Contributions to generative modeling and dictionary learning : theory and application

Michaël Allouche

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2022IPPAX116

Thèse de doctorat

Contributions to Generative Modeling and Dictionary Learning: Theory and Application

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'École Polytechnique

École doctorale n°574 École Doctorale de Mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques appliquées

Thèse présentée et soutenue à Palaiseau, le Vendredi 9 décembre 2022, par

MICHAËL ALLOUCHE

Composition du Jury :

Marylou Gabrié Professeure assistante, Ecole Polytechique (CMAP)	Examinatrice
Stéphane Girard Directeur de recherche, INRIA Grenoble (Statify)	Co-directeur de thèse
Emmanuel Gobet Professeur, École Polytechnique (CMAP)	Directeur de thèse
Blanka Horvath Professeure, University of Oxford	Présidente du jury
Ralf Korn Professeur, Techinical University of Kaiserslautern	Rapporteur
Chen Zhou Professeur, Erasmus University Rotterdam	Rapporteur

Remerciements

Je tiens tout d'abord à remercier très chaleureusement Stéphane et Emmanuel de m'avoir formé, soutenu et accompagné tout au long de cette thèse avec la même pédagogie, disponibilité et bienveillance à mon égard. Vous m'avez partagé votre rigueur et votre passion pour la recherche qui serviront, je l'espère, à continuer de m'améliorer et de travailler à vos côtés.

Merci Stéphane pour ton accueil à Grenoble, ton humour mordant, les apéritifs ping-pong et les debriefs sur les actualités de nos passions communes : le sport, le club de la capitale et la sncf.

Merci Emmanuel pour ta générosité, ton énergie stimulante, et pour ces nombreux moments de partage à discuter, à rigoler et surtout à danser !

Je remercie mes rapporteurs Chen Zhou et Ralf Korn pour leur temps et pour l'intérêt qu'ils ont porté à mes travaux. Je remercie également Blanka Horvath et Marylou Gabrié pour l'honneur qu'elles m'ont fait d'être dans mon jury de thèse.

Cette aventure n'aurait jamais été possible sans la formidable rencontre de Pascal Poupelle, ni le soutien invétéré de mon oncle Bernard Cohen qui m'a transmis tant de savoir et de valeurs depuis mon enfance. Je souhaite leur adresser ma plus grande gratitude et affection.

Je remercie tous les permanents avec qui j'ai eu l'occasion d'échanger ou de travailler et qui ont contribué à mon épanouissement en tant que doctorant. Je pense notamment à Josselin Garnier, Stefano De Marco, Clément Rey, Zoltán Szabó, Stéphane Crépey, Aurélien Alfonsi, Julyan Arbel et Jonathan El Methni.

Je remercie Marine Saux pour sa présence au quotidien, son organisation hors pair de tant d'événements et son rôle pilier au sein du laboratoire.

Je remercie l'équipe administration du CMAP, et notamment Nasséra, Alex et Guillaume pour leur disponibilité et leur soutien à tout moment. Merci aux ingénieurs IT Pierre, Sylvain et Laurent pour ces échanges passionnants et pour vos conseils avisés.

Je remercie toute l'équipe de la Chaire Stress Test pour le financement de ma recherche et pour la confiance qu'ils m'ont témoignée au travers de nombreux projets et séminaires.

Je salue naturellement l'équipe EG qui s'est composée notamment pendant ma thèse de Linda, Clara, Wanqing, Célia, Manon, Elisa, Dorinel, Charu, et sans oublier David mon acolyte de Cholesky. Je salue également les membres de mon bureau, Claire, Eugénie, Solange, Corentin, Dominik, Ignacio et Pierre, ainsi que tous les doctorants du CMAP avec qui j'ai malheureusement trop peu échangé dû à un décalage horaire sur le départ au Magnan.

Je tiens particulièrement à remercier très chaleureusement Cyril Bénézet et Florian Bourgey de m'avoir pris sous leur aile et encouragé tout au long de ma thèse. Vous avez été à mes yeux une source d'inspiration et des modèles à suivre. Je suis très heureux de vous avoir à mes côtés et de partager une si belle amitié.

Je remercie mes entraîneurs et amis de Taekwondo Rida Rachdi et David Vazquez de m'avoir forgé physiquement et mentalement afin de venir à bout de ce combat contre moi-même. Je salue

également tous les adhérents avec qui j'ai eu le plaisir d'échanger des vibrations de plastrons depuis tant d'années.

Je remercie Hugo et mes bandes d'amis du lycée, de yallah et d'itc, que j'ai quelque peu délaissés pour achever cette thèse mais leurs attentions et leurs encouragements m'ont accompagné tout au long de ces années. Merci Lola et Laura pour vos conseils d'anciennes doctorantes aguerries.

Je remercie ma famille, mes oncles, tantes, cousins d'avoir suivi mon parcours avec attention et de m'avoir encouragé à chaque nouveau challenge que je leur présentais. Merci Jojo pour ces soirées "gestite et parite" sur le toit de Lassalle. Merci David et Alex pour ces côtes de bœuf ressourçantes à Barcelone. Merci à ma grand-mère qui a toujours gardé un œil avisé sur mon parcours scolaire. Je remercie également ma belle-famille de m'avoir accueilli avec tendresse, générosité et humour. Merci Eden pour ton énergie débordante et tes fervents encouragements qui m'ont permis de ne jamais "avoir peur" ! Rendez-vous sur la Cannebière fin juin après l'ECN.

Je remercie mes parents de m'avoir transmis leur force de travail, de persévérance et de résilience. Merci Jérémie pour tous ces moments de complicité partagés de Dunois à Carmia en passant par Mouans-Sartoux, Gattone et Nagrand.

Enfin, je remercie ma femme Galith qui a partagé quotidiennement cette aventure à mes côtés avec attention, ferveur et amour. Je lui dédie ce travail qui n'aurait jamais vu le jour sans elle.

Préambule

Cette thèse vise à étudier les méthodes basées sur les données dans les paradigmes de l'Intelligence Artificielle et du *Machine Learning*. Bien que très populaires, ces méthodes sont principalement utilisées dans des travaux empiriques. Par conséquent, fournir des directives théoriques pour la construction de tels modèles est d'une importance primordiale.

Dans la première partie, nous étudions la modélisation générative à l'aide de réseaux de neurones dans deux contextes différents : la simulation d'un mouvement Brownien fractionnaire et de lois à queue lourde dans les cas conditionnels et non conditionnels. Dans tous les travaux, nous analysons la vitesse de convergence de l'erreur uniforme entre la fonction d'intérêt et son approximation par réseaux de neurones. Les performances de nos modèles sont illustrées au travers de simulations et de problèmes réels en finance et en météorologie : génération de rendements négatifs extrêmes d'indices financiers et de précipitations en fonction de leur localisation géographique.

Dans la deuxième partie, nous proposons une nouvelle méthode basée sur l'apprentissage par dictionnaire pour la modélisation des matrices de migration des notations financières. Nous devons faire face à une faible quantité de données, proche de la dimension du problème, une évolution rapide dans le temps des matrices et une collection de contraintes linéaires. Nous présentons une étude numérique avec des données réelles et montrons la performance du modèle à la fois comme indicateur de sentiment économique et comme alternative au modèle de Copule Gaussienne largement utilisé.

Preamble

This thesis aims at investigating data-based methods in the paradigms of Artificial Intelligence and Machine Learning. Although very popular, those methods are mainly used in empirical works. Therefore, providing theoretical guidelines for building such models is of primal importance.

In the first part, we study generative modeling using neural networks in two different settings: the simulation of a fractional Brownian motion, and of heavy-tailed distributions in both conditional and non-conditional cases. In all works, we analyze the convergence rate of the uniform error between the function of interest and its neural network approximation. The performance of our models are illustrated on simulations and real practical problems in finance and in meteorology: generating extreme negative returns of financial indexes and rainfalls as functions of their geographical location.

In the second part, we propose a new method based on dictionary learning for modeling financial rating migration matrices. We have to deal with small amount of data, close to the dimension of the problem, a fast evolution in time of the matrices and a collection of linear constraints. We present a numerical test with real data and show the performance of the model as both an economic sentiment indicator and an alternative to the widely used Gaussian Copula model.

List of contributions

Here is a list of articles (accepted or submitted) that were written during this thesis:

- [4]: M. Allouche, J. El Methni and S. Girard. A refined Weissman estimator for extreme quantiles. *Extremes*, to appear, 2022. Source code
- [5]: M. Allouche, S. Girard and E. Gobet. Estimation of extreme quantiles from heavy-tailed distributions with neural networks. *Submitted paper*, 2022. Source code
- [6]: M. Allouche, S. Girard and E. Gobet. EV-GAN: Simulation of extreme events with ReLU neural networks. *J. Mach. Learn. Res.*, 23(150):1–39, 2022.
- [7]: M. Allouche, S. Girard and E. Gobet. A generative model for fBm with deep ReLU neural networks. *J. Complexity*, page 101667, 2022.
- [8]: M. Allouche, E. Gobet, C. Lage and E. Mangin. Structured dictionary learning rating of migration matrices for credit risk modeling. *Submitted paper*, 2022. Source code

Keywords

Generative model, Neural networks, Extreme-value theory, Heavy-tailed distribution, Quantile estimation, Conditional quantile estimation, fractional Brownian motion, Dictionary Learning.

Plan

The manuscript starts with a general introduction containing the motivations, the state of the art and the problem statements of our research topics. Then, we present our contributions in each of the four chapters. Finally, we conclude by opening perspectives on further works.

Contents

Introduction (en français)

Le premier chapitre est consacré à la contextualisation des concepts et des défis abordés tout au long de cette thèse, composée de contributions dans deux domaines principaux : la modélisation générative et l'apprentissage par dictionnaire.

0.1 Modélisation générative

0.1.1 Contexte

Motivations

Un modèle génératif vise à imiter la loi d'un objet (éventuellement en grande dimension), communément appelé jumeau numérique en génie industriel, voir [121, 144] pour une revue. De tels modèles sont particulièrement utiles, par exemple dans le domaine de la *data-augmentation* : enrichir un ensemble de données, ce qui permet de réduire le surapprentissage (*overfitting*) et d'améliorer les performances des modèles statistiques. Une autre perspective d'intérêt majeur est celle de la confidentialité des données : partager des données générées qui ont les mêmes propriétés statistiques que certaines données confidentielles. Dans le contexte de la modélisation générative, nous distinguons deux points de vue différents. D'une part, l'échantillonnage de mouvements complexes d'objets physiques a été réalisé à l'origine en résolvant des équations mathématiques décrivant le comportement de l'objet sous contraintes et en générant des trajectoires selon différentes conditions initiales. Une telle approche nécessite d'abord de connaître la formule exacte ou de construire à la main le modèle mathématique. La résolution et la construction d'un tel modèle peuvent être très difficiles mais ont montré leur efficacité dans de nombreux domaines. D'autre part, une nouvelle classe de modèles génératifs basés sur les données a récemment émergé dans le paradigme de l'intelligence artificielle (IA). Au lieu de chercher le modèle physique, on peut essayer de l'apprendre directement à partir des données en utilisant du bruit aléatoire comme entrée. De tels algorithmes ont l'avantage d'être lents dans la phase d'inférence et rapides dans la phase de simulation par rapport à leurs homologues les modèles physiques. Grâce à l'évolution numérique et théorique du XXIe siècle, les réseaux de neurones (ou NNs de l'anglais *Neural Networks*) se sont révélés être d'excellents candidats comme fonction d'approximation universelle. Parmi les modèles génératifs de NN développés [71], les plus populaires ont été l'auto-encoder variationnel (ou VAE de l'anglais *Variational Autoencoder*) [115] basé sur l'inférence variationnelle, et les réseaux antagonistes génératifs (ou GANs de l'anglais *Generative Adversarial Networks*) [100], sur lesquels nous nous sommes concentrés, basés sur un jeu minmax. Des modèles plus récents comme les *Normalizing Flows* [161] et les *Diffusion Models* [169] ont gagné en popularité.

De nos jours, la construction et l'optimisation de lourds modèles de NNs sont facilitées par des bibliothèques en libre accès (*e.g.* TensorFlow [1] ou PyTorch [151]), ce qui suscite un intérêt remarquable venant de personnes issues de communautés et de formations mathématiques différentes. Contrairement à de nombreux travaux de modélisation générative développés dans la communauté de l'IA, nous nous sommes efforcés de proposer des bornes d'erreur sur l'approximation uniforme grâce à des informations structurelles ajoutées à nos modèles sur la base d'une analyse

théorique rigoureuse. Nous nous sommes également concentrés sur les vitesses de convergence qui permettent de donner des directives théoriques pour construire un NN afin d'atteindre une erreur fixée. Cette construction des NNs est une question très difficile et souvent réduite dans la littérature à une recherche empirique (*grid search*).

Nous avons contribué à deux sujets majeurs. Premièrement, nous avons étudié la simulation de séries temporelles continues (objet en dimension infinie), en particulier dans un cas non-Markovien, qui n'a pas encore été étudié dans la littérature des NNs génératifs (voir le Chapitre 1). Deuxièmement, nous avons abordé la question de la simulation des valeurs extrêmes à partir de lois à queue lourde, ce qui est impossible dans le cadre classique des NNs génératifs (voir les Chapitres 2 et 3).

Ci-dessous, nous rappelons d'abord le cadre statistique de la modélisation générative. Par la suite, nous exposons l'état de l'art sur les résultats d'approximation des NNs et discutons des résultats théoriques existants dans la littérature sur les GANs. De plus, nous soulevons quelques défis importants pour la simulation de séries temporelles continues (Section 0.1.3) et de lois à queue lourde (Section 0.1.4). Enfin, nous résumons nos contributions (Section 0.1.5).

0.1.2 Cadre statistique

Soit X une variable aléatoire prenant des valeurs dans un espace métrique général $(\mathcal{X}, d_{\mathcal{X}})$ et soit $\mathcal{P}(\mathcal{X})$ l'espace de toutes les mesures de probabilité définies sur la tribu de Borel $\mathcal{B}_{\mathcal{X}}$. De plus, supposons que toutes les probabilités considérées sont dominées par une mesure de référence fixe et connue (*e.g.* Lebesgue) $\pi \in \mathcal{P}(\mathcal{X})$. Ainsi, étant données des observations $\{X_i \in \mathcal{X}\}_{i=1}^n$ tirées selon une loi de densité inconnue p_X sur $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$, l'objectif de la modélisation générative est de trouver une fonction $G : \mathcal{Z} \to \mathcal{X}$, appelée générateur, et une distribution de probabilité p_Z, appelée distribution latente, telle que

$$G(Z) \overset{d}{=} X, \quad Z \sim p_Z. \tag{0.1.1}$$

Question. Quelle classe de fonctions G et de densités p_Z peut-on considérer pour s'assurer que (0.1.1) soit vérifiée ?

La réponse est fournie par le théorème de Kuratowski suivant ([19, Proposition 7.15, p. 121]), également appelé isomorphisme mesurable ([178, p.7]), dans le cas où $\mathcal{Z}$ et $\mathcal{X}$ sont des espaces Polonais et G est une bijection mesurable.

Theorem 0.1.1 (Kuratowski)**.** *Soient $(\mathcal{Z}, \mu_Z)$ et $(\mathcal{X}, \mu_X)$ deux espaces de probabilité Polonais. Alors il existe une bijection mesurable (non unique) $G : \mathcal{Z} \to \mathcal{X}$ telle que*

$$G_{\#}\mu_Z = \mu_X, \quad G^{-1}_{\#}\mu_X = \mu_Z,$$

où $G_{\#}\mu_Z(E) := \mu_Z(G^{-1}(E)), \forall E \subset \mathcal{X}$, représente la poussée en avant (push-forward) de μ_Z par G.

Nous nous concentrons ici sur une famille paramétrique de générateurs $\mathcal{G} := \{G_\theta\}_{\theta\in\Theta}$ et nous désignons par $\mathcal{P} := \{p_\theta\}_{\theta\in\Theta}$ leurs densités paramétriques associées telles que $G_\theta(Z) \overset{d}{=} p_\theta \, \mathrm{d}\pi$. Ainsi, le problème se résume à trouver les meilleurs paramètres $\theta^\star$ tels que $p_{\theta^\star}$ et p_X soient aussi proches que possible, ou de manière équivalente

$$G_{\theta^\star}(Z) \overset{d}{\approx} X, \quad Z \sim p_Z. \tag{0.1.2}$$

Un problème de modélisation générative consiste principalement à choisir trois ingrédients :

1. la paramétrisation de G_θ et la distribution latente p_Z à utiliser comme entrées,

2. les observations $X_1, \dots, X_n$ avec leurs propriétés sous-jacentes,
3. le processus d'optimisation et la distance entre les densités de probabilité p_θ et p_X.

Dans cette thèse, nous abordons les deux premiers points, tandis que le dernier sera mentionné dans la suite lors de la présentation du modèle GAN. Dans le nouveau paradigme de l'IA, il est naturel de considérer le NN comme une paramétrisation G_θ.

Réseaux de neurones

Un NN est une fonction non linéaire construite avec un nombre fixe de neurones, chacun représentant une fonction, et répartis sur plusieurs couches cachées. Les neurones sont mis à l'échelle et translatés dans le réseau par des paramètres appelés respectivement poids et biais. Parmi les nombreuses architectures de réseau existantes, considérons le classique NN à propagation avant (*feedforward*) à une couche cachée (Figure 1) $G_{\theta_K} : \mathbb{R}^{d'} \to \mathbb{R}$ composée de K neurones tel que

$$G_{\theta_K}(\mathbf{z}) = b^{(2)} + \sum_{k=1}^{K} w_k^{(2)} \sigma\left(\left\langle \mathbf{w}_k^{(1)}, \mathbf{z} \right\rangle + b_k^{(1)}\right), \tag{0.1.3}$$

avec comme paramètres $\theta_K = \left\{\mathbf{w}_k^{(1)}, w_k^{(2)}, b_k^{(1)}, b_2\right\}_{k=1}^{K} \in \Theta_K, \mathbf{z} \in \mathbb{R}^{d'}, \langle \cdot, \cdot \rangle$ un produit scalaire dans $\mathbb{R}^{d'}$, et σ la fonction non linéaire appelée fonction d'activation. Cette dernière fait partie d'une classe plus large appelée fonctions de Ridge [153] définie comme

$$g(a_1 z_1 + \cdots + a_{d'} z_{d'}) = g(\langle \mathbf{a}, \mathbf{z} \rangle),$$

avec $g : \mathbb{R} \to \mathbb{R}$, $\mathbf{a} \in \mathbb{R}^{d'}$ et varie uniquement dans une direction donnée par $\mathbf{a}$. Des exemples courants de fonctions d'activation dans la littérature sont

- *cosine squasher*

$$\sigma(x) = \frac{\cos(x + \frac{3\pi}{2}) + 1}{2} \mathbb{1}_{x \in [-\frac{\pi}{2}, \frac{\pi}{2}]} + \mathbb{1}_{x \in (\frac{\pi}{2}, \infty)}, \tag{0.1.4}$$

- *logistic squasher*

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}},$$

- *sigmoidal squashing function*: toute fonction croissante telle que

$$\sigma(x) = \begin{cases} 1, & x \to \infty \\ 0, & x \to -\infty \end{cases}, \tag{0.1.5}$$

- *exponential Linear Unit* (eLU)

$$\sigma_\alpha(x) = \begin{cases} \alpha(\exp(x) - 1) & , \quad x < 0 \\ x, & , \quad x \geq 0, \end{cases}$$

- *Rectifier Linear Unit* (ReLU)

$$\sigma(x) = \max(x, 0). \tag{0.1.6}$$

À la fin du XXe siècle, plusieurs auteurs ont étudié la capacité de $\mathbf{z} = (z_1, \dots, z_{d'}) \in \mathbb{R}^{d'} \mapsto G_{\theta_K}(\mathbf{z})$ à approcher une fonction donnée G dans différentes normes quand $K \to \infty$, si θ_K est bien choisi. De tels résultats sont regroupés dans ce que l'on appelle le théorème d'approximation universel.

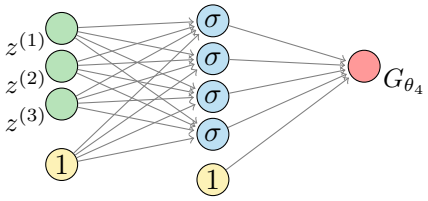


Figure 1: NN à une couche cachée avec $K = 4$ neurones et $d' = 3$.

Théorème d'approximation universel. Rappelons quelques-uns des principaux résultats basés sur l'article de référence [152].

▷ (Gallant, White - 1988)[73] : Si G est une fonction de carré intégrable sur $[0, 2\pi]^{d'}$ alors il existe un "NN de Fourier" G_{θ_K} composé par les fonctions d'activation *cosinus squasher* σ (0.1.4) qui converge uniformément vers G.

▷ (Cybenko - 1989)[46] : Si G est une fonction continue dans le cube unitaire $[0, 1]^{d'}$ de dimension d', alors il existe un NN G_{θ_K} composé de fonctions d'activation sigmoïdales (0.1.5) qui converge uniformément vers G.

▷ (Hornik - 1990)[110] : Résultat similaire prouvé pour toute fonction continue G, notée $G \in \mathcal{C}^0$, sur un ensemble compact avec une fonction d'activation bornée et non constante σ (plus général que (0.1.5) mais n'inclut pas (0.1.6)). Un autre résultat est valable pour toute fonction $G \in \mathbf{L}^p(\mathbb{R}^{d'}, \mathrm{d}\mathbf{z})$.

▷ (Pinkus - 1993)[126] : Résultat similaire prouvé si et seulement si σ n'est pas un polynôme (ce qui inclut (0.1.6)).

Voir le résultat général :

Theorem 0.1.2 (approximation universelle)**.** *Supposons que G soit une fonction continue sur un espace compact $\mathcal{Z} \subset \mathbb{R}^{d'}$ et que σ ne soit pas un polynôme, alors $\forall \varepsilon > 0$, il existe un NN (0.1.3) tel que*

$$\sup_{\mathbf{z} \in \mathcal{Z}} \left| G(\mathbf{z}) - \sum_{k=1}^{K} w_k^{(2)} \sigma\left(\left\langle \mathbf{w}_k^{(1)}, \mathbf{z} \right\rangle + b_k \right) \right| < \varepsilon.$$

Nous venons de voir que les Théorèmes 0.1.1 et 0.1.2 donnent des conditions suffisantes sur $\mathcal{Z}, \mathcal{X}$ et σ pour que (0.1.2) soit vérifiée si G_θ est un NN. En d'autres termes, si G n'est pas une fonction continue avec un support compact, alors elle ne sera en aucun cas uniformément bien approximée par un NN à une couche cachée. Nous aborderons ce problème dans les Chapitres 2 et 3.

Puisque notre objectif est de fournir des résultats théoriques sur les bornes d'erreur d'approximation, nous abordons la question suivante :

Question. Si les hypothèses du Théorème 0.1.2 sont vérifiées, quelle est la vitesse de convergence de G_{θ_K} vers G pour une norme donnée ?

Vitesse de convergence. Comme le souligne habituellement la théorie de l'approximation, la vitesse de convergence dépend de la régularité de la fonction G. Ici, nous considérons la régularité au sens de sa dérivabilité.

Definition 0.1.1 (Espace de Sobolev)**.** Soit $B^{d'}$ la boule unitaire dans $\mathbb{R}^{d'}$ telle que $B^{d'} = \left\{\mathbf{z} \in \mathbb{R}^{d'} : \|\mathbf{z}\|_2 \leq 1\right\}$. Définissons l'espace de Sobolev $\mathcal{S}_p^m = \mathcal{S}_p^m\left(B^{d'}\right)$ contenant toutes les fonctions continues $G \in \mathbb{R}^{d'}$ sur $B^{d'}$ et où les dérivées $D^k G$ existent et sont continues sur $B^{d'}$, pour tout $|k| = \{0, \ldots, m\}$. avec une norme définie par

$$\|G\|_{m,p} := \begin{cases} \left(\sum_{0\leq|k|\leq m} \left\|D^k G\right\|_p^p\right)^{1/p}, & 1 \leq p < \infty \\ \max_{0\leq|k|\leq m} \left\|D^k G\right\|_\infty, & p = \infty. \end{cases}$$

Le résultat suivant est basé sur les bornes obtenues dans $\mathbf{L^2}$ ([131, Théorème 1]) et sur l'extension de la borne supérieure pour toutes les normes $\mathbf{L}^p$ avec $1 \leq p \leq \infty$ ([130, Proposition 1] et [152, Corollaire 6.4]) en utilisant une fonction d'activation sigmoïde (0.1.5).

Theorem 0.1.3 (Vitesse de convergence)**.** *Soit* $S_p^m = S_p^m(B^{d'}) = \left\{G \in \mathcal{S}_p^m : \|G\|_{m,p} \leq 1\right\}$. *Pour chaque* $p \in [1, \infty]$ *et pour tout* $d' \geq 2$ *et* $m \geq 1$*, il existe une fonction d'activation* σ *infiniment dérivable sur* $\mathbb{R}$*, sigmoïde et strictement croissante telle que*

$$\sup_{G\in S_p^m} \inf_{\theta_K\in\Theta_K} \|G - G_{\theta_K}\|_p \leq CK^{-\frac{m}{(d'-1)}},$$

avec C *une constante indépendante de* K*.*

Le résultat ci-dessus montre que diviser l'erreur par 2 conduit à augmenter le nombre de neurones d'au moins $2^{\frac{(d'-1)}{m}}$. Ce phénomène bien connu est appelé la malédiction de la dimensionnalité. Cependant, il existe un moyen de surmonter cette malédiction (erreur $\mathbf{L^2}$ réduite à $C/\sqrt{K}$) en imposant que $\int \|\xi\| \left|\widehat{G}(\xi)\right| \mathrm{d}\xi < \infty$, où $\widehat{G}$ est la transformée de Fourier de G [15].

Réseaux de neurones multicouches. Au début du XXIe siècle, les auteurs ont commencé à s'intéresser à l'intérêt d'augmenter le nombre de couches cachées. Sur la base du théorème de superposition de Kolmogorov [107, 170], un premier résultat de convergence uniforme a été prouvé pour un NN à deux couches cachées [130, Théorème 4]. Quelques années plus tard, l'auteur dans [189] a établi une vitesse de convergence de l'erreur uniforme en utilisant un NN multicouche avec une fonction d'activation ReLU (0.1.6).

Theorem 0.1.4. *Pour tout* d', m *et* $\varepsilon \in [0, 1]$*, il existe un NN ReLU profond qui peut approximer uniformément toute fonction* $G \in S_p^m([0, 1]^{d'})$ *avec une erreur* ε *composé d'au moins*

- $C\left(\log(1/\varepsilon) + 1\right)$ *couches cachées,*
- $C\varepsilon^{-d'/m}\left(\log(1/\varepsilon) + 1\right)$ *neurones,*

avec une certaine constante $C = C(d', m)$*.*

Bien que la complexité en profondeur du résultat précédent ait été améliorée pour les fonctions continues par un terme logarithmique dans [190], le principal terme de complexité en largeur $\varepsilon^{-d'/m}$ reste similaire au cas d'une couche cachée.

Question Si G_θ est un NN et que les conditions du Théorème 0.1.2 sont satisfaites, comment entraîner θ de telle sorte que (0.1.2) soit vérifiée ?

Parmi les différentes méthodes de modélisation générative de NNs pour l'entraînement de θ, nous nous concentrons sur les GANs car ils ont la bonne structure qui nous convient pour fournir des résultats d'approximation.

Réseaux Antagonistes Génératif (GAN)

Proposé par [100], un schéma GAN vise à approximer l'inconnue G par une famille paramétrique de NNs $\mathcal{G} = \{G_\theta : \mathbb{R}^{d'} \to \mathbb{R}^d, \theta \in \Theta\}$ et à apprendre le paramètre optimal $\theta^\star$ à partir d'un ensemble de données $\{X_i \in \mathbb{R}^d, i \in \{1, \dots, n\}\}$ d'un échantillon i.i.d. provenant de la densité inconnue p_X. Cela est réalisé par l'optimisation d'une fonction objective qui peut être interprétée comme un jeu antagoniste entre un générateur et un discriminateur choisi dans une famille paramétrique de fonctions $\mathcal{D} = \{D_\phi : \mathbb{R}^d \to [0,1], \phi \in \Phi\}$. En d'autres termes, $D_\phi(x)$ représente la probabilité qu'une observation x soit tirée de p_X. Le générateur et le discriminateur sont tous deux des NNs aux objectifs opposés : le premier essaie d'imiter les données réelles qui semblent probables par le discriminateur, tandis que le second essaie de distinguer les deux sources. Voir la Figure 2 pour une illustration. Dans [100], ce problème d'optimisation est défini comme :

$$\mathcal{L}(\theta, D_\phi) := \min_{\theta \in \Theta} \max_{D_\phi \in \mathcal{D}} \left[\mathbb{E}_{p_X} \left(\log D_\phi(X) \right) + \mathbb{E}_{p_Z} \left(\log \left(1 - D_\phi \left(G_\theta(Z) \right) \right) \right) \right].$$

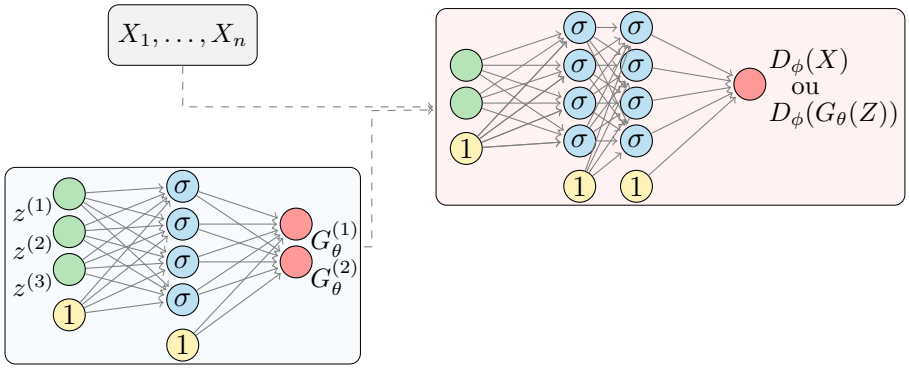


Figure 2: Modèle GAN avec $d' = 3$ et $d = 2$.

Résultats théoriques. Dans cette thèse, nous ne nous concentrons ni sur l'erreur statistique ni sur l'équilibre du jeu antagoniste entre le générateur et le discriminateur. Nous rappelons ici une esquisse des résultats existants, où plus de détails et les preuves complètes peuvent être trouvés dans [22]. Considérons une classe plus large $\mathcal{D}_\infty = \left\{D : \mathbb{R}^d \to [0,1]\right\}$, le discriminateur optimal étant donné tout générateur fixe G_θ est $D_\theta^\star = \frac{p_X}{p_X + p_\theta}$ [100, Proposition 1], ce qui implique [100, Théorème 1] et

$$\begin{aligned} \sup_{D \in \mathcal{D}_\infty} \mathcal{L}(\theta, D) = \mathcal{L}(\theta, D_\theta^\star) &= -\log(4) + 2\mathtt{D}_{\mathtt{JS}}\left(p_X || p_\theta\right) \\ &= -\log(4), \quad \text{si } p_X = p_\theta, \end{aligned}$$

où

$$\mathtt{D}_{\mathtt{JS}}\left(p_X || p_\theta\right) := \int_{\mathcal{X}} \frac{1}{2} p_X(x) \log\left(\frac{2p_X(x)}{p_X(x) + p_\theta(x)} \right) + \frac{1}{2} p_\theta(x) \log\left(\frac{2p_\theta(x)}{p_X(x) + p_\theta(x)} \right) \pi(\mathrm{d}x)$$

représente la divergence de Jensen-Shannon (JS). De plus, on sait que $D_\theta^\star = \underset{D \in \mathcal{D}_\infty}{\arg\max}\, \mathcal{L}(\theta, D)$ est unique si $\pi(p_X = p_{\theta^\star} = 0) = 0$ [22, Théorème 2.1]. À l'évidence, si $p_X \in \mathcal{P}$ alors $p_X = p_{\theta^\star}$ et $D_\theta^\star = 1/2$, *i.e.* le discriminateur optimal classe toutes les données avec une probabilité de $1/2$ car il n'est plus capable de distinguer les données réelles des données générées. Cependant, habituellement $p_X \notin \mathcal{P}$ à cause de la nature paramétrique restreinte de $\mathcal{G}$ et $\mathcal{D}$. De plus, sous des conditions de convexité et de compacité, [22, Théorème 2.2] montre l'existence et l'unicité du meilleur paramètre $\theta^\star$ approximant la densité inconnue sous la divergence JS :

$$\theta^\star = \underset{\theta \in \Theta}{\arg\min}\, \mathcal{L}(\theta, D_\theta^\star) = \underset{\theta \in \Theta}{\arg\min}\, \mathtt{D}_{\mathtt{JS}}\left(p_X || p_\theta\right).$$

En se limitant à la classe paramétrique $\mathcal{D}$ et en supposant que les nouveaux paramètres optimaux du générateur associé $\bar{\theta} \in \Theta$ existent tels que $\sup_{D_\phi \in \mathcal{D}} \mathcal{L}\left(\bar{\theta}, D_\phi\right) \leq \sup_{D_\phi \in \mathcal{D}} \mathcal{L}\left(\theta, D_\phi\right), \forall \theta \in \Theta$, alors [22, Théorème 3.1] stipule que si la classe $\mathcal{D}$ est suffisamment riche pour approximer le discriminateur $D_{\bar{\theta}}^{\star} = \arg\max_{D_\phi \in \mathcal{D}} \mathcal{L}(\bar{\theta}, D_\phi)$ dans $\mathbf{L^2}$ par une distance non supérieure à $\varepsilon > 0$, alors

$$\mathtt{D}_{\mathtt{JS}}(p_X || p_{\bar{\theta}}) - \mathtt{D}_{\mathtt{JS}}(p_X || p_{\theta^\star}) = \mathcal{O}\left(\varepsilon^2\right).$$

Ce résultat met en évidence l'argument selon lequel plus la classe de discriminateurs est riche (i.e. plus de neurones et de couches cachées), plus la densité restreinte $p_{\bar{\theta}}$ sera proche de la densité non restreinte $p_{\theta^\star}$. De plus, avec davantage d'hypothèses de régularité, [22, Théorème 4.1] donne une erreur statistique du meilleur paramètre empirique $\hat{\theta}$ comme

$$\mathbb{E}\mathtt{D}_{\mathtt{JS}}\left(p_X || p_{\hat{\theta}}\right) - \mathtt{D}_{\mathtt{JS}}\left(p_X || p_{\theta^\star}\right) = \mathcal{O}\left(\varepsilon^2 + \frac{1}{\sqrt{n}}\right).$$

Enfin, la convergence des meilleurs paramètres empiriques $\hat{\theta}$ et $\hat{\alpha}$ et la normalité asymptotique de $\hat{\theta}$ sont données respectivement dans [22, Théorème 4.2] et [22, Théorème 4.3]. Par la suite, d'autres architectures et distances ont été proposées, en particulier le célèbre Wasserstein GAN [10], dont les résultats théoriques peuvent être trouvés dans [24] et [103].

Nous sommes maintenant en mesure d'introduire les domaines d'étude considérés au cours de cette thèse : les séries temporelles continues (Section 0.1.3) et les extrêmes (Section 0.1.4).

0.1.3 Séries temporelles continues

Parmi l'immense littérature sur les GANs, de nombreux travaux ont étudié la capacité à générer des séries temporelles (en temps discret), que ce soit dans le domaine financier [186], médical [64] ou météorologique [103], pour ne citer que certains d'entre eux. Par conséquent, afin de fournir des directives quantitatives précises sur G_θ et p_Z dans le cas où X est un processus temporel continu, nous avons décidé de nous concentrer sur le mouvement Brownien fractionnaire (fBm) $\left\{B^H(t)\right\}_{t\in\mathbb{R}}$ avec un paramètre de Hurst $H \in (0,1)$, comprenant le mouvement Brownien standard ($H = 1/2$) comme cas particulier. Un fBm est un processus Gaussien centré avec une fonction de covariance

$$\mathbb{C}\mathrm{ov}\left(B^H(t), B^H(s)\right) = \frac{V_H}{2}\left(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\right), \text{ pour chaque } \quad s,t \geq 0,$$

avec $V_H = \mathbb{V}\mathrm{ar}[B^H(1)]$. La motivation du choix d'un tel modèle pour notre étude est triple. Premièrement, sa simulation stochastique est connue pour être assez délicate (au moins pour $H \neq 1/2$), surtout lorsque le nombre de points de temps devient de plus en plus grand – voir [39, 56] et [13, Section 11.6] pour une revue et [40, 120] pour les contributions récentes. Avoir à portée de main un modèle génératif pour la trajectoire complète est vraiment donc intéressant pour une utilisation pratique. Deuxièmement, il est largement utilisé dans diverses modélisations de la vie réelle : la posture uni et bipède en biomécanique [30], la volatilité des actifs financiers [43, 84], les structures de filaments de vortex observées dans des fluides 3D [70], le prix de l'électricité dans un marché libéré [17], le cycle solaire [154]. Pour d'autres modélisations d'ordre fractionnaire, voir [41]. Troisièmement, comprendre la bonne conception de G_θ pour générer la distribution de fBm pourrait bien ouvrir la voie au traitement de modèles stochastiques plus compliqués écrits comme une équation différentielle stochastique (EDS) dirigée par fBm par exemple.

Sachant que B^H est un processus Gaussien centré dans un espace de Banach $(\mathcal{C}^0([0,1],\mathbb{R}),\|.\|_\infty)$ (voir [124, Proposition 3.6]), B^H admet une représentation en série presque sûre (p.s.) de la forme

$$B^H(t) = \sum_{k=0}^{\infty} u_k(t) Z_k, \quad \forall t \in [0,1], \tag{0.1.7}$$

où $\{u_k\}_{k\in\mathbb{N}}$ est une séquence de fonctions continues non aléatoires, et $\{Z_k\}_{k\in\mathbb{N}}$ est une séquence de variables Gaussiennes standard indépendantes $\mathcal{N}(0,1)$. L'égalité (0.1.7) est vérifiée dans le sens où la série converge p.s. uniformément.

Question. Quelle classe de fonctions u_k peut-on considérer pour s'assurer que (0.1.7) soit vérifiée ?

Dans [142], les auteurs ont proposé plusieurs développements en séries d'ondelettes de fBm sous la forme de (0.1.7) où $\{u_k\}_{k\in\mathbb{N}}$ dépend a) du choix de la base d'ondelettes et b) de la description des termes de basse fréquence. Nous allons tirer parti de l'une des séries d'ondelettes, pour laquelle certains résultats d'optimalité ont été établis dans [14].

Défi. Il est clair qu'une opération de produit dans (0.1.7) est nécessaire entre les entrées $\{Z_k\}_{k\in\mathbb{N}}$ (*i.e.* l'espace latent dans le cadre d'un GAN) et les fonctions $\{u_k\}_{k\in\mathbb{N}}$. Comme une telle opération n'est pas réalisée naturellement dans un réseau à propagation avant, nous devrons étudier comment l'approximer en plus de l'approximation par NN des $\{u_k\}_{k\in\mathbb{N}}$. Un autre point d'attention est qu'une fois entraîné, le NN doit être performant pour tout temps $t \in [0,1]$ contrairement à d'autres méthodes discrètes de simulation connues de fBm [13, Section 11.6].

0.1.4 Extrêmes

Nous commençons par rappeler quelques prérequis en théorie des valeurs extrêmes. Dans le cadre d'une lecture plus exhaustive, on pourra se référer à [52] pour les aspects théoriques et à [42] pour les applications.

Théorie des valeurs extrêmes

Étant donné des variables aléatoires i.i.d. $X_1,\dots,X_n$ avec une fonction de répartition (c.d.f.) associée F et une fonction quantile $q(\cdot) := \inf\{x \in \mathbb{R} : F(x) \geq \cdot\}$, la théorie des valeurs extrêmes s'intéresse au comportement de la loi limite des maxima de l'échantillon $X_{n,n} = \max(X_1,\dots,X_n)$ quand $n \to \infty$. Il est clair que cette distribution peut être dérivée exactement pour tout n puisque

$$\mathbb{P}(X_{n,n} \leq x) = F^n(x).$$

Bien que F soit inconnue, l'idée est d'approximer F^n par une famille de distributions basée sur les plus grandes observations. Cependant, pour éviter les lois limites dégénérées, *i.e.* $F^n(x) \to 0$ pour $x < x^\star$ et $F^n(x) \to 1$ pour $x \geq x^\star$ quand $n \to \infty$ avec $x^\star = \sup\{x : F(x) < 1\}$ l'extrémité droite, une normalisation est nécessaire en supposant qu'il existe une séquence $a_n > 0$ (localisation) et $b_n \in \mathbb{R}$ (échelle) telles que la variable

$$\frac{X_{n,n} - b_n}{a_n}$$

a une loi limite non dégénérée, *i.e.* pour tous les $x \in \mathbb{R}$.

$$\lim_{n\to\infty} F^n(a_n x + b_n) = H(x). \tag{0.1.8}$$

Nous disons pour la classe de fonctions de répartitions F satisfaisant (0.3.8) que F appartient au domaine d'attraction maximale de H, notée en abrégé par $F \in \mathrm{MDA}(H)$.

Question. Quelle classe de fonctions H peut-on considérer pour s'assurer que (0.1.8) soit vérifiée ?
La réponse est fournie par le célèbre Théorème des valeurs extrêmes suivant [69, 89].

Theorem 0.1.5. *La classe des lois des valeurs extrêmes est $H_\gamma(ax+b)$ avec $a>0, b\in\mathbb{R}$ où*

$$H_\gamma(x) = \begin{cases} \exp\left(-(1+\gamma x)_+^{-1/\gamma}\right), & \gamma \neq 0, \\ \exp\left(-x\right), & \gamma = 0. \end{cases}$$

Le paramètre γ est appelé l'indice des valeurs extrêmes et nous distinguons trois sous-classes de distributions avec des comportements de queue différents :

- $\gamma > 0 : F \in$ MDA (Fréchet) [63, Section 3.3.1],
- $\gamma < 0 : F \in$ MDA (Weibull) [63, Section 3.3.2],
- $\gamma = 0 : F \in$ MDA (Gumbel) [63, Section 3.3.3].

La caractérisation des domaines d'attraction repose sur la théorie des fonctions à variation régulière [26].

Definition 0.1.2 (Variation régulière)**.** Une fonction mesurable $f : \mathbb{R}^+ \to \mathbb{R}^+$ est à variation régulière d'indice $\delta \in \mathbb{R}$ à l'infini si pour tous les $x > 0$.

$$\lim_{t\to\infty} \frac{f(xt)}{f(t)} = x^\delta.$$

Cette propriété est désignée par $f \in \mathcal{RV}_\delta$.

On peut montrer que toute fonction $f \in \mathcal{RV}_\delta$ peut s'écrire sous forme de

$$f(t) = t^\delta L(t),$$

où $L \in \mathcal{RV}_0$ est appelée une fonction à variation lente avec

$$\lim_{t\to\infty} \frac{L(xt)}{L(t)} = 1, \tag{0.1.9}$$

pour tout $x > 0$. Dans la suite, nous nous concentrerons sur le domaine d'attraction de Fréchet, auquel appartiennent les lois à queue lourde. Il est caractérisé par une décroissance polynomiale de la fonction de survie $\bar{F} := 1 - F$.

Theorem 0.1.6 ([63, Théorème 3.3.7])**.** *Une c.d.f. $F \in$ MDA(Fréchet) si et seulement si $\bar{F} \in \mathcal{RV}_{-1/\gamma}$ avec $\gamma > 0$.*

De manière équivalente, il est connu [52, Théorème 1.2.1, Proposition B.1.9.9] que la fonction quantile de queue $U(t) := q(1-1/t) \in \mathcal{RV}_\gamma$ pour tout $t > 1$, *i.e.*

$$U(t) = t^\gamma L(t), \tag{0.1.10}$$

où $L \in \mathcal{RV}_0$. Dans le contexte de la modélisation générative pour les lois à queue lourde, nous nous intéressons à l'estimation de (0.1.10) pour de grandes valeurs de t.

Estimation du quantile

Parmi les différentes approches de l'estimation par quantile extrême, nous nous concentrons sur l'approche semi-paramétrique basée sur (0.1.10). Cette dernière est composée d'une partie paramétrique t^γ qui ne dépend que de l'indice de queue $\gamma > 0$, et d'une partie non-paramétrique $L(t)$ où L satisfait (0.1.9). Nous nous intéressons à l'estimation de la fonction quantile (0.1.10), au niveau extrême $1-\alpha_n$ *i.e.* tel que $n\alpha_n \to 0$ quand $n \to \infty$. Cette dernière condition implique que $q(1-\alpha_n)$ est presque sûrement asymptotiquement plus grand que les maxima de l'échantillon. L'idée qui sous-tend l'estimation est de tirer parti de (0.1.10) pour établir le lien entre le quantile extrême d'intérêt $U(1/\alpha_n) = q(1-\alpha_n)$ et un quantile intermédiaire $U(1/\delta_n) = q(1-\delta_n)$ où δ_n est interprété comme un niveau d'ancrage avec $k := \lfloor n\delta_n \rfloor \to \infty$ quand $n \to \infty$ telle que

$$q(1-\alpha_n) = q(1-\delta_n)(\delta_n/\alpha_n)^\gamma \exp\left(\varphi\Big(\log(\delta_n/\alpha_n), \log(1/\delta_n)\Big)\right), \tag{0.1.11}$$

avec pour tout $(x_1, x_2) \in \mathbb{R}_+^2$,

$$\varphi(x_1, x_2) := \log\left(\frac{L(\exp(x_1+x_2))}{L(\exp(x_2))}\right). \tag{0.1.12}$$

Trois termes dans (0.1.11) doivent être estimés :

1. le quantile intermédiaire $q(1-\delta_n)$,
2. l'indice de queue γ,
3. la fonction φ.

L'estimateur le plus connu basé sur l'approche semi-paramétrique est l'estimateur de Weissman [184] qui consiste à choisir 1) la statistique d'ordre $X_{n-k+1,n}$, 2) l'estimateur de Hill $\hat\gamma(\cdot)$ [108] et 3) $\varphi(\cdot,\cdot) = 0$ tels que

$$\hat q(1-\alpha_n; 1-\delta_n) := X_{n-k+1,n}(\delta_n/\alpha_n)^{\hat\gamma(k)},$$

où

$$\hat\gamma(k) := \frac{1}{k}\sum_{i=1}^{k-1} \log X_{n-i+1,n} - \log X_{n-k+1,n}.$$

Au lieu de cela, (0.1.12) peut être évaluée à l'aide de la condition du second ordre qui stipule qu'il existe $\gamma > 0, \rho < 0$ et une fonction A positive ou négative avec $A(t) \to 0$ quand $t \to \infty$ tels que pour tout $x \geq 1$ [93, Equation (13)]

$$\log U(tx) - \log U(t) = \gamma \log x + A(t)\int_1^x y^{\rho-1}\,\mathrm{d}y + o(A(t)), \qquad \text{as } \to \infty. \tag{0.1.13}$$

De plus, $|A|$ est à variation régulière d'indice ρ. Ce paramètre du second ordre détermine le biais de la plupart des estimateurs de quantiles extrêmes : plus ρ est grand, plus le biais asymptotique est important. L'hypothèse (0.1.13) est standard dans la théorie des valeurs extrêmes, puisqu'elle contrôle la vitesse de convergence dans (0.1.12).

Theorem 0.1.7 ([52, Théorème 4.3.8])**.** *Supposons que*

- *la condition de second ordre (0.1.13) soit vérifiée,*
- $k \to \infty, \delta_n \to \infty$ *et* $\sqrt{k}A(1/\delta_n) \to \lambda < \infty$ *quand* $n \to \infty$,

- $n\alpha_n = o(k)$ *et* $\log(n\alpha_n) = o\left(\sqrt{k}\right)$ *quand* $n \to \infty$.

Alors, quand $n \to \infty$

$$\frac{\sqrt{k}}{\log(\delta_n/\alpha_n)}\big(\log \hat{q}(1-\alpha_n; 1-\delta_n) - \log q(1-\alpha_n)\big) \xrightarrow{d} \mathcal{N}\left(\frac{\lambda}{1-\rho}, \gamma^2\right).$$

À partir de la représentation asymptotique ci-dessus de l'estimateur de Weissman, des estimateurs débiaisés ont été introduits grâce à une estimation préalable de ρ conduisant la première composante dominante du biais, voir [94].

Défi. Une première observation est que, même dans le cas univarié ($d = d' = 1$), un NN à propagation avant (0.3.3), avec une fonction d'activation sigmoïdale (0.1.5) ou ReLU (0.1.6), ne peut pas approximer efficacement (0.1.10). Premièrement, la fonction quantile $u \in [0,1) \mapsto q(u)$ ne bénéficie pas d'un support compact puisqu'elle diverge lorsque $u \in 1$ à une vitesse $(1-u)^{-\gamma}$, voir la Figure 3 pour une illustration. Par conséquent, le théorème d'approximation universelle (Théorème 0.1.2) n'est pas vérifié. Deuxièmement, si Z est (comme habituellement fixé) soit bornés (Uniforme) soit de loi à queue légère (Gaussienne), $G_\theta(Z)$ serait respectivement bornée ou de loi à queue légère. À l'inverse, on cherche à générer une variable aléatoire non bornée avec une loi à queue lourde.

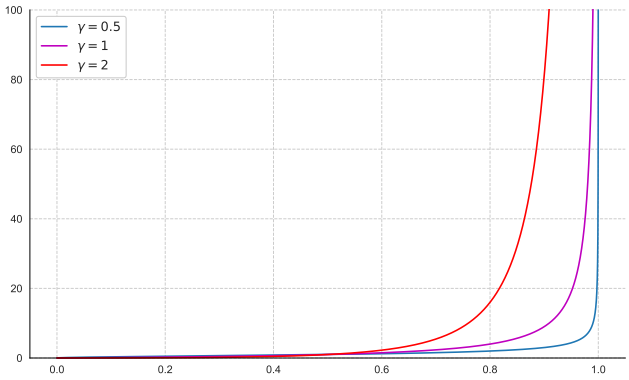


Figure 3: Fonction quantile associée à la loi de Burr $u \in [0,1) \mapsto q(u)$ avec indice de queue $\gamma \in \{1/2, 1, 2\}$ et $\rho = -1$.

0.1.5 Contributions de la thèse

Après avoir présenté le contexte et les domaines d'étude, nous sommes maintenant en mesure de préciser l'énoncé du problème principal de la partie modélisation générative de cette thèse.

Question. Comment choisir G_θ et l'espace latent $(\mathcal{Z}, \mu_Z)$ lorsque

- $\mathcal{X}$ est l'espace des fonctions continues indexées par le temps, muni de la norme sup $d_{\mathcal{X}}$, et lorsque la distribution de X est celle d'un processus stochastique (objet de dimension infinie), éventuellement non-Markovien ?
- X suit une loi à queue lourde (*i.e.* une décroissance polynomiale de la fonction de survie), et $G_\theta(Z)$ doit simuler des quantiles extrêmes (*i.e.* à un niveau de risque $1-\alpha_n$ tel que $n\alpha_n \to 0$ quand $n \to \infty$) ?
- X est conditionnée par une autre variable aléatoire Y ?

Contributions au chapitre 1. Nous abordons d'abord la question de la simulation d'une série temporelle continue avec un NN génératif. Nous fournissons une borne de probabilité large sur l'approximation uniforme de fBm avec un paramètre de Hurst $H \in (0,1)$, par un réseau profond ReLU à propagation avant alimenté par la distribution latente p_Z avec N variables aléatoires Gaussiennes standard indépendantes, avec des bornes sur l'architecture du réseau (nombre de couches cachées et nombre total de neurones). Nous commençons par le cas $H = 1/2$ (mouvement Brownien standard).

Theorem 0.1.8 (version simplifiée de [7, Théorème 2]). *Soit $N \geq 2$ et $\left(\Omega^N, \mathcal{F}^N, \mathbb{P}^N\right)$ un espace de probabilité supportant N variables aléatoires Gaussiennes standard i.i.d. $G_{1:N}$. Par conséquent, il existe une extension $(\Omega, \mathcal{F}, \mathbb{P})$ supportant un mouvement Brownien B tel que $\forall p \in (0,1]$, il existe un réseau de neurones ReLU $\tilde{B}_{N,p}$ et une variable aléatoire finie $C \geq 0$ (indépendante de N et p) tels que*

$$\mathbb{P}\left(\sup_{t\in[0,1]} \left|B(t) - \tilde{B}_{N,p}(t, G_{1:N})\right| \leq CN^{-1/2}\left(1+\log N\right)^{1/2}\right) \geq 1-p.$$

De plus, $\tilde{B}_{N,p}$ est composé au maximum de

1. $\mathcal{O}_c\left(\log\left(\frac{N\rho_N}{(1+\log N)^{1/2}}\right)\right)$ *couches cachées,*
2. $\mathcal{O}_c\left(N\log\left(\frac{N\rho_N}{(1+\log N)^{1/2}}\right)\right)$ *neurones et paramètres,*

avec $\rho_N = -\Phi^{-1}(\frac{p}{2N})$ et Φ^{-1} la fonction quantile de la loi Gaussienne standard.

Ensuite, nous traitons le cas général du fBm.

Theorem 0.1.9 (version simplifiée de [7, Théorème 3]). *Soit $N \geq 2$ et $\left(\Omega^N, \mathcal{F}^N, \mathbb{P}^N\right)$ un espace de probabilité supportant N variables aléatoires Gaussiennes standard i.i.d. $G_{1:N}$. Par conséquent, il existe une extension $(\Omega, \mathcal{F}, \mathbb{P})$ supportant un mouvement Brownien fractionnaire B^H tel que $\forall p \in (0,1]$, pour tout $r \in \mathbb{N}_0$ il existe un réseau de neurones ReLU $\tilde{B}^H_{N,p}$ et une variable aléatoire finie $C \geq 0$ (indépendante de N et p) tels que*

$$\mathbb{P}\left(\sup_{t\in[0,1]} \left|B^H(t) - \widetilde{B}^H_{N,p}(t, G_{1:N})\right| \leq CN^{-H}\left(1+\log(N)\right)^{1/2}\right) \geq 1-p.$$

De plus, $\widetilde{B}^H_{N,p}$ est composé de

1. $\mathcal{O}_c\left(\log\left(\frac{N\rho_N}{(1+\log(N))^{1/2}}\right)\right)$ *couches cachées,*
2. $\mathcal{O}_c\left(N^{1+\frac{H+1}{2r}}\log\left(\frac{N\rho_N}{(1+\log(N))^{1/2}}\right)\left(\frac{\rho_N}{(1+\log(N))^{1/2}}\right)^{\frac{1}{2r}}\right)$ *neurons and parameters,*

où ρ_N est défini dans le Théorème 0.1.8. Les constantes de $\mathcal{O}_c(\cdot)$ peuvent dépendre de r et de H.

Essentiellement, nos résultats indiquent que pour une dimension latente donnée N, il existe un $G_\theta \in \mathcal{G}$ tel que l'égalité (0.3.1) soit vérifiée avec une erreur $N^{-H}\left(1+\log(N)\right)^{1/2}$ en norme sup avec une probabilité $1-p$. De plus, en se concentrant sur les vitesses par rapport à $N \to +\infty$, la profondeur de G_θ est au plus de

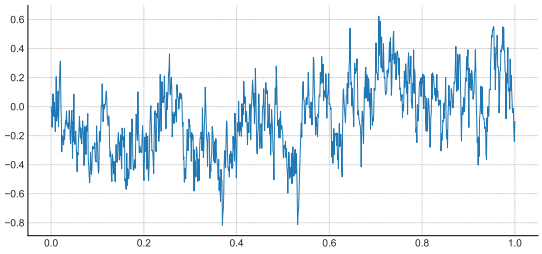
$$\mathcal{O}\left(\log N\right)$$

et sa complexité globale est de

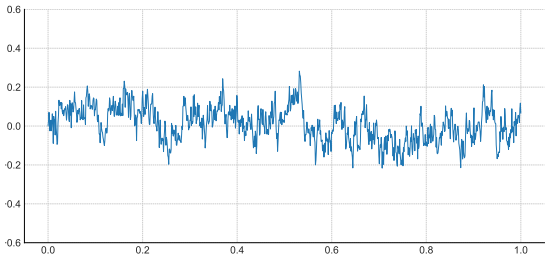
$$\mathcal{O}\left(N^{1+\zeta}\log N\right),$$

où ζ est un paramètre positif qui peut être pris aussi petit que souhaité, et où les $\mathcal{O}(\cdot)$ dépendent de p, ζ et H. En particulier pour le mouvement Brownien ($H = 1/2$) nous pouvons prendre $\zeta = 0$. Une dépendance plus détaillée de p, ζ et H est donnée plus loin. Notre analyse repose, dans le cas du mouvement Brownien standard ($H = 1/2$), sur la construction de Levy et dans le cas général du mouvement Brownien fractionnaire ($H \neq 1/2$), sur la représentation en ondelettes de Lemarié-Meyer. Cette dernière a été implémentée afin d'entraîner notre NN fBm $\widetilde{B}_N^H$ et est illustrée à la Figure 4, confirmant que la régularité des trajectoires et la structure de dépendance de notre NN sont bien préservées.

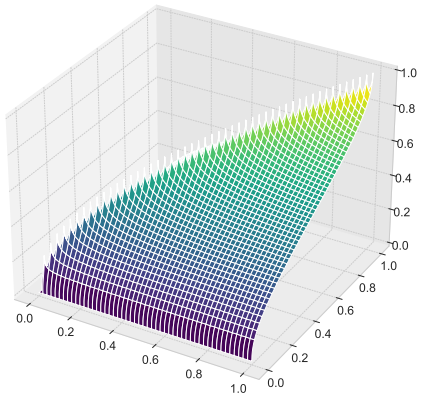
Ces résultats sont originaux à notre connaissance, et devraient jouer un rôle clé dans l'ajustement des méthodes basées sur les GANs concernant le choix de la famille paramétrique des NNs pour la génération de processus stochastiques fractionnaires en temps continu. De plus, ces résultats établissent un lien clair entre la régularité temporelle de la trajectoire (qui pourrait être mesurée sur les données réelles observées) et l'architecture de la paramétrisation à mettre en place.



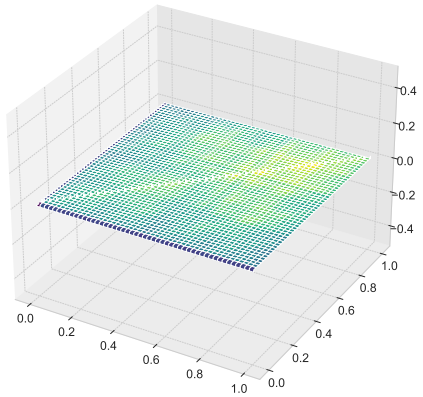
(a) $H = 0.2, \widehat{H} = 0.209$



(b) $\widetilde{H} = 0.208$



(c)



(d)

Figure 4: Simulation d'une trajectoire de fBm (en haut) et de la surface de covariance (en bas) pour $H = 0.2$.
En haut à gauche : Représentation en ondelettes $t \mapsto V_H^{-1/2} B_N^H(t)$ avec l'indice de Hurst estimé $\widehat{H}$.
En haut à droite : Erreur $t \mapsto V_H^{-1/2}(B_N^H(t) - \widetilde{B}_N^H(t))$ avec l'indice de Hurst estimé $\widetilde{H}$ sur le NN fBm $\widetilde{B}_N^H$.
En bas à gauche : fonction réelle normalisée $(t, s) \mapsto V_H^{-1}\mathbb{C}\mathrm{ov}(B^H(t), B^H(s))$.
En bas à droite : erreur $(t, s) \mapsto V_H^{-1}(\mathbb{C}\mathrm{ov}(B^H(t), B^H(s)) - \mathbb{C}\mathrm{ov}(\widetilde{B}_N^H(t), \widetilde{B}_N^H(s)))$ pour $(t, s) \in [0, 1]^2$.

Dans le contexte de la modélisation générative pour les lois à queue lourde, nous nous intéressons au comportement dans la queue des modèles génératifs NN. Nous venons d'en souligner les principaux enjeux, ce qui en fait un sujet difficile et peu traité théoriquement dans la littérature.

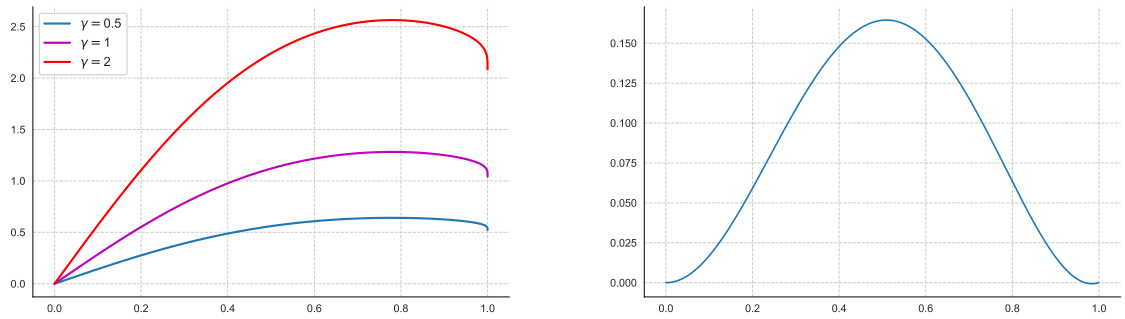
Contributions dans le chapitre 2. Nous proposons une nouvelle paramétrisation des GANs adaptée à l'apprentissage de lois à queue lourde dont le support est restreint à celui du jeu de données d'apprentissage. En tirant avantage de (0.1.10), nous introduisons une fonction d'indice de queue (TIF de l'anglais *Tail Index Function*)

$$u \mapsto f^{\mathrm{TIF}}(u) := -\frac{\log q(u)}{\log\left(\frac{1-u^2}{2}\right)}, \tag{0.1.14}$$

qui est continue, bornée sur $[0,1]$ et tend vers l'indice de queue γ lorsque $u \to 1$, voir Figure 5a. Bien qu'elle permette de respecter l'hypothèse de support compact requise par le Théorème d'approximation universelle, des termes de correction sont ajoutés dans (0.1.14) pour former une fonction d'indice de queue corrigée (CTIF de l'anglais *Corrected Tail Index Function*)

$$u \mapsto f^{\mathrm{CTIF}}(u) := f^{\mathrm{TIF}}(u) - \sum_{k=1}^{6} \kappa_k e_k(u),$$

illustrée à la Figure 5b et où les fonctions $e_1, \ldots, e_6$ et les coefficients $\kappa_1, \ldots, \kappa_6$ sont définis dans la Section 2.2.2. Une telle transformation permet d'améliorer la régularité de (0.1.14) et donc aussi la vitesse de convergence de l'erreur d'approximation par le NN.



(a) $\gamma \in \{1/2, 1, 2\}$ et $\rho = -1$ (b) $\gamma = 1/2$ et $\rho = -3$

Figure 5: Fonction Tail-index (a) et fonction Tail-index corrigée (b) associées à une loi de Burr.

Theorem 0.1.10 (version simplifiée de [6, Théorème 4])**.** *Supposons que les conditions du premier et du second ordre sont toutes deux vérifiées sous certaines hypothèses de régularité supplémentaires (voir la section 2.2.1). Soit σ une fonction ReLU. Pour tout $J \geq 6$, il existe $(a_j, w_j, b_j) \in \mathbb{R}^3$, $j \in \{1, \ldots, J\}$ tels que :*

$$\sup_{u \in [0,1]} \left| f^{\mathrm{CTIF}}(u) - \sum_{j=1}^{J} a_j \sigma\left(w_j u + b_j\right) \right| = \mathcal{O}\left(J^{-\alpha-1}\right),$$

où

1. $\alpha \in (0, -1-\rho)$ *si* $-2 \leq \rho < -1$,
2. $\alpha = 1$ *si* $\rho < -2$.

Notez que, pour $\alpha = 1$, la vitesse ci-dessus ne peut pas être améliorée en général, d'après [189, Théorème 6]. De plus, le résultat d'approximation précédent peut être interprété en termes de

distance de Wasserstein-1 entre la vraie distribution des données et la distribution simulée. En effet, dans le cas univarié, la distance de Wasserstein-1 peut être simplifiée comme suit

$$W_1(q, \widetilde{q}) = \int_0^1 |q(u) - \widetilde{q}(u)| \, \mathrm{d}u,$$

où $u \mapsto \widetilde{q}(u)$ est l'approximation EV-GAN de la fonction quantile inconnue $u \mapsto q(u)$.

Corollary 0.1.1 (version simplifiée de [6, Corollaire 5]). *Supposons que les conditions du Théorème 0.1.10 soient vérifiées avec $\gamma < 1$ et $\rho < -1$. Alors, la distance de Wasserstein-1 peut être contrôlée comme $W_1(q, \widetilde{q}) = \mathcal{O}\left(J^{-\alpha-1}\right)$.*

Notez que $\gamma < 1$ est une condition nécessaire à l'existence de la distance de Wasserstein-1. Les bornes d'approximation ci-dessus sur f^{CTIF} (Théorème 0.1.10) peuvent être traduites en termes de bornes d'approximation sur $f^{\mathrm{TIF},(m)}$ en utilisant un NN "enrichi" pour toutes les m-ième composantes d'une variable aléatoire de dimension d avec $m \in \{1, \ldots, d\}$ et une dimension latente $d' \geq d$.

Corollary 0.1.2 (version simplifiée de [6, Corollaire 6]). *Soit σ une fonction ReLU. Soit $X = (X^{(1)}, \ldots, X^{(d)})^\top$ un vecteur de d-dimension, chaque composante $X^{(m)}$ vérifiant les conditions du Théorème 0.1.10 avec des paramètres $(\gamma^{(m)}, \rho^{(m)})$. Soit $\mathcal{G}_J^{d',d}$ l'espace d'approximation des fonctions TIF constituées de $J \geq 6$ neurones. Alors ,*

$$\inf_{G \in \mathcal{G}_J^{d',d}} \sup_{m=1,\ldots,d} \sup_{z \in [0,1]^{d'}} \left| f^{\mathrm{TIF},(m)}(z^{(m)}) - G^{(m)}(z) \right| = \mathcal{O}\left(J^{-\alpha-1}\right),$$

où

(i) $\alpha \in (0, -1 - \max_{m \in \{1,\ldots,d\}} \rho^{(m)})$ *si* $-2 \leq \rho^{(m)} < -1$ *pour un certain* $m \in \{1, \ldots, d\}$,

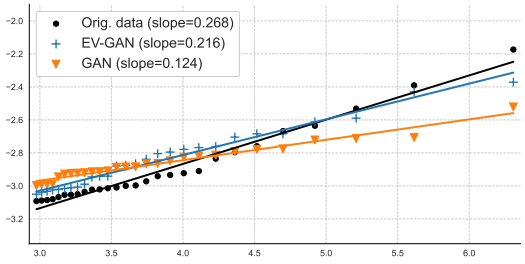
(ii) $\alpha = 1$ *si* $\rho^{(m)} < -2$ *pour tout* $m \in \{1, \ldots, d\}$.

Observons que le pire paramètre de second ordre $\rho^{(m)}$, *i.e.* le plus proche de -1, règle la précision globale de l'EV-GAN par l'ordre de convergence $\alpha + 1$. Donc, nous établissons que la vitesse de convergence de l'erreur est principalement déterminée par le paramètre du second ordre de la distribution des données. Les résultats sont illustrés à la fois sur des lois à queue lourde simulées et sur des indices boursiers financiers réels (voir Figure 6) en comparaison avec un modèle génératif NN classique. Il est démontré que, dans les deux expériences, notre approche surpasse largement la méthode classique.

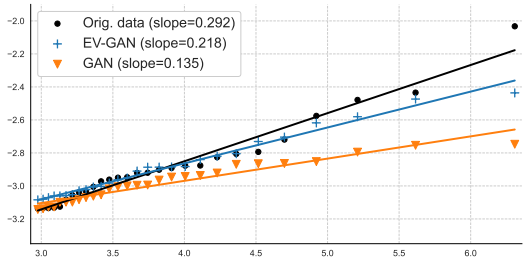
Contributions au chapitre 3. Nous étendons l'estimation de (0.1.10) a) en dehors du support de l'ensemble de données d'apprentissage et b) en prenant en compte une covariable associée au quantile extrême. Ces derniers correspondent respectivement à un estimateur non conditionnel et à un estimateur conditionnel des quantiles extrêmes par NN. Contrairement à la plupart des estimateurs débiaisés qui se concentrent uniquement sur la représentation du second ordre de (0.1.12), nous montrons que la condition de J-ième ordre ($J \geq 2$) bénéficie d'une représentation NN naturelle avec la fonction d'activation eLU qui est l'une des plus polpulaires et qui permet de mieux approximer la fonction d'espacement entre les logarithmes (*log-spacing*)

$$(x_1, x_2) \in \mathbb{R}_+^2 \mapsto \gamma x_1 + \varphi(x_1 + x_2), \tag{0.1.15}$$

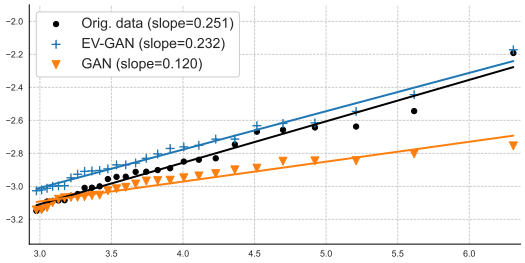
illustrée à la Figure 7. Sur la base de ce résultat, nous dérivons un estimateur de quantile extrême NN qui présente une estimation et une suppression automatiques de tous les J premiers termes de biais.



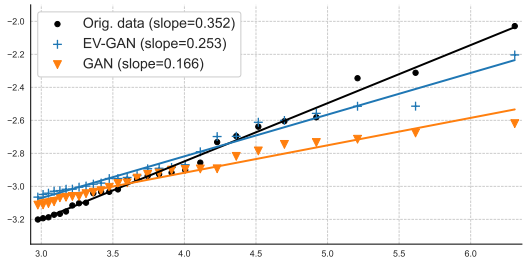
(a) AEX



(b) CAC



(c) KOSPI



(d) NDQ

Figure 6: Log quantile-quantile plots $\log((n+1)/i) \mapsto \log X^{(j)}_{n-i+1,n}$, pour $i \in \{1, \dots, \lceil (1-\xi)n \rceil\}$ et $j \in \{1, \dots, 4\}$ au niveau de probabilité $\xi = 0.95$ sur quatres indices financiers: AMX (Bourse d'Amsterdam, Pays-Bas), CAC (France), KOSPI (Corée du sud), Nasdaq (USA).

Theorem 0.1.11 (version simplifiée de [5, Théorème 2]). *Supposons que la condition du J-ième order (3.3.3) soit vérifiée, ainsi que l'hypothèse de représentation (3.3.5) pour un certain $J \geq 2$. Alors, l'approximation par NN à une couche cachée (3.3.8) du quantile extrême $q(1-\alpha_n)$ est telle que*

$$\inf_{\tilde{\phi} \in \Phi} \left| \log q(1-\alpha_n) - \log \tilde{q}^{\mathtt{NN}_J}_{\tilde{\phi}}(1-\alpha_n; 1-\delta_n) \right| = \mathcal{O}\left(\alpha_n^{-\bar{\rho}_J}\right),$$

où $\bar{\rho}_J = \rho_2 + \cdots + \rho_J$ lorsque $\alpha_n \to 0$ et $\delta_n/\alpha_n \to \infty$ lorsque $n \to \infty$.

Observons que la vitesse de convergence de l'erreur uniforme entre les log-quantiles extrêmes et leur approximation par NN est principalement déterminée par le niveau extrême α_n et par la somme de tous les paramètres d'ordre j, $j \in \{2, \dots, J\}$. Comme prévu, la demande d'une plus grande régularité dans le modèle de valeurs extrêmes (par la condition de J-ième ordre) produit une erreur d'approximation plus petite grâce à une plus grande largeur du NN proposé. L'analyse numérique montre que par rapport à d'autres estimateurs débiaisés sur des données simulées, l'estimateur NN non conditionnel les surpasse dans des situations à queue lourde difficiles où les autres concurrents échouent presque tous.

Supposons maintenant que X est une variable aléatoire associée à un vecteur aléatoire explicatif $Y \in \mathcal{Y} \subset \mathbb{R}^{d_y}, d_y \geq 1$. Nous présentons deux approches pour estimer les quantiles extrêmes conditionnels par un NN. La première est l'extension conditionnelle de l'approche non conditionnelle discutée ci-dessus.

Theorem 0.1.12 (simplified version of [5, Théorème 3]). *Supposons qu'une extension conditionnelle de la condition d'ordre J tienne ($J \geq 2$). De plus, supposons que toutes les fonctions incluses dans l'extension conditionnelle de la fonction d'espacement logarithmique sont continues sur l'ensemble compact $\mathcal{Y} \subset \mathbb{R}^{d_y}$. Soit $\bar{\rho}_{\mathtt{sup}} = \sup_{y \in \mathcal{Y}} \bar{\rho}_J(y)$ avec $\bar{\rho}_J(y) = \rho_2(y) + \cdots + \rho_J(y)$.*

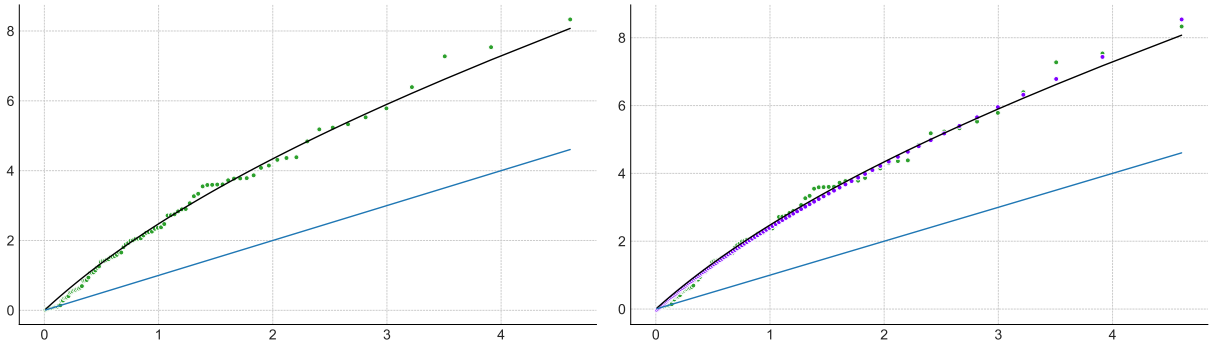


Figure 7: Fonction d'espacement des logarithmes associée à une distribution de Burr ($\gamma = 1, \rho = -1/4$). Courbe noire : fonction théorique $x_1 \mapsto \gamma x_1 + \varphi(x_1 + \log(n/k))$, courbe bleue : approximation de Weissman $x_1 \mapsto \gamma x_1$, points verts : estimation empirique ponctuelle $(\log(k/i), \log X_{n-i+1,n} - \log X_{n-k+1,n})$, points violets : Estimation NN avec $i \in \{1, \dots, k-1\}$, $k = 100$ et $n = 500$.

Alors, il existe une approximation NN conditionnelle profonde du quantile extrême conditionnel $q(1-\alpha_n \mid y)$ *comprenant* $2J(J-1)+1$ *sous-réseaux construits avec un nombre fixe de* $2d_y + 10$ *neurones dans chacune des couches cachées avec au moins une profondeur d'ordre* $\alpha_n^{\bar{\rho}_{\text{sup}}/2}$ *telle que*

$$\inf_{\tilde{\phi}\in\Phi} \sup_{y\in\mathcal{Y}} \left| \log q(1-\alpha_n \mid y) - \log \tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1-\alpha_n; 1-\delta_n \mid y) \right| = \mathcal{O}\left(\alpha_n^{-\bar{\rho}_{\text{sup}}}\right),$$

lorsque $\alpha_n \to 0$ *et* $\delta_n/\alpha_n \to \infty$ *quand* $n \to \infty$.

Dans ce cadre conditionnel général, une profondeur minimale (de l'ordre de $\simeq \alpha_n^{\bar{\rho}_{\text{sup}}/2}$) est requise pour que le premier NN d'extrapolation conditionnelle puisse se rapprocher du quantile extrême avec une erreur donnée (de l'ordre de $\simeq \alpha_n^{-\bar{\rho}_{\text{sup}}}$) alors que, dans la situation précédente, un NN à une couche était suffisant. Le deuxième NN d'extrapolation conditionnelle tire parti d'une hypothèse de modèle de position-dispersion (*locaton-dispersion*) [177] pour se débarrasser de la covariable dans l'étape d'extrapolation.

Theorem 0.1.13 (version simplifiée de [5, Théorème 4])**.** *Supposons que l'hypothèse du modèle de position-dispersion soit vérifiée sous les conditions du Théorème 0.1.11 avec des hypothèses supplémentaires sur les bords et de continuité. Alors, il existe une approximation par NN à une couche cachée du quantile extrême conditionnel* $q(1-\alpha_n \mid y)$ *telle que*

$$\inf_{\tilde{\phi}\in\Phi} \sup_{y\in\mathcal{Y}} \left| \log q(1-\alpha_n \mid y) - \log \tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1-\alpha_n; 1-\delta_n, 1-\tau_n \mid y) \right| = \mathcal{O}(\alpha_n^{-\bar{\rho}_J}) + \mathcal{O}(\tau_n^{-\bar{\rho}_J-\gamma}\delta_n^{\gamma}) \tag{0.1.16}$$

avec $\alpha_n \to 0$, $\delta_n/\tau_n \to 0$ *et* $\delta_n/\alpha_n \to \infty$ *quand* $n \to \infty$.

Il est alors possible de régler la valeur de la séquence supplémentaire δ_n pour équilibrer les deux termes d'erreur dans (0.1.16) :

Corollary 0.1.3 (version simplifiée de [5, Corollaire 5])**.** *Supposons que les hypothèses du Théorème 0.1.13 soient vérifiées.*

- *Si* $\gamma + \bar{\rho}_J > 0$, *alors en prenant* $\delta_n = \alpha_n^{-\bar{\rho}_J/\gamma}\tau_n^{1+\bar{\rho}_J/\gamma}$ *on obtient*

$$\inf_{\tilde{\phi}\in\Phi} \sup_{y\in\mathcal{Y}} \left| \log q(1-\alpha_n \mid y) - \log \tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1-\alpha_n; 1-\delta_n, 1-\tau_n \mid y) \right| = \mathcal{O}(\alpha_n^{-\bar{\rho}_J}).$$

- *Si* $\gamma + \bar{\rho}_J \leq 0$*, alors en prenant* $\delta_n = \xi_n \alpha_n$ *et* $\tau_n = \xi_n^2 \alpha_n$ *avec* $\xi_n \to \infty$ *arbitrairement lent quand* $n \to \infty$ *on obtient*

$$\inf_{\tilde{\phi} \in \Phi} \sup_{y \in \mathcal{Y}} \left| \log q(1 - \alpha_n \mid y) - \log \tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1 - \alpha_n; 1 - \delta_n, 1 - \tau_n \mid y) \right| = \mathcal{O}(\alpha_n^{-\bar{\rho}_J} \xi_n^{-2\bar{\rho}_J - \gamma}).$$

À un terme ξ_n près, on peut retrouver la vitesse de convergence $\alpha_n^{\bar{\rho}_J}$ du cas non-conditionnel, voir le Théorème 0.1.11. Les deux modèles conditionnels sont implétentés pour étudier le comportement et l'interpolation spaciale des précipitations extrêmes en fonction de leur emplacement géographique dans le sud de la France (voir Figure 8).

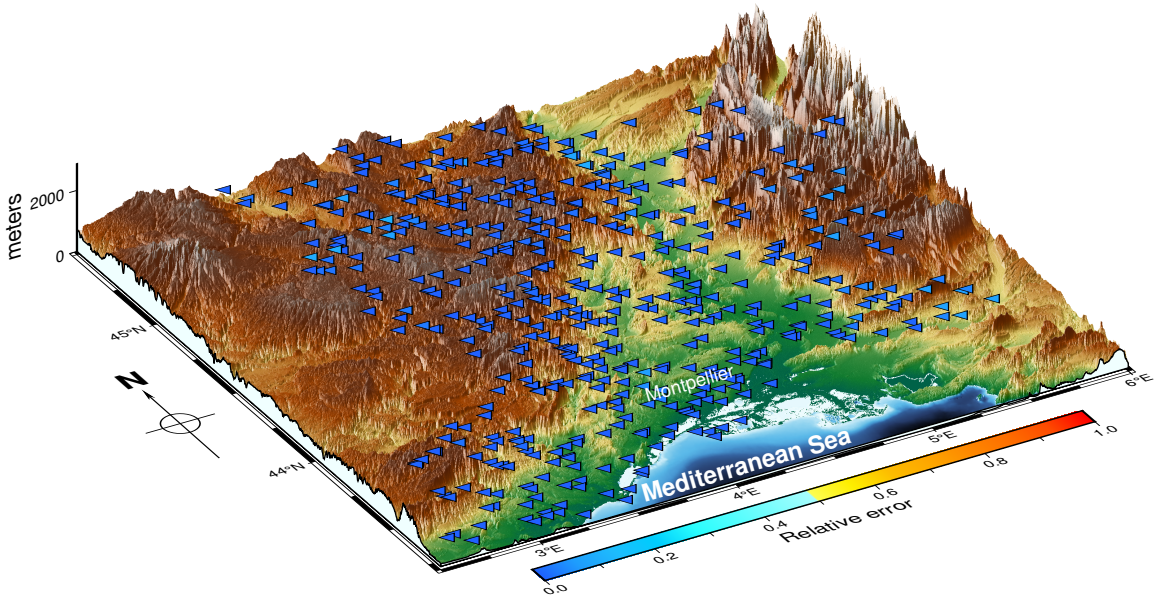


Figure 8: Estimation d'un des modèles NN de quantile extrême conditionnel à chaque station.

0.2 Apprentissage par dictionnaire des matrices de migration des notations

0.2.1 Contexte

Le risque de crédit désigne le risque de subir des pertes dues à des changements inattendus de la qualité de crédit de la contrepartie. Les crises financières qui se sont succédées au cours du XXe siècle ont conduit les superviseurs bancaires à s'intéresser à ce risque et ont abouti à la création d'un ratio de solvabilité, appelé *Ratio Cooke*, qui a été au cœur de la première réglementation bancaire internationale (Bâle I). Malgré ces premiers efforts, l'évolution des métiers de la banque et des crises financières a nécessité une révision en profondeur du cadre réglementaire (Bâle II et III). Selon ces derniers, les banques peuvent utiliser des notations internes et des estimations de l'exposition aux risques afin d'évaluer les exigences de fonds propres réglementaires et les mesures du risque de crédit (*Value at Risk*, *Expected Shortfall*, ...).

Le risque de crédit est résumé dans une matrice de migration de notation structurée qui capture toutes les probabilités de transition possibles pour qu'un obligataire passe d'un état de crédit à un autre sur une période donnée (voir Figure 9). Voir [11, 25, 37, 141] pour de nombreuses références sur les mesures de risque et le risque de crédit. Les matrices de migration des notations (RMM de l'anglais *Rating Migration Matrices*) sont des indicateurs clés pour évaluer le portefeuille de risque de crédit par l'estimation de la qualité de crédit des obligataires.

Le processus d'attribution des notations comprend des modèles et des systèmes experts prenant en compte les caractéristiques spécifiques des obligataires qui évoluent dans le temps en fonction de la situation économique. En pratique, nous observons au temps t une matrice de migration de notation sur un an $\mathbf{P}^t \in \mathbb{R}^{R-1} \otimes \mathbb{R}^R$, qui encode la probabilité de migration de la notation $i \in [R-1] := \{1, \dots, R-1\}$ à la notation $j \in [R]$ au cours d'une période d'un an commençant au temps $t-1$, où R représente l'état par défaut ; à la Figure 9 nous avons $R = 11$. La reconstruction de cette matrice se fait empiriquement en évaluant les fréquences des obligataires passant de la notation i à la notation j entre les temps $t-1$ et t. Les fréquences de migration observées sont affichées dans des RMM qui constituent la pierre angulaire des modèles de migration des notations sur lesquels repose la simulation de portefeuille de risque de crédit.

Modèle Copule à un facteur. La méthode la plus utilisée pour modéliser les RMM est le modèle de Copule Gaussienne à un facteur [127]. Ce dernier appartient à la classe des "modèles à seuil", qui sont très populaires dans la modélisation du risque de crédit [141, Section 11.1]. Voir [27] entre autres pour estimer les mesures de risque sur la distribution des pertes d'un large portefeuille de risque de crédit sous ce modèle.

Pour une notation initiale i au temps t, le modèle de Copule à un facteur suppose que l'événement "migration vers la notation j au temps $t+1$" est donné par

$$\left\{X_i^t \in [c_{i,j}, c_{i,j+1})\right\},$$

où les paramètres $\{c_{i,j}\}_{i\in[R-1],j\in[R]}$ sont des seuils déclenchant la migration de notation et avec un facteur stochastique

$$X_i^t = \rho Z^t + \sqrt{1-\rho^2}\epsilon_i^t,$$

composé d'un risque systémique $Z^t \overset{d}{=} \mathcal{N}(0,1)$ (commun à tous les obligataires), d'un risque idiosyncratique $\epsilon_i^t \overset{d}{=} \mathcal{N}(0,1)$ (spécifique à chaque obligataire), indépendant de Z^t, et d'un paramètre de corrélation $\rho \in (-1,1)$ entre les deux sources de risque. De manière équivalente,

$$\begin{aligned}
&\{\text{"migration vers la notation supérieure à } j \text{ au temps } t+1\text{"}\} \\
&\quad = \bigcup_{k\geq j} \{X_i^t \in [c_{i,k}, c_{i,k+1})\} \\
&\quad = \{X_i^t \geq c_{i,j}\} = \{\rho Z^t + \sqrt{1-\rho^2}\epsilon_i^t \geq c_{i,j}\}.
\end{aligned}$$

Conditionnellement à Z^t, cet événement a la probabilité

$$\Phi\left(\frac{\rho Z^t - c_{i,j}}{\sqrt{1-\rho^2}}\right) =: P_{i,\geq j}^t := P_{i,j}^t + \cdots + P_{i,R}^t, \tag{0.2.1}$$

et la probabilité non-conditionnelle

$$\Phi(-c_{i,j}) =: P_{i,\geq j}^{\mathrm{TTC}}, \tag{0.2.2}$$

où Φ est la c.d.f. d'une loi Gaussienne centrée réduite. La matrice $\mathbf{P}^{\mathrm{TTC}}$ représente la probabilité de défaut *'Through the Cycle'* qui est une moyenne à long terme sur un cycle et se concentre principalement sur les composantes permanentes du risque de défaut, tandis que la matrice $\mathbf{P}^t$ représente la probabilité de défaut *'Point-In-Time'* qui prend en compte à la fois l'effet cyclique et l'effet permanent. En inversant $c_{i,j}$ dans (0.2.2) et en le remplaçant dans (0.2.1) on obtient

$$\begin{aligned}
P_{i,\geq j}^t &:= \Phi\left(\frac{\rho Z^t + \Phi^{-1}(P_{i,\geq j}^{\mathrm{TTC}})}{\sqrt{1-\rho^2}}\right), \\
P_{i,j}^t &:= \Phi\left(\frac{\rho Z^t + \Phi^{-1}(P_{i,\geq j}^{\mathrm{TTC}})}{\sqrt{1-\rho^2}}\right) - \Phi\left(\frac{\rho Z^t + \Phi^{-1}(P_{i,\geq j+1}^{\mathrm{TTC}})}{\sqrt{1-\rho^2}}\right),
\end{aligned}$$

avec la convention $P^{\mathrm{TTC}}_{i,\geq R+1} = 0$. Afin de simuler des $\mathbf{P}^t$, on peut considérer le facteur systémique $(Z^t)_{t\in[T]}$ évoluant comme un processus stationnaire auto-régressif (d'ordre 1)

$$Z^t = \kappa Z^{t-1} + \epsilon_z^t,$$

où le bruit d'innovation ϵ_z^t est indépendant en t et distribué comme $\mathcal{N}(0, 1-\kappa^2)$ de sorte que la distribution stationnaire de $(Z^t)_{t\geq 1}$ soit $\mathcal{N}(0,1)$ avec $\kappa \in (-1,1)$.
La popularité du modèle de Copule Gaussienne est due à sa facilité d'utilisation mais il souffre également d'hypothèses sous-jacentes trop simples. Ces hypothèses faibles conduisent à une mauvaise capture de la structure de dépendance dans les queues de distribution. Cependant, et malgré les critiques formulées après la crise des subprimes [129], le modèle de Copule Gaussienne à un facteur reste très populaire dans le secteur bancaire en raison de son faible coût et de sa capacité à générer des résultats intuitifs et interprétables même avec une faible quantité de données disponibles.

Dans ce contexte, nous avons voulu développer une représentation non-paramétrique des RMM avec une approche basée sur les données comme une alternative et un challenger au modèle paramétrique de la Copule Gaussienne. Dans la suite, nous exposons l'approche par factorisation matricielle que nous avons utilisée, nous soulevons certains défis importants et nous discutons de nos contributions.

	1	2	3	4	5	6	7	8	9	10	11
1	71.48	17.87	5.36	2.38	1.25	0.71	0.43	0.26	0.15	0.08	0.03
2	16.01	57.62	14.41	5.69	2.80	1.54	0.89	0.52	0.30	0.16	0.06
3	5.05	15.14	53.85	13.46	5.89	3.03	1.68	0.96	0.54	0.28	0.11
4	2.45	6.52	14.67	51.35	12.84	5.87	3.06	1.68	0.92	0.47	0.18
5	1.45	3.61	7.23	14.46	49.57	12.39	5.74	2.95	1.55	0.76	0.30
6	0.96	2.31	4.34	7.71	14.46	48.19	12.05	5.51	2.71	1.29	0.48
7	0.70	1.63	2.94	4.90	8.16	14.69	47.00	11.75	5.14	2.28	0.82
8	0.54	1.24	2.18	3.48	5.44	8.71	15.24	45.71	11.43	4.51	1.52
9	0.45	1.02	1.75	2.73	4.09	6.14	9.55	16.38	43.67	10.92	3.28
10	0.43	0.95	1.60	2.43	3.55	5.11	7.46	11.36	19.17	38.35	9.59

Figure 9: Représentation d'une matrice de migration des notations idéalisée de taille 10×11. Toutes les valeurs sont exprimées en pourcentage. La qualité du crédit va de la plus élevée (note 1) à la plus faible (note 10), la valeur du défaut étant 11.

0.2.2 Factorisation matricielle

Il est important, à des fins de gestion des risques, de modéliser l'évolution de $\mathbf{P}^t$, en trouvant une représentation du type

$$\mathbf{P}^t \approx \sum_{k=1}^{K} \alpha_k^t \mathbf{d}_k, \quad \forall t \geq 1,$$

pour des vecteurs dits (déterministes) de base $\mathbf{d}_k$, dont on doit contraindre leur représentation à être des RMM, et pour certains coefficients aléatoires scalaires α_k^t, dont on doit modéliser l'évolution. Sous une forme matricielle (en utilisant l'opérateur vec () pour simplifier, voir Section 4.1.5) pour la définition des notations, on dit que la collection de matrices vectorisées $\mathbf{P} = \{\mathrm{vec}\left(\mathbf{P}^t\right) \in \mathbb{R}^d\}_{t=1}^T \in \mathbb{R}^d \otimes \mathbb{R}^T$, avec $d := (R-1)R$ pour tous les $t = 1, \ldots, T$, admet une factorisation matricielle sur un dictionnaire $\mathbf{D} \in \mathbb{R}^d \otimes \mathbb{R}^K$ composé de K éléments (appelés atomes), s'il existe une combinaison linéaire d'atomes pondérés par des coefficients (appelés codages) $\mathbf{A} = \{\boldsymbol{\alpha}^t \in \mathbb{R}^K\}_{t=1}^T \in \mathbb{R}^K \otimes \mathbb{R}^T$ tel que

$$\mathbf{P} \approx \mathbf{DA}. \tag{0.2.3}$$

Au cours des dernières années, un nouveau paradigme de modèles basés sur les données a émergé dans la communauté ML afin d'extraire des informations structurées d'objets en grande dimension. Une approche classique en ML consiste à utiliser des techniques de factorisation matricielle afin de projeter les données dans une base pertinente. Il est bien connu que la base optimale qui minimise l'erreur d'approximation linéaire est la base de Karhunen-Loève [137, Théorème 9.8], également connue comme les composantes principales dans l'analyse en composantes principales (ACP). Cependant, il est important de souligner que cette base ne pourra pas satisfaire les contraintes linéaires imposées par la structure RMM.

Contraintes des données

Une matrice de migration de notation $\mathbf{M}$ doit satisfaire certaines contraintes pour des raisons à la fois mathématiques et économiques.
Contraintes mathématiques. Chaque ligne de $\mathbf{M}$ est une probabilité discrète, donc $\mathbf{M}$ est une matrice stochastique. L'ensemble des matrices stochastiques est noté par

$$\mathcal{M}^S := \Big\{ \mathbf{M} \in \mathbb{R}^{(R-1)} \otimes \mathbb{R}^R : \sum_{j\in[R]} M_{i,j} = 1, \forall i \in [R-1], \\ M_{i,j} \geq 0, \forall (i,j) \in [R-1] \times [R] \Big\}.$$

Contraintes économiques. Selon leur expertise, certains gestionnaires de risques peuvent juger important d'ajouter des contraintes supplémentaires significatives d'un point de vue économique. Par exemple, la probabilité de défaut des contreparties les mieux notées est plus faible que celle des contreparties de moindre qualité. Ainsi, la collection des matrices de notation satisfaisant les contraintes dites économiques est notée par

$$\mathcal{M}^E := \Big\{ \mathbf{M} \in \mathbb{R}^{(R-1)} \otimes \mathbb{R}^R : M_{i,\geq j} \leq M_{i',\geq j}, \\ \forall j \in [R], \quad 1 \leq i < i' \leq R-1 \Big\}.$$

Une matrice satisfaisant de telles contraintes est appelée matrice idéalisée et est illustrée à la Figure 9.

Question. Comment obtenir (0.2.3) tout en exigeant que $\mathbf{D}$ satisfasse les contraintes linéaires ci-dessus afin de représenter des RMM économiquement interprétables ?

Introduit dans [150], l'apprentissage par dictionnaire (DL de l'anglais *Dictionary Learning*), voir [59] pour un aperçu et [101] pour des résultats théoriques, est une autre technique de représentation matricielle où la base, appelée dictionnaire, est apprise à partir des observations. Contrairement à la décomposition ACP, ni l'orthogonalité ni les contraintes de représentation des vecteurs de base (atomes) ne sont imposées, ce qui permet une plus grande flexibilité pour adapter la représentation souhaitée aux données. De plus, comparé à un dictionnaire prédéfini comme les fonctions de Gabor, les ondelettes ou les vecteurs cosinus locaux [137], l'apprentissage d'un dictionnaire adapté aux observations a montré de meilleurs résultats en pratique [62, 134].

En DL, l'approximation linéaire (0.2.3) est généralement couplée à un critère de régularisation $\mathcal{R}(\mathbf{A})$ appliqué aux codages et aboutit au problème d'optimisation général

$$\min_{\mathbf{D},\mathbf{A}} \|\mathbf{P} - \mathbf{DA}\|_F^2 + \lambda \mathcal{R}(\mathbf{A}), \quad \lambda \geq 0, \tag{0.2.4}$$

où le terme de régularisation doit refléter la représentation attendue des codages, voir [59, Chapitre 4]. La régularisation la plus largement étudiée est $\mathcal{R}(\mathbf{A}) = \|A\|_1$ se référant à ce qu'on appelle le codage parcimonieux (*sparse coding*) (voir [132] pour une vue d'ensemble), où l'optimisation par rapport à $\mathbf{A}$ est connue sous le nom de *basis pursuit* [36] ou de Lasso [176]. Le DL avec représentation parcimonieuse a notamment été étudié en traitement d'images et de vidéos [133, 135, 136], en apprentissage de graphes [179] et en *clustering* [171].

Question. Comment obtenir (0.2.3) tout en exigeant que la série temporelle des éléments $\boldsymbol{\alpha}^t$ de $\mathbf{A}$ soit suffisamment lisse afin de réaliser des prédictions à travers une modélisation de la série temporelle ?

L'application de DL avec une structure temporelle a été principalement étudiée dans le débruitage vidéo [136, 156] où la structure temporelle est exploitée par un opérateur extrayant des patchs de taille fixe dans la fonction objectif représentant une procédure de minimisation de l'énergie. Une autre approche consiste à traiter une représentation auto-régressive (AR) modélisée soit dans le dictionnaire [38] soit dans les codages [191]. Dans le premier cas, un signal audio mixte est décomposé en ses sources temporelles constitutives (atomes du dictionnaire) afin de détecter la présence d'un son spécifique. Dans le second cas, les auteurs présentent un cadre qui permet l'apprentissage et la prévision temporels *data-driven* grâce à une modélisation AR des codages représentée par un terme de régularisation. Notre modèle décrit dans la Section 4.2 s'inspire de cette formulation du problème.

Défi. La modélisation des RMM est un problème difficile car il est nécessaire de trouver une représentation interprétable satisfaisant les contraintes économiques, tandis que l'évolution des RMM peut varier rapidement dans le temps et que l'on dispose d'un historique de données limité (généralement 10 à 20 ans, environ 200 observations) qui est proche de la dimension du problème (généralement $R = 11$ et $d = 110$). Ainsi, la modélisation des RMM sous contraintes de manière non paramétrique basée sur les données représente un défi important, qui n'a pas été abordé jusqu'à présent à notre connaissance.

0.2.3 Contributions de la thèse

Dans le contexte de DL structuré des RMM pour la modélisation du risque de crédit, nous nous sommes intéressés à dériver à partir des données une représentation non paramétrique des RMM, comme une alternative et un challenger au modèle paramétrique de la Copule Gaussienne.

Contributions au chapitre 4. Nous proposons une nouvelle technique de modélisation RMM utilisant une approche DL. Cette dernière prend en compte les contraintes linéaires imposées par la structure RMM et encourage les codages $\mathbf{A}$ à avoir une structure AR grâce à un terme de régularisation temporelle dans (0.2.4) :

$$\mathcal{R}_{AR}(\mathbf{A}, \mathbf{w}) := \sum_{k=1}^{K} \sum_{t=1}^{T-1} \left(\alpha_k^{t+1} - \frac{1}{T}\alpha_k^t - w_k \left(\alpha_k^t - \frac{1}{T}\alpha_k^t \right) \right)^2, \tag{0.2.5}$$

où le paramètre supplémentaire $\mathbf{w}$ nous permet d'estimer les paramètres AR de la série temporelle $\boldsymbol{\alpha}_k =: \mathbf{A}_{k,:}$ pour chaque $k \in [K]$, pour une classification interprétable et une prédiction des RMM. Nous considérons un cadre de réduction de la dimensionnalité $K \ll d$ afin de travailler dans un espace de dimension inférieure avec des informations extraites utiles. Le problème DL proposé est

$$\min_{\substack{\mathbf{D},\mathbf{A},\mathbf{w} \\ \mathbf{D}\in\Omega,\ \alpha_k^t\geq 0,\ t\in[T],\ k\in[K]}} \|\mathbf{P} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda \mathcal{R}_{AR}(\mathbf{A}, \mathbf{w}), \tag{0.2.6}$$

qui est convexe en les variables $\mathbf{D}$,$\mathbf{A}$ et $\mathbf{w}$, lorsque les autres sont fixées, et où Ω est l'ensemble convexe des dictionnaires vérifiant les contraintes idéalisées (voir la section 0.4.2)

$$\Omega := \left\{ \mathbf{D} \in \mathbb{R}^d \otimes \mathbb{R}^K : \mathrm{vec}^{-1}(\mathbf{d}_k) \in \mathcal{M}^E \cap \mathcal{M}^S,\ \forall k \in [K] \right\},$$

avec $\mathrm{vec}^{-1}(.)$ la fonction inverse de vectorisation (voir Section 4.1.5).

Mise à jour du dictionnaire. Nous optons pour une mise à jour séquentielle de chaque atome du dictionnaire : $\mathbf{d}_k$ pour $k \in [K]$. Ce choix est guidé par deux avantages : 1. Le problème est strictement convexe pour chaque atome $\mathbf{d}_k$ (comme indiqué dans la Proposition 0.2.1 ci-dessous), ce qui n'est pas nécessairement vrai pour la matrice entière $\mathbf{D}$. 2. Cette stratégie divise le problème en problèmes plus petits, ce qui rend la résolution moins dépendante de la quantité d'atomes K. La mise à jour des atomes séparément est également la stratégie de la K-SVD largement utilisée (voir [59, Section 3.5]), cependant, le but dans ce cas est de trouver une forme fermée pour le problème d'optimisation, ce qui n'est pas vrai dans notre cas d'étude en raison de la forme des contraintes.

Proposition 0.2.1 (version simplifiée de [8, Proposition 2.1])**.** *Supposons que* $\{\alpha_k^t\}_{t=1}^T$ *est non nul. La minimisation de* (0.2.6) *sur* $\mathbf{d}_k$ *dans* $\mathrm{vec}\left(\mathcal{M}^E \cap \mathcal{M}^S\right)$ *est équivalente à la minimisation d'un problème quadratique strictement convexe avec des contraintes linéaires*

$$\min_{\mathbf{d}_k} \left\| \mathrm{vec}\left(\widetilde{\mathbf{P}}_k\right) - \widetilde{\mathbf{A}}_k \mathbf{d}_k \right\|_F^2, \quad s.t.\ vec^{-1}(\mathbf{d}_k) \in \mathcal{M}^E \cap \mathcal{M}^S,$$

où $\widetilde{\mathbf{P}}_k$ *et* $\widetilde{\mathbf{A}}_k$ *sont explicitement définis (voir* (4.2.9)*).*

Mise à jour du codage. Comme pour la mise à jour du dictionnaire, nous adoptons une stratégie basée sur la mise à jour de chaque $\mathbf{A}_{k,:}$ pour $k \in [K]$. Les raisons sont les mêmes : il est préférable de résoudre un problème d'optimisation plus petit et strictement convexe. Le fait que l'optimisation pour chaque k soit un problème fortement convexe n'est pas simple et est argumenté dans la proposition ci-dessous.

Proposition 0.2.2 (version simplifiée de [8, Proposition 2.2])**.** *Soit* $k \in [K]$ *fixé. Considérons la minimisation de* (0.2.5)-(0.2.6) *sur un codage* $\mathbf{A}_{k,:}$*, soit*

$$\min_{\mathbf{A}_{k,:},\ A_{k,t} \geq 0} \|\mathbf{P} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda \sum_{k=1}^{K} \sum_{t=1}^{T-1} \left(A_{k,t+1} - \bar{\mathbf{A}}_{k,:} - w_k (A_{k,t} - \bar{\mathbf{A}}_{k,:}) \right)^2,$$

où $\bar{\mathbf{A}}_{k,:} := 1/T \sum_{t=1}^T \alpha_k^t$*. Pour tout* $\lambda \geq 0$*, le problème ci-dessus est un problème d'optimisation quadratique fortement convexe avec des contraintes linéaires.*

Mise à jour du coefficient. Nous notons que, pour chaque $k \in [K]$, le problème d'optimisation par rapport à $\mathbf{w}_k$ dans l'équation (0.4.6) est un problème quadratique en dimension 1 avec une solution explicite (voir (4.2.5)).

Pour montrer l'applicabilité du modèle, nous présentons un test numérique avec des données réelles qui bénéficie d'une bonne précision de reconstruction et inclut la classification supervisée des RMM basée à la fois sur un algorithme K-means et sur la représentation du $\mathbf{D}$ entraîné. Nous observons que l'indicateur de sentiment économique estimé qui en résulte est conforme aux événements économiques historiques (voir Figure 10). Enfin, notre approche DL surpasse de manière significative le modèle de Copule Gaussienne largement utilisé, et apparaît donc comme un modèle alternatif efficace.

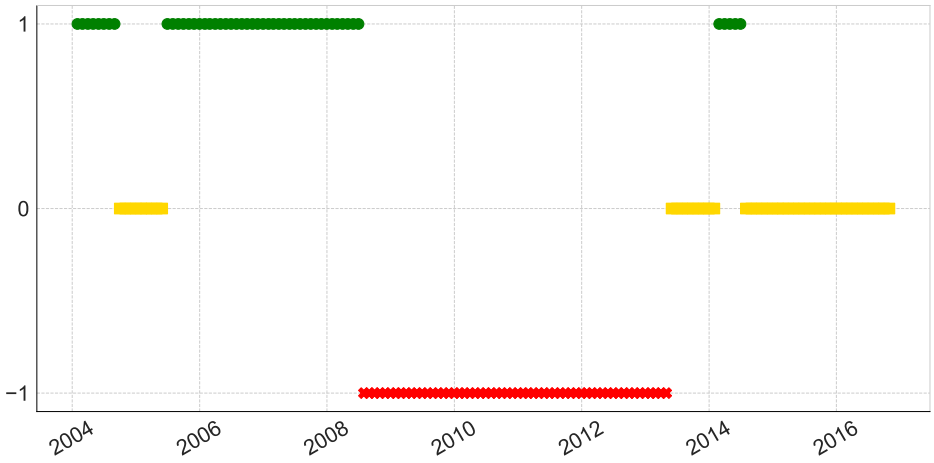


Figure 10: Classification des RMM réels dans les sentiments économiques 3 : $\{-1$ (croix rouge): mauvais, 0 (carré jaune): stable, 1 (point vert): bon$\}$.

Introduction

The first Chapter is dedicated to contextualize the concepts and challenges addressed along this thesis, composed by contributions in two main topics: generative modeling and dictionary learning.

0.3 Generative modeling

0.3.1 Context

Motivations

Generative model aims at mimicking the law of an object (possibly in high dimension), commonly known in industrial engineering as a digital twin, see [121, 144] for a review. Such models are particularly useful for example in the field of data-augmentation: enriching a dataset, leading to reduce overfitting and improve the performance of the statistical models. Another perspective of high interest is for data-privacy: sharing generated data which have the same statistical properties as some confidential ones. In the context of generative modeling, we distinguish two different point of views. On the first hand, sampling complex motions of physical objects was originally done by solving mathematical equations describing the object behavior under constraints and by generating trajectories given different initial conditions. Such an approach requires first to know the exact formula or to build by hand the mathematical model. Solving and building such a model can be very hard but have shown its efficiency in many areas. On the other hand, a new class of data-based generative models has emerged recently in the paradigm of Artificial Intelligence (AI). Instead of looking for the physical model, one can try to learn it directly from the data using random noise as input. Such algorithms have the advantage of being slow in the inference phase and fast in the simulation phase compared to their physical model counterparts. Thanks to the numerical and theoretical evolution in the XXIst century, neural networks (NNs) have proven to be excellent candidates as universal approximation functions. Among the NN generative models developed [71], the most popular ones have been the Variational Autoencoder (VAE) [115] based on Variational Inference, and the Generative Adversarial Network (GAN) [100], on which we focused on, based on a minmax game. More recent models such as Normalizing Flows [161] and Diffusion Models [169] have gained some popularity.

Nowadays, both the construction and the optimization of heavy NN models are made easy by open-source libraries (*e.g.* TensorFlow [1] or PyTorch [151]), resulting in an extraordinary interest of people coming from different communities and mathematical backgrounds. Unlike many generative modeling works developed in the AI community, we have endeavored to propose error bounds on the uniform approximation through structural information added to our models based on rigorous theoretical analysis. We also focused on convergence rates which allow to give theoretical guidelines to build a NN in order to achieve a fixed error. This NN architecture design is a very difficult question and often left in the literature to empirical grid search.

We have contributed to two major topics. First, we have investigated continuous time series simulation (infinite dimensional object), in particular in a non-Markovian case, which has not yet been studied in the generative NN literature (see Chapter 1). Second, we have tackled

the issue of simulating extreme values from heavy-tailed distributions, which is impossible in a classical generative NN framework (see Chapters 2 and 3).

In the following, we first recall the statistical framework of generative modeling. Then, we expose the state of the art on NN approximation results and discuss existing theoretical results in the literature on GANs. Moreover, we raise some important challenges for simulating both continuous time series (Section 0.3.2) and heavy tailed distributions (Section 0.3.3). Finally, we summarize our contributions (Section 0.3.4).

Statistical framework

Let X be a random variable taking values in a general metric space $(\mathcal{X}, d_{\mathcal{X}})$ and let $\mathcal{P}(\mathcal{X})$ be the space of all probability measures defined on the Borel σ-algebra $\mathcal{B}_{\mathcal{X}}$. Additionally, assume that all probabilities considered are dominated by a fixed, known (*e.g.* Lebesgue), reference measure $\pi \in \mathcal{P}(\mathcal{X})$. Then, given some observations $\{X_i \in \mathcal{X}\}_{i=1}^n$ drawn according to an unknown density distribution p_X on $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$, the objective in generative modeling is to find a function $G : \mathcal{Z} \to \mathcal{X}$, called a generator, and a probability distribution p_Z, called latent distribution, such that

$$G(Z) \overset{d}{=} X, \quad Z \sim p_Z. \tag{0.3.1}$$

Question. Which class of functions G and densities p_Z may be considered to ensure that (0.3.1) holds?

The answer is provided by the following Kuratowski's Theorem ([19, Proposition 7.15, p. 121]), also called the measurable isomorphism ([178, p.7]), in the case where $\mathcal{Z}$ and $\mathcal{X}$ are Polish spaces and G is a measurable bijection.

Theorem 0.3.1 (Kuratowski)**.** *Let $(\mathcal{Z}, \mu_Z)$ and $(\mathcal{X}, \mu_X)$ two Polish probability spaces. Then there exists a (non-unique) measurable bijection $G : \mathcal{Z} \to \mathcal{X}$ such that*

$$G_{\#}\mu_Z = \mu_X, \quad G^{-1}_{\#}\mu_X = \mu_Z,$$

where $G_{\#}\mu_Z(E) := \mu_Z(G^{-1}(E)), \forall E \subset \mathcal{X}$, stands for the push-forward of μ_Z by G.

Here we focus on a parametric family of generators $\mathcal{G} := \{G_\theta\}_{\theta \in \Theta}$ and we denote by $\mathcal{P} := \{p_\theta\}_{\theta \in \Theta}$ their associated parametric densities such that $G_\theta(Z) \overset{d}{=} p_\theta \, \mathrm{d}\pi$. Thus, the problem comes down to find the best parameters $\theta^\star$ such that $p_{\theta^\star}$ and p_X are as close as possible, or equivalently

$$G_{\theta^\star}(Z) \overset{d}{\approx} X, \quad Z \sim p_Z. \tag{0.3.2}$$

A generative modeling problem mainly consists in choosing three ingredients:

1. the parametrization of G_θ and the latent distribution p_Z to use as inputs,
2. the observations $X_1, \dots, X_n$ with their underlying properties,
3. the optimization process and the distance between probability density distributions p_θ and p_X.

In this thesis we address the first two points, while the last one will be mentioned in the following when presenting the GAN model. In the new paradigm of AI, it is natural to consider the NN as a parametrization G_θ.

Neural networks

A NN is a non-linear function built with a fixed number of neurons, each one representing a function, and distributed across several hidden layers. Neurons are scaled and translated in the network by parameters called respectively weights and biases. Among many different existing NN architectures, consider the classical one-hidden layer feedforward NN (Figure 11) $G_{\theta_K} : \mathbb{R}^{d'} \to \mathbb{R}$ composed by K neurons such that

$$G_{\theta_K}(\mathbf{z}) = b^{(2)} + \sum_{k=1}^{K} w_k^{(2)} \sigma\left(\left\langle \mathbf{w}_k^{(1)}, \mathbf{z} \right\rangle + b_k^{(1)}\right), \tag{0.3.3}$$

with parameters $\theta_K = \left\{\mathbf{w}_k^{(1)}, w_k^{(2)}, b_k^{(1)}, b_2\right\}_{k=1}^{K} \in \Theta_K, \mathbf{z} \in \mathbb{R}^{d'}$, $\langle \cdot, \cdot \rangle$ a scalar product in $\mathbb{R}^{d'}$, and σ the non-linear function called activation function. The later is part of a wider class called Ridge functions [153] defined as

$$g(a_1 z_1 + \cdots + a_{d'} z_{d'}) = g(\langle \mathbf{a}, \mathbf{z} \rangle),$$

with $g : \mathbb{R} \to \mathbb{R}$, $\mathbf{a} \in \mathbb{R}^{d'}$ and varies only in one direction given $\mathbf{a}$. Common examples of activation functions in literature are

- cosine squasher

$$\sigma(x) = \frac{\cos(x + \frac{3\pi}{2}) + 1}{2} \mathbb{1}_{x \in [-\frac{\pi}{2}, \frac{\pi}{2}]} + \mathbb{1}_{x \in (\frac{\pi}{2}, \infty)}, \tag{0.3.4}$$

- logistic squasher

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}},$$

- sigmoidal squashing function: toute fonction croissante telle que

$$\sigma(x) = \begin{cases} 1, & x \to \infty \\ 0, & x \to -\infty \end{cases}, \tag{0.3.5}$$

- exponential Linear Unit (eLU)

$$\sigma_\alpha(x) = \begin{cases} \alpha(\exp(x) - 1) & , \quad x < 0 \\ x, & , \quad x \geq 0, \end{cases}$$

- Rectifier Linear Unit (ReLU)

$$\sigma(x) = \max(x, 0). \tag{0.3.6}$$

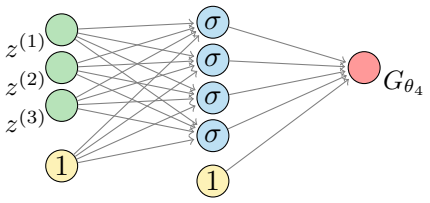


Figure 11: One-hidden layer NN with $K = 4$ neurons and $d' = 3$.

During the end of the XXth century, several authors studied the ability of $\mathbf{z} = (z_1, \ldots, z_{d'}) \in \mathbb{R}^{d'} \mapsto G_{\theta_K}(\mathbf{z})$ to approximate a given function G in different norms as $K \to \infty$, if θ_K is well chosen. Such results are aggregated in what is called the Universal approximation theorem.

Universal approximation theorem. Let us recall some of the main results based on the reference paper [152].

▷ (Gallant, White - 1988)[73]: If G is a square integrable function on $[0, 2\pi]^{d'}$ then there exists a "Fourier NN" G_{θ_K} composed by cosine squasher activation functions σ (0.3.4) which converges uniformly to G.

▷ (Cybenko - 1989)[46]: If G is a continuous function in the d'-dimensional unit cube $[0, 1]^{d'}$, then there exists a NN G_{θ_K} composed by sigmoidal activation functions (0.3.5) which converges uniformly to G.

▷ (Hornik - 1990)[110]: Similar result proved for any continuous function G, denoted $G \in \mathcal{C}^0$, on a compact set with a bounded and non-constant activation function σ (more general than (0.3.5) but it does not include (0.3.6)). Another result holds for any function $G \in \mathbf{L}^p(\mathbb{R}^{d'}, \mathrm{d}\mathbf{z})$.

▷ (Pinkus - 1993)[126]: Similar result proved if and only if σ is not a polynomial (which includes (0.3.6)).

See the general result:

Theorem 0.3.2 (Universal approximation)**.** *Suppose G is a continuous function on a compact space $\mathcal{Z} \subset \mathbb{R}^{d'}$ and σ is not a polynomial, then $\forall \varepsilon > 0$, there exists a NN (0.3.3) such that*

$$\sup_{\mathbf{z} \in \mathcal{Z}} \left| G(\mathbf{z}) - \sum_{k=1}^{K} w_k^{(2)} \sigma \left(\left\langle \mathbf{w}_k^{(1)}, \mathbf{z} \right\rangle + b_k \right) \right| < \varepsilon.$$

We have just seen that Theorems 0.3.1 and 0.3.2 give sufficient conditions on $\mathcal{Z}, \mathcal{X}$ and σ for (0.3.2) to hold if G_θ is a NN. In other words, if G is not a continuous function with a compact support, then by no means it will be well approximated uniformly by a one-hidden layer NN. We will tackle this issue in Chapters 2 and 3.

Since we aim at providing theoretical results on approximation error bounds, we address the following question:

Question. If assumptions of Theorem 0.3.2 hold, what is the rate of convergence of G_{θ_K} to G for a given norm ?

Rate of convergence. As usually highlighted in the approximation theory, the rate of convergence depends on the regularity of the function G. Here we consider the regularity in the sense of its differentiability.

Definition 0.3.1 (Sobolev space)**.** Let $B^{d'}$ denote the unit ball in $\mathbb{R}^{d'}$ such that $B^{d'} = \left\{ \mathbf{z} \in \mathbb{R}^{d'} : \|\mathbf{z}\|_2 \leq 1 \right\}$. Define the Sobolev space $\mathcal{S}_p^m = \mathcal{S}_p^m \left(B^{d'} \right)$ containing all the continuous functions $G \in \mathbb{R}^{d'}$ on $B^{d'}$ and where the derivatives $D^k G$ exist and are continuous on $B^{d'}$, for all $|k| = \{0, \ldots, m\}$. with a norm defined by

$$\|G\|_{m,p} := \begin{cases} \left(\sum_{0 \leq |k| \leq m} \left\| D^k G \right\|_p^p \right)^{1/p}, & 1 \leq p < \infty \\ \max_{0 \leq |k| \leq m} \left\| D^k G \right\|_\infty, & p = \infty. \end{cases}$$

The following result is based on the bounds obtained in $\mathbf{L^2}$ ([131, Theorem 1]) and on the upper bound extension for all norms $\mathbf{L}^p$ with $1 \leq p \leq \infty$ ([130, Proposition 1] and [152, Corollary 6.4]) using a sigmoidal activation function (0.1.5).

Theorem 0.3.3 (Rate of convergence). *Let* $S_p^m = S_p^m(B^{d'}) = \left\{G \in \mathcal{S}_p^m : \|G\|_{m,p} \leq 1\right\}$. *For each* $p \in [1, \infty]$ *and for all* $d' \geq 2$ *and* $m \geq 1$, *there exists an activation function* σ *infinitely differentiable on* $\mathbb{R}$, *sigmoidal and strictly increasing such that*

$$\sup_{G \in S_p^m} \inf_{\theta_K \in \Theta_K} \|G - G_{\theta_K}\|_p \leq C K^{-\frac{m}{(d'-1)}},$$

with C *a constant independent of* K.

The above result shows that dividing the error by 2 leads to increase the number of neurons by at least $2^{\frac{(d'-1)}{m}}$, which explodes as $d' \to \infty$. This well known phenomenon is called the curse of dimensionality. However, there exists a way to overcome this curse ($\mathbf{L^2}$ error reduced to $C/\sqrt{K}$) by imposing that $\int \|\xi\| \, |\widehat{G}(\xi)| \, \mathrm{d}\xi < \infty$, where $\widehat{G}$ is the Fourier transform of G [15].

Multi-layer neural networks. In the early years of the XXIst century, authors started to focus on the benefit of increasing the number of hidden layers. Based on the Kolmogorov superposition theorem [107, 170], a first uniform convergence result was proved for a two-hidden layer NN [130, Theorem 4]. Few years later, the author in [189] established a convergence rate of the uniform error using a multi-layer NN with a ReLU activation function (0.3.6).

Theorem 0.3.4. *For any* d', m *and* $\varepsilon \in [0, 1]$, *there exists a deep ReLU NN that can uniformly approximate any function* $G \in S_p^m([0,1]^{d'})$ *with an error* ε *composed by at least*

- $C\left(\log(1/\varepsilon) + 1\right)$ *hidden layers,*
- $C\varepsilon^{-d'/m}\left(\log(1/\varepsilon) + 1\right)$ *neurons,*

with some constant $C = C(d', m)$.

Although the depth complexity in the previous result has been improved for continuous functions by a log-term in [190], the main width complexity term $\varepsilon^{-d'/m}$ remains similar to the one-hidden layer case.

Question. If G_θ is a NN and conditions of Theorem 0.3.2 are satisfied, how to train θ such that (0.3.2) holds ?

Among the different methods in NN generative modeling for training θ, we focus on GANs since they have the right structure that suits us for providing approximation results.

Generative Adversarial Network (GAN)

Proposed by [100], a GAN scheme is aimed at approximating the unknown G through a parametric family of NNs $\mathcal{G} = \{G_\theta : \mathbb{R}^{d'} \to \mathbb{R}^d, \theta \in \Theta\}$ and to learn the optimal parameter $\theta^\star$ from a data set $\{X_i \in \mathbb{R}^d, i \in \{1, \ldots, n\}\}$ of i.i.d samples from the unknown distribution p_X. It is performed by optimizing an objective function which can be interpreted as an adversarial game between a generator and a discriminator chosen in a parametric family of functions $\mathcal{D} = \{D_\phi : \mathbb{R}^d \to [0,1], \phi \in \Phi\}$. In other words, $D_\phi(x)$ represents the probability that an observation x is drawn from p_X. Both the generator and the discriminator are NNs with opposite objectives: the former tries to mimic real data which seem likely by the discriminator, while the latter tries to distinguish between the two sources. See Figure 12 for an illustration. In [100], this optimization problem is defined as:

$$\mathcal{L}(\theta, D_\phi) := \min_{\theta \in \Theta} \max_{D_\phi \in \mathcal{D}} \left[\mathbb{E}_{p_X}\left(\log D_\phi(X)\right) + \mathbb{E}_{p_Z}\left(\log\left(1 - D_\phi\left(G_\theta(Z)\right)\right)\right)\right].$$

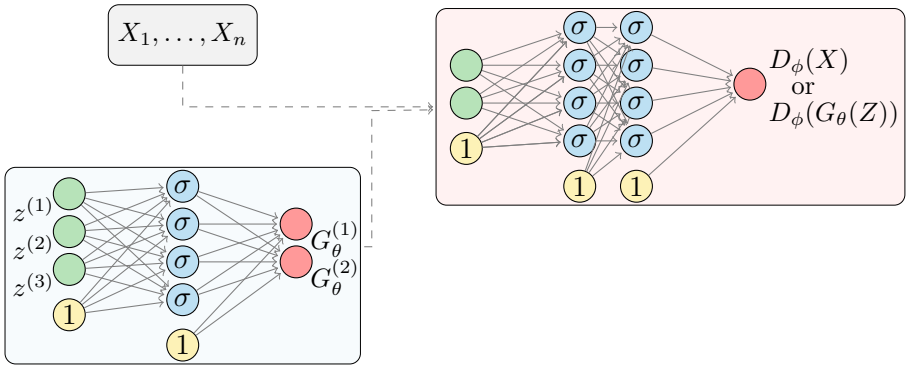


Figure 12: GAN model with $d' = 3$ and $d = 2$.

Theoretical results. In this thesis, we neither focus on the statistical error nor on the equilibrium of the adversarial game between the generator and the discriminator. Here we recall a sketch of existing results, where more details and full proofs can be found in [22]. Consider a wider class $\mathcal{D}_\infty = \left\{D : \mathbb{R}^d \to [0,1]\right\}$, the optimal discriminator given any fixed generator G_θ is $D_\theta^\star = \frac{p_X}{p_X + p_\theta}$ [100, Proposition 1], so it entails [100, Theorem 1] and

$$\begin{aligned}\sup_{D \in \mathcal{D}_\infty} \mathcal{L}(\theta, D) = \mathcal{L}(\theta, D_\theta^\star) &= -\log(4) + 2\mathtt{D}_{\mathtt{JS}}(p_X || p_\theta) \\ &= -\log(4), \quad \text{iff } p_X = p_\theta,\end{aligned}$$

where

$$\mathtt{D}_{\mathtt{JS}}(p_X || p_\theta) := \int_{\mathcal{X}} \frac{1}{2} p_X(x) \log\left(\frac{2p_X(x)}{p_X(x) + p_\theta(x)}\right) + \frac{1}{2} p_\theta(x) \log\left(\frac{2p_\theta(x)}{p_X(x) + p_\theta(x)}\right) \pi(\mathrm{d}x)$$

stands for the Jensen-Shannon (JS) divergence. Additionally, it is known that $D_\theta^\star = \arg\max_{D \in \mathcal{D}_\infty} \mathcal{L}(\theta, D)$ is unique if $\pi(p_X = p_{\theta^\star} = 0) = 0$[22, Theorem 2.1]. Clearly if $p_X \in \mathcal{P}$ then $p_X = p_{\theta^\star}$ and $D_\theta^\star = 1/2$, *i.e.* the optimal discriminator classifies all the data with probability $1/2$ because it is no longer able to distinguish between the real and the generated data. However, usually $p_X \notin \mathcal{P}$ because of the restricted parameterized nature of $\mathcal{G}$ and $\mathcal{D}$. Moreover, under convexity and compactness conditions, [22, Theorem 2.2] shows the existence and uniqueness of the best parameter $\theta^\star$ approaching the unknown density under the JS-divergence:

$$\theta^\star = \arg\min_{\theta \in \Theta} \mathcal{L}(\theta, D_\theta^\star) = \arg\min_{\theta \in \Theta} \mathtt{D}_{\mathtt{JS}}(p_X || p_\theta).$$

Restricted to the parameterized class $\mathcal{D}$ and suppose the new optimal associated generator's parameters $\bar{\theta} \in \Theta$ exist such that $\sup_{D_\phi \in \mathcal{D}} \mathcal{L}(\bar{\theta}, D_\phi) \leq \sup_{D_\phi \in \mathcal{D}} \mathcal{L}(\theta, D_\phi), \forall \theta \in \Theta$, then [22, Theorem 3.1] states that if the class $\mathcal{D}$ is rich enough to approximate the discriminator $D_{\bar{\theta}}^\star = \arg\max_{D_\phi \in \mathcal{D}} \mathcal{L}(\bar{\theta}, D_\phi)$ in the $\mathbf{L^2}$ sense by a distance no more than $\varepsilon > 0$, then

$$\mathtt{D}_{\mathtt{JS}}(p_X || p_{\bar{\theta}}) - \mathtt{D}_{\mathtt{JS}}(p_X || p_{\theta^\star}) = \mathcal{O}\left(\varepsilon^2\right).$$

This result highlights the argument that as the discriminator class gets richer (i.e. more neurons and hidden layers), the closer the restricted density $p_{\bar{\theta}}$ will be from the unrestricted one $p_{\theta^\star}$. In addition, with more regularity assumptions, [22, Theorem 4.1] gives a statistical error of the best empirical parameter $\hat{\theta}$ as

$$\mathbb{E}\mathtt{D}_{\mathtt{JS}}\left(p_X || p_{\hat{\theta}}\right) - \mathtt{D}_{\mathtt{JS}}\left(p_X || p_{\theta^\star}\right) = \mathcal{O}\left(\varepsilon^2 + \frac{1}{\sqrt{n}}\right).$$

Finally, the convergence of the best empirical parameters $\hat{\theta}$ and $\hat{\alpha}$ and the asymptotic normality of $\hat{\theta}$ are given respectively in [22, Theorem 4.2] and [22, Theorem 4.3]. Subsequently, other architectures and distances have been proposed, in particular the well-known Wasserstein GAN [10], where theoretical results can be found in [24] and [103].

We are now in a position to introduce the fields of study considered during this thesis: continuous time-series (Section 0.3.2) and extremes (Section 0.3.3).

0.3.2 Continuous time-series

Among the huge and expanding literature on GANs, lot of works studied the ability to generate time-series (in a discrete time), either in finance [186], in medicine [64] or in meteorology [103], for citing only some of them. Therefore, in order to provide precise quantitative guidelines on G_θ and p_Z on the case X is a continuous time-process, we decided to focus on the fractional Brownian motion (fBm) $\left\{B^H(t)\right\}_{t\in\mathbb{R}}$ with a Hurst parameter $H \in (0,1)$, including the standard Brownian motion ($H = 1/2$) as a particular case. A fBm is a centered Gaussian process with a covariance function

$$\mathbb{C}\mathrm{ov}\left(B^H(t), B^H(s)\right) = \frac{V_H}{2}\left(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\right), \text{ for any } \quad s,t \geq 0,$$

with $V_H = \mathbb{V}\mathrm{ar}[B^H(1)]$. The motivation in choosing such a model for our study is threefold. First, its stochastic simulation is known to be quite delicate (at least for $H \neq 1/2$), especially when the number of time points gets larger and larger – see [39, 56] and [13, Section 11.6] for a review and [40, 120] for recent contributions – hence having at hand a generative model for the full path is really appealing for practical use. Second, it is widely used in various real-life modelings: uni and bipedal postural standing in biomechanics [30], volatility of financial assets [43, 84], vortex filament structures observed in 3D fluids [70]; prices of electricity in a liberated market [17], solar cycle [154]. For other fractional-based modeling, see [41]. Third, understanding the right design of G_θ for generating the fBm distribution may well open the way to handle more complicated stochastic models written as a Stochastic Differential Equation (SDE) driven by fBm for instance.

As B^H is a centered Gaussian process in a Banach space $(\mathcal{C}^0([0,1],\mathbb{R}), \|.\|_\infty)$ (see [124, Proposition 3.6]), B^H admits almost sure (a.s.) series representation of the form

$$B^H(t) = \sum_{k=0}^{\infty} u_k(t) Z_k, \quad \forall t \in [0,1], \tag{0.3.7}$$

where $\{u_k\}_{k\in\mathbb{N}}$ is a sequence of continuous non-random functions, and $\{Z_k\}_{k\in\mathbb{N}}$ is a sequence of independent standard Gaussian variables $\mathcal{N}(0,1)$. Equality (0.3.7) holds in the sense that the series converges a.s. uniformly.

Question. Which class of functions u_k may be considered to ensure that (0.3.7) holds?

In [142] the authors proposed several wavelet series expansions of fBm in the form of (0.3.7) where $\{u_k\}_{k\in\mathbb{N}}$ depend on a) the choice of the wavelet basis and b) the description of the low frequency terms. We will take advantage of one of the wavelet series, where some optimality results have been established in [14].

Challenge. Clearly a product operation in (0.3.7) is required between the inputs $\{Z_k\}_{k\in\mathbb{N}}$ (*i.e.* the latent space in a GAN setting) and the functions $\{u_k\}_{k\in\mathbb{N}}$. Since such an operation is not natively done in a feedforward network (0.3.3), we will need to study how to approximate it on top of the NN approximation of $\{u_k\}_{k\in\mathbb{N}}$. Another point of attention is that once trained,

the NN must be effective for all time $t \in [0,1]$ unlike other known discretized fBm simulation methods [13, Section 11.6].

0.3.3 Extremes

We start by reminding some prerequisites in extreme value theory. As part of a more exhaustive reading, one can refer to [52] for the theoretical aspects and to [42] for the applications.

Extreme value theory

Given i.i.d. random variables $X_1, \dots, X_n$ with associated c.d.f F and quantile function $q(\cdot) := \inf\{x \in \mathbb{R} : F(x) \geq \cdot\}$, the extreme value theory is concerned with the limit distribution behavior of the sample maxima $X_{n,n} = \max(X_1, \dots, X_n)$ as $n \to \infty$. Clearly this distribution can be derived exactly for all n since

$$\mathbb{P}(X_{n,n} \leq x) = F^n(x).$$

Although F is unknown, the idea is to approximate F^n by a family of distributions based on the largest observations. However, to avoid degenerate limit distributions, *i.e.* $F^n(x) \to 0$ for $x < x^\star$ and $F^n(x) \to 1$ for $x \geq x^\star$ as $n \to \infty$ with $x^\star = \sup\{x : F(x) < 1\}$ the right endpoint, a normalization is required assuming that there exist a sequence $a_n > 0$ (location) and $b_n \in \mathbb{R}$ (scale) such that the variable

$$\frac{X_{n,n} - b_n}{a_n}$$

has a non-degenerate limit distribution, *i.e.* for all $x \in \mathbb{R}$

$$\lim_{n\to\infty} F^n(a_n x + b_n) = H(x). \tag{0.3.8}$$

We say for the class of distributions F satisfying (0.3.8) that F belongs to the maximum domain of attraction of H, denoted for short by $F \in \mathrm{MDA}(H)$.

Question. Which class of functions H may be considered to ensure that (0.3.8) holds ?

The answer is provided by the following well-known extreme-value Theorem [69, 89].

Theorem 0.3.5. *The class of extreme value distributions is $H_\gamma(ax+b)$ with $a > 0, b \in \mathbb{R}$ where*

$$H_\gamma(x) = \begin{cases} \exp\left(-(1+\gamma x)_+^{-1/\gamma}\right), & \gamma \neq 0, \\ \exp(-x), & \gamma = 0. \end{cases}$$

The parameter γ is called the extreme value index and we distinguish three subclasses of distributions with different tail behaviors:

- $\gamma > 0 : F \in \mathrm{MDA}$ (Fréchet) [63, Section 3.3.1],
- $\gamma < 0 : F \in \mathrm{MDA}$ (Weibull) [63, Section 3.3.2],
- $\gamma = 0 : F \in \mathrm{MDA}$ (Gumbel) [63, Section 3.3.3].

The characterization of domains of attraction relies on the theory of regularly-varying functions [26].

Definition 0.3.2 (Regularly varying)**.** A measurable function $f : \mathbb{R}^+ \to \mathbb{R}^+$ is regularly varying with index $\delta \in \mathbb{R}$ at infinity if for all $x > 0$

$$\lim_{t\to\infty} \frac{f(xt)}{f(t)} = x^\delta.$$

This property is denoted by $f \in \mathcal{RV}_\delta$.

One can show that any function $f \in \mathcal{RV}_\delta$ can be written as

$$f(t) = t^\delta L(t),$$

where $L \in \mathcal{RV}_0$ is called a slowly varying function with

$$\lim_{t\to\infty} \frac{L(xt)}{L(t)} = 1, \tag{0.3.9}$$

for all $x > 0$. In the following we will focus on the domain of attraction of Fréchet, in which heavy-tailed distributions belong to. It is characterized by a polynomial decay of the survival function $\bar{F} := 1 - F$.

Theorem 0.3.6 ([63, Theorem 3.3.7])**.** *A c.d.f.* $F \in \mathrm{MDA}$*(Fréchet) if and only if* $\bar{F} \in \mathcal{RV}_{-1/\gamma}$ *with* $\gamma > 0$*.*

Equivalently, it is known [52, Theorem 1.2.1, Proposition B.1.9.9] that the tail quantile function $U(t) := q(1 - 1/t) \in \mathcal{RV}_\gamma$ for all $t > 1$, *i.e.*

$$U(t) = t^\gamma L(t), \tag{0.3.10}$$

where $L \in \mathcal{RV}_0$. In the context of generative modeling for heavy-tailed distributions, we are interested in the estimation of (0.3.10) for large values of t.

Quantile estimation

Among the different approaches in extreme quantile estimation, we focus on the semi-parametric one based on (0.3.10). The latter is composed by a parametric part t^γ which only depends on the tail-index $\gamma > 0$, and a non-parametric part $L(t)$ where L satisfies (0.3.9). We are interested in the estimation of the quantile function (0.3.10), at the extreme level $1 - \alpha_n$ *i.e.* such that $n\alpha_n \to 0$ as $n \to \infty$. This latter condition entails that $q(1 - \alpha_n)$ is almost surely asymptotically larger than the sample maxima. The idea underpinning the estimation is to take advantage of (0.3.10) to establish the link between the extreme quantile of interest $U(1/\alpha_n) = q(1 - \alpha_n)$ and an intermediate one $U(1/\delta_n) = q(1 - \delta_n)$ where δ_n is interpreted as an anchor level with $k := \lfloor n\delta_n \rfloor \to \infty$ as $n \to \infty$ such that

$$q(1 - \alpha_n) = q(1 - \delta_n)(\delta_n/\alpha_n)^\gamma \exp\left(\varphi\Big(\log(\delta_n/\alpha_n), \log(1/\delta_n)\Big)\right), \tag{0.3.11}$$

with for all $(x_1, x_2) \in \mathbb{R}_+^2$,

$$\varphi(x_1, x_2) := \log\left(\frac{L(\exp(x_1 + x_2))}{L(\exp(x_2))}\right). \tag{0.3.12}$$

Three terms in (0.3.11) have to be estimated:

1. the intermediate quantile $q(1 - \delta_n)$,
2. the tail index γ,

3. the function φ.

The most well known estimator based on the the semi-parametric approach is the Weissman estimator [184] which consists in choosing 1) the order statistic $X_{n-k+1,n}$, 2) the Hill estimator $\hat{\gamma}(\cdot)$ [108] and 3) $\varphi(\cdot,\cdot) = 0$ such that

$$\hat{q}(1-\alpha_n; 1-\delta_n) := X_{n-k+1,n}(\delta_n/\alpha_n)^{\hat{\gamma}(k)},$$

where

$$\hat{\gamma}(k) := \frac{1}{k}\sum_{i=1}^{k-1} \log X_{n-i+1,n} - \log X_{n-k+1,n}.$$

Instead, (0.3.12) can be evaluated using the second-order condition which states there exist $\gamma > 0, \rho < 0$ and a function A positive or negative with $A(t) \to 0$ as $t \to \infty$ such that for all $x \geq 1$ [93, Equation (13)]

$$\log U(tx) - \log U(t) = \gamma \log x + A(t)\int_1^x y^{\rho-1}\,\mathrm{d}y + o(A(t)), \qquad \text{as } t \to \infty. \tag{0.3.13}$$

Moreover, $|A|$ is regularly-varying with index ρ. This second-order parameter drives the bias of most extreme quantile estimators: the larger ρ is, the larger the asymptotic bias. Assumption (0.3.13) is standard in extreme-value theory, since it controls the rate of convergence in (0.3.12).

Theorem 0.3.7 ([52, Theorem 4.3.8])**.** *Suppose*

- *the second order-condition* (0.3.13) *holds,*
- $k \to \infty, \delta_n \to \infty$ *and* $\sqrt{k}A(1/\delta_n) \to \lambda < \infty$ *as* $n \to \infty$*,*
- $n\alpha_n = o(k)$ *and* $\log(n\alpha_n) = o\left(\sqrt{k}\right)$ *as* $n \to \infty$*.*

Then as $n \to \infty$

$$\frac{\sqrt{k}}{\log(\delta_n/\alpha_n)}\left(\log \hat{q}(1-\alpha_n; 1-\delta_n) - \log q(1-\alpha_n)\right) \xrightarrow{d} \mathcal{N}\left(\frac{\lambda}{1-\rho}, \gamma^2\right).$$

Basing on the above asymptotic representation of the Weissman estimator, bias-reduced estimators have been introduced thanks to a prior estimation of ρ driving the first dominant bias component, see [94].

Challenge. A primary observation is that even in the univariate case ($d = d' = 1$), a feed-forward NN (0.3.3), with either a sigmoidal (0.3.5) or a ReLU (0.3.6) activation function, cannot efficiently approximate (0.3.10). First, the quantile function $u \in [0,1) \mapsto q(u)$ does not benefit from a compact support since it diverges as $u \to 1$ at a rate $(1-u)^{-\gamma}$, see Figure 13 for an illustration. Therefore the universal approximation theorem (Theorem 0.3.2) does not hold. Second, if Z is (as usually set) either bounded (Uniform) or has a light tailed distribution (Gaussian), then $G_\theta(Z)$ would be respectively bounded or light-tailed distributed. Conversely, we aim at generating an unbounded random variable with a heavy-tailed distribution.

0.3.4 Contributions of the thesis

After having introduced the context and the fields of study, we are now in a position to specify the main problem statement of the generative modeling part of this thesis.

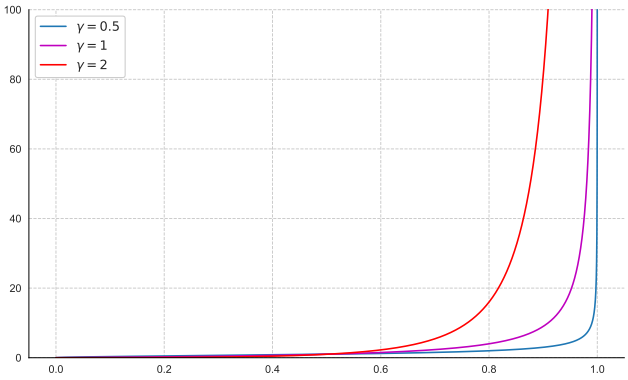


Figure 13: Quantile function associated with the Burr distribution $u \in [0,1) \mapsto q(u)$ with tail-index $\gamma \in \{1/2, 1, 2\}$ and $\rho = -1$.

Question. How to choose G_θ and the latent space $(\mathcal{Z}, \mu_Z)$ when

- $\mathcal{X}$ is the space of continuous functions indexed by time, equipped with the sup norm $d_\mathcal{X}$, and when the distribution of X is that of a stochastic process (infinite dimensional object), possibly non-Markovian?
- X has a heavy-tailed distribution (*i.e.* a polynomial decay of the survival function), and $G_\theta(Z)$ shall simulate extreme quantiles (*i.e.* at a risk level $1 - \alpha_n$ such that $n\alpha_n \to 0$ as $n \to \infty$) ?
- X is conditioned by another random variable Y ?

Contributions in Chapter 1. First we address the issue of simulating a continuous time-series with a generative NN. We provide a large probability bound on the uniform approximation of fBm with Hurst parameter $H \in (0,1)$, by a deep-feedforward ReLU NN fed by the latent distribution p_Z with N independent standard Gaussian random variables, with bounds on the network design (number of hidden layers and total number of neurons). We start with the case $H = 1/2$ (standard Brownian motion).

Theorem 0.3.8 (simplified version of [7, Thorem 2])**.** *Let $N \geq 2$ and $\left(\Omega^N, \mathcal{F}^N, \mathbb{P}^N\right)$ be a probability space supporting N i.i.d. standard Gaussian random variables $G_{1:N}$. Therefore, there exists an extension $(\Omega, \mathcal{F}, \mathbb{P})$ supporting a Brownian motion B such that $\forall p \in (0,1]$, there exist a ReLU neural network $\tilde{B}_{N,p}$ and a finite random variable $C \geq 0$ (independent from N and p) such that*

$$\mathbb{P}\left(\sup_{t\in[0,1]} \left|B(t) - \tilde{B}_{N,p}(t, G_{1:N})\right| \leq C N^{-1/2} \left(1 + \log N\right)^{1/2}\right) \geq 1 - p.$$

Additionally, $\tilde{B}_{N,p}$ is composed at most by

1. $\mathcal{O}_c\left(\log\left(\frac{N\rho_N}{(1+\log N)^{1/2}}\right)\right)$ *hidden layers,*
2. $\mathcal{O}_c\left(N \log\left(\frac{N\rho_N}{(1+\log N)^{1/2}}\right)\right)$ *neurons and parameters,*

with $\rho_N = -\Phi^{-1}(\frac{p}{2N})$ and Φ^{-1} the quantile function of the standard Gaussian distribution.

Then, we handle the general case of the fBm.

Theorem 0.3.9 (simplified version of [7, Thorem 3]). *Let $N \geq 2$ and $\left(\Omega^N, \mathcal{F}^N, \mathbb{P}^N\right)$ be a probability space supporting N i.i.d. standard Gaussian random variables $G_{1:N}$. Therefore, there exists an extension $(\Omega, \mathcal{F}, \mathbb{P})$ supporting a fractional Brownian motion B^H such that $\forall p \in (0,1]$, for all $r \in \mathbb{N}_0$ there exist a ReLU neural network $\tilde{B}^H_{N,p}$ and a finite random variable $C \geq 0$ (independent from N and p) such that*

$$\mathbb{P}\left(\sup_{t\in[0,1]} \left|B^H(t) - \widetilde{B}^H_{N,p}(t, G_{1:N})\right| \leq CN^{-H}\left(1+\log(N)\right)^{1/2}\right) \geq 1-p.$$

Additionally, $\widetilde{B}^H_{N,p}$ is composed by

1. $\mathcal{O}_c\left(\log\left(\frac{N\rho_N}{(1+\log(N))^{1/2}}\right)\right)$ *hidden layers,*
2. $\mathcal{O}_c\left(N^{1+\frac{H+1}{2r}}\log\left(\frac{N\rho_N}{(1+\log(N))^{1/2}}\right)\left(\frac{\rho_N}{(1+\log(N))^{1/2}}\right)^{\frac{1}{2r}}\right)$ *neurons and parameters,*

where ρ_N is defined in Theorem 0.3.8. The constants in $\mathcal{O}_c(\cdot)$ may depend on r and H.

Essentially, our results state that for a given latent dimension N, there is a $G_\theta \in \mathcal{G}$ such that equality (0.3.1) holds with an error $N^{-H}\left(1+\log(N)\right)^{1/2}$ in sup norm with probability $1-p$. Moreover, focusing on the rates with respect to $N \to +\infty$, the depth of G_θ is at most

$$\mathcal{O}\left(\log N\right)$$

and its global complexity is

$$\mathcal{O}\left(N^{1+\zeta}\log N\right),$$

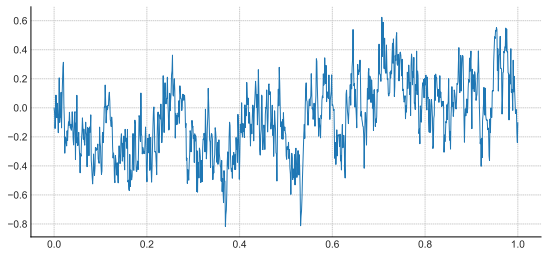
where ζ is a positive parameter that can be taken as small as desired, and where the $\mathcal{O}(\cdot)$ depend on p, ζ and H. In particular for the Brownian motion ($H = 1/2$) we can take $\zeta = 0$. A more detailed dependence on p, ζ and H is given latter. Our analysis relies, in the standard Brownian motion case ($H = 1/2$), on the Levy construction and in the general fractional Brownian motion case ($H \neq 1/2$), on the Lemarié-Meyer wavelet representation. The latter has been implemented in order to train our fBm NN $\widetilde{B}^H_N$ and is illustrated in Figure 14, confirming that both the regularity paths and the dependence structure of our NN are well preserved.

These results are original to the best of our knowledge, and should play a key role in tuning GAN-based methods in the choice of the parametric family of NNs for generating fractional stochastic processes in continuous time. Moreover, these results make a clear connection between the time-regularity of the path (that could be measured on the real observed data) and the architecture of the parameterization to set up.

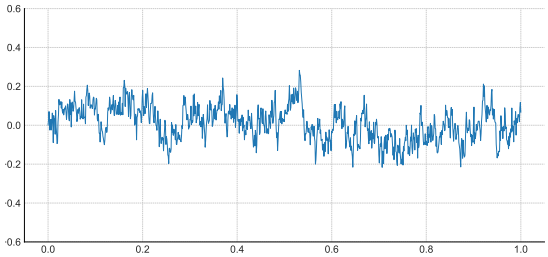
In the context of generative modeling for heavy tailed distributions, we are interested in the tail behavior of the NN generative models. We have just highlighted their main issues, which makes it a difficult subject that is little covered theoretically in the literature.

Contributions in Chapter 2. We propose a new parametrization of GANs adapted to the learning of heavy-tailed distributions with a support restricted to the one in the training dataset. Taking advantage of (0.3.10), we introduce a tail-index function

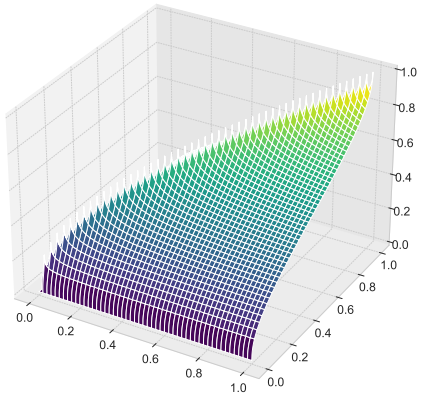
$$u \mapsto f^{\mathrm{TIF}}(u) := -\frac{\log q(u)}{\log\left(\frac{1-u^2}{2}\right)}, \tag{0.3.14}$$



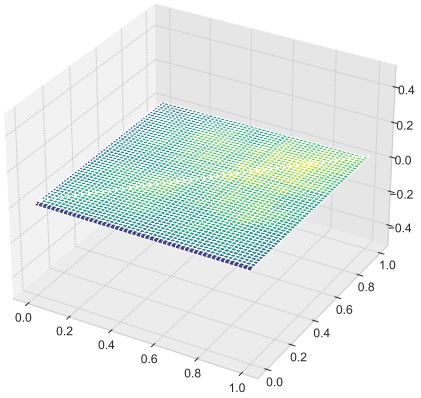
(a) $H = 0.2, \widehat{H} = 0.209$



(b) $\widetilde{H} = 0.208$



(c)



(d)

Figure 14: Simulation of fBm series (top) and covariance surface (bottom) for $H = 0.2$.
Top left: wavelet representation $t \mapsto V_H^{-1/2} B_N^H(t)$ with the estimated Hurst index $\widehat{H}$.
Top right: error $t \mapsto V_H^{-1/2}(B_N^H(t) - \widetilde{B}_N^H(t))$ with the estimated Hurst index $\widetilde{H}$ on the NN fBm $\widetilde{B}_N^H$.
Bottom left: real normalized function $(t,s) \mapsto V_H^{-1}\mathbb{C}\mathrm{ov}(B^H(t), B^H(s))$.
Bottom right: error $(t,s) \mapsto V_H^{-1}(\mathbb{C}\mathrm{ov}(B^H(t), B^H(s)) - \mathbb{C}\mathrm{ov}(\widetilde{B}_N^H(t), \widetilde{B}_N^H(s)))$ for $(t,s) \in [0,1]^2$.

which is continuous, bounded on $[0,1]$ and tends to the tail index γ as $u \to 1$, see Figure 15a. Although it allows to respect the compact support assumption required by the universal approximation Theorem, correction terms are added in (0.3.14) to form a Corrected Tail-index function

$$u \mapsto f^{\mathrm{CTIF}}(u) := f^{\mathrm{TIF}}(u) - \sum_{k=1}^{6} \kappa_k e_k(u),$$

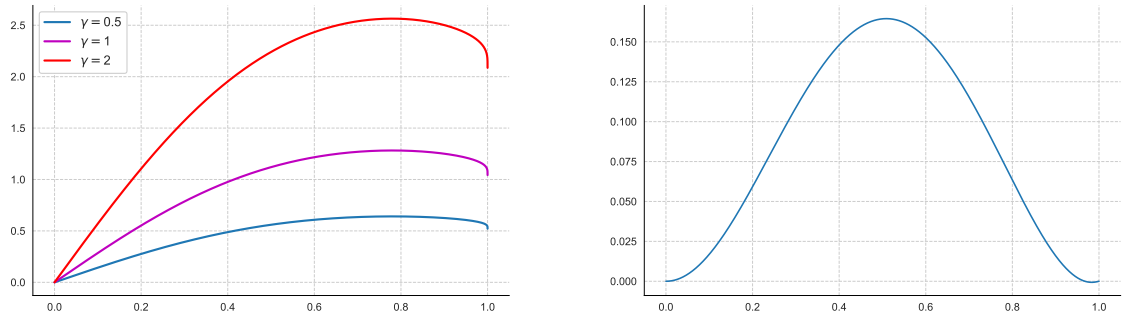
illustrated in Figure 15b and where functions $e_1, \dots, e_6$ and coefficients $\kappa_1, \dots, \kappa_6$ are defined in Section 2.2.2. Such a transformation allows to improve the regularity of (0.3.14) and therefore also the convergence rate of the approximation error by the NN.

Theorem 0.3.10 (simplified version of [6, Thorem 4])**.** *Assume both the first and the second-order conditions hold under some additional regularity assumptions (see Section 2.2.1). Let σ be a ReLU function. For all $J \geq 6$, there exist $(a_j, w_j, b_j) \in \mathbb{R}^3$, $j \in \{1, \dots, J\}$ such that:*

$$\sup_{u \in [0,1]} \left| f^{\mathrm{CTIF}}(u) - \sum_{j=1}^{J} a_j \sigma\left(w_j u + b_j\right) \right| = \mathcal{O}\left(J^{-\alpha-1}\right),$$

where

1. $\alpha \in (0, -1-\rho)$ *if* $-2 \leq \rho < -1$,



(a) $\gamma \in \{1/2, 1, 2\}$ and $\rho = -1$ (b) $\gamma = 1/2$ and $\rho = -3$

Figure 15: Tail-index function (a) and Corrected Tail-index function (b) associated with a Burr distribution.

2. $\alpha = 1$ *if* $\rho < -2$.

Note that, for $\alpha = 1$, the above rate cannot be improved in general, owing to [189, Theorem 6]. Moreover, the previous approximation result can be interpreted in terms of Wasserstein-1 distance between the true data distribution and the simulated one. Indeed, in the univariate case, the Wasserstein-1 distance can be simplified as

$$W_1(q, \widetilde{q}) = \int_0^1 |q(u) - \widetilde{q}(u)| \, \mathrm{d}u,$$

where $u \mapsto \widetilde{q}(u)$ is the EV-GAN approximation of the unknown quantile function $u \mapsto q(u)$.

Corollary 0.3.1 (simplified version of [6, Corollary 5]). *Assume conditions of Theorem 0.3.10 hold with* $\gamma < 1$ *and* $\rho < -1$. *Then, the Wasserstein-1 distance can be controlled as* $W_1(q, \widetilde{q}) = \mathcal{O}\left(J^{-\alpha-1}\right)$.

Note that $\gamma < 1$ is a necessary condition for the Wasserstein-1 distance to exist. The above approximation bounds on f^{CTIF} (Theorem 0.3.10) can be translated in terms of approximation bounds on $f^{\mathrm{TIF},(m)}$ using an "enriched" NN for all m-th components of a d-dimensional random variable with $m \in \{1, \ldots, d\}$ and a latent dimension $d' \geq d$.

Corollary 0.3.2 (simplified version of [6, Corollary 6]). *Let* σ *be a ReLU function. Let* $X = (X^{(1)}, \ldots, X^{(d)})^\top$ *be a* d*-dimensional vector, with each component* $X^{(m)}$ *fulfilling conditions of Theorem 0.3.10 with parameters* $(\gamma^{(m)}, \rho^{(m)})$. *Let* $\mathcal{G}_J^{d',d}$ *be the approximation space of TIF functions made of* $J \geq 6$ *neurons. Then,*

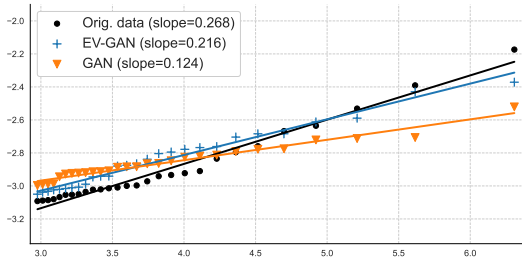
$$\inf_{G \in \mathcal{G}_J^{d',d}} \sup_{m \in \{1,\ldots,d\}} \sup_{z \in [0,1]^{d'}} \left| f^{\mathrm{TIF},(m)}(z^{(m)}) - G^{(m)}(z) \right| = \mathcal{O}\left(J^{-\alpha-1}\right),$$

where

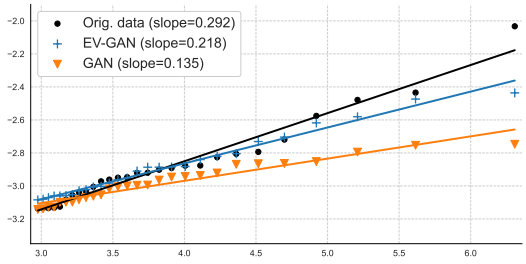
(i) $\alpha \in (0, -1 - \max_{m \in \{1,\ldots,d\}} \rho^{(m)})$ *if* $-2 \leq \rho^{(m)} < -1$ *for some* $m \in \{1, \ldots, d\}$,

(ii) $\alpha = 1$ *if* $\rho^{(m)} < -2$ *for all* $m \in \{1, \ldots, d\}$.

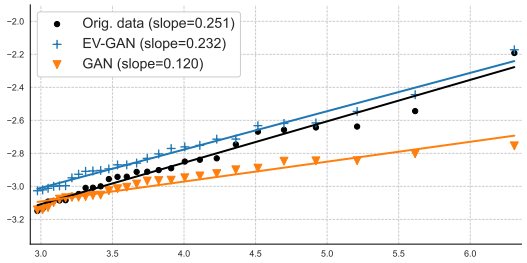
Observe that the worst second-order parameter $\rho^{(m)}$, *i.e.* the closest to -1, tunes the global accuracy of the EV-GAN through the convergence order $\alpha + 1$. Thus, we establish that the rate of convergence of the error is mainly driven by the second-order parameter of the data distribution. The results are illustrated on both simulated heavy-tailed distributions and real



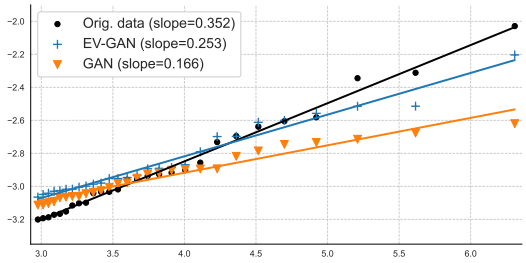
(a) AEX



(b) CAC



(c) KOSPI



(d) NDQ

Figure 16: Log quantile-quantile plots $\log((n+1)/i) \mapsto \log X^{(j)}_{n-i+1,n}$, for $i \in \{1, \dots, \lceil (1-\xi)n \rceil\}$ and $j \in \{1, \dots, 4\}$ at probability level $\xi = 0.95$ on four financial indices: AMX (Amsterdam Exchange, Netherlands), CAC (France), KOSPI (Korea), Nasdaq (USA).

financial stock indexes (see Figure 16) in comparison with a classical NN generative model. It is shown that, in both experiments, our approach largely outperforms the classical method.

Contributions in Chapter 3. We extend the estimation of (0.3.10) a) outside the training dataset support and b) taking into account a covariate associated with the extreme quantile. The latter correspond respectively to a non-conditional and a conditional NN quantile extrapolation estimators. Unlike most bias corrected estimators which focus only on the second order representation of (0.3.12), we show that the J-th order condition ($J \geq 2$) benefits from a natural NN representation with the popular eLU activation function which allows to better approximate the log-spacing function

$$(x_1, x_2) \in \mathbb{R}_+^2 \mapsto \gamma x_1 + \varphi(x_1 + x_2), \tag{0.3.15}$$

illustrated in Figure 17. Based on this result, we derive a NN extreme quantile estimator which features an automatic estimation and removal of all J first bias terms.

Theorem 0.3.11 (simplified version of [5, Theorem 2])**.** *Assume the J-th order condition holds ($J \geq 2$). Then, their is a one hidden-layer NN approximation of the extreme quantile $q(1-\alpha_n)$ such that*

$$\inf_{\tilde{\phi} \in \Phi} \left| \log q(1-\alpha_n) - \log \tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1-\alpha_n; 1-\delta_n) \right| = \mathcal{O}\left(\alpha_n^{-\bar{\rho}_J}\right),$$

where $\bar{\rho}_J = \rho_2 + \cdots + \rho_J$ as $\alpha_n \to 0$ and $\delta_n/\alpha_n \to \infty$ when $n \to \infty$.

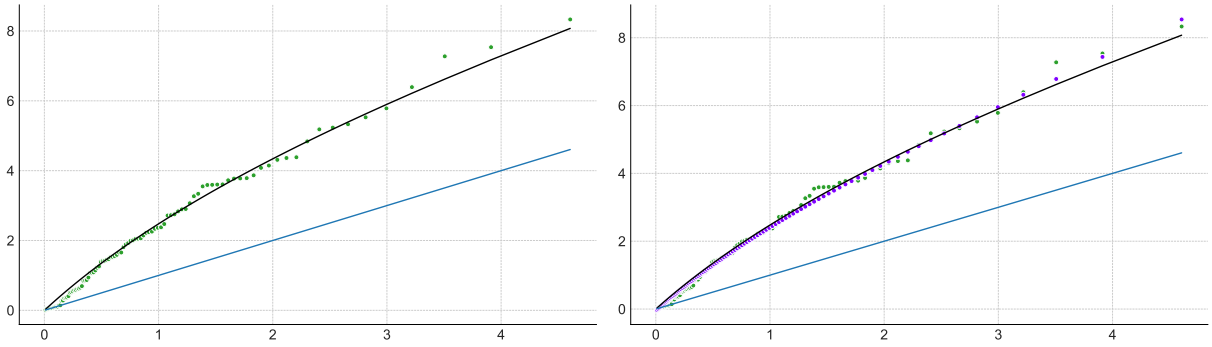


Figure 17: Log-spacing function associated with a Burr distribution ($\gamma = 1, \rho = -1/4$). Black curve: theoretical function $x_1 \mapsto \gamma x_1 + \varphi(x_1 + \log(n/k))$, blue line: Weissman approximation $x_1 \mapsto \gamma x_1$, green dots: empirical pointwise estimation $(\log(k/i), \log X_{n-i+1,n} - \log X_{n-k+1,n})$, purple dots: NN estimation with $i \in \{1, \dots, k-1\}$, $k = 100$ and $n = 500$.

Observe that the convergence rate of the uniform error between extreme log-quantiles and their NN approximation is driven mainly by the extreme level α_n and the sum of all j-th order parameters, $j \in \{2, \dots, J\}$. As expected, requesting higher regularity in the extreme-value model (through the J-th order condition) yields a smaller approximation error thanks to an increasing width of the proposed NN. The numerical analysis shows that compared with other bias reduced estimators on simulated data, the non-conditional NN estimator outperforms them in difficult heavy-tailed situations where other competitors almost all fail.

Suppose now that X is a random variable associated with an explanatory random vector $Y \in \mathcal{Y} \subset \mathbb{R}^{d_y}, d_y \geq 1$. We present two approaches to estimate conditional extreme quantiles by a NN. The first one is the conditional extension of the non-conditional one discussed above.

Theorem 0.3.12 (simplified version of [5, Theorem 3]). *Assume a conditional extension of the J-th order condition holds ($J \geq 2$). Additionally, suppose all functions included in the conditional extension of the log-spacing function are continuous on the compact set $\mathcal{Y} \subset \mathbb{R}^{d_y}$. Let $\bar{\rho}_{\texttt{sup}} = \sup_{y \in \mathcal{Y}} \bar{\rho}_J(y)$ with $\bar{\rho}_J(y) = \rho_2(y) + \cdots + \rho_J(y)$. Then, there exists a deep conditional NN approximation of the conditional extreme quantile $q(1 - \alpha_n \mid y)$ including $2J(J-1)+1$ sub-networks built with fixed $2d_y + 10$ number of neurons in each of the hidden layers with a depth at least of order $\alpha_n^{\bar{\rho}_{\texttt{sup}}/2}$ such that*

$$\inf_{\tilde{\phi} \in \Phi} \sup_{y \in \mathcal{Y}} \left| \log q(1 - \alpha_n \mid y) - \log \tilde{q}_{\tilde{\phi}}^{\texttt{NN}_J}(1 - \alpha_n; 1 - \delta_n \mid y) \right| = \mathcal{O}\left(\alpha_n^{-\bar{\rho}_{\texttt{sup}}}\right),$$

as $\alpha_n \to 0$ and $\delta_n / \alpha_n \to \infty$ when $n \to \infty$.

In this general conditional setting, a minimum depth (of magnitude $\simeq \alpha_n^{\bar{\rho}_{\texttt{sup}}/2}$) is required for the first conditional extrapolation NN to approximate the extreme quantile with a given error (of order $\simeq \alpha_n^{-\bar{\rho}_{\texttt{sup}}}$) while, in the previous situation, a one layer NN was sufficient. The second conditional extrapolation NN takes advantage of a location-dispersion model assumption [177] to get rid of the covariate in the extrapolation step.

Theorem 0.3.13 (simplified version of [5, Theorem 4]). *Assume the location-dispersion model assumption holds under the conditions of Theorem 0.3.11 with additional bounding and continuity assumptions. Then, there exists a one hidden-layer neural network approximation of the conditional extreme quantile $q(1 - \alpha_n \mid y)$ such that*

$$\inf_{\tilde{\phi} \in \Phi} \sup_{y \in \mathcal{Y}} \left| \log q(1 - \alpha_n \mid y) - \log \tilde{q}_{\tilde{\phi}}^{\texttt{NN}_J}(1 - \alpha_n; 1 - \delta_n, 1 - \tau_n \mid y) \right| = \mathcal{O}(\alpha_n^{-\bar{\rho}_J}) + \mathcal{O}(\tau_n^{-\bar{\rho}_J - \gamma} \delta_n^{\gamma}) \tag{0.3.16}$$

with $\alpha_n \to 0$, $\delta_n/\tau_n \to 0$ *and* $\delta_n/\alpha_n \to \infty$ *as* $n \to \infty$.

It is then possible to tune the value of the additional sequence δ_n to balance both error terms in (0.3.16):

Corollary 0.3.3 (simplified version of [5, Corollary 5]). *Assume the assumptions of Theorem 0.3.13 hold.*

- *If* $\gamma + \bar{\rho}_J > 0$, *then letting* $\delta_n = \alpha_n^{-\bar{\rho}_J/\gamma}\tau_n^{1+\bar{\rho}_J/\gamma}$ *yields*

$$\inf_{\tilde{\phi}\in\Phi}\sup_{y\in\mathcal{Y}}\left|\log q(1-\alpha_n \mid y) - \log \tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1-\alpha_n; 1-\delta_n, 1-\tau_n \mid y)\right| = \mathcal{O}(\alpha_n^{-\bar{\rho}_J}).$$

- *If* $\gamma + \bar{\rho}_J \leq 0$, *then letting* $\delta_n = \xi_n\alpha_n$ *and* $\tau_n = \xi_n^2\alpha_n$ *with* $\xi_n \to \infty$ *arbitrarily slowly as* $n \to \infty$ *yields*

$$\inf_{\tilde{\phi}\in\Phi}\sup_{y\in\mathcal{Y}}\left|\log q(1-\alpha_n \mid y) - \log \tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1-\alpha_n; 1-\delta_n, 1-\tau_n \mid y)\right| = \mathcal{O}(\alpha_n^{-\bar{\rho}_J}\xi_n^{-2\bar{\rho}_J-\gamma}).$$

Up to the ξ_n term, one can recover the convergence rate $\alpha_n^{\bar{\rho}_J}$ of the unconditional case, see Theorem 0.3.11. Both conditional models are implemented to investigate the behavior and the spatial interpolation of extreme rainfalls as functions of their geographical location in the southern part of France (see Figure 18).

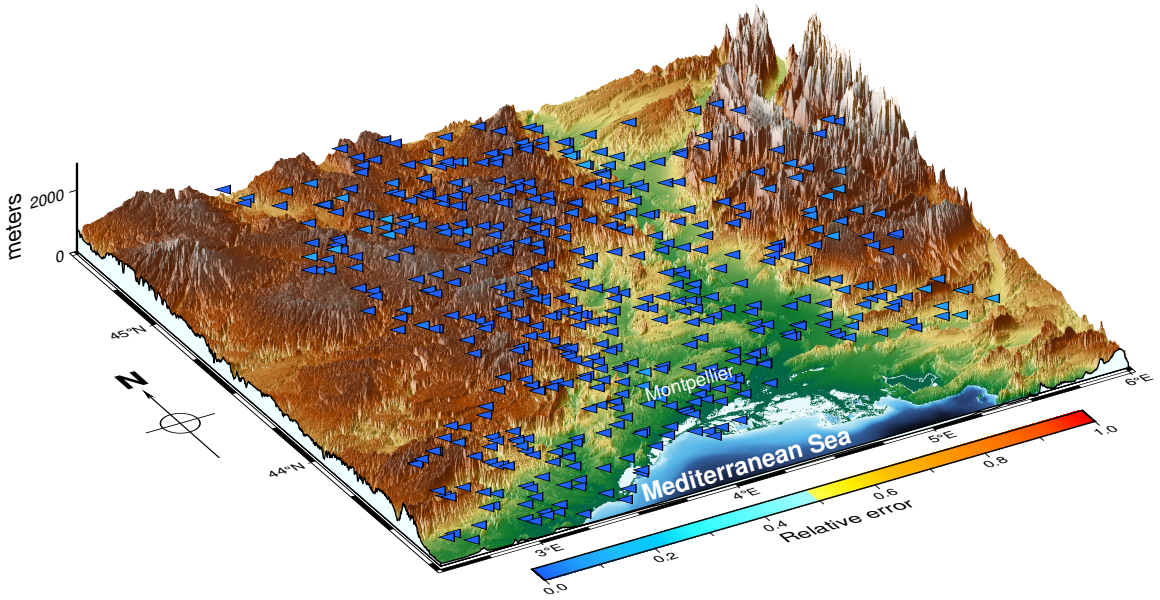


Figure 18: Estimation of one of the conditional extreme quantile NN model at each station.

0.4 Dictionary learning of rating migration matrices

0.4.1 Context

Credit risk refers to the risk of incurring losses due to unexpected changes in the credit quality of the counterparty. The successive financial crises during the XXth century have led the banking supervisors to focus on such a risk and have resulted in the creation of a solvency ratio, called *Cooke ratio*, which was at the heart of the first international banking regulation (Basel I). Despite these initial efforts, the evolution of the banking business and of the financial crises has required a thorough review of the regulatory framework (Basel II and III). According to the

latter, banks can use internal ratings and risk exposure estimations in order to assess regulatory capital requirement and credit risk measures (Value at Risk, Expected Shortfall, ...).

Credit risk is summarized in a structured rating migration matrix which captures all possible transition probabilities that an obligor will migrate from a credit state to another over a given time period (see Figure 19). See [11, 25, 37, 141] for extensive references on risk measures and credit risk. Rating migration matrices (RMM) are key indicators to assess credit risk portfolio through the estimation of the credit quality of the obligors.

Rating allocation process includes models and expert systems taking into account obligor's specific features evolving over time given the economic situation. In practice, we observe at time t a one-year rating migration matrix $\mathbf{P}^t \in \mathbb{R}^{R-1} \otimes \mathbb{R}^R$, which encodes the probability of migrating from rating $i \in [R-1] := \{1, \dots, R-1\}$ to rating $j \in [R]$ within one year period starting at time $t-1$, where R represents the default state; in Figure 19 we have $R = 11$. The reconstruction of this matrix is made empirically by evaluating the frequencies of obligors going from the rating i to rating j between times $t-1$ and t. Observed migration frequencies are displayed into RMM that are the cornerstone of rating migration models upon which credit risk portfolio simulation relies.

One-factor Copula model. The most widely used method for modeling RMM is the one-factor Gaussian copula model [127]. The latter belongs to the class of "threshold models", which are very popular in credit risk modeling [141, Section 11.1]. See [27] among others for estimating risk measures on the loss distribution of a large credit risk portfolio under this model.

For an initial rating i at time t, the one-factor Copula model assumes that the event "migration to the rating j at time $t+1$" is given by

$$\left\{X_i^t \in [c_{i,j}, c_{i,j+1})\right\},$$

where the parameters $\{c_{i,j}\}_{i\in[R-1],j\in[R]}$ are thresholds triggering the rating migration and with a stochastic factor

$$X_i^t = \rho Z^t + \sqrt{1-\rho^2}\epsilon_i^t,$$

composed by a systemic risk $Z^t \overset{d}{=} \mathcal{N}(0,1)$ (common to all obligors), an idiosyncratic risk $\epsilon_i^t \overset{d}{=} \mathcal{N}(0,1)$ (specific to every obligor), independent from Z^t, and a correlation parameter $\rho \in (-1,1)$ between the two sources of risk. Equivalently,

$$\begin{aligned}
&\{\text{"migration to the rating greater than } j \text{ at time } t+1\text{"}\} \\
&\quad = \bigcup_{k\geq j} \{X_i^t \in [c_{i,k}, c_{i,k+1})\} \\
&\quad = \{X_i^t \geq c_{i,j}\} = \{\rho Z^t + \sqrt{1-\rho^2}\epsilon_i^t \geq c_{i,j}\}.
\end{aligned}$$

Conditionally to Z^t, this event has the probability

$$\Phi\left(\frac{\rho Z^t - c_{i,j}}{\sqrt{1-\rho^2}}\right) =: P_{i,\geq j}^t := P_{i,j}^t + \cdots + P_{i,R}^t, \tag{0.4.1}$$

and the unconditional probability

$$\Phi(-c_{i,j}) =: P_{i,\geq j}^{\mathrm{TTC}}, \tag{0.4.2}$$

where Φ is the c.d.f. of a standard Gaussian distribution. The matrix $\mathbf{P}^{\mathrm{TTC}}$ represents the probability of default *'Through the Cycle'* which is a long-run average over a cycle and focuses

mainly on permanent components of default risk, whereas the matrix $\mathbf{P}^t$ represents the probability of default *'Point-In-Time'* which takes into account both cyclical and permanent effect. Inverting $c_{i,j}$ in (0.4.2) and replacing in (0.4.1) gives

$$P^t_{i,\geq j} := \Phi\left(\frac{\rho Z^t + \Phi^{-1}(P^{\mathrm{TTC}}_{i,\geq j})}{\sqrt{1-\rho^2}}\right),$$
$$P^t_{i,j} := \Phi\left(\frac{\rho Z^t + \Phi^{-1}(P^{\mathrm{TTC}}_{i,\geq j})}{\sqrt{1-\rho^2}}\right) - \Phi\left(\frac{\rho Z^t + \Phi^{-1}(P^{\mathrm{TTC}}_{i,\geq j+1})}{\sqrt{1-\rho^2}}\right),$$

with the convention $P^{\mathrm{TTC}}_{i,\geq R+1} = 0$. In order to simulate some $\mathbf{P}^t$, one can consider the systemic factor $(Z^t)_{t\in[T]}$ evolving as a stationary Auto-Regressive (of order 1) process

$$Z^t = \kappa Z^{t-1} + \epsilon^t_z,$$

where the innovation noise ϵ^t_z is independent in t and distributed as $\mathcal{N}(0, 1-\kappa^2)$ such that the stationary distribution of $(Z^t)_{t\geq 1}$ is $\mathcal{N}(0,1)$ with $\kappa \in (-1,1)$.
The popularity of the Gaussian copula model is due to the ease of use but it also suffers from too simple underlying hypothesis. These weak assumptions lead to miscapture the dependence structure in the tails. However, and despite post subprime crisis criticisms [129], the one factor Gaussian copula model remains very popular in the banking industry because of its parsimony and of its ability to generate intuitive and interpretable results even with small amount of data available.

In this context, we wanted to develop a non-parametric representation of RMM with a data-based approach as an alternative and a challenger to the parametric Gaussian Copula model. In the following, we expose the matrix factorization approach we used, raise some important challenges and discuss our contributions.

	1	2	3	4	5	6	7	8	9	10	11
1	71.48	17.87	5.36	2.38	1.25	0.71	0.43	0.26	0.15	0.08	0.03
2	16.01	57.62	14.41	5.69	2.80	1.54	0.89	0.52	0.30	0.16	0.06
3	5.05	15.14	53.85	13.46	5.89	3.03	1.68	0.96	0.54	0.28	0.11
4	2.45	6.52	14.67	51.35	12.84	5.87	3.06	1.68	0.92	0.47	0.18
5	1.45	3.61	7.23	14.46	49.57	12.39	5.74	2.95	1.55	0.76	0.30
6	0.96	2.31	4.34	7.71	14.46	48.19	12.05	5.51	2.71	1.29	0.48
7	0.70	1.63	2.94	4.90	8.16	14.69	47.00	11.75	5.14	2.28	0.82
8	0.54	1.24	2.18	3.48	5.44	8.71	15.24	45.71	11.43	4.51	1.52
9	0.45	1.02	1.75	2.73	4.09	6.14	9.55	16.38	43.67	10.92	3.28
10	0.43	0.95	1.60	2.43	3.55	5.11	7.46	11.36	19.17	38.35	9.59

Figure 19: Representation of an idealized rating migration matrix of size 10×11. All values are in percentage. The credit quality goes from the highest (rating 1) to the lowest (rating 10), the default is 11.

0.4.2 Matrix factorization

It is important for risk management purposes to model the evolution of $\mathbf{P}^t$, by finding a representation of the type

$$\mathbf{P}^t \approx \sum_{k=1}^{K} \alpha^t_k \mathbf{d}_k, \quad \forall t \geq 1,$$

for some so-called (deterministic) basis vectors $\mathbf{d}_k$, which we must constraint their representation to be RMM, and for some scalar random coefficients α_k^t, which we should model the evolution. In a matrix form (using the vec () operator to simplify, see Section 4.1.5) for the definition of the notations, we say that the collection of vectorized matrices $\mathbf{P} = \{\text{vec}\left(\mathbf{P}^t\right) \in \mathbb{R}^d\}_{t=1}^T \in \mathbb{R}^d \otimes \mathbb{R}^T$, with $d := (R-1)R$ for all $t \in \{1, \dots, T\}$, admits a matrix factorization over a dictionary $\mathbf{D} \in \mathbb{R}^d \otimes \mathbb{R}^K$ composed by K elements (called atoms), if there exists a linear combination of atoms weighted by coefficients (called codings) $\mathbf{A} = \{\boldsymbol{\alpha}^t \in \mathbb{R}^K\}_{t=1}^T \in \mathbb{R}^K \otimes \mathbb{R}^T$ such that

$$\mathbf{P} \approx \mathbf{DA}. \tag{0.4.3}$$

Over the last years, a new paradigm of data-based models have emerged in the ML community in order to extract structured information from high-dimensional objects. A classical approach in ML is to use Matrix Factorization techniques in order to project the data in some relevant basis. It is well known that the optimal basis that minimizes the linear approximation error is the Karhunen-Loève basis [137, Theorem 9.8], also known as the principal components in the principal component analysis (PCA). However it is important to highlight that this basis will not be able to satisfy the linear constraints imposed by the RMM structure.

Data constraints

A rating migration matrix $\mathbf{M}$ must satisfy some constraints for both mathematical and economical reasons.

Mathematical constraints. Each row of $\mathbf{M}$ is a discrete probability, hence $\mathbf{M}$ is a stochastic matrix. The set of Stochastic Matrices is denoted by

$$\mathcal{M}^S := \Big\{ \mathbf{M} \in \mathbb{R}^{(R-1)} \otimes \mathbb{R}^R : \sum_{j \in [R]} M_{i,j} = 1, \forall i \in [R-1], \\ M_{i,j} \geq 0, \, \forall (i,j) \in [R-1] \times [R] \Big\}.$$

Economic constraints. Depending on their expertise, some risk managers may consider important to put additional constraints that are meaningful from economic point-of-view. For instance, the likelihood of default for higher-rated counterparties is lower than for the lower-quality ones. Then, the collection of rating matrices satisfying so-called economic constraints is denoted by

$$\mathcal{M}^E := \Big\{ \mathbf{M} \in \mathbb{R}^{(R-1)} \otimes \mathbb{R}^R : M_{i,\geq j} \leq M_{i',\geq j}, \\ \forall j \in [R], \quad 1 \leq i < i' \leq R-1 \Big\}.$$

A matrix satisfying such constraints is called an idealized matrix and is illustrated in Figure 19.

Question. How to achieve (0.4.3) while requiring $\mathbf{D}$ to satisfy the above linear constraints in order to represent economically interpretable RMM?

Introduced in [150], dictionary learning (DL), see [59] for an overview and [101] for theoretical results, is another matrix representation technique where the basis, called dictionary, is learned from the observations. Unlike in the PCA decomposition, neither the orthogonality nor the representation constraints of the basis vectors (atoms) are imposed, allowing more flexibility to adapt the desired representation to the data. Moreover, compared with a predefined dictionary like Gabor functions, wavelets or local cosine vectors [137], learning a dictionary adapted to the observations has shown better results in practice [62, 134].

In DL, the linear approximation (0.4.3) is usually coupled with a regularization criterion $\mathcal{R}(\mathbf{A})$ applied to the codings and yields to the general optimization problem

$$\min_{\mathbf{D},\mathbf{A}} \|\mathbf{P} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda\mathcal{R}(\mathbf{A}), \quad \lambda \geq 0, \tag{0.4.4}$$

where the regularization term shall reflect the expected codings representation, see [59, Chapter 4]. The most widely studied regularization is $\mathcal{R}(\mathbf{A}) = \|\mathbf{A}\|_1$ referring to the so-called sparse coding (see [132] for an overview), where the optimization with respect to $\mathbf{A}$ is known as basis pursuit [36] or the Lasso [176]. DL with sparse representation was notably studied in image and video processing [133, 135, 136], in graph learning [179] and in clustering [171].

Question. How to achieve (0.4.3) while requiring the time series of elements $\boldsymbol{\alpha}^t$ of $\mathbf{A}$ to be smooth enough in order to perform predictions through a time series modeling?

The application of DL with a temporal structure has been mainly studied in video denoising [136, 156] where the temporal structure is exploited through an operator extracting patches of a fixed size in the objective function representing an energy minimization procedure. Another approach is to deal with an auto-regressive (AR) representation modeled either in the dictionary [38] or in the codings [191]. In the former, a mixed audio signal is decomposed into its constituent temporal sources (atoms of the dictionary) in order to detect the presence of a specific sound. In the latter, the authors present a framework which supports data-driven temporal learning and forecasting through an AR modelisation of the codings represented as a regularization term. Our model described in Section 4.2 is inspired from this problem formulation.

Challenge. Modeling RMM is a challenging problem because it is necessary to find an interpretable representation satisfying economic constraints, while the RMM evolution may vary quickly over time and a limited data history is available (usually 10-20 years $\approx$ 200 observations) which is close to the dimension of the problem (usually $R = 11$ and $d = 110$). Thus, modeling constrained RMM in a data-based non-parametric way presents an important challenge, which has not been addressed so far to the best of our knowledge.

0.4.3 Contributions of the thesis

In the context of structured DL of RMM for credit risk modeling, we are interesting to derive from the data a non parametric representation of RMM, as an alternative and a challenger to the parametric Gaussian copula model.

Contributions in Chapter 4. We propose a new RMM modelisation technique using a DL approach. The later takes into account linear constraints imposed by the RMM structure and encourages the codings $\mathbf{A}$ to have an AR structure through a temporal regularization term in (0.4.4):

$$\mathcal{R}_{AR}(\mathbf{A}, \mathbf{w}) := \sum_{k=1}^{K} \sum_{t=1}^{T-1} \left(\alpha_k^{t+1} - \frac{1}{T}\alpha_k^t - w_k \left(\alpha_k^t - \frac{1}{T}\alpha_k^t \right) \right)^2, \tag{0.4.5}$$

where the extra parameter $\mathbf{w}$ allows us to estimate the AR parameters of the time-series $\boldsymbol{\alpha}_k =: \mathbf{A}_{k,:}$ for each $k \in [K]$, for both interpretable clustering and prediction of RMM. We consider a dimensionality reduction framework $K \ll d$ in order to work in a lower dimensional space with extracted meaningful information. The propose DL problem is

$$\min_{\substack{\mathbf{D},\mathbf{A},\mathbf{w} \\ \mathbf{D}\in\Omega,\ \alpha_k^t \geq 0,\ t\in[T],\ k\in[K]}} \|\mathbf{P} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda\mathcal{R}_{AR}(\mathbf{A}, \mathbf{w}), \tag{0.4.6}$$

which is convex in variables $\mathbf{D}$, $\mathbf{A}$ and $\mathbf{w}$, when other are fixed; and where Ω is the convex set of dictionaries verifying the idealized constraints (see Section 0.4.2)

$$\Omega := \left\{ \mathbf{D} \in \mathbb{R}^d \otimes \mathbb{R}^K : \text{vec}^{-1}(\mathbf{d}_k) \in \mathcal{M}^E \cap \mathcal{M}^S, \ \forall k \in [K] \right\},$$

with $\text{vec}^{-1}(.)$ the inverse vectorize function (see Section 4.1.5).
Dictionary update. We opt for a sequential update of each atom of the dictionary: $\mathbf{d}_k$ for $k \in [K]$. This choice is guided by two advantages: 1. The problem is strictly convex for each atom $\mathbf{d}_k$ (as stated in Proposition 0.4.1 below) which is not necessarily true for the whole matrix $\mathbf{D}$. 2. This strategy breaks the problem in smaller problems making the resolution less dependent on the amount of atoms K. Updating atoms separately is also the strategy of the widely used K-SVD (see [59, Section 3.5]), however, the purpose in that case is to find a closed form for the optimization problem, which is not true in our case of study because of the form of constraints.

Proposition 0.4.1 (simplified version of [8, Proposition 2.1])**.** *Assume that $\{\alpha_k^t\}_{t=1}^T$ is non zero. The minimization of (0.4.6) over $\mathbf{d}_k \in \text{vec}\left(\mathcal{M}^E \cap \mathcal{M}^S\right)$ is equivalent to minimizing a strictly convex quadratic problem with linear constraints*

$$\min_{\mathbf{d}_k} \left\| \text{vec}\left(\widetilde{\mathbf{P}}_k\right) - \widetilde{\mathbf{A}}_k \mathbf{d}_k \right\|_F^2, \quad s.t.\ vec^{-1}(\mathbf{d}_k) \in \mathcal{M}^E \cap \mathcal{M}^S,$$

where $\widetilde{\mathbf{P}}_k$ and $\widetilde{\mathbf{A}}_k$ are explicitly defined (see (4.2.9)).

Codings update. Similarly to the dictionary update, we adopt a strategy based on the update of each $\mathbf{A}_{k,:}$ for $k \in [K]$. The reasons are the same: it is preferable to solve a smaller and strictly convex optimization problem. The fact that the optimization for each k is a strongly convex problem is not straightforward and is argued in the proposition below.

Proposition 0.4.2 (simplified version of [8, Proposition 2.2])**.** *Let $k \in [K]$ be fixed. Consider the minimization of (0.4.5)-(0.4.6) over one coding $\mathbf{A}_{k,:}$, i.e.*

$$\min_{\mathbf{A}_{k,:},\ A_{k,t} \geq 0} \|\mathbf{P} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda \sum_{k=1}^{K} \sum_{t=1}^{T-1} \left(A_{k,t+1} - \bar{\mathbf{A}}_{k,:} - w_k (A_{k,t} - \bar{\mathbf{A}}_{k,:}) \right)^2,$$

where $\bar{\mathbf{A}}_{k,:} := 1/T \sum_{t=1}^T \alpha_k^t$. For any $\lambda \geq 0$, the above problem is a strongly convex quadratic optimization problem with linear constraints.

Coefficient update. We note that, for each $k \in [K]$, the optimization problem with respect to $\mathbf{w}_k$ in equation (0.4.6) is a 1-dimensional quadratic problem with an explicit solution (see (4.2.5)).

To show the model applicability, we present a numerical test with real data which enjoys good accuracy for reconstruction and includes the supervised classification of the RMM based on both a K-means algorithm and the representation of the trained $\mathbf{D}$. We observe that the resulting estimated economic sentiment indicator is in line with historical economic events (see Figure 20). Finally, our DL approach significantly outperforms the widely used Gaussian Copula model, and therefore appears to be an efficient alternative model.

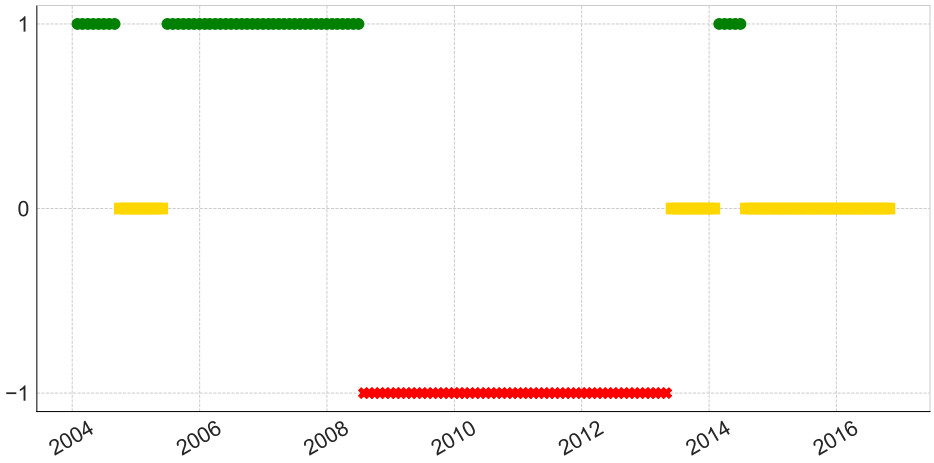


Figure 20: Classification of real RMM in 3 economic sentiments: $\{-1$ (red cross): bad, 0 (yellow square): stable, 1 (green dot): good$\}$.

Part I

Generative modeling

Chapter 1

A generative model for fBm with deep ReLU neural networks

Note. The results of this chapter are based on the paper [7].

Abstract. We provide a large probability bound on the uniform approximation of fractional Brownian motion with Hurst parameter H, by a deep-feedforward ReLU neural network fed with a N-dimensional Gaussian vector, with bounds on the network design (number of hidden layers and total number of neurons). Essentially, up to log terms, achieving an uniform error of $\mathcal{O}\left(N^{-H}\right)$ is possible with $\log(N)$ hidden layers and $\mathcal{O}\left(N \log N\right)$ parameters. Our analysis relies, in the standard Brownian motion case ($H = 1/2$), on the Levy construction and in the general fractional Brownian motion case ($H \neq 1/2$), on the Lemarié-Meyer wavelet representation. This work gives theoretical support on new generative models based on neural networks for simulating continuous-time processes.

1.1 Introduction

Over last few years a new paradigm of generative model has emerged in the new machine learning community with the goal of sampling high-dimensional complex objects (such as images, videos or natural language) from a data set of these objects. If X denotes the random variable taking values in a general metric space $(\mathcal{X}, d_{\mathcal{X}})$ from which we have observations $(X_i)_{i\geq 1}$, the problem of generative model construction amounts to finding a function $G_\theta : \mathbb{R}^N \mapsto \mathcal{X}$ and a latent probability distribution μ on $\mathbb{R}^N$ such that

$$X \overset{d}{=} G_\theta(Z) \text{ and } Z \sim \mu. \tag{1.1.1}$$

Usually, the choice of the dimension N (the so-called latent dimension) is part of the problem. The function G_θ belongs to a parametric family of functions $\mathcal{G} = \{G_\theta\}_{\theta\in\Theta}$, and it is common to consider neural networks: in this work, we follow this approach. Essentially, two main questions have to be addressed to obtain a generative model: a) how to choose $\mathcal{G}$ to have a chance to get the equality in distribution (1.1.1), or at least a good approximation of it for some $G_\theta \in \mathcal{G}$? b) how to learn the parameter θ from the data set? The second question b) has been tackled by [100] in their seminal work of Generative Adversarial Network (GAN). We will not focus on that problematic in this work, there is a tremendous number of works (about 30,000 citations of [100] on Google Scholar at the date of writing this article). Instead, we are to focus on a), *i.e.* quantifying how to choose $\mathcal{G}$ and the latent space (N, μ) when $\mathcal{X}$ is the space of continuous functions indexed by time, equipped with the sup norm $d_{\mathcal{X}}$, and when the distribution of X is that of a stochastic process (infinite dimensional object), possibly non-Markovian.

Among the huge and expanding literature on GANs, lot of works studied the ability to generate time-series (in a discrete time), either in finance [186], in medicine [64] or in meteorology [103], for citing only some of them. However, to the best of our knowledge, none of them is dealing with continuous-time processes. Moreover, designing the architecture of a neural network G_θ with respect to its depth (number of hidden layers), size (number of neurons), type (feed-forward, recurrent, convolutional, etc.) and activation functions (sigmoid, ReLU, etc.), is a very difficult question and therefore often left to empirical grid search. In this work, we aim at tackling these aspects and providing precise quantitative guidelines on $\mathcal{G}$ in the case where X is a fractional Brownian motion (fBm) with Hurst parameter $H \in (0,1)$ including standard Brownian motion ($H = 1/2$) as a particular case.

A fBm is a centered Gaussian process with a specific covariance function [138], detailed definition and properties are given in Section 1.2. Remind that almost all sample paths of fBm are α-Hölder-continuous for any $\alpha < H$, see [128, Corollary 4.2] for the thorough statement. The motivation in choosing such a model for our study is threefold. First, its stochastic simulation is known to be quite delicate (at least for $H \neq 1/2$), especially when the number of time points gets larger and larger – see [39, 56] for a review and [40, 120] for recent contributions – hence having at hand a generative model for the full path is really appealing for practical use. Second, it is widely used in various real-life modelings: uni and bipedal postural standing in biomechanics [30]; volatility of financial assets [43, 84]; vortex filament structures observed in 3D fluids [70]; prices of electricity in a liberated market [17]; solar cycle [154]; for other fractional-based modeling, see [41]. Third, understanding the right design of $\mathcal{G}$ for generating the fBm distribution may well open the way to handle more complicated stochastic models written as a Stochastic Differential Equation (SDE) driven by fBm for instance: indeed, as we will see, the design of the current $\mathcal{G}$ inherits much from the time-regularity of X and this property is lifted to SDE driven by X. This part is left to further investigation.

In this work we study the required depth (number of hidden layers) and complexity (number of neurons and parameters) of a deep-feedforward neural network (NN) for $\mathcal{G}$, with a Rectified Linear Unit (ReLU) for the activation function [98, Chapter 6]: it is referred to as ReLU NN in the sequel. For the latent distribution μ, we consider N independent components and without loss of generality for the simulation purpose, each of them is taken as a standard Gaussian random variable. Essentially, our results state (Theorems 1.2.1 and 1.2.2) that for a given latent dimension N, there is a $G_\theta \in \mathcal{G}$ such that equality (1.1.1) holds with an error $N^{-H}\left(1+\log(N)\right)^{1/2}$ in sup norm with probability $1-p$. Moreover, focusing on the rates with respect to $N \to +\infty$, the depth of G_θ is at most

$$\mathcal{O}\left(\log N\right)$$

and its global complexity is

$$\mathcal{O}\left(N^{1+\zeta}\log N\right),$$

where ζ is a positive parameter that can be taken as small as desired, and where the $\mathcal{O}\left(.\right)$ depend on p, ζ and H. In particular for the Brownian motion ($H = 1/2$) we can take $\zeta = 0$. A more detailed dependence on p, ζ and H is given latter.

These results are original to the best of our knowledge, and should play a key role in tuning GAN-based methods in the choice of the parametric family of NN for generating fractional stochastic processes in continuous time. These results make a clear connection between the time-regularity of the path (that could be measured on the real observed data) and the architecture of the parameterization to set up.

This work is organized as follows. In Section 1.2, we recall few properties of fBm. Our approximations are based on wavelet decomposition and we will provide appropriate materials. Then we state our main quantitative results about depth and complexity of deep ReLU NN for generating fBm. Section 1.3 is devoted to the proofs. For pedagogical and technical reasons, we

start with the case $H = 1/2$ (standard Brownian motion) in Subsection 1.3.1; then we handle the general case of fBm in Subsection 1.3.2. Simulation of fBm with a NN approach is illustrated in Section 1.4. A few technical proofs and numerical illustrations are postponed to Appendix.

Notations: The set of naturals without zero is defined by $\mathbb{N}_0 := \{1, 2, \dots, n, \dots\}$ and $\mathbb{N} := \mathbb{N}_0 \cup \{0\}$; define the set $\mathcal{M} := \left\{2^{n+1}, n \in \mathbb{N}\right\}$; the vector of N standard Gaussian random variables $G_1, \dots, G_N$ is denoted by $G_{1:N}$; the imaginary number $\mathbf{i}^2 = -1$. We write $x = \mathcal{O}_c(y)$ if $|x| \leq c\,|y|$ for some positive constant c which, in the context where it is used, does not depend neither on the latent dimension N nor on the accuracy ε; usually y will be a not-small quantity ($y \geq 1$) as a polynomial or logarithmic function of N or/and ε^{-1} according to the context. Finally, we write $a_N \asymp b_N$ if there exists a constant $c \geq 1$ such that $\forall N \in \mathbb{N}_0$, $c^{-1} \leq a_N/b_N \leq c$.

1.2 Preliminaries and main results

1.2.1 About Fractional Brownian motion

Fractional Brownian motion (fBm) $\left\{B^H(t)\right\}_{t\in\mathbb{R}}$ with a Hurst parameter $H \in (0,1)$ is a Gaussian process, centered ($\mathbb{E}\left[B^H(t)\right] = 0$), with covariance function

$$\mathbb{C}\mathrm{ov}\left(B^H(t), B^H(s)\right) = \frac{V_H}{2}\left(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\right), \text{ for any } \quad s, t \geq 0, \tag{1.2.1}$$

with $V_H = \mathbb{V}\mathrm{ar}[B^H(1)]$. We call $B^H(\cdot)$ a standard fBm if $V_H = 1$. When $H = 1/2$, we will simply write B instead of $B^{1/2}$. Our aim is to approximate the distribution of B^H on a finite interval: owing to the self-similarity property of fBm [149, Proposition 2.1], we can consider, without loss of generality, the interval $[0,1]$, which is our setting from now on.

As B^H is a centered Gaussian process in a Banach space $(\mathscr{C}^0([0,1],\mathbb{R}), \|.\|_\infty)$ (see [124, Proposition 3.6]), B^H admits almost sure (a.s.) series representation of the form

$$B^H(t) = \sum_{k=0}^{\infty} u_k(t) G_k, \quad \forall t \in [0,1], \tag{1.2.2}$$

where $\{u_k\}_{k\in\mathbb{N}}$ is a sequence of continuous non-random functions, and $\{G_k\}_{k\in\mathbb{N}}$ is a sequence of independent standard Gaussian variables $\mathcal{N}(0,1)$. Equality (1.2.2) holds in the sense that the series converges a.s. uniformly. Such representations for fBm are studied in [142] using wavelets.

Let $H \in (0,1)$; [122] showed that there exists a sequence $\{u_k\}_k$ such that the L^2-truncation error is

$$\left(\mathbb{E}\left[\sup_{t\in[0,1]}\left|\sum_{k=N}^{\infty} u_k(t) G_k\right|^2\right]\right)^{1/2} \asymp N^{-H}(1+\log(N))^{1/2}; \tag{1.2.3}$$

in addition, the above convergence rate is optimal among all sequences $\{u_k\}_k$ for which (1.2.2) converges a.s. in sup-norm.

In [142] the authors focused on the a.s. uniform convergence on $[0,1]$ for different wavelet representations series (1.2.2) using a specific mother wavelet function ψ, and the authors of [14, Theorem 5] showed their optimality in the sense of (1.2.3). Not only ψ has to generate an orthonormal basis $\left\{\psi_{j,k}(x) = 2^{j/2}\psi\left(2^j x - k\right)\right\}_{(j,k)\in\mathbb{Z}^2}$ of $\mathbf{L^2}(\mathbb{R}, \mathrm{d}x) = \left\{f : \int_{-\infty}^{\infty} |f(x)|^2\,\mathrm{d}x < \infty\right\}$ [137, Theorem 7.3, p. 278], but also it must respect some other regularity properties discussed hereafter.

In the following, our convention is to write the Fourier transform and its inverse as

$$\widehat{f}(\xi) := \int_{-\infty}^{\infty} f(x)\mathrm{e}^{-\mathbf{i}x\xi}\,\mathrm{d}x, \qquad f(x) := \frac{1}{2\pi}\int_{-\infty}^{\infty} \widehat{f}(\xi)\mathrm{e}^{\mathbf{i}x\xi}\,\mathrm{d}\xi. \tag{1.2.4}$$

1.2.2 Brownian motion: wavelet representation and main result for NN generative model

A first well-known series representation is the so-called Lévy construction of the standard Brownian motion ($V_H = 1$, $H = 1/2$) obtained by plugging in (1.2.2) the basis functions

$$\psi_{j,k}^{\mathrm{FS}}(t) = 2^{j/2}\psi^{\mathrm{FS}}\left(2^j t - k\right), \qquad j \in \mathbb{N},\ k = 0, \dots, 2^j - 1, \tag{1.2.5}$$

where $\psi^{\mathrm{FS}}(x) = 2\left(x\mathbb{1}_{0\le x<1/2} + (1-x)\mathbb{1}_{1/2\le x\le 1}\right)$ is twice the antiderivative of the Haar mother wavelet [102]. The set $\{\psi_{j,k}^{\mathrm{FS}}\}_{j\in\mathbb{N},k=0,\dots,2^j-1}$ defines the Faber-Schauder (F-S) system [66, 164] and forms an orthogonal basis of $\mathbf{L^2}(\mathbb{R}, \mathrm{d}x)$. Thus, given $\{G_1, G_{j,k}\}_{j\ge 0,\, 0\le k<2^j}$ a sequence of independent standard Gaussian random variables $\mathcal{N}(0,1)$, the Lévy construction of the standard Brownian motion states that a.s. the truncated series

$$B^{(n)}(t) := G_1 t + \sum_{j=0}^{n}\sum_{k=0}^{2^j-1} 2^{-(j+1)}\psi_{j,k}^{\mathrm{FS}}(t)G_{j,k} \tag{1.2.6}$$

converges uniformly on $[0,1]$ to a Brownian motion B as $n \to \infty$ (see [172, Section 3.4]). We write $B_N := B^{(n)}$ with $N = 2^{n+1}$, to emphasize that (1.2.6) contains N scalar Gaussian random variables, which is consistent with the latent dimension discussed above. The next result quantifies the a.s. convergence rate of B_N to B, the proof is postponed to Section 1.A.1.

Lemma 1.2.1. *Let $N \in \mathcal{M}$. Then, there exists a finite random variable $C_{(1.2.7)} \ge 0$ such that almost surely*

$$\sup_{t\in[0,1]} |B(t) - B_N(t)| \le C_{(1.2.7)} N^{-1/2}\big(1 + \log(N)\big)^{1/2}. \tag{1.2.7}$$

The above result is somehow well-known and shows that it is enough to approximate with a high probability the function $t \mapsto B_N(t)$ by a ReLU NN with suitable architecture, which is the purpose of the following statement.

Theorem 1.2.1. *Let $N \ge 2$ and $\left(\Omega^N, \mathcal{F}^N, \mathbb{P}^N\right)$ be a probability space supporting N i.i.d. standard Gaussian random variables $G_{1:N}$. Therefore, there exists an extension $(\Omega, \mathcal{F}, \mathbb{P})$ supporting a Brownian motion B such that $\forall p \in (0,1]$, there exist a ReLU neural network*

$$\tilde{B}_{N,p} : \begin{cases} \mathbb{R}^N & \to \mathscr{C}^0([0,1], \mathbb{R}) \\ G_{1:N} := (G_1, \dots, G_N) & \mapsto \tilde{B}_{N,p}(\cdot, G_{1:N}) \end{cases}$$

and a finite random variable $C \ge 0$ (independent from N and p) such that

$$\mathbb{P}\left(\sup_{t\in[0,1]} \left|B(t) - \tilde{B}_{N,p}(t, G_{1:N})\right| \le C N^{-1/2}\left(1 + \log N\right)^{1/2}\right) \ge 1 - p. \tag{1.2.8}$$

Additionally, $\tilde{B}_{N,p}$ is composed at most by

1. $\mathcal{O}_c\left(\log\left(\frac{N\rho_N}{(1+\log N)^{1/2}}\right)\right)$ *hidden layers,*
2. $\mathcal{O}_c\left(N\log\left(\frac{N\rho_N}{(1+\log N)^{1/2}}\right)\right)$ *neurons and parameters,*

with $\rho_N = -\Phi^{-1}(\frac{p}{2N})$ and Φ^{-1} the quantile function of the standard Gaussian distribution.

The proof is postponed to Subsection 1.3.1. The finiteness of C means only $\mathbb{P}(C < +\infty) = 1$ and a careful inspection of the proof would show that C has finite polynomial moments at any order. This will be similar for the fBM-result of Theorem 1.2.2.

Remark 1. It is known that $\Phi^{-1}(u) \sim -\sqrt{-2\log u}$ as $u \to 0^+$, see [58]. Therefore we shall get equivalents of the architecture size, either as $p \to 0$ or as $N \to \infty$ (which results in $\rho_N \to \infty$ anyhow):

1. For a fixed p and as $N \to \infty$, $\rho_N/(1+\log N)^{1/2}$ tends to a constant so the depth and the complexity are respectively of order $\mathcal{O}_c(\log(N))$ and $\mathcal{O}_c(N \log N)$;

2. For a fixed N and as $p \to 0$, the impact on the network size is moderate since both depth and complexity are of order $\mathcal{O}_c(\log\log(1/p))$.

As a complement to the previous marginal asymptotics, the estimates of Theorem 1.2.1 allow to have p dependent on N: for instance, building a ReLU NN with an error tolerance of order $N^{-1/2}(1+\log N)^{1/2}$ with probability $1 - N^{-k}$ (for any given $k > 0$) can be achieved using a depth $\mathcal{O}_c(\log N)$ and a complexity $\mathcal{O}_c(N \log N)$.

The next two remarks apply both to the current Brownian motion case and to the fBM studied in next subsection.

Remark 2. One may wonder how to improve the rate of convergence of Theorem 1.2.1 in terms of complexity. As far as Lemma 1.2.1 is concerned, the convergence rate is optimal in the sense of [122] using a linear approximation w.r.t to the Gaussian inputs. In other words, there cannot be another fBm series expansion with a faster rate of convergence. But once we consider non-linear approximation (with spline functions for instance [45]), there is no reason that using N Gaussians, we cannot approximate (using a suitable NN) the fBm with an accuracy higher than $N^{-1/2}(1+\log(N))^{1/2}$. In particular, and even if it is already quite cheap, there is no reason for the NN depth to be of order $\log(N)$ as we propose. On the other hand, once the latent dimension N is fixed, clearly one cannot use less that $\mathcal{O}_c(N)$ parameters (no matter which NN is used), which is exactly what we propose (up to the logarithmic term).
Finally, taking $p = 0$ is not possible within our method of proof, it is an open question if another construction would allow $p = 0$.

Remark 3. From the GAN point of view, it is enough to know (like in Theorem 1.2.1) that such a ReLU NN can generate a Brownian motion without the knowledge of the NN parameters explicitly; indeed, the GAN optimization algorithm will retrieve the parameters.
Regarding the practical use of this generative model, notice that sampling a Brownian motion path boils down to sample $G_{1:N}$ (with N independent standard Gaussian variables), and then compute $\tilde{B}_{N,p}(t, G_{1:N})$ for all times t required by the situation at hand.

1.2.3 Fractional Brownian motion: wavelet representation and main result for NN generative model

Among the wavelet fBm series representations proposed in [142], we will focus on the following one

$$B^H(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} 2^{-jH} \left(\Psi_H\left(2^j t - k\right) - \Psi_H(-k)\right) G_{j,k}, \tag{1.2.9}$$

with

$$\widehat{\Psi}_H(\xi) := \frac{\widehat{\psi}(\xi)}{(\mathbf{i}\xi)^{H+1/2}}, \tag{1.2.10}$$

and

$$V_H = \mathbb{V}\mathrm{ar}\left[B^H(1)\right] = \frac{1}{2H\sin(\pi H)\Gamma(2H)}, \quad \text{(see Definition 1.2.1)}, \tag{1.2.11}$$

and where $\Gamma(\cdot)$ is the Gamma function. The proof of this representation is recalled in Appendix-Section 1.B for the convenience of the reader. One choice for the wavelet ψ is the Lemarié-Meyer wavelet [125] (see [137, Equations (7.52)-(7.53)-(7.85) and Example 7.10] for more details on its construction) defined by its Fourier transform

$$\widehat{\psi^{\mathrm{M}}}(\xi) := \mathrm{e}^{-\mathrm{i}\frac{\xi}{2}} \begin{cases} \sin\left(\frac{\pi}{2}\nu\left(\frac{3|\xi|}{2\pi}-1\right)\right), & \frac{2}{3}\pi \le |\xi| \le \frac{4}{3}\pi, \\ \cos\left(\frac{\pi}{2}\nu\left(\frac{3|\xi|}{4\pi}-1\right)\right), & \frac{4}{3}\pi \le |\xi| \le \frac{8}{3}\pi, \\ 0, & \text{otherwise}, \end{cases} \tag{1.2.12}$$

where $\nu : \mathbb{R} \to [0,1]$ is a smooth function satisfying

$$\nu(x) = \begin{cases} 0, & \text{if } x \le 0, \\ 1, & \text{if } x \ge 1, \end{cases} \quad \text{and} \quad \nu(x) + \nu(1-x) = 1. \tag{1.2.13}$$

Such properties allow to satisfy the quadrature conditions of the conjugate mirror filter [137, Subsection 7.1.3 p. 270] which specifies the scaling function in the construction of wavelet bases (see [137, Chapter 7] for a complete overview of wavelet bases analysis). Considering the truncated series of (1.2.9) over a specific set $\mathcal{I}_N$ containing at most N indices (j,k), the authors of [14, Section 5 p. 469] have shown that there exists a finite r.v. $C_{(1.2.14)} \ge 0$ such that

$$\sup_{t\in[0,1]} \left|B^H(t) - B^H_N(t)\right| \le C_{(1.2.14)} N^{-H} \left(1+\log(N)\right)^{1/2}. \tag{1.2.14}$$

In other words, if Ψ_H is well chosen such that ψ satisfies conditions $(\mathcal{A}_1), (\mathcal{A}_2), (\mathcal{A}_3)$ listed below then the wavelet decomposition (1.2.9) is optimal [14, Theorem 5] in the sense of (1.2.3).
Back to the construction of (1.2.12), a classical example of ν due to Daubechies [51, p. 119] is

$$\nu(x) = x^4\left(35 - 84x + 70x^2 - 20x^3\right),$$

which entails that $\widehat{\psi^{\mathrm{M}}}$ has 3 vanishing derivatives at $|\xi| = 2\pi/3,\ 4\pi/3,\ 8\pi/3$. Below, we will propose another example of ν with higher order vanishing derivatives at the boundaries in order to get (see the proof in Subsection 1.3.2) a fast decay rate of all the derivatives $(\psi^{\mathrm{M}})^{(k)}$ at infinity, which results in reducing the complexity cost of the ReLU NN architecture that we will build in Theorem 1.2.2. Note that the construction below might not be numerically optimal among the large literature on wavelets and applications in signal processing, however it is a concrete example on which we can base our theoretical result.

Construction of ψ^{M} Let $q_{\mathrm{st}(\beta)}$ be the quantile function of a Student distribution with β degrees of freedom. Thus, considering the function

$$\nu(u) := \frac{1}{1+\exp(-q_{\mathrm{st}(\beta)}(u))}, \qquad u \in [0,1] \tag{1.2.15}$$

conditions (1.2.13) are easily satisfied. To be self content, we now briefly recall and verify $(\mathcal{A}_1), (\mathcal{A}_2), (\mathcal{A}_3)$ from [14, p. 456] to validate the use of such ν:

- $(\mathcal{A}_1)$. $\left\{x \mapsto 2^{j/2}\psi(2^j x - k), j \in \mathbb{Z}, k \in \mathbb{Z}\right\}$ is an orthonomal basis of $\mathbf{L^2}(\mathbb{R}, \mathrm{d}x)$.
- $(\mathcal{A}_2)$. The Fourier transform $\widehat{\psi}$ is 4 times continuously differentiable. Moreover, for any $k = 0,1,2,3$ there is a constant $C > 0$ such that

$$\left|\widehat{\psi}^{(k)}(\xi)\right| \le C(1+|\xi|)^{-3/2}, \text{ for any } \xi \in \mathbb{R}.$$

$(\mathcal{A}_3)$. $\widehat{\psi}(\xi)$ has a zero of order 4 at $\xi = 0$ (*i.e.* for all $k = 0, 1, 2, 3$, $\widehat{\psi}^{(k)}(0) = 0$).

Condition $(\mathcal{A}_1)$ clearly holds since the wavelets $\left\{\psi^{\mathrm{M}}_{j,k}\right\}_{(j,k)\in\mathbb{Z}^2}$ generate an orthonormal basis of $\mathbf{L^2}(\mathbb{R}, \mathrm{d}x)$. Condition $(\mathcal{A}_3)$ is straightforwardly satisfied because $\widehat{\psi^{\mathrm{M}}}$ vanishes at 0. Last, consider $(\mathcal{A}_2)$: the decay of $\widehat{\psi^{\mathrm{M}}}$ and its derivatives at infinity is straightforward since it has compact support. What really needs to be checked is the smoothness property of $\widehat{\psi^{\mathrm{M}}}$: let us justify that it is $\mathscr{C}^\infty$. Observe that this follows from the tentative property $\nu^{(q)}(0^+) = \nu^{(q)}(1^-) = 0$ for all $q \in \mathbb{N}_0$. To see this, remind that the Student distribution belongs to the Fréchet maximum domain of attraction [52, Theorem. 1.2.1(1.)], therefore $q_{\mathrm{st}(\beta)}$ increases as a power function with exponent $\gamma = 1/\beta > 0$ called the tail-index. Second, for all $q \in \mathbb{N}$, $q^{(q)}_{\mathrm{st}(\beta)}(u)$ increases at most as a power function around 0 and 1, see [91, Lemma 15]. Moreover, the sigmoid function $\Delta(x) := (1 + \exp(-x))^{-1}$ clearly satisfies $\Delta^{(q)}(x) = \mathcal{O}_c\left(\exp(-|x|)\right)$ for $q \geq 1$. Hence, applying the Faà di Bruno formula [112, p. 224-226] for expanding the derivative of the composition of $\Delta(\cdot)$ and $q_{\mathrm{st}(\beta)}(\cdot)$ gives

$$\nu^{(q)}(u) = \sum_{l=1}^{q} \frac{1}{l!} \Delta^{(l)}\left(q_{\mathrm{st}(\beta)}(u)\right) \sum_{i_s\in\mathbb{N}_0 : i_1+\cdots+i_l=q} \frac{q!}{i_1! i_2! \ldots i_l!} \prod_{s=1}^{l} q^{(i_s)}_{\mathrm{st}(\beta)}(u), \quad q \in \mathbb{N}_0,$$

which readily leads to $\nu^{(q)}(0^+) = \nu^{(q)}(1^-) = 0$ since the exponential function decays faster than any polynomials. See Figure 1.9 for an illustration of Ψ_H built with (1.2.12) and (1.2.15) .

We are now in a position to state our second main result.

Theorem 1.2.2. *Let $N \geq 2$ and $\left(\Omega^N, \mathcal{F}^N, \mathbb{P}^N\right)$ be a probability space supporting N i.i.d. standard Gaussian random variables $G_{1:N}$. Therefore, there exists an extension $(\Omega, \mathcal{F}, \mathbb{P})$ supporting a fractional Brownian motion B^H such that $\forall p \in (0, 1]$, for all $r \in \mathbb{N}_0$ there exist a ReLU neural network*

$$\tilde{B}^H_{N,p} : \begin{cases} \mathbb{R}^N & \to \mathscr{C}^0([0,1], \mathbb{R}) \\ G_{1:N} := (G_1, \ldots, G_N) & \mapsto \tilde{B}^H_{N,p}(\cdot, G_{1:N}) \end{cases}$$

and a finite random variable $C \geq 0$ (independent from N and p) such that

$$\mathbb{P}\left(\sup_{t\in[0,1]} \left|B^H(t) - \widetilde{B}^H_{N,p}(t, G_{1:N})\right| \leq C N^{-H} \left(1 + \log(N)\right)^{1/2}\right) \geq 1 - p. \tag{1.2.16}$$

Additionally, $\widetilde{B}^H_{N,p}$ is composed by

1. $\mathcal{O}_c\left(\log\left(\frac{N\rho_N}{(1+\log(N))^{1/2}}\right)\right)$ *hidden layers,*
2. $\mathcal{O}_c\left(N^{1+\frac{H+1}{2r}} \log\left(\frac{N\rho_N}{(1+\log(N))^{1/2}}\right)\left(\frac{\rho_N}{(1+\log(N))^{1/2}}\right)^{\frac{1}{2r}}\right)$ *neurons and parameters,*

where ρ_N is defined in Theorem 1.2.1. The constants in $\mathcal{O}_c(\cdot)$ may depend on r and H.

Observe that Remarks 1-2-3 apply similarly to the above Theorem.

1.2.4 Discussion

In Table 1.1 we compare the asymptotic architecture bounds between a BM and a fBm. Note that the BM benefits from a natural construction of the F-S wavelet through ReLU functions. In comparison, the fBm construction suffers from 1) an additional approximation of the wavelet Ψ_H and 2) a larger bound on the sum over $\mathcal{I}_N$, which has only a log impact in the asymptotic

	BM	fBm
error tolerance ($\texttt{TOL}$)	$N^{-1/2}$	N^{-H}
depth	$\log\left(\texttt{TOL}^{-1}\right)$	$\log\left(\texttt{TOL}^{-1}\right)$
complexity	$\texttt{TOL}^{-2}\log\left(\texttt{TOL}^{-1}\right)$	$\texttt{TOL}^{-\left(\frac{1}{H}+\zeta\right)}\log\left(\texttt{TOL}^{-1}\right)$

Table 1.1: For a given confidence probability p, asymptotic complexity rates with respect to tolerance error ($\texttt{TOL}$). The parameter ζ can be taken arbitrary small, constants depending on H, p and $\zeta > 0$ are omitted.

NN architecture (see details in the proof in Subsection 1.3.2). Therefore, both models have the same asymptotic depth (with a constant depending on r) and a very close complexity in terms of the latent dimension N.

The takeaway message from these results is that a NN with N Gaussian r.v. as inputs for approximating a process with a time regularity H (and an approximation error N^{-H} up to log-term) may have at most a depth $\log N$ and a complexity $N \log N$. Although the set $\mathcal{I}_N$ is not explicit for finding the optimal fBm NN parameters, this part can be achieved through the optimization of the GAN model with the appropriate architecture detailed in Subsection 1.3.2.

1.3 Proofs

In this section we will discuss the constructive proofs of the ReLU NN that appear in the main results. We start with the case $H = 1/2$ (standard Brownian motion) in Subsection 1.3.1 and we then handle the general case of fBm in Subsection 1.3.2. Before that, recall the output expression of a 1-hidden layer NN given some input $x \in \mathbb{R}$ and parameterized by $\theta = \left\{w_k^{(1)}, w_k^{(2)}, b_k^{(1)}, b^{(2)}\right\}_{k=1}^{K}$ is

$$\sum_{k=1}^{K} w_k^{(2)} \sigma\left(w_k^{(1)} x + b_k^{(1)}\right) + b^{(2)}, \tag{1.3.1}$$

with $\sigma(x) := \max(0, x)$ the ReLU function. Similarly, a multi-layer NN is just multiple compositions of σ with (1.3.1) between different hidden layers. For readers interested in having references on approximation properties of NN, we may refer to [131, Theorem 1 p. 70] for L_2 error using single hidden layer NN, to [152, Corollary 6.4 p. 170] for uniform approximations, and to a more recent paper [189] which has shown some uniform convergence rate for multi-layers NN.

1.3.1 NN representation of BM

In the following proof of Theorem 1.2.1 we will restrict to $N \in \mathcal{M} = \left\{2^{n+1}, n \in \mathbb{N}\right\}$. However note that if one wants to choose a $N \notin \mathcal{M}$, it will neither impact the error nor the complexity bounds in Theorem 1.2.1. Indeed, it suffices to take $n = \left\lfloor \frac{\log(N)}{\log(2)} - 1 \right\rfloor$ and $N' = 2^{n+1}$ such that $N' \in (\frac{N}{2}, N]$, and then set $\tilde{B}_{p,N}(t, G_{1:N}) := \tilde{B}_{p,N'}(t, G_{1:N'})$. Regarding the error bound and complexity w.r.t. N, use those for N' by easily adjusting constants: indeed, since $N' \leq N$, it follows that $\rho_{N'} \leq \rho_N$, $\frac{1}{(1+\log(N'))} \leq \frac{1}{1-\log(2)} \frac{1}{(1+\log(N))}$ for the complexity bound and ${N'}^{-1/2}(1+\log(N'))^{1/2} \leq \sqrt{2} N^{-1/2}(1+\log(N))^{1/2}$ for the error bound.

From now on, $N = 2^{n+1}$. For ease of notation, let

$$s_{j,k}(t) := \frac{\psi_{j,k}(t)}{2^{j/2}} = \psi\left(2^j t - k\right) \in [0, 1], \tag{1.3.2}$$

be the normalized F-S wavelet, where $\psi = \psi^{\text{FS}}$ in this section. Then, in view of (1.2.6) and Lemma 1.2.1, the objective is to find a ReLU NN with N standard Gaussian variables and the time t as inputs, that can approximate with uniform error and high probability

$$B_N(t) = G_1 t + \sum_{j=0}^{n} \sum_{k=0}^{2^j-1} 2^{-(j/2+1)} s_{j,k}(t) G_{j,k}. \tag{1.3.3}$$

The key advantage with the F-S wavelet (1.2.5) is that the mother wavelet ψ can be built easily with 3 ReLUs and 9 parameters such as

$$\psi(x) = 2\left(\sigma(x) - 2\sigma\left(x - \frac{1}{2}\right) + \sigma\left(x - 1\right)\right). \tag{1.3.4}$$

Clearly a product operation in (1.3.3) is required between the inputs $G_{j,k}$ (*i.e.* the latent space in a GAN setting) and the normalized wavelets $s_{j,k}$ just built (see Figure 1.1). Since such an operation is not natively done in a feedforward network, let us study how to approximate it.

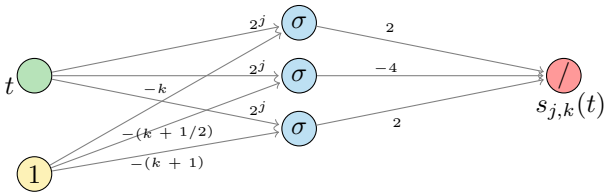


Figure 1.1: Neural network construction of a normalized Faber-Schauder basis function (1.3.2). The circles filled with σ represent a ReLU function, while the ones with a / represent the identity function.

How to make a product with a NN

Let $h(x) = x^2$. The key observation in [189, Proposition 2] based on [175] is that h can be approximated by piece-wise linear interpolation

$$\widetilde{h}_\ell(x) = x - \sum_{j=1}^{\ell} \frac{\psi^{[\circ j]}(x)}{2^{2j}}, \tag{1.3.5}$$

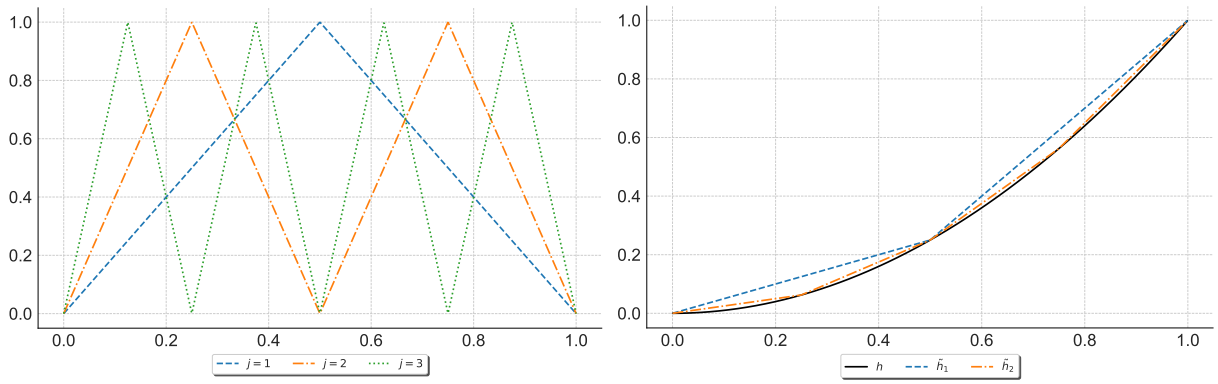
with

$$\psi^{[\circ j]}(x) := \underbrace{\psi \circ \cdots \circ \psi}_{j \text{ times}}(x), \tag{1.3.6}$$

such that

$$\sup_{x\in[0,1]} \left| h(x) - \widetilde{h}_\ell(x) \right| = 2^{-2(\ell+1)}. \tag{1.3.7}$$

Expression (1.3.5) can be interpreted as a NN approximation with ℓ hidden layers, where each composition in (1.3.6) is just the sum of all translated positions of a F-S wavelet, *i.e.* $\forall j \geq 0, \psi^{[\circ j+1]}(x) = \sum_{k=0}^{2^j-1} \psi\left(2^j x - k\right)$. Therefore, instead of making a linear combination of such functions built through a long single hidden layer, the benefit of increasing the depth of the network allows to increase at a geometric rate the number of wavelets and to reduce the complexity cost from $3 \times 2^{\ell-1}$ to 3ℓ neurons.



(a) (b)

Figure 1.2: Plot of $\psi^{[\circ j]}$ for $j = \{1, 2, 3\}$ (a); approximation of $h(x) = x^2$ with $\widetilde{h}_\ell$ for $\ell = \{1, 2\}$

Additionally, one shall be aware that $\widetilde{h}_\ell$ does approximate the square function only inside the interval $[0, 1]$ (see Figure 1.2b). Therefore we introduce a new function

$$\overset{\vee}{h}_\ell(x) = \widetilde{h}_\ell(|x|) = \widetilde{h}_\ell(\sigma(x) + \sigma(-x)), \tag{1.3.8}$$

which applies a ReLU absolute value on the input. Obviously $\overset{\vee}{h}_\ell$ extends the approximation of h on $[-1, 1]$ such that

$$\sup_{|x|\leq 1} \left| h(x) - \overset{\vee}{h}_\ell(x) \right| = 2^{-2(\ell+1)}. \tag{1.3.9}$$

The NN construction of (1.3.8) requires $\ell + 1$ hidden layers and $\mathcal{O}_c(\ell)$ neurons and parameters (see Figure 1.3).

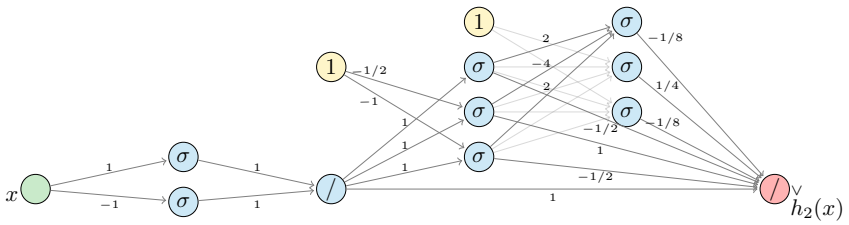


Figure 1.3: Neural network architecture of $\overset{\vee}{h}_\ell$ with $\ell = 2$. Lighter arrows refer to similar parameters which can easily be inferred from (1.3.4). For implementation purpose, one can obviously bypass the identity function in the middle of the network which is put here for the sake of clarity.

Once the square operation is approximately synthetized through a ReLU NN, we can leverage the polarization identity to get the product operation $(x, y) \mapsto xy$. Because the above approximation (1.3.9) is valid only on the interval $[-1, 1]$, it is useful to use a polarization identity with some flexible rescalings of x and y. It writes, for any $a, b > 0$,

$$xy = ab\left(-\left(\frac{x}{2a} - \frac{y}{2b}\right)^2 + \left(\frac{x}{2a} + \frac{y}{2b}\right)^2\right). \tag{1.3.10}$$

The following Proposition provides a uniform error bound on the approximation of the product with a ReLU NN.

Proposition 1.3.1. *Let $k(x, y) := xy$. Then, for any $\ell \in \mathbb{N}_0$, for given $a > 0$ and $b > 0$,*

1. *there exists a NN* $\widetilde{k}_\ell^{a,b} : \mathbb{R}^2 \to \mathbb{R}$ *with* $\ell + 1$ *hidden layers such that*

$$\sup_{x,y:|x|\le a,|y|\le b} \left| k(x,y) - \widetilde{k}_\ell^{a,b}(x,y) \right| \le ab2^{-(2\ell+1)}; \tag{1.3.11}$$

2. *if* $x = 0$ *or* $y = 0$, *then* $\widetilde{k}_\ell^{a,b}(x,y) = 0$;

3. *the ReLU NN* $\widetilde{k}_\ell^{a,b}$ *can be implemented with no more than* $\mathcal{O}_c(\ell)$ *complexity and a depth* $\ell + 1 := \left\lceil \frac{1}{2\log(2)} \log\left(\frac{ab}{\varepsilon}\right) - \frac{1}{2} \right\rceil + 1$, *where* ε *is the error tolerance in sup norm.*

Proof. It is enough to set

$$\widetilde{k}_\ell^{a,b}(x,y) := ab\left(-\overset{\vee}{h}_\ell\left(\frac{x}{2a} - \frac{y}{2b}\right) + \overset{\vee}{h}_\ell\left(\frac{x}{2a} + \frac{y}{2b}\right)\right)$$

and to apply (1.3.9), while observing that when $|x| \le a$ and $|y| \le b$, $\frac{x}{2a} \pm \frac{y}{2b} \in [-1,1]$. □

Final approximation of B_N

Based on Proposition 1.3.1, it seems that we can deduce a uniform bound on the product $s_{j,k}(t)G_{j,k}$ by a linear combination of composition functions of ReLUs (*i.e.* a multi-layer NN). Nevertheless, recall that (1.3.11) only holds for $|x| \le a$ and $|y| \le b$. Thus, although it is clear from (1.3.2) that for all $t \in [0,1]$ we have $s_{j,k}(t) \in [0,1]$, the random variables $G_{j,k}$ need however to be bounded in order to use Proposition 1.3.1: it can be made only with some probability.

Proposition 1.3.2. *Let* $N \in \mathbb{N}_0$ *and* $p \in (0,1]$, *set*

$$\rho_N = -\Phi^{-1}\left(\frac{p}{2N}\right) \ge 0,$$

with Φ^{-1} *the quantile function of the standard Gaussian distribution, and let* $G_{1:N}$ *be i.i.d. standard Gaussian random variables. Then*

$$\mathbb{P}\left(\forall i = 1,\dots,N : |G_i| \le \rho_N\right) \ge 1 - p.$$

Proof. Clearly, the probability on the above left hand side equals

$$1 - \mathbb{P}\left(\bigcup_{i=1}^N \left\{ |G_i| \ge \rho_N \right\}\right) \ge 1 - 2N\Phi\left(-\rho_N\right) = 1 - p.$$

□

Therefore combining Propositions 1.3.1 and 1.3.2 with $a = 1$ and $b = \rho_N$, we can define

$$\widetilde{k}_\ell^{1,\rho_N}(s_{j,k}(t), G_{j,k}) := \rho_N\left(-\overset{\vee}{h}_\ell\left(\frac{s_{j,k}(t)}{2} - \frac{G_{j,k}}{2\rho_N}\right) + \overset{\vee}{h}_\ell\left(\frac{s_{j,k}(t)}{2} + \frac{G_{j,k}}{2\rho_N}\right)\right), \tag{1.3.12}$$

which can be implemented with $\ell + 2$ hidden layers (since we need an additional one to build the $s_{j,k}$) and $\mathcal{O}_c(\ell)$ neurons and parameters (see Figure 1.4).

Remark 4. A slight advantage of (1.3.12) over the polarization identity in [189, Equation (4) p. 106] is that it only requires two square approximations instead of three.

Let $\widetilde{B}_N$ be the NN approximation of (1.3.3) such that

$$\widetilde{B}_N(t) = \widetilde{k}_\ell^{1,\rho_N}(t, G_1) + \sum_{j=0}^{n}\sum_{k=0}^{2^j-1} 2^{-(j/2+1)}\widetilde{k}_\ell^{1,\rho_N}\left(s_{j,k}(t), G_{j,k}\right), \tag{1.3.13}$$

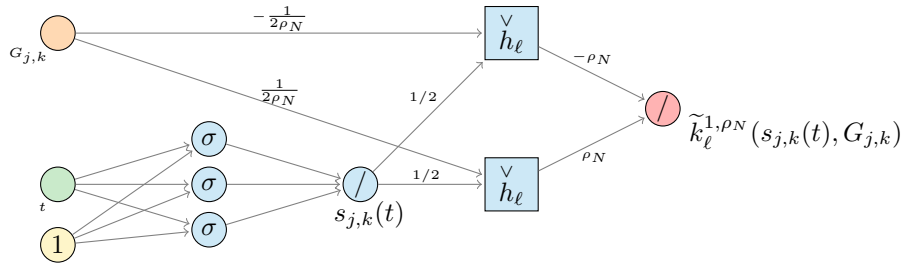


Figure 1.4: Neural network architecture of $\widetilde{k}_\ell^{1,\rho_N}(s_{j,k}(t), G_{j,k})$.

with $\widetilde{k}_\ell^{1,\rho_N}$ defined in (1.3.12). Therefore, on the event $\{|G_i| \leq \rho_N : i = 1, \dots, N\}$ which has a probability greater than $1 - p$, one has

$$\sup_{t\in[0,1]} \left|B_N(t) - \widetilde{B}_N(t)\right| \leq \sup_{t\in[0,1]} \left|tG_1 - \widetilde{k}_\ell^{1,\rho_N}(t, G_1)\right| + \sup_{t\in[0,1]} \left|\sum_{j=0}^{n}\sum_{k=0}^{2^j-1} 2^{-(j/2+1)} \left(s_{j,k}(t)G_{j,k} - \widetilde{k}_\ell^{1,\rho_N}(s_{j,k}(t), G_{j,k})\right)\right|$$

(from Proposition 1.3.1 and since $s_{j,k}(t) \in [0,1]$ and $|G_{j,k}| \leq \rho_N$)

$$\begin{aligned} &\leq \rho_N 2^{-(2\ell+1)} \left(1 + \sum_{j=0}^{n}\sum_{k=0}^{2^j-1} 2^{-(j/2+1)}\right) \\ &= \rho_N 2^{-(2\ell+1)} \left(1 + \frac{N^{1/2}-1}{2\left(\sqrt{2}-1\right)}\right) \quad \text{(recall that } N = 2^{n+1}\text{)} \\ &\leq \rho_N 2^{-2\ell} N^{1/2}. \end{aligned}$$

Hence, with probability at least $1 - p$, combining Lemma 1.2.1 with the above yields

$$\sup_{t\in[0,1]} \left|B(t) - \widetilde{B}_N(t)\right| \leq C_{(1.2.7)} N^{-1/2} \big(1 + \log(N)\big)^{1/2} + \rho_N 2^{-2\ell} N^{1/2}.$$

It follows that, if we choose

$$\ell = \left\lceil \frac{1}{2\log(2)} \log\left(\frac{N\rho_N}{\big(1 + \log(N)\big)^{1/2}}\right)\right\rceil \vee 1, \tag{1.3.14}$$

then (1.2.8) is proved with $\tilde{B}_{N,p}(t, G_{1:N}) := \widetilde{B}_N(t)$. All in all, based on Figure 1.4, the architecture required for the N products in (1.3.13), *i.e.* N sub-networks, yields a total of at most $\ell + 2$ hidden layers and a complexity $\mathcal{O}_c(N\ell)$. Replacing with (1.3.14) gives the stated bounds. □

1.3.2 NN representation of fBm

Now that we are acquainted with the case of BM, we can move on to the more general case which requires additional arguments. In view of (1.2.14) with a fixed $\gamma > 0$, the goal here is to prove that there exists a ReLU NN approximating uniformly

$$B_N^H(t) = \sum_{(j,k)\in\mathcal{I}_N} 2^{-jH} \left(\Psi_H\left(2^j t - k\right) - \Psi_H(-k)\right) G_{j,k}, \tag{1.3.15}$$

with $\mathrm{Card}(\mathcal{I}_N) \leq N$ and $G_{j,k} \sim \mathcal{N}(0,1)$. The proof will be composed in two parts. First we will discuss how Ψ_H can be approximated by ReLU basis functions in $\mathbb{R}$. Second, we will see how to control the error on the product with Gaussians in (1.3.15). In this section we will write $\psi^{\mathrm{M}}(\cdot) = \psi(\cdot)$ for the Lemarié-Meyer wavelet (1.2.12) with $\nu(\cdot)$ as in (1.2.15).

Approximation of Ψ_H

We want to show that for all $\varepsilon \in (0,1)$ there exists a ReLU NN $\widetilde{g}$ such that

$$\sup_{u\in\mathbb{R}} |\Psi_H(u) - \widetilde{g}(u)| \leq \varepsilon. \tag{1.3.16}$$

Note that we cannot apply the universal approximation theorem [47, Theorem 1] which holds for continuous functions with compact support. To tackle the infinite support, the strategy will consist of first approximating Ψ_H in some interval $[-u_{\max}, u_{\max}]$, and then using the fast decay rate of $|\Psi_H(u)|$ for $|u| > u_{\max}$. Indeed, since by construction $\widehat{\psi}$ and its derivatives vanish in the neighborhood of $\xi = 0$, $\widehat{\Psi}_H$ defined in (1.2.10) is $\mathscr{C}^\infty$ with compact support for any parameter $H \in \mathbb{R}$. So for all $(m,q) \in \mathbb{N}^2$, we claim that

$$\left|\Psi_H^{(q)}(u)\right| \leq \frac{C_{H-q,m}}{1+|u|^{m+1}}, \tag{1.3.17}$$

where $C_{H-q,m}$ is a constant depending on $H-q$ and m. The property for $q=0$ is clear: use the inverse Fourier transform and $m+1$ integration by parts, taking advantage that the derivatives of $\widehat{\Psi}_H$ vanish at the boundary of its support (see discussion after (1.2.15)). For $q \neq 0$, observe that $\Psi_H^{(q)}(u) = \Psi_{H-q}(u)$ and the property follows. Now we proceed to (1.3.16), by following the ideas of [189, Theorem 1] with some variations. In (1.3.17), we have a degree of freedom with the choice of the parameter m, it will be fixed at the end of the proof.

Consider a uniform grid of M points $\{u_i = (i-1)\delta - u_{\max}\}_{i=1}^M$ with $M > 1$ and $\delta = \frac{2u_{\max}}{M-1}$ on the domain $[-u_{\max}, u_{\max}]$, assuming $\delta \leq 1/2$. The parameter $u_{\max} > 0$ will be fixed later. Additionally, for $i = 1, \ldots, M$, we define a triangular function

$$\phi_i(u) := \phi\left(\frac{u-u_i}{\delta}\right),$$

where

$$\phi(t) := \sigma(t+1) + \sigma(t-1) - 2\sigma(t),$$

and with the following (obvious) properties:

1. $\phi_i(\cdot)$ is symmetric around u_i,
2. $\sup_{u\in\mathbb{R}} |\phi_i(u)| = \phi_i(u_i) = 1$,
3. $\mathtt{supp}(\phi_i) \in [u_i - \delta,\, u_i + \delta]$,
4. $\sum_{i=1}^M \phi_i(u) \equiv 1, \quad$ for $u \in [-u_{\max},\, u_{\max}]$.

The function ϕ_i is nothing else than another FS wavelet $\psi_{j,k}^{\mathrm{FS}}$ with slightly different scaling and position parameters. Now let $r \in \mathbb{N}_0$ and consider a localized Taylor polynomial function

$$g_1(u) := \sum_{i=1}^M \phi_i(u) P_i(u), \tag{1.3.18}$$

where P_i is the Taylor polynomial of degree $(r-1)$ of $\Psi_H \in \mathscr{C}^\infty$ at the point u_i given by

$$P_i(u) := \sum_{q=0}^{r-1} \frac{\Psi_H^{(q)}(u_i)}{q!}(u-u_i)^q.$$

To approximate the q-power function, we will need the following result.

Proposition 1.3.3. *Let $\ell \in N_0$, $a > 0$ and $b > 0$. For any $q \in \mathbb{N}$, define recursively the ReLU NN with at most $(q-1)(\ell+1)$ hidden layers by*

$$y \mapsto \widetilde{y^q} := \widetilde{k}_\ell^{b,b_q}\left(y, \widetilde{y^{q-1}}\right), \quad q \geq 2,$$

with by convention $\widetilde{y^0} := 1$, $\widetilde{y^1} := y$, where $\widetilde{k}_\ell^{a,b}$ is defined in Proposition 1.3.1 and where

$$b_q := b^{q-1}\left(1 + 2^{-(2\ell+1)}\right)^{q-2}. \tag{1.3.19}$$

It is such that

$$\sup_{y:|y|\leq b} \left|y^q - \widetilde{y^q}\right| \leq b^q \left(\left(1 + 2^{-(2\ell+1)}\right)^{q-1} - 1\right), \tag{1.3.20}$$

$$\sup_{x,y:|x|\leq a,|y|\leq b} \left|xy^q - \widetilde{k}_\ell^{a,b_{q+1}}\left(x, \widetilde{y^q}\right)\right| \leq ab^q \left(\left(1 + 2^{-(2\ell+1)}\right)^{q} - 1\right). \tag{1.3.21}$$

Proof. We set $\eta := 2^{-(2\ell+1)}$ and we proceed by induction. Inequality (1.3.20) holds for $q = 2$ thanks to Proposition 1.3.1. Now take $q \geq 3$, assume (1.3.20) holds for $q-1$. Clearly, this implies

$$\sup_{|y|\leq b} \left|\widetilde{y^{q-1}}\right| \leq b^{q-1}(1+\eta)^{q-2} = b_q. \tag{1.3.22}$$

Therefore,

$$\begin{aligned}
\sup_{|y|\leq b} \left|y^q - \widetilde{y^q}\right| &\leq \sup_{|y|\leq b} \left|y^q - y\,\widetilde{y^{q-1}}\right| + \sup_{|y|\leq b} \left|y\,\widetilde{y^{q-1}} - \widetilde{y^q}\right| \\
&\leq b \sup_{|y|\leq b} \left|y^{q-1} - \widetilde{y^{q-1}}\right| + \sup_{|y|\leq b} \left|y\,\widetilde{y^{q-1}} - \widetilde{k}_\ell^{b,b_q}\left(y, \widetilde{y^{q-1}}\right)\right| \\
&\leq bb^{q-1}\left((1+\eta)^{q-2} - 1\right) + bb^{q-1}(1+\eta)^{q-2}\eta \\
&= b^q\left((1+\eta)^{q-1} - 1\right)
\end{aligned}$$

where, at the last inequality, we have used Proposition 1.3.1 combined with bound (1.3.22). We are done with (1.3.20). Similarly for (1.3.21), we get

$$\sup_{|x|\leq a,|y|\leq b} \left|xy^q - \widetilde{k}_\ell^{a,b_{q+1}}\left(x, \widetilde{y^q}\right)\right| \leq a \sup_{|y|\leq b} \left|y^q - \widetilde{y^q}\right| + \sup_{|x|\leq a,|y|\leq b} \left|x\,\widetilde{y^q} - \widetilde{k}_\ell^{a,b_{q+1}}\left(x, \widetilde{y^q}\right)\right|.$$

Combining (1.3.20) and Proposition 1.3.1 with (1.3.22), we get (1.3.21). □

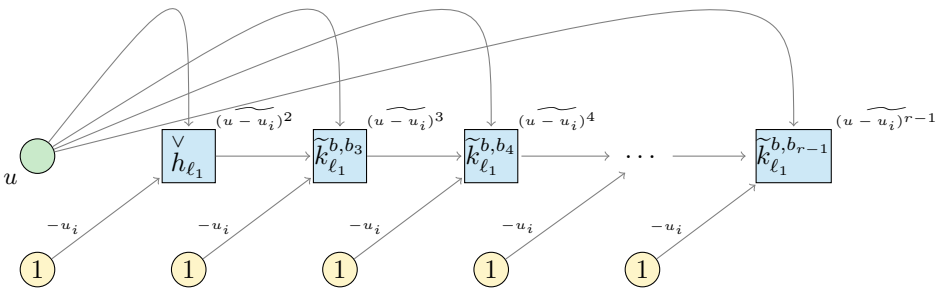


Figure 1.5: Neural network architecture of all power functions of $(u-u_i)^q$ for $q \in \{2, \ldots, r-1\}$ with $b = 2\delta$ and b_q defined in (1.3.19).

We are now in a position to prove (1.3.16). Given the support property of ϕ_i, the strategy consists of splitting the error approximation in three terms:

1. A classical Taylor bound on the main interval yields

$$\begin{aligned}\sup_{|u|\le u_{\max}} |\Psi_H(u) - g_1(u)| &= \sup_{|u|\le u_{\max}} \left| \sum_{i=1}^{M} \phi_i(u) \left(\Psi_H(u) - P_i(u)\right) \right| \\ &\le 2 \max_{i=1,\dots,M} \sup_{u\in \mathtt{supp}(\phi_i)} |\Psi_H(u) - P_i(u)|\end{aligned}$$

since u is in the support of at most two ϕ_i's and $|\phi_i(u)| \le 1$,

$$\begin{aligned}&\le 2 \max_{i=1,\dots,M} \sup_{u\in \mathtt{supp}(\phi_i)} \frac{\left|\Psi_H^{(r)}(u)\right|}{r!} (2\delta)^r \\ &\le \frac{2}{r!} C_{H-r,m} (2\delta)^r,\end{aligned}$$

using (1.3.17). Let $\widetilde{g_{i,q}}(\cdot)$ be the ReLU NN approximation of $u \mapsto \phi_i(u)(u-u_i)^q$ using (1.3.21) with $|\phi_i(u)| \le 1 = a$ and $|u - u_i| \le 2\delta = b$ (see Figure 1.5). In view of (1.3.18), set

$$\widetilde{g}(u) := \sum_{q=0}^{r-1} \frac{1}{q!} \sum_{i=1}^{M} \Psi_H^{(q)}(u_i) \widetilde{g_{i,q}}(u). \tag{1.3.23}$$

Observe, from the second statement of Proposition 1.3.1 that $\widetilde{g_{i,q}}(u) = 0$ for $u \notin \mathtt{supp}(\phi_i)$. So using (1.3.21) (setting $\eta := 2^{-(2\ell_1+1)}$ with $\ell_1 \in \mathbb{N}_0$) leads to

$$\begin{aligned}\sup_{|u|\le u_{\max}} |g_1(u) - \widetilde{g}(u)| &\le \sum_{q=0}^{r-1} \sup_{|u|\le u_{\max}} \frac{\left|\Psi_H^{(q)}(u)\right|}{q!} \sum_{i=1}^{M} \sup_{|u|\le u_{\max}} |\phi_i(u)(u-u_i)^q - \widetilde{g_{i,q}}(u)| \\ &\le 2r \max_{q=0,\dots,r-1} C_{H-q,m} \sum_{q=1}^{\infty} (2\delta)^q \frac{((1+\eta)^q - 1)}{q!}.\end{aligned} \tag{1.3.24}$$

Using that $2\delta \le 1$ and $\eta \le 1/2$, we easily get that the above right hand side is bounded by $r\ \delta\ \eta\ C_e \max_{q=0,\dots,r-1} C_{H-q,m}$ for some universal constant C_e. To sum up, we have proved

$$\sup_{|u|\le u_{\max}} |\Psi_H(u) - \widetilde{g}(u)| \le \mathcal{C}_{H,r,m} \left(\delta^r + \delta\eta\right)$$

where, here and in what follows, $\mathcal{C}_{H,r,m}$ stands for a finite positive constant depending on H, r, m, which value may change from line to line, without changing its name. By taking

$$\delta = \left(\frac{\varepsilon}{6\mathcal{C}_{H,r,m}}\right)^{\frac{1}{r}} \wedge \frac{1}{2} = \mathcal{O}_c\left(\varepsilon^{\frac{1}{r}}\right), \qquad \eta \le \frac{\varepsilon}{6\mathcal{C}_{H,r,m}\delta} = \mathcal{O}_c\left(\varepsilon^{1-\frac{1}{r}}\right), \tag{1.3.25}$$

we have

$$\sup_{|u|\le u_{\max}} |\Psi_H(u) - \widetilde{g}(u)| \le \frac{\varepsilon}{3}.$$

The condition on $\eta \le 1/2$ is satisfied for

$$\ell_1 = \left\lceil \frac{1}{2\log(2)} \log\left(\frac{6C_{H-r,m}\delta}{\varepsilon}\right) - \frac{1}{2} \right\rceil \vee 1 = \mathcal{O}_c\left(\log\left(\varepsilon^{-1}\right)\right). \tag{1.3.26}$$

2. Focusing on the small interval $|u| \in [u_{\max}, u_{\max}+\delta]$ where u belongs to $\mathtt{supp}(\phi_M)$ only, write

$$\begin{aligned}
&\sup_{|u|\in[u_{\max},u_{\max}+\delta]} |\Psi_H(u) - g_1(u)| \\
&\quad = \sup_{|u|\in[u_{\max},u_{\max}+\delta]} |\Psi_H(u) - \phi_M(u)P_M(u)| \\
&\quad \le \sup_{|u|\in[u_{\max},u_{\max}+\delta]} |\Psi_H(u)| + \sup_{|u|\in[u_{\max},u_{\max}+\delta]} |\Psi_H(u) - P_M(u)| \\
&\quad \underset{(1.3.17)}{\le} \frac{C_{H,m}}{1+u_{\max}^{m+1}} + \frac{C_{H-r,m}}{1+u_{\max}^{m+1}}\frac{\delta^r}{r!} \\
&\quad \underset{(1.3.25)}{\le} \frac{\mathcal{C}_{H,r,m}}{1+u_{\max}^{m+1}}.
\end{aligned}$$

Similarly to bound (1.3.24) but taking advantage of the fast decay of $\sup_{u\in[u_{\max},u_{\max}+\delta]} \left|\Psi_H^{(q)}(u)\right|$ yields

$$\sup_{|u|\in[u_{\max},u_{\max}+\delta]} |g_1(u) - \widetilde{g}(u)| \le \mathcal{C}_{H,r,m}\frac{\delta\ \eta}{1+u_{\max}^{m+1}}.$$

All in all, and using $\delta\ \eta \le 1/4$,

$$\sup_{|u|\in[u_{\max},u_{\max}+\delta]} |\Psi_H(u) - \widetilde{g}(u)| \le \frac{\mathcal{C}_{H,r,m}}{1+u_{\max}^{m+1}} \le \frac{\varepsilon}{3}$$

for a new constant $\mathcal{C}_{H,r,m}$ and with the choice

$$u_{\max} := \left(\frac{3\mathcal{C}_{H,r,m}}{\varepsilon}\right)^{\frac{1}{m+1}}. \tag{1.3.27}$$

3. Finally, on the last interval $|u| \in [u_{\max}+\delta, +\infty)$, both $\widetilde{g}(\cdot)$ and $g_1(\cdot)$ vanish, and from (1.3.17), we readily get

$$\sup_{|u|\in[u_{\max}+\delta,+\infty)} |\Psi_H(u) - \widetilde{g}(u)| \le \frac{C_{H,m}}{1+u_{\max}^{m+1}} \le \frac{\varepsilon}{3}.$$

All in all, (1.3.16) is proved with the ReLU NN (1.3.23). Collecting previous asymptotics, we get

$$M = \frac{2u_{\max}}{\delta} + 1 = \mathcal{O}_c\left(\varepsilon^{-\frac{1}{m+1}}\varepsilon^{-\frac{1}{r}}\right). \tag{1.3.28}$$

Error control including Gaussian random variables

We are back to the approximation of (1.3.15). For $(j,k) \in \mathcal{I}_N$ we set

$$Y_{j,k}(t) := \Psi_H\left(2^j t - k\right) - \Psi_H(-k) \quad \text{and} \quad \widetilde{Y}_{j,k}(t) := \widetilde{g}\left(2^j t - k\right) - \widetilde{g}(-k)$$

for its ReLU NN approximation. In view of (1.3.15), let us derive an error bound of the product $Y_{j,k}(t)G_{j,k}$ for $t \in [0,1]$ and $G_{j,k}$ a standard Gaussian random variable. From (1.3.16) with $\varepsilon \le 1$ and (1.3.17), we get

$$\sup_{t\in[0,1]} \left|\widetilde{Y}_{j,k}(t)\right| \vee \sup_{t\in[0,1]} |Y_{j,k}(t)| \le 2\varepsilon + 2\sup_{u\in\mathbb{R}} |\Psi_H(u)| \le 2\left(1 + C_{H,m}\right) =: \bar{C}_H. \tag{1.3.29}$$

Similarly to (1.3.12), we can rewrite for $t \in [0,1]$ and $(j,k) \in \mathcal{I}_N$ the NN product approximation of $\widetilde{Y}_{j,k}(t)G_{j,k}$ with $\ell_2 \in \mathbb{N}_0$ as

$$\widetilde{k}_{\ell_2}^{\bar{C}_H,\rho_N}\left(\widetilde{Y}_{j,k}(t), G_{j,k}\right) = \bar{C}_H\rho_N\left(-\overset{\vee}{h}_{\ell_2}\left(\frac{\widetilde{Y}_{j,k}(t)}{2\bar{C}_H} - \frac{G_{j,k}}{2\rho_N}\right) + \overset{\vee}{h}_{\ell_2}\left(\frac{\widetilde{Y}_{j,k}(t)}{2\bar{C}_H} + \frac{G_{j,k}}{2\rho_N}\right)\right). \tag{1.3.30}$$

Let us work on the event $\{|G_{j,k}| \le \rho_N : (j,k) \in \mathcal{I}_N\}$ which has a probability greater than $1-p$ and let us focus on the approximation error of the first term on the right-hand side of (1.3.30):

$$\begin{aligned}
& \sup_{t\in[0,1]}\left|\overset{\vee}{h}_{\ell_2}\left(\frac{\widetilde{Y}_{j,k}(t)}{2\bar{C}_H} - \frac{G_{j,k}}{2\rho_N}\right) - \left(\frac{Y_{j,k}(t)}{2\bar{C}_H} - \frac{G_{j,k}}{2\rho_N}\right)^2\right| \\
\le & \sup_{t\in[0,1]}\left|\overset{\vee}{h}_{\ell_2}\left(\frac{\widetilde{Y}_{j,k}(t)}{2\bar{C}_H} - \frac{G_{j,k}}{2\rho_N}\right) - \left(\frac{\widetilde{Y}_{j,k}(t)}{2\bar{C}_H} - \frac{G_{j,k}}{2\rho_N}\right)^2\right| \\
+ & \sup_{t\in[0,1]}\left|\left(\frac{\widetilde{Y}_{j,k}(t)}{2\bar{C}_H} - \frac{G_{j,k}}{2\rho_N}\right)^2 - \left(\frac{Y_{j,k}(t)}{2\bar{C}_H} - \frac{G_{j,k}}{2\rho_N}\right)^2\right| \\
\underset{(1.3.9)}{\le} & 2^{-2(\ell_2+1)} + \frac{\sup_{t\in[0,1]}\left|\widetilde{Y}_{j,k}(t) - Y_{j,k}(t)\right|}{2\bar{C}_H} \\
& \qquad \times\left(\frac{\sup_{t\in[0,1]}\left|\widetilde{Y}_{j,k}(t)\right| + \sup_{t\in[0,1]}|Y_{j,k}(t)|}{2\bar{C}_H} + \frac{|G_{j,k}|}{\rho_N}\right) \\
\underset{(1.3.16),(1.3.29)}{\le} & 2^{-2(\ell_2+1)} + \frac{2\varepsilon}{\bar{C}_H}.
\end{aligned}$$

So replacing in (1.3.30) and similarly for the second term, it entails

$$\sup_{(j,k)\in\mathcal{I}_N}\sup_{t\in[0,1]}\left|Y_{j,k}(t)G_{j,k} - \widetilde{k}_{\ell_2}^{\bar{C}_H,\rho_N}\left(\widetilde{Y}_{j,k}(t), G_{j,k}\right)\right| \le 2^{-(2\ell_2+1)}\bar{C}_H\rho_N + 4\varepsilon\rho_N.$$

For the final ReLU NN approximation of (1.3.15), define $\widetilde{B}_N^H$ as

$$\widetilde{B}_N^H(t) := \sum_{(j,k)\in\mathcal{I}_N} 2^{-jH}\widetilde{k}_{\ell_2}^{\bar{C}_H,\rho_N}\left(\widetilde{g}\left(2^j t - k\right) - \widetilde{g}(-k), G_{j,k}\right). \tag{1.3.31}$$

Combining (1.2.14),(1.3.15),(1.3.31) gives (still on the event $\{|G_{j,k}| \le \rho_N : (j,k) \in \mathcal{I}_N\}$)

$$\begin{aligned}
\sup_{t\in[0,1]}\left|B^H(t) - \widetilde{B}_N^H(t)\right| &\le \sup_{t\in[0,1]}\left|B^H(t) - B_N^H(t)\right| + \sup_{t\in[0,1]}\left|B_N^H(t) - \widetilde{B}_N^H(t)\right| \\
&\le C_{(1.2.14)}N^{-H}\left(1+\log(N)\right)^{1/2} + N\left(2^{-(2\ell_2+1)}\bar{C}_H\rho_N + 4\varepsilon\rho_N\right)
\end{aligned}$$

recalling that $\mathrm{Card}\left(\mathcal{I}_N\right) \le N$. It suffices to ensure that the second term at the right-hand side is bounded by $2N^{-H}\left(1+\log(N)\right)^{1/2}$ thanks to the choices

$$\varepsilon = \frac{\left(1+\log(N)\right)^{1/2}}{4\rho_N N^{H+1}} \wedge 1, \tag{1.3.32}$$

$$\ell_2 = \left\lceil \frac{1}{2\log(2)}\log\left(\frac{\bar{C}_H\rho_N N^{H+1}}{\left(1+\log\left(N\right)\right)^{1/2}}\right)\right\rceil \vee 1 = \mathcal{O}_c\left(\log\left(\varepsilon^{-1}\right)\right). \tag{1.3.33}$$

Architecture

The total architecture of $\widetilde{g}$ is composed by M sub-networks, where each $\widetilde{g_{i,q}}$ is built as a cascade of q NN with (ℓ_1+1) hidden layers, *i.e.* $(q-1)$ NN from Proposition 1.3.3 and 1 more from the product with ϕ_i. Therefore, $\widetilde{g}$ requires at most a depth $\mathcal{O}_c(\ell_1)$ and a complexity $\mathcal{O}_c(M\ell_1)$, with constants clearly depending on r. Using M in (1.3.28) and ℓ_1 in (1.3.26), we get the architecture bounds as a function of the accuracy ε, for just one approximation of Ψ_H with $\widetilde{g}$.

As mentioned above, $\widetilde{Y}_{j,k}$ is composed of $2\widetilde{g}$ NN and so it has the same depth but twice the complexity (number of neurons and parameters) of $\widetilde{g}$. Additionally, (1.3.31) requires (ℓ_2+1) hidden layers to perform the multiplications with the Gaussian random variables. Finally, all these operations are computed for N different scaling/transition parameters (j,k). All in all, the total architecture of $\widetilde{B}_N^H$ (see Figure 1.6) is composed of at most

1. $\mathcal{O}_c(\ell_1+\ell_2)$ hidden layers,
2. $\mathcal{O}_c\left(N\left(M\ell_1+\ell_2\right)\right)$ neurons and parameters.

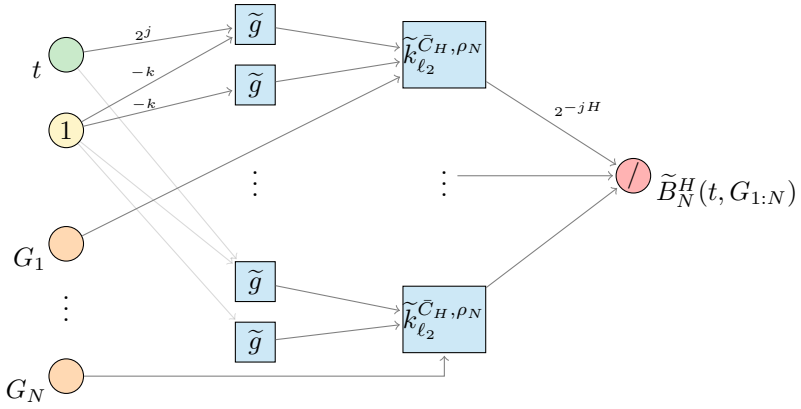


Figure 1.6: Neural network architecture of $\widetilde{B}_N^H(t,G_{1:N})$.

Replacing with (1.3.25), (1.3.28) and (1.3.33) gives the architecture bounds with respect to ε, *i.e.*

1. hidden layers:

$$\mathcal{O}_c(\ell_1+\ell_2)=\mathcal{O}_c\left(\log\left(\varepsilon^{-(1-\frac{1}{r})}\right)+\log\left(\varepsilon^{-1}\right)\right)=\mathcal{O}_c\left(\log\left(\varepsilon^{-1}\right)\right),$$

   where we have observed that the exponent inside the log term can be put in the $\mathcal{O}_c$ since the constants are allowed to depend on r in our notation;

2. neurons and parameters:

$$\begin{aligned}\mathcal{O}_c\left(N\left(M\ell_1+\ell_2\right)\right)&=\mathcal{O}_c\left(N\left(\varepsilon^{-\frac{1}{2r}}\log\left(\varepsilon^{-(1-\frac{1}{r})}\right)+\log\left(\varepsilon^{-1}\right)\right)\right)\\&=\mathcal{O}_c\left(N\varepsilon^{-\frac{1}{2r}}\log\left(\varepsilon^{-1}\right)\right),\end{aligned}$$

   with equilibrium at $r=m+1$ in (1.3.28). Remembering the choice (1.3.32) of ε w.r.t. N gives the announced result.

Remark 5. One can observe that $\widetilde{Y}_{j,k}(\cdot)$ may require only one $\widetilde{g}(\cdot)$ since the term $\Psi_H(-k)$ does not depend on t and so could be approximated by a single NN parameter. It slightly simplifies the construction of Figure 1.6 although it does not change the final asymptotic bounds.

1.4 Numerical results

Computational aspects The numerical experiments have been conducted on the Cholesky computing cluster from Ecole Polytechnique http://meso-ipp.gitlab.labos.polytechnique.fr/user_doc/. All the code was implemented in Python 3.8.2 and using the library PyTorch 1.7.1 for the neural network training.

As mentioned in the introduction, the objective of this paper is neither to focus on the learning task nor to solve the adversarial optimization problem in a GAN setting. As a complement to the above theoretical part, we present here one approach to build a generative NN able to simulate fBm paths. The computing implementation of (1.3.15) on a time grid time $\{t_j = j/T,\, j = 0, \dots, T\}$ is realized in three steps (see more details below): (1) approximate Ψ_H on a specific support, (2) find the set of indices $\mathcal{I}_N$ and (3) draw some noise from the latent space.

1. In order to approximate the function

$$x \mapsto \Psi_H(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{ix\xi} \frac{\widehat{\psi^{\mathrm{M}}}(\xi)}{(i\xi)^{H+1/2}} \, \mathrm{d}\xi, \tag{1.4.1}$$

take M points $\{x_i\}_{i=1}^M$ from a grid with uniform step in a large enough finite interval and compute $\{\Psi_H(x_i)\}_{i=1}^M$ using (1.2.12) and (1.2.15) with a fixed tail-index $\gamma = 1/\beta$ in (1.2.15). As illustrated in Figure 1.9d, one can benefit from a non-uniform grid of points $\{x_i\}_{i=1}^M$ for better approximation. Note that given the high regularity of Ψ_H, a quadrature method with degree D will give an excellent estimation of the integral in (1.4.1). Then, it is straightforward to build a feedforward NN $\widetilde{\Psi}_{H,\theta}$ that can minimize the L^2 distance

$$\mathcal{L}(\theta) = \frac{1}{M} \sum_{i=1}^{M} \left| \Psi_H(x_i) - \widetilde{\Psi}_{H,\theta}(x_i) \right|^2.$$

2. Compute the index set $\mathcal{I}_N$ based on the one proposed in [14, Section 5]. For every integer $J \geq 0$, let $\mathcal{F}_J$ and $\mathcal{P}_J$ be the sets of indices defined as

$$\mathcal{F}_J = \left\{ (j,k) \in \mathbb{Z}^2 : 0 \leq j \leq J \text{ and } |k| \leq (J-j+1)^{-2} 2^{J+4} \right\},$$
$$\mathcal{P}_J = \left\{ (j,k) \in \mathbb{Z}^2 : -J \leq j \leq -1 \text{ and } |k| \leq 2^{[J/2]} \right\}.$$

Next, for $(j,k) \in \mathcal{I}_J = \mathcal{F}_J \cup \mathcal{P}_J$, consider the coefficients

$$c_{j,k} = 2^{-2jH} \left(\Psi_H(2^j - k) - \Psi_H(-k) \right)^2,$$

and with a slight variation from [14], define the truncated set $\mathcal{I}_N$ containing the N largest coefficients $c_{j,k}$ as

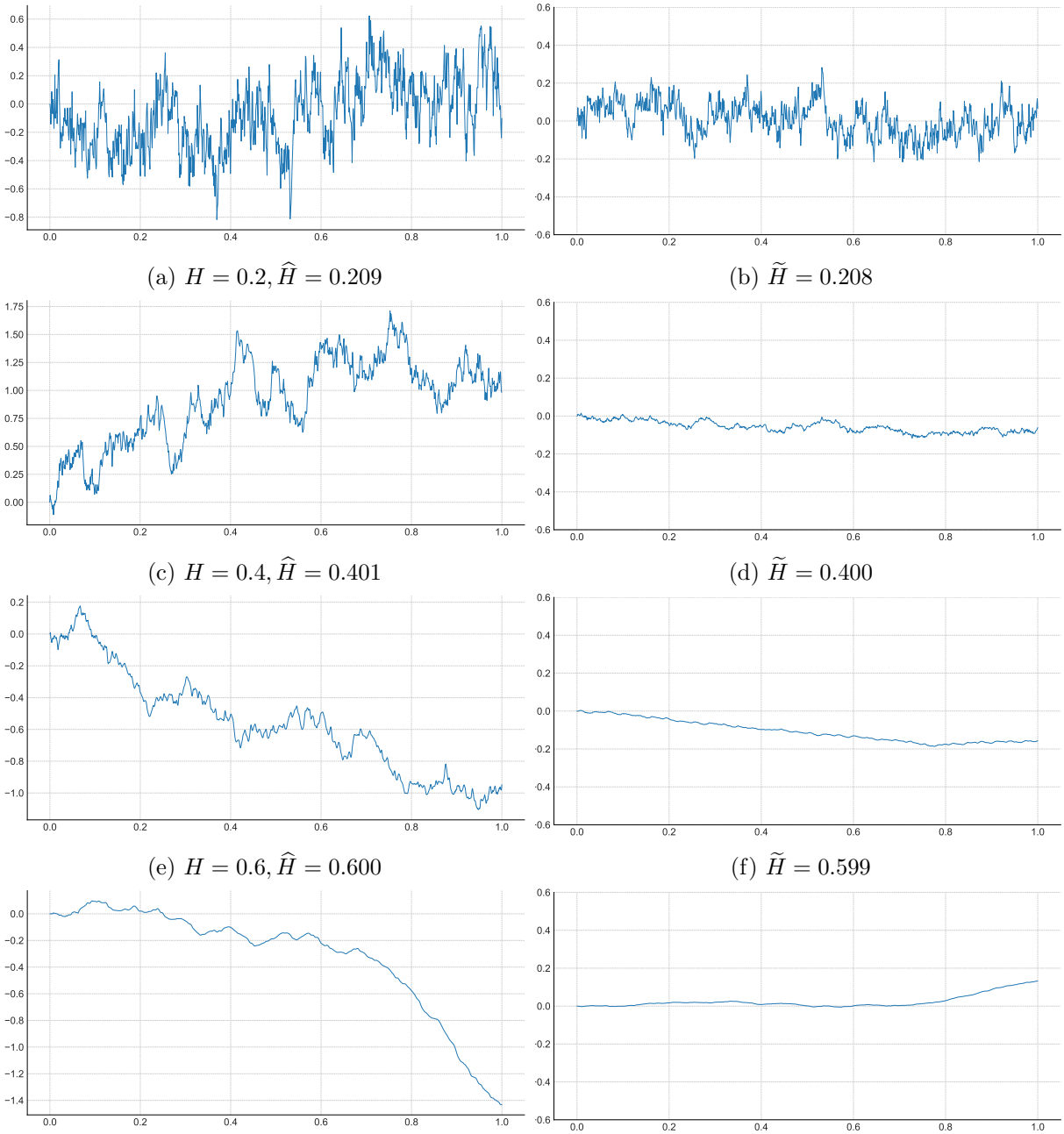
$$\mathcal{I}_N := \bigcup_{i=1}^{N} \mathcal{B}_i,$$

with $\mathcal{B}_{i+1} = \left\{ \underset{(j,k) \in \mathcal{I}_J}{\arg\max}\, c_{j,k} \backslash \mathcal{B}_i \right\}$, $\mathcal{B}_1 = \underset{(j,k) \in \mathcal{I}_J}{\arg\max}\, c_{j,k}$ and $N \leq \mathrm{card}(\mathcal{I}_J)$. Note that (1.3.15) requires the evaluation of Ψ_H at some points $\{x_i\}_{i=1}^M$ with an increasing support with respect to J. Therefore, one must choose a degree D large enough such that the integral values are well estimated at the boundaries of the support.

3. Simulate some independent standard Gaussian random variables $G_{j,k}$. For the sake of simplicity and since the product approximation error (Proposition 1.3.1) is geometrically small with respect to the depth, one can use the real product instead.

In the simulation study, we used $M = 10000$ in $[-1000, 1000]$, including 8000 points in $[-50, 50]$, $J = 10, N = 40000, \gamma = 10000, T = 1000$ and $D = 20000$ with a Gauss-Chebyshev quadrature [2, p. 889]. The neural network $\widetilde{\Psi}_{H,\theta}$ is composed by 10 hidden layers of 200 neurons in order to be consistent with the theoretical result ($\mathcal{O}_c(\log N)$ hidden layers and $\mathcal{O}_c(N \log N)$ parameters). Obviously, one could use a much lighter parametrization. The model was trained with the Adam optimizer [114] with default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, a learning rate of $1e$-3 and a batch-size of 1024 during 1000 iterations. The best θ is selected by minimizing the loss evaluated every 10 iterations on the training data set.

The estimated Hurst parameter $\widehat{H}$ on B_N^H (resp. $\widetilde{H}$ on $\widetilde{B}_N^H$) is computed using a basic absolute-moment estimation by identifying the slope of $\log t \mapsto \log B_N^H(t)$ (resp. $\log \widetilde{B}_N^H(t)$) [75]. In order to assess the quality of the simulated fBm for different H, Figure 1.7 proposes graphical fBm wavelet representations of one trajectory and its error with a NN fBm simulation. The persistence (resp. non-persistence) phenomena is graphically confirmed on B_N^H and the small error shows that it is also the case for $\widetilde{B}_N^H$ as $H < 1/2$ (resp. $H > 1/2$). Additionally, the excellent approximation of the estimated Hurst parameters $\widetilde{H}$ confirms that the regularity of the NN simulated paths is well preserved. On the other hand in Figure 1.8, we compare the real covariance function (1.2.1) with the one on $\widetilde{B}_N^H$. The small error between the two surfaces illustrates that the covariance function of $\widetilde{B}_N^H$ is also well preserved. To conclude, although this method of simulation may not be optimal in the sense of speed or accuracy, the numerical results highlight a good performance of our proposed generative NN model to simulate realistic fBm paths.



(a) $H = 0.2, \widehat{H} = 0.209$

(b) $\widetilde{H} = 0.208$

(c) $H = 0.4, \widehat{H} = 0.401$

(d) $\widetilde{H} = 0.400$

(e) $H = 0.6, \widehat{H} = 0.600$

(f) $\widetilde{H} = 0.599$

(g) $H = 0.8, \widehat{H} = 0.792$

(h) $\widetilde{H} = 0.790$

Figure 1.7: Simulation of fBm for $H = \{0.2, 0.4, 0.6, 0.8\}$ presented as rows. Left: wavelet representation $t \mapsto V_H^{-1/2} B_N^H(t)$ with the estimated Hurst index $\widehat{H}$. Right: error $t \mapsto V_H^{-1/2}(B_N^H(t) - \widetilde{B}_N^H(t))$ with the estimated Hurst index $\widetilde{H}$ on the NN fBm.

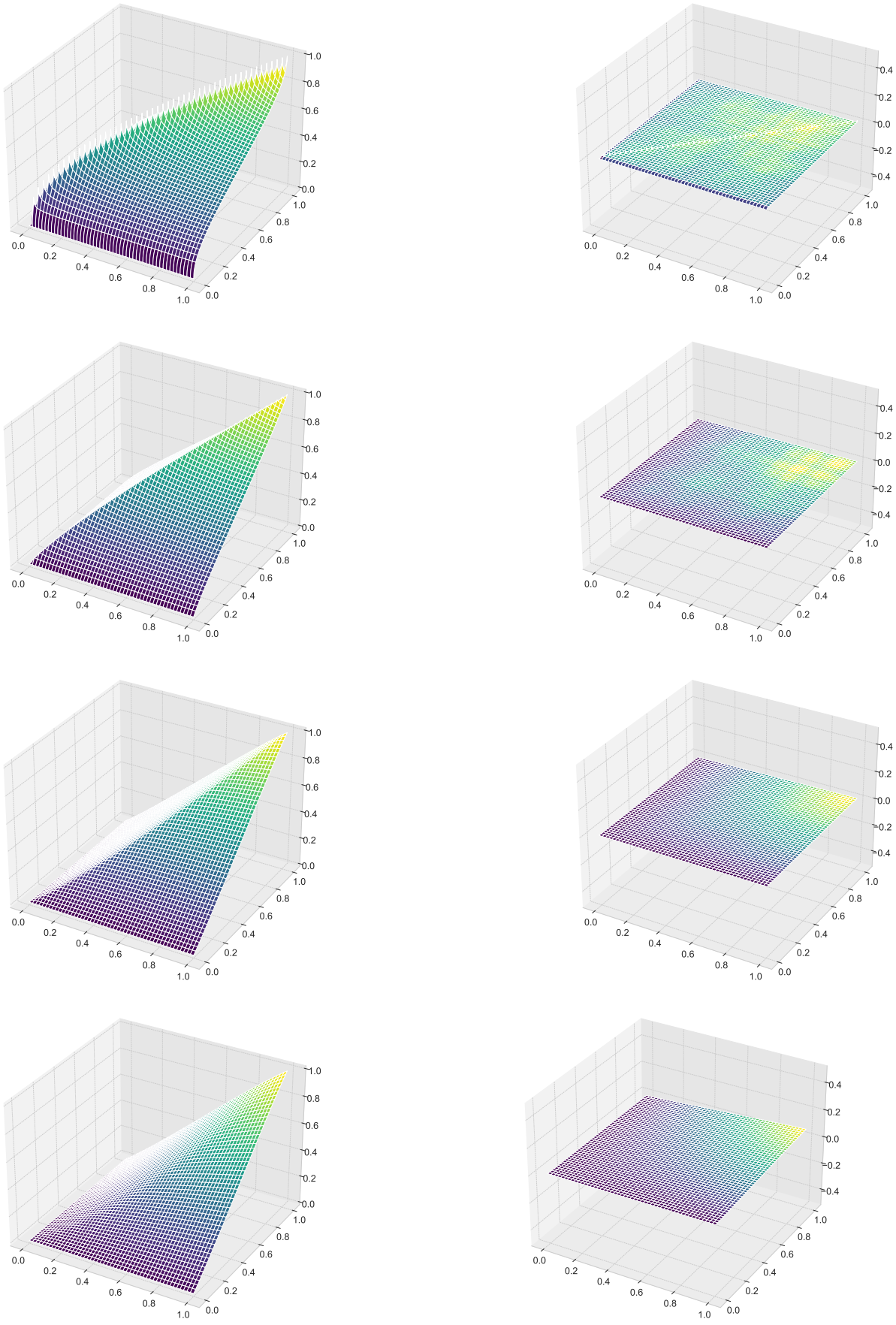


Figure 1.8: fBm covariance surface for $H = \{0.2, 0.4, 0.6, 0.8\}$ presented as rows. Left: real normalized function $(t,s) \mapsto V_H^{-1}\mathbb{C}\mathrm{ov}(B^H(t), B^H(s))$. Right: error $(t,s) \mapsto V_H^{-1}(\mathbb{C}\mathrm{ov}(B^H(t), B^H(s)) - \mathbb{C}\mathrm{ov}(\widetilde{B}_N^H(t), \widetilde{B}_N^H(s)))$ for $(t,s) \in [0,1]^2$.

1.A Complements

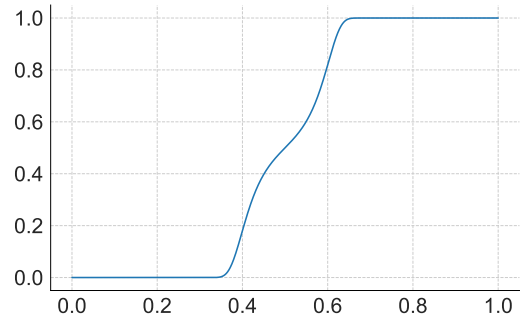
1.A.1 Proof of Lemma 1.2.1

Let us bound the truncated approximation of (1.2.6) for all $n \in \mathbb{N}$,

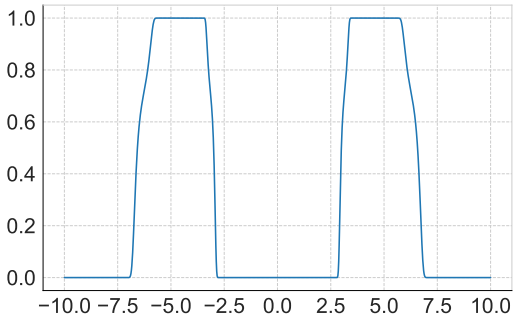
$$\begin{aligned}
&\sup_{t\in[0,1]} \left| B(t) - \left(G_1 t + \sum_{j=0}^{n} \sum_{k=0}^{2^j-1} 2^{-(j+1)} \psi_{j,k}^{\mathrm{FS}}(t) G_{j,k} \right) \right| \\
&\quad = \sup_{t\in[0,1]} \left| \sum_{j=n+1}^{\infty} \sum_{k=0}^{2^j-1} 2^{-(j+1)} \psi_{j,k}^{\mathrm{FS}}(t) G_{j,k} \right| \\
&\quad \leq \sum_{j=n+1}^{\infty} 2^{-(j/2+1)} \sup_{0\leq k\leq 2^j-1} |G_{j,k}| \\
&\quad \leq C \sum_{j=n+1}^{\infty} 2^{-(j/2+1)} \left(\log(j+2^j+1)\right)^{1/2} \quad \text{a.s.} \\
&\quad \leq C \sum_{j=n+1}^{\infty} 2^{-\frac{1}{2}(j+1)} (1+j)^{1/2} \\
&\quad \leq C\ 2^{-\frac{1}{2}(n+1)} (n+1)^{1/2} \\
&\quad \leq C\ N^{-1/2} \big(1+\log(N)\big)^{1/2},
\end{aligned}$$

where C is a non-negative random variable which may change from line to line. In the third line, use the fact that the wavelets have disjoint support in k and so for fixed j, any t belongs to the support of at most one $\psi_{j,k}^{\mathrm{FS}}$, with $\left\|\psi_{j,k}^{\mathrm{FS}}\right\|_{\infty} \leq 2^{j/2}$. In the fourth, invoke [14, Lemma 2]; in the fifth the inequality holds for j large enough; in the sixth use a classical integral test, and lastly replace with N. □

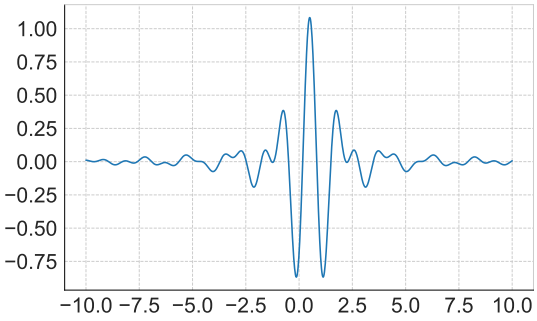
1.A.2 Wavelet representation



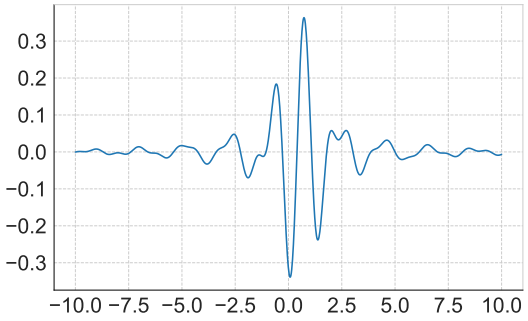
(a) $u \mapsto \nu(u)$



(b) $\xi \mapsto \left|\widehat{\psi}(\xi)\right|$



(c) $x \mapsto \psi(x)$



(d) $x \mapsto \Psi_H(x)$ with $H = 0.2$

Figure 1.9: Lemarié-Meyer wavelet constructed with (1.2.15), (1.2.12), (1.2.10) for $\gamma = 10$.

1.B Harmonizable representation of fBm

1.B.1 Real valued Gaussian measure

Let $f \in \mathbf{L^2}(\mathbb{R}, \mathrm{d}x) = \left\{ f : \mathbb{R} \mapsto \mathbb{R} : \int_{-\infty}^{\infty} |f(x)|^2 \, \mathrm{d}x < \infty \right\}$ and denote $\langle f, g \rangle = \int_{-\infty}^{\infty} f(x)\overline{g(x)} \, \mathrm{d}x$ the inner product in $\mathbf{L^2}(\mathbb{R}, \mathrm{d}x)$ (still valid for complex-valued functions). Define respectively the Fourier transform $\mathcal{F} : f \in \mathbf{L^1}(\mathbb{R}, \mathrm{d}x) \mapsto \widehat{f} = \mathcal{F}(f) : \xi \in \mathbb{R} \mapsto \int_{-\infty}^{\infty} f(x)\mathrm{e}^{-\mathbf{i}x\xi} \, \mathrm{d}x$ and the inverse Fourier transform $\mathcal{F}^{-1} : \varphi \in \mathbf{L^1}(\mathbb{R}, \mathrm{d}\xi) \mapsto \mathcal{F}^{-1}(\varphi) : x \in \mathbb{R} \mapsto \frac{1}{2\pi} \int_{-\infty}^{\infty} \varphi(\xi)\mathrm{e}^{\mathbf{i}x\xi} \, \mathrm{d}\xi$ by the definitions (1.2.4). We recall the Parseval-Plancherel formula for any $f, h \in \mathbf{L^1}(\mathbb{R}, \mathrm{d}x) \cap \mathbf{L^2}(\mathbb{R}, \mathrm{d}x)$:

$$\int_{-\infty}^{\infty} f(x)\overline{h(x)} \, \mathrm{d}x = \frac{1}{2\pi} \int_{-\infty}^{\infty} \widehat{f}(\xi)\overline{\widehat{h}(\xi)} \, \mathrm{d}\xi. \tag{1.B.1}$$

This isometry (up to the constant 2π) allows to extend the Fourier transform $\mathcal{F}$ and its inverse $\mathcal{F}^{-1}$ to square integrable functions, see [137, Section 2.2]. Observe that $\mathcal{F}^{-1}(\varphi)$ is a real-valued function when

$$\varphi \in \mathbf{L^2_r}(\mathbb{R}, \mathrm{d}\xi) = \left\{ \varphi : \mathbb{R} \mapsto \mathbb{C} \text{ s.t. } \int_{-\infty}^{\infty} |\varphi(x)|^2 \, \mathrm{d}x < \infty \text{ and } \varphi(-\xi) = \overline{\varphi(\xi)} \right\},$$

therefore $\mathcal{F}$ is an isometry between $\mathbf{L^2}(\mathbb{R}, \mathrm{d}x)$ and $\mathbf{L^2_r}(\mathbb{R}, \mathrm{d}\xi)$.

Then, see [41, Section 2.1.6], one can define the centered-real valued Gaussian measure by

$$X(\varphi) := \int_{-\infty}^{\infty} \mathcal{F}^{-1}(\varphi)(x)\,\mathrm{d}W(x) \overset{notation}{=} \int_{-\infty}^{\infty} \varphi(\xi)\,\mathrm{d}\widehat{W}(\xi)$$

for $\varphi \in \mathbf{L}^{\mathbf{2}}_{\mathbf{r}}(\mathbb{R}, \mathrm{d}\xi)$, such that

$$\mathbb{C}\mathrm{ov}\left(X(\varphi_1), X(\varphi_2)\right) = \int_{-\infty}^{\infty} \mathcal{F}^{-1}(\varphi_1)(x)\mathcal{F}^{-1}(\varphi_2)(x)\,\mathrm{d}x. \tag{1.B.2}$$

If $(e_n)_n$ is an orthonormal basis of $\mathbf{L}^{\mathbf{2}}(\mathbb{R}, \mathrm{d}x)$, then

$$X(\varphi) = \sum_n \left\langle e_n,\, \mathcal{F}^{-1}(\varphi)\right\rangle G_n \tag{1.B.3}$$

where $G_n = \int_{-\infty}^{\infty} e_n(x)\,\mathrm{d}W(x)$ are i.i.d. standard Gaussian random variables.

1.B.2 Series representation of fBm

For $t \in [0,1]$, take $\varphi_t(\xi) := \frac{\mathrm{e}^{\mathbf{i}t\xi}-1}{(\mathbf{i}\xi)^{H+1/2}} \in \mathbf{L}^{\mathbf{2}}_{\mathbf{r}}(\mathbb{R}, \mathrm{d}\xi)$ and set

$$B^H(t) := \int_{-\infty}^{\infty} \varphi_t(\xi)\,\mathrm{d}\widehat{W}(\xi). \tag{1.B.4}$$

We now verify that this harmonizable representation leads to the fBm-representation (1.2.9), (1.2.10), (1.2.11) consistently with the covariance formula (1.2.1).

For this, as an orthonormal basis of $\mathbf{L}^{\mathbf{2}}(\mathbb{R}, \mathrm{d}x)$, consider the basis functions generated by mother wavelet function ψ: $\left\{\psi_{j,k}(x) = 2^{j/2}\psi(-2^j x - k)\right\}_{(j,k)\in\mathbb{Z}^2}$. Note that usually, to a given mother wavelet ψ, one invokes the function basis $\left\{\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)\right\}_{(j,k)\in\mathbb{Z}^2}$: our choice of changing the sign in front of x is made for getting exactly the representation (1.2.9), (1.2.10) at the end, of course the sign change does not affect the orthonormality property of the basis as it can be easily checked. In view of (1.B.3) and (1.B.1), we have to compute the coefficients

$$c_{j,k}(t) := \left\langle \psi_{jk},\, \mathcal{F}^{-1}(\varphi_t)\right\rangle = \frac{1}{2\pi}\int_{-\infty}^{\infty} \widehat{\psi_{jk}}(\xi)\overline{\varphi_t(\xi)}\,\mathrm{d}\xi. \tag{1.B.5}$$

A direct computation gives

$$\begin{aligned}
\widehat{\psi_{j,k}}(\xi) &= 2^{j/2}\int_{-\infty}^{\infty} \mathrm{e}^{-\mathbf{i}x\xi}\psi(-2^j x - k)\,\mathrm{d}x = 2^{-j/2}\mathrm{e}^{\mathbf{i}\xi k 2^{-j}}\widehat{\psi}(-2^{-j}\xi),\\
c_{j,k}(t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{\mathrm{e}^{-\mathbf{i}\xi t}-1}{(-\mathbf{i}\xi)^{H+1/2}} 2^{-j/2}\mathrm{e}^{\mathbf{i}\xi k 2^{-j}}\widehat{\psi}(-2^{-j}\xi)\,\mathrm{d}\xi\\
&= \frac{2^{-jH}}{2\pi}\int_{-\infty}^{\infty} \frac{\mathrm{e}^{\mathbf{i}\omega 2^j t}-1}{(\mathbf{i}\omega)^{H+1/2}}\mathrm{e}^{-\mathbf{i}\omega k}\widehat{\psi}(\omega)\,\mathrm{d}\omega \quad (\text{setting } \omega = -2^{-j}\xi)\\
&= 2^{-jH}\left(\Psi_H\left(2^j t - k\right) - \Psi_H\left(-k\right)\right)
\end{aligned}$$

recalling the definition (1.2.10) of Ψ_H. All in all, from (1.B.3) we have obtained

$$B^H(t) = \sum_{j=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} 2^{-jH}\left(\Psi_H\left(2^j t - k\right) - \Psi_H\left(-k\right)\right) G_{j,k}.$$

We get the announced representation (1.2.9).

1.B.3 Covariance of B^H

We now establish (1.2.1), with V_H given by (1.2.11). We first state a useful formula.

Lemma 1.B.1. *Let $\beta \in (0,2)$. Then for all $t \in [0,1]$,*

$$\int_0^\infty \frac{1-\cos(ut)}{u^{\beta+1}}\,\mathrm{d}u = 2^{-\beta}\sqrt{\pi}\frac{t^\beta}{\beta}\frac{\Gamma\left(1-\frac{\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)}.$$

Proof. Starting from

$$\int_0^\infty \frac{1-\cos(ut)}{u^{\beta+1}}\,\mathrm{d}u = t^\beta\int_0^\infty \frac{1-\cos(y)}{y^{\beta+1}}\,\mathrm{d}y = \frac{t^\beta}{\beta}\int_0^\infty \frac{\sin(y)}{y^\beta}\,\mathrm{d}y,$$

by change of variable $y = ut$, integrating by part and removing the first term since $\beta \in (0,2)$ and $1-\cos(y) \sim y^2/2$ as $y \to 0$. Recognizing a known integral function [157, Equation (13) p. 387] finishes the proof. □

Now, let $0 \le s \le t \le 1$. Starting from (1.B.4), and using (1.B.2)-(1.B.1), we get

$$\begin{aligned}\mathbb{C}\mathrm{ov}\left(B^H(t),B^H(s)\right) &= \frac{1}{2\pi}\int_{-\infty}^\infty \frac{\left(\mathrm{e}^{\mathrm{i}\xi t}-1\right)\left(\mathrm{e}^{-\mathrm{i}\xi s}-1\right)}{|\xi|^{2H+1}}\,\mathrm{d}\xi \\ &= \frac{1}{\pi}\int_0^\infty \frac{(1-\cos(\xi t))+(1-\cos(\xi s))-(1-\cos(\xi(t-s)))}{|\xi|^{2H+1}}\,\mathrm{d}\xi.\end{aligned}$$

Using Lemma 1.B.1 with $\beta = 2H \in (0,2)$ entails

$$\mathbb{C}\mathrm{ov}\left(B^H(t),B^H(s)\right) = \frac{2^{-2H}}{2H\sqrt{\pi}}\frac{\Gamma(1-H)}{\Gamma(\frac{1}{2}+H)}\left(|t|^{2H}+|s|^{2H}-|t-s|^{2H}\right).$$

The Euler reflection and the Legendre duplication formulas write for all $a \notin \mathbb{Z}$:

$$\Gamma(1-a)\Gamma(a) = \frac{\pi}{\sin(\pi a)}, \quad \Gamma\left(\frac{1}{2}+a\right)\Gamma(a) = 2^{1-2a}\sqrt{\pi}\Gamma(2a).$$

Therefore, we retrieve (1.2.1) with

$$\mathbb{V}\mathrm{ar}\left(B^H(1)\right) = V_H = \frac{2^{-2H}}{H\sqrt{\pi}}\frac{\Gamma(1-H)}{\Gamma(\frac{1}{2}+H)} = \frac{1}{2H\sin(\pi H)\Gamma(2H)} \tag{1.B.6}$$

as announced. □

Chapter 2

EV-GAN: Simulation of extreme events with ReLU neural networks

Note. The results of this chapter are based on the paper [6].

Abstract. Feedforward neural networks based on Rectified linear units (ReLU) cannot efficiently approximate quantile functions which are not bounded, especially in the case of heavy-tailed distributions. We thus propose a new parametrization for the generator of a Generative adversarial network (GAN) adapted to this framework, basing on extreme-value theory. An analysis of the uniform error between the extreme quantile and its GAN approximation is provided: We establish that the rate of convergence of the error is mainly driven by the second-order parameter of the data distribution. The above results are illustrated on simulated data and real financial data. It appears that our approach outperforms the classical GAN in a wide range of situations including high-dimensional and dependent data.

2.1 Introduction

Context of risks. Analyzing extreme events is an important issue in economics, engineering, and life sciences, among other fields, with significant applications such as actuarial risks [12], communication network reliability [162], aircraft safety [155], analysis of epidemics, and so forth... In the last two decades, it has taken even more importance in financial risk management, because of the increasing number of shocks and financial crises. Among the wide range of exercises in this field, stress test [65] has become a main guideline for the regulator in order to assess the banking system resilience against the realizations of various categories of risk (market, credit, operational, climate, etc). To this end, numerical simulation of unfavorable extreme (but plausible) scenarios is a major tool to study the consequences on these risks. Given a stochastic model of risks, various sampling schemes are available (for instance, using importance sampling [29, Chapter 4], MCMC with splitting – [90], or interacting particles system – [55]), with the potential advantage of reducing the statistical fluctuation over a naive Monte Carlo method. Though presumably more informative for a given number M of samples, these methods suffer from a higher computational complexity (notably in high dimension): Thus, one might wonder how to get extra samples in an efficient way, by leveraging the previous M samples. Somehow, the situation is similar to a case where the previous samples are viewed as observed data and where we seek a data-driven method able to sample similarly to that empirical distribution, without necessarily the knowledge of the sampling method that has generated the observed data. This corresponds to the recent paradigm of Generative adversarial network (GAN) models initiated by [100] or of Variational autoencoder (VAE) by [115]. The novelty in our work is relative to the context of risks, where we are interested in a generative data-based model able

to reproduce – with high-fidelity – specific extreme statistical properties of a data set, while being fast in the simulation phase. This challenge arises both in the context of true historical data sets (see experiments in Section 2.4) or when the learning data set has been generated by sophisticated Monte Carlo methods.

Background results. Generally speaking, different types of generative models have been developed lately [71] and in this work, we focus on GANs, which have gained a tremendous popularity from the original work of [100] and its extension using the Wasserstein distance [10]. Kuratowski Theorem [19, Chapter 7]-[178, p.8] ensures that any random variable X on $\mathbb{R}^d$ (and more generally on a Polish space) can be obtained by

$$X \overset{d}{=} G(Z) \tag{2.1.1}$$

for some measurable function G and some latent random variable Z in dimension d' (see Lemma 2.B.1 in the Appendix for a constructive proof with $Z \sim \mathcal{U}([0,1])$ and $d' = 1$) such that for each mth marginal, $m \in \{1, \dots, d\}$, one has $\tilde{X}^{(m)} := G^{(m)}(Z) \overset{d}{=} X^{(m)}$. This result is one key to understand the ability of GANs to simulate realistic samples in a space of high dimension d, starting from a latent space of moderate dimension d'. In practice, the selection of this latent dimension is an open problem in the generative neural networks literature. A GAN scheme is aimed at approximating the unknown G through a parametric family of neural networks (NN) $\mathcal{G} = \{G_\theta : \mathbb{R}^{d'} \to \mathbb{R}^d, \theta \in \Theta\}$ and to learn the optimal parameter $\theta^\star$ from a data set $\{X_i \in \mathbb{R}^d, i = 1, \dots, n\}$ of i.i.d samples from an unknown distribution p_X. It is performed by optimizing an objective function which can be interpreted as an adversarial game between a generator and a discriminator chosen in a parametric family of functions $\mathcal{D} = \{D_\phi : \mathbb{R}^d \to [0,1], \phi \in \Phi\}$. In other words, $D_\phi(x)$ represents the probability that an observation x is drawn from p_X. Both the generator and the discriminator are NNs with opposite objectives: The former tries to mimic real data which seem likely by the discriminator, while the latter tries to distinguish between the two sources. In [100], this optimization problem is defined as:

$$\min_{\theta\in\Theta} \max_{\phi\in\Phi} \left[\mathbb{E}_{p_X}\left(\log D_\phi(X)\right) + \mathbb{E}_{p_Z}\left(\log\left(1 - D_\phi\left(G_\theta(Z)\right)\right)\right)\right].$$

Theoretical results on GANs have been established in [22, 24, 103], see also [159, 186] for the generation of financial time-series and [7] for the generation of fractional Brownian motion.

Extreme events generation using GANs has been investigated in Finance [186], in meteorology [20], in cosmological analysis [68] and in anomaly detection [57]. One strategy is to learn a light-tail model on some transformed data and then transform back the generator outputs for recovering the heavy-tailed data property. Possible transformations include the Lambert W function [186]. Another approach is to use directly a heavy-tailed latent variable in the GAN setting [68, 111]. It is shown in [111] that generator outputs follow the desired heavy-tailed distribution. Alternative metric spaces are also introduced to ensure the loss function to be finite. To be effective, the method however requires the accurate estimation of the tail-index associated with each heavy-tailed marginal distribution, which is a challenging task in extreme-value theory, see next paragraph for details: As a main difference, our approach does not require the estimation of tail-indices. Alternatively, in [20], a distribution shifting is first introduced in order to reduce the lack of training data in the extreme tails. Second, a GAN parametrization conditioned by samples drawn from a generalized Pareto distribution is fitted to the shifted data. Finally, an additional term representing some distance to a desired extremeness is added to the loss function. Although numerical results on images are promising, we do not think that the proposed parametrization gives theoretical support for generating extreme observations in the sense that no error or complexity bounds are provided in the NN architecture of the generator.

Our contributions. In a GAN setting, our purpose is to cope with two prominent issues, that are mostly related to extreme-value theory. First, the number of data available in extreme regions must be relatively small, by definition (even in the case of data that are output of sophisticated sampling methods). Second, we restrict to the challenging situation of heavy-tailed distributions (in the Fréchet maximum domain of attraction), where by definition, extreme data take very large values. Therefore, the usual GAN approach cannot work, as we now explain (and as the reader will check from our numerical experiments in Section 2.4). Consider for a while the case $d = d' = 1$ and say that G in (2.1.1) is approximated by a ReLU NN under the form

$$G_\theta(z) = \sum_{j=1}^{J} a_j \sigma(w_j z + b_j), \tag{2.1.2}$$

where $\sigma(x) := \max(x, 0)$ is the ReLU function, $\theta = \{(a_j, w_j, b_j),\, j = 1, \dots, J\} \in \Theta = \mathbb{R}^{3J}$ and J is the number of units in the hidden layer. On the one-hand, if the latent random variable Z were bounded, the output would be bounded [111, Proposition 1] and by no means, it would be a good candidate for fitting the distribution of the unbounded random variable X. On the other hand, taking for Z a Gaussian vector as it is often chosen, for example in [20], would lead to a light-tailed distribution for $G_\theta(Z)$ [111, Theorem 1] since G_θ is sublinear w.r.t. the input [180], whereas we focus on the heavy-tail case. Similar arguments are given in [185, Theorem 1] to emphasize that the generator cannot generate samples with heavier distribution than its inputs. Clearly, such a parameterization (2.1.2) of the generator cannot be efficient as far as extreme values are concerned. Note that deeper NN would not overcome this issue either.

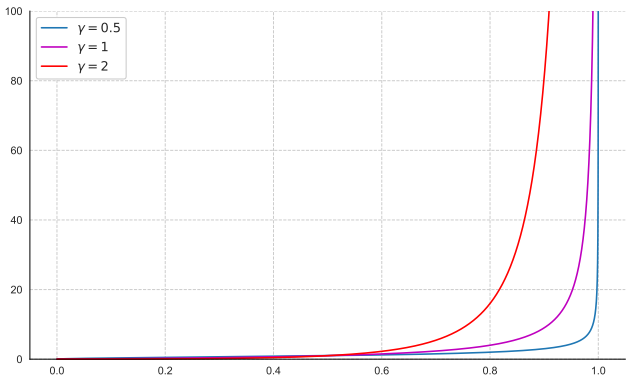


Figure 2.1: Quantile function associated with the Burr distribution $u \in (0, 1) \mapsto q_X(u)$ with tail-index $\gamma \in \{1/2, 1, 2\}$ and second-order parameter $\rho = -1$, see Table 2.4 for the parameterization.

To introduce our new parametrization called Extreme-Value GAN (EV-GAN), consider first a real random variable X with cumulative distribution function F_X defined on $\mathbb{R}$. The inversion method by Von Neumann [60] gives that one can set $G(u) := q_X(u) := \inf\{x : F_X(x) \geq u\}$ with $U \sim \mathcal{U}([0, 1])$. Since we shall focus on distributions in the Fréchet maximum domain of attraction [52, Theorem 1.2.1] with positive tail-index γ, the associated survival function $\bar{F}_X(x) := 1 - F_X(x)$ decays at rate $x^{-1/\gamma}$ when $x \to \infty$, which implies that $q_X(u)$ diverges as $u \to 1$ at rate $(1 - u)^{-\gamma}$. The tail-index γ is thus the main driver of the behavior of extreme quantiles, see Figure 2.1 for an illustration. To be in a position to apply results such as the Universal approximation theorem [46] (any continuous function on $[0, 1]$ can be approximated with arbitrary precision by a one hidden layer NN), we shall transform the quantile function to avoid divergence in the neighborhood of $u = 1$. To this end, for all $(u, y) \in [0, 1) \times (0, \infty)$, let

$$H_u(y) = -\log(y) \,/ (\log(1 - u^2) - \log(2)) \qquad \text{and} \qquad f^{\mathrm{TIF}}(u) = H_u(q_X(u)). \tag{2.1.3}$$

It will appear in the sequel that f^{TIF} is continuous on $[0,1]$ for all F_X in the Fréchet maximum domain of attraction, with $f^{\mathrm{TIF}}(u) \to \gamma$ as $u \to 1$; f^{TIF} is thus referred to as the Tail-index function (TIF). Therefore, a ReLU NN could well approximate f^{TIF} thanks to the Universal approximation theorem, but to get even better approximation, we shall consider a correction of the Tail-index function:

$$f^{\mathrm{CTIF}}(u) = f^{\mathrm{TIF}}(u) - \sum_{k=1}^{6} \kappa_k e_k(u), \quad u \in [0,1],$$

which enjoys higher regularity in the neighborhood of $u = 1$. See Paragraph 2.2.2 for a definition of functions $e_1, \ldots, e_6$ and coefficients $\kappa_1, \ldots, \kappa_6$: The functions $e_1, \ldots, e_6$ are universal (see (2.2.13)) and as such, they do not depend on the distribution parameters, only coefficients $\kappa_1, \ldots, \kappa_6$ may depend on them. Now use a NN to approximate the smooth function f^{CTIF}, deduce an approximation of f^{TIF}, and of the quantile function by composing with H_u^{-1} for each u (in view of (2.1.3)): All in all, we obtain the so-called EV-GAN parametrization defined for all $(z,x) \in [0,1] \times (0,\infty)$ as

$$G_\psi^{\mathrm{TIF}}(z) = H_z^{-1}\left(\sum_{j=1}^{J} a_j \sigma(w_j z + b_j) + \sum_{k=1}^{6} \kappa_k e_k(z)\right), \tag{2.1.4}$$

$$\text{with} \quad H_z^{-1}(x) := \left(\frac{1-z^2}{2}\right)^{-x}. \tag{2.1.5}$$

In the multidimensional setting $d > 1$ and $d' > 1$, our strategy of approximation consists in preserving the same parametric form for each marginal component, and in mixing the latent components to generate dependence between the d coordinates (see Corollary 2.2.5): The m-th coordinate will take the form, with $z = (z^{(1)}, \ldots, z^{(d)})$,

$$G_\psi^{\mathrm{TIF},(m)}(z^{(1)}, \ldots, z^{(d')}) = H_{z^{(m)}}^{-1}\left(\sum_{j=1}^{J} a_j^{(m)} \sigma\left(\sum_{i=1}^{d'} w_j^{(i)} z^{(i)} + b_j\right) + \sum_{k=1}^{6} \kappa_k^{(m)} e_k\left(z^{(m)}\right)\right). \tag{2.1.6}$$

Let us highlight that, in (2.1.6), the mth coordinate of the generator $G_\psi^{\mathrm{TIF}}(z)$ involves the mth coordinate of z which is a $d'-$ dimensional vector. The above construction of the EV-GAN generator thus constraints the latent dimension to be larger than the dimension of the data: $d' \geq d$. The architecture of the associated NN is illustrated on Figure 2.2 in the case $d = 2$ and $d' = 3$. We prove in Theorem 2.2.1 that the above EV-GAN parametrization converges uniformly coordinate-wise, in the log-scale of the $H_\cdot$-transform. Joint convergence for all coordinates is an open question, which is related to the delicate notion of upper tail dependence. However, numerical experiments in multidimensional settings fully support the relevance of this parametrization. We observe that tail dependencies are extremely well reproduced.

The rest of the paper is organized as follows. The transformation of the quantile function q_X associated with an heavy-tailed distribution F_X into a regular function f^{CTIF} is presented in Section 2.2: Under a second-order assumption, we show that f^{CTIF} can be uniformly approximated by a one hidden layer NN with some rate depending on the second-order parameter ρ, which plays a crucial role in extreme-value theory. Auxiliary results and technical proofs are postponed to the Appendix. The performance of the method is illustrated on simulated data (Section 2.4) and real financial data (Section 2.5). It is shown that, in both experiments, our approach largely outperforms the classic GAN method. Some conclusions and directions of future research are discussed in Section 2.6.

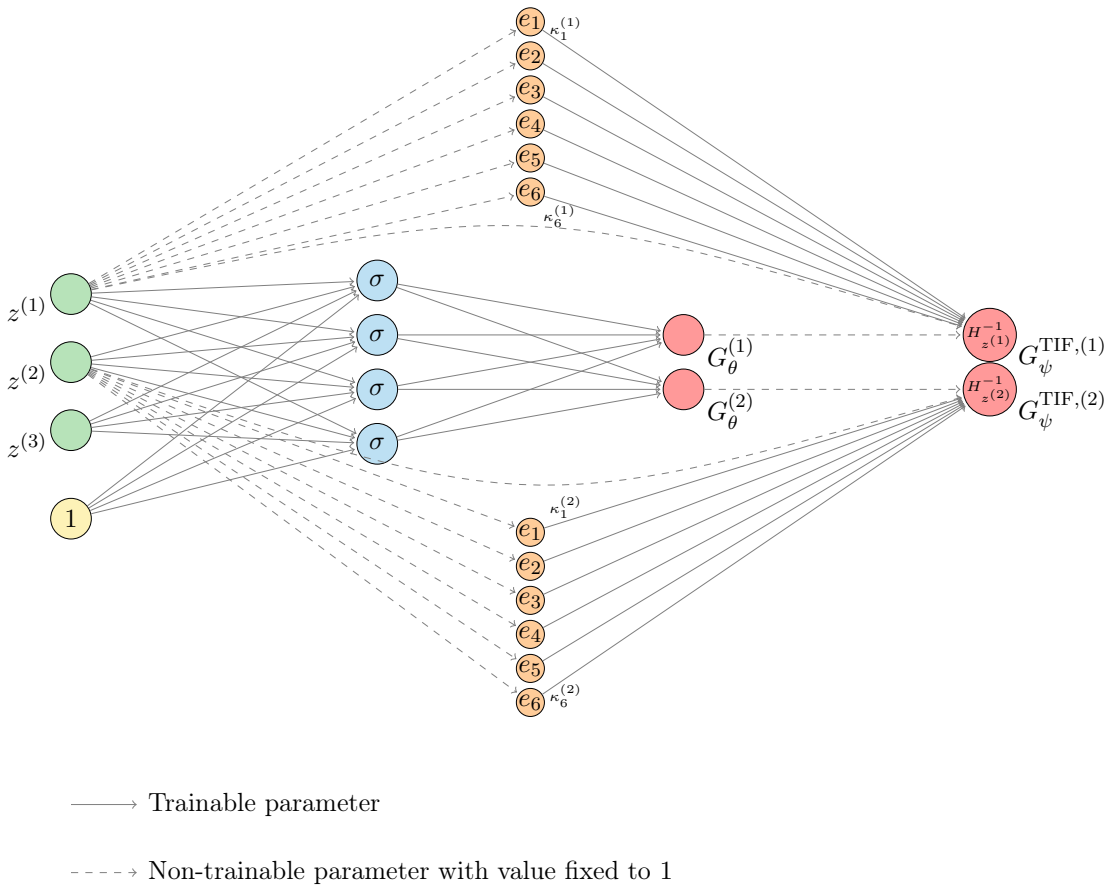


Figure 2.2: Generator of the EV-GAN with one hidden layer, $d' = 3$ and $d = 2$.

2.2 Main results

First, the construction of the proposed transformation of the quantile function is developed and its approximation by a ReLU NN is then investigated.

2.2.1 TIF regularity

In this section, we discuss the construction and the extension of (2.1.3). The objective is to build a tail-index function which may be well approximated by a NN. Let X be a real random variable and denote by F_X its cumulative distribution function supposed to be continuous and strictly increasing. We focus on the case of heavy-tailed distributions, *i.e.* when F_X is attracted to the maximum domain of Pareto-type distributions with tail-index $\gamma > 0$. From [26], the survival function $\bar{F}_X := 1 - F_X$ of such a heavy-tailed distribution can be expressed as

($\mathbf{H_1}$): $\bar{F}_X(x) = x^{-1/\gamma}\ell_X(x)$, where ℓ_X is a slowly-varying function at infinity *i.e.* such that $\ell_X(\lambda x)/\ell_X(x) \to 1$ as $x \to \infty$ for all $\lambda > 0$.

In such a case, $\bar{F}_X$ is said to be regularly-varying with index $-1/\gamma$ at infinity, which is denoted for short by $\bar{F}_X \in RV_{-1/\gamma}$. Similarly, we shall note $\ell_X \in RV_0$. The tail-index γ tunes the tail heaviness of the distribution function F_X. Assumption ($\mathbf{H_1}$) is recurrent in risk assessment, since actuarial and financial data are most of the time heavy-tailed, see for instance the recent studies [9, 35] or the monographs [63, 160]. The Pareto distribution is the simplest example of heavy-tailed distribution, since, in this case, ℓ_X in ($\mathbf{H_1}$) is constant. See Table 2.4 for more sophisticated examples. As a consequence of the above assumptions, the tail quantile function $x \mapsto q_X(1 - 1/x)$ is regularly-varying with index γ at infinity, see [52, Proposition B.1.9.9], or,

equivalently,

$$q_X(u) = (1-u)^{-\gamma} L\left(\frac{1}{1-u}\right), \tag{2.2.1}$$

for all $u \in (0,1)$ with $L \in RV_0$. Without of loss of generality, one can assume that $\eta := \mathbb{P}(X \geq 1) \neq 0$ and, since, we focus on the upper tail behavior of X, introduce the random variable $Y = X$ given $X \geq 1$. It follows that the quantile function of Y is given by

$$q_Y(u) = q_X(1-(1-u)\eta), \tag{2.2.2}$$

for all $u \in (0,1)$. Note that one could also assume $X \geq 1$ and set $\eta = 1$ in order to simplify the following derivations. Finally, we consider the Tail-index function (TIF) obtained by plugging (2.2.2) into (2.1.3):

$$f^{\mathrm{TIF}}(u) = -\frac{\log q_X\big(1-(1-u)\eta\big)}{\log(1-u^2) - \log 2}, \tag{2.2.3}$$

for all $u \in (0,1)$. Extra assumptions on F_X, or equivalently on L, are necessary such that f^{TIF} is differentiable. Consider the Karamata representation of the slowly-varying function L [52, Definition B1.6]:

$$L(x) = c(x) \exp\left(\int_1^x \frac{\varepsilon(t)}{t}\,\mathrm{d}t\right), \tag{2.2.4}$$

where $c(x) \to c_\infty$ as $x \to \infty$ and ε is a measurable function such that $\varepsilon(x) \to 0$ as $x \to \infty$. Our second main assumption then writes:

($\mathbf{H_2}$): $c(x) = c_\infty > 0$ for all $x \geq 1$ and $\varepsilon(x) = x^\rho \ell(x)$ with $\ell \in RV_0$ and $\rho < 0$.

The assumption that c is a constant function is equivalent to assuming that L is normalized [116] and ensures that L is differentiable. As noted in [26], the normalization assumption is not restrictive since slowly-varying functions are of interest only to within asymptotic equivalence. The condition $\varepsilon \in RV_\rho$ with $\rho < 0$ entails that $L(x) \to L_\infty \in (0,\infty)$ as $x \to \infty$. The index of regular variation ρ is referred to as the second-order parameter. It is the main driver of the bias in the estimation of extreme quantiles from heavy-tailed distributions, see Table 2.4 for values of ρ associated with usual distributions. Besides, ($\mathbf{H_2}$) entails that F_X satisfies the so-called second-order condition which is the cornerstone of all proofs of asymptotic normality in extreme-value statistics. Interpretations and examples may be found in [16] and [52]. We also refer to [79, 81] where a similar assumption is introduced in the framework of conditional extremes. Similarly, we shall also consider the assumption:

($\mathbf{H_3}$): ℓ is normalized.

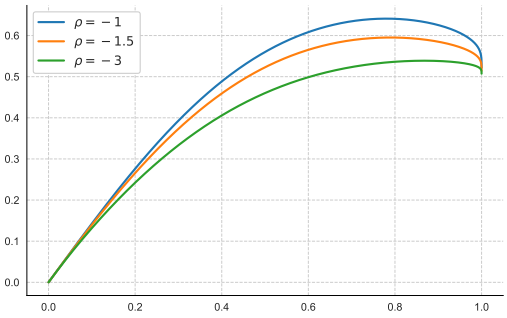
The latter condition ensures that ℓ is differentiable on $(0,1)$ and thus that L and q_X are twice differentiable on $(0,1)$. Regularity properties of the above TIF can then be established, see Figure 2.3 for an illustration on the Burr distribution defined in Table 2.4.

Proposition 2.2.1.

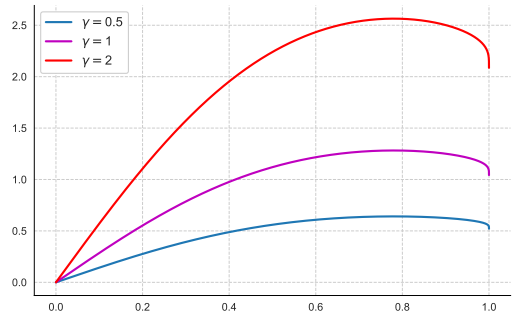
(i) If ($\mathbf{H_1}$) *holds, then* f^{TIF} *is a continuous and bounded function on* $[0,1]$, $f^{\mathrm{TIF}}(0) = 0$ *and* $f^{\mathrm{TIF}}(u) \to \gamma$ *as* $u \to 1$.

(ii) If, moreover, ($\mathbf{H_2}$) *holds, then* f^{TIF} *is continuously differentiable on* $(0,1)$ *and*

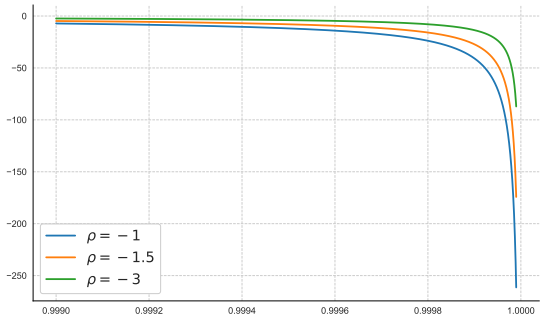
$$\begin{aligned}
\partial_u f^{\mathrm{TIF}}(0) &= \frac{\gamma + \varepsilon\,(1/\eta)}{\log(2)}, \\
\partial_u f^{\mathrm{TIF}}(u) &= \sum_{j=0}^{3} c_j \varphi_j(u) - \frac{\varepsilon\left(\frac{1}{(1-u)\eta}\right)}{(1-u)\log(1-u)}\,(1+o(1)) + \mathcal{O}\left(\frac{(1-u)}{\log(1-u)}\right),
\end{aligned} \tag{2.2.5}$$



(a) $\gamma = 1/2$ and $\rho \in \{-3, -3/2, -1\}$



(b) $\rho = -1$ and $\gamma \in \{1/2, 1, 2\}$



(c) $\gamma = 1/2$ and $\rho \in \{-3, -3/2, -1\}$

Figure 2.3: (a,b): Tail-index function $u \in (0,1) \mapsto f^{\mathrm{TIF}}(u)$ associated with a Burr distribution for different values of tail-index γ and second-order parameter ρ, see Table 2.4 for parameterization details. (c): First derivative of Tail index function $u \in (0,1) \mapsto \partial_u f^{\mathrm{TIF}}(u)$.

as $u \to 1$, *where* $c_0 = c_3 = \beta$, $c_1 = -\gamma/2$, $c_2 = (\gamma - \beta)/2$, $\beta = \gamma \log \eta - \log L_\infty$,

$$\varphi_0(u) = \frac{1}{(1-u)\left(\log(1-u)\right)^2} \text{ and } \varphi_j(u) = \frac{1}{\left(\log(1-u)\right)^j},\ u \in (0,1),\ j = 1,2,3.$$

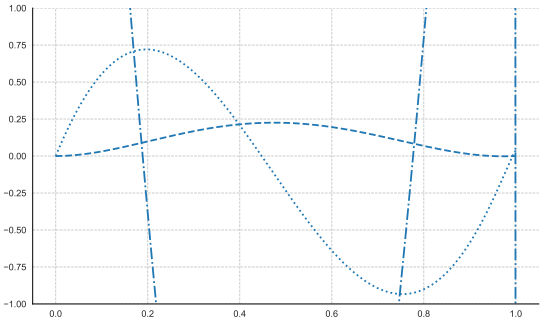
It appears from Proposition 2.2.1(i) that, in contrast to the quantile function, the TIF is bounded on $[0,1]$. Remark that considering the simpler form $H_u(\cdot) = -\log(\cdot)/\log(1-u)$ in (2.1.3) would circumvent the problem as $u \to 1$ but would introduce an artifical singularity as $u \to 0$. Let us also highlight that $\varphi_0(u) \to \infty$ as $u \to 1$ making the first derivative of f^{TIF} unbounded as $u \to 1$, see Figure 2.3c for an illustration on the Burr distribution. Besides, $\varphi_j(u) \to 0$ as $u \to 1$ for all $j \in \{1,2,3\}$ while the second term in (2.2.5) tends to 0 if $\rho < -1$ or tends to ∞ if $\rho > -1$. Moreover, it is readily seen that $\partial_u \varphi_j(u) \to \infty$ as $u \to 1$ for all $j \in \{0, \ldots, 3\}$. As a conclusion, in the case where $\rho < -1$, Proposition 2.2.1(ii) suggests to build a twice differentiable version of f^{TIF} by removing the φ_j components, $j \in \{0, \ldots, 3\}$, in the neighborhood of $u = 1$. To this end, consider

$$f^{\mathrm{CTIF}}(u) := f^{\mathrm{TIF}}(u) - g(u) \sum_{j=0}^{3} c_j \Phi_j(u) - \gamma g(u) - \partial_u f^{\mathrm{TIF}}(0) h(u), \tag{2.2.6}$$

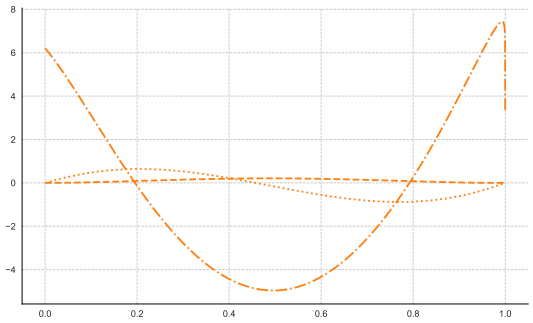
with, for all $u \in (0,1)$,

$$\begin{cases} g(u) = -4u^5 + 5u^4, \\ h(u) = u^3 - 2u^2 + u, \\ \Phi_0(u) = \varphi_1(u), \\ \Phi_1(u) = -\operatorname{li}(1-u), \\ \Phi_2(u) = \Phi_1(u) + (1-u)\varphi_1(u), \\ \Phi_3(u) = \Big(\Phi_1(u) + (1-u)(\varphi_1(u) + \varphi_2(u))\Big)/2. \end{cases} \tag{2.2.7}$$

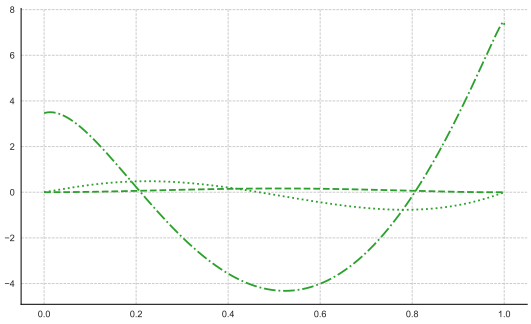
Here, $\operatorname{li}(\cdot)$ denotes the logarithmic integral function defined as $\operatorname{li}(x) := \int_0^x \frac{1}{\log(t)}\,\mathrm{d}t$ for all $0 < x < 1$, with $\operatorname{li}(0) = 0$ and $\operatorname{li}(x) \to -\infty$ as $x \to 1$. Let us remark that $g(\cdot)$ and $h(\cdot)$ are two Hermite spline functions and that, by construction, $\partial_u \Phi_j(u) = \varphi_j(u)$, for all $j \in \{0, \dots, 3\}$. The second term in (2.2.6) thus aims at removing the singular components in the first and second derivative of the TIF function in the neighborhood of $u = 1$. The additional terms $\gamma g(u)$ and $\partial f^{\mathrm{TIF}}(0) h(u)$ ensure that the TIF function as well as its first derivative vanish at $u = 0$. Regularity properties of f^{CTIF} are established in the next Proposition and illustrated on Figure 2.4 in the case of a Burr distribution.



(a) CTIF with $\gamma = 1/2$ and $\rho = -1$



(b) CTIF with $\gamma = 1/2$ and $\rho = -3/2$



(c) CTIF with $\gamma = 1/2$ and $\rho = -3$

Figure 2.4: Illustration of the regularity properties of CTIF on a Burr distribution with $\gamma = 1/2$ and $\rho \in \{-3, -3/2, -1\}$. Corrected tail-index function $u \in (0,1) \mapsto f^{\mathrm{CTIF}}(u)$ (dashed line) and its first two derivatives $u \in (0,1) \mapsto \partial_u f^{\mathrm{CTIF}}(u)$ (dotted line) and $u \in (0,1) \mapsto \partial^2_{uu} f^{\mathrm{CTIF}}(u)$ (dash-dot line).

Proposition 2.2.2.

(i) If $(\mathbf{H_1})$ *holds, then*

$$\lim_{u\to 0} f^{\mathrm{CTIF}}(u) = \lim_{u\to 1} f^{\mathrm{CTIF}}(u) = 0. \tag{2.2.8}$$

(ii) If, moreover, $(\mathbf{H_2})$ *holds with* $\rho < -1$*, then* f^{CTIF} *is continuously differentiable on* $[0,1]$ *and*

$$\lim_{u\to 0} \partial_u f^{\mathrm{CTIF}}(u) = \lim_{u\to 1} \partial_u f^{\mathrm{CTIF}}(u) = 0. \tag{2.2.9}$$

(iii) If, moreover, $(\mathbf{H_3})$ *holds, then* f^{CTIF} *is twice continuously differentiable on* $[0,1)$ *and*

$$\partial^2_{uu} f^{\mathrm{CTIF}}(u) = 20\gamma - 2\left(\frac{\gamma+\varepsilon(1/\eta)}{\log(2)}\right) - \frac{(1+\rho)\varepsilon\left(\frac{1}{\eta(1-u)}\right)}{(1-u)^2\log(1-u)}\left(1+o(1)\right) + \mathcal{O}\left(\frac{1}{\log(1-u)}\right), \tag{2.2.10}$$

as $u \to 1$ *and*

$$\lim_{u\to 0} \partial^2_{uu} f^{\mathrm{CTIF}}(u) = \frac{5\gamma + \varepsilon(1/\eta)\left(5+\rho+\frac{1}{\eta}\frac{\partial \ell(1/\eta)}{\ell(1/\eta)}\right)}{\log(2)} - 5\beta. \tag{2.2.11}$$

(iv) If, moreover, $\rho < -2$*, then* f^{CTIF} *is twice continuously differentiable on* $[0,1]$ *and*

$$\lim_{u\to 1} \partial^2_{uu} f^{\mathrm{CTIF}}(u) = 20\gamma - 2\left(\frac{\gamma+\varepsilon(1/\eta)}{\log(2)}\right). \tag{2.2.12}$$

It appears on Figure 2.4a that for $\rho = -1$, property (2.2.8) holds while first and second derivatives do not vanish at the boundaries of $[0,1]$. When $\rho = -2$ (Figure 2.4b) both properties (2.2.8) and (2.2.9) are satisfied while the second derivative converges to a finite value in the neighborhood of 0, see (2.2.11), and diverges in the neighborhood of 1, see (2.2.10). Finally, $\rho = -3$ (Figure 2.4c) corresponds to the same situation, except that the second derivative also converges in the neighborhood of 1, see (2.2.12).

Let $I \subset \mathbb{R}$. Let us recall that a function $f : I \mapsto \mathbb{R}$ is Hölder continous with exponent $\alpha \in (0,1]$ if the following quantity is finite

$$[f]_\alpha := \sup_{x\neq y\in I} \frac{|f(x)-f(y)|}{|x-y|^\alpha}.$$

This property is denoted for short by $f \in H^\alpha(I)$. The case $\alpha = 1$ corresponds to Lipschitz functions. We shall also note $C^m(I)$ the set of m-th continuously differentiable functions on I, $m \in \mathbb{N}$. Finally, for all $\alpha \in (0,1]$ and $m \in \mathbb{N}$, we denote by $\mathcal{C}^{m,\alpha}(I)$ the Hölder space which consists of all functions $f \in C^m(I)$ such that $\partial^m f \in H^\alpha(I)$. In particular, $\mathcal{C}^{m+1}(I) \subseteq \mathcal{C}^{m,1}(I)$. Using these notations, and focusing on the case where $\rho < -1$, the regularity properties of f^{CTIF} provided by Proposition 2.2.2 can be simplified as:

Corollary 2.2.3. *Assume* $(\mathbf{H_1})$*,* $(\mathbf{H_2})$ *and* $(\mathbf{H_3})$ *hold.*

(i) If $-2 \le \rho < -1$ *then* $f^{\mathrm{CTIF}} \in \mathcal{C}^{1,\alpha}([0,1))$ *for all* $\alpha \in (0, -1-\rho)$*.*

(ii) If $\rho < -2$ *then* $f^{\mathrm{CTIF}} \in C^2([0,1])$*.*

It is thus clear that, the smaller ρ is, the more regular f^{CTIF} is, and therefore higher regularities could be obtained at the price of further restrictions on ρ. We are now in a position to investigate how a NN can approximate such a function.

2.2.2 Approximation error

Lemma 2.B.7 in Appendix 2.B provides the minimum number of ReLU functions to approximate a $\mathcal{C}^{1,\alpha}$ function with a given precision ϵ. Combining this result with Corollary 2.2.3 yields the uniform approximation error of f^{CTIF} by a NN depending on the number of ReLU functions:

Theorem 2.2.1. *Assume* $(\mathbf{H_1})$, $(\mathbf{H_2})$ *and* $(\mathbf{H_3})$ *hold. Let* σ *be a ReLU function. For all* $J \geq 6$, *there exist* $(a_j, w_j, b_j) \in \mathbb{R}^3$, $j = 1, \dots, J$ *such that:*

$$\sup_{u\in[0,1]} \left| f^{\text{CTIF}}(u) - \sum_{j=1}^{J} a_j \sigma\left(w_j u + b_j\right) \right| \leq \frac{[\partial_t f^{\text{CTIF}}]_\alpha}{4} \left\lceil \frac{J-3}{3} \right\rceil^{-\alpha-1} = \mathcal{O}\left(J^{-\alpha-1}\right),$$

where

1. $\alpha \in (0, -1-\rho)$ *if* $-2 \leq \rho < -1$,
2. $\alpha = 1$ *if* $\rho < -2$.

Note that, for $\alpha = 1$, the above rate cannot be improved in general, owing to [189, Theorem 6]. Moreover, the previous approximation result can be interpreted in terms of Wasserstein-1 distance between the true data distribution and the simulated one. Indeed, in the univariate case, the Wasserstein-1 distance can be simplified as

$$W_1(q_Y, \widetilde{q}_Y) = \int_0^1 |q_Y(u) - \widetilde{q}_Y(u)| \, \mathrm{d}u,$$

where $u \mapsto \widetilde{q}_Y(u) := H_u^{-1}(G(u))$, with $H_u^{-1}(\cdot)$ defined in (2.1.5), is the EV-GAN approximation of the unknown quantile function $u \mapsto q_Y(u)$.

Corollary 2.2.4. *Assume conditions of Theorem 2.2.1 hold with* $\gamma < 1$ *and* $\rho < -1$. *Then, the Wasserstein-1 distance can be controlled as* $W_1(q_Y, \widetilde{q}_Y) = \mathcal{O}\left(J^{-\alpha-1}\right)$.

Note that $\gamma < 1$ is a necessary condition for the Wasserstein-1 distance to exist. In view of (2.2.6) and (2.2.7), letting

$$e_1(u) = g(u), \quad e_2(u) = h(u) \quad \text{and} \quad e_{k+3}(u) = g(u)\Phi_k(u) \text{ for } k = 0, \dots, 3 \tag{2.2.13}$$

in (2.1.4), the above approximation bounds on f^{CTIF} can be translated in terms of approximation bounds on f^{TIF} using the "enriched" NN. Note that the approximation space of TIF functions can be done for all components of a d-dimensional random variable, by following the principle (2.1.6), with a latent dimension $d' \geq d$. We obtain the final approximation result whose proof is now an easy combination of Corollary 2.2.3 and Theorem 2.2.1.

Corollary 2.2.5. *Let* σ *be a ReLU function. Let* $X = (X^{(1)}, \dots, X^{(d)})^\top$ *be a* d*-dimensional vector, with each component* $X^{(m)}$ *fulfilling* $(\mathbf{H_1})$, $(\mathbf{H_2})$ *and* $(\mathbf{H_3})$ *with parameters* $(\gamma^{(m)}, \rho^{(m)})$. *Let* $\mathcal{G}_J^{d',d}$ *be the approximation space of TIF functions made of* $J \geq 6$ *neurons:*

$$\mathcal{G}_J^{d',d} := \Bigg\{ G : z \in [0,1]^{d'} \mapsto G(z) = (G^{(1)}(z), \dots, G^{(d)}(z))^\top,$$
$$G^{(m)}(z) = \sum_{j=1}^{J} a_j^{(m)} \sigma\left(\sum_{i=1}^{d'} w_j^{(i)} z^{(i)} + b_j \right) + \sum_{k=1}^{6} \kappa_k^{(m)} e_k\left(z^{(m)}\right),$$
$$a_j^{(m)}, w_j^{(i)}, b_j, \kappa_k^{(m)} \in \mathbb{R} \Bigg\}.$$

Then,

$$\inf_{G\in\mathcal{G}_J^{d',d}} \sup_{m=1,\dots,d} \sup_{z\in[0,1]^{d'}} \left| f^{\mathrm{TIF},(m)}(z^{(m)}) - G^{(m)}(z) \right| = \mathcal{O}\left(J^{-\alpha-1}\right),$$

where

(i) $\alpha \in (0, -1 - \max_{m=1,\dots,d} \rho^{(m)})$ *if* $-2 \le \rho^{(m)} < -1$ *for some* $m = 1, \dots, d$,

(ii) $\alpha = 1$ *if* $\rho^{(m)} < -2$ *for all* $m = 1, \dots, d$.

Here, we have defined

$$f^{\mathrm{TIF},(m)}(z^{(m)}) = -\frac{\log(q_{X^{(m)}}(1-(1-z^{(m)})\eta^{(m)}))}{\log\left(1-\left(z^{(m)}\right)^2\right) - \log 2}$$

as an natural extension of (2.2.3). For optimal parameters $a_j^{(m)}, w_j^{(i)}, b_j, \kappa_k^{(m)}$, the generative model for X is then

$$\tilde{X} = \left(H^{-1}_{Z^{(m)}}\left(G^{(m)}(Z)\right) : m = 1, \dots, d \right) \qquad \text{with} \quad Z \overset{d}{=} \mathcal{U}([0,1]^{d'}). \tag{2.2.14}$$

In the above, one could restrict $G^{(m)}(z)$ to depend only on the m-th coordinate of z: it would not affect the potential quality of approximation of the m-th marginal of X but it would lead to a generative model with independent components which would be too restrictive. Mixing all latent components of z in $G^{(m)}(z)$ allows for generating dependence in the tails, while ensuring good fit of the marginals, as it will be checked in the subsequent experiments.

Observe that the worst second-order parameter $\rho^{(m)}$, *i.e.* the closest to -1, tunes the global accuracy of the EV-GAN through the convergence order $\alpha + 1$. Obtaining a similar result on the d-dimensional Wasserstein-1 distance is beyond the scope of this paper.

One may wonder if deeper ReLU NNs would help in better approximating the generative model for X. From the theoretical point of view, the benefit is unclear, in particular in view of [189, Theorem 1] which states that a $\mathcal{C}^{1,1}$-function[1] can be approximated with error ϵ using a ReLU NN with depth $\mathcal{O}(\log(1/\epsilon))$ and number of weights $\mathcal{O}(\epsilon^{-1/2}\log(1/\epsilon))$. Up to the log factor, this is similar to the above result (Theorem 2.2.1) by setting $\epsilon = J^{-\alpha-1}$. From the numerical point of view, identifying in which circumstances a deep ReLU NN could be useful is part of our further investigations. Let us highlight that Lemma 2.B.7, and thus the whole analysis, can be adapted to any other non-polynomial activation function in view of the Universal approximation theorem [152].

2.3 Implementation

2.3.1 Experimental design

The neural network training is done by alternating generator and discriminator steps. The ranges of hyperparameters that are explored in order to find the best model for each data configuration are reported in Table 2.5. Note that, in order to respect the architecture (2.1.6), the generator is restricted to be a one hidden layer NN. Additionally, we use the optimizer Adam [114] with default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for all tests performed during $1,000$ iterations. No additional normalization techniques are used. Every 5 iterations, two metrics (see Section 2.3.2 below) are computed and, for each metric, the NN parameters associated with the best results among the 200 checkpoints are selected.

[1] His result does not apply to our possible case $\alpha \in (0, 1]$.

2.3.2 Performance assessment

Recall that from (2.2.1), in the heavy-tail model, for all $j \in \{1,\dots,d\}$, $\log q_{X^{(j)}}(u)$ is approximately proportional to $\log(1/(1-u))$ when u is close to 1, with the tail-index γ as proportionality factor. It is therefore common practice to check the heavy-tail assumption on each margin $j \in \{1,\dots,d\}$ by drawing a log quantile-quantile plot, namely the points $(\log((n+1)/i), \log X^{(j)}_{n-i+1,n})$, for $i \in \{1,\dots,\lceil(1-\xi)n\rceil\}$, where $\xi \in [0,1)$ is a given probability level. The performance of a generator can then be visually assessed by comparing the pairs $(\log((n+1)/i), \log X^{(j)}_{n-i+1,n})$ and $(\log((n+1)/i), \log \tilde{X}^{(j)}_{n-i+1,n})$. Here, and in the sequel, $\{\tilde{X}_1,\dots,\tilde{X}_n\}$ denotes the outputs generated either by the EV-GAN model (2.2.14) or by the classic GAN. To further quantify the fit on the tails of the marginal distributions, we define the Mean squared logarithmic error (MSLE) as the squared distance between the logarithm of the original and generated data:

$$\mathrm{MSLE}(\xi) = \frac{1}{d\lceil(1-\xi)n\rceil}\sum_{j=1}^{d}\sum_{i=1}^{\lceil(1-\xi)n\rceil}\left(\log(X^{(j)}_{n-i+1,n}) - \log(\tilde{X}^{(j)}_{n-i+1,n})\right)^2,$$

with $\xi \in \{0.90, 0.95, 0.99\}$. Thus, a 100% relative error on the marginals corresponds to $\mathrm{MSLE}(\xi) = (\log(2))^2 \simeq 0.48$. Considering the dependence structure, one may also graphically compare the estimated Kendall's dependence functions K (or equivalently the $t \mapsto \lambda(t) := t - K(t)$ functions) on the n observations associated with the original sample and the generated one. From the quantitative point of view, the fit of the dependence structure is assessed by the 1-Wasserstein distance between these two Kendall's dependence functions which indeed are cumulative distribution functions. The distance can be computed as a L^1 norm referred to as the Absolute Kendall error (AKE) in the sequel:

$$\mathrm{AKE} = \frac{1}{n}\sum_{i=1}^{n}\left|Z_{i,n} - \tilde{Z}_{i,n}\right|,$$

where $Z_{1,n} \le \cdots \le Z_{n,n}$ (resp. $\tilde{Z}_{1,n} \le \cdots \le \tilde{Z}_{n,n}$) are the order statistics associated with $\{Z_1,\dots,Z_n\}$ (resp. $\{\tilde{Z}_1,\dots,\tilde{Z}_n\}$) and the $\tilde{Z}_i$ are computed similarly to (2.A.1) in Appendix 2.A on the generated sample. We shall also compare Kendall's tau estimated on the original sample $\hat{\tau}_n$, on the generated sample $\tilde{\tau}_n$ and the theoretical value $\tau_{C^G_\mu}$.

2.3.3 Computational aspects

The numerical experiments presented in the next two sections have been conducted on the Cholesky computing cluster from Ecole Polytechnique `http://meso-ipp.gitlab.labos.polytechnique.fr/user_doc/`. It is composed by 2 nodes, where each one includes 2 CPU Intel Xeon Gold 6230 @ 2.1GHz, 20 cores and 4 Nvidia Tesla v100 graphics card. All the code was implemented in Python 3.8.2 and using the library PyTorch 1.7.1 for the GANs' training.

2.4 Validation on simulated data

The data simulation is based on the use of copulas, which allow to model separately the dependence structure and the margins, see Appendix 2.A for a short overview. We focus on the Gumbel copula, denoted by C^G_μ which has been proved to be the only max-stable Archimedean copula [86]. The associated generating function is $\psi^G_\mu(t) = \exp(-t^{1/\mu})$ defined for all $\mu \ge 1$ and $t \ge 0$. It is easily seen that Kendall's dependence function is given by $K_{C^G_\mu}(t) = t - t\log(t)/\mu$ for all $t \in (0,1]$ and Kendall's tau is $\tau_{C^G_\mu} = 1 - 1/\mu$. These two above quantities respectively provide a local and global characterization of the dependence structure induced by the copula.

Besides, $C_1^G = \Pi$ the independence copula, and $C_\mu^G \to M$, the comotonic copula, as $\mu \to \infty$. In the following Section 2.4.1 we restrict ourselves to the dimension $d = 2$, while, in Section 2.4.2, we provide illustrations in higher dimensions.

2.4.1 Bivariate case

Three values of the dependence parameter are investigated: $\mu \in \{1.1, 2, 10\}$ leading to $\tau_{C_\mu^G} \in \{0.1, 0.5, 0.9\}$. The two margins are chosen to be Burr distributed, with common tail-index $\gamma := \gamma_1 = \gamma_2 \in \{0.1, 0.5, 0.9\}$ and second-order parameters $(\rho_1, \rho_2) \in \{(-1, -2), (-1, -3), (-2, -3)\}$, see Table 2.4 for the parametrization of the Burr distribution. Finally, $n = 10,000$ i.i.d data $\{X_1, \ldots, X_n\}$ are simulated from the resulting bivariate model for the above $3 \times 3 \times 3 = 27$ combinations of parameters. Results are reported on Table 2.1 in terms of MSLE(0.99), AKE and Kendall's tau.

When the tail-index γ increases, the tails of the marginal distributions of the simulated get heavier and the MSLE criteria of GAN and EV-GAN methods increase for all considered values of (ρ_1, ρ_2, μ) with a clear soaring when $\gamma = 1$. In this latter case, the expectation of the simulated distribution does not exist. However, from this marginal point of view, EV-GAN outperforms GAN in terms of MSLE for all considered configurations of $(\gamma, \rho_1, \rho_2, \mu)$. This conclusion remains true from the dependence point of view: EV-GAN outperforms GAN in terms of AKE for all the considered configurations of (ρ_1, ρ_2, μ) when $\gamma \in \{0.5, 0.9\}$. This phenomenon is illustrated in Figure 2.5 in the case where $\gamma = 0.9$, $\rho_1 = -1$, $\rho_2 = -3$ and $\mu = 10$. The log quantile-quantile plots associated with both margins are displayed on the top panel. It is easily seen that GAN method is not able to generate data in the distribution tail since the tail heaviness is strongly underestimated. At the opposite, EV-GAN method yields realistic data generation in both marginal tail distributions. Note that, in this case, the dependence structure is well captured by both NNs, see the estimated $\lambda(\cdot)$ functions on the bottom panel.

Finally, it appears on Table 2.1 that Kendall's tau is not a sufficient summary of the dependence structure: All estimated Kendall's tau are close to the theoretical ones even though the AKE is large. This criterion is thus dropped in the real data analysis hereafter since it might yield misleading conclusions.

2.4.2 Multivariate case

The ability of EV-GAN to properly scale in high dimension is now investigated. Using the `R` package `copulas` [117], $n = 10,000$ samples are simulated from a d-variate Gumbel copula for increasing dimensions $d \in \{4, 8, 16, 32, 64, 128, 256, 512, 1024\}$, with a unique dependence parameter $\mu = 2$ and where all margins are Burr distributed with parameters $\gamma = 0.5$ and $\rho = -1$. MSLE results at level $\xi \in \{0.90, 0.95, 0.99\}$ are reported in Table 2.2 for both GAN and EV-GAN methods. Here again, EV-GAN clearly outperforms GAN for all dimensions and levels considered. Indeed, EV-GAN method yields realistic margins up to dimension 512 for high levels of quantiles $\xi \in \{0.90, 0.95\}$. In the case of higher levels ($\xi \in \{0.99\}$) the dimension is limited to 128. In contrast, the classic GAN model is limited more or less to dimension 8 for all levels. Figure 2.6 illustrates the dependence associated with samples in dimension $d \in \{4, 8, 16, 32, 64, 128\}$. First, remark that $\lambda(\cdot)$ associated with the original data tends toward the independence function $t \mapsto t - 1$ as d increases, accordingly to [76, Section 3.3]. Second, it appears that EV-GAN manages to reproduce very well the dependence structure of the original data up to $d = 16$, but tends faster to the independence between the margins for higher dimensions. Removing this trend is part of our future work, see Section 2.6.

MSLE(0.99)

γ	(ρ_1, ρ_2) \ μ	1.1		2		10	
0.1	$(-1,-2)$	0.895	**0.116**	0.545	**0.097**	0.232	**0.037**
	$(-1,-3)$	0.923	**0.103**	0.732	**0.082**	0.553	**0.143**
	$(-2,-3)$	0.677	**0.190**	0.836	**0.083**	0.700	**0.174**
0.5	$(-1,-2)$	3.576	**1.058**	10.673	**1.006**	2.567	**1.321**
	$(-1,-3)$	1.943	**0.958**	6.913	**1.569**	3.812	**3.252**
	$(-2,-3)$	10.809	**1.707**	10.157	**1.201**	1.306	**1.195**
0.9	$(-1,-2)$	47.742	**4.966**	-	**6.473**	-	**8.651**
	$(-1,-3)$	44.949	**3.129**	-	**4.573**	45.900	**3.205**
	$(-2,-3)$	-	**3.860**	36.304	**6.390**	44.814	**5.922**

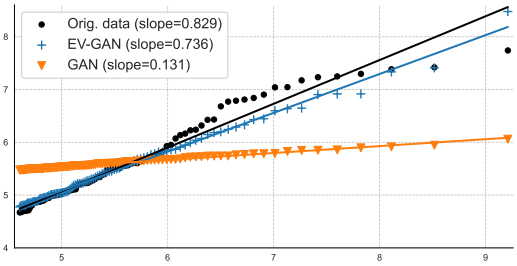
AKE

γ	(ρ_1, ρ_2) \ μ	1.1		2		10	
0.1	$(-1,-2)$	3.122	**2.865**	**7.385**	8.322	**2.807**	2.855
	$(-1,-3)$	3.102	**2.293**	**5.671**	6.958	**1.585**	2.415
	$(-2,-3)$	**2.519**	3.244	**4.596**	7.125	**1.823**	2.340
0.5	$(-1,-2)$	3.052	**2.234**	4.857	**1.772**	2.265	**2.015**
	$(-1,-3)$	6.261	**2.342**	4.538	**1.665**	2.616	**1.301**
	$(-2,-3)$	5.772	**2.134**	12.277	**1.408**	4.245	**1.531**
0.9	$(-1,-2)$	2.555	**2.103**	-	**1.990**	-	**1.932**
	$(-1,-3)$	3.788	**1.861**	-	**1.700**	1.623	**1.429**
	$(-2,-3)$	-	**1.696**	5.632	**1.788**	1.991	**1.181**

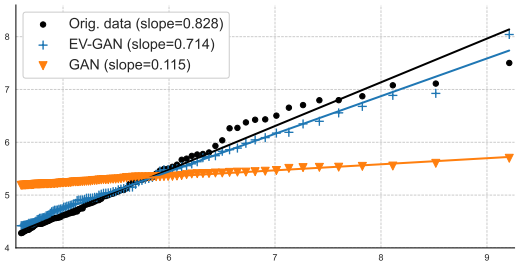
Kendall's tau

γ	(ρ_1, ρ_2) \ μ ($\tau_{C_\mu^\theta}$)	1.1 (0.1)		2 (0.5)		10 (0.9)	
0.1	$(-1,-2)$	0.092	**0.091**	**0.514**	0.531	0.905	**0.895**
	$(-1,-3)$	**0.093**	0.083	0.477	**0.500**	**0.900**	0.905
	$(-2,-3)$	**0.086**	0.083	0.511	**0.480**	**0.899**	0.903
0.5	$(-1,-2)$	**0.090**	0.088	0.493	**0.500**	0.903	**0.900**
	$(-1,-3)$	0.106	**0.096**	0.506	**0.502**	0.901	**0.900**
	$(-2,-3)$	0.093	**0.087**	0.473	**0.502**	0.885	**0.898**
0.9	$(-1,-2)$	0.088	**0.090**	-	**0.503**	-	**0.901**
	$(-1,-3)$	**0.091**	0.088	-	**0.499**	**0.899**	0.897
	$(-2,-3)$	-	**0.089**	0.487	**0.498**	0.900	**0.900**

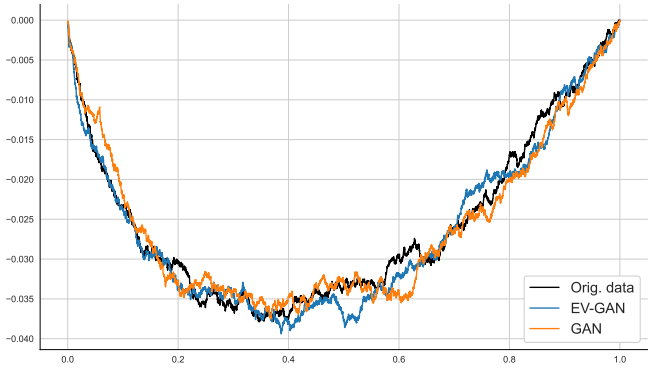
Table 2.1: Comparison between the best GAN (left column) and EV-GAN (right column) results on simulated data for the 27 combinations of parameters using two model selection criteria. Top: MSLE criterion at level $\xi = 0.99$, MSLE(ξ) ≥ 0.48 are not reported, all results are scaled by 10^2. Center: AKE criterion, the results are scaled by 10^3. Bottom: Kendall's tau (using the same models as the ones based on the AKE criterion). Best results are emphasized in bold.



(a) First margin ($\gamma = 0.9, \rho_1 = -1$)



(b) Second margin ($\gamma = 0.9, \rho_2 = -3$)



(c) Estimated $\lambda(\cdot)$ functions

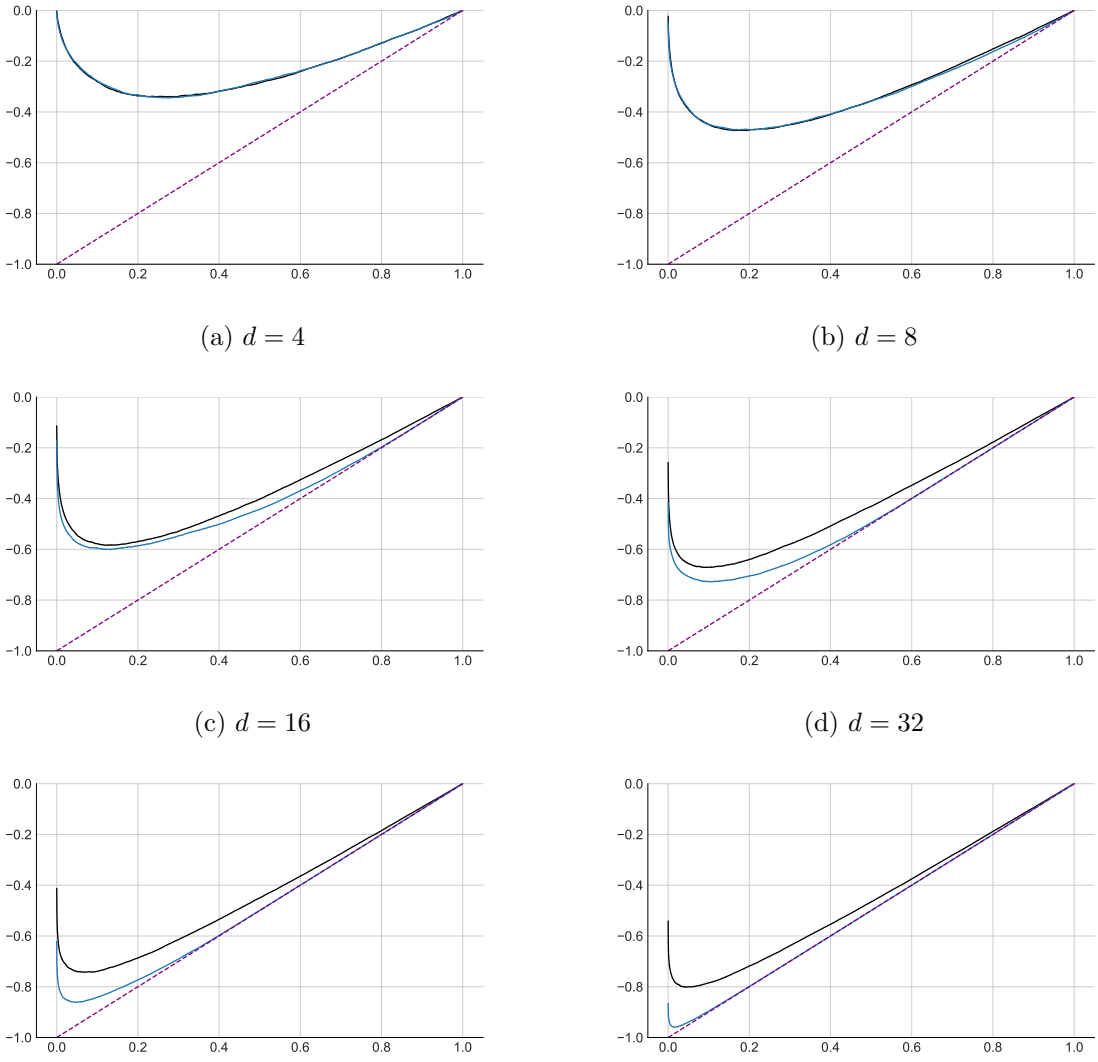
Figure 2.5: Top: log quantile-quantile plots on each margin $\log((n+1)/i) \mapsto \log X^{(j)}_{n-i+1,n}$, for $i \in \{1, \ldots, \lceil (1-\xi)n \rceil\}$ and $j \in \{1, 2\}$ on simulated data at probability level $\xi = 0.99$. The estimated regression lines are superimposed to each scatter plot. The associated slope is an estimation of the tail-index γ. Bottom: estimated $t \in [0, 1] \mapsto \lambda(t)$ functions. Black: original simulated data ($\gamma = 0.9$, $\rho_1 = -1$, $\rho_2 = -3$ and $\mu = 10$), blue: data generated with EV-GAN model, orange: data generated with classic GAN model.

2.5 Illustration on real financial data

Our approach is tested on closing prices of daily financial stock market indices taken from `https://stooq.com/db/h/` on the October 1st, 2020. This database includes 61 world indices from their first day of quotation. Here, we selected six indices: NKX (Nikkei, Japan), KOSPI (Korea), HSI (Hong-Kong), CAC (France), AMX (Amsterdam Exchange, Netherlands), Nasdaq (USA) from three market zones: Asia, Europe, USA.

As a pre-processing step, the daily log-returns are computed for each ticker index. In case of missing data at a given business day, the next available day is removed from the dataset. Also, since we are interested in the modeling of synchronous indices, we kept only the data available at the same date for all selected tickers. Finally, positive returns were discarded since we focus on the generation of losses.

Figure 2.7 proposes a graphical summary of the tail and dependence properties associated with this dataset. First, the log quantile-quantile plots computed on all indices at level $\xi = 0.95$ are approximately linear which provides a graphical evidence of the tail heaviness of all six marginal distributions, with, for all indices, estimated slopes pointing towards a tail-index $\hat{\gamma} \simeq 0.3$ and an estimated second order parameter $\hat{\rho} \simeq -0.7$ using the estimator implemented in the `R` package `evt0` [139]. Second, the $\lambda(\cdot)$ associated with all 15 pairs of indices are also displayed together with the two extreme cases $\lambda_\Pi(\cdot)$ and $\lambda_M(\cdot)$. The strongest dependence is found within



(a) $d = 4$ (b) $d = 8$

(c) $d = 16$ (d) $d = 32$

(e) $d = 64$ (f) $d = 128$

Figure 2.6: Estimated $t \in [0,1] \mapsto \lambda(t)$ functions on multivariate simulated data in dimension $d \in \{4, 8, 16, 32, 64, 128\}$. Black: original simulated data (Burr distribution, $\gamma = 0.5$, $\rho = -1$ and $\mu = 2$), blue: data generated with EV-GAN model, dashed purple: independence case $t \mapsto \lambda_\Pi(t) = t - 1$.

dimension d	MSLE(0.90)		MSLE(0.95)		MSLE(0.99)		AKE	
4	3.134	**0.946**	5.627	**1.726**	16.961	**5.990**	14.	**2.**
8	11.399	**3.391**	17.613	**5.262**	-	**12.682**	32.	**5.**
16	47.632	**11.294**	-	**12.288**	-	**9.515**	-	**28.**
32	47.466	**12.519**	43.610	**10.052**	-	**32.022**	-	**49.**
64	-	**13.455**	-	**12.746**	-	**15.278**	-	**53.**
128	-	**19.365**	-	**14.751**	-	**28.977**	-	**48.**
256	-	**18.073**	-	**30.824**	-	-	-	-
512	-	**19.390**	-	**18.863**	-	-	-	-
1024	-	-	-	-	-	-	-	-

Table 2.2: Performance comparison between the best GAN (left column) and EV-GAN (right column) results on simulated d-variate data with respect to four model selection criteria. First three columns: MSLE(ξ) criterion computed at levels $\xi \in \{0.90, 0.95, 0.99\}$, MSLE($\xi$) ≥ 0.48 are not reported, all results are scaled by 10^2. Last column: AKE criterion, results are scaled by 10^3. Best results are emphasized in bold.

the European market zone (pair AMX,CAC) while weakest dependencies are located between US and Asian market zones. Let us however note that the dependence between Asian, European and US markets may be under-estimated due to different time zones.

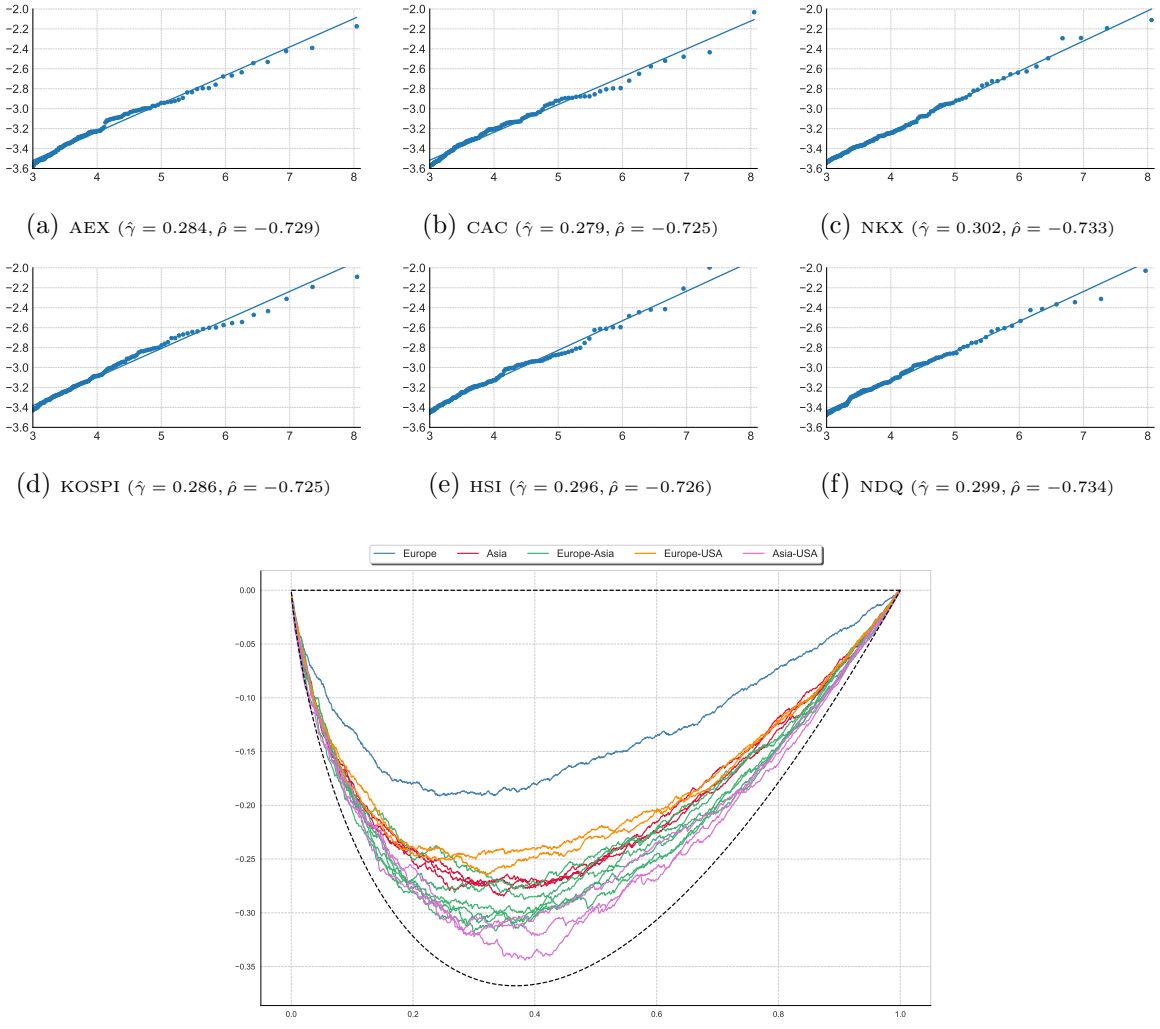
In the following, the performance of GAN and EV-GAN approaches are compared on four datasets of increasing dimensions: NKX ($d = 1$), Europe (AEX, CAC, $d = 2$), Asia (NKX, KOSPI, HSI, $d = 3$) and world (AEX, CAC, NKX, KOSPI, HSI, NDQ, $d = 6$). The training procedure described in Section 2.4 is adopted and results are reported in Table 2.3. EV-GAN outperforms GAN both on tail criteria MSLE(ξ), $\xi \in \{0.90, 0.95, 0.99\}$ and on dependence criterion AKE, even though the condition $\rho < -1$ may not be fulfilled on this dataset. These results are illustrated on Figure 2.8 where it appears that EV-GAN is able to generate financial indices with realistic marginal tail behaviors. Finally, Figure 2.9 provides a comparison of dependence results obtained either using the MSLE or the AKE criteria. Unsurprisingly, the latter yields better results. Here again, the results associated with EV-GAN are visually more satisfying than those of the classic GAN. Information on the selected hyperparameters is provided in Table 3.1.

ticker	NKX		Europe		Asia		World	
dimension d	1		2		3		6	
sample size n	3173		2504		1378		548	
MSLE(0.90)	0.473	**0.133**	3.860	**0.132**	2.353	**0.677**	3.306	**0.874**
MSLE(0.95)	0.742	**0.103**	4.925	**0.178**	1.481	**0.579**	4.467	**1.219**
MSLE(0.99)	1.381	**0.200**	2.792	**0.320**	1.023	**0.538**	5.000	**1.960**
AKE	–	–	16.807	**4.697**	9.760	**4.872**	24.781	**3.533**

Table 2.3: Performance comparison between the best GAN (left column) and the EV-GAN (right column) results on real data with respect to four model selection criteria: using the MSLE(ξ) criterion computed at levels $\xi \in \{0.90, 0.95, 0.99\}$ and the AKE criteria (results are multiplied by 10^3 for the sake of readability).

2.6 Conclusion

In this work, we have introduced a new generative method called EV-GAN dedicated to tail events. It relies on a new parametrization of GANs allowing to generate data coming from a

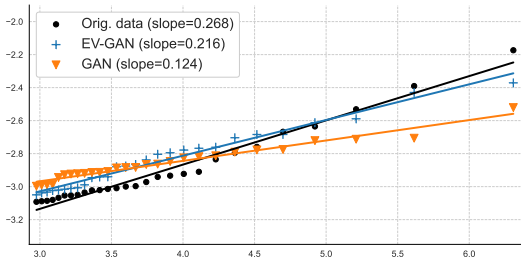


(a) AEX ($\hat{\gamma} = 0.284, \hat{\rho} = -0.729$) (b) CAC ($\hat{\gamma} = 0.279, \hat{\rho} = -0.725$) (c) NKX ($\hat{\gamma} = 0.302, \hat{\rho} = -0.733$)

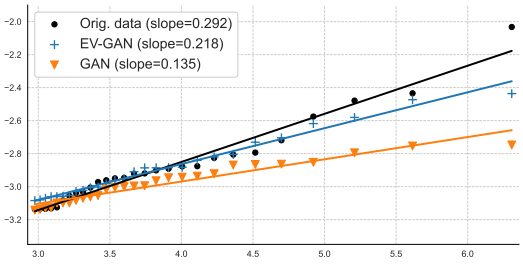
(d) KOSPI ($\hat{\gamma} = 0.286, \hat{\rho} = -0.725$) (e) HSI ($\hat{\gamma} = 0.296, \hat{\rho} = -0.726$) (f) NDQ ($\hat{\gamma} = 0.299, \hat{\rho} = -0.734$)

(g) Estimated $\lambda(\cdot)$ functions

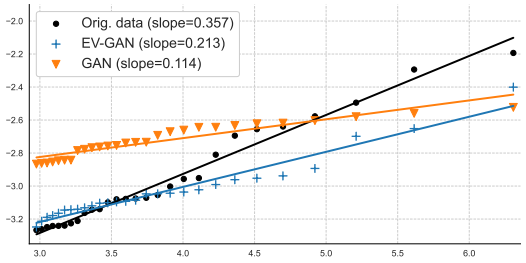
Figure 2.7: Top panels: Log quantile-quantile plots $\log((n+1)/i) \mapsto \log X^{(j)}_{n-i+1,n}$, for $i \in \{1, \ldots, \lceil (1-\xi)n \rceil\}$ on the selected financial indices $j \in \{1, \ldots, 6\}$ at probability level $\xi = 0.95$. The estimated regression line is superimposed to each scatter plot. The associated slope is an estimation of the tail-index. Bottom panel: Estimated $t \in [0,1] \mapsto \lambda(t)$ functions for all 15 pairs of indices. Functions $t \in [0,1] \mapsto \lambda_\Pi(t) = t \log t$ and $t \in [0,1] \mapsto \lambda_M(t) = 0$ respectively associated with independence and comotonic dependence in the bivariate case ($d = 2$) are depicted by black dashed lines.



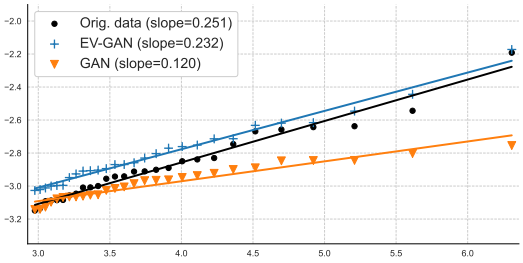
(a) AEX



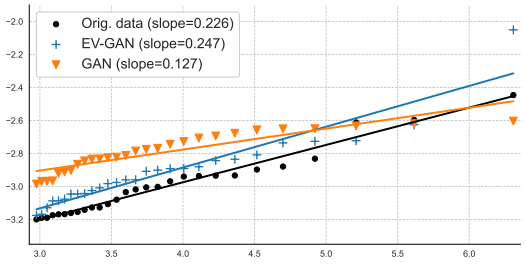
(b) CAC



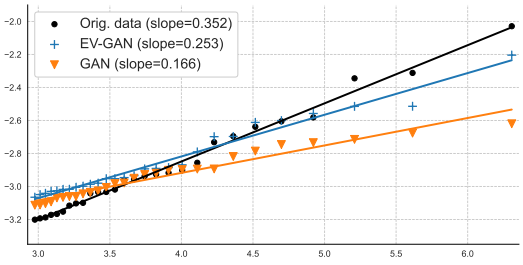
(c) NKX



(d) KOSPI

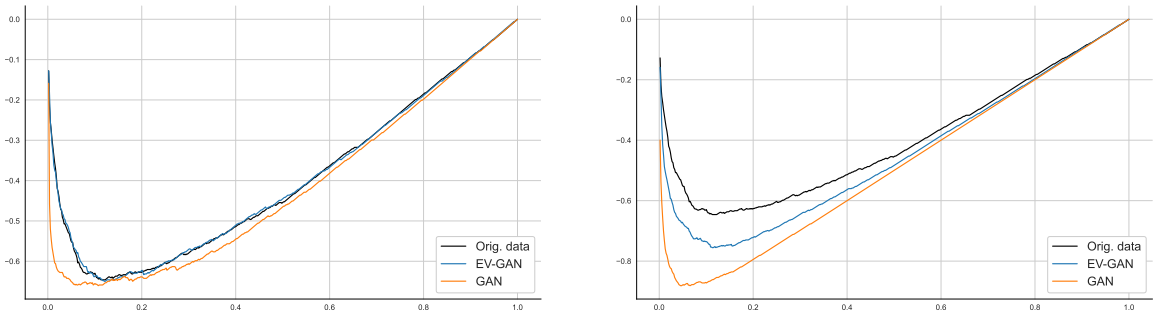


(e) HSI



(f) NDQ

Figure 2.8: Log quantile-quantile plots $\log((n+1)/i) \mapsto \log X^{(j)}_{n-i+1,n}$, for $i \in \{1, \ldots, \lceil (1-\xi)n \rceil\}$ and $j \in \{1, \ldots, 6\}$ associated with the world market zone at probability level $\xi = 0.95$.



(a) (b)

Figure 2.9: Estimated $t \in [0,1] \mapsto \lambda(t)$ functions associated with the World market zone ($d = 6$). Black: original real data, blue: data generated with EV-GAN model, orange: data generated with classic GAN model. (a) AKE criterion, (b) MSLE(0.95) criterion.

heavy-tailed distribution. From the theoretical point of view, the uniform convergence rate of the proposed transformed quantile function f^{TIF} by a one hidden-layer ReLU NN is established within an extreme-value framework. From the practical point of view, we have illustrated on real and simulated data that EV-GAN outperforms classic GAN both in terms of tail behavior of the marginal distributions and in terms of dependence structure.

To complete the current theoretical analysis which ensures accurate approximation of marginals using NN, our further work will be dedicated to investigate mathematically how dependence structure is preserved, leveraging multivariate extreme-value theory. The analysis goes far beyond this work since it is known that dependence structure in the tails can be quite different from one case to another [42].
Finally, we shall investigate the behavior of the proposed EV-GAN corrections in other GAN architectures, using different distances and alternative criteria to MSLE and AKE.

Distribution (parameters)	Density function	γ	ρ
Pareto ($\alpha > 0$)	$\alpha t^{-\alpha-1}$ $(t > 1)$	$1/\alpha$	$-\infty$
Burr ($\alpha, \beta > 0$)	$\alpha\beta t^{\alpha-1}(1+t^{\alpha})^{-\beta-1}$ $(t > 0)$	$1/(\alpha\beta)$	$-1/\beta$
Fréchet ($\alpha > 0$)	$\alpha t^{-\alpha-1}\exp(-t^{-\alpha})$ $(t > 0)$	$1/\alpha$	-1
Fisher ($\nu_1, \nu_2 > 0$)	$\frac{(\nu_1/\nu_2)^{\nu_1/2}}{B(\nu_1/2,\nu_2/2)} t^{\nu_1/2-1}(1+\nu_1 t/\nu_2)^{-(\nu_1+\nu_2)/2}$ $(t > 0)$	$2/\nu_2$	$-2/\nu_2$
Inverse-Gamma ($\alpha, \beta > 0$)	$\frac{\beta^{\alpha}}{\Gamma(\alpha)} t^{-\alpha-1}\exp(-\beta/t)$ $(t > 0)$	$1/\alpha$	$-1/\alpha$
Cauchy ($\sigma > 0$)	$\frac{\sigma}{\pi(\sigma^2+t^2)}$	1	-2
Student ($\nu > 0$)	$\frac{1}{\sqrt{\nu\pi}} \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \left(1+\frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$	$1/\nu$	$-2/\nu$

Table 2.4: A list of heavy-tailed distributions with the associated values of γ and ρ.

Appendix 2.A collects some statistical tools based on copulas used in experiments (Section 2.4 and Section 2.5). Appendix 2.B provides auxiliary results used in Appendix 2.C to prove the

main results of Section 2.2.

2.A Copulas

Let us consider a $d-$ variate stribution function F_X with continuous margins denoted by $F_X^{(j)}$, $j \in \{1, \dots, d\}$. From Sklar's Theorem [168], there exists a unique function C such that

$$F_X\left(x^{(1)}, \dots, x^{(d)}\right) = C\left(F_X^{(1)}(x^{(1)}), \dots, F_X^{(d)}(x^{(d)})\right),$$

with $\left(x^{(1)}, \dots, x^{(d)}\right) \in \mathbb{R}^d$. The function C is called the copula of F_X. Introducing the uniform random variables $U^{(j)} = F^{(j)}(X^{(j)})$ for all $j \in \{1, \dots, d\}$, the copula C is the $d-$ dimensional distribution function of the random vector $\left(U^{(1)}, \dots, U^{(d)}\right)$ with uniform margins on $[0, 1]$. Copulas are a flexible tool to impose a given dependence structure on the marginal distributions of interest, see [145] for a detailed account on copulas. The independence between margins corresponds to the product copula $\Pi(u^{(1)}, \dots, u^{(d)}) = u^{(1)} \dots u^{(d)}$ while comotonic dependence corresponds to the Fréchet copula $M(u^{(1)}, \dots, u^{(d)}) = \min(u^{(1)}, \dots, u^{(d)})$.

Archimedean copulas. An Archimedean copula C_μ is defined for all $(u^{(1)}, \dots, u^{(d)}) \in [0, 1]^d$ by

$$C_\mu\left(u^{(1)}, \dots, u^{(d)}\right) = \psi_\mu\left(\psi_\mu^{-1}(u^{(1)}) + \dots + \psi_\mu^{-1}(u^{(d)})\right),$$

where $\psi_\mu : [0, \infty) \to [0, 1]$ is a parametric function which has to verify certain properties listed for instance in [140].

Kendall's dependence function. Kendall's dependence function [87] characterizes the dependence structure associated with a copula C and is the univariate cumulative distribution function defined by $K_C(t) = \mathbb{P}\left(C\left(U^{(1)}, \dots, U^{(d)}\right) \leq t\right)$ for all $t \in [0, 1]$. In the case of an Archimedean copula C_μ, it can be derived as [76]:

$$K_{C_\mu}(t) = t + \sum_{j=1}^{d-1} \frac{(-\psi_\mu^{-1}(t))^j}{j!} \psi_\mu^{(j)}(\psi_\mu^{-1}(t)),$$

and we shall thus consider $\lambda_{C_\mu}(t) := t - K_{C_\mu}(t)$. It is then easily seen that $\lambda_M(t) = 0$ and

$$\lambda_\Pi(t) = t \sum_{j=1}^{d-1} \frac{(-\log(t))^j}{j!}$$

for all $t \in (0, 1]$.

Kendall's tau (bivariate case). Kendall's tau [113] is a measure of dependence between two random variables. Let us then assume $d = 2$ and let X and $\tilde{X}$ be two bivariate random vectors from F_X. Kendall's tau is defined as the probability of concordance minus the probability of discordance of $X = (X^{(1)}, X^{(2)})$ and $\tilde{X} = (\tilde{X}^{(1)}, \tilde{X}^{(2)})$. It can be shown [145, Theorem 5.1.3] that this quantity only depends on the copula C of F_X and is given by

$$\tau_C = 4\mathbb{E}C(U^{(1)}, U^{(2)}) - 1 = 4 \int_0^1 \int_0^1 C(u, v)\, \mathrm{d}C(u, v) - 1,$$

with $\tau_M = 1$ and $\tau_\Pi = 0$ as special cases. In case of an Archimedean copula C_μ, Kendall's tau and Kendall's dependence functions are linked [85]:

$$\tau_{C_\mu} = 1 + 4 \int_0^1 \lambda_{C_\mu}(v)\, \mathrm{d}v,$$

meaning that τ_{C_μ} can be interpreted as a summary of the dependence information encoded in $\lambda_{C_\mu}(\cdot)$.

Sampling (bivariate case). Sampling a random pair (U, V) from a bivariate copula C can be achieved by first simulating independently $(U, W) \sim \mathcal{U}([0,1]^2)$ and then letting $V = C_u^{-1}(W)$ where C_u is the conditional copula defined by

$$C_u(v) = \mathbb{P}(V \leq v | U = u) = \partial_u C(u, v).$$

In the case of bivariate Archimedean copulas, the conditional copula and its inverse are given by [18]:

$$C_{\mu,u}(v) = \frac{\partial_u \left(\psi_\mu^{-1}\right)(u)}{\partial_u \left(\psi_\mu^{-1}\right)(C(u,v))},$$
$$C_{\mu,u}^{-1}(y) = \psi_\mu \left((\partial_u \psi_\mu)^{-1} \left(\frac{y}{\partial_u \left(\psi_\mu^{-1}\right)(u)} \right) - \psi_\mu^{-1}(u) \right).$$

We also refer to [188] and [109] for alternative methods based on Kendall's dependence function and Laplace transform respectively.

Inference. The estimation of Kendall's dependence function is based on the pseudo-observations $\{Z_1, \ldots, Z_n\}$ from the cumulative distribution function K and computed as

$$Z_i = \frac{1}{n-1} \sum_{j \neq i}^{n} \mathbb{1}\left\{ X_j^{(1)} < X_i^{(1)}, \ldots, X_j^{(d)} < X_i^{(d)} \right\}, \tag{2.A.1}$$

for all $i \in \{1, \ldots, n\}$, see [87]. The estimator of K is computed using the associated empirical cumulative distribution function:

$$\hat{K}_n(t) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left\{ Z_i \leq t \right\},$$

and we set $\hat{\lambda}_n(t) = t - \hat{K}_n(t)$, for all $t \in [0,1]$. Similarly, Kendall's tau is estimated by

$$\hat{\tau}_n = \frac{4}{n} \sum_{i=1}^{n} Z_i - 1.$$

2.B Auxiliary results

We begin with a constructive proof of a particular case of Kuratowski Theorem [19, Chapter 7]-[178, p.8].

Lemma 2.B.1. *Let X be a random variable on $\mathbb{R}^d$. There exists a measurable function $G : (0,1) \to \mathbb{R}^d$ such that $X \overset{d}{=} G(U)$ with $U \sim \mathcal{U}([0,1])$.*

Proof. Let $Q : \mathbb{R}^d \to (0,1)^d$ be the component-wise logistic bijective function defined as $Q^{(m)}(x) = 1/(1 + \exp(-x^{(m)}))$ for all $m \in \{1, \ldots, d\}$. Let us also consider a continuous surjection $S : [0,1] \to [0,1]^d$ associated with a Space filling curve (like Peano or Hilbert curves, see [163]). Define the inverse function $S^{-1}(x) := \inf\{t \in [0,1] : S(t) = x\}$, for any $x \in [0,1]^d$: it is measurable and is such that $S(S^{-1}(x)) = x$ since S is continuous. Then, $k := S^{-1} \circ Q$ is a measurable function from $\mathbb{R}^d$ to $(0,1)$, $k^{-1} = Q^{-1} \circ S$ is measurable too and satisfies $k^{-1}(k(x)) = x$ for any x. Additionally, let $Y := k(X)$ be a random variable on $(0,1)$ with cumulative distribution function F_Y so that $F_Y^{-1}(U) \overset{d}{=} Y$, set $G(u) = k^{-1}\left(F_Y^{-1}(u)\right)$, for all $u \in (0,1)$. Then, for any bounded test function $\varphi : (0,1) \to \mathbb{R}^d$ we get

$$\mathbb{E}\varphi(G(U)) = \mathbb{E}\varphi\left(k^{-1}\left(F_Y^{-1}(U)\right)\right) = \mathbb{E}\varphi\left(k^{-1}(Y)\right) = \mathbb{E}\varphi(X),$$

which proves that $X \overset{d}{=} G(U)$. □

The following three lemmas provide asymptotic expansions that will reveal useful to establish the behavior of the TIF as well as its derivatives in the neighborhood of $u = 0$ and $u = 1$.

Lemma 2.B.2.

(i) The following asymptotic expansions hold, as $u \to 1$:

$$\frac{1}{\log\left(\frac{1-u^2}{2}\right)} = \frac{1}{\log(1-u)} + \frac{1-u}{2\left(\log(1-u)\right)^2} + \mathcal{O}\left(\frac{(1-u)^2}{\left(\log(1-u)\right)^2}\right), \tag{2.B.1}$$

$$\partial_u\left[\frac{1}{\log\left(\frac{1-u^2}{2}\right)}\right] = \frac{1}{(1-u)\left(\log(1-u)\right)^2} - \frac{1}{2\left(\log(1-u)\right)^2} + \frac{1}{\left(\log(1-u)\right)^3} + \mathcal{O}\left(\frac{(1-u)}{\left(\log(1-u)\right)^2}\right), \tag{2.B.2}$$

$$\partial^2_{uu}\left[\frac{1}{\log\left(\frac{1-u^2}{2}\right)}\right] = \frac{1}{(1-u)^2\left(\log(1-u)\right)^2} + \frac{2}{(1-u)^2\left(\log(1-u)\right)^3} - \frac{1}{(1-u)\left(\log(1-u)\right)^3} + \frac{3}{(1-u)\left(\log(1-u)\right)^4} + \frac{1}{4\left(\log(1-u)\right)^2} + \mathcal{O}\left(\frac{1}{\left(\log(1-u)\right)^3}\right). \tag{2.B.3}$$

(ii) Assume $(\mathbf{H_1})$ *and* $(\mathbf{H_2})$ *hold. Then,*

$$q_Y(u) = \eta^{-\gamma}(1-u)^{-\gamma} L\left(\frac{1}{(1-u)\eta}\right), \tag{2.B.4}$$

$$\partial_u q_Y(u) = \eta^{-\gamma}(1-u)^{-(\gamma+1)} L\left(\frac{1}{(1-u)\eta}\right)\left(\gamma + \varepsilon\left(\frac{1}{(1-u)\eta}\right)\right), \tag{2.B.5}$$

$$\log q_Y(u) = -\gamma\log(1-u) - \beta + \frac{1}{\rho}\varepsilon\left(\frac{1}{(1-u)\eta}\right)\big(1 + o(1)\big), \ \text{as } u \to 1, \tag{2.B.6}$$

$$\partial_u \log q_Y(u) = (1-u)^{-1}\left(\gamma + \varepsilon\left(\frac{1}{(1-u)\eta}\right)\right). \tag{2.B.7}$$

(iii) Assume $(\mathbf{H_1})$, $(\mathbf{H_2})$ *and* $(\mathbf{H_3})$ *hold. Then, as $u \to 1$,*

$$\partial^2_{uu} q_Y(u) = \eta^{-\gamma}(1-u)^{-(\gamma+2)} L\left(\frac{1}{(1-u)\eta}\right) \times \left[\gamma^2 + \gamma + (1 + 2\gamma + \rho + o(1))\varepsilon\left(\frac{1}{(1-u)\eta}\right)\right], \tag{2.B.8}$$

$$\partial^2_{uu} \log q_Y(u) = (1-u)^{-2}\left(\gamma + \varepsilon\left(\frac{1}{(1-u)\eta}\right)(1 + \rho + o(1))\right). \tag{2.B.9}$$

Proof. *(i)* The proof of (2.B.1)–(2.B.3) is straightforward but requires tedious calculations which can be checked by a formal calculation software (using `sympy` in `Python` for instance, see below). Details are omitted here.

```
import sympy as spy
u = spy.symbols('u')

f = 1 / spy.log((1 - u ** 2) / 2)
```

```
# series as u->1
f.series(u, 1, 2, dir="-")

f_first = spy.diff(f, u)
f_first.series(u, 1, 1, dir="-")

f_second = spy.diff(f_first, u)
f_second.series(u, 1, 1, dir="-")
```

(ii) Under $(\mathbf{H_1})$, Equations (2.2.1), (2.2.2) and (2.2.4) entail

$$q_Y(u) = \eta^{-\gamma}(1-u)^{-\gamma} L\left(\frac{1}{(1-u)\eta}\right),$$

which proves (2.B.4) and moreover, owing to $(\mathbf{H_2})$,

$$\log q_Y(u) = -\gamma \log(1-u) + \log(c_\infty) - \gamma \log \eta + \int_1^{\frac{1}{(1-u)\eta}} \frac{\varepsilon(t)}{t}\,\mathrm{d}t. \tag{2.B.10}$$

By differentiating, we get

$$\partial_u q_Y(u) = q_Y(u) \times \partial_u(\log q_Y(u)) = \eta^{-\gamma}(1-u)^{-(\gamma+1)} L\left(\frac{1}{(1-u)\eta}\right)\left(\gamma + \varepsilon\left(\frac{1}{(1-u)\eta}\right)\right),$$

and (2.B.5) is proved. Now, $t \mapsto \varepsilon(t)/t$ is regularly varying with index $\rho - 1 < -1$ and thus, $\int_1^\infty \varepsilon(t)/t\,\mathrm{d}t$ is finite leading to:

$$\log L_\infty = \log c_\infty + \int_1^\infty \frac{\varepsilon(t)}{t}\,\mathrm{d}t.$$

Replacing in (2.B.10) yields:

$$\log q_Y(u) = -\gamma \log(1-u) - \beta - \int_{\frac{1}{(1-u)\eta}}^\infty \frac{\varepsilon(t)}{t}\,\mathrm{d}t.$$

Moreover, Karamata's theorem [52, Equation (B.1.9)] states that

$$\int_x^\infty \frac{\varepsilon(t)}{t}\,\mathrm{d}t = -\frac{1}{\rho}\varepsilon(x)(1+o(1)),$$

as $x \to \infty$ so that (2.B.6) is proved. Finally, (2.B.7) is a direct consequence of (2.B.4) and (2.B.5).

(iii) From (2.B.5), letting $U(u) = \eta^{-\gamma}(1-u)^{-(\gamma+1)} L\left(\frac{1}{(1-u)\eta}\right)$, one has

$$\partial^2_{uu} q_Y(u) = \partial_u\left[U(u)\left(\gamma + \varepsilon\left(\frac{1}{(1-u)\eta}\right)\right)\right]. \tag{2.B.11}$$

Using the form of L under $(\mathbf{H_2})$ and $x\frac{\partial_x L(x)}{L(x)} = \varepsilon(x)$, we obtain

$$\partial_u U(u) = \eta^{-\gamma}(1-u)^{-(\gamma+2)} L\left(\frac{1}{(1-u)\eta}\right)\left(\gamma + 1 + \varepsilon\left(\frac{1}{(1-u)\eta}\right)\right). \tag{2.B.12}$$

In addition, recalling that ε is differentiable under $(\mathbf{H_3})$ yields

$$\partial_u\left[\varepsilon\left(\frac{1}{(1-u)\eta}\right)\right] = \eta^{-\rho}(1-u)^{-(\rho+1)}\ell\left(\frac{1}{(1-u)\eta}\right)\left(\rho+\frac{1}{(1-u)\eta}\frac{\partial\ell\left(\frac{1}{(1-u)\eta}\right)}{\ell\left(\frac{1}{(1-u)\eta}\right)}\right)$$

$$= \frac{1}{(1-u)}\varepsilon\left(\frac{1}{(1-u)\eta}\right)\left(\rho+\frac{1}{(1-u)\eta}\frac{\partial\ell\left(\frac{1}{(1-u)\eta}\right)}{\ell\left(\frac{1}{(1-u)\eta}\right)}\right) \tag{2.B.13}$$

$$= \frac{1}{(1-u)}\varepsilon\left(\frac{1}{(1-u)\eta}\right)(\rho+o(1))\,. \tag{2.B.14}$$

Collecting (2.B.11), (2.B.12) and (2.B.14) entails

$$\begin{aligned}\partial^2_{uu}q_Y(u) &= \eta^{-\gamma}(1-u)^{-(\gamma+2)}L\left(\frac{1}{(1-u)\eta}\right)\\ &\quad\times\left[\left(\gamma+\varepsilon\left(\frac{1}{(1-u)\eta}\right)\right)^2+\gamma+(1+\rho+o(1))\varepsilon\left(\frac{1}{(1-u)\eta}\right)\right]\\ &= \eta^{-\gamma}(1-u)^{-(\gamma+2)}L\left(\frac{1}{(1-u)\eta}\right)\\ &\quad\times\left[\gamma^2+\gamma+(1+2\gamma+\rho+o(1))\varepsilon\left(\frac{1}{(1-u)\eta}\right)\right]\end{aligned}$$

which proves (2.B.8). Finally, (2.B.7) and (2.B.13) entail

$$\begin{aligned}\partial^2_{uu}\log q_Y(u) &= (1-u)^{-2}\left(\gamma+\varepsilon\left(\frac{1}{(1-u)\eta}\right)\right)\\ &\quad+(1-u)^{-2}\varepsilon\left(\frac{1}{(1-u)\eta}\right)\left(\rho+\frac{1}{(1-u)\eta}\frac{\partial\ell\left(\frac{1}{(1-u)\eta}\right)}{\ell\left(\frac{1}{(1-u)\eta}\right)}\right)\end{aligned} \tag{2.B.15}$$

and (2.B.9) is proved owing to $(\mathbf{H_3})$. □

Lemma 2.B.3. *Let* li *be the logarithmic integral function defined for all* $u\in(0,1)$ *as*

$$\mathrm{li}(u) = \int_0^u \frac{1}{\log(t)}\,\mathrm{d}t.$$

Then, for any $p>0$, $u^p\,\mathrm{li}(1-u)\to 0$ *as* $u\to 0$.

Proof. This stems from the convexity inequality $\log(1/t)\geq 1-t$ for $t\in(0,1]$. □

Lemma 2.B.4. *For all* $u\in(0,1)$, *let* $\Phi(u)=\sum_{j=0}^3 c_j\Phi_j(u)$. *One has:*

$$\begin{aligned}\partial^2_{uu}\left[g(u)\left(\gamma+\Phi(u)\right)\right] &= -20\gamma+\frac{c_0}{(1-u)^2\left(\log(1-u)\right)^2}+\frac{2c_0}{(1-u)^2\left(\log(1-u)\right)^3}\\ &\quad+\frac{c_1}{(1-u)\left(\log(1-u)\right)^2}+\frac{2c_2}{(1-u)\left(\log(1-u)\right)^3}+\frac{3c_3}{(1-u)\left(\log(1-u)\right)^4}\\ &\quad+\mathcal{O}\left(\frac{1}{\log(1-u)}\right), \qquad \text{as } u\to 1,\end{aligned} \tag{2.B.16}$$

$$\partial^2_{uu}\left[g(u)\left(\gamma+\Phi(u)\right)\right]\to 5\beta, \qquad \text{as } u\to 0. \tag{2.B.17}$$

Proof. Differentiating Φ yields for all $u \in (0,1)$,

$$\begin{aligned}\partial_u \Phi(u) &= \sum_{j=0}^{3} c_j \varphi_j(u),\\ \partial^2_{uu}\Phi(u) &= \frac{c_0}{(1-u)^2\left(\log(1-u)\right)^2} + \frac{2c_0}{(1-u)^2\left(\log(1-u)\right)^3} + \frac{c_1}{(1-u)\left(\log(1-u)\right)^2}\\ &\quad + \frac{2c_2}{(1-u)\left(\log(1-u)\right)^3} + \frac{3c_3}{(1-u)\left(\log(1-u)\right)^4}.\end{aligned}$$

Besides, for all $u \in (0,1)$,

$$\begin{aligned}\partial^2_{uu}\left[g(u)\left(\gamma+\Phi(u)\right)\right] &= 20u^2(3-4u)\left(\gamma+\Phi(u)\right) + 40u^3(1-u)\,\partial_u\Phi(u)\\ &\quad + u^4(5-4u)\,\partial^2_{uu}\Phi(u). \qquad (2.B.18)\end{aligned}$$

Remarking that $\Phi(u) = \mathcal{O}\left(1/\log(1-u)\right)$ and $(1-u)\partial_u\Phi(u) = \mathcal{O}\left(1/(\log(1-u))^2\right)$ as $u \to 1$ proves (2.B.16). Similarly, Lemma 2.B.3 entails that $\mathrm{li}(1-u) = \mathcal{O}\left(1/u\right)$ as $u \to 0$ and thus $\Phi(u) = c_3/(2u^2)(1+o(1))$, $\partial_u\Phi(u) = -c_3/u^3(1+o(1))$ and $\partial^2_{uu}\Phi(u) = 3c_3/u^4(1+o(1))$ as $u \to 0$. Replacing in (2.B.18) and taking the limit as $u \to 0$ gives (2.B.17). □

The next Lemma provides a sufficient condition for a given function to belong to $\mathcal{C}^{0,\alpha}([0,1])$.

Lemma 2.B.5. *Let $g : [0,1] \to \mathbb{R}$ be a continuous function on $[0,1]$ and differentiable on $(0,1)$ such that $|\partial_u g(u)| \le Cu^{\alpha-1}$ for all $u \in (0,1)$ with $0 < \alpha \le 1$ and $C > 0$. Then, $g \in \mathcal{C}^{0,\alpha}([0,1])$.*

Proof. Let $0 \le a < b \le 1$, then

$$|g(b) - g(a)| \le \int_a^b Cx^{\alpha-1}\,\mathrm{d}x \le \int_a^b C(x-a)^{\alpha-1}\,\mathrm{d}x = \frac{C}{\alpha}(b-a)^{\alpha},$$

and the conclusion follows. □

Our goal here is to study the uniform convergence rate of the approximation error of a $\mathcal{C}^{1,\alpha}([0,1])$ or $C^2([0,1])$ function f by a NN. To this end, consider a triangular function $\widehat{\sigma} : \mathbb{R} \to [-1,1]$ built using three translated ReLU functions $x \in \mathbb{R} \mapsto \sigma(x) := \max(0, x)$:

$$\widehat{\sigma}(t) := \sigma(t+1) - 2\sigma(t) + \sigma(t-1) = \begin{cases} 1, & \text{if } t = 0,\\ 1+t, & \text{if } -1 < t < 0,\\ 1-t, & \text{if } 0 < t < 1,\\ 0, & \text{otherwise.}\end{cases}$$

It is then possible to control the uniform error between the function f and its piecewise linear approximation based on triangular functions, depending on the regularity of f.

Lemma 2.B.6. *Let $\widehat{\sigma}$ be a triangular function and $f : [0,1] \to \mathbb{R}$. For all $M \in \mathbb{N}\backslash\{0\}$, let $\delta = 1/M$ and $t_j = j/M$ for $j = 0, \dots, M$. If $f \in \mathcal{C}^{1,\alpha}([0,1])$ with $\alpha \in (0,1]$, then*

$$\sup_{t\in[0,1]} \left| f(t) - \sum_{j=0}^{M} f(t_j)\widehat{\sigma}\left(\frac{t-t_j}{\delta}\right)\right| \le \frac{[\partial_t f]_\alpha}{4} M^{-\alpha-1}. \qquad (2.B.19)$$

Proof. Clearly,

$$\sup_{t\in[0,1]} \left| f(t) - \sum_{j=0}^{M} f(t_j)\widehat{\sigma}\left(\frac{t-t_j}{\delta}\right)\right| =: \max_{i=0,\dots,M-1} \sup_{t\in[t_i,t_{i+1}]} \left|\Delta_i(t)\right|,$$

where

$$\Delta_i(t) := f(t) - \left(f(t_i) \left(\frac{t_{i+1} - t}{\delta} \right) + f(t_{i+1}) \left(\frac{t - t_i}{\delta} \right) \right).$$

Two first order Taylor expansions yield that there exist $t_i' \in (t_i, t)$ and $t_i'' \in (t, t_{i+1})$ such that

$$\begin{aligned} \Delta_i(t) &= f(t) - \left[\big(f(t) + \partial_t f(t_i')(t_i - t) \big) \left(\frac{t_{i+1} - t}{\delta} \right) + \big(f(t) + \partial_t f(t_i'')(t_{i+1} - t) \big) \left(\frac{t - t_i}{\delta} \right) \right] \\ &= \frac{(t_{i+1} - t)(t - t_i)}{\delta} \left(\partial_t f(t_i') - \partial_t f(t_i'') \right). \end{aligned}$$

Remarking $(t_{i+1} - t)(t - t_i)$ is maximum on $[t_i, t_{i+1}]$ at $t = (t_{i+1} + t_i)/2$ entails

$$|\Delta_i(t)| \leq \frac{\delta}{4} \left| \partial_t f(t_i') - \partial_t f(t_i'') \right| \leq \frac{\delta}{4} [\partial_t f]_\alpha (t_i'' - t_i')^\alpha \leq \frac{1}{4} [\partial_t f]_\alpha \delta^{\alpha+1},$$

and the result is proved. □

Finally, one can determine the minimum number $J(\epsilon)$ of ReLU functions to approximate f with a given precision ϵ. The above construction in Lemma 2.B.6 involves $(M+1)$ triangular functions corresponding to $J = 3(M+1)$ ReLU functions. Fixing bound (2.B.19) to ϵ provides M as a function of ϵ, and we obtain:

Lemma 2.B.7. *Let σ be a ReLU function and $f \in \mathcal{C}^{1,\alpha}([0,1])$ with $\alpha \in (0,1]$. For all $\epsilon > 0$, let $J(\epsilon) = 3(M(\epsilon)+1)$ with $M(\epsilon) \in \mathbb{N}$ such that*

$$M(\epsilon) \geq \left(\frac{[\partial_t f]_\alpha}{4\epsilon} \right)^{1/(\alpha+1)}.$$

Then, there exist $(a_j, w_j, b_j) \in \mathbb{R}^3$, $j = 1, \ldots, J(\epsilon)$ such that

$$\sup_{t \in [0,1]} \left| f(t) - \sum_{j=1}^{J(\epsilon)} a_j \sigma(w_j t + b_j) \right| \leq \epsilon.$$

The above lemma is not that surprising, a similar result is stated in [189, Theorem 1] up to a log factor but under the condition that $1 + \alpha$ is an integer.

2.C Proof of main results

Proof. of Proposition 2.2.1. (i) The continuity of f^{TIF} on $(0,1)$ is a consequence of the assumptions on F_X. Besides, $q_X(1-\eta) = 1$ and thus

$$f^{\mathrm{TIF}}(0) = \log(q_X(1-\eta))/\log 2 = 0.$$

From $(\mathbf{H_1})$, the cumulative distribution function F_X has an unbounded right-hand support, and thus, from (2.2.2), $q_Y(u) \to \infty$ as $u \to 1$. Thus, replacing in (2.2.1) and taking the log yields

$$\log q_Y(u) = -\gamma \log\big((1-u)\eta\big) \left(1 - \frac{\log L\left(\frac{1}{(1-u)\eta}\right)}{\gamma \log\big((1-u)\eta\big)} \right).$$

Since L is slowly varying, $\log L(v)/\log v \to 0$ as $v \to \infty$ [26, Proposition 1.3.6] and then,

$$\log q_Y(u) = -\gamma \log(1-u)(1 + o(1)), \text{ as } u \to 1.$$

Similarly, as $u \to 1$, $\log\left(\frac{1-u^2}{2}\right) = \log(1-u)(1+o(1))$, which leads to $f^{\mathrm{TIF}}(u) \to \gamma$ as $u \to 1$. Finally, f^{TIF} is bounded on $[0,1]$ and the conclusion follows.
(ii) First, $q_Y(0) = 1$ directly yields

$$\partial_u f^{\mathrm{TIF}}(0) = \frac{\gamma + \varepsilon\left(1/\eta\right)}{\log(2)}.$$

Second, collecting (2.B.1) and (2.B.7), it follows, as $u \to 1$,

$$\begin{aligned}\frac{\partial_u \log q_Y(u)}{\log\left(\frac{1-u^2}{2}\right)} &= \frac{\gamma}{(1-u)\log(1-u)} + \frac{\gamma}{2\left(\log(1-u)\right)^2} + \frac{\varepsilon\left(\frac{1}{(1-u)\eta}\right)}{(1-u)\log(1-u)} + \mathcal{O}\left(\frac{(1-u)}{\left(\log(1-u)\right)^2}\right) \\ &= \frac{\gamma}{(1-u)\log(1-u)} + \frac{\gamma}{2}\varphi_2(u) + \frac{\varepsilon\left(\frac{1}{(1-u)\eta}\right)}{(1-u)\log(1-u)} + \mathcal{O}\left(\frac{(1-u)}{\left(\log(1-u)\right)^2}\right).\end{aligned}$$

In addition, from (2.B.2) and (2.B.6), we have, as $u \to 1$,

$$\begin{aligned}\log q_Y(u)\partial_u\left[\frac{1}{\log\left(\frac{1-u^2}{2}\right)}\right] &= \frac{-\gamma}{(1-u)\log(1-u)} - \frac{\beta}{(1-u)\left(\log(1-u)\right)^2} + \frac{\gamma}{2\log(1-u)} \\ &\quad - \frac{(2\gamma-\beta)}{2\left(\log(1-u)\right)^2} - \frac{\beta}{\left(\log(1-u)\right)^3} + \frac{\varepsilon\left(\frac{1}{(1-u)\eta}\right)(1+o(1))}{\rho(1-u)\left(\log(1-u)\right)^2} \\ &\quad + \mathcal{O}\left(\frac{(1-u)}{\log(1-u)}\right) \\ &= \frac{-\gamma}{(1-u)\log(1-u)} - \beta\varphi_0(u) + \frac{\gamma}{2}\varphi_1(u) - \frac{(2\gamma-\beta)}{2}\varphi_2(u) + \beta\varphi_3(u) \\ &\quad + \frac{\varepsilon\left(\frac{1}{(1-u)\eta}\right)(1+o(1))}{\rho(1-u)\left(\log(1-u)\right)^2} + \mathcal{O}\left(\frac{(1-u)}{\log(1-u)}\right).\end{aligned}$$

Summing up the two above expansions and inverting the signs yield

$$\begin{aligned}\partial_u f^{\mathrm{TIF}}(u) &= \beta\varphi_0(u) - \frac{\gamma}{2}\varphi_1(u) + \frac{\gamma-\beta}{2}\varphi_2(u) + \beta\varphi_3(u) \\ &\quad - \frac{\varepsilon\left(\frac{1}{(1-u)\eta}\right)}{(1-u)\log(1-u)}\left(1 + \frac{1}{\rho\log(1-u)}(1+o(1))\right) + \mathcal{O}\left(\frac{(1-u)}{\log(1-u)}\right),\end{aligned}$$

which proves the result. □

Proof. of Proposition 2.2.2. For all $u \in (0,1)$, let $\Phi(u) = \sum_{j=0}^{3} c_j \Phi_j(u)$.
(i) First, note that $\Phi(u) \to 0$ as $u \to 1$, $h(1) = 0$ and $g(1) = 1$. Besides, Proposition 2.2.1(i) shows that $f^{\mathrm{TIF}}(u) \to \gamma$ as $u \to 1$ and therefore $f^{\mathrm{CTIF}}(u) \to 0$ as $u \to 1$. Second, Lemma 2.B.3 entails that $\mathrm{li}(1-u) = \mathcal{O}\left(1/u\right)$ as $u \to 0$ and thus $\Phi(u) = c_3/(2u^2)(1+o(1))$. It follows that $g(u)\Phi(u) \to 0$ as $u \to 0$. Clearly, one also has $g(0) = h(0) = 0$. Besides, Proposition 2.2.1(i) shows that $f^{\mathrm{TIF}}(0) = 0$ and therefore $f^{\mathrm{CTIF}}(u) \to 0$ as $u \to 0$.
(ii) First, differentiating (2.2.6) and taking account of $g'(1) = h'(1) = 0$, $\Phi(u) \to 0$ as $u \to 1$ yields

$$\partial_u f^{\mathrm{CTIF}}(u) = \partial_u f^{\mathrm{TIF}}(u) - \partial_u\Phi(u)g(u) + o(1) = \partial_u\Phi(u)\left(1-g(u)\right) + o(1),$$

as $u \to 1$, since $\partial_u f(u) = \partial_u\Phi(u) + o(1)$ when $\rho < -1$, in view of (2.2.5) in Proposition 2.2.1(ii). Remarking that $1 - g(u) = o(1-u)$ and recalling from the proof of Lemma 2.B.4 that $(1-$

$u)\partial_u\Phi(u) = \mathcal{O}\left(1/(\log(1-u))^2\right)$ as $u \to 1$ prove that $\partial_u f^{\mathrm{CTIF}}(u) \to 0$ as $u \to 1$. Second, taking account of $g'(0) = 0$ and $h'(0) = 1$ yields

$$\partial_u f^{\mathrm{CTIF}}(u) = -g(u)\partial_u\Phi(u) - \Phi(u)\partial_u g(u) + o(1),$$

as $u \to 0$. Recall from the proof of Lemma 2.B.4 that $\Phi(u) = c_3/(2u^2)(1+o(1))$ and $\partial_u\Phi(u) = -c_3/u^3(1+o(1))$ as $u \to 0$. Since $g(u) = o(u^3)$ and $\partial_u g(u) = o(u^2)$ as $u \to 0$, it follows that $\partial_u f^{\mathrm{CTIF}}(u) \to 0$ as $u \to 0$ and (2.2.9) is proved.
(iii) The first part of the proof is based on successive applications of Lemma 2.B.2. From (2.B.1) and (2.B.9), one has, as $u \to 1$:

$$\partial^2_{uu}\left[\log q_Y(u)\right]\frac{1}{\log\left(\frac{1-u^2}{2}\right)} = \frac{\gamma}{(1-u)^2\log(1-u)} + \frac{\gamma}{2(1-u)\left(\log(1-u)\right)^2} + \frac{\varepsilon\left(\frac{1}{(1-u)\eta}\right)}{(1-u)^2\log(1-u)}(1+\rho+o(1)) + \mathcal{O}\left(\frac{1}{\left(\log(1-u)\right)^2}\right).$$

Similarly, from (2.B.2) and (2.B.7), as $u \to 1$,

$$\partial_u\left[\log q_Y(u)\right]\partial_u\left[\frac{1}{\log\left(\frac{1-u^2}{2}\right)}\right] = \frac{\gamma}{(1-u)^2\left(\log(1-u)\right)^2} - \frac{\gamma}{2(1-u)\left(\log(1-u)\right)^2} + \frac{\gamma}{(1-u)\left(\log(1-u)\right)^3} + \frac{\varepsilon\left(\frac{1}{(1-u)\eta}\right)}{(1-u)^2\left(\log(1-u)\right)^2} + \mathcal{O}\left(\frac{1}{\left(\log(1-u)\right)^2}\right),$$

and, from (2.B.3) and (2.B.6),

$$\log(q_Y(u))\partial^2_{uu}\left[\frac{1}{\log\left(\frac{1-u^2}{2}\right)}\right] = -\frac{\gamma}{(1-u)^2\log(1-u)} - \frac{2\gamma+\beta}{(1-u)^2\left(\log(1-u)\right)^2} - \frac{2\beta}{(1-u)^2\left(\log(1-u)\right)^3} + \frac{\gamma}{(1-u)\left(\log(1-u)\right)^2} - \frac{3\gamma-\beta}{(1-u)\left(\log(1-u)\right)^3} - \frac{3\beta}{(1-u)\left(\log(1-u)\right)^4} + \frac{\varepsilon\left(\frac{1}{(1-u)\eta}\right)(1+o(1))}{\rho(1-u)^2(\log(1-u))^2} + \frac{2\varepsilon\left(\frac{1}{(1-u)\eta}\right)(1+o(1))}{\rho(1-u)^2(\log(1-u))^3} - \frac{\gamma}{4\log(1-u)} + \mathcal{O}\left(\frac{1}{\left(\log(1-u)\right)^2}\right).$$

Collecting the above three asymptotic expansions yields, as $u \to 1$,

$$\partial^2_{uu} f^{\mathrm{TIF}}(u) = \frac{\beta}{(1-u)^2\left(\log(1-u)\right)^2} + \frac{2\beta}{(1-u)^2\left(\log(1-u)\right)^3} - \frac{\gamma}{2(1-u)\left(\log(1-u)\right)^2} + \frac{\gamma-\beta}{(1-u)\left(\log(1-u)\right)^3} + \frac{3\beta}{(1-u)\left(\log(1-u)\right)^4} + \frac{\gamma}{4\log(1-u)} - \frac{(1+\rho)\varepsilon\left(\frac{1}{(1-u)\eta}\right)}{(1-u)^2\log(1-u)}(1+o(1)) + \mathcal{O}\left(\frac{1}{\left(\log(1-u)\right)^2}\right). \tag{2.C.1}$$

In addition, note that $h''(1) = 2$ and $\partial_u f^{\mathrm{TIF}}(0) = (\gamma + \varepsilon(1/\eta))/\log(2)$ in view of Proposition 2.2.1(ii), so that collecting (2.B.16) in Lemma 2.B.4 with (2.C.1) proves (2.2.10). The second part of the proof consists in remarking that $\log q_Y(0) = 0$ by construction and $\partial_u \left[\frac{1}{\log\left(\frac{1-u^2}{2}\right)}\right](0) = 0$. Therefore, taking account of (2.B.15), it follows:

$$\partial^2_{uu} f^{\mathrm{TIF}}(0) = \frac{\partial^2_{uu}\left[\log\left(q_Y(u)\right)\right](0)}{\log(2)} = \frac{\gamma + \varepsilon(1/\eta)\left(1 + \rho + \frac{1}{\eta}\frac{\partial \ell(1/\eta)}{\ell(1/\eta)}\right)}{\log(2)}. \tag{2.C.2}$$

Finally, note that $h''(0) = -4$ and $\partial_u f^{\mathrm{TIF}}(0) = (\gamma + \varepsilon(1/\eta))/\log(2)$ in view of Proposition 2.2.1(ii), so that collecting (2.B.17) in Lemma 2.B.4 with (2.C.2) proves (2.2.11).
(iv) is a direct consequence of *(iii)*. □

Proof. of Corollary 2.2.4. Theorem 2.2.1 yields, uniformly on $u \in [0,1]$:

$$\left|\frac{\log q_Y(u) - \log \widetilde{q}_Y(u)}{\log((1-u^2)/2)}\right| \leq c(J),$$

with $c(J) := \frac{[\partial_t f^{\mathrm{CTIF}}]_\alpha}{4}\left\lceil \frac{J-3}{3}\right\rceil^{-\alpha-1} \to 0$ as $J \to \infty$. It follows, for all $u \in [0,1]$:

$$q_Y(u)\left(\frac{1-u^2}{2}\right)^{c(J)} \leq \widetilde{q}_Y(u) \leq q_Y(u)\left(\frac{1-u^2}{2}\right)^{-c(J)}.$$

Substracting $q_Y(u)$ and integrating, we obtain $W_1(q, \widetilde{q}_Y) \leq \max\{D(-c(J)), -D(c(J))\}$ where

$$D(t) := \int_0^1 q_Y(u)\left(\left(\frac{1-u^2}{2}\right)^t - 1\right)\mathrm{d}u \tag{2.C.3}$$

is defined for all (γ, t) such that $\gamma - t < 1$. Recall that $\gamma < 1$ and let us thus consider J large enough so that $\gamma + c(J) < 1$. Expanding (2.C.3) as $t \to 0$ yields

$$D(t) = t\int_0^1 q_Y(u)\log((1-u^2)/2)\,\mathrm{d}u\,(1 + o(1)),$$

and the conclusion follows. □

Proof. of Corollary 2.2.3. *(i)* When $-2 \leq \rho \leq -1$, Proposition 2.2.2(iii) implies $f^{\mathrm{CTIF}} \in C^2([0,1))$ and

$$\left|\partial^2_{uu} f^{\mathrm{CTIF}}(u)\right| \leq C(1-u)^{\alpha-1}, \quad \forall u \in (0,1),$$

for any fixed $\alpha \in (0, -\rho - 1)$. Thus, applying Lemma 2.B.5 to $\partial_u f^{\mathrm{CTIF}}$ yields $f^{\mathrm{CTIF}} \in \mathcal{C}^{1,\alpha}([0,1])$.
(ii) is a direct consequence of Proposition 2.2.2(iv). □

Hyperparameters ranges								
	latent dimension	batch size	neurons G.	learning rate G.	hidden layers D.	neurons D.	learning rate D.	training loop D.
A	$[10, 100]$	$[5 - 64]$	$[10, 500]$	$[0.0001, 0.01]$	$[1, 4]$	$[10, 500]$	$[0.0001, 0.01]$	$[1, 5]$
B	$[10, 2000]$	$[5 - 256]$	$[10, 500]$	$[0.0001, 0.01]$	$[1, 4]$	$[10, 500]$	$[0.0001, 0.01]$	$[1, 5]$
1. Selected hyperparameters on bivariate simulated data (setting A)								
MSLE(0.90)								
EV-GAN	31 (18)	32 (22)	47 (22)	0.0005 (0.0004)	2 (0)	28 (27)	0.0005 (0.0004)	1 (0)
GAN	30 (8)	28 (21)	43 (24)	0.0007 (0.0004)	2 (0)	29 (26)	0.0007 (0.0004)	1 (0)
MSLE(0.95)								
EV-GAN	36 (13)	31 (18)	68 (54)	0.0004 (0.0004)	2 (0)	45 (39)	0.0004 (0.0004)	1 (0)
GAN	30 (8)	26 (20)	44 (24)	0.001 (0.0004)	2 (0)	30 (26)	0.0007 (0.0004)	1 (0)
MSLE(0.99)								
EV-GAN	31 (13)	33 (22)	46 (22)	0.0005 (0.0005)	2 (0)	26 (27)	0.0005 (0.0005)	1 (0)
GAN	31 (8)	29 (21)	5426 (24)	0.0007 (0.0004)	3 (1)	27 (27)	0.0007 (0.0004)	1 (0)
AKE								
EV-GAN	40 (14)	30 (15)	90 (67)	0.0003 (0.0003)	2 (0)	63 (41)	0.0003 (0.0004)	1 (0)
GAN	36 (14)	24 (9)	70 (31)	0.002 (0.0003)	2 (0)	57 (42)	0.003 (0.0003)	1 (0)
2. Selected hyperparameters on multivariate simulated data (setting B)								
MSLE(0.90)								
EV-GAN	700 (523)	114 (88)	214 (172)	0.0007 (0.0004)	3 (1)	57 (31)	0.0007 (0.0004)	1 (1)
GAN	306 (412)	61 (37)	138 (152)	0.0006 (0.0005)	3 (1)	34 (22)	0.0006 (0.0005)	1 (0)
MSLE(0.95)								
EV-GAN	1065 (509)	171 (105)	167 (132)	0.0007 (0.0004)	2 (1)	75 (30)	0.0007 (0.0004)	2 (1)
GAN	306 (412)	61 (37)	138 (152)	0.0006 (0.0005)	3 (1)	34 (22)	0.0006 (0.0005)	1 (0)
MSLE(0.99)								
EV-GAN	1065 (509)	171 (105)	167 (132)	0.0007 (0.0004)	2 (1)	75 (30)	0.0007 (0.0004)	2 (1)
GAN	306 (412)	61 (37)	138 (152)	0.0006 (0.0005)	3 (1)	34 (22)	0.0006 (0.0005)	1 (0)
AKE								
EV-GAN	345 (394)	51 (33)	120 (65)	0.0003 (0.0004)	3 (1)	35 (12)	0.0003 (0.0004)	1 (0)
GAN	411 (464)	121 (107)	95 (50)	0.0003 (0.0004)	2 (0)	30 (13)	0.0003 (0.0004)	2 (1)
3. Selected hyperparameters on real data (setting A)								
MSLE(0.90)								
EV-GAN	21 (9)	9 (5)	18 (10)	0.0001 (0)	3 (0)	12 (2)	0.0006 (0.0005)	1 (0)
GAN	28 (5)	8 (0)	28 (5)	0.0001 (0.)	3 (0)	10 (0)	0.0008 (0.0005)	2 (1)
MSLE(0.95)								
EV-GAN	19 (9)	15 (13)	13 (5)	0.0003 (0.0005)	3 (1)	14 (4)	0.0006 (0.0005)	1 (0)
GAN	45 (37)	10 (4)	70 (87)	0.0001 (0.)	3 (0)	58 (95)	0.0006 (0.0005)	2 (1)
MSLE(0.99)								
EV-GAN	18 (10)	5 (0)	10 (0)	0.0001 (0.)	2 (1)	13 (2)	0.0001 (0.)	1 (0)
GAN	45 (37)	10 (4)	70 (87)	0.0001 (0.)	3 (1)	58 (95)	0.0006 (0.0005)	2 (1)
AKE								
EV-GAN	20 (12)	7 (2)	20 (12)	0.0003 (0.0005)	3 (0)	11 (2)	0.0006 (0.0005)	1 (0)
GAN	46 (36)	11 (4)	68 (89)	0.0001 (0.)	3 (1)	58 (94)	0.0006 (0.0005)	1 (0)

Table 2.5: Hyperparameters ranges used for tuning GANs across the experiments and mean (standard deviation) of selected hyperparameters in three situations: 1. simulated bivariate data, selection according to the MSLE(0.99) and AKE criteria, 2. simulated multivariate data, selection according to MSLE(ξ) for $\xi \in \{0.90, 0.95, 0.99\}$, 3. real data, selection according to the MSLE(0.95) and AKE criteria.

Chapter 3

Estimation of extreme quantiles from heavy-tailed distributions with neural networks

Note. The results of this chapter are based on the paper [5]

Abstract. We propose new parametrizations for neural networks in order to estimate extreme quantiles in both non-conditional and conditional heavy-tailed settings. All proposed neural network estimators feature a bias correction based on an extension of the usual second-order condition to an arbitrary order. The convergence rate of the uniform error between extreme log-quantiles and their neural network approximation is established. The finite sample performances of the non-conditional neural network estimator are compared to other bias-reduced extreme-value competitors on simulated data. It is shown that our method outperforms them in difficult heavy-tailed situations where other estimators almost all fail. The source code is available at `https://github.com/michael-allouche/nn-quantile-extrapolation.git`. Finally, the conditional neural network estimators are implemented to investigate the behavior of extreme rainfalls as functions of their geographical location in the southern part of France.

3.1 Introduction

Nowadays, dealing with extreme events is a major issue when assessing climate risk [118, Section 7]. We have in mind the Vergas (Venezuela) tragedy, the storm Alex in the Roya Valley (France) or the hurrican Harvey (Texas, USA), where tremendous rainfall occurred. The challenge of modeling the statistical behavior of such events will be exacerbated in the next future with the climate change and will impact other meteorological variables (temperatures, heat waves, ...) according to the last Intergovernmental Panel on Climate Change report [166].

Our objective is to infer out-of-sample quantiles from heavy-tailed distributions based on observations, namely using n data points, we seek to estimate the so-called *extreme quantiles* corresponding to tail probabilities α_n smaller than $1/n$. In the climate framework considered in the illustrative example of Section 3.7, the data come either from historical station measurements or, taking into account the climate change, from climate forecast scenarios through physical models (see DRIAS for projections in France). In both cases, few extreme historical data are usually available to estimate tail quantities which is therefore a challenging problem. Besides, an accurate quantile estimator is an essential tool in risk quantification, which also reveals useful for tail events simulation in the context of generative modeling [6, 21, 186]. Let us however note that such models usually have a distribution support restricted to the one observed in

the training dataset. Our contributions are both theoretical and numerical, and can potentially address applications in climatic risk ([118, Section 7]), in finance where the data are known to be heavy-tailed ([63, p. 9]), in the study of extreme bank losses [32], in flood risk assessment [173] and in oceanographic data [54]. We also refer to the books [16, 52, 63] for a general overview of the theoretical background on extreme quantile estimation.

One of the most famous estimators in such a context is the Weissman estimator [184] described in Section 3.2 thereafter. Basing on its asymptotic representation [52, Theorem 4.3.8], bias-reduced estimators have been introduced [94] thanks to a prior estimation of additional parameters driving the dominant bias component. Our first main contribution is to show that all first J bias terms benefit from a natural (one hidden layer) neural network (NN) representation with the popular eLU activation function, where J is linked to the number of neurons in the network. Based on this result, we derive a NN extreme quantile estimator which features an automatic estimation and removal of all J first bias terms. Second, one as well as multi layer extensions of this NN estimator are introduced to tackle the conditional case, *i.e.* when the extreme quantiles depend on a covariate. To the best of our knowledge, this is the first attempt at reducing the estimation bias in conditional extreme quantiles.

This paper is organized as follows. An extrapolation principle for estimating extreme quantiles in the non-conditional heavy-tailed case is introduced in Section 3.2 with an emphasis on bias corrected estimators. The construction of the NN estimator is presented in Section 3.3 with its associated approximation error (Theorem 3.3.1). Similarly, an extrapolation principle for estimating conditional extreme quantiles is proposed in Section 3.4 and two conditional NN estimators are derived in Section 3.5 including their approximation properties (Theorem 3.5.1 and Theorem 3.5.2). The finite sample properties of the NN estimators are first illustrated on simulated data in the unconditional case (Section 3.6) where they are compared to extreme-value competitors. Second, the conditional NN estimators are tested on real data (Section 3.7) which consist in daily rainfall measurements among 524 stations in the southern part of France, see Figure 3.1. Proofs and algorithms are postponed to the Supplementary Material.

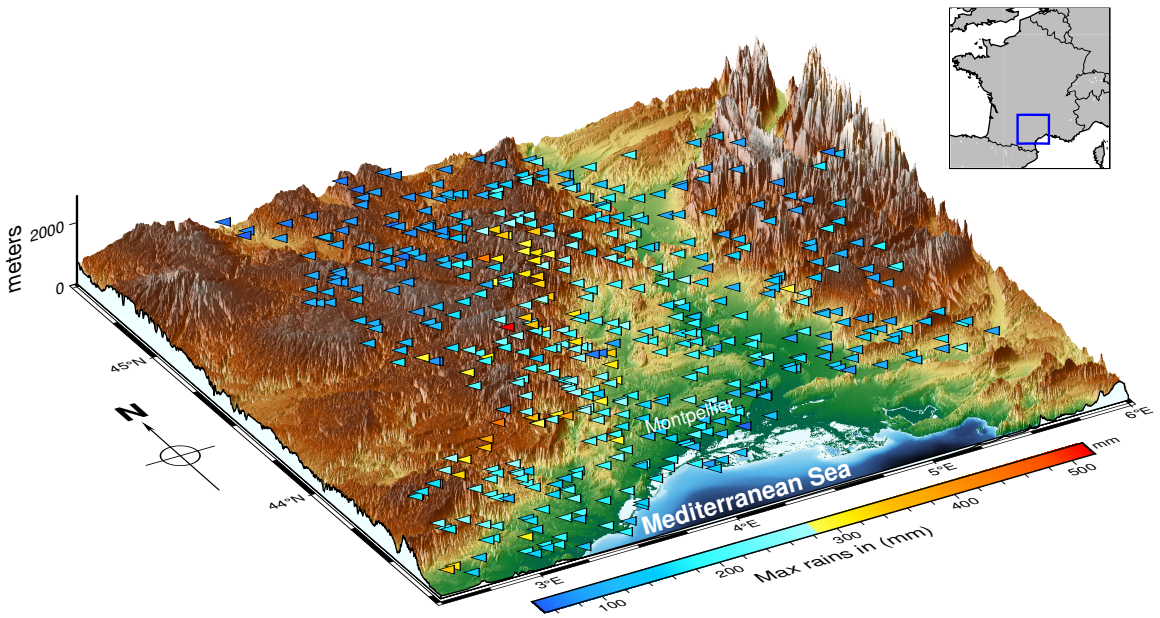


Figure 3.1: Historical (1958-2000) daily rainfall maxima in millimeters per station in the Cévennes-Vivarais region of France.

3.2 Extrapolation principle for extreme quantiles

Let $X_1, \ldots, X_n$ be an i.i.d sample from an unknown cumulative distribution function (c.d.f) F. The associated order statistics are denoted by $X_{1,n} \leq \cdots \leq X_{n,n}$. We are interested in the estimation of the quantile function defined by $q(\cdot) := F^{\leftarrow}(\cdot) = \inf\{x \in \mathbb{R} : F(x) \geq \cdot\}$, at the extreme level $1 - \alpha_n$ *i.e.* such that $n\alpha_n \to 0$ as $n \to \infty$. This latter condition entails that $q(1 - \alpha_n)$ is almost surely asymptotically larger than the sample maxima $X_{n,n}$.

Heavy-tailed distributions. Focusing on distributions in the Maximum domain of attraction of Fréchet, it is known from [52, Theorem 1.2.1] and [52, Proposition B.1.9.9] that the tail quantile function $U(t) := q(1 - 1/t)$ defined for all $t > 1$, is regularly-varying with index $\gamma > 0$ (this property is denoted by $U \in \mathcal{RV}_\gamma$ in the sequel) *i.e.* $U(t) = t^\gamma L(t)$ where γ is the so-called tail-index and $L \in \mathcal{RV}_0$ is a slowly-varying function at infinity *i.e.* L is positive and, for all $z > 0$,

$$\lim_{t\to\infty} \frac{L(tz)}{L(t)} = 1. \tag{3.2.1}$$

The index γ tunes the tail heaviness of F: the larger the index, the heavier the right tail. Examples include the (generalized) Pareto, Burr, Fisher, Inverse gamma and Student distributions, see Table 3.3 in the Supplementary Material for the associated tail indices. The idea underpinning the estimation is to take advantage of $U \in \mathcal{RV}_\gamma$ to establish a link between the extreme quantile of interest $q(1 - \alpha_n) = U(1/\alpha_n)$ and an intermediate one $q(1 - \delta_n) = U(1/\delta_n)$ where δ_n is interpreted as an anchor level such that $k := \lfloor n\delta_n \rfloor \to \infty$ as $n \to \infty$. To this end, introduce $x_1 = \log(z) \geq 0$, $x_2 = \log(t) \geq 0$ and the log-spacing function defined as

$$f(x_1, x_2) = \log U(\exp(x_1 + x_2)) - \log U(\exp(x_2)) = \gamma x_1 + \varphi(x_1, x_2), \tag{3.2.2}$$

with

$$\varphi(x_1, x_2) := \log L(\exp(x_1 + x_2)) - \log L(\exp(x_2)). \tag{3.2.3}$$

Considering $z = \delta_n/\alpha_n$ and $t = 1/\delta_n$ or equivalently $x_1 = \log(\delta_n/\alpha_n)$ and $x_2 = \log(1/\delta_n)$, it immediately follows that

$$q(1 - \alpha_n) = q(1 - \delta_n)\,(\delta_n/\alpha_n)^\gamma \exp\Big(\varphi\big(\log(\delta_n/\alpha_n), \log(1/\delta_n)\big)\Big).$$

An estimator of the extreme quantile can then be obtained using a two-step approach. First, a parametric model $\tilde{\varphi}_\theta$ is introduced for φ yielding the approximation of $q(1 - \alpha_n)$:

$$\tilde{q}_\phi(1 - \alpha_n; 1 - \delta_n) = q(1 - \delta_n)\,(\delta_n/\alpha_n)^\gamma \exp\Big(\tilde{\varphi}_\theta\big(\log(\delta_n/\alpha_n), \log(1/\delta_n)\big)\Big), \tag{3.2.4}$$

with parameter $\phi = (\gamma, \theta)$. Second, for a given level δ_n, estimate both $q(1-\delta_n)$ by the associated order statistic $X_{n-k+1,n}$ and ϕ by a dedicated estimator to get:

$$\hat{q}_{\hat{\phi}}(1 - \alpha_n; 1 - \delta_n) = X_{n-k+1,n}\,(\delta_n/\alpha_n)^{\hat{\gamma}} \exp\Big(\tilde{\varphi}_{\hat{\theta}}\big(\log(\delta_n/\alpha_n), \log(1/\delta_n)\big)\Big),$$

where $\hat{\phi} = (\hat{\gamma}, \hat{\theta})$. See Figure 3.2 (left panel) for an illustration of (3.2.2) associated with a Burr distribution and its pointwise estimation based on order statistics.

Weissman estimator. In this setting, the simplest method consists in choosing $\tilde{\varphi}_\theta = 0$ in (3.2.4), so that $\phi = \gamma$, to get the so-called Weissman estimator [184]:

$$\hat{q}^{\mathtt{W}}_{\hat{\phi}}(1 - \alpha_n; 1 - \delta_n) = X_{n-k+1,n}\,(\delta_n/\alpha_n)^{\hat{\gamma}^{\mathtt{H}}(k)}, \tag{3.2.5}$$

where $\hat{\gamma}^{\mathtt{H}}(\cdot)$ is the Hill estimator [108]. This approach relies on the approximation of the slowly-varying function L in (3.2.3) by a constant, which may not be precise enough in practice.

Bias corrected estimators. The above term (3.2.3) can be evaluated using the well-known second-order condition [93, Equation (13)] on the tail quantile function which states that there exist $\gamma > 0, \rho_2 \leq 0$ and a function A_2 positive or negative with $A_2(t) \to 0$ as $t \to \infty$ such that for all $z \geq 1$:

$$\log U(zt) - \log U(t) = \gamma \log z + A_2(t) \int_1^z z_2^{\rho_2 - 1} \, \mathrm{d}z_2 + o(A_2(t)), \quad \text{as } t \to \infty. \tag{3.2.6}$$

Moreover, $|A_2|$ is regularly-varying with index ρ_2. This second-order parameter drives the bias of most extreme quantile estimators: the larger ρ_2 is, the larger the asymptotic bias. Assumption (3.2.6) is standard in extreme-value theory, since it controls the rate of convergence in (3.2.1). Examples of distributions satisfying (3.2.6) can be found in [16, Section 3.3] and [52, Section 2.3], along with thorough discussions on this second-order condition. For instance, the (generalized) Pareto, Burr, Fréchet, Student, Fisher and Inverse-Gamma distributions all satisfy this condition, see Table 3.3. Ignoring the $o(\cdot)$ term in (3.2.6) and assuming

$$A_2(t) = \gamma \beta_2 t^{\rho_2}, \tag{3.2.7}$$

with $\rho_2 < 0$ and $\beta_2 \neq 0$, give rise to the parametric model defined for every $x_1, x_2 \geq 0$ by

$$\tilde{\varphi}_\theta^{\mathtt{CW}}(x_1, x_2) = \gamma \beta_2 \exp(\rho_2 x_2)[\exp(\rho_2 x_1) - 1]/\rho_2, \tag{3.2.8}$$

with $\theta = (\beta_2, \rho_2)$. Replacing in (3.2.4) yields the quantile approximation

$$\tilde{q}_\phi^{\mathtt{CW}}(1 - \alpha_n; 1 - \delta_n) = q(1 - \delta_n) \left(\frac{\delta_n}{\alpha_n} \right)^\gamma \exp\left(\gamma \beta_2 \left(\frac{1}{\delta_n} \right)^{\rho_2} \frac{(\delta_n/\alpha_n)^{\rho_2} - 1}{\rho_2} \right), \tag{3.2.9}$$

and the associated Corrected Weissman estimator introduced in [94],

$$\hat{q}_{\hat{\phi}}^{\mathtt{CW}}(1 - \alpha_n; 1 - \delta_n) = X_{n-k+1,n} \left(\frac{\delta_n}{\alpha_n} \right)^{\hat{\gamma}} \exp\left(\hat{\gamma} \hat{\beta}_2 \left(\frac{1}{\delta_n} \right)^{\hat{\rho}_2} \frac{(\delta_n/\alpha_n)^{\hat{\rho}_2} - 1}{\hat{\rho}_2} \right). \tag{3.2.10}$$

The quality of (3.2.10) hinges on a reliable estimation of γ and θ, see Section 3.6.2 for details. In the following, we propose an extension of (3.2.9) to an higher order approximation and an estimation of the associated parameters by a NN.

3.3 A neural network estimator of extreme quantiles

In a NN setting (see *i.e.* [99] for a general perspective), our purpose is to build an approximation of the log-spacing function (3.2.2) by taking advantage of higher order conditions on $U(\cdot)$ unlike classical bias-reduced estimators which are based on the second-order condition (3.2.6). We focus on the class of one-hidden layer (feedforward) NN:

$$x \in \mathbb{R} \mapsto \sum_{i=1}^{d} \nu_i^{(1)} \sigma^{\mathtt{E}} \left(\nu_i^{(2)} x + \nu_i^{(3)} \right) \in \mathbb{R}, \tag{3.3.1}$$

with parameters $\{(\nu_i^{(1)}, \nu_i^{(2)}, \nu_i^{(3)}),\ i \in \{1, \ldots, d\}\} \in \Theta \subset \mathbb{R}^{3d}$ where d is the number of neurons in the hidden layer and with eLU (exponential linear unit) activation functions: $\sigma^{\mathtt{E}}(x) = \exp(x) - 1$ if $x < 0$ and $\sigma^{\mathtt{E}}(x) = x$ otherwise. It is appealing that $\tilde{\varphi}_\theta^{\mathtt{CW}}$ in (3.2.8) can be rewritten using two eLU functions as

$$\tilde{\varphi}_\theta^{\mathtt{CW}}(x_1, x_2) = \frac{\gamma \beta_2}{\rho_2} \left(\sigma^{\mathtt{E}}(\rho_2(x_1 + x_2)) - \sigma^{\mathtt{E}}(\rho_2 x_2) \right). \tag{3.3.2}$$

In order to build higher order approximations of $\varphi(x_1, x_2)$ using more than two activation functions, we first consider a J-th order condition, introduced in [183] for all $J \geq 2$, on the tail

quantile function. Assume there exist $\gamma > 0$ and, for all $j \in \{2, \ldots, J\}$, $\rho_j \leq 0$ as well as positive or negative functions A_j with $A_j(t) \to 0$ as $t \to \infty$, $|A_j| \in \mathcal{RV}_{\rho_j}$, such that

$$\log U(tz) - \log U(t) = \gamma \log y + \sum_{j=2}^{J} \prod_{\ell=2}^{j} A_\ell(t) R_j(z) + o\left(\prod_{j=2}^{J} A_j(t) \right), \tag{3.3.3}$$

as $t \to \infty$ for all $z > 0$, where:

$$R_j(z) = \int_1^z z_2^{\rho_2 - 1} \int_1^{z_2} z_3^{\rho_3 - 1} \cdots \int_1^{z_{j-1}} z_j^{\rho_j - 1} \, \mathrm{d}z_j \ldots \mathrm{d}z_3 \, \mathrm{d}z_2, \tag{3.3.4}$$

and similarly to (3.2.7), suppose

$$A_j(t) = c_j t^{\rho_j}, \tag{3.3.5}$$

where $c_j \neq 0$ and $\rho_j < 0$ for $j \in \{2, \ldots, J\}$. Clearly, when $J = 2$, we recover the second-order condition (3.2.6). Moreover, $J = 3$ and $J = 4$ yield back respectively the third-order [72, 147] and fourth-order conditions [92]. In the following, we show that the key quantity to quantify the approximation convergence rate is $\bar{\rho}_J := \rho_2 + \cdots + \rho_J$. Second, we introduce a NN approximation of $\varphi(x_1, x_2)$ using $J(J-1)$ eLU functions which is a very natural extension of (3.3.2):

$$\tilde{\varphi}_\theta^{\mathtt{NN}_J}(x_1, x_2) := \sum_{i=1}^{J(J-1)/2} w_i^{(1)} \left(\sigma^{\mathtt{E}}\left(w_i^{(2)} x_1 + w_i^{(3)} x_2 \right) - \sigma^{\mathtt{E}}(w_i^{(4)} x_2) \right), \tag{3.3.6}$$

for some $\theta = \left\{ (w_i^{(1)}, w_i^{(2)}, w_i^{(3)}, w_i^{(4)}), i \in \{1, \ldots, J(J-1)/2\} \right\} \in \Theta := (\mathbb{R} \times \mathbb{R}_-^3)^{J(J-1)/2}$. Lemma 3.B.2 in the Supplementary Material states that there exists a one-hidden layer eLU NN approximation $\tilde{\varphi}_{\tilde{\theta}}^{\mathtt{NN}_J}$ of φ with the same representation as in (3.3.6), parameterized by some unknown $\tilde{\theta} \in \Theta$, and with a controlled error. Note that such a result is not a direct consequence of the Universal approximation theorem [46] which ensures that a continuous function can be uniformly approximated on a compact set with arbitrary precision by a one hidden layer NN. Indeed, φ does not have a compact support and, moreover, the extrapolation framework makes necessary to control the approximation of $\varphi(x_1, x_2)$ when both x_1 and x_2 tend to infinity. Recall that the parametric model (3.3.6) encompasses (3.3.2) as a particular case when $J = 2$. Third, for all $\tilde{\phi} = (\tilde{w}_0, \tilde{\theta}) \in \Phi := \mathbb{R}_+ \times \Theta$, consider the NN approximation of the log-spacing function

$$\tilde{f}_{\tilde{\phi}}^{\mathtt{NN}_J}(x_1, x_2) = \tilde{w}_0 x_1 + \tilde{\varphi}_{\tilde{\theta}}^{\mathtt{NN}_J}(x_1, x_2), \tag{3.3.7}$$

and, combining (3.2.4) with (3.3.7), the NN approximation of the extreme quantile is defined as

$$\tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1 - \alpha_n; 1 - \delta_n) := q(1 - \delta_n) \exp\left(\tilde{f}_{\tilde{\phi}}^{\mathtt{NN}_J}(\log(\delta_n/\alpha_n), \log(1/\delta_n)) \right). \tag{3.3.8}$$

The approximating NN includes $d = J(J-1)$ neurons and CJ^2 parameters, where throughout $C > 0$ is a constant independent of J.

Theorem 3.3.1. *Assume the J-th order condition (3.3.3) holds together with (3.3.5) for some $J \geq 2$. Then, their exists a one hidden-layer NN approximation (3.3.8) of the extreme quantile $q(1 - \alpha_n)$ such that*

$$\inf_{\tilde{\phi} \in \Phi} \left| \log q(1 - \alpha_n) - \log \tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1 - \alpha_n; 1 - \delta_n) \right| = \mathcal{O}\left(\alpha_n^{-\bar{\rho}_J} \right), \tag{3.3.9}$$

with $\alpha_n \to 0$ and $\delta_n / \alpha_n \to \infty$ when $n \to \infty$.

In view of (3.3.9), the error between the extreme log-quantile and its NN approximation is driven by $\bar{\rho}_J$. As expected, requesting higher regularity in the extreme-value model (through the J-th order condition) yields a smaller approximation error thanks to an increasing width of the proposed NN. Finally, we are in a position to define the NN extreme quantile estimator

$$\hat{q}^{\mathtt{NN}_J}_{\hat{\phi}}(1-\alpha_n;1-\delta_n) := X_{n-k+1,n}\,\exp\left(\hat{f}^{\mathtt{NN}_J}_{\hat{\phi}}\left(\log\left(\delta_n/\alpha_n\right),\log\left(1/\delta_n\right)\right)\right),$$

where the estimated parameters $\hat{\phi} \in \Phi$ are computed thanks to the optimization process described in Section 2.3.

3.4 Extrapolation for conditional extreme quantiles

Suppose now that X is a random variable associated with an explanatory random vector $Y \in \mathcal{Y} \subset \mathbb{R}^{d_y}, d_y \geq 1$, where $\mathcal{Y}$ is assumed to be a compact set in the sequel. Denoting by $F(\cdot \,|\, y)$ the conditional c.d.f of X given $\{Y = y\}$ for some $y \in \mathcal{Y}$, the conditional quantile function is defined by $q(\cdot \,|\, y) = \inf\{x \in \mathbb{R} : F(x \,|\, y) \geq \cdot\}$ and the quantile function is defined for all $t \geq 1$ by $U(t|y) := q(1-1/t|y)$. The usual unconditional extrapolation principles can be extended to this new setting basing on maximum domain attraction assumptions [48, 50, 82]. More specifically, when the conditional distribution of X given $\{Y = y\}$ is assumed to be heavy-tailed, which is our framework in the sequel, the Weissman estimator (3.2.5) can be adapted as follows:

$$\hat{q}^{\mathtt{W}}_{\hat{\phi}}(1-\alpha_n;1-\delta_n \,|\, y) = \hat{q}(1-\delta_n \,|\, y)\,(\delta_n/\alpha_n)^{\hat{\gamma}(y)},$$

see [49]. The above conditional Weissman estimator relies on two quantities: $\hat{q}(1-\delta_n \,|\, y)$ which is an estimator of the intermediate conditional quantile $q(1-\delta_n \,|\, y)$ and $\hat{\gamma}(y)$, an estimator of the conditional tail-index $\gamma(y)$.

Estimation of intermediate conditional quantiles. In the conditional framework, the additional challenge is that conditional intermediate quantiles can no longer be estimated by order statistics. Among the numerous methods dedicated to the estimation of conditional quantiles, two main lines of works can be identified. On the first hand, direct methods characterize the conditional quantile of level $\alpha \in (0,1)$ as $q(1-\alpha \,|\, y) = \arg\min_{\tau\in\mathbb{R}} \mathbb{E}\left[\check{\rho}_{1-\alpha}\left(X-\tau\right) \,|\, y\right]$, where $v \in \mathbb{R} \mapsto \check{\rho}_{1-\alpha}(v) := v\left(1-\alpha-\mathbb{1}_{(-\infty,0]}\{v\}\right)$ is the so-called check-function. Estimators of the conditional quantile are then obtained by replacing the conditional expectation by some nonparametric estimator and solving the associated optimization problem, see among others [106] for spline based methods and [192] for kernel smoothing techniques. On the other hand, the indirect method consists in first estimating the conditional c.d.f $F(\cdot \,|\, y)$, and then compute the quantile via numerical inversion. Nonparametric estimators of $F(\cdot \,|\, y)$ include kernel estimators [174] and nearest neighbor estimators [23].

Estimation of the conditional tail-index. Moving windows and nearest neighbors approaches have been developed in a fixed design setting [78, 80]. Kernel methods are proposed in [49, 83] to tackle the random design case.

In the next section, we show how to combine an indirect method to estimate the intermediate quantile (the nearest neighbor estimator, see Section 3.7) with a NN to estimate conditional extrapolation schemes following the ideas of Section 3.3. We also refer to [67, Section 3.5] for the approximation of the nearest neighbors distribution using the Hellinger distance and to [74] for the investigation of their asymptotic properties. Other indirect estimators of conditional extreme quantiles using nearest neighbor techniques are investigated in [80, 82] while direct estimators of conditional extreme quantiles are proposed in [181, 182].

3.5 NN estimators of conditional extreme quantiles

We present two approaches to estimate conditional extreme quantiles by a NN. The first one is the conditional extension of the model presented in Section 3.3. The second one takes advantage of a location-dispersion model to get rid of the covariate in the extrapolation.

3.5.1 Conditional Extrapolation Neural Networks (CENN)

As in the unconditional situation, the tail quantile function $U(\cdot \mid y)$ is assumed to be regularly-varying with a conditional tail-index $\gamma(y) > 0$ *i.e.* $U(t \mid y) = t^{\gamma(y)} L(t \mid y)$, where $L(\cdot \mid y) \in \mathcal{RV}_0$. Similarly to (3.2.2), the conditional log-spacings function is defined by

$$f(x_1, x_2 \mid y) = \log U(\exp(x_1 + x_2) \mid y) - \log U(\exp(x_2) \mid y) = \gamma(y) x_1 + \varphi(x_1, x_2 \mid y),$$

with $\varphi(x_1, x_2 \mid y) := \log L(\exp(x_1 + x_2) \mid y) - \log L(\exp(x_2) \mid y)$ for $(x_1, x_2, y) \in \mathbb{R}_+^2 \times \mathbb{R}^{d_y}$ and

$$q(1 - \alpha_n; 1 - \delta_n \mid y) = q(1 - \delta_n \mid y) \, (\delta_n / \alpha_n)^{\gamma(y)} \exp\Big(\varphi\big(\log(\delta_n/\alpha_n), \log(1/\delta_n) \mid y\big)\Big). \tag{3.5.1}$$

The same methodology as in Section 3.2 is applied here, where the conditional extension of the J-th order condition (3.3.3) is

$$\log U(tz \mid y) - \log U(t \mid y) = \gamma(y) \log z + \sum_{j=2}^{J} \prod_{\ell=2}^{j} A_\ell(t \mid y) R_j(z \mid y) + o\left(\prod_{j=2}^{J} A_j(t \mid y) \right), \tag{3.5.2}$$

as $t \to \infty$ for $z > 0$, with for all $j \in \{2, \ldots, J\}$,

$$R_j(z \mid y) = \int_1^z z_2^{\rho_2(y)-1} \int_1^{z_2} z_3^{\rho_3(y)-1} \cdots \int_1^{z_{j-1}} z_j^{\rho_j(y)-1} \, \mathrm{d}z_j \ldots \mathrm{d}z_3 \, \mathrm{d}z_2,$$

and

$$A_j(t \mid y) = c_j(y) t^{\rho_j(y)}, \tag{3.5.3}$$

Since $w_i^{(1)}, w_i^{(2)}, w_i^{(3)}, w_i^{(4)}, i \in \{1, \ldots, J(J-1)/2\}$ and γ depend now on the covariate, the idea is to replace in (3.3.6) and (3.3.7) each parameter by a NN to approximate the conditional quantity. Hence, we consider $\tilde{f}_{\tilde{\phi}}^{\mathtt{NN}_J}(x_1, x_2 \mid y) = \tilde{w}_{\tilde{\theta}^{(0)}}^{\mathtt{NN}_J}(y) x_1 + \tilde{\varphi}_{\tilde{\theta}}^{\mathtt{NN}_J}(x_1, x_2 \mid y)$, with

$$\tilde{\varphi}_{\tilde{\theta}}^{\mathtt{NN}_J}(x_1, x_2 \mid y) := \sum_{i=1}^{J(J-1)/2} \tilde{w}_{\tilde{\theta}_i^{(1)}}^{\mathtt{NN}}(y) \left(\sigma^{\mathtt{E}}\left(\tilde{w}_{\tilde{\theta}_i^{(2)}}^{\mathtt{NN}}(y) x_1 + \tilde{w}_{\tilde{\theta}_i^{(3)}}^{\mathtt{NN}}(y) x_2 \right) - \sigma^{\mathtt{E}}\left(\tilde{w}_{\tilde{\theta}_i^{(4)}}^{\mathtt{NN}}(y) x_2 \right) \right),$$

where, for all $j \in \{1, \ldots, 4\}$ and $i \in \{1, \ldots, J(J-1)/2\}$, $\tilde{w}_{\tilde{\theta}^{(0)}}^{\mathtt{NN}_J}$ and $\tilde{w}_{\tilde{\theta}_i^{(j)}}^{\mathtt{NN}}$ are $2J(J-1)+1$ deep ReLU NNs with respectively $d^{(0)}$ and $d^{(j)}$ neurons in each of the $p^{(0)}$ and $p^{(j)}$ hidden layers. Recall that the ReLU activation function is defined by $x \in \mathbb{R} \mapsto \sigma(x) = \max(x, 0)$. Unlike (3.3.1), we apply $-\sigma(\cdot)$ in the output layer of $\tilde{w}_{\tilde{\theta}_i^{(2)}}^{\mathtt{NN}}, \tilde{w}_{\tilde{\theta}_i^{(3)}}^{\mathtt{NN}}$ and $\tilde{w}_{\tilde{\theta}_i^{(4)}}^{\mathtt{NN}}$ in order to force a negative output. This choice of a ReLU NN is motivated theoretically by [190] but is not essential from a numerical point of view. Thus, from (3.5.1), one can build an approximation of the conditional extreme quantile $q(1 - \alpha_n \mid y)$:

$$\tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1 - \alpha_n; 1 - \delta_n \mid y) = q(1 - \delta_n \mid y) \exp\left(\tilde{f}_{\tilde{\phi}}^{\mathtt{NN}_J}\left(\log(\delta_n/\alpha_n), \log(1/\delta_n) \mid y \right) \right). \tag{3.5.4}$$

The approximating NN includes $J(J-1)/2 \left(\sum_{j=1}^{4} p^{(j)} d^{(j)} + 2 \right) + d^{(0)}$ neurons and $C \, J^2 \sum_{j=1}^{4} p^{(j)} (d^{(j)})^2$ parameters. As a consequence of Lemma 3.B.2 in the Supplementary Material, the following result on the NN approximation of the extreme quantile holds:

Theorem 3.5.1. *Assume the J-th order condition (3.5.2) holds together with (3.5.3) for some $J \geq 2$. Additionally, suppose $w_i^{(1)}(\cdot), w_i^{(2)}(\cdot), w_i^{(3)}(\cdot), w_i^{(4)}(\cdot), i = 1, \ldots, J(J-1)/2$, and $\gamma(\cdot)$ are continuous on $\mathcal{Y} \subset \mathbb{R}^{d_y}$. Let $\bar{\rho}_{\texttt{sup}} = \sup_{y \in \mathcal{Y}} \bar{\rho}_J(y)$ with $\bar{\rho}_J(y) = \rho_2(y) + \cdots + \rho_J(y)$. Then, there exists a deep NN approximation (3.5.4) of the conditional extreme quantile $q(1-\alpha_n \,|\, \cdot)$ including $2J(J-1)+1$ sub-networks built for all $j \in \{0, \ldots, 4\}$ with fixed $d^{(j)} = 2d_y + 10$ number of neurons in each of the hidden layers of depths*

$$
\begin{aligned}
p_n^{(0)} = p_n^{(2)} &> c\, \alpha_n^{\bar{\rho}_{\texttt{sup}}/2} \left(\log(\delta_n/\alpha_n)\right)^{1/2}, \\
p_n^{(1)} &> c\, \alpha_n^{\bar{\rho}_{\texttt{sup}}/2}, \\
p_n^{(3)} = p_n^{(4)} &> c\, \alpha_n^{\bar{\rho}_{\texttt{sup}}/2} \left(\log(1/\delta_n)\right)^{1/2},
\end{aligned}
$$

where $c > 0$ is an arbitrary constant, $\alpha_n \to 0$ and $\delta_n/\alpha_n \to \infty$ as $n \to \infty$ such that

$$
\inf_{\tilde{\phi} \in \Phi} \sup_{y \in \mathcal{Y}} \left| \log q(1-\alpha_n \,|\, y) - \log \tilde{q}_{\tilde{\phi}}^{\texttt{NN}_J}(1-\alpha_n; 1-\delta_n \,|\, y) \right| = \mathcal{O}\left(\alpha_n^{-\bar{\rho}_{\texttt{sup}}}\right).
$$

In this general conditional setting, a minimum depth (of magnitude $\simeq \alpha_n^{\bar{\rho}_{\texttt{sup}}/2}$) is required for the CENN to approximate the extreme quantile with a given error (of order $\simeq \alpha_n^{-\bar{\rho}_{\texttt{sup}}}$) while, in the previous situation, a one layer NN was sufficient. We are in a position to define the conditional NN extrapolation quantile estimator

$$
\hat{q}_{\hat{\phi}}^{\texttt{NN}_J}(1-\alpha_n; 1-\delta_n \,|\, y) := \hat{q}(1-\delta_n \,|\, y)\, \exp\left(\hat{f}_{\hat{\phi}}^{\texttt{NN}_J}\left(\log\left(\delta_n/\alpha_n\right), \log\left(1/\delta_n\right) \,|\, y\right)\right),
$$

where $\hat{q}(1-\delta_n \,|\, y)$ is an estimator of the intermediate conditional quantile.

3.5.2 Location-Dispersion Neural Networks (LDNN)

The location-dispersion regression model introduced in [177] assumes that

$$
X = a(Y) + b(Y)Z, \tag{3.5.5}
$$

where $a : \mathcal{Y} \to \mathbb{R}$ and $b : \mathcal{Y} \to \mathbb{R}^+$ are defined respectively as the regression and the dispersion functions while $Z \in \mathbb{R}$ is a real random variable. Denoting by $q_Z(\cdot)$ and $U_Z(\cdot)$ respectively the quantile and tail quantile functions of Z, it follows from (3.5.5) that $U(t \,|\, y) = a(y) + b(y)U_Z(t)$, or equivalently $q(1-\alpha_n \,|\, y) = a(y) + b(y)q_Z(1-\alpha_n)$ and, therefore, considering three levels of quantiles $0 < \alpha_n < \delta_n < \tau_n < 1$ yields

$$
\begin{aligned}
\frac{q(1-\alpha_n \,|\, y) - q(1-\delta_n \,|\, y)}{q(1-\delta_n \,|\, y) - q(1-\tau_n \,|\, y)} = \frac{\frac{U_Z(1/\alpha_n)}{U_Z(1/\delta_n)} - 1}{1 - \frac{U_Z(1/\tau_n)}{U_Z(1/\delta_n)}} &= \frac{\exp\left(f_Z(\log(\delta_n/\alpha_n),\, \log(1/\delta_n))\right) - 1}{1 - \exp\left(f_Z(\log(\delta_n/\tau_n), \log(1/\delta_n))\right)} \\
&=: g\big(\log(\delta_n/\alpha_n), \log(1/\delta_n), \log(\delta_n/\tau_n)\big),
\end{aligned}
$$

where f_Z is defined similarly to (3.2.2) by $f_Z(x_1, x_2) = \log U_Z(\exp(x_1 + x_2)) - \log U_Z(\exp(x_2))$, and $g(x_1, x_2, x_3) = \big(\exp(f_Z(x_1, x_2)) - 1\big)/\big(1 - \exp(f_Z(x_3, x_2))\big)$ for all $(x_1, x_2, x_3) \in \mathbb{R}_+^3$. Let us stress that f_Z and g do not depend on the covariate. Rearranging the terms yields

$$
q(1-\alpha_n \,|\, y) = q(1-\delta_n \,|\, y) \left(1 + \left(1 - \frac{q(1-\tau_n \,|\, y)}{q(1-\delta_n \,|\, y)}\right) g(\log(\delta_n/\alpha_n), \log(1/\delta_n), \log(\delta_n/\tau_n))\right).
$$

One can then approximate conditional extreme quantiles from the location-dispersion regression model when Z is assumed to be heavy-tailed, *i.e.* $U_Z(t) = t^\gamma L_Z(t)$ with $\gamma > 0$ and $L_Z \in \mathcal{RV}_0$. It straightforwardly follows that $U(\cdot \,|\, y) \in \mathcal{RV}_\gamma$ meaning that X given $\{Y = y\}$ is heavy-tailed with tail-index independent of the covariate [3]. In other words, X inherits its tail behavior

from Z which does not depend on y. Let us also note that the regular variation property yields $U(t \mid y)/U_Z(t) \to b(y)$ as $t \to \infty$. The location-dispersion regression model (3.5.5) can thus be interpreted as a particular case of the proportional tails model [61] for heteroscedastic extremes, see [53, 77, 88] for alternative solutions.

Following the methodology introduced in the unconditional case (see Section 3.3), the conditional extreme quantile $q(1-\alpha_n \mid y)$ can be approximated using two intermediate conditional quantiles $q(1-\delta_n \mid y)$ and $q(1-\tau_n \mid y)$:

$$\tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1-\alpha_n; 1-\delta_n, 1-\tau_n \mid y) = \\ q(1-\delta_n \mid y)\left(1+\left(1-\frac{q(1-\tau_n \mid y)}{q(1-\delta_n \mid y)}\right)\tilde{g}_{\tilde{\phi}}^{\mathtt{NN}_J}\big(\log(\delta_n/\alpha_n), \log(1/\delta_n), \log(\delta_n/\tau_n)\big)\right), \tag{3.5.6}$$

with

$$\tilde{g}_{\tilde{\phi}}^{\mathtt{NN}_J}\big(\log(\delta_n/\alpha_n), \log(1/\delta_n), \log(\delta_n/\tau_n)\big) = \frac{\exp\left(\tilde{f}_{\tilde{\phi}}^{\mathtt{NN}_J}(\log(\delta_n/\alpha_n),\, \log(1/\delta_n))\right) - 1}{1 - \exp\left(\tilde{f}_{\tilde{\phi}}^{\mathtt{NN}_J}(\log(\delta_n/\tau_n),\, \log(1/\delta_n))\right)},$$

and where $\tilde{f}_{\tilde{\phi}}^{\mathtt{NN}_J}$ is defined in (3.3.6) and (3.3.7). The approximating NN includes $C\,J^2$ parameters and $d = J(J-1)/2$ neurons. We can thus extend Theorem 3.3.1 to the conditional framework.

Theorem 3.5.2. *Assume (3.5.5) and conditions of Theorem 3.3.1 hold for U_Z. Suppose $a(\cdot)$ and $b(\cdot)$ are continuous functions on $\mathcal{Y}$ and that $b(\cdot)$ is bounded from below by a positive constant. Then, there exists a one hidden-layer neural network approximation (3.5.6) of the conditional extreme quantile $q(1-\alpha_n \mid \cdot)$ such that*

$$\inf_{\tilde{\phi}\in\Phi} \sup_{y\in\mathcal{Y}} \left|\log q(1-\alpha_n \mid y) - \log \tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1-\alpha_n; 1-\delta_n, 1-\tau_n \mid y)\right| = \mathcal{O}(\alpha_n^{-\bar{\rho}_J}) + \mathcal{O}(\tau_n^{-\bar{\rho}_J-\gamma}\delta_n^{\gamma})$$

with $\alpha_n \to 0$, $\delta_n/\tau_n \to 0$ and $\delta_n/\alpha_n \to \infty$ as $n \to \infty$.

It is then possible to tune the sequences δ_n and τ_n to balance both error terms:

Corollary 3.5.1. *Assume the assumptions of Theorem 3.5.2 hold.*

- *If $\gamma + \bar{\rho}_J > 0$, then letting $\delta_n = \alpha_n^{-\bar{\rho}_J/\gamma}\tau_n^{1+\bar{\rho}_J/\gamma}$ yields*

$$\inf_{\tilde{\phi}\in\Phi} \sup_{y\in\mathcal{Y}} \left|\log q(1-\alpha_n \mid y) - \log \tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1-\alpha_n; 1-\delta_n, 1-\tau_n \mid y)\right| = \mathcal{O}\left(\alpha_n^{-\bar{\rho}_J}\right).$$

- *If $\gamma + \bar{\rho}_J \le 0$, then letting $\delta_n = \xi_n\alpha_n$ and $\tau_n = \xi_n^2\alpha_n$ with $\xi_n \to \infty$ arbitrarily slowly as $n \to \infty$ yields*

$$\inf_{\tilde{\phi}\in\Phi} \sup_{y\in\mathcal{Y}} \left|\log q(1-\alpha_n \mid y) - \log \tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1-\alpha_n; 1-\delta_n, 1-\tau_n \mid y)\right| = \mathcal{O}\left(\alpha_n^{-\bar{\rho}_J}\xi_n^{-2\bar{\rho}_J-\gamma}\right).$$

Up to the ξ_n term, one can recover the convergence rate $\alpha_n^{-\bar{\rho}_J}$ of the unconditional case, see Theorem 3.3.1. The conditional NN extreme quantile estimator is defined as

$$\hat{q}_{\hat{\phi}}^{\mathtt{NN}_J}(1-\alpha_n; 1-\delta_n, 1-\tau_n \mid y) = \\ \hat{q}(1-\delta_n \mid y) + (\hat{q}(1-\delta_n \mid y) - \hat{q}(1-\tau_n \mid y))\,\tilde{g}_{\hat{\phi}}^{\mathtt{NN}_J}\big(\log(\delta_n/\alpha_n), \log(1/\delta_n), \log(\delta_n/\tau_n)\big),$$

where, again, the intermediate conditional quantiles $q(1-\delta_n \mid y)$ and $q(1-\tau_n \mid y)$ can be estimated using the nearest neighbor estimator, see Section 3.7 for an illustration.

3.6 Validation on simulated data (unconditional case)

The finite sample behaviour of the (unconditional) extreme quantile NN estimator is illustrated on simulated data. To this end, we first describe both the estimator implementation and the model selection technique. Then, we briefly present some other bias-reduced estimators taken from the extreme-value literature. Next, we list the heavy-tailed distributions as well as the performance criteria used to compare all considered estimators.

3.6.1 Implementation of the NN estimator of extreme quantiles

The NN model $\tilde{f}^{\mathtt{NN}_J}_{\tilde{\phi}}$ of the log-spacing function (see Section 3.3) is fitted to the data by minimizing some distance between two estimations of the $N = (n-1)(n-2)/2$ log-spacings:

$$\hat{\phi} = \arg\min_{\tilde{\phi}\in\Phi} \frac{1}{N} \sum_{k=2}^{n-1} \sum_{i=1}^{k-1} \left| \hat{S}_{i,k} - \tilde{f}^{\mathtt{NN}_J}_{\tilde{\phi}}(\log(k/i), \log(n/k)) \right|^s, \ s \in \{1,2\}, \tag{3.6.1}$$

where, for $i \in \{1,\dots,k-1\}$ and $k \in \{2,\dots,n-1\}$, $\hat{S}_{i,k} := \log(X_{n-i+1,n}) - \log(X_{n-k+1,n})$ is the empirical estimate of $\log q(1-i/n) - \log q(1-k/n)$.

All experiments have been conducted on the Cholesky computing cluster from Ecole Polytechnique `http://meso-ipp.gitlab.labos.polytechnique.fr/user_doc`. It is composed by 4 nodes, where each one includes 2 CPU Intel Xeon Gold 6230 @ 2.1GHz, 20 cores and 4 Nvidia Tesla v100 graphics card. The code was implemented in Python 3.8.2 and using the library PyTorch 1.7.1. We used the optimizer Adam [114] with default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for all tests performed during $M = 500$ iterations. Additionally, the ranges of the NN hyperparameters explored to find the best model are reported in Table 3.1. See Figure 3.2 (right panel) for an illustration of the NN estimation of the log-spacing function on Burr data, compared with the crude linear Weissman approximation.

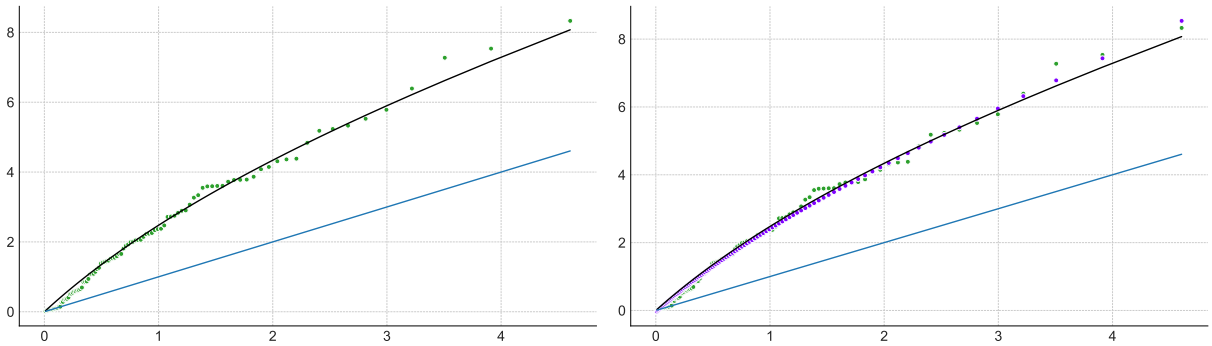


Figure 3.2: Log-spacings associated with a Burr distribution ($\gamma = 1, \rho = -1/4$). Black curve: theoretical function $x_1 \mapsto f(x_1, \log(n/k))$, blue line: Weissman approximation $x_1 \mapsto \gamma x_1$, green dots: empirical pointwise estimation $(\log(k/i), \log X_{n-i+1,n} - \log X_{n-k+1,n})$, purple dots: NN estimation $\tilde{f}^{\mathtt{NN}_J}_{\hat{\phi}}(\log(k/i), \log(n/k))$ with $i \in \{1,\dots,k-1\}$, $k = 100$ and $n = 500$.

Setting	J	batch size	loss function
Non-conditional	$\{2,3,4,5\}$	$\{256, 512, 1024\}$	$s \in \{1,2\}$
Conditional	$\{2,3,4,5\}$	$\{256, 512, 1024\}]$	$s = 1$

Table 3.1: Hyperparameters ranges used for tuning NNs across the experiments.

Model selection Algorithm 1 selects the parameters $\hat{\phi} = \hat{\phi}_{m^\star}(\mathcal{A}^\star)$ associated with the best architecture $\mathcal{A}^\star$ in Table 3.1 and iteration $m^\star \in \{1, \ldots, n_M\}$ corresponding to the smallest median absolute deviation $\texttt{MAD}\left(\left\{\hat{q}^{\texttt{NN}_J}_{\hat{\phi}_m}\left(1-\alpha_n; 1-\frac{k}{n}\right),\ k \in \{k_1, \ldots, k_2\}\right\}\right)$ where, for any set $\mathcal{E} \subset \mathbb{R}$, the mediane absolute deviation is defined as $\texttt{MAD}(\mathcal{E}) = \underset{\mathfrak{e}\in\mathcal{E}}{\text{median}}\,|\mathfrak{e} - \text{median}(\mathcal{E})|$. In all the experiments, we used $k_1 = [3n/100]$ and $k_2 = [3n/4]$.

3.6.2 Competitors

Seven bias reduced extreme quantile estimators are considered. They can be sorted in two main families. First, one can plug a bias-reduced estimator of the tail-index γ in the Weissman estimator (3.2.5). As an example, basing on the second-order condition (3.2.6) and (3.2.7), the Corrected-Hill (CH) estimator $\hat{\gamma}^{\texttt{CH}}$ is proposed in [31]. Similarly, [96] and [95] introduced a tuning parameter $p \geq 0$ in $\hat{\gamma}^{\texttt{CH}}$ to get respectively the reduced-bias mean-of-order-p and partially reduced-biais estimators denoted by $\hat{\gamma}^{\texttt{CH}_p}$ and $\hat{\gamma}^{\texttt{PRB}_p}$. To select p, one can either follow a path stability criterion [96, Algorithm 4.2] or plug an "optimal" deterministic value [96, p. 1739], denoted by $p^\star$ leading to the estimators $\hat{\gamma}^{\texttt{CH}_{p^\star}}$ and $\hat{\gamma}^{\texttt{PRB}_{p^\star}}$. Second, one may reduce simultaneously the extrapolation bias and the bias coming from the estimation of the tail-index. This idea is implemented in the Corrected Weissman estimator (CW), see (3.2.10). More recently, a Refined Weissman (RW) estimator has been proposed in [4] featuring an adapted choice of the intermediate sequence $k^{\texttt{H}}$ in the Hill estimator (different from the one used in the intermediate quantile estimator). Replacing the Hill estimator in (3.2.5) by $\hat{\gamma}^{\texttt{CH}}(k), \hat{\gamma}^{\texttt{CH}_p}(k), \hat{\gamma}^{\texttt{PRB}_p}(k), \hat{\gamma}^{\texttt{CH}_{p^\star}}(k), \hat{\gamma}^{\texttt{PRB}_{p^\star}}(k)$ and $\hat{\gamma}^{\texttt{H}}(k^{\texttt{H}})$ leads respectively to the estimators $\hat{q}^{\texttt{CH}}_{\hat{\phi}}, \hat{q}^{\texttt{CH}_p}_{\hat{\phi}}, \hat{q}^{\texttt{PRB}_p}_{\hat{\phi}}, \hat{q}^{\texttt{CH}_{p^\star}}_{\hat{\phi}}, \hat{q}^{\texttt{PRB}_{p^\star}}_{\hat{\phi}}$ and $\hat{q}^{\texttt{RW}}_{\hat{\phi}}$. See [4] for a detailed account on these bias-reduced extreme quantile estimators.

3.6.3 Experimental design

The comparative study is achieved on six heavy-tailed distributions. The first five distributions: Burr, Fréchet, Fisher, generalized Pareto distribution (GPD), Inverse Gamma, and Student belong to the Hall-Welsh class [104] which assumes that there exist $c_1 > 0, c_2 \neq 0$ such that $U(t) = c_1 t^\gamma\,(1 + c_2 t^{\rho_2} + o(t^{\rho_2}))$. These five distributions satisfy the second-order condition (3.2.6) with (3.2.7), see Table 3.3 for their definitions and associated values of γ and ρ_2. The sixth distribution, denoted by $\text{NHW}(\gamma, \rho_2)$, is defined for all $\gamma \geq \exp(-2)/2$ and $\rho_2 < 0$ by its tail quantile function $U(t) = t^\gamma \exp(A_2(t)/\rho_2)$ where $A_2(t) = \rho_2 t^{\rho_2} \log(t)/2$, $t \geq 1$, is the auxiliary function associated with the second-order condition (3.2.6). It thus appears that the NHW distribution does not belong to the Hall-Welsh class and does not verify (3.2.7) either. Based on the simulation study of [4], we focus on the following challenging situations for extreme-value estimators, large values of γ and/or large values of ρ_2: Burr distribution ($\gamma \in \{1/8, 1/4, 1/2\}$, $\rho_2 = -1/8$), NHW distribution ($\gamma = 1$, $\rho_2 \in \{-1/8, -1/4, -1/2, -1, -2\}$), Fisher distribution ($\nu_1 = 1$, $\nu_2 \in \{2, 16\}$) leading to $(\gamma, \rho_2) \in \{1/8, 1\} \times \{-1/8, -1\}$, GPD ($\gamma = 1/8$) leading to $\rho_2 = -1/8$, Inverse Gamma distribution ($\zeta = 1$) leading to $\gamma = 1$ and $\rho_2 = -1$, and Student distribution ($\nu = 1$) leading to $\gamma = 1$ and $\rho_2 = -2$. For each of these 21 considered configurations, $n_R = 500$ replicated data sets of size $n = 500$ are simulated and the extreme quantile of order $1 - \alpha_n = 1 - 1/(2n)$ is estimated using the NN estimator, the Weissman estimator and the seven bias-reduced estimators described in the above paragraph. Their performance is assessed using the Relative median-squared error (RMedSE):

$$\text{RMedSE}\left(\hat{q}_{\hat{\phi}}, \frac{1}{2n}\right) = \underset{r\in\{1,\ldots,n_R\}}{\text{median}}\left[\left(\frac{\hat{q}^{(r)}_{\hat{\phi}}(1-\frac{1}{2n}; 1-\frac{k^\star(r)}{n})}{q(1-\frac{1}{2n})} - 1\right)^2\right], \tag{3.6.2}$$

where $\hat{q}^{(r)}_{\hat{\phi}}(1-\frac{1}{2n}; 1-\frac{k^\star(r)}{n})$ denotes an estimator of $q(1-\frac{1}{2n})$ (either the NN one or some of its competitors) computed with the anchor index $k^\star(r)$ selected using [4, Algorithm 1] with initial

points $a^{(0)} = [3n/100]$ and $c^{(0)} = [3n/4]$ on the rth replication, $r \in \{1, \ldots, n_R\}$.

3.6.4 Results

It appears from the results in Table 3.2 that the NN approach is an efficient tool for estimating extreme quantiles in difficult heavy-tailed situations where other estimators almost all fail. The NN estimator indeed provides the best results in 12 out of 21 times. As a comparison, RW, CW, W and $\text{PRB}_{p^\star}$ estimators give the best results respectively only in 3, 3, 2 and 1 out of 21 times. Moreover, Figure 3.3 illustrates that the NN estimate features a nice stability in terms of bias and RMedSE for a wide range of k values on selected situations from Table 3.2. This phenomenon may be highly appreciated even when the NN estimator is not ranked first on the RMedSE criteria basis, see for instance the bottom pannel of Figure 3.3. As a conclusion, even though the NN method is numerically more expensive than its competitors, it provides a very effective estimator for all heavy-tailed situations. Here, the NNs are built with at most 20 neurons ($J \leq 5$), which remains acceptable from the computational cost point of view (computing the NN estimate on 500 replications in multiprocessing with 40 cores took 2 hours with a batch size of 256 and $J = 5$).

3.7 Illustration on rainfall data (conditional case)

The conditional NN estimators are tested on daily rainfall observations from 1958 to 2000 in the Cévennes-Vivarais region (southern part of France), see Figure 3.1. The region covers 256×283 km^2 where the Rhône River flows between two major mountainous massifs: the Massif Central and the Alps, respectively in the western and eastern sides. The northwestern quarter is a quite homogeneous high plateau, whereas the southern part is a large river plain bordered by the Mediterranean Sea. This region is very sensitive to extreme precipitations and flash floods [143]. Besides, the rainfall distribution exhibits different statistical properties, depending on both the time scale (whether hourly or daily) and the spatial scale (whether in a flat region close to the sea, or in the mountains) [33]. Although daily rainfall maxima used to be modeled with a Gumbel distribution [118, Section 7.2.2], better fits with the GPD are now preferred to overcome the underestimation of the extreme rainfall levels [33, 119]. The dataset is provided by the French meteorological service Météo-France and includes the $n_D = 15,706$ daily rainfall measurements in millimeters and the location of $n_S = 524$ stations, leading to a dataset of size $n = n_D \times n_S$. Observe that the highest daily rainfalls are located over the eastern slope of the Massif Central (Cévennes mountains range), which is a known phenomena [143]. The variable of interest X is the daily rainfall and the covariate Y is the three-dimensional location (longitude, latitude and altitude).

3.7.1 Data processing

Intermediate conditional quantiles $q(\cdot \mid y)$ are estimated locally by order statistics on a small neighborhood around the geographical location of interest y. To define this neighborhood, we fix the number of neighbors n_K and apply the nearest neighbors estimator on the covariate Y to cluster all the stations using the Mahalanobis distance $D(Y_t, Y_{t'}) := \sqrt{(Y_t - Y_{t'})^\top \Sigma^{-1} (Y_t - Y_{t'})}$ for all $(t, t') \in \{1, \ldots, n_S\}^2$, where Σ^{-1} is the inverse of the corresponding covariance matrix. Next, we merge all the historical values of the $n_K - 1$ closest stations of each station $t \in \{1, \ldots, n_S\}$, leading to $n_o = n_D \times n_K$ observations which are assumed to be i.i.d within each neighborhood. We denote by $X^{(1,n_o)}(Y_t) \leq \cdots \leq X^{(n_o,n_o)}(Y_t)$ the order statistics at a given station $t \in \{1, \ldots, n_S\}$. In addition, we introduce $n_h \in \{1, \ldots, n_o - 1\}$ and focus on the highest unique historical rainfalls $\left(X^{(n_o-i+1,n_o)}(Y_t), i \in \{1, \ldots, n_h\}\right)$, for each station $t \in \{1, \ldots, n_S\}$. The estimation of the conditional extreme quantile is investigated at level $1 - \alpha_n = 1 - 1/n_o$

	NN	W	RW	CW	CH	CH_p	PRB_p	$\text{CH}_{p^\star}$	$\text{PRB}_{p^\star}$
Burr ($\gamma = 1/8$)									
$\rho = -1/8$	0.0392	-	**0.0364**	-	0.5375	0.2713	0.3745	0.3578	0.1203
Burr ($\gamma = 1/4$)									
$\rho = -1/8$	0.1567	-	**0.1421**	-	-	-	-	-	0.6357
Burr ($\gamma = 1/2$)									
$\rho = -1/8$	**0.2847**	-	0.4298	-	-	-	-	-	-
Burr ($\gamma = 1$)									
$\rho = -1/8$	**0.3133**	-	0.8625	-	-	-	-	-	-
$\rho = -1/4$	**0.1962**	-	0.5423	-	-	-	-	-	0.6617
$\rho = -1/2$	0.2142	-	0.3291	-	0.0949	0.1021	0.1488	**0.0874**	0.1185
$\rho = -1$	0.1877	-	0.2438	**0.1289**	0.4120	0.3737	0.3761	0.3658	0.4261
$\rho = -2$	**0.1432**	0.2065	0.1488	0.2115	0.3394	0.3384	0.2893	0.2933	0.3058
NHW ($\gamma = 1/8$)									
$\rho = -1/8$	**0.0275**	-	0.0340	0.0699	0.2442	0.2194	0.3285	0.2202	0.3157
NHW ($\gamma = 1/4$)									
$\rho = -1/8$	**0.0570**	-	0.0816	0.1482	0.3290	3209	0.3890	0.3212	0.3935
NHW ($\gamma = 1/2$)									
$\rho = -1/8$	**0.1168**	-	0.1794	0.3683	0.5309	0.5284	0.5697	0.5155	0.5586
NHW ($\gamma = 1$)									
$\rho = -1/8$	**0.2709**	-	0.3885	0.5644	0.7789	0.7016	0.7379	0.7891	0.8039
$\rho = -1/4$	**0.2163**	-	0.3095	0.4888	0.6851	0.6920	0.6825	0.6897	0.7252
$\rho = -1/2$	**0.1615**	0.5927	0.2217	0.2481	0.4589	0.4939	0.4803	0.4595	0.4727
$\rho = -1$	0.1596	**0.0679**	0.1557	0.1549	0.2340	0.2582	0.444	0.2302	0.2353
$\rho = -2$	0.1082	**0.0738**	0.1302	0.1576	0.2112	0.1953	0.2080	0.1865	0.1879
Fisher ($\rho = -\gamma$)									
$\gamma = 1/8$	**0.0506**	-	0.0765	-	-	-	-	-	0.6126
$\gamma = 1$	0.1792	-	0.2871	**0.0882**	0.2722	0.2736	0.2323	0.2378	0.3409
GPD ($\rho = -\gamma$)									
$\gamma = 1/8$	0.0391	-	**0.0364**	-	0.5375	0.2534	0.3266	0.3578	0.1203
Inverse Gamma ($\rho = -\gamma$)									
$\gamma = 1$	0.1863	0.9259	0.1731	**0.1269**	0.2163	0.2317	0.2232	0.2030	0.2181
Student ($\rho = -2\gamma$)									
$\gamma = 1$	**0.1515**	0.5781	0.1961	0.3565	0.5654	0.5024	0.5273	0.5157	0.5439

Table 3.2: RMedSE associated with nine estimators of the extreme quantile $q(\alpha_n = 1/(2n))$ on six heavy-tailed distributions. The best result is emphasized in bold. RMedSEs larger than 1 are not reported.

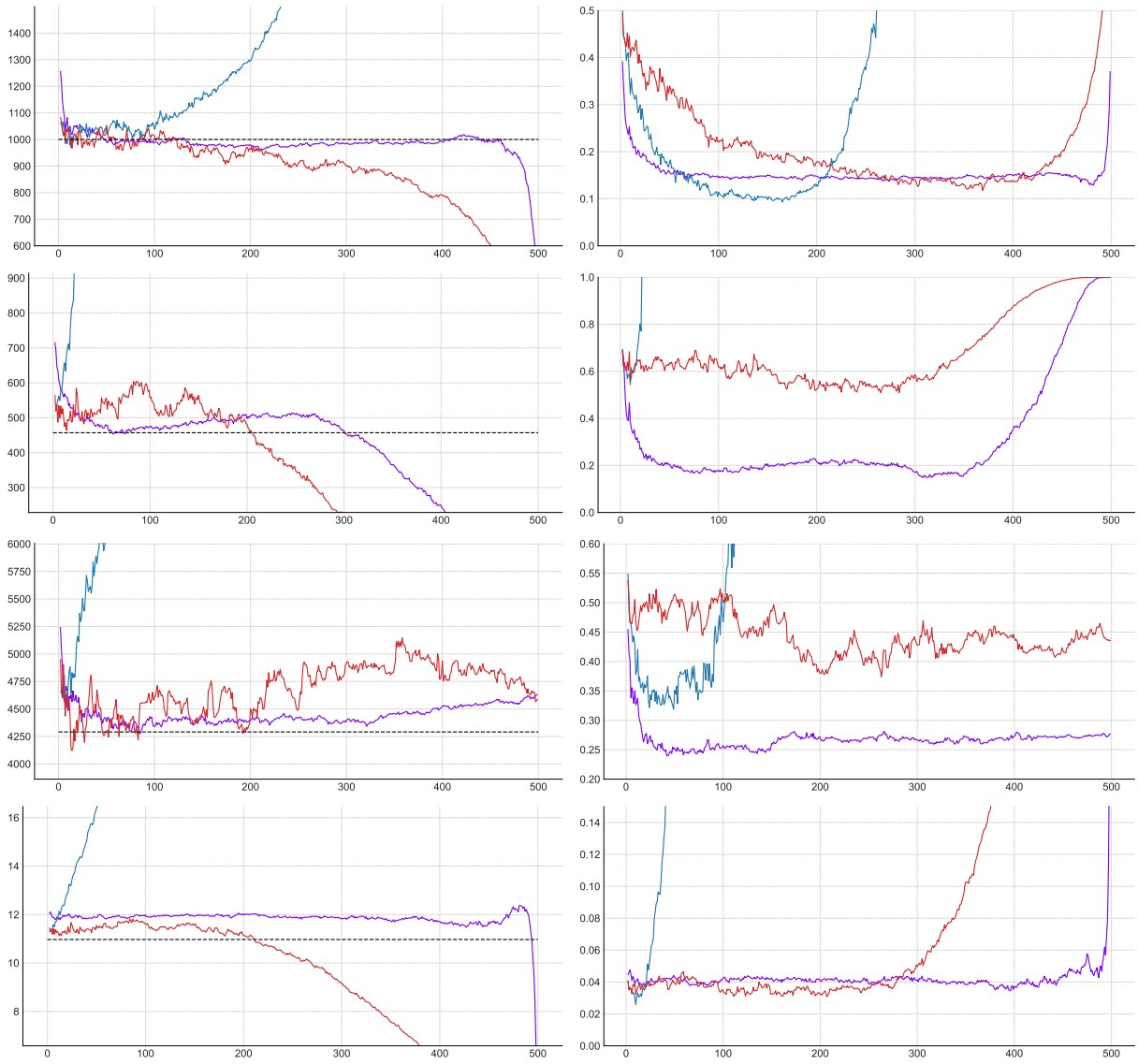


Figure 3.3: Illustration on simulated data sets of size $n = 500$ from a Burr distribution with $\gamma = 1$ and $\rho \in \{-2, -1/4\}$, a NHW distribution with $\gamma = 1$ and $\rho = -1/8$, and a GPD with $\gamma = 1/8$ (from top to bottom). Median of the estimators (left panel) of the extreme quantile (black dashed line) at level $1 - \alpha_n = 1 - 1/(2n)$ and RMedSE (right panel), as functions of $k \in \{2, \ldots, n-1\}$, computed on $n_R = 500$ replications, associated with W (blue), RW (red) and NN (purple) estimators.

by storing in a test set all maximum order statistics $X^{(n_o,n_o)}(Y_t)$ for further comparison; and keeping the remaining $(n_h - 1)$ ones in a train set for computation of the estimates.

Tail-index estimation Before moving to the implementation of the conditional extrapolation NNs, it is necessary to check whether the data are heavy-tailed. Additionally, we verify that the tail-index γ is independent from the covariate in all the considered n_K-neighborhoods as assumed in the location-dispersion model of Section 3.5.2. To this end, we first fix $n_h = 100$ and, for $n_K \in \{10, 15, \ldots, 50\}$, compute the Hill estimator $\hat{\gamma}_t^{\mathtt{H}}(k^\star)$, where $k^\star$ is selected by [4, Algorithm 1], for each station $t \in \{1, \ldots, n_S\}$ (see Figure 3.4a for an illustration on one station). Based on a graphical diagnosis (Figure 3.4b), we select $n_K = 45$ which highlights the lowest spread and skewness of the Hill estimates. The distribution of the estimated tail indices obtained with $n_h = 100$ and $n_K = 45$ (Figure 3.4c) has a small standard-deviation (0.031) around its mean (0.189), which confirms the hypothesis of a constant tail-index in the Cévennes-Vivarais region. Second, we validate the choice of $n_h = 100$ graphically (Figure 3.4d) leading to a small standard-deviation (0.029) of the slopes associated with the quantile-quantile plots around their mean (0.195).

In the next two paragraphs, we propose an estimation of the conditional extreme quantile $q(1-1/n_o|Y_t)$ at each station $t \in \{1, \ldots, n_S\}$ based on the two methods discussed in Section 3.5.1 and 3.5.2. Even if the assumption on γ is imposed only in the location-dispersion model, we keep the same dataset built with $n_h = 100$ and $n_K = 45$ for both approaches.

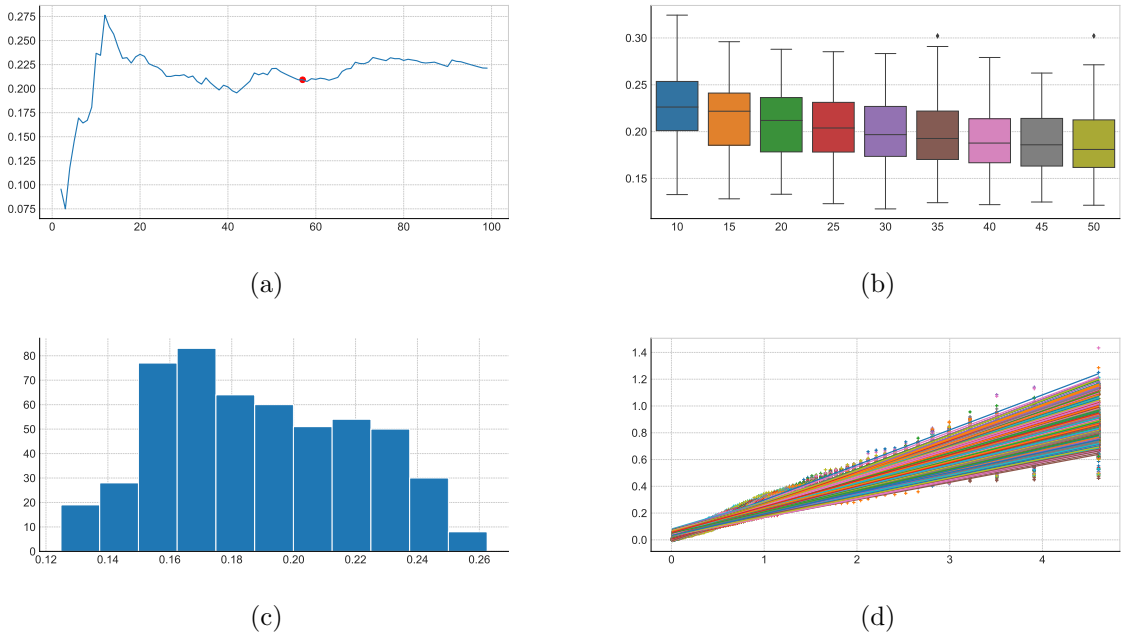


Figure 3.4: Illustrations on real data. (a) Example of Hill estimate as a function of $k \in \{2, \ldots, n_h - 1\}$, within the neighborhood of a given station with $n_h = 100$ and $n_K = 45$. The selected $k^\star$ is depicted by the red circle. (b) Box-plots of estimated $\hat{\gamma}^{\mathtt{H}}$'s as functions of n_K with $n_h = 100$. (c) Histogram of estimated $\hat{\gamma}^{\mathtt{H}}$'s for all stations $t \in \{1, \ldots, n_S\}$ with $n_K = 45$ and $n_h = 100$. (d) quantile-quantile plot $\log(n_h/i) \mapsto \log(X^{(n_o-i+1,n_o)}(Y_t)) - \log(X^{(n_o-n_h+1,n_o)}(Y_t))$, $t \in \{1, \ldots, n_S\}$, $i \in \{1, \ldots, n_h - 1\}$ with $n_h = 100$ and $n_K = 45$.

3.7.2 Conditional Extrapolation Neural Network (CENN)

Let us describe the implementation of the NN estimator of the (conditional) extreme quantile introduced in Section 3.5.1. Starting from the real data previously processed, first normalize

the covariate Y between 0 and 1 for training stability purposes [123]. Second, compute within each neighborhood $t \in \{1,\dots,n_S\}$ the $N = n_S n_h(n_h-1)/2$ empirical estimates $\hat{S}^{(i,k)}(Y_t) := \log\left(X^{(n_o-i+1,n_o)}(Y_t)\right) - \log\left(X^{(n_o-k+1,n_o)}(Y_t)\right)$ of the conditional log-spacings $\log q(1-i/n_o\,|\,Y_t) - \log q(1-k/n_o\,|\,Y_t)$, for $i \in \{1,\dots,k-1\}$ and $k \in \{2,\dots,n_h\}$. Next, build the test set containing the $N^{\text{test}} = n_S(n_h-1)$ empirical estimates $\hat{S}^{(1,k)}(Y_t)$ of the log-spacings, and the train set containing the $N^{\text{train}} = n_S(n_h-1)(n_h-2)/2$ remaining ones. The NN approximation $\tilde{f}^{\mathtt{NN}_J}_{\tilde{\phi}}$ of the conditional log-spacing function is fitted to the training data by minimizing the L_1 distance between two estimations of the log-spacings:

$$\hat{\phi} = \underset{\tilde{\phi}\in\Phi}{\arg\min} \frac{1}{N^{\text{train}}} \sum_{t=1}^{n_S} \sum_{k=3}^{n_h} \sum_{i=2}^{k-1} \left| \hat{S}^{(i,k)}(Y_t) - \tilde{f}^{\mathtt{NN}_J}_{\tilde{\phi}}(\log(k/i), \log(n_o/k), Y_t) \right|. \tag{3.7.1}$$

Model selection Algorithm 2 selects the parameters $\hat{\phi} = \hat{\phi}_{m^\star}(\mathcal{A}^\star)$ associated with the best architecture $\mathcal{A}^\star$ in Table 3.1 and iteration $m^\star \in \{1,\dots,n_M\}$ corresponding to the smallest $\text{median}_{t\in\{1,\dots,n_S\}} \mathtt{MAD}\left(\left\{\hat{q}^{\mathtt{NN}_J}_{\hat{\phi}_m}\left(1-\frac{1}{n_o}; 1-\frac{k}{n_o}\,|\,Y_t\right),\ k \in \{k_1,\dots,k_2\}\right\}\right)$. In all experiments we used $k_1 = [3n_h/100]$ and $k_2 = [3n_h/4]$. The performance criteria (3.6.2) is adapted to the conditional case as

$$\text{RMedSE}\left(\hat{q}^{\mathtt{NN}_J}_{\hat{\phi}_{m^\star}}, t, \frac{1}{n_o}\right) = \underset{t\in\{1,\dots,n_S\}}{\text{median}} \left(\frac{\hat{q}^{\mathtt{NN}_J}_{\hat{\phi}_{m^\star}}(1-\frac{1}{n_o}; 1-\frac{k^\star(t)}{n_o}\,|\,Y_t)}{X^{(n_o,n_o)}(Y_t)} - 1 \right)^2, \tag{3.7.2}$$

where $\hat{q}^{\mathtt{NN}_J}_{\hat{\phi}_{m^\star}}(1-1/n_o; 1-k^\star(t)/n_o\,|\,Y_t)$ denotes the NN estimation computed on the selected anchor index $k^\star(t) \in \{k_1,\dots,k_2\}$ using [4, Algorithm 1] with initial points $a^{(0)} = k_1$ and $c^{(0)} = k_2$, at the t-th station, $t \in \{1,\dots,n_S\}$.

3.7.3 Location-Dispersion Neural Network (LDNN)

Let us turn to the implementation of the Location-Dispersion NN estimator of the conditional extreme quantile introduced in Section 3.5.2. Starting from the real data processed in Section 3.7.1, first compute the $N = n_S n_h(n_h-1)(n_h-2)/6$ empirical estimates

$$\hat{G}^{(i,j,k)}(Y_t) := \frac{X^{(n_o-i+1,n_o)}(Y_t) - X^{(n_o-k+1,n_o)}(Y_t)}{X^{(n_o-k+1,n_o)}(Y_t) - X^{(n_o-j+1,n_o)}(Y_t)}.$$

of the modified conditional spacings $g(\log(k/i), \log(n_o/k), \log(k/j))$ within each neighborhood $t \in \{1,\dots,n_S\}$ for all $i \in \{1,\dots,k-1\}$, $k \in \{2,\dots,j-1\}$ and $j \in \{3,\dots,n_h\}$. Next, perform a similar train-test splitting as the one in Section 3.5.1, resulting in $N^{\text{train}} = n_S(n_h-1)(n_h-2)(n_h-3)/6$ and $N^{\text{test}} = n_S(n_h-1)(n_h-2)/2$. Thus, the Location-Dispersion NN approximation $\tilde{g}^{\mathtt{NN}_J}_{\tilde{\phi}}$ of g is fitted to the training data by minimizing the L_1 distance between two estimations of the spacings

$$\hat{\phi} = \underset{\tilde{\phi}\in\Phi}{\arg\min} \frac{1}{N^{\text{train}}} \sum_{t=1}^{n_S} \sum_{j=4}^{n_h} \sum_{k=3}^{j-1} \sum_{i=2}^{k-1} \left| \hat{G}^{(i,j,k)}(Y_t) - \tilde{g}^{\mathtt{NN}_J}_{\tilde{\phi}}(\log(k/i), \log(n_o/k), \log(k/j)) \right|, \tag{3.7.3}$$

where

$$\tilde{g}^{\mathtt{NN}_J}_{\tilde{\phi}}(\log(k/i), \log(n_o/k), \log(k/j)) = \frac{\exp\left(\tilde{f}^{\mathtt{NN}_J}_{\tilde{\phi}_1}(\log(k/i), \log(n_o/k))\right) - 1}{1 - \exp\left(\tilde{f}^{\mathtt{NN}_J}_{\tilde{\phi}_2}(\log(k/j), \log(n_o/k))\right)} \tag{3.7.4}$$

is the Location-Dispersion NN approximation with $\tilde{\phi} = \{\tilde{\phi}_1, \tilde{\phi}_2\}$. For a larger flexibility, we built two NN in (3.7.4) with a similar architecture but with a different initialization of

weights $\{\tilde{\phi}_1, \tilde{\phi}_2\}$. During the training, it may happen that $\hat{G}^{\mathtt{NN}_J,(i,j,k)}(\tilde{\phi})$ is not defined if $\tilde{f}_{\hat{\phi}}^{\mathtt{NN}_J}(\log(k/j), \log(n_o/k)) = 0$ in (3.7.4) for some pair (k, j). In this case, we do not take into account the gradient associated with these inputs in the optimization part.

Model selection Similarly to the previous case, Algorithm 3 selects the parameters $\hat{\phi} := \hat{\phi}_{m^\star}(\mathcal{A}^\star)$ associated with the best architecture $\mathcal{A}^\star$ in Table 3.1 and iteration $m^\star \in \{1, \dots, n_M\}$ corresponding to the median MAD:

$$\underset{t\in\{1,\dots,n_S\}}{\text{median}}\ \mathtt{MAD}\left(\left\{\hat{q}_{\hat{\phi}_m}^{\mathtt{NN}_J}\left(1-\frac{1}{n_o}; 1-\frac{k}{n_o}; 1-\frac{j}{n_o} \mid Y_t\right),\ k \in \{k_1, \dots, k_2\}, j \in \{j_1, \dots, j_2\}, k < j\right\}\right),$$

with $k_1 = [3n_h/100]$, $k_2 = [3n_h/4]$, $j_1 = [n_h/2]$ and $j_2 = n_h$. Moreover, in order to select the two anchors points $k^\star$ and $j^\star$, we introduce Algorithm 4 with $k_{\mathtt{U}} = [3n_h/100]$, $k_{\mathtt{D}} = [3n_h/4]$, $j_{\mathtt{L}} = [4n_h/100]$ and $j_{\mathtt{R}} = n_h$, which is a 2-dimensional extension of [4, Algorithm 1]. The performance criteria considered now is similar to (3.7.2):

$$\mathrm{RMedSE}\left(\hat{q}_{\hat{\phi}}^{\mathtt{NN}_J}, t, \frac{1}{n_o}\right) = \underset{t\in\{1,\dots,n_S\}}{\text{median}}\left(\frac{\hat{q}_{\hat{\phi}}^{\mathtt{NN}_J}(1-\frac{1}{n_o}; 1-\frac{k^\star(t)}{n_o}; 1-\frac{j^\star(t)}{n_o} \mid Y_t)}{X^{(n_o,n_o)}(Y_t)} - 1\right)^2.$$

3.7.4 Results

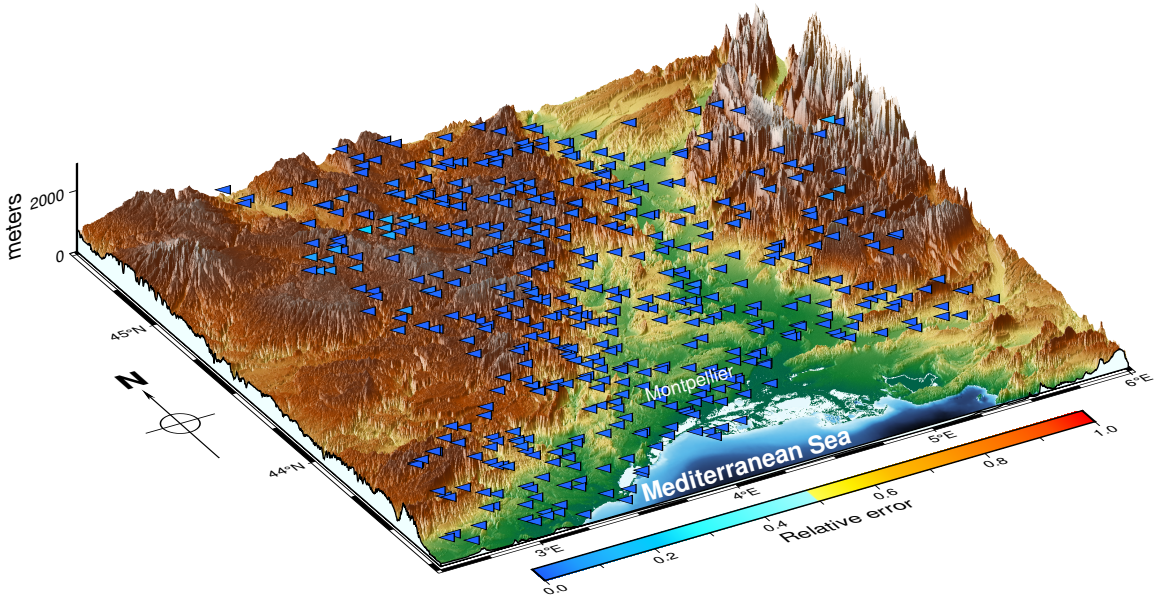
The selected hyperparameters of both CENN and LDNN models are respectively $\{J = 5, \text{batch size} = 512\}$ and $\{J = 2, \text{batch size} = 1,024\}$. While the former has an heavy parametrization (2,050 parameters estimated from 2,541,924 data), the latter requires a large training dataset (10 parameters estimated from 82,188,876 data). In the following, we first study the conditional extrapolation performance in the tails of both NNs at gauged stations. Then, we show an application to spatial interpolation through the CENN model.

Extrapolation at gauged stations. Figure 3.5 displays $\left(\frac{\hat{q}_{\hat{\phi}}^{\mathtt{NN}_J}(1-1/n_o;\cdot \mid Y_t)}{X^{(n_o,n_o)}(Y_t)} - 1\right)^2$, the squared relative error, between two estimates of the conditional extreme quantile of order $1 - 1/n_o$ for each station $t \in \{1, \dots, n_S\}$ under CENN and LDNN models. It appears that both models are very efficient for estimating conditional extreme quantiles. Additionally, the good results of the LDNN confirm the assumption of a tail-index independent of the covariate.

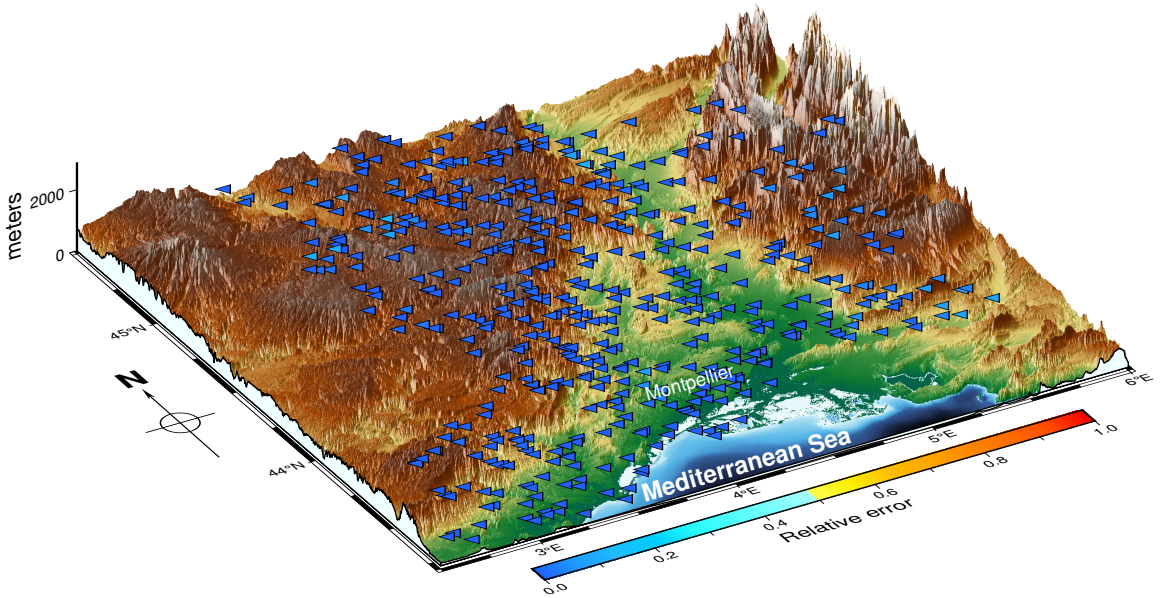
Spatial interpolation. We extend the previous analysis to the estimation of conditional extreme quantiles at all pixels in Figure 3.1, thus including ungauged locations. While this can be achieved with the two models, we limit ourselves to presenting the results associated with the CENN method, see Figure 3.6. The idea is to consider all 11,598,961 pixels in the high resolution map of Figure 3.1, and provide an estimation of the conditional extreme quantile of order $1 - \alpha_n = 1 - 1/n_o$, at locations not two far from a raingauged station. More specifically, the estimation is performed at points y of the covariate such that

$$D(y, Y_{t^\star}) \leq \kappa \sigma_{\mathtt{MAD}}(t^\star)/\sqrt{n_K}, \tag{3.7.5}$$

with $t^\star = \arg\min_{t\in\{1,\dots,n_S\}} D(y, Y_t)$, $\sigma_{\mathtt{MAD}}(t^\star) = \mathtt{MAD}(\{D_{t^\star}^{n_K-k+1,n_K},\ k \in \{1, \dots, n_K - 1\}\})$ and where $D_{t^\star}^{n_K-k+1,n_K}$ is the Mahalanobis distance between $Y_{t^\star}$ and its k-th nearest neighbor. In practice, we use $\kappa = 8$, leading to an extrapolation at 6,636,817 points. Observe that the largest daily precipitations occur in the Cévennes mountains range, which is in line with both Figure 3.1 and the literature [143].



(a) CENN (RMedSE=0.0047)



(b) LDNN (RMedSE=0.0022)

Figure 3.5: Estimation of the conditional extreme quantile at order $1 - \alpha_n = 1 - 1/n_o$ at each station. Squared relative error associated with the CENN (a) and LDNN (b) models.

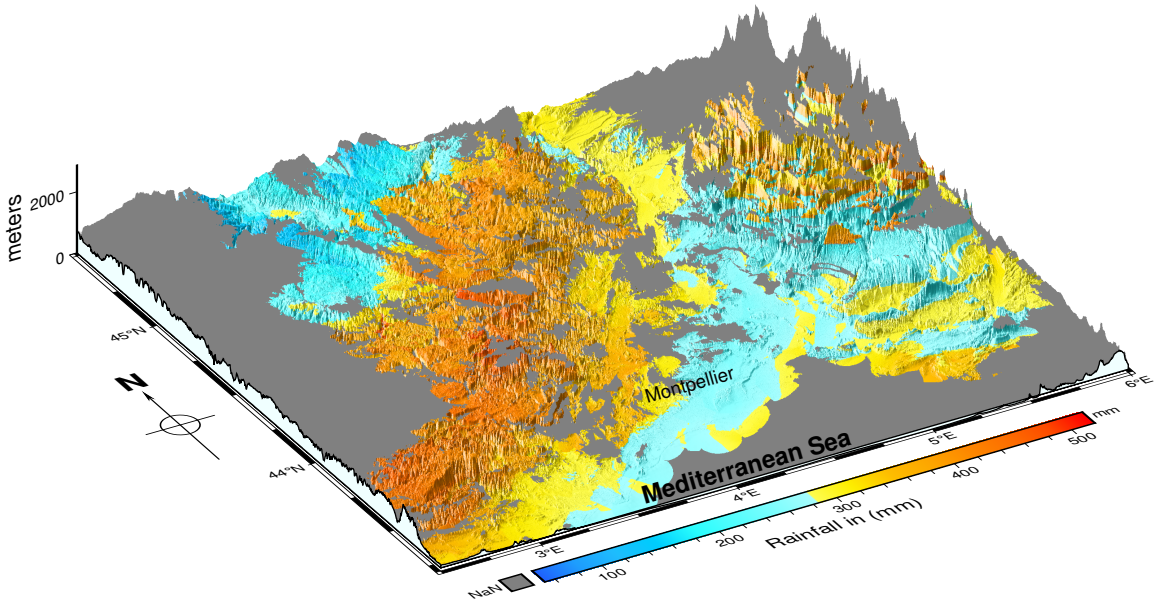


Figure 3.6: Spatial interpolation by the CENN quantile estimator at order $1 - \alpha_n = 1 - 1/n_o$. The gray region corresponds to Not a Number (NaN) when condition (3.7.5) is not satisfied.

3.8 Conclusion

We have introduced, to the best of our knowledge, the first NN approach dedicated to extreme quantile estimation in both non-conditional and conditional settings. In particular, two NNs of conditional extreme quantiles are proposed in order to tackle the cases where the tail-index depends or is independent of the covariate. From the theoretical point of view, the uniform convergence rates of the approximations underpinning the estimators are established within an extreme-value framework. From the practical point of view, our estimators have been tested on both simulated and real data; showing in the former case that the non-conditional NN estimator outperforms most of usual estimators in challenging heavy-tailed situations. In the rainfall data application, both conditional NN estimators reproduce properly the tails at raingauged stations, and are moreover able to perform spatial interpolation to estimate extreme rainfalls at ungauged locations.

Our further work will consist in adapting our NN estimators of extreme quantiles to other risk measures such as expected shortfall, or expectiles. To complete the current theoretical analysis which ensures accurate approximation in the univariate case, we will investigate (in the non-conditional case) multivariate extreme quantile estimation basing on recent characterizations through optimal transport [105].

3.A Algorithms

3.A.1 Model selection

We noticed that the following model selection techniques may be misleading during the first iterations when the NN is not well trained yet, leading to the same output for all anchor points k. Therefore, we decided to restrict the search of $m^\star$ after 10 iterations (step 3 of Algorithm **??**, step 7 of Algorithm 2 and Algorithm 3).

Algorithm 1: Model selection (non-conditional case)

Input: *approximation order:* J,
extrapolation quantile level: $\alpha_n \in (0,1)$,
initial left point: $k_1 \in \{2,\dots,n-2\}$,
initial right point: $k_2 \in \{3,\dots,n-1\}$

Output: *selected parameters:* $\hat{\phi}_{m^\star}(\mathcal{A}^\star)$

for *all architecture* $\mathcal{A}$ *in Table 3.1* **do**
  for $m = 1 : n_M$ **do**
    Optimize (3.6.1) to get $\hat{\phi}_m(\mathcal{A})$
    $Z_m(\mathcal{A}) \leftarrow \texttt{MAD}\left(\left\{\hat{q}^{\texttt{NN}_J}_{\hat{\phi}_m(\mathcal{A})}\left(1-\alpha_n; 1-\frac{k}{n}\right),\ k \in \{k_1,\dots,k_2\}\right\}\right)$

$(m^\star, \mathcal{A}^\star) \leftarrow \underset{(m\in\{10,\dots,n_M\},\,\mathcal{A})}{\arg\min} Z_m(\mathcal{A})$

Algorithm 2: Model selection (CENN)

Input: *order condition:* J,
extrapolation quantile level: $\alpha_n \in [0,1]$,
initial left point: $k_1 \in \{2,\dots,n_h-1\}$,
initial right point: $k_2 \in \{3,\dots,n_h\}$

Output: *selected parameters:* $\hat{\phi}_{m^\star}(\mathcal{A}^\star)$

for *all architecture* $\mathcal{A}$ *in Table 3.1* **do**
  for $m = 1 : n_M$ **do**
    Optimize (3.7.1) to get $\hat{\phi}_m(\mathcal{A})$
    for $t = 1 : n_S$ **do**
      $Z_m^{(t)}(\mathcal{A}) \leftarrow \texttt{MAD}\left(\left\{\hat{q}^{\texttt{NN}_J}_{\hat{\phi}_m(\mathcal{A})}\left(1-\alpha_n; 1-\frac{k}{n_o} \mid Y_t\right),\ k \in \{k_1,\dots,k_2\}\right\}\right)$
    $\bar{Z}_m(\mathcal{A}) \leftarrow \underset{t\in\{1,\dots,n_S\}}{\text{median}}\left\{Z_m^{(t)}(\mathcal{A})\right\}$

$(m^\star, \mathcal{A}^\star) \leftarrow \underset{(m\in\{10,\dots,n_M\},\,\mathcal{A})}{\arg\min} \bar{Z}_m(\mathcal{A})$

Algorithm 3: Model selection (LDNN)

Input: *order condition:* J,
extrapolation quantile level: $\alpha_n \in [0,1]$,
initial left point: $k_1 \in \{2,\dots,n_h-3\}$,
initial right point: $k_2 \in \{3,\dots,n_h-2\}$
initial upper point: $j_1 \in \{4,\dots,n_h-1\}$,
initial down point: $j_2 \in \{5,\dots,n_h\}$

Output: *selected parameters:* $\hat{\phi}_{m^\star}(\mathcal{A}^\star)$

for *all architecture* $\mathcal{A}$ *in Table 3.1* **do**
  for $m = 1 : n_M$ **do**
    Optimize (3.7.3) to get $\hat{\phi}_m(\mathcal{A}^\star)$
    for $t = 1 : n_S$ **do**

$$Z_m^{(t)}(\mathcal{A}) \leftarrow \texttt{MAD}\left(\left\{\hat{q}^{\texttt{NN}_J}_{\hat{\phi}_m(\mathcal{A})}\left(1-\alpha_n; 1-\tfrac{k}{n_o}; 1-\tfrac{j}{n_o} \mid Y_t\right), \; k \in \{k_1,\dots,k_2\}, j \in \{j_1,\dots,j_2\}, k<j\right\}\right)$$

    $$\bar{Z}_m(\mathcal{A}) \leftarrow \underset{t\in\{1,\dots,n_S\}}{\text{median}} \left\{Z_m^{(t)}(\mathcal{A})\right\}$$

$$(m^\star, \mathcal{A}^\star) \leftarrow \underset{(m\in\{10,\dots,n_M\},\, \mathcal{A})}{\arg\min} \bar{Z}_m(\mathcal{A})$$

3.A.2 Selection of the sample fractions

Algorithm 4: Selection of k and j using random forests 2D

Input: *triangular matrix:* $\mathcal{Z} = \{Z_{k,j}\}_{(k,j)\in\{2,\dots,j-1\}\times\{3,\dots,n-1\}}$,
number of trees: $T \in \mathbb{N} \setminus \{0\}$,
initial top point: $k_{\mathrm{T}}^{(0)} \in \{2,\dots,n-3\}$,
initial down point: $k_{\mathrm{D}}^{(0)} \in \{3,\dots,n-2\}$,
initial left point: $j_{\mathrm{L}}^{(0)} \in \{4,\dots,n-1\}$,
initial right point: $j_{\mathrm{R}}^{(0)} \in \{5,\dots,n\}$

Output: *selected points:* $k^\star$, $j^\star$

for $t = 1 : T$ **do**

$j_{\mathrm{L}}^{(t)} \sim \texttt{randint}(j_{\mathrm{L}}^{(0)}, j_{\mathrm{R}}^{(0)} - 1)$
$j_{\mathrm{R}}^{(t)} \sim \texttt{randint}(j_{\mathrm{L}}^{(t)} + 1, j_{\mathrm{R}}^{(0)})$
$k_{\mathrm{D}}^{(t)} \sim \texttt{randint}(k_{\mathrm{T}}^{(0)} + 1, k_{\mathrm{D}}^{(0)} \vee j_{\mathrm{L}}^{(t)})$
$k_{\mathrm{T}}^{(t)} \sim \texttt{randint}(k_{\mathrm{T}}^{(0)}, k_{\mathrm{D}}^{(t)} - 1)$
$k^{(t)}, j^{(t)} \leftarrow \mathrm{Tree2D}(\mathcal{Z}, j_{\mathrm{L}}^{(t)}, j_{\mathrm{R}}^{(t)}, k_{\mathrm{T}}^{(t)}, k_{\mathrm{D}}^{(t)})$

$k^\star \leftarrow \mathrm{median}(k^{(1)},\dots,k^{(T)}), \quad j^\star \leftarrow \mathrm{median}(j^{(1)},\dots,j^{(T)})$

Algorithm 5: Tree2D

Input: *triangular matrix:* $\mathcal{Z} = \{Z_{k,j}\}_{(k,j)\in\{2,\dots,j-1\}\times\{3,\dots,n\}}$,
initial top point: $k_{\mathrm{T}}^{(0)} \in \{2,\dots,n-3\}$,
initial down point: $k_{\mathrm{D}}^{(0)} \in \{3,\dots,n-2\}$,
initial left point: $j_{\mathrm{L}}^{(0)} \in \{4,\dots,n-1\}$,
initial right point: $j_{\mathrm{R}}^{(0)} \in \{5,\dots,n\}$

Output: *selected points:* k, j

$k_{\mathrm{M}} \leftarrow \left[\frac{k_{\mathrm{T}}+k_{\mathrm{D}}}{2}\right], \quad j_{\mathrm{M}} \leftarrow \left[\frac{j_{\mathrm{L}}+j_{\mathrm{R}}}{2}\right]$

while $(k_{\mathrm{M}} - k_{\mathrm{T}}) > 1$ *or* $(j_{\mathrm{M}} - j_{\mathrm{L}}) > 1$ **do**

  $V_{\mathrm{TL}} \leftarrow \mathrm{EmpiricalVariance2D}(\mathcal{Z}, k_{\mathrm{M}}, k_{\mathrm{T}}, j_{\mathrm{M}}, j_{\mathrm{L}})$

  $V_{\mathrm{TR}} \leftarrow \mathrm{EmpiricalVariance2D}(\mathcal{Z}, k_{\mathrm{M}}, k_{\mathrm{T}}, j_{\mathrm{R}}, j_{\mathrm{M}})$

  $V_{\mathrm{DL}} \leftarrow \mathrm{EmpiricalVariance2D}(\mathcal{Z}, k_{\mathrm{D}}, k_{\mathrm{M}}, j_{\mathrm{M}}, j_{\mathrm{L}})$

  $V_{\mathrm{DR}} \leftarrow \mathrm{EmpiricalVariance2D}(\mathcal{Z}, k_{\mathrm{D}}, k_{\mathrm{M}}, j_{\mathrm{R}}, j_{\mathrm{M}})$

  if $\min(V_{\mathrm{TL}}, V_{\mathrm{TR}}, V_{\mathrm{DL}}, V_{\mathrm{DR}}) = V_{\mathrm{TL}}$ **then**

    $k_{\mathrm{D}} \leftarrow k_M, \quad j_R \leftarrow j_{\mathrm{M}}$

  else if $\min(V_{\mathrm{TL}}, V_{\mathrm{TR}}, V_{\mathrm{DL}}, V_{\mathrm{DR}}) = V_{\mathrm{TR}}$ **then**

    $k_{\mathrm{D}} \leftarrow k_{\mathrm{M}}, \quad j_{\mathrm{L}} \leftarrow j_{\mathrm{M}}$

  else if $\min(V_{\mathrm{TL}}, V_{\mathrm{TR}}, V_{\mathrm{DL}}, V_{\mathrm{DR}}) = V_{\mathrm{DL}}$ **then**

    $k_{\mathrm{T}} \leftarrow k_{\mathrm{M}}, \quad j_{\mathrm{R}} \leftarrow j_{\mathrm{M}}$

  else

    $k_{\mathrm{U}} \leftarrow k_{\mathrm{M}}, \quad j_{\mathrm{L}} \leftarrow j_{\mathrm{M}}$

  $k_{\mathrm{M}} \leftarrow \left[\frac{k_{\mathrm{T}}+k_{\mathrm{D}}}{2}\right], \quad j_{\mathrm{M}} \leftarrow \left[\frac{j_{\mathrm{L}}+j_{\mathrm{R}}}{2}\right]$

Algorithm 6: EmpiricalVariance2D

Input: *triangular matrix:* $\mathcal{Z} = \{Z_{k,j}\}_{(k,j)\in\{2,\dots,j-1\}\times\{3,\dots,n\}}$,
initial points: $(k_a, k_b, j_a, j_b) \in \mathbf{N}^4$,

Output: *empirical variance:* $\hat{\sigma}^2$

Compute

$$\bar{Z} \leftarrow \frac{1}{(k_a - k_b) + (j_a - j_b) + 2} \sum_{k=k_b}^{k_a} \sum_{j=j_b}^{j_a} Z_{k,j}.$$

Compute

$$\hat{\sigma}^2 \leftarrow \frac{1}{(k_a - k_b) + (j_a - j_b) + 2} \sum_{k=k_b}^{k_a} \sum_{j=j_b}^{j_a} \left(Z_{k,j} - \bar{Z}\right)^2,$$

3.B Proofs

We start with a technical tool designed to simplify the intergral term (3.3.4) in the J-th order condition (3.3.3).

Lemma 3.B.1. *Let $K_t(s) := (s^t - 1)/t$ be defined for all $s \geq 1$ and $t < 0$. Then, for all $z \geq 1$ and $\rho_j < 0$ for $j \geq 2$, one can equivalently express (3.3.4) as:*

$$R_j(z) = \sum_{\ell=2}^{j} a_{\ell,j} K_{\bar{\rho}_\ell}(z),$$

where $\bar{\rho}_\ell = \rho_2 + \cdots + \rho_\ell$, and for some coefficients $a_{\ell,j} \in \mathbb{R}$.

Proof. For all $s \geq 1$, $p < 0$ and $q < 0$, one has

$$\int_1^s z^p K_q(z)\,\mathrm{d}z = \frac{1}{q}\left(K_{p+q+1}(s) - K_{p+1}(s)\right). \tag{3.B.1}$$

Replacing in (3.3.4) yields for all $j \geq 2$ and $y \geq 1$,

$$R_j(z) = \int_1^z z_2^{\rho_2 - 1} \int_1^{z_2} z_3^{\rho_3 - 1} \cdots \int_1^{z_{j-2}} z_{j-1}^{\rho_{j-1}-1} K_{\rho_j}(z_{j-1})\,\mathrm{d}z_{j-1} \ldots \mathrm{d}z_3\,\mathrm{d}z_2,$$

with

$$\int_1^{z_{j-1}} z_j^{\rho_j - 1}\,\mathrm{d}z_j = \frac{z_{j-1}^{\rho_j} - 1}{\rho_j} = K_{\rho_j}(z_{j-1}).$$

Assume $\rho_j < 0$ for all $j \geq 2$, then from (3.B.1), one can show by recursion that,

$$\begin{aligned}
R_j(z) &= \frac{1}{\rho_j}\left(\ldots\left(\frac{1}{\bar{\rho}_j - \bar{\rho}_3}\left(\frac{K_{\bar{\rho}_j}(z) - R_2(z)}{\bar{\rho}_j - \bar{\rho}_2} - R_3(z)\right) - \ldots\right) - R_{j-1}(z)\right), \text{ for } j \geq 4,\\
R_2(z) &= K_{\bar{\rho}_2}(z) = \frac{z^{\rho_2} - 1}{\rho_2},\\
R_3(z) &= \frac{1}{\rho_3}\Big(K_{\bar{\rho}_3}(z) - K_{\bar{\rho}_2}(z)\Big),
\end{aligned}$$

which concludes the proof. □

The next result presents how, starting from the J-th order condition, a NN approximation of $\varphi(x_1, x_2)$ can be built using $J(J-1)$ eLU functions.

Lemma 3.B.2. *Assume the J-th order condition (3.3.3) holds for some $J \geq 2$ with (3.3.5). Let $\bar{\rho}_J = \rho_2 + \cdots + \rho_J$ and $\tilde{\varphi}_\theta^{\mathtt{NN}_J}$ be the function defined for $x_1 > 0$ and $x_2 > 0$ by (3.3.6) for some $\theta = \left\{(w_i^{(1)}, w_i^{(2)}, w_i^{(3)}, w_i^{(4)}), i \in \{1, \ldots, J(J-1)/2\}\right\} \in \Theta := (\mathbb{R} \times \mathbb{R}_-^3)^{J(J-1)/2}$. Then, for all $\varepsilon > 0$, there exists $x_\varepsilon > 0$ such that*

$$\varphi(x_1, x_2) = \tilde{\varphi}_\theta^{\mathtt{NN}_J}(x_1, x_2) + \Delta(\exp(x_1), \exp(x_2)) \prod_{j=2}^{J} A_j(\exp(x_2)),$$

with $|\Delta(\exp(x_1), \exp(x_2))| \leq \varepsilon \exp(x_1(\bar{\rho}_J + \varepsilon))$ for all $x_1, x_2 \geq x_\varepsilon$.

Proof. Combining the J-th order condition (3.3.3) and [72, Theorem 2.1], we get that, for every $\varepsilon > 0$, there exists $t_0 > 0$ such that, for all $t \geq t_0$ and $tz \geq t_0$,

$$\log U(tz) - \log U(t) = \gamma \log z + \sum_{j=2}^{J} \prod_{\ell=2}^{j} A_\ell(t) R_j(z) + \Delta(z, t) \prod_{j=2}^{J} A_j(t),$$

with

$$|\Delta(z,t)| := \left| \frac{1}{A_J(t)} \left(\dots \left(\frac{1}{A_3(t)} \left(\frac{\log U(tz) - \log U(t) - \gamma \log z}{A_2(t)} - R_2(z) \right) - R_3(z) \right) - \dots \right) - R_J(z) \right| \leq \varepsilon z^{\bar{\rho}_J + \varepsilon}. \tag{3.B.2}$$

Thus, $L(t) = t^{-\gamma} U(t)$ yields

$$\log\left(\frac{L(tz)}{L(t)}\right) = \sum_{j=2}^{J} \prod_{\ell=2}^{j} A_\ell(t) R_j(z) + \Delta(z,t) \prod_{j=2}^{J} A_j(t),$$

or equivalently, considering $t = \exp(x_2)$, $z = \exp(x_1)$ and taking account of (3.2.3):

$$\begin{aligned} \varphi(x_1,x_2) &= \log\left(\frac{L(\exp(x_1+x_2))}{L(\exp(x_2))}\right) \\ &= \sum_{j=2}^{J} \prod_{\ell=2}^{j} A_\ell(\exp(x_2)) R_j(\exp(x_1)) + \Delta(\exp(x_1),\exp(x_2)) \prod_{j=2}^{J} A_j(\exp(x_2)). \end{aligned} \tag{3.B.3}$$

Using assumption (3.3.5) and replacing in (3.B.3), it follows:

$$\varphi(x_1,x_2) = \sum_{j=2}^{J} \prod_{\ell=2}^{j} c_\ell \exp(\rho_\ell x_2) R_j(\exp(x_1)) + \Delta(\exp(x_1),\exp(x_2)) \prod_{j=2}^{J} A_j(\exp(x_2)),$$

and thus, letting $\bar{\rho}_j = \rho_2 + \dots + \rho_j$ and $\bar{c}_j = c_2 \times \dots \times c_j$, one has

$$\varphi(x_1,x_2) = \sum_{j=2}^{J} \bar{c}_j \exp(\bar{\rho}_j x_2) R_j(\exp(x_1)) + \Delta(\exp(x_1),\exp(x_2)) \prod_{j=2}^{J} A_j(\exp(x_2)).$$

Introduce

$$\tilde{\varphi}_\theta^{\mathtt{NN}_J}(x_1,x_2) = \sum_{j=2}^{J} \bar{c}_j \exp(\bar{\rho}_j x_2) R_j(\exp(x_1)),$$

so that

$$\varphi(x_1,x_2) = \tilde{\varphi}_\theta^{\mathtt{NN}_J}(x_1,x_2) + \Delta(\exp(x_1),\exp(x_2)) \prod_{j=2}^{J} A_j(\exp(x_2)).$$

Taking account of Lemma 3.B.1, we have

$$\begin{aligned} \tilde{\varphi}_\theta^{\mathtt{NN}_J}(x_1,x_2) &= \sum_{j=2}^{J} \bar{c}_j \exp(\bar{\rho}_j x_2) \sum_{\ell=2}^{j} a_{\ell,j} K_{\bar{\rho}_\ell}(\exp(x_1)), \\ &= \sum_{j=2}^{J} \sum_{\ell=2}^{j} \frac{\bar{c}_j a_{\ell,j}}{\bar{\rho}_\ell} \left(\exp(\bar{\rho}_\ell x_1 + \bar{\rho}_j x_2) - \exp(\bar{\rho}_j x_2) \right). \end{aligned}$$

Re-indexing, we get

$$\tilde{\varphi}_\theta^{\mathtt{NN}_J}(x_1,x_2) = \sum_{i=1}^{J(J-1)/2} w_i^{(1)} \left(\exp\left(w_i^{(2)} x_1 + w_i^{(3)} x_2 \right) - \exp(w_i^{(4)} x_2) \right), \tag{3.B.4}$$

with, $w_i^{(1)} \in \mathbb{R}$, $w_i^{(2)} < 0$, $w_i^{(3)} < 0$, $w_i^{(4)} < 0$ for all $i \in \{1, \dots, J(J-1)/2\}$. Replacing $\sigma^{\mathtt{E}}$ in (3.B.4) yields the expression (3.3.6) of $\tilde{\varphi}_\theta^{\mathtt{NN}_J}$ in terms of eLU functions. The result is proved. □

Proof of Theorem 3.3.1. Let (ε_n) be a sequence in $(0, -\bar{\rho}_J)$. We have, in view of the triangle inequality:

$$\begin{aligned}
&\inf_{\tilde{\phi}\in\Phi} \left|\log q(1-\alpha_n) - \log \tilde{q}^{\mathtt{NN}_J}_{\tilde{\phi}}(1-\alpha_n; 1-\delta_n)\right| \\
&= \inf_{\tilde{\phi}\in\Phi} \left|f(\log(\delta_n/\alpha_n), \log(1/\delta_n)) - \tilde{f}^{\mathtt{NN}_J}_{\tilde{\phi}}(\log(\delta_n/\alpha_n), \log(1/\delta_n))\right| \\
&\leq \inf_{\tilde{w}_0\in\mathbb{R}_+} |\gamma - \tilde{w}_0| \log(\delta_n/\alpha_n) \\
&+ \inf_{\tilde{\theta}\in\Theta} \left|\varphi(\log(\delta_n/\alpha_n), \log(1/\delta_n)) - \tilde{\varphi}^{\mathtt{NN}_J}_{\tilde{\theta}}(\log(\delta_n/\alpha_n), \log(1/\delta_n))\right| \\
&\leq \left|\varphi(\log(\delta_n/\alpha_n), \log(1/\delta_n)) - \tilde{\varphi}^{\mathtt{NN}_J}_{\theta}(\log(\delta_n/\alpha_n), \log(1/\delta_n))\right| \\
&\leq |\Delta(\delta_n/\alpha_n, 1/\delta_n)| \left|\prod_{j=2}^{J} A_j(1/\delta_n)\right|,
\end{aligned}$$

from Lemma 3.B.2. Moreover, since $\delta_n/\alpha_n \to \infty$ and $1/\delta_n \to \infty$, for n large enough,

$$|\Delta(\delta_n/\alpha_n, 1/\delta_n)| \leq \varepsilon_n \left(\frac{\delta_n}{\alpha_n}\right)^{\bar{\rho}_J+\varepsilon_n}, \tag{3.B.5}$$

while, under assumption (3.3.5),

$$\left|\prod_{j=2}^{J} A_j(1/\delta_n)\right| = |\bar{c}_J|\, \delta_n^{-\bar{\rho}_J},$$

where $\bar{c}_J = c_2 \times \cdots \times c_J$. As a conclusion,

$$\alpha_n^{\bar{\rho}_J} \inf_{\hat{\phi}\in\Phi} \left|\log q(1-\alpha_n) - \log \tilde{q}^{\mathtt{NN}_J}_{\tilde{\phi}}(1-\alpha_n; 1-\delta_n)\right| \leq |\bar{c}_J|\varepsilon_n \left(\frac{\delta_n}{\alpha_n}\right)^{\varepsilon_n},$$

and letting $\varepsilon_n = \exp(-\mathcal{W}(\log(\delta_n/\alpha_n))) \to 0$ as $n \to \infty$, where $\mathcal{W}$ is the Lambert-W function [44], yields $\log(1/\varepsilon_n)/\varepsilon_n = \log(\delta_n/\alpha_n)$, the result is proved. □

Proof of Theorem 3.5.1. Introducing

$$\begin{aligned}
\varphi_n(y) &= \varphi\left(\log(\delta_n/\alpha_n), \log(1/\delta_n) \,|\, y\right), \\
\tilde{\varphi}^{\mathtt{NN}_J}_{n,\tilde{\theta}}(y) &= \tilde{\varphi}^{\mathtt{NN}_J}_{\tilde{\theta}}\left(\log(\delta_n/\alpha_n), \log(1/\delta_n) \,|\, y\right), \\
V_n(y) &= \Delta(\delta_n/\alpha_n, 1/\delta_n \,|\, y) \prod_{j=2}^{J} A_j(1/\delta_n \,|\, y), \\
H_{n,i}(y) &= \sigma^{\mathtt{E}}\left(w_i^{(2)}(y)\log(\delta_n/\alpha_n) + w_i^{(3)}(y)\log(1/\delta_n)\right) - \sigma^{\mathtt{E}}\left(w_i^{(4)}(y)\log(1/\delta_n)\right), \qquad (3.B.6) \\
\tilde{H}^{\mathtt{NN}}_{n,\tilde{\theta}_i^{(2-4)}}(y) &= \sigma^{\mathtt{E}}\left(\tilde{w}^{\mathtt{NN}}_{\tilde{\theta}_i^{(2)}}(y)\log(\delta_n/\alpha_n) + \tilde{w}^{\mathtt{NN}}_{\tilde{\theta}_i^{(3)}}(y)\log(1/\delta_n)\right) - \sigma^{\mathtt{E}}\left(\tilde{w}^{\mathtt{NN}}_{\tilde{\theta}_i^{(4)}}(y)\log(1/\delta_n)\right), \qquad (3.B.7)
\end{aligned}$$

for all $i \in \{1, \ldots, J(J-1)/2\}$ and where $\Delta(\delta_n/\alpha_n, 1/\delta_n \,|\, y)$ is defined similarly to (3.B.2) in the unconditional case. Remark that all functions $w^{(0)}(\cdot) = \gamma(\cdot)$ and $w_i^{(j)}(\cdot)$, $i \in \{1, \ldots, J(J-1)/2\}$, $j \in \{1, \ldots, 4\}$ are assumed to be continuous on $\mathcal{Y}$ w.r.t. the covariate y. It is known from [190,

Theorem 2] that there exists a deep ReLU NN that can uniformly approximate any of these continuous functions on a compact set with an error

$$\epsilon(p_n^{(j)}) := \max_{i\in\{1,\dots,J(J-1)/2\}} \inf_{\tilde\theta_i^{(j)}} \sup_{y\in\mathcal{Y}} \left| w_i^{(j)}(y) - \tilde w^{\mathtt{NN}}_{\tilde\theta_i^{(j)}}(y)\right| = O((p_n^{(j)})^{-2}) \tag{3.B.8}$$

requiring $2d_y+10$ neurons in each of the $p_n^{(j)}$ hidden layers, $j\in\{0,\dots,4\}$. This error is optimal with respect to the depth [190, Theorem 1(a)]. We have, in view of the triangle inequality:

$$\begin{aligned}\inf_{\tilde\phi\in\Phi}\sup_{y\in\mathcal{Y}} \left|\log q(1-\alpha_n\mid y) - \log \tilde q^{\mathtt{NN}_J}_{\tilde\phi}(1-\alpha_n; 1-\delta_n\mid y)\right| &\le \inf_{\tilde\theta^{(0)}}\sup_{y\in\mathcal{Y}} \left|\gamma(y) - \tilde w^{\mathtt{NN}_J}_{\tilde\theta^{(0)}}(y)\right| \log(\delta_n/\alpha_n)\\ &\quad + \inf_{\tilde\theta}\sup_{y\in\mathcal{Y}} \left|\varphi_n(y) - \tilde\varphi^{\mathtt{NN}_J}_{n,\tilde\theta}(y)\right|.\end{aligned}$$

The first term can easily be controlled thanks to (3.B.8):

$$\inf_{\tilde\theta^{(0)}}\sup_{y\in\mathcal{Y}} \left|\gamma(y) - \tilde w^{\mathtt{NN}_J}_{\tilde\theta^{(0)}}(y)\right| \log(\delta_n/\alpha_n) \le \epsilon(p_n^{(0)})\log(\delta_n/\alpha_n).$$

Next, rearranging and applying the triangle inequality entail

$$\begin{aligned}\inf_{\tilde\theta}\sup_{y\in\mathcal{Y}} \left|\varphi_n(y) - \tilde\varphi^{\mathtt{NN}_J}_{n,\tilde\theta}(y)\right| &= \inf_{\tilde\theta}\sup_{y\in\mathcal{Y}} \left| \sum_{i=1}^{J(J-1)/2} w_i^{(1)}(y) H_{n,i}(y) - \sum_{i=1}^{J(J-1)/2} \tilde w^{\mathtt{NN}}_{\tilde\theta_i^{(1)}}(y)\tilde H^{\mathtt{NN}}_{n,\tilde\theta_i^{(2-4)}}(y) + V_n(y)\right|\\
&= \inf_{\tilde\theta}\sup_{y\in\mathcal{Y}} \left| \sum_{i=1}^{J(J-1)/2} \left(w_i^{(1)}(y) - \tilde w^{\mathtt{NN}}_{\tilde\theta_i^{(1)}}\right) H_{n,i}(y)\right.\\
&\qquad \left. + \sum_{i=1}^{J(J-1)/2} \left(H_{n,i}(y) - \tilde H^{\mathtt{NN}}_{n,\tilde\theta_i^{(2-4)}}(y)\right)\tilde w^{\mathtt{NN}}_{\tilde\theta_i^{(1)}}(y) + V_n(y)\right|\\
&\le \sum_{i=1}^{J(J-1)/2} \inf_{\tilde\theta_i^{(1)}}\sup_{y\in\mathcal{Y}} \left| w_i^{(1)}(y) - \tilde w^{\mathtt{NN}}_{\tilde\theta_i^{(1)}}\right| \sup_{y\in\mathcal{Y}} |H_{n,i}(y)| \qquad (3.B.9)\\
&\quad + \sum_{i=1}^{J(J-1)/2} \inf_{\tilde\theta_i^{(2-4)}}\sup_{y\in\mathcal{Y}} \left| H_{n,i}(y) - \tilde H^{\mathtt{NN}}_{n,\tilde\theta_i^{(2-4)}}(y)\right| \inf_{\tilde\theta^{(1)}}\sup_{y\in\mathcal{Y}} \left|\tilde w^{\mathtt{NN}}_{\tilde\theta_i^{(1)}}(y)\right| \qquad (3.B.10)\\
&\quad + \sup_{y\in\mathcal{Y}} |V_n(y)|. \qquad (3.B.11)\end{aligned}$$

The three terms (3.B.9), (3.B.10) and (3.B.11) are considered separately. First, note that

$$\sup_{y\in\mathcal{Y}} |H_{n,i}(y)| \le 1, \tag{3.B.12}$$

since $w_i^{(j)}(y)\le 0$ for all $i\in\{1,\dots,J(J-1)/2\}$, $j=1,2,3$ and $y\in\mathcal{Y}$ in (3.B.6). Combining (3.B.8) and (3.B.12) yields

$$(3.B.9) \le \epsilon(p_n^{(1)})\frac{J(J-1)}{2}.$$

Next, focusing on (3.B.10), and taking account of $\left(\tilde w^{\mathtt{NN}}_{\tilde\theta_i^{(2)}}(\cdot), \tilde w^{\mathtt{NN}}_{\tilde\theta_i^{(3)}}(\cdot), \tilde w^{\mathtt{NN}}_{\tilde\theta_i^{(4)}}(\cdot)\right)\in\mathbb{R}^3_-$ by construc-

tion, one has for all $i \in \{1, \dots, J(J-1)/2\}$,

$$\begin{aligned}
&\inf_{\tilde{\theta}_i^{(2-4)}} \sup_{y\in\mathcal{Y}} \left| H_{n,i}(y) - \tilde{H}^{\mathtt{NN}}_{n,\tilde{\theta}_i^{(2-4)}}(y) \right| \\
&\leq \inf_{\tilde{\theta}_i^{(2-3)}} \sup_{y\in\mathcal{Y}} \left| \exp\left(w_i^{(2)}(y)\log(\delta_n/\alpha_n) + w_i^{(3)}(y)\log(1/\delta_n) \right) - \exp\left(\tilde{w}^{\mathtt{NN}}_{\tilde{\theta}_i^{(2)}}(y)\log(\delta_n/\alpha_n) + \tilde{w}^{\mathtt{NN}}_{\tilde{\theta}_i^{(3)}}(y)\log(1/\delta_n) \right) \right| \\
&+ \inf_{\tilde{\theta}_i^{(4)}} \sup_{y\in\mathcal{Y}} \left| \exp\left(w_i^{(4)}(y)\log(1/\delta_n) \right) - \exp\left(\tilde{w}^{\mathtt{NN}}_{\tilde{\theta}_i^{(4)}}(y)\log(1/\delta_n) \right) \right| \\
&\leq \inf_{\tilde{\theta}_i^{(2-3)}} \sup_{y\in\mathcal{Y}} \left| 1 - \exp\left(\log(\delta_n/\alpha_n)\left(\tilde{w}^{\mathtt{NN}}_{\tilde{\theta}_i^{(2)}}(y) - w_i^{(2)}(y) \right) + \log(1/\delta_n)\left(\tilde{w}^{\mathtt{NN}}_{\tilde{\theta}_i^{(3)}}(y) - w_i^{(3)}(y) \right) \right) \right| \qquad (3.B.13) \\
&+ \inf_{\tilde{\theta}_i^{(4)}} \sup_{y\in\mathcal{Y}} \left| 1 - \exp\left(\log(1/\delta_n)\left(\tilde{w}^{\mathtt{NN}}_{\tilde{\theta}_i^{(4)}}(y) - w_i^{(4)}(y) \right) \right) \right| . \qquad (3.B.14)
\end{aligned}$$

Let us first consider

$$h_{n,\tilde{\theta}_i^{(2-3)}}(y) = \log(\delta_n/\alpha_n)\left(\tilde{w}^{\mathtt{NN}}_{\tilde{\theta}_i^{(2)}}(y) - w_i^{(2)}(y) \right) + \log(1/\delta_n)\left(\tilde{w}^{\mathtt{NN}}_{\tilde{\theta}_i^{(3)}}(y) - w_i^{(3)}(y) \right),$$

and remark that $\log(\delta_n/\alpha_n)\epsilon(p_n^{(2)}) \to 0$ and $\log(1/\delta_n)\epsilon(p_n^{(3)}) \to 0$ as $n \to \infty$ imply

$$\sup_{y\in\mathcal{Y}} \left| h_{n,\tilde{\theta}_i^{(2-3)}}(y) \right| \leq \log(\delta_n/\alpha_n)\epsilon(p_n^{(2)}) + \log(1/\delta_n)\epsilon(p_n^{(3)}) \leq \log 2$$

for n large enough. Since $|1 - \exp(u)| \leq 2\,|u|$ for any $|u| \leq \log 2$, it follows

$$(3.B.13) = \inf_{\tilde{\theta}_i^{(2-3)}} \sup_{y\in\mathcal{Y}} \left| 1 - \exp\left(h_{n,\tilde{\theta}_i^{(2-3)}}(y) \right) \right| \leq 2\left(\log(\delta_n/\alpha_n)\epsilon(p_n^{(2)}) + \log(1/\delta_n)\epsilon(p_n^{(3)}) \right).$$

Second, applying the same method to control (3.B.14) yields

$$\inf_{\tilde{\theta}_i^{(2-4)}} \sup_{y\in\mathcal{Y}} \left| H_{n,i}(y) - \tilde{H}^{\mathtt{NN}}_{n,\tilde{\theta}_i^{(2-4)}}(y) \right| \leq 2\left(\log(\delta_n/\alpha_n)\epsilon(p_n^{(2)}) + \log(1/\delta_n)\left(\epsilon(p_n^{(3)}) + \epsilon(p_n^{(4)}) \right) \right),$$

since $\log(1/\delta_n)\epsilon(p_n^{(4)}) \to 0$ as $n \to \infty$. Moreover, in view of (3.B.8) we have

$$\inf_{\tilde{\theta}_i^{(1)}} \sup_{y\in\mathcal{Y}} \left| \tilde{w}^{\mathtt{NN}}_{\tilde{\theta}_i^{(1)}}(y) \right| \leq \sup_{y\in\mathcal{Y}} \left| w_i^{(1)}(y) \right| + \epsilon(p^{(1)}) \leq C_1 + \epsilon(p_n^{(1)}),$$

where $C_1 \geq 0$ is a constant since $w_i^{(1)}(\cdot)$ is by assumption a continuous function on a compact set. Together with (3.B.13) and (3.B.14), this entails

$$(3.B.10) \leq J(J-1)\left(\log(\delta_n/\alpha_n)\epsilon(p_n^{(2)}) + \log(1/\delta_n)\left(\epsilon(p_n^{(3)}) + \epsilon(p_n^{(4)}) \right) \right)\left(C_1 + \epsilon(p_n^{(1)}) \right).$$

Finally, under assumption (3.5.3), the last term (3.B.11) can be rewritten using

$$V_n(y) = \Delta\left(\delta_n/\alpha_n, 1/\delta_n \mid y \right) \delta_n^{-\bar{\rho}_J(y)} \bar{c}_J(y),$$

with $\bar{c}_J(y) = \prod_{j=2}^J c_j(y)$ and $\bar{\rho}_J(y) = \sum_{j=2}^J \rho_j(y)$. Taking advantage of (3.B.5) yields

$$\sup_{y\in\mathcal{Y}} \left| \Delta\left(\delta_n/\alpha_n, 1/\delta_n \mid y \right) \right| \left(\delta_n/\alpha_n \right)^{-\bar{\rho}_J(y)} \leq \varepsilon_n \left(\delta_n/\alpha_n \right)^{\varepsilon_n}, \qquad (3.B.15)$$

where $\varepsilon_n = \exp(-\mathcal{W}(\log(\delta_n/\alpha_n)))$ is defined in Lemma 3.B.2. Therefore,

$$\begin{aligned}\sup_{y\in\mathcal{Y}}|V_n(y)| &\leq \sup_{y\in\mathcal{Y}}|\Delta\left(\delta_n/\alpha_n, 1/\delta_n \mid y\right)|\,\delta_n^{-\bar{\rho}_J(y)}\sup_{y\in\mathcal{Y}}\bar{c}_J(y)\\ &\leq \sup_{y\in\mathcal{Y}}|\Delta\left(\delta_n/\alpha_n, 1/\delta_n \mid y\right)|\left(\frac{\delta_n}{\alpha_n}\right)^{-\bar{\rho}_J(y)}\sup_{y\in\mathcal{Y}}\alpha_n^{-\bar{\rho}_J(y)}\sup_{y\in\mathcal{Y}}\bar{c}_J(y),\end{aligned}$$

and combining with (3.B.15), it yields

$$\sup_{y\in\mathcal{Y}}|V_n(y)| \leq c_{\sup}\,\varepsilon_n\,(\delta_n/\alpha_n)^{\varepsilon_n}\,\alpha_n^{-\bar{\rho}_{\sup}},$$

where $\bar{c}_{\sup} := \sup_{y\in\mathcal{Y}}\bar{c}_J(y)$ and $\bar{\rho}_{\sup} := \sup_{y\in\mathcal{Y}}\bar{\rho}_J(y)$. All in all, one has

$$\begin{aligned}&\inf_{\tilde{\theta}}\sup_{y\in\mathcal{Y}}\left|\varphi(y) - \tilde{\varphi}_{\tilde{\theta}}^{\mathtt{NN}_J}(y)\right|\\ &\quad= \mathcal{O}\left(\epsilon(p_n^{(1)})\right) + \mathcal{O}\left(\log(\delta_n/\alpha_n)\epsilon\left(p_n^{(2)}\right)\right) + \mathcal{O}\left(\log(1/\delta_n)\left(\epsilon\left(p_n^{(3)}\right) + \epsilon\left(p_n^{(4)}\right)\right)\right) + \mathcal{O}\left(\alpha_n^{-\bar{\rho}_{\sup}}\right),\end{aligned}$$

leading to

$$\begin{aligned}&\alpha_n^{\bar{\rho}_{\sup}}\inf_{\tilde{\phi}\in\Phi}\sup_{y\in\mathcal{Y}}\left|\log q(1-\alpha_n \mid y) - \log\tilde{q}_{\tilde{\phi}}^{\mathtt{NN}_J}(1-\alpha_n; 1-\delta_n \mid y)\right|\\ &\quad= \mathcal{O}\left(\alpha_n^{\bar{\rho}_{\sup}}\log(\delta_n/\alpha_n)\left(\epsilon\left(p_n^{(0)}\right) + \epsilon\left(p_n^{(2)}\right)\right)\right) + \mathcal{O}\left(\alpha_n^{\bar{\rho}_{\sup}}\log(1/\delta_n)\left(\epsilon(p_n^{(3)}) + \epsilon(p_n^{(4)})\right)\right)\\ &\quad+ \mathcal{O}\left(\alpha_n^{\bar{\rho}_{\sup}}\epsilon(p_n^{(1)})\right) + \mathcal{O}(1)\\ &\quad= \mathcal{O}\left(\alpha_n^{\bar{\rho}_{\sup}}\log(\delta_n/\alpha_n)\left(\left(p_n^{(0)}\right)^{-2} + \left(p_n^{(2)}\right)^{-2}\right)\right) + \mathcal{O}\left(\alpha_n^{\bar{\rho}_{\sup}}\log(1/\delta_n)\left(\left(p_n^{(3)}\right)^{-2} + \left(p_n^{(4)}\right)^{-2}\right)\right)\\ &\quad+ \mathcal{O}\left(\alpha_n^{\bar{\rho}_{\sup}}\left(p_n^{(1)}\right)^{-2}\right) + \mathcal{O}(1),\end{aligned}$$

and the result follows. □

Proof of Theorem 3.5.2. Introducing

$$\begin{aligned}g_n &= g(\log(\delta_n/\alpha_n), \log(1/\delta_n), \log(\delta_n/\tau_n)),\\ \tilde{g}_n(\tilde{\phi}) &= \tilde{g}_{\tilde{\phi}}^{\mathtt{NN}_J}(\log(\delta_n/\alpha_n), \log(1/\delta_n), \log(\delta_n/\tau_n)),\\ f_{\delta_n,\cdot} &= f_Z(\log(\delta_n/\cdot), \log(1/\delta_n)),\\ \tilde{f}_{\delta_n,\cdot}(\tilde{\phi}) &= \tilde{f}_{\tilde{\phi}}^{\mathtt{NN}_J}(\log(\delta_n/\cdot), \log(1/\delta_n)),\\ \omega_n(\tilde{\phi}) &= \frac{\tilde{g}_n(\tilde{\phi})}{g_n} - 1,\\ \lambda_n(y) &= \frac{q(1-\tau_n \mid y)}{q(1-\delta_n \mid y)},\end{aligned}$$

we have

$$\begin{aligned}
&\inf_{\tilde{\phi}\in\Phi}\sup_{y\in\mathcal{Y}}\left|\log q(1-\alpha_n\,|\,y)-\log\tilde{q}^{\mathtt{NN}_J}_{\tilde{\phi}}(1-\alpha_n;1-\delta_n,1-\tau_n\,|\,y)\right|\\
&=\inf_{\tilde{\phi}\in\Phi}\sup_{y\in\mathcal{Y}}\left|\log(1+(1-\lambda_n(y))g_n)-\log(1+(1-\lambda_n(y))\tilde{g}_n(\tilde{\phi}))\right|\\
&=\inf_{\tilde{\phi}\in\Phi}\sup_{y\in\mathcal{Y}}\left|\log\left(\frac{1+(1-\lambda_n(y))(1+\omega_n(\tilde{\phi}))g_n}{1+(1-\lambda_n(y))g_n}\right)\right|\\
&=\inf_{\tilde{\phi}\in\Phi}\sup_{y\in\mathcal{Y}}\left|\log\left(1+\frac{(1-\lambda_n(y))\omega_n(\tilde{\phi})g_n}{1+(1-\lambda_n(y))g_n}\right)\right|\\
&=\inf_{\tilde{\phi}\in\Phi}\sup_{y\in\mathcal{Y}}\left|\log\left(1+\frac{(1-\lambda_n(y))\omega_n(\tilde{\phi})}{(1/g_n)+(1-\lambda_n(y))}\right)\right|\\
&=:\inf_{\tilde{\phi}\in\Phi}\sup_{y\in\mathcal{Y}}\left|\log\left(1+\Lambda_n(\tilde{\phi}\,|\,y)\right)\right|. \qquad (3.B.16)
\end{aligned}$$

Besides, remark that

$$\inf_{\tilde{\phi}\in\Phi}\sup_{y\in\mathcal{Y}}\left|\Lambda_n(\tilde{\phi}\,|\,y)\right|=\inf_{\tilde{\phi}\in\Phi}\left|\omega_n(\tilde{\phi})\right|\frac{1-\inf_{y\in\mathcal{Y}}\lambda_n(y)}{(1/g_n)+(1-\inf_{y\in\mathcal{Y}}\lambda_n(y))}$$

and $\inf_{y\in\mathcal{Y}}\lambda_n(y)\to 0$ as $n\to\infty$ since $b(\cdot)$ is bounded from below on $\mathcal{Y}$. Since $\delta_n/\tau_n\to 0$ and $\delta_n/\alpha_n\to\infty$, we get $g_n\to\infty$ so that

$$\inf_{\tilde{\phi}\in\Phi}\sup_{y\in\mathcal{Y}}\left|\Lambda_n(\tilde{\phi}\,|\,y)\right|\sim\inf_{\tilde{\phi}\in\Phi}\left|\omega_n(\tilde{\phi})\right|,$$

as $n\to\infty$. Let us now consider

$$d_{\delta_n,\cdot}(\tilde{\phi})=\frac{\exp(\tilde{f}_{\delta_n,\cdot}(\tilde{\phi}))-1}{\exp(f_{\delta_n,\cdot})-1},$$

so that $\omega_n(\tilde{\phi})=d_{\delta_n,\alpha_n}(\tilde{\phi})/d_{\delta_n,\tau_n}(\tilde{\phi})-1$. Let us then remark that $|\exp(u)-1|\leq 2\,|u|$ for any $|u|\leq\log 2$ implies

$$\begin{aligned}
\inf_{\tilde{\phi}\in\Phi}\left|\exp(\tilde{f}_{\delta_n,\cdot}(\tilde{\phi})-f_{\delta_n,\cdot})-1\right|&\leq\inf_{\substack{\tilde{\phi}\in\Phi\\ |\tilde{f}_{\delta_n,\cdot}(\tilde{\phi})-f_{\delta_n,\cdot}|\leq\log 2}}\left|\exp(\tilde{f}_{\delta_n,\cdot}(\tilde{\phi})-f_{\delta_n,\cdot})-1\right|\\
&\leq 2\inf_{\substack{\tilde{\phi}\in\Phi\\ |\tilde{f}_{\delta_n,\cdot}(\tilde{\phi})-f_{\delta_n,\cdot}|\leq\log 2}}\left|\tilde{f}_{\delta_n,\cdot}(\tilde{\phi})-f_{\delta_n,\cdot}\right|\\
&=:\eta_{\delta_n,\cdot}
\end{aligned}$$

with $\eta_{\delta_n,\cdot}\to 0$ as $n\to\infty$ from Theorem 3.3.1. As a consequence, one has

$$\inf_{\tilde{\phi}\in\Phi}\left|d_{\delta_n,\cdot}(\tilde{\phi})-1\right|=\inf_{\tilde{\phi}\in\Phi}\left|\frac{\exp(f_{\delta_n,\cdot})\left(\exp\left(\tilde{f}_{\delta_n,\cdot}(\tilde{\phi})-f_{\delta_n,\cdot}\right)-1\right)}{\exp(f_{\delta_n,\cdot})-1}\right|\leq\frac{\eta_{\delta_n,\cdot}}{1-\exp(-f_{\delta_n,\cdot})}.$$

Now, $f_{\delta_n,\alpha_n}\to\infty$ and $f_{\delta_n,\tau_n}\to-\infty$ since $\delta_n/\alpha_n\to\infty$ and $\delta_n/\tau_n\to 0$ as $n\to\infty$ which entails that

$$\begin{aligned}
\inf_{\tilde{\phi}\in\Phi}\left|d_{\delta_n,\alpha_n}(\tilde{\phi})-1\right|&=\mathcal{O}(\eta_{\delta_n,\alpha_n}),\\
\inf_{\tilde{\phi}\in\Phi}\left|d_{\delta_n,\tau_n}(\tilde{\phi})-1\right|&=\mathcal{O}\left(\eta_{\delta_n,\tau_n}\exp(f_{\delta_n,\tau_n})\right).
\end{aligned}$$

Besides, applying twice the triangle inequality yields

$$\left|\frac{d_{\delta_n,\alpha_n}(\tilde{\phi})}{d_{\delta_n,\tau_n}(\tilde{\phi})} - 1\right| \leq \frac{|d_{\delta_n,\alpha_n}(\tilde{\phi}) - 1|}{1 - |d_{\delta_n,\tau_n}(\tilde{\phi}) - 1|} + \frac{|d_{\delta_n,\tau_n}(\tilde{\phi}) - 1|}{1 - |d_{\delta_n,\tau_n}(\tilde{\phi}) - 1|}$$

and therefore

$$\begin{aligned}\inf_{\tilde{\phi}\in\Phi} \left|\omega_n(\tilde{\phi})\right| &= \mathcal{O}(\eta_{\delta_n,\alpha_n}) + \mathcal{O}\left(\eta_{\delta_n,\tau_n} \exp(f_{\delta_n,\tau_n})\right) \\ &= \mathcal{O}(\alpha_n^{-\bar{\rho}_J}) + \mathcal{O}(\tau_n^{-\bar{\rho}_J-\gamma}\delta_n^{\gamma} L_Z(1/\tau_n)/L_Z(1/\delta_n)),\end{aligned}$$

from Theorem 3.3.1. Since $\bar{\rho}_2 < 0$, one can show using Karamata's representation [52, Equation (B.1.9)] that the slowly-varying function L_Z tends to a constant at infinity, so that

$$\inf_{\tilde{\phi}\in\Phi} \left|\omega_n(\tilde{\phi})\right| = \mathcal{O}(\alpha_n^{-\bar{\rho}_J}) + \mathcal{O}(\tau_n^{-\bar{\rho}_J-\gamma}\delta_n^{\gamma}),$$

which, in turn, implies that

$$\inf_{\tilde{\phi}\in\Phi} \sup_{y\in\mathcal{Y}} \left|\Lambda_n(\tilde{\phi} \mid y)\right| \to 0$$

as $n \to \infty$. All in all, and taking account of $|\log(1+u)| \leq 2\,|u|$ for any $|u| \leq 1/2$, one has in view of (3.B.16):

$$\begin{aligned}&\inf_{\tilde{\phi}\in\Phi} \sup_{y\in\mathcal{Y}} \left|\log q(1-\alpha_n \mid y) - \log \tilde{q}_{\tilde{\phi}}^{\texttt{NN}_J}(1-\alpha_n; 1-\delta_n, 1-\tau_n \mid y)\right| \\ &\quad\leq 2 \inf_{\tilde{\phi}\in\Phi} \sup_{y\in\mathcal{Y}} \left|\Lambda_n(\tilde{\phi}, y)\right| \\ &\quad\leq 3 \inf_{\tilde{\phi}\in\Phi} \left|\omega_n(\tilde{\phi})\right| \\ &\quad= \mathcal{O}(\alpha_n^{-\bar{\rho}_J}) + \mathcal{O}(\tau_n^{-\bar{\rho}_J-\gamma}\delta_n^{\gamma}),\end{aligned}$$

which proves the result. □

Proof of Corollary 3.5.1. (i) If $\gamma + \bar{\rho}_J > 0$, balancing the two terms in Theorem 3.5.2 yields

$$\delta_n = \alpha_n^{-\bar{\rho}_J/\gamma} \tau_n^{1+\bar{\rho}_J/\gamma}.$$

One can then check that:

$$\begin{aligned}\delta_n/\tau_n &= (\alpha_n/\tau_n)^{-\bar{\rho}_J/\gamma} \to 0, \\ \delta_n/\alpha_n &= (\tau_n/\alpha_n)^{1+\bar{\rho}_J/\gamma} \to \infty,\end{aligned}$$

since $\alpha_n/\tau_n \to 0$ as $n \to \infty$.
(ii) If $\gamma + \bar{\rho}_J \leq 0$, then, necessarily $\alpha_n^{-\bar{\rho}_J} = o(\tau_n^{-\bar{\rho}_J-\gamma}\delta_n^{\gamma})$. Therefore, letting $\delta_n = \xi_n\alpha_n$ and $\tau_n = \xi_n^2\alpha_n$ with $\xi_n \to \infty$ as $n \to \infty$ proves the result. □

3.C Generalized regular variation

3.C.1 Second-order condition

Table 3.3 provides the parameters associated with the second-order condition (3.2.6) for some classical heavy-tailed distributions. Note that the case $\gamma = 1$ in the Fréchet distribution and GPD coincides respectively with the Inverse Gamma and Burr distributions.

Distribution (parameters)	Density function	γ	ρ_2
Generalized Pareto ($\xi > 0$)	$(1+\xi t)^{-1-1/\xi}$, $t > 0$	ξ	$-\xi$
Burr ($\zeta, \theta > 0$)	$\zeta\theta t^{\zeta-1}\left(1+t^{\zeta}\right)^{-\theta-1}$, $t > 0$	$1/(\zeta\theta)$	$-1/\theta$
Fisher ($\nu_1, \nu_2 > 0$)	$\frac{(\nu_1/\nu_2)^{\nu_1/2}}{B(\nu_1/2,\nu_2/2)} t^{\nu_1/2-1}\left(1+\frac{\nu_1}{\nu_2}t\right)^{-(\nu_1+\nu_2)/2}$, $t > 0$	$2/\nu_2$	$-2/\nu_2$
Inverse Gamma ($\zeta > 0$)	$\frac{1}{\Gamma(\zeta)} t^{-\zeta-1}\exp(-1/t)$, $t > 0$	$1/\zeta$	$-1/\zeta$
Student ($\nu > 0$)	$\frac{1}{\sqrt{\nu}B\left(\nu/2,1/2\right)}\left(1+\frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$	$1/\nu$	$-2/\nu$

Table 3.3: Examples of heavy-tailed distributions satisfying the second-order condition (3.2.6) with the associated values of γ and ρ_2. Here, $\Gamma(\cdot)$ and $B(\cdot,\cdot)$ denote respectively the Gamma and Beta functions.

3.C.2 Third-order condition

The third order regular variation property [92] states that, in addition to the second-order condition, there exist $\rho_3 \neq 0$ and a function A_3 positive or negative with $A_3(t) \to 0$ as $t \to \infty$ such that for all $y \geq 1$:

$$\frac{1}{A_3(t)}\left(\frac{1}{A_2(t)}\left(\log U(yt) - \log U(t) - \gamma\log y\right) - \int_1^y y_2^{\rho_2-1}\,\mathrm{d}y_2\right) \to \int_1^y y_2^{\rho_2-1}\int_1^{y_2} y_3^{\rho_3-1}\,\mathrm{d}y_3\,\mathrm{d}y_2,$$

as $t \to \infty$. Moreover, $|A_3|$ is regularly-varying with index ρ_3 at infinity. Note that, if $\rho_2 < 0$ and $\rho_3 < 0$ then,

$$\int_1^y y_2^{\rho_2-1}\int_1^{y_2} y_3^{\rho_3-1}\,\mathrm{d}y_3\,\mathrm{d}y_2 = \frac{1}{\rho_3}\left(\frac{y^{\rho_2+\rho_3}-1}{\rho_2+\rho_3} - \frac{y_2^{\rho_2}-1}{\rho_2}\right).$$

3.C.3 Fourth-order condition

The fourth order regular variation property [92] states that, in addition to the third-order condition, there exist $\rho_4 \neq 0$ and a function A_4 positive or negative with $A_4(t) \to 0$ as $t \to \infty$ such that for all $y \geq 1$:

$$\begin{aligned}\frac{1}{A_4(t)}&\left(\frac{1}{A_3(t)}\left(\frac{1}{A_2(t)}\left(\log U(yt) - \log U(t) - \gamma\log y\right) - \int_1^y y_2^{\rho_2-1}\,\mathrm{d}y_2\right) - \int_1^{y_2} y_3^{\rho_3-1}\,\mathrm{d}y_3\right)\\ &\to \int_1^y y_2^{\rho_2-1}\int_1^{y_2} y_3^{\rho_3-1}\int_1^{y_3} y_4^{\rho_4-1}\,\mathrm{d}y_4\,\mathrm{d}y_3\,\mathrm{d}y_2,\end{aligned}$$

as $t \to \infty$. Moreover, $|A_4|$ is regularly-varying with index ρ_4 at infinity. Note that, if $\rho_2 < 0, \rho_3 < 0$ and $\rho_4 < 0$ then,

$$\begin{aligned}&\int_1^y y_2^{\rho_2-1}\int_1^{y_2} y_3^{\rho_3-1}\int_1^{y_3} y_4^{\rho_4-1}\,\mathrm{d}y_4\,\mathrm{d}y_3\,\mathrm{d}y_2\\ &\quad= \frac{1}{\rho_4}\left(\frac{1}{\rho_3+\rho_4}\left(\frac{y^{\rho_2+\rho_3+\rho_4}-1}{\rho_2+\rho_3+\rho_4} - \frac{y_2^{\rho_2}-1}{\rho_2}\right) - \frac{1}{\rho_3}\left(\frac{y^{\rho_2+\rho_3}-1}{\rho_2+\rho_3} - \frac{y_2^{\rho_2}-1}{\rho_2}\right)\right).\end{aligned}$$

Part II

Dictionary learning

Chapter 4

Structured Dictionary Learning of Rating Migration Matrices for Credit Risk Modeling

Note. The results of this chapter are based on the paper [8]

Abstract. Rating Migration Matrix is a crux to assess credit risks. Modeling and predicting these matrices are then an issue of great importance for risk managers in any financial institution. As a challenger to usual parametric modeling approaches, we propose a new structured dictionary learning model with auto-regressive regularization that is able to meet key expectations and constraints: small amount of data, fast evolution in time of these matrices, economic interpretability of the calibrated model. To show the model applicability, we present a numerical test with real data and a comparison study with the widely used parametric Gaussian Copula model: it turns out that our new approach based on dictionary learning significantly outperforms the Gaussian Copula model. The source code and the data are available at `https://github.com/michael-allouche/dictionary-learning-RMM.git` for the sake of reproducibility of our research.

4.1 Introduction

4.1.1 Banking context

Credit risk refers to the risk of incurring losses due to unexpected changes in the credit quality of the counterparty. Such a risk is summarized in a structured rating migration matrix which captures all possible transition probabilities that an obligor will migrate from a credit state to another over a given time period (see Figure 4.1). According to the financial regulation guidelines (Basel II and III), banks can use internal ratings and risk exposure estimations in order to assess regulatory capital requirement and credit risk measures (VaR, ES, ...). See [11, 37, 141, 25] for extensive references on risk measures and credit risk. Rating migration matrices (RMM) are key indicators to assess credit risk portfolio through the estimation of the credit quality of the obligors. Rating allocation process includes models and expert systems taking into account obligor's idiosyncratic features evolving over time given the economic situation. Observed migration frequencies are displayed into RMM that are the cornerstone of rating migration models upon which credit risk portfolio simulation relies. The most widely used method for modeling RMM is the one factor Gaussian Copula (GC) model [127] which assumes that a single factor represents the underlying systemic credit quality in the economy and defines a stationary economic cycle. See [27] among others for estimating risk measures on the loss

distribution of a large credit risk portfolio under this model. The popularity of the GC model is due to the ease of use but it also suffers from too simple underlying hypothesis. These weak assumptions lead to miscapture the dependence structure in tails. However, and despite post subprime crisis criticisms [129], the one factor GC model remains very popular in the banking industry because of its parsimony and of its ability to generate intuitive and interpretable results. The aim of this work is to derive from the data a non parametric representation of RMM, as an alternative and a challenger to the parametric GC model. This work is devoted to the design of a new methodology with thorough tests and a comparison with the GC model.

	1	2	3	4	5	6	7	8	9	10	11
1	71.48	17.87	5.36	2.38	1.25	0.71	0.43	0.26	0.15	0.08	0.03
2	16.01	57.62	14.41	5.69	2.80	1.54	0.89	0.52	0.30	0.16	0.06
3	5.05	15.14	53.85	13.46	5.89	3.03	1.68	0.96	0.54	0.28	0.11
4	2.45	6.52	14.67	51.35	12.84	5.87	3.06	1.68	0.92	0.47	0.18
5	1.45	3.61	7.23	14.46	49.57	12.39	5.74	2.95	1.55	0.76	0.30
6	0.96	2.31	4.34	7.71	14.46	48.19	12.05	5.51	2.71	1.29	0.48
7	0.70	1.63	2.94	4.90	8.16	14.69	47.00	11.75	5.14	2.28	0.82
8	0.54	1.24	2.18	3.48	5.44	8.71	15.24	45.71	11.43	4.51	1.52
9	0.45	1.02	1.75	2.73	4.09	6.14	9.55	16.38	43.67	10.92	3.28
10	0.43	0.95	1.60	2.43	3.55	5.11	7.46	11.36	19.17	38.35	9.59

Figure 4.1: Representation of an idealized rating migration matrix of size 10×11. All values are in percentage. The credit quality goes from the highest (rating 1) to the lowest (rating 10), the default is 11.

4.1.2 Matrix Factorization for RMM

Let us start from the data. In practice, we observe at time t a one-year rating migration matrix $\mathbf{P}^t \in \mathbb{R}^{R-1} \otimes \mathbb{R}^R$, which encodes the probability of migrating from rating $i = 1, \ldots, R-1$ to rating $j = 1, \ldots, R$ within one year period starting at time $t-1$; in Figure 4.1 we have $R = 11$. The reconstruction of this matrix is made empirically by evaluating the frequencies of obligors going from the rating i to rating j between times $t-1$ and t. It is important for risk management purposes to model the evolution of $\mathbf{P}^t$, by finding a representation of the type

$$\mathbf{P}^t \approx \sum_{k=1}^{K} \alpha_k^t \mathbf{d}_k, \quad \forall t \geq 1, \tag{4.1.1}$$

for some so-called (deterministic) basis vectors $\mathbf{d}_k$ and for some scalar random coefficients α_k^t which we should model the evolution. In a matrix form (using the vec () operator to simplify, see Section 4.1.5), we say that the collection of vectorized matrices $\mathbf{P} = \{\text{vec}\left(\mathbf{P}^t\right) \in \mathbb{R}^d\}_{t=1}^T \in \mathbb{R}^d \otimes \mathbb{R}^T$, with

$$d := (R-1)R$$

for all $t = 1, \ldots, T$, admits a matrix factorization over a dictionary $\mathbf{D} \in \mathbb{R}^d \otimes \mathbb{R}^K$ composed by K elements (called atoms), if there exists a linear combination of atoms weighted by coefficients (called codings) $\mathbf{A} = \{\boldsymbol{\alpha}^t \in \mathbb{R}^K\}_{t=1}^T \in \mathbb{R}^K \otimes \mathbb{R}^T$ such that

$$\mathbf{P} \approx \mathbf{DA}. \tag{4.1.2}$$

4.1.3 Objective

In this work, the objective is to achieve (4.1.2) while requiring

- $\mathbf{D}$ to satisfy some linear constraints (see Section 4.1.5) in order to represent economically interpretable RMM,
- the time series of elements $\boldsymbol{\alpha}^t$ of $\mathbf{A}$ to be smooth enough in order to perform predictions through a time series modeling,
- consider a dimensionality reduction framework $K \ll d$ in order to work in a lower dimensional space with extracted meaningful information.

However, the RMM evolution may vary quickly over time and a limited data history is available (usually 10-20 years $\approx$ 200 observations) which is close to the dimension of the problem (usually $R = 11$ and $d = 110$). Thus, modeling constrained RMM in a data-based non-parametric way presents an important challenge, which has not been addressed so far to the best of our knowledge.

4.1.4 State of the art

Over the last years, a new paradigm of data-based models have emerged in the Machine Learning (ML) community in order to extract structured information from high-dimensional objects. A classical approach in ML is to use Matrix Factorization techniques in order to project the data in some relevant basis. It is well known that the optimal basis that minimizes the linear approximation error is the Karhunen-Loève basis [137, Theorem 9.8], also known as the principal components in the principal component analysis (PCA).

Introduced in [150], dictionary learning (DL), see [59] for an overview and [101] for theoretical results, is another matrix representation technique where the basis, called dictionary, is learned from the observations. Unlike in the PCA decomposition, neither the orthogonality nor the representation constraints of the basis vectors (atoms) are imposed, allowing more flexibility to adapt the desired representation to the data. Moreover, compared with a predefined dictionary like Gabor functions, wavelets or local cosine vectors [137], learning a dictionary adapted to the observations has shown better results in practice [62, 134].

In DL, the linear approximation (4.1.2) is usually coupled with a regularization criterion $\mathcal{R}(\mathbf{A})$ applied to the codings and yields to the general optimization problem

$$\min_{\mathbf{D},\mathbf{A}} \|\mathbf{P} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda \mathcal{R}(\mathbf{A}), \quad \lambda \geq 0, \tag{4.1.3}$$

where the regularization term shall reflect the expected codings representation, see [59, Chapter 4]. The most widely studied regularization is $\mathcal{R}(\mathbf{A}) = \|A\|_1$ referring to the so-called sparse coding (see [132] for an overview), where the optimization with respect to A is known as basis pursuit [36] or the Lasso [176]. DL with sparse representation was notably studied in image and video processing [133, 135, 136], in graph learning [179] and in clustering [171]. In the case of spatial data, and more precisely in image processing, Total Variation (TV) plays an important role. In one dimension we have $\mathcal{R}_{TV}(\mathbf{A}) = \sum_{t=1}^{T-1} \sum_{k=1}^{K} \left|\alpha_k^{t+1} - \alpha_k^t\right|$, which is the integral of the absolute value of the gradient [158]. The intuition of this type of regularization in images is to allow a smooth transition between close codings and can be understood as a prior in a Bayesian model, see [34].

Here, we rather focus on DL with a temporal structure. This application has been mainly studied in video denoising [136, 156] where the temporal structure is exploited through an operator extracting patches of a fixed size in the objective function representing an energy minimization procedure. Another approach is to deal with an auto-regressive (AR) representation modeled either in the dictionary [38] or in the codings [191]. In the former, a mixed audio signal

is decomposed into its constituent temporal sources (atoms of the dictionary) in order to detect the presence of a specific sound. In the latter, the authors present a framework which supports data-driven temporal learning and forecasting through an AR modelization of the codings represented as a regularization term. Our model described in Section 4.2 is inspired from this problem formulation.

Our main and original contributions are to

- propose a new RMM modelization technique using DL approach
- derive a DL solution with linear constraints and a temporal regularization term for both interpretable clustering and prediction of RMM
- retrieve an economic health indicator on real data.

The paper is organized as follows. We introduce our proposed model and the associated optimization procedure in Section 4.2. Then we provide a numerical study on real data in Section 4.3 with two applications: prediction and clustering of RMM with economic interpretations.

4.1.5 Notations and data constraints

Notations.

Let $\mathbf{M} \in \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2}$ be a matrix with d_1 rows and d_2 columns. We use the notation for the i-th row $\mathbf{M}_{i,:}$, for the j-th column $\mathbf{M}_{:,j} := \mathbf{m}_j$ and for the sum column-wise

$$M_{i,\geq j} := M_{i,j} + \cdots + M_{i,d_2}.$$

The vectorization operator $\mathrm{vec}\,(\cdot)$ and its inverse $\mathrm{vec}^{-1}\,(\cdot)$ are defined as column-major order, *i.e.*

$$\begin{aligned}&\mathrm{vec}^{-1}\mathrm{vec}\,(\mathbf{M}) :=\\ &\quad \mathrm{vec}^{-1}\left([M_{1,1}, M_{2,1}, \ldots, M_{d_1,d_2}]^\top \in \mathbb{R}^{d_1 d_2}\right) = \mathbf{M}.\end{aligned}$$

The Frobenius norm is defined by:

$$\|\mathbf{M}\|_F = \sqrt{\mathrm{Tr}(\mathbf{M}\mathbf{M}^\top)} = \sqrt{\sum_{i\in[d_1], j\in[d_2]} M_{i,j}^2},$$

where $[d_i] := \{1, ..., d_i, \}$ for $i \in \{1, 2\}$. The orthogonal linear projection of a vector $\mathbf{u} \in \mathbb{R}^u$ onto the space generated by the columns of a matrix $\mathbf{V} \in \mathbb{R}^{v_1} \otimes \mathbb{R}^{v_2}$, with $u = v_1$, is denoted by $\mathrm{Proj}_{\mathbf{V}}\,(\mathbf{u})$. For a matrix $\mathbf{U} \in \mathbb{R}^{u_1} \otimes \mathbb{R}^{u_2}$, $(u_1 = v_1)$ with columns $\{\mathbf{U}_{:,1}, ..., \mathbf{U}_{:,u_2}\}$, the projection is defined by

$$\begin{aligned}&\mathrm{Proj}_{\mathbf{V}}\,(\mathbf{U}) \in \mathbb{R}^{v_2} \otimes \mathbb{R}^{u_2},\\ &\quad \text{with columns } \{\mathrm{Proj}_{\mathbf{V}}\,(\mathbf{U}_{:,1}), ..., \mathrm{Proj}_{\mathbf{V}}\,(\mathbf{U}_{:,u_2})\}.\end{aligned}$$

Data constraints

A rating migration matrix $\mathbf{M}$ must satisfy some constraints for both mathematical and economical reasons.

Mathematical constraints. Each row of $\mathbf{M}$ is a discrete probability, hence $\mathbf{M}$ is a stochastic matrix. The set of Stochastic Matrices is denoted by

$$\begin{aligned}\mathcal{M}^S := \Big\{&\mathbf{M} \in \mathbb{R}^{(R-1)} \otimes \mathbb{R}^R : \sum_{j\in[R]} M_{i,j} = 1, \forall i \in [R-1],\\ &M_{i,j} \geq 0, \forall (i,j) \in [R-1] \times [R]\Big\}.\end{aligned}$$

Economic constraints. Depending on their expertise, some risk managers may consider important to put additional constraints that are meaningful from economic point-of-view. For instance, the likelihood of default for higher-rated counterparties is lower than for the lower-quality ones. Then, the collection of rating matrices satisfying so-called economic constraints is denoted by

$$\mathcal{M}^E := \Big\{ \mathbf{M} \in \mathbb{R}^{(R-1)} \otimes \mathbb{R}^R : \; M_{i,\geq j} \leq M_{i',\geq j}, \\ \forall j \in [R], \quad 1 \leq i < i' \leq R-1 \Big\}.$$

A matrix satisfying such constraints is called an idealized matrix and is illustrated in Figure 4.1.

4.2 Dictionary learning: modeling and solving

4.2.1 Defining the regularization term

In the case of time-series DL, the expected time dependency can be encoded in the regularization part. We propose a regularization term with an extra parameter $\mathbf{w}$ that will be used to infer the behavior of the codings as a time-series.

Defining

$$\bar{\alpha}_k := \frac{1}{T} \sum_{t=1}^{T} \alpha_k^t,$$

the proposed DL problem is:

$$\min_{\substack{\mathbf{D},\mathbf{A},\mathbf{w} \\ \mathbf{D}\in\Omega,\ \alpha_k^t\geq 0,\ t\in[T],\ k\in[K]}} \|\mathbf{P} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda \mathcal{R}_{AR}(\mathbf{A}, \mathbf{w}) \tag{4.2.1}$$

with regularization:

$$\mathcal{R}_{AR}(\mathbf{A}, \mathbf{w}) := \sum_{k=1}^{K} \sum_{t=1}^{T-1} \left(\alpha_k^{t+1} - \bar{\alpha}_k - w_k(\alpha_k^t - \bar{\alpha}_k) \right)^2, \tag{4.2.2}$$

where the extra parameter $\mathbf{w}$ allows us to estimate the AR parameters of the time-series $\boldsymbol{\alpha}_k$ for each $k \in [K]$, as it will be detailed below. The available set Ω is the convex set of dictionaries verifying the idealized constraints (see Section 4.1.5)

$$\Omega := \left\{ \mathbf{D} \in \mathbb{R}^d \otimes \mathbb{R}^K : \mathrm{vec}^{-1}(\mathbf{d}_k) \in \mathcal{M}^E \cap \mathcal{M}^S, \ \forall k \in [K] \right\}.$$

Heuristics for the regularization strategy The AR model is an important time-series structure, largely applied in finance and other contexts (see [146]). For a fixed $k \in [K]$, we say that the time-series $\boldsymbol{\alpha}_k$ is auto-regressive of order 1, if

$$\alpha_k^{t+1} = \mu_k + w_k \alpha_k^t + \epsilon_k^t, \quad \text{for all } t > 1, \tag{4.2.3}$$

where μ_k is a constant called drift, w_k is the AR coefficient and $(\epsilon_k^t)_{t=1}^T$ are independent centered Gaussian variables with some variance parameter σ_k^2.

Starting from the DL model (4.1.3), we encourage the codings $\mathbf{A}$ to have an AR structure (4.2.3) through the regularization term. Thus, assuming an AR structure of $\boldsymbol{\alpha}_k$ for each $k \in [K]$, the log-likelihood with respect to parameters μ_k, w_k, and σ_k, up to a constant term, is:

$$\ell(\boldsymbol{\alpha}_k, \mu_k, w_k, \sigma_k) := -\frac{1}{2\sigma_k^2} \sum_{t=1}^{T-1} \left(\alpha_k^{t+1} - \mu_k - w_k \alpha_k^t \right)^2 - (T-1)\log(\sigma_k)$$

(see [167, Chapter 3.6]), with solutions:

$$\tilde{\mu}_k, \tilde{w}_k, \tilde{\sigma}_k = \underset{\mu_k, w_k, \sigma_k}{\arg\max}\, \ell(\boldsymbol{\alpha}_k, \mu_k, w_k, \sigma_k).$$

It readily follows that the optimal parameter μ_k is

$$\tilde{\mu}_k = \frac{1}{T-1}\sum_{t=1}^{T-1}(\alpha_k^{t+1} - \tilde{w}_k\alpha_k^t) \approx (1-\tilde{w}_k)\bar{\boldsymbol{\alpha}}_k =: \widehat{\mu}_k$$

where the approximation holds for large values of T. Thus, the optimization of w_k boils down to (up to a small error)

$$\widehat{w}_k = \underset{w_k}{\arg\min}\sum_{t=1}^{T-1}\Big(\alpha_k^{t+1} - \bar{\alpha}_k - w_k(\alpha_k^t - \bar{\alpha}_k)\Big)^2. \tag{4.2.4}$$

Doing so, we obtain the regularization term of (4.2.2).

Later (in the hyper-parameter selection step in Section 4.3.1), we will need also to retrieve all AR coefficients from observations. Obtaining the optimal w_k is straightforward from (4.2.4):

$$\widehat{w}_k = \frac{\sum_{t=1}^{T-1}(\alpha_k^{t+1} - \bar{\boldsymbol{\alpha}}_k)(\alpha_k^t - \bar{\boldsymbol{\alpha}}_k)}{\sum_{t=1}^{T}(\alpha_k^t - \bar{\boldsymbol{\alpha}}_k)^2}. \tag{4.2.5}$$

Regarding σ_k, we proceed similarly and we get that $\tilde{\sigma}_k$ is close to

$$\widehat{\sigma}_k^2 = \frac{\sum_{t=1}^{T-1}\Big(\alpha_k^{t+1} - \widehat{\mu}_k - \widehat{w}_k\alpha_k^t\Big)^2}{T-1}.$$

We observe that, replacing $\widehat{w}_k$ in (4.2.4) (and then in (4.2.1)) would result in a non-convex function in terms of $\boldsymbol{\alpha}_k$, which would increase the difficulty of the optimization problem (4.2.1). Therefore, optimize this parameter w_k as an extra variable is the best choice regarding convexity purposes.

Remark 6. The model presented in this work can be easily generalized to an AR model of order $p \in \mathbf{N}$. The choice to introduce it in order 1 simplifies our notation and is adequate to our case of application, see Section 4.3.1. The DL optimization strategy, Section 4.2.2, applies likewise to an AR model of order $p \in \mathbf{N}$.

The importance of the parameter μ_k A similar AR regularization, inspired in graph theory, is proposed in [191]. The difference between the latter and our model is that their μ_k is considered to be zero. We discuss in this paragraph why in our case of application this choice would not work.

Indeed, because of Equation (4.1.1) and the fact that Ω is a convex set, we expect $\{\alpha_k^t\}_{k=1}^K$ to be coefficients of a linear combination of $\{\mathbf{d}_k\}_{k=1}^K$ that approximates $\mathbf{P}^t \in \mathcal{M}^S$ for each t fixed, as explained in Equation (4.1.1). Then:

$$\sum_{k=1}^{K}\alpha_k^t \approx 1, \quad \text{for all } t \in [T]. \tag{4.2.6}$$

On the other hand, if $\mu_k = 0$, and $\boldsymbol{\alpha}_k$ are AR time-series of order 1 with coefficients μ_k and w_k , the estimator $\widehat{\mu}_k$ gives that:

$$0 \approx \widehat{\mu}_k = (1-w_k)\bar{\boldsymbol{\alpha}}_k.$$

Either $w_k = 1$, which restricts a lot the possible time-modeling of $\boldsymbol{\alpha}_k$. Or, if $w_k \neq 1$ for all $k \in [K]$, then $\frac{1}{T}\sum_{t=1}^{T}\alpha_k^t \approx 0$ and summing in k, we get $\sum_{k=1}^{K}\sum_{t=1}^{T}\alpha_k^t \approx 0$, which contradicts (4.2.6). This contradiction shows the difficulty of fitting an AR model with drift 0 in the case where dictionaries lie in a convex set and where the codings are expected to be a convex combination.

4.2.2 Dictionary learning optimization strategy

Problem (4.2.1) is not a convex optimization problem, as it is usually the case in DL problems. Nevertheless, the problem is convex in variables $\mathbf{D}$, $\mathbf{A}$ and $\mathbf{w}$, as we can observe in (4.2.1). This property encourages the use of a policy that consists in alternating the minimization in $\mathbf{A}$, $\mathbf{D}$ and $\mathbf{w}$. This largely applied strategy does not ensure a global solution of problem (4.2.1), but it is a straightforward way of finding a local minima of problem.

The quadratic problems presented in this section are solved by a Interior Point Method, see ([148, Section 16.8],[187]). Interior point methods (IPMs) are very well-suited to solving quadratic optimization problems, particularly when sizes of problems grow large, see [97].

Dictionary update.

We opt for a sequential update of each atom of the dictionary: $\mathbf{d}_k$ for $k \in [K]$. This choice is guided by two advantages: 1. The problem is strictly convex for each atom $\mathbf{d}_k$ (as stated in Proposition 4.2.1 below) which is not necessarily true for the whole matrix $\mathbf{D}$. 2. This strategy breaks the problem in smaller problems making the resolution less dependent on the amount of atoms K. Updating atoms separately is also the strategy of the widely used K-SVD (see [59, Section 3.5]), however, the purpose in that case is to find a closed form for the optimization problem, which is not true in our case of study because of the form of constraints.

Proposition 4.2.1. *Assume that $\{\alpha_k^t\}_{t=1}^T$ is non zero. The minimization of (4.2.1) over $\mathbf{d}_k \in \operatorname{vec}\left(\mathcal{M}^E \cap \mathcal{M}^S\right)$ is equivalent to minimizing a strictly convex quadratic problem with linear constraints*

$$\min_{\mathbf{d}_k} \left\| \operatorname{vec}\left(\widetilde{\mathbf{P}}_k\right) - \widetilde{\mathbf{A}}_k \mathbf{d}_k \right\|_F^2, \quad s.t.\ vec^{-1}\left(\mathbf{d}_k\right) \in \mathcal{M}^E \cap \mathcal{M}^S, \tag{4.2.7}$$

where $\widetilde{\mathbf{P}}_k$ and $\widetilde{\mathbf{A}}_k$ are explicitly defined in (4.2.9).

Observe that the condition on α_k is expected to be systematically satisfied since each element α_k^t is non-negative.

Proof. Start from the reconstruction error in (4.2.1) and write

$$\|\mathbf{P} - \mathbf{D}\mathbf{A}\|_F^2 = \left\| \mathbf{P} - \sum_{j \neq k}^{K} \mathbf{d}_j \mathbf{A}_{j,:} - \mathbf{d}_k \mathbf{A}_{k,:} \right\|_F^2. \tag{4.2.8}$$

From this, it is obvious that the function $\mathbf{d}_k \mapsto \|\mathbf{P} - \mathbf{D}\mathbf{A}\|_F^2$ to minimize is quadratic and convex. However, under this form, it is not yet clear it is strictly convex. To establish this property, define

$$\begin{aligned} \widetilde{\mathbf{P}}_k &:= \mathbf{P} - \sum_{j \neq k}^{K} \mathbf{d}_j \mathbf{A}_{j,:}, \\ \widetilde{\mathbf{A}}_k &:= \begin{bmatrix} A_{k,1} & 0 & \dots & 0 \\ 0 & A_{k,1} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & A_{k,1} \\ \vdots & \vdots & \vdots & \vdots \\ A_{k,T} & 0 & \dots & 0 \\ 0 & A_{k,T} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & A_{k,T} \end{bmatrix} \in \mathbb{R}^{dT} \otimes \mathbb{R}^{d}. \end{aligned} \tag{4.2.9}$$

The quantity in (4.2.8) is thus equal to

$$\left\| \text{vec}\left(\widetilde{\mathbf{P}}_k\right) - \widetilde{\mathbf{A}}_k \mathbf{d}_k \right\|_F^2 .$$

Note that $\widetilde{\mathbf{A}}_k^\top \widetilde{\mathbf{A}}_k$ is diagonal matrix equal to $\sum_{t=1}^T (\alpha_k^t)^2 \mathcal{I}_{\mathbb{R}^d}$. □

Codings update

Similarly to the dictionary update, we adopt a strategy based on the update of each $\mathbf{A}_{k,:}$ for $k \in [K]$. The reasons are the same: it is preferable to solve a smaller and strictly convex optimization problem. The fact that the optimization for each k is a strongly convex problem is not straightforward and is argued in the proposition below.

Proposition 4.2.2. *Let $k \in [K]$ be fixed. Consider the minimization of (4.2.1)-(4.2.2) over one coding $\mathbf{A}_{k,:}$, i.e.*

$$\min_{\mathbf{A}_{k,:},\ A_{k,t}\geq 0} \|\mathbf{P} - \mathbf{D}\mathbf{A}\|_F^2 \tag{4.2.10}$$

$$+ \lambda \sum_{k=1}^{K} \sum_{t=1}^{T-1} \left(A_{k,t+1} - \bar{\mathbf{A}}_{k,:} - w_k (A_{k,t} - \bar{\mathbf{A}}_{k,:}) \right)^2 . \tag{4.2.11}$$

For any $\lambda \geq 0$, the above problem is a strongly convex quadratic optimization problem with linear constraints.

Proof. First, there is a symmetric non-negative matrix $\mathbf{H}^{w_k} \in \mathbb{R}^T \otimes \mathbb{R}^T$ such that

$$\begin{aligned} \mathcal{R}_{AR}^k(A_{k,:}, w_k) &= \sum_{t=1}^{T-1} \left(A_{k,t+1} - \bar{\mathbf{A}}_{k,:} - w_k (A_{k,t} - \bar{\mathbf{A}}_{k,:}) \right)^2 \\ &= \langle \mathbf{A}_{k,:}^\top, \mathbf{H}^{w_k} \mathbf{A}_{k,:}^\top \rangle \end{aligned}$$

since for w_k fixed, $\mathcal{R}_{AR}^k(\cdot, w_k)$ is a quadratic problem without linear term that can be represented by a symmetric matrix. Obviously, it is non-negative.

Similarly to the proof for the dictionary update, we define a matrix $\widetilde{\mathbf{D}}_k$:

$$\widetilde{\mathbf{D}}_k := \begin{bmatrix} \mathbf{D}_{1,k} & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{D}_{d,k} & 0 & \dots & \dots & 0 \\ 0 & \mathbf{D}_{1,k} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & \mathbf{D}_{d,k} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & \mathbf{D}_{1,k} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & \mathbf{D}_{d,k} \end{bmatrix} \in \mathbb{R}^{dT} \otimes \mathbb{R}^T ;$$

note that the minimization problem (4.2.10) is equivalent to

$$\min_{\mathbf{A}_{k,:},\ A_{k,t}\geq 0} \left\| \text{vec}\left(\widetilde{\mathbf{P}}_k\right) - \widetilde{\mathbf{D}}_k A_{k,:}^\top \right\|_F^2 + \lambda \langle \mathbf{A}_{k,:}^\top, \mathbf{H}^{w_k} \mathbf{A}_{k,:}^\top \rangle, \tag{4.2.12}$$

which is a quadratic constrained optimization problem with quadratic term given by the matrix:

$$\mathbf{C}_k := \widetilde{\mathbf{D}}_k^\top \widetilde{\mathbf{D}}_k + \lambda \mathbf{H}^{w_k}.$$

Since $\widetilde{\mathbf{D}}_k^\top \widetilde{\mathbf{D}}_k = \|\mathbf{d}_k\|_2^2 \mathcal{I}_{\mathbb{R}^T}$ and that $\|\mathbf{d}_k\|_2$ is uniformly bounded from below on $\text{vec}\left(\mathcal{M}^E \cap \mathcal{M}^S\right)$, and since $\mathbf{H}^{w_k}$ is symmetric non-negative, the matrix $\mathbf{C}_k$ is symmetric positive definite with a uniform lower bound for its eigenvalues. The announced statement is proved. □

4.2.3 Coefficient update

We note that, for each $k \in [K]$, the optimization problem with respect to $\mathbf{w}_k$ in equation (4.2.1) is a 1-dimensional quadratic problem with explicit solution given by Equation (4.2.5).

Remark 7. For each $k \in [K]$, the solution of problem (4.2.7) and (4.2.10) decreases the objective value of the respective optimization problems. However, it is an open question to justify that this strategy provides a solution to problems (4.2.1) with respect to $\mathbf{D}$, $\mathbf{A}$ and $\mathbf{w}$.

Algorithm 7: Dictionary Learning (DL)

Input: *matrix of vectorized RMM:* $\mathbf{P} \in \mathbb{R}^d \otimes \mathbb{R}^T$,
number of atoms: $K \in \{1, 2, ...\}$,
regularization parameter: $\lambda > 0$
number of iterations: $N \in \{1, 2, ...\}$

Output: *optimized dictionary, codings and drift:* $\mathbf{D}, \mathbf{A}, \mathbf{w}$

initialize $\mathbf{D} \in \mathbb{R}^d \otimes \mathbb{R}^K$ and $\mathbf{A} \in \mathbb{R}^K \otimes \mathbb{R}^T$
for $i = 1 : N$ **do**
  # Dictionary update
  for $k = 1 : K$ **do**
    update $\mathbf{d}_k$ with QP s.t. $\text{vec}^{-1}(\mathbf{d}_k) \in \mathcal{M}^E \cap \mathcal{M}^S$
  # Codings update
  for $k = 1 : K$ **do**
    update $\boldsymbol{\alpha}_k$ with QP s.t. $\alpha_k^t \geq 0,\ t \in [T]$
  # Coefficient update
  for $k = 1 : K$ **do**
    update w_k with Equation (4.2.5)

4.3 Experiments

Our proposed DL method with temporal AR regularization will be evaluated on real RMM provided by BNP Paribas. The dataset contains $T = 192$ one-year observed transition frequency matrices with shape $R = 11$ issued monthly from 52 sectors composed by large European capitalization companies between January 2004 and December 2019. This period contains in particular the subprime crisis but not the COVID-19 pandemic.

4.3.1 Experimental design

Ideally, we should apply our DL method on each sector. However given the dataset, the observed matrices are very sparse which refers to another problem formulation (missing data). To overcome this issue we computed a (confidential) weighted sum among the 52 sectors to form a shareable (on git) set of matrices $\mathbf{P} = \{\text{vec}\left(\mathbf{P}^t\right) \in \mathbb{R}^d\}_{t=1}^T$ with $d = 110$ and $T = 192$.

Those matrices are still noisy and so they might not respect the economic constraints, *i.e.* $\mathbf{P}^t \in \mathcal{M}^S, \forall t \in [192]$. Based on this dataset, we performed a classical 80/20 non-random train-test split in time, stored respectively in $\mathbf{P}^{\text{Train}} \in \mathbb{R}^d \otimes \mathbb{R}^{T^{\text{Train}}}$ and $\mathbf{P}^{\text{Test}} \in \mathbb{R}^d \otimes \mathbb{R}^{T^{\text{Test}}}$. In all the following experiments we use $K = 3$ for ease of interpretation since we want to represent the RMM as a combination of three regimes, assuming that each one represents an economic state. Larger values of K perform similar results but lead to a more sophisticated economic analysis.

AR lag estimation

First let us check the relevance of the AR(1) model in (4.2.2). Starting from the optimization problem (4.2.1) with $\lambda = 0$, we trained the DL model on $\mathbf{P}^{\text{Train}}$ and applied the well-known partial autocorrelation function (PACF) [28, Section 3.2.5] on the codings $\mathbf{A}^{\text{Train}}_{k,:}$, for all $k \in [3]$. For each series, we identified just one statistically significant lag, suggesting a possible AR(1) model adapted to the data. See Figure 4.2 for $k = 1$, while the others behave similarly.

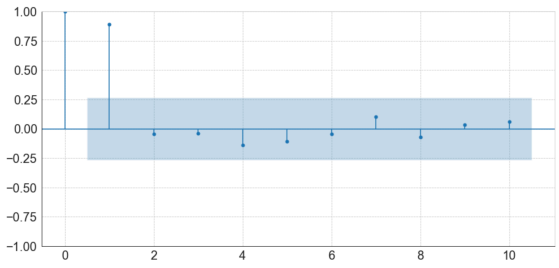


Figure 4.2: PACF of $\mathbf{A}^{\text{Train}}_{1,:}$ for the first 10 lags. The shaded region represents the 95% confidence interval.

Hyper-parameter selection

The best hyperparmeter λ is chosen automatically through the procedure described in Algorithm 8. The latter allows to evaluate both the prediction and the reconstruction capacity of the model. We applied iteratively Algorithm 8 for $\lambda \in \{0.01, 0.1, 1, 3, 5, 6, 7, 10\}$ and stored the results in Table 4.1 which highlights the benefit of our proposed regularization in the RMM predictions. The parameter associated with the smallest reconstruction error is $\lambda = 6$.

In the next section we illustrate the results of our DL model and propose two ML applications. First a time-series prediction of the RMM and second their unsupervised clustering in order to infer an estimation of the global economic sentiment.

4.3.2 Results

Computational aspects The numerical experiments have been conducted on a Macbook Pro (13-inch, M1, 2020), 512 Go SSD, 16 Go RAM. All the code was implemented in Python 3.10. It takes less than a minute to train the model with $K = 3$ during 500 iterations. Clearly from Algorithm 7 the training time is linear with respect to K.

Dictionary representation

Once learned, it appears that the dictionary managed to extract three candidates (atoms) to be good representatives for all RMM included in our dataset. In Figure 4.3 are represented these

Algorithm 8: Hyper-parameter selection

Input: *data train:* $\mathbf{P}^{\text{Train}} \in \mathbb{R}^d \otimes \mathbb{R}^{T^{\text{Train}}}$,
data test: $\mathbf{P}^{\text{Test}} \in \mathbb{R}^d \otimes \mathbb{R}^{T^{\text{Test}}}$,
number of atoms: $K \in \{1, 2, ...\}$,
lambda: $\lambda > 0$

Output: *reconstruction error:* $\mathcal{E}$

$\mathbf{D}^{\text{Train}}, \mathbf{A}^{\text{Train}}, \mathbf{w}^{\text{Train}} \leftarrow \text{DL}(\mathbf{P}^{\text{Train}}, K, \lambda, 500)$,
where the function DL refers to Algorithm 7

$\mathbf{A}^{\text{Test}} \leftarrow \text{Proj}_{\mathbf{D}^{\text{Train}}}\left(\mathbf{P}^{\text{Test}}\right)$

$A^{\text{Sim}}_{k,t} \leftarrow \widehat{\mu}_k + A^{\text{Test}}_{k,t} w^{\text{Train}}_k + \varepsilon^t_k$ with

$$\begin{aligned} \widehat{\mu}_k &= \bar{\alpha}^{\text{Train}}_k (1 - w^{\text{Train}}_k), \\ \varepsilon^t_k &\sim \mathcal{N}\left(0, \widehat{\sigma}^2_k\right), \\ \widehat{\sigma}^2_k &\leftarrow \widehat{\mathbb{V}\text{ar}}[\boldsymbol{\alpha}^{\text{Train}}_k]\left(1 - (w^{\text{Train}}_k)^2\right), \end{aligned}$$

for all $k \in [K]$ and $t \in [T^{\text{Test}} - 1]$

$\mathbf{P}^{\text{Reco}} \leftarrow \mathbf{D}^{\text{Train}} \mathbf{A}^{\text{Train}}$

$\mathbf{P}^{\text{Sim}} \leftarrow \mathbf{D}^{\text{Train}} \mathbf{A}^{\text{Sim}}$

$\mathcal{E} \leftarrow 0.8 \left\| \mathbf{P}^{\text{Test}}_{:,1:} - \mathbf{P}^{\text{Sim}} \right\|^2_F + 0.2 \left\| \mathbf{P}^{\text{Train}} - \mathbf{P}^{\text{Reco}} \right\|^2_F$
without the first test value

λ	0.01	0.1	1	3	5	6	7	10
error	6.1	5.1	4.7	4.6	4.5848	**4.5846**	4.5847	4.593

Table 4.1: Reconstruction error from Algorithm 8 associated with various λ. The best result is emphasized in bold.

atoms in a matrix form generating a stable (strong diagonal, see Figure 4.3a), an upgrade (strong lower diagonal, see Figure 4.3b) and a downgrade (strong upper diagonal, see Figure 4.3c) risk configuration. Although automatically obtained by our algorithm, observe that these representatives make fully sense in terms of economic interpretation, and should reveal the underlying characteristics of RMM in our data set.

Codings.

The AR regularization (4.2.2) enforces an AR behavior of the codings making them more regular. The choice of the best prediction and analysis will then be a trade-off between the reconstruction and the regularity of the time evolution of the codings. Figure 4.4 depicts this evolution. Note that for larger values of λ the evolution in time of the coding is smoother. This property tends to advantage the prediction of future matrices.

Clustering

Let study now the benefit of our proposed regularized DL model in order to obtain an interpretable classification of the RMM. We fit a KMeans algorithm on the standardized $\mathbf{A}^{\text{Train}}$ in 3 clusters and predict the classes of the standardized $\mathbf{A}^{\text{Train}}$. Assuming that the atoms and the clusters represent different economic states, we obtain in Figure 4.5 a classification in time of the observed RMM. Additionally to the historical financial context, we present how to infer an economic sentiment indicator based on both the codings' classification and the dictionary. To do so, we store in Table 4.2 the weights of the atoms assigned to each cluster. Thus, combining

	1	2	3	4	5	6	7	8	9	10	11
1	91.17	6.23	0.00	1.33	0.10	0.78	0.30	0.09	0.01	0.00	0.00
2	0.31	93.87	3.22	0.91	0.51	0.70	0.39	0.09	0.01	0.00	0.00
3	0.31	5.07	79.24	10.93	3.04	0.93	0.38	0.09	0.01	0.00	0.00
4	0.01	1.34	6.67	83.00	5.37	3.14	0.38	0.09	0.01	0.00	0.00
5	0.01	0.25	0.89	6.27	80.05	9.86	2.05	0.61	0.01	0.00	0.00
6	0.00	0.02	0.55	0.71	6.46	85.48	5.86	0.92	0.01	0.00	0.00
7	0.00	0.02	0.06	0.23	0.34	5.52	88.85	4.98	0.01	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.04	4.66	95.30	0.01	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	4.69	17.71	77.59	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.40	11.48	0.00	87.85	0.26

(a) $\text{vec}^{-1}(\mathbf{d}_1)$

	1	2	3	4	5	6	7	8	9	10	11
1	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	9.23	61.82	24.71	2.10	0.73	0.00	0.00	0.89	0.22	0.29	0.01
3	1.50	14.93	79.33	0.00	0.57	1.82	0.06	0.93	0.34	0.34	0.18
4	1.12	0.49	23.07	63.87	7.09	1.42	1.05	0.91	0.37	0.39	0.23
5	1.12	0.49	4.02	31.24	53.29	6.85	0.76	0.34	0.59	1.00	0.31
6	0.59	0.99	0.70	4.38	31.29	45.97	9.31	3.16	0.94	1.55	1.13
7	0.58	0.80	0.88	1.24	6.48	33.83	31.56	16.50	4.30	1.31	2.52
8	0.58	0.72	0.85	0.76	2.67	9.67	39.09	24.71	9.94	5.44	5.58
9	0.58	0.72	0.78	0.78	2.73	4.11	10.93	0.00	67.80	5.91	5.67
10	0.00	0.00	0.00	0.00	1.15	1.22	18.25	0.00	48.79	0.00	30.59

(b) $\text{vec}^{-1}(\mathbf{d}_2)$

	1	2	3	4	5	6	7	8	9	10	11
1	67.86	21.46	4.94	3.51	0.44	1.14	0.00	0.00	0.00	0.00	0.64
2	0.00	81.81	11.37	4.59	0.44	0.60	0.16	0.01	0.21	0.15	0.64
3	0.00	0.00	91.67	6.10	0.44	0.60	0.16	0.01	0.20	0.16	0.64
4	0.00	0.00	0.00	81.92	16.29	0.60	0.16	0.01	0.19	0.18	0.64
5	0.00	0.00	0.00	2.90	85.00	7.09	2.22	0.97	0.54	0.64	0.64
6	0.00	0.00	0.00	1.24	5.01	72.67	14.51	1.58	1.56	0.99	2.44
7	0.00	0.00	0.00	0.54	1.64	6.03	75.40	9.56	3.33	0.75	2.75
8	0.00	0.00	0.00	0.22	0.93	0.97	20.89	56.45	10.38	3.67	6.48
9	0.00	0.00	0.00	0.21	0.15	1.76	0.00	15.43	0.00	32.09	50.36
10	0.00	0.00	0.00	0.00	0.00	0.00	2.12	1.88	0.00	43.07	52.92

(c) $\text{vec}^{-1}(\mathbf{d}_3)$

Figure 4.3: Matrix representation of the atoms in a trained dictionary with $K = 3, \lambda = 6$.

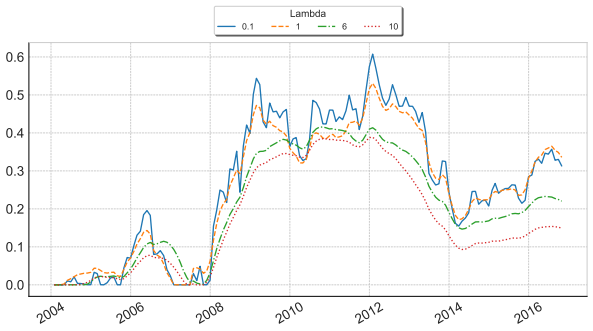


Figure 4.4: Time evolution of $\mathbf{A}^{\text{Train}}_{3,:}$ for $\lambda \in \{0.1, 1, 6, 10\}$.

with Figure 4.3, we can easily deduce that the labels associated to the clusters green, yellow, red are respectively a good, a stable and a bad economic sentiment indicator. Such an allocation can

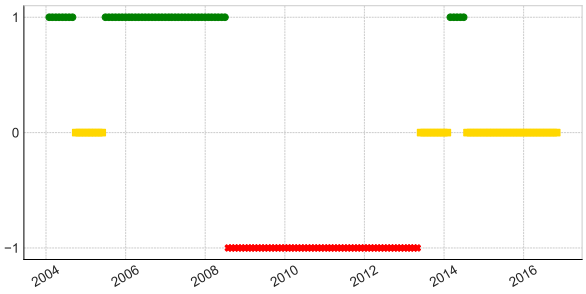


Figure 4.5: Unsupervised classification of the codings $\mathbf{A}^{\text{Train}}$ in 3 clusters: $\{-1$ (red cross), 0 (yellow square), 1 (green dot)$\}$.

be confirmed graphically in Figure 4.5 which captures effectively the financial bubble between 2006-2008, as well as the subrprime crisis.

cluster \ k	1	2	3
green	81.7	11.7	6.6
yellow	60.3	23.6	16.2
red	57.0	6.6	36.4

Table 4.2: Centroid of $\mathbf{A}^{\text{Train}}$ in each cluster: green, yellow, red. The values are adjusted in percentage.

4.4 Challenging the one-factor Gaussian Copula model

Let us compare our proposed non-parametric DL approach with the parametric one-factor GC model based on the same real RMMs as tested in the previous section.

4.4.1 One-factor Gaussian Copula model

For an initial rating i at time t, the one-factor Copula model assumes that the event "migration to the rating greater than j at time $t+1$" is given by

$$\left\{X_i^t \geq c_{i,j}\right\},$$

where the parameters $\{c_{i,j}\}_{i\in[R-1],j\in[R]}$ are thresholds triggering the rating migration and with a stochastic factor

$$X_i^t = \rho Z^t + \sqrt{1-\rho^2}\epsilon_i^t,$$

composed by a systemic risk factor Z^t (common to all obligors), an idiosyncratic risk factor $\epsilon_i^t \overset{d}{=} \mathcal{N}(0,1)$ (specific to every obligor), independent from Z^t, and a correlation parameter $\rho \in (-1,1)$ between the two sources of risk. The systemic factor Z evolves through time and has a stationary distribution given by a standard Gaussian distribution. Thus, conditionally to Z^t, this event "migration to the rating greater than j at time $t+1$" has the probability

$$\Phi\left(\frac{\rho Z^t - c_{i,j}}{\sqrt{1-\rho^2}}\right) =: P_{i,\geq j}^t, \tag{4.4.1}$$

and the unconditional probability is

$$\Phi(-c_{i,j}) =: P^{\mathrm{TTC}}_{i,\geq j}, \tag{4.4.2}$$

where Φ is the c.d.f. of a standard Gaussian distribution. The matrix $\mathbf{P}^{\mathrm{TTC}}$ represents the probability of default *'Through the Cycle'* which is a long-run average over a cycle and focuses mainly on permanent components of default risk, whereas the matrix $\mathbf{P}^t$ represents the probability of default *'Point-In-Time'* which takes into account both cyclical and permanent effect. Inverting $c_{i,j}$ in (4.4.2) and replacing in (4.4.1) gives

$$P^t_{i,\geq j} = \Phi\left(\frac{\rho Z^t + \Phi^{-1}(P^{\mathrm{TTC}}_{i,\geq j})}{\sqrt{1-\rho^2}}\right), \tag{4.4.3}$$

$$P^t_{i,j} := \Phi\left(\frac{\rho Z^t + \Phi^{-1}(P^{\mathrm{TTC}}_{i,\geq j})}{\sqrt{1-\rho^2}}\right) - \Phi\left(\frac{\rho Z^t + \Phi^{-1}(P^{\mathrm{TTC}}_{i,\geq j+1})}{\sqrt{1-\rho^2}}\right), \tag{4.4.4}$$

with the convention $P^{\mathrm{TTC}}_{i,\geq R+1} = 0$.

4.4.2 Parameter estimation

Starting from (4.4.3), under the CG model one should have for all $i \in [R-1]$ and $j \in [R]$

$$\Phi^{-1}\left(P^t_{i,\geq j}\right) = \frac{\rho Z^t + \Phi^{-1}(P^{\mathrm{TTC}}_{i,\geq j})}{\sqrt{1-\rho^2}},$$

where ρ is the unknown parameter. Since the matrix $\mathbf{P}^{\mathrm{TTC}}$ interprets, in the CG model, as the migration matrix in the stationary systemic regime, each $P^{\mathrm{TTC}}_{i,\geq j}$ can be estimated as a long-run time average, i.e.

$$\widehat{P^{\mathrm{TTC}}_{i,\geq j}} := \frac{1}{T}\sum_{t=1}^{T} P^t_{i,\geq j}.$$

Then, denoting for all $t \in [T]$, $P^{\Phi,t}_{i,\geq j} := \Phi^{-1}\left(P^t_{i,\geq j}\right)$ and $P^{\Phi,\mathrm{TTC}}_{i,\geq j} := \Phi^{-1}(\widehat{P^{\mathrm{TTC}}_{i,\geq j}})$, we should get

$$P^{\Phi,t}_{i,\geq j} \approx \alpha_1^t + \alpha_2 P^{\Phi,\mathrm{TTC}}_{i,\geq j},$$

where (in the stationary regime)

$$\alpha_1^t \overset{d}{\approx} \mathcal{N}\left(0, \frac{\rho^2}{1-\rho^2}\right), \quad \alpha_2 \approx \frac{1}{\sqrt{1-\rho^2}}.$$

Therefore, solving the least square regression problem

$$\min_{\alpha_2} \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{R}\sum_{i=1}^{R-1}\left(P^{\Phi,t}_{i,\geq j} - \alpha_2 P^{\Phi,\mathrm{TTC}}_{i,\geq j}\right)^2,$$

entails

$$\widehat{\alpha}_2 = \frac{\sum_{t=1}^{T}\sum_{j=1}^{R}\sum_{i=1}^{R-1} P^{\Phi,t}_{i,\geq j} P^{\Phi,\mathrm{TTC}}_{i,\geq j}}{T\sum_{j=1}^{R}\sum_{i=1}^{R-1}(P^{\Phi,\mathrm{TTC}}_{i,\geq j})^2},$$

and the parameter estimation

$$\widehat{\rho} = \pm\sqrt{1 - \left(\frac{1}{\widehat{\alpha}_2}\right)^2}. \tag{4.4.5}$$

We pick the plus sign, for the sake of interpretation of Z as a systematic risk factor (the higher Z, the larger the default probabilities). Then, computing the mean residuals for all $t \in [T]$,

$$\widehat{\alpha_1}^t = \frac{1}{R(R-1)} \sum_{j=1}^{R} \sum_{i=1}^{R-1} \left(P_{i,\geq j}^{\Phi,t} - \widehat{\alpha}_2 P_{i,\geq j}^{\Phi,\mathrm{TTC}} \right),$$

we obtain an estimate

$$\widehat{Z}^t = \frac{\sqrt{1-\widehat{\rho}^2}}{\widehat{\rho}} \widehat{\alpha_1}^t. \tag{4.4.6}$$

Finally, we obtain the reconstructed RMM $\mathbf{P}^{\mathtt{GC}}$ similarly to (4.4.4), i.e.

$$P_{i,j}^{\mathtt{GC},t} = \Phi\left(\widehat{P}_{i,\geq j}^{\Phi,t}\right) - \Phi\left(\widehat{P}_{i,\geq j+1}^{\Phi,t}\right),$$

with

$$\widehat{P}_{i,\geq j}^{\Phi,t} := \widehat{\alpha_1}^t + \widehat{\alpha}_2 P_{i,\geq j}^{\Phi,\mathrm{TTC}},$$

and where by convention $\widehat{P}_{i,R+1}^{\Phi,t} = -\infty$.

4.4.3 Results

The GC model will be evaluated on $\mathbf{P}^{\mathrm{Train}}$ in order 1) to check if the underlying assumptions of the model are verified, and 2) to compare the reconstruction error with our proposed DL model.

Validation. On the first hand, Figure 4.6a displays the empirical distribution of the estimated (4.4.6) on $\mathbf{P}^{\mathrm{Train}}$ with associated $\widehat{\rho} = 0.66$. If the data were coherent with a CG model, this histogram would be close to the stationary distribution of Z, i.e. a standard Gaussian distribution, which is far to be the case. On the other hand, Figure 4.6b illustrates that the model captures in this term (which should represent the business cycle) the market's volatility during the 2007-2013 period. In addition, note that since the real data may not satisfy the economic constraints, neither will the reconstructed CG RMM.

Reconstruction. Table 4.3 reports the Root Mean Square Error (RMSE) $\mathbf{P}^{\mathrm{Reco}} \mapsto \sqrt{\|\mathbf{P}^{\mathrm{Train}} - \mathbf{P}^{\mathrm{Reco}}\|_F^2 / T^{\mathrm{Train}}}$ of the reconstructed RMM using the CG and the DL models. It appears that even with only $K = 2$ atoms, the reconstruction error with the DL approach is about twice lower than with the CG one. This gap is accentuated as we increase the number of atoms.

model	CG	DL(0.1)	DL(1)	DL(6)	DL(10)
error	0.513	**0.304**	0.307	0.310	0.312

Table 4.3: Reconstruction RMSE of $\mathbf{P}^{\mathrm{Train}}$ using the CG model, and the DL (λ) model with $K = 2$ atoms and $\lambda \in \{0.1, 1, 6, 10\}$. The best result is emphasized in bold.

4.5 Conclusion

Modeling RMM is a challenging problem because it is necessary to find an interpretable representation, satisfying economic constraints, while the data are involving in time, not numerous and in a dimension close to the number of observations. We propose a new data-based method

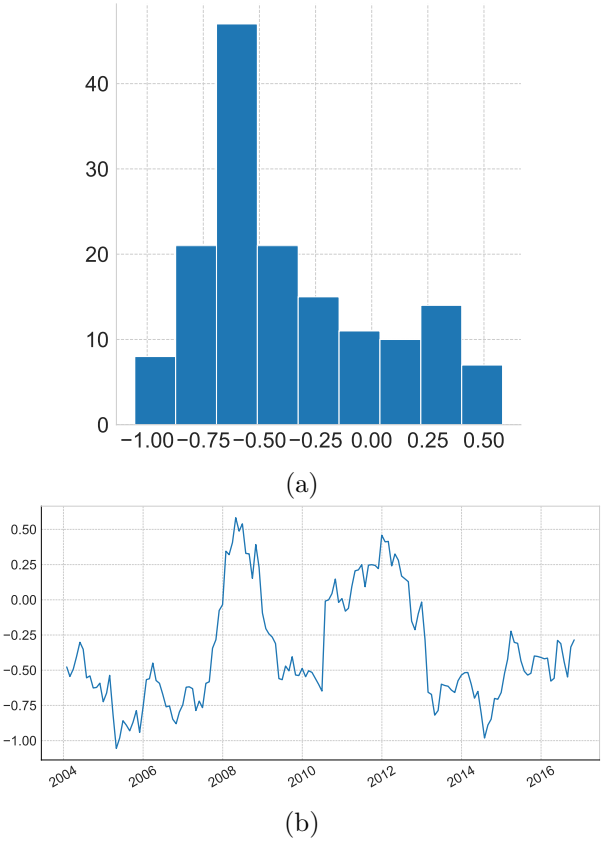


Figure 4.6: Histogram (4.6a) and series (4.6b) of $\left\{\widehat{Z}^t, t \in [T]\right\}$ estimated on $\mathbf{P}^{\text{Train}}$.

using a dictionary learning approach, which implementation boils down to solve small-dimension quadratic optimization problems with linear constraints, leading to a fast algorithm. On the modeling size, we overcome the challenge of a constrained dictionary learning problem and progress in temporal comprehension of the data through the AR regularization. When tested on a real data-set, the method enjoys good accuracy for reconstruction and includes the interpretable classification with respect to economic sentiment indicator. Compared to the popular Gaussian Copula model, it appears that the latter is not appropriate to fit well the real data based on 1) model assumptions not satisfied, 2) twice bigger reconstruction error. Therefore our DL model appears to be an efficient alternative to the widely used Gaussian Copula model.

As perspectives for further works, beyond the better reconstruction ability of our DL approach compared to the GC modeling, it is worthy mentioning that the new model can be used for simulation purposes (to generate scenarios of evolution of RMMs), with some applications to risk analysis of credit portfolio; this will be considered in further research. Moreover, it would be interesting to test the prediction performance of the new model. In addition, in terms of financial interpretation, the clustering analysis suggests that it is possible to connect the temporal evolution of the codings with important macro-economical variables such as GDP or financial indices. The integration of these real indicators in the DL model, is one of the possible extension of the model to be considered in future works.

4.6 Conclusion

Modeling RMM is a challenging problem because it is necessary to find an interpretable representation, satisfying economic constraints, while the data are involving in time, not numerous and in a dimension close to the number of observations. We propose a new data-based method

using a dictionary learning approach, which implementation boils down to solve small-dimension quadratic optimization problems with linear constraints, leading to a fast algorithm. On the modeling size, we overcome the challenge of a constrained dictionary learning problem and progress in temporal comprehension of the data through the AR regularization. When tested on a real data-set, the method enjoys good accuracy for reconstruction and includes the interpretable classification with respect to economic sentiment indicator. Compared to the popular Gaussian Copula model, it appears that the latter is not appropriate to fit well the real data based on 1) model assumptions not satisfied, 2) twice bigger reconstruction error. Therefore our DL model appears to be an efficient alternative to the widely used Gaussian Copula model.

As perspectives for further works, beyond the better reconstruction ability of our DL approach compared to the GC modeling, it is worthy mentioning that the new model can be used for simulation purposes (to generate scenarios of evolution of RMMs), with some applications to risk analysis of credit portfolio; this will be considered in further research. Moreover, it would be interesting to test the prediction performance of the new model. In addition, in terms of financial interpretation, the clustering analysis suggests that it is possible to connect the temporal evolution of the codings with important macro-economical variables such as GDP or financial indices. The integration of these real indicators in the DL model, is one of the possible extension of the model to be considered in future works.

Perspectives

This work has contributed to the investigation of new data-based methods applied for modeling non-standard objects. Here we give some perspectives on potential extensions for each chapter in both theoretical and numerical point of views.

In the first part, we focused on generative modeling using neural networks (NNs) in two different settings: the simulation of fractional Brownian motion (fBm), and of heavy tailed distributions.

On the the first hand, potential extensions of the work in Chapter 1 could consist in investing theoretically more complicated stochastic models written as Stochastic Differential Equations driven by a fBm. One research track is to take advantage of the Polynomial Chaos Expansion studied in a NN framework [165]. Additionally, we could study numerically the optimization of the fBm GAN through the learning of real simulated fBm paths and highlight a deterioration of the convergence rate according to their regularity.

On the other hand, potential extensions of generative methods dedicated to tail events are the following. In Chapter 2, to complete the current theoretical analysis on the EV-GAN model which ensures accurate approximation of marginals, we shall investigate mathematically how the dependence structure is preserved, leveraging multivariate extreme-value theory. Additionally, we shall study numerically the behavior of our proposed model corrections in other GAN architectures, and using different distances and normalization techniques in the training. In Chapter 3, to complete the theoretical analysis on the NN extreme quantile estimator which ensures accurate approximation in both non-conditional and conditional settings in the univariate case, our further work will be dedicated to investigate (in the non-conditional case) multivariate extreme quantile estimation basing on recent characterizations through optimal transport [105]. Additionally, we shall investigate the NN approximation of extreme quantiles to other risk measures such as expected shortfall, or expectiles and then implementing the associated estimators. An interesting line of research could be to mix both works and to study, for example, if the 10% largest observations are better generated by the EV-GAN or by the NN quantile extrapolation model, since we believe that the latter can be be easily transformed to a generative model. Another track could be study the modelisation of other non-standard objects by NN generative models.

In the second part, we investigated a new method based on dictionary learning for modeling financial rating migration matrices. In Chapter 4, it would be interesting to take advantage of the simulation property of our model in order to generate scenarios of evolution of RMMs, with some applications to risk analysis of credit portfolio. In addition, to complete the current model we should investigate the integration of real indicators, such as Gross Domestic Product or financial indices.

List of Algorithms

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. `https://www.tensorflow.org/`.

[2] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. National Bureau of Standards Applied Mathematics Series, No. 55. U. S. Government Printing Office, Washington, D. C., 1964.

[3] A. A. Ahmad, E. H. Deme, A. Diop, S. Girard, and A. Usseglio-Carleve. Estimation of extreme quantiles from heavy-tailed distributions in a location-dispersion regression model. *Electron. J. Stat.*, 14(2):4421–4456, 2020.

[4] M. Allouche, J. El Methni, and S. Girard. A refined Weissman estimator for extreme quantiles. *Extremes*, to appear, 2022.

[5] M. Allouche, S. Girard, and E. Gobet. Estimation of extreme quantiles from heavy-tailed distributions with neural networks. `https://hal.archives-ouvertes.fr/hal-03751980`, 2022.

[6] M. Allouche, S. Girard, and E. Gobet. EV-GAN: Simulation of extreme events with ReLU neural networks. *J. Mach. Learn. Res.*, 23(150):1–39, 2022.

[7] M. Allouche, S. Girard, and E. Gobet. A generative model for fBm with deep ReLU neural networks. *J. Complexity*, page 101667, 2022.

[8] M. Allouche, E. Gobet, C. Lage, and E. Mangin. Structured Dictionary Learning of Rating Migration Matrices for Credit Risk Modeling. `https://hal.archives-ouvertes.fr/hal-03715954`, 2022.

[9] J. Alm. Signs of dependence and heavy tails in non-life insurance data. *Scand. Actuar. J.*, 2016(10):859–875, 2016.

[10] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. volume 70 of *Proc. of Mach. Learn. Res.*, pages 214–223. PMLR, 2017.

[11] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Math. Finance*, 9(3):203–228, 1999.

[12] S. Asmussen and H. Albrecher. *Ruin probabilities*. Advanced Series on Statistical Science & Applied Probability, 14. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, second edition, 2010.

[13] S. Asmussen and P. W. Glynn. *Stochastic simulation: algorithms and analysis*, volume 57 of *Stochastic Modelling and Applied Probability*. Springer, New York, 2007.

[14] A. Ayache and M. S. Taqqu. Rate optimality of wavelet series approximations of fractional Brownian motion. *J. Fourier Anal. Appl.*, 9(5):451–471, 2003.

[15] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inform. Theory*, 39(3):930–945, 1993.

[16] J. Beirlant, Y. Goegebeur, J. Segers, and J. Teugels. *Statistics of Extremes: Theory and Applications*. Wiley, 2004.

[17] M. Bennedsen. A rough multi-factor model of electricity spot prices. *Energy Econ.*, 63:301–313, 2017.

[18] C. Bernard and C. Czado. Conditional quantiles and tail dependence. *Journal of Multivariate Analysis*, 138:104–126, 2015.

[19] D. P. Bertsekas and S. E. Shreve. *Stochastic optimal control*, volume 139 of *Mathematics in Science and Engineering*. Academic Press, Inc., New York-London, 1978.

[20] S. Bhatia, A. Jain, and B. Hooi. ExGAN: Adversarial generation of extreme samples. arXivpreprintarXiv:2009.08454, 2020.

[21] S. Bhatia, A. Jain, and B. Hooi. ExGAN: Adversarial generation of extreme samples. In *Proc. AAAI Conf. Artif. Intell.*, volume 35, pages 6750–6758, 2021.

[22] G. Biau, B. Cadre, M. Sangnier, and U. Tanielian. Some theoretical properties of GANs. *Ann. Stat.*, 48(3):1539–1566, 2020.

[23] G. Biau and L. Devroye. *Lectures on the nearest neighbor method*. Springer, 2015.

[24] G. Biau, M. Sangnier, and U. Tanielian. Some theoretical insights into Wasserstein GANs. arXivpreprintarXiv:2006.02682, 2020.

[25] T. R. Bielecki and M. Rutkowski. *Credit risk: modelling, valuation and hedging*. Springer Finance. Springer-Verlag, Berlin, 2002.

[26] N. H. Bingham, C. M. Goldie, and J. L. Teugels. *Regular variation*, volume 27 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1987.

[27] F. Bourgey, E. Gobet, and C. Rey. Metamodel of a large credit risk portfolio in the gaussian copula model. *SIAM J. Financial Math.*, 11(4):1098–1136, 2020.

[28] G. E. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. John Wiley and Sons, 2008.

[29] J. A. Bucklew. *Introduction to rare event simulation*. Springer Series in Statistics. Springer-Verlag, New York, 2004.

[30] C. Burdet and P. Rougier. Analysis of center-of-pressure data during unipedal and bipedal standing using fractional Brownian motion modeling. *J. Appl. Biomech.*, 23(1):63–69, 2007.

[31] F. Caeiro, M. Gomes, and D. Pestana. Direct reduction of bias of the classical Hill estimator. *Revstat Stat J*, 3(2):113–136, 2005.

[32] J. Cai, J. Einmahl, L. de Haan, and C. Zhou. Estimation of the marginal expected shortfall: the mean when a related variable is extreme. *J. R. Stat. Soc. B*, 77:417–442, 2015.

[33] D. Ceresetti, G. Molinié, and J.-D. Creutin. Scaling properties of heavy rainfall at short duration: A regional analysis. *Water Resour. Res.*, 46(9), 2010.

[34] A. Chambolle, V. Caselles, M. Novaga, D. Cremers, and T. Pock. An introduction to total variation for image analysis, 2010.

[35] V. Chavez-Demoulin, P. Embrechts, and S. Sardy. Extreme-quantile tracking for financial time series. *J. Ecolom.s*, 181(1):44–52, 2014.

[36] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.

[37] U. Cherubini, E. Luciano, and W. Vecchiato. *Copula methods in finance*. Wiley Finance Series. John Wiley & Sons, Ltd., Chichester, 2004.

[38] Y. Cho and L. K. Saul. Learning dictionaries of stable autoregressive models for audio scene analysis. In *Proceedings of the 26th Annual Int. Conf. on Mach. Learn.*, pages 169–176, 2009.

[39] J.-F. Coeurjolly. Simulation and identification of the fractional Brownian motion: A bibliographical and comparative study. *J.. Stat. Softw.*, 5:1–53, 2000.

[40] J.-F. Coeurjolly and E. Porcu. Fast and exact simulation of complex-valued stationary Gaussian processes through embedding circulant matrix. *J. Comput. Graph. Statist.*, 27(2):278–290, 2018.

[41] S. Cohen and J. Istas. *Fractional fields and applications*, volume 73 of *Mathématiques & Applications*. Springer, Heidelberg, 2013.

[42] S. Coles, J. Heffernan, and J. Tawn. Dependence measures for extreme value analyses. *Extremes*, 2(4):339–365, 1999.

[43] F. Comte and E. Renault. Long memory in continuous-time stochastic volatility models. *Math. Finance*, 8(4):291–323, 1998.

[44] R. Corless, G. Gonnet, D. Hare, D. Jeffrey, and D. Knuth. On the Lambert W function. *Adv. Comput. Math.*, 5(1):329–359, 1996.

[45] J. Creutzig, T. Müller-Gronbach, and K. Ritter. Free-knot spline approximation of stochastic processes. *J. Complexity*, 23(4-6):867–889, 2007.

[46] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals, Syst.*, 2(4):303–314, 1989.

[47] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems*, 2(4):303–314, 1989.

[48] A. Daouia, L. Gardes, and S. Girard. On kernel smoothing for extremal quantile regression. *Bernoulli*, 19:2557–2589, 2013.

[49] A. Daouia, L. Gardes, S. Girard, and A. Lekina. Kernel estimators of extreme level curves. *Test*, 20(14):311–333, 2011.

[50] A. Daouia, I. Gijbels, and G. Stupfler. Extremile regression. *J. Amer. Statist. Assoc.*, 117(539):1579–1586, 2022.

[51] I. Daubechies. *Ten lectures on wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM J. Sci. Comput., 1992.

[52] L. de Haan and A. Ferreira. *Extreme value theory*. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2006.

[53] L. de Haan, A. K. Tank, and C. Neves. On tail trend detection: modeling relative risk. *Extremes*, 18:141–178, 2015.

[54] C. de Valk. Approximation and estimation of very small probabilities of multivariate extreme events. *Extremes*, 19:687–717, 2016.

[55] P. Del Moral and J. Garnier. Genealogical particle analysis of rare events. *Ann. Appl. Probab.*, 15(4):2496–2534, 2005.

[56] T. Dieker. Simulation of fractional Brownian motion. Master's thesis, Department of Mathematical Sciences, University of Twente, 2004.

[57] N. Dionelis, M. Yaghoobi, and S. A. Tsaftaris. Tail of distribution GAN (TailGAN): Generative adversarial-network-based boundary formation. In *2020 Sensor Signal Processing for Defence Conference (SSPD)*, pages 1–5. IEEE, 2020.

[58] D. E. Dominici. The inverse of the cumulative standard normal probability function. *Integral Transforms Spec. Funct.*, 14(4):281–292, 2003.

[59] B. Dumitrescu and P. Irofti. *Dictionary learning algorithms and applications*. Springer, 2018.

[60] R. Eckhardt. Stam Ulam, John Von Neumann and the Monte-Carlo method. *Los Alamos Science*, Special Issue:131–143, 1987.

[61] J. H. J. Einmahl, L. de Haan, and C. Zhou. Statistics of heteroscedastic extremes. *J. R. Stat. Soc. B*, 78:31–51, 2016.

[62] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.*, 15(12):3736–3745, 2006.

[63] P. Embrechts, C. Klüppelberg, and T. Mikosch. *Modelling Extremal Events for Insurance and Finance*. Springer-Verlag, Berlin, 1997.

[64] C. Esteban, S. L. Hyland, and G. Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *http://arxiv.org/abs/1706.02633*, 2017.

[65] European Banking Authority. Guidelines on the revised common procedures and methodologies for the supervisory review and evaluation process (SREP) and supervisory stress testing. *Available at* https://eba.europa.eu/regulation-and-policy/supervisory-review-and-evaluation-srep-and-pillar-2/guidelines-for-common-procedures-and-methodologies-for-the-supervisory-review-and-evaluation-process-srep-and-supervisory-stress-testing, EBA/GL/2014/13, 2014.

[66] G. Faber. Über stetige Funktionen. *Math. Ann.*, 66(1):81–94, 1908.

[67] M. Falk, J. Hüsler, and R.-D. Reiss. *Laws of small numbers: extremes and rare events*. Birkhäuser/Springer Basel AG, Basel, 2011.

[68] R. M. Feder, P. Berger, and G. Stein. Nonlinear 3D cosmic web simulation with heavy-tailed generative adversarial networks. *Physical Review D.*, 102(10):103504, 18, 2020.

[69] R. A. Fisher and L. H. C. Tippett. Limiting forms of the frequency distribution of the largest or smallest member of a sample. In *Mathematical proceedings of the Cambridge philosophical society*, volume 24, pages 180–190. Cambridge University Press, 1928.

[70] F. Flandoli and M. Gubinelli. Random currents and probabilistic models of vortex filaments. In *Seminar on Stochastic Analysis, Random Fields and Applications IV*, volume 58 of *Progr. Probab.*, pages 129–139. Birkhäuser, Basel, 2004.

[71] D. Foster. *Generative deep learning: teaching machines to paint, write, compose, and play.* O'Reilly Media, 2019.

[72] I. Fraga Alves, L. de Haan, and T. Lin. Third order extended regular variation. *Publications de l'Institut Mathématique*, 80(94):109–120, 2006.

[73] A. R. Gallant and H. White. There exists a neural network that does not make avoidable mistakes. In *ICNN*, pages 657–664, 1988.

[74] A. K. Gangopadhyay. A note on the asymptotic behavior of conditional extremes. *Statist. Probab. Lett.*, 25(2):163–170, 1995.

[75] M. Garcin. Estimation of time-dependent Hurst exponents with variational smoothing and application to forecasting foreign exchange rates. *Phys. A*, 483:462–479, 2017.

[76] M. Garcin, D. Guegan, and B. Hassani. A novel multivariate risk measure: the Kendall VaR. Technical report, 2018.

[77] L. Gardes. A general estimator for the extreme value index: applications to conditional and heteroscedastic extremes. *Extremes*, 18(3):479–510, 2015.

[78] L. Gardes and S. Girard. A moving window approach for nonparametric estimation of the conditional tail index. *J. Multivariate Anal.*, 99(10):2368–2388, 2008.

[79] L. Gardes and S. Girard. Conditional extremes from heavy-tailed distributions: An application to the estimation of extreme rainfall return levels. *Extremes*, 13(2):177–204, 2010.

[80] L. Gardes and S. Girard. Conditional extremes from heavy-tailed distributions: an application to the estimation of extreme rainfall return levels. *Extremes*, 13(2):177–204, 2010.

[81] L. Gardes and S. Girard. Functional kernel estimators of large conditional quantiles. *Electron. J. Stat.*, 6:1715–1744, 2012.

[82] L. Gardes, A. Guillou, and C. Roman. Estimation of extreme conditional quantiles under a general tail-first-order condition. *Ann. Inst. Statist. Math.*, 72(4):915–943, 2020.

[83] L. Gardes and G. Stupfler. Estimation of the conditional tail index using a smoothed local Hill estimator. *Extremes*, 17(1):45–75, 2014.

[84] J. Gatheral, T. Jaisson, and M. Rosenbaum. Volatility is rough. *Quant. Finance*, 18(6):933–949, 2018.

[85] C. Genest and J. MacKay. The joy of copulas: bivariate distributions with uniform marginals. *Amer. Statist.*, 40(4):280–283, 1986.

[86] C. Genest and L.-P. Rivest. A characterization of Gumbel's family of extreme value distributions. *Statist. Probab. Lett.*, 8(3):207–211, 1989.

[87] C. Genest and L.-P. Rivest. Statistical inference procedures for bivariate Archimedean copulas. *J. Amer. Statist. Assoc.*, 88(423):1034–1043, 1993.

[88] S. Girard, G. Stupfler, and A. Usseglio-Carleve. Extreme conditional expectile estimation in heavy-tailed heteroscedastic regression models. *Ann. Statist.*, 49(6):3358–3382, 2021.

[89] B. Gnedenko. Sur la distribution limite du terme maximum d'une série aléatoire. *Ann. of Math. (2)*, 44:423–453, 1943.

[90] E. Gobet and G. Liu. Rare event simulation using reversible shaking transformations. *SIAM J. Sci. Comput.*, 37(5):A2295–A2316, 2015.

[91] E. Gobet, J. G. López-Salas, and C. Vázquez. Quasi-regression Monte-Carlo scheme for semi-linear PDEs and BSDEs with large scale parallelization on GPUs. *Arch. Comput. Methods Eng.*, 27(3):889–921, 2020.

[92] Y. Goegebeur and T. de Wet. Estimation of the third-order parameter in extreme value statistics. *Test*, 21(2):330–354, 2012.

[93] M. Gomes, L. de Haan, and L. Peng. Semi-parametric estimation of the second order parameter in statistics of extremes. *Extremes*, 5(4):387–414, 2002.

[94] M. Gomes and D. Pestana. A sturdy reduced-bias extreme quantile (VaR) estimator. *J. Amer. Statist. Assoc.*, 102(477):280–292, 2007.

[95] M. I. Gomes, M. F. Brilhante, F. Caeiro, and D. Pestana. A new partially reduced-bias mean-of-order p class of extreme value index estimators. *Comput. Statist. Data Anal.*, 82:223–237, 2015.

[96] M. I. Gomes, M. F. Brilhante, and D. Pestana. New reduced-bias estimators of a positive extreme value index. *Comm. Statist. Simulation Comput.*, 45(3):833–862, 2016.

[97] J. Gondzio. Interior point methods 25 years later. *European J. Oper. Res.*, 218(3):587–601, 2012.

[98] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2016.

[99] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2016.

[100] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Adv. Neural Inf. Process. Syst.*, volume 27, 2014.

[101] R. Gribonval, R. Jenatton, F. Bach, M. Kleinsteuber, and M. Seibert. Sample complexity of dictionary learning and other matrix factorizations. *IEEE Trans. Inform. Theory*, 61(6):3469–3486, 2015.

[102] A. Haar. Zur Theorie der orthogonalen Funktionensysteme. *Math. Ann.*, 69(3):331–371, 1910.

[103] M. Haas and S. Richter. Statistical analysis of Wasserstein GANs with applications to time series forecasting. `arXivpreprintarXiv:2011.03074`, 2020.

[104] P. Hall and A. H. Welsh. Adaptive estimates of parameters of regular variation. *Ann. Statist.*, 13(1):331–341, 1985.

[105] M. Hallin. Measure transportation and statistical decision theory. *Annu. Rev. Stat. Appl.*, 9(1):401–424, 2022.

[106] X. He and P. Ng. Quantile splines with several covariates. *J. Statist. Plann. Inference*, 75(2):343–352, 1999.

[107] R. Hecht-Nielsen. Kolmogorov's mapping neural network existence theorem. In *Proceedings of the international conference on Neural Networks*, volume 3, pages 11–14. IEEE Press New York, 1987.

[108] B. M. Hill. A simple general approach to inference about the tail of a distribution. *Ann. Statist.*, 3(5):1163–1174, 1975.

[109] M. Hofert. Sampling Archimedean copulas. *Comput. Statist. Data Anal.*, 52(12):5163–5174, 2008.

[110] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Netw.*, 4(2):251–257, 1991.

[111] T. Huster, J. E. Cohen, Z. Lin, K. Chan, C. Kamhoua, N. Leslie, C. J. Chiang, and V. Sekar. Pareto GAN: Extending the representational power of GANs to heavy-tailed distributions. *arXiv preprint arXiv:2101.09113*, 2021.

[112] W. P. Johnson. The curious history of Faà di Bruno's formula. *Amer. Math. Monthly*, 109(3):217–234, 2002.

[113] M. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[114] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd Int. Conf. on Learn Represent., ICLR*, 2015.

[115] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *2nd Int. Conf. on Learn Represent., ICLR*, 2014.

[116] E. Kohlbecker. Weak asymptotic properties of partitions. *Trans. Amer. Math. Soc.*, 88(2):346–365, 1958.

[117] I. Kojadinovic and J. Yan. Modeling multivariate distributions with continuous margins using the copula r package. *J. Stat. Soft.*, 34(9):1–20, 2010.

[118] N. Kottegoda and R. Rosso. *Statistics, probability and reliability for civil and environmental engineers*. Mc-Graw-Hill Publishing Company, 2008.

[119] D. Koutsoyiannis. On the appropriateness of the gumbel distribution for modelling extreme rainfall. In *ESF Exploratory*, pages 24–25, 2003.

[120] Y. Kozachenko, A. Pashko, and O. Vasylyk. Simulation of generalized fractional Brownian motion in $C([0,T])$. *Monte Carlo Methods Appl.*, 24(3):179–192, 2018.

[121] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn. Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51(11):1016–1022, 2018.

[122] T. Kühn and W. Linde. Optimal series representation of fractional Brownian sheets. *Bernoulli*, 8(5):669–696, 2002.

[123] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural netw.: Tricks of the trade*, pages 9–48. Springer, 2012.

[124] M. Ledoux and M. Talagrand. *Probability in Banach spaces*, volume 23 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3) Results in Mathematics and Related Areas (3)*. Springer-Verlag, Berlin, 1991.

[125] P. G. Lemarié and Y. Meyer. Ondelettes et bases Hilbertiennes. *Rev. Mat. Iberoamericana*, 2(1-2):1–18, 1986.

[126] M. Leshno, W. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Netw.*, 6(6):861–867, 1993.

[127] D. X. Li. On default correlation: A copula function approach. *J. Fixed Income*, 9(4):43–54, 2000.

[128] J. Li and Z. Qian. Fine properties of fractional Brownian motions on Wiener space. *J. Math. Anal. Appl.*, 473(1):141–173, 2019.

[129] D. MacKenzie and T. Spears. 'the formula that killed wall street': The gaussian copula and modelling practices in investment banking. *Soc. Stud. Sci.*, 44(3):393–417, 2014.

[130] V. Maiorov and A. Pinkus. Lower bounds for approximation by MLP neural networks. *Neurocomputing*, 25(1-3):81–91, 1999.

[131] V. E. Maiorov. On best approximation by ridge functions. *J. Approx. Theory*, 99(1):68–94, 1999.

[132] J. Mairal, F. Bach, and J. Ponce. Sparse modeling for image and vision processing. *arXiv preprint arXiv:1411.3230*, 2014.

[133] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual Int. Conf. on Mach. Learn.*, pages 689–696, 2009.

[134] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Trans. Image Process.*, 17(1):53–69, 2008.

[135] J. Mairal, G. Sapiro, and M. Elad. Multiscale sparse image representationwith learned dictionaries. In *IEEE Int. Conf. Image Process.*, volume 3, pages III–105, 2007.

[136] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. *Multiscale Modeling & Simul.*, 7(1):214–241, 2008.

[137] S. Mallat. *A wavelet tour of signal processing*. Elsevier/Academic Press, Amsterdam, third edition, 2009.

[138] B. B. Mandelbrot and J. W. Van Ness. Fractional Brownian motions, fractional noises and applications. *SIAM Rev.*, 10:422–437, 1968.

[139] B. G. Manjunath and F. Caeiro. *evt0: Mean of order p, peaks over random threshold Hill and high quantile estimates*, 2013. R package version 1.1-3.

[140] A. McNeil and J. Nešlehová. Multivariate Archimedean copulas, d-monotone functions and l_1-norm symmetric distributions. *Ann. Stat.*, 37(5B):3059–3097, 2009.

[141] A. J. McNeil, R. Frey, and P. Embrechts. *Quantitative risk management*. Princeton Series in Finance. Princeton University Press, Princeton, NJ, revised edition, 2015. Concepts, techniques and tools.

[142] Y. Meyer, F. Sellan, and M. S. Taqqu. Wavelets, generalized white noise and fractional integration: the synthesis of fractional Brownian motion. *J. Fourier Anal. Appl.*, 5(5):465–494, 1999.

[143] G. Molinié, D. Ceresetti, S. Anquetin, J.-D. Creutin, and B. Boudevillain. Rainfall regime of a mountainous mediterranean region: Statistical analysis at short time steps. *J. Appl. Meteorol. Climatol.*, 51(3):429–448, 2012.

[144] E. Negri, L. Fumagalli, and M. Macchi. A review of the roles of digital twin in cps-based production systems. *Procedia Manuf.*, 11:939–948, 2017.

[145] R. Nelsen. *An introduction to copulas*. Springer Series in Statistics. Springer, New York, second edition, 2006.

[146] K. Neusser. *Time Series Econometrics*. Number 978-3-319-32862-1 in Springer Texts in Business and Economics. Springer, June 2016.

[147] C. Neves. From extended regular variation to regular variation with application in extreme value statistics. *J. Math. Anal. Appl.*, 355(1):216–230, 2009.

[148] J. Nocedal and S. Wright. *Numerical Optimization*. Springer New York, NY, 1999.

[149] I. Nourdin. *Selected aspects of fractional Brownian motion*, volume 4 of *Bocconi & Springer Series*. Springer ; Bocconi University Press, Milan, 2012.

[150] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vis. Res.*, 37(23):3311–3325, 1997.

[151] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Adv. Neural Inf. Process Syst. 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[152] A. Pinkus. Approximation theory of the MLP model in neural networks. In *Acta numerica, 1999*, volume 8 of *Acta Numer.*, pages 143–195. Cambridge Univ. Press, Cambridge, 1999.

[153] A. Pinkus. *Ridge functions*, volume 205 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2015.

[154] A. Pontieri, F. Lepreti, L. Sorriso-Valvo, A. Vecchio, and V. Carbone. A simple model for the solar cycle. *Sol. Phys.*, 213(1):195–201, 2003.

[155] M. Prandini and O. Watkins. Probabilistic aircraft conflict detection. *HYBRIDGE WP3: Reachability analysis for probabilistic hybrid systems*, 2005.

[156] M. Protter and M. Elad. Image sequence denoising via sparse and redundant representations. *IEEE Trans. Image Process.*, 18(1):27–35, 2009.

[157] A. P. Prudnikov, Y. A. Brychkov, and O. I. Marichev. *Integrals and series. Vol. 1*. Gordon & Breach Science Publishers, New York, 1986.

[158] Z. Qiao, G. Redler, B. Epel, and H. Halpern. A balanced total-variation-chambolle-pock algorithm for epr imaging. *J. Magn. Reson.*, 328:107009, 2021.

[159] C. Remlinger, J. Mikael, and R. Elie. Conditional versus adversarial Euler-based generators for time series. arXivpreprintarXiv:2102.05313, 2021.

[160] S. Resnick. *Heavy-Tail Phenomena: Probabilistic and Statistical Modeling*. Springer, 2007.

[161] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *Int. Conf. on Mach. Learn.*, pages 1530–1538, 2015.

[162] P. Robert. *Stochastic networks and queues*, volume 52 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, french edition, 2003. Stochastic Modelling and Applied Probability.

[163] H. Sagan. *Space-filling curves*. Springer Science & Business Media, 2014.

[164] J. Schauder. Eine Eigenschaft des Haarschen Orthogonalsystems. *Math. Z.*, 28(1):317–320, 1928.

[165] C. Schwab and J. Zech. Deep learning in high dimension: Neural network approximation of analytic functions in $L^2(\mathbb{R}^d, \gamma_d)$. Technical Report 2021-40, Seminar for Applied Mathematics, ETH Zürich, 2021.

[166] P. Shukla, J. Skea, R. Slade, A. A. Khourdajie, R. van Diemen, D. McCollum, M. Pathak, S. Some, P. Vyas, R. Fradera, M. Belkacemi, A. Hasija, G. Lisboa, S. Luz, and J. M. (eds.). *Climate Change 2022: Mitigation of Climate Change. Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press. Cambridge, UK and New York, NY, 2022.

[167] R. Shumway and D. Stoffer. *Time Series and Its Applications*. Springer, New York, 2011.

[168] M. Sklar. Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris*, 8:229–231, 1959.

[169] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd Int. Conf. on Mach. Learn.*, volume 37, pages 2256–2265, 2015.

[170] D. A. Sprecher. A universal mapping for kolmogorov's superposition theorem. *Neural Netw.*, 6(8):1089–1094, 1993.

[171] P. Sprechmann and G. Sapiro. Dictionary learning and sparse coding for unsupervised clustering. In *IEEE Int. Conf. Acoust. Speech Signal Process.*, pages 2042–2045, 2010.

[172] J. M. Steele. *Stochastic calculus and financial applications*, volume 45 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, 2001.

[173] H. Steenbergen, B. Lassing, A. Vrouwenvelder, and P. Waarts. Reliability analysis of flood defence systems. *Heron*, 49:51–73, 2004.

[174] C. Stone. Consistent nonparametric regression. *Ann. Statist.*, 5:595–620, 1977.

[175] M. Telgarsky. Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101*, 2015.

[176] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996.

[177] I. Van Keilegom and L. Wang. Semiparametric modeling and estimation of heteroscedasticity in regression analysis of cross-sectional data. *Electron. J. Stat.*, 4:133–160, 2010.

[178] C. Villani. *Optimal transport*, volume 338 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 2009.

[179] C. Vincent-Cuaz, T. Vayer, R. Flamary, M. Corneli, and N. Courty. Online graph dictionary learning. In *Int. Conf. on Mach. Learn.*, pages 10564–10574. PMLR, 2021.

[180] M. Vladimirova, J. Arbel, and P. Mesejo. Bayesian neural networks become heavier-tailed with depth. In *NeurIPS 2018-Thirty-second Conf. on Neural Inf. Process Syst.*, pages 1–7, 2018.

[181] H. J. Wang and D. Li. Estimation of extreme conditional quantiles through power transformation. *J. Amer. Statist. Assoc.*, 108:1062–1074, 2013.

[182] H. J. Wang, D. Li, and X. He. Estimation of high conditional quantiles for heavy-tailed distributions. *J. Amer. Statist. Assoc.*, 107:1453–1464, 2012.

[183] X. Wang and S. Cheng. General regular variation of the n-th order and 2nd order Edgeworth expansions of the extreme value distribution. II. *Acta Math. Sin. (Engl. Ser.)*, 22(1):27–40, 2006.

[184] I. Weissman. Estimation of parameters and large quantiles based on the k largest observations. *J. Amer. Statist. Assoc.*, 73(364):812–815, 1978.

[185] M. Wiese, R. Knobloch, and R. Korn. Copula & marginal flows: Disentangling the marginal from its joint. *arXiv preprint arXiv:1907.03361*, 2019.

[186] M. Wiese, R. Knobloch, R. Korn, and P. Kretschmer. Quant GANs: deep generation of financial time series. *Quant. Finance*, 20(9):1419–1440, 2020.

[187] M. Wright. The interior-point revolution in optimization: History, recent developments, and lasting consequences. *Bull. Am. Math. Soc.*, 42:39–57, 2004.

[188] F. Wu, E. Valdez, and M. Sherris. Simulating from exchangeable Archimedean copulas. *Commun. Stat. - Simul. Comput.*, 36(5):1019–1034, 2007.

[189] D. Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Netw.*, 94:103–114, 2017.

[190] D. Yarotsky. Optimal approximation of continuous functions by very deep ReLU networks. In *Proceedings Mach. Learn. Res.*, pages 639–649, 2018.

[191] H.-F. Yu, N. Rao, and I. S. Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. *Adv. Neural Inf. Process Syst.*, 29, 2016.

[192] K. Yu and M. Jones. Local linear quantile regression. *J. Amer. Statist. Assoc.*, 93(441):228–237, 1998.

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre : Contributions à la Modélisation Générative et à l'Apprentissage de Dictionnaire : Théorie et Application

Mots clés : Modèle génératif, Réseaux de neurones, Théorie des valeurs extrêmes, Loi à queue lourde, Estimation de quantile, Estimation de quantile conditionnelle, Mouvement Brownien fractionnaire, Apprentissage de dictionnaire.

Résumé : Cette thèse vise à étudier les méthodes basées sur les données dans les paradigmes de l'Intelligence Artificielle et du *Machine Learning*. Bien que très populaires, ces méthodes sont principalement utilisées dans des travaux empiriques. Par conséquent, fournir des directives théoriques pour la construction de tels modèles est d'une importance primordiale.
Dans la première partie, nous étudions la modélisation générative à l'aide de réseaux de neurones dans deux contextes différents : la simulation d'un mouvement Brownien fractionnaire et de lois à queue lourde dans les cas conditionnels et non conditionnels. Dans tous les travaux, nous analysons la vitesse de convergence de l'erreur uniforme entre la fonction d'intérêt et son approximation par réseaux de neurones. Les performances de nos modèles sont illustrées au travers de simulations et de problèmes réels en finance et en météorologie : génération de rendements négatifs extrêmes d'indices financiers et de précipitations en fonction de leur localisation géographique.
Dans la deuxième partie, nous proposons une nouvelle méthode basée sur l'apprentissage par dictionnaire pour la modélisation des matrices de migration des notations financières. Nous devons faire face à une faible quantité de données, proche de la dimension du problème, une évolution rapide dans le temps des matrices et une collection de contraintes linéaires. Nous présentons une étude numérique avec des données réelles et montrons la performance du modèle à la fois comme indicateur de sentiment économique et comme alternative au modèle de Copule Gaussienne largement utilisé.

Title : Contributions to Generative Modeling and Dictionary Learning: Theory and Application

Keywords : Generative model, Neural networks, Extreme-value theory, Heavy-tailed distribution, Quantile estimation, Conditional quantile estimation, fractional Brownian motion, Dictionary Learning

Abstract : This thesis aims at investigating data-based methods in the paradigms of Artificial Intelligence and Machine Learning. Although very popular, those methods are mainly invoked in empirical works. Therefore, providing theoretical guidelines for building such models is of primal importance.
In the first part, we study generative modeling using neural networks in two different settings: the simulation of a fractional Brownian motion and of heavy-tailed distributions in both conditional and non-conditional cases. For all works, we analyze the convergence rate of the uniform error between the function of interest and its neural network approximation. The performance of our models are illustrated on simulations and real practical problems in Finance and Meteorology: generating extreme negative returns of financial indexes and rainfalls as functions of their geographical location.
In the second part, we propose a new method based on dictionary learning for modeling financial rating migration matrices. We have to deal with small amount of data, close to the dimension of the problem, a fast evolution in time of the matrices and a collection of linear constraints. We present a numerical test with real data and show the performance of the model as both an economic sentiment indicator and an alternative to the widely used Gaussian Copula model.

Institut Polytechnique de Paris
91120 Palaiseau, France