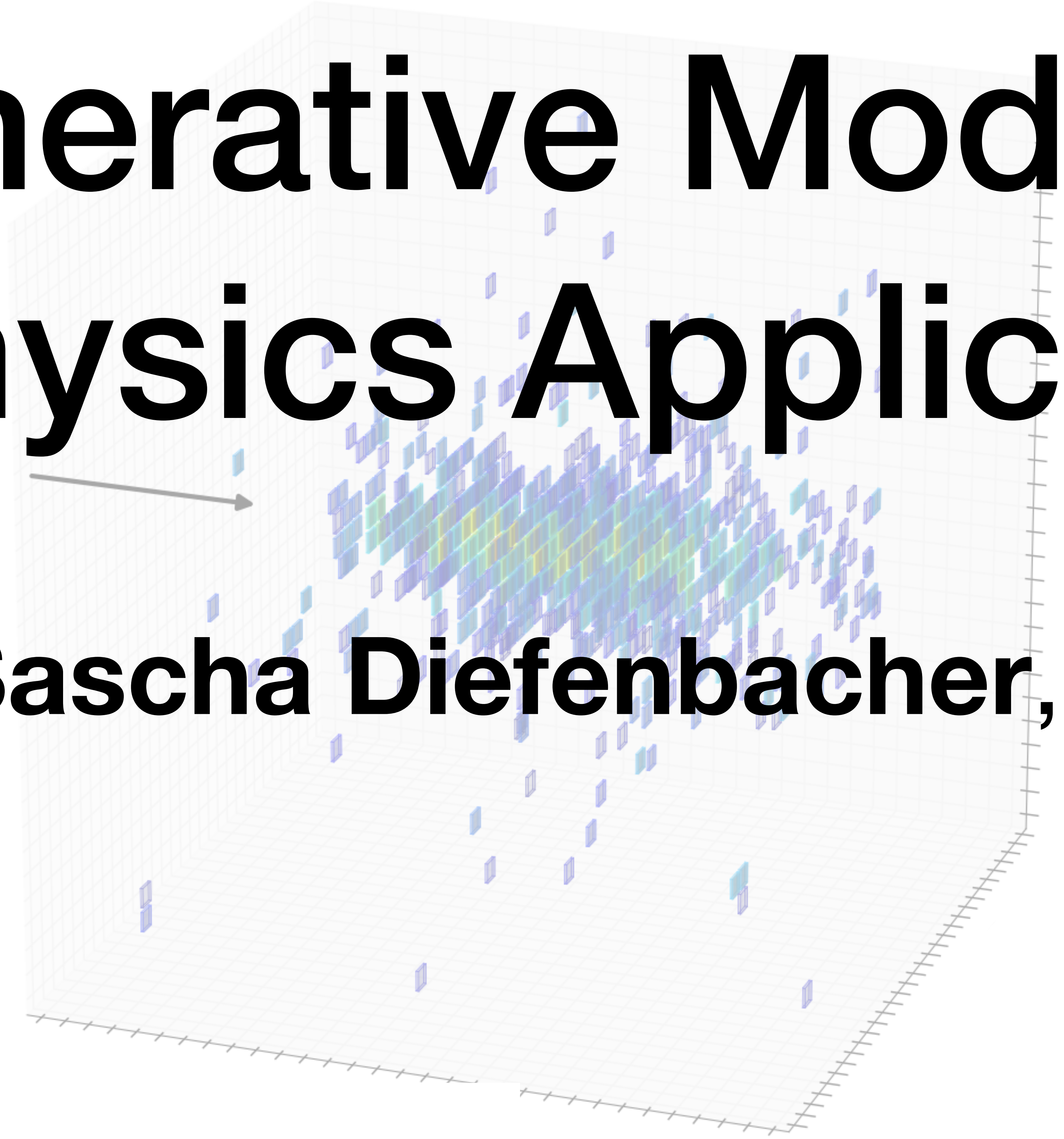
Generative Models and Physics Applications



Sascha Diefenbacher,

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

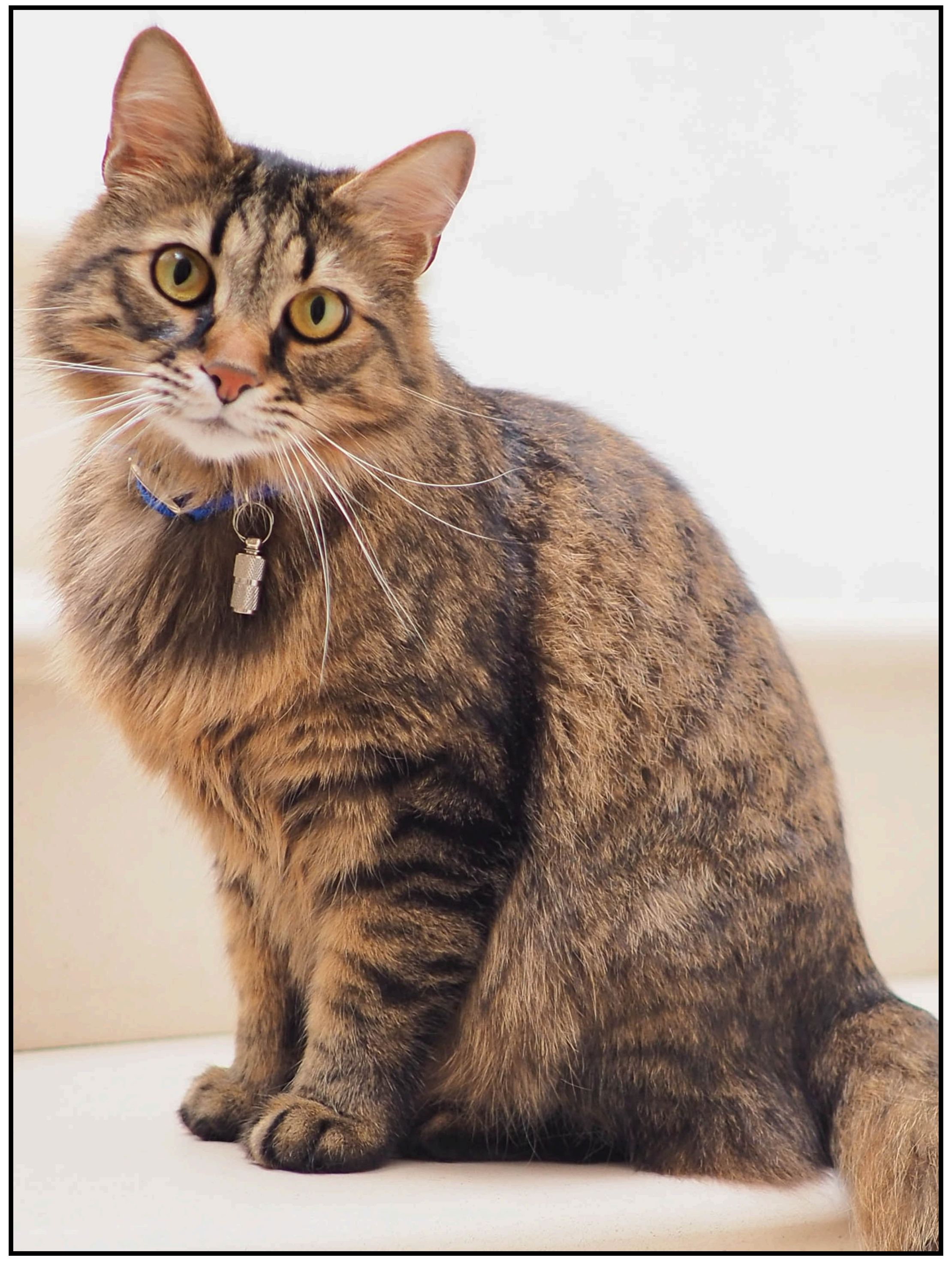
DESY.

Literature

- Further Reading:
  - Mehta et. al., *A high-bias, low-variance introduction to Machine Learning for physicists*, 1803.08823
  - Goodfellow et. al., *Generative Adversarial Networks* 1406.2661
  - Arjovsky et. al., *Wasserstein GAN*, 1701.07875
  - Gulrajani et. al., *Improved Training of Wasserstein GANs 1704.00028*

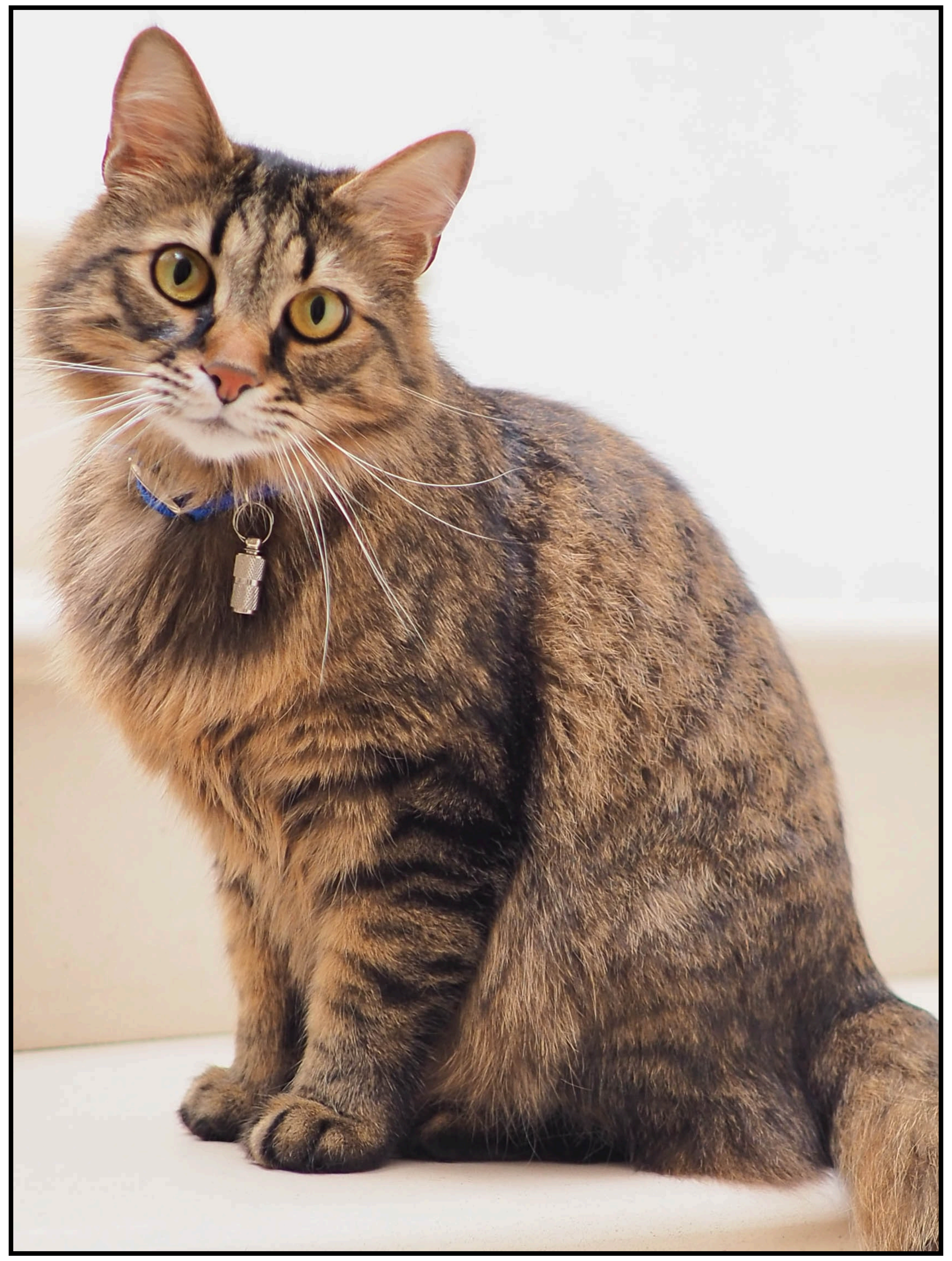
Classification

High Dim. Data



Classification

High Dim. Data | **Low Dim. Label**

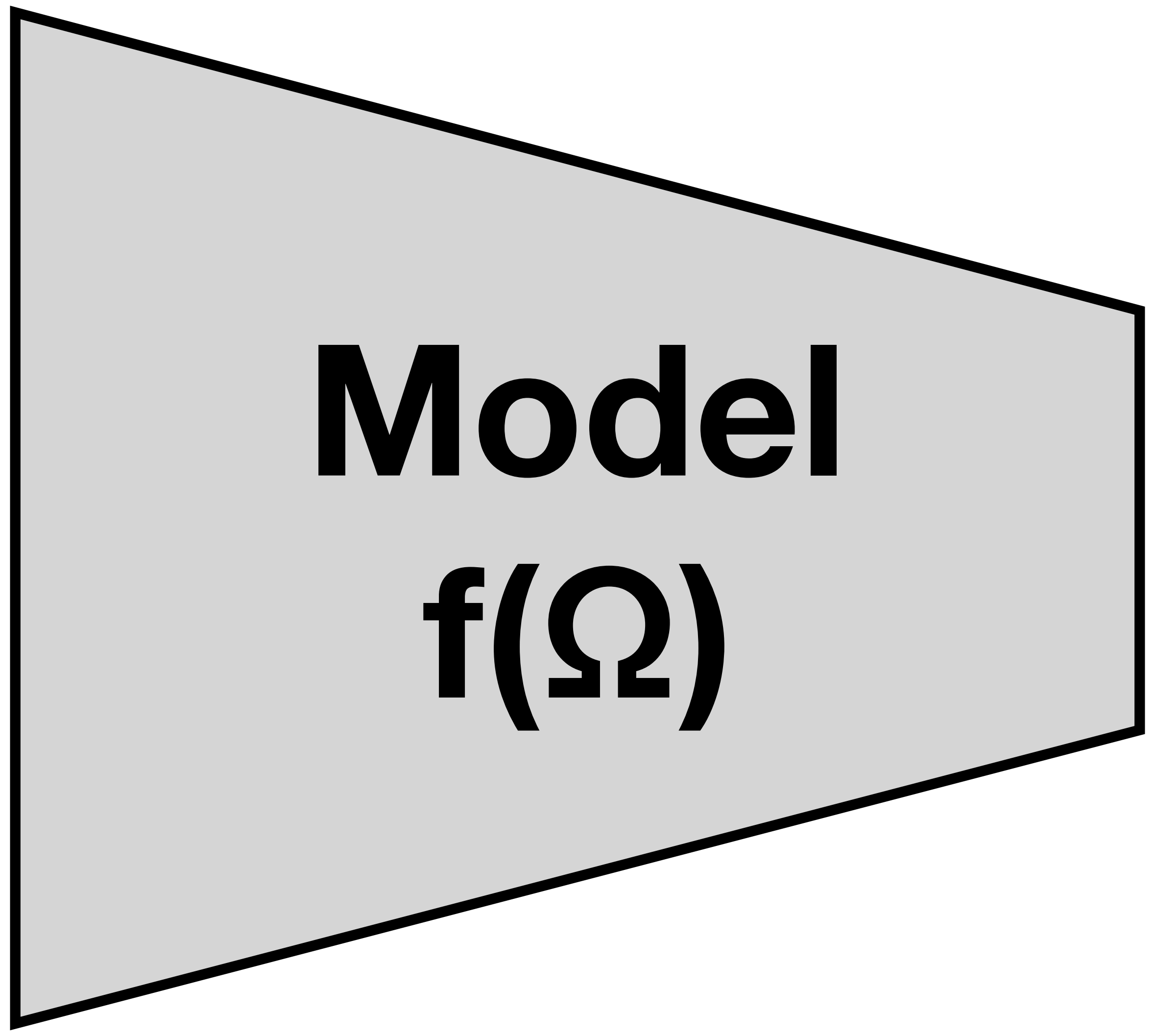


Dog

Cat

Classification

High Dim. Data



Classification

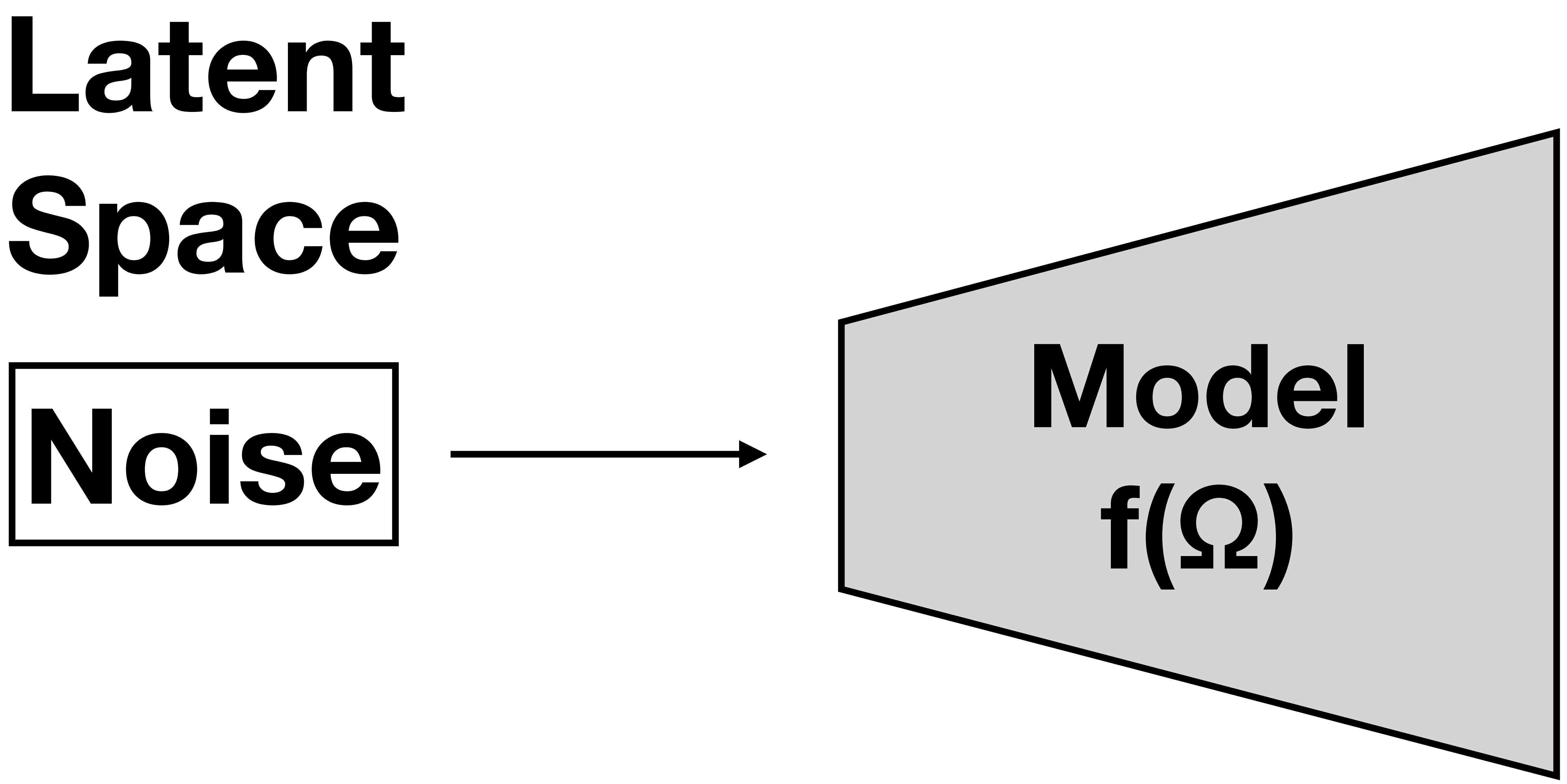
High Dim. Data

Model
f(Ω)

Prediction

Cat/Dog?

Generation



Generation

Latent Space

High Dim. Data

Noise → **Model f(Ω)** → 

Choice of Loss Function

- Classification:
  - Easy to interpret model outputs
    - Model performance simple to measure
      - Accuracy, Area Under Curve, …

Choice of Loss Function

- Classification:
  - Easy to interpret model outputs
    - Model performance simple to measure
      - Accuracy, Area Under Curve, …
    - Wide range of available loss functions
      - Only needs to compare predictions with true labels
      - Common example: Cross-Entropy

Choice of Loss Function

- Generative:
  - How do you measure the model performance?
  - How is this expressed mathematically (and differentiable)

Choice of Loss Function

- Generative:
  - How do you measure the model performance?
  - How is this expressed mathematically (and differentiable)

http://thesecatsdonotexist.com

Generation Difficulties

- Image Set:
  - Does the new set have the same properties as the data?

Training Data:

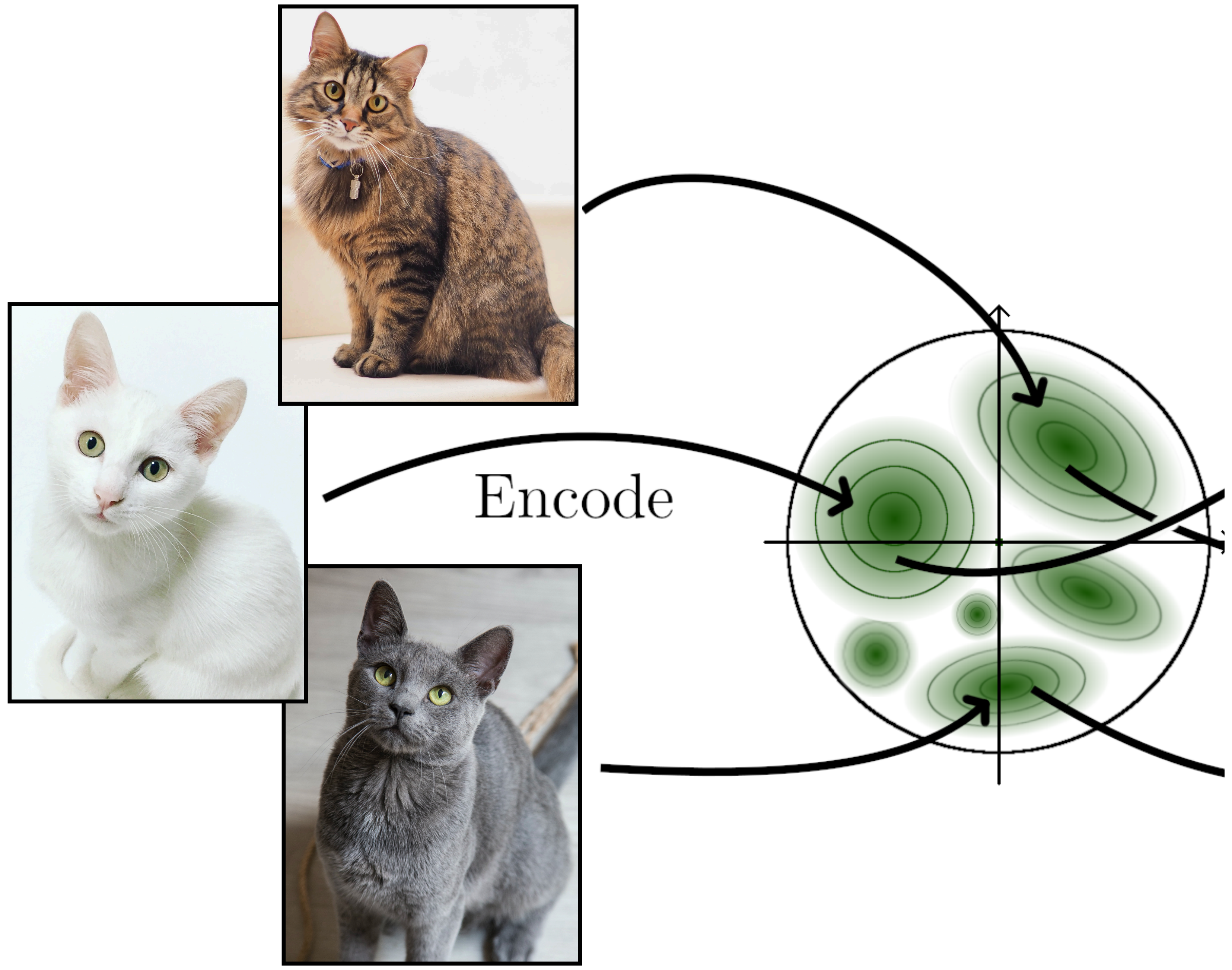
Generation Difficulties

- Image Set:
  - Does the new set have the same properties as the data?

Training Data:

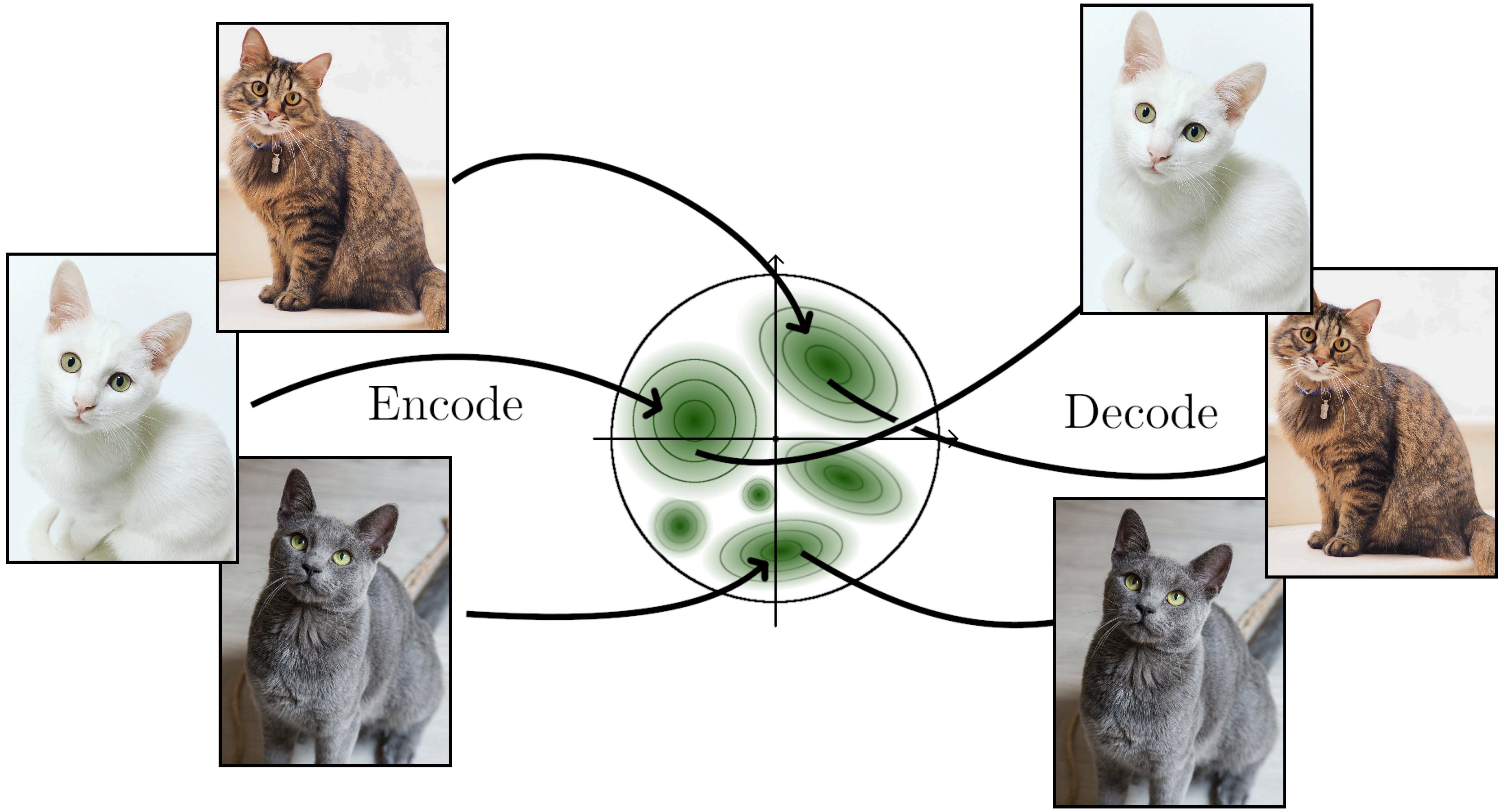
Generated Data:

AutoEncoder



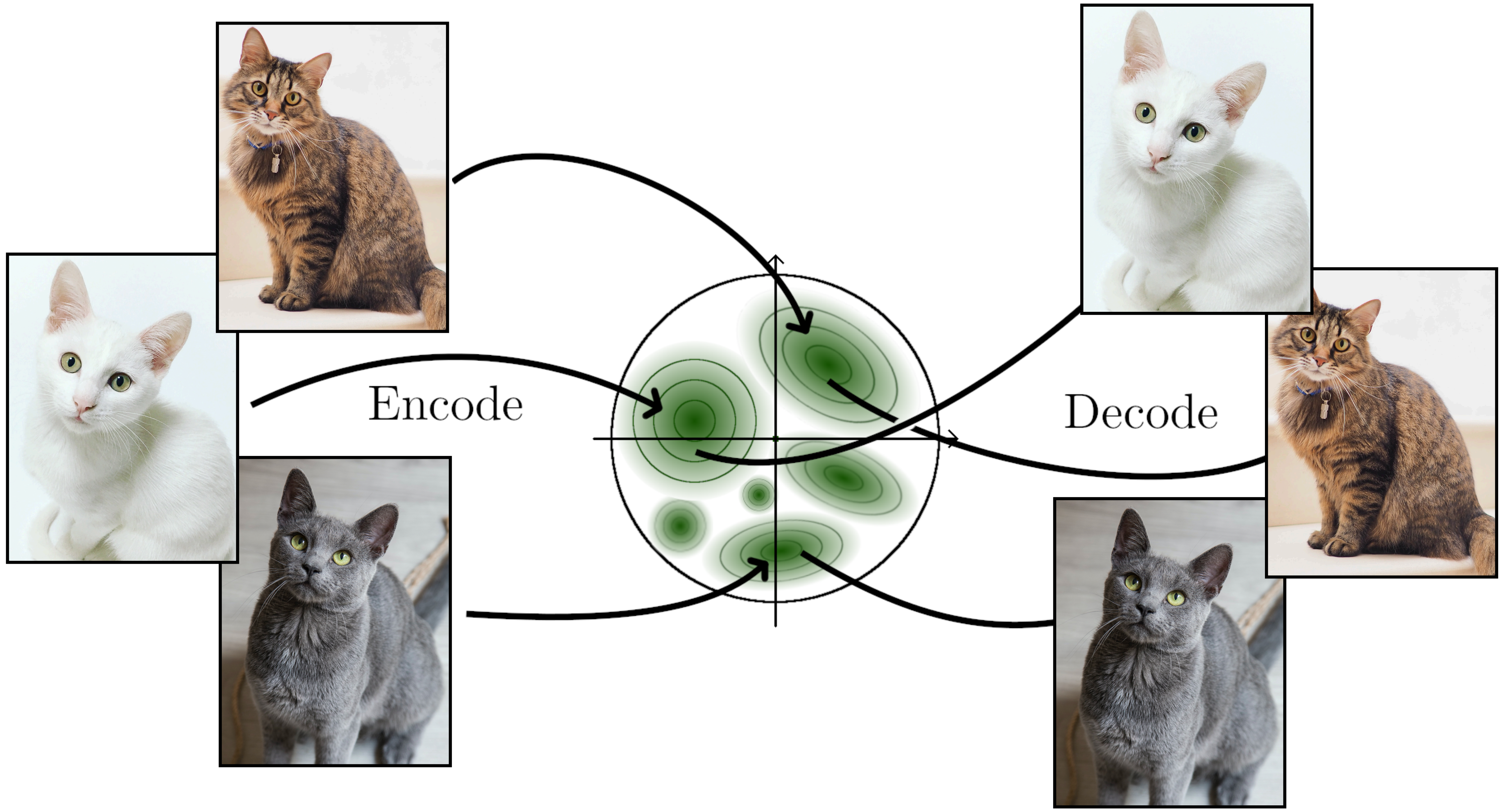
- Encoding function E(x)=z map high dimensional data X to low dimensional latent space Z

AutoEncoder



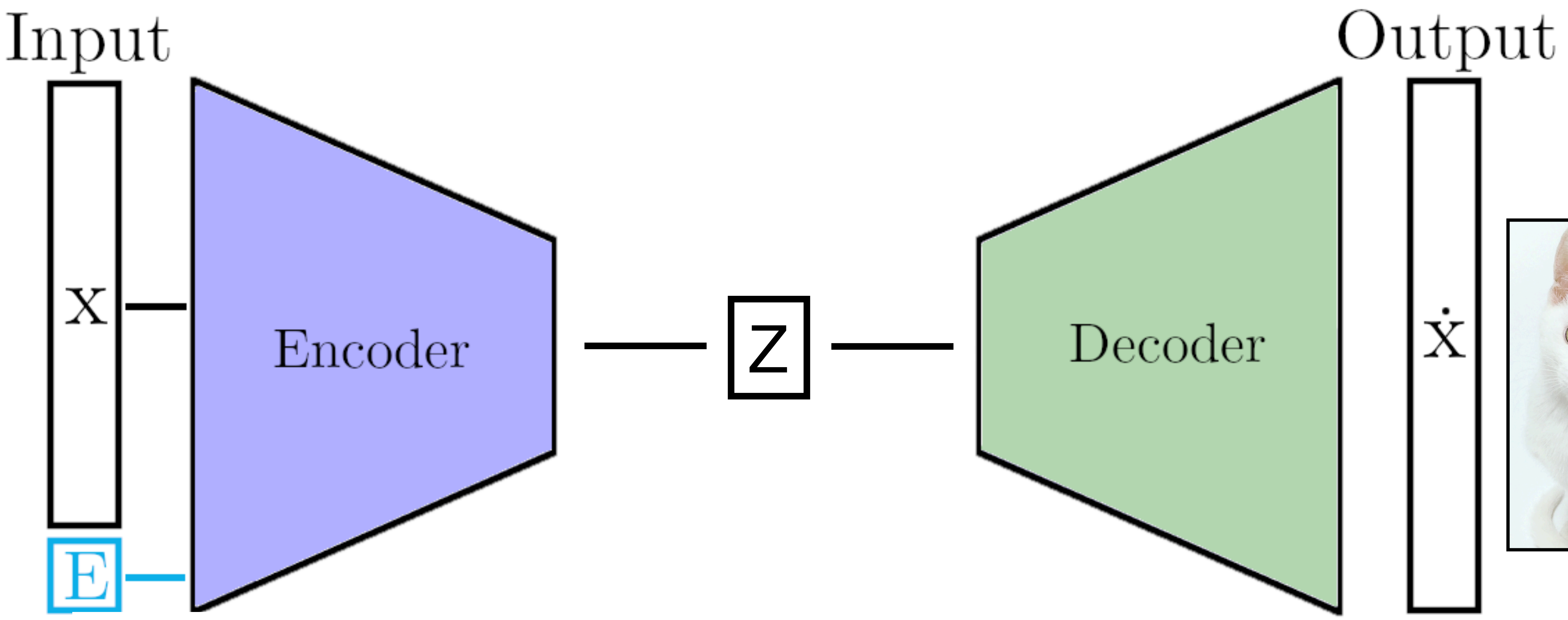
- Encoding function E(x)=z map high dimensional data X to low dimensional latent space Z
- Decoding function D(z)=x map latent space Z back to data X

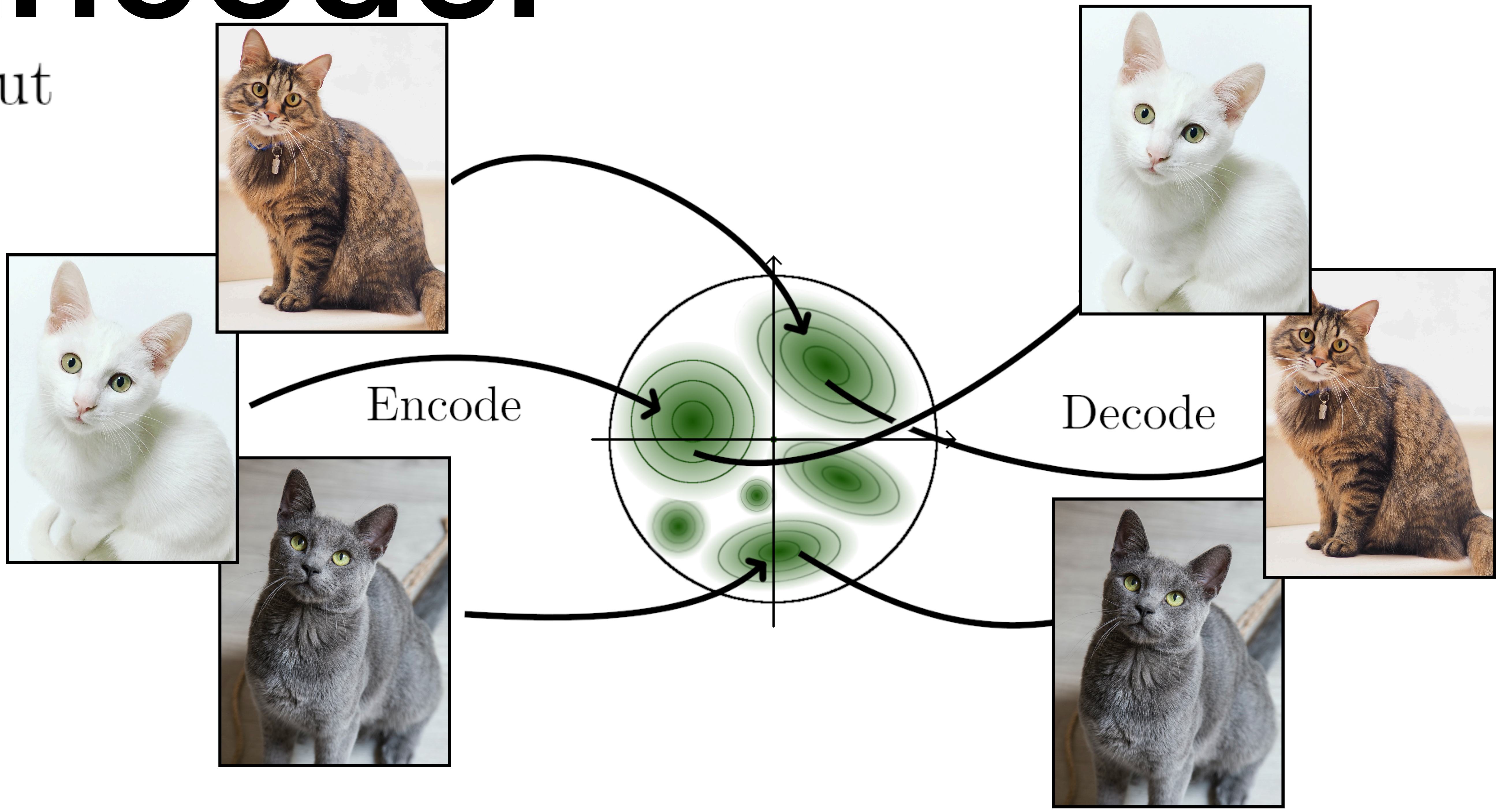
AutoEncoder



- Encoding function E(x)=z map high dimensional data X to low dimensional latent space Z
- Decoding function D(z)=x map latent space Z back to data X
- Compare Input and Output pixel by pixel with mean squared error

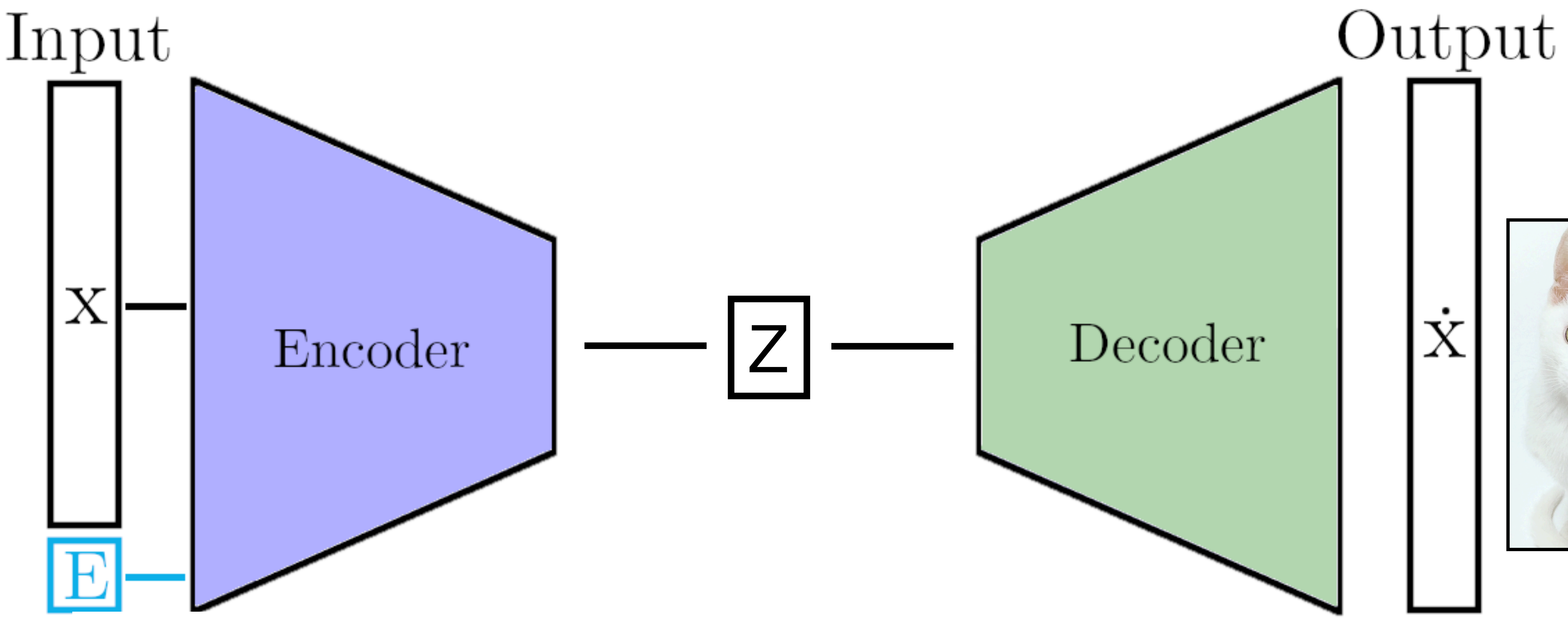
AutoEncoder

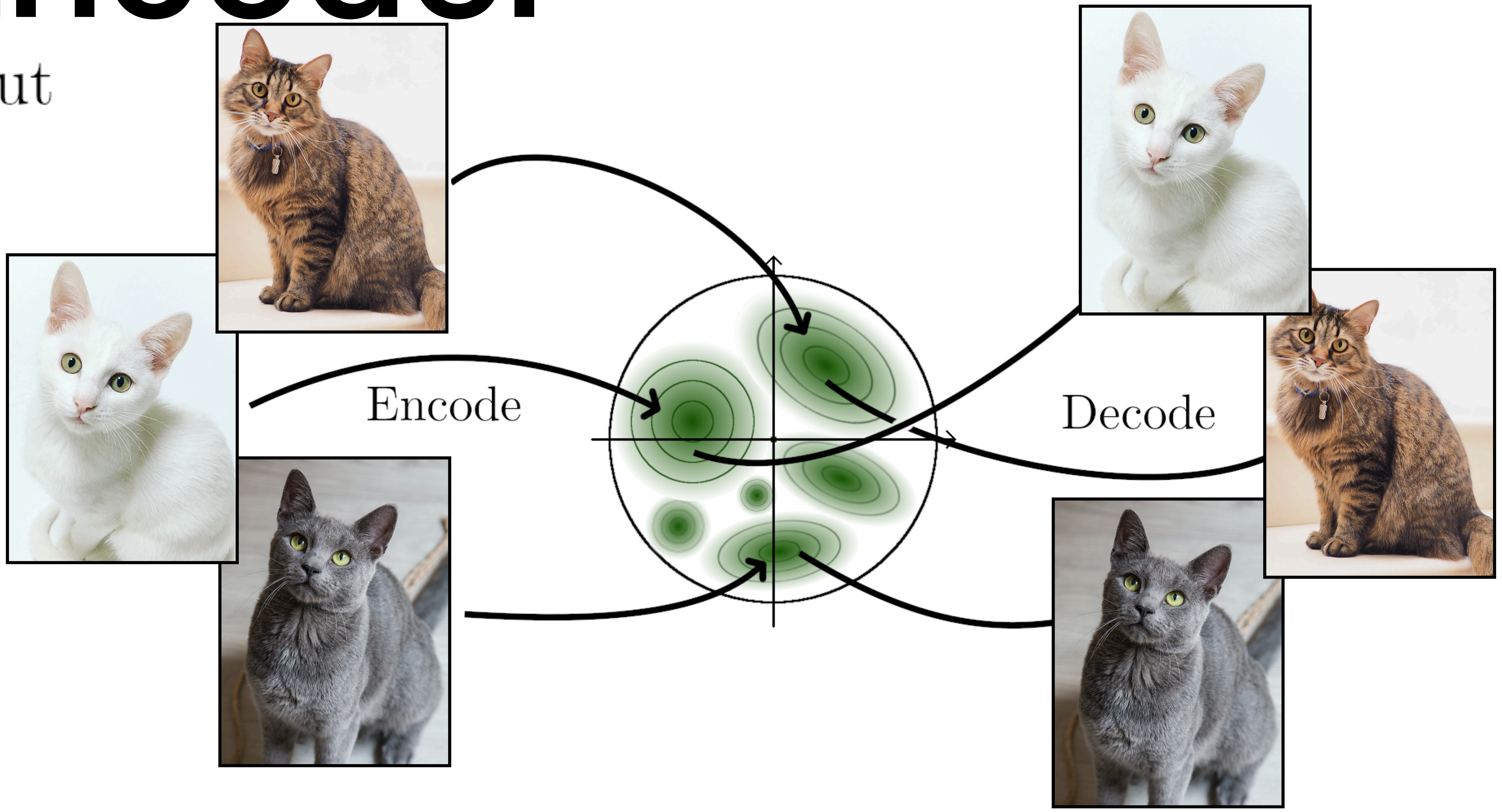




- Sample for Z and pass it to D(Z) → Generate new samples

AutoEncoder

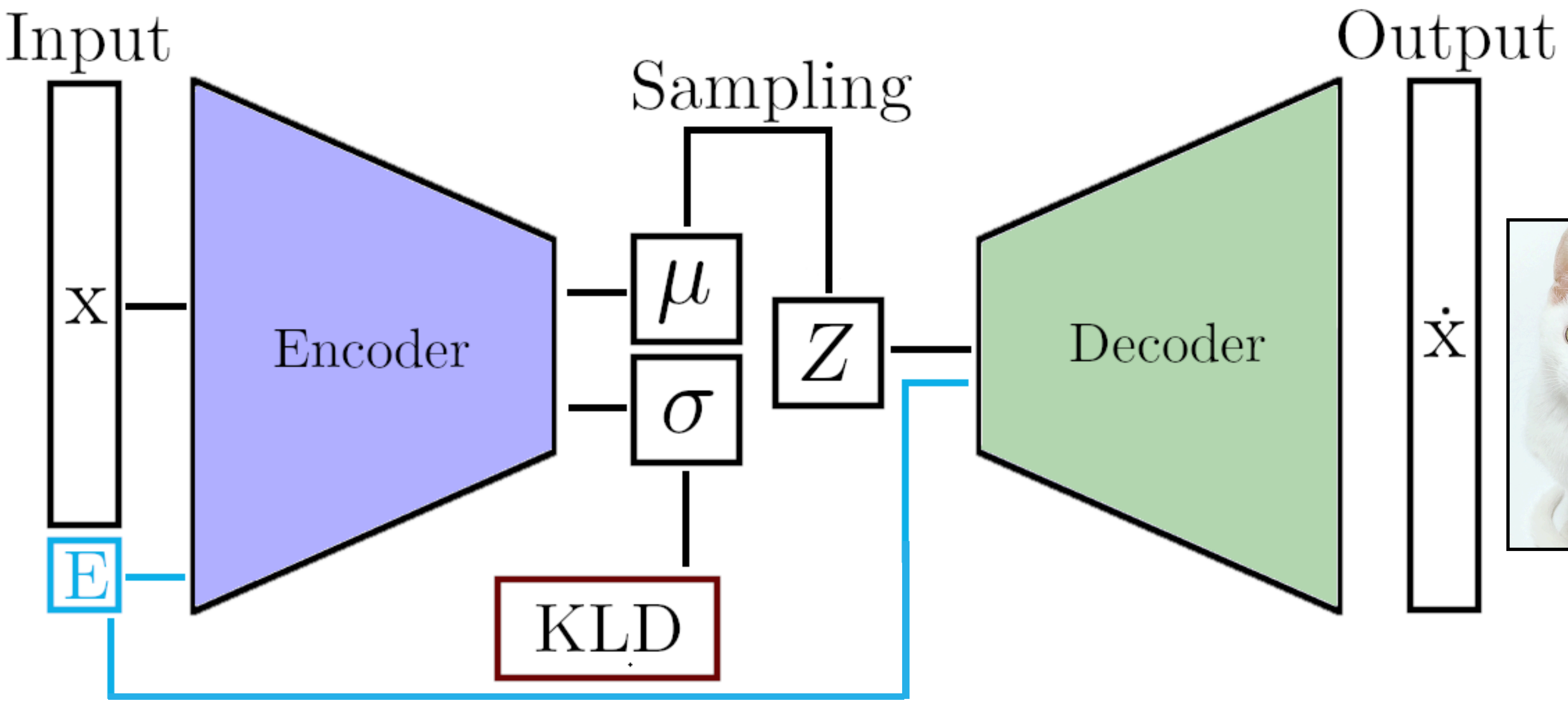


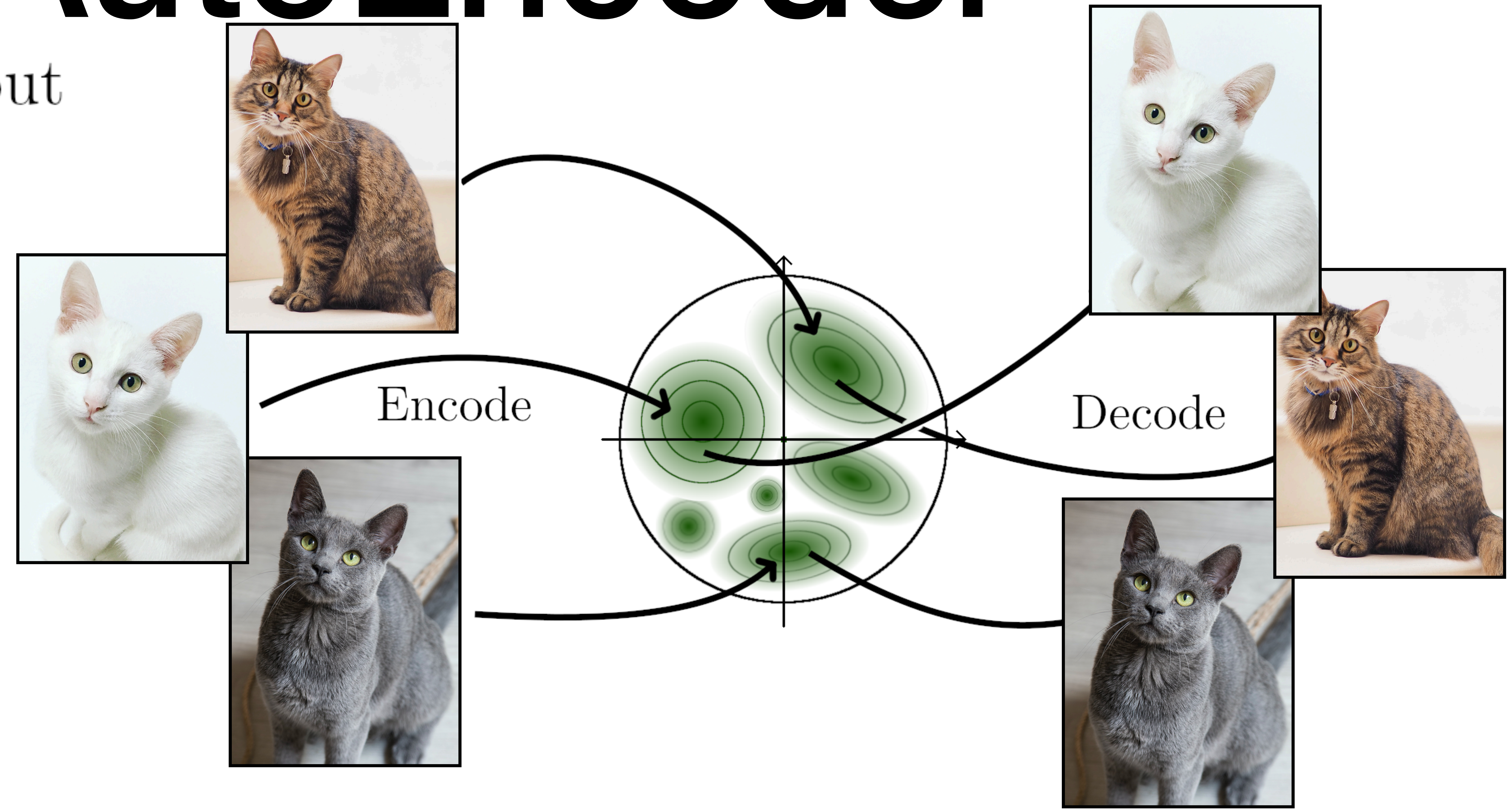


- Sample for Z and pass it to D(Z) ➡ Generate new samples
- Problem: Need regularised later space to sample form

➡ Variational AutoEncoder

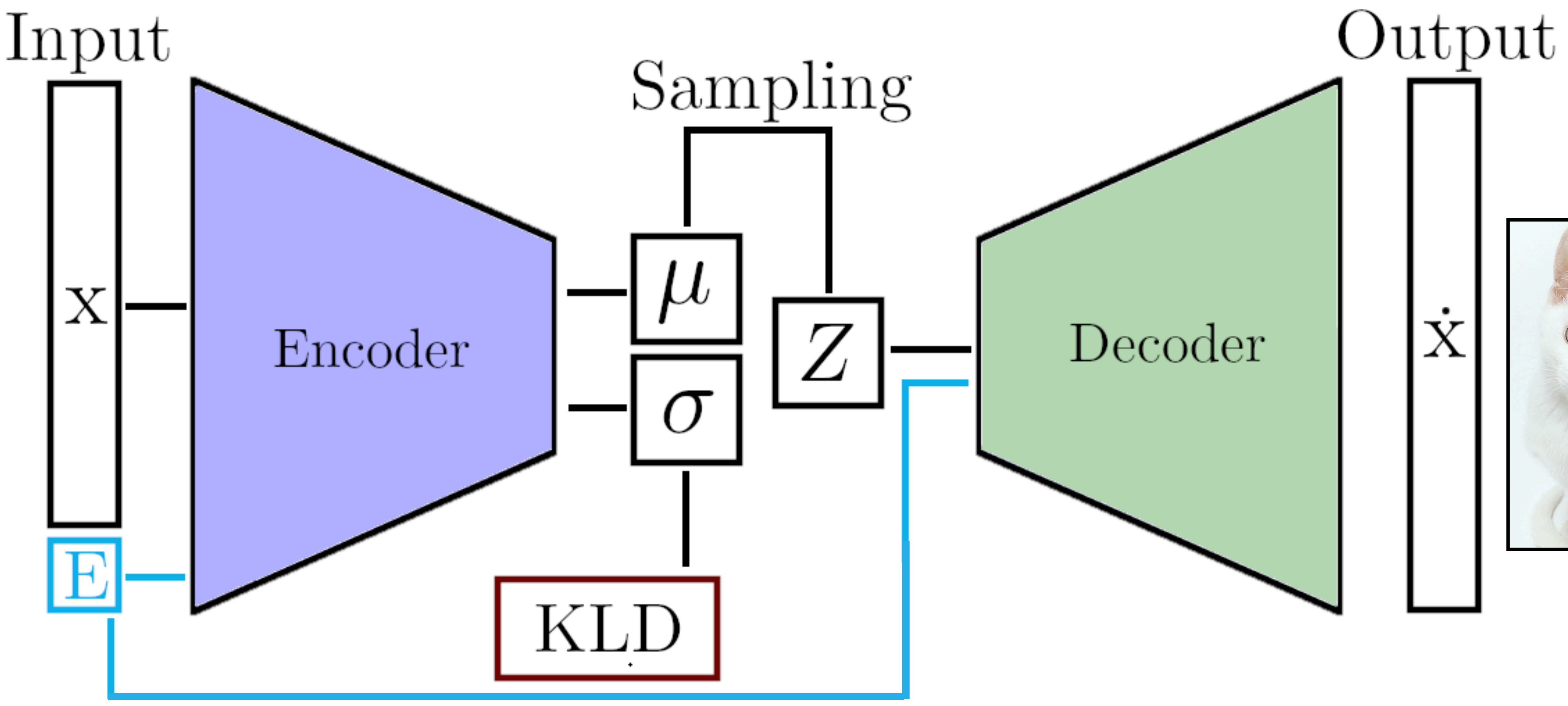
Variational AutoEncoder

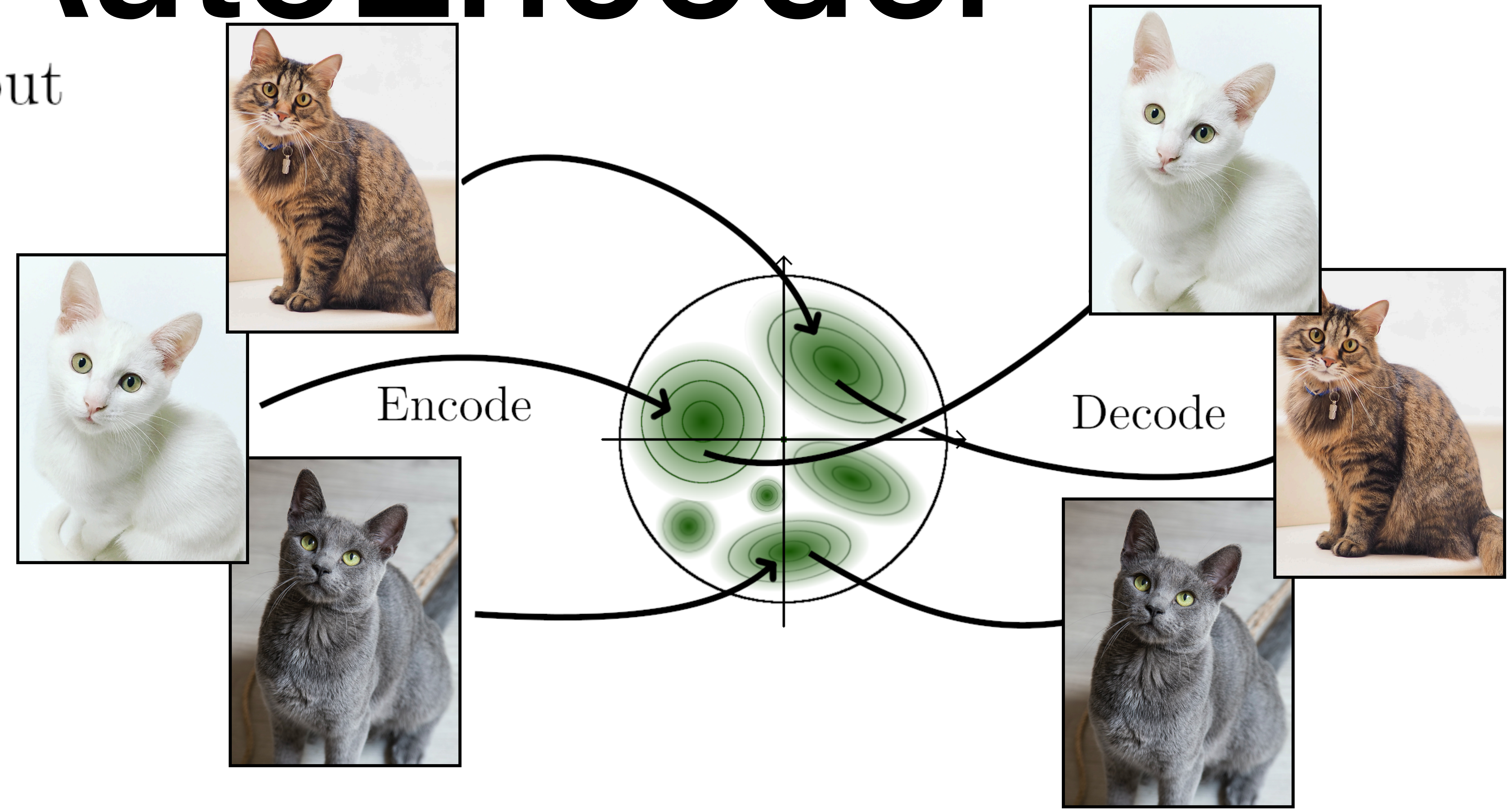




- Latent space: Series of Gaussians, regularised match N(μ=0, σ=1)

Variational AutoEncoder

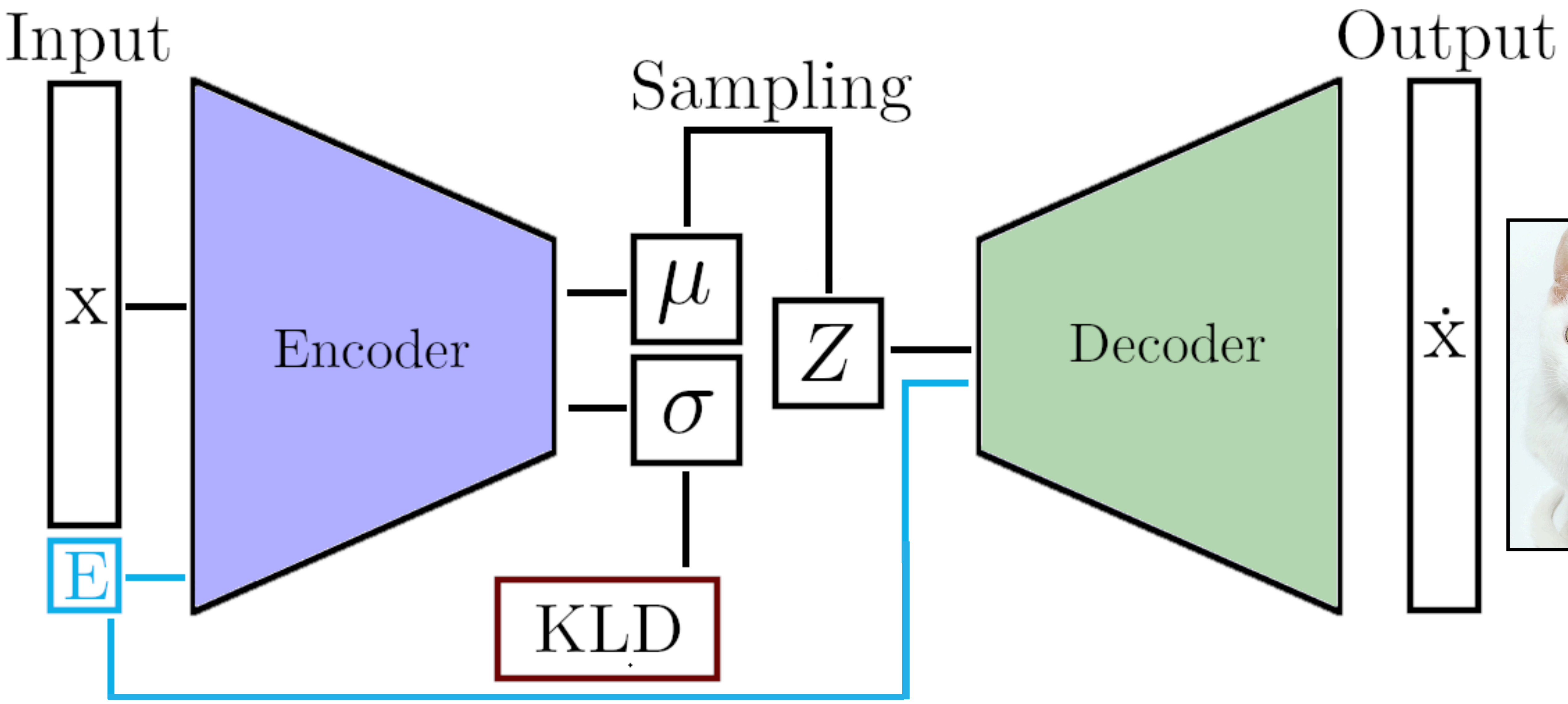


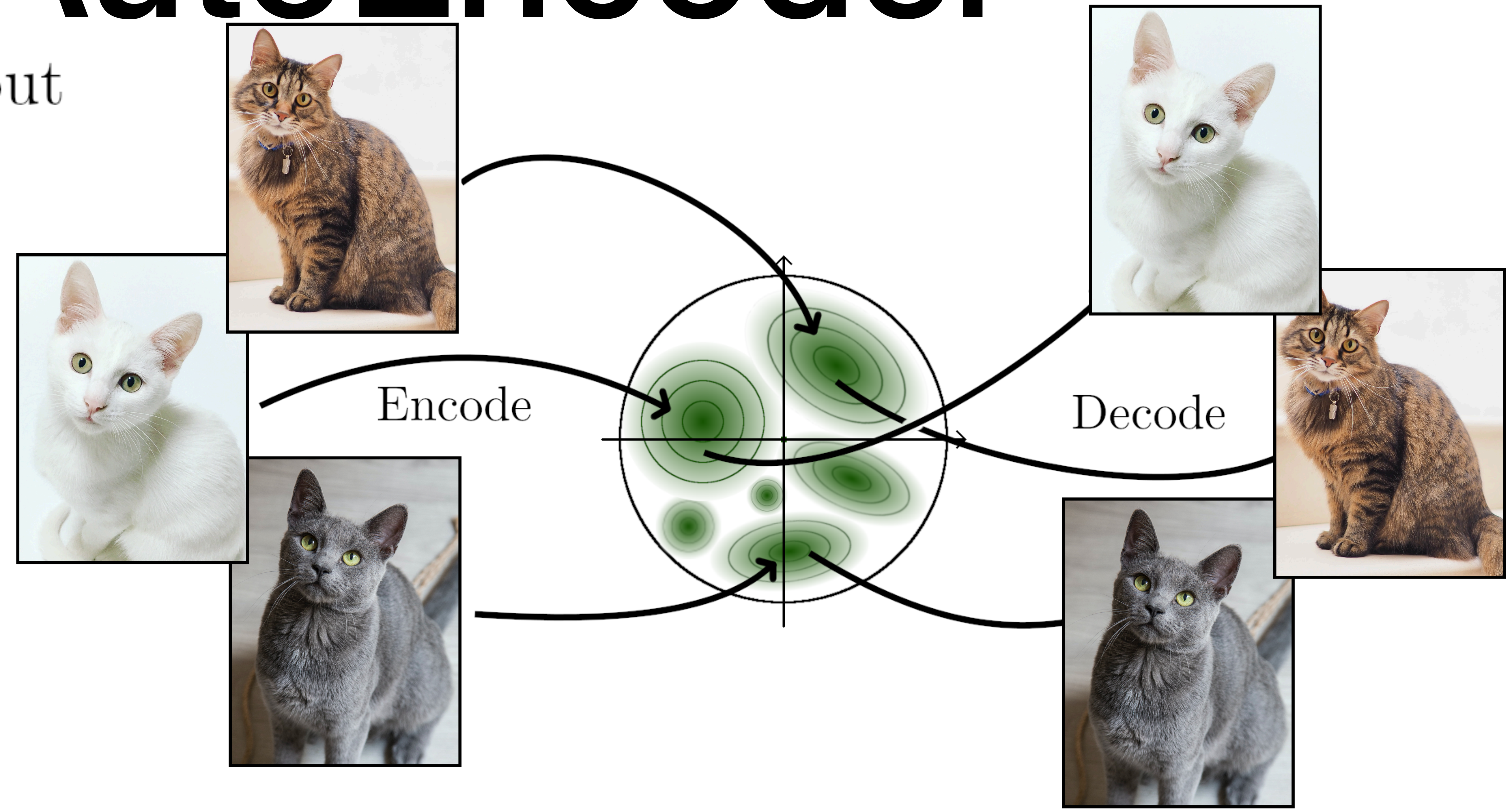


- Latent space: Series of Gaussians, regularised match N(μ=0, σ=1)
  - Using Gaussians lets us use Kullback–Leibler divergence

$$\sum_{i=1}^{n} \sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1$$

Variational AutoEncoder



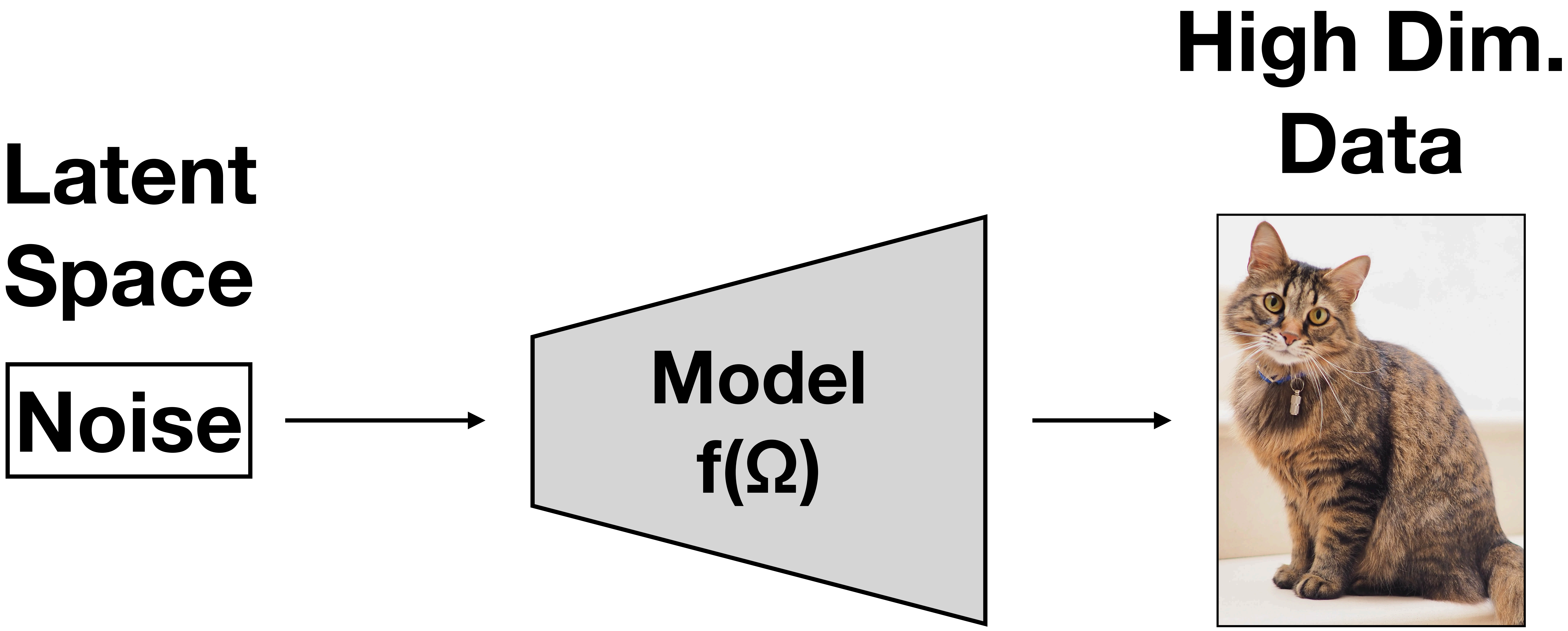


- Latent space: Series of Gaussians, regularised match N(μ=0, σ=1)
  - Using Gaussians lets us use Kullback–Leibler divergence

$$\sum_{i=1}^{n} \sigma_i^2+\mu_i^2-\log(\sigma_i)-1$$

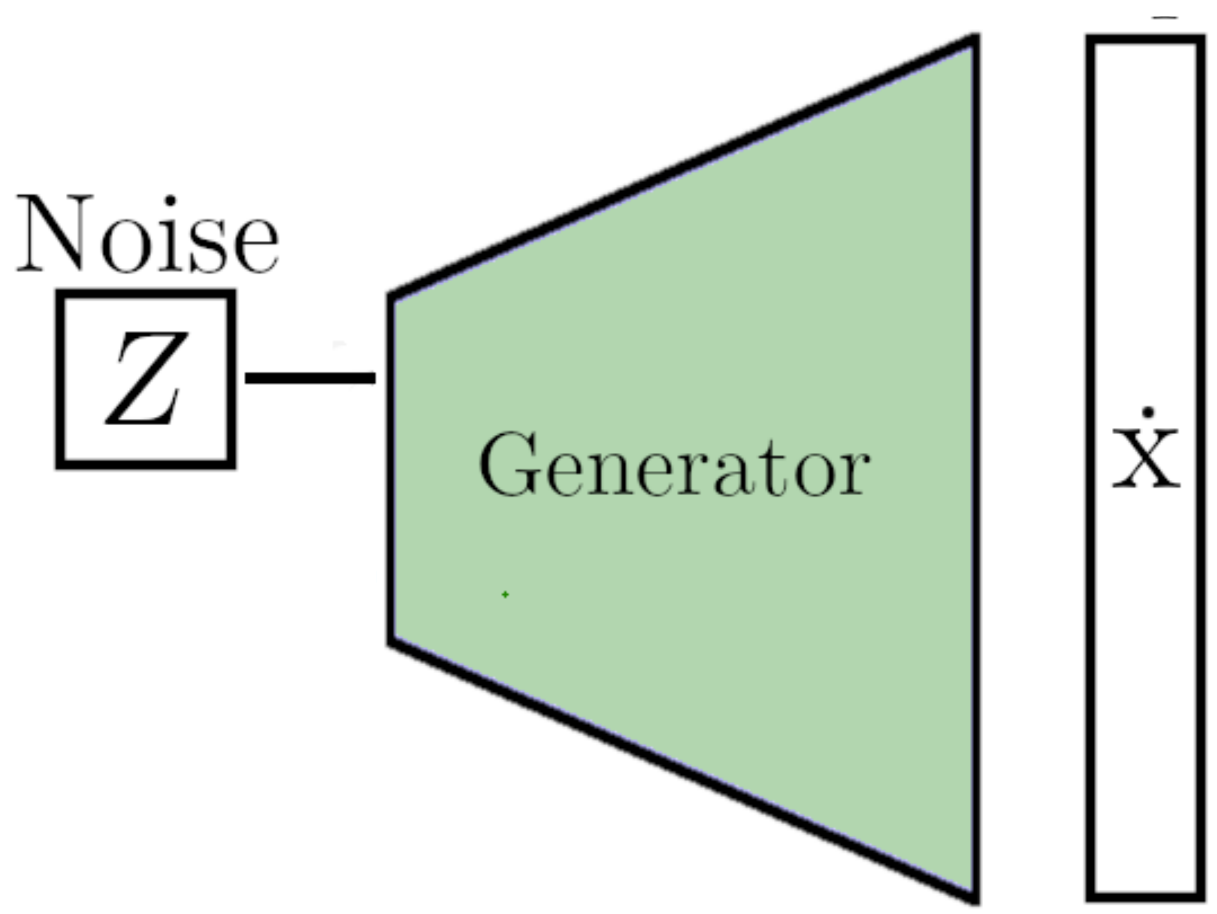
- Compare Input and Output again using MSE

Generative Adversarial Network



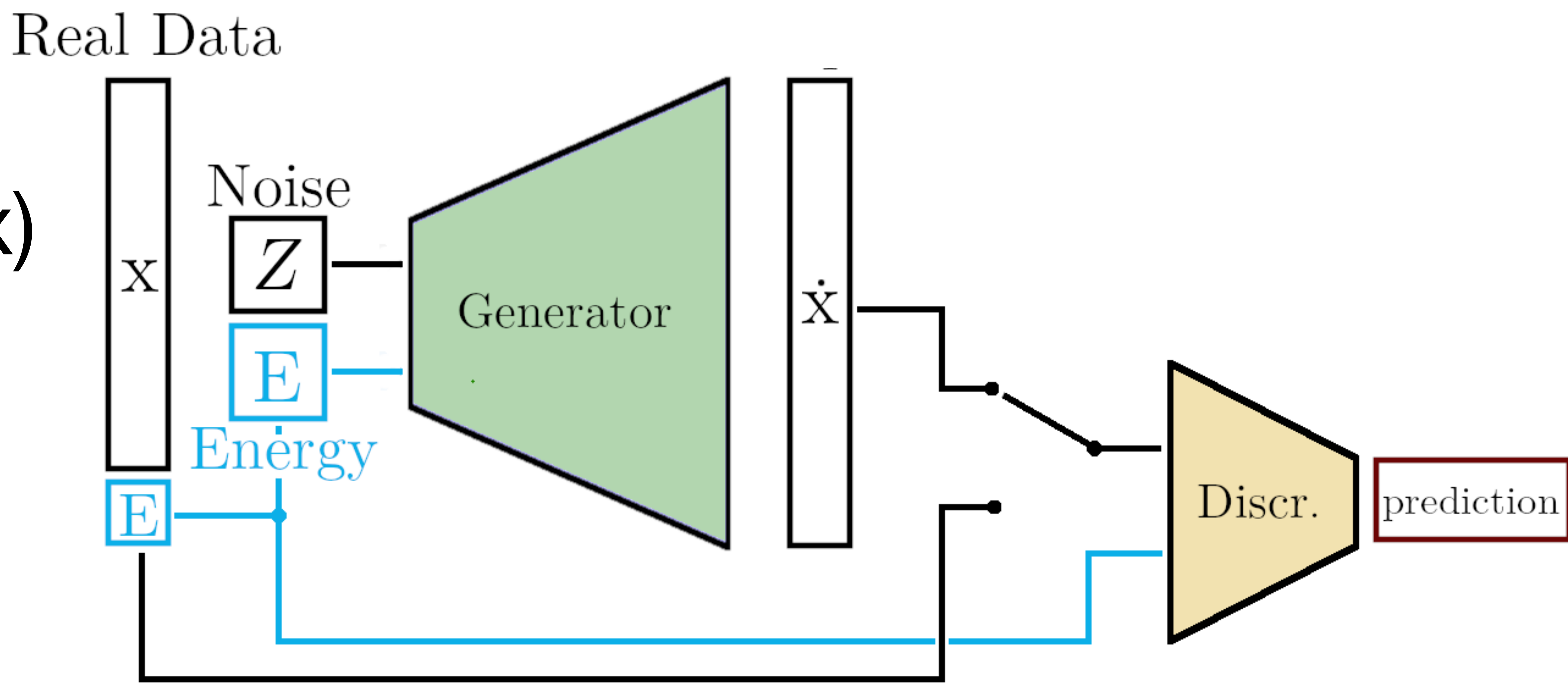
Generative Adversarial Network

- Generator Network G(z)=x
  - Maps noise Z to Data X



Generative Adversarial Network

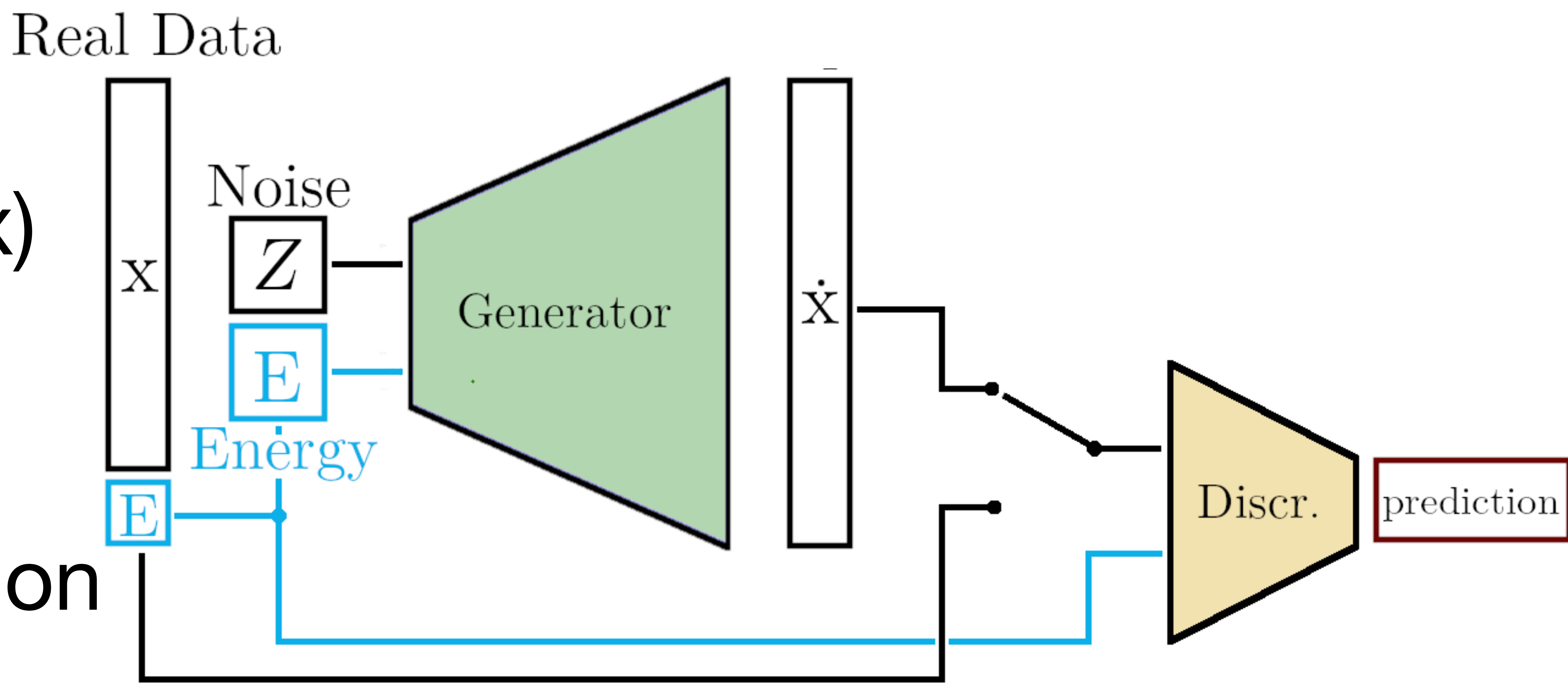
- Generator Network G(z)=x
  - Maps noise Z to Data X
- Discriminator D(G(z)) and D(x)
  - Learns difference between real and fake



$$L = \mathrm{BCE}(p(\mathrm{real}), 1) + \mathrm{BCE}(p(\mathrm{fake}), 0)$$

Generative Adversarial Network

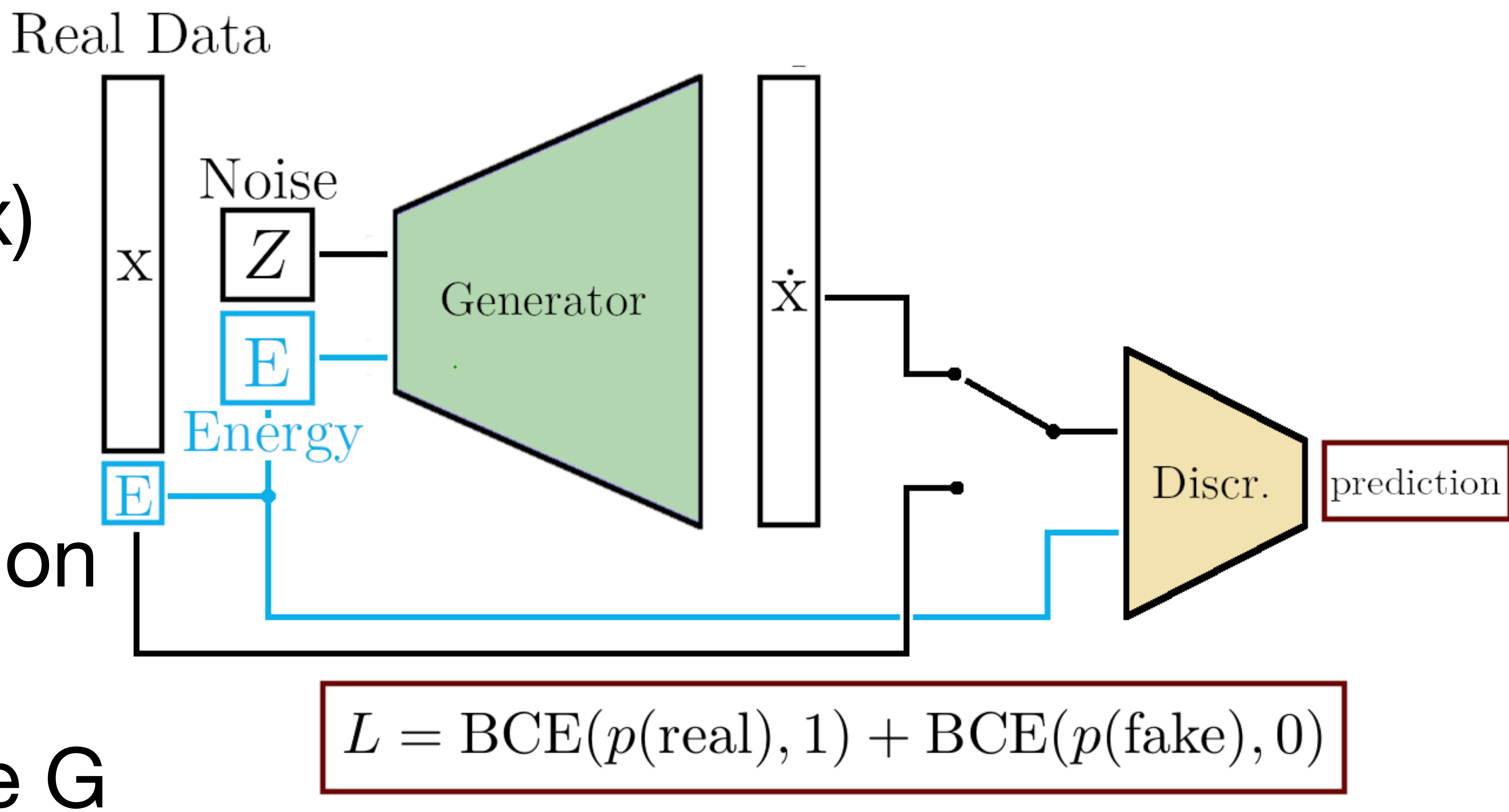
- Generator Network G(z)=x
  - Maps noise Z to Data X
- Discriminator D(G(z)) and D(x)
  - Learns difference between real and fake
- D(G(z)) is differentiable function measuring performance



$$L = \mathrm{BCE}(p(\mathrm{real}), 1) + \mathrm{BCE}(p(\mathrm{fake}), 0)$$

Generative Adversarial Network

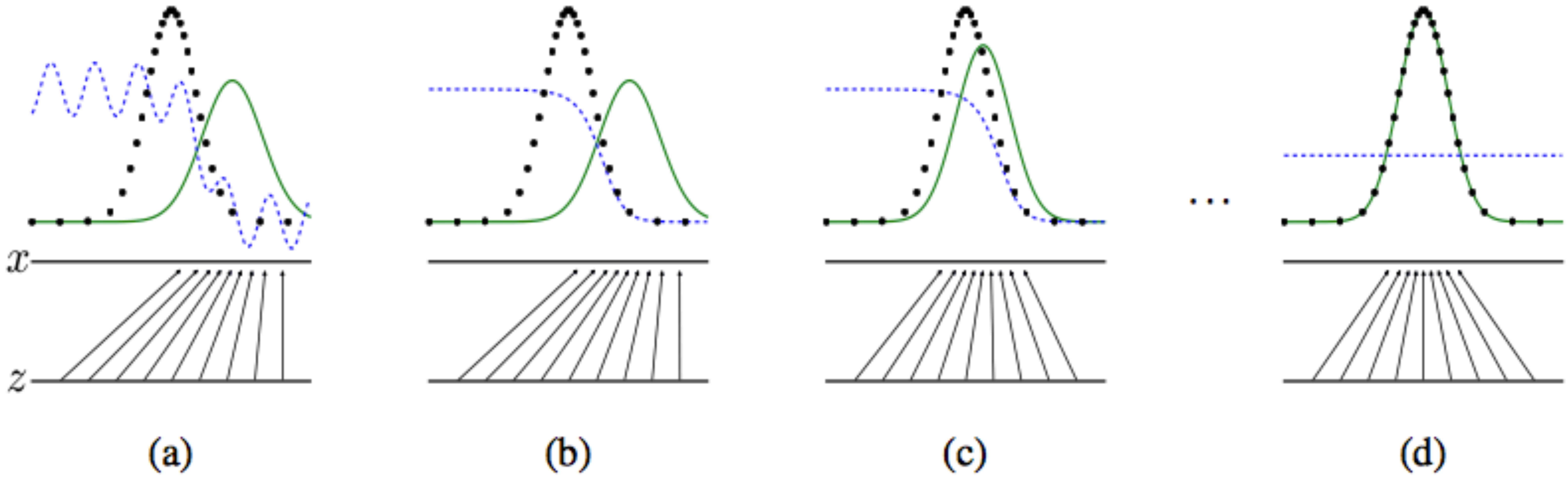
- Generator Network G(z)=x
  - Maps noise Z to Data X
- Discriminator D(G(z)) and D(x)
  - Learns difference between real and fake
- D(G(z)) is differentiable function measuring performance
- Use D(G(z)) as loss to update G



Generative Adversarial Network

Code Example

Generative Adversarial Network

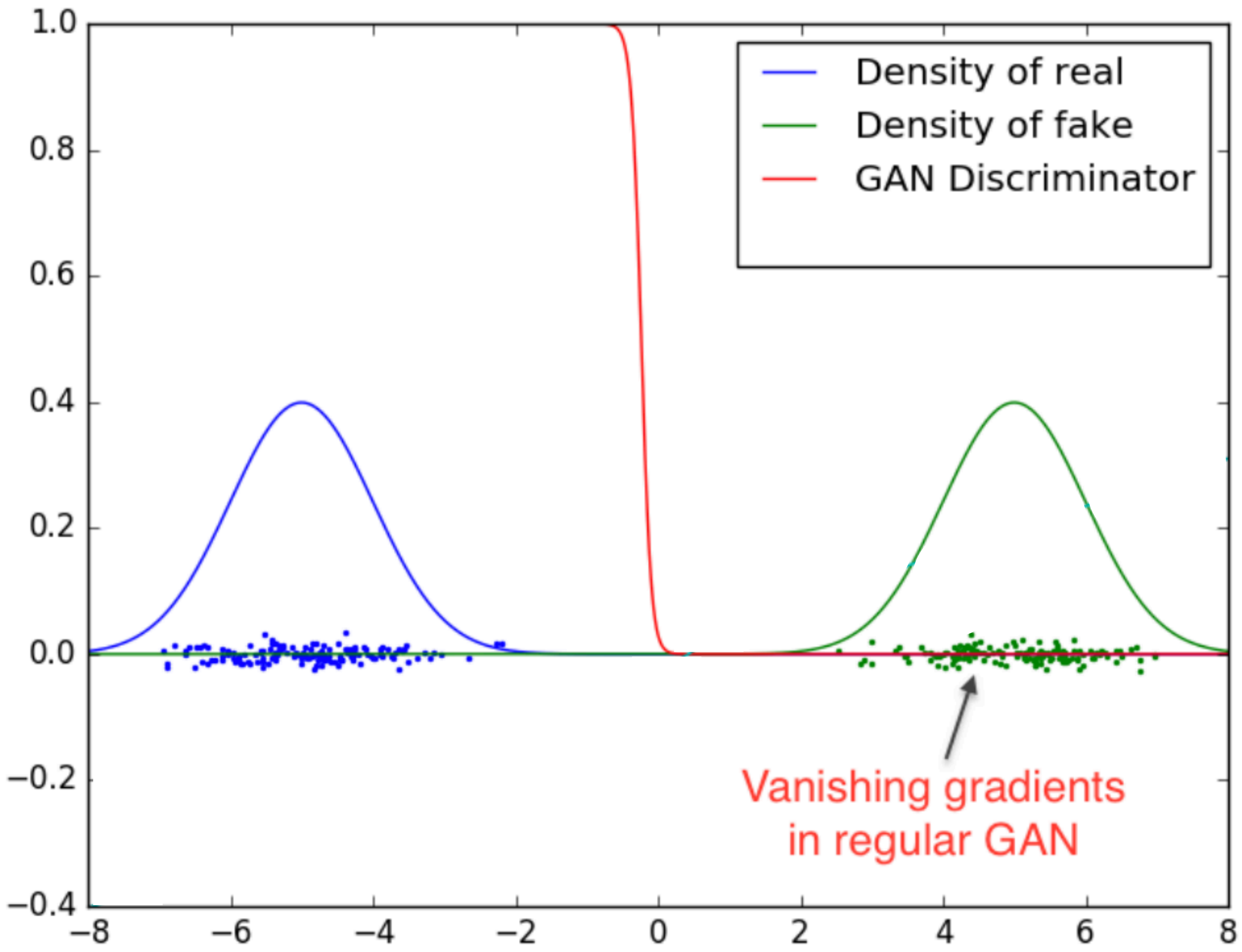


Goodfellow et al.- arXiv:1406.2661

- Discriminator of GAN approximates Jensen Shannon Divergence
- Guides Generated distribution to match real distribution

Wasserstein GAN

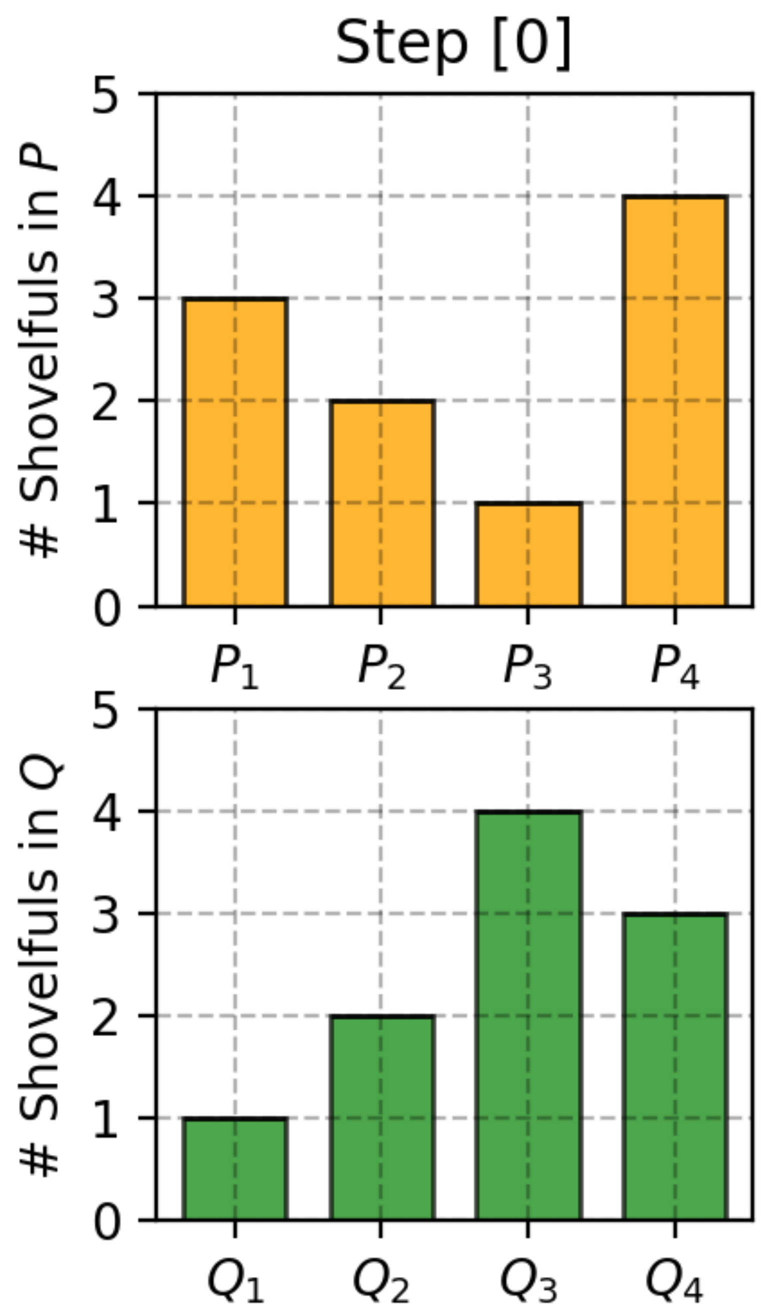
- JSD only based on overlap
  - Problem for very separated distributions
  - Vanishing Gradient
- Alternative to JSE: Earth Mover Distance (aka Wasserstein Distance)



Arjovsky et al.- arXiv:1701.07875

Wasserstein GAN

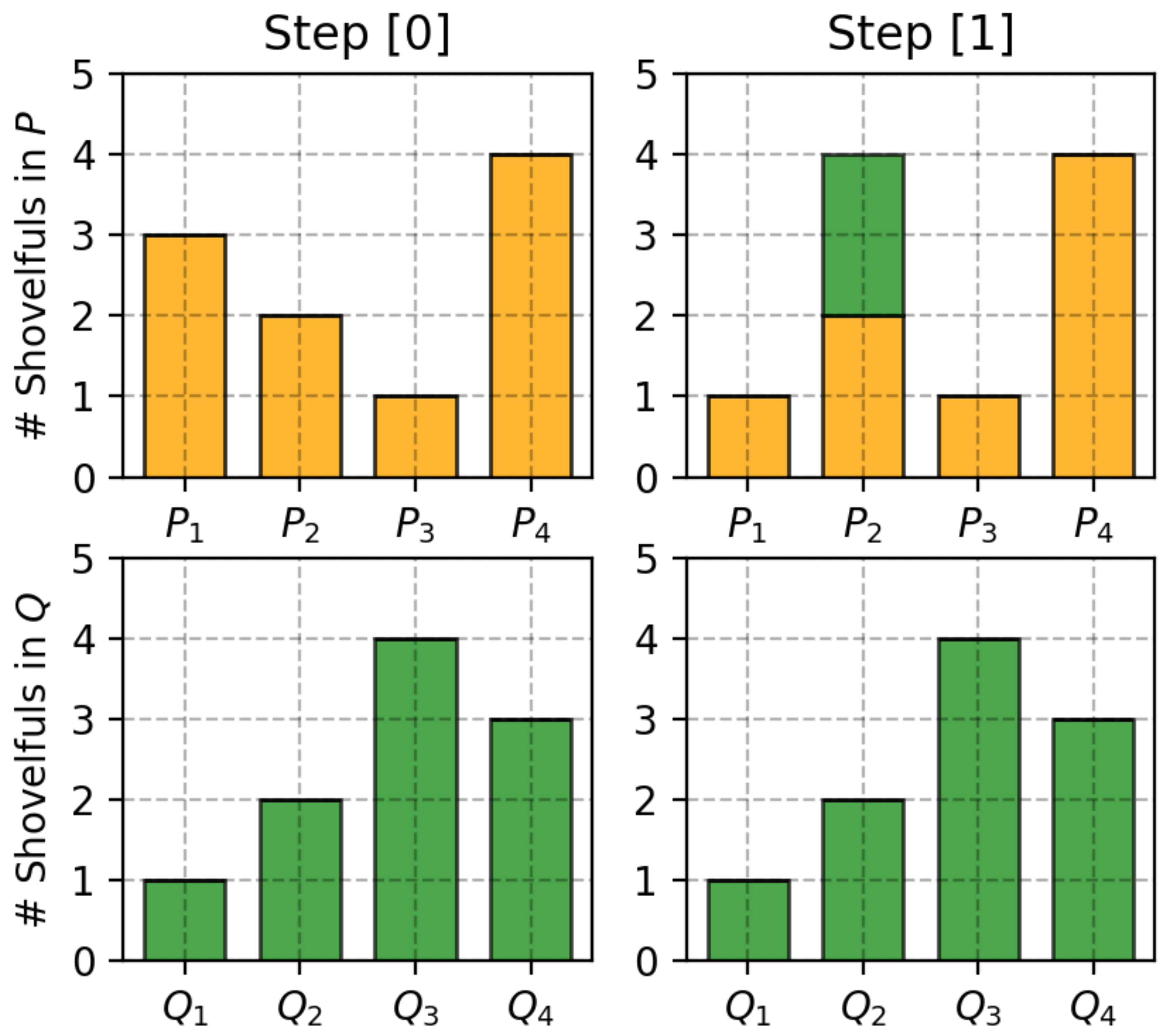
- Earth Mover Distance
- Optimal transport problem
- Most energy efficient way to match two distributions
  - Energy defined as (mass times distance)



https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html

Wasserstein GAN

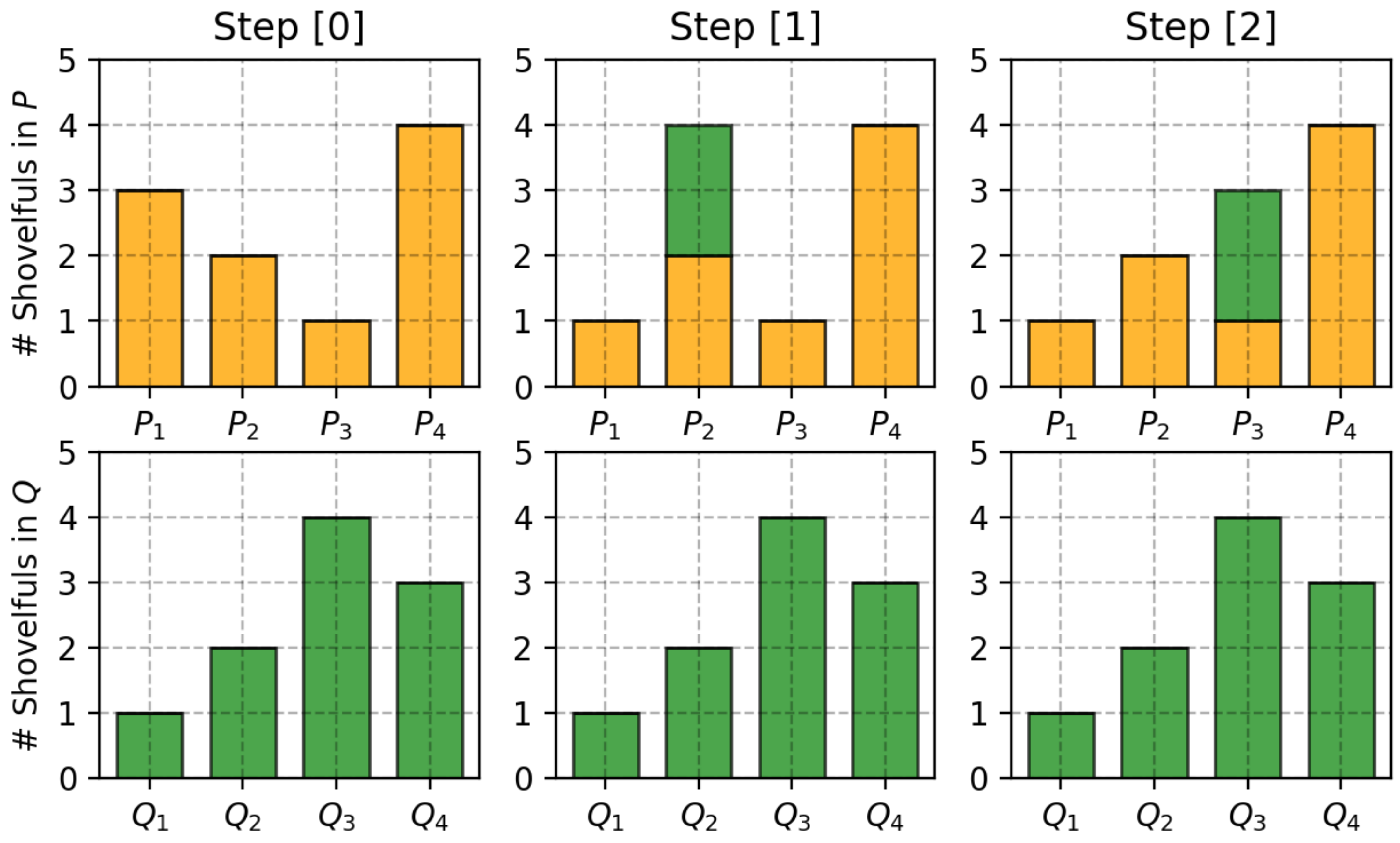
- Earth Mover Distance
- Optimal transport problem
- Most energy efficient way to match two distributions
  - Energy defined as (mass times distance)



https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html

Wasserstein GAN

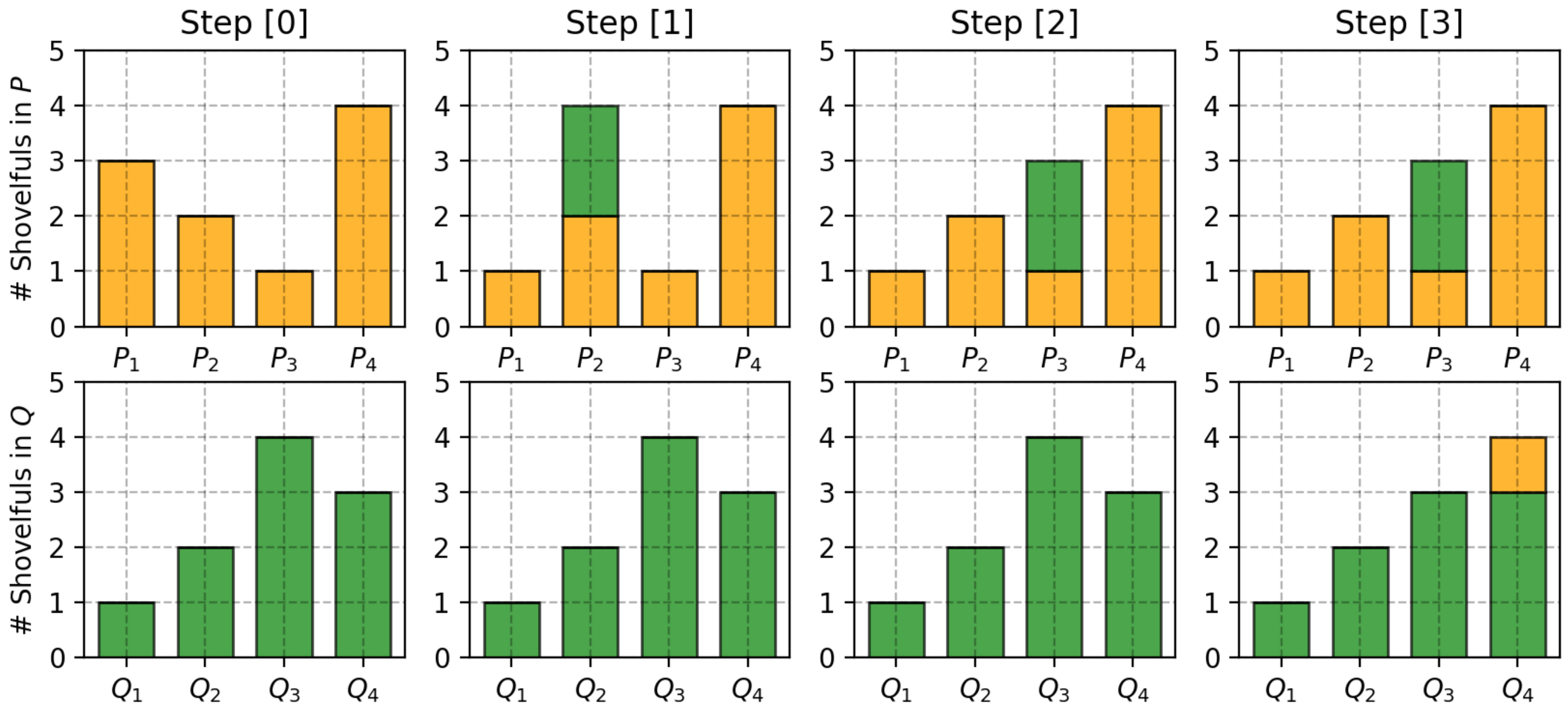
- Earth Mover Distance
- Optimal transport problem
- Most energy efficient way to match two distributions
  - Energy defined as (mass times distance)



https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html

Wasserstein GAN

- Earth Mover Distance
- Optimal transport problem
- Most energy efficient way to match two distributions
  - Energy defined as (mass times distance)
- Sensitive to distance



https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html

Wasserstein GAN

- Earth Mover Distance
  - Direct calculation not feasible

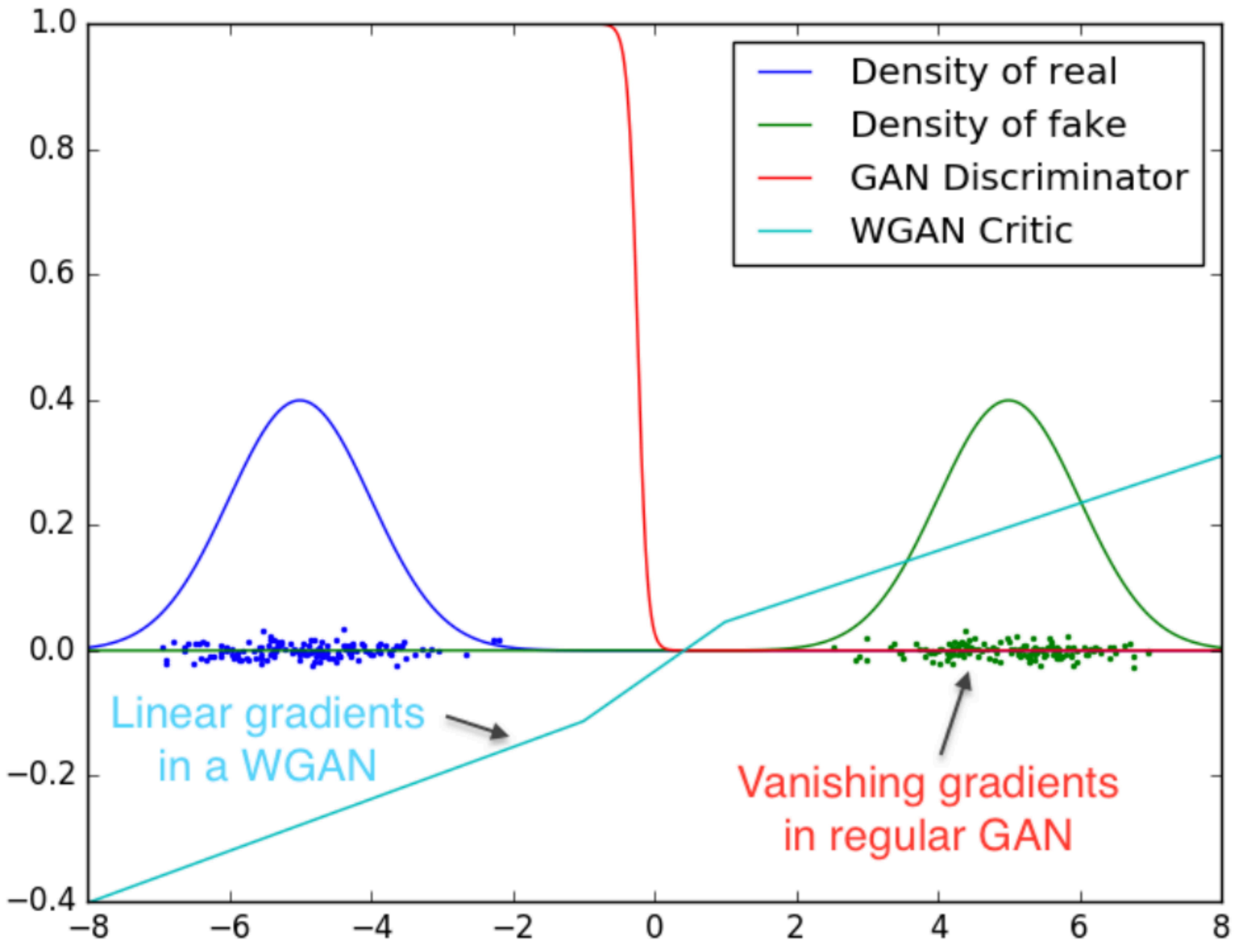
$$W(p_r, p_g) = \frac{1}{K} \sup_{\|f\|_L \leq K} \mathbb{E}_{x\sim p_r}[f(x)] - \mathbb{E}_{x\sim p_g}[f(x)]$$

  - Gives upper bound, but needs K-Lipschitz continuous function f
  - Constrain Discriminator to fulfil this requirement ➡ Critic
  - Use Critic to estimate EMD

$$W(p_r, p_g) = \max_{w\in W} \mathbb{E}_{x\sim p_r}[f_w(x)] - \mathbb{E}_{z\sim p_r(z)}[f_w(g_\theta(z))]$$

Wasserstein GAN

- JSD only based on overlap
  - Problem for very separated distributions
  - Vanishing Gradient
- Alternative to JSE: Earth Mover Distance (aka Wasserstein Distance)
  - Gradient no longer vanishes



Arjovsky et al.- arXiv:1701.07875

Wasserstein GAN

Code Example

Generative Models in Physics

- Monte Carlo simulations are invaluable part of particle physics
  - Detailed, full (matrix element, hadronisation, detector effects) simulation takes significant time and resources

Generative Models in Physics

- Monte Carlo simulations are invaluable part of particle physics
  - Detailed, full (matrix element, hadronisation, detector effects) simulation takes significant time and resources
- Increasing collider luminosities calls for more Monte Carlo data

Generative Models in Physics

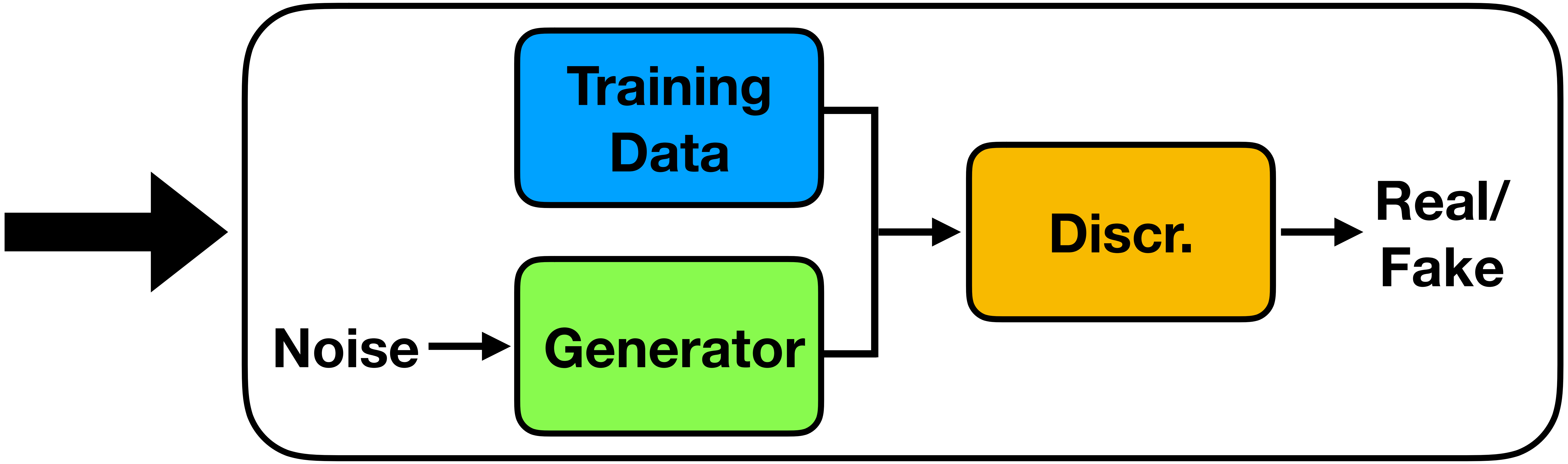
- Monte Carlo simulations are invaluable part of particle physics
  - Detailed, full (matrix element, hadronisation, detector effects) simulation takes significant time and resources
- Increasing collider luminosities calls for more Monte Carlo data
  - Need a way to speed up simulations

Generative Models in Physics

- Monte Carlo simulations are invaluable part of particle physics
  - Detailed, full (matrix element, hadronisation, detector effects) simulation takes significant time and resources
- Increasing collider luminosities calls for more Monte Carlo data
  - Need a way to speed up simulations
- Train Generative Network to emulate simulators
  - Networks are significantly faster than classical methods

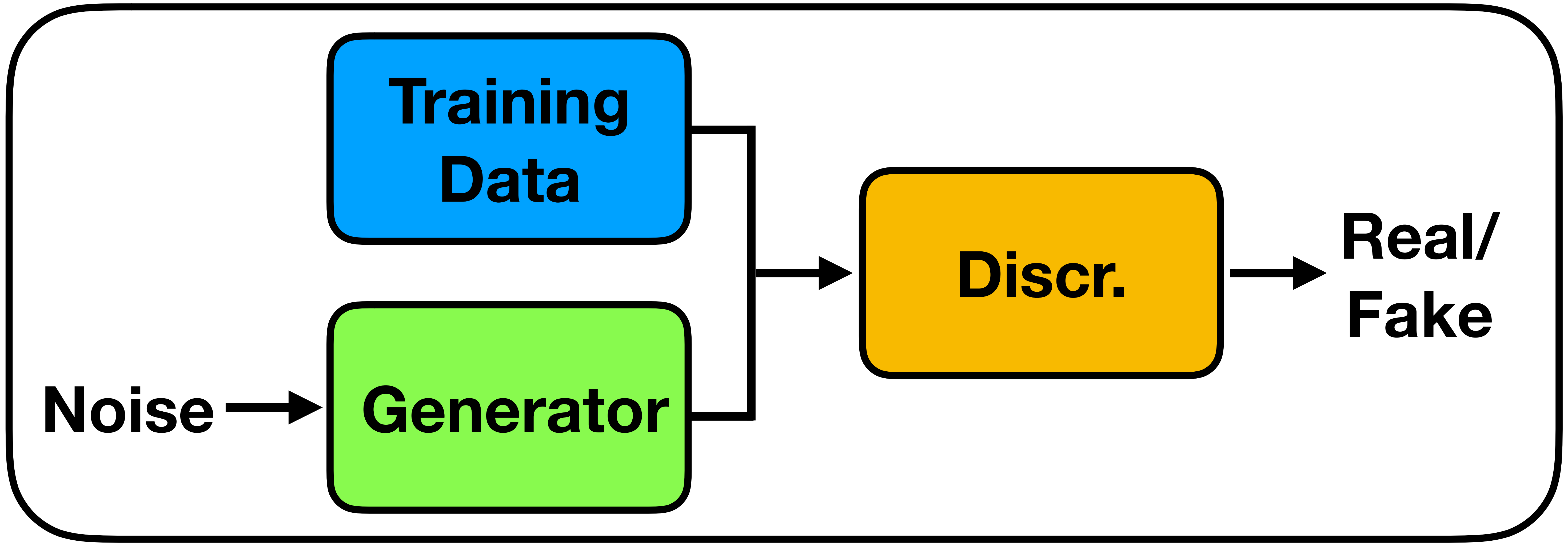
Generative Models in Physics

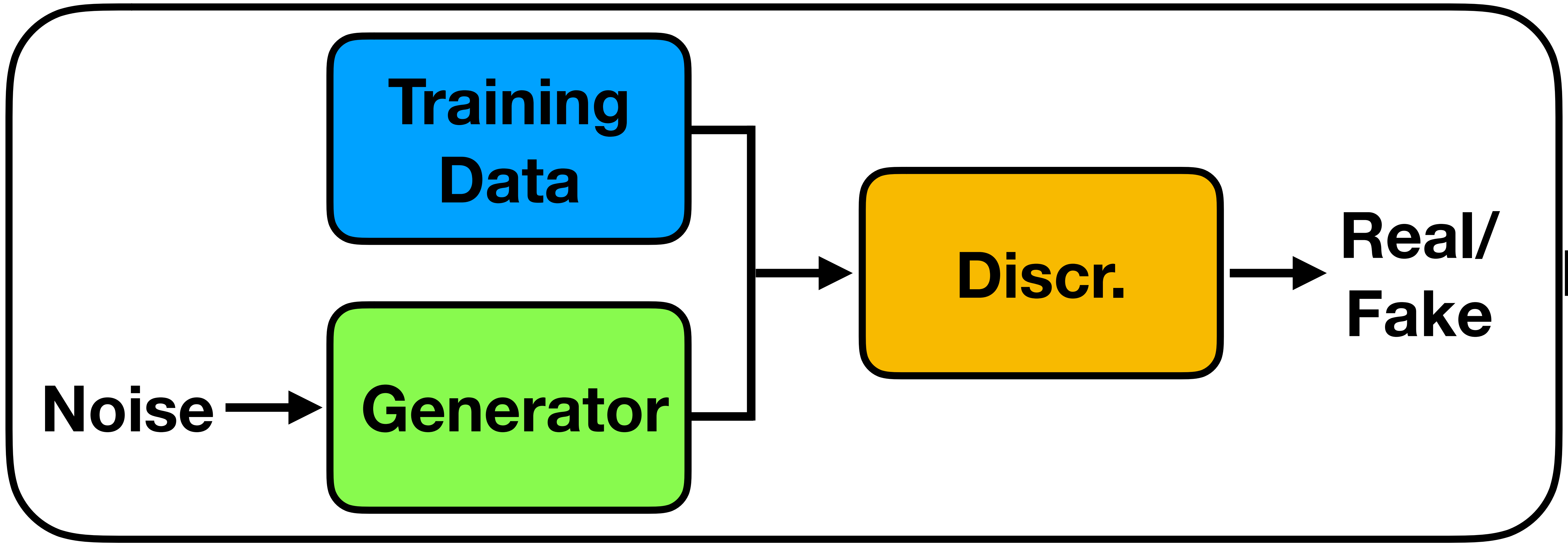
Real data,
N points



Generative Models in Physics

Real data, N points





GANed data, M >> N points

Potential Problem

Info(N real points) Info(M GANed points)

Potential Problem

Info(N real points) = Info(M GANed points)

Little advantage to be gained from GAN

Potential Problem

Info(N real points) $=$ Info(M GANed points)

Little advantage to be gained from GAN

Info(N real points) $<$ Info(M GANed points)

Potential Problem

Info(N real points) $=$ Info(M GANed points)

Little advantage to be gained from GAN

Info(N real points) $<$ Info(M GANed points)

GAN can speed up simulations

Potential Problem

Info(N real points) $=$ Info(M GANed points)

Little advantage to be gained from GAN

Info(N real points) $<$ Info(M GANed points)

GAN can speed up simulations

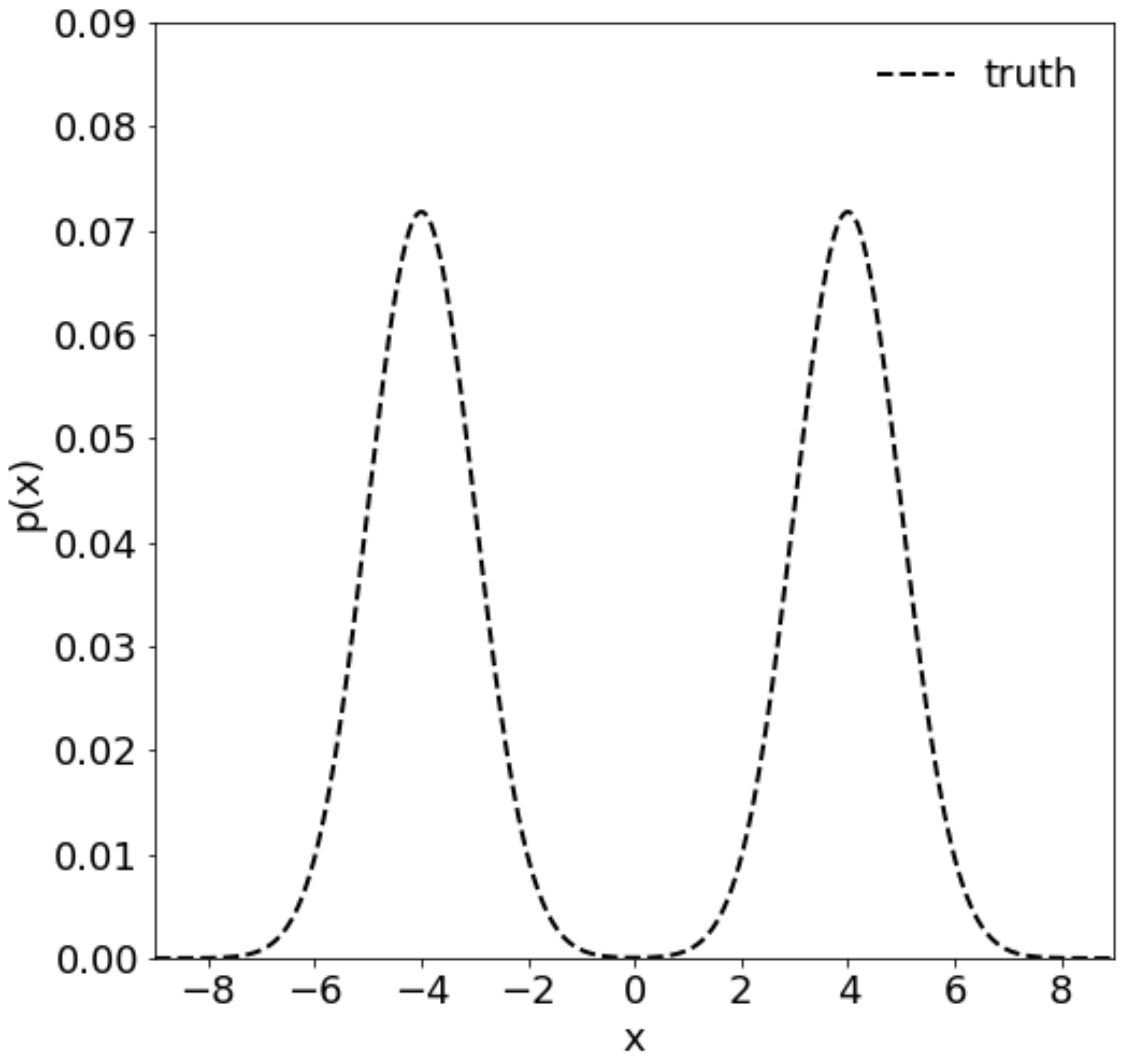
➡ Check with small scale test

Butter et al.: **GANplifying Event Samples:** 2008.06545

1-D Toy Model

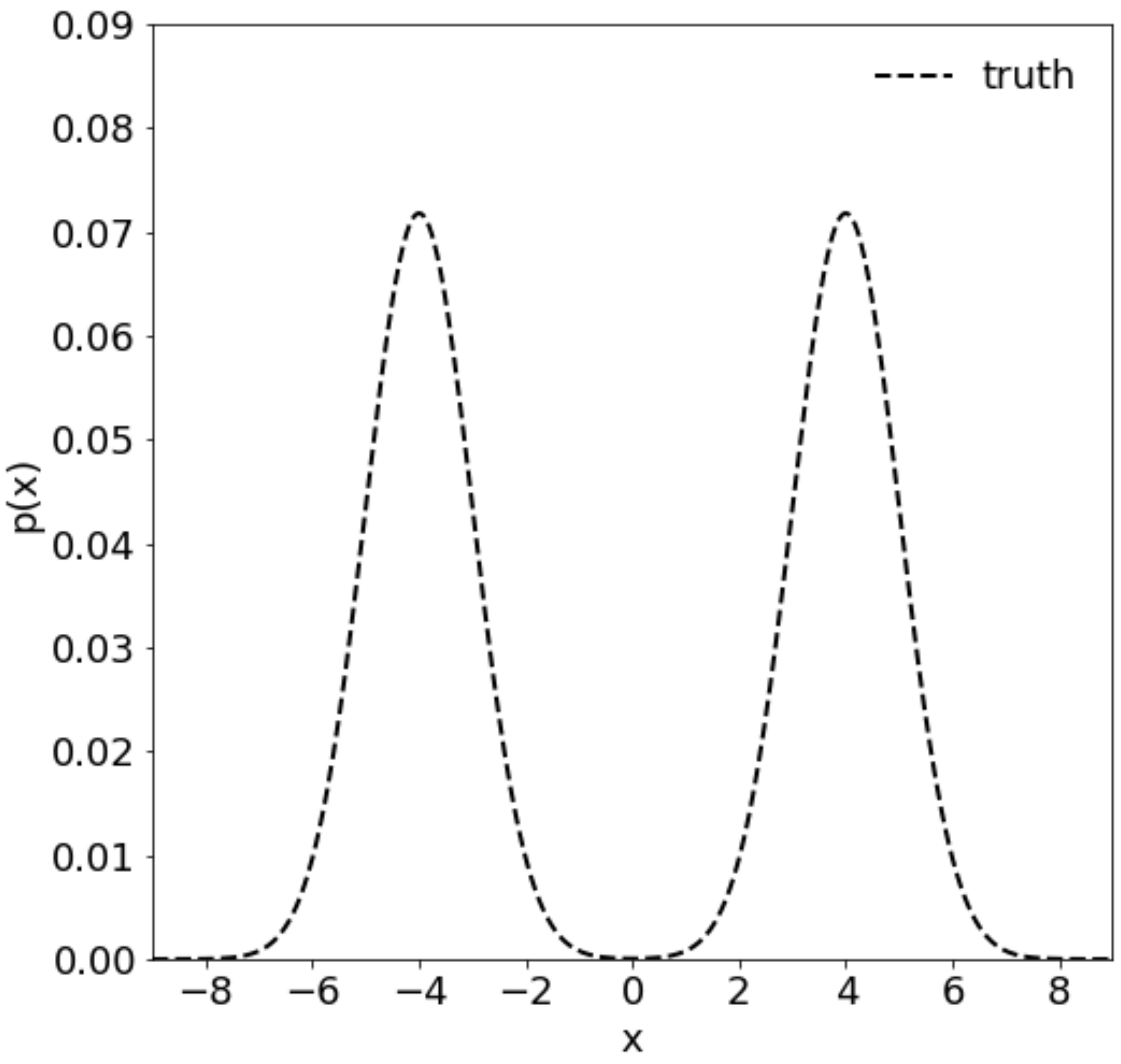
- Camel back function: double peak Gaussian

$$p(X) = \frac{1}{2}(N_{-4,1}(x) + N_{4,1}(x))$$



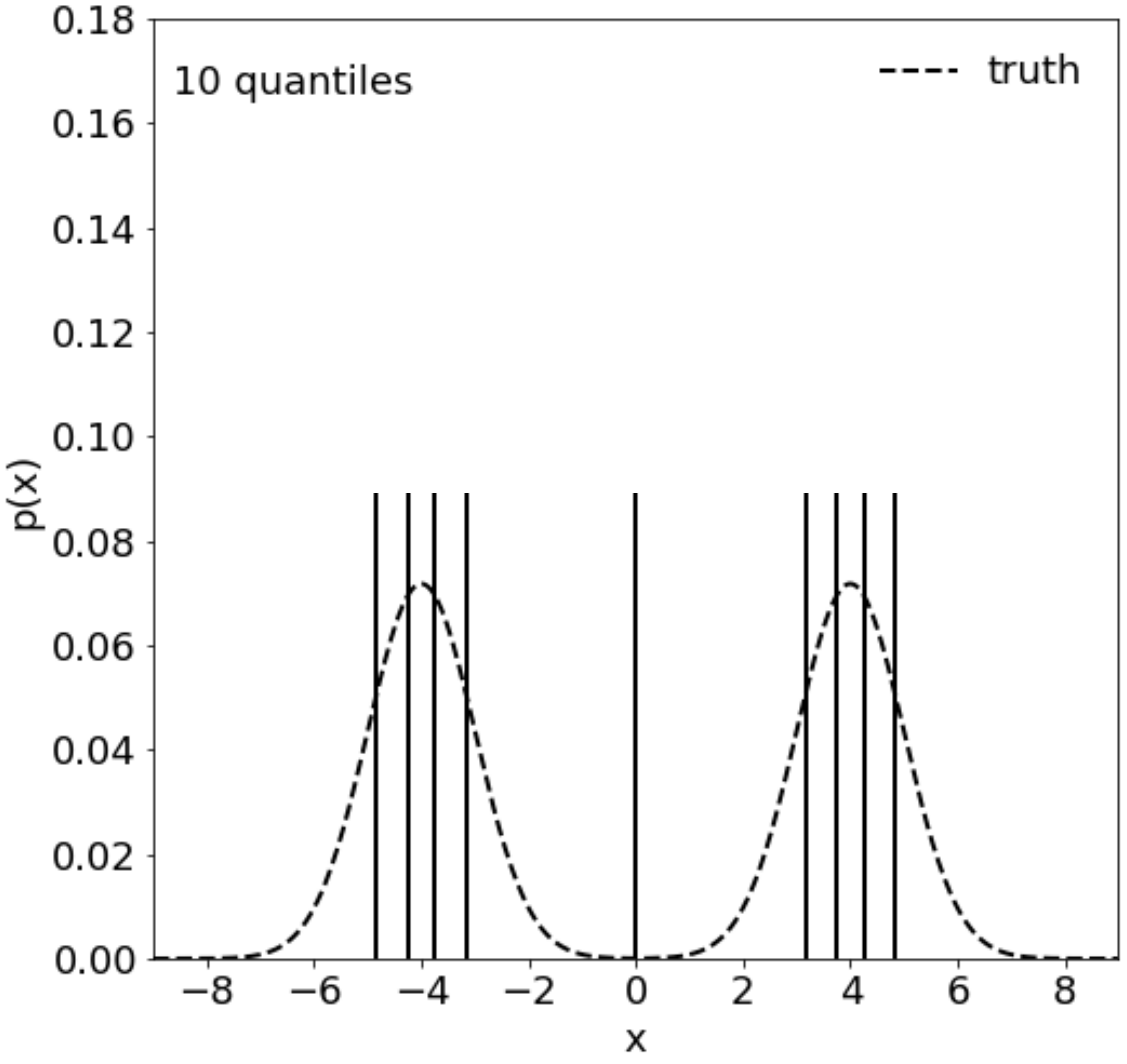
Quantiles

- Need measurement how well function is described



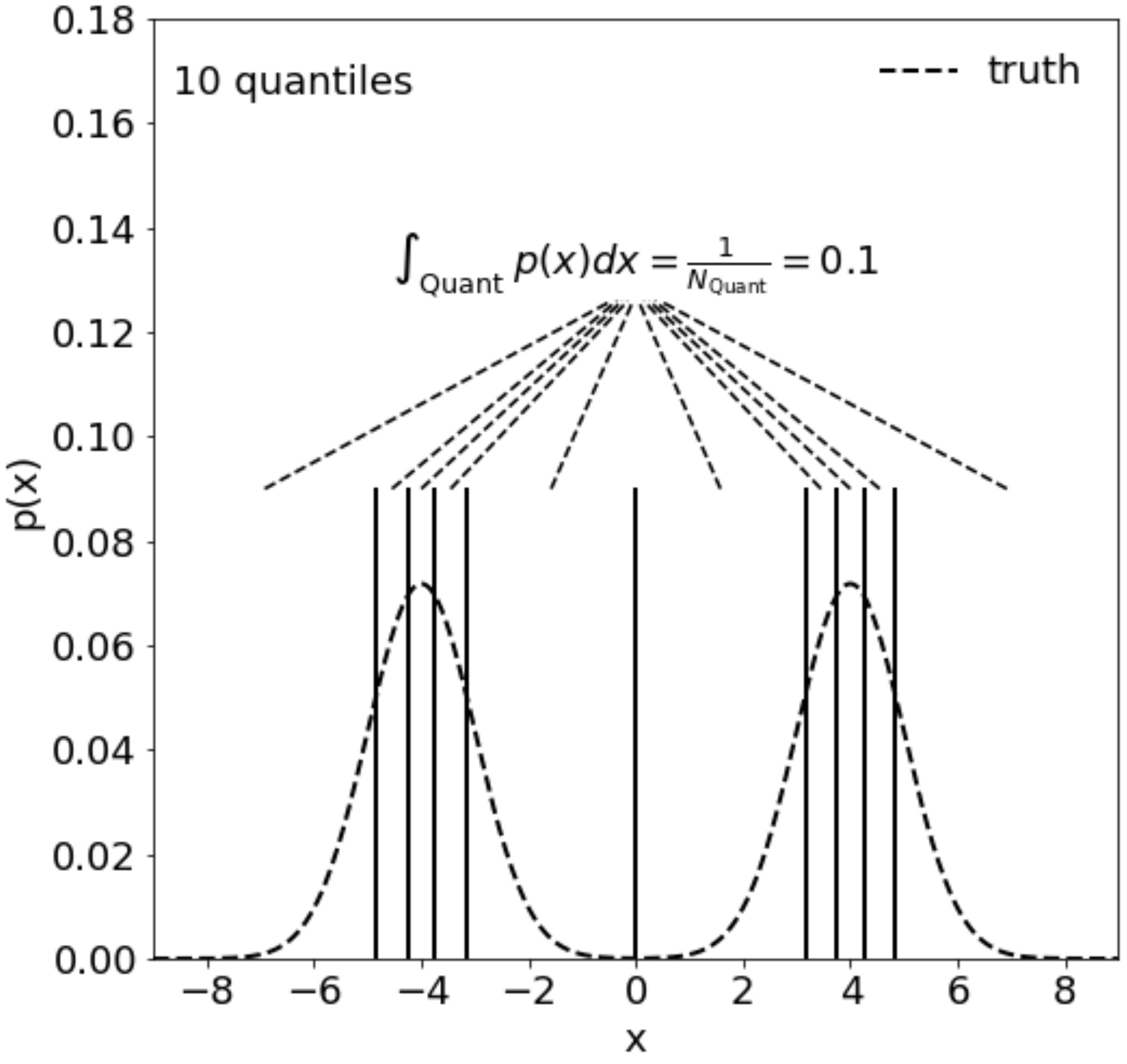
Quantiles

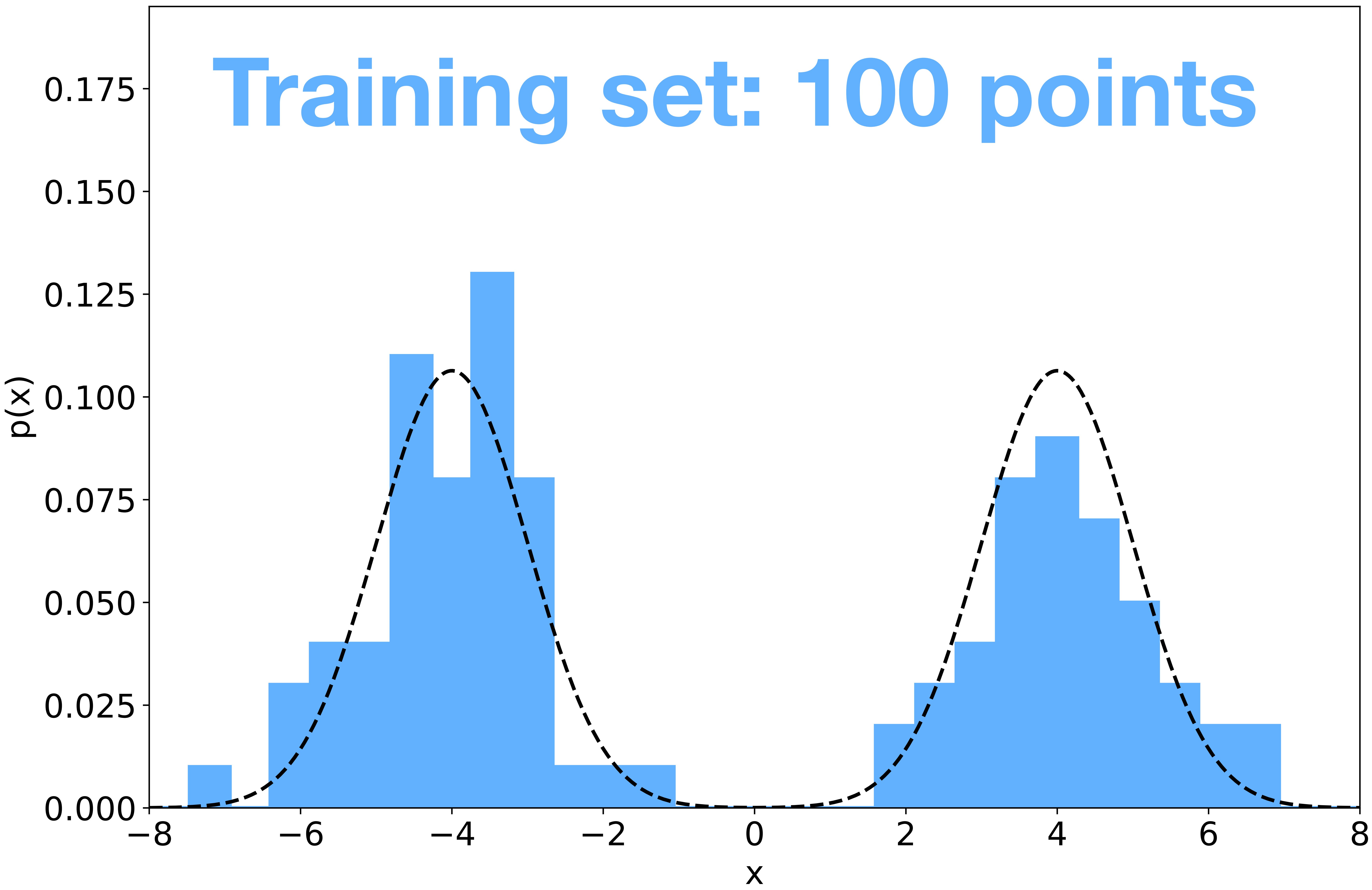
- Need measurement how well function is described
- Define 10 quantiles on true distribution



Quantiles

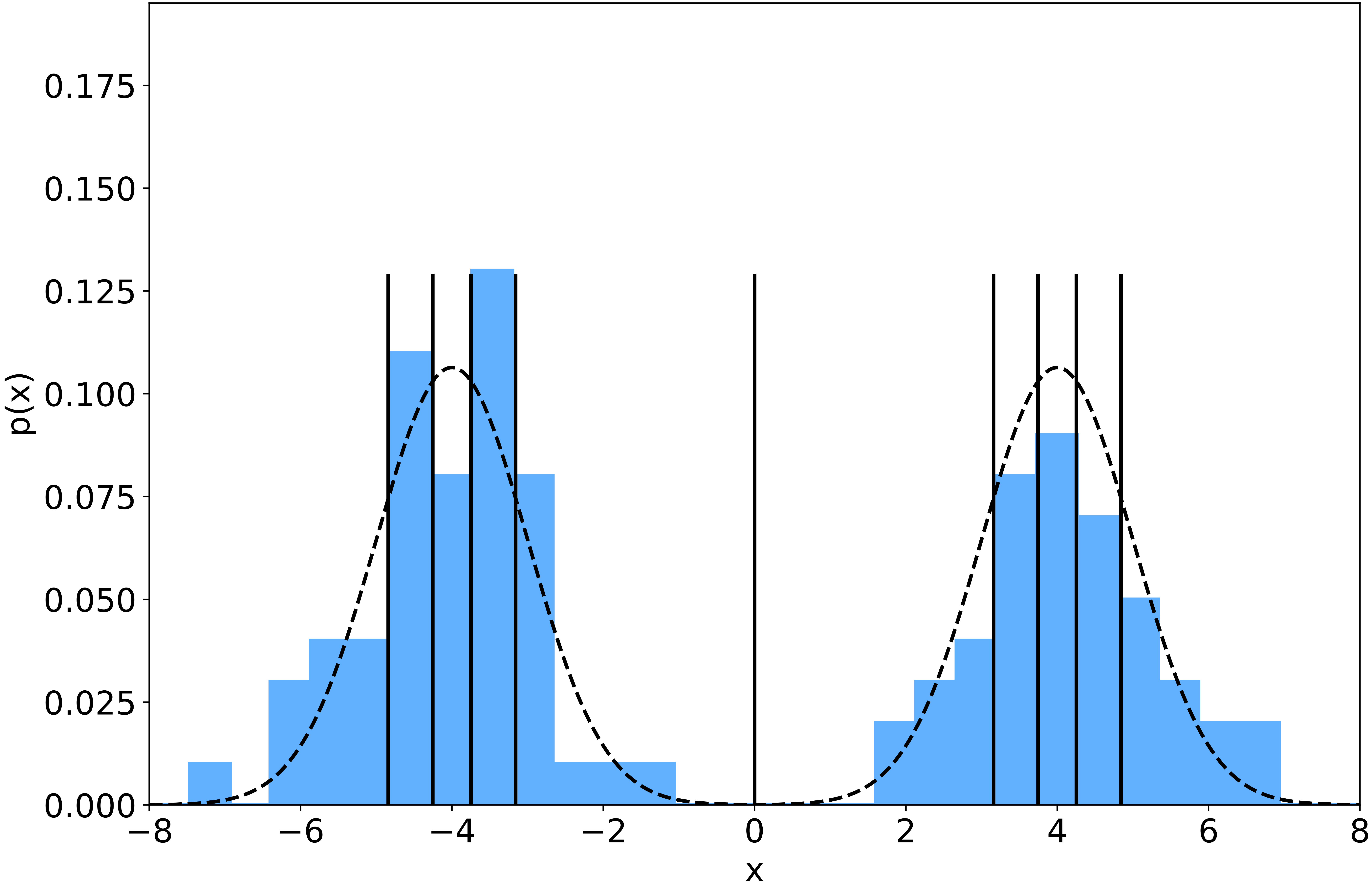
- Need measurement how well function is described
- Define 10 quantiles on true distribution
- Each quantile contains equal probability



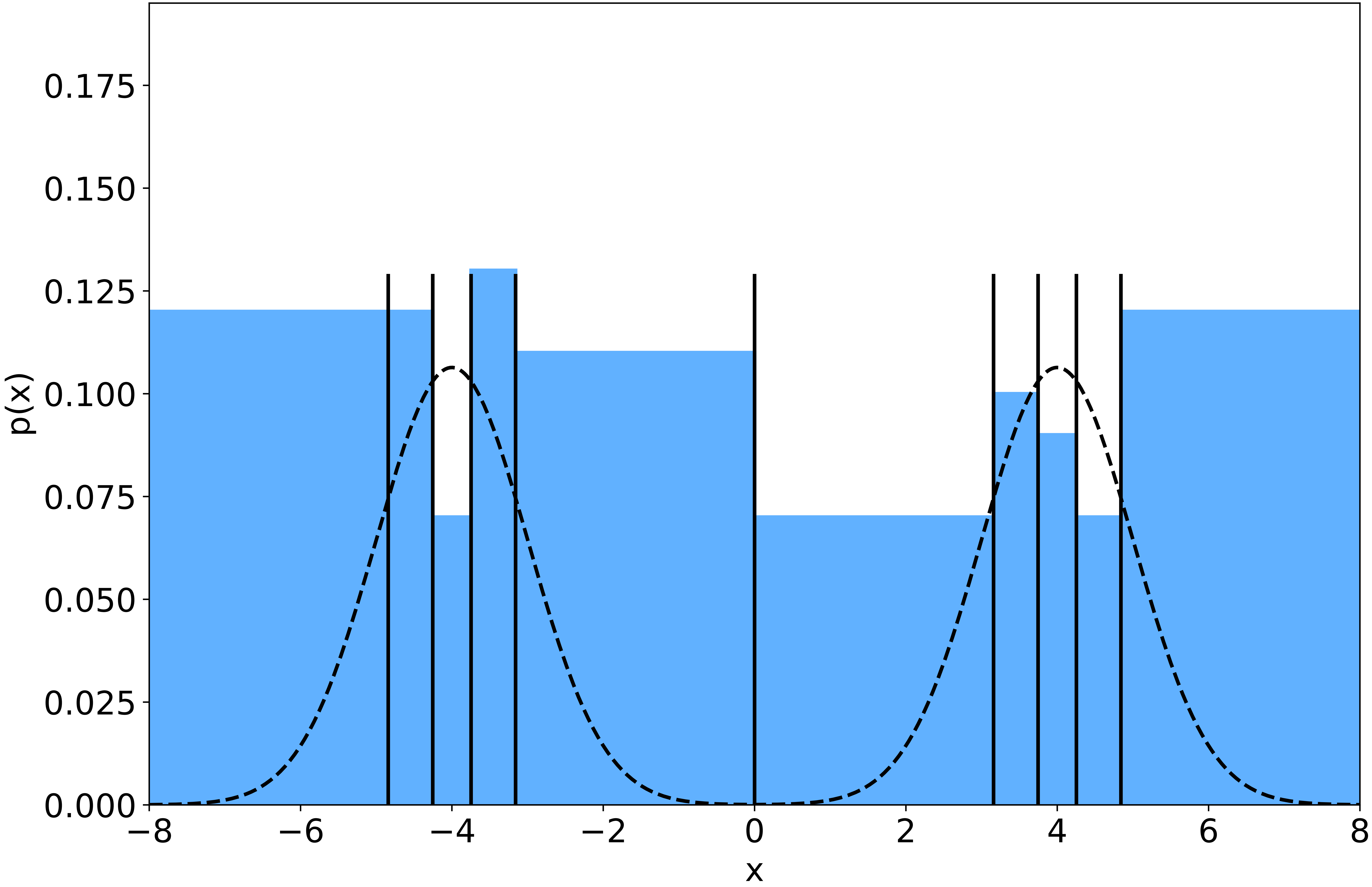


Training set: 100 points
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
p(x)
−8
−6
−4
−2
0
2
4
6
8
x

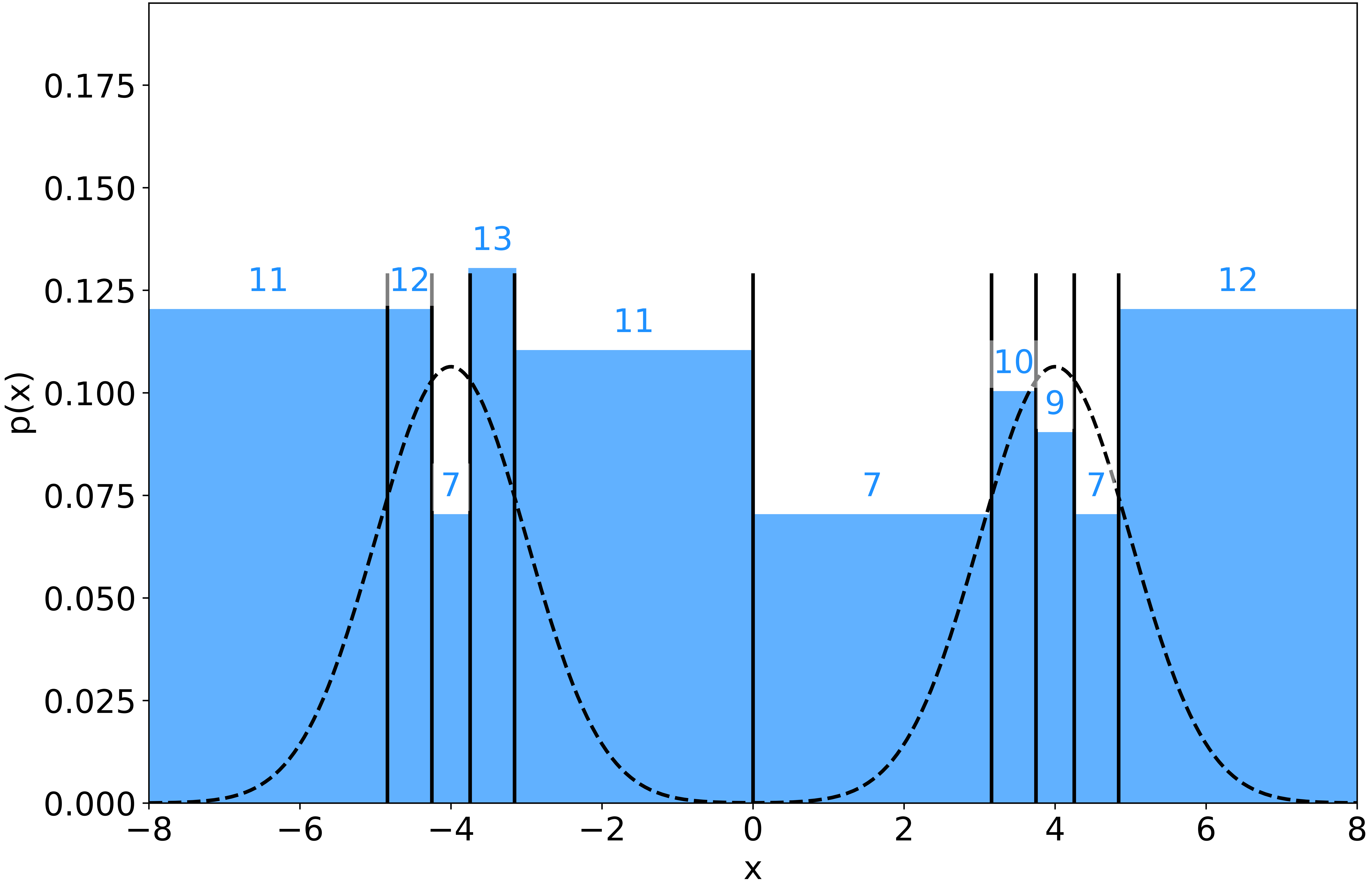
How well is the distribution described?



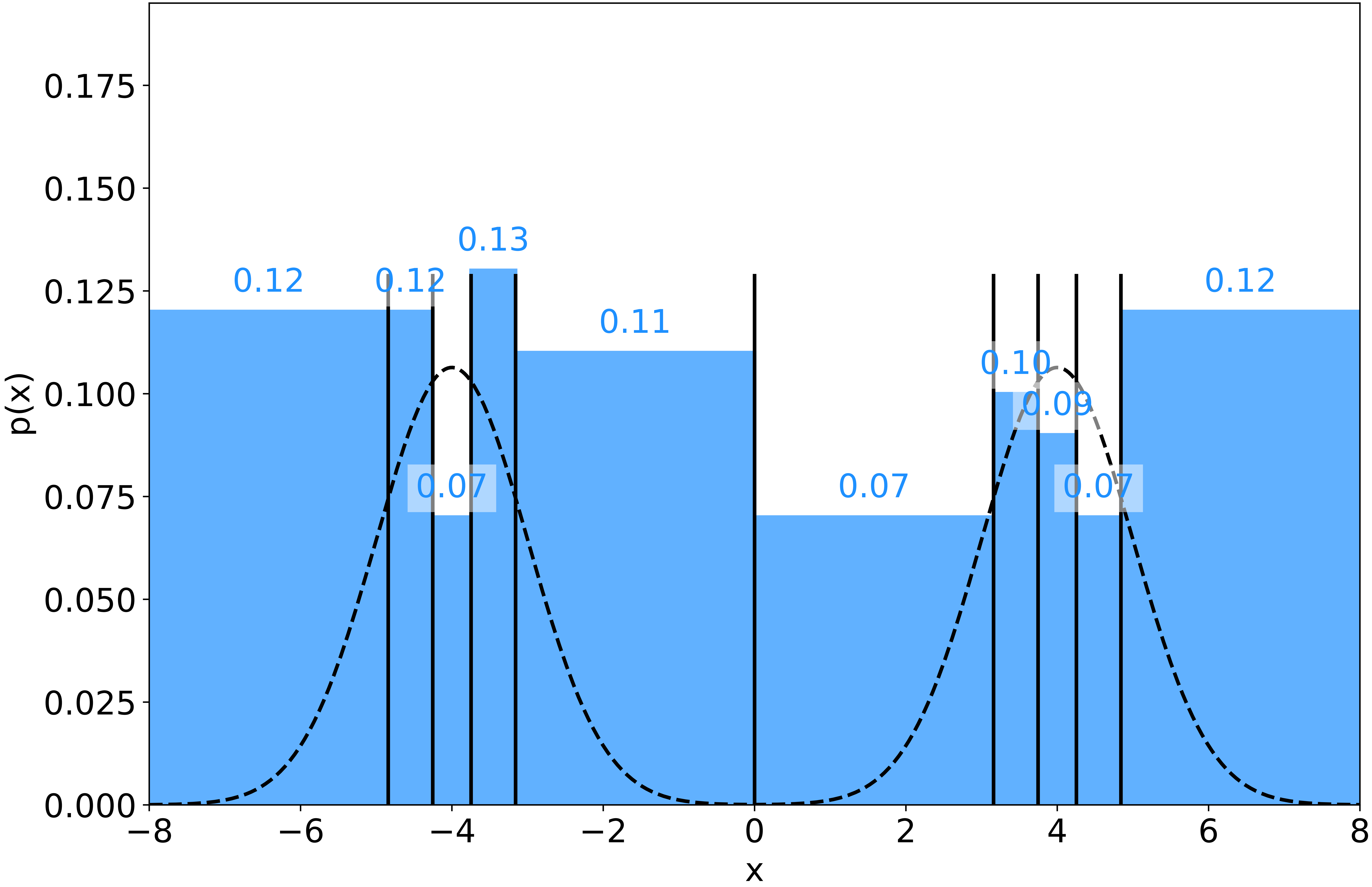
How well is the distribution described?



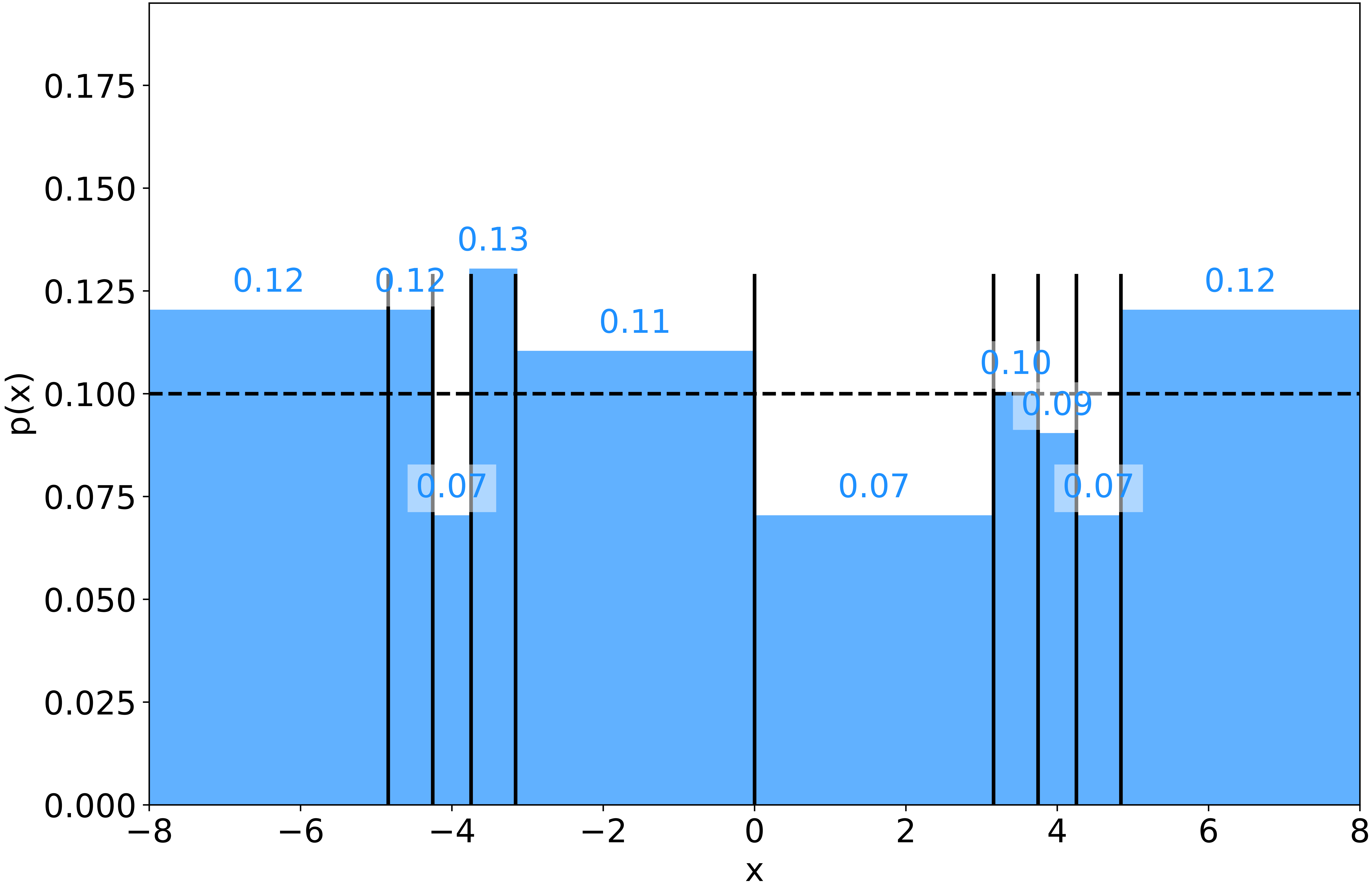
How well is the distribution described?



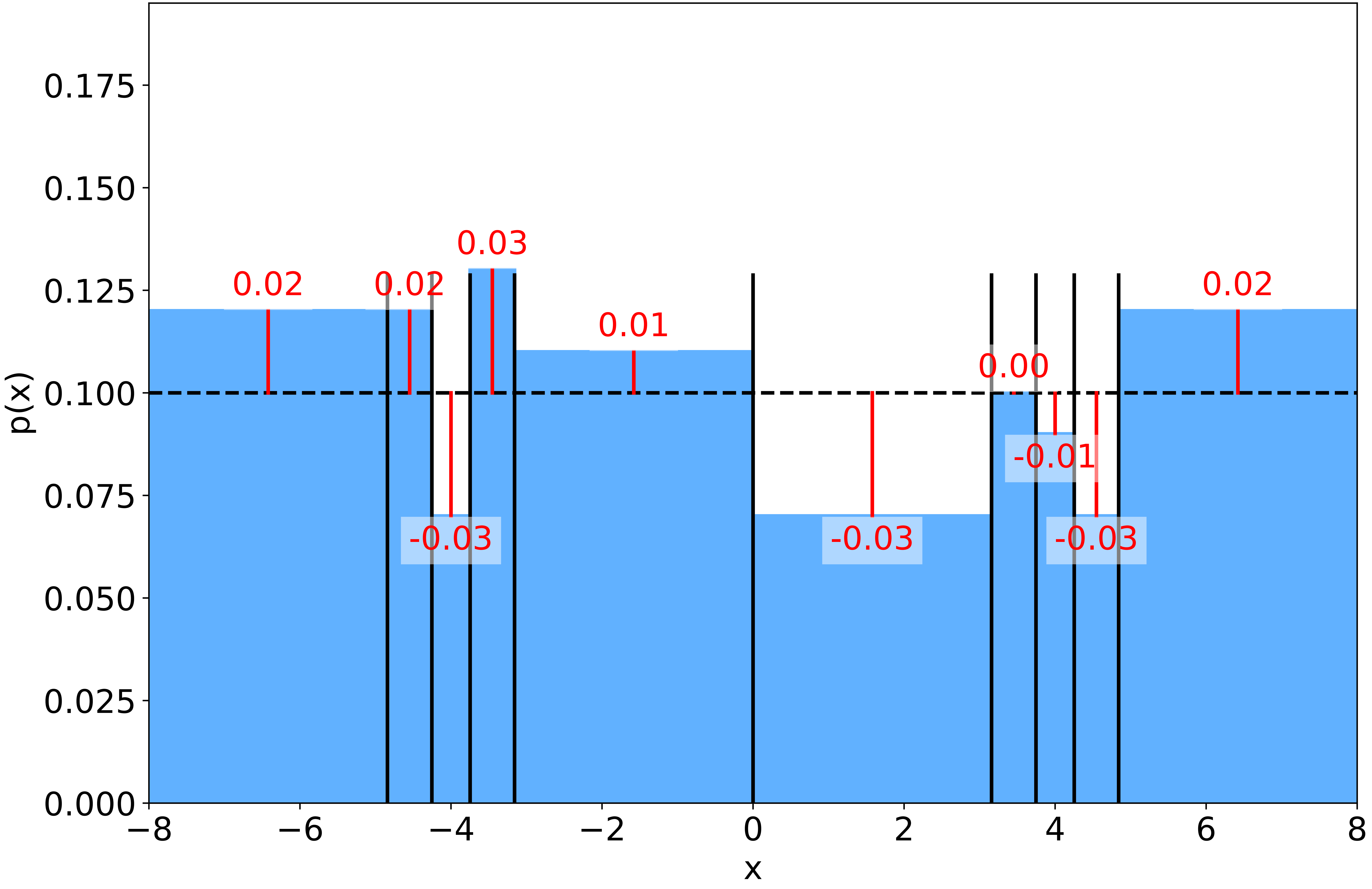
How well is the distribution described?



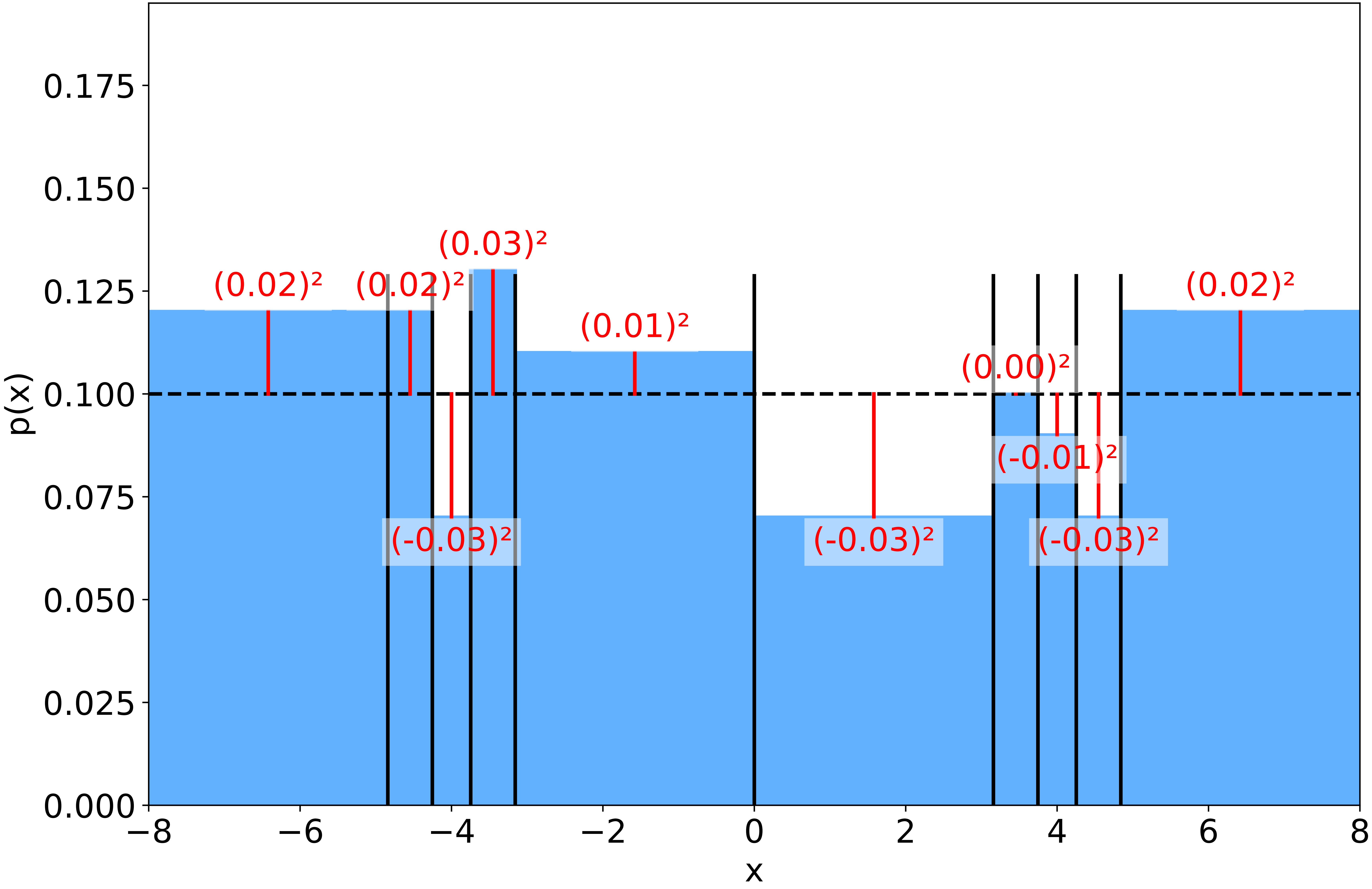
How well is the distribution described?



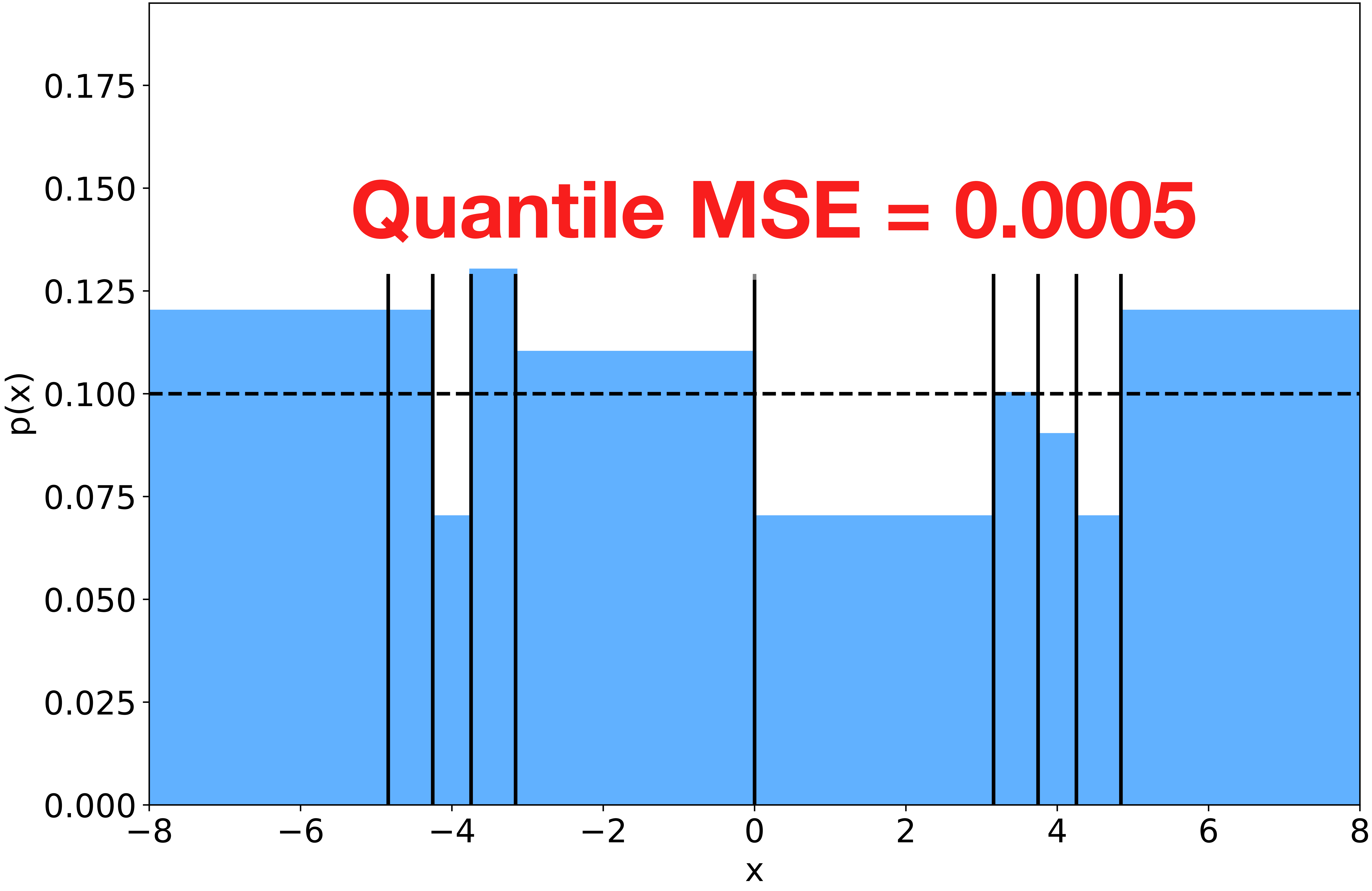
How well is the distribution described?



How well is the distribution described?

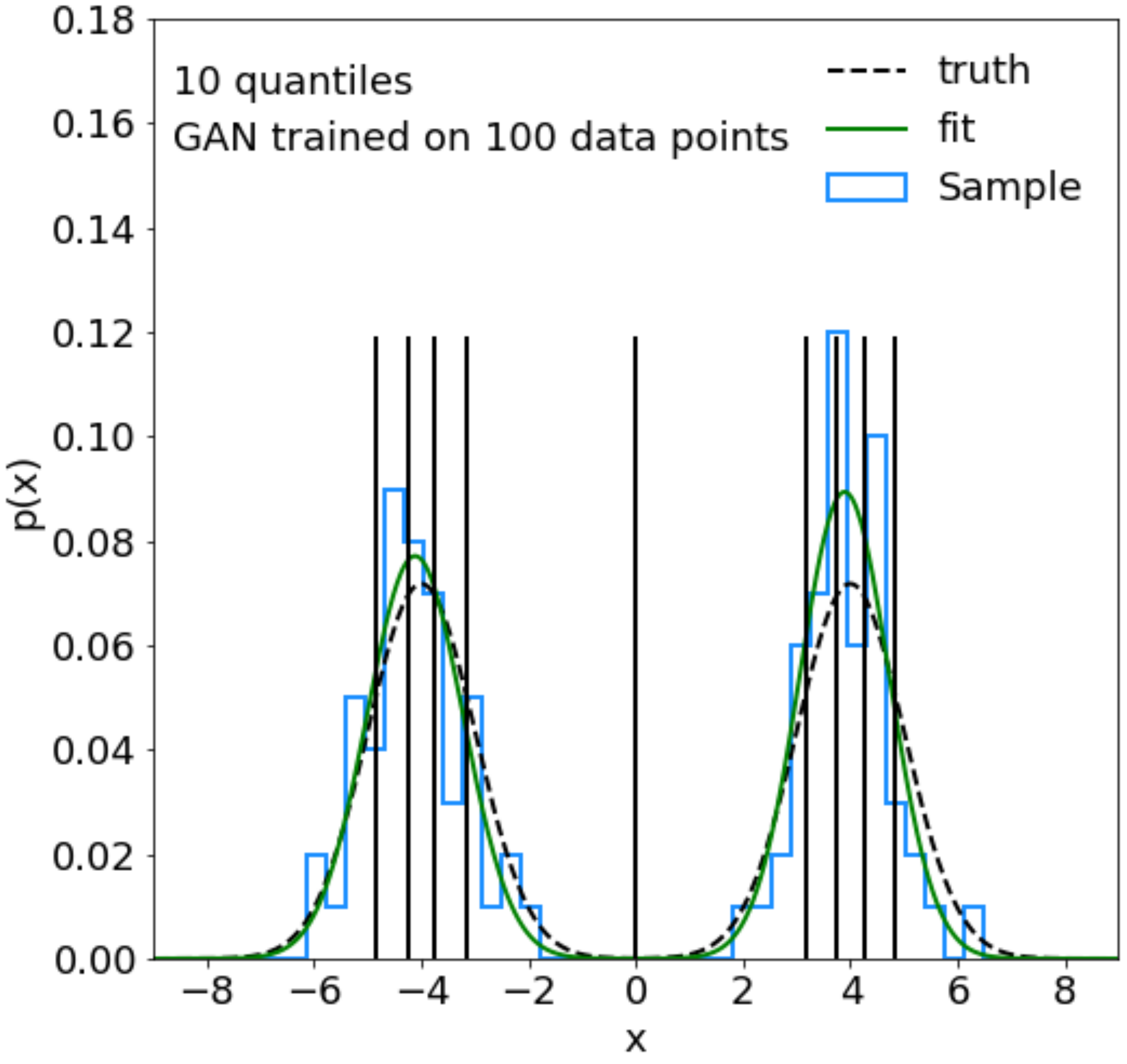


How well is the distribution described?



Parameter Fit

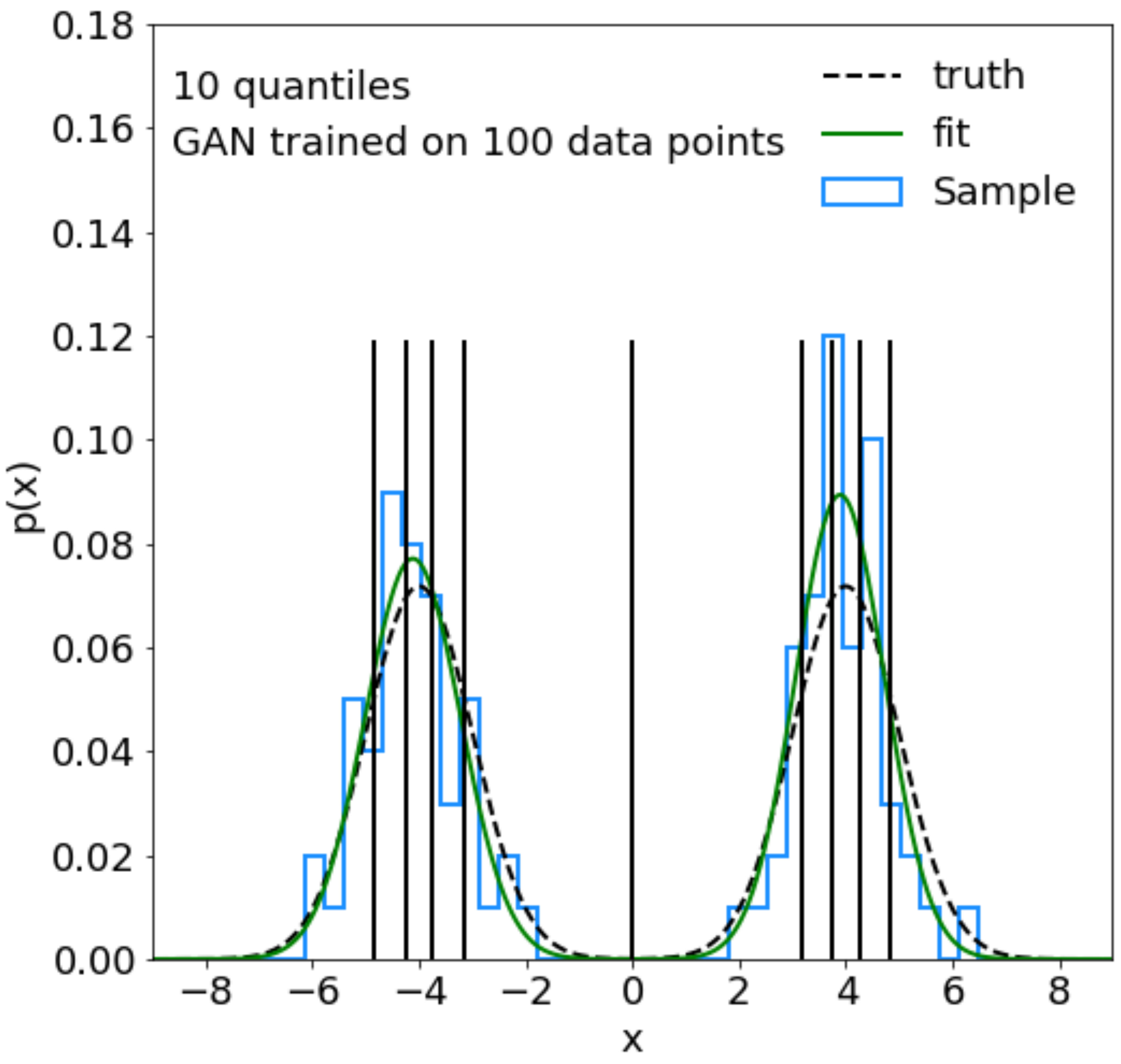
- Gives upper performance benchmark



Parameter Fit

- Gives upper performance benchmark
- Fit 5 parameter camel back function to training samples

$$p(X) = a\ N_{\mu_1,\sigma_1}(x) \\ +(1-a)N_{\mu_2,\sigma_2}(x)$$

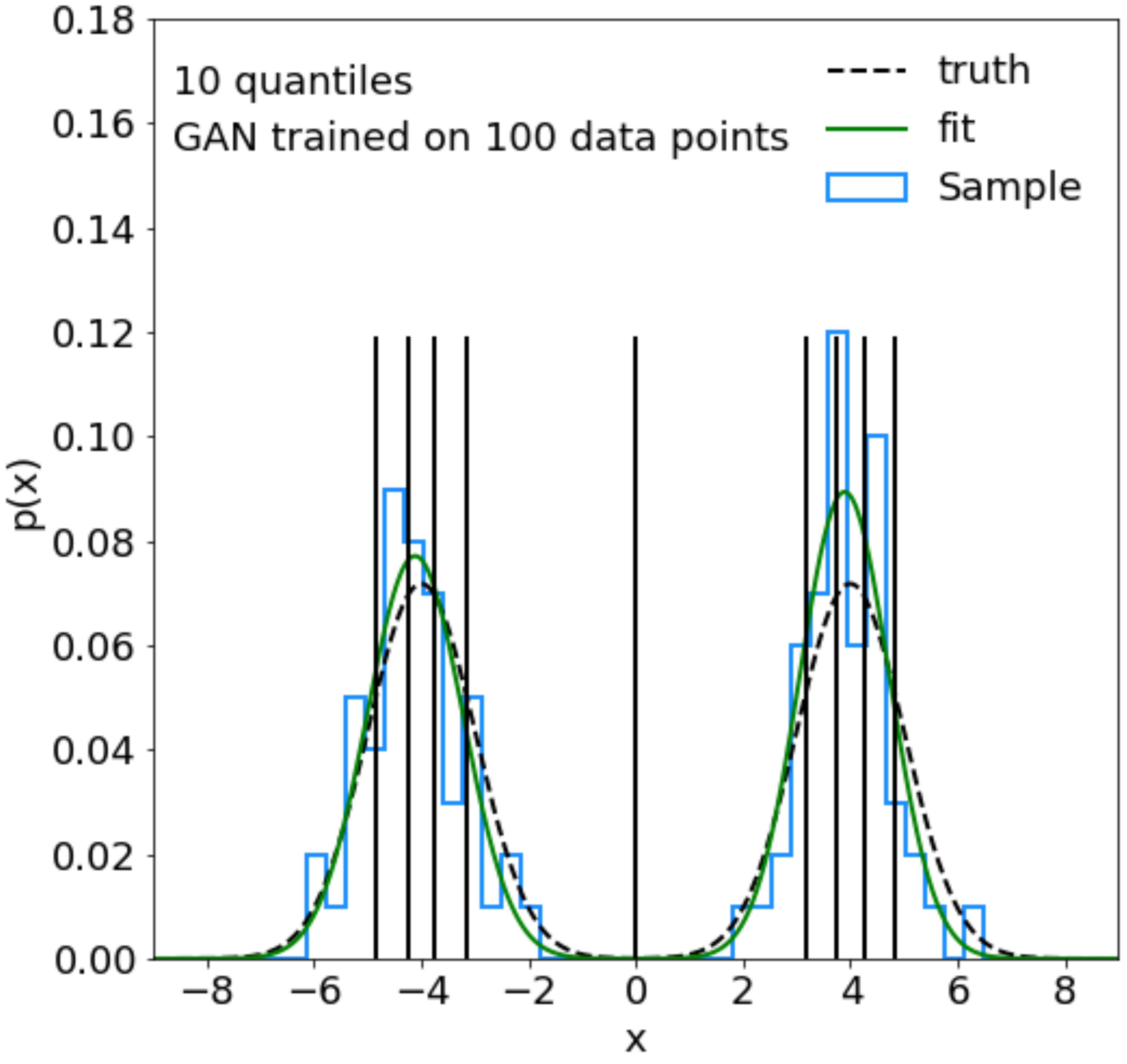


Parameter Fit

- Gives upper performance benchmark
- Fit 5 parameter camel back function to training samples

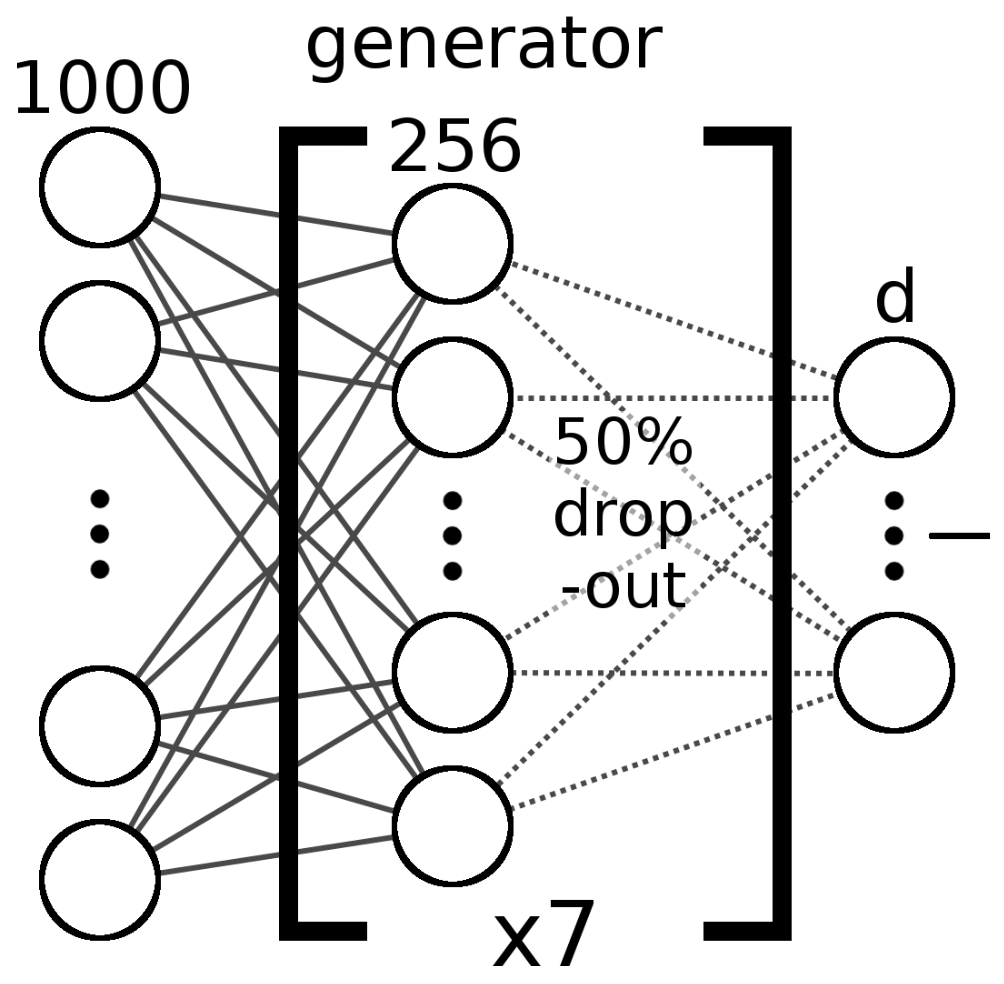
$$p(X) = a\, N_{\mu_1,\sigma_1}(x) \\ +(1-a)N_{\mu_2,\sigma_2}(x)$$

- Analytically calculate integral for each quantile



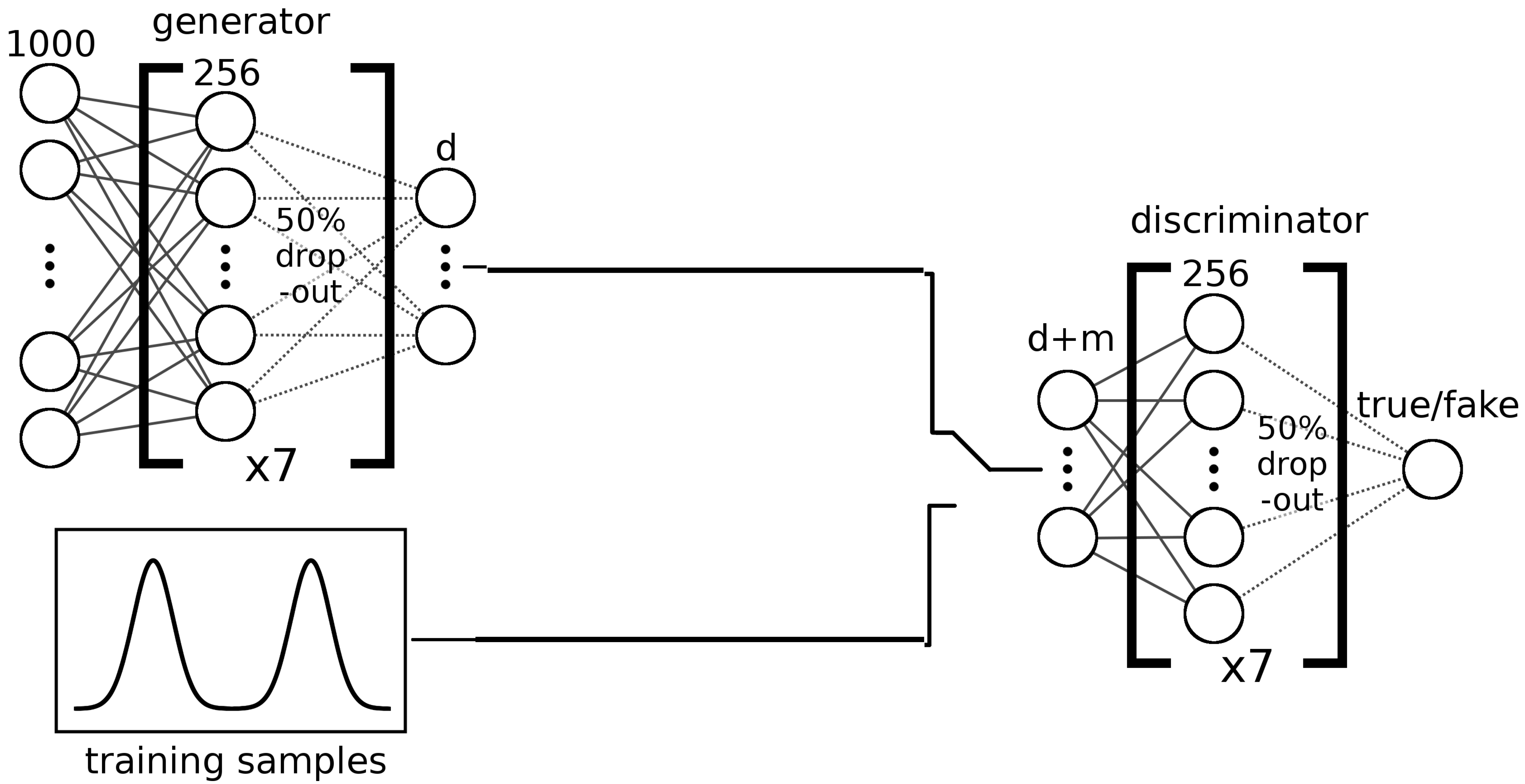
Generative Network

- Train GAN on 100 data points from training sample



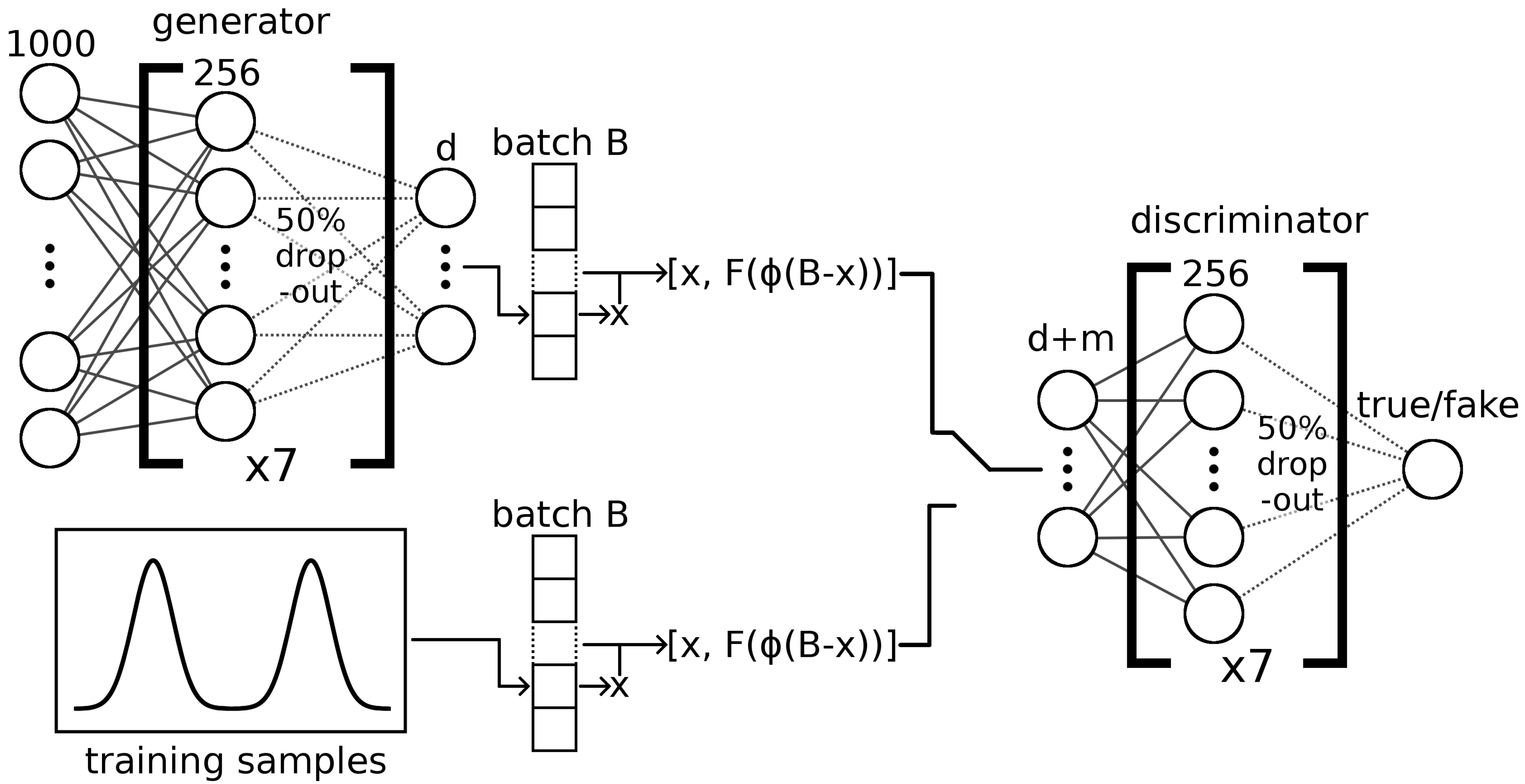
Generative Network

- Train GAN on 100 data points from training sample



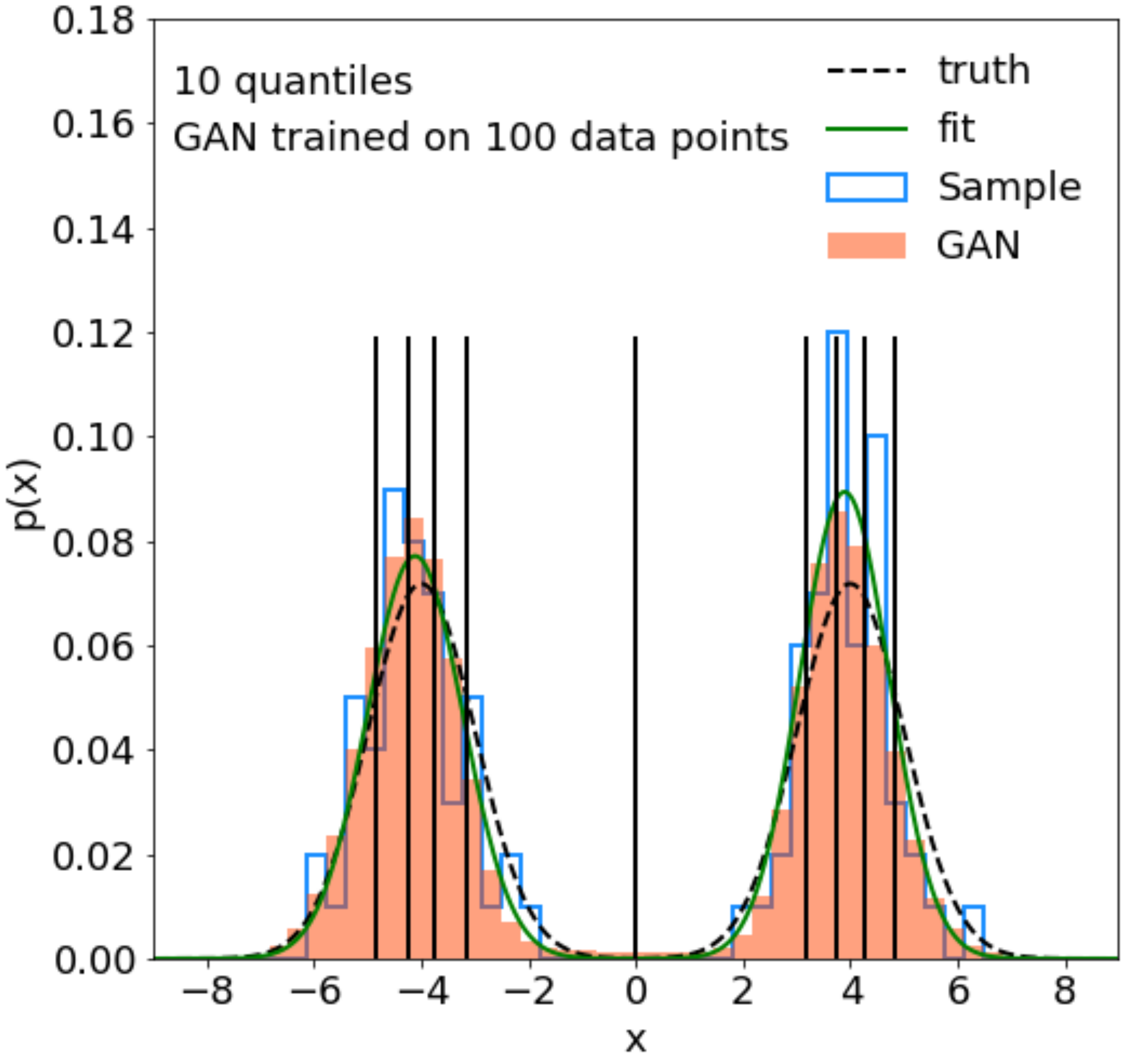
Generative Network

- Train GAN on 100 data points from training sample
- Regularisation through
  - Dropout
  - Added training noise
  - Batch-statistics



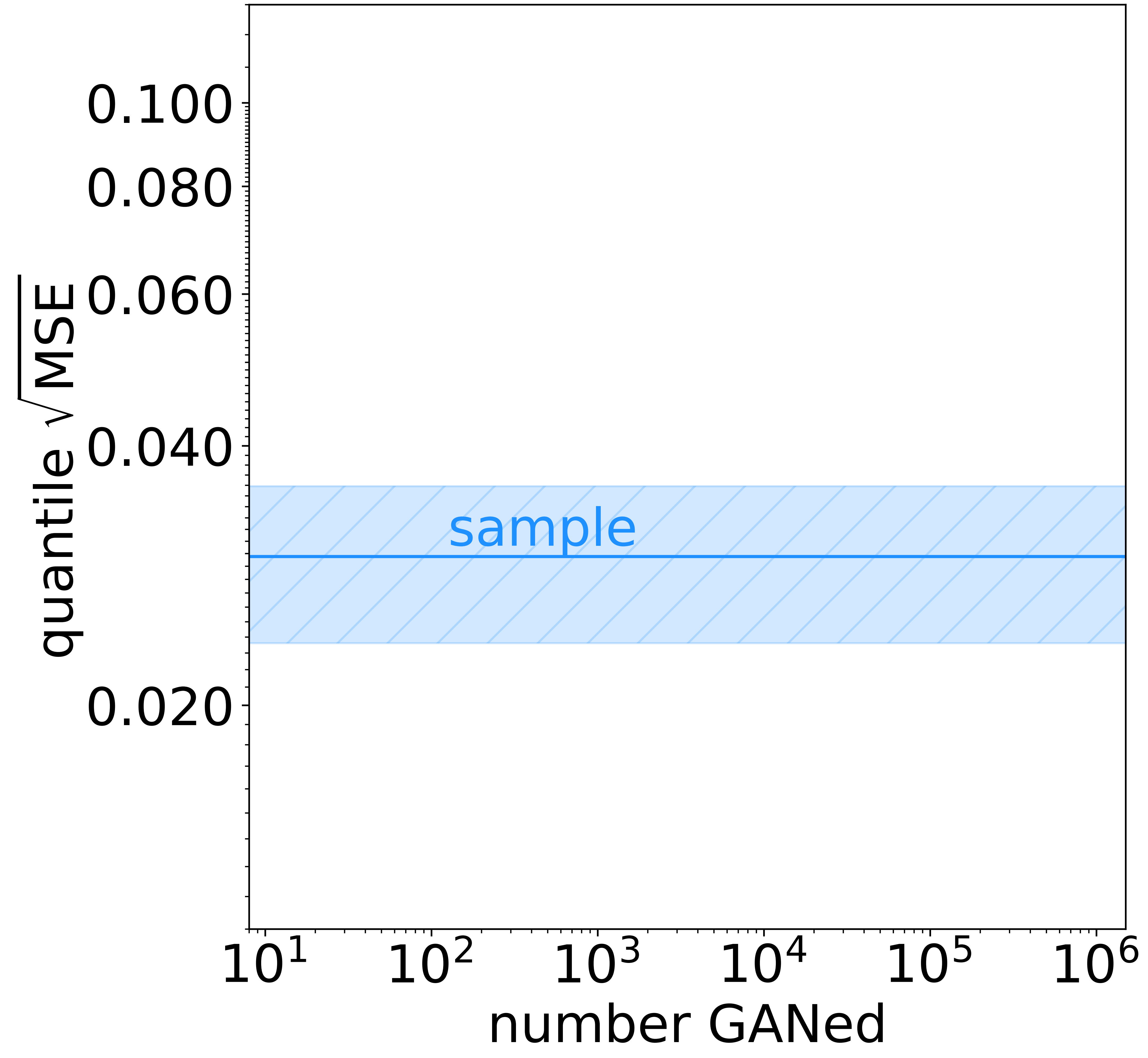
Generative Network

- Generate $O(10^7)$ data points using GAN
- Calculate fraction of points in each quantile
- Calculate Quantile MSE



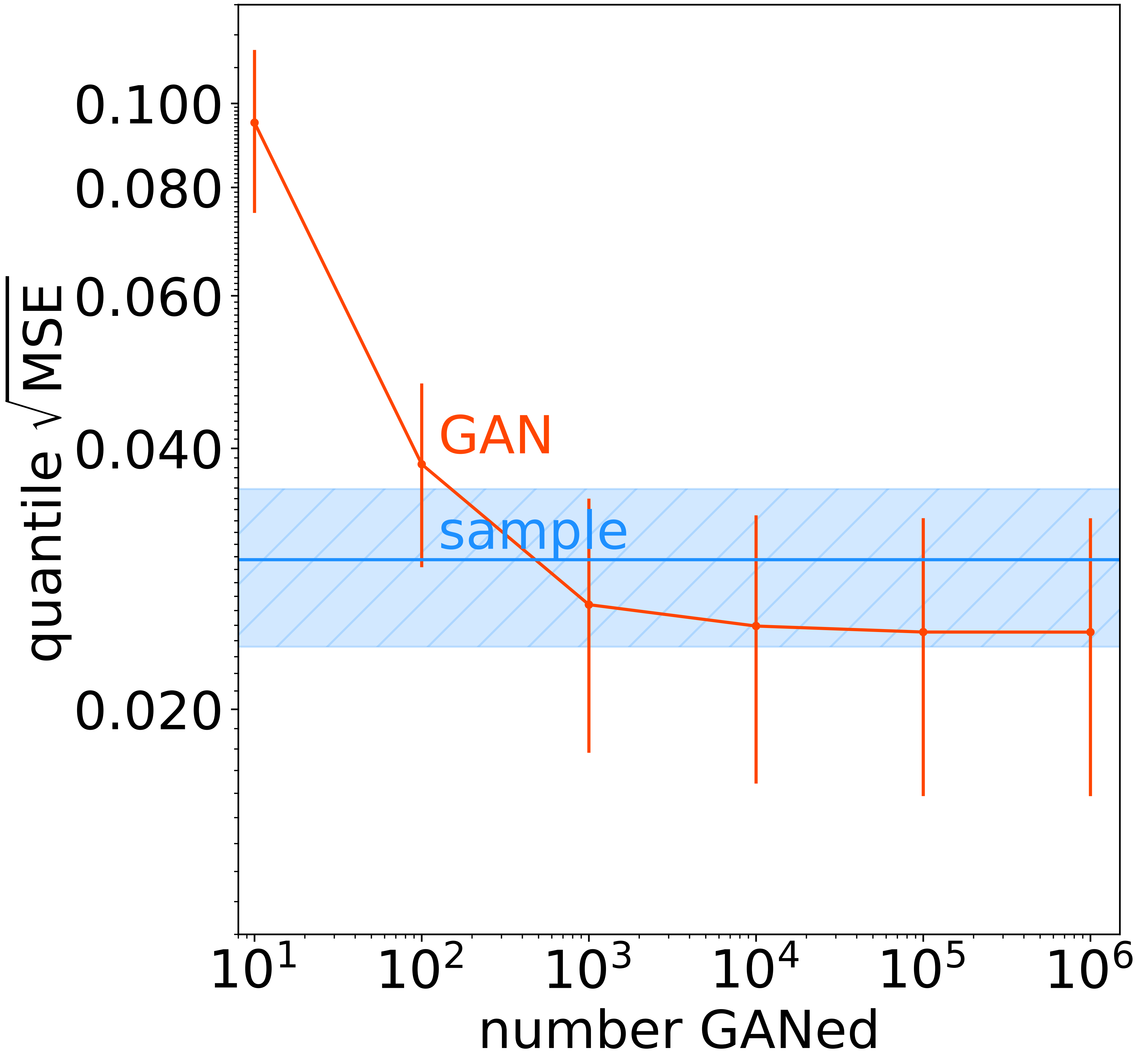
Generative Network

- Calculate MSE for 100 independent training samples



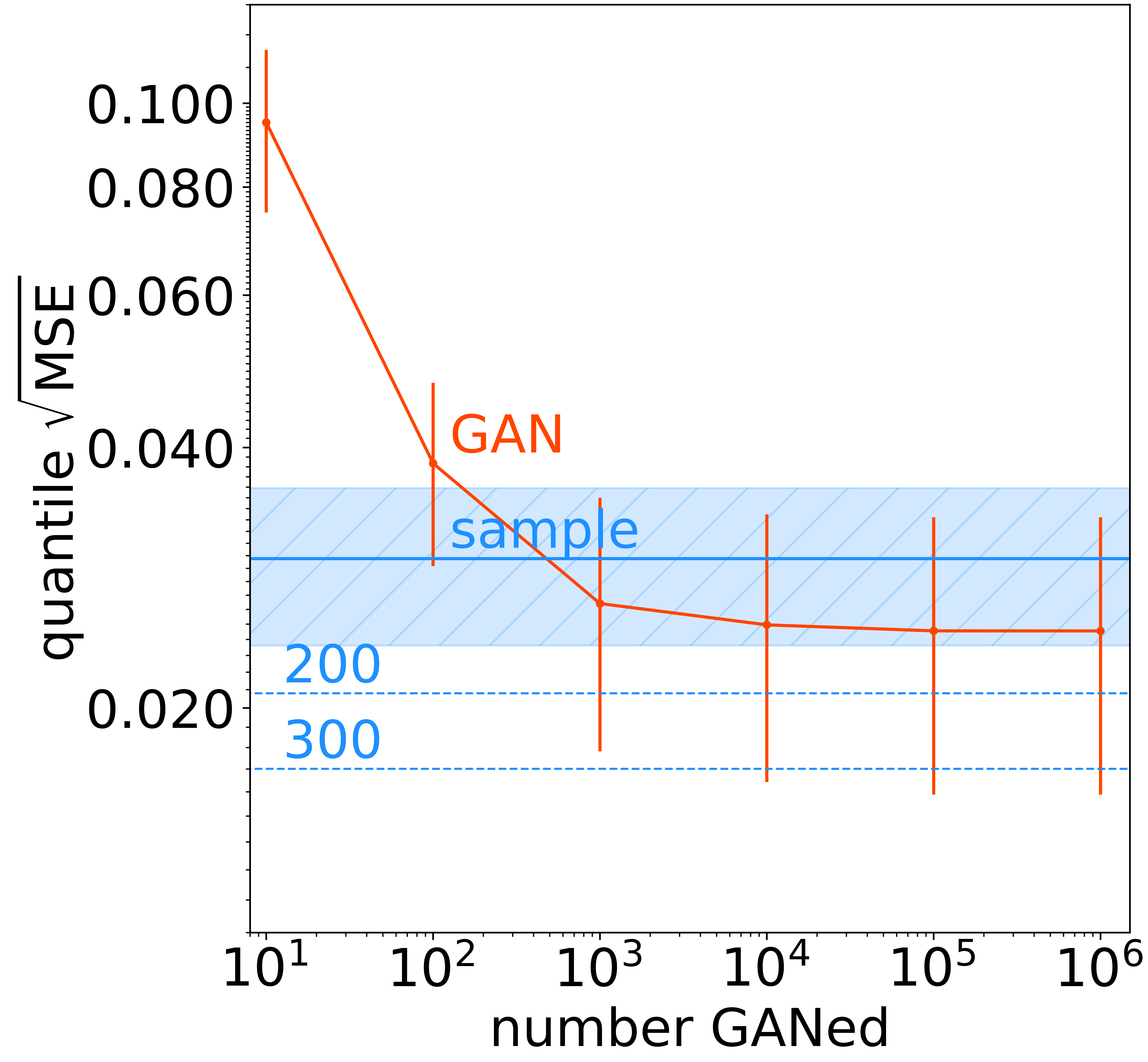
Generative Network

- Calculate MSE for 100 independent training samples
- Train GANs on samples and calculate MSE on for these



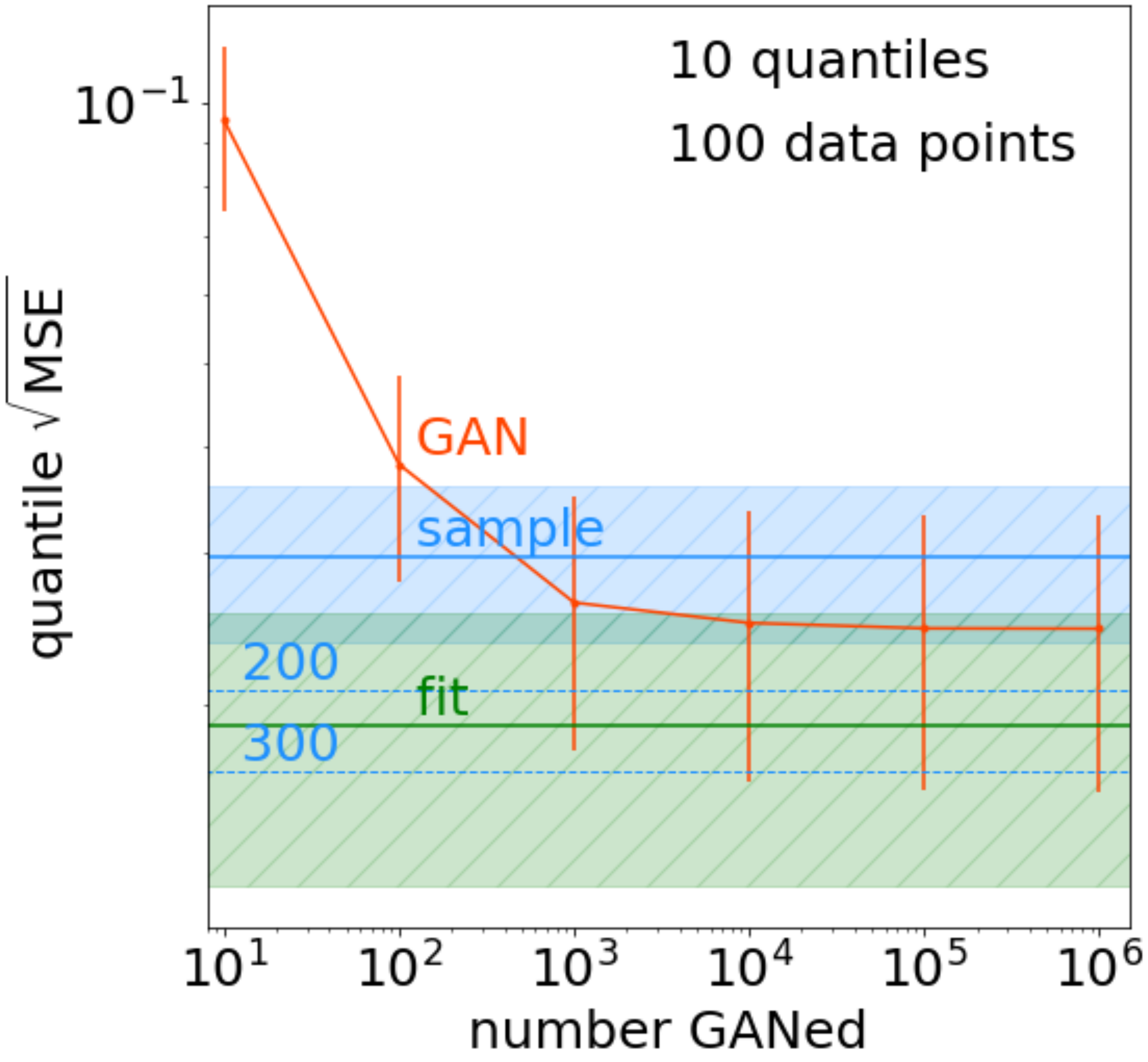
Generative Network

- Calculate MSE for 100 independent training samples
- Train GANs on samples and calculate MSE for these
- Reference lines for 200 and 300 samples



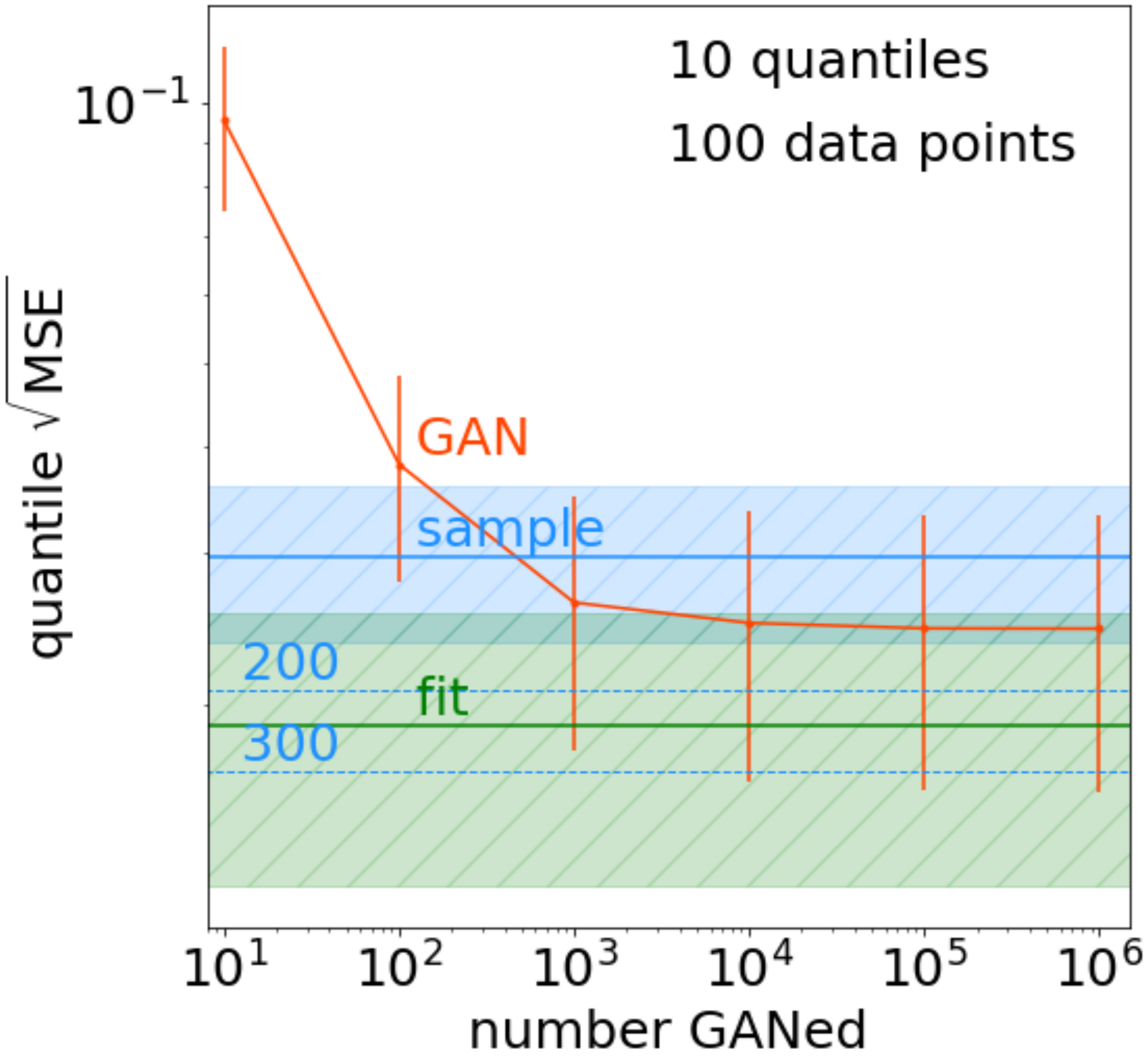
Generative Network

- Calculate MSE for 100 independent training samples
- Train GANs on samples and calculate MSE for these
- Reference lines for 200 and 300 samples
- Calculate MSE for fits on training samples



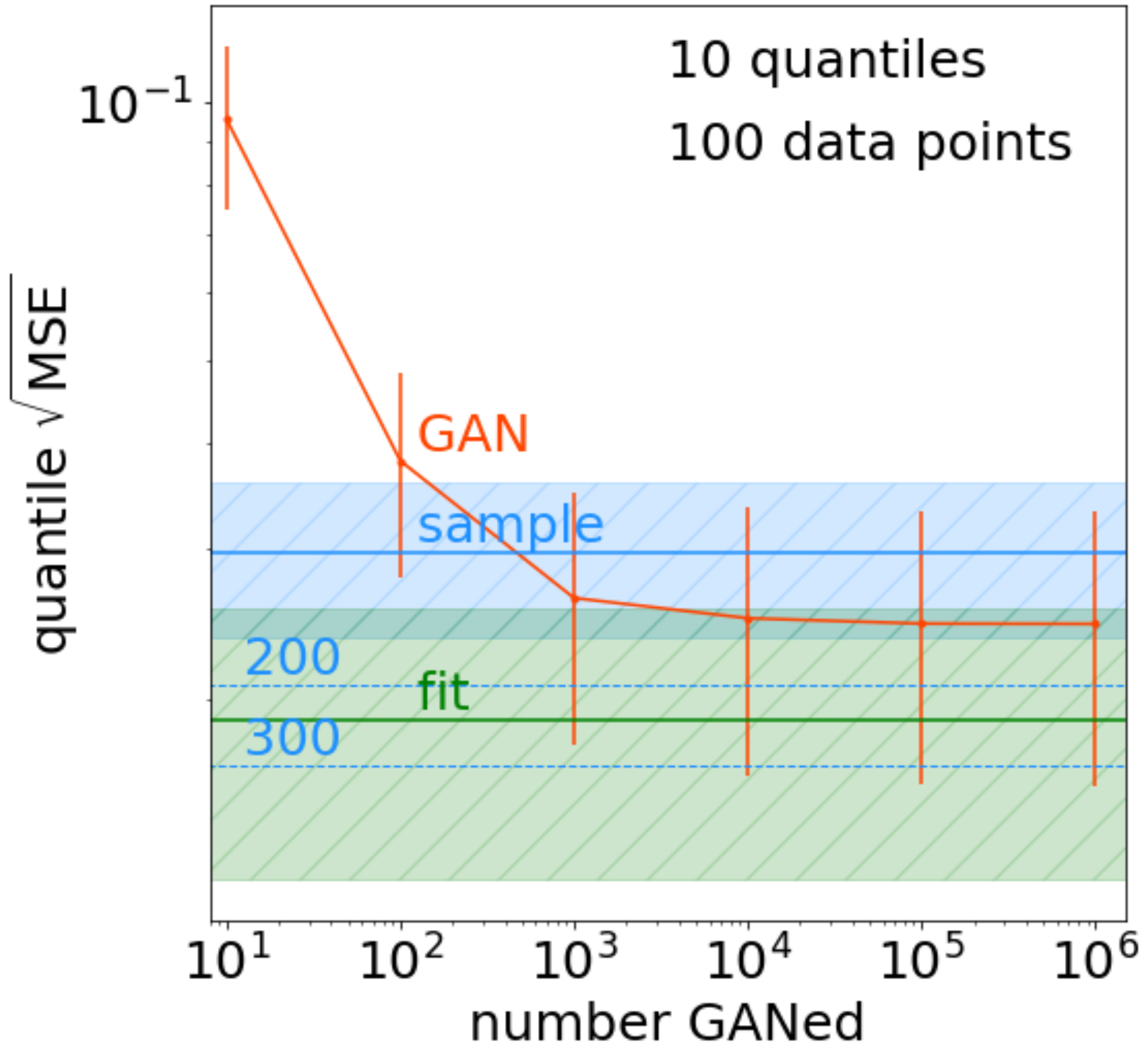
Generative Network

- GAN describes distribution better than training data
- Needs 10,000 GANed points to match 150 true points



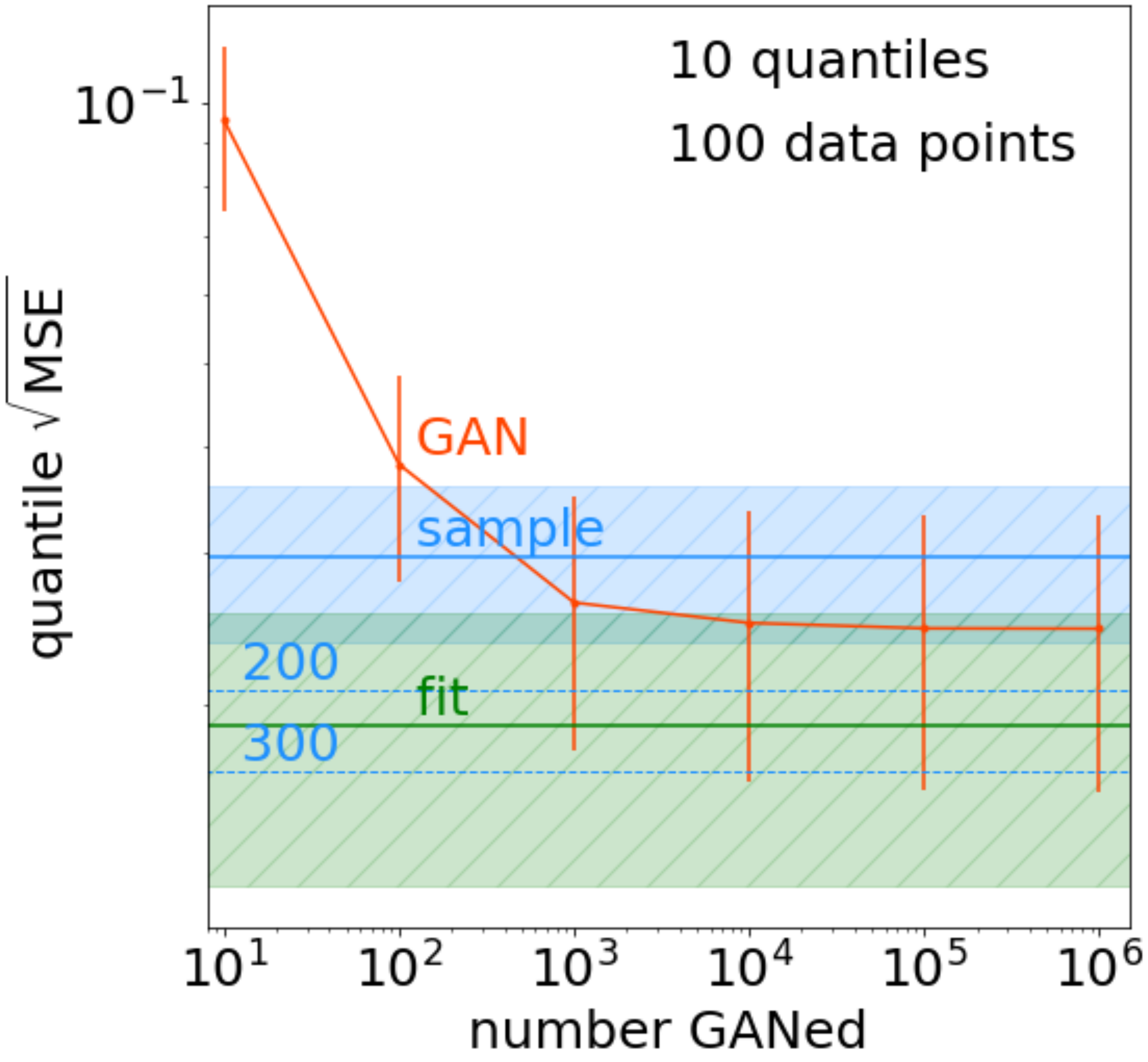
Generative Network

- How is this possible?



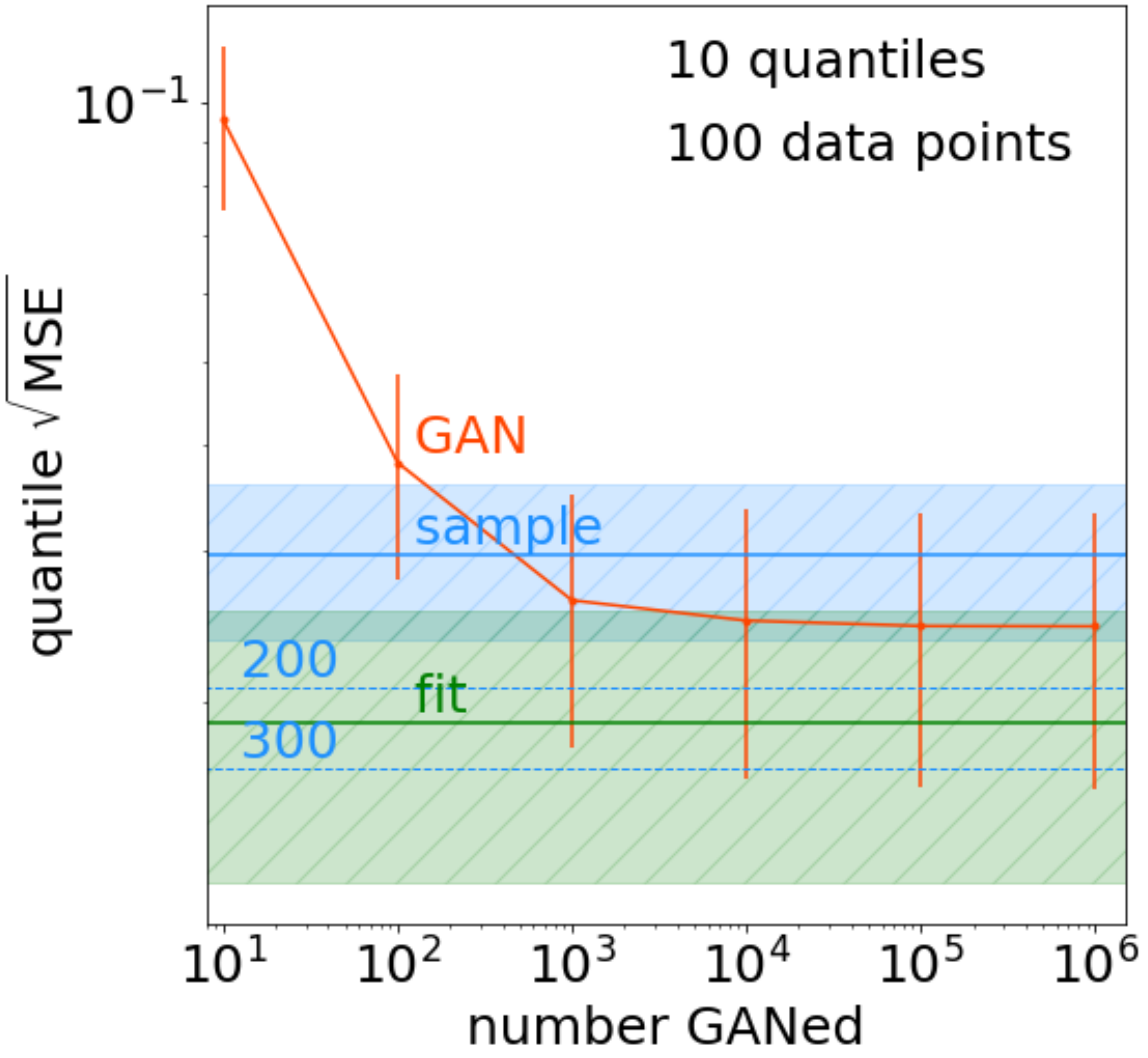
Generative Network

- How is this possible?
- In terms of information:
  - sample: only data points
  - fit: data + true function
  - GAN: data + smooth, continuous function



Generative Network

- How is this possible?
- In terms of information:
  - sample: only data points
  - fit: data + true function
  - GAN: data + smooth, continuous function
- This allows the GAN to interpolate



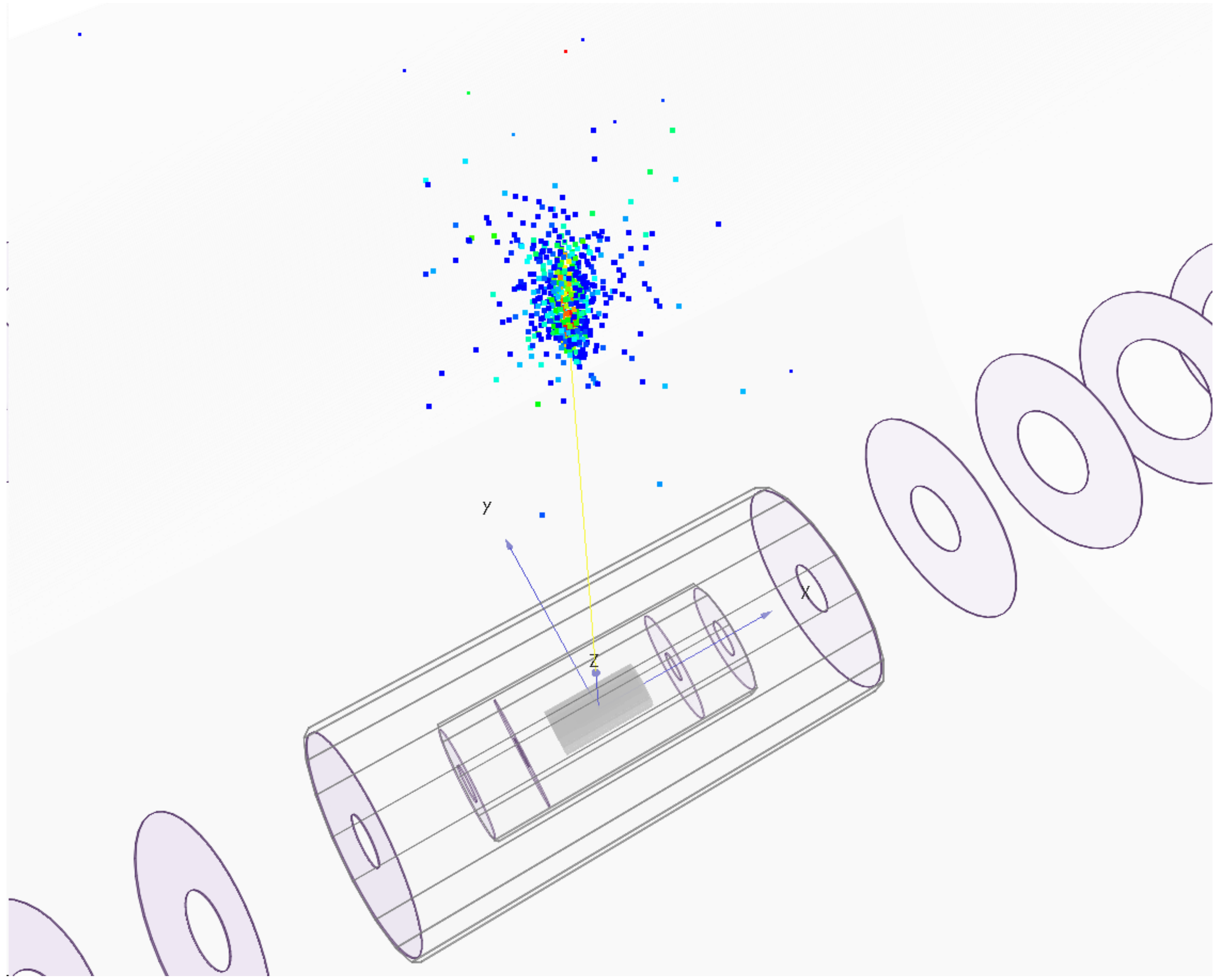
Statistical Properties

- For interpolatable dataset:
- Info(N real points) < Info(M GANed points)
- GANs have potential to amplify dataset
- Highly promising for application as simulation accelerators

Butter et al.: **GANplifying Event Samples:** 2008.06545

Shower Simulation

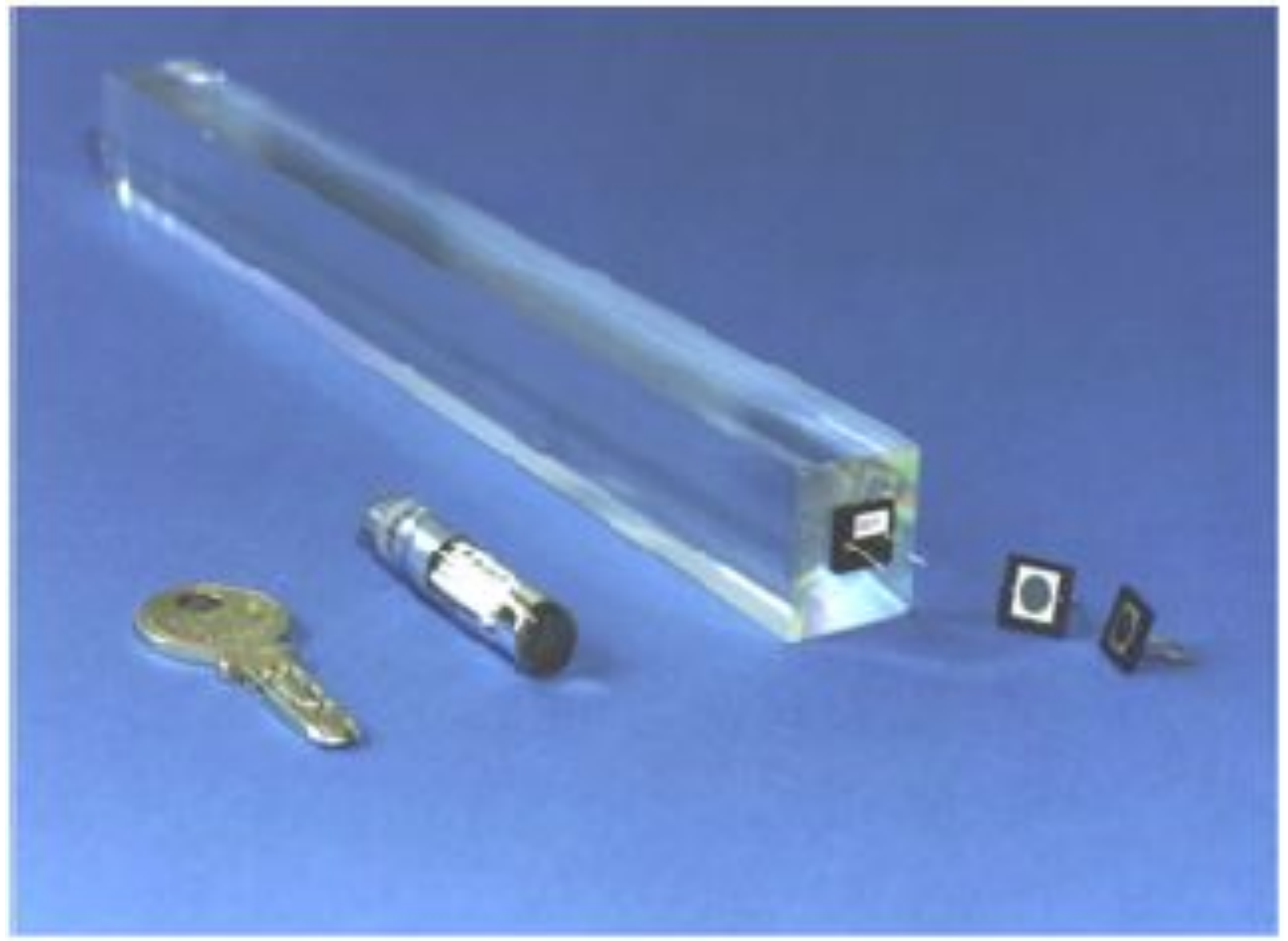
Simulator	Hardware	Batchsize	Time/shower*
GEANT4	CPU	N/A	4082 ± 170 ms



* average time for 10-100 GeV showers

- Classically done using Geant4
- First principle simulation modelling individual particle interactions
- Very computationally expensive
- Timing even more significant for higher luminosities
- Significant resources needed

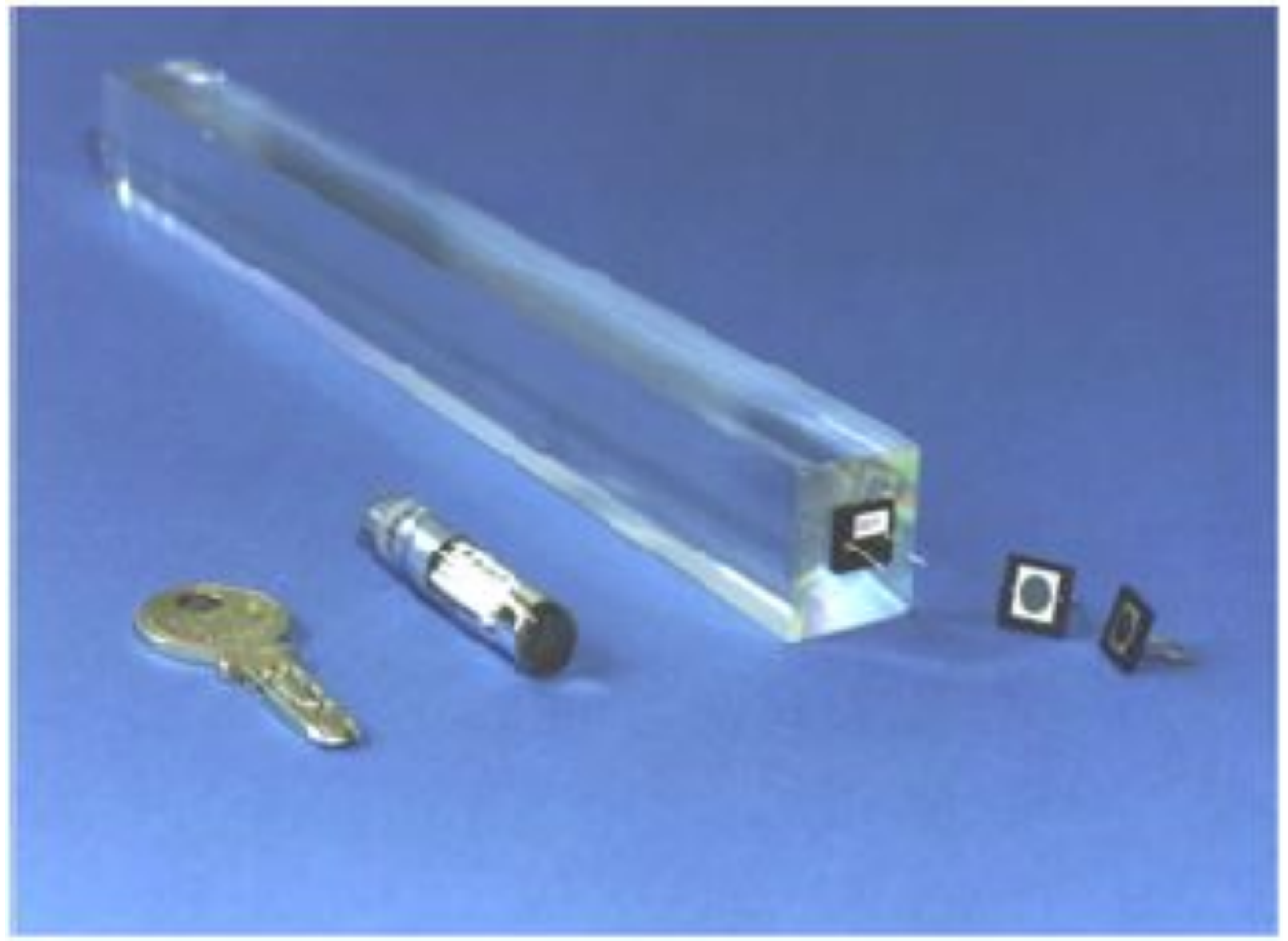
Calorimeters



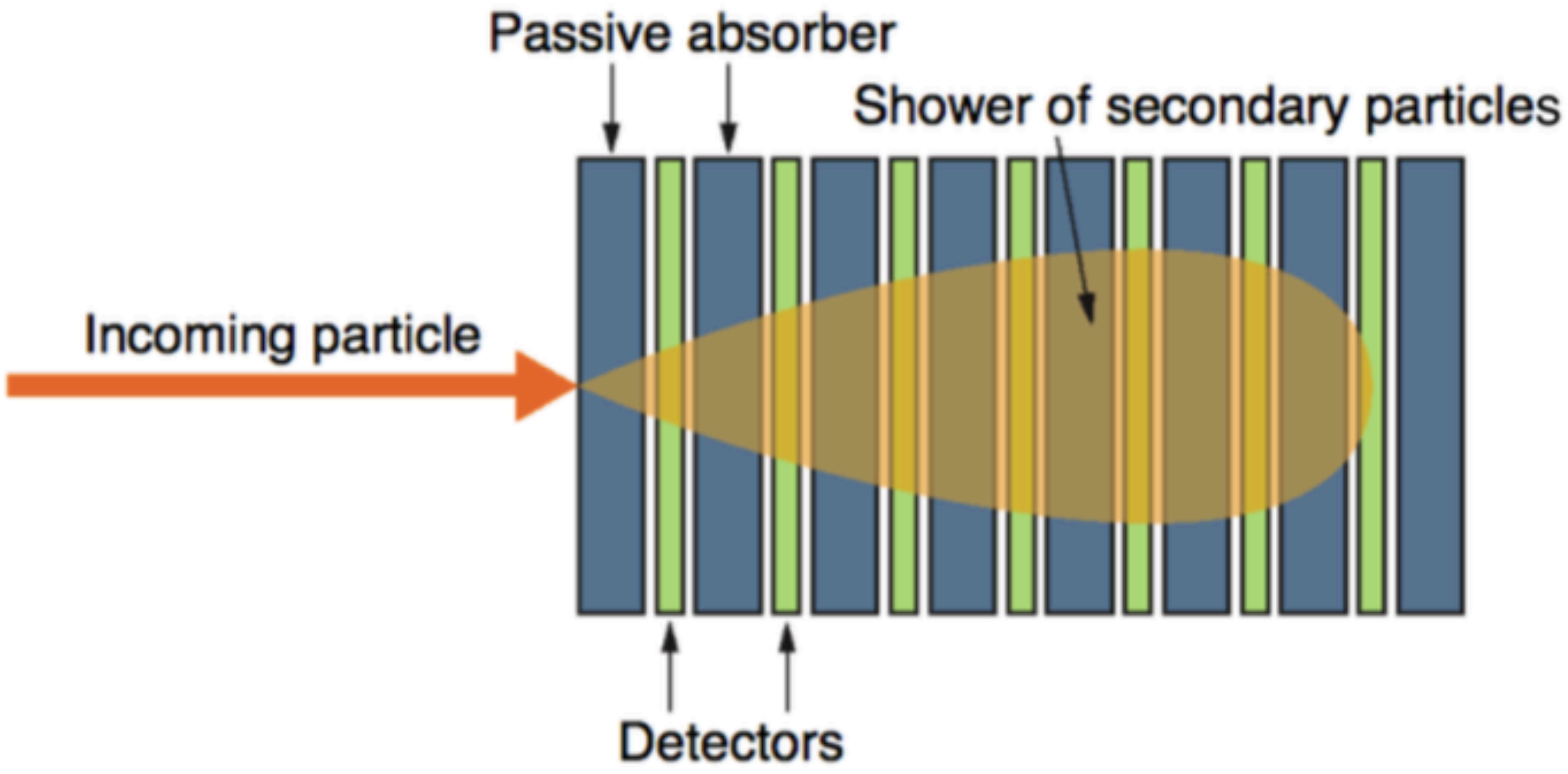
http://ihp-lx.ethz.ch/CompMethPP/lhc/pictures/

- Homogeneous Calorimeter
  - Made up of single active block
  - Great energy resolution
  - No spatial information

Calorimeters

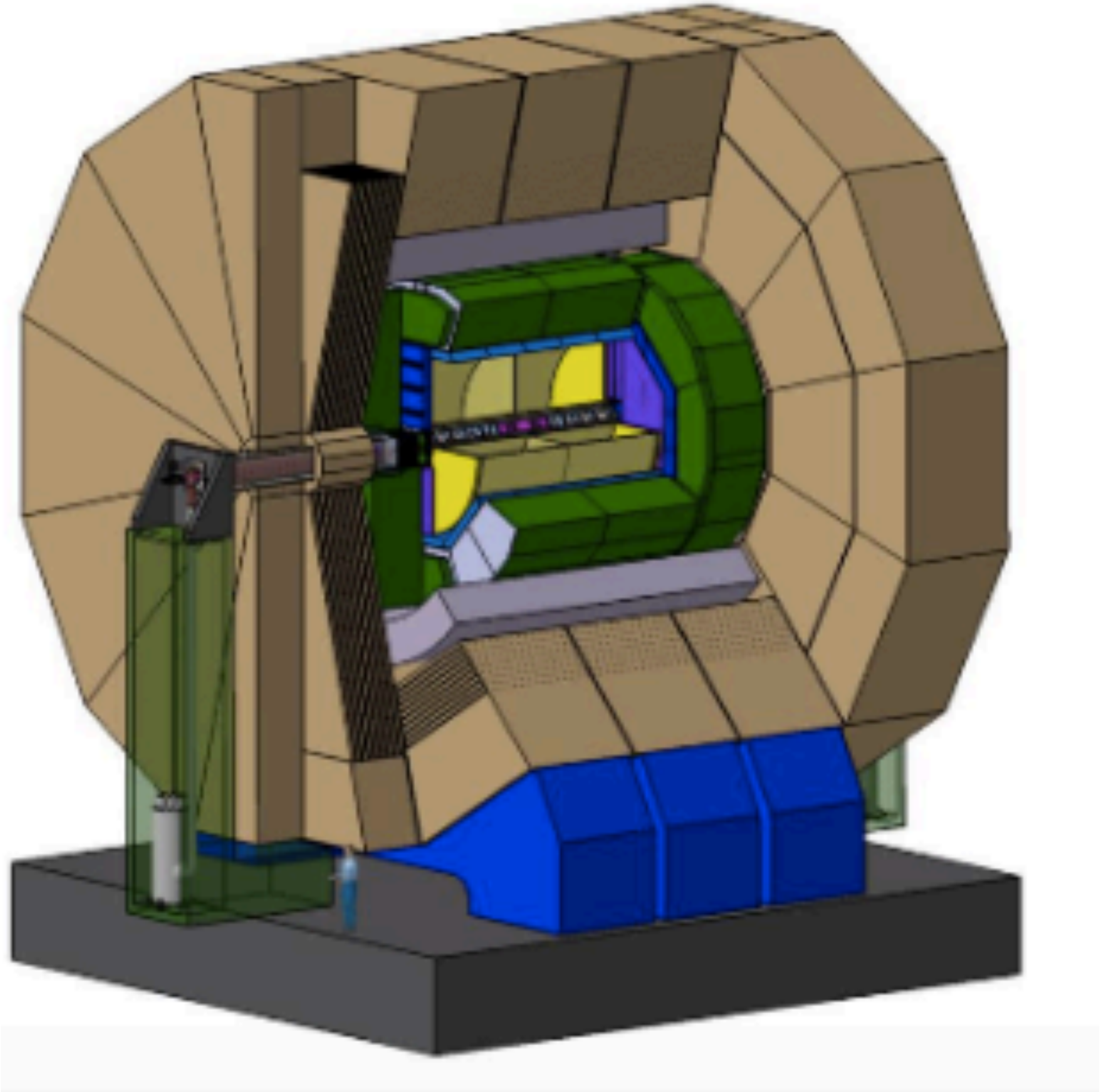


http://ihp-lx.ethz.ch/CompMethPP/lhc/pictures/



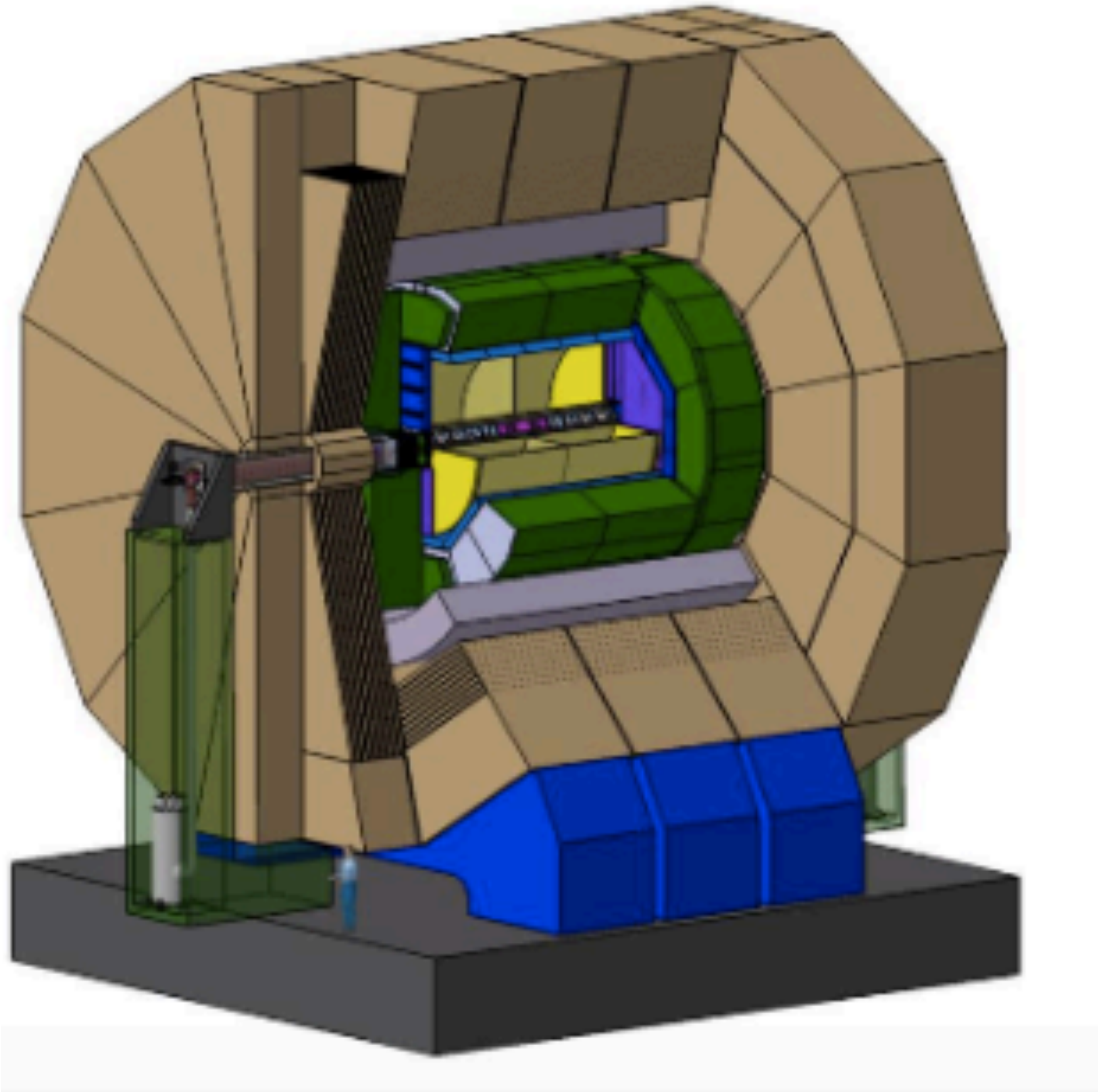
- Homogeneous Calorimeter
  - Made up of single active block
  - Great energy resolution
  - No spatial information
- Sampling Calorimeter
  - Alternating passive and active layers
  - Only part of energy directly observed
  - Maintains spacial structure

ILD Calorimeter



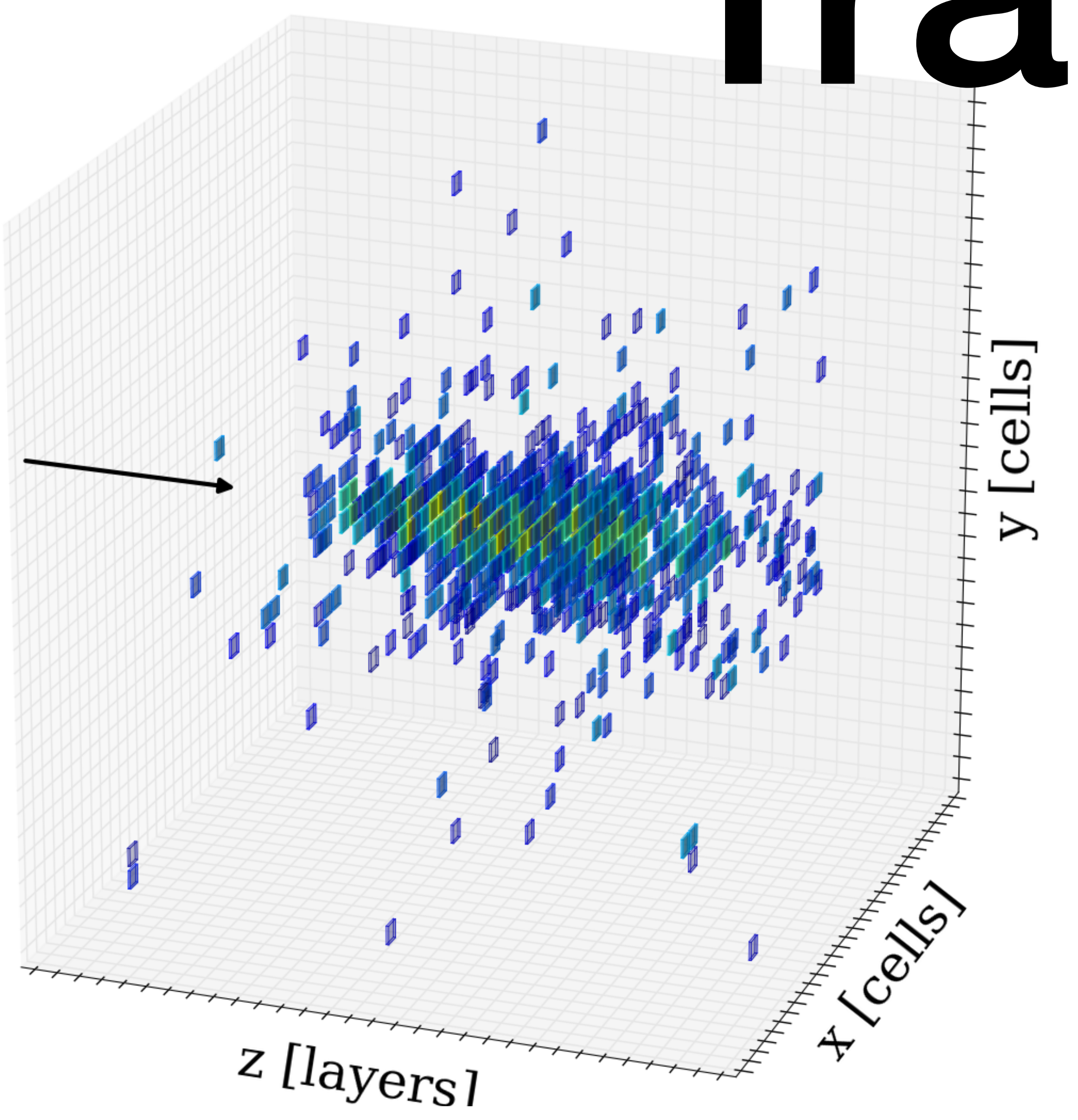
- International Large Detector (ILD)
  - Detector for International Linear Collider (ILC)

ILD Calorimeter

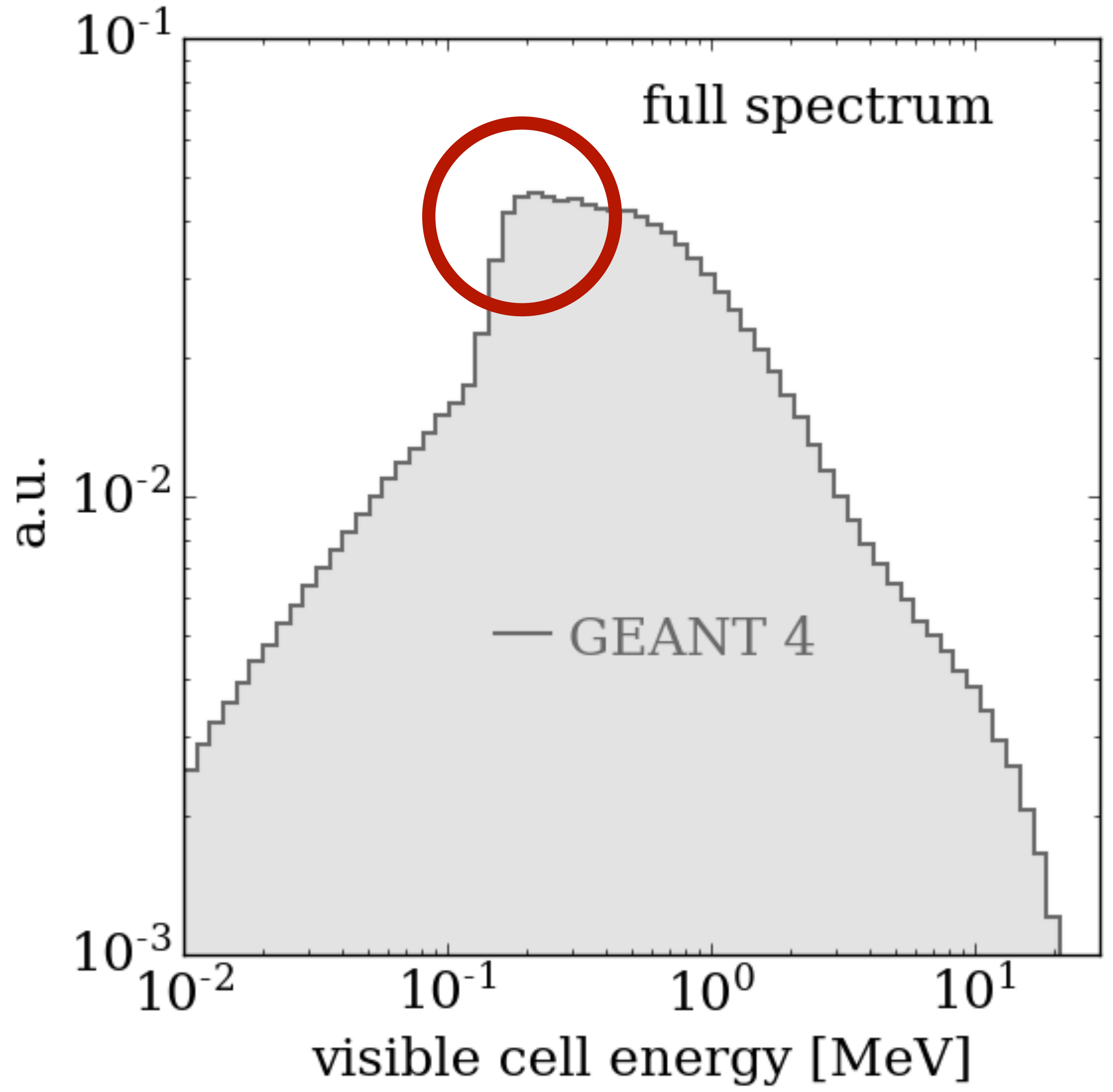


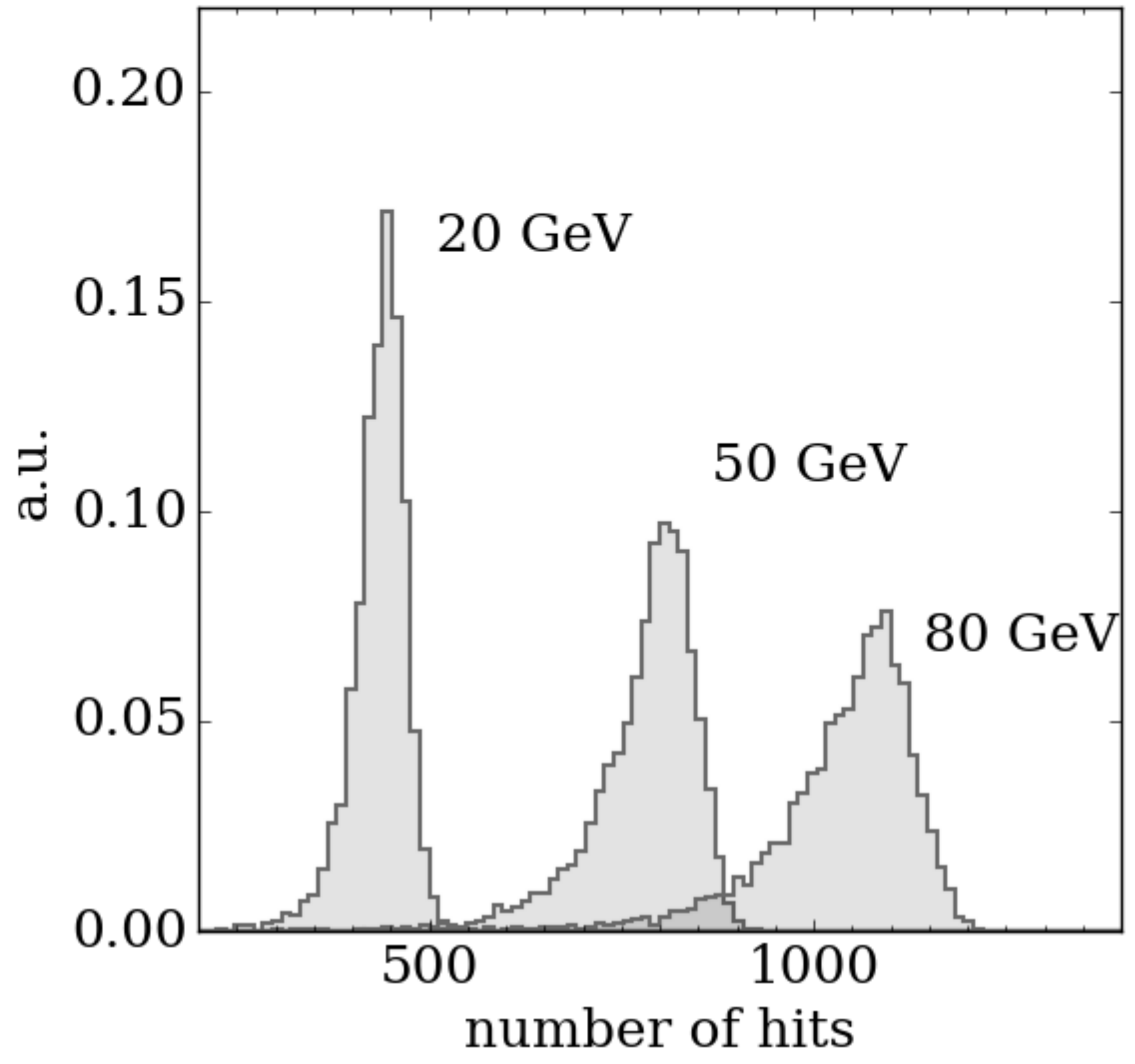
- International Large Detector (ILD)
  - Detector for International Linear Collider (ILC)
- ILD electromagnetic calorimeter
  - Highly granular sampling calorimeter
  - Active silicon, passive tungsten
  - 30 layers, 5mm x 5mm cells

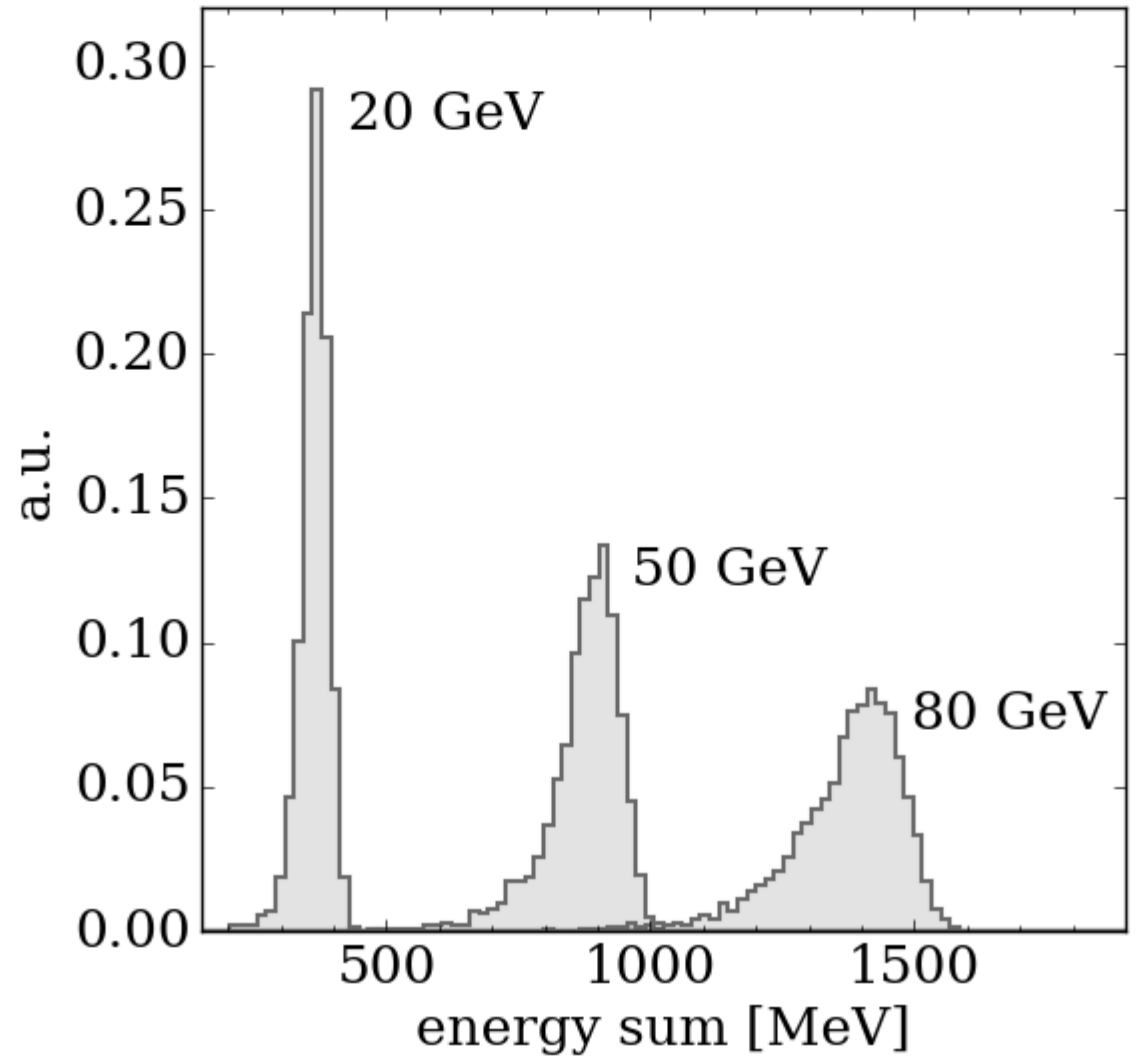
Training Dataset



- 950k photon showers
- Continuous incident energy from 10 GeV to 100 GeV
- Constant incident point and angle
- Each shower 30x30x30 image

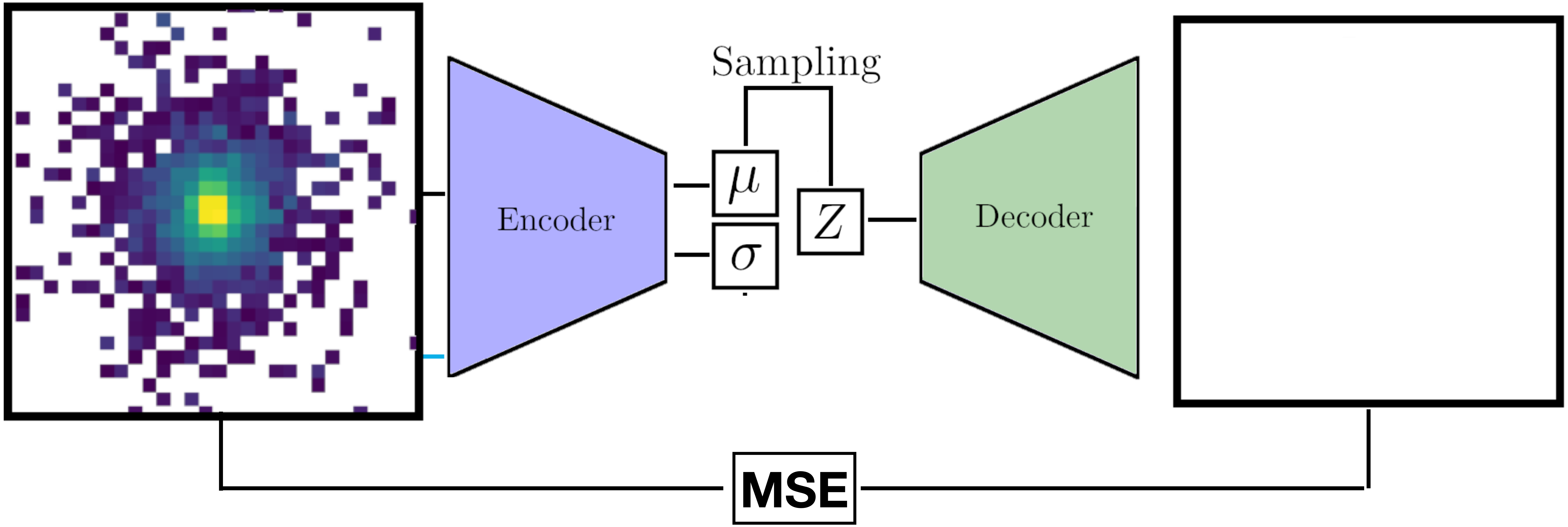






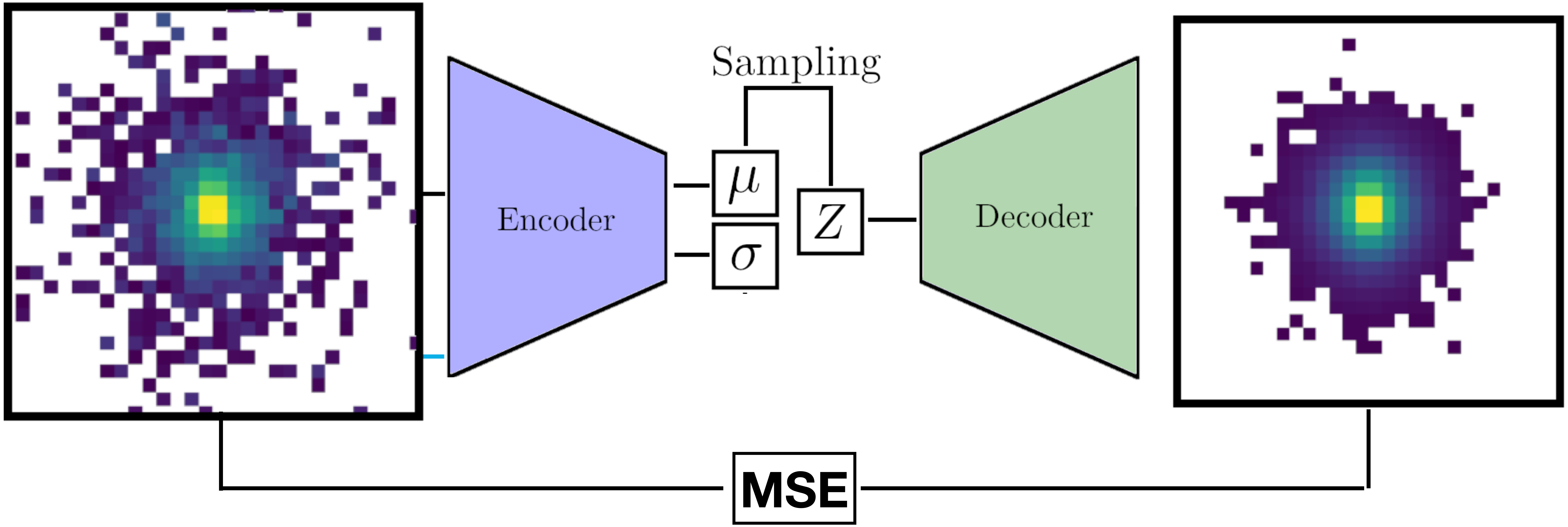
Buhmann et al.: **Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed:** 2005.05334

VAEs and Sparsity



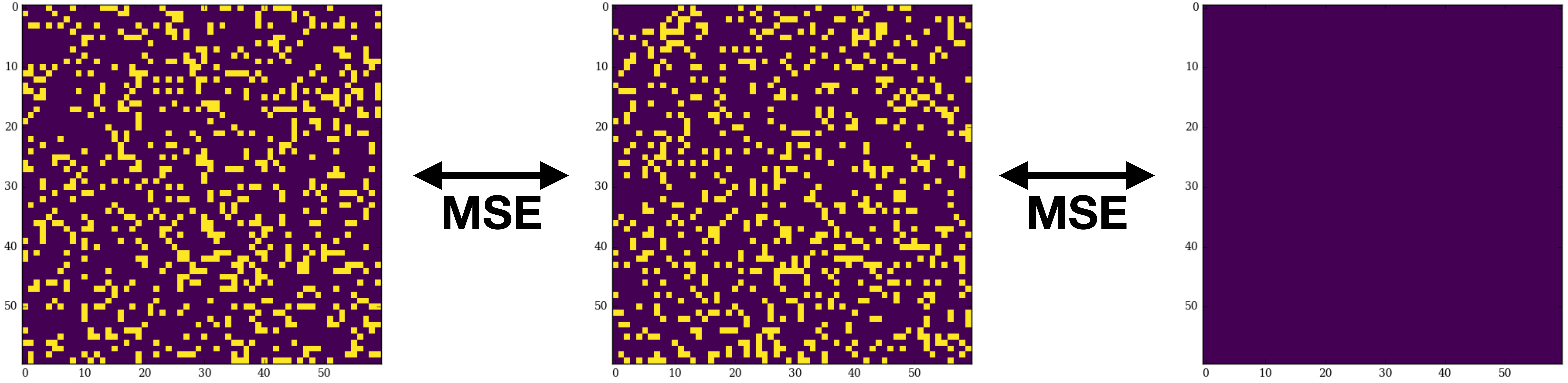
- Initial VAE tests on 2D version of Data
  - Mean Squared Error to compare input and reconstruction

VAEs and Sparsity

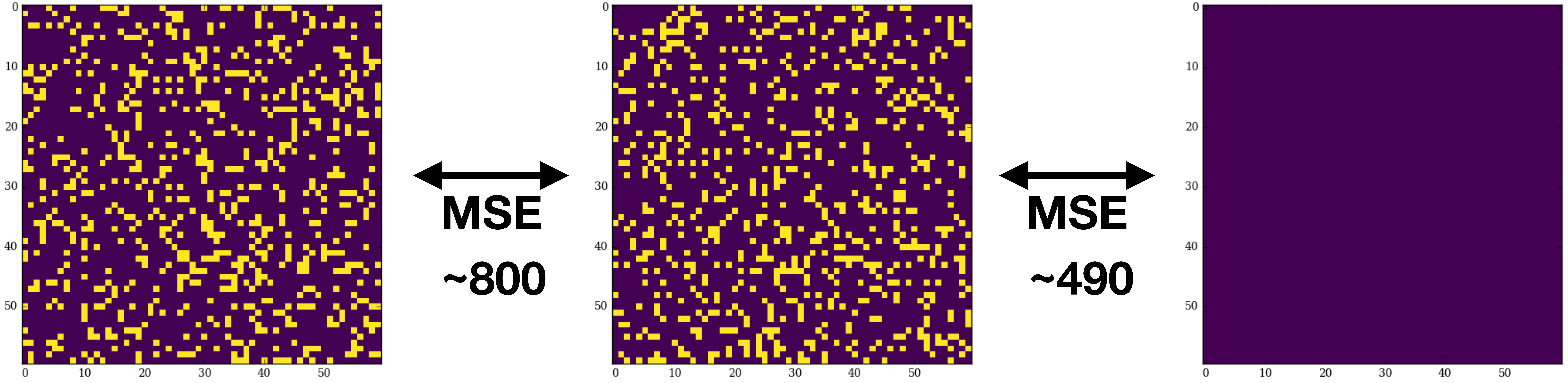


- Initial VAE tests on 2D version of Data
  - Mean Squared Error to compare input and reconstruction
  - Unable to reproduce outer hits

VAEs and Sparsity

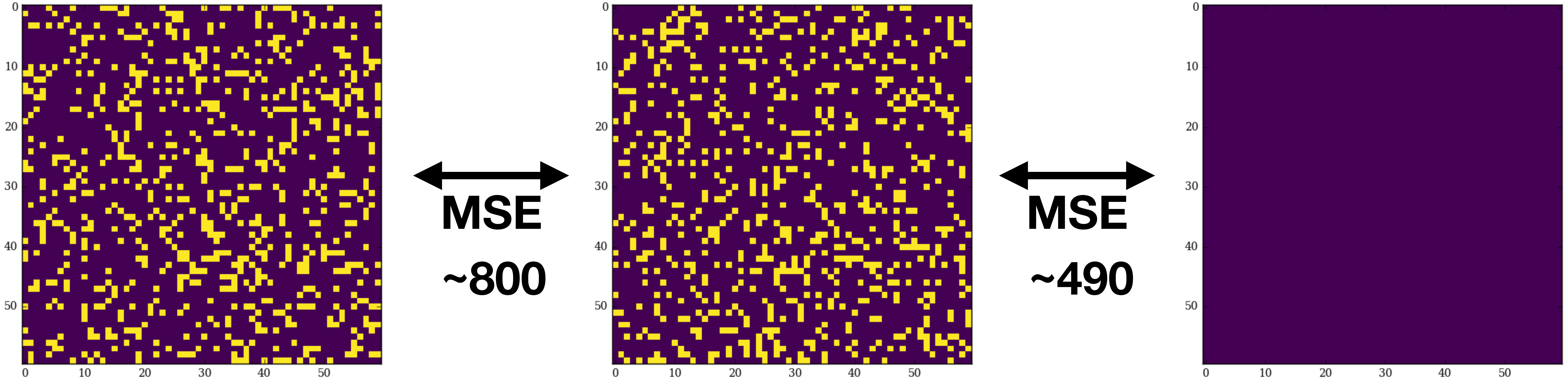


VAEs and Sparsity



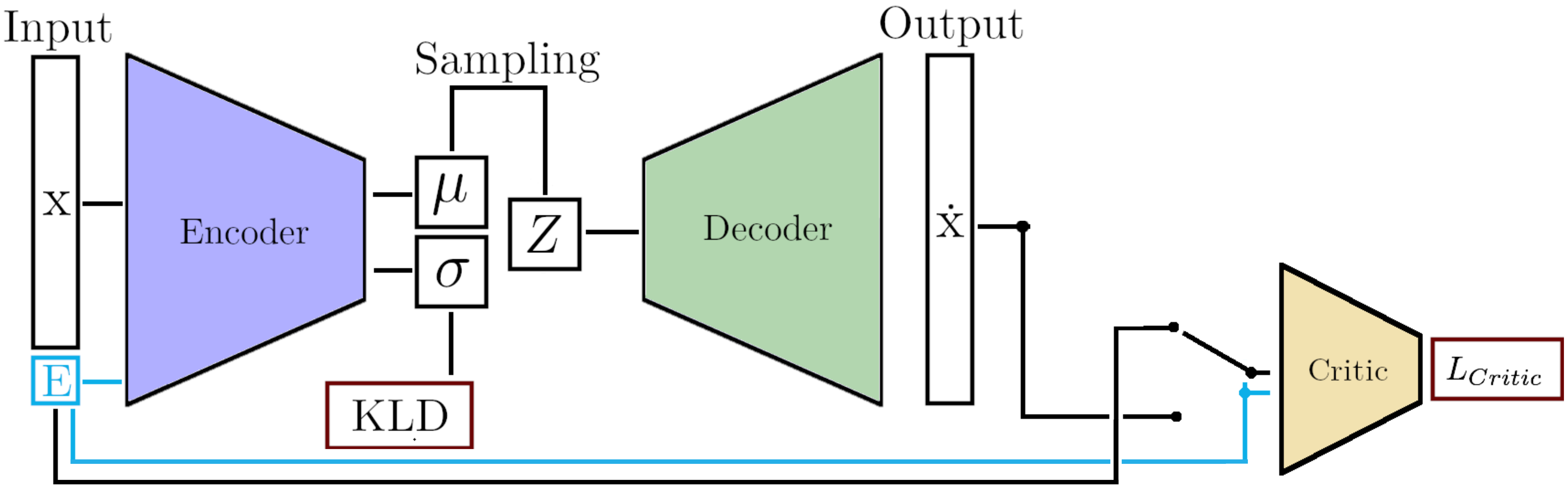
- MSE between two random sparse images is larger than between random and empty

VAEs and Sparsity



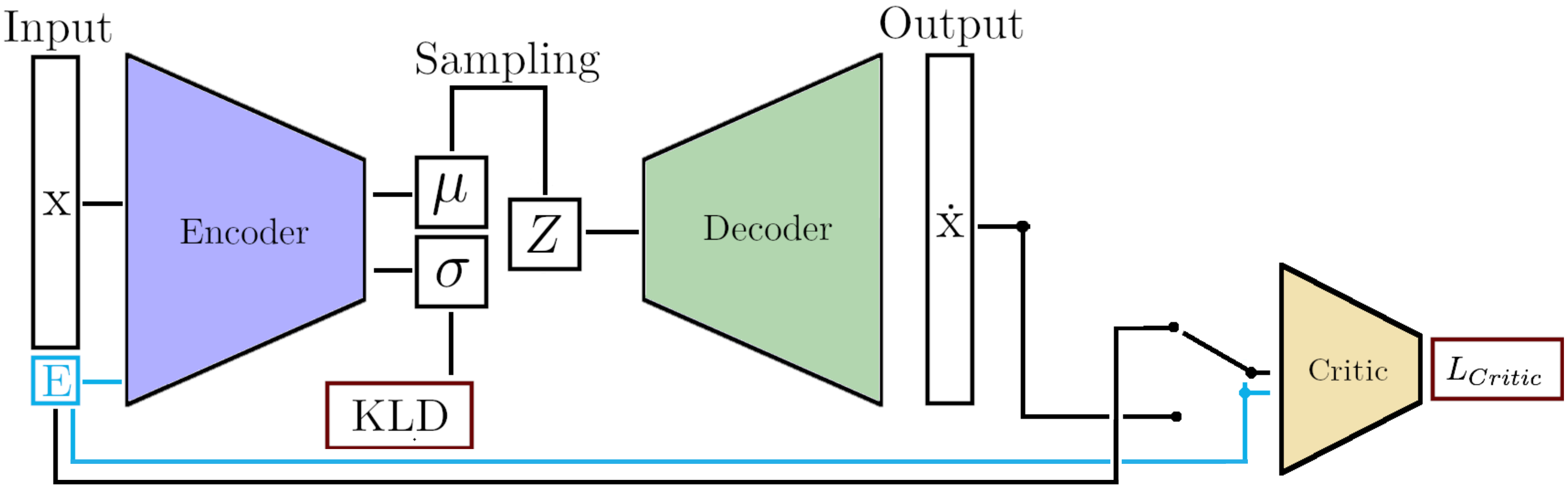
- MSE between two random sparse images is larger than between random and empty
- For outer shower hits:
    - Presence more important than exact position
- Needs more fitting reconstruction loss than MSE

VAE-GAN



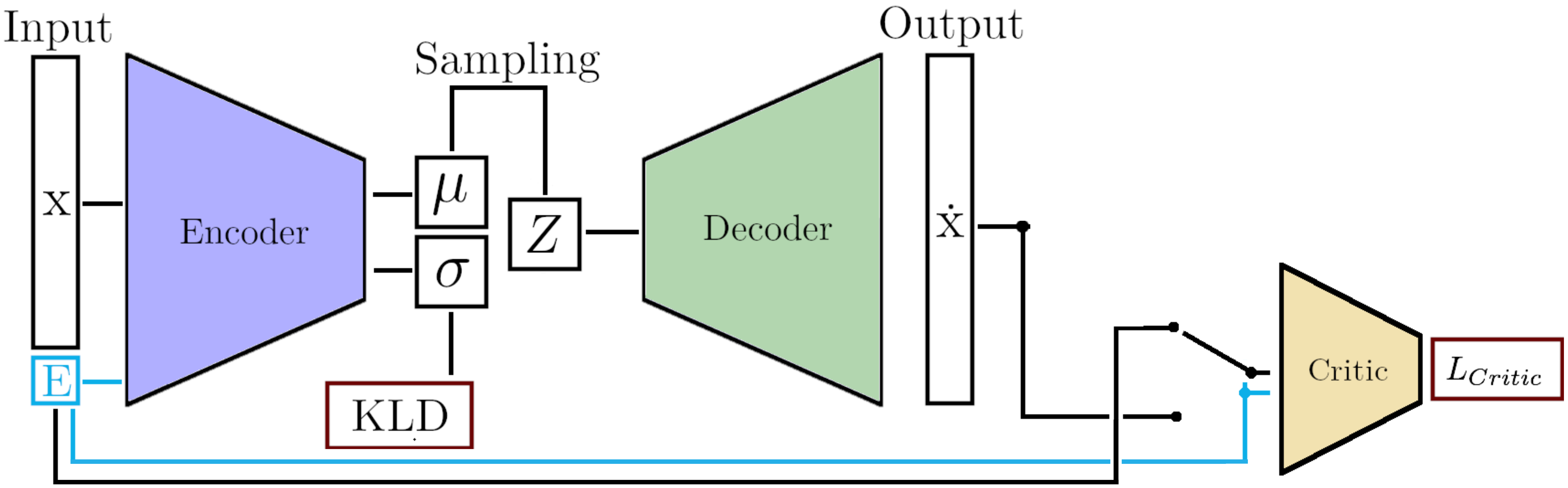
- Combines VAE and GAN ideas

VAE-GAN



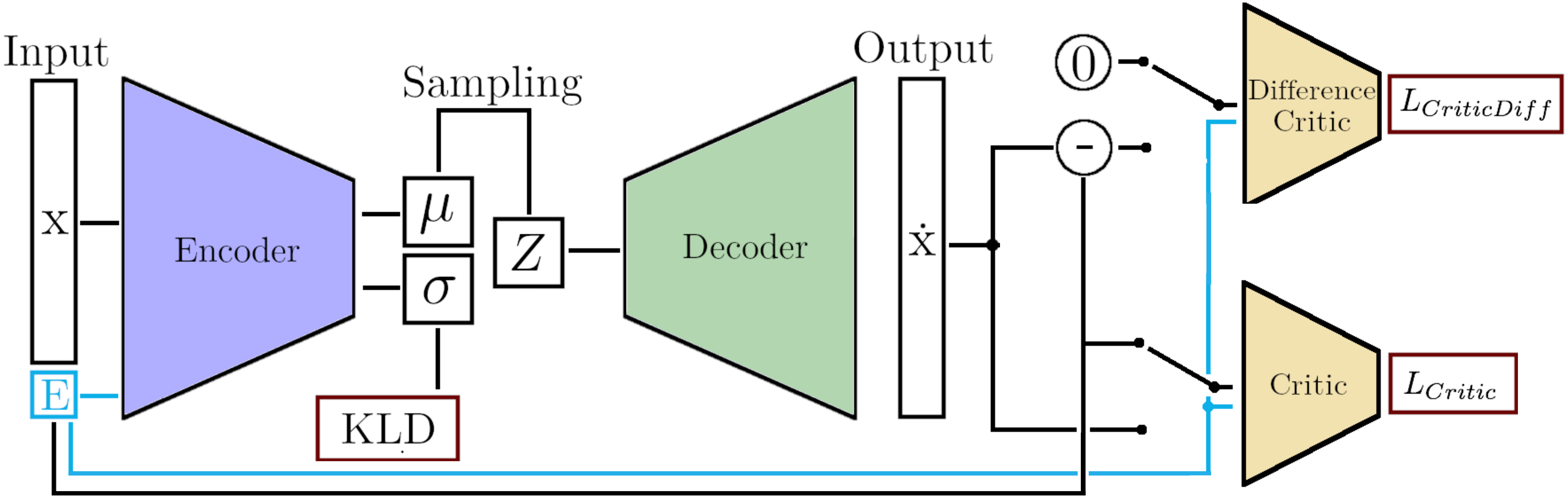
- Combines VAE and GAN ideas
  - Enhances pixel wise Mean Squared Error loss with GAN-like adversarial network

VAE-GAN



- Combines VAE and GAN ideas
  - Enhances pixel wise Mean Squared Error loss with GAN-like adversarial network
  - Significant improvement in generated shower quality

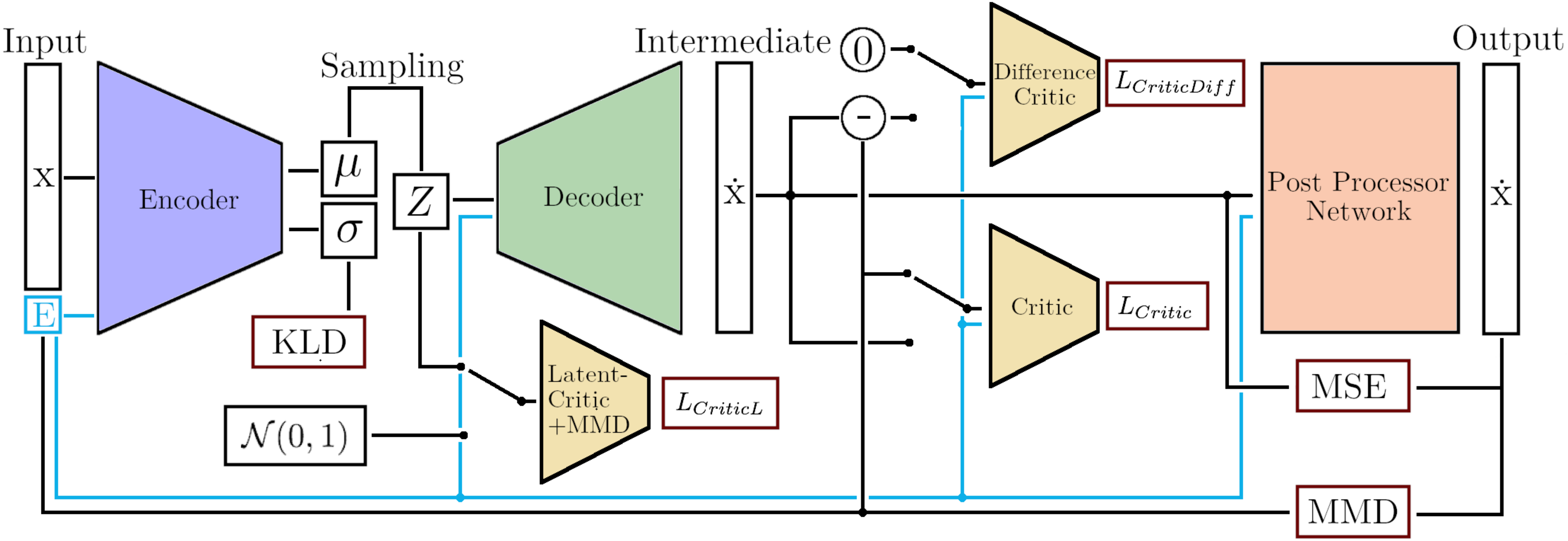
Bounded Information Bottleneck AutoEncoder



- Further expansion of the VAE-GAN structure
- Critic network that judges shower quality
- Second critic that judges reconstruction

Slava Voloshynovskiy et al.:
Information bottleneck through variational glasses: 1912.00830

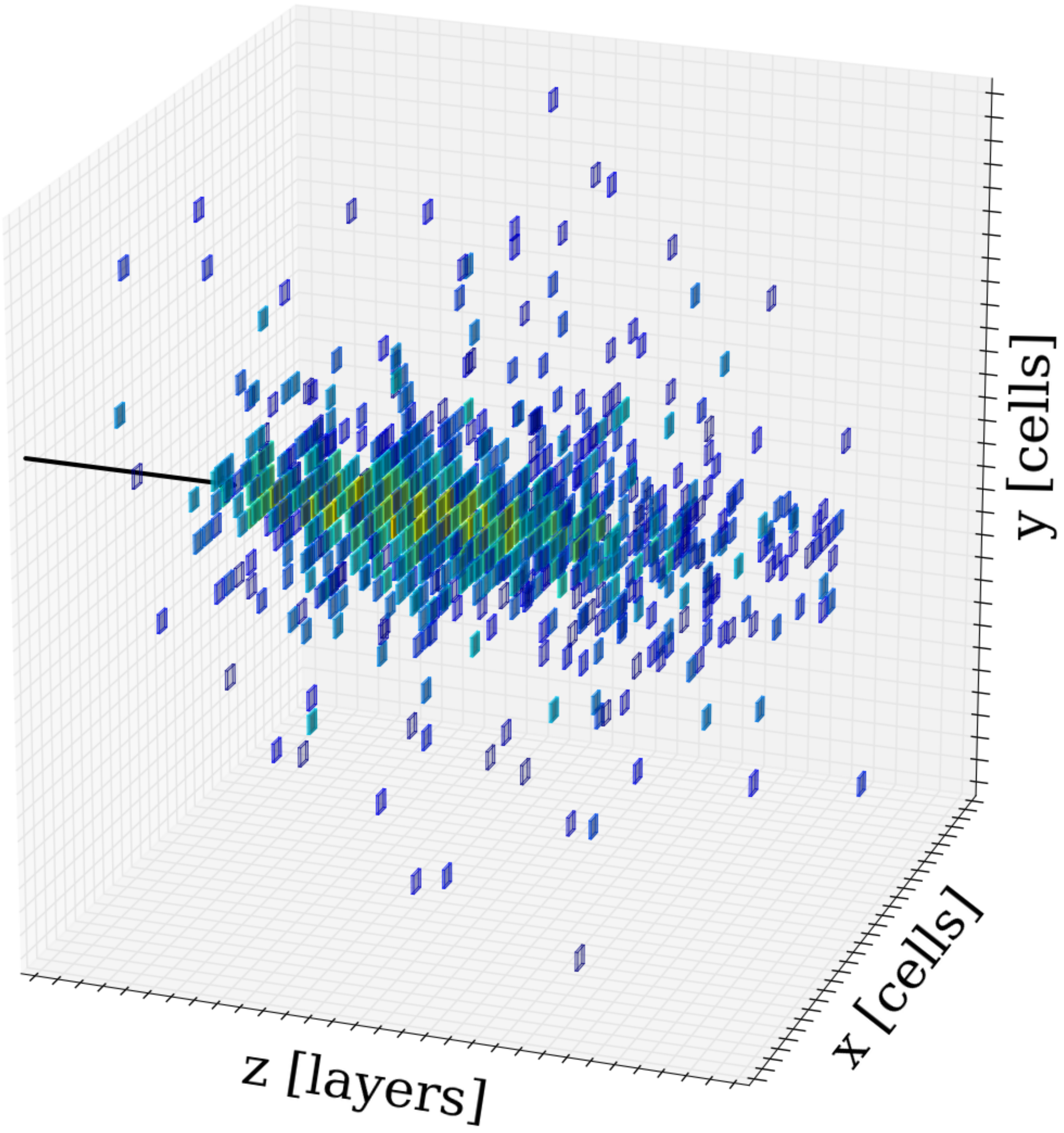
BIB-AE Post Processor



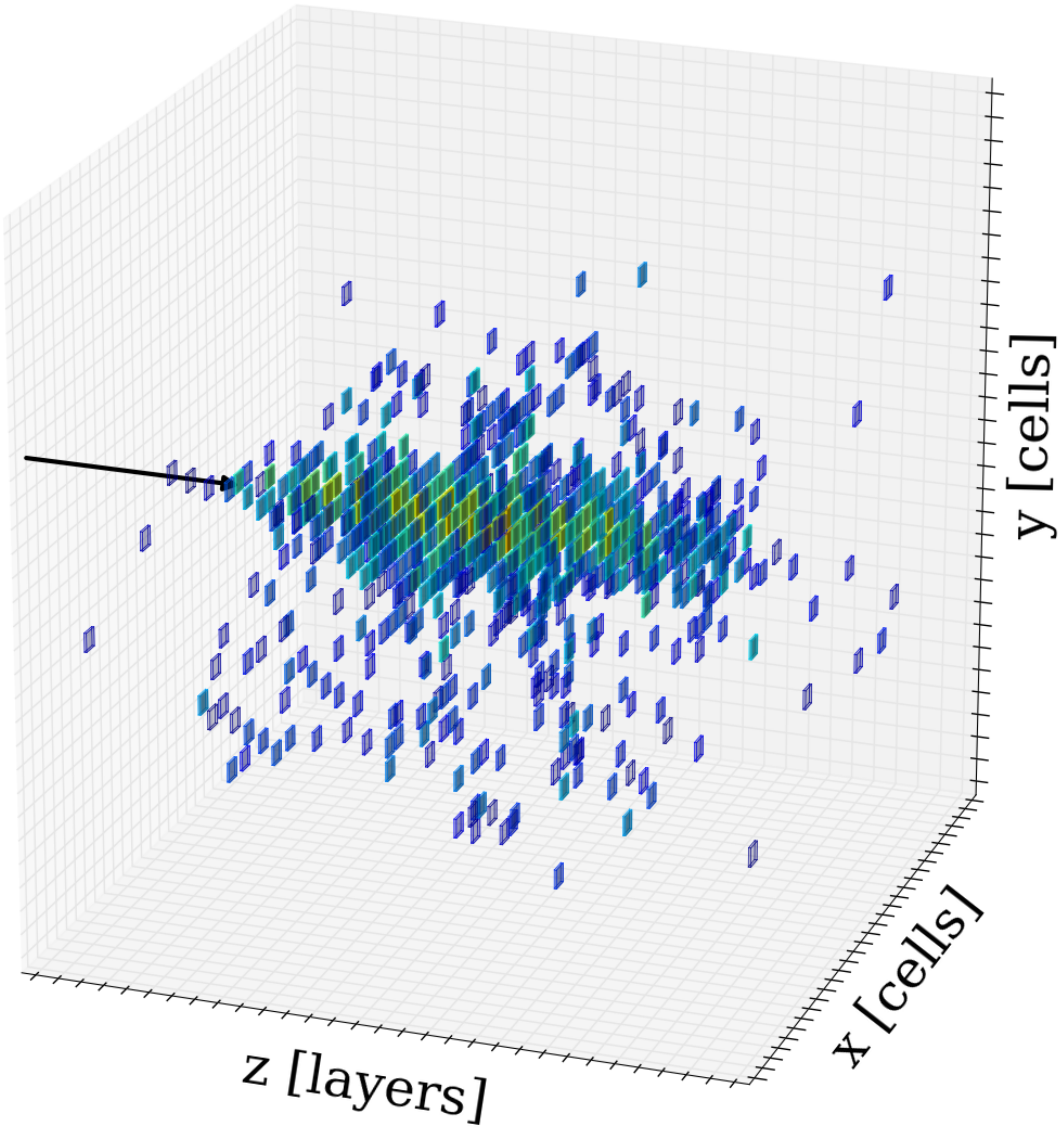
- Additional Critic for latent regularisation
- Final Post Processor Network for fine tuning

Results

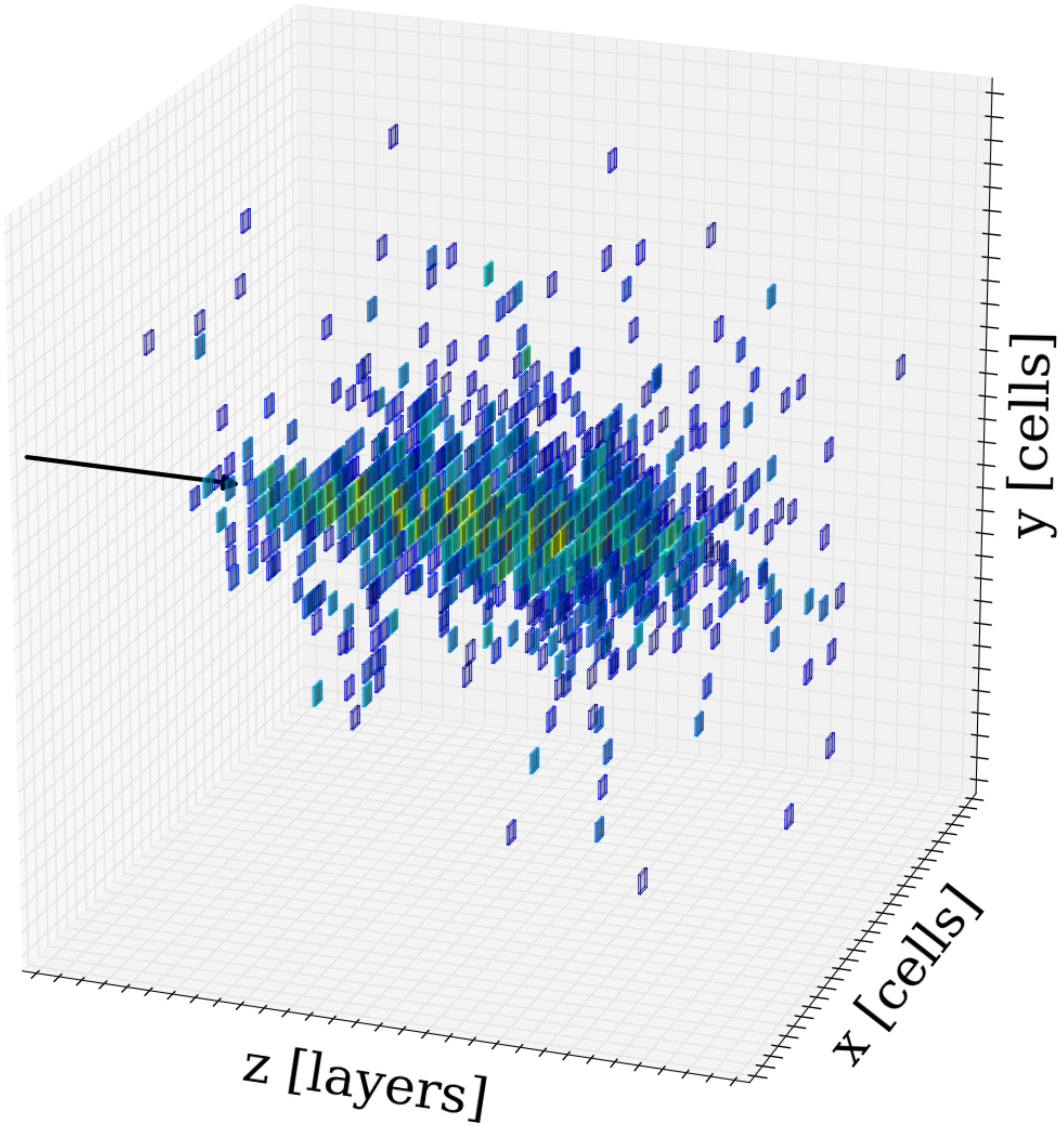
GAN



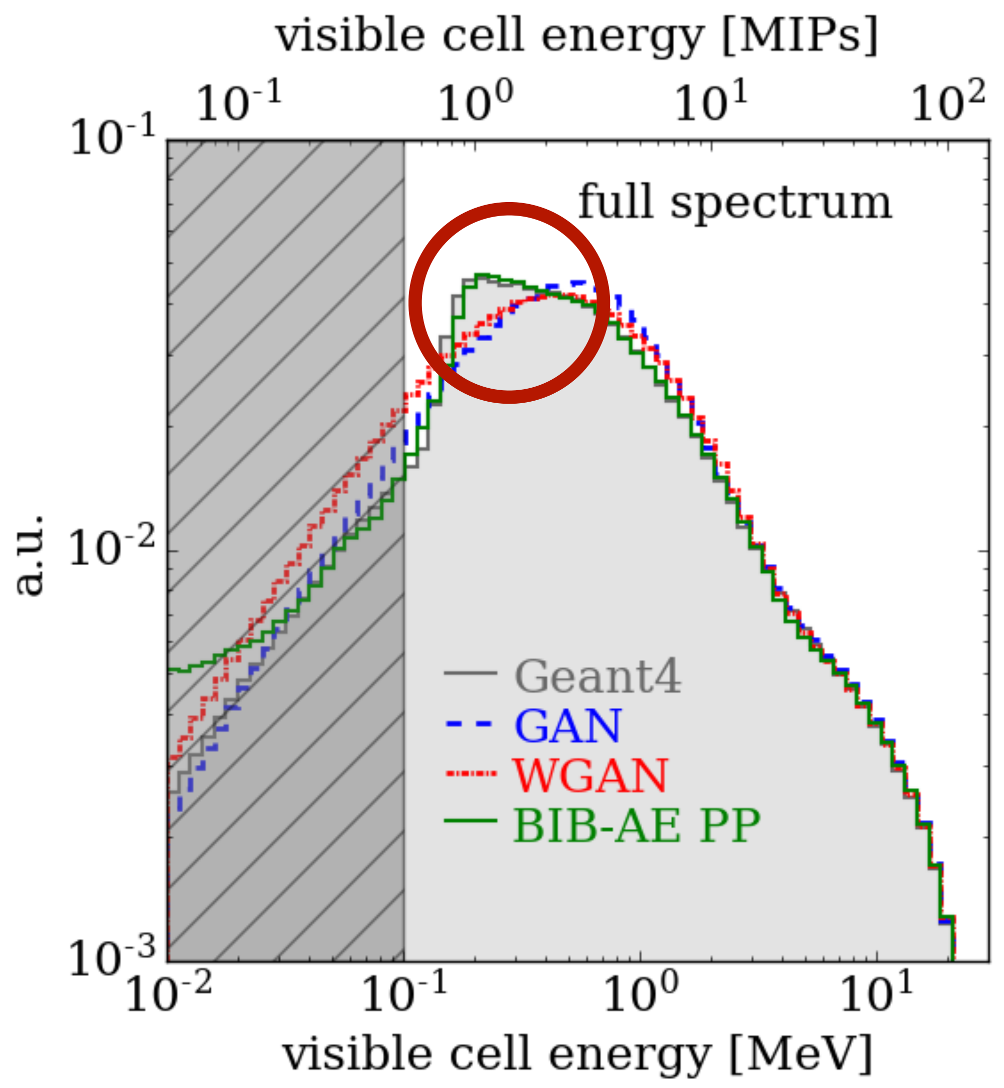
WGAN



BIB-AE

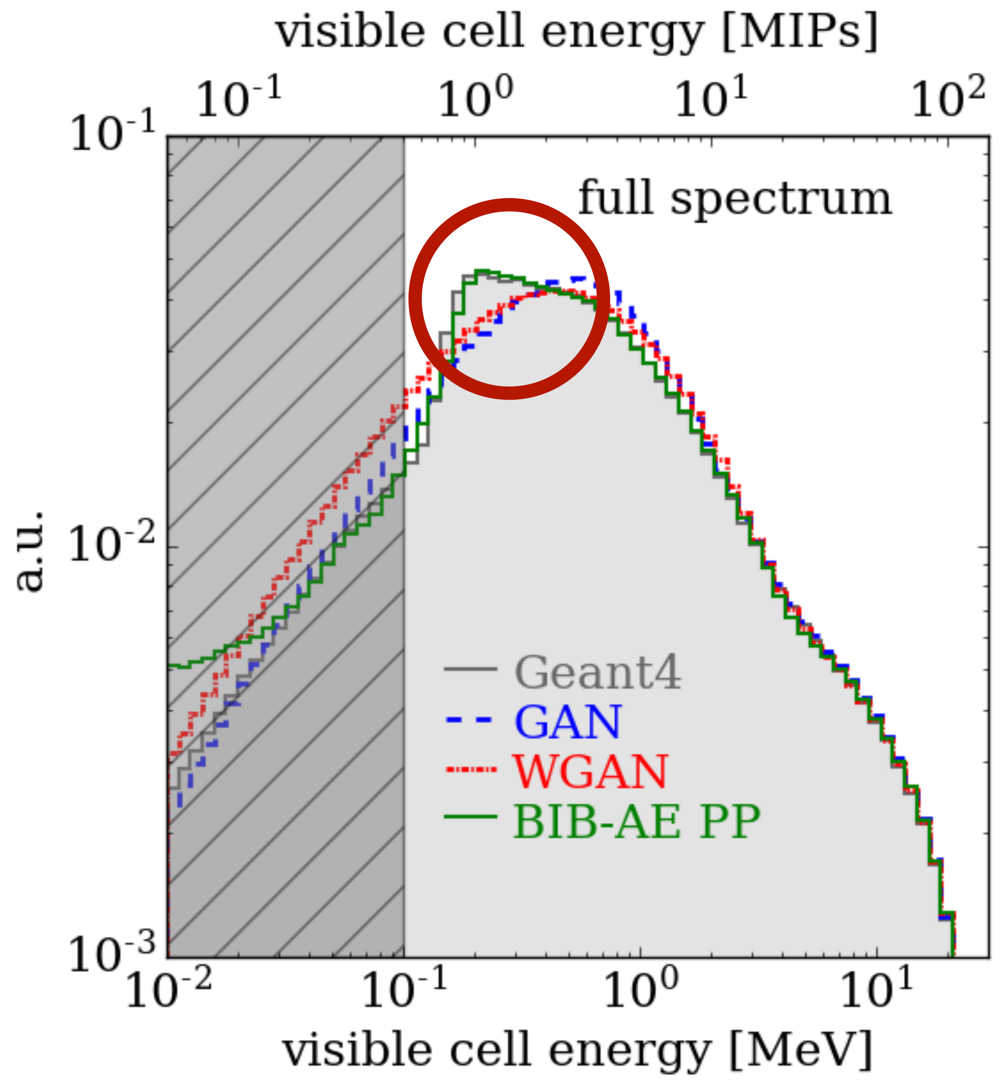


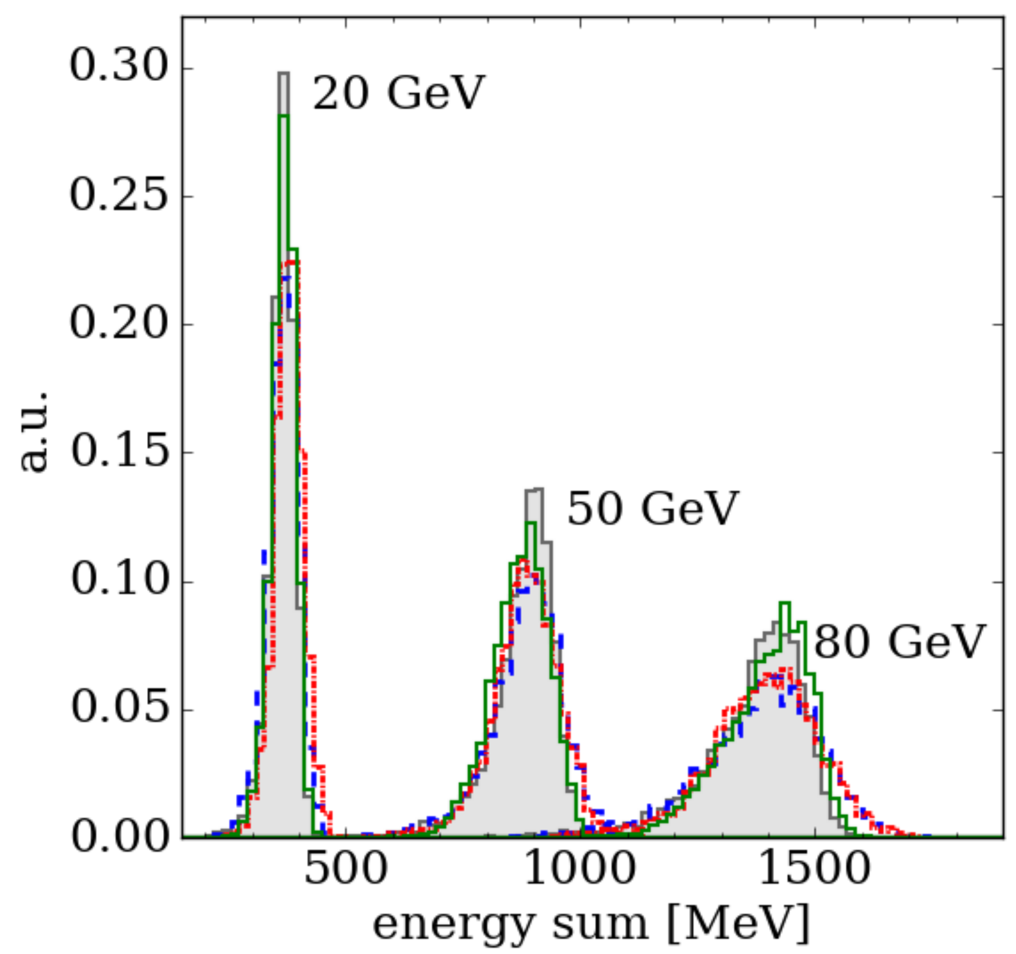
Differential Distributions

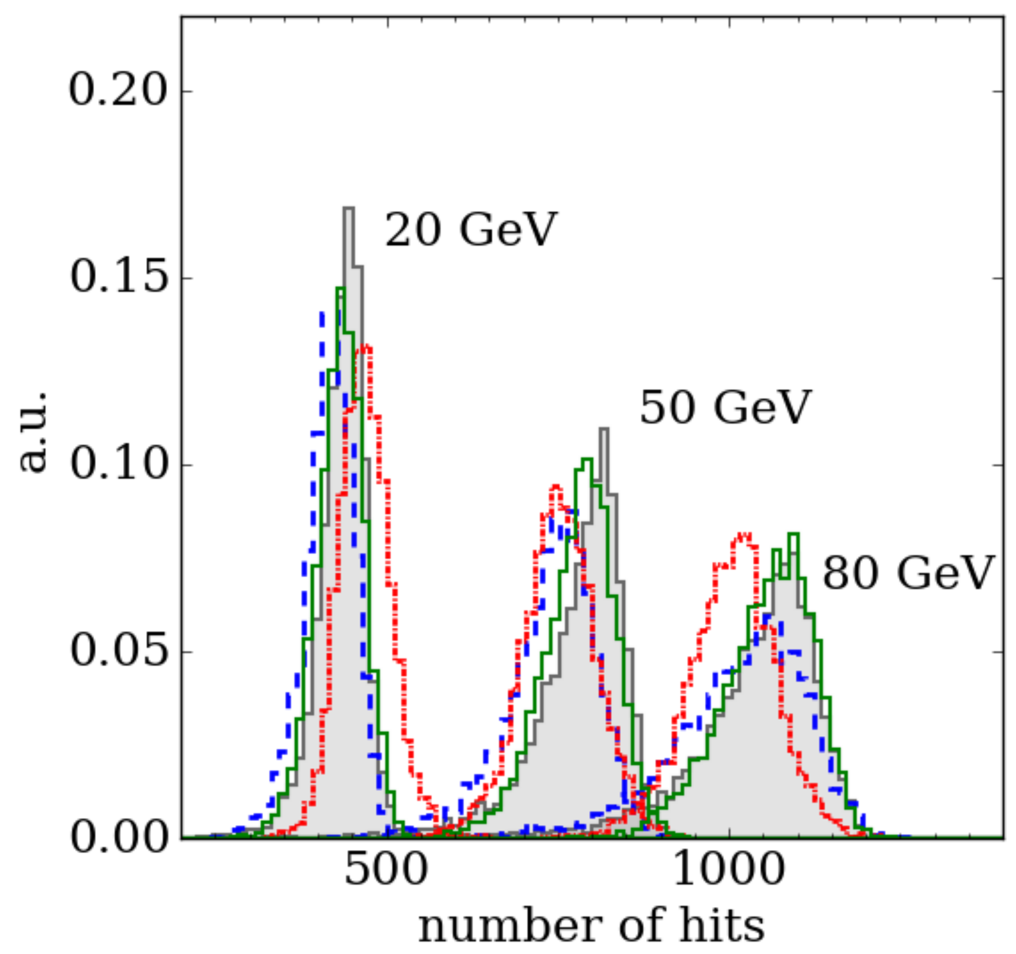


Buhmann et al.: **Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed:** 2005.05334

Differential Distributions

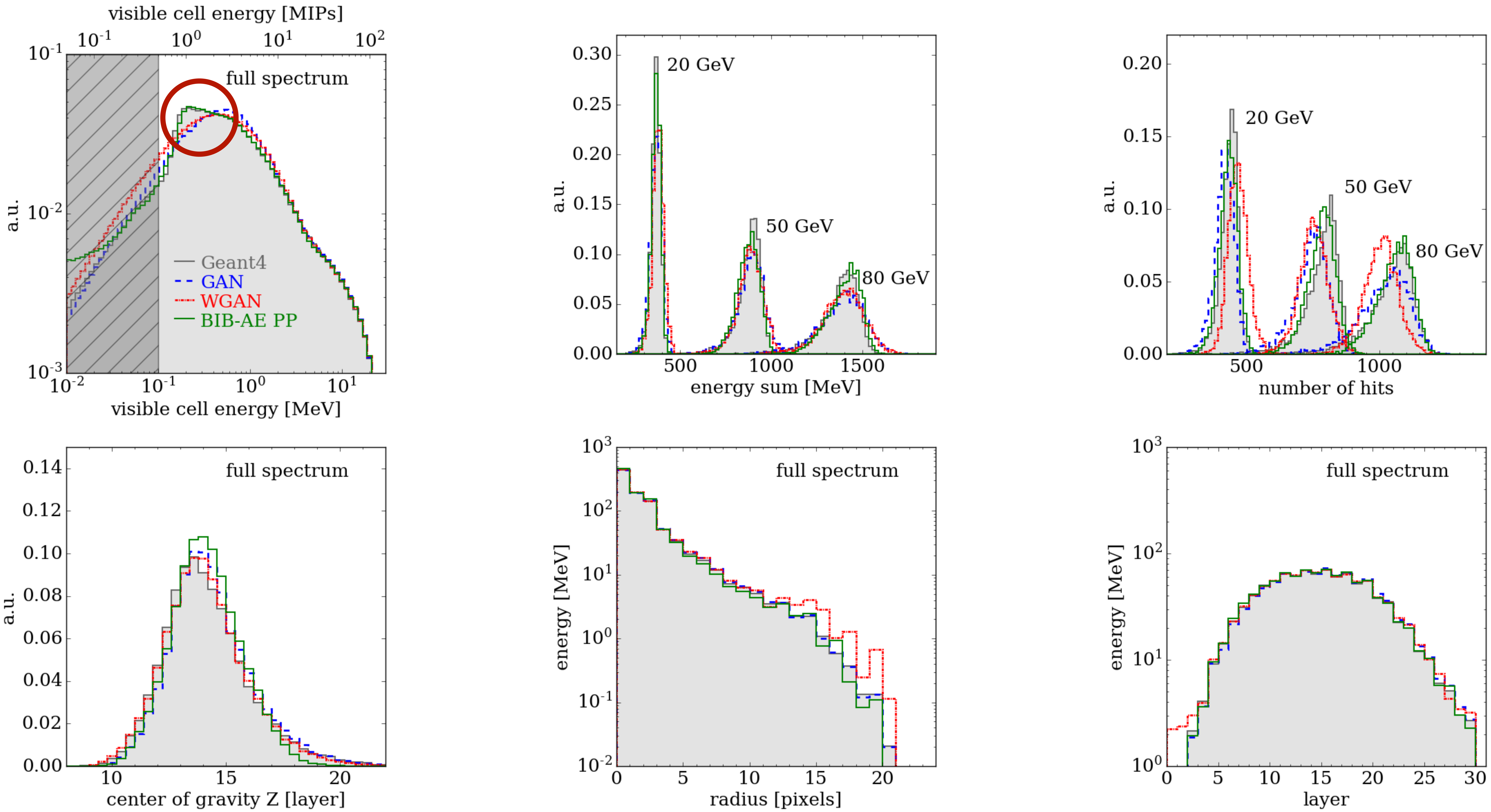






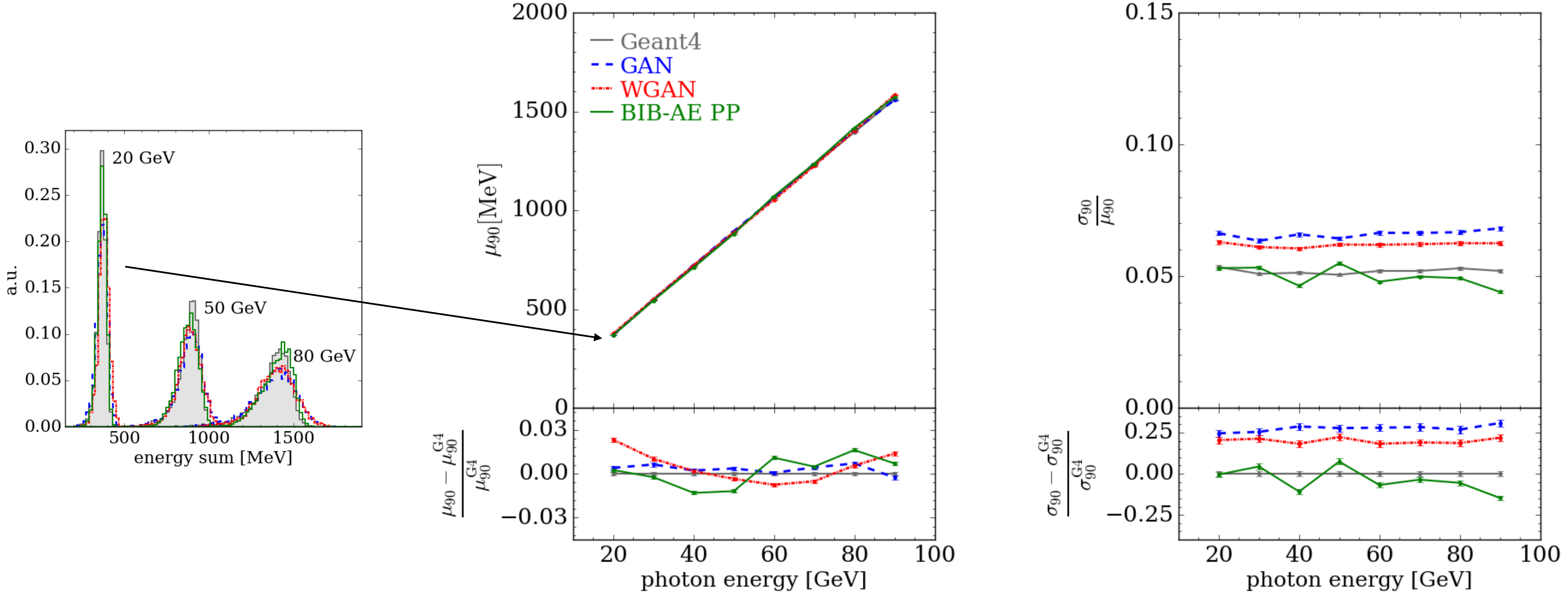
Buhmann et al.: **Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed:** 2005.05334

Differential Distributions



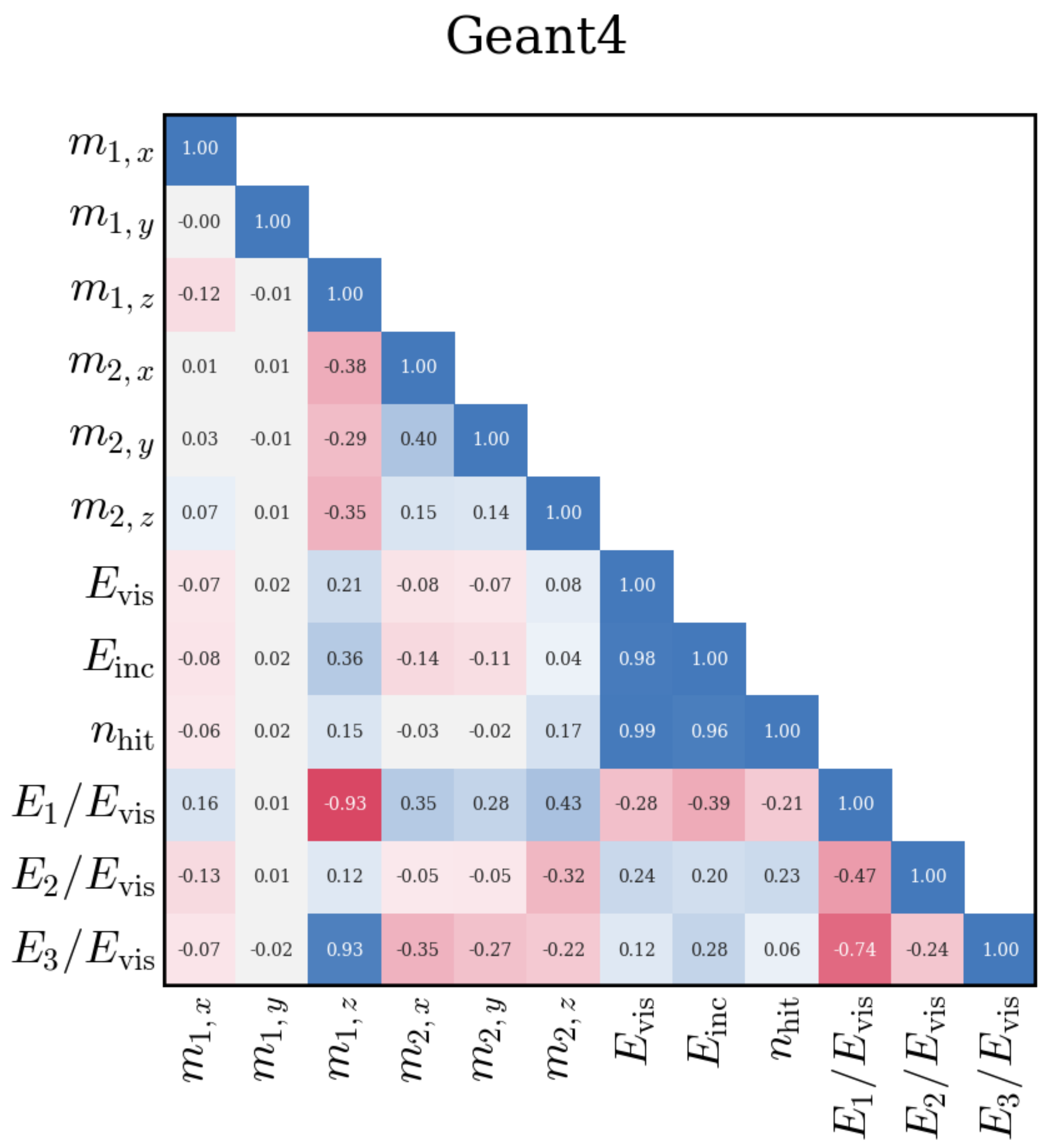
Buhmann et al.: **Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed:** 2005.05334

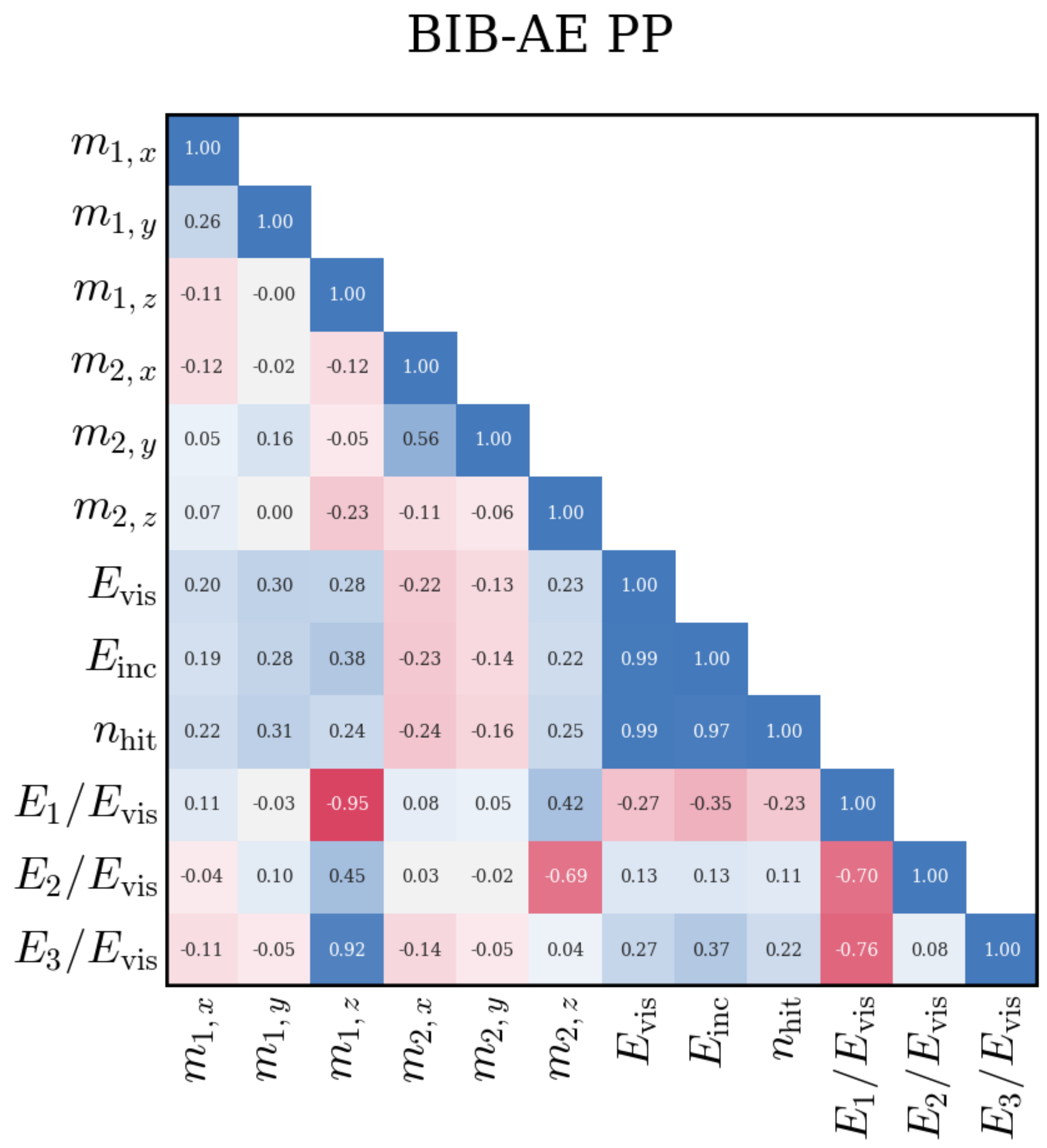
Linearity and Resolution*



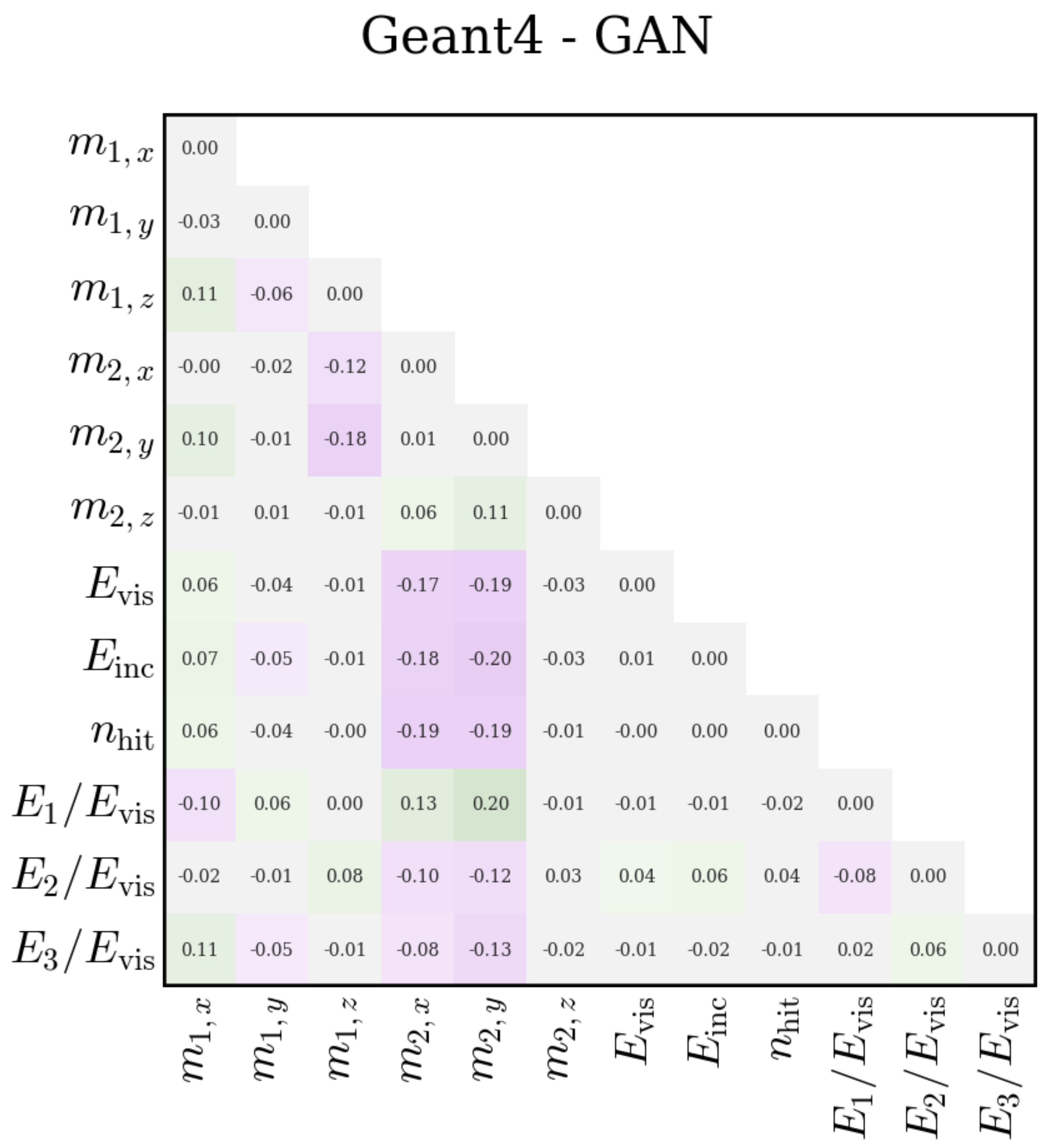
* actually not the ECAL resolution as not correction for sampling fraction variation performed

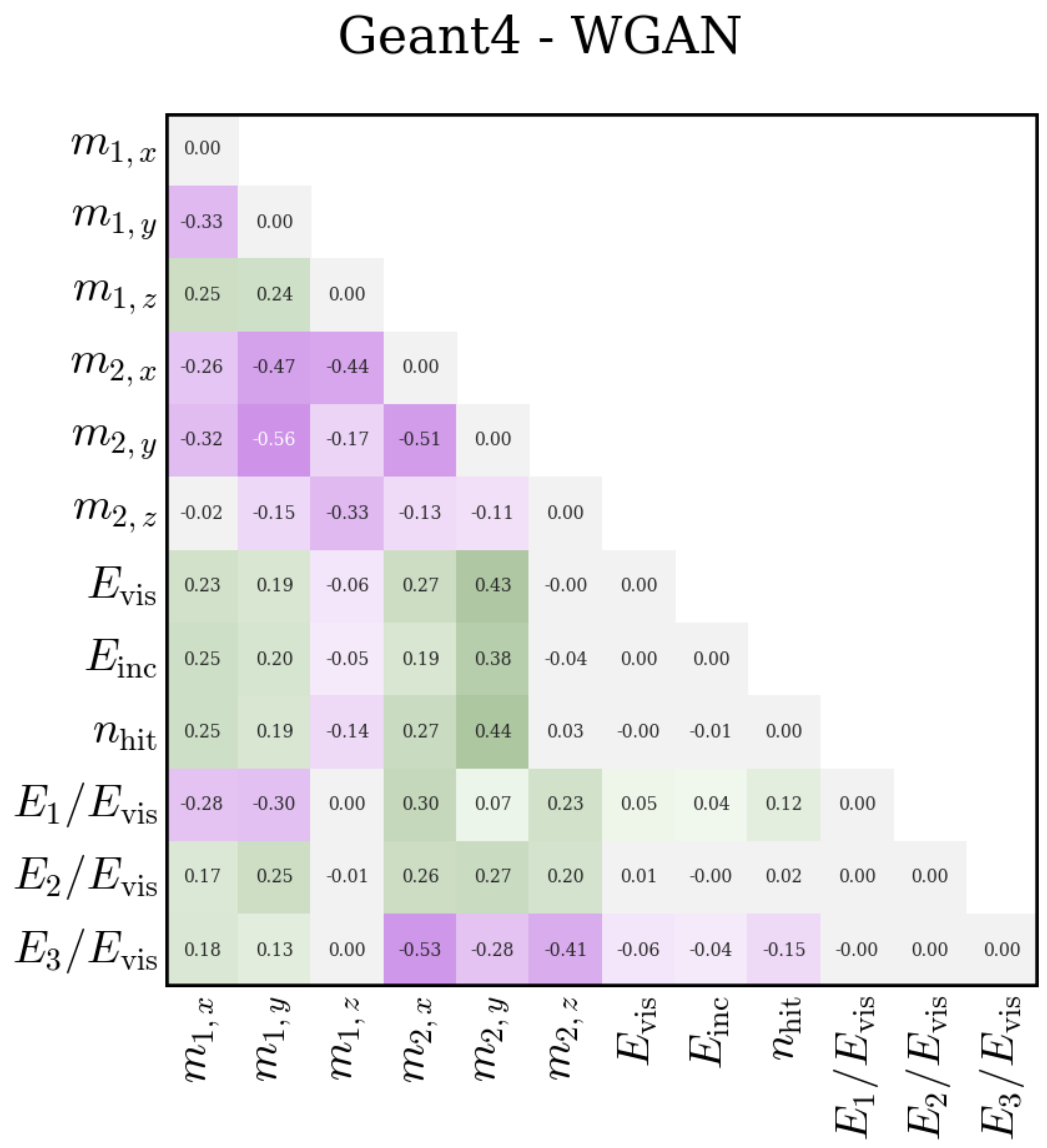
Correlations

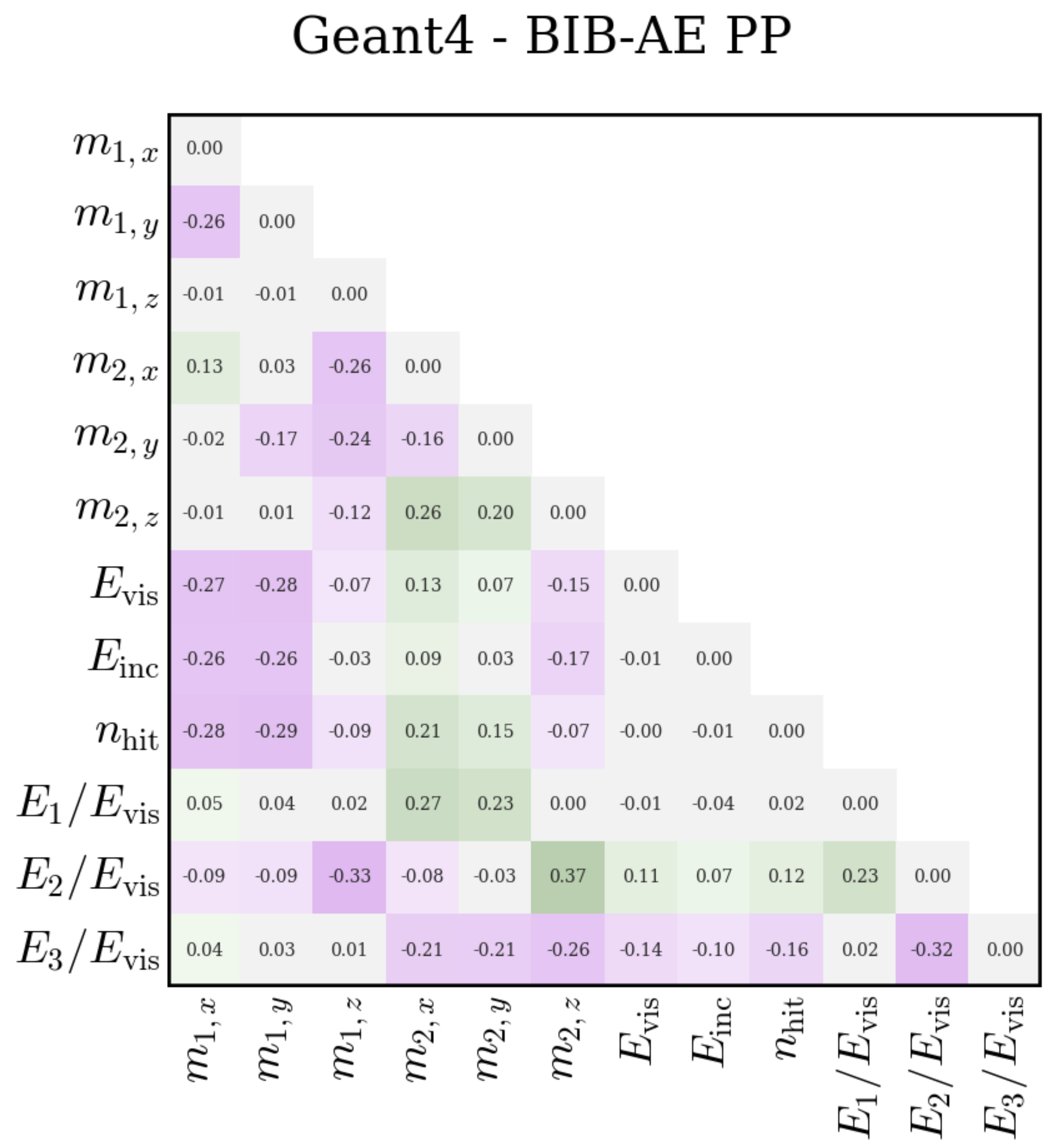




Correlations







Computation Time

- For 10-100 GeV showers:
    - 3 Orders of magnitude speedup compared to GEANT4

Simulator	Hardware	Batchsize	Time/shower	Speedup
GEANT4	CPU	N/A	4082 ± 170 ms	-
BIB-AE	CPU	1	426.3 ± 3.6 ms	x10
BIB-AE	GPU V100	1	3.19 ± 0.01 ms	x1279
BIB-AE	GPU V100	100	1.42 ± 0.01 ms	x2874

Thank you