CS/DS 552: Class 10

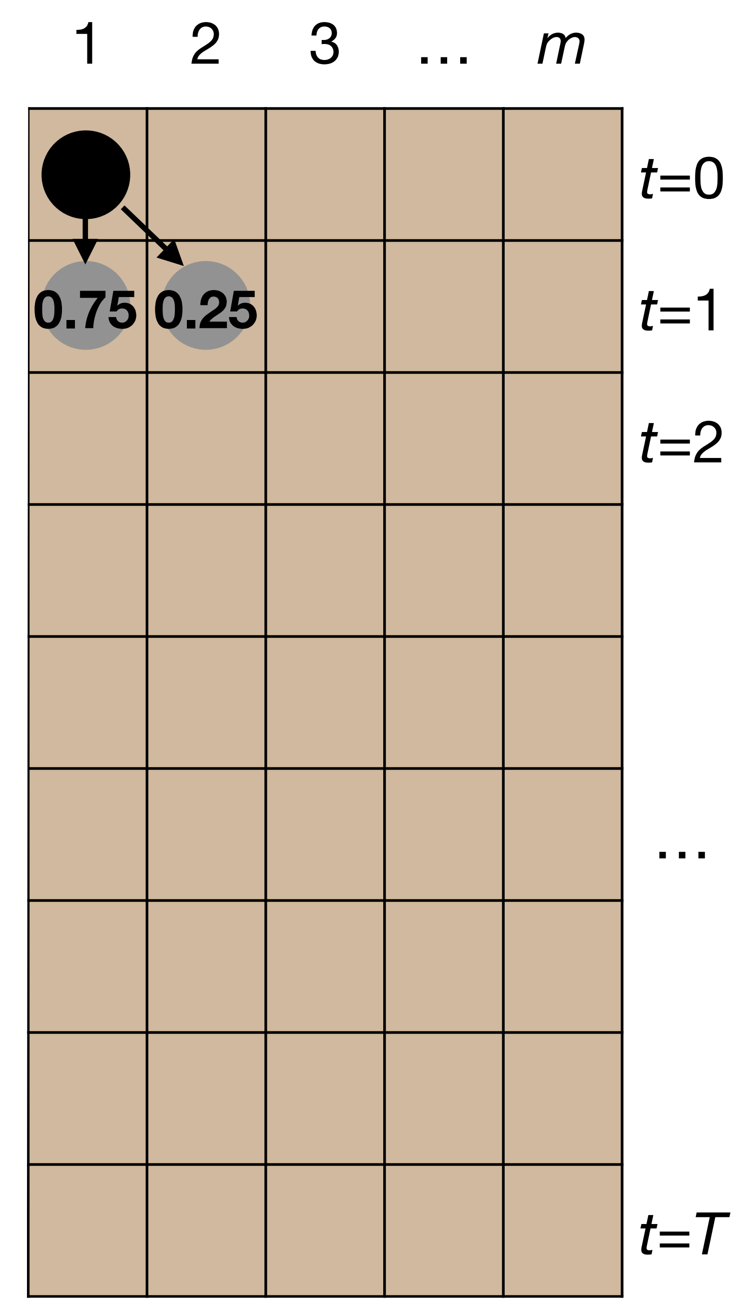
Jacob Whitehill

Discrete diffusion: Plinko

- Consider a game where you place a disc at position $x \in \{1,\ldots,m\}$ at row t=0.
- Let $z_0 = x$.
- At each timestep, the disc falls from row t to row t+1, and its position z_t changes probabilistically to z_{t+1}:

$$Q(z_{t+1} \mid z_t) = \begin{cases} 0.25 & \text{if} \quad z_{t+1} = z_t - 1 \\ 0.5 & \text{if} \quad z_{t+1} = z_t \ \ and\ 1 < z_t < m \\ 0.75 & \text{if} \quad z_{t+1} = z_t \ \ and\ (z_t = 1 \ \ or\ z_t = m) \\ 0.25 & \text{if} \quad z_{t+1} = z_t + 1 \end{cases}$$

(This updated equation includes edge effects.)



Discrete diffusion: Markov process

- We can model the sequence $\{x, z_1, \ldots z_T\}$ as a Markov chain.

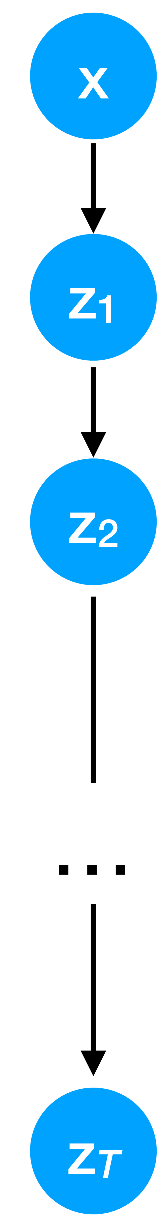
- Assume there is some distribution *P*(*x*) for where we drop the disc at *t*=0.

- We have conditional independence:

$$Q(z_t \mid z_0, \ldots, z_{t-1}) = Q(z_t \mid z_{t-1}) \ \forall t$$

(where we define $z_0 = x$).

$$Q(z_{t+1} \mid z_t) = \begin{cases} 0.25 & \text{if} \quad z_{t+1} = z_t - 1 \\ 0.5 & \text{if} \quad z_{t+1} = z_t \ and \ 1 < z_t < m \\ 0.75 & \text{if} \quad z_{t+1} = z_t \ and \ (z_t = 1 \ or \ z_t = m) \\ 0.25 & \text{if} \quad z_{t+1} = z_t + 1 \end{cases}$$



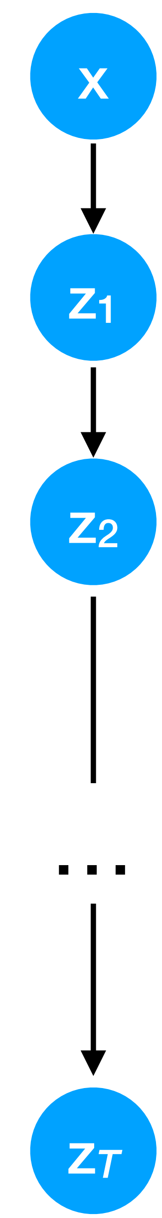
Discrete diffusion: Markov process

- Suppose we set $P(x)$ = Unif[1, …, m].
- For $P(z_{t+1} \mid z_t)$ shown below, what will be the final distribution $P(z_T)$?

  A. Disc will probably be in the middle.

  B. Disc will probably be along either side.

  C. Disc will be anywhere with equal probability.



$$Q(z_{t+1} \mid z_t) = \begin{cases} 0.25 & \text{if} \quad z_{t+1} = z_t - 1 \\ 0.5 & \text{if} \quad z_{t+1} = z_t \;\; and \; 1 < z_t < m \\ 0.75 & \text{if} \quad z_{t+1} = z_t \;\; and \; (z_t = 1 \;\; or \; z_t = m) \\ 0.25 & \text{if} \quad z_{t+1} = z_t + 1 \end{cases}$$

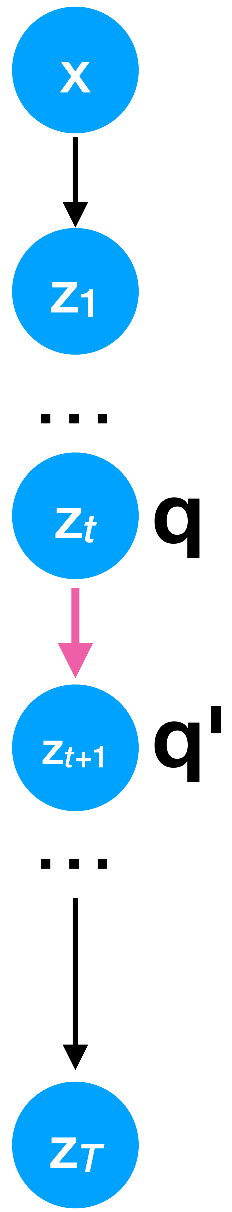
Discrete diffusion: Markov process

- Let **q**=$Q(z_t)$ be the probability distribution over the disc's position in row t.

- We can calculate **q'**=$Q(z_{t+1})$ of the disc's next position as follows:

$$\mathbf{q}' = \mathbf{D}\mathbf{q}$$

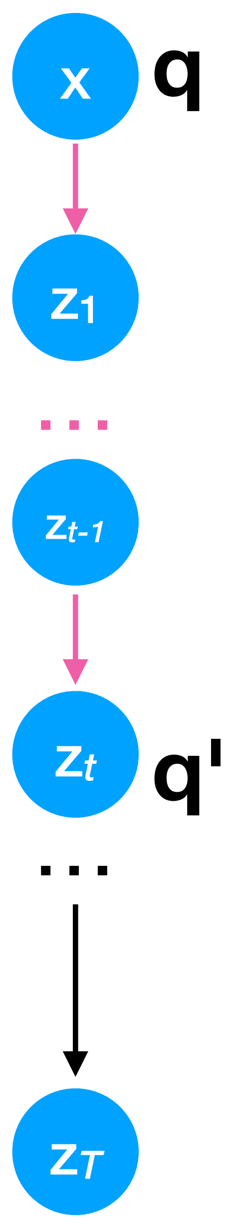
$$= \begin{bmatrix} 0.75 & 0.25 & 0 & \dots & 0 \\ 0.25 & 0.5 & 0.25 & \dots & 0 \\ 0 & 0.25 & 0.5 & \dots & 0 \\ & & \vdots & & \\ 0 & \dots & 0.25 & 0.5 & 0.25 \\ 0 & \dots & 0 & 0.25 & 0.75 \end{bmatrix} \mathbf{q}$$



Discrete diffusion: Markov process

- For this particular **D** (and others), we can compute $Q(z_t \mid x)$ for **any** t as:

$$\begin{aligned} \mathbf{q}' &= \mathbf{D}^t \mathbf{q} \\ &= (\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top)^t \mathbf{q} \\ &= \mathbf{U}\boldsymbol{\Lambda}^t \mathbf{U}^\top \mathbf{q} \end{aligned}$$



- Hence, computing **q'** is very fast for any t.

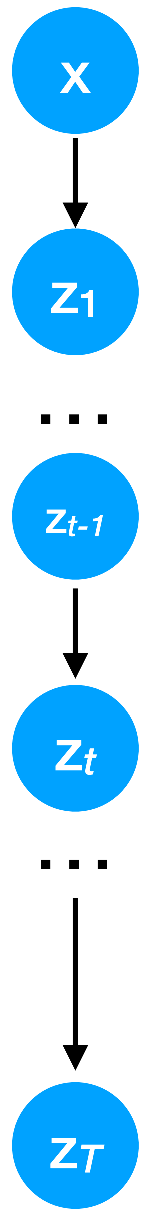
- Here, $\mathbf{D}^t$ is called the **diffusion kernel** and determines the state distribution at any time t.

Discrete diffusion: sampling

- Hence, if we want to sample from $Q(z_t \mid x)$ for any t, we can use either of 2 strategies:

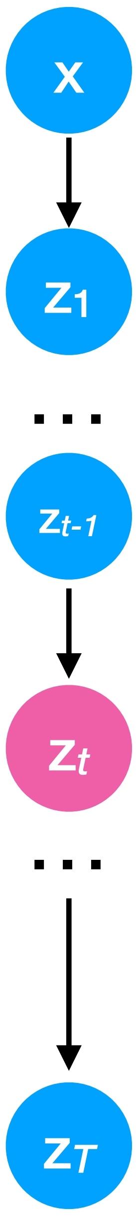
  - **Ancestral sampling**:

    1. Set $z_0 = x$.
    2. Sample $z_1 \sim Q(z_1 \mid z_0)$.
    3. …
    4. Sample $z_t \sim Q(z_t \mid z_{t-1})$.



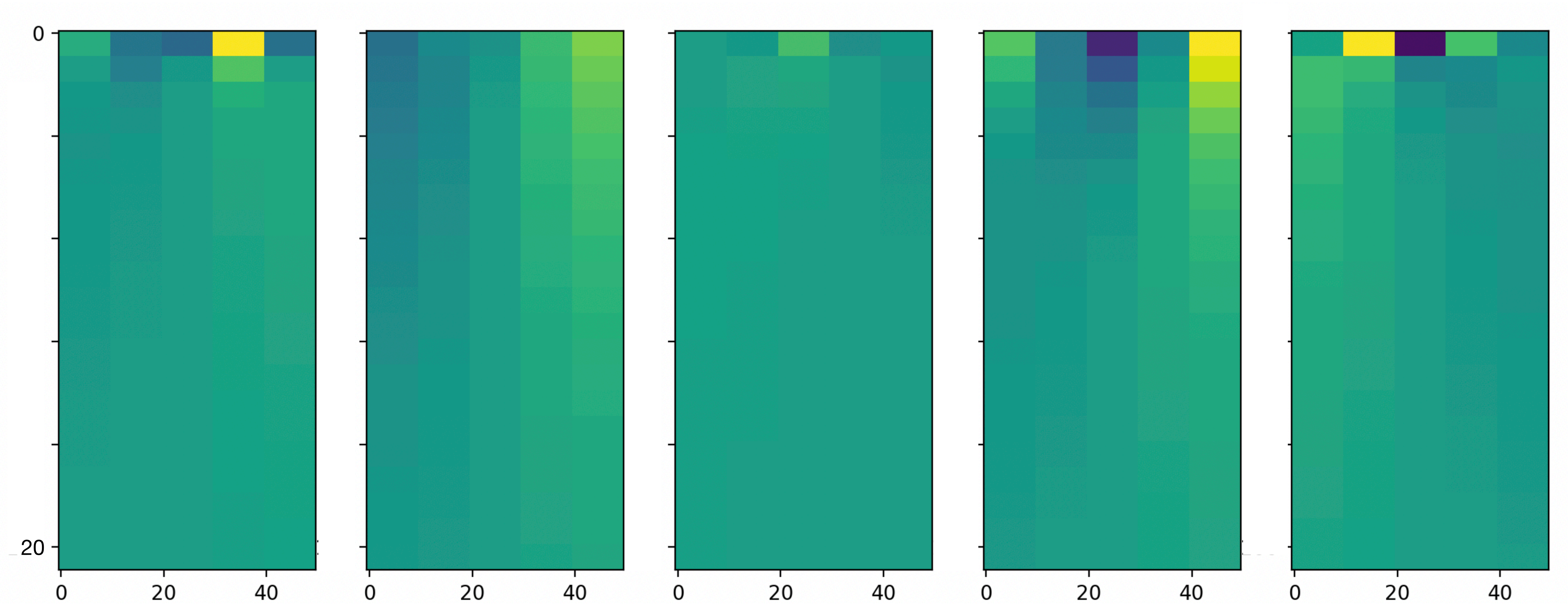
Discrete diffusion: sampling

- Hence, if we want to sample from $Q(z_t \mid x)$ for any t, we can use either of 2 strategies:
  - **Diffusion kernel**:
    1. Compute $Q(z_t \mid x)$ using the diffusion kernel.
    2. Sample $z_t \sim Q(z_t \mid x)$.



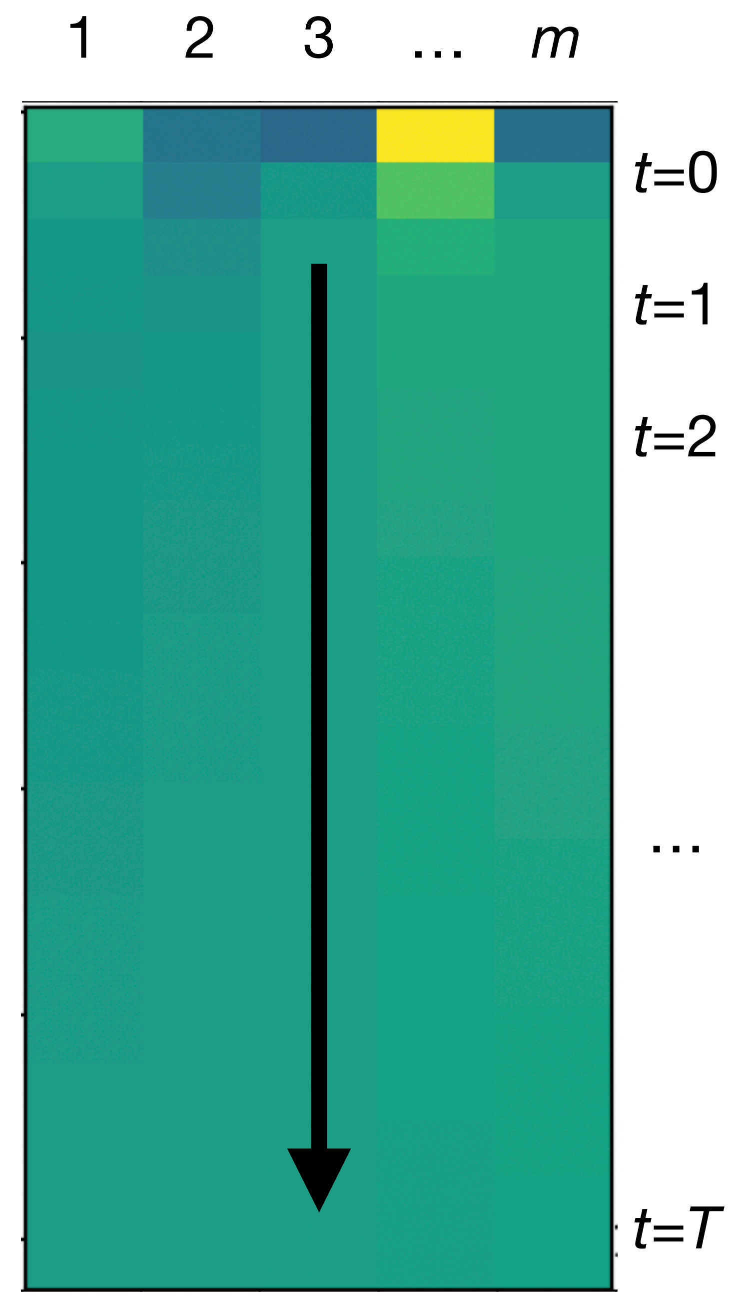
Discrete diffusion: Forward process

- Examples for 5 different starting distributions $P(x)$:



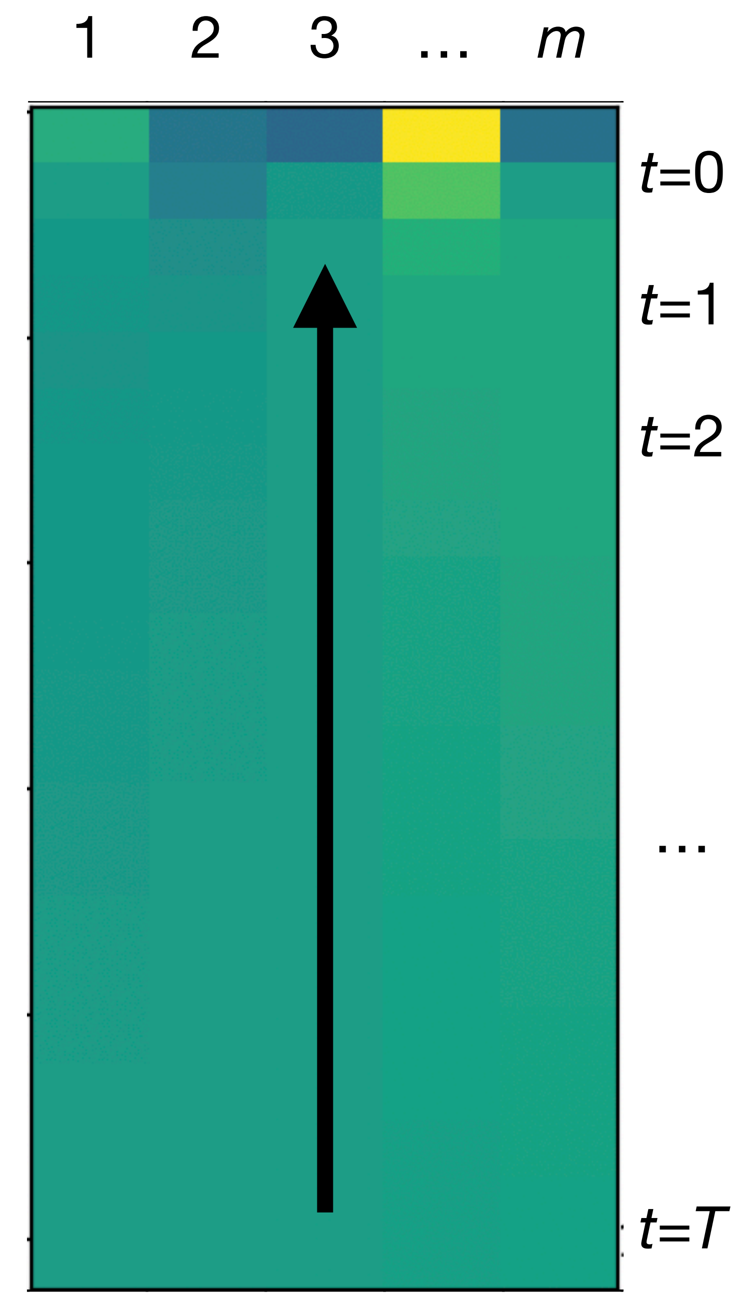
Discrete diffusion: Forward process

- In other words, **the Plinko diffusion converts any arbitrary distribution *P*(*x*) into Unif[1, …, *m*]**:
  - $z_0 = x \sim P(x)$
    $z_1 \sim Q(z_1 \mid z_0)$
    $z_2 \sim Q(z_2 \mid z_1)$
    …
    $z_T \sim Q(z_T \mid z_{T\text{-}1}) \approx Q(z_T) = \text{Unif}[1, \ldots, m]$



Discrete diffusion: Reverse process

- **Can we play the game in reverse** (Oknilp) **and convert Unif[1, …, *m*] into any arbitrary distribution *P*(*x*)?**
  - $z_T \sim Q(z_T) = \text{Unif}[1, \ldots, m]$
    $z_{T-1} \sim Q(z_{T-1} \mid z_T)$
    …
    $z_2 \sim Q(z_2 \mid z_3)$
    $z_1 \sim Q(z_1 \mid z_2)$
    $x \sim Q(z_0 \mid z_1)$
- Yes, as long as we know all the conditional distributions $Q(z_{t-1} \mid z_t)$.



Discrete diffusion: Reverse process

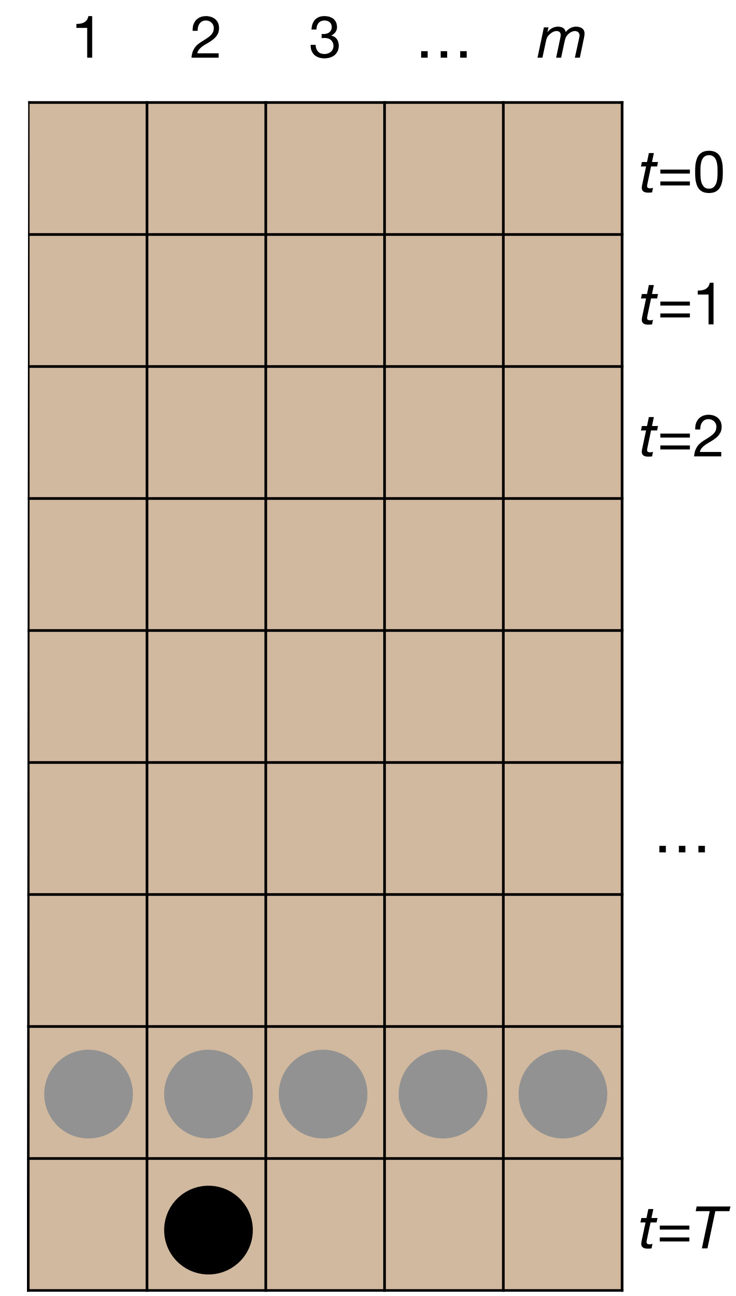
- To compute $Q(z_{t-1} \mid z_t)$, we can apply Bayes' rule:

$$Q(z_{t-1}|z_t) = \frac{Q(z_t \mid z_{t-1})Q(z_{t-1})}{Q(z_t)}$$

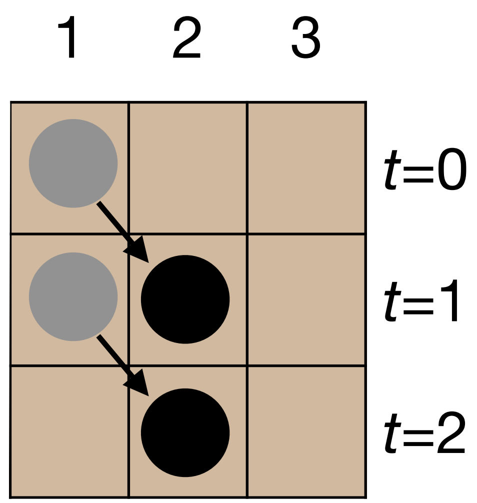
$$\propto Q(z_t \mid z_{t-1})Q(z_{t-1})$$

- I.e., multiply probability of transitioning from z_{t-1} to z_t, by probability of being in state z_{t-1}.

$$Q(z_{t+1} \mid z_t) = \begin{cases} 0.25 & \text{if} \quad z_{t+1} = z_t - 1 \\ 0.5 & \text{if} \quad z_{t+1} = z_t \ \textit{and}\ 1 < z_t < m \\ 0.75 & \text{if} \quad z_{t+1} = z_t \ \textit{and}\ (z_t = 1 \ \textit{or}\ z_t = m) \\ 0.25 & \text{if} \quad z_{t+1} = z_t + 1 \end{cases}$$



Solution



- Suppose the initial state distribution $P(x)=[0.8\ 0.1\ 0.1]^T$.
- Let $Q(z_t \mid z_{t-1})$ be as before:

$$Q(z_{t+1} \mid z_t) = \begin{cases} 0.25 & \text{if} \quad z_{t+1} = z_t - 1 \\ 0.5 & \text{if} \quad z_{t+1} = z_t \ and\ 1 < z_t < m \\ 0.75 & \text{if} \quad z_{t+1} = z_t \ and\ (z_t = 1 \ or\ z_t = m) \\ 0.25 & \text{if} \quad z_{t+1} = z_t + 1 \end{cases}$$

- $Q(Z_0=1 \mid Z_1=2) > Q(Z_1=1 \mid Z_2=2)$ because the disc **very likely started out (*t*=0) in slot 1**, whereas by $t=1$, the disc's likely position had "diffused" across slots.

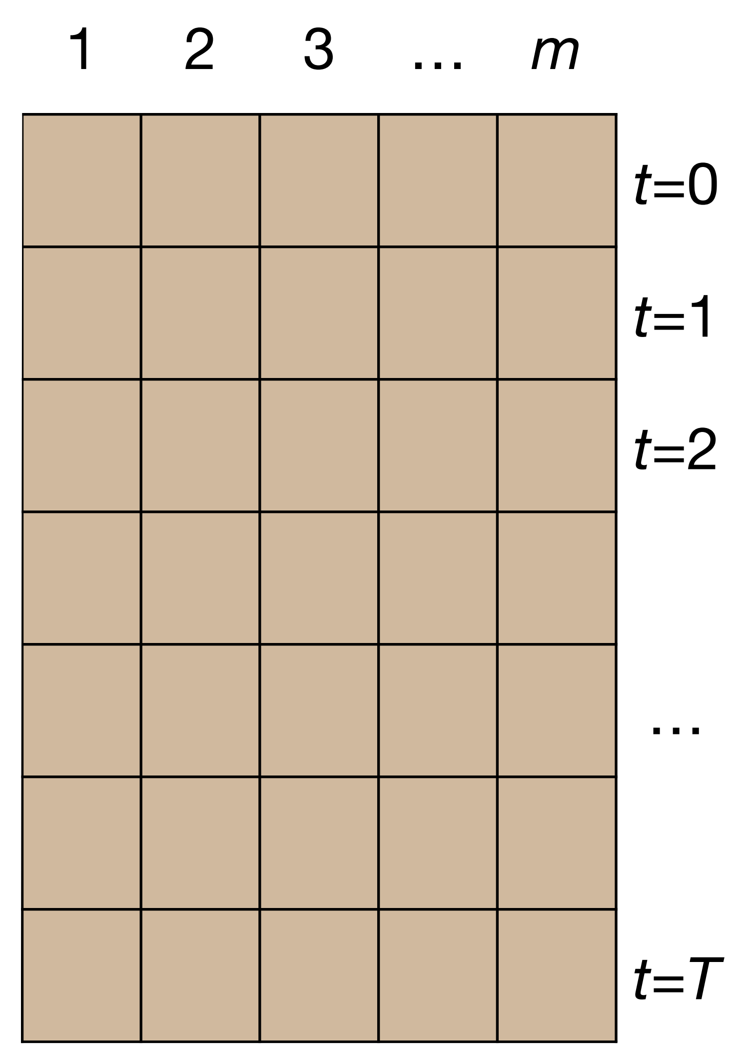
$$\begin{aligned} Q(z_{t-1}|z_t) &= \frac{Q(z_t \mid z_{t-1})Q(z_{t-1})}{Q(z_t)} \\ &\propto Q(z_t \mid z_{t-1})Q(z_{t-1}) \end{aligned}$$

Exercise

$$\mathbf{D} = \begin{bmatrix} 0.75 & 0.25 & 0 & \dots & 0 \\ 0.25 & 0.5 & 0.25 & \dots & 0 \\ 0 & 0.25 & 0.5 & \dots & 0 \\ & & \vdots & & \\ 0 & \dots & 0.25 & 0.5 & 0.25 \\ 0 & \dots & 0 & 0.25 & 0.75 \end{bmatrix}$$

- Find and correct the mistakes in the code below, which is designed to reverse-sample 1000 trajectories over *T* timesteps.

```
D = ... # as defined above
samples = []
for _ in range(1000):
    z = np.random.choice(range(M), p=M*[1./M])
    for t in range(T, 0, -1):
        p_prev = Qz[t] * D[:,z]
        p_prev /= p_prev.max()
        z = np.random.choice(range(M), p=M*[1./M])
    samples.append(z)
```



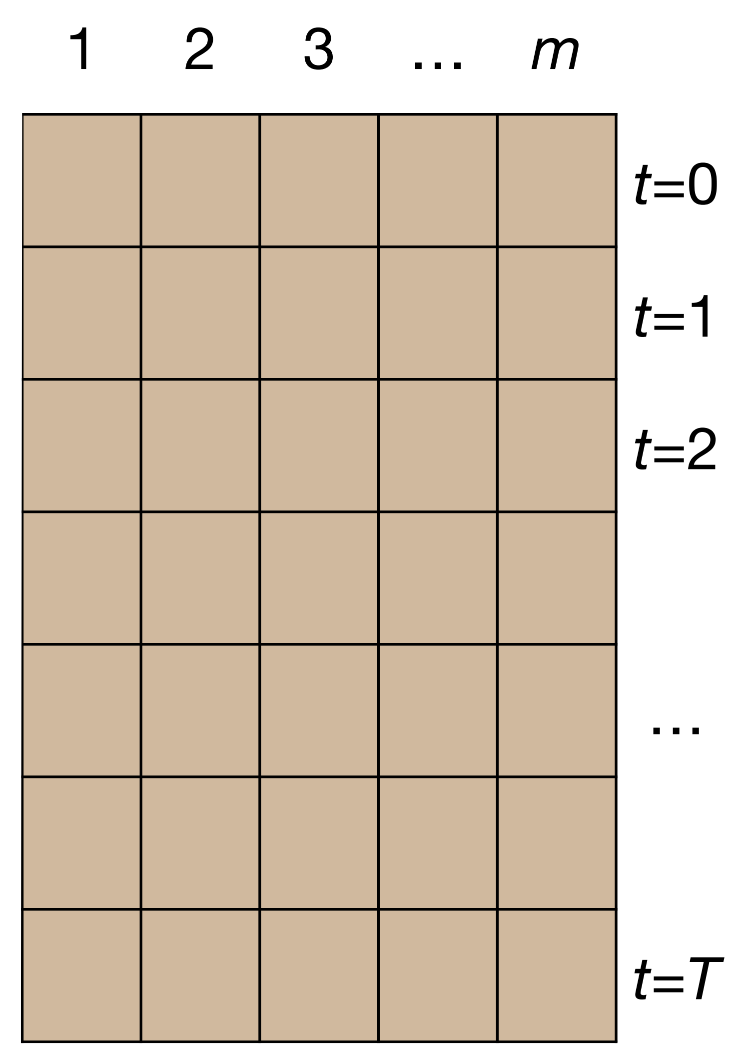
$$Q(z_{t-1}|z_t) = \frac{Q(z_t \mid z_{t-1})Q(z_{t-1})}{Q(z_t)}$$

$$\propto Q(z_t \mid z_{t-1})Q(z_{t-1})$$

Solution

$$
\mathbf{D} = \begin{array}{cc} & z_{t-1} \\ z_t & \begin{bmatrix} 0.75 & 0.25 & 0 & \dots & 0 \\ 0.25 & 0.5 & 0.25 & \dots & 0 \\ 0 & 0.25 & 0.5 & \dots & 0 \\ & & \vdots & & \\ 0 & \dots & 0.25 & 0.5 & 0.25 \\ 0 & \dots & 0 & 0.25 & 0.75 \end{bmatrix} \end{array}
$$

- Find and correct the mistakes in the code below, which is designed to reverse-sample 1000 trajectories over *T* timesteps.




```
D = ... # as defined above
samples = []
for _ in range(1000):
    z = np.random.choice(range(M), p=M*[1./M])
    for t in range(T, 0, -1):
        p_prev = Qz[t-1] * D[z,:]
        p_prev /= p_prev.sum()
        z = np.random.choice(range(M), p=p_prev)
    samples.append(z)
```

$$
Q(z_{t-1}|z_t) = \frac{Q(z_t \mid z_{t-1})Q(z_{t-1})}{Q(z_t)}
$$

$$
\propto Q(z_t \mid z_{t-1})Q(z_{t-1})
$$

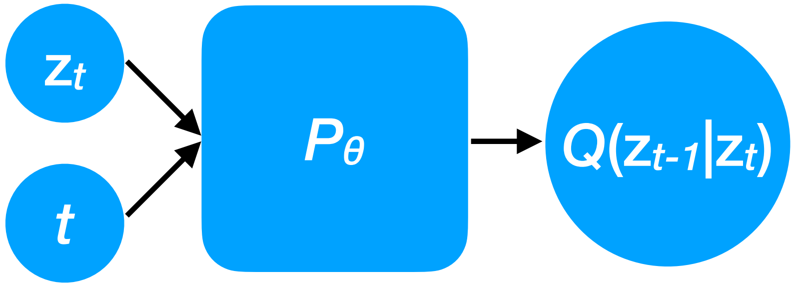
For **fixed** z_t, consider all possible z_{t-1}.

Function approximation

- The previous exercise was unrealistic:
  - If we could compute $Q(z_t)$ for each t, then we could just directly sample $Q(z_0)$ and be done!
- What if $\mathbf{x}$ were high-dimensional and/or continuous?
  - Estimating and storing $Q(z_t)$ as a "look-up table" would be intractable.
- We need a more generalizable approach…

Function approximation

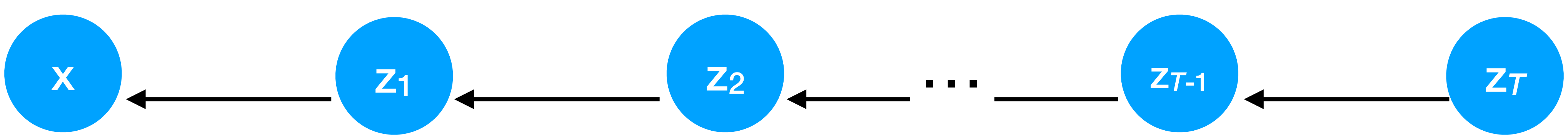
- We will use a NN P_θ to approximate $Q(\mathbf{z}_{t\text{-}1} \mid \mathbf{z}_t)$ for each t.
- Rather than train one NN per timestep, we will train a *single* NN that accepts a timestep parameter t as additional input:



- The exact formats of t (e.g., positional encoding) and $Q(\mathbf{z}_{t\text{-}1} \mid \mathbf{z}_t)$ (i.e., sufficient statistics) depend on the particular application.

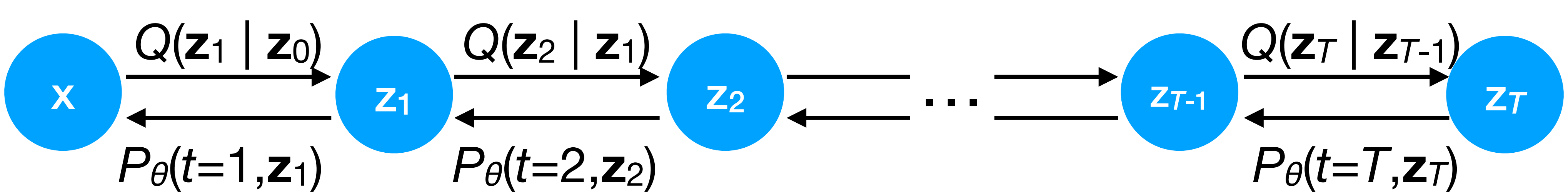
Diffusion as a latent variable model (LVM)

- Like VAEs, diffusions are another kind of LVM.



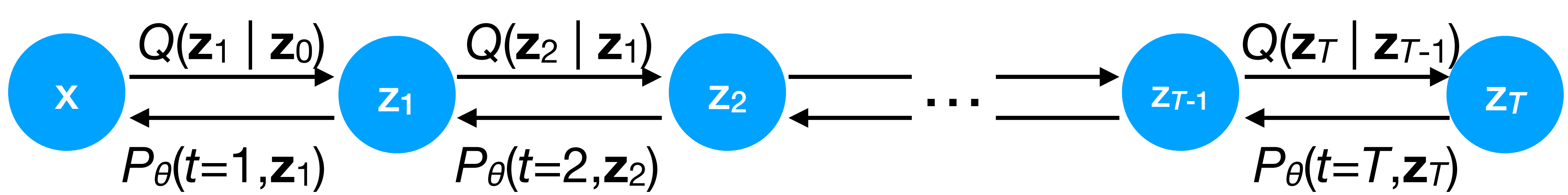
Diffusion as a latent variable model (LVM)

- Like VAEs, diffusions are another kind of LVM.
- Like VAEs, there is a decoder P_θ that maps from $\mathbf{z}_t$ to $\mathbf{z}_{t\text{-}1}$.



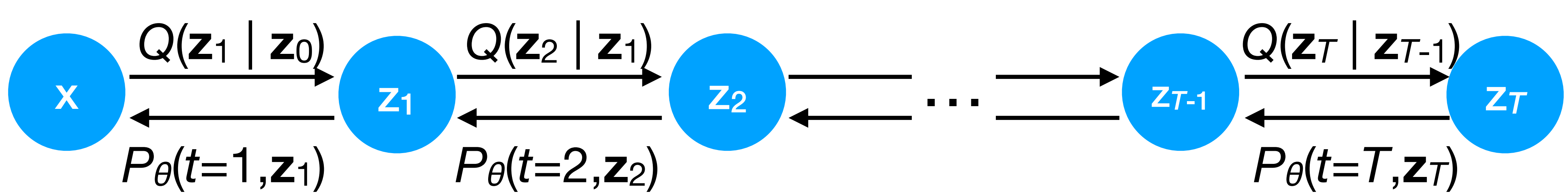
Diffusion as a latent variable model (LVM)

- Like VAEs, diffusions are another kind of LVM.
- Like VAEs, there is a decoder P_θ that maps from $\mathbf{z}_t$ to $\mathbf{z}_{t\text{-}1}$.
- Unlike VAEs, the encoder Q in a diffusion is **fixed** and has **no trainable parameters**.



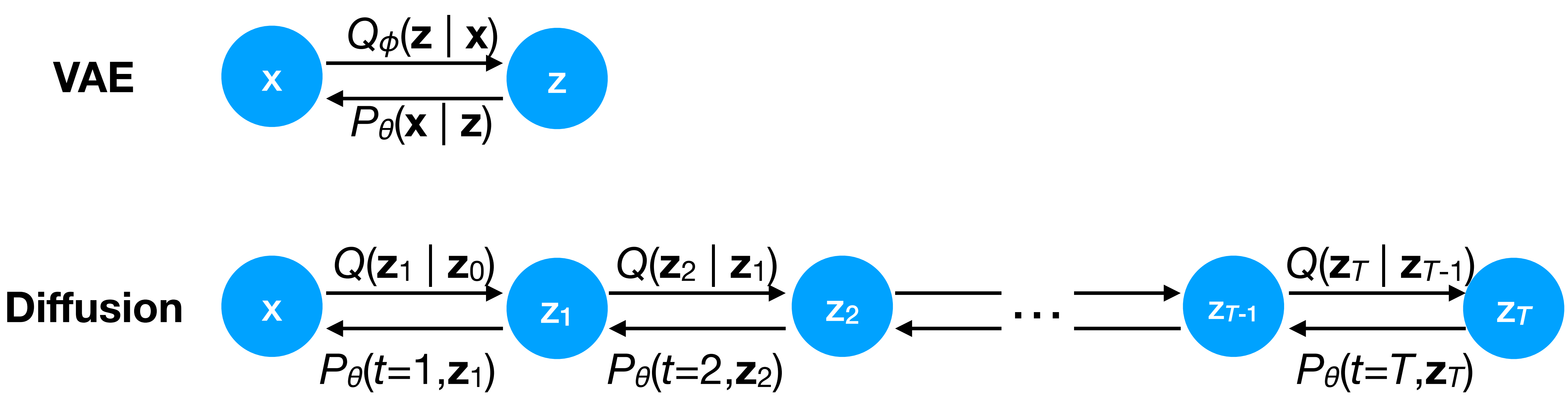
Diffusion as a latent variable model (LVM)

- Like VAEs, diffusions are another kind of LVM.
- Like VAEs, there is a decoder P_θ that maps from $\mathbf{z}_t$ to $\mathbf{z}_{t\text{-}1}$.
- Unlike VAEs, the encoder Q in a diffusion is **fixed** and has **no trainable parameters**.
- Unlike VAEs, there is an **entire sequence** of T latent variables.



VAE v. Diffusion

- Is information about **x** encoded in:
  1. Latent variable **z** in a VAE? **Yes**: from **z** we can approximately reconstruct **x** using decoder $P_\theta(\mathbf{x} \mid \mathbf{z})$.
  2. Latent variable $\mathbf{z}_T$ in a diffusion? **No**: we can sample from $P(\mathbf{x})$ starting from $\mathbf{z}_T$ but not reconstruct a specific **x**.



(Continuous-state) diffusions

Diffusions

- In the 1-d continuous-space diffusion, the position z_t of the disc at time t can be any real number.

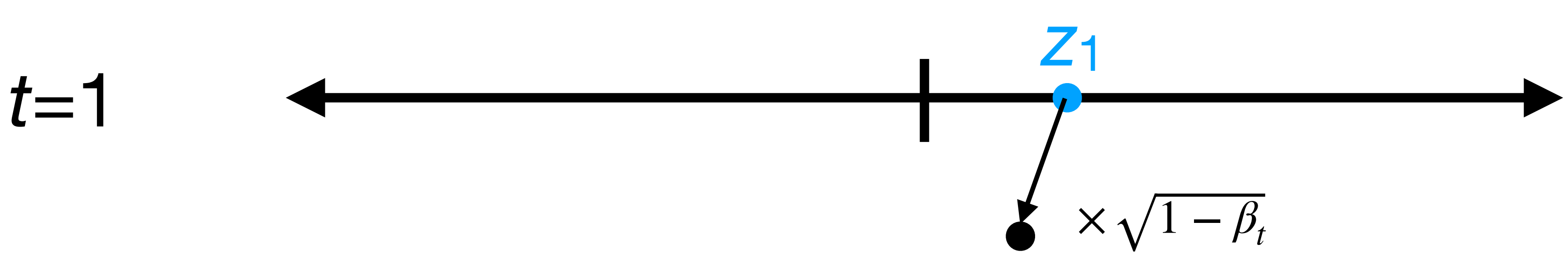
Diffusions

- In the 1-d continuous-space diffusion, the position z_t of the disc at time t can be any real number.
- The disc moves from position z_t at time t=1 to position z_2 at time t=2 probabilistically by:

Diffusions

- In the 1-d continuous-space diffusion, the position z_t of the disc at time t can be any real number.
- The disc moves from position z_t at time t=1 to position z_2 at time t=2 probabilistically by:
  1. Multiplying it by $\sqrt{1-\beta_t} < 1$.

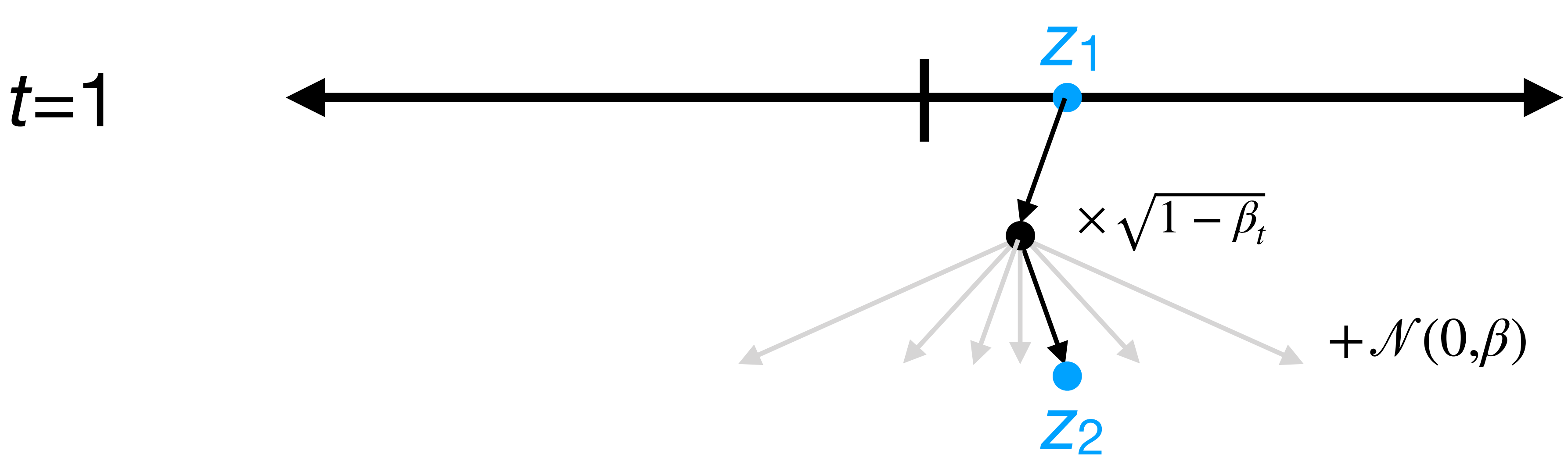
$$z_{t+1} = \sqrt{1-\beta_t} z_t$$



Diffusions

- In the 1-d continuous-space diffusion, the position z_t of the disc at time t can be any real number.
- The disc moves from position z_t at time t=1 to position z_2 at time t=2 probabilistically by:
  1. Multiplying it by $\sqrt{1-\beta_t} < 1$.
  2. Adding Gaussian noise.

$$z_{t+1} = \sqrt{1-\beta_t} z_t + \sqrt{\beta_t} \epsilon \text{ where } \epsilon \sim \mathcal{N}(0,1)$$

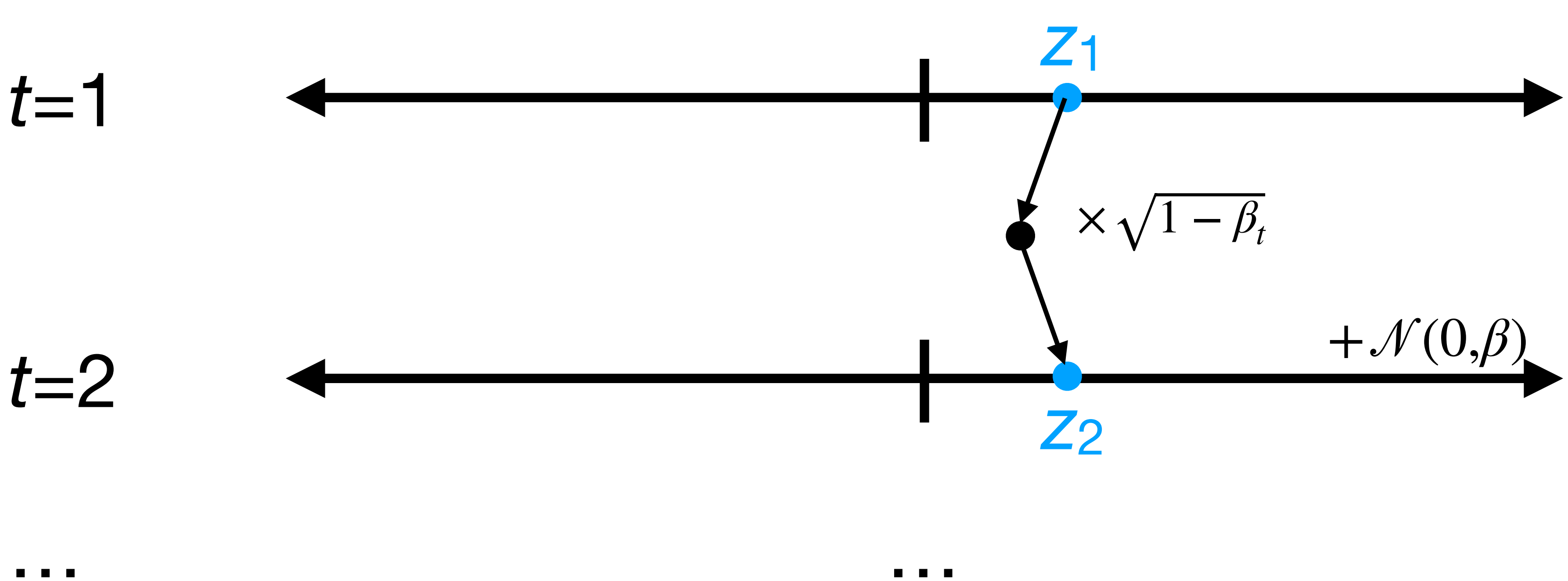


Diffusions

- In the 1-d continuous-space diffusion, the position z_t of the disc at time t can be any real number.
- The disc moves from position z_t at time t=1 to position z_2 at time t=2 probabilistically by:
  1. Multiplying it by $\sqrt{1-\beta_t} < 1$.
  2. Adding Gaussian noise.

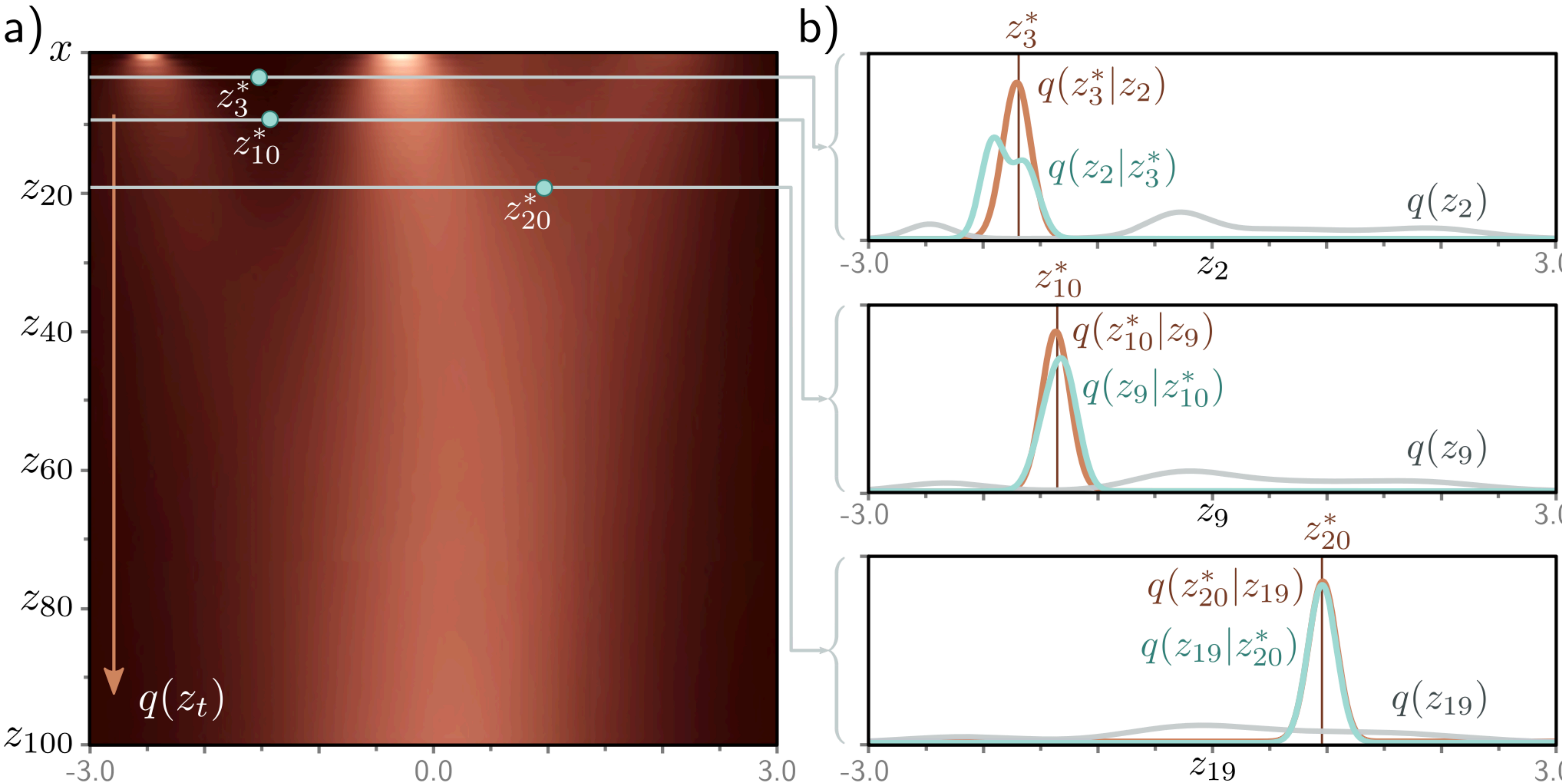
$$z_{t+1} = \sqrt{1-\beta_t} z_t + \sqrt{\beta_t}\epsilon \text{ where } \epsilon \sim \mathcal{N}(0,1)$$

$$Q(z_{t+1} \mid z_t) = \mathcal{N}(\sqrt{1-\beta_t} z_t, \beta_t)$$



Diffusions

- Over many timesteps with small β, the initial probability distribution $P(x)$ gets diffused until it eventually reaches standard normal $\mathcal{N}(0,1)$.



*From Prince textbook, fig. 18.5.

Diffusions

- More generally, we can let $\mathbf{z}_t \in \mathbb{R}^m$ can be any real-valued vector.
- We update $\mathbf{z}_t$ at time t=1 to $\mathbf{z}_2$ at time t=2 probabilistically by:
  1. Multiplying it by $\sqrt{1-\beta_t} < 1$.
  2. Adding standard Gaussian noise.

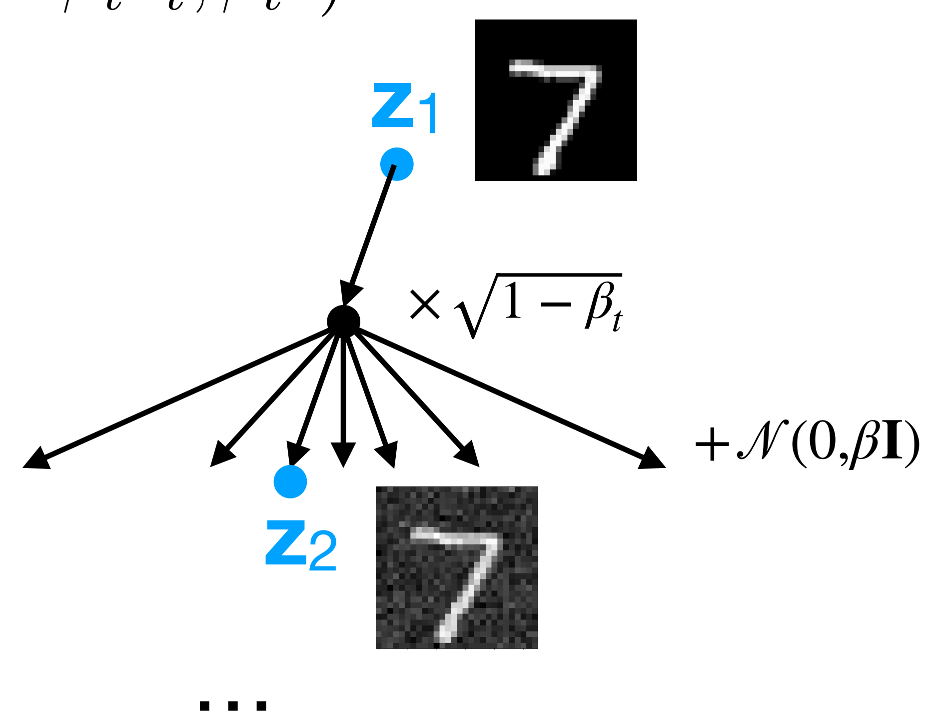
$$\mathbf{z}_{t+1} = \sqrt{1-\beta_t}\mathbf{z}_t + \sqrt{\beta_t}\epsilon \text{ where } \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t) = \mathcal{N}(\sqrt{1-\beta_t}\mathbf{z}_t, \beta_t\mathbf{I})$$

t=1

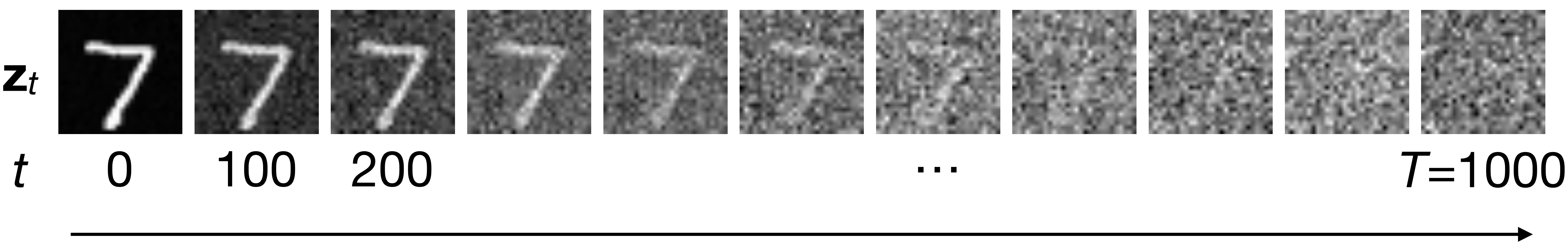
t=2

…



Diffusions: forward process

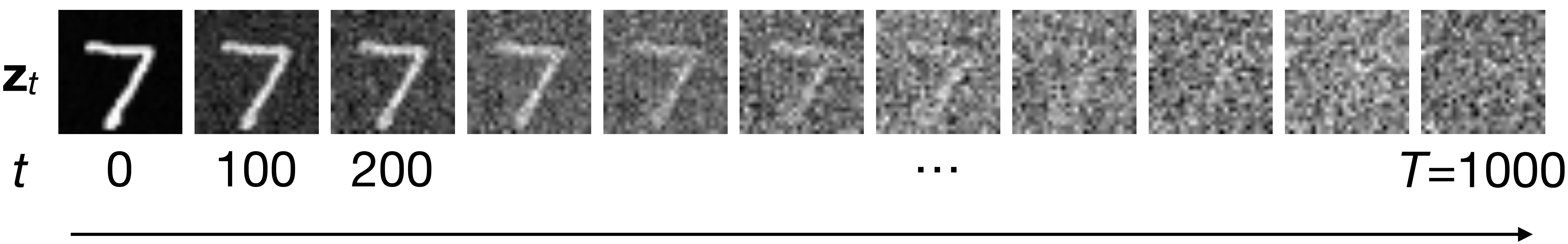
- In the forward process for t=0, …, T, we see $\mathbf{z}_t$ change from an image to pure Gaussian noise $\mathcal{N}(\mathbf{0}, \mathbf{I})$:



Forward update: $Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t) = \mathcal{N}(\sqrt{1-\beta_t}\mathbf{z}_t, \beta_t \mathbf{I})$

Diffusion kernel

- Similar to Plinko, there is a diffusion kernel that allows us to “skip” from $\mathbf{x}=\mathbf{z}_0$ to any $\mathbf{z}_t$.



Diffusion kernel:

$$Q(\mathbf{z}_t \mid \mathbf{x}) = \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}, (1-\alpha_t)\mathbf{I})$$

$$\text{where } \alpha_t = \prod_{t'=1}^{t}(1-\beta_{t'})$$

Gaussian random variables

- Let $u \sim \mathcal{N}(\mu_u, \sigma_u^2)$ and $v \sim \mathcal{N}(\mu_v, \sigma_v^2)$ be independent Gaussian random variables.
- Let *c* be a scalar constant.
- Then:
  - $u + v$ has mean $(\mu_u + \mu_v)$ and variance $(\sigma_u^2 + \sigma_v^2)$.

Gaussian random variables

- Let $u \sim \mathcal{N}(\mu_u, \sigma_u^2)$ and $v \sim \mathcal{N}(\mu_v, \sigma_v^2)$ be independent Gaussian random variables.

- Let c be a scalar constant.

- Then:

  Means add — Variances add

  - $u + v$ has mean $(\mu_u+\mu_v)$ and variance $(\sigma_u^2+\sigma_v^2)$.

Gaussian random variables

- Let $u \sim \mathcal{N}(\mu_u, \sigma_u^2)$ and $v \sim \mathcal{N}(\mu_v, \sigma_v^2)$ be independent Gaussian random variables.
- Let c be a scalar constant.
- Then:

Means add

Variances add

  - $u + v$ has mean $(\mu_u+\mu_v)$ and variance $(\sigma_u^2+\sigma_v^2)$.
  - cu has mean $c\mu_u$ and variance $c^2\sigma_u^2$.

Gaussian random variables

- Let $u \sim \mathcal{N}(\mu_u, \sigma_u^2)$ and $v \sim \mathcal{N}(\mu_v, \sigma_v^2)$ be independent Gaussian random variables.

- Let c be a scalar constant.

- Then:

  Means add　　　　Variances add

  - $u + v$ has mean $(\mu_u+\mu_v)$ and variance $(\sigma_u^2+\sigma_v^2)$.
  - cu has mean $c\mu_u$ and variance $c^2\sigma_u^2$.

  Mean scales linearly　　　　Variance scales quadratically

Diffusion kernel

- Consider how the forward update works 2 timesteps in succession:

$$\mathbf{z}_1 = \sqrt{1-\beta_1}\mathbf{x} + \sqrt{\beta_1}\epsilon$$

$$Q(\mathbf{z}_1 \mid \mathbf{x}) = \mathcal{N}(\sqrt{1-\beta_1}\mathbf{x}, \beta_1\mathbf{I})$$

Diffusion kernel

- Consider how the forward update works 2 timesteps in succession:

$$\mathbf{z}_1 = \sqrt{1-\beta_1}\mathbf{x} + \sqrt{\beta_1}\epsilon$$

$$Q(\mathbf{z}_1 \mid \mathbf{x}) = \mathcal{N}(\sqrt{1-\beta_1}\mathbf{x}, \beta_1 \mathbf{I})$$

$$\mathbf{z}_2 = \sqrt{1-\beta_2}\mathbf{z}_1 + \sqrt{\beta_2}\epsilon$$

Diffusion kernel

- Consider how the forward update works 2 timesteps in succession:

$$
\begin{aligned}
\mathbf{z}_1 &= \sqrt{1-\beta_1}\mathbf{x} + \sqrt{\beta_1}\epsilon \\
Q(\mathbf{z}_1 \mid \mathbf{x}) &= \mathcal{N}(\sqrt{1-\beta_1}\mathbf{x}, \beta_1 \mathbf{I}) \\
\mathbf{z}_2 &= \sqrt{1-\beta_2}\mathbf{z}_1 + \sqrt{\beta_2}\epsilon \\
&= \sqrt{1-\beta_2}(\sqrt{1-\beta_1}\mathbf{x} + \sqrt{\beta_1}\epsilon) + \sqrt{\beta_2}\epsilon
\end{aligned}
$$

Diffusion kernel

- Consider how the forward update works 2 timesteps in succession:

$$
\begin{aligned}
\mathbf{z}_1 &= \sqrt{1-\beta_1}\mathbf{x} + \sqrt{\beta_1}\epsilon \\
Q(\mathbf{z}_1 \mid \mathbf{x}) &= \mathcal{N}(\sqrt{1-\beta_1}\mathbf{x}, \beta_1\mathbf{I}) \\
\mathbf{z}_2 &= \sqrt{1-\beta_2}\mathbf{z}_1 + \sqrt{\beta_2}\epsilon \\
&= \sqrt{1-\beta_2}(\sqrt{1-\beta_1}\mathbf{x} + \sqrt{\beta_1}\epsilon) + \sqrt{\beta_2}\epsilon \\
&= \sqrt{1-\beta_1}\sqrt{1-\beta_2}\mathbf{x} + \sqrt{1-\beta_2}\sqrt{\beta_1}\epsilon + \sqrt{\beta_2}\epsilon
\end{aligned}
$$

Diffusion kernel

- Consider how the forward update works 2 timesteps in succession:

$$\mathbf{z}_1 = \sqrt{1-\beta_1}\mathbf{x} + \sqrt{\beta_1}\epsilon$$

$$Q(\mathbf{z}_1 \mid \mathbf{x}) = \mathcal{N}(\sqrt{1-\beta_1}\mathbf{x}, \beta_1\mathbf{I})$$

$$\begin{aligned}
\mathbf{z}_2 &= \sqrt{1-\beta_2}\mathbf{z}_1 + \sqrt{\beta_2}\epsilon \\
&= \sqrt{1-\beta_2}(\sqrt{1-\beta_1}\mathbf{x} + \sqrt{\beta_1}\epsilon) + \sqrt{\beta_2}\epsilon \\
&= \sqrt{1-\beta_1}\sqrt{1-\beta_2}\mathbf{x} + \sqrt{1-\beta_2}\sqrt{\beta_1}\epsilon + \sqrt{\beta_2}\epsilon
\end{aligned}$$

$$(\sqrt{1-\beta_2}\sqrt{\beta_1})^2 + (\sqrt{\beta_2})^2 = (1-\beta_2)\beta_1 + \beta_2$$

Variances add

Diffusion kernel

- Consider how the forward update works 2 timesteps in succession:

$$
\begin{aligned}
\mathbf{z}_1 &= \sqrt{1-\beta_1}\mathbf{x} + \sqrt{\beta_1}\epsilon \\
Q(\mathbf{z}_1 \mid \mathbf{x}) &= \mathcal{N}(\sqrt{1-\beta_1}\mathbf{x}, \beta_1 \mathbf{I}) \\
\mathbf{z}_2 &= \sqrt{1-\beta_2}\mathbf{z}_1 + \sqrt{\beta_2}\epsilon \\
&= \sqrt{1-\beta_2}(\sqrt{1-\beta_1}\mathbf{x} + \sqrt{\beta_1}\epsilon) + \sqrt{\beta_2}\epsilon \\
&= \sqrt{1-\beta_1}\sqrt{1-\beta_2}\mathbf{x} + \sqrt{1-\beta_2}\sqrt{\beta_1}\epsilon + \sqrt{\beta_2}\epsilon \\
(\sqrt{1-\beta_2}\sqrt{\beta_1})^2 + (\sqrt{\beta_2})^2 &= (1-\beta_2)\beta_1 + \beta_2 \\
&= 1 - (1-\beta_1)(1-\beta_2)
\end{aligned}
$$

Diffusion kernel

- Consider how the forward update works 2 timesteps in succession:

$$\begin{aligned}
\mathbf{z}_1 &= \sqrt{1-\beta_1}\mathbf{x} + \sqrt{\beta_1}\epsilon \\
Q(\mathbf{z}_1 \mid \mathbf{x}) &= \mathcal{N}(\sqrt{1-\beta_1}\mathbf{x}, \beta_1\mathbf{I}) \\
\mathbf{z}_2 &= \sqrt{1-\beta_2}\mathbf{z}_1 + \sqrt{\beta_2}\epsilon \\
&= \sqrt{1-\beta_2}(\sqrt{1-\beta_1}\mathbf{x} + \sqrt{\beta_1}\epsilon) + \sqrt{\beta_2}\epsilon \\
&= \sqrt{1-\beta_1}\sqrt{1-\beta_2}\mathbf{x} + \sqrt{1-\beta_2}\sqrt{\beta_1}\epsilon + \sqrt{\beta_2}\epsilon \\
(\sqrt{1-\beta_2}\sqrt{\beta_1})^2 + (\sqrt{\beta_2})^2 &= (1-\beta_2)\beta_1 + \beta_2 \\
&= 1 - (1-\beta_1)(1-\beta_2) \\
&= 1 - \alpha_2
\end{aligned}$$

Diffusion kernel

- Consider how the forward update works 2 timesteps in succession:

$$
\begin{aligned}
\mathbf{z}_1 &= \sqrt{1-\beta_1}\mathbf{x} + \sqrt{\beta_1}\epsilon \\
Q(\mathbf{z}_1 \mid \mathbf{x}) &= \mathcal{N}(\sqrt{1-\beta_1}\mathbf{x}, \beta_1 \mathbf{I}) \\
\mathbf{z}_2 &= \sqrt{1-\beta_2}\mathbf{z}_1 + \sqrt{\beta_2}\epsilon \\
&= \sqrt{1-\beta_2}(\sqrt{1-\beta_1}\mathbf{x} + \sqrt{\beta_1}\epsilon) + \sqrt{\beta_2}\epsilon \\
&= \sqrt{1-\beta_1}\sqrt{1-\beta_2}\mathbf{x} + \sqrt{1-\beta_2}\sqrt{\beta_1}\epsilon + \sqrt{\beta_2}\epsilon \\
(\sqrt{1-\beta_2}\sqrt{\beta_1})^2 + (\sqrt{\beta_2})^2 &= (1-\beta_2)\beta_1 + \beta_2 \\
&= 1 - (1-\beta_1)(1-\beta_2) \\
&= 1 - \alpha_2 \\
\implies & \\
Q(\mathbf{z}_2 \mid \mathbf{x}) &= \mathcal{N}(\sqrt{\alpha_2}\mathbf{x}, (1-\alpha_2)\mathbf{I})
\end{aligned}
$$

Diffusion kernel

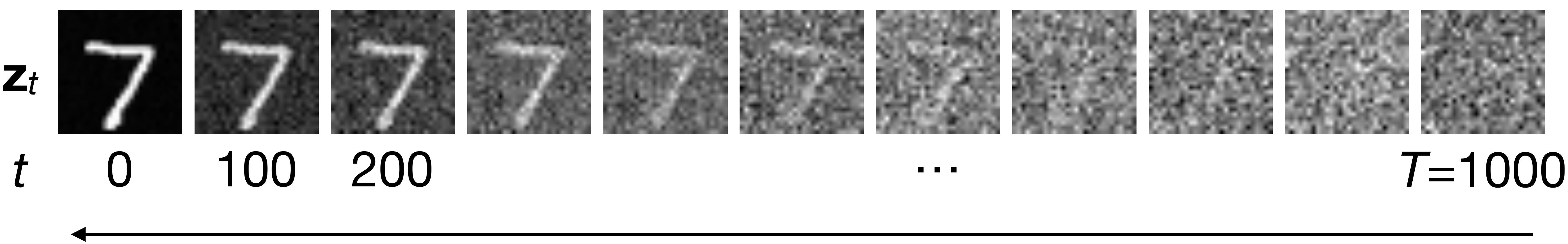
- More generally:

$$Q(\mathbf{z}_t \mid \mathbf{x}) = \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}, (1-\alpha_t)\mathbf{I})$$

$$\text{where } \alpha_t = \prod_{t'=1}^{t}(1-\beta_{t'})$$

Diffusions: reverse process

- Unlike in Plinko, computing the backward update is intractable…



Backwards update: $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}) = ?$

Diffusions: reverse process

- From Bayes' rule, we have:

$$Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}) = \frac{Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t) Q(\mathbf{z}_t)}{Q(\mathbf{z}_{t+1})}$$

- The numerator's first term is just the forward update.

Diffusions: reverse process

- From Bayes' rule, we have:

$$Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}) = \frac{Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t) Q(\mathbf{z}_t)}{Q(\mathbf{z}_{t+1})}$$

- The numerator's first term is just the forward update.
- But what about the other terms?

$$Q(\mathbf{z}_t) = \int_{\mathbf{x}} Q(\mathbf{z}_t \mid \mathbf{x}) P(\mathbf{x}) d\mathbf{x}$$

Diffusions: reverse process

- From Bayes' rule, we have:

$$Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}) = \frac{Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t)Q(\mathbf{z}_t)}{Q(\mathbf{z}_{t+1})}$$

- The numerator's first term is just the forward update.
- But what about the other terms?

$$Q(\mathbf{z}_t) = \int_{\mathbf{x}} Q(\mathbf{z}_t \mid \mathbf{x})P(\mathbf{x})d\mathbf{x}$$

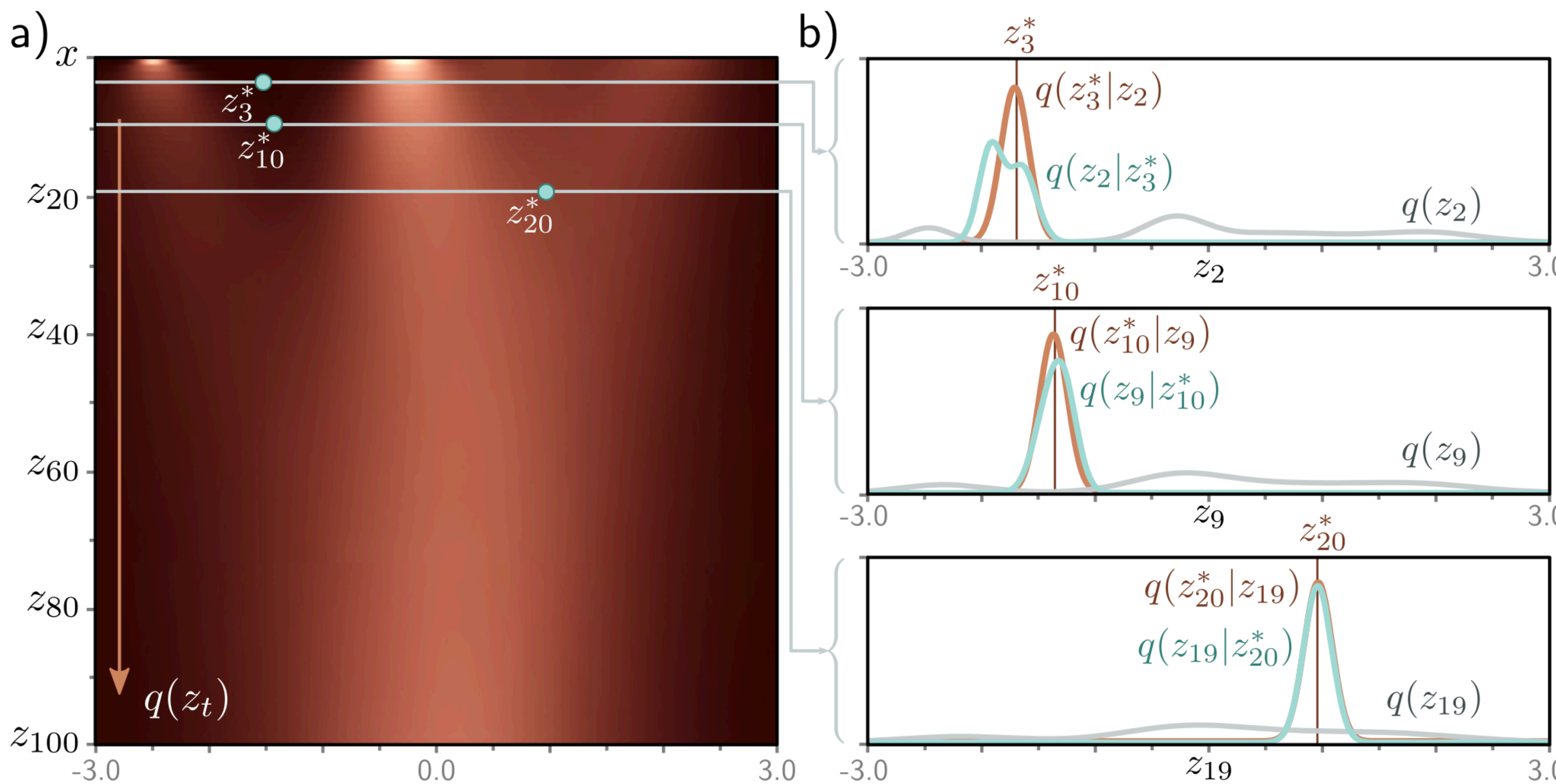
  - Unlike in Plinko, we have no formula for $P(\mathbf{x})$, and certainly no way to calculate this integral.

Diffusions: reverse process

- In practice, however, if we use small β, then the probability:

$$Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}) = \frac{Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t)Q(\mathbf{z}_t)}{Q(\mathbf{z}_{t+1})}$$

is *approximately* Gaussian:



*From Prince textbook, fig. 18.5.

Conditional reverse update

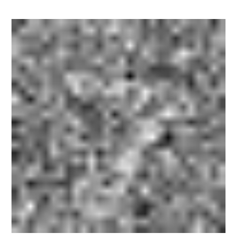
- We will thus train a NN decoder P_θ to estimate $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$.
- Similar to a VAE decoder, P_θ will output the mean $\mu_{\mathbf{z}_t}$ of a Gaussian distribution $\mathcal{N}(\mu_{\mathbf{z}_t}, \sigma^2\mathbf{I})$, where σ^2 can either be learned or set as a hyperparameter.

Conditional reverse update

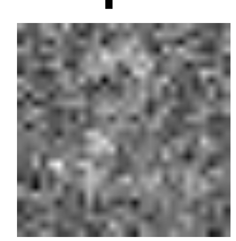
- We will thus train a NN decoder P_θ to estimate $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$.
- To train P_θ, we will use the *conditional* reverse update $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}, \mathbf{x})$ instead of $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$ (which we don't know).
  - This comes directly from the forthcoming ELBO derivation…

Conditional reverse update

- We will thus train a NN decoder P_θ to estimate $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$.
- To train P_θ, we will use the *conditional* reverse update $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}, \mathbf{x})$ instead of $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$ (which we don't know).
  - This comes directly from the forthcoming ELBO derivation…
- Consider: Given $\mathbf{z}_{t+1}=$ , which of the following images is more likely to equal $\mathbf{z}_t$?



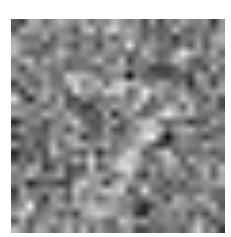
A



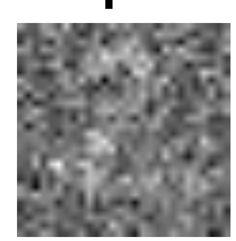
B

Conditional reverse update

- We will thus train a NN decoder P_θ to estimate $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$.
- To train P_θ, we will use the *conditional* reverse update $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}, \mathbf{x})$ instead of $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$ (which we don't know).
  - This comes directly from the forthcoming ELBO derivation…
- Consider: Given $\mathbf{z}_{t+1}$= , and $\mathbf{x}$= , which of the following images is more likely to equal $\mathbf{z}_t$?



A



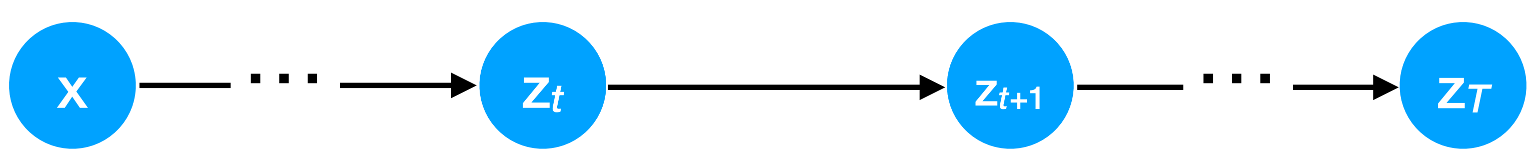
B

Conditional reverse update

- We will thus train a NN decoder P_θ to estimate $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$.

- To train P_θ, we will use the *conditional* reverse update $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}, \mathbf{x})$ instead of $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$ (which we don't know).

$$Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}, \mathbf{x}) \propto Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t, \mathbf{x}) Q(\mathbf{z}_t \mid \mathbf{x})$$



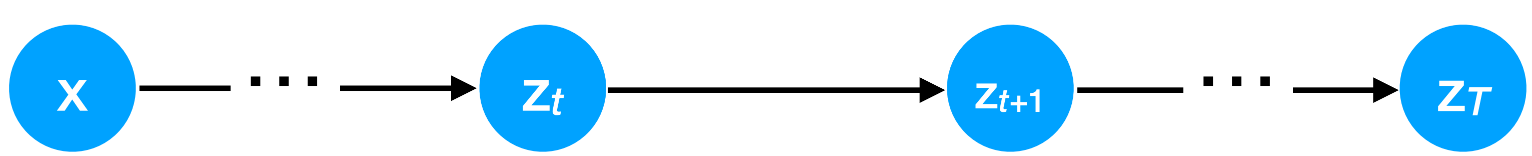
Conditional reverse update

- We will thus train a NN decoder P_θ to estimate $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$.

- To train P_θ, we will use the *conditional* reverse update $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}, \mathbf{x})$ instead of $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$ (which we don't know).

Cond. indep. due to Markov

$$\begin{aligned} Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}, \mathbf{x}) &\propto Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t, \mathbf{x}) Q(\mathbf{z}_t \mid \mathbf{x}) \\ &= Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t) Q(\mathbf{z}_t \mid \mathbf{x}) \end{aligned}$$

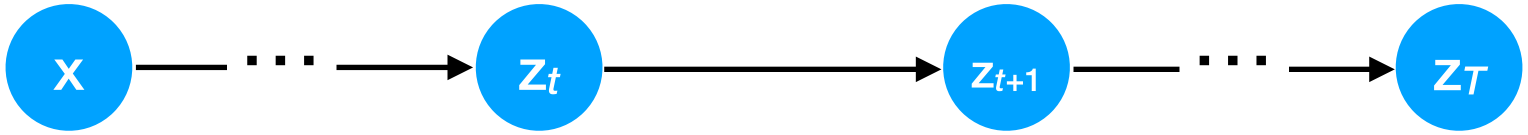


Conditional reverse update

- We will thus train a NN decoder P_θ to estimate $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$.

- To train P_θ, we will use the *conditional* reverse update $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}, \mathbf{x})$ instead of $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$ (which we don't know).

$$
\begin{aligned}
Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}, \mathbf{x}) &\propto Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t, \mathbf{x}) Q(\mathbf{z}_t \mid \mathbf{x}) \\
&= \underbrace{Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t)}_{\text{Forward update}} \underbrace{Q(\mathbf{z}_t \mid \mathbf{x})}_{\text{Diffusion kernel}}
\end{aligned}
$$

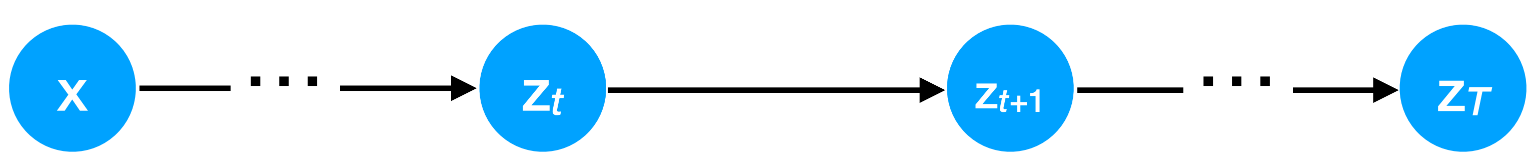


Conditional reverse update

- We will thus train a NN decoder P_θ to estimate $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$.

- To train P_θ, we will use the *conditional* reverse update $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}, \mathbf{x})$ instead of $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$ (which we don't know).

$$
\begin{aligned}
Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}, \mathbf{x}) &\propto Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t, \mathbf{x}) Q(\mathbf{z}_t \mid \mathbf{x}) \\
&= \underbrace{Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t)}_{\text{Gaussian}} \underbrace{Q(\mathbf{z}_t \mid \mathbf{x})}_{\text{Gaussian}}
\end{aligned}
$$

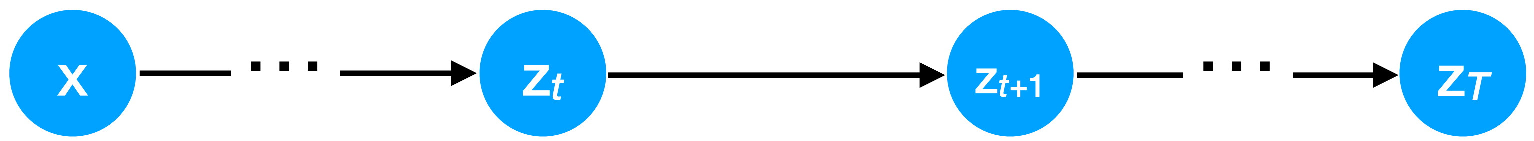


Conditional reverse update

- We will thus train a NN decoder P_θ to estimate $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$.
- To train P_θ, we will use the *conditional* reverse update $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}, \mathbf{x})$ instead of $Q(\mathbf{z}_t \mid \mathbf{z}_{t+1})$ (which we don't know).

$$
\begin{aligned}
Q(\mathbf{z}_t \mid \mathbf{z}_{t+1}, \mathbf{x}) &\propto Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t, \mathbf{x}) Q(\mathbf{z}_t \mid \mathbf{x}) \\
&= Q(\mathbf{z}_{t+1} \mid \mathbf{z}_t) Q(\mathbf{z}_t \mid \mathbf{x}) \\
&= \mathcal{N}(\ldots, \ldots)
\end{aligned}
$$

Product of two Gaussians
is also Gaussian

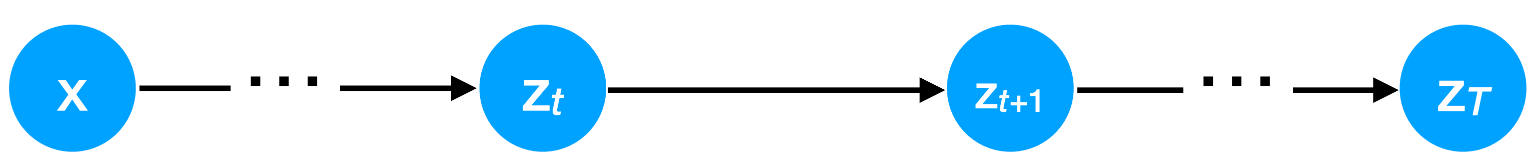


Diffusions: an LVM

- Similar to VAEs, we set $P(\mathbf{z}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and we then formulate the diffusion objective as the log-likelihood of **x**:

$$\log P(\mathbf{x}) = \log \int_{\mathbf{z}_{1,\ldots,T}} P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T) d\mathbf{z}_{1,\ldots,T}$$

Law of total probability

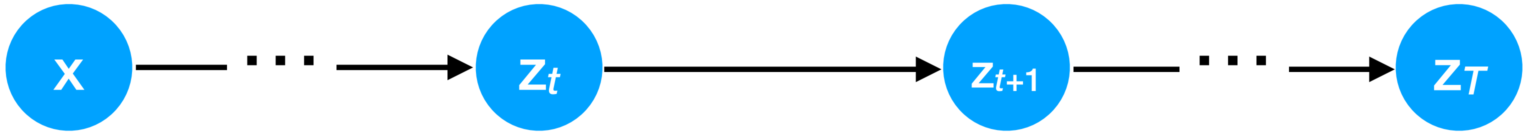


Diffusions: an LVM

- Similar to VAEs, we set $P(\mathbf{z}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and we then formulate the diffusion objective as the log-likelihood of **x**:

$$\log P(\mathbf{x}) = \log \int_{\mathbf{z}_{1,\ldots,T}} P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T) d\mathbf{z}_{1,\ldots,T}$$

$$= \log \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \frac{P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T)}{Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x})} d\mathbf{z}_{1,\ldots,T}$$ True for all non-zero Q

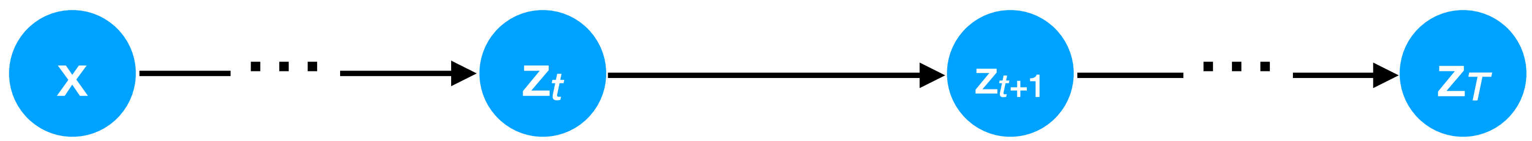


Diffusions: an LVM

- Similar to VAEs, we set $P(\mathbf{z}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and we then formulate the diffusion objective as the log-likelihood of **x**:

$$\begin{aligned}
\log P(\mathbf{x}) &= \log \int_{\mathbf{z}_{1,\ldots,T}} P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T) d\mathbf{z}_{1,\ldots,T} \\
&= \log \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \frac{P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T)}{Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x})} d\mathbf{z}_{1,\ldots,T} \\
&\geq \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \log \frac{P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T)}{Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x})} d\mathbf{z}_{1,\ldots,T} \quad \text{Jensen's inequality}
\end{aligned}$$

ELBO

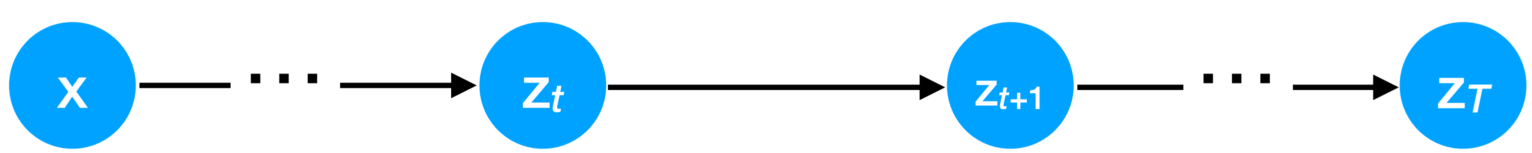


Diffusions: an LVM

- Similar to VAEs, we set $P(\mathbf{z}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and we then formulate the diffusion objective as the log-likelihood of **x**:

$$
\begin{aligned}
\log P(\mathbf{x}) &= \log \int_{\mathbf{z}_{1,\ldots,T}} P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T) d\mathbf{z}_{1,\ldots,T} \\
&= \log \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \frac{P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T)}{Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x})} d\mathbf{z}_{1,\ldots,T} \\
&\geq \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \log \frac{P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T)}{Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x})} d\mathbf{z}_{1,\ldots,T} \\
&= \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \left(\log P(\mathbf{x} \mid \mathbf{z}_1) + \sum_{t=2}^{T} \log \left[\frac{P(\mathbf{z}_{t-1} \mid \mathbf{z}_t)}{Q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x})} \right] + \log \frac{P(\mathbf{z}_T)}{Q(\mathbf{z}_T \mid \mathbf{x})} \right) d\mathbf{z}_{1,\ldots,T}
\end{aligned}
$$

Lots of algebra — see Prince, chapter 18.



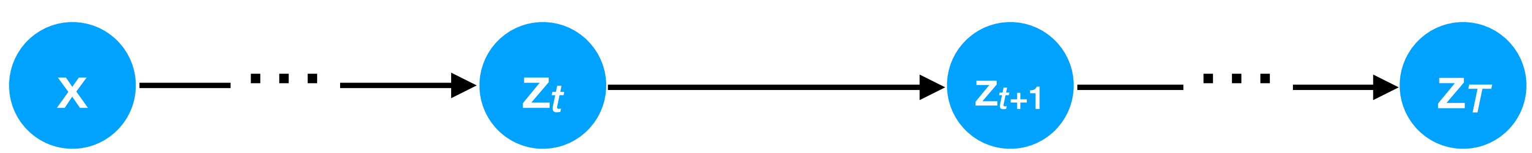
Diffusions: an LVM

- Similar to VAEs, we set $P(\mathbf{z}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and we then formulate the diffusion objective as the log-likelihood of **x**:

$$
\begin{aligned}
\log P(\mathbf{x}) &= \log \int_{\mathbf{z}_{1,\ldots,T}} P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T) d\mathbf{z}_{1,\ldots,T} \\
&= \log \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \frac{P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T)}{Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x})} d\mathbf{z}_{1,\ldots,T} \\
&\geq \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \log \frac{P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T)}{Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x})} d\mathbf{z}_{1,\ldots,T} \\
&= \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \left(\log P(\mathbf{x} \mid \mathbf{z}_1) + \sum_{t=2}^{T} \log \left[\frac{P(\mathbf{z}_{t-1} \mid \mathbf{z}_t)}{Q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x})} \right] + \boxed{\cancel{\log \frac{P(\mathbf{z}_T)}{Q(\mathbf{z}_T \mid \mathbf{x})}}} \right) d\mathbf{z}_{1,\ldots,T} \\
&\approx \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \left(\log P(\mathbf{x} \mid \mathbf{z}_1) + \sum_{t=2}^{T} \log \left[\frac{P(\mathbf{z}_{t-1} \mid \mathbf{z}_t)}{Q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x})} \right] \right) d\mathbf{z}_{1,\ldots,T}
\end{aligned}
$$

$Q(\mathbf{z}_T | \mathbf{x}) \approx P(\mathbf{z}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I}$

log 1 = 0

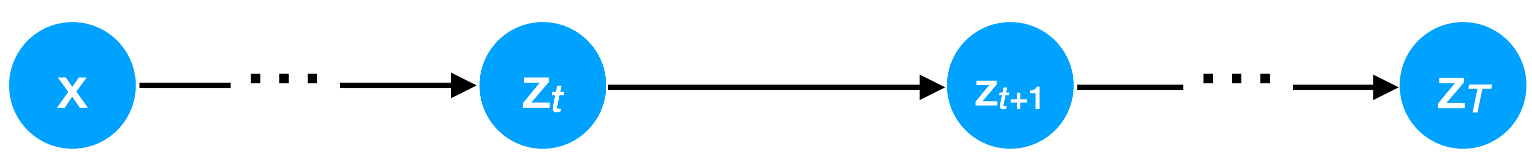


Diffusions: an LVM

- Similar to VAEs, we set $P(\mathbf{z}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and we then formulate the diffusion objective as the log-likelihood of **x**:

$$
\begin{aligned}
\log P(\mathbf{x}) &= \log \int_{\mathbf{z}_{1,\ldots,T}} P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T) d\mathbf{z}_{1,\ldots,T} \\
&= \log \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \frac{P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T)}{Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x})} d\mathbf{z}_{1,\ldots,T} \\
&\geq \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \log \frac{P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T)}{Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x})} d\mathbf{z}_{1,\ldots,T} \\
&= \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \left(\log P(\mathbf{x} \mid \mathbf{z}_1) + \sum_{t=2}^{T} \log \left[\frac{P(\mathbf{z}_{t-1} \mid \mathbf{z}_t)}{Q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x})} \right] + \log \frac{P(\mathbf{z}_T)}{Q(\mathbf{z}_T \mid \mathbf{x})} \right) d\mathbf{z}_{1,\ldots,T} \\
&\approx \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \left(\log P(\mathbf{x} \mid \mathbf{z}_1) + \sum_{t=2}^{T} \log \left[\frac{P(\mathbf{z}_{t-1} \mid \mathbf{z}_t)}{Q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x})} \right] \right) d\mathbf{z}_{1,\ldots,T} \\
&= \mathbb{E}_{Q(\mathbf{z}_1 \mid \mathbf{x})} [\log P_\theta(\mathbf{x} \mid \mathbf{z}_1)] - \sum_{t=2}^{T} \mathbb{E}_{Q(\mathbf{z}_t \mid \mathbf{x})} D_{\mathrm{KL}} [Q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x}) \parallel P_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t)]
\end{aligned}
$$

Definitions of expectation and KL divergence



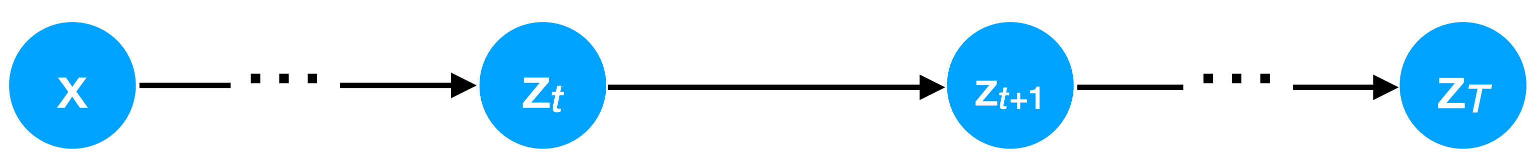
Diffusions: an LVM

- Similar to VAEs, we set $P(\mathbf{z}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and we then formulate the diffusion objective as the log-likelihood of **x**:

$$
\begin{aligned}
\log P(\mathbf{x}) &= \log \int_{\mathbf{z}_{1,\ldots,T}} P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T) d\mathbf{z}_{1,\ldots,T} \\
&= \log \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \frac{P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T)}{Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x})} d\mathbf{z}_{1,\ldots,T} \\
&\geq \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \log \frac{P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T)}{Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x})} d\mathbf{z}_{1,\ldots,T} \\
&= \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \left(\log P(\mathbf{x} \mid \mathbf{z}_1) + \sum_{t=2}^{T} \log \left[\frac{P(\mathbf{z}_{t-1} \mid \mathbf{z}_t)}{Q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x})} \right] + \log \frac{P(\mathbf{z}_T)}{Q(\mathbf{z}_T \mid \mathbf{x})} \right) d\mathbf{z}_{1,\ldots,T} \\
&\approx \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \left(\log P(\mathbf{x} \mid \mathbf{z}_1) + \sum_{t=2}^{T} \log \left[\frac{P(\mathbf{z}_{t-1} \mid \mathbf{z}_t)}{Q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x})} \right] \right) d\mathbf{z}_{1,\ldots,T} \\
&= \mathbb{E}_{Q(\mathbf{z}_1 \mid \mathbf{x})} \left[\log P_\theta(\mathbf{x} \mid \mathbf{z}_1) \right] - \sum_{t=2}^{T} \mathbb{E}_{Q(\mathbf{z}_t \mid \mathbf{x})} D_{\mathrm{KL}} \left[Q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x}) \parallel P_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t) \right]
\end{aligned}
$$

Sum of expected KL divergences between P_θ and conditional reverse updates.

Expected reconstruction loss from $\mathbf{z}_1$ to **x**



Diffusions: an LVM

- Similar to VAEs, we set $P(\mathbf{z}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and we then formulate the diffusion objective as the log-likelihood of **x**:

$$
\begin{aligned}
\log P(\mathbf{x}) &= \log \int_{\mathbf{z}_{1,\ldots,T}} P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T) d\mathbf{z}_{1,\ldots,T} \\
&= \log \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \frac{P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T)}{Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x})} d\mathbf{z}_{1,\ldots,T} \\
&\geq \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \log \frac{P(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_T)}{Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x})} d\mathbf{z}_{1,\ldots,T} \\
&= \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \left(\log P(\mathbf{x} \mid \mathbf{z}_1) + \sum_{t=2}^{T} \log \left[\frac{P(\mathbf{z}_{t-1} \mid \mathbf{z}_t)}{Q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x})} \right] + \log \frac{P(\mathbf{z}_T)}{Q(\mathbf{z}_T \mid \mathbf{x})} \right) d\mathbf{z}_{1,\ldots,T} \\
&\approx \int_{\mathbf{z}_{1,\ldots,T}} Q(\mathbf{z}_1, \ldots, \mathbf{z}_T \mid \mathbf{x}) \left(\log P(\mathbf{x} \mid \mathbf{z}_1) + \sum_{t=2}^{T} \log \left[\frac{P(\mathbf{z}_{t-1} \mid \mathbf{z}_t)}{Q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x})} \right] \right) d\mathbf{z}_{1,\ldots,T} \\
&= \underbrace{\mathbb{E}_{Q(\mathbf{z}_1 \mid \mathbf{x})} \left[\log P_\theta(\mathbf{x} \mid \mathbf{z}_1) \right]}_{\text{MSE}} - \underbrace{\sum_{t=2}^{T} \mathbb{E}_{Q(\mathbf{z}_t \mid \mathbf{x})} D_{\mathrm{KL}} \left[Q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x}) \parallel P_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t) \right]}_{\text{Closed-formula exists for two Gaussians}}
\end{aligned}
$$

$$\mathbf{x} \to \cdots \to \mathbf{z}_t \to \mathbf{z}_{t+1} \to \cdots \to \mathbf{z}_T$$