Deep Generative Modeling using Variational Auto-Encoders

Jakub Tomczak

AMLAB, University of Amsterdam

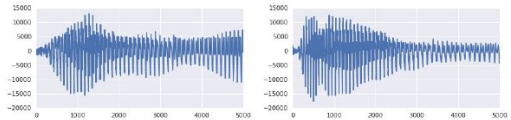
Amsterdam, 2018

Artificial Intelligence

“ i want to talk to you . ”
“i want to be with you . ”
“i do n’t want to be with you . ”
i do n’t want to be with you .
she did n’t want to be with him .

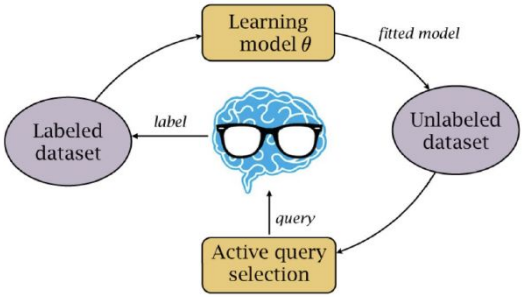
he was silent for a long moment .
he was silent for a moment .
it was quiet for a moment .
it was dark and cold .
there was a pause .
it was my turn .

Text analysis

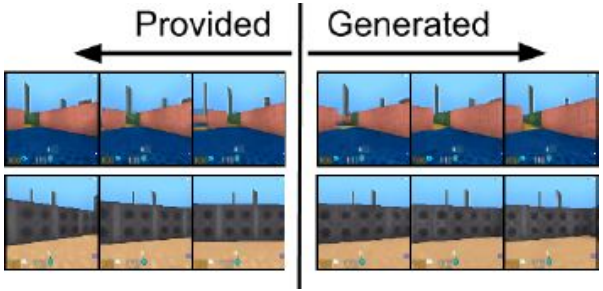


Audio analysis

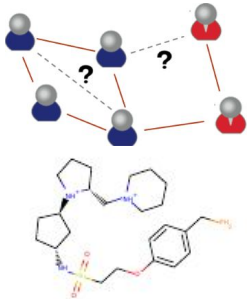
Image analysis



Active Learning



Reinforcement Learning



Graph analysis

and more...

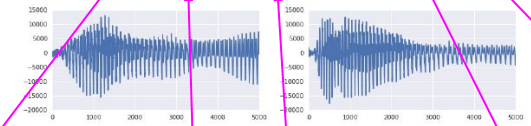
Artificial Intelligence

(Deep) Generative Modeling

“ i want to talk to you . ”
“i want to be with you . ”
“i do n’t want to be with you . ”
i do n’t want to be with you .
she did n’t want to be with him .

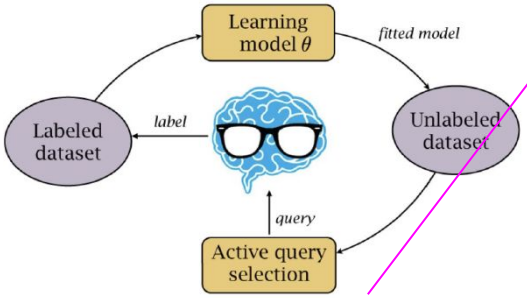
he was silent for a long moment .
he was silent for a moment .
it was quiet for a moment .
it was dark and cold .
there was a pause .
it was my turn .

Text analysis

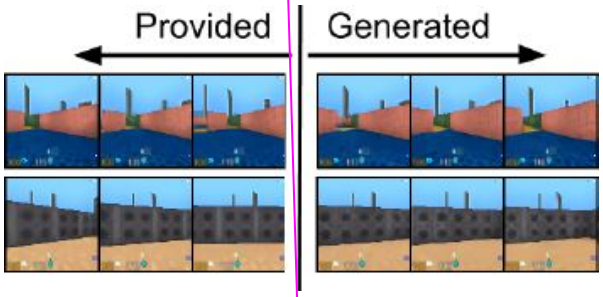


Audio analysis

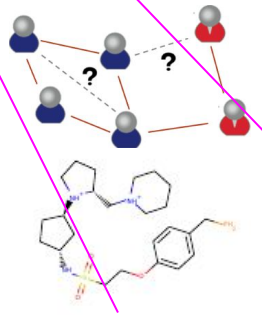
Image analysis



Active Learning



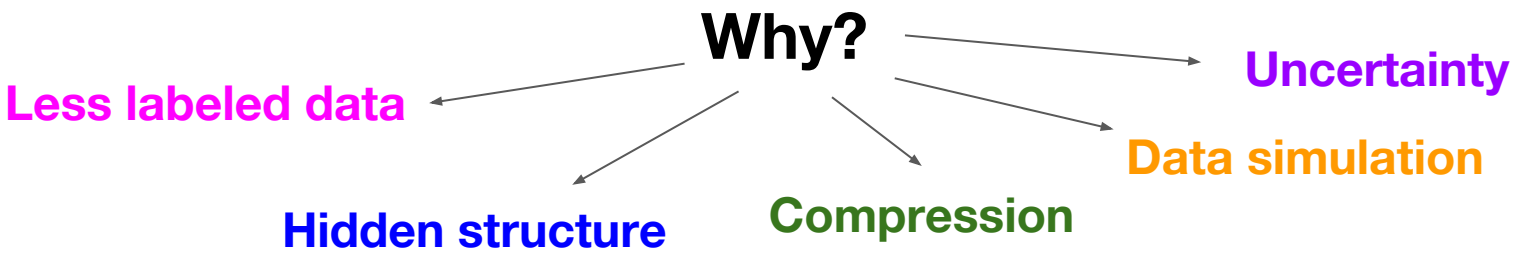
Reinforcement Learning



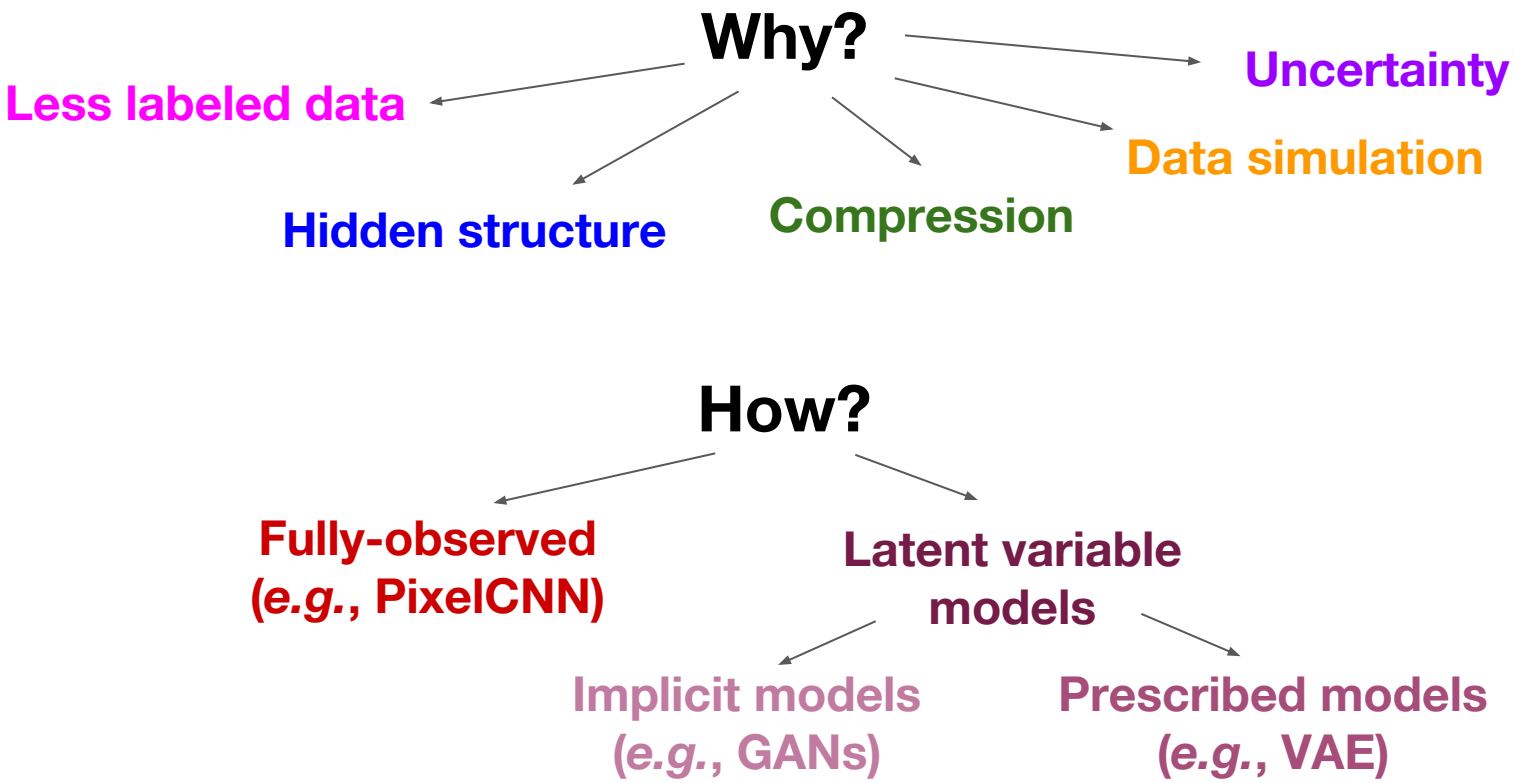
Graph analysis

and more...

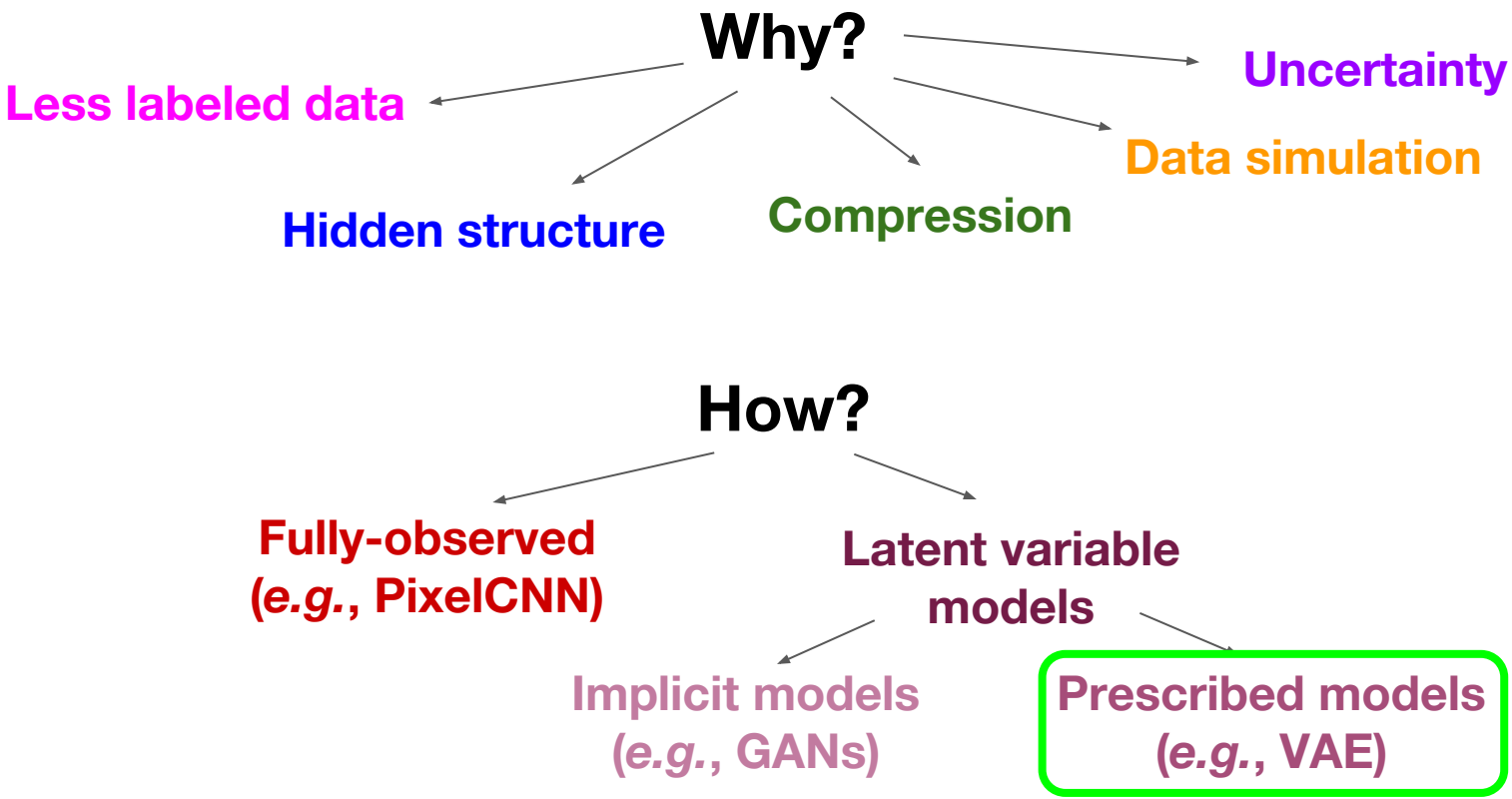
(Deep) generative modeling



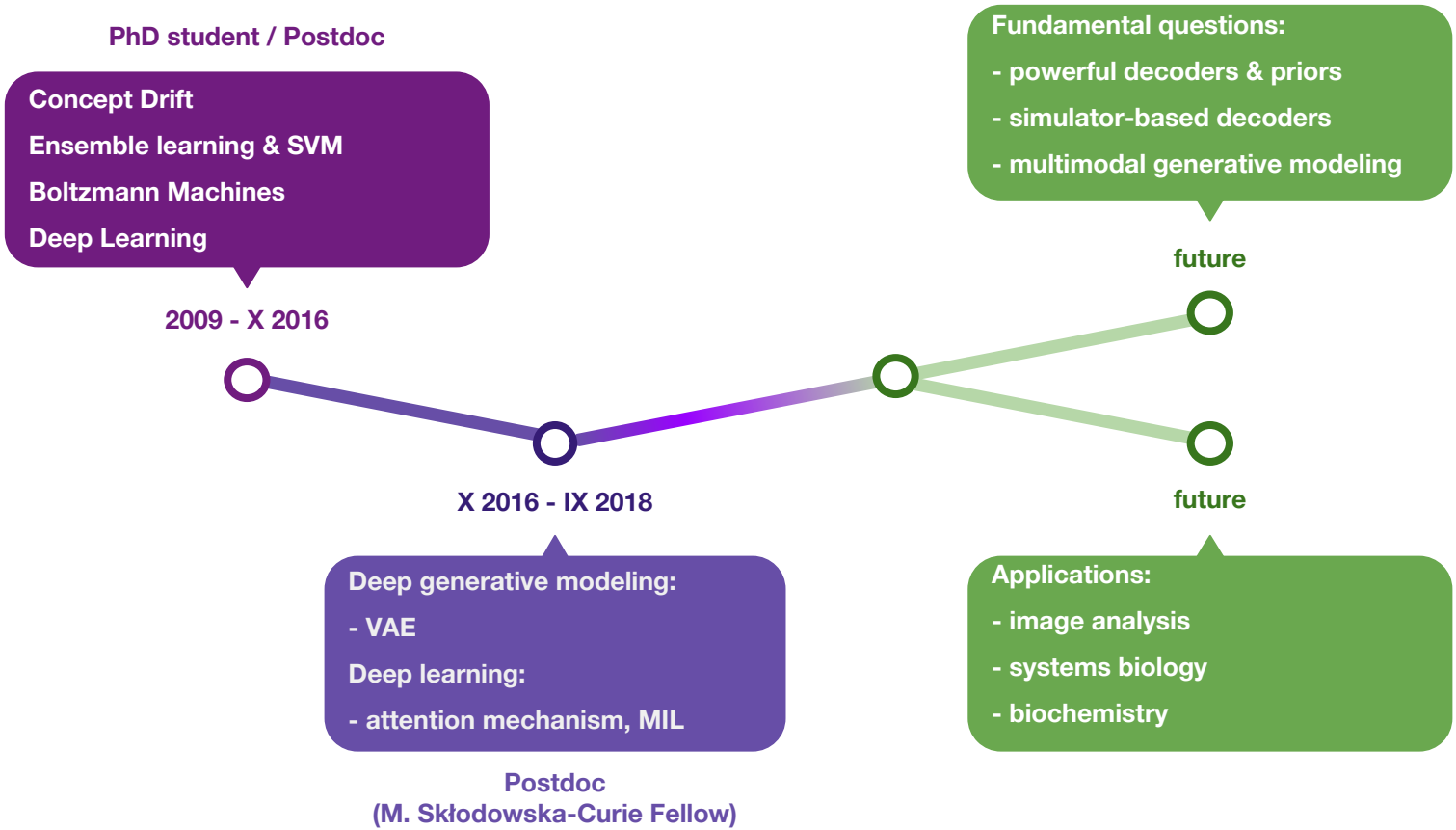
(Deep) generative modeling



(Deep) generative modeling



My vision


PhD student / Postdoc
Concept Drift
Ensemble learning & SVM
Boltzmann Machines
Deep Learning
2009 - X 2016
X 2016 - IX 2018
Deep generative modeling:
- VAE
Deep learning:
- attention mechanism, MIL
Postdoc
(M. Skłodowska-Curie Fellow)
Fundamental questions:
- powerful decoders & priors
- simulator-based decoders
- multimodal generative modeling
future
future
Applications:
- image analysis
- systems biology
- biochemistry

Generative modeling using Latent Variable Models

Modeling in a high-dimensional space is difficult.

Generative modeling using Latent Variable Models

Modeling in a high-dimensional space is difficult.

Generative modeling using Latent Variable Models

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 using Latent Variable Models

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 using Latent Variable Models

Modeling in a high-dimensional space is difficult.

→modeling all dependencies among pixels.

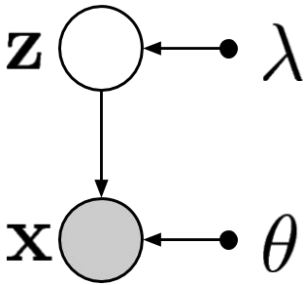
$$p(x) = \prod_{d=1}^{c} \psi_c(x_c)$$

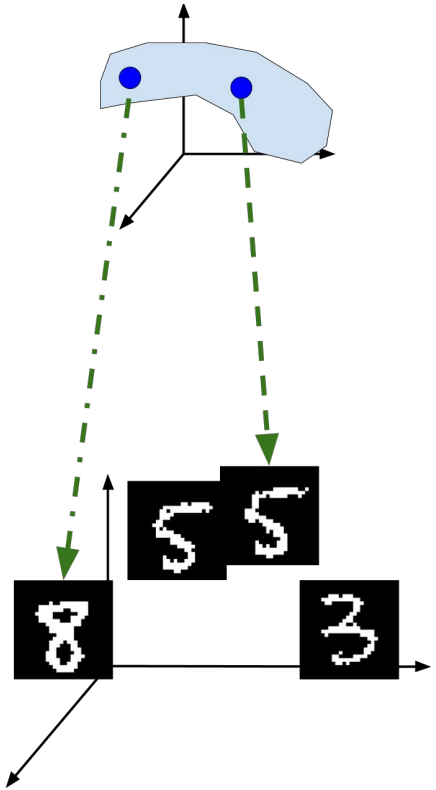
Very inefficient!

A possible **solution**? →**Latent Variable Models**

Generative modeling using Latent Variable Models

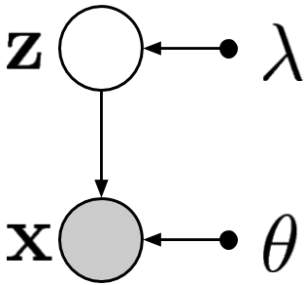
$$p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

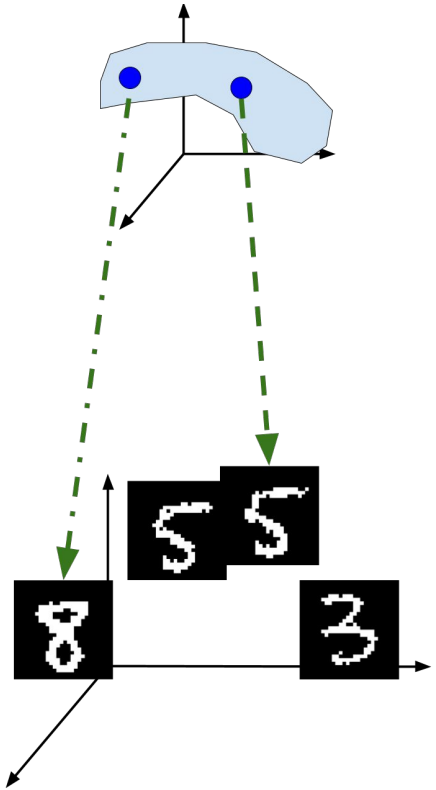




Generative modeling using Latent Variable Models

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





How to train it efficiently?

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

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

Variational posterior

$$= \log \int \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\ p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

$$\geq \int q_\phi(\mathbf{z}|\mathbf{x})\ \log \frac{p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\ \mathrm{d}\mathbf{z}$$

$$= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p_\lambda(\mathbf{z})]$$

Variational inference for Latent Variable Models

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

$$= \log \int \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\ p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

Jensen's inequality

$$\geq \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\ \mathrm{d}\mathbf{z}$$

$$= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p_\lambda(\mathbf{z})]$$

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 inference for Latent Variable Models

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

$$= \log \int \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\ p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

$$\geq \int q_\phi(\mathbf{z}|\mathbf{x})\ \log \frac{p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\ \mathrm{d}\mathbf{z}$$

$$= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x})|p_\lambda(\mathbf{z})]$$

decoder

encoder

prior

Variational inference for Latent Variable Models

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

$$= \log \int \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\ p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

$$\geq \int q_\phi(\mathbf{z}|\mathbf{x})\ \log \frac{p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\ \mathrm{d}\mathbf{z}$$

$$= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x})|p_\lambda(\mathbf{z})]$$

decoder (Neural Net)

encoder (Neural Net)

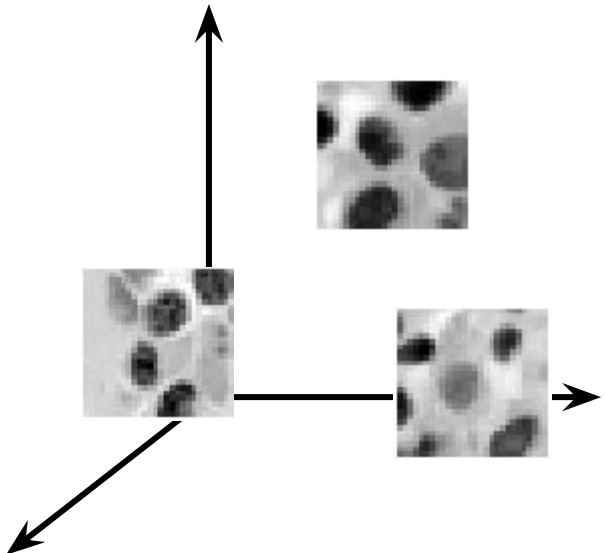
prior

\+ reparameterization trick

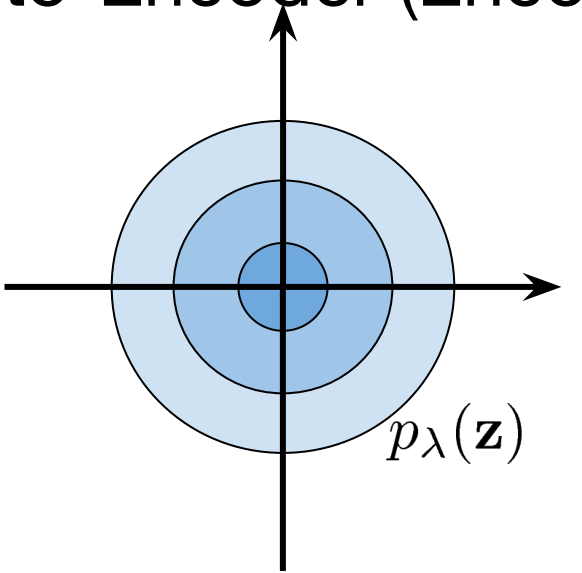
= Variational Auto-Encoder

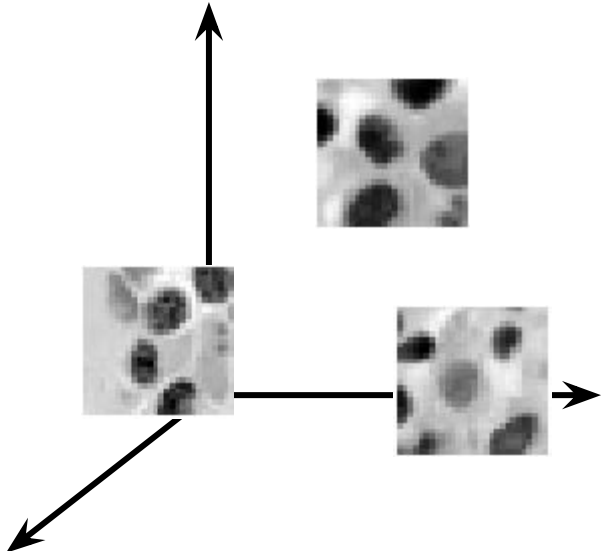
Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Variational Auto-Encoder (Encoding-Decoding)

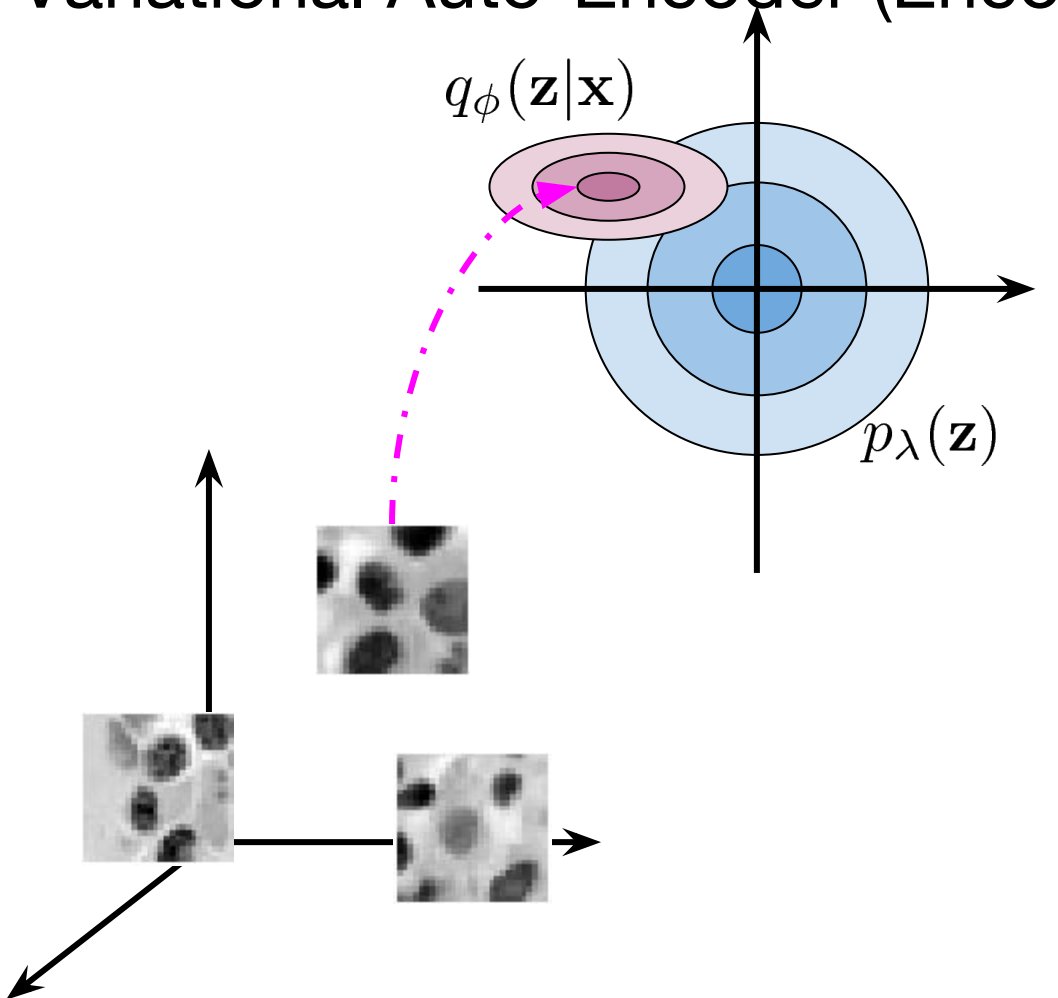


Variational Auto-Encoder (Encoding-Decoding)

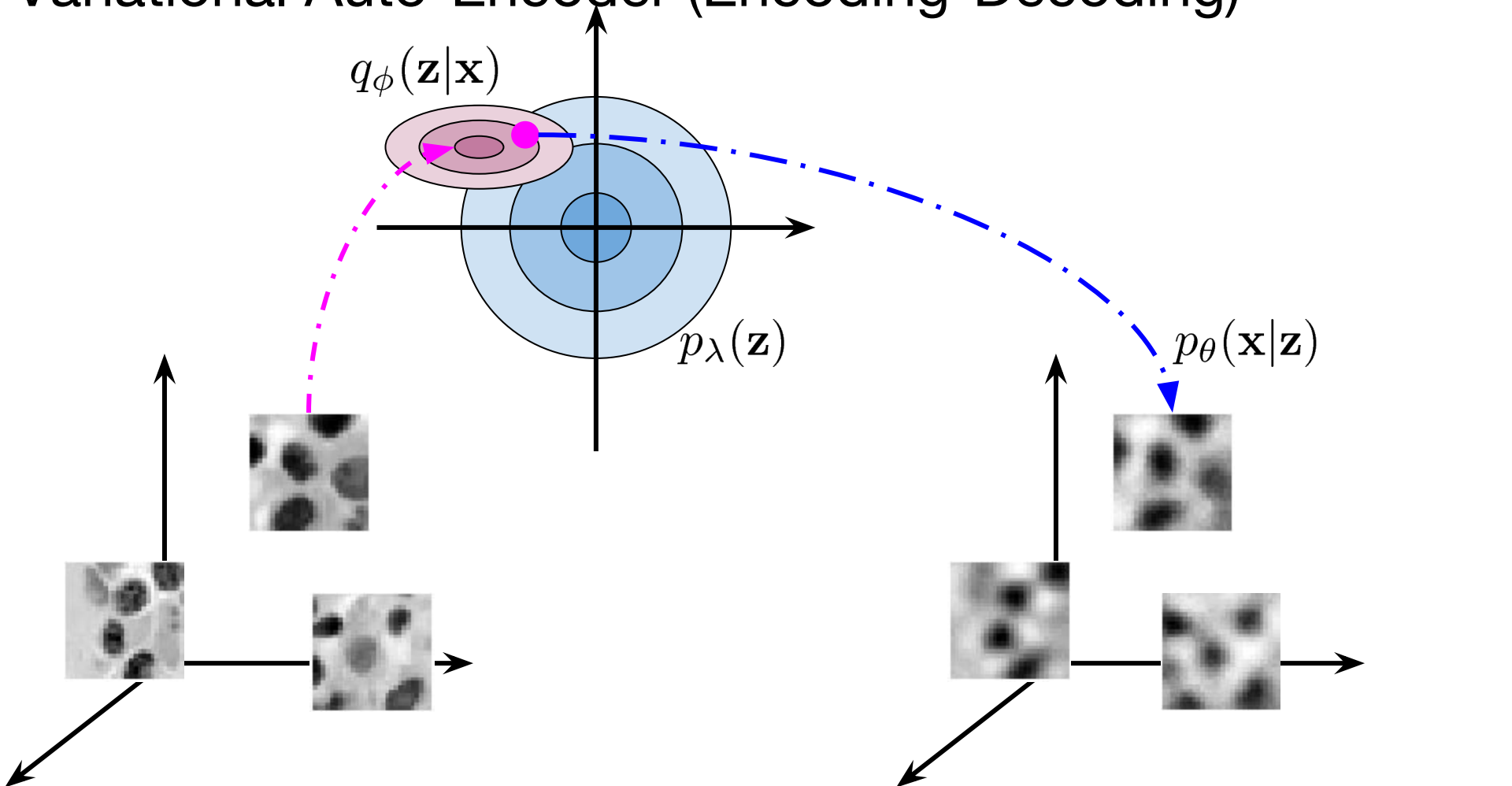




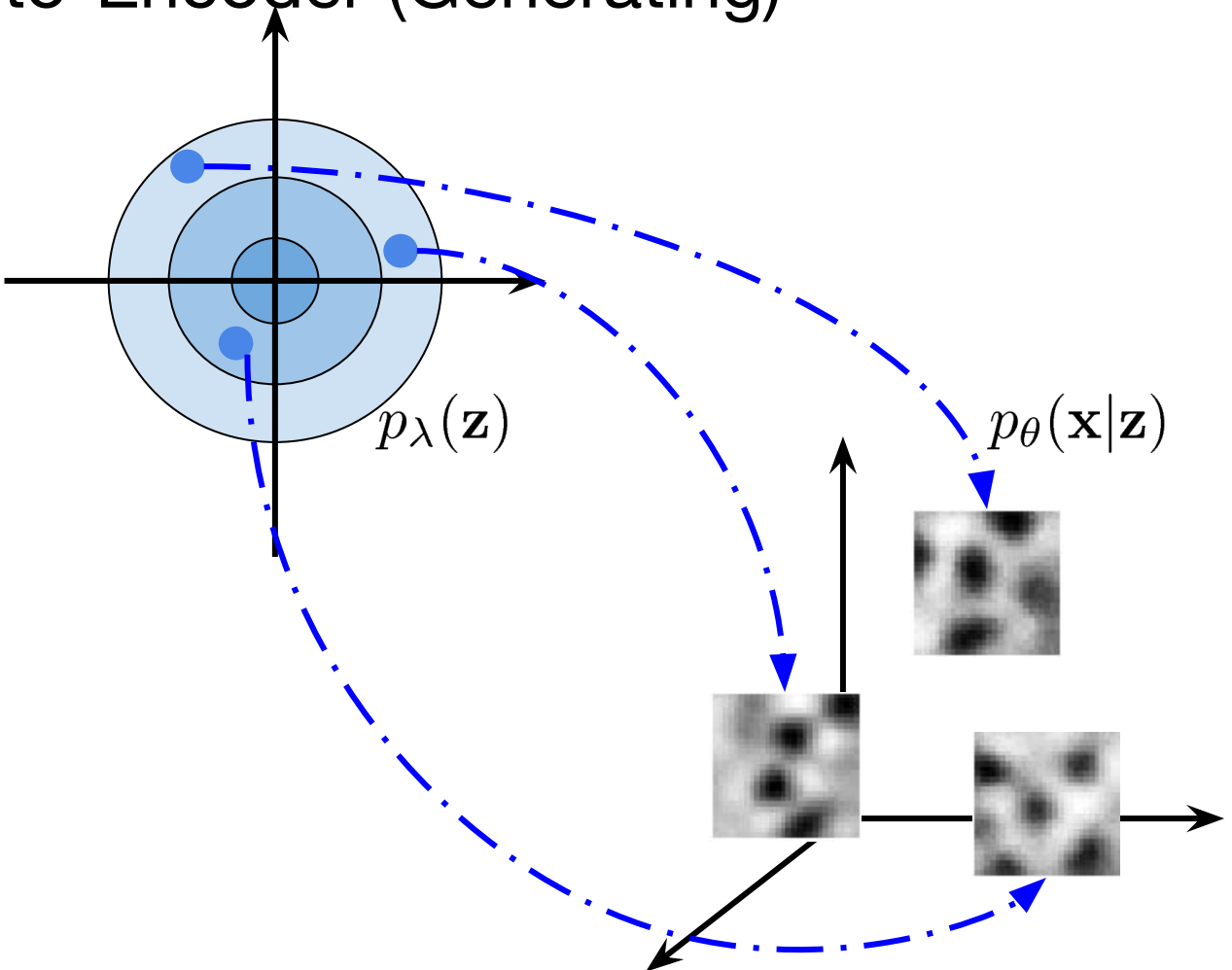
Variational Auto-Encoder (Encoding-Decoding)



Variational Auto-Encoder (Encoding-Decoding)

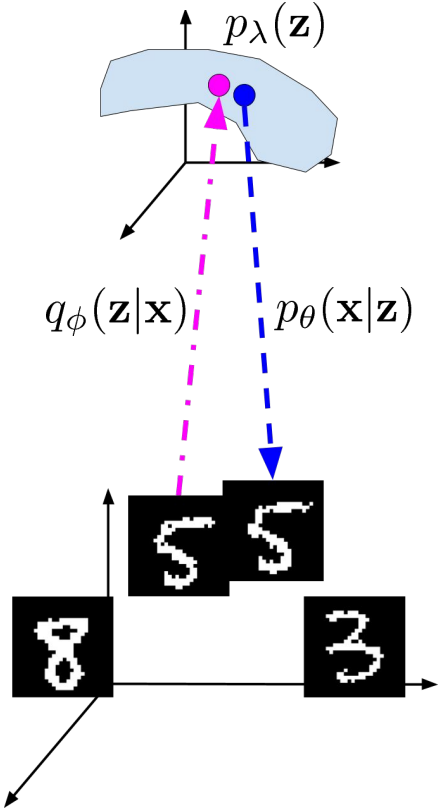


Variational Auto-Encoder (Generating)



Variational Auto-Encoder

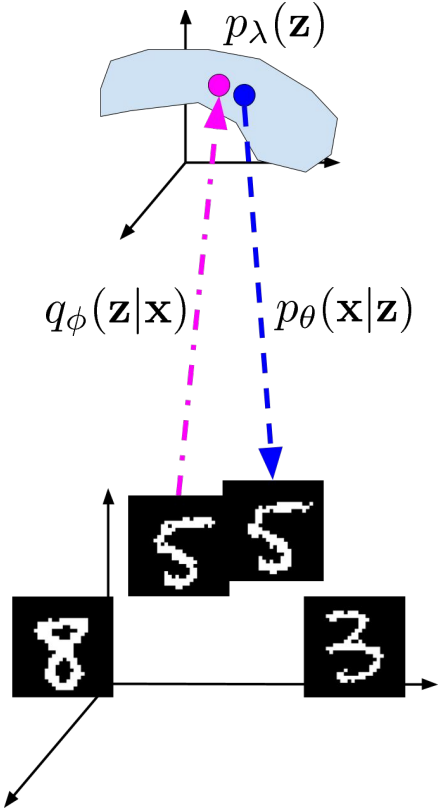
$$q_\phi(\mathbf{z}|\mathbf{x}) \propto p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})$$



Variational Auto-Encoder

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

Fully-connected
ConvNets
PixelCNN
Other

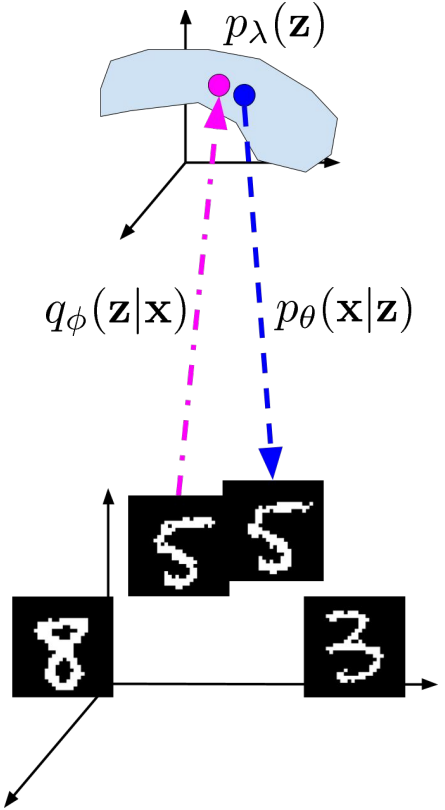


Variational Auto-Encoder

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

Normalizing flows
Volume-preserving flows
non-Gaussian distributions

Fully-connected
ConvNets
PixelCNN
Other



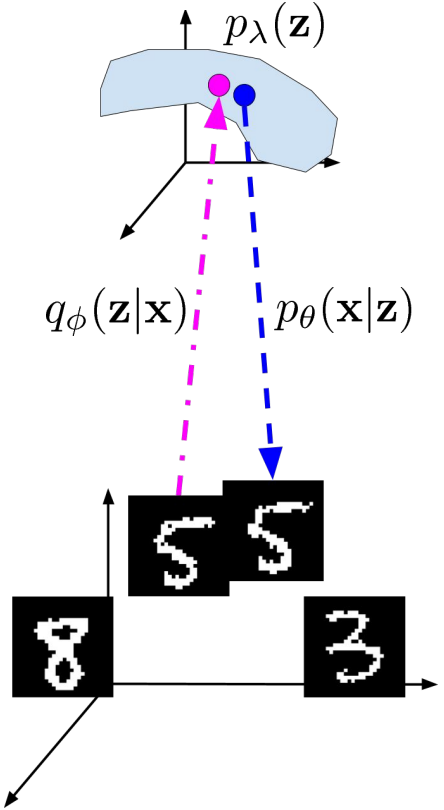
Variational Auto-Encoder

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

Normalizing flows
Volume-preserving flows
non-Gaussian distributions

Fully-connected
ConvNets
PixelCNN
Other

Autoregressive Prior
Objective Prior
Stick-Breaking Prior
VampPrior



Variational Auto-Encoder

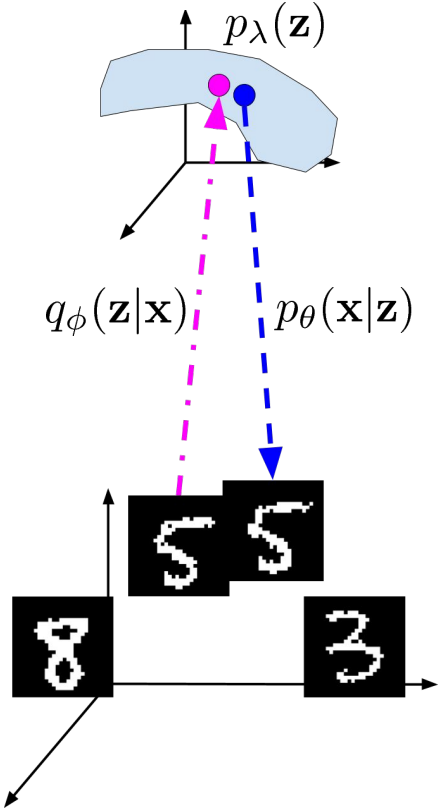
$$q_\phi(\mathbf{z}|\mathbf{x}) \propto p_\theta(\mathbf{x}|\mathbf{z})\, p_\lambda(\mathbf{z})$$

Normalizing flows
Volume-preserving flows
non-Gaussian distributions

Fully-connected
ConvNets
PixelCNN
Other

Autoregressive Prior
Objective Prior
Stick-Breaking Prior
VampPrior

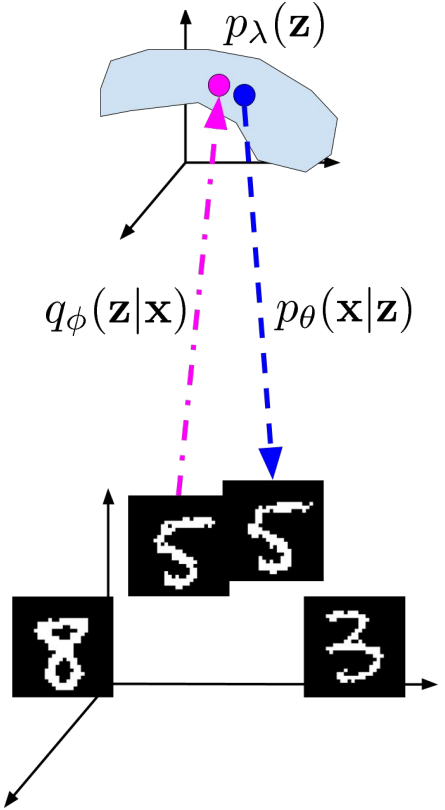
Importance Weighted AE
Renyi Divergence
Stein Divergence



Variational Auto-Encoder

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

Normalizing flows
Volume-preserving flows
non-Gaussian distributions



Improving posterior using Normalizing Flows

- Diagonal posterior - **insufficient** and **inflexible**.
- How to get more flexible posterior?
  - ➢ Apply a series of *T* invertible transformations $\mathbf{f}^{(t)}$ to $\mathbf{z}^{(0)} \sim q(\mathbf{z}|\mathbf{x})$
- New objective:

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}^{(0)}|\mathbf{x})}\left[\ln p(\mathbf{x}|\mathbf{z}^{(T)}) + \sum_{t=1}^{T} \ln \left|\det \frac{\partial \mathbf{f}^{(t)}}{\partial \mathbf{z}^{(t-1)}}\right|\right] - \mathrm{KL}\left(q(\mathbf{z}^{(0)}|\mathbf{x})||p(\mathbf{z}^{(T)})\right).$$

Rezende, D. J., & Mohamed, S. (2015). Variational inference with normalizing flows. arXiv preprint arXiv:1505.05770. ICML 2015

Improving posterior using Normalizing Flows

- Diagonal posterior - **insufficient** and **inflexible**.
- How to get more flexible posterior?
  - ➢ Apply a series of *T* invertible transformations $\mathbf{f}^{(t)}$ to $\mathbf{z}^{(0)} \sim q(\mathbf{z}|\mathbf{x})$
- New objective:

$$\ln p(\mathbf{x}) \geq$$

Change of variables:

$$q(\mathbf{z}^{(T)}|\mathbf{x}) = q(\mathbf{z}^{(0)}|\mathbf{x}) \prod_{t=1}^{T} \left| \det \frac{\partial f^{(t)}}{\partial \mathbf{z}^{(t-1)}} \right|^{-1}$$

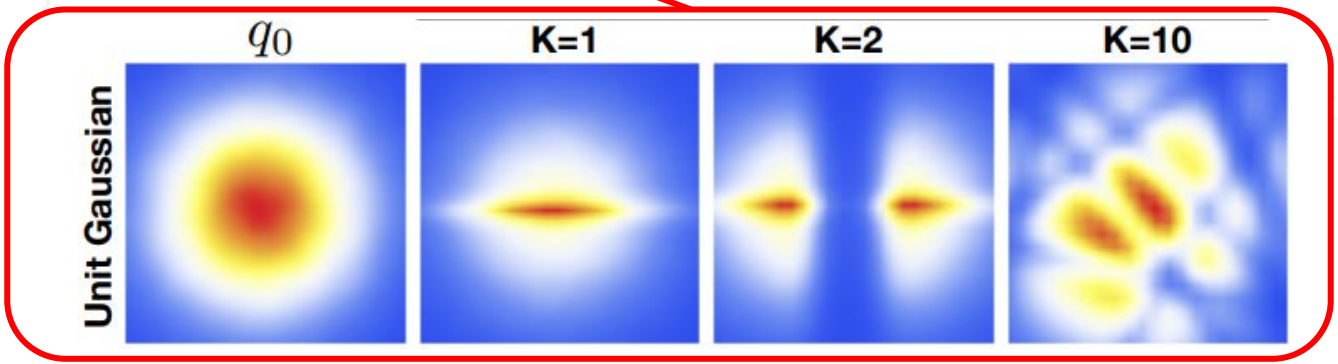
$$)||p(\mathbf{z}^{(T)})).$$

Rezende, D. J., & Mohamed, S. (2015). Variational inference with normalizing flows. arXiv preprint arXiv:1505.05770. ICML 2015

Improving posterior using Normalizing Flows

- Diagonal posterior - **insufficient** and **inflexible**.
- How to get more flexible posterior?
  - ➢ Apply a series of *T* invertible transformations $\mathbf{f}^{(t)}$ to $\mathbf{z}^{(0)} \sim q(\mathbf{z}|\mathbf{x})$
- New objective:

$$\ln p(\mathbf{x}) \geq$$



$$)||p(\mathbf{z}^{(T)})).$$

Rezende, D. J., & Mohamed, S. (2015). Variational inference with normalizing flows. arXiv preprint arXiv:1505.05770. ICML 2015

Improving posterior using Normalizing Flows

- Diagonal posterior - **insufficient** and **inflexible**.
- How to get more flexible posterior?
  - ➢ Apply a series of *T* invertible transformations $\mathbf{f}^{(t)}$ to $\mathbf{z}^{(0)} \sim q(\mathbf{z}|\mathbf{x})$
- **New objective**:

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}^{(0)}|\mathbf{x})}\left[\ln p(\mathbf{x}|\mathbf{z}^{(T)}) + \sum_{t=1}^{T} \ln \left|\det \frac{\partial \mathbf{f}^{(t)}}{\partial \mathbf{z}^{(t-1)}}\right|\right] - \mathrm{KL}\big(q(\mathbf{z}^{(0)}|\mathbf{x})||p(\mathbf{z}^{(T)})\big).$$

Rezende, D. J., & Mohamed, S. (2015). Variational inference with normalizing flows. arXiv preprint arXiv:1505.05770. ICML 2015

Improving posterior using Normalizing Flows

- Diagonal posterior - **insufficient** and **inflexible**.
- How to get more flexible posterior?
  - ➢ Apply a series of *T* invertible transformations $\mathbf{f}^{(t)}$ to $\mathbf{z}^{(0)} \sim q(\mathbf{z}|\mathbf{x})$
- **New objective**:

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}^{(0)}|\mathbf{x})}\left[\ln p(\mathbf{x}|\mathbf{z}^{(T)}) + \sum_{t=1}^{T} \ln \left|\det \frac{\partial \mathbf{f}^{(t)}}{\partial \mathbf{z}^{(t-1)}}\right|\right] - \mathrm{KL}\left(q(\mathbf{z}^{(0)}|\mathbf{x})||p(\mathbf{z}^{(T)})\right).$$

Jacobian determinant: (i) general normalizing flow (|det J| is **easy** to calculate);

(ii) volume-preserving flow, *i.e.*, |det J| = 1.

Rezende, D. J., & Mohamed, S. (2015). Variational inference with normalizing flows. arXiv preprint arXiv:1505.05770. ICML 2015

Volume-preserving flows

Improving Variational Auto-Encoders using Householder Flow

Jakub M. Tomczak, Max Welling
University of Amsterdam
J.M.Tomczak@uva.nl, M.Welling@uva.nl

Abstract

Variational auto-encoders (VAE) are scalable and powerful generative models. However, the choice of the variational posterior determines tractability and flexibility of the VAE. Commonly, latent variables are modeled using the normal distribution with a diagonal covariance matrix. This results in computational effi-

Tomczak, J. M., & Welling, M. (2016). Improving Variational Inference with Householder Flow. arXiv preprint arXiv:1611.09630. NIPS Workshop on Bayesian Deep Learning 2016

Householder Flow

- How to obtain more **flexible** posterior and preserve |det J|=1?
- Model **full-covariance posterior** using **orthogonal matrices**.
- **Proposition**: Apply a linear transformation:

$$\mathbf{z}^{(1)} = \mathbf{U}\mathbf{z}^{(0)}, \;\; \mathbf{z}^{(1)} \sim \mathcal{N}(\mathbf{U}\mu, \mathbf{U}\ \mathrm{diag}(\sigma^2)\ \mathbf{U}^\top)$$

  and since $\mathbf{U}$ is orthogonal, Jacobian-determinant is 1.

- **Question**: *Is it possible to model an orthogonal matrix efficiently?*

Tomczak, J. M., & Welling, M. (2016). Improving Variational Inference with Householder Flow. arXiv preprint arXiv:1611.09630. NIPS Workshop on Bayesian Deep Learning 2016

Householder Flow

Theorem
Any orthogonal matrix with the basis acting on the K-dimensional subspace can be expressed as a product of exactly K Householder transformations.

Sun, X., & Bischof, C. (1995). A basis-kernel representation of orthogonal matrices. SIAM Journal on Matrix Analysis and Applications, 16(4), 1184-1196.

- **Question**: *Is it possible to model an orthogonal matrix efficiently?* **YES**

Tomczak, J. M., & Welling, M. (2016). Improving Variational Inference with Householder Flow. arXiv preprint arXiv:1611.09630. NIPS Workshop on Bayesian Deep Learning 2016

Householder Flow

In the **Householder transformation** we reflect a vector around a hyperplane defined by a **Householder vector** $\mathbf{v}_t \in \mathbb{R}^M$

$$\mathbf{z}^{(t)} = \underbrace{\left(\mathbf{I} - 2\frac{\mathbf{v}_t\mathbf{v}_t^\top}{||\mathbf{v}_t||^2}\right)}_{Householder\ matrix} \mathbf{z}^{(t-1)} = \mathbf{H}_t\ \mathbf{z}^{(t-1)}.$$

Very efficient: small number of parameters, |J|=1, easy amortization (!).

Tomczak, J. M., & Welling, M. (2016). Improving Variational Inference with Householder Flow. arXiv preprint arXiv:1611.09630. NIPS Workshop on Bayesian Deep Learning 2016

Householder Flow (MNIST)

Method	ELBO
VAE	-93.9
VAE+HF(T=1)	-87.8
VAE+HF(T=10)	**-87.7**
VAE+NICE(T=10)	-88.6
VAE+NICE(T=80)	**-87.2**
VAE+HVI(T=1)	-91.7
VAE+HVI(T=8)	-88.3
VAE+PlanarFlow(T=10)	**-87.5**
VAE+PlanarFlow(T=80)	**-85.1**

Volume-preserving

Non-linear

Tomczak, J. M., & Welling, M. (2016). Improving Variational Inference with Householder Flow. arXiv preprint arXiv:1611.09630. NIPS Workshop on Bayesian Deep Learning 2016

General normalizing flow

Sylvester Normalizing Flows for Variational Inference

Rianne van den Berg[*] University of Amsterdam
Leonard Hasenclever[*] University of Oxford
Jakub M. Tomczak University of Amsterdam
Max Welling University of Amsterdam

Abstract

Variational inference relies on flexible approximate posterior distributions. Normalizing flows provide a general recipe to con-

Variational inference searches for the best posterior approximation within a parametric family of distributions. Hence, the true posterior distribution can only be recovered exactly if it happens to be in the chosen family. In particular, with widely used simple variational families such as diagonal covariance Gaussian distributions

van den Berg, R., Hasenclever, L., Tomczak, J. M., & Welling, M. (2018). Sylvester Normalizing Flows for Variational Inference, UAI 2018 (oral presentation)

Sylvester Flow

- Can we have a **non-linear** flow with a **simple** Jacobian-determinant?
- Let us consider the following normalizing flow:

$$\mathbf{z}^{(t)} = \mathbf{z}^{(t-1)} + \mathbf{A}h(\mathbf{B}\mathbf{z}^{(t)} + \mathbf{b})$$

where A is *D*x*M*, B is *M*x*D*.

- How to calculate the Jacobian-determinant efficiently?

  ➢ **Sylvester's determinant identity**

van den Berg, R., Hasenclever, L., Tomczak, J. M., & Welling, M. (2018). Sylvester Normalizing Flows for Variational Inference, UAI 2018 (oral presentation)

Sylvester Flow

- Can we have a **non-linear** flow with a **simple** Jacobian-determinant?
- Let us consider the following normalizing flow:

$$\mathbf{z}^{(t)} = \mathbf{z}^{(t-1)} + \mathbf{A}h(\mathbf{B}\mathbf{z}^{(t)} + \mathbf{b})$$

where A is *D*x*M*, B is *M*x*D*.

- How to calculate the Jacobian-determinant efficiently?
  - ➢ Sylvester's determinant identity

van den Berg, R., Hasenclever, L., Tomczak, J. M., & Welling, M. (2018). Sylvester Normalizing Flows for Variational Inference, UAI 2018 (oral presentation)

Sylvester Flow

- Can we have a **non-linear** flow with a **simple** Jacobian-determinant?
- Let us consider the following normalizing flow:

$$\mathbf{z}^{(t)} = \mathbf{z}^{(t-1)} + \mathbf{A}h(\mathbf{B}\mathbf{z}^{(t)} + \mathbf{b})$$

where A is *D*x*M*, B is *M*x*D*.

- How to calculate the Jacobian-determinant efficiently?

➢ **Sylvester's determinant identity**

van den Berg, R., Hasenclever, L., Tomczak, J. M., & Welling, M. (2018). Sylvester Normalizing Flows for Variational Inference, UAI 2018 (oral presentation)

Sylvester Flow

Theorem
For all $\mathbf{A} \in \mathbb{R}^{D \times M}, \mathbf{B} \in \mathbb{R}^{M \times D}$

$$\det\left(\mathbf{I}_D + \mathbf{AB}\right) = \det\left(\mathbf{I}_M + \mathbf{BA}\right).$$

- How to calculate the Jacobian-determinant efficiently?

  ➢ **Sylvester's determinant identity**

van den Berg, R., Hasenclever, L., Tomczak, J. M., & Welling, M. (2018). Sylvester Normalizing Flows for Variational Inference, UAI 2018 (oral presentation)

Sylvester Flow

- How to use the Sylvester's determinant identity?

$$\det \frac{\partial \mathbf{z}^{(t)}}{\partial \mathbf{z}^{(t-1)}} = \det \left(\mathbf{I}_M + \operatorname{diag} \left(h'(\mathbf{B}\mathbf{z}^{(t-1)} + \mathbf{b}) \right) \mathbf{B}\mathbf{A} \right)$$

- How to parameterize matrices **A** and **B**?

$$\mathbf{z}^{(t)} = \mathbf{z}^{(t-1)} + \mathbf{Q}\mathbf{R}_1 h(\mathbf{R}_2 \mathbf{Q}^\top \mathbf{z}^{(t-1)} + \mathbf{b})$$

$\mathbf{Q}$ is orthogonal

$\mathbf{R}_1, \mathbf{R}_2$ are triangular

van den Berg, R., Hasenclever, L., Tomczak, J. M., & Welling, M. (2018). Sylvester Normalizing Flows for Variational Inference, UAI 2018 (oral presentation)

Sylvester Flow

- How to use the Sylvester's determinant identity?

$$\det \frac{\partial \mathbf{z}^{(t)}}{\partial \mathbf{z}^{(t-1)}} = \det \left(\mathbf{I}_M + \operatorname{diag} \left(h'(\mathbf{B}\mathbf{z}^{(t-1)} + \mathbf{b}) \right) \mathbf{B}\mathbf{A} \right)$$

- How to parameterize matrices **A** and **B**?

$$\mathbf{z}^{(t)} = \mathbf{z}^{(t-1)} + \mathbf{Q}\mathbf{R}_1 h(\mathbf{R}_2 \mathbf{Q}^\top \mathbf{z}^{(t-1)} + \mathbf{b})$$

$\mathbf{Q}$ is orthogonal — **Householder matrices**, **permutation matrix**, **orthogonalization procedure**

$\mathbf{R}_1, \mathbf{R}_2$ are triangular

van den Berg, R., Hasenclever, L., Tomczak, J. M., & Welling, M. (2018). Sylvester Normalizing Flows for Variational Inference, UAI 2018 (oral presentation)

Sylvester Flow

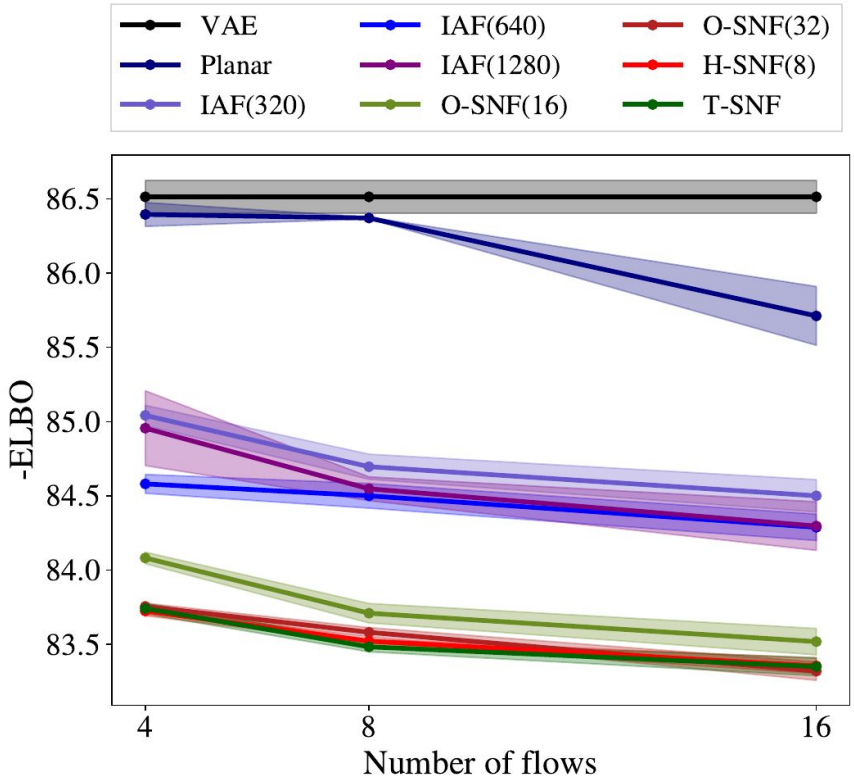
- The Jacobian-determinant:

$$\det\left(\frac{\partial \mathbf{z}^{(t)}}{\partial \mathbf{z}^{(t-1)}}\right) = \det\left(\mathbf{I}_M + \operatorname{diag}\left(h'(\mathbf{R}_2\mathbf{Q}^T\mathbf{z}^{(t-1)} + \mathbf{b})\right)\mathbf{R}_2\mathbf{Q}^T\mathbf{Q}\mathbf{R}_1\right)$$
$$= \det\left(\mathbf{I}_M + \operatorname{diag}\left(h'(\mathbf{R}_2\mathbf{Q}^T\mathbf{z}^{(t-1)} + \mathbf{b})\right)\mathbf{R}_2\mathbf{R}_1\right)$$

- As a result, for properly chosen *h*, the determinant is **upper-triangular** and, thus, **easy to calculate**.

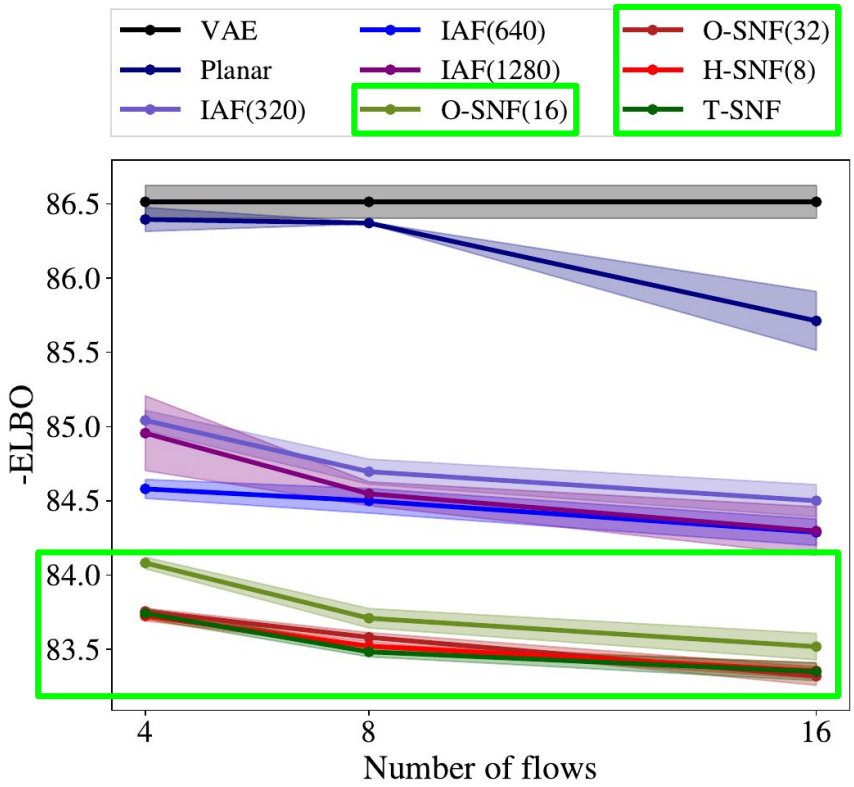
van den Berg, R., Hasenclever, L., Tomczak, J. M., & Welling, M. (2018). Sylvester Normalizing Flows for Variational Inference, UAI 2018 (oral presentation)

Sylvester Flow (MNIST)



van den Berg, R., Hasenclever, L., Tomczak, J. M., & Welling, M. (2018). Sylvester Normalizing Flows for Variational Inference, UAI 2018 (oral presentation)

Sylvester Flow (MNIST)



van den Berg, R., Hasenclever, L., Tomczak, J. M., & Welling, M. (2018). Sylvester Normalizing Flows for Variational Inference, UAI 2018 (oral presentation)

Sylvester Flow

Model	Freyfaces		Omniglot		Caltech 101	
	-ELBO	NLL	-ELBO	NLL	-ELBO	NLL
VAE	4.53 ± 0.02	4.40 ± 0.03	104.28 ± 0.39	97.25 ± 0.23	110.80 ± 0.46	99.62 ± 0.74
Planar	$\mathbf{4.40 \pm 0.06}$	$\mathbf{4.31 \pm 0.06}$	102.65 ± 0.42	96.04 ± 0.28	109.66 ± 0.42	98.53 ± 0.68
IAF	4.47 ± 0.05	4.38 ± 0.04	102.41 ± 0.04	96.08 ± 0.16	111.58 ± 0.38	99.92 ± 0.30
O-SNF	4.51 ± 0.04	4.39 ± 0.05	99.00 ± 0.29	93.82 ± 0.21	106.08 ± 0.39	94.61 ± 0.83
H-SNF	4.46 ± 0.05	4.35 ± 0.05	$\mathbf{99.00 \pm 0.04}$	$\mathbf{93.77 \pm 0.03}$	$\mathbf{104.62 \pm 0.29}$	$\mathbf{93.82 \pm 0.62}$
T-SNF	4.45 ± 0.04	4.35 ± 0.04	99.33 ± 0.23	93.97 ± 0.13	105.29 ± 0.64	94.92 ± 0.73

van den Berg, R., Hasenclever, L., Tomczak, J. M., & Welling, M. (2018). Sylvester Normalizing Flows for Variational Inference, UAI 2018 (oral presentation)

Sylvester Flow

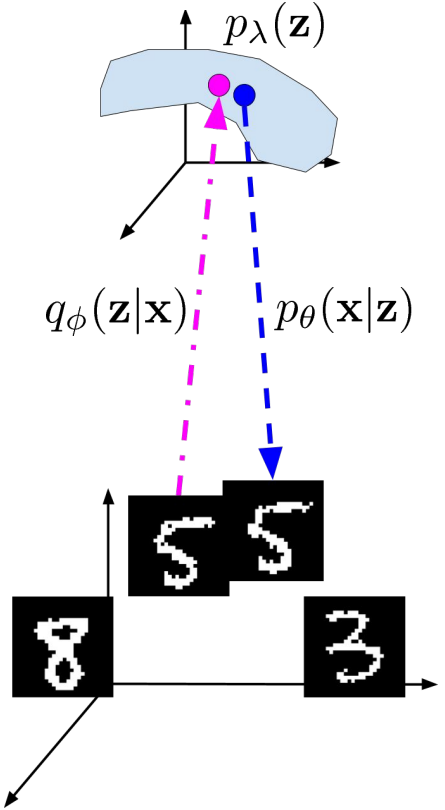
Model	Freyfaces		Omniglot		Caltech 101	
	-ELBO	NLL	-ELBO	NLL	-ELBO	NLL
VAE	4.53 ± 0.02	4.40 ± 0.03	104.28 ± 0.39	97.25 ± 0.23	110.80 ± 0.46	99.62 ± 0.74
Planar	$\mathbf{4.40 \pm 0.06}$	$\mathbf{4.31 \pm 0.06}$	102.65 ± 0.42	96.04 ± 0.28	109.66 ± 0.42	98.53 ± 0.68
IAF	4.47 ± 0.05	4.38 ± 0.04	102.41 ± 0.04	96.08 ± 0.16	111.58 ± 0.38	99.92 ± 0.30
O-SNF	4.51 ± 0.04	4.39 ± 0.05	99.00 ± 0.29	93.82 ± 0.21	106.08 ± 0.39	94.61 ± 0.83
H-SNF	4.46 ± 0.05	4.35 ± 0.05	$\mathbf{99.00 \pm 0.04}$	$\mathbf{93.77 \pm 0.03}$	$\mathbf{104.62 \pm 0.29}$	$\mathbf{93.82 \pm 0.62}$
T-SNF	4.45 ± 0.04	4.35 ± 0.04	99.33 ± 0.23	93.97 ± 0.13	105.29 ± 0.64	94.92 ± 0.73

van den Berg, R., Hasenclever, L., Tomczak, J. M., & Welling, M. (2018). Sylvester Normalizing Flows for Variational Inference, UAI 2018 (oral presentation)

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

VAE with a VampPrior

Jakub M. Tomczak
University of Amsterdam

Max Welling
University of Amsterdam

Abstract

Many different methods to train deep generative models have been introduced in the past. In this paper, we propose to extend the varia-

efficient through the application of the *reparameterization trick* resulting in a highly scalable framework now known as the *variational auto-encoders* (VAE) [19, 33]. Various extensions to deep generative models have been proposed that aim to enrich the variational

Tomczak, J. M., & Welling, M. (2018). VAE with a VampPrior, AISTATS 2018 (oral presentation, 14% of accepted papers)

New Prior

- Let’s re-write the ELBO:

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

Tomczak, J. M., & Welling, M. (2018). VAE with a VampPrior, AISTATS 2018

New Prior

- Let’s re-write the ELBO:

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

Empirical distribution

Tomczak, J. M., & Welling, M. (2018). VAE with a VampPrior, AISTATS 2018

New Prior

- Let's re-write the ELBO:

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

Aggregated posterior

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

Tomczak, J. M., & Welling, M. (2018). VAE with a VampPrior, AISTATS 2018

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:

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

Tomczak, J. M., & Welling, M. (2018). VAE with a VampPrior, AISTATS 2018

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**.

$$p^*_\lambda(\mathbf{z}) = \frac{1}{N}\sum_{n=1}^{N} q_\phi(\mathbf{z}|\mathbf{x}_n)$$

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

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

Tomczak, J. M., & Welling, M. (2018). VAE with a VampPrior, AISTATS 2018

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**.

$$p_\lambda^*(\mathbf{z}) = \frac{1}{N}\sum_{n=1}^{N} q_\phi(\mathbf{z}|\mathbf{x}_n)$$

infeasible

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

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

Tomczak, J. M., & Welling, M. (2018). VAE with a VampPrior, AISTATS 2018

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:

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

Tomczak, J. M., & Welling, M. (2018). VAE with a VampPrior, AISTATS 2018

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**.

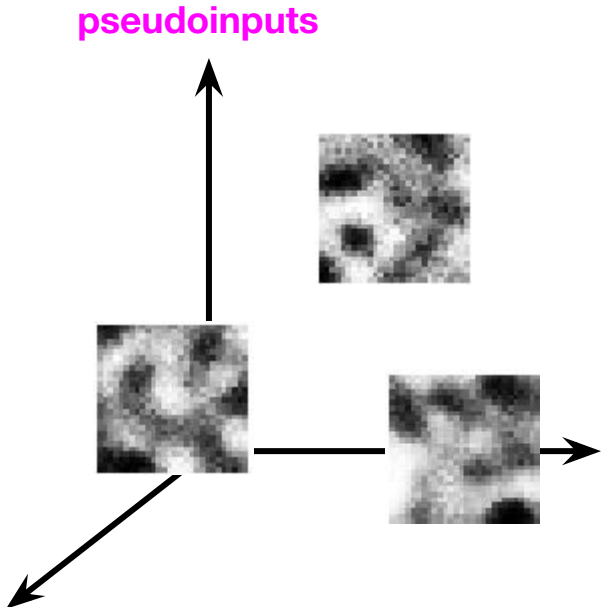
- We approximate it using *K* **pseudo-inputs** instead of *N* observations:

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

they are trained from scratch by SGD

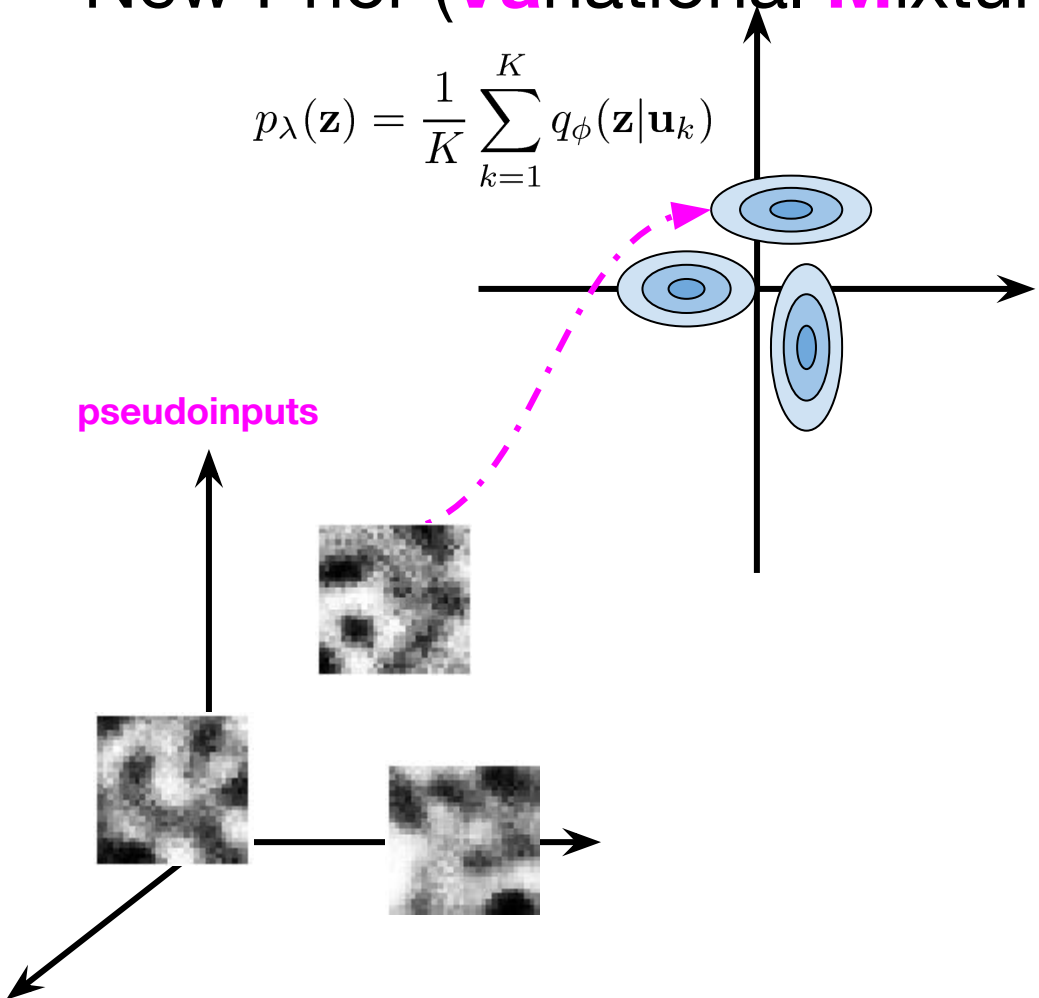
Tomczak, J. M., & Welling, M. (2018). VAE with a VampPrior, AISTATS 2018

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



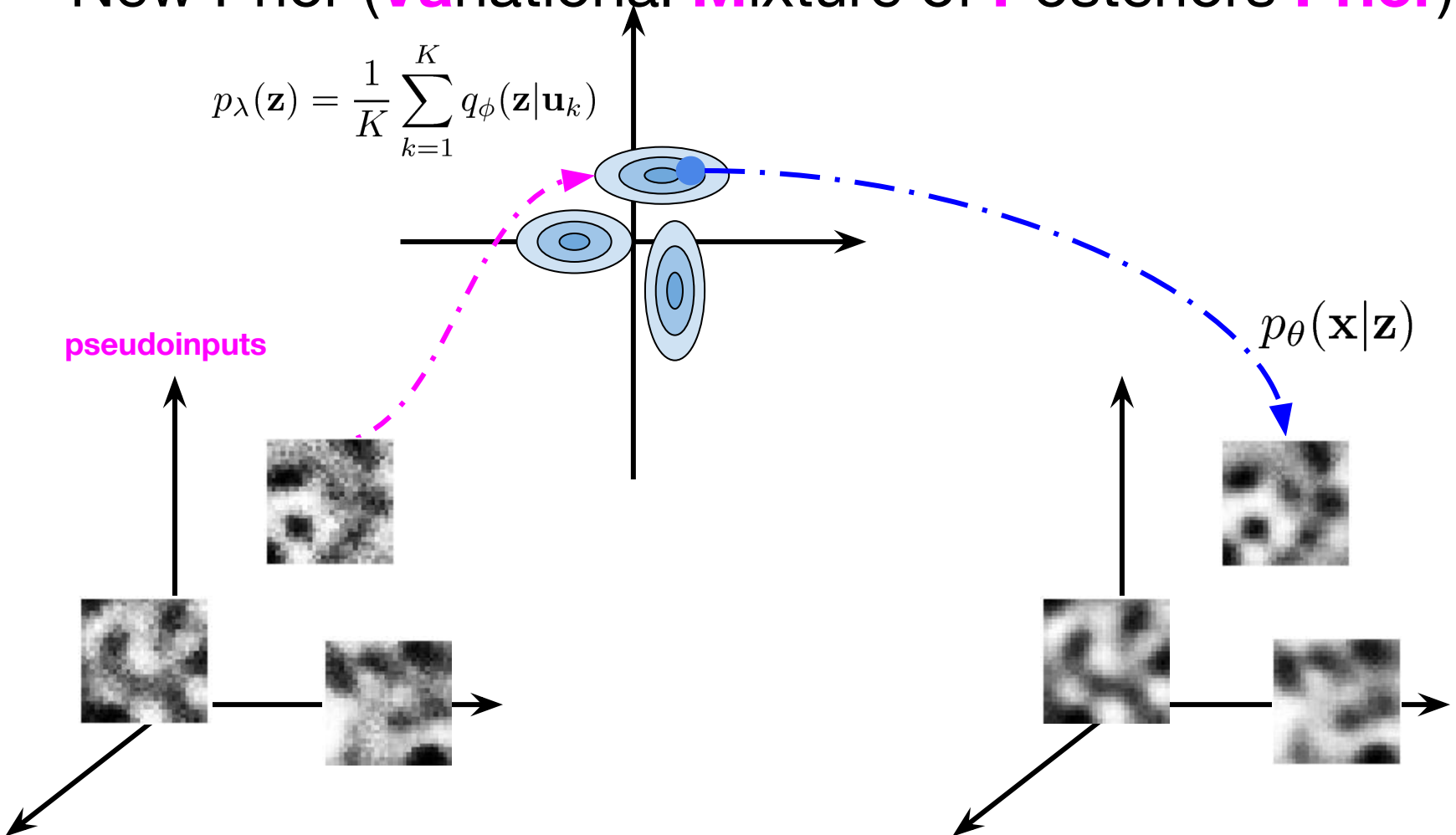
New Prior (Variational Mixture of Posteriors Prior)

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



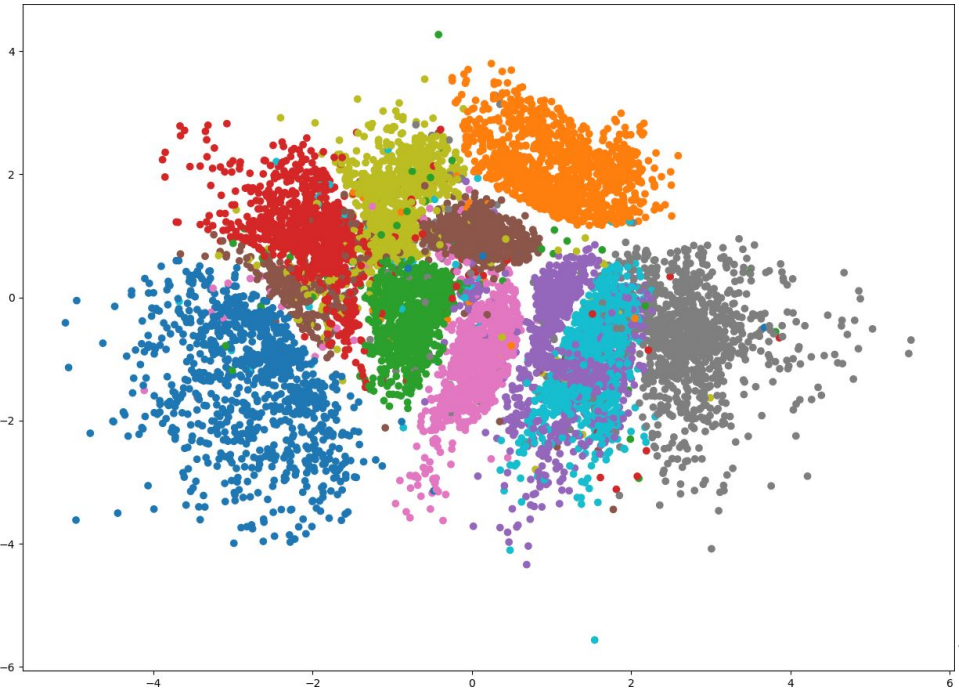
New Prior (Variational Mixture of Posteriors Prior)

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

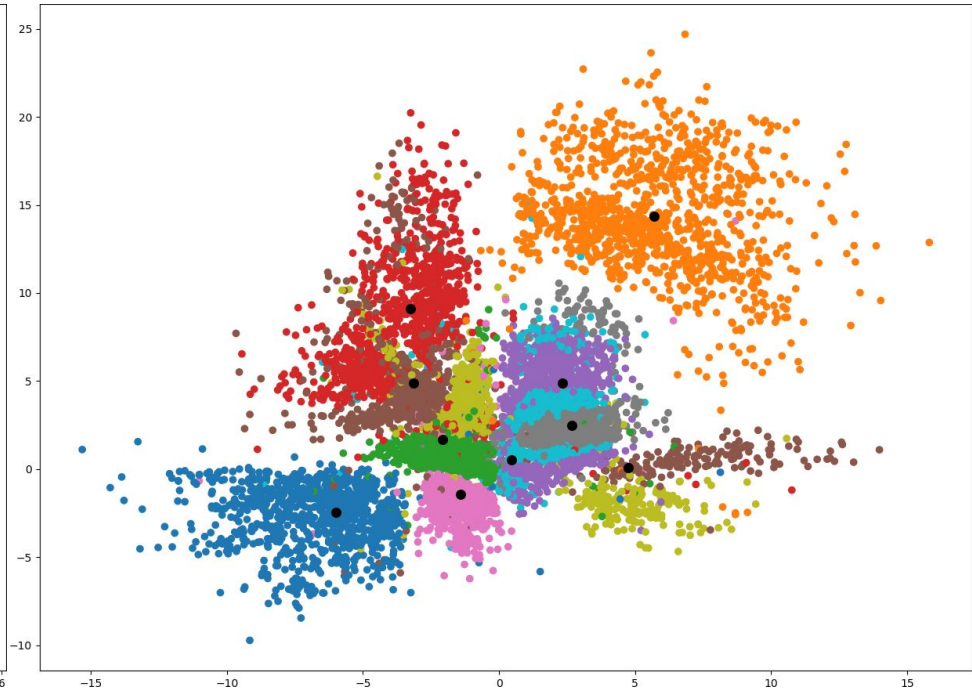


Toy problem (MNIST): VAE with dim(z)=2

Latent space representation + psedoinputs (black dots)



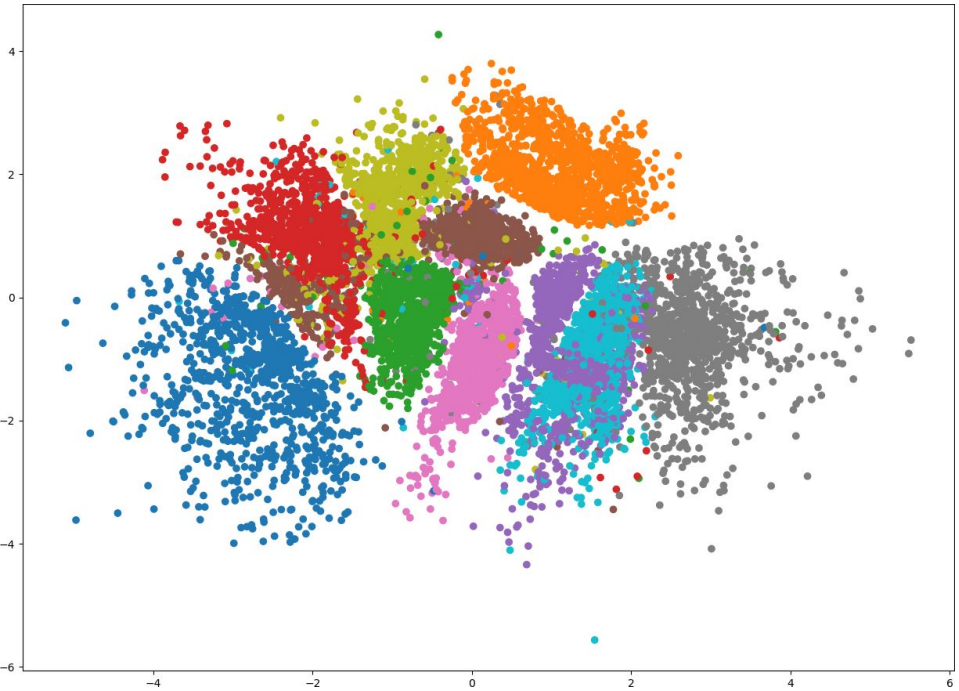
standard



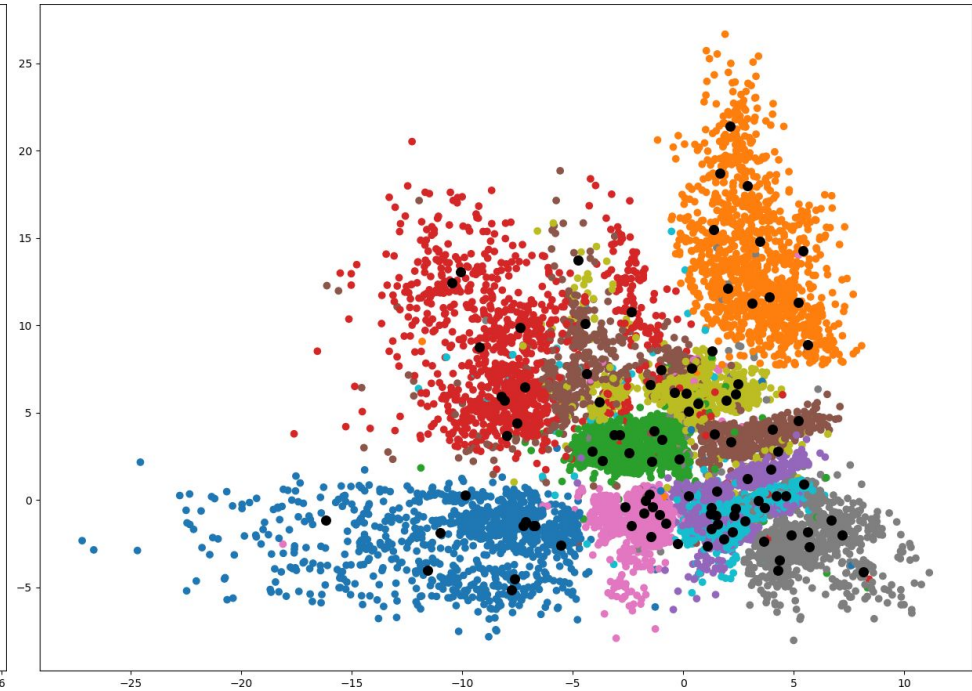
K=10

Toy problem (MNIST): VAE with dim(z)=2

Latent space representation + psedoinputs (black dots)



standard



K=100

Experiments

	VAE ($L = 1$)		HVAE ($L = 2$)		convHVAE ($L = 2$)		PixelHVAE ($L = 2$)	
Dataset	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**

Tomczak, J. M., & Welling, M. (2018). VAE with a VampPrior, AISTATS 2018

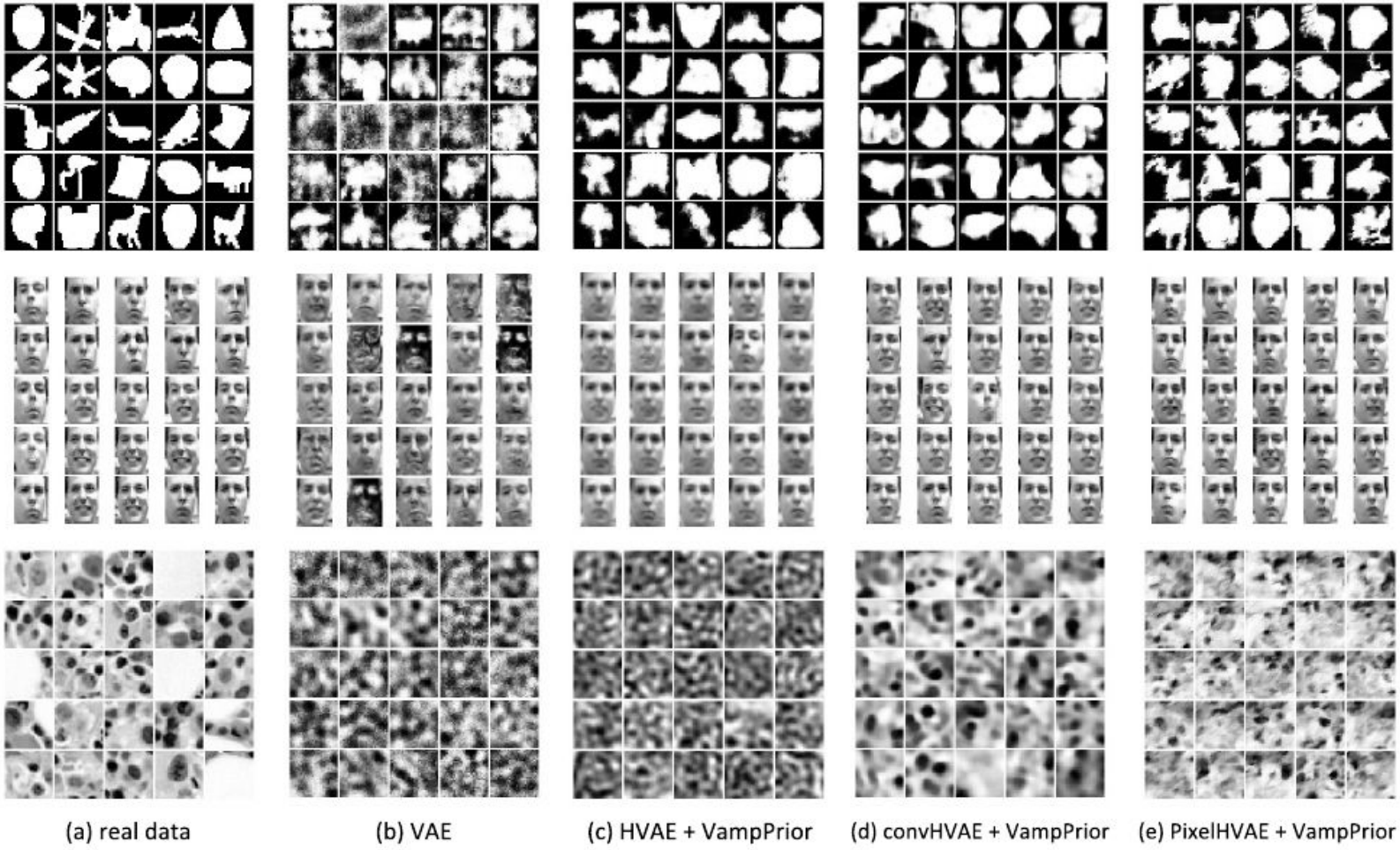
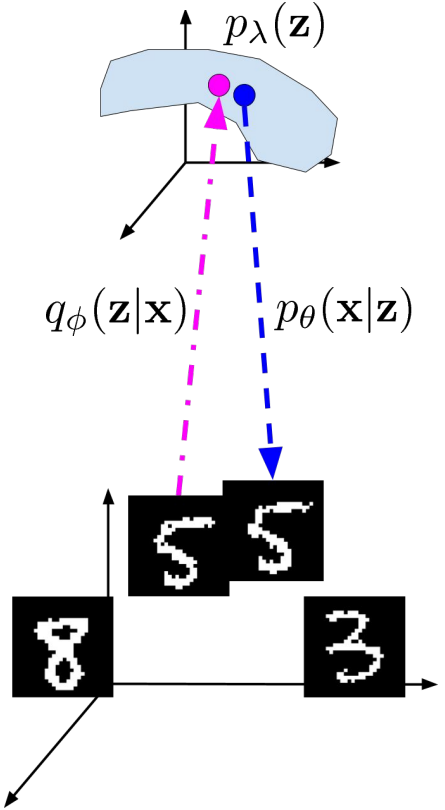


Figure 5: (a) Real images from test sets and images generated by (b) the vanilla VAE, (c) the HVAE ($L = 2$) + VampPrior, (d) the convHVAE ($L = 2$) + VampPrior and (e) the PixelHVAE ($L = 2$) + VampPrior.

Variational Auto-Encoder

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

Fully-connected
ConvNets
PixelCNN
Other



Other projects

Other projects: Non-Gaussian encoder

Hyperspherical Variational Auto-Encoders

Tim R. Davidson* **Luca Falorsi*** **Nicola De Cao*** **Thomas Kipf** **Jakub M. Tomczak**

University of Amsterdam

Abstract

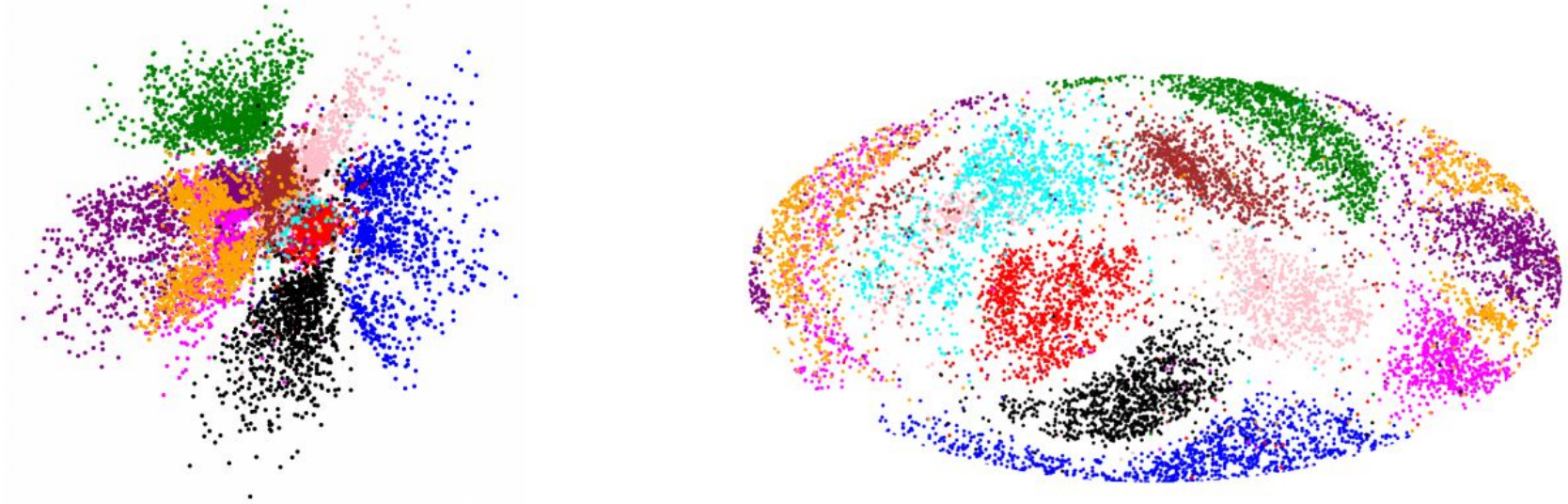
The Variational Auto-Encoder (VAE) is one of the most used unsupervised machine learning models. But although the default choice of a Gaussian distribution for both the prior

Given two hidden units, an autoencoder quickly discovers the latent circle, while a normal VAE becomes highly unstable. This is to be expected as a Gaussian prior is concentrated around the origin, while the KL-divergence tries to reconcile the differences between $\mathcal{S}^1$ and $\mathbb{R}^2$.

The fact that some data types like *directional data* are

Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., & Tomczak, J. M. (2018). Hyperspherical Variational Auto-Encoders. UAI 2018 (oral presentation)

Other projects: Non-Gaussian encoder



(a) $\mathbb{R}^2$ latent space of the $\mathcal{N}$-VAE.

(b) Hammer projection of $\mathcal{S}^2$ latent space of the $\mathcal{S}$-VAE.

Figure 2: Latent space visualization of the 10 MNIST digits in 2 dimensions of both $\mathcal{N}$-VAE (left) and $\mathcal{S}$-VAE (right). (Best viewed in color)

Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., & Tomczak, J. M. (2018). Hyperspherical Variational Auto-Encoders. UAI 2018 (oral presentation)

Other projects: Permutation-invariant operator

Attention-based Deep Multiple Instance Learning

Maximilian Ilse [*1] **Jakub M. Tomczak** [*1] **Max Welling** [1]

Abstract

Multiple instance learning (MIL) is a variation of supervised learning where a single class label is assigned to a bag of instances. In this paper, we state the MIL problem as learning the Bernoulli distribution of the bag label where the bag label probability is fully parameterized by neural networks. Furthermore, we propose a neural network-based permutation-invariant aggregation operator that corresponds to the attention

bag influences the bag label. In other words, it is possible to interpret instances as ROIs. In the medical domain the former task is of great interest because of legal issues[1] and its use in clinical practice. In order to solve the primary task of a bag classification different methods are proposed, such as utilizing similarities among bags (Cheplygina et al., 2015b), embedding instances to a compact low-dimensional representation that is further fed to a bag-level classifier (Andrews et al., 2003; Chen et al., 2006), and combining responses of an instance-level classifier (Ramon & De Raedt, 2000; Raykar et al., 2008; Zhang et al., 2006). Only the

Ilse, M., Tomczak, J. M., & Welling, M. (2018). Attention-based deep multiple instance learning. ICML 2018

Other projects: Permutation-invariant operator

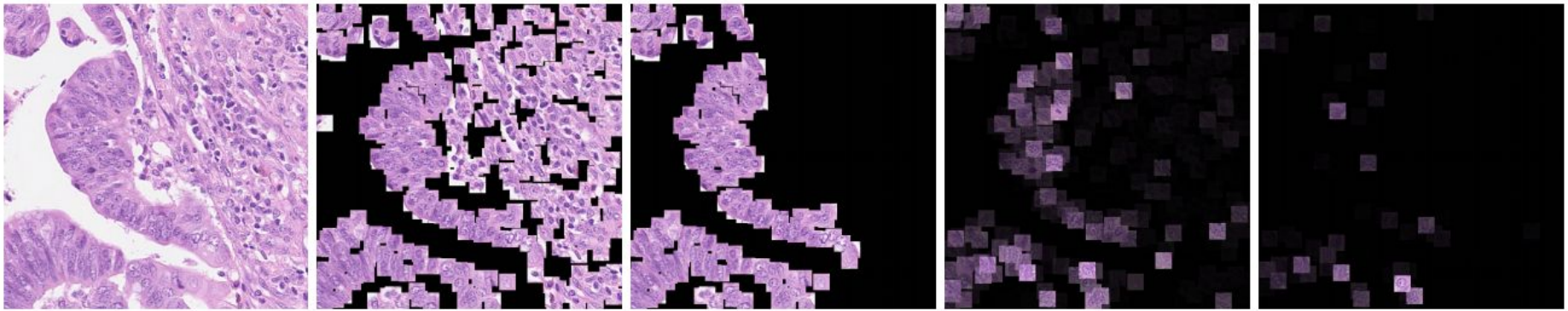


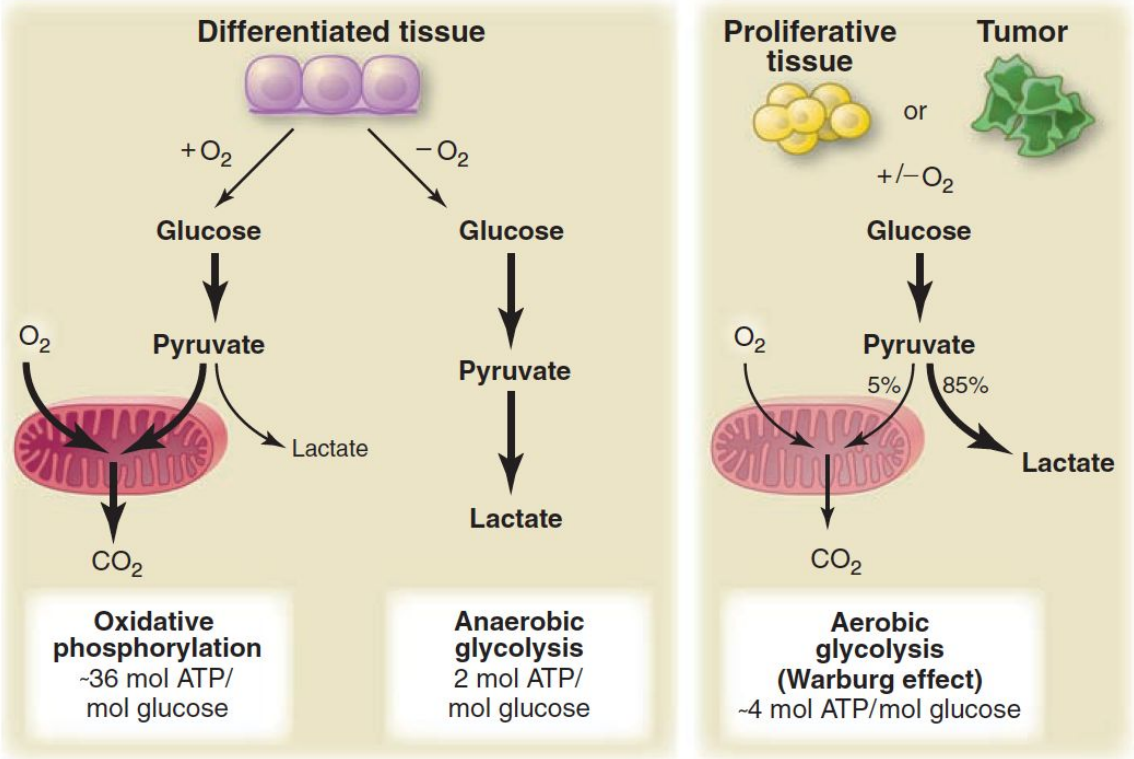
Figure 11. Colon cancer example 2: (a) H&E stained histology image. (b) 27×27 patches centered around all marked nuclei. (c) Ground truth: Patches that belong to the class epithelial. (d) Attention heatmap: Every patch from (b) multiplied by its attention weight. (e) Instance+max heatmap: Every patch from (b) multiplied by its score from the INSTANCE+max model.

Ilse, M., Tomczak, J. M., & Welling, M. (2018). Attention-based deep multiple instance learning. ICML 2018

Prospective project:
Simulator-based Generative Modeling for Aerobic Cancer Glycolysis

Together with **Ewelina Węglarz-Tomczak** (SILS, UVA), **Hans V. Westerhoff** (SILS, UVA), **Max Welling** (AMLAB, UVA)

Prospective project:
Simulator-based Generative Modeling for Aerobic Cancer Glycolysis

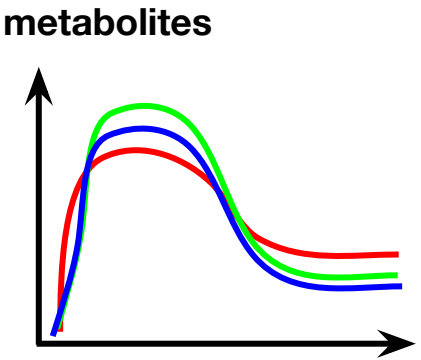


Cancer cells, somewhat counter intuitively, prefer **fermentation** as a source of energy rather than the more efficient **mitochondrial pathway of oxidative phosphorylation** (OxPhos)

Together with **Ewelina Węglarz-Tomczak** (SILS, UVA), **Hans V. Westerhoff** (SILS, UVA), **Max Welling** (AMLAB, UVA)

Prospective project:

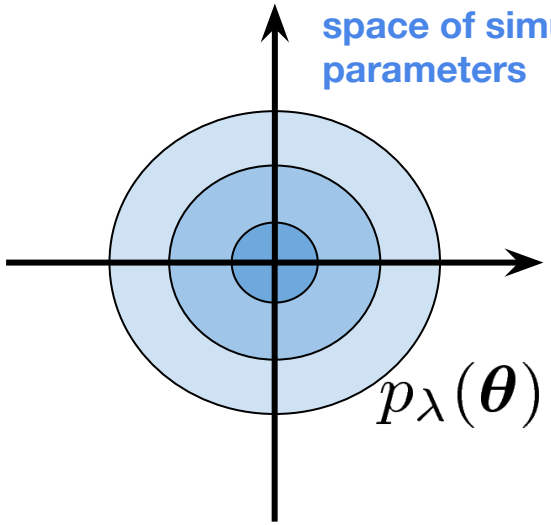
Simulator-based Generative Modeling for Aerobic Cancer Glycolysis



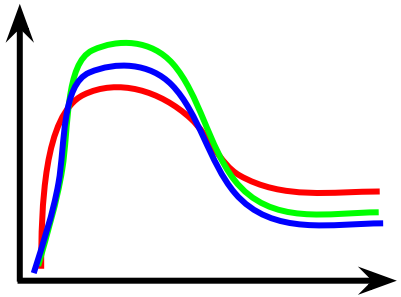
Together with **Ewelina Węglarz-Tomczak** (SILS, UVA), **Hans V. Westerhoff** (SILS, UVA), **Max Welling** (AMLAB, UVA)

Prospective project:

Simulator-based Generative Modeling for Aerobic Cancer Glycolysis



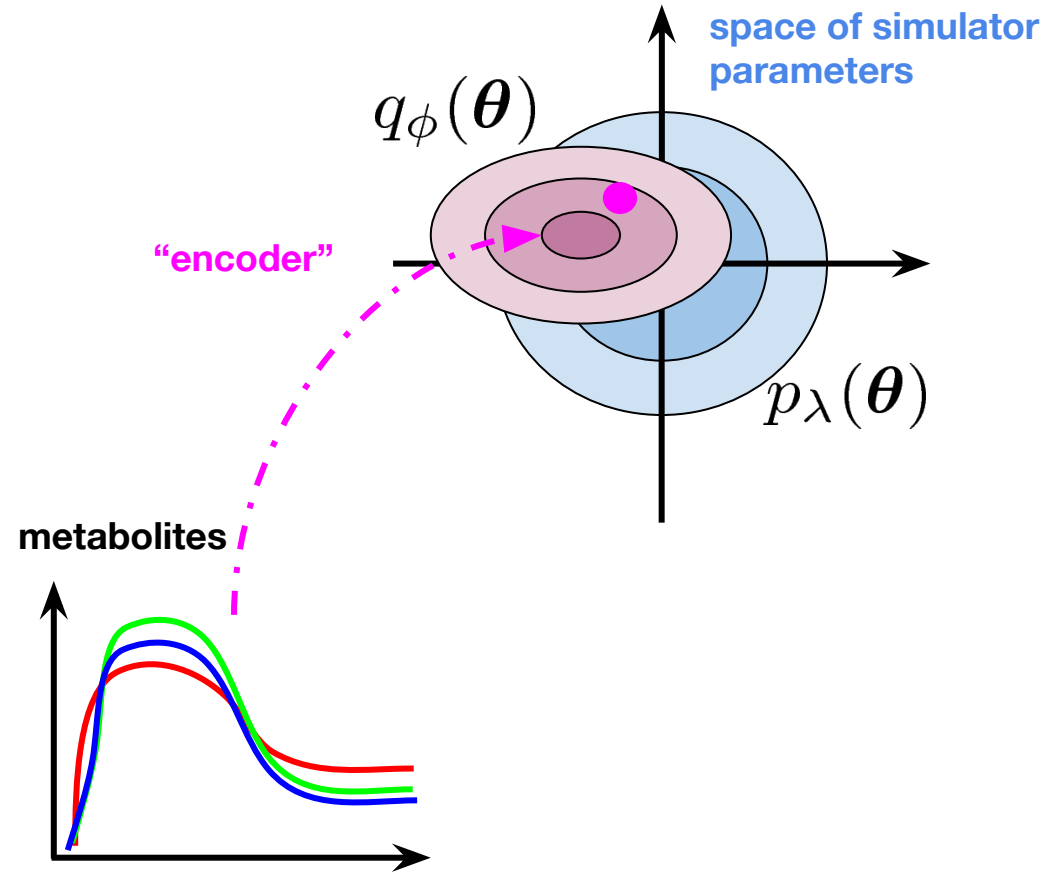
metabolites



Together with **Ewelina Węglarz-Tomczak** (SILS, UVA), **Hans V. Westerhoff** (SILS, UVA), **Max Welling** (AMLAB, UVA)

Prospective project:

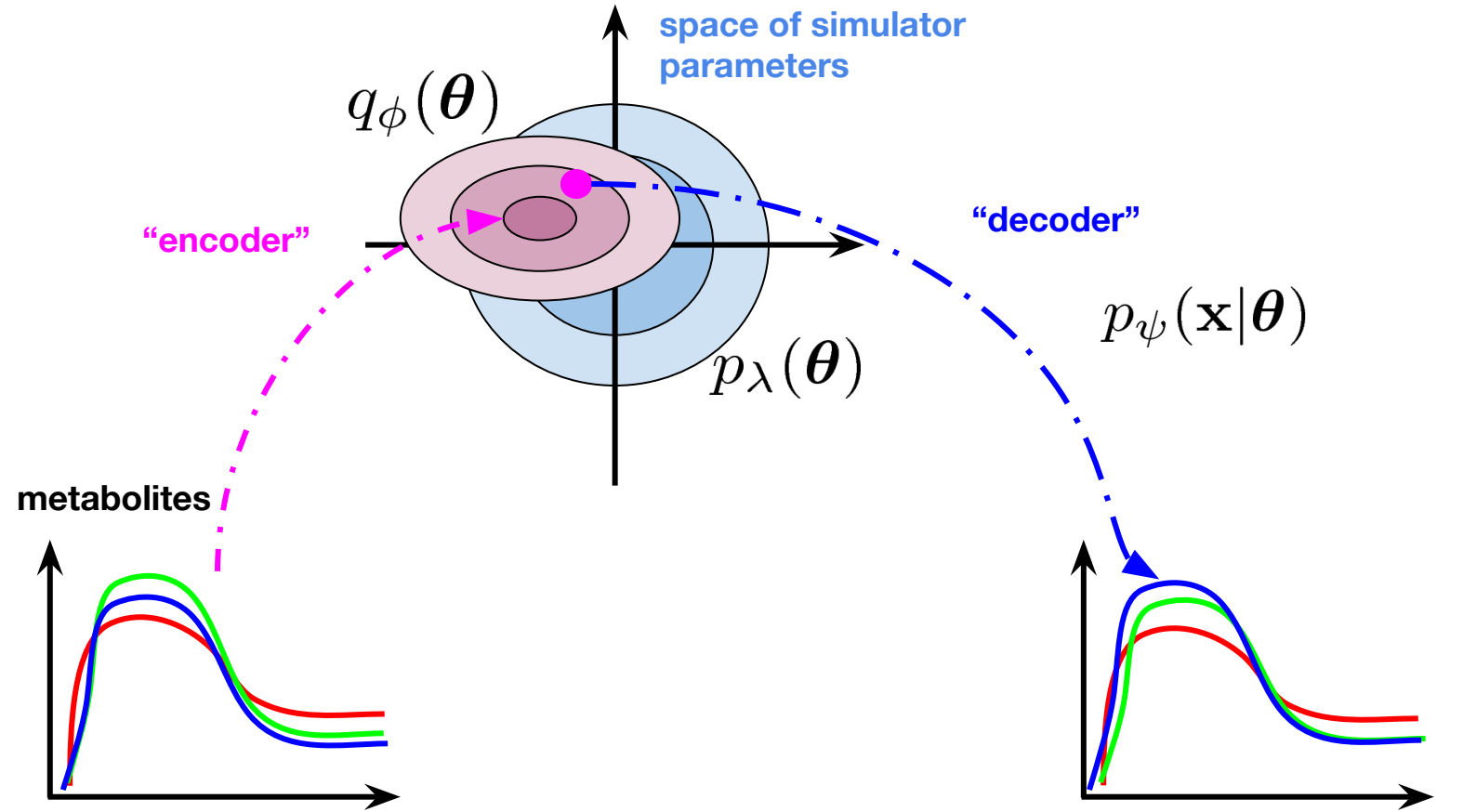
Simulator-based Generative Modeling for Aerobic Cancer Glycolysis



Together with **Ewelina Węglarz-Tomczak** (SILS, UVA), **Hans V. Westerhoff** (SILS, UVA), **Max Welling** (AMLAB, UVA)

Prospective project:

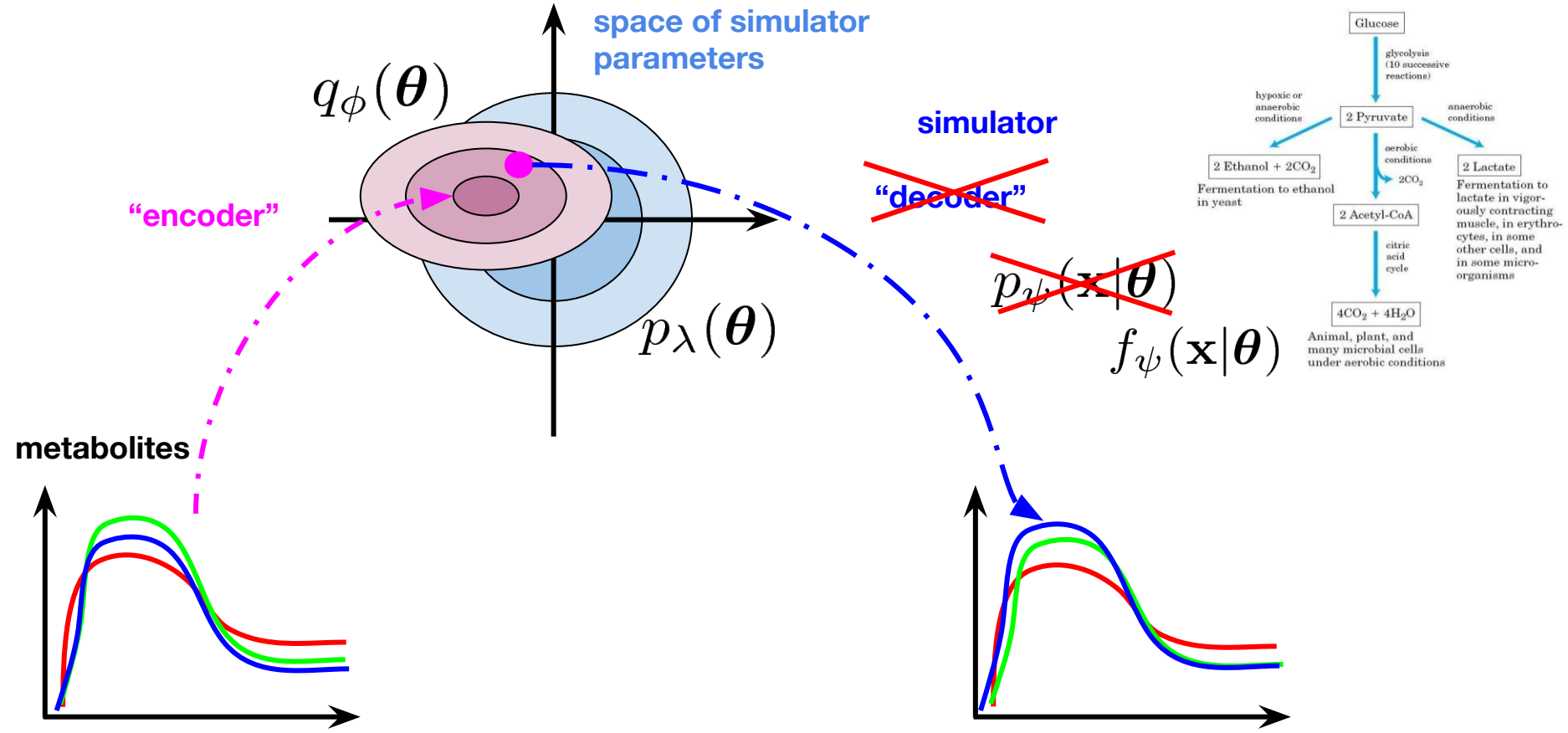
Simulator-based Generative Modeling for Aerobic Cancer Glycolysis



Together with **Ewelina Węglarz-Tomczak** (SILS, UVA), **Hans V. Westerhoff** (SILS, UVA), **Max Welling** (AMLAB, UVA)

Prospective project:

Simulator-based Generative Modeling for Aerobic Cancer Glycolysis



Together with **Ewelina Węglarz-Tomczak** (SILS, UVA), **Hans V. Westerhoff** (SILS, UVA), **Max Welling** (AMLAB, UVA)

Conclusion

Deep Generative Modeling: a way to reason about the reality.

Variational Auto-Encoders: encoder, decoders and **priors**..

Deep Generative Modeling for **image analysis, compression, ...**

Deep Generative Modeling with **simulator-based generators**.

Conclusion

Deep Generative Modeling: a way to reason about the reality.

Variational Auto-Encoders: **encoder, decoders** and **priors**.

Deep Generative Modeling for **image analysis, compression, ...**

Deep Generative Modeling with **simulator-based generators**.

Conclusion

Deep Generative Modeling: a way to reason about the reality.

Variational Auto-Encoders: encoders, decoders and **priors**.

Deep Generative Modeling for **image analysis, compression, ...**

Deep Generative Modeling with **simulator-based generators**.

Conclusion

Deep Generative Modeling: a way to reason about the reality.

Variational Auto-Encoders: encoders, decoders and **priors**.

Deep Generative Modeling for **image analysis**, **compression**, **...**

Deep Generative Modeling with **simulator-based generators**.

Conclusion

Deep Generative Modeling: a way to reason about the reality.

Variational Auto-Encoders: encoders, decoders and **priors**.

Deep Generative Modeling for **image analysis**, **compression**, **...**

Deep Generative Modeling with **simulator-based generators**.

Webpage:
https://jmtomczak.github.io/

Code on github:
https://github.com/jmtomczak/

Contact:
jakubmkt@gmail.com

The research conducted by Jakub M. Tomczak was funded by the European Commission within the Marie Skłodowska-Curie Individual Fellowship (Grant No. 702666, "Deep learning and Bayesian inference for medical imaging").