Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling

De Bortoli et al., 2023, Neurips

Understanding the Intersection of Schrödinger Bridges, Diffusion Processes, and Generative Models…

Take-home messages

Core contributions of the paper:

1. Introduces Diffusion Schrödinger Bridge (DSB) as a principled framework combining Schrödinger bridges and score-based generative models.
2. **Provides an efficient approximation to Iterative Proportional Fitting (IPF).**
3. Seems to produce better quality results than SBMs in practice?
4. More efficient at inference time than SBMs.

Why should we care?

- Standard diffusion models require thousands of steps, making training and inference computationally expensive! The DSB introduced here should provide a much more efficient generative model.
- Prior methods for Schrodinger bridges focuses on uniqueness and were quite abstract without algorithmic implementations.
- We might want to set the prior distribution to something that isn't Gaussian… ? E.g., We might want to map between distributions using unpaired samples (image style transfer)

DSB integrates diffusion and Schrödinger Bridge theory to achieve principled and practical generative modeling.

Take-home messages

Before we start, if I was to (crudely) over-simplify the DSB method as best possible…

(i) We just take a diffusion model where **we sample from a forward process to create noised samples** and then **learn a backward process through some neural network that can remove the noise**.

(ii) We then take another diffusion model where **we sample from the backward process** which takes us from a latent space sample to a ‘less noisy’ sample. We then use this as a sample to **learn a forward process that can re-add the noise**.

(iii) We then iteratively repeat (i) and (ii) until we get a final convergence. These iterative procedure provides consistency with the data and prior distributions. The forward and backward processes become more ‘aligned’.

However, they use some tricks so as not to need to train a whole diffusion model on each iteration…..

What is a Schrodinger bridge?

*A path measure is a probability distribution over entire trajectories/paths of a stochastic process, describing the likelihood of different sequences of states over time.

Problem Statement: Given two distributions π_0 (initial) and π_1 (final), find a stochastic process P (path measure*) that:

(i) starts with π_0 and ends at π_1

(ii) **minimizes the relative entropy between P and a reference process Q.

$$\pi^\star = \arg\min \{ \mathrm{KL}(\pi | p) \ : \ \pi \in \mathscr{P}_{N+1}, \ \pi_0 = p_{\mathrm{data}}, \ \pi_N = p_{\mathrm{prior}} \} \,.$$

The desired process, which transforms π_0 into π1

Brownian motion or another diffusion process

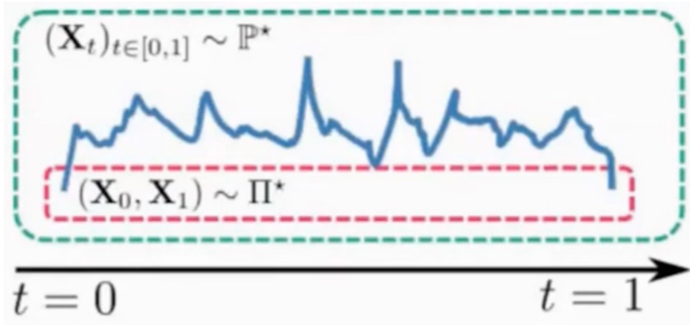
**Note that without (ii) we get the static schrodinger bridge formulation which ignores intermediate dynamics and is thus similar to optimal transport

Schrödinger Bridges can be viewed as a generalization of optimal transport with entropic regularization:

1. Optimal transport minimizes a cost (e.g., Wasserstein distance) for moving mass between π_0 and π_1.

2. Schrödinger Bridges add stochasticity (via the reference process Q) and minimize a probabilistic cost.

A Schrödinger Bridge (SB) is a stochastic process that provides a probabilistic interpolation between two probability distributions, π_0 and π_1, while minimizing a cost (KL divergence?) subject to constraints.


$(\mathbf{X}_t)_{t\in[0,1]} \sim \mathbb{P}^\star$
$(\mathbf{X}_0, \mathbf{X}_1) \sim \Pi^\star$
$t = 0$
$t = 1$

How is an SB solved?

This is my constraint that I need to satisfy!

Iterative Proportional Fitting (IPF).

Brownian motion or another diffusion process

1. Start with a reference process Q.
2. Alternate backward and forward updates (many times).

Initially this would be the Brownian motion

Odd iteration $$\pi^{2n+1} = \arg\min \left\{ \mathrm{KL}(\pi|\pi^{2n}) \ : \ \pi \in \mathscr{P}_{N+1}, \ \pi_N = p_{\mathrm{prior}} \right\},$$

Find a new path measure π that is as close as possible (in terms of KL divergence) to the forward path measure π^{2n} from the previous step.

Ensure that π^{2n+1} such that its final-time marginal matches p_{prior} Finding optimal path measure for backward process

Even iteration $$\pi^{2n+2} = \arg\min \left\{ \mathrm{KL}(\pi|\pi^{2n+1}) \ : \ \pi \in \mathscr{P}_{N+1}, \ \pi_0 = p_{\mathrm{data}} \right\}.$$

Find a new path measure π that is as close as possible (in terms of KL divergence) to the forward path measure π_{2n+1} from the previous step.

Ensure that π^{2n+2} such that its final-time marginal matches p_{data} Finding optimal path measure for forward process

3. Reach convergence.

Prior solutions to IPF:

Solvable by updating joint density p (shown in appendix D2)

Requires approximating the potentials (that transform reference process to desired process)

Theoretically feasible but computationally prohibitive for high-dimensional problems due to the complexity of solving forward and backward PDEs or sampling-based approximations.

Linking forward and backward processes

Proposition 2. *Assume that* $\mathrm{KL}(p_{\text{data}} \otimes p_{\text{prior}} | p_{0,N}) < +\infty$. *Then for any* $n \in \mathbb{N}$, π^{2n} *and* π^{2n+1} *admit positive densities w.r.t. the Lebesgue measure denoted as* p^n *resp.* q^n *and for any* $x_{0:N} \in \mathcal{X}$, *we have* $p^0(x_{0:N}) = p(x_{0:N})$ *and*

Backward process

Forward process

$$q^n(x_{0:N}) = p_{\text{prior}}(x_N) \prod_{k=0}^{N-1} p^n_{k|k+1}(x_k|x_{k+1}), \quad p^{n+1}(x_{0:N}) = p_{\text{data}}(x_0) \prod_{k=0}^{N-1} q^n_{k+1|k}(x_{k+1}|x_k).$$

Reverse time probability density at iteration n.

Anchors reverse process at t=N

Forward process p is used to transition back in time

forward time probability density at iteration n+1.

Anchors forward process at data

Backward process to transition forward in time

What are they saying?

Just like a standard noising…

1. Start with a reference process (gives us a baseline measure $\pi^{2n} = p(x_{0:N})$) which was constrained at the data dist.
2. Now we use this p^n to construct a new backward measure $\pi^n = q^{n+1}$. We do this by starting with the known prior distribution at time N and "propagating it backward" using transition probabilities derived from p^n.
3. Then we use this q^n to construct a new forward measure $\pi^{2n+2} = p^{n+1}$. We do this by starting with the known pdata at time 0 and "propagating it forward" using transition probabilities derived from q^n. The result is $p^{n+1}(x_{0:N})$, a new forward path measure that matches p_{data} at time 0 and is now also more consistent with the backward constraints introduced by q^n.
4. Repeats steps 2 and 3 till convergence….

We need to flip the transition directions using Bayes rule!

In practice we have access to $p^n_{k+1|k}$ and $q^n_{k|k+1}$. Hence, to compute $p^n_{k|k+1}$ and $q^n_{k+1|k}$ we use

$$p^n_{k|k+1}(x_k|x_{k+1}) = \frac{p^n_{k+1|k}(x_{k+1}|x_k)\,p^n_k(x_k)}{p^n_{k+1}(x_{k+1})}, \quad q^n_{k+1|k}(x_{k+1}|x_k) = \frac{q^n_{k|k+1}(x_k|x_{k+1})\,q^n_{k+1}(x_{k+1})}{q^n_k(x_k)}.$$

Gaussian approximation of transitions

In practice we have access to $p^n_{k+1|k}$ and $q^n_{k|k+1}$. Hence, to compute $p^n_{k|k+1}$ and $q^n_{k+1|k}$ we use

$$p^n_{k|k+1}(x_k|x_{k+1}) = \frac{p^n_{k+1|k}(x_{k+1}|x_k)p^n_k(x_k)}{p^n_{k+1}(x_{k+1})}, \quad q^n_{k+1|k}(x_{k+1}|x_k) = \frac{q^n_{k|k+1}(x_k|x_{k+1})q^n_{k+1}(x_{k+1})}{q^n_k(x_k)}.$$

The authors use a gaussian approximation:

tion 2.1. If at step $n \in \mathbb{N}$ we have $p^n_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; x_k + \gamma_{k+1} f^n_k(x_k), 2\gamma_{k+1}\mathbf{I})$ where $p^0 = p$ and $f^0_k = f$, then we can approximate the reverse-time transitions in Proposition 2 by

$$\begin{aligned} q^n_{k|k+1}(x_k|x_{k+1}) &= p^n_{k+1|k}(x_{k+1}|x_k) \exp[\log p^n_k(x_k) - \log p^n_{k+1}(x_{k+1})] \\ &\approx \mathcal{N}(x_k; x_{k+1} + \gamma_{k+1} b^n_{k+1}(x_{k+1}), 2\gamma_{k+1}\mathbf{I}), \end{aligned}$$

The backward (and forward) transitions can be approximated by a Gaussian with some backward (forward) drift term.

$p_k \approx p_{k+1}$, a Taylor expansion of $\log p_{k+1}$ at x_{k+1} and $f(x_k) \approx f(x_{k+1})$.

$$b^n_{k+1}(x_{k+1}) = -f^n_k(x_{k+1}) + 2\nabla \log p^n_{k+1}(x_{k+1}).$$

$$f^{n+1}_k(x_k) = -b^n_{k+1}(x_k) + 2\nabla \log q^n_k(x_k)$$

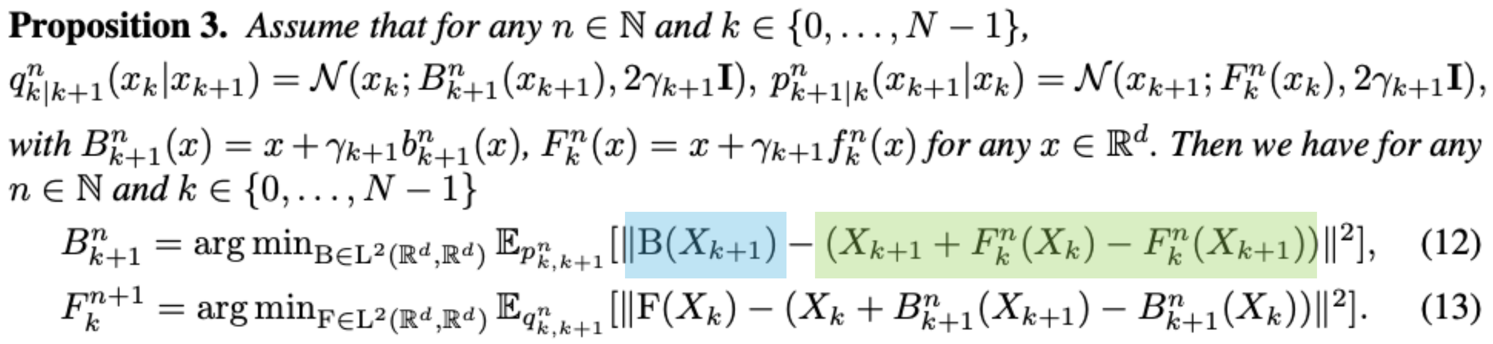
Approximating these functions using score matching would be expensive…
i.e., we would need to fit a diffusion model on every iteration…

Iterative mean matching proportional fitting

They reduce the iterations to a set of least-squares regression problems:

Iteratively update $B^n(k,x)$ and $F^n(k,x)$ by minimizing a discrepancy between the two processes in terms of their dynamics.

1. To find b_n, you adjust it so that the backward increments match those implied by the current forward measure.
2. To find f_{n+1}, you adjust it so that the forward increments match those implied by the updated backward measure.



Proposition 3. *Assume that for any* $n \in \mathbb{N}$ *and* $k \in \{0, \dots, N-1\}$, $q^n_{k|k+1}(x_k|x_{k+1}) = \mathcal{N}(x_k; B^n_{k+1}(x_{k+1}), 2\gamma_{k+1}\mathbf{I})$, $p^n_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; F^n_k(x_k), 2\gamma_{k+1}\mathbf{I})$, *with* $B^n_{k+1}(x) = x + \gamma_{k+1} b^n_{k+1}(x)$, $F^n_k(x) = x + \gamma_{k+1} f^n_k(x)$ *for any* $x \in \mathbb{R}^d$. *Then we have for any* $n \in \mathbb{N}$ *and* $k \in \{0, \dots, N-1\}$

$$B^n_{k+1} = \arg\min_{\mathrm{B} \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{p^n_{k,k+1}}[\|\mathrm{B}(X_{k+1}) - (X_{k+1} + F^n_k(X_k) - F^n_k(X_{k+1}))\|^2], \quad (12)$$

$$F^{n+1}_k = \arg\min_{\mathrm{F} \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{q^n_{k,k+1}}[\|\mathrm{F}(X_k) - (X_k + B^n_{k+1}(X_{k+1}) - B^n_{k+1}(X_k))\|^2]. \quad (13)$$

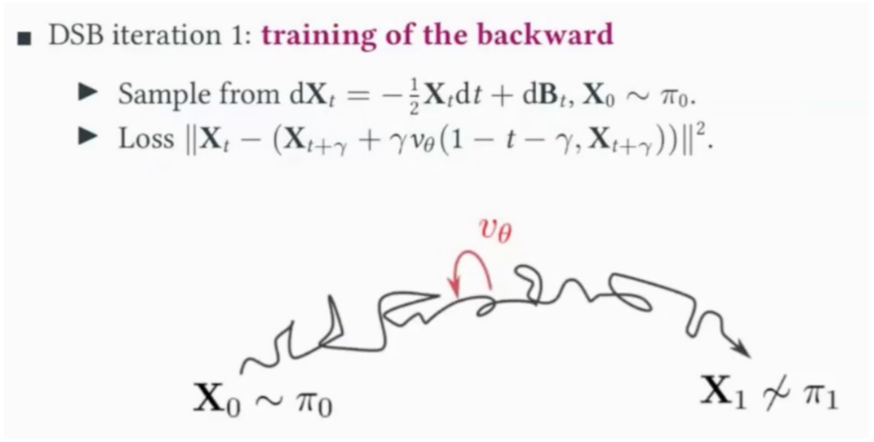
Predicts X_k from X_{k+1}.

Forward drift acts as our 'ground truth' transition from X_k from X_{k+1}.

They frame is as an optimisation problem.

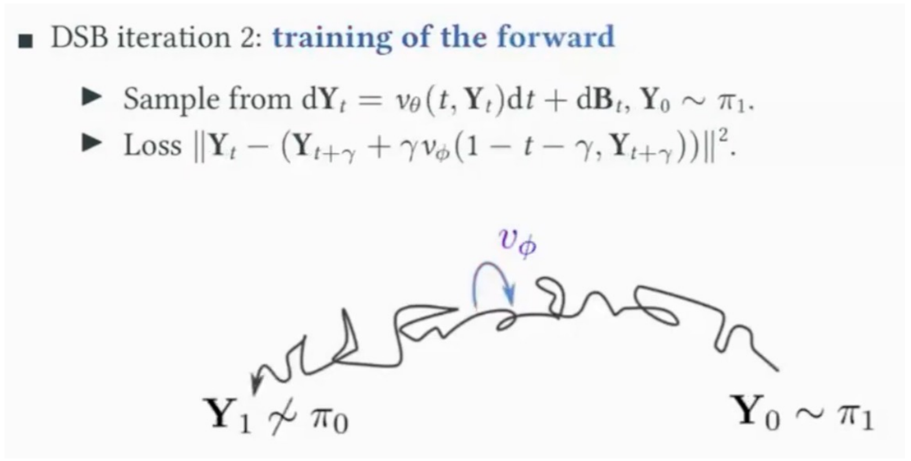
Instead of explicitly computing gradients of densities, one now just needs samples and can learn these functions F_n and B_n through empirical risk minimization, typically using neural networks to parameterize these functions.

Algorithm



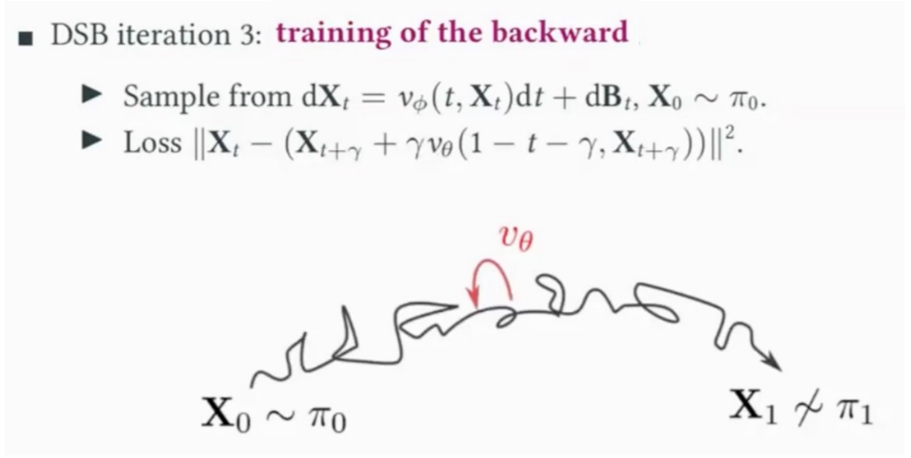
- DSB iteration 1: **training of the backward**
  - Sample from $\mathrm{d}\mathbf{X}_t = -\frac{1}{2}\mathbf{X}_t \mathrm{d}t + \mathrm{d}\mathbf{B}_t$, $\mathbf{X}_0 \sim \pi_0$.
  - Loss $\|\mathbf{X}_t - (\mathbf{X}_{t+\gamma} + \gamma v_\theta(1-t-\gamma, \mathbf{X}_{t+\gamma}))\|^2$.

Create a sample from forward process (e.g. add noise to image) and then learn to remove the noise with backward process.



- DSB iteration 2: **training of the forward**
  - Sample from $\mathrm{d}\mathbf{Y}_t = v_\theta(t, \mathbf{Y}_t)\mathrm{d}t + \mathrm{d}\mathbf{B}_t$, $\mathbf{Y}_0 \sim \pi_1$.
  - Loss $\|\mathbf{Y}_t - (\mathbf{Y}_{t+\gamma} + \gamma v_\phi(1-t-\gamma, \mathbf{Y}_{t+\gamma}))\|^2$.

Create a sample from backward process (start with prior distribution and remove noise) and then learn to add the noise with forward process



- DSB iteration 3: **training of the backward**
  - Sample from $\mathrm{d}\mathbf{X}_t = v_\phi(t, \mathbf{X}_t)\mathrm{d}t + \mathrm{d}\mathbf{B}_t$, $\mathbf{X}_0 \sim \pi_0$.
  - Loss $\|\mathbf{X}_t - (\mathbf{X}_{t+\gamma} + \gamma v_\theta(1-t-\gamma, \mathbf{X}_{t+\gamma}))\|^2$.

Create a sample from forward process and then further learn to remove the noise with backward process.

We are iterating the 'training' of two diffusion models: backward and forward!

The first iteration is then like a standard diffusion model with fixed Brownian motion in the forward process.

Diffusion Schrodinger bridge

Let’s compare SBM and DSB side by side…

Score based model		**Diffusion schrodinger bridge**
Fit a single model once.		Iteratively fit models (until convergence)
Fixed forward dynamics (e.g., Gaussian noise schedules).	$\Longleftrightarrow$	Forward dynamics $f_\theta(x,t)$ are learned to match $\pi_0 \rightarrow \pi_1$
Neural network $s\theta(x,t)$ predicts the score function $\nabla_x \log p_t(x)$.	$\Longleftrightarrow$	Backward dynamics $g_\phi(x,t)$ are learned to match $\pi_1 \rightarrow \pi_0$