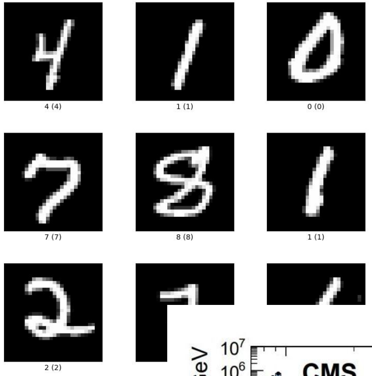
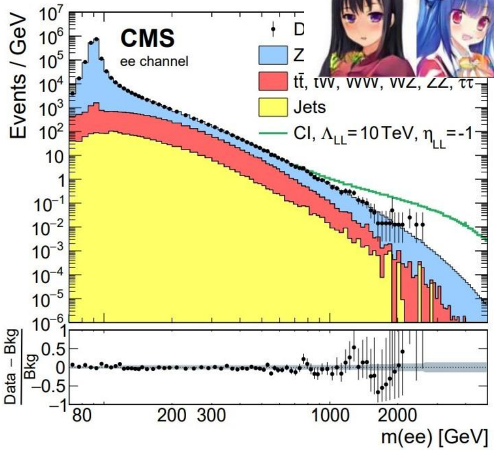
NF - Normalizing Flows

Alexander Demin
HSE University

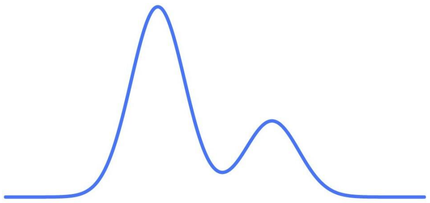
Generative models

trying to learn the distribution of our data





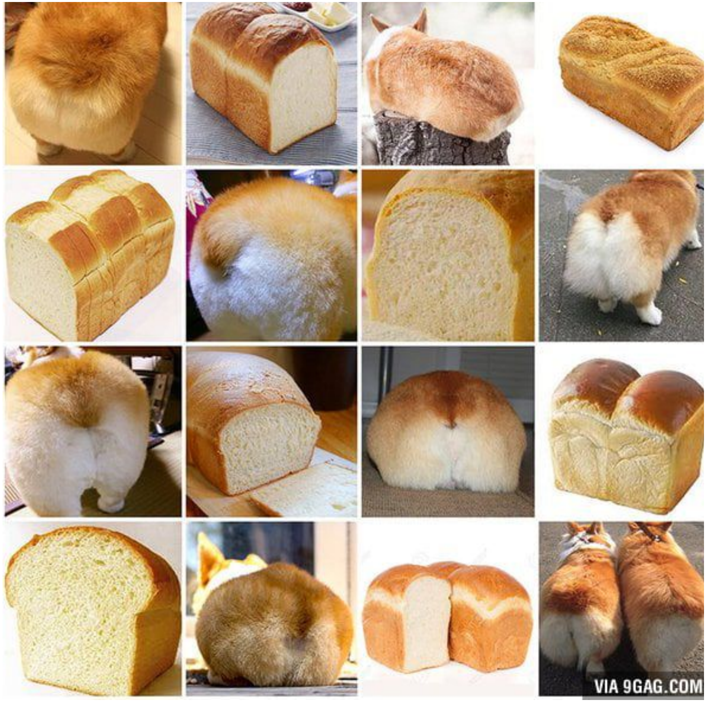
⇔



$$p_x(x)$$

generally unknown and complex

Sampling & Evaluating



$$p_x(x)$$

→ sample X from p

← assess p(X)

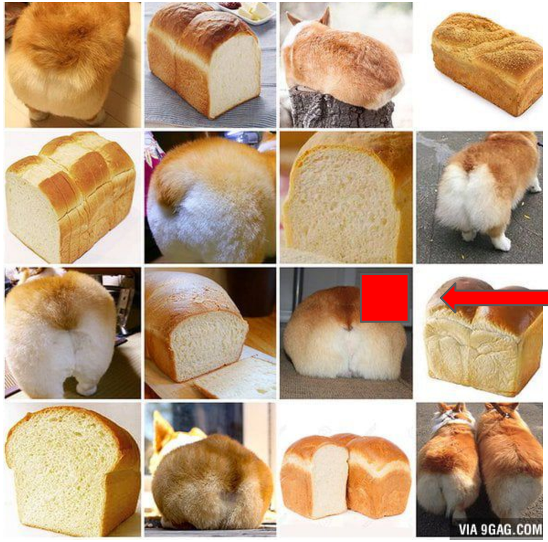
What are Normalizing Flows?

Normalizing Flows are generative models built on **invertible transformations**

- Easy to compute p(x)
- Easy to sample from p(x)
- Straightforward to train

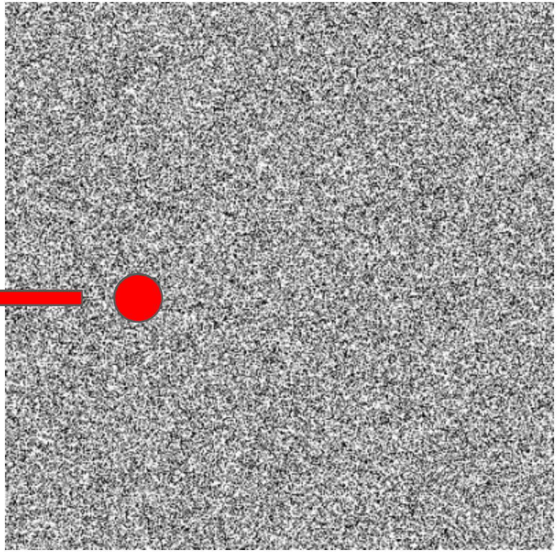
GLOW 2018

The Idea behind



$$z = f(x)$$

$$x = f^{-1}(z)$$



$p_x(x)$

complex and unknown

$p_z(z)$

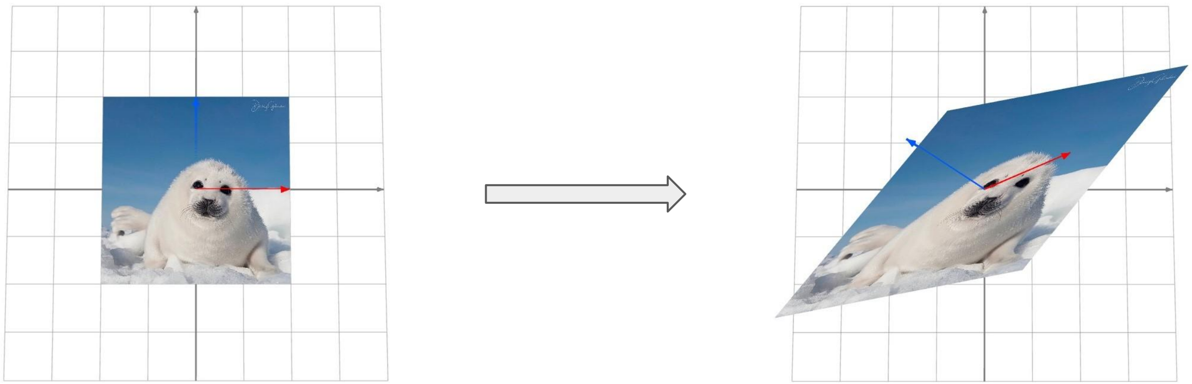
simple and known

Loss Function

We can make use of Max - Likelihood !

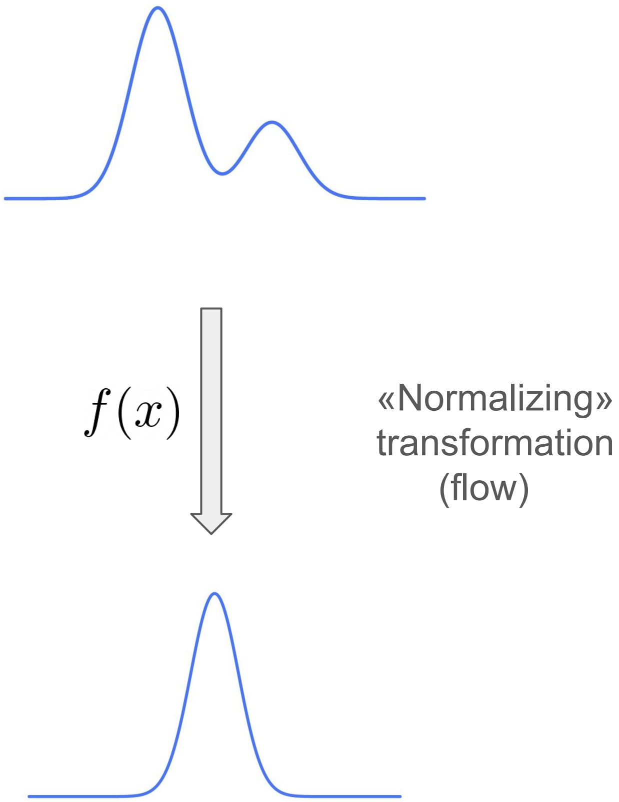
$$\max_{\theta} \; p_x(X) \Leftrightarrow \max_{\theta} \; p_z(f(X|\theta)) \left|\frac{\partial f}{\partial x}\right|$$

$$p_x(X) = \; p_z(f(X)) \left|\frac{\partial f}{\partial x^T}\right|$$

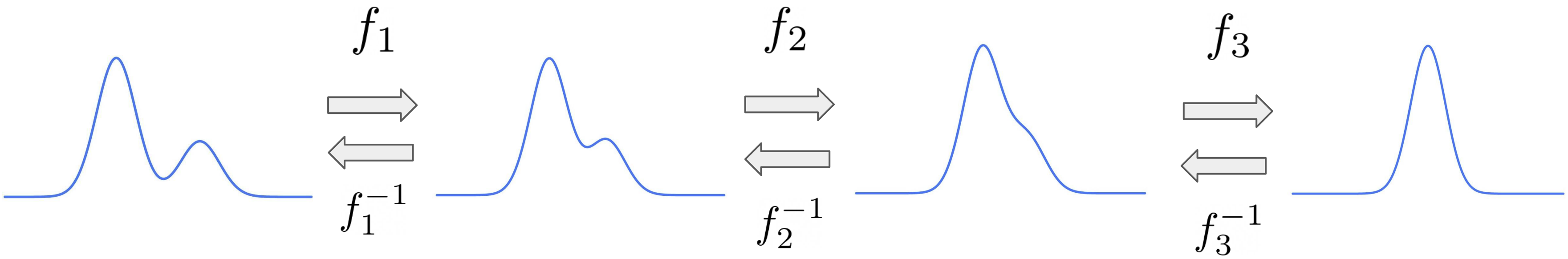


Conditions on f(x)

- Bijection (diffeomorphism)
- Tractable Jacobian
- Easy to compute the inverse



Composition of Flows



$$f = f_n \circ f_{n-1} \circ \ldots \circ f_1$$

$$f^{-1} = f_1^{-1} \circ f_2^{-1} \circ \ldots \circ f_n^{-1}$$

$$\left| \frac{\partial f}{\partial x} \right| = \prod_i \left| \frac{\partial f_i}{\partial x} \right|$$

Composition of Flows

$$\max_{\theta} \; p_x(X) \Leftrightarrow \max_{\theta} \; p_z(f(X|\theta)) \left| \frac{\partial f}{\partial x} \right|$$

$$\Downarrow$$

$$\max_{\theta} \; \sum_i log \; p_z(f(x_i|\theta)) + \sum_i log \left| \frac{\partial f_i}{\partial x} \right|$$

Linear Flows

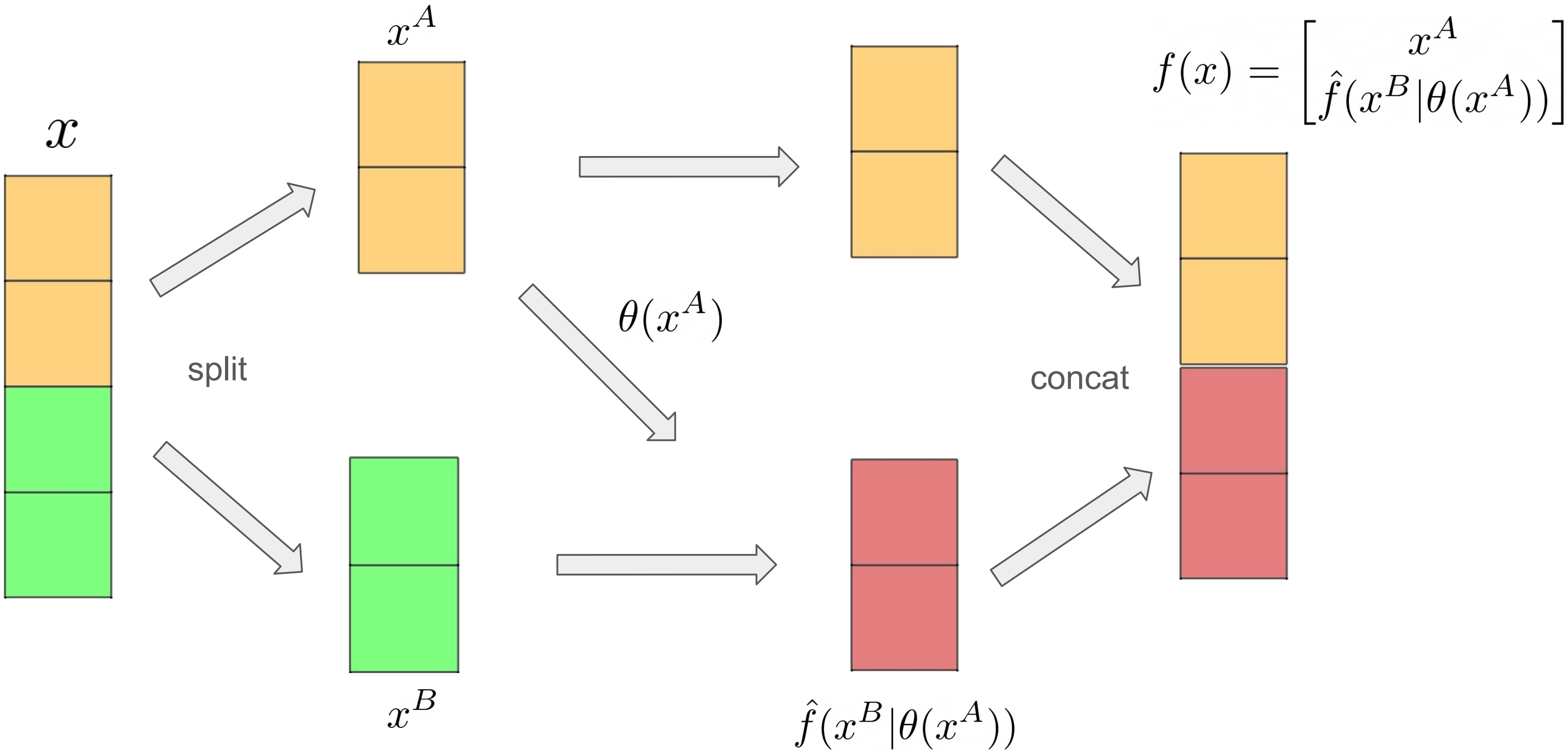
$$f(x) = Ax + b$$

- Generally invertible
- Jacobian is simply A
- Issues:
  - Inexpressive (closed under composition)
  - Determinant / inverse could be O(d^3)

	Inverse	Determinant
Full	O(d^3)	O(d^3)
Diagonal	O(d)	O(d)
Triangular	O(d^2)	O(d)
LU	O(d^2)	O(d)

Coupling Flows

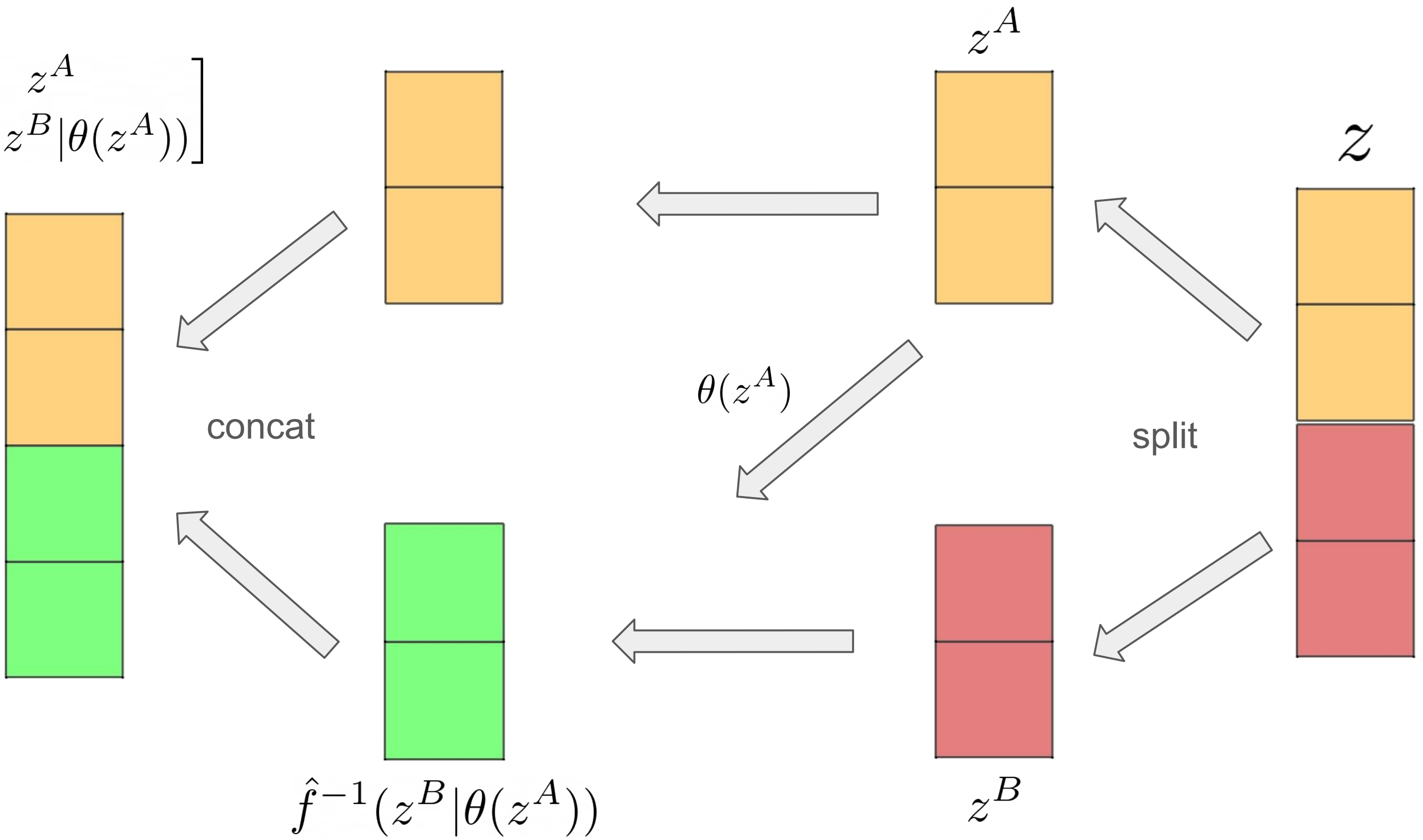
a general approach to construct non-linear flows



Coupling Flows

a general approach to construct non-linear flows

$$f^{-1}(z) = \begin{bmatrix} z^A \\ \hat{f}^{-1}(z^B|\theta(z^A)) \end{bmatrix}$$



Coupling Flows

a general approach to construct non-linear flows

$$f(x) = \begin{bmatrix} x^A \\ \hat{f}(x^B | \theta(x^A)) \end{bmatrix}$$

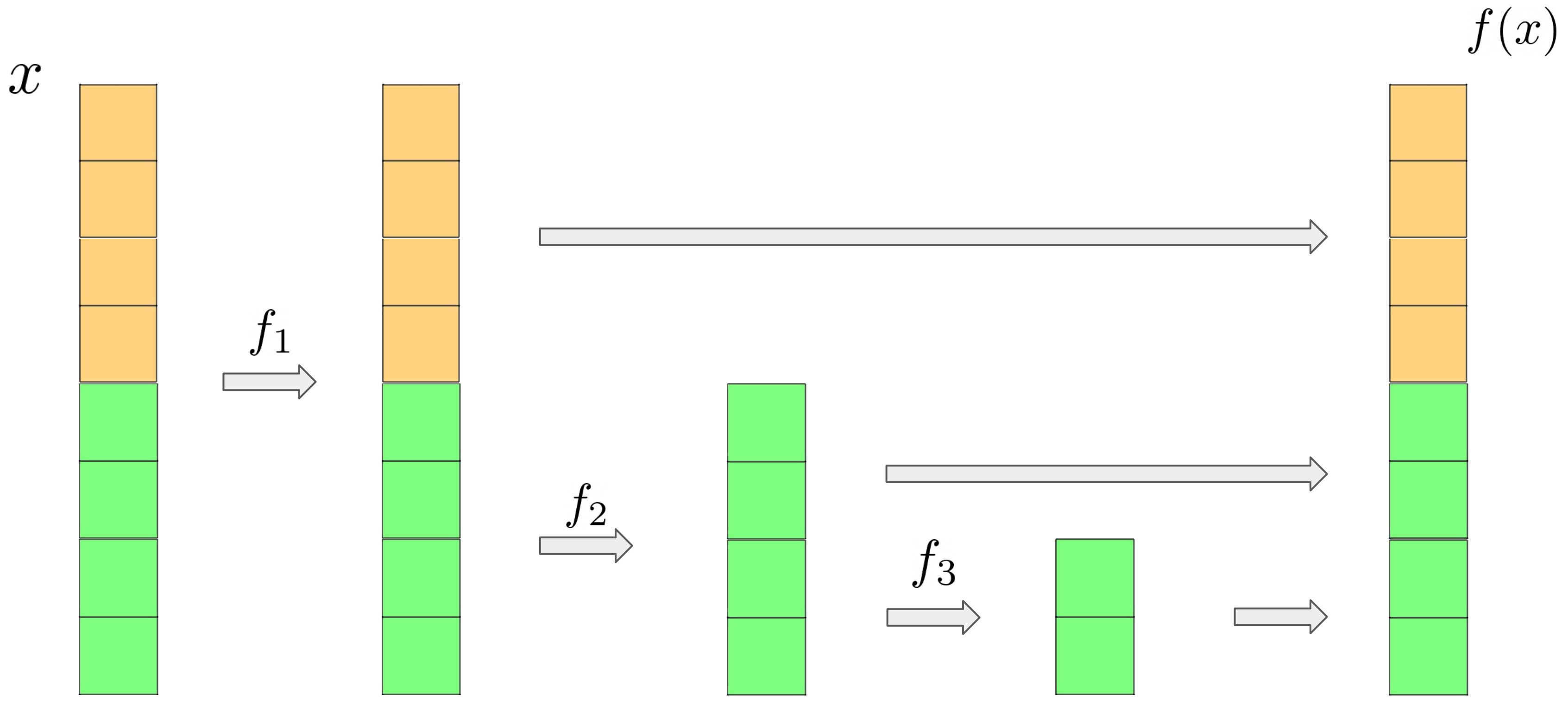
- Θ can be arbitrarily complex (Linear, CNN, …)
- $\hat{f}$ Must be invertible (Linear, PReLU …)

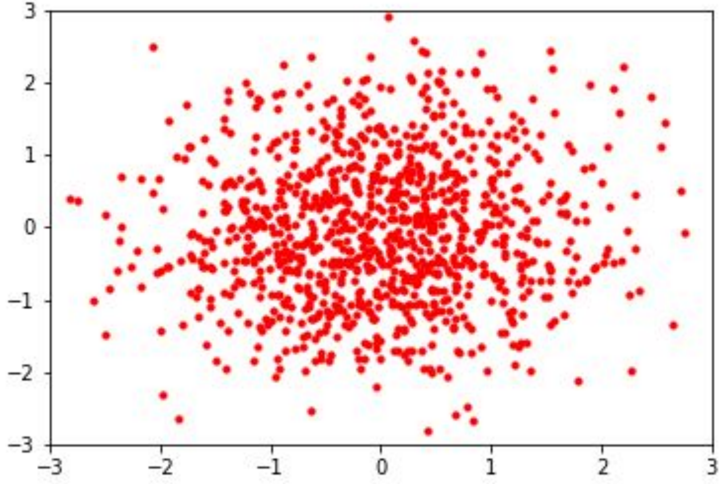
Jacobian

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial x^A}{\partial x^A} & \frac{\partial x^A}{\partial x^B} \\ \frac{\partial \hat{f}(x^B)}{\partial x^A} & \frac{\partial \hat{f}(x^B)}{\partial x^B} \end{bmatrix} = \begin{bmatrix} I & \mathbf{0} \\ \frac{\partial \hat{f}(x^B)}{\partial x^A} & \frac{\partial \hat{f}(x^B)}{\partial x^B} \end{bmatrix}$$

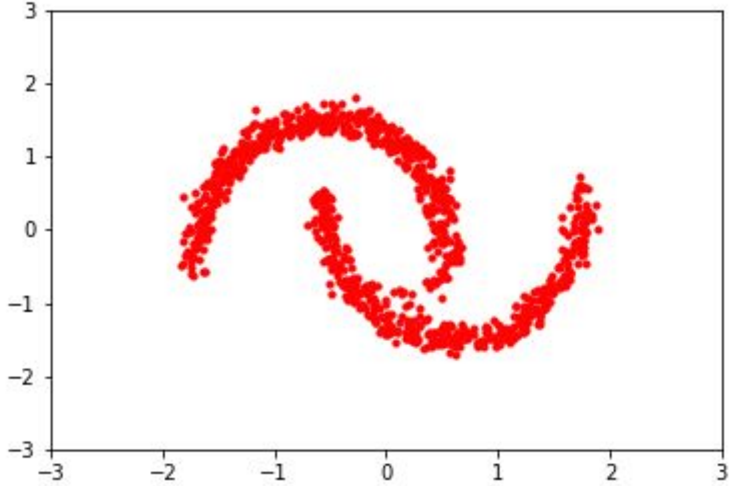
Multiscale Flows

dimensionality reduction





Sampling



Normalizing

Thanks !

References

- slide 2: Mnist (MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges), CERN CMS (not published article), Anime girls (MakeGirlsMoe - Create Anime Characters with A.I.!)
- slide 3: free pics from Google
- slide 4: GLOW ([1807.03039] Glow: Generative Flow with Invertible 1x1 Convolutions (arxiv.org))
- slide 5-6: free pics from Google
- slide 15: Moons (Eric Jang: Tips for Training Likelihood Models (evjang.com))