Imperial College London

Deep Learning (ELEC60009/96033)

Lecture 7 **Generating Data**

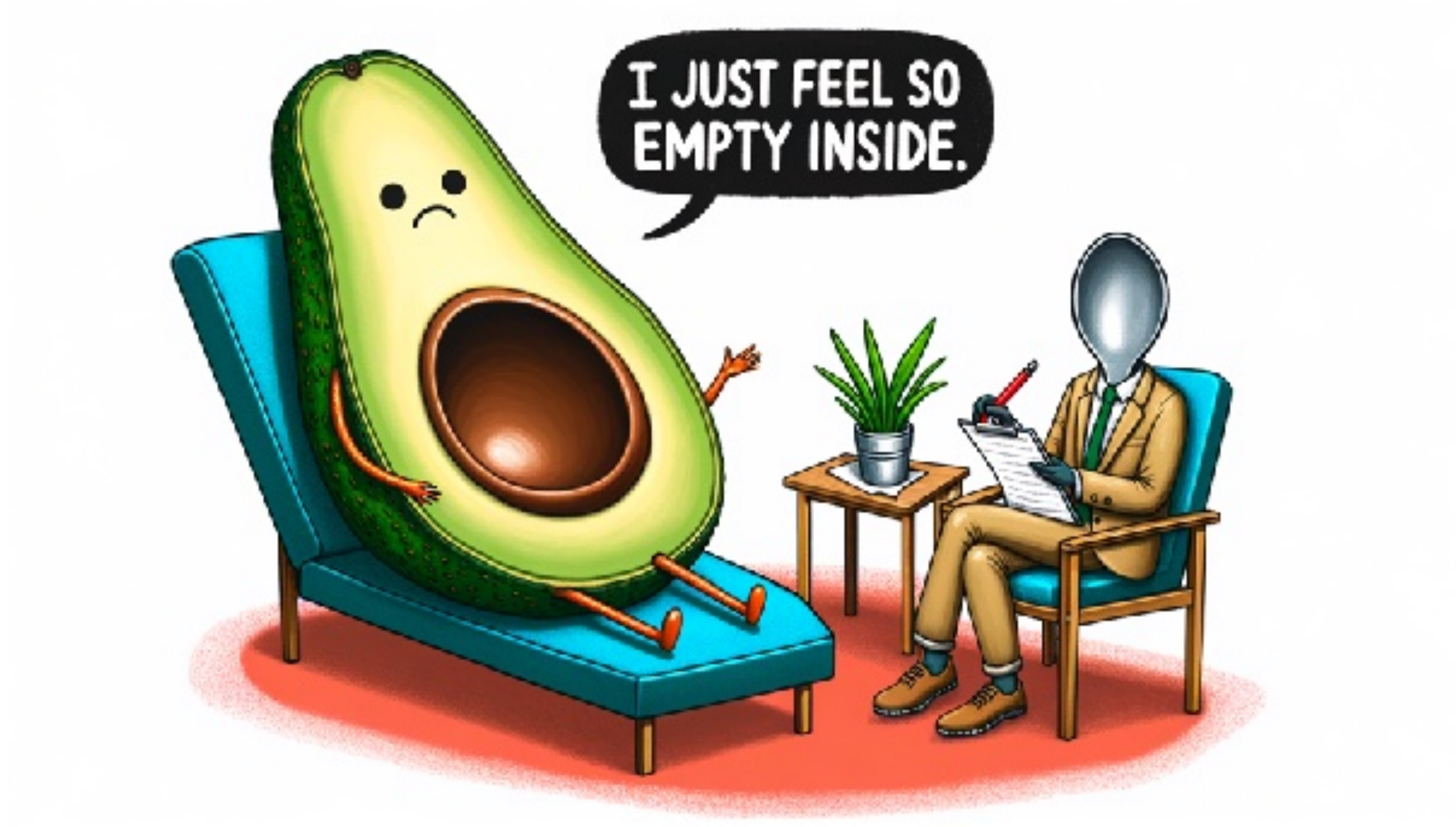
Seyed Moosavi & Chen Qin & Krystian Mikolajczyk

Spring 2023-2024

```
Silent neurons swirl,
Creative output unfolds,
Generative art born.
```

— *ChatGPT*

“Generative AI”



An illustration of an avocado sitting in a therapist's chair, saying 'I just feel so empty inside' with a pit-sized hole in its center. The therapist, a spoon, scribbles notes.

Photo of a lychee-inspired spherical chair, with a bumpy white exterior and plush interior, set against a tropical wallpaper.

A modern architectural building with large glass windows, situated on a cliff overlooking a serene ocean at sunset.

OpenAI

“Generative AI”

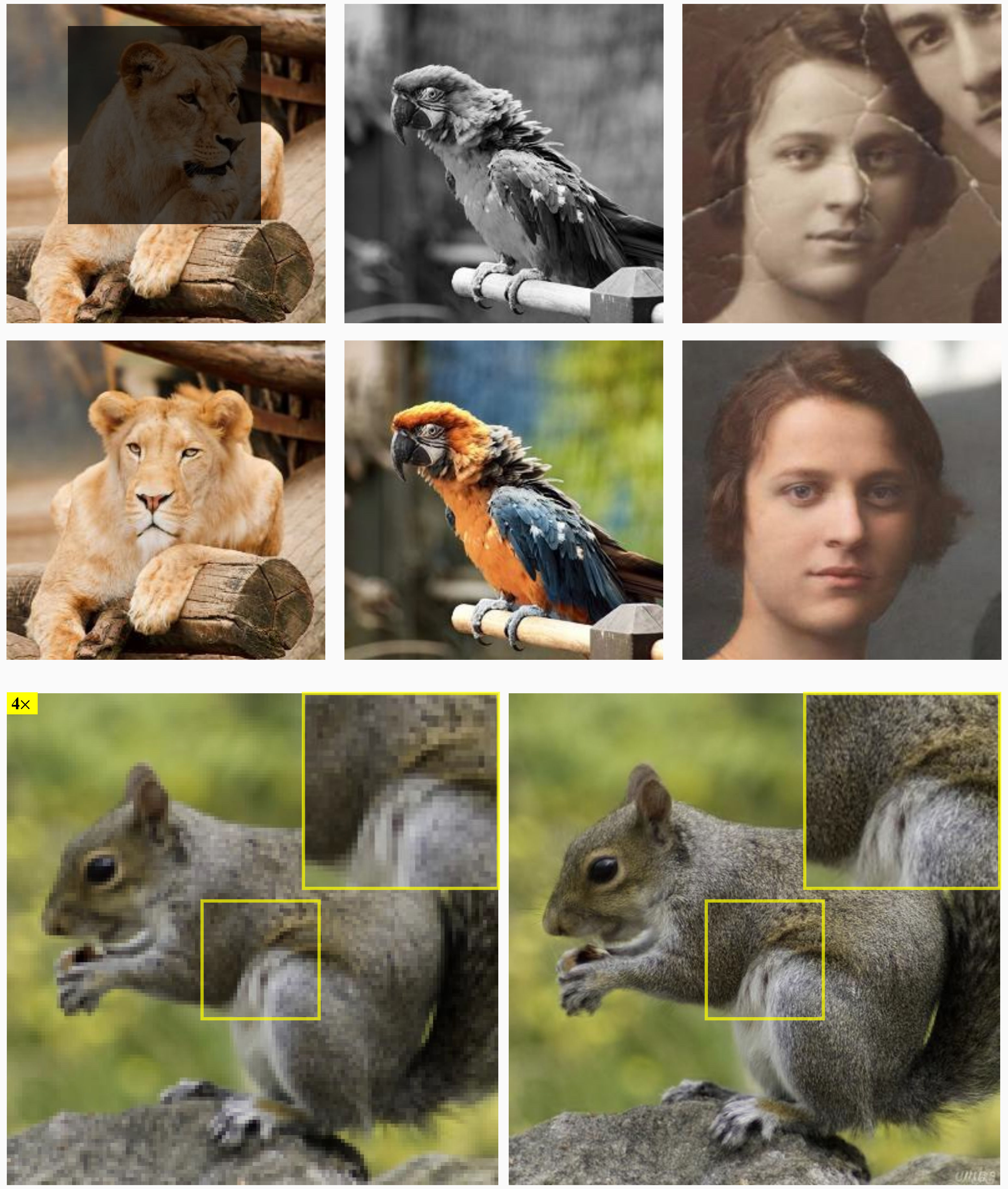
A close up view of a glass sphere that has a zen garden within it. There is a small dwarf in the sphere who is raking the zen garden and creating patterns in the sand.

An extreme close-up of an gray-haired man with a beard in his 60s, he is deep in thought pondering the history of the universe as he sits at a cafe in Paris, his eyes focus on people offscreen as they walk as he sits mostly motionless, he is dressed in a wool coat suit coat with a button-down shirt , he wears a brown beret and glasses and has a very professorial appearance, and the end he offers a subtle closed-mouth smile as if he found the answer to the mystery of life, the lighting is very cinematic with the golden light and the Parisian streets and city in the background, depth of field, cinematic 35mm film.

Extreme close up of a 24 year old woman’s eye blinking, standing in Marrakech during magic hour, cinematic film shot in 70mm, depth of field, vivid colors, cinematic

OpenAI

Conditional generation



Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model
Wang et al., *ICLR 2023*.

Exploiting Deep Generative Prior for Versatile Image Restoration and Manipulation
Pan et al., *ECCV 2020*.

What is generative modelling?

In a broad sense *a process where algorithms create new data samples from observed patterns.*

What is generative modelling?

In a broad sense *a process where algorithms create new data samples from observed patterns.*

Simple scenario given a bunch of samples from a (possibly unknown) distribution, how do you generate more samples?

Step 1 learn/fit a distribution.

Step 2 sample the distribution.

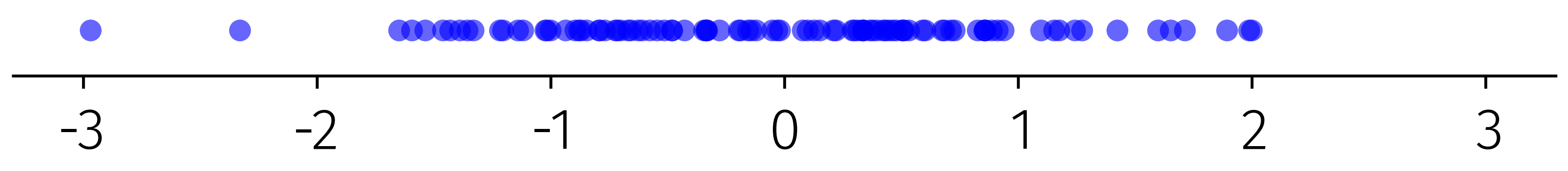
What is generative modelling?

In a broad sense *a process where algorithms create new data samples from observed patterns.*

Simple scenario given a bunch of samples from a (possibly unknown) distribution, how do you generate more samples?

Step 1 learn/fit a distribution.

Step 2 sample the distribution.



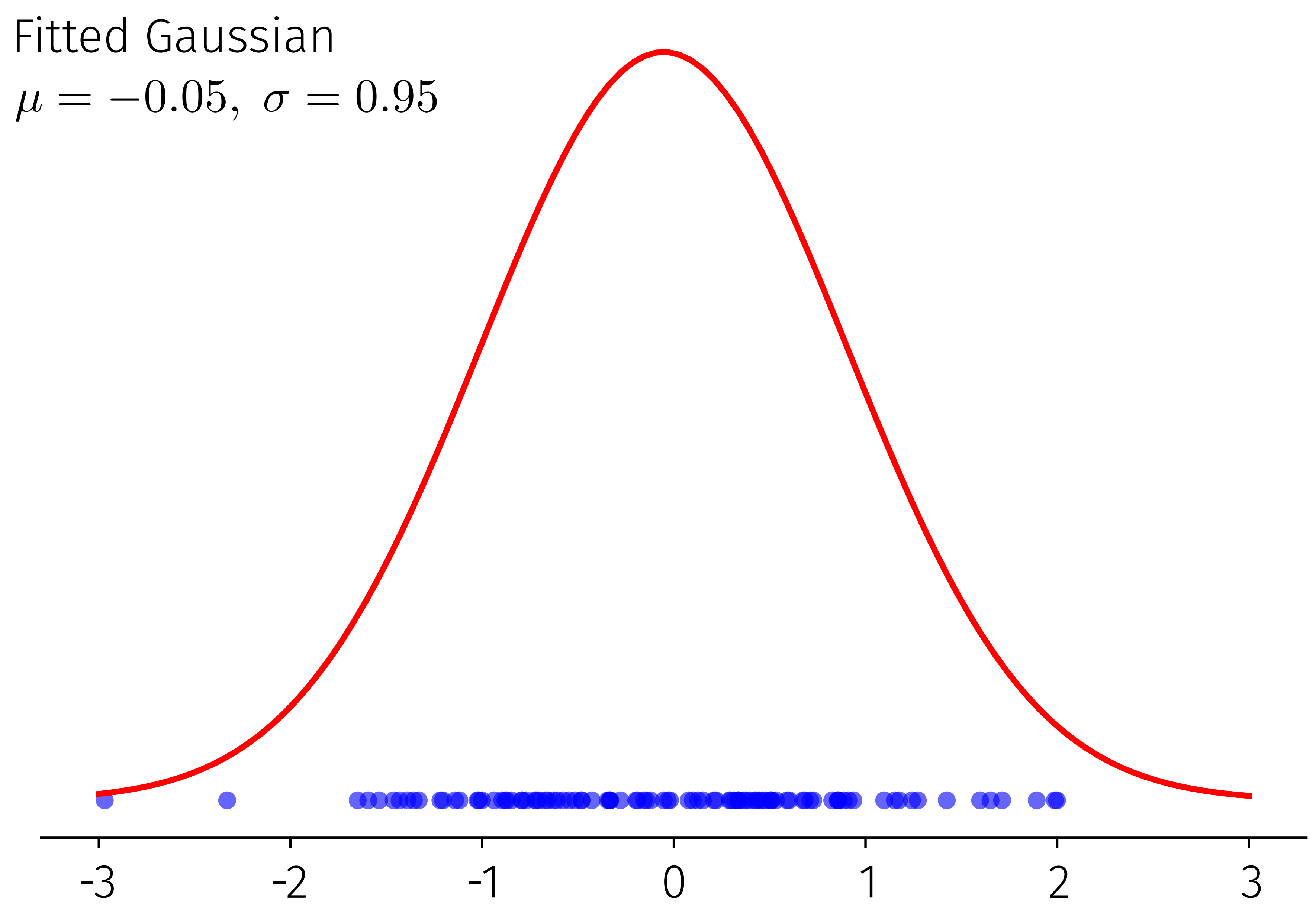
What is generative modelling?

In a broad sense *a process where algorithms create new data samples from observed patterns.*

Simple scenario given a bunch of samples from a (possibly unknown) distribution, how do you generate more samples?

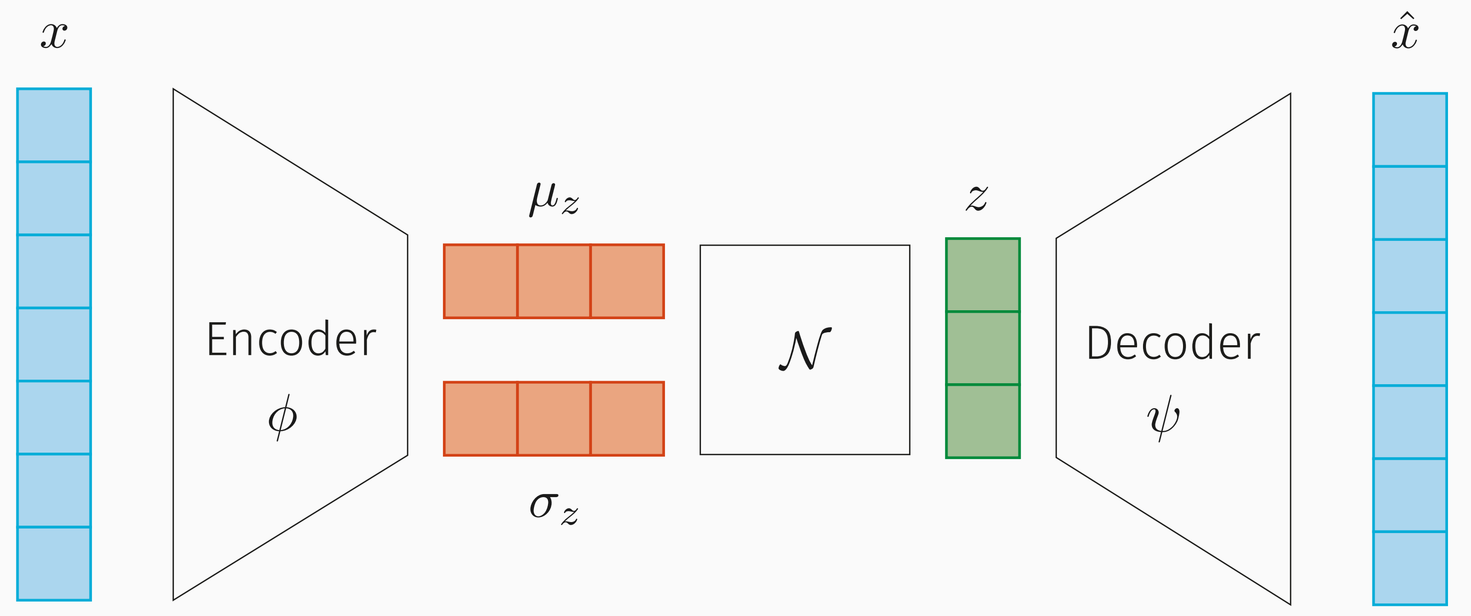
Step 1 learn/fit a distribution.
Maximum Likelihood Estimation

Step 2 sample the distribution.
Box-Muller

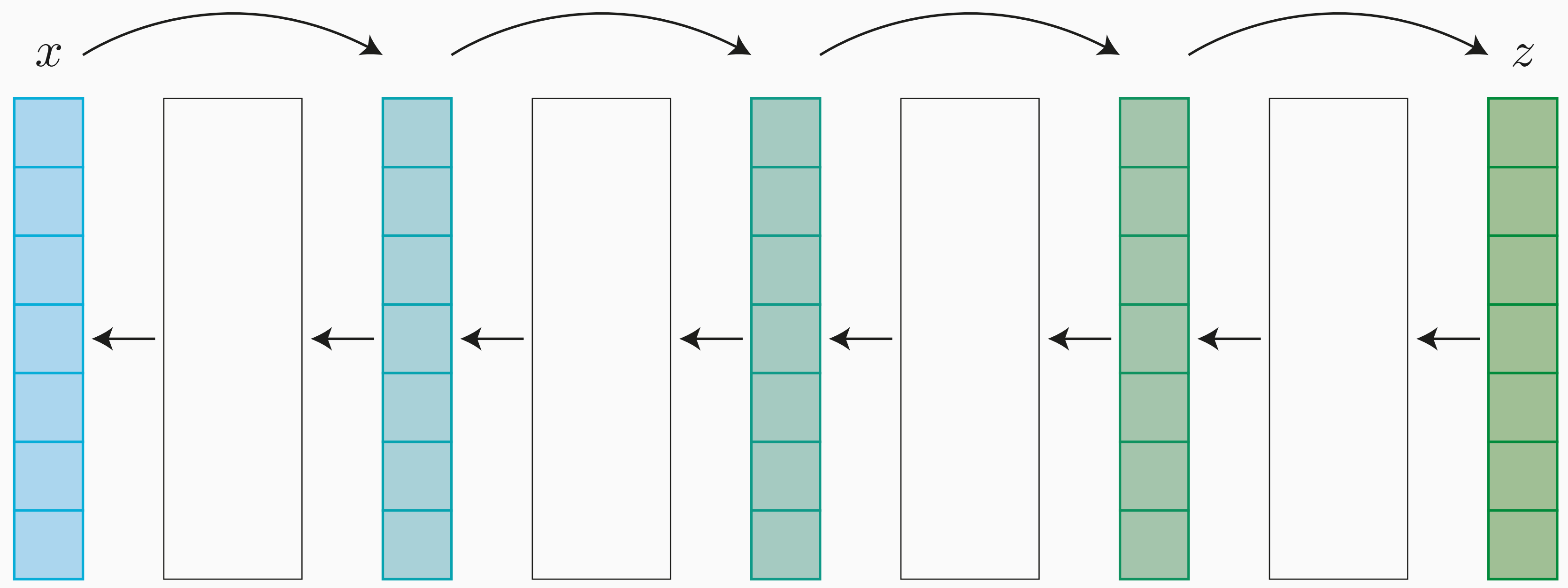


Learning to generate

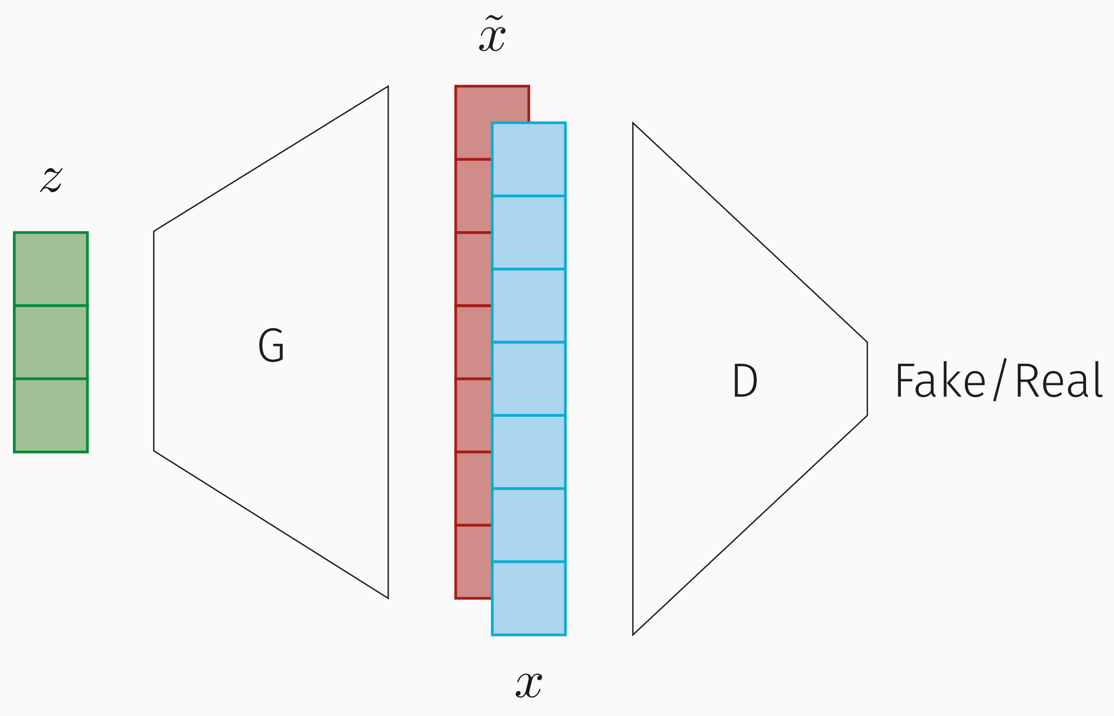
Variational Autoencoders (VAEs)



Diffusion Models



Generative Adversarial Networks (GANs)



Quiz true or false?

- All generative models, in one way or another, directly learn the data distribution.
- Likelihood-based generative models are not suitable for inpainting.
- Generative models can be used to classify data.
- Autoencoders (last lecture) can be used to generate data.

Recall...

Autoencoders (last lecture) learn representations in an unsupervised way.

Good representations can be used to generate data.

Can we use Autoencoders to generate data?

Recall...

Autoencoders (last lecture) learn representations in an unsupervised way.

Good representations can be used to generate data.

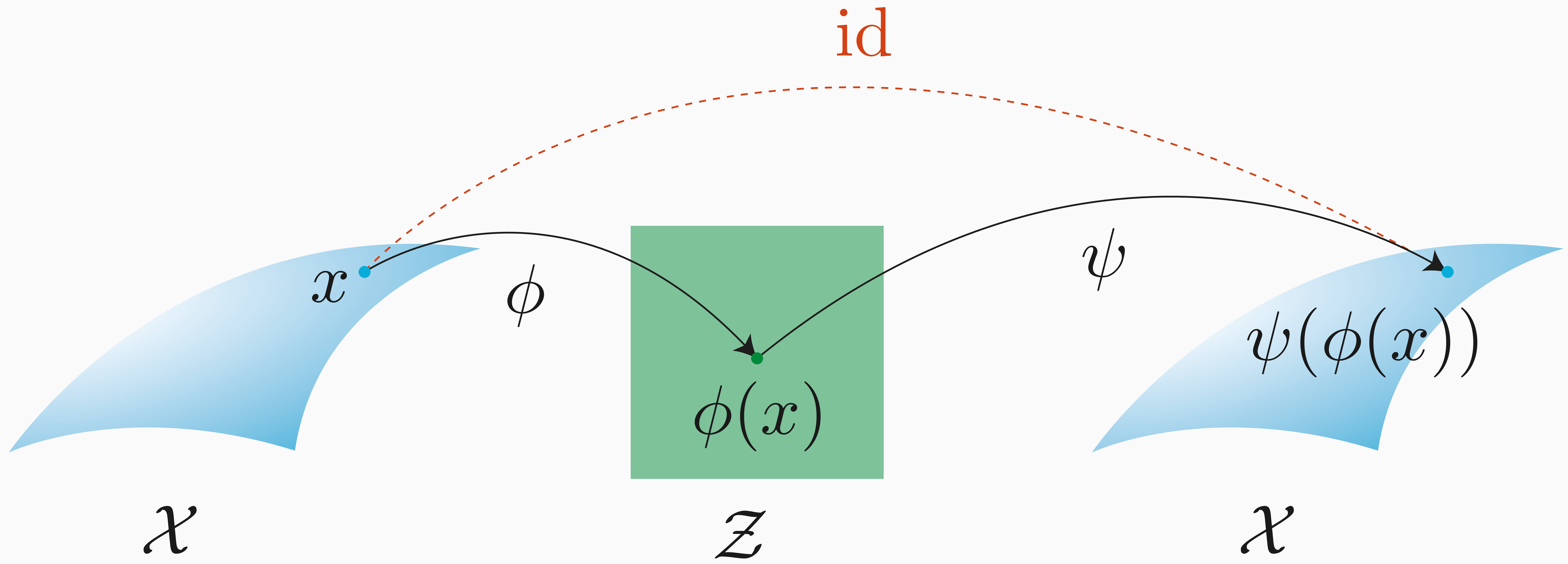
Can we use Autoencoders to generate data?

Simple Autoencoders may not be suitable for data generation!

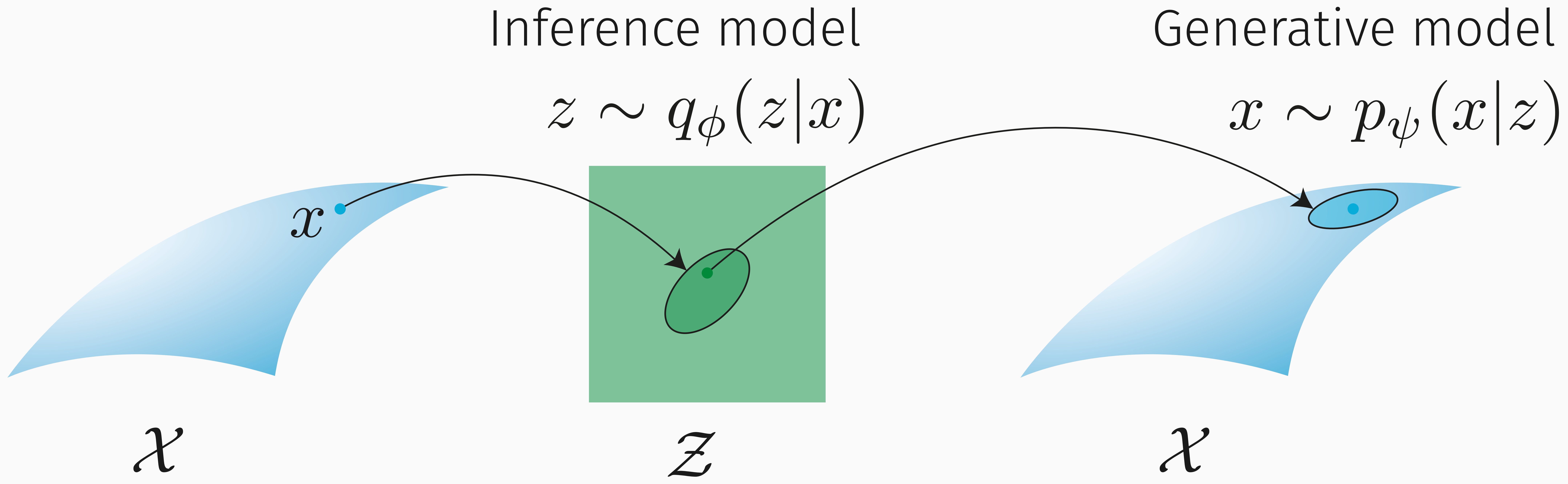
- No regularization.
- Latent space is “unstructured”.

Variational Autoencoders

A probabilistic approach



A probabilistic approach



- Imposing a latent space structure.
- Prevent overfitting.

Variational AutoEncoder (VAE)

For example, for the *generative model*

$$\mu^{\psi} = \psi(z),$$
$$p_{\psi}(x|z) = \mathcal{N}(x; \mu^{\psi}, I_d),$$

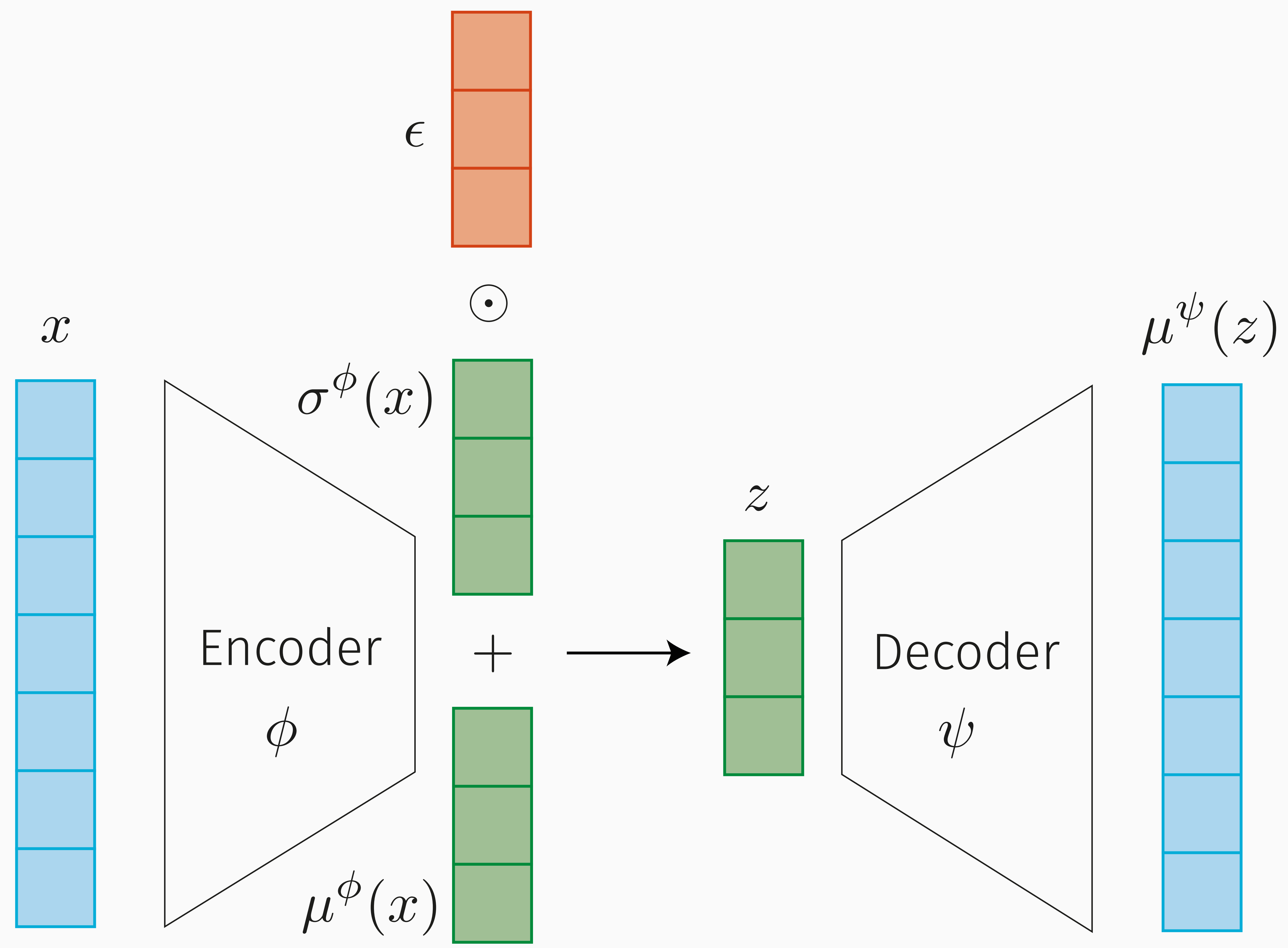
and for the *inference model*

$$(\mu^{\phi}, \sigma^{\phi}) = \phi(x),$$
$$q_{\phi}(z|x) = \mathcal{N}\left(z; \mu^{\phi}, \mathrm{diag}(\sigma^{\phi})\right).$$

Also, we assume a standard Gaussian distribution over the latent representations, i.e.,

$$p(z) = \mathcal{N}(z; 0, I_p).$$

Variational AutoEncoder (VAE) diagram



Intractability of maximum likelihood

How to learn ψ?

Perhaps via maximizing **log-likelihood**?

$$\max_{\psi} \log p_{\psi}(x)$$

Intractability of maximum likelihood

How to learn ψ?

Perhaps via maximizing **log-likelihood**?

$$\max_{\psi} \log p_{\psi}(x) = \max_{\psi} \log \int p(z) p_{\psi}(x|z) dz$$

Intractability of maximum likelihood

How to learn ψ?

Perhaps via maximizing **log-likelihood**?

$$\max_{\psi} \log p_{\psi}(x) = \max_{\psi} \log \int p(z) p_{\psi}(x|z) dz$$
$$= \max_{\psi} \int q_{\phi}(z|x) \log p_{\psi}(x) dz$$

Variational Autoencoder (VAE) jointly learns ϕ and ψ!

Quiz Kullback-Leibler divergence

Recall the definition of **Kullback–Leibler (KL) divergence** between probability distributions $p(x)$ and $q(x)$

$$D_{KL}(q(x), p(x)) = \mathbb{E}_{q(x)}[\log q(x) - \log p(x)].$$

- D_{KL} is between $-\infty$ and ∞.
- $D_{KL} = 0 \Rightarrow p(x) = q(x)$.
- $p(x) = q(x) \Rightarrow D_{KL} = 0$.
- D_{KL} is symmetric, i.e., $D_{KL}(p, q) = D_{KL}(q, p)$.

Evidence Lower Bound (ELBO)

For any choice of inference model $q(z|x)$, we have:

$$\log p(x) = \mathbb{E}_{q(z|x)}[\log p(x)]$$

Evidence Lower Bound (ELBO)

For any choice of inference model $q(z|x)$, we have:

$$\begin{aligned}\log p(x) &= \mathbb{E}_{q(z|x)}[\log p(x)] \\ &= \mathbb{E}_{q(z|x)}[\log p(x|z) + \log p(z) - \log p(z|x)]\end{aligned}$$

Evidence Lower Bound (ELBO)

For any choice of inference model $q(z|x)$, we have:

$$\begin{aligned}
\log p(x) &= \mathbb{E}_{q(z|x)}[\log p(x)] \\
&= \mathbb{E}_{q(z|x)}[\log p(x|z) + \log p(z) - \log p(z|x)] \\
&= \mathbb{E}_{q(z|x)}[\log p(x|z) + \log p(z) - \log q(z|x) + \log q(z|x) - \log p(z|x)]
\end{aligned}$$

Evidence Lower Bound (ELBO)

For any choice of inference model $q(z|x)$, we have:

$$\begin{aligned}
\log p(x) &= \mathbb{E}_{q(z|x)}[\log p(x)] \\
&= \mathbb{E}_{q(z|x)}[\log p(x|z) + \log p(z) - \log p(z|x)] \\
&= \mathbb{E}_{q(z|x)}[\log p(x|z) + \log p(z) - \log q(z|x) + \log q(z|x) - \log p(z|x)] \\
&= \underbrace{\mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x), p(z))}_{:=\mathcal{L}_{ELBO}} + \underbrace{D_{KL}(q(z|x), p(z|x))}_{\geq 0}
\end{aligned}$$

Evidence Lower Bound (ELBO)

For any choice of inference model $q(z|x)$, we have:

$$
\begin{aligned}
\log p(x) &= \mathbb{E}_{q(z|x)}[\log p(x)] \\
&= \mathbb{E}_{q(z|x)}[\log p(x|z) + \log p(z) - \log p(z|x)] \\
&= \mathbb{E}_{q(z|x)}[\log p(x|z) + \log p(z) - \log q(z|x) + \log q(z|x) - \log p(z|x)] \\
&= \underbrace{\mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x), p(z))}_{:=\mathcal{L}_{ELBO}} + \underbrace{D_{KL}(q(z|x), p(z|x))}_{\geq 0}
\end{aligned}
$$

Hence, $\mathcal{L}_{ELBO}(\phi, \psi)$ is a lower bound for the log-likelihood (a.k.a. evidence), and the equality holds iff $q(z|x) = p(z|x)$.

Optimizing ELBO

Maximizing $\mathcal{L}_{ELBO}(\phi, \psi)$ w.r.t. ϕ and ψ:

$$\max_{\phi,\psi} \mathcal{L}_{ELBO}(\phi, \psi) = \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log p_\psi(x|z)]}_{\text{reconstruction error}} - \underbrace{D_{KL}(q_\phi(z|x), p(z))}_{\text{regularization term}}.$$

Optimizing ELBO

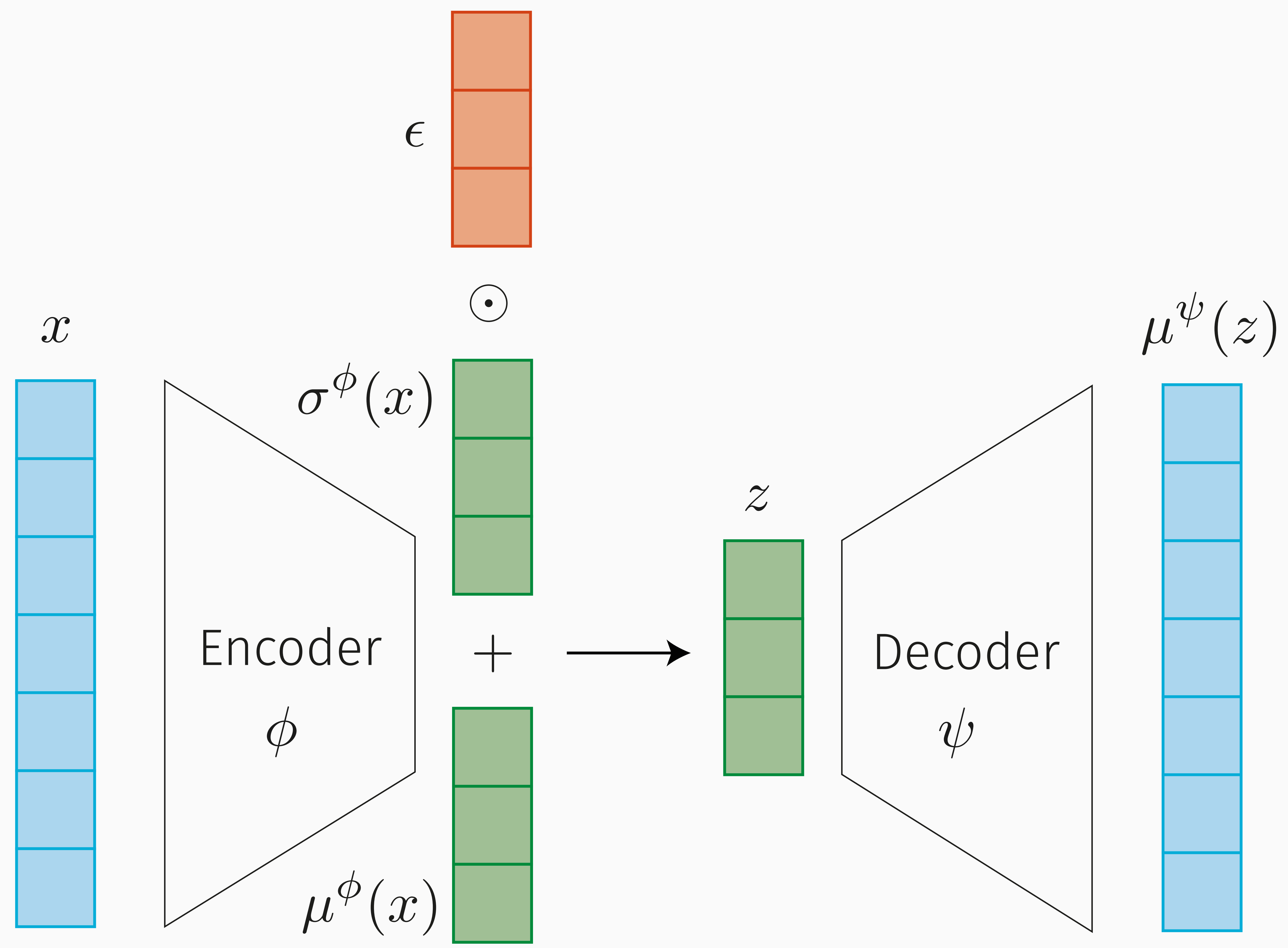
Maximizing $\mathcal{L}_{ELBO}(\phi, \psi)$ w.r.t. ϕ and ψ:

$$\max_{\phi,\psi} \mathcal{L}_{ELBO}(\phi, \psi) = \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log p_\psi(x|z)]}_{\text{reconstruction error}} - \underbrace{D_{KL}(q_\phi(z|x), p(z))}_{\text{regularization term}}.$$

Two birds with one stone

- Improves the generative model $p_\psi(x)$.
- The inference model $q_\phi(z|x)$ becomes closer to the true posterior $p_\psi(z|x)$.

Variational AutoEncoder (VAE) diagram



Gradient of ELBO

If ϕ and ψ are neural networks, we can use SGD to optimize $\mathcal{L}_{ELBO}$. But how?

Gradient of ELBO

If ϕ and ψ are neural networks, we can use SGD to optimize $\mathcal{L}_{ELBO}$. But how?

- We can use a single-sample MC estimator of the first term:

$$\mathbb{E}_{q_\phi(z|x)}[\log p_\psi(x|z)] \approx \log p_\psi(x|z)$$

  for some $z \sim q_\phi(z|x)$.

Gradient of ELBO

If ϕ and ψ are neural networks, we can use SGD to optimize $\mathcal{L}_{ELBO}$. But how?

- We can use a single-sample MC estimator of the first term:

$$\mathbb{E}_{q_\phi(z|x)}[\log p_\psi(x|z)] \approx \log p_\psi(x|z)$$

  for some $z \sim q_\phi(z|x)$.

- For the second term, we use the fact that D_{KL} between two Gaussians, namely $q(z|x)$ and $p(z)$, can be written in closed-form:

$$D_{KL}(q_\phi(z|x), p(z)) = \frac{1}{2}\left(\|\mu^\phi(x)\|_2^2 + \|\sigma^\phi(x)\|_2^2 - 2\sum_{k=1}^{p} \log \sigma_k^\phi(x)\right) + \text{const}$$

Simplified ELBO

Hence, simplified ELBO can be written as

$$\begin{aligned}\tilde{\mathcal{L}}_{ELBO}(\phi,\psi) &= -\|x-\mu^{\psi}(z)\|_2^2 - D_{KL}(q_{\phi}(z|x), p(z)) + \text{const}\\ &= -\|x-\mu^{\psi}(z)\|_2^2 - \frac{1}{2}\left(\|\mu^{\phi}(x)\|_2^2 + \|\sigma^{\phi}(x)\|_2^2 - 2\sum_{k=1}^{p}\log\sigma_k^{\phi}(x)\right) + \text{const.}\end{aligned}$$

Simplified ELBO

Hence, simplified ELBO can be written as

$$\begin{aligned}\tilde{\mathcal{L}}_{ELBO}(\phi, \psi) &= -\|x - \mu^{\psi}(z)\|_2^2 - D_{KL}(q_{\phi}(z|x), p(z)) + \text{const} \\ &= -\|x - \mu^{\psi}(z)\|_2^2 - \frac{1}{2}\left(\|\mu^{\phi}(x)\|_2^2 + \|\sigma^{\phi}(x)\|_2^2 - 2\sum_{k=1}^{p} \log \sigma_k^{\phi}(x)\right) + \text{const.}\end{aligned}$$

Gradient w.r.t. ψ is straightforward. How to compute the gradient w.r.t. ϕ?

Reparametrization trick

If we first sample $z \sim q_\phi(z|x)$ and then take the gradient, $\nabla_\phi \|x - \mu^\psi(z)\|_2^2 = 0$.

Not very useful!

Reparametrization trick

If we first sample $z \sim q_\phi(z|x)$ and then take the gradient, $\nabla_\phi \|x - \mu^\psi(z)\|_2^2 = 0$.

Not very useful!

Let's take one step back.

How to compute the gradient of the expected reconstruction error?

$$\nabla_\phi \mathbb{E}_{q_\phi(z|x)}[\log p_\psi(x|z)] \neq \mathbb{E}_{q_\phi(z|x)}[\nabla_\phi \log p_\psi(x|z)].$$

Reparametrization trick

Since z is Gaussian, one can reparameterize it as

$$z(\epsilon) = \mu^{\phi}(x) + \sigma^{\phi}(x) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I_p)$$

Hence,

$$\begin{aligned}
\nabla_{\phi} \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\psi}(x|z)] &= \nabla_{\phi} \mathbb{E}_{p(\epsilon)}[\log p_{\psi}(x|z(\epsilon))] \\
&\approx \nabla_{\phi} \left\| x - \mu^{\psi}\left(\mu^{\phi}(x) + \sigma^{\phi}(x) \odot \epsilon\right) \right\|_2^2,
\end{aligned}$$

for a randomly sampled ϵ.

Let's put everything together...

It leads to the following loss function:

$$\mathcal{L}_{VAE}(\phi, \psi) = \sum_i \left\| x_i - \mu^{\psi}(z_i) \right\|_2^2 + \gamma(\phi),$$

$$\gamma(\phi) = \frac{1}{2} \sum_i \left(\left\| \mu^{\phi}(x_i) \right\|_2^2 + \left\| \sigma^{\phi}(x_i) \right\|_2^2 - 2 \sum_{k=1}^{p} \log \sigma_k^{\phi}(x_i) \right),$$

where $z_i = \mu^{\phi}(x_i) + \sigma^{\phi}(x_i) \odot \epsilon$, for a randomly sampled ϵ.

Special case

Assume $\gamma = 0$ and $\sigma^{\phi} = 0$, $\mathcal{L}_{VAE}$ will be reduced to

$$\left\| x_i - \mu^{\psi}(\mu^{\phi}(x_i)) \right\|_2^2 .$$

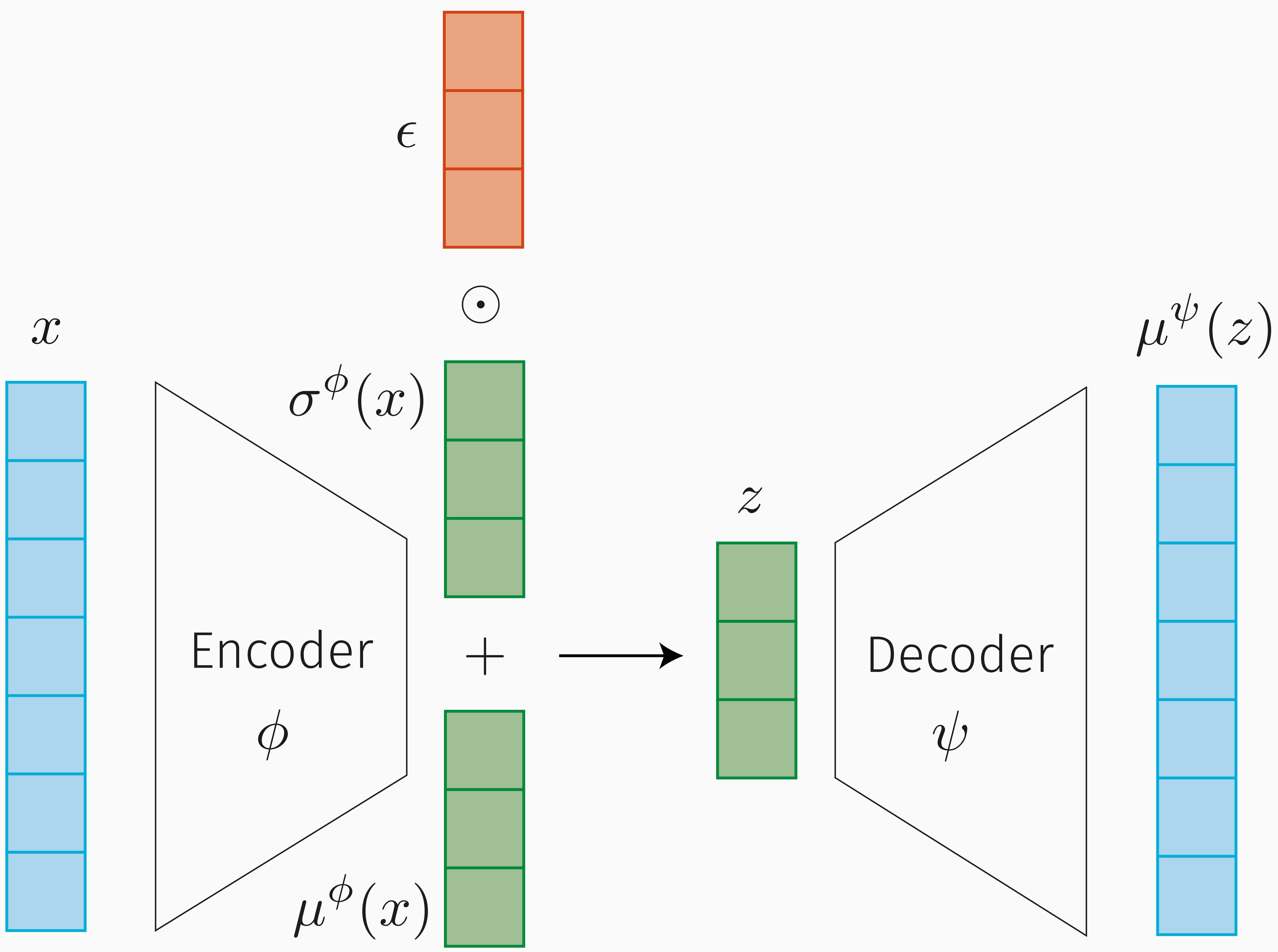
Special case

Assume $\gamma = 0$ and $\sigma^{\phi} = 0$, $\mathcal{L}_{VAE}$ will be reduced to

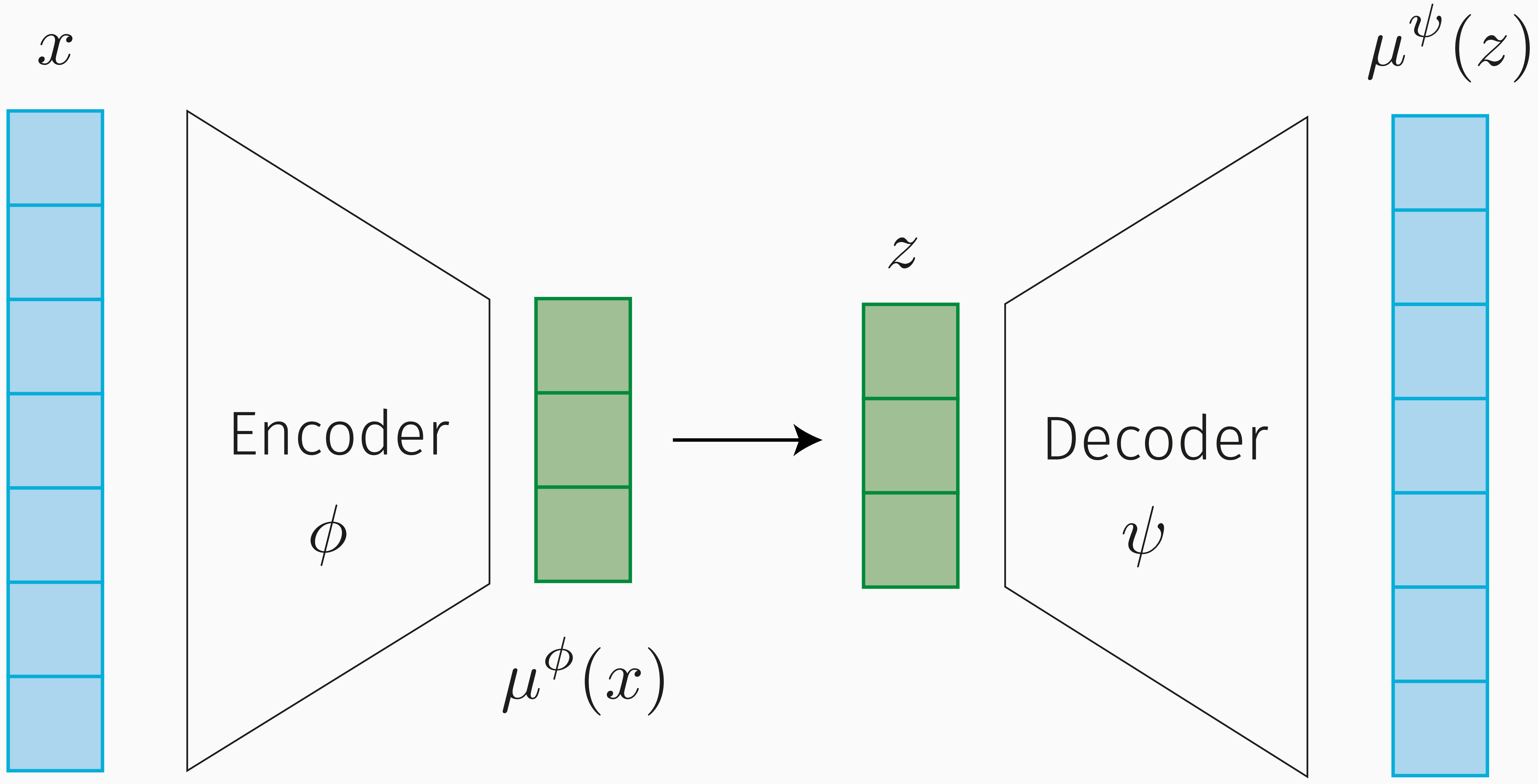
$$\left\| x_i - \mu^{\psi}(\mu^{\phi}(x_i)) \right\|_2^2 .$$

Standard autoencoder!

Special case



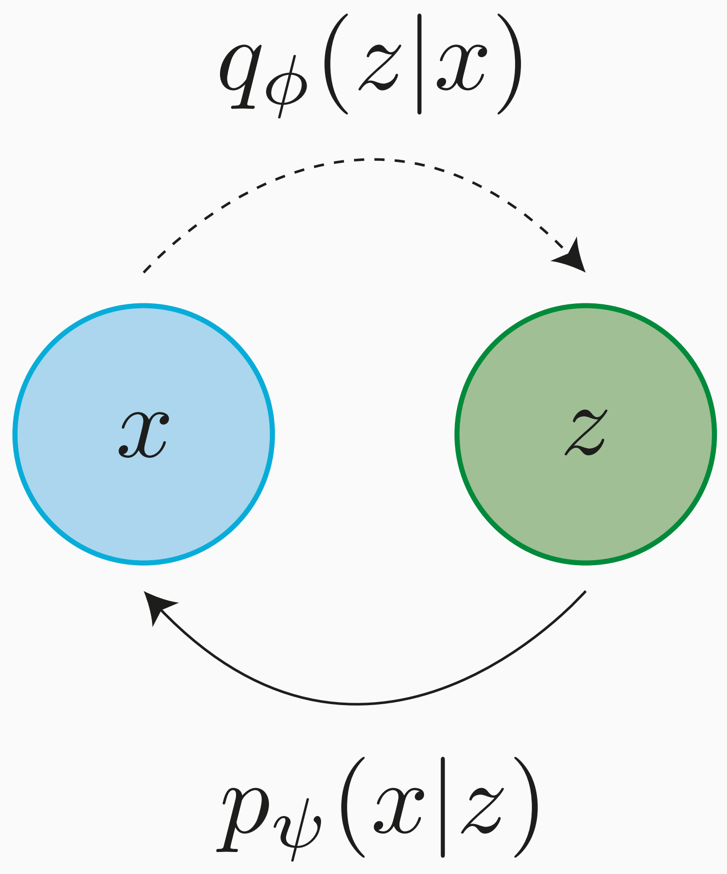
Special case



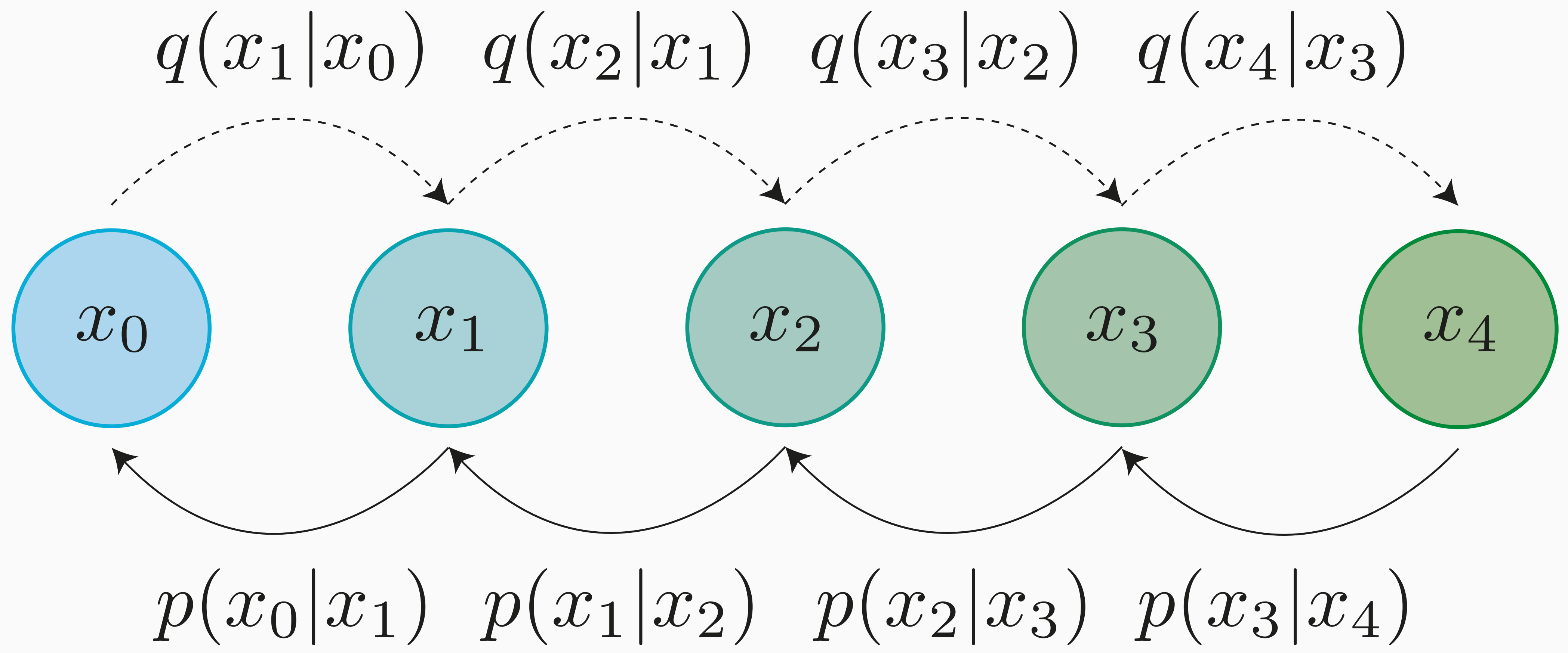
Diffusion Models

Hierarchical VAEs

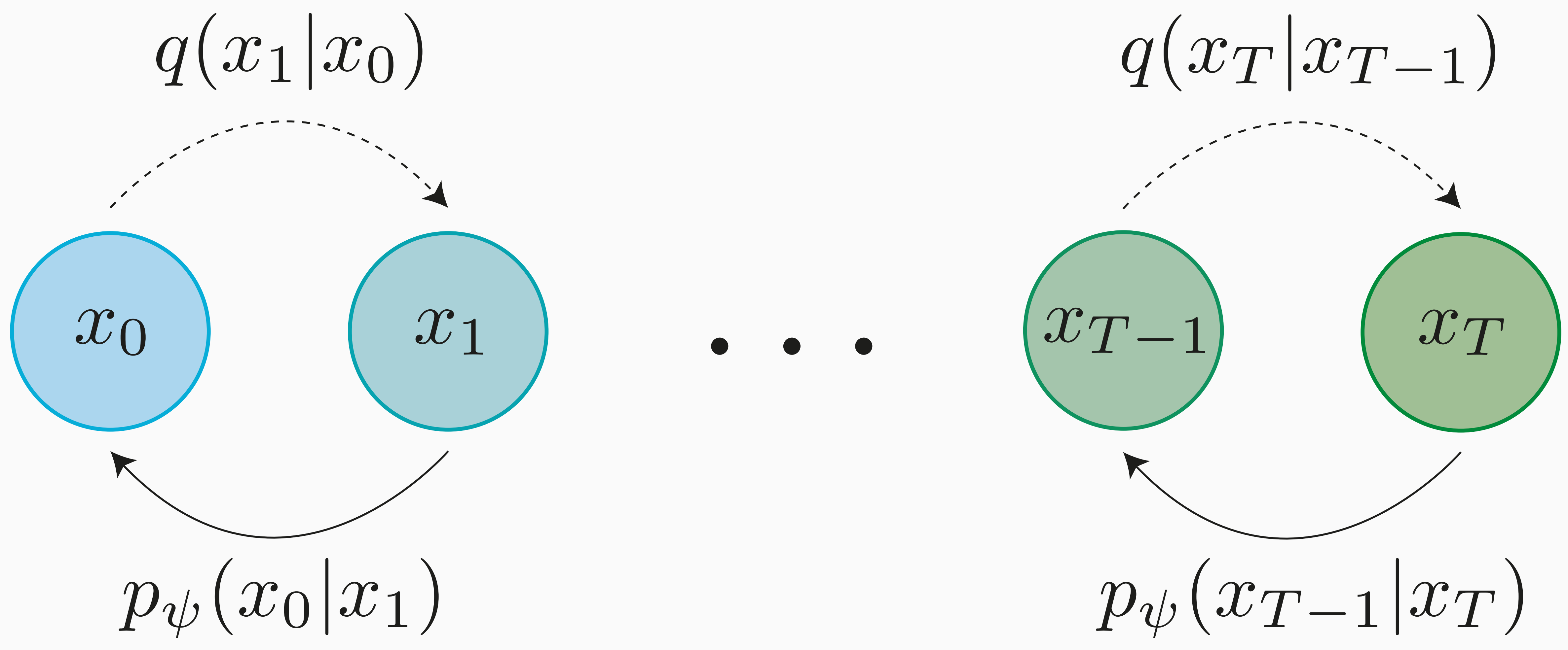
Standard VAE



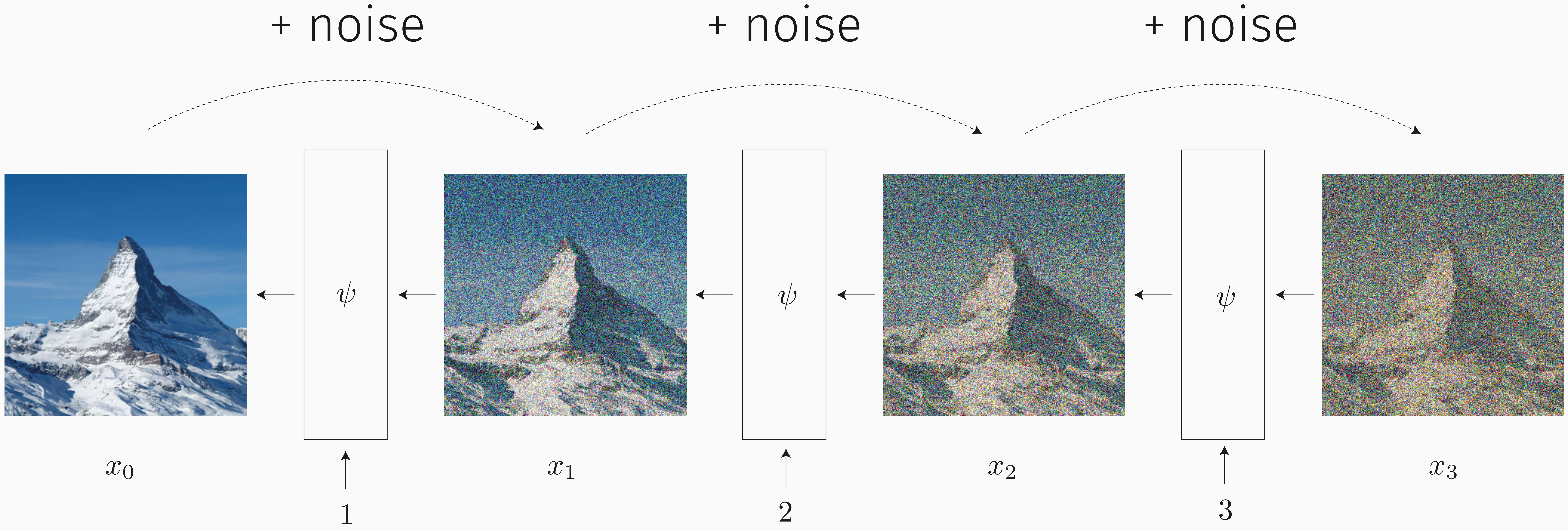
Hierarchical VAE



Denoising Diffusion Probabilistic Models (DDPM)



Denoising Diffusion Probabilistic Models (DDPM)



Learn to denoise!

Denoising Diffusion Probabilistic Models (DDPM)

The **forward process** is non-learnable; that, q has a fixed distribution. The **reverse process** is parametrized. In particular,

$$q(x_i|x_{i-1}) = \mathcal{N}\left(x_i; \sqrt{1-\beta_i}x_{i-1}, \beta_i I\right),$$
$$p_\psi(x_{i-1}|x_i) = \mathcal{N}\left(x_{i-1}; \mu^\psi(x_i, i), \beta_i I\right),$$
$$p(x_T) = \mathcal{N}(0, I).$$

for $0 \leq i \leq T$.

ELBO for DDPM I

Let's rewrite ELBO for DDPM ($z = x_{1:T}$ and $x = x_0$).

$$ELBO_{DDPM} = \mathbb{E}_{q(x_{1:T}|x_0)} \left[\log p_\psi(x_{1:T}, x_0) - \log q(x_{1:T}|x_0)\right].$$

Since

$$q(x_{1:T}|x_0) = \prod_{i=1}^{T} q(x_i|x_{i-1}),$$

$$p_\psi(x_{1:T}, x_0) = p(x_T) \prod_{i=1}^{T} p_\psi(x_{i-1}|x_i),$$

we can expand $ELBO_{DDPM}$ as

$$\mathbb{E}_{q(x_{1:T}|x_0)} \left[\log p(x_T) + \sum_{i=1}^{T} \log p_\psi(x_{i-1}|x_i) - \log q(x_i|x_{i-1})\right]$$

ELBO for DDPM II

We can use Bayes rule

$$q(x_i|x_{i-1}) = q(x_i|x_{i-1}, x_0) = \frac{q(x_{i-1}|x_i, x_0)q(x_i|x_0)}{q(x_{i-1}|x_0)},$$

and rewrite ELBO as

$$\mathbb{E}_{q(x_{1:T}|x_0)}\left[\log p(x_T) + \sum_{i=1}^{T} \log p_\psi(x_{i-1}|x_i) + \log q(x_{i-1}|x_0) - \log q(x_{i-1}|x_i, x_0) - \log q(x_i|x_0)\right]$$

ELBO for DDPM III

Using KL-divergence we can make it look a bit nicer:

$$\mathbb{E}_{q(x_{1:T}|x_0)}\left[\underbrace{\log p_\psi(x_0|x_1)}_{L_0} - \underbrace{D_{KL}(q(x_T|x_0), p(x_T))}_{L_T} - \sum_{i=2}^{T-1} \underbrace{D_{KL}(q(x_{i-1}|x_i, x_0), p_\psi(x_{i-1}|x_i))}_{L_{i-1}}\right]$$

L_T measures the Gaussianity of x_T.

L_0 is the final (reverse process) reconstruction loss.

L_{i-1} are KL diveregences between Gaussians too, measuring the denoising quality of step $i-1$ (of reverse process).

ELBO for DDPM III

Using KL-divergence we can make it look a bit nicer:

$$\mathbb{E}_{q(x_{1:T}|x_0)}\left[\underbrace{\log p_\psi(x_0|x_1)}_{L_0} - \underbrace{D_{KL}(q(x_T|x_0), p(x_T))}_{L_T} - \sum_{i=2}^{T-1} \underbrace{D_{KL}(q(x_{i-1}|x_i, x_0), p_\psi(x_{i-1}|x_i))}_{L_{i-1}}\right]$$

L_T measures the Gaussianity of x_T.

L_0 is the final (reverse process) reconstruction loss.

L_{i-1} are KL diveregences between Gaussians too, measuring the denoising quality of step $i-1$ (of reverse process).

Only L_{i-1}'s and L_0 are trainable! L_0 can simply be $L_0 = \|x_0 - \mu^\psi(x_1, 1)\|_2^2$. How about L_{i-1}?

First step $q(x_i|x_0)$

Note that

$$
\begin{aligned}
x_1 &= \sqrt{1-\beta_1}x_0 + \sqrt{\beta_1}\epsilon_1 \\
x_2 &= \sqrt{1-\beta_2}x_1 + \sqrt{\beta_2}\epsilon_2 \overset{d}{=} \sqrt{1-\beta_2}\sqrt{1-\beta_1}x_0 + \sqrt{1-(1-\beta_1)(1-\beta_2)}\tilde{\epsilon}_2 \\
&\vdots \\
x_i &\overset{d}{=} \prod_{j=1}^{i}\sqrt{1-\beta_j}x_0 + \sqrt{1-\prod_{j=1}^{i}(1-\beta_j)}\tilde{\epsilon}_i
\end{aligned}
$$

Let $\bar{\alpha}_i := \prod_{j=1}^{i} 1-\beta_j$, we get

$$q(x_i|x_0) = \mathcal{N}(x_i, \sqrt{\bar{\alpha}_i}x_0, (1-\bar{\alpha}_i)I)$$

Second step $q(x_{i-1}|x_i, x_0)$

Recall that

$$q(x_{i-1}|x_i, x_0) \propto q(x_i|x_{i-1}, x_0)q(x_{i-1}|x_0)$$

is a multiplication of Gaussians.

Second step $q(x_{i-1}|x_i, x_0)$

Recall that

$$q(x_{i-1}|x_i, x_0) \propto q(x_i|x_{i-1}, x_0)q(x_{i-1}|x_0)$$

is a multiplication of Gaussians.

Second step $q(x_{i-1}|x_i, x_0)$

Recall that

$$q(x_{i-1}|x_i, x_0) \propto q(x_i|x_{i-1}, x_0) q(x_{i-1}|x_0)$$

is a multiplication of Gaussians.

Multiplication of two Gaussian distributions

If $p_1(x) = \mathcal{N}(x; \mu_1; \sigma_1^2 I)$ and $p_2(x) = \mathcal{N}(x; \mu_2; \sigma_2^2 I)$, then

$$p_1(x) p_2(x) = \mathcal{N}\left(x; \frac{\sigma_2^2 \mu_1 + \sigma_1^2 \mu_2}{\sigma_1^2 + \sigma_2^2}, \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} I\right).$$

Second step $q(x_{i-1}|x_i, x_0)$ (cont'd)

Substituting as follows

$$\mu_1 = \frac{1}{\sqrt{1-\beta_i}} x_i, \sigma_1^2 = \frac{\beta_i}{1-\beta_i} \quad \mu_2 = \sqrt{\bar{\alpha}_{i-1}} x_0, \sigma_2^2 = 1 - \bar{\alpha}_{i-1}$$

we get,

$$q(x_{i-1}|x_i, x_0) = \mathcal{N}(x_{i-1}; \tilde{\mu}(x_i, x_0, i), \tilde{\beta}_i I),$$

where

$$\tilde{\mu}(x_i, x_0, i) = \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i} x_0 + \frac{\sqrt{1-\beta_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i} x_i$$

$$\tilde{\beta}_i = \frac{1-\bar{\alpha}_{i-1}}{1-\bar{\alpha}_i} \beta_i$$

Third step L_{i-1}

Using the fact that $q(x_{i-1}|x_i, x_0)$ and $p_\psi(x_{i-1}|x_i)$ are Gaussians

$$\begin{aligned}
\mathbb{E}_q L_{i-1} &= \mathbb{E}_q D_{KL}(q(x_{i-1}|x_i, x_0), p_\psi(x_{i-1}|x_i)) \\
&= \frac{1}{2\beta_i}\mathbb{E}_q[\|\tilde{\mu}(x_i, x_0, i) - \mu^\psi(x_i, i)\|^2] + \text{ const.}
\end{aligned}$$

One can already use L_{i-1} in this form to optimizie DDPM; however, a reparametrized version works better in practice.

Reparametrizing L_{i-1}

We can instead use the following parametrization

$$x_i(x_0, \epsilon) = \sqrt{\bar{\alpha}_i} x_0 + \sqrt{1 - \bar{\alpha}_i}\epsilon$$

and simplify $\tilde{\mu}_i$ as

$$\tilde{\mu}(x_i, x_0, i) = \frac{1}{\sqrt{1 - \beta_i}} \left(x_i - \frac{\beta_i}{\sqrt{1 - \bar{\alpha}_i}} \epsilon \right).$$

Similarly, we can reparametrize μ^ψ as

$$\mu^\psi(x_i, i) = \frac{1}{\sqrt{1 - \beta_i}} \left(x_i - \frac{\beta_i}{\sqrt{1 - \bar{\alpha}_i}} \epsilon_\psi(x_i, i) \right).$$

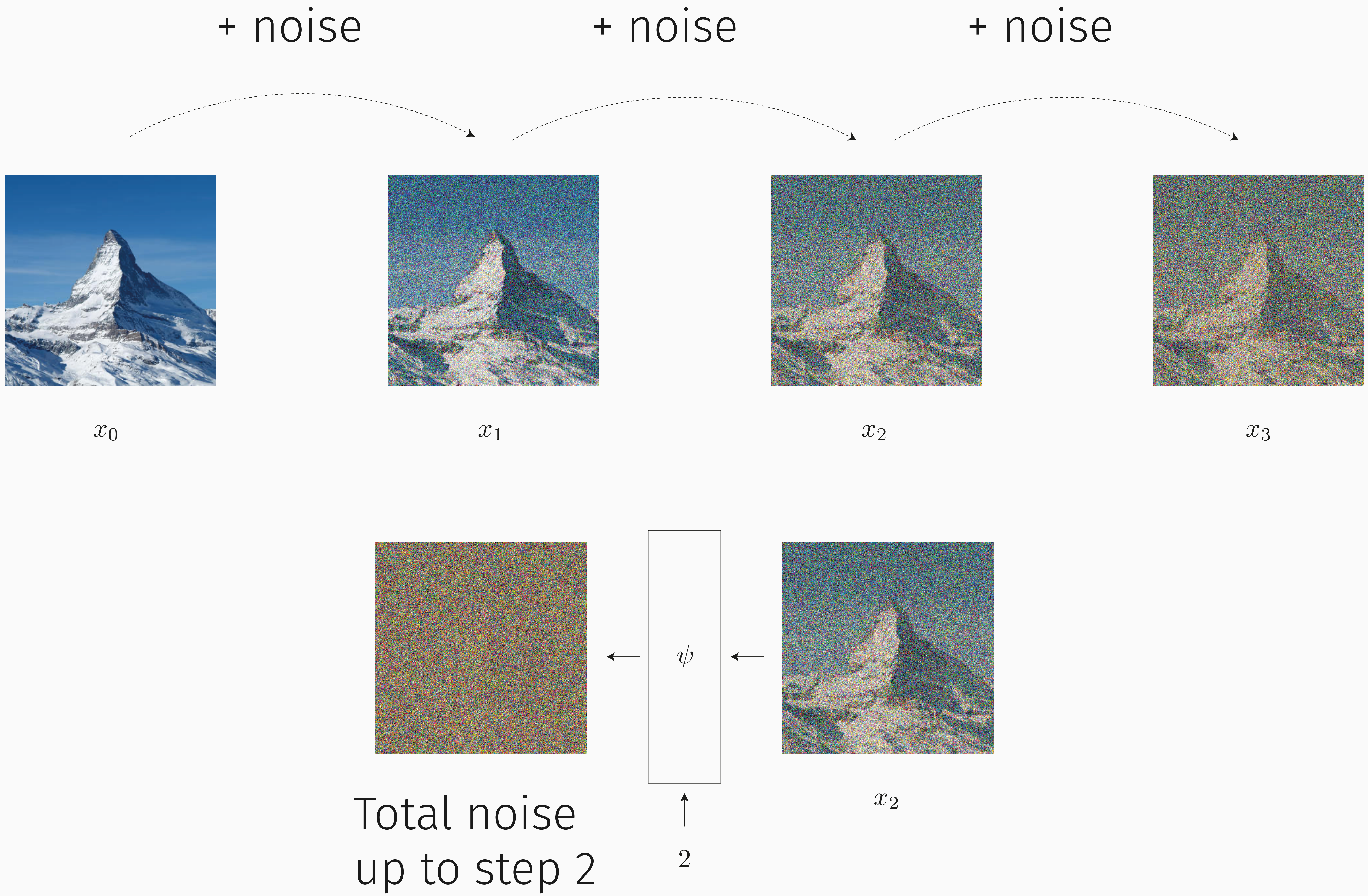
Reparameterizing L_{i-1} (cont'd)

Finally, L_{i-1} can be rewritten as

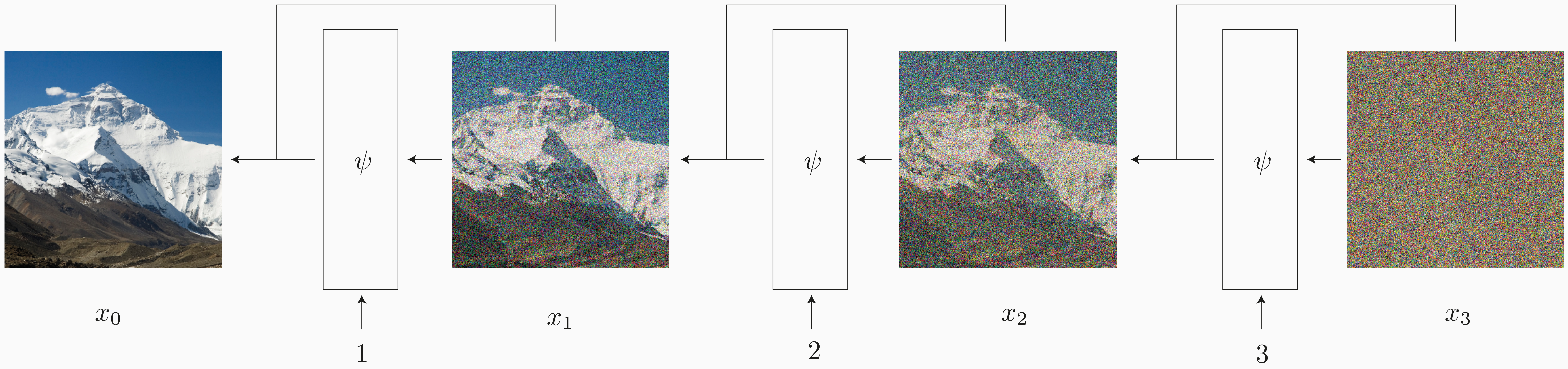
$$\mathbb{E}_\epsilon L_{i-1} = \frac{\beta_i}{2(1-\beta_i)(1-\bar{\alpha}_i)} \mathbb{E}_\epsilon [\|\epsilon - \epsilon_\psi(x_i(x_0, \epsilon), i)\|^2] + \text{ const.}$$

The parametrized model ϵ_ψ is trying to predict the total noise added to x_0 at step i.

Training



Sampling



Training and sampling DDPM

Training

1. Sample a random datapoint x_0 from the training set.
2. Pick a random $i \in \{1, 2, \ldots, T\}$.
3. Sample $\epsilon \sim \mathcal{N}(0, I)$.
4. Update the parameters ψ using GD: $\nabla_\psi L_{i-1}$.
5. Repeat until convergence.

Sampling

1. Generate a noise sample $x_T \sim (0, I)$.
2. For $i = T, \ldots, 1$: Sequentially sample the reverse conditionals $x_{i-1} \sim p_\psi(x_{i-1} | x_i)$.

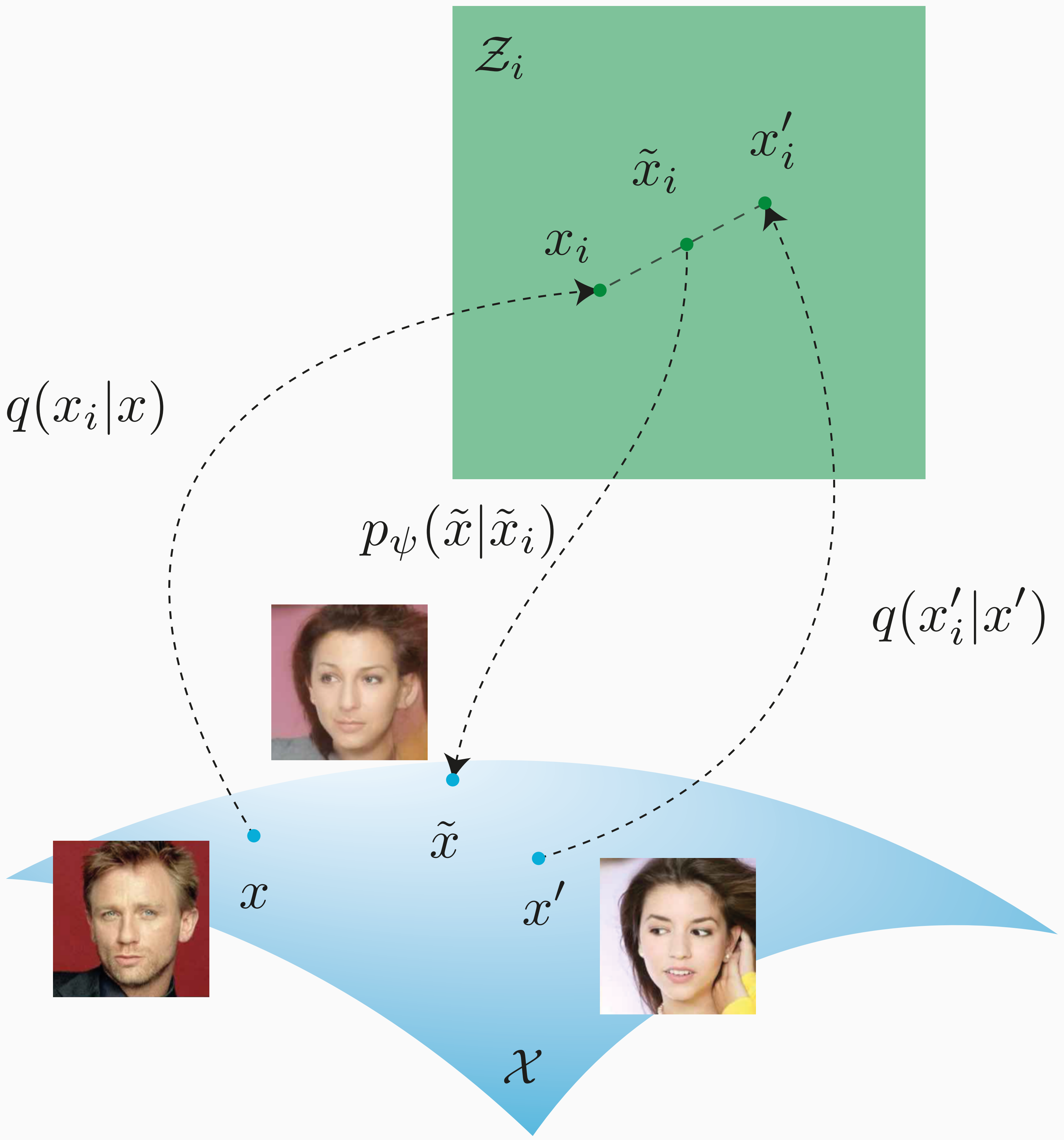
DDPM generated samples

LSUN

CelebA-HQ

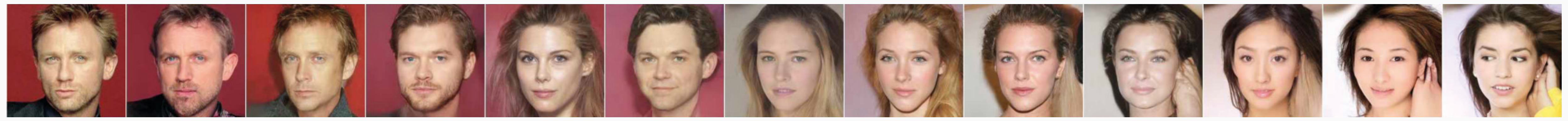
Denoising Diffusion Probabilistic Models, Ho et al., *NeurIPS 2020*.

Quiz interpolation using DDPM

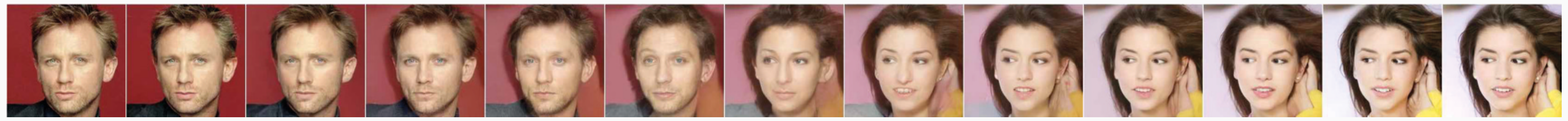


Interpolation is done in

$\mathcal{Z}_i$



$\mathcal{Z}_j$

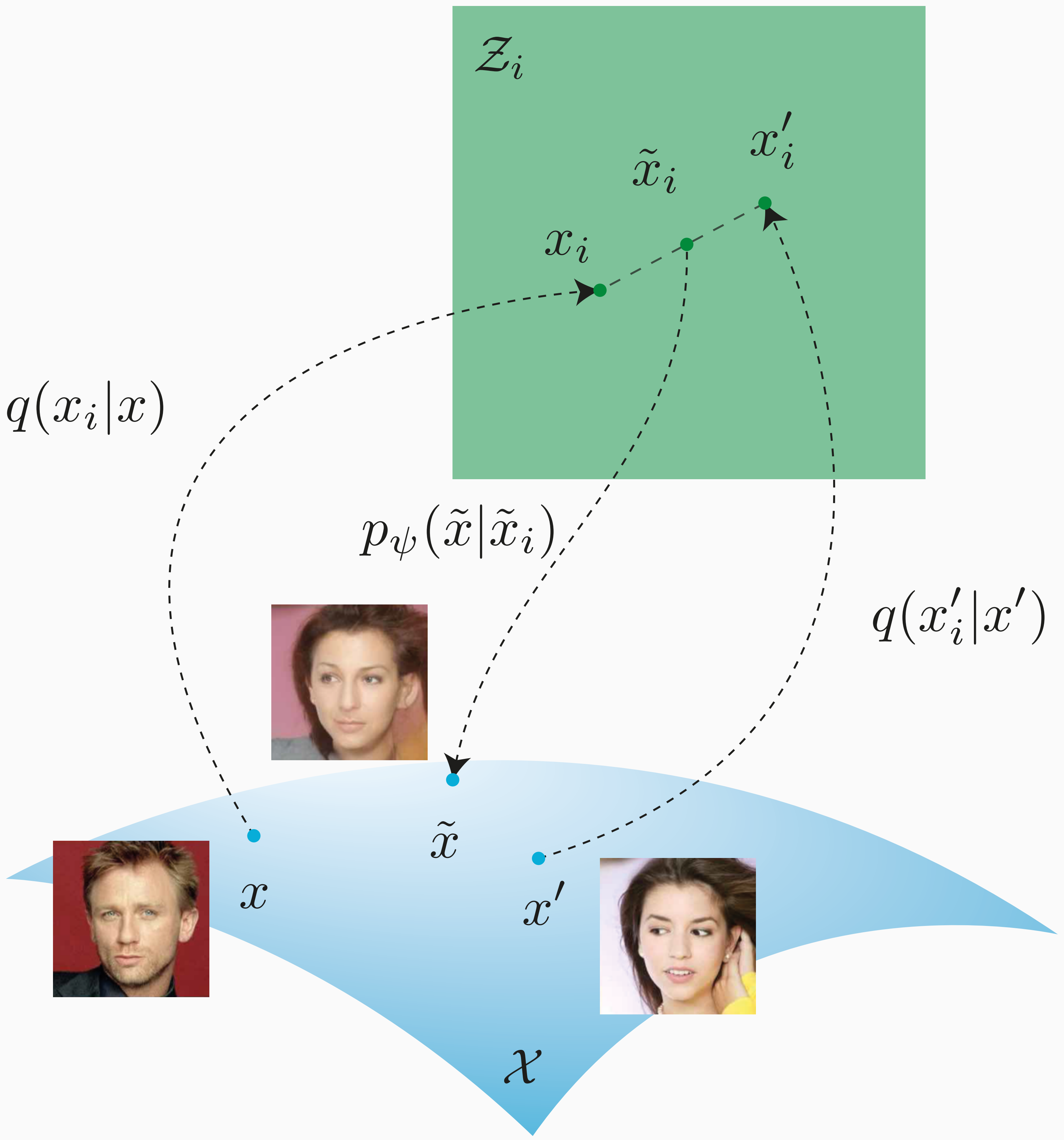


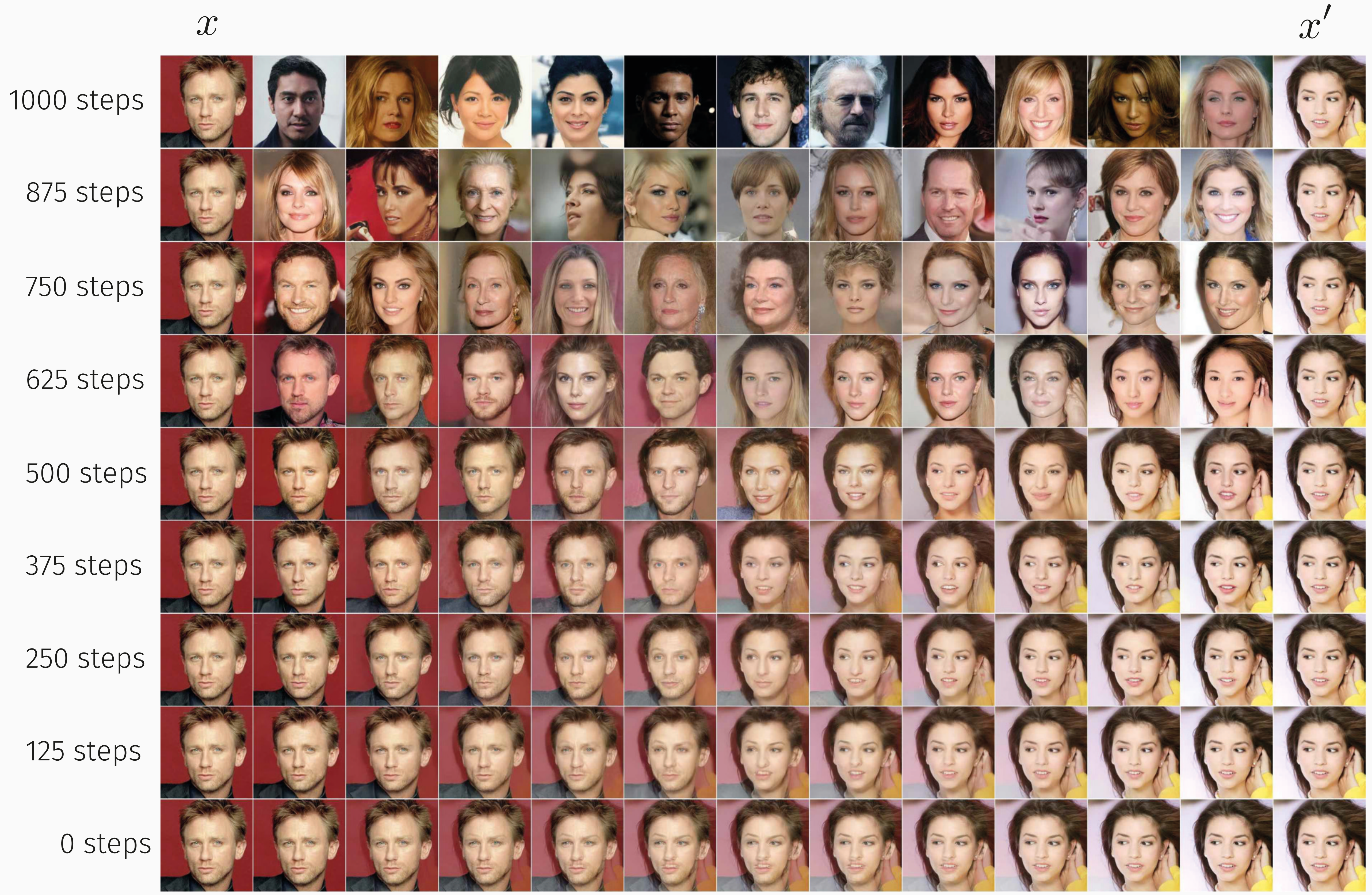
- $i > j$?

or

- $j > i$?

Quiz interpolation using DDPM





Denoising Diffusion Probabilistic Models, Ho et al., *NeurIPS 2020*.

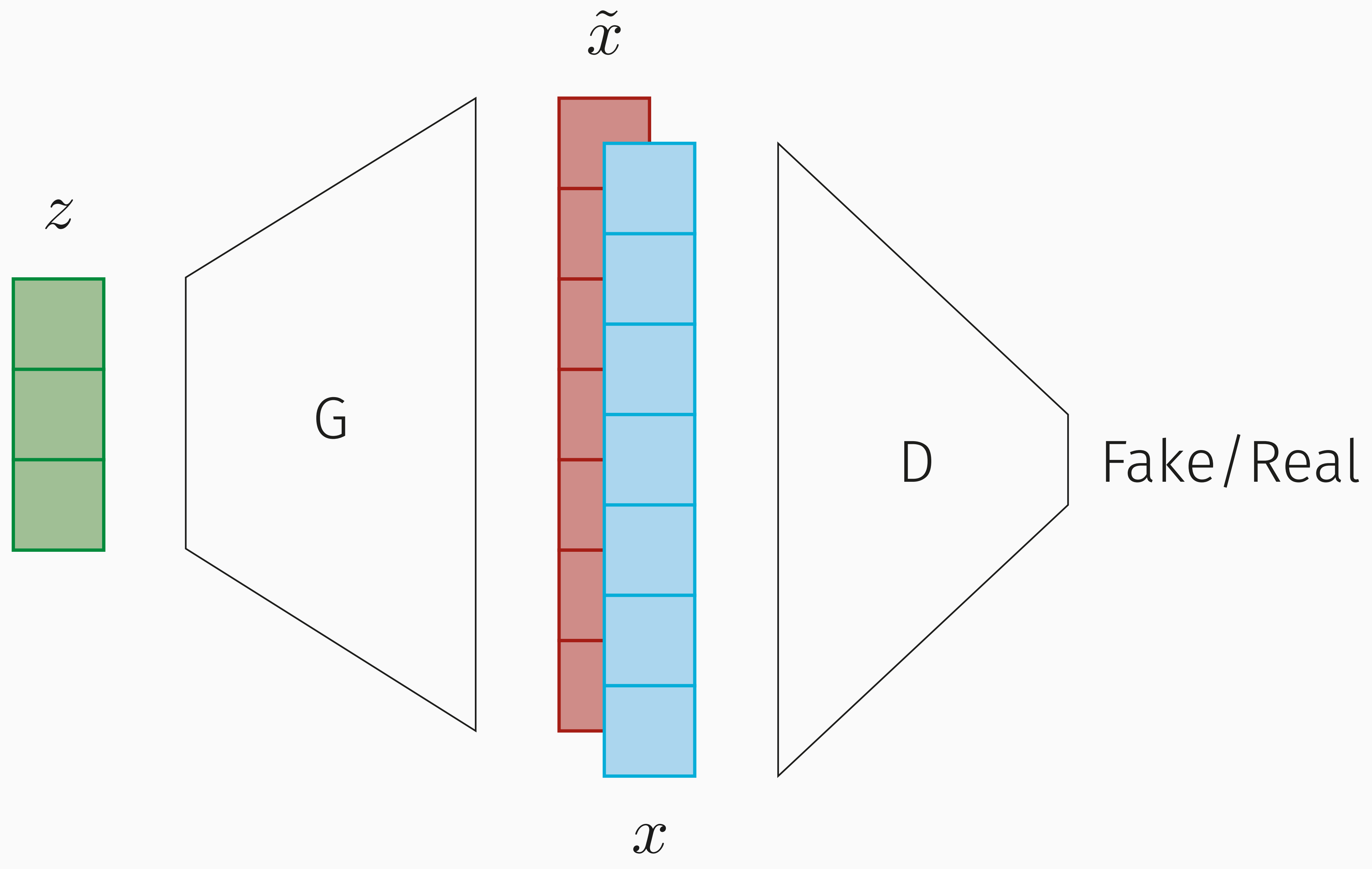
Generative Adversarial Networks

Generative Adversarial Network (GAN)

Generative modeling as a two-player game:

Discriminator D learns to distinguish *training (real)* from *generated (fake)* samples.

Generator G generates data that can fool the discriminator.



GAN loss functions (just a few examples)

Original GAN

$$\min_G \max_D \mathbb{E}_{p(x)}[\log(D(x))] + \mathbb{E}_{p(z)}[\log(1 - D(G(z)))]$$

LSGAN

$$\min_D \mathbb{E}_{p(x)}[(D(x) - 1)^2] + \mathbb{E}_{p(z)}[D(G(z))^2]$$

$$\min_G \mathbb{E}_{p(z)}[(D(G(z)) - 1)^2]$$

Wasserstein GAN

$$\min_G \max_{D \in 1\text{-Lip}} \mathbb{E}_{p(x)}[D(x)] - \mathbb{E}_{p(z)}[D(G(z))]$$

Intriguing result...

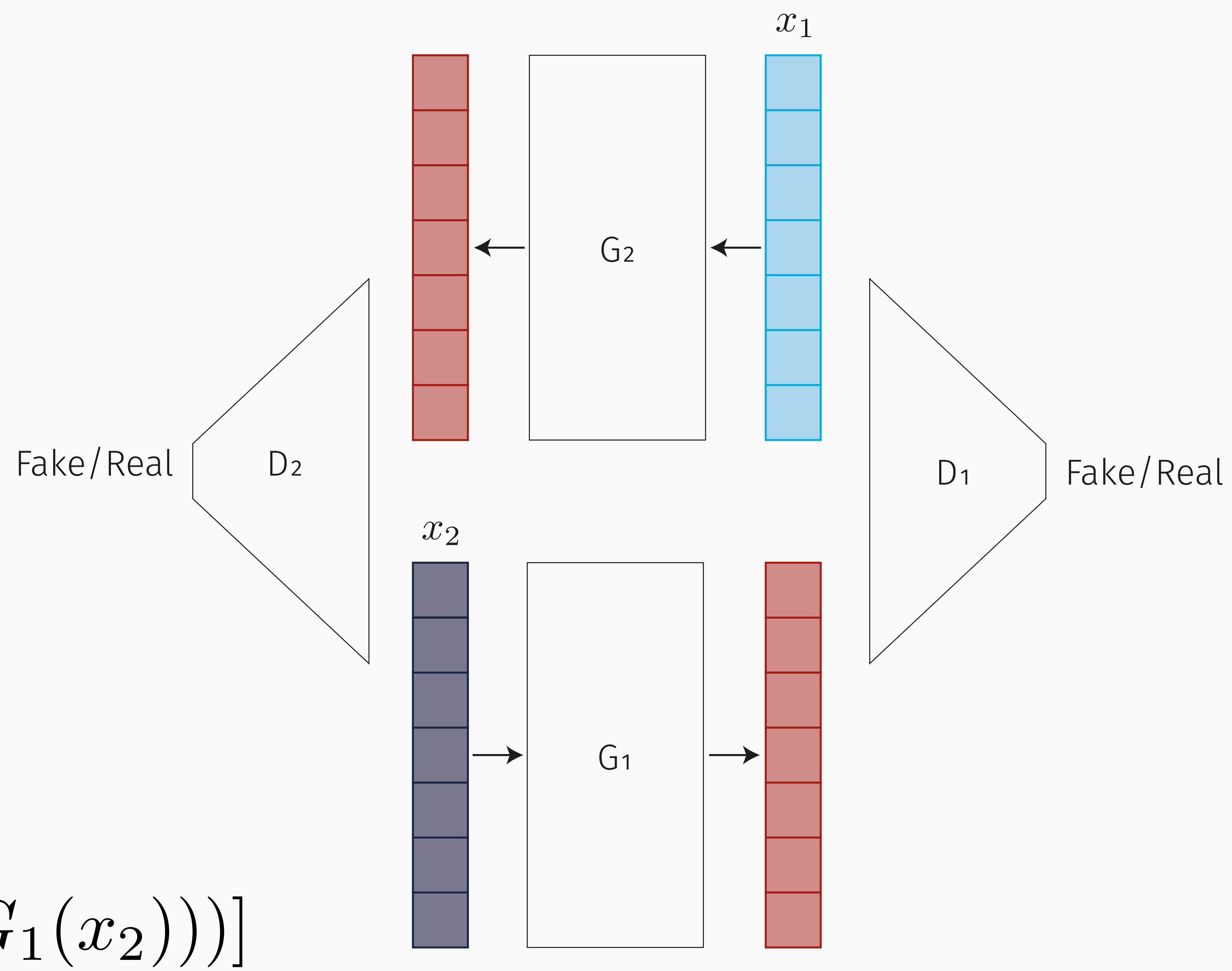
“We find that most models can reach similar scores with enough hyperparameter optimization and random restarts.”

Are GANs Created Equal? A Large-Scale Study, Lucic et al., *NeurIPS 2018*.

There is a lot more to the story...

Cycle GAN

A neat idea to "learn to transform" based on *unpaired* data!



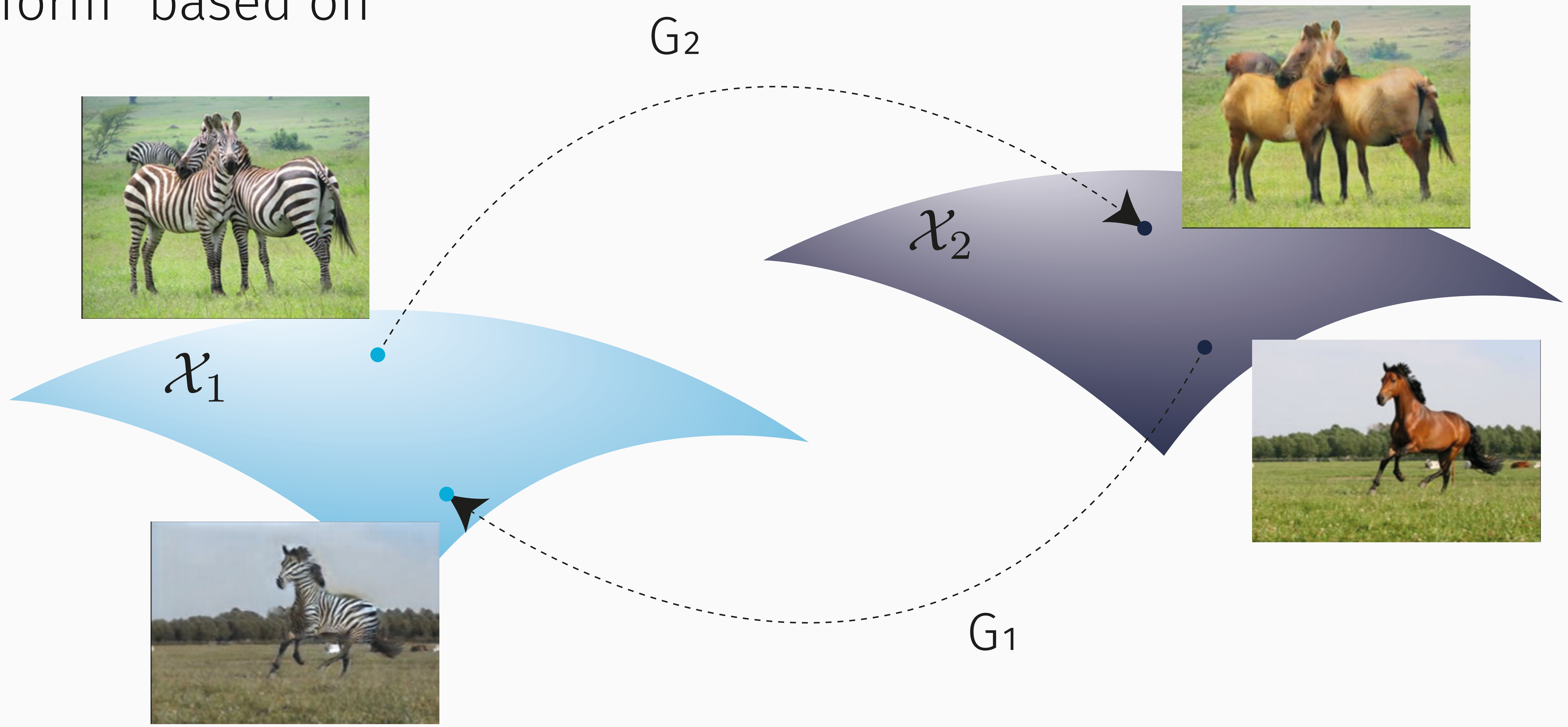
$$\min_{G_1,G_2}\max_{D_1,D_2}\quad \mathbb{E}_{p(x_1)}[\log(D_1(x_1))]+\mathbb{E}_{p(x_2)}[\log(1-D_1(G_1(x_2)))]$$
$$+\mathbb{E}_{p(x_2)}[\log(D_2(x_2))]+\mathbb{E}_{p(x_1)}[\log(1-D_2(G_2(x_1)))]$$
$$+\mathbb{E}_{p(x_2)}[||G_2(G_1(x_2))-x_2||]+\mathbb{E}_{p(x_1)}[||G_1(G_2(x_1))-x_1||]$$

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, Zhu et al., *ICCV 2017*.

There is a lot more to the story...

Cycle GAN

A neat idea to "learn to transform" based on *unpaired* data!



Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, Zhu et al., *ICCV 2017*.

Silent neurons swirl,
Creative output unfolds,
Generative art born.

— ChatGPT