Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling

Arnaud Doucet

with Valentin De Bortoli, James Thornton & Jeremy Heng

Department of Statistics, Oxford University

CIRM - 27th September 2021

Generative Modeling using Generative Adversarial Networks

Progress on face generation using GANs (source: www.medium.com)

- Applications: inverse problems (denoising, inpainting, super-resolution), compression, structure prediction (proteins & molecules) and neural network pretraining.

Generative Modeling

- Massive advances in generative modeling driven by VAEs (Kingma & Welling, 2014; Rezende, Mohamed & Wiestra, 2014), GANs (Goodfellow et al., 2014), autoregressive models (van den Oord et al., 2016).

- Score-based generative models aka denoising diffusion models were proposed by Sohl-Dickstein et al. (2015) but have only become popular recently (Ho et al., 2020; Song et al., 2021).

- Score-based generative models exhibit SOTA performance on several audio and image synthesis tasks; see e.g. (Ho et al., NeurIPS 2020), (Song et al., ICLR 2021) & (Dhariwal & Nichol, arXiv:2105.05233).

- Score-based algorithms are SOTA when solving Bayesian inverse problems for imaging; see e.g. (Laumont et al., 2020; Kadkhodaie & Simoncelli, 2020; Kawar et al., 2021).

Generative Modeling using Diffusion Models

Diffusion Models Beat GANs on Image Synthesis - OpenAI, 2021

Markov chains 101

- Consider a Markov chain with $X_0 \sim p_0$ and $X_{k+1} \sim p_{k+1|k}(\cdot|X_k)$ then

$$p(x_{0:N}) = p_0(x_0) \prod_{k=0}^{N-1} p_{k+1|k}(x_{k+1}|x_k)$$

- One has the *backward* decomposition

$$p(x_{0:N}) = p_N(x_N) \prod_{k=0}^{N-1} p_{k|k+1}(x_k|x_{k+1}), \text{for } p_{k|k+1}(x_k|x_{k+1}) = \frac{p_k(x_k)p_{k+1|k}(x_{k+1}|x_k)}{p_{k+1}(x_{k+1})}$$

where $p_k(x_k)$ denotes the marginal of X_k satisfying

$$p_k(x_k) = \int p_{k|k-1}(x_k|x_{k-1})p_{k-1}(x_{k-1})\mathrm{d}x_{k-1}$$

- One can sample from $p(x_{0:N})$ by *ancestral sampling*

Sample $X_N \sim p_N(\cdot)$ then $X_k \sim p_{k|k+1}(\cdot|X_{k+1})$ for $k = N-1, ..., 0$

Application to Generative Modeling

- For generative modeling, we let $p_0 = p_{\text{data}}$ and set $p_{k+1|k}$ such that $p_N \approx p_{\text{prior}}$ for $N \gg 1$ where $p_{\text{prior}} = \mathcal{N}(x; 0_d, I_d)$ is a "prior" easy-to-sample density.

- Pick for $p_{k+1|k}$ a MCMC kernel that is p_{prior}-invariant so that $p_N(x) \approx p_{\text{prior}}(x)$ for N large enough

$$X_{k+1} = \alpha X_k + \sqrt{1-\alpha^2}\,\epsilon_{k+1}, \quad \epsilon_{k+1} \sim \mathcal{N}(0_d, I_d);$$

  i.e. add noise!

- Use ancestral sampling but replace p_N by $p_{\text{prior}} \approx p_N$ for new sample generation, i.e.

$$\text{Sample } X_N \sim p_{\text{prior}}(\cdot) \text{ then } X_k \sim p_{k|k+1}(\cdot|X_{k+1}) \text{ for } k = N-1, ..., 0$$

- **Key Problem**: One needs to approximate the backward transitions $p_{k|k+1}$, i.e. learn to denoise.

Approximating Backward Transitions

- We restrict ourselves to

$$p_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; x_k + \gamma f(x_k), 2\gamma I_d),$$

- Using $p_k \approx p_{k+1}$, a Taylor expansion of $\log p_{k+1}$ at x_k and $f(x_k) \approx f(x_{k+1})$ for $||x_{k+1} - x_k|| = o(1)$

$$\begin{aligned} p_{k|k+1}(x_k|x_{k+1}) &= p_{k+1|k}(x_{k+1}|x_k) \exp[\log p_k(x_k) - \log p_{k+1}(x_{k+1})] \\ &\approx \mathcal{N}(x_k; x_{k+1} - \gamma f(x_{k+1}) + 2\gamma \underbrace{\nabla \log p_{k+1}(x_{k+1})}_{\text{"score"}}, 2\gamma I_d). \end{aligned}$$

- The score is not available but $p_{k+1}(x_{k+1}) = \int p_0(x_0) p_{k+1|0}(x_{k+1}|x_0) \mathrm{d}x_0$ and we have a Fisher's like identity

$$\nabla \log p_{k+1}(x_{k+1}) = \mathbb{E}_{p_{\mathbf{0}|k+\mathbf{1}}}[\nabla_{x_{k+\mathbf{1}}} \log p_{k+1|0}(x_{k+1}|X_0)].$$

Estimating the Scores using Score Matching

- The score can be estimated by regression, i.e.

$$s_{k+1} = \arg\min_s \ \mathbb{E}_{p_{\mathbf{0},k+\mathbf{1}}}[||s(X_{k+1}) - \nabla_{x_{k+\mathbf{1}}} \log p_{k+1|0}(X_{k+1}|X_0)||^2].$$

- In practice, we restrict ourselves to neural networks and estimate all scores simultaneously i.e. $s_{\theta^\star}(k, x_k) \approx \nabla \log p_k(x_k)$ where

$$\theta^\star \approx \arg\min_\theta \sum_{k=1}^{N} \mathbb{E}_{p_{\mathbf{0},k}}[||s_\theta(k, X_k) - \nabla_{x_k} \log p_{k|0}(X_k|X_0)||^2].$$

- If $p_{k+1|0}(x_{k+1}|x_0)$ is not available, then use

$$\nabla \log p_{k+1}(x_{k+1}) = \mathbb{E}_{p_{k|k+\mathbf{1}}}[\nabla_{x_{k+\mathbf{1}}} \log p_{k+1|k}(x_{k+1}|X_k)].$$

Recap

- Use noisy samples from data to train a neural network such that

$$s_{\theta^\star}(k, x_k) \approx \nabla \log p_k(x_k).$$

- Generate new samples using $X_N \sim p_{\text{prior}}$ then

$$X_k = X_{k+1} - \gamma f(X_{k+1}) + 2\gamma_{k+1} s_{\theta^\star}(k+1, X_{k+1}) + \sqrt{2\gamma} Z_{k+1}, \quad Z_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0_d, I_d).$$

We let $\{Y_k\}_{k=0}^N = \{X_{N-k}\}_{k=0}^N$ which satisfies the forward recursion $Y_0 \sim p_{\text{prior}}$

$$Y_{k+1} = Y_k - \gamma f(Y_k) + 2\gamma_{k+1} s_{\theta^\star}(N-k, Y_k) + \sqrt{2\gamma} Z_{k+1}.$$

- Variational inference formulation in (Ho et al., 2020); i.e. minimize w.r.t. θ $\text{KL}(\text{forward noising}||\text{backward denoising}_\theta)$.

From Discrete to Continuous-Time (Song et al., 2021)

- The dynamics $p_{k+1|k}(x'|x) = \mathcal{N}(x'; x + \gamma f(x), 2\gamma I_d)$ is an Euler discretization of

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t, \quad X_0 \sim p_{\text{data}}.$$

- For $f(x) = 0$, it is a Brownian motion ($p_{\text{prior}}(x) = \mathcal{N}(x; 0_d, 2T)$) and for $f(x) = \alpha x$ an OU process ($p_{\text{prior}}(x) = \mathcal{N}(x; 0_d, \alpha^{-1} I_d)$).

- The reverse-time process $(Y_t)_{t\in[0,T]} = (X_{T-t})_{t\in[0,T]}$ satisfies

$$\mathrm{d}Y_t = \{-f(Y_t) + 2\nabla \log p_{T-t}(Y_t)\}\mathrm{d}t + \sqrt{2}\mathrm{d}B_t, \quad Y_0 \sim p_T.$$

- The generative model $(Y_t)_{t\in[0,T]}$ satisfies

$$\mathrm{d}Y_t = \{-f(Y_t) + 2\nabla \log s_{\theta^*}(T-t, Y_t)\}\mathrm{d}t + \sqrt{2}\mathrm{d}B_t, \quad Y_0 \sim p_{\text{prior}}.$$

From Discrete to Continuous-Time

- Assume there exists $M \geq 0$ such that for any $t \in [0, T]$ and $x \in \mathbb{R}^d$

$$||s_{\theta^\star}(t, x) - \nabla \log p_t(x)|| \leq M,$$

with $s_{\theta^\star} \in C([0, T] \times \mathbb{R}^d, \mathbb{R}^d)$ and regularity conditions on p_{data} and its gradients.

- Then there exist $0 \geq B_\alpha, C_\alpha, D_\alpha < \infty$ s.t. for any N and $\{\gamma_k\}_{k=1}^N$ the following hold:

$$\text{For } \alpha > 0, ||\mathcal{L}(X_0) - p_{\text{data}}|| \leq B_\alpha \exp[-\alpha^{1/2} T] + C_\alpha (M + \bar{\gamma}^{1/2}) \exp[D_\alpha T]$$
$$\text{For } \alpha = 0, ||\mathcal{L}(X_0) - p_{\text{data}}|| \leq B_0 (T^{-1} + T^{-1/2}) + C_0 (M + \bar{\gamma}^{1/2}) \exp[D_0 T];$$

where $T = \sum_{k=1}^N \gamma_k$, $\bar{\gamma} = \sup_{k \in \{1,\dots,N\}} \gamma_k$

- First term on r.h.s. bound is error between p_T and p_{prior} and decreases with T. Second term is error between continuous-time processes and approximation, increases with α and T.

When Diffusion Models Fail

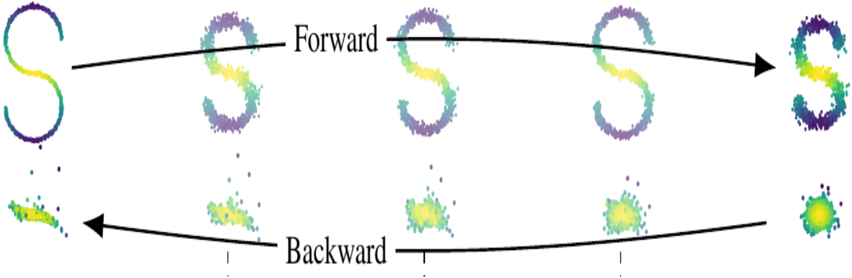


Illustration of failure on toy 2-D example: N is too small so p_N is very different from p_{prior}. Hence the reverse diffusion initialized according to p_{prior} provides samples at time 0 very different from p_{data}

Revisiting Generative Modeling using Schrödinger Bridges

- Consider a *reference* density $p(x_{0:N})$, find $\pi^\star(x_{0:N})$ such that

$$\pi^\star = \arg\min\{\mathrm{KL}(\pi||p) : \pi_0 = p_{\text{data}},\ \pi_N = p_{\text{prior}}\}.$$

- Using notation $\mu(x_{0:N}) := \mu_{0,N}(x_0, x_N)\mu_{|0,N}(x_{1:N-1}|x_0, x_N)$, one has

$$\mathrm{KL}(\pi||p) = \mathrm{KL}(\pi_{0,N}|p_{0,N}) + \mathbb{E}_{\pi_{\mathbf{0},N}}[\mathrm{KL}(\pi_{|0,N}||p_{|0,N})]$$

so $\pi^\star(x_{0:N}) = \pi^{\mathrm{s},\star}(x_0, x_N)p_{|0,N}(x_{1:N-1}|x_0, x_N)$ where $\pi^{\mathrm{s},\star}(x_0, x_N)$ solves

$$\pi^{\mathrm{s},\star} = \arg\min\{\mathrm{KL}(\pi^{\mathrm{s}}||p_{0,N}) : \pi_0^{\mathrm{s}} = p_{\text{data}},\ \pi_N^{\mathrm{s}} = p_{\text{prior}}\}.$$

- If $p_{N|0}(x_N|x_0) = \mathcal{N}(x_N; x_0, \sigma^2)$, this is an entropy-regularized OT problem

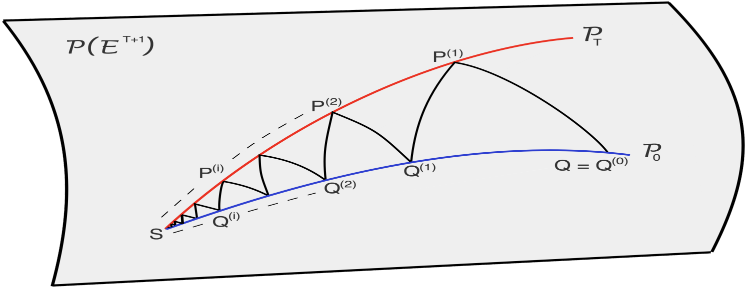
$$\pi^{\mathrm{s},\star} = \arg\min\{\mathbb{E}_{\pi^{\mathrm{s}}}[||X_0 - X_N||^2] - 2\sigma^2 H(\pi^{\mathrm{s}}) : \pi_0^{\mathrm{s}} = p_{\text{data}},\ \pi_N^{\mathrm{s}} = p_{\text{prior}}\}.$$

- Schrödinger Bridge can be solved using Iterative Proportional Fitting (Schrödinger 1932; Fortet, 1940; Sinkhorn, 1967; Kullback, 1968): *Plus ça change, plus c'est la même chose*.

Solving the Schrödinger Bridge Problem

- Iterative Proportional Fitting (IPF): set $\pi^(0) = p$ and for $n \geq 1$

$$Q^{(n)} := \pi^{(2n+1)} = \arg\min\{\mathrm{KL}(\pi||\pi^{(2n)}), \quad \pi_N = p_{\text{prior}}\},$$
$$P^{(n)} := \pi^{(2n+2)} = \arg\min\{\mathrm{KL}(\pi||\pi^{(2n+1)}), \quad \pi_0 = p_{\text{data}}\}.$$



Alternating projections $Q^{(n)}$ with marginal p_{prior} and $P^{(n)}$ with marginal p_{data} converge towards the Schrödinger bridge (Fortet, 1940; Kullback, 1968; Rüschendorf, 1995; Léger, 2021; De Bortoli et al., 2021).

Solving the Schrödinger Bridge Problem

- First IPF step requires solving $\pi^{(1)} = \arg\min\{\mathrm{KL}(\pi||\pi^{(0)}), \ \ \pi_N = p_{\text{prior}}\}$ but as $\pi^{(0)} = p$
$$\mathrm{KL}(\pi||\pi^{(0)}) = \mathrm{KL}(\pi_N|p_N) + \mathbb{E}_{\pi_N}[\mathrm{KL}(\pi_{|N}||p_{|N})]$$
so
$$\pi^{(1)}(x_{0:N}) = p_{\text{prior}}(x_N)p(x_{0:N-1}|x_N) = p_{\text{prior}}(x_N)\prod_{k=N-1}^{0} p_{k|k+1}(x_k|x_{k+1})$$
- Approximation to first iteration of IPF corresponds to existing Score-Based Generative models!
- Second IPF step requires solving $\pi^{(2)} = \arg\min\{\mathrm{KL}(\pi||\pi^{(1)}), \ \ \pi_0 = p_{\text{data}}\}$ but
$$\mathrm{KL}(\pi||\pi^{(1)}) = \mathrm{KL}(\pi_0|\pi_0^{(1)}) + \mathbb{E}_{\pi_0}[\mathrm{KL}(\pi_{|0}||\pi_{|0}^{(1)})]$$
so
$$\pi^{(2)}(x_{0:N}) = p_{\text{data}}(x_0)\pi^{(1)}(x_{1:N}|x_0) = p_{\text{data}}(x_0)\prod_{k=1}^{N} \pi_{k+1|k}^{(1)}(x_{k+1}|x_k)$$

Solving the Schrödinger Bridge Problem

- In 1st iter, the backward dynamics of the forward process $\pi^{(0)} = p$ is initialized by p_{prior} at time N to define the backward process $\pi^{(1)}$.

- In 2nd iter, the forward dynamics of the backward process $\pi^{(1)}$ is initialized by p_{data} at time 0 to define the forward process $\pi^{(2)}$.

- In 3rd iteration, the backward dynamics of the forward process $\pi^{(2)}$ is initialized by p_{prior} at time N to define the backward process $\pi^{(3)}$.

- Loosely speaking, we use score matching ideas at each iteration to learn the scores of the forward or backward process.

Continuous-Time IPF

- IPF can be formulated in continuous time

$$\Pi^\star = \arg\min\{\mathrm{KL}(\Pi||\mathbb{P}) : \Pi \in \mathcal{P}(\mathcal{C}), \Pi_0 = p_{\text{data}}, \Pi_T = p_{\text{prior}}\}.$$

Similarly, we define the IPF $(\Pi^{(n)})$ recursively $\Pi^0 = \mathcal{P}$ using

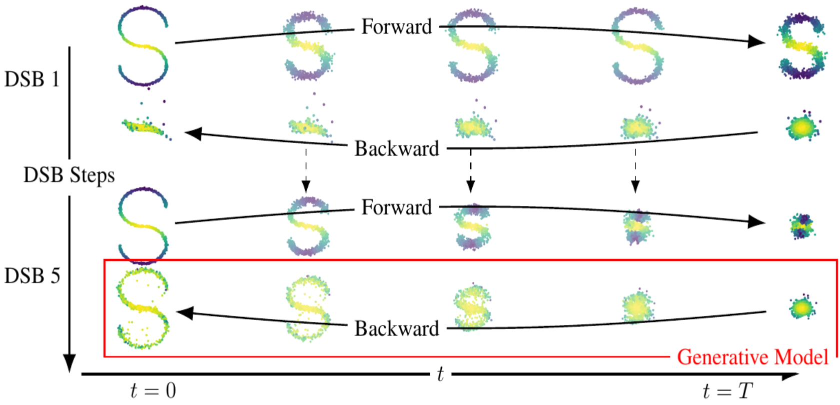
$$\Pi^{(2n+1)} = \arg\min\{\mathrm{KL}(\Pi||\Pi^{(2n)}) : \Pi \in \mathcal{P}(\mathcal{C}), \Pi_T = p_{\text{prior}}\},$$
$$\Pi^{(2n+2)} = \arg\min\{\mathrm{KL}(\Pi||\Pi^{(2n+1)}) : \Pi \in \mathcal{P}(\mathcal{C}), \Pi_0 = p_{\text{data}}\}.$$

- Under regularity conditions, then

$$(\Pi^{(2n+1)})^R : \mathrm{d}Y_t^{(2n+1)} = b_{T-t}^{(n)}(Y_t^{(2n+1)})\mathrm{d}t + \sqrt{2}\mathrm{d}B_t, Y_0^{(2n+1)} \sim p_{\text{prior}};$$
$$\Pi^{(2n+2)} : \mathrm{d}X_t^{(2n+2)} = f_t^{(n+1)}(X_t^{(2n+2)})\mathrm{d}t + \sqrt{2}\mathrm{d}B_t, X_0^{(2n+2)} \sim p_{\text{data}};$$

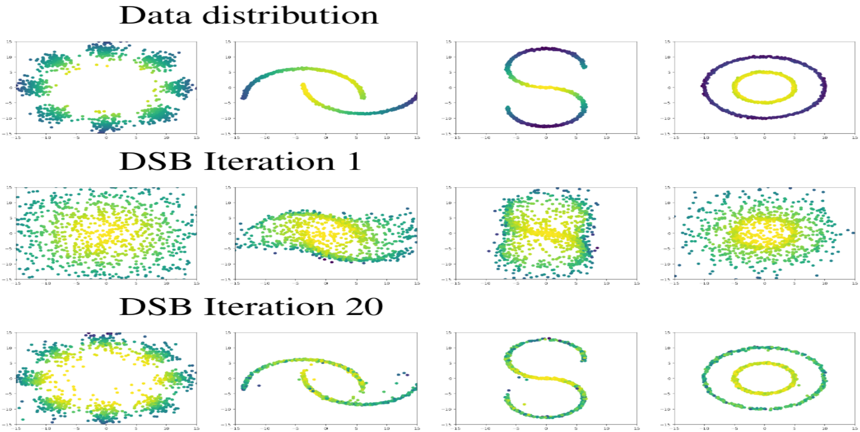
for $b_t^{(n)}(x) = -f_t^{(n)}(x) + 2\nabla \log p_t^{(n)}(x)$, $f_t^{(n+1)}(x) = -b_t^{(n)}(x) + 2\nabla \log q_t^{(n)}(x)$, with $f_t^{(0)}(x) = f(x)$, and $p_t^{(n)}$, $q_t^{(n)}$ the densities of $\Pi_t^{(2n)}$ and $\Pi_t^{(2n+1)}$.

Illustration of Diffusion Schrödinger Bridge



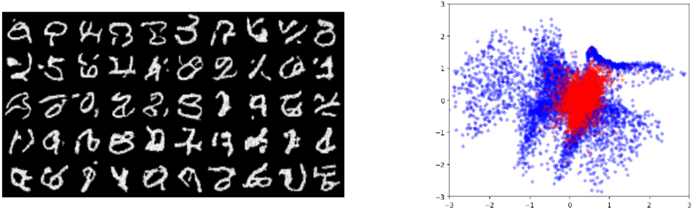
Revisiting 2-D toy example with Diffusion Schrödinger Bridge. After 5 iterations, we obtain a satisfactory generative model.

Applications: 2-D distributions

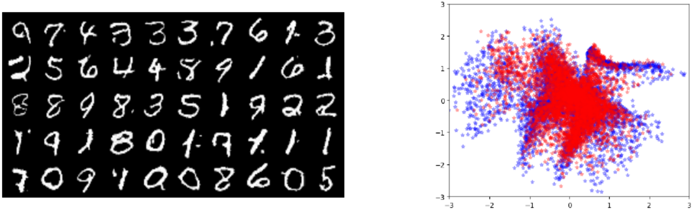


Data distributions p_{data} vs distribution at $t = 0$ for $T = 0.2$ after 1 and 20 DSB steps

DSB 1

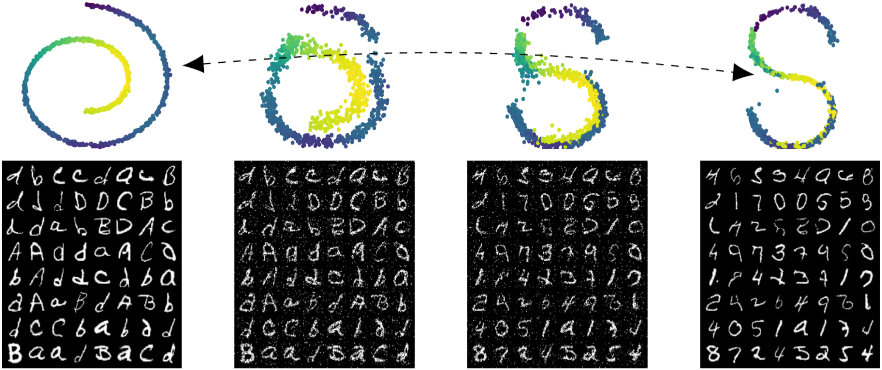


DSB 8



Generated samples ($N = 12$) and two-dimensional visualization of samples (red) compared to original MNIST data (blue) using pre-trained VAE ($d = 784$)

Applications: Datasets Interpolation



First row: Swiss-roll to S-curve (2D). Step 9 of DSB with $T = 1$ ($N = 50$). From left to right: $t = 0, 0.4, 0.6, 1$. Second row: EMNIST to MNIST. Step 10 of DSB with $T = 1.5$ ($N = 30$). From left to right: $t = 0, 0.4, 1.25, 1.5$.

Discussion

- Generative modeling can be reformulated as a Schrödinger Bridge problem.
- Diffusion Schrödinger Bridge approximates its solution using (discretized) forward-backward diffusions and score matching ideas.
- Experiments show it can speed up Score-Based Generative Models and is complementary to alternative acceleration techniques.
- Applicable to numerous optimal transport problems and Bayesian inverse problems.
- How does it scale with dimension? What are the statistical properties of score matching? Why does it work?

References

- V. De Bortoli, J. Thornton, J. Heng & A. Doucet, Diffusion Schrödinger bridge with applications to score-based generative modeling. arXiv:2106.01357.
- J. Ho, A. Jain and P. Abbeel, Denoising diffusion probabilistic models. NeurIPS 2020.
- J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan and S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, ICML 2015.
- Y. Song, J. Sohl-Dickstein, D.P. Kingma, A.Kumar, S. Ermon and B. Poole, Score-based generative modeling through stochastic differential equations, ICLR 2021.