

# CS480/680: Introduction to Machine Learning

## Lec 23: Diffusion Models

Yaoliang Yu

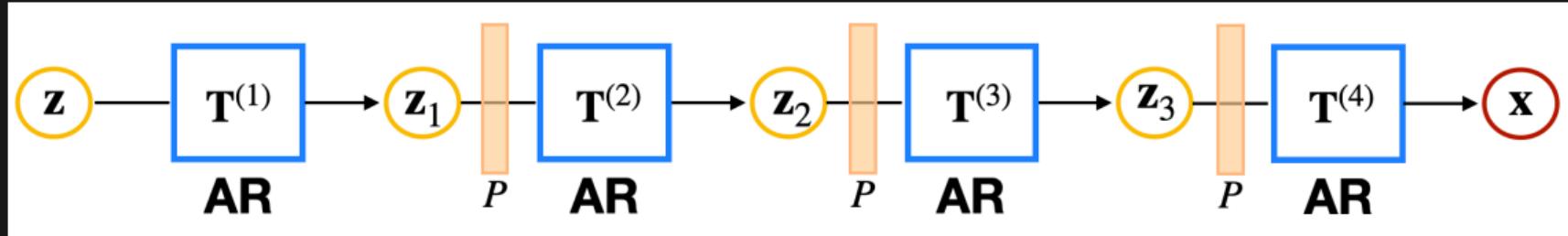


UNIVERSITY OF  
**WATERLOO**

FACULTY OF MATHEMATICS  
DAVID R. CHERITON SCHOOL  
OF COMPUTER SCIENCE

July 29, 2024

# Auto-Regressive (AR) Flow Recalled



$$(\mathbf{T}_\# r)(\mathbf{x}) = r(\mathbf{z}) / \det(\nabla \mathbf{T}^{(1)} \mathbf{z}) / \det(\nabla \mathbf{T}^{(2)} \mathbf{z}_1) / \det(\nabla \mathbf{T}^{(3)} \mathbf{z}_2) / \det(\nabla \mathbf{T}^{(4)} \mathbf{z}_3)$$

$$x_j = z_j \cdot \exp(\alpha_j(z_1, \dots, z_{j-1})) + \mu_j(z_1, \dots, z_{j-1}) =: T_j(z_1, \dots, z_{j-1}, z_j)$$

Now let the number of layers approach  $\infty$ !

# Neural Ordinary Differential Equations (ODE)

$$\mathbf{x}_{t+1} \approx \mathbf{x}_t + \eta_t \cdot \mathbf{f}_t(\mathbf{x}_t) =: \mathbf{T}_t(\mathbf{x}_t)$$
$$d\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t) dt$$

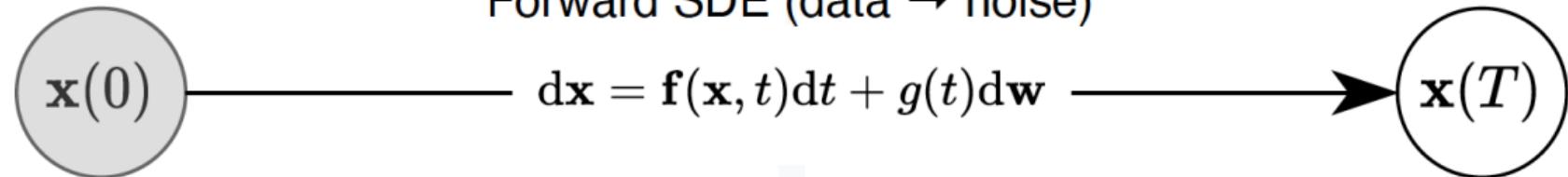
- Suppose  $\mathbf{x}_t \sim p_t$
- Apply change-of-variable-formula we know  $\mathbf{x}_{t+1} \sim p_{t+1}$ , where

$$\begin{aligned}\log p_{t+1}(\mathbf{x}_{t+1}) &= \log p_t(\mathbf{x}_t) - \log |\det \partial_{\mathbf{x}} \mathbf{T}_t(\mathbf{x}_t)| \\ &= \log p_t(\mathbf{x}_t) - \log |\det [\text{Id} + \eta_t \cdot \partial_{\mathbf{x}} \mathbf{f}_t(\mathbf{x}_t)]| \\ &\approx \log p_t(\mathbf{x}_t) - \eta_t \cdot \langle \partial_{\mathbf{x}}, \mathbf{f}_t(\mathbf{x}_t) \rangle\end{aligned}$$

- Continuous change-of-variable formula:

$$\boxed{\frac{d \log p_t(\mathbf{x}_t)}{dt} = - \langle \partial_{\mathbf{x}}, \mathbf{f}_t(\mathbf{x}_t) \rangle}$$

Forward SDE (data → noise)



score function

$\mathbf{x}(0)$

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})] dt + g(t)d\bar{\mathbf{w}}$$

$\mathbf{x}(T)$

Reverse SDE (noise → data)

# Stochastic Differential Equations (SDE)

$$d\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t) dt + g_t d\mathbf{w}_t$$

$$\mathbf{x}_{t+1} \approx \mathbf{x}_t + \eta_t \cdot \mathbf{f}_t(\mathbf{x}_t) + g_t \sqrt{\eta_t} \cdot \mathbf{z}, \quad \text{where } \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

- Intuitively,  $\mathbf{x}_{t+1}$  is now a deformed, **noisy** version of  $\mathbf{x}_t$ 
  - drift term  $\mathbf{f}_t$ : the infinitesimal change of  $E[\mathbf{x}_u | \mathbf{x}_t]$  when  $u \downarrow t$
  - diffusion term  $g_t$ : the infinitesimal change of  $\text{Var}[\mathbf{x}_u | \mathbf{x}_t]$  when  $u \downarrow t$

- Kolmogorov forward equation (a.k.a. Fokker-Planck equation):

$$\boxed{\partial_t p_t = - \langle \partial_{\mathbf{x}}, p_t \mathbf{f}_t \rangle + \frac{1}{2} g_t^2 \langle \partial_{\mathbf{x}} \partial_{\mathbf{x}}^\top, p_t \rangle}, \quad \mathbf{x}_t \sim p_t$$

- Kolmogorov backward equation (with fixed end time  $t > s$ ):

$$-\partial_s p_s = \langle \mathbf{f}_s, \partial_{\mathbf{x}} p_s \rangle + \frac{1}{2} g_t^2 \partial_{\mathbf{x}} \partial_{\mathbf{x}}^\top p_s$$

# ODE $\Leftrightarrow$ SDE

---

$$d\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t) dt$$

$$d\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t) dt + g_t d\mathbf{w}_t$$

- Any ODE is a (trivial) SDE with  $g_t \equiv \mathbf{0}$
- Conversely, any SDE is equivalent to an ODE:

$$\boxed{\mathbf{f}_t \leftarrow \mathbf{f}_t - \frac{1}{2} g_t^2 \partial_{\mathbf{x}} \log p_t}$$

- The **score function** plays an important role:

$$\boxed{\mathbf{s}(\mathbf{x}) = \mathbf{s}_p(\mathbf{x}) := \partial_{\mathbf{x}} \log p(\mathbf{x})}$$

# Marginal and Conditional Gaussians

- Marginal  $\mathbf{X}$  and conditional  $\mathbf{Y} \mid \mathbf{X}$  Gaussian:

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \Sigma), \quad \text{i.e., } \mathbf{X} = \boldsymbol{\mu} + \Sigma^{1/2} \mathbf{Z}_1$$

$$p(\mathbf{y} \mid \mathbf{x}) = \mathcal{N}(\mathbf{y} \mid A\mathbf{x} + \mathbf{b}, S), \quad \text{i.e., } \mathbf{Y} = A\mathbf{X} + \mathbf{b} + S^{1/2} \mathbf{Z}_2,$$

- Equivalent to joint Gaussian:

$$p(\mathbf{x}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ A\boldsymbol{\mu} + \mathbf{b} \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma A^\top \\ A\Sigma & A\Sigma A^\top + S \end{bmatrix}\right)$$

- Refactorize into marginal  $\mathbf{Y}$  and conditional  $\mathbf{X} \mid \mathbf{Y}$ , again Gaussian

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y} \mid A\boldsymbol{\mu} + \mathbf{b}, A\Sigma A^\top + S), \quad \text{i.e., } \mathbf{Y} = A\boldsymbol{\mu} + \mathbf{b} + A\Sigma^{1/2} \mathbf{Z}_1 + S^{1/2} \mathbf{Z}_2$$

$$p(\mathbf{x} \mid \mathbf{y}) = \mathcal{N}\left(\mathbf{x} \mid \Xi(\Sigma^{-1}\boldsymbol{\mu} + A^\top S^{-1}(\mathbf{y} - \mathbf{b})), \Xi\right), \quad \text{where } \Xi = (\Sigma^{-1} + A^\top S^{-1} A)^{-1}$$

- Linear transformation of Gaussian is still Gaussian:

$$[\mathbf{X} - \Sigma A^\top (A\Sigma A^\top + S)^{-1} \mathbf{Y}] \perp\!\!\!\perp \mathbf{Y}$$

– simply verify the covariance is zero

- Mean and covariance determine a Gaussian:

$$\begin{aligned}\mathbb{E}[\mathbf{X} | \mathbf{Y}] &= \mathbb{E}[\mathbf{X} - \Sigma A^\top (A\Sigma A^\top + S)^{-1} \mathbf{Y} | \mathbf{Y}] + \Sigma A^\top (A\Sigma A^\top + S)^{-1} \mathbf{Y} \\ &= \mathbb{E}[\mathbf{X} - \Sigma A^\top (A\Sigma A^\top + S)^{-1} \mathbf{Y}] + \Sigma A^\top (A\Sigma A^\top + S)^{-1} \mathbf{Y} \\ &= \boldsymbol{\mu} - \Sigma A^\top (A\Sigma A^\top + S)^{-1} (A\boldsymbol{\mu} + \mathbf{b}) + \Sigma A^\top (A\Sigma A^\top + S)^{-1} \mathbf{Y} \\ &= (\Sigma - \Sigma A^\top (A\Sigma A^\top + S)^{-1} A\Sigma) \Sigma^{-1} \boldsymbol{\mu} + \Sigma A^\top (A\Sigma A^\top + S)^{-1} (\mathbf{Y} - \mathbf{b})\end{aligned}$$

$$\begin{aligned}\text{Var}[\mathbf{X} | \mathbf{Y}] &= \text{Var}[\mathbf{X} - \Sigma A^\top (A\Sigma A^\top + S)^{-1} \mathbf{Y} | \mathbf{Y}] \\ &= \text{Var}[\mathbf{X} - \Sigma A^\top (A\Sigma A^\top + S)^{-1} \mathbf{Y}] \\ &= \Sigma - 2\Sigma A^\top (A\Sigma A^\top + S)^{-1} A\Sigma + \Sigma A^\top (A\Sigma A^\top + S)^{-1} A\Sigma \\ &= (\Sigma^{-1} + A^\top S^{-1} A)^{-1}\end{aligned}$$

– Sherman-Morrison-Woodbury identity

# Reverse-time SDE

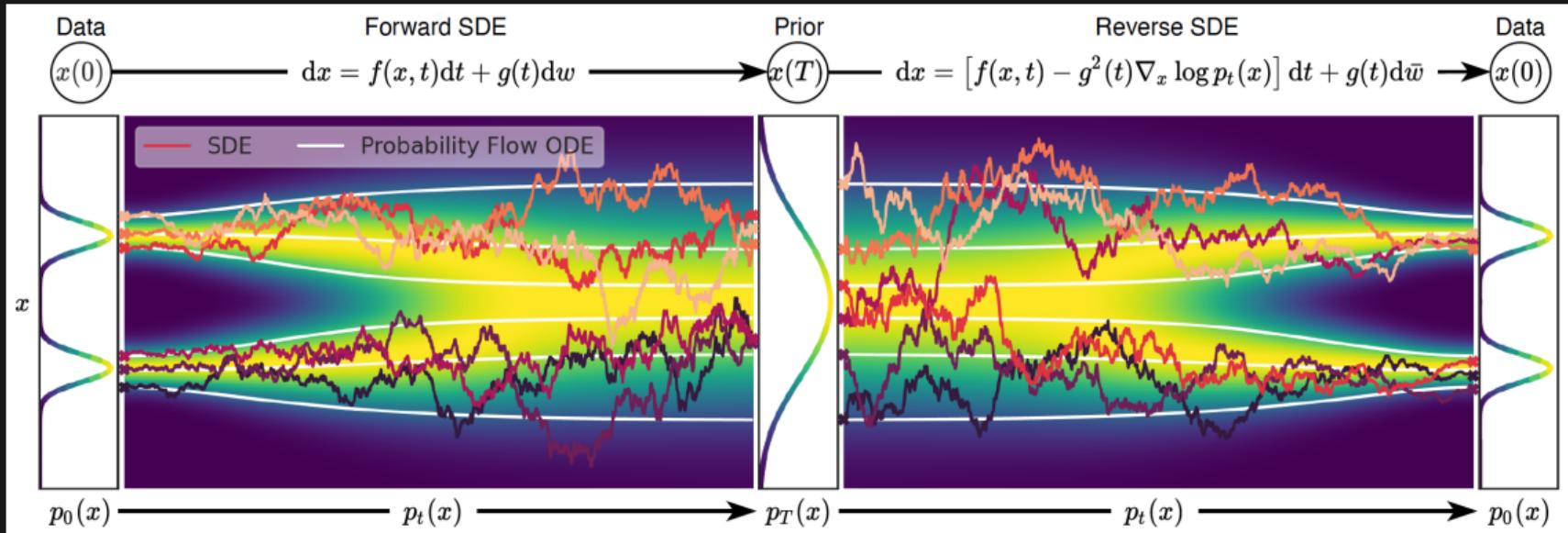
$$\begin{aligned} d\mathbf{x}_{t+1} &= \mathbf{f}_t(\mathbf{x}_t) dt + g_t d\mathbf{w}_t \\ d\bar{\mathbf{x}}_{t+1} &= \bar{\mathbf{f}}_t(\bar{\mathbf{x}}_t) dt + g_t d\bar{\mathbf{w}}_t, \quad \text{where} \end{aligned}$$

$$\boxed{\bar{\mathbf{f}}_t = \mathbf{f}_t - g_t^2 \partial_{\mathbf{x}} \log p_t}$$

- Time flows backwards for the bar quantities
- Forward SDE: diffuses date into noise
- Reverse SDE: molds noise into data
- $\mathbf{f}_t$  and  $G_t$  together specify  $\bar{\mathbf{f}}_t$ : key is to estimate the score  $\partial_{\mathbf{x}} \log p_t$

---

B. D. O. Anderson. "Reverse-time diffusion equation models". *Stochastic Processes and their Applications*, vol. 12, no. 3 (1982), pp. 313–326.



Y. Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021.

# Score Matching

$$\begin{aligned}\mathbb{F}(p\|q) &:= \frac{1}{2} \mathbb{E}_{\mathbf{X} \sim q} \|\partial_{\mathbf{x}} \log p(\mathbf{X}) - \partial_{\mathbf{x}} \log q(\mathbf{X})\|_2^2 \\ &= \mathbb{E}_{\mathbf{X} \sim q} \left[ \frac{1}{2} \|\mathbf{s}_p(\mathbf{X})\|_2^2 + \langle \partial_{\mathbf{x}}, \mathbf{s}_p(\mathbf{X}) \rangle + \frac{1}{2} \|\mathbf{s}_q(\mathbf{X})\|_2^2 \right] \\ &\approx \hat{\mathbb{E}}_{\mathbf{X} \sim q} \left[ \frac{1}{2} \|\mathbf{s}_p(\mathbf{X})\|_2^2 + \langle \partial_{\mathbf{x}}, \mathbf{s}_p(\mathbf{X}) \rangle \right]\end{aligned}$$

- Under mild conditions,  $\mathbb{F}(p\|q) = 0 \iff p \propto q$
- A Convenient way to estimate the score  $\mathbf{s}_q$  and hence the density  $q$
- The model score function  $\mathbf{s}_p$  can be chosen as a neural net

# Denoising Auto-Encoder

- Suppose also have a latent variable  $Z$  with joint density  $q(\mathbf{x}, \mathbf{z})$
- Exchange differentiation with integration we obtain:

$$\begin{aligned}\mathbb{F}(p\|q) &:= \frac{1}{2} \mathbb{E}_{\mathbf{x} \sim q} \|\partial_{\mathbf{x}} \log p(\mathbf{X}) - \partial_{\mathbf{x}} \log q(\mathbf{X})\|_2^2 \\ &= \frac{1}{2} \mathbb{E}_{(\mathbf{x}, \mathbf{z}) \sim q} [\|\mathbf{s}_p(\mathbf{X}) - \partial_{\mathbf{x}} \log q(\mathbf{X}|\mathbf{Z})\|_2^2 + \|\mathbf{s}_q(\mathbf{X})\|_2^2 - \|\partial_{\mathbf{x}} \log q(\mathbf{X}|\mathbf{Z})\|_2^2] \\ &\approx \frac{1}{2} \hat{\mathbb{E}}_{(\mathbf{x}, \mathbf{z}) \sim q} \|\mathbf{s}_p(\mathbf{X}) - \partial_{\mathbf{x}} \log q(\mathbf{X}|\mathbf{Z})\|_2^2\end{aligned}$$

- Useful when the conditional density  $\partial_{\mathbf{x}} \log q(\mathbf{X}|\mathbf{Z})$  is easy to obtain

# Score-based Diffusion Generative Models

$$d\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t) dt + g_t d\mathbf{w}_t$$

$$\mathbf{x}_{t+1} \approx \mathbf{x}_t + \eta_t \cdot \mathbf{f}_t(\mathbf{x}_t) + g_t \sqrt{\eta_t} \cdot \mathbf{z}, \quad \text{where } \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

- Key is to estimate the score  $\mathbf{s}_t(\mathbf{x}) = \partial_{\mathbf{x}} \log p_t$
- Apply denoising auto-encoder score matching:

$$\min_{\boldsymbol{\theta}} \hat{\mathbb{E}}_{t \sim \mu, (\mathbf{X}_t, \mathbf{X}_0) \sim q(\mathbf{x}_t, \mathbf{x}_0)} \lambda_t \|\mathbf{s}_t(\mathbf{X}_t; \boldsymbol{\theta}) - \partial_{\mathbf{x}} \log q(\mathbf{X}_t | \mathbf{X}_0)\|_2^2$$

- $\mathbf{X}_0 \sim q(\mathbf{x})$ , the data density
- $q(\mathbf{x}_t | \mathbf{x}_0)$  can be derived from the forward SDE, in closed-form if  $\mathbf{f}_t$  is affine

# Inference After Learning

$$d\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t) dt + g_t d\mathbf{w}_t$$

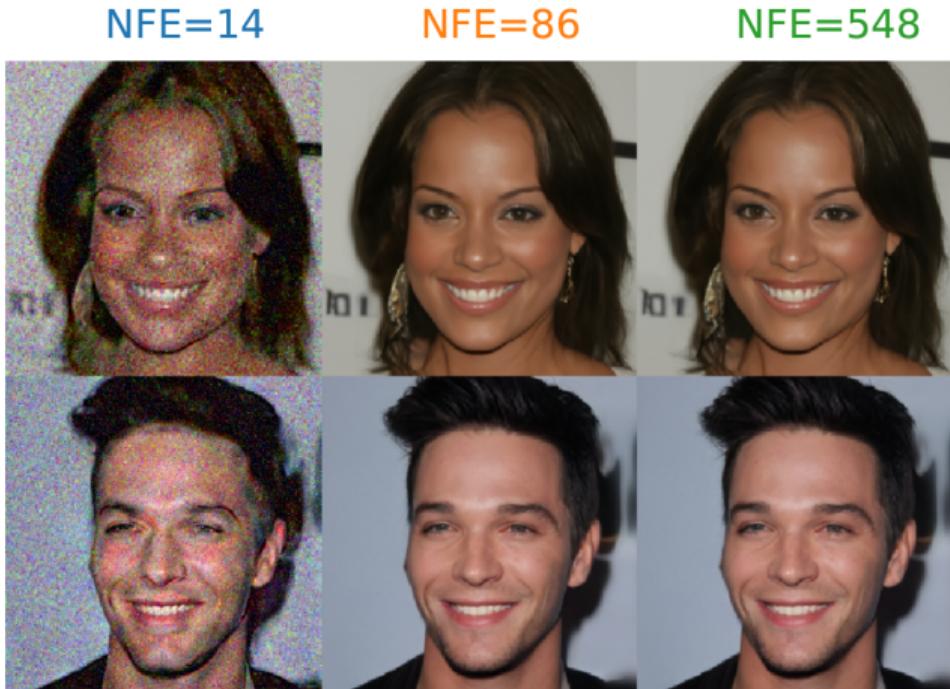
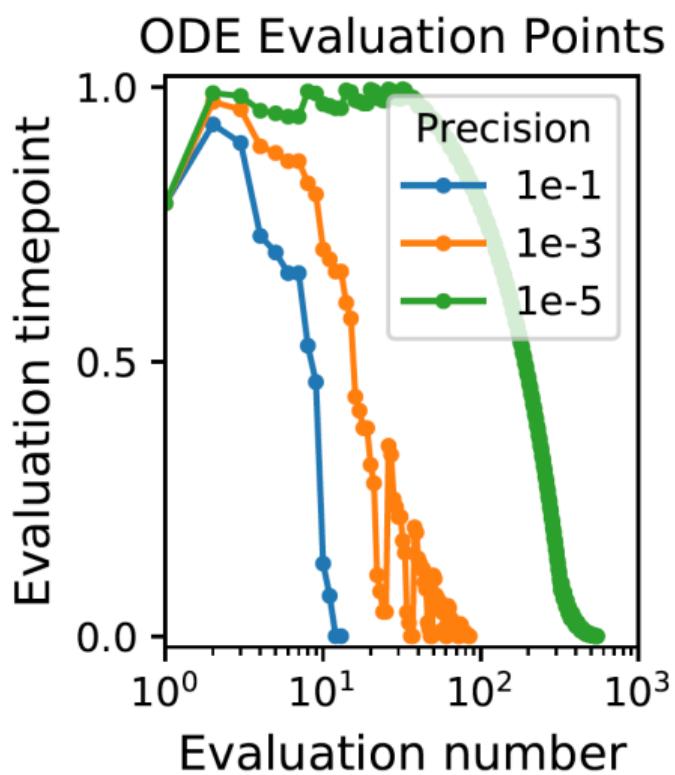
$$d\bar{\mathbf{x}}_{t+1} = \boxed{\mathbf{f}_t - g_t^2 \mathbf{s}_t(\bar{\mathbf{x}}_t; \boldsymbol{\theta})} dt + g_t d\bar{\mathbf{w}}_t$$

$$d\mathbf{x}_{t+1} = \boxed{\mathbf{f}_t - \frac{1}{2}g_t^2 \mathbf{s}_t(\mathbf{x}_t; \boldsymbol{\theta})} dt$$

- Run the reverse SDE or the equivalent ODE
  - sample  $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \text{Id})$
  - apply numerical SDE or ODE solver (e.g., Euler-Maruyama)

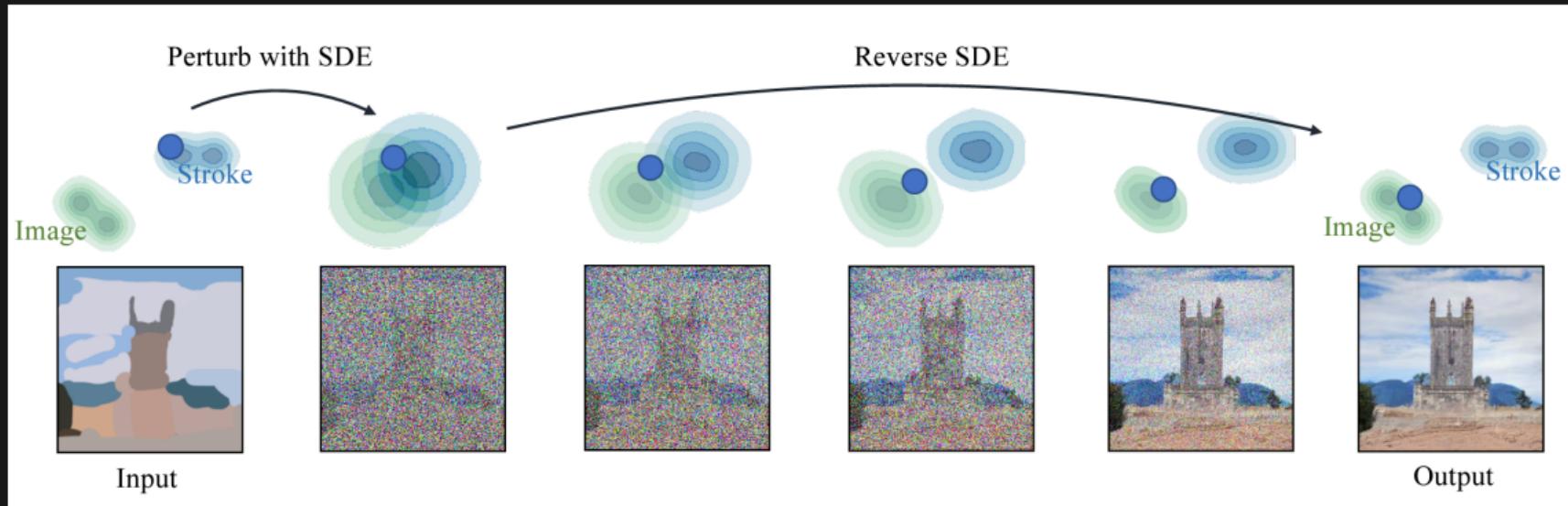
---

D. J. Higham. "An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations". *SIAM Review*, vol. 43, no. 3 (2001), pp. 525–546.



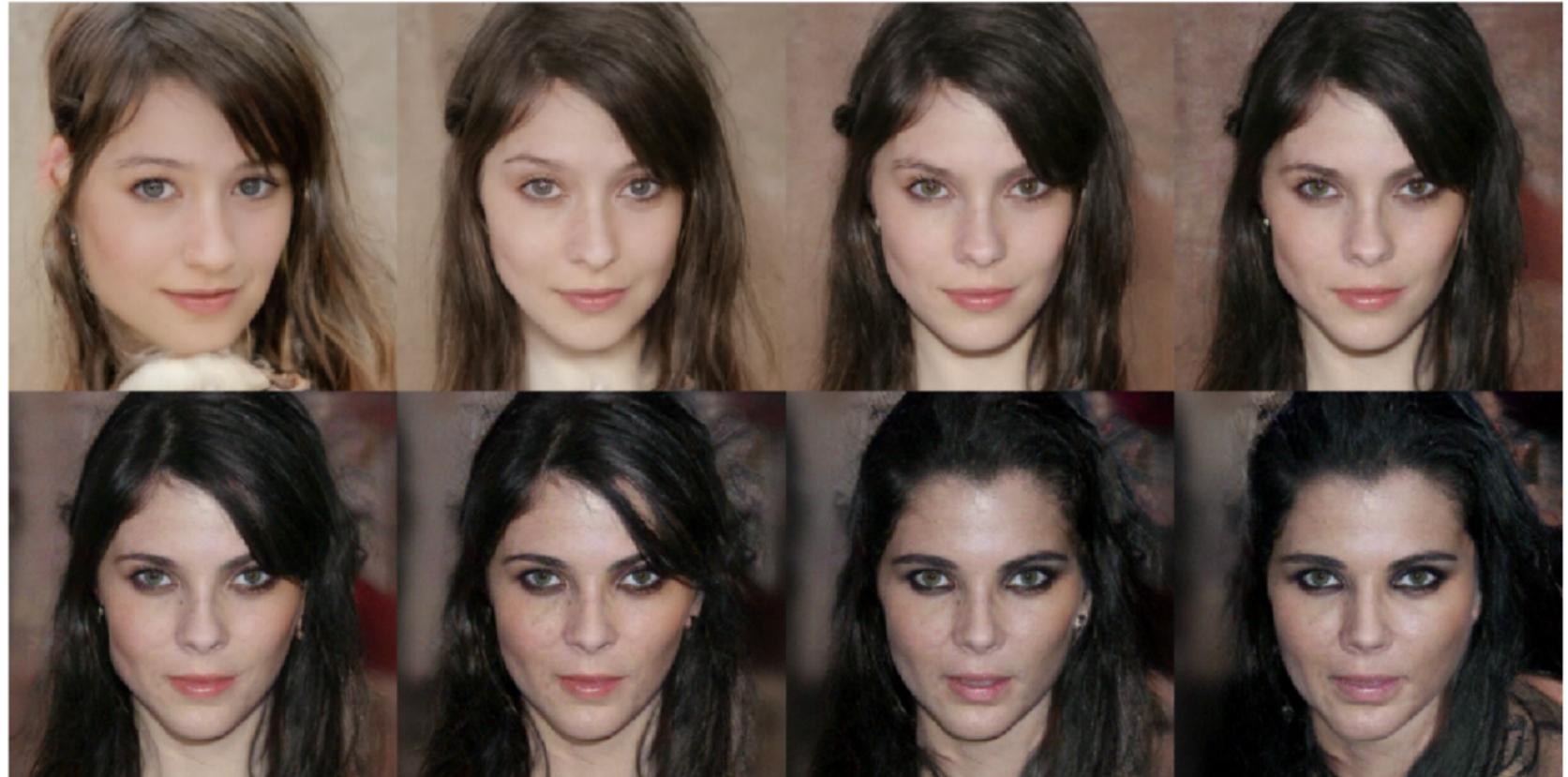


# SDEdit

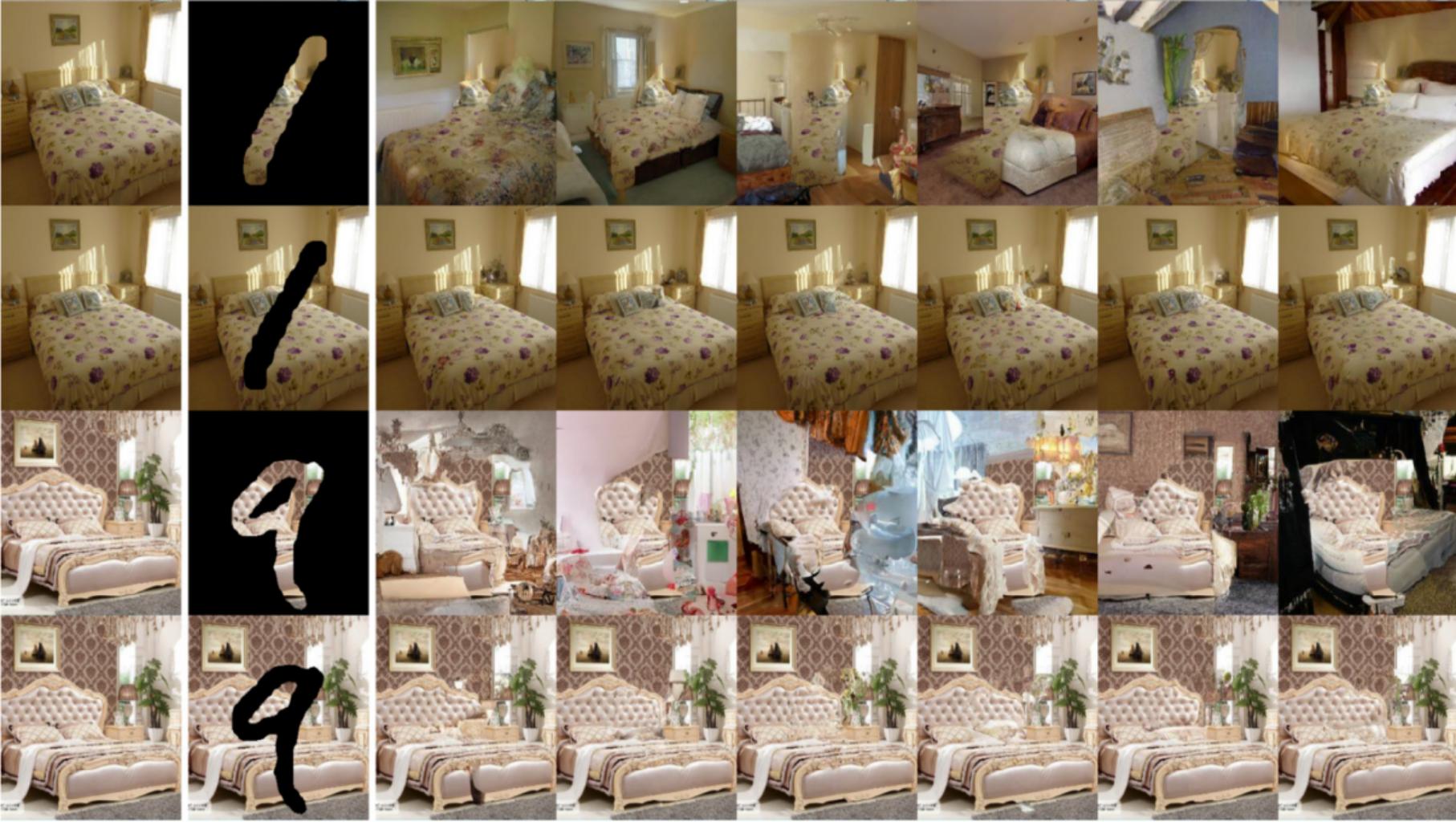


C. Meng et al. "SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2022.

# Interpolation

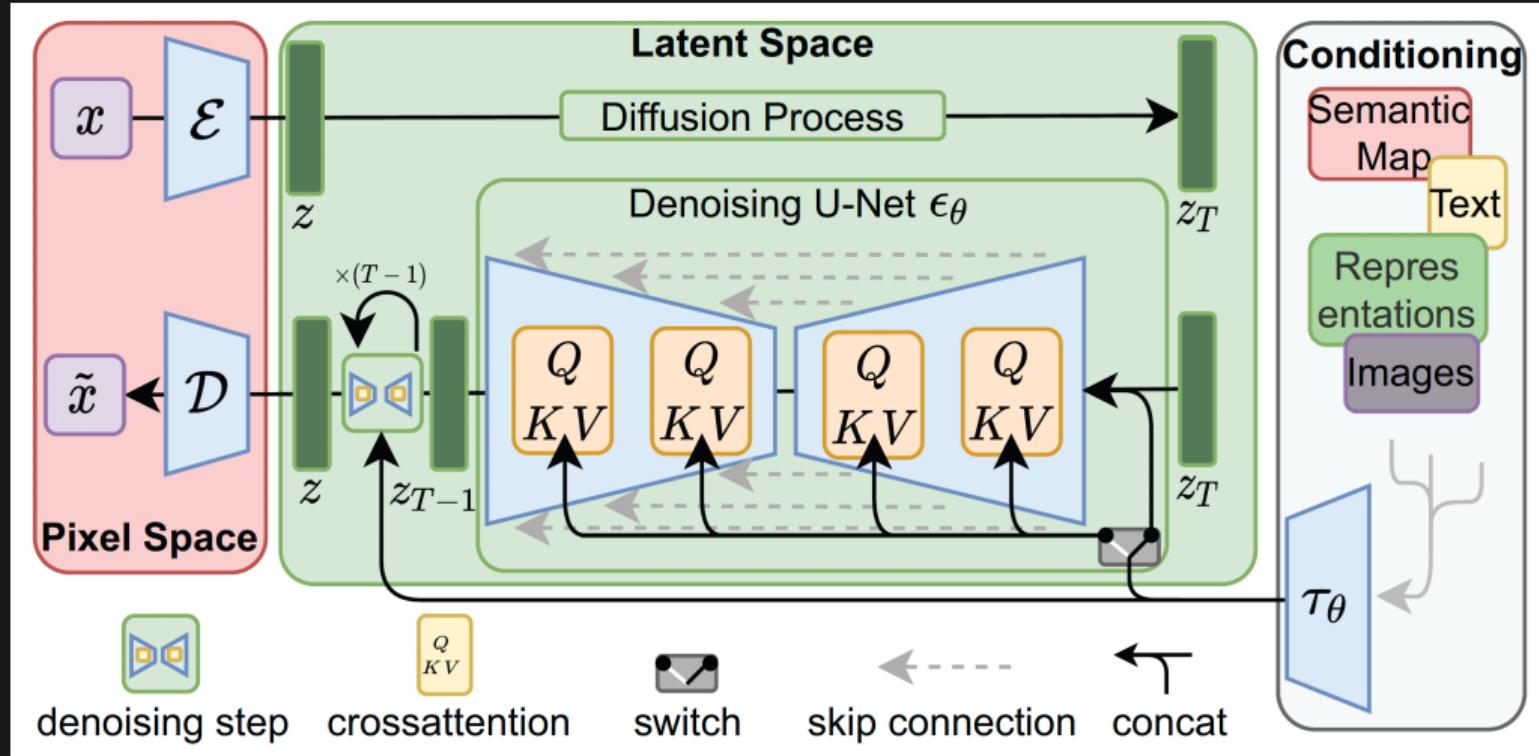


Y. Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021.





# Stable Diffusion



R. Rombach et al. "High-Resolution Image Synthesis with Latent Diffusion Models". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10674–10685.

