VARIATIONAL INFERENCE WITH NORMALIZING FLOWS

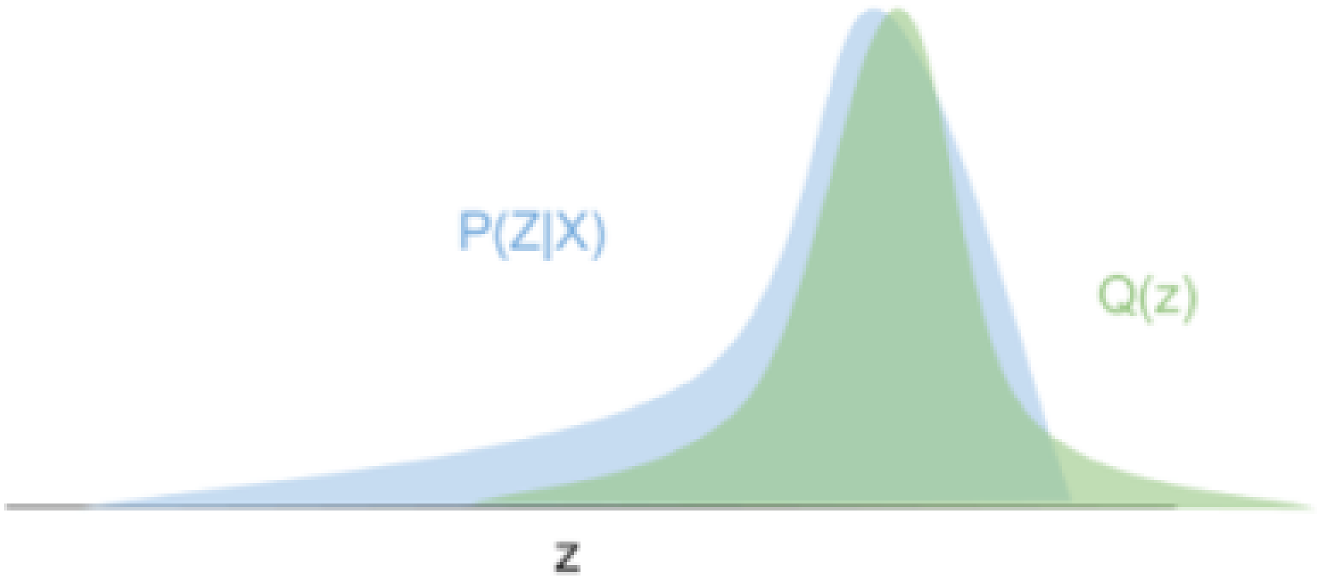
ICML 2015

박성현

Data and Visual Analytics Lab

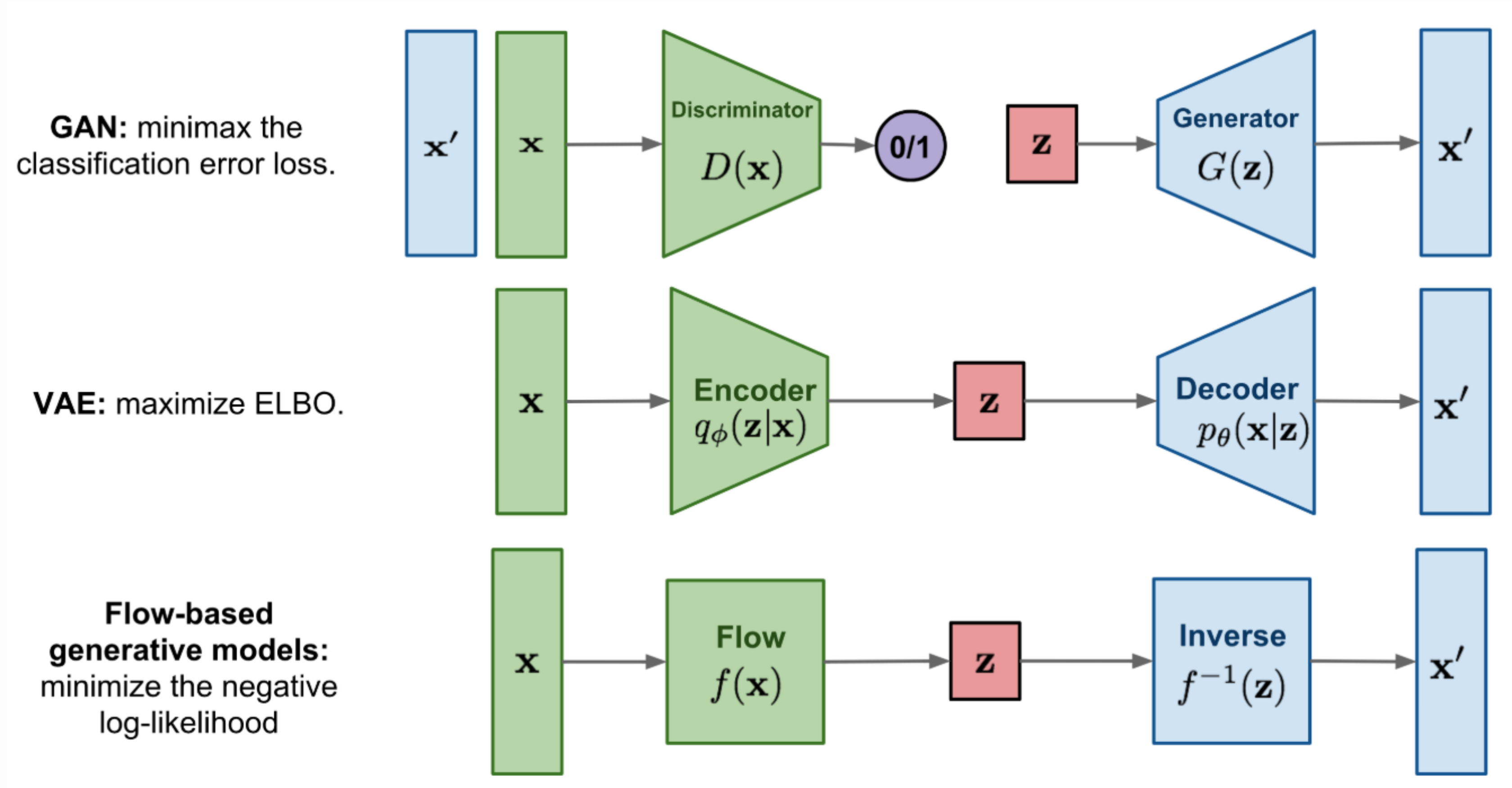
Background: Variational Inference

- Variational Inference : “$posterior$” 분포 $p(z|x)$를 다루기 쉬운 확률분포 $q(z)$로 근사 → Approximate ***posterior*** distribution



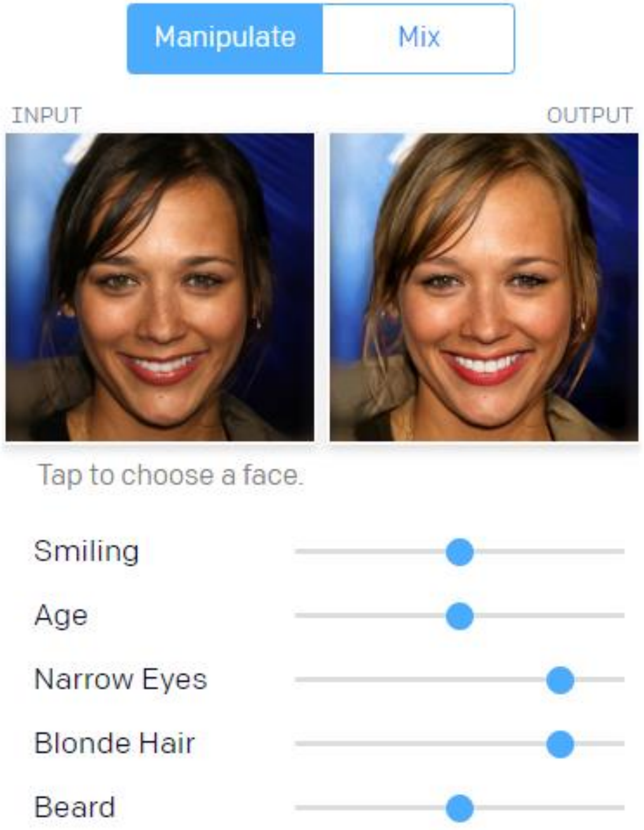
Background: Flow-Based Model

- Flow-based generative models allow to learn highly non-Gaussian posterior densities by learning invertible transformations of simple densities into complex ones.



Background: Flow-Based Model

- Flow-based generative models allow to learn highly non-Gaussian posterior densities by learning invertible transformations of simple densities into complex ones.



[Glow results & demo]

Background: advantages of Flow-Based Model

- Exact latent-variable inference and log-likelihood evaluation
  - In VAEs, one can only approximately infer the value of latent variables corresponding to data points.
  - GANs have no encoder at all to infer the latents.
  - In reversible generative models, this can be done exactly without approximation.

- Efficient inference and synthesis
  - While autoregressive models, such as Pixel-RNNs (and Pixel CNNs) are also reversible, synthesis from such models are hard to parallelize.
  - Flow-based models are efficient to parallelize for both inference and synthesis.

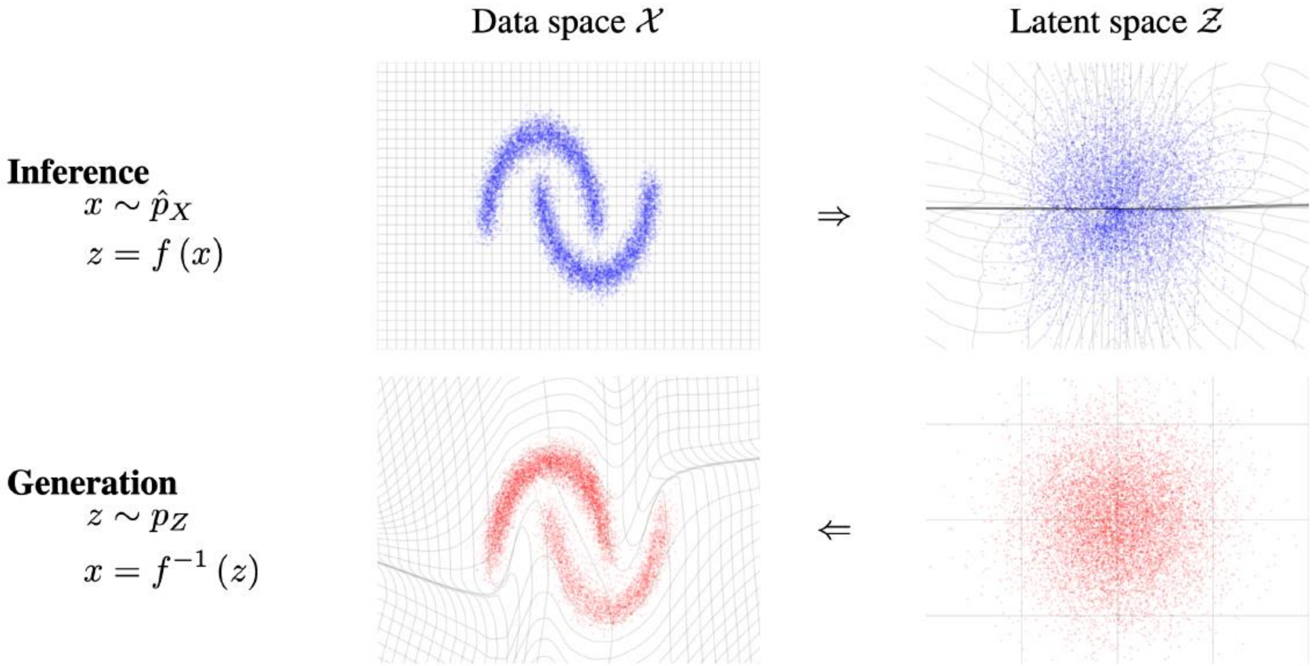
Background: advantages of Flow-Based Model

- Useful latent space for downstream tasks
  - The hidden layers of autoregressive models have unknown marginal distributions, making it more difficult to perform valid manipulation of data.
  - In GANs, datapoints may not be directly represented in a latent space since they have no encoder and may not have full support over the data distribution.

- Significant potential for memory savings
  - Computing gradients in reversible neural networks requires an amount of memory that is constant instead of linear in their depth.

Background: Flow-Based Model

- In flow-based models, given a data instance $x \in X$, a prior p_Z on a latent variable $z \in Z$, and a bijection (one-to-one mapping) $f: X \to Z$, we use the change of variable to model distribution on X:



$$p_X(x) = p_Z(f(x)) \left| \det \left(\frac{\partial f(x)}{\partial x^T} \right) \right|$$

Background: Change of variable

- Change of variable : a basic technique used to simplify problems in which the original variables are replaced with functions of other variables.

Theorem: *Assume* Ω_1 *and* Ω_2 *are open sets in* $\mathbb{R}^n$. *Assume that*

$$\boxed{\Omega_1 \xrightarrow{\Phi} \Omega_2}$$

is a bijection of class C^1 *whose inverse is also of class* C^1. *Assume that* f *is a Lebesgue measurable function on* Ω_2. *Then* $f \circ \Phi$ *is Lebesgue measurable on* Ω_1 *and*

$$\boxed{\int_{\Omega_2} f(y)dy = \int_{\Omega_1} f(\Phi(x))|J(x)|dx.}$$

This formula is valid in two senses: If $f \geq 0$, *then it is true without further qualification. In general,* $f \in L^1(\Omega_2)$ *if and only if* $f \circ \Phi\,|J| \in L^1(\Omega_1)$, *and then the formula is valid.*

Background: Change of variable

Original density $$\mathbb{E}[f(X)] = \int f(x)p_X(x)dx$$

Transform

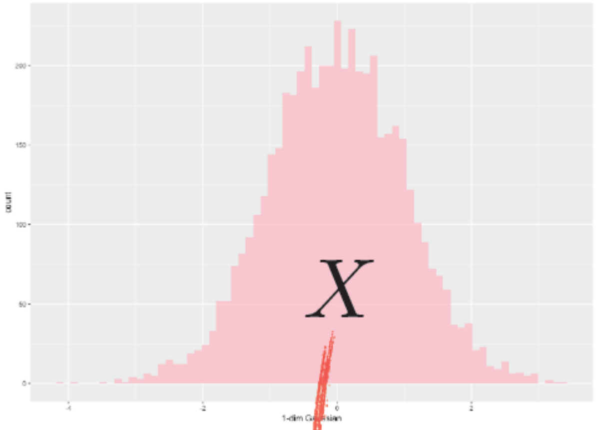
$$\mathbb{E}[f(\Phi(X))] = \int f(\Phi(x))p_X(x)dx$$

$$||$$

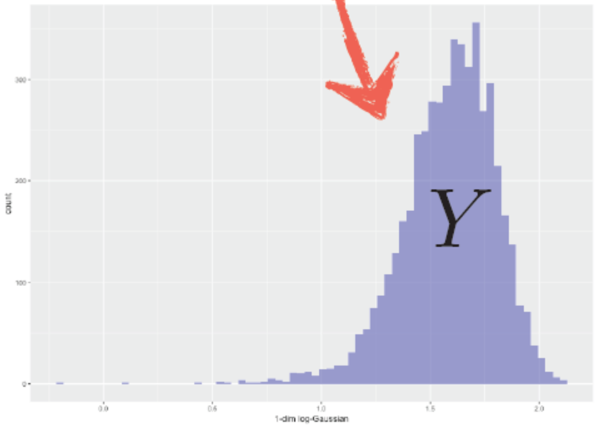
$$\mathbb{E}[f(\Phi(X))] = \int f(\Phi(x))p_Y(\Phi(x))|J(x)|dx$$

Change of Variable

Target density $$\mathbb{E}[f(Y)] = \int f(y)p_Y(y)dy$$

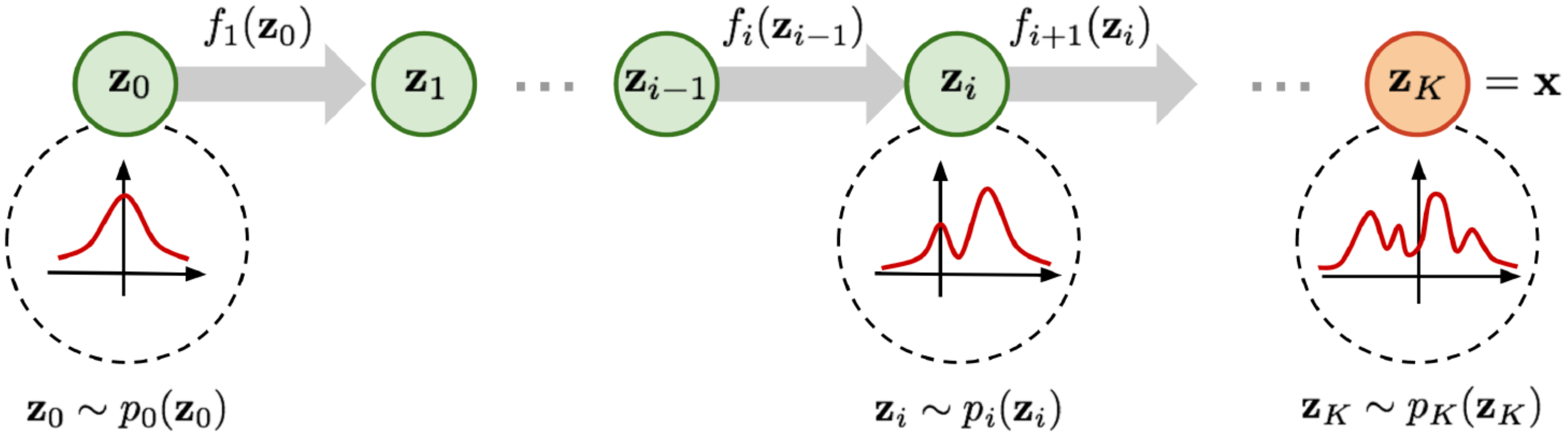


Transform Φ



Normalizing Flow

- A normalizing flow transforms a probability density through a sequence of invertible mappings.



- By repeatedly applying the rule for change of variables, the initial density flows through the sequence of invertible mappings and yields a valid probability distribution.

Normalizing Flow

- The basic rule for transformation of densities considers an invertible, smooth mapping $f: \mathbb{R}^d \rightarrow \mathbb{R}^d$ with inverse $f^{-1} = g$.
- If we use this mapping to transform a random variable z with distribution $q(z)$, the resulting random variable $z' = f(z)$ has a distribution:

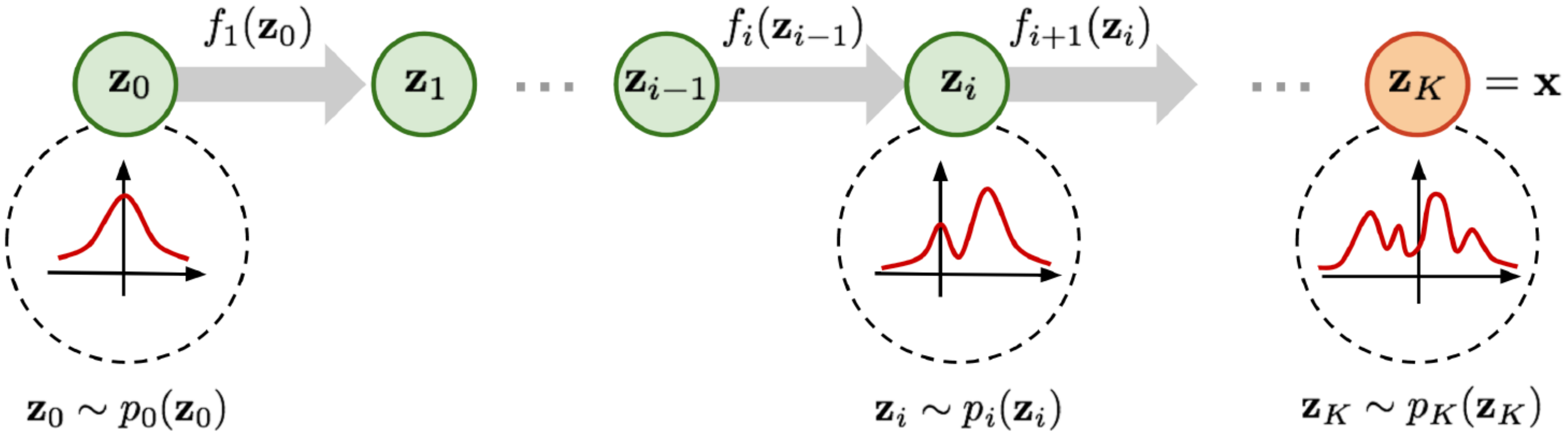
$$q(z') = q(z) \left| \det \frac{\partial f^{-1}}{\partial z'} \right| = q(z) \left| \det \frac{\partial f}{\partial z} \right|^{-1}$$

- The last equality comes from:

$$\frac{\partial f^{-1}(z')}{\partial z'} = \frac{\partial z}{\partial z'} = \left(\frac{\partial z'}{\partial z} \right)^{-1} = \left(\frac{\partial f(z)}{\partial z} \right)^{-1}$$

Normalizing Flow

- We can construct arbitrarily complex densities from a simple distribution z_0 composing f_i and successively applying the invertible transformations: $f_k \circ \cdots \circ f_2 \circ f_1$



$$q(z_{i+1}) = q(z_i) \left| \det \frac{\partial f_i^{-1}}{\partial z_{i+1}} \right| = q(z_i) \left| \det \frac{\partial f_i}{\partial z_i} \right|^{-1}$$

Normalizing Flow

- To allow for scalable inference using finite normalizing flows, we must specify:
  - a class of invertible transformations f that can be used
  - an efficient mechanism for computing the determinant of the Jacobian.

- While we can build invertible parameteric functions with invertible neural networks, they typically have a complexity of $O(LD^3)$ for computing the Jacobian determinants. (L: # of layers, D: latent dimensionality.)

- Thus flow-based models use functions f that allow for low-cost computation of the Jacobian determinants, or completely eliminates the needs of computing it.

Invertible Linear-time Transformations

(1) Planar Flows $f(z) = z + uh(w^T z + b)$

$$\psi(\mathbf{z}) = h'(\mathbf{w}^\top \mathbf{z} + b)\mathbf{w}$$

$$\left|\det \frac{\partial f}{\partial \mathbf{z}}\right| = |\det(\mathbf{I} + \mathbf{u}\psi(\mathbf{z})^\top)| = |1 + \mathbf{u}^\top \psi(\mathbf{z})|.$$

$$\mathbf{z}_K = f_K \circ f_{K-1} \circ \ldots \circ f_1(\mathbf{z})$$

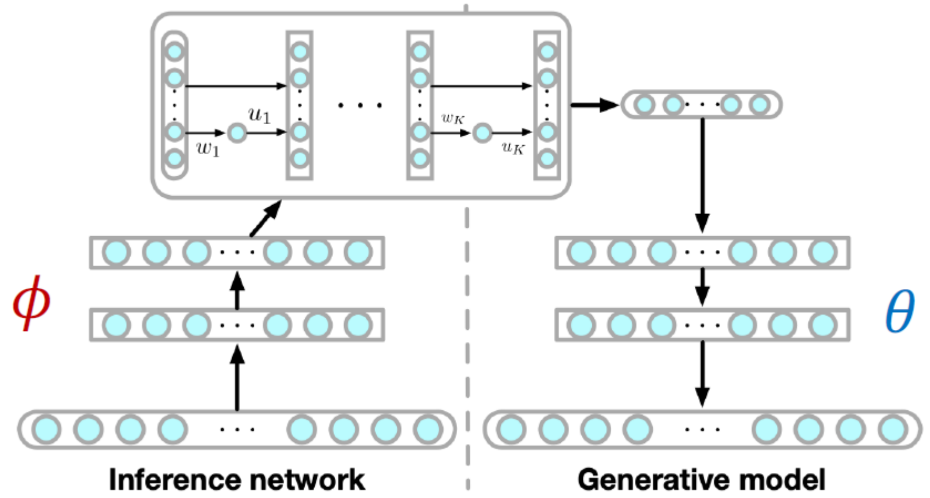
$$\ln q_K(\mathbf{z}_K) = \ln q_0(\mathbf{z}) - \sum_{k=1}^{K} \ln |1 + \mathbf{u}_k^\top \psi_k(\mathbf{z}_{k-1})|.$$

(2) Radial Flows $f(z) = z + \dfrac{\beta(z - z_0)}{\alpha + |z - z_0|}$

Variational Inference with Normalizing Flow

Algorithm 1 Variational Inf. with Normalizing Flows

Parameters: ϕ variational, $\boldsymbol{\theta}$ generative
while not converged **do**
  $\mathbf{x} \leftarrow$ {Get mini-batch}
  $\mathbf{z}_0 \sim q_0(\bullet|\mathbf{x})$
  $\mathbf{z}_K \leftarrow f_K \circ f_{K-1} \circ \ldots \circ f_1(\mathbf{z}_0)$
  $\mathcal{F}(\mathbf{x}) \approx \mathcal{F}(\mathbf{x}, \mathbf{z}_K)$
  $\Delta\boldsymbol{\theta} \propto -\nabla_\theta \mathcal{F}(\mathbf{x})$
  $\Delta\phi \propto -\nabla_\phi \mathcal{F}(\mathbf{x})$
end while



- The algorithmic complexity of joint sampling and computing the log-det Jacobian terms of the inference model scales as $O(LN^2) + O(KD)$.
- L: # of layers, N: avg. hidden layer size, K: flow length, D: latent dimensionality.

Experiments

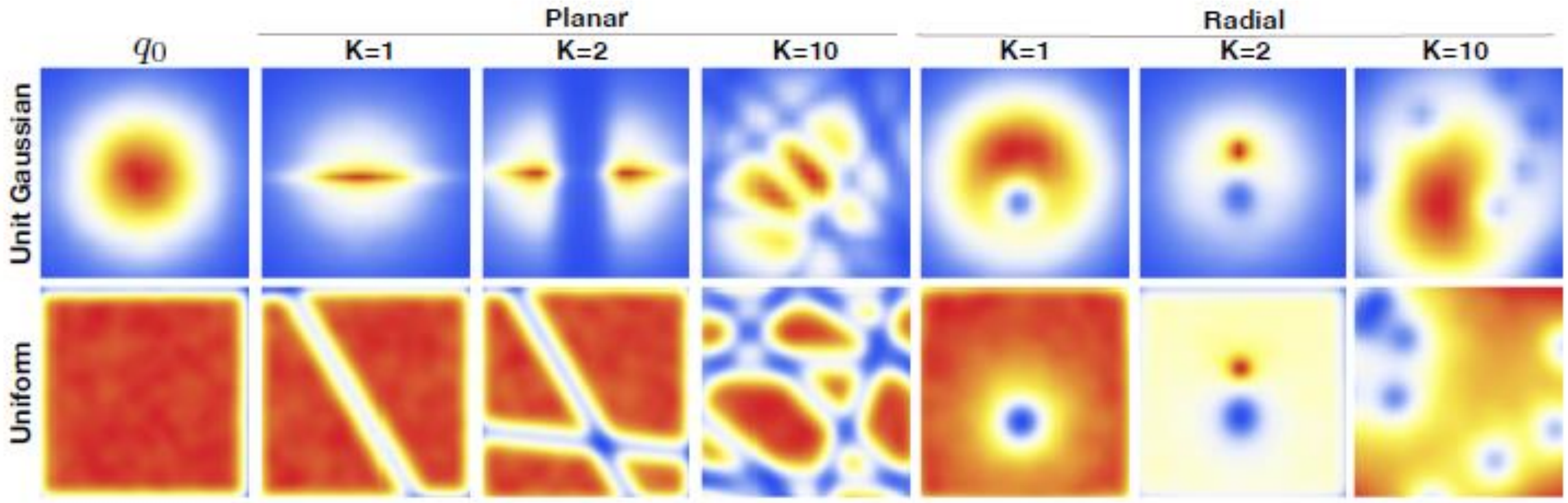
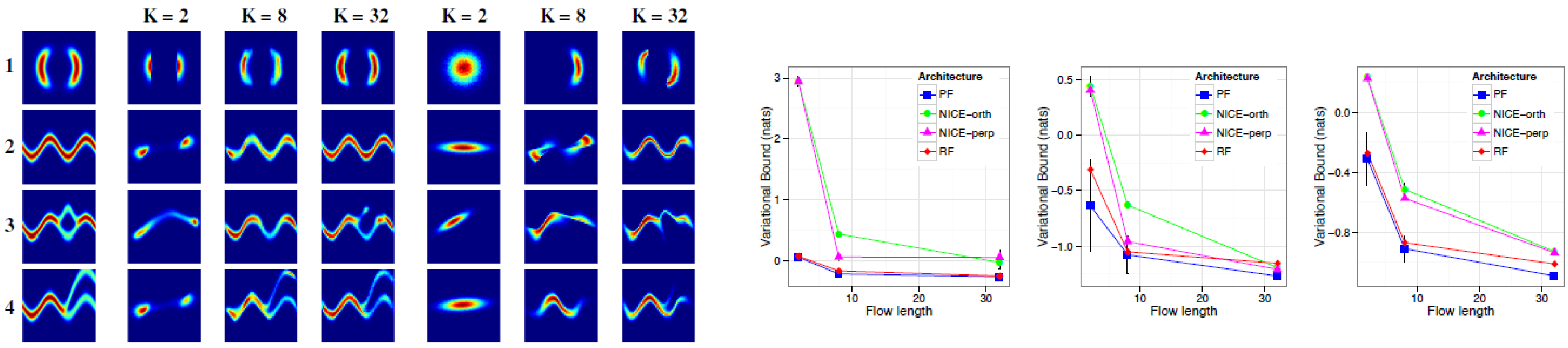


Figure 1. Effect of normalizing flow on two distributions.

Experiments



(a) (b) Norm. Flow (c) NICE (d) Comparison of KL-divergences.

Figure 3. Approximating four non-Gaussian 2D distributions. The images represent densities for each energy function in table 1 in the range $(-4, 4)^2$. (a) True posterior; (b) Approx posterior using the normalizing flow (13); (c) Approx posterior using NICE (19); (d) Summary results comparing KL-divergences between the true and approximated densities for the first 3 cases.

Thank you!

Background: Change of variable

- Given a random variable z which follows a known probability distribution $z \sim \pi(z)$, we can construct a new random variable z' using a one-to-one mapping function $z' = f(z)$.
- Since f is invertible, $z = f^{-1}(z')$.

$$\int p(z')dz' = \int \pi(z)\,dz = 1$$

- We infer the unknown probability density function of the new variable $p(z')$ using change of variables.

$$p(z') = \pi(z)\left|\frac{dz}{dz'}\right| = \pi\big(f^{-1}(z')\big)\left|\frac{df^{-1}}{dz'}\right| = \pi\big(f^{-1}(z')\big)|(f^{-1})'(z')|$$

Background: Change of variable

- The multivariate version is similar:

$$\mathbf{z} \sim \pi(\mathbf{z}), \qquad \mathbf{z}' \sim f(\mathbf{z}), \qquad \mathbf{z} = f^{-1}(\mathbf{z}')$$

$$p(\mathbf{z}') = \pi(z) \left| \det \frac{\partial \mathbf{z}}{\partial \mathbf{z}'} \right| = \pi\left(f^{-1}(\mathbf{z}')\right) \left| \det \frac{\partial f^{-1}}{\partial \mathbf{z}'} \right|$$

- However, we now need to compute the determinant of the Jacobian matrix.

Background: Jacobian Matrix

- Given a function $f: R^n \rightarrow R^m$, the matrix J of all first-order partial derivatives of this function is called the Jacobian matrix, where each entry $J_{ij} = \frac{df_i}{dx_j}$

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

- For a bijection (one-to-one mapping), the Jacobian matrix will be a square matrix.

Background: Determinant

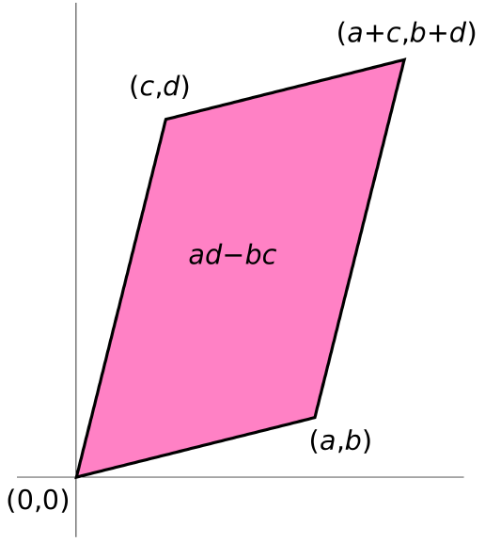
- The determinant of $n \times n$ matrix M is defined as:

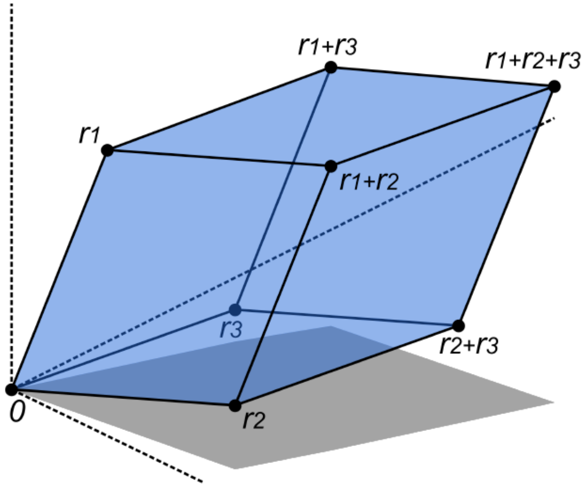
$$\det M = \det \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma_i}$$

- where σ is a permutation of $\{1,2,\ldots,n\}$ and S_n the set of all permutations.
- $\operatorname{sgn}(\sigma)$ is a function which returns 1 if the reordering given by σ can be achieved by interchanging two entries by an even number of times and -1 otherwise.

Background: Determinant

- For $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, $\det A = ad - bc$ represent the change in the area transformed by A.





- For higher-dimensional cases, the determinant represent the change in the volume.