Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling

—

a computational optimal transport project

Hugo Simon*

October 13, 2022

Abstract

In the last years, generative modeling has become a classical task in machine learning and image processing. Among all current methods aiming at solving this problem, Score-based Generative Modeling has recently shown great promises in outperforming Generative Adversarial Networks. It consists of simulating a noising diffusion process in order to reverse its dynamic. A limitation of this approach is that the forward-time Stochastic Differential Equation must be run for a sufficiently long time for the final distribution to be approximately Gaussian while ensuring that the corresponding time-discretization error is controlled. On the opposite, Schrödinger Bridge does not necessarily involve mixing processes, is thus more general and can generate samples from the data distribution in finite time. Hereby is presented Diffusion Schrödinger Bridge, an original approximation of the Iterative Proportional Fitting procedure to solve the Schrödinger Bridge problem in continuous state space, as an analogue to the popular Sinkhorn algorithm in discrete state space.

1 Introduction

1.1 Generative modeling

In generative modeling we are interested into designing algorithms which transform a given distribution p_{prior} into a given data distribution p_{data} . Only through samples do we have access to the data distribution. In the last years, generative modeling has become a classical task in machine learning and image processing. Energy-Based models, Generative Adversarial Networks, normalizing flows, Variational Auto-encoders, and diffusion score-matching algorithms are just a few of the frameworks that have been proposed to solve this challenge. The latter has showed significant promise in terms of outperforming GANs in terms of visual quality (Dhariwal et al. 2021), and can be recast as the discretization of some diffusion process making it mathematically interesting as well.

*Email: hugo.simon@telecom-paris.fr

1.2 Score-based Generative Modeling

1.2.1 Basic concepts

Let us recall some of the basics concepts of score-based generative modeling. We have access to a mixing process e.g. a Ornstein-Ulhenbeck process

$$\mathrm{d}\mathbf{X}_t = -\lambda \mathbf{X}_t \, \mathrm{d}t + \sqrt{2} \, \mathrm{dB}_t, \quad \mathbf{X}_0 \sim p_{\mathrm{data}}$$

with $\lambda > 0$. The process $(\mathbf{X}_t)_{t\in[0,T]}$ is interpreted as a *noising process* such as described in Sohl-Dickstein et al. 2015, perturbing the data distribution p_{data} . We denote p_{prior} the invariant distribution of the noising process, here $p_{\mathrm{prior}} = \mathcal{N}(0, 1/\lambda)$. If $T > 0$ is large enough then the distribution of $\mathbf{X}_T$ is close to p_{prior} due to the ergodicity of the Ornstein-Ulhenbeck process. Then one can (approximately) sample from p_{data} by first sampling from p_{prior} and reversing the dynamics of $(\mathbf{X}_t)_{t\in[0,T]}$. Surprisingly this time-reversal operation leads to another diffusion process with explicit drift and diffusion matrix.

$$\mathrm{d}\mathbf{Y}_t = \{\lambda \mathbf{Y}_t + 2\nabla \log p_{T-t}(\mathbf{Y}_t)\} \, \mathrm{d}t + \sqrt{2} \, \mathrm{dB}_t, \quad \mathbf{Y}_0 \sim p_{\mathrm{prior}}$$

where $\boldsymbol{p}_t$ is the density of $\mathbf{X}_t$. This means the backward dynamic is retrieved by injecting forward dynamic informationas marginals scores $\nabla \log p_{T-t}$. Song et al. 2020, Ho et al. 2020 shown a generative model can be obtained using an Euler-Maruyama discretization of the previous diffusion process. We end up with the following Markov chain.

$$Y^\star_{k+1} = Y^\star_k + \gamma_{k+1} \{\lambda Y^\star_k + 2\nabla \log p_{T-t_k}(Y^\star_k)\} + \sqrt{2\gamma_{k+1}} \mathbf{Z}_k, \quad \mathbf{Z}_k \sim \mathcal{N}(0,1)$$

where $\{\gamma_{k+1}\}_{k=0}^{N-1}$ is a sequence of stepsizes and $t_k = \sum_{j=0}^{k-1} \gamma_{j+1}$. The Markov chain $\{Y^\star_k\}_{k=0}^{N}$ cannot be computed in practice because we do not have access to the logarithmic gradient (Stein score) $\nabla \log p_t$. In score-based generative modeling techniques this score is approximated by a neural network s_θ solving the following variational problem (Song et al. 2021).

$$\theta^\star = \mathrm{argmin}_\theta \, \mathrm{E}\left[\|\mathbf{Z}/\sigma_t + s_\theta(\mathbf{X}_t)\|^2\right] \tag{1}$$

where we recall that solutions of the Orstein-Ulhenbeck process can be written as $\mathbf{X}_t = m_t \mathbf{X}_0 + \sigma_t \mathbf{Z}$, with $\mathbf{Z} \sim \mathcal{N}(0,1), m_t = \exp[-\lambda t]$ and $\sigma_t^2 = (1 - \exp[-2\lambda t])/\lambda$. The variational formulation for s_θ can be obtained using the following formula

$$\nabla \log p_t(x) = \mathbf{E}\left[(m_t \mathbf{X}_0 - \mathbf{X}_t)/\sigma_t^2 \mid \mathbf{X}_t = x\right] = -\mathbf{E}\left[\mathbf{Z}/\sigma_t \mid \mathbf{X}_t = x\right]$$

Finally the generative model is given by the following Markov chain

$$Y_{k+1} = Y_k + \gamma_{k+1} \{\lambda Y_k + 2 s_\theta(T - t_k, Y_k)\} + \sqrt{2\gamma_{k+1}} \mathbf{Z}_k, \quad \mathbf{Z}_k \sim \mathcal{N}(0,1) \tag{2}$$

which is easy to simulate and sample from.

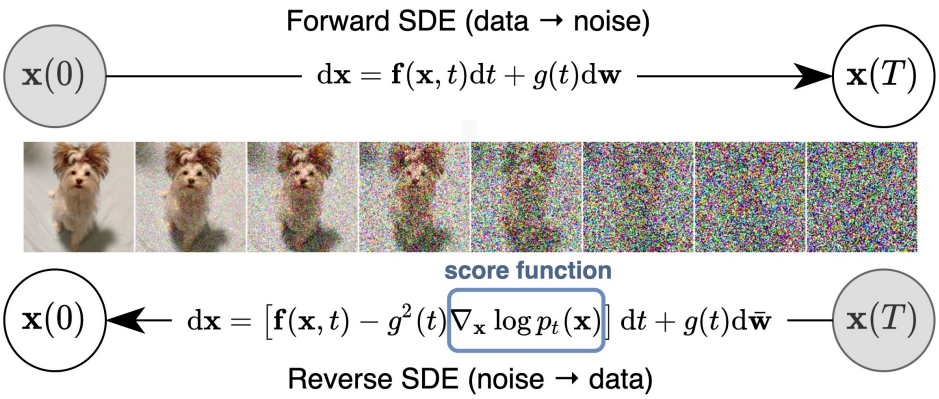

Figure 1: More complex noising processes can be used, optimizing the mixing rate.

More generally, solving a reverse Stochastic Differential Equation (SDE) yields a score-based generative model. Transforming data to a simple noise distribution can be accomplished with an SDE. It can be reversed to generate samples from noise if we know the score of the distribution at each intermediate time step. These ideas are at the core of every score-based generative modeling method.

1.2.2 Current limitations

One of the main disadvantages of score-based generative modeling is that it necessitates a large number of step sizes in order for the initial forward dynamics to be close to the distribution p_{prior} , as well as small enough stepsizes for the neural network approximation to hold.

The Diffusion Schrödinger Bridge presented in the follwing is a new approach that generalizes previous score-based methods, reducing the number of stepsizes required to construct score-based generative modeling. This contribution also eliminates the requirement that p_{prior} be Gaussian, allowing for future applications in high-dimensional optimal transport.

2 Schrödinger Bridge

2.1 The problem

The Schrödinger Bridge (SB) problem is an old and classical problem (see Schrödinger 1932) appearing in applied mathematics, optimal control and probability. In the continuous-time setting, it takes the following (dynamic) form. Consider as reference diffusion $(\mathbf{X}_t)_{t\in[0,T]}$ with distribution $\mathbb{P}$ describing the process adding noise to the data. We aim to find π^* such that $\pi_0 = p_{\text{data}}$ and $\pi_T = p_{\text{prior}}$ and minimize the Kullback-Leibler divergence between π^* and $\mathbb{P}$. More precisely

$$\pi^\star = \operatorname{argmin} \{\mathrm{KL}(\pi \mid \mathbb{P}), \pi_0 = p_{\text{data}} \,, \pi_T = p_{\text{prior}} \}$$

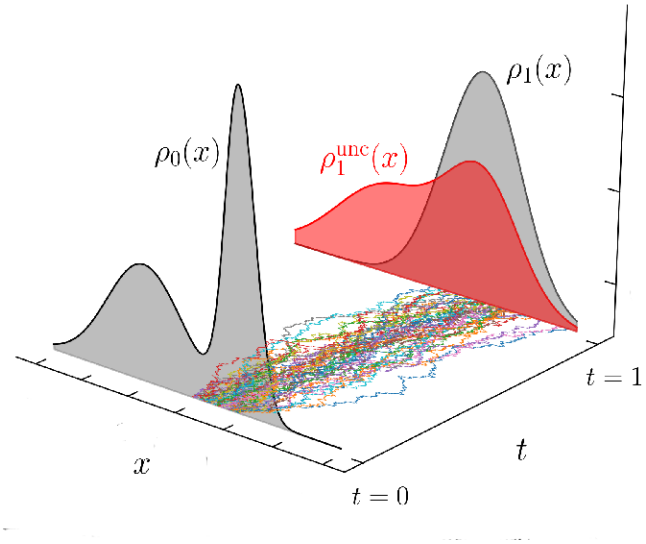

Figure 2: Knowing departure distribution and reference dynamic, what is the actual dynamic knowing also the arrival ? It actually can be seen as an entropy-regularized optimal transport problem on path spaces.

2.2 Iterative Proportional Fitting

In order to find the solution of the SB problem Iterative Proportional Fitting (IPF) (Kullback 1968) operates iteratively by successively solving half-bridge problems. Starting with the reference dynamic $\pi^0 = \mathbb{P}$, then project alternatively

$$\pi^{2n+1} = \text{argmin} \left\{ \text{KL}\left(\pi \mid \pi^{2n}\right), \pi_T = p_{\text{prior}} \right\}$$
$$\pi^{2n+2} = \text{argmin} \left\{ \text{KL}\left(\pi \mid \pi^{2n+1}\right), \pi_0 = p_{\text{data}} \right\}$$

Doing so, we define a sequence of distributions $(\pi^n)_n \in \mathbb{N}$ such that for large n, π^n is close to the Schrödinger bridge π^*. Hence, a generative model of p_{data} is obtained by sampling from π_0^{2n+1}.

In the discrete state-space case, Iterative Proportional Fitting is also known as the Sinkhorn algorithm, which is already well studied and which solves very efficiently the Schrödinger Bridge. However, in continuous state-space and continuous time, IPF iterates are difficult to approximate. The main differences are the following

1. Arrival state-space p_{data} is not known analytically and has to be sampled from.
2. Continuous time has to be discretized.
3. The dynamics are infinite dimensional and they can not easily be reversed with a simple transposition as in Sinkhorn. They can be approximated through parametric functions, and reversed via a well-defined loss.

To implement these approximations, we next introduce Diffusion Schrödinger Bridge (DSB), a new algorithm which uses score-matching approaches to approximate the IPF algorithm in continuous state-space. Thus DSB can be seen as a refinement of existing score-based generative modeling methods.

3 Diffusion Schrödinger Bridge

3.1 Relation between SB and SGM

We first show how SB problem is related to score-based generative modeling. Assume that $\pi^{2n} = (\pi_t^{2n})_{t\in[0,T]}$ is the measure associated with the diffusion

$$\mathrm{d}\mathbf{X}_t^n = f_t^n\left(\mathbf{X}_t^n\right)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t, \quad \mathbf{X}_0^n \sim p_{\mathrm{data}}$$

then we show that $\left(\pi_{T-t}^{2n+1}\right)_t$ is associated with the diffusion

$$\mathrm{d}\mathbf{Y}_t^n = b_{T-t}^n\left(\mathbf{Y}_t^n\right)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t, \quad \mathbf{Y}_0^n \sim p_{\mathrm{prior}}$$

where

$$b_t^n(x) = -f_t^n(x) + 2\nabla \log p_t^n(x)$$

with p_t^n the density of π_t^{2n}. Repeating this procedure we obtain that π^{2n+2} is associated with the diffusion

$$\mathrm{d}\mathbf{X}_t^{n+1} = f_t^{n+1}\left(\mathbf{X}_t^{n+1}\right)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t, \quad \mathbf{X}_0^{n+1} \sim p_{\mathrm{data}}$$

where

$$f_t^{n+1}(x) = -b_t^n(x) + 2\nabla \log q_t^n(x)$$

with q_t^n the density of π_t^{2n+1}. Note how this is a way of symmetrizing current SGM methods through Schrödinger Bridge framework. Hence, we can then iterate this procedure in an analogous way as Iterative Proportional Fitting.

3.2 Implementation

Of course we can not sample from these dynamics directly and we discretize them using Euler-Maruyama approximation.

The forward dynamic becomes $X_0 \sim p_{\mathrm{data}}$ and $\forall k \in [\![1, N]\!]$

$$X_{k+1} = X_k + \gamma_{k+1} f_k(X_k) + \sqrt{2\gamma}\mathbf{Z}_{k+1}, \quad \mathbf{Z}_{k+1} \sim \mathcal{N}(0,1)$$

The backward dynamic becomes $X_N \sim p_{\mathrm{prior}}$ and $\forall k \in [\![0, N-1]\!]$

$$X_{k-1} = X_k + \gamma_{k+1} b_{k+1}(X_k) + \sqrt{2\gamma}\tilde{\mathbf{Z}}_k, \quad \mathbf{Z}_k \sim \mathcal{N}(0,1)$$

The logarithmic gradients are then approximated using score-matching techniques. Although for memory reasons we do not approximate the scores but the drift functions defined by

$$B_{k+1}^n(x) = x + \gamma_{k+1} b_{k+1}^n(x), \quad F_k^n(x) = x + \gamma_{k+1} f_k^n(x), \quad \forall x \in \mathbb{R}^d$$

The variational approximation of the scores are then induced by

Proposition 1. *Assume that for any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$,*

$$q_{k|k+1}^n(x_k \mid x_{k+1}) = \mathcal{N}\left(x_k; B_{k+1}^n(x_{k+1}), 2\gamma_{k+1}\mathbf{I}\right), p_{k+1|k}^n(x_{k+1} \mid x_k) = \mathcal{N}\left(x_{k+1}; F_k^n(x_k), 2\gamma_{k+1}\mathbf{I}\right),$$

Then we have for any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$

$$B_{k+1}^n = \arg\min_{\mathrm{B} \in \mathrm{L}^2\left(\mathbb{R}^d, \mathbb{R}^d\right)} \mathbb{E}_{p_{k,k+1}^n}\left[\| \mathrm{B}(X_{k+1}) - (X_{k+1} + F_k^n(X_k) - F_k^n(X_{k+1}))\|^2\right]$$

$$F_k^{n+1} = \arg\min_{\mathrm{F} \in \mathrm{L}^2\left(\mathbb{R}^d, \mathbb{R}^d\right)} \mathbb{E}_{q_{k,k+1}^n}\left[\left\| \mathrm{F}(X_k) - \left(X_k + B_{k+1}^n(X_{k+1}) - B_{k+1}^n(X_k)\right)\right\|^2\right]$$

The empirical loss are defined as Monte Carlo estimates of the above expectations.

Finally, we use neural networks $B_{\beta^n}(k,x) \approx B_k^n(x)$ and $F_{\alpha^n}(k,x) \approx F_k^n(x)$ to approximate functions in $\mathrm{L}^2\left(\mathbb{R}^d, \mathbb{R}^d\right)$.

The discretization, the variational approximation, and the parametric function framework thereby define the DSB algorithm.

Algorithm 1 Implementation of Diffusion Schrödinger Bridge

Input: Samples from p_{data} and p_{prior}, stepsize γ, number of timesteps N.

1. Sample forward $X_0 \sim p_{\text{data}}$ and $X_{k+1} = F_k^\alpha(X_k) + \sqrt{2\gamma}\mathbf{Z}_{k+1}$ (Ornstein-Ulhenbeck process as initialization)
2. Compute backward loss on batches and update weights
   - $\hat{\ell}^b(\beta) = \mathbf{Mean}\left(\|B_{k+1}^\beta(X_{k+1}) - (X_{k+1} + F_k^\alpha(X_{k+1}) - F_k^\alpha(X_k))\|^2\right)$
   - $\beta \leftarrow$ Gradient Step$(\hat{\ell}^b(\beta))$
3. Sample backward $X_N \sim p_{\text{prior}}$ and $X_{k-1} = B_k^\beta(X_k) + \sqrt{2\gamma}\tilde{\mathbf{Z}}_k$
4. Compute forward loss on batches and update weights
   - $\hat{\ell}^f(\alpha) = \mathbf{Mean}\left(\|F_k^\alpha(X_k) - (X_k + B_{k+1}^\beta(X_{k+1}) - B_{k+1}^\beta(X_k))\|^2\right)$
   - $\alpha \leftarrow$ Gradient Step$(\hat{\ell}^f(\alpha))$
5. Iterate until convergence

Output: Sampling function for p_{data} from p_{prior} (and *vice-versa*)

Each of the iterations of {1, 2, 3, 4} is a DSB step and corresponds to 2 alternating projections of the IPF scheme, one forward from p_{data} , one backward from p_{prior} .

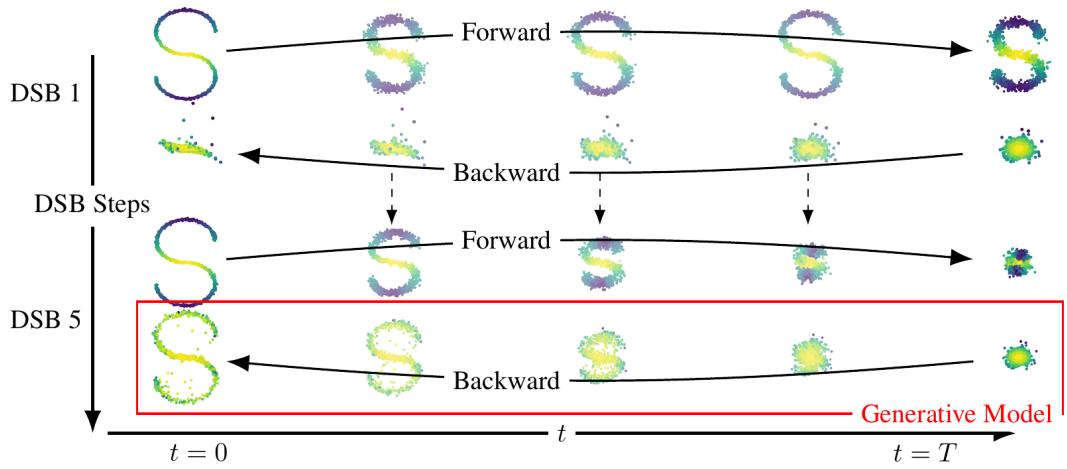

Figure 3: Illustrations of DSB solving Schrödinger Bridge between S-shaped p_{data} and a Gaussian p_{prior} . Note how in the first forward sampling, the process does not need to already converge to the prior.

3.3 Theoretical results

One interesting result concerning SGM previously introduced is

Theorem 1 (Approximation bound for Ornstein–Uhlenbeck based SGM)**.** *Assume* $\exists \mathrm{M} \geq 0$ *such that* $\forall (t, x) \in [0, T] \times \mathbb{R}^d$

$$\|s_{\theta^\star}(t, x) - \nabla \log p_t(x)\| \leq \mathrm{M} \tag{3}$$

with $s_{\theta^\star} \in \mathrm{C}\left([0, T] \times \mathbb{R}^d, \mathbb{R}^d\right)$.

Assume some mild other bounds on the score: $p_{data} \in \mathrm{C}^3\left(\mathbb{R}^d, (0, +\infty)\right)$ *is bounded and that there exist* $d_1, A_1, A_2, A_3 \geq 0, \beta_1, \beta_2, \beta_3 \in \mathbb{N}$, $\beta_1 = 1$ *and* $\mathrm{m}_1 > 0$ *such that for any* $x \in \mathbb{R}^d$ *and* $i \in \{1, 2, 3\}$

$$\left\|\nabla^i \log p_{data}(x)\right\| \leq A_i\left(1 + \|x\|^{\beta_i}\right), \quad \langle \nabla \log p_{data}(x), x\rangle \leq -\mathrm{m}_1\|x\|^2 + d_1\|x\|,$$

Then for any $\alpha \geq 0$, *there exist* $B_\alpha, C_\alpha, D_\alpha \geq 0$ *such that for any* $N \in \mathbb{N}$ *and* $\{\gamma_k\}_{k=1}^N$ *with* $\gamma_k > 0$ *for any* $k \in \{1, \ldots, N\}$, *the following bounds on the total variation distance hold:*

$$\|\mathcal{L}(X_0) - p_{data}\|_{\mathrm{TV}} \leq C_\alpha\left(\mathrm{M} + \gamma^{1/2}\right)\exp\left[D_\alpha T\right] + B_\alpha \exp\left[-\alpha^{1/2} T\right]$$

where $\mathcal{L}(X_0)$ *is the distribution of* X_0 *(backsampled from* X_N*).*

Condition 3 ensures that the neural network approximates the score with a given precision $M \geq 0$. Under this condition and conditions on p_{data} , **Theorem 1** states how the Markov chain defined by 2 approximates p_{data} in the total variation norm $\|\cdot\|_{TV}$.

The bounds of **Theorem 1** show that there is a trade-off between the mixing properties of the forward diffusion which increases with α, and the quality of the discrete-time approximation which deteriorates as α and T increase.

Though this does not provide approximation guarantees for general SGM where the noising process is not a Ornstein–Uhlenbeck process, it is hoped the behavior remain the same concerning more general $(f_t)_t$ dynamics. Concerning Diffusion Schrödinger Bridge, this will provide approximation guarantees for the reversing of the dynamic which is crucial for IPF to project the right joint.

Another theoretical results is on IPF convergence rate.

Proposition 2 (Convergence rate of IPF)**.** *It is known that IPF converges at geometric rate in the case where p_{data} and p_{prior} are compactly supported (see Peyré et al. 2020 for the discrete case, Chen et al. 2015 otherwise). Moreover, convergence rate is $o(1/n)$ in the non-compact setting.*

This allows in particular to use without hard restrictions a variety of p_{prior} to sample any p_{data} .

3.4 Numerical results

We evaluate the validity of the approach on toy two dimensional examples. Contrary to existing SGM approaches we do not require that the number of steps is large enough for $p_N \approx p_{\text{prior}}$ to hold.

To approximate B_k^n and F_k^n, we first use a sequence of Multi Layer Perceptron (MLP) indexed by the step k with ReLU activation function of shape $(2, 32, 32, 2)$ (the last layer is of course not activated to not constrained drift to be positive). Then we implemented (nearly) the same fully connected network as in the original paper. It uses positional encoding described in Vaswani et al. 2017 instead of a sequence of networks, and its precise architecture is described in Figure 4.

We compared DSB with SGM described in the introduction and defined by the Markov chain 2 and the variational loss 1. An Ornstein-Ulhenbeck process with value $\lambda = 10$ is used for both the noising process of the SGM, and the forward sampling initialization of DSB. We use a constant discretization step $\gamma = 0.03$, and a number of time steps $N = 20$ (which means $T = 0.6$). Since T is small, we do not have $p_N \approx p_{\text{prior}}$ and the reverse-time process obtained after the first DSB iteration (corresponding to original SGM methods) does not yield a satisfactory generative model for the SGM. However, multiple iterations of DSB improve the quality of the synthesis.

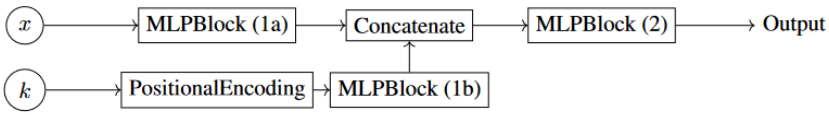

Figure 4: Multi Layer Perceptron with Positional Encoding. It consists of MLP Block with ReLU activation. **MLPBlock(1a)** is shape $(2, 16, 32)$, **MLPBlock(1b)** is shape $(16, 16, 32)$ (note that in the original paper, it is curiously of shape $(1, 16, 32)$), **MLPBlock(2)** is shape $(64, 128, 128, 2)$ (the last layer is of course not activated).

For datasets, we use mixture of Gaussian densities and banana-shaped densities. The latter, which is linked to the Rosenbrock function, has a highly non-standard support with very long and narrow tails and it is challenging in MCMC framework to sample from this target (see Tran et al. 2013 for more details). In the first following example, we use a Gaussian density for p_{prior} and a mixture of a banana-shaped and 2 Gaussian densities as p_{data} . In the second example, we use 2 mixtures of a banana-shaped and 2 Gaussian densities for p_{prior} and p_{data} , but the latter has been deformed (pushed-forward) by an affine transform $Ax + h$ with $A = \begin{pmatrix} -2.5 & 0.6 \\ 1 & -1 \end{pmatrix}$ and $h = \begin{pmatrix} -0.4 & -2 \end{pmatrix}^\top$.

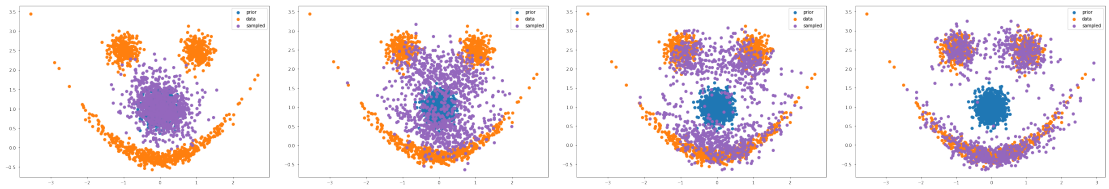

Figure 5: Backward-sampling a mixture of densities from a gaussian prior after 5 DSB steps. Intermediate states constitute data interpolation.

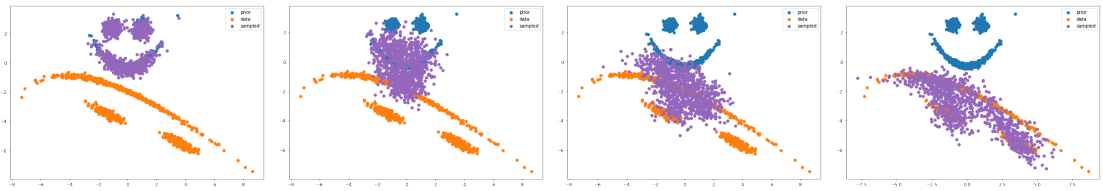

Figure 6: Backward sampling a mixture of densities from a affine-transformed prior after 5 DSB steps. Intermediate states constitute data interpolation.

Without surprises, DSB performs better than the classical SGM for the latter suits less to the given problems (short time and non gaussian prior). However, we observed DSB lack of robustness, as a careful tuning of parameters N, γ and the number of DSB steps, are required.

Note that as all sequential architectures (SGM, LSTM, Langevin sampling...), DSB involve computations which are not parallelizable, as in forward and backward sampling, each step is required to compute the following one.

We also studied the impact of choosing simpler loss functions, modifying the previous ones, that can be interpreted as just "softly gluing" a dynamic with its reversing, i.e. with losses defined as $\tilde{\ell}^f(\alpha) = \mathbf{Mean}\left(\|F_k^\alpha(X_k) - X_{k+1}\|^2\right)$ computed on a backward sampling batch, and $\tilde{\ell}^b(\beta) = \mathbf{Mean}\left(\|B_{k+1}^\beta(X_{k+1}) - X_k\|^2\right)$ computed on a forward sampling batch. We didn't noticed differences in the quality of approximation. This could be a way of not having to forward-pass several times through forward network F^α when computing backward loss $\ell^b(\beta)$ and *vice-versa*.

4 Conclusion and perspective

Contrary to existing score-based generative modeling methods which require the reference process to converge to p_{prior} . the convergence of the algorithm is determined by the convergence of the IPF.

The DSB algorithm can be used on top of existing algorithms. Hence, the proposed method can be seen as a refinement of original score-based generative models and all techniques used to improve the quality/speed of theses methods can be implemented for DSB.

DSB does not require p_{prior} to be Gaussian. In fact we only require having access to samples from p_{prior} which can be another dataset. In particular we are able to perform dataset interpolation. This paves the way for further applications for high dimensional optimal transport.

DSB does not achieve state-of-the-art generative modeling results due to compute limitations as existing architectures are used in order to parameterize the score approximations. These architectures are deep and notably instable. This requires careful selection of the parameters of DSB. One perspective of improvement is thus to find dedicated architectures that are more stables.

Besides, a lot of theoretical results can still be done. The original paper does not provide a lot of results concerning its algorithm, especially approximations guarantees for reversing more general dynamics under a given score approximation, which are crucial for IPF results to apply.

5 Connexion with the course

The connections with the course **Computational Optimal Transport** are pretty explicit and straightforward. In class, we studied discrete Schrödinger bridge problem and proved geometric convergence of Sinkhorn algorithm. In this project, we studied a way of bringing the Sinkhorn algorithm principles into the continuous time and state-space world, modulo some approximations, to solve the continuous Schrödinger bridge problem.

References

Chen, Yongxin et al. (2015). *Entropic and displacement interpolation: a computational approach using the Hilbert metric*. arXiv: 1506.04255 [math.OC].

Dhariwal, Prafulla et al. (2021). *Diffusion Models Beat GANs on Image Synthesis*. arXiv: 2105.05233 [cs.LG].

Ho, Jonathan et al. (2020). *Denoising Diffusion Probabilistic Models*. arXiv: 2006.11239 [cs.LG].

Kullback, S. (1968). "Probability Densities with Given Marginals". In: *The Annals of Mathematical Statistics* 39.4, pp. 1236 –1243. DOI: 10.1214/aoms/1177698249. URL: https://doi.org/10.1214/aoms/1177698249.

Peyré, Gabriel et al. (2020). *Computational Optimal Transport*. arXiv: 1803.00567 [stat.ML].

Schrödinger, E. (1932). "Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique". fr. In: *Annales de l'institut Henri Poincaré* 2.4, pp. 269–310. URL: http://www.numdam.org/item/AIHP_1932__2_4_269_0/.

Sohl-Dickstein, Jascha et al. (2015). *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. arXiv: 1503.03585 [cs.LG].

Song, Yang et al. (2020). *Generative Modeling by Estimating Gradients of the Data Distribution*. arXiv: 1907.05600 [cs.LG].

Song, Yang et al. (2021). *Score-Based Generative Modeling through Stochastic Differential Equations*. arXiv: 2011.13456 [cs.LG].

Tran, Minh-Ngoc et al. (2013). *Adaptive Metropolis-Hastings Sampling using Reversible Dependent Mixture Proposals*. arXiv: 1305.2634 [stat.ME].

Vaswani, Ashish et al. (2017). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL].