Generative Adversarial Networks

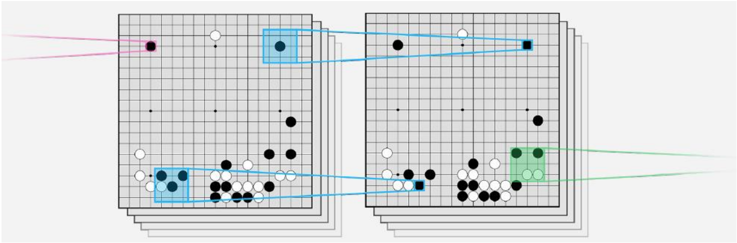
By Yaohang Li, Ph.D.
Department of Computer Science
Old Dominion University
yaohang@cs.odu.edu

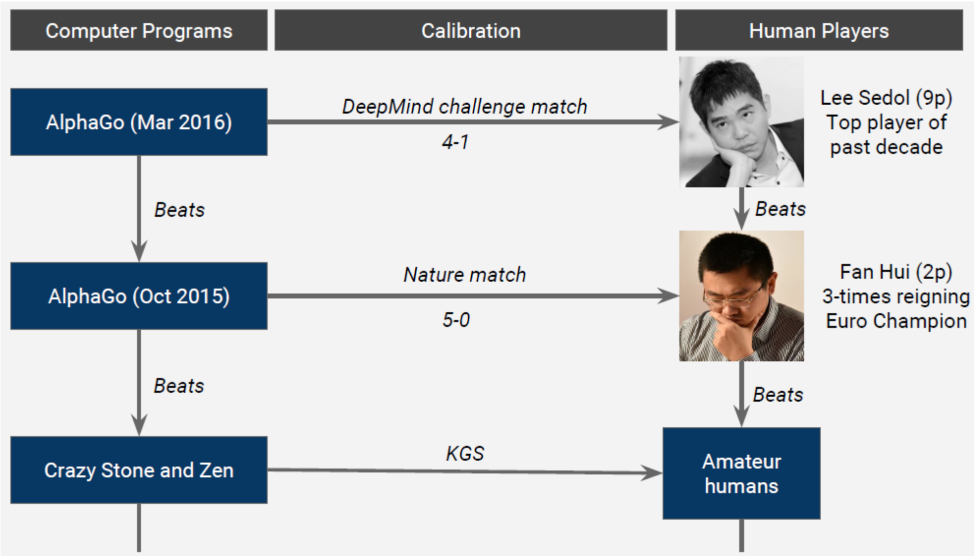
Generative Adversarial Network (GAN)

- **Adversarial Learning**
- **GAN Architecture**
- **DCGAN**
- **GAN Applications**
- **The Fundamentals of GAN**
- **FAT-GAN**

Adversarial Learning

- **AlphaGo**
  - Convolutional Neural Network (CNN)
- **AlphaGo Zero**
  - Adversarial Network





Source: DeepMind

Generative Adversarial Network (GAN)

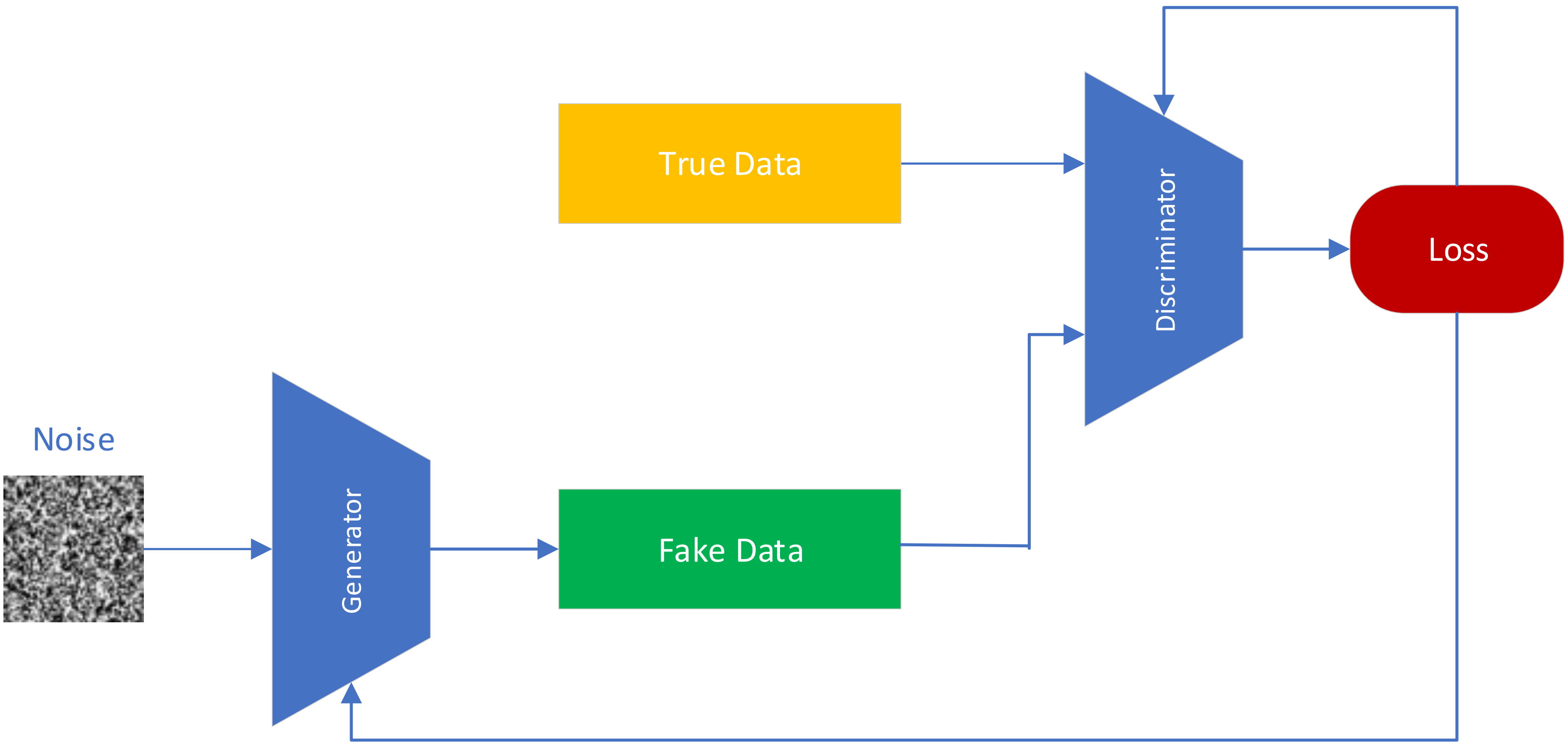
- **Generative Adversarial Networks (GAN)**
  - Introduced by Ian Goodfellow et al. in 2014
  - Deep neural network architectures comprised of two nets
    - A Generator
    - A Discriminator
  - Both nets are trying to optimize a different and opposing loss function in a zero-zum game
- **Potential of GAN**
  - Can be trained to mimic any distribution of data
  - Create worlds eerily similar to our own in any domain

"The most interesting idea in the last 10 years in machine learning" – Yann LeCun

Generative Adversarial Network (GAN)

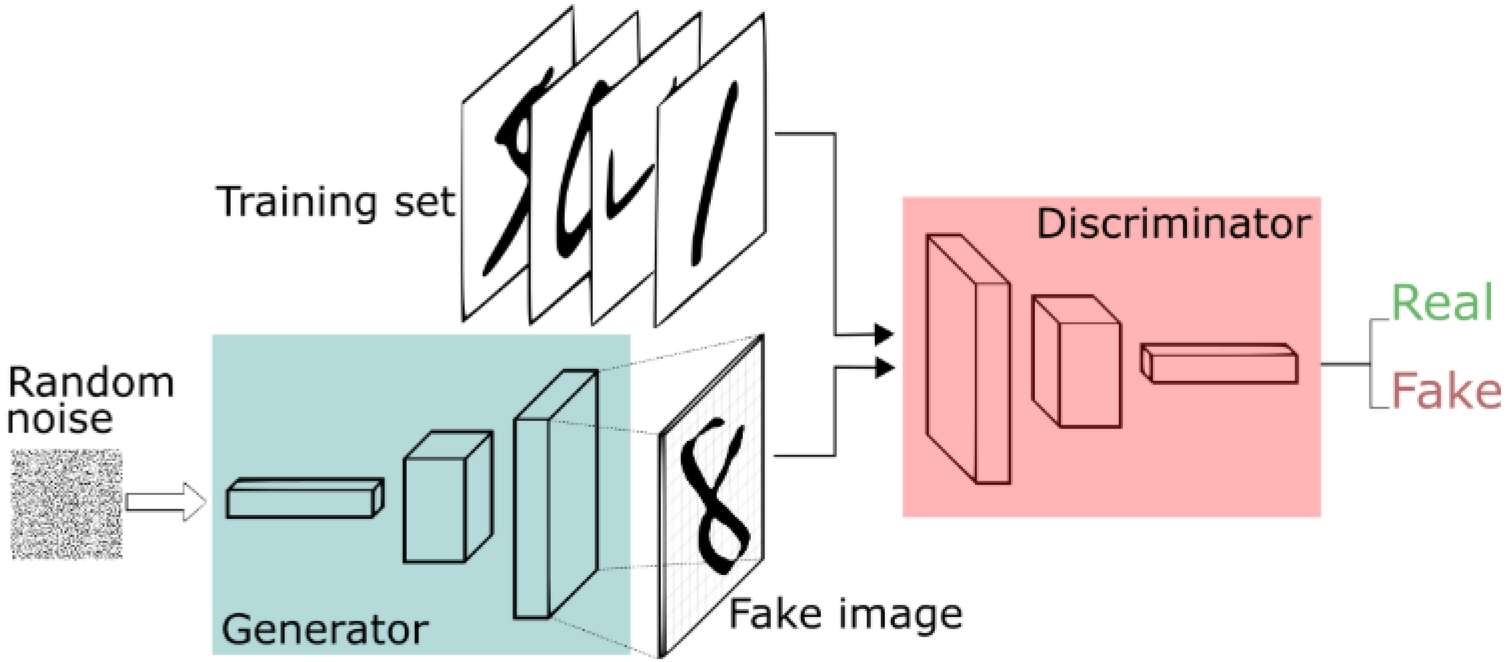
- Adversarial Learning
- **GAN Architecture**
- DCGAN
- GAN Applications
- The Fundamentals of GAN
- FAT-GAN

General Architecture of GAN



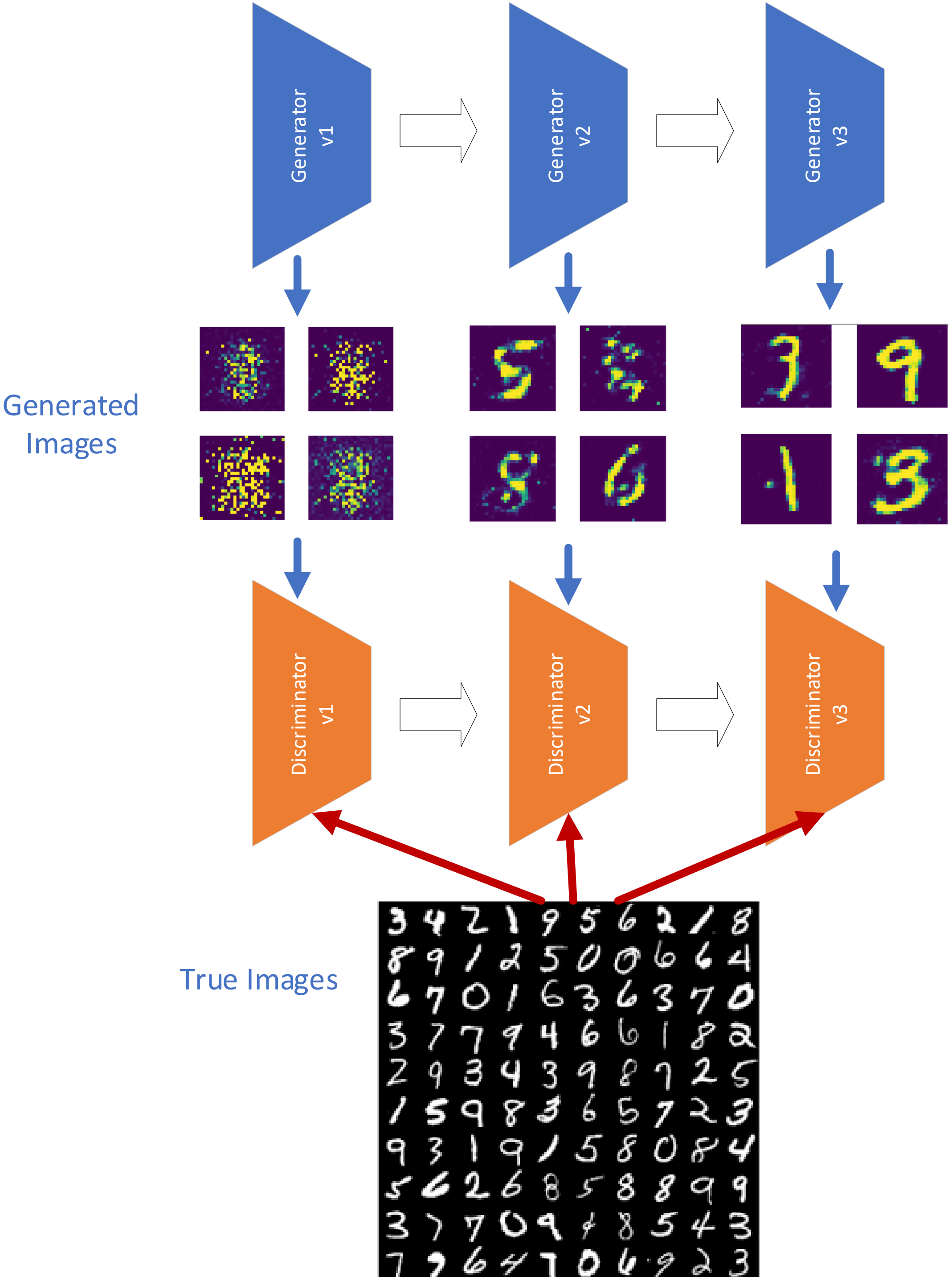
A GAN Example

- **Generating Hand-writing Digits**



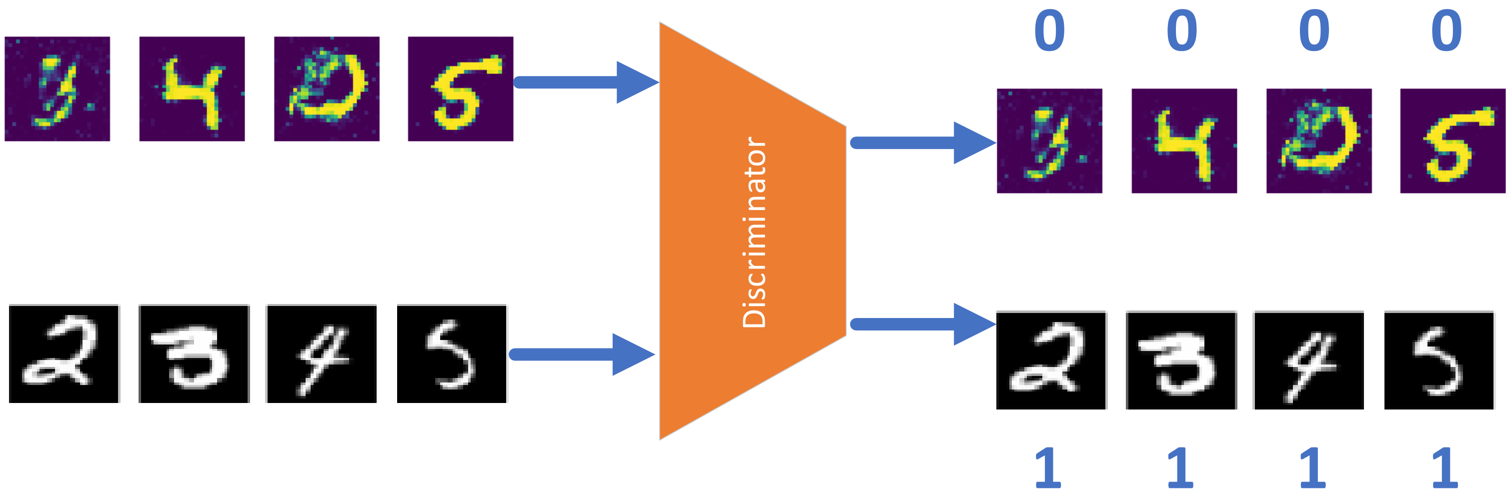
mage credit: Thalles Silva

Step by Step GAN



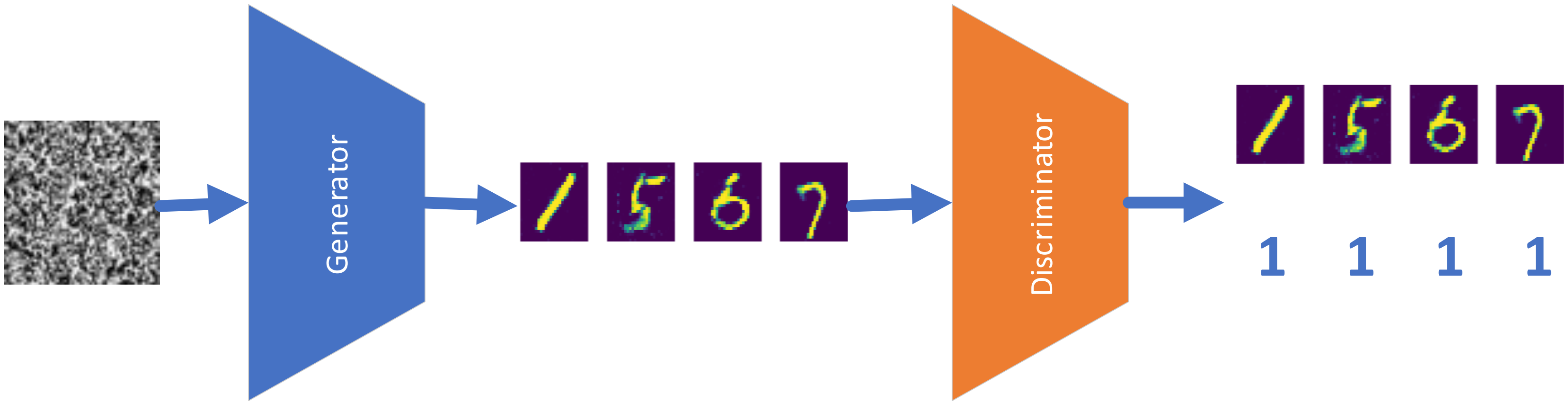
Training a Discriminator

- **Binary Discriminator**



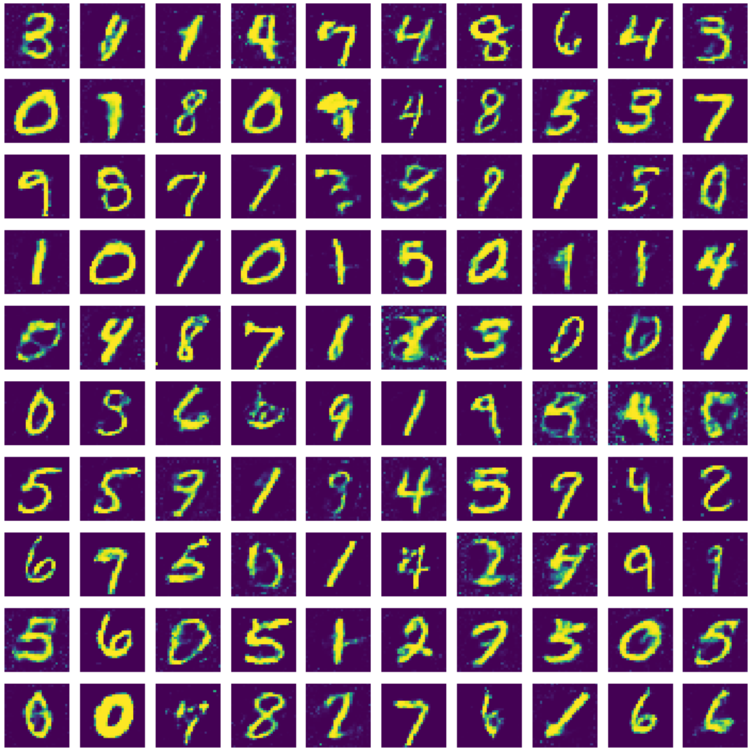
Training the Generator

- **Try to Fool the Discriminator**



Demo: MNIST GAN

- **Jupyter notebook GAN_MNIST.ipynb**

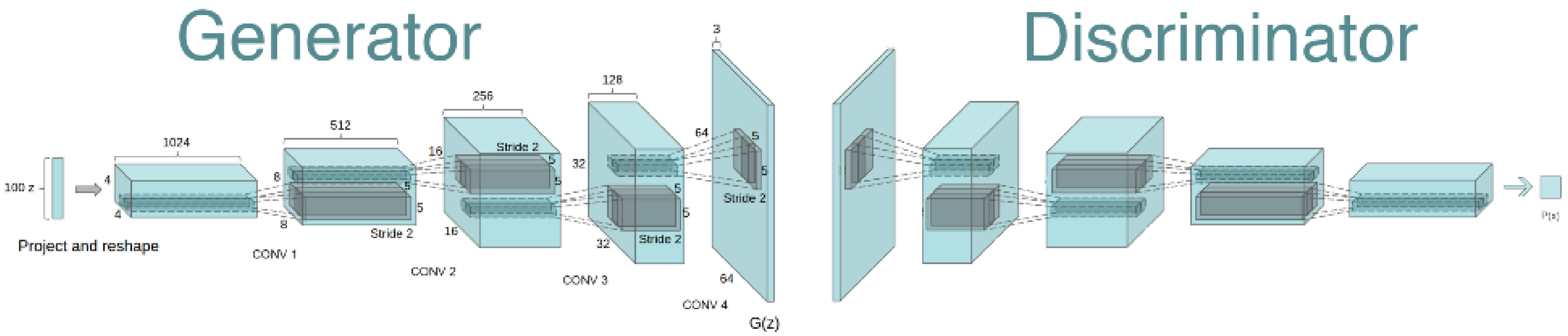


Generative Adversarial Network (GAN)

- Adversarial Learning
- GAN Architecture
- **DCGAN**
- GAN Applications
- The Fundamentals of GAN
- FAT-GAN

Deep Convolutional GAN (DCGAN)

- Improve the sensitivity of Discriminator
- Improve the approximation of Generator

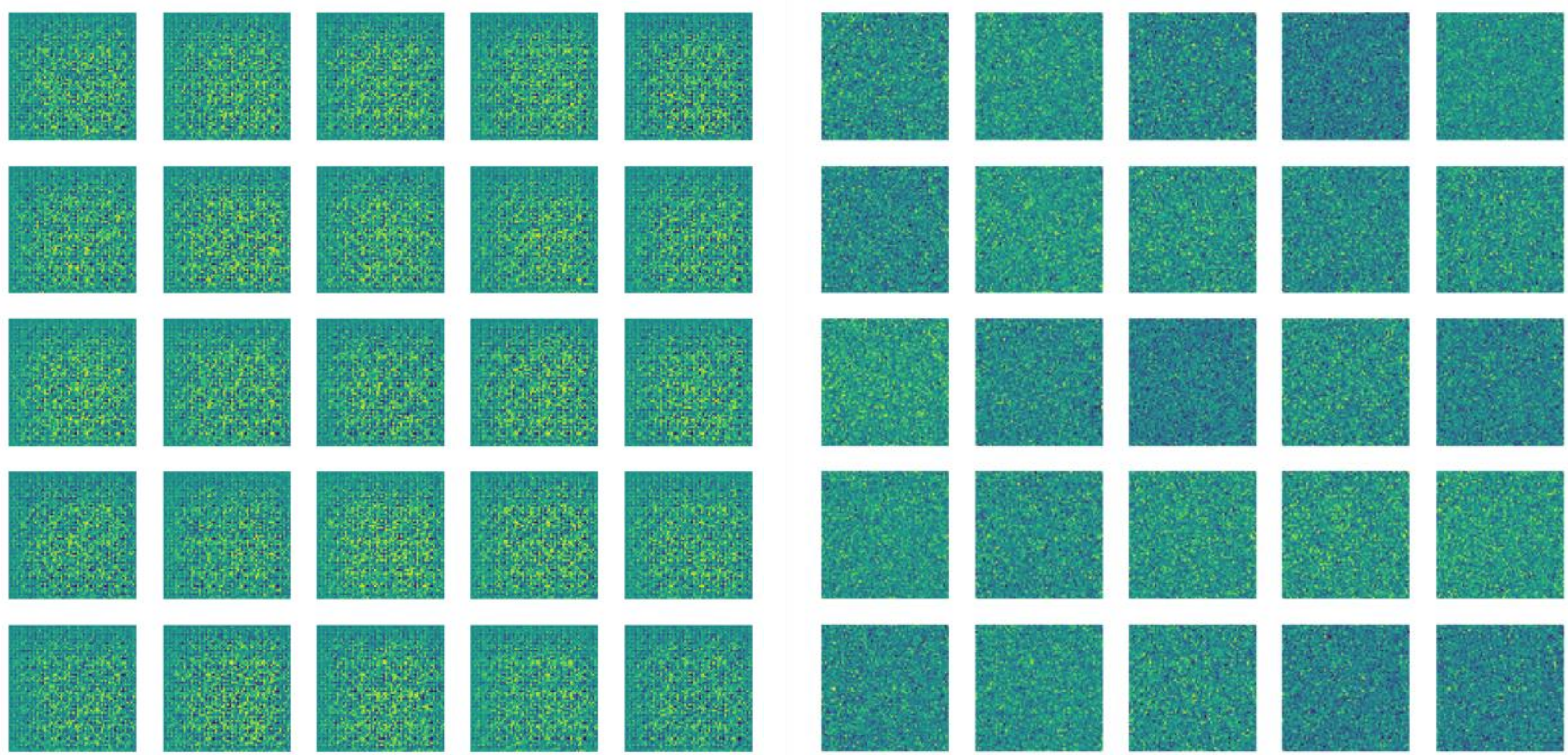


Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv:1511.06434 (2015).

DCGAN Architecture

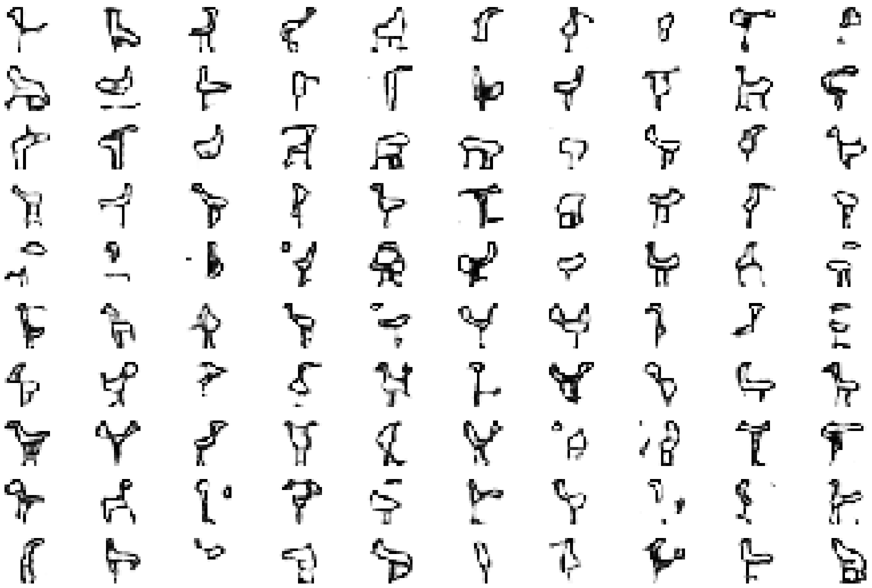
- **Convolution Layer in both Discriminator and Generator**
  - Strided Convolution (Discriminator)
  - Fractional-strided Convolution (Generator)
- **Batch Normalization**
- **Remove Fully-Connected Hidden Layers for Deeper Architecture**
- **ReLU in Generator except for output (Tanh)**
- **LeakyReLU for Discriminator**

DCGAN on MNIST



Demo: Using GAN to draw your own flamingo

Jupyter notebook GAN_flamingo.ipynb



Generative Adversarial Network (GAN)

- Adversarial Learning
- GAN Architecture
- DCGAN
- **GAN Applications**
- The Fundamentals of GAN
- FAT-GAN

The Power of GAN

- **Can be trained to mimic any distribution of data**
- **Applications**
  - Artificial Arts
  - Virtual Reality
  - New Characters
  - Artificial Music

Generating Virtual Arts

http://www.businessinsider.com/robot-can-paint-as-well-as-van-gogh-2015-9

AI Arts

- **AI arts sold for $432,500 in Christie's**

CHRISTIE'S

AUCTIONS PRIVATE SALES LOCATIONS DEPARTMENTS STORIES SERVICES

Is artificial intelligence set to become art's next medium?

12 December 2018

PHOTOGRAPHS & PRINTS |

AI artwork sells for $432,500 — nearly 45 times its high estimate — as Christie's becomes the first auction house to offer

Virtual Reality

Source: Metz and Collins, 2018

4.5 Years of GAN Progress on Face Generation

Ian Goodfellow
@goodfellow_ian

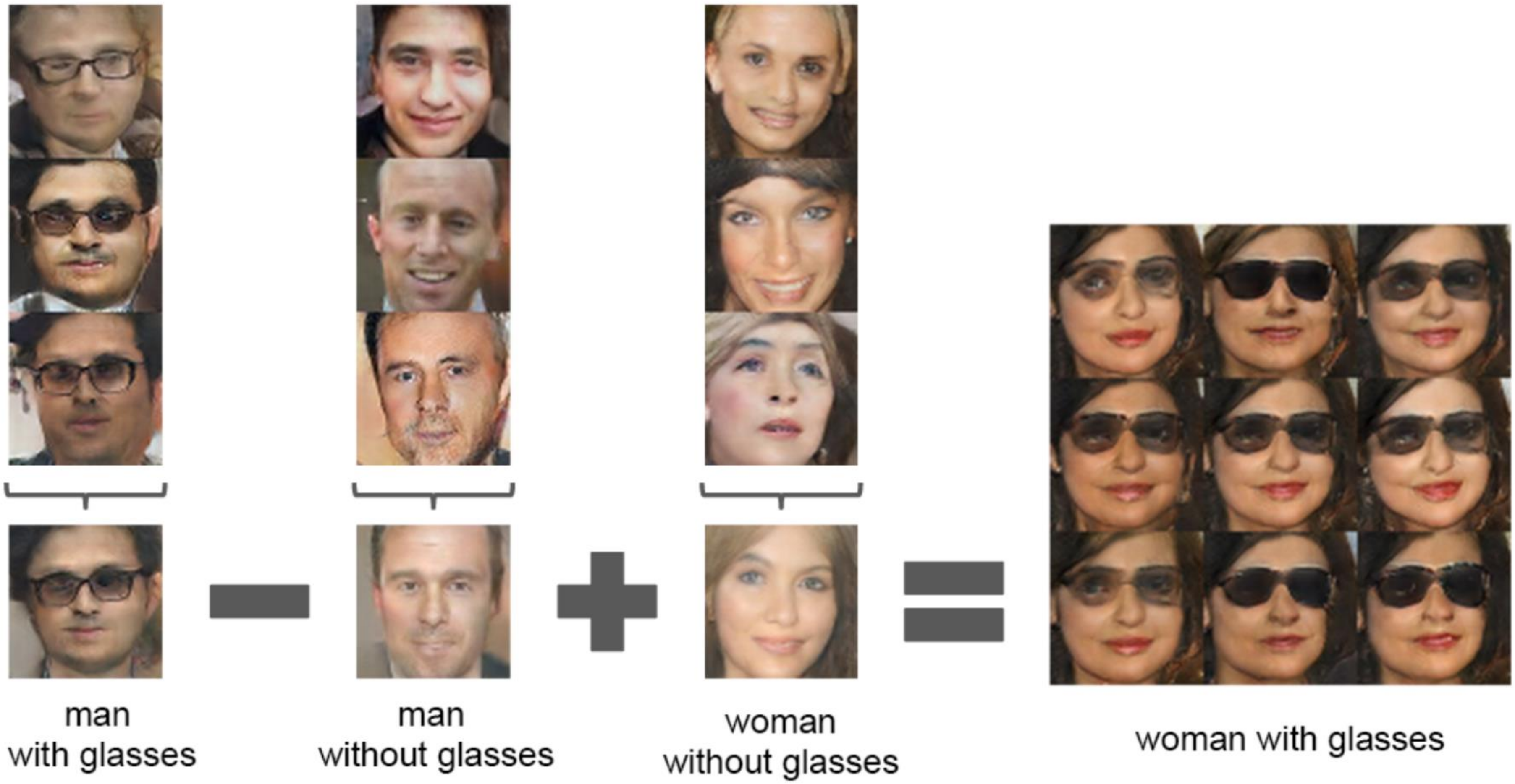
4.5 years of GAN progress on face generation.
arxiv.org/abs/1406.2661 arxiv.org/abs/1511.06434
arxiv.org/abs/1606.07536 arxiv.org/abs/1710.10196
arxiv.org/abs/1812.04948

3,754 8:40 PM - Jan 14, 2019

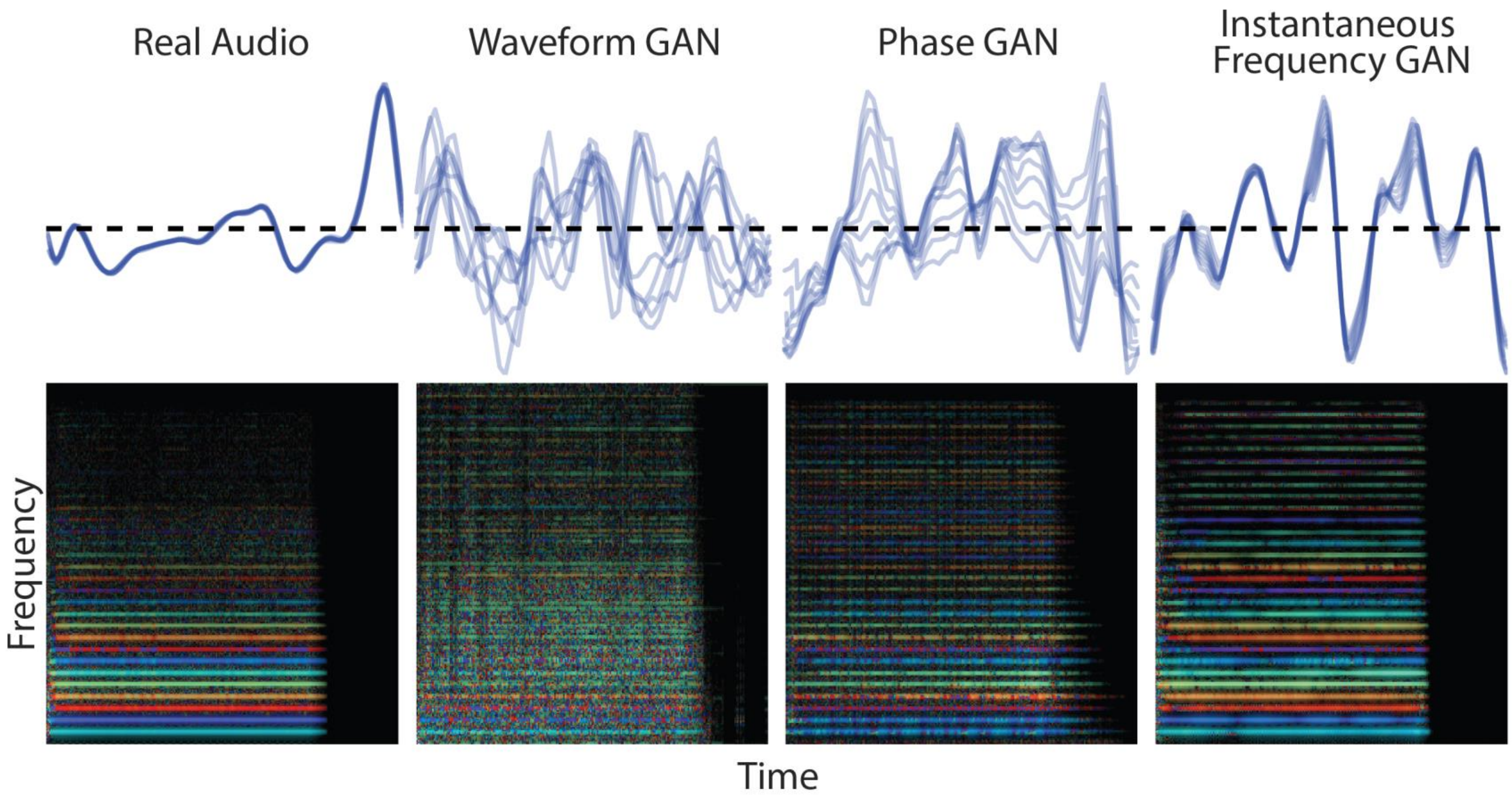
GAN Face Generation Today

https://www.thispersondoesnotexist.com/

Latent Vectors Generated by GAN



GAN Music



Source: gansynth

Generation of Poems

I snatched a letter from my
mother's hand
And ran into the hope of
turning back.
She adored me

Ours learned the wisdom of
ancient groups
Of ancient kings.
I remember the wild useless war

Your help has given
countless people around the
world a voice - the one thing
they need most.

Oh! horrid Night! Melody release
thee from my aching Heart, And
Fate copied on my Mind the
wandring Misfortunes?

Home, home!
Home, sweet home!
Home, sweet!
Home! home! home! home!

Generation of Cartoon Figures

Source of images: https://zhuanlan.zhihu.com/p/24767059
From Dr. HY Lee’s notes.

Natural Language to Images

Image Beautification

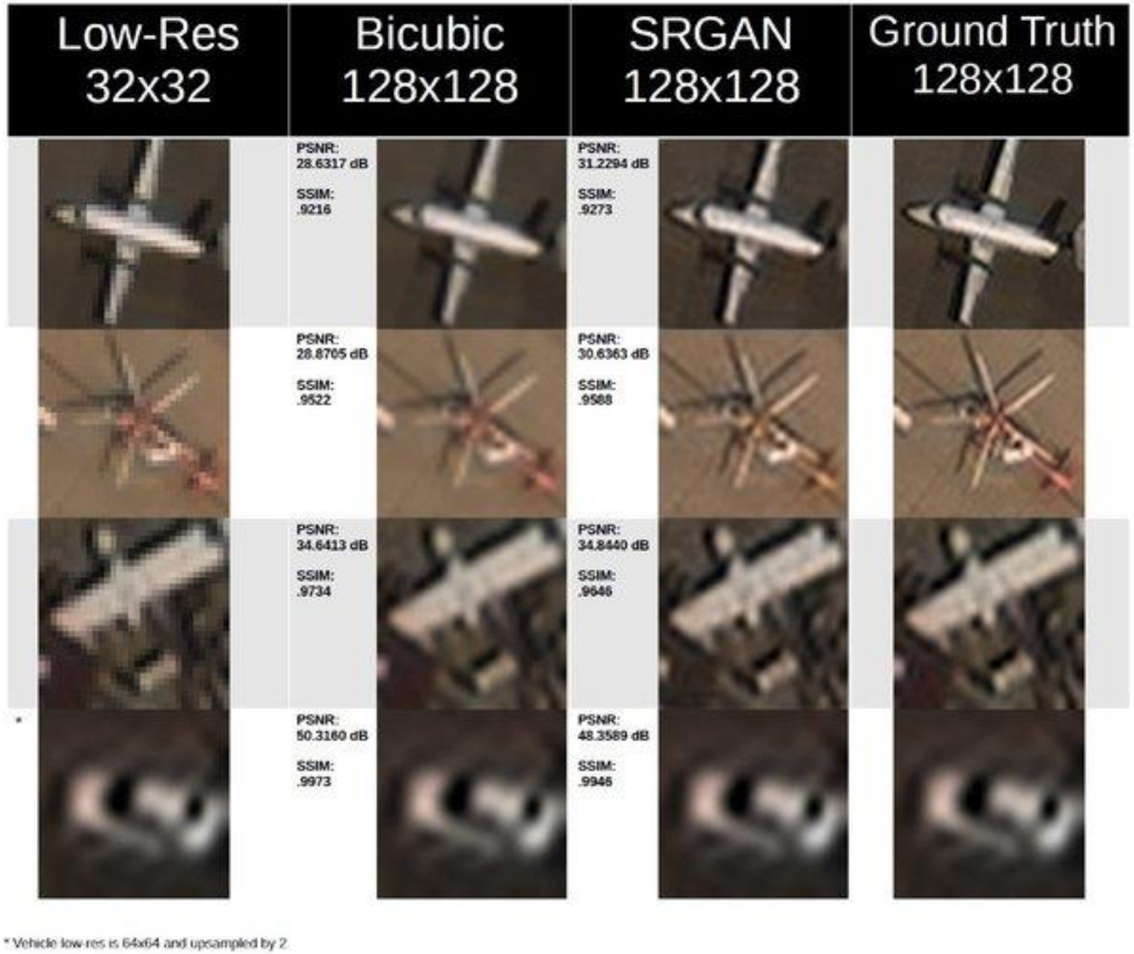
Source: BeHolder-GAN

Image Super-Resolution

- **Convert Low Resolution Images into High Resolution Images**

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, Ledig et al., arXiv:1609.04802

Image Super-Resolution in Military Applications



Jonathan Chin and Asif Mehmood "Generative adversarial networks based super resolution of satellite aircraft imagery", Proc. SPIE 10995, Pattern Recognition and Tracking XXX, 109950W, 2019; https://doi.org/10.1117/12.2524720

Generative Adversarial Network (GAN)

- Adversarial Learning
- GAN Architecture
- DCGAN
- GAN Applications
- **The Fundamentals of GAN**
- FAT-GAN

Fundamentals of GAN

- **Generator G**
  - A Function: Input z, Output x
  - Given a prior distribution $P_{prior}(z)$, a probability distribution $P_G(x)$ is defined by function G
- **Discriminator D**
  - A Function: Input x, Output a scalar
  - Evaluate the difference between $P_G(x)$ and $P_{data}(x)$

Kullback–Leibler Divergence

- **Kullback–Leibler divergence (Relative Entropy)**
  - measures how one probability distribution is different from a reference probability distribution
  - Given probability distributions *P* and *Q*
    - Discrete version

$$D_{\mathrm{KL}}(P||Q) = -\sum_x P(x)\log\left(\frac{Q(x)}{P(x)}\right)$$

    - Continuous version

$$D_{\mathrm{KL}}(P||Q) = -\int P(x)\log\left(\frac{Q(x)}{P(x)}\right)dx$$

Properties of Kullback–Leibler Divergence

- **Explanation of KL divergence**

Cross Entropy of ***P*** and ***Q***

Entropy of ***P***

$$D_{\mathrm{KL}}(P||Q) = -\sum_x P(x) \log\left(\frac{Q(x)}{P(x)}\right)$$

$$= -\sum_x P(x) \log Q(x) - \left(-\sum_x P(x) \log P(x)\right)$$

- **Properties of KL divergence**
  - Non-symmetric
  - Non-negative

Jensen-Shannon Divergence

- **Jensen-Shannon Divergence**
  - Measures the similarity between two probability distributions
  - A symmetrized and smoothed version of the Kullback–Leibler divergence
  - Definition

$$JSD(P||Q) = \frac{1}{2} D_{\mathrm{KL}}(P||M) + \frac{1}{2} D_{\mathrm{KL}}(M||Q)$$

where

$$M = \frac{1}{2}(P + Q)$$

  - Bounds

$$0 \leq JSD(P||Q) \leq \log(2)$$

GAN Cost Function

- **An optimization problem**
  - Find an optimal generator G* such that

$$G^* = \arg\min_G \max_D V(G,D)$$

  - A minimax algorithm
- **Jensen-Shannon Divergence (Information Radius)**
  - $V = E_{x\sim P\_data}[\log D(x)] + E_{x\sim P\_G}[\log(1-D(x))]$
  - Measures the similarity between two distributions

$\max_D V(G, D)$

- $\max_D V(G, D)$
  - Given a generator G
  - $\max_D V(G,D)$ evaluates the "difference" between P_G and P_{data}
- **What is the optimal D^* that maximize $V(G, D)$?**

$$V = E_{x \sim P_{data}}[\log D(x)] + E_{x \sim P_G}[\log(1 - D(x))]$$

$$= \sum_x P_{data}(x) \log D(x) + \sum_x P_G(x) \log(1 - D(x))$$

Then

$$D^* = P_{data}(x) / (P_{data}(x) + P_G(x))$$

$\min_G \max_D V(G, D)$

$\max_D V(G, D)$
$= V(G, D^*)$ **where** $D^* = P_{data}(x)/(P_{data}(x) + P_G(x))$

$$= E_{x\sim P_{data}}[\log D^*(x)] + E_{x\sim P_G}[\log(1 - D^*(x))]$$
$$= \sum_x P_{data}(x)\log D^*(x) + \sum_x P_G(x)\log(1 - D^*(x))$$
$$= -2\log 2 + 2JSD(P_{data}||P_G)$$

What is G^* **with** $\min_G \max_D V(G, D)$**?**

$$JSD(P_{data}||P_G) = 0$$

i.e., $P_{data} = P_G$

Problems in Training a GAN

- **Training a GAN is notoriously difficult**
  - Perfect Discriminator
  - Mode Collapse
  - Non-convergence
  - Imbalance Generator and Discriminator Training
  - Model parameter oscillation
  - Destabilization
  - Vanishing gradient

Training GAN in Practice

- **Discriminator (Binary Classifier)**
  - Given G
    - Sample x_{true} from P_{data}
    - Sample x_G from generator P_G
  - Maximize

$$\sum_{x_{true}} \log D(x_{true}) + \sum_{x_G} \log(1 - D(x_G))$$

If x is true

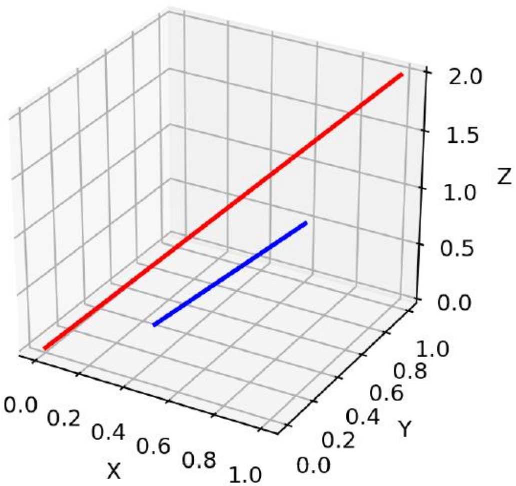
If x is fake

  - Minimize cross-entropy
    - Positive sample: minimize $-\log D(x)$
    - Negative sample: minimize $-\log(1 - D(x))$

Too Perfect Discriminator

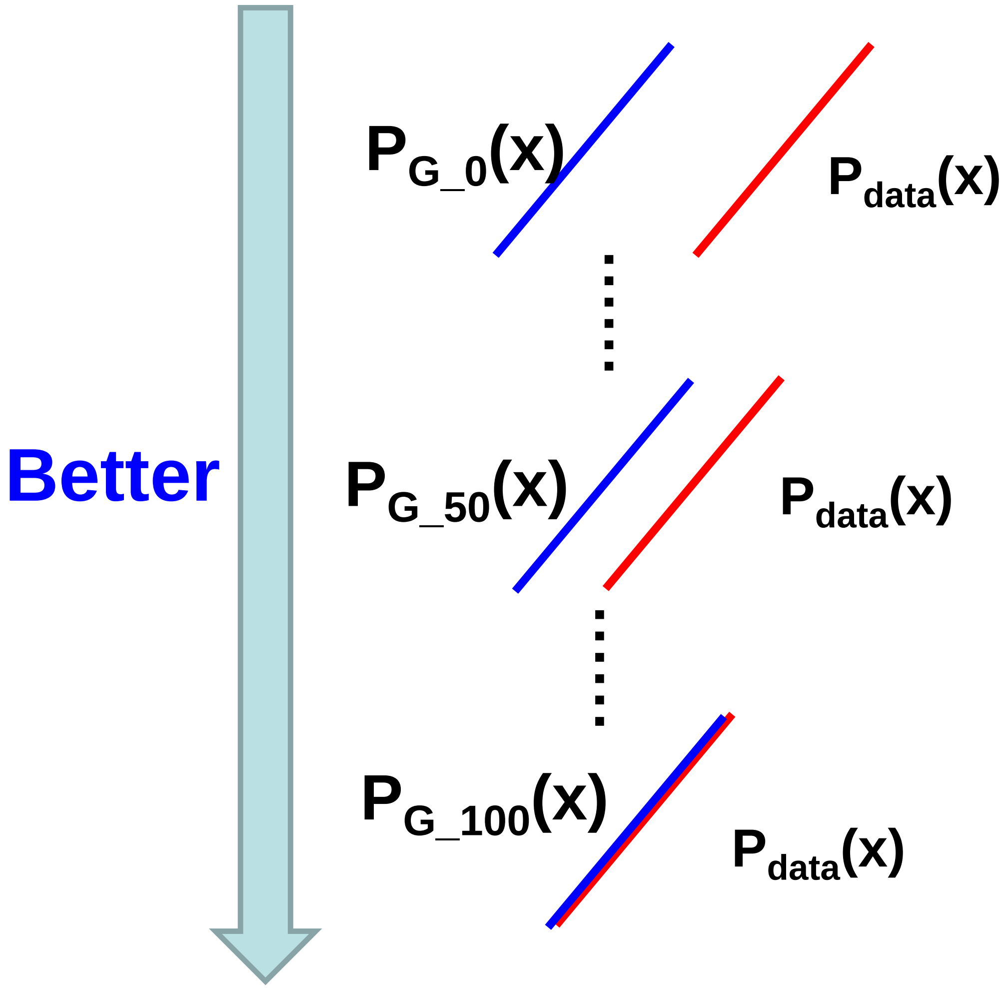
- **Generator**
  - Given D
    - Minimize

$$\cancel{\sum_{x_{true}} \log D(x_{true})} + \sum_{x_G} \log(1 - D(x_G))$$



- **Problem**
  - A nearly perfect discriminator
  - No guideline to generate x_G close to x_{true}
  - Low dimensional manifolds in high-dimensional space barely have overlaps

Too Perfect Discriminator



$JSD(P_{G\_0} \| P_{data}) = \log 2$

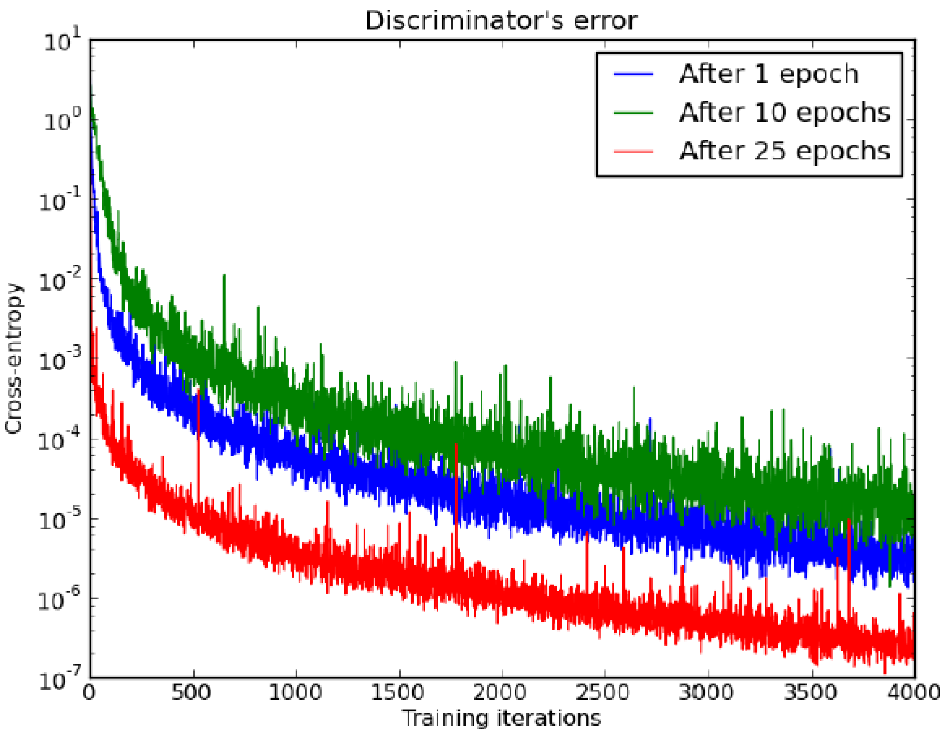
$JSD(P_{G\_50} \| P_{data}) = \log 2$

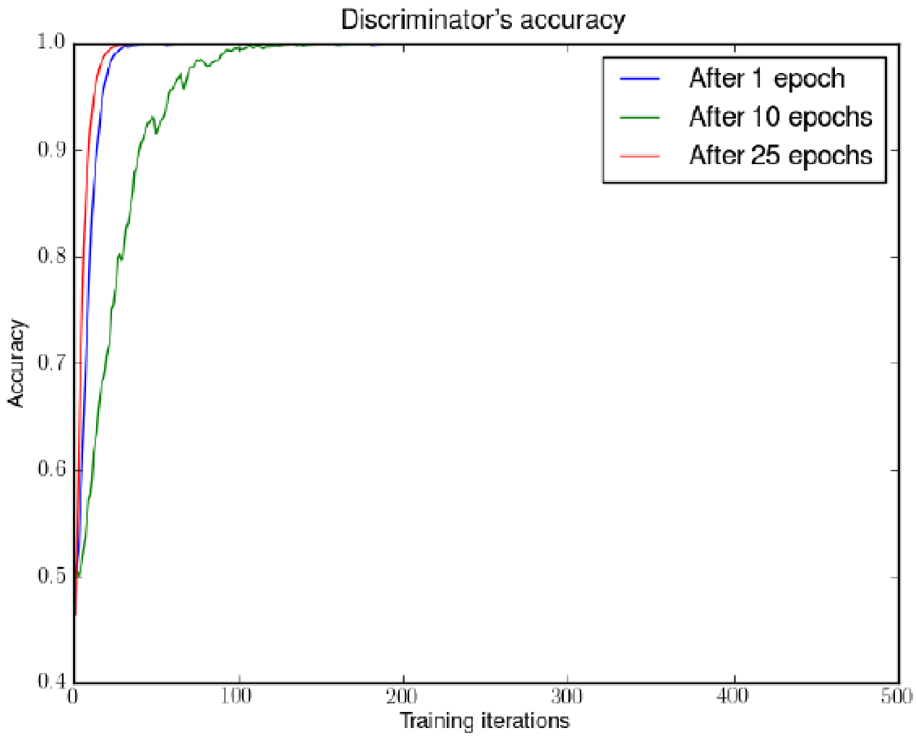
Not really better ……

$JSD(P_{G\_100} \| P_{data}) = 0$

Discriminator too perfect

Discriminator is too strong: for all three Generators, JSD = 0

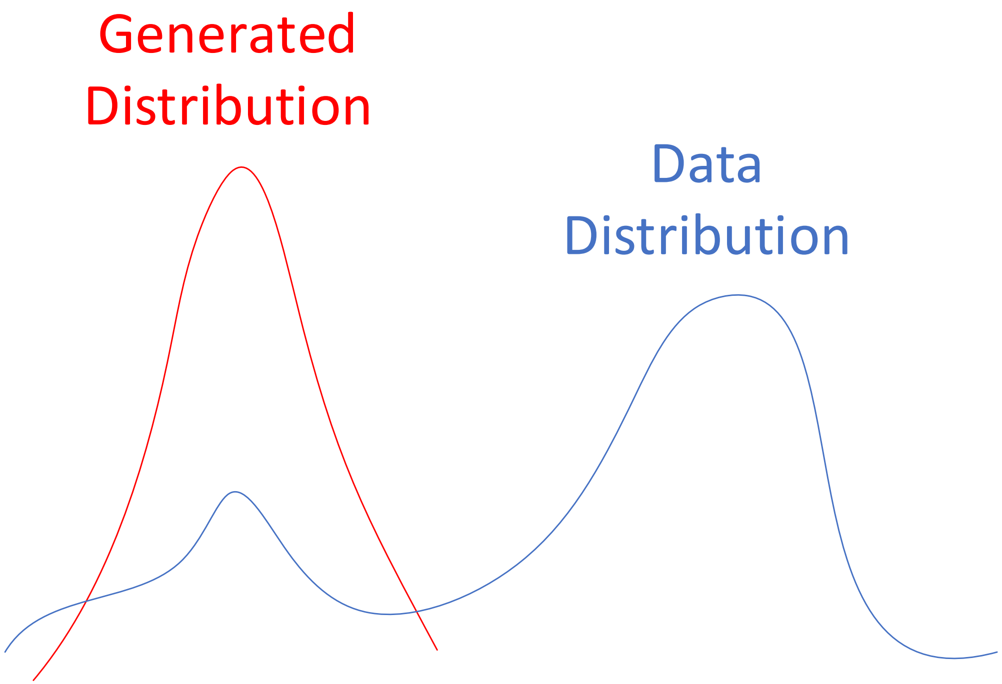




Martin Arjovsky, Léon Bottou, Towards Principled Methods for Training Generative Adversarial Networks, 2017, arXiv preprint

Mode Collapse

- **Generating the same sample**

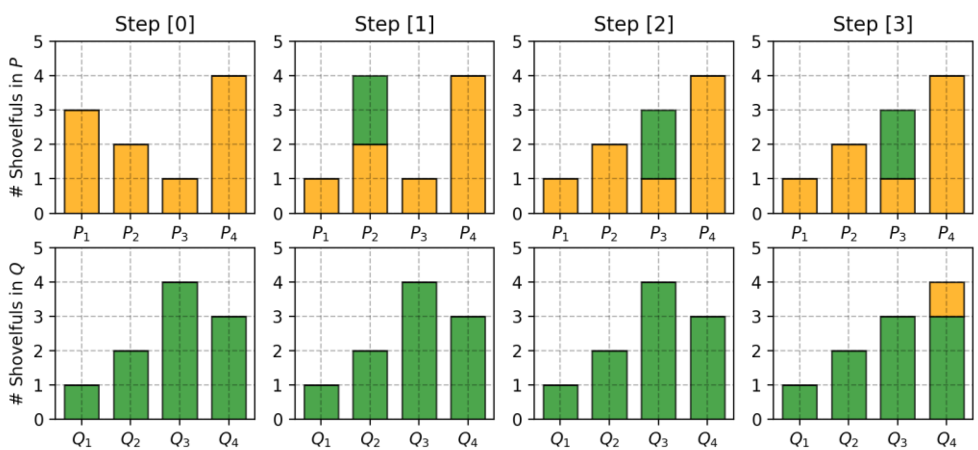


Wasserstein Distance

- **Wasserstein Distance (Earth Mover's Distance)**
  - Measure of the distance between two probability distributions
  - Interpreted as the minimum energy cost of moving and transforming a pile of dirt in the shape of one probability distribution to that of the other
  - Toy Example: Matching Distributions P and Q

P1=3,P2=2,P3=1,P4=4

Q1=1,Q2=2,Q3=4,Q4=3



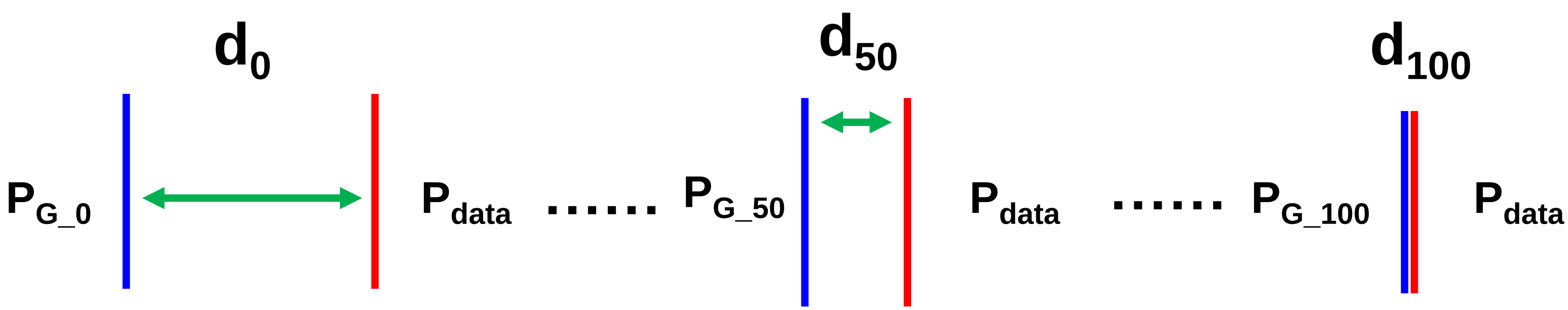
Wasserstein GAN

- **Discriminator**
  - No longer a direct critic of telling the fake samples apart from the real ones
  - Tries to bring $\boldsymbol{x_G}$ distribution closer to $\boldsymbol{x_{true}}$ distribution

- **Wasserstein Loss**

$$\boldsymbol{D_{loss} = E\big(D(x_{true})\big) - E\big(D(x_G)\big)}$$
$$\boldsymbol{G_{loss} = -E(D(x_G))}$$

Wasserstein Distance vs. JSD



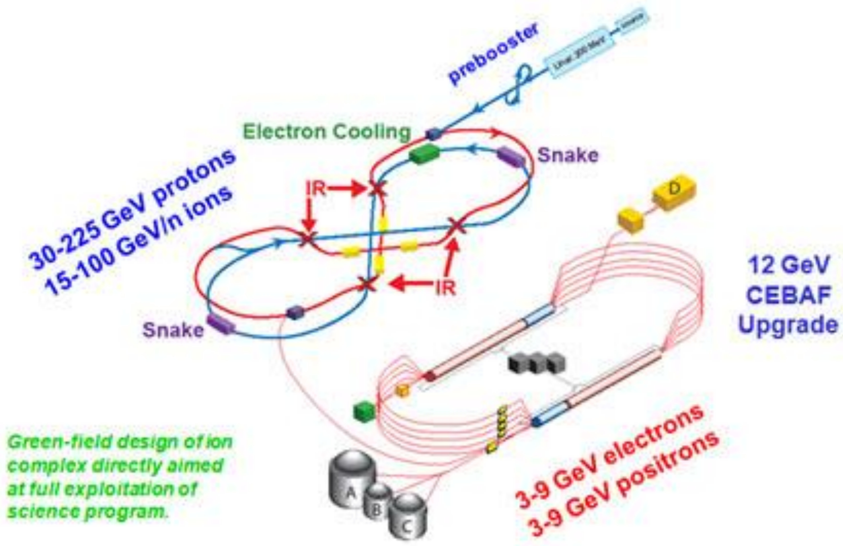
$JS(P_{G\_0}, P_{data}) = \log 2$ $JS(P_{G\_50}, P_{data}) = \log 2$ $JS(P_{G\_100}, P_{data}) = 0$

$W(P_{G\_0}, P_{data}) = d_0$ $W(P_{G\_50}, P_{data}) = d_{50}$ $W(P_{G\_100}, P_{data}) = 0$

Generative Adversarial Network (GAN)

- Adversarial Learning
- GAN Architecture
- DCGAN
- GAN Applications
- The Fundamentals of GAN
- **FAT-GAN**

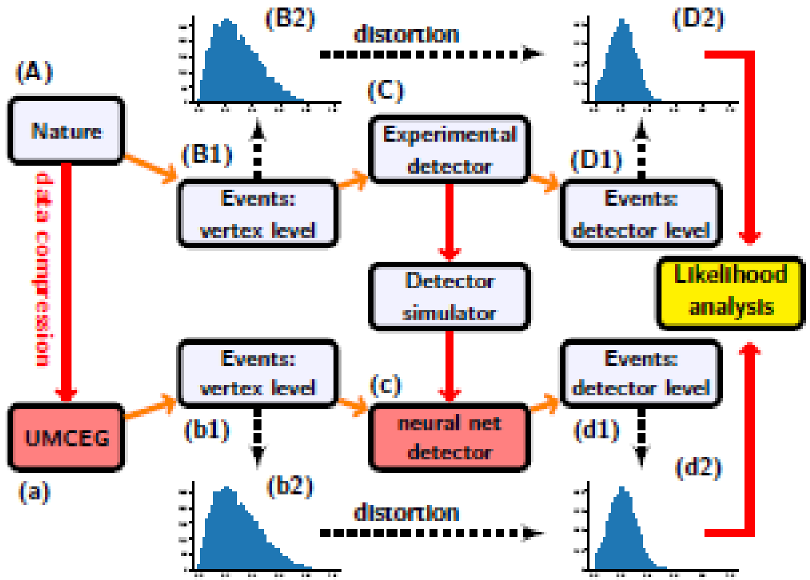
Simulation of Physics Events



Jefferson Lab Electron-Proton Collider

GAN-based Event Generators

- **Learning from real electron-proton scattering data**
  - Capture rich underlying distributions over data
    - Difficult to model using explicit parameters
- **Faithfully reproducing particle reaction events**
  - No assumptions on femtometer-scale physics theory
- **Overcome the limitations of MCEGs**
- **Proof-of-concept on inclusive electrons**

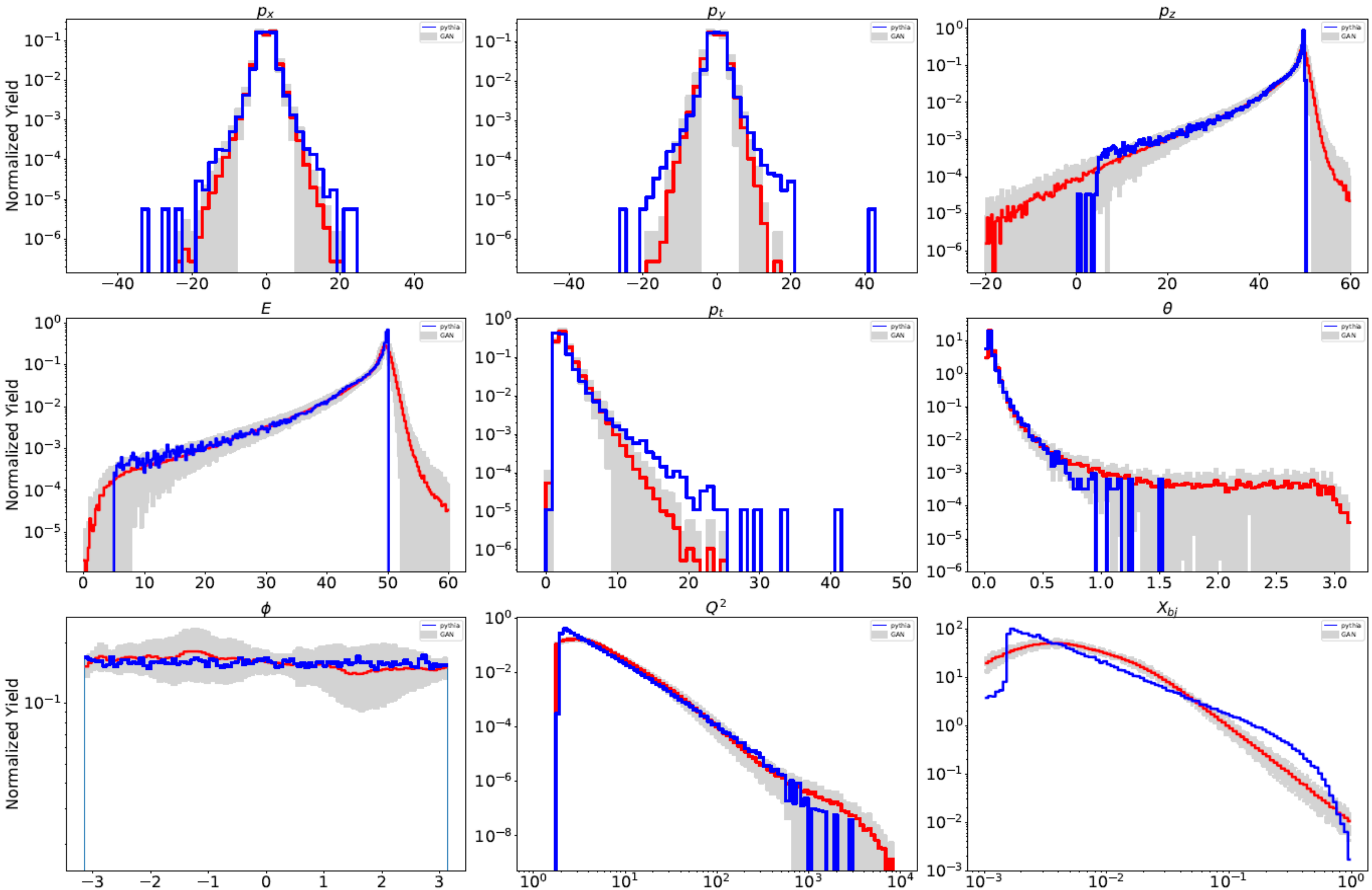


Additional (GAN) Challenges

- Precise Event Feature Distributions
  - Replicate the nature of particle reactions faithfully
- Obeying the fundamental Physics Laws
  - Energy Conservation
  - Momentum Conservation

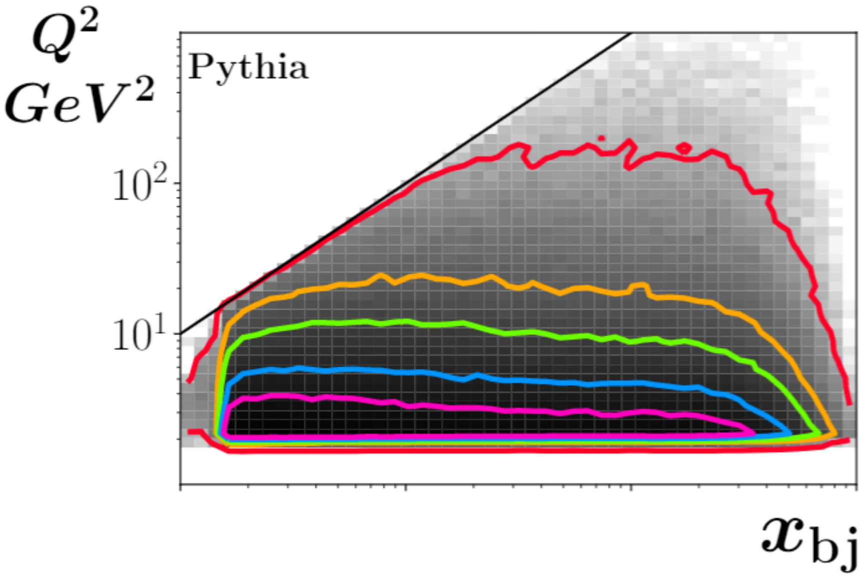
Direct Simulation GAN

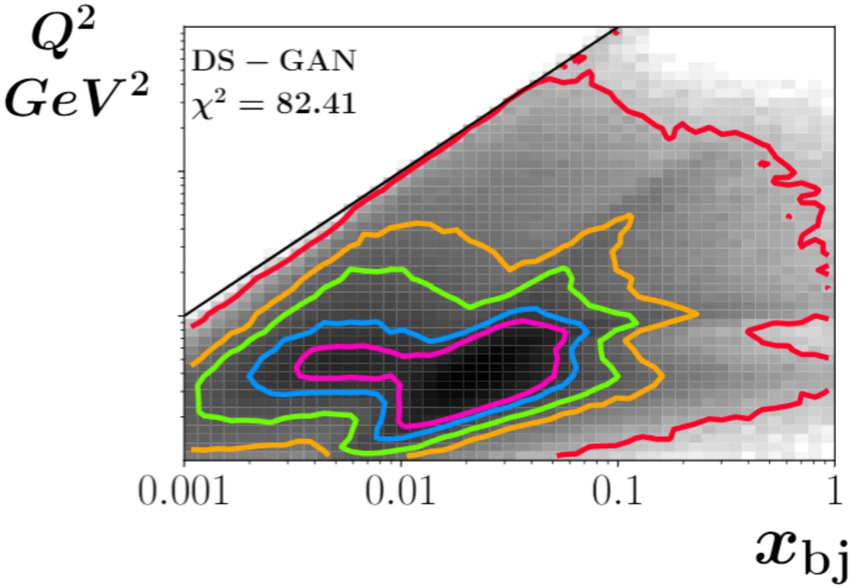
- **Directly learn from electron three-momentum vector (px, py, pz)**



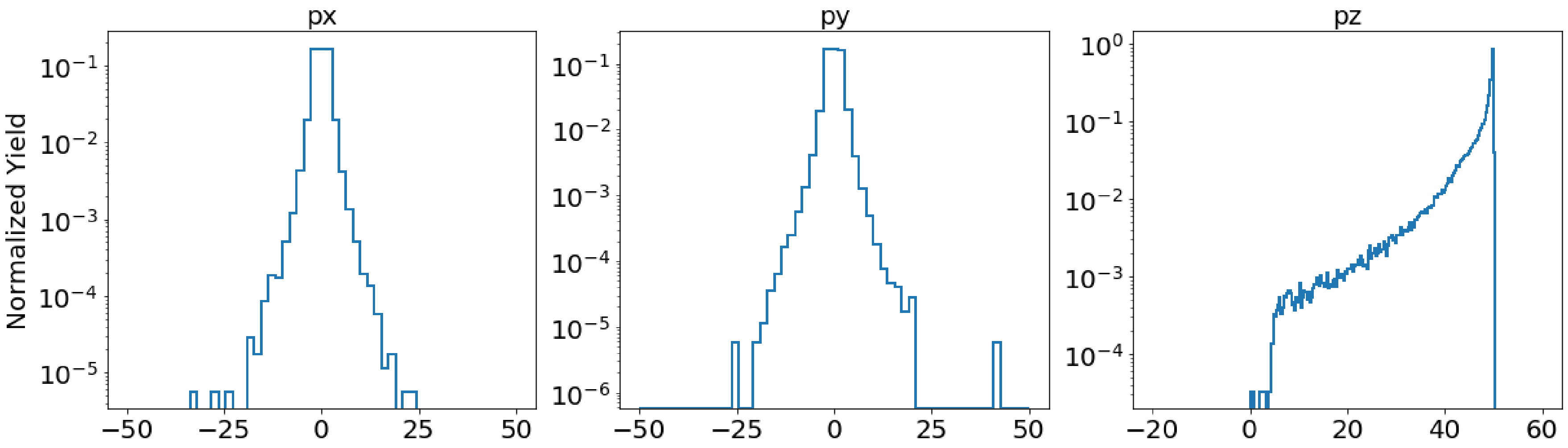
Direct Simulation GAN (cont.)

- **Inter-correlation between physical quantities**





Momenta Distributions of Electrons



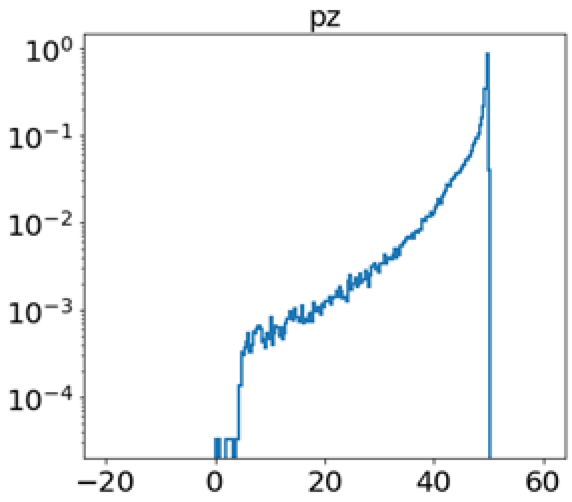
Feature-Augmented Transformed GAN (FAT-GAN)

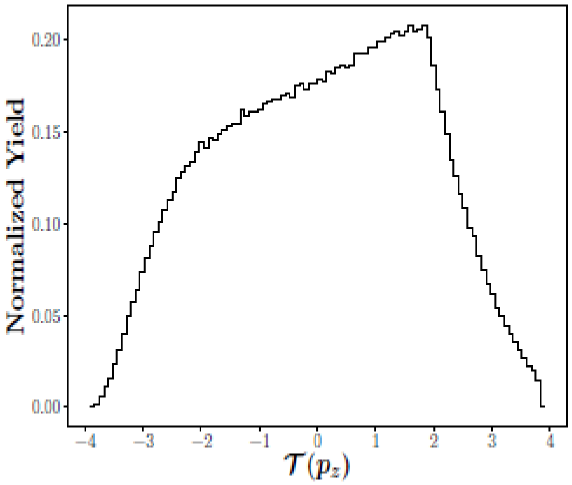
- **FAT-GAN**
  - Features Transformation
    - Select generated features
      - Not necessary meaningful physics properties
    - Easier to be generated by the generator
  - Features Augmentation
    - Expand feature space
    - Improve sensitivity of the discriminator
  - Maximum Mean Discrepancy (MMD)
    - Improve Distribution Match
  - Wasserstein Loss
    - Reduce the chance of mode collapse
    - Enhance GAN convergence

Features Transformation

$$\mathcal{T}(p_z) = \log(E_\mathrm{b} - p_z)$$

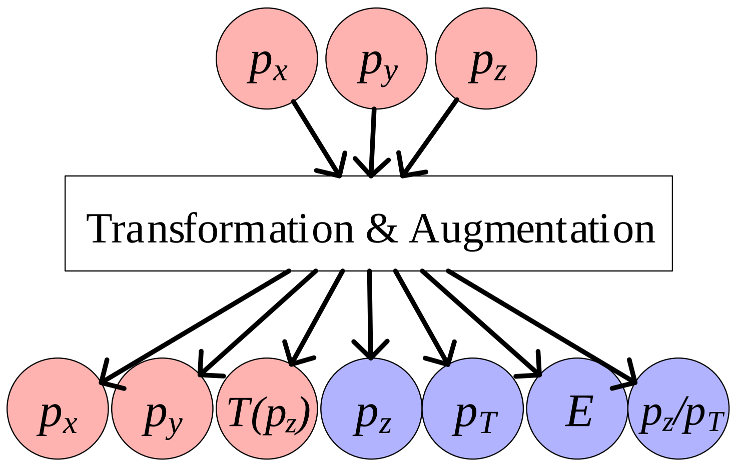
- **Conversion to eliminate sharp edges**
- **Guarantee no generation of non-physical electrons**





Features Augmentation

- **Augment the Feature Space to improve the Sensitivity of Discriminator**



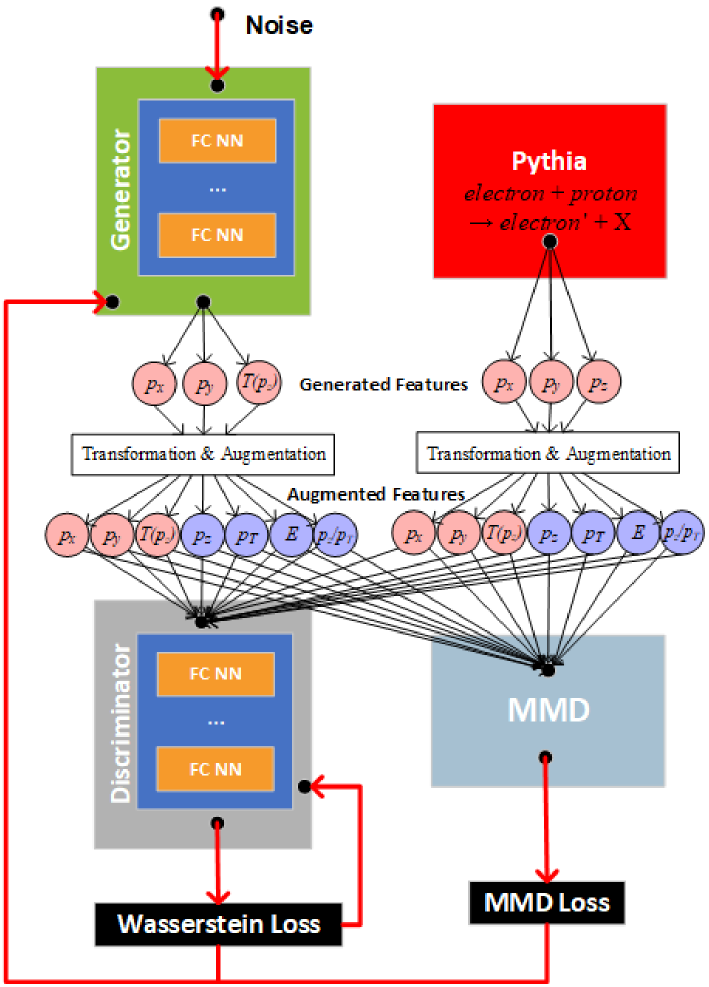
Maximum Mean Discrepancy (MMD)

- MMD
  - A kernel-based two-sample test to compare two distributions
  - Determine if the two sets of samples are drawn from different distributions

$$\begin{aligned}\mathrm{MMD}^2(\boldsymbol{p}, \widetilde{\boldsymbol{p}}) = {} & \mathbb{E}_{\boldsymbol{p}_a, \boldsymbol{p}_{a'} \sim P_{\boldsymbol{p}}}[k(\boldsymbol{p}_a, \boldsymbol{p}_{a'})] \\ & + \mathbb{E}_{\boldsymbol{p}_b, \boldsymbol{p}_{b'} \sim P_{\widetilde{\boldsymbol{p}}}}[k(\boldsymbol{p}_b, \boldsymbol{p}_{b'})] \\ & - 2\,\mathbb{E}_{\boldsymbol{p}_a \sim P_{\boldsymbol{p}}, \boldsymbol{p}_b \sim P_{\widetilde{\boldsymbol{p}}}}[k(\boldsymbol{p}_a, \boldsymbol{p}_b)]\end{aligned}$$

  - $k(\boldsymbol{p}_a, \boldsymbol{p}_b)$ is a Gaussian kernel

FAT-GAN Architecture



- Discriminator Loss

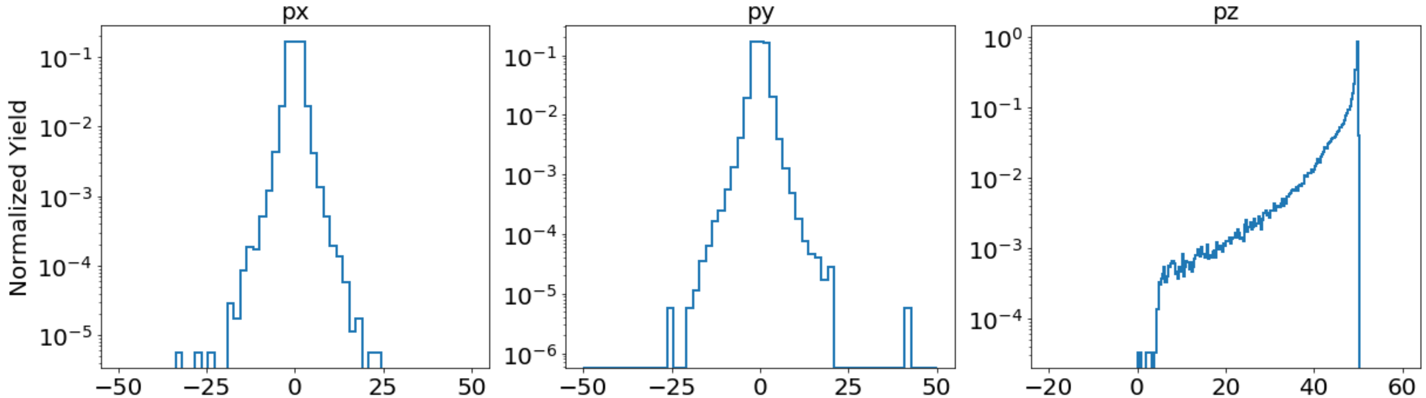
$$L_D = \left(\mathbb{E}[D(\widetilde{\boldsymbol{p}})] - \mathbb{E}[D(\boldsymbol{p})]\right) + \lambda\,\mathbb{E}_{\widehat{\boldsymbol{p}}\sim P_{\widehat{\boldsymbol{p}}}}\left[(\|\nabla_{\widehat{\boldsymbol{p}}} D(\widehat{\boldsymbol{p}})\|_2 - 1)^2\right]$$

- Generator Loss

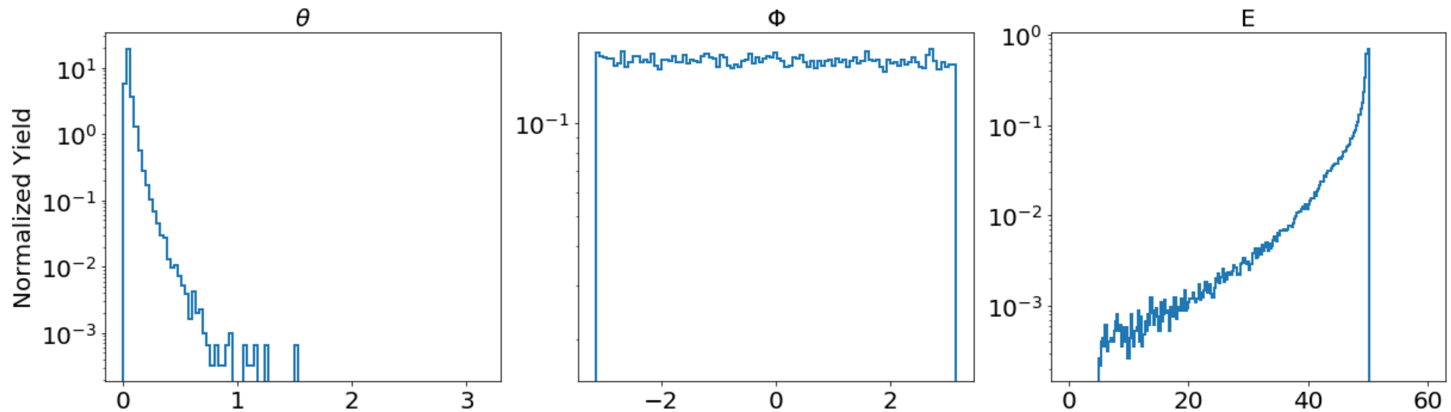
$$L_G = -\mathbb{E}[D(\widetilde{\boldsymbol{p}})] + \eta\,\mathrm{MMD}^2(\boldsymbol{p}, \widetilde{\boldsymbol{p}})$$

Comparison between Representations in Cartesian Coordinates and Spherical Coordinates

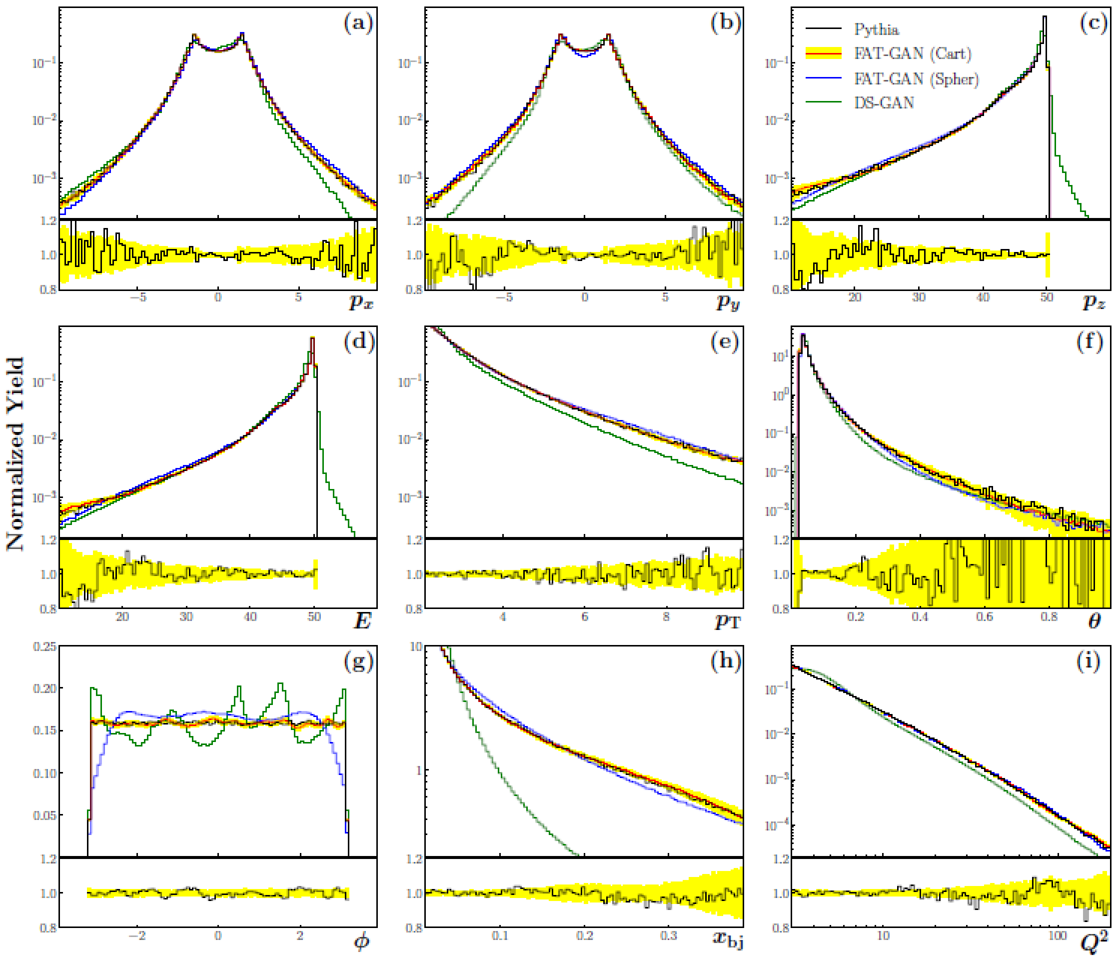
- **Representation in Cartesian Coordinates [FAT-GAN (Cart)]**



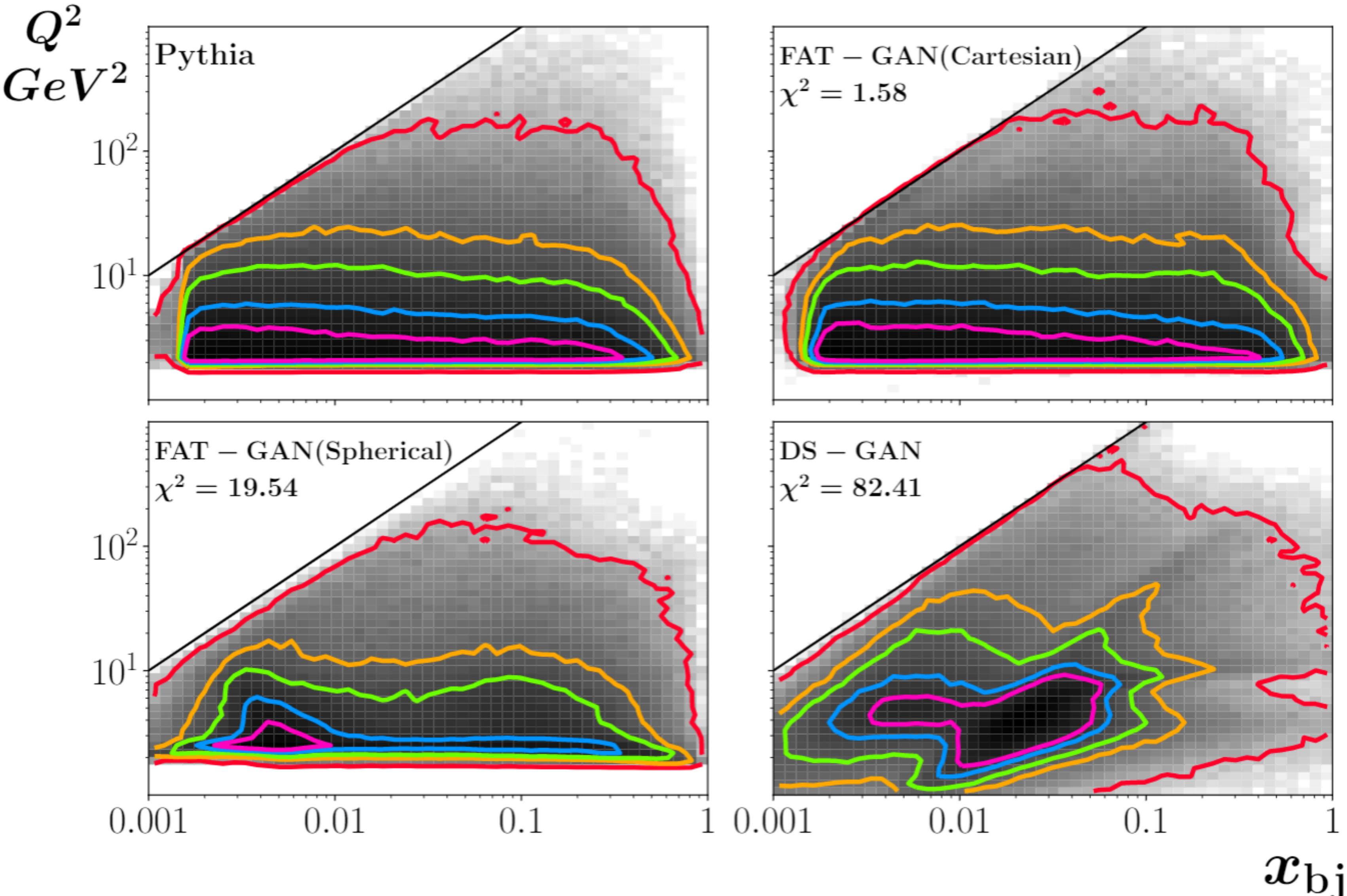
- **Representation in Spherical Coordinates [FAT-GAN (Spher)]**



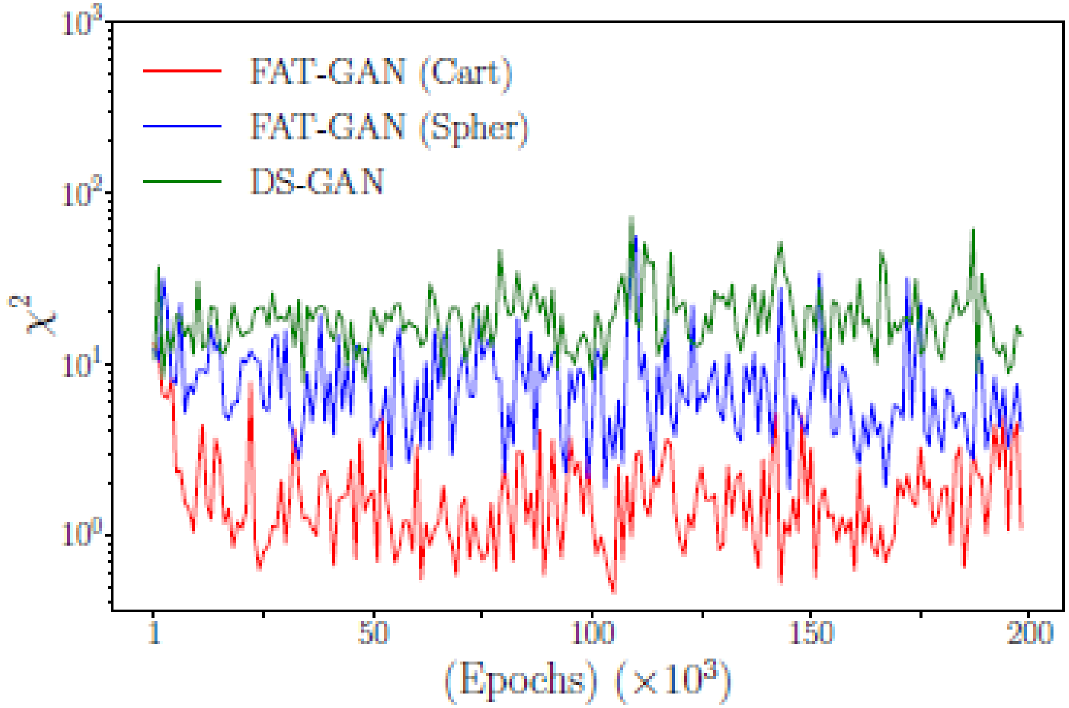
Distributions of Generated Physical Properties



Features Inter-correlations



Convergence Comparison



Summary

- **GAN**
  - Adversarial Training
  - Can mimic any data distributions
- **Mathematics under a GAN**
- **Problems of Training a GAN**
  - Attractive but Difficult to Train
- **FAT-GAN**
- **DCGAN**