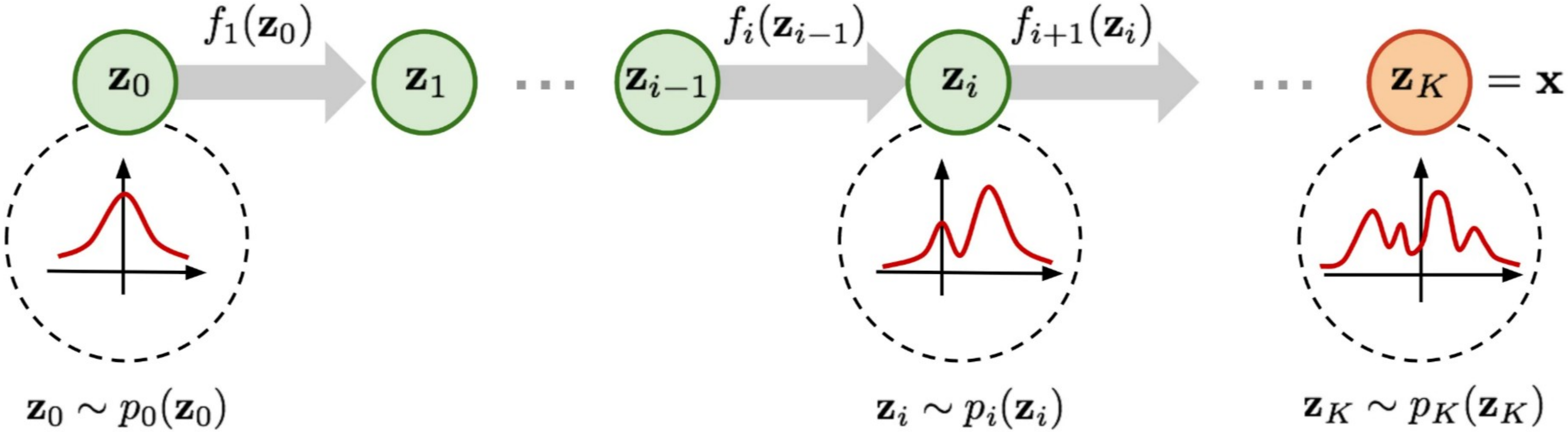
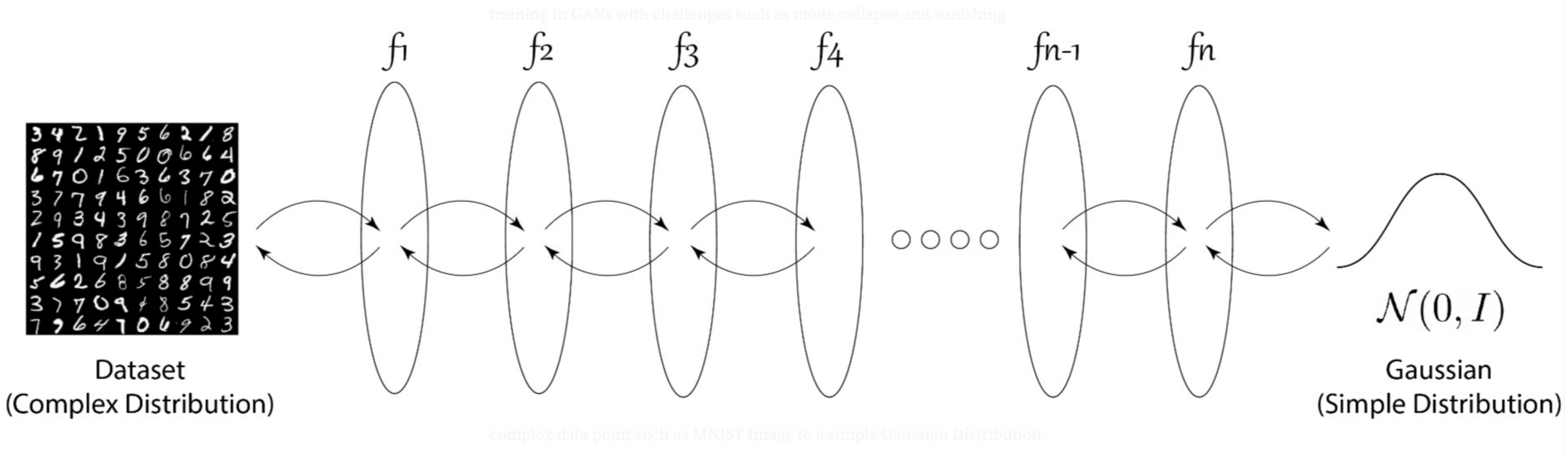
# Normalizing Flows

Mostafa Tavassolipour

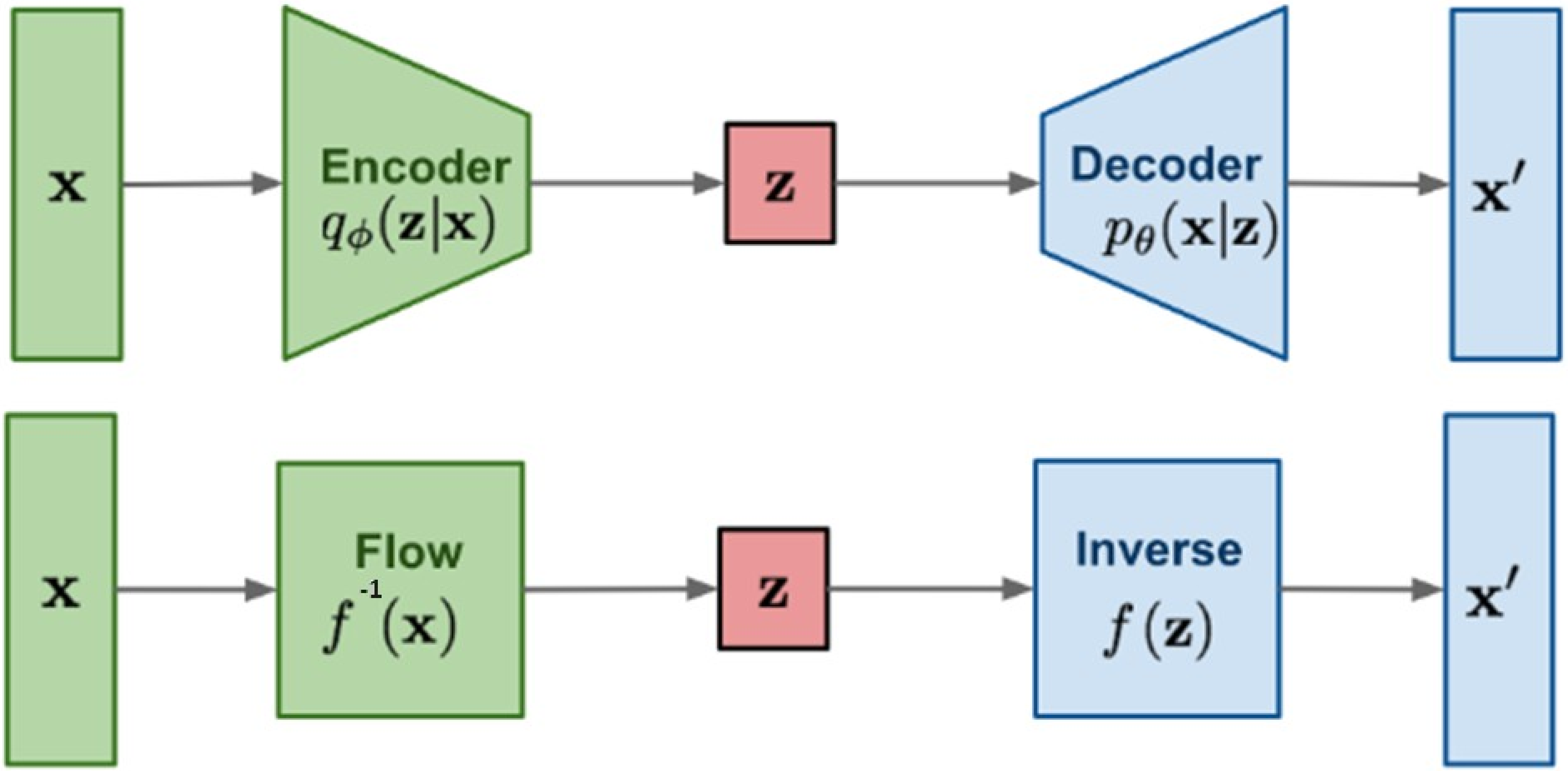
# Normalizing Flows



# Example: MNIST Dataset

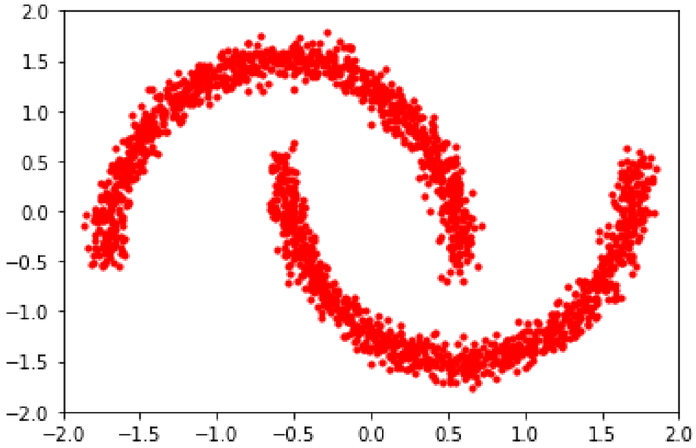


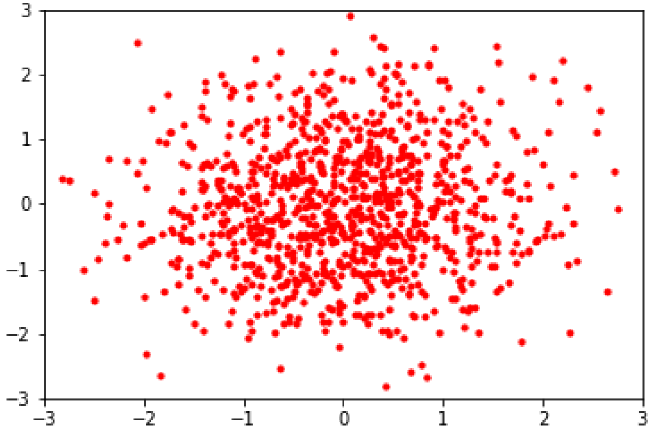
# Normalizing Flows vs. VAE

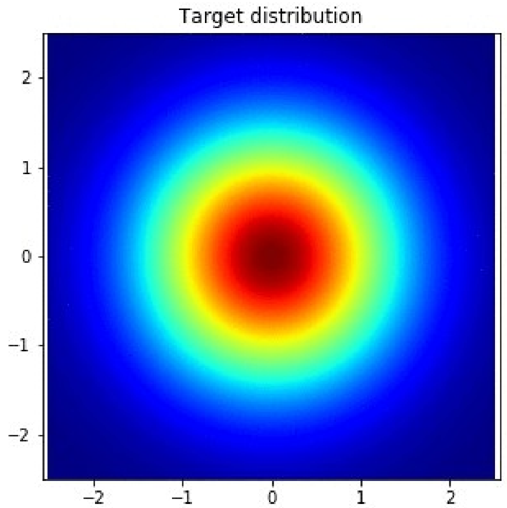


# Marginal Likelihood: VAE vs NF

# From Normal to Complex Distribution



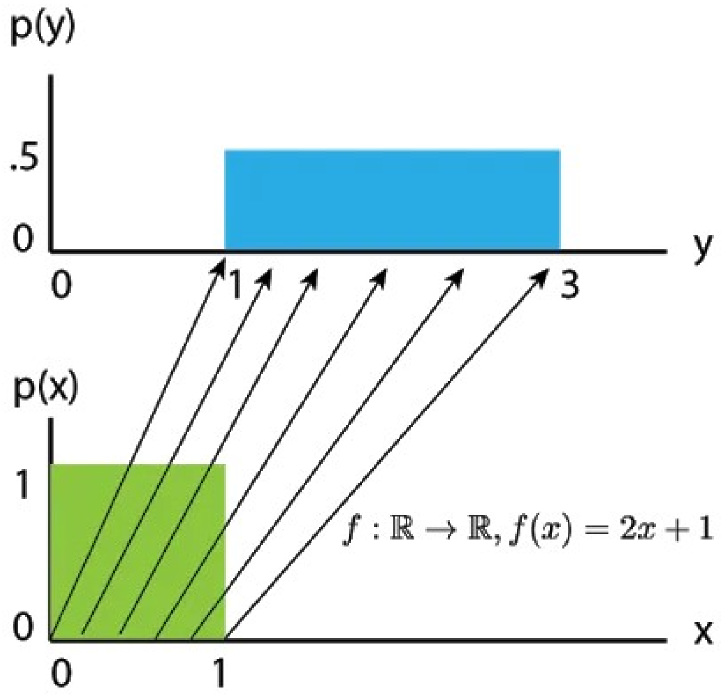



# Normalizing Flow: another Example

# Samples from Normalizing Flow

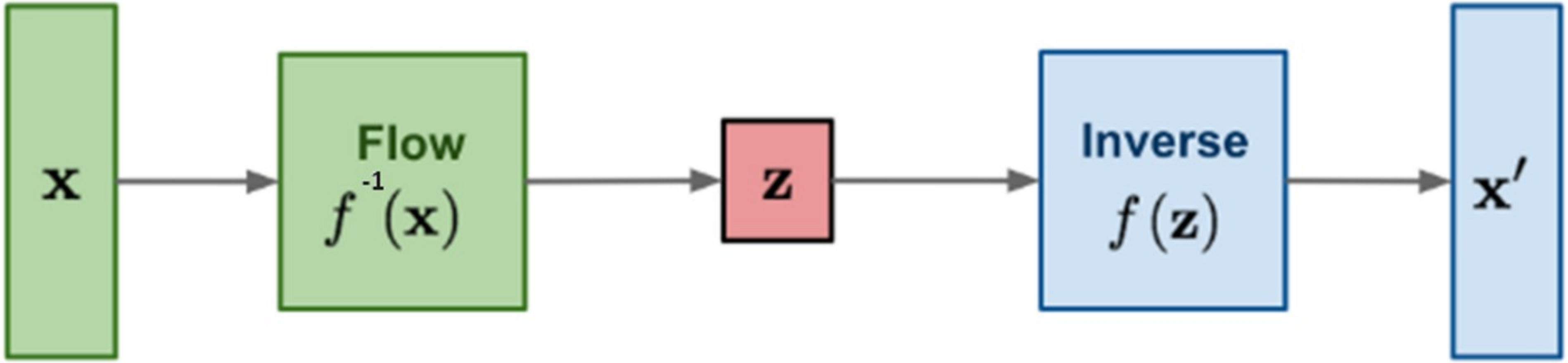
Samples from the Glow model (Source)

# Change of Variable Formula

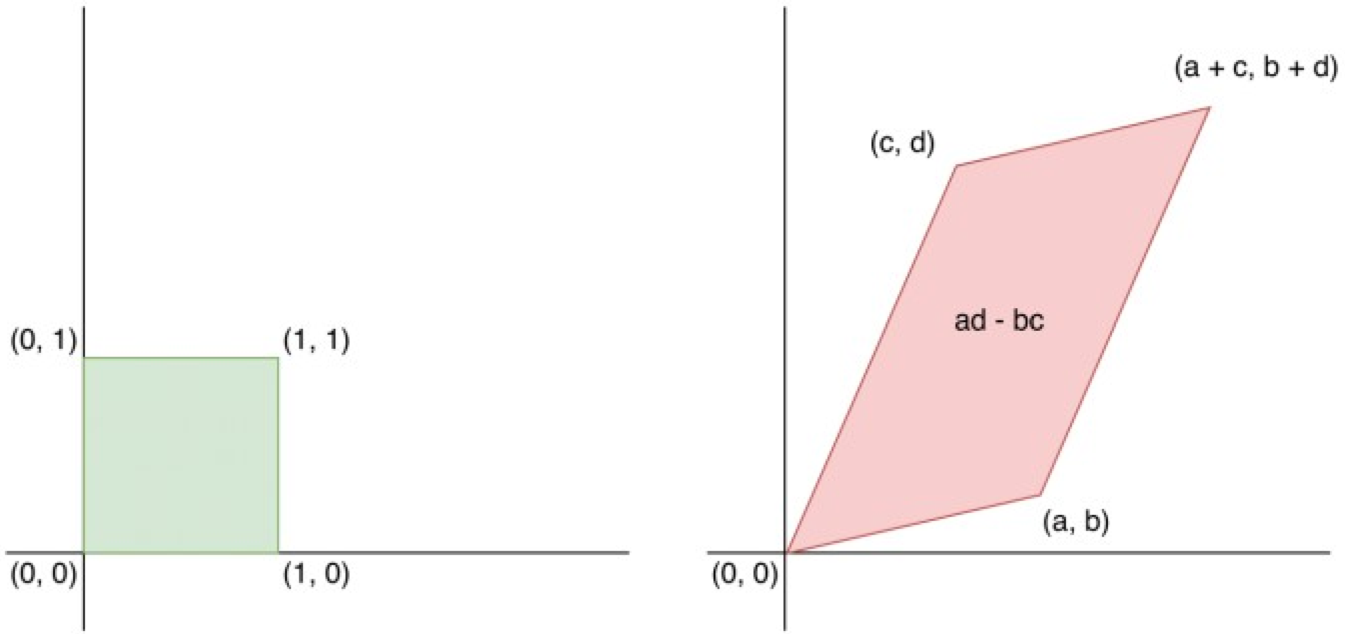



$$P_Y(y) = \frac{1}{2} P_X(x) = \frac{1}{2} P_X\left(\frac{y-1}{2}\right)$$

# Change of Variable Formula: 1-Dimensional

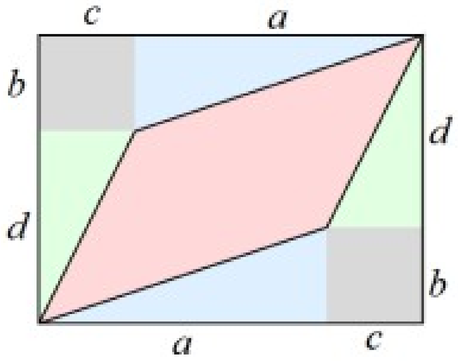


# Change of variable: n-Dimensional



# Determinants and Volume

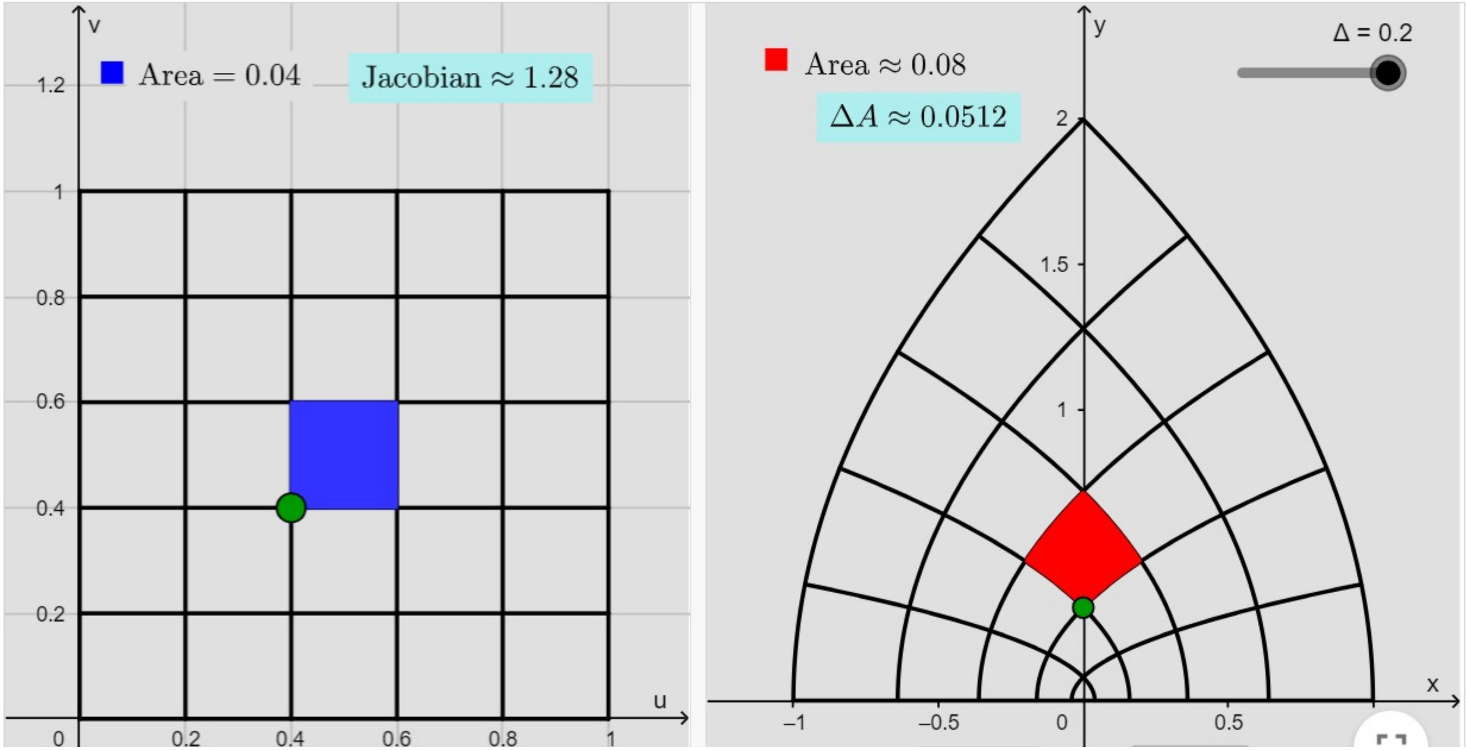
$$\det(A) = \det\begin{pmatrix} a & c \\ b & d \end{pmatrix} = ad - bc$$




$$(a+c)(b+d) - ab - 2bc - cd = ad - bc$$

# Jacobian Determinant

$$\Delta A = |J| \times Area\, of\, square$$



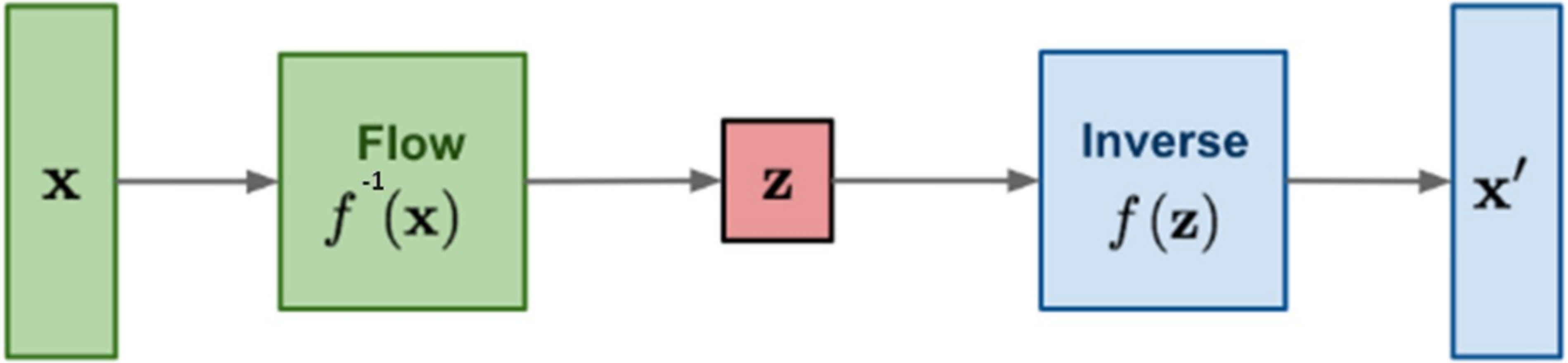

https://www.geogebra.org/m/qM777NYH

# Jacobian Matrix

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \underset{\rightarrow}{f} \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$\frac{\partial(y_1, \ldots, y_n)}{\partial(x_1, \ldots, x_n)} ¿ \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_n} \end{bmatrix}$$

# Change of Variable Formula: n-Dimensional



# Learning and Inference

- Learning via **Maximum Likelihood:**

$$\max_{\theta} \log p_X(\mathcal{D};\theta) = \sum_{x \in \mathcal{D}} \log p_Z\left(f_\theta^{-1}(x)\right) + \log \left| \det \left( \frac{\partial f_\theta^{-1}(x)}{\partial x} \right) \right|$$

- **Sampling:**

$$z \sim p_Z(z) \quad x = f_\theta(z)$$

- **Latent Representation:**

$$z = f_\theta^{-1}(x)$$

# Calculation of the Determinant

- Computing the determinant for an  matrix is : prohibitively expensive within a learning loop!

- **Key idea**: Choose transformations so that the resulting Jacobian matrix has **special structure**. For example, the determinant of a **triangular** matrix is the product of the diagonal entries, i.e., an  operation.

# Models with Normalizing Flows

- **NICE** (Non-linear Independent Component Estimation, 2015)
- **RealNVP** (Real-valued Non-Volume Preserving, 2017)

- Inverse Autoregressive Flow (Kingma et al., 2016)
- Masked Autoregressive Flow (Papamakarios et al., 2017)
- I-resnet (Behrmann et al, 2018)
- Glow (Kingma et al, 2018)
- MintNet (Song et al., 2019)
- And many more

# NICE

- Unit Jacobian determinant:

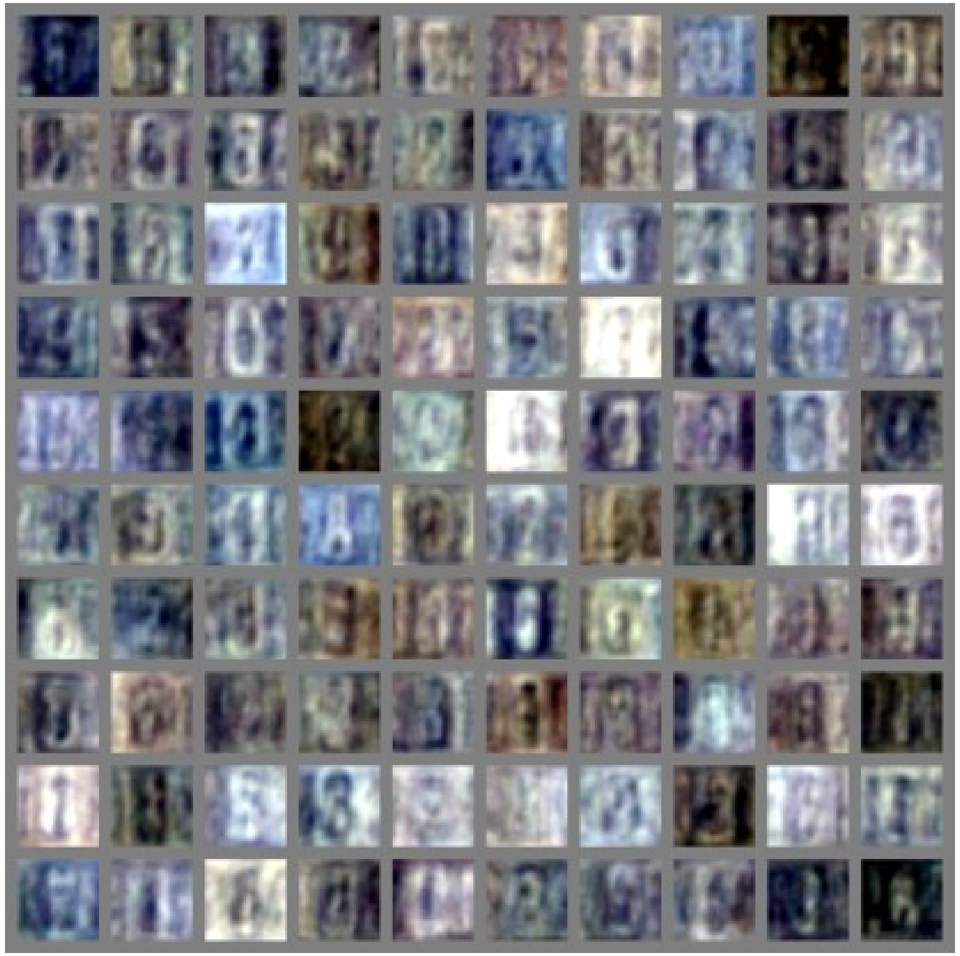
**Inverse:**

# NICE: Results

(a) Model trained on MNIST

(b) Model trained on TFD

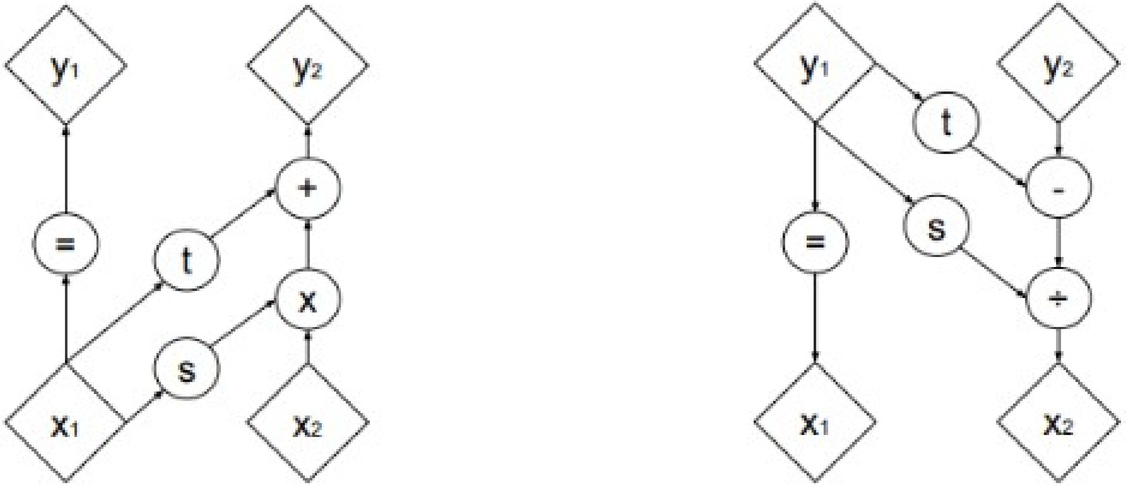
# NICE: Results on Complex Data



(c) Model trained on SVHN

(d) Model trained on CIFAR-10

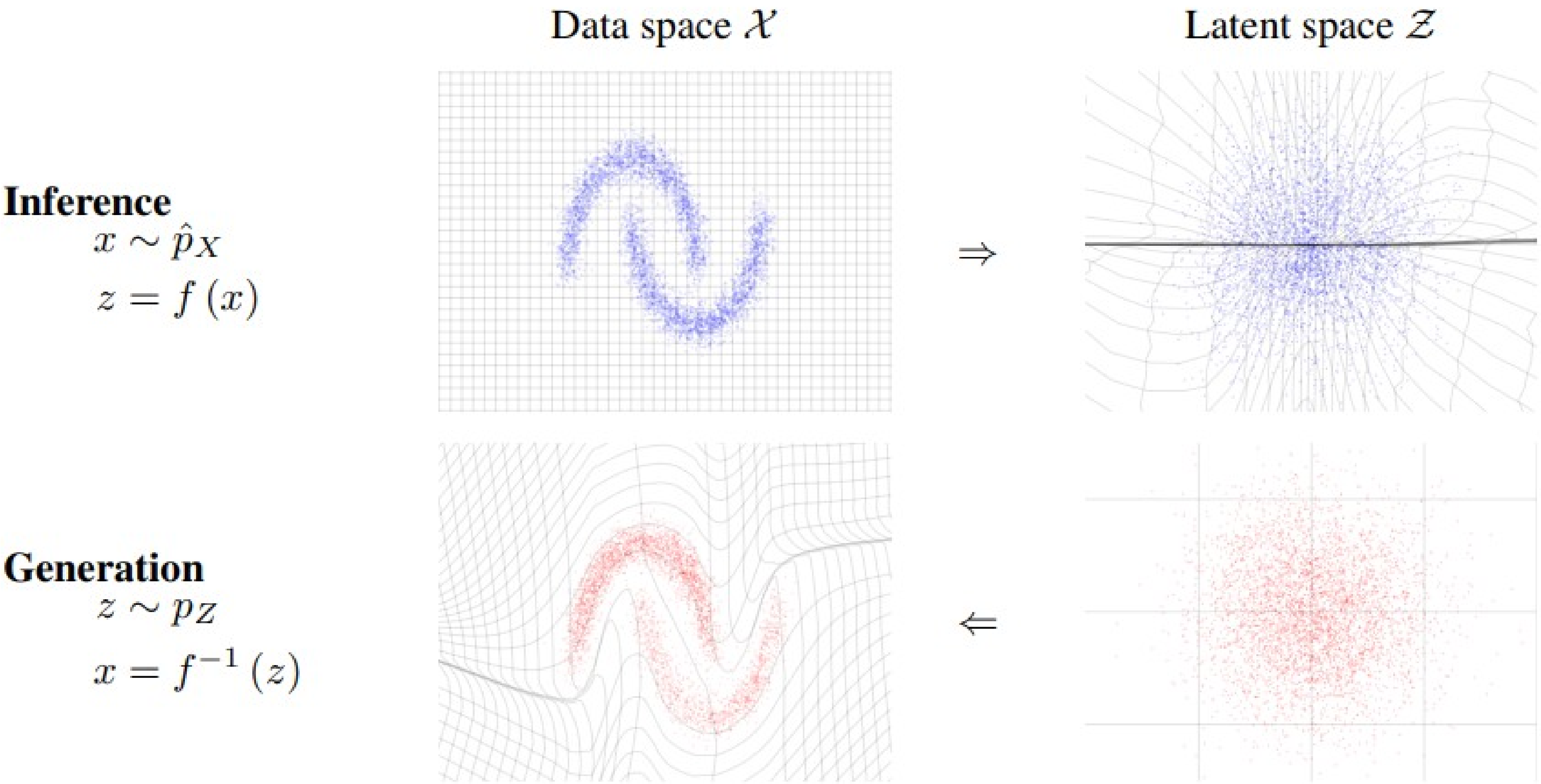
# RealNVP




(a) Forward propagation (b) Inverse propagation

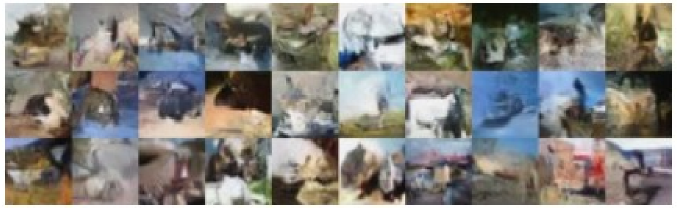
$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$$

# RealNVP


Data space $\mathcal{X}$
Latent space $\mathcal{Z}$
**Inference**
$x \sim \hat{p}_X$
$z = f(x)$
$\Rightarrow$
**Generation**
$z \sim p_Z$
$x = f^{-1}(z)$
$\Leftarrow$

# Samples generated via Real-NVP

# RealNVP: More Results



CIFAR-10

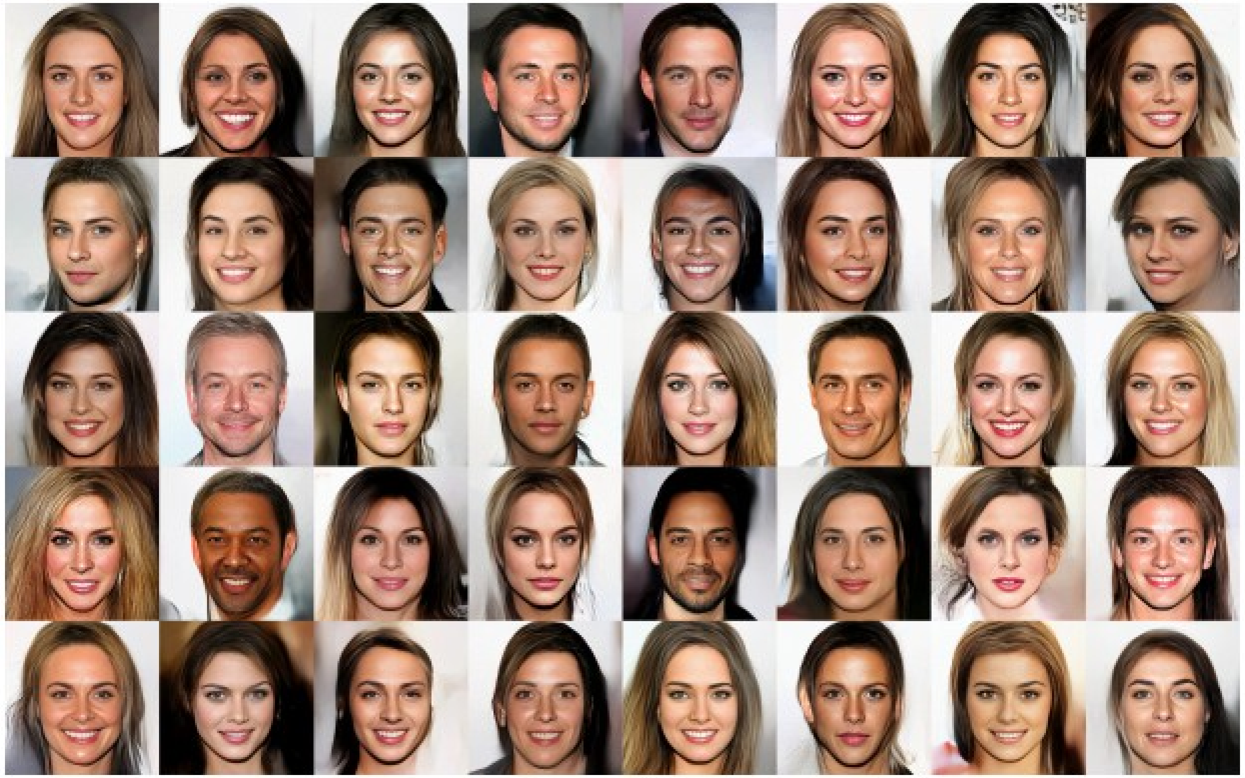
Imagenet (32x32)

Imagenet (64x64)

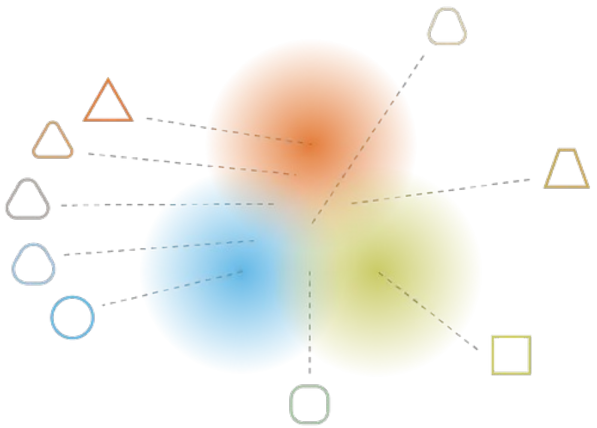
CelebA

LSUN

# GLOW: Result on CelebA



# GLOW: Interpolation



# GLOW: Sample Generated Face