Likelihood training of Schrödinger bridge using Forward-backward SDEs theory

Benjamin Maurel
MVA - ENSAE
École Normale Supérieure de Paris-Saclay
Gif-sur-Yvettes, 91190, France
benjamin.maurel@ensae.fr

Jérémie Stym-Popper
MF MVA - ENSAE
École Normale Supérieure de Paris-Saclay
Gif-sur-Yvettes, 91190, France
jeremie.stym-popper@ensae.fr

Gabriel Watkinson
MVA - ENSAE
École Normale Supérieure de Paris-Saclay
Gif-sur-Yvettes, 91190, France
gabriel.watkinson@ensae.fr

Introduction

Scored-based Generative Models (SGMs) have gained significant attention in the field of deep image generation. Since their introduction by Song et al. (2020), SGMs have emerged as the state-of-the-art approach for image generation. Unlike likelihood-based models such as VAEs or normalizing flows, which learn the probability density function directly, or adversarial-based models like GANs, SGMs learn a mapping between a simple distribution, typically Gaussian, and a complex, high-dimensional, and often intractable data distribution, such as images. These models offer several advantages over existing approaches, including high-quality sample generation comparable to GANs without the instability of adversarial training, flexible model architectures and the ability to solve inverse problems without re-training models. However, SGMs also suffer from pitfalls that limit their applicability, such as difficult training procedure, the impossibility to use a complex distribution as the prior, the fact that the drift can only be linear or degenerate, and slow training and sampling processes compared to other methods.

To address these limitations, recent works by De Bortoli et al. (2021); Wang et al. (2021); Vargas et al. (2021) have proposed a new class of generative models inspired by Optimal Transport theory, specifically using Schrödinger Bridges (SB) Schrödinger (1932). SBs, similar to SGMs, aim to learn a mapping between two distributions in a finite horizon. However, SBs offer greater flexibility by allowing the mapping between arbitrary distributions, making them more appealing. The relationship between SBs and SGMs was not clearly established, as SGMs rely on Stochastic Differential Equations (SDEs), whereas SBs are based on Partial Differential Equations (PDEs).

In this paper, Chen et al. (2023) bridges the gap between these seemingly distinct mathematical concepts by drawing inspiration from the field of Stochastic Optimal Control and utilizing Forward-Backward Stochastic Differential Equations (FBSDEs). The authors propose a method that enables the solution of PDEs with SDEs, offering computational efficiency and improved precision compared to solving PDEs directly. This approach introduces a novel training method for SBs, resembling the framework used in flow-based diffusion models. Additionally, the authors also use a more traditional Iterative Proportional Fitting approach (IPF; Kullback (1968); De Bortoli et al. (2021)).

Link to our GitHub: https://github.com/gwatkinson/mva_sb_generative, containing code of our experiments and further images and resources

In this report, we will first introduce the two classical methods, Scored-based Generative Models and Schrödinger Bridges, before exploring the Forward-Backward Stochastic Differential Equations method that establishes a connection between the two. Finally, we will discuss further extensions of this paper, including an application of the method with the Hamilton-Jacobi-Bellman PDE.

1 Some reminders about Score-based Generative Model

Given a dataset $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ of N samples in $\mathbb{R}^d$, the goal of generative models is to learn the underlying data distribution p_{data}, as to be able to generate new samples from this distribution. The go-to approach for this kind of problem is to use maximum likelihood estimation (MLE) to learn the parameters of a parametric model p_θ that approximates the distribution. However, this relies on the assumption that p_θ be a normalized density function. To do so, we have to calculate the normalizing constant, which is intractable in high dimension. Therefore, to train such models using MLE, we either have to restrict the architectures, e.g. to invertible networks in flow models, or approximate the normalizing constant, using MCMC methods for example. This is not ideal, as it limits the flexibility of the model and the quality of the approximation.

Another solution is to use the score function $s_\theta(x) = \nabla_x \log p_\theta(x)$ instead of the density function p_θ. This makes the normalization constant disappear thanks to the gradient, allowing a wider range of architectures. Score matching methods can then be used to learn the score function s_θ, followed Langevin dynamics to sample from the learned score function, giving access to the wanted distribution. However, this naive approach doesn't work well in practice, especially in low density regions.

The solution is to add a diffusion process to the score matching method, which is the idea behind Score-based Generative Models (SGM). In SGMs, inputs are "diffused" toward a random noise. This diffusion process is characterized by a stochastic differential equation (SDE):

$$\mathrm{d}\mathbf{X}_t = f(t, \mathbf{X}_t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{W}_t, \quad \mathbf{X}_0 \sim p_{\text{data}} \tag{1}$$

where $f(\cdot, t) : \mathbb{R}^n \to \mathbb{R}^n$, $g(t) \in \mathbb{R}$ are the drift and diffusion coefficients, and $\mathbf{W}_t \in \mathbb{R}^n$ is a standard Wiener process. For a large time step, we have that $p_T \approx p_{\text{prior}}$ that closely resemble a Gaussian distribution, which is easy to sample from.

The associated backward SDE is:

$$\mathrm{d}\mathbf{X}_t = \left[f(t, \mathbf{X}_t) - g(t)^2 \nabla_x \log p_t(\mathbf{X}_t)\right] \mathrm{d}t + g(t)\mathrm{d}\mathbf{W}_t, \quad \mathbf{X}_T \sim p_T \tag{2}$$

The score function appears naturally in the backward SDE, which is the reason why SGMs are called score-based. In the case where the drift is simple (e.g. linear or degenerate), it is possible to learn a score network $s(t, x; \theta)$ by regressing the outputs to the ground truth, i.e $\mathbb{E}\left[\lambda(t) \left\|s(t, x; \theta) - \nabla_x \log p_{t|x_0}\right\|^2\right]$, where the expectation is taken over the forward SDE. $\lambda(t)$ is a weighting function that is crucial in practice, but hard to choose well.

More recently, Song et al. (2021) have found a lower-bounds to the log-likelihood of the SGM, resulting in an Evidence Lower bound (ELBO) :

$$\log p_0^{\text{SGM}}(\mathbf{x}_0) \geq \mathbb{E}\left[\log p_T(\mathbf{X}_t)\right] - \int_0^T \mathbb{E}\left[\frac{1}{2} g^2 \left\|\mathbf{s}_t\right\|^2 - \nabla_x \cdot (g^2 s_t - f)\right] \mathrm{d}t \tag{3}$$

This can be obtained bounding the KL divergence by score-matching loss, where the weighting function $\lambda(t) = g(t)^2$. This lower bound is important since it will be one of the element that links together the SGM model and the Schrödinger-Bridge problem.

Once the model is trained, we can just substitute the score function by the learned one in the backward SDE (2) to sample from the data distribution.

2 Schrödinger Bridge

Unlike SGMs, Schrödinger Bridges (SB) are not generative models, but rather a way to interpolate between two distributions μ_0 and μ_1. There were first introduced by Schrödinger (1932) and are related to the field of regularized optimal transport. The Schrödinger Bridge problem can be defined as follows:

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

where Π^0 is the reference measure, which is set to the path measure of the forward diffusion process in (1).

The **Benamou-Brenier** reformulation of the entropy-regularized optimal transport problem gives the following partial differential equations (PDEs):

$$\begin{cases} \frac{\partial \Psi}{\partial t} = -\nabla_x \Psi^\top f - \frac{1}{2}\mathrm{Tr}(g^2 \nabla_x^2 \Psi) \\ \frac{\partial \widehat{\Psi}}{\partial t} = -\nabla_x \cdot (\widehat{\Psi} f) + \frac{1}{2}\mathrm{Tr}(g^2 \nabla_x^2 \widehat{\Psi}) \end{cases} \quad \text{s.t.} \quad \begin{array}{rcl} \Psi(0,\cdot)\widehat{\Psi}(0,\cdot) & = & p_{\text{data}} \\ \Psi(T,\cdot)\widehat{\Psi}(T,\cdot) & = & p_{\text{prior}} \end{array} \tag{4}$$

where Ψ and $\widehat{\Psi}$ can also be seen as solutions to the Schrödinger system of equations (we recall that the trace of the Hessian is the Laplace operator). f and g are functions defined on the same space as above.

If Ψ and $\widehat{\Psi}$ are solutions of these PDEs, then the solution of the SB problem is the path measure (i.e, that is in $\mathcal{P}(p_{\text{data}}, p_{\text{prior}})$ of the following SDEs:

$$\begin{aligned} \mathrm{d}\mathbf{X}_t &= \left[f + g^2 \nabla_x \log \Psi(t, \mathbf{X}_t) \right] \mathrm{d}t + g\mathrm{d}\mathbf{W}_t, \quad \mathbf{X}_0 \sim p_{\text{data}} \\ \mathrm{d}\mathbf{X}_t &= \left[f - g^2 \nabla_x \log \widehat{\Psi}(t, \mathbf{X}_t) \right] \mathrm{d}t + g\mathrm{d}\mathbf{W}_t, \quad \mathbf{X}_T \sim p_{\text{prior}} \end{aligned} \tag{5}$$

This establishes a first link between PDEs founding the SB problem and SDEs characterizing the SGM model. Those SDEs are really similar to the ones of the SGM model, except for the terms $\nabla_x \log \Psi(t, \mathbf{X}_t)$ which adds non-linearity to the drift and $\nabla_x \log \widehat{\Psi}(t, \mathbf{X}_t)$ which can be seen as a modified score function. In this situation, we have to learn both the score function and the drift function, while SGM only learns the score.

To create samples using this method, we have to solve the PDEs (4) to obtain $\widehat{\Psi}$. However, those are hard to solve, the proposed method of FBSDEs (Forward-Backward SDEs) is shown to be a convenient way to link the optimality conditions of the SB problem to the ELBO maximization of the likelihood $\log_0^{\mathrm{SGM}}(\mathbf{x}_0)$.

3 Proposed Method

There are two main methods to solve PDEs: linearization-based method and sampling-based method. The latter is a consequence of the Feynman-Kac lemma that provides a solution to parabolic PDEs with a conditional expectation (suitable for sampling), which establishes a link between this kind of PDE with stochastic processes. Here, the Nonlinear Feynman-Kac formula enables to link parabolic PDEs to forward-backward coupled SDEs as below. First, let's introduce a more general framework, so that (1), (2) and (5) fall inside:

$$\begin{cases} \mathrm{d}\mathbf{X}_t = f(t, \mathbf{X}_t)\mathrm{d}t + G(t, \mathbf{X}_t)\mathrm{d}\mathbf{W}_t & \mathbf{X}_0 = \mathbf{x}_0 \\ \mathrm{d}\mathbf{Y}_t = -h(t, \mathbf{X}_t, \mathbf{Y}_t, \mathbf{Z}_t)\mathrm{d}t + \mathbf{Z}(t, \mathbf{X}_t)^\top \mathrm{d}\mathbf{W}_t & \mathbf{Y}_T = \phi(\mathbf{X}_T) \end{cases} \tag{6}$$

Then, with the Itô Formula (Exarchos and Theodorou, 2018), it is shown that if $\mathbf{v} \equiv v(t, \mathbf{X}_t)$ and $G(t, \mathbf{X}_t)$ are solution of the following PDE:

$$\frac{\partial v}{\partial t} + \frac{1}{2}\mathrm{Tr}(\nabla_{\mathbf{x}}^2 v G G^\top) + \nabla_{\mathbf{x}} v^\top f + h(t, \mathbf{x}, v, G^\top \nabla_{\mathbf{x}} v) = 0, \quad v(T, \mathbf{x}) = \phi(\mathbf{x}) \tag{7}$$

Then, with $\mathbf{Y}_t$ and $\mathbf{Z}_t$ the variables from (6), we have that :

$$v(t, \mathbf{X}_t) = \mathbf{Y}_t \quad \text{and} \quad G(t, \mathbf{X}_t)^\top \nabla_{\mathbf{x}} v(t, \mathbf{X}_t) = \mathbf{Z}_t$$

This give the possibility to solve either one of the problems, the SDEs (6) or the second-order parabolic PDE (7). This is convenient computational wise, as the Forward-Backward SDE method (Pereira et al., 2020) provides a way to solve second-order parabolic PDE.

Similarly, instead of directly solving the PDEs linked to the SB problem, it is possible to solve the underlying forward and backward SDEs. Using this framework for SBs, we obtain that:

$$\begin{cases} \mathrm{d}\mathbf{X}_t = (f + g\mathbf{Z}_t)\mathrm{d}t + g\mathrm{d}\mathbf{W}_t \\ \mathrm{d}\mathbf{Y}_t = \dfrac{1}{2}\mathbf{Z}_t^\top \mathbf{Z}_t \mathrm{d}t + \mathbf{Z}_t^\top \mathrm{d}\mathbf{W}_t \\ \mathrm{d}\widehat{\mathbf{Y}}_t = \left(\dfrac{1}{2}\widehat{\mathbf{Z}}_t^\top \widehat{\mathbf{Z}}_t + \nabla_{\mathbf{x}}.(g\widehat{\mathbf{Z}}_t - f) + \widehat{\mathbf{Z}}_t^\top \mathbf{Z}_t\right)\mathrm{d}t + \widehat{\mathbf{Z}}_t^\top \mathrm{d}\mathbf{W}_t \end{cases} \tag{8}$$

where, in the second-order parabolic PDE (7), v is replaced with $\log \Psi$ and $\log \widehat{\Psi}$. The constraints are $X_0 = x_0$ and $\mathbf{Y}_T + \widehat{\mathbf{Y}}_T = \log p_{\text{prior}}(\mathbf{X}_T)$.

Then, the nonlinear Feynman-Kac relations between the FBSDEs (8) and PDEs (4) are:

$$\mathbf{Y}_t = \log \Psi(t, \mathbf{X}_t), \quad \mathbf{Z}_t = g\nabla_{\mathbf{x}} \log \Psi(t, \mathbf{X}_t)$$
$$\widehat{\mathbf{Y}}_t = \log \widehat{\Psi}(t, \mathbf{X}_t), \quad \widehat{\mathbf{Z}}_t = g\nabla_{\mathbf{x}} \log \widehat{\Psi}(t, \mathbf{X}_t)$$

We also get that $\mathbf{Y}_t + \widehat{\mathbf{Y}}_t = \log(\Psi(t, \mathbf{X}_t)\widehat{\Psi}(t, \mathbf{X}_t)) = \log p_t^{\text{SB}}(\mathbf{X}_t)$.

Furthermore, by considering approximations of $\mathbf{Z}_t$ and $\widehat{\mathbf{Z}}_t$, with g known (by design), it becomes possible to approximate $\nabla_x \log \widehat{\Psi}(t, \mathbf{X}_t)$ and $\nabla_x \log \Psi(t, \mathbf{X}_t)$. In this context, $\mathbf{Z}_t$ and $\widehat{\mathbf{Z}}t$ can be interpreted as policies that are learned to align the distributions p_T and p_{prior}. While this requirement is imposed in the SB model, it is not present in SGM, where the prior distribution is approximated. Consequently, $\mathbf{Z}_t$ directs the forward process towards the prior distribution, while $\widehat{\mathbf{Z}}_t$ adjusts the backward process accordingly.

From there, Chen et al. (2023) show that, given a path sampled from the forward SDE of (8), the solutions to the backward SDEs at $t = 0$ provide an unbiased estimation of the log-likelihood of the data point x_0:

$$\mathbb{E}\left[\mathbf{Y}_0 + \widehat{\mathbf{Y}}_0 | \mathbf{X}_0 = x_0\right] = \log p_0^{\text{SB}}(x_0) = \log p_{\text{data}}(x_0) \tag{9}$$

Now, similarly to the log-likelihood of SGM models (3), we have that:

$$\begin{aligned} \log p_0^{\text{SB}}(x_0) &= \mathbb{E}\left[\mathbf{Y}_T + \widehat{\mathbf{Y}}_T - \int_0^T \mathrm{d}\mathbf{Y}_t + \widehat{\mathbf{Y}}_t \middle| \mathbf{X}_0 = x_0\right] \\ &= \mathbb{E}\left[\mathbf{Y}_T + \widehat{\mathbf{Y}}_T | \mathbf{X}_0 = x_0\right] \\ &\quad - \int_0^T \mathbb{E}\left[\frac{1}{2}\mathbf{Z}_t^\top \mathbf{Z}_t + \frac{1}{2}\widehat{\mathbf{Z}}_t^\top \widehat{\mathbf{Z}}_t + \nabla_{\mathbf{x}}.(g\widehat{\mathbf{Z}}_t - f) + \widehat{\mathbf{Z}}_t^\top \mathbf{Z}_t\right]\mathrm{d}t \end{aligned} \tag{9}$$

This allows them to derive the theorem that solves all our issues:

$$\log p_0^{\text{SB}}(x_0) = \mathbb{E}\left[\log p_T(\mathbf{X}_T)\right] - \int_0^T \mathbb{E}\left[\frac{1}{2}\|\mathbf{Z}_t\|^2 + \frac{1}{2}\|\widehat{\mathbf{Z}}_t\|^2 + \nabla_x \cdot (g\widehat{\mathbf{Z}}_t - f) + \widehat{\mathbf{Z}}_t^\top \mathbf{Z}_t\right]\mathrm{d}t \tag{10}$$

We now have an ELBO to maximize with every term tractable. That allows us to use two trainable networks to learn the optimal controls $\mathbf{Z}_t^\theta$ and $\widehat{\mathbf{Z}}_t^\phi$ only, that are the optimal forward and backward drifts for SB.

This is a generalisation of SGM, indeed, for $(\mathbf{Z}_t^\theta, \widehat{\mathbf{Z}}_t^\phi) = (0, gs_t)$, the SB log-likelihood objective is the same as (3). SBs allow for a wider array of drifts.

4 Going further : SB FB-SDE to solve PDEs

Neural networks can be trained to sample paths that solve partial differential equations (PDEs). In this section, we aim to explore the connection between the SB-FBSDE method and the solution of complex PDEs.

4.1 An introduction to the Hamilton-Jacobi-Bellman PDE

One widely known example of a PDE is the Hamilton-Jacobi-Bellman (HJB) equation. It is commonly used in optimal control theory, as its solution provides the sought-after value function for an optimal control problem. The HJB equation also plays a crucial role in dynamic programming by offering efficient methods for solving variational problems. The HJB PDE can be expressed as follows:

$$\frac{\partial V(t,x)}{\partial t} + \min_{u} \left\{ \frac{\partial V(t,x)}{\partial x} \times F(x,u) + C(x,u) \right\} = 0 \quad \text{s.t.} V(T,x) = \phi(x(T)) \tag{11}$$

Here, $\phi(\cdot)$ represents the function that determines the terminal state cost, $C(\cdot)$ is a quadratic cost function, and $F(x(t), u(t)) = \dot{x}(t), , \forall t \in [0,T]$.

For instance, we can consider the mathematical framework proposed by Pereira et al. (2019) for stochastic optimal control (SOC) problems, which can be formulated as:

$$J(t, \mathbf{x}(t); \mathbf{u}(t)) = \mathbb{E}_{\mathbb{Q}} \left[\int_t^T \left(q(s, \mathbf{x}(s)) + \frac{1}{2} \mathbf{u}(s)^\top \mathbf{R} \mathbf{u}(s) \right) \mathrm{d}s + \phi(\mathbf{x}(T) \right] \tag{SOC-HJB}$$

where

$$\begin{cases} \mathrm{d}\mathbf{x}(t) = (\mathbf{f}(t, \mathbf{x}(t)) + \mathbf{G}(t, \mathbf{x}(t))\mathbf{u}(t))\mathrm{d}t + \sigma \mathbf{G}(t, \mathbf{x}(t))\mathbf{u}(t)\mathrm{d}v(t) + \mathbf{\Sigma}(t, \mathbf{x}(t))\mathrm{d}\mathbf{w}(t) \\ \mathbf{x}(0) = \xi \end{cases}$$

and $\mathbf{w}$ is a vector of mutually independent one-dimensional variables following a standard Brownian motion (thus leading to the probability measure $\mathbb{Q}$). The objective is to find the function value for the SOC-HJB problem, given by:

$$\begin{cases} V(t, \mathbf{x}(t)) = \inf\limits_{\mathbf{u}(t) \in \mathcal{U}([t,T])} J(t, \mathbf{x}(t); \mathbf{u}(t)) \\ V(T, \mathbf{x}(T)) = \phi(\mathbf{x}(T)) \end{cases} \tag{12}$$

Consequently, the following partial differential equation arises:

$$\begin{cases} \dfrac{\partial V(t,\mathbf{x})}{\partial t} + q + \nabla_{\mathbf{x}} V(t,\mathbf{x})^\top \mathbf{f} - \dfrac{1}{2} \nabla_{\mathbf{x}} V(t,\mathbf{x})^\top \mathbf{G} \hat{\mathbf{R}}^{-1} \mathbf{G}^\top \nabla_{\mathbf{x}} V(t,\mathbf{x}) + \dfrac{1}{2} \mathrm{Tr}(\nabla_{\mathbf{x}}^2 V(t,\mathbf{x}) \mathbf{\Sigma} \mathbf{\Sigma}^\top) = 0 \\ V(T, \mathbf{x}(T)) = \phi(\mathbf{x}(T)) \end{cases} \tag{PDE-HJB}$$

where $\hat{\mathbf{R}} := \mathbf{R} + \sigma^2 \mathbf{G}^\top \nabla_{\mathbf{x}}^2 V(t,\mathbf{x})$ and the optimal control is $\mathbf{u}^\star(t,\mathbf{x}) = -\hat{\mathbf{R}}^{-1} \mathbf{G}(t,\mathbf{x})^\top \nabla_{\mathbf{x}} V(t,\mathbf{x})$. Then, PDE-HJB is a parabolic PDE.

One approach to tackling this PDE is by employing Itô's formula to construct an FBSDE scheme, as demonstrated in Pereira et al. (2019):

$$\begin{cases} \mathrm{d}X = f\,\mathrm{d}t + \mathbf{G}\mathbf{u}^\star\,\mathrm{d}t + C\mathrm{d}\epsilon \\ \mathrm{d}V = H(V)\,\mathrm{d}t + \nabla_{\mathbf{x}} V^\top \mathbf{G}\mathbf{u}^\star\,\mathrm{d}t + \nabla_{\mathbf{x}} V^\top\, C\mathrm{d}\epsilon \\ \mathrm{d}\nabla_{\mathbf{x}} V = H(\nabla_{\mathbf{x}} V)\,\mathrm{d}t + \nabla_{\mathbf{x}}^2 V \mathbf{G}\mathbf{u}^\star\,\mathrm{d}t + \nabla_{\mathbf{x}}^2 V C\mathrm{d}\epsilon \\ \mathbf{x}(0) = \xi, \quad V(T) = \phi(x(T)), \quad \nabla_{\mathbf{x}} V(T) = \phi(\mathbf{x}(T)) \end{cases} \tag{13}$$

Recent work by Han et al. (2018) incorporates deep learning into the FBSDE representation (similarly to SB-FBSDE) and demonstrates its feasibility and numerical efficiency. The resulting algorithm is known as Deep FBSDEs. Building upon the work in Han et al. (2018), Wang et al. (2019) improves the Deep-FBSDE approach by incorporating importance sampling within the FBSDE representation. Additionally, they propose a novel and scalable DNN architecture based on Long Short-Term Memory neural networks. This enhanced algorithm overcomes the limitations of the work in Han et al. (2018) in terms of scalability and training speed.

However, it is important to note that without a starting distribution, an ending distribution, and two coupled PDEs, it is not possible to establish a direct link between this problem and the Schrödinger Bridge problem. Therefore, SB-FB-SDE methods are not applicable in this case.

4.2 Mean-Field Games to create a link between HJB and SB FB-SDE

To bridge the gap between SB and PDEs, we introduce the Mean-Field Games theory. Caluya and Halder (2021) show that these two problems are naturally connected. Mean-field game (MFG) theory aims to solve a set of coupled partial differential equations (PDEs) that describe the behaviour of a large number of rational agents or players in an interactive system. At its most general form, the MFG equations consist of the following PDEs:

1. Hamilton-Jacobi-Bellman (HJB) equation:
$$-\frac{\partial u}{\partial t} + H(t, x, \nabla u, \rho) - \frac{1}{2}\nabla^2 u = F(x, \rho)$$
This equation governs the dynamics of the value function $u(t, x)$ of an individual agent. It represents the agent's optimization problem by considering the influence of the mean field ρ on the agent's decision-making process. The Hamiltonian function H captures the agent's preferences and constraints.
2. Fokker-Planck (FP) equation:
$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \nabla_\rho H(x, \nabla u, \rho)) - \frac{1}{2}\nabla^2 \rho = 0$$
This equation describes the evolution of the mean field $\rho(t, x)$ representing the distribution of the agents in the system. The function b captures the drift term that depends on the agents' behaviour, the gradient of the value function v, and the density ρ.

These PDEs form the core equations of MFG theory, capturing the interplay between individual optimization problems (HJB equation) and the evolution of the distribution of agents (FP equation).

It finds applications in various fields, including economics, finance, traffic modelling, biology and social sciences, among others.

In Liu et al. (2022), they link directly this MFG equations to the Schrödinger Bridge problem through two steps :

1. Replace the Hamiltonian by its linear: $H(x, \nabla u, \rho) := \frac{1}{2}\|\sigma \nabla u\|^2 - \nabla u^T f(x, \rho)$
$$\begin{cases} -\dfrac{\partial u(x,t)}{\partial t} + \dfrac{1}{2}\|\sigma \nabla u\|^2 - \nabla u^\top f - \dfrac{1}{2}\sigma^2 \Delta u = F(x, \rho) & \rho(x, 0) = \rho_0(x) \\ \dfrac{\partial \rho(x,t)}{\partial t} - \nabla \cdot \big(\rho(\sigma^2 \nabla u - f)\big) - \dfrac{1}{2}\sigma^2 \Delta \rho = 0 & \rho(x, T) = \rho_{\text{target}}(x) \end{cases} \tag{14}$$
2. Apply the Hopf-Cole transform:
$$\begin{cases} \psi(x, t) := \exp(-u(x, t)) \\ \widehat{\psi}(x, t) := \rho(x, t)\exp(u(x, t)) \end{cases} \tag{15}$$

With some additional calculation, we obtain the following PDEs:
$$\begin{cases} \dfrac{\partial \psi(x,t)}{\partial t} = -\nabla \psi^\top f - \dfrac{1}{2}\sigma^2 \Delta \psi + F\psi \\ \dfrac{\partial \widehat{\psi}(x,t)}{\partial t} = -\nabla \cdot (\widehat{\psi} f) + \dfrac{1}{2}\sigma^2 \Delta \widehat{\psi} - F\widehat{\psi} \end{cases} \quad \text{s.t.} \quad \begin{matrix} \psi(0, \cdot)\widehat{\psi}(0, \cdot) = \rho_0 \\ \psi(T, \cdot)\widehat{\psi}(T, \cdot) = \rho_{\text{target}} \end{matrix} \tag{16}$$

When $F(x, \rho) = 0$, we obtain exactly the two PDEs associated to the Schrödinger Bridge (4). Liu et al. (2022) uses that to generalize the idea of Schrödinger Bridge and take into account, $F(x, \rho)$ allowing them to use the identity (9) to derive new losses to train their network to solve Mean-Field Games.

4.3 Experiments on Mean-Field Games

Driven by the work of Liu et al. (2022), we choose to explore the possibility of this new SB-FBSDE generalized for Mean Field Games.

One important point to notice is this framework only solves MFG with hard distributional constraints (ρ_0, ρ_{target}).

Mean Field Games can, among other thing, describe how a crowd evolves. We will try to see, using the generalized framework of SB-FBSDE, if we can understand the mechanism of a crowd and also the limits of the framework.

4.3.1 The setup

To model the evacuation of a building using MFG, we consider a large population of individuals who need to exit the building safely and efficiently. Each individual aims to minimize their own evacuation time while avoiding collisions with others. The behaviour of the agents is captured by their strategies, which specify their movements through the building. The dynamics of the system can be described using PDEs derived from Hamilton-Jacobi-Bellman equations as described earlier.

If we re-use the equations (14) with u corresponding to the common strategy and ρ the density of people. The function F in the equation for the strategy represents a combination of obstacle effects such as walls, doors, or other physical constraints $F_{obstacle}$ that affect the strategy based on the spatial variable x, as well as the interactions among individuals $F_{congestion}$ captured through the density ρ.

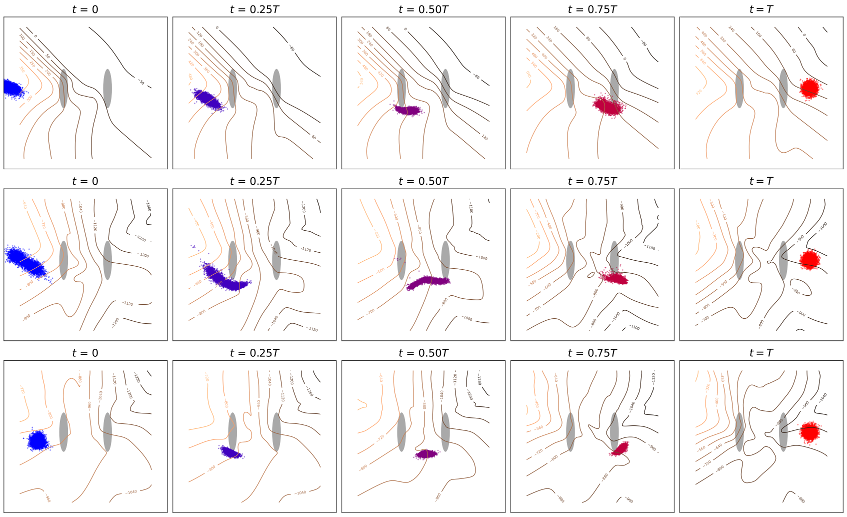

Figure 1: Forward process. From top to bottom, Step n° 10, 20, 40 of forward-backward pass.

On the Figure 1, we can see that at step 10, half of the potential ("policy") has not been learned yet. At $t = 0.75T$, the distribution comes back up and is partially stuck against the second wall. At step 20, there is still some deformation of the distribution toward the interior of the obstacles, and the population is more spread out. Lastly, at step 40, the trajectory is optimized and compact.

Further work on our GitHub highlight the impact of the congestion parameter K:

$$F_{\text{congestion}}(x, \rho) = \mathbb{E}_{y \sim \rho}\left[\frac{K}{\|x - y\|^2 + 1}\right]$$

It also underlines, as described in Zuriguel et al. (2011), that adding an obstacle just before the exit help reduce the density.

Conclusion

In conclusion, Chen et al. (2023) have proposed a new method leveraging the power of Schrödinger Bridges in the context of generative modelling. To do so, they had to use tools from Stochastic Optimal Control to solve complex PDEs. In this process, they made a link between Schrödinger Bridges and Score-based Generative Models using the lower-bound of the log-likelihood developed in Song et al. (2021).

We then explored the use of Schrödinger Bridges Forward Backward SDEs to solve the more general Mean-Field Games Caluya and Halder (2021); Liu et al. (2022), that consist of the Hamilton-Jacobi-Bellman and Fokker-Planck PDEs, placing this method at the intersection of generative modelling, entropy regularized optimal transport and reinforcement learning.

The use of Schrödinger Bridges is an elegant solution to generative modelling, backed by a well-founded mathematical framework and detailed practical implementations. However, it is hard to quantify if it actually outperforms current SGM models on generative tasks, without exhaustive and complex high dimensional experiments. But this approach is clearly interesting and allows for a broader interpretation of SGMs.

References

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 17695–17709. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/940392f5f32a7ade1cc201767cf83e31-Paper.pdf. 1

Gefei Wang, Yuling Jiao, Qian Xu, Yang Wang, and Can Yang. Deep generative learning via schrödinger bridge. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10794–10804. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/wang21l.html. 1

Francisco Vargas, Pierre Thodoroff, Austen Lamacraft, and Neil Lawrence. Solving schrödinger bridges via maximum likelihood. *Entropy*, 23(9):1134, 2021. 1

Erwin Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. In *Annales de l'institut Henri Poincaré*, volume 2, pages 269–310, 1932. 1, 2

Tianrong Chen, Guan-Horng Liu, and Evangelos A. Theodorou. Likelihood training of schrödinger bridge using forward-backward sdes theory, 2023. 1, 4, 8

Solomon Kullback. Probability densities with given marginals. *The Annals of Mathematical Statistics*, 39:1236–1243, 1968. 1

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models, 2021. 2, 8

Ioannis Exarchos and Evangelos A. Theodorou. Stochastic optimal control via forward and backward stochastic differential equations and importance sampling. *Automatica*, 87:159–165, 2018. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2017.09.004. URL https://www.sciencedirect.com/science/article/pii/S0005109817304740. 3

Marcus Pereira, Ziyi Wang, Tianrong Chen, Emily Reed, and Evangelos Theodorou. Feynman-kac neural network architectures for stochastic control using second-order fbsde theory. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger, editors, *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 728–738. PMLR, 10–11 Jun 2020. URL https://proceedings.mlr.press/v120/pereira20a.html. 3

Marcus A Pereira, Ziyi Wang, Tianrong Chen, Emily Reed, and Evangelos A Theodorou. Deep 2fbsdes for systems with control multiplicative noise, 2019. 5

Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, aug 2018. doi: 10.1073/pnas.1718942115. URL https://doi.org/10.1073%2Fpnas.1718942115. 5

Ziyi Wang, Marcus Pereira, Ioannis Exarchos , and Evangelos Theodorou. Learning deep stochastic optimal control policies using forward-backward SDEs. In *Robotics: Science and Systems XV*. Robotics: Science and Systems Foundation, jun 2019. doi: 10.15607/rss.2019.xv.070. URL https://doi.org/10.15607%2Frss.2019.xv.070. 5

Kenneth F. Caluya and Abhishek Halder. Wasserstein proximal algorithms for the schrödinger bridge problem: Density control with nonlinear drift, 2021. 6, 8

Guan-Horng Liu, Tianrong Chen, Oswin So, and Evangelos A. Theodorou. Deep generalized schrödinger bridge, 2022. 6, 7, 8

Iker Zuriguel, Alvaro Janda, Angel Garcimartín, Celia Lozano, Roberto Arévalo, and Diego Maza. Silo clogging reduction by the presence of an obstacle. *Phys. Rev. Lett.*, 107:278001, Dec 2011. doi: 10.1103/PhysRevLett.107.278001. URL https://link.aps.org/doi/10.1103/PhysRevLett.107.278001. 7