Accelerating Simulation with GANs

arXiv:1701.05927, arXiv:1705.02355, arXiv:1712.10321, arXiv:1711.08813

Luke de Oliveira
Vai Technologies & Lawrence Berkeley National Lab

Joint work with:

Ben Nachman
Lawrence Berkeley National Lab

Michela Paganini
Yale University & Lawrence Berkeley National Lab

@lukede0
lukedeo@vaitech.io
https://ldo.io

Outline

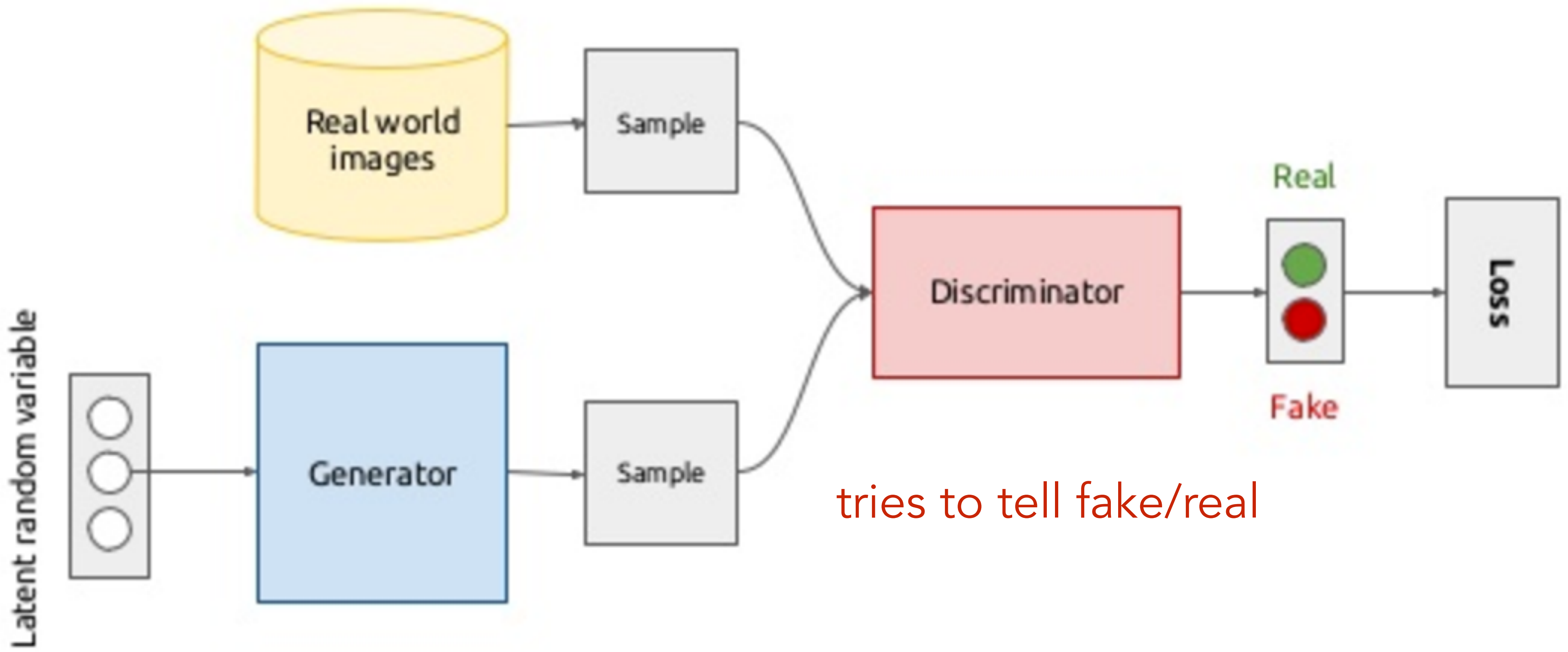
- Introduction to Generative Adversarial Networks
- Learning to produce calorimeter showers
- Concluding remarks

Adversarial Deep Learning

- What is an adversary? Best to think about context of games
- As we design our "game," we have a goal in mind for it's outcome
- We can pit multiple ***players*** (usually neural networks) against each other with cost functions to minimize/ maximize while playing, usually want to characterize emergent behavior / convergence of such a game

Generative Adversarial Networks (GAN)

Turn generative modeling into a two player, non-cooperative game.



tries to tell fake/real

tries to produce real looking samples

Setting up the Classical GAN system

$$V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x;\theta_D)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(x;\theta_G);\theta_D))]$$

- First term, probability that a real sample is classified as real

- Second term, probability that a fake sample is classified as fake

- Generator wants to minimize this, discriminator wants to maximize this

GANs in HEP

- Ultimate goal: quickly and accurately simulate particles interacting with individual detector components - speed up slow simulation

- Intermediary goal: can we speed up calorimeter simulation, which is the current bottleneck?

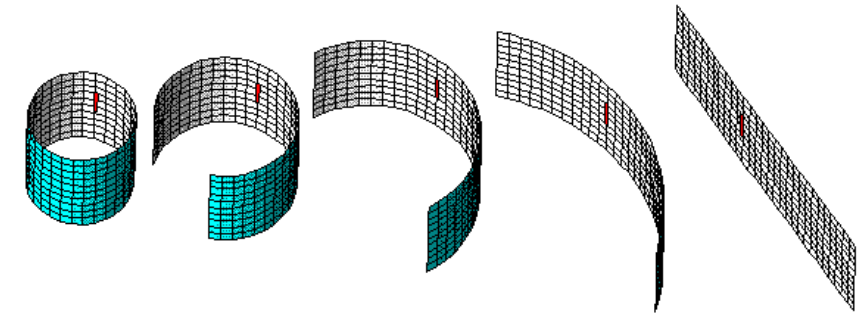
- First step: can we learn to generate jet images using a Generative Adversarial Network?

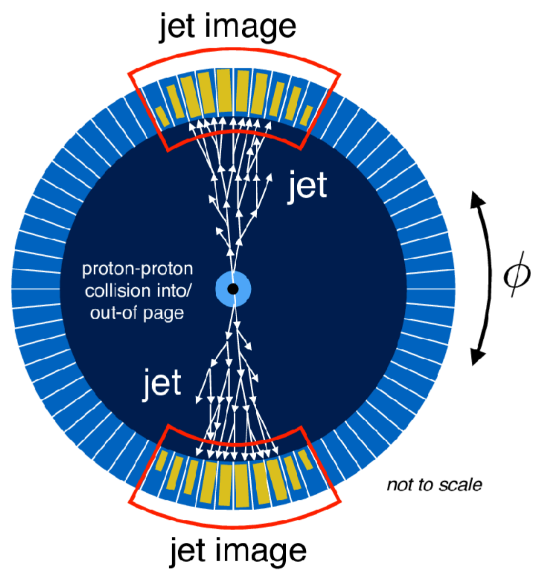
LAGAN

arXiv:1701.05927

Jet Images: Particle Physics → Computer Vision

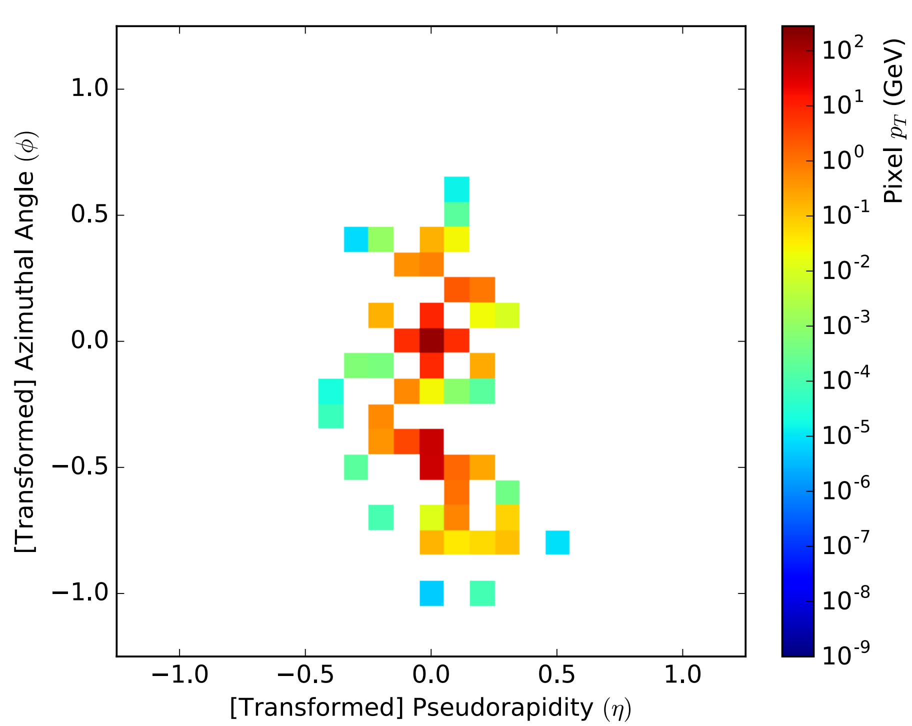
- Now quite ubiquitous: unroll a detector to treat depositions like an image



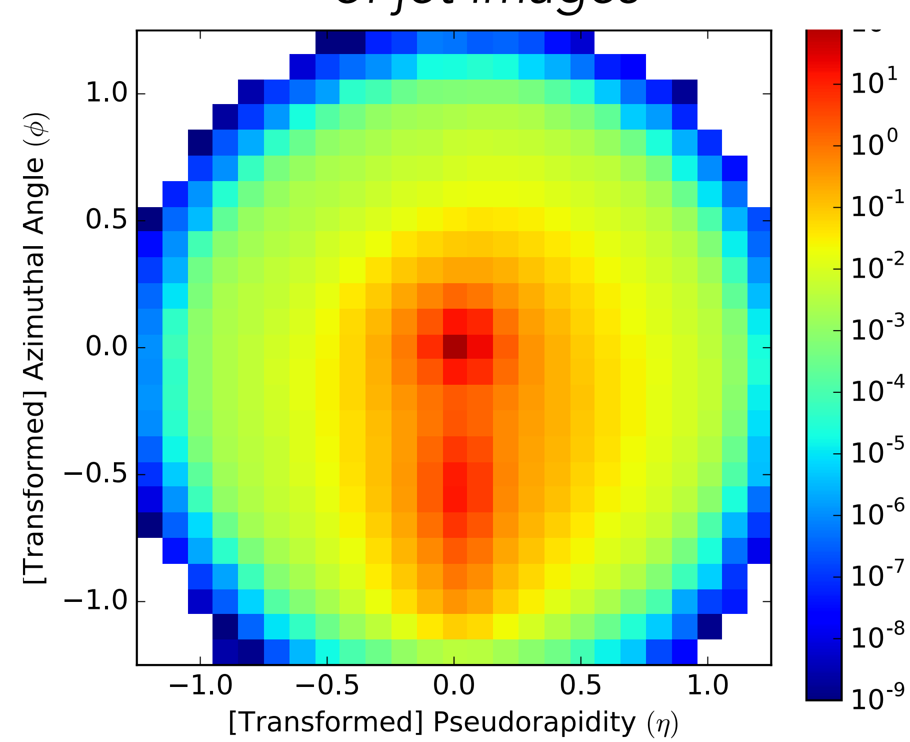


- For a more detailed explanation and related links, see Benjamin Nachman's talk from ACAT 2017
- Jet image: 2D representation of the radiation pattern within a jet

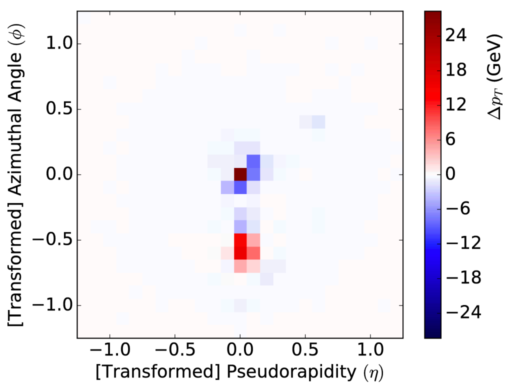
A jet image



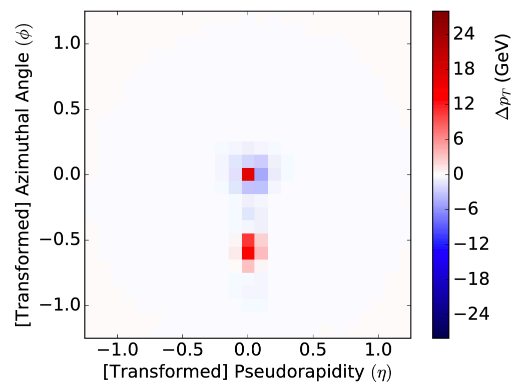
Average over thousands of jet images



Qualitative Assessment

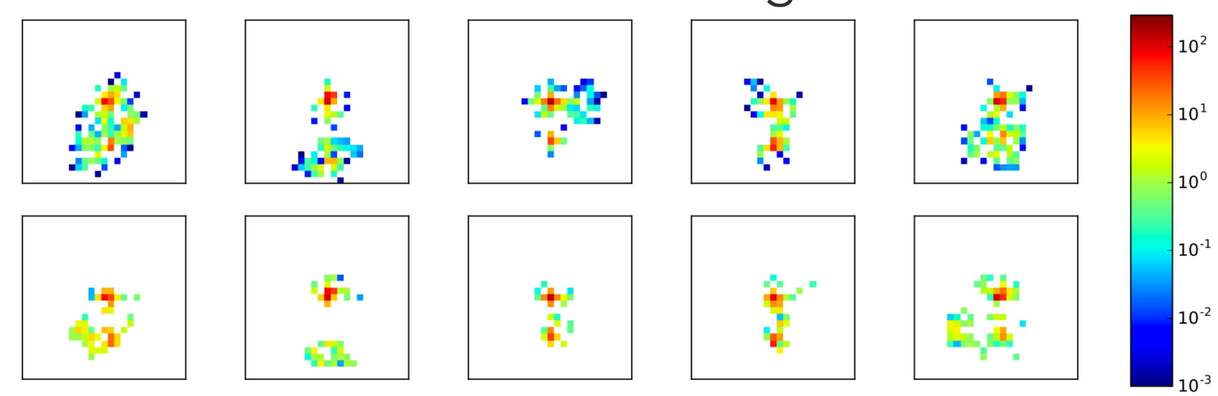


GAN-generated
signal - background

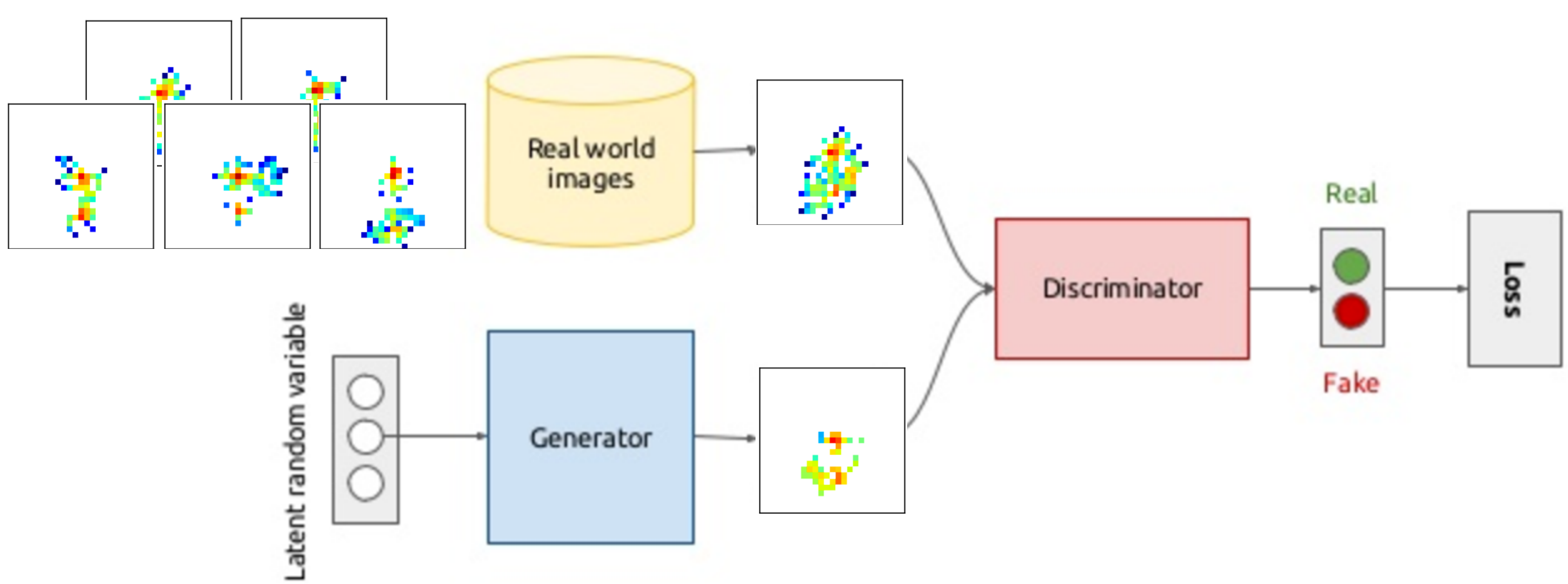


Real signal - background

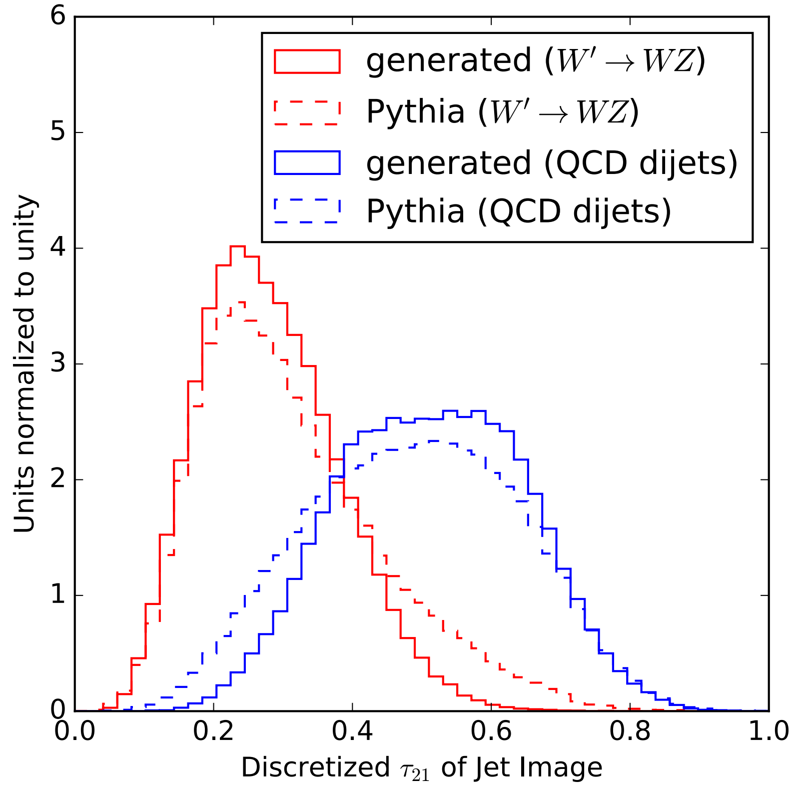
5 random Jet Images

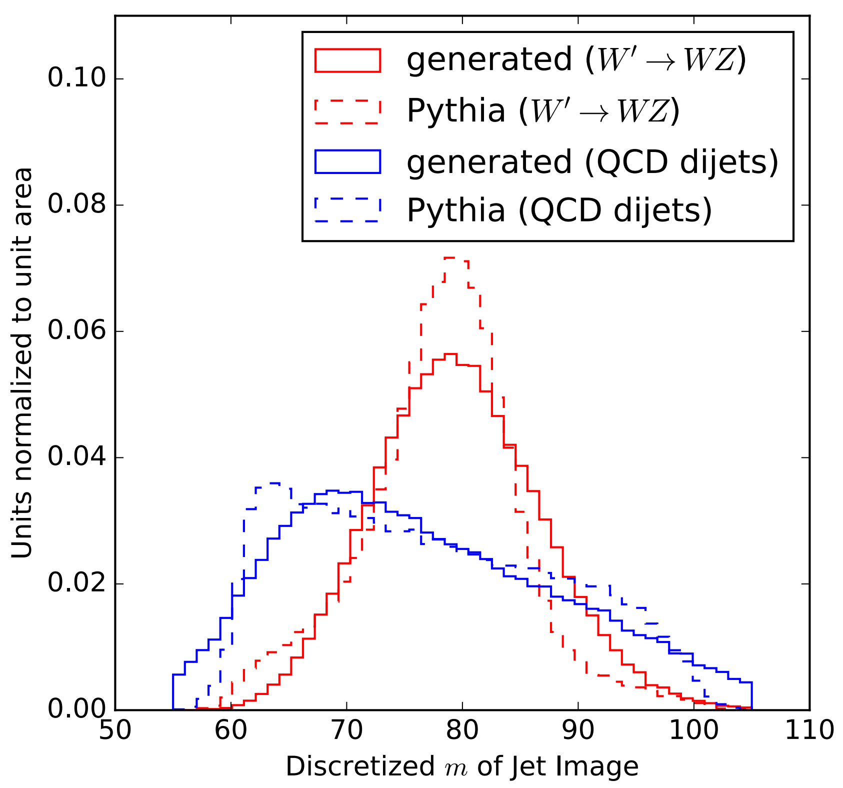


nearest LAGAN-generated
neighbor



Checking Physical Properties





$$p_{\mathrm{T}}^2(I) = \left(\sum_{i=0}^{N} I_i \cos(\phi_i)\right)^2 + \left(\sum_{i=0}^{N} I_i \sin(\phi_i)\right)^2$$

$$m^2(I) = \left(\sum_{i=0}^{N} I_i\right)^2 - p_{\mathrm{T}}^2(I) - \left(\sum_{i=0}^{N} I_i \sinh(\eta_i)\right)^2$$

n-subjettiness

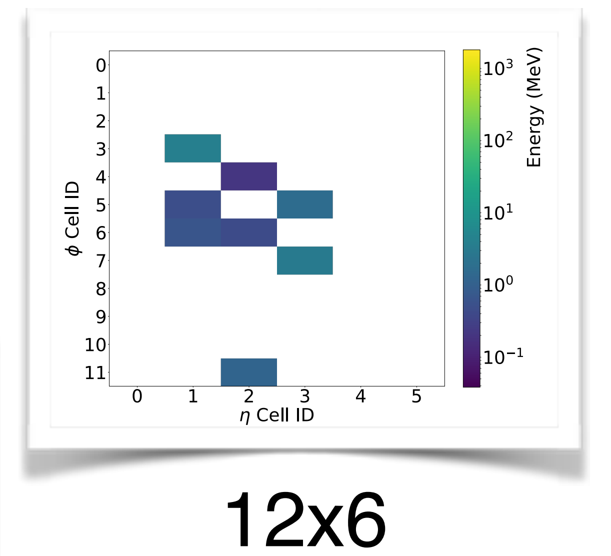
jet mass

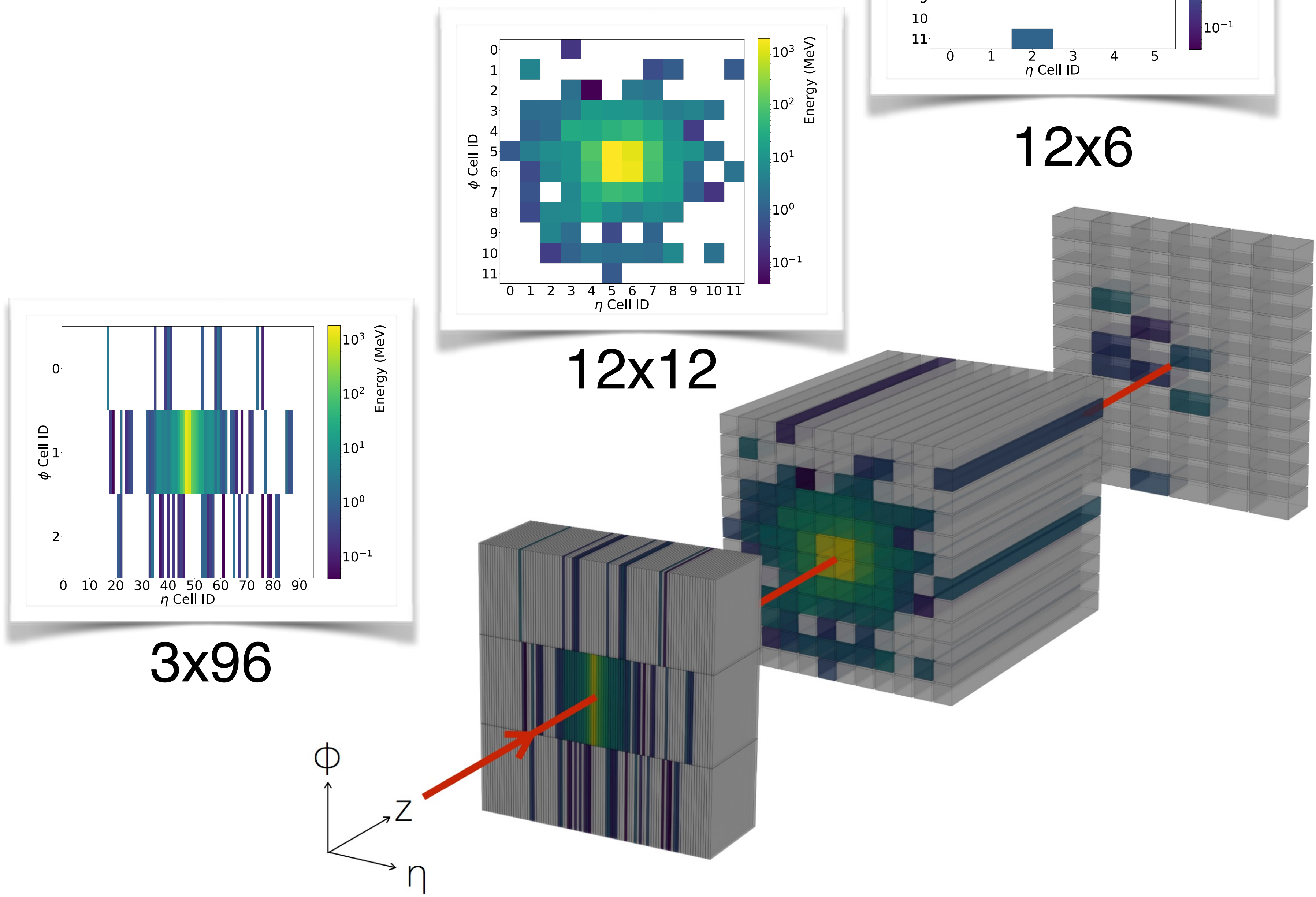
CaloGAN

arXiv:1705.02355, arXiv:1712.10321, arXiv:1711.08813

Calorimeter Images

- Closer and closer to raw detector output (e^+, π^+, y)
- Goal: obtain simulation speedups with GAN
  - *Challenges*:
    - spatio-temporal dependence
    - sparsity
    - dynamic range
    - location specificity
  - *Advantages*:
    - compositionality
    - quantifiable properties





arXiv:1705.02355

Open Dataset of Calo Images

- 3 layer, heterogeneous segmentation and resolution (designed to approximate ATLAS LAr calorimeter)

Layer	z segmentation [mm]	η segmentation [mm]	ϕ segmentation [mm]
0	90	5	160
1	347	40	40
2	43	80	40

- 3 types of particle: e^+, π^+, y

- Variable position and angle of incidence (5cm in x and y; 0°, 5° and 20° in theta and phi)

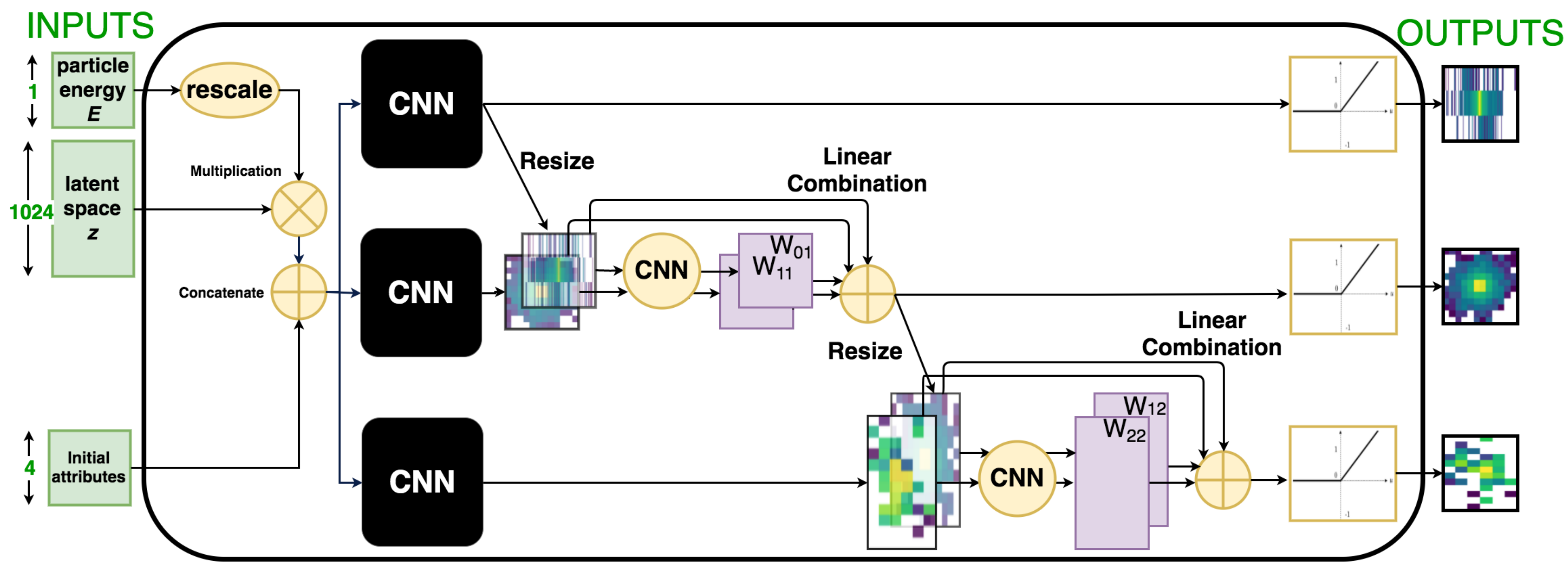
- Open, available, re-usable, citable

Model Requirements

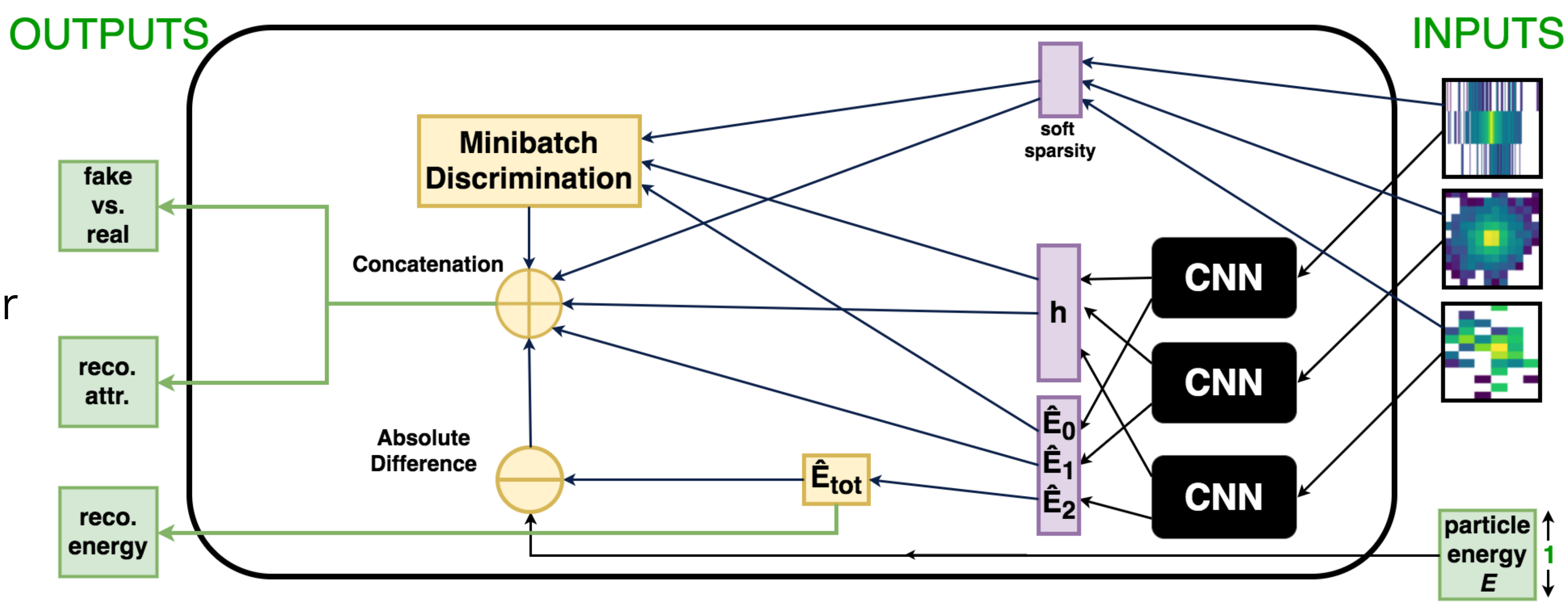
- Obey (to an approximation) conservation of energy and other physical traits such as sparsity
- Allow conditioning on physical attributes (incident angle, energy, etc.)
- Model sequential relationship through layers

CaloGAN Architecture

Generator



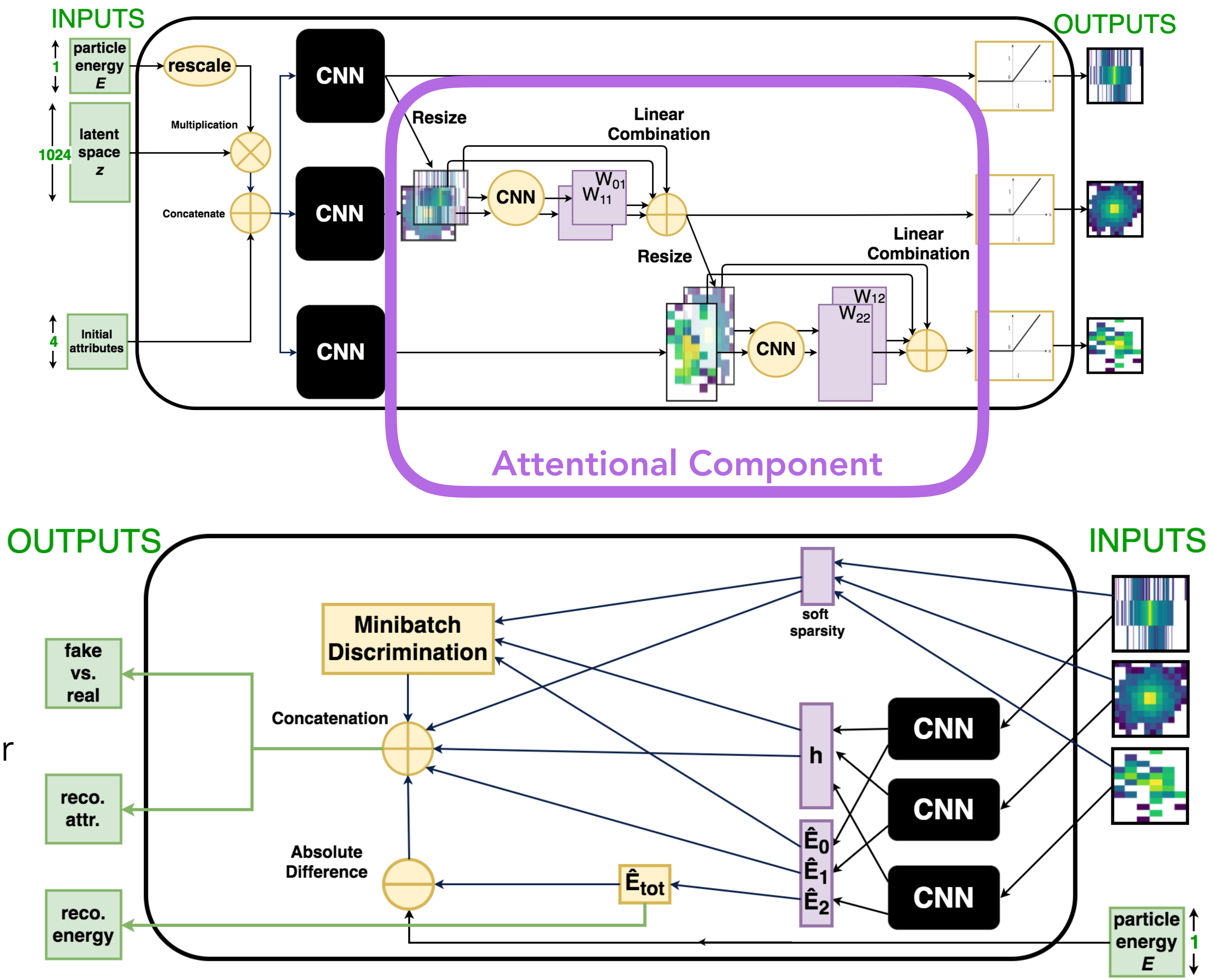
Discriminator



arXiv:1705.02355

CaloGAN Architecture

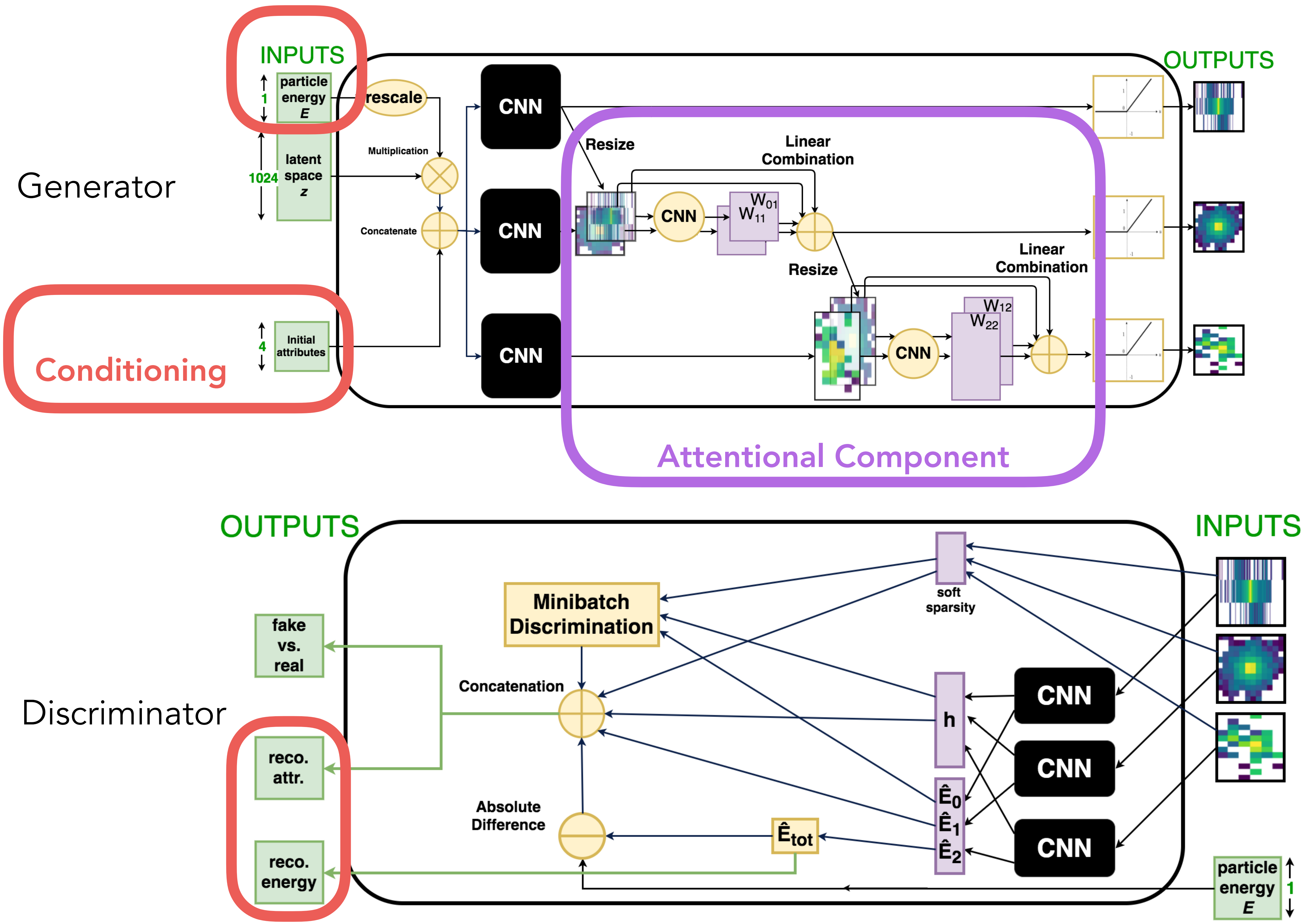
Generator



Discriminator

arXiv:1705.02355

CaloGAN Architecture



arXiv:1705.02355

Performance at Calo Level

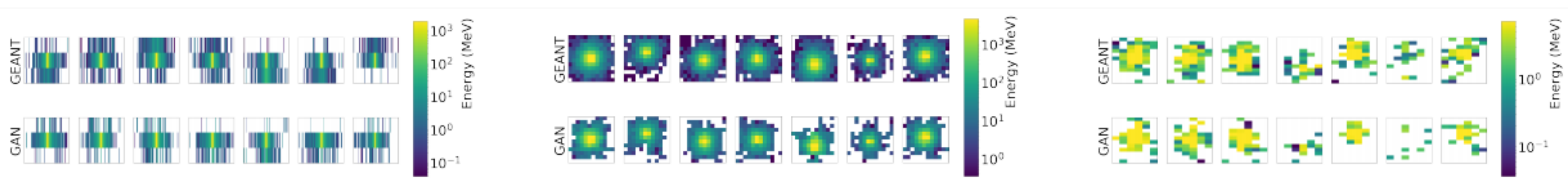


Figure 2. Nearest GAN-generated neighbors (bottom) for seven random GEANT4-generated e^+ showers (bottom) for the first layer (left), second layer (middle), and last layer (right) of the calorimeter.

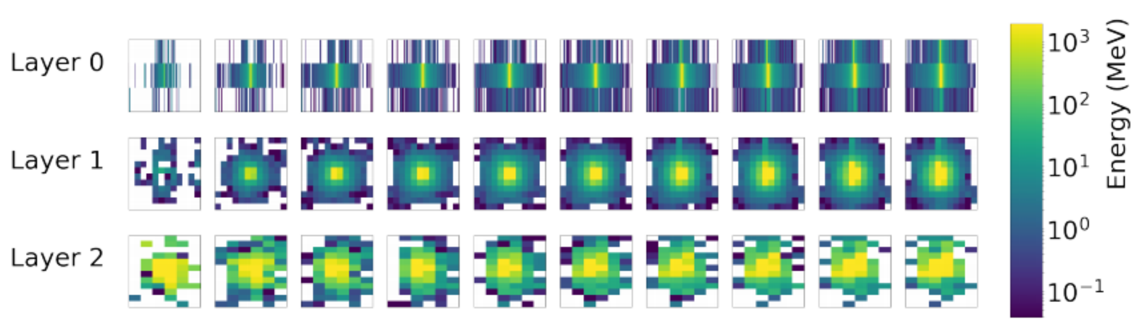
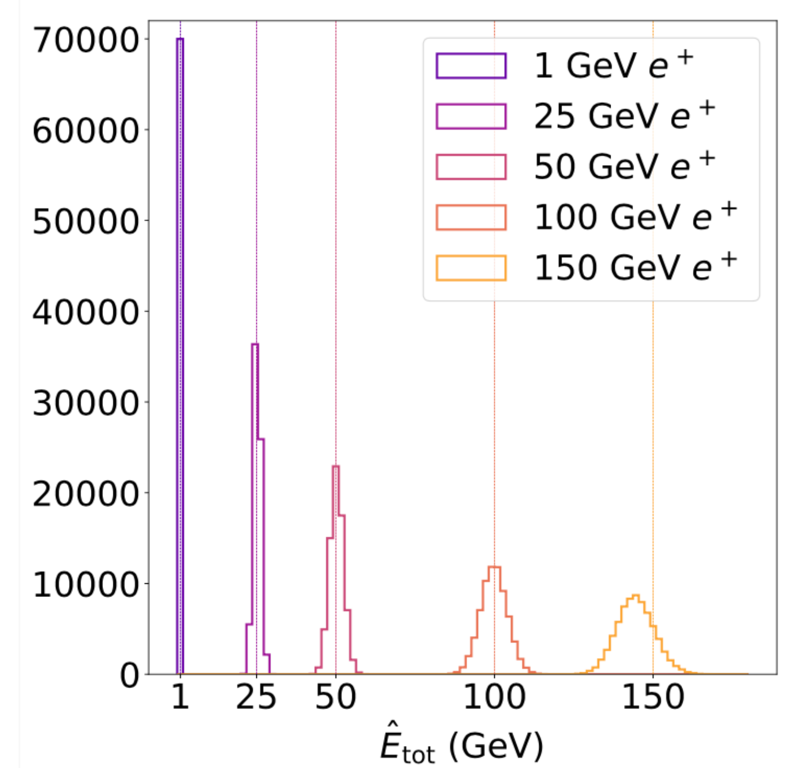


Figure 3. Interpolation across physical range of incident energy as a conditioning latent factor for e^+ showers, with energy increasing from 1 GeV to 100 GeV from left to right. Each point in the interpolation is an average of 10 showers, with each point along the traversal build from an identical latent prior z.

arXiv:1711.08813



arXiv:1705.02355

N.B., We also compare using shower shapes (see backup)

Computational Speed-up

- Since a GAN can be expressed using linear algebra primatives, can exploit modern computing platforms (such as GPU) and libraries (such as TensorFlow)

- With large batches, generation time per-particle decreases dramatically

Generation Method	Hardware	Batch Size	milliseconds/shower
GEANT4	CPU	N/A	1772
CALOGAN	CPU	1	13.1
		10	5.11
		128	2.19
		1024	2.03
	GPU	1	14.5
		4	3.68
		128	0.021
		512	0.014
		1024	0.012

Up to a 10^5 speed-up compared to GEANT4!

arXiv:1705.02355

N.B., See backup for benchmarking parameters

Concluding Remarks

- GANs have shown promise in learning complicated, high dimensional physical realizations
- GANs are extensible, and can easily incorporate domain constraints
- Speed-up from GANs compared to traditional modeling is promising
- Achievements so far using basic GAN formulation! Doesn't include recent work on ProgressiveGANs or Spectral Normalization - can only get better!

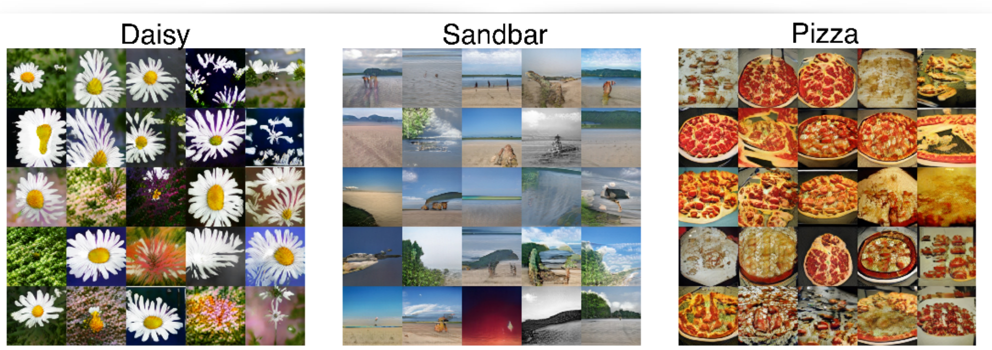


Figure 7: 128x128 pixel images generated by SN-GANs trained on ILSVRC2012 dataset. The inception score is 21.1±.35.

arxiv:1802.05957

Figure 5: 1024 × 1024 images generated using the CELEBA-HQ dataset. See Appendix F for a larger set of results, and the accompanying video for latent space interpolations.

arxiv:1710.10196

Thanks!

Backup

Attribute Conditioning

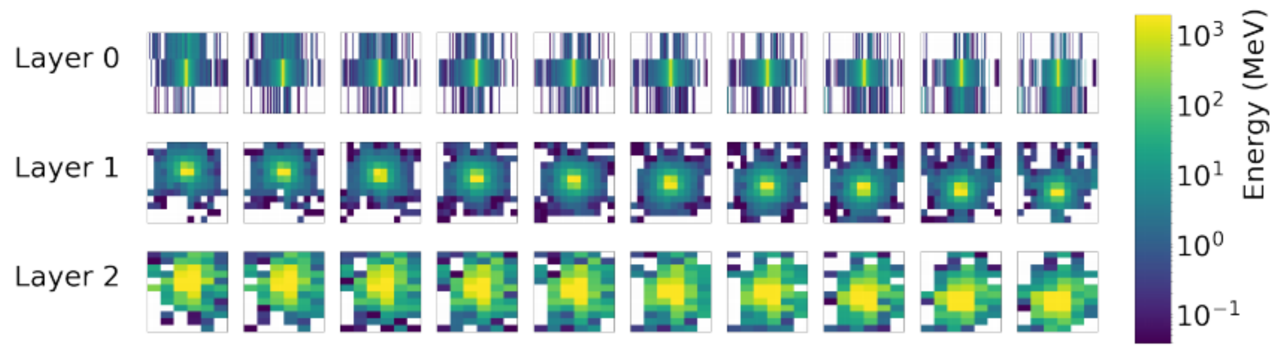


Figure 4. Interpolation across physical range of x_0 as a conditioning latent factor for e^+ showers. Note in the ATLAS coordinate system, x represents the vertical direction in this dataset. Each point in the interpolation is an average of 10 showers, with each point along the traversal build from an identical latent prior z.

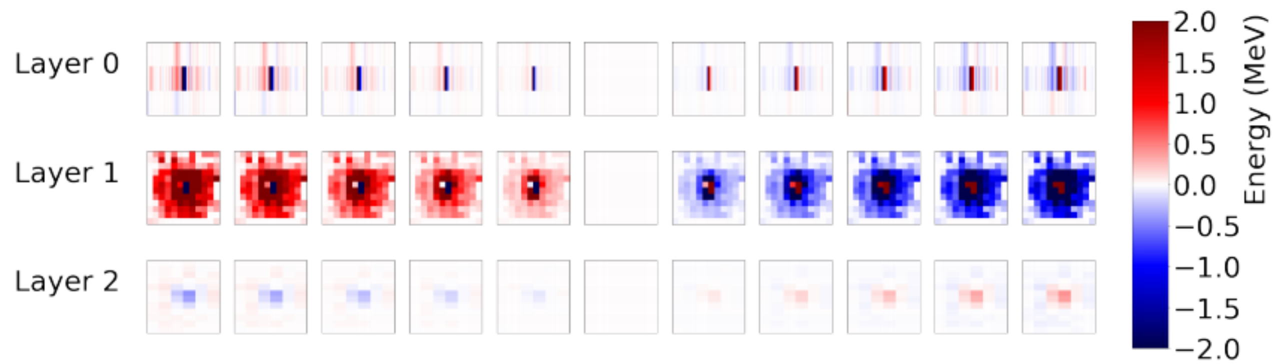
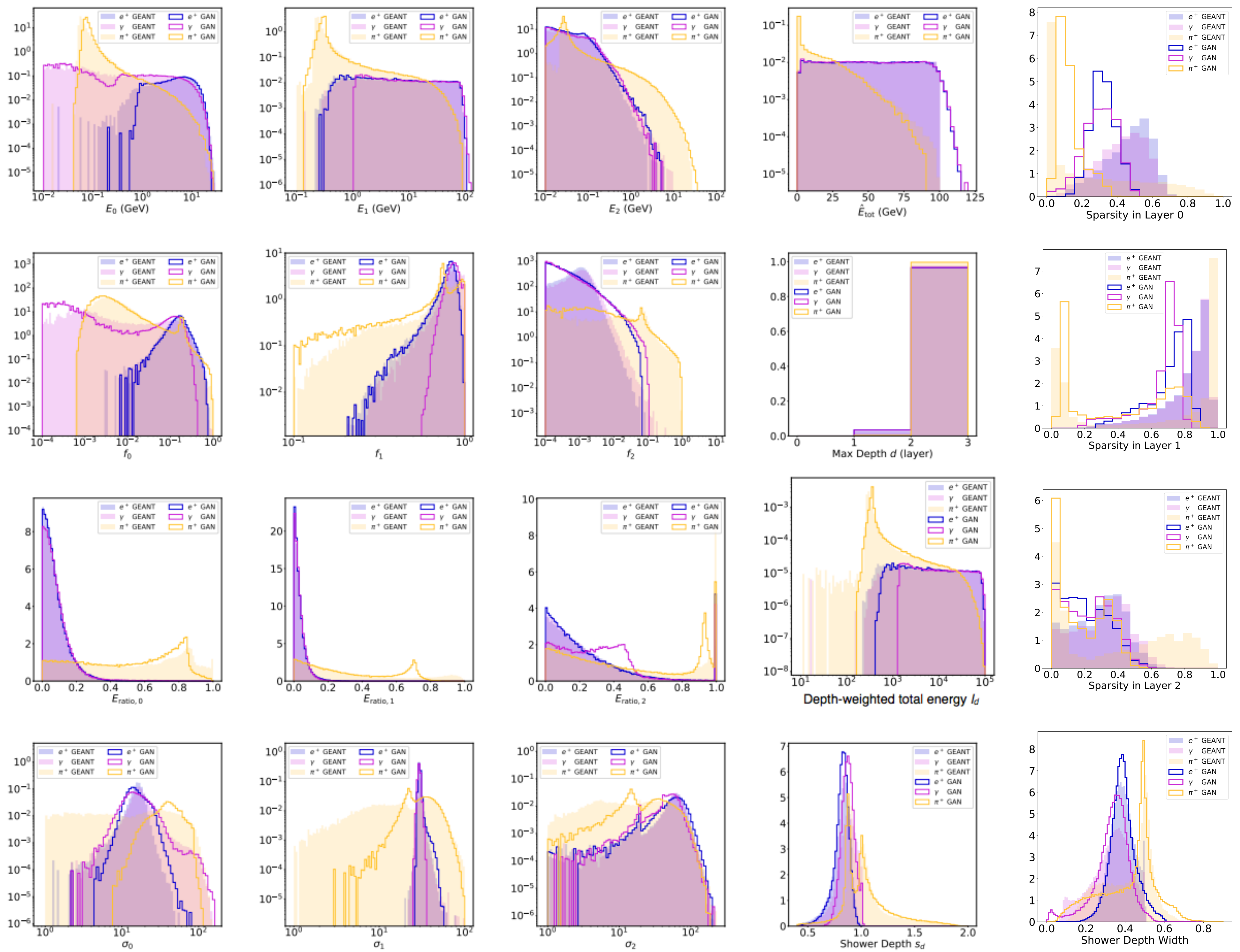


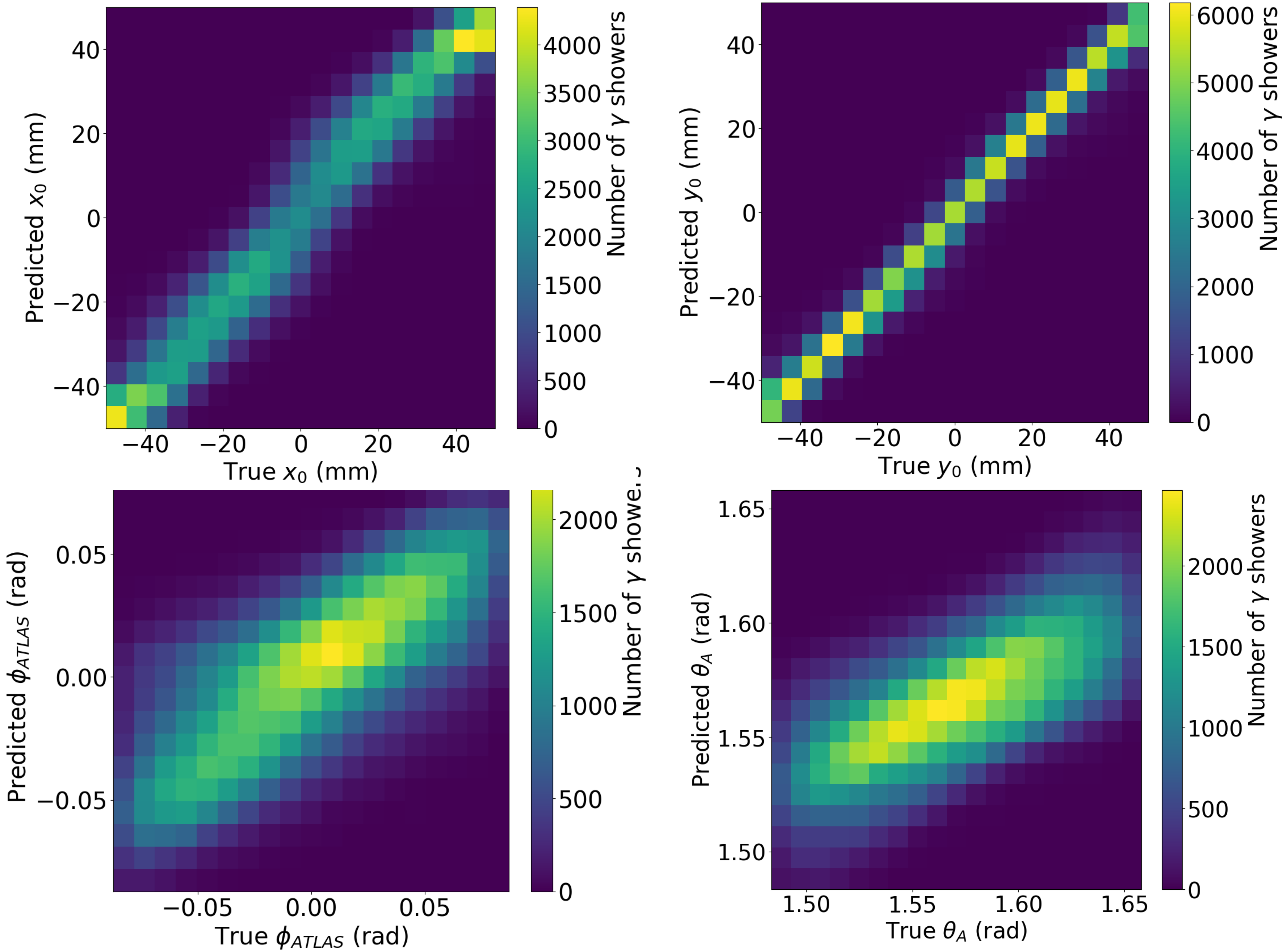
Figure 5. Interpolation across physical range of θ as a conditioning latent factor for $e+$ showers, with θ increasing from left to right. Each point in the interpolation is an average of 10 showers subtracted from the middle point along the interpolation path, with each point along the traversal build from an identical latent prior z.

Verification with 1D projections

Physical one-dimensional statistics of the shower probability distribution can help us verify how well our surrogate performs



Attribute Regression



As a byproduct, the CaloGAN learns to regress out initial conditions

Benchmarking Parameters

- Numerical results are obtained over an average of 100 runs.
- Geant4 and CaloGAN on CPU are benchmarked on nearly identical compute-nodes on the PDSF distributed cluster at the National Energy Research Scientific Computing Center (NERSC).
- CaloGAN on GPU hardware is benchmarked on an Amazon Web Service (AWS) p2.8xlarge instance, where a single NVIDIA® K80 is used for the purposes of benchmarking.