# Rapport de Stage

Thomas Seror

Oxford University - Department of Statistics

## Contents

# Cadre du stage

J'ai effectué mon stage de Mathématiques de M1 au sein du département de statistiques de l'université d'Oxford au Royaume-Uni. À cette occasion, j'ai pu découvrir et expérimenter le monde de la recherche universitaire durant 4 mois (du 1er mars au 30 juin 2022).

Étant intéressé par les statistiques et l'apprentissage automatique, je me suis tourné vers le laboratoire de statistiques d'Oxford au sein duquel j'ai été encadré par Pr. Arnaud Doucet. Après plusieurs réunions virtuelles avant le début de mon stage, j'ai choisi de travailler sur les modèles génératifs de processus stochastiques. L'ensemble de mon travail de recherche, bien que parfois théorique, était surtout axé sur l'aspect applicatif qui m'intéressait davantage. Tout au long de mon stage, j'ai eu l'occasion d'utiliser les GPU (processeurs graphiques) et les serveurs du laboratoire pour entraîner des modèles, ce qui m'a permis de me familiariser avec les outils utilisés dans ce domaine.

J'ai eu la chance d'aller sur place à Oxford pour rencontrer l'équipe de chercheurs postdoctoraux qui travaillent sur le même sujet. Toutes les semaines, nous faisions une réunion d'1h30 avec l'équipe dont je faisais partie; c'est le moment où je présentais mes avancées et les difficultés que je rencontrais. Le fait de travailler en équipe s'est avéré être un vrai atout; en effet, cela m'a permis de surmonter des points de blocage dans mes recherches à plusieurs reprises.

Le but de mes recherches consistait à construire des modèles génératifs dans des espaces de fonctions. Dans la suite de ce rapport, je vais dans un premier temps décrire succinctement les méthodes de modèles génératifs par diffusion (score-based generative models), qui sont des modèles très efficaces dans des espaces vectoriels de dimensions finis. Dans un second temps, je vous présenterai la méthode la plus utilisée pour réaliser des modèles génératifs sur des espaces de fonctions (de dimension infinie), qui consiste à utiliser des processus gaussiens. Enfin, vous trouverez l'ébauche d'article remise à mon équipe, qui synthétise les résultats de mon travail mettant en relation les modèles de diffusion et les processus stochastiques.

Je tiens à remercier chaleureusement le professeur Arnaud Doucet pour avoir accepté de m'encadrer durant ces quatre mois et pour m'avoir permis de découvrir la vie de laboratoire en Angleterre. Je remercie également Emile Mathieu, qui a su me guider et répondre à toutes mes questions lors de ce stage, ainsi que Valentin de Bortoli et Michael Hutchinson avec qui j'ai eu la chance de collaborer.

# Score-based generative modelling & processus stochastiques

## 1 Introduction au score-based generative modelling

La théorie que je vais vous présenter succinctement s'appuie principalement sur Song et al. (2021b).

L'idée derrière les score-based generative models (SGM) est de construire un processus de diffusion $\{X(t)\}_{t=0}^{T}$ indexé par une variable continue $t \in [0, T]$, tel que $X(0) \sim p_0$, pour lequel on a un dataset d'échantillons i.i.d. et $X(T) \sim p_T$, une densité dont on peut facilement générer des échantillons. $p_0$ est la distribution des données (data distribution) et $p_T$ est la distribution a priori (prior distribution).

Des choix typiques de processus de diffusion sont les processus de Wiener ou d'Ornstein-Uhlenbeck. On sait qu'un processus de diffusion est un processus de Markov par inversion du temps Haussmann, Pardoux (1986). Ainsi, nous pouvons générer des échantillons de la data distribution $p_0$ en échantillonant des bruits gaussiens $X(T) \sim p_T$ et en simulant le processus de diffusion inverse (processus inverse).

### 1.1 Processus forward: perturbation des données

Un processus de diffusion $\{X(t)\}_{t\in[0,T]}$ peut être modélisé comme la solution d'une équation différentielle stochastique (SDE) d'Ito:

**Definition 1** (Ito SDE)**.**

$$dX_t = f(X_t, t)dt + G(X_t, t)dB_t$$

*où $B_t$ est un mouvement brownien standard, $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ un coefficient de "drift" et $G : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^{d\times d}$ un coefficient de diffusion.*

Les SDEs de cette forme peuvent être discrétisées et simulée via une discrétisation d'Euler-Maruyama:

**Definition 2** (Ito SDE discretisation)**.** *Pour les temps $0 = \tau_0 < \cdots < \tau_N = T$, $\Delta t_i = \tau_{i+1} - \tau_i$, et $\Delta B_k \sim B_{\tau_{k+1}} - B_{\tau_k}$, une SDE d'Ito peut être discrétisée comme:*

$$X_{k+1} = X_k + f(X_k, \tau_k)\Delta t_k + G(X_k, \tau_k)\Delta B_k$$

### 1.2 Processus inverse: génération des données

Le processus inverse $\{Y_t\}_{t\in[0,T]} = \{X_{T-t}\}_{t\in[0,T]}$ est également un processus de diffusion, et on a le résultat suivant:

**Theorem 3** (Reverse-time Ito SDE Anderson (1982)). *Pour une SDE d'Ito de la forme de la Definition 1, avec certaines propriétés de régularité qui garantissent l'existence et l'unicité d'une solution Kushner (1974), et l'existence d'une densité de probabilité $p(X_t, t)$ pour $t_0 \leq t \leq T$ comme solution unique de l'équation de Kolmogorov associée. On définit aussi $\bar{W}_t$ par $\bar{W}_0 = 0$ et*

$$d\bar{W}_t = dW_t + \nabla \cdot g(Y_t, t) + g(Y_t, t)\nabla \log p(Y_t, t)$$

*alors un modèle d'inversion du temps pour $X_t$ est défini par*

$$dY_t = \bar{f}(Y_t, t)dt + G(Y_t, t)d\bar{W}_t$$

*où*

$$\bar{f}(Y_t, t) = f(Y_t, t) - \nabla \cdot [G(Y_t, t)G(Y_t, t)^\top] - G(Y_t, t)G(Y_t, t)^\top \nabla \log p(Y_t, t) \tag{1}$$

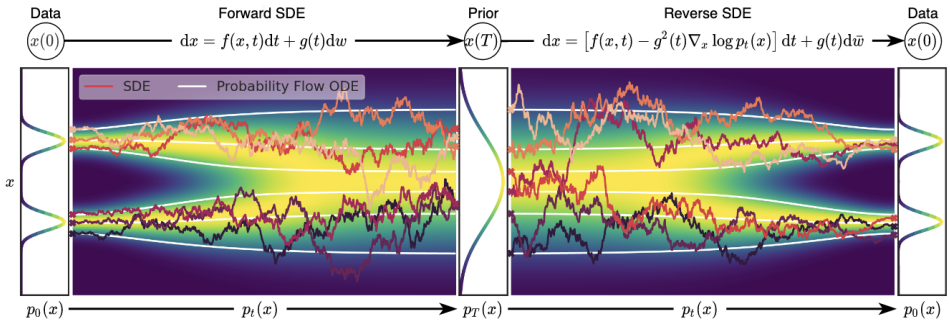


Figure 1: Exemple 1D des processus forward et inverse

## 1.3 Cas choisi

Dans la suite nous choisisons de nous placer dans le cas particulier défini par :

$$\begin{aligned} f(X_t, t) &= -\beta(t)X_t \\ G(X_t, t) &= \sqrt{2\beta(t)} \\ \beta(t) &: \mathbb{R} \to \mathbb{R}_+^* \end{aligned}$$

i.e.

$$dX_t = -\beta(t)X_t dt + \sqrt{2\beta(t)}dB_t$$

Pour cette SDE, nous pouvons calculer la distribution limite (invariante)

**Lemma 4** (Distribution limite)**.** *Une SDE d'Ito de la forme*

$$dX_t = -\beta X_t dt + \sqrt{2\beta} dB_t$$

*admet la distribution limite*

$$\lim_{t\to\infty} p(X_t) = \mathcal{N}(0, I)$$

La forme de Equation (1) devient

$$\begin{aligned}\bar{f}(Y_t, t) &= -\beta(t) X_t - 2\beta(t)\nabla \log p(Y_t, t) \\ &= -\beta(t)(X_t + 2\nabla \log p(Y_t, t))\end{aligned}$$

### 1.4 Training

Notre modèle génératif consiste à inverser le processus de diffusion à partir d'un bruit gaussien pour générer des échantillons.

Malheuresement, nous ne disposons pas d'expression analytique du *Stein score* $\nabla \log p(X_t, t)$. Nous devons alors l'estimer, ici par un réseau de neurones, $s_\theta : \mathbb{R}^n \times [0, T] \to \mathbb{R}^n$ tel que $s^\star_\theta(X_t, t) \approx \nabla_{X_t} \log p_t(X_t)$.

Puisque le Stein score satisfait

$$\nabla X_t \log p_t(X_t) = \mathbb{E}_{p_{0|t}} \left[\nabla_{X_t} \log p(X_t|X_0)\right]$$

le paramètre optimal $\theta^\star$ est choisi par optimisation tel que

$$\theta^\star \in \arg\min_\theta \ \mathbb{E}_t \left\{ \lambda(t) \mathbb{E}_{X(0), X(t)} \left[ \left| s_\theta(X(t), t) - \nabla_{X(t)} \log p(X(t)|X(0)) \right|_2^2 \right] \right\}$$

avec $\lambda : [0, T] \to \mathbb{R}^+$ (un hyperparamètre à choisir), $t \sim U([0, T])$, $X(0) \sim p_0(X)$ (la densité de notre dataset) et $X(t) \sim p(X(t)|X(0))$ (qui est normalement distribuée et dont la moyenne et la variance sont connues).

## 2 Processus Stochastiques

### 2.1 Définitions, remarques

**Definition 5** (Processus Stochastique)**.** *Soit $(\Omega, \mathcal{F}, P)$ un espace de probabilité, $X$ un ensemble d'indices arbitraire et $Y$ un espace métrique. Un processus stochastique (ou aléatoire) est une famille de variables aléatoires (i.e., des applications mesurables) définies sur le même espace de probabilité $(\Omega, \mathcal{F}, P)$ indexée par $X$ et à valeurs dans $Y$. Un processus stochastique est noté par $\{f_x\}_{x\in X}$. La valeur de la variable aléatoire $f_x$ en un certain $\omega \in \Omega$ est désignée par $f_x(\omega)$.*

**Definition 6** (Réalisation d'un processus stochastique)**.** *Soit $Y^X$ l'ensemble des applications définies sur $X$ en tout point et à valeurs dans $Y$. Soit $\omega \in \Omega$ et désignons par $f(\omega) \in Y^X$ l'application $x \mapsto f_x(\omega)$. Une telle application est appelée réalisation du processus stochastique $\{f_x\}_{x \in X}$.*

Les processus stochastiques permettent de représenter une évolution à temps discret ou continu d'une fonction aléatoire. Ces outils sont d'un grand intérêt, en effet, ils permettent de modéliser un très grand nombre de phénomènes qui semblent varier de manière aléatoire, comme par exemple des cours de bourse, l'évolution d'une population de bactéries ou bien le mouvement d'une molécule de gaz.

Notre but est de créer des modèles génératifs sur des ensembles de processus stochastiques. Le problème peut-être formulé comme tel :

Soit $f^{(1)}, ..., f^{(n)}$ des réalisations d'un processus stochastique, dont nous connaissons seulement un certain nombre $L$ d'observations $\{f^{(i)}(x_k)\}_{k=1,...,L}$ pour chaque réalisation $f^{(i)}$. Nous supposons que $f^{(i)} \overset{\text{iid}}{\sim} p_0$ où $p_0$ est une densité inconnue sur $Y^X$, notre but est de générer de nouveaux échantillons $f \sim p_0$.

## 2.2 Processus Gaussiens

Une approche utilisée dans la littérature pour ce problème est celle des processus gaussiens.

**Definition 7** (Processus Gaussien)**.** *Un processus stochastique $\{f_x\}_{x \in X}$ est gaussien si et seulement si pour tout ensemble fini d'indices $x_1, ..., x_k \in X$, la variable*

$$\mathbf{F}_{x_1,...,x_k} = (f_{x_1}, ..., f_{x_k})$$

*est gaussienne.*

Une propriété importante du processus gaussien est qu'il est entièrement défini par son processus moyen et sa fonction de covariance. Bishop (2006)

Il est donc possible d'effectuer une régression à partir du processus moyen observé $\hat{m}$ et en construisant une fonction de covariance $\hat{k}$ à partir des observations. On estimera alors la densité $p_0$ par le processus gaussien de moyenne $\hat{m}$ et de fonction de covariance $\hat{k}$. La méthode de construction de $\hat{k}$ est détaillée dans Rasmussen (2004).

Cependant cette méthode atteint rapidement ses limites. De la même manière que certaines variables aléatoires ne peuvent pas être estimées par des variables gaussiennes, certains processus stochastiques ne peuvent pas être estimées par des processus gaussiens. Un exemple simple est la multimodalité qui ne peut pas s'exprimer à partir de variables gaussiennes unimodales.

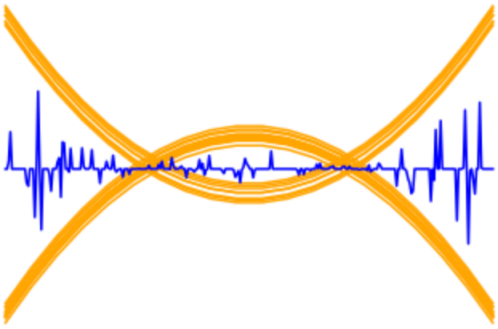

Figure 2: Tentative d'estimer un processus stochastique bimodal (dont plusieurs réalisations sont affichées en orange) par un processus gaussien (en bleu)

### 2.3 Notre approche

Il ressort de ce qui précède que se limiter à des processus gaussiens est insuffisant pour estimer des processus stochastiques quelconques.

Les SGM (score-based generative models) sont extrêmement efficaces pour estimer des densités de probabilités dans $\mathbb{R}^d$ (Song et al. (2021b)). Nous aimerions tirer parti de leur efficacité dans les espaces de fonctions, mais les SGM ne sont pas adaptées à des densités dans des espaces vectoriels de dimension infini.

La partie suivante correspond à l'ébauche d'article remise à mon équipe; elle consiste en une synthèse du résultat de mes recherches, dans laquelle nous détaillons notre approche. Celle-ci consiste à faire de la réduction dimensionnelle de manière intelligente pour se ramener à $\mathbb{R}^d$ et appliquer des SGM.

# Score-based neural processes

Thomas Seror

**Abstract**

Score-based generative modeling (SGM) has proven to be a very effective method for modeling density on finitely-dimensional spaces. Using dimensionality reduction, we leverage the efficiency of SGM to learn densities in functional spaces. We demonstrate our method's effectiveness for modeling various multimodal datasets.

***Keywords***— Riemannian manifold, Generative Modelling, Score-based generative models, Diffusion, Time-reversal

## 1 Introduction

Score-based generative models are a strong framework for generating samples from finitely-dimensional spaces. We are interested in generating samples from the infinitely-dimensional space of stochastic processes. One example of generative model for functional data is the Gaussian process, it is widely studied as it is very flexible but it has many limitations, such as an inability to express multimodality and heteroskedasticity.

Our idea, as detailed in 2, is to reduce the infinitely-dimensional space of stochastic processes to a finitely-dimensional space, cleverly chosen for the dataset of functions we want to generate. We can then apply standard score-based generative models. The choice of the reduced space of functions is key as we want to encapsulate the most informations on the dataset in a finite - preferably low - number of spectral coordinates. In 3, we show how our model successfully works on multimodal distributions that a GP fails to fit.

## 2 Method

### 2.1 Stochastic processes

Our approach is to model stochastic processes (SPs) in spectral space. That is, we will use an orthonormal basis of deterministic continuous functions $e_k$, and decomposes SPs into a collection of real-valued random variables $Z_k$ such that a centred process $f$ can be written as a realisation of the random variable

$$x \mapsto \sum_{k=1}^{\infty} Z_k e_k(x)$$

However, score-based diffusion models only work for finitely dimensional spaces. A solution to this problem is to carefully choose the basis of function $(e_k)_k$ such that we can take a truncation of the former sum without jeopardizing the quality of the generated samples.

## 2.2 Karhunen-Loeve theorem

If we have access to the covariance function of the dataset, we can do dimensionality reduction using the Karhunen-Loeve decomposition. In the same fashion as PCA, it yields the best possible basis in the sense that any (eigenvalue-ordered) subset of that basis minimises the total mean squared error.

Let $F = \{F(x)\}_{x \in X}$ a real-valued second-order random process with mean function $m$ and continuous covariance function $(x, x') \in X^2 \mapsto K_F(x, x')$ with $X$ compact. We can also use an arbitrary kernel $K$. Let $(e_k)_k$ be an orthonormal basis on $L^2(X)$ formed by the eigenfunctions of the linear operator $T_K : f \mapsto \int_x K(x, \cdot) f(x) \mathrm{d}x$ with associated eigenvalues $\lambda_k$, i.e. $T_K e_k = \lambda_k e_k$.

Then for every $x \in x$, $F(x)$ admits the following representation (Sullivan, 2015, Theorem 11.4)

$$F(x) = m(x) + \sum_{k=1}^{\infty} Z_k \sqrt{\lambda_k} e_k(x) \tag{2}$$

where the convergence is in $L^2$, uniform in $x$, with

$$Z_k = \langle F(\cdot), \sqrt{\lambda_k} e_k(\cdot) \rangle_{L^2(X)} = \int_X (F(x) - m(x)) \sqrt{\lambda_k} e_k(x) \mathrm{d}x. \tag{3}$$

Furthermore, if $K = K_F$, we have $\mathbb{E}[Z_k] = 0, \forall k \in \mathbb{N} \quad \text{and} \quad \mathbb{E}[Z_i Z_j] = \delta_{ij}, \forall (i, j) \in \mathbb{N}^2$. For the specific case where $F$ is a Gaussian process (GP), we have $Z_k \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$.

**Consistent stochastic process** Given the decomposition for the stochastic process from Equation (2), we have that observations $\{y_1, \ldots, y_n\}$ are independent conditional on the realisation $F$, that is on the realisation $Z \sim p(Z)$. Indeed the joint distribution of multiple observations $y = \{y_1, \ldots, y_n\}$ at locations $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ is given by $p(y|\boldsymbol{x}) = \int p_\theta(F) \prod_i p(y_i|F, x_i) dF = \int p_\theta(Z) \prod_i p(y_i|Z, x_i) dZ$ where $p_\theta(Z)$ is the learnt SGMs and $p_\theta(F)$ is the density induced by Equation (2)—see **??**. The learnt stochastic process model is thus both *exchangeable* and *consistent*.

## 2.3 Spectral projection pre-processing

In practice we only know a finite number $L$ of points per process, we need to approximate the eigensystem $\{(\lambda_k, e_k)\}_{k=1,\ldots,L}$

Let $K$ be a (known) kernel, the eigensystem of $T_K$ is defined by the following set of equations, $\forall k = 1, \ldots, N$

$$\lambda_k e_k(x) = T_K e_k(x) = \int_X K(x, x') e_k(x') dp(x') \approx \frac{1}{L} \sum_{n=1}^{L} K(x, x_n) e_k(x_n) \tag{4}$$

with $\boldsymbol{x} = \{x_k\}_{k=1,\ldots,L} \in X$ the known points of the process.

Plugging in $\boldsymbol{x}$, we get the following eigenvalue problem $\hat{\lambda}_k \hat{\mathbf{e}}_k = \frac{1}{L} \mathbf{K} \hat{\mathbf{e}}_k$, where $\hat{\mathbf{e}}_k = e_k(\boldsymbol{x})$ and $\mathbf{K}$ is the $L \times L$ Gram matrix with entries $\mathbf{K}_{ij} = K(x_i, x_j)$. Solving for the eigensystem of $\frac{1}{L}\mathbf{K}$ yields estimates $\hat{\lambda}_k$ of $L$ non-negative eigenvalues $\lambda_k$ such that $\hat{\lambda}_k \xrightarrow{L \to \infty} \lambda_k$ (Baker, 1979, Theorem 3.4). Then one can plug back the

solved eigensystem $\{(\hat{\lambda}_k, \hat{\mathbf{e}}_k)\}_{k=1,\dots,L}$ into Equation (4), to get an estimate of the eigenfunctions of $T_K$:

$$e_k(x) \approx \hat{e}_k(x) \triangleq \hat{\lambda}_k^{-1} \frac{1}{L} \sum_{n=1}^{L} k(x, x_n) \hat{\mathbf{e}}_n \quad \forall n = 1, \dots, N$$

The random variables $Z_k$ can in turn be estimated by plugging in this estimate in Equation (5). That is for each process realisation $f^i \sim p$, one usually only has access to finite evaluations $\{(y_m^i, x_m^i)\}_{m=1,\dots,M}$ with $y_m^i = f(x_m^i)$, then

$$Z_k^i \approx \hat{z}_k^i \triangleq \frac{1}{M} \sum_{m=1}^{M} (y_m^i - \bar{y}_m) \sqrt{\hat{\lambda}_k} \hat{e}_k(x_m^i) \quad \forall k = 1, \dots, N, \tag{5}$$

with $\bar{y}_m$ the empirical mean of $Y_x$ at $x = x_m$.

Ideally $K = K_Y$ the covariance of our process $Y$, but if we don't know $K_Y$ we can treat $K$ as an hyperparameter.

**Truncation order** For a given eigensystem $\{(\hat{\lambda}_k, \hat{\mathbf{e}}_k)\}_{k=1,\dots,L}$, we define the truncation order $N$ as $N = \min_{n \in \mathbb{N}} \left( \sum_{k=1}^{n} \hat{\lambda}_k \geq t \sum_{k=1}^{L} \hat{\lambda}_k \right)$ where $t \in [0, 1]$ is a threshold parameter, fixed to $t = 0.99$ for our experimental results. Alternatively, we can treat the truncation order $N$ as an hyperparameter.

## 2.4 Conditioning

Modelling the distribution $p(f)$ is of limited interest, one is often interested in the conditional $p(f|\{y_m^c, x_m^c\}_{m \in C})$ given a context of samples.

One approach is the one taken by Dutordoir et al. (2022); Trippe et al. (2022), which rely on *augmenting* the evaluated state with the (forward diffused) context state.

Another approach proposed in Song et al. (2021b), is to learn a conditional score network, that is in our setting: $\mathbf{s}_\theta(t, \mathbf{z}; \{y^c, x^c\})$ with $\mathbf{z} = (z_1, \dots, z_N)$. As the context is a set, the score network should be permutation invariant w.r.t. it, i.e. $\mathbf{s}_\theta(t, \mathbf{z}; \sigma(\{y^c, x^c\})) = \mathbf{s}_\theta(t, \mathbf{z}; \{y^c, x^c\})$ $\forall \sigma \in \Sigma_C$. Similarly to neural processes, one can enforce this with a self-attention layer (permutation equivariant) followed with an (permutation invariant) aggregator (e.g. mean).

## 2.5 Training

When dealing with a dataset, the spectral coordinates $\{\hat{z}_k^i\}$ can be computationally expensive to estimate. This is why in practice, we will pre-process the dataset to get a converted dataset in the spectral space, which will be stored in CSV. Score-based diffusion models (SGMs) will be used on the spectral space to fit the real-valued random variables $Z_k$, using the distribution of $\hat{z}_k$.

When using the covariance function, the Karhunen-Loeve theorem states that $\mathbb{E}(Z_k^2) = 1$. But when using a different kernel, it is not always true. SGMs work better for distributions with variance 1, this is why we choose to reduce the spectral dataset such that $\mathrm{Var}(\hat{z}_k) = 1 \quad \forall k$

In the training, when using a standard DSM loss, the SGM will treat each coordinate of the spectral space as equal. However, this is not the case here as

each coordinate contributes more to the stochastic process than the next one. This can lead to extremely long training times to get a good fit. An easy way to solve this issue is to prioritize the first coordinates by weighting each contribution of a coordinate to the loss function by the corresponding estimated eigenvalue $\hat{\lambda}_k$.

In practice we choose the weight of each coordinate to be $\hat{\lambda}_k^\alpha$ where $\alpha \geq 0$ is an hyperparameter.

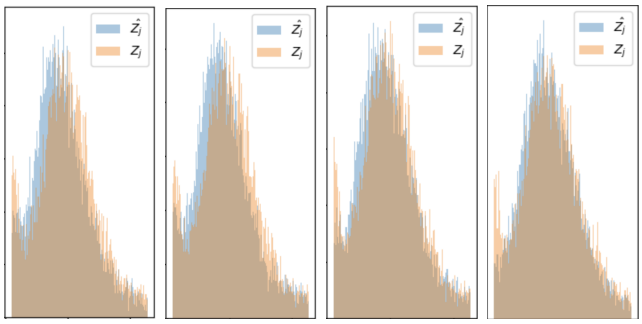


Figure 3: $\hat{z}_0$ density for MNIST dataset (see 3.2) for $\alpha = 0, 1, 1.5, 1.75$ (left to right)

# 3 Experimental results

In this section, we present the different generated samples of our model. First we look in 3.1 at results generated from two one-dimensional datasets, one synthetic and one from real data.

Next we will look in 3.2 at how our model performs for a 2-dimensional dataset.

## 3.1 1D datasets

**Quadratic** The first dataset we study is the `quadratic` dataset from Lim et al. (2022), a very simple dataset designed to test our model's capacity to generate bi-modality. A sample $f^{(i)}$ can be written as $f^{(i)}(x) = s^{(i)}x^2 + \epsilon^{(i)}$ where $s^{(i)} \sim \mathcal{U}(\{-1, 1\})$ and $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$. We can see on the next plot that Gaussian processes (GP) modelling perform very poorly on this dataset as GPs are unable to express bi-modality.

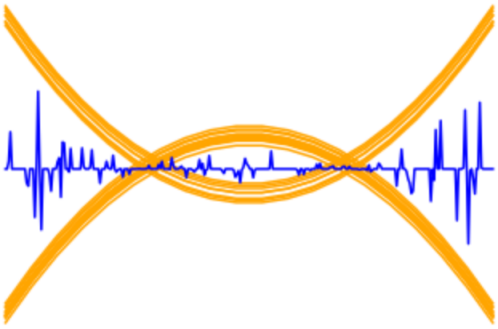

Figure 4: GP fit of the `quadratic` dataset - in orange the dataset, in blue the GP fit

Here are the generated samples of our model, using a squared-exponential kernel with lengthscale 5, and truncation order $N = 5$. We feature generated samples in the original space $X = [-10, 10]$ and in the spectral space, in addition to the first eigenvalues/eigenfunctions $(\lambda_k, e_k)$ used. We can see qualitatively on the plot that bi-modality is very well expressed in the original space and in the spectral space.

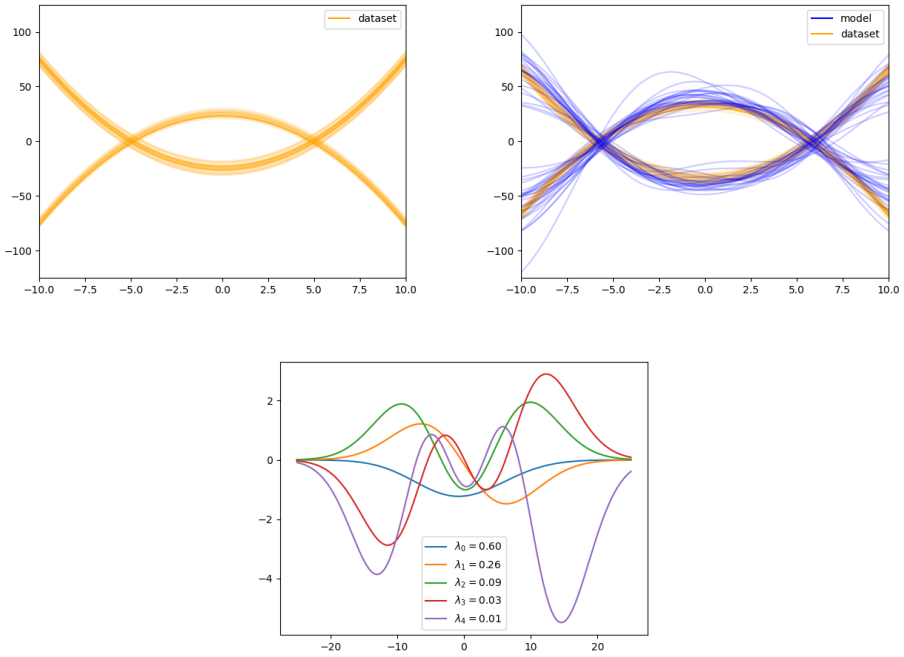


Figure 5: `Quadratic`, dataset only (left), with generated samples (right), first eigenfunctions (bottom)

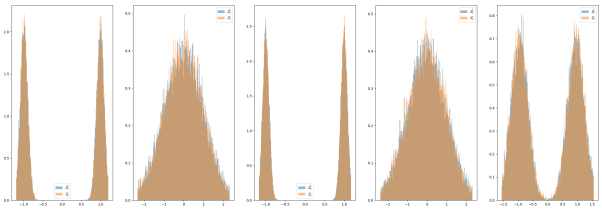

Figure 6: Generated samples for the `quadratic` dataset in the spectral space

**Melbourne** The second one-dimensional dataset we study is `Melbourne` [1] is a real-life multimodal dataset where each sample is the number of pedestrians on one street in Melbourne (randomly chosen from a pool of 10 streets), recorded throughout different times of the day. The dataset is composed of 2271 samples, where 1816 of them are used for training, the rest for testing. Here are the generated samples of our model, using the empirical covariance matrix as a kernel, and truncation order $N = 24$

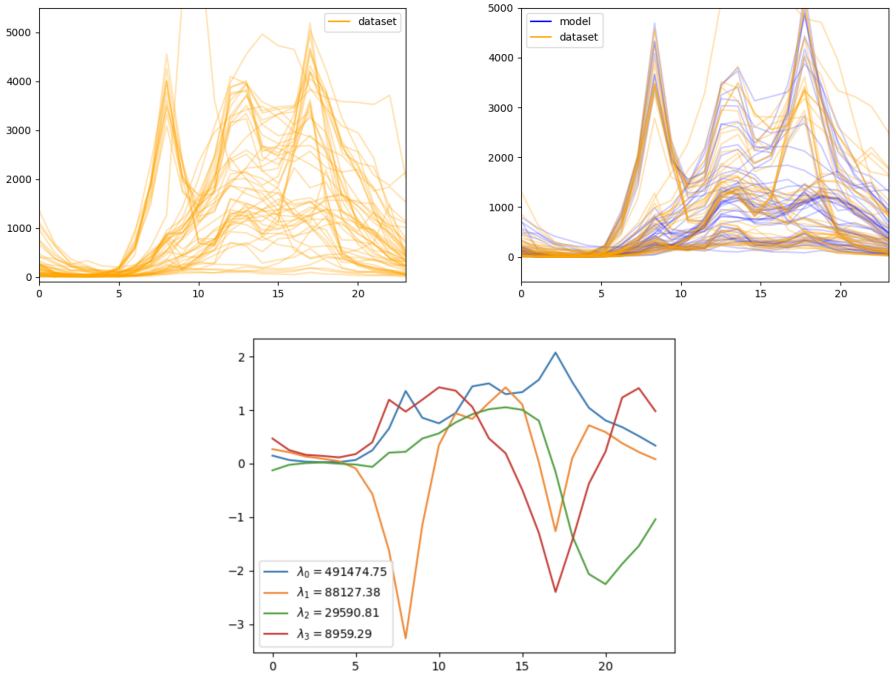


Figure 7: `Melbourne`, dataset only (left), with generated samples (right), first eigenfunctions (bottom)

[1] http://www.timeseriesclassification.com/description.php?Dataset=MelbournePedestrian

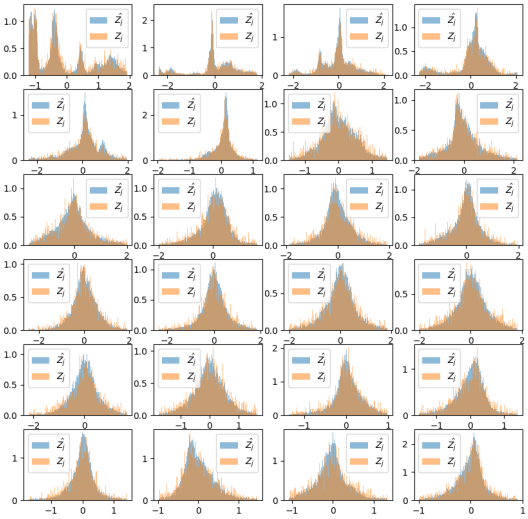

Figure 8: Generated samples for the `Melbourne` dataset in the spectral space

## 3.2 MNIST

We would like to test our model on a 2-dimensional dataset. We choose to use the MNIST handwritten digits database [2], with a 48000 samples training set, and a 12000 samples test set. The original space is $X = [\![1, 28]\!]^2$, and each pixel has a value in $Y = [0, 255]$.

**Results** For $N = 77$, using the covariance matrix

[2] http://yann.lecun.com/exdb/mnist/

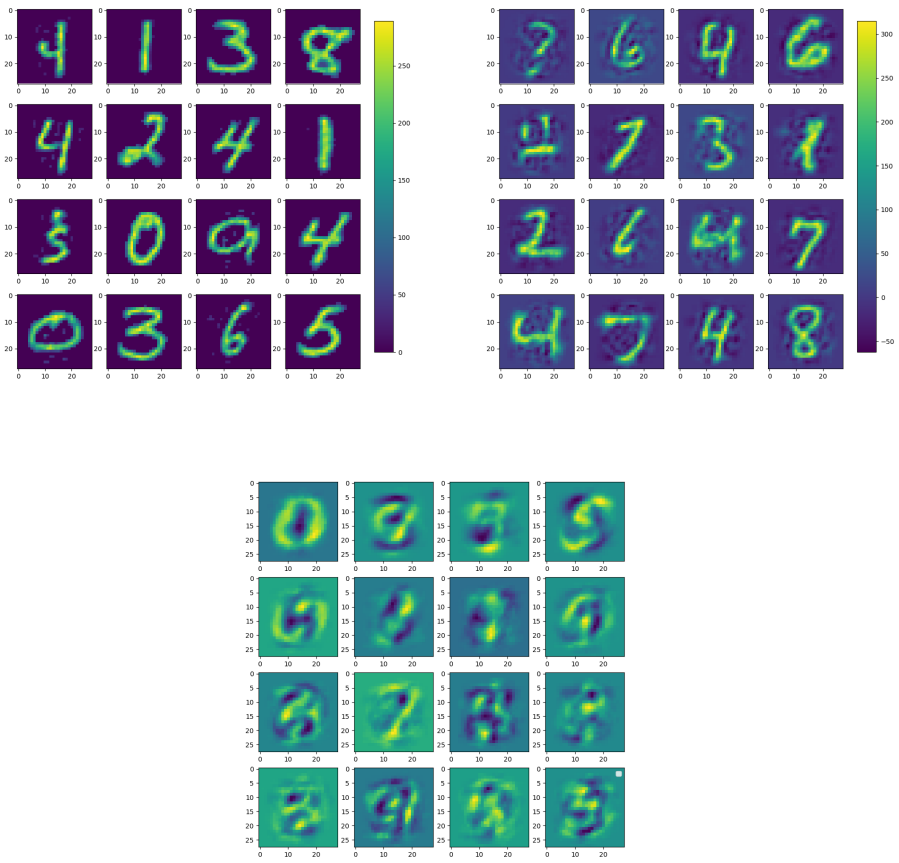

Figure 9: MNIST, dataset (left), generated samples (right), first eigenfunctions (bottom)

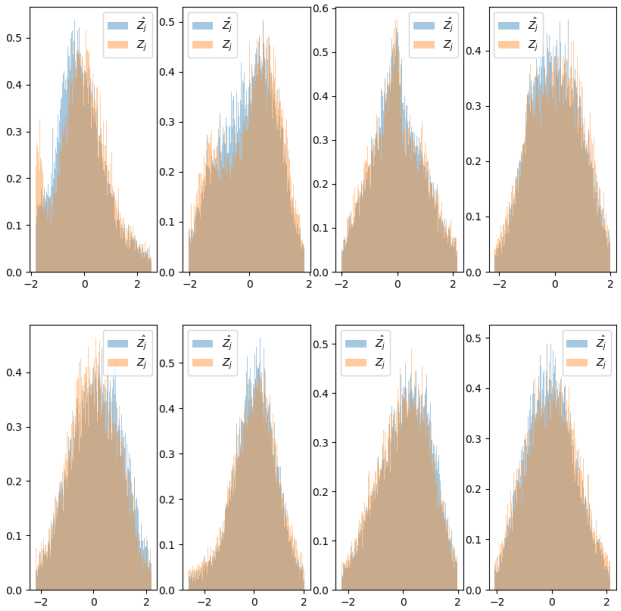

Figure 10: Generated samples for the MNIST dataset in the spectral space, first 8 coordinates

# 4 Discussion

We've showed that our method is suitable for learning functional data distributions, better than gaussian processes. For datasets where we can compute an empirical covariance, the Karhunen-Loeve theorem makes our method very efficient. Otherwise, there is some challenge in the choice of the kernel to use; a wrong kernel can lead to poor results. Future work around this method could adapt it and look at its performance for conditional distributions.

# References


*Anderson Brian DO.* Reverse-time diffusion equation models // Stochastic Processes and their Applications. 1982. 12, 3. 313–326.

*Baker Christopher T. H.* Numerical Integration in the Treatment of Integral Equations // Numerische Integration: Tagung Im Mathematischen Forschungsinstitut Oberwolfach Vom 1. Bis 7. Oktober 1978. Basel: Birkhäuser Basel, 1979. 44–53.

*Bao Fan, Li Chongxuan, Zhu Jun, Zhang Bo.* Analytic-DPM: an Analytic Estimate of the Optimal Reverse Variance in Diffusion Probabilistic Models // arXiv preprint arXiv:2201.06503. 2022.

Pattern Recognition and Machine Learning. // . 2006.

*Cattiaux Patrick, Conforti Giovanni, Gentil Ivan, Léonard Christian.* Time reversal of diffusion processes under a finite entropy condition // arXiv preprint arXiv:2104.07708. 2021.

*Dhariwal Prafulla, Nichol Alex.* Diffusion models beat GAN on Image Synthesis // arXiv preprint arXiv:2105.05233. 2021.

*Dutordoir Vincent, Saul Alan, Ghahramani Zoubin, Simpson Fergus.* Neural Diffusion Processes. June 2022.

*Eslami S. M. Ali, Jimenez Rezende Danilo, Besse Frederic, Viola Fabio, Morcos Ari S., Garnelo Marta, Ruderman Avraham, Rusu Andrei A., Danihelka Ivo, Gregor Karol, Reichert David P., Buesing Lars, Weber Theophane, Vinyals Oriol, Rosenbaum Dan, Rabinowitz Neil, King Helen, Hillier Chloe, Botvinick Matt, Wierstra Daan, Kavukcuoglu Koray, Hassabis Demis.* Neural Scene Representation and Rendering // Science. June 2018. 360, 6394. 1204–1210.

*Garnelo Marta, Rosenbaum Dan, Maddison Christopher, Ramalho Tiago, Saxton David, Shanahan Murray, Teh Yee Whye, Rezende Danilo, Eslami S. M. Ali.* Conditional Neural Processes // Proceedings of the 35th International Conference on Machine Learning. 80. July 2018a. 1704–1713. (Proceedings of Machine Learning Research).

*Garnelo Marta, Schwarz Jonathan, Rosenbaum Dan, Viola Fabio, Rezende Danilo J, Eslami SM, Teh Yee Whye.* Neural processes // arXiv preprint arXiv:1807.01622. 2018b.

Normalizing Field Flows: Solving Forward and Inverse Stochastic Differential Equations Using Physics-Informed Flow Models. // . Aug. 2021.

*Haussmann Ulrich G, Pardoux Etienne.* Time reversal of diffusions // The Annals of Probability. 1986. 14, 4. 1188–1205.

*Ho Jonathan, Jain Ajay, Abbeel Pieter.* Denoising diffusion probabilistic models // Advances in Neural Information Processing Systems. 2020.

*Huang Chin-Wei, Lim Jae Hyun, Courville Aaron.* A Variational Perspective on Diffusion-Based Generative Models and Score Matching // arXiv preprint arXiv:2106.02808. 2021.

*Jolicoeur-Martineau Alexia, Piché-Taillefer Rémi, Combes Rémi Tachet des, Mitliagkas Ioannis.* Adversarial score matching and improved sampling for image generation // International Conference on Learning Representations. 2021.

*Kim Hyunjik, Mnih Andriy, Schwarz Jonathan, Garnelo Marta, Eslami Ali, Rosenbaum Dan, Vinyals Oriol, Teh Yee Whye.* Attentive Neural Processes // International Conference on Learning Representations. 2019.

*Kushner Harold.* Stochastic Differential Equations (II Gihman and AV Skorohod). 1974.

$\mathcal{F}$-EBM: Energy Based Learning of Functional Data. // . Feb. 2022.

Gaussian Processes in Machine Learning. // . 2004.

*Song Yang, Durkan Conor, Murray Iain, Ermon Stefano.* On Maximum Likelihood Training of Score-Based Generative Models // Advances in Neural Information Processing Systems. 2021a.

*Song Yang, Ermon Stefano.* Improved techniques for training score-based generative models // Advances in Neural Information Processing Systems. 2020.

*Song Yang, Sohl-Dickstein Jascha, Kingma Diederik P., Kumar Abhishek, Ermon Stefano, Poole Ben.* Score-Based Generative Modeling through Stochastic Differential Equations // International Conference on Learning Representations. 2021b.

*Sullivan T. J.* Introduction to Uncertainty Quantification. 63. 2015. xii+342. (Texts in Applied Mathematics).

*Trippe Brian L., Yim Jason, Tischer Doug, Broderick Tamara, Baker David, Barzilay Regina, Jaakkola Tommi.* Diffusion Probabilistic Modeling of Protein Backbones in 3D for the Motif-Scaffolding Problem. June 2022.

*Watson Daniel, Ho Jonathan, Norouzi Mohammad, Chan William.* Learning to Efficiently Sample from Diffusion Probabilistic Models // arXiv preprint arXiv:2106.03802. 2021.

# A Background

## A.1 Euclidean Score-based Generative Modelling

We recall here briefly the key concepts behind SGMs on the Euclidean space $\mathbb{R}^d$ and refer the readers to Song et al. (2021b) for a more detailed introduction. We consider a forward *noising* process $(\mathbf{X}_t)_{t\geq 0}$ defined by the following Stochastic Differential Equation (SDE)

$$\mathrm{d}\mathbf{X}_t = -\mathbf{X}_t \mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t, \quad \mathbf{X}_0 \sim p_0, \tag{6}$$

where $(\mathbf{B}_t)_{t\geq 0}$ is a $d$-dimensional Brownian motion and $p_0$ is the data distribution. The available data gives us an empirical approximation of $p_0$. The process $(\mathbf{X}_t)_{t\geq 0}$ is simply an Ornstein–Ulhenbeck (OU) process which converges with geometric rate to $\mathrm{N}(0, \mathrm{Id})$. Under mild conditions on $p_0$, the time-reversed process $(\mathbf{Y}_t)_{t\geq 0} = (\mathbf{X}_{T-t})_{t\in[0,T]}$ also satisfies an SDE (Cattiaux et al., 2021; Haussmann, Pardoux, 1986) given by

$$\mathrm{d}\mathbf{Y}_t = \{\mathbf{Y}_t + 2\nabla \log p_{T-t}(\mathbf{Y}_t)\}\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t, \quad \mathbf{Y}_0 \sim p_T, \tag{7}$$

where $p_t$ denotes the density of $\mathbf{X}_t$. By construction, the law of $\mathbf{Y}_{T-t}$ is equal to the law of $\mathbf{X}_t$ for $t \in [0, T]$ and in particular $\mathbf{Y}_T \sim p_0$. Hence, if one could sample from $(\mathbf{Y}_t)_{t\in[0,T]}$ then its final distribution would be the data distribution $p_0$. Unfortunately we cannot sample exactly from (7) as $p_T$ and the scores $(\nabla \log p_t(x))_{t\in[0,T]}$ are intractable. Hence SGMs rely on a few approximations. First, $p_T$ is replaced by the reference distribution $\mathrm{N}(0, \mathrm{Id})$ as we know that $p_T$ converges geometrically towards it. Second, the following denoising score matching identity is exploited to estimate the scores

$$\nabla_{x_t} \log p_t(x_t) = \textstyle\int_{\mathbb{R}^d} \nabla_{x_t} \log p_{t|0}(x_t|x_0)\ p_{0|t}(x_0|x_t)\mathrm{d}x_0,$$

where $p_{t|0}(x_t|x_0)$ is the transition density of the OU process (6) which is available in closed-form. It follows directly that $\nabla \log p_t$ is the minimizer of $\ell_t(\mathbf{s}) = \mathbb{E}[\|\mathbf{s}(\mathbf{X}_t) - \nabla_{x_t} \log p_{t|0}(\mathbf{X}_t|\mathbf{X}_0)\|^2]$ over functions $\mathbf{s}$ where the expectation is over the joint distribution of $\mathbf{X}_0, \mathbf{X}_t$. This result can be leveraged by considering a neural network $\mathbf{s}_\theta : [0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ trained by minimizing the loss function $\ell(\theta) = \int_0^T \lambda_t \ell_t(\mathbf{s}_\theta(t, \cdot))\mathrm{d}t$ for some weighting function $\lambda_t > 0$. Finally, an Euler–Maruyama discretization of (7) is performed using a discretization step $\gamma$ such that $T = \gamma N$ for $N \in \mathbb{N}$

$$Y_{n+1} = Y_n + \gamma\{Y_n + 2\mathbf{s}_\theta(T - n\gamma, Y_n)\} + \sqrt{2\gamma} Z_{n+1}, \quad Y_0 \sim \mathrm{N}(0, \mathrm{Id}), \quad Z_n \overset{\text{i.i.d.}}{\sim} \mathrm{N}(0, \mathrm{Id}).$$

The above showcases the basics of SGMs but we highlight that many improvements have been proposed; see (e.g. Song, Ermon, 2020; Jolicoeur-Martineau et al., 2021; Dhariwal, Nichol, 2021). In particular, selecting an adaptive stepsize $(\gamma_n)_{n\in\mathbb{N}}$ (Bao et al., 2022; Watson et al., 2021) and using a predictor-corrector scheme (Song et al., 2021b) instead of a simple Euler–Maruyama discretization drastically improves performance. Finally, SGMs can also be derived through variational and maximum likelihood techniques (Ho et al., 2020; Huang et al., 2021; Song et al., 2021a).

## A.2 Density estimation

For all diffusion processes, there exists a corresponding deterministic process whose trajectories share the same marginal probability densities $\{p(X_t)\}_{t\in[0,T]}$.

This deterministic process satisfies an ODE Song et al. (2021b)

$$dX_t = f(X_t, t) - \frac{1}{2}\nabla \cdot [G(X_t, t)G(X_t, t)^\top] - \frac{1}{2}G(X_t, t)G(X_t, t)^\top \nabla \log p(X_t, t)dt$$

which can be solved with classical numerical solvers (e.g. Runge-Kutta).

# B Related work

**Neural process family** Neural Processes (Garnelo et al., 2018b) and Conditional Neural Processes (Eslami et al., 2018; Garnelo et al., 2018a) are closely related methods that create a representation of an stochastic process realization by aggregating representations of a context. NPs are generative models that uses an explicit likelihood, generally a fully factorized Gaussian, to estimate the posterior predictive distribution. Attentive (Conditional) Neural Processes (Kim et al., 2019, A(C)NP) introduced both self-attention and cross-attention into the NP family.

**Karhunen-Loeve approaches** (Lim et al., 2022) (Guo et al., 2021)