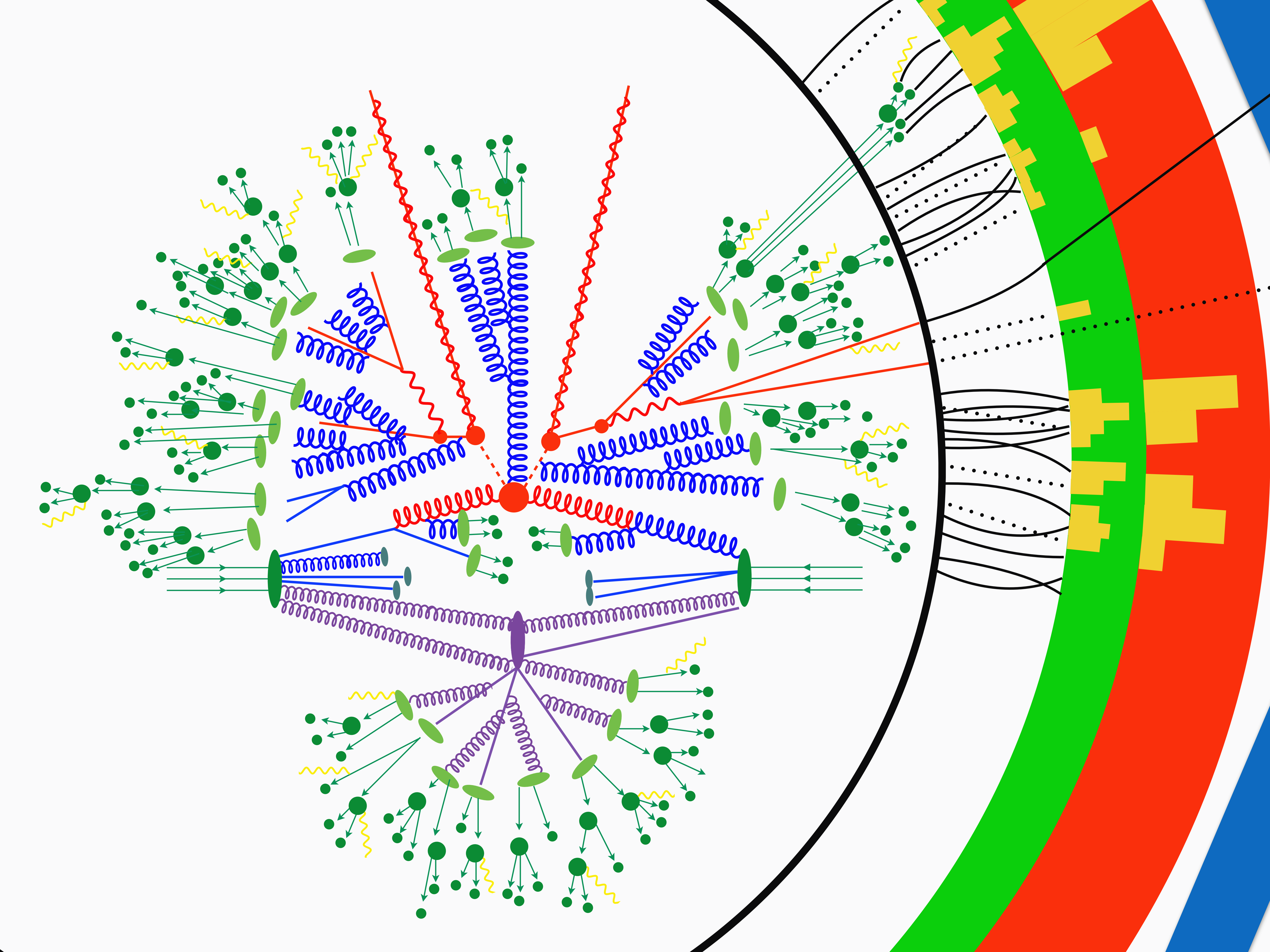
Learning Particle Physics by Example

Location-Aware Generative Adversarial Networks for Physics Synthesis

Luke de Oliveira, Michela Paganini*, Benjamin Nachman



HEP Simulation

THEORY

$$\mathcal{L} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} + i\bar{\psi} \not{D} \psi + h.c. + \bar{\psi}_i y_{ij} \psi_j \phi + h.c. + |D_\mu \phi|^2 - V(\phi)$$

HARD INTERACTIONS (ME CALCULATIONS)

MADGRAPH 5

HADRONIZATION & PARTON SHOWERING

DETECTOR SIM. & INTERACTIONS

Geant 4

DIGITIZATION & PILE-UP

...

Outstanding Issues

Full Simulation is slow

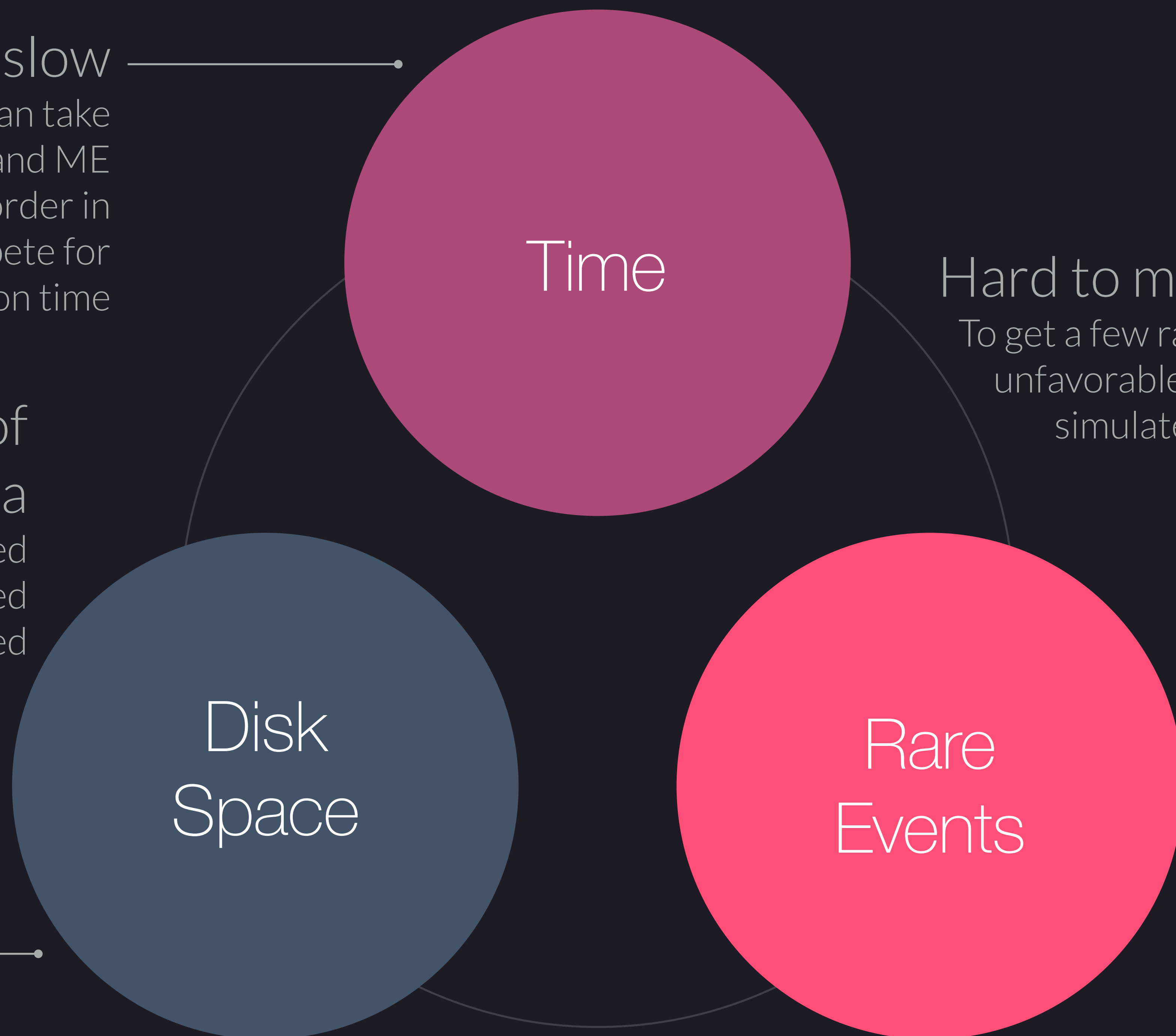
Detector simulation can take *O*(min/event), and ME calculations to high order in perturbation can compete for total generation time

Hard to model tail-statistics

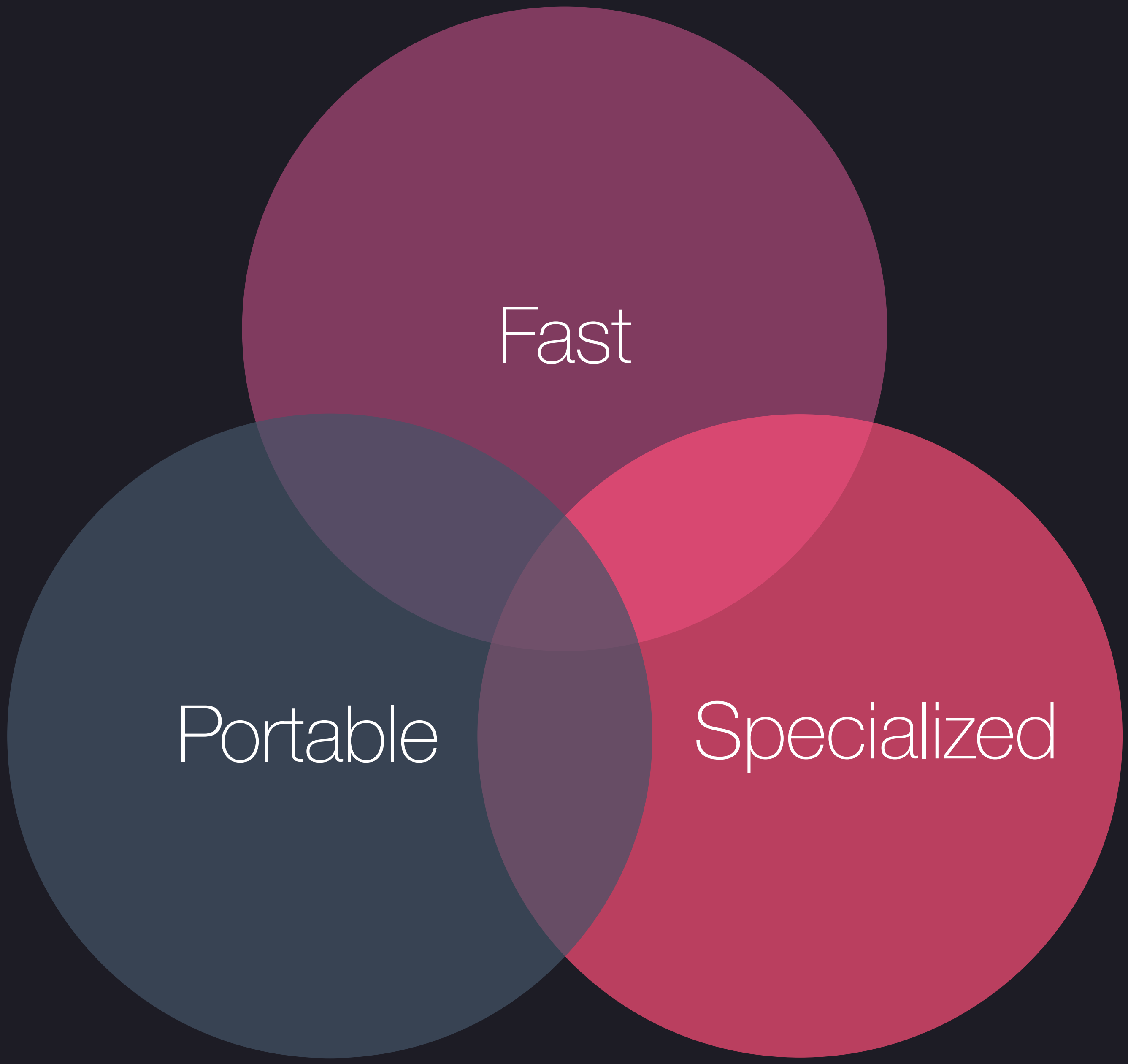
To get a few rare events in kinematically unfavorable regions, we often have to simulate a large number of events

Petabytes of Simulated Data

Large amounts of simulated data needs to be stored and transferred



Looking for a Solution



Generative Models

Traditional Solutions

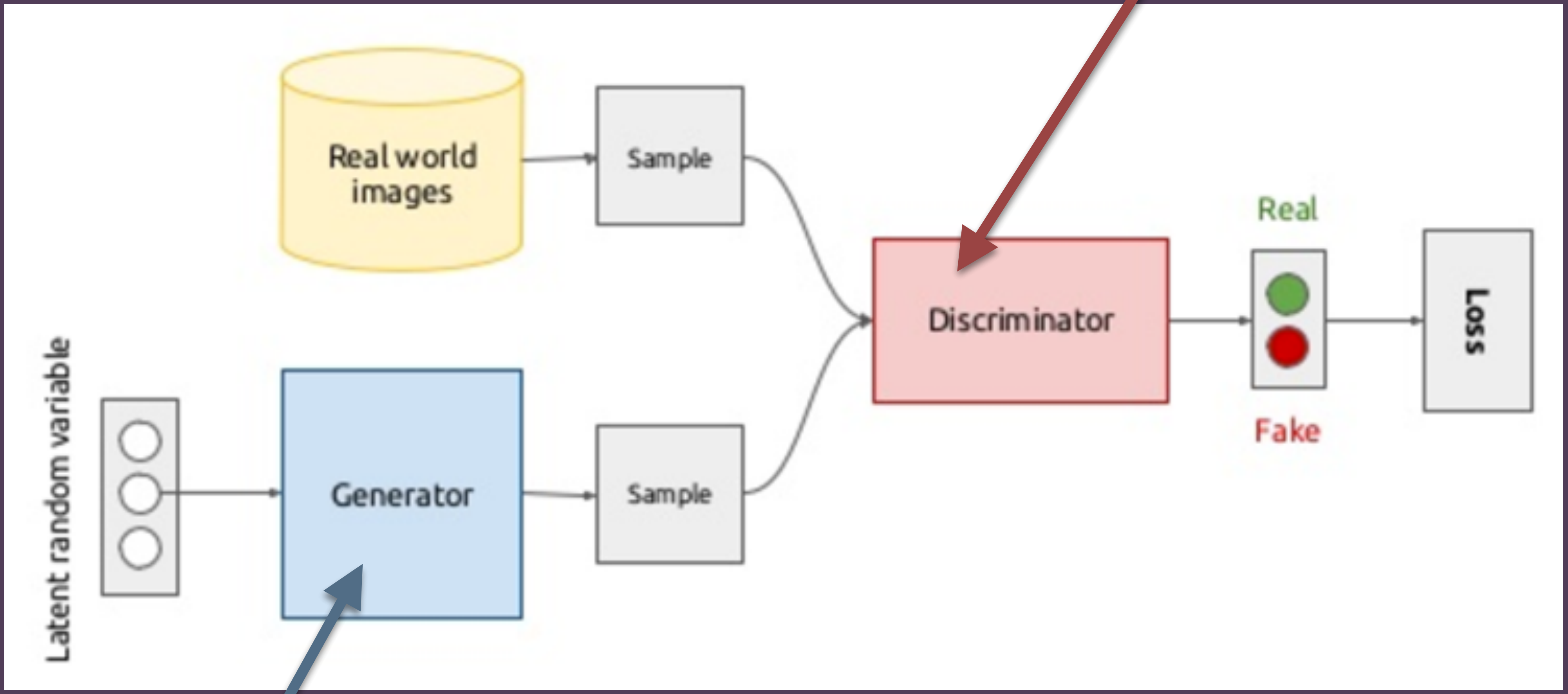
- Hidden Markov Models
- Gaussian Mixture Models
- (Restricted) Boltzmann Machines
- Helmholtz Machines

More Modern Solutions:

- Deep Belief Nets
- Variational Auto-Encoders
- Autoregressive Models (PixelCNN, WaveNet, …)
- Generative Adversarial Networks

Generative Adversarial Networks

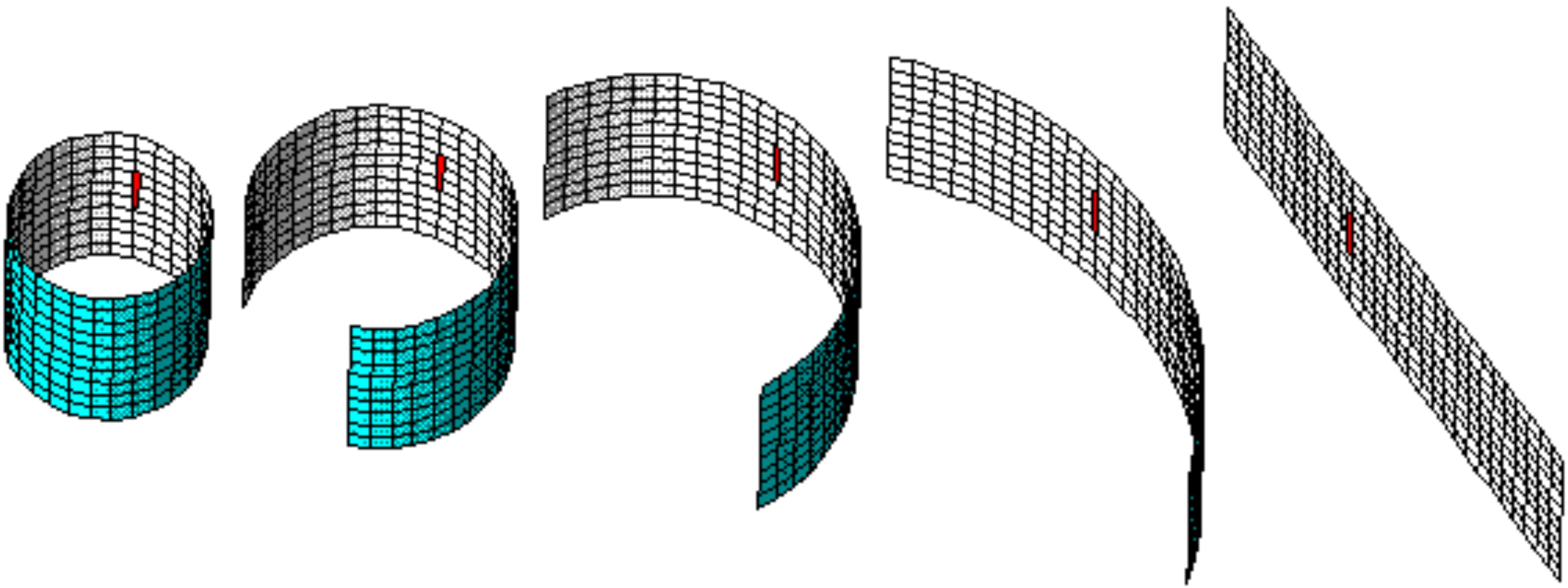
tries to distinguish real images from generated images



tries to turn noise into credible samples

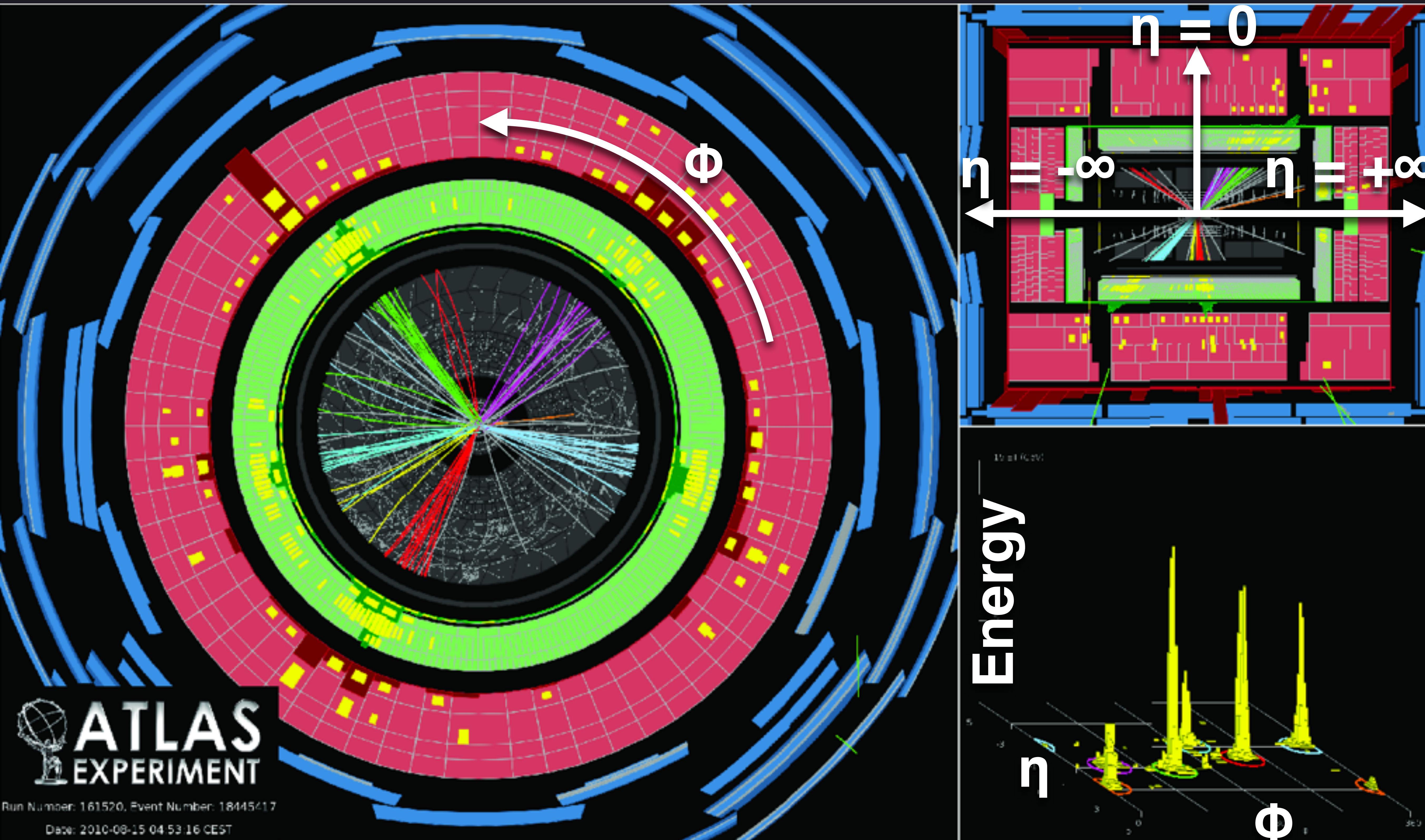
Step 0: Simplified Task

Step 0: Simplified Task



single layer calorimeter

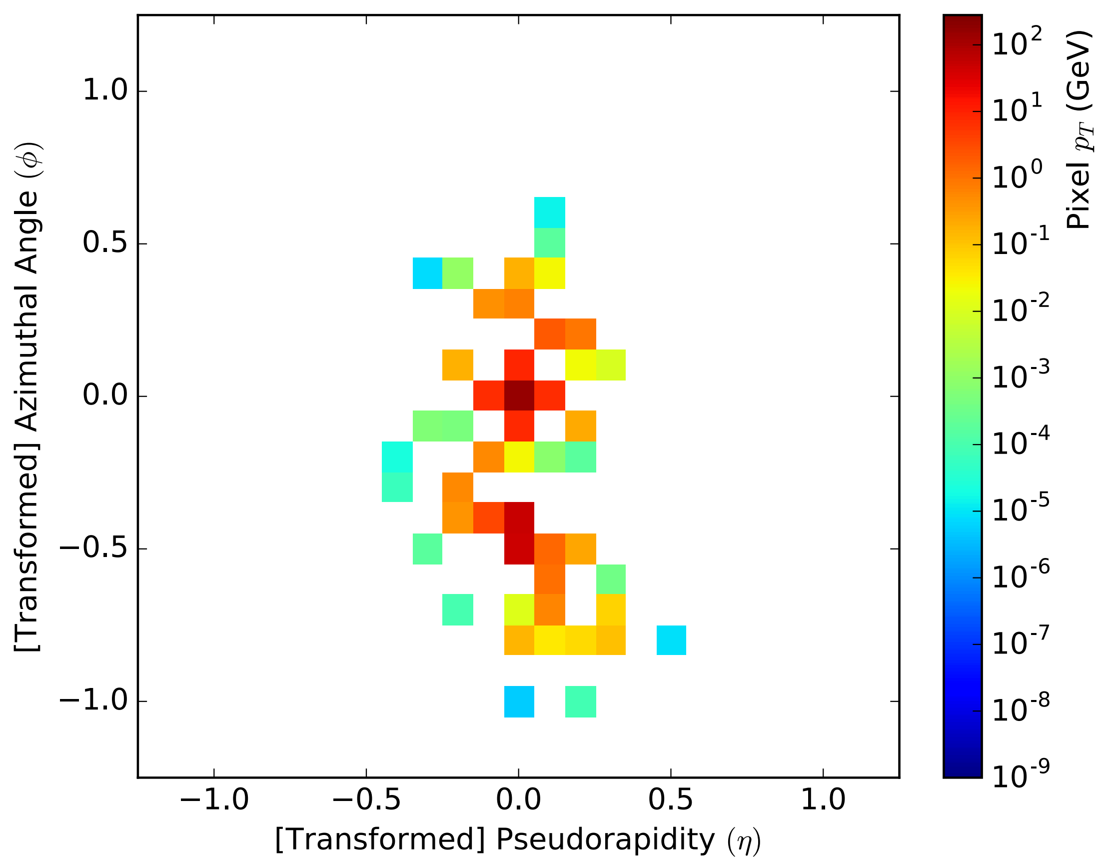
Jets


Φ
η = 0
η = -∞
η = +∞
Energy
η
Φ
ATLAS
EXPERIMENT
Run Number: 161520, Event Number: 18445417
Date: 2010-08-15 04:53:16 CEST

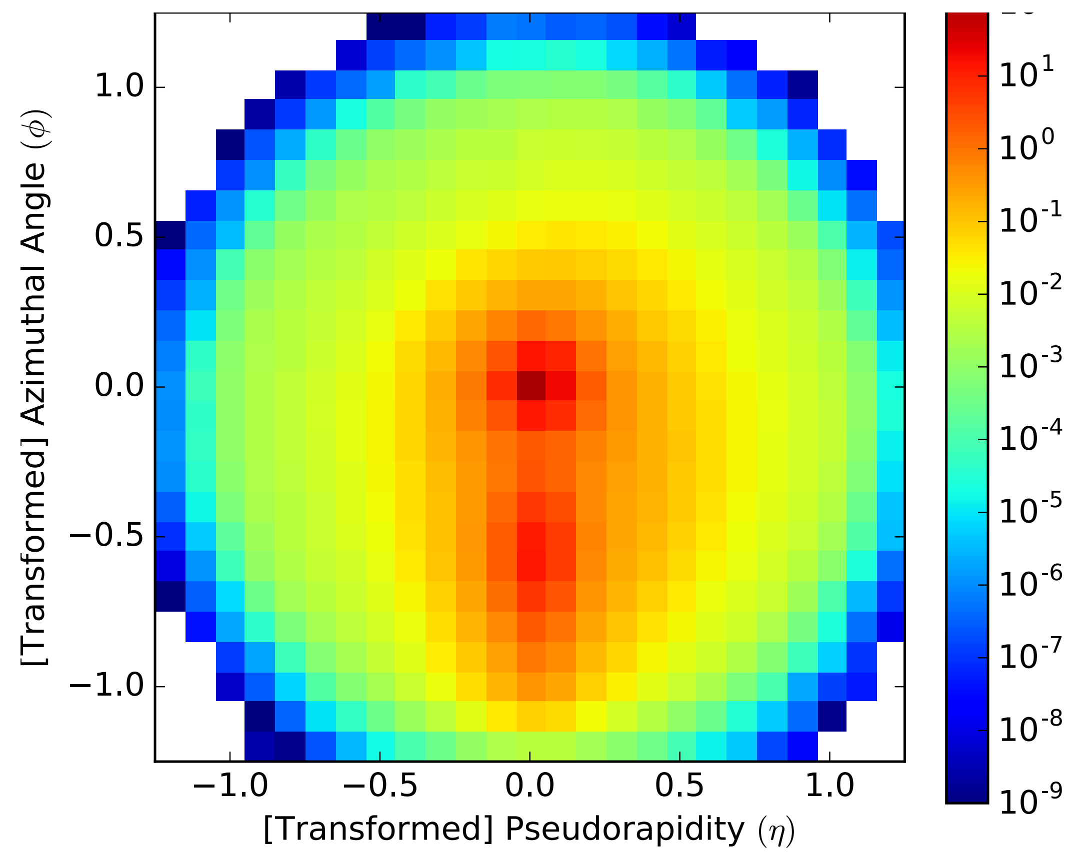
Jet Images

Pixelate ⟶ Translate ⟶ Rotate ⟶ Re-Grid ⟶ Flip

Single Jet Image



Average of Thousands of Jet Images



References:

J. Cogan, M. Kagan, E. Strauss and A. Schwartzman, *Jet-Images: Computer Vision Inspired Techniques for Jet Tagging*, JHEP 02 (2015) 118 [1407.5675].

L. de Oliveira, M. Kagan, L. Mackey, B. Nachman and A. Schwartzman, *Jet-images — deep learning edition*, JHEP 07 (2016) 069 [1511.05190].

L. G. Almeida, M. Backovi´c, M. Cliche, S. J. Lee and M. Perelstein, *Playing Tag with ANN: Boosted Top Identification with Pattern Recognition*, JHEP 07 (2015) 086 [1501.05968].

P. T. Komiske, E. M. Metodiev and M. D. Schwartz, *Deep learning in color: towards automated quark/gluon jet discrimination*, [1612.01551].

J. Barnard, E. N. Dawe, M. J. Dolan and N. Rajcic, *Parton Shower Uncertainties in Jet Substructure Analyses with Deep Neural Networks,* [1609.00607].

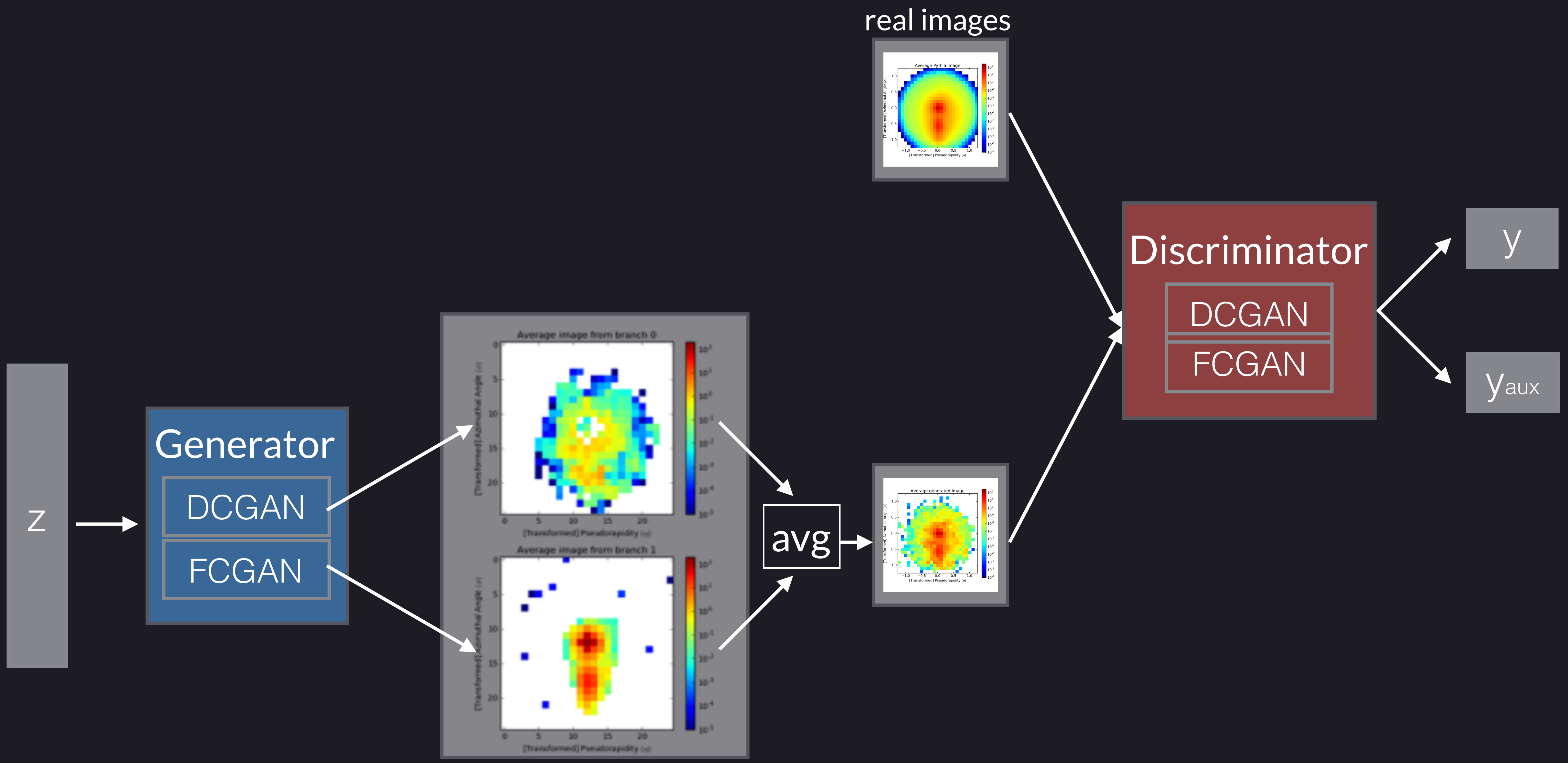
P. Baldi, K. Bauer, C. Eng, P. Sadowski and D. Whiteson, *Jet Substructure Classification in High-Energy Physics with Deep Neural Networks*, Phys. Rev. D93 (2016), no. 9 094034 [1603.09349].

Samples

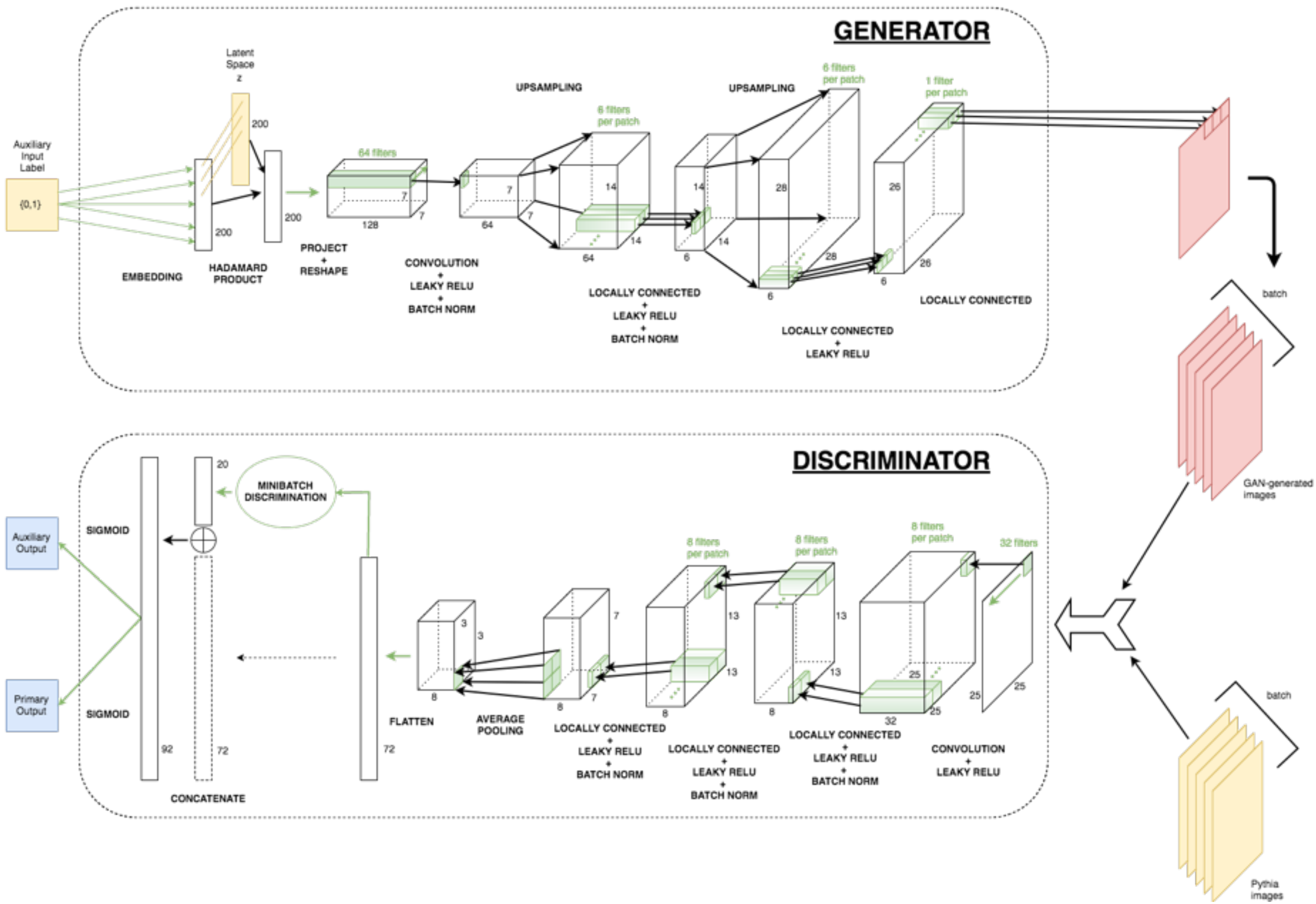
- Jet images produced with PYTHIA 8.219
- Two processes: QCD vs boosted W from W’—>WZ
- pT range = [250, 300] GeV
- Jet clustering with anti-kt (R=1.0) with FASTJET 3.2.1
- Trimming by re-clustering the constituents into R=0.3 kt subjets and dropping those with <5% jet pT

Benchmark Models

- DCGAN — convolutional layers in both *G* and *D*
- FCGAN — fully-connected layers in both *G* and *D*
- HYBRIDGAN — a combination of the two:

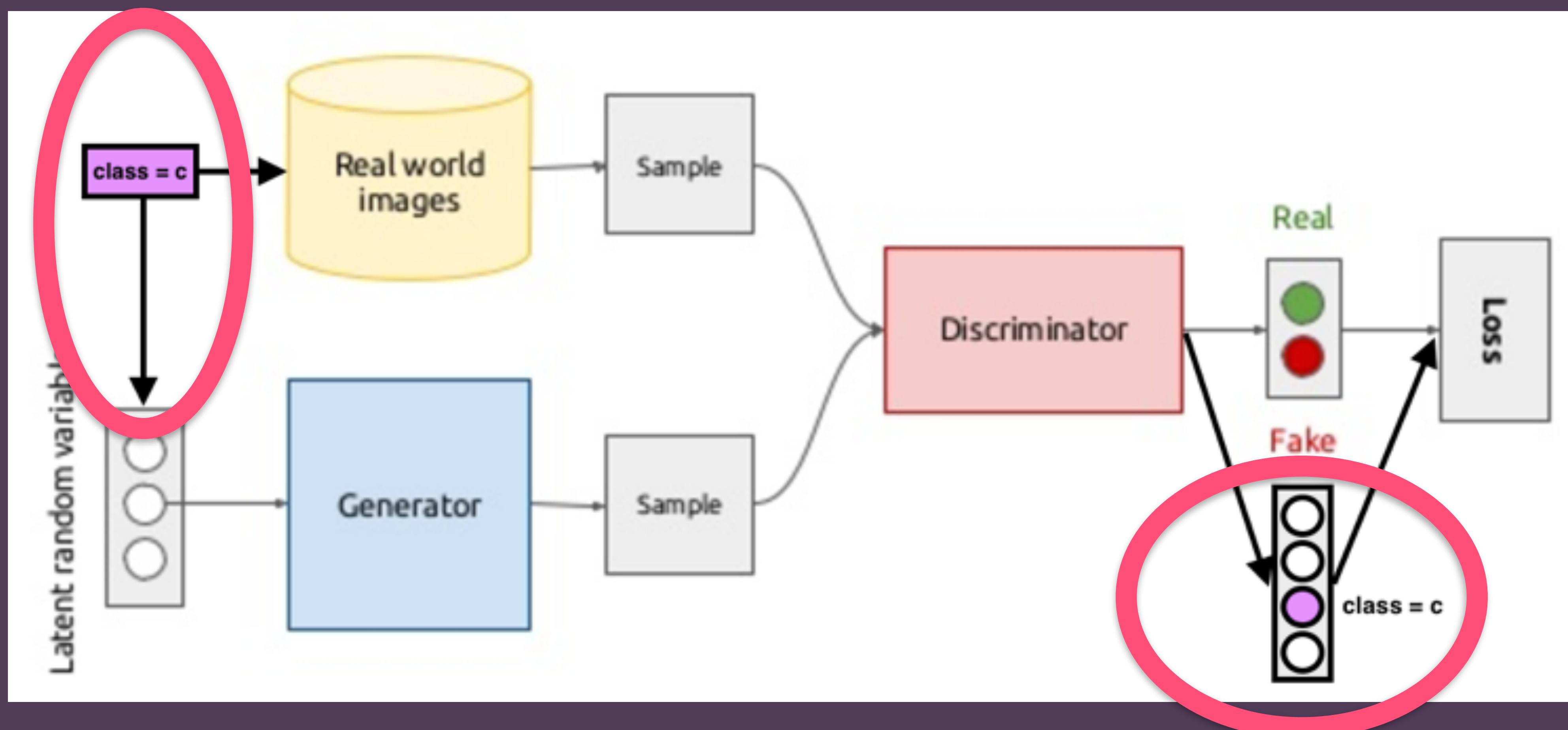


LAGAN (Location Aware Generative Adversarial Network)



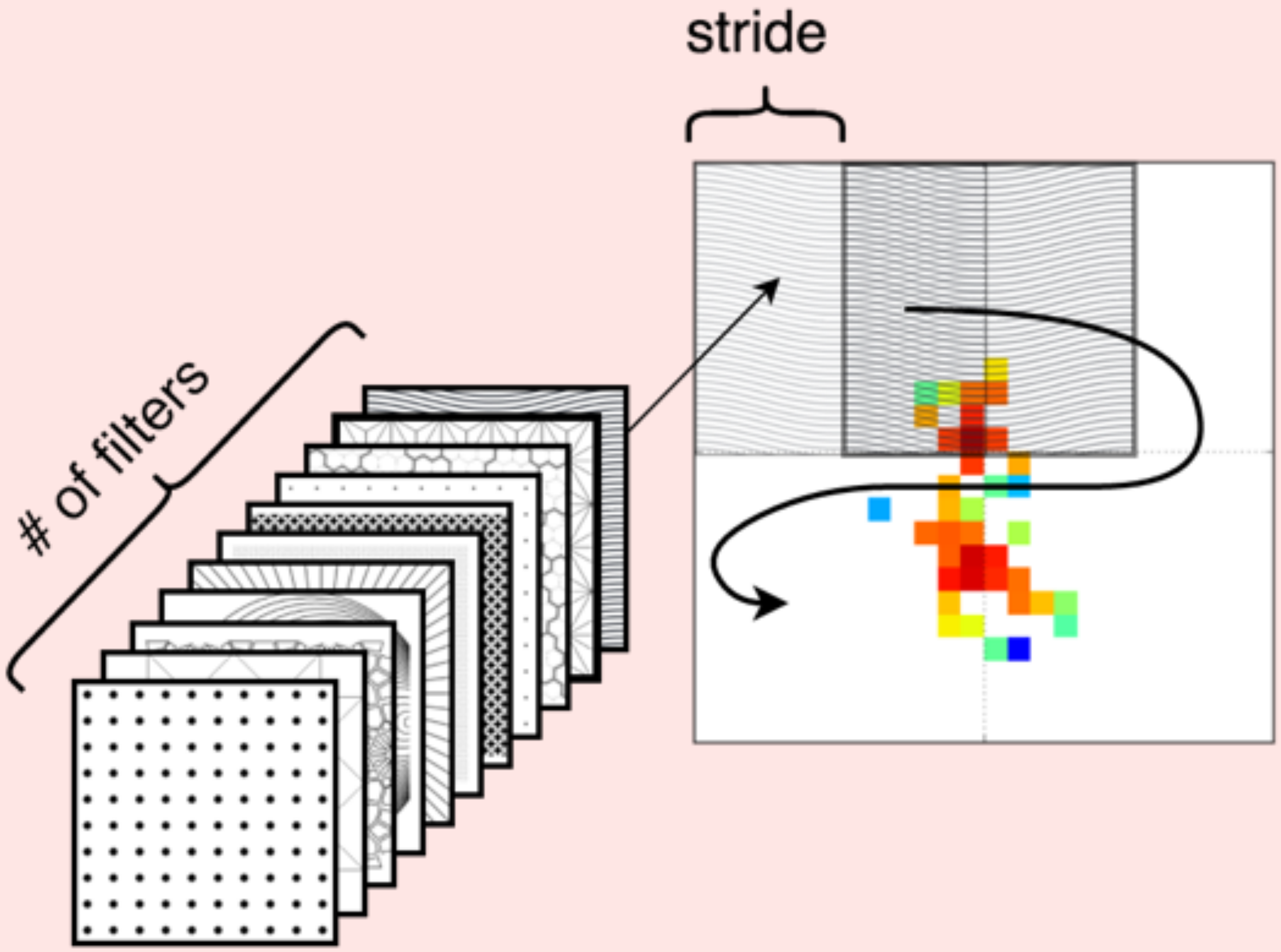
Starting from ACGAN (Auxiliary Classifier GAN)

Request a specific class
(here, W or QCD)



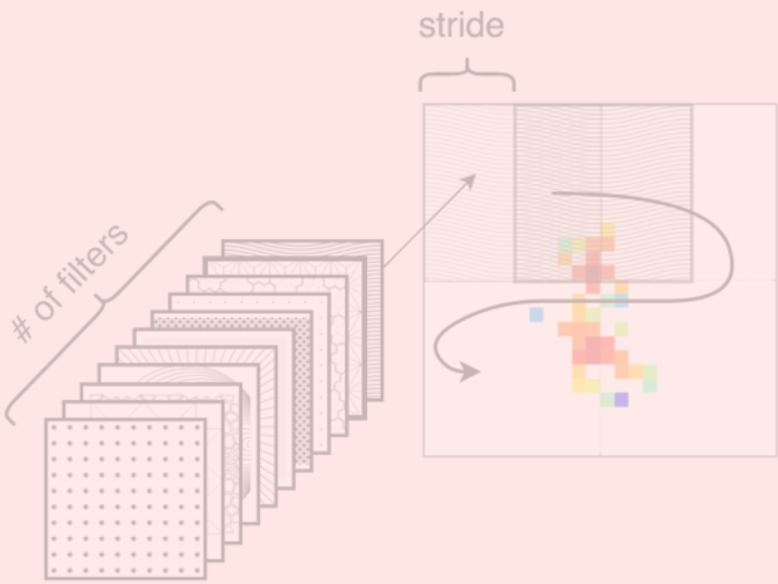
Correctly classify samples

Layer Comparison

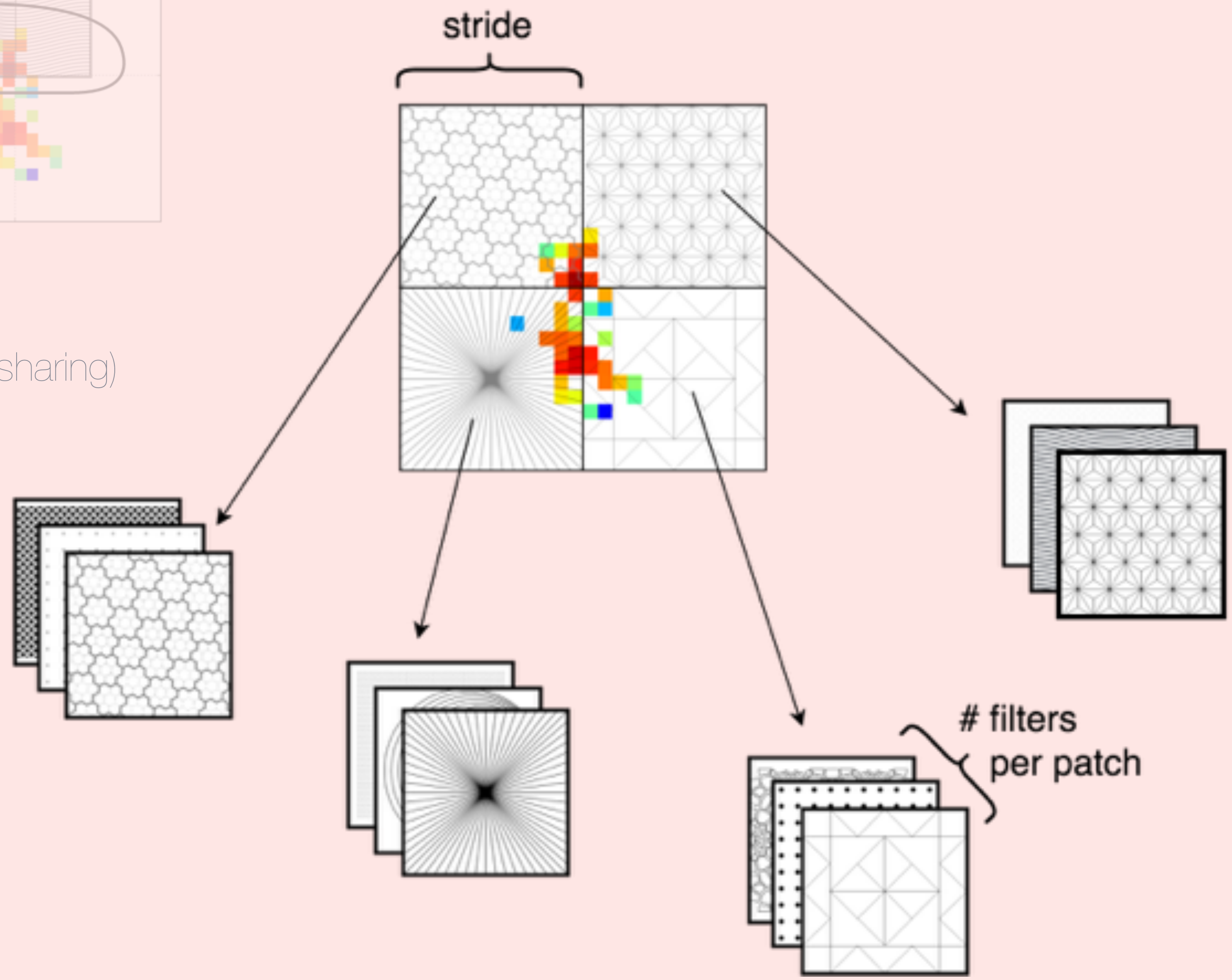


Convolutional Layer (weight sharing)

Layer Comparison



Convolutional Layer (weight sharing)

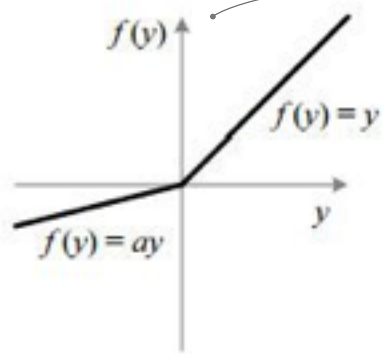


Locally-Connected Layer

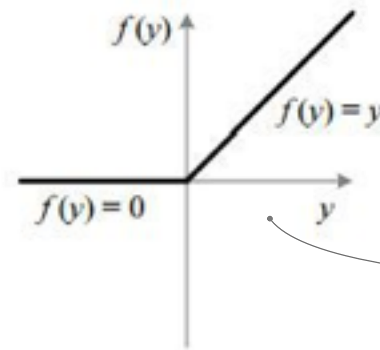
Architecture Guidelines

recommended in GAN literature

- Activation:



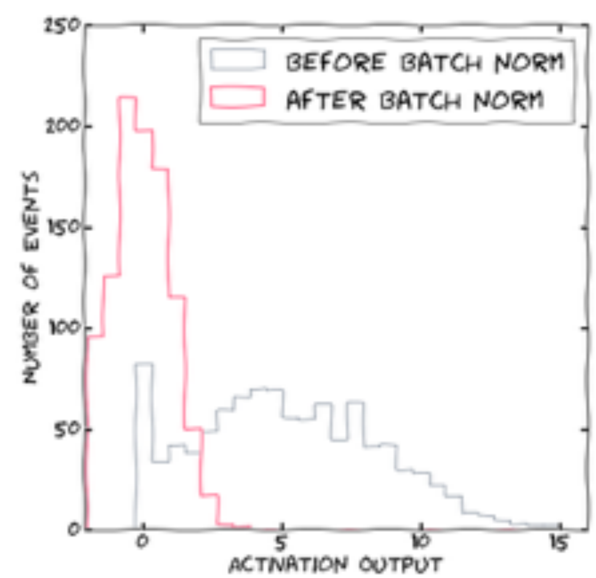
Leaky ReLU



ReLU

to induce sparsity

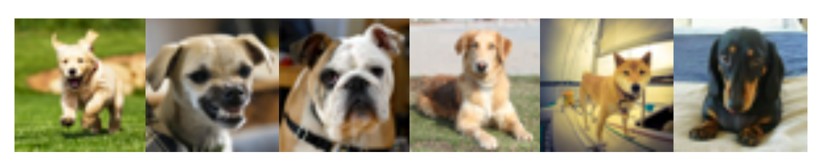
- Batch Normalization:



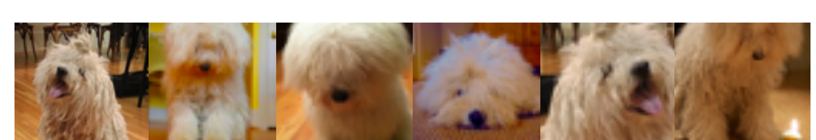
helps with high dynamic range

- Mini-batch Discrimination:

helps reduce mode collapse



→ High Entropy



→ Low Entropy

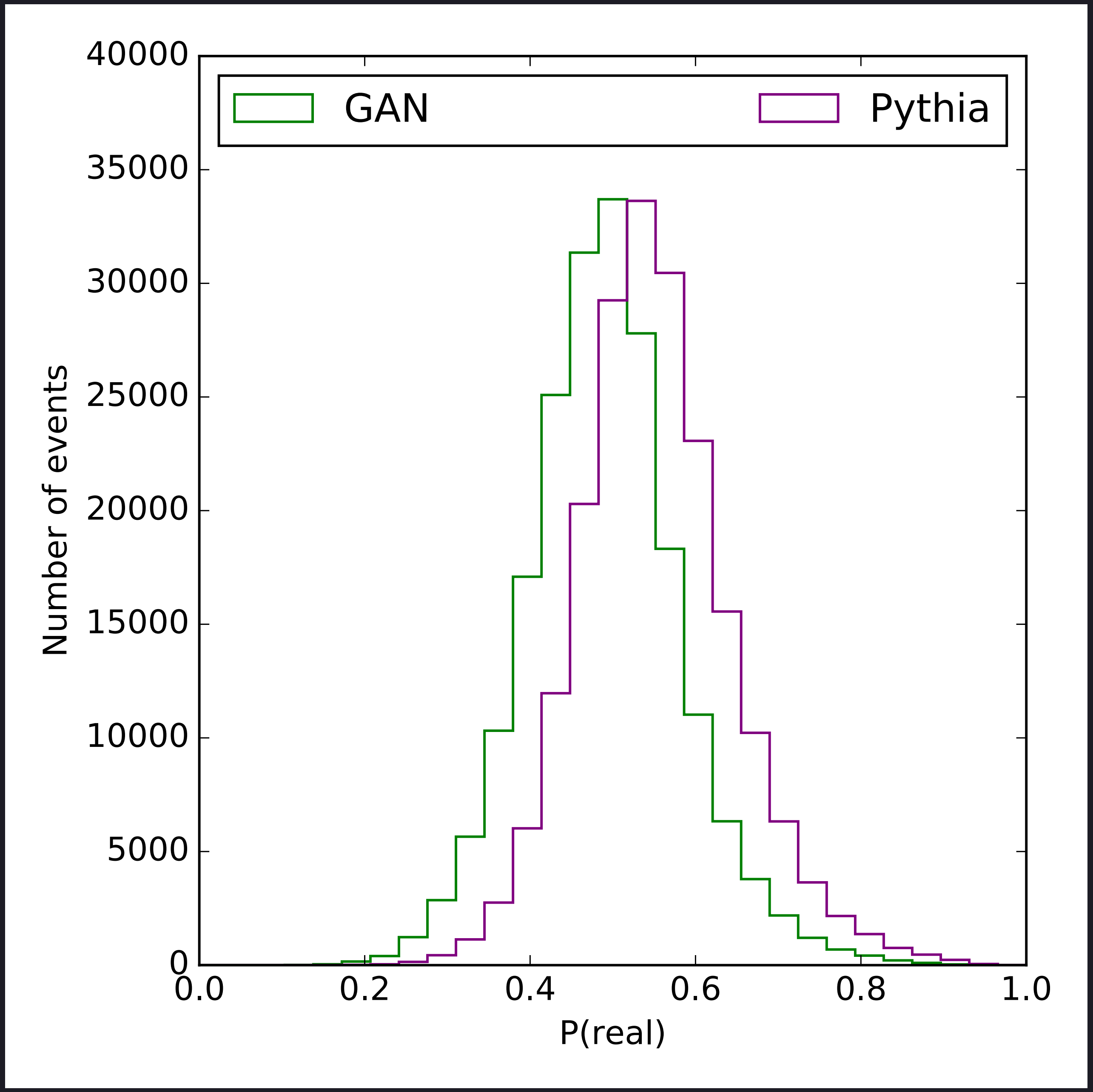
Keras Generator

```
def generator(latent_size, return_intermediate=False):

    # this is the z space commonly referred to in GAN papers
    latent = Input(shape=(latent_size, ))

    # this will be our label
    image_class = Input(shape=(1, ), dtype='int32')
    emb = Flatten()(Embedding(2, latent_size, input_length=1,
                              init='glorot_normal')(image_class))
    # hadamard product between z-space and a class conditional embedding
    h = merge([latent, emb], mode='mul')

    loc = Sequential([
        # DCGAN-style project & reshape,
        Dense(128 * 7 * 7, input_dim=latent_size),
        Reshape((7, 7, 128)),

        # block 1: (None, 7, 7, 128) => (None, 14, 14, 64),
        Conv2D(64, 5, 5, border_mode='same', init='he_uniform'),
        LeakyReLU(),
        BatchNormalization(),
        UpSampling2D(size=(2, 2)),

        # block 2: (None, 14, 14, 64) => (None, 28, 28, 6),
        ZeroPadding2D((2, 2)),
        LocallyConnected2D(6, 5, 5, init='he_uniform'),
        LeakyReLU(),
        BatchNormalization(),
        UpSampling2D(size=(2, 2)),

        # block 3: (None, 28, 28, 6) => (None, 25, 25, 1),
        LocallyConnected2D(6, 3, 3, init='he_uniform'),
        LeakyReLU(),
        LocallyConnected2D(1, 2, 2, bias=False, init='glorot_normal'),
        Activation('relu')
    ])

    fake_image = loc(h)

    return Model(input=[latent, image_class], output=fake_image)
```

Performance Evaluation

Discriminator Output

Output of the discriminator (primary task) at the epoch that was chosen to analyze the LAGAN performance in the paper:



- Ideal scenario: *D* outputs 1/2 for all samples, regardless of their origin

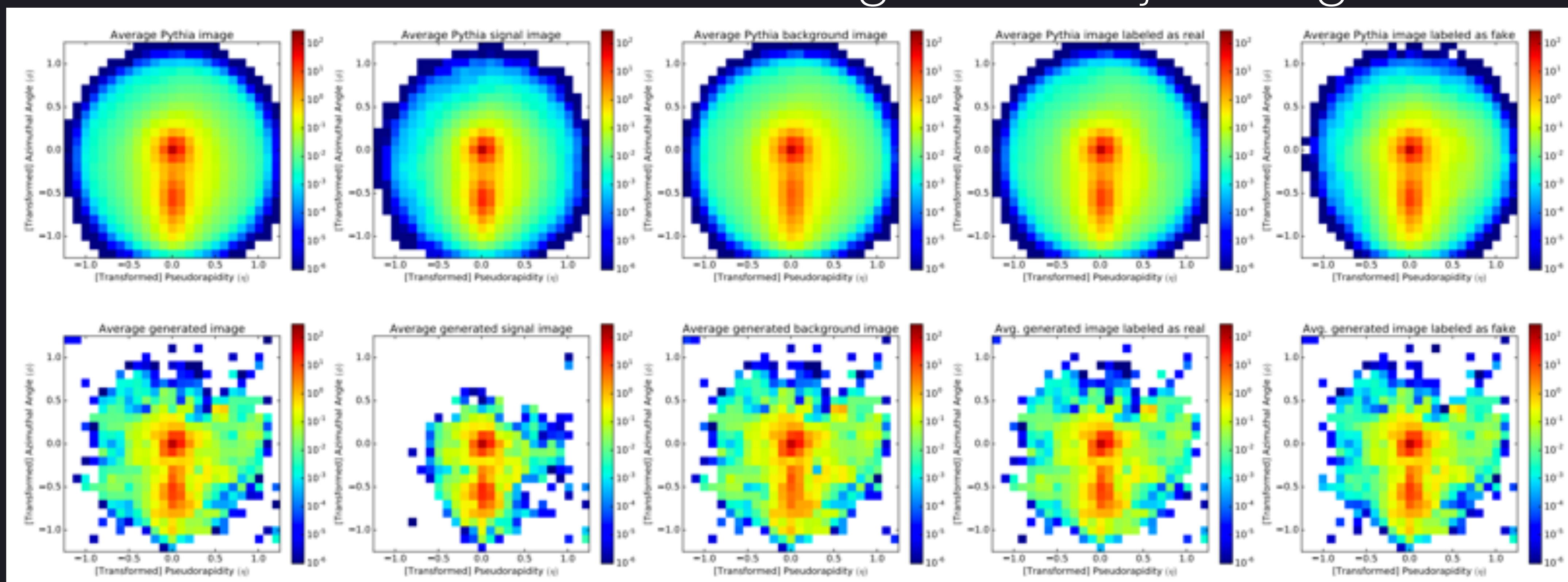
- For a well trained *D*, this corresponds to *G* producing realistic enough samples, so that *D* can't tell them apart from the real ones

Qualitative: Pythia vs LAGAN images

Qualitative assessment of generated jet images:

avg Pythia

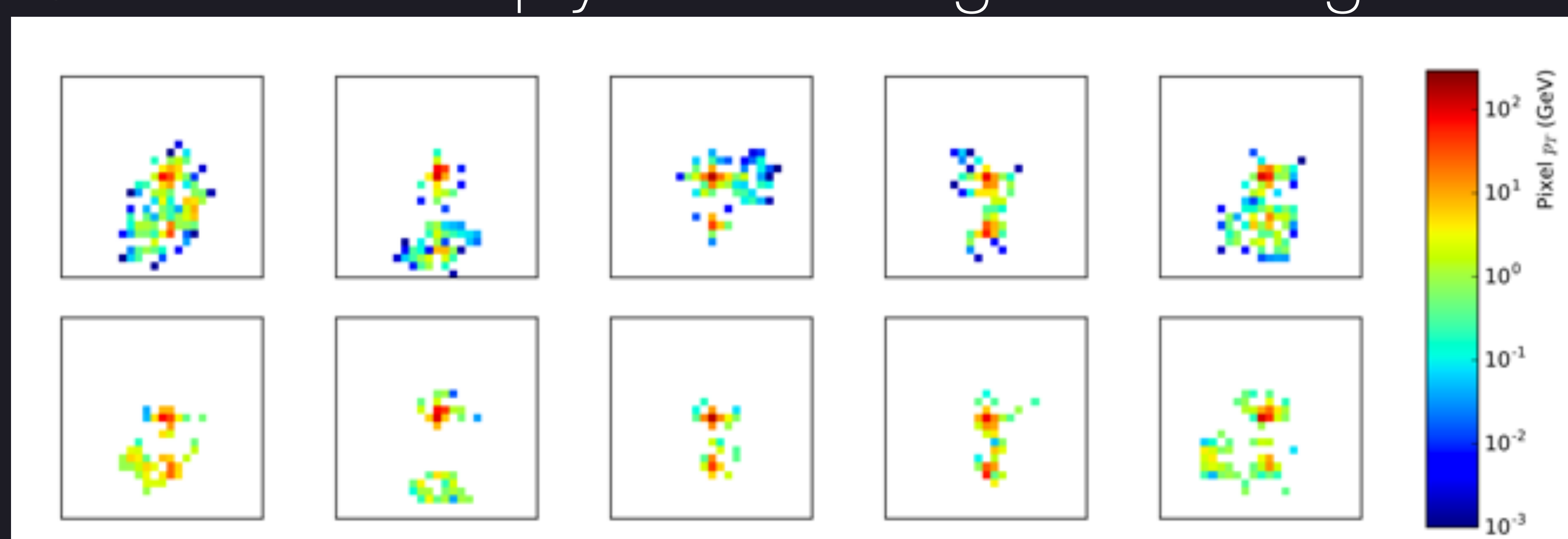
avg LAGAN



The LAGAN is not simply memorizing the training set

5 random Pythia images

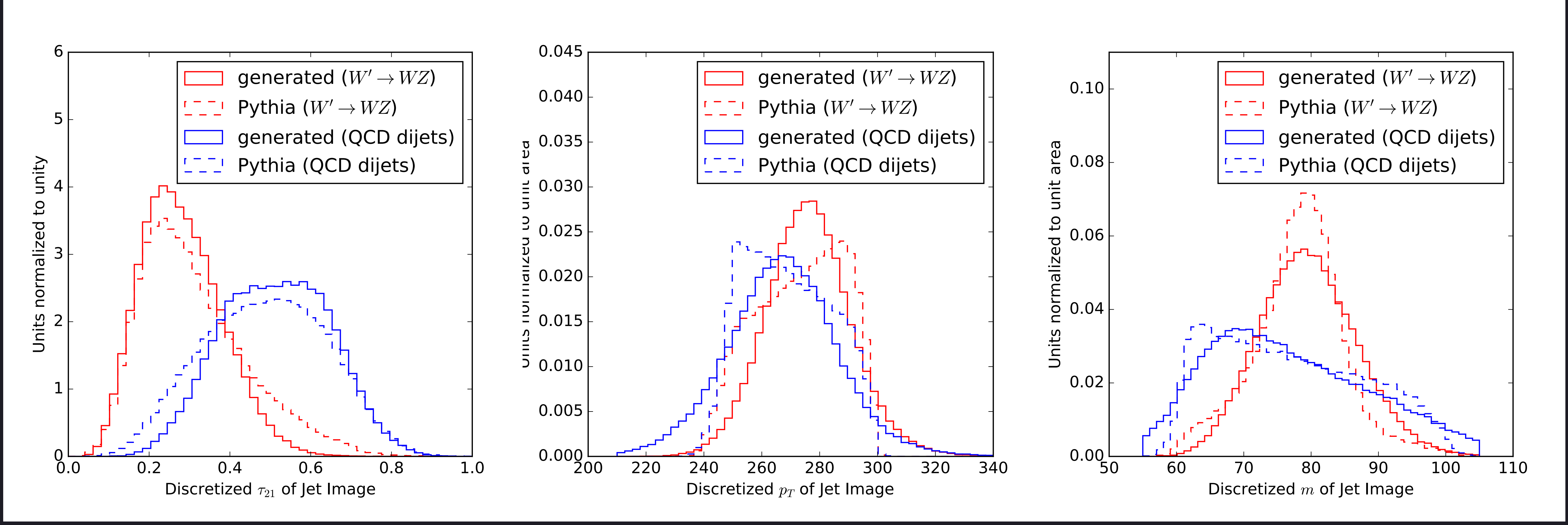
their nearest neighbor in the LAGAN generated dataset



Physical Distributions

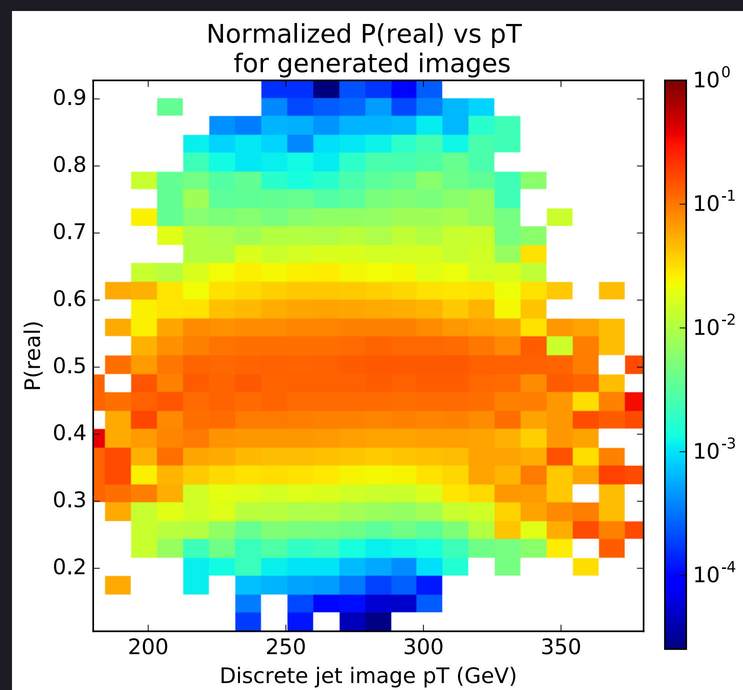
Check: does the LAGAN recover the true data distribution as projected onto a set of meaningful 1D manifolds? ✓

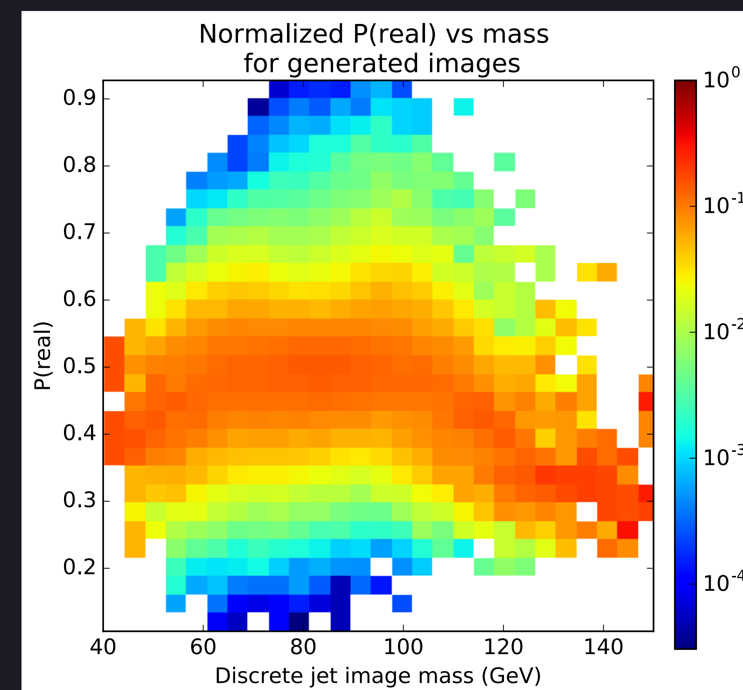
— signal
— background



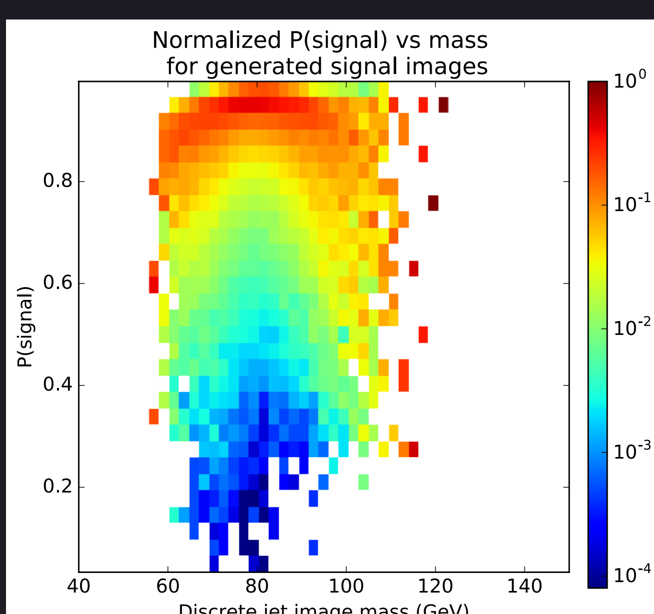
Discriminator output versus mass and pT

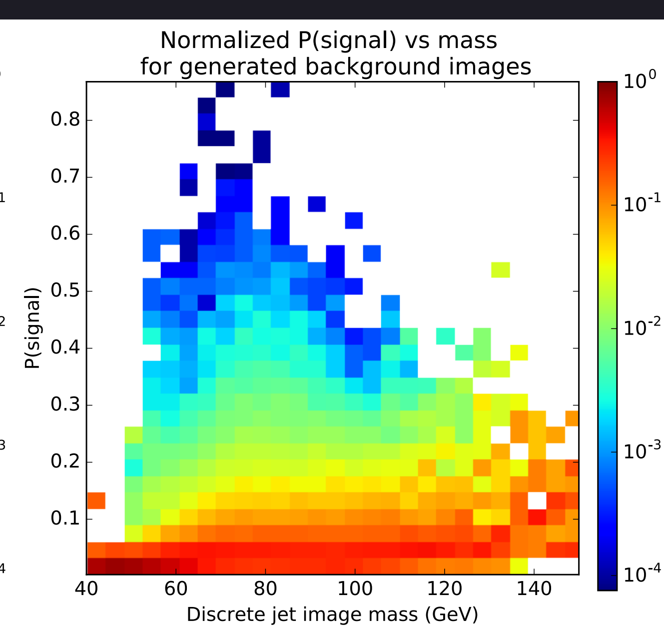
Primary classifier's performance is quite stable over whole range of m, pT

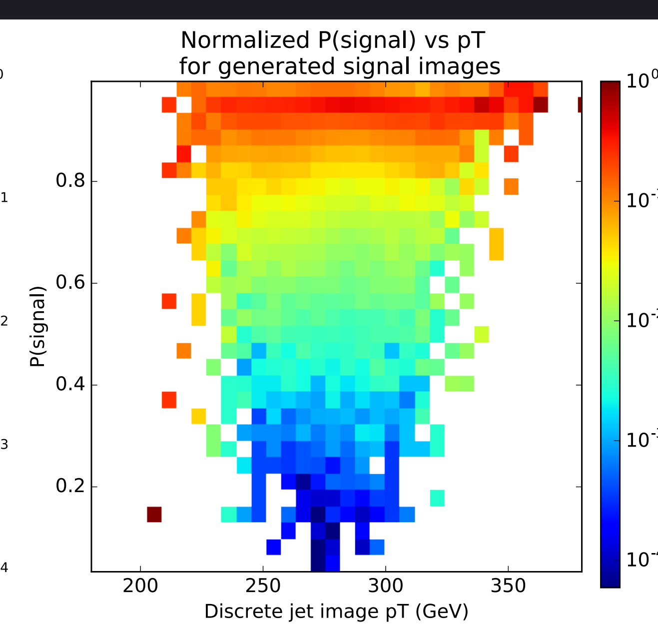


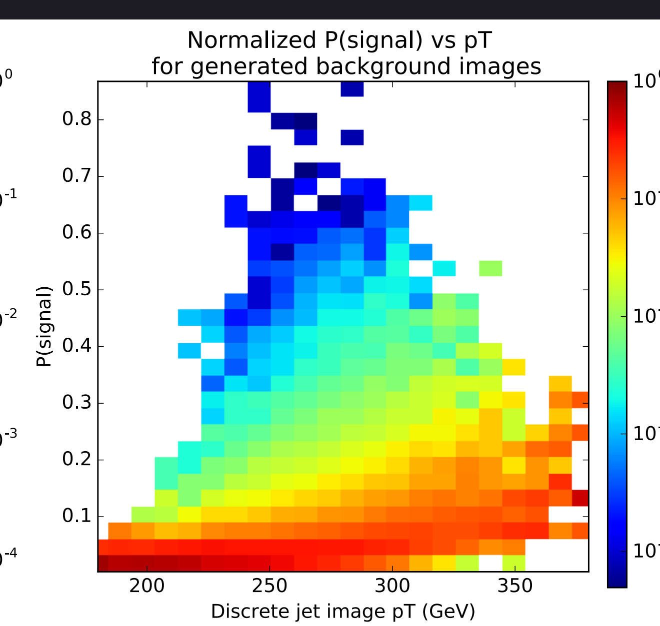


Excellent performance for aux task. Small m and pT dependent features









Performance

Distribution of any number of meaningful manifolds

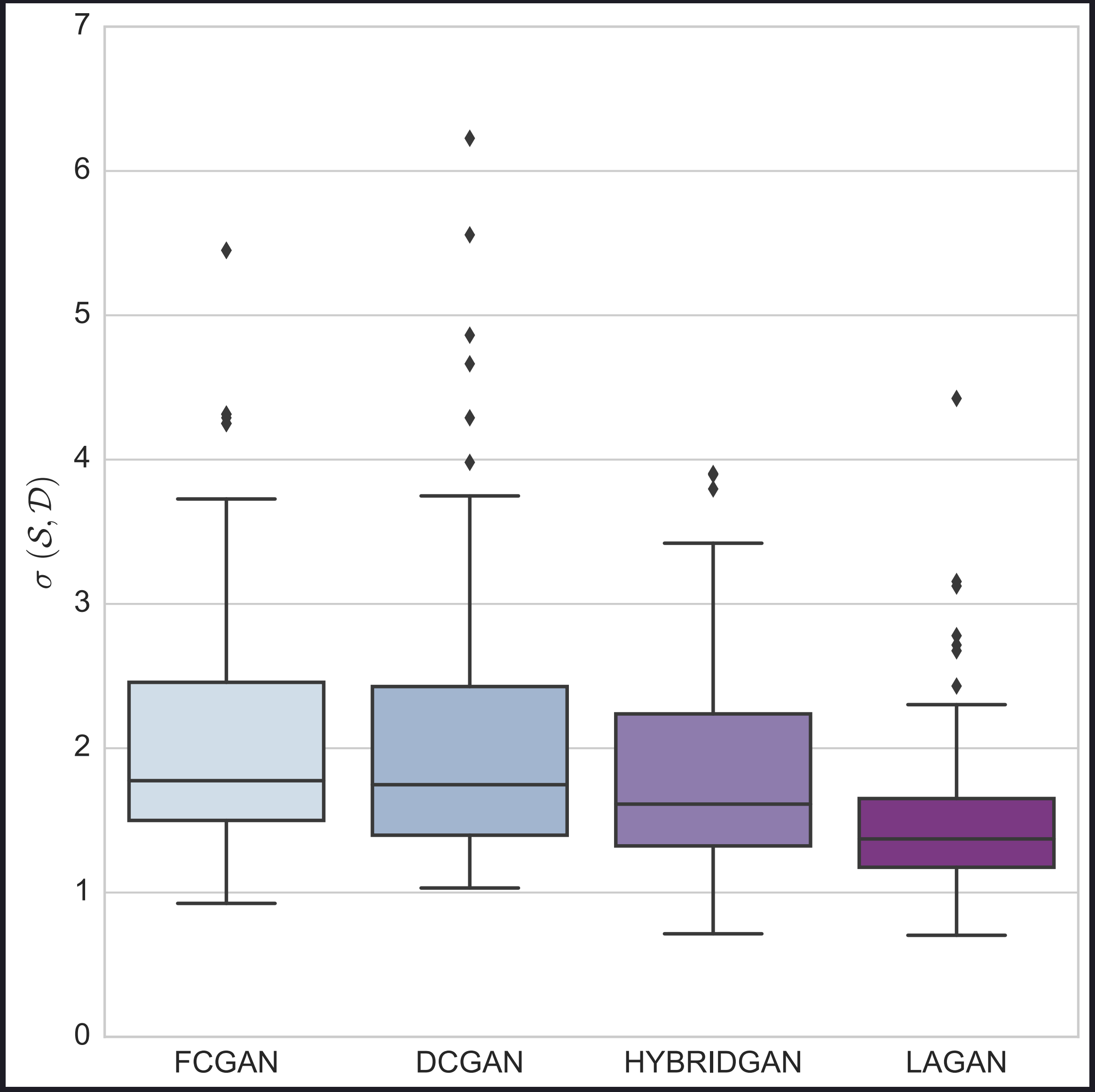
$$\sigma(\mathcal{S}, \mathcal{D}) = \max_{c \in \mathcal{C}} d(\mathcal{M}_{\mathcal{D}}(x|c), \mathcal{M}_{\mathcal{S}}(x|c))$$

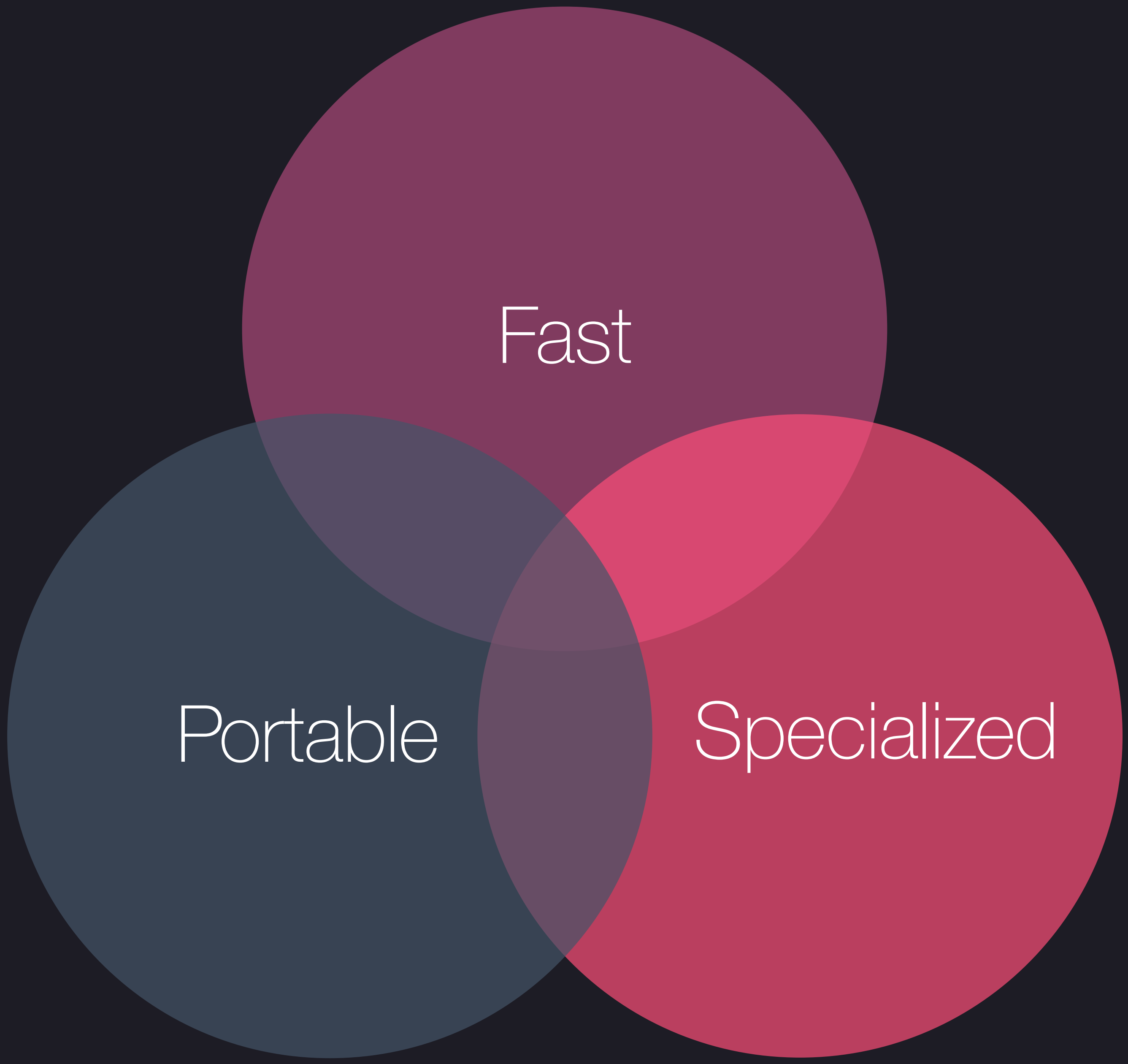
Distribution of generated images

True data distribution of jet images

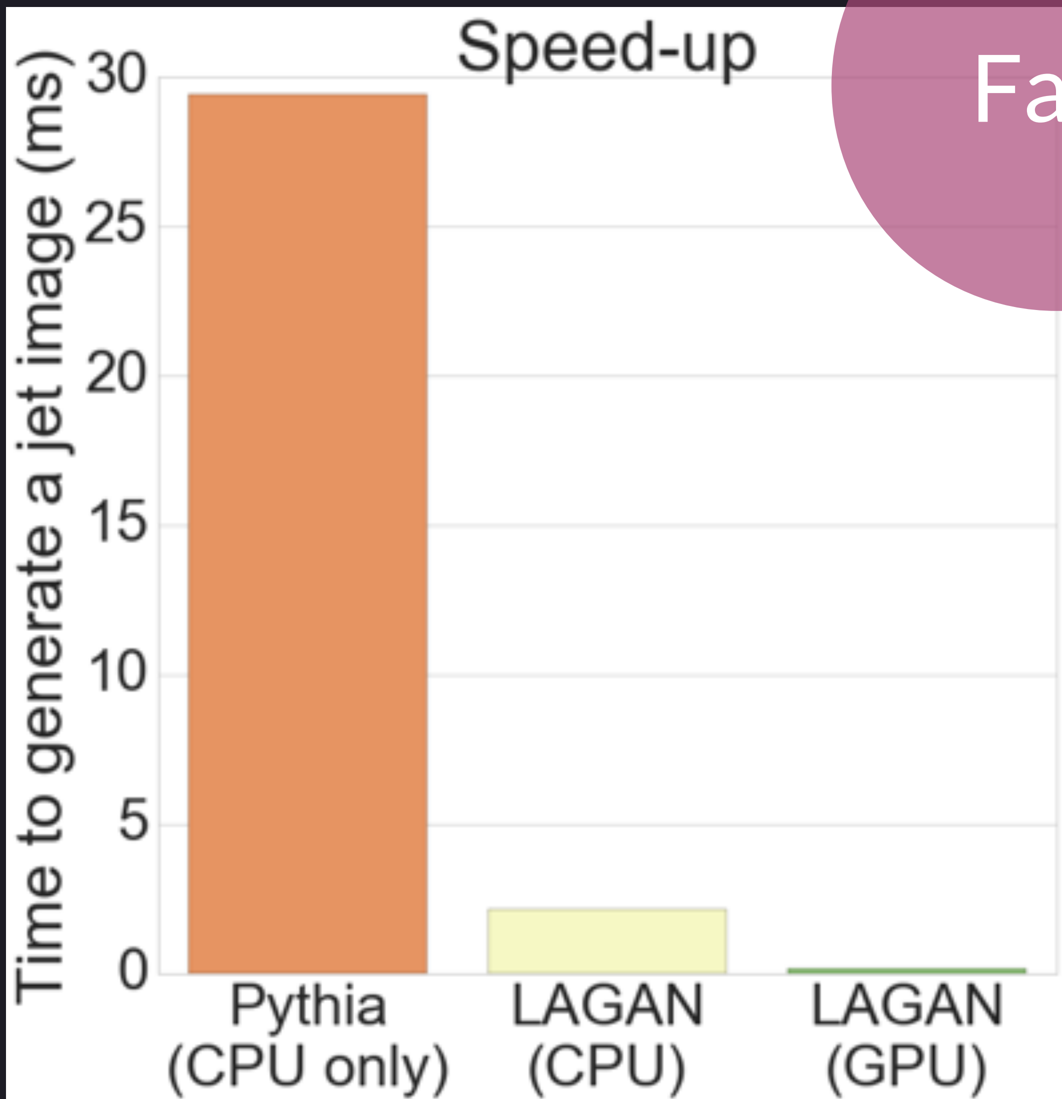
class

Performance comparison among architectures

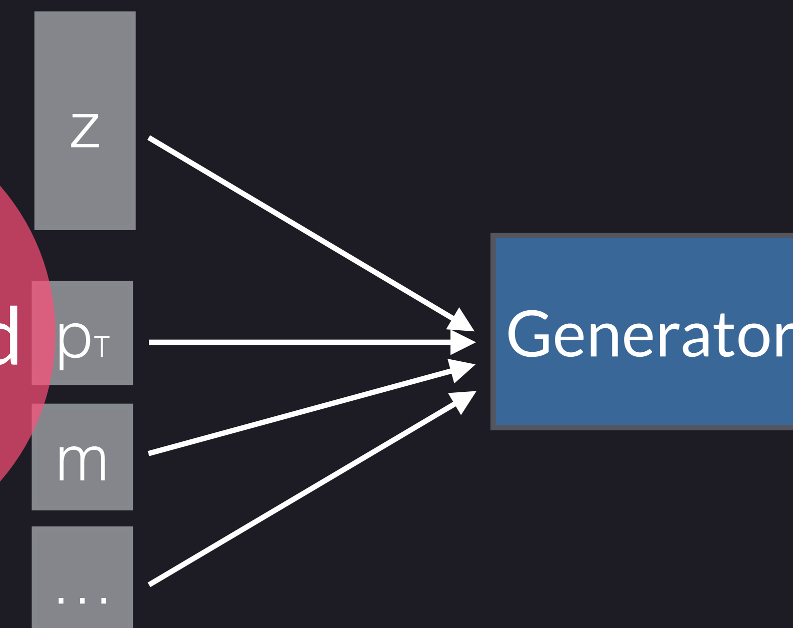



Fast
Portable
Specialized

Fast



Specialized



Portable

Trained LAGAN weights:

DOI 10.5281/zenodo.400706

20 MB

Source code:

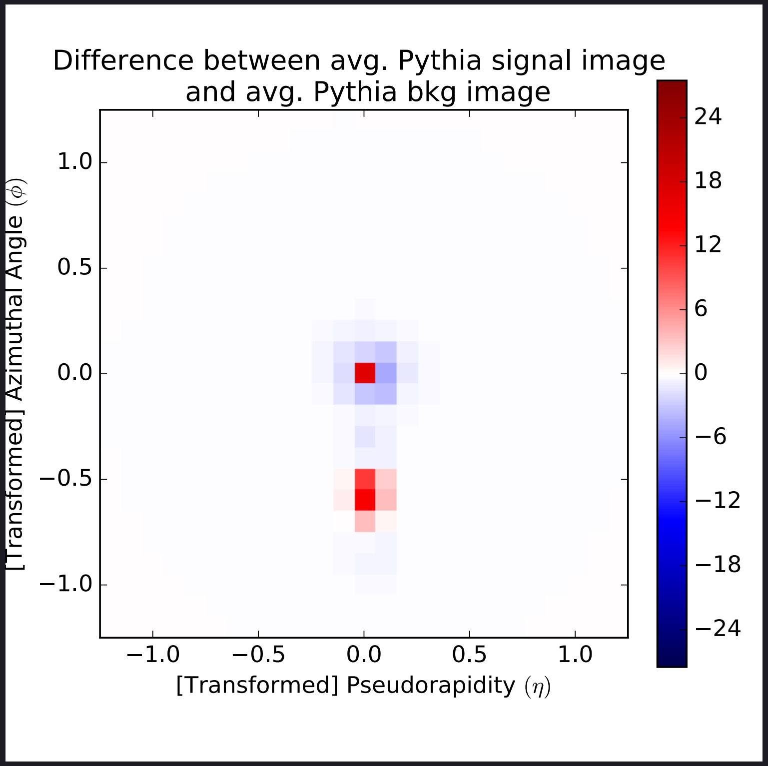
DOI 10.5281/zenodo.400708

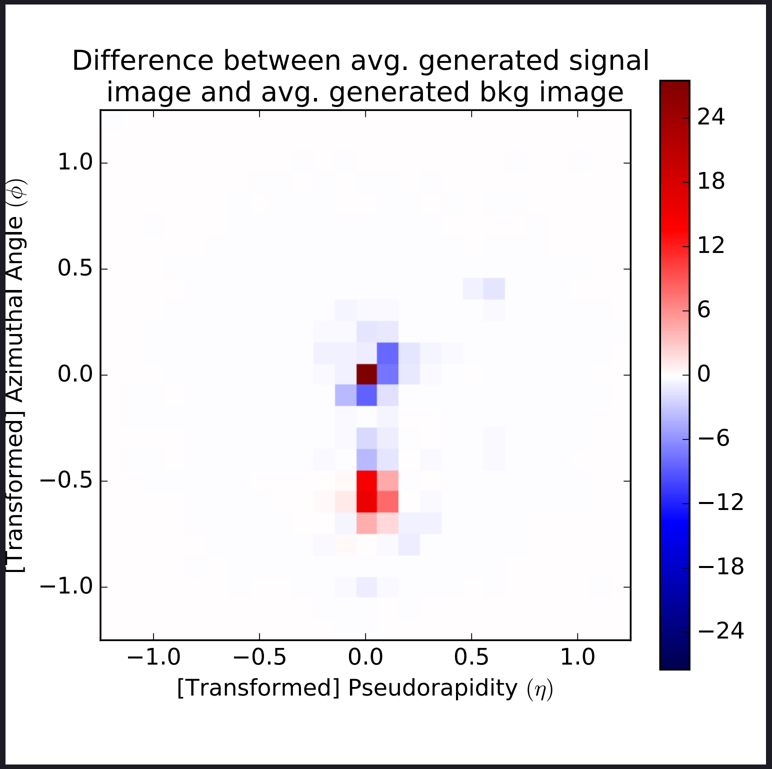
480 kB

Peeking through the Generator

Location of physical features to distinguish W bosons from QCD

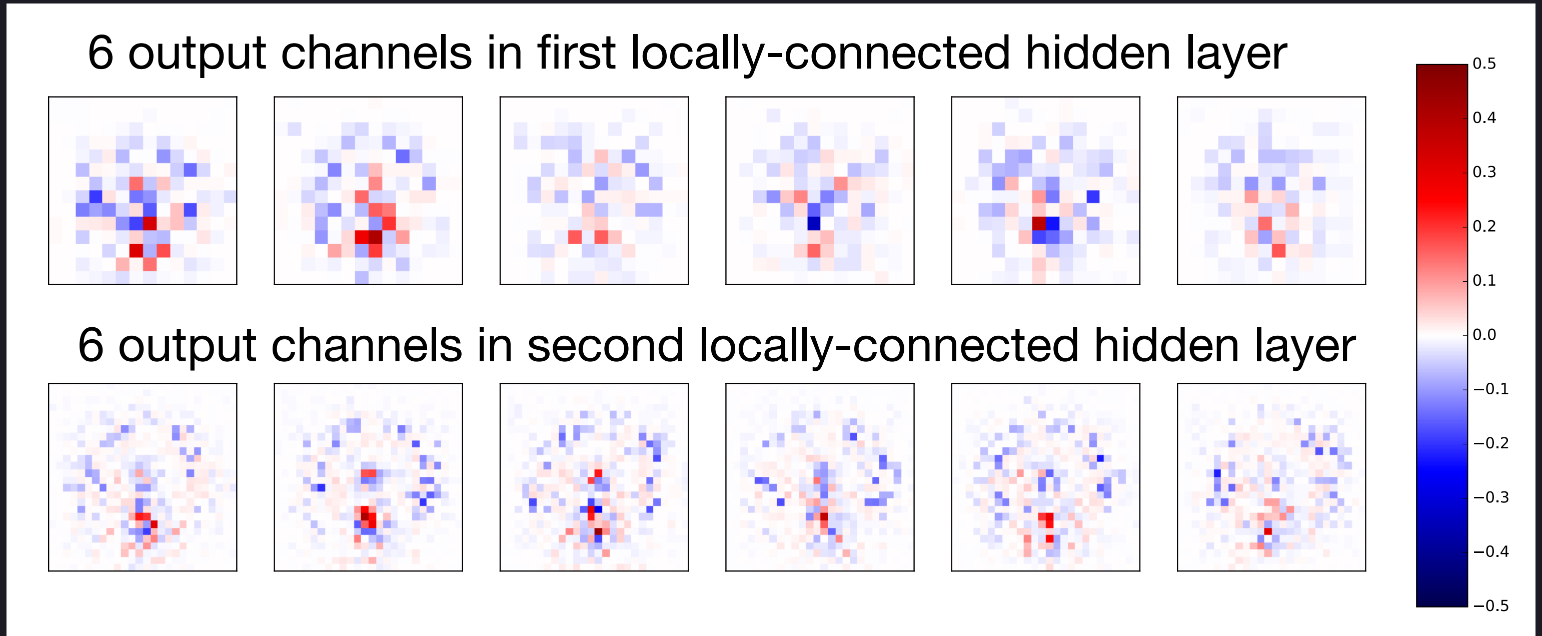
in Pythia





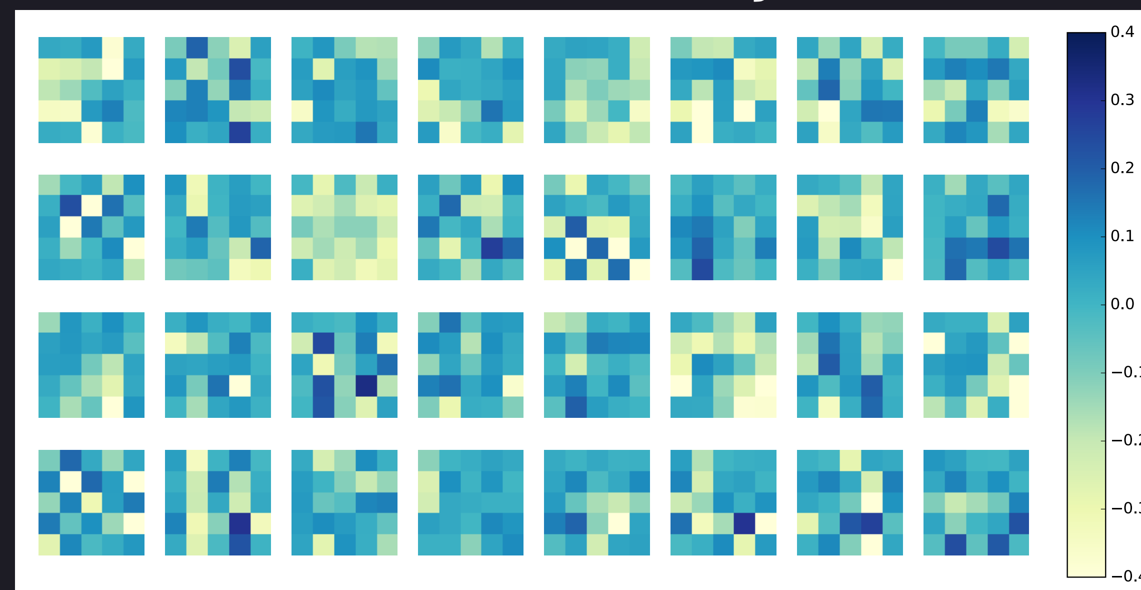
in LAGAN

How early do these features develop within the layers of the generator?

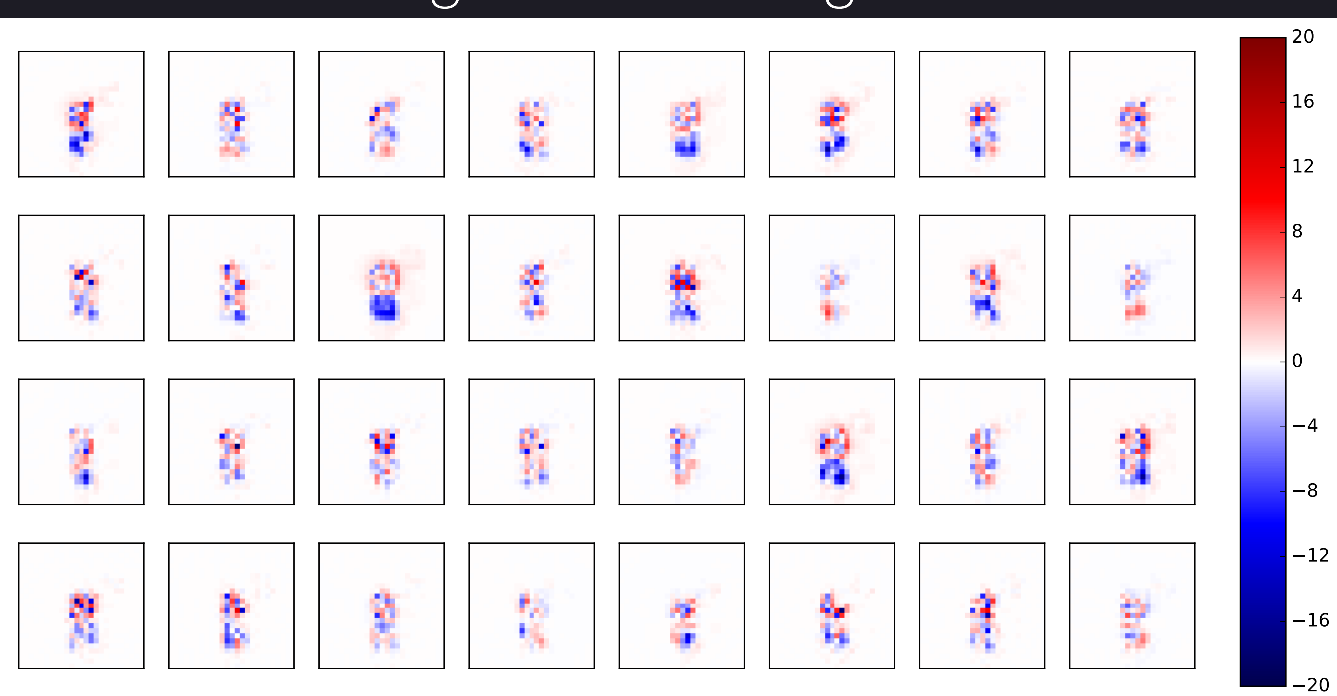


Peeking through the Discriminator

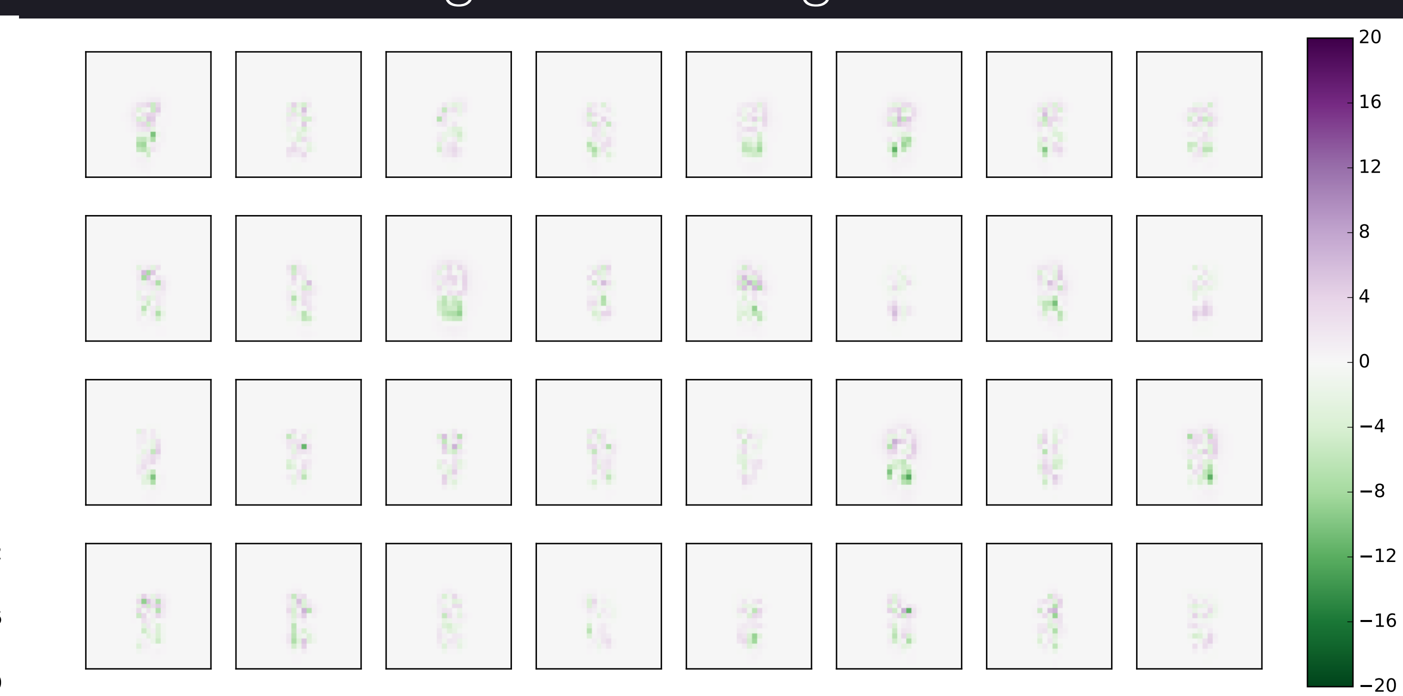
Convolutional filters in the first layer of the discriminator



their convolved version with the difference between the average signal and background generated image

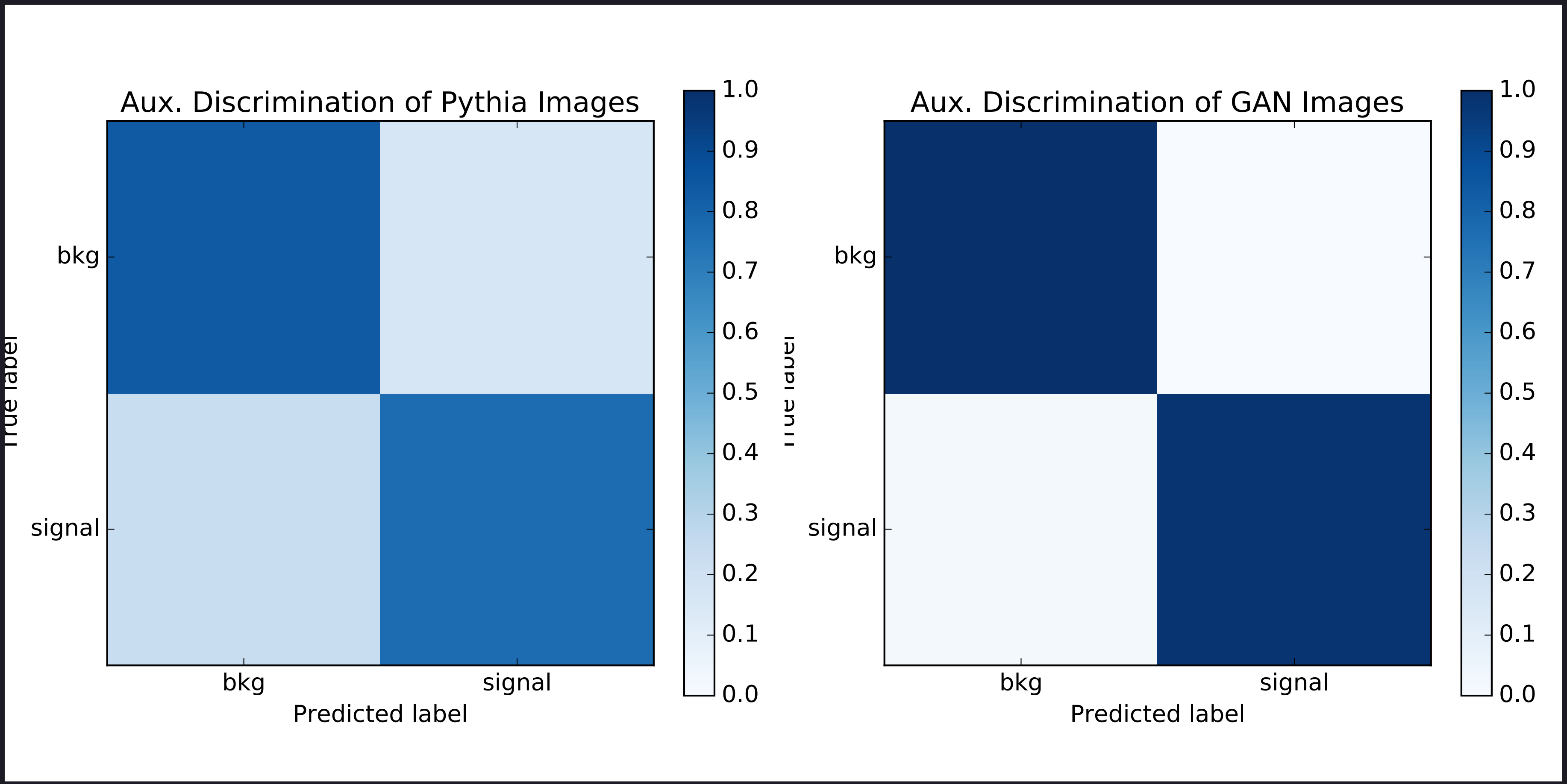


their convolved version with the difference between average Pythia and average generate image



GAN images for classification

- GAN produces very signal or very background looking images
- Unlikely to explore gray area between labels because it's unfavorable under loss formulation
- Label flipping should help addressing this issue — maybe not aggressive enough
- Output of aux task clearly shows this issue
- GAN images still useful for data augmentation



Conclusion & Outlook

- Resized problem of generating simulation to first address the simplified 2D case, using jet images
- GANs — validated as a tool for scientific simulation
- Interdisciplinary collaboration needed
  - out-of-the-box, vanilla solutions are not enough
- LAGAN — optimized for sparse, unbounded, and highly non-linearly location-dependent data distributions

Reproduce This!

- This paper on the arXiv [1701.05927]
- Github repository: centralized location with all links
- Download the training dataset of Pythia jet images, or generate them yourself using this Docker image
- Train a DCGAN, fully-connected GAN, hybrid GAN, or LAGAN using the code we provide in `models`, or load our pre-trained weights
- Use the jupyter notebook in `analysis` to make plots like ours