E9: 309 ADL 02-12-2020

http://leap.ee.iisc.ac.in/sriram/teaching/ADL2020/

Housekeeping

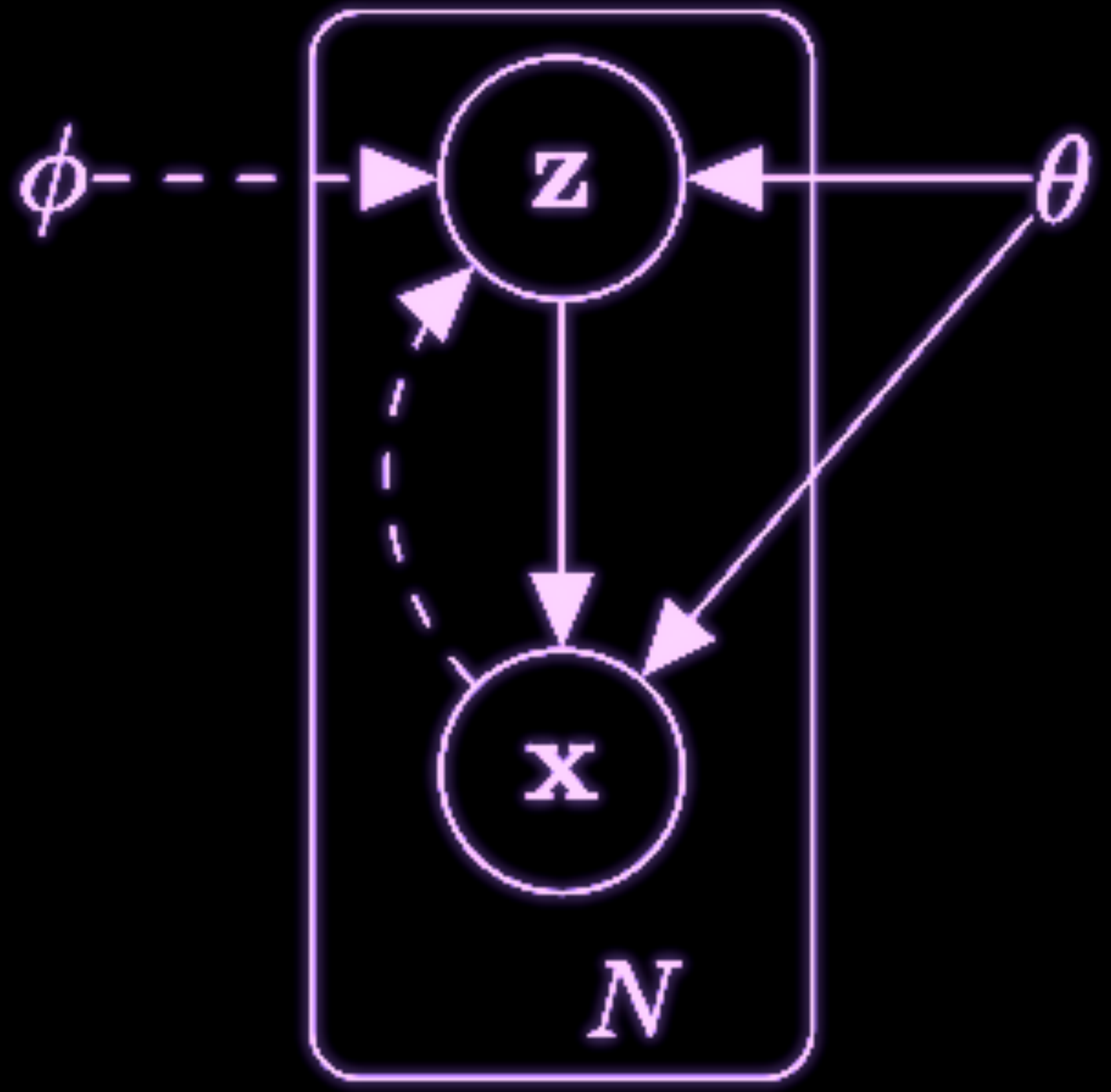
- Mid-term exam
  - December 6th (Sunday) [Topics covered up to Dec 2nd]
  - Mode of exam
    - Time to respond - 3 hours
      - Exam paper uploaded in Teams Channel and response (photo-scanned and uploaded in your folder).
      - Open book, open notes
        - Strictly no online communication or help sought.
          - Academic integrity and ethics strongly followed.

Recap of previous class

Variational auto encoders

- The data $\mathbf{x}$ and latent variable $\mathbf{z}$
- The forward model
  - ✓ Sample the latent variable
  - ✓ Sample the data given the latent $p_\theta(\mathbf{x}|\mathbf{z})$
- The marginal distribution $\int p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z}) d\mathbf{z}$
  - ✓ maybe intractable
- The posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$

➡ may also be intractable



Variational lower bound

- Defining the variational lower bound

$$L(\mathbf{x};\phi,\theta) = \int q_\phi(\mathbf{z}|\mathbf{x}) log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} dz = \mathbb{E}_q\left(log\left[\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]\right)$$

  - ✓ Called ELBO (evidence lower bound of the data)

$$L(\mathbf{x};\phi,\theta) = log\ p_\theta(\mathbf{x}) - KL(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))$$

    - Maximizing the lower bound
      - ★ Maximizes the log likelihood of the data
      - ★ Minimized the KL divergence between the true posterior distribution and appromixation.

Reparameterization

* Take a random variable independent of the data and parameters

$$\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$$

* Transform the random variable to the latent variable with parameters

$$\mathbf{z} = g(\boldsymbol{\epsilon}, \phi, \mathbf{x})$$

* Now the gradient computation of the expectation can be simplified.

  ✓ Decoupling the sampling and the gradient computations on the term involving $\mathbb{E}_q\left(log\ p_\theta(\mathbf{x}|\mathbf{z})\right)$

Assumptions and Approximations

$$L(\mathbf{x};\phi,\theta) = KL(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(z)) + \mathbb{E}_q\left(log\ p_\theta(\mathbf{x}|\mathbf{z})\right)$$

* Assume

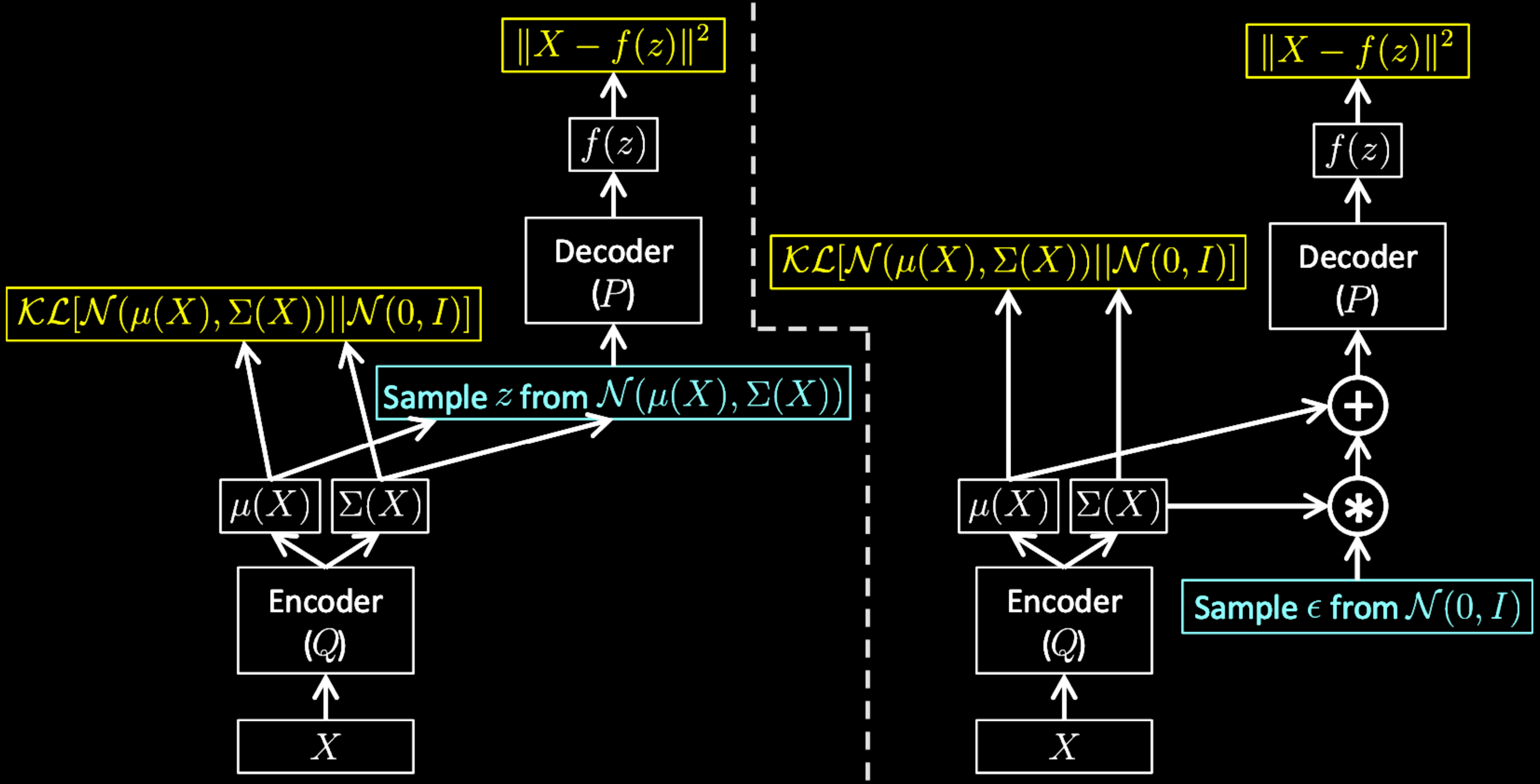
$$p_\theta(z) = \mathcal{N}(\mathbf{z};\mathbf{0},\mathbf{I})$$

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z};\boldsymbol{\mu}(\mathbf{x}),\boldsymbol{\sigma}(\mathbf{x})^2\mathbf{I})$$

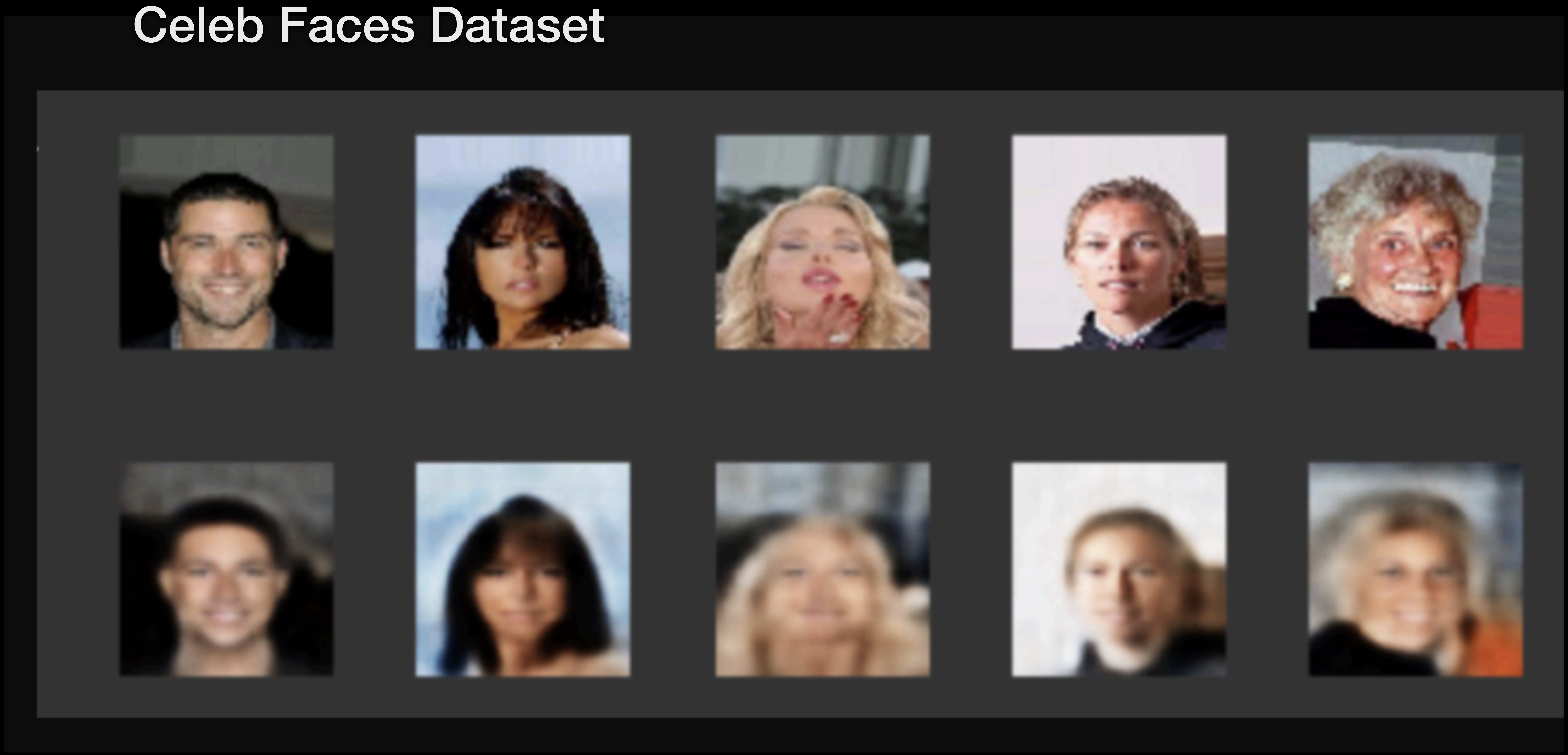
➡ The conditional distribution

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x};\boldsymbol{f}(\mathbf{z}),\mathbf{I})$$

The algorithm



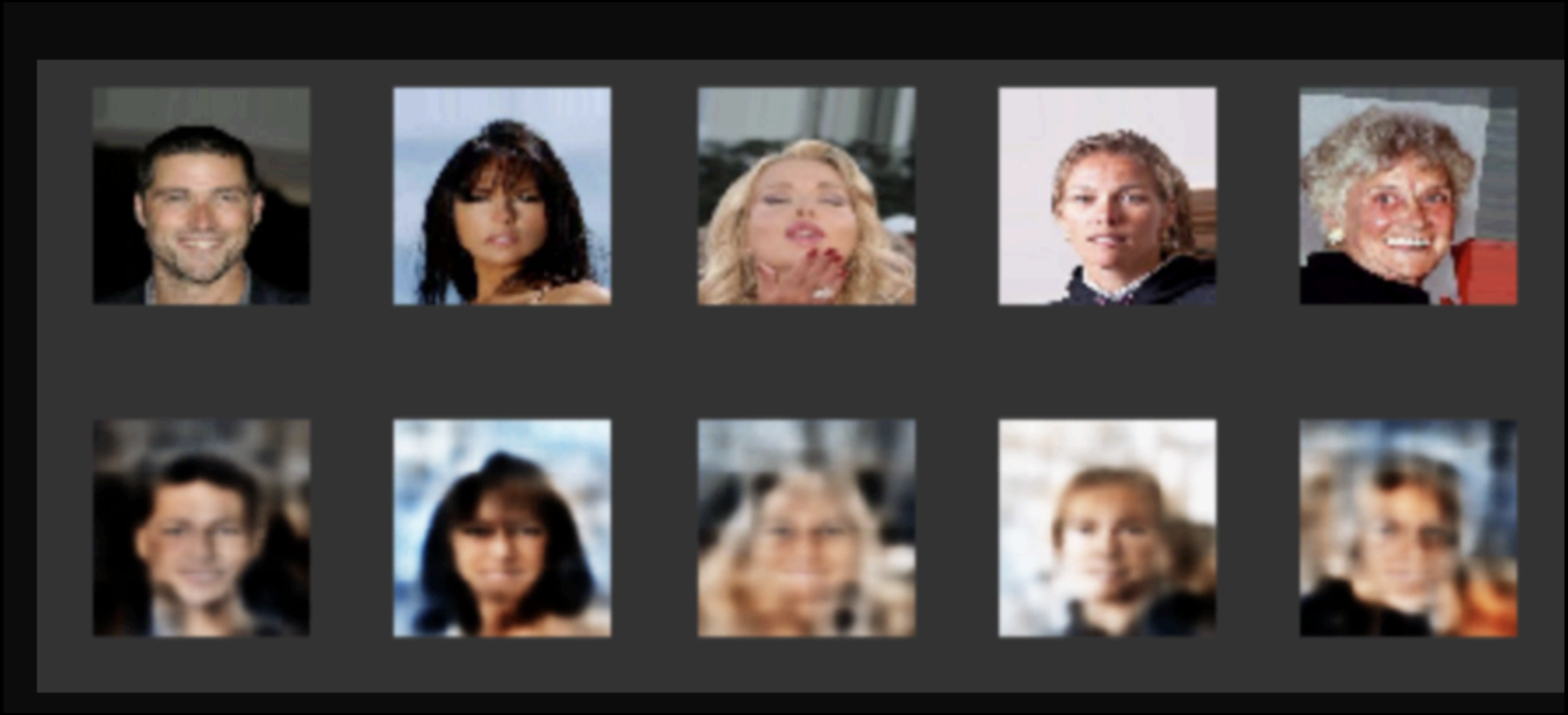
Standard AE - reconstruction



https://towardsdatascience.com/generating-new-faces-with-variational-autoencoders-d13cfcb5f0a8

Standard AE with added noise - reconstruction

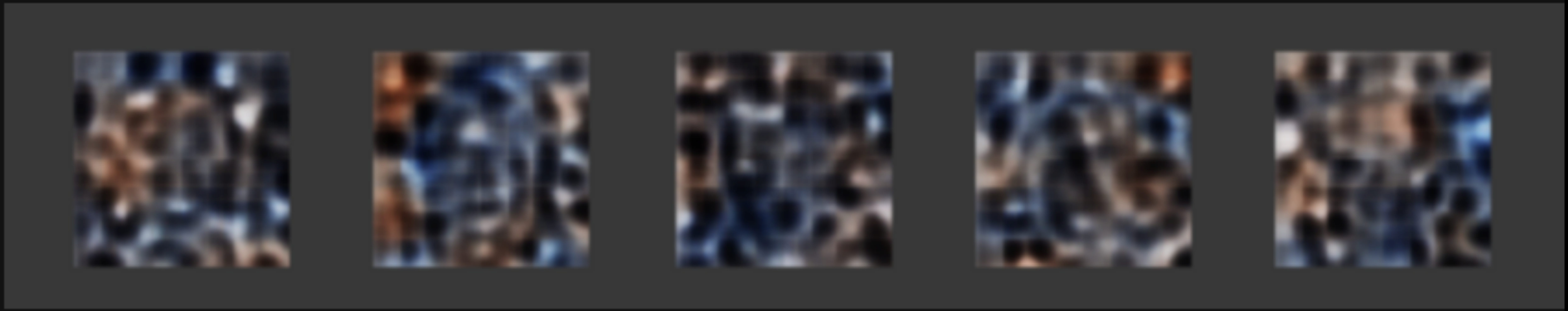
Celeb Faces Dataset



https://towardsdatascience.com/generating-new-faces-with-variational-autoencoders-d13cfcb5f0a8

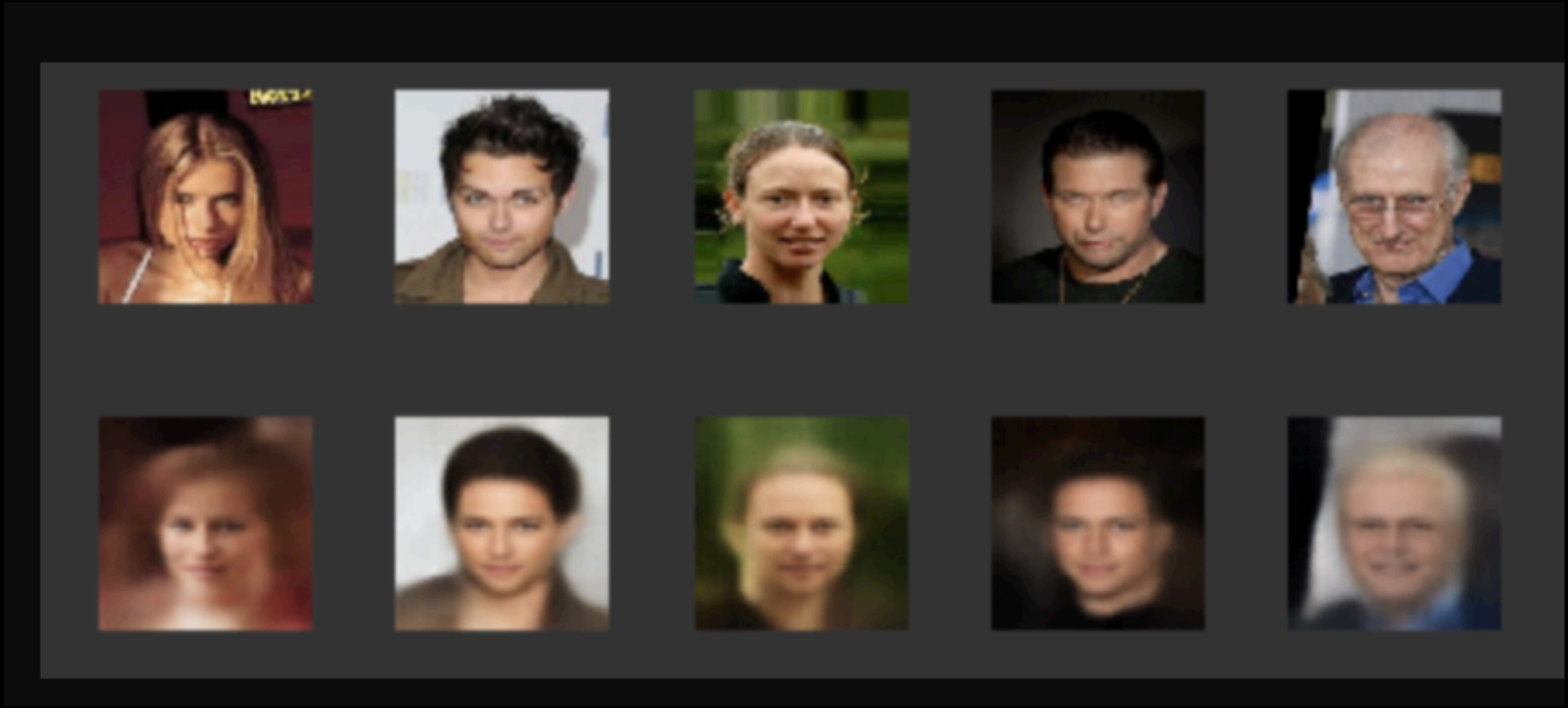
Standard AE only noise - reconstruction

Using noise as latent vector



https://towardsdatascience.com/generating-new-faces-with-variational-autoencoders-d13cfcb5f0a8

Variational Autoencoder - Using the sample

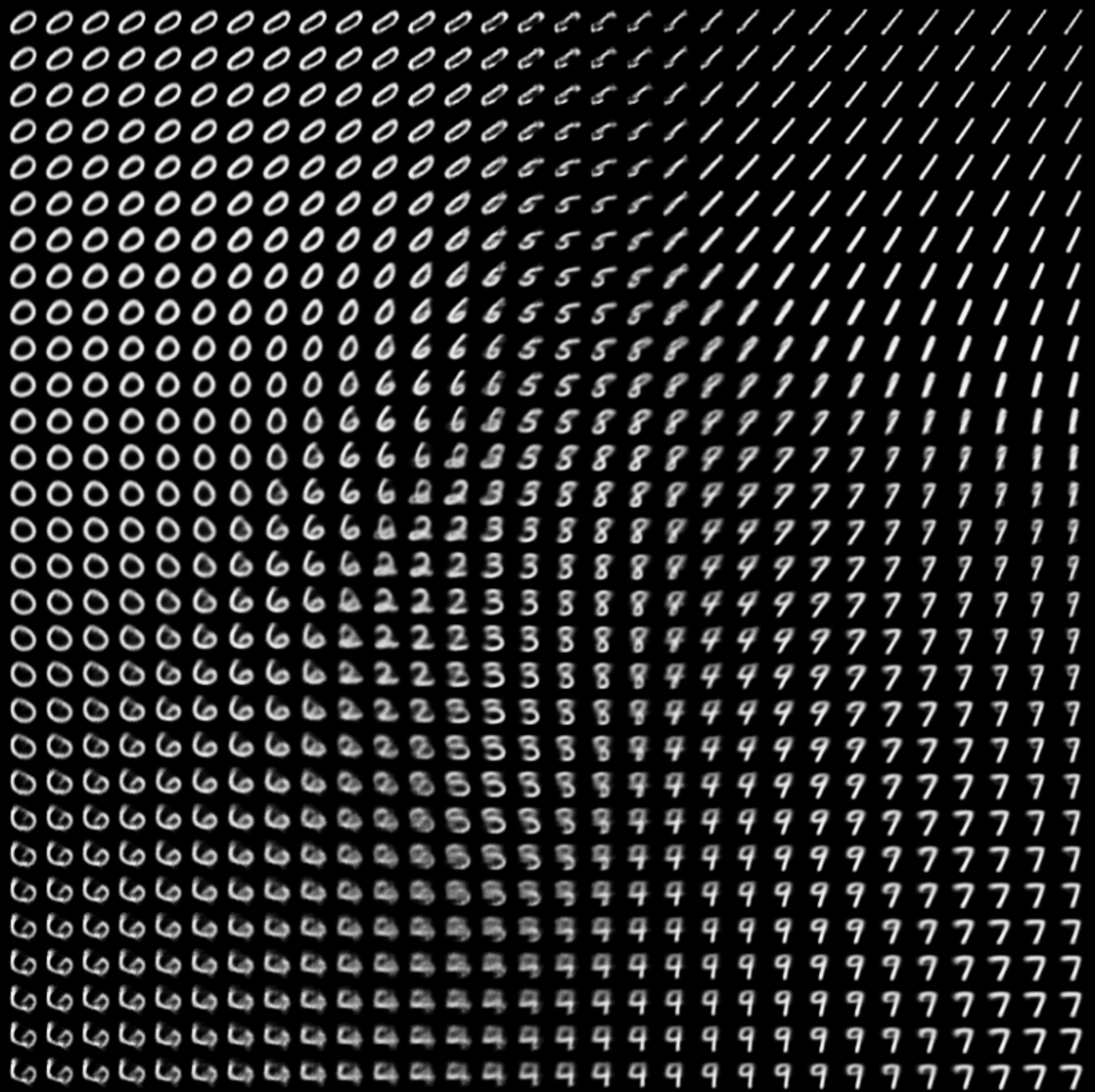


Variational Autoencoder - Using the random noise

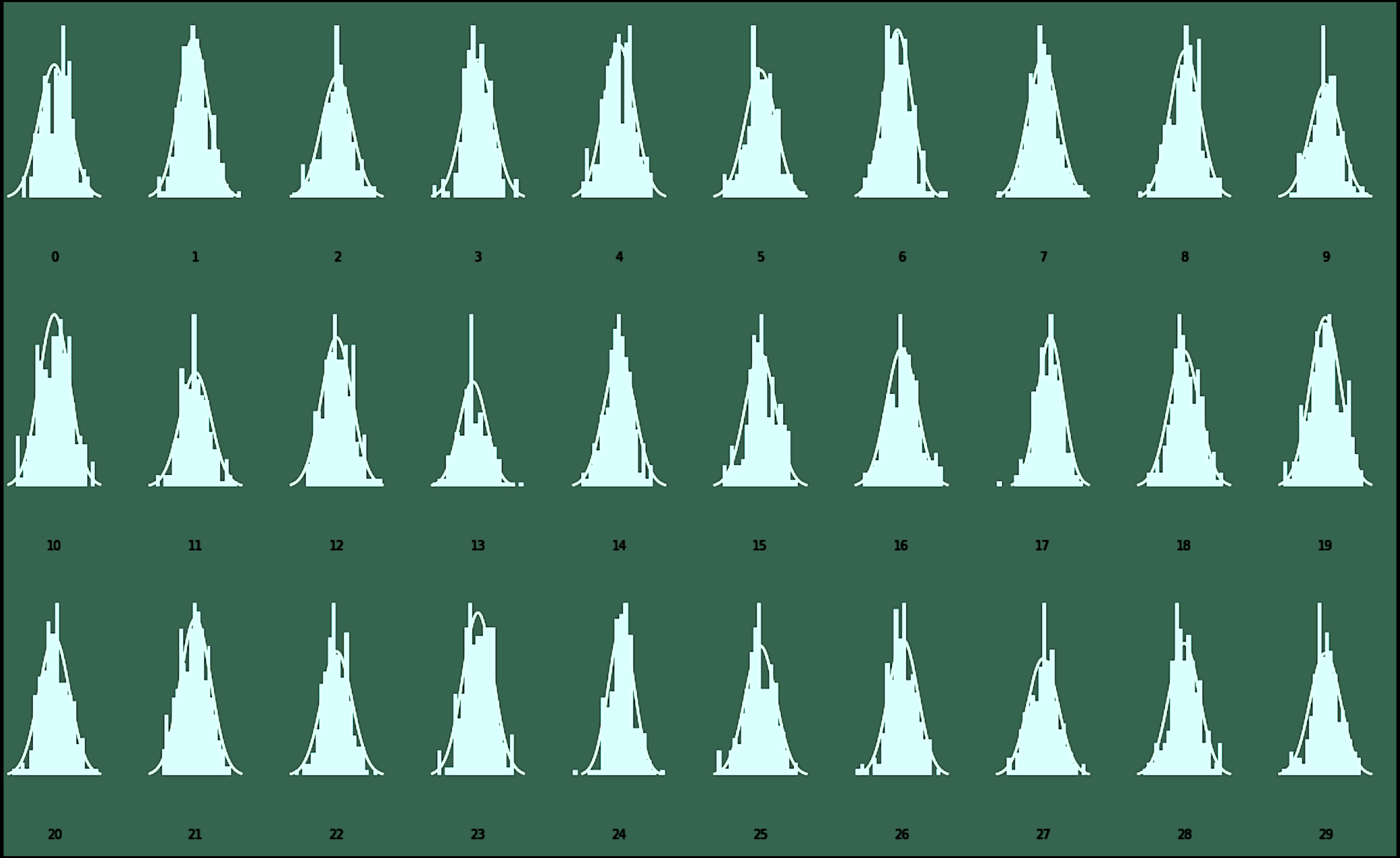
Using random noise generated from standard Gaussian - latent dim (200D)

VAE

- MNIST dataset
  - ➡ linearly sampling from 2-D latent space.
  - ➡ The latent space tends to cluster
    - ✓ Similar digits can be found by minor variations
    - ✓ There can be non-digits also in the output



VAE - latent distribution



Variational auto encoder

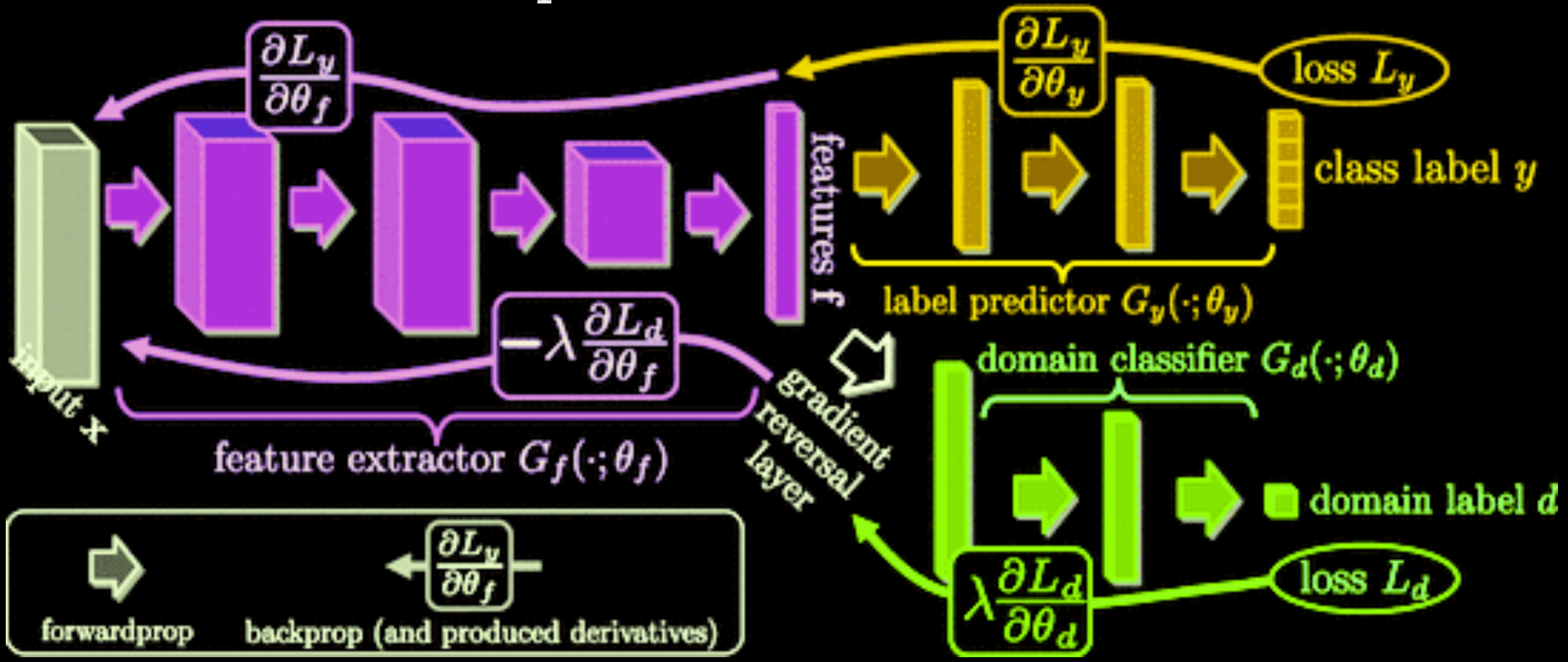
Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." *arXiv preprint arXiv:1312.6114* (2013).

- more Reading
  - Kingma, Diederik P., and Max Welling. "An introduction to variational autoencoders." arXiv preprint arXiv:1906.02691 (2019).
- More reading
  - Doersch, Carl. "Tutorial on variational autoencoders." arXiv preprint arXiv:1606.05908 (2016).

Adversarial loss

- ✴ Increasing the loss on certain nuisance directions
  - ➡ To increase the invariance of the representations

Domain Adversarial Loss

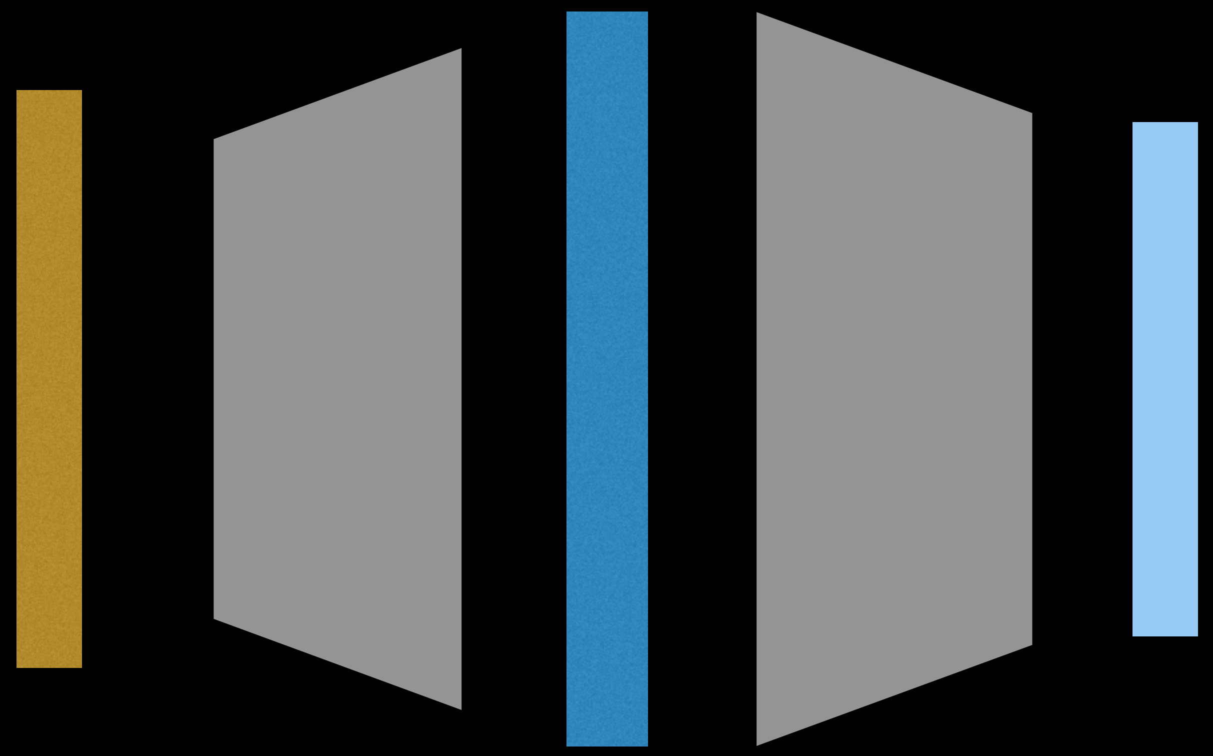


Generative Adversarial Networks

- Generator networks
  - ➡ Differentiable network
  - ➡ Draws a random noise sample
    - ✓ Convert the noise to the data distribution

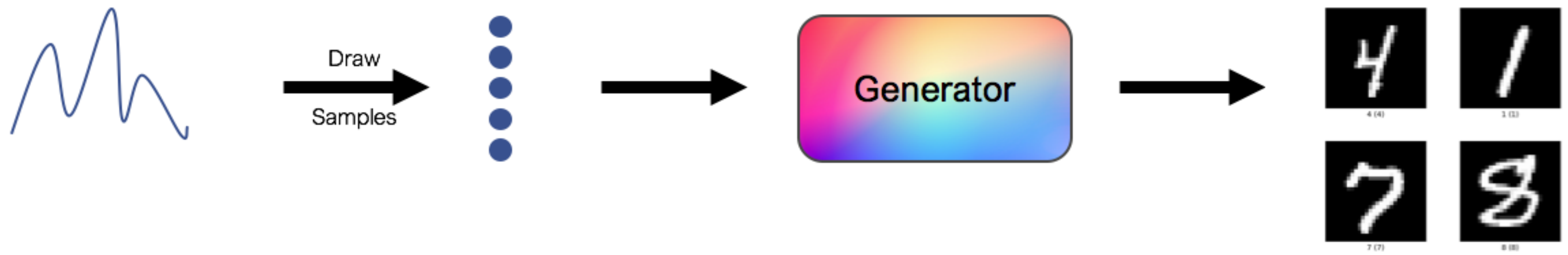
$$\mathbf{x}' \sim p_{\theta^G}(\mathbf{x}|\mathbf{z})$$

$$\mathbf{z} \sim p(\mathbf{z})$$



$$\mathbf{x}' = G(\mathbf{z}, \boldsymbol{\theta}^G)$$

GAN - generator



We start with a random distribution

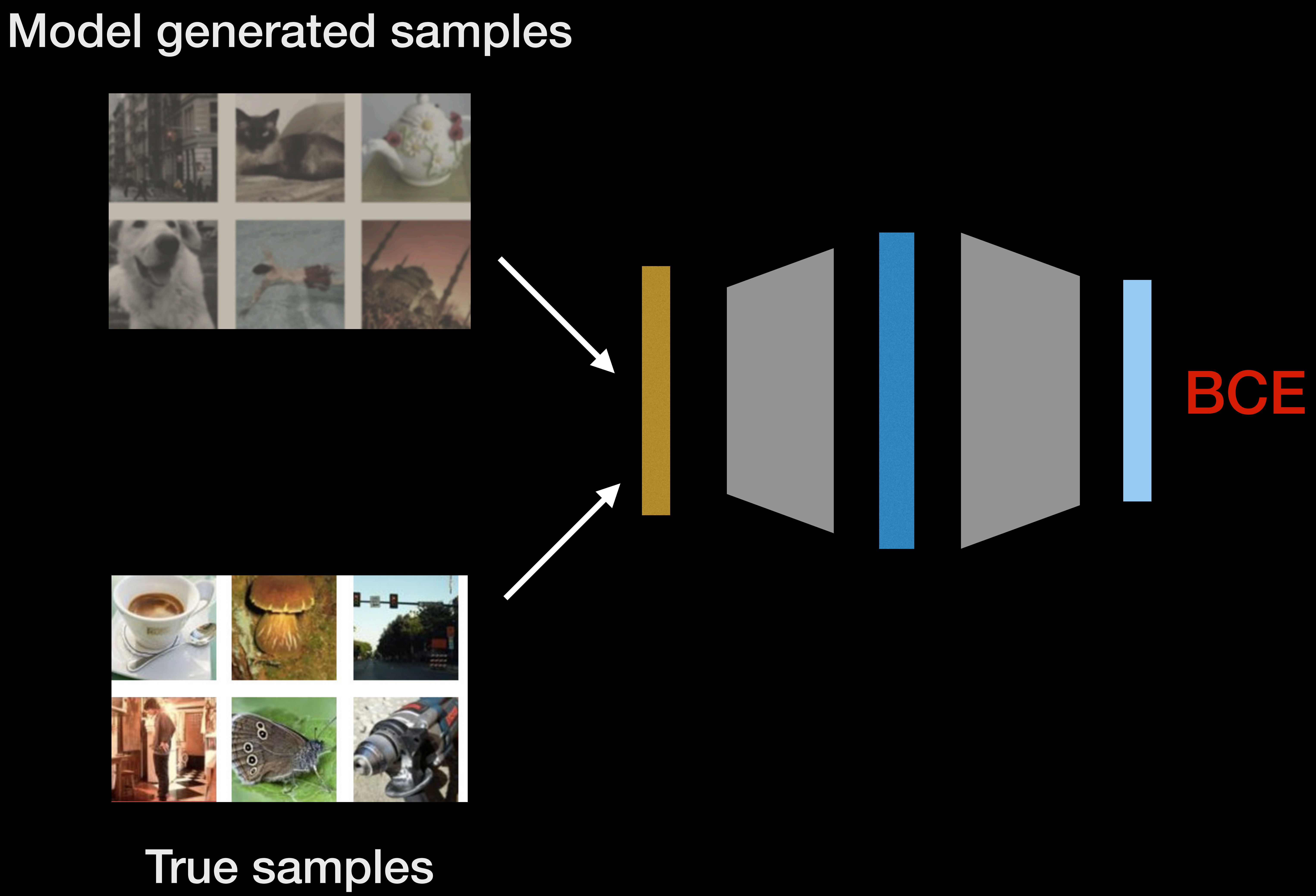
Samples drawn from the random distribution are sent to the generator as input

The generator transforms samples drawn from the random distribution to samples that resembles those that are drawn from the actual distribution

Transformed samples as the generator's output

Generative Adversarial Networks

- Discriminator network
  - Two class classifier
    - ✓ Label 0 for $\mathbf{x}'$
    - ✓ Label 1 for $\mathbf{x}$
  - Model learning



GAN - discriminator

The discriminator takes images as input, and predicts the most likely class (real or fake) of the input image

Min-max game

- The binary cross entropy loss at the discriminator network is

$$E_D = -\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[log\ D(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[log\ D(G(\mathbf{z}))]$$

$$E_G = -E_D$$

- The discriminator is trained using B.C.E loss

$$\theta_D{}^n = \theta_D{}^{n-1} - \eta \frac{\partial E_D}{\partial \theta_D}$$

Min-max game

- The binary cross entropy loss at the discriminator network is

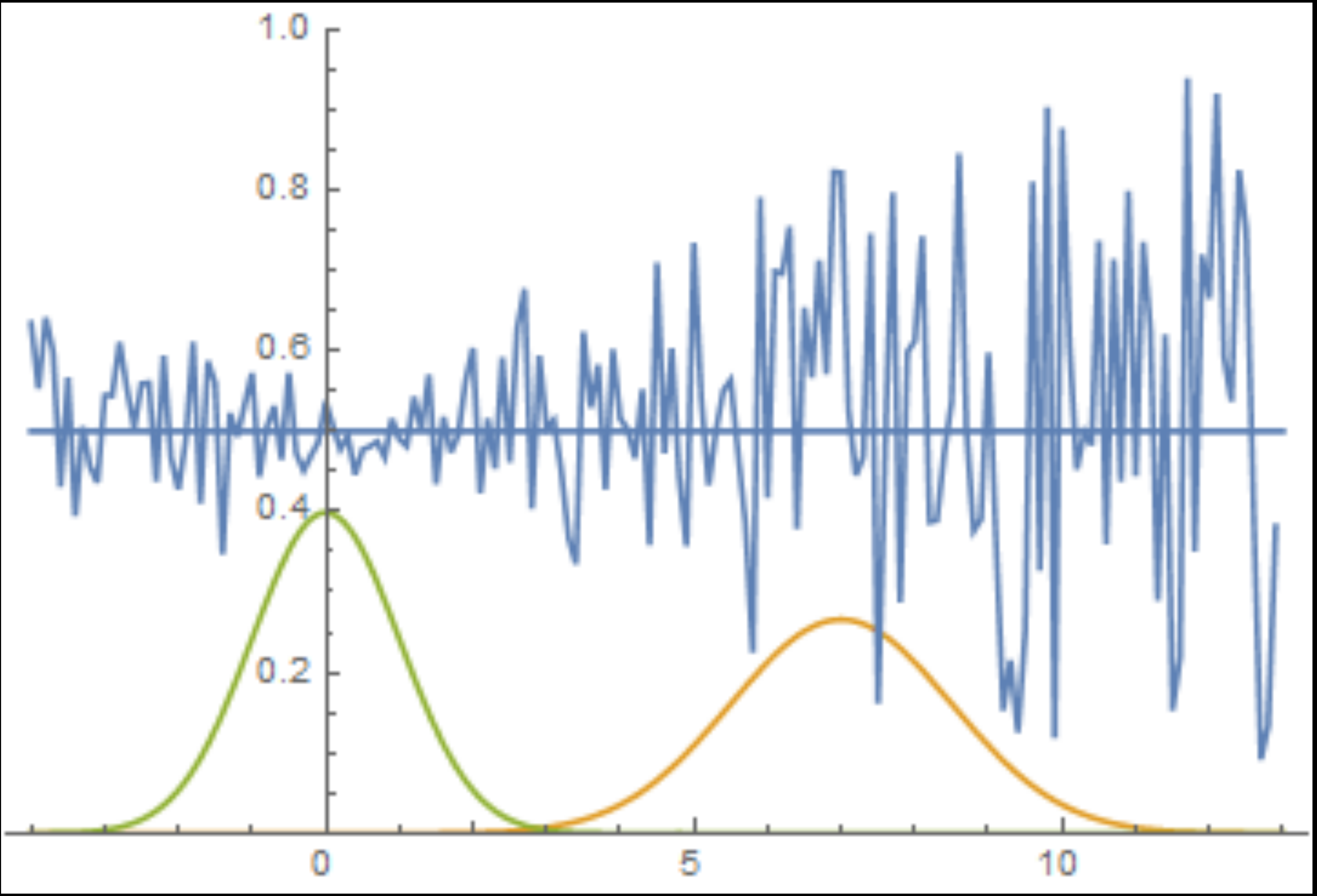
$$E_D = -\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[log\ D(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[log\ D(G(\mathbf{z}))]$$

$$E_G = -E_D$$

- The discriminator is trained using B.C.E loss

$$\theta_G{}^n = \theta_G{}^{n-1} - \eta \frac{\partial E_G}{\partial \theta_G}$$

GAN - Simple example



GAN - considerations

- Usually the discriminator "wins"
    - This is a good thing—the theoretical justifications are based on assuming D is perfect
- Usually D is bigger and deeper than G
    - The D starts off with much faster saturation.
- Learning of G happens even when D is saturated
- Equilibrium - Discriminator is defeated and outputs half for both real and model generated samples.