Variational Inference with Normalising Flows
1st Presentation

Y.Chen, Q.Tian, A.Komissarov

TU Berlin

December 17, 2018

Overview

1. Normalizing Flows
   - Variational Inference
   - Principle of Normalizing Flows
   - Finite Flows
2. Invertible Linear-Time Transformations
   - Planar Flows
   - Radial Flows
3. Variational Auto-Encoder(VAE)
   - Variational Auto-Encoder(VAE)
   - VAE with Normalizing Flows
   - MLPs as probabilistic encoders and decoders

Normalizing Flows

Variational Inference

Main Idea:

The intractable posterior distributions $p(\mathbf{z}|\mathbf{x})$ are approximated by a class of well-known, simple probability distribution families $q(\mathbf{z})$, over which we search for the best approximation to the true posterior.

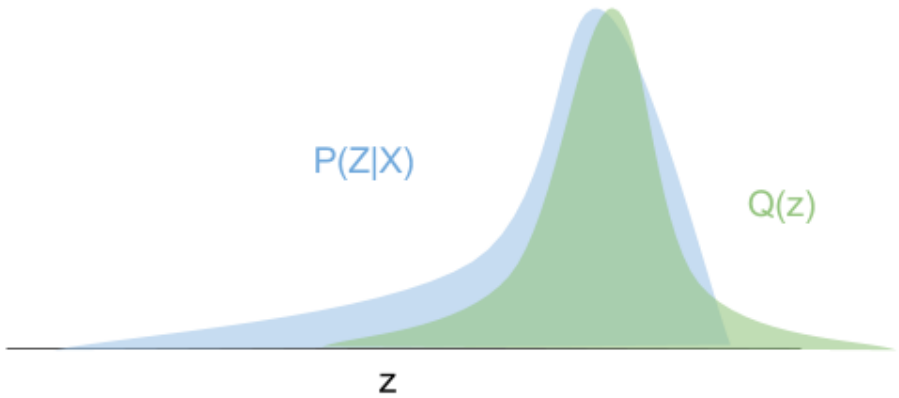


Figure: Approximating true posterior $P(Z|X)$ by Gaussian distribution $Q(Z)$

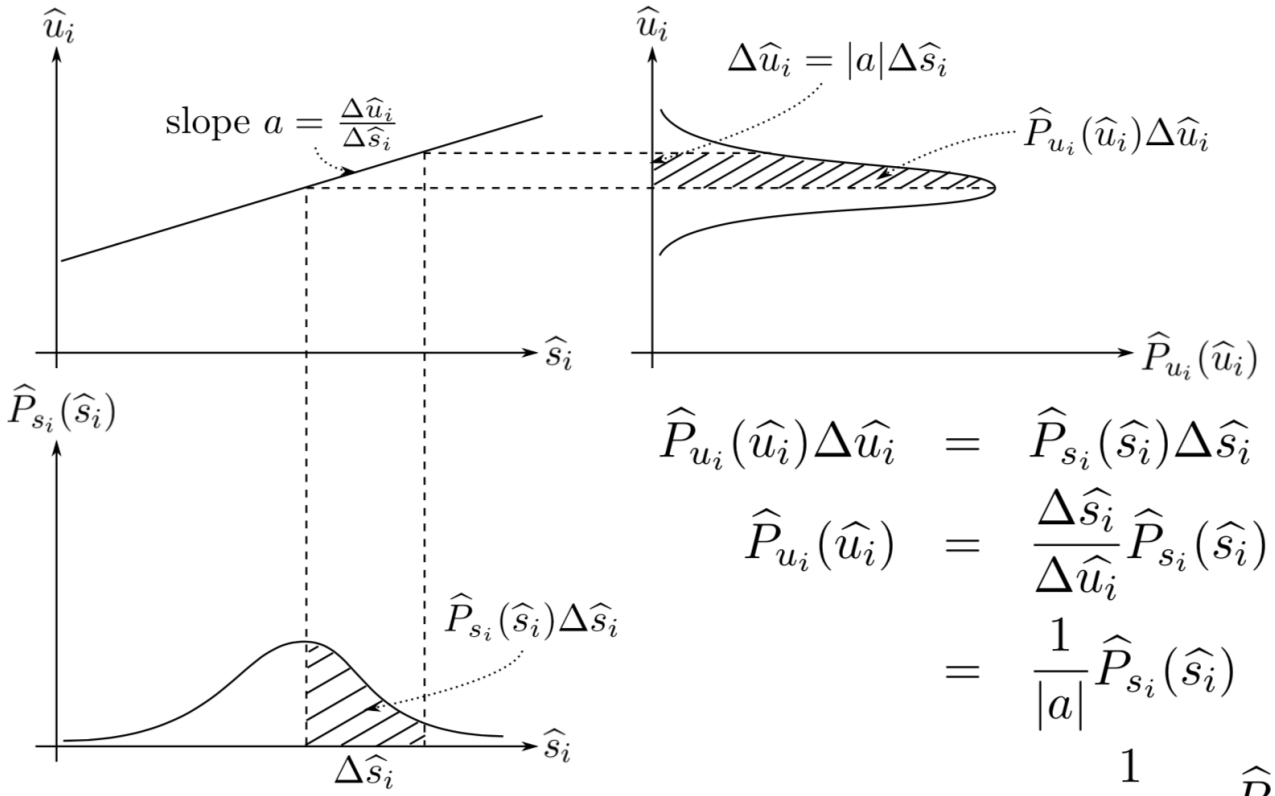
Normalizing Flows

An ideal family of variational distributions $q_\phi(\mathbf{z}|\mathbf{x})$ is the one that highly flexible, preferably flexible enough to obtain the true posterior in an asymptotic regime.

Definition:

A normalizing flow describes the transformation from a probability density through a sequence of *invertible*, *smooth* mappings to another one.

The Basic Rule for Transforming a Random Variable



$$
\begin{aligned}
\widehat{P}_{u_i}(\widehat{u}_i)\Delta\widehat{u}_i &= \widehat{P}_{s_i}(\widehat{s}_i)\Delta\widehat{s}_i \\
\widehat{P}_{u_i}(\widehat{u}_i) &= \frac{\Delta\widehat{s}_i}{\Delta\widehat{u}_i}\widehat{P}_{s_i}(\widehat{s}_i) \\
&= \frac{1}{|a|}\widehat{P}_{s_i}(\widehat{s}_i) \\
&= \frac{1}{\left|d\widehat{u}_i/d\widehat{s}_i\right|}\widehat{P}_{s_i}(\widehat{s}_i) \\
&= \left|\frac{d\widehat{s}_i}{d\widehat{u}_i}\right|\widehat{P}_{s_i}(\widehat{s}_i)
\end{aligned}
$$

Figure: Conservation of Probability Densities

Finite Flows

Given:

- a random variable $\mathbf{z}_0$ with distribution q_0
- a chain of K(finite length) transformations f_K

The final random variable $\mathbf{z}_K$, and its density $q_K(\mathbf{z})$ is:

$$\mathbf{z}_K = f_K \circ ... \circ f_2 \circ f_1(\mathbf{z}_0) \tag{1}$$

$$q_K(\mathbf{z}_K) = q_0(\mathbf{z}_0) \prod_{k=1}^{K} |\det \frac{\partial f_k^{-1}}{\partial \mathbf{z}_{k-1}}| = q_0(\mathbf{z}_0) \prod_{k=1}^{K} |\det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}}|^{-1} \tag{2}$$

$$\ln q_K(\mathbf{z}_K) = \ln q_0(\mathbf{z}_0) - \sum_{k=1}^{K} \ln |\det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}}| \tag{3}$$

where (1) is called the **flow**, and (3) is a **normalizing flow**

Effects of Finite Flows

- Expansion: reducing the density in that region while increasing the density outside the region.
- Contraction: increasing the density in its interior while reducing the density outside.

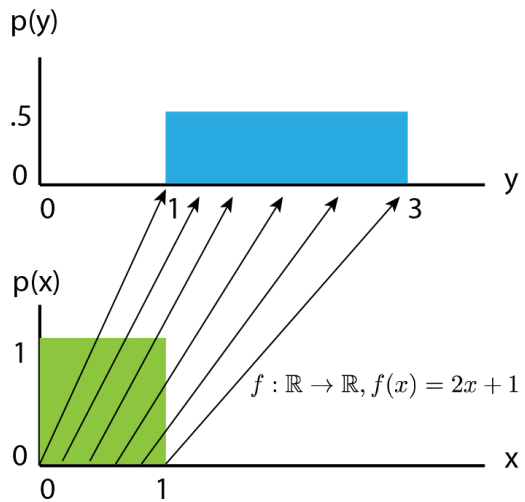


Figure: An example of expansion

Invertible Linear-time Transformations

Computational Cost of the Determinant

An invertible parametric functions $q(\mathbf{z}')$ could be built:

$$q(\mathbf{z}') = q(\mathbf{z})|\det \frac{\partial f^{-1}}{\partial \mathbf{z}'}| = q(\mathbf{z})|\det \frac{\partial f}{\partial \mathbf{z}}|^{-1} \tag{4}$$

However, it's computational consumption:

- The Jacobian determinant $\rightsquigarrow \mathcal{O}(LD^3)$
- The gradients of the Jacobian determinant $\rightsquigarrow \mathcal{O}(LD^3)$

In the ideal case, the computations of the determinant, and even the Jacobian matrix are trivial.

Planar Flows

A family of transformations of the form:

$$f(\mathbf{z}) = \mathbf{z} + \mathbf{u} \cdot h(\mathbf{w}^T \mathbf{z} + b)$$

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

where: $h(\cdot) = \tanh(\cdot)$, $\psi(\cdot) = h'(\mathbf{w}^T \mathbf{z} + b)\mathbf{w}$

The log-density $\log q_K(\mathbf{z})$ is implicitly given by:

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

Planar Flows have a parameter set: $\lambda = \{\mathbf{w} \in \mathbb{R}^D, \mathbf{u} \in \mathbb{R}^D, b \in \mathbb{R}\}$

Radial Flows

A family of transformations that modify an initial density q_0 around a reference point $\mathbf{z}_0$:

$$f(\mathbf{z}) = \mathbf{z} + \beta \cdot h(\alpha, r)(\mathbf{z} - \mathbf{z}_0)$$

$$|det \frac{\partial f}{\partial \mathbf{z}}| = [1 + \beta h(\alpha, r)]^{d-1}[1 + \beta h(\alpha, r) + \beta h'(\alpha, r)r]$$

where: $r = \|\mathbf{z} - \mathbf{z}_0\|$, $h(\alpha, r) = 1/(\alpha + r)$, $h'(\alpha, r) = -1/(\alpha + r)^2$

The log-density $\log q_K(\mathbf{z})$ is:

$$\ln q_K(\mathbf{z}_K) = \ln q_0(\mathbf{z}) - \sum_{k=1}^{K} \ln |\det \frac{\partial f}{\partial \mathbf{z}_{k-1}}|$$

Radial flows have a parameter set: $\lambda = \{\mathbf{z}_0 \in \mathbb{R}^D, \alpha \in \mathbb{R}^+, \beta \in \mathbb{R}\}$

Effects of Normalizing Flows

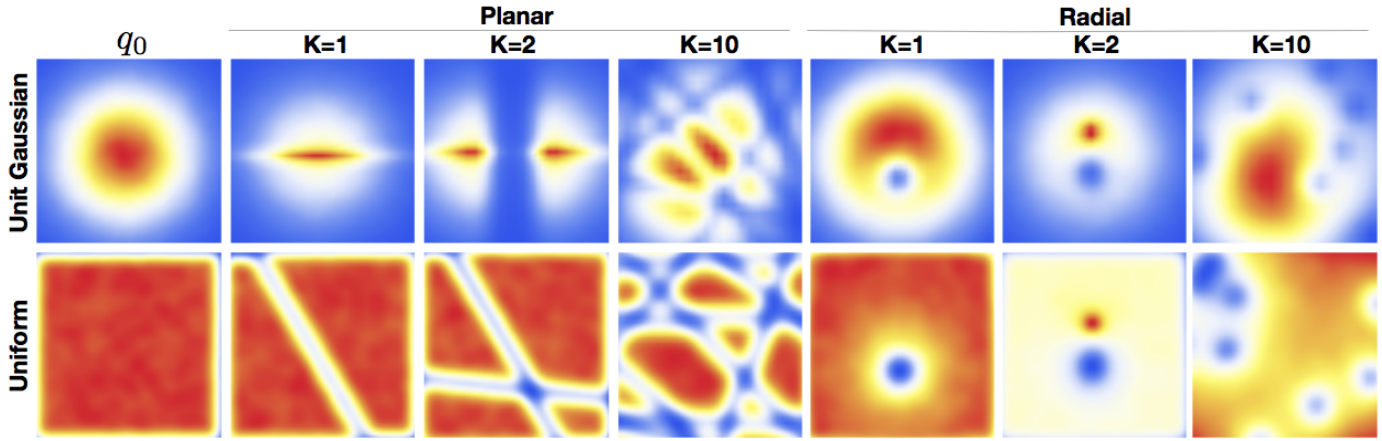


Figure: Effects of Normalizing Flows on two Distributions

Another Examples of Planar Flows

Test Energy Functions

Given the specific formulas of the following four test energy functions with form $p(\mathbf{z}) \sim \exp(-U(\mathbf{z}))$:

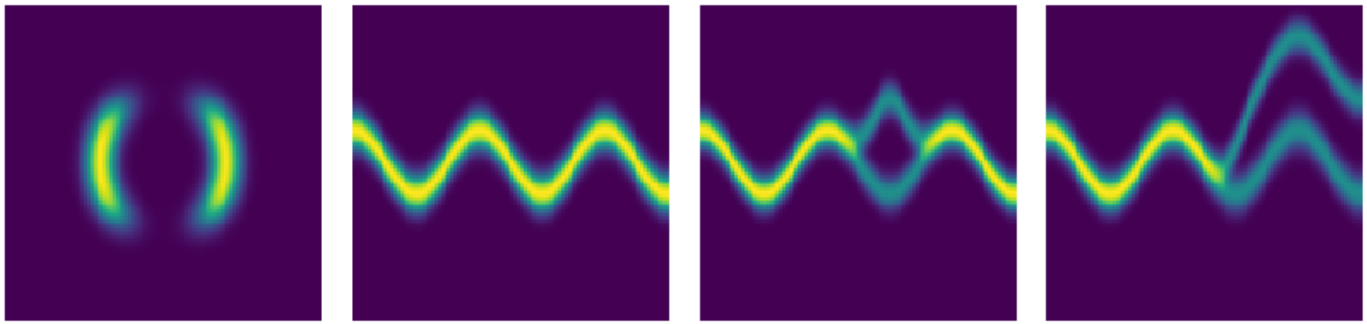
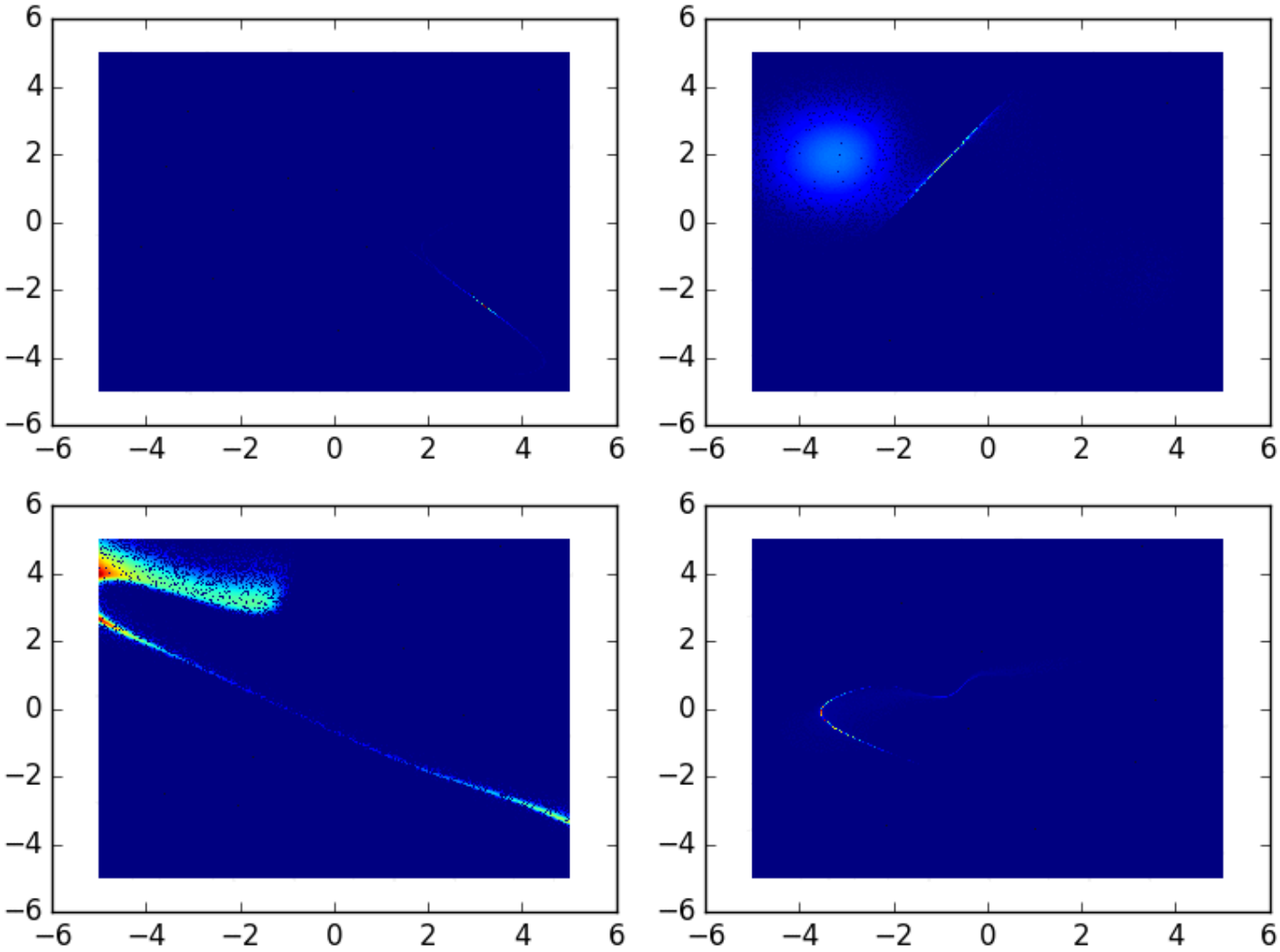


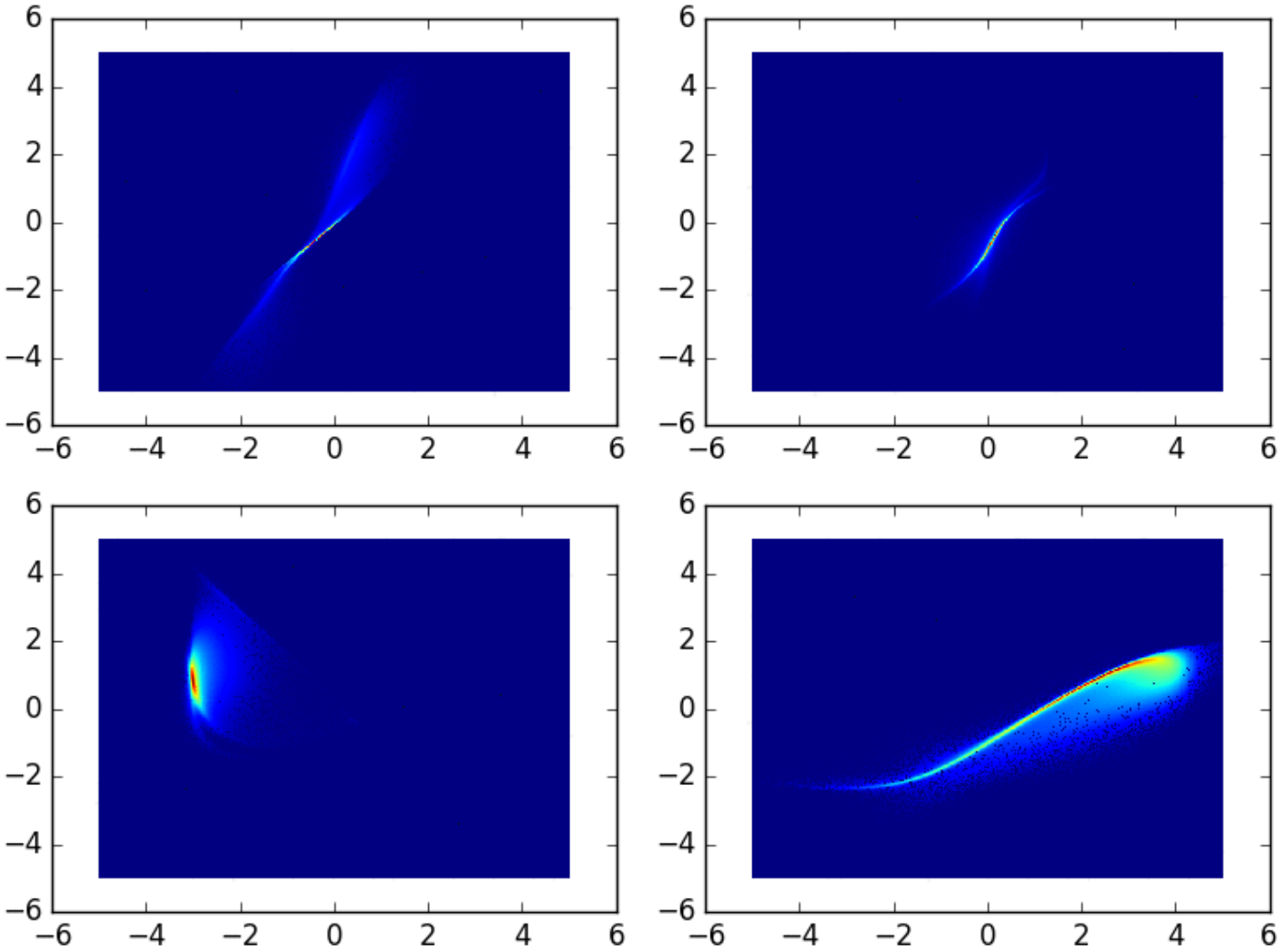
Figure: True Test Energy Functions

Then, utilizing a set of planar flows $f(\cdot)$ to get $\mathbf{z}_0 \sim \mathcal{N}(0, \mathcal{I}) \overset{f_K \circ \ldots \circ f_1(\mathbf{z})}{\longrightarrow} p(\mathbf{z})$

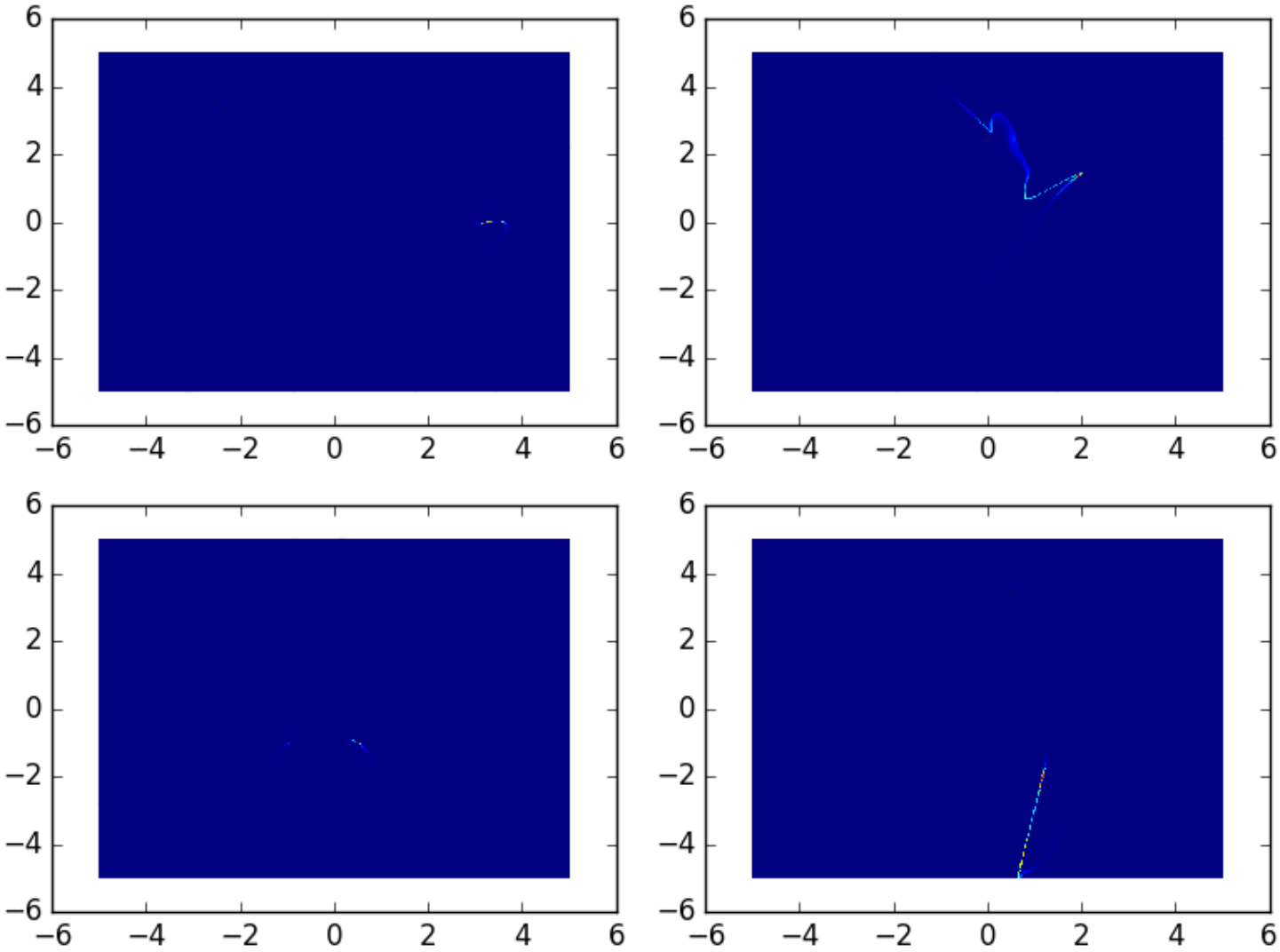
Power of Planar Flows with K=4



Power of Planar Flows with K=8



Power of Planar Flows with K=32



Variational Auto-Encoder(VAE)

The Evidence Lower Bound(ELBO)

Consider a general probabilistic model with observations $\mathbf{x}$, latent variables $\mathbf{z}$, and model parameters $\boldsymbol{\theta}$.

ELBO:

$$\begin{aligned}\log p_\theta(\mathbf{x}) &= \log \int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \\ &= \log \int \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}\end{aligned}$$

With the help of Jensen's Inequality: $\log(\int \mathbf{x} d\mathbf{x}) \geq \int q \log \frac{\mathbf{x}}{q} d\mathbf{x}$

$$\log p_\theta(\mathbf{x}) \geq -\mathbb{D}_{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})] + \mathbb{E}_q[\log p_\theta(\mathbf{x}|\mathbf{z})] = -\mathcal{F}(\mathbf{x})$$

Framework of VAE

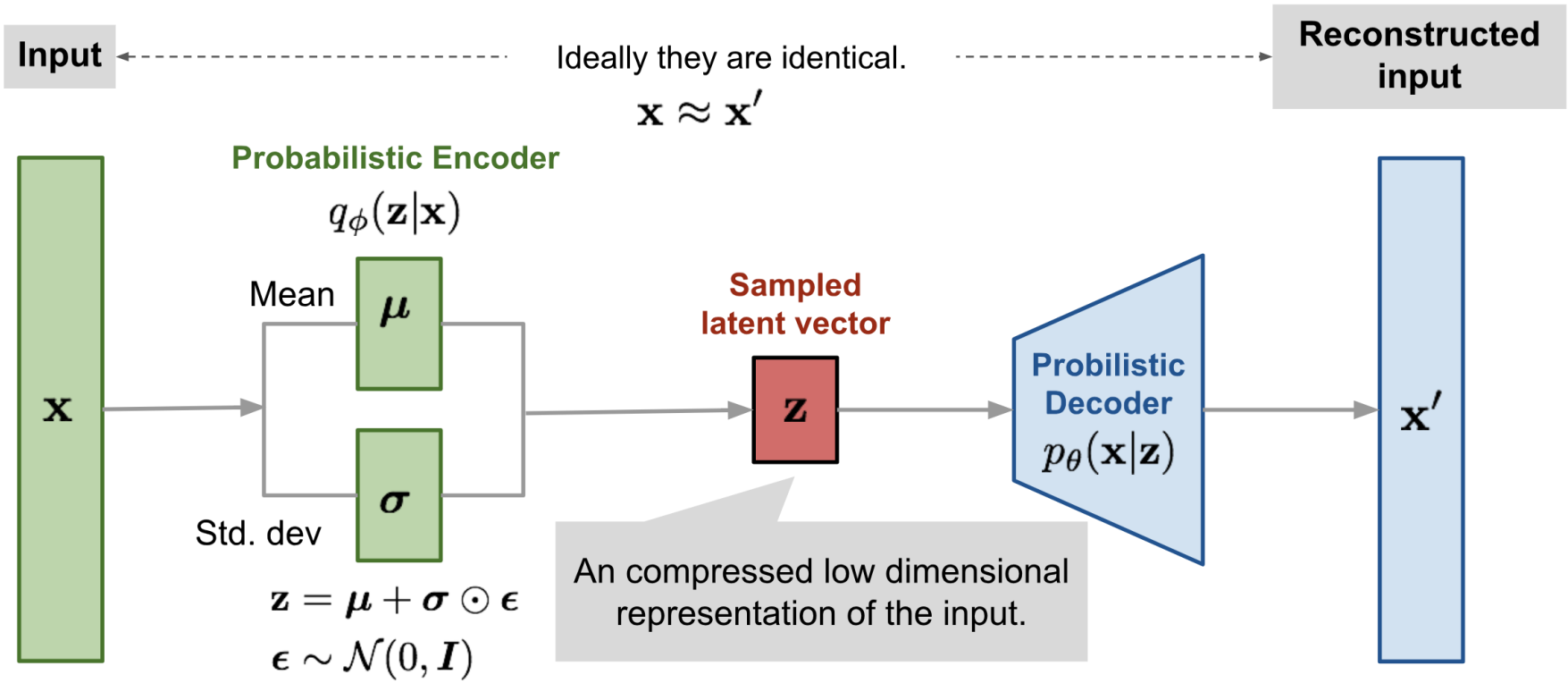


Figure: Inference and Generative Models

VAE with Normalizing Flows

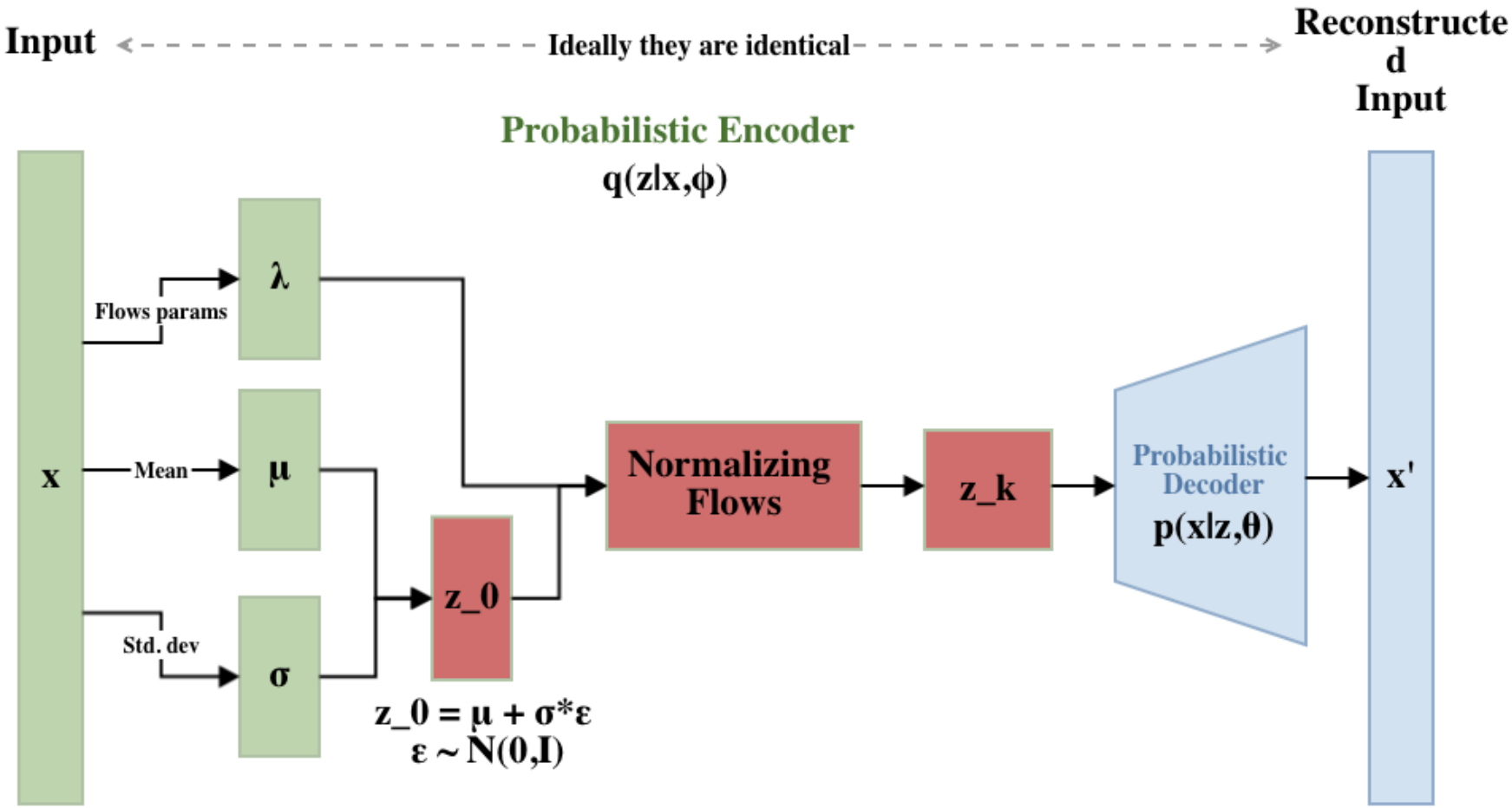


Figure: Flow Chart: Normalizing Flows with Encoder and Decoder

The Evidence Lower Bound(ELBO)

$$-\mathcal{F}(\mathbf{x}) = -\mathbb{D}_{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})] + \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z})]$$

If we parameterize the approximate posterior distribution with a planar flow of length K, then, $q_\phi(\mathbf{z}|\mathbf{x}) := q_K(\mathbf{z}_K)$, and we have the prior knowledge about $\mathbf{z}$, that is, $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$

$$\begin{aligned}
-\mathcal{F}(\mathbf{x}) &= -\mathbb{E}_{q_0(\mathbf{z}_0)}[\log q_0(\mathbf{z}_0) - \sum_{k=1}^{K} \ln|1 + \mathbf{u}_k^T \psi_k(\mathbf{z}_{k-1})| \\
&\quad - \log p(\mathbf{z}) - \log p_\theta(\mathbf{x}|\mathbf{z}_K)] \\
&= -\mathbb{D}_{KL}[q_0(\mathbf{z}_0)||p(\mathbf{z})] - \mathbb{E}_{q_0}[\log p_\theta(\mathbf{x}|\mathbf{z}_K)] \\
&\quad - \mathbb{E}_{q_0}[\sum_{k=1}^{K} \ln|1 + \mathbf{u}_k^T \psi_k(\mathbf{z}_{k-1})|]
\end{aligned}$$

MLPs as probabilistic encoders and decoders

choice 1: Bernoulli MLP as decoder

Let $p_\theta(\mathbf{x}|\mathbf{z})$ be a Bernoulli distribution which is computed from latent variable $\mathbf{z}$:

$$\log p_\theta(\mathbf{x}|\mathbf{z}) = \sum_{i=1}^{n} x_i \log \hat{x}_i + (1 - x_i) \log(1 - \hat{x}_i)$$

$$\hat{\mathbf{x}} = f_\sigma(\mathbf{W}_2 \tanh(\mathbf{W}_1 \mathbf{z}_K + \mathbf{b}_1) + \mathbf{b}_2)$$

where $f_\sigma(\cdot)$ is sigmoid transfer function, and $\boldsymbol{\theta} = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2\}$ are parameters of Decoder(MLP).

MLPs as probabilistic encoders and decoders

choice 2: Gaussian MLP as a encoder or decoder

Let $p_\theta(\mathbf{x}|\mathbf{z})$ be a Gaussian distribution with diagonal covariance structure:

$$\log p_\theta(\mathbf{x}|\mathbf{z}) = \log \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are created by MLP with parameter $\boldsymbol{\theta}$ when used as a decoder. Note that when this is used as a encoder $q_\phi(\mathbf{z}|\mathbf{x})$, then $\mathbf{x}$ and $\mathbf{z}$ should be swapped, and the parameter should be modified with $\boldsymbol{\phi}$

Q&A

Thank you for your attention!