Mémoire présenté devant le jury de l'EURIA en vue de l'obtention du Diplôme d'Actuaire EURIA et de l'admission à l'Institut des Actuaires

Le 18 septembre 2024

Par : Kouassi Alphonse Nathan EBROTIE
Titre : Modélisation et prédiction du Best Estimate par réseaux de neurones génératifs

Confidentialité : Non

Les signataires s'engagent à respecter la confidentialité indiquée ci-dessus

Membre présent du jury de l'Institut des Actuaires :
Pierre CORREGE
Jean-Christophe MERER
Signature :

Entreprise :
Malakoff Humanis
Signature :

Membres présents du jury de l'EURIA :
Franck VERMET

Directeur de mémoire en entreprise :
Elise DURAND & Naoufal EL BEKRI

Signature :

Invité :
Signature :

Autorisation de publication et de mise en ligne sur un site de diffusion de documents actuariels
(après expiration de l'éventuel délai de confidentialité)

Signature du responsable entreprise :

Signature du candidat :

EURIA
EURo Institut d'Actuariat

6, avenue le Gorgeu
CS 93837
29238 Brest Cedex 3

T +33 (0)2 98 01 66 55
euria@univ-brest.fr

Résumé

La réglementation Solvabilité II (SII) impose aux assureurs un certain nombre de contraintes dont l'évaluation de leurs provisions techniques de la façon la plus juste possible, ces provisions ayant une composante dite *Best Estimate (BE)*. Un autre enjeu de SII, concerne l'estimation prospective de certaines métriques comme le ratio de Solvabilité dans le cadre de l'ORSA ou pour des allocations stratégiques d'actifs.

Afin de respecter ces exigences du régulateur, les assureurs utilisent généralement des modèles d'*Asset and Liability Management* (ALM) afin d'effectuer des projections des différents flux financiers en fonction de scénarios économiques. À partir de ces projections, les différents éléments d'intérêts (BE, ratio, etc.) peuvent être obtenus en utilisant des méthodes basées sur des simulations comme l'approche classique de Simulations dans les Simulations (SdS). Toutefois, en pratique, cette méthode fondée sur le Monte-Carlo est très couteuse en temps de calcul. C'est pour cette raison que des techniques de réduction de temps de calcul ont été développées. Dans la littérature actuarielle, plusieurs mémoires se sont attardés sur la mise en place de ces méta-modèles pour des fins identiques à ceux évoqués plus haut, mais adoptent des approches alternatives qui ne se généralisent pas souvent et ne permettent pas parfois de capturer efficacement les distributions temporelles des éléments d'intérêts, sachant que cette distribution peut être utile pour des problématiques d'estimations prospectives.

Avec l'avènement d'outils comme *ChatGPT*, *DALL-E*, *Midjourney*, les modèles génératifs font parler d'eux de façon spectaculaire, et laissent entrevoir de nouvelles opportunités en termes de modélisation au vu des capacités des modèles sous-jacents. C'est dans la perspective de tirer profit des possibilités qu'offrent ces modèles génératifs dans des tâches de modélisation que ce mémoire s'inscrit. Notre étude se focalise sur l'utilisation des modèles génératifs afin d'approximer le *Best Estimate*. Cette estimation s'intègre également dans un cadre prospectif et peut permettre de réaliser des anticipations aussi bien sur le BE que sur l'évolution de métriques comme le SCR et le ratio de Solvabilité.

Mots-clés : *Solvabilité II, Best Estimate, proxy, réseaux de neurones génératifs.*

Abstract

The Solvency II (SII) regulations impose a number of constraints on insurers, including the need to assess their technical provisions as accurately as possible, since these provisions include a component known as Best Estimate (BE). Another SII challenge concerns the prospective estimation of certain metrics, such as the Solvency ratio, for ORSA purposes or for strategic asset allocation.

To meet these regulatory requirements, insurers generally use *Asset and Liability Management* (ALM) models to project the various cash flows according to economic scenarios. From these projections, the various interest elements (BE, ratio, etc.) can be obtained using simulation-based methods such as the classic Simulations within Simulations (SdS) approach. In practice, however, this Monte-Carlo-based method is very time-consuming. For this reason, techniques for reducing calculation time have been developed. In the actuarial literature, a number of papers have focused on the implementation of these meta-models for purposes identical to those mentioned above, but adopt alternative approaches that do not often generalize and sometimes fail to efficiently capture the time distributions of the elements of interest, bearing in mind that this distribution can be useful for forward-looking estimation problems.

With the advent of tools such as *ChatGPT*, *DALL-E*, *Midjourney*, generative models are making spectacular headlines, and hint at new opportunities in terms of modeling given the capabilities of the underlying models. It is with a view to taking advantage of the possibilities offered by these generative models in modeling tasks that this dissertation takes place. Our study focuses on the use of generative models to approximate the *Best Estimate*. This estimate is also part of a forward-looking framework, and can be used to forecast both the BE and the evolution of metrics such as the SCR and Solvency ratio.

Keywords : *Solvency II, Best Estimate, proxy, generative neural networks.*

Remerciements

Il n'est pas toujours aisé de trouver un environnement dans lequel on peut être en perpétuel apprentissage tout en se sentant respecté, écouté, aidé et encouragé dans ses initiatives. Pourtant, c'est ce qui a été mon quotidien durant mon stage et mon alternance au sein de l'équipe ALM & Solvabilité II sous la houlette de Hedi LAKHDAR que je remercie.

Ma reconnaissance à l'ensemble des membres de cette chaleureuse et dynamique équipe au sein de laquelle j'ai évolué pendant près d'une année et demie : merci de m'avoir donné l'opportunité d'apprendre auprès de vous tous dans une sympathique ambiance.

Merci à Tristan, Imen, Aissatou, Margot, Coralie et Ronan pour leurs bonnes humeurs, leurs disponibilités et leurs nombreux conseils.

A Naoufal, j'exprime toute ma gratitude, depuis nos collaborations dans le cadre de projets académiques, jusqu'au stage et l'alternance, il n'a cessé de m'inspirer et d'être un véritable mentor. C'est grâce à lui que je me suis découvert ce fort intérêt pour les sujets à l'intersection de la modélisation financière et des nouvelles techniques de data science.

Mes vives remerciements à Elise DURAND, ma tutrice d'alternance, d'avoir accepté de me transmettre son savoir-faire tout en tolérant mes travers, pour son dévouement continuel, les différentes formations qu'elle a pu dispenser depuis le début de mon stage et surtout pour sa bienveillance.

Pour finir, je remercie d'une part le corps professoral de l'EURIA pour la qualité de la formation, en particulier Monsieur Franck VERMET, mon tuteur académique, pour les échanges fructueux que nous avons pu avoir et d'autre part ma famille, mes proches pour leur soutient indéfectible.

A la mémoire de mon père.

Note de synthèse

Contexte et motivations

Dans le cadre de Solvabilité II, les compagnies d'assurance sont soumises à des exigences accrues en matière de quantification des risques et de maintien de fonds propres suffisants pour garantir leur stabilité sur le long terme. Concrètement, la réglementation attend d'eux une évaluation rigoureuse des provisions techniques, avec une attention particulière portée à la composante *Best Estimate Liabilities (BEL)*, introduite lors de l'entrée en vigueur de SII. En outre, les assureurs doivent non seulement démontrer leur solvabilité sur un horizon d'un an, mais aussi effectuer des analyses prospectives pour identifier les risques potentiels et adapter leurs stratégies en conséquence.

Le calcul du BEL étant dépendant de la projection des flux futurs, les assureurs ont recours à des modèles d'*Asset and Liability Management*(ALM) qui permettent de capter les mécanismes d'interactions actif-passif afin d'obtenir les *cash flows*. Cependant, ces modèles reposent (en théorie) sur des simulations intensives basées sur la méthode Monte-Carlo, et plus particulièrement sur la technique de Simulations dans les Simulations (SdS). Bien que cette approche offre une grande précision, elle est également très exigeante en termes de temps de calcul. Cette réalité a incité le secteur à explorer et à développer des techniques de réduction du temps de calcul, visant à rendre ces processus plus efficients sans sacrifier la précision dans l'estimation.

De nombreux travaux se sont intéressés à la mise en place d'alternatives à la SdS. Ces techniques peuvent être catégorisées en approches paramétriques (*Least Squares Monte Carlo* et *Curve fitting*), purement financières (*replicating portfolio*) et de *Machine Learning*(ML). Nous nous sommes intéressés dans ce mémoire à la famille des méthodes d'approximations qui se servent du ML. En explorant la littérature, il ressort que cette approche a connu beaucoup d'engouement ces dernières années avec l'utilisation de modèles de plus en plus sophistiqués, depuis les régressions polynomiales, jusqu'aux réseaux de neurones récurrents en passant par les méthodes de *Boosting*. Même si les résultats issus de ces modèles sont intéressants, ils sont limités ou ne se généralisent pas très bien quand l'horizon temporel s'éloigne de l'instant de calibration initial.

Contribution

D'un autre côté, les modèles génératifs ont été largement explorés dans le domaine de l'apprentissage statistique avec des applications dans de très nombreux domaines comme la finance, l'imagerie médicale ou même dans les outils que nous utilisons au quotidien. Ces applications ont démontré que ces modèles excellent dans les tâches qui leur sont confiées et qui consistent généralement à générer des données ou du contenu ayant certaines propriétés. Cela n'est guère surprenant puisque ces modèles ont été construits pour apprendre à effectuer de la génération en apprenant le processus sous-jacent ayant permis d'obtenir les données d'intérêt. Un aspect intéressant qu'offre ces modèles, s'ils sont convenablement construits, est leur capacité à générer des données de manière guidée. En guidant la génération de données en fonction de certaines informations, nous pouvons les doter d'une capacité prédictive, même si ce n'est pas leur vocation première.

Dans le contexte de l'approximation de métriques réglementaires, les modèles génératifs apparaissent comme une piste naturelle, car leur capacité à apprendre la structure des données sous-jacentes ouvre la voie à une modélisation flexible et précise des flux futurs. Ainsi, nous proposons d'étendre le paradigme discriminatif des modèles proxy de *Machine Learning* au cadre génératif pour estimer le *Best Estimate*.

Ces modèles génératifs peuvent être catégorisés selon deux grandes familles : les modèles implicites et les modèles fondés sur la vraisemblance. La principale différence se situe dans la manière dont ils génèrent les données : les implicites ne cherchent pas exactement la distribution qui a permis d'obtenir les données alors que c'est le cas pour les modèles de vraisemblance.

Nous avons retenu un modèle pour chacune des deux familles évoquées précédemment afin de répondre à notre problématique tout en menant une analyse comparative entre les deux modèles. Nous sommes partis des architectures originelles et nous avons apporté quelques modifications inspirées de la littérature scientifique sur ces modèles afin d'aboutir à des architectures conditionnelles qui favorisent la génération guidée. Ainsi, pour le modèle implicite, c'est le *Generative Adversarial Networks* de Wasserstein avec conditionnement et pénalisation de gradients que nous notons COND-WGAN-GP qui a été adopté. Il est illustré par la figure 1.

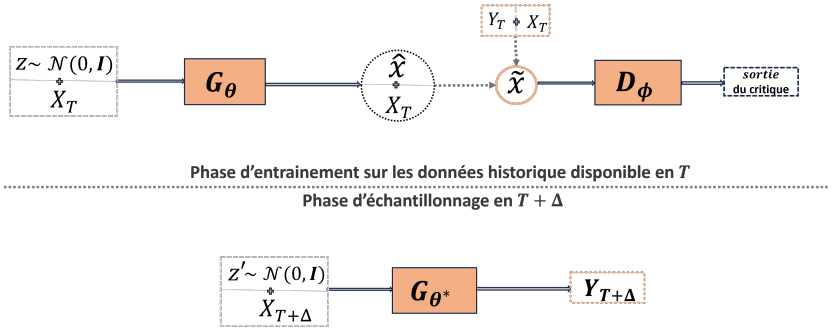

FIGURE 1 – Illustration COND-WGAN-GP

Et pour le modèle de vraisemblance, un Auto-encodeur variationnel avec conditionnement CVAE illustré par la figure 2.

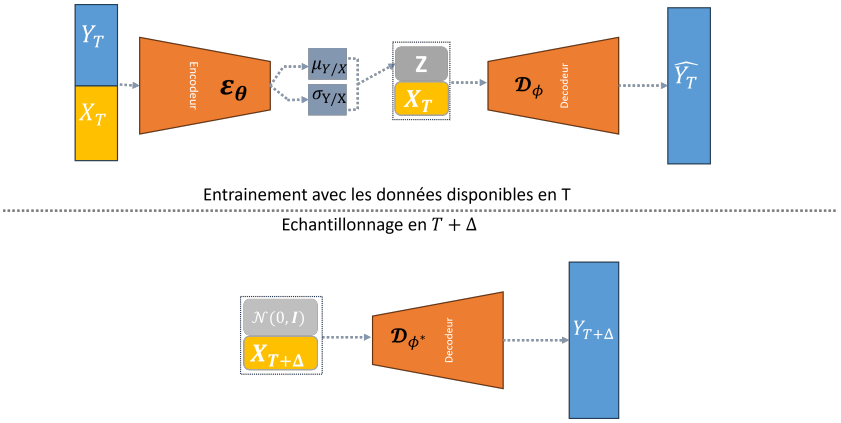

FIGURE 2 – Illustration CVAE

De plus, BEL = Best Estimate Garanti (BEG)+*Future Discretionary Benefits* (FDB)

Ainsi, sans perte de la généralité, nous faisons le choix de modéliser uniquement la FDB car c'est cette quantité qui nécessite l'usage du modèle ALM afin d'effectuer des simulations. La méthodologie est ensuite mise en œuvre sur un portefeuille réel.

Application sur un portefeuille d'assurance vie

Le portefeuille sur lequel nous avons effectué nos études est un portefeuille du périmètre épargne-retraite. L'objectif de modélisation s'écrit : $Y = FDB = f_\Phi(X)$ avec f_Φ étant soit le COND-WGAN-GP soit le CVAE et les variables explicatives obtenues par le biais d'une étude interne qui identifie les variables pertinentes pour un proxy. Plus exactement, X=(Taux, TRC, Taux Cible, PM_{ALM}). Les données à disposition s'étendent de la clôture 2022 à celle de 2023.

Sur ce premier modèle, la calibration s'effectue sur 80 % des données de 2022, choisies aléatoirement et l'échantillonnage est effectué sur les 20% restants. Pour les problématiques de prédictions, une métrique usuelle est l'erreur quadratique moyenne (*Root Mean Square Error*, RMSE) définie par : $RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i^{\text{réel}} - Y_i^{\text{prédit}})^2}$. Cependant, dans notre contexte, cette métrique s'avère inappropriée, car les modèles génératifs que nous utilisons ne visent pas à prédire des valeurs individuelles spécifiques. Ainsi, l'évaluation des modèles s'effectue sur la base de métriques permettant de caractériser des distributions : la moyenne ($\mathcal{M}$), l'écart-type (σ), le quantile à 25% (Q_1) et à 75% (Q_3), l'intervalle interquartile (IQ), la norme L_2 (L_2) entre la densité empirique des données générées et celle des données réelles et le test statistique de Kolmogorov-Smirnov qui permet de savoir si deux ensembles de données proviennent de la même distribution. Les résultats quantitatifs de ce premier modèle sont matérialisés par le tableau 1 et la figure 3. Pour des raisons de confidentialité, les valeurs exactes ne seront pas divulguées pour les métriques quantitatives ; seules des erreurs relatives absolues seront mentionnées. De même, certaines valeurs sur les axes (ordonnée ou abscisse) des figures ou graphiques ne seront pas affichées.

Métriques	$\mathcal{M}$	σ	Q_1	Q_3	IQ	L_2
$\Delta_{\text{métrique}}^{\text{COND-WGAN-GP}}$	0,50%	2,07%	1,66%	0,19%	0,80%	0,03%
$\Delta_{\text{métrique}}^{\text{CVAE}}$	0,49%	1,66%	1,42%	0,15%	0,71%	0,01%

TABLE 1 – Comparaison quantitative de la performance des différents modèles. Lecture : plus la valeur de la variation est proche de 0, meilleur est le modèle pour la métrique concernée.

Où, pour une métrique donnée et modèle $\in \{\text{COND-WGAN-GP}, \text{CVAE}\}$,

$\Delta_{\text{métrique}}^{\text{modèle}} = \frac{|\text{prédit} - \text{réelle}|}{\text{réelle}}$ correspond à la variation relative absolue entre la valeur prédite et la valeur réelle. De même, le test de Kolmogorov-Smirnov atteste que les vrais échantillons et ceux générés proviennent de la même distribution.

Une autre idée a été de tester à quel point les modèles sont capables de prendre en compte l'ajout de variables afin d'ajuster leurs prédictions. Nous avons ajouté ces informations de manière progressive, et il en ressort que les modèles sont capables de les capturer, comme illustré par les figures 4 et 5.

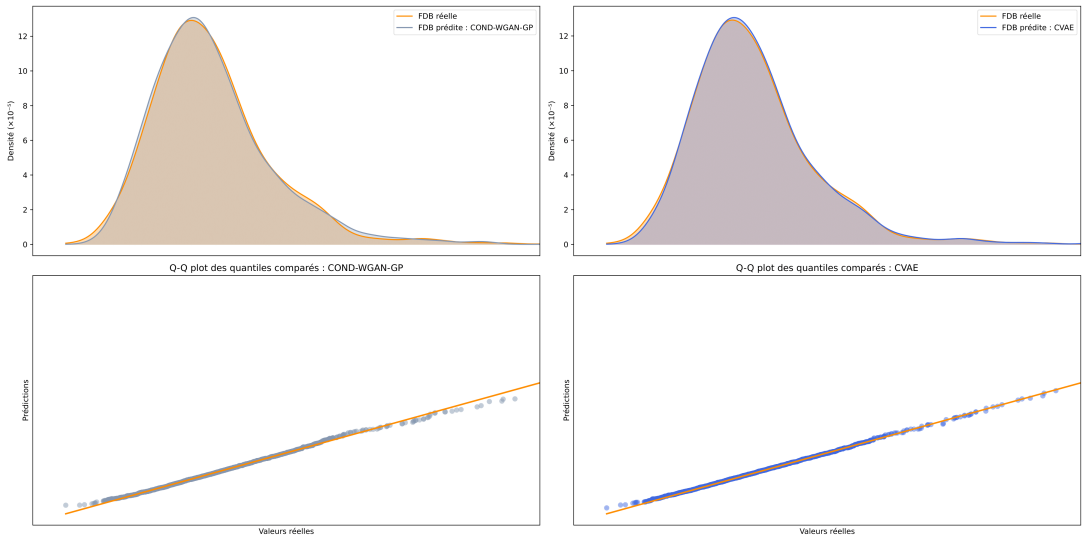

FIGURE 3 – Visualisation de la distribution réelle de la FDB et celle prédite avec le COND-WGAN-GP (à gauche) et avec le CVAE (à droite) ainsi que de leur *quantile-quantile plot*.

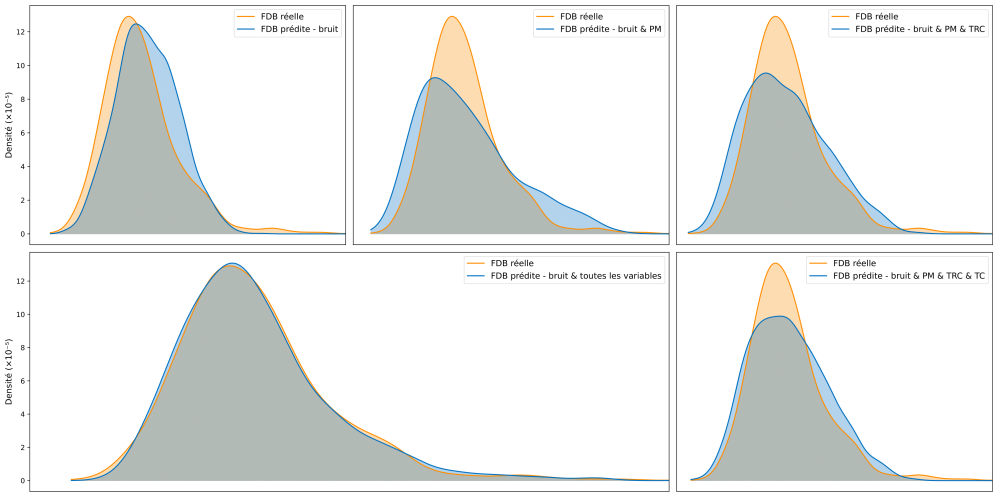

FIGURE 4 – Prise en compte de l'ajout progressif de variables par le COND-WGAN-GP. Lecture : dans le sens des aiguilles d'une montre.

La dernière partie de l'étude a été d'effectuer des sensibilités sur les modèles.

Nous avons préalablement repensé le fonctionnement du modèle général. En fait, pour un trimestre ou une année donnée, en l'absence de modèle ALM, les projections de TRC et de PM_{ALM} ne sont pas observables directement. Par conséquent, des sous-modèles

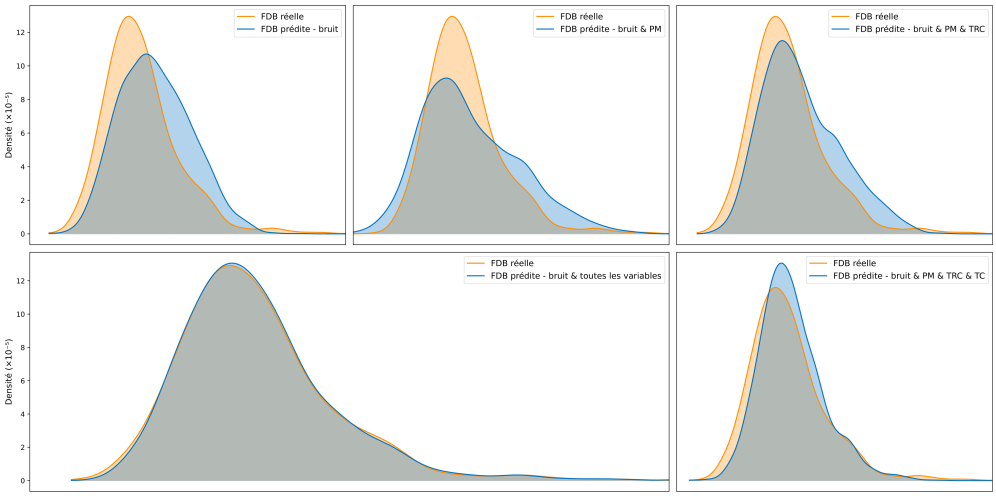

FIGURE 5 – Prise en compte de l'ajout progressif de variables avec modèle CVAE. Lecture : dans le sens des aiguilles d'une montre.

ont été mis en place pour estimer ces quantités. L'objectif de modélisation est alors :

$$FDB = f_\theta(\widehat{X}) \quad \text{où} \quad \widehat{X} = (\text{Taux}, \widehat{\text{TRC}}, \text{Taux Cible}, \widehat{PM_{ALM}})$$

Le modèle $\widehat{\text{TRC}}$ prend comme variables explicatives l'allocation d'actifs et les données émanant du GSE tels que les indices action et immobilier, l'inflation et les taux.

En réalité, l'estimation de la $\widehat{PM_{ALM}}$, se fait via un modèle intermédiaire de $flexing$. En effet, les $PM_{\text{déterministe}}$ étant connues, il suffit d'estimer les coefficients de *flexing* correspondants. Ce modèle prend comme *inputs* le TRC et la courbe des taux.

Le résultat de l'évaluation des modèles sur leur capacité à modéliser ces quantités est illustrée par les figures 6 et 7.

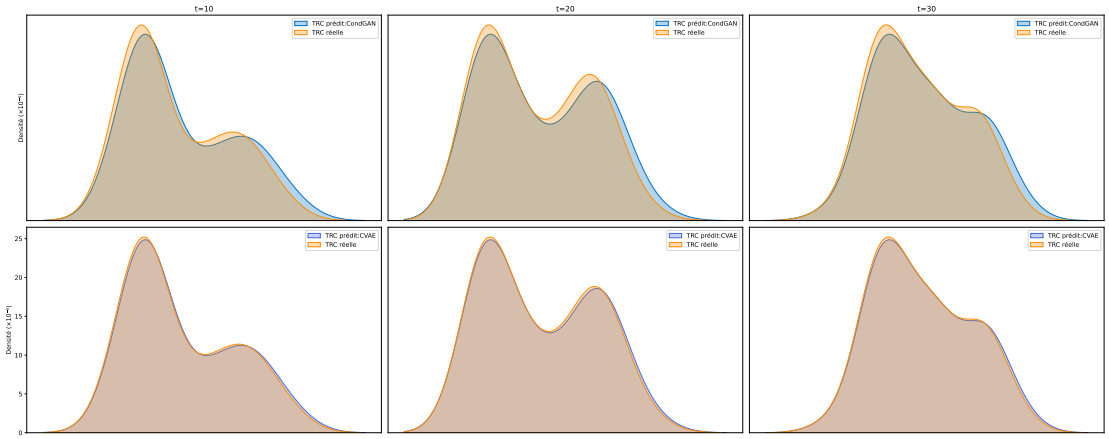

FIGURE 6 – Prédictions du TRC aux instants $t = 10, 20, 30$. Première ligne : COND-WGAN-GP, deuxième ligne : CVAE.

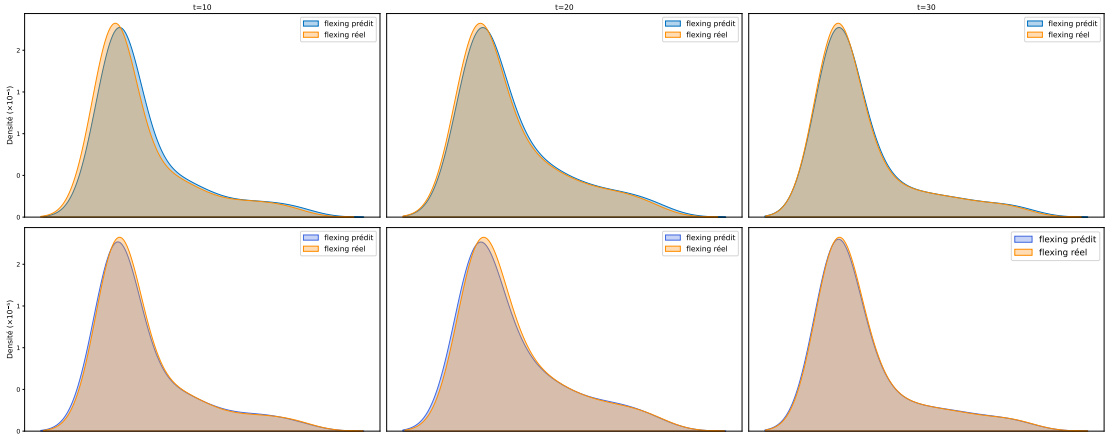

FIGURE 7 – Prédictions du coefficient de *flexing* aux instants $t = 10, 20, 30$. Première ligne : COND-WGAN-GP, deuxième ligne : CVAE.

Avec ces modèles, nous avons approximé le *BE* dans des cas de chocs à la baisse sur le facteur de risque immobilier et de choc à la hausse sur les taux avec des niveaux de chocs correspondants à ceux de SII. Les résultats obtenus sont présentés dans le tableau 2.

Il apparaît que les modèles capturent les effets des deux chocs sur le BE, mais avec des erreurs plus importantes pour le choc de taux que pour le choc immobilier, probablement en raison de la volatilité des taux et de la variabilité du choc de taux par rapport au choc immobilier, qui est fixe.

L'erreur en SCR a été également quantifiée en adoptant l'approche proportionnelle (voir 3.2) dont les hypothèses sont vérifiées par notre entité, du moins sur la période

Modèle	COND-WGAN-GP	CVAE
$\Delta_{BE}^{\text{taux}}$	2,05%	1,37%
$\Delta_{BE}^{\text{central}}$	1,53%	1,11%
$\Delta_{BE}^{\text{immobilier}}$	1,79%	1,14%

TABLE 2 – Résultats BE pour des chocs de marché.$\forall$ choc $\in$ {Taux, immobilier} $\Delta_{BE}^{\text{choc}} = \frac{|\widehat{BE}-BE|}{BE}$ où $\widehat{BE}$ correspond au BE prédit par le modèle. Lecture : plus la valeur de la variation est proche de 0, meilleur est le modèle concerné.

considérée. Ainsi, nous obtenons, en prenant comme instant initial l'année 2021, une erreur relative de moins de 2% pour les deux modèles. Bien que ce calcul ait été effectué pour un horizon de temps proche de l'instant de calibration, ces résultats indiquent que les deux modèles sont capables d'approximer le BE, car en substituant le BE réel par le celui prédit par nos modèles, l'erreur relative est faible.

Pour finir, l'ensemble des *backtests* effectués et des analyses de sensibilité montrent que les deux modèles parviennent à bien capturer les distributions des variables d'intérêt. Cependant, à la lumière des métriques d'évaluation et des graphiques des distributions, le CVAE semble offrir une précision supérieure à celle du modèle implicite.

Conclusion

Dans ce mémoire, nous avons exploré diverses techniques alternatives pour le calcul du *Best Estimate*, en nous concentrant particulièrement sur celles reposant sur le *Machine Learning*. Nous avons proposé d'étendre ces approches à la classe des modèles génératifs. Ces modèles, dotés d'une capacité à modéliser des distributions complexes, qu'elles soient statiques ou dynamiques, peuvent, lorsqu'ils sont correctement construits, répondre efficacement à notre problématique de prédiction du *Best Estimate*. Nous avons présenté et testé deux de ces modèles, le COND-WGAN-GP et le CVAE, sur un portefeuille épargne-retraite. Les résultats obtenus sont encourageants : une fois la calibration effectuée, ces modèles se révèlent non seulement moins chronophages, mais aussi plus flexibles, car ils permettent de se passer de certaines variables, comme l'a montré l'étude de prise en compte progressive des informations. Cette flexibilité pourrait potentiellement faire de ces approches une alternative intéressante à la méthode SdS dans certains contextes. Il convient aussi de préciser que la méthode est appliquée dans ce mémoire sur des volumes de données raisonnables ; son coût computationnel pourrait ainsi devenir important en présence de jeux de données de taille beaucoup plus grande.

Toutefois, certaines limites subsistent, notamment en ce qui concerne la quantité de données nécessaire pour calibrer le modèle (voir tableau 3.7) et le manque de comparai-

son avec des compétiteurs plus classiques.

De plus, des études plus approfondies auraient permis de *challenger* les modèles sur leur capacité à se généraliser dans le temps, notamment avec plusieurs scénarios ORSA. Par ailleurs, notre étude n'ayant exploré que deux modèles parmi la vaste famille des modèles génératifs, il serait intéressant de se pencher sur d'autres catégories, telles que les modèles de diffusion et les *Normalizing Flows*.

Enfin, au-delà de ce cas d'application, l'approche proposée peut également être appliquée à d'autres problématiques comme le calcul du capital économique dans un modèle interne, en utilisant les modèles génératifs pour produire les distributions futures des fonds propres économiques.

Executive summary

Context and motivations

Under Solvency II, insurance companies are subject to increased requirements in terms of quantifying risks and maintaining sufficient capital to guarantee their long-term stability. In concrete terms, the regulations expect them to rigorously assess technical provisions, with particular attention paid to the *Best Estimate Liabilities (BEL)* component, introduced when SII came into force. In addition, insurers must not only demonstrate their solvency over a one-year horizon, but also carry out forward-looking analyses to identify potential risks and adapt their strategies accordingly.

As BEL calculations depend on the projection of future *cash flows*, insurers use Asset and Liability Management models to capture the mechanisms of asset-liability interactions in order to obtain *cash flows*. However, these models rely on intensive Monte-Carlo simulations, and more specifically on the Simulations within Simulations (SdS) technique. While this approach offers high accuracy, it is also very demanding in terms of computing time. This reality has prompted the industry to explore and develop computation time reduction techniques, aimed at making these processes more efficient without sacrificing estimation accuracy.

A number of studies have focused on the development of alternatives to SdS. These techniques can be categorized into parametric (Least Squares Monte Carlo and Curve fitting), purely financial (replicating portfolio) and Machine Learning(ML) approaches. In this dissertation, we focus on the family of ML-based approximation methods. A review of the literature shows that this approach has become very popular in recent years, with the use of increasingly sophisticated models ranging from polynomial regressions to recurrent neural networks and boosting methods. Although the results obtained from these models are interesting, they are limited or do not generalize very well when the time horizon moves away from the initial calibration time.

Contribution

On the other hand, generative models have been widely explored in the field of statistical learning, with applications in a wide range of fields such as finance, medical imaging or even in the tools we use every day. These applications have demonstrated that these models excel at the tasks they are given, which generally involve generating data or content with certain properties. This is hardly surprising, given that these models were built to learn how to perform generation by learning the underlying process by which the data of interest was obtained. An interesting aspect of these models, if properly constructed, is their ability to generate data in a guided way. By guiding the generation of data according to certain information, we can endow them with a predictive capacity, even if this is not their primary vocation.

In the context of approximating regulatory metrics, generative models appear as a natural avenue, as their ability to learn the underlying data structure paves the way for a flexible and accurate modeling of future cash flows. Thus, we propose to extend the discriminative paradigm of Machine Learning proxy models to the generative framework in order to estimate the Best Estimate.

These generative models can be categorized into two main families : implicit models and likelihood-based models. The main difference lies in the way in which they generate the data : implicit models do not look exactly for the distribution that was used to obtain the data, whereas this is the case for likelihood models.

We have chosen one model from each of the two families mentioned above, in order to address our problem while at the same time carrying out a comparative analysis between the two models. We started from the original architectures and made a few modifications inspired by the scientific literature on these models, in order to achieve conditional architectures that favor guided generation. For the implicit model, we adopted Wasserstein's *Generative Adversarial Networks* with conditioning and grandient penalty, which we call COND-WGAN-GP. It is illustrated in figure 8.

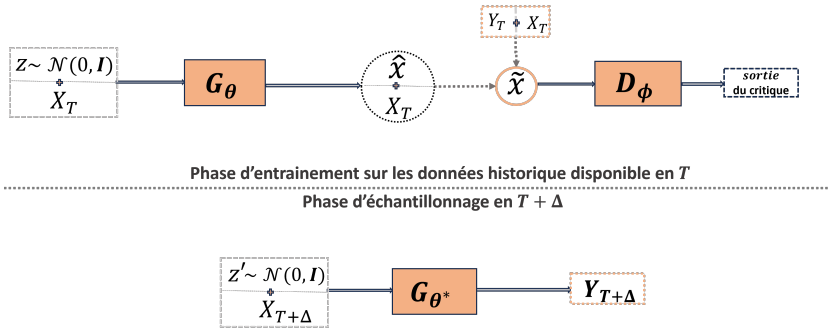

FIGURE 8 – Illustration COND-WGAN-GP

And for the likelihood model, a variational Auto-encoder with conditioning CVAE illustrated by figure 9.

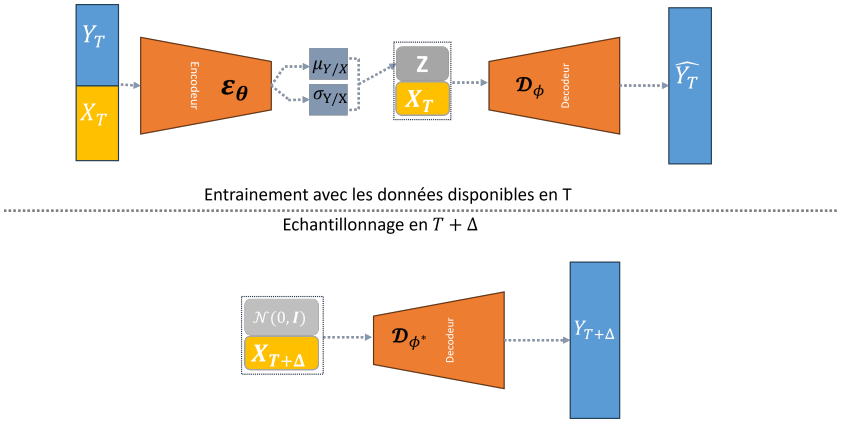

FIGURE 9 – Illustration CVAE

Moreover, BEL = Best Estimate Guarantee (BEG)+*Future Discretionary Benefits* (FDB).

Thus, without loss of generality, we choose to model only the FDB, as it is this quantity that requires the use of the ALM model to perform simulations. The methodology is then implemented on a real portfolio.

Application to a life insurance portfolio

The portfolio on which we have carried out our studies is a savings and retirement portfolio. The modeling objective is written as follows $Y = FDB = f_\Phi(X)$ with f_Φ being either the COND-WGAN-GP or the CVAE and the explanatory variables obtained through an internal study that identifies the relevant variables for a proxy. More precisely, X=(Rate, CRR, Target Rate, PM_{ALM}). The data available covers the period from the close of 2022 to the close of 2023.

In this first model, calibration is performed on 80% of the 2022 data, selected at random, and sampling is carried out on the remaining 20%. For prediction problems, a common metric is the root mean square error (*Root Mean Square Error*, RMSE) defined by : $RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i^{\text{real}} - Y_i^{\text{predicted}})^2}$. However, in our context, this metric proves inappropriate, as the generative models we use do not aim to predict specific individual values. The models are therefore evaluated on the basis of metrics that characterize distributions : mean ($\mathcal{M}$), standard deviation (σ), quantile at 25% (Q_1) and at 75% (Q_3), interquartile range (IQ), the norm L_2 (L_2) between the empirical density of the generated data and that of the real data, and the Kolmogorov-Smirnov statistical test to determine whether two sets of data come from the same distribution. The quantitative results of this first model are shown in the table 3 and the figure 10. For reasons of confidentiality, exact values will not be disclosed for quantitative metrics ; only absolute relative errors

will be mentioned. Similarly, certain values on the axes (ordinate or abscissa) of figures or graphs will not be displayed.

Metrics	$\mathcal{M}$	σ	Q_1	Q_3	IQ	L_2
$\Delta^{\text{COND-WGAN-GP}}_{\text{metric}}$	0,50%	2,07%	1,66%	0,19%	0,80%	0,03%
$\Delta^{\text{CVAE}}_{\text{metric}}$	0,49%	1,66%	1,42%	0,15%	0,71%	0,01%

TABLE 3 – Quantitative comparison of the performance of different models. Reading : the closer the variation value is to 0, the better the model for the metric in question

.

Where, for a given metric and model $\in$ COND-WGAN-GP, CVAE$\}$, $\Delta^{\text{model}}_{\text{metric}} = \frac{|\text{predicted}-\text{actual}|}{\text{actual}}$ corresponds to the absolute relative variation between the predicted and actual values. Similarly, the Kolmogorov-Smirnov test attests that the true and generated samples come from the same distribution.

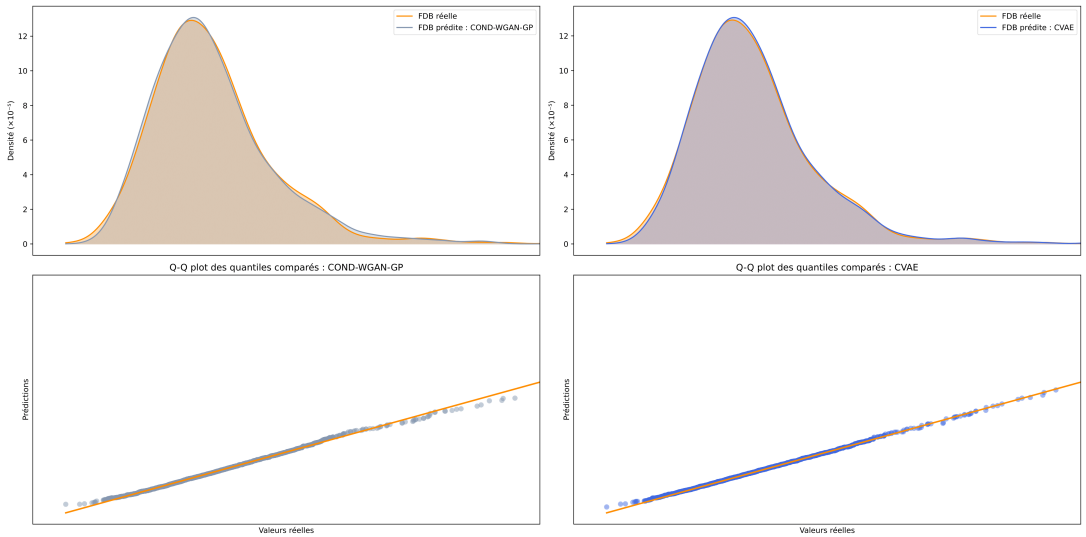

FIGURE 10 – Visualization of the actual distribution of the FDB and that predicted with the COND-WGAN-GP (left) and with the CVAE (right) as well as their *quantile-quantile plot*.

Another idea was to test how well the model is able to take into account the addition of variables to adjust its predictions. We added the variables gradually, and found that the models were able to capture this information, as illustrated by figures 11 and 12.

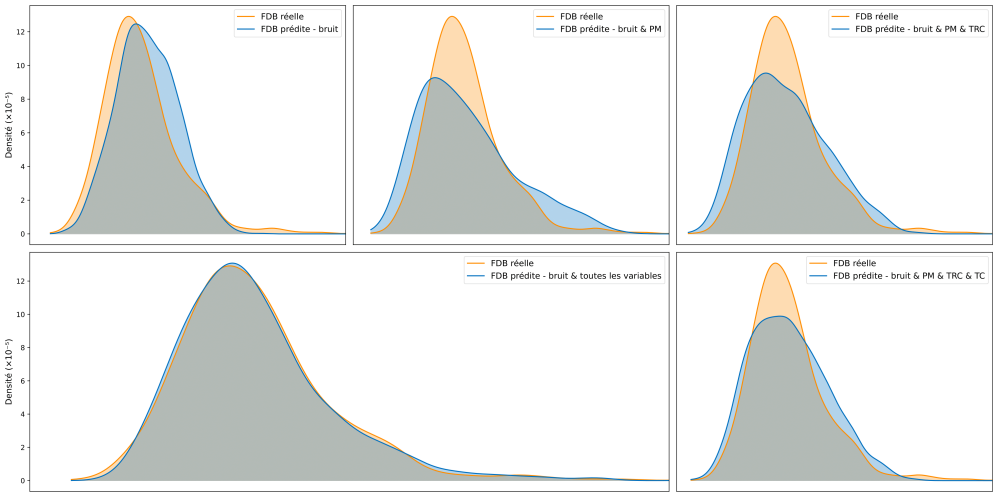

FIGURE 11 – Progressive incorporation of variables by COND-WGAN-GP. Reading : clockwise.

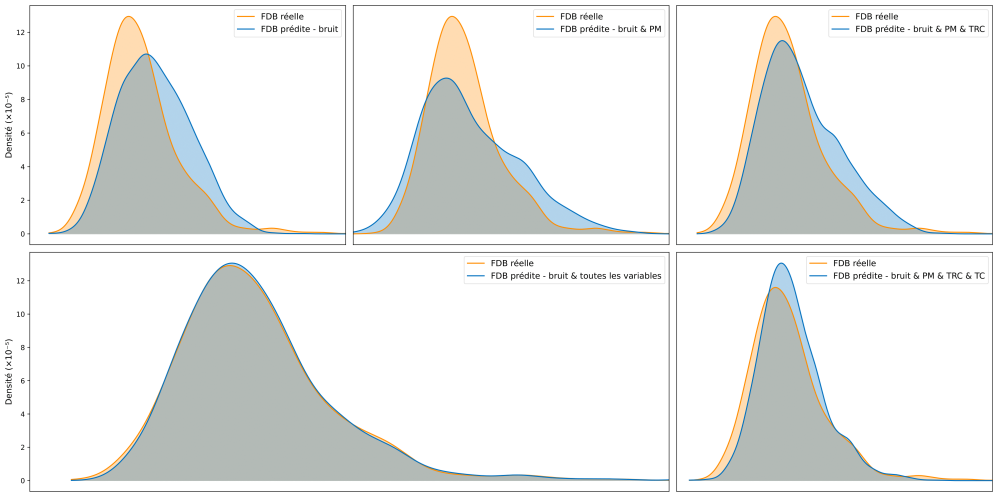

FIGURE 12 – Progressive incorporation of variables by CVAE. Reading : clockwise.

The final part of the study consisted in carrying out model sensitivities.

First of all, we rethought how the general model works. Indeed, for a given quarter or year, in the absence of an ALM model, the TRC and PM_{ALM} projections are not directly observable. Consequently, sub-models have been set up to estimate these quantities. The modeling objective is then :

$$FDB = f_\theta(\widehat{X}) \quad \text{where} \quad \widehat{X} = (\text{Rate}, \widehat{\text{TRC}}, \text{Target rate}, \widehat{PM_{ALM}})$$

The $\widehat{PM_{ALM}}$ model takes asset allocation and GSE data such as equity and real

estate indices, inflation and interest rates as explanatory variables.

In reality, the estimation of $\widehat{PM_{ALM}}$ is done via an intermediate *flexing* model. Since the $PM_{\text{deterministic}}$ are known, all we need to do is estimate the corresponding *textitflexing* coefficients. This model takes as inputs the CRR and the yield curve. 13 and 14 illustrate the results of the models' evaluation of their ability to model these quantities.

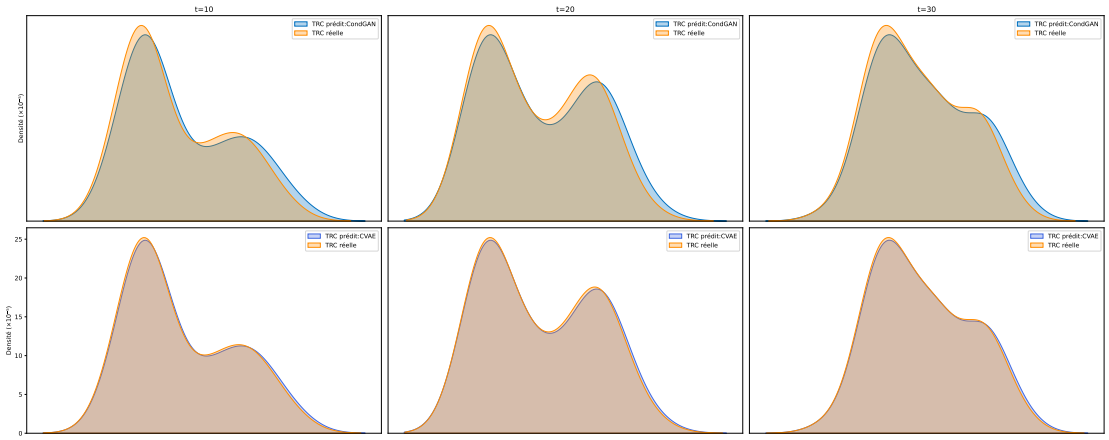

FIGURE 13 – TRC predictions at time instants $t = 10, 20, 30$. First line : COND-WGAN-GP, second line : CVAE.

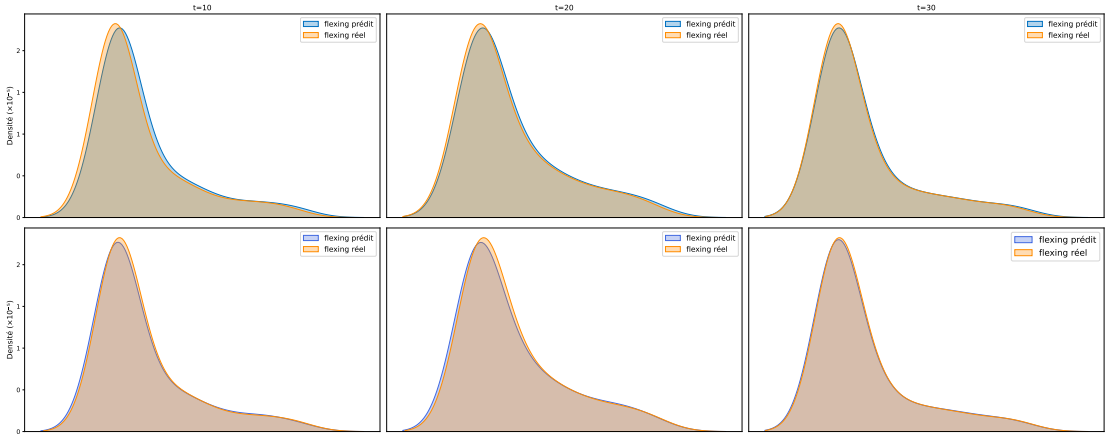

FIGURE 14 – Predictions of the *flexing* coefficient at times $t = 10, 20, 30$. First line : COND-WGAN-GP, second line : CVAE.

With these models, we have approximated the BE in cases of downward shocks on the real estate risk factor and upward shocks on rates. The results obtained are presented in table 4.

Model	COND-WGAN-GP	CVAE
$\Delta_{BE}^{\text{rate}}$	2,05%	1,37%
$\Delta_{BE}^{\text{main}}$	1,53%	1,11%
$\Delta_{BE}^{\text{real estate}}$	1,79%	1,14%

TABLE 4 – BE results for market shocks.$\forall$ choc $\in$ {rate,real estate} $\Delta_{BE}^{\text{choc}} = \frac{|\widehat{BE}-BE|}{BE}$ où $\widehat{BE}$ corresponds to the BE predicted by the model. Reading : the closer the variation value is to 0, the better the model concerned.

It appears that the models capture the effects of both shocks on the BE, but with larger errors for the interest rate shock than for the real estate shock, probably due to the volatility of interest rates and the variability of the interest rate shock compared with the real estate shock, which is fixed.

The SCR error has also been quantified by adopting the proportional approach (see 3.2), whose assumptions are verified by our entity, at least over the period under consideration. Thus, taking 2021 as the initial point in time, we obtain a relative error of less than 2% for both models. Although this calculation was carried out for a time horizon close to the calibration instant, these results indicate that both models are capable of approximating the BE, since by substituting the actual BE with the one predicted by our models, the relative error is low.

Finally, all the backtests carried out and the sensitivity analyses show that both models succeed in capturing the distributions of the variables of interest. However, in the light of the evaluation metrics and the graphs of the distributions, the CVAE seems to offer greater accuracy than the implicit model.

Conclusion

In this thesis, we have explored various alternative techniques for calculating the *Best Estimate*, focusing particularly on those based on *Machine Learning*. We propose to extend these approaches to the class of generative models. These models, with their ability to model complex distributions, whether static or dynamic, can, when correctly constructed, effectively address our *Best Estimate* prediction problem. We have presented and tested two of these models, COND-WGAN-GP and CVAE, on a retirement savings portfolio. The results are encouraging : once calibrated, these models not only prove to be less time-consuming, but also more flexible, as they can do without certain variables, as shown by the study of the progressive inclusion of information. This flexibility could potentially make these approaches an interesting alternative to the SdS method in certain

contexts.It should also be noted that the method is applied in this thesis to datasets of reasonable size ; its computational cost could therefore become significant when dealing with much larger datasets.

However, certain limitations remain, particularly regarding the amount of data required to calibrate the model (see Table 3.7) and the lack of comparison with more traditional competitors.

In addition, more in-depth studies would have allowed us to challenge the models on their ability to generalize over time, particularly with several ORSA scenarios. Furthermore, as our study explored only two models from the vast family of generative models, it would be interesting to look at other categories, such as diffusion models and *Normalizing Flows*.

Finally, beyond this application case, the proposed approach can also be applied to other issues, such as the calculation of economic capital in an internal model, using generative models to produce future distributions of economic equity.

Table des matières

Table des figures

Liste des tableaux

Introduction

L'assurance occupe une place centrale dans les systèmes économiques modernes puisqu'elle favorise l'essor économique et assure la protection des individus des conséquences financières émanant des incertitudes de la vie, qui sont généralement des phénomènes aléatoires. Son cycle de production est dès lors inversé puisqu'elle reçoit une prime avant de fournir la prestation correspondante, ne sachant pas à priori le moment où le risque couvert surviendra et quel sera son coût. Ainsi, le secteur assurantiel a une position délicate et il est essentiel pour les politiques d'avoir des moyens de s'assurer que cette industrie reste solide puisque sa fébrilité entrainerait, indéniablement, un affaiblissement du système économique.

La réglementation apparaît comme un outil naturel pour atteindre cet objectif. En tant qu'entreprises du système économique, les assureurs sont soumis à la réglementation de base, mais au vu de leur place sociale, de nouvelles contraintes s'ajoutent avec l'introduction de normes prudentielles dont la plus notable est Solvabilité II.

Avec l'entrée en vigueur de Solvabilité II, les assureurs sont soumis à une réglementation encore plus stricte et prudente en termes de quantification des incertitudes inhérents à leur activité et au niveau de fonds propres assurant leur pérennité.

De manière pratique, il leur incombe d'évaluer leurs provisions techniques de manière précise avec une nouvelle composante dite *Best Estimate Liabilities (BEL)* et de fournir des indicateurs définis par le régulateur, qui permettent de mesurer leur niveau de solvabilité sous un horizon temporel d'un an. Mais cela ne s'arrête à une vision ponctuelle à un an puisque Solvabilité II, prévoit dans son pilier 2 (voir 1.2), un module qui concerne des études prospectives afin de mieux cerner les risques de l'entité, l'*Own Risk and Solvency Assesment (ORSA)* et parfois même permet à l'organisme d'assurance d'aiguiller ses choix. L'estimation prospective revêt ainsi un caractère important pour les organismes.

Pour respecter ces exigences du régulateur, les assureurs utilisent généralement des modèles ALM afin d'effectuer des projections des différents flux financiers en fonction de scénarios économiques. À partir de ces projections, les différents éléments d'intérêts (BE, ratio, etc) peuvent être obtenu en utilisant des méthodes basées sur le Monte-Carlo comme l'approche classique de Simulations dans les Simulations (SdS). Toutefois, en pratique, cette méthode fondée sur le Monte-Carlo est très couteuse en temps de calcul et

ne favorise donc pas les études prospectives. C'est pour cette raison que des techniques de réduction de temps ont été développés.

Dans la littérature actuarielle, plusieurs mémoires se sont attardés sur la mise en place de ces méta-modèles pour des fins identiques à ceux évoqués plus haut.

Généralement, ces travaux s'appuient sur des techniques d'interpolation, de *krigeage* [Zurfluh, 2019], de régression linéaire, de réseaux de neurones *feedforward* classique. Cependant, comme l'a souligné [Cescutti, 2016], ces méthodes alternatives ont du mal à "reproduire" les résultats issus des "vrais" modèles ALM pour un horizon de temps éloigné de l'instant initial, puisqu'en raison de la manière dont ils sont construits, ils ne captent pas ou sont limités dans la prise en compte du vieillissement des modèles points à travers le temps. Récemment, des modèles neuronaux récurrents ont été introduits afin de palier ce problème [Baschung, 2023, Morisse, 2022] puisqu'ils permettent, de prime abord, une modélisation plus efficace.

Ces modèles ont fourni des résultats prometteurs, mais restent souvent spécifiques au portefeuille sur lequel il a été appliqué et ne permettent pas toujours de capter correctement les distributions temporelles des différentes métriques.

De cette littérature actuarielle, il ressort que l'essentiel des travaux se sont concentrés sur des méthodes de *Machine Learning* standards alors que des techniques de modélisation générative commencent à faire leurs preuves, majoritairement dans l'univers de la finance [Florian *et al.*, 2021] sans oublier l'intérêt soudain et grandissant pour les *Large Language Models* dans de nombreux domaines dont l'assurance.

Nos travaux s'inscrivent dans une approche exploratoire afin d'investiguer sur la capacité de ces modèles dans des tâches de modélisation prospective. Nous explorerons et mettrons en lumière le potentiel des modèles génératifs tels que les *Generative Adversarial Networks*, dans un contexte de modélisation actuarielle.

Fondamentalement, ces modèles génératifs étant conçus dans l'objectif d'apprendre à générer de nouvelles données fidèles en un certain sens aux données d'origines (dont la distribution est généralement inconnue), peuvent à priori être de bons candidats dans des tâches d'estimation de certaines métriques et de leur distribution au fil du temps. Dans notre cas nous nous intéressons au *Best Estimate*.

Afin de proposer une réponse à cette problématique, qui concerne la mise en place d'une méthode alternative à la SdS, nos travaux se concentrent autour de quatre parties principales. D'entrée de jeu, nous situons le contexte en exposant les notions clés afin de permettre au lecteur de se familiariser avec le sujet et de bien cerner les contours du problème traité. Ensuite, une partie est dédiée aux outils mathématiques et au cadre de modélisation. S'ensuit une section opérationnelle visant à illustrer un cas concret, tout en soulignant les limites de notre étude et en proposant des perspectives d'amélioration.

Chapitre 1

Eléments de contexte

Ce mémoire traite de la mise en place d'un modèle à des fins d'approximation du *Best Estimate*. Avant de dérouler le cœur du sujet, il est convenable d'introduire l'environnement (produits, réglementations, etc.) de notre étude.

1.1 Assurance de personnes

1.1.1 Généralités

Par définition, l'assurance correspond à une activité de transfert de risques d'un adhérent vers un organisme d'assurance moyennant une prime reçue préalablement, sans savoir si le risque couvert surviendra et combien il coutera en cas de survenance (notion de cycle inversé en assurance).

Le risque sur lequel porte le contrat est un évènement aléatoire. La réalisation de cet évènement peut donner lieu à un préjudice matériel, on parle dans ce cas d'assurance non-vie et plus précisément d'IARD (Incendies, Accidents et Risques Divers) ou peut porter sur des dommages corporels, on parle alors d'assurance de personnes qui contient l'assurance vie. L'assurance de personne couvre les risques :

- Prévoyance : risque décès (branche vie), l'arrêt de travail et la dépendance (branche non-vie)
- Santé (branche non-vie)
- Epargne-retraite (branche vie)

Nos travaux s'inscrivent dans le cadre de l'assurance vie et plus particulièrement l'épargne-retraite, qui représente historiquement le moyen favori d'épargne en France [1].

L'intérêt de ce type d'assurance concerne la possibilité de se constituer une épargne avec des options intéressantes comme la possibilité de verser une rente à des bénéficiaires en cas de décès du titulaire du contrat ou de simplement récupérer le montant d'épargne constitué si le souscripteur est toujours en vie ou de racheter le contrat à n'importe quel moment.

1. https ://www.economie.gouv.fr/cedef/assurance vie

L'autre avantage indéniable concerne la fiscalité qui peut être favorable à bien des égards puisqu'elle permet une transmission de patrimoine et en cas de rachat (total/partiel), la fiscalité s'applique uniquement sur les plus-values et les intérêts éventuels.

1.1.2 Périmètres et types de supports

En assurance de personnes, comme évoqué, une différentiation subsiste selon qu'il s'agit de l'épargne, de la prévoyance ou de la santé. Les spécificités de ces périmètres sont présentées ci-dessous :

Epargne retraite

Il s'agit de l'ensemble des produits qui offrent au souscripteur la possibilité de constituer une épargne qui lui sera restituée lors de sa retraite. La spécificité de ce type de contrat, c'est que l'épargnant ne peut accéder à son dû que pendant sa retraite ou selon certaines conditions définies dans les clauses du contrat. De plus, les droits restent acquis tout au long de la vie du contrat, même en cas de non-paiement des primes (par exemple le cas d'un salarié ayant quitté l'entreprise).

Généralement, en assurance vie et spécifiquement en épargne retraite, deux supports s'offrent au souscripteur pour investir son épargne :

- **Les supports en unités de compte (UC).** Il s'agit de placement sur des fonds gérés par des gestionnaires d'actifs constitués de classes d'actifs plus risqués (en comparaison aux supports en euros) comme les actions. En fonction, de son appétence au risque, l'investisseur peut ajuster son placement. Même si ce type de support est souvent plus exposé au risque, le gain potentiel est supérieur. Pour un souscripteur ayant choisi les supports UC, l'assureur ne lui garantit pas de capital, mais uniquement un certain nombre de parts évalué par rapport à la performance des marchés financiers.

- **Les supports en euros.** Ce type de placement est "plus sécurisé" (comparé à l'investissement sur des supports en UC) puisque l'épargne est investi essentiellement sur des actifs avec des rendements quasi-garantit comme des emprunts étatiques. Dans ce type de support l'assureur garantit à l'assuré une revalorisation de son capital à un taux minimum garantit (TMG) avec une contrainte réglementaire d'un versement du taux minimum garantit correspondant à au moins 85 % du résultat financier auquel on ajoute au moins 90% de résultat technique, cela correspond à la participation au bénéfice (PB) réglementaire.
  Cette participation aux bénéfices s'appréhende comme la quote-part de bénéfice reversée à un détenteur de support en euros et peut correspondre au montant réglementaire, contractuel couplé avec une part discrétionnaire (lorsque la compagnie d'assurance décide de verser des PB supplémentaires aux assurés en raison de bonnes performances). La modélisation de cette partie discrétionnaire se fait via des modèles stochastiques (influence ainsi la *Future Discretionary Benefits (FDB)*).

Prévoyance-santé

La prévoyance-santé se définit comme un type d'assurance qui permet au souscripteur de se protéger contre les conséquences financières découlant de la maladie, l'accident, l'hospitalisation et le décès dans les cas où la couverture de base (Sécurité sociale) est insuffisante.

La grande différence avec l'épargne retraite, c'est qu'il n'y a pas de notion de constitution/restitution de capital dans la mesure où si le risque que couvre le contrat ne s'est pas réalisé, le souscripteur ne récupère pas son capital ou ne peut le transférer à un bénéficiaire. Il faut avoir cotisé dans l'année pour bénéficier de la garantie.

Sous ce prisme, plusieurs risques sont couverts, comme illustré sur le schéma ci-dessous :

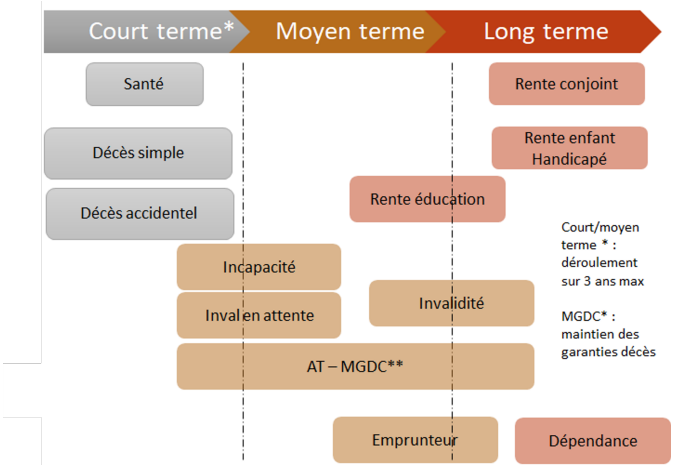

FIGURE 1.1 – Risques couverts en prévoyance santé selon la durée (source : interne)

Dans nos travaux, nous nous intéresserons qu'au segment de l'épargne-retraite.

Une fois les éléments clé de l'assurance vie présentés, nous procédons dans la suite à la description du contexte prudentiel sous-jacent à notre étude.

1.2 Réglementation Solvabilité II

1.2.1 Aperçu général

Le milieu assurantiel est indéniablement influencé par l'aspect juridique et réglementaire. Cela peut s'expliquer non seulement par une volonté manifeste des autorités étatiques de « protéger » le capital des assurés, mais aussi par le fait que les assureurs sont des acteurs majeurs de l'économie (opérations d'achat/vente de titres). En imposant un certain nombre de mesures, les gouvernements s'assurent que les entités assurantielles sont des acteurs fiables et robustes, sur lesquels ils peuvent s'appuyer, d'où la nécessité

de leur solvabilité.

Depuis solvabilité II, la solvabilité d'un organisme d'assureur s'apprécie par le ratio entre le montant de ses fonds propres et le montant minimal qu'il doit détenir. Si ce ratio est inférieur à 100%, l'organisme n'est pas solvable et dans ce cas l'autorité de contrôle intervient (en France, c'est l'Autorité de Contrôle Prudentiel et de Résolution).

Sous Solvabilité I, l'objectif était d'évaluer le montant minimum de fonds propres que devait détenir l'organisme d'assurance, l'Exigence de Marge de Solvabilité (EMS). La méthodologie de calcul était simple à mettre en place et à comprendre mais ne permettait pas une prise en compte réelle des risques auxquels les assureurs sont exposés. Contrairement, Solvabilité II, s'inscrit dans une approche plus réaliste en se focalisant de manière plus détaillée sur les risques auxquels les assureurs sont soumis dans la détermination du montant minimal.

Au niveau bilantiel, sous Solvabilité I, la valorisation se fait au coût historique alors qu'elle s'effectue en valeur économique sous Solvabilité II. De manière plus précise :

- Les actifs sont valorisés selon leurs valeurs de marché, c'est-à-dire évalués au montant pour lequel ils seraient échangés dans des conditions normales de marché.
- Les passifs sont évalués en valeur économique, représentant le montant auquel un autre assureur accepterait de les reprendre.

En introduisant le 1^{er} janvier 2016, Solvabilité II, l'objectif visé va au-delà de la simple mise en place d'une nouvelle manière de mesurer la stabilité financière des entités assurantielle au niveau de l'Union européenne, puisque cette nouvelle norme prudentielle embarque un ensemble de réformes au niveau de la gouvernance (pilier 2 de Solvabilité II), de la communication (pilier 3) et de l'évaluation du bilan en lui-même (pilier 1). Ci-dessous une présentation synthétique des 3 piliers de Solvabilité II :

Pilier 1 : les exigences quantitatives

L'objectif de ce pilier est l'harmonisation des méthodes de calculs de la valeur des actifs ainsi que des provisions techniques tout en alignant les capitaux requis aux risques réellement pris par la compagnie. Les principaux éléments de ce pilier sont :

- L'évaluation « *market consistent* » des actifs et des passifs d'assurance (meilleure estimation pour les provisions techniques) ;
- En remplacement de l'EMS, deux nouvelles exigences en capital :

- Le *Solvency Capital Requirement (SCR)* : est équivalent au capital qui permet à l'organisme d'assurance de faire face à des pertes imprévues tout en continuant son activité. C'est aussi, les fonds propres dont doit disposer l'organisme pour être certain, avec probabilité 99,5 % de ne pas faire faillite sous un horizon d'un an.

- Le *Minimum Capital Requirement (MCR)* : correspond au niveau de fonds propres en dessous duquel l'organisme d'assurance se voit retiré son agrément par les entités de contrôle. C'est aussi le niveau de fonds propres qui permet d'assurer avec probabilité de 80 % que ma compagnie ne fait pas faillite. C'est généralement entre 25 % et 45% du SCR.

• Mise en place de critères d'éligibilité et de classement des fonds propres par qualité.

Pilier 2 : les exigences qualitatives

Ce pilier concerne le système de gouvernance, le principe de la personne prudente et l'*Own Risk Assessment (ORSA).*

Au niveau du système de gouvernance, la réforme prévoit des exigences de compétence et d'honorabilité, des politiques validées et révisées fréquemment et surtout le principe des "4 yeux" qui fait intervenir 4 fonctions clés : actuariat, audit interne, conformité et gestion des risques.

D'un autre côté, le principe de la personne prudente stipule qu'il faut être en mesure d'identifier, mesurer, gérer, suivre et contrôler les risques inhérents aux différents placements et instruments financiers. Par exemple, il faut conserver la part d'actifs non cotés à un certain seuil de prudence, utiliser les dérivés uniquement à des fins de réduction de risques et non de spéculations.

L'ORSA est une évaluation en interne des différents risques et de la solvabilité en vérifiant le besoin global de solvabilité, le profil de risque réel de l'entreprise et le respect en permanence des contraintes réglementaires.

Pilier 3 : Le reporting prudentiel et l'information au public

Ce pilier s'intéresse à la communication financière et la mise en place d'états prudentiels communs à tous les contrôleurs européens. Le rapport qui est destiné au public est le *Solvency and Financial Conditions Report (SFCR)* et celui à l'attention du régulateur est le *Regular Supervisory Report (RSR).*

Afin de pouvoir mesurer la stabilité financière des entités assurantielles, des métriques ont été introduites par le régulateur.

1.2.2 Principales métriques réglémentaire

Comme évoqué précédemment, une différence subsiste entre Solvabilité I et Solvabilité II au niveau Bilantiel. Le nouveau bilan (simplifié) sous Solvabilité II est illustré par la figure ci-dessous :

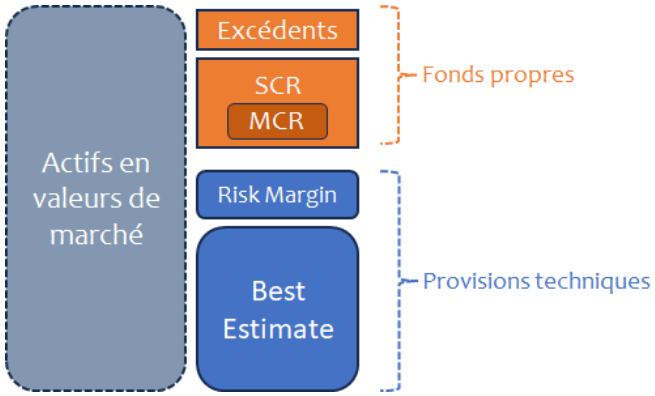

FIGURE 1.2 – Bilan simplifié Solvabilité II

En accord avec ce qui a été dit à la section 1.2.1, Solvabilité II introduit des outils de mesure de la stabilité des organismes d'assurance à travers son pilier 1. Il s'agit notamment du **SCR**, du **MCR** (défini à la section 1.2.1) et du ***Best Estimate***.

En pratique, la détermination des fonds propres disponibles passe par l'évaluation des provisions techniques qui ne sont pas directement observées sur le marché. Ainsi, son calcul s'effectue en sommant la marge de risque (*risk margin*) et une quantité correspondant à la meilleure estimation (Best Estimate) des flux futurs actualisés espérés par l'entité assurantielle.

La Marge pour risque est assimilée au coût du capital nécessaire à la couverture de l'exigence en capital supplémentaire. De manière formelle, il se calcule selon la formule suivante :

$$\text{Risk Margin} = \text{CoC} \sum_{t \geq 0} \frac{\text{SCR}_t}{(1 + r_{t+1})^{t+1}} \tag{1.1}$$

avec :

$$\begin{cases} \text{CoC} & \text{le coût du capital, réglementairement fixé à 6\%} \\ \text{SCR}_t & \text{l'exigence en capital supplémentaire} \\ r_t & \text{le taux sans risque à l'instant } t \end{cases}$$

Nous nous attarderons dans la suite (en termes de métriques) sur le *Best Estimate*, le

SCR et le ratio de Solvabilité (le quotient fonds propres / SCR) au vu de la place central qu'ils occupent dans nos travaux.

Solvency Capital Requirement (SCR)

Le SCR (voir 1.2.1), est une métrique qui permet au régulateur de juger si la compagnie peut faire face à ses engagements sous un horizon d'un an. Cette métrique peut se calculer en formule standard ou en modèle interne.

Calcul en formule standard

En formule standard, on suppose que le profil de risque de l'entité assurantielle est le profil moyen d'un assureur au niveau européen. Sous cette supposition, le calcul du SCR se fait en appliquant des chocs au scénario central. En fait, le capital de l'assureur doit pouvoir faire face à un choc qui correspond au pire scénario sur 200 qu'on teste, c'est le quantile à 99,5 %.

La formule standard est la méthode conventionnelle au niveau européen afin de permettre une comparaison entre les organismes. Elle est simple à mettre en œuvre et peut être appliquée sans justification (contrairement au modèle interne).

Sa mise en œuvre est modulaire et dite *Bottom-up* (de bas vers le haut) au regard de la "pieuvre" récapitulative de tous les modules ci-dessous :

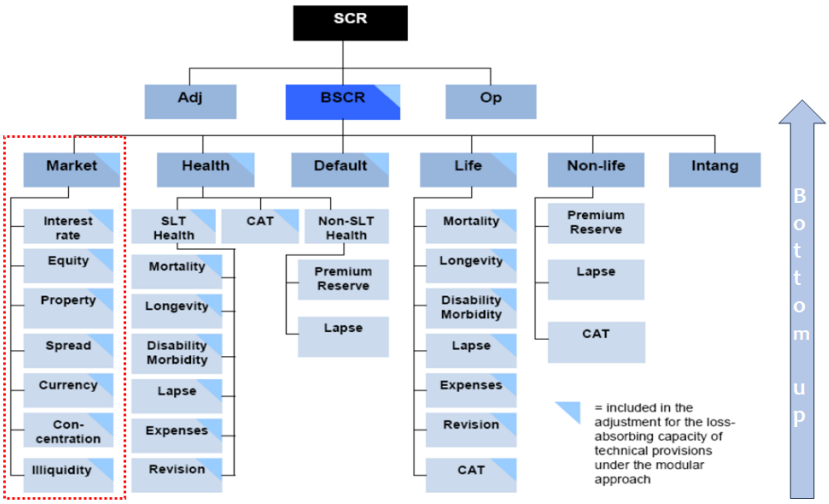

FIGURE 1.3 – Modules SII. Image récupérée sur le site ACPR.

Sa mise en œuvre peut se résumer aux étapes suivantes :

- **Etape 1 :** Détermination d'un capital économique pour chaque risque élémentaire (par exemple : taux, action) via une différence entre la *Net Asset Value* centrale

et la *Net Asset Value* choquée. Les différents chocs correspondent à des déviations extrêmes calibrées par le regulateur sous un historique.

- **Etape 2 :** Agrégation intra-modulaire via une matrice de corrélation afin de prendre en compte les interdépendances éventuelles. On procède à une agrégation des différents capitaux au sein de chaque sous-module (marché, vie, etc).

- **Etape 3 :** Agrégation inter-modulaire par matrice de corrélation. On agrège juste les capitaux des différents modules.

Bien que cette approche soit simple, elle a aussi des limites : les différents chocs sont calibrés arbitrairement et ne sont pas forcément adaptés à tous les organismes d'assurance puisque ceux-ci peuvent avoir des profils de risques différents de la moyenne européenne ; de plus, certains risques ne sont pas pris en compte par la formule standard comme le risque souverain.

Calcul en modèle interne

Au vu des limites de la formule standard, si un organisme estime que la formule standard ne correspond pas parfaitement à son profil de risque, il peut effectuer une demande auprès de l'autorité de régulation afin de calculer son SCR sous un modèle interne total ou partiel.

En modèle interne, le calcul du SCR passe par la détermination de la distribution des fonds propres économiques à horizon 1 an puisque le SCR s'appréhende conceptuellement comme la *Value at Risk* à 99,5 % de ces fonds propres.

Dans notre contexte, la détermination du SCR se fera en formule standard.

Ratio de Solvabilité

Il s'agit du rapport entre les fonds propres et le SCR. L'intérêt de cette métrique dans le contexte assurantiel est crucial, car elle fait office de véritable outil de pilotage.

Sous le paradigme rendement – risque introduit par Markowitz [Markowitz, 1952], il apparaît que pour un portefeuille d'actifs donné, en raison de l'aversion au risque de l'investisseur, pour un niveau de rendement identique, il optera pour l'investissement le moins risqué. Toutefois, rien n'est spécifié lorsque l'actif interagit avec le passif comme c'est le cas en assurance vie. Un tel état de faits suppose de s'intéresser à une métrique qui intègre cette interaction entre actif et passif afin de mieux appréhender le niveau de risque/choc que l'assureur peut continuer à supporter.

Les fonds propres étant la résultante d'une interaction entre actif et passif peuvent se prêter au jeu. En effectuant le rapport avec le SCR, on a une métrique utile puisque lorsque le ratio vaut 1, cela signifie que les fonds propres permettent juste de couvrir

l'exigence en capital. Avoir le ratio de solvabilité le plus maximal possible permet dès lors à l'assureur de faire face à d'éventuels chocs.

Best Estimate

Le *Best Estimate* est défini par le régulateur (Article 77, directive Solvabilité II) comme "*La moyenne des flux de trésorerie futurs, pondérés par leur probabilité et compte tenu de la valeur temporelle de l'argent, estimée sur la base de la courbe des taux sans risques pertinents.*"

Dans le référentiel du calcul du Best Estimate, on distingue le *Best Estimate* déterministe (BE_{det}) du *Best Estimate* stochastique (BE_{sto}).

Le *Best Estimate* (BE) déterministe correspond à la somme simpliste des flux futurs actualisés selon un scénario déterministe (désigné généralement par scénario central). A titre d'exemple, sous ce prisme, les actifs de type action ou immobilier ont des rendements correspondant à leurs rendements moyens. En revanche, le *Best Estimate* stochastique est le calcul de la meilleure estimation des flux futurs actualisés suivant un grand nombre de scénarios aléatoires.

Le régulateur préconise aux organismes d'assurance vie d'avoir recours au Best Estimate stochastique afin de tenir compte de la valeur temporelle des options et garanties [Européenne, 2015]. En effet, en considérant un seul scénario central déterministe, il peut s'avérer que certaines options des différents contrats du portefeuille ne se déclenchent pas alors qu'en réalité en cas de contexte économique défavorable, ces mêmes options qui ne se sont pas déclenchées peuvent l'être.

Pour rester fidèle à la vision de prudence de Solvabilité II qui concerne la quantification plus réaliste des risques et l'évaluation plus juste des actifs et des passifs, des simulations doivent être effectuées afin d'incorporer les incertitudes liés aux différentes garanties. Plus exactement, en notant $TVOG$ (*Time Value of Options and Guarantees*), la valeur temporelle des options et garanties des différents contrats, on a :

$TVOG$ = *Best Estimate* stochastique - *Best Estimate* déterministe.

Pour bien cerner cette notion de TVOG dans le calcul du *Best Estimate*, considérons l'exemple chiffré (fictif) suivant :

Soit deux contrats X et Y d'un portefeuille épargne-retraite avec les caractéristiques ci-après :

	X	Y
Taux garanti	0%	2%
PB	85%	85%

Taux sans risque	4%
Rendement des actifs	4%
PM initiale	1000
Maturité	1 an
Frais et impôts	0

TABLE 1.1 – Caractéristiques des contrats fictifs

Le calcul du *Best Estimate* déterministe donne :

	X	Y
Produits financiers (PF)	4% PM = 40	4% PM = 40
PB	Max(85% 40 ;0% PM) = 34	Max(85 % 40 ;3 % PM) = 34
Prestations (Pres.)	1000+34 = 1034	1000+34 = 1034
Frais	0	0
Best Estimate	1034/(1+4 %) = 994,2	1034/(1+4 %) = 994,2

TABLE 1.2 – Calcul du *Best Estimate* déterministe des contrats fictifs

Pour le *Best Estimate* stochastique, supposons que les actifs rapportent les rendements moyens ci-après (selon 3 scénarios équiprobables) :

- Scénario 1 : rendement = 4 %
- Scénario 2 : rendement = 6 %
- Scénario 3 : rendement = 1 %

On obtient :

	scénario 1		**scénario 2**		**scénario 3**	
	X	Y	X	Y	X	Y
PF	4% PM = 40	40	6% PM = 60	60	1% PM = 10	10
PB	34	34	51	51	8,5	30
Pres.	1034	1034	1051	1051	1008,5	1030
Pres. actualisées	994,2	994,2	991,5	991,5	998,5	1019,8

TABLE 1.3 – Calcul du *Best Estimate* stochastique des contrats fictifs

En définitive, on a les résultats suivants :

	X	Y
BE_{det}	994,2	994,2
BE_{sto}	(994,2+991,5+998,5)/3= 994,7	(994,2+991,5+1019,8)/3 = 1001,8
TVOG	**0,5**	**7,6**

TABLE 1.4 – Résultat final

Ainsi, avec cet exemple fictif, on appréhende mieux la nécessité du stochastique pour la prise en compte du TVOG dans l'estimation du Best Estimate dans les contrats épargne-retraite.

En pratique, nous avons aussi la relation : $BEL = BEG + FDB$ où le BEG est le BE calculé au taux moyen garanti avant participation aux bénéfices et la FDB la partie discrétionnaire qui intègre la participation aux bénéfices.

Les différentes métriques d'intérêt introduites, nous proposons dans la suite une formalisation plus rigoureuse du *Best Estimate* ainsi que des méthodes d'estimation dans la littérature puisque ce dernier est l'essence de notre étude.

1.3 Etat de l'art des techniques d'estimation du Best Estimate

L'objectif de cette partie est d'introduire le *Best Estimate* dans un cadre formel et rigoureux, puis d'exposer quelques travaux s'inscrivant dans le sens de son approximation.

1.3.1 Généralités sur le Best Estimate

Probabilité risque neutre et historique

Solvabilité II introduit dans le cadre de la valorisation des passifs d'assurance, deux univers de probabilité : *risque neutre* et *monde réel*.

Ces deux univers trouvent leurs intérêts dans la manière dont Solvabilité II exige de valoriser le passif. Toujours selon l'article 77, la valorisation doit se faire selon des méthodes qui permettent de capturer l'effet du temps sur les garanties embarquées. Ces garanties embarquées ressemblent à des options financières, d'autant plus que les différentes revalorisations se font en fonction des performances des actifs financiers. De ce fait, l'évaluation de ce passif peut "s'apparenter" à celle d'options financières.

La valorisation des options financières s'inscrit dans une littérature très spécifique depuis les travaux de [Scholes *et al.*, 1973] et laisse ressortir le fait que le prix d'une option est l'espérance de ses flux futurs (*payoffs*) sous une probabilité dite *risque neutre*.

Cette philosophie d'évaluation issue de la finance de marché a influencé l'assurance à partir de la fin des années 1980 début de 1990. En particulier les travaux de Bryis en 1994 ont établi [Bryis, 1994] le lien entre les flux de certains contrats d'assurance embarquant des participations aux bénéfices et des options financières. Ces derniers ont alors proposé l'extension de ce cadre financier à l'assurance vie.

A titre d'exemple [Bryis, 1994] :

- Les options de rachat sont semblables à des *put* (option de vente) américain ;
- Les options de taux technique et de participation aux bénéfices ressemblent à des options européennes standards ;
- Les options de versement libres ou programmés s'apparentent à des *swaptions*.

C'est cette approche de valorisation qui a été adoptée comme principe de valorisation sous Solvabilité II.

A ce stade, il est convenable d'apporter quelques clarifications sur cette probabilité risque neutre. Les principales hypothèses sous-jacentes à l'utilisation de cette probabilité dans la valorisation de produits financiers sont :

- Le marché est complet : on peut toujours trouver un portefeuille d'actifs permettant de répliquer parfaitement les flux d'un produit financier donné.
- Il y a Absence d'Opportunité d'Arbitrage (AOA) : il n'est pas possible de réaliser un bénéfice strictement positif de manière certaine en partant d'un investissement nul.

En supposant les deux conditions précédentes, [Harrisson, 1979, Pliska, 1981] prouvent que la mesure de probabilité risque neutre, est l'unique mesure équivalente au sens de Radon-Nikodym à la probabilité historique et qui assure le caractère martingale des prix actualisés au taux sans risque (voir la théorie du calcul stochastique [Lamberton, 2012]).

La probabilité historique aussi appelée probabilité réelle est l'univers de probabilité qui tient compte de la réalité des variations de marché dans la valeur des actifs. Il y a notamment la notion de prime de risque qui est prise en compte sous cet univers puisqu'en pratique plus un actif est risqué plus l'investisseur a tendance à demander une compensation pour le risque qu'il prend. Dans l'univers théorique de probabilité dite "risque-neutre", l'investisseur arbitre indifféremment entre les actifs car dans cet espace, leur espérance de taux de rendement est identique.

L'utilité de ces deux mesures de probabilité est justifiée par la nature des risques en assurance vie. Comme introduit précédemment, la mesure risque neutre intervient dans une optique de valorisation et permet la prise en compte des risques purement financiers qui sont en théorie réplicables et la mesure monde réel permet la prise en compte des risques non réplicables spécifiques à l'assurance vie.

Formalisme

Considérons les notations suivantes :

- F_t les flux de l'année t ;
- r_t le taux sans risque à la date t ;
- $\mathbb{P}$ la probabilité historique associée aux risques non réplicables ;
- $\mathbb{Q}$ la probabilité risque-neutre associée aux risques financiers réplicables ;
- $(\mathcal{F}_t)_t$ la filtration naturelle associée à l'ensemble des processus permettant de définir l'environnement économique de l'entreprise.

A l'instant $t = 0$, le *Best Estimate (BE)* s'écrit :

$$\mathrm{BE}_0 = \mathbb{E}^{\mathbb{Q}} \left[\sum_{s=0}^{\infty} \frac{F_s}{(1+r_s)^s} \right]$$

Pour un instant $t > 0$, on a :

$$\mathrm{BE}_t = \mathbb{E}^{\mathbb{P}\otimes\mathbb{Q}} \left[\sum_{s=t}^{\infty} \frac{F_s}{(1+r_t)^{s-t}} \middle| \mathcal{F}_t \right]$$

Pour simplifier, en pratique,

$$\mathrm{BE}_t = \mathbb{E}^{\mathbb{P}} \left[\mathbb{E}^{\mathbb{Q}} \left[\sum_{s=t}^{\infty} \frac{F_s}{(1+r_t)^{s-t}} \middle| \mathcal{F}_t \right] \right]$$

En notant : $X_t = \mathbb{E}^{\mathbb{Q}} \left[\sum_{s=t}^{\infty} \frac{F_s}{(1+r_s)^{s-t}} \mid \mathcal{F}_t \right]$, on a : $\mathrm{BE}_t = \mathbb{E}^{\mathbb{P}} [\mathrm{X}_t]$.

Le problème s'appréhende comme le calcul de l'espérance d'une variable aléatoire. Les différentes approches qui seront présentées dans la suite sont juste des manières d'estimer cette quantité.

1.3.2 Méthode standard de Simulation dans les Simulations (SdS)

Le principe général consiste en une application imbriquée de techniques de Monte-Carlo. La théorie générale de l'approche Monte-Carlo consiste à approcher l'espérance d'un processus aléatoire à partir d'échantillons indépendants et identiquement distribués de ce processus aléatoire. La garantie théorique de cette approche s'appuie sur la loi des grands nombres.

De manière formelle, si on considère une variable aléatoire Z d'espérance finie, alors pour tout échantillon (issue de Z) de variables indépendantes et identiquement distribuées $Z_1, \ldots, Z_n$ ($n \in \mathbb{N}^*$), on a : $\frac{1}{n} \sum_{j=1}^{n} Z_j \to \mathbb{E}(Z)$, presque surement (loi forte).

Le principe de la SdS est le suivant :

Etape 1 : On se place sous la probabilité historique $\mathbb{P}$ et on simule $N \in \mathbb{N}^*$ fois l'ensemble des processus qui définissent la filtration $(\mathcal{F}_s)_{s\geq 0}$.
On obtient ainsi N scénarios traduisant l'environnement économique probable en t.

Supposons que les N scénarios soient définis selon la filtration $(\mathcal{F}_s^n)_{n=1,\dots,N}$.

Etape 2 : On se place sous la probabilité risque neutre $\mathbb{Q}$ et on simule pour chaque état du monde $(\mathcal{F}_s^1, \mathcal{F}_s^2, ..., \mathcal{F}_s^N)$, $K \in \mathbb{N}$ scénarios sur un horizon $s = t, \dots, T$ (T très grand) puis on effectue une approximation par la méthode de Monte-Carlo.

En considérant chacune des filtrations, on peut construire une suite $(X_t^n)_{n=1,\dots,N}$ telle que :

$$\mathrm{X}_t^n = \mathbb{E}^{\mathbb{Q}}\left[\sum_{s=t}^{\infty} \frac{F_s}{(1+r_s)^{s-t}} \middle| \mathcal{F}_t^n\right] \approx \frac{1}{K}\sum_{k=1}^{K}\sum_{s=t}^{T} \frac{F_s^k}{(1+r_t)^{s-t}} := \widehat{\mathrm{X}_t^n}$$

Soit,

$$\mathbb{E}^{\mathbb{P}}\left[\mathrm{X}_t\right] \approx \frac{1}{N}\sum_{n=1}^{N} \widehat{\mathrm{X}_t^n}$$

Finalement :

$$\mathbb{E}^{\mathbb{P}}\left[\mathrm{X}_t\right] \approx \frac{1}{N\times K}\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{s=t}^{T} \frac{F_s^k}{(1+r_s)^{s-t}} \tag{1.2}$$

Ainsi, la complexité algorithmique de (1.2) est : $\mathcal{O}(N \times K \times (T-t+1))$. Les valeurs de N, K et T étant grandes en pratique (convergence de l'estimateur), il en est de même pour le temps de calcul. Cette approche est résumée par la figure 1.4.

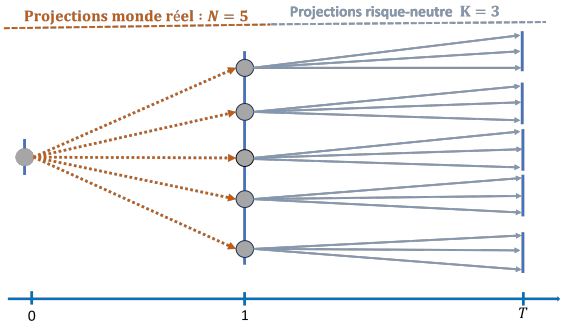

FIGURE 1.4 – Illustration de la méthode SdS

1.3.3 Approches paramétriques : une alternative aux SdS

Dans la littérature, on retrouve comme substitut à l'approche SdS, les méthodes paramétriques de type *Least Squares Monte Carlo* et le *Curve Fitting* [Rouchati, 2016].

Least Squares Monte Carlo (LSMC)

Cette méthode a été populaire à l'issue des travaux de [Longstaff, 2001] qui l'ont utilisé dans le cadre de l'évaluation d'options américaines, bien qu'introduite en 1996 par [Carriere, 1996]. Dans le contexte de l'évaluation sous Solvabilité II, ce sont les travaux de [Bauer, 2010] qui ont été les précurseurs.

L'idée centrale de cette approche concerne la réduction du nombre de simulations risque-neutre utilisé dans le cadre de l'approche de Simulation dans les Simulations puis d'effectuer une estimation par régression. En effet, réduire le nombre de simulations risque-neutre pour chacune des N simulations monde réel réduit le temps de calcul, mais implique une imprécision dans l'estimation. Pour prendre en compte cette imprécision, une régression par moindres carrées est effectuée (calibrage de formes paramétriques polynomiales). Cette technique peut être illustrée par le graphique suivant :

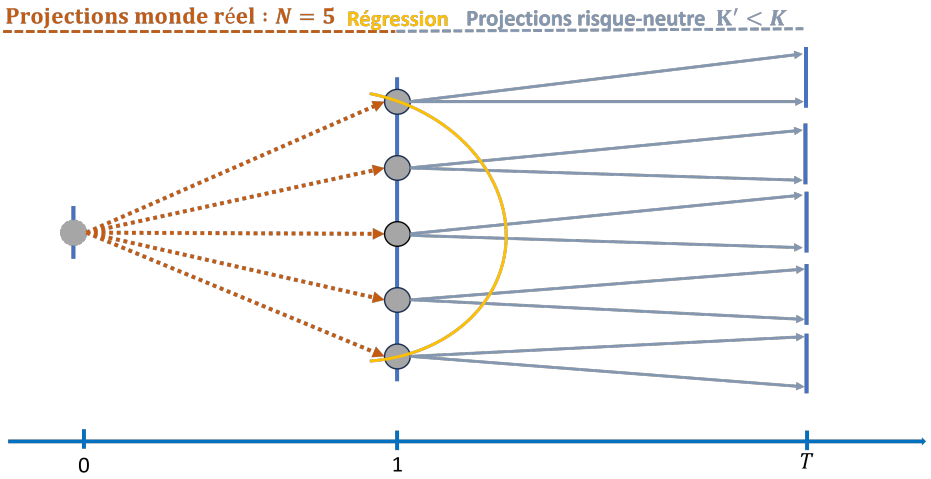

FIGURE 1.5 – Illustration de la méthode LSMC

Curve Fitting

Cette approche dans la littérature actuarielle remonte aux travaux de [Koursaris, 2011].

Le *Curve Fitting* s'inscrit dans le même ordre d'idée que le LSMC consistant à calibrer une forme paramétrique qui permet d'expliquer le *Best Estimate* en fonction des facteurs de risques.

Avec cette méthode, le nombre de simulations en monde réel est réduit et l'estimation est faite en utilisant une interpolation plutôt qu'une régression. En effet, il est question ici d'interpolation, car on dispose de quelques données précises et on souhaite estimer les valeurs entre ces points de données de manière efficace contrairement à la régression où on cherche une fonction qui s'ajuste au mieux notre ensemble de données, mais sans forcément passer par chaque point. Cette approche s'illustre par le graphique suivant :

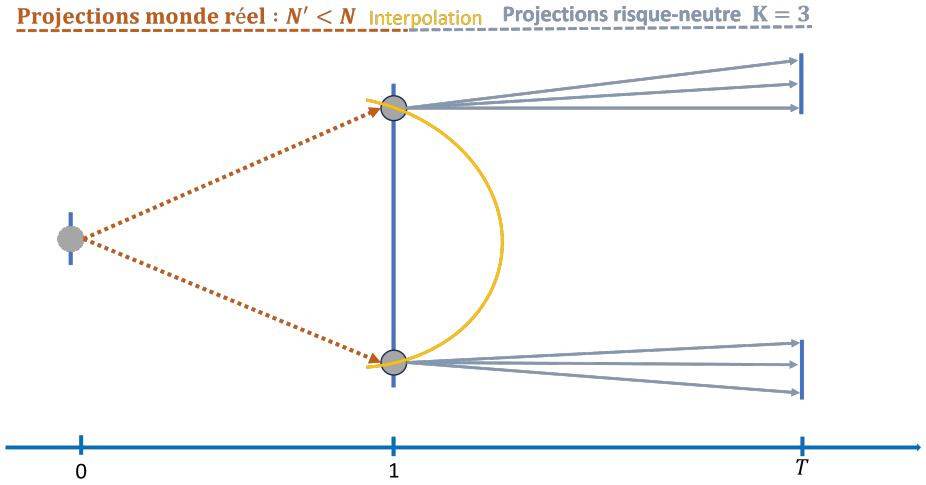

FIGURE 1.6 – Illustration de la méthode *Curve fitting*

De manière générale, toutes les approches qui intègrent l'estimateur de Monte-Carlo peuvent faire l'objet d'amélioration en utilisant des techniques de réduction de variance ou des méthodes de type *quasi Monte-Carlo* afin d'augmenter la vitesse de convergence de l'estimateur.

1.3.4 *Replicating portfolio*

Une méthode dans la littérature consiste à utiliser des « portefeuilles répliquants ». Ces portefeuilles répliquants font allusion à la mise en place de portefeuille d'actifs pour reproduire les flux de passifs pour les différentes simulations. Un approfondissement du sujet est disponible dans [Schrager, 2008].

1.3.5 Approximations basées sur le *Machine Learning*

Le *Machine Learning* est une branche de l'intelligence artificielle qui vise à mettre en place des algorithmes permettant à une machine ou un ordinateur d'accomplir des tâches spécifiques sans être explicitement programmé.

En se basant sur des données, ces algorithmes apprennent à détecter des *patterns*, faire des prédictions, ou prendre des décisions. On distingue dans la manière d'apprendre de ces algorithmes essentiellement des techniques dites d'apprentissage supervisé, non supervisé et d'apprentissage par renforcement.

- **L'apprentissage supervisé :** dans ce cadre, les données contiennent un jeu de variables qui expliquent à priori le phénomène modélisé et la variable à expliquer elle-même. Le but de l'algorithme sera de déceler la fonction qui lie la variable à expliquer et les variables explicatives. C'est l'univers classique lorsque l'objectif est d'effectuer une régression.

- **L'apprentissage non supervisé :** sous ce paradigme, les données ne sont pas explicitement scindées en variables explicatives et variable à expliquer. Le rôle de l'algorithme sera d'étiqueter les données en partant de rien. C'est typiquement l'environnement de modélisation pour des tâches de classification.

- **L'apprentissage par renforcement :** se focalise sur la manière dont un agent (algorithme, etc) peut apprendre à décider de manière séquentielle en interagissant avec un environnement. Cette approche diffère des techniques d'apprentissage supervisé et non supervisé en ce sens qu'elle implique une rétroaction directe sous forme de récompenses ou de punitions pour guider l'apprentissage, plutôt que de s'appuyer uniquement sur des données statiques ou des labels prédéfinis.

Dans la littérature actuarielle portant sur l'approximation du $BestEstimate$, les modèles de *Machine Learning* qu'on trouve sont généralement des modèles d'apprentissage supervisé (exemple de [Morisse, 2022]).

Considérons X la matrice représentant les variables explicatives contenues dans la base de données et Y le vecteur représentant les observations de la variable à expliquer (exemple du BE dans notre cas) avec $d \in \mathbb{N}^*$ le nombre de variables explicatives, $n \in \mathbb{N}^*$ le nombre d'observations.

L'idée fondamentale de ces modèles proxy de *Machine Learning* est de trouver une fonction $f : \mathbb{R}^{n \times d} \longrightarrow \mathbb{R}^p$ telle que : $Y \approx f(X)$, p étant la dimension de la variable cible. La fonction f peut être un krigeage, un réseau de neurones ou tout autre modèle.

Nos travaux s'inscrivent dans le cadre des méthodes d'approximation fondées sur le *Machine Learning*. Nous proposons d'étendre ces modèles au cadre génératif dans un contexte d'approximation du *Best Estimate*.

Chapitre 2

Modèles génératifs

L'objectif de ce chapitre est d'exposer le socle théorique inhérent à la modélisation générative. Nous débuterons par une présentation de cette classe de modèles, avant d'aborder les aspects théoriques sous-jacents.

2.1 Introduction

Les modèles génératifs sont des types de modèles en apprentissage automatique qui modélisent la distribution des données sur lesquelles ils sont entraînés. Ils peuvent ensuite générer de nouvelles données qui ressemblent à l'historique d'entraînement, au sens probabiliste et statistique.

Dans notre contexte, une distribution représente la manière dont les observations d'une variable aléatoire sont réparties. Elle décrit les probabilités associées à l'ensemble des valeurs que peut prendre notre variable.

Ainsi, en cherchant principalement à apprendre une distribution, les modèles génératifs modélisent le processus sous-jacent qui a permis d'obtenir les données. En effet, ils capturent les mécanismes et les interactions qui génèrent les données observées, leur permettant ainsi de créer de nouveaux échantillons cohérents (selon certains critères) avec les données originales.

De nombreuses applications sont possibles, par exemple en assurance, en dehors de notre étude, les modèles génératifs peuvent être utilisé pour divers buts :

- Modélisation des risques : [Solveig *et al.*, 2022] laisse entrevoir la possibilité d'utiliser des modèles à capacité générative afin de générer des scénarios économiques utilisés dans les modèles de risque de marché des compagnies qui doivent répondre aux exigences de Solvabilité II.

- Détection de fraudes et d'anomalies : les réseaux de neurones génératifs peuvent modéliser les distributions des données légitimes, permettant ainsi de repérer les éléments qui s'éloignent du comportement attendu, ce qui peut indiquer une fraude potentielle.

- Complétion de données : ces modèles sont de bons candidats si l'on souhaite augmenter des bases de données actuarielles, effectuer des imputations de données manquantes de manière réaliste [Colvez *et al.*, 2022] et d'autres utilisations comme celles énumérées dans [Ngwenduna, 2020].

Outre l'assurance, les modèles génératifs trouvent de l'intérêt dans d'autres domaines :

- L'une de leurs applications les plus populaires concerne la génération, la traduction, l'amélioration de contenus visuels et sonores. Un exemple très connu est l'outil *CycleGAN*. Dans [Zhu *et al.*, 2017], les auteurs proposent d'utiliser une architecture générative qui permet de transformer des images de différents styles sans nécessiter de correspondances exactes entre les images d'entraînement, offrant ainsi de nombreuses possibilités en traitement et création d'images.

- On retrouve de nombreuses applications dans le domaine de la finance de marché. Par exemple, dans la couverture financière, les modèles génératifs sont notamment utilisés pour améliorer la précision et l'efficacité des stratégies de couverture depuis le papier séminal sur le *deep hedging* [Hans *et al.*, 2018].
  Dans ce papier, il est question d'approximer les décisions de trading de produits dérivés via des réseaux de neurones dans un contexte d'apprentissage par renforcement, permettant ainsi d'optimiser les stratégies de couverture dans des environnements de marché complexes et réalistes. La méthode étant *data-driven*, pour un apprentissage efficace de l'agent, il faut une grande quantité de données qui traduisent différents scénarios. Pour palier ce problème, les *deep generators* (modèles de simulations de conditions de marché) basés sur des modèles génératifs se sont avérés très utiles puisqu'ils sont capables de simuler des environnements de marché détaillés et très réalistes, permettent aux stratégies de *deep hedging* d'être testées et ajustées de manière continue, assurant une réponse efficace aux fluctuations de marché.

- Pour finir, en imagerie médicale, ces modèles sont très utiles dans l'amélioration de la qualité des images produites en reconstruction d'images ou pour la détection et la classification automatique de pathologies [Kun *et al.*, 2022].

2.1.1 Distinction entre modèles génératifs et discriminatifs

Soit une variable aléatoire Y représentant un phénomène d'intérêt et X un ensemble de variables qui pourraient à priori avoir un lien avec Y. En apprentissage automatique, X désigne le vecteur de variables explicatives et Y la variable à expliquer ou la cible. Il s'agit ici d'un contexte d'apprentissage supervisé.

Dans l'approche discriminative, l'objectif est d'identifier les valeurs prises par la variable Y selon les caractéristiques de X. En d'autres termes, ces modèles tentent de répondre à la question : connaissant X, quelles sont les valeurs probables de Y ? Il s'agit donc de modéliser la loi conditionnelle de Y sachant X. Tandis qu'avec les modèles génératifs, on cherche à déterminer les valeurs de la variable X qui ont permis d'avoir Y. Il est question de modéliser la loi jointe de (X, Y) puis d'en déduire la loi conditionnelle $(X|Y)$. Un approfondissement de cette distinction peut être trouvée dans [Jebara, 2012].

En plus de cet aspect, les modèles génératifs sont capables de modéliser directement Y si l'on souhaite avoir des données semblables. En d'autres termes, ils sont aptes (en théorie) à trouver des caractéristiques intrinsèques qui permettent d'obtenir Y. Lorsqu'on modélise directement Y, on parle de modèle génératif et quand il s'agit de $(X|Y)$, on parle de **modèles génératifs conditionnels**.

Après avoir exposé ces distinctions fondamentales, nous passons à la théorie des réseaux de neurones, qui constituent les éléments de base des modèles génératifs.

2.1.2 Rappels et généralités sur les réseaux de neurones

Les réseaux de neurones artificiels, apparus au milieu du $20^{\text{ième}}$ siècle, sont une modélisation mathématique inspirée du fonctionnement neuronal humain, permettant de résoudre numériquement certains problèmes.

Les travaux pionniers de McCulloch et Pitts en 1943 ([Mcculloch *et al.*, 1943]) ont formalisé cette analogie avec le neurone réel. Par la suite, diverses recherches ont suivi, avec des débuts modestes, mais prometteurs dans les années 1950, suivis de périodes de stagnation et de renouveau, culminant avec les avancées spectaculaires de l'apprentissage profond au $21^{\text{ième}}$ siècle. Une revue détaillée sur l'essor de l'intelligence artificielle peut être consultée dans [Cardon *et al.*, 2018].

Les réseaux de neurones comme approximateurs de fonctions

Un aspect crucial pour comprendre le fonctionnement des réseaux de neurones est leur capacité à approximer des fonctions. Bien que l'on dispose généralement de données d'entrée et de données cibles, la fonction sous-jacente reliant ces deux ensembles est souvent inconnue. Le réseau de neurones a pour objectif d'estimer cette fonction, lui permettant ainsi de prédire des sorties pour de nouvelles entrées qui n'étaient pas présentes

dans les données d'entraînement initiales.

La garantie théorique de cette capacité d'approximation est formalisée par le théorème d'approximation universel (voir [Cybenko, 1989] et [Hornik *et al.*, 1989]), qui stipule que : "*sous certaines conditions, un réseau de neurones à une seule couche cachée avec un nombre fini de neurones peut approximer n'importe quelle fonction continue sur un intervalle donné avec une précision arbitraire*".

Ce théorème, prouvé par Cybenko pour les fonctions sigmoïdes et étendu par Hornik, met en évidence la puissante capacité de modélisation des réseaux de neurones puisque théoriquement il nous suffit de reussir à représenter la relation entre les variables d'entrée et de sortie sous forme de fonction continue. En pratique, cela implique que les réseaux de neurones, avec un entraînement approprié et une architecture adéquate, peuvent généraliser des résultats même pour des données qu'ils n'ont jamais vues.

Fonctionnement et optimisation d'un réseau de neurones

Dans un réseau de neurones artificiel, chaque neurone reçoit des entrées, effectue des transformations et utilise une fonction de seuil pour déterminer la sortie. Ces éléments constituent quelques bases fondamentales des réseaux de neurones qu'il est essentiel de présenter.

- **Neurone :** un neurone peut s'interpréter comme une unité de traitement de l'information. Il reçoit un vecteur d'entrée $x = [x_1, x_2, \ldots, x_n]$, effectue une combinaison linéaire pondérée $z = \sum_{i=1}^{n} w_i x_i + b$, puis applique une fonction d'activation $\sigma(z)$ pour produire la sortie $\mathbf{o}$. Ici :
  - w_i sont les poids associés à chaque entrée x_i ;
  - b désigne le biais. C'est une constante qui permet au modèle d'être plus flexible et de mieux s'adapter aux données ;
  - $\sigma(z)$ est la fonction d'activation qui introduit la non-linéarité afin de modéliser des relations complexes.

- **Couche :** une couche se définit comme un ensemble de neurones. On distingue néanmoins plusieurs types de couches : la couche d'entrée, qui reçoit les données initiales ; la couche de sortie, qui fournit la prédiction finale du modèle ; et les couches cachées, qui sont des couches intermédiaires qui appliquent des transformations non-linéaires. Pour une couche l, la sortie est définie par $\mathbf{o}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{o}^{(l-1)} + \mathbf{b}^{(l)})$, où $\mathbf{W}^{(l)}$ est la matrice des poids, $\mathbf{b}^{(l)}$ le vecteur des biais et $\mathbf{o}^{(l-1)}$ la sortie de la couche précédente $l-1$. Ce fonctionnement est illustré par la figure 2.1.

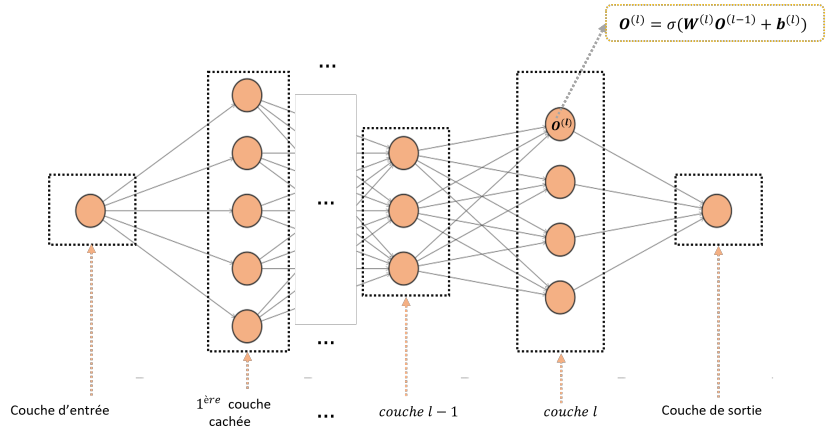

FIGURE 2.1 – Illustration du processus d'optimisation des réseaux de neurones

Pour que les réseaux de neurones fonctionnent efficacement, leurs paramètres $\theta = \{\mathbf{W}, \mathbf{b}\}$ doivent être ajustés sur des données d'entraînement pour minimiser l'écart entre les prédictions et les valeurs réelles, en utilisant une fonction d'erreur $J(\theta)$ et des techniques d'optimisation. L'objectif est donc de résoudre :

$$\theta^* = \arg\min_{\theta} J(\theta)$$

Cela se fait par des techniques de descente de gradient, où les paramètres sont mis à jour selon la règle : $\theta_{t+1} := \theta_t - \eta \nabla J(\theta_t)$, avec $\nabla J(\theta_t)$ représentant le gradient de la fonction d'erreur par rapport aux paramètres θ à l'itération t et $\eta > 0$ le pas d'apprentissage. Pour améliorer l'efficacité et la rapidité de la convergence, des techniques d'optimisation avancées, comme l'*Adaptive Moment Estimation* (ADAM), [Kingma *et al.*, 2014a] sont souvent utilisées. Ci-dessous une illustration du processus d'optimisation :

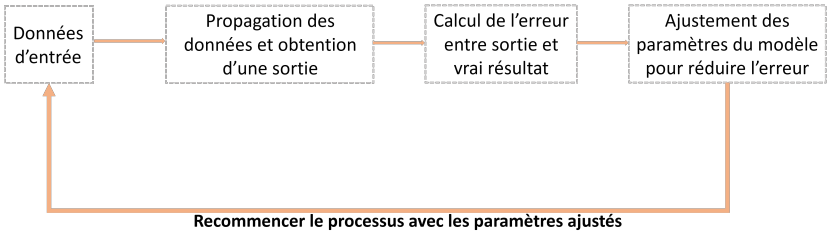

FIGURE 2.2 – Illustration du processus d'optimisation des réseaux de neurones

Une fois les fondamentaux du fonctionnement et de l'optimisation des réseaux de neurones posés, il est essentiel de savoir construire un réseau en fonction du besoin, on parle alors de choix d'architectures.

Dans la littérature, de nombreuses architectures existent, chacune adaptée à des types de problèmes spécifiques. Nous en exposerons quelques-unes.

2.1.3 Architectures de réseaux de neurones

Dans cette partie, nous nous affranchirons délibérément de la théorie rigoureuse en donnant uniquement des descriptions générales des différentes architectures. Un approfondissement mathématique des architectures retenues pour notre application est donnée dans la section 2.4.

Réseaux de neurones à propagation avant

Les réseaux de neurones à propagation avant constituent une classe de modèles où l'information circule dans une seule direction, de l'entrée vers la sortie, sans rétroaction. Un exemple est le perceptron multicouches (MLP). C'est une extension du perceptron simple, introduit par Frank Rosenblatt en 1958 [Rosenblatt, 1958]. Ce type de réseau a gagné en popularité dans les années 1980 grâce à l'algorithme de rétropropagation introduit par [Rumelhart *et al.*, 1986]. Dans la version de base, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante (réseau *fully connected*) comme sur l'illustration 2.1.

Réseaux de neurones convolutionnels (CNN)

Bien que leur concept de base existe depuis les années 1980 (voir les travaux de [Fukushima et J., 1980] par exemple), c'est à l'issue des travaux de [LeCun *et al.*, 1998] qu'ils ont connu leur popularité. Avec leur forte capacité d'extraction de caractéristiques (*features*) ils sont largement utilisés dans des tâches de reconnaissance faciale, de détection d'objets et même dans les voitures autonomes. Un CNN, dans sa version simplifiée, est composée de couches de convolution, de couches de *pooling* et de couches entièrement connectées. Les couches de convolution appliquent des filtres pour extraire les caractéristiques des images, tandis que les couches de *pooling* réduisent la dimensionnalité.

Réseaux de Neurones Récurrents (RNN)

Les RNNs (ou *Recurrent Neural Networks*) sont une classe de réseaux de neurones conçus pour traiter des données séquentielles. Contrairement aux réseaux de neurones traditionnels (réseau à propagation avant), les RNNs ont la capacité de conserver une mémoire interne des informations traitées précédemment, ce qui les rend particulièrement adaptés pour des tâches où le contexte historique est important.

En ce qui concerne les problématiques où il est question de "générer", une classe de modèles plus appropriée existe, et c'est de celle-ci que traitent les sections suivantes.

2.1.4 Vue d'ensemble des modèles génératifs

Pour rappel, l'objectif de la modélisation générative est de capturer la distribution sous-jacente des données observées afin de pouvoir générer de nouvelles données similaires, conformément aux métriques définies en fonction de la tâche.

Formellement, étant donné un ensemble de données $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$ échantillonnées à partir d'une distribution inconnue $p_{\mathcal{D}}(x)$, l'objectif est d'apprendre une distribution paramétrée $p_\theta(x)$ qui approxime $p_{\mathcal{D}}(x)$. Pour atteindre cet objectif, deux grandes approches se distinguent dans la littérature :

Approche explicite

les modèles issues de cette approche sont fondés sur la vraisemblance. En effet, ils cherchent à apprendre directement la distribution des données $p_\theta(x)$ en maximisant la vraisemblance des données observées. Le principe de base est de définir une fonction de vraisemblance $p(x|\theta)$ qui décrit la probabilité des données x étant donné les paramètres du modèle θ, et de trouver les paramètres θ qui maximisent cette fonction. Cela se traduit par la résolution du problème d'optimisation suivant :

$$\theta^* = \arg\max_\theta \log p(x|\theta)$$

Parmi les modèles les plus représentatifs de cette catégorie, on trouve les *Variational Autoencoders (VAE)*, les modèles de diffusion et les *Normalizing Flows (NFs)*. La section 2.3 s'attardera plus en détail sur les VAE, car ils s'inscrivent dans le cadre des travaux menés. En ce qui concerne les modèles de diffusion et les NFs, une description succincte est fournie en annexe A.4, car n'ayant pas été utilisé dans nos applications. Un approfondissement peut être trouvé dans [Yang *et al.*, 2023] et [Papamakarios *et al.*, 2021].

Modèles génératifs implicites

Contrairement aux modèles explicites, ces modèles ne cherchent pas à modéliser directement $p_\theta(x)$. Ils se concentrent plutôt sur la génération de nouvelles données indiscernables des données réelles, sans estimer explicitement la probabilité de chaque point de données. Au lieu de modéliser directement $p_\theta(x)$, ils apprennent un processus de génération G_θ à partir d'une variable aléatoire z, appelée vecteur latent. Ce vecteur est en réalité la représentation des données dans un espace de dimension inférieure à celui des données d'entrée, appelé espace latent, qui permet de structurer ces dernières de manière compacte et compressée.

Parmi les nombreux modèles qui adoptent cette approche, les Generative Adversarial Networks (GANs) se distinguent par leur importance et leur impact dans le domaine. Introduits pour la première fois par [Goodfellow *et al.*, 2014], les GANs constituent un exemple emblématique de modèles génératifs implicites. En raison de leur pertinence dans nos travaux, nous en fournirons une explication détaillée dans la section suivante.

2.2 Generative Adversarial Networks (GANs)

2.2.1 Présentation générale

Dans le papier introductif, il est question de mettre en compétition deux réseaux de neurones dans un cadre de jeu à somme nulle : le générateur et le discriminant.

- Le générateur, noté G_θ, est un réseau de neurones paramétré par un vecteur de poids θ. Son rôle est de prendre en entrée un vecteur latent z, échantillonné à partir d'une distribution simple (généralement une distribution gaussienne), et de produire une sortie $G(z)$, qui est censée ressembler à une donnée réelle. Le but du générateur est de créer des échantillons qui soient suffisamment réalistes pour être indiscernables des données réelles ;
- Le discriminant, noté D_ϕ, est un autre réseau de neurones, paramétré par un ensemble de poids ϕ. Il a pour objectif de distinguer entre les échantillons réels et ceux générés par G_θ. Le discriminant prend en entrée une donnée x et retourne un score $D(x)$, représentant la vraisemblance que x soit une donnée réelle. Idéalement, $D(x)$ doit être proche de 1 pour une donnée réelle et proche de 0 pour une donnée générée.

L'objectif global de l'apprentissage est ainsi de produire des échantillons réalistes en faisant en sorte que ces deux réseaux s'améliorent mutuellement à travers une compétition continue. Ci-dessous une illustration de leur fonctionnement générique :

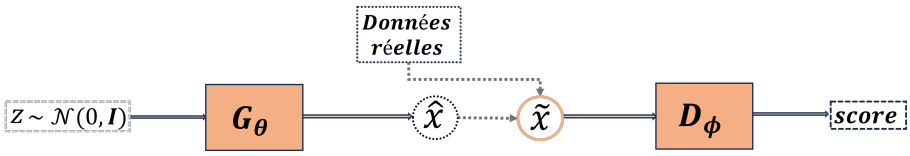

FIGURE 2.3 – Illustration d'une architecture générique de GAN.

Les paramètres θ et ϕ des réseaux de neurones G et D ne seront parfois pas explicitement mentionnés afin d'alléger les notations.

Cette compétition est formulée comme un jeu min-max où le générateur G essaie de minimiser une fonction de coût V, tandis que le discriminant D essaie de la maximiser :

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\mathcal{D}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{2.1}$$

En reformulant l'objectif du discriminant pour distinguer les données réelles des données générées, la fonction de perte peut être exprimée à l'aide de la fonction $\mathrm{Err}(u, v)$, qui mesure l'erreur entre u et v, de la manière suivante :

$$L_D = \mathrm{Err}(D(x), 1) + \mathrm{Err}(D(G(z)), 0) \tag{2.2}$$

Si l'étiquette "1" représente les données réelles et "0" les données générées, alors l'équation (2.2) quantifie la précision avec laquelle la classification des vraies données est proche de 1 et celle des données générées proche de 0.

Dans la même logique, comme l'objectif du générateur est de tromper le discriminant, on peut décrire sa fonction de perte par l'équation suivante :

$$L_G = \text{Err}(D(G(z)), 1) \tag{2.3}$$

car l'objectif est d'attribuer l'étiquette "vraie" aux données générées et par conséquent que $D(G(x))$ soit le plus proche possible de 1.

Les équations (2.2) et (2.3) peuvent être assimilées à des problèmes de classification binaire. Pour ces types de problèmes, une fonction de perte couramment utilisée dans la littérature est l'entropie croisée binaire ([Cox, 1958] ; [Mao *et al.*, 2023]), qui mesure la différence entre deux distributions de probabilité. Dans le cadre de la classification binaire, si y est l'étiquette réelle et $\hat{y}$ est la probabilité prédite que l'étiquette soit réelle, l'entropie croisée binaire est définie par :

$$Entropie(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \tag{2.4}$$

En combinant les équations (2.2),(2.3) et (2.4), on retrouve la fonction objectif du GAN à optimiser donnée par l'équation (2.1) qui décrit une compétition entre générateur et discriminant.

L'objectif ultime de cette compétition est d'atteindre un équilibre de Nash où le générateur produit des échantillons indiscernables des données réelles, rendant le discriminant incapable de faire mieux que de deviner de manière aléatoire.

La garantie théorique de cet équilibre est donnée par les théorèmes (2.2.1) et (2.2.2) :

Théorème 2.2.1 *(Discriminant optimal, voir [Goodfellow et al., 2014], Proposition 1).*

Pour tout générateur G fixé, le discriminant optimal est donné par :

$$D^*_G(x) = \frac{p_{\mathcal{D}}(x)}{p_{\mathcal{D}}(x) + p_G(x)} \tag{2.5}$$

avec $p_G(x)$ la distribution des données générées.

En substituant l'optimal pour D, l'objectif revient à :

$$V(D^*_G, G) = \mathbb{E}_{x \sim p_{\mathcal{D}}}[\log \frac{p_{\mathcal{D}}}{p_{\mathcal{D}} + p_G}] + \mathbb{E}_{z \sim p_G}[\log \frac{p_G}{p_{\mathcal{D}} + p_G}] \tag{2.6}$$

Avant de résoudre 2.6, nous devons introduire les définitions de divergence de Kullback-Leibler et de Jensen-Shannon.

Definition 2.2.1 *(Divergence de Kullback-Leibler, voir [Kullback et al., 1951]).*

Soient $\mathbb{P}$ et $\mathbb{Q}$ deux mesures de probabilité avec $\mathbb{P}$ absolument continue par rapport à $\mathbb{Q}$. La divergence de Kullback-Leibler (KL) entre $\mathbb{P}$ et $\mathbb{Q}$ est définie par :

$$d_{KL}(\mathbb{P} \parallel \mathbb{Q}) = \mathbb{E}_{\mathbb{P}}\left[\log \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}\right].$$

Definition 2.2.2 *(Divergence de Jensen-Shannon ($\mathcal{JS}$), voir [Arjovsky et al., 2017]).*

La divergence de Jensen-Shannon ($\mathcal{JS}$) entre deux mesures $\mathbb{P}$ et $\mathbb{Q}$ est définie par

$$d_{\mathcal{JS}}(\mathbb{P} \parallel \mathbb{Q}) = \frac{1}{2} d_{KL}\left(\mathbb{P} \parallel \frac{\mathbb{P}+\mathbb{Q}}{2}\right) + \frac{1}{2} d_{KL}\left(\mathbb{Q} \parallel \frac{\mathbb{P}+\mathbb{Q}}{2}\right)$$

En d'autres termes, ces deux métriques permettent de mesurer à quel point deux distributions sont similaires. Leur intérêt se trouve dans la simplification de l'objectif d'apprentissage du modèle.

Théorème 2.2.2 *(Générateur optimal, voir [Goodfellow et al., 2014], Théorème 1)*

En reprenant l'équation (2.6), l'optimum global G^* est atteint pour $p_{\mathcal{D}} = p_G$. Et, dans ce cas, on a : $V(D^*_G, G^*) = -log(4)$ et $D^*_{G^*} = \frac{1}{2}$.

Les preuves des théorèmes (2.2.1) et (2.2.2) sont en annexe A.1.

2.2.2 Optimisation et entrainement

L'optimisation des GANs repose sur la mise à jour alternée du générateur et du discriminant afin d'atteindre l'objectif : générer des données indiscernables des vraies données. En pratique, malgré ces garanties théoriques, la version originale des GANs souffre de certains problèmes dont les plus notables sont les suivants :

Non-convergence

Les GANs sont souvent difficiles à entraîner et peuvent ne pas converger, c'est-à-dire que le générateur et le discriminant peuvent ne jamais atteindre un équilibre stable. En effet, le jeu minimax dans la construction des GANs s'apparente au fait de trouver un équilibre de Nash, entre le générateur et le discriminant. Cet équilibre revient à trouver

un point selle de la fonction objective V. La convergence locale de l'algorithme initiale est le plus souvent garantie lorsque $V(\theta, \phi)$ est localement convexe en θ et concave en ϕ. Dans le cadre des GANs, c'est rarement le cas, ce qui peut entraîner des oscillations pendant l'entraînement et l'équilibre peut ne pas être atteint. On pourra explorer le papier [Lars *et al.*, 2018], qui discute des aspects purement théoriques des conditions sous lesquelles de nombreux GANs convergent.

Mode collapse

Dans l'article [Goodfellow *et al.*, 2014], les auteurs désignent ce phénomène sous le nom de *Helvetica scenario*. Il s'agit d'une situation où le générateur peine à produire une diversité d'échantillons, se limitant à générer des données fortement similaires.

Vanishing gradient

C'est lorsque les gradients du générateur deviennent tellement faibles qu'ils peuvent disparaître en raison d'un discriminant qui devient un peu trop performant, rendant l'entraînement du générateur inefficace. En effet, lorsque le discriminant est optimal, l'optimisation du générateur rencontre des difficultés, car les gradients de la divergence $\mathcal{JS}$ ont tendance à diminuer rapidement lorsque les distributions $p_{\mathcal{D}}$ et $p_{\mathcal{G}}$ sont soit très similaires, soit très éloignées. En pratique, cela conduit à une convergence oscillant entre lenteur et rapidité.

Pour faire face à ces difficultés d'entraînement des GANs, plusieurs approches ont été explorées. Parmi celles-ci, des algorithmes basés sur la régularisation ont été développés pour atténuer les problèmes de divergence, prévenir le surapprentissage et améliorer la généralisation des modèles. Par ailleurs, de nouvelles architectures ont également été proposées. Nous présenterons ici celles qui ont joué un rôle dans le cadre de nos travaux.

2.2.3 Wasserstein GANs

Les Wasserstein GANs (WGANs) [Arjovsky *et al.*, 2017] ont été introduits pour améliorer la stabilité de l'entraînement des GANs et résoudre les problèmes tels que le mode *collapse* et le *vanishing gradient*. Contrairement aux GANs classiques qui utilisent la divergence de Jensen-Shannon ($\mathcal{JS}$) comme mesure de dissimilarité entre les distributions de données réelles et générées, les WGANs utilisent la distance de Wasserstein, aussi connue sous le nom de distance *Earth-Mover(EM)*.

Definition 2.2.3 *(Distance de Wasserstein)*

La distance de Wasserstein ou de Earth-Mover(EM) entre deux distributions de probabilité $P_{\mathcal{D}}$ et P_G est définie comme suit :

$$W(P_{\mathcal{D}}, P_G) = \inf_{\gamma \in \Pi(P_{\mathcal{D}}, P_G)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|]$$

où $\Pi(P_{\mathcal{D}}, P_G)$ *est l'ensemble des distributions jointes* $\gamma(x, y)$ *avec distributions marginales* $P_{\mathcal{D}}$ *et* P_G*.*

La définition ne se prête pas à une application directe aux GANs, car elle nécessite une minimisation sur l'ensemble des mesures de probabilité conjointes, ce qui est difficile à paramétrer avec des réseaux de neurones. Cependant, elle peut être approximée par le dual de la distance de Kantorovich-Rubinstein, grâce au théorème de Kantorovich-Rubinstein.

Théorème 2.2.3 *(Dualité de Kantorovich–Rubinstein, voir [Berger et al., 2009])*

$$W(P_{\mathcal{D}}, P_G) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim P_{\mathcal{D}}}[f(x)] - \mathbb{E}_{x\sim P_G}[f(x)]$$

où $\|f\|_L := \sup_{x,y\in\mathbb{R}^n} \frac{|f(x)-f(y)|}{\|x-y\|}$ désigne la constante de Lipschitz d'une fonction f.

Dans les WGANs, le discriminant D, joue un rôle fondamentalement différent de celui d'un discriminant classique dans un GAN standard. Au lieu de simplement distinguer entre les échantillons réels et générés, le discriminant est utilisé pour approximer la fonction f qui permet de calculer la distance de Wasserstein. L'objectif se formalise de la manière suivante :

$$\min_G \max_{\|D\|_L \leq 1} W(G, D) = \mathbb{E}_{x\sim P_{\mathcal{D}}}[D(x)] - \mathbb{E}_{z\sim P_z}[D(G(z))]$$

En pratique, plusieurs techniques permettent d'assurer la contrainte lipchitz :

- ***Clipping*** :

  Le *clipping* des poids est une méthode simple et intuitive pour s'assurer que la partie discriminante respecte la contrainte 1-Lipschitz. Cette technique, proposée par [Arjovsky *et al.*, 2017], consiste à contraindre les poids du discriminant à rester dans un intervalle fixe $[-c, c]$, où $c > 0$ et généralement petit. En pratique, pour chaque pas de temps i on applique la transformation :

$$\theta_{D,i} \leftarrow \begin{cases} -c & \text{si} \quad \theta_{D,i} < -c \\ \theta_{D,i} & \text{si} \quad -c \leq \theta_{D,i} \leq c \\ c & \text{si} \quad \theta_{D,i} > c \end{cases}$$

  Bien que cette méthode soit simple en implémentation et pratique, en restreignant trop sévèrement les poids, elle peut réduire la capacité d'apprentissage du discriminant, conduisant parfois à des performances sous-optimales. Ainsi, une autre

approche a été adoptée.

- **Pénalisation des gradients :**
  La pénalisation des gradients est une méthode plus sophistiquée pour assurer la contrainte 1-Lipschitz. Proposée par [Gulrajani *et al.*, 2017], cette technique ajoute un terme de régularisation à la fonction de coût du discriminant, favorisant ainsi les gradients de D à être proches de 1.

$$\mathcal{L}_D = \mathbb{E}_{x\sim P_{\mathcal{D}}}[D(x)] - \mathbb{E}_{z\sim p_z}[D(G(z))] + \lambda \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{2.7}$$

  avec :

$$\begin{cases} \lambda : \text{coefficient de régularisation qui contrôle le poids du terme de pénalisation ;} \\ \hat{x} : \text{un point interpolé entre un échantillon réel } x \text{ et un échantillon généré } G(z), \\ \quad \text{calculé comme suit : } \hat{x} = \epsilon x + (1-\epsilon)G(z) \quad \text{où} \quad \epsilon \sim U[0,1]. \end{cases}$$

  Cette pénalisation des gradients est une meilleure approximation de la contrainte 1-Lipschitz, mais nécessite une calibration de l'hyperparamètre λ.

La perte du générateur reste la même :

$$L_G = -\mathbb{E}_{z\sim p_z(z)}[D(G(z))] \tag{2.8}$$

De nombreuses variantes des GANs ont été développées. [Chakraborty *et al.*, 2023] donne une revue détaillée des principales architectures dérivées du GAN original.

2.2.4 Conditional GAN

Après avoir discuté des WGANs et des techniques permettant d'assurer la contrainte 1-Lipschitz, il est pertinent d'explorer une autre dimension des GANs qui vise à ajouter des contraintes supplémentaires pour améliorer la génération de données : les modèles conditionnels. Ces modèles, connus sous le nom de Conditional GANs (cGANs) [Mehdi *et al.*, 2014], permettent de contrôler et de guider en quelque sorte le processus de génération de données en conditionnant le générateur et le discriminant sur des informations additionnels. Les fonctions de perte pour le générateur et le discriminant sont modifiées pour inclure les variables de conditionnement qu'on notera y. Ainsi, la fonction de perte devient :

Pour le discriminant :

$$\mathcal{L}_D = -\mathbb{E}_{(x,y)\sim P_{\mathcal{D}}}[\log D(x|y)] - \mathbb{E}_{(z,y)\sim p_z p_y}[\log(1 - D(G(z|y)|y))]$$

Pour le générateur : $\mathcal{L}_G = -\mathbb{E}_{(z,y)\sim p_z p_y}[\log D(G(z|y))]$

Pour la plupart des architectures dérivées des GANs, il est possible d'introduire un conditionnement, car cela revient théoriquement à apprendre la distribution conditionnelle $(x|y)$. Ainsi, pour un WGAN conditionnel, l'une des architectures choisies dans la suite pour nos études, on a :

$$\begin{aligned}\mathcal{L}_D = \mathbb{E}_{(x,y)\sim P_{\mathcal{D}}}[D(x|y)] - \mathbb{E}_{(z,y)\sim p_z p_y}[D(G(z|y)|y)] \\ + \lambda \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}\left[(\|\nabla_{\hat{x}} D(\hat{x}|y)\|_2 - 1)^2\right]\end{aligned} \tag{2.9}$$

$$\mathcal{L}_G = -\mathbb{E}_{(z,y)\sim p_z p_y}[D(G(z|y)|y)] \tag{2.10}$$

Des détails complémentaires sont donnés dans la section 2.4.

La théorie des GANs ayant été présenté, nous nous intéressons à notre second modèle, l'autoencodeur variationnel, introduit par [Kingma *et al.*, 2014b].

2.3 Auto-encodeurs variationnels (VAE)

Avant d'introduire la théorie relative aux auto-encodeurs variationnels, il est pertinent d'exposer la notion d'auto-encodeur.

2.3.1 Généralités sur les Auto-encodeurs (AE)

Un auto-encodeur est un type de réseau de neurones non supervisé utilisé pour apprendre des représentations efficaces des données, généralement dans le but de réduire la dimensionnalité. Il se compose de deux parties principales :

- **Encodeur :** transforme les entrées x en une représentation latente z à travers une fonction d'encodage f_θ paramétrée par des poids θ tel que : $z = f_\theta(x)$.
- **Décodeur :** transforme z en une reconstruction des entrées $\hat{x}$ à travers une fonction de décodage g_ϕ paramétrée par des poids ϕ tel que $\hat{x} = g_\phi(z)$.

L'objectif d'un auto-encodeur est de minimiser la différence entre l'entrée x et la sortie reconstruite $\hat{x}$. La figure 2.4 est une illustration de son fonctionnement.

Plus formellement, pour toute donnée $x \in \mathbb{R}^n$, si on note $\mathcal{R}$ le réseau auto-encodeur, $\mathcal{D} : \mathbb{R}^n \to \mathbb{R}^d$ le décodeur et $\mathcal{E} : \mathbb{R}^d \to \mathbb{R}^n$ l'encodeur, on a : $\mathcal{R} = \mathcal{D} \circ \mathcal{E}$ tel que $\|\mathcal{R}(x) - x\| \leq \rho$, avec ρ la précision de reconstruction.

On cherche l'ensemble des paramètres ζ de $\mathcal{R}$ tel que :

$$\zeta^* = \arg\min_{\zeta} \mathcal{L}(x, \hat{x}) = \arg\min_{\zeta} \|\mathcal{R}(x) - x\|,$$

L'optimisation de cette structure neuronale suit le même principe que l'apprentissage classique des réseaux de neurones. Il est essentiel de souligner que les deux réseaux sont entraînés simultanément, avec une rétropropagation du gradient du décodeur vers l'encodeur. Outre l'aspect réduction de dimensionnalité, les AE sont utilisés par exemple

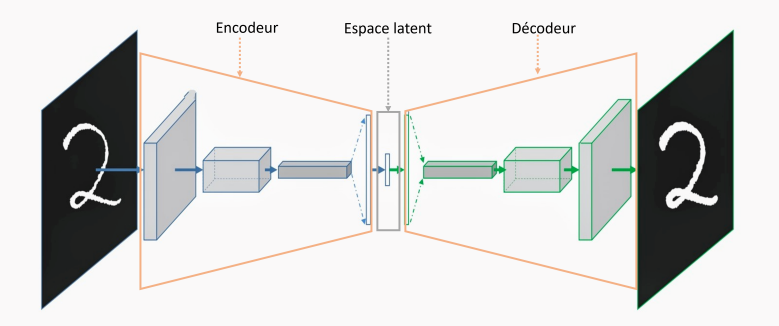

FIGURE 2.4 – Exemple de reconstruction d'image avec un auto-encodeur.

dans des tâches de débruitage (voir figure 2.5) et de restauration ou même d'extraction de caractéristiques pour alimenter des modèles de *Machine Learning*.

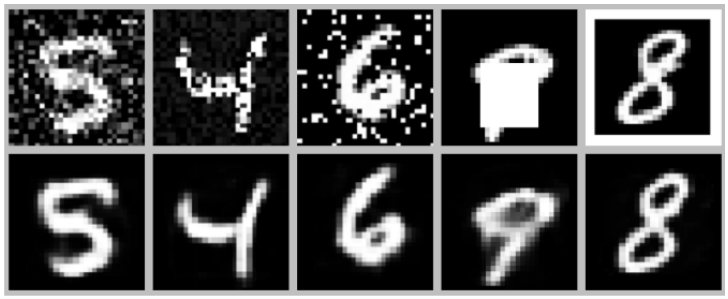

FIGURE 2.5 – Exemple de débruitage avec AE sur des données issues de la base MNIST. En première ligne, les images bruitées ; en second, celles débruitées par l'AE. Image extraite de [Agostinelli *et al.*, 2013].

Les auto-encodeurs de base peuvent être étendus de diverses manières pour améliorer leurs performances ou pour répondre à des besoins spécifiques comme les Auto-encodeurs convolutifs qui utilisent des couches de convolution pour capturer les caractéristiques spatiales des données, particulièrement utiles pour les images ou encore le très célèbre auto-encodeur variationnel. Pour une revue de littérature étendue, on pourra consulter [Shuangshuang *et al.*, 2023].

2.3.2 De l'autoencodeur à l'autoencodeur variationnel

La question qu'on pourrait se poser serait de savoir si l'AE ne pourrait pas faire office de modèle génératif ?

A priori, cela est possible puisque le décodeur modélise la loi de $(x|z)$. Mais, plusieurs problèmes se posent. Un autoencodeur traditionnel apprend une représentation compacte et déterministe des données. Par exemple, si nous avons des images de chiffres manuscrits,

l'autoencodeur peut apprendre à encoder chaque image en un vecteur de caractéristiques et à décoder ce vecteur pour reconstruire l'image d'origine. Toutefois, lorsqu'il s'agit de générer de nouveaux exemples, le modèle est médiocre, car chaque entrée correspond à une seule représentation latente fixe. C'est pour pallier ce problème que les autoencodeurs variationnels ont été introduits.

Un VAE modélise une distribution probabiliste dans l'espace latent. Pour notre exemple d'image, plutôt que d'apprendre un vecteur de caractéristiques fixe pour chaque image, il apprend une distribution pour chaque image. Avec cette distribution latente, il sera possible de générer de nouvelles images en échantillonnant cette distribution.

Considérons le jeu de données $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$ correspondant à des réalisations de la variable aléatoire x distribuée selon $p_{\mathcal{D}}$. Notre objectif est donc d'apprendre $p_{\mathcal{D}}$.

Dans les VAE, on suppose que le processus génératif $p_{\mathcal{D}}$ implique une certaine variable aléatoire z et une paramétrisation θ telle que $\forall i, z_i \sim p_{\theta^*}(z)$ (distribution à priori) et $x_i \sim p_{\theta^*}(x|z)$.

θ et $z_1, \ldots, z_n$ sont inconnus pour l'instant et la seule information que nous avons, c'est que nous avons besoin de θ pour pouvoir générer x à partir de z. Autrement dit, il est question de maximiser la vraisemblance des données générées sur $\mathcal{D}$. En statistiques, il a été prouvé que maximiser $p_\theta(x)$ revient à maximiser $\log(p_\theta(x))$. Pour des questions de facilités numériques et d'algorithmie, on privilégie généralement la maximisation de la log-vraisemblance. Ainsi, le problème revient à :

$$\theta^* = \arg\max_{\theta} \log p_\theta(x) \tag{2.11}$$

Résoudre l'équation (2.11) revient à trouver $p_\theta(x)$ qui n'a pas d'expression analytique, mais qui s'écrit sous la forme $p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$. En effet, si on considère la distribution conjointe $\{x, z\}$, on a comme marginale pour x, $p_\theta(x) = \int p_\theta(x, z)dz$ avec $p_\theta(x, z) = p_\theta(x|z)p_\theta(z)$.

Comme x est connu, une bonne idée serait de voir ce qu'il est possible d'obtenir à partir de $p_\theta(z|x)$, la distribution à postériori. Toutefois, en pratique, la distribution $p_\theta(z|x)$ n'est généralement pas accessible directement. Par conséquent, nous chercherons à l'approximer à l'aide de $q_\phi(z|x)$. Ce qui est naturel, c'est que $q_\phi(z|x)$ soit très proche de $p_\theta(z|x)$. Cela correspond à trouver ϕ^* tel que :

$$\phi^* = \arg\min_{\phi} d(q_\phi(z|x), p_\theta(z|x))$$

Avec d choisi convenablement pour mesurer la dissimilarité entre les deux distributions. Dans la littérature, un bon candidat de mesure de dissimilarité entre distributions est la divergence de Kullback-Leibler (voir 2.2.1). Ainsi,

$$
\begin{aligned}
d(q_\phi(z|x), p_\theta(z|x)) &= KL(q_\phi(z|x) \parallel p_\theta(z|x)) \\
&= \int \log\left(\frac{q_\phi(z|x)}{p_\theta(z|x)}\right) p_\theta(z|x)\, dz \\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log q_\phi(z|x) - \log p_\theta(z|x)\right] \\
&= \mathbb{E}_{q_\phi(x|z)}\left[\log q_\phi(z|x) - \log p_\theta(x|z) - \log p_\theta(x) + \log p_\theta(x)\right] \\
KL(q_\phi(z|x) \parallel p_\theta(z|x)) &= -\underbrace{\left(\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - KL(q_\phi(z|x) \parallel p_\theta(z))\right)}_{(1)} + \log p_\theta(x)
\end{aligned}
$$

(1) est appelée *Evidence Lower Bound (ELBO)*. En re-écrivant l'expression, on a alors : $KL(q_\phi(z|x) \parallel p_\theta(z|x)) = -ELBO + \log p_\theta(x)$.

Comme $KL(q_\phi(z|x) \parallel p_\theta(z|x))$ est positive, alors $\log p_\theta(x) \geq ELBO$. Rappellons que notre objectif est de minimiser $KL(q_\phi(z|x) \parallel p_\theta(z|x))$. Ainsi, réaliser cet objectif revient à maximiser $ELBO$. De plus, en maximisant $ELBO$, on réalise l'objectif de maximiser $\log p_\theta(x)$. L'objectif se résume alors à :

$$
\max_{\theta,\phi} ELBO(\theta, \phi; x) = \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right]}_{(2)} - \underbrace{KL(q_\phi(z|x) \parallel p_\theta(z))}_{(3)}
$$

On choisit pour simplifier, $q_\phi(z|x) = \mathcal{N}(z; \mu; \sigma I)$, $p_\theta(z) = \mathcal{N}(z; 0; I)$ et $p_\theta(x|z) = \mathcal{N}(x; f_\theta(z), \sigma_x^2 I)$.

Dans ce contexte, on a : $(3) = \frac{1}{2}(tr(\sigma I) + \mu^T \mu - d - \log\det(\sigma I))$

et $(2) = \mathbb{E}_{q_\phi(z|x)}\left[-\frac{1}{2\sigma_x^2}\|x - f_\theta(z)\|^2\right] - \frac{d}{2}\log(2\pi\sigma_x^2)$ avec d la dimension de l'espace latent. L'objectif final de (2) revient à $\min_\theta \mathbb{E}_{q_\phi(z|x)}\left[\|x - f_\theta(z)\|^2\right]$, qui est l'erreur quadratique moyenne (MSE) entre les données d'entrée x et les données reconstruites $f_\theta(z)$.

Pour des raisons opérationnelles, le réseau $q_\phi(z|x)$ donne en sortie une moyenne μ et un écart type σ, ce qui permet d'écrire $z = \mu + \sigma\epsilon$ où $\epsilon \sim \mathcal{N}(0, I)$, on parle d'astuce de reparamétrisation. La figure 2.6 est une illustration de l'architecture finale d'un VAE.

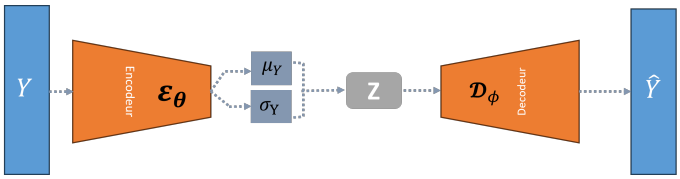

FIGURE 2.6 – Architecture d'un VAE dans sa version originelle avec l'astuce de reparamétrisation

De même que pour les GANs, les VAEs peuvent également avoir une version conditionnelle qui sera présentée en raison de leur utilité dans nos travaux.

2.3.3 Conditional VAE

Le CVAE est une extension du VAE qui permet de générer des données en fonction de conditions spécifiques, notées y. L'objectif est de maximiser la log-vraisemblance conditionnelle $\log p_\theta(x|y)$, plutôt que la log-vraisemblance marginale $\log p_\theta(x)$ comme dans le VAE. ELBO pour le CVAE s'écrit de manière similaire à celui du VAE, mais chaque terme est maintenant conditionné par y :

$$\log p_\theta(x|y) \geq \mathbb{E}_{q_\phi(z|x,y)} \left[\log p_\theta(x|z,y)\right] - KL(q_\phi(z|x,y) \parallel p_\theta(z|y))$$

L'objectif final du CVAE est de maximiser cette ELBO conditionnelle :

$$\max_{\theta,\phi} \mathbb{E}_{q_\phi(z|x,y)} \left[\log p_\theta(x|z,y)\right] - KL(q_\phi(z|x,y) \parallel p_\theta(z|y))$$

La fonction de perte s'écrit :

$$\begin{aligned} \mathcal{L}(\theta,\phi;x,y) =& \frac{1}{2}\left(\mathrm{tr}(\sigma I) + \mu^T\mu - d - \log\det(\sigma I)\right) \\ &+ \frac{1}{2\sigma^2}\mathbb{E}_{q_\phi(z|x,y)}\left[\|x - f_\theta(z,y)\|^2\right] + \frac{d}{2}\log(2\pi\sigma^2) \end{aligned} \tag{2.12}$$

Cette architecture conditionnelle est pertinente pour nos travaux, car conditionner un modèle revient à informer le système sur ce que l'on attend de lui, réduisant ainsi l'espace de possibles. C'est une anticipation informée : on n'explore pas tous les futurs possibles, mais seulement ceux qui sont cohérents avec les conditions imposées. Cette anticipation est une forme de prédiction, car elle suppose que, sous certaines conditions, certains résultats sont plus probables ou logiques que d'autres. Ce qui cadre parfaitement avec notre objectif. Dans la section suivante, nous donnerons davantage de détails.

2.4 Approche proposée et cadre de modélisation

L'objectif de cette section est de présenter les architectures retenues dans le cadre de notre étude et de décrire dans quelle mesure elles permettent de répondre à notre problématique.

2.4.1 Motivations

Pour rappel, notre objectif est de modéliser et prédire la FDB à partir de variables financières, et donc de déterminer le *Best Estimate*. Si X désigne l'ensemble des variables économiques, notre problème se formule comme un apprentissage supervisé consistant à trouver une fonction f telle que $FDB = f(X)$. Dans notre étude, f est un réseau de neurones génératif paramétré f_θ.

Nous considérons un ensemble de n observations et d variables explicatives : $X \in \mathbb{R}^{n\times d}$ et $FDB \in \mathbb{R}^{n\times 1}$. Un critère essentiel pour le choix de f_θ est sa capacité à satisfaire la

relation suivante :

$$f_\theta : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times 1}$$

Parmi les différents modèles répondant à cette contrainte, notre choix s'est porté sur les GAN et les VAE. Bien que leur objectif initial ne soit pas la prédiction directe, ces modèles peuvent être adaptés pour inclure un contexte de génération, ce qui les rapproche d'une tâche de prédiction. De plus, ils représentent efficacement les approches implicite et basées sur la vraisemblance.

2.4.2 Méthodologie

Dans cette section, nous décrivons de manière détaillée les architectures utilisées ainsi que leurs spécificités.

Modèle génératif implicite retenu

Comme évoqué, nous utiliserons un GAN modifié par rapport à sa version originale. L'architecture retenue est un GAN de Wasserstein conditionnel avec pénalité des gradients que nous dénotons par Cond-WGAN-GP. Considérons toujours X et la cible Y. Ci-dessous un schéma illustrant le fonctionnement du GAN dans sa version basique.

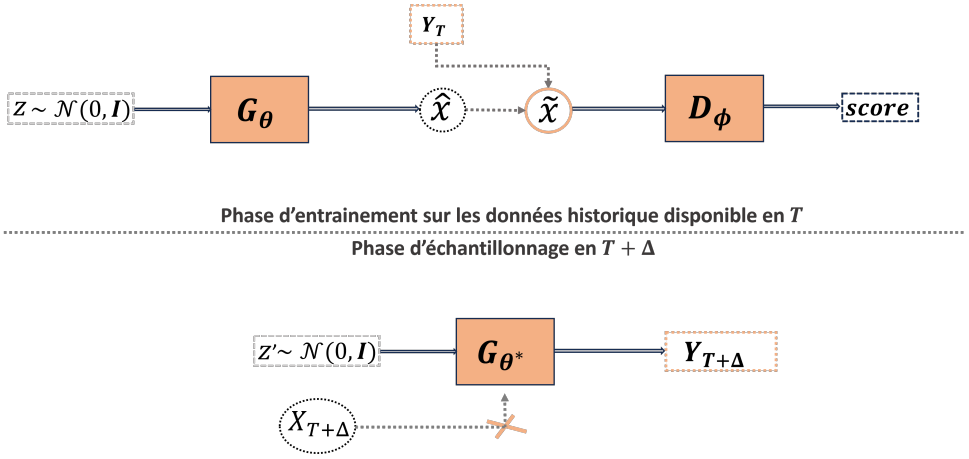

Figure 2.7 – Architecture d'un GAN vanille.

Comme on peut le voir sur l'illustration, le modèle permet de modéliser directement Y tandis que dans notre contexte Y dépend de certaines informations exogènes comme les conditions économiques. Ainsi, le modèle n'est pas directement utilisable à des fins de prédictions. C'est pour cette raison que nous avons adopté l'architecture conditionnelle.

L'idée, c'est de rendre le modèle capable de produire des échantillons de Y contextuellement. Ce contexte correspond aux conditions X. Ainsi, en modifiant X le modèle, devrait produire le Y qui lui correspond. L'objectif du modèle génératif est le même, générer des nouveaux échantillons, mais cette fois-ci, il le fait de manière guidée, selon

certaines contraintes. Ce procédé s'apparente à faire de la prédiction. Schématiquement, il est présenté par la figure 2.8.

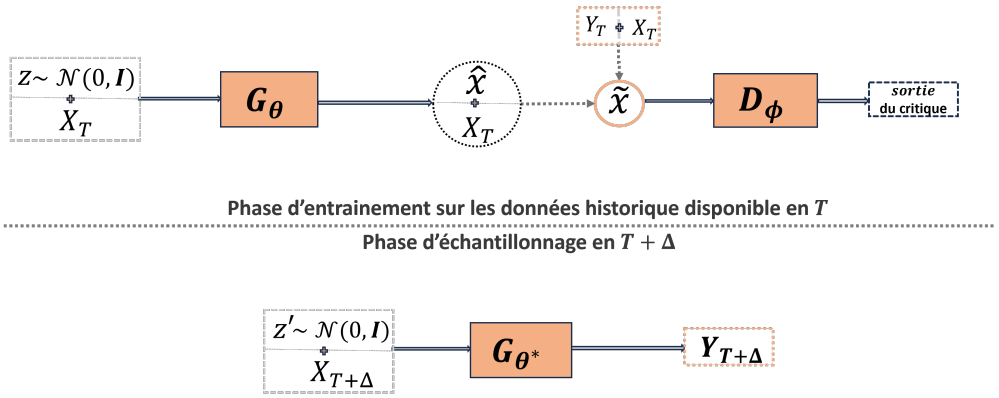

FIGURE 2.8 – Architecture WGAN avec conditionnement

Une fois le modèle à mettre en œuvre défini, la question qui se pose concerne le choix des réseaux générateur et discriminant. Pour effectuer ce choix, un descriptif des principaux avantages et faiblesses des architectures potentielles est proposée dans le tableau 2.1.

Architecture	Avantages	Inconvénients
CNN	Efficace pour extraire des motifs locaux. Adapté pour extraire des *features* complexes.	Moins adapté aux données séquentielles pures. Ne capture pas bien les dépendances à long terme.
RNN	Capture les dépendances temporelles. Adapté aux données séquentielles.	Difficile à entraîner sur des longues séquences. Moins efficace pour les motifs locaux.
MLP	Simple et rapide à mettre en œuvre. Convient pour des données de petite taille et non structurées. Rapide à entraîner.	Ne gère pas bien les données séquentielles. Performance limitée sur des problèmes complexes.

TABLE 2.1 – Comparaison succincte de quelques architectures

L'objectif de notre étude étant de modéliser des données séquentielles, l'architecture récurrente est un bon candidat comme générateur. Et, le discriminant étant le guide dans son objectif d'apprentissage, une architecture permettant d'extraire des caractéristiques de manière efficace est adaptée comme discriminant.

Architecture du réseau générateur

Comme évoqué précédemment, les réseaux récurrents sont des types de réseaux artificiels qui favorisent le traitement de données séquentielles, car le sens de transmission de l'information ne se fait que dans un seul sens, d'une couche vers l'autre, mais aussi à l'intérieur de la couche elle-même.

Dans les RNNs, le terme "cellule" est souvent utilisé pour désigner les neurones. Les cellules sont les composantes de base qui traitent les informations à chaque pas de temps dans une séquence. Chaque cellule RNN prend en entrée l'état actuel et l'état caché de l'étape précédente pour produire un nouvel état caché et, éventuellement, une sortie. L'état caché est le vecteur qui contient des informations sur les *input* passées de la séquence et qui permet aux RNNs de "se souvenir" des informations précédentes.

Si on considère les données $\mathcal{D} = (x_t)_{t=1}^{T}$, les calculs dans une cellule RNN peuvent être décrits de la manière suivante :

L'état caché h_t à l'instant t est calculé en utilisant l'entrée actuelle x_t et l'état caché de l'instant précédent h_{t-1}. Cela se fait à l'aide d'une fonction d'activation σ qui est très souvent une tangente hyperbolique tanh :

$$h_t = \sigma(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$

où :

- W_{hh} est la matrice de poids reliant l'état caché précédent h_{t-1} à h_t et W_{xh} est la matrice de poids reliant l'entrée actuelle x_t à l'état caché actuel h_t ;
- b_h, le vecteur de biais pour l'état caché.

La sortie y_t à l'instant t est calculée en utilisant l'état caché h_t :

$$y_t = \sigma(W_{hy}h_t + b_y)$$

où W_{hy} est la matrice de poids reliant l'état caché actuel h^t à la sortie y^t, b_y est le vecteur de biais associé à la sortie. La figure 2.9 est une représentation du fonctionnement d'un RNN.

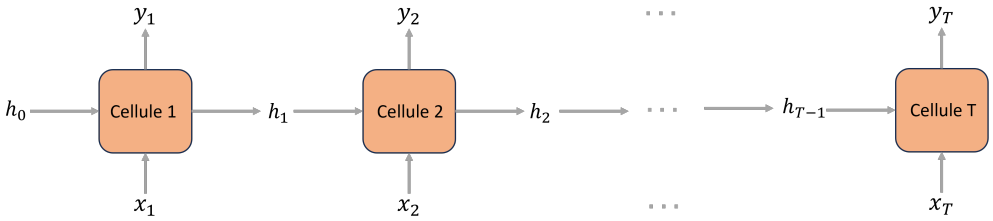

FIGURE 2.9 – fonctionnement structurel d'un RNN

Les RNNs traditionnels souffrent de *vanishing gradient* lors de l'étape de rétropropagation, ce qui rend leur apprentissage parfois inefficace. Pour atténuer ces problèmes, des variantes des RNNs ont été développées, telles que les *Long Short-Term Memory* (LSTM) [Hochreiter *et al.*, 1997] et les *Gated Recurrent Units* (GRU) [Cho *et al.*, 2014], qui sont une version simplifiée, mais tout aussi performante que les LSTM dans la gestion des dépendances à long terme.

Gated Recurrent Unit

Les GRU sont une simplification des LSTM, tout en conservant des mécanismes pour gérer les dépendances à long terme. Principalement, ils combinent les portes d'entrée et d'oubli en une seule porte de mise à jour, et possèdent une porte de réinitialisation. Les équations du GRU dont l'architecture est illustrée en figure 2.10 sont les suivantes :

- La porte de mise à jour $z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$. Cette porte contrôle dans quelle mesure l'état précédent h_{t-1} doit être conservé et mis à jour avec le nouvel état candidat $\tilde{h}_t$;
- La porte de réinitialisation $r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$ décide de la quantité d'information précédente à oublier ;
- L'état candidat $\tilde{h}_t = \tanh(W x_t + r_t \odot (U h_{t-1}) + b)$ représente la nouvelle information qui pourrait être ajoutée à l'état ;
- L'état final h_t est une combinaison linéaire de l'état précédent et de l'état candidat, pondérée par la porte de mise à jour.

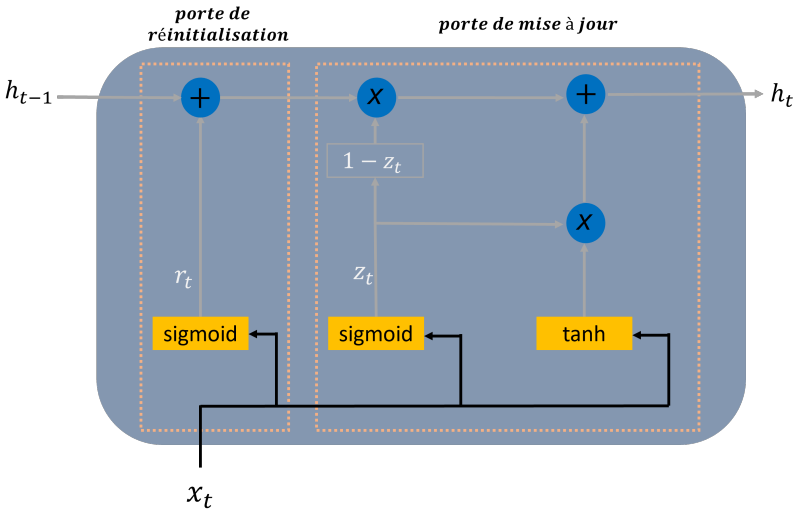

FIGURE 2.10 – Architecture GRU

Architecture du réseau discriminant

Pour rappel, dans notre modèle GAN, l'objectif du discriminant est de pouvoir guider le générateur dans sa tâche, ainsi un réseau ayant des capacités à détecter des caractéristiques cachées est convenable. Le choix d'un réseau convolutif est donc naturel. La philosophie derrière cette architecture est d'exploiter les relations locales et globales dans les données en utilisant des convolutions, une opération mathématique qui est très performante pour extraire des caractéristiques à différents niveaux de complexité.

De manière plus détaillée, Les CNNs sont composés de plusieurs couches distinctes, chacune ayant un rôle spécifique dans le traitement et l'extraction des caractéristiques :

- **Couches de convolution :**
  Ces couches appliquent des noyaux de convolution (ou filtres) sur l'entrée. Chaque filtre produit une carte de caractéristiques qui met en évidence certaines spécificités dans les données. En fonction de la nature des données, un type particulier de convolution peut être utilisé. Par exemple, la couche **Convolution 1D** est utilisée pour des données séquentielles ou temporelles : $(f*x)(i) = \sum_m f(m)\cdot x(i+m)$, avec f le filtre et x l'entrée et la **Convolution 2D** est adaptée au traitement des images. Les filtres sont appliqués sur deux dimensions : hauteur et largeur, permettant de capturer des caractéristiques spatiales : $(f*x)(i,j) = \sum_m \sum_n f(m,n)\cdot x(i+m, j+n)$. La formulation mathématique de la convolution s'étend également au cas continu.

- **Couches de *pooling***
  Ces couches réduisent la dimension des cartes de caractéristiques tout en préservant les informations les plus importantes. La couche de *pooling* la plus courante est le *max-pooling*, qui prend le maximum d'une région locale.
- **Couches d'activation**
  Après chaque convolution, une fonction d'activation non-linéaire est appliquée pour introduire de la non-linéarité dans le modèle.

- **Couches *fully connected***
  après les couches de convolution et de *pooling*, les cartes de caractéristiques sont aplaties et passées à travers une ou plusieurs couches entièrement connectées afin d'avoir la sortie finale du réseau.

Modèle génératif explicite sélectionné

De façon analogue au modèle implicite, nous utilisons un VAE modifié par un conditionnement, le CVAE. Si on considère toujours X et la cible Y, la figure 2.6 montre que le VAE permet de modéliser directement Y sans tenir compte d'informations exogènes. Ainsi, pour atteindre notre objectif d'effectuer des prédictions, il est nécessaire d'imposer un contexte de génération au modèle. Le fonctionnement schématique du CVAE est illustré par la figure 2.11.

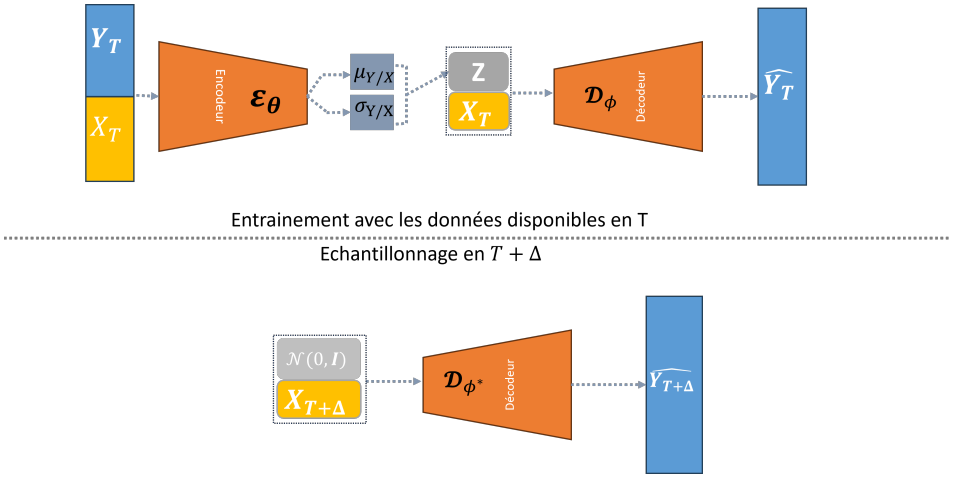

FIGURE 2.11 – Fonctionnement CVAE

Pour les mêmes raisons qui ont motivé le choix du GRU comme générateur dans le COND-WGAN-GP, nous choisissons également cette architecture comme encodeur et décodeur dans le CVAE.

Les éléments fondamentaux ainsi que l'environnement de modélisation ayant été introduits, nous proposons une application à l'estimation du *Best Estimate* sur un portefeuille épargne-retraite.

Chapitre 3

Application

Dans ce chapitre, il est question de se servir des modèles préalablement introduits afin de modéliser le *Best Estimate* d'un assureur vie. Pour des raisons de confidentialité, certaines informations ou données seront anonymisées.

3.1 Portefeuille d'étude et modèle ALM

Notre étude est effectuée sur un portefeuille épargne-retraite qui comprend plusieurs types de produits tels que le contrat "Madelin"et le contrat "Article 83".

Pour rappel, notre objectif est la modélisation de la partie stochastique du *Best Estimate*, la FDB. Cette quantité s'obtient après la prise en compte des interactions entre l'actif et le passif par le modèle ALM. Ainsi, il est essentiel de comprendre le fonctionnement dudit modèle.

Le modèle ALM permet de prendre en compte les interactions entre l'actif et le passif de l'entreprise. Pour ce faire, il intègre plusieurs *inputs* :

- Les *model point* actifs et passifs qui permettent une description de tous les actifs et du passif inhérents au portefeuille ;
- Les hypothèses qui incluent, par exemple les règles de gestion propre à l'entreprise dans le modèle, l'objectif d'allocations d'actif, le niveau des frais et bien d'autres ;
- Les scénarios économiques issus d'un outil de génération de scénarios économiques permettant de modéliser l'évolution des principaux facteurs de risques tels que les taux, les indices actions et immobilier auxquels l'entreprise est exposée.

3.1.1 Modélisation des actifs

Les actifs sont modélisés directement dans le modèle ALM en les projetant jusqu'à un horizon T de fin de projection. En effet, cette projection permet d'obtenir la valeur probable des produits financiers futurs qui permettra de déterminer les montants de participations aux bénéfices. Cette projection des actifs se fait selon les différentes classes

et leurs caractéristiques intrinsèques. Généralement, les informations contenues dans le *model points* concernent par exemple : le type de l'actif, son libellé, sa notation, son prix d'acquisition, sa valeur de marché, son code CIC, son *spread*, etc.

Les principales classes d'actifs sont les actions, les obligations (à taux fixes, taux variables, indexées sur l'inflation), les OPCVMs, l'immobilier et le *cash* ou la tresorerie.

Dans leur gestion, les compagnies d'assurance attribuent, en fonction de leur politique, certains poids à ces différents actifs. Par exemple, dans notre cas, plus de la moitié du portefeuille est constitué d'obligations, un quart d'OPCVM et le reste des autres actifs avec une minorité d'actions. Cette allocation d'actifs étant une cible, cela implique que le modèle doit effectuer des opérations d'achats et de ventes afin de converger et surtout la maintenir malgré les conditions changeantes du marché.

De plus, conformément aux contraintes du régulateur, les projections sont effectuées dans un univers risque neutre. Ainsi, la courbe des taux qui sera utilisée dans ces projections est la courbe des taux de l'*European Insurance and Occupational Pensions Authority* (EIOPA).

En termes de projection, l'une des variables essentielles modélisées est la valeur de marché (VM) des actifs. En fonction de l'actif, le calcul n'est pas le même.

Obligations

Pour une obligation, à l'instant t, $VM(t) = \sum_{j=1}^{T-t} \frac{\text{Coupon}_j}{(1+r_j)^j} + \frac{\text{Nominal}}{(1+r_T)^T}$, où r_j est le taux de marché en j. Toutefois, selon qu'il s'agit d'une obligation à taux variable (OTV) ou indexée sur l'inflation (OATi), le mode de calcul diffère. Pour les OTV, les coupons dépendent du taux de marché alors que pour les OATi, ils dépendent de l'inflation.

Actions

La valorisation des actions pour un instant t est obtenue en capitalisant la valeur de marché de l'instant $t-1$, après avoir intégré les effets d'interaction entre actifs et passifs, en utilisant le rendement de l'indice boursier. Cela se formalise par :

$$VM(t) = VM(t-1) \times (1 + R_{t,\text{ind}})$$

où $R_{t,\text{ind}}$ représente le taux de rendement de l'indice boursier pour l'année t.

Immobilier

La méthode de valorisation des actifs immobiliers suit un processus similaire à celui des actions, avec la particularité que la capitalisation se base sur l'indice immobilier.

OPCVM

Étant donné qu'un OPCVM est constitué de plusieurs classes d'actifs et/ou produits financiers, l'évaluation de sa valeur de marché doit se faire en analysant la valeur des différentes classes d'actifs qui le composent.

3.1.2 Modélisation du passif

Le principal *input* de ce modèle est le *model point*, qui regroupe toutes les informations concernant les assurés. Dans un premier temps, un modèle de projection est utilisé sans tenir compte des interactions entre l'actif et le passif, afin de calculer le *Best Estimate Garanti* (BEG). Ce dernier est déterministe car il est basé sur un scénario central.

Les différents flux de passifs sont les suivants :

Périmètre	**Prestations**	**Frais**	**Autres**
Epargne	Décès, Rachats, Autres	Frais sur encours, Frais sur prestations	Provisions mathématiques, Intérêts techniques
Rente	Sortie de capital, Rentes versées, Rachats	Frais sur encours, Frais sur prestations	Provisions mathématiques, Intérêts techniques

TABLE 3.1 – Flux du passifs

Dans le calcul du BEG, seuls les TMG et les taux techniques sont pris en compte. Ce BEG se calcule comme une différence entre les flux sortants (prestations, frais, taxes éventuelles) et les flux entrants (primes principalement).

Comme le montre le tableau 3.1, plusieurs éléments sont modélisés dans notre modèle, notamment les rachats qui peuvent être structurels ou dynamiques. Toutefois, une des variables importante dans notre modélisation étant la provision mathématique (PM), nous explicitons brièvement comment elle varie. Dans le modèle, celle-ci évolue chaque année essentiellement en fonction des prestations, des intérêts techniques, de la participation aux bénéfices, de la revalorisation et peut être diminué en raison des rachats et des sorties de rentes. Étant donné que ces variables dépendent des conditions de marché, il est essentiel de prendre en compte cet environnement économique, comme l'exige la réglementation Solvabilité *ii*. Dans le modèle ALM, cette prise en compte s'effectue principalement à travers les tables de scénarios économiques produites par les générateurs de scénarios économiques, que nous expliquerons dans la section suivante.

3.1.3 Générateur de scénarios économiques

Le Générateur de Scénarios Économiques (GSE) est un outil de simulation stochastique qui permet de modéliser, pour chaque simulation, l'évolution des principaux facteurs de risques financiers auxquels l'assureur est exposé au fil du temps comme le niveau des taux, les indices boursiers, l'inflation.

Le processus de mise en œuvre d'un GSE risque neutre se résume dans les étapes suivantes :

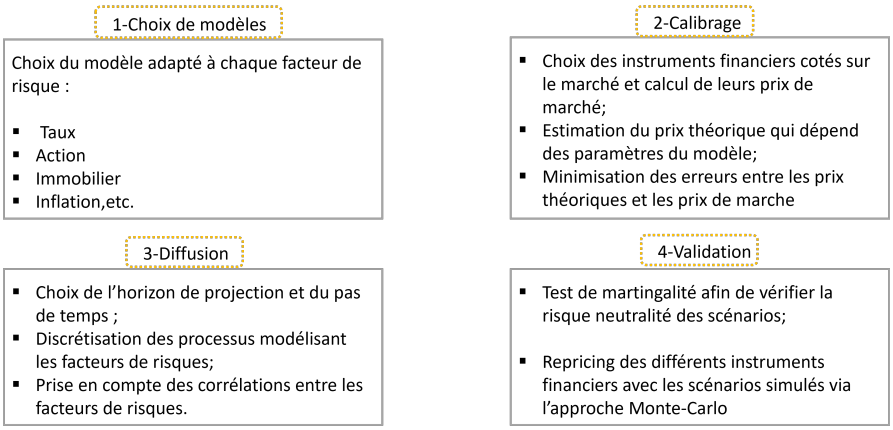

FIGURE 3.1 – Processus de mise en œuvre d'un GSE risque neutre.

Pour notre étude, nous présenterons deux des principaux modèles de diffusion.

Modèle de taux

Les taux sont simulés par le *Displaced Diffusion Libor Market Model* (DDLMM) dont la dynamique (avec $p \in \mathbb{N}^*$ facteurs) est régi par l'équation suivante :

$$dF_k(t) = (F_k(t) + \delta) \sum_{q=1}^{p} \xi_k^q(t) \times dZ_{k+1}^q(t)$$

avec :

- $F_k(t)$: le taux *forward sur* $[T_k, T_{k+1}]$;
- δ : le *shift* ou paramètre de déplacement utilisé dans la génération des taux d'intérêt négatifs ;
- ξ_k^q : la volatilité associée au $q^{\text{ième}}$ facteur pour le taux *forward* k ;
- Z_{k+1}^q : un mouvement Brownien sous la mesure forward neutre pour l'échéance T_{k+1}.

Les corrélations entre les taux à différentes échéances sont représentées par la volatilité ξ_k^q, exprimée de manière déterministe par : $\xi_k^q(t) = f(t)g(T_k - t)b^q(T_k - t)$.

Cette volatilité dépend de trois fonctions temporelles. La première est la fonction d'échelle $f(t)$: $f(t) = \theta + (1-\theta)e^{-kt}$. Ensuite, la fonction de forme $g(T_k - t)$, qui dépend du temps restant jusqu'à l'échéance, est donnée par : $g(T_k - t) = (a + b(T_k - t))e^{-c(T_k - t)} + d$. Enfin, la fonction de structure de corrélation entre taux *forward*, notée $b^q(T_k - t)$, dépend également du temps restant jusqu'à l'échéance.

Modèle action

La modélisation des actions se fait à travers une version modifiée du modèle de Black & Scholes, connue sous le nom de *Black & Scholes with Deterministic Volatility* (BSDV) (voir [Mehalla *et al.*, 2024]). Contrairement au modèle standard, le BSDV intègre un cadre de taux stochastiques avec une volatilité déterministe et variant en fonction du temps.

Considérons qu'il n'y a pas de versements de dividendes et désignons par S_t l'indice boursier modélisé. Pour $t \in [0, T]$, la dynamique de cet indice est donnée par :

$$\frac{dS_t}{S_t} = r_t dt + \sigma_t d\mathcal{W}_t^{\mathbb{P}}$$

avec :

- r_t le taux d'intérêt à court terme, issu de notre modèle de taux ;
- σ_t la volatilité déterministe à l'instant t ;
- $\mathcal{W}^{\mathbb{P}}$ le mouvement Brownien sous la mesure de probabilité historique $\mathbb{P}$.

3.1.4 *Flexing* et algorithme de participation aux bénéfices

Une fois ces quelques éléments propres à la modélisation actif-passif introduits, nous explicitons le fonctionnement général du modèle.

Comme évoqué, le modèle passif est utilisé dans un premier temps afin de calculer les flux propres au passif sans interaction avec l'actif. Une fois ces flux de BEG obtenus, il faut les intégrer au modèle ALM afin de tenir compte des différentes interactions. Cela se fait par le biais d'un processus dit de *flexing*.

Concrètement, cette technique consiste à ajuster les différents flux déterministes après la projection de l'actif en utilisant un "coefficient de *flexing*". En effet, les différents flux déterministes étant projetés au Taux Minimum Garanti, il y a lieu d'effectuer des revalorisations qui dépendent de l'évolution des actifs.

Pour chaque pas de temps t, le coefficient de *flexing* correspond à :

$$Coef_{flexing}(t) = \frac{PM_{ALM}(t)}{PM_{\text{déterministe}}(t)} \tag{3.1}$$

avec :

$$\begin{aligned}
&PM_{ALM}(t) = PM_{ALM}(t-1) + IT_{ALM}(t) + PB_{servie}(t) - Chargements_{ALM}(t)\\
&IT_{ALM}(t) = IT_{ALM}(t-1) \times Coef_{flexing}(t-1)\\
&Chargements_{ALM}(t) = Chargements_{ALM}(t-1) \times Coef_{flexing}(t-1).
\end{aligned}$$

Cette technique de *flexing* permet au modèle d'être plus efficient puisqu'il se base sur les flux déterministes préexistants et applique un coefficient de proportionnalité en fonction des conditions de marché.

La FDB étant dépendante des bénéfices futurs, il est essentiel de comprendre la politique de participation aux bénéfices de l'entreprise. Cette politique intègre aussi bien des contraintes réglementaires que des enjeux purement concurrentiels. Ainsi, la politique de gestion de la Provision pour Participation aux Excédents (PPE) peut varier d'une entreprise à une autre.

Dans le modèle ALM utilisé dans le cadre de ce mémoire, la règle de gestion peut se résumer ainsi : la participation aux bénéfices pour l'année en cours est distribuée, et toute différence par rapport à l'objectif que l'entreprise c'était fixé est ajustée via la PPE : en cas de surplus, l'excédent est ajouté à la PPE, tandis qu'en cas de déficit, la PPE compense le manque à gagner. En pratique, cette politique est résumée comme suit :

- **Etape 1 : calcul des produits financiers** : l'algorithme commence en déterminant le montant de produits financiers qui est disponible pour la PB. C'est sur ce montant que tous les autres calculs se feront.

- **Etape 2 : prise en compte des contraintes de redistribution réglementaires** : à ce stade, il est question de vérifier si le montant des produits financiers permet de faire face aux contraintes réglementaires. Sinon, trois options sont possibles : d'abord, on peut envisager la réalisation de plus-values latentes sur le portefeuille d'actifs afin de faire face aux intérêts technique. Ensuite, une reprise peut être effectuée au niveau de la PPE, avec certaines contraintes, comme la limitation de la reprise à la moitié du montant disponible par an. Enfin, si ces solutions s'avèrent inefficaces, le dernier recours consiste à renoncer à la marge sur le résultat.
- **Etape 3 : intégration du taux cible :** Ce taux est celui attendu par l'assuré. Il est fixé par l'entreprise afin de rester compétitif et ne pas inciter l'assuré à mettre fin au contrat. Il est donc généralement plus élevé que le TMG et le taux minimum réglementaire sans être non plus excessivement élevé puisque cela pourrait engendrer des pertes. À l'instant t, $Taux_{cible}(t) = 0,8\times$ Taux Moyen d'emprunt d'Etat.

- **Etape 4 : calcul du taux réellement servi :** il s'agit de la détermination du vrai montant que les assurés vont recevoir. Il correspond au Minimum entre la cible et le montant obtenu malgré l'utilisation des différents leviers lorsque les produits financiers n'arrivent pas à couvrir la cible.

Les éléments propres au portefeuille et au modèle ALM ayant été exposé, nous appliquons nos modèles génératifs à la modélisation de la FDB. Nous commencerons par expliciter comment la base de données a été obtenu ainsi que les détails techniques spécifiques aux différents modèles.

3.2 Mise en œuvre de l'approche

Pour rappel, notre objectif est la mise en place d'un proxy f_θ tel qu'en fonction de l'ensemble de nos variables explicatives X, on ait $Y = FDB = f_\Phi(X)$ avec f_Φ étant soit le COND-WGAN-GP soit le CVAE.

Il est convenable d'expliciter comment la base de données $\mathcal{D} = \{Y, X\}$ a été obtenue.

3.2.1 Construction de la base de données

Les variables constitutives de notre base de données ont été obtenues essentiellement en s'appuyant sur des études internes qui ont permis de les identifier comme pertinentes.

Ainsi, ces variables pertinentes sont : les intérêts techniques, les produits financiers, les chargements sur encours et le taux cible. Toutefois, les variables qui seront utilisées dans la base sont :

- **Le Taux de Rendement Comptable** (TRC) :
  Le choix de cette variable s'explique par le fait qu'elle représente les produits financiers et est un indicateur utilisé en pratique dans les prises de décisions puisque :

  A tout instant t, $\text{TRC}(t) = \frac{\text{Produits financiers en net}}{\frac{1}{2}(\text{Total actifs en t + Total actifs en t-1})}$

- **Les Provisions Mathématiques ALM ($\mathbf{PM}_{ALM}$)** :
  Les chargements sur encours et les intérêts techniques étant des pourcentages de la PM, son choix est naturel. De plus, il faut noter que d'après 3.1, la PM_{ALM} correspond au produit entre la $PM_{\text{déterministe}}$ et le coefficient de flexing.

- **Le Taux Cible** (TC) :
  Comme évoqué dans l'algorithme de participation aux bénéfices, les taux cibles sont des éléments indispensables et seront ainsi directement des inputs du modèle.
- **La courbe des taux ($courbe_{tx}$)** sera également une entrée additionnelle.

Il est important de noter que ces variables sont celles qui sont utilisées dans la phase d'entraînement initiale du modèle. Il est évident que dans des conditions réalistes de production des indicateurs, certaines données ne sont connues qu'après le lancement du modèle ALM. Notre objectif étant de se passer de ce modèle, nous mettrons en place des solutions pour être très réaliste dans la section 3.3.2.

Les données historiques ont été extraites de l'outil de modélisation interne ***Addactis Modeling***. Les données s'étendent de la clôture 2022 à celle de 2023 avec des granularités trimestrielles et annuelles. En totalité, nous avons des données par simulation sur huit trimestres et deux données annuelles. Pour un trimestre ou un annuel, les simulations

sont de tailles $N = 2000$ et l'horizon de projection est de $T = 50$ années pour chaque facteur de risque ou variable explicative d'intérêt. Ci-dessous un aperçu des différents jeu de données :

$$X = \begin{pmatrix} t_{1,1} & \cdots & t_{1,T} & tc_{1,1} & \cdots & tc_{1,T} & \cdots & trc_{1,1} & \cdots & trc_{1,T} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{N,1} & \cdots & t_{N,T} & tc_{N,1} & \cdots & tc_{N,T} & \cdots & trc_{N,1} & \cdots & trc_{N,T} \end{pmatrix} = (X_{i,j})_{i\in[\![1,N]\!],j\in[\![1,T]\!]}$$

$$Y = \begin{pmatrix} FDB_1 \\ \vdots \\ FDB_N \end{pmatrix} = (Y_i)_{i\in[\![1,N]\!]}^T$$

Avec $t_{i,j}$, $tc_{i,j}$, $PM_{i,j}$, $trc_{i,j}$ les notations respectives dans la base de données pour désigner les taux, les taux cibles, les provisions mathématiques et le TRC pour chaque simulation $i \in 1, \ldots, N$ et chaque pas de projection $j \in 1, \ldots, T$.

Après avoir décrit les différentes variables et notre base de données définitive, nous explicitons dans la suite les spécificités techniques des modèles mis en place.

3.2.2 Spécificités techniques liées à la modélisation

Pour calibrer le modèle, une approche classique en *Machine Learning* a été adoptée : 80 % de la base de données complète (2022) a été utilisée comme échantillon d'entrainement et les 20 % restants comme données de validation. Tous ces sous-échantillons ont été choisis aléatoirement. Concernant le traitement des données, une normalisation standard ($(X_{i,j} - \mathbb{E}(X_{i,j}))/\sqrt{Var(X_{i,j})}$) a été effectuée sur l'ensemble de toute la base pour éviter que certaines valeurs trop grandes ou trop petites ne biaise la calibration du modèle.

3.2.3 Détails des modèles, entrainement et optimisation

COND-WGAN-GP

Pour rappel, l'objectif de modélisation est : $Y = f_\Phi(X)$, $\Phi = (\theta, \phi)$, avec θ et ϕ respectivement les paramètres inhérents au générateur et au discriminant.

Dans nos expérimentations numériques, l'architecture qui a été mise en place pour le COND-WGAN-GP est illustrée par la figure 3.2.

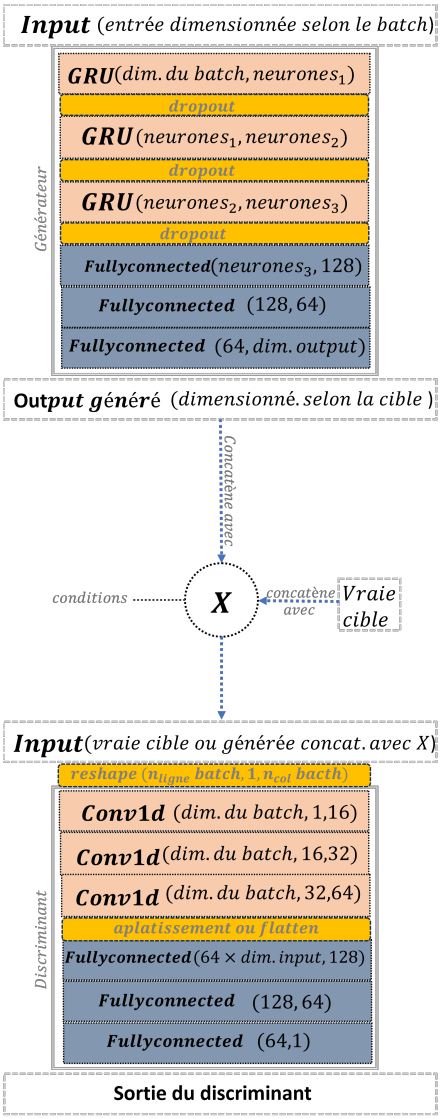

FIGURE 3.2 – Architecture du COND-WGAN-GP mis en place.

Afin de calibrer le modèle, en dehors des paramètres θ et ϕ, des hyperparamètres doivent aussi être optimisés. Il s'agit de :

- **Taille des *batch* :**
  Cet hyperparamètre est généralement utilisé pour faciliter l'apprentissage des différents algorithmes. En effet, Au lieu d'entraîner le modèle sur l'intégralité des données en une seule fois, le processus d'apprentissage se fait progressivement à l'aide de sous-échantillons, appelés lots ou *batch*. La taille du *batch* peut influencer la convergence de l'algorithme et les temps de calculs.

- **Pas d'apprentissage ou *learning rate* :**
  Il s'agit d'un paramètre propre à l'algorithme d'optimisation que nous utilisons. Il permet une convergence de vitesse appropriée. Dans notre contexte, nous avons un pas pour l'optimisation du générateur et un autre pour l'optimisation du discriminant.

  L'algorithme d'optimisation que nous utilisons est l'***Adaptive Moment Estimation* (Adam)**, qui combine les avantages de la descente de gradient stochastique classique avec des techniques de calcul adaptatif des taux d'apprentissage, améliorant ainsi la convergence et la performance du modèle.

- **Nombre de neurones dans chaque couche :**
  Il est important d'optimiser ces quantités, car trop peu de neurones dans les couches cachées peuvent entraîner un sous-apprentissage, tandis que trop de neurones peuvent rendre le modèle sujet au surapprentissage ou le rendre plus complexe que ce qui est nécessaire, ce qui peut augmenter les temps de calcul. Sur la figure 3.2, cela correspond aux éléments $neurones_1, neurones_2$ et $neurones_1$ pour les couches GRU.

- **Taux de *dropout* :**
  Il s'agit d'une technique de régularisation utilisée pour prévenir le surapprentissage [Srivastava *et al.*, 2014]. L'idée centrale est d'empêcher le réseau de devenir trop "adapté" aux données d'entraînement, ce qui pourrait réduire sa capacité à généraliser aux nouvelles données. En pratique, à chaque itération de l'entraînement, une fraction aléatoire des neurones dans le réseau est temporairement désactivée. Par exemple, si le taux de déconnexion est de 0,5, alors 50 % des neurones seront désactivés aléatoirement.

- **Paramètre de pénalité du gradient λ :**
  Comme évoqué plus haut, c'est l'un des hyperparamètres essentiels dans les GANs de Wasserstein permettant de stabiliser l'entraînement des modèles.

L'objectif de l'optimisation des hyperparamètres est de trouver la combinaison la plus optimale de ces paramètres, permettant ainsi un entraînement efficace et performant du modèle. Cette optimisation se réalise à l'aide de la bibliothèque `Python` *Optuna*. La théorie sous-jacente à cet outil est détaillée dans l'article qui l'introduit [Akiba *et al.*, 2019]. Son fonctionnement repose sur la minimisation d'une métrique définie par l'utilisateur en effectuant une recherche sur une plage de valeurs pour chaque hyperparamètre. Un tableau récapitulatif en annexe B.2 récence les optimaux retenus.

CVAE

L'architecture du CVAE est présentée en figure 3.3.

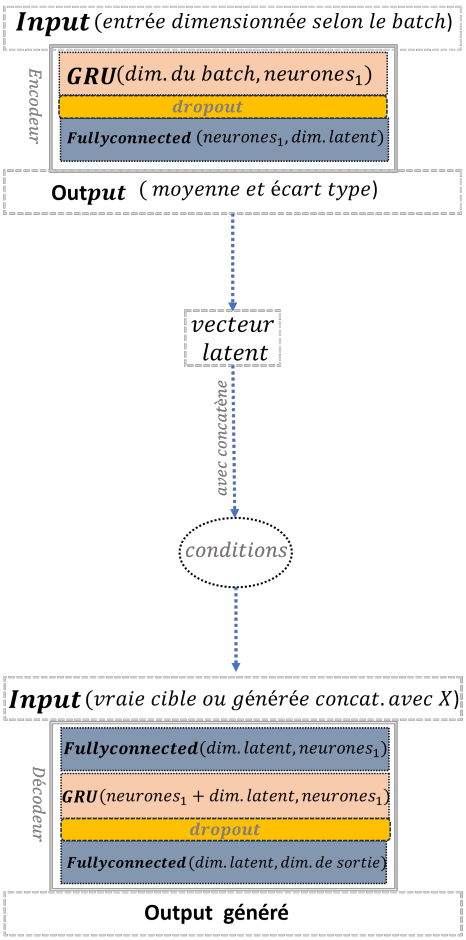

FIGURE 3.3 – Architecture du CVAE mis en place.

Une optimisation des hyperparamètres du CVAE, identiques à ceux du COND-WGAN-GP, ainsi que de la dimension de l'espace latent, a également été réalisée. Les valeurs optimales de ces hyperparamètres sont présentées en annexe B.2.

Les algorithmes permettant la mise en œuvre des deux modèles sont disponibles en annexe A.3. De plus, toutes les expérimentations numériques ont été réalisées sous *PyTorch* avec un ordinateur portable de processeur Intel Core i5-8250U CPU, 1.60GHz de 12 Go de mémoire RAM.

3.2.4 Métriques d'évaluation des modèles

Pour les problématiques de prédictions, une métrique usuelle est l'erreur quadratique moyenne (*Root Mean Square Error*) définie par : $RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i^{\text{réel}} - Y_i^{\text{prédit}})^2}$. Cependant, dans notre contexte, cette métrique s'avère inappropriée, car les modèles génératifs que nous utilisons ne visent pas à prédire des valeurs individuelles spécifiques. Il est donc plus pertinent d'adopter des métriques qui évaluent la qualité de la distribution générée par rapport à la distribution réelle des données. Les métriques retenues dans ce cadre sont les suivantes :

- **Moyenne** $(\mathcal{M})$ **:** $\mathcal{M} = \frac{1}{n}\sum_{i=1}^{n} Y_i$, où Y_i représente la $i-$ème réalisation de la variable Y.
- **Ecart-type**(σ) : $\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$, où $\bar{Y}$ est la moyenne de la variable concernée.

- **Quantile à 25%** (Q_1) et à **75%** (Q_3) :
  Une variable aléatoire X de fonction de répartition $F_X(x) : \mathbb{R} \to [0,1]$, a pour fonction quantile $F_X^- \to \mathbb{R}; \alpha \to inf\{x; F_X(x) \geq \alpha\}$. $\forall \alpha \in]0,1[$, $F_X^-(\alpha)$ correspond au quantile d'ordre α de X.
  Le quantile à 25% (respectivement 75 %) de X est la valeur en dessous de laquelle se situent 25% (respectivement 75 %) des observations de X.
- **Intervalle interquartile** (IQ) **:** correspond à $Q_3 - Q_1$
  L'IQ se concentre sur les valeurs centrales, en minimisant l'impact des valeurs qui se situent aux extrêmes.

- **Norme** L_2 (L_2) :
  Pour deux fonctions f et g, cette quantité est définie par :

$$\|f - g\|_2 = \left(\sum_{i=1}^{n} |f(x_i) - g(x_i)|^2\right)^{1/2}$$

  Dans notre contexte, il est question de calculer cette métrique entre la densité empirique des données générées et celle des données réelles.

- **Test statistique de Kolmogorov-Smirnov :**

  Il s'agit d'un test non paramétrique qui permet d'établir s'il y a une différence significative entre deux distributions. Ce test est basé sur la comparaison des fonctions de répartition des deux distributions. Les hypothèses du test sont les suivantes : H_0 : les deux échantillons proviennent de la même distribution ; H_1 : les deux échantillons proviennent de différentes distributions.

Si on considère $\{X_1, X_2, \ldots, X_m\}$ et $\{Y_1, Y_2, \ldots, Y_n\}$ deux échantillons indépendants issu des distributions F et G, respectivement. Les fonctions de répartition empirique correspondantes sont définies par :

$$F_m(x) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{\{X_i \leq x\}}; G_n(y) = \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}_{\{Y_j \leq y\}}$$

La statistique du test est définie par : $D_{m,n} = \sup_x |F_m(x) - G_n(x)|$.
Après avoir calculé cette statistique, une *p-value* est obtenue pour évaluer la significativité de la différence observée. Si la *p-value* est inférieure à un seuil choisi (5 % dans notre cas), on rejette H_0 et on conclut que les deux échantillons proviennent de distributions différentes.

Les différentes métriques de mesure de la qualité des modèles ayant été introduites, nous présentons les résultats que nous avons obtenus pour chacun des différents modèles.

3.3 Résultats et analyses

Les résultats obtenus ainsi que leur analyse seront présentés en fonction des différents travaux réalisés. Pour des raisons de confidentialité, les valeurs exactes ne seront pas divulguées pour les métriques quantitatives ; seules des erreurs relatives absolues seront mentionnées. De même, certaines valeurs sur les axes (ordonnée ou abscisse) des figures ou graphiques ne seront pas affichées.

3.3.1 Modèle général

Dans cette partie, nous utilisons les données de l'année 2022 et nous nous intéressons au modèle : $FDB = f_\theta(X)$ où X=(Taux, TRC, Taux Cible, PM_{ALM}).

L'idée est de calibrer le modèle et de valider sa capacité à se généraliser en supposant que toutes les variables explicatives soient directement observées.

Résultats lors de l'entrainement

Cette étape est la calibration initiale du modèle en se basant sur les architectures que nous avons élaborées.
Un élément intéressant à investiguer est le comportement des fonctions de pertes des différents modèles.

Dans la philosophie générale des GANs, le générateur et le discriminant sont en compétition dans un jeu à somme nulle, ce qui explique le caractère antagoniste que nous observons sur la figure 3.4 entre leurs fonctions de perte définie précédement par les équations (2.9) et (2.10).

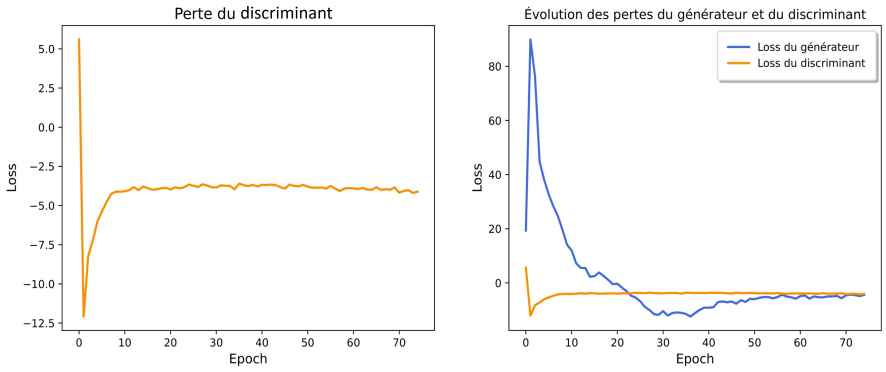

FIGURE 3.4 – Evolution de la perte du COND-WGAN-GP pendant l'entraînement.

De plus, comme mentionné dans la section introductive des modèles génératifs, les Wasserstein GANs avec pénalité sur les gradients sont conçus pour atténuer les problèmes d'instabilité souvent rencontrés lors de l'entraînement des GANs traditionnels. Dans notre cas, nous observons non seulement ce comportement antagoniste, mais également une convergence des fonctions de perte vers un équilibre stable, sans les fluctuations brusques et imprévisibles caractéristiques des GANs classiques. Cela suggère que notre COND-WGAN-GP est bien calibré.

De même, nous analysons la perte du CVAE illustrée par la figure suivantes :

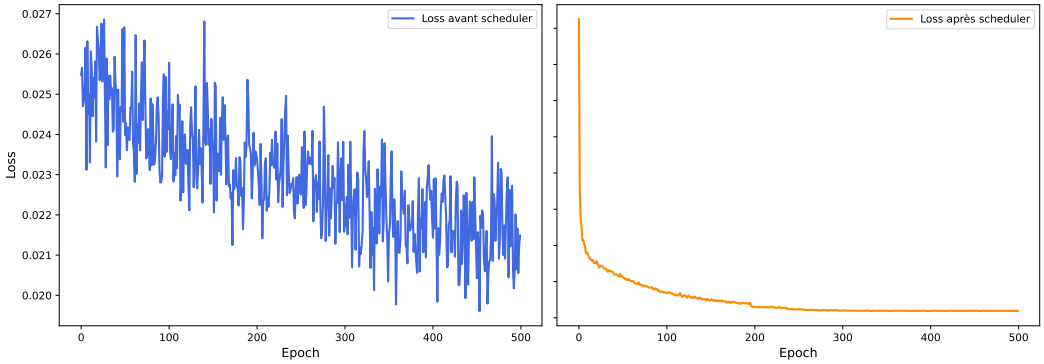

FIGURE 3.5 – Evolution de la perte du CVAE pendant l'entraînement avant et après le *learning rate* adaptatif.

Comme évoqué, l'allure de la fonction de perte est un indicateur de la qualité de calibration du modèle. Sur le graphique illustratif, on observe que lors de l'entraînement direct du modèle, la fonction de perte définie par l'équation 2.12 est difficilement minimisée, même après de nombreuses itérations (graphique de gauche de la figure 3.5). Ce comportement peut être le signe que le pas d'apprentissage (learning rate) n'est pas

optimisé pour les différentes phases de l'entraînement. Pour remédier à ce problème, nous avons mis en place une approche consistant à utiliser un *scheduler* afin de faire varier dynamiquement le pas d'apprentissage après un certain nombre d'optimisations. Par exemple, toutes les 10 itérations, le pas d'apprentissage est divisée par deux. Cette approche a permis de stabiliser l'apprentissage et d'apporter une correction au problème évoqué (graphique de droite de la figure 3.5).

Etape d'échantillonnage

Une fois, le modèle calibré, nous l'évaluons sur la base de test à l'aide des métriques que nous avons préalablement introduites. Ci-dessous, les valeurs de ces indicateurs quantitatifs :

Métriques	$\mathcal{M}$	σ	Q_1	Q_3	IQ	L_2
$\Delta^{\text{COND-WGAN-GP}}_{\text{métrique}}$	0,50%	2,07%	1,66%	0,19%	0,80%	0,03%
$\Delta^{\text{CVAE}}_{\text{métrique}}$	0,49%	1,66%	1,42%	0,15%	0,71%	0,01%

TABLE 3.2 – Comparaison quantitative de la performance des différents modèles. Lecture : plus la valeur de la variation est proche de 0, meilleur est le modèle pour la métrique concernée.

Où, pour une métrique donnée et modèle $\in \{\text{COND-WGAN-GP}, \text{CVAE}\}$,

$\Delta^{\text{modèle}}_{\text{métrique}} = \frac{|\text{prédit} - \text{réelle}|}{\text{réelle}}$ correspond à la variation relative absolue entre la valeur prédite et la valeur réelle.

De plus, pour le test de Kolmogorov-Smirnov, nous avons :

Modèle	$p - value$
COND-WGAN-GP	99.0%
CVAE	99.6%

TABLE 3.3 – Résultat test de Kolmogorov-Smirnov. Plus la valeur est proche de 100%, meilleur est le modèle.

Au regard de cette analyse quantitative, nous constatons que tous les indicateurs semblent indiquer que nos modèles génératifs réussissent bien dans leur tâche. Nous

complétons ces métriques par une comparaison graphique de la distribution des vraies valeurs FDB et celle prédite par nos deux modèles, en incluant une analyse des quantiles via des graphiques de quantile-quantile (QQ plots) illustré par la figure 3.6.

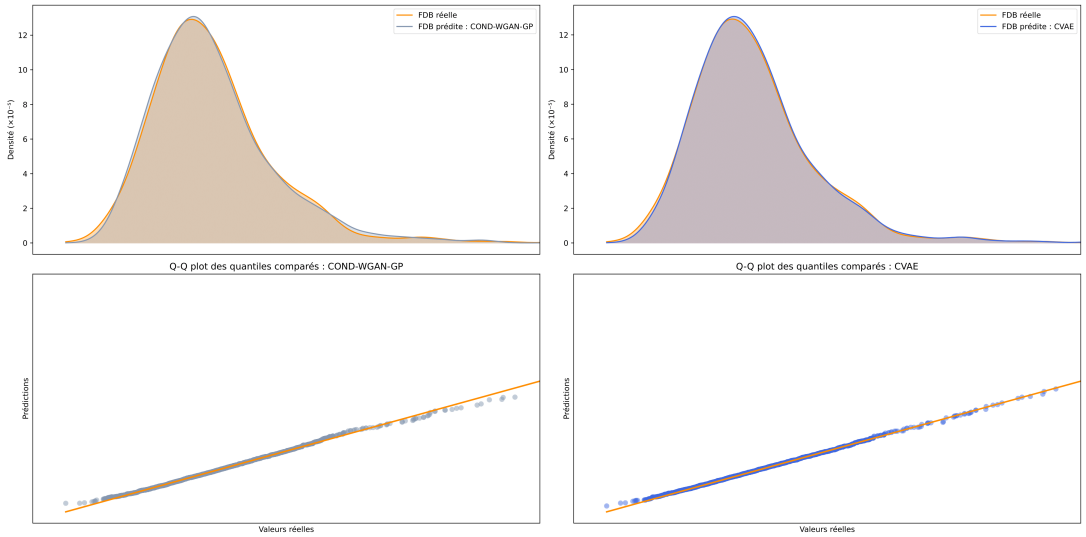

FIGURE 3.6 – Comparaison graphique entre la distribution réelle de la FDB et celle prédite avec le COND-WGAN-GP (à gauche) et avec le CVAE (à droite) ainsi que de leur *quantile-quantile plot*.

Dans la suite, nous avons investigué, à quel point nos modèles sont capables de capturer les différentes informations ajoutées.

Pour rappel, dans la calibration des modèles, nous concaténons à un moment, nos vraies données avec du bruit afin de guider la génération. L'objectif ici est de tester le modèle en lui donnant en premier lieu du bruit et de rajouter des informations au fur et à mesure. S'il est capable de capturer toutes ces informations, il devrait reproduire lors de l'ajout de la dernière variable la distribution complète illustrée avec la figure 3.6. Les résultats obtenus lors de ce test sont matérialisés par la figure 3.7 et le tableau 3.4.

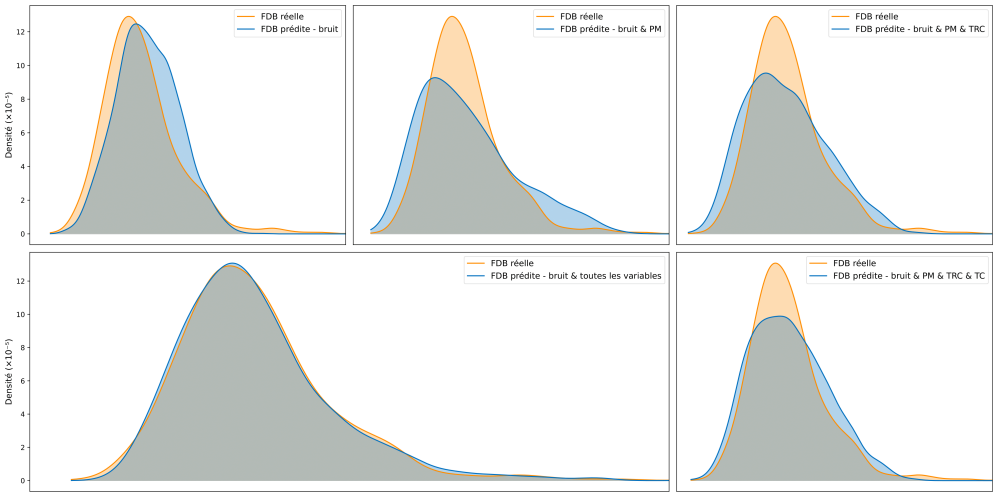

FIGURE 3.7 – Graphique illustrant la capacité du modèle COND-WGAN-GP à prendre en compte l'ajout progressif d'informations. Lecture : dans le sens des aiguilles d'une montre.

Variations	$\Delta_{\mathcal{M}}$	Δ_{σ}	Δ_{Q_1}	Δ_{Q_3}	Δ_{IQ}	Δ_{L_2}
Bruit	10,79%	21,55%	28,11%	11,88%	0,50%	0,27%
Bruit+PM	13,32%	24,57%	34,43%	17,49%	36,08%	0,29%
Bruit+PM+TRC	3,06%	17,19%	30,40%	14,42%	23,47%	0,21%
Bruit+PM+TRC+TC	1,42%	5,92%	18,27%	10,47%	8,37%	0,14%
Bruit+toutes les variables	0,50%	4,07%	1,66%	0,19%	0,80%	0,03%

TABLE 3.4 – Métriques quantitatives illustrant la capacité du modèle COND-WGAN-GP à prendre en compte l'ajout progressive d'informations. Lecture : plus la valeur de la variation est proche de 0, meilleur est le modèle pour la métrique concernée.

Où le $\Delta_{\text{métrique}}$ est la variation relative entre la vraie valeur de la métrique et celle obtenue par le modèle ; "+" désigne un ajout de variable.

Cette étude a également été réalisée pour le CVAE. Les différents résultats sont matérialisés par la figure 3.8 et le tableau 3.5.

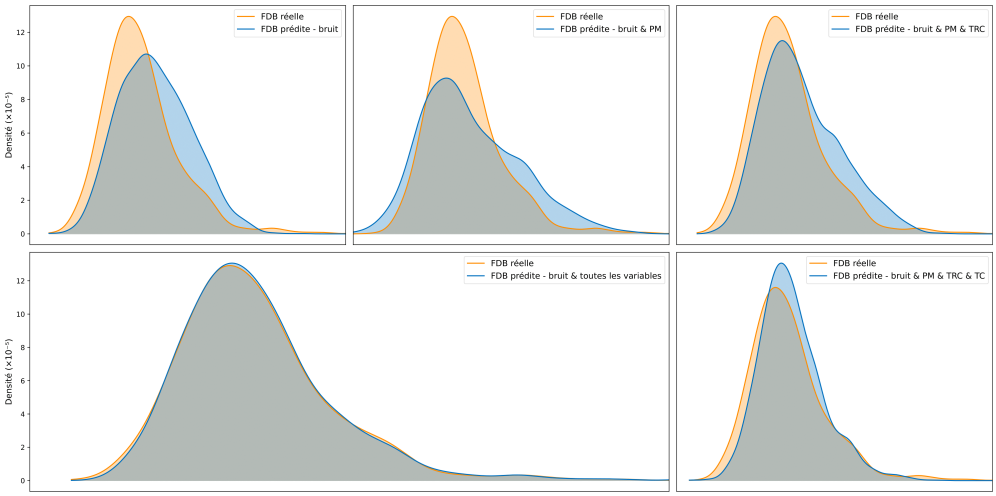

FIGURE 3.8 – Prise en compte de l'ajout progressif d'informations avec modèle CVAE. Lecture : dans le sens des aiguilles d'une montre.

Variations	$\Delta_{\mathcal{M}}$	Δ_{σ}	Δ_{Q_1}	Δ_{Q_3}	Δ_{IQ}	Δ_{L_2}
Bruit	25,32%	8,80%	31,07%	28,74%	20,36%	0,34%
Bruit+PM	12,70%	24,25%	20,15%	27,71%	44,26%	0,28%
Bruit+PM+TRC	2,56%	14,70%	12,64%	12,07%	11,83%	0,15%
Bruit+PM+TRC+TC	1,01%	7,93%	8,79%	4,56%	8,45%	0,09%
Bruit+toutes les variables	0,49%	1,66%	1,42%	0,15%	0,71%	0,01%

TABLE 3.5 – Métriques quantitatives illustrant la capacité du modèle CVAE à prendre en compte l'ajout progressive d'informations. Lecture : plus la valeur de la variation est proche de 0, meilleur est le modèle pour la métrique concernée.

Pour les deux modèles, les indicateurs visuels et quantitatifs montrent qu'en partant uniquement du bruit et en ajoutant progressivement des variables, et donc de l'information, les deux modèles sont capables d'intégrer ces ajouts dans leurs prédictions. Cela est cohérent, car dans la construction de nos modèles, ils doivent être en mesure de prendre en compte le contexte qui leur est fourni. Cela traduit une flexibilité des modèles par rapport aux variables d'entrées.

Temps de calculs et effet de la granularité des données

Le temps de calculs pour la calibration des différents modèles est fourni dans le tableau 3.6.

Modèle	Calibrage	Echantillonnage
COND-WGAN-GP	3H25 min	$<$ 20s
CVAE	2H15 min	$<$ 20s

TABLE 3.6 – Temps de calculs par modèle. H : heures, min : minutes, s : secondes.

De plus, nous avons analysé l'effet de la profondeur des données sur la qualité des prédictions du modèle en termes de moyenne et d'écart type. Cette comparaison est disponible dans le tableau 3.7.

$\Delta_{\text{Métrique}}$	$\mathcal{M}$	σ
Données T_1	23,27%	14,17%
Données T_1 **&** T_3	9,02%	7,23%
Données T_1 **à** T_4	4,67%	1,08%

TABLE 3.7 – Evolution de la moyenne et de l'écart-type en fonction de la granularité des données d'entrainement avec T_i correspondant au $i^{\text{ème}}$ trimestre. Modèle : CVAE.

Bien qu'une fois calibré, le coût d'échantillonnage soit quasiment nul (voir tableau 3.6), les performances des modèles s'améliorent à mesure que la profondeur des données d'entraînement augmente (voir tableau 3.7), comme c'est généralement le cas pour la plupart des modèles d'apprentissage profond.

3.3.2 Etudes de sensibilité

Dans cette section, les tests sont effectués sur les données de l'année 2023.

Modèle avec approximation du TRC et de la PM ALM

En pratique, pour un trimestre ou un annuel donné, en l'absence de modèle ALM, nous n'observons pas directement les projections de TRC et de PM_{ALM}. Ainsi, des sous-modèles ont été implémentés afin d'obtenir des estimations de ces quantités.

Le modèle général devient :

$$FDB = f_\Phi(\widehat{X}) \quad \text{où} \quad \widehat{X} = (\text{Taux}, \widehat{\text{TRC}}, \text{Taux Cible}, \widehat{PM_{ALM}})$$

Les variables du modèle $\widehat{\text{TRC}}$ sont l'allocation d'actifs et les données émanant du GSE tels que les indices action et immobilier, l'inflation, les taux.

En ce qui concerne l'estimation $\widehat{PM_{ALM}}$, elle se fait via un modèle intermédiaire de *flexing*. En effet, les $PM_{\text{déterministe}}$ étant connues, il suffit d'estimer les coefficients de flexing correspondants. Ce modèle prendra comme *inputs* le TRC et la courbe des taux.

La calibration des modèles TRC et *flexing*, s'est faite sur les données 2022 de la même manière que pour le modèle sans approximations. Ces deux variables étant chacune de dimension 50, les résultats présentés dans les figures 3.9 et 3.10 sont uniquement pour les pas de temps 10, 20 et 30.

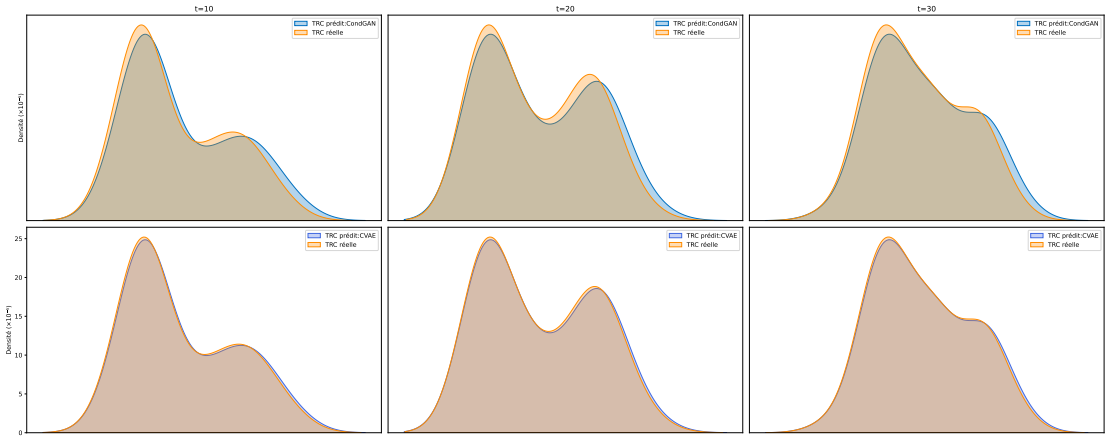

FIGURE 3.9 – Visualisation des prédictions du TRC aux instants $t = 10, 20, 30$. Première ligne : COND-WGAN-GP, deuxième ligne : CVAE.

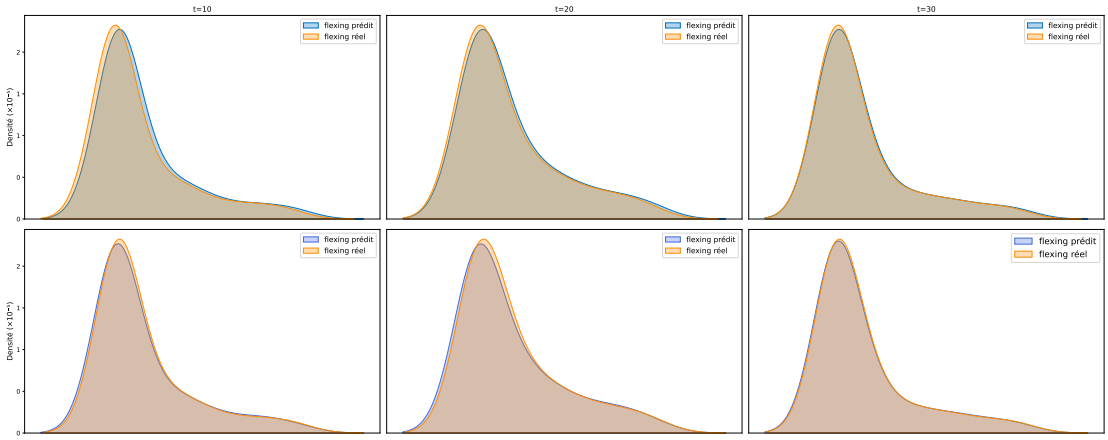

FIGURE 3.10 – Visualisation des prédictions du coefficient de *flexing* aux instants $t = 10, 20, 30$. Première ligne : COND-WGAN-GP, deuxième ligne : CVAE.

Pour les instants considérés, les approximations des modèles sont satisfaisants et indiquent qu'en absence de PM_{ALM} et de TRC comme c'est le cas sans modèle ALM,

les sous-modèles peuvent produire des *inputs* pour le modèle général d'estimation de la FDB.

Sensibilités du modèle face à des chocs de marché

Nous considérons les données annuelles 2023 et confrontons notre modèle à des situations de chocs de marché sous Solvabilité II. Le choc appliqué sur le facteur de risque immobilier correspond à une baisse de 25 % de la valeur de marché des actifs immobiliers et le choc de taux correspond à un choc à la hausse de la courbe de taux en fonction de la maturité.

Le modèle que nous utilisons dans la suite, est celui avec approximation du TRC et de la PM_{ALM} par le biais du coefficient de *flexing*. Après application de ces chocs, nous comparons la variation relative absolue des différents *Best Estimate* obtenus dans le tableau 3.8.

Modèle	COND-WGAN-GP	CVAE
$\Delta_{BE}^{\text{taux}}$	2,05%	1,37%
$\Delta_{BE}^{\text{immo}}$	1,79%	1,14%

TABLE 3.8 – Résultats BE pour des chocs de marché.$\forall$ choc $\in$ {Taux,immobilier}, $\Delta_{BE}^{\text{choc}} = \frac{|\widehat{BE}-BE|}{BE}$ où $\widehat{BE}$ correspond au BE prédit par le modèle. Lecture : plus la valeur de la variation est proche de 0, meilleur est le modèle concerné.

Pour les deux chocs, il ressort que les modèles réussissent à capturer leur effet sur le BE, mais pas dans les mêmes proportions. En effet, l'erreur relative des deux modèles est plus importante pour le choc de taux que pour le choc immobilier. Cela pourrait s'expliquer par le fait que les taux soient plus volatiles que l'immobilier et dans la structure des chocs appliqués, le choc de taux n'est pas fixe comme c'est le cas pour l'immobilier.

Effet sur le SCR

Pour cette partie, nous quantifions l'erreur de nos modèles en termes de SCR dans un contexte de calcul de marge pour risque définie par l'équation 1.1. En effet, le calcul de la marge pour risque nécessite de connaitre les SCR futurs qui sont en pratique difficiles à calculer pour les périmètres ayant des passifs longs comme c'est dans notre cas. Ainsi, des méthodes simplifiées de calcul de ce SCR comme la méthode proportionnelle sont envisageables [CEIOPS, 2009] sous réserve de satisfaire certaines hypothèses telles que l'invariabilité de la structure des sous-modules de souscription au fil du temps, la constance du SCR intangible sur la durée considérée. Sous ces hypothèses, nous avons :

$$SCR(t) = \frac{SCR(0)}{BE(0)} \times BE(t) \tag{3.2}$$

En utilisant les données historiques de notre portefeuille, nous constatons que ces hypothèses sont valides et la relation 3.2 est respectée (voir test de proportionnalité en annexe B.1). Nous nous proposons ainsi de mesurer à quel point nos erreurs dans la prédiction du BE influencent ce SCR. En utilisant cette formule avec comme instant initial, l'année 2021, nous avons en termes de SCR les résultats consignés dans le tableau 3.9.

Modèle	COND-WGAN-GP	CVAE
Δ_{SCR}	1,53%	1,11%

TABLE 3.9 – Résultats SCR. Lecture : plus la valeur de la variation est proche de 0, meilleur est le modèle.

$\Delta_{SCR} = \frac{|\widehat{SCR(t)} - SCR(t)|}{SCR(t)}$ où $\widehat{SCR(t)}$ correspond à une substitution de $BE(t)$ obtenu par notre modèle génératif dans 3.2.

Bien que ce calcul ait été effectué pour un horizon de temps proche de l'instant de calibration, les résultats dans le tableau 3.9 indiquent que les deux modèles sont capables d'approximer le BE, car en substituant le BE réel par celui prédit par nos modèles, l'erreur relative est faible.

En définitive, sur l'ensemble de tous les *backtests* réalisé ainsi que des sensibilités, il ressort que les deux modèles sont capables d'apprendre correctement les distributions des variables d'intérêts. Toutefois, au regard de nos métriques d'évaluation et des figures illustratives des distributions, le CVAE semble avoir une précision plus importante que le modèle implicite.

Dans la suite, nous explicitons certaines limites de nos modèles et de notre étude.

3.4 Limites de l'étude

D'entrée de jeu, la première limite que nous relevons est commune à la majorité des modèles de *Machine Learning*. En effet, ces modèles étant *data-driven*, leur capacité de généralisation dépend grandement de la taille des données sur lesquels ils sont calibrés.

Dans notre contexte, qu'il s'agisse du COND-WGAN-GP ou du CVAE ils ont tous les deux besoins d'une profondeur de données importante dans la phase de calibration. Par exemple, pour le COND-WGAN-GP, nous avons commencé à calibrer le modèle sur des données trimestrielles, puis annuelles. En observant la moyenne et l'écart type des prédictions, nous avons constaté que ceux-ci s'amélioraient à mesure que la granularité des données augmentaient. C'est ce qui a motivé notre choix de calibrage du modèle

final sur des données trimestrielles et annuelles. Les performances des modèles restent acceptables même si davantage de données issues de scénario adverses pourraient à priori les améliorer. Cette contrainte liée aux données a motivé des travaux de recherches dans le sens de la calibration de modèle génératifs dans des contextes où il n'y a pas assez de données. Voir par exemple le papier de [Buehler *et al.*, 2020].

En sus, les différents tests et sensibilités ont uniquement été réalisés sur le portefeuille sur lequel la calibration a été effectuée, alors qu'il aurait été intéressant d'évaluer également les modèles sur un autre portefeuille du même type et sur une entité complètement différente. De plus, même si les modèles n'ont pas été testés sur des entités complètement différentes, il est probable que leurs performances se dégradent dès lors que la structure de l'entité diffère de celle sur laquelle nous avons effectué notre étude. Pour accroître la capacité du modèle à se généraliser, il serait judicieux d'étendre la manière dont les modèles sont calibrés à d'autres entités en intégrant par exemple dans les données une variable qui permet d'identifier le type de périmètre et l'entité afin que le modèle apprenne également dans sa génération contextuelle à le faire selon l'entité ou le périmètre.

Chapitre 4

Conclusion

Le point de départ de ce mémoire a été de constater qu'avec la réglementation Solvabilité II, les compagnies d'assurances sont contraintes d'estimer leurs provisions techniques de la manière la plus prudente possible. Ces provisions comprennent une composante appelée le Best Estimate, qui nécessite l'usage de techniques de simulations numériques telles que la méthode standard de Simulation dans les Simulations (SdS). Conformément à cette réglementation, les assureurs doivent également effectuer, en plus des calculs ponctuels, des évaluations prospectives de certains indicateurs, comme le Best Estimate, dans le cadre de l'ORSA ou pour des prises de décisions stratégiques. Toutefois, cette approche SdS étant chronophage, de nombreuses alternatives ont vu le jour.

Après ce constat général, nous avons passé en revue l'état de l'art des différentes méthodologies alternatives, incluant des approches de *Machine Learning*, qui constituent le cadre de notre modélisation. Les modèles de Machine Learning utilisés jusqu'à présent s'inscrivaient essentiellement dans un contexte discriminatif. La principale contribution de nos travaux a été de les étendre à la classe des modèles génératifs. Ainsi, nous avons exploré théoriquement les principales architectures existantes et avons proposé quelques-unes qui semblaient répondre à la problématique posée.

La phase suivante a consisté à les mettre en œuvre afin de vérifier leur capacité à modéliser la distribution du *Best Estimate* et à prédire sa valeur. Pour ce faire, nous avons modélisé la partie stochastique du Best Estimate, à savoir la FDB, puisque c'est cette partie qui requiert en pratique l'utilisation de méthodes simulatoires.

Au terme de cette étude, en nous basant sur les métriques introduites dans ce mémoire et sur les visualisations réalisées, il ressort que notre approche est efficace pour la modélisation et la prédiction du *Best Estimate*. Par ailleurs, au-delà de ne pas être chronophage une fois la calibration effectuée, elle est plus flexible puisqu'elle offre la possibilité de se passer de certaines variables comme nous l'a montré l'étude de prise en compte progressive des informations. Cette flexibilité pourrait potentiellement en faire une alternative intéressante à la SdS dans certains contextes. Toutefois, elle présente éga-

lement des limites communes à la majorité des techniques de *Machine Learning*, telles que la nécessité d'une grande quantité de données d'entraînement et des temps de calibration parfois longs, en fonction de la complexité du problème. Nos analyses auraient également pu être orientées sur le même périmètre, mais en considérant différents portefeuilles, afin d'évaluer la sensibilité du modèle par rapport à sa calibration initiale.

De plus, un modèle hybride peut être mis en place en associant le CVAE au critique du WGAN. En effet, les sorties du CVAE peuvent être considérées comme les données générées par le générateur dans un WGAN classique, et ces sorties pourraient être améliorées par le processus antagoniste avec le critique. Comme les sorties du CVAE sont déjà satisfaisantes, leur association avec le discriminant d'un GAN devrait améliorer le résultat final.

Notre étude n'ayant exploré que deux modèles parmi la grande famille des modèles génératifs, il serait intéressant de s'intéresser aux autres catégories comme les modèles de diffusion et les *Normalizing flows*.

Enfin, les outils introduits dans ce mémoire peuvent être utilisés dans d'autres contextes, par exemple pour le calcul du capital économique, dans une vision de modèle interne. En effet, le calcul de ce capital dans un modèle interne peut se faire en utilisant des techniques simulatoires afin de déterminer la distribution des fonds propres économiques à horizon 1 an. Ainsi, il est possible d'utiliser les modèles génératifs pour apprendre à prédire cette distribution future.

Annexe A

Annexes du chapitre 2

A.1 Preuves discriminant et générateur optimaux

Dans cette section de preuves, les paramètres θ et ϕ des réseaux de neurones G et D ne seront pas explicitement mentionnés afin d'alléger les notations.

Preuve discriminant optimal.

Étant donné un G fixé, l'objectif de D est de maximiser $V(D,G)$. Ainsi :

$$V(G,D) = \int_x p_{\mathcal{D}}(x)\log(D(x))\,dx + \int_z p_z(z)\log(1-D(G(z)))\,dz$$
$$V(D,G) = \int_x \left[p_{\mathcal{D}}(x)\log(D(x)) + p_G(x)\log(1-D(x))\right]\,dx$$

$D^* = \arg\max_D V(D,G) = \arg\max_D \int_x p_{\mathcal{D}}(x)\log(D(x)) + p_G(x)\log(1-D(x))dx$
En écrivant $g(D) = a\log(D)+b\log(1-D)$ et en résolvant $\frac{\partial g}{\partial D} = 0$, on a bien $D^* = \frac{a}{a+b}$.

Preuve générateur optimal.

D'abord, nous prouvons que $p_G = p_{\text{data}} \Rightarrow V(D^*_G, G^*) = -\log 4$.

Partant de :

$$V(D^*_G, G) = \mathbb{E}_{x\sim p_{\mathcal{D}}}[\log \frac{p_{\mathcal{D}}}{p_{\mathcal{D}}+p_G}] + \mathbb{E}_{z\sim p_G}[\log \frac{p_G}{p_{\mathcal{D}}+p_G}]$$

Pour $p_{\mathcal{D}} = p_G$, on a : $D^*_G = \frac{1}{2}$ et $V(D^*_G, G^*) = -log(4)$

Prouvons maintenant que $V(D_G^*, G^*) = -\log 4 \Rightarrow \quad p_G = p_{\mathbb{D}}$

On a :

$$
\begin{aligned}
V(D_G^*, G^*) &= \mathbb{E}_{x\sim p_{\mathcal{D}}}\left[\log \frac{p_{\mathcal{D}}}{p_{\mathcal{D}}+p_G}\right] + \mathbb{E}_{x\sim p_G}\left[\log \frac{p_G}{p_{\mathcal{D}}+p_G}\right] \\
&= -\log 4 + \log 4 + \mathbb{E}_{x\sim p_{\mathcal{D}}}\left[\log \frac{p_{\mathcal{D}}}{p_{\mathcal{D}}+p_G}\right] + \mathbb{E}_{x\sim p_G}\left[\log \frac{p_G}{p_{\mathcal{D}}+p_G}\right] \\
&= -\log 4 + \mathbb{E}_{x\sim p_{\mathcal{D}}}\left[\log\left(2\times \frac{p_{\mathcal{D}}}{p_{\mathcal{D}}+p_G}\right)\right] + \mathbb{E}_{x\sim p_G}\left[\log\left(2\times \frac{p_G}{p_{\mathcal{D}}+p_G}\right)\right] \\
&= -\log 4 + \mathbb{E}_{x\sim p_{\mathcal{D}}}\left[\log\left(\frac{p_{\mathcal{D}}}{\frac{p_{\mathcal{D}}+p_G}{2}}\right)\right] + \mathbb{E}_{x\sim p_G}\left[\log\left(\frac{p_G}{\frac{p_{\mathcal{D}}+p_G}{2}}\right)\right] \\
&= -\log 4 + KL\left(p_{\mathcal{D}}\middle\|\frac{p_{\mathcal{D}}+p_G}{2}\right) + KL\left(p_G\middle\|\frac{p_{\mathcal{D}}+p_G}{2}\right) \\
V(D_G^*, G^*) &= -\log 4 + 2\cdot \mathcal{JS}(p_{\mathcal{D}} \parallel p_G)
\end{aligned}
$$

Comme par hypothèse, $V(D_G^*, G^*) = -log(4)$, on a : $\mathcal{JS}(p_{\mathcal{D}} \parallel p_G) = 0$, ce qui entraîne que $p_G = p_{\mathcal{D}}$.

A.2 Long Short-Term Memory

Les LSTM [Hochreiter *et al.*, 1997] sont conçus pour mieux capturer les dépendances à long terme. Ils introduisent une structure de cellule plus complexe comprenant des portes (ou*gate*) pour contrôler le flux d'informations. La structure d'une Cellule LSTM intègre de nouvelles composantes :

- Une *forget gate* ou Porte d'oubli (f_t : détermine quelles informations de l'état précédent ne sont plus utiles et doivent être oubliées ;
- Une *input gate* ou porte d'entrée (i_t) : identifie les nouvelles informations qui doivent être stockées dans l'état actuel de la cellule ;
- La porte de sortie ou *output gate* (o_t) : indique les informations de l'état de cellule qui doivent être utilisées pour effectuer les calculs.

Les équations de ce type de cellules sont les suivantes :

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), h_t = o_t \odot \tanh(C_t)$$

Avec $\odot$ le produit élément par élément.

A.3 Algorithme COND-WGAN-GP et CVAE

Algorithm 1 Calibrage du modèle CVAE.

for chaque *epoch* d'entraînement **do**
  for chaque minibatch (x, y) du jeu de données d'entraînement **do**
    • Transférer x et y sur le dispositif (CPU ou GPU).
    • Mettre à zéro les gradients de l'optimiseur.
    • Passer x et y à l'encodeur pour obtenir μ et log(var).
    • Rééchantillonner le vecteur latent z en utilisant μ et log(var).
    • Passer z et x au décodeur pour générer $\hat{y}$ (sortie reconstruite).
    • Calculer la perte totale (erreur de reconstruction + divergence KL).
    • Rétropropager la perte et mettre à jour les paramètres du modèle par descente de gradient :

$$\theta \leftarrow \theta - \eta \nabla_\theta \,(\text{Perte totale})$$

  end for
  • Calculer la perte moyenne sur l'ensemble du jeu de données d'entraînement pour cette *epoch*.
  • Ajuster le taux d'apprentissage avec le scheduler en fonction de la perte moyenne.
  • Afficher la perte moyenne de l'époque actuelle et le taux d'apprentissage.
end for
• Sauvegarder le modèle calibré dans un fichier.
Sortie : Modèle CVAE calibré avec les paramètres optimisés.

Algorithm 2 Calibrage du modèle COND-WGAN-GP.

for nombre d'*epoch* d'entraînement **do**
  for chaque minibatch (x, y) du jeu de données **do**
    • Échantillonner un minibatch de vecteurs latents $\{z^{(1)}, \ldots, z^{(m)}\}$ à partir d'une distribution normale multivariée.
    • Construire l'entrée conditionnelle pour le générateur : $z_{\text{cond}} = \text{Concat}(z, x)$.
    • Générer des données synthétiques $\tilde{y} = G(z_{\text{cond}})$.
    • Construire les paires conditionnelles :
    – Vraies données conditionnées : (y, x)
    – Fausses données conditionnées : $(\tilde{y}, x)$
    for k étapes du critique (discriminant) **do**
      • Calculer les sorties du critique pour les vraies et fausses données.
      • Calculer la pénalité de gradient pour renforcer la contrainte de Lipschitz.
      • Mettre à jour les paramètres du critique par descente de gradient :

$$\theta_D \leftarrow \theta_D - \eta_D \nabla_{\theta_D} (\text{Perte Critique} + \lambda_{\text{gp}} \times \text{Pénalité de gradient})$$

    end for
    • Calculer la perte du générateur à partir des sorties du critique.
    • Mettre à jour les paramètres du générateur par descente de gradient :

$$\theta_G \leftarrow \theta_G - \eta_G \nabla_{\theta_G} (\text{Perte générateur})$$

  end for
end for
Sortie : Modèle calibré avec les paramètres optimisés.

A.4 Diffusions et Normalizing Flows

A.4.1 Modèles de diffusion

Dans la philosophie des modèles de diffusion, il est question de générer des données à partir de bruit, mais pas de n'importe quelle manière. Dans un premier temps, les données originales sont bruitées au fur et à mesure et dans un second temps un processus inverse est enclenché pour avoir de nouvelles données.

Le papier séminal qui introduit les modèles de diffusion dans le contexte de l'apprentissage profond est sans doute celui de [Sohl-Dickstein *et al.*, 2015]. Les auteurs s'inspirent de la physique statistique et proposent un modèle qui dégrade lentement la structure d'une distribution de données complexe à travers un processus de diffusion avant (*forward*). Un autre processus de diffusion arrière (*backward*) est ensuite appris pour restaurer la structure des données à partir de la distribution bruitée, permettant de générer des échantillons réalistes à partir de la distribution simple du bruit.

En partant de ces idées, de nombreux autres travaux ont suivi pour les améliorer. Ces travaux récents ont conduit dans la littérature à trois grandes approches de modélisation du processus *forward*/*backward* : les *Denoising Diffusion Probabilistic Models (DDPMs)* [Ho *et al.*, 2020], les *Score-based Generative Models (SGMs)* [Song *et al.*, 2019] et les *Score Stochastic Differential Equations (Score SDEs)* [Song *et al.*, 2020].

Denoising Diffusion Probabilistic Models

Dans les DDPMs, le processus de bruitage et de génération est modélisé par des chaînes de Markov. Le processus *forward* introduit du bruit graduellement dans les données, tandis que le processus *backward* utilise des réseaux de neurones pour éliminer le bruit et générer de nouvelles données. Ci-dessous une illustration imagée de l'objectif visée (les détails techniques seront donné dans la suite) :

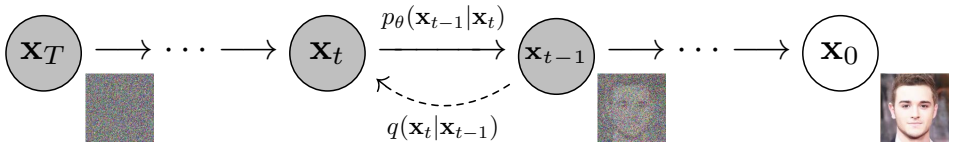

FIGURE A.1 – Illustration graphique DDPMs, extrait du papier original.

Processus de bruitage des données

Considérons notre ensemble de données $\mathcal{D} = \{x^i\}_{i=0,T}$. Supposons que x_0 provienne d'une distribution de données $q(x_0)$. Le processus de diffusion *forward* ajoute du bruit gaussien à chaque étape, transformant ainsi x_0 en une distribution gaussienne standard

x_T. Ce processus est décrit par :

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

où β_t est l'intensité du bruit. La distribution à n'importe quel pas de temps t en fonction de x_0 et d'un bruit gaussien ϵ est :

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, \sqrt{1-\bar{\alpha}_t}I)$$

Étape de diffusion inverse

Le processus de diffusion inverse supprime progressivement le bruit ajouté lors du processus *forward* pour récupérer les données originales. Modélisé par des distributions gaussiennes paramétrées par un réseau de neurones, ce processus prédit la distribution des données à l'étape précédente : $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma^2_\theta(x_t, t)I)$. Le modèle global de génération de données est :$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$.

L'optimisation de ce modèle s'apparente à celui d'un VAE dans la mesure où il est question de maximiser la probabilité des données observées en optimisant la borne supérieure variationnelle (*Variational Lower Bound*) de la log-vraisemblance négative.

Score-based Generative Models (SGMs)

Les SGMs reposent sur le concept de score de Stein [Hyvärinen, 2005], défini par $\nabla_x \log p(x)$. L'idée clé des SGMs est de perturber les données avec une séquence de bruit gaussien d'intensité graduelle et d'estimer les fonctions de score pour toutes ces distributions de données bruitées en entraînant un réseau de neurones.

Un réseau de neurones $s_\theta(x, t)$ est entraîné pour estimer la fonction de score $\nabla_x \log q(x_t)$ avec l'objectif d'entraînement : $\mathbb{E}_{t,x_0,x_t}\left[\lambda(t)\sigma_t^2 \left\|\nabla_x \log q(x_t) - s_\theta(x_t, t)\right\|^2\right]$

avec $\lambda(t)$ est une fonction de pondération.

Une fois entraîné, l'échantillonnage est effectué par des approches itératives comme la méthode dynamique de Langevin recuit.

Score SDEs

L'apport de cette approche est de formuler le processus de perturbation et de débruitage comme des solutions aux équations différentielles stochastiques (EDS). Les données sont bruitées avec un processus de diffusion régi par l'équation différentielle stochastique suivante :

$$dx = f(x, t)dt + g(t)dw, \quad t \in [0, T], \tag{A.1}$$

où $f(x, t)$ est le *drift* et $g(t)$le coefficient de diffusion associé au processus de Wiener standard $w(t)$.

Dans manière général, [Anderson *et al.*, 1982] montre que tout processus de diffusion sous la forme de A.1, peut être inversé en résolvant l'EDS à temps inversé suivante :

$$dx = \left[f(x,t) - g^2(t)\nabla_x \log p_t(x)\right] dt + g(t)d\hat{w},$$

avec $\hat{w}$ le mouvement brownien standard inversé.

Ci-dessous un résumé schématique du fonctionnement du modèle :

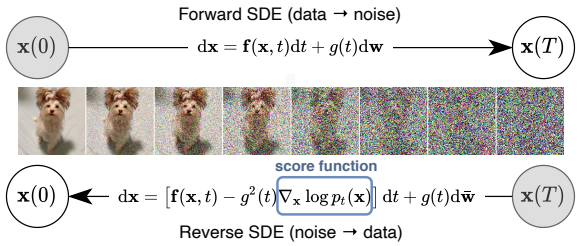

FIGURE A.2 – Résumé du processus dans les *Score-SDEs*, extrait du papier original.

La solution de cette EDS est approximée par un réseau de neurones dépendant du temps $s_\theta(x,t)$, qui en pratique tente d'approcher la fonction de score $\nabla_x \log p_t(x)$. Plutôt que d'estimer directement la fonction de score en raison des contraintes computationnelles, il estime $\nabla_x \log p_{0t}(x_t|x_0)$ et la calibration se résume au calcul de la quantité suivante : $\mathbb{E}_{t,x_0,x_t}\left[\lambda(t)\|s_\theta(x_t,t) - \nabla_x \log p_{0t}(x_t|x_0)\|^2\right]$

À la fin du processus d'entraînement, l'échantillonnage est effectué en utilisant des techniques de type Euler-Maruyama (EM) ou la méthode de flux d'équations différentielles ordinaires. Cette dernière méthode reformule l'équation de diffusion comme une équation différentielle ordinaire (ODE) :$dx = \left[f(x,t) - \frac{1}{2}g^2(t)\nabla_x \log p_t(x)\right] dt$

A.4.2 Normalizing flows

La notion de *normalizing flows* (NFs) a été initialement introduite dans les travaux de [Tabak *et al.*, 2010] et [Tabak *et al.*, 2013], puis popularisée par [Rezende *et al.*, 2015].

L'idée principale des NFs est de partir d'une distribution de probabilité simple et de lui appliquer une série de transformations pour obtenir une distribution cible plus complexe.

Considérons un ensemble de données $\mathcal{D}$ et une distribution cible $p_\mathcal{D}$. Il est question d'appliquer une transformation T, inversible et différentiable (tout comme T^{-1}), telle que $x = T(z)$, avec x provenant de $p_\mathcal{D}$ et z de la distribution de base p_z.

Sous ces conditions, la densité de x est obtenue par un changement de variables :

$$p_\mathcal{D}(x) = p_z(T^{-1}(x))\,|\det \mathbf{J}_{T^{-1}}(x)| \quad \text{avec} \quad \mathbf{J}_T(z) \quad \text{le} \quad \text{jacobien.}$$

Une propriété fondamentale des transformations inversibles et différentiables est leur capacité à être composées. Pour deux transformations T_1 et T_2, $T_2 \circ T_1$ reste inversible et différentiable : $(T_2 \circ T_1)^{-1} = T_1^{-1} \circ T_2^{-1}$ et $\det \mathbf{J}_{T_2 \circ T_1}(z) = \det \mathbf{J}_{T_2}(T_1(z)) \cdot \det \mathbf{J}_{T_1}(z)$.

Ainsi, pour obtenir une distribution cible, il est légitime de prendre T comme une série de transformations $f_1, \ldots, f_k$, où $T = f_k \circ \cdots \circ f_1$. Chaque transformation f_k prend un échantillon et le transforme jusqu'à obtenir x.

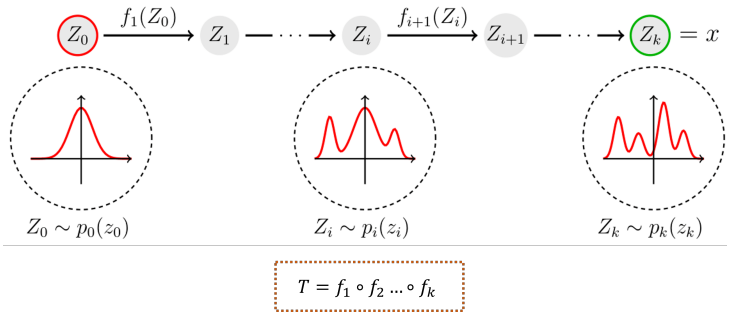

FIGURE A.3 – Exemple de transformation d'un NF à partir de $Z_0 \sim \mathcal{N}(0, 1)$ vers $x \sim p_{\mathcal{D}}$.

La calibration du modèle consiste à réduire la divergence de Kullback-Leibler :

$$D_{KL}[p_{\mathcal{D}}(x) \parallel p_{Z_k}(z_k; \theta)] = \mathbb{E}_{p_{\mathcal{D}}(x)} \left[\log \left(\frac{p_{\mathcal{D}}}{p_{Z_k}(z_k)} \right) \right]$$

Les NFs traditionnels transforment une distribution statique en une autre distribution cible. Cependant, certains problèmes nécessitent de modéliser des distributions évoluant au fil du temps, comme les séries temporelles financières. Ainsi pour ces cas, les NFs ont été étendu pour capturer des dynamiques temporelle dans les processus stochastiques.

Par exemple, [Deng *et al.*, 2020] propose un modèle utilisant un NF paramétré par θ et un mouvement brownien, où $X_t = F_\theta(W_t; t)$. L'apprentissage se fait selon la fonction de perte :

$$\sum_{i=1}^{n} \log p_{W_{t_i} | W_{t_{i-1}}}(w_{t_i}) - \log \left| \det \frac{\partial F_\theta}{\partial W_{t_i}}(w_{t_i}; t_i) \right|$$

Cette approche étant limitée à certaines catégories de processus. Une nouvelle approche plus efficace utilisant un mouvement brownien avec changement de temps, où $X_t = f_\theta(W_{\phi(t)}, \phi(t))$ [El Bekri *et al.*, 2024].

Annexe B

Annexes du chapitre 3

B.1 Visualisation SCR méthode proportionnelle

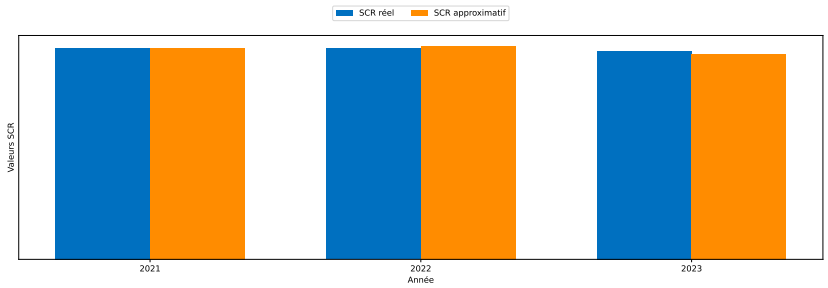

FIGURE B.1 – SCR réel et SCR approximé par la méthode proportionelle

B.2 Hyperparamètres optimaux

Taille des batch	***Learning rate***	**Nombre de neurones**	**Taux de *dropout***	λ
220	G : 0,017% D : 0,012%	GRU_1 : 223 GRU_2 : 227 GRU_3 : 39	40,6 %	1,704

TABLE B.1 – Hyperparamètres optimaux - COND-WGAN-GP.

Les hyperparamètres du CVAE sont dans le tableau B.2.

Taille des batch	***Learning rate***	**Nombre de neurones**	**Taux de *dropout***	**dimension de l'espace latent**
220	encodeur : 0,035% décodeur : 0,035%	$GRU_{encodeur}$: 484 $GRU_{dcodeur}$: 484	13,4 %	28

TABLE B.2 – Hyperparamètres optimaux - CVAE

Concernant le *learning rate*, comme mentionné dans la section 3.3.1, un pas d'apprentissage dynamique est utilisé grâce à l'emploi d'un *scheduler*. Ainsi, lors de la première itération, le pas est de 0,035 %, puis à la $10^{\text{ième}}$ itération, il est divisé par deux, et ainsi de suite.

Bibliographie

[Agostinelli *et al.*, 2013] Agostinelli, F., Anderson, L. et Honglak (2013). Adaptive multi-column deep neural networks with application to robust image denoising. *In Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

[Akiba *et al.*, 2019] Akiba, T., Sano, S., Yanase, T., Ohta, T. et Koyama, M. (2019). Optuna : A next-generation hyperparameter optimization framework. *In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA. Association for Computing Machinery.

[Anderson *et al.*, 1982] Anderson, Q, B. et D (1982). Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326.

[Arjovsky *et al.*, 2017] Arjovsky, M., Chintala, S. et Bottou, L. (2017). Wasserstein generative adversarial networks. *In* Precup, D. et Teh, Y. W., éditeurs : *Proceedings of the 34th International Conference on Machine Learning*, volume 70 de *Proceedings of Machine Learning Research*, pages 214–223. PMLR.

[Baschung, 2023] Baschung, L. (2023). Estimation de la fdb : mise en place d'un proxy sur la base d'indicateurs financier. *Institut des Actuaires*.

[Bauer, 2010] Bauer, B. (2010). Solvency ii and nested simulations - a least-squares monte carlo approach. *Proceedings of the 2010 ICA congress*.

[Berdah, 2020] Berdah, I. (2020). Calibration de modèles d'apprentissage sur des modèles épargne. *Institut des Actuaires*.

[Berger *et al.*, 2009] Berger, Eckmann, Harpe, P. et F. Hirzebruch, N. Hitchin, L. H. A. K. G. L. M. R. D. S. Y. G. S. (2009). Optimal transport : Old and new. *volume 338 of Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, Berlin, Heidelberg*.

[Bond-Taylor *et al.*, 2021] Bond-Taylor, S., Leach, A., Long, Y. et Willcocks, C. G. (2021). Deep generative modelling : A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7327–7347.

[Bortoli, 2023] Bortoli, V. D. (2023). cours generative modeling, master mathématiques, vision et apprentissage (mva), ens cachan.

[Bryis, 1994] Bryis, V. (1994). Life insurance in a contingent claim framework : pricing and regulatory implications. *The Geneva Papers on Risk and Insurance Theory 19, 53-72.*

[Buehler *et al.*, 2020] Buehler, H., Horvath, B., Lyons, T., Arribas, I. P. et Wood, B. (2020). A data-driven market simulator for small data environments. *arXiv preprint arXiv :2006.14498.*

[Cardon *et al.*, 2018] Cardon, J.-P., Cointet, A. et Mazières (2018). La revanche des neurones : L'invention des machines inductives et la controverse de l'intelligence artificielle. *Réseaux 2018/5 (n° 211), p. 173-220.*

[Carriere, 1996] Carriere, J. (December 1996). *Valuation of early-exercise price of options using simulations and non-parametric regression*, chapitre Volume 19, Issue 1,Pages 19-30. Insurance : Mathematics and Economics.

[CEIOPS, 2009] CEIOPS (2009). Ceiops advice for level 2 implementing measures on solvency ii :technical provision-article 86h, simplified methods and techniques to calculate technical provisions. *https://register.eiopa.europa.eu/CEIOPS-Archive/Documents/Advices/CEIOPS-L2-Advice-Simplifications-for-TP.pdf*.

[Cescutti, 2016] Cescutti, V. (2016). Estimés de solvabilité par méta-modélisation. *Institut des Actuaires.*

[Chakraborty *et al.*, 2023] Chakraborty, U., Reddy K S, Shraddha M. Naik, M., Panja, B. et Manvitha (2023). Ten years of generative adversarial nets (gans) : A survey of the state-of-the-art.

[Cho *et al.*, 2014] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. et Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation.

[Colvez *et al.*, 2022] Colvez, H., A., COULOUMY, A. et KAIS (2022). Generative neural networks for synthetic data generation in insurance : context and use cases,100

[Cox, 1958] Cox, D. (1958). The regression analysis of binary sequences. j. series b. *Journal of the Royal Statistical Society.*

[Cybenko, 1989] Cybenko (1989). Approximation by superpositions of a sigmoidal function. *Math. Control Signal Systems 2, 303–314 (1989). https ://doi.org/10.1007/BF02551274.*

[Deng *et al.*, 2021] Deng, R., Brubaker, M. A., Mori, G. et Lehrmann, A. M. (2021). Continuous latent process flows. *In Neural Information Processing Systems.*

[Deng *et al.*, 2020] Deng, R., Chang, B., Brubaker, M. A., Mori, G. et Lehrmann, A. (2020). Modeling continuous stochastic processes with dynamic normalizing flows. *In* Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. et Lin, H., éditeurs : *Advances in Neural Information Processing Systems*, volume 33, pages 7805–7815. Curran Associates, Inc.

[Domange, 2024] Domange, M. (2024). Compléments de mathématiques financières. *Cours Master 2 EURIA 2023/2024.*

[El Bekri *et al.*, 2024] El Bekri, L., Drumetz, F. et Vermet (2024). Time changed normalizing flows for accurate sde modeling. *In ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[Européenne, 2015] Européenne, C. (Janvier 2015). Règlement délégué (ue) 2015/35 https ://acpr.banque-france.fr/europe-et-international/assurances/reglementation-europeenne/solvabilite-ii.

[Florian *et al.*, 2021] Florian, Eckerli, J. et Osterrieder (2021). Generative adversarial networks in finance : an overview. *arXiv preprint arXiv :2106.06364*.

[Foster, 2023] Foster, D. (2023). *Generative Deep Learning : Teaching Machines to Paint, Write, Compose, and Play, 2^{nd}edition.O'REILLY*.

[Fukushima et J., 1980] Fukushima, K. et J. (1980). Neocognitron : A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics. 36 (4) : 193–202*.

[Gauville, 2017] Gauville, R. (2017). Projection du ratio de solvabilité : des méthodes de machine learning pour contourner les contraintes opérationnelles de la méthode des sds. *Institut des Actuaires*.

[Goodfellow *et al.*, 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. et Bengio, Y. (2014). Generative adversarial nets. *In* Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. et Weinberger, K., éditeurs : *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

[Gulrajani *et al.*, 2017] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. et Courville, A. C. (2017). Improved training of wasserstein gans. *In* Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. et Garnett, R., éditeurs : *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Guo *et al.*, 2017] Guo, X., Liu, X., Zhu, E. et Yin, J. (2017). Deep clustering with convolutional autoencoders. *In International Conference on Neural Information Processing*.

[Hans *et al.*, 2018] Hans, Buehler, L., Gonon, J., Teichmann, B. et Wood (2018). Deep hedging.

[Harrisson, 1979] Harrisson, K. (1979). Martingales and arbitrage in multiperiod securities markets. *Journal of Economic Theory*.

[Ho *et al.*, 2020] Ho, J., Jain, A. et Abbeel, P. (2020). Denoising diffusion probabilistic models. *In Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

[Hochreiter *et al.*, 1997] Hochreiter, S., Schmidhuber, J. et K (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Hornik *et al.*, 1989] Hornik, K., Stinchcombe, M. et White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

[Hyvärinen, 2005] Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709.

[Jebara, 2012] Jebara, T. (2012). *Machine learning : discriminative and generative*, volume 755. Springer Science & Business Media.

[Kingma *et al.*, 2014a] Kingma, DP, J. et Ba (2014a). Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*.

[Kingma *et al.*, 2014b] Kingma, DP, M. et Welling (2014b). Auto-Encoding Variational Bayes. *In 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

[Kingma et Max, 2019] Kingma, W. et Max (2019). An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*.

[Koursaris, 2011] Koursaris, A. (2011). Improving capital approximation using the curve-fitting approach.

[Kullback *et al.*, 1951] Kullback, S., L. et R. (1951). On information and sufficiency. *The Annals of Mathematical Statistics, 22 (1) : 79–86*.

[Kun *et al.*, 2022] Kun, Liu, X., Ning, S. et Liu (2022). Medical image classification based on semi-supervised generative adversarial network and pseudo-labelling. *In . Sensors 2022, 22,9967. https ://doi.org/10.3390/s22249967*.

[Lamberton, 2012] Lamberton, L. (2012). Introduction au calcul stochastique appliqué à la finance. *Editions Ellipses*.

[Lars *et al.*, 2018] Lars, Mescheder, A., Geiger, S. et Nowozin (2018). Which training methods for gans do actually converge ? *http ://arxiv.org/pdf/1801.04406v4*.

[LeCun *et al.*, 1998] LeCun, Y., Bottou, L., Bengio, Y. et Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Longstaff, 2001] Longstaff, S. (2001). Valuing american options by simulation : A simple least-squares approach. *Review of Financial studies*.

[Mao *et al.*, 2023] Mao, A., Mohri, M. et Zhong, Y. (2023). Cross-entropy loss functions : Theoretical analysis and applications. *In* Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S. et Scarlett, J., éditeurs : *Proceedings of the 40th International Conference on Machine Learning*, volume 202 de *Proceedings of Machine Learning Research*, pages 23803–23828. PMLR.

[Markowitz, 1952] Markowitz, H. (1952). Portfolio selection.

[Mcculloch *et al.*, 1943] Mcculloch, J., P. et W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*.

[Mehalla *et al.*, 2024] Mehalla, G., Darkiewicz, M., Krzeminski, C. et Francony (2024). Consistent equity risk-neutral valuation under climate stress tests. milliman whitepaper.

[Mehdi *et al.*, 2014] Mehdi, Mirza, M. et Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv :1411.1784*.

[Milnor et W., 1997] Milnor, J. et W. (1997). *Topology from the differentiable viewpoint*. Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ. Based on notes by David W. Weaver, Revised reprint of the 1965 original.

[Morisse, 2022] Morisse, A. (2022). Application des réseaux de neurones récurrents à l'estimation des calculs réglementaire. *Institut des Actuaires*.

[Ngwenduna, 2020] Ngwenduna (2020). Generative adversarial networks for actuarial use. *International Actuarial Colloquium*.

[Padala *et al.*, 2018] Padala, Manisha, S. et Gujar (2018). Generative adversarial networks (gans) : What it can generate and what it cannot ? *ArXiv, abs/1804.00140*.

[Papamakarios *et al.*, 2021] Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S. et Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(1).

[Parisi, 1981] Parisi, G. (1981). Correlation functions and computer simulations. *Nuclear Physics B*, 180(3):378–384.

[Pliska, 1981] Pliska, H. (1981). Martingales and stochastic integrals in the theory of continuous trading. *Stochastic Processes and their Applications*.

[Radford *et al.*, 2015] Radford, A., Metz, L. et Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. cite arxiv :1511.06434Comment : Under review as a conference paper at ICLR 2016.

[Rezende *et al.*, 2015] Rezende, J., K, M., L, S. et M (2015). Variational inference with normalizing flows. *In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1530–1538. JMLR.org.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron : A probabilistic model for information storage and organization in the brain. *nature, 323 (6088) : 533–536*.

[Rouchati, 2016] Rouchati, F. (2016). Projection du ratio de solvabilité d'un assureur retraite : les méthodes paramétriques (curve fitting et least squares monte carlo) peuvent-elles se substituer à la méthode des simulations dans les simulations ? *Mémoire, Institut des Actuaires*.

[Rudin, 2006] Rudin (2006). Real and complex analysis. *Tata McGraw-Hill Education*.

[Rumelhart *et al.*, 2015] Rumelhart, H., G. et Williams, R. (2015). An introduction to convolutional neural networks. *Psychological*.

[Rumelhart *et al.*, 1986] Rumelhart, H., G, W. et R (1986). The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review, 65 (6) : 386–408*.

[Schindelin, 2003] Schindelin (2003). A new metric for probability distributions. *IEEE Transactions on Information Theory, 49(7) :1858–1860*.

[Scholes *et al.*, 1973] Scholes, M., F. et Black. (1973). The pricing of options and corporate liabilities. *Journal of political Economy*, 81(3):637–654.

[Schrager, 2008] Schrager (2008). Replicating portfolios for insurance liabilities.

[Shuangshuang *et al.*, 2023] Shuangshuang, Chen, W. et Guo (2023). Auto-encoders in deep learning—a review with new perspectives. *Mathematics*, 11(8).

[Sohl-Dickstein *et al.*, 2015] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. et Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. *In International conference on machine learning*, pages 2256–2265. PMLR.

[Solveig *et al.*, 2022] Solveig, Flaig, G. et Junike (2022). Scenario generation for market risk models using generative neural networks. *Risks*.

[Song *et al.*, 2019] Song, J., Y. et Ermon, S. (2019). *Generative modeling by estimating gradients of the data distribution*. Curran Associates Inc., Red Hook, NY, USA.

[Song *et al.*, 2020] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S. et Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv :2011.13456*.

[Srivastava *et al.*, 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. et Salakhutdinov, R. (2014). Dropout : a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

[Tabak *et al.*, 2013] Tabak, G., C., V, T. et P (2013). A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics, vol. 66, no. 2, pp. 145–164*.

[Tabak *et al.*, 2010] Tabak, G., V.-E. et Vanden-Eijnden (2010). Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1): 217–233.

[Toubon, 2024] Toubon, H. (2024). Contexte réglémentaire et prudentiel. *Cours Master 2 EURIA 2023/2024*.

[Valentin et Cerisier, 2021] Valentin et Cerisier (2021). Orsa : Application de méthodes de machine learning dans le calcul de la solvabilité infra-annuelle. *Mémoire, Institut des Actuaires*.

[Yang *et al.*, 2023] Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B. et Yang, M.-H. (2023). Diffusion models : A comprehensive survey of methods and applications. *ACM Comput. Surv.*, 56(4).

[Zhang *et al.*, 2023] Zhang, W., Liu, Z., Chen, J., Wang, K. et Li (2023). On the properties of kullback-leibler divergence between multivariate gaussian distributions. *https ://arxiv.org/abs/2102.05485*.

[Zhu *et al.*, 2017] Zhu, J.-Y., Park, T., Isola, P. et Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *In Computer Vision (ICCV), 2017 IEEE International Conference on*.

[Zurfluh, 2019] Zurfluh, E. (2019). Utilisation du machine learning dans l'estimation du ratio de solvabilité. *Institut des Actuaires*.