Learning Particle Physics by Example:

Location-Aware Generative Adversarial Networks for Physics Synthesis

Authors:
Michela Paganini
Luke de Oliveira
Benjamin Nachman

Affiliations:
Yale University, Berkeley Lab
Berkeley Lab, VAI Technologies
Berkeley Lab

February 8, 2017

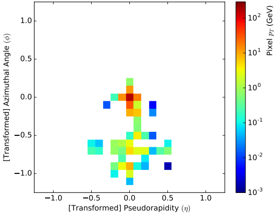
Outline

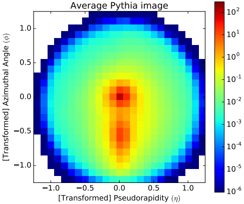
1. Introduction
    - Jet Images
    - GAN, DCGAN, & ACGAN
2. Our Contribution
    - LAGAN
    - Performance Evaluation Techniques
3. Conclusions & Outlook
4. Backup

Introduction

Jet Images

- 2D representation in $\eta - \phi$ space of a jet radiation pattern[1]
- Pixel intensity $= p_T = E_{\text{cell}}/\cosh(\eta_{\text{cell}})$ (non-trivial pre-processing)
- Produced with `Pythia`, clustered with `FastJet`, processed using method described in [2]
- Extensive literature on jet image discrimination[3]





- Sparse (10-15%), with location-dependent properties, intensities varying over ~ 6 orders of magnitude

---

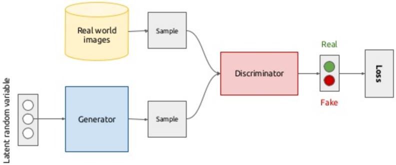
[1] J. Cogan et al., *Jet-Images: Computer Vision Inspired Techniques for Jet Tagging* [arXiv:1407.5675]

[2] L. de Oliveira et al., *Jet-Images - Deep Learning Edition* [arXiv:1511.05190]

[3] [arXiv:1501.05968], [arXiv:1612.01551], [arXiv:1603.09349], [arXiv:1701.08784]

GAN

- **Generative Adversarial Networks**[4]: a framework to train deep generative models as a two player non-cooperative game between a generator network, G, and a discriminator network, D.



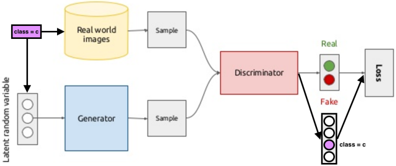
- System's loss:

$$\mathcal{L}_S = \underbrace{\mathbb{E}[\log(\mathbb{P}(D(I) = 0 \mid I \in \mathcal{S}))]}_{\substack{\text{term associated with the discriminator} \\ \text{perceiving a generated sample as fake}}} + \underbrace{\mathbb{E}[\log(\mathbb{P}(D(I) = 1 \mid I \in \mathcal{N}))]}_{\substack{\text{term associated with the discriminator} \\ \text{perceiving a real sample as real}}}$$

[4] I. J. Goodfellow et al., *Generative Adversarial Networks*, [arXiv:1406.2661]

DCGAN and ACGAN

- **Deep Convolutional Generative Adversarial Networks**[5]: CNNs work well with images in supervised learning – let's use them in GANs to generate new images
- **Auxiliary Classifier Generative Adversarial Networks**[6]: add label conditioning as a second task to the discriminator.



- Additional loss:

$$\mathcal{L}_C = \mathbb{E}[\log(\mathbb{P}(C = c \mid I \in \mathcal{S}))] + \mathbb{E}[\log(\mathbb{P}(C = c \mid I \in \mathcal{N}))]$$

[5] A. Radford et al., *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*, [arXiv:1511.06434]

[6] A. Odena et al., *Conditional Image Synthesis With Auxiliary Classifier GANs*, [arXiv:1610.09585].

Our Contribution

LAGAN - Location Aware GAN

We propose:

1. using ACGAN framework to **condition on class label** (signal = boosted W, background = QCD)
2. augmenting DCGAN framework with **locally-connected** layers

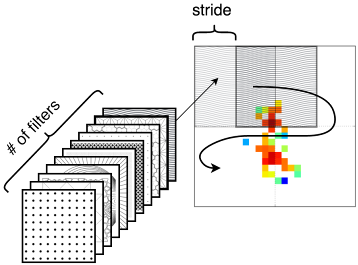


Figure 1: Convolutional Layer

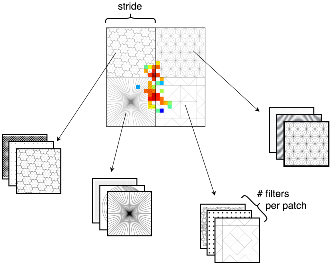
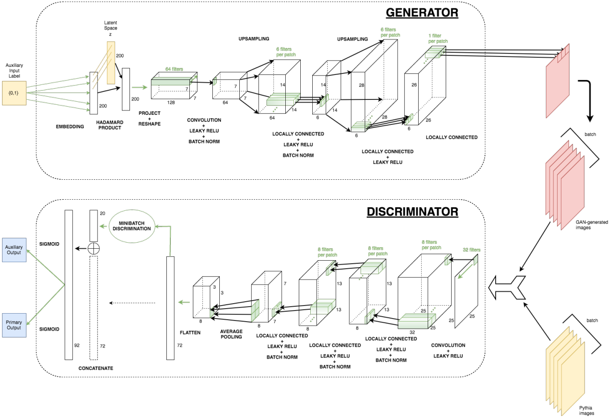


Figure 2: Locally Connected Layer

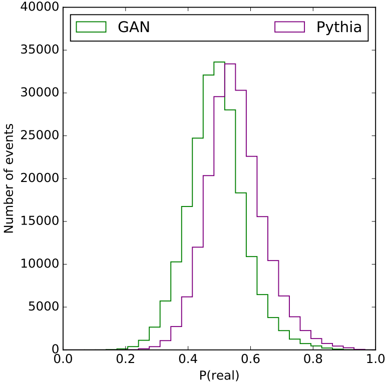
LAGAN (cont.)

3. **leaky ReLU** in G and D, **ReLU** in last layer of G to achieve *sparsity*
4. **minibatch discrimination** in the last layer of D to increase sample *diversity*
5. **batch normalization** for training *stability* in light of the large dynamical range
6. **label flipping** to reduce chances of *mode collapse*



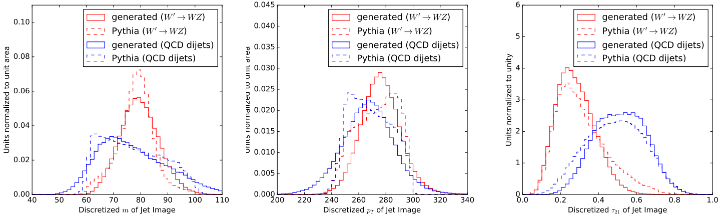
Training

- D tries to correctly identify images from G (fake) vs. images sampled from the data distribution (real)
- G tries to fool D into thinking its images are real
- Alternate G and D training
- Equilibrium when G reproduces data distribution, D outputs $\mathbb{P}$(real)$= 1/2$ everywhere



Physics Observables

Jet images offer **clear techniques for evaluating GAN performance** $\rightarrow$ many jet observables to reduce the 25×25 feature space down to 1D manifolds, preserved under generation



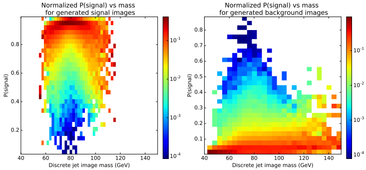
LAGAN is **reproducing 1D projections of the data distribution**, but it's also **internally using their representation for discrimination**

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

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

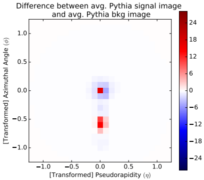
$$\tau_n(I) \propto \sum_i I_i \min_a \left(\sqrt{(\eta_i - \eta_a)^2 + (\phi_i - \phi_a)^2}\right),$$

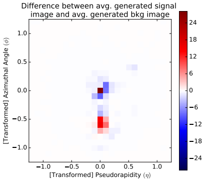
$$\tau_{21}(I) = \tau_2(I)/\tau_1(I)$$



Discriminating Power

The model preserves the "physics", *i.e.* the ability to recover the difference between boosted W bosons- and QCD-originated jet images.





The notion of 2-subjettiness and other radiation patterns are learned early on in the G:

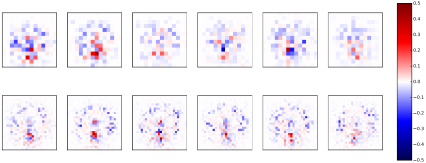


Figure 3: Activations for the channels in the outputs of the two locally connected layers that form G, highlighting the difference between the average signal and average background samples.

Shortcomings

- Despite using label flipping (5% for D, 9% for G), our model still produces very easily classifiable images
  - no in depth exploration of gray area between boosted W and QCD initiated jet images
  - GAN images are not yet a viable *exclusive* substitute for a classifier's training set
  - but still useful for data augmentation

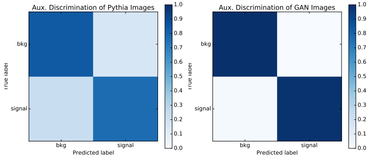


Figure 4: Normalized confusion matrices with percentage of signal and background images that the auxiliary classifier successfully labels, for Pythia images (left) and GAN images (right).

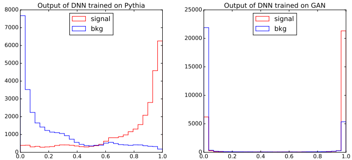


Figure 5: Output of 2 MaxOut nets - one trained on Pythia, one trained on GAN images - evaluated on Pythia images to discriminate W bosons from QCD.

Conclusions & Outlook

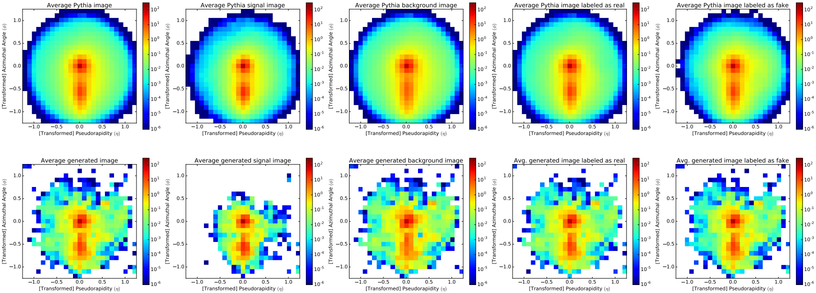
Conclusions & Outlook

- Necessary but not sufficient conditions are met to warrant further studies and research in this direction
- Future work:
  - Include detector simulation
  - Move from 2D to 3D
  - Look at event images instead of single jet images
  - Pythia correction to data via adversarial training
  - Extend to other domains
  - ...

Backup

Visual Inspection

Here is what GAN images look like compared to Pythia images:



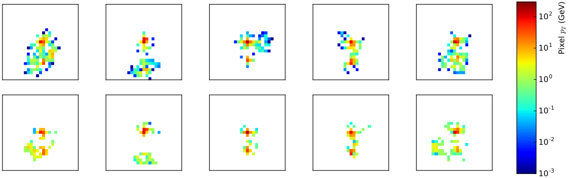


Figure 6: Randomly selected Pythia images (top row) and their nearest generated neighbor (bottom row).