SCORE-BASED GENERATIVE MODELS ET LIENS AVEC LES PONTS DE SCHRÖDINGER

JULIEN DROUHET

Table des matières

Résumé

Le problème de génération de données synthétiques a trouvé, lors de la dernière décennie, de nombreuses conséquences pratiques et est devenu au fil du temps un domaine de recherche d'intérêt du fait de ses applications industrielles. En particulier, l'avènement du Deep Learning a permis d'effectuer de grandes avancées dans la création fidèle de données de haute définition. Le début de cette décennie coïncide avec l'arrivée d'un nouveau type de modèle génératif, largement plus performant que ses prédécesseurs (voir Figure 1) : les Score-Based Generative Models (SGM).
La première partie de ce texte est constituée d'une brève description des SGMs, tandis que les deuxième et troisième sections sont dédiées à la présentation de deux exemples de travaux de recherche récents sur le sujet : le premier, plus numérique, explore l'utilisation de nouvelles dynamiques dans le processus de bruitage ; le second, plus théorique, propose un lien entre les problématiques de *Score Matching* et les ponts de Schrödinger.

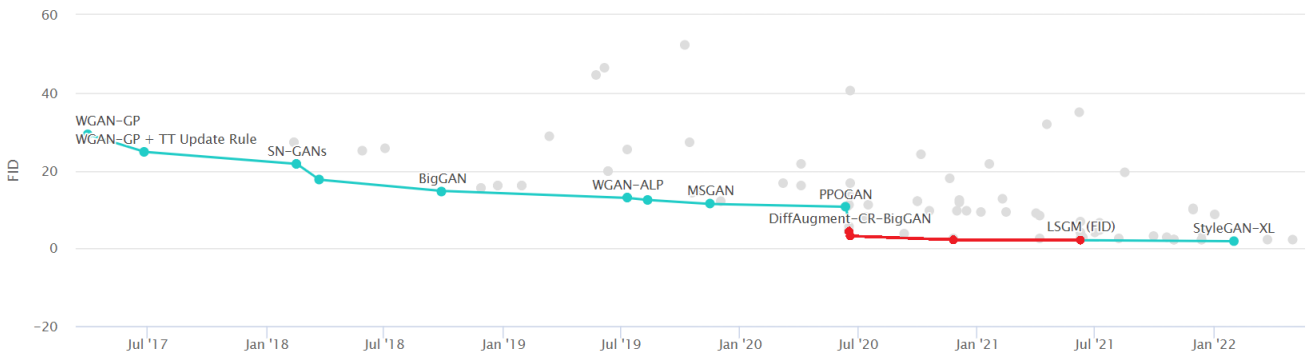

Figure 1. Score FID des meilleurs réseaux génératifs au cours du temps. On peut observer une nette amélioration des performances à partir de l'étude des SGMs. Les modèles génératifs représentés en rouge correspondent à des SGMs.

1. Score-Based Generative Models (SGM)

1.1. **Génération de données synthétiques et idée intuitive des SGMs.** Considérons un ensemble de données représenté par une certaine distribution de probabilité p_{data} sur $\mathbb{R}^D$, $D > 1$. L'objectif de la génération de données synthétiques est, à partir d'un nombre fini d'exemples issus de la distribution p_{data}, de définir un algorithme

permettant de transformer un bruit blanc en la distribution p_{data} [1].
Autrement dit, si l'on se donne une distribution facilement reproductible p_{noise} comme par exemple une loi gaussienne et que l'on suppose que cet algorithme peut être décrit par une fonction G, alors on cherche G de sorte que :

$$G_\sharp p_{\text{noise}} \simeq p_{\text{data}},$$

où $G_\sharp$ est ici la mesure image par G. Soulignons le fait que, dans notre cas, on ne pourra pas trouver de fonction G qui convient et que cette expression a pour unique but d'illustration.

L'idée des SGMs, proposée initialement en 2017 dans [1, 6], découle de la remarque suivante : il est très simple de bruiter une donnée de sorte qu'elle ressemble à un bruit blanc. Le principe des SGMs consiste à étudier ce processus d'ajout de bruit afin d'en déduire le processus inverse : débruiter un bruit blanc afin d'obtenir une donnée issue de p_{data}.
Bien qu'il existe différente façons de débruiter un tel signal, nous concentrons ce texte sur les travaux actuels fondés sur les Équations Différentielles Stochastiques (EDS), débutés par Song et al. ([13]).

1.2. **Formalisme.** Fixons un horizon de temps $T > 0$. L'idée introduite par Song et al. ([12]) est de bruiter les images de notre ensemble de données par une diffusion [2], régie par l'EDS suivante :

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\mathrm{d}t + \mathbf{G}(\mathbf{x}_t, t)\mathrm{d}\mathbf{w}_t, \quad 0 \le t \le T, \tag{1}$$

où $\mathbf{w}$ est un mouvement brownien standard dans $\mathbb{R}^D$, et où $f : \mathbb{R}^D \times [0,T] \to \mathbb{R}^D$ et $G : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ sont respectivement les coefficients de drift et de diffusion. Ici, la condition initiale est : $\mathbf{x}_0 \sim p_{\text{data}}$.
Supposons que l'on ait choisi cette diffusion de telle sorte qu'elle converge vers une distribution d'équilibre facilement échantillonnable et notée p_{noise}. Alors on devrait s'attendre à ce que la loi de $\mathbf{x}_T$ soit proche de cette distribution d'équilibre et que, en inversant la flèche du temps, l'on puisse partir de p_{noise} pour arriver vers une distribution proche de p_{data}. Cette approche est motivée par le résultat suivant, du à Song et al. (voir [12]) et inspiré par des travaux plus anciens comme ceux de Haussmann et Pardoux ([7]).

Théorème 1.1. Notons $\bar{\mathbf{x}}_t := \mathbf{x}_{T-t}$, $0 \le t \le T$. Alors $\bar{\mathbf{x}}$ vérifie l'EDS suivante :

$$\mathrm{d}\bar{\mathbf{x}}_t = \left\{-\mathbf{f}(\bar{\mathbf{x}}_t, T-t) - \mathbf{G}(\bar{\mathbf{x}}_t, T-t)\mathbf{G}(\bar{\mathbf{x}}_t, T-t)^{\mathrm{T}}\nabla_{\bar{\mathbf{x}}} \log p_{T-t}(\bar{\mathbf{x}}_t)\right\}\mathrm{d}t + \mathbf{G}(\bar{\mathbf{x}}_t, t)\mathrm{d}\mathbf{w}_t, \quad 0 \le t \le T, \tag{2}$$

où p_{T-t} est la distribution marginale de $\mathbf{x}$ au temps $T-t$.

L'idée des SGMs consiste donc à démarrer un processus $(\bar{\mathbf{x}}_t)_{0 \le t \le T}$ avec la distribution d'équilibre p_{noise} et suivant l'EDS (2) afin d'obtenir une approximation de p_{data} via $\bar{\mathbf{x}}_T$. Le processus de génération est illustré dans la Figure 2. On définit $\mathbf{s}(\mathbf{x}, t) := \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ la *fonction de score*, qui correspond en pratique au gradient de la log-vraisemblance.

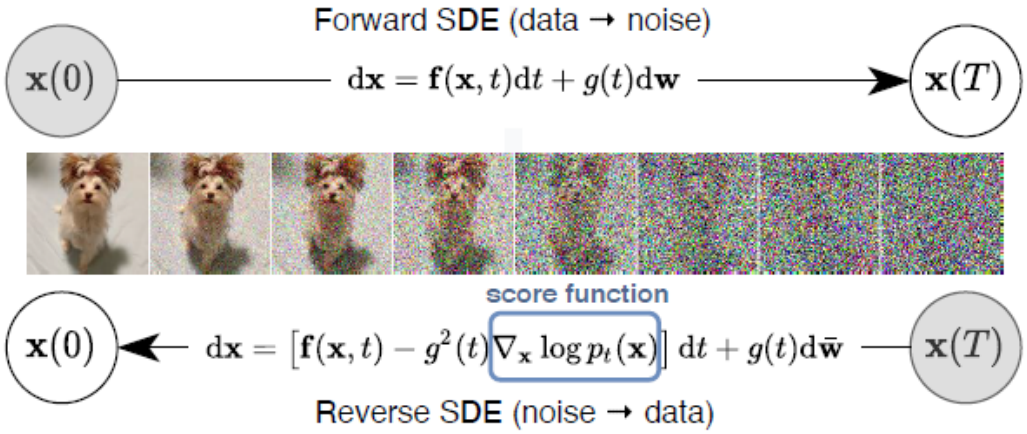

FIGURE 2. Illustration de la génération d'images via le processus en temps inversé. Figure issue de Song et al. [12].

1. En pratique, on peut considérer le cas classique de la génération d'images et du dataset CIFAR-10, constitué de 60,000 images en couleurs constituées de 32 pixels de côté. Ici, la dimension est donc $D = 3 \times 32 \times 32 = 3072$.

2. Sauf mention explicite du contraire, dans la suite, on notera respectivement p_t, $p_{s,t}$ et $p_{s|t}$ les lois marginales, conjointes et conditionnelles du processus $\mathbf{x}$ au temps $s, t \in [0,T]$.

Cependant, la génération de trajectoires suivant l'EDS (2) soulève un problème majeur : nous n'avons pas accès à la fonction de score $\mathbf{s}$.
En pratique, la fonction de score est approchée par un réseau de neurones $\mathbf{s}_\theta$, paramétré par $\theta \in \Theta \subset \mathbb{R}^W$, $W > 1$. Pour coïncider avec les méthodes de Deep Learning, il faut donc trouver une fonction de coût dont $\mathbf{s}$ est un minimiseur.
Notons $p_{s|t}$ la loi de $\mathbf{x}_s$ conditionnellement à $\mathbf{x}_t$ et $p_{s,t}$ la loi conjointe, pour $s, t \in [0, T]$. Dans la suite, on assimile une distribution avec sa densité par rapport à la mesure de Lebesgue [3]. On a, pour tout $t \in [0, T]$:

$$\begin{aligned}\mathbf{s}(\mathbf{x}_t, t) &= \frac{\nabla_{\mathbf{x}} p_t(\mathbf{x}_t)}{p_t(\mathbf{x}_t)} \\ &= \int_{\mathbb{R}^D} \frac{\nabla_{\mathbf{x}} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) p_{0,t}(\mathbf{x}_0, \mathbf{x}_t)}{p_t(\mathbf{x}_t)} \mathrm{d}\mathbf{x}_0 \\ &= \int_{\mathbb{R}^D} \nabla_{\mathbf{x}} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) \mathrm{d}\mathbf{x}_0 \\ &= \mathbb{E}\left[\nabla_{\mathbf{x}} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) \middle| \mathbf{x}_t\right].\end{aligned}$$

En utilisant l'expression variationnelle de l'espérance conditionnelle, on en déduit que :

$$\mathbf{s}(\mathbf{x}_t, t) = \underset{f \in L^2(p_t)}{\operatorname{argmin}} \mathbb{E}_{\mathbf{x}_t \sim p_t(\mathbf{x}_t)} \left\| f(\mathbf{x}_t) - \nabla_{\mathbf{x}} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) \right\|_2^2.$$

Puisque, en pratique, le réseau de neurones $\mathbf{s}_\theta$ approche la fonction $\mathbf{s}$ en les coordonnées jointes $(\mathbf{x}, t)$, il faut développer une fonction de coût uniforme en la variable de temps. On choisit donc de prendre une somme pondérée des fonctions de coûts trouvées ci-dessus. On en déduit le Denoising Score Matching (DSM) :

$$\mathcal{L}_{\mathrm{DSM}}(\theta) := \int_0^{\mathrm{T}} \lambda(t) \mathbb{E}_{\mathbf{x}_t \sim p_t(\mathbf{x}_t)} \left\| \mathbf{s}_\theta(\mathbf{x}_t) - \nabla_{\mathbf{x}} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) \right\|_2^2 \mathrm{d}t, \quad \theta \in \Theta,$$

où $\lambda : [0, T] \to \mathbb{R}_+$. C'est cette fonction de coût que l'on cherchera donc à minimiser pour obtenir la meilleure approximation $\mathbf{s}_\theta$ de $\mathbf{s}$.

Remarque 1.1. Numériquement, on simule le processus $\mathbf{x}_t$ (ainsi que le processus à temps inversé associé) en utilisant une dynamique de Langevin ou un schéma d'Euler. De ce fait, on discrétise également la variable de temps pour $\mathbf{s}$, qui devient donc une fonction de $\{0, \cdots, N\} \times \mathbb{R}^D$ dans $\mathbb{R}^D$, et l'intégrale dans la définition du DSM est remplacée par une intégrale.

Bilan. Le processus de génération de données synthétiques par SGM peut donc se résumer ainsi :

(1) On détermine une approximation $\mathbf{s}_\theta$ du score via un réseau de neurones minimisant le DSM. La simulation des trajectoires se fait par schéma d'Euler ou par dynamiques de Langevin.
(2) On utilise cette approximation du score pour la génération de données synthétiques, en utilisant l'EDS :

$$\mathrm{d}\bar{\mathbf{x}}_t = \left\{-\mathbf{f}(\bar{\mathbf{x}}_t, T-t) - \mathbf{G}(\bar{\mathbf{x}}_t, T-t)\mathbf{G}(\bar{\mathbf{x}}_t, T-t)^{\mathrm{T}} \mathbf{s}_\theta(\bar{\mathbf{x}}_t, T-t)\right\} \mathrm{d}t + \mathbf{G}(\bar{\mathbf{x}}_t, t)\mathrm{d}\mathbf{w}_t, \quad 0 \le t \le T.$$

1.3. **Cas Ornstein-Uhlenbeck.** Une diffusion naturelle à utiliser dans le processus de bruitage des données est celle d'Ornstein-Uhlenbeck, dont on rappelle l'EDS :

$$\mathrm{d}\mathbf{x}_t = -\alpha \mathbf{x}_t \mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{w}_t, \quad 0 \le t \le T, \tag{3}$$

où $\alpha \ge 0$.

Remarque 1.2. Les processus d'Ornstein-Uhlenbeck correspondent en réalité à $\alpha > 0$ et ont été étudié dans [8]. Nous incluons également le cas $\alpha = 0$ car ce dernier est également étudié dans la littérature (e.g. [11]). En pratique, les deux cas sont traités identiquement, ce qui justifie également le fait de prendre $\alpha \ge 0$.

On rappelle qu'un tel processus admet une expression explicite (conditionnellement à sa valeur initiale) :

$$\mathbf{x}_t = \begin{cases} \mathbf{x}_0 + \sqrt{2}\mathbf{w}_t & \text{si } \alpha = 0, \\ \mathbf{x}_0 e^{-\alpha t} + \frac{e^{-\alpha t}}{\sqrt{\alpha}} \mathbf{w}_{e^{2\alpha t} - 1} & \text{sinon.} \end{cases}$$

Cela nous permet de déduire une loi limite pour ces processus (correspondant respectivement aux cas traités dans [11] et [8]) : $p_{\mathrm{noise}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{0}, 2T\mathbf{I})$ si $\alpha = 0$ et $p_{\mathrm{noise}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{0}, \frac{1}{\alpha}\mathbf{I})$ sinon.
Par ailleurs, le Théorème 1.1 assure que le processus en temps inversé est régi par l'EDS :

$$\mathrm{d}\bar{\mathbf{x}}_t = \{\alpha \bar{\mathbf{x}}_t + \nabla_{\mathbf{x}} \log p_{T-t}(\bar{\mathbf{x}}_t)\} \mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{w}_t, \quad 0 \le t \le T.$$

3. En pratique, les distributions que l'on considérera auront une densité par rapport à la mesure de Lebesgue.

Remarque 1.3. Il est en réalité possible de trouver la dynamique à simuler sans pour autant utiliser le calcul stochastique, mais seulement à partir d'approximations formelles du schéma d'Euler.
On se place, pour simplifier, dans le cas où $\alpha = 1$. Le schéma d'Euler, avec un pas de discrétisation γ, de l'EDS (3) correspond à la suite de variables aléatoires définies par :

$$\begin{cases} X_0 \sim p_{\text{data}}, \\ X_{k+1} = (1-\gamma)X_k + \sqrt{2\gamma}Z_{k+1}, \quad 0 \leq k \leq N-1, \end{cases}$$

avec $Z_1, \cdots, Z_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Dans la suite, on fera les calculs à une constante multiplicative $C > 0$ près.
Soit $k \in \{0, \cdots, N-1\}$. Par définition de la suite $(X_k)_{0 \leq k \leq N}$, on a : $X_{k+1}|X_k \sim \mathcal{N}((1-\gamma)X_k, 2\gamma\mathbf{I})$. Par la formule de Bayes, on trouve que la densité de la loi de $X_k|X_{k+1}$ peut écrire :

$$\begin{aligned} p_{k|k+1}(x_k|x_{k+1}) &= \frac{p_{k+1|k}(x_{k+1}|x_k)p_k(x_k)}{p_{k+1}(x_{k+1})} \\ &= C \exp\left(-\frac{1}{4\gamma}\|x_{k+1} - (1-\gamma)x_k\|_2^2 + \log p_k(x_k) - \log p_{k+1}(x_{k+1})\right). \end{aligned}$$

En supposant $p_{k+1} \simeq p_k$ et $x_{k+1} \simeq x_k$, un calcul à l'ordre 1 en γ dans l'exponentielle permet de trouver l'approximation suivante :

$$p_{k|k+1}(x_k, x_{k+1}) \simeq C \exp\left(-\frac{1}{4\gamma}\|x_k - x_{k+1} - \{x_{k+1} + 2\gamma\nabla \log p_k(x_{k+1})\}\|_2^2\right),$$

ce qui revient à dire que $X_k|X_{k+1} \sim \mathcal{N}(X_{k+1} + 2\gamma\nabla \log p_k(X_{k+1}), 2\gamma\mathbf{I})$. Autrement dit, pour tout $k \in \{0, \cdots, N-1\}$:

$$X_k = X_{k+1} + 2\gamma\nabla \log p_k(X_{k+1}) + \sqrt{2\gamma}\tilde{Z}_{k+1},$$

où $\tilde{Z}_{k+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Il s'agit de la discrétisation d'Euler de l'EDS en temps inversé trouvée à partir du Théorème 1.1.
Bien que ce raisonnement soit purement formel et que l'utilisation du calcul stochastique semble plus convaincante, ce raisonnement permet de trouver de manière simple et rapide le résultat souhaité. En particulier, il s'exporte plus facilement pour former des nouvelles méthodes d'estimations sous forme de *Score Matching*[4] dans d'autres domaines. On pourra garder cette idée à l'esprit dans la Section 3, où un calcul quasi-identique permet de définir les ponts de Schrödinger diffusifs.

L'utilisation de ces dynamiques ont montré de très bons résultats, à la fois d'un point de vue de la diversité que de la qualité des données reproduites, sur des ensembles de données classiques comme CIFAR-10. En particulier, l'arrivée des SGMs dans les années 2020 a supplanté les anciens réseaux génératifs en termes de performances(voir Figure 1 pour l'exemple de CIFAR-10). Cependant, l'entraînement d'un SGM est généralement très long et nécessite quelques modifications pratiques dans l'implémentation numérique, du fait de la convergence relativement lente des processus associés. En pratique, l'amélioration de la qualité d'un SGM peut s'effectuer par l'augmentation de la valeur de l'horizon de temps T (ou, de manière équivalente, de la réduction du pas de temps dans les schémas de discrétisation), ce qui explique la relative longueur des entraînements.

1.4. **Borne théorique de convergence.** Terminons cette section par le résultat principal de convergence existant concernant les SGMs, du à De Bortoli et al. ([2]).
On considère donc une approximation du score $\mathbf{s}_\theta$ ainsi que la discrétisation d'Euler du processus d'Ornstein-Uhlenbeck en temps inversé étudié en Sous-Section 1.3 :

$$X_k = X_{k+1} + \gamma_{k+1}\alpha X_{k+1} + 2\gamma_{k+1}\mathbf{s}_\theta(X_{k+1}, k+1) + \sqrt{2\gamma_{k+1}}Z_{k+1}, \quad 0 \leq k \leq N-1, \tag{4}$$

où $\gamma_1, \cdots, \gamma_N$ sont les pas de temps, $\gamma_1 + \cdots + \gamma_N = T$ et $Z_1, \cdots, Z_N \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I})$. Rappelons ici que la loi de X_N est est égale à p_{noise} telle que définie en Sous-Section 1.3 et soulignons le fait que, du fait de la discrétisation, $\mathbf{s}_\theta$ est une fonction de $\{1, \cdots, N\} \times \mathbb{R}^D$ dans $\mathbb{R}^D$.

Théorème 1.2 ([3], Théorème 1)**.** Supposons $\mathbf{s}_\theta$ continue et qu'il existe $M > 0$ tel que :

$$\|\mathbf{s}_\theta - \nabla_{\mathbf{x}} p_\cdot\|_\infty \leq M.$$

Supposons en outre que $p_{\text{data}} \in \mathcal{C}^3\left(\mathbb{R}^D, (0, +\infty)\right)$ est bornée et qu'il existe des constantes $d_1, A_1, A_2, A_3 \geq 0$, $\beta_1, \beta_2, \beta_3 \in \mathbb{N}$ et $m_1 > 0$ telles que, pour tous $\mathbf{x} \in \mathbb{R}^D$ et $i = 1, 2, 3$, on ait :

$$\left\|\nabla_{\mathbf{x}}^i \log p_{\text{data}}(\mathbf{x})\right\| \leq A_i\left(1 + \|\mathbf{x}\|^{\beta_i}\right), \quad \langle \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}), \mathbf{x}\rangle \leq -m_1\|\mathbf{x}\|^2 + d_1\|\mathbf{x}\|,$$

4. On parle de *Score Matching* lorsque l'on cherche à approcher les fonctions de score $\nabla \log p$ pour certaines densités, à la manière du Denoising Score Matching.

avec $\beta_1 = 1$. Alors, pour tout $\alpha \geq 0$, il existe des constantes $B_\alpha, C_\alpha, D_\alpha \geq 0$ telles que, pour tout $N \in \mathbb{N}$ et toute suite $(\gamma_k)_{1\leq k\leq N} \in \left(\mathbb{R}_+^*\right)^N$ telle que $\gamma_1 + \cdots + \gamma_N = T$, nous avons les bornes suivantes en variation totale :

(1) Si $\alpha = 0$, alors $\|\mathcal{L}(X_0) - p_{\text{data}}\|_{TV} \leq C_0(M + \bar{\gamma}^{1/2})e^{D_0 T} + B_0(T^{-1} + T^{-1/2})$.

(2) Si $\alpha > 0$, alors $\|\mathcal{L}(X_0) - p_{\text{data}}\|_{TV} \leq C_\alpha(M + \bar{\gamma}^{1/2})e^{D_\alpha T} + B_\alpha e^{-\sqrt{\alpha}T}$.

Ici, on a noté $\bar{\gamma} = \sup_{k=1,\cdots,N} \gamma_k$ et $\mathcal{L}(X_0)$ la loi de X_0 définit par l'équation (4).

2. Critically-Damped Langevin Dynamics (CLD)

L'objectif de ce nouveau modèle de SGM, introduit dans [5], est de palier la lenteur de l'entraînement des SGMs en renforçant la convergence des modèles fondés sur des processus d'Ornstein-Uhlenbeck.
L'idée des Critically-Damped Langevin Dynamics (CLD dans la suite) se fonde sur une interprétation physique des dynamiques en jeu. En effet, un parallèle peut être dressé entre la force de rappel mise en jeu dans les processus d'Ornstein-Uhlenbeck (paramétrée par la constance α) et celle du modèle de l'oscillateur harmonique en physique. Dockhorn et al. ([5]) proposent donc d'utiliser le modèle de l'oscillateur amorti comme base pour leur modèle de SGM. Par ailleurs, ils proposent de se placer dans un régime critiquement-amorti ("Critically-Damped"), où la convergence est plus rapide.

2.1. Description du modèle.

Afin de mieux coller au modèle physique de l'oscillateur amorti, Dockhorn et al. ([5]) proposent de rajouter un *paramètre de vitesse* dans la diffusion, et d'écrire des dynamiques semblables à celles du mouvement. On considère donc le processus $\mathbf{u}_t := (\mathbf{x}_t, \mathbf{v}_t)^{\mathrm{T}} \in \mathbb{R}^{2D}$, vérifiant l'EDS :

$$\begin{pmatrix} \mathrm{d}\mathbf{x}_t \\ \mathrm{d}\mathbf{v}_t \end{pmatrix} = \begin{pmatrix} M^{-1}\mathbf{v}_t \\ -\mathbf{x}_t \end{pmatrix} \beta \mathrm{d}t + \begin{pmatrix} \mathbf{0}_d \\ -\Gamma M^{-1}\mathbf{v}_t \end{pmatrix} \beta \mathrm{d}t + \begin{pmatrix} \mathbf{0}_d \\ \sqrt{2\Gamma\beta}\,\mathrm{d}\mathbf{w}_t \end{pmatrix}, \tag{5}$$

où $\beta, M, \Gamma > 0$ sont respectivement des paramètres de facteur d'échelle temporelle, de masse et de friction. Le premier terme du membre de droite dans l'équation (5) est appelé *composante hamiltonienne*, tandis que les deux derniers termes constituent une dynamique d'Ornstein-Uhlenbeck sur $\mathbf{v}$: on parlera donc de *composante Ornstein-Uhlenbeck*.

Remarque 2.1. Comme pour le modèle de l'oscillateur amorti en physique, les dynamiques de l'EDS (5) admettent deux régimes différents : les régimes *sous-amorti* et *sur-amorti*. Le régime sous-amorti correspond au cas où la friction est trop faible et où la composante hamiltonienne prédomine. Plus quantitativement, il s'agit du cas où $\Gamma^2 < 2M$. Dans ce régime, la convergence est d'autant plus forte que la friction est forte. Au contraire, lorsque $\Gamma^2 > 2M$ (c'est-à-dire le régime sur-amorti), le système admet un comportement pseudo-oscillatoire, ce qui ralentit considérablement la convergence du processus vers un régime d'équilibre. Au regard de cette discussion, les auteurs de [5] proposent de se placer dans le régime sous-amorti tout en prenant la plus grande valeur possible pour le coefficient de friction, c'est-à-dire dans le cas critiquement-amorti : $\Gamma^2 = 2M$. Dans la suite, nous supposons cette égalité vraie.
La Figure 3 montre les résultats d'un entraînement de CLD dans les régimes sous-amorti, critiquement-amorti et sur-amorti. On y observe que ce régime semble en effet le plus efficace.

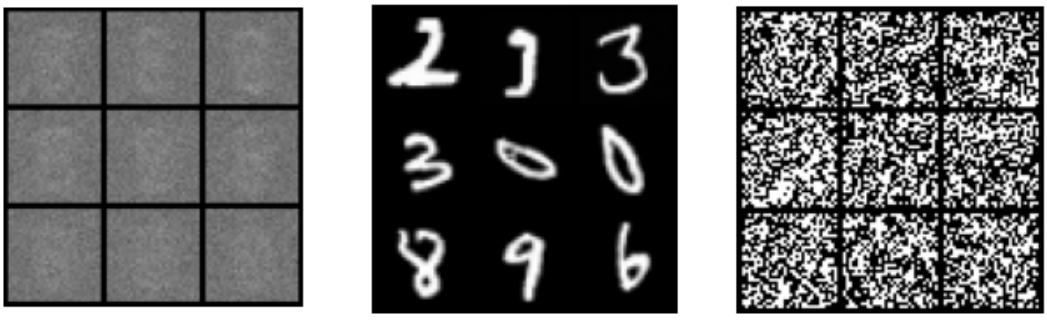

Figure 3. Résultat d'un entraînement d'un CLD après 95,000 itérations en régime sous-amorti **(Gauche)**, critiquement-amorti **(Milieu)** et sur-amorti **(Droite)**. L'ensemble de données est MNIST, constitué de 70,000 images de nombres écrits à la main, de dimension $28 \times 28 = 784$ pixels. On observe que la convergence est très lente dans le cas sous-amorti ($\Gamma^2 = 0.5M$ ici) et que la dynamique diverge dans le régime sur-amorti ($\Gamma^2 = 4M$ ici). Seul le cas critiquement-amorti semble converger en temps raisonnable. Le résultat a été obtenu via l'adaptation du code de [5], en collaboration avec Émile Pierret.

Du Théorème 1.1, on tire la dynamique du processus en temps inversé $\bar{\mathbf{u}} := (\bar{\mathbf{x}}, \bar{\mathbf{v}})^{\mathrm{T}}$:

$$\begin{pmatrix} \mathrm{d}\bar{\mathbf{x}}_t \\ \mathrm{d}\bar{\mathbf{v}}_t \end{pmatrix} = \underbrace{\begin{pmatrix} -M^{-1}\bar{\mathbf{v}}_t \\ \bar{\mathbf{x}}_t \end{pmatrix}\beta \mathrm{d}t}_{A_H} + \underbrace{\begin{pmatrix} \mathbf{0}_d \\ -\Gamma M^{-1}\bar{\mathbf{v}}_t \end{pmatrix}\beta \mathrm{d}t + \begin{pmatrix} \mathbf{0}_d \\ \sqrt{2\Gamma\beta}\,\mathrm{d}\mathbf{w}_t \end{pmatrix}}_{A_O} + \underbrace{\begin{pmatrix} \mathbf{0}_d \\ 2\Gamma\left\{\mathbf{s}(\bar{\mathbf{u}}, T-t) + M^{-1}\bar{\mathbf{v}}_t\right\} \end{pmatrix}\beta \mathrm{d}t}_{S}, \tag{6}$$

où $\mathbf{s}(\cdot, t)$ désigne de nouveau la fonction de score $\nabla_{\mathbf{u}} \log p_t$. Notons qu'ici, on peut distinguer trois composantes distinctes. On retrouve les composantes hamiltonienne (A_H) et Ornstein-Uhlenbeck (A_O), mais à ces deux dernières s'ajoute une troisième faisant intervenir la fonction de score, qui est notée S. Soulignons le fait que, sans la composante S, l'EDS serait facilement manipulable, de telle sorte que la majorité de la difficulté provient de l'apparition du score dans l'EDS en temps inversé. Les auteurs de [5] se servent de cette remarque pour développer un nouveau processus d'échantillonnage, plus performant que le schéma d'Euler. Nous reviendrons sur ce point dans la sous-section suivante.

Insistons sur le fait que le paramètre de vitesse est ici *arbitraire*. En particulier, le choix de la loi de $\mathbf{u}_0$ n'est plus simplement fixé par les données comme pour le cas de la première section. Dockhorn et al. ([5]) proposent l'initialisation suivante : $\mathbf{u}_0 \sim p_{\mathrm{data}} \otimes \mathcal{N}(\mathbf{0}, \gamma M \mathbf{I})$, avec $\gamma \in (0,1)$ un hyperparamètre. Pour ce qui est de l'initialisation du processus en temps inversé, il s'agit de trouver une distribution vers laquelle $\mathbf{u}$ converge. C'est l'objet de la Proposition suivante.

Proposition 2.1. Le processus défini par l'EDS (5) converge en loi, lorsque $t \to +\infty$, vers :

$$p_{\mathrm{EQ}}(\mathbf{u}) := \mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{I})\,\mathcal{N}(\mathbf{v}; \mathbf{0}, M\mathbf{I}).$$

2.2. **Processus d'échantillonage.** Les auteurs de [5] proposent dans leur article une manière plus adaptée pour simuler les trajectoires, tirant profit de l'EDS considérée : le Symmetric Splitting CLD Sampler (SSCS).

Commençons par rappeler que l'équation de Fokker-Planck associée à l'EDS (6) est :

$$\partial_t p_{T-t}(\bar{\mathbf{u}}_t) = (\mathcal{L}_A^* + \mathcal{L}_S^*)\, p_{T-t}(\bar{\mathbf{u}}_t),$$

où $\mathcal{L}_A^*$ et $\mathcal{L}_S^*$ sont les opérateurs de Fokker-Planck associés à $A := A_H + A_O$ et S, respectivement. Leurs expressions explicites sont :

$$\begin{aligned} \mathcal{L}_A^*\phi(\bar{\mathbf{u}}) &:= \beta M^{-1}\bar{\mathbf{v}} \cdot \nabla_{\bar{\mathbf{x}}}\phi(\bar{\mathbf{u}}) - \beta\bar{\mathbf{x}} \cdot \nabla_{\bar{\mathbf{v}}}\phi(\bar{\mathbf{u}}) + \Gamma\beta M^{-1}\nabla_{\bar{\mathbf{v}}} \cdot (\bar{\mathbf{u}}\phi(\bar{\mathbf{u}})) + \Gamma\beta\Delta_{\bar{\mathbf{v}}}\phi(\bar{\mathbf{u}}), \\ \mathcal{L}_S^*\phi(\bar{\mathbf{u}}) &:= -2\Gamma\beta\nabla_{\bar{\mathbf{v}}} \cdot \left[\left(\mathbf{s}(\bar{\mathbf{u}}, T-t) + M^{-1}\bar{\mathbf{v}}\right)\phi(\bar{\mathbf{u}})\right]. \end{aligned}$$

Ainsi, on peut donner une formule pour la loi $\bar{\mathbf{u}}_t$:

$$\bar{\mathbf{u}}_t = e^{t(\mathcal{L}_A^* + \mathcal{L}_S^*)}\bar{\mathbf{u}}_0,$$

où on rappelle que $\bar{\mathbf{u}}_0 \sim p_{\mathrm{EQ}}$.
Bien que cette expression soit purement théorique et ne donne donc pas en pratique accès à la loi de $\bar{\mathbf{u}}$, c'est un des ingrédients clés dans la définition du SSCS.

Le second ingrédient consiste en le Proposition ci-dessous.

Proposition 2.2. Soit $\delta t > 0$. On a :

$$e^{\frac{\delta t}{2}\mathcal{L}_A^*}\bar{\mathbf{u}}_t \sim \mathcal{N}\left(\bar{\boldsymbol{\mu}}_{\delta t/2}(\bar{\mathbf{u}}_t), \bar{\boldsymbol{\Sigma}}_{\delta t/2}\right),$$

où :

$$\begin{aligned} \bar{\boldsymbol{\mu}}_{\delta t/2}(\bar{\mathbf{u}}_t) &:= \begin{pmatrix} 2\beta\Gamma^{-1}\frac{\delta t}{2}\bar{\mathbf{x}}_t - 4\beta\Gamma^{-2}\bar{\mathbf{v}}_t + \bar{\mathbf{x}}_t \\ \beta\frac{\delta t}{2}\bar{\mathbf{x}}_t - 2\beta\Gamma^{-1}\bar{\mathbf{v}}_t + \bar{\mathbf{v}}_t \end{pmatrix} e^{-2\beta\Gamma^{-1}\frac{\delta t}{2}}, \\ \bar{\boldsymbol{\Sigma}}_{\delta t/2} &:= \Sigma_{\delta t/2} \otimes \mathbf{I}_d, \end{aligned}$$

et :

$$\Sigma_{\delta t/2} := \begin{pmatrix} e^{4\beta\Gamma^{-1}\frac{\delta t}{2}} - 1 - 4\beta\Gamma^{-1}\frac{\delta}{2} - 8\left(\beta\Gamma^{-1}\frac{\delta t}{2}\right)^2 & -4\Gamma^{-1}\left(\beta\frac{\delta t}{2}\right)^2 \\ -4\Gamma^{-1}\left(\beta\frac{\delta t}{2}\right)^2 & \left(\frac{\Gamma}{2}\right)^2\left(e^{4\beta\Gamma^{-1}\frac{\delta t}{2}} - 1\right) + \beta\Gamma\frac{\delta t}{2} - 2\left(\beta\frac{\delta t}{2}\right)^2 \end{pmatrix} e^{-4\beta\Gamma^{-1}\frac{\delta t}{2}}.$$

Autrement dit, si la composante S était en réalité nulle (ou négligeable), on pourrait obtenir un expression explicite pour $\bar{\mathbf{u}}$.
Dockhorn et al. ([5]) proposent donc, plutôt que d'appliquer directement l'opérateur $e^{t(\mathcal{L}_A^* + \mathcal{L}_S^*)}$, d'appliquer séparément les opérateurs $e^{t\mathcal{L}_A^*}$ et $e^{t\mathcal{L}_S^*}$. Cette procédure est justifiée par la Proposition qui suit, et qui découle de la formule de Baker–Campbell–Hausdorff.

Proposition 2.3. Soit $t \in [0,T]$ et $N \in \mathbb{N}^*$. Alors il existe une constante C, dépendant seulement de T, M et β, telle que :

$$\left\| e^{t(\mathcal{L}_A^* + \mathcal{L}_S^*)} - \left[e^{\frac{\delta t}{2}\mathcal{L}_A^*} e^{\delta t \mathcal{L}_S^*} e^{\frac{\delta t}{2}\mathcal{L}_A^*} \right]^N \right\| \leq C\delta t^2, \tag{7}$$

où $\delta t = t/N$.

Ainsi, plutôt qu'un schéma d'Euler, Dockhorn et al. ([5]) proposent − au prix d'une erreur quadratique en le pas de temps − de déduire $\bar{\mathbf{u}}_{t+\delta t}$ à partir de $\bar{\mathbf{u}}_t$ via l'application successive des opérateurs $e^{t\mathcal{L}_A^*}$ et $e^{t\mathcal{L}_S^*}$.
En revanche, nous ne disposons pas de formule explicite pour l'action de $e^{t\mathcal{L}_S^*}$. Nous devons donc, en pratique, le remplacer par son schéma d'Euler correspondant :

$$e^{\delta t \mathcal{L}_S^{\text{Euler}}} \bar{\mathbf{u}}_t := \bar{\mathbf{u}}_t + \begin{pmatrix} \mathbf{0}_d \\ 2\beta\Gamma \left\{ \mathbf{s}\left(\bar{\mathbf{u}}_t, T-t\right) + M^{-1}\bar{\mathbf{v}}_t \right\} \end{pmatrix}.$$

En particulier, la borne sur l'erreur commise devient linéaire en le pas de temps, ce qui peut réduire considérablement l'efficacité de la méthode [5]. Cependant, lorsque le terme S est négligeable face aux autres termes, on peut s'attendre à une meilleure simulation qu'un schéma d'Euler. Dockhorn et al. ([5]) ont testé cette hypothèse dans un modèle-jouet, et ont conclu que ce schéma de discrétisation est en effet bien plus performant qu'un schéma d'Euler lorsque le pas de temps est grand.

2.3. **Hybrid Score Matching (HSM).** Finalement, remarquons que contrairement au SGM présenté dans la première section, on ne dispose pas de la loi marginale $p_t(\mathbf{x}_t)$, pourtant nécessaire au calcul du Denoising Score Matching. Bien que l'on pourrait simuler des trajectoires pour l'approcher, cela serait assez coûteux en temps puisque l'initialisation de la dynamique est plus complexe que pour un SGM classique. Les auteurs proposent plutôt d'utiliser une nouvelle fonction de coût, appelée le *Hybrid Score Matching* (HSM). Le principe du HSM est, lors de l'échantillonnage, de démarrer les diffusions à partir de p_{data}, mais ensuite de marginaliser selon la vitesse initiale, où on rappelle que $p_0(\mathbf{v}_0) = \mathcal{N}(\mathbf{0}, \gamma M \mathbf{I})$. En pratique, l'expression du HSM est :

$$\mathcal{L}_{\text{HSM}}(\theta) := \int_0^T \mathbb{E}_{\mathbf{x}_0 \sim p_0(\mathbf{x}_0)} \mathbb{E}_{\mathbf{u}_t \sim p_t(\mathbf{u}_t|\mathbf{x}_0)} \left[\lambda(t) \| s_{\boldsymbol{\theta}}\left(\mathbf{u}_t, t\right) - \nabla_{\bar{\mathbf{v}}} \log p_t\left(\mathbf{u}_t|\mathbf{x}_0\right) \|_2^2 \right] \mathrm{d}t, \quad \theta \in \Theta,$$

où $\lambda : [0,T] \to \mathbb{R}_+$.

Remarque 2.2. En réalité, il est possible de montrer que, pour tout $\theta \in \Theta$:

$$\mathcal{L}_{\text{HSM}}(\theta) = \mathcal{L}_{\text{DSM}}(\theta) + \int_0^T \lambda(t) \left\{ \mathbb{E}_{\mathbf{x}_0 \sim p_0(\mathbf{x}_0),\, \mathbf{u}_t \sim p_t(\mathbf{u}_t|\mathbf{x}_0)} \|\nabla_{\mathbf{v}} \log p_t(\mathbf{u}_t|\mathbf{u}_0)\|_2^2 - \mathbb{E}_{\mathbf{u}_0 \sim p_0(\mathbf{x}_0),\, \mathbf{u}_t \sim p_t(\mathbf{u}_t|\mathbf{u}_0)} \|\nabla_{\mathbf{v}} \log p_t(\mathbf{u}_t|\mathbf{u}_0)\|_2^2 \right\} \mathrm{d}t,$$

où le second terme ne dépend pas de θ. Ainsi, la minimisation du DSM et celle du HSM sont deux problèmes équivalents.
Le HSM apparaît à cet égard comme une réduction de la variance du DSM, dans le sens où l'espérance selon $p_0(\mathbf{v}_0)$ est résolue analytiquement, alors qu'elle aurait du être approchée par échantillonage dans le cas du DSM [6]. Ceci a pour conséquence de réduire effectivement la variance lors du calcul de la fonction de coût afin d'entraîner notre réseau de neurones.

2.4. **Performances du CLD.** En pratique, les performances du CLD sont au moins comparables à celles des SGMs actuels, et permet une convergence bien plus rapide que ses concurrents. Dockhorn et al. ([5]) ont montré que le score FID sur l'ensemble de données CIFAR-10 du CLD est inférieur (ou égal) à celui de la plupart des SGMs présentés dans la littérature récente, et que la convergence est atteinte quatre fois plus vite que ces derniers.

3. Liens avec les ponts de Schrödinger

Cette dernière section est dédiée à la présentation du lien entre les ponts de Schrödinger (plus précisément : les ponts de Schrödinger diffusifs, ou *Diffusion Schrödinger Bridge*) et les SGMs, et a pour objectif principal de donner un second exemple des liens concrets potentiellement développables entre les SGMs et d'autres modèles génératifs. Soulignons qu'en pratique, la formulation sous forme de *Score Matching* des ponts de Schrödinger diffusifs présentés

5. Rappelons que la méthode d'Euler approche le processus à l'ordre $\mathcal{O}(\delta t)$.
6. Soulignons le fait que la différence majeure entre le DSM et le HSM provient du fait que l'espérance est calculée selon $p_0(\mathbf{u}_0)$ dans le cas du DSM, alors qu'elle est calculée selon $p_0(\mathbf{x}_0)$ dans le HSM.

ici est moins performantes que d'autres méthodes qui ne seront qu'énoncées ici, et n'a qu'un intérêt théorique.
La présentation faite ici provient majoritairement de l'article de De Bortoli et al. ([3]).

3.1. **Ponts de Schrödinger.** Commençons par rappeler la définition d'un pont de Schrödinger. On se place dans le cas du temps discret, et on considère donc $\mathcal{P}_{N+1} := \mathcal{P}\left(\left(\mathbb{R}^D\right)^{N+1}\right)$, l'ensemble des mesures de probabilités sur $\mathcal{X} := \left(\mathbb{R}^D\right)^{N+1}$, $N \geq 2$ étant fixé et correspondant au nombre de pas dans la discrétisation. Pour insister sur le fait que l'on considère des suites de points, on notera dans la suite $x_{0:N}$ − plutôt que x − un élément de $\mathcal{X}$.
Fixons une probabilité de référence $p \in \mathcal{P}_{N+1}$. Le problème du pont de Schrödinger consiste à trouver une probabilité $\pi^* \in \mathcal{P}_{N+1}$ telle que :

$$\pi^* = \operatorname{argmin}\left\{D_{\mathrm{KL}}(\pi|p) \ : \ \pi \in \mathcal{P}_{N+1} \text{ t.q. } \pi_0 = p_{\mathrm{data}},\ \pi_N = p_{\mathrm{noise}}\right\}, \tag{8}$$

où D_{KL} désigne la divergence de Kullback-Leibler.
Il est alors possible de se servir de π^* afin de générer des données synthétiques. En effet, si l'on simule un processus via $X_N \sim p_{\mathrm{noise}}$ et via la dynamique $X_k \sim \pi^*_{k|k+1}(X_{k+1})$ pour $k \in \{0, \cdots, N-1\}$, alors X_0 permet de simuler p_{data}.

Remarque 3.1. Formulés en 1932 par Schrödinger ([10]), les ponts de Schrödinger ont été construits afin de répondre à la question suivante : si l'on observe des particules suivant un mouvement brownien à $t = 0$ (autrement dit : elles suivent une distribution gaussienne) et que, à $t = T$, elles ne suivent plus une distribution gaussienne, quelle est l'évolution la plus probable ?

En pratique, le problème (8) ne peut pas être résolu analytiquement, à l'exception de rares cas. En revanche, on peut le résoudre numériquement via une méthode appelée *Iterative Proportional Fitting* (IPF). Cette méthode récursive est initialisée par $\pi^0 = p$, puis, pour tout $n \in \mathbb{N}$:

$$\pi^{2n+1} = \operatorname{argmin}\left\{D_{\mathrm{KL}}(\pi|\pi^{2n}) \ : \ \pi \in \mathcal{P}_{N+1} \text{ t.q. } \pi_N = p_{\mathrm{noise}}\right\},$$
$$\pi^{2n+2} = \operatorname{argmin}\left\{D_{\mathrm{KL}}(\pi|\pi^{2n+1}) \ : \ \pi \in \mathcal{P}_{N+1} \text{ t.q. } \pi_0 = p_{\mathrm{data}}\right\}.$$

3.2. **Ponts de Schrödinger diffusifs.** Les ponts de Schrödinger diffusifs consistent en une alternative aux méthodes de calcul de l'IPF (i.e. aux algorithmes permettant le calcul de la suite $(\pi^n)_n$), moins difficile à mettre en place. Elle consiste à adapter la méthode vue dans la Remarque 1.3 afin de construire un schéma dont la loi approche les lois π^n.
On dispose de la représentation suivante pour les π^n, $n \in \mathbb{N}$.

Proposition 3.1. Supposons que $D_{\mathrm{KL}}(p_{\mathrm{data}} \otimes p_{\mathrm{noise}}|p_{0,N}) < +\infty$. Alors, pour tout $n \in \mathbb{N}$, π^{2n} et π^{2n+1} ont une densité strictement positive par rapport à la mesure de Lebesgue, notées respectivement p^n et q^n. De plus, pour tout $x_{0:N} \in \mathcal{X}$, on a $p^0(x_{0:N}) = p(x_{0:N})$ et :

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

Remarque 3.2. En pratique, nous aurons accès à $q^n_{k|k+1}$ et $p^n_{k+1|k}$, mais nous pouvons toujours utiliser les formules précédentes à l'aide de la formule de Bayes.

La Proposition 3.1 met l'emphase sur le fait que, dans l'IPF, les termes pairs et impairs de la suite (π^n) jouent le rôle de processus allant dans des directions temporelles opposées. Autrement dit, les termes pairs semblent suivre la flèche du temps dans un sens, et l'on passe au terme suivant en utilisant les termes impairs, dont le temps est inversé. Cela motive l'idée d'approcher les deux suites de densités similairement à ce qui est fait dans les schémas d'Euler.

L'idée des ponts de Schrödinger diffusifs est donc d'approcher la formule de récurrence définissant l'IPF de la même façon que l'on approche une diffusion à temps inversé à l'aide d'un schéma d'Euler. Supposons donc qu'à une certaine étape $n \in \mathbb{N}$, on ait [7] :

$$p^n_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; x_k + \gamma_{k+1} f^n_k(x_k), 2\gamma_{k+1}\mathbf{I}),$$

7. On retiendra le parallèle entre cette situation et le schéma d'Euler pour un SGM construit à partir d'un processus d'Ornstein-Uhlenbeck, voir Remarque 1.3.

où f_k^n est une fonction. Pour $n=0$, on prend $f_k^0(x) := -\alpha x$, $\alpha \geq 0$.
Alors, à l'aide de la Proposition 3.1 et du fait que $p_k^n \simeq p_{k+1}^n$, on peut faire l'approximation :

$$\begin{aligned} q_{k|k+1}^n(x_k|x_{k+1}) &= p_{k+1|k}^n(x_{k+1}|x_k) \exp\left(\log p_k^n(x_k) - \log p_{k+1}^n(x_{k+1})\right) \\ &\simeq \mathcal{N}\left(x_k; x_{k+1} + \gamma_{k+1} b_{k+1}^n(x_{k+1}), 2\gamma_{k+1}\mathbf{I}\right), \end{aligned}$$

où $b_{k+1}^n(x_{k+1}) := -f_k^n(x_{k+1}) + 2\nabla \log p_{k+1}^n(x_{k+1})$.
De la même manière, on a :

$$p_{k+1|k}^{n+1}(x_{k+1}|x_k) \simeq \mathcal{N}\left(x_{k+1}; x_k + \gamma_{k+1} f_k^{n+1}(x_k), 2\gamma_{k+1}\mathbf{I}\right),$$

avec $f_k^{n+1}(x_k) := -b_{k+1}^n(x_k) + 2\nabla \log q_k^n(x_k)$.
Pour résumer, pour construire les suites de densités souhaitées, il suffit de construire les suites de fonctions f_k^n et b_k^n. Par ailleurs, on trouve que $f_k^{n+1}(x_k) = f_k^n(x_k) + 2\nabla \log q_k^n(x_k) - 2\nabla \log p_{k+1}^n(x_{k+1})$ (et de même pour b_k^{n+1} en fonction de b_k^n). En itérant cette formule de récurrence, on trouve que pour tous $n \in \mathbb{N}$, $x \in \mathbb{R}^D$ et $k \in \{0, \cdots, N-1\}$:

$$f_k^n(x) = -\alpha x + 2\sum_{j=0}^{n-1} \nabla \log q_k^j(x) - 2\sum_{j=0}^{n-1} \nabla \log p_{k+1}^j(x),$$

de sorte que l'on peut estimer f_k^{n+1} et b_k^{n+1} en estimant par *Score Matching* − à la manière du DSM dans le cas du SGM − les scores $\nabla \log p_{k+1}^i$ et $\nabla \log q_k^i$, $i = 0, \cdots, n$.

3.3. **Formulation sous forme de Score Matching.** Pour simplifier les notations, on pose :

$$\begin{aligned} B_{k+1}^n(x) &:= x + \gamma_{k+1} b_{k+1}^n(x) \\ F_k^{n+1}(x) &:= x + \gamma_{k+1} f_k^{n+1}(x) \end{aligned}$$

En utilisant la loi conditionnelle de x_{k+1} sachant x_k, ainsi que le Théorème de Convergence Dominée, on trouve que :

$$\begin{aligned} \nabla \log p_{k+1}^n(x_{k+1}) &= \int_{\mathbb{R}^D} \frac{F_k^n(x_k) - x_{k+1}}{2\gamma_{k+1}} p_{k|k+1}^n(x_k|x_{k+1}) \mathrm{d}x_k \\ &= \mathbb{E}_{(X_k, X_{k+1}) \sim p_{k,k+1}^n} \left[\frac{F_k^n(X_k) - X_{k+1}}{2\gamma_{k+1}} \middle| X_{k+1} = x_{k+1} \right]. \end{aligned}$$

On en déduit la Proposition suivante.

Proposition 3.2 ([3], Proposition 28)**.** Pour tout $N \in \mathbb{N}^*$ et tout $k \in \{0, \cdots, N-1\}$, on a :

$$\begin{aligned} \nabla \log p_{k+1}^n &= \underset{u \in L^2(\mathbb{R}^D, \mathbb{R}^D)}{\operatorname{argmin}} \mathbb{E}_{(X_k, X_{k+1}) \sim p_{k,k+1}^n} \left\| u(X_{k+1}) - \frac{F_k^n(X_k) - X_{k+1}}{2\gamma_{k+1}} \right\|_2^2, \\ \nabla \log q_k^n &= \underset{v \in L^2(\mathbb{R}^D, \mathbb{R}^D)}{\operatorname{argmin}} \mathbb{E}_{(X_k, X_{k+1}) \sim p_{k,k+1}^n} \left\| v(X_{k+1}) - \frac{B_{k+1}^n(X_k) - X_{k+1}}{2\gamma_{k+1}} \right\|_2^2. \end{aligned}$$

En pratique, les scores sont approchés par des réseaux de neurones : $u_{\alpha^j}(k,x)$, $v_{\beta^j}(k,x)$, $j = 1, \cdots, N$. Les densités p^n et q^n sont alors échantillonnées via les schéma d'Euler associés aux EDS de drifts f_k^n et b_k^n. Par exemple, pour échantillonner p^n, on considère la suite :

$$X_{k+1}^n = (1 - \alpha\gamma_{k+1}) X_k^n + 2\gamma_{k+1} \left(\sum_{j=0}^n u_{\alpha^j}(k+1, X_k^n) - \sum_{j=0}^n v_{\beta^j}(k, X_k^n) \right) + \sqrt{2\gamma_{k+1}} Z_{k+1}^n,$$

avec $\gamma_{k+1} > 0$ le pas de temps, et $Z_{k+1}^n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
Il faut donc $2n$ réseaux de neurones pour cette procédure, ce qui peut s'avérer très coûteux en termes de mémoire. Une alternative à cette méthode consiste à approcher directement les drifts f_k^n et b_k^n, afin de n'avoir qu'un nombre constant de réseaux de neurones à stocker à chaque étape. C'est l'objet des Proposition 3 et Proposition 29 de De Bortoli et al. ([3]). Par exemple :

Proposition 3.3 ([3], Proposition 29)**.** Pour tout $n \in \mathbb{N}$ et tout $k \in \{0, \cdots, N-1\}$, on a :

$$b_{k+1}^n = \underset{b \in L^2(\mathbb{R}^D, \mathbb{R}^D)}{\operatorname{argmin}} \mathbb{E}_{(X_k, X_{k+1}) \sim p_{k,k+1}^n} \left\| b\left(X_{k+1}\right) - \frac{F_k^n\left(X_k\right) - F_k^n\left(X_{k+1}\right)}{\gamma_{k+1}} \right\|_2^2,$$

$$f_k^{n+1} = \underset{f \in L^2(\mathbb{R}^D, \mathbb{R}^D)}{\operatorname{argmin}} \mathbb{E}_{(X_k, X_{k+1}) \sim p_{k,k+1}^n} \left\| f\left(X_k\right) - \frac{B_{k+1}^n\left(X_{k+1}\right) - B_{k+1}^n\left(X_k\right)}{\gamma_{k+1}} \right\|_2^2.$$

Dans ce cas, il suffit d'approcher les drifts par deux réseaux de neurones $f_{\alpha^n}(k,x) \simeq f_k^n(x)$ et $b_{\beta^n}(k,x) \simeq b_k^n(x)$. On peut alors simuler p^n et q^n via des schémas d'Euler, comme précédemment.

Finalement, bien que cette approche ne soit pas aussi efficace que l'état de l'art en génération de données synthétiques, elle est la première méthode permettant de résoudre les problèmes de ponts de Schrödinger en grande dimension. En outre, les ponts de Schrödinger diffusifs nécessitent en pratique moins de pas de temps lors de leur discrétisation que les SGMs, ce qui les rend très prometteurs pour l'avenir.

4. Ouvertures

Du fait de l'arrivée très récente des SGMs, de nombreuses questions restent encore ouvertes, tant d'un point de vue théorique que numérique.

- Peu de bornes théoriques existent, en particulier pour les modèles récents comme le CLD. De plus, il n'existe que très peu de littérature concernant l'application des SGMs à d'autres structures de données que les images (e.g. les données sur des variétés ou les séries temporelles). Soulignons cependant que certains articles concernant cette dernière problématiques ont vu le jour en 2022 (e.g. [4]). Finalement, nous pouvons également chercher à déterminer de nouveaux liens entre les SGMs et les précédents modèles génératifs (comme Liu et al. [9], qui réutilise le Score Matching dans le cadre des GANs).
- D'un point de vue numérique, de nombreux procédés sont proposés dans la littérature (Denoising Diffusion Implicit Models, knowledge distillation, diffusions non-gaussiennes...) pour améliorer la vitesse de convergence des SGMs. Il serait donc intéressant de comparer leurs performances relatives. Par ailleurs, soulignons que l'équation (1) donne une multitude de modèles possibles pour les SGMs via le choix du drift et du coefficient de diffusion. Ainsi, la recherche de nouvelles dynamiques pour améliorer les performances des SGMs reste encore d'actualité du fait de l'émergence récente de ces modèles.

Références

[1] F. Bordes et al. Learning to Generate Samples from Noise through Infusion Training. 2017. doi : 10.48550/ARXIV.1703.06975. url : https://arxiv.org/abs/1703.06975.

[2] V. De Bortoli et al. Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling. 2021. doi : 10.48550/ARXIV.2106.01357. url : https://arxiv.org/abs/2106.01357.

[3] V. De Bortoli et al. Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling. 2021. doi : 10.48550/ARXIV.2106.01357. url : https://arxiv.org/abs/2106.01357.

[4] V. De Bortoli et al. Riemannian Score-Based Generative Modeling. 2022. doi : 10.48550/ARXIV.2202.02763. url : https://arxiv.org/abs/2202.02763.

[5] T. Dockhorn et al. Score-Based Generative Modeling with Critically-Damped Langevin Diffusion. 2021. doi : 10.48550/ARXIV.2112.07068. url : https://arxiv.org/abs/2112.07068.

[6] A. Goyal et al. Variational Walkback : Learning a Transition Operator as a Stochastic Recurrent Net. 2017. doi : 10.48550/ARXIV.1711.02282. url : https://arxiv.org/abs/1711.02282.

[7] U. G. Haussmann et E. Pardoux. « Time Reversal of Diffusions ». In : The Annals of Probability 14.4 (1986), p. 1188-1205. issn : 00911798. url : http://www.jstor.org/stable/2243859 (visité le 14/05/2022).

[8] J. Ho et al. « Denoising Diffusion Probabilistic Models ». In : Advances in Neural Information Processing Systems. T. 33. Curran Associates, Inc., 2020, p. 6840-6851. url : https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[9] S. Liu et al. DiffGAN-TTS : High-Fidelity and Efficient Text-to-Speech with Denoising Diffusion GANs. 2022. doi : 10.48550/ARXIV.2201.11972. url : https://arxiv.org/abs/2201.11972.

[10] E. Schrödinger. « Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique ». fr. In : Annales de l'institut Henri Poincaré 2.4 (1932), p. 269-310. url : http://www.numdam.org/item/AIHP_1932__2_4_269_0/.

[11] Y. Song et S. Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. 2019. doi : 10.48550/ARXIV.1907.05600. url : https://arxiv.org/abs/1907.05600.

[12] Y. Song et al. Score-Based Generative Modeling through Stochastic Differential Equations. 2020. doi : 10.48550/ARXIV.2011.13456. url : https://arxiv.org/abs/2011.13456.

[13] Y. Song et al. Sliced Score Matching : A Scalable Approach to Density and Score Estimation. 2019. doi : 10.48550/ARXIV.1905.07088. url : https://arxiv.org/abs/1905.07088.