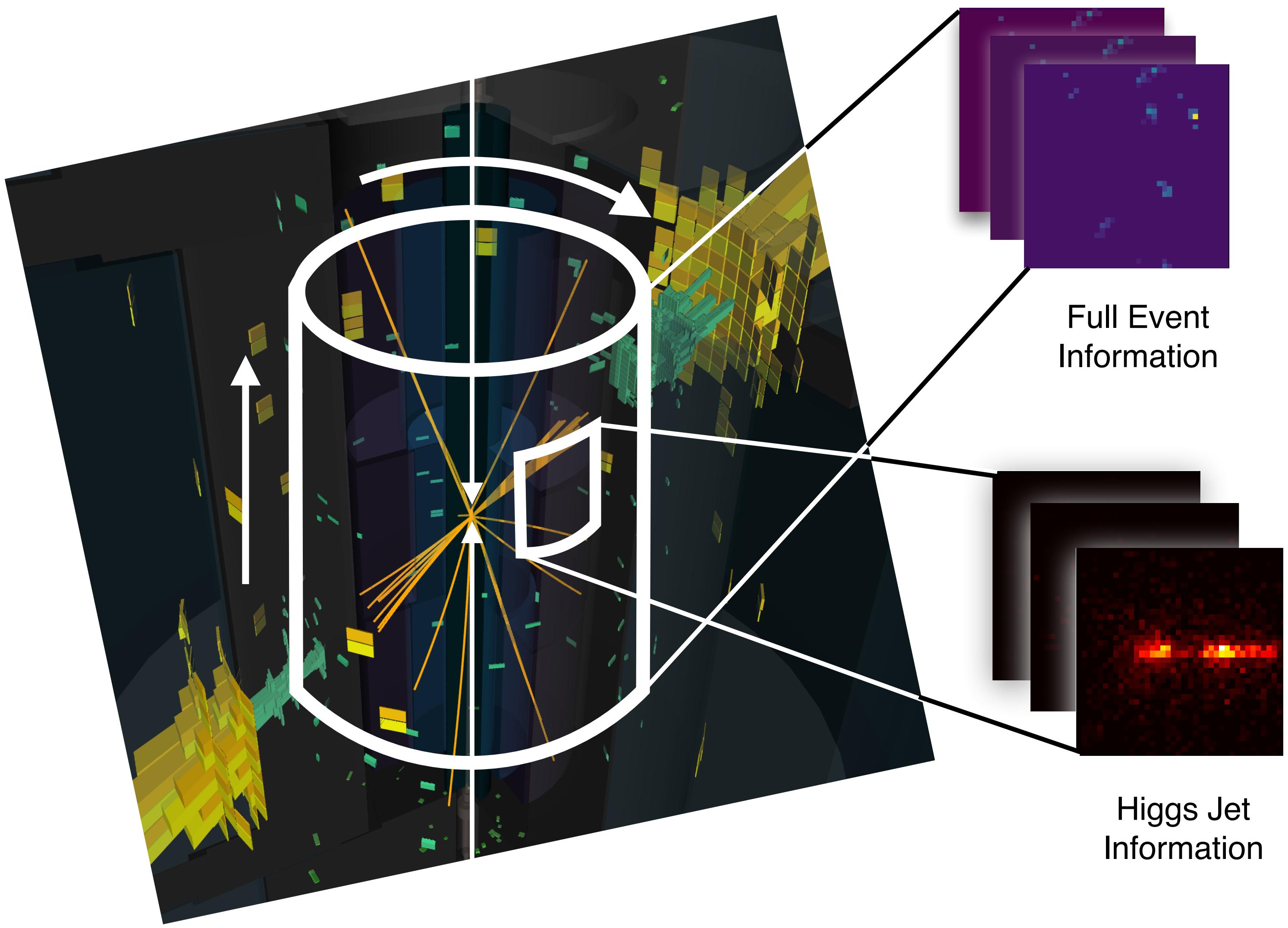
HEP data as an image



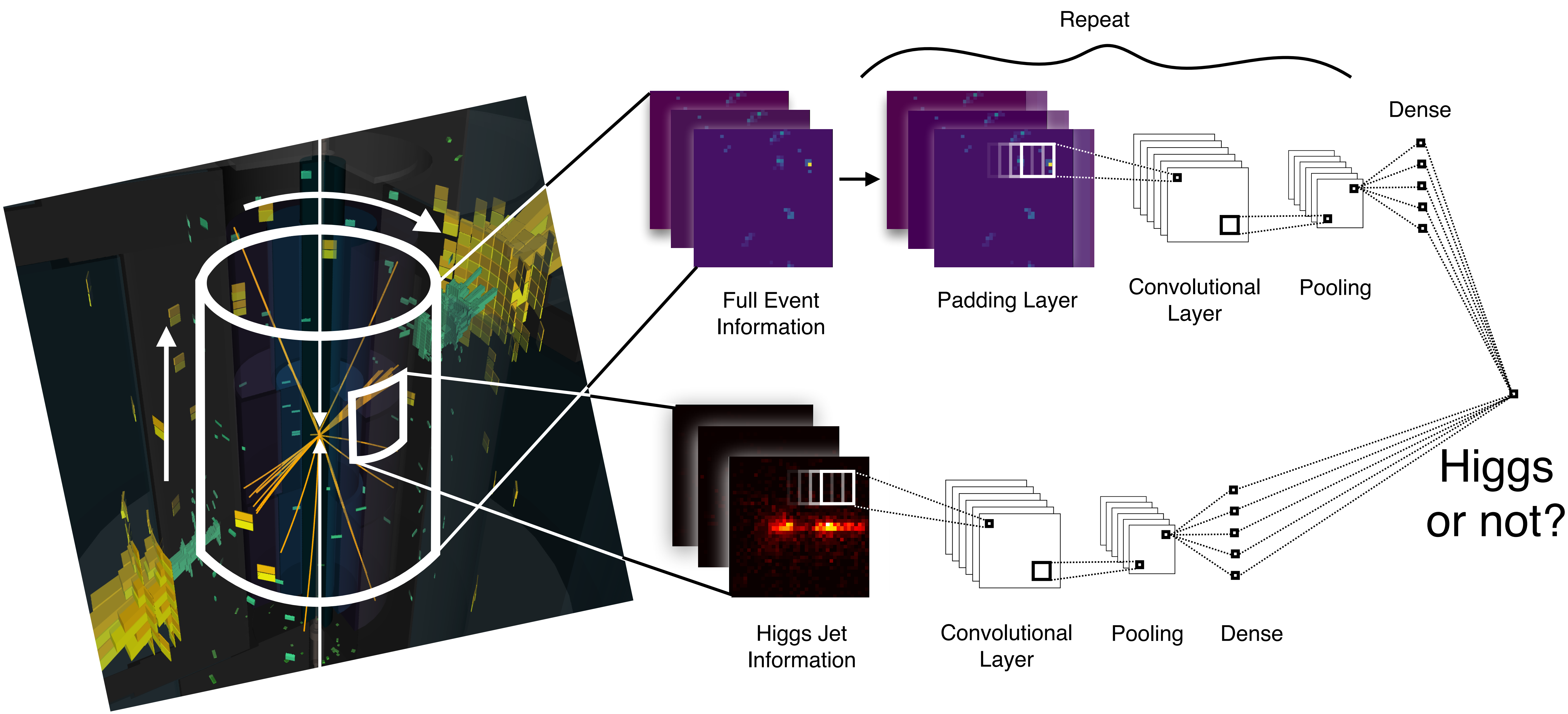
HEP data as an image



HEP data as an image

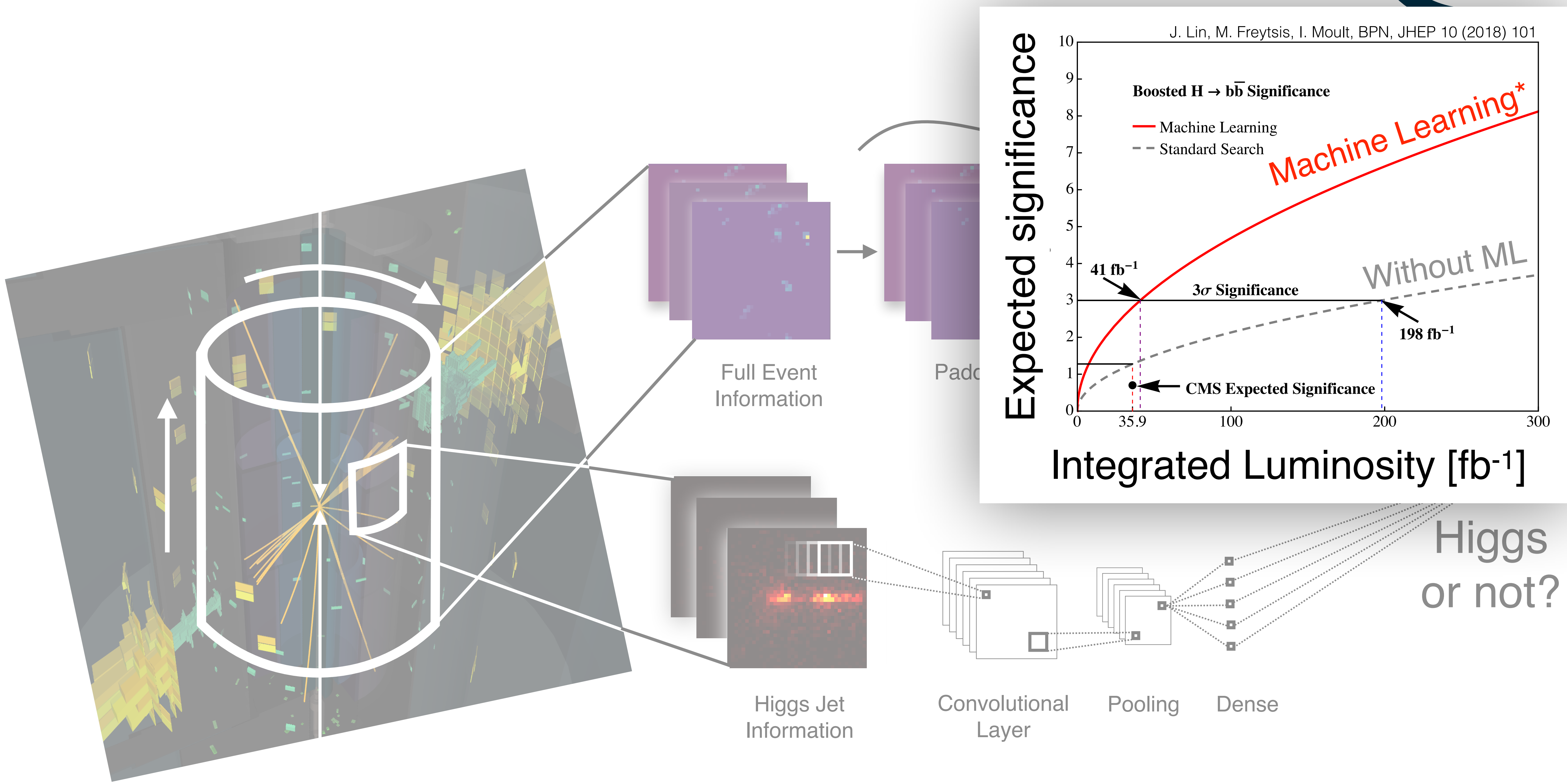


HEP data as an image



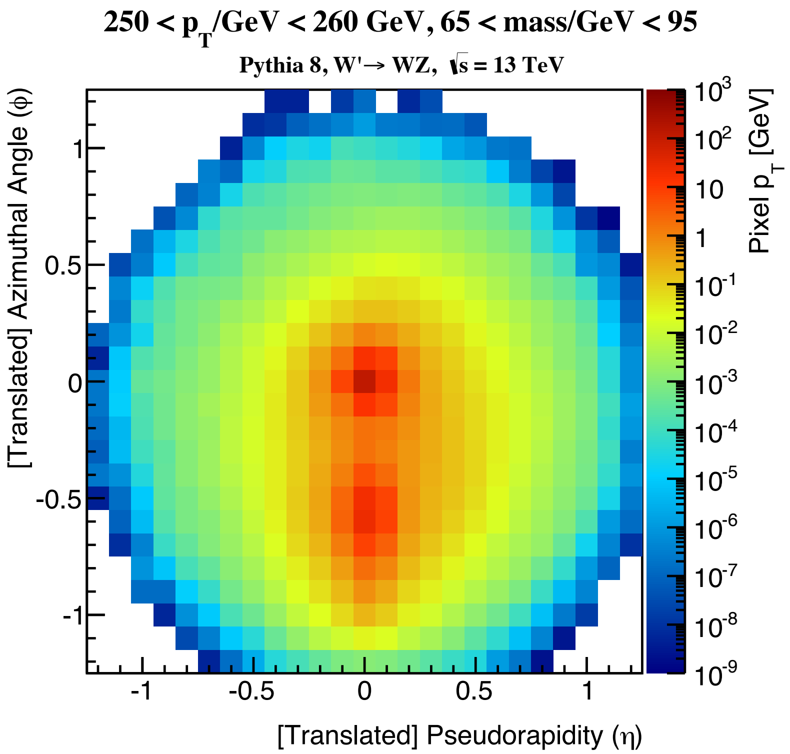
Can combine local and global information from jet images and "event" images.

HEP data as an image



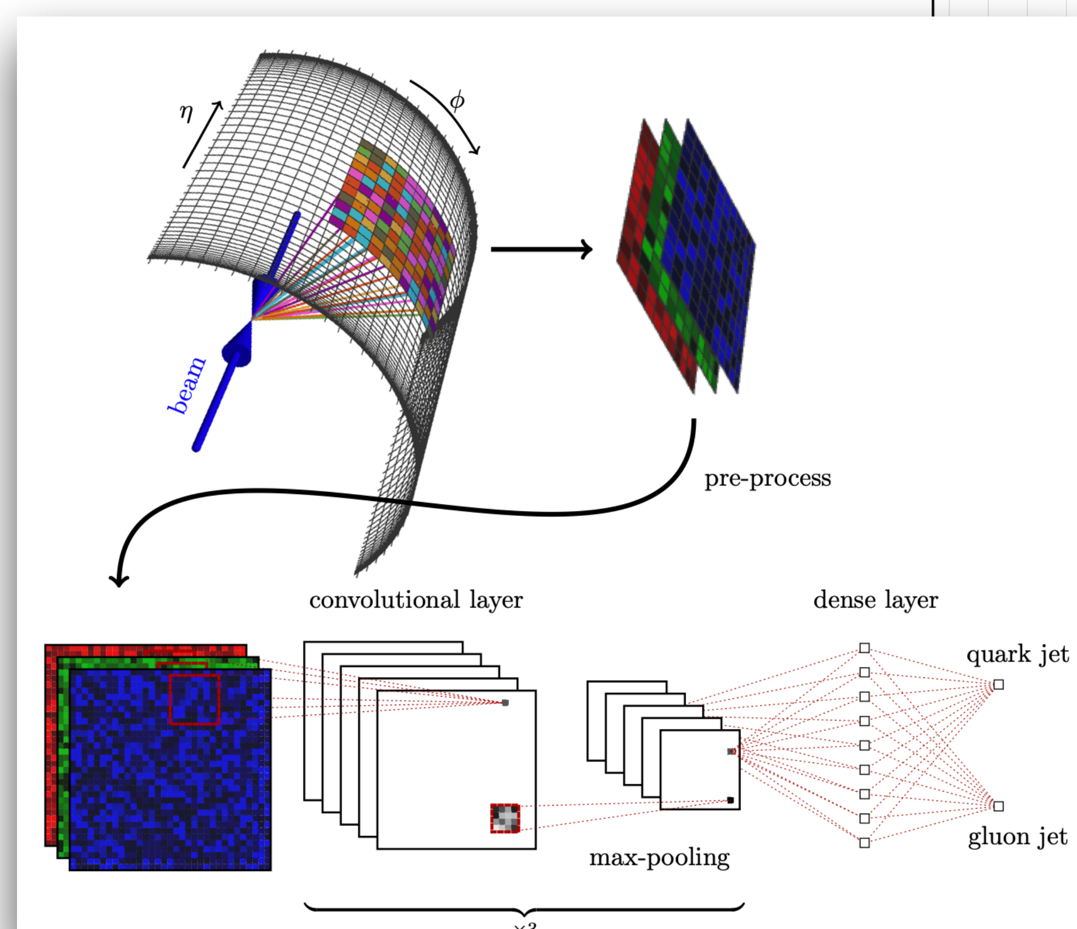
Can combine local and global information from jet images and "event" images.

More HEP images

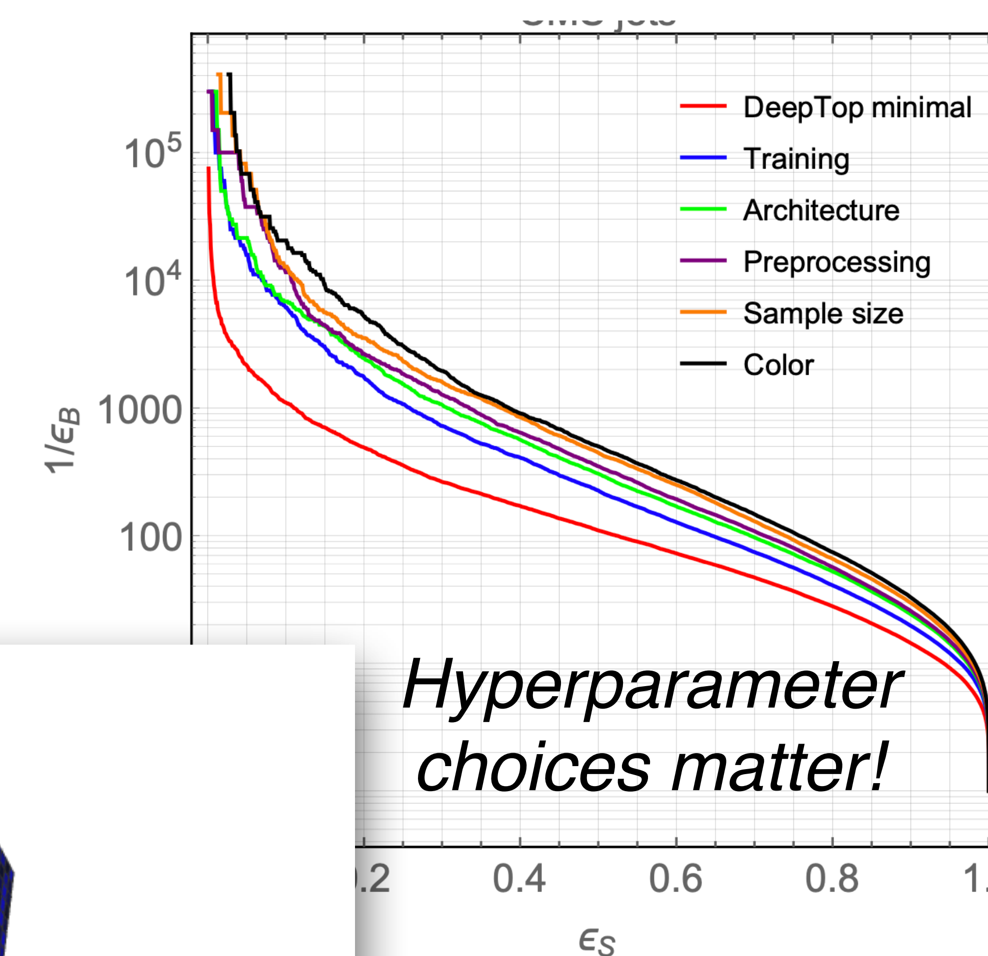


L. De Oliveira et al., JHEP 07 (2016) 069

Many topologies: top quarks, W/Z/H bosons, BSM particles, q/g, etc.



P. Komiske, E. Metodiev, M Schwartz, JHEP 01 (2017) 110

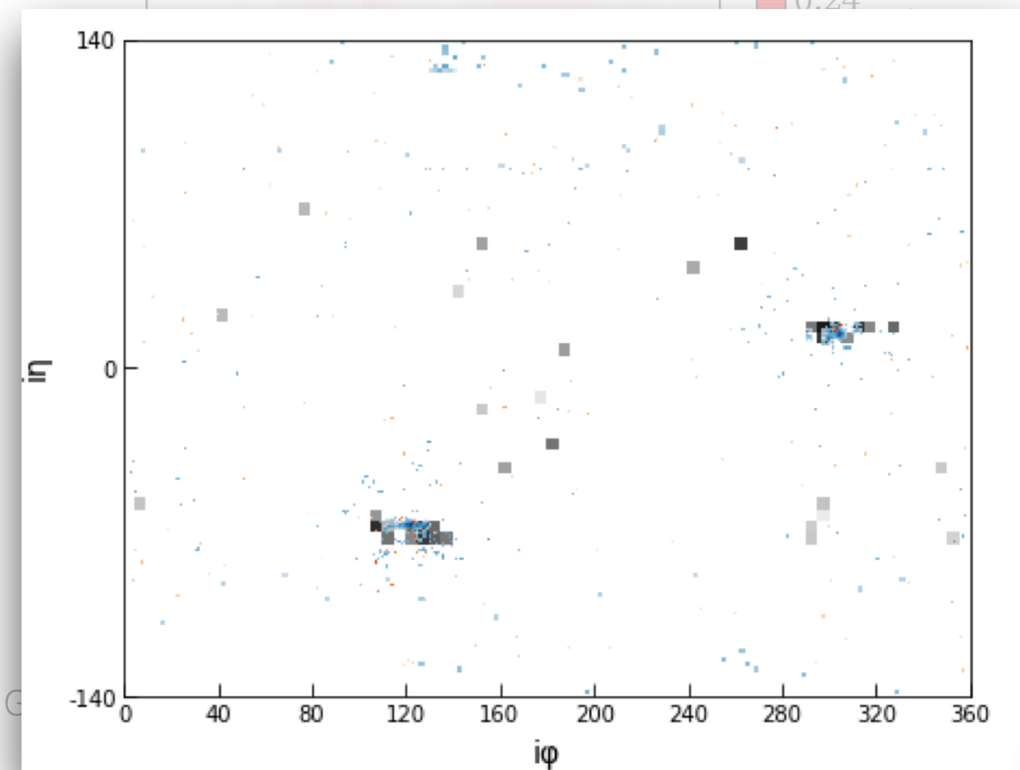


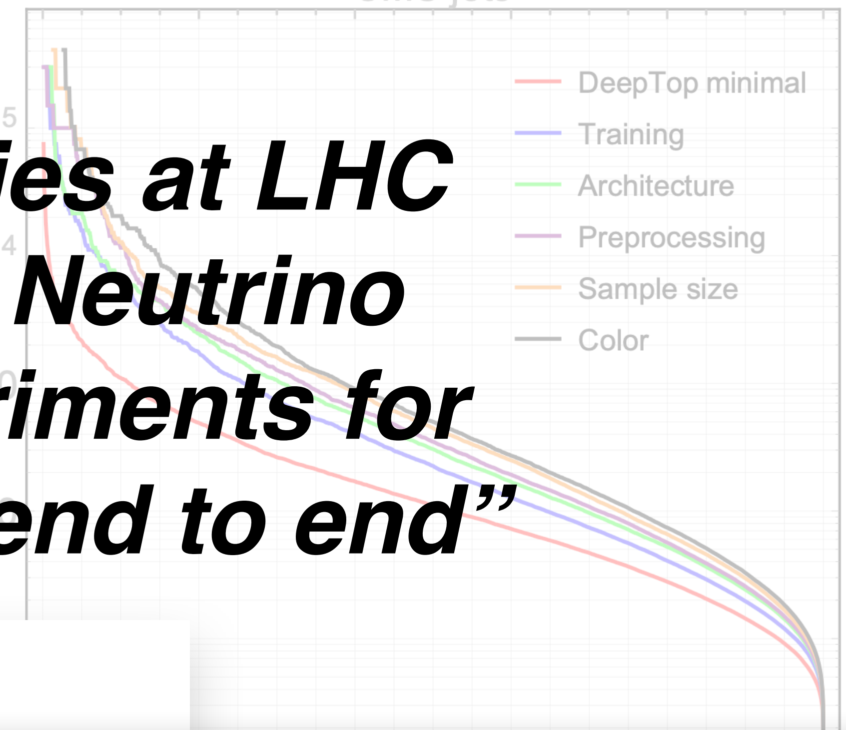
S. Macalusco and D. Shih, JHEP 10 (2018) 121

Multi-channel: use calorimeter & tracking information to make RGB image.

More HEP images

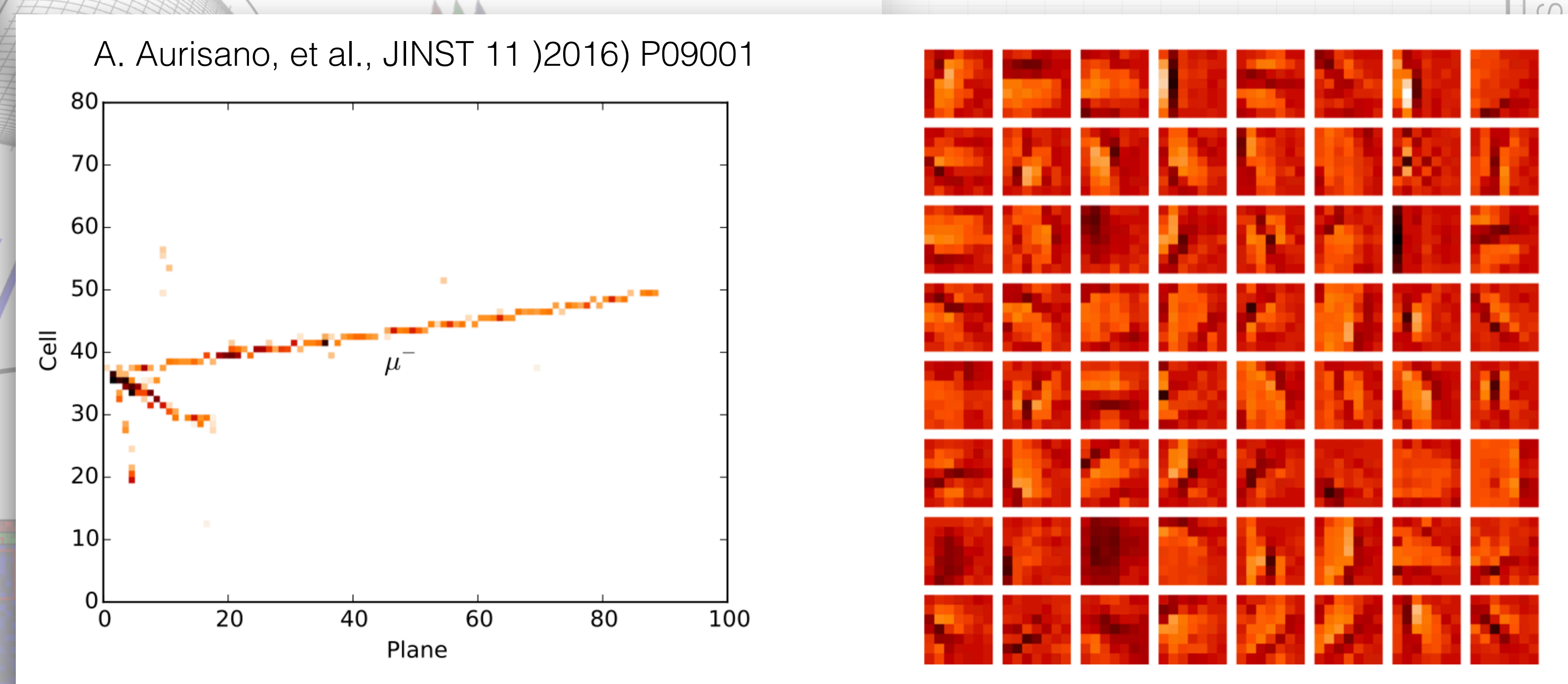
M. Andrews et al., https://arxiv.org/abs/1902.08276





S. Macalusco and D. Shih, JHEP 10 (2018) 121

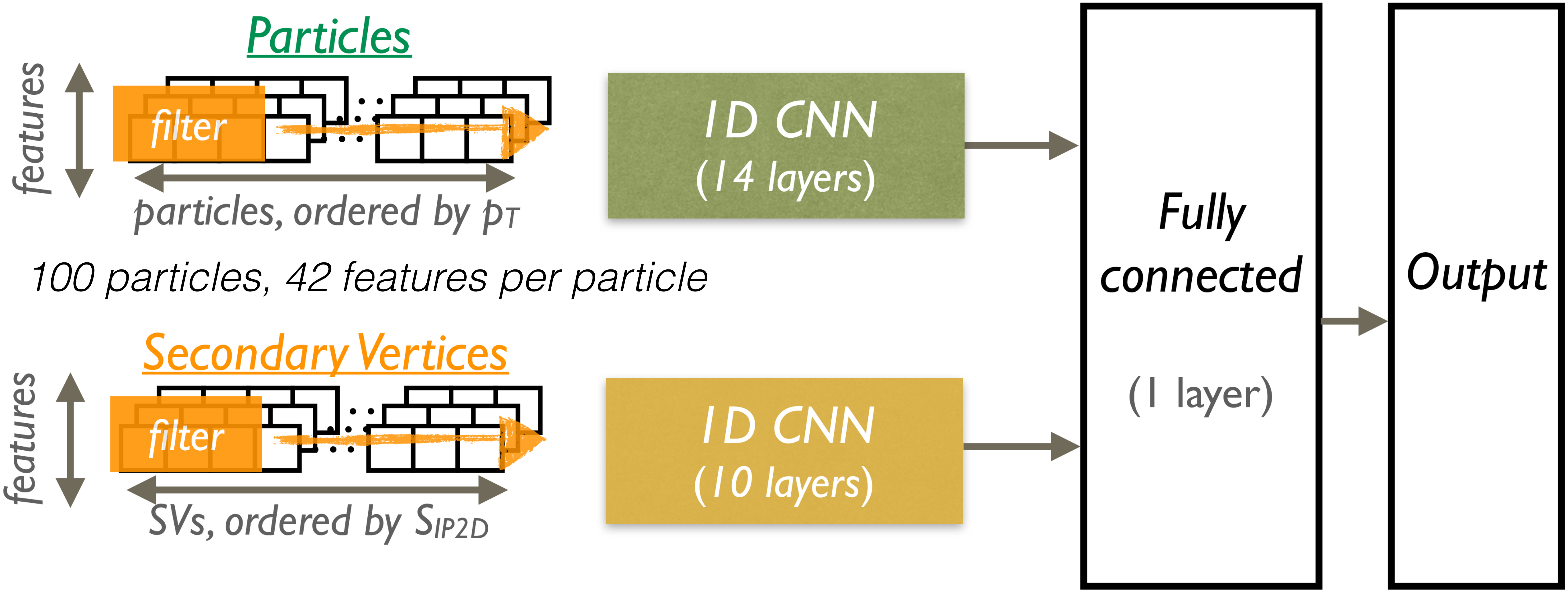
Studies at LHC and Neutrino experiments for fully "end to end"



P. Komiske, E. Metodiev, M Schwartz, JHEP 01 (2017) 110

A last word about CNNs for now

One can use CNNs as automated "feature extractors" even if the inputs are not images.



This is the structure of the CMS Collaboration Deep AK8 jet classifier.

CMS Collaboration, JINST 15 (2020) P06005

See 1709.04464 for image refs.

Deep neural networks for HEP classification

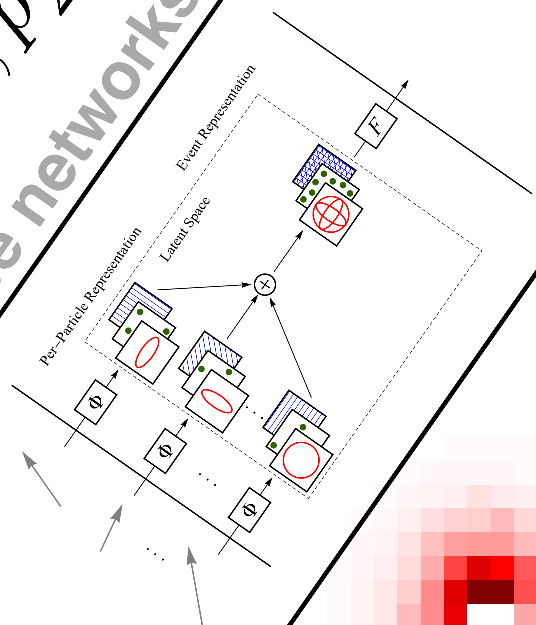
Fixed sets

$$J = \{p_1^\mu, p_2^\mu, \ldots, p_n^\mu\}$$

Dense networks

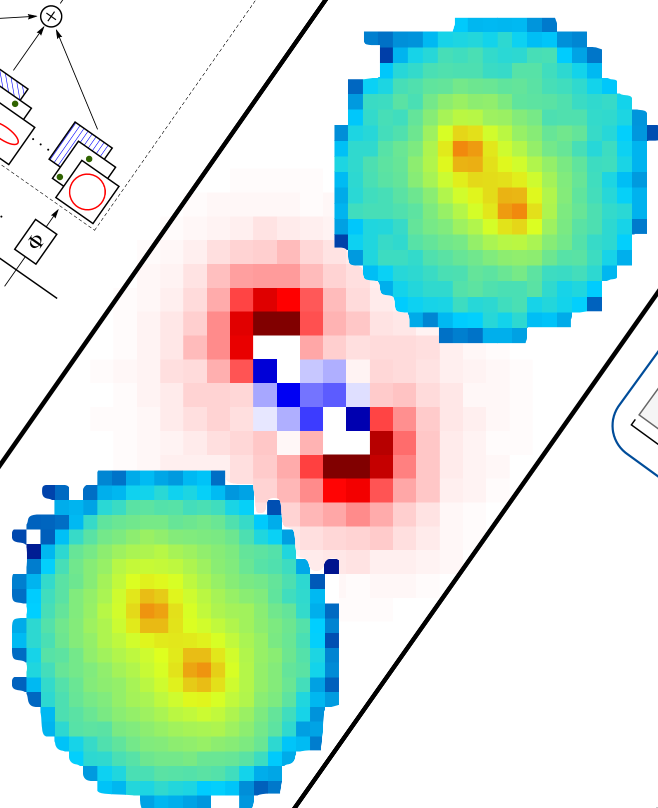
Variable sets

Deep sets



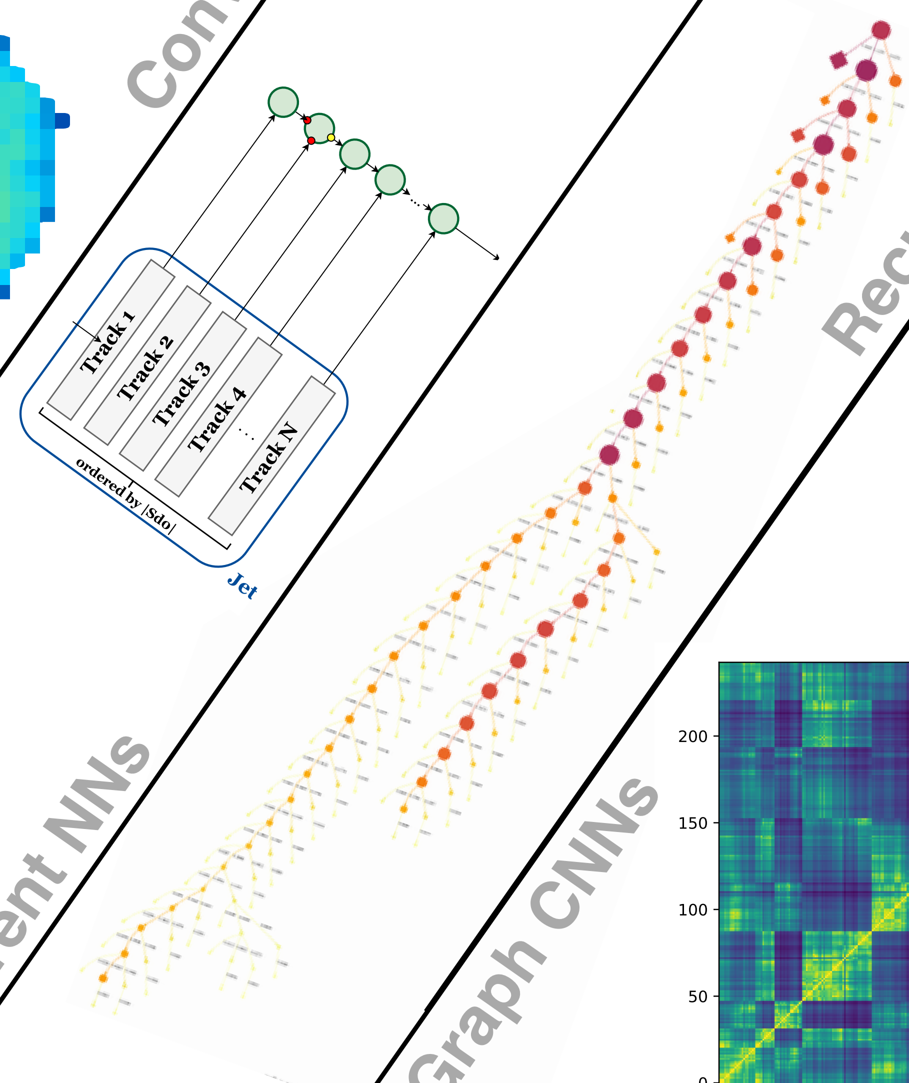
Images

Convolutional NNs (CNNs)



Sequences

Recurrent NNs

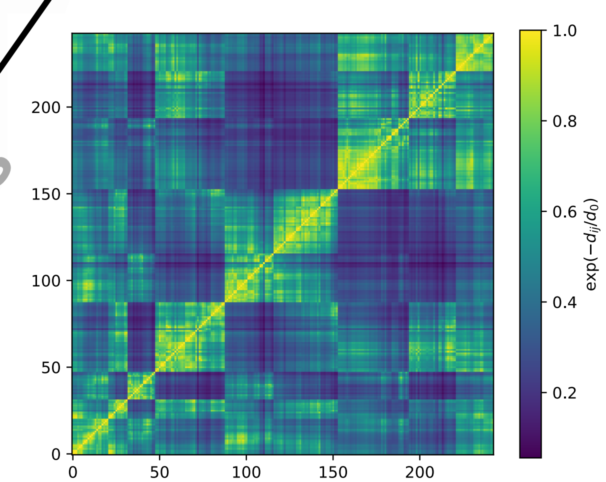


Trees

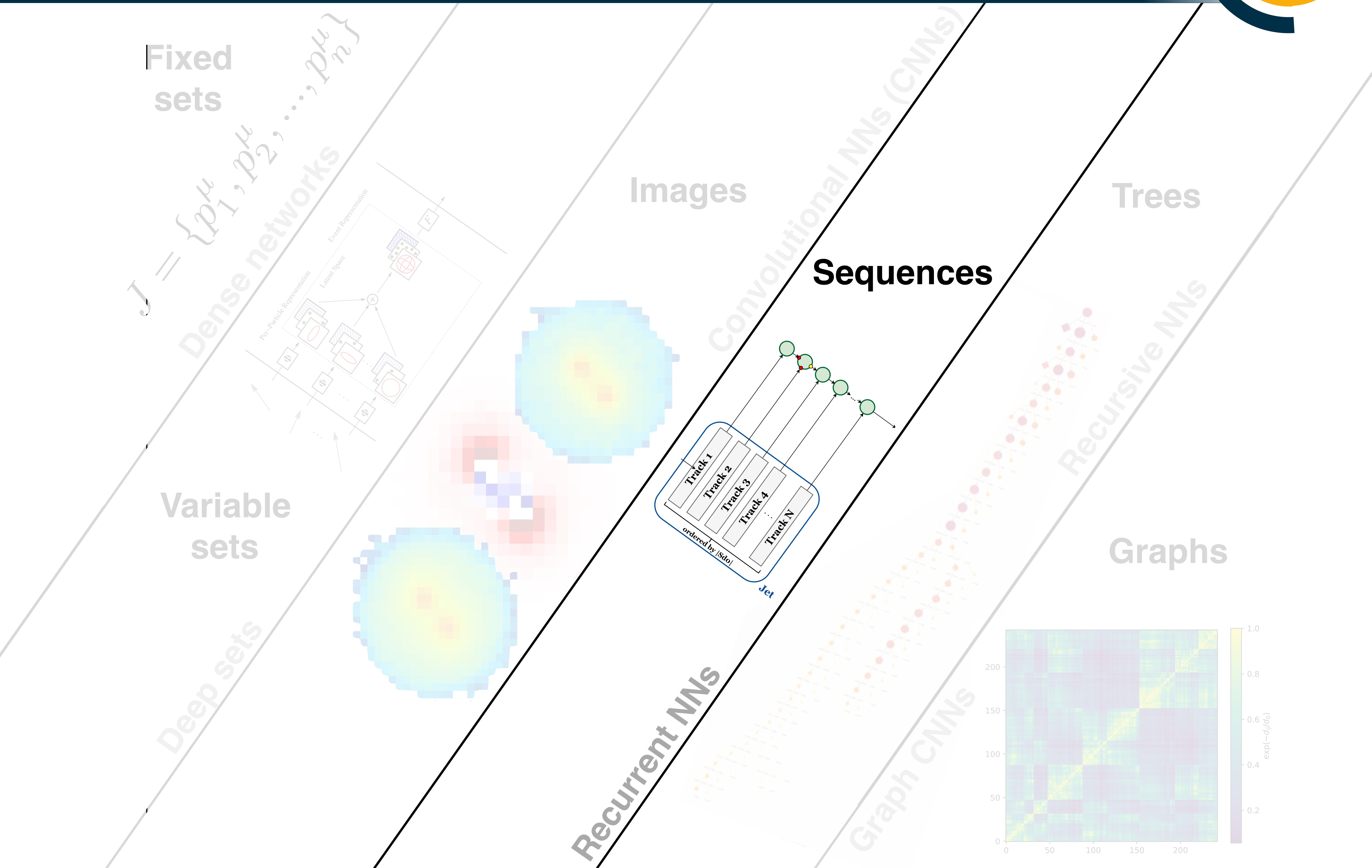
Recursive NNs

Graphs

Graph CNNs



Deep neural networks for HEP classification



Sequence learning

One key challenge with images is that they have a fixed size.

In many contexts, this is ideal, because the data also have a fixed size. However, this is not always the case.

For example, events / jets have a variable number of particles.

One can represent these particles as a sequence in order to apply variable-length approaches that can access the full feature granularity.

Sequence learning with RNNs

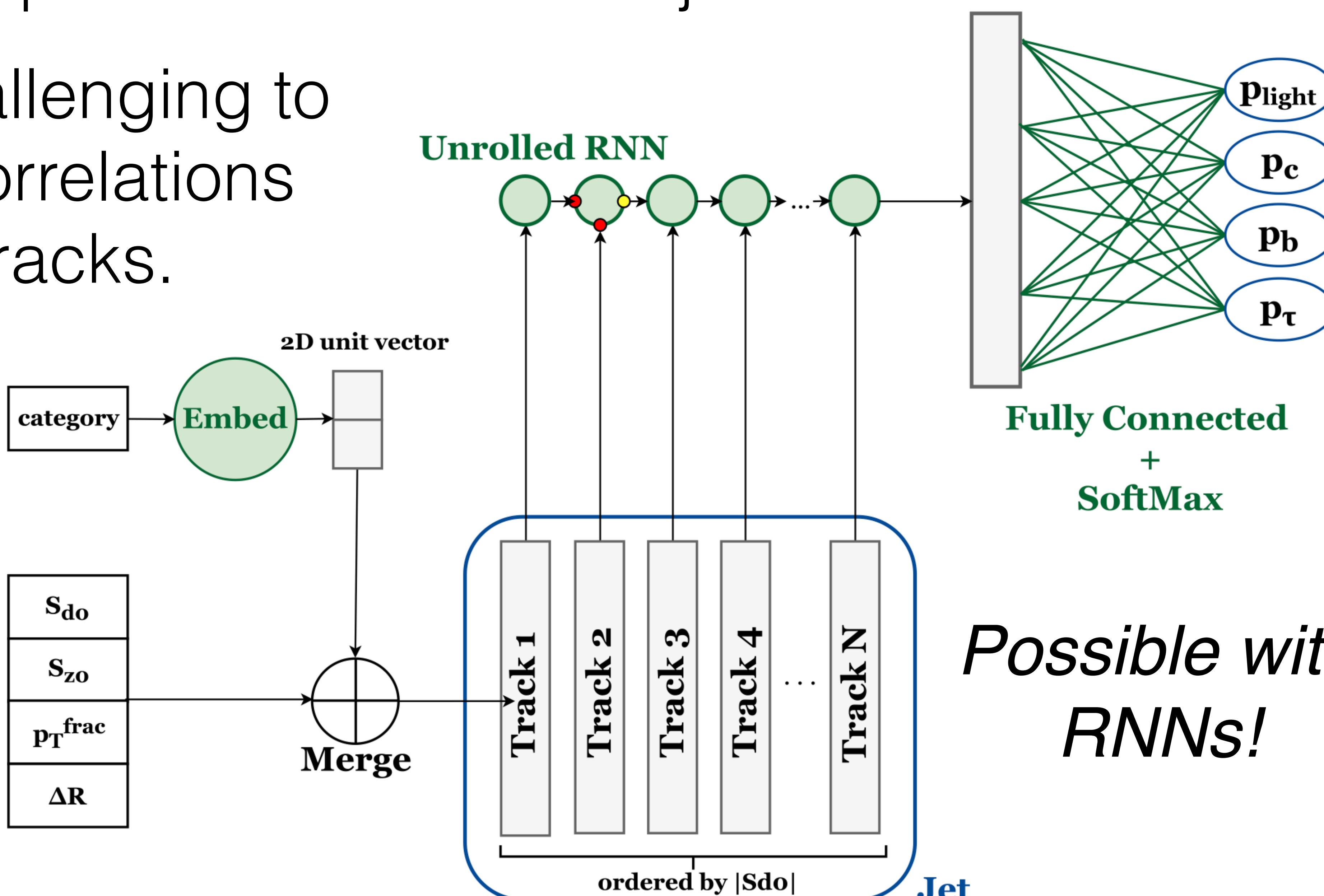
Flavor tagging (classify jets from b-quark or not) has a long history of ML. Use features of the charged-particle tracks inside jets.

In the past, challenging to incorporate correlations between tracks.

Sequence learning with RNNs

Flavor tagging (classify jets from b-quark or not) has a long history of ML. Use features of the charged-particle tracks inside jets.

In the past, challenging to incorporate correlations between tracks.

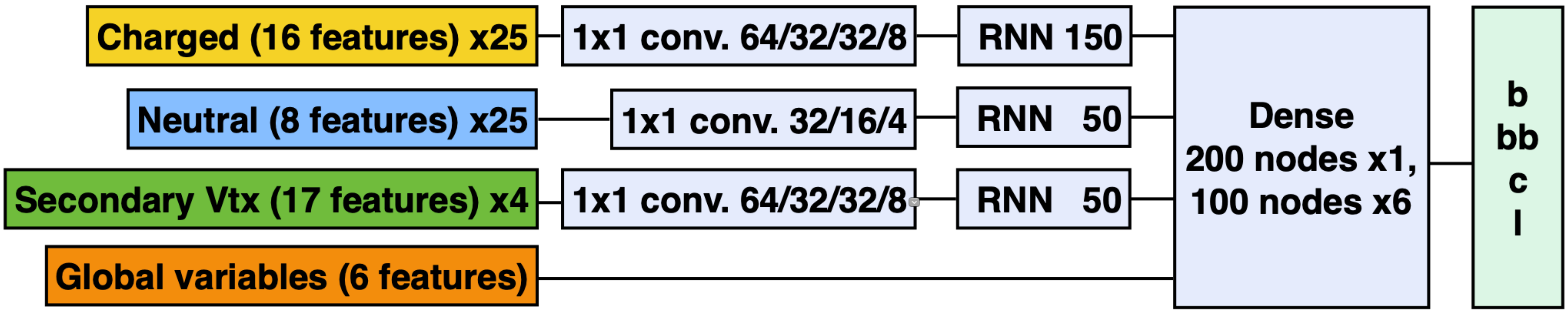


Possible with RNNs!

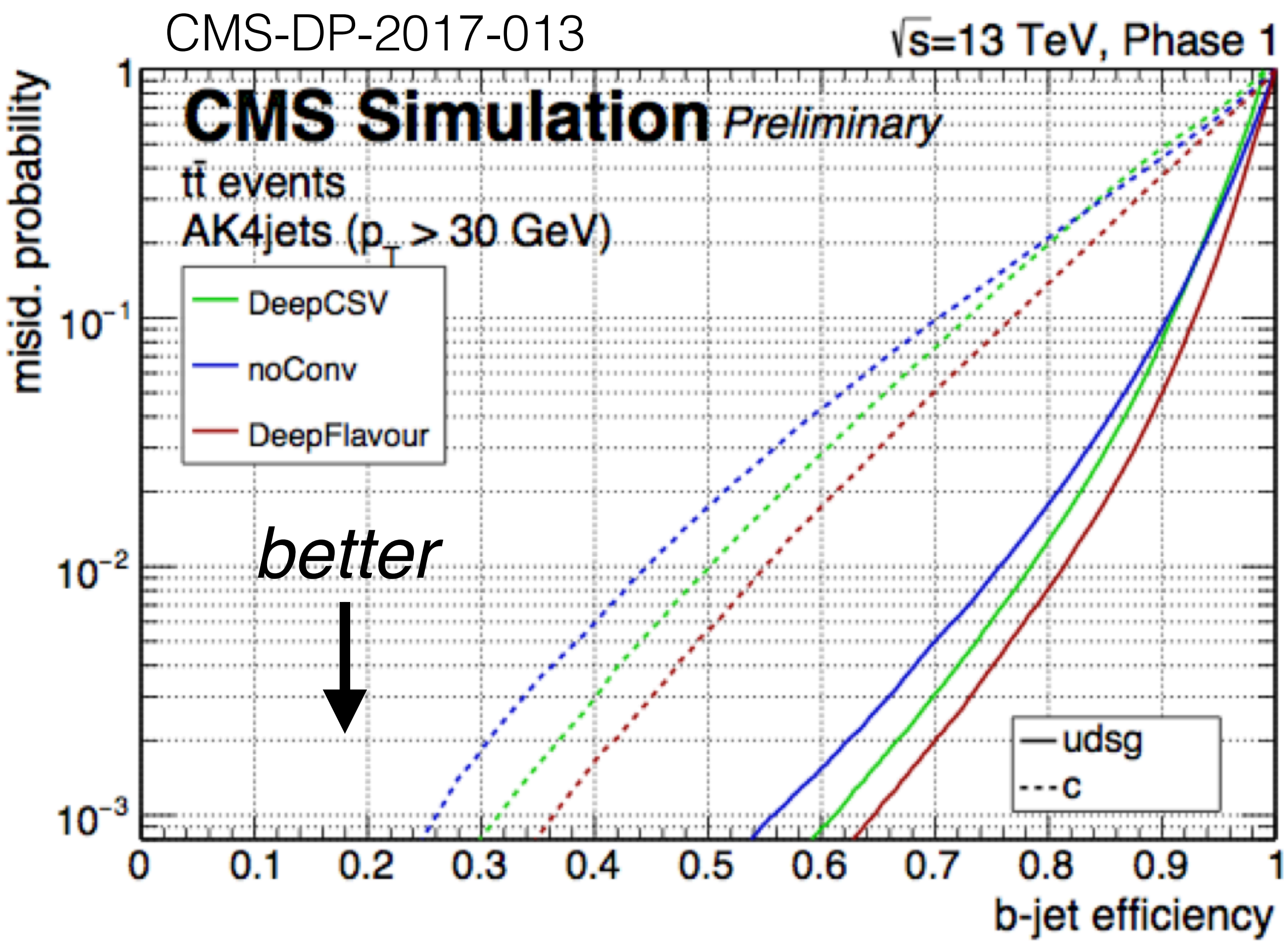
ATL-PHYS-PUB-2017-003

See also D. Guest et al., PRD 94 (2016) 112002

Hybrid methods



CMS-DP-2017-013



RNN + 1x1 CNNs for dimensionality reduction.

This reduction improved the performance of the overall classifier.

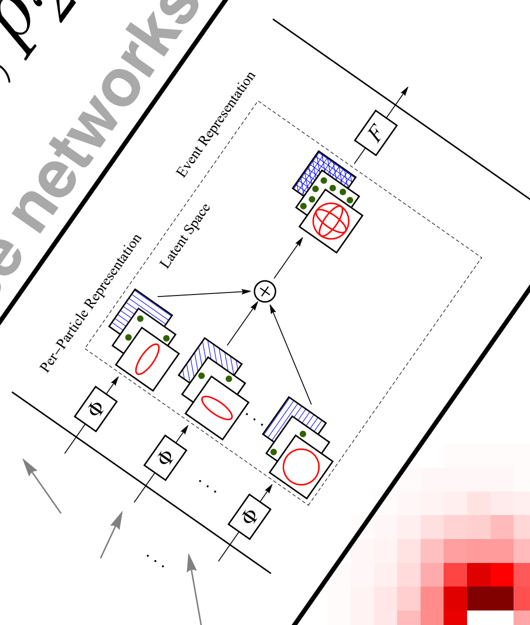
See 1709.04464 for image refs.

Deep neural networks for HEP classification

Fixed sets

$$J = \{p_1^\mu, p_2^\mu, \ldots, p_n^\mu\}$$

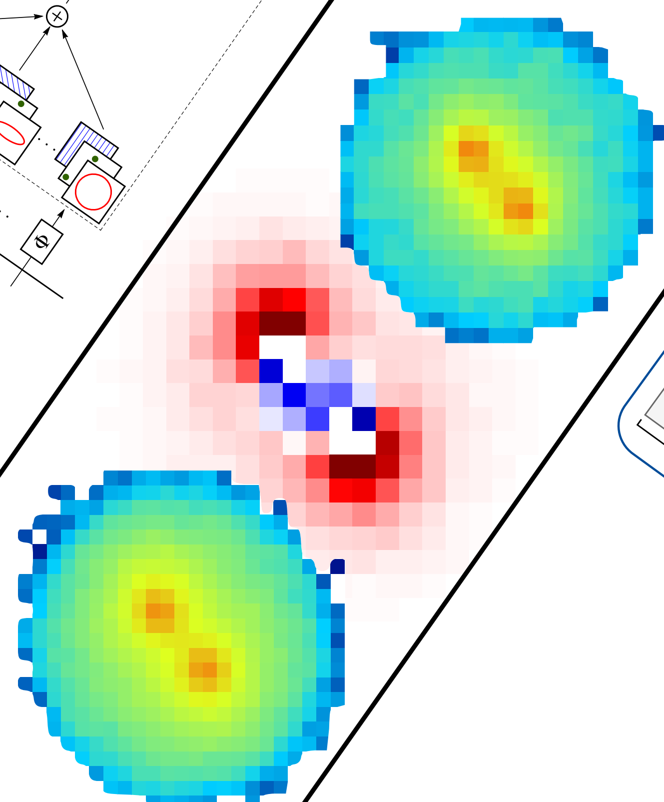
Dense networks



Variable sets

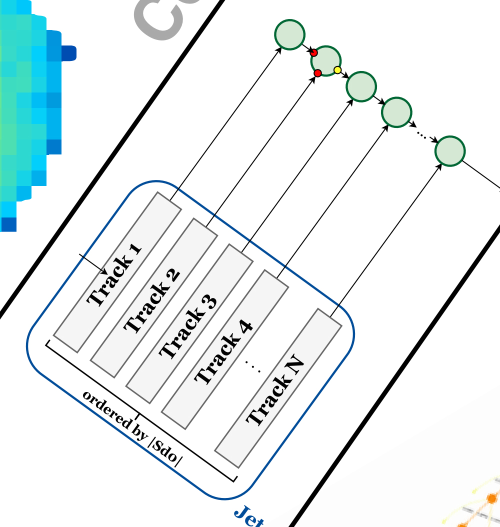
Deep sets

Images



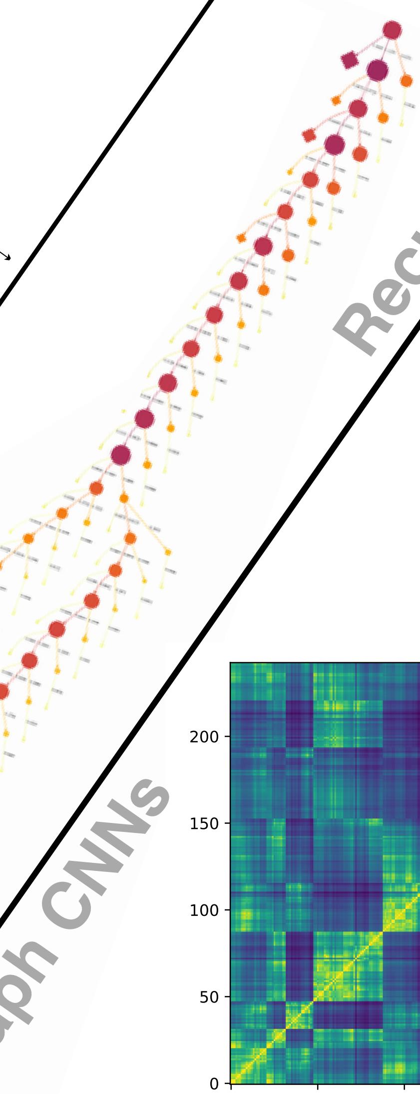
Convolutional NNs (CNNs)

Sequences



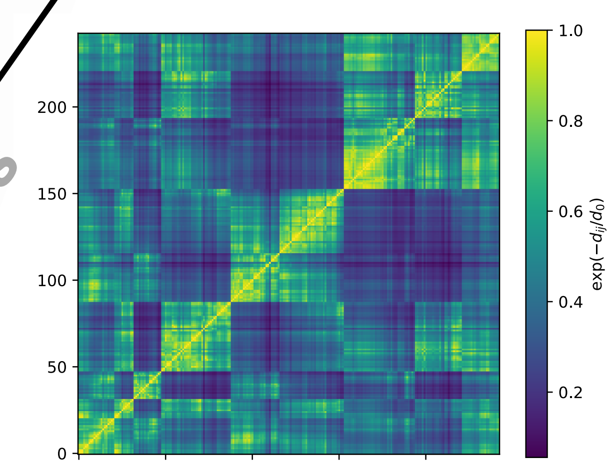
Recurrent NNs

Trees



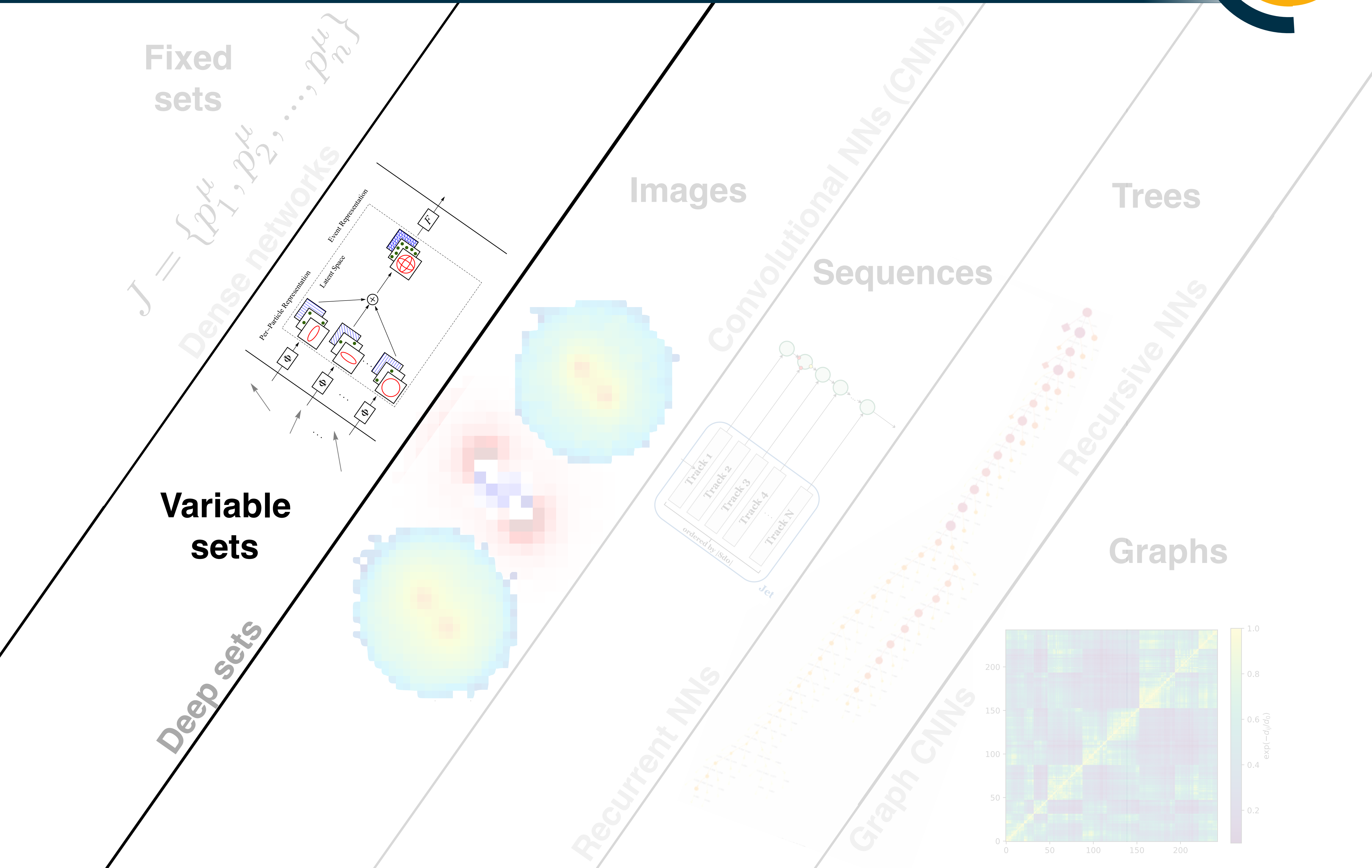
Recursive NNs

Graphs



Graph CNNs

Deep neural networks for HEP classification



Learning with sets

A challenge with sequence learning is that thanks to quantum mechanics, there is often no unique order.

A common scenario is that we have a variable-length **SET** of particles and we would like to learn from them directly.

Solution: set learning / point cloud approaches

Solution 1: Deep sets / Particle flow Networks

Factorize the problem into two networks: one that embeds the set into a fixed-length latent space and one that acts on a permutation invariant operation on that latent space:

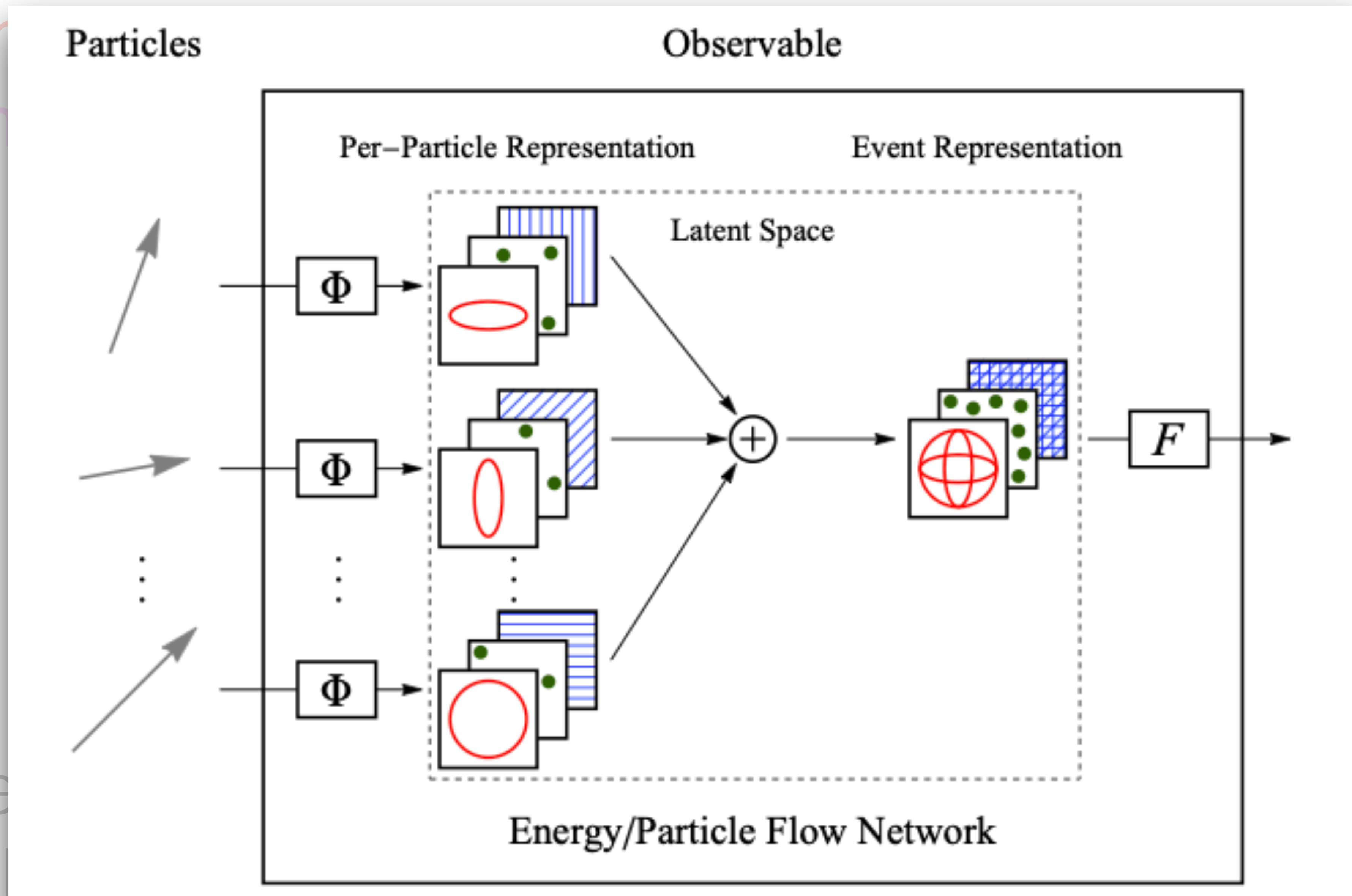
$$f(\{x_1, \ldots, x_M\}) = F\left(\sum_{i=1}^{M} \Phi(x_i)\right)$$

Due to the sum, this structure can operate on any length set and the order of the inputs doesn't matter.

M. Zaheer et al. https://arxiv.org/abs/1703.06114; P. Komiske, E. Metodiev, & J. Thaler, JHEP 01 (2019) 121

Solution 1: Deep sets / Particle flow Networks

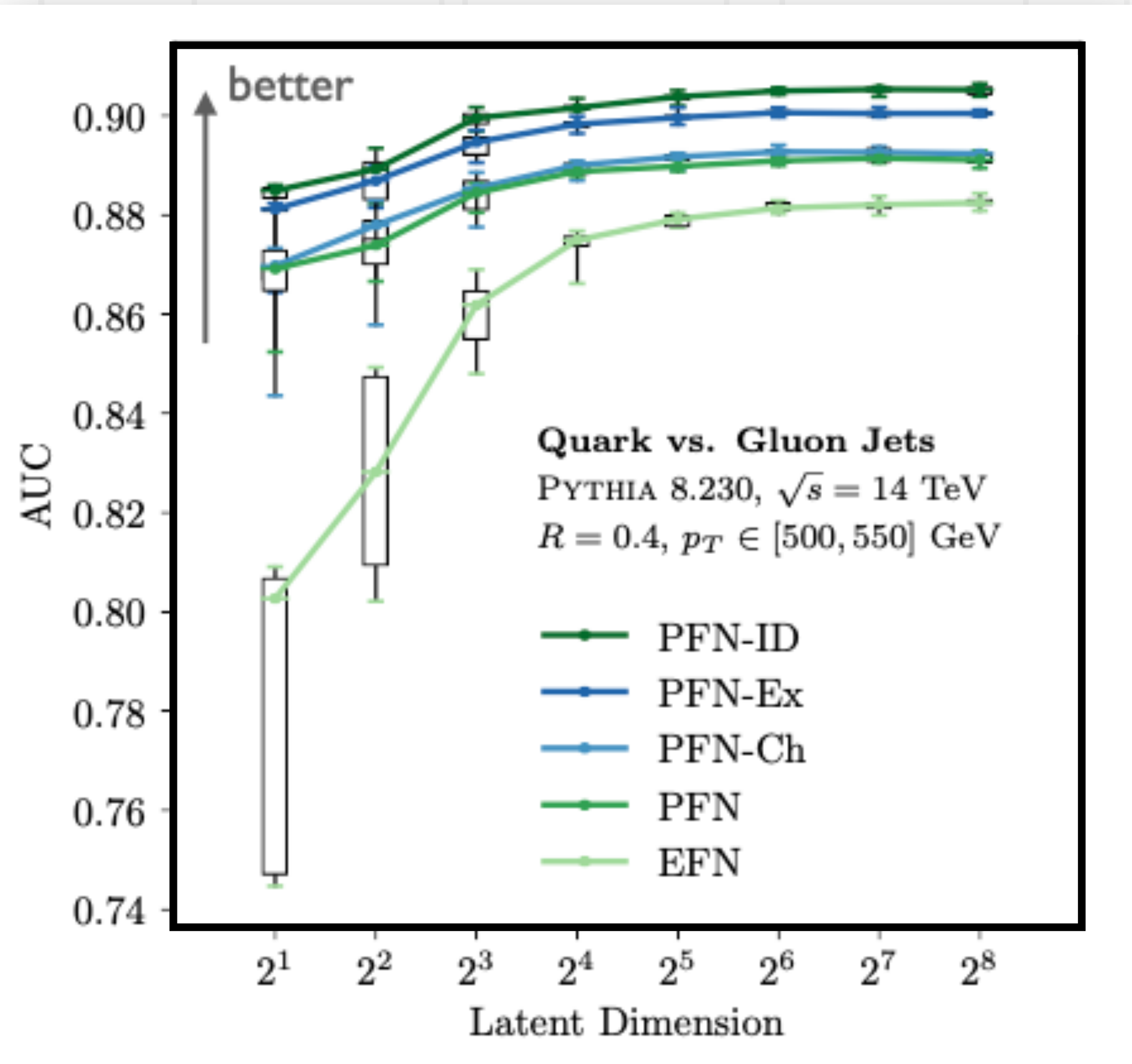
Factorize the problem into two networks: one that embeds the set i … t acts on a perm … space:



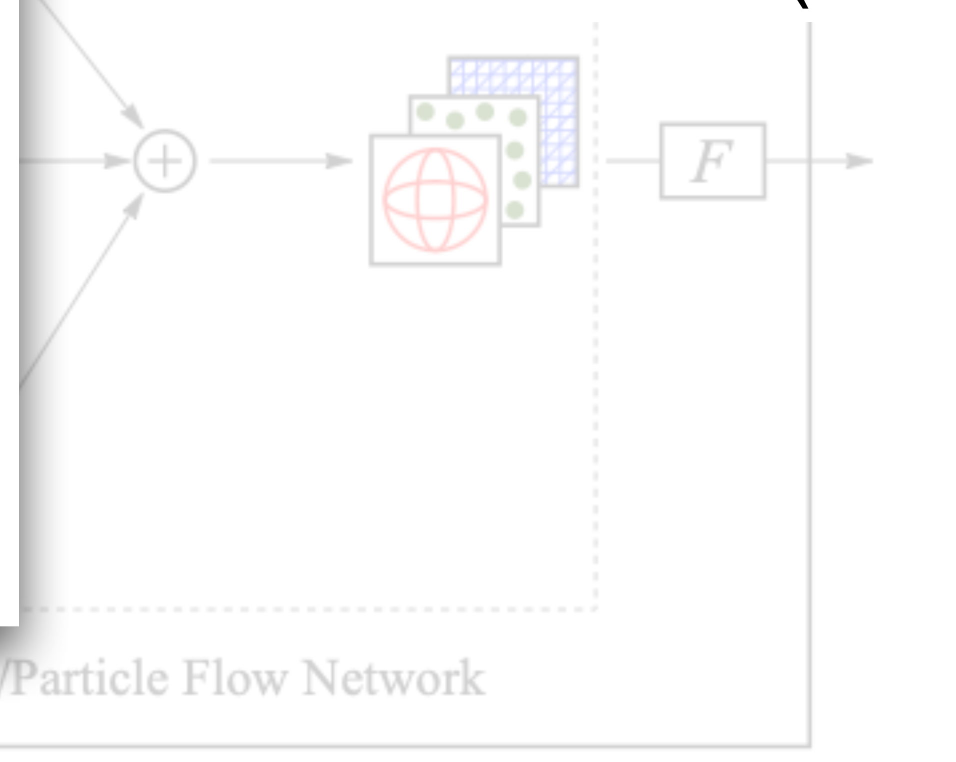
Due … any lengt … atter.

M. Zaheer et al. https://arxiv.org/abs/1703.06114; P. Komiske, E. Metodiev, & J. Thaler, JHEP 01 (2019) 121

Solution 1: Deep sets / Particle flow Networks

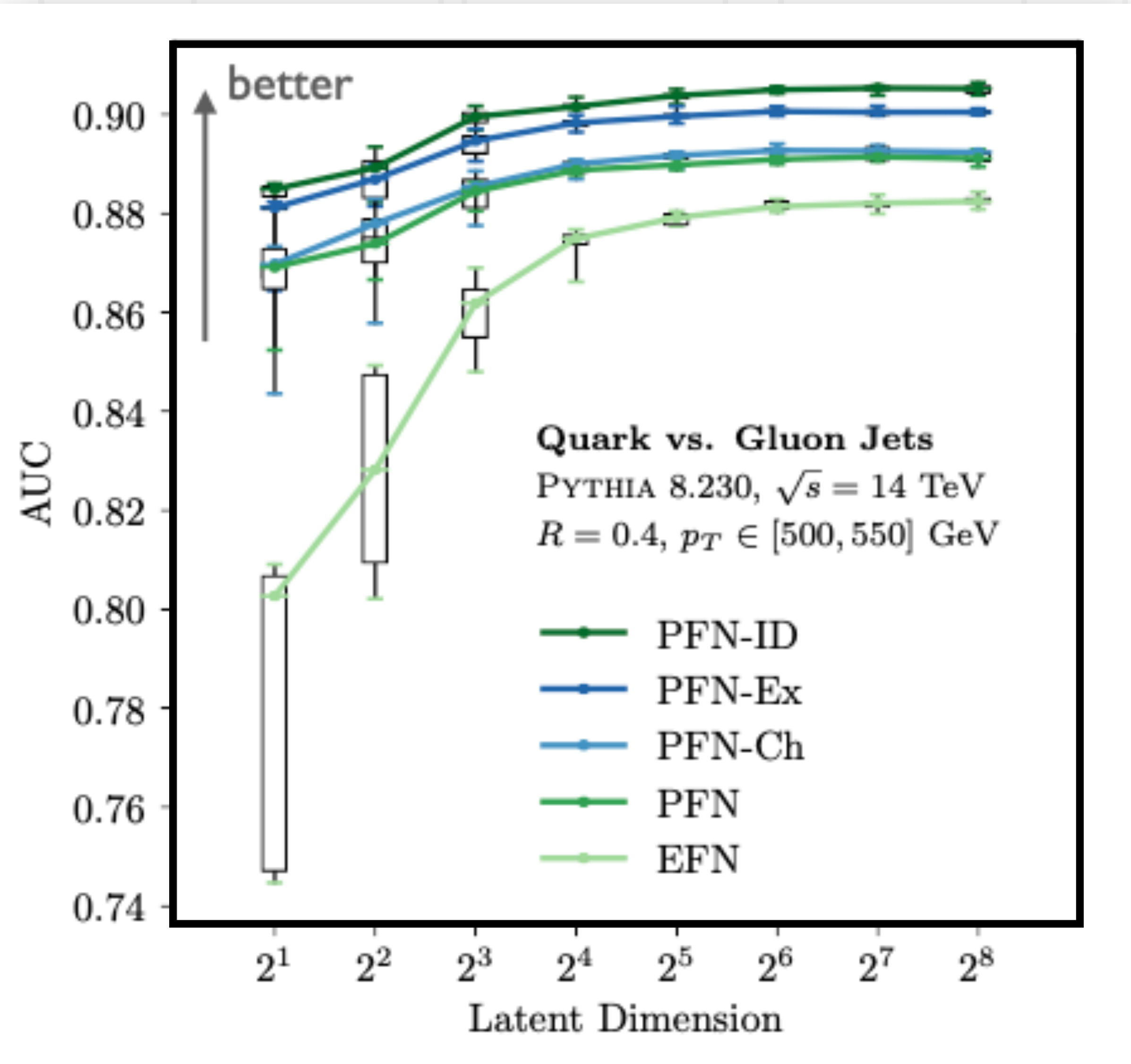


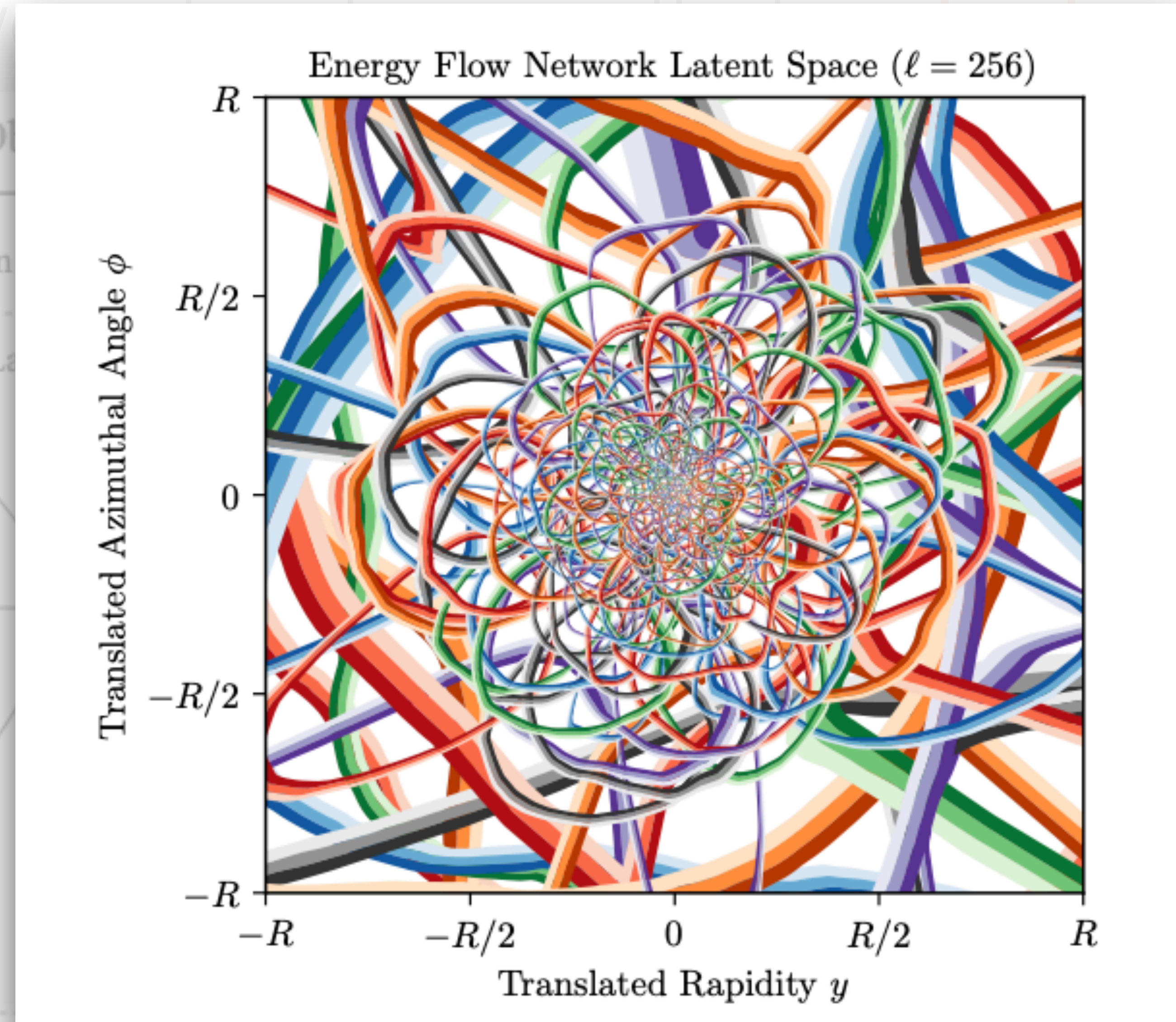
- Can readily incorporate per-particle features
- Can be made infrared and collinear safe (EFN) safe



M. Zaheer et al. https://arxiv.org/abs/1703.06114; P. Komiske, E. Metodiev, & J. Thaler, JHEP 01 (2019) 121

Solution 1: Deep sets / Particle flow Networks

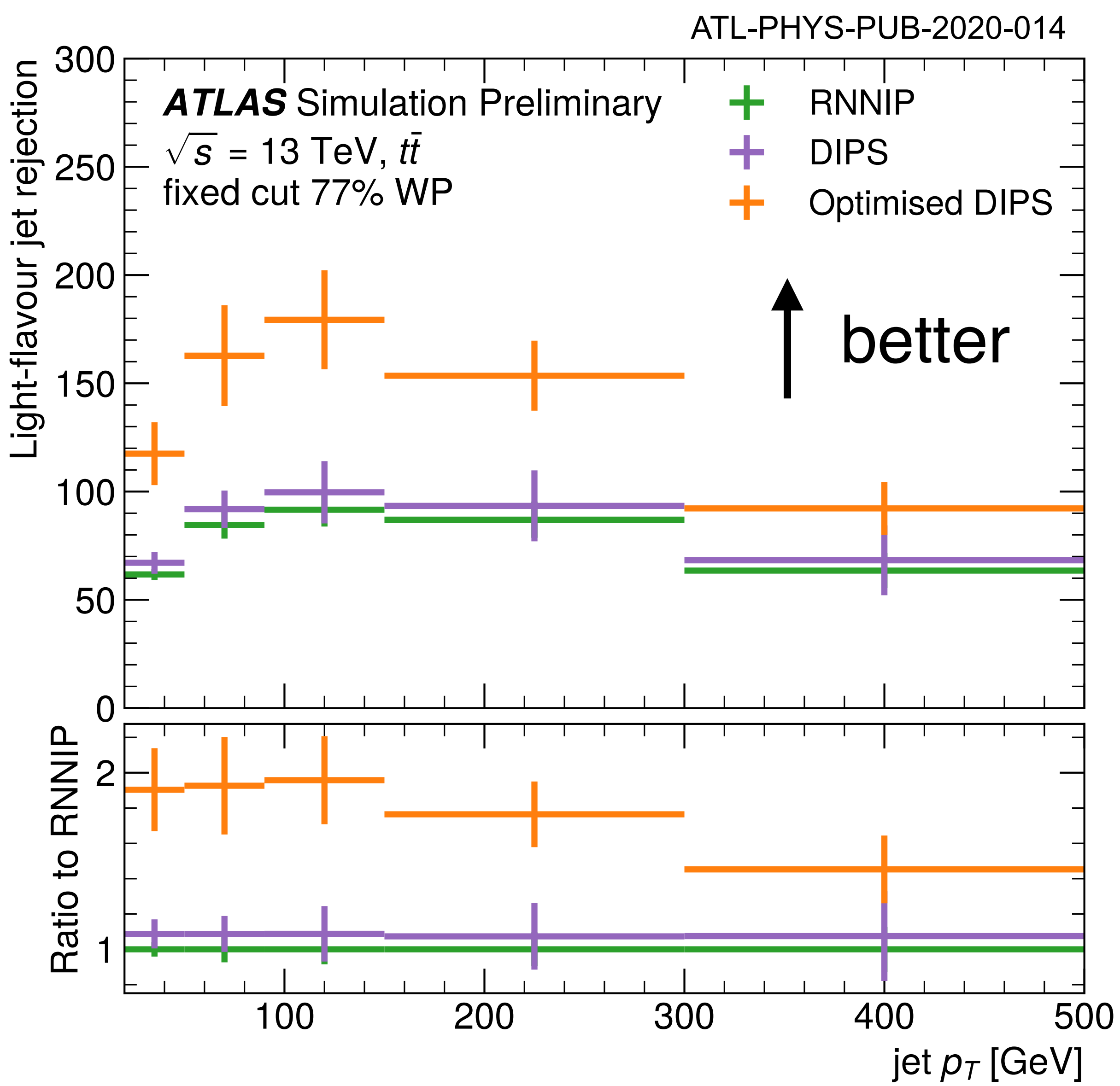




Latent space in IRC safe case is interpretable (and predictable!)

M. Zaheer et al. https://arxiv.org/abs/1703.06114; P. Komiske, E. Metodiev, & J. Thaler, JHEP 01 (2019) 121

Solution 1: Deep sets / Particle flow Networks



Faster to train than RNN so can do R&D on input features to improve overall performance.

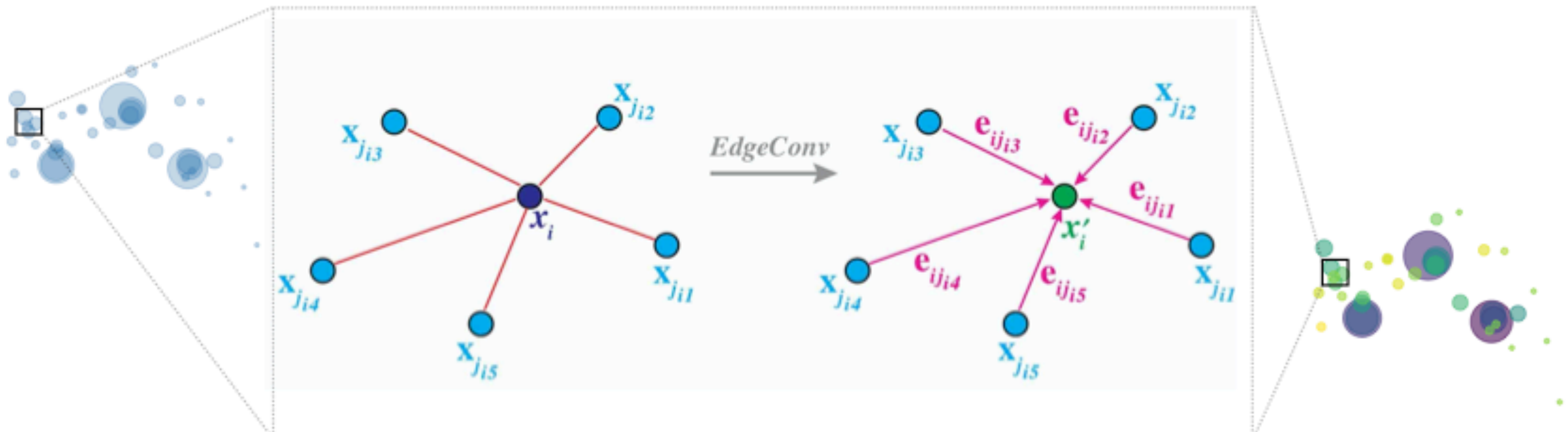
Latent space in IRC safe case is interpretable (and predictable!)

M. Zaheer et al. https://arxiv.org/abs/1703.06114; P. Komiske, E. Metodiev, & J. Thaler, JHEP 01 (2019) 121

Solution 2: Graph methods

Classic CNNs operate on a fixed grid and are not invariant under the permutation of points

Can generalize CNNs to act on graphs



Need to define distances using particle properties

e.g. Y. Wang et al. https://arxiv.org/abs/1801.07829 and H. Qu and L. Gouskos, PRD 101 (2020) 056019

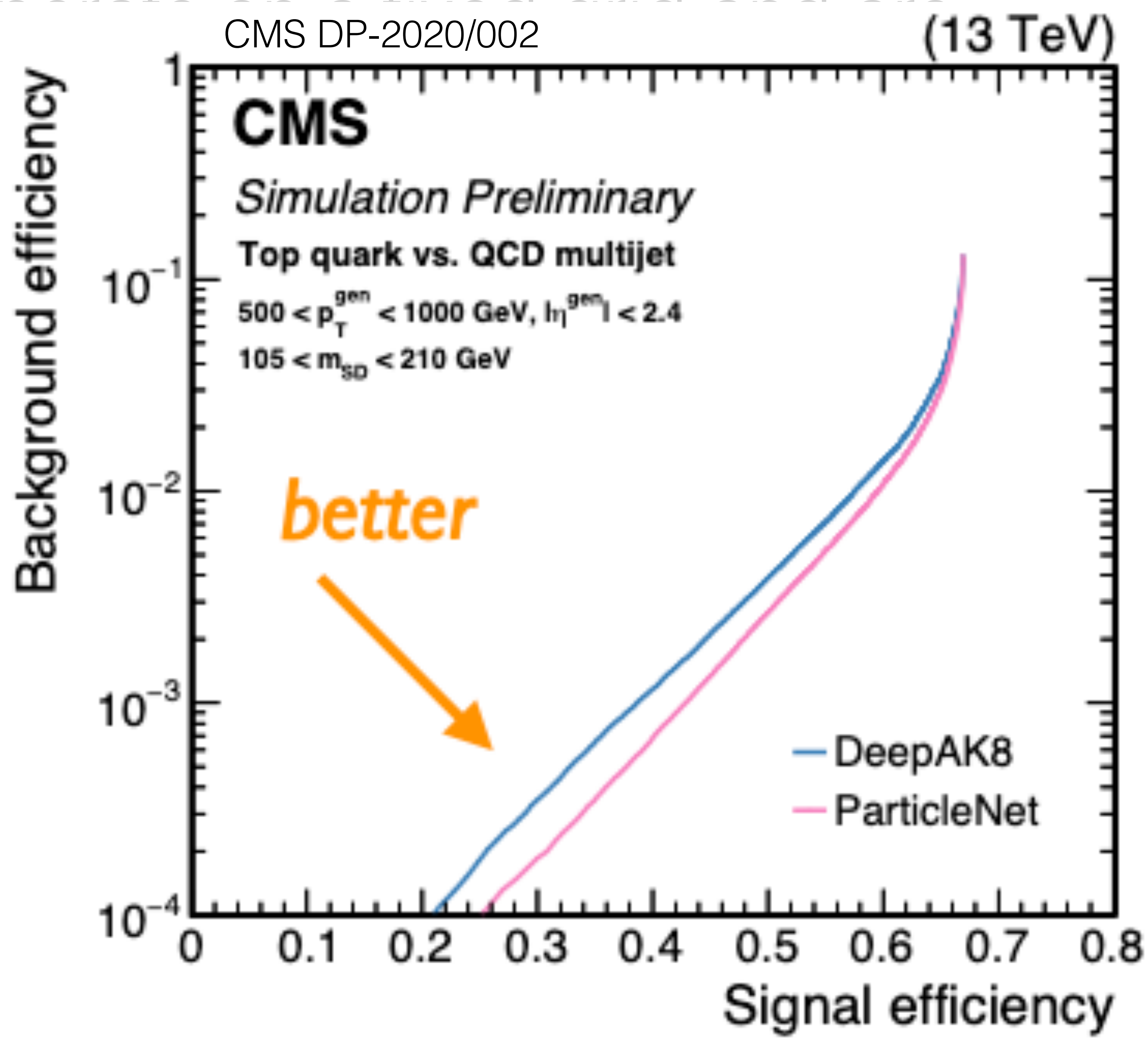
Solution 2: Graph methods

Classic CNNs o

not invariant un

Can generaliz

Competitive performance to other state-of-the-art methods



Need to define dis

Y. Wang et al. https://arxiv.org/abs/1801.07829 and H. Qu and L. Gouskos, PRD 101 (2020) 056019

Deep neural networks for HEP classification

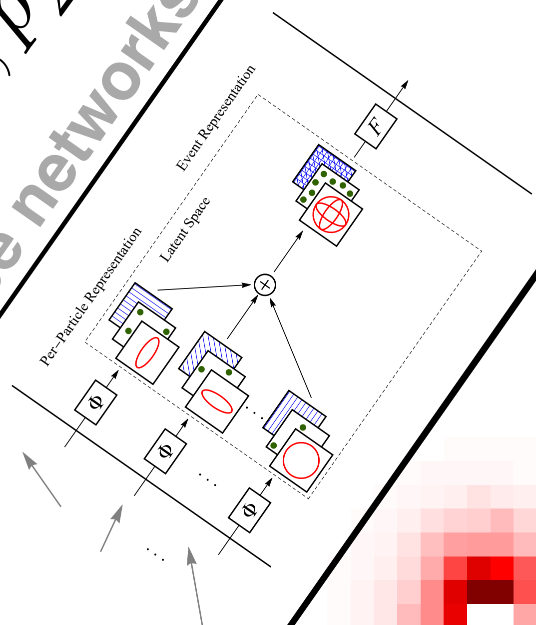
Fixed sets

$J = \{p_1^\mu, p_2^\mu, \dots, p_n^\mu\}$

Dense networks

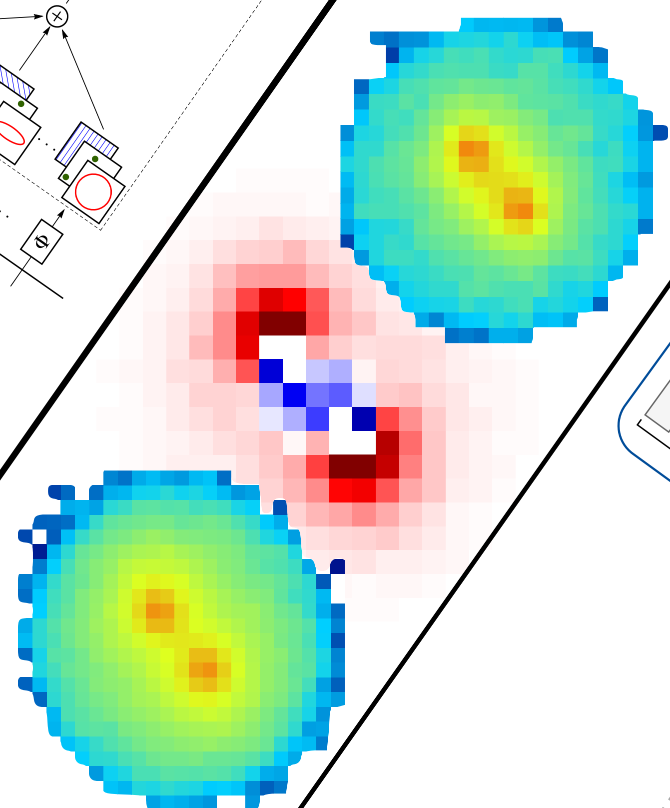
Variable sets

Deep sets



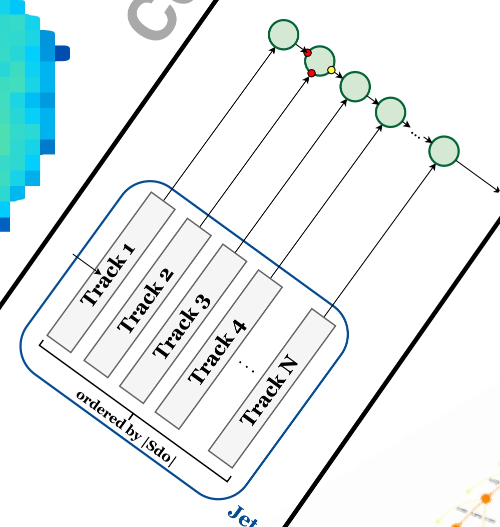
Images

Convolutional NNs (CNNs)



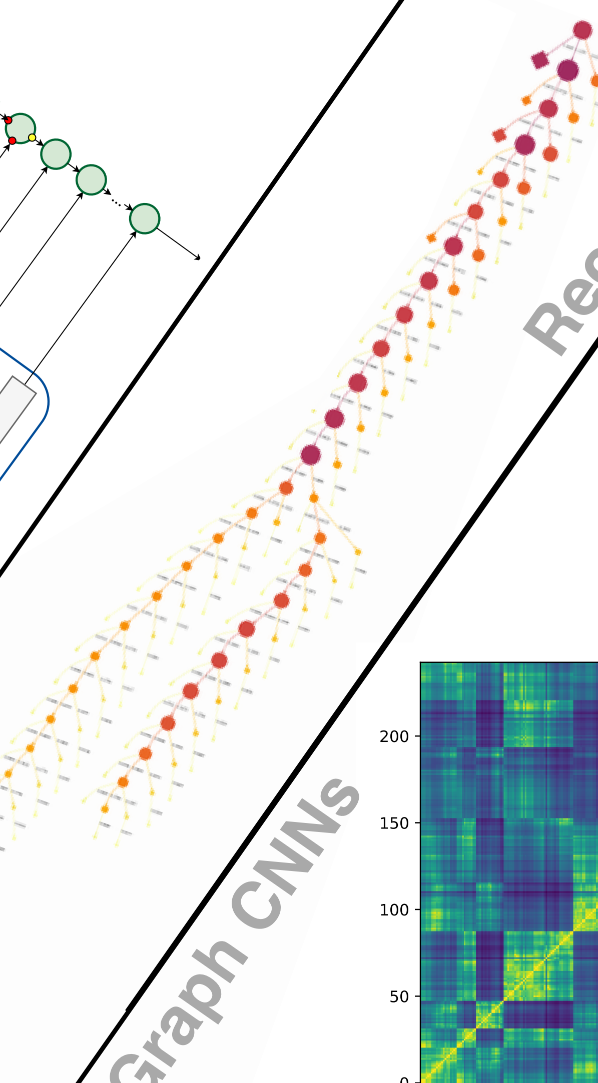
Sequences

Recurrent NNs



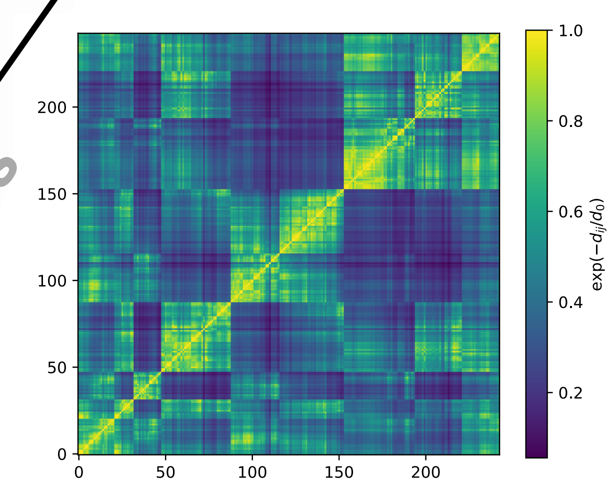
Trees

Recursive NNs



Graphs

Graph CNNs



Beyond Classification

Classification

provide
examples
for training

arbitrarily
many
categories

Generation ← **Regression**

map noise
to structure

+related topics like anomaly detection, simulation-based inference, …

Beyond Classification

Classification

provide
examples
for training

arbitrarily
many
categories

Generation

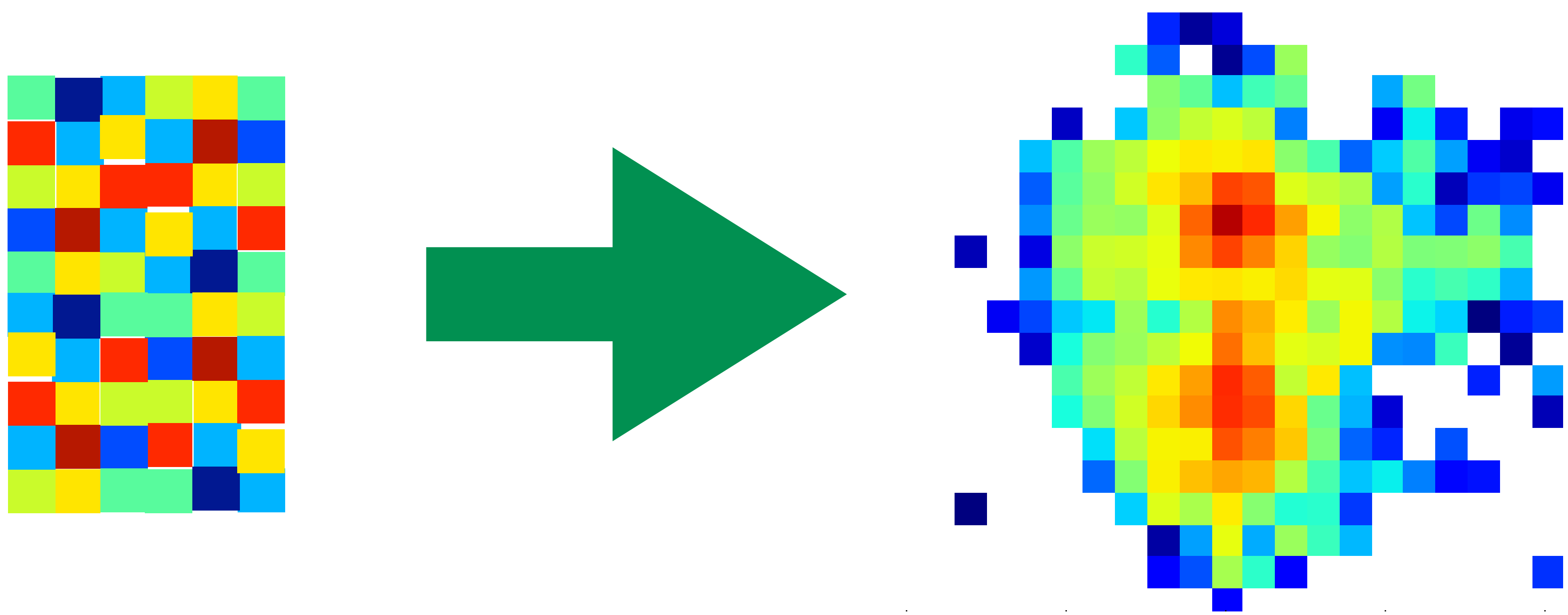
Regression

map noise
to structure

+related topics like anomaly detection, simulation-based inference, …

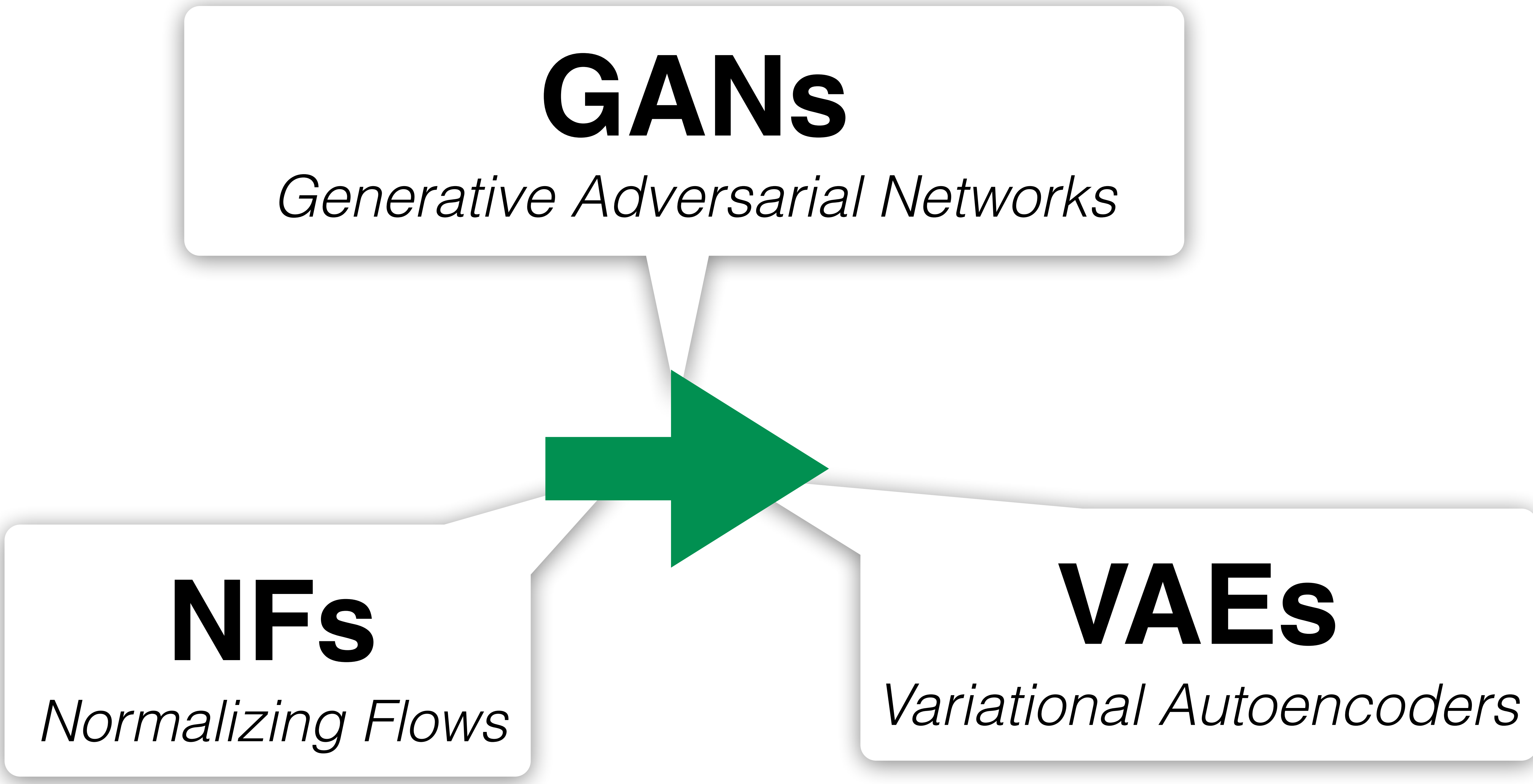
Introduction: generative models

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



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

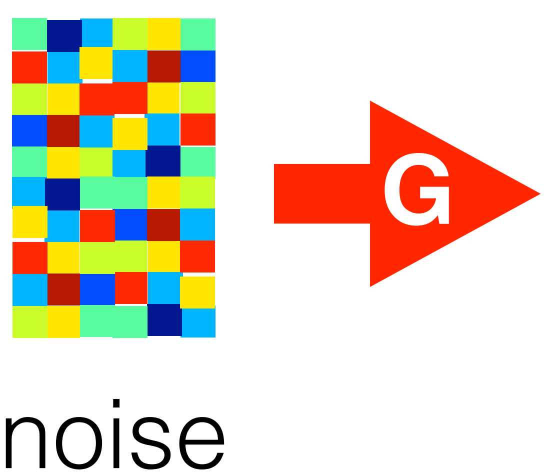
Tools

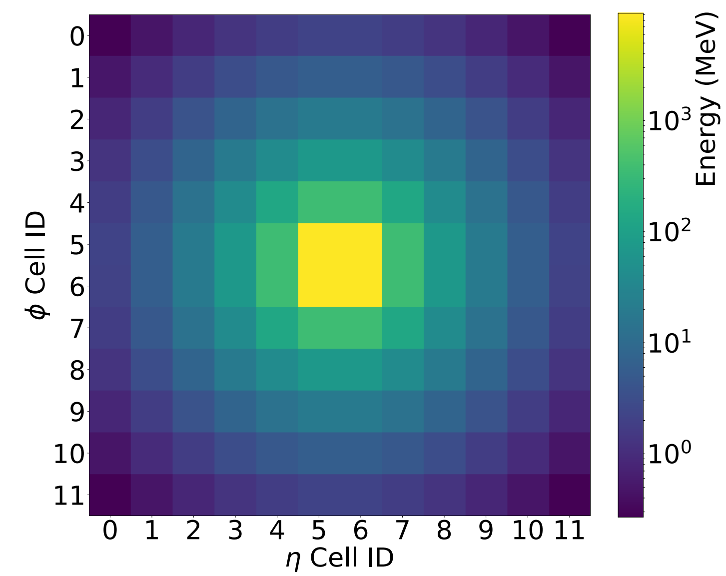


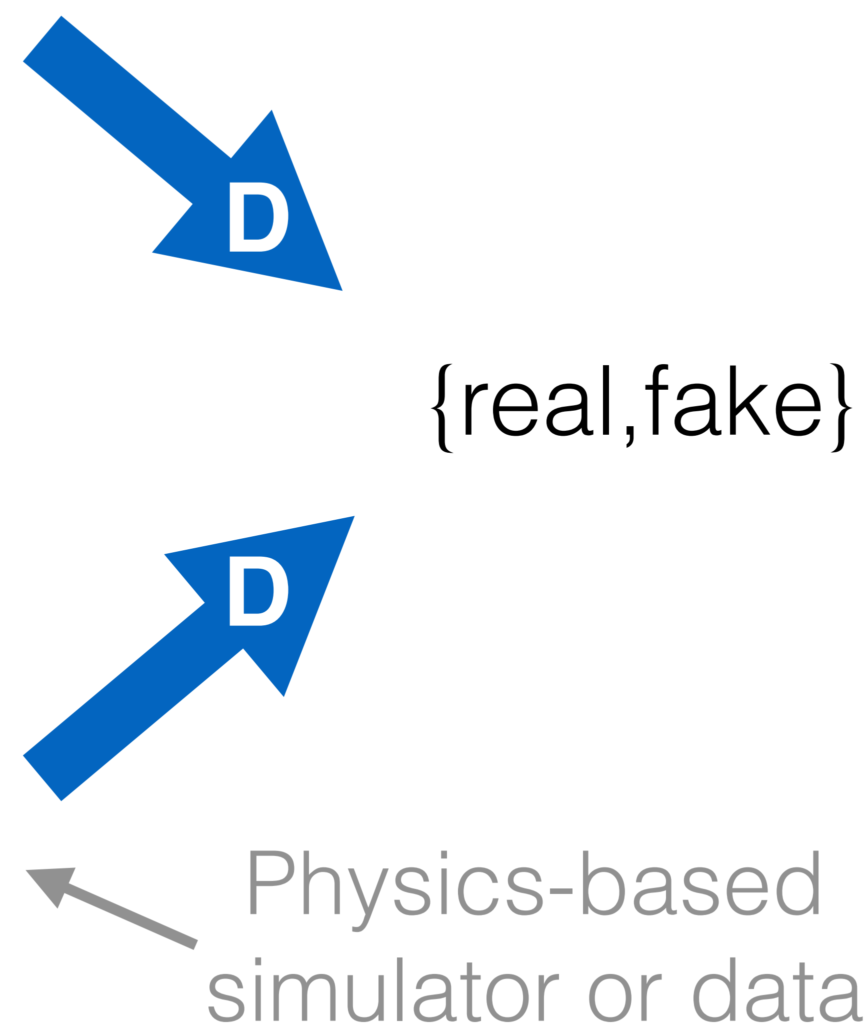
Deep generative models: the map is a deep neural network.

Introduction: GANs

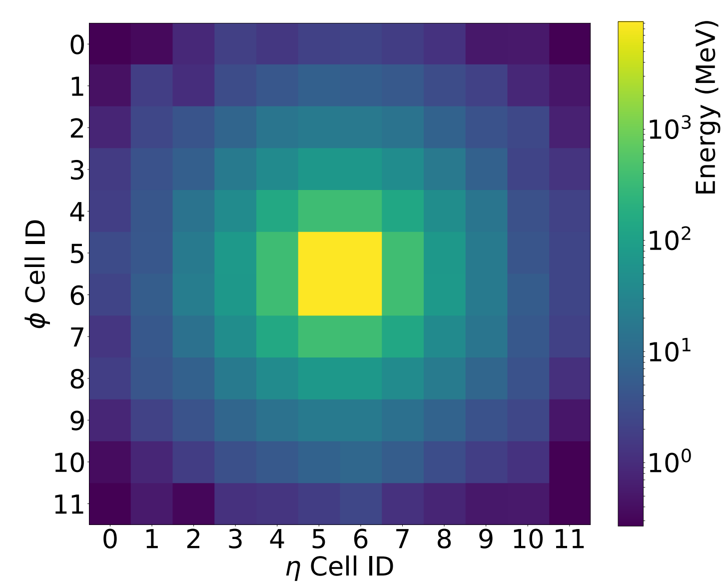
Generative Adversarial Networks (GANs):
A two-network game where one ***maps noise to structure*** *and one* ***classifies images as fake or real****.*





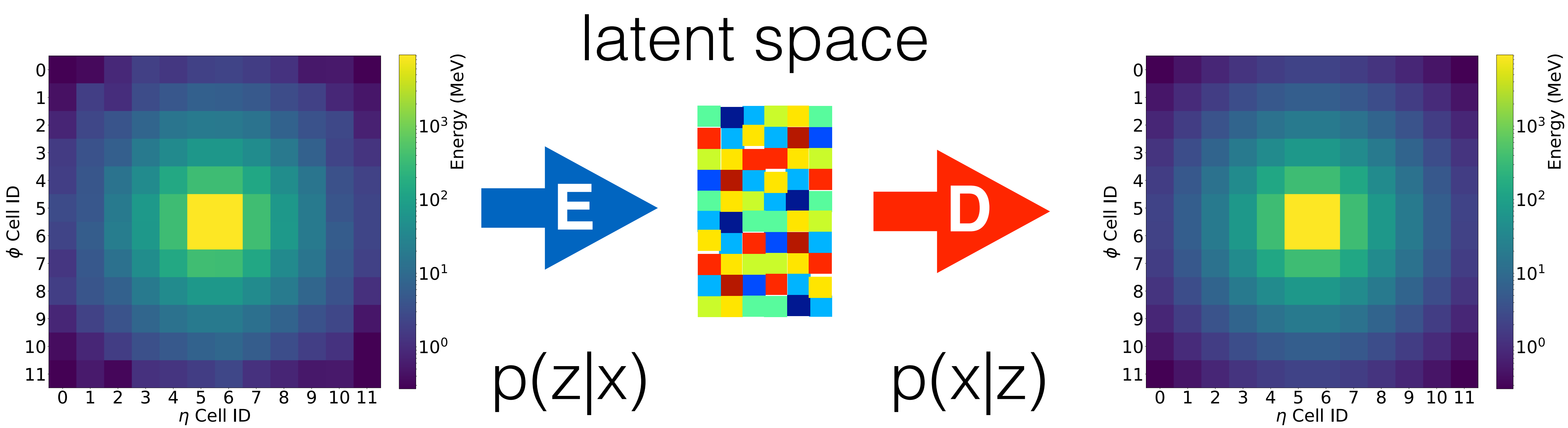


When **D** is maximally confused, **G** will be a good generator



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

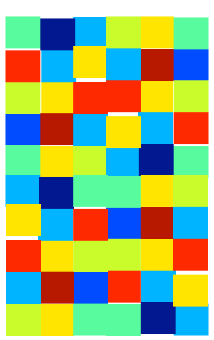
Probabilistic ***encoder***

Probabilistic ***decoder***

Introduction: NFs

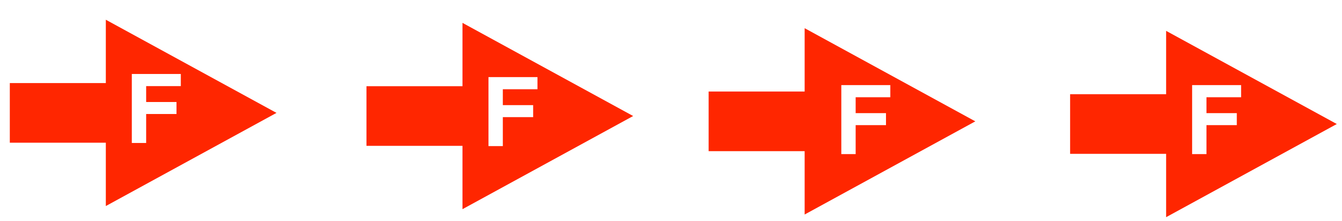
Normalizing Flows (NFs):
A series of invertible transformations mapping a known density into the data density.

Optimize via maximum likelihood



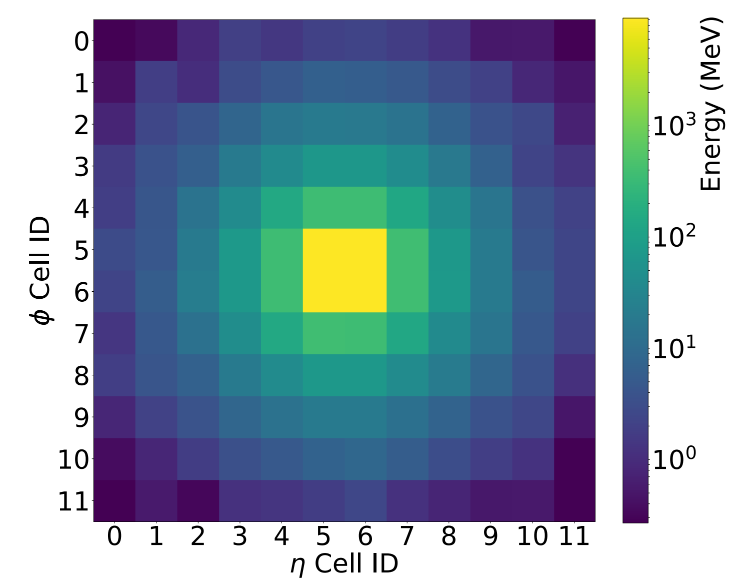
latent space

$p(z)$



Invertible transformations with tractable Jacobians

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

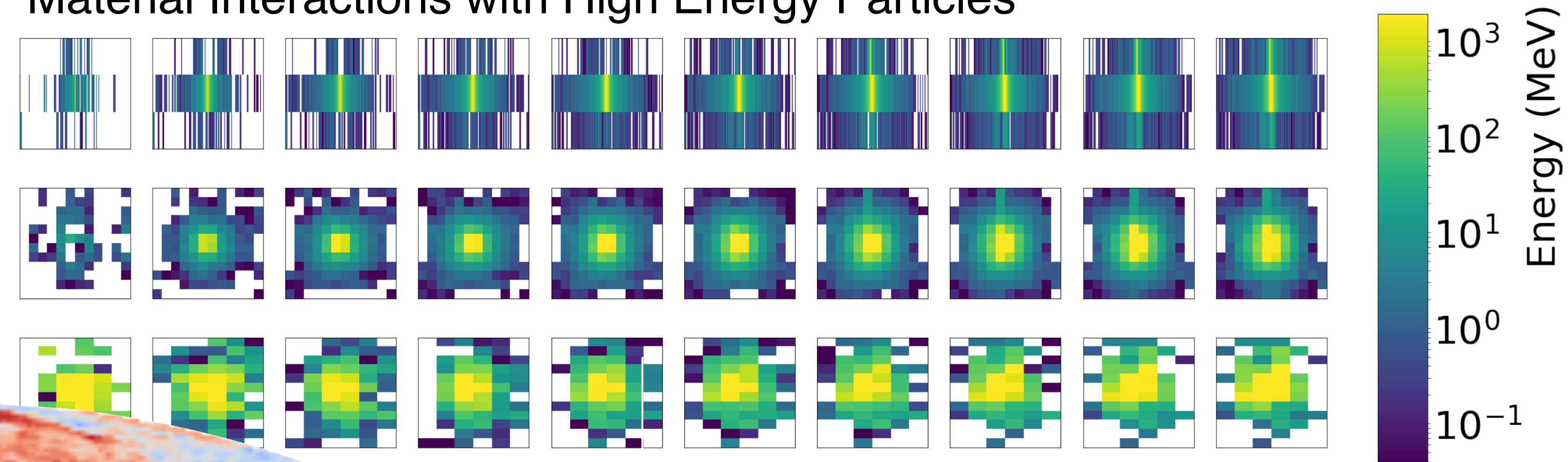


$p(x)$

Generative Models for Particle/Nuclear/Astro

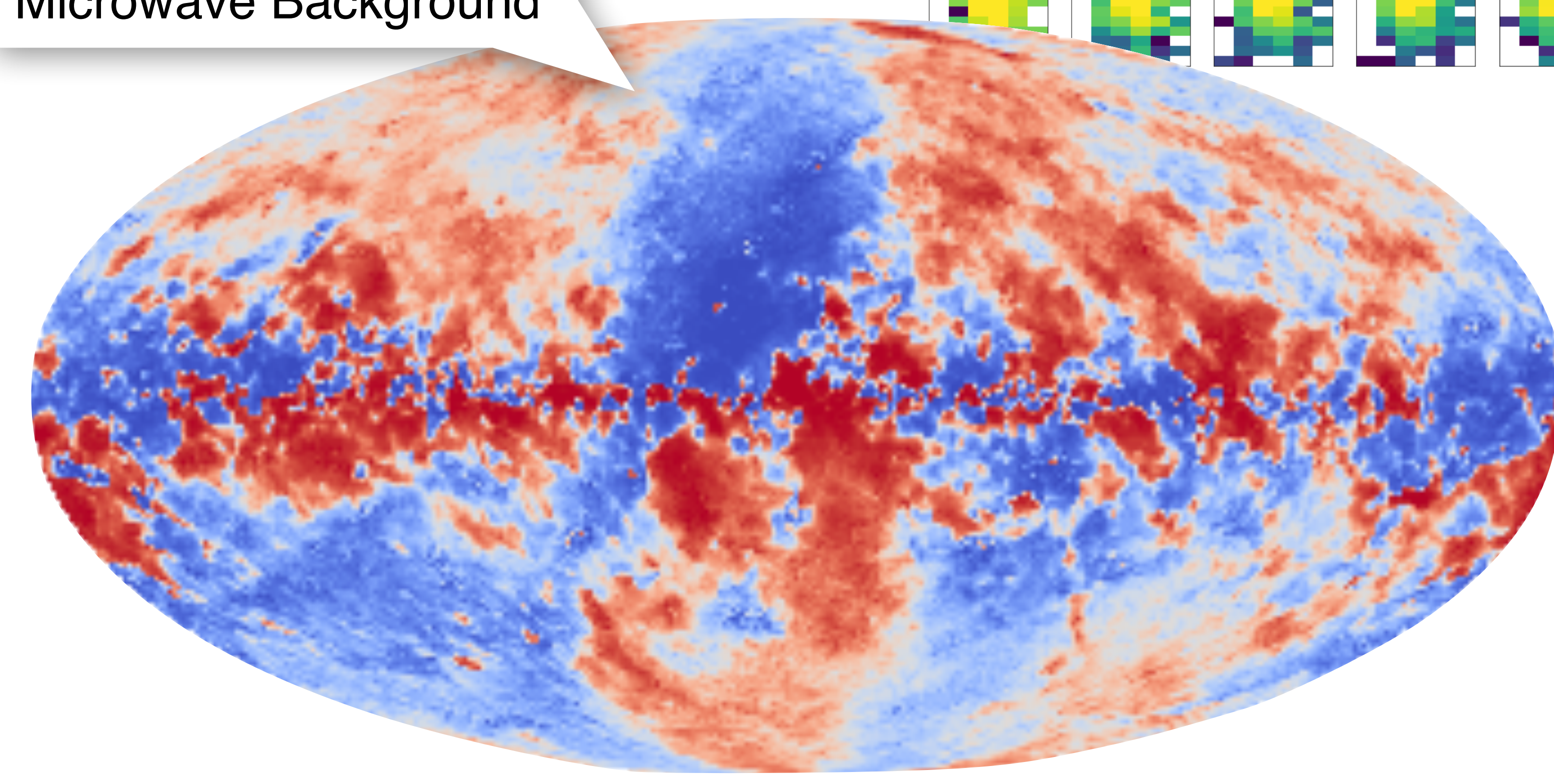
All of these pictures are fake!

Material Interactions with High Energy Particles



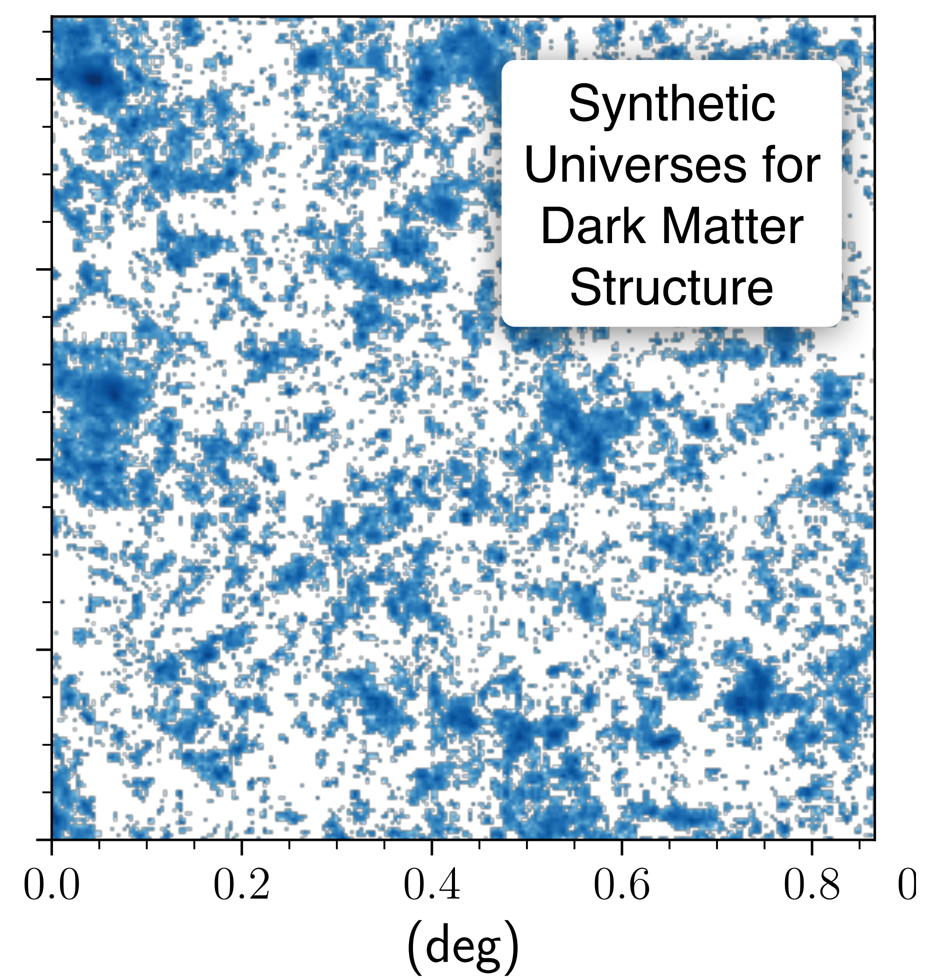
M. Paganini, L. De Oliveira, B. Nachman, Phys. Rev. Lett. 120 (2018) 042003

Synthetic Galactic radiation for Cosmic Microwave Background



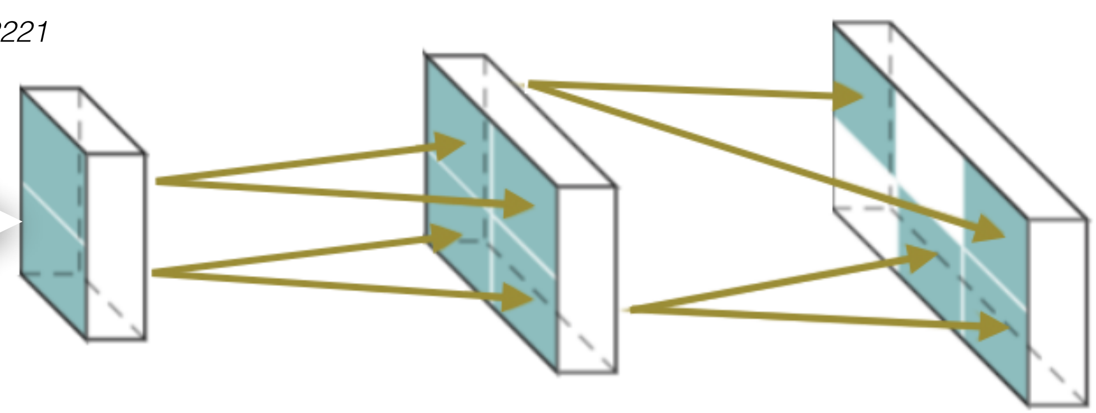
N. Krachmalnicoff and G. Puglisi, arXiv:2011.02221

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

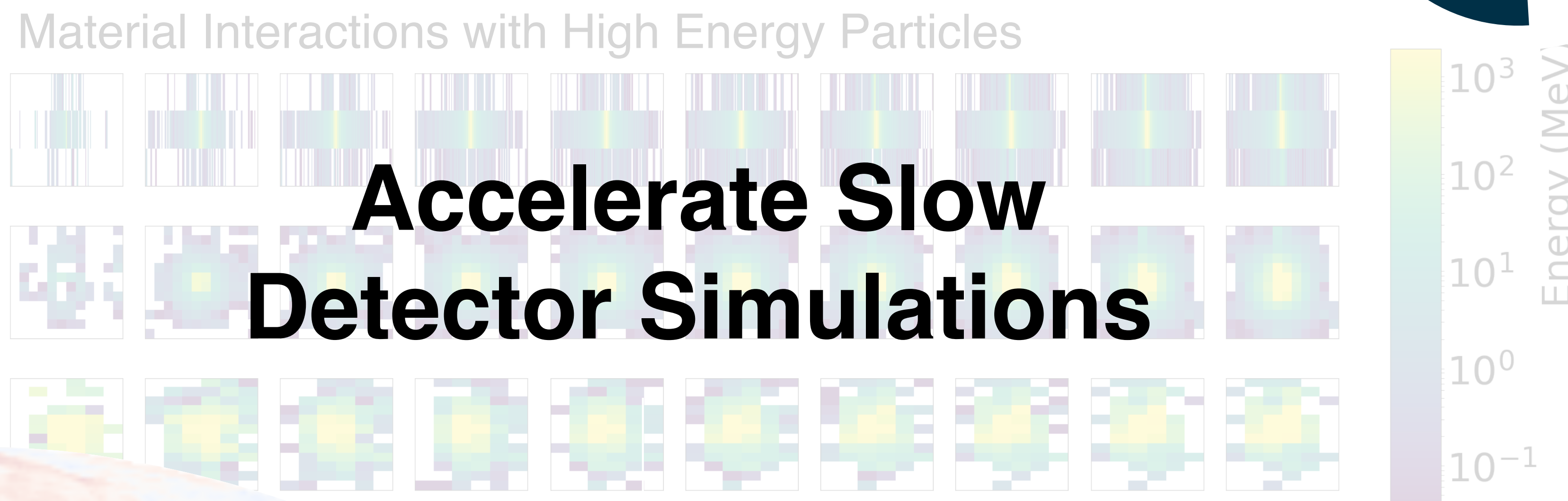


Y. S. Lai, D. Neill, M. Płoskoń, F. Ringer, arXiv:2012.06582

Generative Models for Particle/Nuclear/Astro

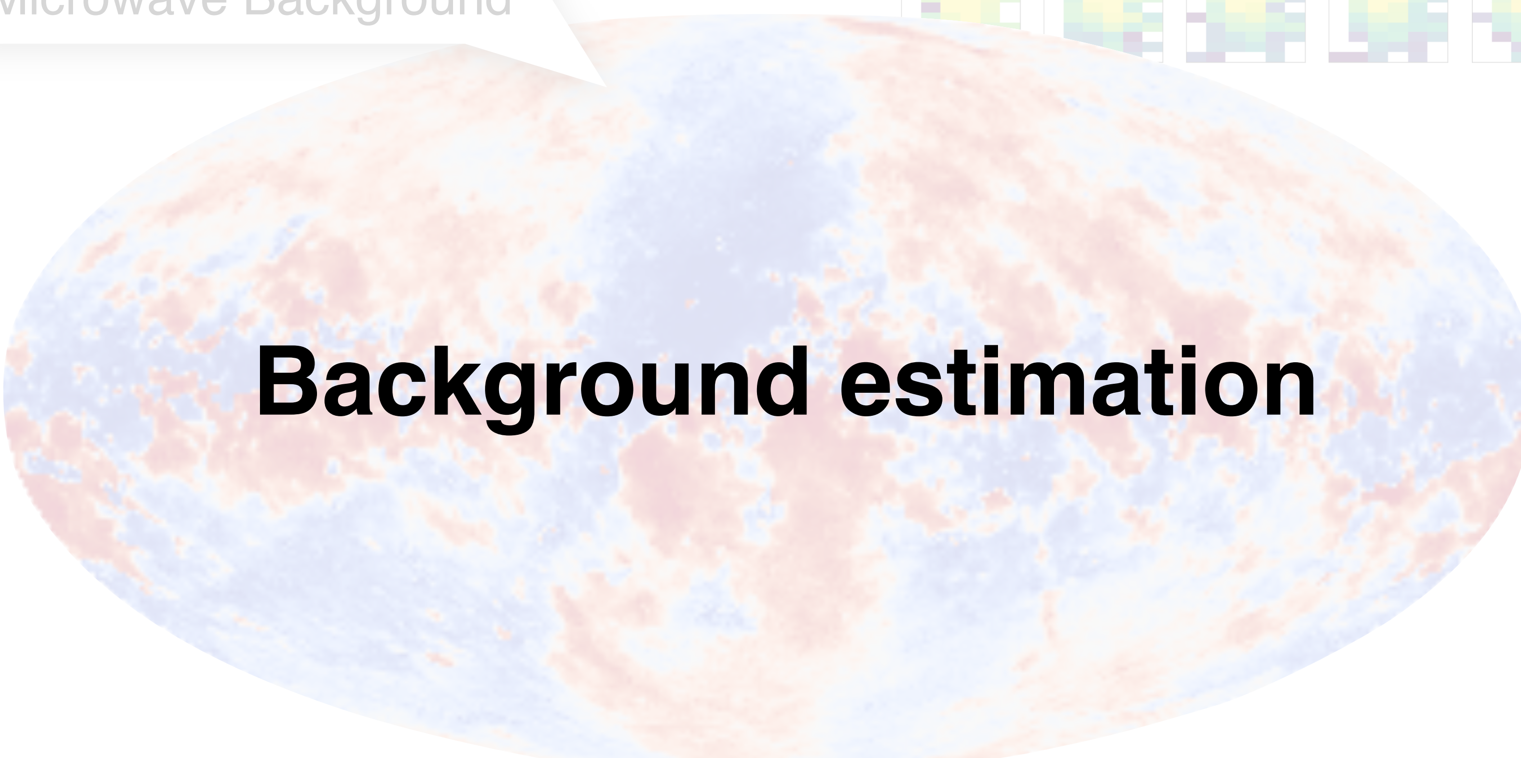
All of these pictures are fake!

Synthetic Galactic radiation for Cosmic Microwave Background



Accelerate Slow Detector Simulations

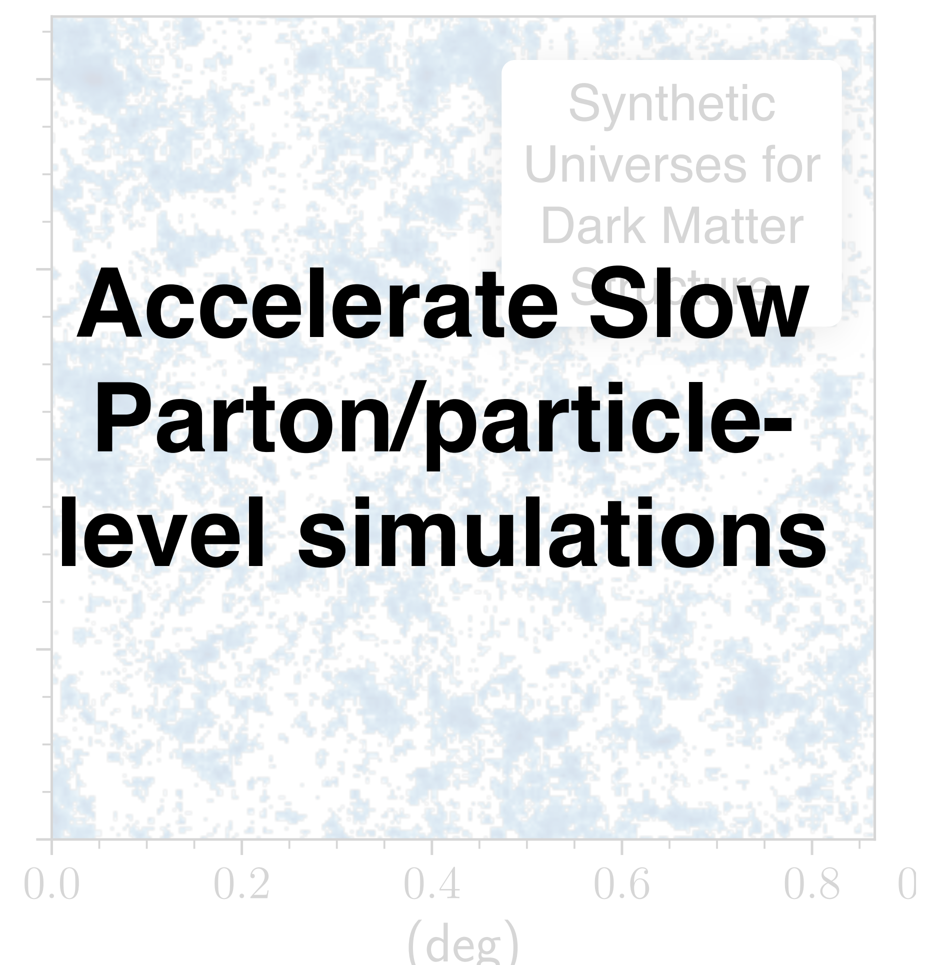
M. Paganini, L. De Oliveira, B. Nachman, Phys. Rev. Lett. 120 (2018) 042003



Background estimation

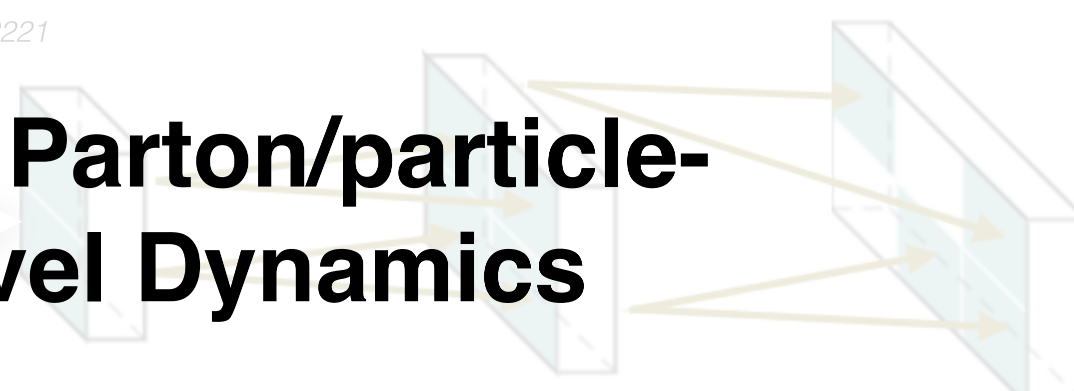
N. Krachmalnicoff and G. Puglisi, arXiv:2011.02221

Synthetic Universes for Dark Matter Structure



Accelerate Slow Parton/particle-level simulations

The Structure of Radiation in the Quantum Strong Force



Infer Parton/particle-level Dynamics

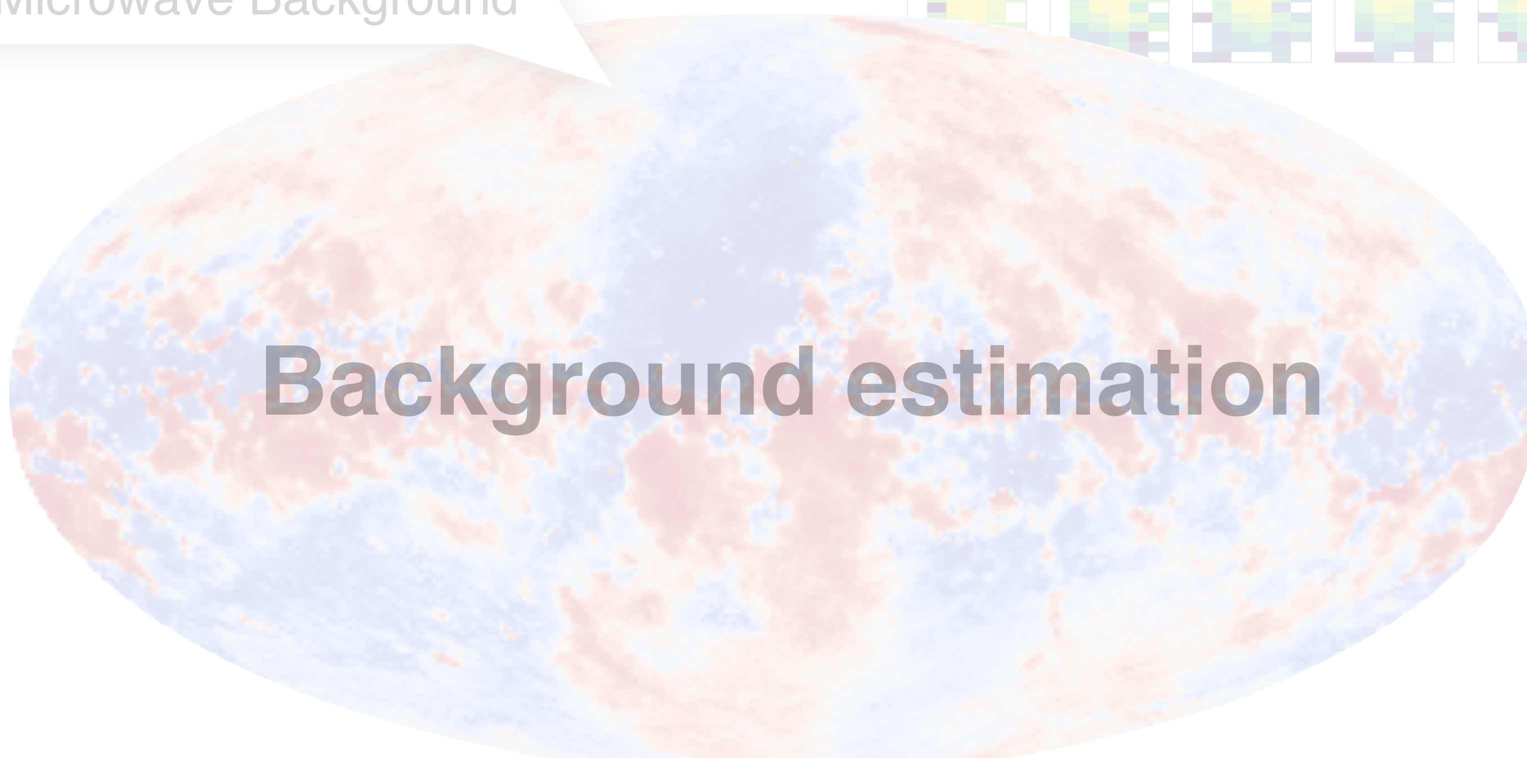
Y. S. Lai, D. Neill, M. Płoskoń, F. Ringer, arXiv:2012.06582

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

Generative Models for Particle/Nuclear/Astro

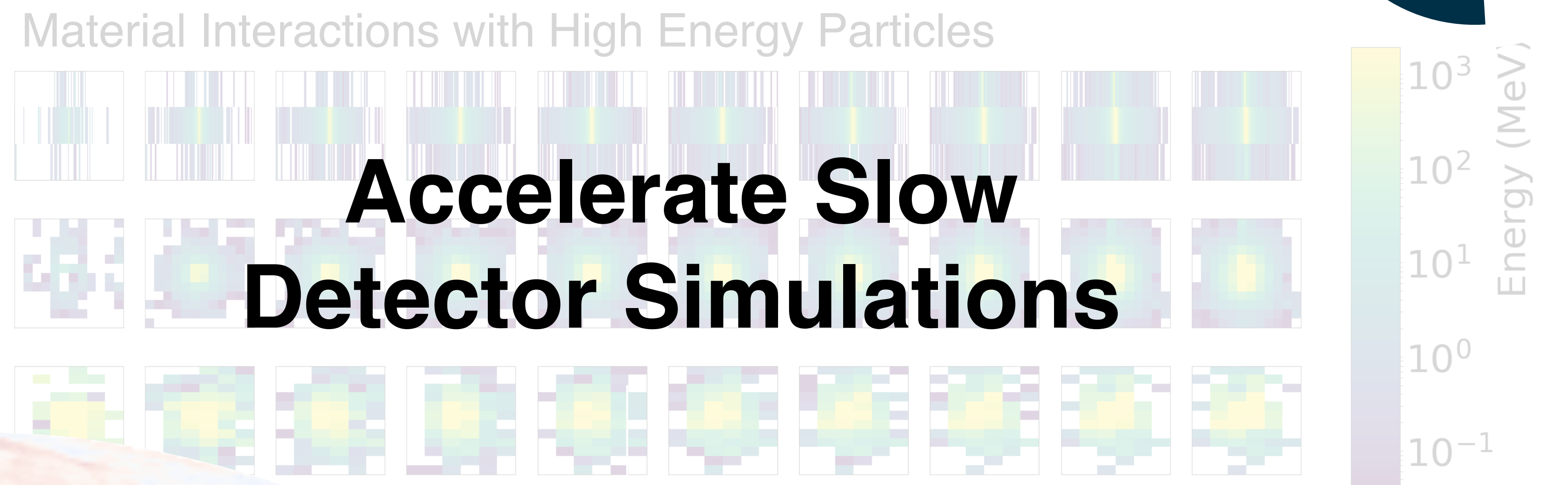
All of these pictures are fake!

Synthetic Galactic radiation for Cosmic Microwave Background



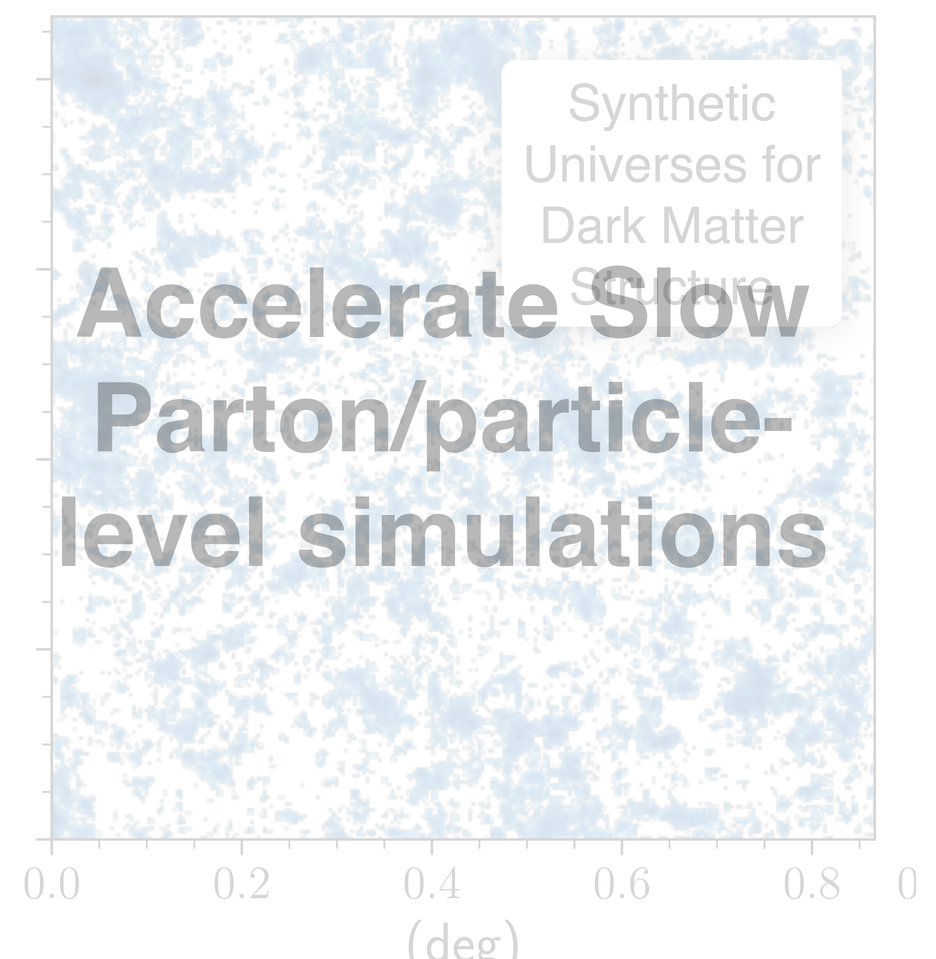
Background estimation

N. Krachmalnicoff and G. Puglisi, arXiv:2011.02221



Accelerate Slow Detector Simulations

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

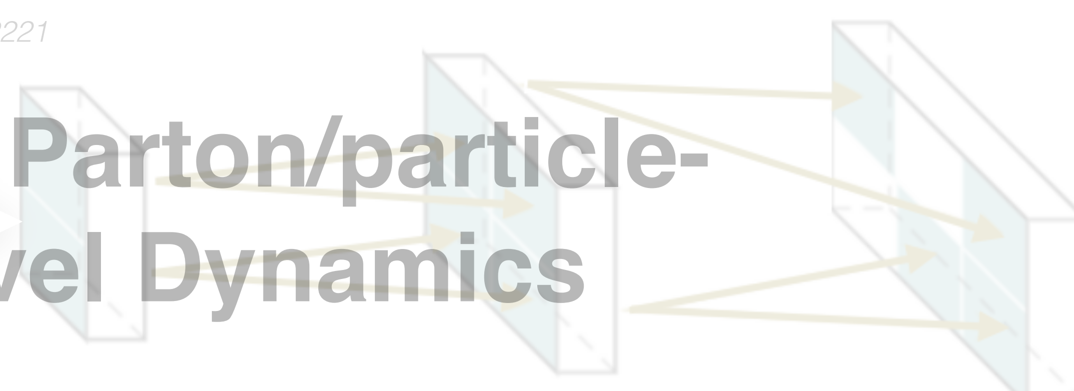


Synthetic Universes for Dark Matter Structure

Accelerate Slow Parton/particle-level simulations

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

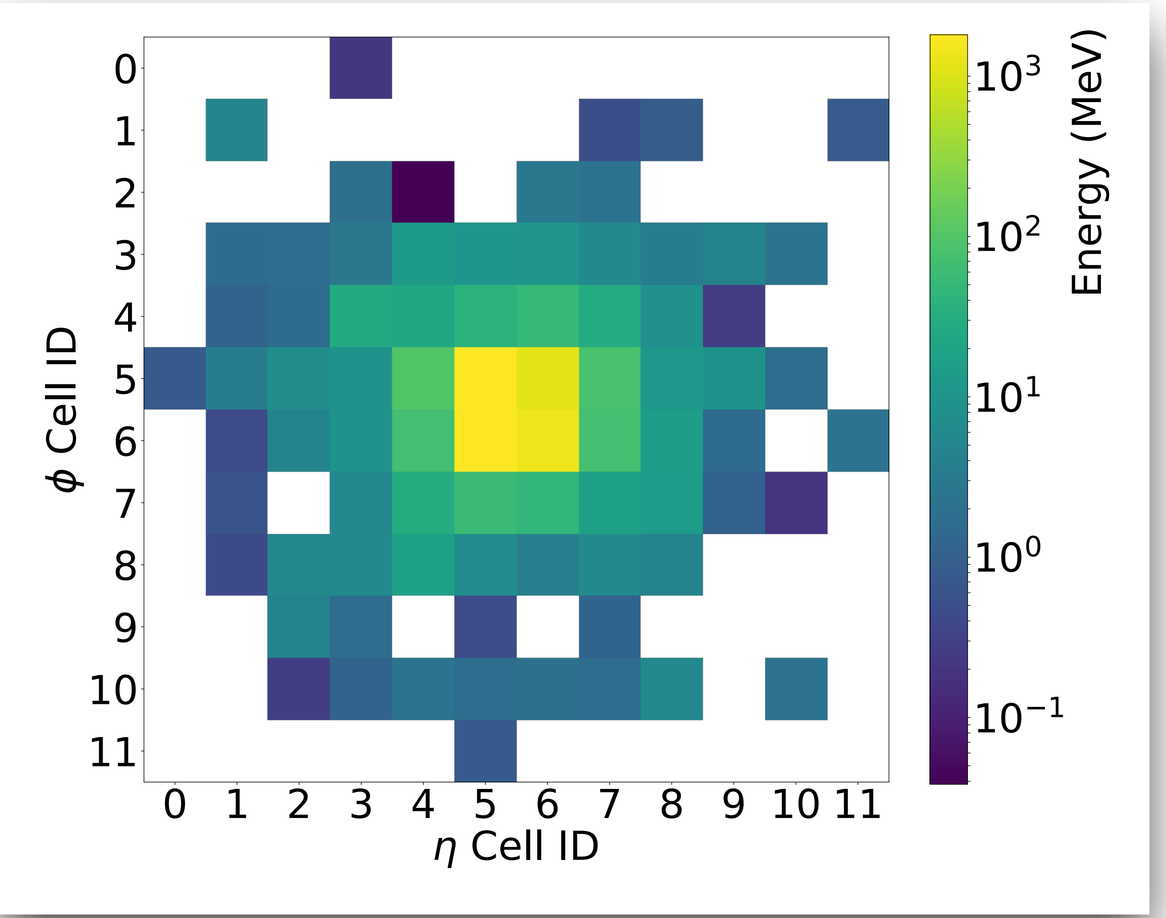
The Structure of Radiation in the Quantum Strong Force



Infer Parton/particle-level Dynamics

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

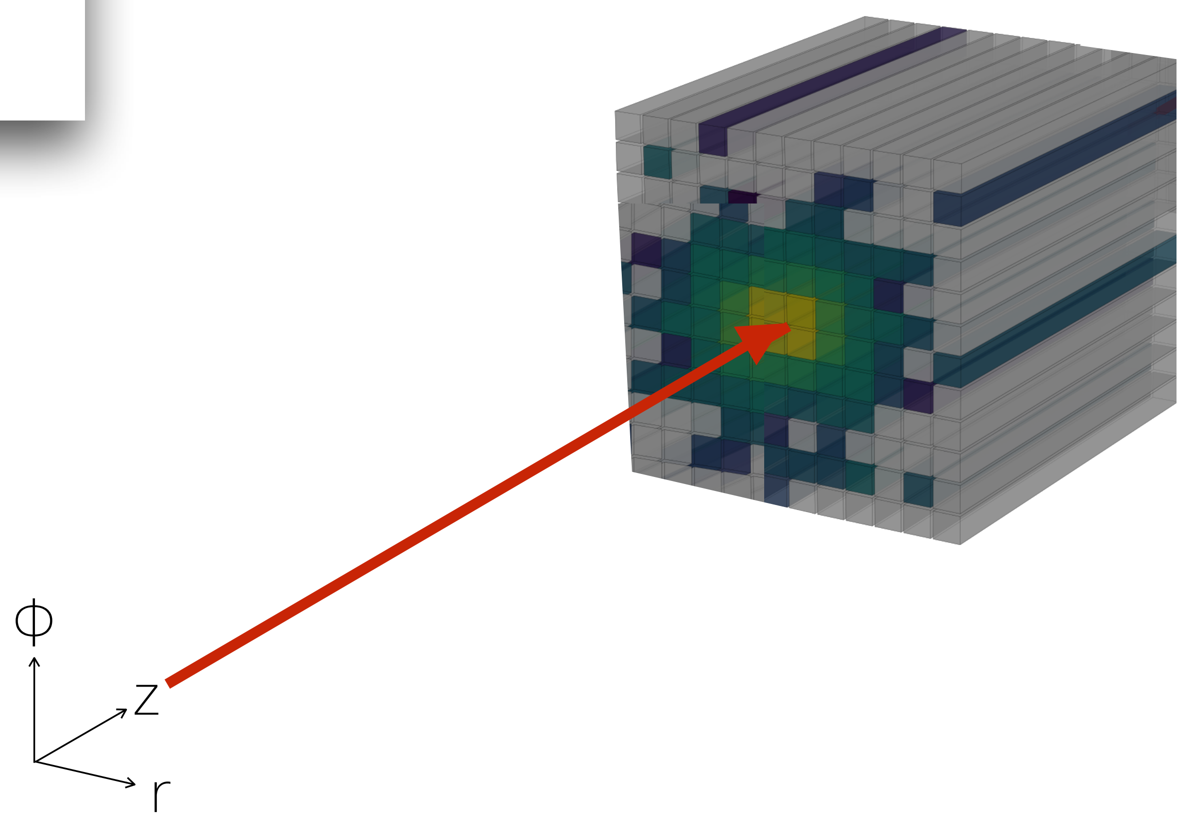
Accelerating Detector Simulations



Calorimeters are often the slowest to simulate

stopping particles requires simulating interactions of all energies

Grayscale images: Pixel intensity = energy deposited

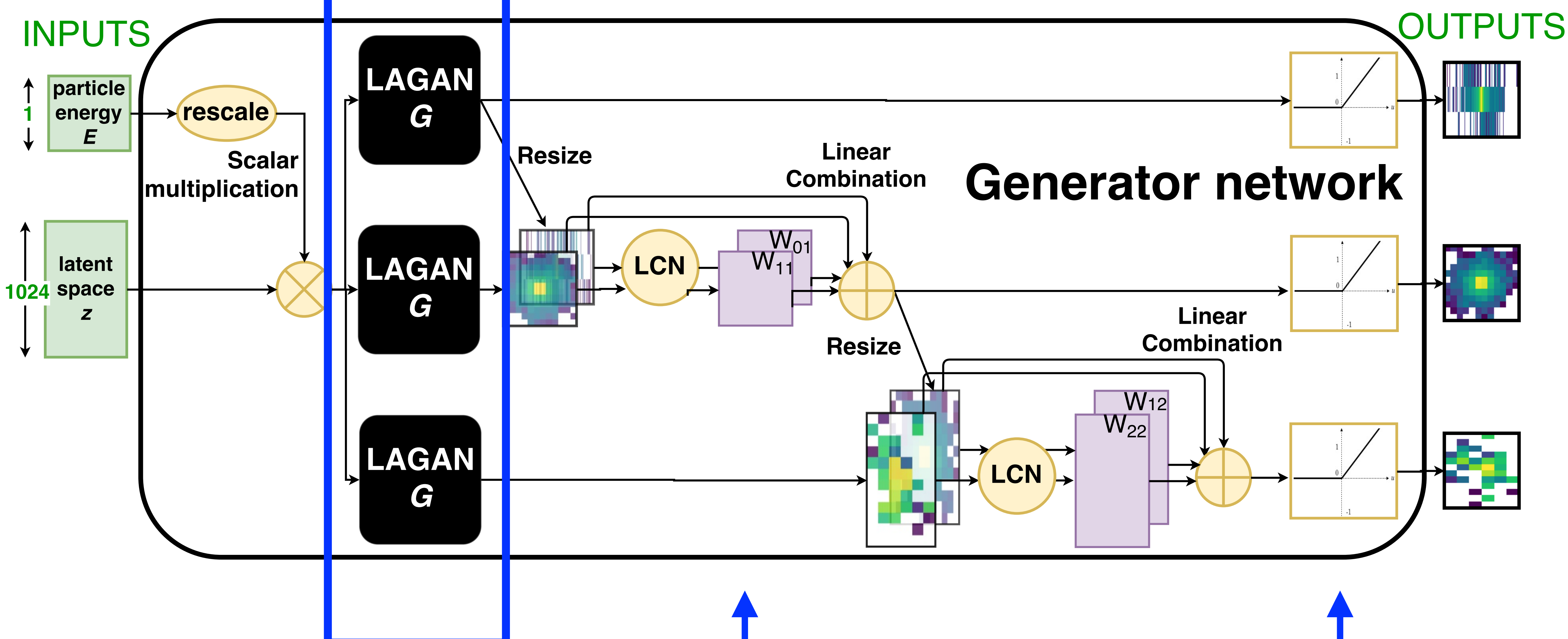


M. Paganini, L. de Oliveira, B. Nachman, PRL 120 (2018) 042003, 1705.02355

Introducing CaloGAN

One image per calo layer

One network per particle type; input particle energy



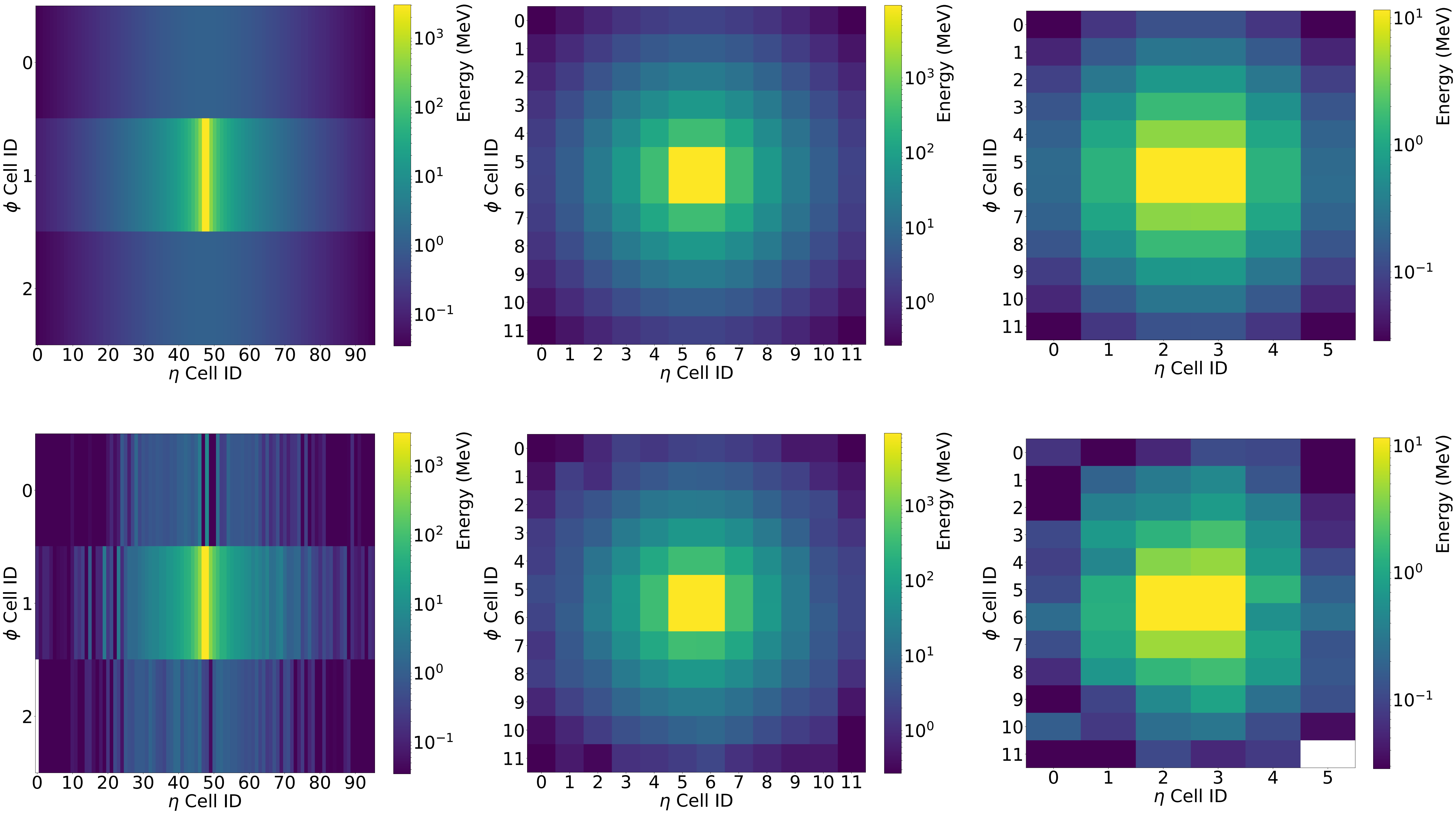
LA = Locally Aware, like a CNN

use layer i as input to layer i+1

ReLU to encourage sparsity

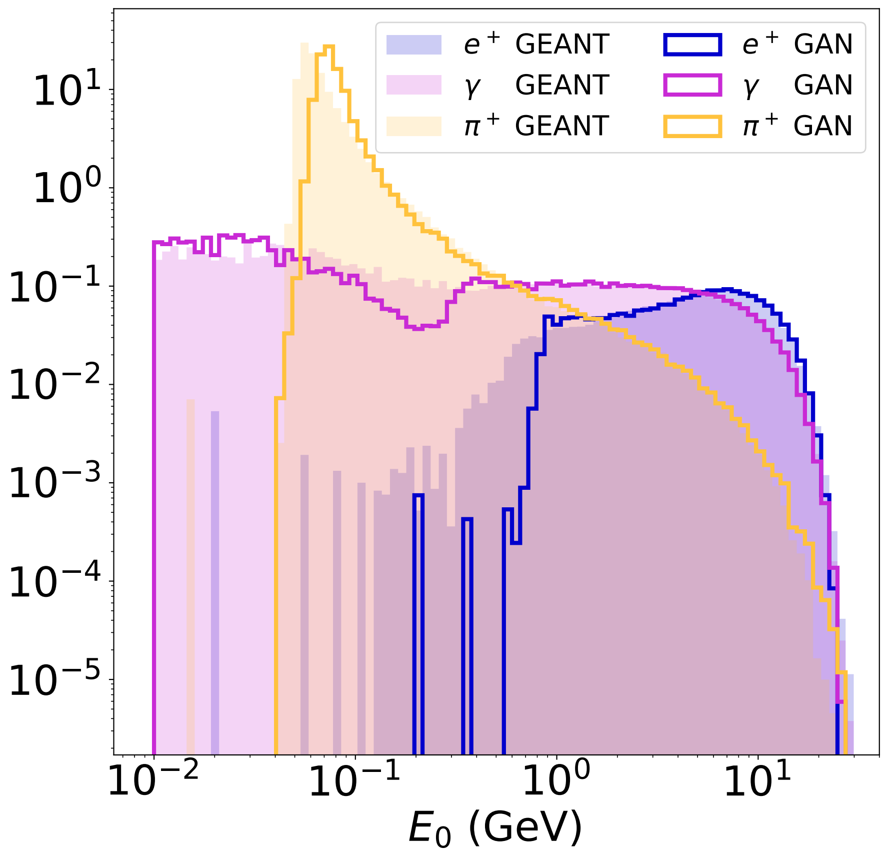
Performance: average images

Geant4

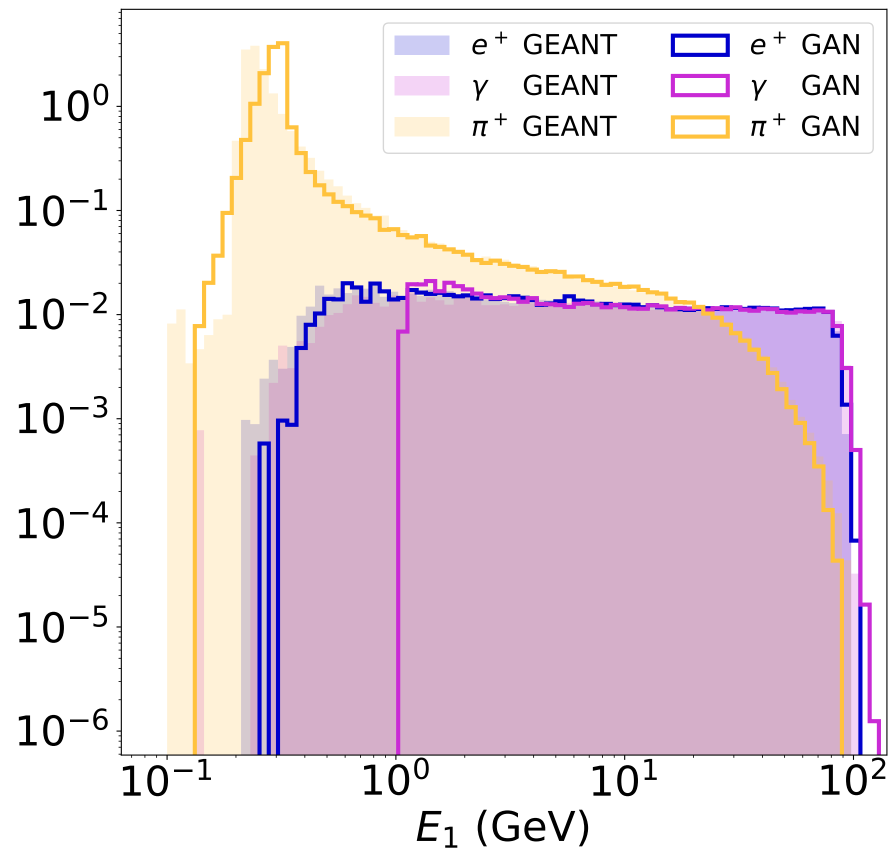


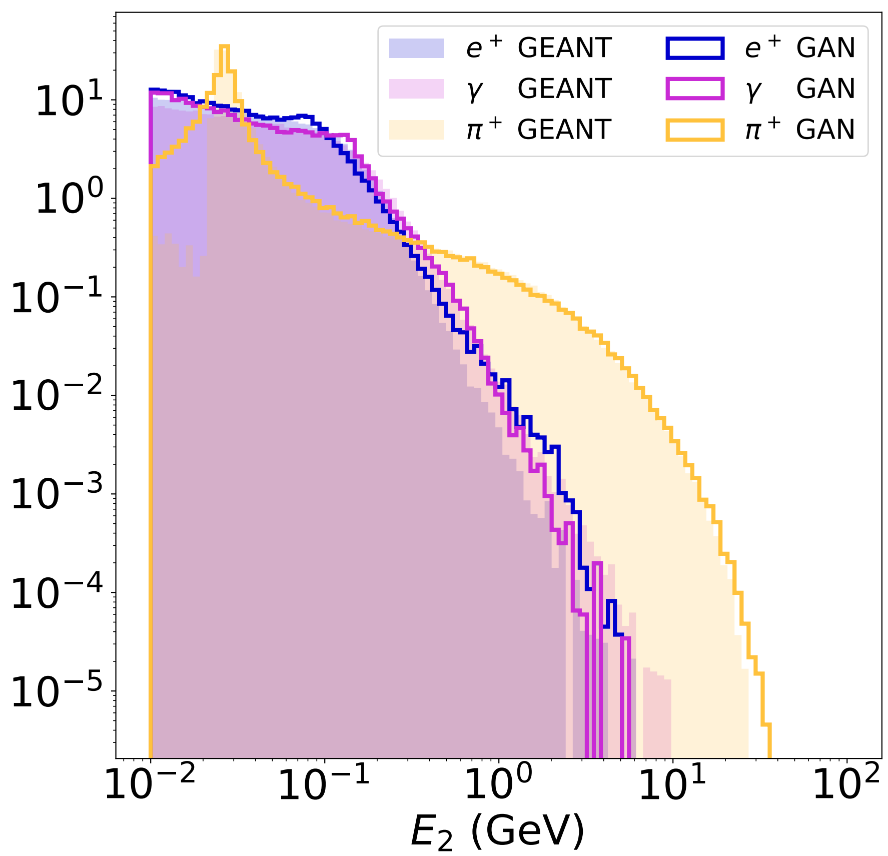
CaloGAN

Performance: energy per layer



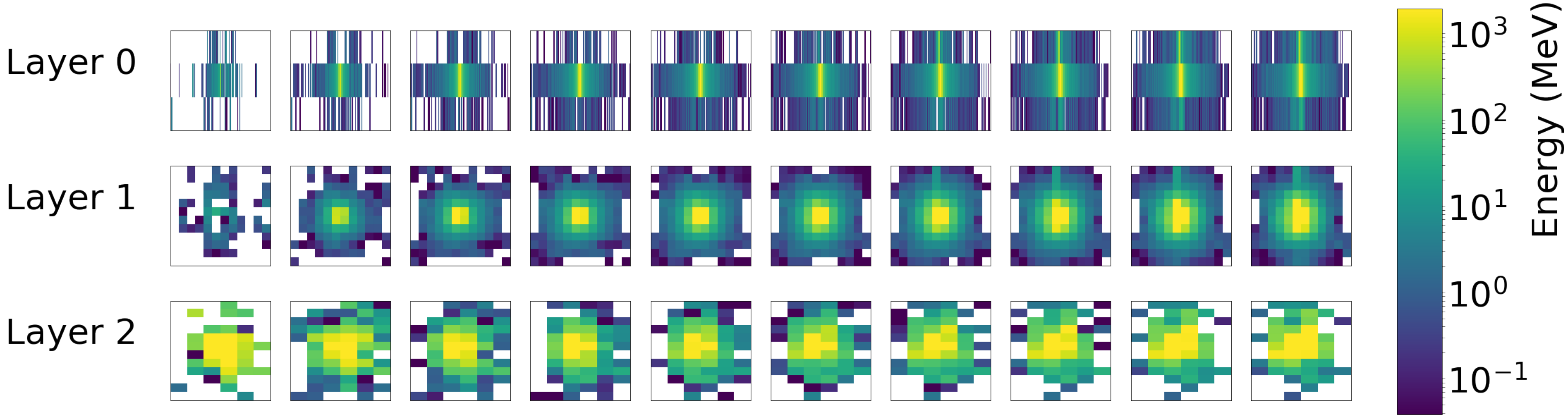
Pions deposit much less energy in the first layers; leave the calorimeter with significant energy



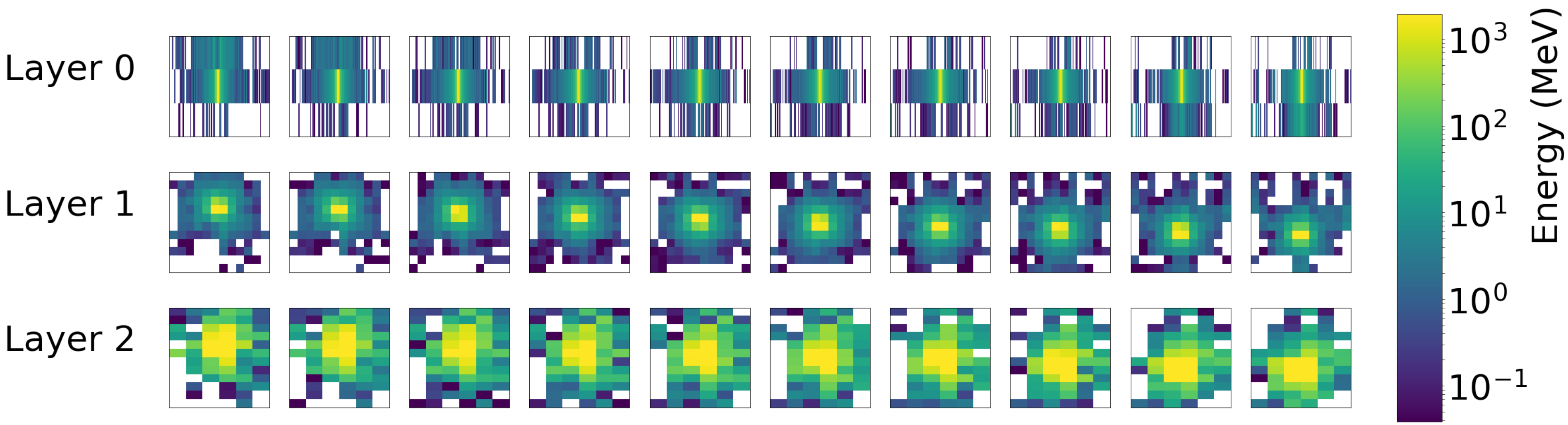


Conditioning

Fix noise, scan latent variable corresponding to energy



Fix noise, scan latent variable corresponding to x-position



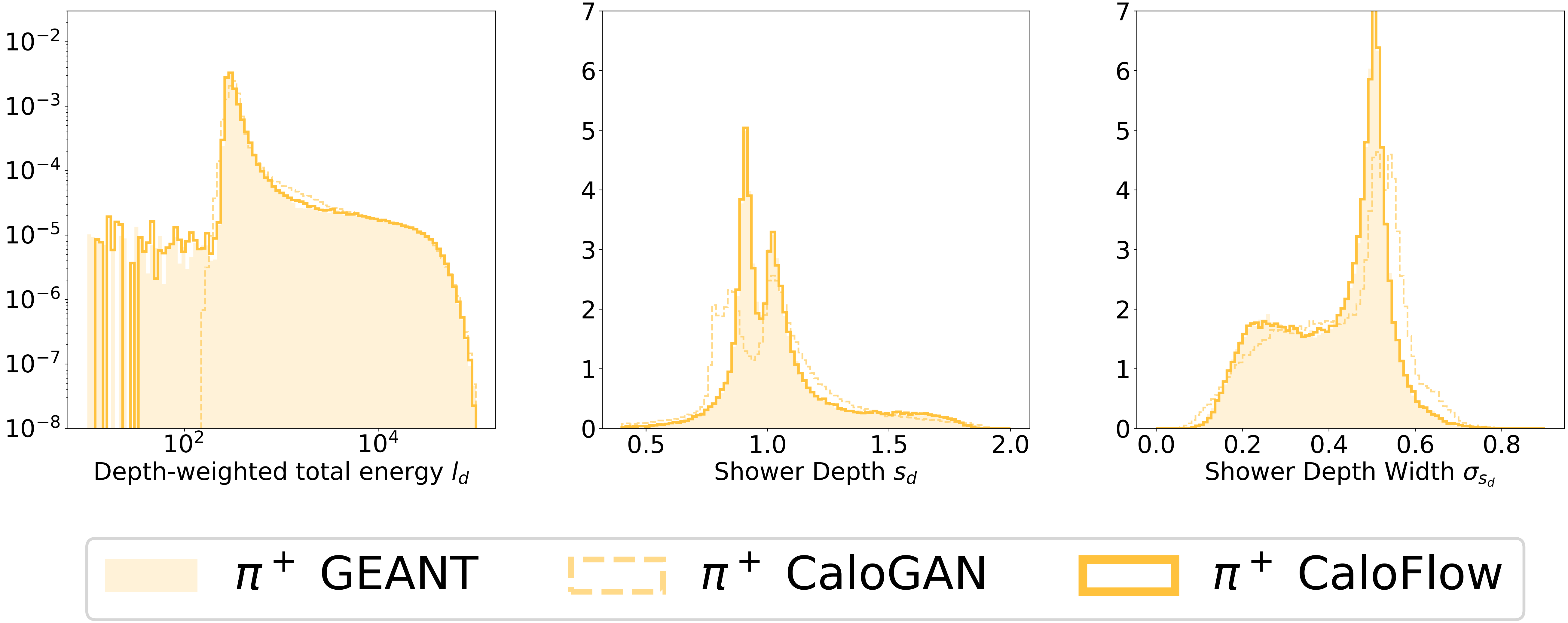
Timing

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.



many other papers - see Living Review, 2102.02770

C. Krause and D. Shih, 2106.05285

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

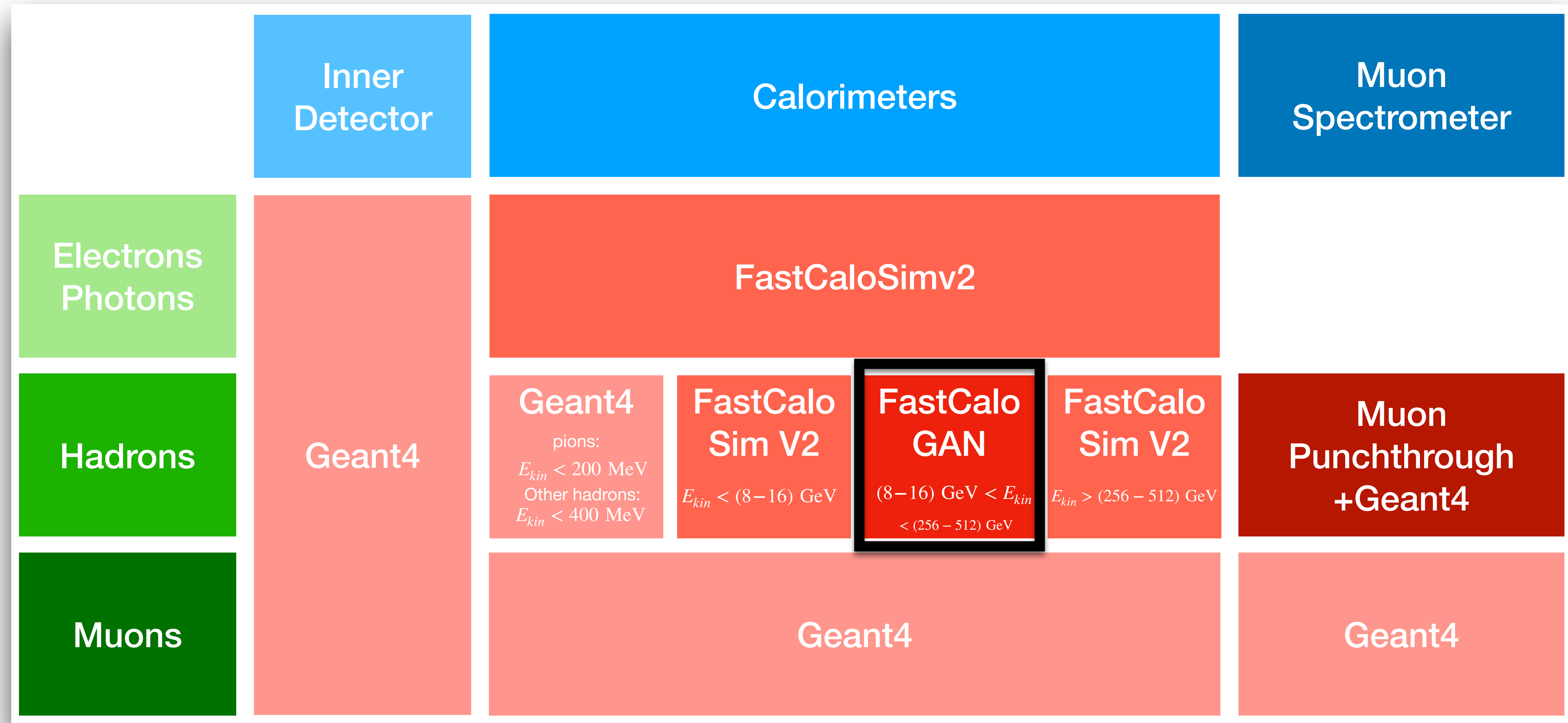
Depth-weighted total energy l_d Shower Depth s_d Shower Depth Width σ_{s_d}

π^+ GEANT π^+ CaloGAN π^+ CaloFlow

many other papers - see Living Review, 2102.02770

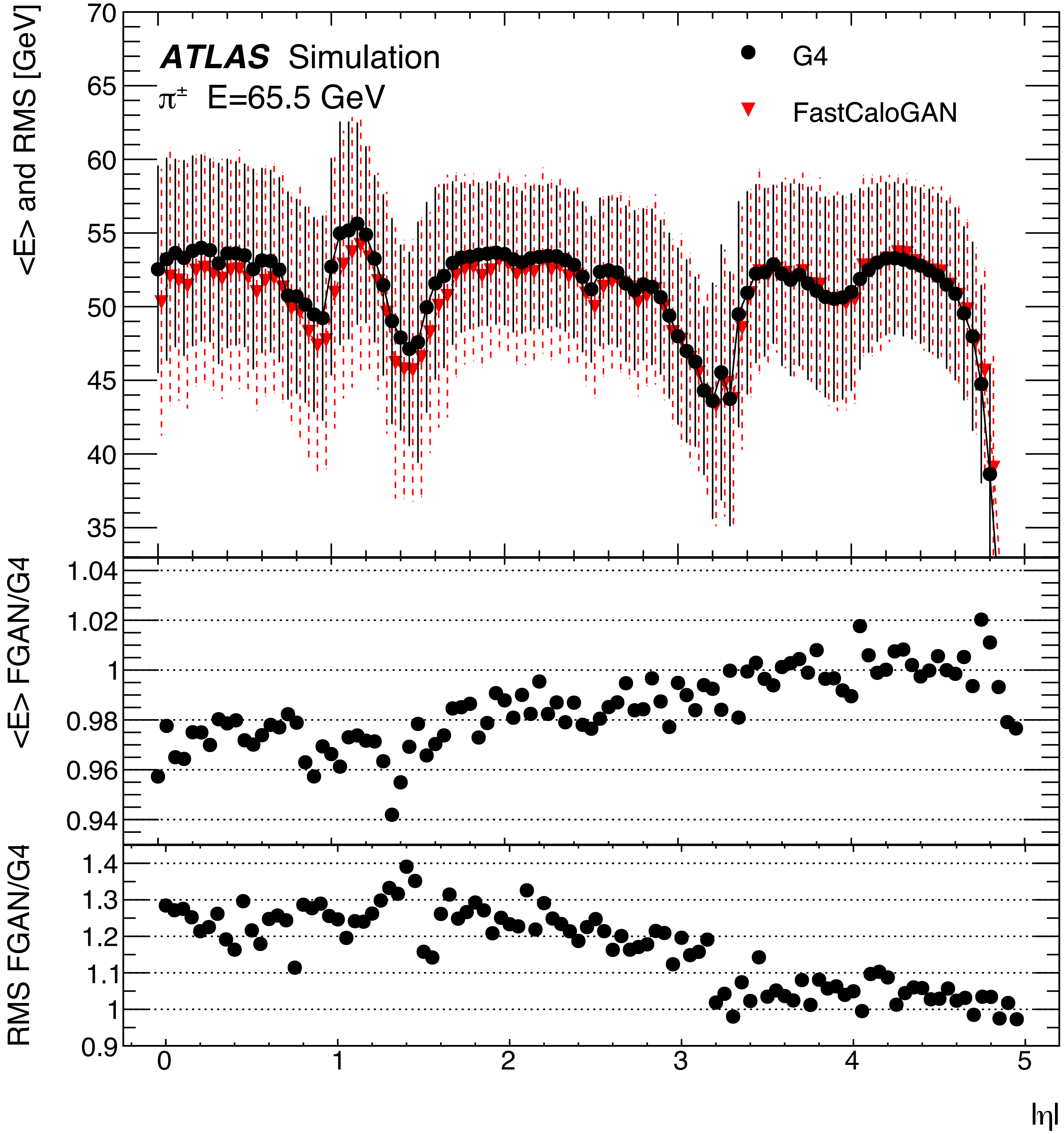
ATLAS Collaboration, 2109.02551

Integration into real detector sim.



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

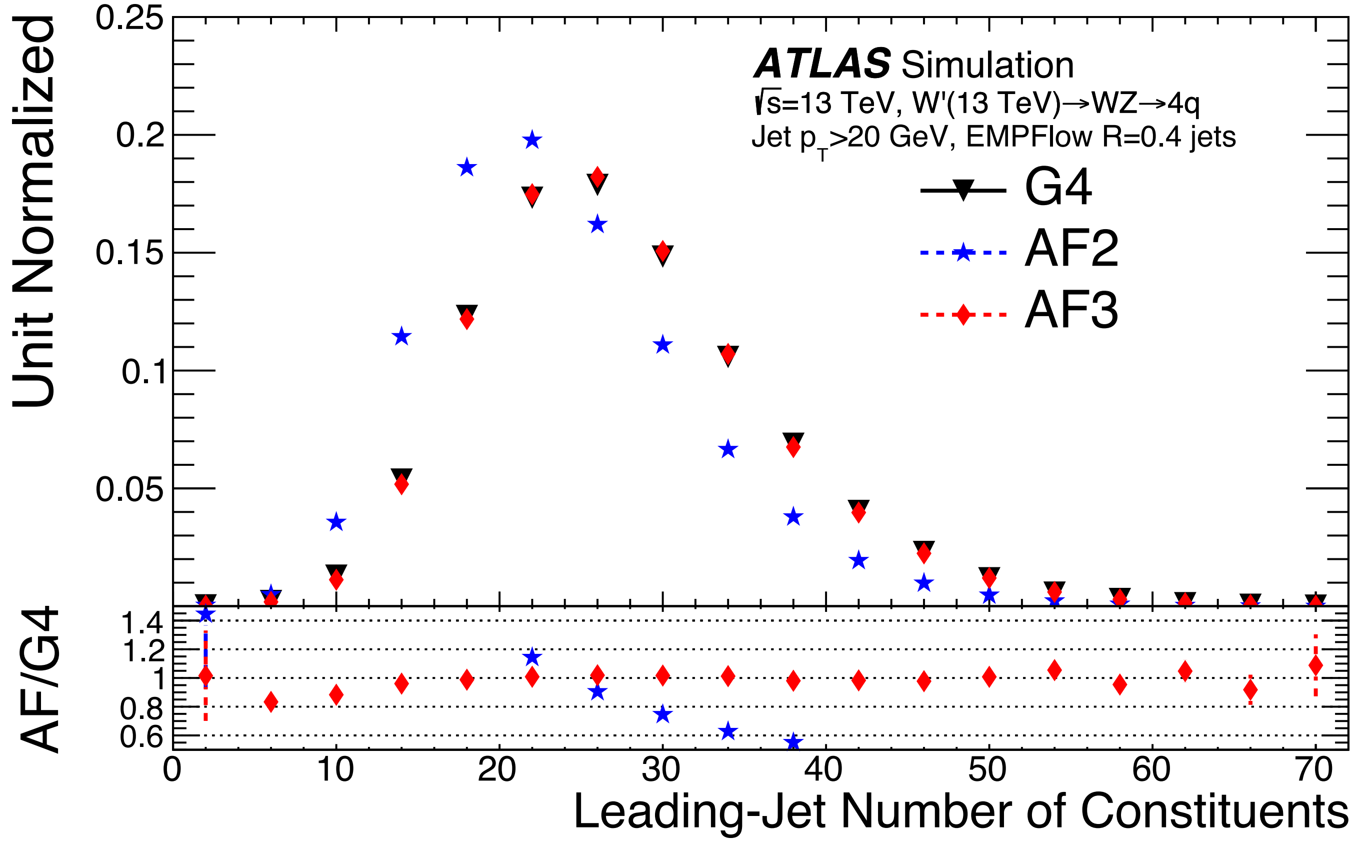
Integration into real detector sim.



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

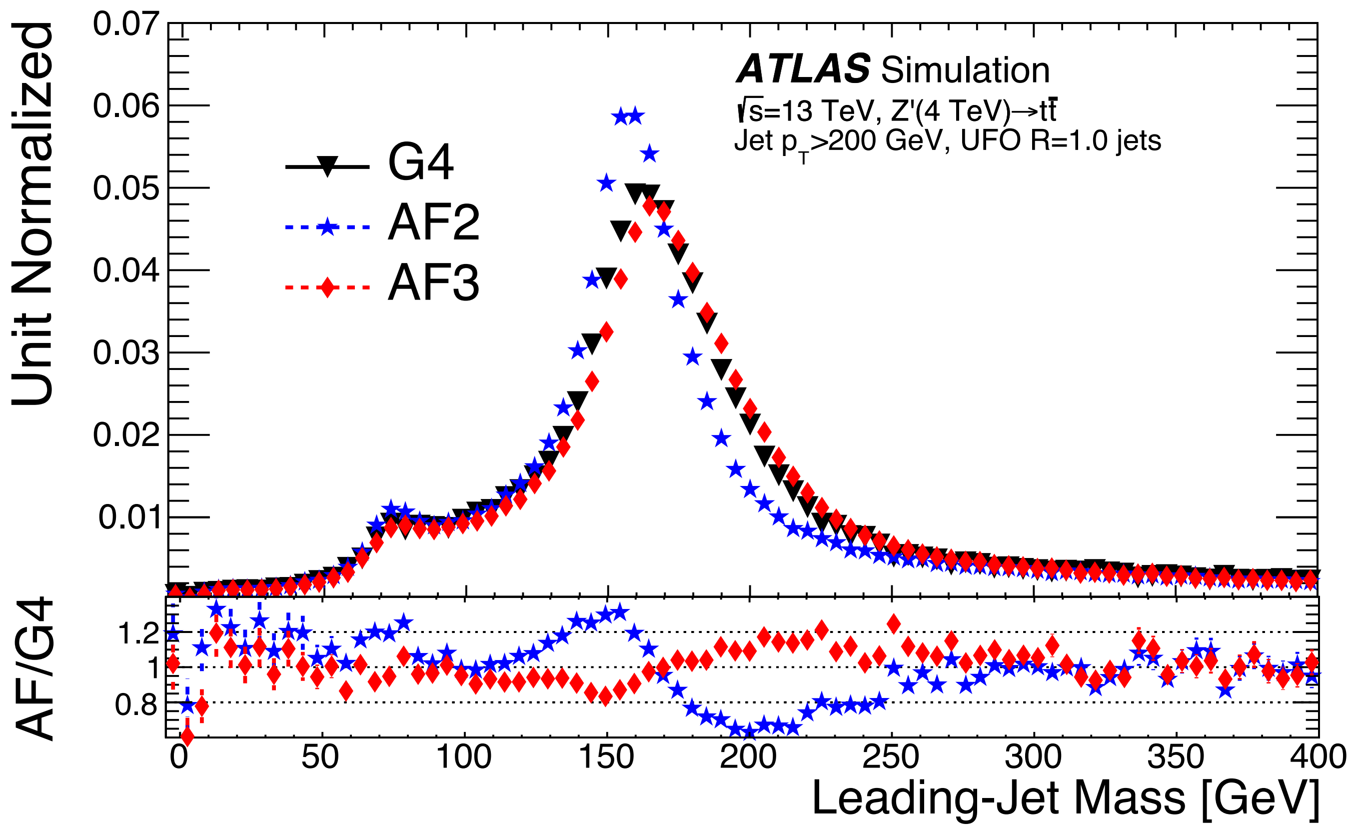
There is one GAN per η slice

Integration into real detector sim.

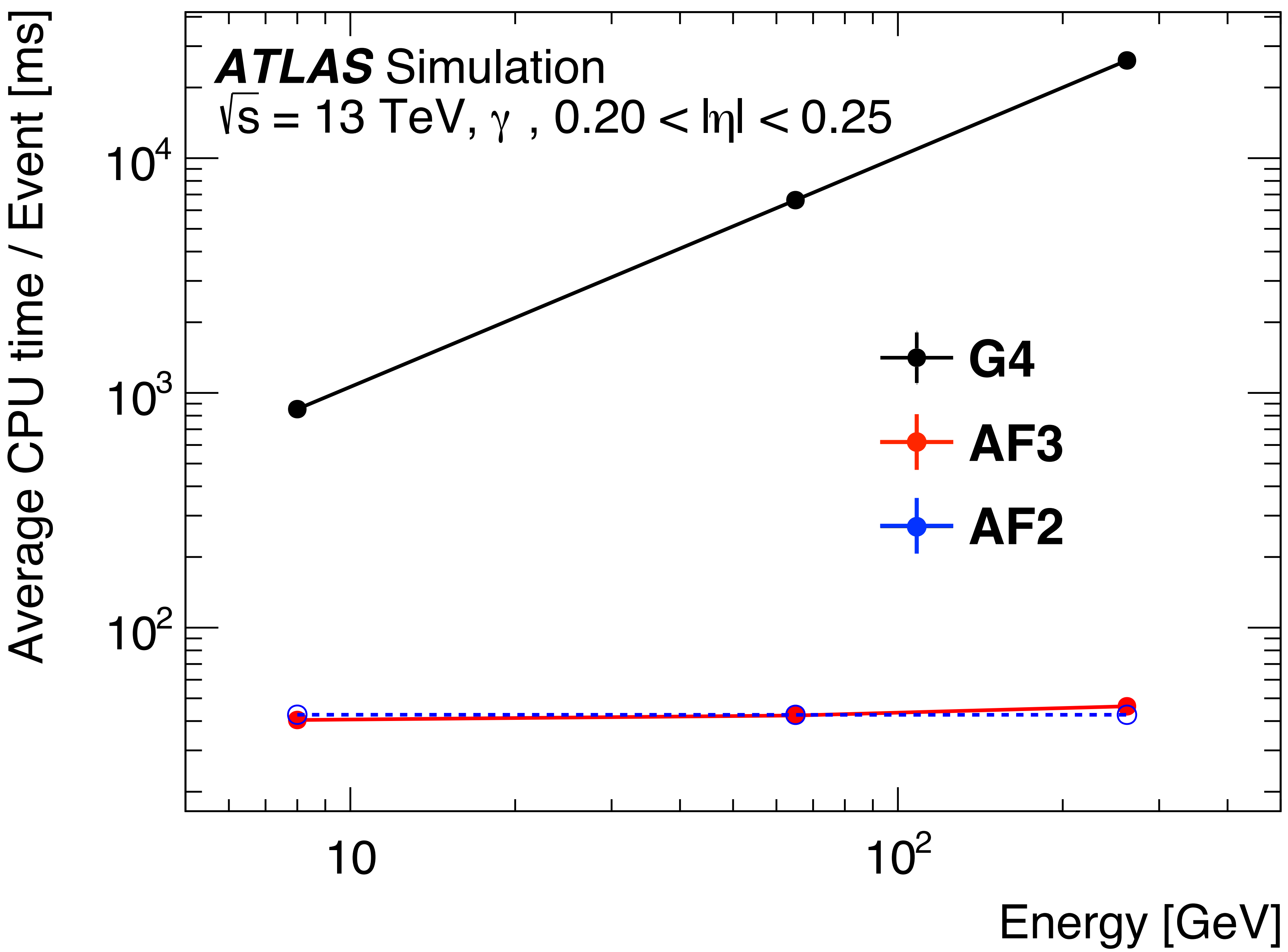


The new fast simulation (**AF3**) significantly improves jet substructure with respect to the older one (**AF2**)

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



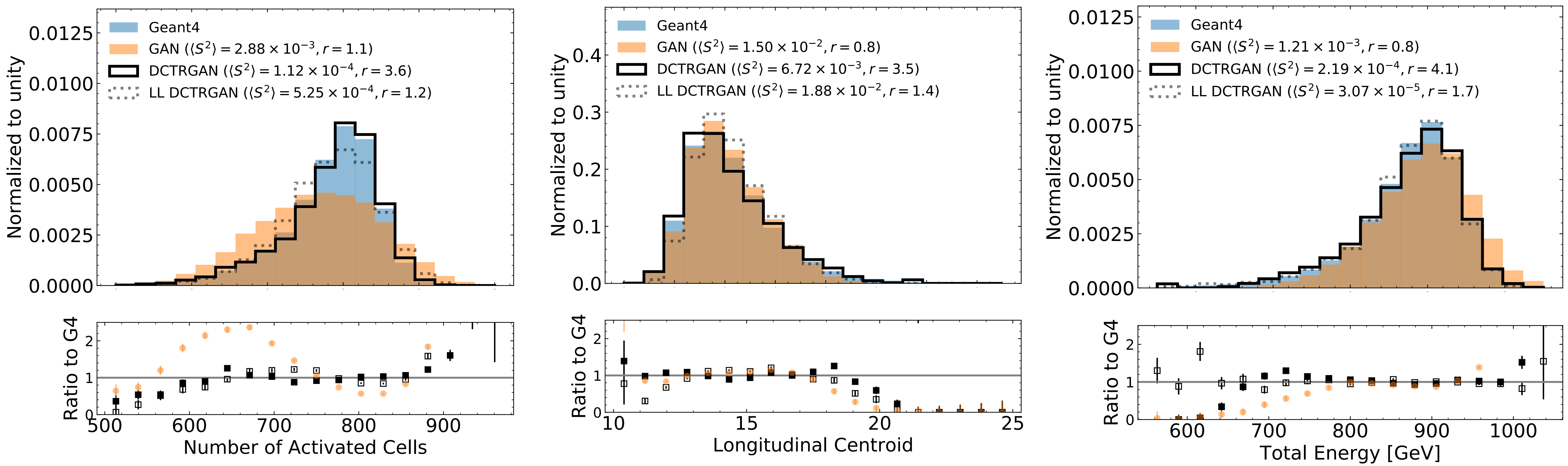
Integration into real detector sim.



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

Refining Simulations

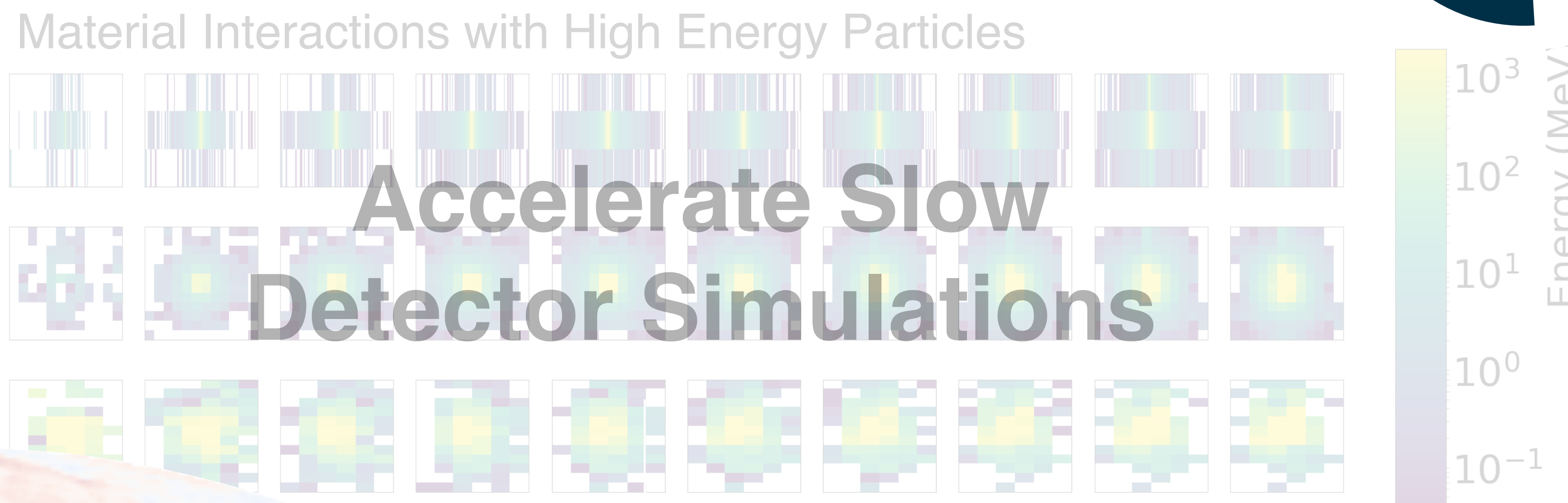
As we move towards precision, we may need to complement primary generative models with post-hoc correction models (e.g. via reweighting)



See also 2106.00792 (“LaSeR”) and 2107.08648 (optimal transport-based)

Generative Models for Particle/Nuclear/Astro

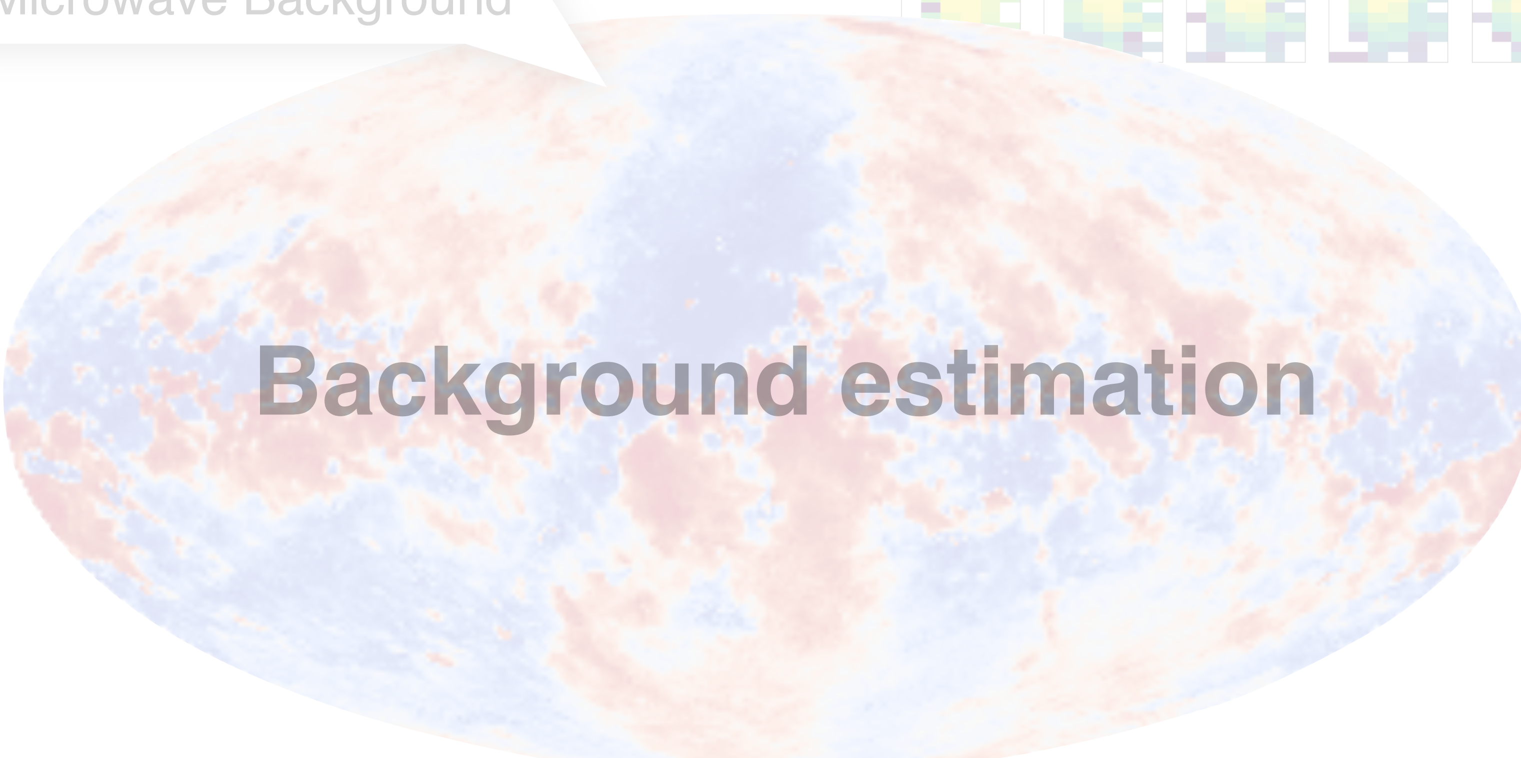
All of these pictures are fake!



Accelerate Slow Detector Simulations

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

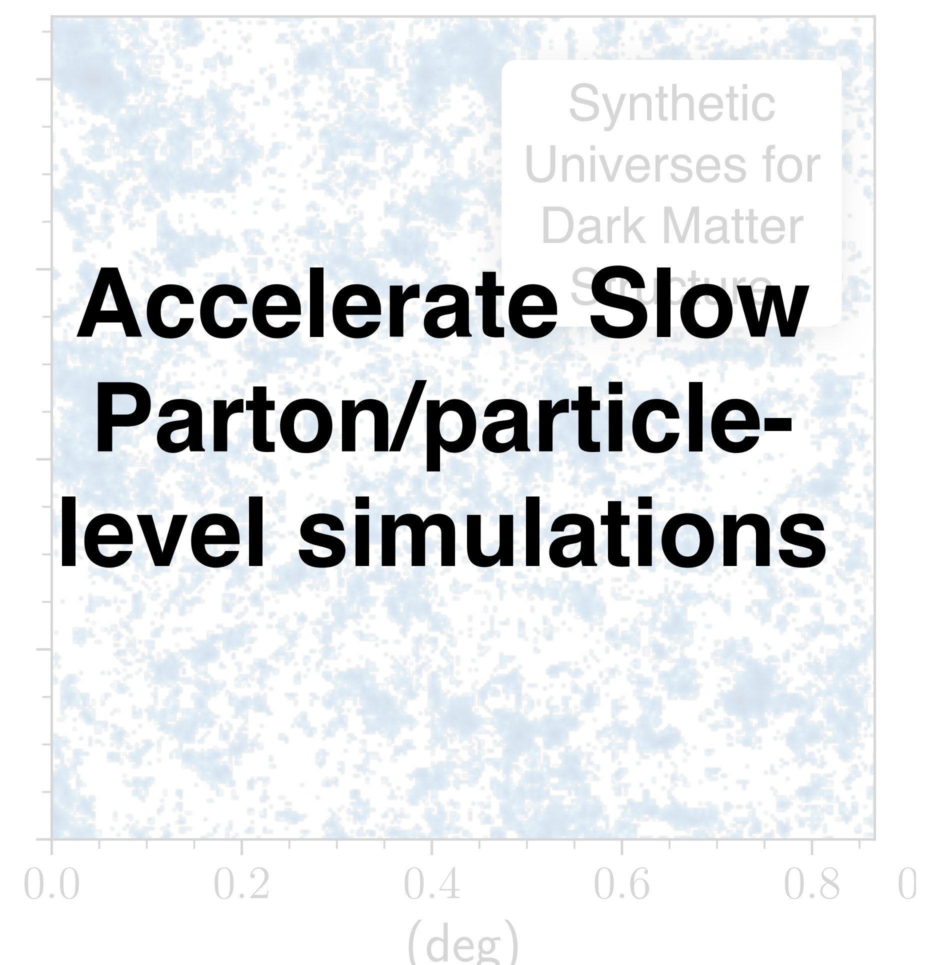
Synthetic Galactic radiation for Cosmic Microwave Background



Background estimation

N. Krachmalnicoff and G. Puglisi, arXiv:2011.02221

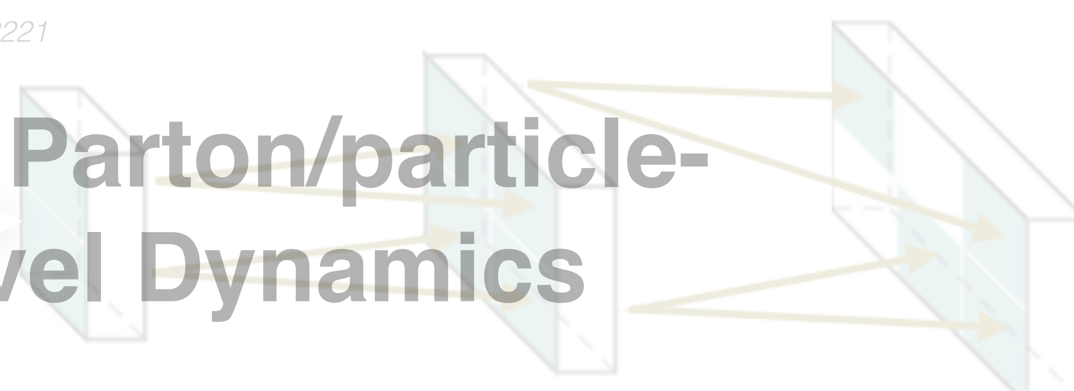
Synthetic Universes for Dark Matter Structure



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 Parton/Particle Sim.*

Flat jet images with GANs

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

1701.05927

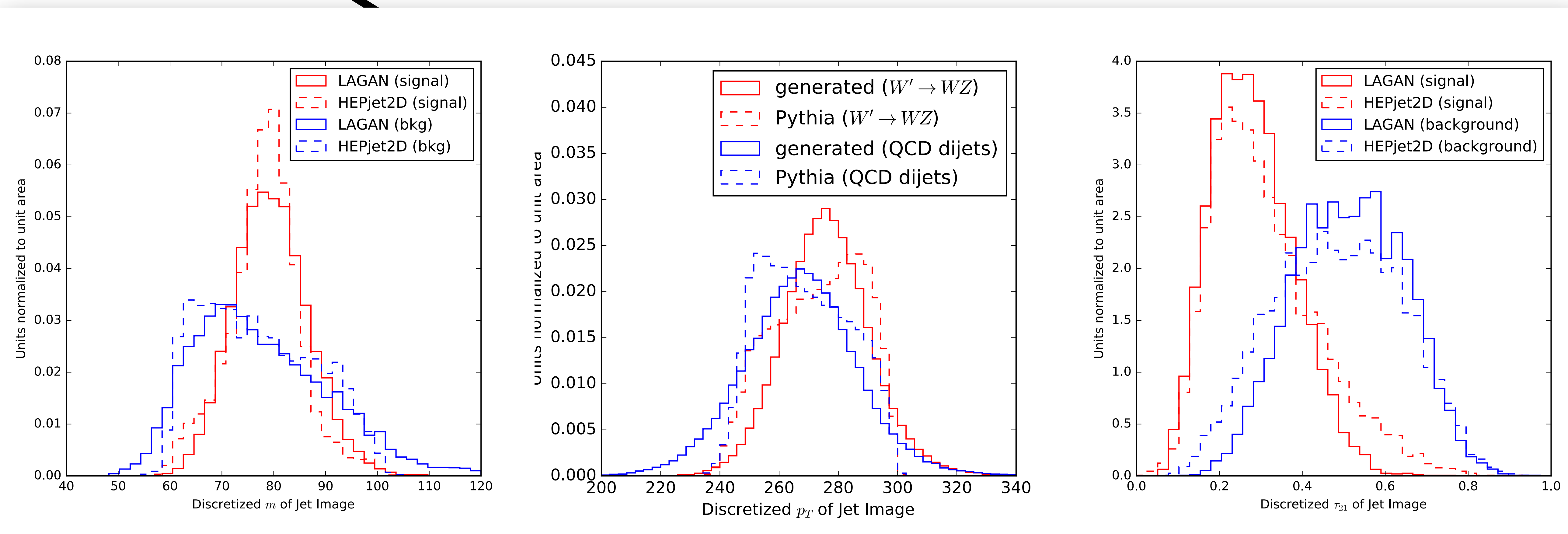
*these are just representative examples - see Living Review, 2102.02770

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

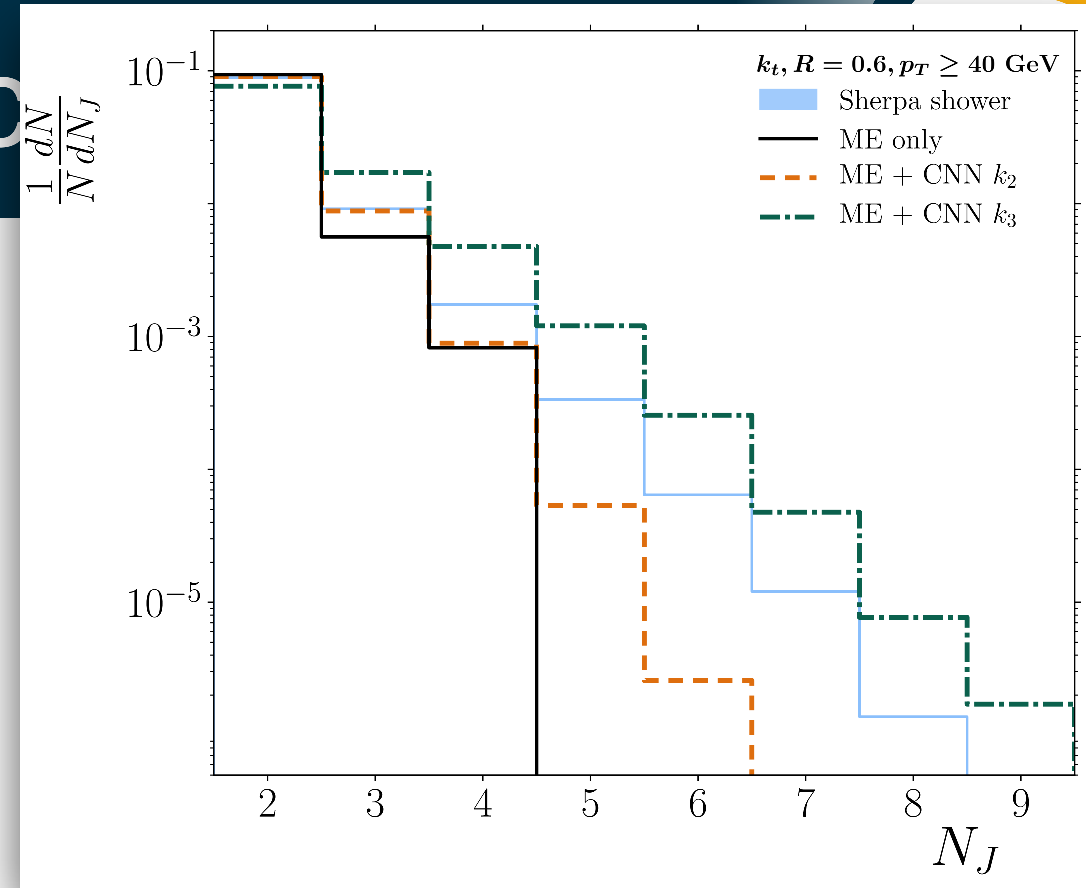
Weight sharing across space

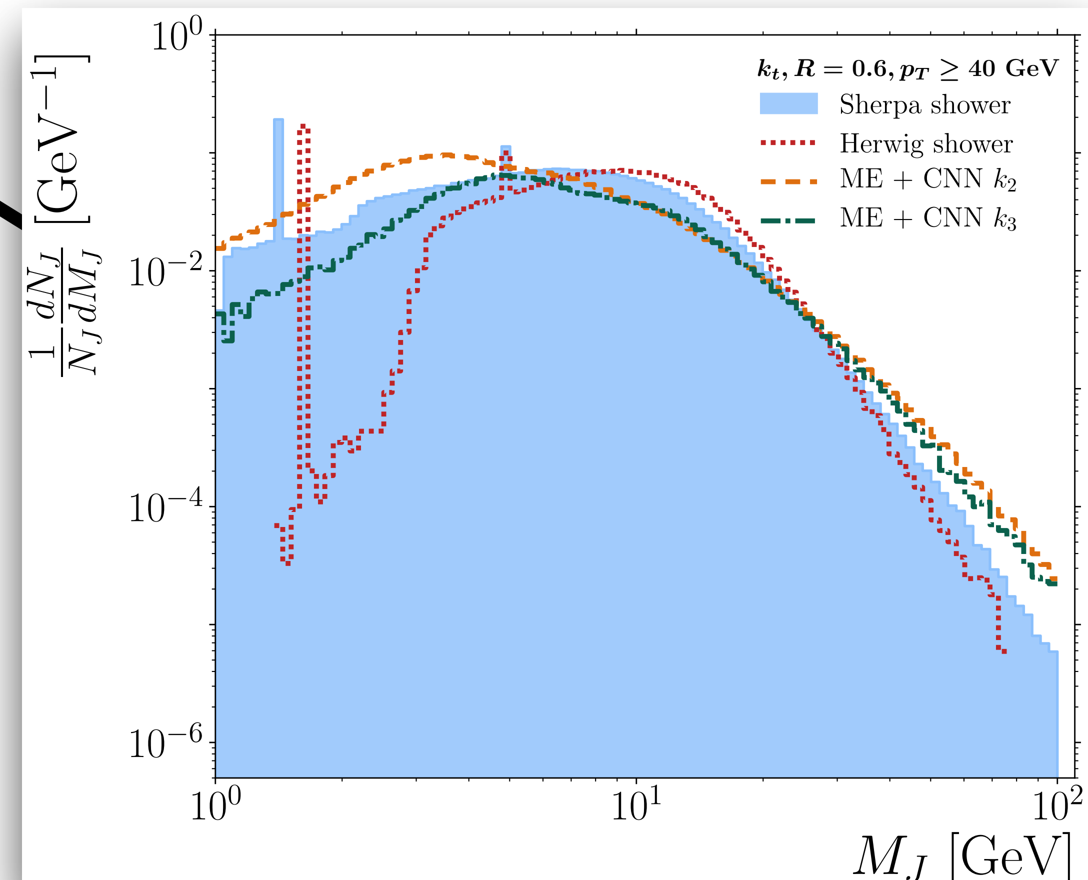
*these are just representative examples - see Living Review, 2102.02770

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





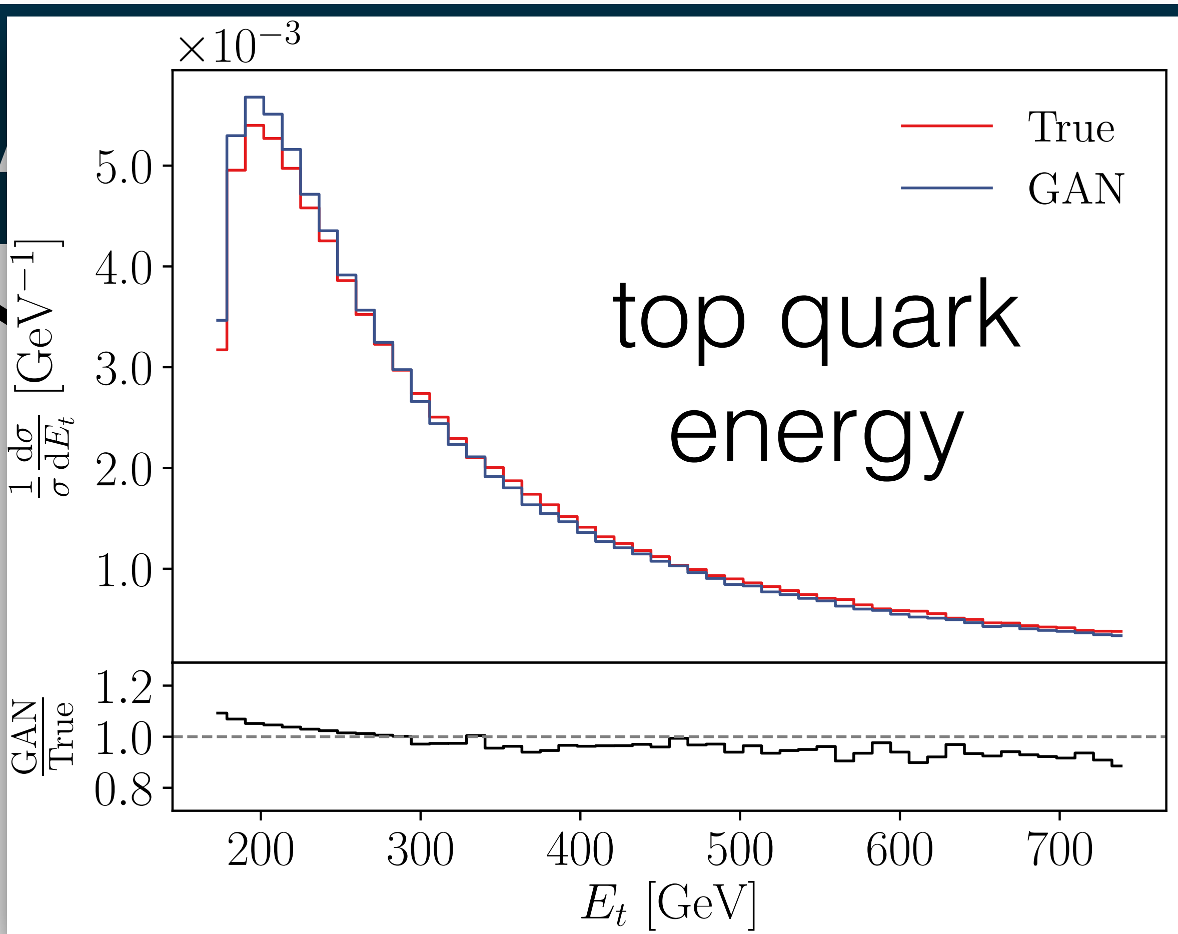
Weight sharing across space + "time"

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

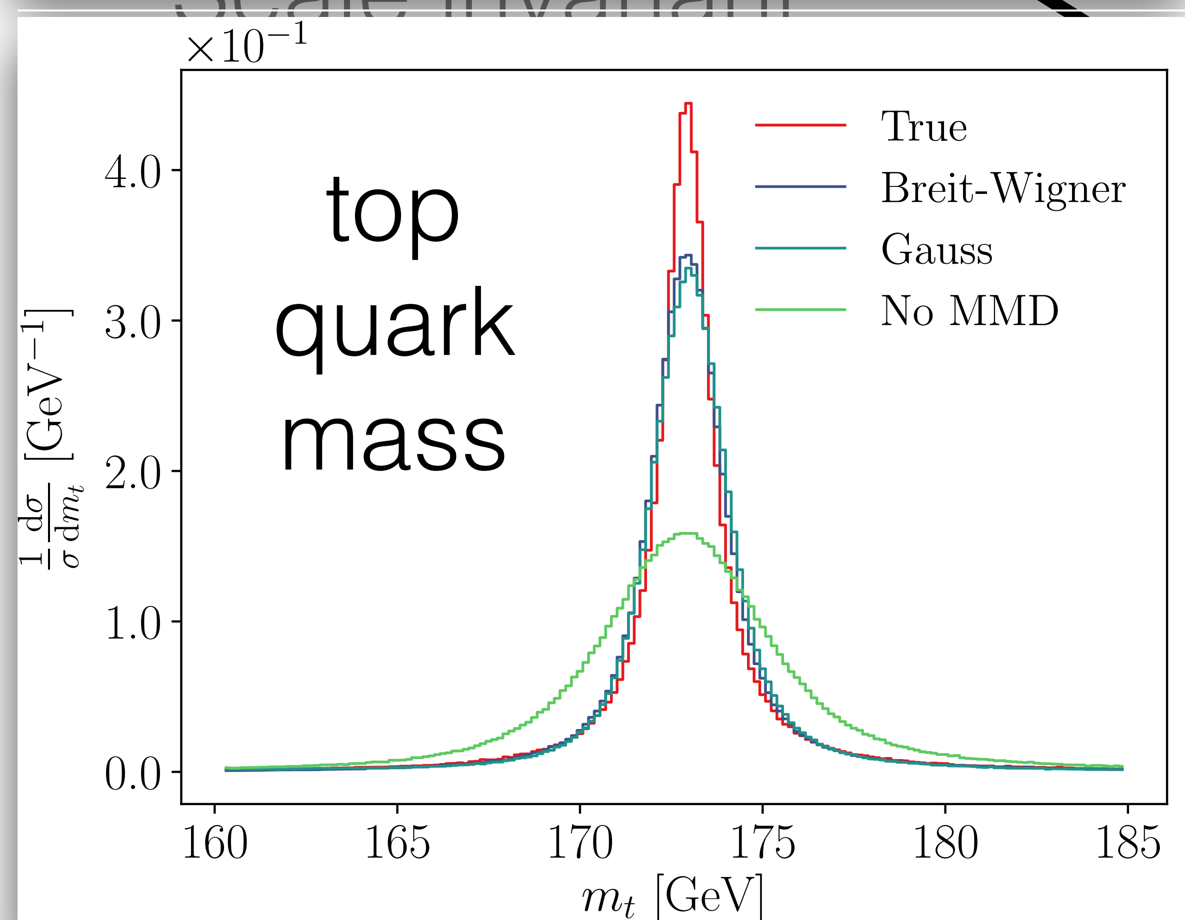
Particle Sim.*

ANs

nman



Scale invariant



MMD = maximum mean discrepancy

Fixed number of 4-vectors, allow for intermediate resonances

A. Butter, T. Plehn, R. Winterhalder

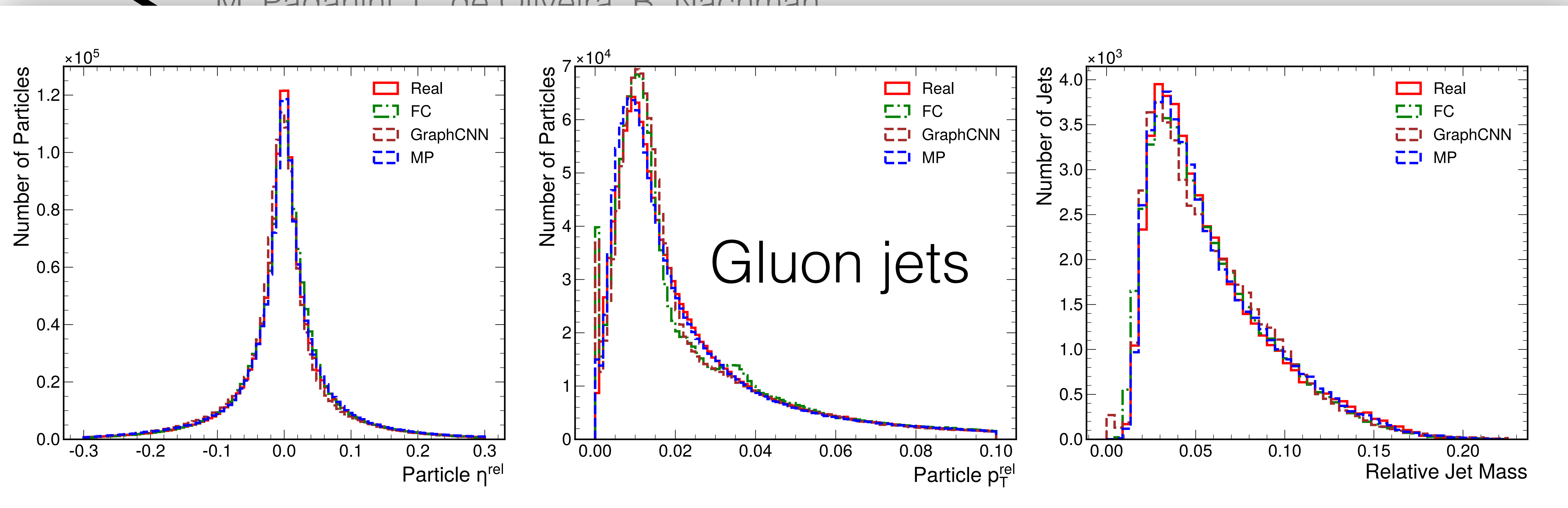
1907.03764

*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

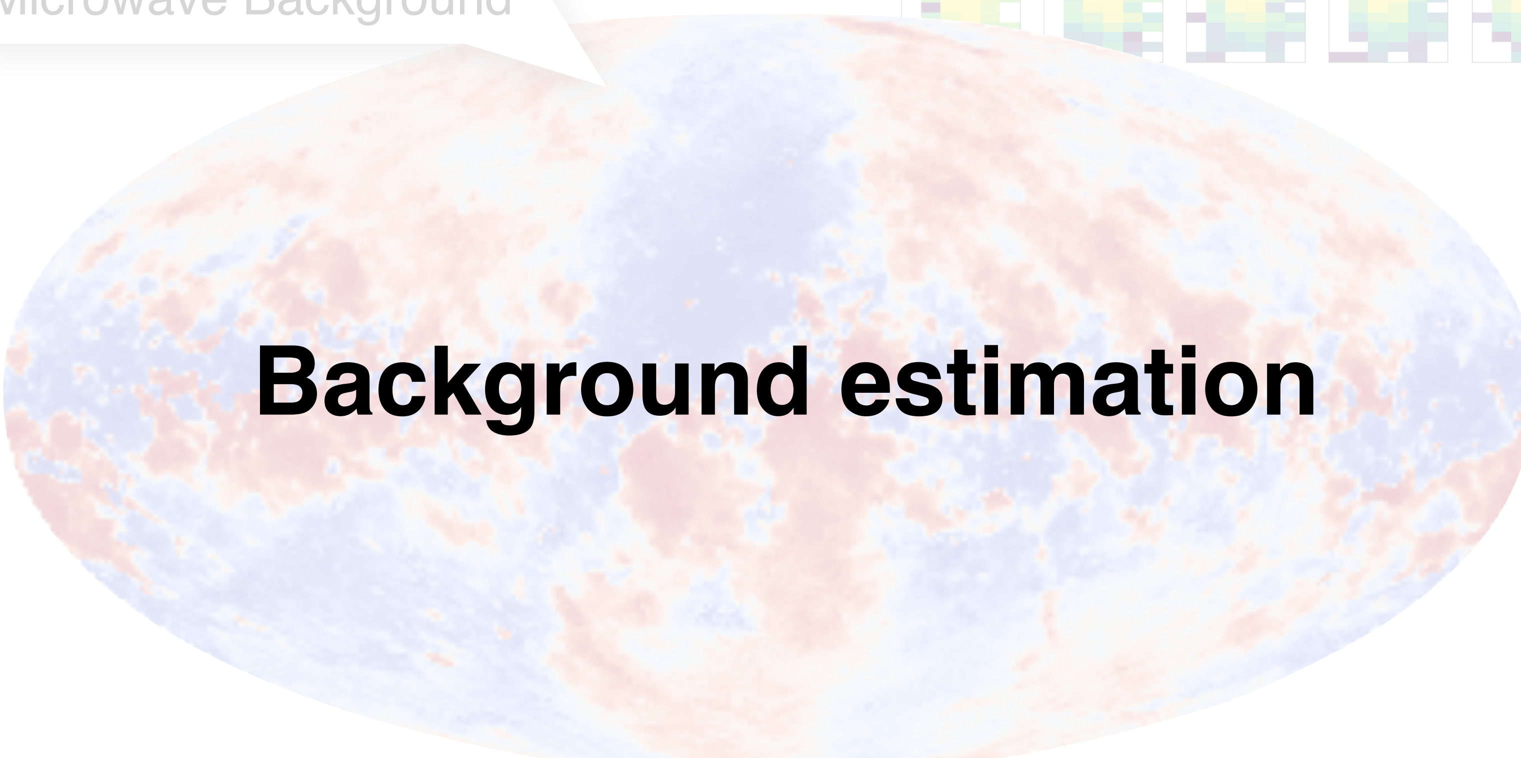
?

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

Generative Models for Particle/Nuclear/Astro

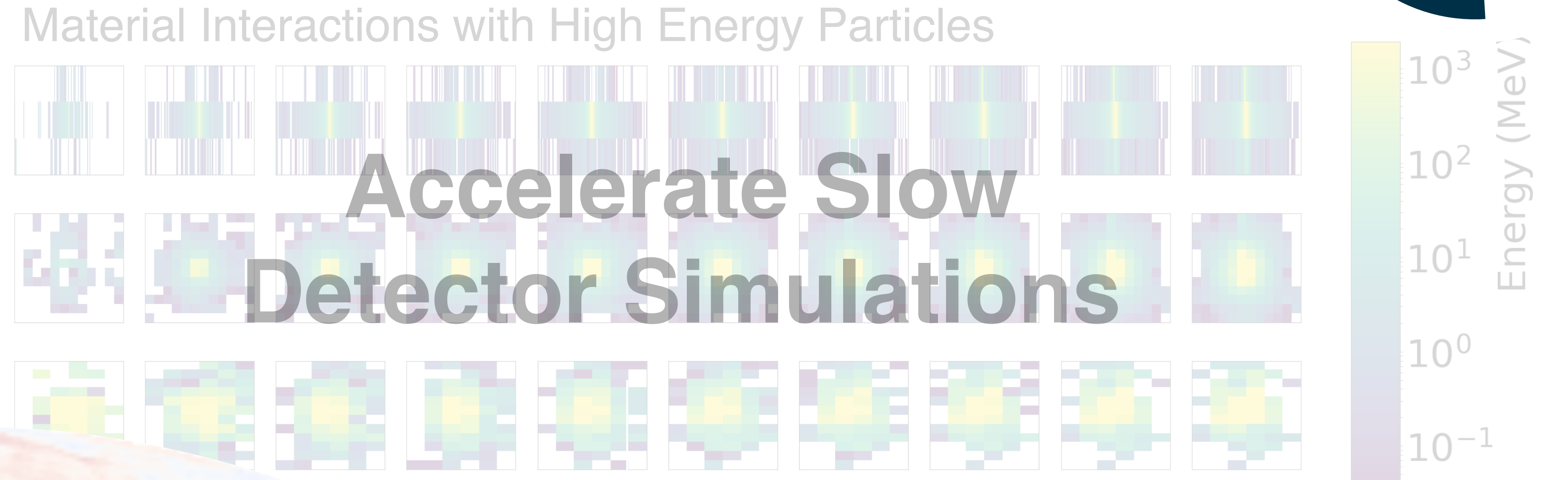
All of these pictures are fake!

Synthetic Galactic radiation for Cosmic Microwave Background



N. Krachmalnicoff and G. Puglisi, arXiv:2011.02221

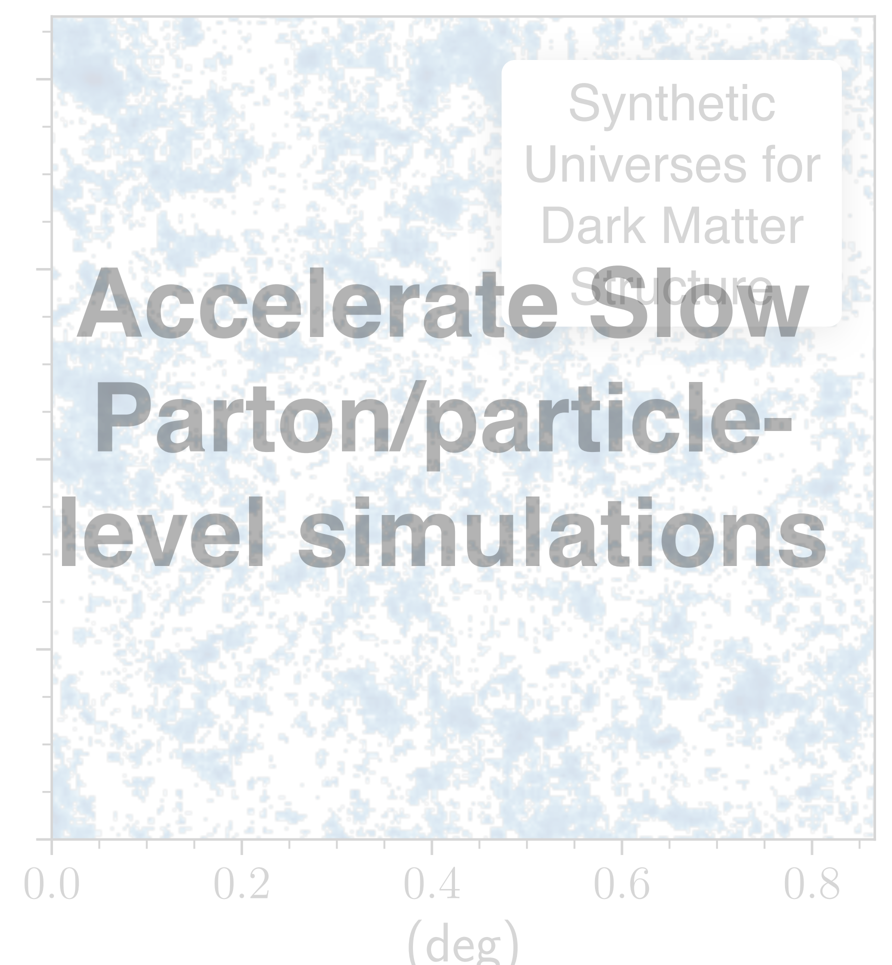
Background estimation



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

Accelerate Slow Detector Simulations

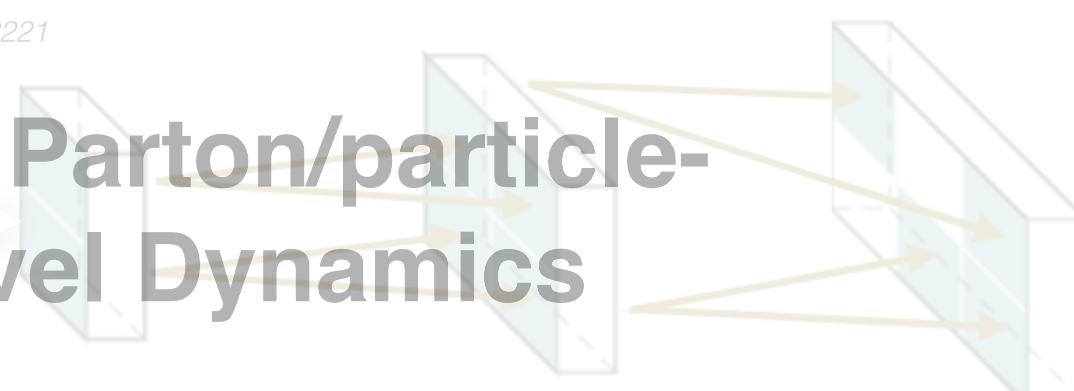
Synthetic Universes for Dark Matter Structure



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

Accelerate Slow Parton/particle-level simulations

The Structure of Radiation in the Quantum Strong Force



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

Infer Parton/particle-level Dynamics

Background Estimation

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

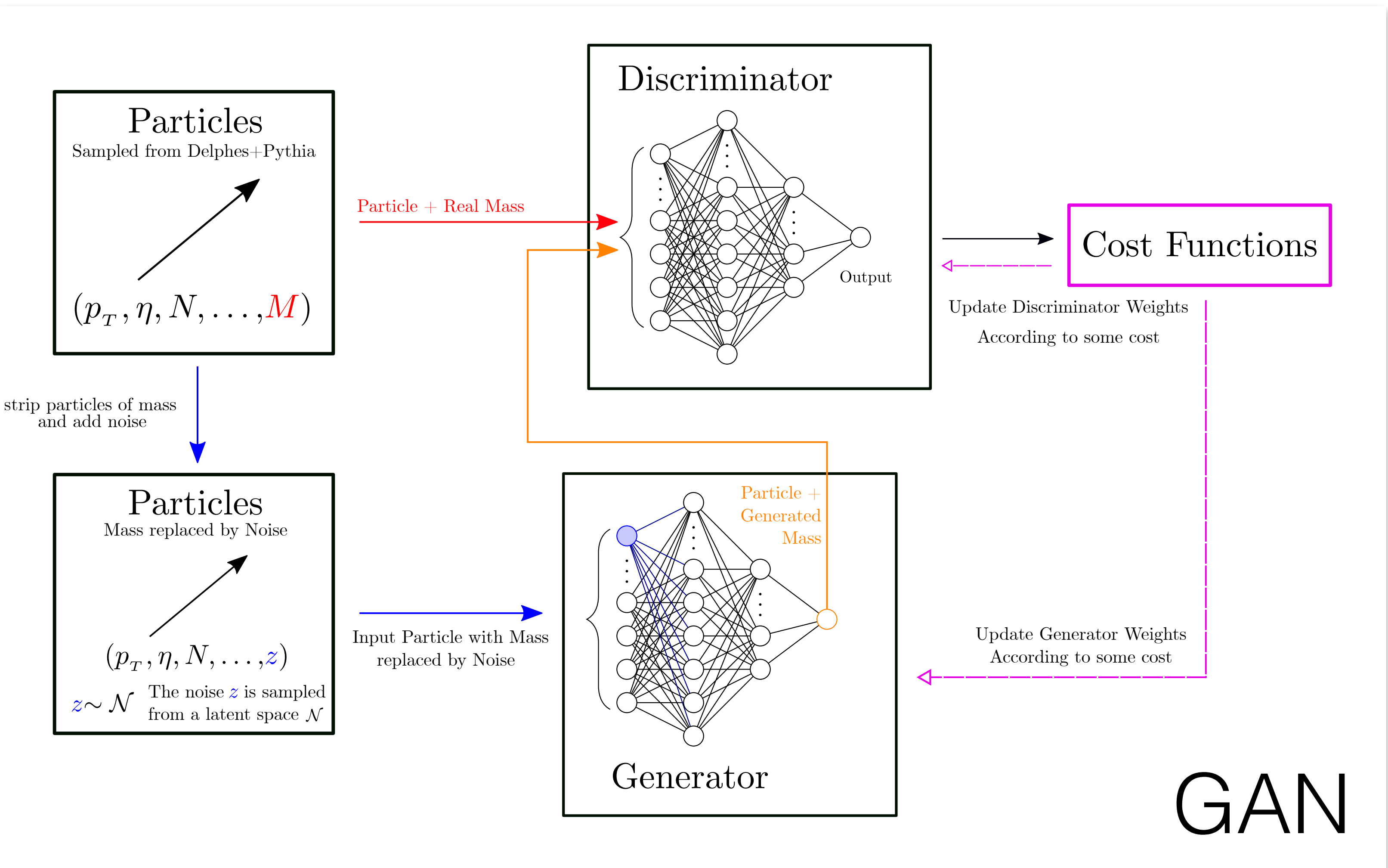
N.B. everything in I've shown before this, we trained on simulation, not on data (!)

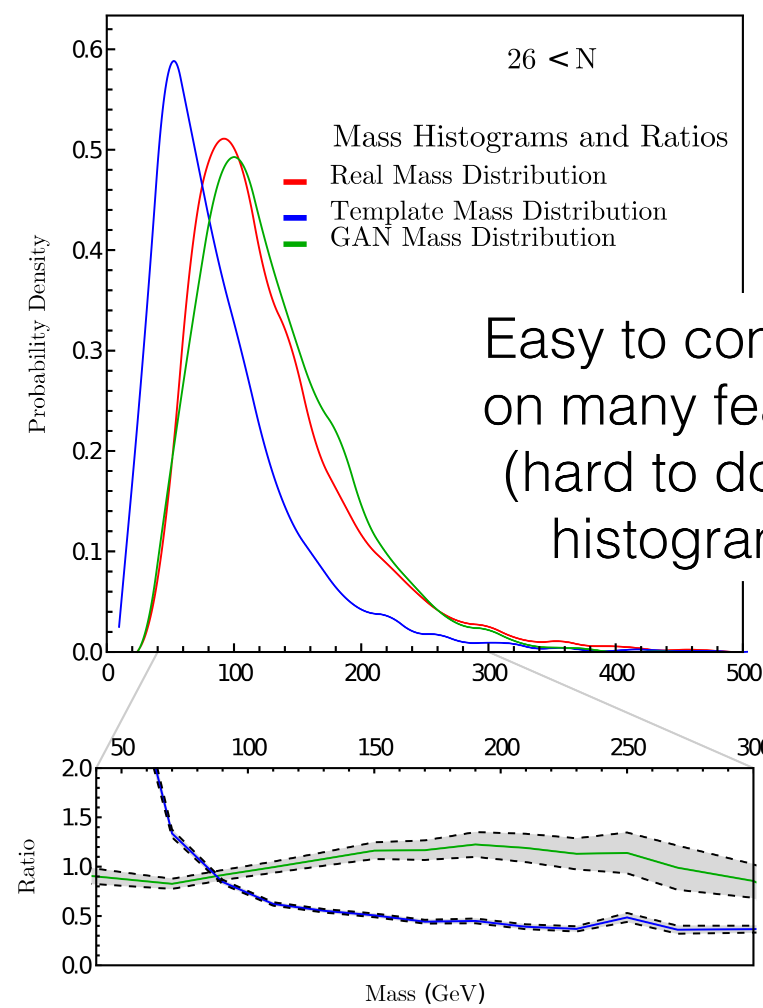
J. Lin, W. Bhimji, B. Nachman 1903.02556

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



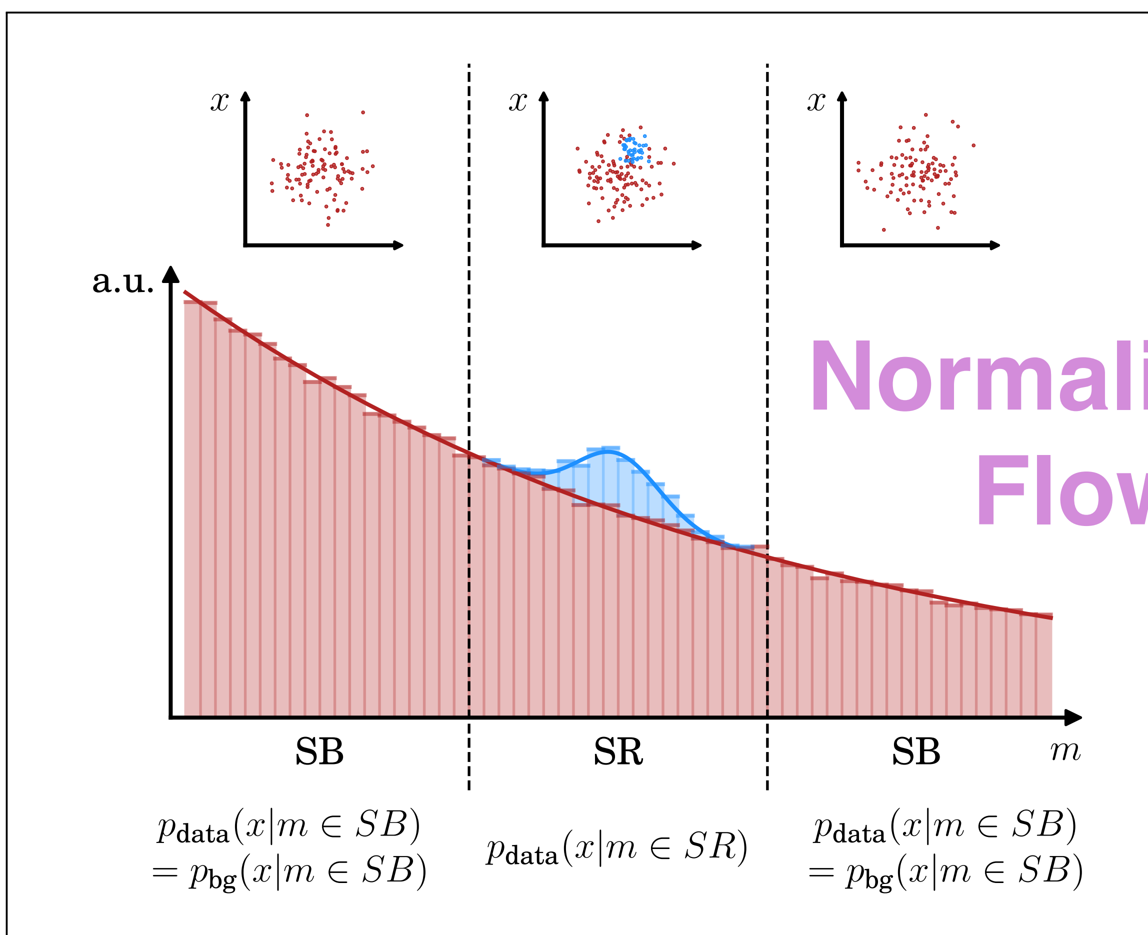


Easy to condition on many features (hard to do with histograms)

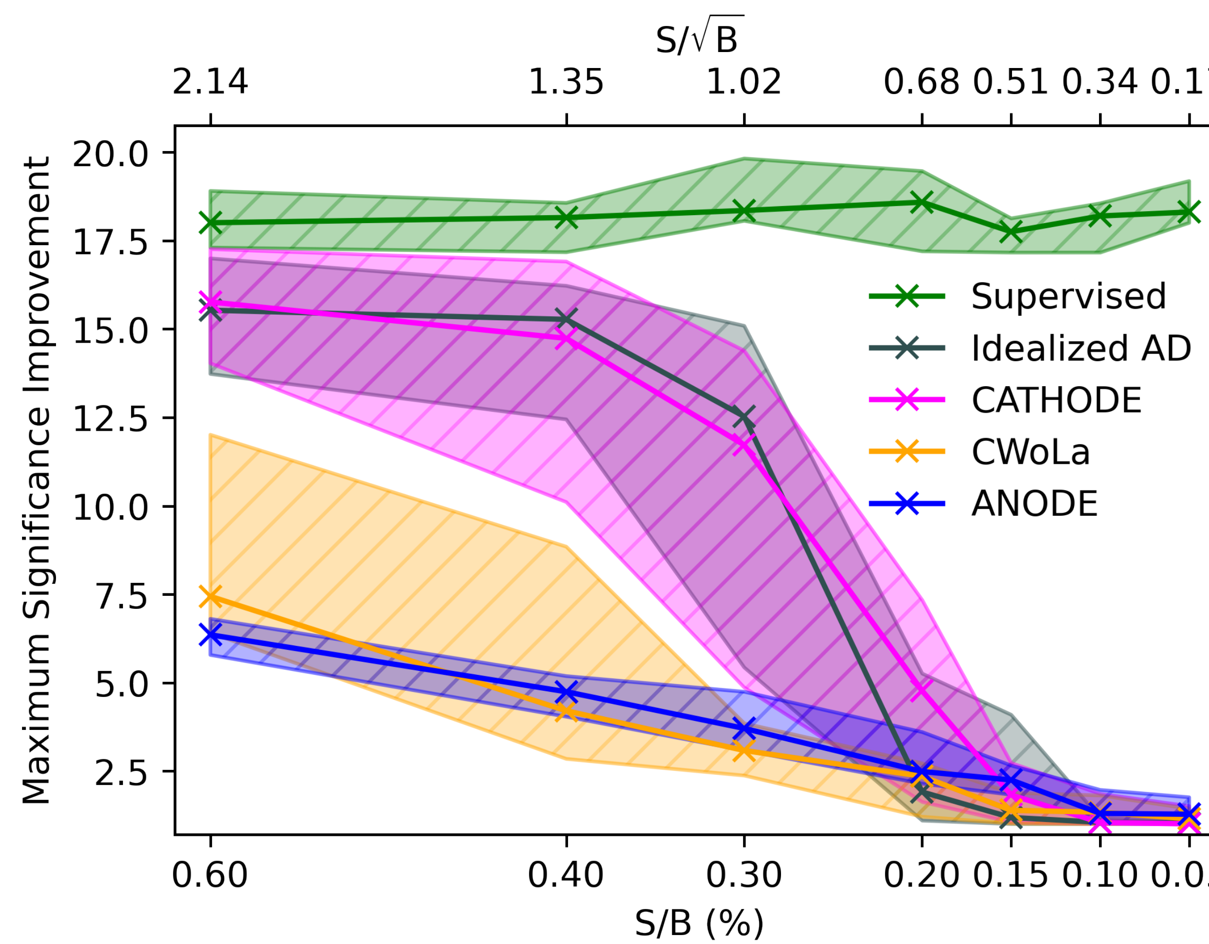
Background Estimation

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



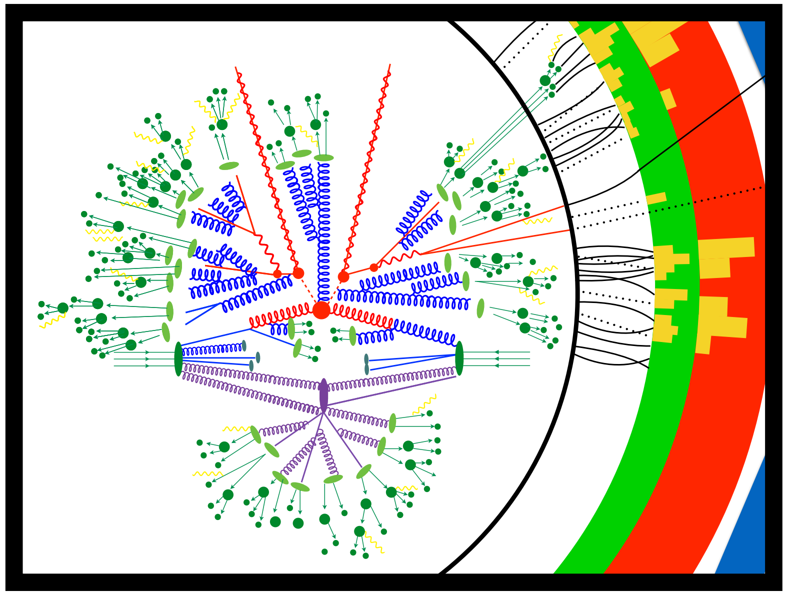
Anomaly detection

A. Butter et al., 2202.09375

Related: Data Compression

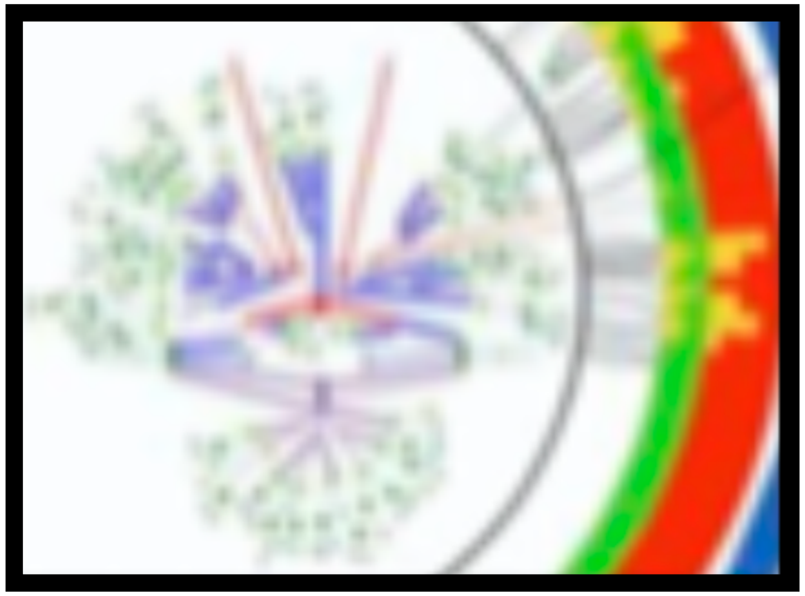
You can think of surrogate models as compressing the data into the parameters of the neural network.

No compression



Many numbers per **event**

Compress per event



Small set of numbers per **event**

Compress entire dataset

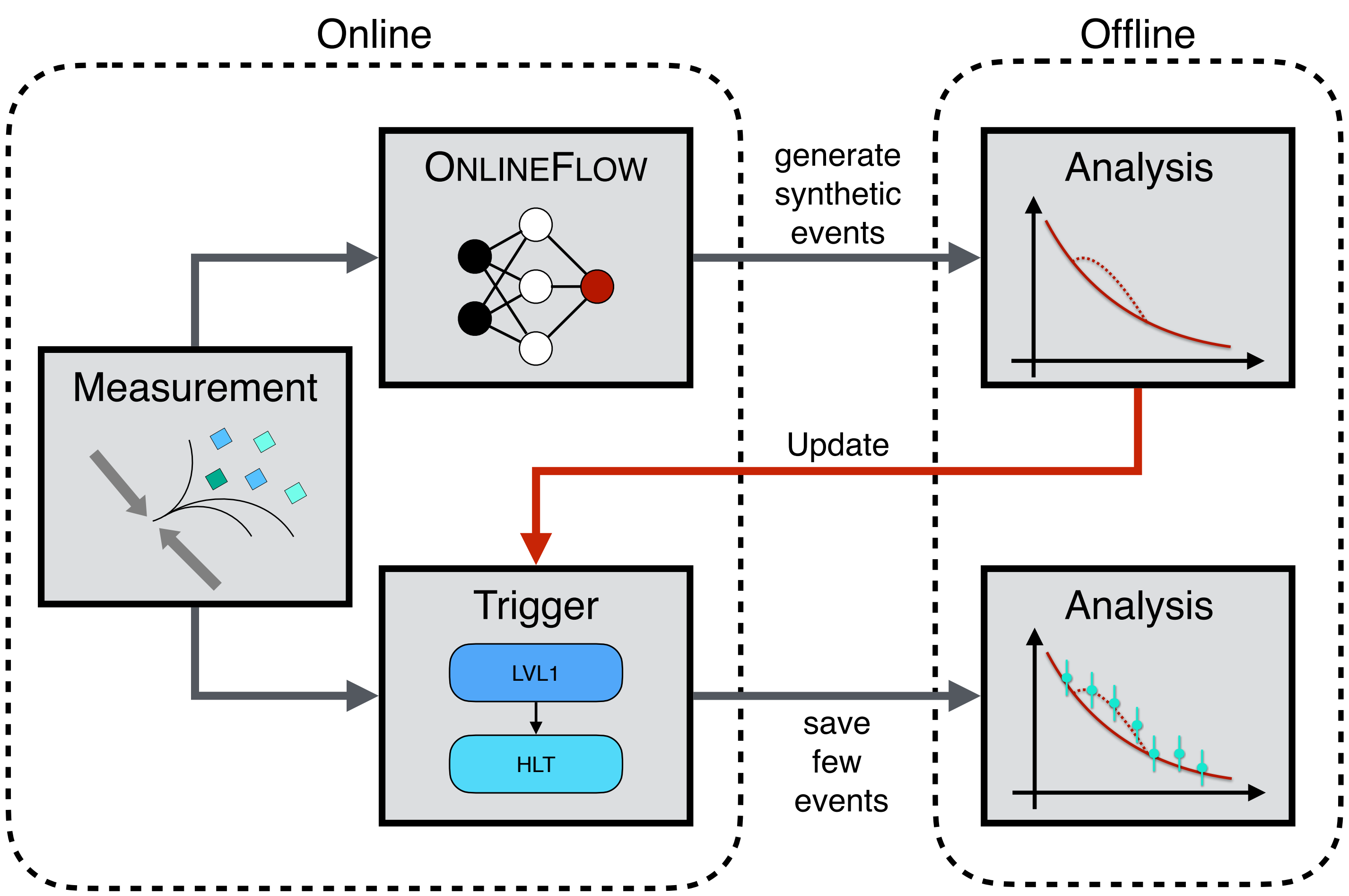
$$p(x\,|\,\theta)$$

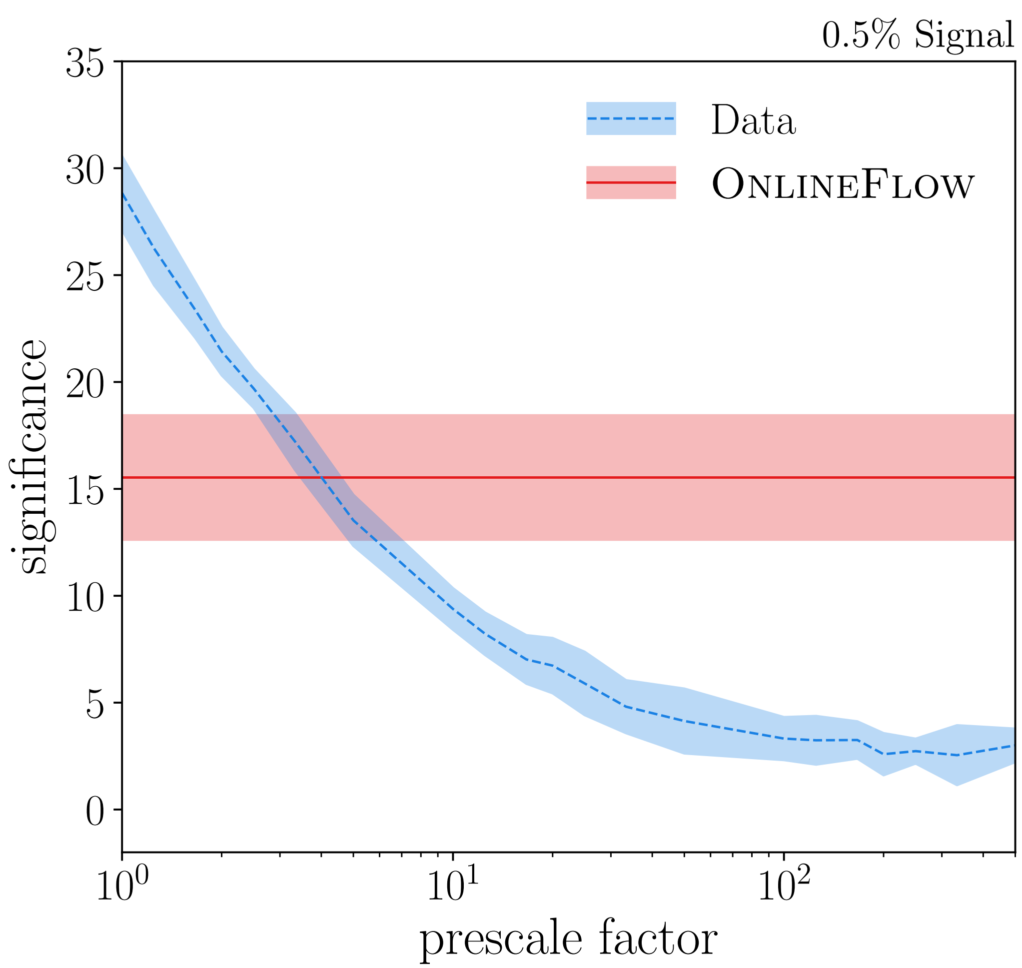
Small set of numbers per **dataset**

Related: Data Compression

You can think of surrogate models as compressing the data into the parameters of the neural network.

Can this also be used for anomaly detection?



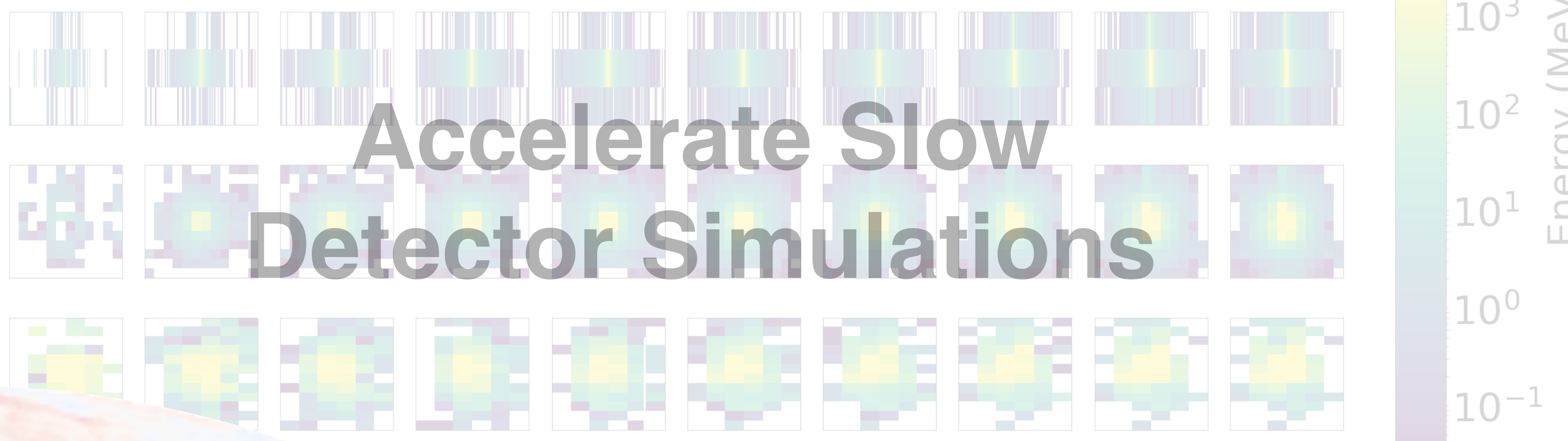


(amount of data discarded by standard trigger)

Generative Models for Particle/Nuclear/Astro

All of these pictures are fake!

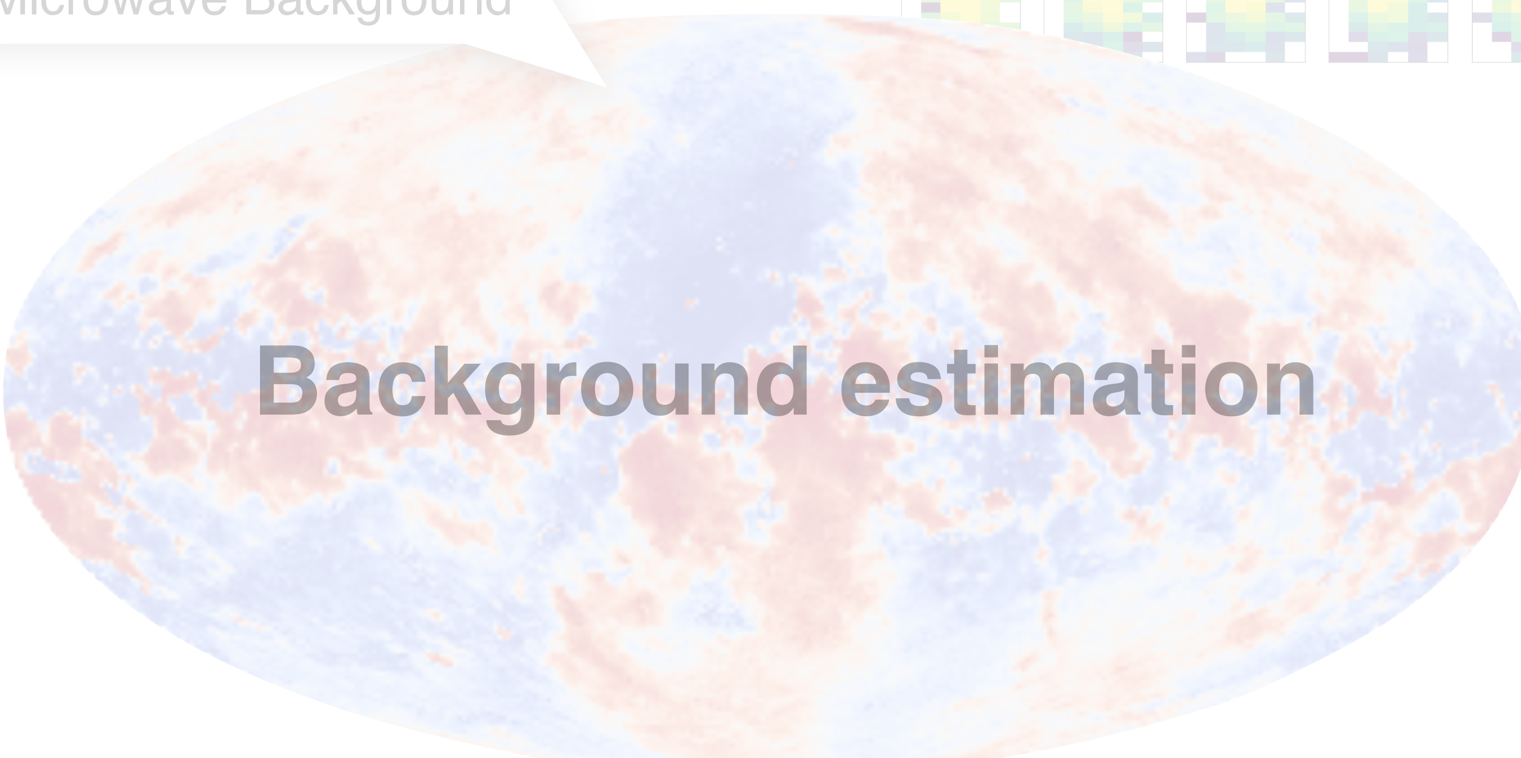
Material Interactions with High Energy Particles



Accelerate Slow Detector Simulations

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

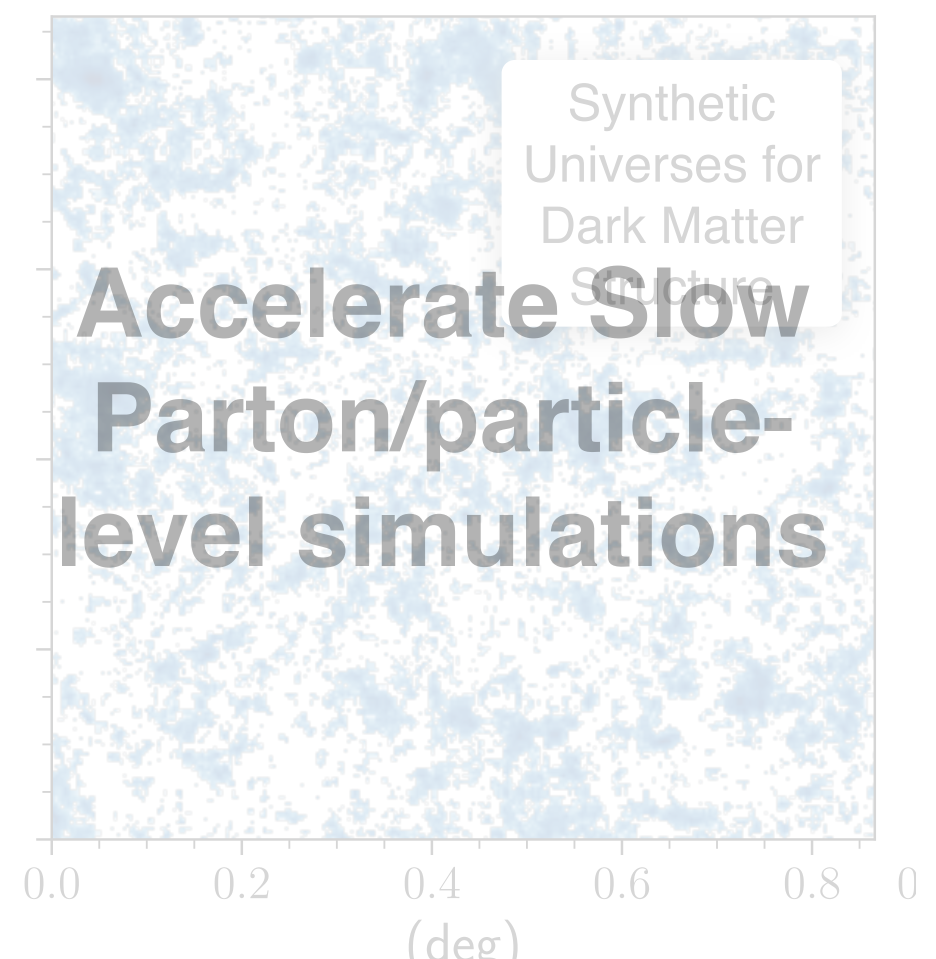
Synthetic Galactic radiation for Cosmic Microwave Background



Background estimation

N. Krachmalnicoff and G. Puglisi, arXiv:2011.02221

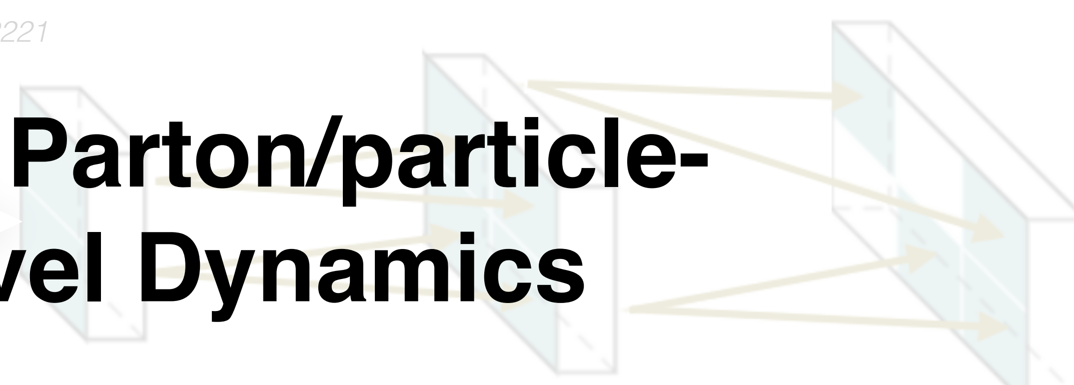
Synthetic Universes for Dark Matter Structure



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

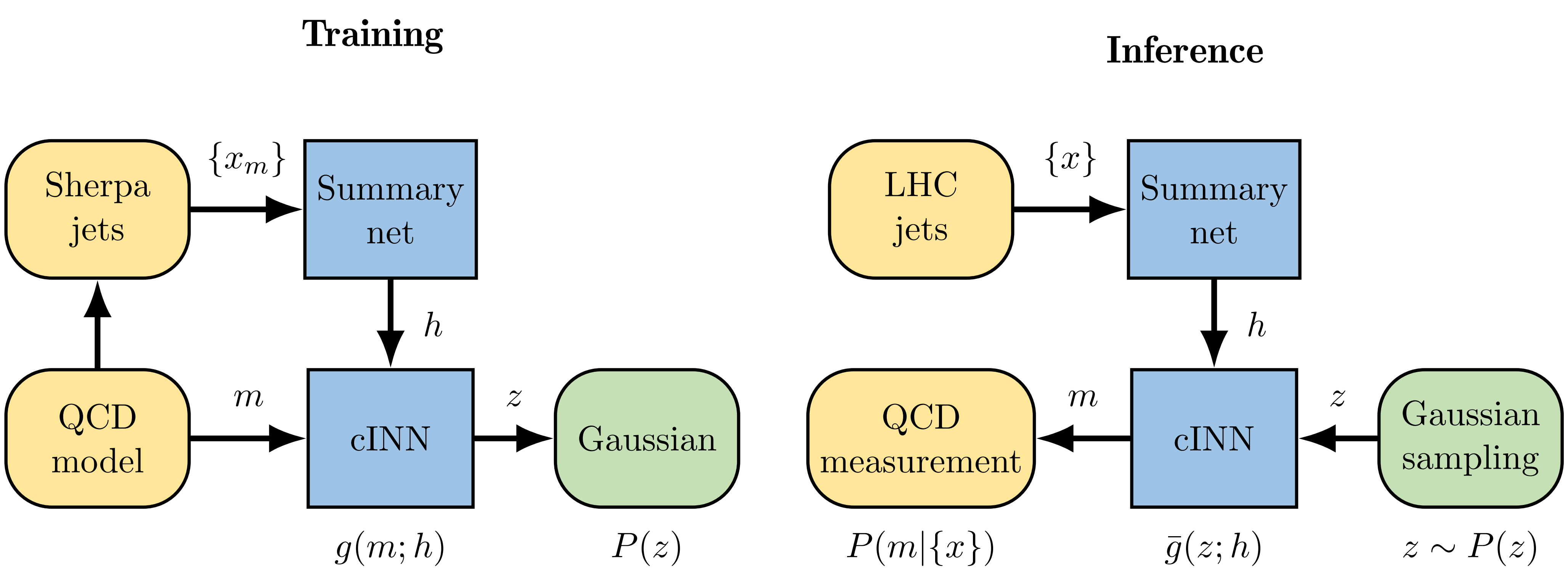
Infering Parton/particle-level Dynamics

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

Infering Parton/particle-level Dynamics

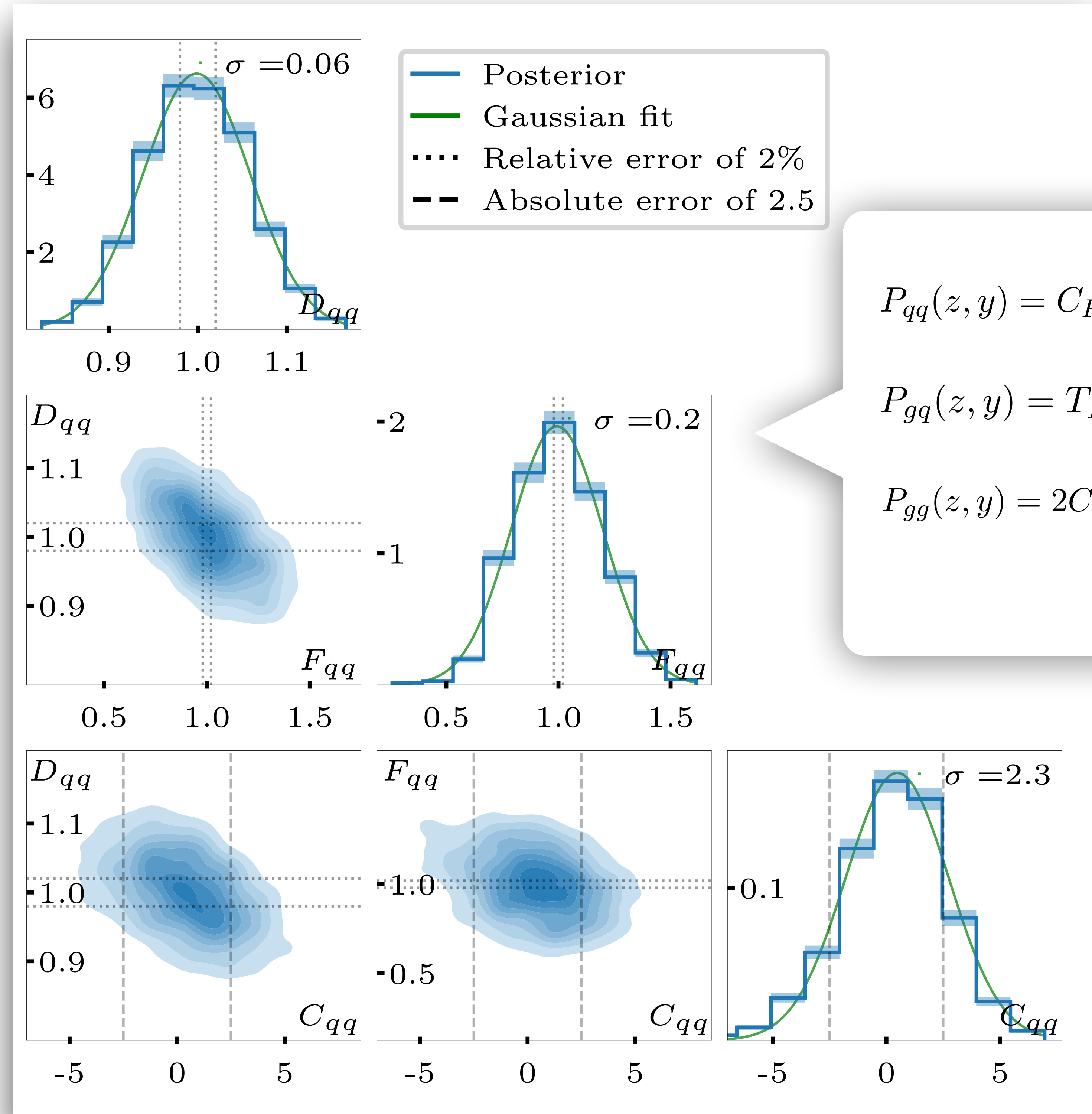
Can we use generative models directly for inference?
(and not “just” for augmenting/accelerating simulation)

Example 1: Inferring fragmentation functions



See also 1804.09720 (“JUNIPR”) and 2012.06582 (GAN-based)

Infering Parton/particle-level Dynamics

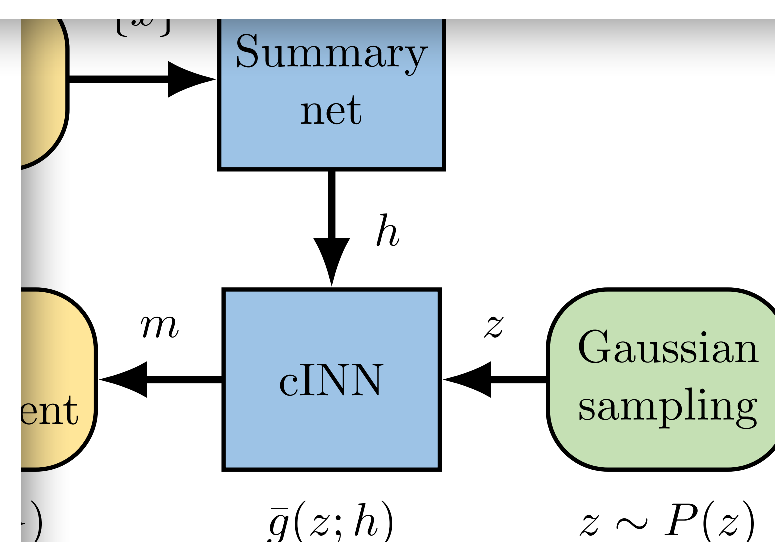


directly for inference?

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

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

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

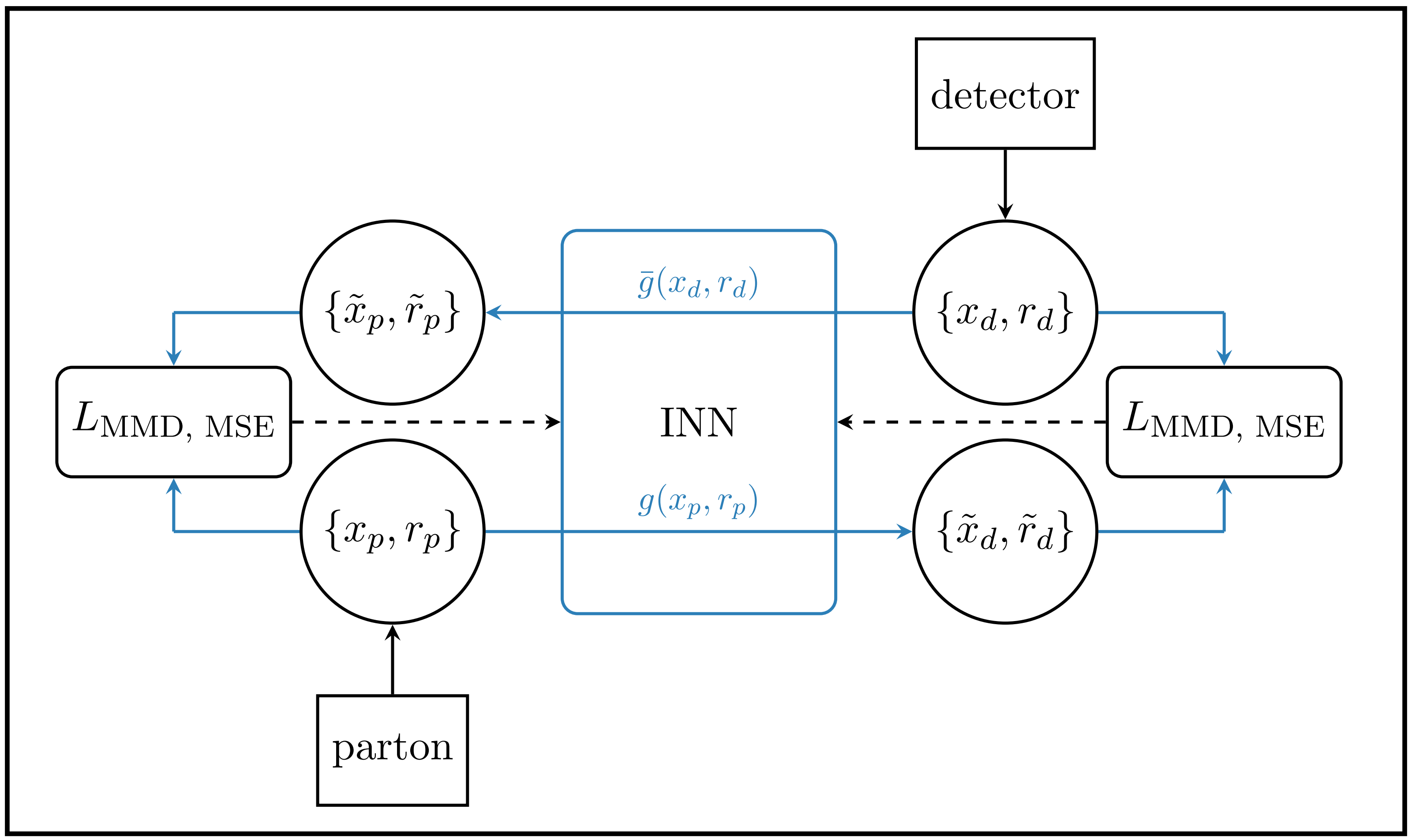


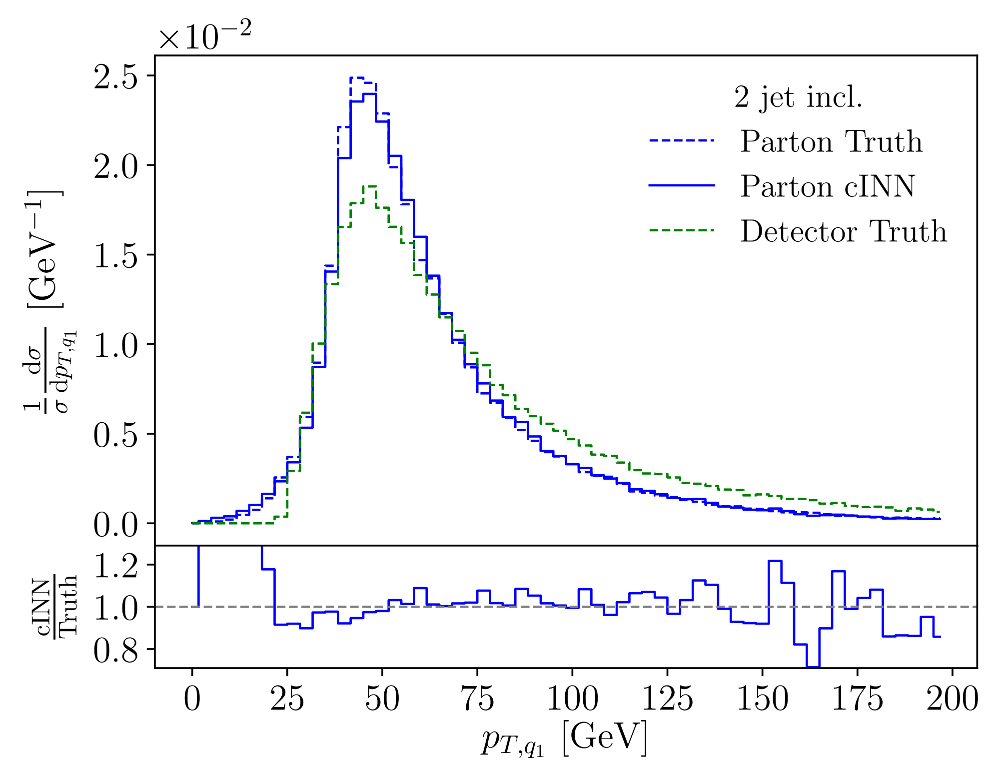
See also 1804.09720 (“JUNIPR”) and 2012.06582 (GAN-based)

Infering Parton/particle-level Dynamics

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

Example 2: Unfolding

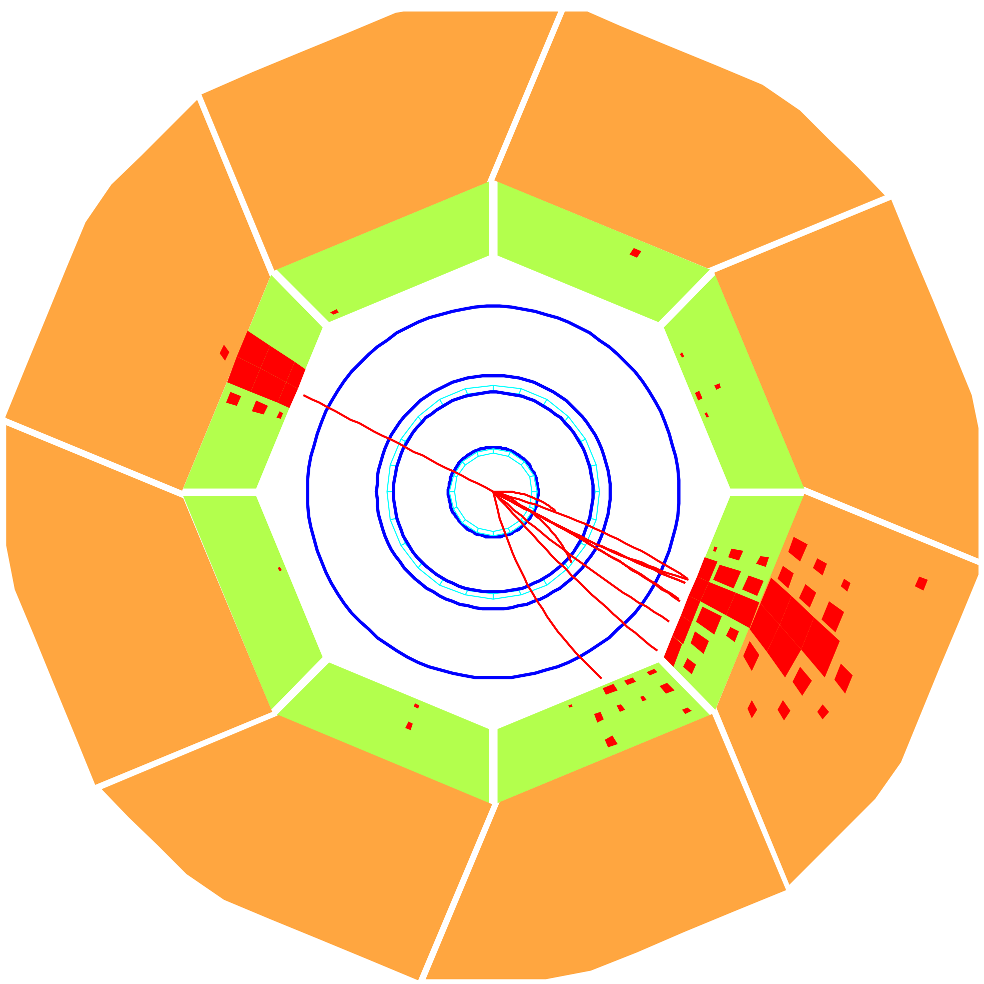


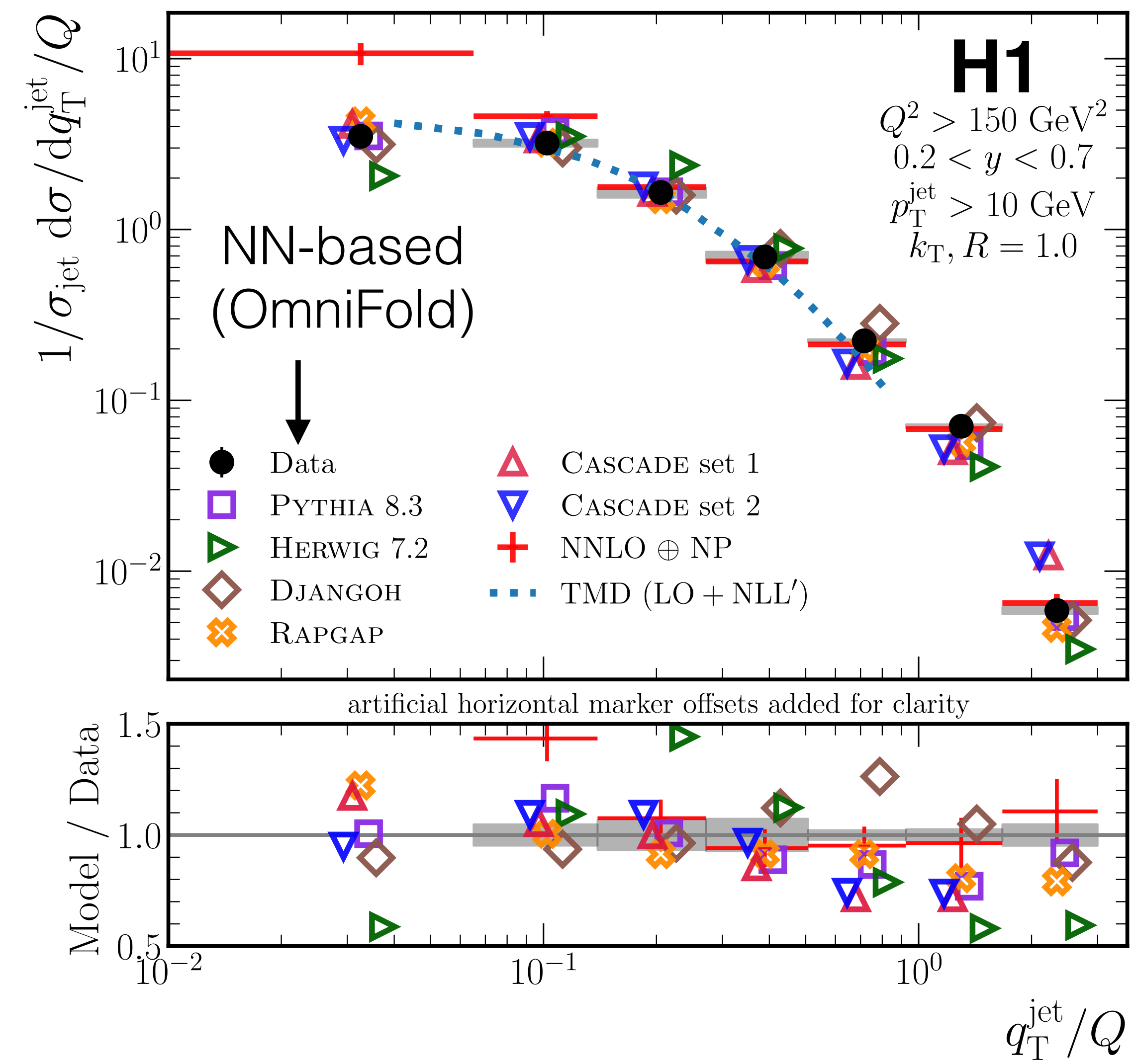


See also 1911.09107 ("OmniFold") and 2101.08944 ("OTUS")

H1 Collaboration, 2108.12376

Infering Parton/particle-level Dynamics





These (and related) methodologies are being studied for *ep* collisions!

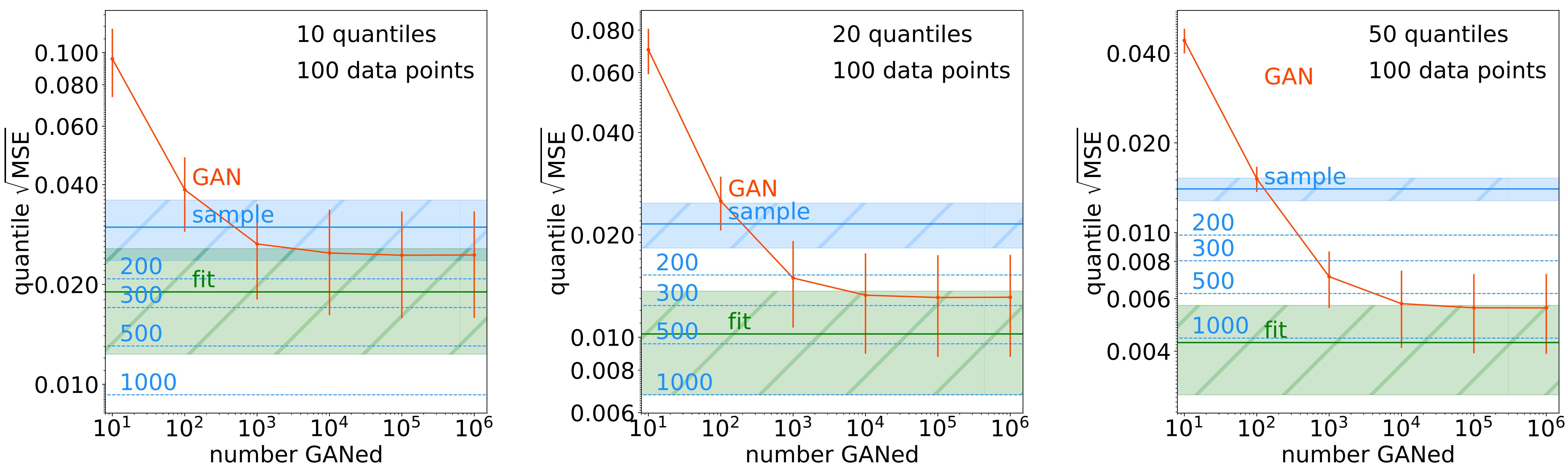
See also 1911.09107 (“OmniFold”) and 2101.08944 (“OTUS”)

A. Butter et al., 2006.06685; S Bieringer et al., 2202.07352

Uncertainties

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

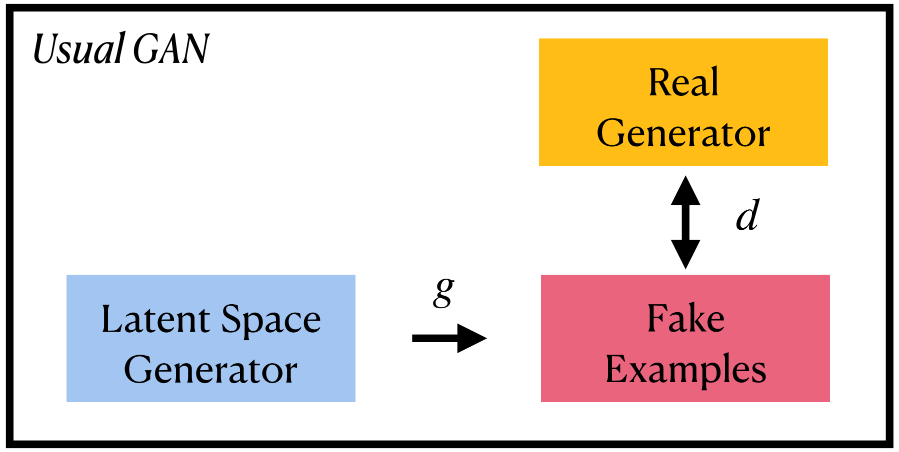
One question is about the **statistical power** of samples from a generative model. This depends on the implicit or explicit information we encode in the networks.

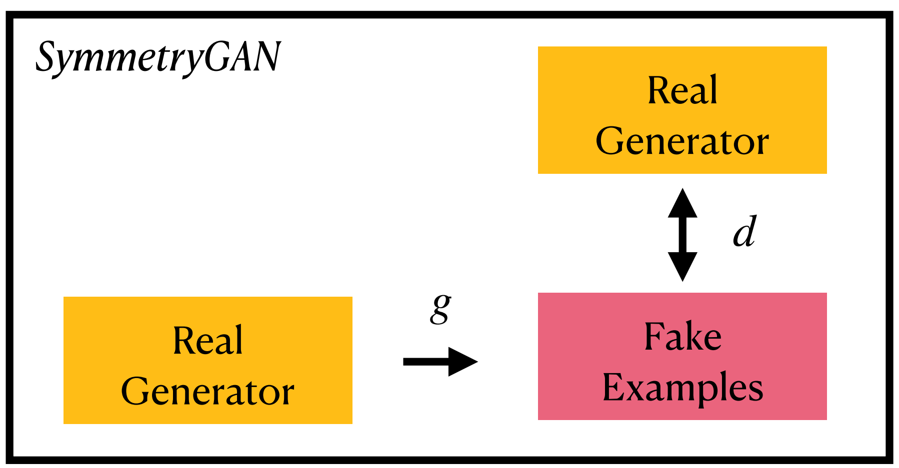


See also 1909.03081, 2002.06307, 2104.04543 (Generative Bayesian NNs), and 2107.08979 (“resampling”)

K. Desai, B. Nachman, J. Thaler, 2112.05722

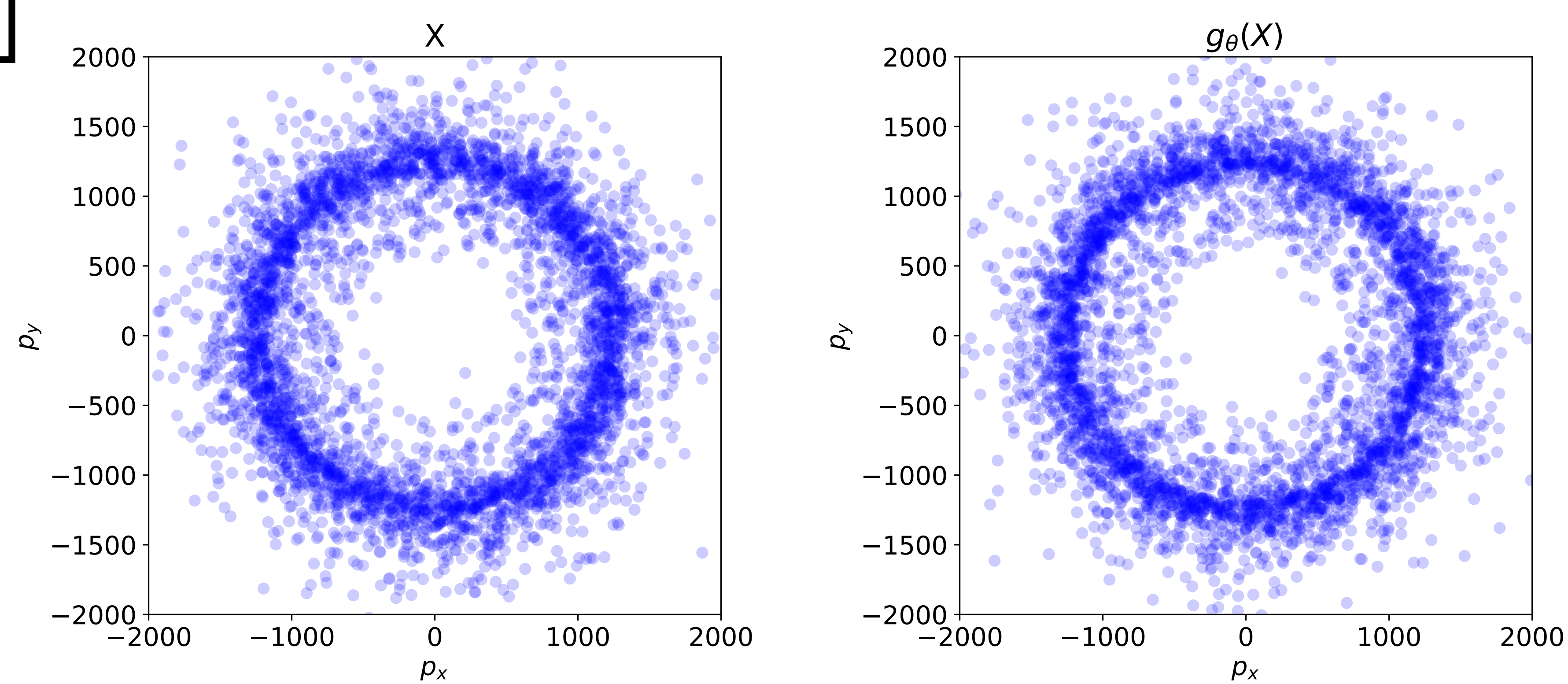
What else can we do?



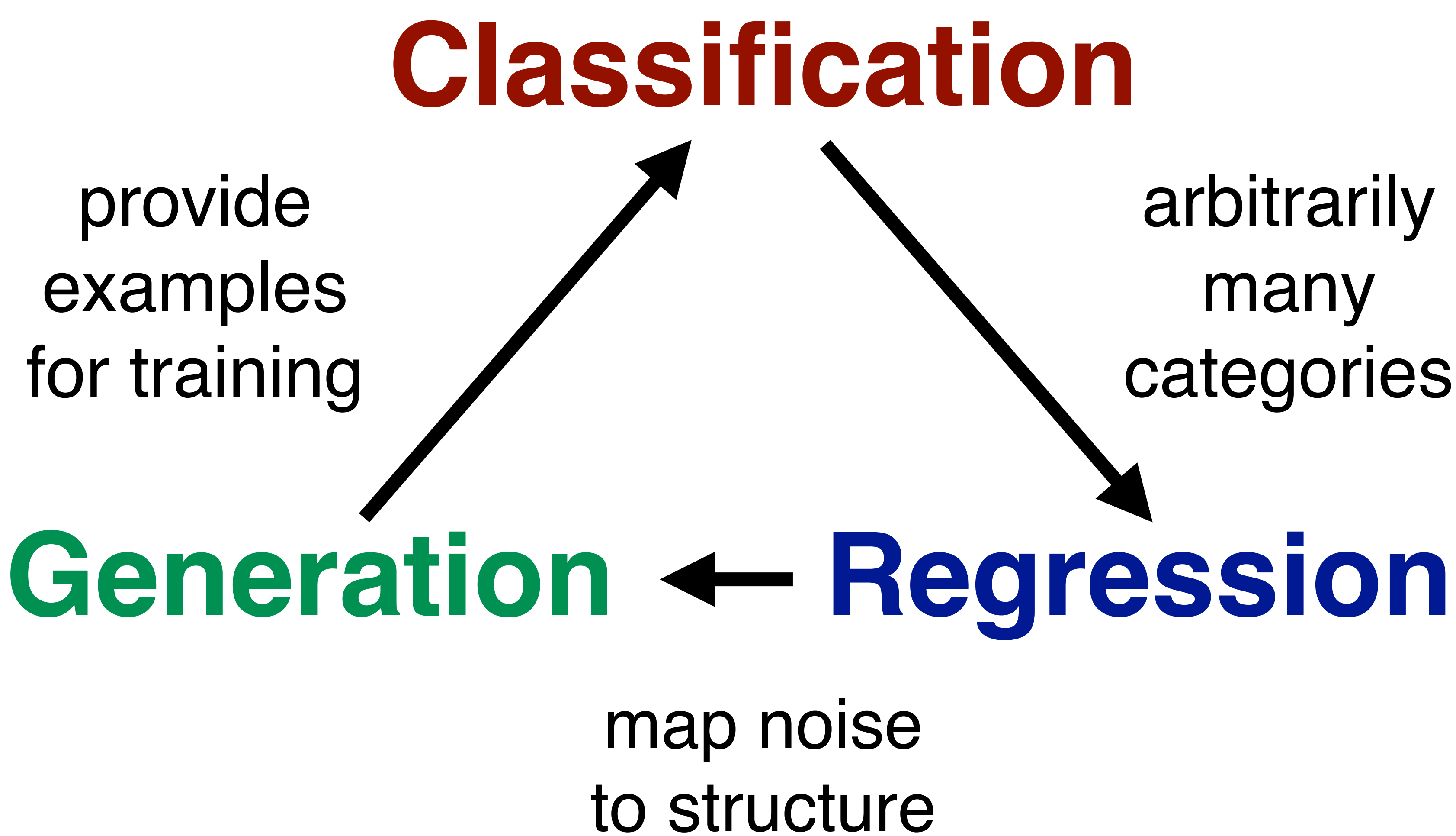


The framework of generative models is quite flexible and we can do more than generate events.

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



Landscape of approaches



+related topics like anomaly detection, simulation-based inference, …

Further reading

https://iml-wg.github.io/HEPML-LivingReview/

HEPML-LivingReview

A Living Review of Machine Learning for Particle Physics

Modern machine learning techniques, including deep learning, is rapidly being applied, adapted, and developed for high energy physics. The goal of this document is to provide a nearly comprehensive list of citations for those developing and applying these approaches to experimental, phenomenological, or theoretical analyses. As a living document, it will be updated as often as possible to incorporate the latest developments. A list of proper (unchanging) reviews can be found within. Papers are grouped into a small set of topics to be as useful as possible. Suggestions are most welcome.

download review

See also https://arxiv.org/abs/2102.02770

Relation to QML

Now we'll hear quite a bit about Quantum Machine Learning

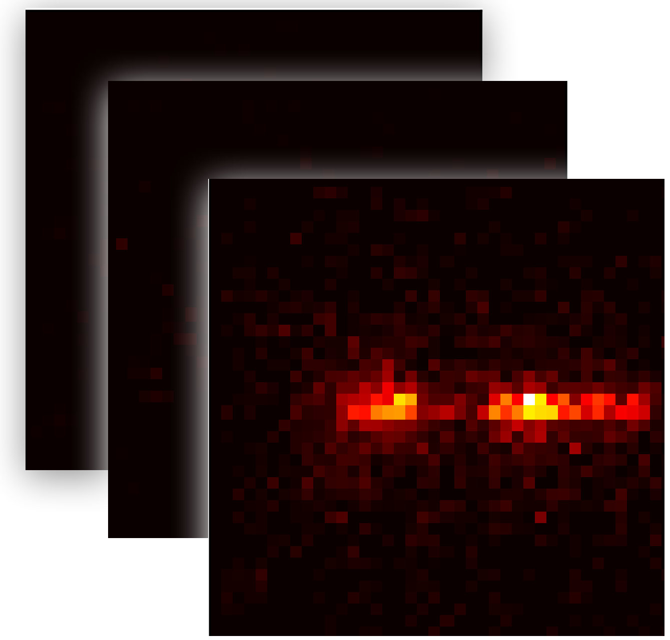
It seems like the jury is still out about the near term utility of QML
(see also Sulaiman's talk tomorrow for a counter point to many recent papers in this area)

QML is an exciting tool and I'll certainly watch developments with interest. I look forward to the results and discussion today and tomorrow!

I'll stress that classical ML is already impacting HEP science and it will continue to grow in importance (!)

Conclusions and Outlook

Particle physics data are complex and unique - modern machine learning tools will help us them to the fullest potential to discover something new and fundamental!



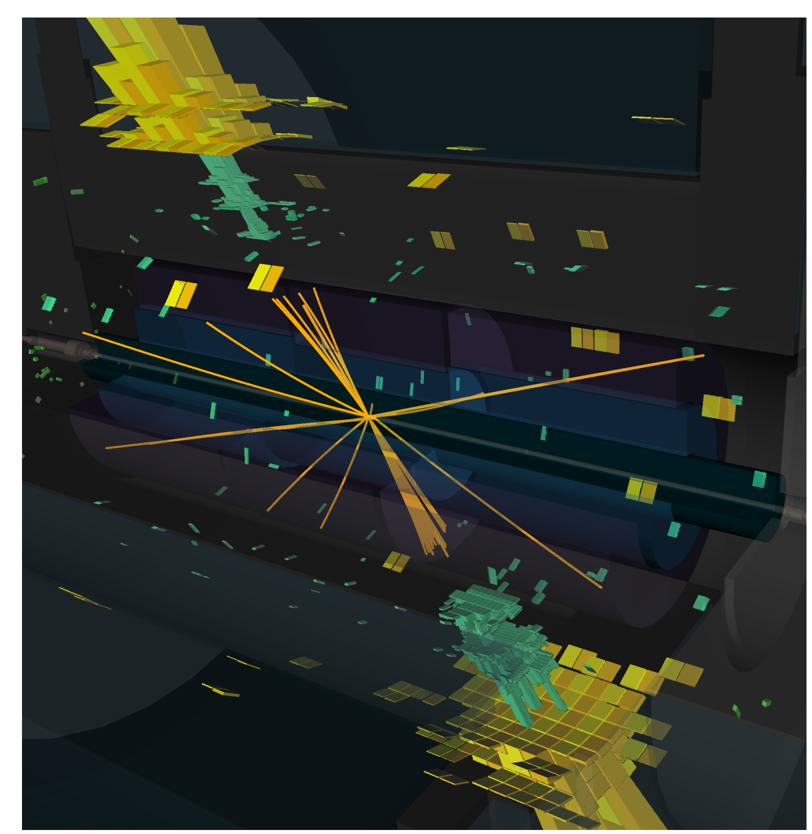
Classification

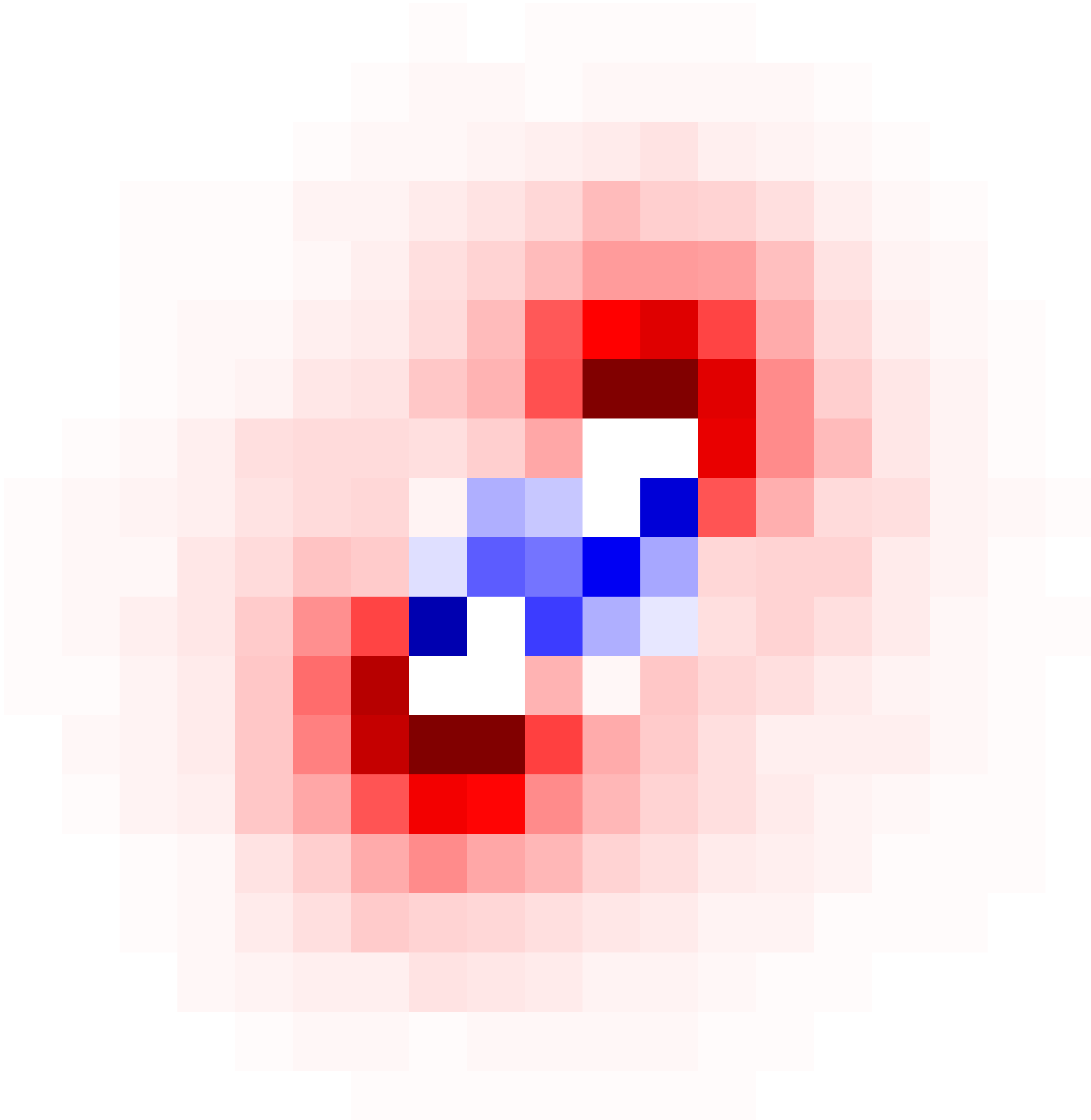
provide examples for training

arbitrarily many categories

Generation ← **Regression**

map noise to structure





Fin.

Backup

107