

Generative AI Models

ECE 598 LV – Lecture 3

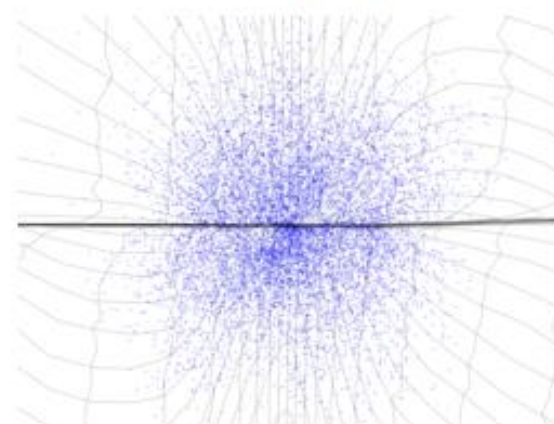
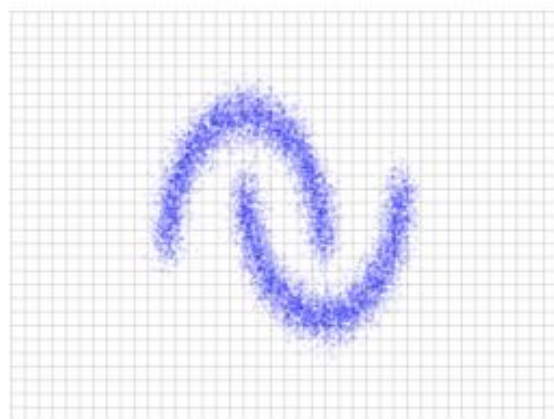
Lav R. Varshney

27 January 2022

Data space \mathcal{X} Latent space \mathcal{Z} **Inference**

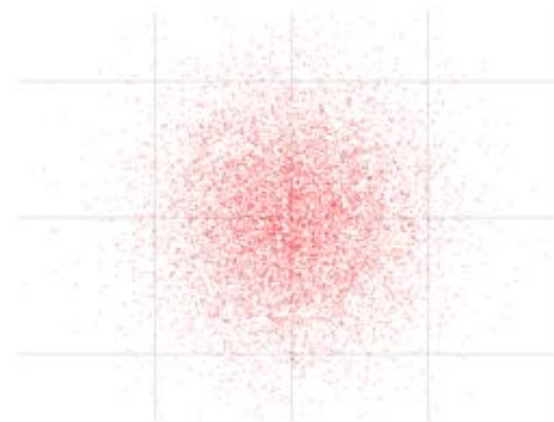
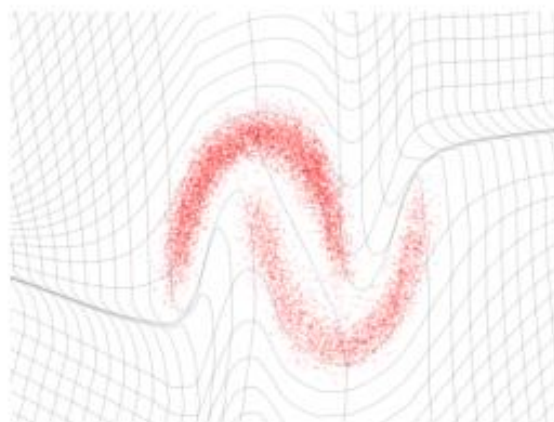
$$x \sim \hat{p}_X$$

$$z = f(x)$$

**Generation**

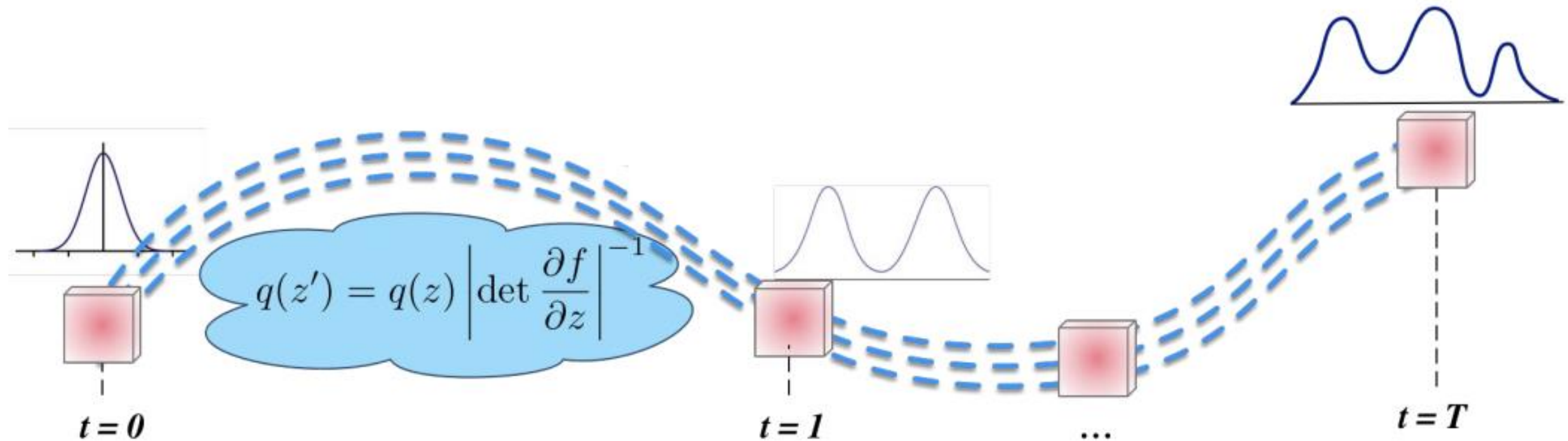
$$z \sim p_Z$$

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



an invertible, stable, mapping between a data distribution \hat{p}_X and a latent distribution p_Z (typically a Gaussian). Here we show a mapping that has been learned on a toy 2-d dataset. The function $f(x)$ maps samples x from the data distribution in the upper left into approximate samples z from the latent distribution, in the upper right. This corresponds to exact inference of the latent state given the data. The inverse function, $f^{-1}(z)$, maps samples z from the latent distribution in the lower right into approximate samples x from the data distribution in the lower left. This corresponds to exact generation of samples from the model. The transformation of grid lines in \mathcal{X} and \mathcal{Z} space is additionally illustrated for both $f(x)$ and $f^{-1}(z)$.

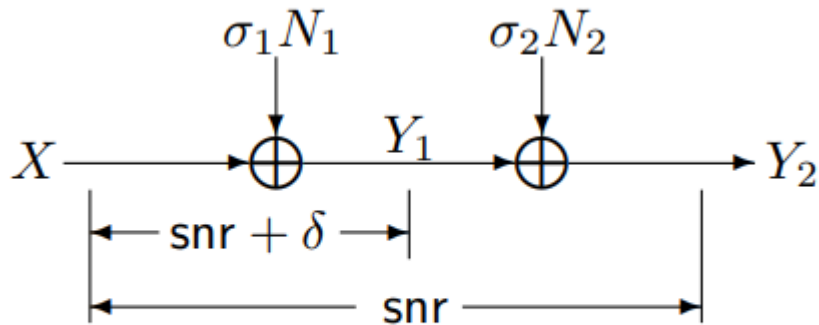
Normalizing Flows



Distribution flows through a sequence of invertible transforms

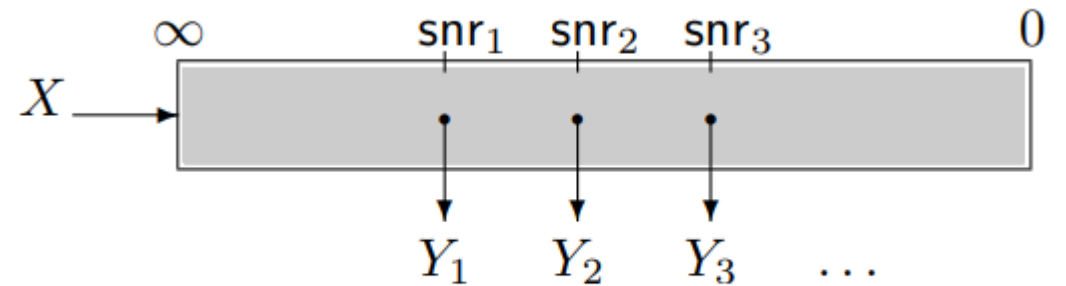
Rezende and Mohamed, 2015

Not Normalizing Flows



An SNR-incremental Gaussian channel.

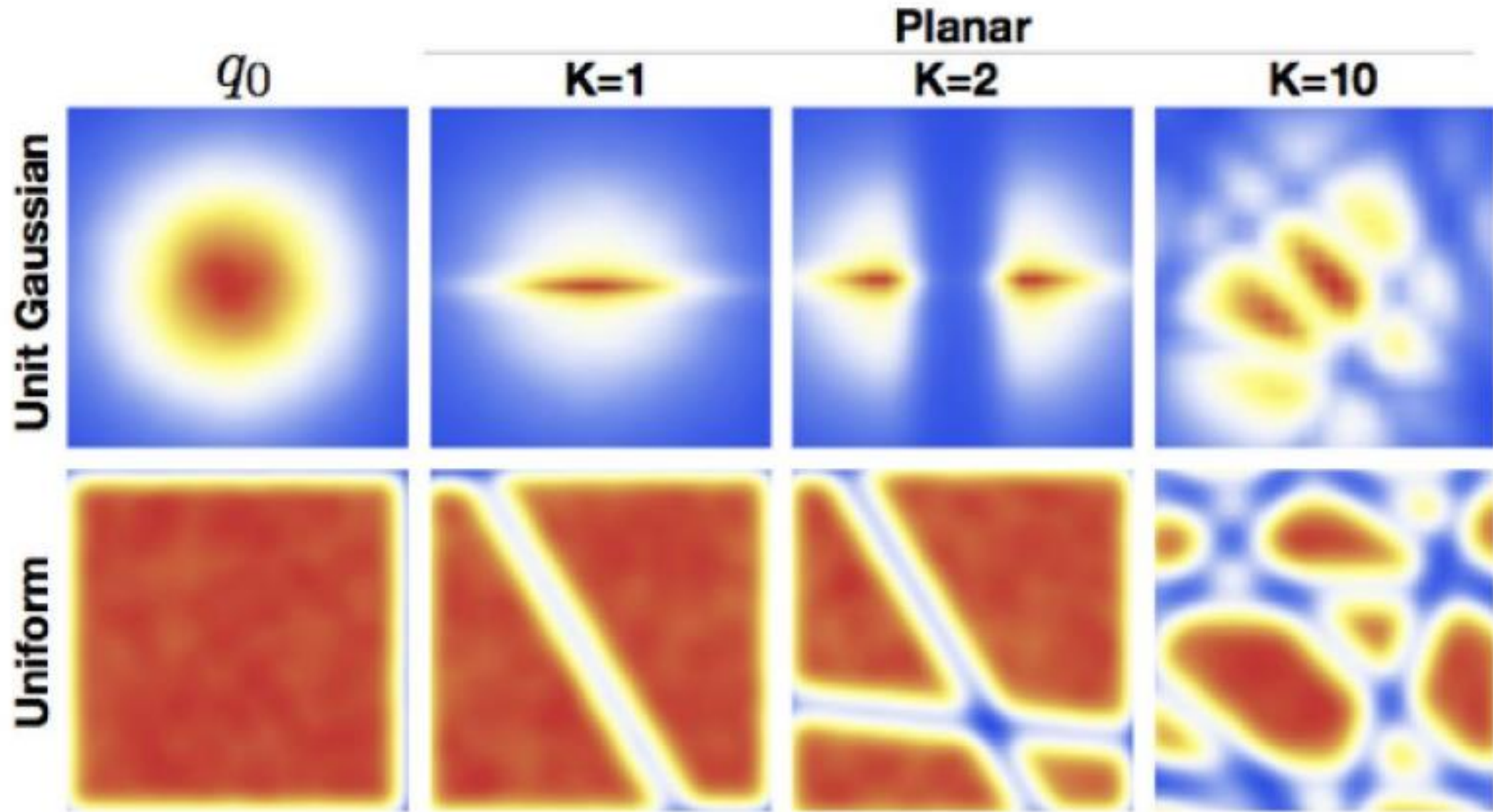
finite

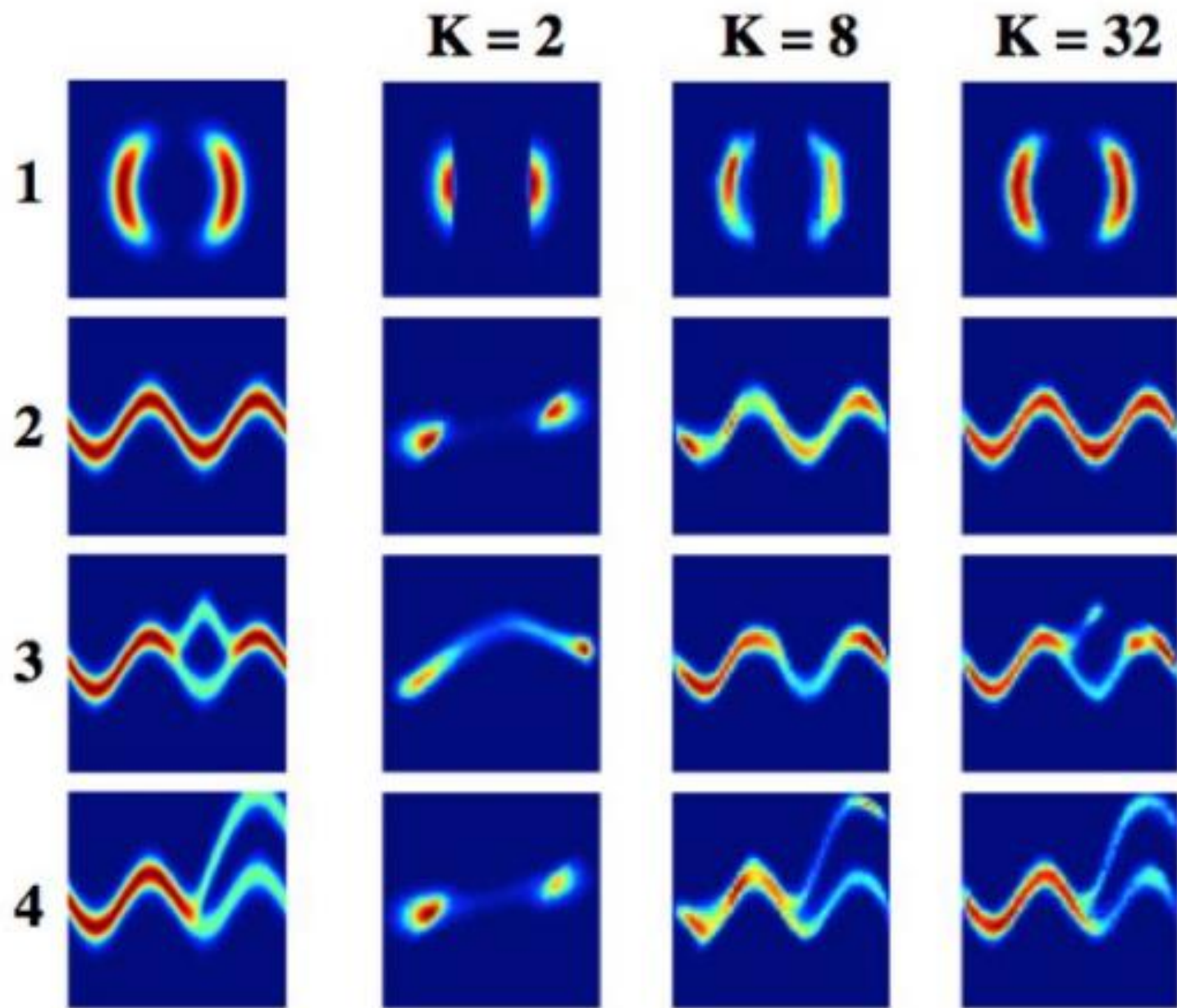


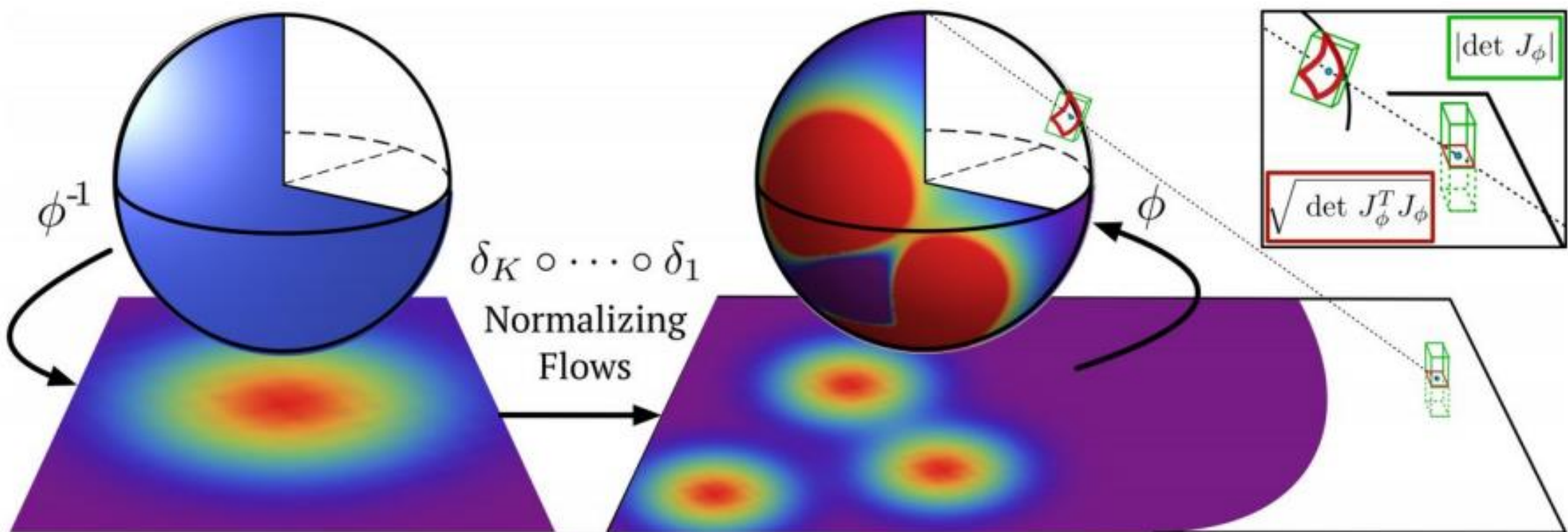
A Gaussian pipe where noise is added gradually.

infinitesimal

Dongning Guo, S. Shamai and S. Verdú, "Mutual information and minimum mean-square error in Gaussian channels," *IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1261-1282, April 2005.

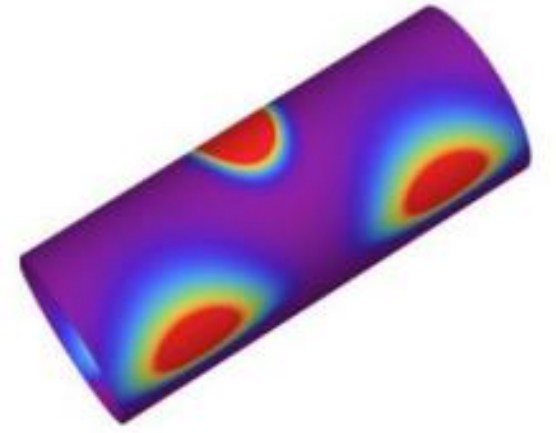
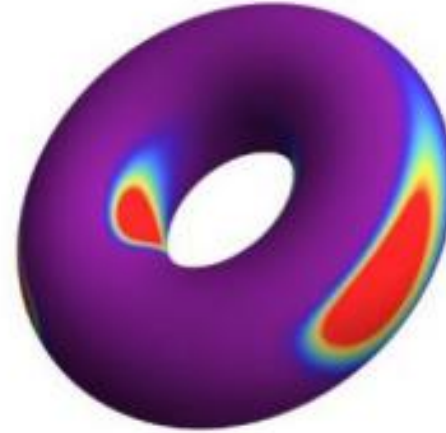
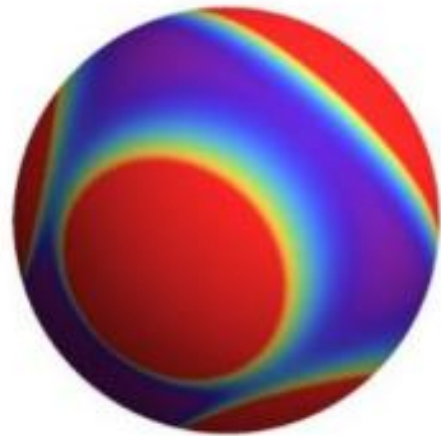
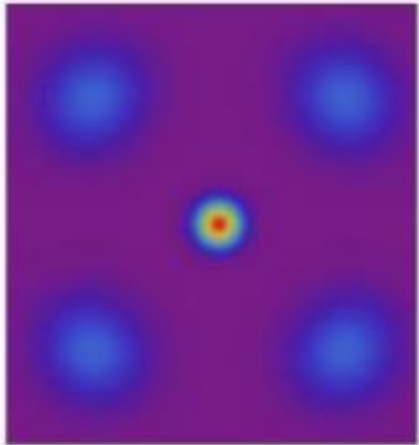
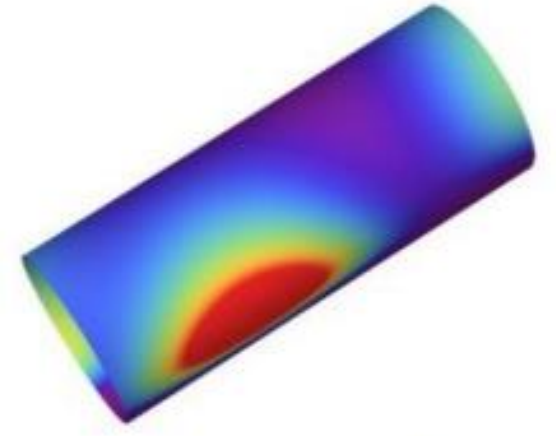
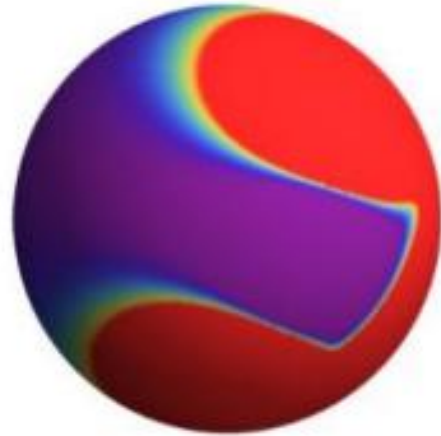
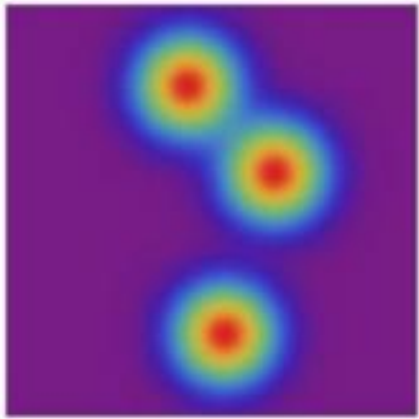






$$\log q_K(\mathbf{z}_K) = \log q_0(\mathbf{z}_0) - \frac{1}{2} \sum_{k=1}^K \log \det \left| \mathbf{J}_\phi^\top \mathbf{J}_\phi \right|$$

Gemici et al., 2016

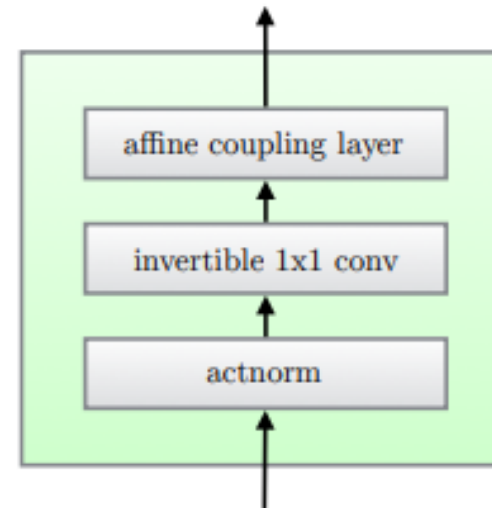


GLOW architecture

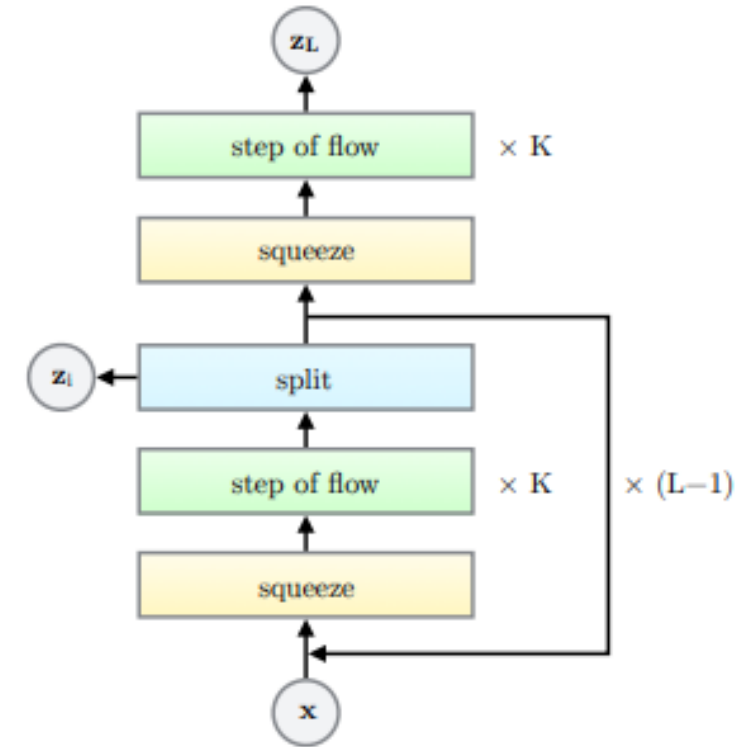
affine coupling layer basically learns a nonlinear independent components representation with easy inverse and easy log det Jacobian, since disentangling is a good representation

invertible convolution is generalization of learned rotation

activation normalization (actnorm) learns affine transformation of activations using a scale and bias parameter per channel



(a) One step of our flow.



(b) Multi-scale architecture (Dinh et al., 2016).

GLOW architecture

Description	Function	Reverse Function	Log-determinant
Actnorm. See Section 3.1.	$\forall i, j : \mathbf{y}_{i,j} = \mathbf{s} \odot \mathbf{x}_{i,j} + \mathbf{b}$	$\forall i, j : \mathbf{x}_{i,j} = (\mathbf{y}_{i,j} - \mathbf{b})/\mathbf{s}$	$h \cdot w \cdot \text{sum}(\log \mathbf{s})$
Invertible 1×1 convolution. $\mathbf{W} : [c \times c]$. See Section 3.2.	$\forall i, j : \mathbf{y}_{i,j} = \mathbf{W}\mathbf{x}_{i,j}$	$\forall i, j : \mathbf{x}_{i,j} = \mathbf{W}^{-1}\mathbf{y}_{i,j}$	$h \cdot w \cdot \log \det(\mathbf{W}) $ or $h \cdot w \cdot \text{sum}(\log \mathbf{s})$ (see eq. (10))
Affine coupling layer. See Section 3.3 and (Dinh et al., 2014)	$\mathbf{x}_a, \mathbf{x}_b = \text{split}(\mathbf{x})$ $(\log \mathbf{s}, \mathbf{t}) = \text{NN}(\mathbf{x}_b)$ $\mathbf{s} = \exp(\log \mathbf{s})$ $\mathbf{y}_a = \mathbf{s} \odot \mathbf{x}_a + \mathbf{t}$ $\mathbf{y}_b = \mathbf{x}_b$ $\mathbf{y} = \text{concat}(\mathbf{y}_a, \mathbf{y}_b)$	$\mathbf{y}_a, \mathbf{y}_b = \text{split}(\mathbf{y})$ $(\log \mathbf{s}, \mathbf{t}) = \text{NN}(\mathbf{y}_b)$ $\mathbf{s} = \exp(\log \mathbf{s})$ $\mathbf{x}_a = (\mathbf{y}_a - \mathbf{t})/\mathbf{s}$ $\mathbf{x}_b = \mathbf{y}_b$ $\mathbf{x} = \text{concat}(\mathbf{x}_a, \mathbf{x}_b)$	$\text{sum}(\log(\mathbf{s}))$

Normalizing Flows

<https://openai.com/blog/glow/>

Normalizing Flows

https://youtu.be/wLG4JN_q8Ac?t=10963