Introduction to Deep Generative Modeling

Jakub M. Tomczak

VRIJE UNIVERSITEIT AMSTERDAM

BLOG ABOUT DEEP GENERATIVE MODELING

If you are interested in going deeper into deep generative modeling, please take a look at my blog: **[Blog]**

- **Intro**: **[Link]**
- **ARMs**: **[Link]**
- **Flows**: **[Link]**, **[Link]**
- **VAEs**: **[Link]**
- **Hybrid modeling**: TBD

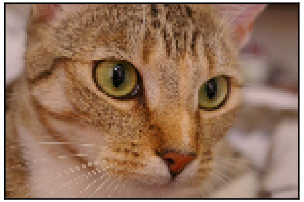
DEEP GENERATIVE MODELING: IS LEARNING CLASSIFIERS ENOUGH?

Let’s assume we have a perfectly trained neural net.

What happens if we add noise to an image?

Let's assume we have a perfectly trained neural net.

What happens if we add noise to an image?

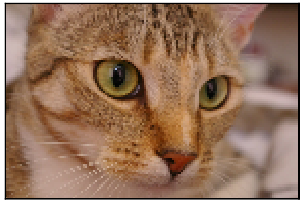


$p(y = cat|\mathbf{x}) = 0.90$
$p(y = dog|\mathbf{x}) = 0.05$
$p(y = horse|\mathbf{x}) = 0.05$

+

noise

=

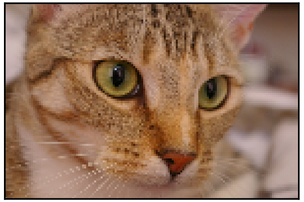


$p(y = cat|\mathbf{x}) = 0.05$
$p(y = dog|\mathbf{x}) = 0.05$
$p(y = horse|\mathbf{x}) = 0.90$

DEEP GENERATIVE MODELING: IS LEARNING CLASSIFIERS ENOUGH?

Let's assume we have a perfectly trained neural net.

What happens if we add noise to an image?

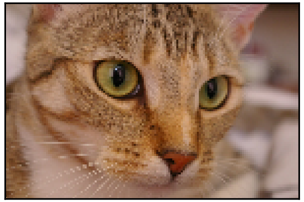


$p(y = cat|\mathbf{x}) = 0.90$
$p(y = dog|\mathbf{x}) = 0.05$
$p(y = horse|\mathbf{x}) = 0.05$

+

noise

=



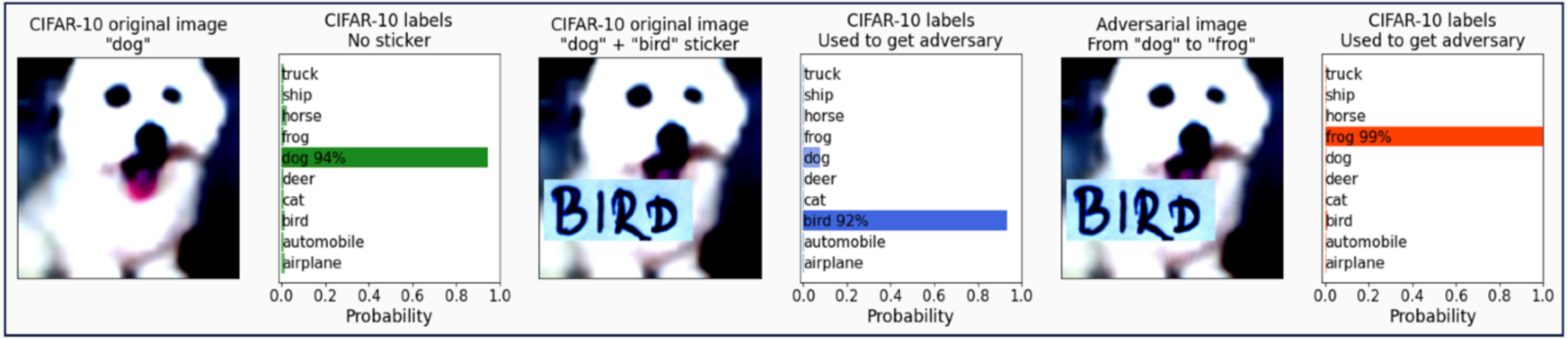
$p(y = cat|\mathbf{x}) = 0.05$
$p(y = dog|\mathbf{x}) = 0.05$
$p(y = horse|\mathbf{x}) = 0.90$

It may fail completely…

DEEP GENERATIVE MODELING: IS LEARNING CLASSIFIERS ENOUGH?

Let's assume we have a perfectly trained neural net.

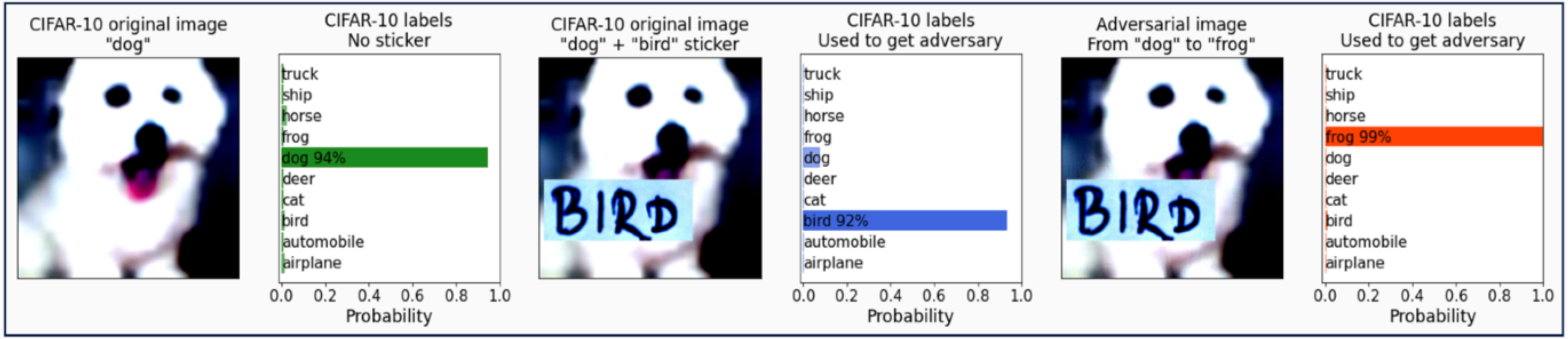
What happens if we add (adversarial) noise to an image?



 S. Fort, "Pixels still beat text: Attacking the OpenAI CLIP model with text patches and adversarial pixel perturbations", [Link]

Let’s assume we have a perfectly trained neural net.

What happens if we add (adversarial) noise to an image?

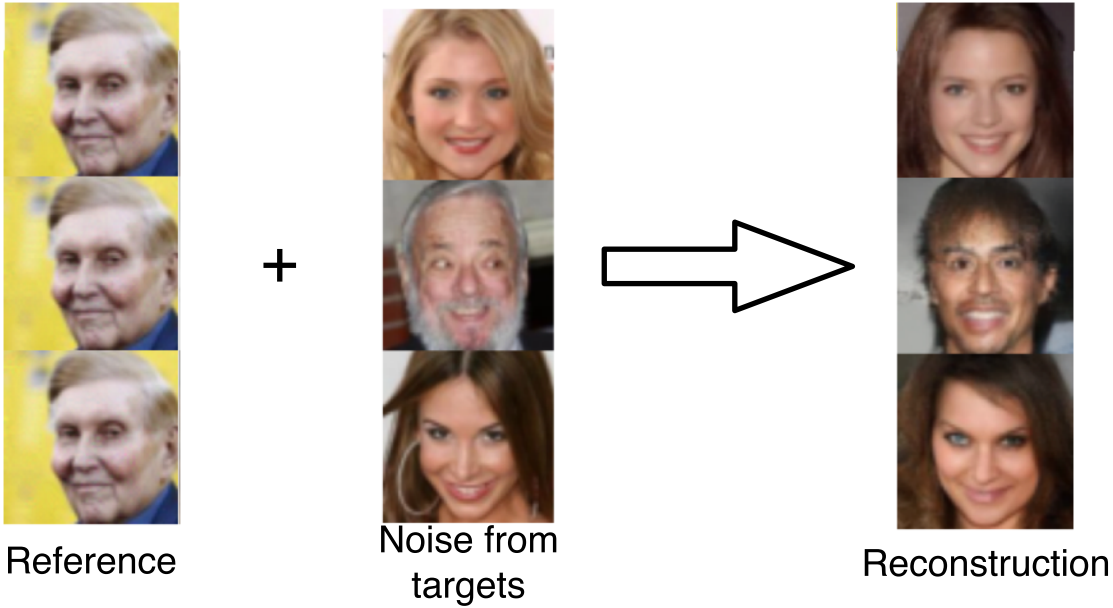


It fails completely…

 S. Fort, “Pixels still beat text: Attacking the OpenAI CLIP model with text patches and adversarial pixel perturbations”, [Link]

Let's assume we have a perfectly trained neural net.

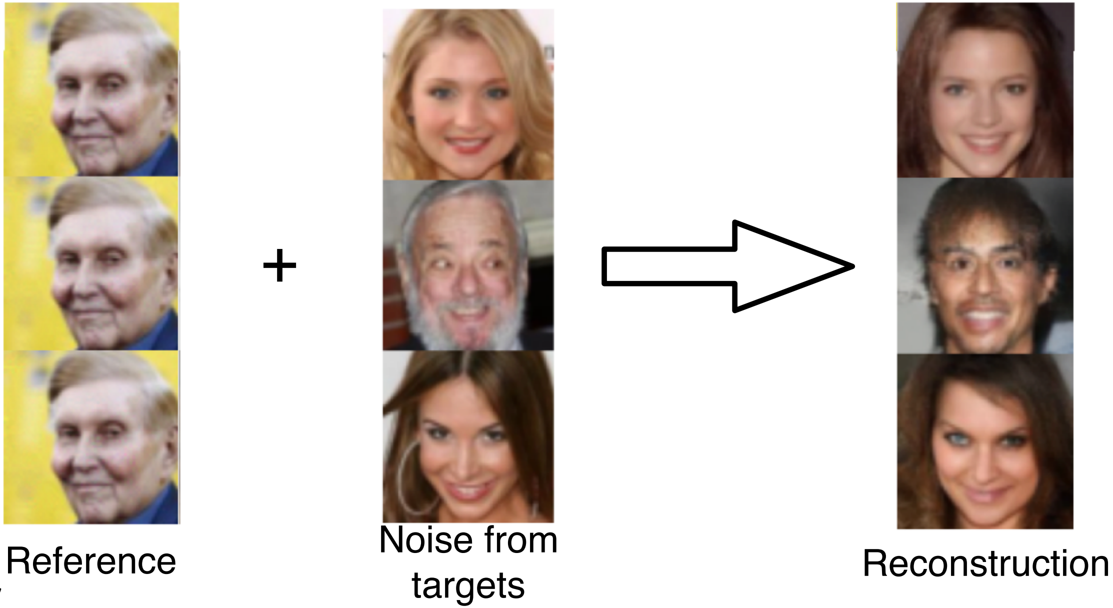
What happens if we add (adversarial) noise to an image?



 A. Kuzina, M. Welling, J.M. Tomczak, "Diagnosing Vulnerability of Variational Auto-Encoders to Adversarial Attacks", [Link]

Let’s assume we have a perfectly trained neural net.

What happens if we add (adversarial) noise to an image?



It fails completely…

 A. Kuzina, M. Welling, J.M. Tomczak, “Diagnosing Vulnerability of Variational Auto-Encoders to Adversarial Attacks”, [Link]

DEEP GENERATIVE MODELING: IS LEARNING CLASSIFIERS ENOUGH?

We clearly see that training a neural network (i.e., a conditional distribution):

$$p(y\,|\,\mathbf{x}) = \mathrm{softmax}\left(NN(\mathbf{x})\right)$$

is **not enough**!

Granny Smith	0.1%
iPod	99.7%
library	0.0%
pizza	0.0%
toaster	0.0%
dough	0.0%

DEEP GENERATIVE MODELING: IS LEARNING CLASSIFIERS ENOUGH?

We clearly see that training a neural network (i.e., a conditional distribution):

$$p(y \mid \mathbf{x}) = \mathrm{softmax}\left(NN(\mathbf{x})\right)$$

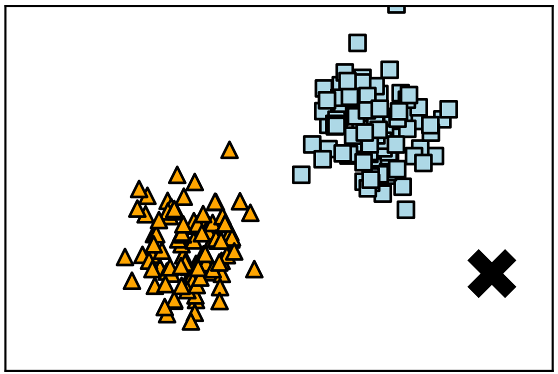
is **not enough**!

What can we do then?

Or, how to modify the **wrong certainty**?

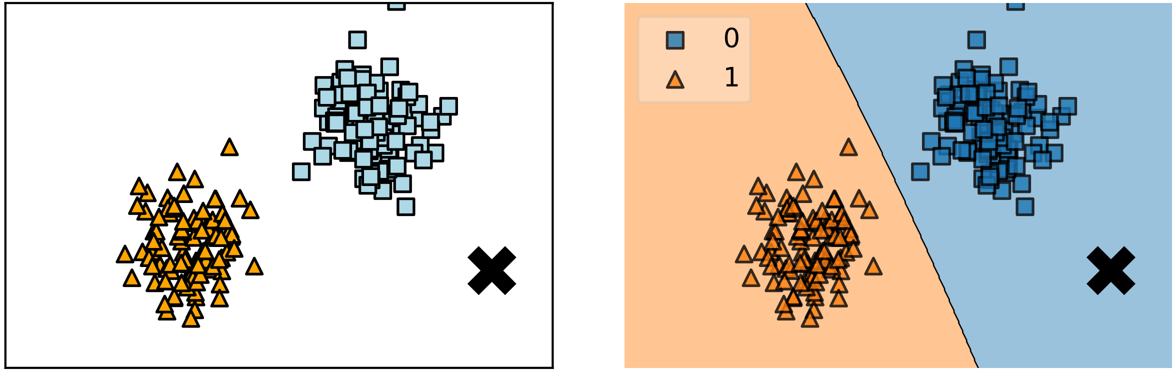
Granny Smith	0.1%
iPod	99.7%
library	0.0%
pizza	0.0%
toaster	0.0%
dough	0.0%

DEEP GENERATIVE MODELING: WHY DO WE NEED THE JOINT DISTRIBUTION?



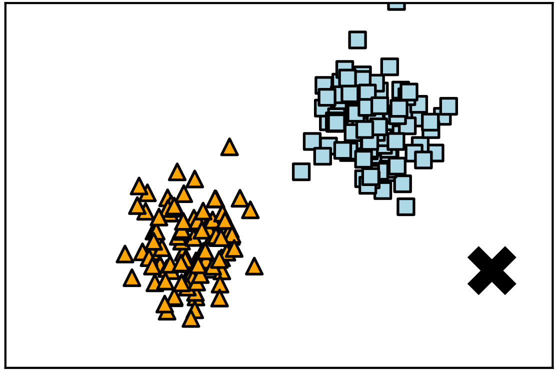
Data

DEEP GENERATIVE MODELING: WHY DO WE NEED THE JOINT DISTRIBUTION?

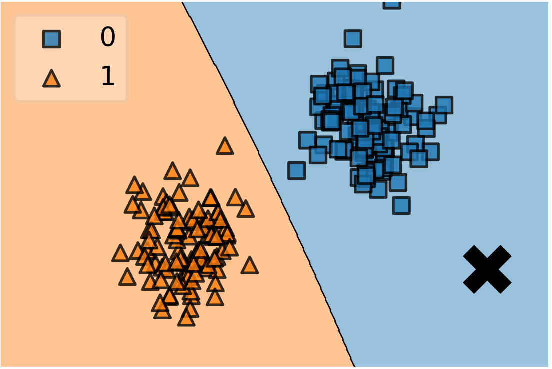


Data $\qquad$ $p(y|\mathbf{x})$

$p(blue|\mathbf{x})$ is high
= certain decision!

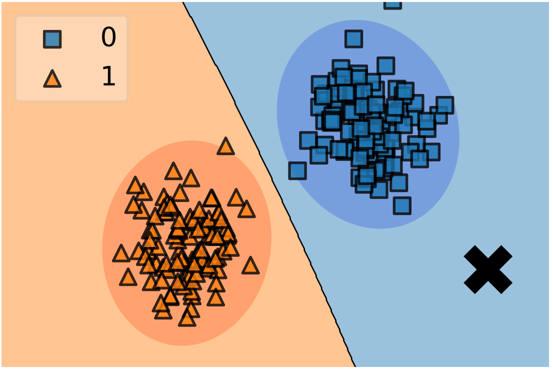


Data



$p(y|\mathbf{x})$

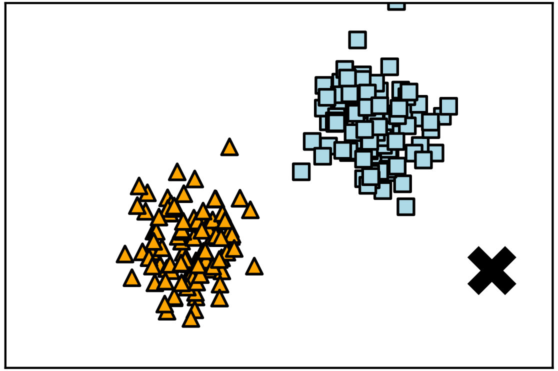
$p(blue|\mathbf{x})$ is high
= certain decision!



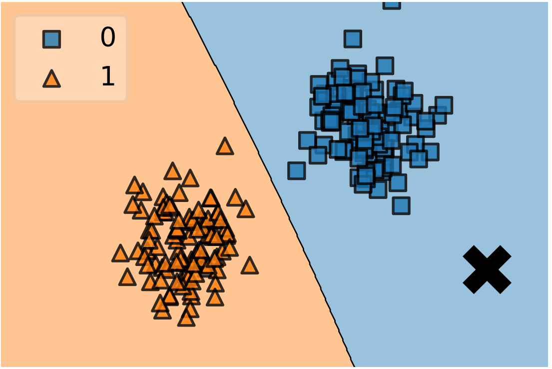
$p(\mathbf{x}, y) = p(y|\mathbf{x})\ p(\mathbf{x})$

$p(blue|\mathbf{x})$ is high
and $p(\mathbf{x})$ is low
= uncertain decision!

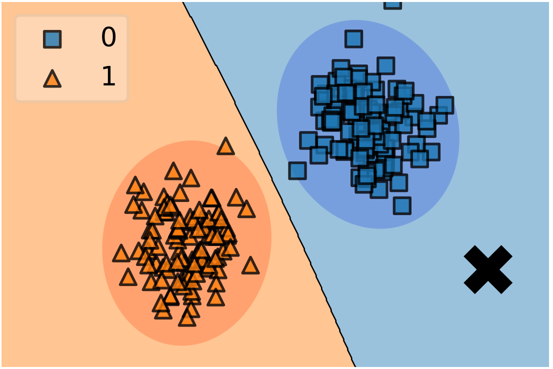
DEEP GENERATIVE MODELING: WHY DO WE NEED THE JOINT DISTRIBUTION?



Data



$p(y|\mathbf{x})$



$p(\mathbf{x}, y) = p(y|\mathbf{x})\ p(\mathbf{x})$

$p(blue|\mathbf{x})$ is high
= certain decision!

$p(blue|\mathbf{x})$ is high
and $p(\mathbf{x})$ is low
= uncertain decision!

Thus, learning the conditional is a part of the story!
How can we learn p(x)?

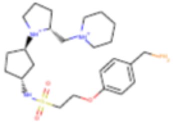
DEEP GENERATIVE MODELING: WHERE CAN WE USE IT?

" 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 .

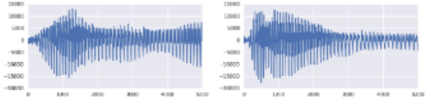
he was silent for a long moment .
he was silent for a moment .
it was quiet for a moment .
it was dark and cold .
there was a pause .
it was my turn .

Text analysis

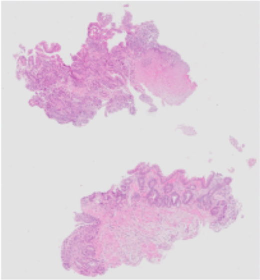
Image analysis



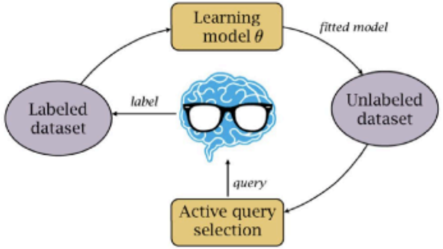
Graph analysis



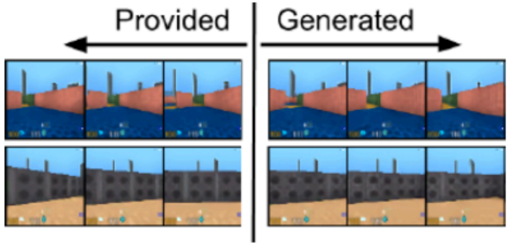
Audio analysis



Medical data



Active Learning



Reinforcement Learning

and more...

DEEP GENERATIVE MODELING: HOW WE CAN FORMULATE IT?

- **Deep Generative Models**
  - **Autoregressive models** (e.g., PixelCNN)
  - **Flow-based models** (e.g., RealNVP)
  - **Latent variable models**
    - **Implicit models** (e.g., GANs)
    - **Prescribed models** (e.g., VAEs)

DEEP GENERATIVE MODELING: HOW WE CAN FORMULATE IT?

Deep Generative Models

+ Energy-based models

- **Autoregressive models** (e.g., PixelCNN)
- **Flow-based models** (e.g., RealNVP)
- **Latent variable models**
  - **Implicit models** (e.g., GANs)
  - **Prescribed models** (e.g., VAEs)

+ Diffusion models

DEEP GENERATIVE MODELING: HOW WE CAN FORMULATE IT?

Generative models	Training	Likelihood	Sampling	Lossy compression	Lossless compression
Autoregressive models	stable	exact	slow	no	yes
Flow-based models	stable	exact	fast/slow	no	yes
Implicit models	unstable	no	fast	no	no
Prescribed model	stable	approximate	fast	yes	no

DEEP GENERATIVE MODELING: HOW WE CAN FORMULATE IT?

- **Deep Generative Models**
  - **Autoregressive models** (e.g., PixelCNN)
  - **Flow-based models** (e.g., RealNVP)
  - **Latent variable models**
    - **Implicit models** (e.g., GANs)
    - **Prescribed models** (e.g., VAEs)

AUTOREGRESSIVE MODELS (ARMS)

We are interested in modeling

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

where $\mathbf{x} \in \{0,1,\ldots,255\}^{D\times 3}$ is an RGB image (for instance).

AUTOREGRESSIVE MODELS (ARMS)

We are interested in modeling

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

where $\mathbf{x} \in \{0,1,...,255\}^{D\times 3}$ is an RGB image (for instance).

We can use the **product rule**:

$$p(\mathbf{x}) = p(x_1)\prod_{d=2}^{D} p(x_d | \mathbf{x}_{<d})$$

where $\mathbf{x}_{<d} = [x_1, x_2, \ldots, x_{d-1}]^\top$

We are interested in modeling

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

where $\mathbf{x} \in \{0,1,\ldots,255\}^{D\times 3}$ is an RGB image (for instance).

We can use the **product rule**:

$$p(\mathbf{x}) = p(x_1) \prod_{d=2}^{D} p(x_d | \mathbf{x}_{<d})$$

where $\mathbf{x}_{<d} = [x_1, x_2, \ldots, x_{d-1}]^\top$

Example:

$$p(\mathbf{x}) = p(x_1)p(x_2 | x_1)p(x_3 | x_1, x_2)$$

We are interested in modeling

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

where $\mathbf{x} \in \{0,1,\ldots,255\}^{D\times 3}$ is an RGB image (for instance).

We can use the **product rule**:

$$p(\mathbf{x}) = p(x_1)\prod_{d=2}^{D} p(x_d | \mathbf{x}_{<d})$$

where $\mathbf{x}_{<d} = [x_1, x_2, \ldots, x_{d-1}]^\top$

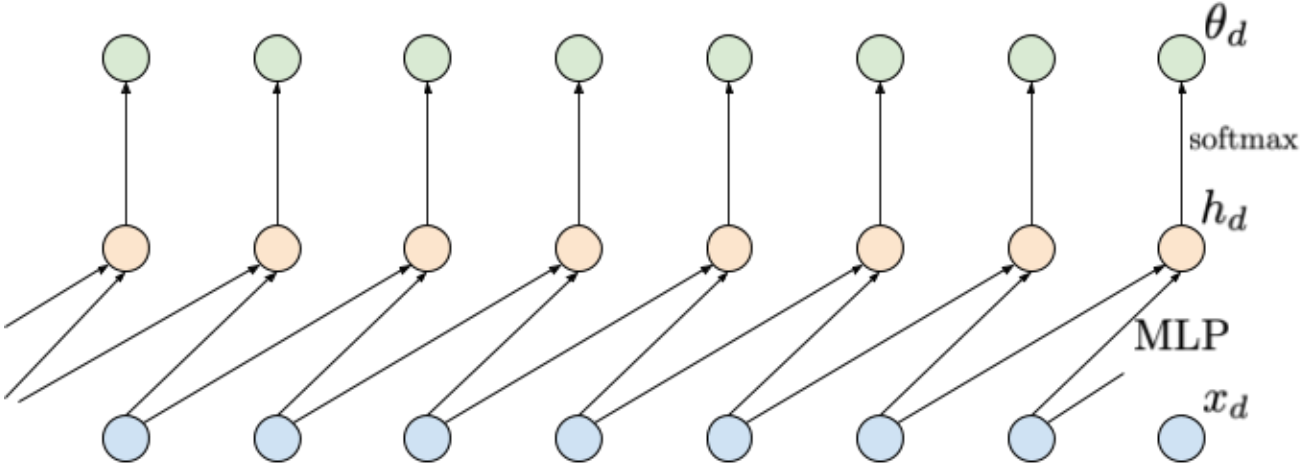
Training objective:

$$\ln p(\mathbf{x}) = \ln p(x_1) + \sum_{d=2}^{D} \ln p(x_d | \mathbf{x}_{<d})$$

Question: How we can model the conditionals $p(x_d | \mathbf{x}_{<d})$ **efficiently**?

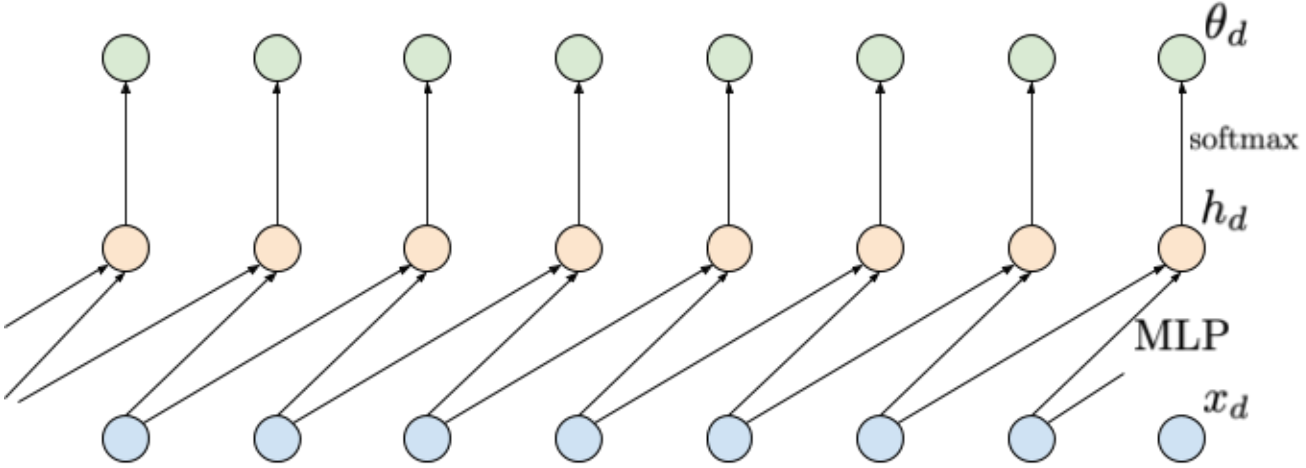
Question: How we can model the conditionals $p(x_d \,|\, \mathbf{x}_{<d})$ **efficiently**?

Approach 1: **Finite memory**



Question: How we can model the conditionals $p(x_d | \mathbf{x}_{<d})$ efficiently?

Approach 1: Finite memory

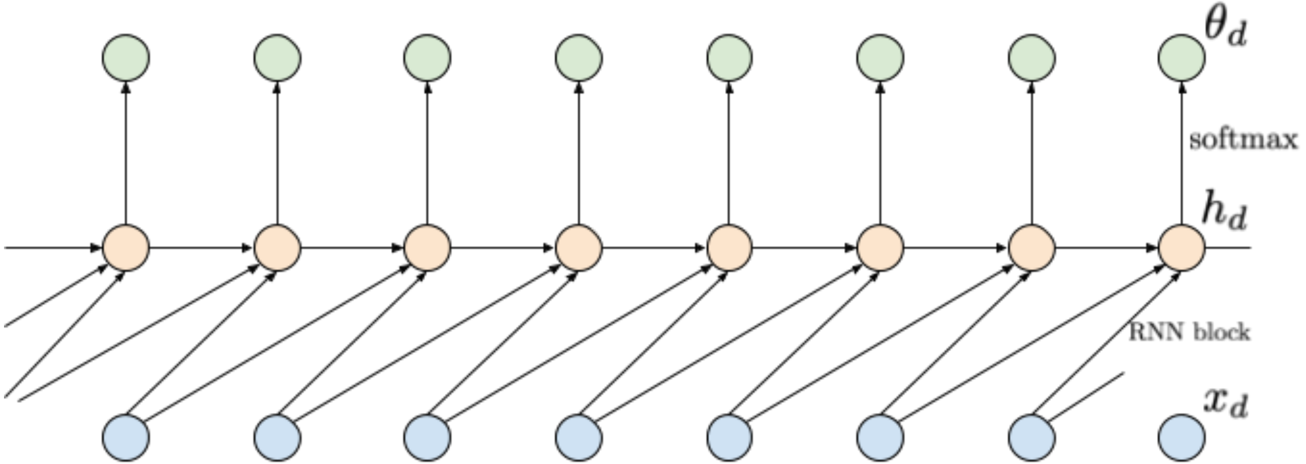


Easy!

Limited dependencies!
How many we should take?

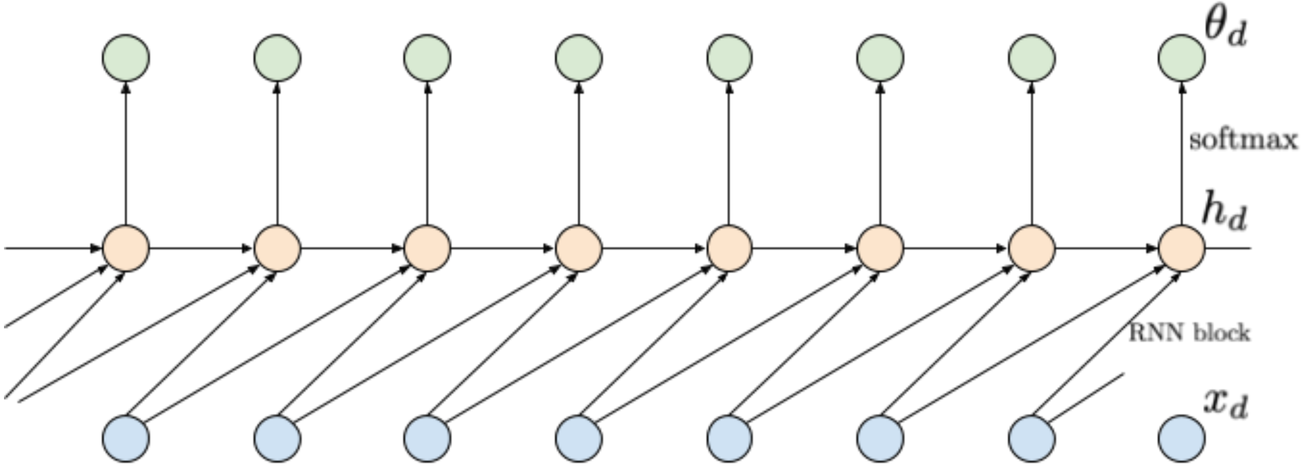
Question: How we can model the conditionals $p(x_d | \mathbf{x}_{<d})$ **efficiently**?

Approach 2: **Long-range memory with RNNs**



Question: How we can model the conditionals $p(x_d | \mathbf{x}_{<d})$ **efficiently**?

Approach 2: **Long-range memory with RNNs**

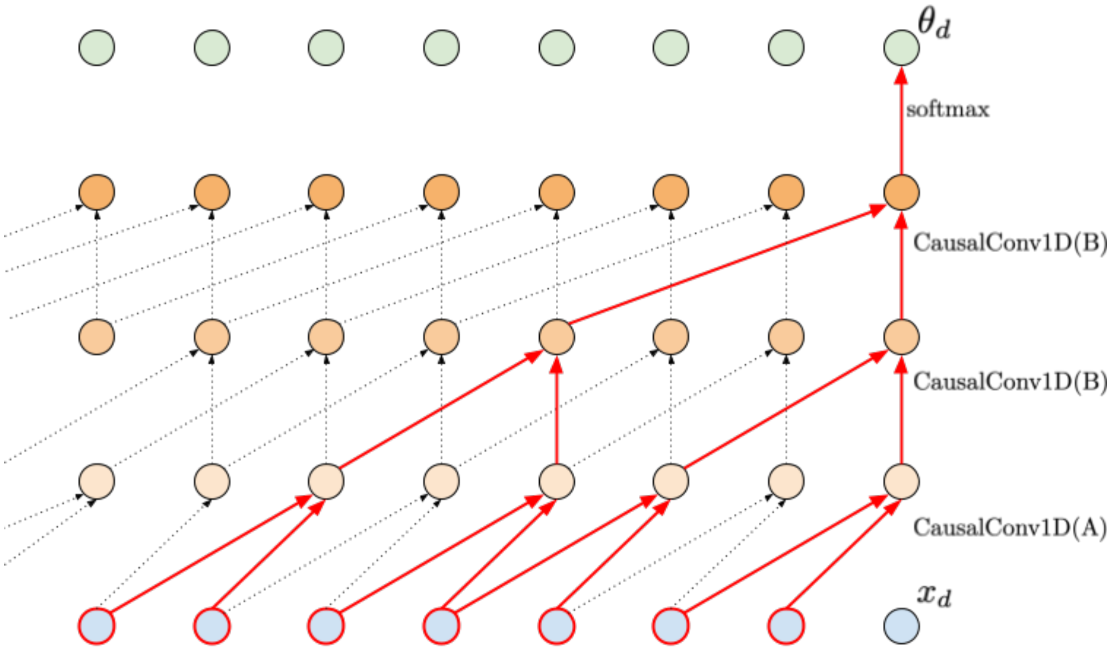


Easy!
Long-range dependencies!

Sequential -> slow
Vanishing gradient problem

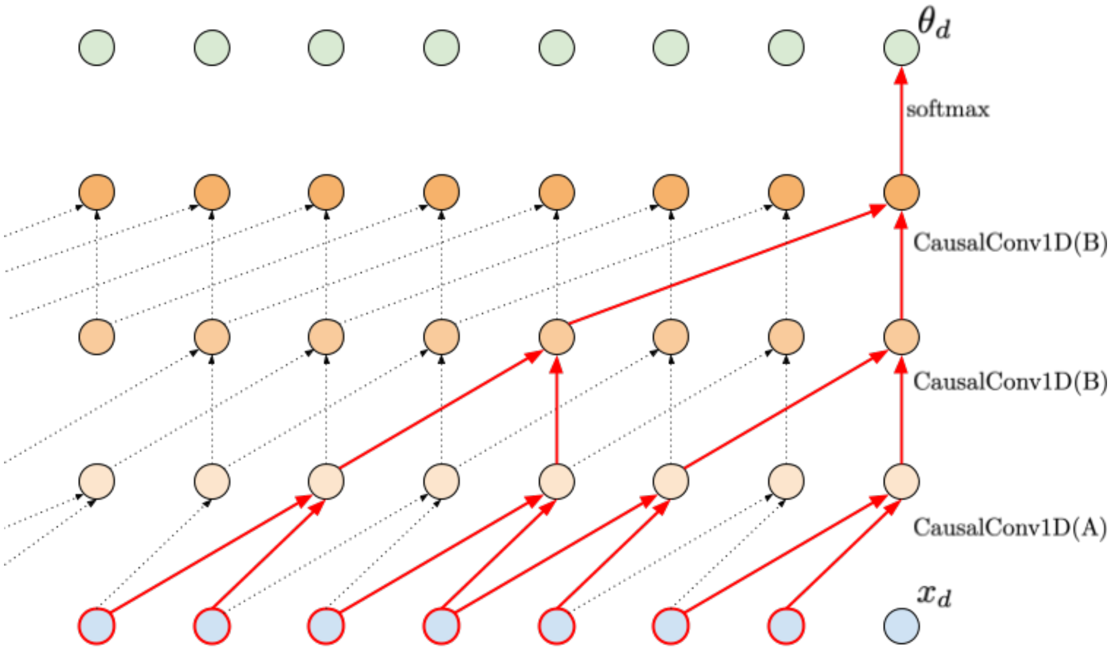
Question: How we can model the conditionals $p(x_d \mid \mathbf{x}_{<d})$ **efficiently**?

Approach 3: **Long-range memory with CNNs**



Question: How we can model the conditionals $p(x_d | \mathbf{x}_{<d})$ **efficiently**?

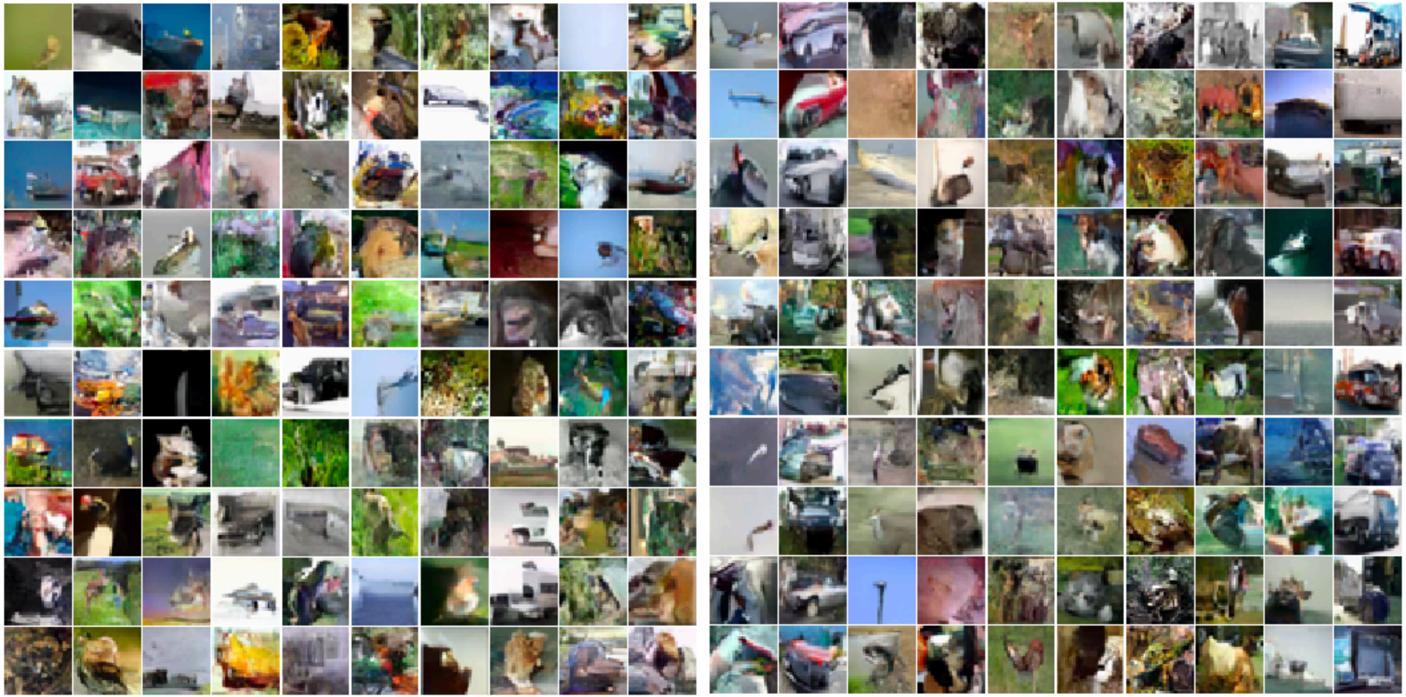
Approach 3: **Long-range memory with CNNs**



Easy!
Long-range dependencies!
No training issues!

Slow generation

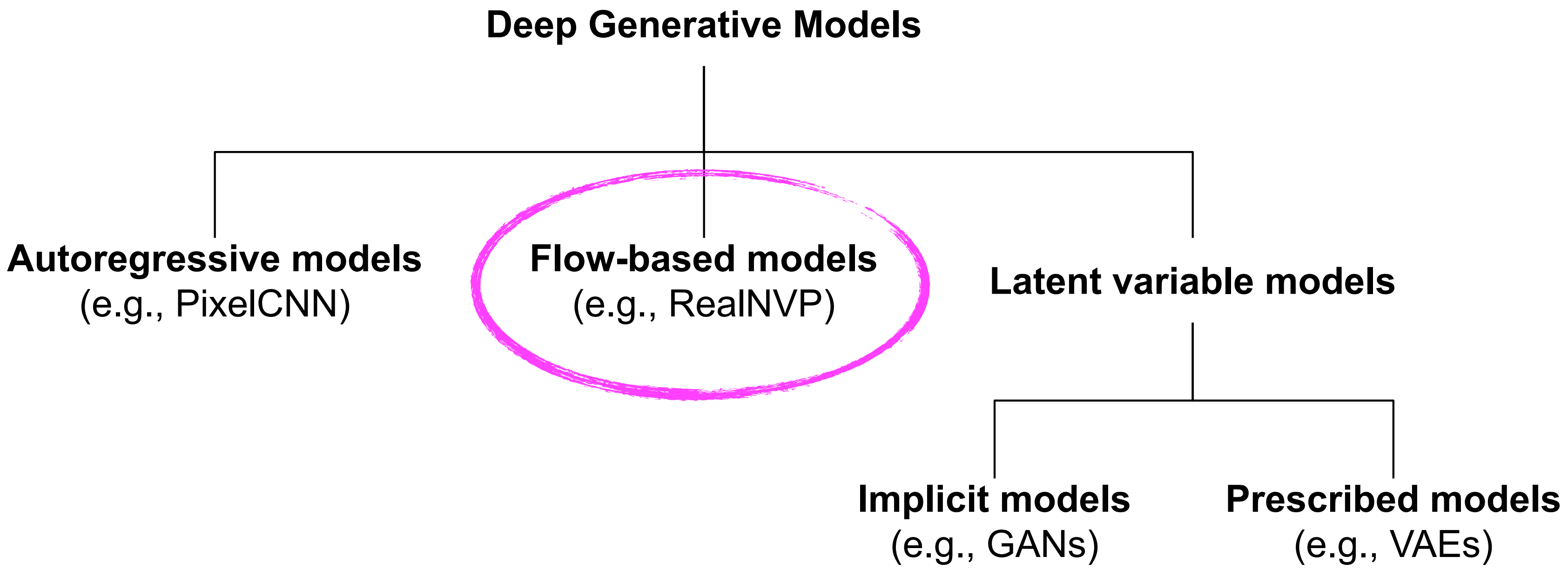
AUTOREGRESSIVE MODELS (ARMS)



Samples from a PixelCNN

 Chen, Xi, et al. "Pixelsnail: An improved autoregressive generative model." *ICML 2018*

DEEP GENERATIVE MODELING: HOW WE CAN FORMULATE IT?



Let us consider a simple example.

FLOWS (FLOW-BASED MODELS)

Let us consider a simple example.

We have a random variable $z \in \mathbb{R}$ with $\pi(z) = \mathcal{N}(z | 0,1)$.

We are interested in a distribution of $x = 0.75z + 1$.

Let us consider a simple example.

We have a random variable $z \in \mathbb{R}$ with $\pi(z) = \mathcal{N}(z|0,1)$.

We are interested in a distribution of $x = 0.75z + 1$.

What is the answer?

Let us consider a simple example.

We have a random variable $z \in \mathbb{R}$ with $\pi(z) = \mathcal{N}(z \,|\, 0,1)$.

We are interested in a distribution of $x = 0.75z + 1$.

What is the answer? $\mathcal{N}(x \,|\, 1,0.75)$!

FLOWS (FLOW-BASED MODELS)

Let us consider a simple example.

We have a random variable $z \in \mathbb{R}$ with $\pi(z) = \mathcal{N}(z|0,1)$.

We are interested in a distribution of $x = 0.75z + 1$.

What is the answer? $\mathcal{N}(x|1,0.75)$!

How can we calculate that? Through the **change of variables formula**:

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

Let us consider a simple example.

We have a random variable $z \in \mathbb{R}$ with $\pi(z) = \mathcal{N}(z | 0,1)$.

We are interested in a distribution of $x = 0.75z + 1$.

What is the answer? $\mathcal{N}(x | 1,0.75)$!

How can we calculate that? Through the **change of variables formula**:

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

Change of volume (Jacobian)

Invertible function

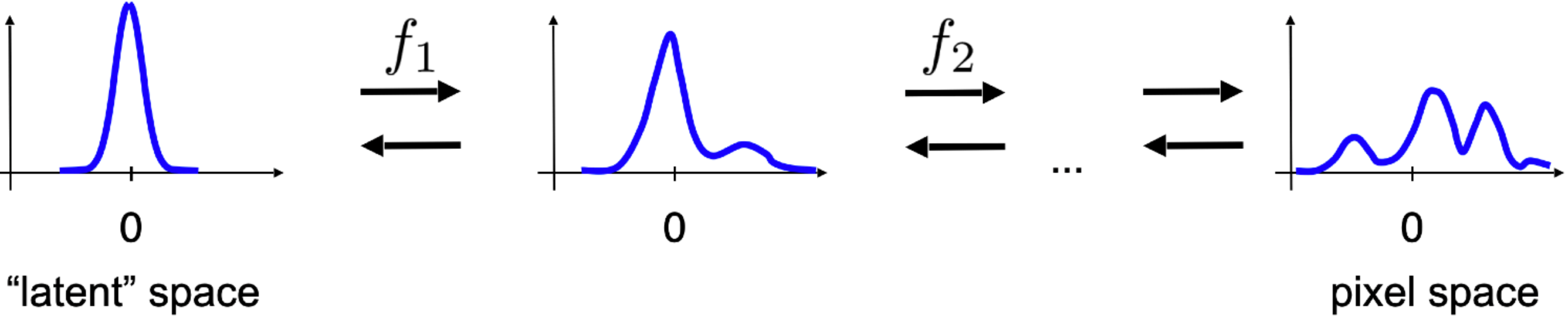
FLOWS (FLOW-BASED MODELS)

We change a random variable **x** to another random variable **z** using **invertible** transformations, $\mathbf{x}, \mathbf{z} \in \mathbb{R}^D$:

$$p(\mathbf{x}) = \pi\left(\mathbf{z}_0 = f^{-1}(\mathbf{x})\right) \prod_{i=1}^{K} \left| \mathbf{J}_{f_i}(z_{i-1}) \right|^{-1}$$

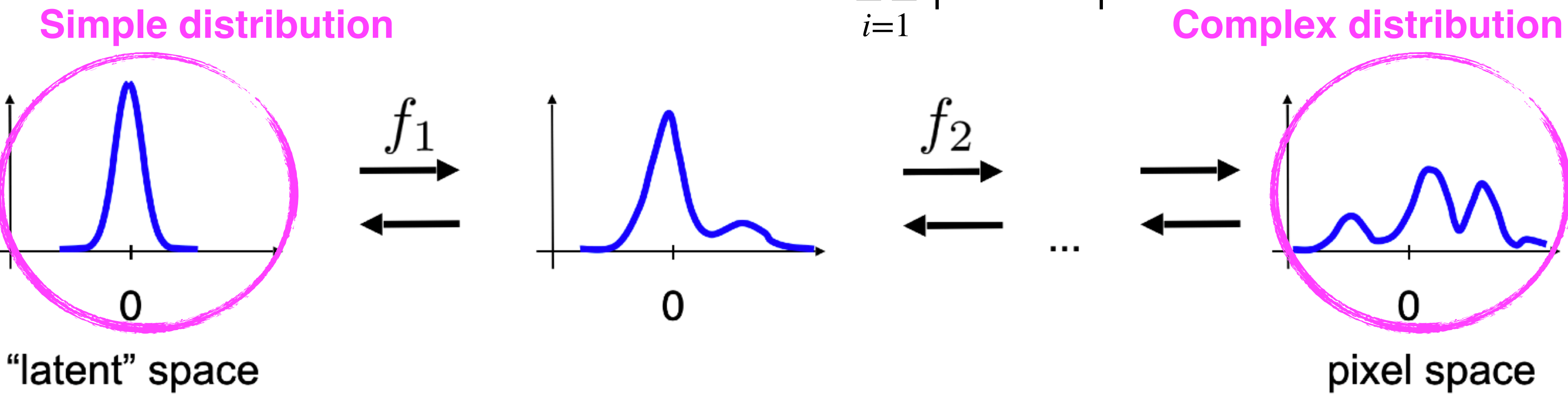
We change a random variable **x** to another random variable **z** using **invertible** transformations, $\mathbf{x}, \mathbf{z} \in \mathbb{R}^D$:

$$p(\mathbf{x}) = \pi\left(\mathbf{z}_0 = f^{-1}(\mathbf{x})\right) \prod_{i=1}^{K} \left| \mathbf{J}_{f_i}(z_{i-1}) \right|^{-1}$$



We change a random variable **x** to another random variable **z** using **invertible** transformations, $\mathbf{x}, \mathbf{z} \in \mathbb{R}^D$:

$$p(\mathbf{x}) = \pi\left(\mathbf{z}_0 = f^{-1}(\mathbf{x})\right) \prod_{i=1}^{K} \left| \mathbf{J}_{f_i}(z_{i-1}) \right|^{-1}$$

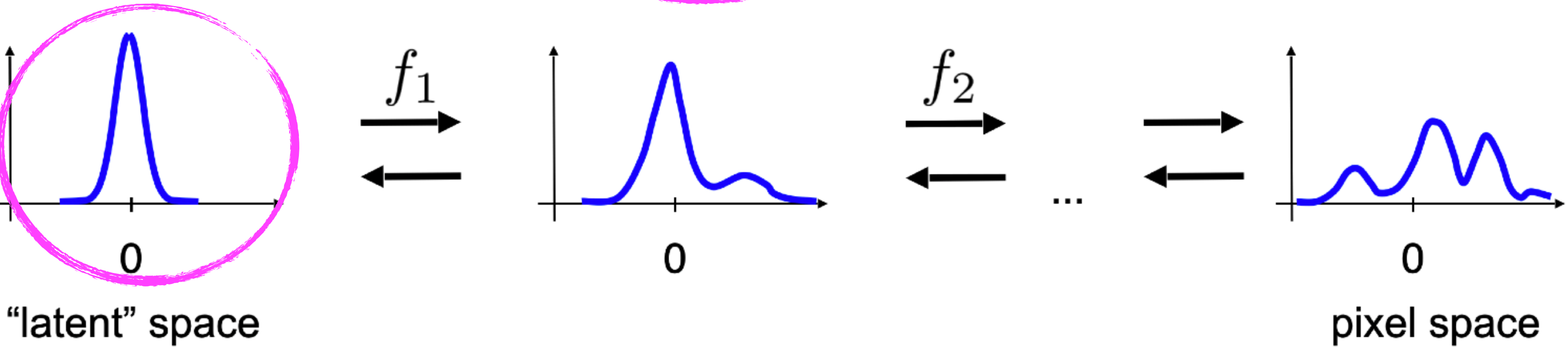


FLOWS (FLOW-BASED MODELS)

We change a random variable **x** to another random variable **z** using **invertible** transformations, $\mathbf{x}, \mathbf{z} \in \mathbb{R}^D$:

Known, e.g., Gaussian

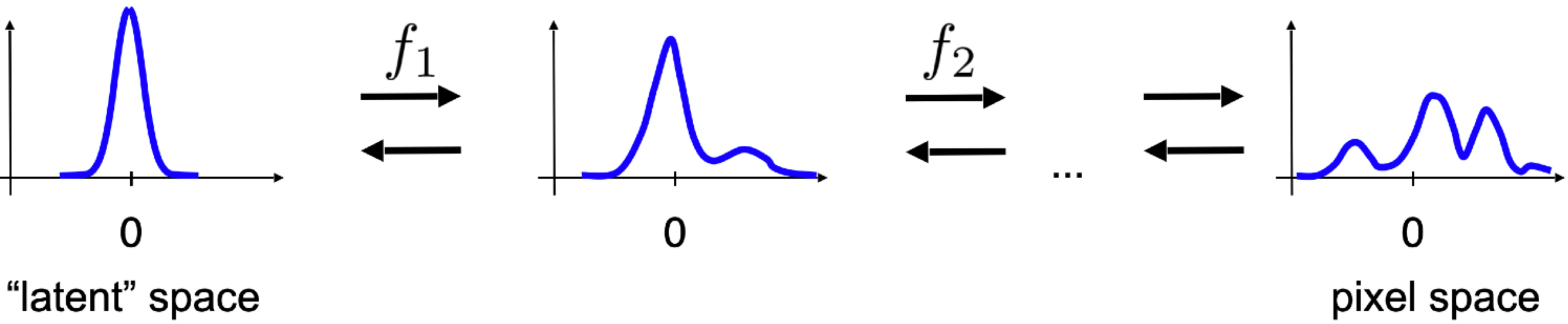
$$p(\mathbf{x}) = \pi\left(\mathbf{z}_0 = f^{-1}(\mathbf{x})\right) \prod_{i=1}^{K} \left| \mathbf{J}_{f_i}(z_{i-1}) \right|^{-1}$$



We change a random variable **x** to another random variable **z** using **invertible** transformations, $\mathbf{x}, \mathbf{z} \in \mathbb{R}^D$:

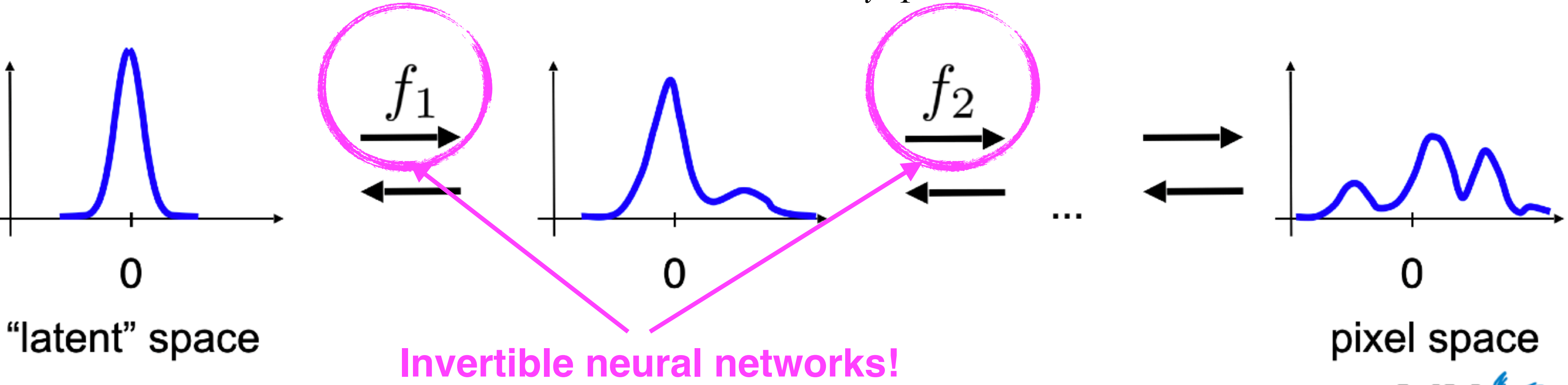
Jacobian must be tractable

$$p(\mathbf{x}) = \pi\left(\mathbf{z}_0 = f^{-1}(\mathbf{x})\right) \prod_{i=1}^{K} \left| \mathbf{J}_{f_i}(z_{i-1}) \right|^{-1}$$



We change a random variable **x** to another random variable **z** using **invertible** transformations, $\mathbf{x}, \mathbf{z} \in \mathbb{R}^D$:

$$p(\mathbf{x}) = \pi\left(\mathbf{z}_0 = f^{-1}(\mathbf{x})\right) \prod_{i=1}^{K} \left| \mathbf{J}_{f_i}(z_{i-1}) \right|^{-1}$$



FLOWS (FLOW-BASED MODELS)

We change a random variable **x** to another random variable **z** using **invertible** transformations, $\mathbf{x}, \mathbf{z} \in \mathbb{R}^D$:

$$p(\mathbf{x}) = \pi\left(\mathbf{z}_0 = f^{-1}(\mathbf{x})\right) \prod_{i=1}^{K} \left| \mathbf{J}_{f_i}(z_{i-1}) \right|^{-1}$$

Training objective:

$$\ln p(\mathbf{x}) = \ln \pi\left(\mathbf{z}_0 = f^{-1}(\mathbf{x})\right) - \sum_{i=1}^{K} \ln \left| \mathbf{J}_{f_i}(z_{i-1}) \right|$$

Two main components

1) **Coupling layer**:

$$\mathbf{y}_a = \mathbf{x}_a$$
$$\mathbf{y}_b = \exp\left(s\left(\mathbf{x}_a\right)\right) \odot \mathbf{x}_b + t\left(\mathbf{x}_a\right)$$

is invertible by design:

$$\mathbf{x}_b = \left(\mathbf{y}_b - t(\mathbf{y}_a)\right) \odot \exp\left(-s(\mathbf{y}_a)\right)$$
$$\mathbf{x}_a = \mathbf{y}_a$$

2) **Permutation layer**

Two main components

1) **Coupling layer**:

$$\mathbf{y}_a = \mathbf{x}_a$$
$$\mathbf{y}_b = \exp\left(s\left(\mathbf{x}_a\right)\right) \odot \mathbf{x}_b + t\left(\mathbf{x}_a\right)$$

Jacobian is tractable!

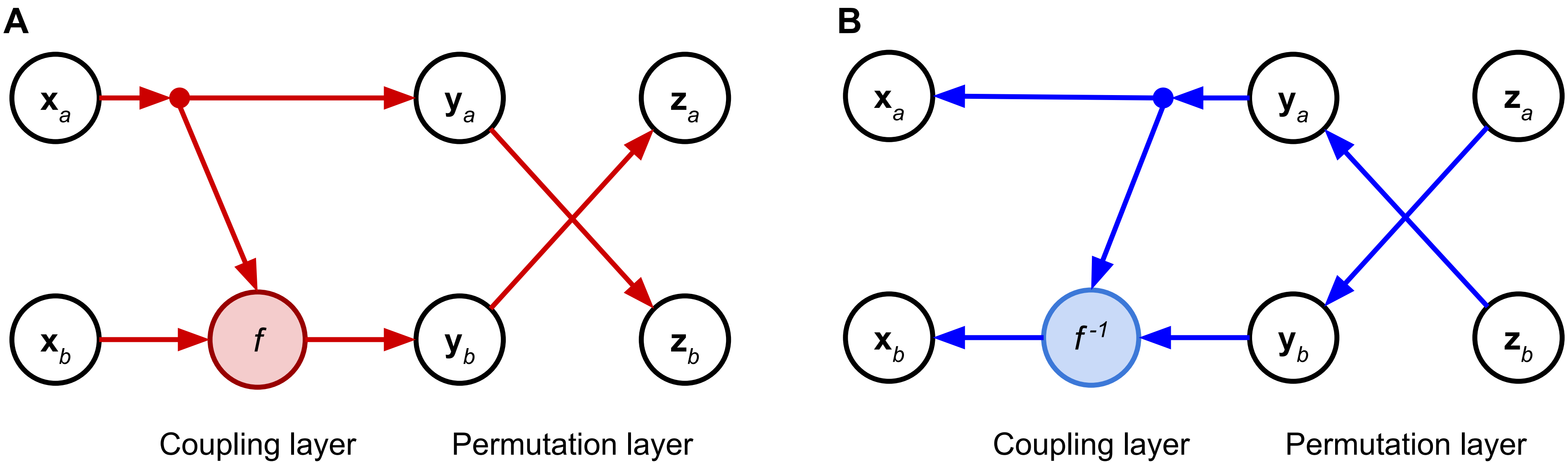
$$\det(\mathbf{J}) = \prod_{j=1}^{D-d} \exp\left(s\left(\mathbf{x}_a\right)\right)_j = \exp\left(\sum_{j=1}^{D-d} s\left(\mathbf{x}_a\right)_j\right)$$

is invertible by design:

$$\mathbf{x}_b = \left(\mathbf{y}_b - t(\mathbf{y}_a)\right) \odot \exp\left(-s(\mathbf{y}_a)\right)$$
$$\mathbf{x}_a = \mathbf{y}_a$$

2) **Permutation layer** $\det(\mathbf{J}) = 1$

FLOWS (FLOW-BASED MODELS): INVERTIBLE LAYERS



A: Forward pass. **B**: Inverse pass.

FLOWS (FLOW-BASED MODELS)

 Kingma, D.P., and Prafulla D. "Glow: Generative flow with invertible 1x1 convolutions." *NeurIPSx 2018*

DEEP GENERATIVE MODELING: HOW WE CAN FORMULATE IT?

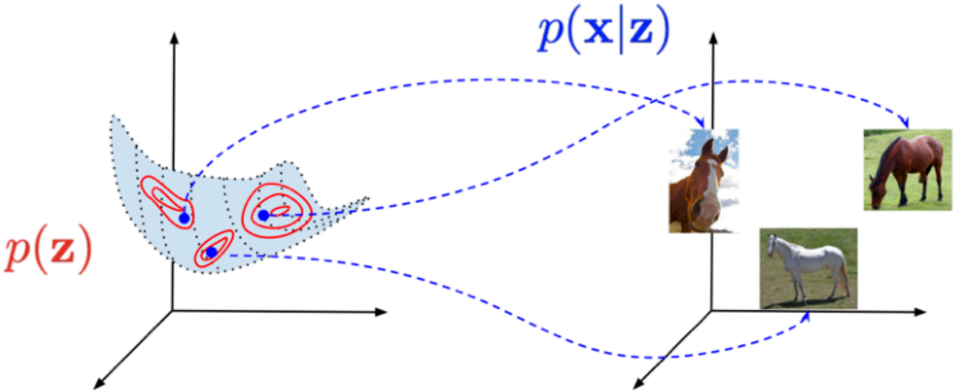
Deep Generative Models

- **Autoregressive models** (e.g., PixelCNN)
- **Flow-based models** (e.g., RealNVP)
- **Latent variable models**
  - **Implicit models** (e.g., GANs)
  - **Prescribed models** (e.g., VAEs)

Let's consider a **latent variable model** where we distinguish:

- **latent variables** $\mathbf{z} \in \mathcal{Z}^M$
- **observable variables** $\mathbf{x} \in \mathcal{X}^D$

Latent variables lie on a **low-dimensional manifold**.



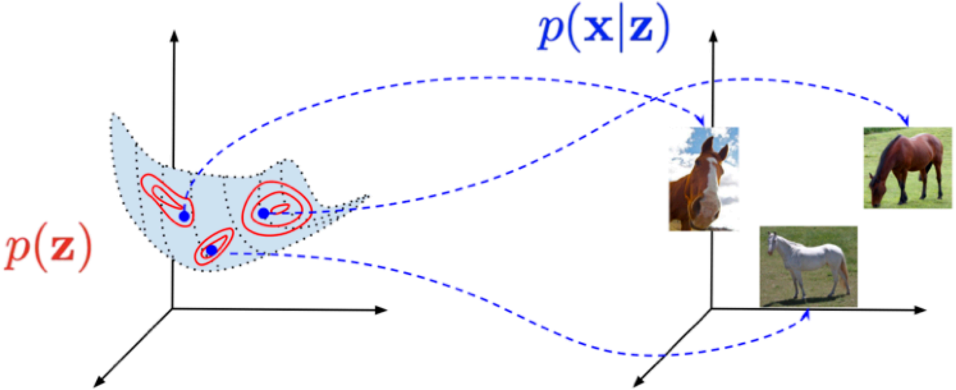
Generative process:

1. $\mathbf{z} \sim p(\mathbf{z})$
2. $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$

Let's consider a **latent variable model** where we distinguish:

- **latent variables** $\mathbf{z} \in \mathcal{Z}^M$
- **observable variables** $\mathbf{x} \in \mathcal{X}^D$

Latent variables lie on a **low-dimensional manifold**.



Generative process:

1. $\mathbf{z} \sim p(\mathbf{z})$
2. $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$

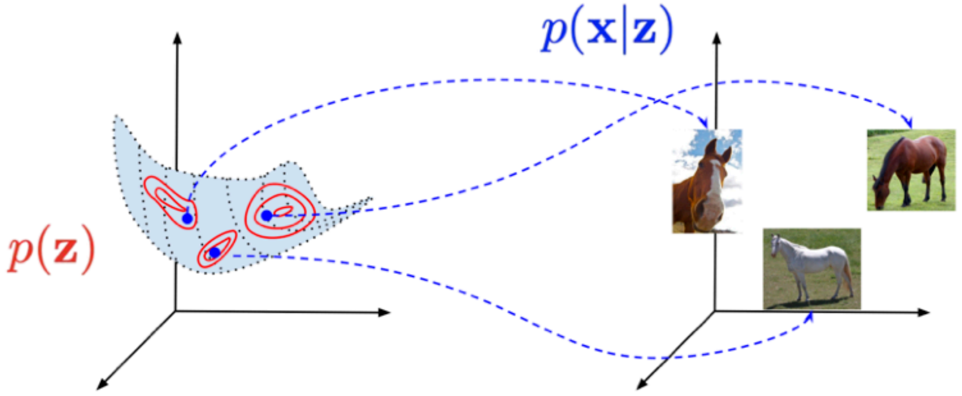
The objective function:

$$\ln p(\mathbf{x}) = \ln \int p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}$$

Let's consider a **latent variable model** where we distinguish:

- **latent variables** $\mathbf{z} \in \mathcal{Z}^M$
- **observable variables** $\mathbf{x} \in \mathcal{X}^D$

Latent variables lie on a **low-dimensional manifold**.



Generative process:

1. $\mathbf{z} \sim p(\mathbf{z})$
2. $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$

The objective function:

$$\ln p(\mathbf{x}) = \ln \int p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}$$

The integral is intractable…

$$\ln p(\mathbf{x}) = \ln \int p(\mathbf{x} \,|\, \mathbf{z}) p(\mathbf{z}) \; \mathrm{d}\mathbf{z}$$

$$= \ln \int \frac{q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})} p(\mathbf{x} \,|\, \mathbf{z}) p(\mathbf{z}) \; \mathrm{d}\mathbf{z}$$

$$= \ln \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z})} \left[\frac{p(\mathbf{x} \,|\, \mathbf{z}) p(\mathbf{z})}{q_\phi(\mathbf{z})} \right]$$

$$\geq \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z})} \ln \left[\frac{p(\mathbf{x} \,|\, \mathbf{z}) p(\mathbf{z})}{q_\phi(\mathbf{z})} \right]$$

$$= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z})} \left[\ln p(\mathbf{x} \,|\, \mathbf{z}) + \ln p(\mathbf{z}) - \ln q_\phi(\mathbf{z}) \right]$$

$$= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z})} \left[\ln p(\mathbf{x} \,|\, \mathbf{z}) \right] - \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z})} \left[\ln q_\phi(\mathbf{z}) - \ln p(\mathbf{z}) \right]$$

$$\ln p(\mathbf{x}) = \ln \int p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

$$= \ln \int \frac{q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})} p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

Variational posteriors

$$= \ln \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})}\left[\frac{p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})}\right]$$

$$\geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \ln \left[\frac{p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})}\right]$$

$$= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})}\left[\ln p(\mathbf{x}\,|\,\mathbf{z}) + \ln p(\mathbf{z}) - \ln q_\phi(\mathbf{z})\right]$$

$$= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})}\left[\ln p(\mathbf{x}\,|\,\mathbf{z})\right] - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})}\left[\ln q_\phi(\mathbf{z}) - \ln p(\mathbf{z})\right]$$

$$\ln p(\mathbf{x}) = \ln \int p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

$$= \ln \int \frac{q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})} p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

$$= \ln \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\frac{p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})}\right]$$

$$\geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \ln \left[\frac{p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})}\right]$$

$$= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\ln p(\mathbf{x}\,|\,\mathbf{z}) + \ln p(\mathbf{z}) - \ln q_\phi(\mathbf{z})\right]$$

$$= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\ln p(\mathbf{x}\,|\,\mathbf{z})\right] - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\ln q_\phi(\mathbf{z}) - \ln p(\mathbf{z})\right]$$

$$
\begin{aligned}
\ln p(\mathbf{x}) &= \ln \int p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&= \ln \int \frac{q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})} p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&= \ln \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\frac{p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})} \right] \\
&\geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \ln \left[\frac{p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})} \right] \qquad \text{Jensen's inequality} \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\ln p(\mathbf{x}\,|\,\mathbf{z}) + \ln p(\mathbf{z}) - \ln q_\phi(\mathbf{z}) \right] \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\ln p(\mathbf{x}\,|\,\mathbf{z}) \right] - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\ln q_\phi(\mathbf{z}) - \ln p(\mathbf{z}) \right]
\end{aligned}
$$

$$
\begin{aligned}
\ln p(\mathbf{x}) &= \ln \int p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&= \ln \int \frac{q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})} p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&= \ln \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\frac{p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})} \right] \\
&\geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \ln \left[\frac{p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})} \right] \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\ln p(\mathbf{x}\,|\,\mathbf{z}) + \ln p(\mathbf{z}) - \ln q_\phi(\mathbf{z}) \right] \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\ln p(\mathbf{x}\,|\,\mathbf{z}) \right] - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\ln q_\phi(\mathbf{z}) - \ln p(\mathbf{z}) \right]
\end{aligned}
$$

Jensen's inequality

$$
\begin{aligned}
\ln p(\mathbf{x}) &= \ln \int p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&= \ln \int \frac{q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})} p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&= \ln \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\frac{p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})} \right] \\
&\geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \ln \left[\frac{p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})} \right] \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\ln p(\mathbf{x}\,|\,\mathbf{z}) + \ln p(\mathbf{z}) - \ln q_\phi(\mathbf{z}) \right] \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\ln p(\mathbf{x}\,|\,\mathbf{z}) \right] - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \left[\ln q_\phi(\mathbf{z}) - \ln p(\mathbf{z}) \right]
\end{aligned}
$$

$$
\begin{aligned}
\ln p(\mathbf{x}) &= \ln \int p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&= \ln \int \frac{q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})} p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z} \\
&= \ln \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})}\left[\frac{p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})}\right] \\
&\geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \ln \left[\frac{p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})}\right] \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})}\left[\ln p(\mathbf{x}\,|\,\mathbf{z}) + \ln p(\mathbf{z}) - \ln q_\phi(\mathbf{z})\right] \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})}\left[\ln p(\mathbf{x}\,|\,\mathbf{z})\right] - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})}\left[\ln q_\phi(\mathbf{z}) - \ln p(\mathbf{z})\right]
\end{aligned}
$$

$$\ln p(\mathbf{x}) = \ln \int p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

$$= \ln \int \frac{q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})} p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

$$= \ln \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})}\left[\frac{p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})}\right]$$

$$\geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})} \ln \left[\frac{p(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})}\right]$$

Reconstruction error **"Regularization" term**

$$= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})}\left[\ln p(\mathbf{x}\,|\,\mathbf{z}) + \ln p(\mathbf{z}) - \ln q_\phi(\mathbf{z})\right]$$

$$= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})}\left[\ln p(\mathbf{x}\,|\,\mathbf{z})\right] - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})}\left[\ln q_\phi(\mathbf{z}) - \ln p(\mathbf{z})\right]$$

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln p(\mathbf{x}\,|\,\mathbf{z})\right] - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln q_\phi(\mathbf{z}\,|\,\mathbf{x}) - \ln p(\mathbf{z})\right]$$

ELBO: Evidence Lower Bound

 Kingma, D.P., and Welling, M.. "Auto-encoding variational bayes." *ICLR 2014*

VARIATIONAL AUTO-ENCODERS

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln p(\mathbf{x}\,|\,\mathbf{z})\right] - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln q_\phi(\mathbf{z}\,|\,\mathbf{x}) - \ln p(\mathbf{z})\right]$$

We consider **amortized inference**: $q_\phi(\mathbf{z}\,|\,\mathbf{x})$

In other words, a single parameterization for each new input **x**.

 Kingma, D.P., and Welling, M.. "Auto-encoding variational bayes." *ICLR 2014*

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln p(\mathbf{x}\,|\,\mathbf{z})\right] - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln q_\phi(\mathbf{z}\,|\,\mathbf{x}) - \ln p(\mathbf{z})\right]$$

We consider **amortized inference**: $q_\phi(\mathbf{z}\,|\,\mathbf{x})$

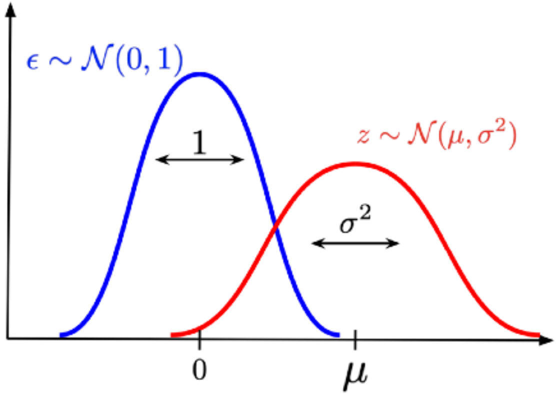
In other words, a single parameterization for each new input **x**.

Moreover, we use **reparameterization trick**:

Every Gaussian variable could be defined as:

$z = \mu + \sigma \cdot \varepsilon$

where $\varepsilon \sim \mathcal{N}(0,1)$



Kingma, D.P., and Welling, M.. "Auto-encoding variational bayes." *ICLR 2014*

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln p(\mathbf{x}\,|\,\mathbf{z})\right] - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln q_\phi(\mathbf{z}\,|\,\mathbf{x}) - \ln p(\mathbf{z})\right]$$

We consider **amortized inference**: $q_\phi(\mathbf{z}\,|\,\mathbf{x})$

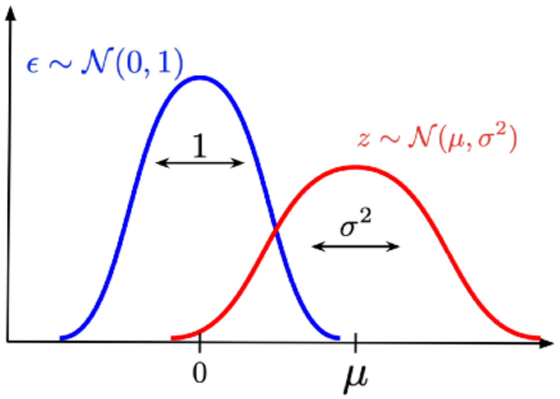
In other words, a single parameterization for each new input **x**.

Moreover, we use **reparameterization trick**:

It reduces the variance of the gradients.

It allows to get randomness outside **z**.

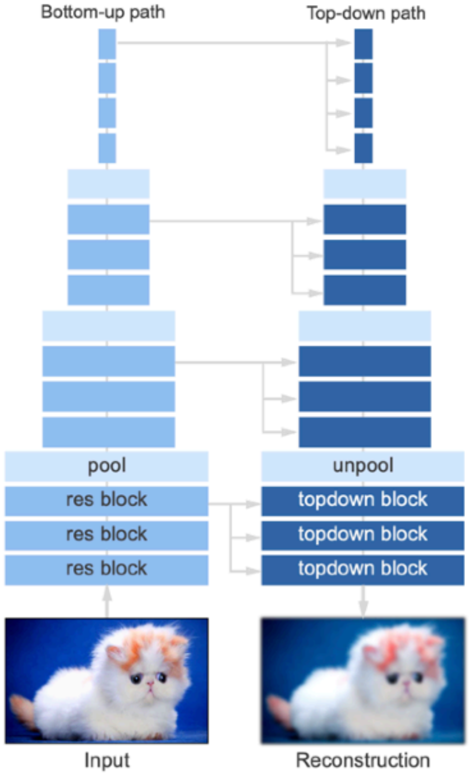
$$z = \mu + \sigma \cdot \varepsilon$$



 Kingma, D.P., and Welling, M.. "Auto-encoding variational bayes." *ICLR 2014*

VARIATIONAL AUTO-ENCODERS

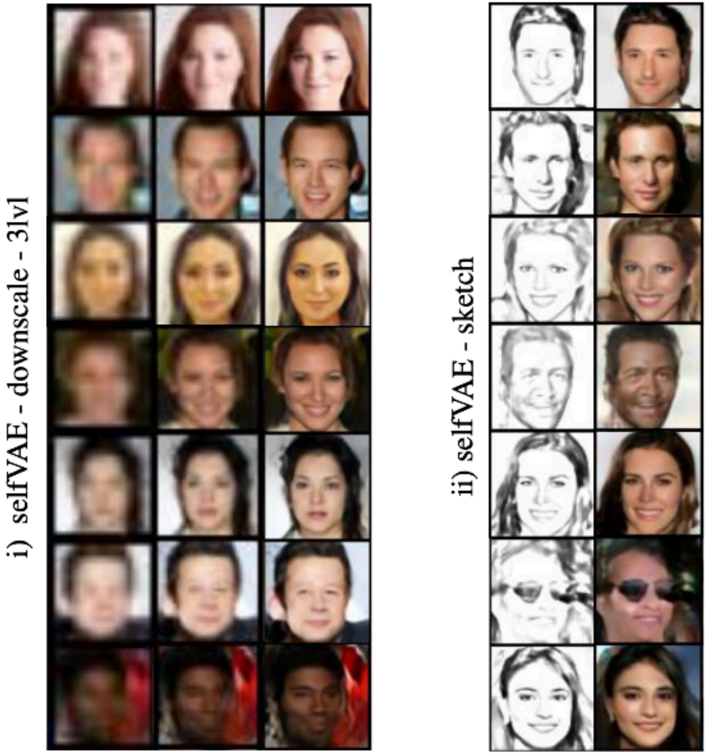
Generations



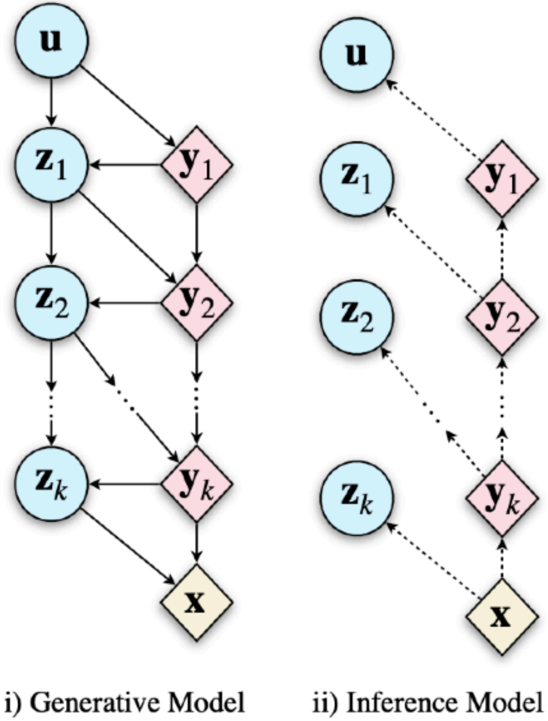
Very Deep VAE

67 Child, R. "Very Deep VAEs Generalize Autoregressive Models and Can Outperform Them on Images." *ICLR 2021*

VARIATIONAL AUTO-ENCODERS



Generations



Hierarchical VAE

Gatopoulos, I., and Tomczak, J.M., "Self-Supervised Variational Auto-Encoders." *arXiv preprint arXiv:2010.02014* (2020).

CONCLUSION

- Here: **the likelihood-based generative models**.

CONCLUSION

- Here: **the likelihood-based generative models**.
- We **skipped** Generative Adversarial Nets & others.

CONCLUSION

- Here: **the likelihood-based generative models**.
- We **skipped** Generative Adversarial Nets & others.
- Why generative modeling?

$$p(\mathbf{x}, y) = p(y \mid \mathbf{x})\, p(\mathbf{x})$$

CONCLUSION

- Here: **the likelihood-based generative models**.
- We **skipped** Generative Adversarial Nets & others.
- Why generative modeling?

$$p(\mathbf{x}, y) = p(y \,|\, \mathbf{x})\, p(\mathbf{x})$$

- Important directions:

➡ Better uncertainty quantification

➡ New parameterization (new neural networks)

➡ Out-of-Distribution

➡ Continual learning

THANK YOU FOR YOUR ATTENTION

Jakub M. Tomczak
Computational Intelligence group
Vrije Universiteit Amsterdam

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

Github: https://github.com/jmtomczak

Twitter: https://twitter.com/jmtomczak