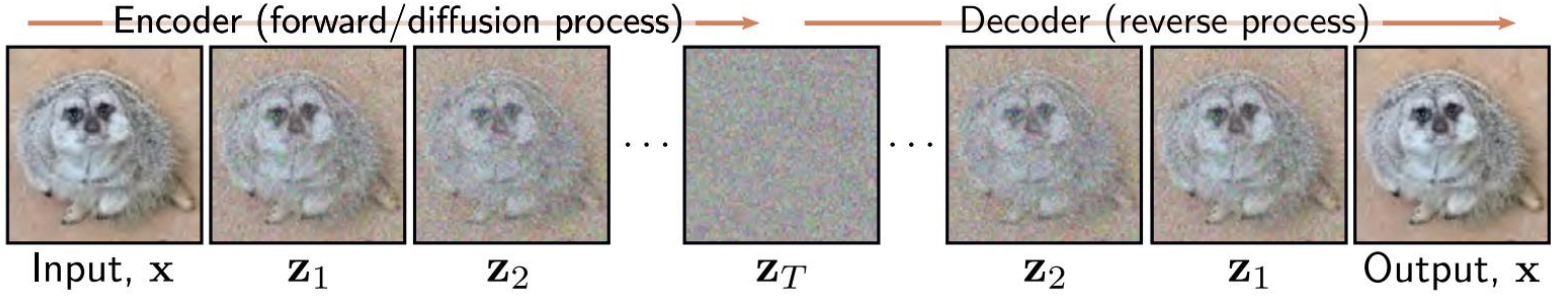
Talk 15: Diffussion Models

Overview

- **Diffusion Models:**
  - An **encoder** and a **decoder** to process data samples through latent variables.
- **Encoder:**
  - Blends input with white noise.
  - Achieves standard normal distribution in final latent variable.
- **Decoder Role:**
  - Houses all the learned parameters.
  - Trained networks reverse the encoder's steps, removing noise and recreating data.
- **Data Generation:**
  - New data is generated by sampling from the final latent distribution and processing through the decoder.



Encoder (forward process)

- The diffusion or forward process maps a data example **x** through a series of intermediate variables $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T$ with the same size as **x** according to:

$$\mathbf{z}_1 = \sqrt{1-\beta_1} \cdot \mathbf{x} + \sqrt{\beta_1} \cdot \boldsymbol{\varepsilon}_1$$

$$\mathbf{z}_t = \sqrt{1-\beta_t} \cdot \mathbf{z}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\varepsilon}_t, \quad \forall t=2,\cdots,T,$$

  - $\boldsymbol{\epsilon}_t$: noise drawn from a standard normal distribution.
  - $\boldsymbol{\beta}_t \in [0, 1]$ increase and determine how quickly the noise is blended and are collectively known as the noise schedule.
  - Sufficient steps T → standard normal distribution $q(z_T|x) = q(z_T)$

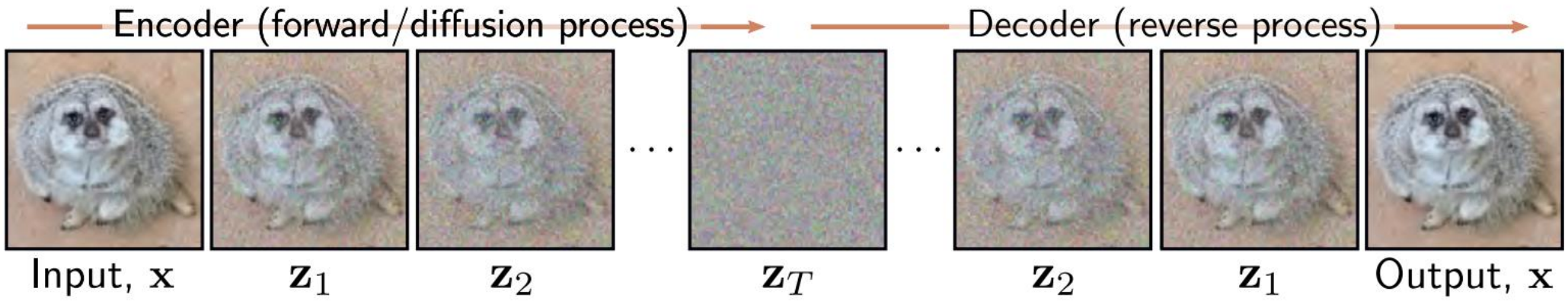
$$q(\mathbf{z}_1 \mid x) = N\left[\sqrt{1-\beta_1} \cdot \mathbf{x}, \beta_1 \mathbf{I}\right],$$

$$q(\mathbf{z}_t \mid \mathbf{z}_{t-1}) = N\left[\sqrt{1-\beta_t} \cdot \mathbf{z}_{t-1}, \beta_t \mathbf{I}\right], \quad \forall t=2,\cdots,T,$$

Encoder (forward process)

- The joint distribution of all of the latent variables $\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_T}$ given input $\mathbf{x}$ is:

$$q(\mathbf{z}_{1\ldots T} \mid \mathbf{x}) = q(\mathbf{z}_1 \mid \mathbf{x}) \prod_{t=2}^{T} q(\mathbf{z}_t \mid \mathbf{z}_{t-1})$$



The forward process: gradually mixes the data with noise until only noise remains.

The decoder passes the data back through the latent variables, removing noise at each stage.

New examples are generated by sampling noise vectors $\mathbf{z}_T$ and passing them through the decoder.

Diffusion kernel $q(z_t|x)$

- We can directly draw samples $\mathbf{z}_t$ given initial data point $\mathbf{x}$ without computing the intermediate variables $\mathbf{z}_1 \ldots \mathbf{z}_{t-1}$

$$\mathbf{z}_1 = \sqrt{1-\beta_1}\cdot\mathbf{x} + \sqrt{\beta_1}\cdot\boldsymbol{\varepsilon}_1$$

$$\mathbf{z}_2 = \sqrt{1-\beta_2}\cdot\mathbf{z}_1 + \sqrt{\beta_2}\cdot\boldsymbol{\varepsilon}_2$$

$$= \sqrt{1-\beta_2}\cdot\left(\sqrt{1-\beta_1}\cdot\mathbf{x} + \sqrt{\beta_1}\cdot\boldsymbol{\varepsilon}_1\right) + \sqrt{\beta_2}\cdot\boldsymbol{\varepsilon}_2$$

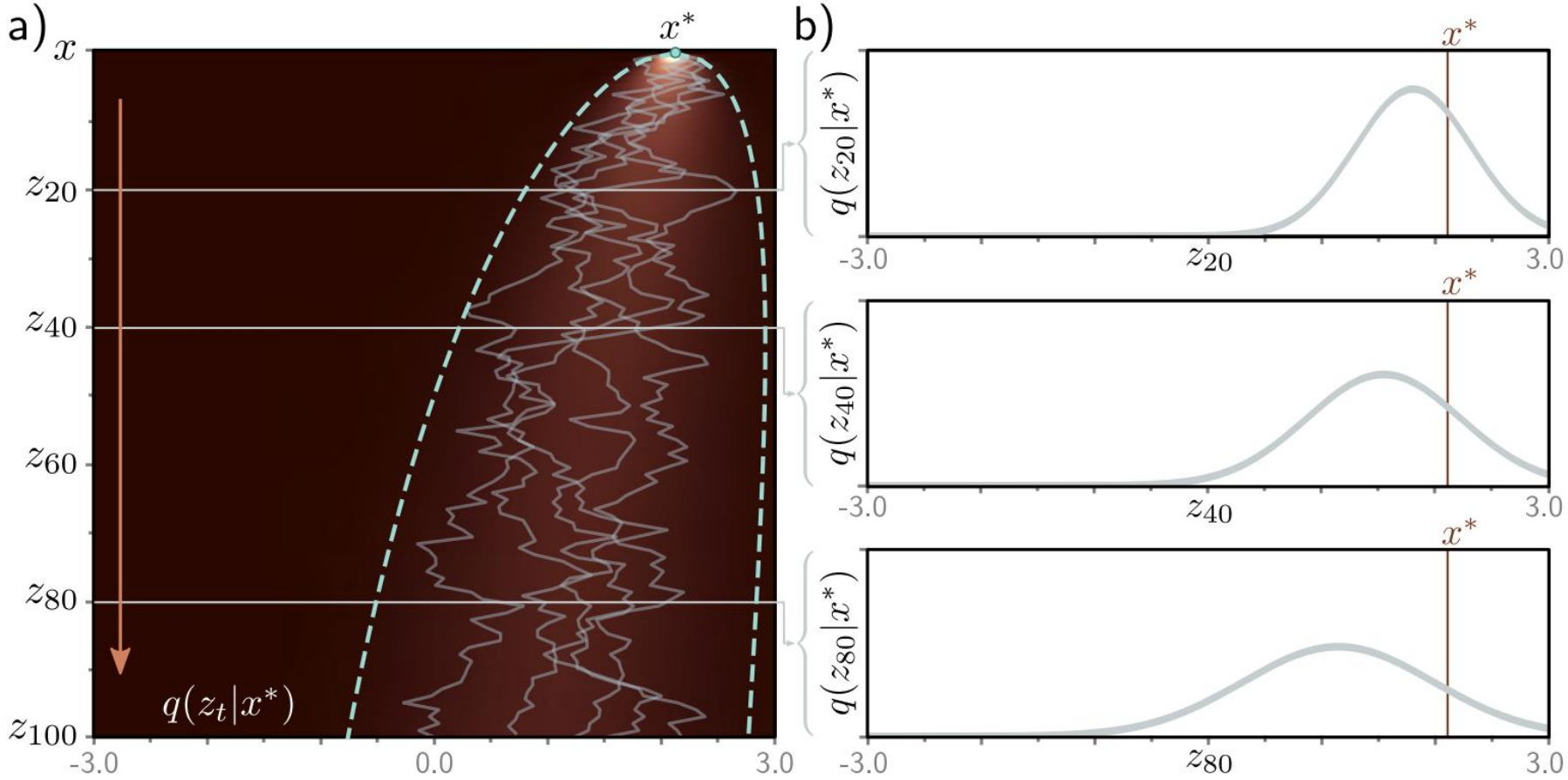
$$= \sqrt{1-\beta_2}\sqrt{1-\beta_1}\cdot\mathbf{x} + \sqrt{1-\beta_2}\sqrt{\beta_1}\cdot\boldsymbol{\varepsilon}_1 + \sqrt{\beta_2}\cdot\boldsymbol{\varepsilon}_2$$

$$= \sqrt{(1-\beta_2)(1-\beta_1)}\cdot\mathbf{x} + \sqrt{1-(1-\beta_2)(1-\beta_1)}\,\boldsymbol{\varepsilon}$$

$$\mathbf{z}_t = \sqrt{\alpha_t}\cdot\mathbf{x} + \sqrt{1-\alpha_t}\cdot\boldsymbol{\varepsilon}, \qquad \alpha_t = \prod_{k=1}^{t}(1-\beta_k)$$

$$q(\mathbf{z}_t \mid \mathbf{x}) = N\left[\sqrt{\alpha_t}\cdot\mathbf{x}, (1-\alpha_t)\mathbf{I}\right]$$

Diffusion kernel $q(z_t|x)$



- The point $x* = 2.0$ is propagated through the latent variables using diffusion equation.
- The diffusion kernel $q(z_t|x*)$ is the probability distribution over variable z_t given, started from $x*$.
- The normal distribution whose mean moves toward zero and whose variance increases as t increases. Heatmap shows $q(z_t|x*)$ for each variable.

Marginal distributions $q(\mathbf{z}_t)$

- The marginal distribution $q(\mathbf{z}_t)$ is the probability of observing a value of $\mathbf{z}_t$ given the distribution of possible starting points **x** and the possible diffusion paths for each starting point.

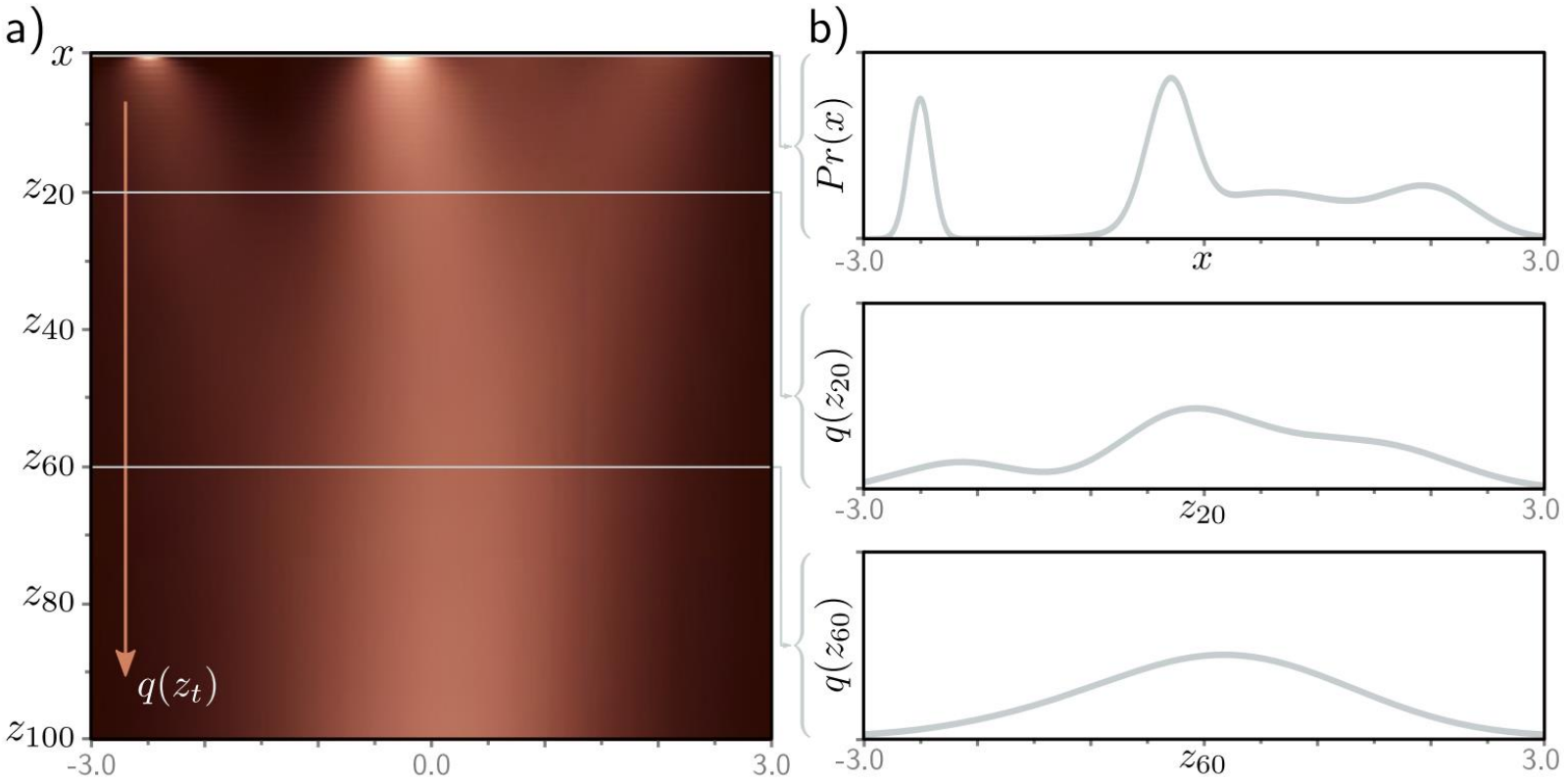
- Marginalize over all the variables except $\mathbf{z}_t$:

$$q(\mathbf{z}_t) = \iint q(\mathbf{z}_{1:t}, \mathbf{x}) d\mathbf{z}_{1:(t-1)} d\mathbf{x}$$
$$= \iint q(\mathbf{z}_{1:t} \mid \mathbf{x}) p(\mathbf{x}) d\mathbf{z}_{1:(t-1)} d\mathbf{x}$$

since we now have an expression for the diffusion kernel $q(\mathbf{z}_t|\mathbf{x})$:

$$q(\mathbf{z}_t) = \int q(\mathbf{z}_t \mid \mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

Marginal distributions $q(\mathbf{z}_t)$



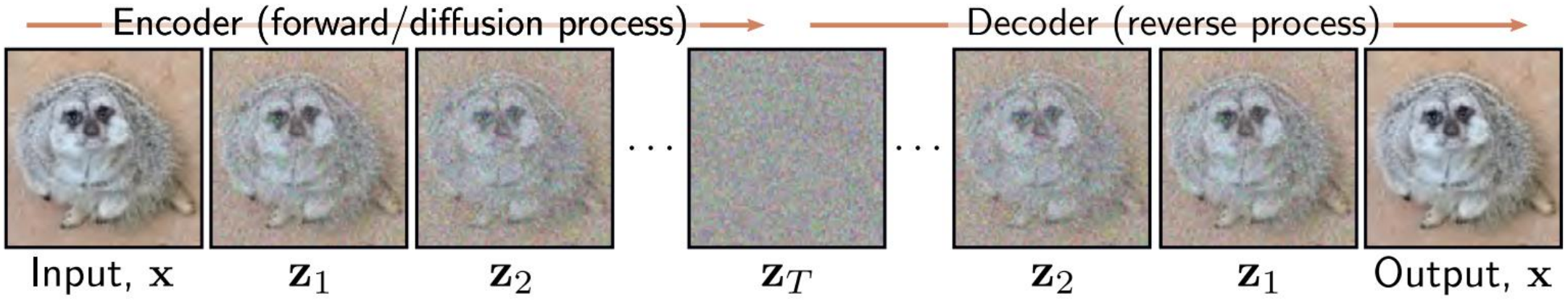
- Given an initial density $P\ (x)$ (top row), the diffusion process gradually blurs the distribution moves it toward a standard normal distribution.
- Each subsequent horizontal line of heatmap represents a marginal distribution $q(z_t)$. b) The top graph shows the initial distribution $P\ (x)$. The other two graphs show the marginal distributions $q(z_{20})$ and $q(z_{60})$, respectively.

Conditional distribution $q(\mathbf{z}_{t-1}|\mathbf{z}_t)$

- The conditional probability $q(\mathbf{z}_t \mid \mathbf{z}_{t-1})$ as the mixing process. To reverse this process, we apply Bayes' rule:

$$q(\mathbf{z}_{t-1} \mid \mathbf{z}_t) = \frac{q(\mathbf{z}_t \mid \mathbf{z}_{t-1})q(\mathbf{z}_{t-1})}{q(\mathbf{z}_t)}$$

- When we build the decoder, we will approximate the reverse process using a normal distribution.



Conditional distribution $q(z_{t-1}|z_t)$

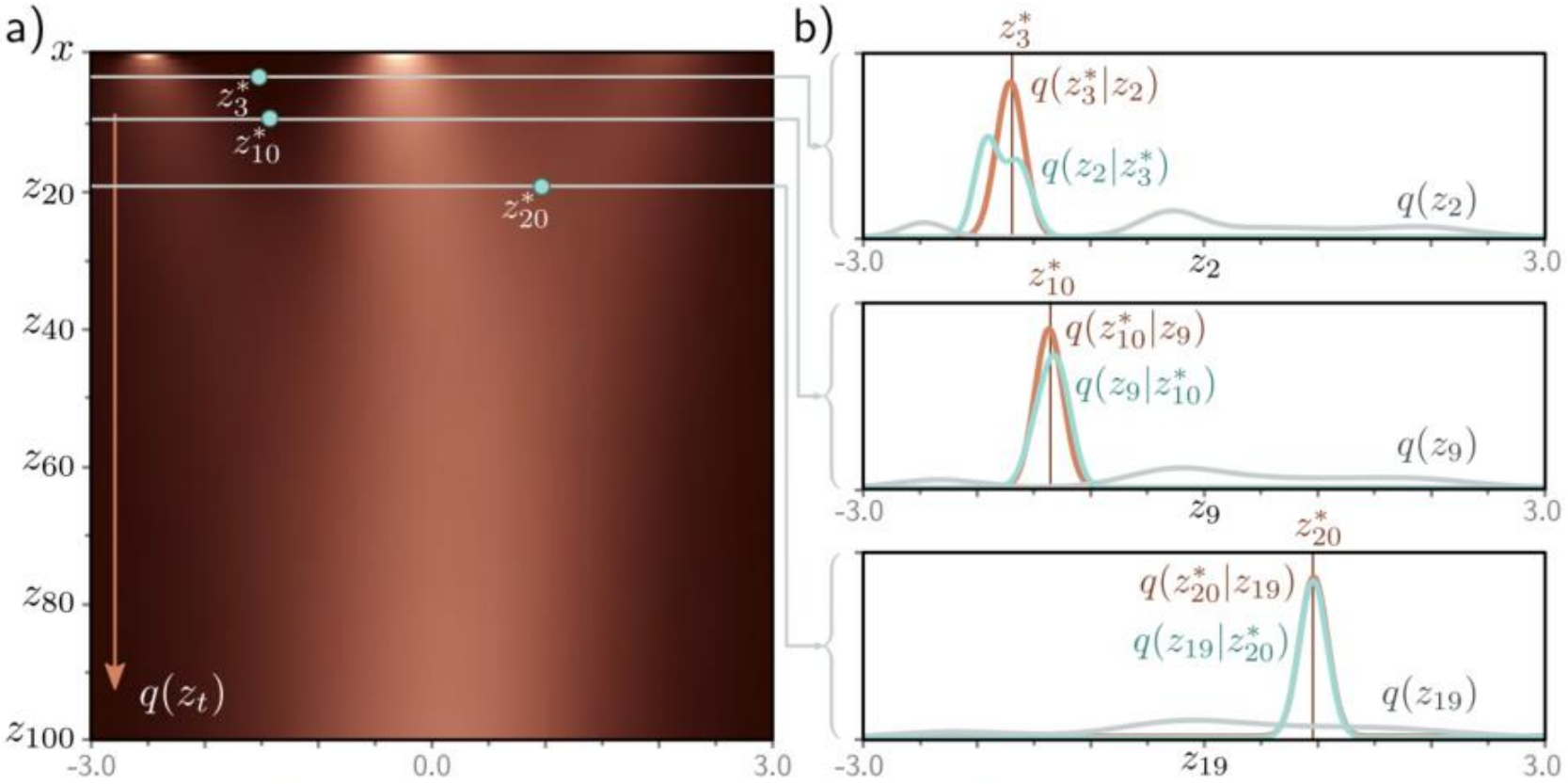


Figure 18.5 Conditional distribution $q(z_{t-1}|z_t)$. a) The marginal densities $q(z_t)$ with three points z_t^* highlighted. b) The probability $q(z_{t-1}|z_t^*)$ (cyan curves) is computed via Bayes' rule and is proportional to $q(z_t^*|z_{t-1})q(z_{t-1})$. In general, it is not normally distributed (top graph), although often the normal is a good approximation (bottom two graphs). The first likelihood term $q(z_t^*|z_{t-1})$ is normal in z_{t-1} (equation 18.2) with a mean that is slightly further from zero than z_t^* (brown curves). The second term is the marginal density $q(z_{t-1})$ (gray curves).

Conditional diffusion distribution $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})$

- We could not find the conditional distribution $q(\mathbf{z}_{t-1}|\mathbf{z}_t)$ because we do not know the marginal distribution $q(\mathbf{z}_{t-1})$. However, we know the starting variable x and the distribution $q(\mathbf{z}_{t-1}|\mathbf{x})$ which is normally distributed.

$$q(\mathbf{z}_{t-1}\,|\,\mathbf{z}_t,\mathbf{x}) = \frac{q(\mathbf{z}_t\,|\,\mathbf{z}_{t-1},\mathbf{x})q(\mathbf{z}_{t-1}\,|\,\mathbf{x})}{q(\mathbf{z}_t\,|\,\mathbf{x})} \;\propto\; q(\mathbf{z}_t\,|\,\mathbf{z}_{t-1})q(\mathbf{z}_{t-1}\,|\,\mathbf{x})$$

$$= N\left(\mathbf{z}_t\,|\,\sqrt{1-\beta_t}\cdot\mathbf{z}_{t-1},\beta_t\mathbf{I}\right)\cdot N\left(\mathbf{z}_{t-1}\,|\,\sqrt{\alpha_{t-1}}\cdot\mathbf{x},(1-\alpha_{t-1})_t\mathbf{I}\right)$$

$$\propto N\left(\mathbf{z}_{t-1}\,|\,\frac{1}{\sqrt{1-\beta_t}}\cdot\mathbf{z}_t,\frac{\beta_t}{1-\beta_t}\mathbf{I}\right)\cdot N\left(\mathbf{z}_{t-1}\,|\,\sqrt{\alpha_{t-1}}\cdot\mathbf{x},(1-\alpha_{t-1})_t\mathbf{I}\right)$$

$$= N\left(\mathbf{z}_{t-1}\left|\frac{1-\alpha_{t-1}}{1-\alpha_t}\sqrt{1-\beta_t}\mathbf{z}_t+\frac{\beta_t\sqrt{\alpha_{t-1}}}{1-\alpha_t}\mathbf{x},\frac{\beta_t(1-\alpha_{t-1})}{1-\alpha_t}\mathbf{I}\right.\right)$$

$$\mathrm{Norm}_{\mathbf{v}}\left[\mathbf{A}\mathbf{w},\mathbf{B}\right] \propto \mathrm{Norm}_{\mathbf{w}}\left[\left(\mathbf{A}^T\mathbf{B}^{-1}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{B}^{-1}\mathbf{v},\left(\mathbf{A}^T\mathbf{B}^{-1}\mathbf{A}\right)^{-1}\right],$$

$$\mathrm{Norm}_{\mathbf{w}}[\mathbf{a},\mathbf{A}]\cdot\mathrm{Norm}_{\mathbf{w}}[\mathbf{b},\mathbf{B}] \;\propto\; \mathrm{Norm}_{\mathbf{w}}\left[\left(\mathbf{A}^{-1}+\mathbf{B}^{-1}\right)^{-1}\left(\mathbf{A}^{-1}\mathbf{a}+\mathbf{B}^{-1}\mathbf{b}\right),\left(\mathbf{A}^{-1}+\mathbf{B}^{-1}\right)^{-1}\right],$$

Conditional diffusion distribution $q(z_{t-1}|z_t, x)$

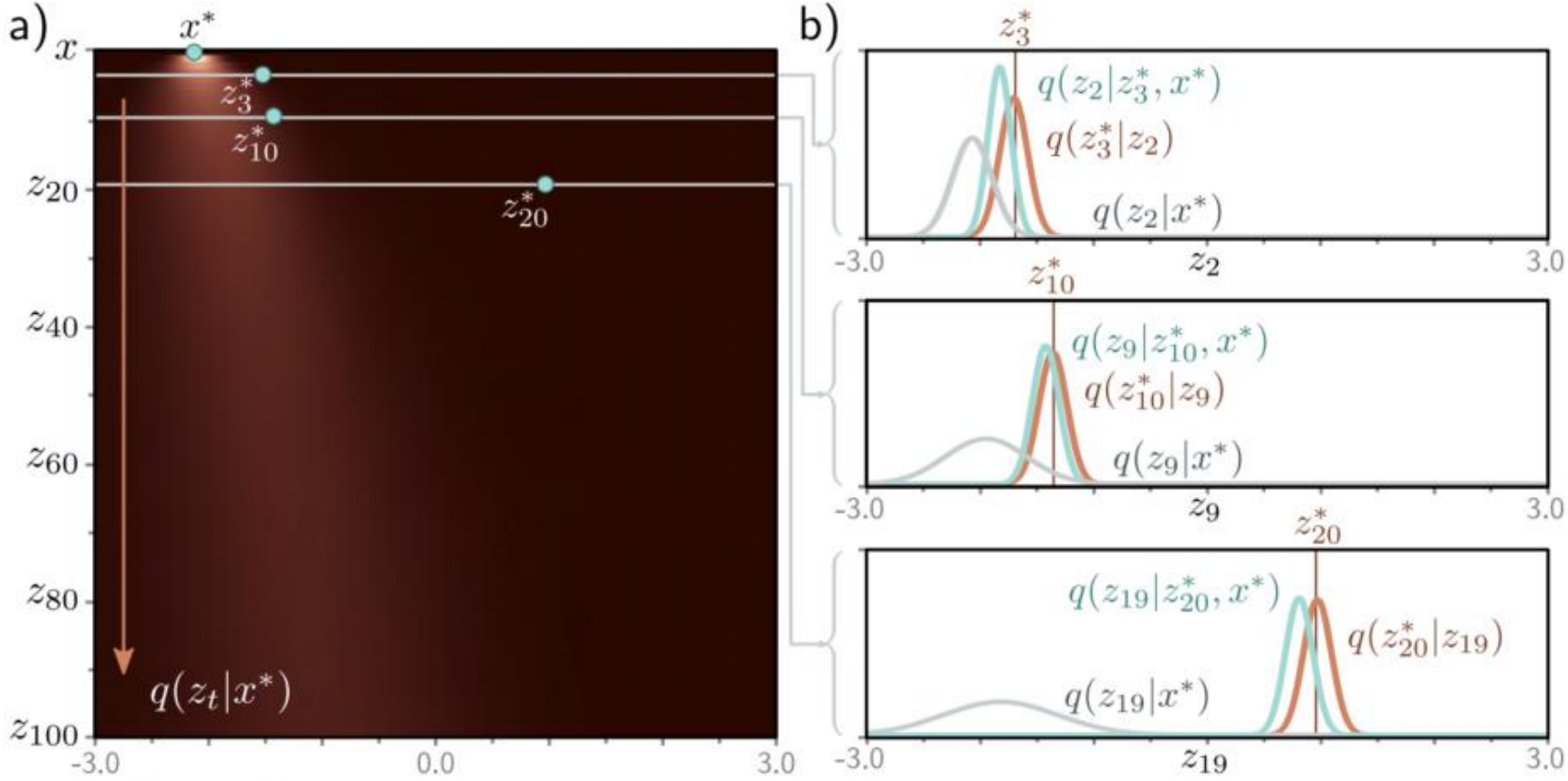


Figure 18.6 Conditional distribution $q(z_{t-1}|z_t, x)$. a) Diffusion kernel for $x^* = -2.1$ with three points z_t^* highlighted. b) The probability $q(z_{t-1}|z_t^*, x^*)$ is computed via Bayes' rule and is proportional to $q(z_t^*|z_{t-1})q(z_{t-1}|x^*)$. This *is* normally distributed and can be computed in closed form. The first likelihood term $q(z_t^*|z_{t-1})$ is normal in z_{t-1} (equation 18.2) with a mean that is slightly further from zero than z_t^* (brown curves). The second term is the diffusion kernel $q(z_{t-1}|x^*)$ (gray curves).

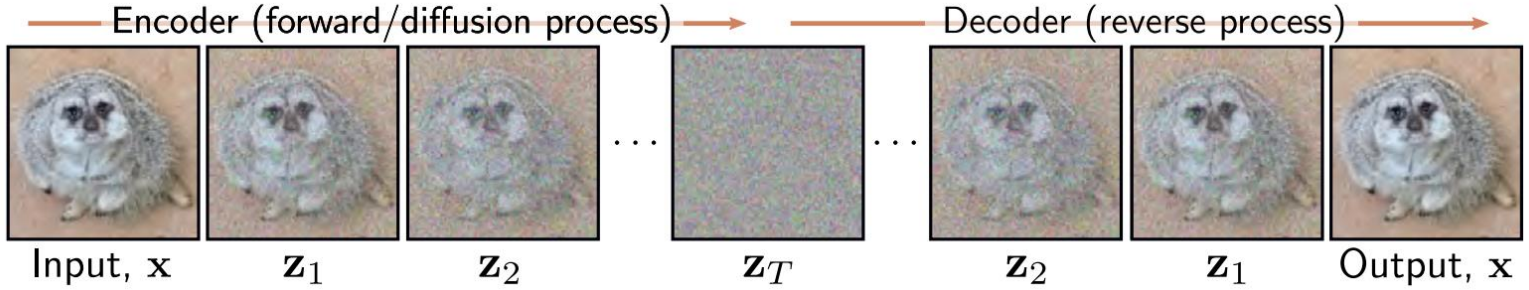
Decoder model (Reverse process)

- **The reverse process:** a series of probabilistic mappings back from latent variable $\mathbf{z_T}$ to $\mathbf{z_{T-1}}$, from $\mathbf{z_{T-1}}$ to $\mathbf{z_{T-2}}$, and so on, until we reach the data **x**.
- The true reverse distributions $q(\mathbf{z_{t-1}}|\mathbf{z_t})$ of the diffusion process are complex multi-modal distributions that depend on the data distribution *Pr*(**x**).

$$p(\mathbf{z}_T) = N(\mathbf{z}_T \mid 0, \mathbf{I}),$$

$$p(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \Phi_t) = N(\mathbf{z}_{t-1} \mid f_t(\mathbf{z}_t, \Phi_t), \sigma_t^2 \mathbf{I})), \quad \forall t=T,\cdots,1 \quad \mathbf{x} = \mathbf{z}_0$$

  - $f_t[\mathbf{z_t}, \phi_t]$: a neural network that computes the mean of the normal distribution. $\{\sigma^2_t\}$: predetermined
- **Generation:** We start by drawing $\mathbf{z_T}$ from $\mathbf{Pr(z_T)}$.



Training

- The joint distribution of the observed variable **x** and the latent variables $\{\mathbf{z}_t\}$ is:

$$p(\mathbf{x}, \mathbf{z}_{1:T} \mid \Phi_{1:T}) = p(\mathbf{z}_T) \prod_{t=1}^{T} p(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \Phi_{1:T}), \qquad \text{where } \mathbf{z}_0 = \mathbf{x}$$

- The likelihood $p(\mathbf{x} \mid \Phi_{1:T})$ is found by marginalizing over the latent variables:

$$p(\mathbf{x} \mid \Phi_{1:T}) = \int p(\mathbf{x}, \mathbf{z}_{1:T} \mid \Phi_{1:T}) d\mathbf{z}_{1:T}$$

- Maximize the log-likelihood of the training data $\{\mathbf{x}_i\}$ with respect to the parameters $\boldsymbol{\phi}$:

$$\hat{\Phi}_{1:T} = \arg\max_{\Phi_{1:T}} \left[\sum_{k=1}^{N} \ln[\, p(\mathbf{x}_k \mid \Phi_{1:T})] \right]$$

- Variational Approximation: Use lower bound on the likelihood and optimize the parameters $\boldsymbol{\phi}_{1\ldots T}$ with respect to this bound exactly as we did for the VAE

Evidence lower bound (ELBO)

- To derive the lower bound, we multiply and divide the log-likelihood by **the encoder distribution** $q(\mathbf{z}_{1...T}|\mathbf{x})$ and apply Jensen's inequality:

$$\log[p(\mathbf{x}|\Phi_{1:T})] = \log\left[\int p(\mathbf{x},\mathbf{z}_{1:T}|\Phi_{1:T})d\mathbf{z}_{1:T}\right]$$

$$= \log\left[\int \frac{p(\mathbf{x},\mathbf{z}_{1:T}|\Phi_{1:T})}{q(\mathbf{z}_{1:T}|\mathbf{x})} q(\mathbf{z}_{1:T}|\mathbf{x})d\mathbf{z}_{1:T}\right]$$

$$\geq \int q(\mathbf{z}_{1:T}|\mathbf{x}) \log\left[\frac{p(\mathbf{x},\mathbf{z}_{1:T}|\Phi_{1:T})}{q(\mathbf{z}_{1:T}|\mathbf{x})}\right] d\mathbf{z}_{1:T}$$

$$\text{ELBO}(\Phi_{1:T}) = \int q(\mathbf{z}_{1:T}|\mathbf{x}) \log\left[\frac{p(\mathbf{x},\mathbf{z}_{1:T}|\Phi_{1:T})}{q(\mathbf{z}_{1:T}|\mathbf{x})}\right] d\mathbf{z}_{1:T}$$

Simplifying the ELBO

- We first substitute in the definitions for the numerator and denominator:

$$\log\left[\frac{p(\mathbf{x},\mathbf{z}_{1:T}\mid\Phi_{1:T})}{q(\mathbf{z}_{1:T}\mid\mathbf{x})}\right]=\log\left[\frac{p(\mathbf{z}_T)\prod_{t=1}^{T}p(\mathbf{z}_{t-1}\mid\mathbf{z}_t,\Phi_t)}{q(\mathbf{z}_1\mid\mathbf{x})\prod_{t=2}^{T}q(\mathbf{z}_t\mid\mathbf{z}_{t-1})}\right]$$

$$=\log\left[\frac{p(\mathbf{z}_0\mid\mathbf{z}_1,\Phi_1)}{q(\mathbf{z}_1\mid\mathbf{x})}\right]+\log\left[\frac{\prod_{t=2}^{T}p(\mathbf{z}_{t-1}\mid\mathbf{z}_t,\Phi_t)}{\prod_{t=2}^{T}q(\mathbf{z}_t\mid\mathbf{z}_{t-1})}\right]+\log p(\mathbf{z}_T)$$

Using relation

$$q(\mathbf{z}_t\mid\mathbf{z}_{t-1})=q(\mathbf{z}_t\mid\mathbf{z}_{t-1},\mathbf{x})=\frac{q(\mathbf{z}_{t-1}\mid\mathbf{z}_t,\mathbf{x})q(\mathbf{z}_t\mid\mathbf{x})}{q(\mathbf{z}_{t-1}\mid\mathbf{x})}$$

$$\log\left[\frac{p(\mathbf{x},\mathbf{z}_{1:T}\mid\Phi_{1:T})}{q(\mathbf{z}_{1:T}\mid\mathbf{x})}\right]=\log\left[\frac{p(\mathbf{x}\mid\mathbf{z}_1,\Phi_1)}{q(\mathbf{z}_1\mid\mathbf{x})}\right]+\log\left[\frac{\prod_{t=2}^{T}p(\mathbf{z}_{t-1}\mid\mathbf{z}_t,\Phi_t)}{\prod_{t=2}^{T}q(\mathbf{z}_{t-1}\mid\mathbf{z}_t,\mathbf{x})}\frac{q(\mathbf{z}_{t-1}\mid\mathbf{x})}{q(\mathbf{z}_t\mid\mathbf{x})}\right]+\log p(\mathbf{z}_T)$$

$$=\log\left[p(\mathbf{x}\mid\mathbf{z}_1,\Phi_1)\right]+\log\left[\frac{\prod_{t=2}^{T}p(\mathbf{z}_{t-1}\mid\mathbf{z}_t,\Phi_t)}{\prod_{t=2}^{T}q(\mathbf{z}_{t-1}\mid\mathbf{z}_t,\mathbf{x})}\right]+\log\left[\frac{p(\mathbf{z}_T)}{q(\mathbf{z}_T\mid\mathbf{x})}\right]$$

Simplifying the ELBO

- The simplified ELBO is hence:

$$\mathrm{ELBO}[\Phi_{1:T}] = \int q(\mathbf{z}_{1:T} \mid \mathbf{x}) \log \left[\frac{p(\mathbf{x}, \mathbf{z}_{1:T} \mid \Phi_{1:T})}{q(\mathbf{z}_{1:T} \mid \mathbf{x})} \right] \mathrm{d}\mathbf{z}_{1:T}$$

$$\approx \int q(\mathbf{z}_{1:T} \mid \mathbf{x}) \left(\log \left[p(\mathbf{x} \mid \mathbf{z}_1, \Phi_1) \right] + \log \left[\prod_{t=2}^{T} \frac{p(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \Phi_t)}{q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x})} \right] \right) \mathrm{d}\mathbf{z}_{1:T}$$

$$= E_{q(\mathbf{z}_1 \mid \mathbf{x})} \left[\log \left[p(\mathbf{x} \mid \mathbf{z}_1, \Phi_1) \right] \right] - \sum_{t=2}^{T} E_{q(\mathbf{z}_t \mid \mathbf{x})} D_{KL} \left[q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x}) \,\|\, p(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \Phi_t) \right]$$

Analyzing the ELBO

- The first probability term in the ELBO

$$p(\mathbf{x}\,|\,\mathbf{z}_1,\Phi_1)=N(\mathbf{x}\,|\,f_1(\mathbf{z}_1,\Phi_1),\sigma_1^2\mathbf{I})),$$

- The KL divergence terms in the ELBO measure the distance between $\Pr(\mathbf{z_{t-1}}|\mathbf{z_t},\boldsymbol{\phi_t})$ and $q(\mathbf{z_{t-1}}|\mathbf{z_t},\mathbf{x})$:

$$p(\mathbf{z}_{t-1}\,|\,\mathbf{z}_t,\Phi_t)=N(\mathbf{z}_{t-1}\,|\,f_t(\mathbf{z}_t,\Phi_t),\sigma_t^2\mathbf{I})),\quad \forall t=T,\cdots,1$$

$$q(\mathbf{z}_{t-1}\,|\,\mathbf{z}_t,\mathbf{x})=N\left(\mathbf{z}_{t-1}\left|\frac{1-\alpha_{t-1}}{1-\alpha_t}\sqrt{1-\beta_t}\mathbf{z}_t+\frac{\beta_t\sqrt{\alpha_{t-1}}}{1-\alpha_t}\mathbf{x},\frac{\beta_t(1-\alpha_{t-1})}{1-\alpha_t}\mathbf{I}\right.\right),\quad \forall t=T,\cdots,1$$

- The expression simplifies to the squared difference between the means plus a constant C:

$$D_{KL}\left[q(\mathbf{z}_{t-1}\,|\,\mathbf{z}_t,\mathbf{x})\,\|\,p(\mathbf{z}_{t-1}\,|\,\mathbf{z}_t,\Phi_t)\right]=\frac{1}{2\sigma_t^2}\left\|\frac{1-\alpha_{t-1}}{1-\alpha_t}\sqrt{1-\beta_t}\mathbf{z}_t+\frac{\beta_t\sqrt{\alpha_{t-1}}}{1-\alpha_t}\mathbf{x}-f_t(\mathbf{z}_t,\Phi_t)\right\|^2+C$$

Diffusion loss function

- To fit the model, we maximize the ELBO with respect to the parameters $\boldsymbol{\phi}_{1\ldots T}$. We recast this as a minimization by multiplying with minus one and approximating the expectations with samples to give the loss function:

$$L[\boldsymbol{\phi}] = \sum_{i=1}^{I}\Bigg(\overbrace{-\log\Big[\text{Norm}_{\mathbf{x}_i}\left[\mathbf{f}_1[\mathbf{z}_{i1},\boldsymbol{\phi}_1],\sigma_1^2\mathbf{I}\right]\Big]}^{\text{reconstruction term}} + \sum_{t=2}^{T}\frac{1}{2\sigma_t^2}\Bigg\|\underbrace{\frac{(1-\alpha_{t-1})}{1-\alpha_t}\sqrt{1-\beta_t}\mathbf{z}_{it}+\frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t}\mathbf{x}_i}_{\text{target, mean of } q(\mathbf{z}_{t-1}|\mathbf{z}_t,\mathbf{x})} - \underbrace{\mathbf{f}_t[\mathbf{z}_{it},\boldsymbol{\phi}_t]}_{\text{predicted } \mathbf{z}_{t-1}}\Bigg\|^2\Bigg),$$

- $\mathbf{x}_i$:the i-th data point
- $\mathbf{z}_{it}$: the associated latent variable at diffusion step t.

Reparameterization of target

- The original diffusion update was given by:

$$\mathbf{z}_t = \sqrt{\alpha_t} \cdot \mathbf{x} + \sqrt{1-\alpha_t} \cdot \varepsilon$$

- **x** can be expressed as the diffused image minus the noise that was added to it:

$$\mathbf{x} = \frac{1}{\sqrt{\alpha_t}} \mathbf{z}_t - \frac{\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}} \cdot \varepsilon$$

- Substituting this into the target terms gives:

$$D_{KL}\left[q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x}) \,\|\, p(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \Phi_t)\right] = \frac{1}{2\sigma_t^2} \left\| \frac{1-\alpha_{t-1}}{1-\alpha_t} \sqrt{1-\beta_t}\, \mathbf{z}_t + \frac{\beta_t \sqrt{\alpha_{t-1}}}{1-\alpha_t} \mathbf{x} - f_t(\mathbf{z}_t, \Phi_t) \right\|^2 + C$$

$$= \frac{1}{2\sigma_t^2} \left\| \left(\frac{1}{\sqrt{1-\beta_t}} \mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}\sqrt{1-\beta_t}} \varepsilon \right) - f_t(\mathbf{z}_t, \Phi_t) \right\|^2 + C$$

Reparameterization of network

- The model $\hat{\mathbf{z}}_{t-1} = f_t(\mathbf{z}_t, \Phi_t)$ is replaced with a new model $\hat{\varepsilon}_{t-1} = g_t(\mathbf{z}_t, \Phi_t)$, which predicts the noise $\boldsymbol{\epsilon}$. Using the following transform

$$f_t(\mathbf{z}_t, \Phi_t) = \frac{1}{\sqrt{1-\beta_t}}\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}\sqrt{1-\beta_t}} g_t(\mathbf{z}_t, \Phi_t)$$

$$D_{KL}\left[q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x}) \,\|\, p(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \Phi_t)\right] = \frac{1}{2\sigma_t^2}\left\| \left(\frac{1}{\sqrt{1-\beta_t}}\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}\sqrt{1-\beta_t}}\varepsilon \right) - f_t(\mathbf{z}_t, \Phi_t) \right\|^2$$

$$= \frac{1}{2\sigma_t^2}\frac{\beta_t^2}{(1-\alpha_t)(1-\beta_t)}\left\| g_t(\mathbf{z}_t, \Phi_t) - \varepsilon \right\|^2$$

- The Cost function becomes

$$L(\Phi) = \sum_{k=1}^{N}\left[\frac{1}{2\sigma_t^2}\left\| \mathbf{x}_k - f_1(\mathbf{z}_{1k}, \Phi_1) \right\|^2 + \sum_{t=2}^{T} \frac{1}{2\sigma_t^2}\frac{\beta_t^2}{(1-\alpha_t)(1-\beta_t)}\left\| g_t(\mathbf{z}_{tk}, \Phi_t) - \varepsilon_{it} \right\|^2 \right]$$

Reparameterization of network

- Using the following transform

$$f_t(\mathbf{z}_t, \Phi_t) = \frac{1}{\sqrt{1-\beta_t}}\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}\sqrt{1-\beta_t}} g_t(\mathbf{z}_t, \Phi_t)$$

$$\mathbf{z}_t = \sqrt{\alpha_t}\cdot\mathbf{x} + \sqrt{1-\alpha_t}\cdot\boldsymbol{\varepsilon}, \qquad \alpha_t = \prod_{k=1}^{t}(1-\beta_k)$$

- We have

$$\frac{1}{2\sigma_1^2}\left\|\mathbf{x}_k - f_1(\mathbf{z}_{1k}, \Phi_1)\right\|^2 = \frac{1}{2\sigma_1^2}\left\|\frac{\beta_1}{\sqrt{(1-\alpha_1)(1-\beta_1)}} g_1(\mathbf{z}_{1k}, \Phi_1) - \frac{\beta_1}{\sqrt{(1-\alpha_1)(1-\beta_1)}}\varepsilon_{1k}\right\|^2$$

- The Cost function becomes

$$L(\Phi) = \sum_{k=1}^{N}\left[\sum_{t=1}^{T} \frac{1}{2\sigma_t^2}\frac{\beta_t^2}{(1-\alpha_t)(1-\beta_t)}\left\|g_t(\mathbf{z}_{tk}, \Phi_t) - \varepsilon_{it}\right\|^2\right]$$

Reparameterization of network

- The cost function

$$L(\Phi)=\sum_{k=1}^{N}\left[\sum_{t=1}^{T}\frac{1}{2\sigma_t^2}\frac{\beta_t^2}{(1-\alpha_t)(1-\beta_t)}\left\|g_t(\mathbf{z}_{tk},\Phi_t)-\varepsilon_{it}\right\|^2\right]$$

- In practice, the weight factors (which might be different at each time step) are ignored, giving an even simpler formulation:

$$L(\Phi)=\sum_{k=1}^{N}\left[\sum_{t=1}^{T}\left\|g_t(\mathbf{z}_{tk},\Phi_t)-\varepsilon_{it}\right\|^2\right]$$

$$=\sum_{k=1}^{N}\left[\sum_{t=1}^{T}\left\|g_t(\sqrt{\alpha_t}\cdot\mathbf{x}_k+\sqrt{1-\alpha_t}\cdot\varepsilon_{kt},\Phi_t)-\varepsilon_{kt}\right\|^2\right]$$

Implementation

Algorithm 18.1: Diffusion model training

Input: Training data $\mathbf{x}$

Output: Model parameters $\boldsymbol{\phi}_t$

repeat

  for $i \in \mathcal{B}$ **do** // For every training example index in batch

    $t \sim \text{Uniform}[1, \ldots T]$ // Sample random timestep

    $\boldsymbol{\epsilon} \sim \text{Norm}[\mathbf{0}, \mathbf{I}]$ // Sample noise

    $\ell_i = \left\| \mathbf{g}_t\left[\sqrt{\alpha_t}\mathbf{x}_i + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}, \boldsymbol{\phi}_t\right] - \boldsymbol{\epsilon} \right\|^2$ // Compute individual loss

  Accumulate losses for batch and take gradient step

until converged

Implementation

Algorithm 18.2: Sampling

Input: Model, $\mathbf{g}_t[\bullet, \boldsymbol{\phi}_t]$

Output: Sample, $\mathbf{x}$

$\mathbf{z}_T \sim \text{Norm}_{\mathbf{z}}[\mathbf{0}, \mathbf{I}]$ // Sample last latent variable

for $t = T \ldots 2$ **do**

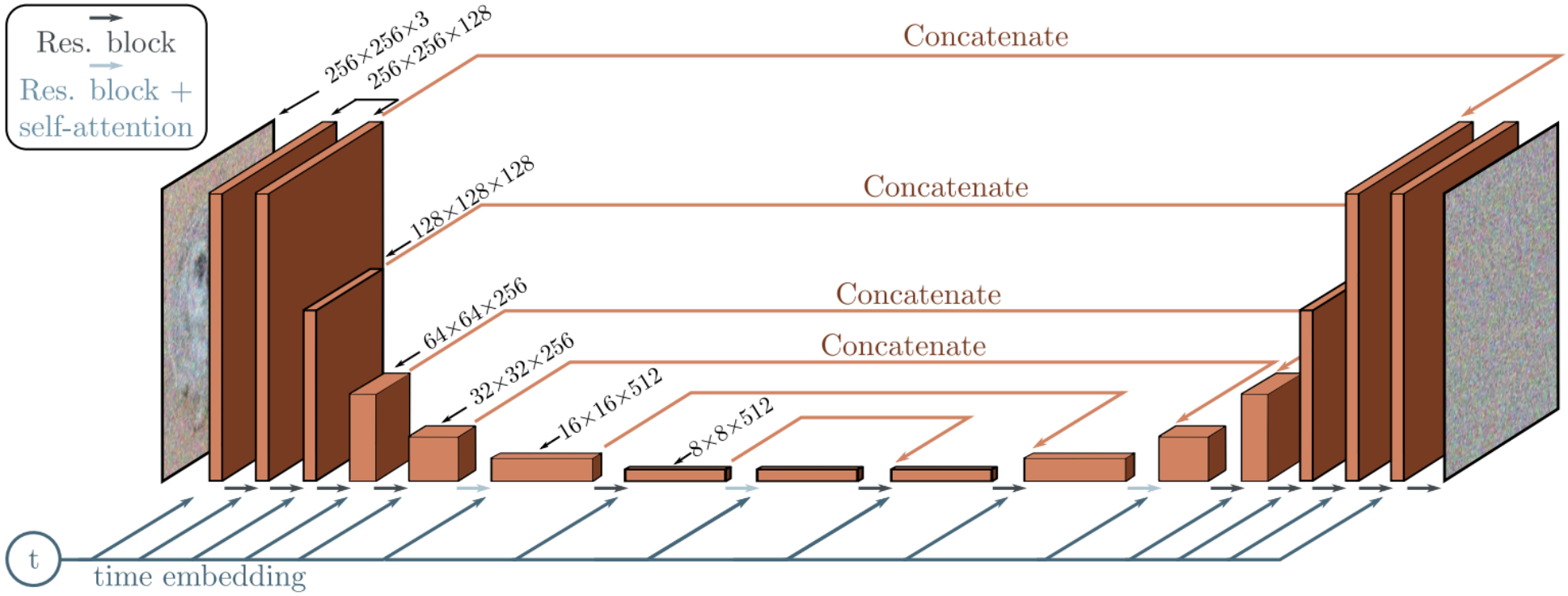
$\hat{\mathbf{z}}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}\sqrt{1-\beta_t}}\mathbf{g}_t[\mathbf{z}_t, \boldsymbol{\phi}_t]$ // Predict previous latent variable

$\boldsymbol{\epsilon} \sim \text{Norm}_{\boldsymbol{\epsilon}}[\mathbf{0}, \mathbf{I}]$ // Draw new noise vector

$\mathbf{z}_{t-1} = \hat{\mathbf{z}}_{t-1} + \sigma_t\boldsymbol{\epsilon}$ // Add noise to previous latent variable

$\mathbf{x} = \frac{1}{\sqrt{1-\beta_1}}\mathbf{z}_1 - \frac{\beta_1}{\sqrt{1-\alpha_1}\sqrt{1-\beta_1}}\mathbf{g}_1[\mathbf{z}_1, \boldsymbol{\phi}_1]$ // Generate sample from $\mathbf{z}_1$ without noise

Application to images



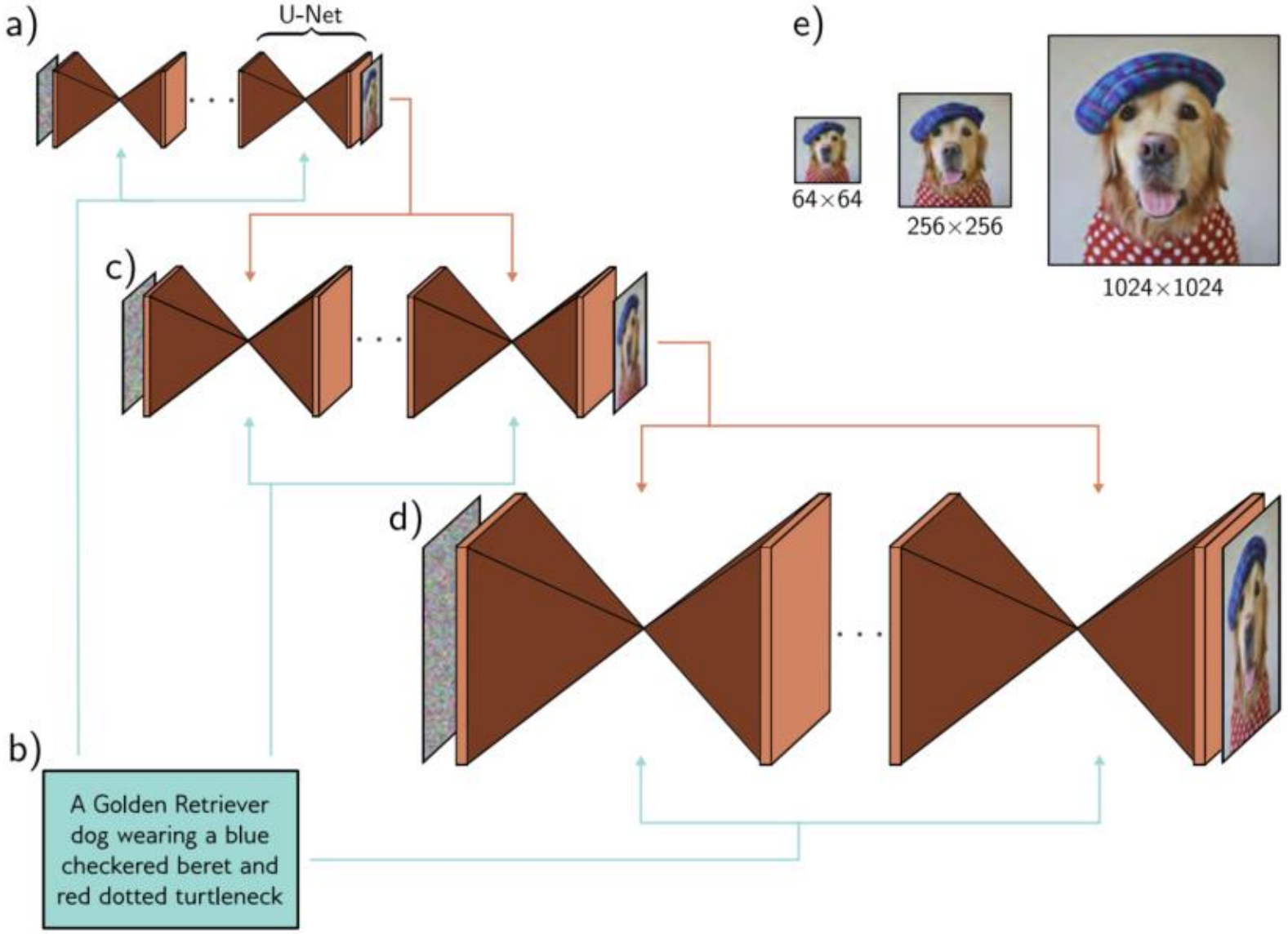
- **Figure 18.9** U-Net as used in diffusion models for images
  - Encoder: reduces the scale and increases the number of channels
  - Decoder: increases the scale and reduces the number of channels
  - Network: residual blocks + global self-attention

Conditional generation

- If the data has associated labels c, these can be exploited to control the generation. Classifier guidance modifies the denoising update from $\mathbf{z}_t$ to $\mathbf{z}_{t-1}$ to take into account class information c. In practice, this means adding an extra term into the final update step to yield:

$$\mathbf{z}_{t-1} = \hat{\mathbf{z}}_{t-1} + \sigma_t^2 \frac{\partial \log[p(c \mid \mathbf{z}_t)]}{\partial \mathbf{z}_t} + \sigma_t^2 \varepsilon$$

- The new term depends on the gradient of a classifier $p(c \mid \mathbf{z}_t)$ that is based on the latent variable $\mathbf{z}_t$. This maps features from the downsampling half of the U-Net to the class c. Like the U-Net, it is usually shared across all time steps and takes time as an input. The update from $\mathbf{z}_t$ to $\mathbf{z}_{t-1}$ now makes the class c more likely.



Cascaded conditional generation based on a text prompt.

- a) A diffusion model consisting of a series of U-Nets is used to generate a 64×64 image.
- b) This generation is conditioned on a sentence embedding computed by a language model.
- c) A higher resolution 256×256 image is generated and conditioned on the smaller image and the text encoding.
- d) This is repeated to create a 1024×1024 image.
- e) Final image sequence. Adapted from Saharia et al (2022b).

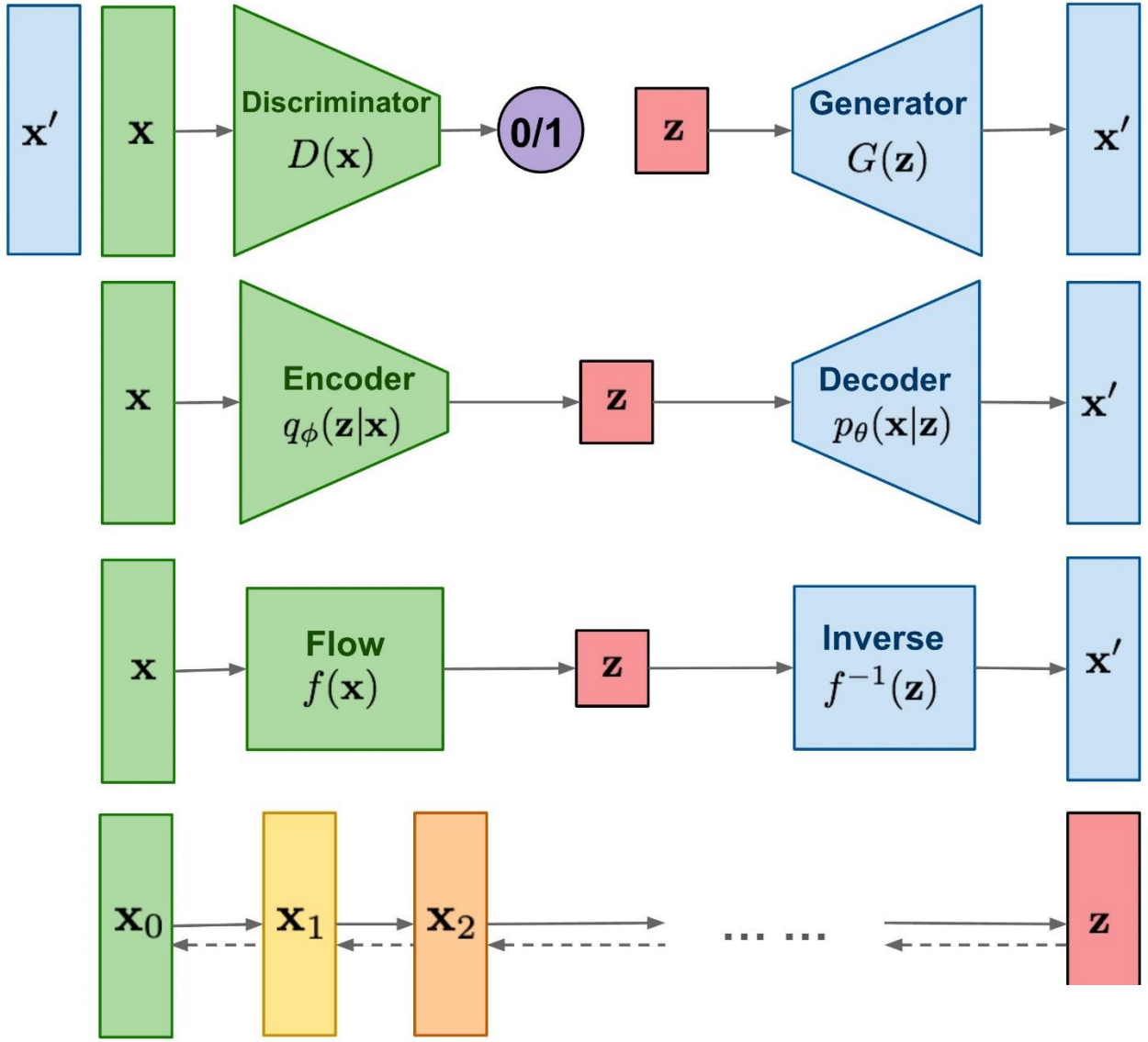
Comparisons

GAN: Adversarial training

VAE: maximize variational lower bound

Flow-based models: Invertible transform of distributions

Diffusion models: Gradually add Gaussian noise and then reverse



Summary

- **Basic Mechanism:**
  - Data examples are mapped through latent variables, blending with random noise.
  - Eventually, representation resembles white noise.
- **Reverse Process:**
  - Denoising approximated by a normal distribution.
  - Predicted via deep learning models.
- **Loss Function:**
  - Based on evidence lower bound (ELBO).
  - Leads to a least-squares formulation.
- **Implementation:**
  - U-Net used for denoising steps.
  - Sampling slower than other generative models.
- **Optimizations and Conditioning:**
  - Conditioning on class, images, and text enhances text-to-image synthesis.