CS492D: Diffusion Models and Their Applications

Denoising Diffusion Probabilistic Models 1

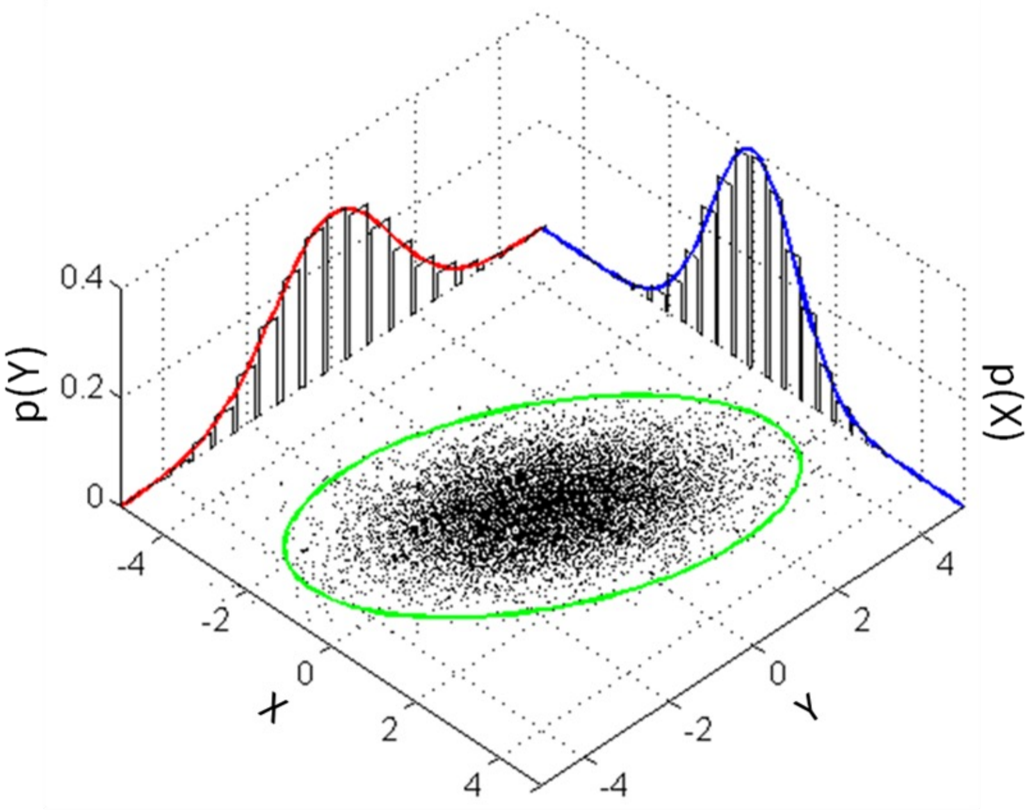
LECTURE 3
MINHYUK SUNG

Fall 2024
KAIST

Previously in CS492D

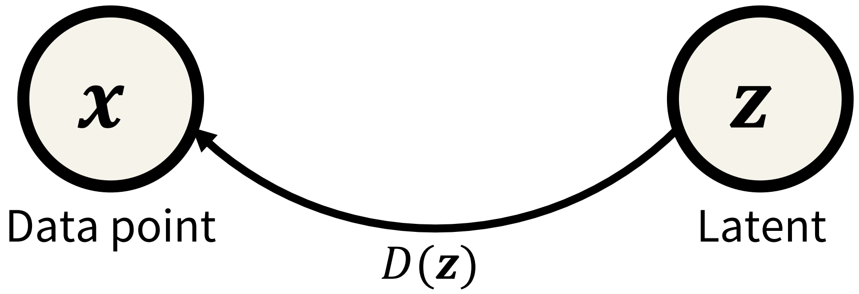
From a **statistical perspective**, we will view a dataset as

- there being a probability distribution of the data, and
- the given points are samples from the probability distribution.



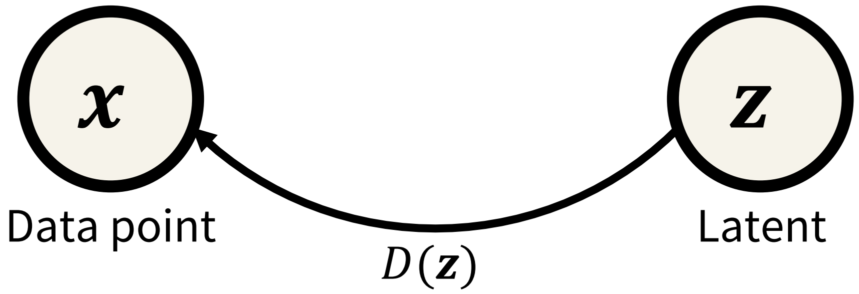
Previously in CS492D

- Map a *simple* distribution $p(\boldsymbol{z})$ (e.g., a standard normal distribution $\mathcal{N}(\boldsymbol{x}; \mathbf{0}, \mathbf{I})$) to the data distribution $p(\boldsymbol{x})$.
  - $\boldsymbol{z}$: Latent variable
  - $p(\boldsymbol{z})$: Latent distribution
- Sample from $p(\boldsymbol{z})$ and map it to a data point.



Previously in CS492D

- How to map a latent distribution $p(\boldsymbol{z})$ to the data distribution $p(\boldsymbol{x})$ using a neural network?
- How to guarantee that a latent is mapped to a data point of the data distribution?
- We need an additional neural network.

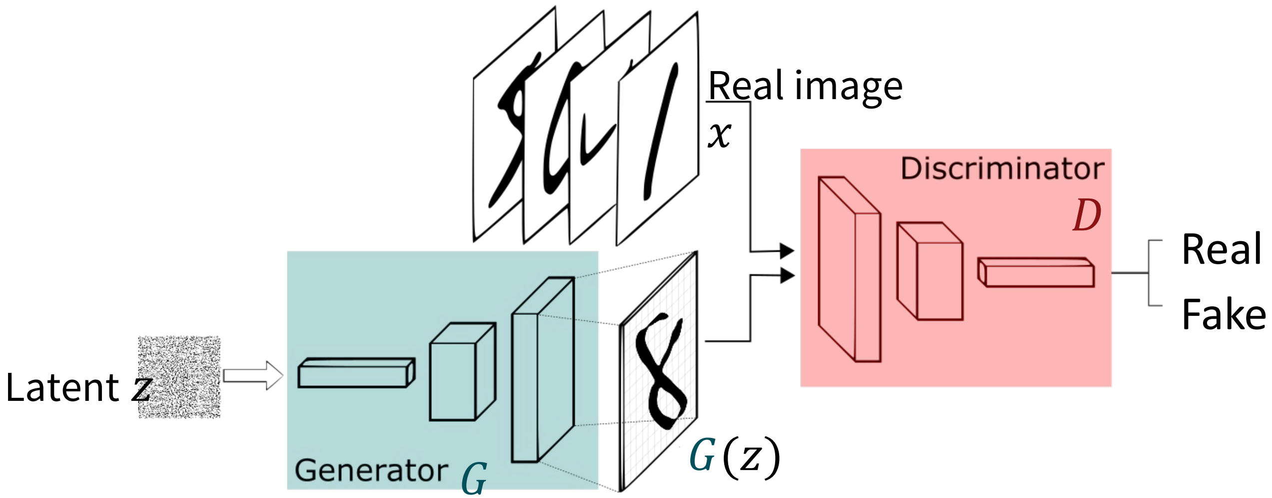


Previously in CS492D

Generative Adversarial Network (GAN)

Introduce a discriminator and have it compete with the generator (decoder).

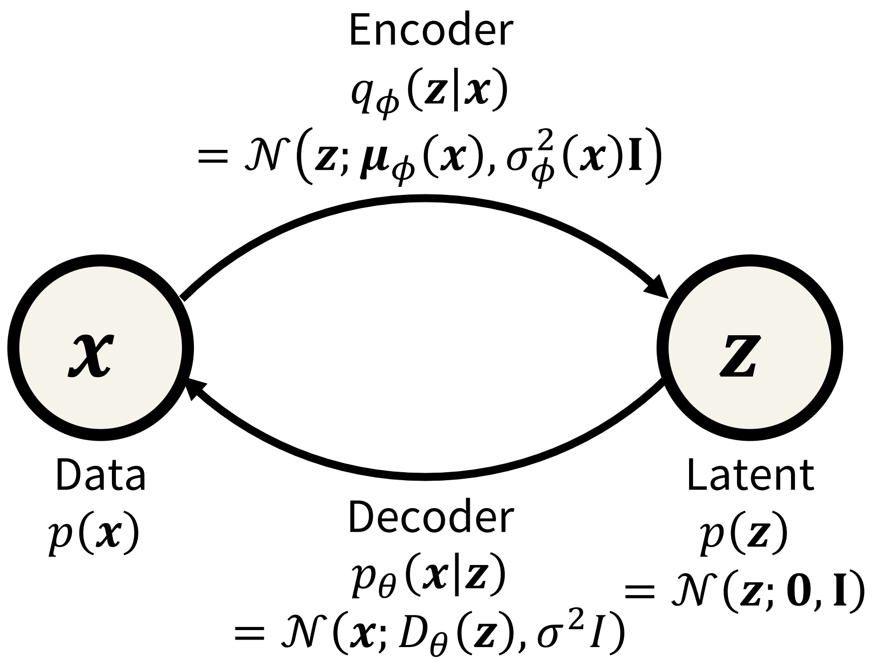
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$



Previously in CS492D

Variational Autoencoder (VAE)

Introduce an encoder to learn a proxy of the posterior distribution.



Previously in CS492D

Basics

- Maginal distribution
- Expected value
- Bayes' rule
- Kullback–Leibler (KL) Divergence
- Jensen's inequality

Previously in CS492D

Bayes' Rule

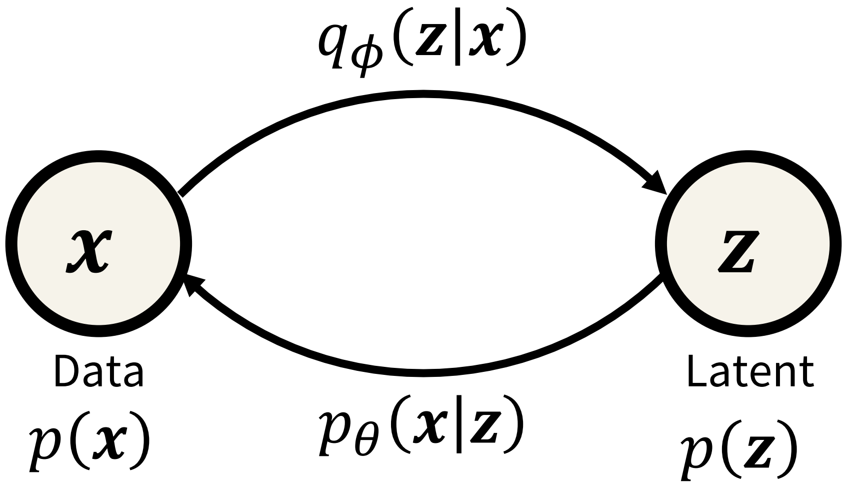
Likelihood
Decoder

Posterior
Encoder

$$\boxed{p(\boldsymbol{z}|\boldsymbol{x})} = \frac{\boxed{p(\boldsymbol{x}|\boldsymbol{z})}\boxed{p(\boldsymbol{z})}}{\boxed{p(\boldsymbol{x})}}$$

Prior
Latent

Marginal
Data



Previously in CS492D

Evidence Lower Bound (ELBO)

- We cannot directly maximize $p(\boldsymbol{x}) = \frac{p(\boldsymbol{x},\boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{x})}$ since $p(\boldsymbol{z}|\boldsymbol{x})$ is unknown.

- Let's maximize lower bound of $\log p(\boldsymbol{x})$:

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

Evidence Lower Bound (ELBO)

Let's decompose ELBO:

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

$$= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})] - \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})}\right]$$

$$= \boxed{\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})]} - \boxed{D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z})\right)}$$

Reconstruction term
to be *maximized*.

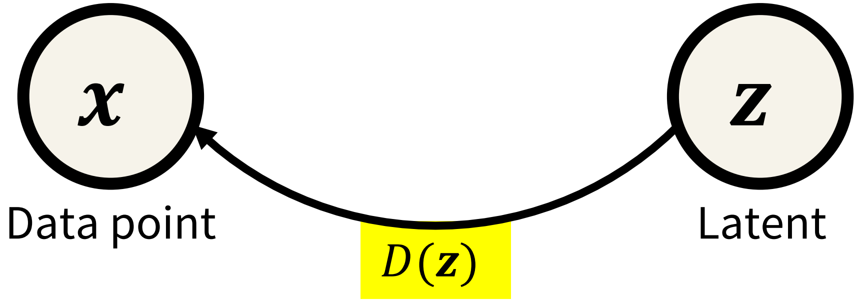
Prior matching term
to be *minimized*.

Back to VAE…

In VAE, we want model $p(\boldsymbol{x})$ as

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} = \int p(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}$$

where $p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; D(\boldsymbol{z}), \sigma^2 I)$ and $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; \mathbf{0}, \mathbf{I})$.

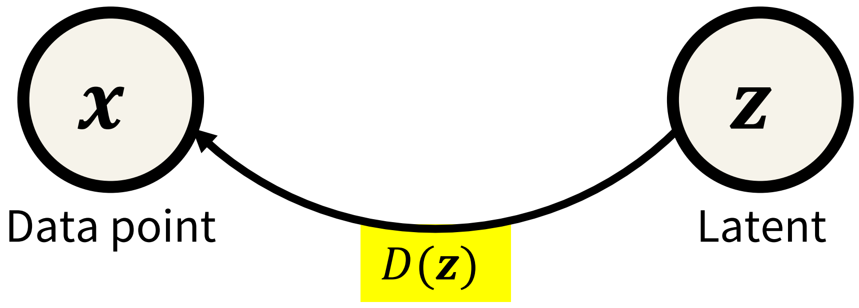


Variational Autoencoder (VAE)

In VAE, we want model $p(\boldsymbol{x})$ as

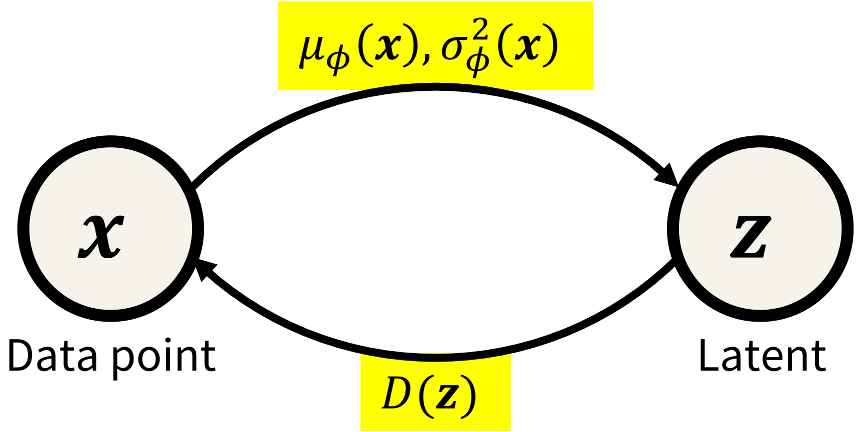
$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} = \int p_\theta(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}$$

where $p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x};\ D_\theta(\boldsymbol{z}), \sigma^2 I)$ and $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \mathbf{0}, \mathbf{I})$.



Variational Autoencoder (VAE)

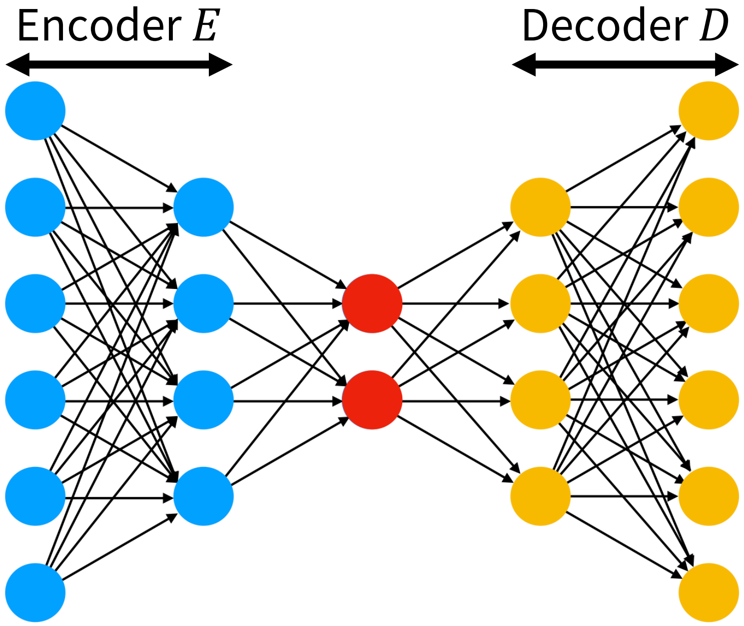
- How to maximize $p(\boldsymbol{x})$? Maximize ELBO!
- We need the proxy distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ → **Encoder**
- Let $q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}_\phi(\boldsymbol{x}), \sigma_\phi^2(\boldsymbol{x})\mathbf{I}\right)$



Variational Autoencoder (VAE)

Same with autoencoder but,

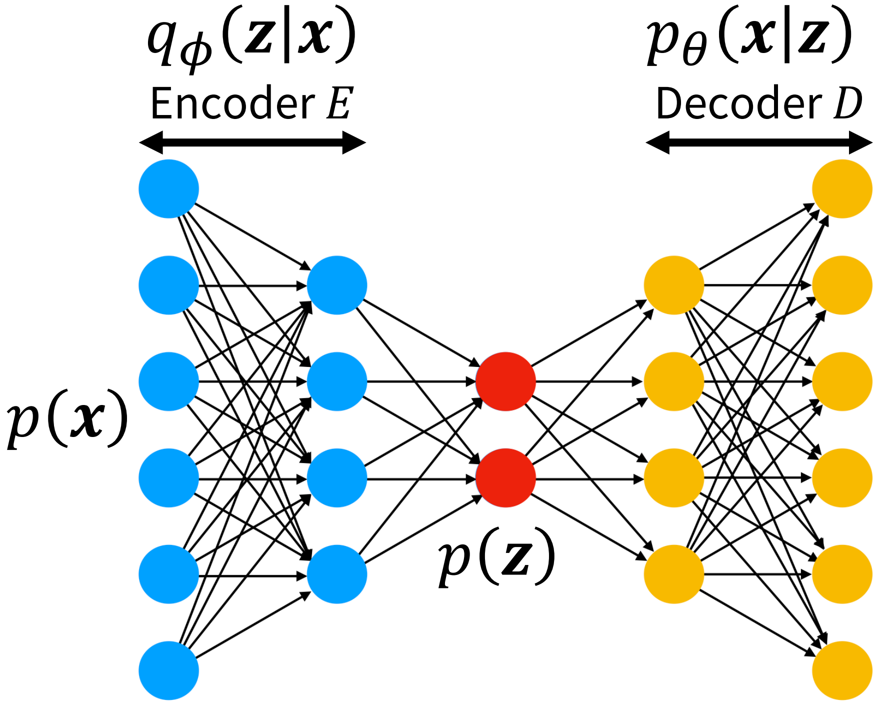
- From input $\boldsymbol{x}$, the encoder predicts $\mu_\phi(\boldsymbol{x}), \sigma^2_\phi(\boldsymbol{x})$.
- The decoder takes a sample $\boldsymbol{z} \sim \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}_\phi(\boldsymbol{x}), \sigma^2_\phi(\boldsymbol{x})\mathbf{I}\right)$ as input.



Variational Autoencoder (VAE)

Summary

Data distribution	$p(\boldsymbol{x})$
Encoder	$q_\phi(\mathbf{z}\|\boldsymbol{x}) = \mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_\phi(\boldsymbol{x}), \sigma_\phi^2(\boldsymbol{x})\mathbf{I}\right)$
Latent distribution	$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$
Decoder	$p_\theta(\boldsymbol{x}\|\mathbf{z}) = \mathcal{N}(\boldsymbol{x}; D_\theta(\mathbf{z}), \sigma^2 I)$



Training

How to maximize ELBO?

$$\underset{\theta,\phi}{\operatorname{argmax}}\, \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z})\right)$$

Approximates using a Monte Carlo estimate:

$$\underset{\theta,\phi}{\operatorname{argmax}} \frac{1}{N}\sum_{i=1}^{N} \log p_\theta\left(\boldsymbol{x}|\boldsymbol{z}^{(i)}\right) - D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z})\right)$$

where $\boldsymbol{z}^{(i)} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x})$ for the given $\boldsymbol{x}$.

[EXTRA] Monte Carlo Method

Law of Large Numbers, LLN

Where X_1, X_2, ... is iid(independent and identically distributed) RV

$$\overline{X}_n = \frac{1}{n}(X_1 + \cdots + X_n) \quad \overline{X}_n \to \mu \quad \text{as } n \to \infty.$$

$$\mathrm{E}_{\mathbf{x}\sim p}\left[f(\mathbf{x})\right] = \int_{\mathcal{X}} f(\mathbf{x})p(\mathbf{x})d\mathbf{x}$$

$$\int_{\mathcal{X}} f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \approx \frac{1}{n}\sum_{i=1}^{N} f(\mathbf{x}_i)$$

Training

How to maximize ELBO?

$$\underset{\theta,\phi}{\operatorname{argmax}}\, \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z})\right)$$

Approximates using a Monte Carlo estimate:

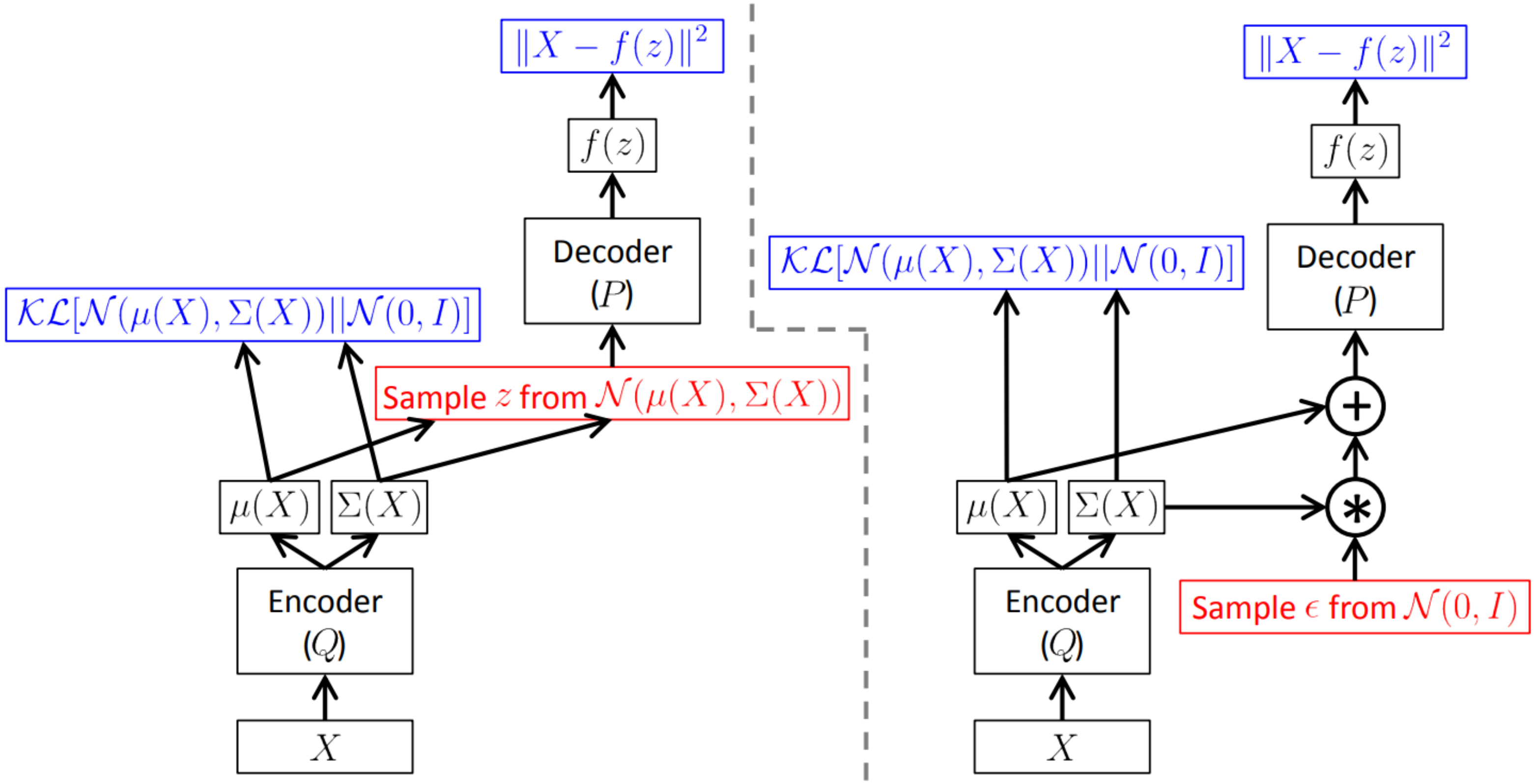
$$\underset{\theta,\phi}{\operatorname{argmax}} \frac{1}{N}\sum_{i=1}^{N} \log p_\theta\left(\boldsymbol{x}|\boldsymbol{z}^{(i)}\right) - D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z})\right)$$

Reparameterization Trick

where $\boxed{\boldsymbol{z}^{(i)} = \boldsymbol{\mu}_\phi(\boldsymbol{x}) + \sigma_\phi(\boldsymbol{x})\boldsymbol{\epsilon} \text{ where } \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I})}$.

[EXTRA] Reparameterization Trick

Not differentiable! -> Reparameterization Trick



[EXTRA] Reparameterization Trick

Why does it work?

Let's say we want to take the gradient w.r.t. θ of the following expectation,

$$\mathbb{E}_{p(z)}[f_\theta(z)]$$

$$\begin{aligned}
\nabla_\theta \mathbb{E}_{p(z)}[f_\theta(z)] &= \nabla_\theta \left[\int_z p(z) f_\theta(z) dz \right] \\
&= \int_z p(z) \left[\nabla_\theta f_\theta(z) \right] dz \\
&= \mathbb{E}_{p(z)} \left[\nabla_\theta f_\theta(z) \right]
\end{aligned}$$

[EXTRA] Reparameterization Trick

But if p is also parameterized by θ?

$$\begin{aligned}\nabla_\theta \mathbb{E}_{p(z)}[f_\theta(z)] &= \nabla_\theta \Big[\int_z p(z) f_\theta(z) dz \Big] \\ &= \int_z p(z) \Big[\nabla_\theta f_\theta(z) \Big] dz \\ &= \mathbb{E}_{p(z)} \Big[\nabla_\theta f_\theta(z) \Big]\end{aligned}$$

$$\begin{aligned}\nabla_\theta \mathbb{E}_{p_\theta(z)}[f_\theta(z)] &= \nabla_\theta \Big[\int_z p_\theta(z) f_\theta(z) dz \Big] \\ &= \int_z \nabla_\theta \Big[p_\theta(z) f_\theta(z) \Big] dz \\ &= \int_z f_\theta(z) \nabla_\theta p_\theta(z) dz + \int_z p_\theta(z) \nabla_\theta f_\theta(z) dz \\ &= \underbrace{\int_z f_\theta(z) \nabla_\theta p_\theta(z) dz}_{\text{What about this?}} + \mathbb{E}_{p_\theta(z)} \Big[\nabla_\theta f_\theta(z) \Big]\end{aligned}$$

[EXTRA] Reparameterization Trick

So, Reparameterization Trick!

$$
\begin{aligned}
\nabla_\theta \mathbb{E}_{p_\theta(z)}[f_\theta(z)] &= \nabla_\theta \Big[\int_z p_\theta(z) f_\theta(z) dz \Big] \\
&= \int_z \nabla_\theta \Big[p_\theta(z) f_\theta(z) \Big] dz \\
&= \int_z f_\theta(z) \nabla_\theta p_\theta(z) dz + \int_z p_\theta(z) \nabla_\theta f_\theta(z) dz \\
&= \underbrace{\int_z f_\theta(z) \nabla_\theta p_\theta(z) dz}_{\text{What about this?}} + \mathbb{E}_{p_\theta(z)} \Big[\nabla_\theta f_\theta(z) \Big]
\end{aligned}
$$

$$\Rightarrow$$

$$\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$$

$$\mathbf{z} = g_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}, \mathbf{x})$$

$$\mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{z})}[f(\mathbf{z}^{(i)})] = \mathbb{E}_{p(\boldsymbol{\epsilon})}[f(g_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}, \mathbf{x}^{(i)}))]$$

$$
\begin{aligned}
\nabla_\theta \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{z})}[f(\mathbf{z}^{(i)})] &= \nabla_\theta \mathbb{E}_{p(\boldsymbol{\epsilon})}[f(g_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}, \mathbf{x}^{(i)}))] && (1) \\
&= \mathbb{E}_{p(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}} f(g_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}, \mathbf{x}^{(i)}))] && (2) \\
&\approx \frac{1}{L} \sum_{l=1}^{L} \nabla_{\boldsymbol{\theta}} f(g_{\boldsymbol{\theta}}(\epsilon^{(l)}, \mathbf{x}^{(i)})) && (3)
\end{aligned}
$$

Training

Recall $p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; D_\theta(\boldsymbol{z}), \sigma^2\mathbf{I})$.

$$\log p_\theta\left(\boldsymbol{x}|\boldsymbol{z}^{(i)}\right) = \log\left(\frac{1}{\sqrt{(2\pi\sigma^2)^d}}\exp\left(-\frac{\|\boldsymbol{x} - D_\theta(\boldsymbol{z})\|^2}{2\sigma^2}\right)\right)$$

$$= -\frac{1}{2\sigma^2}\boxed{\|\boldsymbol{x} - D_\theta(\boldsymbol{z})\|^2} - \cancel{\log\sqrt{(2\pi\sigma^2)^d}}$$

This is why it is called
the reconstruction term.

Constant

[EXTRA] Why Gaussian in p(x|z)?

Starting with *q(z|x), p(z), p(x|z), …* everything is gaussian…
What does **p(x|z)**'s gaussian stands for?

$$p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x};\ D_\theta(\boldsymbol{z}), \sigma^2 I)$$

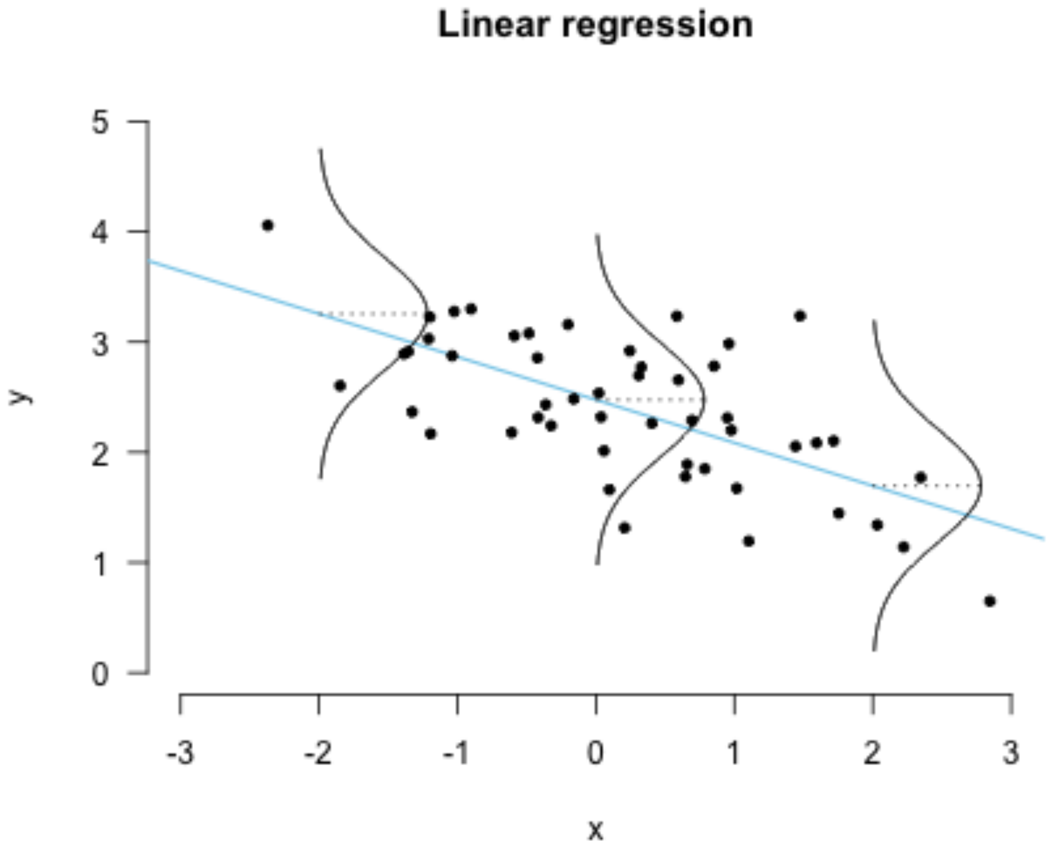
First, let's think about **Linear Regression**

Real continuous data has noise, and
we assume predicting this as Gaussian

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m + \varepsilon$$

So, the pdf is as follows,

$$f(y \mid \hat{y}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\hat{y})^2}{2\sigma^2}}$$



[EXTRA] Gaussian's MLE = MSE

Then the Loss will be looking like…

$$\mathcal{L} = \prod_{i=1}^{n} f(y_i \mid \hat{y}_i, \sigma^2) = (2\pi\sigma^2)^{-n/2} e^{-\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{2\sigma^2}}$$

And same as we done, it is same as **MSE**!

Let's come back to VAE,

Because we are making the generative model,

It has to have randomness, predicting the probability,

assuming the gaussian of p(x|z)

[EXTRA] And actually… We can also use BCE

Because dataset like MNIST has value of [0~1],

$$\mathrm{BCE}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

In the perspective with interpreting distribution as Bernoulli

We can use BCE Loss! (For MSE, we assumed continuous distribution)

But it will be good for datasets having close pixels to 0 and 1

(SAME in AutoEncoder)

[EXTRA] How about Prior matching Term?

$p(x) = \mathcal{N}(x; \mu, \sigma^2 I) \qquad q(x) = \mathcal{N}(x; 0, I)$

$$p(x) = \frac{1}{(2\pi)^{k/2}|\Sigma_p|^{1/2}} e^{-\frac{1}{2}(x-\mu)^\top \Sigma_p^{-1}(x-\mu)}$$

$$D_{\mathrm{KL}}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx.$$

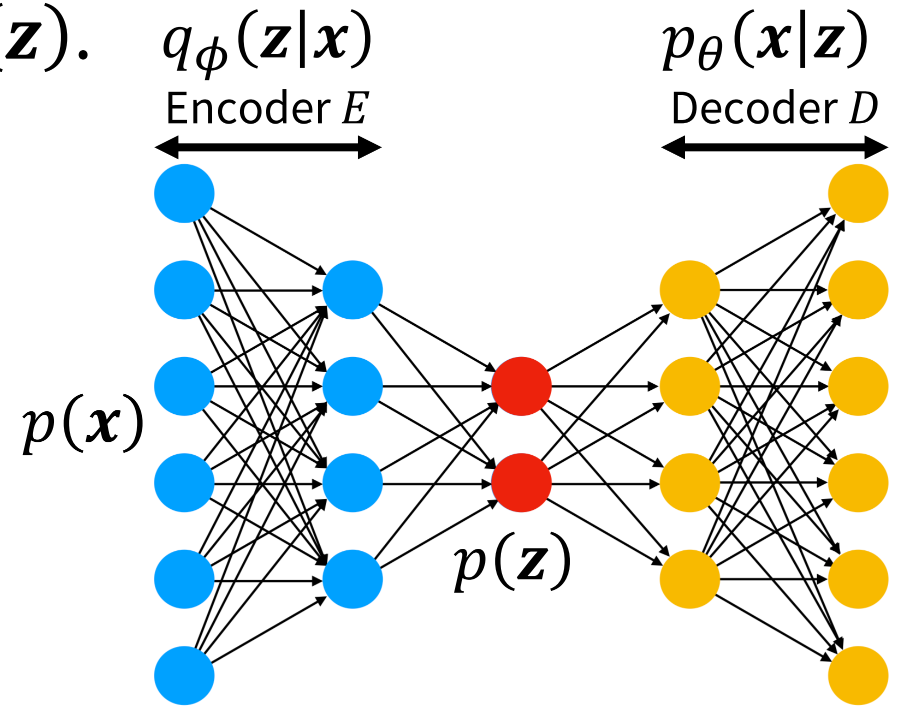
$$D_{\mathrm{KL}}(p\|q) = \frac{1}{2}\left(k(\sigma^2 - 1 - \log \sigma^2) + \|\mu\|^2\right)$$

```
def loss_function(recon_x, x, mu, logvar):
    MSE = F.mse_loss(recon_x, x, reduction='sum')
    KLD = -0.5 * torch.sum(1+logvar-mu.pow(2)-logvar.exp())
    return MSE+KLD
```

```
BCE = F.binary_cross_entropy(recon_x, x, reduction='sum')
```

Training

1. Feed a data point $\boldsymbol{x}$ to the encoder to predict $\boldsymbol{\mu}_\phi(\boldsymbol{x})$ and $\sigma_\phi^2(\boldsymbol{x})$.
2. Sample a latent variable $\boldsymbol{z}$ from $q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}_\phi(\boldsymbol{x}), \sigma_\phi^2(\boldsymbol{x})\mathbf{I}\right)$.
3. Feed $\boldsymbol{z}$ to the decoder to predict $\hat{\boldsymbol{x}} = D_\theta(\boldsymbol{z})$.
4. Compute the gradient decent through the negative ELBO.

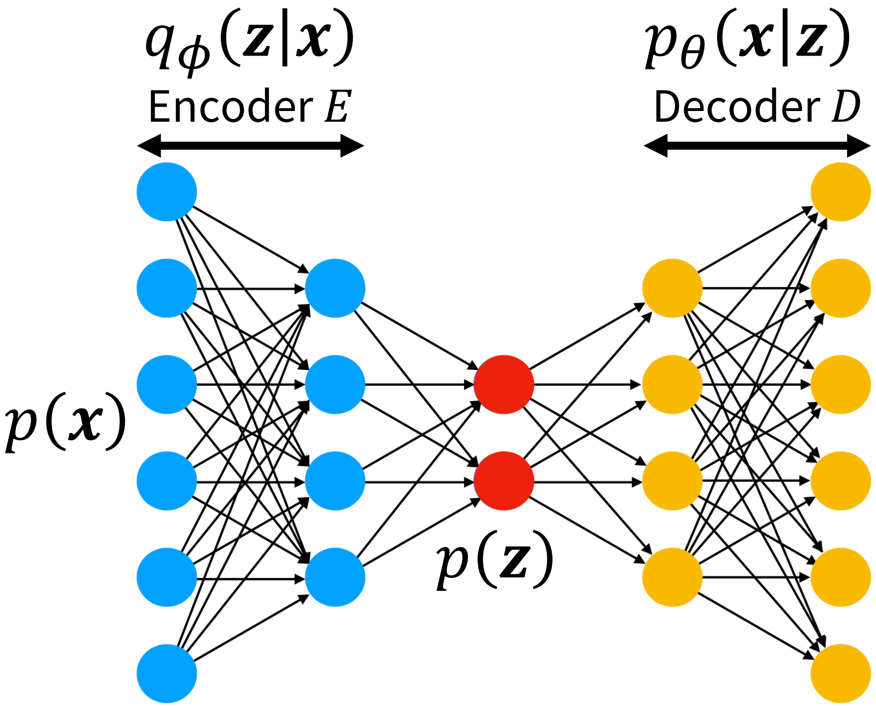


Q. Why is the sampling differentiable?

Generation

1. Sample a latent variable $\boldsymbol{z}$ from $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \mathbf{0}, \mathbf{I})$.

2. Feed $\boldsymbol{z}$ to the decoder to predict $\widehat{\boldsymbol{x}} = D_\theta(\boldsymbol{z})$.



Dimensions

Q. Should the dimensions of the input data and the latent variables be the same?

[EXTRA] VAE Overall Understanding

We are setting p(z) to be standard gaussian distribution.

Reconstruction Loss makes generated x to be similar to data x,

Prior matching term makes q(z|x) to be similar to p(z)

- Then, what **meanings** would **latent vectors** be trained for?

- q(z|x), p(z), p(x|z) has gaussian for its distribution, what does each stands for?

- Is the model or training **enough** for high quality generation?

  -> p(z)? p(x|z)? -> OK! But the problem is simple gaussian of q(z|x)

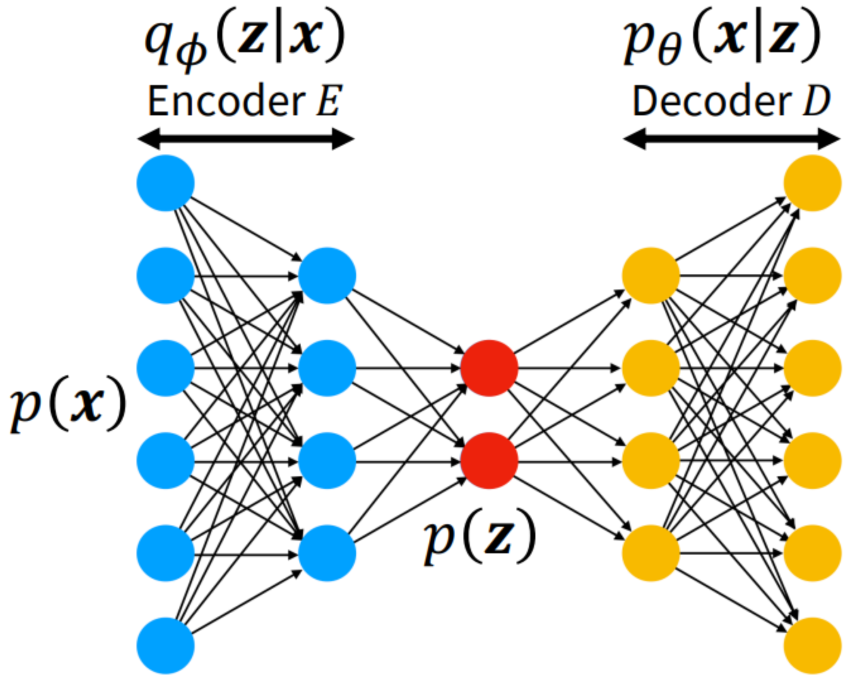
  (Let's see this later)

[EXTRA] VAE Overall Understanding

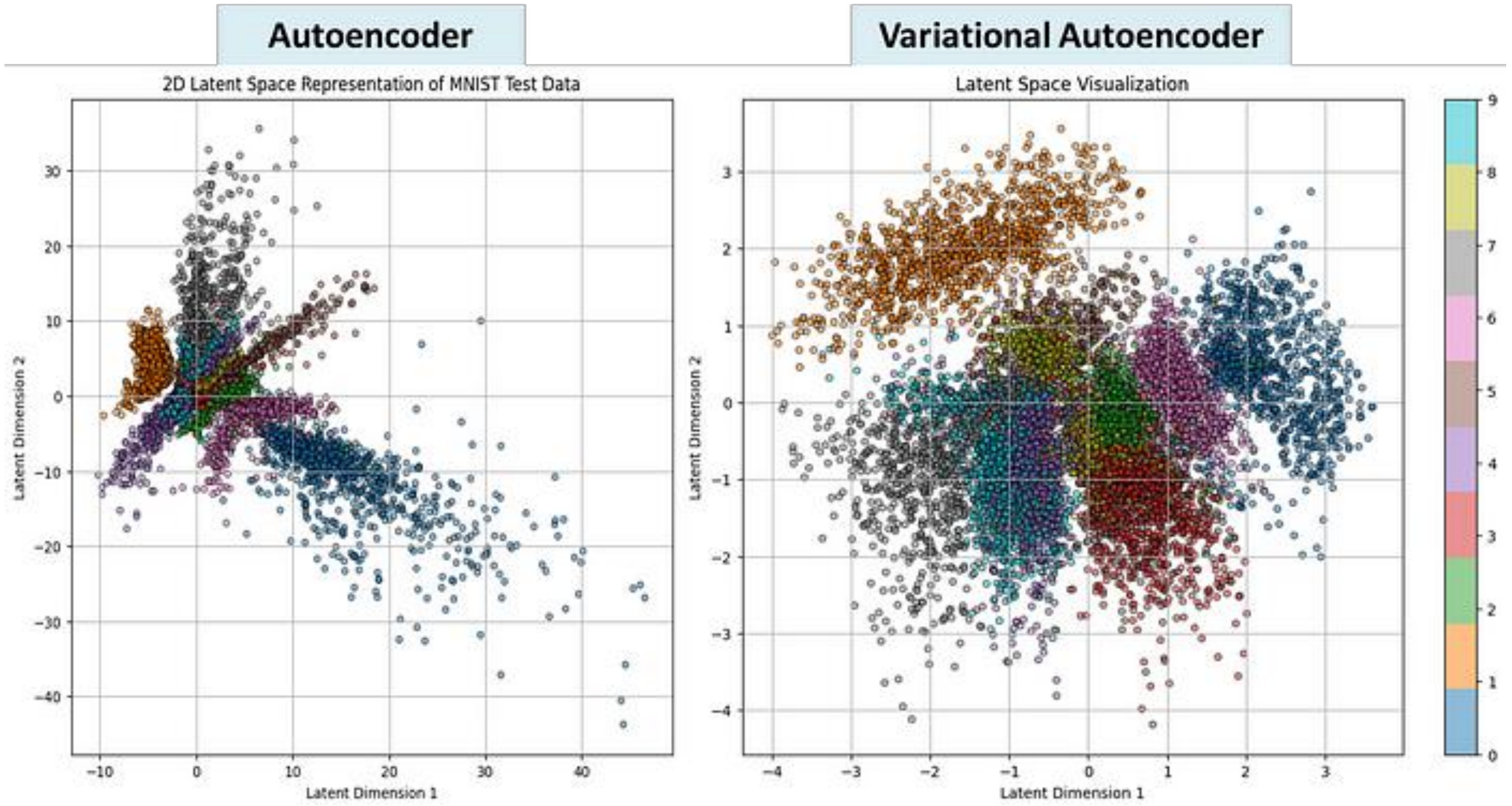
- Then, what **meanings** would **latent vectors** be trained for?

- q(z|x), p(z), p(x|z) has gaussian for its distribution, what does each stands for?

- Is the model or training **enough** for high quality generation?



[EXTRA] The difference between AE?



[EXTRA] VAE Overall Understanding

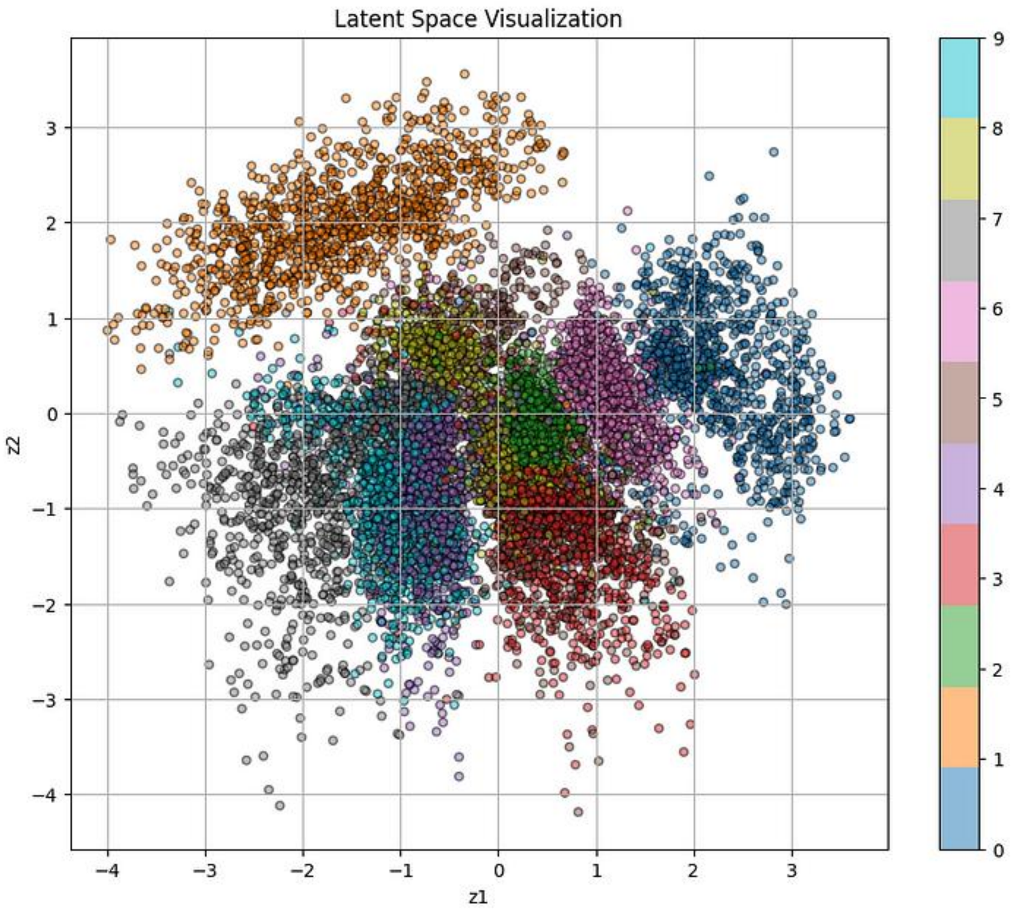
As we set p(z), q(z|x) to be gaussian distribution,

It seems to follow its distribution

But it seems to have **mixed** distribution

In **different** numbers (MNIST dataset)

With **2D** latent dimension



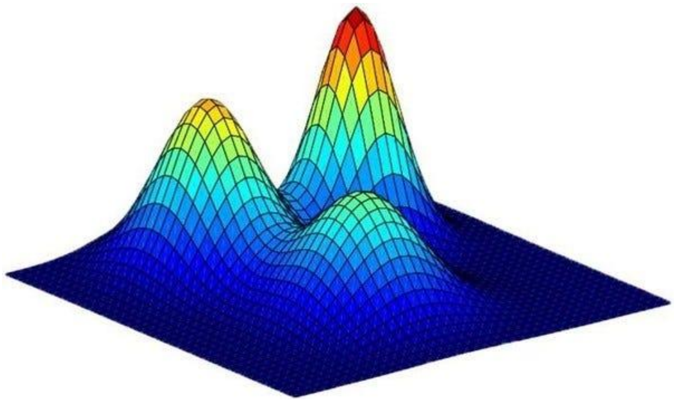
[EXTRA] VAE with multimodal distribution

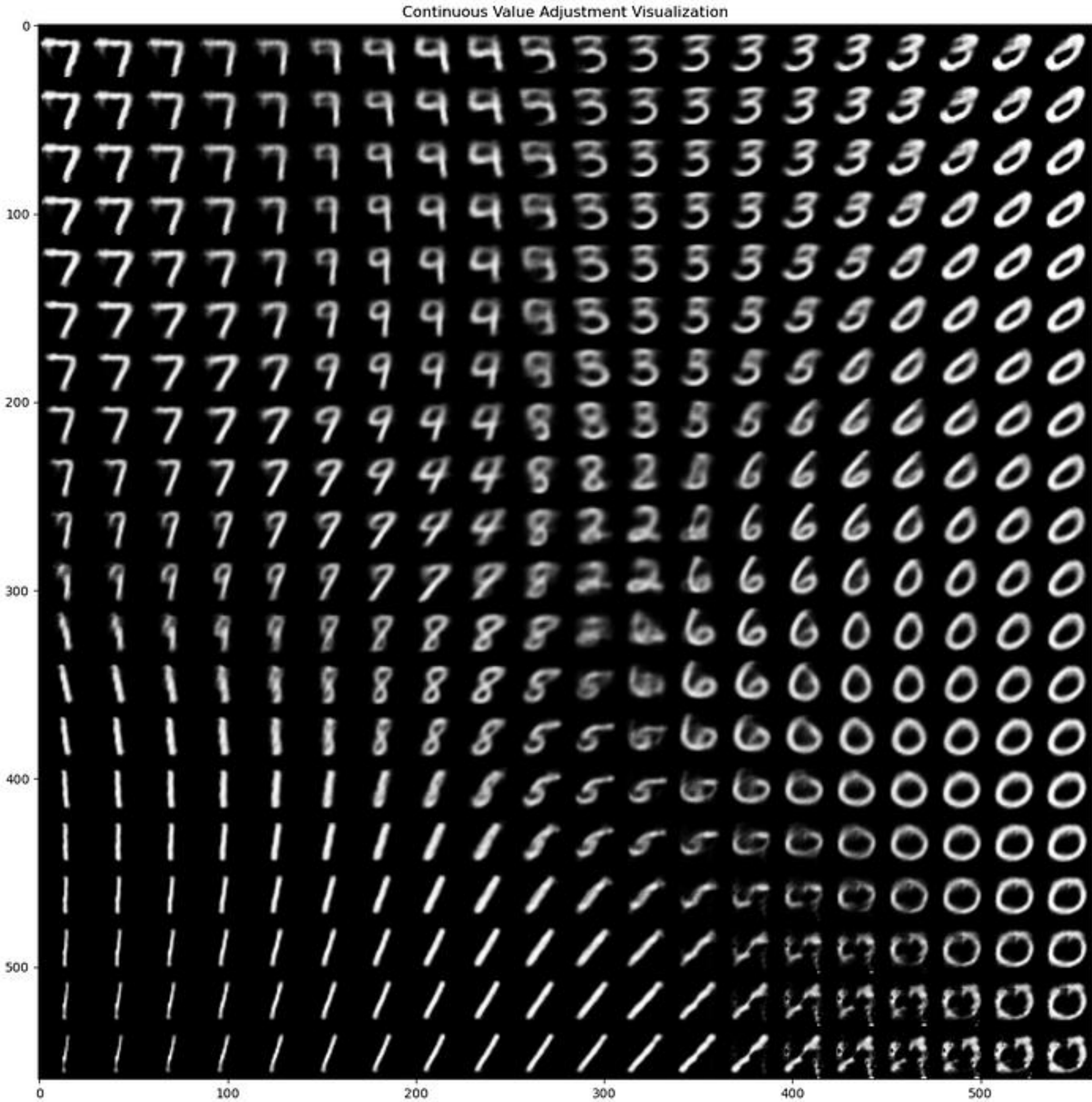
Of course, 2D is not enough!

But still, if we want to have different p(z),

Is it appropriate to learn with the same loss function?

How about MoG(Mixture of Gaussians?)

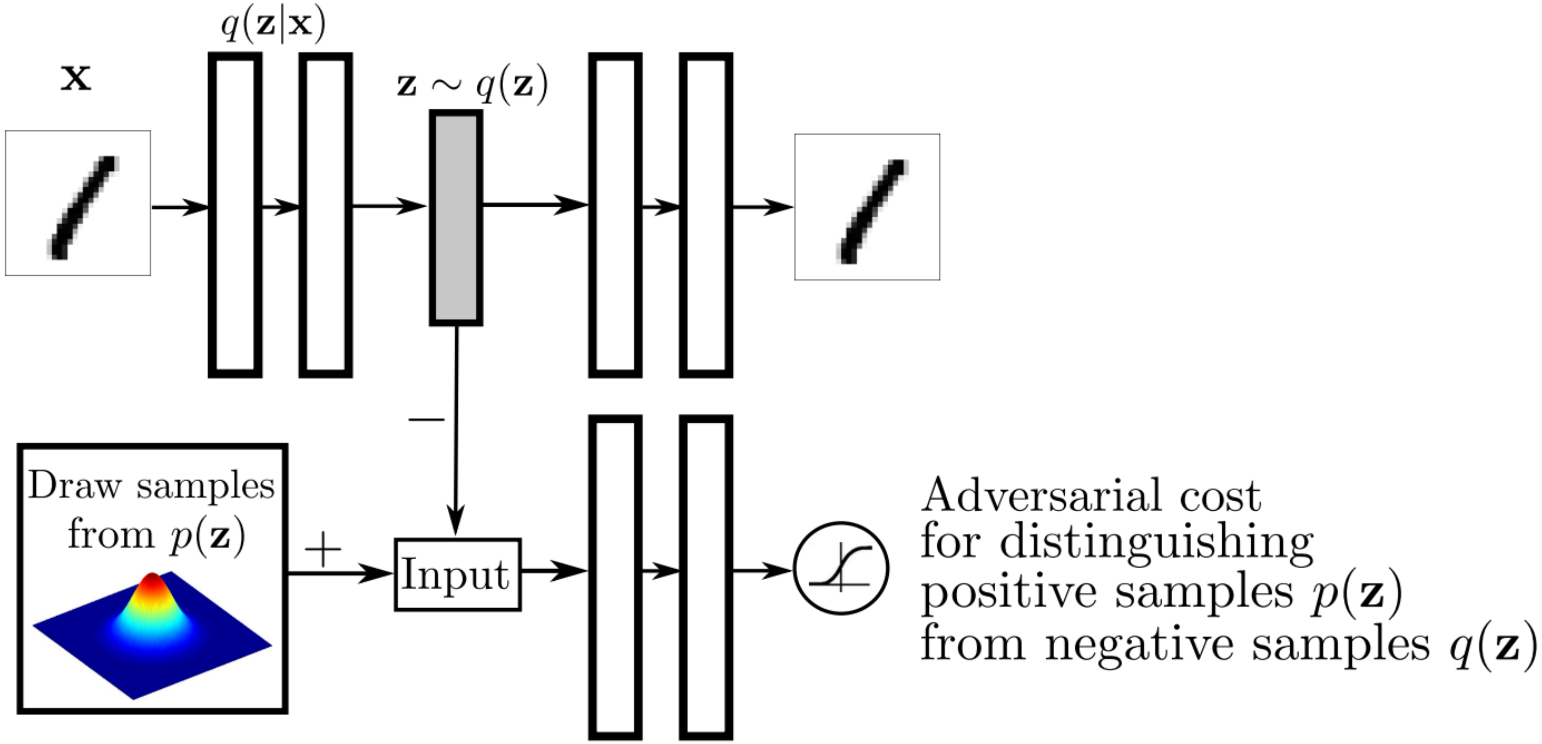




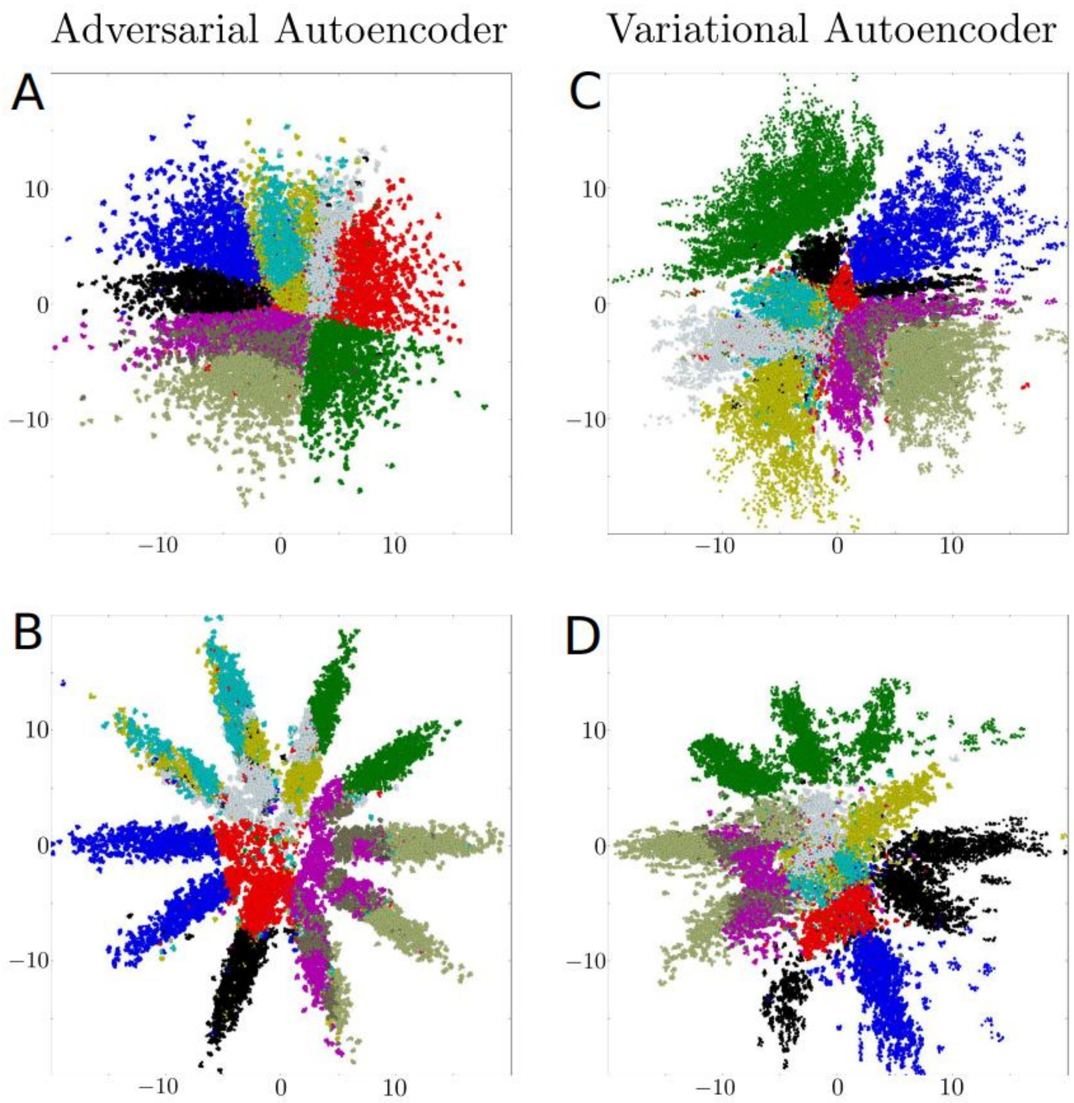
[EXTRA] Adversarial AutoEncoder

Because of Prior Matching Loss and q(z|x)'s single gaussian, making it difficult to follow the q(z)

One of the solutions… AAE (Adversarial AutoEncoder)



[EXTRA] Adversarial AutoEncoder

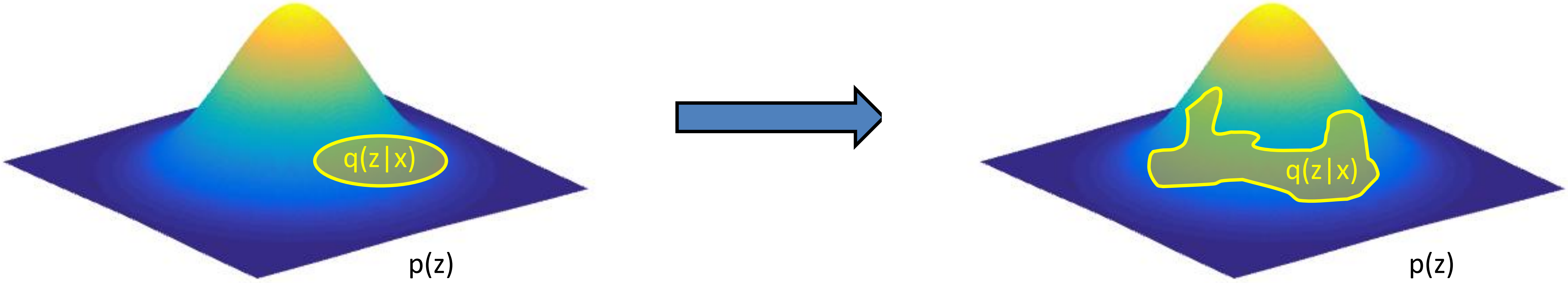


[EXTRA] Limitation of q(z|x) in VAE

p(z) being a standard gaussian distribution is not a limitation.

But the problem is simple gaussian of q(z|x)

We want to be more complex to make the model ideal!



[EXTRA] We can also decompose all from p(x)

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

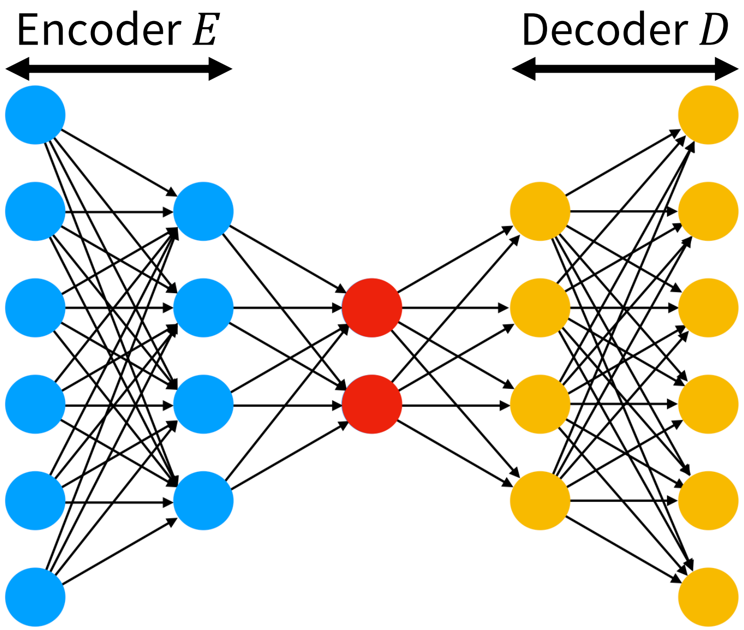
Limitations of VAEs

Typical failure cases of VAEs

Limitations of VAEs

Is a Gaussian distribution sufficient as the variational approximation for the posterior distribution?

$$q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}_\phi(\boldsymbol{x}), \sigma_\phi^2(\boldsymbol{x})\mathbf{I}\right)$$



Limitations of VAE

We maximize *not* $\log p(\boldsymbol{x})$ but its lower bound (ELBO).

Q. What is the difference between the two?

$$\left(\log p(\boldsymbol{x}) - \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right]\right)$$

[This was the homework from the last class.]

Limitations of VAE

A. $\log p(\boldsymbol{x}) - \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right] = D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}|\boldsymbol{x})\right).$

- The lower bound becomes tight when the variational distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is identical to the true posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$.

- Will the true posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$ be close to a Gaussian distribution…?

Limitations of VAE

What is a better method for approximating the posterior distribution in a variational way?

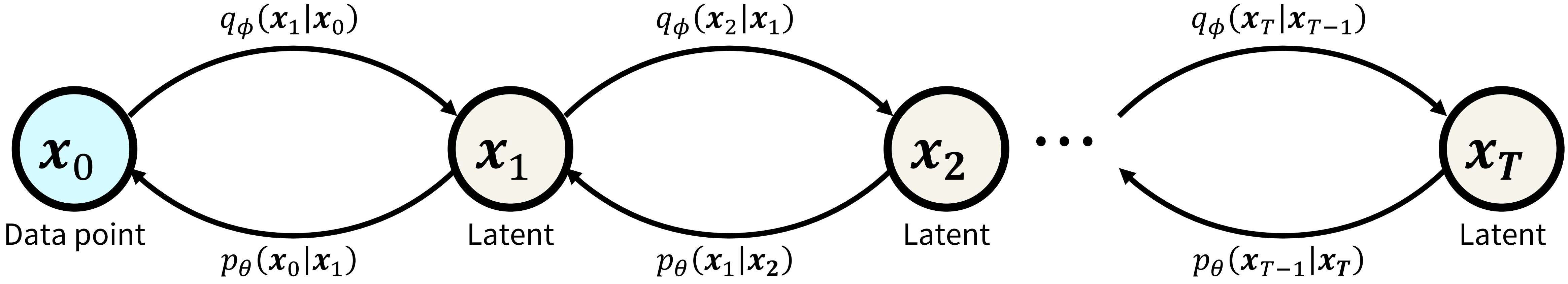
VAE Variants

- Vector-Quantized VAE (VQ-VAE) [Oord et al., NeurIPS 2017]
- Beta-VAE [Higgins et al., ICLR 2017]
- Wasserstein VAE (WAE) [Tolstikhin et al., ICLR 2018]
- VAE-GAN [Larsen et al., ICML 2016]
- Normalizing Flow VAE [Rezende and Mohamed, ICML 2015]

Hierarchical VAEs

Make a recursive (hierarchical) VAE.

- Data point: $\boldsymbol{x} \rightarrow \boldsymbol{x}_0$
- Latent variable(s): $\boldsymbol{z} \rightarrow \boldsymbol{x}_{1:T}$



Markovian Hierarchical VAEs

Let's consider a Markovian process.

A Markov process is a stochastic process where the probability of each event depends only on the previous state.

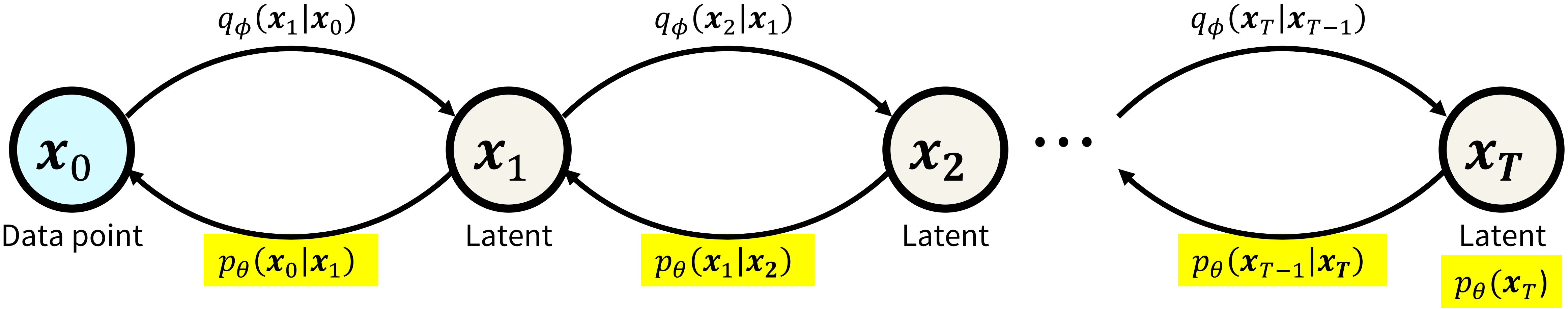
"What happens next depends only on the state of affairs now!"

Markovian Hierarchical VAEs

Let's consider a Markovian process.

Joint distribution:

$$p_\theta(\boldsymbol{x}_{0:T}) = p_\theta(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$$

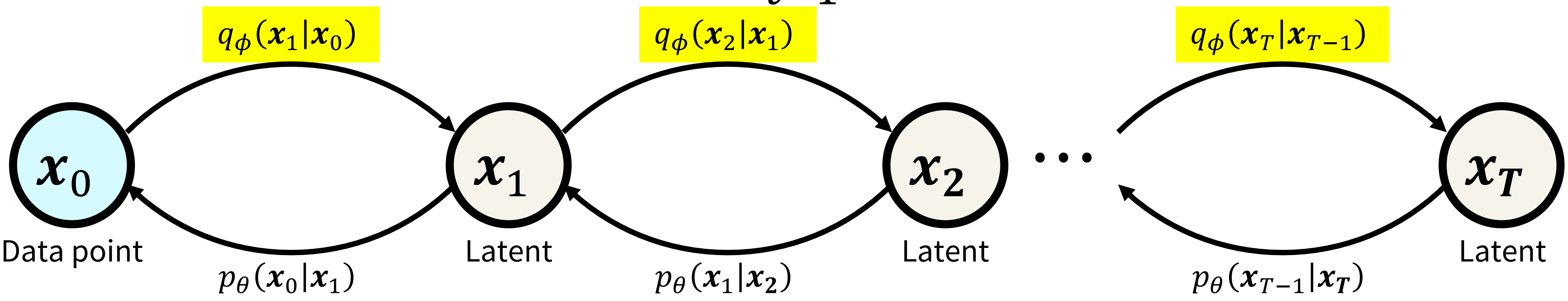


Markovian Hierarchical VAEs

Let's consider a Markovian process.

Variational posterior:

$$q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} q_\phi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$$



Markovian Hierarchical VAEs

$$
\begin{aligned}
\log p(\boldsymbol{x}_0) &= \log \int p_\theta(\boldsymbol{x}_{0:T}) d\boldsymbol{x}_{1:T} \\
&= \log \int p_\theta(\boldsymbol{x}_{0:T}) \frac{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} d\boldsymbol{x}_{1:T} \\
&= \log \mathbb{E}_{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\frac{p_\theta(\boldsymbol{x}_{0:T})}{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \right] \\
&\geq \mathbb{E}_{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \right]
\end{aligned}
$$

VAEs → Diffusion Models

Flow-Based Models

- Normalizing flow
- Nonlinear Independent Components Estimation (NICE)
- Real Non-Volume Preserving (Real NVP)
- Generative Flow (Glow)
- Masked autoregressive flow (MAF)
- Continuous Normalizing Flow (CNF)

We'll revisit this later!

Denoising Diffusion Probabilistic Models (DDPM)

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

Denoising Diffusion Probabilistic Models

Consider a special case of the Markovian hierarchical VAEs where:

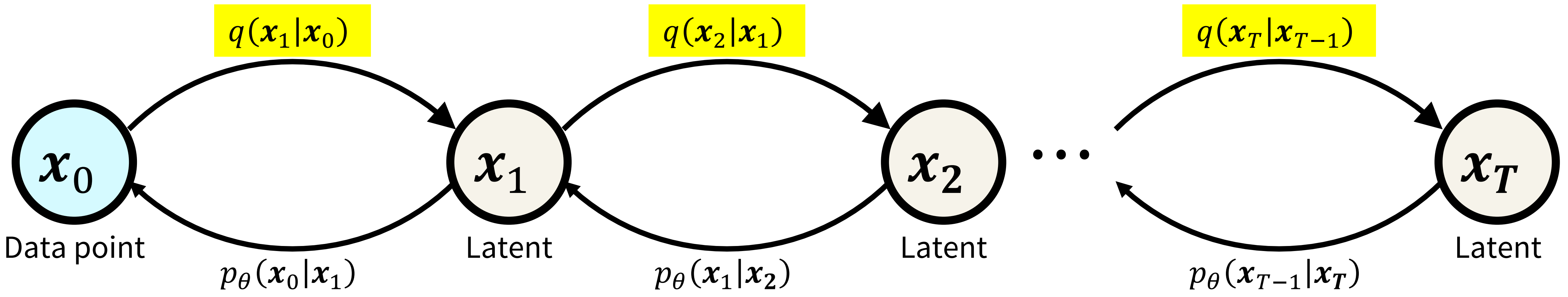
- the latent dimension is the same as the data dimension, and
- the variational posteriors $q_\phi(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)$ are not learned but predefined:

$$q_\phi(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) \to q(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)$$

Terminology

Forward process (predefined):

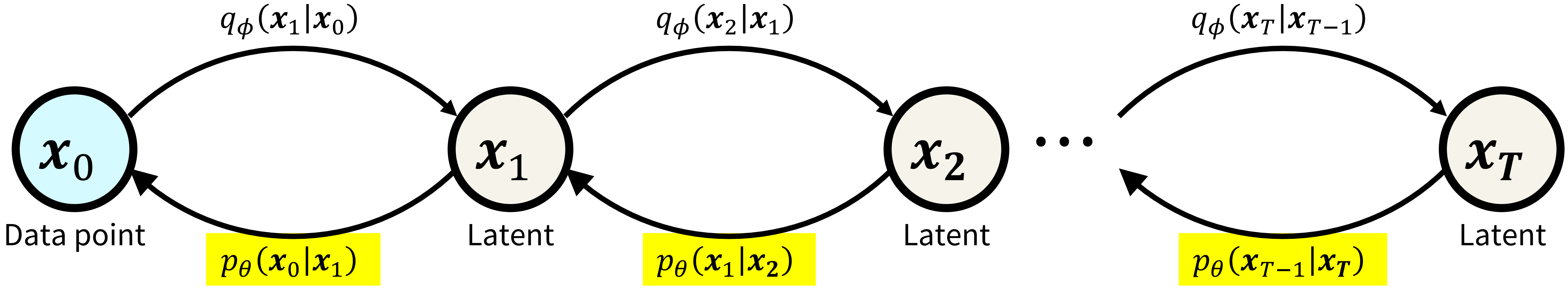
$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$



Terminology

Reverse process (learned):

$$p_\theta(\boldsymbol{x}_0, \boldsymbol{x}_{1:T}) = p_\theta(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1} | \boldsymbol{x}_t)$$



Forward Process (Data → Latent)

In the forward process, the transition distribution $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is specifically predefined as:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{1-\boxed{\beta_t}}\boldsymbol{x}_{t-1}, \boxed{\beta_t}\mathrm{I}\right)$$

Time-dependent parameter

where $\{\beta_t \in (0,1)\}_{t=1}^{T}$ and $\beta_1 \leq \beta_2 \leq \cdots \leq \beta_T$.

"Adding Gaussian noise iteratively!"

VP-SDE vs. VE-SDE

- $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \mathbf{I}\right)$,

  is called Variance Preserving (VP) form.

- There are other options. For example:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}\left(\boldsymbol{x}_t; \boldsymbol{x}_{t-1}, \left(\sigma_i^2 - \sigma_{i-1}^2\right)\mathbf{I}\right),$$

  which is called Variance Exploding form.

Choice of β_t

- Learned.
- Constant.
- Linearly or quadratically increased.
- Follows a cosine function
  (Nichol and Dhariwal, Improved Denoising Diffusion Probabilistic Models, ICML 2021).
- Note that the reverse step $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ becomes a Gaussian form only when β_t is small ($\beta_t \ll 1$).

How to maximize ELBO in this case?

Disclaimer: We'll skip some complicated equations in the following slides.

Sohl-Dickstein et al., Deep Unsupervised Learning using Nonequilibrium Thermodynamics, ICML 2015.
Lilian Weng , What are Diffusion Models?.
Calvin Luo, Understanding Diffusion Models: A Unified Perspective.

ELBO

How to minimize the *negative* ELBO in this case?

$$-\log p(\boldsymbol{x}_0) = -\log \int p_\theta(\boldsymbol{x}_{0:T}) d\boldsymbol{x}_{1:T} \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[-\log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right]$$

$$= \cdots =$$

$$-\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)]$$

Reconstruction term

$$+\mathbb{E}_{q(\boldsymbol{x}_{T-1}|\boldsymbol{x}_0)}\left[D_{KL}\left(q(\boldsymbol{x}_T|\boldsymbol{x}_{T-1}) \parallel p(\boldsymbol{x}_T)\right)\right]$$

Prior matching term

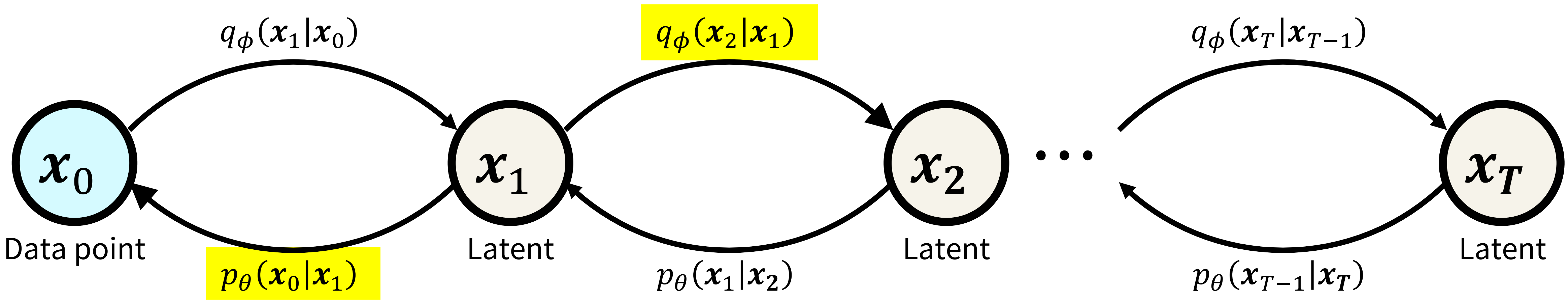
$$+\sum_{t=1}^{T-1} \mathbb{E}_{q(\boldsymbol{x}_{t-1},\boldsymbol{x}_{t+1}|\boldsymbol{x}_0)}\left[D_{KL}\left(q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \parallel p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})\right)\right]$$

Consistency term

Consistency Term

$$\boxed{\sum_{t=1}^{T-1} \mathbb{E}_{q(\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t+1} | \boldsymbol{x}_0)} \left[D_{KL}\left(q(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}) \parallel p_\theta(\boldsymbol{x}_t | \boldsymbol{x}_{t+1}) \right) \right]}$$

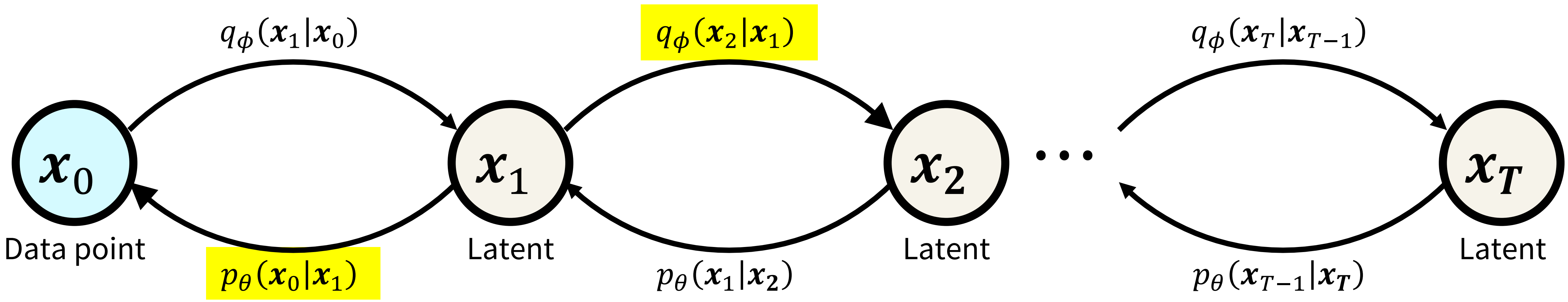
Make the forward and reverse steps be consistent at each time step.



Consistency Term

$$\sum_{t=1}^{T-1} \mathbb{E}_{q(\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t+1} | \boldsymbol{x}_0)} \left[D_{KL}\left(q(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}) \parallel p_\theta(\boldsymbol{x}_t | \boldsymbol{x}_{t+1}) \right) \right]$$

Expectation over two random variables; computationally expensive.



ELBO

Can we avoid having two random variables in an expectation?

Let's re-decompose the ELBO using the fact that

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0).$$

Q. Why $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0)$?

ELBO

Decompose the negative ELBO in a different way :

$$-\log p(\boldsymbol{x}_0) = -\log \int p_\theta(\boldsymbol{x}_{0:T}) d\boldsymbol{x}_{1:T} \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[-\log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right]$$

$$= \cdots =$$

$$-\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)]$$

Reconstruction term $\mathcal{L}_0$

$$+D_{KL}\big(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T)\big)$$

New prior matching term $\mathcal{L}_T$

$$+\sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\big[D_{KL}\big(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\big)\big]$$

Denoising matching term $\mathcal{L}_{t-1}$

Reconstruction Term $\mathcal{L}_0$

$$\boxed{-\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)]}$$

The same loss term as in VAE, but applied only to the final reverse step.

ELBO

Decompose the negative ELBO in a different way :

$$-\log p(\boldsymbol{x}_0) = -\log \int p_\theta(\boldsymbol{x}_{0:T}) d\boldsymbol{x}_{1:T} \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[-\log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right]$$

$$= \cdots =$$

$$-\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)]$$

$$\boxed{+D_{KL}\big(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T)\big)}$$

New prior matching term $\mathcal{L}_T$

$$+\sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{KL}\big(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\big)\right]$$

Prior Matching Term $\mathcal{L}_T$

Prior Matching Term $\mathcal{L}_T$

$$\boxed{D_{KL}\big(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T)\big)}$$

- Identical to the KL divergence term in VAE.
- Note that there is nothing to be optimized; $q(\boldsymbol{x}_T|\boldsymbol{x}_0)$ are $p(\boldsymbol{x}_T)$ are predefined.

Forward Convergence

Then, $q(\mathbf{x}_T|\mathbf{x}_0) = \mathcal{N}(\boldsymbol{x}; \mathbf{0}, \mathbf{I})$?

Yes, under certain assumptions.

Recall

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\mathbf{I}\right)$$

where $\{\beta_t \in (0,1)\}_{t=1}^{T}$ and $\beta_1 < \beta_2 < \cdots < \beta_T$.

Forward Convergence

Then, $q(\mathbf{x}_T|\mathbf{x}_0) = \mathcal{N}(\boldsymbol{x}; \mathbf{0}, \mathbf{I})$?

Yes, under certain assumptions.

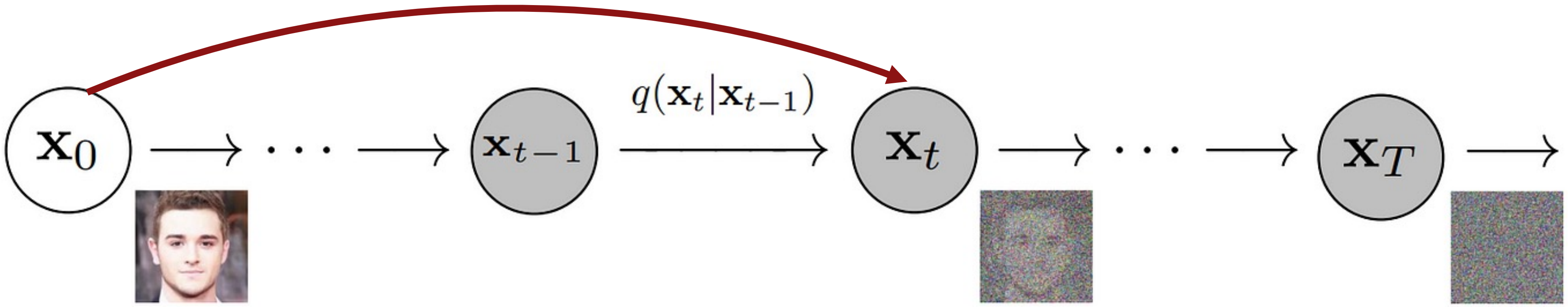
Let $\alpha_t = 1 - \beta_t$.

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$

where $\{\alpha_t \in (0,1)\}_{t=1}^T$ and $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_T$.

$q(\boldsymbol{x}_t|\boldsymbol{x}_0)$

Can we derive $q(\mathbf{x}_t|\mathbf{x}_0)$ from the sequence of $q(\boldsymbol{x}_{t'}|\boldsymbol{x}_{t'-1})$ for $t = 1, \dots, t'$?



Basics: Combination of Gaussian Variables

Suppose $\boldsymbol{x}_1 \sim \mathcal{N}(\mu_1, \sigma_1^2 \mathbf{I})$ and $\boldsymbol{x}_2 \sim \mathcal{N}(\mu_2, \sigma_2^2 \mathbf{I})$.

Q. What is the distribution of $\boldsymbol{x}_1 + \boldsymbol{x}_2$?

Basics: Combination of Gaussian Variables

Suppose $\boldsymbol{x}_1 \sim \mathcal{N}(\mu_1, \sigma_1^2 \mathbf{I})$ and $\boldsymbol{x}_2 \sim \mathcal{N}(\mu_2, \sigma_2^2 \mathbf{I})$.

A. $\boldsymbol{x}_1 + \boldsymbol{x}_2 \sim \mathcal{N}(\mu_1 + \mu_2, (\sigma_1^2 + \sigma_2^2)\mathbf{I})$

Basics: Combination of Gaussian Variables

Suppose $\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and

$\boldsymbol{x}_1 = \sigma_1 \boldsymbol{\varepsilon}_1$ and $\boldsymbol{x}_2 = \sigma_2 \boldsymbol{\varepsilon}_2$.

Q. What is the distribution of $\boldsymbol{x}_1 + \boldsymbol{x}_2$?

Basics: Combination of Gaussian Variables

Suppose $\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and

$\boldsymbol{x}_1 = \sigma_1 \boldsymbol{\varepsilon}_1$ and $\boldsymbol{x}_2 = \sigma_2 \boldsymbol{\varepsilon}_2$.

A. $\boldsymbol{x}_1 + \boldsymbol{x}_2 \sim \mathcal{N}(\mathbf{0}, (\sigma_1^2 + \sigma_2^2)\mathbf{I})$.

$\boldsymbol{x}_1 + \boldsymbol{x}_2 = \sqrt{\sigma_1^2 + \sigma_2^2}\,\boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon}$ is another standard normal sample.

Forward Convergence

$$q(\boldsymbol{x}_1|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_1; \sqrt{\alpha_1}\boldsymbol{x}_0, (1-\alpha_1)\mathbf{I})$$

$$q(\boldsymbol{x}_2|\boldsymbol{x}_1) = \mathcal{N}(\boldsymbol{x}_2; \sqrt{\alpha_2}\boldsymbol{x}_1, (1-\alpha_2)\mathbf{I})$$

Q. What is the distribution of $q(\boldsymbol{x}_2|\boldsymbol{x}_0)$?

Hint. Let's use the reparamaterization trick:

$$\boldsymbol{x}_1 = \sqrt{\alpha_1}\boldsymbol{x}_0 + \sqrt{1-\alpha_1}\boldsymbol{\epsilon}_0$$

$$\boldsymbol{x}_2 = \sqrt{\alpha_2}\boldsymbol{x}_1 + \sqrt{1-\alpha_2}\boldsymbol{\epsilon}_1$$

$$\boxed{\boldsymbol{\epsilon}_0, \boldsymbol{\epsilon}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})}$$

Forward Convergence

A. $\boldsymbol{x}_2 = \sqrt{\alpha_2}\boxed{\boldsymbol{x}_1} + \sqrt{1-\alpha_2}\boldsymbol{\epsilon}_1$

$$= \sqrt{\alpha_2}\boxed{\left(\sqrt{\alpha_1}\boldsymbol{x}_0 + \sqrt{1-\alpha_1}\boldsymbol{\epsilon}_0\right)} + \sqrt{1-\alpha_2}\boldsymbol{\epsilon}_1$$

$$= \sqrt{\alpha_2\alpha_1}\boldsymbol{x}_0 + \boxed{\sqrt{\alpha_2(1-\alpha_1)}\boldsymbol{\epsilon}_0 + \sqrt{1-\alpha_2}\boldsymbol{\epsilon}_1}$$

$$= \sqrt{\alpha_2\alpha_1}\boldsymbol{x}_0 + \boxed{\sqrt{(1-\alpha_2\alpha_1)}\bar{\boldsymbol{\epsilon}}_0}$$

$$\therefore q(\boldsymbol{x}_2|\boldsymbol{x}_0) = \mathcal{N}(\sqrt{\alpha_2\alpha_1}\boldsymbol{x}_1, (1-\alpha_2\alpha_1)\mathbf{I})$$

Forward Convergence

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}$$

$$= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}\right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}$$

$$= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{\alpha_t(1-\alpha_{t-1})}\boldsymbol{\epsilon}_{t-2} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}$$

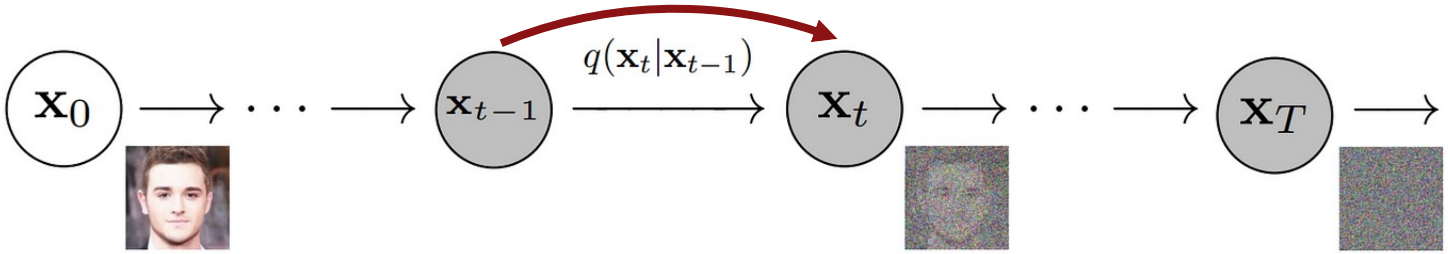
$$= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{(1-\alpha_t\alpha_{t-1})}\bar{\boldsymbol{\epsilon}}_{t-2}$$

$$= \cdots$$

$$= \sqrt{\prod_{i=1}^{t}\alpha_i}\,\boldsymbol{x}_0 + \sqrt{\left(1-\prod_{i=1}^{t}\alpha_i\right)}\bar{\boldsymbol{\epsilon}}_0$$

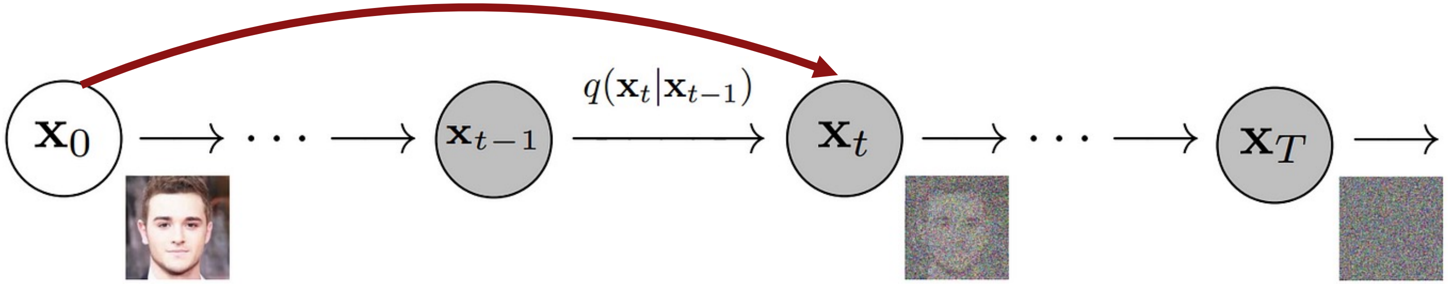
Forward Convergence

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$



$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ *Also a normal distribution!*

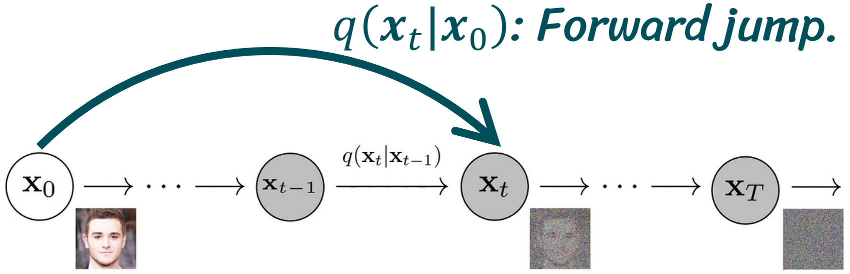


$q(\boldsymbol{x}_t|\boldsymbol{x}_0)$

Given $\boldsymbol{x}_0$, $\boldsymbol{x}_t$ at any arbitrary timestep t can be directly sampled from a Gaussian distribution without a Markov chain:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right).$$

Note that $\bar{\alpha}_1 > \bar{\alpha}_2 > \cdots > \bar{\alpha}_T$.



Forward Convergence

$$\boxed{q(\boldsymbol{x}_T|\boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_T; \sqrt{\bar{\alpha}_T}\boldsymbol{x}_0, (1-\bar{\alpha}_T)\mathbf{I}\right)}$$

where $\bar{\alpha}_T = \prod_{t=1}^{T}(1-\beta_t)$.

Q. When $\{\beta_t \in (0,1)\}_{t=1}^{T}$, What is

$$\lim_{T\to\infty} \bar{\alpha}_T = \lim_{T\to\infty} \prod_{t=1}^{T}(1-\beta_t)\,?$$

Forward Convergence

$$\boxed{q(\boldsymbol{x}_T|\boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_T; \cancelto{0}{\sqrt{\bar{\alpha}_T}}\boldsymbol{x}_0, (1 - \cancelto{0}{\bar{\alpha}_T})\mathbf{I}\right)}$$

where $\bar{\alpha}_T = \prod_{t=1}^{T}(1 - \beta_t)$.

As $T \to \infty$, $q(\boldsymbol{x}_T|\boldsymbol{x}_0)$ converges to the standard normal distribution $\mathcal{N}(\boldsymbol{x}; \mathbf{0}, \mathbf{I})$.

Prior Matching Term $\mathcal{L}_T$

$$\boxed{D_{KL}\big(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T)\big)}$$

Close to zero by the definition of the forward transition distribution $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$. *Nothing to do for the optimization.*

ELBO

Decompose the negative ELBO in a different way :

$$-\log p(\boldsymbol{x}_0) = -\log \int p_\theta(\boldsymbol{x}_{0:T}) d\boldsymbol{x}_{1:T} \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[-\log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right]$$

$$= \cdots =$$

$$-\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)]$$

$$\cancel{+D_{KL}\left(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T)\right)} \rightarrow 0$$

$$\boxed{+\sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{KL}\left(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)\right]}$$

Denoising matching term $\mathcal{L}_{t-1}$

Denoising Matching Term $\mathcal{L}_{t-1}$

Denoising Matching Term $\mathcal{L}_{t-1}$

$$\boxed{\sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{KL}\left(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)\right]}$$

The variational distribution $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ should be close to $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ for each t.

Denoising Matching Term $\mathcal{L}_{t-1}$

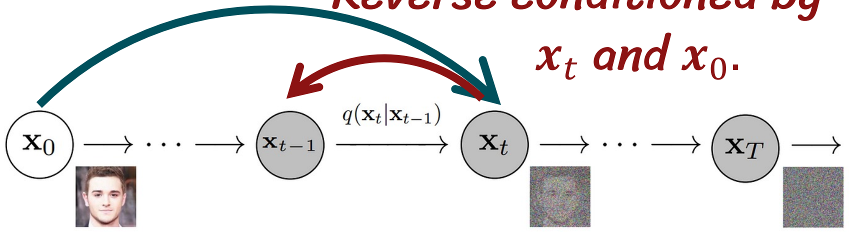
What is $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$?

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \boxed{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \mathbf{x}_0)}\frac{\boxed{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}}{\boxed{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}}$$

Same as $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, the forward transition, since it's a Markovian process.

We have seen how to compute these.

$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$: Reverse conditioned by $\boldsymbol{x}_t$ and $\boldsymbol{x}_0$.



$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}$$

Q. What are $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)$, and $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$?

$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}$$

- $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}\left(\sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I}\right)$
- $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I}\right)$
- $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$

Sohl-Dickstein et al., Deep Unsupervised Learning using Nonequilibrium Thermodynamics, ICML 2015.
Lilian Weng , What are Diffusion Models?.
Calvin Luo, Understanding Diffusion Models: A Unified Perspective.

$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})\frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}$$

$$\propto \exp\left(-\frac{1}{2}\left(\frac{(\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{x}_{t-1})}{1-\alpha_t} + \frac{(\boldsymbol{x}_t - \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0)}{1-\bar{\alpha}_{t-1}} - \frac{(\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0)}{1-\bar{\alpha}_t}\right)\right)$$

$$= \cdots$$

$$= \mathcal{N}(\tilde{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\sigma}_t^2\mathbf{I})$$

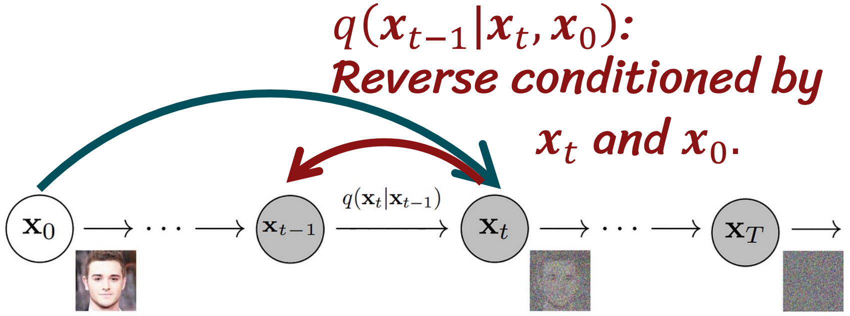
Another normal distribution!

where $\tilde{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0$ and $\tilde{\sigma}_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$.

$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$

- The mean $\tilde{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0$ is a function of both $\boldsymbol{x}_t$ and $\boldsymbol{x}_0$.

- The covariance $\tilde{\sigma}_t^2\mathbf{I} = \left(\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\right)\mathbf{I}$ is predefined from the user-defined $\{\beta_t\}_{t=1}^T$.



$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$

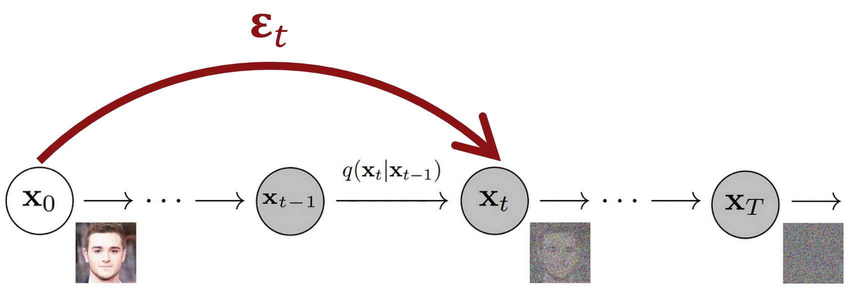
From the forward jump $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$,

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}_t \quad \text{where } \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

If $\boldsymbol{x}_t$ and $\boldsymbol{x}_0$ are given, define $\boldsymbol{\varepsilon}_t$ as

$$\boldsymbol{\varepsilon}_t = \frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{x}_t - \frac{\sqrt{\bar{\alpha}_t}}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{x}_0$$

Q. Rewrite $\tilde{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0)$ as a function of $\boldsymbol{x}_t$ and $\boldsymbol{\varepsilon}_t$.



$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$

A.

$$\tilde{\mu}(\mathbf{x}_t, \boldsymbol{\varepsilon}_t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0$$

$$= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\varepsilon}_t\right)$$

Denoising Matching Term $\mathcal{L}_{t-1}$

Back to the denoising matching term…

$$\boxed{\sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{KL}\left(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)\right]}$$

How to model the variational distribution $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$?

Denoising Matching Term $\mathcal{L}_{t-1}$

- For $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\tilde{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\sigma}_t^2\mathbf{I})$,

  the variance $\tilde{\sigma}_t^2$ is *not* a function of $\boldsymbol{x}_t$ and $\boldsymbol{x}_0$.

- Hence, define the variational distribution $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ as

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\mu_\theta(\boldsymbol{x}_t, t), \tilde{\sigma}_t^2\mathbf{I}),$$

  where $\mu_\theta(\boldsymbol{x}_t, t)$ is the mean predictor.

Denoising Matching Term

How to compute

$$\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{KL}\left(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)\right]?$$

Q. When $p(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x};\ \boldsymbol{\mu}_p, \sigma^2\mathbf{I}\right)$ and $q(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x};\ \boldsymbol{\mu}_q, \sigma^2\mathbf{I}\right)$,

What is $D_{\mathrm{KL}}(p \parallel q)$?

[A similar problem as the homework from the last class.]

Denoising Matching Term

A.

$$p(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x};\ \boldsymbol{\mu}_p, \sigma^2\mathbf{I}\right)$$

$$q(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x};\ \boldsymbol{\mu}_q, \sigma^2\mathbf{I}\right)$$

$$D_{\mathrm{KL}}(p \parallel q) = \frac{1}{2\sigma^2}\left\|\boldsymbol{\mu}_q - \boldsymbol{\mu}_p\right\|^2$$

Denoising Matching Term

How to compute

$$\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{KL}\left(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)\right]?$$

$$\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{KL}\left(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)\right]$$
$$= \frac{1}{2\tilde{\sigma}_t^2}\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[||\mu_\theta(\boldsymbol{x}_t, t) - \tilde{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0)||^2]$$

$\boldsymbol{x}_0$ Predictor

Q. What if we have a $\boldsymbol{x}_0$ predictor $\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t, t)$ instead of the mean predictor $\mu_\theta(\boldsymbol{x}_t, t)$? Note that

$$\tilde{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}\boldsymbol{x}_0$$

$\boldsymbol{x}_0$ Predictor

A. $\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{KL}\left(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)\right]$

$$= \frac{1}{2\tilde{\sigma}_t^2}\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[||\mu_\theta(\mathbf{x}_t,t) - \tilde{\mu}(\boldsymbol{x}_t,\boldsymbol{x}_0)||^2]$$

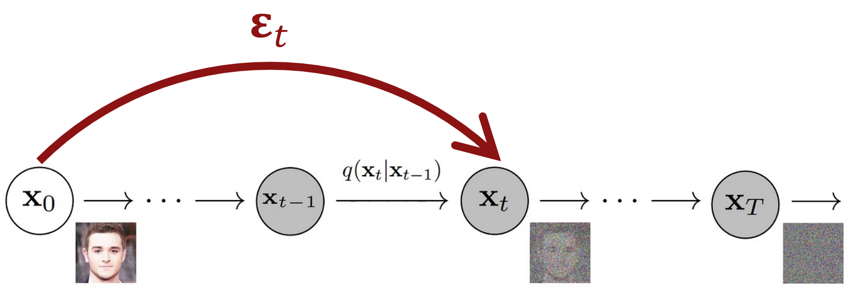
$$= \frac{1}{2\tilde{\sigma}_t^2}\frac{\bar{\alpha}_{t-1}\beta_t^2}{(1-\bar{\alpha}_t)^2}\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[||\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t,t) - \boldsymbol{x}_0||^2]$$

$$= \omega_t\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[||\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t,t) - \boldsymbol{x}_0||^2]$$

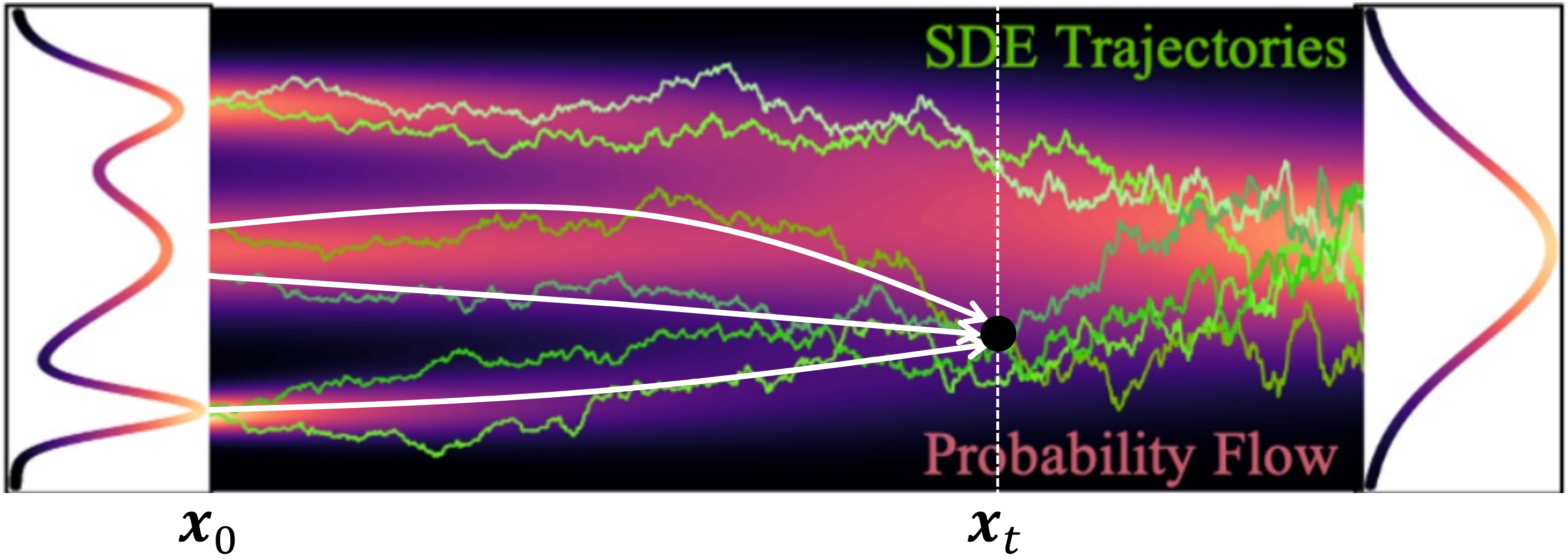
x_0 Predictor

$$\boxed{\omega_t \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[\|\hat{\boldsymbol{x}}_\theta(\mathbf{x}_t, t) - \boldsymbol{x}_0\|^2]}$$

- $\boldsymbol{x}_t$ is sampled from $\boldsymbol{x}_0$.
- From $\boldsymbol{x}_t$, predict the *expected* value of $\boldsymbol{x}_0$ that would result in sampling $\boldsymbol{x}_t$ from it through the forward jump.



x_0 Predictor



$\boldsymbol{x}_0$ Predictor

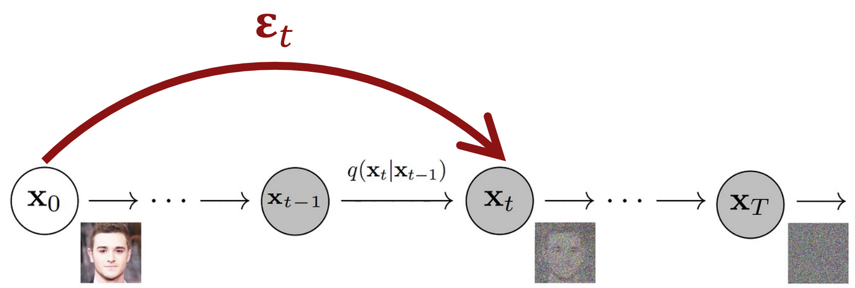
- Note that our goal is to sample $\boldsymbol{x}_0$ from a standard normal sample $\boldsymbol{x}_T$ and through latent variables $\boldsymbol{x}_{T-1}, \boldsymbol{x}_{T-2}, \cdots, \boldsymbol{x}_1$.

- But for every $\boldsymbol{x}_t$, we directly predict the expected value of $\boldsymbol{x}_0$ from $\boldsymbol{x}_t$.

$\boldsymbol{\varepsilon}_t$ Predictor

Q. What if we have a $\boldsymbol{\varepsilon}_t$ predictor $\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{x}_t, t)$ instead of the mean predictor $\mu_\theta(\boldsymbol{x}_t, t)$? Note that

$$\tilde{\mu}(\mathbf{x}_t, \boldsymbol{\varepsilon}_t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\varepsilon}_t\right)$$



$\boldsymbol{\varepsilon}_t$ Predictor

A. $\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{KL}\left(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)\right]$

$$= \frac{1}{2\tilde{\sigma}_t^2}\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[\|\mu_\theta(\mathbf{x}_t,t) - \tilde{\mu}(\boldsymbol{x}_t,\boldsymbol{x}_0)\|^2]$$

$$= \frac{1}{2\tilde{\sigma}_t^2}\frac{(1-\bar{\alpha}_t)^2}{\bar{\alpha}_t(1-\bar{\alpha}_t)}\mathbb{E}_{q_\phi(\boldsymbol{x}_t|\boldsymbol{x}_0)}[\|\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{x}_t,t) - \boldsymbol{\varepsilon}_t\|^2]$$

$$= \omega_t'\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[\|\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{x}_t,t) - \boldsymbol{\varepsilon}_t\|^2]$$

$\boldsymbol{\varepsilon}_t$ Predictor

$$\boxed{\omega'_t \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[\|\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{\varepsilon}_t\|^2]}$$

From $\boldsymbol{x}_t$, predict the *expected* value of $\boldsymbol{\varepsilon}_t$ that would result in sampling $\boldsymbol{x}_t$ from $\boldsymbol{x}_0$ through the forward jump.