CS 446/ECE 449: Machine Learning

Lecture 22: Diffusion Models

Han Zhao
04/09/2024

ILLINOIS
Computer Science
GRAINGER COLLEGE OF ENGINEERING

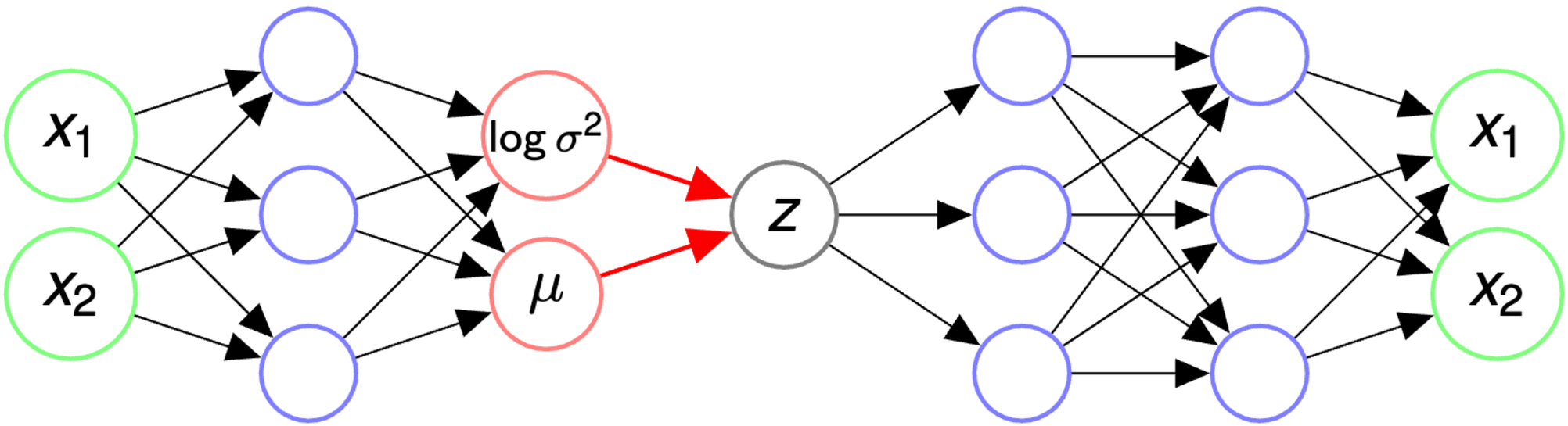
Reference for Diffusion Models

- “Deep Learning: Foundations and Concepts”, Chapter 20
- “Tutorial on Diffusion Models for Imaging and Vision”, Stanley H. Chan, https://arxiv.org/abs/2403.18103
- “Denoising Diffusion Probabilistic Models”, https://arxiv.org/abs/2006.11239
- “Variational Diffusion Models”, https://arxiv.org/abs/2107.00630

Photo credit: part of the images in this slides are from “Tutorial on Diffusion Models for Imaging and Vision”.

Recap: Variational Auto-encoder

Recall from Lecture 20 we introduced the variational auto-encoder (VAE):



$q_\phi(z \mid x)$ $p_\theta(x \mid z)$

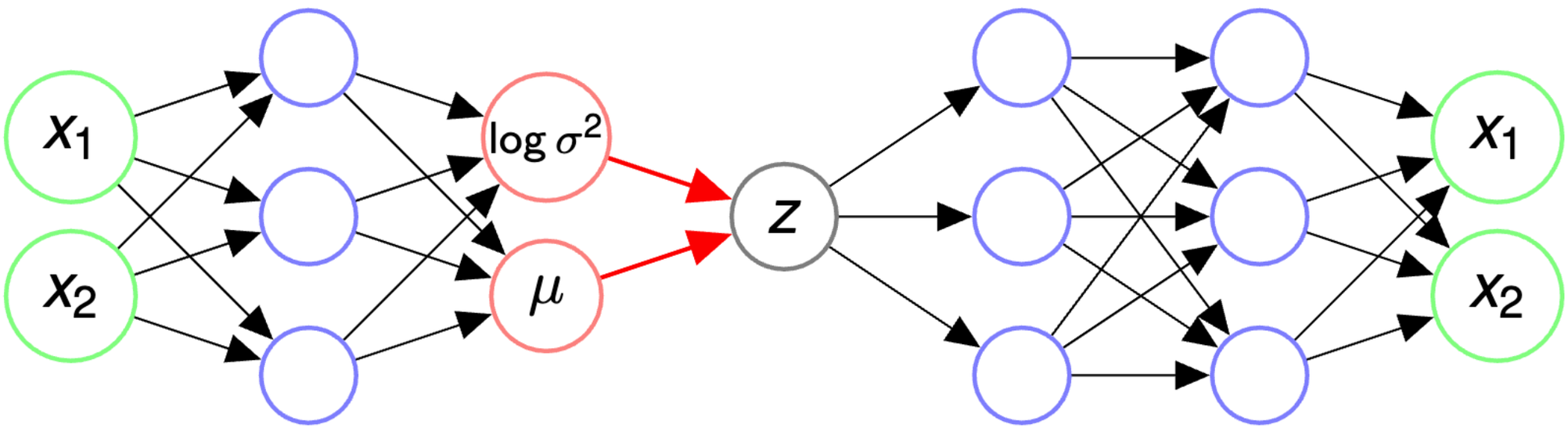
$$\mathcal{L}(p_\theta, q_\phi) \approx -D_{KL}(q_\phi, p) + \frac{1}{N}\sum_{i=1}^{N} \log p_\theta(x|z^i)$$

Notation:

- Encoder: q_ϕ : mapping each input x to a Gaussian distribution $\mathcal{N}(z \mid \mu_\phi(x), \sigma^2_\phi(x))$
- Decoder: p_θ: mapping each hidden code z to a Gaussian distribution $\mathcal{N}(x \mid \mu_\theta(z), \sigma^2 I)$

Recap: Variational Auto-encoder

Recall from Lecture 20 we introduced the variational auto-encoder (VAE):



Note: p is a pre-defined simple prior distribution over the hidden codes $\mathcal{Z}$, usually chosen to be $\mathcal{N}(0,I)$

Recap: Variational Auto-encoder

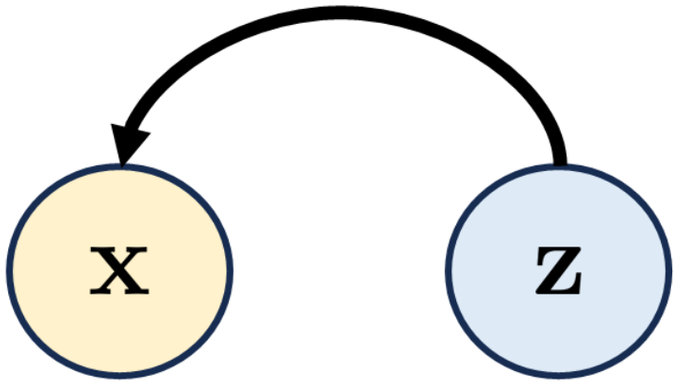
Recall from Lecture 20 we introduced the variational auto-encoder (VAE):

$$\mathcal{L}(p_\theta, q_\phi) \approx -D_{KL}(q_\phi, p) + \frac{1}{N}\sum_{i=1}^{N} \log p_\theta(x|z^i)$$

Prior Matching

Reconstruction Loss

(2) We find $\boldsymbol{\theta}$ to maximize $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$

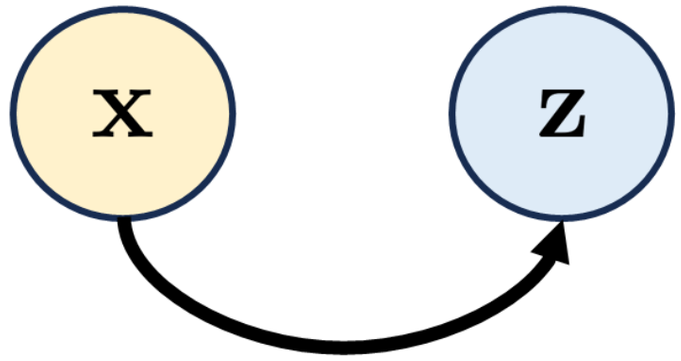


$\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$

(1) During training, you tell us $(\mathbf{z}, \mathbf{x})$

Reconstruction

(1) During training, you tell us $(\mathbf{z}, \mathbf{x})$



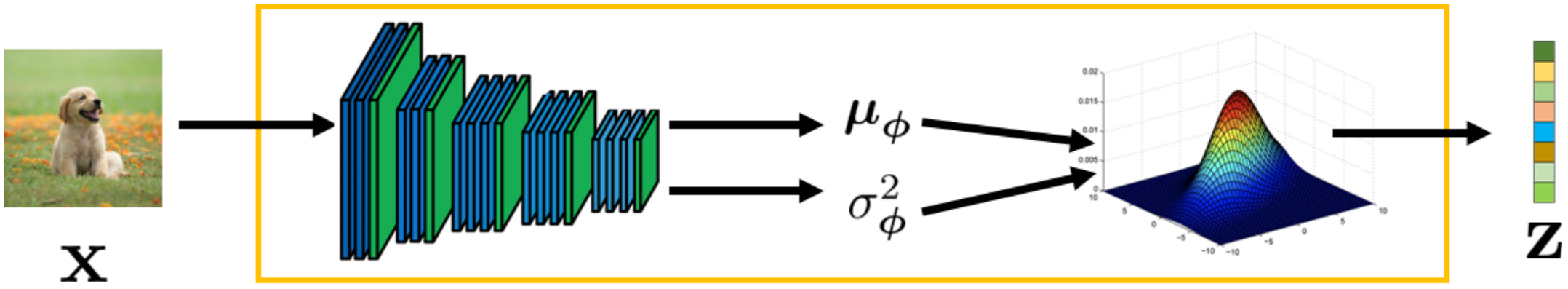
(2) We find $\boldsymbol{\phi}$ to minimize $\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))$

Prior Matching

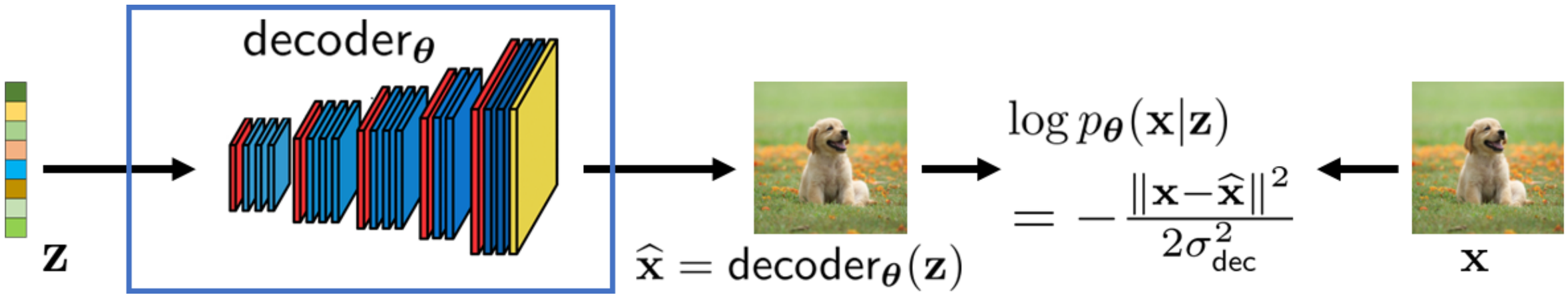
Recap: Variational Auto-encoder

Training stage of VAE:

Encoder:



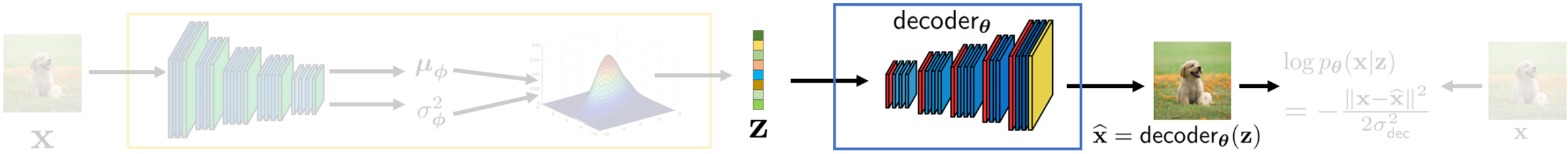
Decoder:



Recap: Variational Auto-encoder

Inference stage of VAE:

Encoder gets dropped, only using the decoder

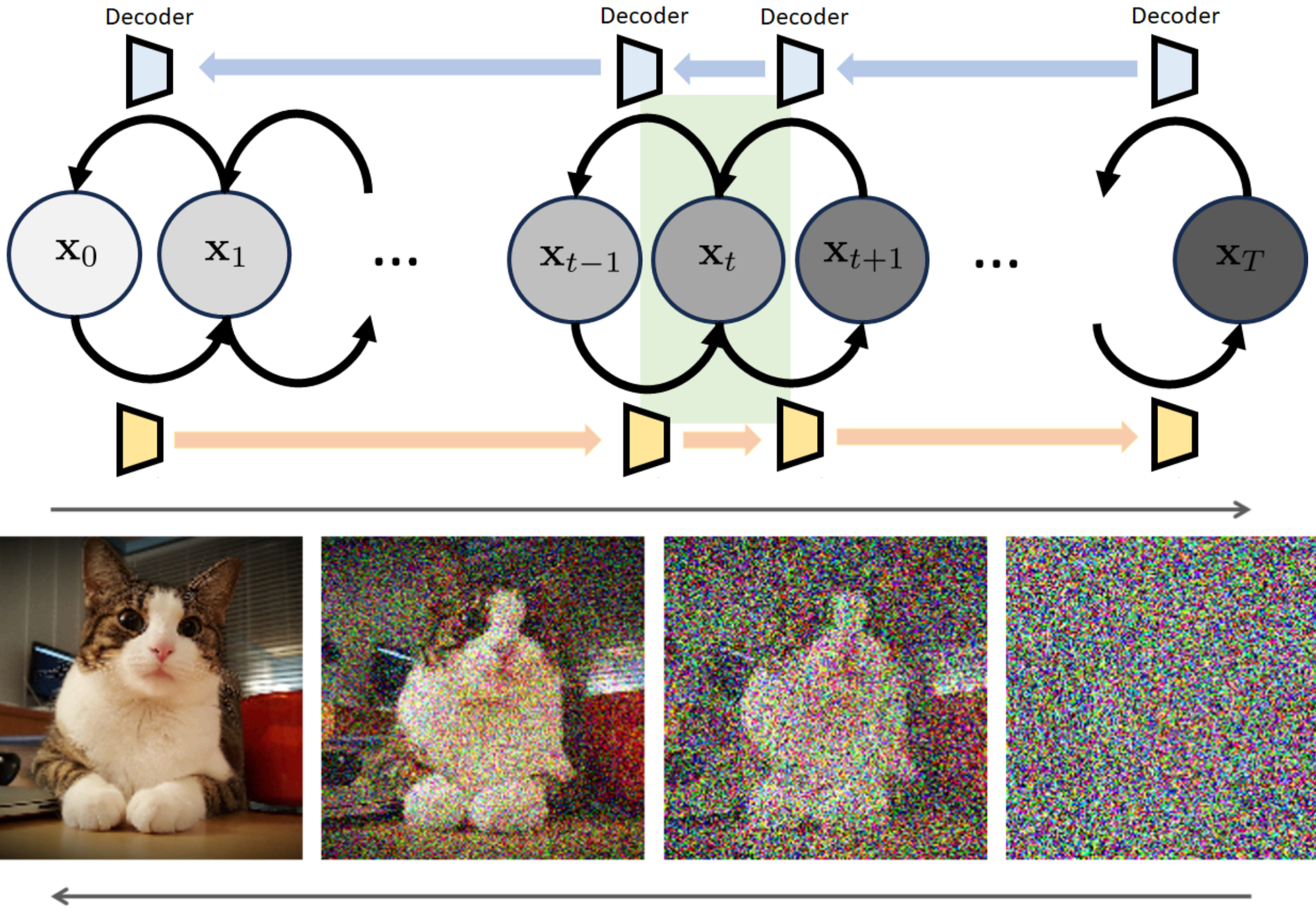


Like sampling from a GAN:

- First sample $z \sim p = \mathcal{N}(0,I)$
- Pass the sampled z through the decoder to obtain $\hat{x} = \text{Mean}(p_\theta(z))$

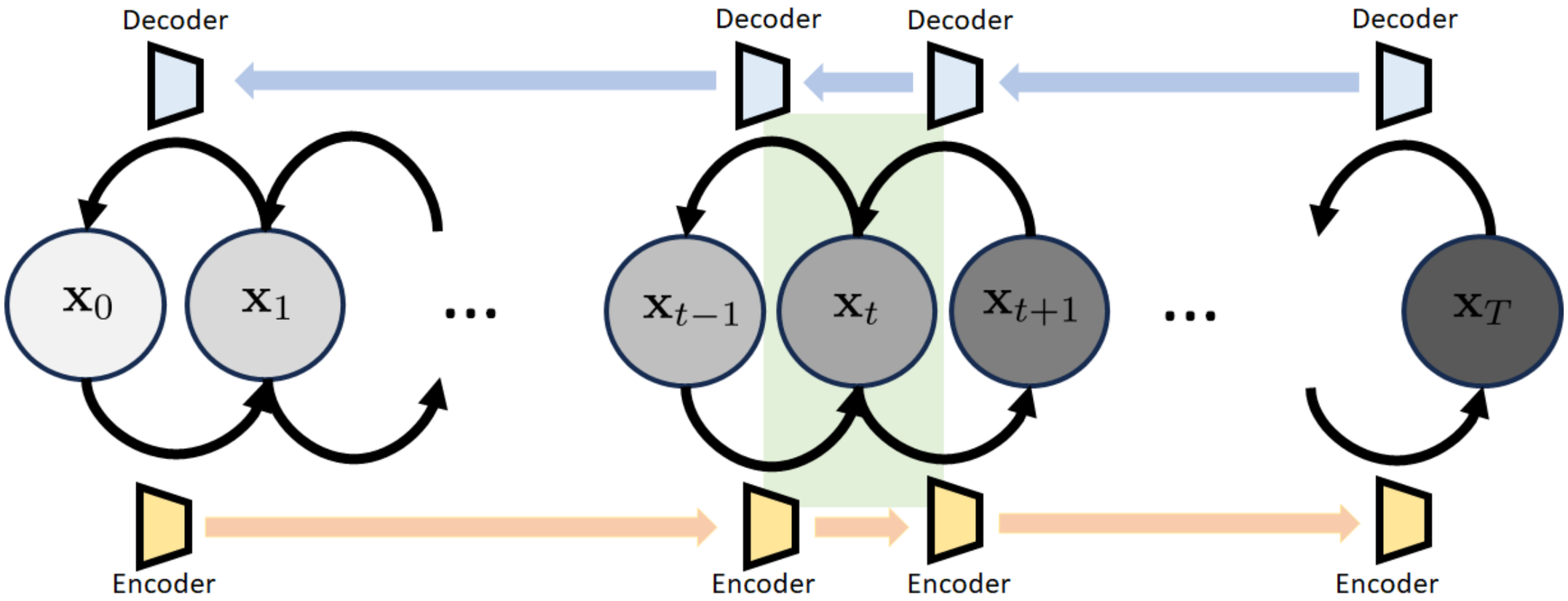
Diffusion Models

Roughly speaking, diffusion models are iterative VAEs



Diffusion Models

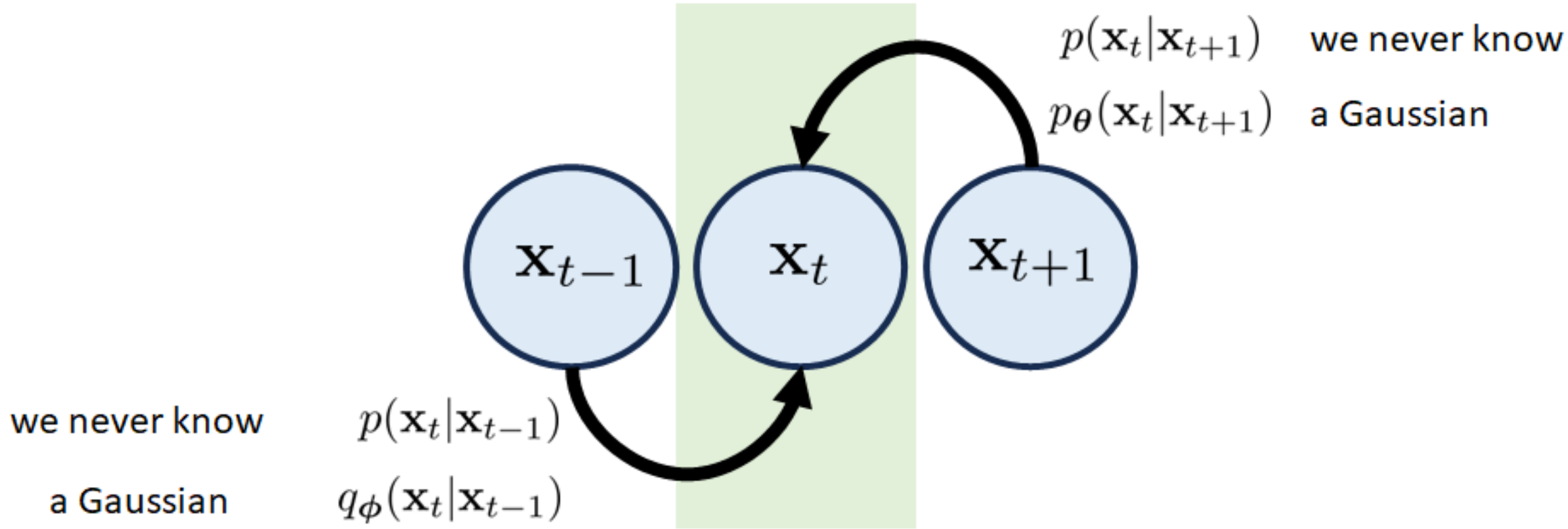
More formally,



- We have a sequence of states, $x_0, \ldots, x_T$
- There is an unknown underlying joint distribution $p(x_0, \ldots, x_T)$ over the sequence of states
- The initial state $x_0 = x$ is the observable input image
- The end state $x_T = z \sim \mathcal{N}(0,I)$ is supposed to be some hidden code sampled from a simple prior distribution, e.g., high-dim standard Gaussian

Diffusion Models

Basic building block:

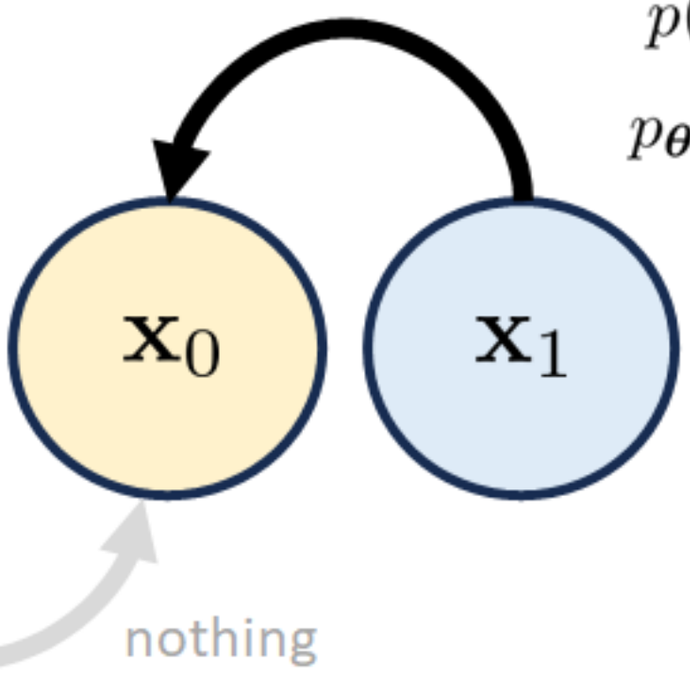


Similar to what we have in VAE:

- We use q_ϕ as the encoder to approximate the underlying true conditional distribution $p(x_t \mid x_{t-1})$
- We use p_θ as the decoder to approximate the underlying true conditional distribution $p(x_t \mid x_{t+1})$

Diffusion Models

Initial block:



$p(\mathbf{x}_0|\mathbf{x}_1)$ we never know

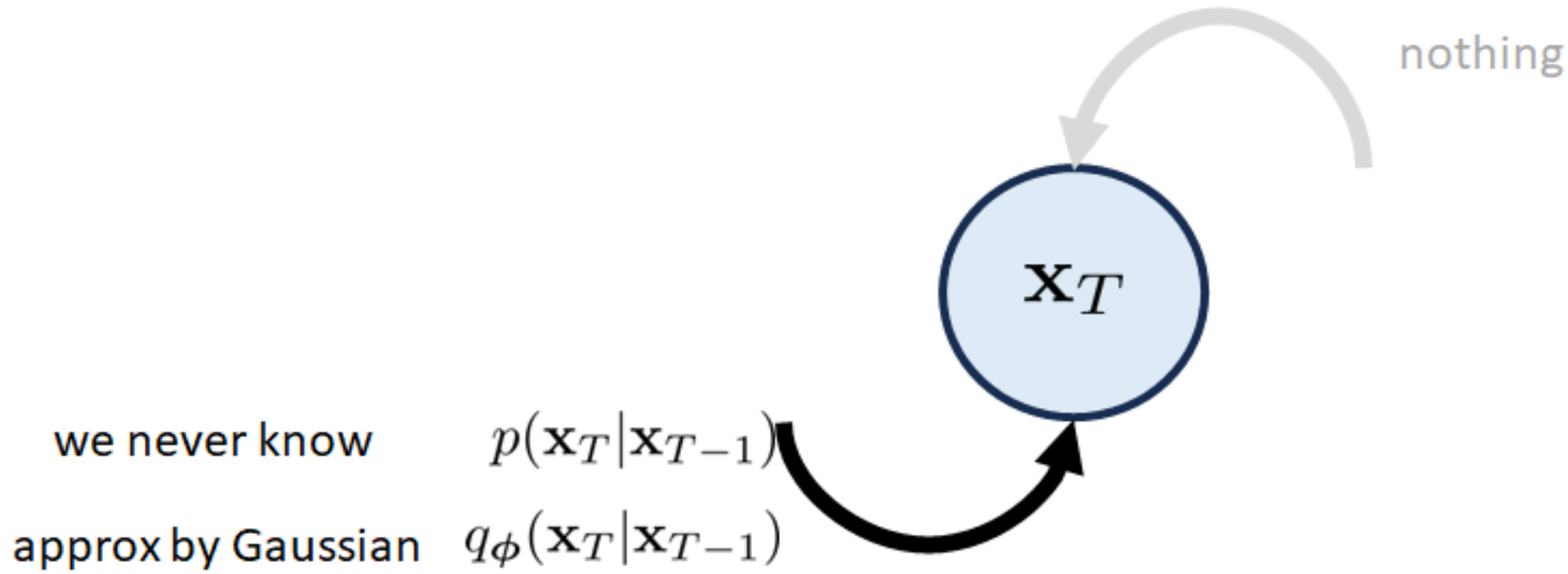
$p_{\boldsymbol{\theta}}(\mathbf{x}_0|\mathbf{x}_1)$ approx by Gaussian

Similar to what we have in VAE:

- We use q_ϕ as the encoder to approximate the underlying true conditional distribution $p(x_t \mid x_{t-1})$
- We use p_θ as the decoder to approximate the underlying true conditional distribution $p(x_t \mid x_{t+1})$

Diffusion Models

Final block:



Similar to what we have in VAE:

- We use q_ϕ as the encoder to approximate the underlying true conditional distribution $p(x_t \mid x_{t-1})$
- We use p_θ as the decoder to approximate the underlying true conditional distribution $p(x_t \mid x_{t+1})$

Diffusion Models

Encoder $q_\phi(x_t|x_{t-1})$:

$$q_\phi(x_t|x_{t-1}) := \mathcal{N}\left(x_t|\sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I\right)$$

The overall forward process (gradually adding noise to samples):

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

- The scaling factor $0 < \alpha_t < 1$
- Equivalently, the above transition probability gives the following state equation

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t$$

where $\epsilon_t \sim \mathcal{N}(0,I)$ is a random Gaussian noise with 0 mean and I covariance at the time step $t \leq T$.

Intuition: at each step, we add some Gaussian noise in hope that at the end when T is large, we will converge to $\mathcal{N}(0,I)$.

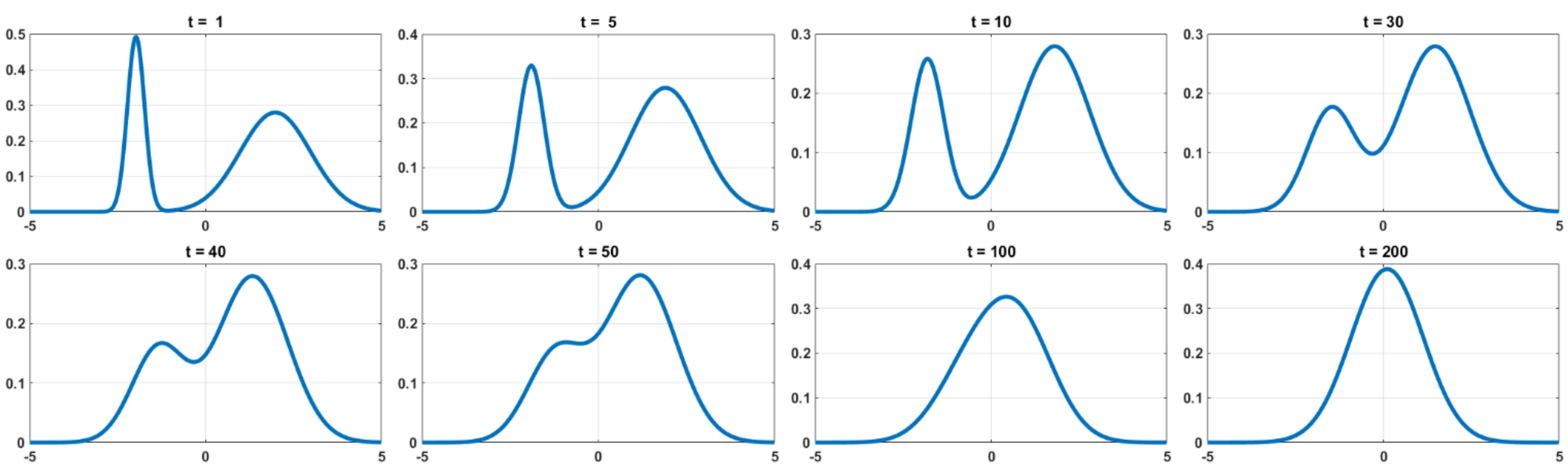
Diffusion Models

Encoder $q_\phi(x_t \mid x_{t-1})$:

$$q_\phi(x_t \mid x_{t-1}) := \mathcal{N}\left(x_t \mid \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t) I\right)$$

Example:

$$x_0 \sim p_0(x) = \pi_1 \mathcal{N}(x \mid \mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x \mid \mu_2, \sigma_2^2)$$



Diffusion Models

Encoder $q_\phi(x_t \mid x_{t-1})$:

$$q_\phi(x_t \mid x_{t-1}) := \mathcal{N}\left(x_t \mid \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t) I\right)$$

Why the specific constants $\sqrt{\alpha_t}$ and $1-\alpha_t$?

Consider the special case where $\alpha_t = \alpha \in (0,1), \forall t$. By the iterative state transition equation, we have

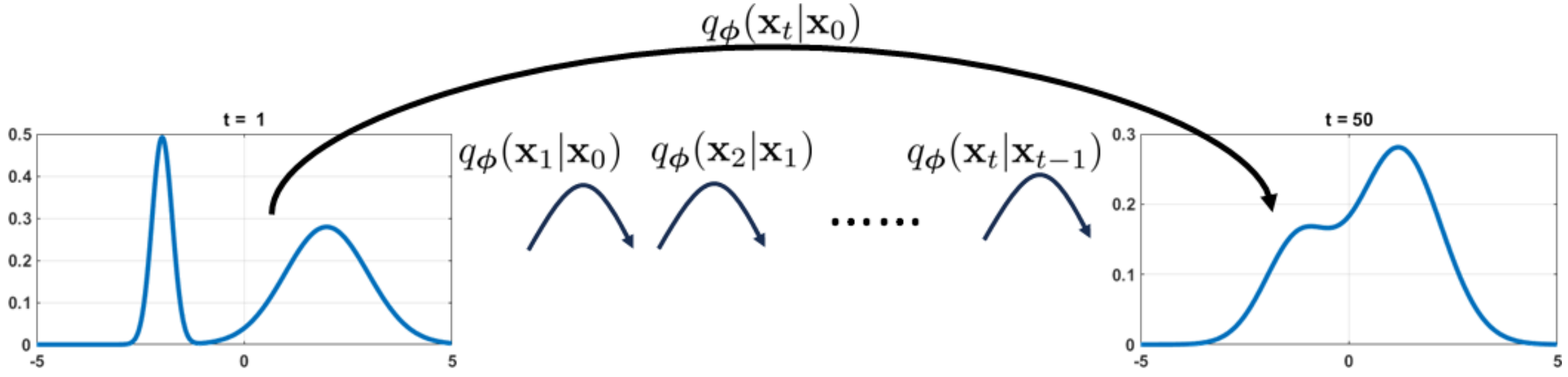
$$x_t \sim q_\phi(x_t \mid x_0) = \mathcal{N}(x_t \mid \sqrt{\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t) I)$$

where $\bar{\alpha}_t := \prod_{i=1}^{t} \alpha_i$

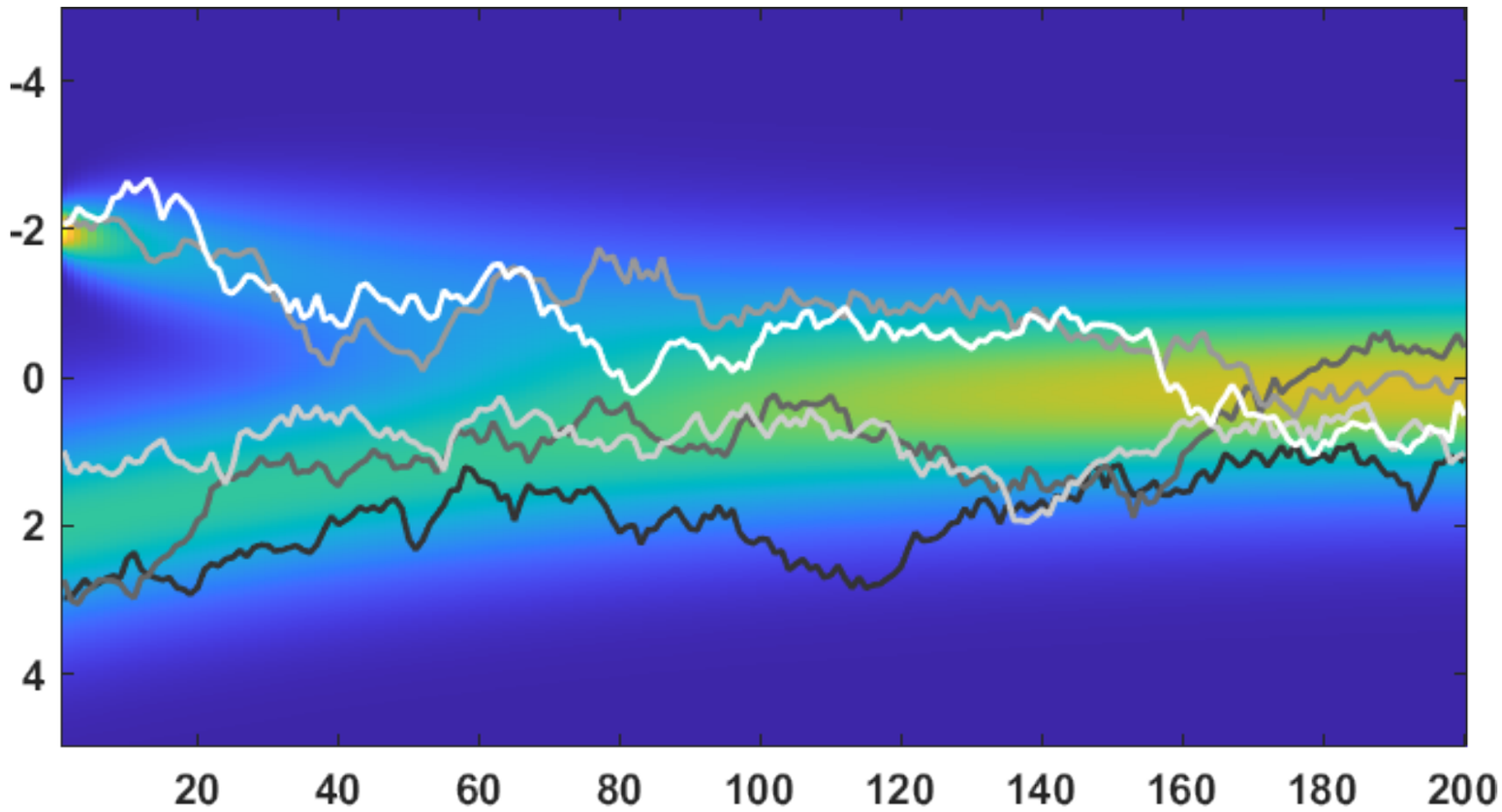
Convince yourself that as $t \to \infty$, the conditional distribution $q_\phi(x_t \mid x_0)$ will converge to $\mathcal{N}(0, I)$ for any initial distribution over x_0.

Diffusion Models

Encoder $q_\phi(x_t \mid x_{t-1})$:



Trajectory of a Gaussian mixture turning into a standard Gaussian:



Diffusion Models

Decoder $p_\theta(x_t \mid x_{t+1})$:

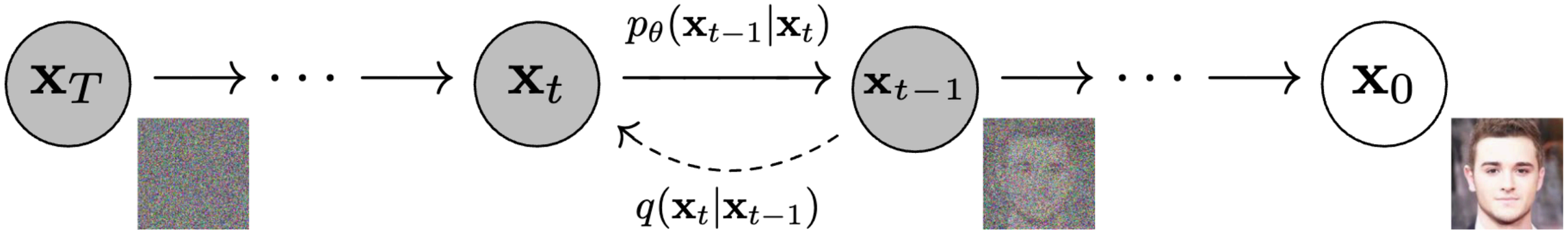
$$p_\theta(x_t \mid x_{t+1}) := \mathcal{N}\left(x_t \mid \mu_\theta(x_{t+1}, t+1), \sigma_\theta(x_{t+1}, t+1)\right)$$

- Again, similar to the case of VAE, we use neural networks to learn the parameters of a Gaussian distributions, including both the mean and the (diagonal) covariance matrix

The overall backward process (gradually generating samples from noise):

$$p_\theta(x_{0:T}) := p(x_T) \prod_{i=1}^{T} p_\theta(x_{t-1} \mid x_t)$$

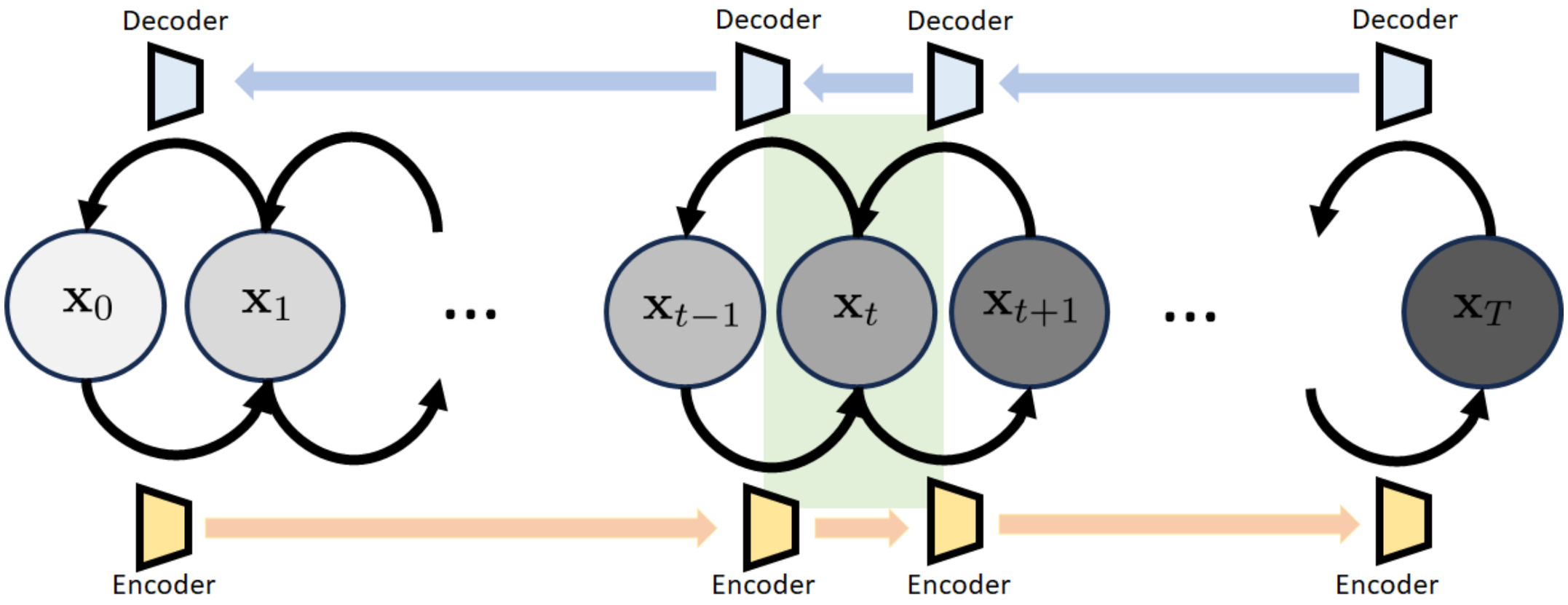
Note: $p(x_T) = p(z) = \mathcal{N}(0, I)$



Diffusion Models

Training of diffusion models:

- Similar to VAE, we will need to derive the tractable evidence lower bound (ELBO) of the true log-likelihood function $\log p_\theta(x_0)$
- Similar to VAE, the ELBO will include both the forward process (encoder) and the backward process (decoder)
- Due to its iterative nature, the overall loss will include more components than what we have in VAE



Diffusion Models

Training of diffusion models: (HW 5)

$$\mathrm{ELBO}_{\phi,\theta}(x) = \mathbb{E}_{q_\phi(x_1 \mid x_0)}[\log p_\theta(x_0 \mid x_1)]$$

$$-D_{\mathrm{KL}}\left(q_\phi(x_T \mid x_0) \,\|p(x_T)\right)$$

$$-\sum_{t=2}^{T} \mathbb{E}_{q_\phi(x_t \mid x_0)}\left[D_{\mathrm{KL}}\left(q_\phi(x_{t-1} \mid x_t, x_0) \,\|p_\theta(x_{t-1} \mid x_t)\right)\right]$$

- (Reconstruction loss): The first term measures how good the initial block is
- (Prior matching): The second term measures how good the final block is
- (Consistency): The last term measures how good the transition blocks are

Diffusion Models

Training of diffusion models:

$$\mathrm{ELBO}_{\phi,\theta}(x_0) = \mathbb{E}_{q_\phi(x_1 \mid x_0)}[\log p_\theta(x_0 \mid x_1)]$$

$$-D_{\mathrm{KL}}\left(q_\phi(x_T \mid x_0) \,\|p(x_T)\right)$$

$$-\sum_{t=2}^{T} \mathbb{E}_{q_\phi(x_t \mid x_0)}\left[D_{\mathrm{KL}}\left(q_\phi(x_{t-1} \mid x_t, x_0) \,\|p_\theta(x_{t-1} \mid x_t)\right)\right]$$

(HW 5): How to compute $q_\phi(x_{t-1} \mid x_t, x_0)$?

- Use the Bayes theorem
- It is also a Gaussian distribution (why?)

Diffusion Models

Training of diffusion models:

After around ~5 pages of algebraic derivations (no conceptual challenges, though):

$$\mathrm{ELBO}_\theta(x_0) = -\sum_{t=1}^{T} \frac{1}{2\sigma^2} \frac{(1-\alpha_t)\bar{\alpha}_{t-1}}{(1-\bar{\alpha}_t)^2} \mathbb{E}_{q_\phi(x_t \mid x_0)} \left[\|\hat{x}_\theta(x_t) - x_0\|_2^2 \right]$$

Forward process

Reconstruction loss

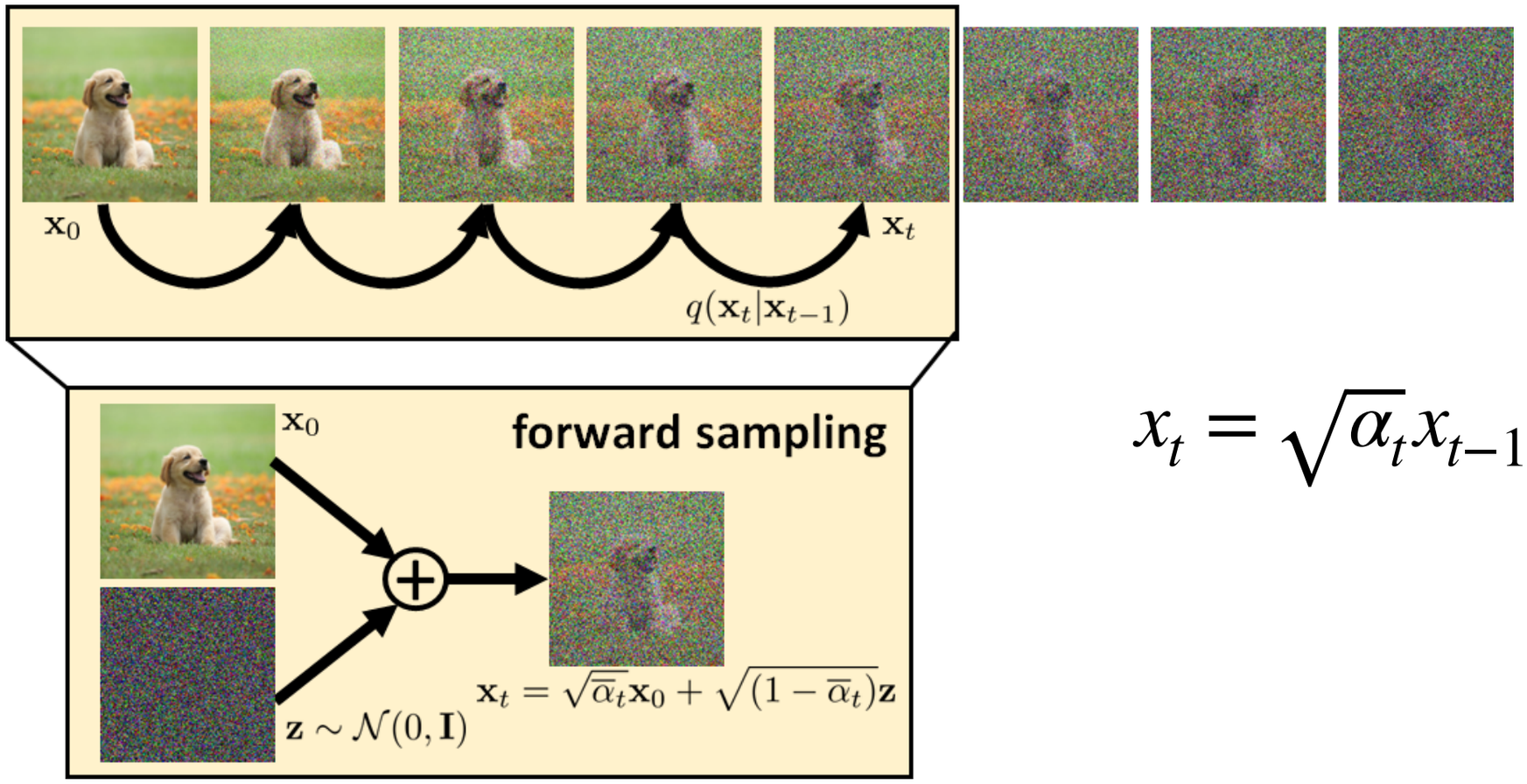
Diffusion Models

Training of diffusion models:

After around ~5 pages of algebraic derivations (no conceptual challenges, though):

$$\text{ELBO}_\theta(x_0) = -\sum_{t=1}^{T} \frac{1}{2\sigma^2} \frac{(1-\alpha_t)\bar{\alpha}_{t-1}}{(1-\bar{\alpha}_t)^2} \mathbb{E}_{q_\phi(x_t \mid x_0)} \left[\|\hat{x}_\theta(x_t) - x_0\|_2^2 \right]$$

- The overall loss function is, again, reconstruction from noisy samples



$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t$$

Diffusion Models

Training of diffusion models:

After around ~5 pages of algebraic derivations (no conceptual challenges, though):

$$\text{ELBO}_\theta(x_0) = -\sum_{t=1}^{T} \frac{1}{2\sigma^2} \frac{(1-\alpha_t)\bar{\alpha}_{t-1}}{(1-\bar{\alpha}_t)^2} \mathbb{E}_{q_\phi(x_t \mid x_0)} \left[\|\hat{x}_\theta(x_t) - x_0\|_2^2 \right]$$

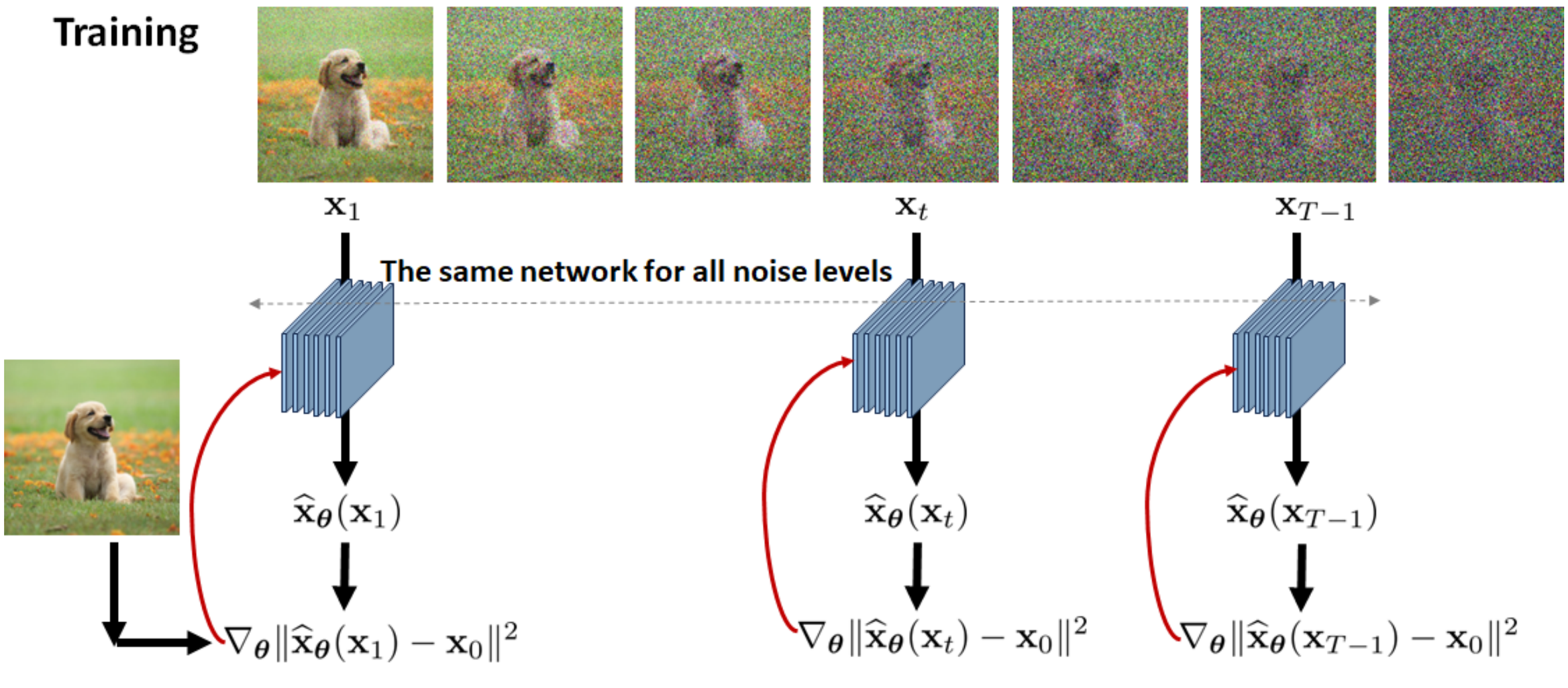
- The overall loss function is, again, reconstruction from noisy samples
- Our parametrized model (neural network) is now $\hat{x}_\theta : \mathcal{X} \to \mathcal{X}$, which takes a noise image and gives back a denoised one
- The denoising process is carefully scheduled along the whole sequential process $t = 1, \ldots, T$

Diffusion Models

Training of diffusion models:

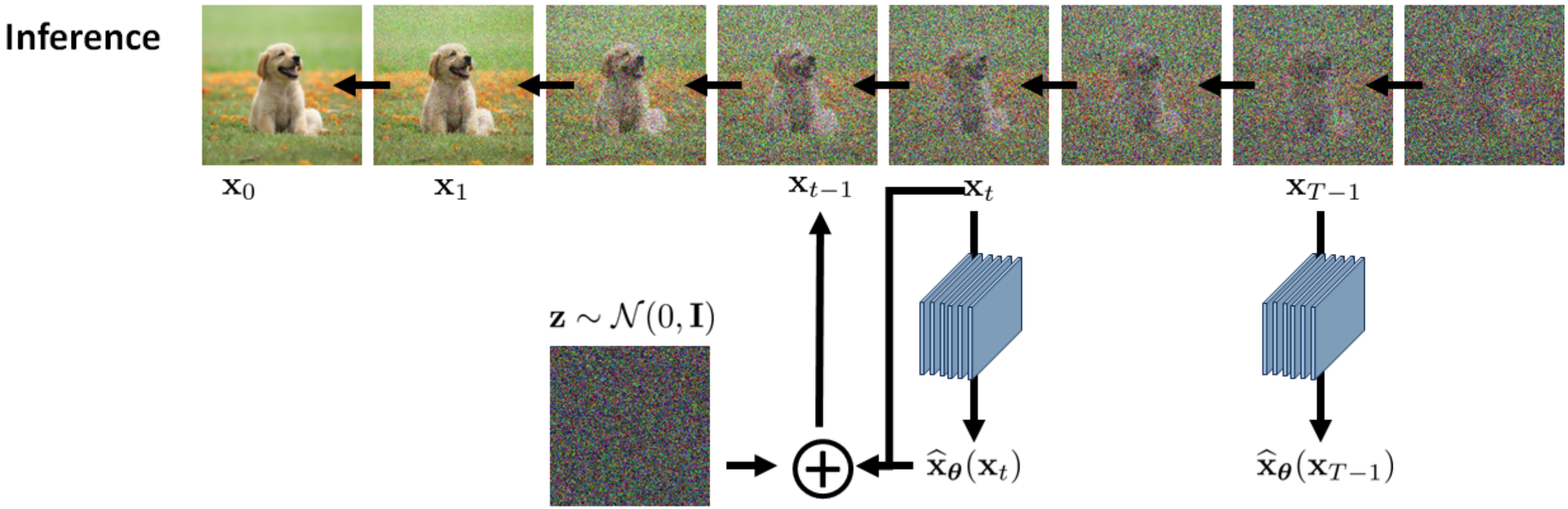
After around ~5 pages of algebraic derivations (no conceptual challenges, though):

$$\mathrm{ELBO}_\theta(x_0) = -\sum_{t=1}^{T} \frac{1}{2\sigma^2} \frac{(1-\alpha_t)\bar{\alpha}_{t-1}}{(1-\bar{\alpha}_t)^2} \mathbb{E}_{q_\phi(x_t \mid x_0)} \left[\|\hat{x}_\theta(x_t) - x_0\|_2^2 \right]$$



Diffusion Models

Inference of diffusion models:



- Start from the $t = T$ step to sample from $\mathcal{N}(0,I)$
- For $t = T - 1,\ldots,0$, gradually apply the trained denoising network to sample from $p_\theta(x_{t-1} \mid x_t)$
- Recall that in the backward process we define $p_\theta(x_{t-1} \mid x_t)$ to be a Gaussian so the sampling is tractable

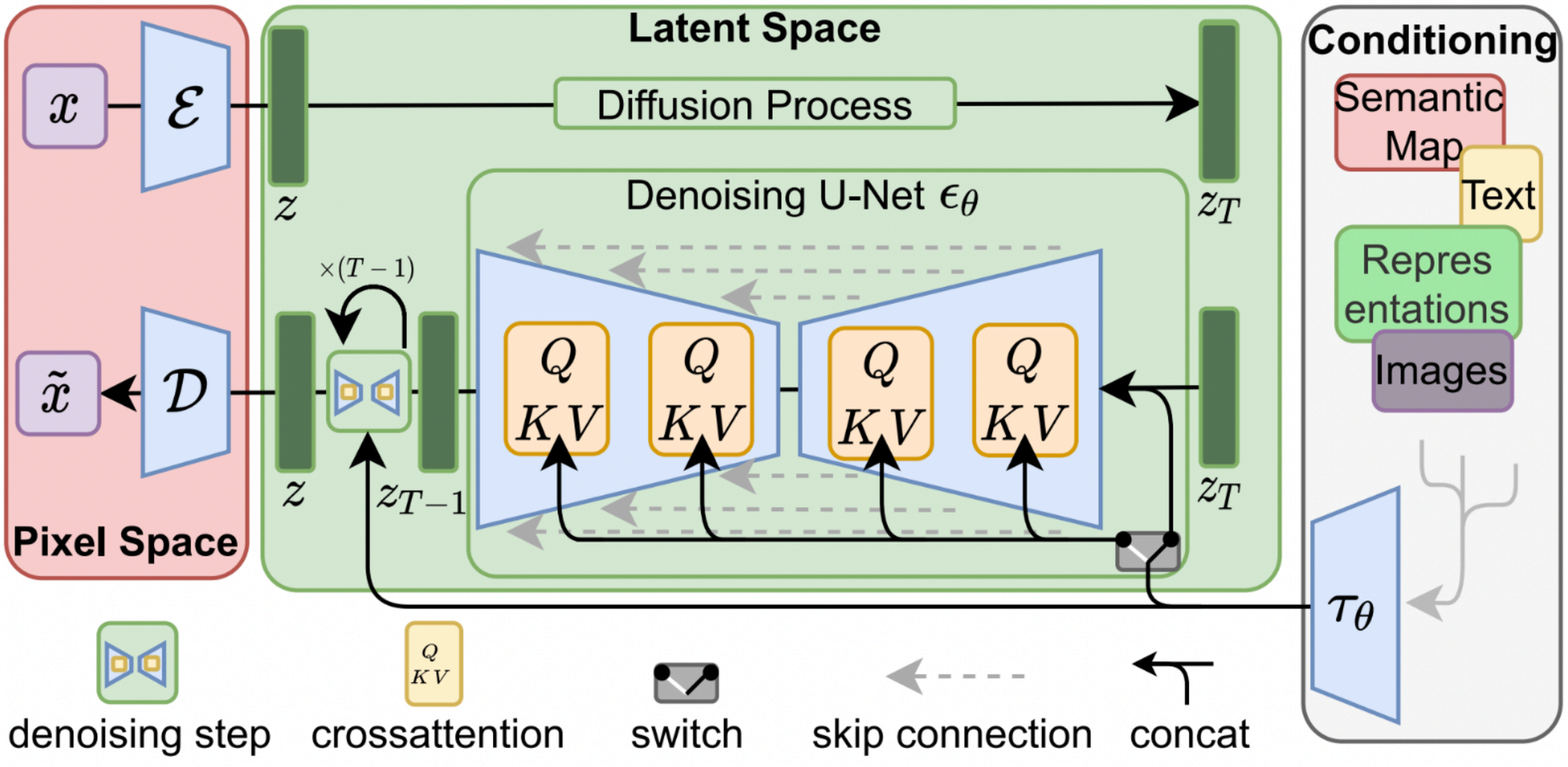
Diffusion Models

Examples:

Diffusion Models

Examples:

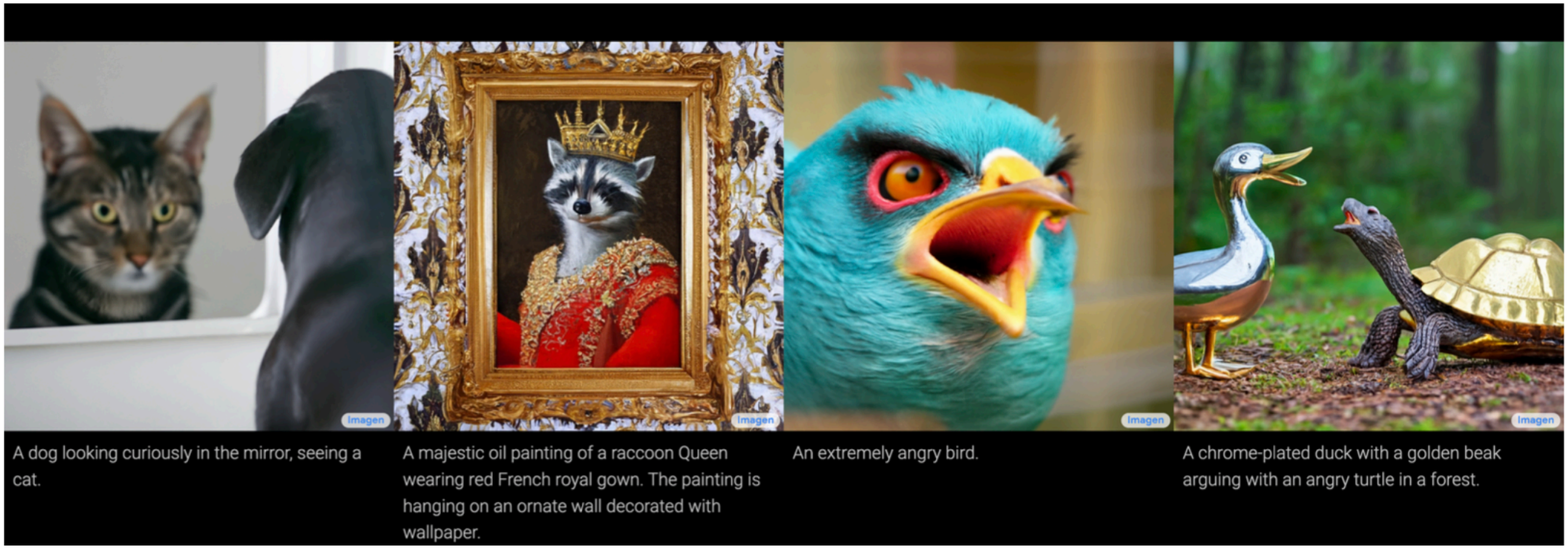
Can be combined with textual encoding to enable conditional generation, i.e., text-to-image (Imagen), text-to-video etc.

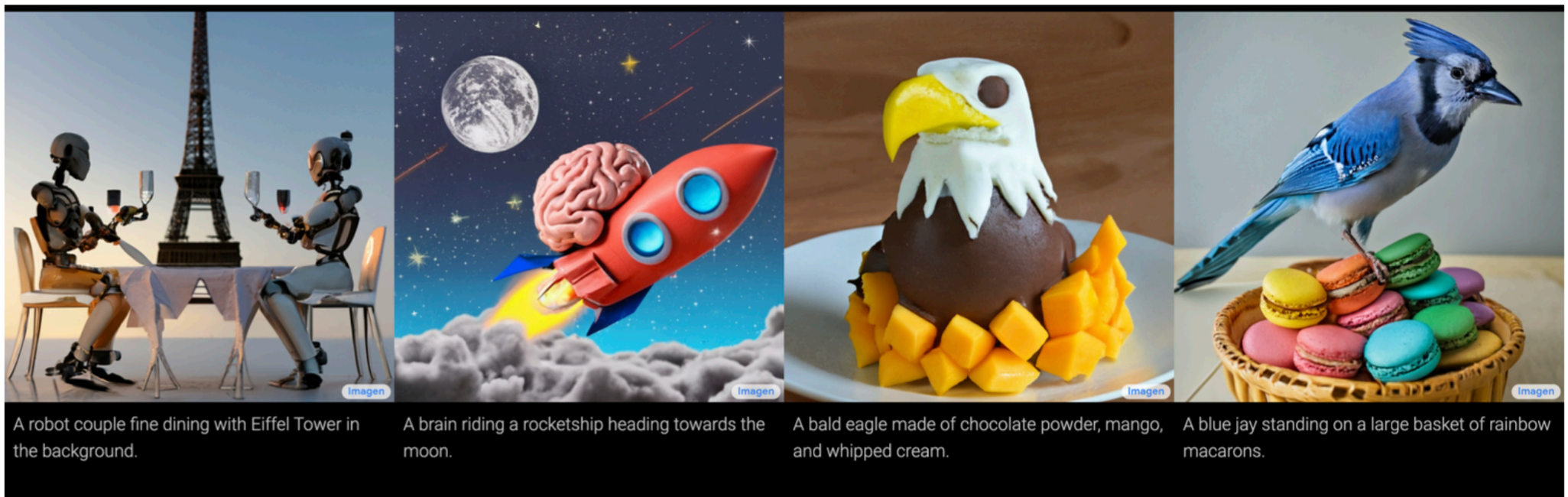


Diffusion Models

Examples:

Can be combined with textual encoding to enable conditional generation, i.e., text-to-image (Imagen), text-to-video etc.





Diffusion Models

Applications:

- Midjourney
- Huggingface stable diffusion
- Dall-E 3
- ……