VAE with a VampPrior

Jakub Tomczak

Tübingen, 22nd of March 2018

Generative modeling

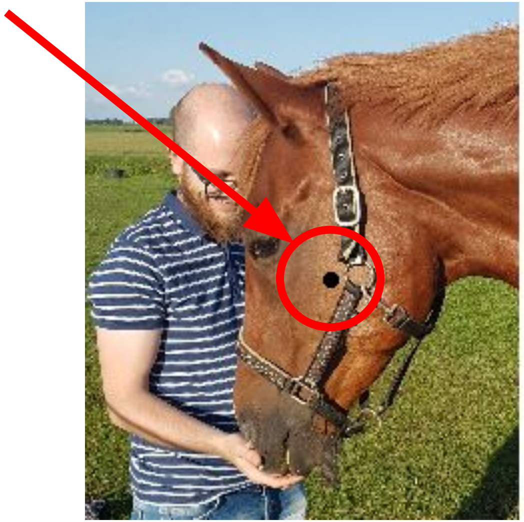
Modeling in a high-dimensional space is difficult.

Generative modeling

Modeling in a high-dimensional space is difficult.

Generative modeling

Modeling in a high-dimensional space is difficult.



Generative modeling

Modeling in a high-dimensional space is difficult.

→modeling all dependencies among pixels.

$$p(x) = \prod_{d=1}^{c} \psi_c(x_c)$$

Generative modeling

Modeling in a high-dimensional space is difficult.

→modeling all dependencies among pixels.

$$p(x) = \prod_{d=1}^{c} \psi_c(x_c)$$

Very inefficient!

Generative modeling

Modeling in a high-dimensional space is difficult.

→modeling all dependencies among pixels.

$$p(x) = \prod_{d=1}^{c} \psi_c(x_c)$$

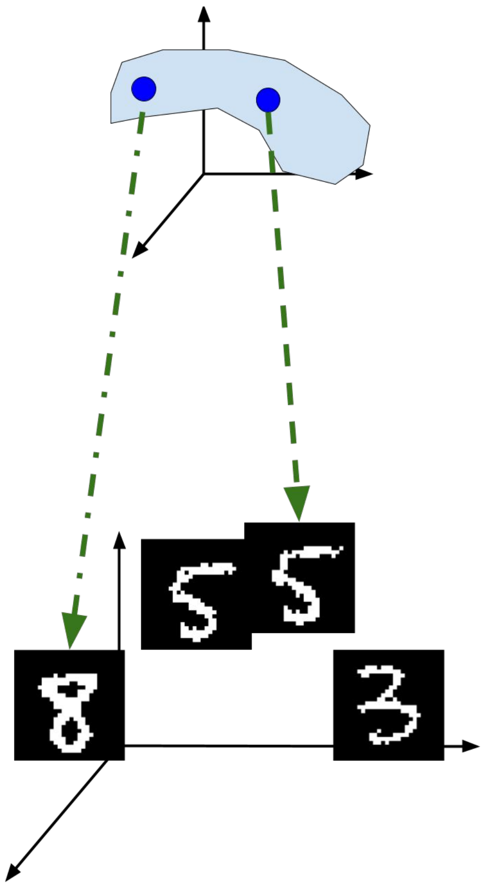
Very inefficient!

A possible **solution**? →**Models with latent variables**

Latent Variable Models

Latent variable model:

$$p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z}) \; p_\lambda(\mathbf{z}) \; \mathrm{d}\mathbf{z}$$

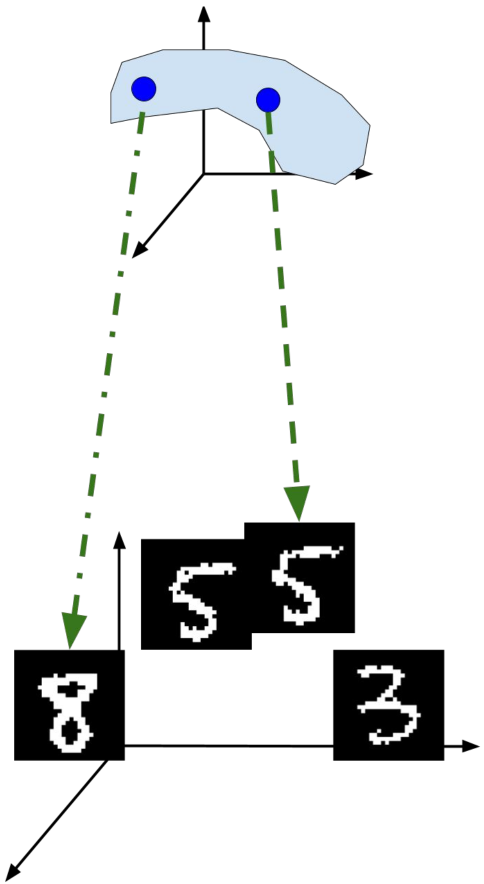


Latent Variable Models

Latent variable model:

$$p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

First sample $\mathbf{z}$.
Second sample $\mathbf{x}$ for given $\mathbf{z}$.

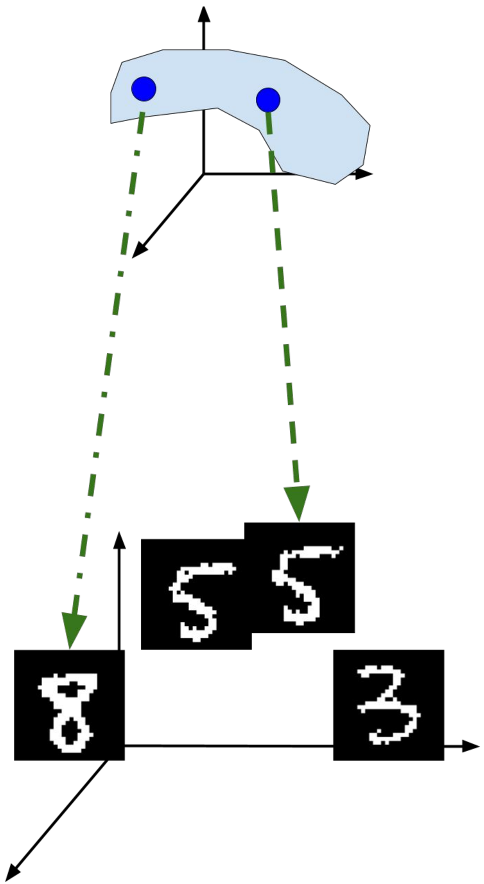


Latent Variable Models

Latent variable model:

$$p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

First sample $\mathbf{z}$.
Second sample $\mathbf{x}$ for given $\mathbf{z}$.



Latent Variable Models

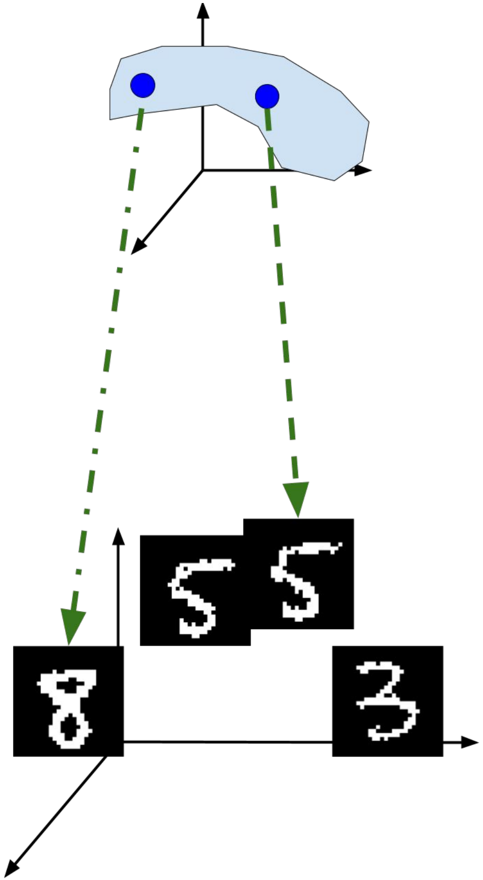
Latent variable model:

$$p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

How to calculate this integral?

If $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{W}\mathbf{z} + \mathbf{b}, \Psi)$ and $p_\lambda(\mathbf{z}) = \mathcal{N}(\mu_0, \Sigma_0)$, then we get **Factor Analysis.**

What if we take a **non-linear transformation** of **z**?
→**an infinite mixture of Gaussians**



Latent Variable Models

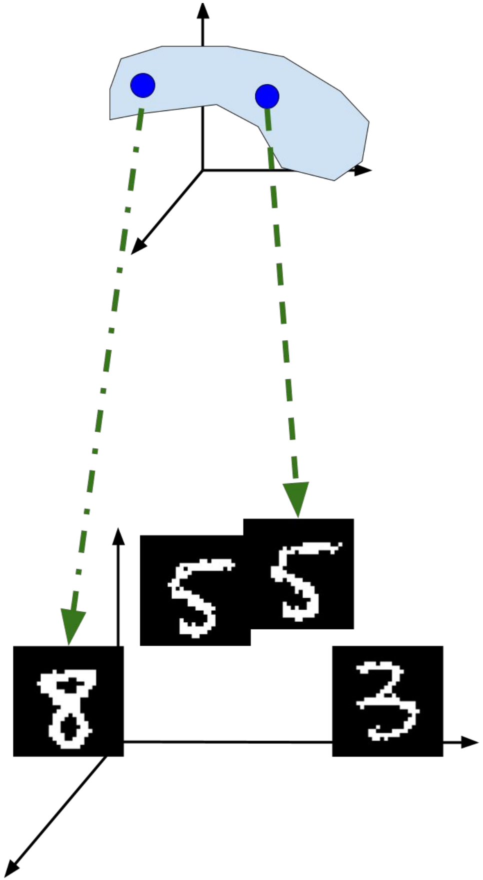
Latent variable model:

$$p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

If $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{W}\mathbf{z} + \mathbf{b}, \Psi)$ and $p_\lambda(\mathbf{z}) = \mathcal{N}(\mu_0, \Sigma_0)$, then we get **Factor Analysis**.

What if we take a **non-linear transformation** of **z**?

→**an infinite mixture of Gaussians**



Latent Variable Models

Latent variable model:

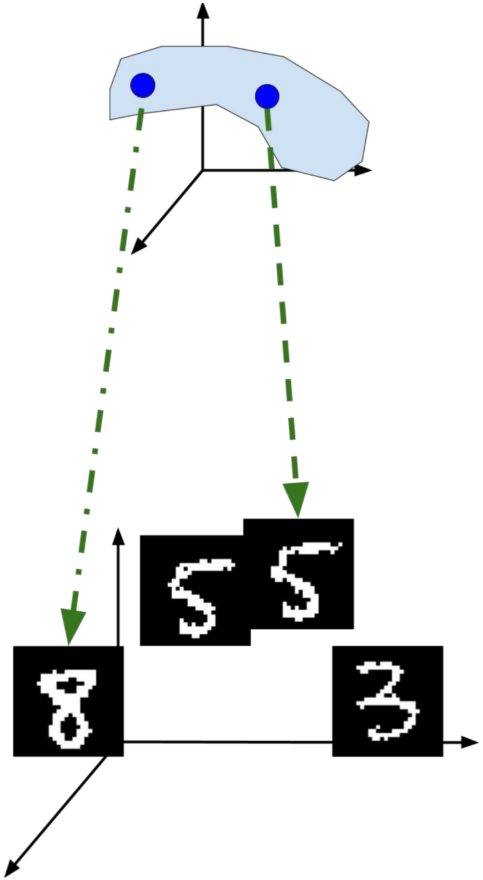
$$p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

If $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{W}\mathbf{z} + \mathbf{b}, \Psi)$ and $p_\lambda(\mathbf{z}) = \mathcal{N}(\mu_0, \Sigma_0)$, then we get **Factor Analysis**.

Convenient but **limiting**!

What if we take a **non-linear transformation** of **z**?

→**an infinite mixture of Gaussians**



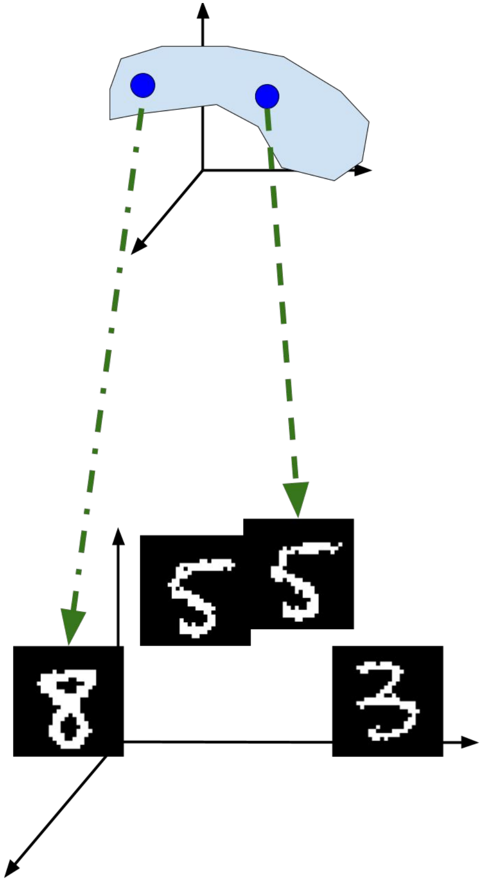
Latent Variable Models

Latent variable model:

$$p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})\; p_\lambda(\mathbf{z})\; \mathrm{d}\mathbf{z}$$

If $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{W}\mathbf{z} + \mathbf{b}, \Psi)$ and $p_\lambda(\mathbf{z}) = \mathcal{N}(\mu_0, \Sigma_0)$, then we get **Factor Analysis**.

What if we take a **non-linear transformation** of **z**?
→**an infinite mixture of Gaussians**



Latent Variable Models

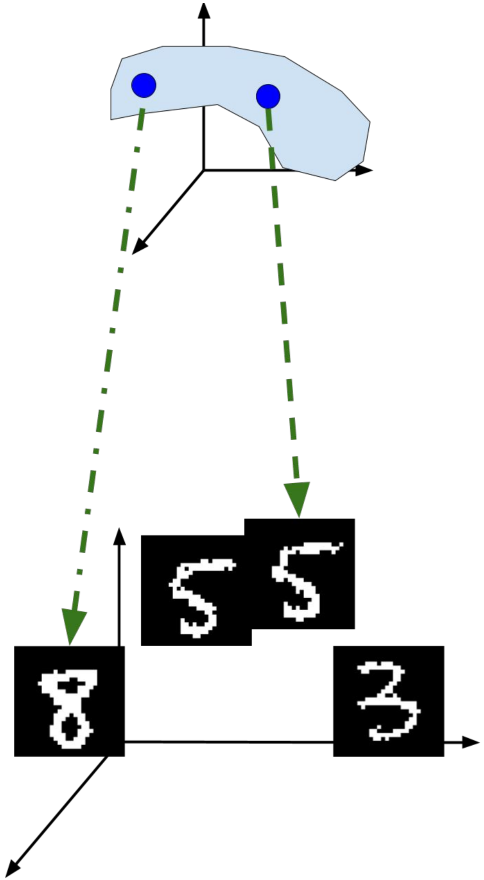
Latent variable model:

$$p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

If $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{W}\mathbf{z} + \mathbf{b}, \Psi)$ and $p_\lambda(\mathbf{z}) = \mathcal{N}(\mu_0, \Sigma_0)$,
then we get **Factor Analysis**.

What if we take a **non-linear transformation** of **z**?
→**an infinite mixture of Gaussians**

Neural network



MacKay, D. J., & Gibbs, M. N. (1999). Density networks. Statistics and neural networks: advances at the interface. Oxford University Press, Oxford, 129-144.

Latent Variable Models

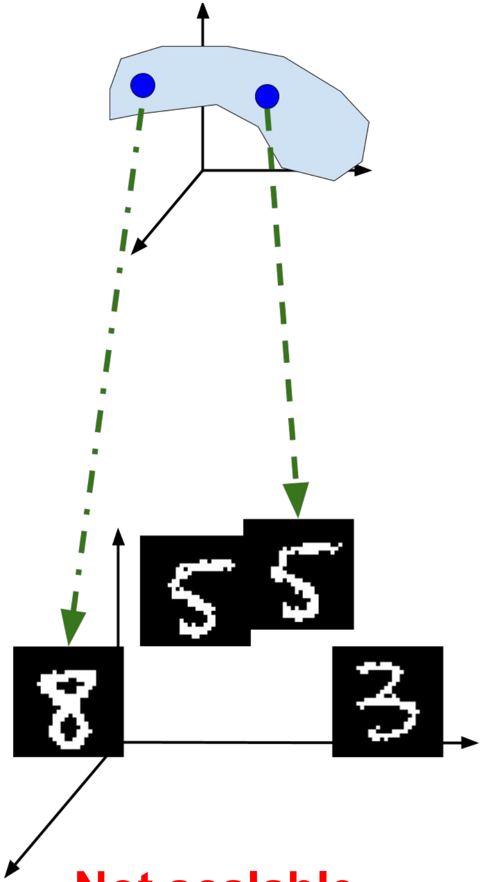
Latent variable model:

$$p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

If $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{W}\mathbf{z} + \mathbf{b}, \Psi)$ and $p_\lambda(\mathbf{z}) = \mathcal{N}(\mu_0, \Sigma_0)$,
then we get **Factor Analysis**.

What if we take a **non-linear transformation** of **z**?
→**an infinite mixture of Gaussians**

Neural network



Not scalable...

MacKay, D. J., & Gibbs, M. N. (1999). Density networks. Statistics and neural networks: advances at the interface. Oxford University Press, Oxford, 129-144.

Variational inference for Latent Variable Models

$$
\begin{aligned}
\log p(\mathbf{x}) &= \log \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&= \log \int \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\ p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&\geq \int q_\phi(\mathbf{z}|\mathbf{x})\ \log \frac{p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\ \mathrm{d}\mathbf{z} \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p_\lambda(\mathbf{z})]
\end{aligned}
$$

Variational inference for Latent Variable Models

Variational posterior

$$
\begin{aligned}
\log p(\mathbf{x}) &= \log \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&= \log \int \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\ p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&\geq \int q_\phi(\mathbf{z}|\mathbf{x})\ \log \frac{p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\ \mathrm{d}\mathbf{z} \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p_\lambda(\mathbf{z})]
\end{aligned}
$$

Variational inference for Latent Variable Models

$$
\begin{aligned}
\log p(\mathbf{x}) &= \log \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&= \log \int \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\ p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&\geq \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\ \mathrm{d}\mathbf{z} \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p_\lambda(\mathbf{z})]
\end{aligned}
$$

Jensen’s inequality

Variational inference for Latent Variable Models

$$
\begin{aligned}
\log p(\mathbf{x}) &= \log \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&= \log \int \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\ p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&\geq \int q_\phi(\mathbf{z}|\mathbf{x})\ \log \frac{p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\ \mathrm{d}\mathbf{z} \\
&= \underbrace{\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{Reconstruction error}} - \underbrace{\mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p_\lambda(\mathbf{z})]}_{\text{Regularization}}
\end{aligned}
$$

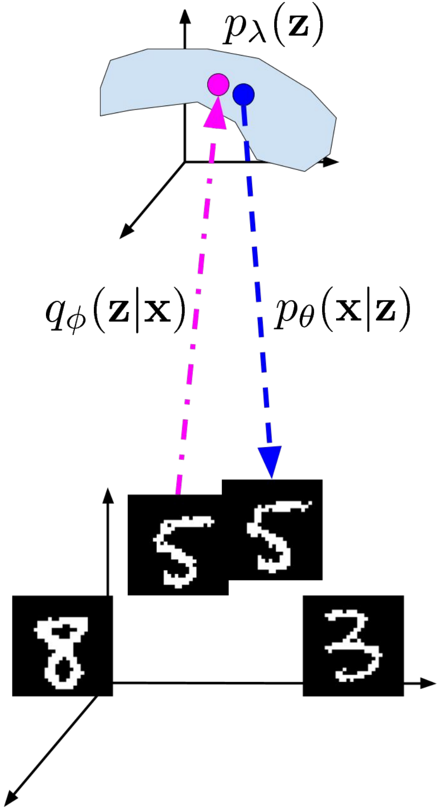
Variational Auto-Encoder

Let us assume the following distributions:

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\big(\mathbf{z}|\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2)\big)$$ **encoder**

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathrm{Bern}\big(\theta(\mathbf{z})\big)$$ **decoder**

$$p_\lambda(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$$ **prior**



Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Variational Auto-Encoder

Let us assume the following distributions:

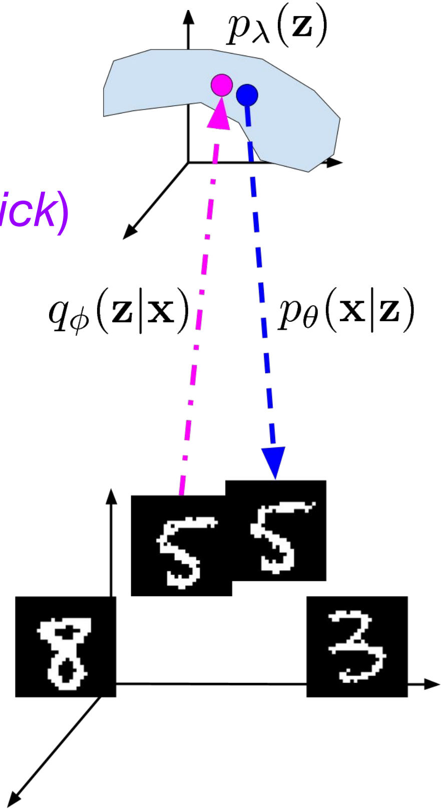
$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\big(\mathbf{z}|\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2)\big)$$ **encoder**

sampling is easy
(reparameterization trick)

$$\boxed{\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \varepsilon}$$

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathrm{Bern}\big(\theta(\mathbf{z})\big)$$ **decoder**

$$p_\lambda(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$$ **prior**



Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Variational Auto-Encoder

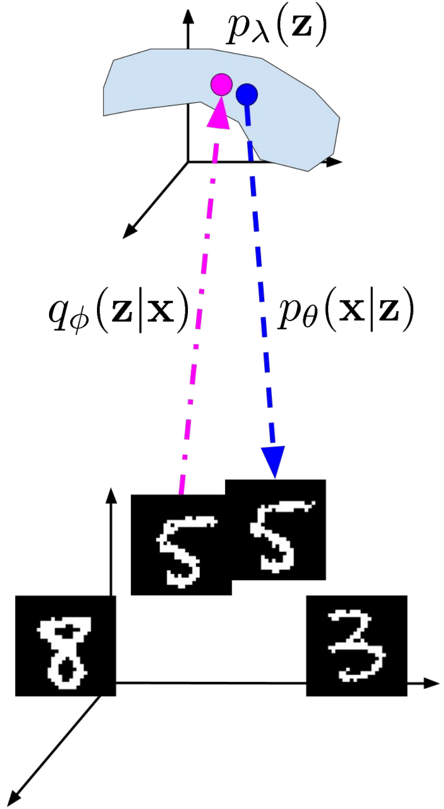
Let us assume the following distributions:

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\big(\mathbf{z}|\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2)\big)$$ **encoder**

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathrm{Bern}\big(\theta(\mathbf{z})\big)$$ **decoder**

$$p_\lambda(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$$ **prior**

or any other distribution



Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Variational Auto-Encoder

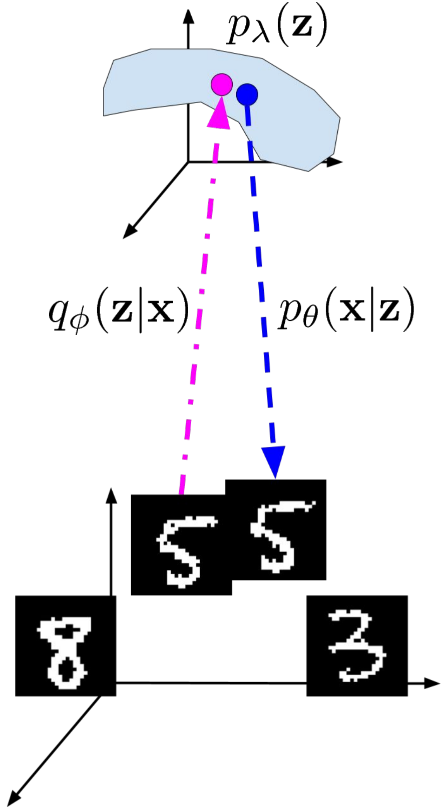
Let us assume the following distributions:

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\big(\mathbf{z}|\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2)\big)$$ **encoder**

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathrm{Bern}\big(\theta(\mathbf{z})\big)$$ **decoder**

$$p_\lambda(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$$ **prior**

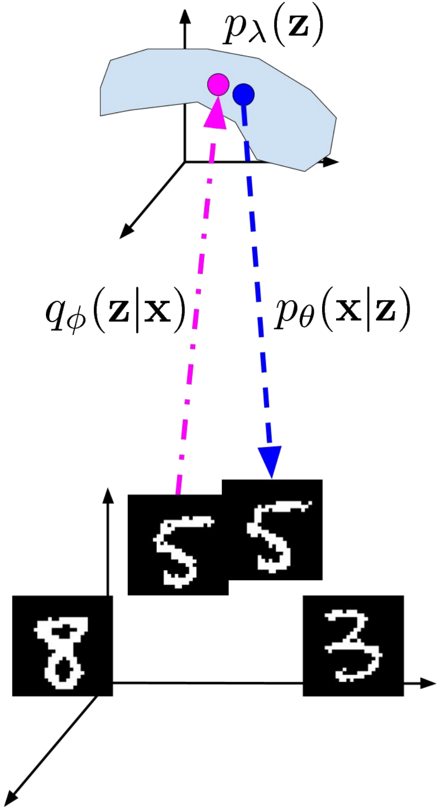
simplest case



Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Variational Auto-Encoder

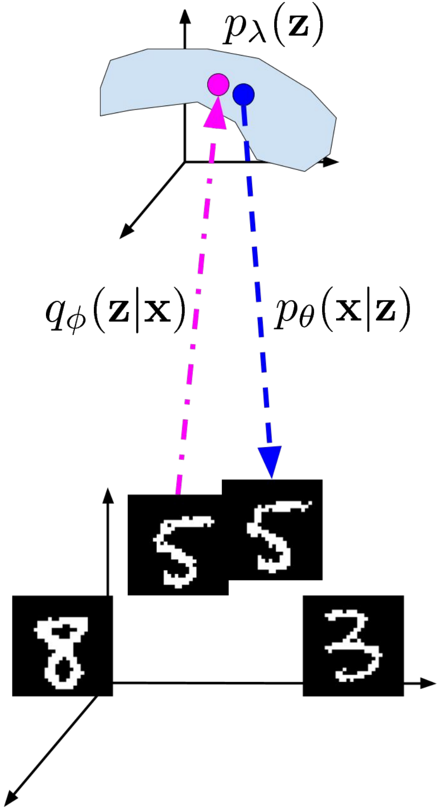
$$q_\phi(\mathbf{z}|\mathbf{x}) \propto p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})$$



Variational Auto-Encoder

$$q_\phi(\mathbf{z}|\mathbf{x}) \propto p_\theta(\mathbf{x}|\mathbf{z})\, p_\lambda(\mathbf{z})$$

Fully-connected
ConvNets
PixelCNN

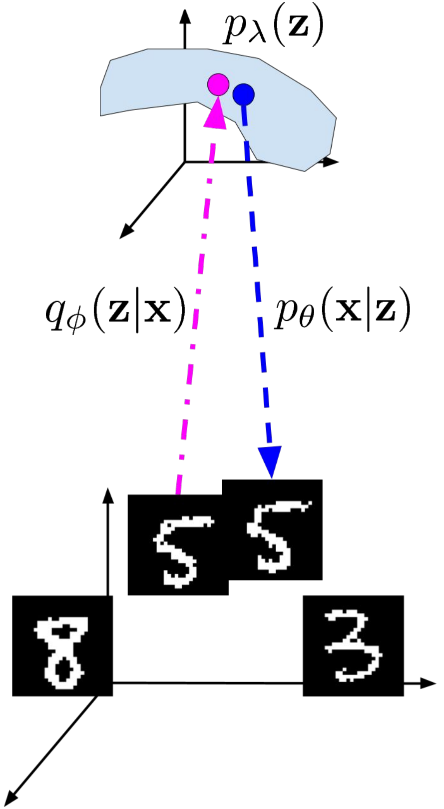


Variational Auto-Encoder

$$q_\phi(\mathbf{z}|\mathbf{x}) \propto p_\theta(\mathbf{x}|\mathbf{z})\, p_\lambda(\mathbf{z})$$

Normalizing flows
Volume-preserving flows

Fully-connected
ConvNets
PixelCNN



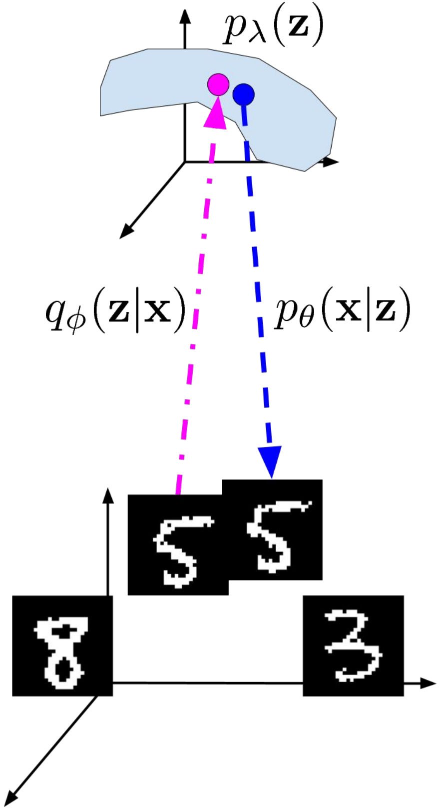
Variational Auto-Encoder

$$q_\phi(\mathbf{z}|\mathbf{x}) \propto p_\theta(\mathbf{x}|\mathbf{z}) p_\lambda(\mathbf{z})$$

Normalizing flows
Volume-preserving flows

Fully-connected
ConvNets
PixelCNN

Autoregressive Prior
Objective Prior
Stick-Breaking Prior
VampPrior



Variational Auto-Encoder

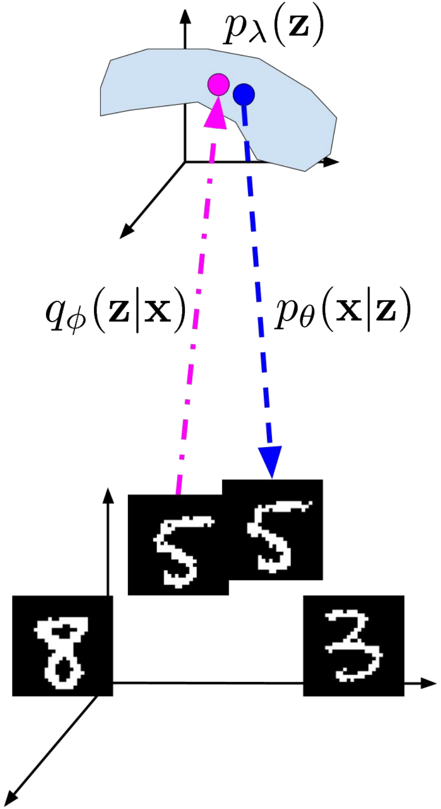
$$q_\phi(\mathbf{z}|\mathbf{x}) \propto p_\theta(\mathbf{x}|\mathbf{z})\, p_\lambda(\mathbf{z})$$

Normalizing flows
Volume-preserving flows

Fully-connected
ConvNets
PixelCNN

Autoregressive Prior
Objective Prior
Stick-Breaking Prior
VampPrior

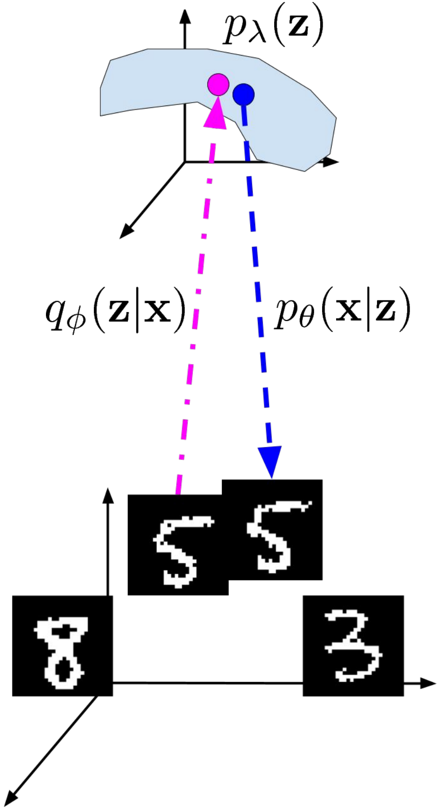
Importance Weighted AE
Renyi Divergence
Stein Divergence



Variational Auto-Encoder

$$q_\phi(\mathbf{z}|\mathbf{x}) \propto p_\theta(\mathbf{x}|\mathbf{z})\, p_\lambda(\mathbf{z})$$

Autoregressive Prior
Objective Prior
Stick-Breaking Prior
VampPrior



New Prior

- Let's re-write the ELBO:

$$
\begin{aligned}
\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[\,\ln p(\mathbf{x})\right] \geq & \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}[\ln p_{\theta}(\mathbf{x}|\mathbf{z})]\right] + \\
& + \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[\mathbb{H}[q_{\phi}(\mathbf{z}|\mathbf{x})]\right] + \\
& - \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[-\ln p_{\lambda}(\mathbf{z})]
\end{aligned}
$$

New Prior

- Let's re-write the ELBO:

$$\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[\ln p(\mathbf{x})\right] \geq \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})]\right] +$$
$$+ \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[\mathbb{H}[q_\phi(\mathbf{z}|\mathbf{x})]\right] +$$
$$- \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[-\ln p_\lambda(\mathbf{z})]$$

Empirical distribution

New Prior

- Let's re-write the ELBO:

Reconstruction error

$$\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\ln p(\mathbf{x})\big] \geq \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}[\ln p_{\theta}(\mathbf{x}|\mathbf{z})]\big] +$$

$$+ \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\mathbb{H}[q_{\phi}(\mathbf{z}|\mathbf{x})]\big] +$$

$$- \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[-\ln p_{\lambda}(\mathbf{z})]$$

New Prior

- Let's re-write the ELBO:

$$\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\ln p(\mathbf{x})\big] \geq \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})]\big] +$$

$$+\,\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\mathbb{H}[q_\phi(\mathbf{z}|\mathbf{x})]\big] +$$

Encoder's entropy

$$-\,\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}\big[-\ln p_\lambda(\mathbf{z})\big]$$

New Prior

- Let's re-write the ELBO:

$$\begin{aligned}\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\ln p(\mathbf{x})\big] \geq & \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})]\big] + \\ & + \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\mathbb{H}[q_\phi(\mathbf{z}|\mathbf{x})]\big] + \\ & - \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[-\ln p_\lambda(\mathbf{z})]\end{aligned}$$

Cross Entropy

New Prior

- Let's re-write the ELBO:

$$\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\ln p(\mathbf{x})\big] \geq \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\Big[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})]\Big] +$$
$$+\ \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\Big[\mathbb{H}[q_\phi(\mathbf{z}|\mathbf{x})]\Big] +$$
$$-\ \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}\big[-\ln p_\lambda(\mathbf{z})\big]$$

Aggregated posterior

$$q(\mathbf{z}) = \mathbb{E}_{q(\mathbf{x})}[q_\phi(\mathbf{z}|\mathbf{x})]$$
$$= \frac{1}{N}\sum_{n=1}^{N} q_\phi(\mathbf{z}|\mathbf{x}_n)$$

New Prior

- Let’s re-write the ELBO:

max. $$\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\ln p(\mathbf{x})\big] \geq \boxed{\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})]\big]} +$$ **Variance**

$$+\boxed{\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\mathbb{H}[q_\phi(\mathbf{z}|\mathbf{x})]\big]} +$$ **Variance**

$$-\ \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}\big[-\ln p_\lambda(\mathbf{z})\big]$$

New Prior

- Let's re-write the ELBO:

$$\text{max.}\ \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\ln p(\mathbf{x})\big] \geq \boxed{\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})]\big]} + \quad \textbf{Variance}$$

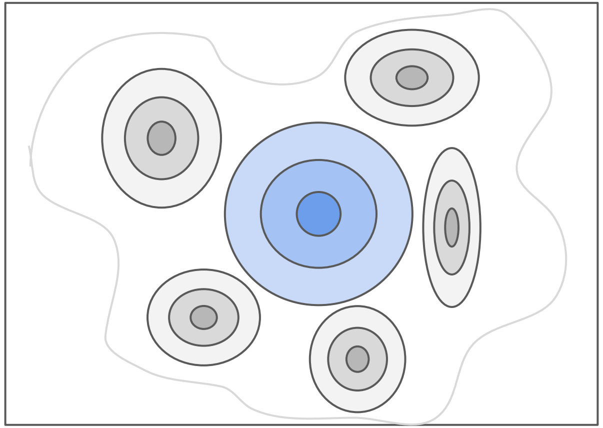
$$+ \boxed{\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\big[\mathbb{H}[q_\phi(\mathbf{z}|\mathbf{x})]\big]} + \quad \textbf{Variance}$$

$$- \boxed{\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[-\ln p_\lambda(\mathbf{z})]}$$

$$\text{min.}\ \boxed{\mathbb{H}[q(\mathbf{z})] + KL[q(\mathbf{z})||p_\lambda(\mathbf{z})]}$$

New Prior

$$\text{min.}\ \mathbb{H}[q(\mathbf{z})] + KL[q(\mathbf{z})||p_\lambda(\mathbf{z})]$$

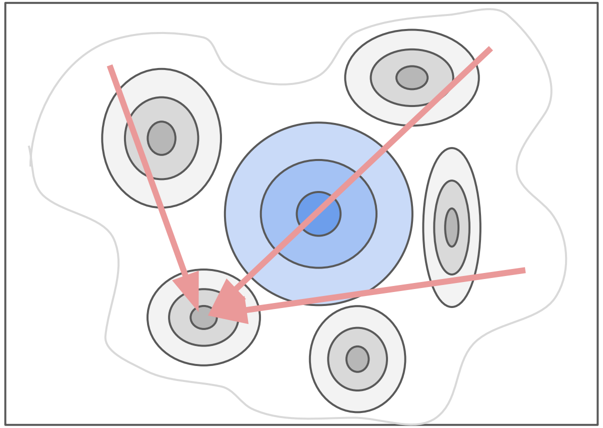


Prior

Aggregated posterior

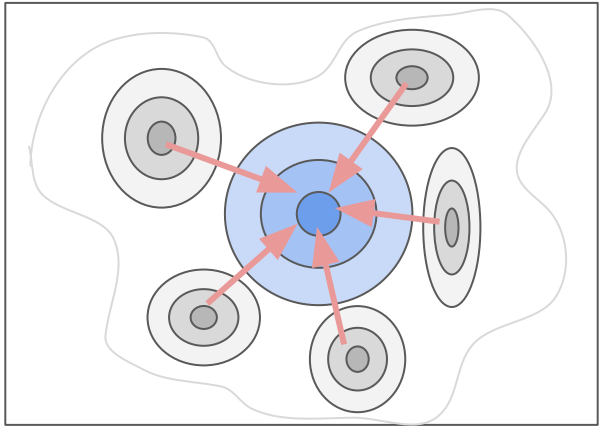
New Prior

$$\text{min. } \mathbb{H}[q(\mathbf{z})] + KL[q(\mathbf{z})||p_\lambda(\mathbf{z})]$$



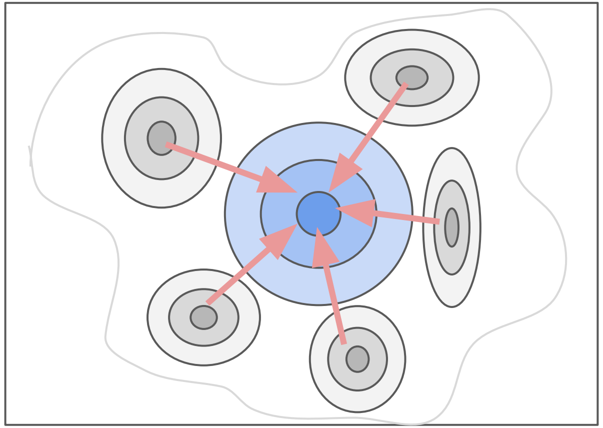
New Prior

$$\text{min. } \mathbb{H}[q(\mathbf{z})] + KL[q(\mathbf{z})||p_\lambda(\mathbf{z})]$$



New Prior

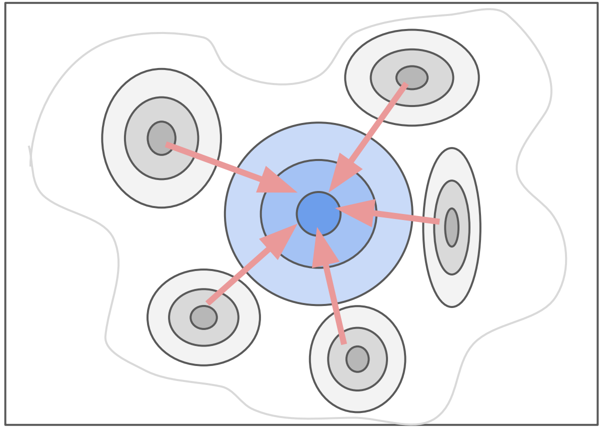
$$\text{min. } \mathbb{H}[q(\mathbf{z})] + KL[q(\mathbf{z})||p_\lambda(\mathbf{z})]$$



Standard prior is too strong and overregularizes the encoder.

New Prior

$$\text{min. } \mathbb{H}[q(\mathbf{z})] + KL[q(\mathbf{z})||p_\lambda(\mathbf{z})]$$



Standard prior is too strong and overregularizes the encoder.

What is the “optimal” prior?

New Prior (**Va**riational **M**ixture of **P**osteriors **Prior**)

- We look for **the optimal prior** using the Lagrange function:

$$\max_{p_\lambda(\mathbf{z})} -\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[-\ln p_\lambda(\mathbf{z})] + \beta\Big(\int p_\lambda(\mathbf{z})\mathrm{d}\mathbf{z} - 1\Big)$$

- The solution is simply **the aggregated posterior**.

- We approximate it using *K* **pseudo-inputs** instead of *N* observations:

$$p_\lambda(\mathbf{z}) = \frac{1}{K}\sum_{k=1}^{K} q_\phi(\mathbf{z}|\mathbf{u}_k)$$

New Prior (Variational Mixture of Posteriors Prior)

- We look for the optimal prior using the Lagrange function:

$$\max_{p_\lambda(\mathbf{z})} -\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[-\ln p_\lambda(\mathbf{z})] + \beta\Big(\int p_\lambda(\mathbf{z})\mathrm{d}\mathbf{z} - 1\Big)$$

- The solution is simply **the aggregated posterior**. $\boxed{p^*_\lambda(\mathbf{z}) = \frac{1}{N}\sum_{n=1}^{N} q_\phi(\mathbf{z}|\mathbf{x}_n)}$

- We approximate it using K **pseudo-inputs** instead of N observations:

$$p_\lambda(\mathbf{z}) = \frac{1}{K}\sum_{k=1}^{K} q_\phi(\mathbf{z}|\mathbf{u}_k)$$

New Prior (Variational Mixture of Posteriors Prior)

- We look for the optimal prior using the Lagrange function:

$$\max_{p_\lambda(\mathbf{z})} -\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[-\ln p_\lambda(\mathbf{z})] + \beta\left(\int p_\lambda(\mathbf{z})\mathrm{d}\mathbf{z} - 1\right)$$

- The solution is simply **the aggregated posterior.** $\boxed{p^*_\lambda(\mathbf{z}) = \frac{1}{N}\sum_{n=1}^{N} q_\phi(\mathbf{z}|\mathbf{x}_n)}$

- We approximate it using K **pseudo-inputs** instead of N observations:

infeasible

$$p_\lambda(\mathbf{z}) = \frac{1}{K}\sum_{k=1}^{K} q_\phi(\mathbf{z}|\mathbf{u}_k)$$

New Prior (**Va**riational **M**ixture of **P**osteriors **Prior**)

- We look for the optimal prior using the Lagrange function:

$$\max_{p_\lambda(\mathbf{z})} -\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[-\ln p_\lambda(\mathbf{z})] + \beta\Big(\int p_\lambda(\mathbf{z})\mathrm{d}\mathbf{z} - 1\Big)$$

- The solution is simply **the aggregated posterior**.

- We approximate it using *K* **pseudo-inputs** instead of *N* observations:

$$p_\lambda(\mathbf{z}) = \frac{1}{K}\sum_{k=1}^{K} q_\phi(\mathbf{z}|\mathbf{u}_k)$$

New Prior (**Va**riational **M**ixture of **P**osteriors **Prior**)

- We look for the optimal prior using the Lagrange function:

$$\max_{p_\lambda(\mathbf{z})} -\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[-\ln p_\lambda(\mathbf{z})] + \beta\Big(\int p_\lambda(\mathbf{z})\mathrm{d}\mathbf{z} - 1\Big)$$

- The solution is simply **the aggregated posterior**.

- We approximate it using K **pseudo-inputs** instead of N observations:

$$p_\lambda(\mathbf{z}) = \frac{1}{K}\sum_{k=1}^{K} q_\phi(\mathbf{z}|\mathbf{u}_k)$$

they are trained from scratch

New Prior (**Va**riational **M**ixture of **P**osteriors **Prior**)

- Is the VampPrior different than the Mixture of Gaussians? $p_{\lambda}(\mathbf{z}) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}\big(\mu_k, \mathrm{diag}(\sigma_k^2)\big)$
- VampPrior: the prior and the posterior must “**cooperate**” during training.

VampPrior

$$\frac{1}{K}\sum_{k=1}^{K}\Big\{\Big(\frac{q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{x})\ \frac{\partial}{\partial\phi_i}q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{u}_k) - q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{u}_k)\ \frac{\partial}{\partial\phi_i}q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{x})}{\frac{1}{K}\sum_{k=1}^{K}q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{u}_k)\ q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{x})}\Big)+$$

$$+\Big(\frac{\big(q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{x})\ \frac{\partial}{\partial\mathbf{z}_{\phi}}q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{u}_k) - q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{u}_k)\ \frac{\partial}{\partial\mathbf{z}_{\phi}}q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{x})\big)\ \frac{\partial}{\partial\phi_i}\mathbf{z}_{\phi}^{(l)}}{\frac{1}{K}\sum_{k=1}^{K}q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{u}_k)\ q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{x})}\Big)\Big\}$$

standard/ MoG

$$\frac{1}{p_{\lambda}(\mathbf{z}_{\phi}^{(l)})\ q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{x})}\Big(q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{x})\frac{\partial}{\partial\mathbf{z}_{\phi}}p_{\lambda}(\mathbf{z}_{\phi}^{(l)}) - p_{\lambda}(\mathbf{z}_{\phi}^{(l)})\frac{\partial}{\partial\mathbf{z}_{\phi}}q_{\phi}(\mathbf{z}_{\phi}^{(l)}|\mathbf{x})\Big)\frac{\partial}{\partial\phi_i}\mathbf{z}_{\phi}^{(l)}$$

New Prior (**Va**riational **M**ixture of **P**osteriors **Prior**)

- Is the VampPrior different than the Mixture of Gaussians? $p_\lambda(\mathbf{z}) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}\left(\mu_k, \mathrm{diag}(\sigma_k^2)\right)$
- VampPrior: the prior and the posterior must "**cooperate**" during training.

VampPrior

$$\frac{1}{K}\sum_{k=1}^{K}\Big\{ \Big(\frac{q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{x})\ \frac{\partial}{\partial\phi_i}q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{u}_k) - q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{u}_k)\ \frac{\partial}{\partial\phi_i}q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{x})}{\frac{1}{K}\sum_{k=1}^{K}q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{u}_k)\ q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{x})}\Big) + \\ + \Big(\frac{\left(q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{x})\ \frac{\partial}{\partial\mathbf{z}_\phi}q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{u}_k) - q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{u}_k)\ \frac{\partial}{\partial\mathbf{z}_\phi}q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{x})\right)\ \frac{\partial}{\partial\phi_i}\mathbf{z}_\phi^{(l)}}{\frac{1}{K}\sum_{k=1}^{K}q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{u}_k)\ q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{x})}\Big)\Big\}$$

standard/ MoG

$$\frac{1}{p_\lambda(\mathbf{z}_\phi^{(l)})\ q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{x})}\Big(q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{x})\frac{\partial}{\partial\mathbf{z}_\phi}p_\lambda(\mathbf{z}_\phi^{(l)}) - p_\lambda(\mathbf{z}_\phi^{(l)})\frac{\partial}{\partial\mathbf{z}_\phi}q_\phi(\mathbf{z}_\phi^{(l)}|\mathbf{x})\Big)\frac{\partial}{\partial\phi_i}\mathbf{z}_\phi^{(l)}$$

New Prior (**Va**riational **M**ixture of **P**osteriors **Prior**)

- VampPrior is closely related to the **Empirical Bayes**.
  - We propose a new approach that learns parameters of the prior and combines the variational inference with the EB approach.

- VampPrior is closely related to the **Information Bottleneck**.
  - The aggregated posterior naturally plays the role of the prior.
  - The VampPrior brings the VAE and the IB formulations together.

Hierarchical VampPrior VAE

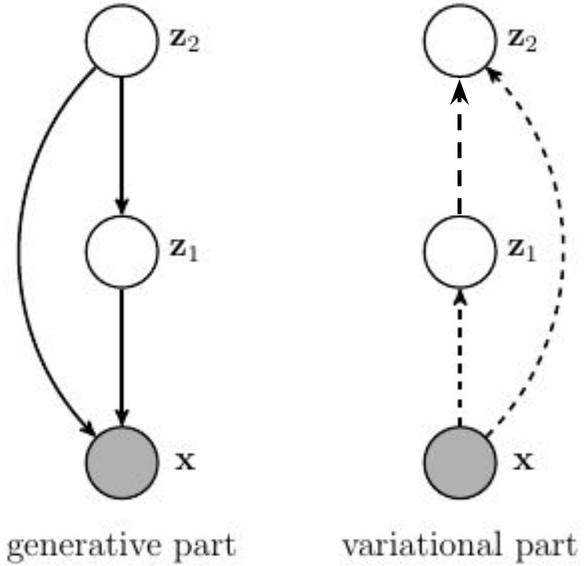
Typical issue in hierarchical VAE: **inactive stochastic units**

$$p(\mathbf{z}_2) = \frac{1}{K} \sum_{k=1}^{K} q_{\psi}(\mathbf{z}_2 | \mathbf{u}_k),$$

$$p_{\lambda}(\mathbf{z}_1 | \mathbf{z}_2) = \mathcal{N}\big(\mathbf{z}_1 | \mu_{\lambda}(\mathbf{z}_2), \mathrm{diag}(\sigma^2_{\lambda}(\mathbf{z}_2))\big),$$

$$q_{\phi}(\mathbf{z}_1 | \mathbf{x}, \mathbf{z}_2) = \mathcal{N}\big(\mathbf{z}_1 | \mu_{\phi}(\mathbf{x}, \mathbf{z}_2), \mathrm{diag}(\sigma^2_{\phi}(\mathbf{x}, \mathbf{z}_2))\big),$$

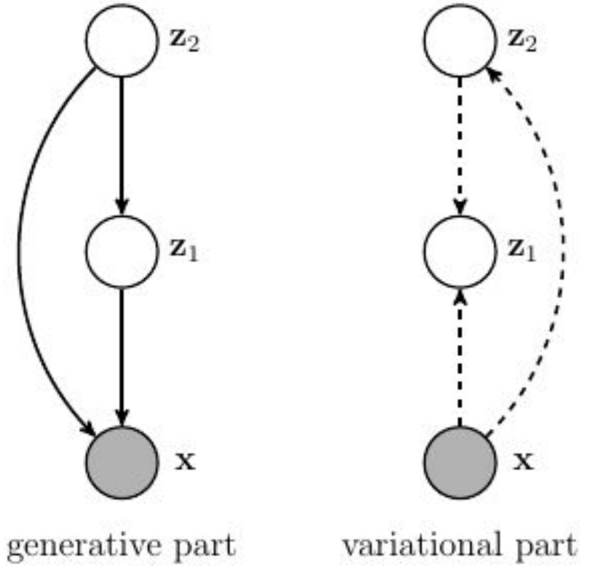
$$q_{\psi}(\mathbf{z}_2 | \mathbf{x}) = \mathcal{N}\big(\mathbf{z}_2 | \mu_{\psi}(\mathbf{x}), \mathrm{diag}(\sigma^2_{\psi}(\mathbf{x}))\big)$$



Hierarchical VampPrior VAE

Typical issue in hierarchical VAE: **inactive stochastic units**

$$
\begin{aligned}
p(\mathbf{z}_2) &= \frac{1}{K}\sum_{k=1}^{K} q_{\psi}(\mathbf{z}_2|\mathbf{u}_k),\\
p_{\lambda}(\mathbf{z}_1|\mathbf{z}_2) &= \mathcal{N}\big(\mathbf{z}_1|\mu_{\lambda}(\mathbf{z}_2), \mathrm{diag}(\sigma^2_{\lambda}(\mathbf{z}_2))\big),\\
q_{\phi}(\mathbf{z}_1|\mathbf{x},\mathbf{z}_2) &= \mathcal{N}\big(\mathbf{z}_1|\mu_{\phi}(\mathbf{x},\mathbf{z}_2), \mathrm{diag}(\sigma^2_{\phi}(\mathbf{x},\mathbf{z}_2))\big),\\
q_{\psi}(\mathbf{z}_2|\mathbf{x}) &= \mathcal{N}\big(\mathbf{z}_2|\mu_{\psi}(\mathbf{x}), \mathrm{diag}(\sigma^2_{\psi}(\mathbf{x}))\big)
\end{aligned}
$$



Hierarchical VampPrior VAE

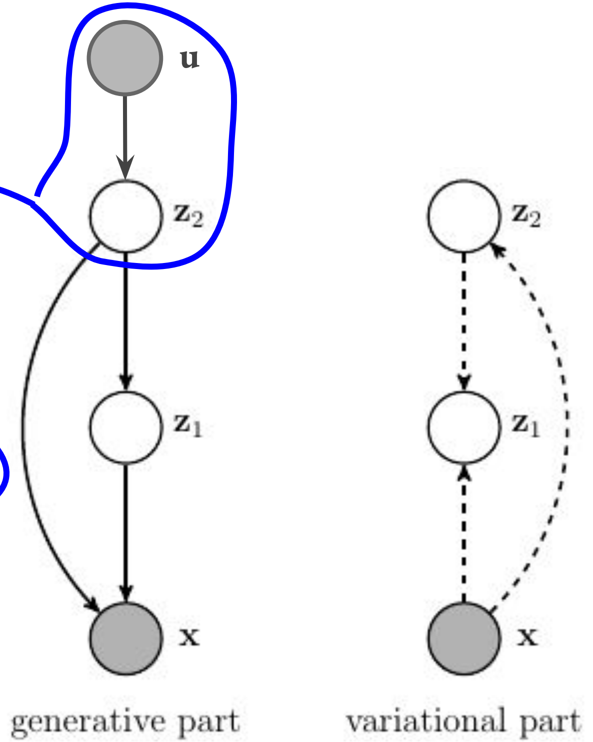
Typical issue in hierarchical VAE: **inactive stochastic units**

$$p(\mathbf{z}_2) = \frac{1}{K}\sum_{k=1}^{K} q_\psi(\mathbf{z}_2|\mathbf{u}_k),$$

$$p_\lambda(\mathbf{z}_1|\mathbf{z}_2) = \mathcal{N}\big(\mathbf{z}_1|\mu_\lambda(\mathbf{z}_2), \mathrm{diag}(\sigma^2_\lambda(\mathbf{z}_2))\big),$$

$$q_\phi(\mathbf{z}_1|\mathbf{x}, \mathbf{z}_2) = \mathcal{N}\big(\mathbf{z}_1|\mu_\phi(\mathbf{x}, \mathbf{z}_2), \mathrm{diag}(\sigma^2_\phi(\mathbf{x}, \mathbf{z}_2))\big),$$

$$q_\psi(\mathbf{z}_2|\mathbf{x}) = \mathcal{N}\big(\mathbf{z}_2|\mu_\psi(\mathbf{x}), \mathrm{diag}(\sigma^2_\psi(\mathbf{x}))\big)$$



It counteracts inactive stochastic hidden units problem!

Experiments

Dataset	VAE ($L = 1$)		HVAE ($L = 2$)		convHVAE ($L = 2$)		PixelHVAE ($L = 2$)	
	standard	VampPrior	standard	VampPrior	standard	VampPrior	standard	VampPrior
staticMNIST	−88.56	**−85.57**	−86.05	**−83.19**	−82.41	**−81.09**	−80.58	**−79.78**
dynamicMNIST	−84.50	**−82.38**	−82.42	**−81.24**	−80.40	**−79.75**	−79.70	**−78.45**
Omniglot	−108.50	**−104.75**	−103.52	**−101.18**	−97.65	**−97.56**	−90.11	**−89.76**
Caltech 101	−123.43	**−114.55**	−112.08	**−108.28**	−106.35	**−104.22**	**−85.51**	−86.22
Frey Faces	4.63	**4.57**	4.61	**4.51**	4.49	**4.45**	4.43	**4.38**
Histopathology	6.07	**6.04**	5.82	**5.75**	5.59	**5.58**	4.84	**4.82**

Experiments

Table 2: Test LL for static MNIST.

Model	LL
VAE ($L=1$) + NF [32]	-85.10
VAE ($L=2$) [6]	-87.86
IWAE ($L=2$) [6]	-85.32
HVAE ($L=2$) + SG	-85.89
HVAE ($L=2$) + MoG	-85.07
HVAE ($L=2$) + VampPrior *data*	-85.71
HVAE ($L=2$) + VampPrior	$\mathbf{-83.19}$
AVB + AC ($L=1$) [28]	-80.20
VLAE [7]	$\mathbf{-79.03}$
VAE + IAF [18]	-79.88
convHVAE ($L=2$) + VampPrior	-81.09
PixelHVAE ($L=2$) + VampPrior	-79.78

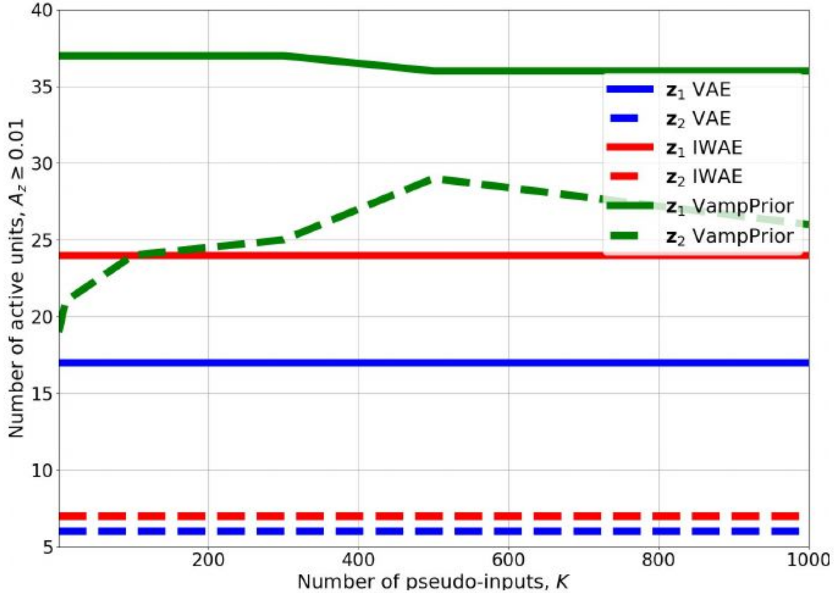


Figure 3: A comparison between two-level VAE and IWAE with the standard normal prior and theirs VampPrior counterpart in terms of number of active units for varying number of pseudo-inputs on static MNIST.

Experiments

Table 3: Test LL for dynamic MNIST.

Model	LL
VAE ($L=2$) + VGP [40]	-81.32
CaGeM-0 ($L=2$) [25]	-81.60
LVAE ($L=5$) [36]	-81.74
HVAE ($L=2$) + VampPrior *data*	-81.71
HVAE ($L=2$) + VampPrior	$\mathbf{-81.24}$
VLAE [7]	-78.53
VAE + IAF [18]	-79.10
PixelVAE [15]	-78.96
convHVAE ($L=2$) + VampPrior	-79.78
PixelHVAE ($L=2$) + VampPrior	$\mathbf{-78.45}$

Table 4: Test LL for OMNIGLOT.

Model	LL
VR-max ($L=2$) [24]	-103.72
IWAE ($L=2$) [6]	-103.38
LVAE ($L=5$) [36]	-102.11
HVAE ($L=2$) + VampPrior	$\mathbf{-101.18}$
VLAE [7]	-89.83
convHVAE ($L=2$) + VampPrior	-97.56
PixelHVAE ($L=2$) + VampPrior	$\mathbf{-89.76}$

Table 5: Test LL for Caltech 101 Silhouettes.

Model	LL
IWAE ($L=1$) [24]	-117.21
VR-max ($L=1$) [24]	-117.10
HVAE ($L=2$) + VampPrior	$\mathbf{-108.28}$
VLAE [7]	$\mathbf{-78.53}$
convHVAE ($L=2$) + VampPrior	-104.22
PixelHVAE ($L=2$) + VampPrior	-86.22

Experiments

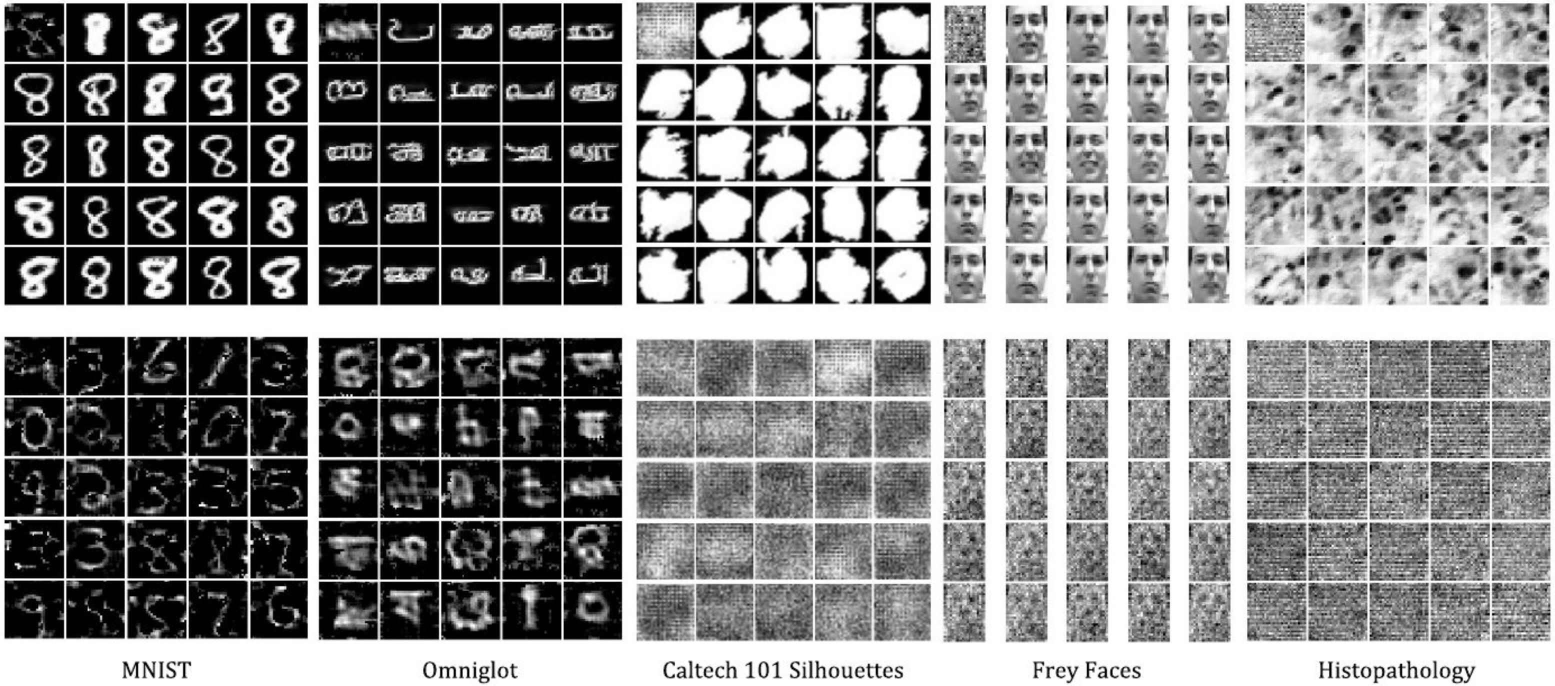


Figure 4: (*top row*) Images generated by PixelHVAE + VampPrior for chosen pseudo-input in the left top corner. (*bottom row*) Images represent a subset of trained pseudo-inputs for different datasets.

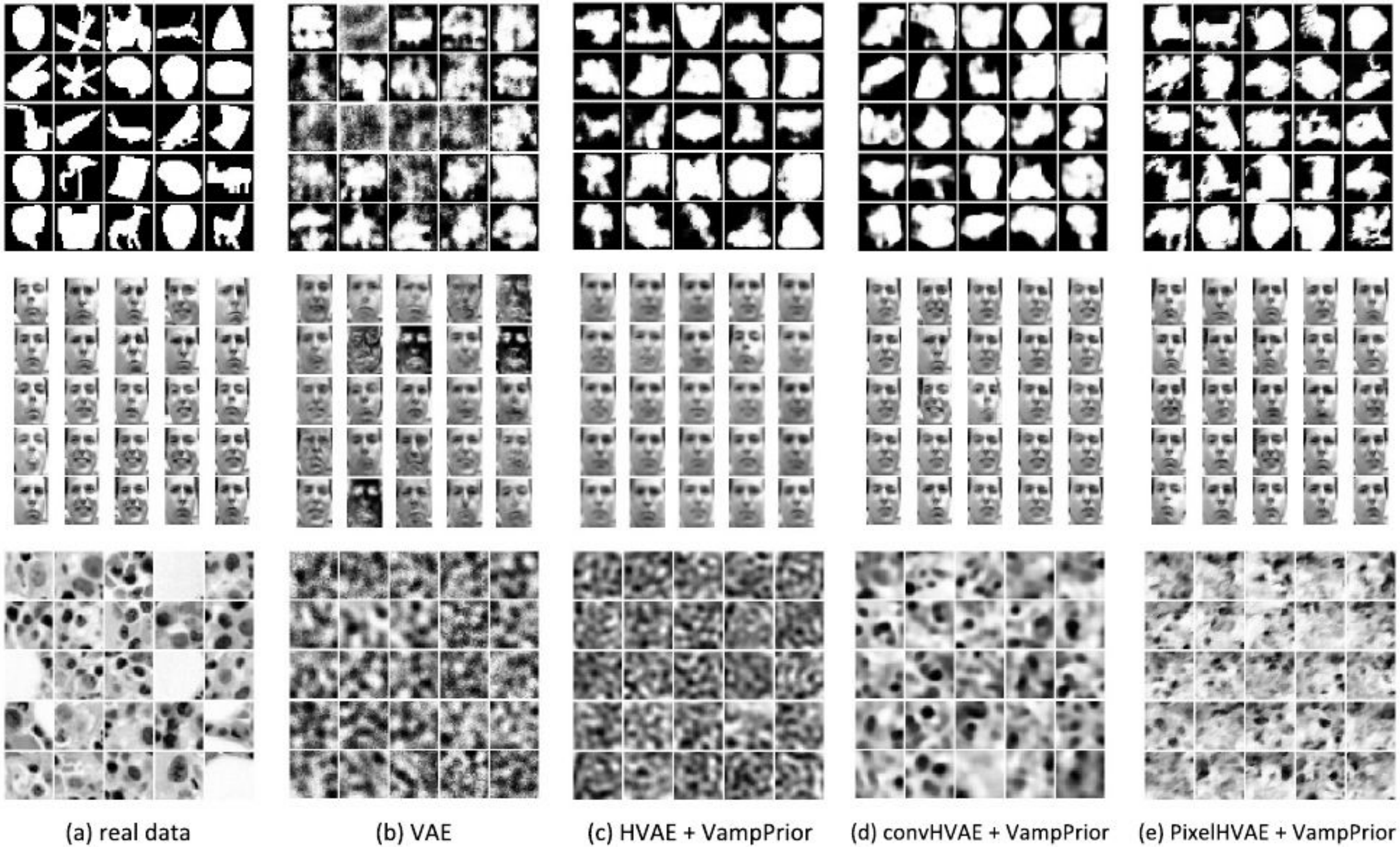


Figure 5: (a) Real images from test sets and images generated by (b) the vanilla VAE, (c) the HVAE ($L = 2$) + VampPrior, (d) the convHVAE ($L = 2$) + VampPrior and (e) the PixelHVAE ($L = 2$) + VampPrior.

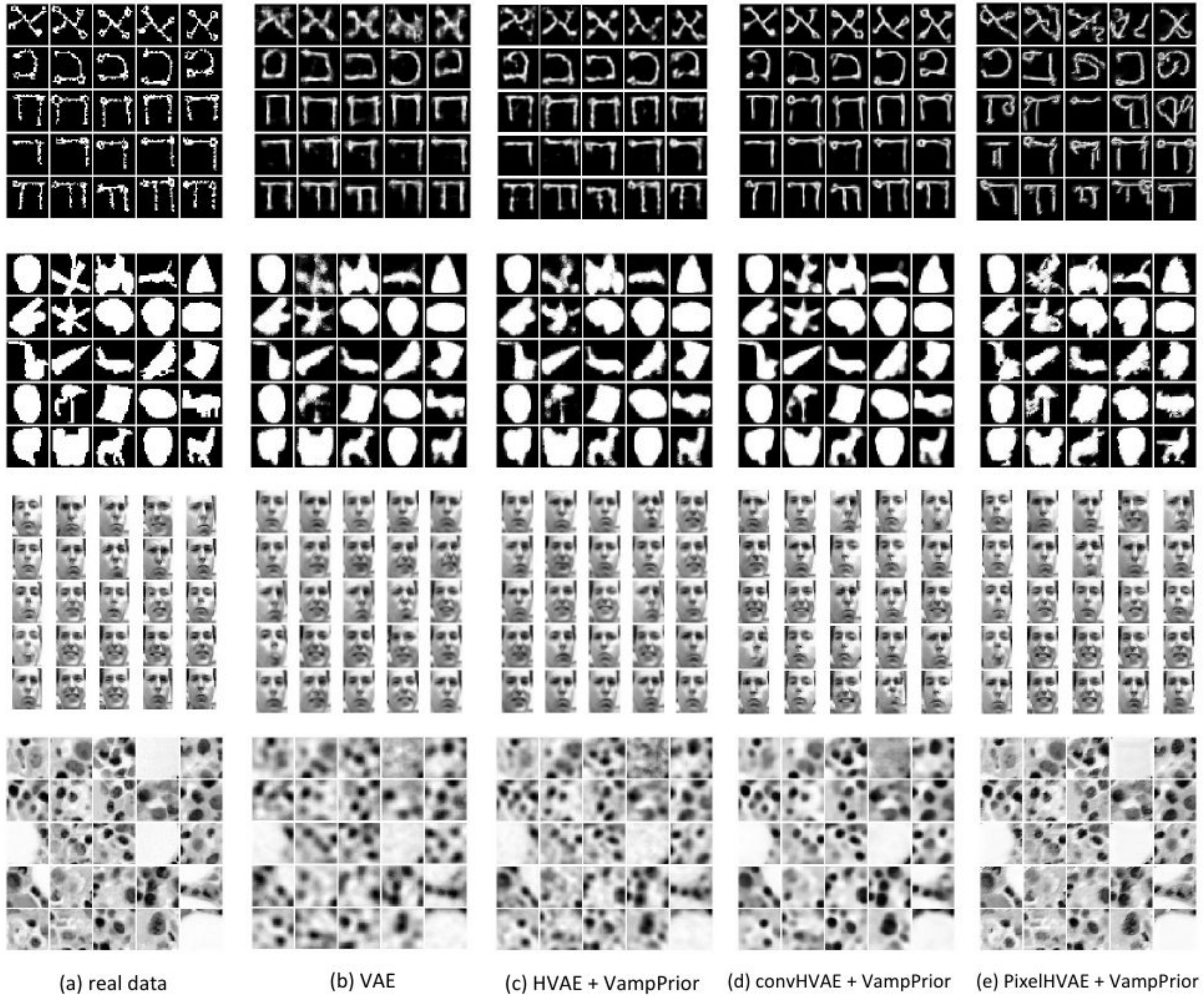


Figure 6: (a) Real images from test sets, (b) reconstructions given by the vanilla VAE, (c) the HVAE ($L = 2$) + VampPrior, (d) the convHVAE ($L = 2$) + VampPrior and (e) the PixelHVAE ($L = 2$) + VampPrior.

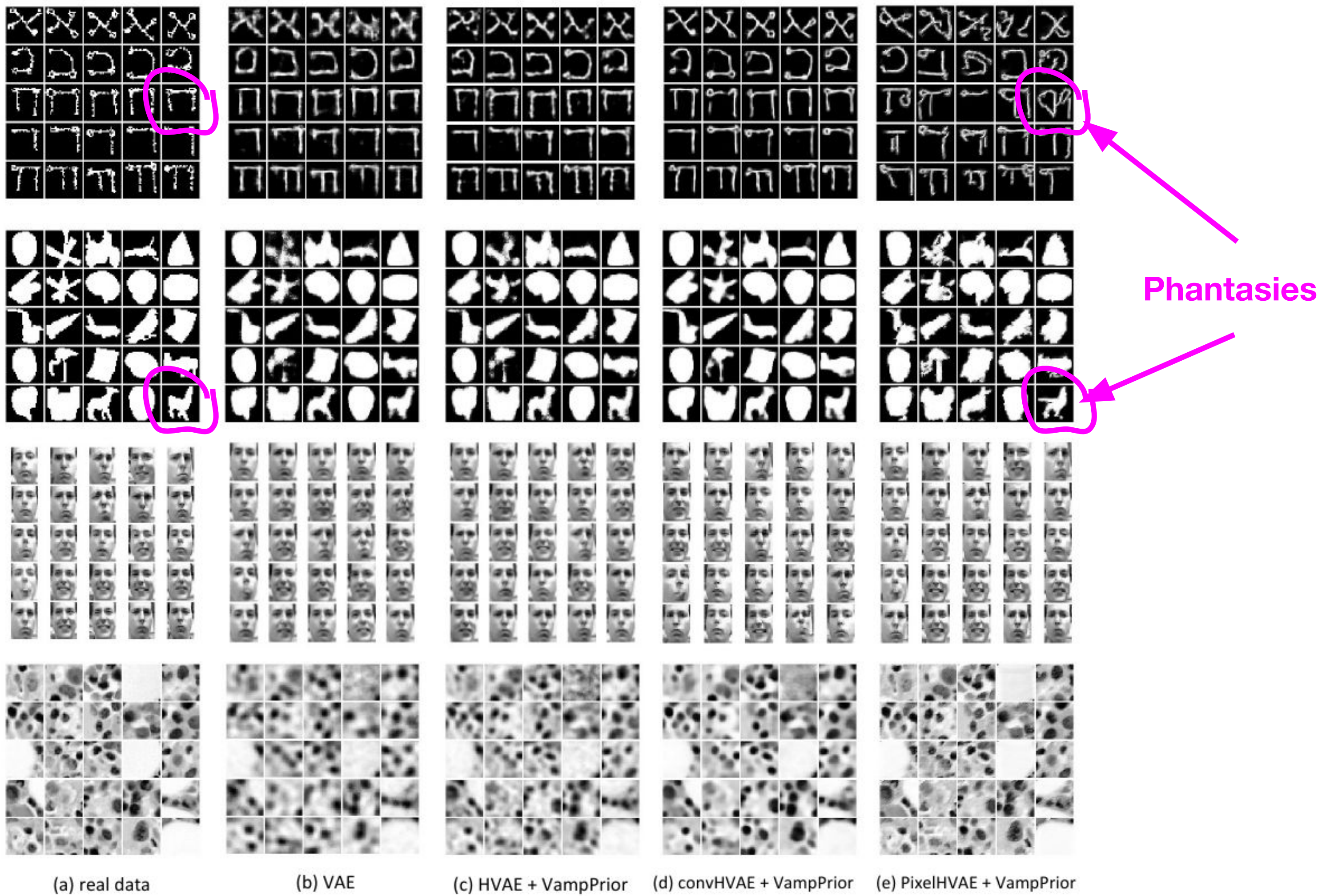


Figure 6: (a) Real images from test sets, (b) reconstructions given by the vanilla VAE, (c) the HVAE ($L = 2$) + VampPrior, (d) the convHVAE ($L = 2$) + VampPrior and (e) the PixelHVAE ($L = 2$) + VampPrior.

Conclusion

The **prior** in VAE is extremely important.

VampPrior = approximated aggregated posterior as the optimal prior

Hierarchical VampPrior VAE → **less** inactive stochastic units.

Multimodal prior → **better** generative process

Conclusion

The **prior** in VAE is extremely important.

VampPrior = approximated aggregated posterior as the optimal prior

Hierarchical VampPrior VAE → **less** inactive stochastic units.

Multimodal prior → **better** generative process

Conclusion

The prior in VAE is extremely important.

VampPrior = **approximated aggregated posterior as the optimal prior**

Hierarchical VampPrior VAE → **less** inactive stochastic units.

Multimodal prior → **better** generative process

Conclusion

The prior in VAE is extremely important.

VampPrior = **approximated aggregated posterior** as **the optimal prior**

Hierarchical VampPrior VAE → **less** inactive stochastic units.

Multimodal prior → **better** generative process

Conclusion

The prior in VAE is extremely important.

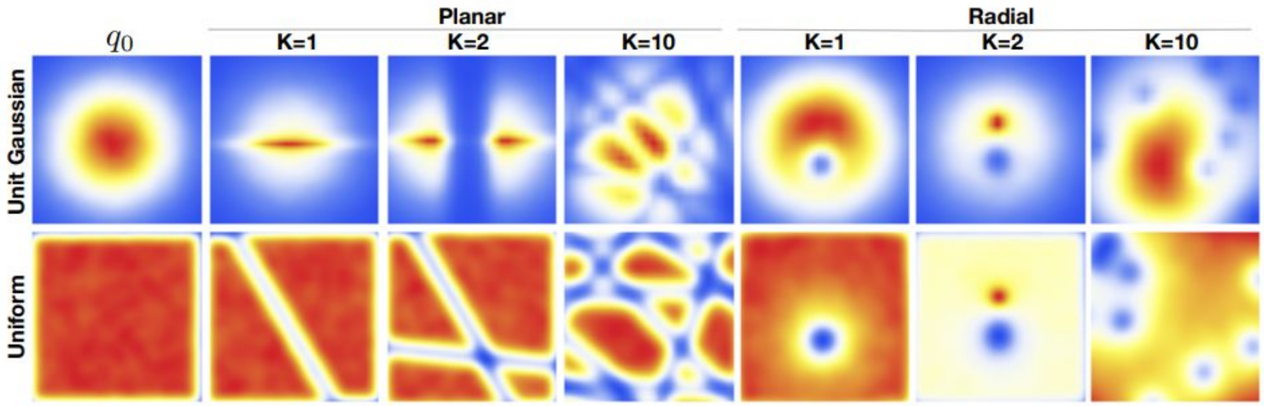
VampPrior = **approximated aggregated posterior** **as the optimal prior**

Hierarchical VampPrior VAE → **less** inactive stochastic units.

Multimodal prior → **better** generative process

Future directions

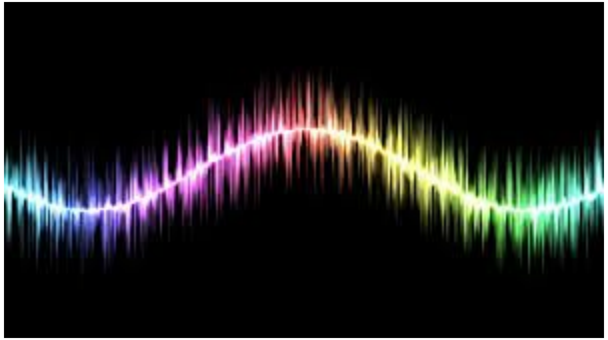
VampPrior +
Normalizing flows

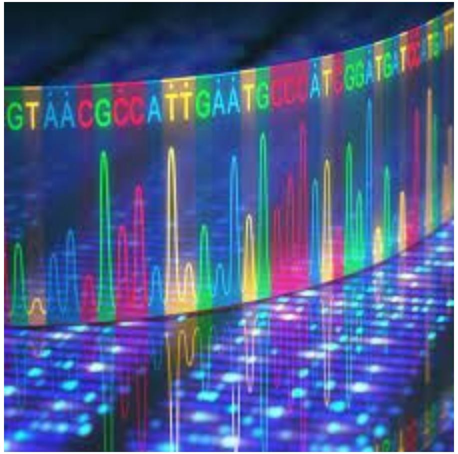


Future directions

VampPrior for other data (sequential, sound, text, genomics, etc.)

→RNN posteriors



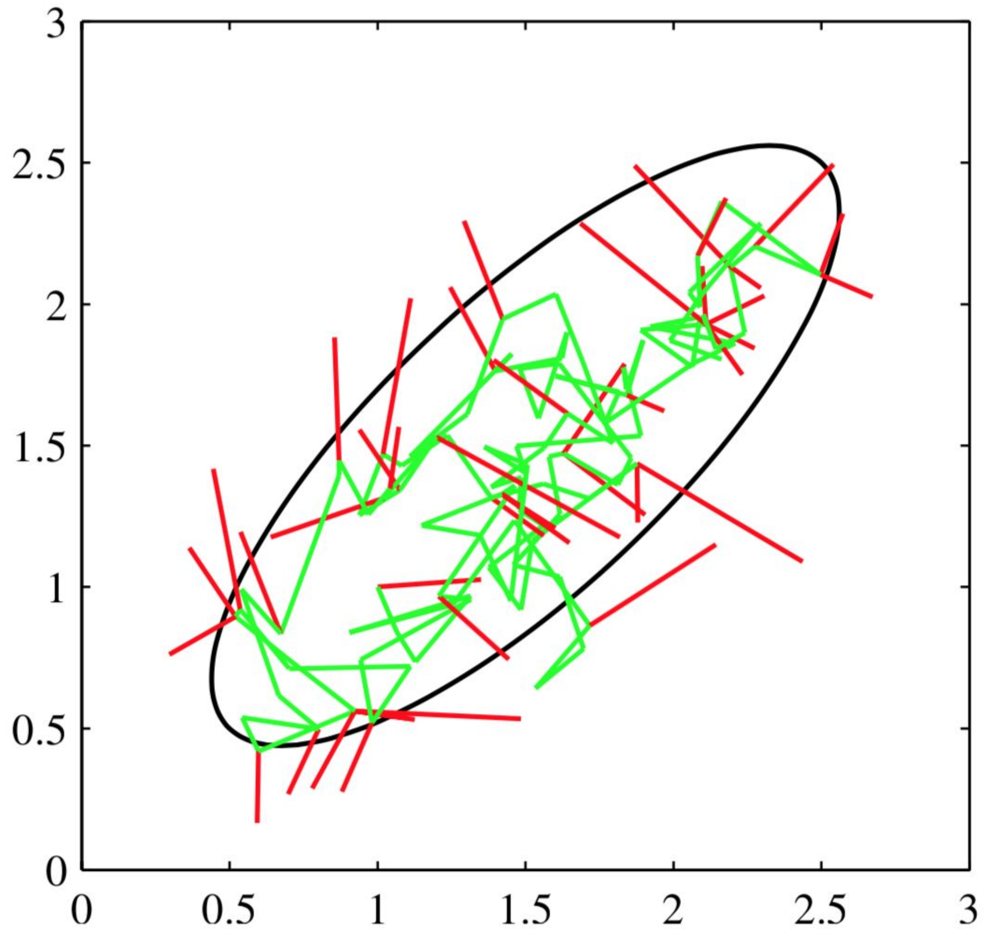


Future directions

How to (better) learn pseudoinputs?

→MCMC?

→Wake-Sleep?



Webpage:

https://jmtomczak.github.io/

Code on github:

https://github.com/jmtomczak/

Contact:

jakubmkt@gmail.com