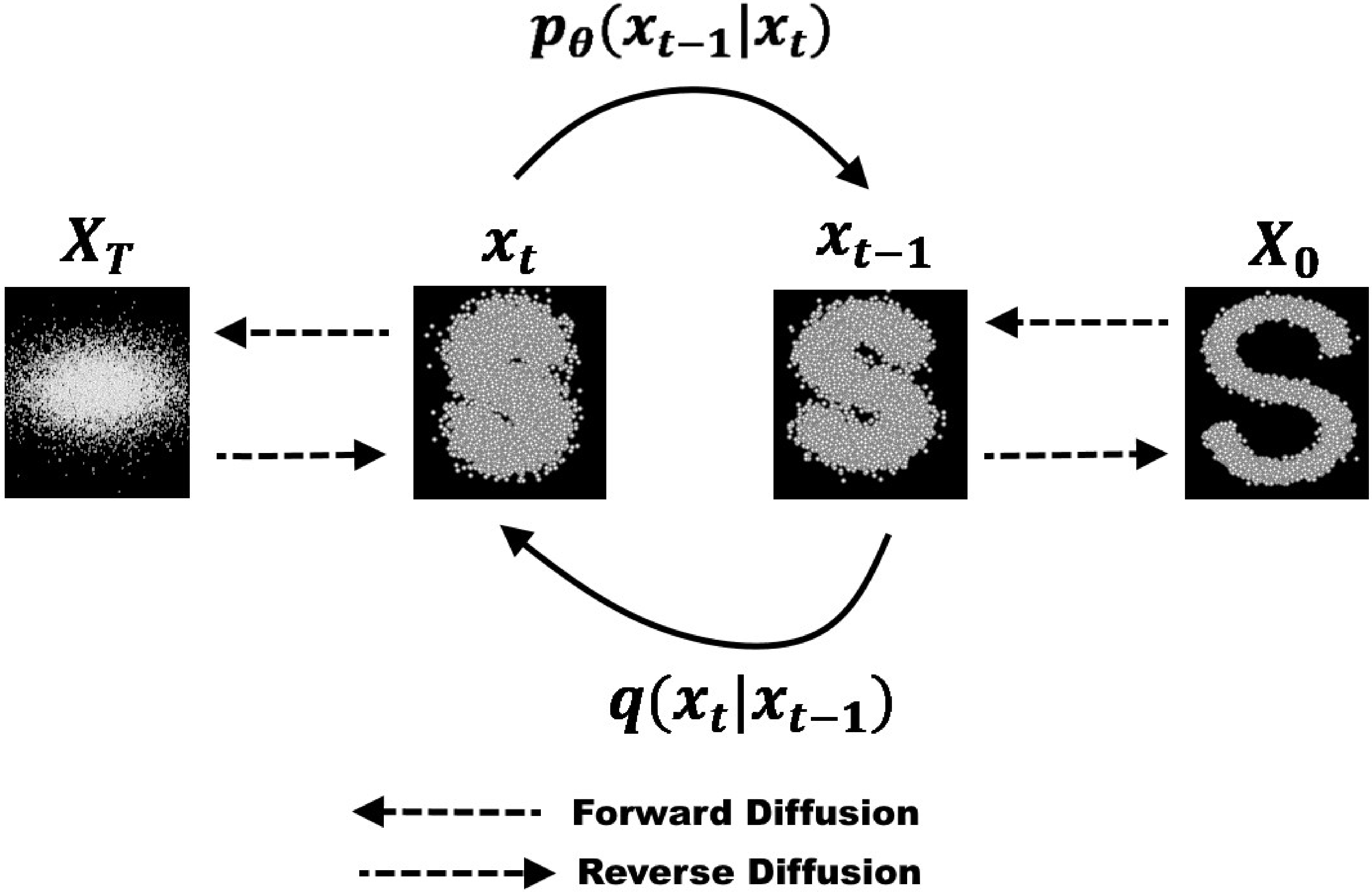
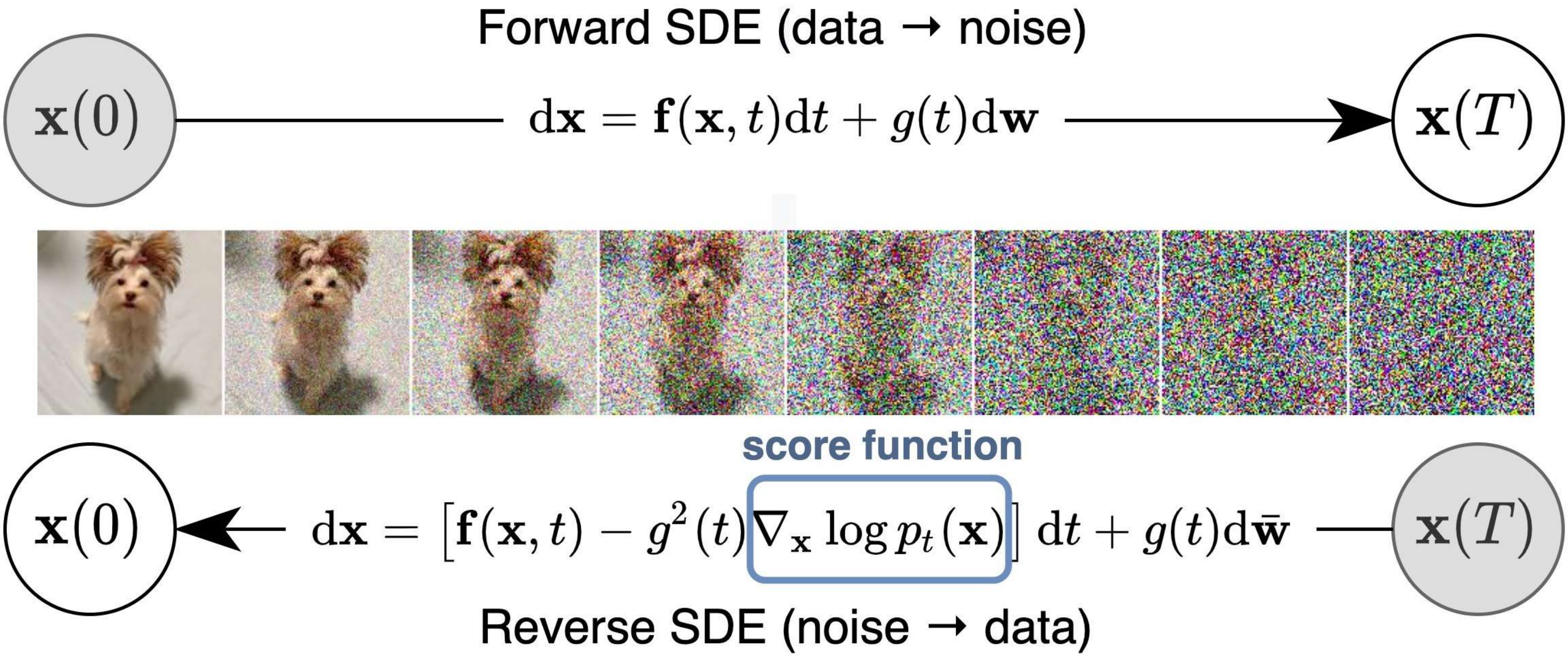
# Restricted Boltzmann machines and generative diffusion models: is there a connection?

Eric Paquet
NRC

24 October 2024



$p_\theta(x_{t-1}|x_t)$
$X_T$
$x_t$
$x_{t-1}$
$X_0$
$q(x_t|x_{t-1})$
Forward Diffusion
Reverse Diffusion


Forward SDE (data → noise)
$\mathbf{x}(0)$
$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$
$\mathbf{x}(T)$
score function
$\mathbf{x}(0)$
$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$
$\mathbf{x}(T)$
Reverse SDE (noise → data)

# Key differences

- Explicit latent variables: In RBMs, the **hidden units** are explicit latent variables that interact with the visible units. In contrast, in generative diffusion models, the latent variables (noise) are implicit, and there is no separate hidden layer.
- Sampling method: RBMs use **Gibbs sampling** (or similar MCMC methods) for generating samples, while diffusion models rely on **stochastic differential equations** (SDEs) to generate samples through a diffusion process.
- Training objective: RBMs are typically trained using **maximum likelihood** or its approximation (**contrastive divergence**). In contrast, diffusion models are trained by **matching the score function** over multiple time steps of the reverse diffusion process.

# Denoising versus Gibbs sampling

- Diffusion

$$\mathbf{x}_T \rightarrow \mathbf{x}_{T-1} \rightarrow \ldots \rightarrow \mathbf{x}_t \rightarrow \ldots \rightarrow \mathbf{x}_1 \rightarrow \mathbf{x}_0$$

- Gibbs sampling

$$\boxed{\mathbf{h}^{(T-1)} \sim p\left(\mathbf{h} \middle| \mathbf{v}^{(T)}\right) \rightarrow \mathbf{v}^{(T-1)} \sim p\left(\mathbf{v} \middle| \mathbf{h}^{(T-1)}\right)} \rightarrow$$

$$\mathbf{h}^{(T-2)} \sim p\left(\mathbf{h} \middle| \mathbf{v}^{(T-1)}\right) \rightarrow \mathbf{v}^{(T-2)} \sim p\left(\mathbf{v} \middle| \mathbf{h}^{(T-2)}\right) \rightarrow \ldots \rightarrow$$

$$\mathbf{h}^{(t)} \sim p\left(\mathbf{h} \middle| \mathbf{v}^{(t+1)}\right) \rightarrow \mathbf{v}^{(t)} \sim p\left(\mathbf{v} \middle| \mathbf{h}^{(t)}\right) \rightarrow \ldots \rightarrow$$

$$\mathbf{h}^{(1)} \sim p\left(\mathbf{h} \middle| \mathbf{v}^{(2)}\right) \rightarrow \mathbf{v}^{(1)} \sim p\left(\mathbf{v} \middle| \mathbf{h}^{(1)}\right) \rightarrow$$

$$\underbrace{\mathbf{h}^{(0)} \sim p\left(\mathbf{h} \middle| \mathbf{v}^{(1)}\right) \rightarrow \mathbf{v}^{(0)} \sim p\left(\mathbf{v} \middle| \mathbf{h}^{(0)}\right)}_{\mathbf{x}_0}$$

# Contrastive divergence-1 (CD-1) versus score-matching

- CD-1 $$\underbrace{\mathbf{h} \sim p\left(\mathbf{h} \middle| \mathbf{v}^{(i)}\right)}_{\mathcal{D} \sim p\left(\mathbf{h} \middle| \mathbf{v}^{(i)}\right)} \rightarrow \underbrace{\mathbf{v}' \sim p\left(\mathbf{v} \middle| \mathbf{h}\right) \rightarrow \mathbf{h}' \sim p\left(\mathbf{h} \middle| \mathbf{v}'\right)}_{\mathcal{M} \sim p(\mathbf{h},\mathbf{v})}$$

$$\Delta w_{ij} = \varepsilon\left[\left\langle v_i h_j \right\rangle_{\mathcal{D}} - \left\langle v_i' h_j' \right\rangle_{\mathcal{M}}\right]$$

- This is quite different from score-matching:

$$\mathcal{L}(\theta) = \mathbb{E}_{t \sim \mathcal{U}[0,T]} \lambda(t) \mathbb{E}_{p(\mathbf{x}_0) p_{0t}(\mathbf{x}_t | \mathbf{x}_0)} \left[ \left\| \nabla_{\mathbf{x}_t} \log p_{0t}(\mathbf{x}_t) - \mathbf{s}_\theta(\mathbf{x}_t, t) \right\|_{\Lambda_t}^2 \right], \quad \lambda(t) \in \mathbb{R}^+$$

- If there is a connection, it should be made through the score function

# Why is it so important to employ a score-based diffusion model

- Energy:

$$E(\mathbf{h},\mathbf{v}) = -\mathbf{b}^T\mathbf{v} - \mathbf{c}^T\mathbf{h} - \mathbf{v}^T\mathbf{W}\mathbf{h}$$

- The score function of the RBM is related to the gradient of the free energy function of the **visible units**

$$\log p(\mathbf{v}) = -F(\mathbf{v}) - \log Z \Rightarrow \nabla_{\mathbf{v}} \log p(\mathbf{v}) = -\nabla_{\mathbf{v}} F(\mathbf{v})$$

$$F(\mathbf{v}) = -\log\sum_{\mathbf{h}} \exp\left(-E(\mathbf{v},\mathbf{h})\right) = -\mathbf{b}^T\mathbf{v} - \sum_j \log\left(1+\exp\left(c_j + \sum_i w_{ij} v_i\right)\right)$$

$$\frac{\partial F(\mathbf{v})}{\partial v_i} = -b_i - \sum_j w_{ij}\sigma\left(c_j + \sum_i w_{ij} v_i\right)$$

# Score-based diffusion models

- Forward noising process (drift and diffusion matrices):

$$d\mathbf{x} = \mathbf{F}_t \mathbf{x}\, dt + \mathbf{G}_t d\mathbf{w}, \quad \mathbf{x} \in \mathbb{R}^D, \mathbf{F} \in \mathbb{R}^{D\times D}, \mathbf{G} \in \mathbb{R}^{D\times D}, \mathbf{w} \in \mathbb{R}^D$$

- The reverse-time diffusion process has a closed-form solution:

$$d\mathbf{x} = \left[\mathbf{F}_t \mathbf{x} - \mathbf{G}_t \mathbf{G}_t^T \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] dt + \mathbf{G}_t d\bar{\mathbf{w}}$$

- Score matching:

$$\mathbb{E}_{p(\mathbf{x}_t)}\left[\frac{1}{2}\left\|\mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\right\|_{\Lambda_t}^2\right] = \mathbb{E}_{p(\mathbf{x}_0)p_{0t}(\mathbf{x}_t|\mathbf{x}_0)}\left[\frac{1}{2}\left\|\mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{x}_0)\right\|_{\Lambda_t}^2\right] + \Omega$$

- Objective function:

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,T]} \lambda(t) \mathbb{E}_{p(\mathbf{x}_0)p_{0t}(\mathbf{x}_t|\mathbf{x}_0)}\left[\left\|\nabla_{\mathbf{x}_t} \log p_{0t}(\mathbf{x}_t|\mathbf{x}_0) - \mathbf{s}_\theta(\mathbf{x}_t, t)\right\|_{\Lambda_t}^2\right], \quad \lambda(t) \in \mathbb{R}^+$$

# Both processes are Markovian

- The **continuous** diffusion process is Markovian since it admits **a Fokker-Planck-Kolmogorov equation:**

$$\frac{\partial p_t(\mathbf{x})}{\partial t} = -\nabla \cdot \left\{ \left[ \mathbf{F}_t \mathbf{x} - \frac{1}{2} \mathbf{G}_t \mathbf{G}_t^T \nabla_{\mathbf{x}} p_t(\mathbf{x}) \right] p_t(\mathbf{x}) \right\}$$

- In the case of the RBM, Gibbs sampling has a Markovian structure:

$$\boxed{\mathbf{h}^{(T-1)} \sim p\left(\mathbf{h} \middle| \mathbf{v}^{(T)}\right) \rightarrow \mathbf{v}^{(T-1)} \sim p\left(\mathbf{v} \middle| \mathbf{h}^{(T-1)}\right)} \rightarrow$$

$$\mathbf{h}^{(T-2)} \sim p\left(\mathbf{h} \middle| \mathbf{v}^{(T-1)}\right) \rightarrow \mathbf{v}^{(T-2)} \sim p\left(\mathbf{v} \middle| \mathbf{h}^{(T-2)}\right) \rightarrow \ldots \rightarrow$$

$$\mathbf{h}^{(t)} \sim p\left(\mathbf{h} \middle| \mathbf{v}^{(t+1)}\right) \rightarrow \mathbf{v}^{(t)} \sim p\left(\mathbf{v} \middle| \mathbf{h}^{(t)}\right) \rightarrow \ldots \rightarrow$$

$$\mathbf{h}^{(1)} \sim p\left(\mathbf{h} \middle| \mathbf{v}^{(2)}\right) \rightarrow \mathbf{v}^{(1)} \sim p\left(\mathbf{v} \middle| \mathbf{h}^{(1)}\right) \rightarrow$$

$$\underbrace{\mathbf{h}^{(0)} \sim p\left(\mathbf{h} \middle| \mathbf{v}^{(1)}\right) \rightarrow \mathbf{v}^{(0)} \sim p\left(\mathbf{v} \middle| \mathbf{h}^{(0)}\right)}_{\mathbf{x}_0}$$

# The situation is not as simple for the discrete case

- If the stochastic differential equation is integrated with the **Euler-Maruyama** method, the denoising process is Markovian (unless the model is made non-Markovian by construction):

$$\hat{\mathbf{x}}_{t-\Delta t} = \mathbf{F}_t \hat{\mathbf{x}}_t + \mathbf{G}_t \mathbf{G}_t^T \mathbf{s}_\theta \left( \hat{\mathbf{x}}_t, t \right) + \mathbf{G}_t d\overline{\mathbf{w}}$$

- Unfortunately, this approach results in low accuracy and is unstable when the step size is insufficiently small.

# Diffusion exponential integrator sampler (DEIS): Markovian or non-Markovian, that is the question

- DEIS solves the reverse equation with an exponential integrator (EI) by taking advantage of the semilinear structure of the reverse process

$$\hat{\mathbf{x}}_{t-\Delta t} = \mathbf{\Psi}\left(t-\Delta t, t\right)\hat{\mathbf{x}}_t + \left[\int_t^{t-\Delta t} \mathbf{\Psi}\left(t-\Delta t, \tau\right)\mathbf{G}_\tau \mathbf{G}_\tau^T \mathbf{s}_\theta\left(\hat{\mathbf{x}}_\tau, \tau\right) d\tau\right] + \int_t^{t-\Delta t} \mathbf{\Psi}\left(t-\Delta t, t\right)\mathbf{G}_\tau d\overline{\mathbf{w}}$$

- With the transition matrix being given by

$$\frac{\partial \mathbf{\Psi}\left(t-\Delta t, t\right)}{\partial t} = \mathbf{F}_t \mathbf{\Psi}\left(t-\Delta t, t\right), \quad \mathbf{\Psi}\left(t, t\right) = \mathbf{I}$$

- https://arxiv.org/abs/2204.13902

# Non-Markovian structure of the DEIS: the devil is in the detail

- The solution for the variance-preserving stochastic differential equation:

| $\mathbf{F}_t$ | $\mathbf{G}_t$ |
|---|---|
| $\frac{1}{2}\frac{d\log\beta_t}{dt}\mathbf{I}$ | $\sqrt{-\frac{d\log\beta_t}{dt}}\mathbf{I}$ |

$$\mathbf{s}_\theta\left(\mathbf{x}_t,t\right) \approx -\mathbf{L}_t^{-T}\boldsymbol{\varepsilon}_\theta\left(\mathbf{x}_t,t\right)$$

$$\hat{\mathbf{x}}_{t-\Delta t} = \underbrace{\sqrt{\beta_{t-\Delta t}}\left(\frac{\hat{\mathbf{x}}_t - \sqrt{1-\beta_t}\boldsymbol{\varepsilon}_\theta\left(\hat{\mathbf{x}}_t,t\right)}{\sqrt{\beta_t}}\right)}_{\hat{\mathbf{x}}_0} + \sqrt{1-\beta_{t-\Delta t} - \frac{1-\beta_{t-\Delta t}}{1-\beta_t}\left(1-\frac{\beta_t}{\beta_{t-\Delta t}}\right)}\,\boldsymbol{\varepsilon}_\theta\left(\hat{\mathbf{x}}_t,t\right) + \sqrt{\frac{1-\beta_{t-\Delta t}}{1-\beta_t}\left(1-\frac{\beta_t}{\beta_{t-\Delta t}}\right)}\,\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}\left(\mathbf{0},\mathbf{I}\right)$$

- Which is not Markovian!
- The discrete diffusion process might become non-Markovian due to the integration method, an undesirable feature from an RBM point of view.

# What is the solution (perhaps)?

- Predictor-corrector method.
- Predictor: Euler-Maruyama.
- Corrector: Langevin equation (stochastic equation).
- Because of its stochastic nature, the Langevin equation helps to escape local minima.
- The Fokker-Planck equation may be derived from the Langevin equation.
- The Fokker-Planck equation has a Markovian structure.

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \varepsilon_i \mathbf{s}_\theta\left(\mathbf{x}_i, i\right) + \sqrt{2\varepsilon_i}\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$$

# Evidence lower bound (ELBO) and generative diffusion models

- "Variational autoencoders (VAE) are trained using the ELBO as a proxy loss function for the log-likelihood"
- Generative diffusion models use score matching and noise prediction
- ELBO of continuous-time diffusion models (importance sampling)

$$-\mathcal{L}(\mathbf{x}) = \frac{1}{2} E_{t \sim \mathcal{U}(0,1), \boldsymbol{\varepsilon} \sim \mathcal{N}(0,1)} \left[ -\frac{d\lambda}{dt} \left[ = \frac{1}{p(\lambda)} \right] \left\| \hat{\boldsymbol{\varepsilon}}_{\boldsymbol{\theta}}\left(\mathbf{z}_t; \lambda_t\right) - \boldsymbol{\varepsilon} \right\|_2^2 \right] + \kappa$$

- Related to the log signal-to-noise ratio (log-SNR), the noise schedule is a strictly monotonically decreasing function (bijection):

$$\mathbf{z}_t = \alpha_\lambda + \sigma_\lambda \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\lambda = \log\left(\alpha_\lambda^2 / \sigma_\lambda^2\right)$$

$$\lambda = -2 \log \tan(\pi t/2)$$

# Theorem https://arxiv.org/abs/2303.00848

- If the weighting is a monotonically increasing function of time, then the weighted diffusion objective

$$\mathcal{L}_w(\mathbf{x}) = \frac{1}{2} E_{t\sim\mathcal{U}(0,1),\boldsymbol{\varepsilon}\sim\mathcal{N}(0,1)}\left[-\frac{d\lambda}{dt} w(\lambda_t) \left\|\hat{\boldsymbol{\varepsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t;\lambda_t) - \boldsymbol{\varepsilon}\right\|_2^2\right]$$

- is equivalent to the ELBO with data augmentation (additive noise).

$$w(\lambda) = \operatorname{sech}(\lambda/2), \quad = \exp(-\lambda/2), \quad \ldots$$

- Comparing apples with apples!

# Conclusions

- In a few words: SDEs, Markovian, ELBO, integration method
- The connection between Restricted Boltzmann Machines (RBMs) and score-based generative models lies in their shared focus on learning the gradient of the log-probability distribution (the score function). While RBMs implicitly learn the score through the energy function and contrastive divergence, score-based models explicitly learn the score function and use it to generate samples via stochastic differential equations (SDEs). Both approaches involve energy-based modelling and sampling via gradients, but their specific training and sampling methods differ.
- Thank you!