

Génération conditionnelle et inconditionnelle de signaux sismiques à l'aide de modèles de diffusion

H. Gabrielidis^{1,2}, F. Gatti¹, S. Vialle^{2,3}, G. Jacquet¹

¹ Université Paris-Saclay, CentraleSupélec, ENS Paris-Saclay, CNRS, LMPS - Laboratoire de Mécanique Paris-Saclay UMR 9026
91190 Gif sur Yvette, France

{hugo.gabrielidis,filippo.gatti}@centralesupelec.fr

² Université Paris-Saclay, CNRS, LISN - Laboratoire interdisciplinaire des sciences du numérique, 91405, Orsay, France
stephane.vialle@centralesupelec.fr

³ CentraleSupélec, 91192 Gif sur Yvette, France

Résumé —

La génération de signaux sismiques synthétiques joue un rôle essentiel dans diverses applications géophysiques, notamment l'imagerie, l'exploration sismique et la simulation de tremblements de terre. Les méthodes traditionnelles de génération de signaux sismiques reposent souvent sur des modèles empiriques ou des simulations numériques, limitant ainsi leur capacité à saisir la nature complexe et dynamique des données sismiques réelles. Ces dernières années, les techniques d'apprentissage profond sont apparues comme des outils puissants pour les tâches de génération de séries temporelles, images, vidéos, texte. Dans cette étude, nous proposons un nouveau modèle d'apprentissage profond qui utilise la diffusion générative pour permettre la génération de signaux sismiques à large bande de fréquence (0-30 Hz), de manière conditionnelle et inconditionnelle. En adoptant une approche basée sur la diffusion, le modèle capture les dépendances temporelles inhérentes aux données sismiques. Les résultats montrent que notre modèle d'apprentissage profond basé sur la diffusion générative permet la génération de signaux sismiques à large bande de fréquence et de haute fidélité. Les signaux synthétiques ainsi générés présentent des caractéristiques sismiques réalistes, en temps et en fréquence, les rendant adaptés à des applications telles que l'augmentation des données sismiques, l'amélioration de la modélisation sismique synthétique et la détection d'anomalies sismiques.

Mots clés — Deep Learning, Signaux Sismiques, Modèles de diffusions

1 Introduction

Dans cette étude nous présentons une architecture permettant de générer des signaux sismiques réalistes à large bande (0-30 Hz) en conditionnant la génération à l'aide d'un signal sismique de basse fréquences. La tâche de génération de séries temporelles, faite de manière conditionnée ou pas, a toujours été un problème complexe, mais, de nos jours, les méthodes utilisant des réseaux de neurones connaissent un large succès [19, 4], notamment en s'inspirant du domaine de la "computer vision" et de l'imagerie [11, 3, 22]. Au sein de la panoplie de méthodes développées, les modèles de diffusion [11] se démarquent par leur facilité d'entraînement, par leur performance et leur faible taille [14, 10, 23] et par la possibilité d'utiliser une grande variété d'architecture pré-existantes. Dans le cadre du problème de "super-résolution", qui fait l'objet de ce papier, il s'agit de générer plusieurs séries temporelles à large bande fréquentielle (haute résolution), en conditionnant le processus par un signal à basses fréquences (résolution faible) qui doit rester le même pour toutes les séries ainsi générées. Dans le domaine de la génération de signaux sismiques, les études se concentrent principalement sur l'imagerie sismique [20, 5] et sur la génération conditionnelle et inconditionnelle de signaux sismiques à l'aide de réseaux de neurones [8, 6]. Néanmoins celles-ci se concentrent habituellement sur une application des modèles GAN (Generative Adversarial Networks [9]), modèle étant connu pour les difficultés de convergence lors de leurs entraînements (e.g., le phénomène de *mode collapse* [25] et les difficultés de convergence liées à leurs architectures soulignées par [1]). Un modèle de diffusions pour le débruitage de signaux sismiques a été présenté par [26], travail qui a inspiré une partie de nos recherches et ce papier.

2 Méthode

2.1 Processus de diffusion

Les modèles probabiliste de diffusions (ou juste modèles de diffusion) sont une classes de modèle utilisant un échantillonnage en chaîne de Markov de Monte-Carlo pour générer des données provenant d'une distribution de bruit simple telle que la distribution normale. La chaîne de Markov d'un modèle de diffusion se compose de deux processus : le *forward process* et le *backward process* (voir Figure 1).

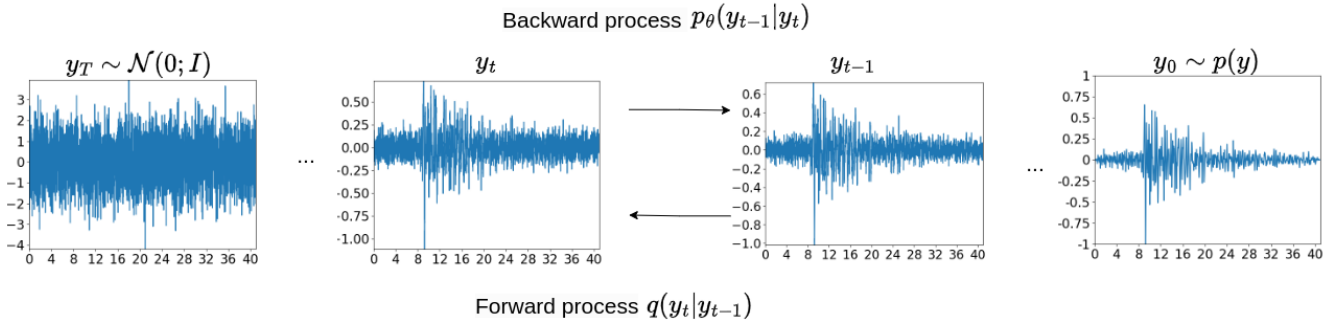


FIGURE 1 – Schéma du processus de diffusion : un bruit blanc $y_T \sim \mathcal{N}(0, I)$ est progressivement transformé en signal sismique $y_0 \sim p(y)$, en estimant la probabilité conditionnée $p_\theta(y_{t-1}|y_t)$, associée aux paramètres θ du réseau de neurones. Lorsque celui-ci est conditionné par un signal à basse fréquence x , la probabilité de transition de Markov devient $p_\theta(y_{t-1}|y_t, x)$.

En nommant y le sismogrammes à 3 composantes, le *forward process* consiste à générer des échantillons $(y_t)_{t=1}^T$, où T est la durée du processus de diffusion, en ajoutant progressivement un bruit Gaussien à une série temporelle y_0 de départ, choisie dans la base de données à disposition. Afin de générer les instances y_1, y_2, \dots, y_T on définit en amont une variance de bruit $(\beta_t)_{t=1}^T$ et les instances $(y_t)_{t=1}^T$ sont générées selon la probabilité de transition q Gaussienne, de moyenne $y_{t-1}\sqrt{1-\beta_t}$ et d'auto-corrélation $\beta_t I$, i.e. :

$$q(y_t|y_{t-1}) := \mathcal{N}(y_{t-1}\sqrt{1-\beta_t}, \beta_t I) \quad (1)$$

Généralement, $T = 1000$ et β_t est incrementé linéairement entre 0.0001 et 0.02 [11]. De l'Équation 1, nous pouvons déduire l'expression suivante de la distribution de probabilité q :

$$q(y_t|y_0) := \mathcal{N}(\sqrt{\alpha_t}y_0, (1-\alpha_t)I) \quad \alpha_t = 1 - \beta_t, \quad \bar{\alpha}_t = \prod_{s=1}^t \alpha_s \quad (2)$$

Le *backward process* est défini comme l'inverse du *forward process* et permet une approximation de y_0 en partant de y_T , avec y_T échantillonné d'une loi normale centrée réduite $p(y_T) = \mathcal{N}(y_T; 0, I)$. Le processus inverse estime y_0 en maximisant la vraisemblance $p_\theta(y_0|y_T)$, définie comme le produit des probabilités de transition, selon l'expression suivante :

$$p_\theta(y_{t-1}|y_t) := \mathcal{N}(\mu_\theta(y_t, t), \sigma_t^2 I) \quad p_\theta(y_0|y_T) := p(y_T) \prod_{t=1}^T p_\theta(y_{t-1}|y_t) \quad (3)$$

avec $\mu_\theta(y_t, t)$ et σ_t^2 définis comme la moyenne et la variance estimées par le modèle de y_{t-1} . En pratique, un réseau de neurones ε_θ est utilisé pour paramétrer $p_\theta(y_{t-1}|y_t)$. Ce réseau de neurones infère le bruit Gaussien $\varepsilon_\theta(y_t, t)$ nécessaire à estimer $y_{t-1} \sim p_\theta(y_{t-1}|y_t)$, selon l'expression suivante (Ho *et al* [11]) :

$$y_{t-1} = \frac{1}{\sqrt{\alpha_t}} \cdot \left(y_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}} \varepsilon_\theta(y_t, t) \right) + \sqrt{\beta_t} z, \quad z \sim \mathcal{N}(0, I) \quad (4)$$

Comme suggéré par Ho *et al* [11], la variance du processus stochastique y_{t-1} est une variance uniquement dépendante du temps. $\varepsilon_\theta(y_t, t)$ représente l'estimation du niveau de bruit contenu dans y_t au pas de temps t . L'enjeu des modèles de diffusion est donc d'estimer le bruit généré par le *forward process*, en permettant ainsi d'obtenir y_0 à partir de y_T .

Traditionnellement [11], les modèles de diffusions sont entraînés à maximiser l’*Evidence Lower Bound* (ELBO [13]), qui correspond à minimiser la borne supérieure de la vraisemblance logarithmique négative, selon l’expression suivante :

$$\begin{aligned} \mathbb{E}_{y_0 \sim p_\theta} [-\log p_\theta(Y_0)] &\leq \mathbb{E}_{y_0 \sim p_\theta} \left[\mathbb{E}_{y_{1:T} \sim q(\cdot|y_0)} \left[-\log \frac{p_\theta(Y_{0:T})}{q(\cdot|Y_0)} \right] \right] = \\ &= \mathbb{E}_{y_0 \sim p_\theta} \left[\mathbb{E}_{y_{1:T} \sim q(\cdot|y_0)} \left[-\ln p(Y_T) - \sum_{t \geq 1} \ln \frac{p_\theta(Y_{t-1}|Y_t)}{q(Y_t|Y_{t-1})} \right] \right] = L(\theta) \end{aligned} \quad (5)$$

Minimiser la fonction coût $L(\theta)$ dans l’Équation 5 correspond à minimiser la divergence de Kullback-Leibler $\mathbb{D}_{KL}(q(Y_{1:T}|y_0) \| p_\theta(y_0, Y_{1:T}))$. Il est possible de réécrire et simplifier la fonction coût $L(\theta)$ sous la forme suivante :

$$L(\theta) = \mathbb{E}_{t, y_0, \varepsilon, x} \left[\|\varepsilon - \varepsilon_\theta \left(\sqrt{\bar{\alpha}_t} y_0 + \sqrt{1 - \bar{\alpha}_t} \varepsilon, t \right)\|_2^2 \right] \quad (6)$$

L’avantage de la fonction coût dans l’Équation 6 est qu’elle n’introduit pas des termes de pénalisation additionnels, dont les coefficients sont difficiles à calibrer (comme dans le cadre des GAN [17, 7]). Les procédures d’entraînement et de génération sont résumés dans les algorithmes 1 et 2.

Algorithm 1 Algorithme d’entraînement par Gradient Stochastique sur N_b mini-batches de taille n .

```

1:  $i = 0, \theta^{(i)} := \theta_0$ 
2: for  $L(\theta^{(i)}) > tol$  do
3:   for  $b \sim \mathcal{U}(\{1, \dots, N_b\})$  do
4:      $\ell = 0$ 
5:     for  $j \sim \mathcal{U}(\{1, \dots, n\})$  do
6:       for  $y_{0,j} \in \mathcal{D}$  do
7:         for  $t \sim \mathcal{U}(\{1, \dots, T\})$  do
8:            $\varepsilon \sim \mathcal{N}(0, I)$ 
9:

```

$$\ell += \|\varepsilon - \varepsilon_\theta \left(\sqrt{\bar{\alpha}_t} y_{0,j} + \sqrt{1 - \bar{\alpha}_t} \varepsilon, t \right)\|_2^2$$

```

10:      end for
11:    end for
12:  end for
13:   $L(\theta^{(i)}) := \frac{\ell}{N_b}$ 
14:   $\theta^{(i+1)} := \theta^{(i)} - \eta \cdot \nabla_{\theta} L(\theta^{(i)})$ 
15: end for
16: end for

```

Algorithm 2 Algorithme de prédiction

```

1:  $y_T \sim \mathcal{N}(0, I)$ 
2: for  $t = T, \dots, 1$  do
3:    $z \sim \mathcal{N}(0, I)$  if  $t > 1$  else  $z = 0$ 
4:    $y_{t-1} = \frac{1}{\sqrt{\alpha_t}} (y_{t-1} - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \varepsilon_\theta(y_t, t)) + \sqrt{\beta_t} z$ 
5: end for
6:

```

Lorsque nous conditionnons l’entraînement et la génération avec un signal basse fréquence x , alors $\varepsilon_\theta(y_t, t)$ devient $\varepsilon_\theta(y_t, x, t)$, ce qui revient à introduire x dans le réseau de neurones, dont l’architecture est décrite à la section suivante.

2.2 Architecture du modèle

Dans cette section, nous décrivons les architectures du modèle ε_θ (voir section précédente) de prédiction des signaux sismiques à large-bande fréquentielle y conditionné par le signal à basse fréquence x généré par simulation numérique. L’objectif est donc d’estimer le bruit Gaussien du *forward process* à l’aide de $\varepsilon_\theta(y_t, t)$ ou $\varepsilon_\theta(y_t, x, t)$ en obtenant, à chaque étape de la diffusion, y_{t-1} à partir de y_t . À cette fin, plusieurs architectures ont été étudié, l’architecture présentant les meilleurs résultats combine un UNet [21] avec une adaptation des blocs ResBlockDown et ResBlockUp, utilisés dans le modèle BigGAN [2] ainsi que des mécanismes de *cross attention* [27].

L'architecture UNet originellement utilisée dans des tâches de segmentations, démontre d'excellentes performances dans le cadre de la diffusion. En effet, les algorithmes de diffusion ne nécessitent pas de projection explicite des données dans un espace latent, contrairement aux Variational Auto-Encoder (VAE). Il est donc possible d'utiliser des architectures avec des connexions résiduelles par concaténation (type UNet), qui aident à propager les informations des couches antérieures vers les couches plus profondes et en contribuant à atténuer le problème de la disparition des gradients (*vanishing gradients*) et en transmettant directement les informations à travers le réseau. Dans la figure 2 nous adaptions cette architecture UNet en utilisant les blocs BigGANUp et BigGANDown décrits précédemment.

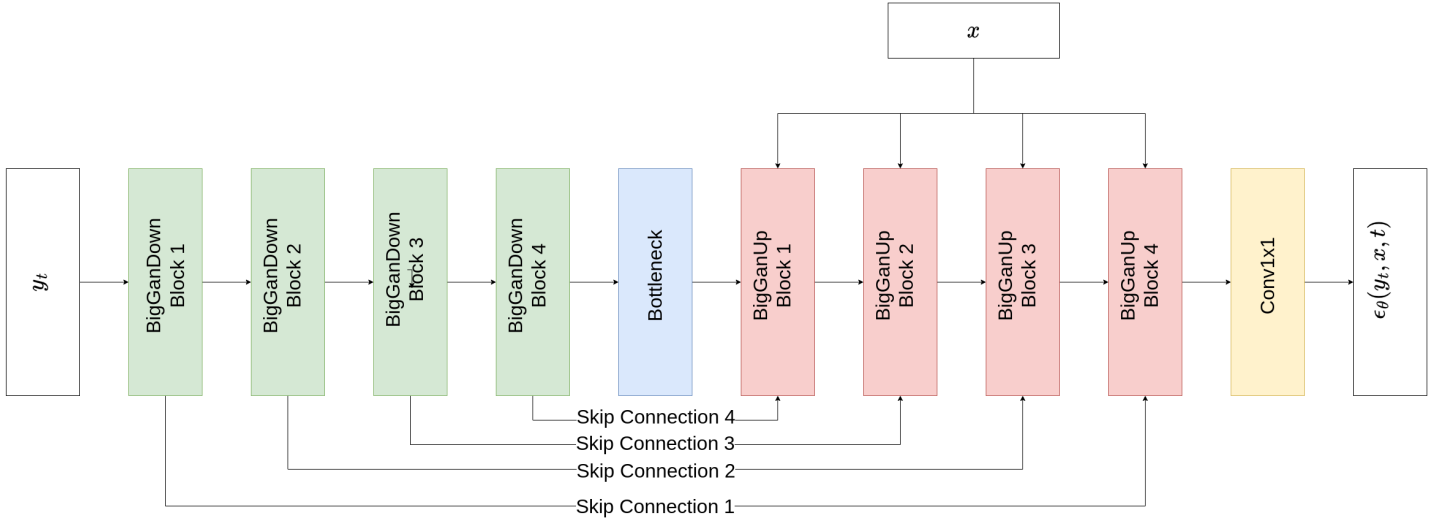


FIGURE 2 – Architecture du réseau de neurones. Le signal y_t , le signal basse fréquence x et l'instant temporel t sont les entrées du modèle. Ce modèle estime le bruit ϵ à l'aide de ϵ_θ , afin de reconstruire y_0 depuis y_t

Les blocs BigGANdown et BigGANUp sont constitués de deux branches chacun. La branche profonde intègre plusieurs noyaux de convolution pour réduire la taille des *features*; l'autre branche produit le nombre requis de filtres de sortie. Les deux branches sont reliées en sommant leurs sorties. La composante temporelle est ajoutée à l'entrée de chaque bloc *down* et *up*, après un *embedding* réalisé par encodage sinusoïdale [15], suivi deux fois par une couche dense et une fonction d'activation geLU. Lors d'une génération conditionnée par le signal à basse fréquence x , des couches de *cross-attention* sont utilisés dans les blocs BigGANUp de la façon suivante : $Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V$, avec

$$Q = W_Q^{(i)} * z_t, K = W_K^{(i)} * x, V = W_K^{(i)} * x$$

avec z_t comme représentation intermédiaire de l'entrée y_t produite par les différentes couches du modèle et Q, K, V en tant des matrices de poids apprises pendant l'entraînement.

Dans le cadre d'une génération non-conditionnée, les couches de *cross-attention* sont remplacés par des couches de *self-attention*. Dans ce cas :

$$Q = W_Q^{(i)} * z_t, K = W_K^{(i)} * z_t, V = W_K^{(i)} * z_t$$

3 Configuration des expérimentation

3.1 Jeux de données

Nous entraînons nos modèles sur la database STEAD [18], dont nous avons extrait de celui-ci aléatoirement 51200 signaux. Pour de meilleurs performances, les signaux sont normalisés entre $[-1, 1]$ et une fenêtre de Hamming est appliquée réduisant le signal au début de l'onde P et à la fin de l'onde S.

Dans le cadre de la génération inconditionnelle nous utilisons uniquement le signal hautes fréquences en tant qu’objectif et en partant d’un bruit extrait d’une loi normale centrée réduite. Dans le contexte d’une génération conditionnelle le signal basses fréquences x est ajouté en tant qu’entrée du modèle durant l’entraînement et la génération. Le dataset est décomposé en trois datasets, un dataset d’entraînement composé de 80% des signaux, un dataset de validation composé de 10% des signaux permettant d’optimiser les hyperparamètres et un dataset test également composé de 10% des signaux permettant de juger les performances des modèles.

3.2 Entraînement et évaluation

Afin d’entraîner nos modèles nous utilisons l’optimiseur AdamW [16], dont il a été prouvé qu’il présente de meilleures capacités de généralisation que Adam [12] avec un *learning rate* de $1e^{-4}$ et un *weight decay* de 0.1. Les modèles sont entraînés durant 200 *epochs*. Le *batch size* utilisé est de 64 par GPU pour l’entraînement sur 3 GPU A100 (disponible au dans le supercalculateur Jean Zay ou au mésocentre Paris Saclay). L’entraînement du modèle final a duré 5 heures, soit 15h GPUs A100 au total. Lors de l’évaluation de notre modèle, nous avons tout d’abord utilisé l’algorithme 2 de génération décrit à la section 2.1 en utilisant 1000 pas de temps, puis l’algorithme DDIM [24] afin d’améliorer la vitesse de génération. Dans les deux cas la qualité de génération s’est montré équivalente.

4 Résultats des expérimentations

Comme montré figure 3, nos modèles sont capables de générer des signaux sismiques de hautes fréquences réalistes à partir de signaux de basses fréquences.

Dans le contexte d’une génération conditionnelle les métriques *Erreur quadratique moyenne (MSE)* et *ratio signal bruit (SNR)* sont utilisés pour comparer le signal hautes fréquences réel y_0 avec le signal généré par le modèle \hat{y}_0 . En vue de comparer notre modèle nous établissons une *baseline*, celle-ci est une architecture simple de type UNet générant le signal hautes fréquences directement à partir du signal basses fréquences. Notre *baseline* comporte autant de paramètres que notre modèle de diffusion et est également entraîné durant 200 *epochs*. Comme il est possible de le voir table 1 notre modèle présente des résultats légèrement meilleurs que la baseline proposée.

Modèle	MSE	SNR	Nombre de paramètres
UNet	$4e^{-3}$	12.04	14.3M
Notre modèle de diffusion	$2e^{-3}$	16.43	13.0M

TABLE 1 – Résultats des modèles sur le *testing set*

De plus, comme on peut l’observer figure 4, étant donné que notre modèle part d’un bruit gaussien il est possible de générer plusieurs signaux hautes fréquences à l’aide d’une unique base basses fréquences cela en faisant varier le bruit initial et le nombre de pas de diffusion utilisés lors de la génération. Il est également intéressant d’observer que les signaux hautes fréquences générés sont fidèles au signal de base pour les parties basses fréquences. Cette capacité de générer plusieurs signaux à partir d’une même base, également appelée *one-to-many*, est utile dans le cadre d’une augmentation du jeu de données et il permet de générer plusieurs signaux sismiques à large bande fréquentielle pour les études d’aléa et vulnérabilité sismiques. De plus, ces signaux étant conditionnés à basse fréquence par des simulations numériques, la physique du problème n’est pas perdue lors de la génération de données synthétiques.

5 Performances obtenues

Dans cette section nous décrivons les différentes performances de nos modèles et les méthodes utilisées afin de les améliorer. L’un des problèmes majeurs liés aux algorithmes de diffusions décrits dans Ho *et al* est le temps de génération. En effet, afin de générer y_0 à partir de y_T le modèle doit effectuer T passages dans le réseau avec $T = 1000$. Cette contrainte liée à l’aspect chaîne de Markov des réseaux

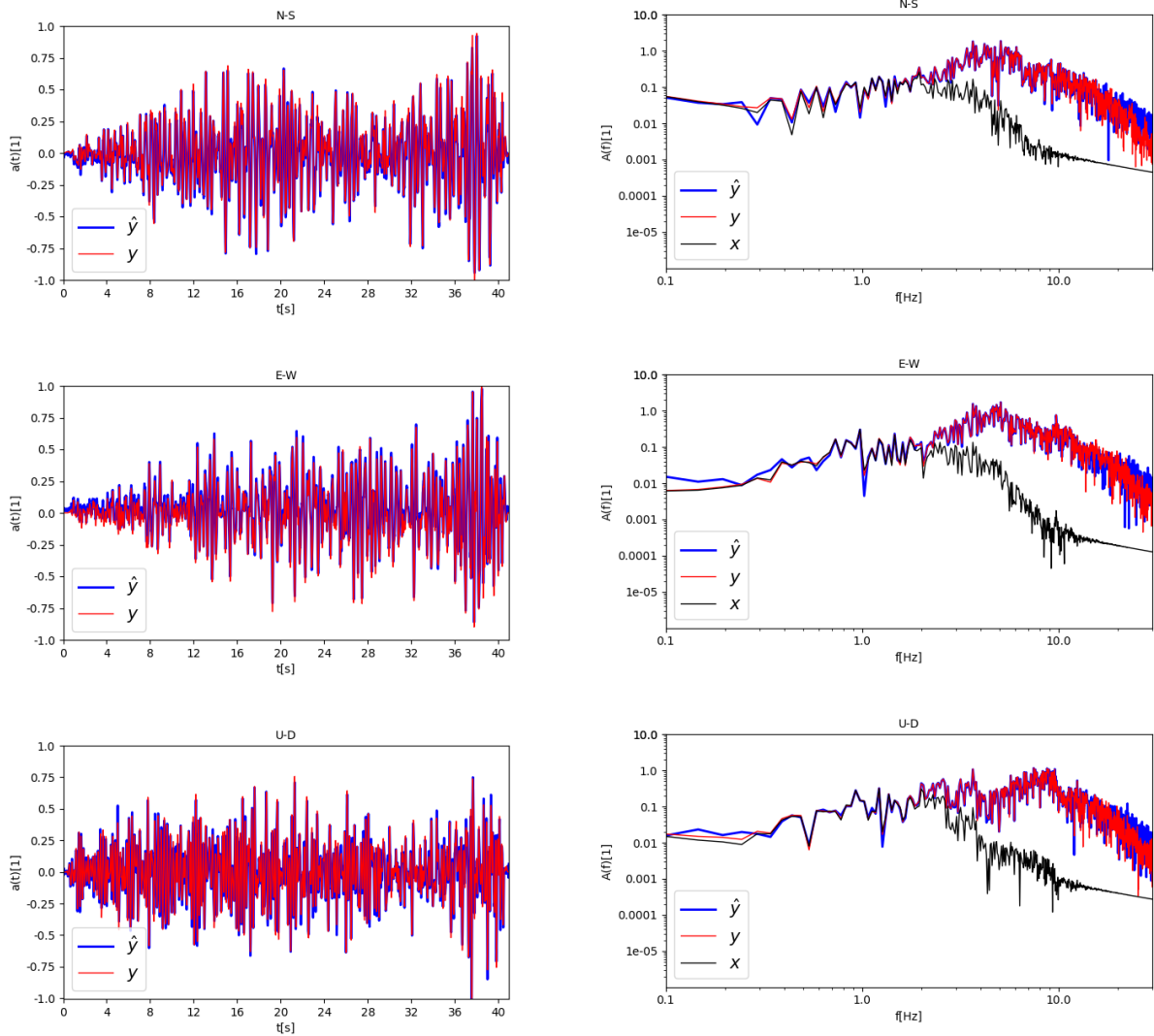


FIGURE 3 – Graphe d’amplitude et de fréquence des signaux générées et réels

de diffusions engendre une génération peu compétitive par rapport aux autres archétypes de modèles génératifs. Ainsi, en utilisant un GPU A100, pour la génération d’un *batch* de 100 signaux hautes fréquences notre modèle conditionnel initial prend 80s. Afin d’améliorer la vitesse de génération plusieurs approches ont été conçues. Dont l’approche *Denoising Implicit Models* de Song *et al* [24] que nous avons retenue. Celle-ci propose de généraliser les modèles de diffusion à une approche sans chaîne de Markov tout en gardant le même objectif d’entraînement que celui présenté à la section 2.1. Cette approche permet ainsi de réutiliser les modèles ayant été entraîné avec la méthode de diffusion classique, et permet de réduire le nombre de passages dans le réseau de 1000 à 8 dans notre cas. L’implémentation de cette méthode nous permet d’accélérer la génération de signaux d’un facteur 20 tout en une maintenant qualité de reconstruction similaire.

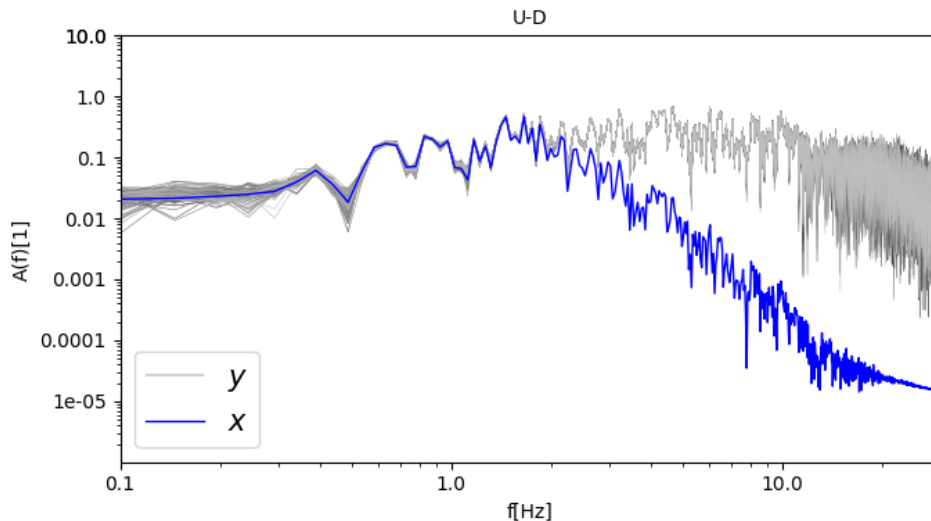


FIGURE 4 – Génération de 100 signaux hautes fréquences y_i à partir d’une unique base basses fréquences x

6 Conclusion

Dans ce papier, nous avons montré que l’utilisation d’algorithmes de diffusion pour la génération conditionnelle et inconditionnelle de signaux sismiques est possible. Cette approche permet un entraînement plus simple à l’aide de modèles plus variés par rapport aux méthodes concurrentes. Notre modèle de diffusion conditionnel obtient des résultats supérieurs avec moins de paramètres requis et est également capable de générer un grand nombre de signaux hautes fréquences en partant d’une unique base basses fréquences. Il est alors possible d’utiliser les signaux générés à des fins d’augmentation des données sismiques dans le but d’amélioration de la modélisation sismique synthétique ou de détection d’anomalies sismiques.

Dans de futurs travaux nous nous concentrerons sur la génération de plusieurs signaux hautes fréquences réalistes à partir d’une même base basse fréquence. A cette fin nous explorerons plus en détails les différentes méthodes de génération, appelées également *sampler*, créées pour les modèles de diffusion.

Références

- [1] Samuel A. BARNETT. *Convergence Problems with Generative Adversarial Networks (GANs)*. arXiv :1806.11382 [cs, stat]. Juin 2018. URL : <http://arxiv.org/abs/1806.11382>.
- [2] Andrew BROCK, Jeff DONAHUE et Karen SIMONYAN. *Large Scale GAN Training for High Fidelity Natural Image Synthesis*. arXiv :1809.11096 [cs, stat] version : 2. Fév. 2019. URL : <http://arxiv.org/abs/1809.11096>.
- [3] Jade COPET et al. *Simple and Controllable Music Generation*. arXiv :2306.05284 [cs, eess]. Juin 2023. URL : <http://arxiv.org/abs/2306.05284>.
- [4] Chris DONAHUE, Julian MCAULEY et Miller PUCKETTE. *Adversarial Audio Synthesis*. arXiv :1802.04208 [cs] version : 3. Fév. 2019. URL : <http://arxiv.org/abs/1802.04208>.
- [5] Ricard DURALL et al. “Deep Diffusion Models for Seismic Processing”. In : *Computers & Geosciences* 177 (août 2023). arXiv :2207.10451 [physics], p. 105377. ISSN : 00983004. DOI : 10.1016/j.cageo.2023.105377. URL : <http://arxiv.org/abs/2207.10451>.
- [6] Manuel A. FLOREZ et al. “Data-Driven Synthesis of Broadband Earthquake Ground Motions Using Artificial Intelligence”. In : *Bulletin of the Seismological Society of America* 112.4 (avr. 2022), p. 1979-1996. ISSN : 0037-1106. DOI : 10.1785/0120210264. URL : <https://doi.org/10.1785/0120210264>.
- [7] Swetava GANGULI, Pedro GARZON et Noa GLASER. *GeoGAN : A Conditional GAN with Reconstruction and Style Loss to Generate Standard Layer of Maps from Satellite Images*. arXiv :1902.05611 [cs]. Avr. 2019. URL : <http://arxiv.org/abs/1902.05611>.

- [8] Filippo GATTI et Didier CLOUTEAU. “Towards blending Physics-Based numerical simulations and seismic databases using Generative Adversarial Network”. en. In : *Computer Methods in Applied Mechanics and Engineering* 372 (déc. 2020), p. 113421. ISSN : 00457825. DOI : 10.1016/j.cma.2020.113421. URL : <https://linkinghub.elsevier.com/retrieve/pii/S004578252030606X>.
- [9] Ian J. GOODFELLOW et al. *Generative Adversarial Networks*. arXiv :1406.2661 [cs, stat]. Juin 2014. URL : <http://arxiv.org/abs/1406.2661>.
- [10] Seungu HAN et Junhyeok LEE. “NU-Wave 2 : A General Neural Audio Upsampling Model for Various Sampling Rates”. In : *Interspeech 2022*. arXiv :2206.08545 [cs, eess]. Sept. 2022, p. 4401-4405. DOI : 10.21437/Interspeech.2022-45. URL : <http://arxiv.org/abs/2206.08545>.
- [11] Jonathan HO, Ajay JAIN et Pieter ABBEEL. *Denoising Diffusion Probabilistic Models*. arXiv :2006.11239 [cs, stat]. Déc. 2020. URL : <http://arxiv.org/abs/2006.11239>.
- [12] Diederik P. KINGMA et Jimmy BA. *Adam : A Method for Stochastic Optimization*. arXiv :1412.6980 [cs]. Jan. 2017. DOI : 10.48550/arXiv.1412.6980. URL : <http://arxiv.org/abs/1412.6980>.
- [13] Diederik P. KINGMA et Max WELLING. “Auto-Encoding Variational Bayes”. en. In : arXiv :1312.6114 (déc. 2022). arXiv :1312.6114 [cs, stat]. URL : <http://arxiv.org/abs/1312.6114>.
- [14] Junhyeok LEE et Seungu HAN. “NU-Wave : A Diffusion Probabilistic Model for Neural Audio Upsampling”. In : *Interspeech 2021*. arXiv :2104.02321 [cs, eess]. Août 2021, p. 1634-1638. DOI : 10.21437/Interspeech.2021-36. URL : <http://arxiv.org/abs/2104.02321>.
- [15] Hezheng LIN et al. *CAT : Cross Attention in Vision Transformer*. arXiv :2106.05786 [cs]. Juin 2021. URL : <http://arxiv.org/abs/2106.05786>.
- [16] Ilya LOSHCHILOV et Frank HUTTER. *Decoupled Weight Decay Regularization*. arXiv :1711.05101 [cs, math]. Jan. 2019. URL : <http://arxiv.org/abs/1711.05101>.
- [17] Alice LUCAS et al. “Generative Adversarial Networks and Perceptual Losses for Video Super-Resolution”. In : *IEEE Transactions on Image Processing* 28.7 (juill. 2019). arXiv :1806.05764 [cs], p. 3312-3327. ISSN : 1057-7149, 1941-0042. DOI : 10.1109/TIP.2019.2895768. URL : <http://arxiv.org/abs/1806.05764>.
- [18] S. Mostafa MOUSAVI et al. “Stanford EArthquake Dataset (STEAD) : A Global Data Set of Seismic Signals for AI”. In : *IEEE Access* 7 (2019). Conference Name : IEEE Access, p. 179464-179476. ISSN : 2169-3536. DOI : 10.1109/ACCESS.2019.2947848.
- [19] Aaron van den OORD et al. *WaveNet : A Generative Model for Raw Audio*. arXiv :1609.03499 [cs] version : 2. Sept. 2016. URL : <http://arxiv.org/abs/1609.03499>.
- [20] G. RONCORONI et al. “Synthetic seismic data generation with deep learning”. In : *Journal of Applied Geophysics* 190 (juill. 2021), p. 104347. ISSN : 0926-9851. DOI : 10.1016/j.jappgeo.2021.104347. URL : <https://www.sciencedirect.com/science/article/pii/S092698512100094X>.
- [21] Olaf RONNEBERGER, Philipp FISCHER et Thomas BROX. *U-Net : Convolutional Networks for Biomedical Image Segmentation*. arXiv :1505.04597 [cs]. Mai 2015. URL : <http://arxiv.org/abs/1505.04597>.
- [22] Chitwan SAHARIA et al. *Image Super-Resolution via Iterative Refinement*. arXiv :2104.07636 [cs, eess]. Juin 2021. URL : <http://arxiv.org/abs/2104.07636>.
- [23] Flavio SCHNEIDER. *ArchiSound : Audio Generation with Diffusion*. arXiv :2301.13267 [cs, eess] version : 1. Jan. 2023. URL : <http://arxiv.org/abs/2301.13267>.
- [24] Jiaming SONG, Chenlin MENG et Stefano ERMON. *Denoising Diffusion Implicit Models*. arXiv :2010.02502 [cs]. Oct. 2022. DOI : 10.48550/arXiv.2010.02502. URL : <http://arxiv.org/abs/2010.02502>.
- [25] Hoang THANH-TUNG et Truyen TRAN. *On Catastrophic Forgetting and Mode Collapse in Generative Adversarial Networks*. arXiv :1807.04015 [cs, stat]. Mars 2020. URL : <http://arxiv.org/abs/1807.04015>.
- [26] Daniele TRAPPOLINI et al. “DiffSD : Diffusion models for seismic denoising”. In : *EGU General Assembly 2023, Vienna, Austria, 24–28 Apr 2023, EGU23-13811*. 13811. Vienna, Austria, 2023. DOI : <https://doi.org/10.5194/egusphere-egu23-13811>.
- [27] Ashish VASWANI et al. *Attention Is All You Need*. arXiv :1706.03762 [cs]. Août 2023. URL : <http://arxiv.org/abs/1706.03762>.