Université de Montréal

Modélisation de la structure du silicium amorphe à l'aide d'algorithmes d'apprentissage profond

par
Massimiliano Comin

Département de physique
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures et postdoctorales
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en physique

août, 2018

Université de Montréal
Faculté des études supérieures et postdoctorales

Ce mémoire intitulé:

Modélisation de la structure du silicium amorphe à l'aide d'algorithmes d'apprentissage profond

présenté par:

Massimiliano Comin

a été évalué par un jury composé des personnes suivantes:

Normand Mousseau,	président-rapporteur
Laurent Lewis,	directeur de recherche
Aaron Courville,	membre du jury

Mémoire accepté le: 31 août 2018

RÉSUMÉ

Le silicium amorphe est le système canonique pour l'étude des matériaux désordonnés de par son importance technologique et son intérêt théorique fondamental. En effet les détails de sa structure atomique sont encore aujourd'hui mal connus, et son étude théorique se base essentiellement sur des simulations numériques. Mais les méthodes Monte Carlo pour la génération des réseaux aléatoires continus voient leur réalisme dépendre fortement de la description du paysage énergétique considérée.

Alors que les approches *ab initio* fournissent une description fidèle, leur application se limite à des systèmes de quelques centaines d'atomes au maximum. Les potentiels empiriques constituent en revanche une alternative efficace permettant la simulation de systèmes allant jusqu'à un million d'atomes au prix d'une fiabilité réduite.

Cependant les avancées récentes en apprentissage automatique ont permis l'émergence de modèles génératifs profonds capables d'approximer des fonctions complexes en haute dimension à partir d'observations, qui ont démontré un grand succès dans des tâches de synthèse d'images et sonore. De par leur efficacité, ces derniers ouvrent alors la voie à un meilleur compromis entre performance et réalisme pour la modélisation des systèmes désordonnés.

Dans le but d'étudier cette alternative, un réseau de neurones convolutif a été entraîné avec succès pour approximer la surface d'énergie potentielle de Stillinger-Weber du silicium amorphe avec une erreur quadratique moyenne 5.095 meV par atome, correspondant à 0.16% de l'énergie atomique. Ensuite, un modèle génératif profond, l'Auto-Encodeur de Wasserstein, a été entraîné pour l'apprentissage de la distribution atomique du silicium amorphe. Celui-ci génère des configurations qualitativement réalistes présentant un désordre structurel trop prononcé, ce qui confirme la viabilité de la méthode.

Mots clés: Silicium amorphe, Apprentissage profond.

ABSTRACT

Amorphous silicon is a canonical system for the study of disordered materials because of both its technological importance and fundamental interest. The details of its atomic structure are not yet well-known, and its theoretical study relies mainly on numerical simulations. But Monte Carlo approaches for generating continuous random networks show a realism that depend heavily on the considered description of the energy landscape.

Ab initio methods provide a faithful description but are limited to small systems, typically of a few hundreds of atoms. On the other hand, empirical potentials are efficient alternatives as they enable the modeling of large-scale systems up to a million atoms, at the price of a reduced reliability.

Recent advances in machine learning have led to the emergence of powerful deep generative models that are able to approximate complex high-dimensional functions from a dataset, which have shown great success in difficult generation tasks such as image and audio synthesis. Their efficiency lead the way to a better compromise between performance and realism for the modelization of disordered systems.

In order to explore this alternative, a convolutional neural network is trained to approximate the potential energy surface of amorphous silicon as given by the Stillinger-Weber potential, which resulted in a root mean square error of 5.05 meV per atom, corresponding to 0.16% of the atomic energy. Then a deep generative model, the Wasserstein Auto-Encoder, is trained to generate amorphous configurations. The resulting model generates qualitatively realistic configurations, although with a strong structural disorder, thus confirming viability of the method.

Keywords Amorphous Silicon, Deep Learning.

TABLE DES MATIÈRES

RÉSUMÉ . . . . . **iii**

ABSTRACT . . . . . **iv**

TABLE DES MATIÈRES . . . . . **v**

LISTE DES TABLEAUX . . . . . **vii**

LISTE DES FIGURES . . . . . **ix**

LISTE DES ANNEXES . . . . . **xvi**

LISTE DES ABRÉVIATIONS . . . . . **xvii**

NOTATION . . . . . **xix**

DÉDICACE . . . . . **xx**

REMERCIEMENTS . . . . . **xxi**

INTRODUCTION **1**

CHAPITRE 1 : PROPRIÉTÉS ET MODÉLISATION DU SILICIUM AMORPHE **3**

1.1 Motivation 3

1.2 Propriétés thermodynamiques des solides amorphes 3

1.3 Fabrication du silicium amorphe 6

1.4 Propriétés structurelles du silicium amorphe 7

1.5 Modélisation de la structure atomique 13

1.6 Approches automatiques pour la modélisation de la structure atomique . 18

CHAPITRE 2 : APPRENTISSAGE AUTOMATIQUE **20**

2.1 Contexte général et définitions 21

2.2 Cadre probabiliste de l'apprentissage supervisé 25
2.3 Estimateurs, Biais et Variance 31
2.4 Convergence de l'apprentissage et capacité 35
2.5 Apprentissage non supervisé 37
2.6 Entraînement d'un algorithme d'apprentissage 39
2.7 Régularisation 42

CHAPITRE 3 : APPRENTISSAGE PROFOND : RÉSEAUX DE NEURONES 45
3.1 Perceptron et Perceptron Multicouche 45
3.2 Rétropropagation des gradients 51
3.3 Réseaux convolutifs 52
3.4 Modèles génératifs profonds à variables latentes 56
3.5 Réseaux Accusatoires Génératifs (GAN) 58
3.6 Auto-Encodeurs Variationnels (VAE) 64

CHAPITRE 4 : APPRENTISSAGE DE LA SURFACE D'ÉNERGIE POTENTIELLE 69
4.1 La surface d'énergie potentielle 69
4.2 Représentation par des fonctions de symétrie généralisées 72
4.3 Applications spécifiques du potentiel généralisé 75

CHAPITRE 5 : MODÈLES GÉNÉRATIFS PROFONDS POUR LA STRUCTURE ATOMIQUE DU SILICIUM AMORPHE 83
5.1 Construction de l'ensemble de données 84
5.2 Apprentissage de la SEP 86
5.3 Modèles génératifs profonds 93

CONCLUSION ET PERSPECTIVES 107

BIBLIOGRAPHIE 110

LISTE DES TABLEAUX

1.I Propriétés structurelles expérimentales d'échantillons de silicium cristallin et amorphe : densité (ρ), coordination (C), position des pics (r) et déviation standard (σ) des deux premiers pics de la FDR [31]. 12

2.I Exemples d'applications à la physique des matériaux de certains algorithmes d'apprentissage supervisé et non supervisé [53]. . . . 23

2.II Propriétés des différents problèmes d'apprentissage automatique, où on considère une entrée x, une cible y et une fonction f cherchant à approximer leur relation. Les fonctions objectif mesurent respectivement l'erreur de classification, l'erreur quadratique et la log-vraisemblance négative. Elles constituent une mesure de l'erreur commise par le modèle f, il est donc souhaitable de les minimiser. 24

3.I Liste de certaines fonctions de transfert communes en apprentissage profond et de leurs propriétés. (M=monotone, D=dérivée monotone, I=approxime l'identité en 0, P=renvoie une probabilité) . 50

5.I Paramètres du potentiel de SW modifié [61]. 85

5.II Propriétés de l'ensemble de données $\mathbb{D}$: nombre d'exemples N, valeur moyenne de l'énergie potentielle totale (cibles) $\langle E \rangle$, déviation standard de l'énergie potentielle totale σ_E, valeur moyenne de la distance entre premiers voisins $\langle r_0 \rangle$, valeur moyenne de l'angle de liaison $\langle \theta_0 \rangle$. 86

5.III Comparaison de la performance des deux architectures MLP et CNN pour la régression des énergies potentielles. Chaque modèle est formé d'une première couche appliquant la transformation en fonctions de symétrie. Le MLP est composé de 5 couches denses avec des fonctions de transfert ReLU et une couche de sortie linéaire. Le CNN est composé de 5 blocs résiduels convolutifs avec des filtres scalaires, des fonctions de transfert ReLU et une couche de sortie linéaire. L'entraînement a été fait avec ADAM dans les deux cas avec les hyper-paramètres $\eta = 10^{-4}, \beta_1 = 0.5, \beta_2 = 0.9, \lambda = 10^{-6}$. 91

5.IV Hyper-paramètres utilisés pour l'entraînement du WAE : taille des mini-lots B, pas de gradient pour l'auto-encodeur et le discriminateur η_{AE}, η_D, premier et deuxième facteur d'inertie pour ADAM β_1, β_2, facteur de l'échéancier des paramètres w et dimension de l'espace latent $\dim z$. 98

5.V Comparaison des propriétés structurelles des échantillons générés par le WAE, de l'ensemble de données $\mathbb{D}$ et de l'échantillon expérimental recuit [31] : densité (ρ), coordination (C), position des pics (r) et déviation standard (σ) des deux premières couches de voisins, telles que définies par $r_c = 2.95$. 103

LISTE DES FIGURES

1.1 Dépendance qualitative du volume et de l'entropie en fonction de la température pour différentes transitions de phase (modifié de [65]). 5

1.2 Entropie d'excès entre les phases liquide et amorphe en fonction de la température (modifié de [65]). 6

1.3 Configuration atomique du silicium cristallin (c-Si) *(gauche)* et du silicium amorphe (a-Si) *(droite)* générée par la méthode décrite en [4]. 8

1.4 *a)* Conformation de type *bateau* et *b)* conformation de type *chaise* pour des anneaux à six atomes [65]. 9

1.5 Fonction de distribution radiale expérimentale d'un échantillon de silicium amorphe pur après un recuit à 600° C [31] comparée à celle d'un cristal (modélisation numérique). La structure cristalline peut être reconstituée à partir des positions des couches successives d'atomes qui sont bien définies, contrairement au cas du silicium amorphe. 12

1.6 Illustration d'une transposition de liens dans la méthode WWW [63]. 16

2.1 Illustration du concept de sous-apprentissage et de sur-apprentissage pour un ensemble de points $y = x^2 + N(0,\sigma)$, où $N(0,\sigma)$ est une distribution Gaussienne centrée de variance σ. Un modèle a) linéaire, b) polynomial de degré 2 et c) fonctionnel de haute capacité sont ajustés sur le même ensemble de points. Le modèle a) commet une grande erreur d'ajustement car l'hypothèse d'une relation linéaire est trop simple. Le modèle c) présente l'erreur la plus faible sur l'ensemble de points mais n'explique pas la loi sous-jacente aux données car l'ajout d'une nouvelle observation résulterait en une grande erreur d'ajustement. 32

2.2 Erreur d'entraînement et de généralisation en fonction de la capacité [19]. Lorsque la capacité est faible, l'erreur d'entraînement et de généralisation sont grandes : c'est le régime de sous-apprentissage. Lorsque l'on augmente la capacité, l'erreur d'entraînement décroît toujours, mais au delà de la capacité optimale le gap entre celle-ci et l'erreur de généralisation augmente : c'est le régime de sur-apprentissage. 33

2.3 Illustration du dilemme biais-variance [19] : lorsque la capacité augmente, le biais (bleu) tend à diminuer tandis que la variance (vert) tend à augmenter. Il y a une capacité optimale correspondant au minimum de l'erreur de généralisation (rouge), qui démarque la séparation entre les régimes de sur-apprentissage et de sous-apprentissage. 35

3.1 Illustration du fonctionnement d'un neurone formel en analogie avec un neurone biologique. Les entrées correspondent aux dendrites, les poids aux synapses, la fonction de transfert au noyau et la sortie à l'axone. 47

3.2 Graphe d'un Perceptron Multicouche de profondeur $D = 2$: chaque vertex représente un neurone formel et chaque lien représente une connexion entre les neurones. Chaque neurone d'une couche est connecté à tous les neurones de la couche précédente. 49

3.3 Schéma de l'application de quatre couches convolutives à une image. La largeur correspond à la dimension des canaux, tandis que les dimensions transversales correspondent aux dimensions spatiales de l'image. Chaque couche convolutive est appliquée avec un pas $s = 2$, ce qui réduit les dimensions spatiales de l'image. La sortie de chaque couche est un tenseur de rang 3 dont les dimensions sont (k,n,n) où k est le nombre de filtres appliqué et n la dimension spatiale de sortie de l'image. Si le réseau est entraîné par mini-lots de données, ces opérations sont faites en parallèle, et chaque mini-lot d'images est un tenseur de rang 4 de dimensions (b,k,n,n), où b est la taille du mini-lot. 55

3.4 Schéma du fonctionnement d'un GAN pour la génération de configurations atomiques de a-Si. Le générateur G transforme le bruit aléatoire entrant en une configuration atomique et essaye de tromper le discriminateur D. Le discriminateur essaye de différentier les configurations réelles (de l'ensemble d'entraînement) des configurations fausses (générées). 59

3.5 Architecture du VAE. L'encodeur $q_\phi(\mathbf{x})$ projette les données $\mathbf{x}$ vers une description latente $\mathbf{z}$. Il est entraîné à faire correspondre la distribution de ces dernières $Q_\phi(\mathbf{z}|\mathbf{x})$ à une distribution antérieure $p(\mathbf{z})$ (typiquement $\mathcal{N}(0,1)$. Le décodeur $g_\theta(\mathbf{z})$ reconstruit ces données à partir de $\mathbf{z}$. Il est entraîné à minimiser l'erreur de reconstruction entre $\mathbf{x}$ et $\tilde{\mathbf{x}}$. 67

4.1 Fonctions de symétrie radiale $\mathcal{G}^1$,$\mathcal{G}^2$ et $\mathcal{G}^3$ en fonction de la distance inter-atomique entre deux atomes quelconques [6]. Pour les fonctions $\mathcal{G}^2$ et $\mathcal{G}^3$ un rayon de coupure $r_c = 11.3$ Bohr a été utilisé et le paramètre η en c) a été fixé à 3.0Bohr^{-2}. 74

4.2 Contribution angulaire aux fonctions de symétrie $\mathcal{G}_4$ et $\mathcal{G}_5$ pour un système triatomique. La symétrie par rapport à $\theta = 180°$ est bien présente, et le maximum de la fonction cosinus est contrôlé par le paramètre $\lambda = \pm 1$. 75

4.3 Architecture du réseau de neurones pour la régression du potentiel inter-atomique [6]. La dimension des mini-lots correspond aux différents atomes de la configuration. Les coordonnées cartésiennes sont premièrement transformées en fonctions de symétrie puis ces dernières sont traitées par un perceptron multicouche. Les énergies sont finalement sommées pour obtenir l'énergie totale. 76

4.4 Fonctions de distribution radiale d'une cellule de 64 atomes de silicium liquide à 3000 K, obtenues à partir de simulations de dynamique moléculaire à 3000 K pendant 20 ps (8 ps pour la DFT) [8]. On remarque une bonne correspondance entre la FDR issue du potentiel DFT (noir) et celle obtenue à partir du potentiel appris par le réseau de neurones (rouge). Les FDR issues d'autres potentiels empiriques s'en écartent notablement. 78

4.5 Comparaison de la fonction définie en 4.3 et de la force associée $F = -\nabla E$ avec l'ajustement associé appris par le réseau de neurones [6]. 79

4.6 Énergie en fonction du volume atomique pour différentes phases du silicium calculées à l'aide de la DFT et l'approximation de la densité locale et calculées avec le potentiel issu du réseau de neurones [25]. 80

5.1 Fonctions de distribution radiale de 4 éléments de l'ensemble de données. 87

5.2 Architecture du MLP pour l'apprentissage de la SEP. Chaque bloc représente une couche de neurones dont la couleur spécifie le type, le nom indique le nombre de paramètres et la fonction de transfert, et les dimensions du tenseur de sortie de la couche sont indiquées. La première dimension correspond toujours à celle des mini-lots : chaque élément dans cette dimension est traité en parallèle par le réseau. Le bloc $\mathcal{G}$ correspond au calcul des fonctions de symétrie tandis que le bloc Σ effectue une somme. 88

5.3 Architecture du CNN pour l'apprentissage de la SEP, et structure des blocs résiduels utilisés. L'opération $\frown$ correspond à la concaténation sur la dimension des canaux du tenseur. Chaque bloc représente une couche de neurones dont la couleur spécifie le type, le nom indique le nombre de paramètres et la fonction de transfert, et les dimensions du tenseur de sortie de la couche sont indiquées. La première dimension correspond toujours à celle des mini-lots : chaque élément dans cette dimension est traité en parallèle par le réseau. Les convolutions ont toutes des filtres de taille 1, le nombre de paramètres indiqué correspond donc au nombre de filtres de la couche. 89

5.4 Erreur quadratique moyenne (sur 128 exemples) d'entraînement du MLP et du CNN. Les courbes pleines sont le résultat d'un filtrage de hautes fréquences de type Savitzky-Golay. 90

5.5 Architecture du WAE utilisé pour la modélisation de la structure atomique du a-Si. Chaque bloc représente une couche de neurones, dont la couleur spécifie le type, le nom indique le nombre de paramètres et la fonction de transfert (ou la nature de la couche) et les dimensions du tenseur de sortie de la couche sont indiquées. Les couches d'entrée **x** et **y** représentent respectivement l'approvisionnement d'un mini-lot de coordonnées atomiques et d'énergies potentielles. La couche d'entrée **z** représente l'approvisionnement d'un vecteur latent aléatoire tiré d'une distribution normale $\mathcal{N}(0,1)$ ou $\mathcal{N}(\mu,\Sigma)$, où μ,Σ sont les vecteurs de sortie de l'encodeur. La couche $\mathcal{G}$ représente le calcul des 108 fonctions de symétrie. La première dimension des tenseurs est celle des mini-lots, la deuxième celle des canaux et la troisième est la dimension spatiale sur laquelle les convolutions sont appliquées. L'opération $\frown$ symbolise la concaténation sur la deuxième dimension des tenseurs entrants. 96

5.6 Erreur d'entraînement du WAE en fonction des itérations de gradient. Une époque correspond à 3593 itérations. Seule la partie accusatoire de la fonction objectif de l'auto-encodeur $\mathbb{E}_{\mathbf{x}\sim\mathbb{D}} \log f_\gamma(q_\phi(\mathbf{x},\mathbf{y}))$ est illustrée. Les courbes pleines sont le résultat d'un filtrage de hautes fréquences de Savitzky-Golay. 99

5.7 Évolution de l'erreur de reconstruction évaluée sur des mini-lots de l'ensemble de validation, en échelle logarithmique. L'erreur a été évaluée à chaque 128 itérations de gradient. 100

5.8 Configuration atomique générée par le WAE d'énergie -259.28 eV. Le code de couleur des liens décrit une échelle allant de 2Å à 2.95Å. On remarque un désordre prononcé dans la distribution des longueurs de liaison, mais aucun défaut structurel majeur. La figure a été conçue à l'aide de OVITO [52]. 101

5.9 Fonction de distribution radiale (FDR) d'un échantillon généré par le WAE après 10 époques. La FDR est normalisée par la densité de l'échantillon (5.00e-2 atomes/Å^3). 103

5.10 Distribution des angles de liaison entre premiers voisins et ajustement Gaussien de celle-ci. La valeur moyenne et l'écart-type de l'ajustement sont comparables à celles de la distribution i.e. 108.42° et 28.59° respectivement. 104

LISTE DES ANNEXES

Annexe I: Algorithme du gradient avec inertie et ADAM . . . . . **xxii**

Annexe II: Algorithmes du GAN, WGAN, VAE et WAE . . . . . **xxvi**

1 GAN. xxvi

2 WGAN. xxvii

3 VAE. xxviii

4 WAE. xxix

LISTE DES ABRÉVIATIONS

a-Si	Silicium amorphe
AA	Apprentissage automatique
ADAM	*Adaptive moment estimate*
AG	Algorithme du gradient
AGS	Algorithme du gradient stochastique
ART	Technique d'activation et de relaxation
c-Si	Silicium cristallin
CFP	Conditions aux frontières périodiques
CNN	Réseau neuronal convolutif
DFT	Théorie de la fonctionnelle de la densité
FDR	Fonction de distribution radiale
GAN	Réseau accusatoire génératif
MAP	Maximum a posteriori
MD	Dynamique moléculaire
MLP	Perceptron multicouche
RAC	Réseau aléatoire continu
ReLU	Unité de rectification linéaire
RMSE	Erreur quadratique moyenne
RMC	Monte Carlo inverse
RN	Réseau de neurones

SEP	Surface d'énergie potentielle
SW	Stillinger-Weber
VAE	Auto-Encodeur variationnel
WAE	Auto-Encodeur de Wasserstein
WGAN	Réseau accusatoire génératif de Wasserstein
WWW	Wooten Winer et Weaire

NOTATION

$\mathbb{R}$	Ensemble des nombres réels
$\mathbb{C}$	Ensemble des nombres complexes
$\mathbb{Z}$	Ensemble des nombres entiers relatifs
$\mathbb{E}$	Espérance mathématique
$\mathbb{B}$	Biais d'un estimateur statistique
$\mathbb{V}$	Variance mathématique
$\mathbb{D}$	Ensemble de données
$\mathcal{X}$	Espace topologique
$\mathbf{X}$	Ensemble de tenseurs
$\mathbf{x}$	Tenseur
$p_{\mathcal{X}}$	Distribution de probabilité sur $\mathcal{X}$
$p(\mathbf{x})$	Densité de probabilité de $\mathbf{x}$
$P(\mathbf{x})$	Fonction de distribution cumulative de $\mathbf{x}$
f_{ω}	Fonction paramétrée par ω
F_{ω}	Distribution des valeurs de f_{ω}

À Ondine.

REMERCIEMENTS

J'aimerais premièrement remercier mon directeur de recherche Laurent J. Lewis pour m'avoir donné l'opportunité de m'épanouir sur un projet audacieux et motivant et sans qui ce travail n'aurait jamais vu le jour. Je souhaite aussi remercier Aaron Courville pour ses conseils et ses suggestions dans un domaine qui ne m'était pas familier. Merci aussi à Normand Mousseau, Mickaël Trochet et Simon Gelin pour la disponibilité des données et des codes dont j'ai fait usage ainsi que pour l'aide qu'ils m'ont offerte pour l'utilisation de LAMMPS et de ART.

J'aimerais remercier Amanda, Jeremi, Marie-Maude, Simon, et mes collègues de bureau Sergueï et Eric pour les agréables dîners et bons moments passés qui ont contribué à la belle ambiance du département.

Je remercie Vlad pour les innombrables cafés et paquets de cigarettes partagés ou volés. Merci à Anas et Etienne pour leur sincérité autant dans notre amitié que dans notre passion pour la musique.

Je remercie chaleureusement Alessandro, Charlotte, Cécile, Gabrielle, Martin, Mélanie, Omar et Yasmine pour leur amitié et leur soutien dans toutes les circonstances qui ont accompagné ces deux dernières années.

Finalement, je remercie infiniment mes parents, Letizia et Fabio, ainsi que mes frères Niccolò et Andrea pour leur amour et leur soutien intemporel dans tout ce que j'ai entrepris et tout ce que je souhaite faire et devenir.

INTRODUCTION

L'absence d'ordre à longue portée dans les matériaux amorphes empêche leur structure atomique d'être univoquement déterminée à partir des expériences de diffraction. Il est donc nécessaire de reconstruire cette dernière à partir de simulations numériques, dont la fiabilité peut être subséquemment évaluée par comparaison directe des propriétés calculées à partir du modèle avec les mesures expérimentales.

Le système canonique pour l'étude des systèmes désordonnés est le silicium amorphe, dont la structure atomique est considérée comme étant celle d'un réseau aléatoire continu avec une coordination proche de quatre, une faible variance dans les longueurs de liens chimiques et un désordre plus conséquent dans les angles de liaison. Il est possible de modéliser sa structure atomique par des simulations de dynamique moléculaire, des approches Monte Carlo basées sur la permutation de liens ou l'exploration de la surface d'énergie configurationnelle [3, 4, 40, 63]. La qualité de ces algorithmes dépend fortement du niveau de théorie utilisé dans la description énergétique du système. Bien que les calculs *ab initio* fournissent une description précise des interactions microscopiques, et donc de la structure atomique des solides, leur coût computationnel prohibitif n'en limite l'application qu'à de petits systèmes. Les simulations numériques de grande échelle sont pourtant nécessaires pour des applications réelles, et bien que des potentiels empiriques aient été développés à cet effet, leur performance ne se démarque qu'au prix de leur fiabilité. Ainsi, la recherche d'alternatives à la fois efficaces et réalistes à ces approches est encore aujourd'hui un domaine de recherche ouvert et actif.

La complexité de la description énergétique des systèmes condensés a pour origine l'augmentation exponentielle de la dimension de l'espace des états avec le nombre de particules. Ce phénomène, également connu dans le domaine de l'apprentissage automatique sous le nom de fléau de la dimension, a cependant pu être contourné dans les avancées récentes en apprentissage profond. Les modèles résultants sont capables de traiter efficacement des problèmes de haute dimension à l'aide de réseaux de neurones profonds, qui sont des approximateurs de fonctions universels. Ces derniers permettent

notamment d'approximer efficacement le paysage énergétique d'un système atomique lorsqu'entraînés sur un ensemble d'observations, pouvant être générées avec le niveau de théorie souhaité. De ce fait, ils constituent déjà des alternatives intéressantes aux potentiels empiriques [6–8].

En outre, les modèles génératifs profonds permettent de caractériser et d'approximer la distribution d'ensembles de données complexes et de générer efficacement de nouvelles réalisations de celle-ci. Leur performance inégalée sur des tâches génératives variées ouvre la voie à leur application pour la modélisation de la structure atomique des matériaux, une tâche d'apprentissage non supervisé encore inexplorée. La qualité des échantillons générés par ces modèles dépend majoritairement de la qualité de l'ensemble de données, ce qui signifie que la partie la plus coûteuse en performances est la formation de l'ensemble d'entraînement et l'entraînement du modèle lui-même. À terme, l'algorithme obtenu est très performant et peut générer des nouvelles configurations atomiques efficacement.

Ainsi, l'enjeu de ce travail est d'établir une preuve de concept pour la performance et la qualité de modélisation de la structure atomique du silicium amorphe à l'aide des techniques d'apprentissage profond. Après une introduction à la problématique de la modélisation du silicium amorphe (Chapitre 1), une exposition des algorithmes d'apprentissage (Chapitre 2) et d'apprentissage profond spécifiquement (Chapitre 3) décrit le vaste choix de modèles utilisables. Ensuite, une discussion des applications antérieures de l'apprentissage profond à l'approximation du paysage énergétique du silicium amorphe (Chapitre 4) met en lumière les stratégies nécessaires au succès de cette approche. Finalement, une présentation des modèles génératifs développés et de leur performance confirme la viabilité de l'approche et souligne les défis pouvant entraver sa performance (Chapitre 5).

CHAPITRE 1

PROPRIÉTÉS ET MODÉLISATION DU SILICIUM AMORPHE

1.1 Motivation

Les solides cristallins ont été largement étudiés lors de la naissance de la cristallographie géométrique qui décrit leur configuration microscopique comme des structures régulières idéales. Cette simplification a permis d'étudier en profondeur les propriétés émergentes des cristaux à l'aide d'outils théoriques aujourd'hui bien connus et maîtrisés. Cependant, les matériaux réels se distinguent inévitablement de ces idéalisations, ce qui a donné naissance aux théories des défauts cristallins et des matériaux désordonnés [15, 42]. L'absence d'ordre à longue portée des solides amorphes présente un grand défi à la fois théorique et expérimental pour leur étude : beaucoup d'aspects de leur structure et de ses liens avec leurs propriétés sont encore mal compris aujourd'hui [65].

Le silicium est un élément très abondant dans la croûte terrestre, il a été communément utilisé (sous la forme de silicates) depuis l'Antiquité à des fins architecturales et décoratives. Il s'agit aussi d'un semi-conducteur intrinsèque efficace et de faible coût, ce qui en a fait un élément crucial dans le développement des technologies de l'information et de l'énergie solaire. Son étude ainsi motivée par des besoins à la fois technologiques et théoriques est devenue une référence pour le développement et la validation des modèles de calcul.

1.2 Propriétés thermodynamiques des solides amorphes

Le refroidissement d'un liquide, abaissant l'énergie cinétique moyenne des atomes, mène à une transition de phase structurelle vers un état solide. Les solides adoptent souvent une configuration spatiale périodique, le réseau cristallin, qui de par sa haute symétrie constitue le minimum global d'énergie. La transition vers l'état cristallin se

manifeste par une discontinuité, à la température de cristallisation T_c, dans le volume $V(T) = -\frac{\partial G}{\partial P}$ et l'entropie $S(T) = -\frac{\partial F}{\partial T}$ en fonction de la température T. Selon la classification de Ehrenfest, qui qualifie une transition de phase d'ordre n lorsqu'elle exhibe une discontinuité dans la n-ième dérivée de l'énergie libre thermodynamique, il s'agit d'une transition de premier ordre.

Cependant, la transition cristalline peut être contournée si la vitesse de refroidissement $\dot{T} = \frac{dT}{dt}$ est assez grande. Le système adopte alors continûment, dans un intervalle restreint autour de la température de transition vitreuse $T_v < T_c$, une phase solide structurellement désordonnée qui se caractérise par l'absence de symétrie globale. Cette phase désordonnée est de plus haute énergie que la phase cristalline, mais reste cependant stable malgré l'absence d'ordre à longue portée dans sa structure. Comme le montre la figure 1.1 le volume, l'entropie, mais aussi la chaleur spécifique $C = T\frac{\partial S}{\partial T}$ et toutes les quantités reliées aux dérivées de l'énergie libre du liquide changent continûment vers la phase vitreuse : il ne s'agit donc pas à proprement parler d'une transition de phase thermodynamique entre états d'équilibre.

En effet, T_v augmente de manière approximativement logarithmique avec la vitesse de refroidissement $\dot{T}$, ce qui indique qu'il s'agit d'un phénomène cinétique. La raison provient de la forte dépendance en température du temps de relaxation structurel τ, qui est le temps caractéristique que met le système à adapter sa configuration atomique à un changement de température. Celui-ci est de l'ordre de 10^{-12} s à T_c, ce qui permet effectivement aux atomes de se conformer rapidement en une structure d'énergie minimale. À $T_v - 50^\circ$ K, $\tau \approx 10^{10}$ années (approximativement l'âge de l'univers) : le temps de réarrangement structurel dépasse alors toute échelle de temps accessible expérimentalement et les atomes sont *gelés* dans une configuration. Ainsi, la phase vitreuse se présente comme un état métastable cinétiquement bloqué où l'entropie et la densité dépendent de son histoire thermique [13, 65].

Néanmoins, malgré la grande importance des effets cinétiques sur la définition de T_v

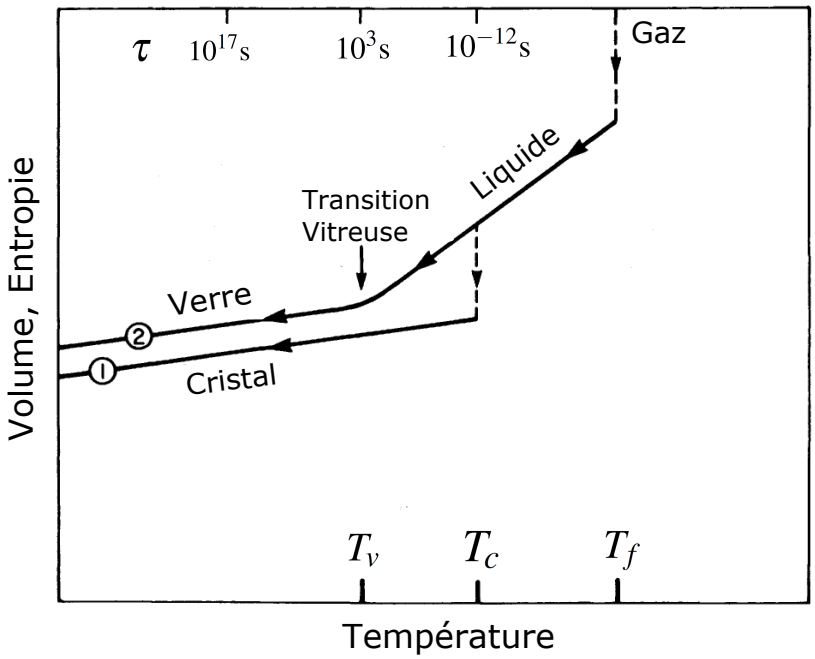

FIGURE 1.1 : Dépendance qualitative du volume et de l'entropie en fonction de la température pour différentes transitions de phase (modifié de [65]).

la transition vitreuse semble être la manifestation d'une transition de phase thermodynamique dans la limite où $\dot{T} \to 0$. La figure 1.2 présente la dépendance en température de l'entropie d'excès entre les phases liquide et cristalline $S_{ex} = S_l - S_v$. Comme la chaleur spécifique du liquide est bien supérieure à celle du cristal, S_{ex} diminue fortement entre T_c et T_v. En extrapolant cette tendance, on remarque que l'entropie d'excès s'annule à une température non nulle : $S_{ex}(T_0) = 0$. Ceci signifie qu'à des températures $T < T_0$, le liquide aurait en principe une entropie plus basse que le cristal. Ce paradoxe apparent, appelé le paradoxe de Kauzmann [26], n'est jamais observé puisqu'entre T_v et T_0 advient la transition vitreuse.

Ces considérations soulignent que T_0 forme une borne inférieure pour T_v lorsque $\dot{T} \to 0$, c'est-à-dire que l'existence des verres ne dépend pas de phénomènes cinétiques : ces derniers ne forment qu'une contrainte thermodynamique sur la valeur de T_0 [17].

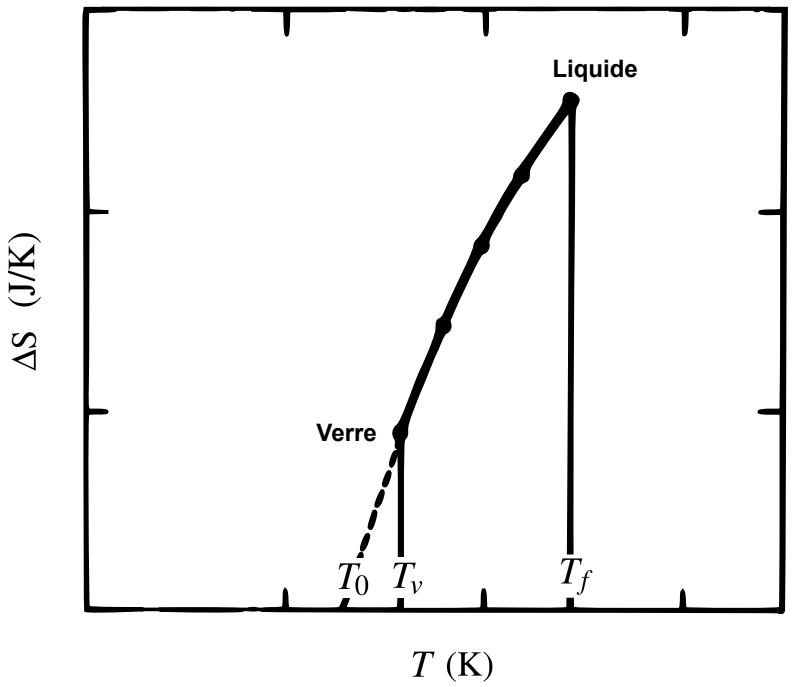

FIGURE 1.2 : Entropie d'excès entre les phases liquide et amorphe en fonction de la température (modifié de [65]).

Cependant, tous les solides amorphes ne subissent pas une transition vitreuse. Par exemple, lors de la transition vers la phase amorphe le silicium exhibe une discontinuité dans son nombre de coordination[1], passant de 6 pour le liquide à 4 pour l'amorphe, qui traduit une transition structurelle de premier ordre. Le silicium amorphe n'est donc pas un verre et il ne peut donc être fabriqué par des méthodes de refroidissement rapide du liquide.

1.3 Fabrication du silicium amorphe

Les méthodes de préparation des verres exploitent la dépendance de la température de vitrification avec le taux de refroidissement, la plus simple étant le refroidissement rapide de la phase liquide permettant de contourner la transition cristalline.

Le dioxyde de silicium SiO_2, communément appelé silice, peut se fabriquer avec

[1] Ou coordinance, le nombre de plus proches voisins moyen des atomes.

des taux de refroidissement de l'ordre $10^{-4} - 10^{-1}$ K/s qui sont aisément atteignables en laissant la silice liquide se refroidir naturellement. Certains verres métalliques demandent cependant des taux plus élevés de l'ordre de $10^2 K/s$ qui peuvent être atteints par la méthode de trempe brusque (*splat-quenching*).

La phase amorphe du silicium pur ne peut être obtenue par refroidissement rapide car le taux de refroidissement nécessaire serait inaccessible. La technique de condensation solide (*quench from vapor*) permet d'obtenir la phase amorphe directement à partir de la phase gazeuse en déposant la vapeur de silicium sur un substrat froid, formant alors une couche mince. D'autres techniques se basent sur la vaporisation par un faisceau d'électrons ou la décomposition par plasma d'un composé de silicium. La technique la plus fiable en termes de pureté et de qualité reste cependant le bombardement ionique sur un cristal de silicium pur à l'aide d'ions énergétiques ($\simeq$ MeV). La couche de a-Si ainsi obtenue présente peu de poches vides et aucune impureté, et peut s'étendre sur une épaisseur de plusieurs micromètres [31]. Dans le but d'obtenir des structures plus relaxées, il est commun d'effectuer un recuit : il s'agit d'élever graduellement la température de l'échantillon puis de lui faire subir un refroidissement contrôlé ce qui permet aux défauts éventuels de se conformer localement en un nouvel équilibre.

1.4 Propriétés structurelles du silicium amorphe

Le silicium est un métalloïde covalent possédant 10 électrons de cœur et 4 électrons de valence selon la configuration atomique [Ne]$3s^2 3p^2$. Il est un des rares éléments ayant une densité plus faible à l'état solide qu'à l'état liquide. Dans sa phase cristalline il se présente sous la structure diamant, représentée à la figure 1.3, c'est-à-dire deux réseaux cubiques à faces centrées superposés à un quart de diagonale l'un par rapport à l'autre. Les électrons y forment alors des orbitales hybrides sp^3, qui sont des combinaisons linéaires d'une orbitale s et trois orbitales p, qui favorisent donc la structure tétraédrique du réseau avec des angles de liaison tous égaux à $\arccos(-\frac{1}{3}) = 109.47°$.

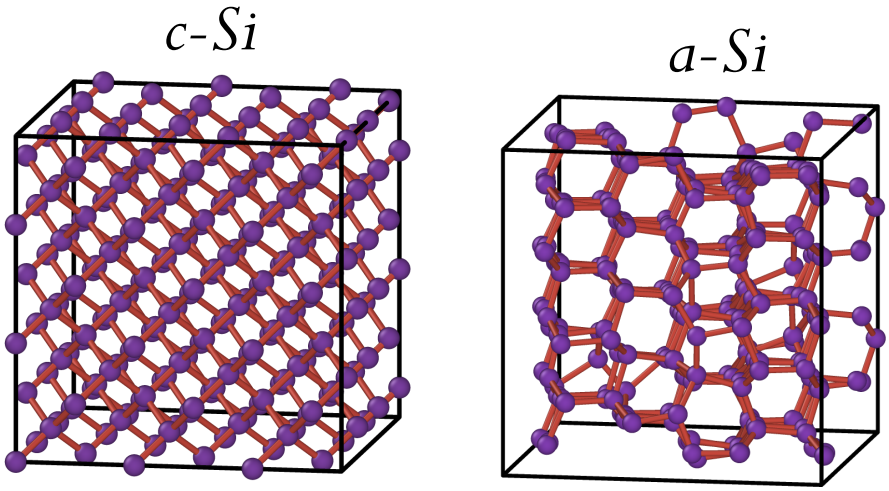

FIGURE 1.3 : Configuration atomique du silicium cristallin (c-Si) *(gauche)* et du silicium amorphe (a-Si) *(droite)* générée par la méthode décrite en [4].

La structure atomique est donc formée de tétraèdres parfaitement coordonnés et l'angle entre deux tétraèdres, appelé l'angle dièdre, vaut $\phi = 60^\circ$ pour le réseau diamant. De plus, chaque atome est parcouru par douze chemins fermés distincts ayant un nombre minimal de liens, appelés anneaux irréductibles, tous composés de six atomes. Tous les anneaux possèdent un nombre pair de liens. Ces propriétés topologiques donnent lieu à une conformation de type *chaise* illustrée à la figure 1.4 [65].

Le cristal possède une symétrie de translation le long des axes cristallins ce qui facilite grandement son étude théorique. Il devient possible de décrire la structure de bandes dans le réseau réciproque, qui est la transformée de Fourier du réseau cristallin, par l'application du théorème de Bloch, et le vecteur d'onde représente un bon nombre quantique pour identifier les états d'énergie électroniques.

De plus, il est possible de reconstituer directement la structure cristalline à partir des expériences de diffraction par rayons X, neutrons ou électrons. En mesurant l'intensité de la diffusion d'un faisceau monochromatique de longueur d'onde λ en fonction de l'angle de diffusion 2θ, reliés au vecteur d'onde par $Q = \frac{4\pi}{\lambda}\sin(\theta)$, on obtient le facteur de structure statique $S(\mathbf{Q})$. Celui-ci est une caractérisation unidimensionnelle des propriétés structurales de l'échantillon, qui, couplé à la symétrie du réseau cristallin, permet

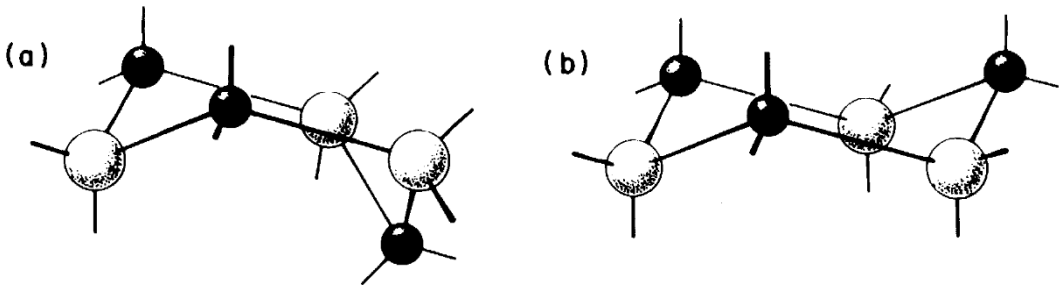

FIGURE 1.4 : *a)* Conformation de type *bateau* et *b)* conformation de type *chaise* pour des anneaux à six atomes [65].

alors de déduire complètement la structure atomique.

Si on considère un système de N particules de coordonnées $\{\mathbf{r}_i\}_{i\in[1,...N]}$ occupant un volume V à une température T, c'est-à-dire dans l'ensemble canonique, la densité de probabilité que chaque particule i soit dans un élément de volume $\mathrm{d}\mathbf{r}_i$ centré en $\mathbf{r}_i$ s'écrit

$$p_N(\mathbf{r}_1,\ldots,\mathbf{r}_N)\mathrm{d}\mathbf{r}_1\ldots\mathrm{d}\mathbf{r}_N\,. \tag{1.1}$$

Clairement, p_N peut être calculée à partir de la fonction de partition canonique en marginalisant (intégrant sur) toutes les impulsions ce qui donne

$$p_N(\mathbf{r}_1,\ldots,\mathbf{r}_N)=\frac{e^{-\beta U(\mathbf{r}_1,\ldots,\mathbf{r}_N)}}{Z_N(V,T)}\,, \tag{1.2}$$

où $Z_N(V,T)=\int p_N(\mathbf{r}_1,\ldots,\mathbf{r}_N)\mathrm{d}\mathbf{r}_1\ldots\mathrm{d}\mathbf{r}_N$. Il est alors commun de définir des densités de probabilité partielles

$$p_N^n(\mathbf{r}_1,\ldots,\mathbf{r}_N)=\int \mathrm{d}\mathbf{r}_{n+1}\ldots\int \mathrm{d}\mathbf{r}_N\; p_N(\mathbf{r}_1,\ldots,\mathbf{r}_N)\,, \tag{1.3}$$

qui représentent la densité de probabilité de trouver chaque particule i en $\mathbf{r}_i$ pour les particules $i\in[1,...,n]$ pour toute configuration de positions des particules $n+1,...,N$. Puisque les particules sont indistinguables, il y a $N!/(N-n)!$ manières d'arranger n particules dans les n positions pour N particules. C'est ainsi que sont définies les fonctions

de distribution $\rho_N^{(n)}$

$$\rho_N^{(n)}(\mathbf{r}_1,\ldots,\mathbf{r}_N) = \frac{N!}{(N-n)!} p_N^n(\mathbf{r}_1,\ldots,\mathbf{r}_N)\ . \tag{1.4}$$

Ces fonctions de distribution, qui ne sont pas des densités de probabilité puisque leur normalisation est $N!/(N-n)!$, ont un sens physique précis. La distribution $\rho_N^{(1)}(\mathbf{r})$ est simplement la densité locale $\rho(\mathbf{r})$. Puisque $\int \rho_N^{(1)}(\mathbf{r})\mathrm{d}\mathbf{r} = N$, si la configuration est homogène (comme c'est le cas dans certains fluides) $\rho_N^{(1)}(\mathbf{r}) = \frac{N}{V}$ est constant et égal à la densité moyenne de particules.
La distribution $\rho_N^{(2)}(\mathbf{r}_1,\mathbf{r_2})$ ne dépend que de la distance relative des particules $r_{12} = \|\mathbf{r}_1 - \mathbf{r}_2\|$ car il n'y a pas de direction ni d'origine préférentielle dans l'espace. On définit alors $\rho_N^{(2)}(r) = \rho^2 G(r)$, où $\rho = \frac{N}{V}$ et $G(r)$ est la **fonction de distribution radiale**[2] (FDR) ou fonction de corrélation de paires. Elle représente la densité moyenne de particules dans une coquille sphérique à une distance r d'une particules centrale quelconque, moyennée sur toutes les particules centrales [16].

La FDR permet de caractériser l'ordre local d'une configuration atomique, et peut être calculée à partir du facteur de structure $S(\mathbf{Q})$ essentiellement par le biais d'une transformée de Fourier car

$$S(\mathbf{Q}) = 1 + \rho \int \mathrm{d}\mathbf{r}\ G(r) e^{i\mathbf{Q}\cdot\mathbf{r}}\ . \tag{1.5}$$

La FDR intervient aussi dans les équations thermodynamiques de différentes quantités mécaniques comme l'énergie ou dans l'équation du viriel pour la pression

$$E = \frac{3}{2}NkT + 2\pi\rho N \int_0^\infty \mathrm{d}r\, r^2 G(r)\phi(r) \tag{1.6}$$

$$p = \rho kT - \frac{2\pi}{3}\rho^2 \int_0^\infty \mathrm{d}r\, r^3 G(r)\phi(r)\ , \tag{1.7}$$

où $\phi(r)$ est le potentiel d'interaction. Celui-ci peut en principe aussi être calculé à partir

[2] $G(r)$ est en fait appelé la fonction de distribution radiale *réduite*. Elle est reliée à la FDR $J(r)$ par $J(r) = rG(r) + 4\pi r^2\rho$.

de $G(r)$ bien qu'il ne s'agisse pas d'une tâche triviale.

La FDR d'un échantillon cristallin, présentée à la figure 1.5, est composée de pics étroits clairement séparés reflétant la présence de couches successives d'atomes parfaitement ordonnées. Puisque ces couches sont nettement séparées, la définition des $n^{ièmes}$ voisins n'est pas ambiguë. Il est alors possible de reconstruire la structure cristalline de manière complète et univoque en considérant les symétries du cristal.

Au contraire, le fort désordre structurel présent dans la structure atomique du silicium amorphe (figure 1.3) se traduit par un chevauchement significatif des couches successives d'atomes (figure 1.5). Si on modélise les pics de la FDR par des fonctions Gaussiennes, il est possible de décomposer leur variance en un facteur de désordre thermique et un facteur de désordre statique : $\sigma^2 = \sigma_T^2 + \sigma_D^2$, ce dernier étant nul pour le cristal [65].

Ainsi la présence d'un grand facteur de désordre statique pour les couches de voisins au delà de la première rend la définition de second ou de troisième voisin caduque dans le cas du a-Si. L'absence de symétrie de translation ne permet plus de déduire univoquement la structure microscopique à partir de la FDR, et empêche également l'application du théorème de Bloch pour le calcul des propriétés électroniques : la description des propriétés structurelles dans l'espace réciproque, commune dans le cas cristallin, perd entièrement son sens puisque le vecteur d'onde n'est plus un bon nombre quantique.

Seuls les premiers pics de la FDR sont bien définis, reflétant un ordre local dominant, tandis qu'à des plus grandes distances inter-atomiques la FDR tend vers 1, c'est-à-dire que la densité $\rho(\mathbf{r})$ tend vers densité moyenne de l'échantillon ρ, ce qui explicite l'isotropie du matériau.

La topologie du silicium amorphe peut donc être partiellement décrite à partir des statistiques fournies par la FDR expérimentale. Son analyse se fait en adaptant au mieux chaque pic par une fonction Gaussienne de manière à pouvoir en extraire des informations structurelles. La position, l'écart-type et l'intégrale de chaque pic permettent de cal-

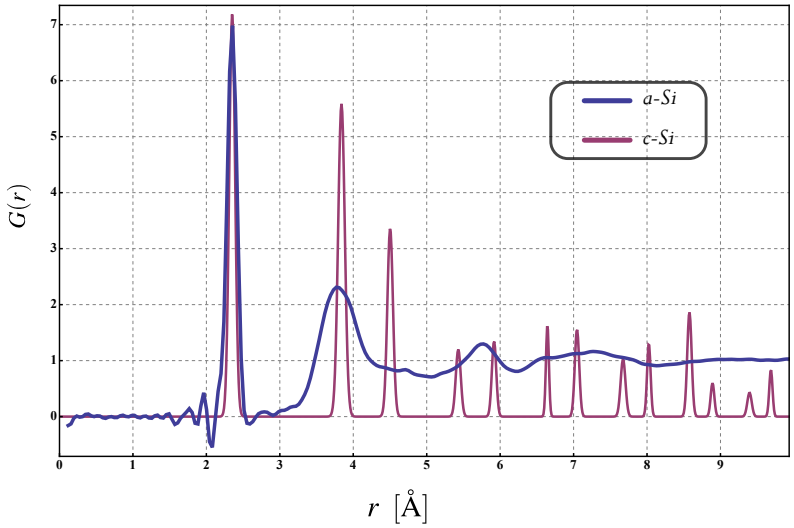

FIGURE 1.5 : Fonction de distribution radiale expérimentale d'un échantillon de silicium amorphe pur après un recuit à 600° C [31] comparée à celle d'un cristal (modélisation numérique). La structure cristalline peut être reconstituée à partir des positions des couches successives d'atomes qui sont bien définies, contrairement au cas du silicium amorphe.

culer la distribution des distances inter-atomiques, la distribution des angles de liaison ou encore le nombre de coordination. Le tableau 1.I compare les propriétés structurelles issues de la FDR pour un échantillon cristallin, amorphe recuit et amorphe tel-qu'implanté [31].

	ρ (g/cm^3)	C_1 (at.)	r_1 (Å)	σ_1 (Å)	C_2 (at.)	r_2 (Å)	σ_2 (Å)
Cristal	2.329	4.02	2.356	0.057	12.15	3.841	0.066
Recuit	2.285	3.88	2.352	0.065	12.43	3.810	0.238
Tel-qu'implanté	2.285	3.79	2.351	0.064	12.15	3.808	0.257

TABLEAU 1.I : Propriétés structurelles expérimentales d'échantillons de silicium cristallin et amorphe : densité (ρ), coordination (C), position des pics (r) et déviation standard (σ) des deux premiers pics de la FDR [31].

La position du premier pic correspond à la distance moyenne entre deux atomes voisins, qui reste inchangée par rapport à la valeur de 2.35Å du silicium cristallin, en considérant les marges d'erreur expérimentales. L'écart-type de la distance inter-atomique

de $\sigma_r = 0.065$Å ce qui correspond à un écart relatif $\left(\frac{\sigma_r}{r_0}\right)$ d'uniquement 2.76% ; la distribution de longueur des liens chimiques est essentiellement égale à celle du silicium cristallin, dans l'échantillon tel-qu'implanté et recuit. L'intégrale du premier pic correspond à la coordination moyenne, qui vaut $z = 3.79$ pour l'échantillon tel-qu'implanté et $z = 3.88$ après recuit.

La position du deuxième pic détermine la distance moyenne des seconds ou troisièmes voisins et permet donc d'obtenir la distribution des angles de liaison par la formule $\cos\theta = \frac{\mathbf{r}_1 \cdot \mathbf{r}_2}{r_1 r_2}$. La distribution des seconds voisins, centrée en $r_2 = 3.8$ Å, présente un plus grand désordre avec un un écart-type de $\sigma_\theta = 0.240$Å ce qui correspond à un angle de liaison de $\theta = 107.83^\circ$ avec $\sigma_\theta = 10.45 \pm 0.09^\circ$ (tel qu'implanté) et $\sigma_\theta = 9.63 \pm 0.08^\circ$ (après recuit).

La position du troisième pic est significativement élargie et ne ressort que très peu du fond continu issu du chevauchement des coquilles de voisins, indiquant la présence d'un désordre considérable dans la distribution de l'angle dièdre. Celui-ci est distribué selon un large spectre de valeurs centré sur 60°, et il y a présence d'anneaux composés d'un nombre pair et impair de liens. Les statistiques d'anneaux du silicium amorphe restent peu connues expérimentalement du fait de la difficulté de leur extraction à partir de la FDR ; de plus les propriétés structurales mentionnées ici sont inévitablement dépendantes des procédures d'ajustement des Gaussiennes sur les pics de la FDR expérimentale [31].

L'impossibilité de décrire complètement la structure microscopique du silicium amorphe à partir de la FDR requiert le développement de modèles numériques, dont l'efficacité est évaluée en comparant les propriétés mécaniques, thermodynamiques et structurelles que l'on en déduit avec les valeurs expérimentales.

1.5 Modélisation de la structure atomique

Le premier type de modèle pour la structure atomique des solides désordonnés covalents est celui du réseau aléatoire continu (RAC), introduit en 1932 par W.H. Zacharia-

sen [64], qui est encore le modèle de choix aujourd'hui du fait de son bon accord avec les données expérimentales, contrairement aux modèles micro-cristallins, pour les matériaux de la colonne IV du tableau périodique et pour les composés III-V. Un RAC est formellement défini comme un ensemble de coordonnées atomiques et une liste de liens entre les atomes, il est dit idéal si la coordination est exactement de quatre et si les distances inter-atomiques sont fixes. En pratique, les RAC pour le silicium amorphe ne sont pas idéaux puisque les données structurelles issues de la FDR expérimentale suggèrent au contraire une coordination légèrement inférieure à quatre et une faible variance dans la longueur des liens chimiques.

Il est a priori possible de générer des RAC à partir de principes premiers en résolvant numériquement l'équation de Schrödinger en prenant en compte toutes les interactions de Coulomb entre les N atomes considérés. Les simulations de dynamique moléculaire (MD) *ab initio* calculent les énergies et les forces du système à chaque pas de temps discret selon des calculs de structure électronique, généralement issus de la théorie de la fonctionnelle de la densité (DFT). Bien que ces simulations soient réalistes, leur coût computationnel est très prohibitif : malgré la disponibilité d'implémentations efficaces et d'ordinateurs très performants, les applications sont souvent limitées à quelques centaines d'atomes et à des temps de simulation de quelques picosecondes.

Une alternative efficace consiste alors à utiliser des potentiels empiriques, c'est-à-dire à abandonner la description quantique du potentiel au profit d'une interaction effective entre les atomes. Il s'agit de fonctions émulant l'énergie potentielle du système à partir de formes fonctionnelles paramétrées, construites sur la base de considérations physiques, qui sont ajustées pour reproduire une vaste gamme de propriétés physiques observées expérimentalement pour un certain nombre de systèmes canoniques. Malgré leur degré d'explication réduit par rapport aux calculs *ab initio*, les potentiels empiriques ont fait preuve d'un réalisme satisfaisant et ont rendu possibles des simulations à grande échelle.

Un potentiel empirique souvent utilisé pour la modélisation de l'énergie potentielle

du silicium amorphe, et qui sera utilisé dans ce travail, est le potentiel de Stillinger-Weber (SW) [51, 61]. L'énergie potentielle est développée comme la somme d'énergies atomiques individuelles, et le potentiel est composé d'un terme d'interaction à deux corps et d'un terme d'interaction à trois corps.

$$E = \varepsilon \sum_i E_i = \varepsilon \sum_i \left[\sum_{j>i} \phi_2(r_{ij}) + \sum_{\substack{j\neq i \\ k>j}} \phi_3(r_{ij}, r_{ik}) \right], \quad r_{ij}, r_{ik} < \sigma a \tag{1.8a}$$

$$\phi_2(r) = A\left(B\left(\frac{r}{\sigma}\right)^{-p} - 1\right) \exp\left(\frac{1}{\frac{r}{\sigma} - a}\right) \tag{1.8b}$$

$$\phi_3(r_{ij}, r_{ik}) = \lambda\left(\cos\theta_{ijk} + \frac{1}{3}\right)^2 \exp\left(\frac{\gamma}{\frac{r_{ij}}{\sigma} - a}\right) \exp\left(\frac{\gamma}{\frac{r_{ik}}{\sigma} - a}\right). \tag{1.8c}$$

Le vecteur $\mathbf{r}_i$ est le vecteur position de l'atome i, le vecteur $\mathbf{r}_{ij}$ désigne le vecteur déplacement $\mathbf{r}_j - \mathbf{r}_i$ et l'angle θ_{ijk} désigne l'angle entre les vecteurs $\mathbf{r}_{ik}$ et $\mathbf{r}_{ik}$. Les contributions des différents voisins sont prises en compte jusqu'à un rayon de coupure $r_c = \sigma a$, et on remarque que la contribution du terme à trois corps ϕ_3 s'annule lorsque $\cos\theta = -\frac{1}{3}$, c'est-à-dire l'angle de liaison dans le cristal. Les paramètres $\varepsilon, A, B, \sigma, a, \gamma, \lambda, p$ sont ajustés et fixés pour la description de matériaux spécifiques. En particulier ε définit l'échelle globale d'énergie tandis que A et λ définissent l'échelle relative des termes ϕ_2 et ϕ_3. Il existe plusieurs paramétrisations de ce potentiel pour le silicium, dont une particulièrement adaptée au silicium amorphe, qui consiste à augmenter de 50% la contribution du terme à trois corps, ce qui résulte en de meilleures propriétés structurelles [61]. Le potentiel SW reste relativement efficace à évaluer et peut être utilisé dans des simulations de dynamique moléculaire, bien que d'autres algorithmes existent pour la modélisation de la structure du silicium amorphe.

L'algorithme de Wooten, Winer et Weaire (WWW) [63] est une approche efficace permettant de générer des RAC aux propriétés compatibles avec les expériences. La méthode débute avec une cellule cristalline de N atomes de silicium dans la configuration diamant avec des conditions aux frontières périodiques (CFP). La structure est donc décrite par $3N$ coordonnées et $2N$ liens et la méthode consiste à désordonner cette structure

en appliquant successivement des transpositions de liens tel qu'illustré à la figure 1.6.

La variation d'énergie due à une transposition doit être calculée à l'aide d'un potentiel harmonique avec une liste de voisins fixée, et l'implémentation originale de l'algorithme pour le silicium amorphe utilise le potentiel de Keating [27] défini par

$$E = \sum_{i<j} \frac{3\alpha}{8d^2}(r_{ij}^2 - d^2)^2 + \sum_{\substack{i\neq j,k \\ j<k}} \frac{3\beta}{8d^2}(\mathbf{r}_{ij}\cdot\mathbf{r}_{ik} + \frac{1}{3}d^2)^2 \,, \tag{1.9}$$

où les sommes sont effectuées sur les atomes i et leurs premiers voisins j,k dont les vecteurs de séparation sont $\mathbf{r}_{ij},\mathbf{r}_{ik}$, le paramètre d est la distance d'équilibre entre les atomes et les paramètres α,β contrôlent l'impact de l'élongation des liens et de leur déformation respectivement.

La méthode définit ensuite une température fictive T, et chaque transposition est acceptée ou rejetée selon un critère de Métropolis, c'est-à-dire avec une probabilité d'acceptation $p(\Delta\varepsilon) = \min[1, e^{-\frac{\Delta\varepsilon}{k_b T}}]$.

La température est diminuée au cours du processus et la structure est régulièrement refroidie à 0 °K de manière à simuler la relaxation structurelle du matériau. Le modèle initial de 216 atomes fait par Wooten, Winer et Weaire en 1984 présente une densité de

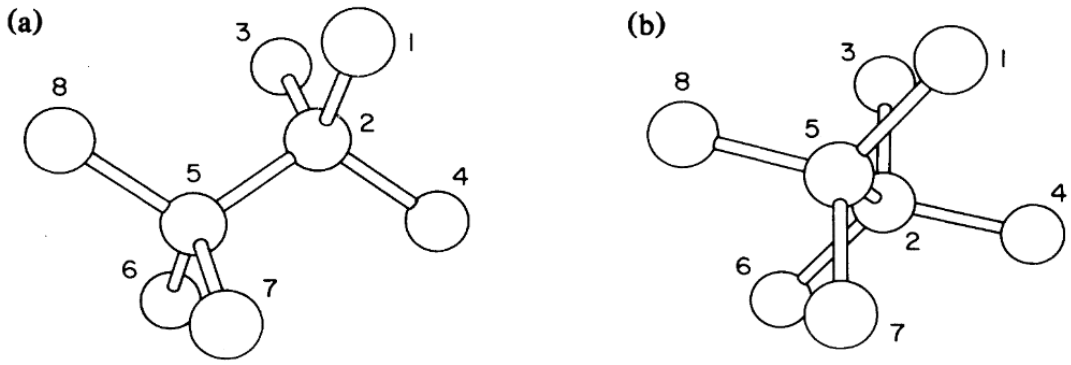

FIGURE 1.6 : Illustration d'une transposition de liens dans la méthode WWW [63].

1.04 fois celle du cristal, un écart-type angulaire de $\sigma_\theta = 10.9^\circ$ et un écart relatif de longueur des liens de $\Delta r = 2.7\%$.

Plus récemment, dans le but de générer des structures plus grandes et plus relaxées, G. T. Barkema et N. Mousseau [4] ont amélioré cette méthode en optimisant grandement les étapes les plus coûteuses en temps de calcul : les étapes de relaxation ont été écourtées et une alternative plus efficace du critère de Métropolis est utilisée. De plus, le point de départ n'est plus une structure diamant, mais une structure désordonnée de même densité que le cristal, avec une coordination de $z = 4$ exactement, et le potentiel de Stillinger-Weber, permettant une description plus réaliste des interactions, est utilisé lors de la minimisation finale.

Une autre méthode de modélisation pour la structure atomique du silicium amorphe est la technique d'Activation et de Relaxation (ART) [3, 35, 40]. Il s'agit d'une méthode générique pour explorer la surface d'énergie potentielle en recherchant des chemins de transition énergétique de manière itérative. Le système n'étant plus sujet à une évolution temporelle discrète requérant le calcul répété des forces et des énergies du système, les structures d'équilibre peuvent être générées efficacement même pour de grands systèmes. En partant d'une configuration initialement dans un minimum local, un événement de l'algorithme consiste à rechercher un point de selle (activation) pour ensuite converger vers un nouveau minimum local (relaxation).

L'étape d'activation se fait en suivant le vecteur propre correspondant à la plus grande valeur propre négative du Hessien de l'énergie, ce qui assure la convergence vers un point de selle. La diagonalisation complète de la matrice Hessienne étant coûteuse, la plus grande valeur propre est calculée avec des méthodes de puissance itérée comme l'algorithme de Lanczos. La configuration est donc poussée le long ce vecteur propre tout en minimisant les forces dans toutes les autres directions. Tant que la plus petite valeur propre ne change pas de signe la convergence est assurée, le cas échéant un nouvel événement est lancé.

L'étape de relaxation peut être achevée par n'importe quel algorithme du gradient

et toute fonction d'énergie potentielle peut être utilisée. Typiquement, l'algorithme des gradients conjugués ainsi que le potentiel de Stillinger-Weber sont utilisés.

Bien que les algorithmes exposés ici soient efficaces et couramment utilisés pour la modélisation de la structure du silicium amorphe, ils souffrent inévitablement du compromis entre performance et réalisme provenant du coût inhérent à une bonne description des interactions atomiques. La recherche de nouvelles approches contournant ce compromis est donc bénéfique au développement de modèles de plus en plus réalistes pour des grands systèmes.

1.6 Approches automatiques pour la modélisation de la structure atomique

Les modèles numériques de RAC présentés jusqu'ici voient leur réalisme dépendre de manière décisive du niveau de théorie mis en jeu dans la description des interactions atomiques. Une telle description, pour être rigoureuse, doit nécessairement être quantique mais reste impraticable pour de grands systèmes. Bien que les potentiels empiriques se soient démarqués par leur efficacité quant à engendrer des RAC en accord avec les expériences, des approches automatiques ne nécessitant pas de description énergétique du système, et ne souffrant donc pas de l'obstacle computationnel qui en découle, ont émergé.

Les approches de Monte Carlo inverse (RMC, *Reverse Monte Carlo*) permettent de générer des configurations minimisant directement, par construction, le désaccord avec un ensemble d'observations expérimentales. En partant d'une configuration initiale appropriée, un atome est déplacé aléatoirement et les propriétés de celle-ci sont recalculées. Le désaccord avec les données expérimentales est quantifié par la fonction de coût $\chi = \sum(y - y_{\text{exp}})^2/\sigma^2$ où y et y_{exp} sont les propriétés calculées et expérimentales et σ quantifie la précision de la mesure. L'évènement est alors accepté selon un critère de Metropolis i.e. avec une probabilité de $\min(1, e^{-\Delta\chi^2/2})$. Pour la modélisation du a-Si

il est naturel de minimiser le désaccord avec la FDR expérimentale, mais un problème important de cette méthode réside dans le fait qu'il y a en général plusieurs configurations possibles donnant lieu au même accord avec la FDR et la méthode ne fournit aucun moyen de déterminer laquelle est la plus réaliste. Il est nécessaire d'ajouter des contraintes *ad hoc* à la fonction de coût pour obtenir des configurations qui s'accordent bien avec toutes les propriétés expérimentales [12].

D'autre part, les avancées récentes en apprentissage automatique (*Machine Learning*) ont mis en lumière leur capacité à approximer efficacement des fonctions en haute dimension ainsi qu'à discriminer des propriétés complexes à partir d'observations, ce qui permet le développement de modèles génératifs puissants. Ce domaine émergent forme un nouveau cadre pour la conception de modèles de structure atomique ainsi qu'une alternative efficace et précise à la description énergétique des systèmes dans des phases condensées. Les algorithmes d'apprentissage automatique permettent ainsi une généralisation non biaisée du développement de potentiels empiriques puisqu'en apprenant à partir d'observations ils peuvent atteindre le niveau de théorie voulu au moment de la formation de l'ensemble de données.

Ces algorithmes ont déjà été appliqués avec succès lors de simulations de dynamique moléculaire pour approximer très efficacement la surface d'énergie potentielle du a-Si, telle que donnée par des calculs de DFT [6]. Mais ils ouvrent aussi la porte à des modèles génératifs contournant les procédures itératives des algorithmes exposés précédemment, en apprenant directement la distribution des positions atomiques. Après une introduction aux concepts de l'apprentissage automatique et profond, ces deux idées seront donc explorées dans plus de détail.

CHAPITRE 2

APPRENTISSAGE AUTOMATIQUE

L'apprentissage automatique (AA) est né historiquement comme l'outil permettant de développer l'intelligence artificielle, c'est-à-dire le développement de programmes informatiques capables de simuler des processus cognitifs de haut niveau comme l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique. L'impossibilité de définir concrètement notre manière d'effectuer ces tâches rend leur programmation explicite inefficace.

De ce fait le développement d'algorithmes dynamiques permettant d'*apprendre* comment effectuer ces tâches à partir d'observations, reflétant notre propre manière d'apprendre, a gagné en ampleur dans les dernières décennies. L'apprentissage automatique désigne alors l'évolution d'un modèle de traitement de l'information à l'aide d'observations, dans le but de mener à bien une tâche pour lequel il n'a pas été directement programmé.

La révolution des technologies de l'information, caractérisée notamment par la croissance exponentielle de la puissance de calcul des ordinateurs et par l'avènement de moyens de communication extrêmement efficaces (Internet) a fourni les ingrédients clés des avancées récentes en AA : la disponibilité de traitement et d'accès à de grands ensembles de données a rendu l'apprentissage statistique possible et efficace [44].

Les méthodes d'apprentissage sont alors apparues comme la seule direction viable pour l'élaboration d'algorithmes portant sur des tâches complexes : elles font l'objet d'intenses recherches menant à des avancées majeures dans les domaines de la reconnaissance visuelle (empreintes digitales, visages, conduite automatique), de la théorie des jeux (Échecs, Go) ou du traitement et de la synthèse vocale (*Siri*, *Cortana*). Leur application au domaine de la physique de la matière condensée a déjà permis de prédire avec succès des diagrammes de phases, des structures cristallines et des propriétés mécaniques [11, 53].

2.1 Contexte général et définitions

L'apprentissage à partir d'observations désigne le processus de construction d'un modèle général de traitement de l'information à partir d'un ensemble de données particulières du monde réel : le but est à la fois de prédire un comportement face à de nouvelles données et d'approximer des relations sous-jacentes aux observations. Un algorithme d'apprentissage peut donc être utilisé pour différentes tâches, incluant la classification de données, l'approximation de fonctions ou de densités de probabilités ou encore le partitionnement de données. Définir l'apprentissage pour un algorithme n'est pas simple, cependant Mitchell (1997) [39] en a donné une définition succincte :

"*Un programme informatique est dit d'apprendre à partir d'une expérience E par rapport à une classe de tâches T et mesure de performance P, si sa performance sur des tâches dans T augmente avec l'expérience E telle que mesurée par P.*"

Il est possible d'imaginer un grand nombre de différentes expériences E, tâches T et mesures P, dont nous ne verrons que les exemples les plus pertinents à ce travail dans les prochaines sections. Les observations dont on dispose initialement définissent le type de tâche que peut effectuer l'algorithme. Selon les cas, on distingue deux classes d'apprentissage : l'apprentissage **supervisé** et l'apprentissage **non-supervisé**.

L'apprentissage supervisé correspond au cas où on dispose de données d'entrée et de données de sortie, les cibles, pour lesquelles on cherche à approximer la fonction qui les lie. Les données d'entrée peuvent être par exemple des configurations atomiques, et les cibles leur énergie ou la phase dans lequel ledit solide se trouve. Lorsque les données de sorties sont discrètes, comme dans la prédiction de phase, on parle de **classification** tandis que lorsque les données de sortie sont continues, comme dans la prédiction d'énergie, on parle de **régression** [21].

L'algorithme d'apprentissage cherche à identifier une fonction permettant de faire des prédictions précises au-delà des observations d'entraînement : il doit être capable de *généraliser* correctement. Ainsi, l'ensemble des observations dont on dispose pour l'entraînement de l'algorithme est séparé en trois parties :

- **L'ensemble d'entraînement**, sur lequel l'algorithme est entraîné,
- **L'ensemble de validation**, utilisé pour calibrer les paramètres de l'algorithme,
- **L'ensemble de test**, utilisé pour l'évaluation de la performance de généralisation.

Les ensembles de validation et de test ne sont jamais utilisés pour l'entraînement du modèle, ce qui permet donc d'évaluer sa performance de manière non-biaisée.

L'apprentissage non supervisé correspond aux cas où l'on ne dispose que des données d'entrée et pour lesquelles on cherche par exemple à identifier des variables explicatives permettant de les partitionner, à estimer leur distribution de probabilité pour générer de nouvelles données où encore à les compresser ou enlever le bruit [21, 41, 53].

À titre d'exemple, supposons que l'on dispose d'un ensemble de configurations atomiques $\mathbf{X} = \{\mathbf{x}_i, i \in [1,...,N]\}$. Si on dispose aussi de l'ensemble des énergies correspondantes $\{\varepsilon_i, i \in [1,...,N]\}$, la tâche d'apprentissage supervisé consiste à apprendre la fonction $f : \mathbf{x} \rightarrow \varepsilon$, c'est-à-dire à approximer la fonction d'énergie à partir de l'ensemble des observations. Si on ne dispose que de $\mathbf{X}$ et que l'on souhaite générer de nouvelles configurations atomiques issues de la même distribution de probabilité, la tâche d'apprentissage non supervisé correspond à approximer la distribution de probabilité $p(\mathbf{x})$ à partir de l'ensemble d'observations. Des exemples d'applications possibles au domaine de la physique des matériaux sont présentés dans le tableau 2.I.

L'apprentissage statistique revient souvent à tirer des inférences d'un nombre réduit d'expériences dans un espace de possibilités de dimension élevée. Dans le cas des

	Algorithmes	Applications
Apprentissage supervisé	Moindres carrés régularisés Machines à vecteur de support	Relations structure-propriété
	Régression de Ridge régularisée	Hamiltoniens, prédiction de cristaux
	Réseaux de neurones	Structures, classification de cristaux
	Arbres de décision	Structures, identification de descripteurs
Apprentissage non supervisé	K-moyennes Champs de Markov aléatoires Analyse en composantes principales Regroupement hiérarchique	Relations structure-propriété, identification de descripteurs, réduction de bruit, transitions de phase

TABLEAU 2.I : Exemples d'applications à la physique des matériaux de certains algorithmes d'apprentissage supervisé et non supervisé [53].

configurations atomiques de N atomes, ayant $3N$ degrés de liberté, la dépendance exponentielle de la dimension de l'espace des états (classique ou quantique) avec le nombre de degrés de liberté est un obstacle computationnel majeur au calcul de l'évolution physique du système.

Ce phénomène est mieux connu dans le domaine de l'apprentissage automatique sous le nom de **fléau de la dimension** [9, 10]. Il empêche notamment l'application des méthodes locales aux problèmes d'apprentissage en haute dimension du fait que cela requière un nombre exponentiellement grand d'observations.

Par exemple, il est possible d'inférer les propriétés d'une nouvelle configuration en analysant les plus proches voisins de celle-ci dans l'espace des configurations. Deux points très proches sont susceptibles de partager des propriétés (énergie, volume, etc..) communes. Pour ce faire il est cependant nécessaire de disposer d'un échantillonnage dense de l'espace des configurations, de manière à toujours avoir une configuration de l'ensemble de données proche du point sur lequel on désire inférer des caractéristiques. Mais le nombre de points nécessaire pour atteindre une densité de points ρ fixe dans

un espace de dimension D croît exponentiellement avec D, ce qui explicite l'origine du fléau de la dimension.

Il est alors nécessaire de développer des méthodes statistiques plus raffinées pour les problèmes d'apprentissage, qui sont souvent formulés comme un problème d'optimisation d'une mesure de performance sur l'ensemble d'entraînement. Le tableau 2.II présente des exemples de mesures de performances communes pour différentes tâches d'apprentissage.
Les prochaines sections décrivent le cadre formel dans lequel s'inscrit l'apprentissage automatique ainsi que le processus d'entraînement de tels algorithmes.

	Classification	**Régression**	**Estimation de densité**
Signification de la cible y	Une classe parmi m classes	Une valeur réelle à prédire	Pas de cible
Domaine de y	$y \in [1, m]$	$y \in \mathbb{R}$	Pas de cible
Ce que $f(x)$ **vise à prédire**	Classe(x)	$\mathbb{E}[y\|x]$	La densité $p(x)$
Fonction objectif	$I_{\{f(x)\neq y\}}$	$(f(x)-y)^2$	$-\log f(x)$

TABLEAU 2.II : Propriétés des différents problèmes d'apprentissage automatique, où on considère une entrée x, une cible y et une fonction f cherchant à approximer leur relation. Les fonctions objectif mesurent respectivement l'erreur de classification, l'erreur quadratique et la log-vraisemblance négative. Elles constituent une mesure de l'erreur commise par le modèle f, il est donc souhaitable de les minimiser.

2.2 Cadre probabiliste de l'apprentissage supervisé

Le développement d'un modèle de prédiction pour les observables physiques se caractérise par la formulation d'hypothèses, attestant d'une connaissance antérieure, qui servent de base à une théorie permettant de faire des prédictions dont la validité est par la suite évaluée empiriquement. Ce mécanisme fondamental de la méthode scientifique peut s'exprimer mathématiquement en considérant une théorie prédictive comme une fonction f qui associe à une variable d'entrée $\mathbf{x}$ une prédiction $\hat{\mathbf{y}} = f(\mathbf{x})$ de la variable $\mathbf{y}$. La loi f peut être par exemple la deuxième loi de Newton ou l'équation de Schrödinger. Aussi bien dans la méthode scientifique que dans l'apprentissage automatique, le but est de trouver la fonction f assurant les meilleures prédictions. Les principes contrôlant cette recherche diffèrent cependant : l'apprentissage automatique se développe et rend compte des observations essentiellement à travers la théorie des probabilités et les statistiques appliquées [19, 41].

Considérons un ensemble d'observations et de cibles $\mathbb{D} = \{\mathbf{x}, \mathbf{y}\} \in \mathcal{X} \times \mathcal{Y}$ issu de la loi jointe $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y}|\mathbf{x})$ définie sur l'espace vectoriel $\mathcal{X} \times \mathcal{Y}$ où $\mathcal{X}$ est l'espace des variables d'entrée et $\mathcal{Y}$ l'espace des variables de sortie. La classification et la régression peuvent alors être vues comme les cas correspondant à $\mathcal{Y} \subseteq \mathbb{Z}^n$ ou $\mathcal{Y} \subseteq \mathbb{R}^n$ respectivement. L'ensemble $\mathbb{D}$ est donc formé par un ensemble de N réalisations de la distribution $p(\mathbf{x}, \mathbf{y})$ que l'on suppose indépendantes et identiquement distribuées (IID). Des variables IID sont des variables aléatoires indépendantes issues de la même distribution de probabilité, ce qui implique que la loi jointe de chaque paire de ces variables se factorise :

$$p(\mathbf{z}_1 = \mathbf{a}, \mathbf{z}_2 = \mathbf{b}) = p(\mathbf{z}_1 = \mathbf{a})p(\mathbf{z}_2 = \mathbf{b}) \,, \qquad \mathbf{z} = \mathbf{x}, \mathbf{y} \,. \tag{2.1}$$

L'hypothèse IID est importante dans le formalisme théorique de l'apprentissage, car elle est une condition d'application du théorème de Glivenko-Cantelli [58] qui stipule que la fonction de répartition (aussi appelée fonction de distribution cumulative) empirique ca-

ractérisant les observations $\mathbb{D}$, F_N, converge en norme presque sûrement[1] vers la fonction de répartition de laquelle $\mathbb{D}$ est tirée, F :

$$\|F_N - F\|_\infty = \sup_{x\in\mathbb{R}} |F_n(x) - F(x)| \longrightarrow 0, \qquad \text{presque sûrement .} \tag{2.2}$$

Ce théorème permet donc de s'assurer que l'ensemble des observations est bien représentatif de la distribution de probabilité sous-jacente, ce qui permet la justification probabiliste des méthodes d'apprentissage.

La tâche d'apprentissage supervisé consiste à trouver une fonction f qui approxime au mieux la relation $\mathbf{x} \to \mathbf{y}$ à partir de $\mathbb{D}$. On peut définir de manière très générale un **estimateur** (statistique) comme une fonction de $\mathbb{D}$, dont il est souhaitable qu'elle approxime une certaine variable. Étant donné que cet ensemble est issu d'un tirage aléatoire, l'estimateur f sera lui-même une variable aléatoire. Dans ce cadre, approximer la relation $\mathbf{x} \to \mathbf{y}$ signifie pour f de s'approcher de la loi conditionnelle $p(\mathbf{y}|\mathbf{x})$.

Nous allons considérer majoritairement des modèles paramétriques, c'est-à-dire que l'estimateur f est représenté par une fonction paramétrée f_θ[2] qui est vecteur dans un espace de fonctions $\mathbf{H}_\theta$, paramétré par $\theta \in \mathbb{R}^k$, appelé l'espace des hypothèses [19, 41].

La définition préalable de cet espace équivaut à considérer certaines fonctions comme plus plausibles que d'autres ce qui reflète les hypothèses faites sur la relation $\mathbf{x} \to \mathbf{y}$ *a priori*. Par exemple, il est commun de choisir de travailler dans l'espace des polynômes de degré k, dont l'espace des fonctions linéaires est un cas particulier pour $k = 1$.

Pour approximer le plus précisément possible la relation $\mathbf{x} \to \mathbf{y}$ il faut donc faire un choix de l'espace des hypothèses adapté, mais il faut aussi avoir suffisamment de données dans $\mathbb{D}$ de manière à pouvoir extraire toutes ses caractéristiques : un échantillonnage

[1] En théorie des probabilités, un événement est dit presque sûr s'il a une probabilité de un : lorsque l'univers (l'ensemble des issues possibles d'une expérience aléatoire) est fini, il s'agit d'un événement certain. Une convergence presque sûre est donc une convergence en probabilités.

[2] Lorsque le contexte est clair, on adoptera la notation $f_\theta = f_\theta(\mathbf{x})$ ainsi que $p(\theta) = p(f_\theta)$.

trop pauvre d'une distribution complexe rend la tâche d'apprentissage sous-déterminée. Dans certains cas il est donc impossible de trouver un modèle optimal, ce qui résulte en une erreur finie dans l'estimateur $\hat{\mathbf{y}} = f_\theta(\mathbf{x})$ dont nous pouvons rendre compte en écrivant

$$\mathbf{y} = f_\theta(\mathbf{x}) + \varepsilon \,, \tag{2.3}$$

où ε est une erreur aléatoire suivant une distribution $g(\varepsilon)$ inconnue. Celle-ci dépend en général des données d'entrée, mais nous pouvons supposer pour simplifier que l'ensemble d'entraînement est assez grand pour que cette dépendance ne soit pas importante[3] [53].

Le modèle f_θ suit une distribution de probabilité $p(\theta|\mathbb{D})$ qui décrit la probabilité que l'équation 2.3 soit satisfaite **après** l'observation de l'ensemble de données, avec une distribution d'erreurs g donnée : on l'appelle donc la distribution postérieure (*posterior*). La loi de Bayes donne alors

$$p(\theta|\mathbb{D}) = \frac{p(\mathbb{D}|\theta)}{p(\mathbb{D})} p(\theta) \,. \tag{2.4}$$

Ici, $p(\mathbb{D}|\theta)$ est la probabilité d'observer le jeu de données $\mathbb{D}$ étant donné f et g : on appelle cette distribution la vraisemblance (*likelihood*). La distribution $p(\theta)$ est appelée la distribution à priori (*prior*) du fait qu'elle représente la probabilité de satisfaire 2.3 étant donné g **avant** l'observation. Le terme $p(\mathbb{D})$ ne dépend pas de f_θ et peut donc être considéré comme une constante de normalisation.

La loi de Bayes fournit donc une interprétation probabiliste de l'apprentissage automatique. L'espace des hypothèses est initialement défini de manière à contraindre la forme de f à travers la distribution à priori : les fonctions n'appartenant pas à $\mathbf{H}_\theta$ sont assignées d'une probabilité de zéro. Ces probabilités représentent la croyance préalable

[3] Comme nous ne disposons pas d'informations sur cette distribution, elle sera toujours implicitement marginalisée dans les développements subséquents i.e. $p(\mathbf{z}) \equiv p_g(\mathbf{z}) = p(\mathbf{z}|g)$

qu'une fonction particulière satisfait l'équation 2.3. Au fur et à mesure que l'ensemble de données est observé, ces probabilités sont mises à jour dans la distribution postérieure : il s'agit là du processus d'apprentissage. Une étape d'apprentissage correspond donc à l'application en chaîne de la loi de Bayes. Si on a déjà observé l'ensemble $\mathbb{D}_1$, l'apprentissage par l'observation de l'ensemble $\mathbb{D}_2$ est décrit par

$$p(\theta|\mathbb{D}_2,\mathbb{D}_1) = \frac{p(\mathbb{D}_2|\theta,\mathbb{D}_1)}{p(\mathbb{D}_2|\mathbb{D}_1)} p(\theta|\mathbb{D}_1) \,. \tag{2.5}$$

où la distribution postérieure devient subséquemment la distribution à priori de l'étape suivante [53].

Il existe plusieurs manières de formaliser la recherche de l'estimateur optimal $\hat{\mathbf{y}}^* = f^* = f_{\theta^*}$. Une approche commune est celle de maximiser la probabilité de satisfaire 2.3, étant donné $\mathbb{D}$ et une distribution d'erreur g. Celle-ci est la distribution postérieure $p(\theta|\mathbb{D})$, et le modèle la maximisant est appelé l'estimateur du **maximum a posteriori** (MAP) [5]. La recherche de cet estimateur correspond donc à trouver les paramètres optimaux θ^* tels que

$$\begin{aligned}
\theta^* &= \arg\max_\theta p(\theta|\mathbb{D}) && \text{(2.6a)}\\
&= \arg\max_\theta p(\mathbb{D}|\theta)p(\theta) && \text{(2.6b)}\\
&= \arg\max_\theta \prod_i p(\{\mathbf{x}_i,\mathbf{y}_i\}|\theta)p(\theta) && \text{(2.6c)}\\
&= \arg\min_\theta -\sum_i \log p(\{\mathbf{x}_i,\mathbf{y}_i\}|\theta) - \log p(\theta) \,. && \text{(2.6d)}
\end{aligned}$$

Ainsi, la recherche de l'estimateur du MAP pour un modèle paramétré par le vecteur θ se traduit en un problème d'optimisation. La fonction devant être minimisée est appelée la **fonction objectif**, ou fonction de perte, et dans le cas de la méthode du maximum a posteriori elle est formée de la somme de la log-vraisemblance $\log p(\mathbb{D}|\theta)$ et du logarithme de la distribution à priori $\log p(\theta)$. Ce dernier est appelé terme de régularisation car il englobe les contraintes du problème d'optimisation sur l'espace des hypothèses.

Une autre méthode d'estimation communément utilisée en statistiques est celle du maximum de vraisemblance [1], qui consiste à ne considérer que la log-vraisemblance dans la fonction objectif. Elle est donc un cas particulier de l'estimation MAP où on considère une distribution a priori uniforme i.e. $p(\theta) = \text{constante}$. L'estimateur du MAP peut alors être vu comme une version régularisée de l'estimateur par maximum de vraisemblance, c'est pourquoi il est souvent préféré à ce dernier du fait de sa plus grande généralité, bien que les deux soient très communs dans les domaines de l'apprentissage automatique et de l'inférence statistique.

Un cadre encore plus général consiste à fonder la recherche de l'estimateur optimal sur la minimisation du **risque empirique**. Celui-ci est défini comme l'évaluation d'une fonction objectif générale $\mathcal{L}$ mesurant l'erreur d'estimation commise sur l'ensemble d'entraînement (des exemples de fonctions objectif ont déjà été présentés au tableau 2.II), et l'algorithme minimise alors le risque empirique

$$R_e(\theta) = \frac{1}{N} \sum_i \mathcal{L}(f_\theta(\mathbf{x}_i), \mathbf{y}_i) \; . \tag{2.7}$$

L'estimation du MAP est donc un cas particulier de la minimisation du risque empirique R_e où

$$R_e(\theta) = -p(\mathbb{D}|\theta)p(\theta) \; . \tag{2.8}$$

La fonction objectif sert de *mesure* de la performance du modèle en ce qu'elle définit une métrique sur $\mathbf{Y}$. Cette mesure permet de calculer le **risque** : il s'agit de l'espérance

de l'erreur d'estimation commise sur la loi jointe $p(\mathbf{x},\mathbf{y})$, telle que mesurée par $\mathcal{L}$[4]

$$R(\theta) = \mathbb{E}_{p(\mathbf{x},\mathbf{y})}[\mathcal{L}(f_\theta(\mathbf{x}),y)] \tag{2.9a}$$

$$= \int_{\mathcal{X}\times\mathcal{Y}} \mathcal{L}\mathrm{dP}(\mathbf{x},\mathbf{y}) \tag{2.9b}$$

$$= \int_{\mathcal{X}\times\mathcal{Y}} \mathcal{L}(f_\theta(\mathbf{x}),y)p(\mathbf{x},\mathbf{y})\mathrm{d}x\mathrm{d}y\ . \tag{2.9c}$$

Cette espérance est impossible à calculer sans connaître soit la fonction de répartition ou la densité de probabilité des variables aléatoires $(\mathbf{x},\mathbf{y})$. Le risque empirique peut alors être vu comme un estimé Monte Carlo du risque, puisque l'ensemble d'entraînement $\mathbb{D}$ est un échantillonnage IID de la loi jointe $p(\mathbf{x},\mathbf{y})$. La recherche de l'estimateur optimal par le biais de la minimisation du risque empirique correspond alors à trouver le vecteur de paramètres optimal θ^* satisfaisant

$$\theta^* = \arg\min_\theta R_e(\theta) \tag{2.10}$$

$$= \arg\min_\theta \sum_i \mathcal{L}(f_\theta(\mathbf{x}_i),\mathbf{y}_i)\ . \tag{2.11}$$

[4]On adopte la notation standard pour la fonction de répartition P associée à la densité de probabilité p, telles que $P(\mathbf{z}) = \int_{-\infty}^{\mathbf{z}} p(\mathbf{z})\mathrm{d}\mathbf{z}$

2.3 Estimateurs, Biais et Variance

Il est courant en physique expérimentale de chercher à ajuster une fonction sur des données empiriques dans le but d'extrapoler une forme analytique pour la relation entre deux quantités mesurées. Supposons que le résultat d'une mesure consiste en un ensemble de points $\mathbb{D} = \{x_i, y_i, i \in [1,...,N]\}$ et que l'expérimentateur cherche à en tirer la forme mathématique de la loi $f : x \to y$, tel qu'illustré à la figure 2.1.

Une solution possible, mais naïve, consiste à former un polynôme de degré N et d'ajuster ses coefficients $\theta = \{a_0, a_1, ..., a_N\}$ de manière à passer exactement par chacun des points $\{x_i, y_i\}$ du plan (ceci est toujours possible en vertu du théorème fondamental de l'algèbre). Une faible fluctuation dans les observations entraînerait une grande erreur dans un tel ajustement (figure 2.1c)). On dit alors que le modèle a une grande **variance**, car l'erreur d'estimation commise, en supposant que ces N points sont issus d'un polynôme de degré N, pénalise la performance de généralisation. Ceci peut éventuellement mener le modèle à **pulvériser** l'ensemble d'entraînement, c'est-à-dire qu'il ne fera aucune erreur sur cet ensemble et qu'il l'aura en un certain sens *mémorisé*, et on parle alors de **sur-apprentissage**.

Au contraire l'ajustement linéaire dans la figure 2.1a) pourrait manquer des relations pertinentes entre les données d'entrée et de sortie si la relation sous-jacente est non linéaire ; le modèle commet alors une erreur d'approximation et aura un grand **biais** du fait qu'il considère des hypothèses trop simples. Ceci conduit aussi à une faible performance de généralisation caractérisée par un régime de **sous-apprentissage** [19, 21, 41]. L'heuristique adoptée dans le cadre de l'apprentissage automatique est alors celle du *rasoir d'Occam* : si deux hypothèses expliquent également bien les observations, la plus simple doit être préférée[5]. Dans cet exemple, l'hypothèse faite par l'expérimentateur est la forme de la fonction à ajuster sur $\mathbb{D}$; elle peut être un polynôme d'un certain degré, une combinaison de Gaussiennes, etc... Un polynôme de degré faible sera naturellement une hypothèse *plus simple* qu'un polynôme de haut degré.

[5] Où, comme le disait Wittgenstein : «Si un signe n'a pas d'usage, il n'a pas de signification. Tel est le sens de la devise d'Occam».

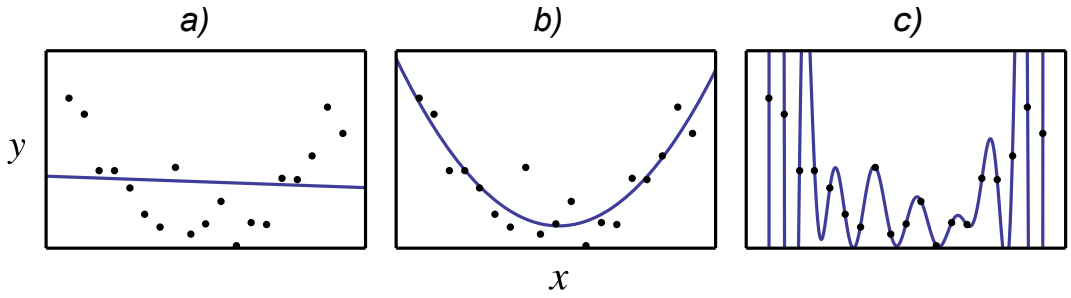

FIGURE 2.1 : Illustration du concept de sous-apprentissage et de sur-apprentissage pour un ensemble de points $y = x^2 + N(0, \sigma)$, où $N(0, \sigma)$ est une distribution Gaussienne centrée de variance σ. Un modèle a) linéaire, b) polynomial de degré 2 et c) fonctionnel de haute capacité sont ajustés sur le même ensemble de points. Le modèle a) commet une grande erreur d'ajustement car l'hypothèse d'une relation linéaire est trop simple. Le modèle c) présente l'erreur la plus faible sur l'ensemble de points mais n'explique pas la loi sous-jacente aux données car l'ajout d'une nouvelle observation résulterait en une grande erreur d'ajustement.

Les hypothèses faites sur la nature de la fonction recherchée, contraignant sa complexité et donc sa capacité de modélisation, ont donc une influence importante sur la qualité du modèle résultant. Ces hypothèses, comme le degré du polynôme considéré ou la forme de la fonction à ajuster, sont un exemple d'**hyper-paramètres**. Ils se différencient des paramètres d'un modèle, comme les coefficients polynomiaux, en ce qu'ils ne sont pas optimisés pour un ajustement optimal de la fonction sur l'ensemble de données. Ils représentent des choix fixés sur la structure de l'algorithme d'apprentissage.

La figure 2.2 montre la tendance générale de l'erreur d'entraînement et de généralisation. Il existe en effet une capacité optimale pour un problème d'apprentissage donné qui se situe entre le régime de sous-apprentissage et le régime de sur-apprentissage ; la recherche d'un ajustement optimal nécessite donc une attention particulière aux facteurs influant sur la capacité du modèle.

La minimisation du risque empirique dépend grandement de la définition de l'espace des hypothèses, c'est-à-dire de la distribution a priori. Nous avons vu que les contraintes imposées sur $\mathbf{H}_\theta$ peuvent se traduire en un terme de régularisation additionnel dans la fonction objectif $\mathcal{L}(\theta)$, ou être directement incorporées dans le choix de $\mathbf{H}$. Le choix de ces contraintes permet de contrôler le biais et la variance du modèle, et a donc un

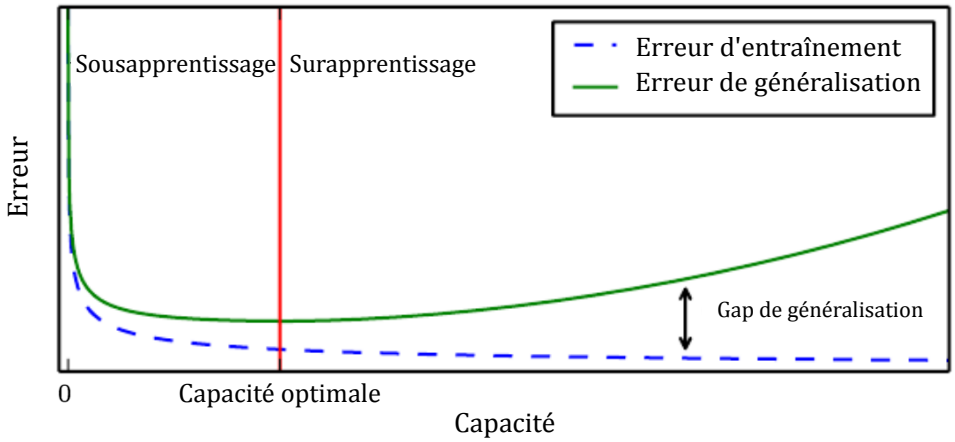

FIGURE 2.2 : Erreur d'entraînement et de généralisation en fonction de la capacité [19]. Lorsque la capacité est faible, l'erreur d'entraînement et de généralisation sont grandes : c'est le régime de sous-apprentissage. Lorsque l'on augmente la capacité, l'erreur d'entraînement décroît toujours, mais au delà de la capacité optimale le gap entre celle-ci et l'erreur de généralisation augmente : c'est le régime de sur-apprentissage.

impact sur le sous-apprentissage et le sur-apprentissage [19, 21]. Supposons qu'il existe une relation fonctionnelle entre les données d'entrée et de sortie telle que

$$\mathbf{y} = h(\mathbf{x}) + \varepsilon \,, \tag{2.12}$$

où h est la fonction exacte générant les variables de sortie et ε est une variable aléatoire représentant le bruit issu du processus d'acquisition des données, que l'on suppose suivre une loi normale $\mathcal{N}(0, \sigma^2)$. Le fait que ces données soient bruitées implique qu'il y aura une erreur *irréductible* dans cet estimateur : on l'appelle **l'erreur de Bayes**.
Le biais de f_θ est défini comme étant la différence entre son espérance et la valeur de $\mathbf{y}$:

$$\mathbb{B}[f_\theta] = \mathbb{E}[f_\theta(\mathbf{x}) - \mathbf{y}] \,. \tag{2.13}$$

La variance de f_θ est définie comme l'espérance prise sur les écarts du modèle par

rapport à son espérance au carré :

$$\mathbb{V}[f_\theta] = \mathbb{E}[(f_\theta(\mathbf{x}) - \mathbb{E}[f_\theta(\mathbf{x})])^2] \,. \tag{2.14}$$

Le lien entre le biais et la variance du modèle et la minimisation du risque empirique se distingue clairement dans la méthode très répandue des moindres carrés, qui correspond à considérer la fonction objectif

$$\mathcal{L}(f_\theta(\mathbf{x}), \mathbf{y}) = (f_\theta(\mathbf{x}) - \mathbf{y})^2 \,. \tag{2.15}$$

On peut exprimer le risque associé à $\mathcal{L}$, à partir de l'équation 2.9, directement en fonction du biais et de la variance du modèle. Remarquons premièrement que 2.12 implique que

$$\begin{aligned} \mathbb{V}[y] &= \mathbb{E}[(y - \mathbb{E}[y])^2] = \mathbb{E}[(y-h)^2] && \text{(2.16a)} \\ &= \mathbb{E}[\varepsilon^2] = \sigma^2 \,. && \text{(2.16b)} \end{aligned}$$

Le risque s'écrit alors

$$\begin{aligned} \mathbb{E}[(f_\theta - \mathbf{y})^2] &= \mathbb{E}[f_\theta^2] + \mathbb{E}[y^2] - 2\mathbb{E}[f_\theta \mathbf{y}] && \text{(2.17a)} \\ &= \mathbb{V}[f_\theta] + \mathbb{E}[f_\theta]^2 + \mathbb{V}[y] + \mathbb{E}[y]^2 - 2h\mathbb{E}[f_\theta] && \text{(2.17b)} \\ &= \mathbb{V}[f_\theta] + \mathbb{V}[y] + (h^2 - 2h\mathbb{E}[f_\theta] + \mathbb{E}[f_\theta]^2) && \text{(2.17c)} \\ &= \mathbb{V}[f_\theta] + \mathbb{B}[f_\theta] + \sigma^2 \,. && \text{(2.17d)} \end{aligned}$$

Cette expression du risque souligne clairement le **dilemme biais-variance** : l'erreur de Bayes σ^2 forme une borne inférieure au risque, et nous avons maintenant explicité le fait que la minimisation du risque résulte nécessairement en un compromis entre la minimisation du biais et de la variance du modèle tel qu'illustré à la figure 2.3.

La définition de la fonction objectif et le choix de $\mathbf{H}$ sont des hyper-paramètres importants, qui avec le nombre de paramètres k permettent de contrôler la complexité du modèle. Il est maintenant important de clarifier cette notion pour évaluer et contrôler le

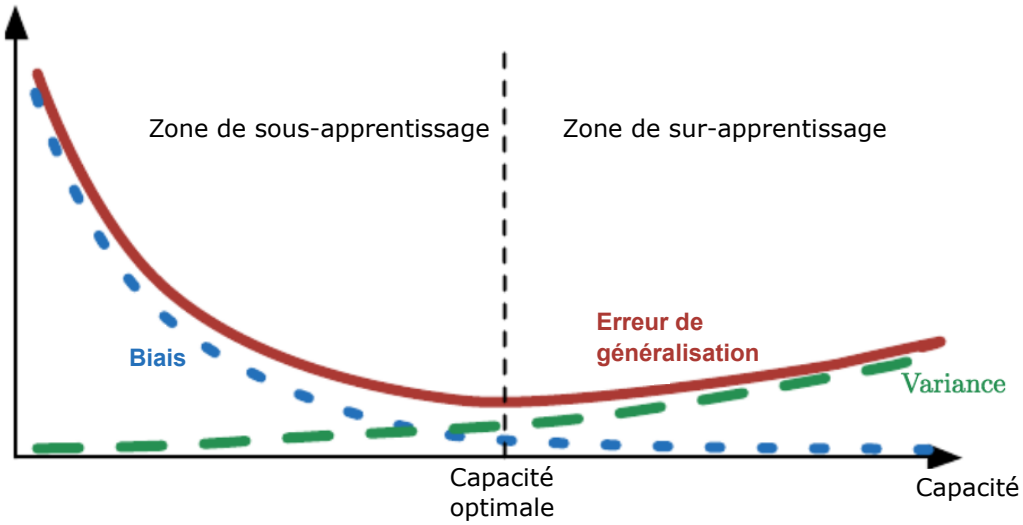

FIGURE 2.3 : Illustration du dilemme biais-variance [19] : lorsque la capacité augmente, le biais (bleu) tend à diminuer tandis que la variance (vert) tend à augmenter. Il y a une capacité optimale correspondant au minimum de l'erreur de généralisation (rouge), qui démarque la séparation entre les régimes de sur-apprentissage et de sous-apprentissage.

sous-apprentissage et le sur-apprentissage.

2.4 Convergence de l'apprentissage et capacité

Le but de la tâche d'apprentissage supervisé n'est pas de minimiser le risque empirique : bien qu'il soit une mesure de l'erreur commise par le modèle, il ne permet pas d'identifier la performance de l'algorithme en dehors de l'ensemble d'entraînement. Celle-ci est mesurée par l'erreur de généralisation, qui est est définie comme la différence entre le risque et le risque empirique

$$R_g[f] \equiv R[f] - R_e[f] \, . \tag{2.18}$$

c'est-à-dire la différence entre l'erreur sur la loi jointe $p(\mathbf{x}, \mathbf{y})$ et l'erreur d'entraînement. Encore une fois, il est possible de considérer l'ensemble de test comme un échantillonnage de la loi jointe, ce qui permet d'avoir un estimé Monte Carlo de l'erreur de généra-

lisation fournissant alors une mesure directe de l'apprentissage de la loi sous-jacente par le modèle.

Alors que le risque empirique est voué à diminuer jusqu'à atteindre éventuellement l'erreur de Bayes, car $\theta \to \theta^* = \arg\min_\theta R_e(\theta)$, l'erreur de généralisation présentera un minimum en fonction de la capacité du modèle qui permet de séparer empiriquement le régime de sous-apprentissage et celui de sur-apprentissage, comme le montre la figure 2.3. Le dilemme biais-variance peut donc être contrôlé en regardant l'erreur de généralisation en fonction de la capacité, mais ce dernier terme reste à décrire. La capacité est une mesure du pouvoir expressif d'une fonction f_θ, c'est-à-dire de sa capacité à approximer une large famille de fonctions. Nous avons vu précédemment qu'elle est contrôlable a priori au travers du choix de la fonction objectif et de l'espace des hypothèses, mais celle-ci augmente aussi lors de l'entraînement.

Une définition plus formelle de la capacité d'un modèle est celle de la dimension Vapnik-Chervonenkis [59], définie comme le cardinal du plus grand ensemble de points que l'algorithme peut pulvériser :

$$d_f = \max\{k | \text{card}(E) = k, f \text{ pulvérise } E\} \ . \tag{2.19}$$

Un résultat important de la théorie de Vapnik-Chernonenkis montre que l'erreur de généralisation est bornée supérieurement par une fonction $M(C,N)$ qui croît de façon monotone en fonction de D_f et décroît de façon monotone en termes du nombre d'observations dans l'ensemble d'entraînement N [19, 59, 60].

$$R_g[f] \leq \mathcal{M}(d_f, N) \ .$$

Ceci signifie que lorsque la taille de l'ensemble d'entraînement augmente, le risque empirique devient une bonne approximation du risque. Aussi, cette inégalité permet de justifier l'existence du régime de sur-apprentissage puisqu'une plus grande capacité mène à une plus grande borne supérieure, laissant donc la place à une plus grande erreur de généralisation.

Cette borne supérieure permet une justification théorique du fait que les algorithmes d'apprentissage fonctionnent, mais elle reste rarement utilisée en pratique. Ceci est dû au fait qu'il peut être parfois difficile d'estimer la capacité d'un algorithme d'apprentissage en pratique, mais aussi parce que cette borne est rarement atteinte.

La théorie de l'apprentissage montre qu'il est possible pour un algorithme d'apprentissage de généraliser correctement à partir d'un ensemble fini d'exemples. En pratique, du fait qu'ils suivent un processus inductif, les algorithmes d'apprentissage ne fournissent que des règles probabilistes : le modèle est alors l'approximation qui minimise l'erreur de généralisation avec la plus haute probabilité sur la plupart de l'ensemble de données.

Le *no free lunch theorem* [19, 62] affirme cependant que tous les algorithmes de classification ont, en moyenne sur toutes les distributions de données, la même performance de généralisation. Ceci signifie qu'aucun algorithme d'apprentissage ne peut être universellement meilleur que les autres, c'est-à-dire sur toutes les tâches possibles. Le but de l'apprentissage automatique n'est donc pas de trouver un algorithme général s'appliquant efficacement sur toutes les distributions, mais plutôt de rechercher quels algorithmes sont les plus adaptés selon la tâche considérée.

2.5 Apprentissage non supervisé

Bien que le formalisme présenté jusqu'ici a été développé dans le cadre de l'apprentissage supervisé, une grande partie des développements précédents s'appliquent encore dans le cadre de l'apprentissage non supervisé. Ne disposant pas de données de sortie, l'apprentissage non supervisé a pour but de déterminer les structures sous-jacentes des données d'entrée, c'est-à-dire de déterminer les propriétés de la loi marginale $p(\mathbf{x})$. La délimitation entre l'apprentissage supervisé et non supervisé n'est pas formellement définissable dans le sens qu'il n'existe pas de test objectif permettant de déterminer si un trait appartient aux données d'entrée ou s'il s'agit d'une donnée sortie fournie par un

«superviseur» [19]. Cependant, les régimes de sous-apprentissage et le sur-apprentissage sont encore définis et il convient encore de séparer l'ensemble de données en ensembles d'entraînement, de validation et de test.

L'estimation du maximum de vraisemblance ou du maximum a posteriori, et plus généralement la minimisation du risque empirique, s'appliquent également à l'apprentissage non supervisé. Le modèle est un estimateur qui vérifie encore la loi de Bayes 2.4 avec maintenant $\mathbb{D} = \{\mathbf{x}_i\}$. La tâche non supervisée peut consister par exemple à trouver un partitionnement des données, à estimer la densité de probabilité $p(\mathbf{x})$, à réduire la dimension ou même à détecter des anomalies [21, 53].

L'apprentissage non supervisé nécessite donc la définition d'une fonction objectif spécifique pour la tâche. Par exemple, l'ajustement d'une courbe sur un ensemble de données expérimentales peut être vu comme une tâche d'apprentissage non supervisé où la fonction objectif quantifie l'erreur d'ajustement. Il s'agit donc de former une représentation *simple* des données tout en préservant le maximum d'information. Il existe plusieurs manières de définir ce qu'est une représentation simple : les plus communes sont les représentations de basse dimension, les représentations creuses et les représentations indépendantes. Par exemple, l'analyse en composantes principales est un algorithme d'apprentissage non supervisé qui représente les observations, de haute dimension et possiblement corrélées, en un ensemble plus petit de composantes principales décorrélées atteignant donc une représentation à la fois de basse dimension et indépendante. Les données sont donc projetées sur un sous-espace orthogonal dont les axes ont le maximum de variance, permettant ainsi de tenir compte au maximum de la variabilité des données et préservant donc le maximum d'information.

Les algorithmes d'apprentissage non supervisé les plus pertinents à la modélisation de structure atomique sont les modèles génératifs profonds, en particulier les Réseaux Accusatoires Génératifs et les Auto-Encodeurs Variationnels dont il sera question au chapitre 3.

2.6 Entraînement d'un algorithme d'apprentissage

Le cadre de la minimisation du risque empirique pour un modèle paramétré f_θ transforme la tâche d'apprentissage en un problème d'optimisation multivariée en haute dimension, en général non linéaire et non convexe, pour la fonction scalaire $\mathcal{L} : \mathbb{R}^k \to \mathbb{R}$.

$$\theta^* = \arg\min_\theta \mathcal{L}(\theta, \mathbb{D}), \qquad \theta \in \mathbb{R}^k \,. \tag{2.20}$$

La forme analytique de $\mathcal{L}$ étant impossible à connaître sans une expression pour la loi jointe $p(\mathbf{x}, \mathbf{y})$, le problème doit nécessairement être résolu numériquement. Bien qu'il existe une grande variété d'algorithmes d'optimisation pouvant être appliqués à ce problème, les plus communément utilisés en apprentissage automatique sont les algorithmes d'optimisation différentiable[6]. Il s'agit de méthodes itératives qui se basent sur une approximation locale de la fonction, supposée ici de classe $C^\infty(\mathbb{R})$, à l'aide de son expansion de Taylor

$$\begin{aligned} \mathcal{L}(\theta) &= \sum_{\substack{i=0 \\ |\alpha|=i}}^{\infty} \frac{D^\alpha \mathcal{L}(\theta_0)}{\alpha!}(\theta - \theta_0)^\alpha \qquad |\alpha| = \sum_i \alpha_i, \quad \alpha! = \prod_i \alpha_i! \\ &\approx \mathcal{L}(\theta_0) + \nabla\mathcal{L}(\theta_0)^{\mathrm{T}}(\theta - \theta_0) + \frac{1}{2}(\theta - \theta_0)^{\mathrm{T}} \nabla^2 \mathcal{L}(\theta_0)(\theta - \theta_0) \,, \end{aligned} \tag{2.21}$$

et le but est d'atteindre la condition optimale

$$\frac{\partial \mathcal{L}}{\partial \theta}(\theta^*) = 0 \tag{2.22}$$

On distingue alors deux types de méthodes : les algorithmes de premier et de deuxième ordre. Les premiers n'utilisent que l'information du Jacobien $\nabla_\theta \mathcal{L}$, on parle de l'algorithme du gradient, tandis que les seconds se servent aussi du Hessien $\nabla^2_\theta \mathcal{L}$, on parle alors de méthodes de Newton. Bien que l'information de courbure contenue dans la matrice

[6] En effet, le grand nombre de paramètres présent dans les algorithmes d'apprentissage rend les méthodes d'optimisation globale infaisables : il s'agit d'une autre conséquence du fléau de la dimension.

Hessienne soit avantageuse pour la convergence de ces méthodes, elle reste peu utilisée en apprentissage automatique à cause de son grand coût computationnel[7]. En pratique, la vaste majorité des algorithmes d'apprentissage utilisent des méthodes de premier ordre, et les seules méthodes de deuxième ordre utilisées sont celles de type quasi-Newton (L-BFGS, Gradients Conjugués), qui n'exhibent cependant qu'une faible amélioration de qualité. Il est aussi important de noter qu'il n'y a aucune garantie de convergence pour ces algorithmes dans le cadre de l'optimisation non-convexe, ce qui implique que leur efficacité ne peut être jugée qu'empiriquement.

L'algorithme du gradient (AG) exécute à chaque itération un déplacement dans l'espace des paramètres $\mathbb{R}^k$ vers la direction du plus grand décroissement de $\mathcal{L}$. Après l'initialisation des paramètres à θ_0, ceux-ci sont mis à jour selon la règle

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta, \mathbb{D}) \tag{2.23}$$

$$= \theta_t - \frac{1}{N} \eta \nabla_\theta \sum_{i=0}^{i=N} \mathcal{L}(f_\theta(\mathbf{x}^{(i)}), \mathbf{y}^{(i)}) \,, \tag{2.24}$$

où η est le **pas de gradient** (*learning rate*) : c'est un hyper-paramètre contrôlant la taille du pas effectué par l'algorithme. Clairement, lorsque θ est proche d'un minimum local ou global le processus devient stationnaire, c'est-à-dire que $\theta_{t+1} \approx \theta_t$, et l'algorithme est arrêté.

S'agissant d'un algorithme de premier ordre itératif, l'AG est sensible à l'initialisation θ_0 des paramètres. En particulier l'initialisation aléatoire des paramètres est nécessaire pour briser la symétrie du modèle et éviter l'existence de paramètres suivant la même évolution, qui seraient donc redondants. De plus, la convergence de l'AG peut être lente lorsque la courbure de la fonction objectif varie beaucoup entre différentes directions. Celle-ci dépend grandement du pas de gradient : une valeur trop faible de η résulte en une convergence lente tandis qu'une valeur trop grande mène éventuellement

[7] Le calcul de la matrice Hessienne prend $O(k^2)$ en mémoire, ce qui est très prohibitif.

à une divergence. En effet lorsqu'elle dépasse la plus grande valeur propre du Hessien $\varepsilon > \lambda_{\max}(\nabla^2 \mathcal{L})$, l'AG résulte en une augmentation de la valeur de $\mathcal{L}$ au lieu d'une diminution.

La fonction objectif est le plus souvent additive en termes des observations[8] ; mais évaluer la fonction objectif sur l'ensemble d'entraînement au complet peut devenir rapidement très inefficace ($O(N)$), en particulier au vu du grand nombre de données nécessaires pour l'apprentissage statistique. Le gradient calculé dans l'équation 2.23 est une espérance sur l'ensemble d'entraînement, qui peut donc être approximée en utilisant un petit échantillon aléatoire de cet ensemble grâce à l'additivité de $\mathcal{L}$. L'algorithme du gradient stochastique (AGS) consiste alors à évaluer le gradient sur un échantillon de taille M différent à chaque itération :

$$\mathbf{g} = \frac{1}{M} \sum_{i=1}^{i=M} \nabla_\theta \mathcal{L}(\mathbb{D}_i, \theta) \ . \tag{2.25}$$

Le traitement de chaque terme dans un mini-lot peut être efficacement parallélisé ce qui résulte en un gain notable en performance. En pratique, l'ensemble d'entraînement est mélangé puis divisé en $\frac{N}{M}$ lots qui sont fournis cycliquement à l'AGS. On définit ainsi une **époque** d'entraînement comme l'itération de l'AGS sur l'ensemble d'entraînement entier, c'est-à-dire sur l'ensemble des lots dont il est formé. Le nombre d'époques d'entraînement est aussi un hyper-paramètre, que l'on ajuste empiriquement pour correspondre au minimum de l'erreur de généralisation.

La taille des lots, tout comme le pas de gradient, est un hyper-paramètre crucial pour la performance de l'AGS. Une grande valeur de M conduit à un meilleur estimé du gradient et à un gain de vitesse de traitement au prix d'un nombre réduit d'itérations par époque. En revanche une faible valeur de M conduit à un estimé bruité du gradient, qui peut avoir un effet régularisateur[9], et permet aussi à l'AGS d'effectuer des mises à jour

[8] C'est le cas par exemple pour la log-vraisemblance et l'erreur quadratique moyenne.

[9] En effet l'introduction de bruit dans une procédure d'optimisation itérative, semblable aux fluctuations thermiques du mouvement d'une particule, peut avoir des effets bénéfiques en aidant par exemple

plus fréquentes au sein d'une époque. Ceci résulte en un gain net dans la vitesse de convergence au prix de mises à jours ayant une plus grande variance, ce qui engendre des fluctuations de la fonction objectif. Des faibles valeurs entre 32 et 256 sont donc préférées du fait que le bruit d'estimation permet de s'échapper des minima très localisés et de préférer les minima délocalisés et plats qui mènent à une meilleure généralisation.

La taille des lots et le pas de gradient ne sont pas indépendants : une plus grande taille de lots M réduit la variance de l'estimé du gradient et permet donc l'utilisation d'un pas de gradient η plus grand.

L'AGS tel que décrit ici présente cependant des difficultés, à la fois dans le choix des hyper-paramètres tels que η, M, mais aussi au niveau algorithmique. En pratique, une grande variété de modifications de l'AGS ont été développées et résultent en une performance accrue. Ces stratégies d'optimisation sont expliquées en détail dans l'annexe I.

2.7 Régularisation

Le dilemme biais-variance est un problème fondamental pour l'entraînement d'un algorithme d'apprentissage. Le but de ces algorithmes est d'obtenir la meilleure performance de généralisation possible, et pour ce faire il faut éviter de tomber dans le régime de sur-apprentissage. L'ensemble des techniques destinées explicitement à réduire l'erreur de généralisation, possiblement au coût d'une erreur d'entraînement plus grande, est appelé la régularisation [60].

Les techniques de régularisation peuvent prendre la forme de contraintes sur le domaine des paramètres, de termes additionnels dans la fonction objectif, de stratégies d'entraînement, ou même de perturbations des paramètres ou des données d'entrée.

à échapper à des mauvais minima locaux. C'est d'ailleurs une technique utilisée explicitement dans les méthodes de recuit simulé (*Simulated annealing*).

Les pénalités de norme sur les paramètres sont une stratégie de régularisation commune visant à restreindre le domaine des paramètres pour contrôler la capacité du modèle. Formellement cela consiste à construire une nouvelle fonction objectif régularisée $\widetilde{\mathcal{L}}$

$$\widetilde{\mathcal{L}}(\theta) = \mathcal{L}(\theta) + \lambda\Omega(\theta)\,, \tag{2.26}$$

où $\lambda \in [0,\infty)$ est un hyper-paramètre contrôlant la contribution relative du terme de régularisation Ω par rapport à la fonction objectif originale $\mathcal{L}$. Il est possible de pénaliser n'importe quelle norme des paramètres parmi les normes de L^p[10]. La $p-$norme d'un vecteur $\mathbf{z}$ est définie par

$$\|\mathbf{z}\|_p = \left(\sum_{i=1}^{n} |z_i|^p\right)^{\frac{1}{p}}. \tag{2.27}$$

Les cas les plus communs sont $p = 1,2,\infty$, qui correspondent respectivement à la norme de Manhattan, la norme Euclidienne usuelle et la norme infinie $\|\mathbf{z}\|_\infty = \max_i\{|z_i|\}$. Pour des tenseurs arbitraires, la p-norme se généralise en sommant sur tous les éléments du tenseur, ce qui correspond à vectoriser le tenseur avant de prendre la norme usuelle.

La régularisation L^2, aussi appelée la régression de Ridge ou la régularisation de Tikhonov, permet de pénaliser plus fortement les paramètres de grande norme que ceux de faible norme en prenant $\Omega(\theta) = \frac{1}{2}\|\theta\|_2$. Couplée à l'AGS, la régularisation L^2 correspond à ajouter un échéancier sur les paramètres puisque la règle de mise à jour avec la nouvelle fonction objectif devient

$$\theta \leftarrow \theta - \varepsilon\nabla_\theta\widetilde{\mathcal{L}}(\theta) \tag{2.28}$$

$$\leftarrow \theta - \varepsilon\nabla_\theta\mathcal{L}(\theta) - \varepsilon\lambda\theta \tag{2.29}$$

$$\leftarrow \theta(1-\varepsilon\lambda) - \varepsilon\nabla_\theta\mathcal{L}(\theta)\,, \tag{2.30}$$

ce qui correspond donc à un échéancier exponentiel des paramètres[11] de facteur $1-\varepsilon\lambda$.

[10] Les espaces L^p sont définis ici comme des espaces vectoriels sur $\mathbb{R}^n$ munis de la $p-$norme.

[11] Il est cependant important de noter que pour ADAM cette équivalence ne tient plus : l'échéancier sur les paramètres doit être intégré séparément et il est plus recommandé d'utilisation que la régularisation L^2 [32]

Cette régularisation est aussi localement équivalente à l'arrêt prématuré de l'entraînement si on considère une approximation quadratique de la fonction objectif et on utilise l'AGS. Dans ces conditions, il est possible de montrer que le nombre d'itérations de gradient τ vérifie [19, 21]

$$\tau \approx \frac{1}{\varepsilon\alpha} \,. \tag{2.31}$$

Ainsi, le nombre d'itérations joue un rôle inversement proportionnel au facteur de régularisation L^2 et consiste en une source de variance. Un plus grand nombre d'itérations permet en effet aux paramètres θ d'explorer une plus large portion de l'espace des hypothèses ce qui augmente la capacité du modèle. Il est intéressant de noter aussi que la régularisation L^2 est équivalente à l'estimation du MAP lorsque l'on considère une distribution antérieure suivant une loi normale $N(0, \frac{1}{\lambda}\mathcal{I})$.

La régularisation L^1, correspondant à $\Omega(\theta) = \|\theta\|_1$, est également couramment utilisée dans le but d'obtenir des paramètres plus creux (i.e. contenant beaucoup de zéros) de manière à réduire le nombre de paramètres redondants. Le gradient de la fonction objectif régularisée devient alors

$$\nabla_\theta \widetilde{\mathcal{L}}(\theta) = \nabla_\theta \mathcal{L}(\theta) + \lambda \operatorname{sgn}(\theta) \,. \tag{2.32}$$

Ainsi, les paramètres ne contribuant que peu à la minimisation de la fonction objectif originale seront fortement poussés par le gradient du terme de régularisation à prendre une valeur nulle, puisqu'elle est le seul zéro de la fonction signe. Remarquons que la régularisation L^1 est aussi équivalente à l'estimation du MAP, cette fois-ci en considérant une distribution antérieure Laplacienne isotrope Laplace$(0, \frac{1}{\lambda}\mathcal{I})$.

CHAPITRE 3

APPRENTISSAGE PROFOND : RÉSEAUX DE NEURONES

L'apprentissage profond (*Deep Learning*) désigne la sous-classe de méthodes d'apprentissage automatique permettant la modélisation de données à un haut niveau d'abstraction grâce à des architectures profondes, articulées en une cascade de transformations non-linéaires successives formant alors une hiérarchie de représentations. Les algorithmes d'apprentissage profond sont structurés en réseaux de neurones artificiels, inspirés du fonctionnement des neurones biologiques, qui sont formés d'un graphe de neurones dont la topologie est appelée l'**architecture**. Ils peuvent être utilisés aussi bien pour l'apprentissage supervisé que pour l'apprentissage non supervisé et ont notamment permis des avancées majeures dans les domaines de la reconnaissance et la synthèse vocale et d'images.

Ce chapitre présente le fonctionnement des réseaux de neurones à propagation avant, et en particulier les couches denses et convolutives qui forment la base de leur architecture. Les dernières sections présentent les modèles génératifs profonds qui seront utilisés par la suite dans ce travail.

3.1 Perceptron et Perceptron Multicouche

Les réseaux de neurones artificiels sont inspirés des premiers travaux sur les neurones biologiques qui ont permis d'identifier les principales caractéristiques de la communication des neurones dans notre système nerveux et ont permis de les modéliser par un neurone formel, tel que représenté à la figure 3.1. Celui-ci représente le traitement de l'information reçue par un neurone des autres neurones auquel il est relié : les données d'entrée $\mathbf{x}$ représentent les dendrites (connexions entrantes d'un neurone) et la sortie y représente le cône d'émergence du neurone (point de départ de l'axone, la connexion

sortante d'un neurone). Les actions excitatrices et inhibitrices des synapses sont représentées par un vecteur de paramètres $\mathbf{w}$: ce sont les **poids synaptiques**, ou poids. Le traitement final du signal se fait par une fonction non linéaire appelée **fonction de transfert**. Ainsi, un neurone formel caractérisé par des poids $\mathbf{w}$ et une fonction d'activation ϕ calcule sa sortie selon

$$\hat{y} = \phi(\mathbf{w} \cdot \mathbf{x} + b) \,, \tag{3.1}$$

où b est un autre paramètre appelé le **seuil**, ou le **biais**. Un neurone biologique ne s'active en effet qu'à partir d'une valeur de seuil, et reste sinon inactif. Ceci a été retranscrit dans le neurone formel tel que proposé par McCulloch et Pitts [38] par l'utilisation de la fonction de Heaviside ou la fonction signe comme fonction d'activation.

Un seul neurone formel, muni d'une règle d'apprentissage, forme alors un algorithme simple de classification binaire. Le **perceptron**, inventé en 1957 par F. Rosenblatt [49], utilise un unique neurone formel avec la règle d'apprentissage $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta(\hat{y} - y)\mathbf{x}$, où y est la classe réelle, pour former un classifieur binaire.

Le perceptron peut être étendu de manière à obtenir une sortie vectorielle, pour une application à la classification multi-classes par exemple. On dispose alors d'une couche de neurones artificiels caractérisée par un ensemble de vecteurs de poids $\{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_H\}$ formant une matrice de poids $\mathbf{W}$ et un vecteur de biais $\mathbf{b} \in \mathbb{R}^H$, où H est la largeur de la couche. On utilise typiquement la même fonction d'activation pour tous les neurones d'une couche, celle-ci appliquant maintenant l'opération

$$\mathbf{y} = \phi(\mathbf{W} \cdot \mathbf{x} + \mathbf{b}) \,. \tag{3.2}$$

La couche de neurones ainsi formée acquière maintenant un caractère plus général que le perceptron : elle peut être appliquée pour tous les problèmes d'apprentissage supervisé et non supervisé et par-dessus tout devient un approximateur universel. En effet le **théorème d'approximation universelle** stipule qu'un réseau de neurones constitué d'une seule couche, avec une fonction d'activation bornée et strictement croissante, peut ap-

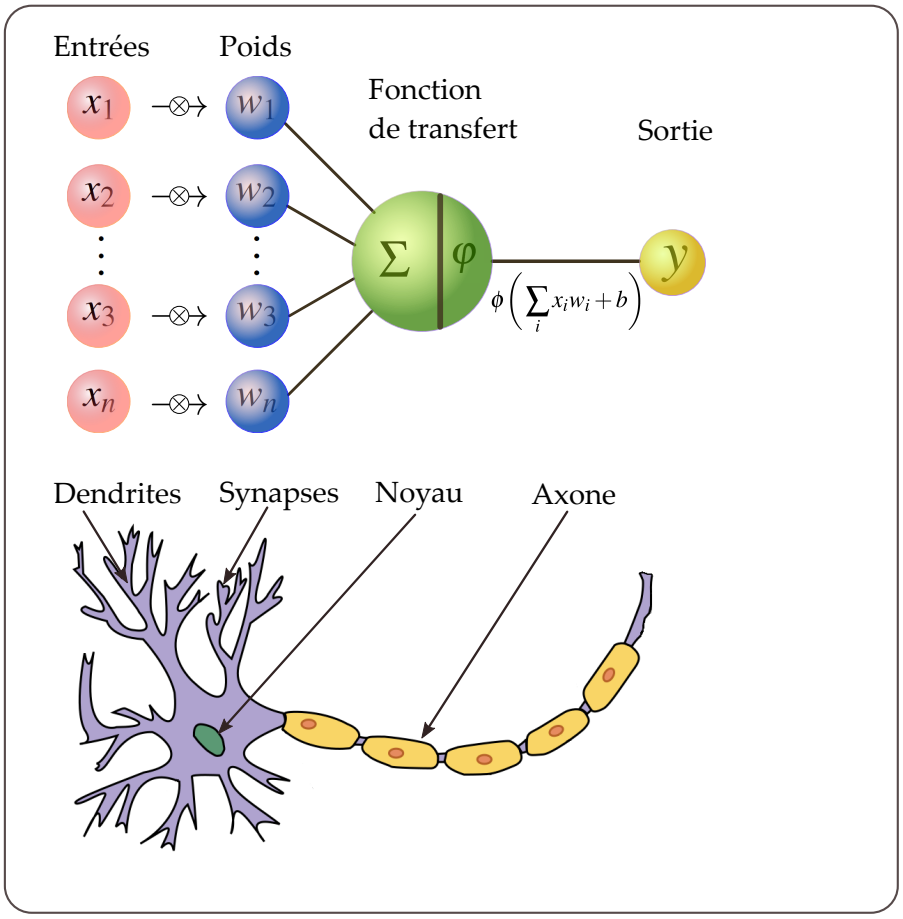

FIGURE 3.1 : Illustration du fonctionnement d'un neurone formel en analogie avec un neurone biologique. Les entrées correspondent aux dendrites, les poids aux synapses, la fonction de transfert au noyau et la sortie à l'axone.

proximer toute fonction mesurable[1] entre espaces de dimension finie avec une précision arbitraire, étant donné un nombre suffisant, mais fini, de neurones H.

Il est important de noter cependant que le théorème ne spécifie pas le nombre de neurones requis pour une telle approximation, qui pourrait être exponentiellement grand. De plus, même si un tel réseau de neurones peut en principe approximer une fonction donnée, il n'y a aucune garantie qu'il puisse *apprendre* cette représentation.

Un réseau de neurones profond est constitué d'une composition de couches de neu-

[1] Il est suffisant de considérer dans notre cas que toute fonction définie sur un sous-ensemble compact de $\mathbb{R}^n$ est mesurable.

rones successives ; il est d'usage de représenter les données d'entrée par une couche d'entrée (n'ayant aucun paramètre ni activation) et d'appeler les couches intermédiaires les couches cachées. On parle de réseaux à propagation avant, ou acycliques, pour des réseaux de neurones ne comportant pas de boucle de retour d'information (*feedback*) dans leur architecture, c'est-à-dire que les données d'entrée $\mathbf{x}$ ne sont pas réinjectées dans le réseau pendant la propagation[2].

La généralisation naturelle du perceptron est le **perceptron multicouche** (MLP, *Multilayer Perceptron*), qui forme donc le premier exemple de réseau de neurones profond à propagation avant. Il est formé d'un ensemble de couches $\{C_1, ..., C_D\}$, où D est la **profondeur du réseau**, de largeurs $\{H_1, ..., H_D\}$ et fonctions d'activation $\{\phi_1, ..., \phi_D\}$. La profondeur du réseau, la largeur et l'activation de chaque couche sont des hyperparamètres, qui avec le type de couche et de propagation forment l'architecture du réseau. La figure 3.2 représente l'architecture d'un Perceptron Multicouche. Chaque cellule correspond à un neurone, chacun portant un vecteur de paramètres $\mathbf{W}_i^{(l)}$, et les liens indiquent une connexion entre les neurones. On remarque que chaque neurone d'une couche est connecté à tous les neurones de la couche précédente et suivante.

L'architecture générique des réseaux de neurones à propagation avant est une séquence de couches de neurones, dont la connectivité définit le type de couche, chacune appliquant une transformation linéaire paramétrée suivie d'une fonction non-linéaire, le plus souvent non paramétrée et appliquée par composantes[3]. Un réseau de neurones à propagation avant peut donc être vu comme un graphe orienté acyclique dont les éléments sont des neurones formels et dont la topologie est déterminée par les opérations effectuées par chaque couche.

Si on dénote le tenseur de sortie de la couche (l) par $\mathbf{x}^{(l)}$, où $\mathbf{x}^{(0)}$ représente la couche

[2] Un exemple de réseaux incluant ce type de connexions sont les réseaux récurrents, dont il ne sera pas question dans ce mémoire.

[3] Certaines fonctions d'activation font cependant exception à cette règle, par exemple le noyau Gaussien.

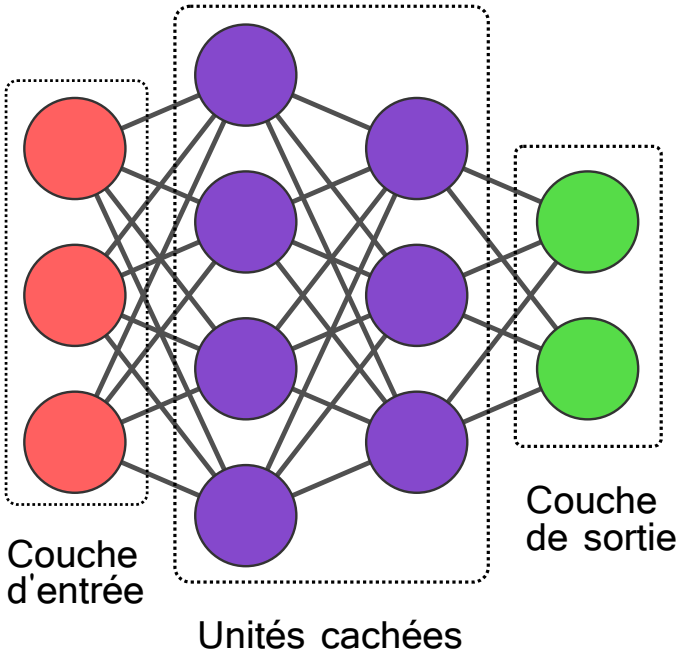

FIGURE 3.2 : Graphe d'un Perceptron Multicouche de profondeur $D = 2$: chaque vertex représente un neurone formel et chaque lien représente une connexion entre les neurones. Chaque neurone d'une couche est connecté à tous les neurones de la couche précédente.

d'entrée (qui ne fait que de passer les données d'entrée à la couche suivante), les paramètres de la couche par $\mathbf{W}^{(l)}, \mathbf{b}^{(l)}$, l'opération linéaire par $\star$ et la fonction non-linéaire par ϕ_l, la couche (l) effectuera alors la transformation

$$\mathbf{x}^{(l+1)} = \phi_l(\mathbf{W}^{(l)} \star \mathbf{x}^{(l)} + \mathbf{b}^{(l)}) \tag{3.3}$$

On appelle la pré-activation de la couche (l) le tenseur de sortie avant l'application de la fonction de transfert : $\mathbf{a}^{(l)} = \mathbf{W}^{(l)} \star \mathbf{x}^{(l)} + \mathbf{b}^{(l)}$. Chaque couche est alors déterminée par le rang et la dimension des tenseurs $\mathbf{W}^{(l)}, \mathbf{b}^{(l)}$, par l'opération linéaire $\star$ et par la fonction non-linéaire ϕ_l. L'architecture d'un réseau de neurones désigne alors l'ensemble de ces hyper-paramètres ainsi que tout hyper-paramètre additionnel déterminant la topologie du réseau. Remarquons qu'aucune restriction n'a été faite sur le rang des tenseurs en jeu dans 3.3 car selon l'architecture considérée le rang des tenseurs peut varier ; la notation restera donc la même pour des tenseurs et des vecteurs.

L'architecture du perceptron multicouche correspond à n'avoir que des couches ayant

pour opération linéaire le produit matriciel avec la matrice de poids $\mathbf{W}$, qu'on appelle des couches denses puisqu'elles correspondent à avoir chaque neurone de la couche (donc chaque colonne $\mathbf{W}_i$ de la matrice de poids) connecté à tous les neurones d'entrée. Ainsi, une couche dense ayant n_2 neurones prenant comme entrée un vecteur de dimension n_1 aura une matrice de poids de dimension (n_1, n_2) ce qui résulte en un vecteur de sortie de dimension n_2.

Les réseaux de neurones étant entraînés avec l'AGS, cette opération est faite en parallèle sur la dimension correspondant à un mini-lot de données. Une couche dense prend donc en entrée un tenseur de dimensions (b, n_1) (si le tenseur d'entrée est de rang supérieur à 2, toutes les dimensions supérieures à la deuxième sont aplaties sur cette dernière) et renvoie un tenseur de dimensions (b, n_2) en sortie, où b est la taille des mini-lots.

Une vaste gamme de non-linéarités peuvent être utilisées dans l'architecture des réseaux de neurones, chacune ayant sa spécificité. L'image de la fonction d'activation $\mathrm{Im}\phi$ sert souvent à déterminer le domaine des variables de sortie. Par exemple si le modèle doit prédire une probabilité, la couche de sortie sera munie d'une fonction d'activation ayant pour image $[0,1] \subset \mathbb{R}$. Le tableau 3.I présente les fonctions d'activation les plus communes en apprentissage profond ainsi que leurs propriétés et leurs applications.

Nom	$\phi(x)$	$\mathrm{Dom} f$	$\mathrm{Im} f$	Classe	Propriétés
Identité / Rampe	x	$\mathbb{R}$	$\mathbb{R}$	C^∞	MDI
Sigmoïde / Logistique	$\frac{1}{1+e^{-x}}$	$\mathbb{R}$	$(0,1)$	C^∞	MP
Tangente Hyperbolique	$\tanh(x)$	$\mathbb{R}$	$(-1,1)$	C^∞	MI
Gaussienne	e^{-x^2}	$\mathbb{R}$	$(0,1]$	C^∞	P
Rectifieur linéaire (ReLU)	$\max(0,x)$	$\mathbb{R}$	$\mathbb{R}^+$	C^0	MD
softplus	$\log(1+e^x)$	$\mathbb{R}$	$\mathbb{R}^+$	C^∞	MD
softmax	$\frac{e^{\mathbf{x}}}{\sum_i e^{x_i}}$	$\mathbb{R}$	$(0,1]$	C^∞	MDP

TABLEAU 3.I : Liste de certaines fonctions de transfert communes en apprentissage profond et de leurs propriétés. (M=monotone, D=dérivée monotone, I=approxime l'identité en 0, P=renvoie une probabilité)

3.2 Rétropropagation des gradients

Les réseaux à propagation avant sont entraînés par descente de gradient stochastique. Lors de la propagation avant, le réseau évalue la sortie $\mathbf{y}$ et calcule éventuellement la fonction objectif $\mathcal{L}(\theta)$[4]. La mise à jour des paramètres se fait par une propagation arrière des gradients $\nabla_\theta \mathcal{L}(\theta)$ selon l'algorithme de la **rétropropagation du gradient** [50].

Il s'agit d'un algorithme de différentiation automatique exploitant la règle de la dérivée en chaîne pour calculer efficacement les gradients numériques, en partant de la fonction objectif pour redescendre jusqu'à la première couche cachée. Ainsi l'algorithme de rétropropagation calcule les gradients de la fonction objectif de manière itérative

$$\begin{cases} \nabla_{\mathbf{W}^{(l)}} \mathcal{L} = (\nabla_{\mathbf{W}^{(l)}} \mathbf{x}^{(l+1)})^\intercal (\nabla_{\mathbf{x}^{(l+1)}} \mathbf{x}^{(l+2)})^\intercal \dots (\nabla_{\mathbf{W}^{(D-1)}} \mathbf{x}^{(D)})^\intercal \nabla_{\mathbf{x}^{(D)}} \mathcal{L}\,, \\ \nabla_{\mathbf{W}^{(l)}} \mathbf{x}^{(l+1)} = (\nabla_{\mathbf{a}^{(l)}} \phi_l)^\intercal \nabla_{\mathbf{W}^{(l)}} \mathbf{a}^{(l)}\,, \\ \nabla_{\mathbf{W}^{(l)}} \mathbf{a}^{(l)} = \mathbf{x}^{(l)}\,. \end{cases} \tag{3.4}$$

Dans la notation ci-dessus le produit scalaire et la transposition sont implicitement appliqués sur le sous-espace commun aux deux tenseurs : l'expression $(\nabla_{\mathbf{y}} \mathbf{x})^\intercal \nabla_{\mathbf{x}} \mathcal{L}$ implique que la transposition et le produit scalaire sont appliqués dans les dimensions associées à $\mathbf{x}$ des tenseurs $\nabla_{\mathbf{y}} \mathbf{x}$ et $\nabla_{\mathbf{x}} \mathcal{L}$.

L'équation 3.4 met en évidence un problème central dans l'entraînement des réseaux de neurones : le problème de la dissipation et de l'explosion du gradient. En effet, chaque paramètre a un gradient proportionnel à tous les gradients liés à celui-ci dans les couches supérieures du réseau : la connectivité du réseau joue donc un rôle important dans ces problèmes. Ainsi, si ces gradients sont trop petits ou trop grands, l'effet est magnifié sur la profondeur du réseau ce qui résulte en un apprentissage nul ou divergent respectivement.

La rétropropagation du gradient est implémentée en pratique grâce à la différentiation automatique, dont elle est le seul algorithme [19]. La différentiation automatique se

[4] Le vecteur θ sert maintenant de notation pour un ensemble contenant tous les tenseurs de paramètre d'un réseau de neurones. Ce n'est donc pas un vecteur à proprement parler, mais plutôt une notation commode.

distingue de la différentiation numérique ou symbolique. La différentiation numérique utilise la méthode des différences finies, qui est inefficace pour des fonctions de haute dimension et souffre de problèmes de précision et de stabilité numérique. La différentiation symbolique œuvre directement sur des expressions mathématiques et souffre aussi de problèmes d'efficacité.

Dans les librairies logicielles d'apprentissage profond la tendance a été d'utiliser des graphes symboliques pour décrire l'ensemble des opérations effectuées en conjonction avec la rétropropagation des gradients. Ceci permet à la librairie d'optimiser le graphe de calcul pour la stabilité numérique et la performance computationnelle. Chaque opérateur de calcul possède sa propre méthode définissant son gradient, qui est appelée lors de la rétropropagation. Ceci permet une grande flexibilité pour l'optimisation des calculs au prix de ne pouvoir utiliser que des opérateurs symboliques implémentés au préalable dans la librairie. Toutes les implémentations dans ce mémoire on été faites en Python à l'aide des libraires logicielles Theano et Lasagne [14, 56].

3.3 Réseaux convolutifs

Les réseaux neuronaux convolutifs (CNN, *Convolutional Neural Networks*), ou simplement réseaux convolutifs, sont des réseaux acycliques utilisant la convolution au lieu de la multiplication matricielle comme transformation linéaire. Leur fonctionnement est inspiré de celui des neurones biologiques du cortex visuel des animaux [37], dans lequel les neurones d'une couche ne sont connectés qu'à un sous-ensemble des neurones de la couche précédente, ce qui permet une spécialisation spatiale lors du pavage du champ visuel. Leur performance sur des données avec une topologie discrète, comme les séries temporelles (1D) et les images (2D), est inégalée.

La convolution discrète entre un vecteur de poids $\mathbf{w}$ (aussi appelé filtre dans le do-

maine du traitement de signal) et un vecteur d'entrée $\mathbf{x}$ est définie par

$$\begin{aligned}(\mathbf{x} * \mathbf{w})_i &= \sum_j \mathbf{x}_{i-j}\mathbf{w}_j \\ &= \sum_j \mathbf{x}_j \mathbf{w}_{i-j} \ ,\end{aligned} \tag{3.5}$$

et constitue donc l'opération linéaire d'une couche convolutive. La convolution est aussi commutative, associative et distributive, et correspond à un filtre linéaire de $\mathbf{x}$ par $\mathbf{w}$. En vertu du théorème de la convolution, elle peut être vue comme un produit dans l'espace de Fourier, c'est-à-dire que

$$(\mathbf{x} * \mathbf{w}) = f[\mathbf{x}]f[\mathbf{w}] \ , \tag{3.6}$$

où f dénote la transformée de Fourier. Ainsi les fréquences présentes dans le filtre permettent de sélectionner les fréquences dans le signal entrant.

Chaque élément de la convolution $(\mathbf{x} * \mathbf{w})_i$ est constitué du produit du filtre $\mathbf{w}$ avec un sous-ensemble de $\mathbf{x}$ que l'on appelle une tuile (on suppose ici que $\dim \mathbf{w} < \dim \mathbf{x}$). L'entrée $\mathbf{x}$ est donc pavée en petites zones, qui se chevauchent souvent pour obtenir une meilleure représentation de la structure de l'entrée, qui sont traitées individuellement par un neurone formel. L'utilisation du même neurone formel pour toutes les tuiles de l'entrée correspond donc à une opération de convolution. Le **champ récepteur** d'un neurone correspond à la taille de la tuile du signal entrant auquel il est sensible.

Chaque filtre correspondant à un neurone formel, on en déduit qu'une couche convolutive se distingue d'une couche dense du fait que chaque neurone n'est connecté qu'à un sous-ensemble des données d'entrées, c'est-à-dire que chaque neurone se spécialise sur une région spatiale limitée. L'application complète d'un filtre sur une entrée est donc un ensemble de neurones qui partagent les mêmes paramètres appelée le **noyau de convolution** (*feature map*). Une couche convolutive est typiquement formée de plusieurs noyaux, c'est-à-dire de plusieurs filtres dont l'application est insérée dans une dimension supplémentaire du tenseur de sortie appelée la dimension des canaux, par analogie aux canaux de couleur des images RVB (rouge, vert, bleu).

Les réseaux convolutifs sont donc sensibles à la structure locale du signal entrant tout en réduisant le nombre de paramètres par rapport aux réseaux denses. Ils s'en distinguent particulièrement par les traits suivants :

1. **Connectivité locale** : Le champ récepteur limité des couches convolutives permet de produire une réponse accentuée sur la structure locale associée à chaque filtre du signal entrant. La représentation obtenue est aussi plus légère en mémoire, avec un nombre de paramètres réduit. Ceci conduit à une entraînement plus robuste pour un volume de données fixé.

2. **Poids partagés** : Dans un même noyau, les neurones partagent les mêmes paramètres i.e. le même filtre. Il y a cependant un filtre différent pour chaque canal entrant.

3. **Invariance par translation** : Les neurones d'un même noyau étant identiques, le motif détecté est indépendant de sa localisation dans le signal entrant.

La figure 3.3 illustre le fonctionnement d'un réseau convolutif lorsqu'appliqué à une image. On remarque que chaque neurone d'une couche ne voit qu'un sous-ensemble des données entrantes, ce qui illustre le concept de connectivité locale. Il est courant dans les réseaux convolutifs d'utiliser l'unité de rectification linéaire (ReLU) comme fonction de transfert car elle ne souffre pas de problèmes de saturation numérique et permet de sélectionner efficacement les caractéristiques utiles à la minimisation de la fonction objectif.

Si un réseau convolutif est entraîné pour la classification d'images, il est souhaitable de réduire les dimensions spatiales des instances le long du traitement de l'image par le réseau tout en augmentant le nombre de filtres. Ceci permet au réseau d'apprendre à reconnaître un grand nombre des caractéristiques spatialement réduites (e.g. des yeux, une bouche, un nez, etc...), qui fournissent une représentation commode à la classification de l'image entrante. La dimension du tenseur de sortie d'une couche convolutive est contrôlée par trois hyper-paramètres :

1. **Profondeur de la couche** : nombre de noyaux de convolutions.

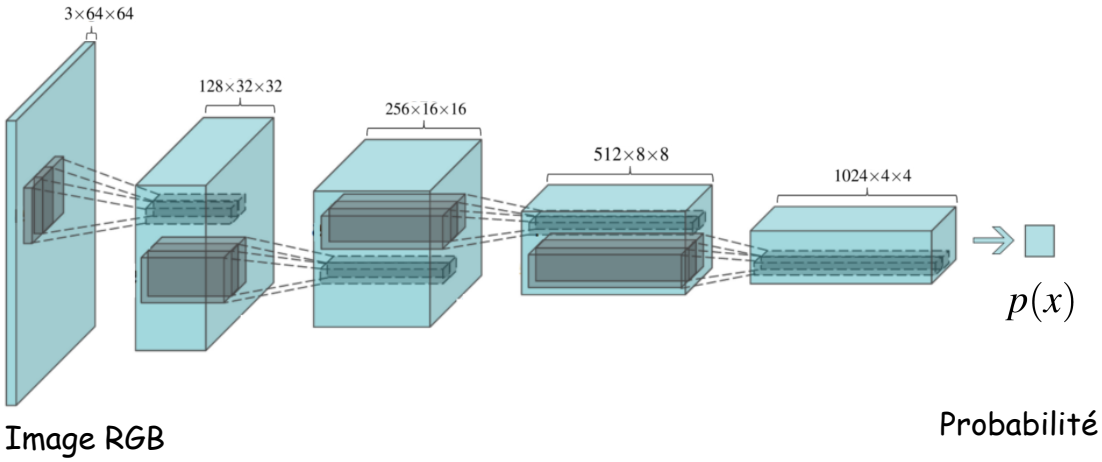

FIGURE 3.3 : Schéma de l'application de quatre couches convolutives à une image. La largeur correspond à la dimension des canaux, tandis que les dimensions transversales correspondent aux dimensions spatiales de l'image. Chaque couche convolutive est appliquée avec un pas $s = 2$, ce qui réduit les dimensions spatiales de l'image. La sortie de chaque couche est un tenseur de rang 3 dont les dimensions sont (k,n,n) où k est le nombre de filtres appliqué et n la dimension spatiale de sortie de l'image. Si le réseau est entraîné par mini-lots de données, ces opérations sont faites en parallèle, et chaque mini-lot d'images est un tenseur de rang 4 de dimensions (b,k,n,n), où b est la taille du mini-lot.

2. **Pas** : contrôle le chevauchement des champs réceptifs. Plus le pas est petit, plus les champs réceptifs se chevauchent et le volume de sortie sera grand.

3. **Marge** à 0 : nombre de zéros insérés de part et d'autre du signal entrant. Cela permet de contrôler la dimension spatiale du volume de sortie (il est parfois souhaitable de conserver la même dimension que celle d'entrée).

Lorsque le signal entrant est unidimensionnel (comme une série temporelle), il est représenté par un mini-lot de dimensions (b,c,n). Alors, une convolution avec avec d filtres de convolution de taille k, un pas s et une marge p aura une dimension de sortie (b,d,o) où o est la dimension spatiale de sortie de la convolution qui est donnée par

$$o = \frac{n-k+2p}{s} + 1 \,. \tag{3.7}$$

Cette formule s'applique séparément pour toute dimension spatiale additionnelle, en pre-

nant en compte le pas, la marge et la dimension des filtres dans chaque dimension (qui sont habituellement les mêmes). Il est aussi possible de calculer le champ réceptif d'une couche convolutive (l) à partir de la couche précédente par

$$\begin{aligned} j_l &= j_{l-1} s_l \\ r_{l+1} &= r_l + (k_{l+1} - 1) j_l \ , \end{aligned} \tag{3.8}$$

où j_l est le saut entre deux éléments du signal d'entrée fait entre deux neurones adjacents de la couche (l).

Les couches convolutives présentent donc plusieurs avantages par rapport aux couches denses. Lorsque le tenseur d'entrée présente une forte structure locale, c'est-à-dire de fortes corrélations entre des entrées voisines comme le cas des pixels d'une image, une couche dense doit apprendre ces corrélations plusieurs fois, pour chaque recouvrement du tenseur d'entrée, ce qui implique une forte redondance.

Une couche convolutive évite cette redondance puisqu'un seul filtre passé sur le tenseur d'entrée peut détecter la même structure à différents endroits. Ceci se traduit donc en un forte réduction du nombre de paramètres et une plus grande efficacité. Réduire le nombre de paramètres, a fortiori les paramètre redondants, permet de prévenir le surapprentissage mais aussi de faciliter l'apprentissage en baissant la dimension de l'espace ou prend place l'optimisation. Les couches convolutives peuvent être utilisées en conjonction avec les couches denses dans une vaste gamme d'architectures de réseaux de neurones. Elles sont particulièrement adaptées au traitement de données présentant une forte structure spatiale, comme les images, l'audio, où même le silicium amorphe lorsqu'on a affaire à un espace discrétisé.

3.4 Modèles génératifs profonds à variables latentes

La modélisation générative est un domaine de l'apprentissage automatique qui traite de l'approximation d'une distribution $p(\mathbf{x})$ définie sur des données $\mathbf{X}$ dans un espace à haute dimension $\mathcal{X}$. Cette distribution peut être, par exemple, la distribution des positions

atomiques du silicium amorphe. Il s'agit d'une tâche d'apprentissage non supervisé dont le but est de capturer les dépendances entre les différentes variables formant les données $\mathbf{x}$, de manière à pouvoir déterminer la probabilité qu'une observation appartienne à la distribution cible et de générer des nouvelles observations y appartenant.

Le développement de modèles génératifs a longtemps été un défi dans le domaine de l'apprentissage automatique car la plupart des approches souffrent de problèmes sévères. Ces modèles demandent souvent de faire de hypothèses fortes sur la nature des données ainsi que des approximations grossières ce qui mène éventuellement à des modèles non optimaux. Aussi, des techniques d'inférence inefficaces telles que les méthodes Monte Carlo par chaînes de Markov sont parfois nécessaires. Cependant les réseaux de neurones forment des approximateurs de fonctions très efficaces lorsqu'entraînés par descente de gradient avec l'algorithme de rétropropagation, qualité qui en fait des candidats idéaux pour le développement de modèles génératifs.

Les modèles génératifs profonds les plus populaires, de par leur succès dans des applications variées de génération de données complexes en haute dimension, sont les Réseaux Accusatoires Génératifs (*Generative Adversarial Networks*, GAN) et les Auto-Encodeurs Variationnels (*Variational Autoencoders*, VAE). Ils se sont tous deux démarqués dans les dernières années de par leur grande capacité à générer des données réalistes comme des images représentant des objets de la vie quotidienne, des animaux ou des visages, ainsi que de la synthèse vocale ou même des modèles physiques de scènes [33, 36, 46].

Les GAN et les VAE sont des modèles dits à variables latentes[5]. Il s'agit de variables qui ne sont pas observées, mais plutôt inférées par un modèle dans le but d'expliquer les variables observées. Ces variables explicatives servent au modèle génératif à décider quelles vont être les caractéristiques importantes des données qu'il faut produire ou analyser. Avant que l'on puisse dire qu'un modèle est représentatif d'un ensemble de

[5] Du Latin *lateo*, «qui reste caché», par opposition aux variables observées.

données, il faut s'assurer que pour chaque point de l'ensemble $\mathbf{x}$ il y a une ou plusieurs conformations de variables latentes qui causent le modèle à générer un point très proche de $\mathbf{x}$.

Formellement, soit un vecteur de variables latentes $\mathbf{z}$ dans un espace à haute dimension $\mathcal{Z}$ suivant une distribution $p(\mathbf{z})$ dont on peut aisément tirer. Soit aussi une famille de fonctions déterministes $f_\theta(\mathbf{z})$ paramétrées par un vecteur $\theta \in \Theta$ fixé, où $f : \mathcal{Z} \times \Theta \to \mathcal{X}$. Alors $f_\theta(\mathbf{z})$ est une variable aléatoire dans $\mathcal{X}$. On désire alors optimiser θ tel que lorsque $\mathbf{z} \sim p(\mathbf{z})$, avec une grande probabilité, $f_\theta(\mathbf{z})$ sera proche des données de $\mathbf{X}$.

Ainsi, les deux modèles génératifs profonds à variables latentes considérés dans ce mémoire basent leur procédure générative sur l'entraînement d'un réseau de neurones profond f_θ pour approximer la distribution conditionnelle $p(\mathbf{x}|\mathbf{z})$.

3.5 Réseaux Accusatoires Génératifs (GAN)

Les réseaux accusatoires génératifs sont des modèles génératifs profonds qui consistent à mettre en compétition deux fonctions, un générateur et un discriminateur, dans un scénario de théorie des jeux, tel qu'illustré à la figure 3.4.

Le **générateur** est un réseau de neurones g_θ de paramètres θ qui génère des données à partir de variables latentes $\mathbf{z} \in \mathcal{Z}$ issues d'une distribution antérieure $p_\mathcal{Z}$, souvent prise uniforme ou normale. Il est entraîné à faire converger sa distribution $p_G = G_\theta(\mathbf{x}|\mathbf{z})$[6] vers la distribution des données $p_\mathcal{X}$.

Le **discriminateur** est un réseau de neurones f_ω de paramètres ω qui estime la probabilité qu'une observation provienne de la distribution des données $p_\mathcal{X}$, dont $\mathbb{D}$ est un

[6] On notera les fonctions paramétrées, comme les réseaux de neurones, par des lettres minuscules, comme $g_\theta(\mathbf{z})$ et $f_\omega(\mathbf{x})$ et la distribution de probabilité y étant associée par des lettres majuscules, $G_\theta(\mathbf{x}|\mathbf{z})$ et $F_\omega(p|\mathbf{x})$ respectivement.

ensemble de réalisations, plutôt que de la distribution du générateur p_G [18, 20].

$$\begin{aligned} g_\theta(\mathbf{z}) &: \mathcal{Z} \to \mathcal{X} \\ f_\omega(\mathbf{x}) &: \mathcal{X} \to [0,1]\ . \end{aligned} \tag{3.9}$$

Ainsi, f_ω et g_θ sont des fonctions différentiables par rapport à leurs paramètres θ, ω qui sont donc entraînés par descente de gradient. Leur fonction objectif est définie en termes des deux joueurs : le discriminateur minimise $\mathcal{L}_D(\theta, \omega)$ et le générateur minimiser $\mathcal{L}_G(\theta, \omega)$, et chacun ne peut le faire qu'en modifiant ses propres paramètres. Ainsi, la solution à ce problème d'optimisation, ou plutôt de théorie des jeux, est un point de selle appelé l'équilibre de Nash [20, 47]. Il consiste en un couple (θ^*, ω^*) qui est un minimum local de $\mathcal{L}_D$ par rapport à ω et un minimum local de $\mathcal{L}_G$ par rapport à θ. L'entraînement d'un GAN se fait par l'AGS alternativement entre f_ω et g_θ. À chaque

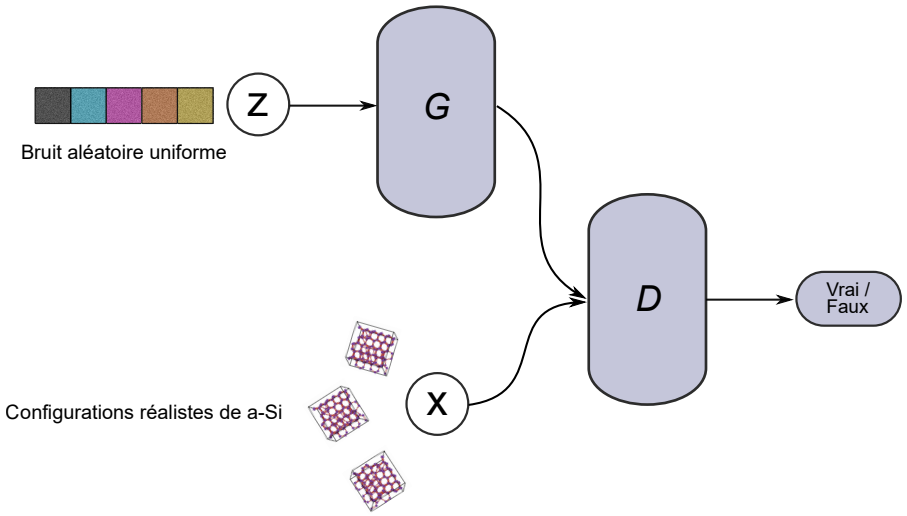

FIGURE 3.4 : Schéma du fonctionnement d'un GAN pour la génération de configurations atomiques de a-Si. Le générateur G transforme le bruit aléatoire entrant en une configuration atomique et essaye de tromper le discriminateur D. Le discriminateur essaye de différentier les configurations réelles (de l'ensemble d'entraînement) des configurations fausses (générées).

itération, un mini-lot d'observations $\mathbf{x}$ et de variables latentes $\mathbf{z}$ sont tirés de $\mathbb{D}$ et $p_\mathcal{Z}$ respectivement. L'AGS est alors appliqué simultanément pour minimiser $\mathcal{L}_D$ par rapport à ω et $\mathcal{L}_G$ par rapport à θ. L'algorithme d'entraînement d'un GAN est présenté dans l'al-

gorithme 1 de l'annexe II. Dans la formulation originale du GAN, la fonction objectif est $V = \mathcal{L}_G = -\mathcal{L}_D$

$$V(\theta, \omega) = \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\log f_\omega(\mathbf{x})] + \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[1 - \log f_\omega(g_\theta(\mathbf{z}))] \,. \tag{3.10}$$

Dans ce cas l'entraînement est un jeu à somme nulle, aussi appelé un jeu minimax du fait que les paramètres optimaux du discriminateur et du générateur mettent en jeu une minimisation ainsi qu'une maximisation[7]

$$\theta^* = \arg\min_\theta \max_\phi V(\theta, \phi) \tag{3.11}$$

$$\phi^* = \arg\max_\phi \min_\theta V(\theta, \phi) \,. \tag{3.12}$$

Pour un générateur fixe g_θ, le discriminateur optimal est

$$f^*(\mathbf{x}) = \frac{p_\mathcal{X}(\mathbf{x})}{p_\mathcal{X}(\mathbf{x}) + p_G(\mathbf{x})} \,, \tag{3.13}$$

qui se réduit à $f^*(\mathbf{x}) = \frac{1}{2}$ lorsque $p_G = p_\mathcal{X}$ et qui est donc l'équilibre de Nash du jeu minimax. Il est alors possible de montrer que lorsque le discriminateur est optimal, le minimum de $V(\theta, \omega)$ est $-\log 4$ et que

$$V(\theta, \omega^*) = D_{JS}(p_\mathcal{X} || p_G) - \log 4 \,, \tag{3.14}$$

où D_{JS} est la divergence de Jensen-Shannon, qui est toujours non négative et vaut zéro si et seulement si les deux distributions sont égales. Ceci implique que lorsque le discriminateur est optimal, le minimum global de l'objectif $\mathcal{V}$ est unique et correspond au cas où le générateur modélise parfaitement la distribution des données i.e. $p_G = p_\mathcal{X}$ [18, 20].

La fonction objectif du GAN a été généralisée de manière à correspondre à la mini-

[7] Remarquons ici que, comme dans la plupart des réseaux de neurones, les fonctions objectif sont hautement non convexes. Il n'y a aucune garantie de convergence vers un minimum, où un point de selle comme l'équilibre de Nash, avec l'algorithme du gradient pour ces fonctions.

misation variationnelle de toute une famille de divergences, les $f-$divergences, définies par

$$D_f(p||q) = \int_\Omega q(x) f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x \,, \tag{3.15}$$

où $f : \mathbb{R}^+ \to \mathbb{R}$ est une fonction convexe telle que $f(1) = 0$, et les distributions p et q sont définies sur Ω [43]. Par exemple, l'utilisation d'une fonction objectif quadratique résulte en la minimisation de la divergence χ^2 de Pearson entre $p_G + p_\mathcal{X}$ et $2p_G$ lorsque le discriminateur est optimal. Le modèle résultant est le *Least Squares GAN*, ou LSGAN, dont les fonctions objectif sont

$$\begin{aligned} \mathcal{L}_D &= \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[(f_\omega(\mathbf{x}) - b)^2] + \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[(f_\omega(g_\theta(\mathbf{z})) - a)^2] \\ \mathcal{L}_G &= \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[(f_\omega(g_\theta(\mathbf{z})) - c)^2] \,. \end{aligned} \tag{3.16}$$

L'équivalence avec la divergence χ^2 de Pearson est atteinte lorsque $b-c = 1$ et $b-a = 2$. Cette implémentation du GAN permet un entraînement plus stable et résulte en des modèles génératifs de meilleure qualité que la formulation originale [36].

Une avancée importante aussi bien dans la fondation théorique des GAN que dans leur performance pratique est le développement du Wasserstein GAN [2]. Il s'agit d'une modification de la formulation du GAN de manière à minimiser, sous certaines conditions, la distance de Wasserstein entre $p_\mathcal{X}$ et p_G définie par

$$W(p_\mathcal{X}, p_G) = \inf_{\gamma\in\prod(p_\mathcal{X}, p_G)} \mathbb{E}_{(x,y)\sim\gamma}\|x-y\| \,, \tag{3.17}$$

où $\prod(p_\mathcal{X}, p_G)$ est l'ensemble de toutes les distributions jointes $\gamma(x,y)$ dont les marginaux sont respectivement $p_\mathcal{X}$ et p_G. Intuitivement, $\gamma(x,y)$ indique combien de masse doit être transportée de x à y de manière à transformer $p_\mathcal{X}$ en p_G, la distance de Wasserstein correspond alors au coût du transport optimal de cette masse.

La distance de Wasserstein présente des propriétés avantageuses pour l'optimisation

par rapport aux $f-$divergences. Si on considère un générateur G_θ muni d'une distribution antérieure $p(\mathbf{z})$ vérifiant $\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\|\mathbf{z}\|] < \infty$, les propriétés suivantes s'appliquent à $W(p_\mathcal{X}, p_G)$, mais pas aux $f-$divergences :

- Si g_θ est continu par rapport à θ, $W(p_\mathcal{X}, p_G)$ l'est aussi.
- Si g_θ est localement Lipschitz, $W(p_\mathcal{X}, p_G)$ est continu partout et différentiable presque partout.
- $W(p_\mathcal{X}, p_G) \to 0$ implique la convergence en distribution $p_\mathcal{X} \to p_G$.

Ces propriétés indiquent que la distance de Wasserstein est une fonction objectif mieux adaptée à l'apprentissage de distributions dont le support est contraint à des variétés de basse dimension [2].

L'infimum en 3.17 est intraitable en pratique, mais la dualité de Kantorovich-Rubinstein indique que

$$W(p_\mathcal{X}, p_G) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim p_\mathcal{X}}[f(x)] - \mathbb{E}_{x\sim p_G}[f(x)] \,, \tag{3.18}$$

où le supremum est pris sur toutes les fonctions 1-Lipschitz $f : \mathbf{X} \to \mathbb{R}$. Si on considère plutôt le supremum sur les fonctions K-Lipschitz $\|f\|_L \leq K$ on obtient $K \cdot W(p_\mathcal{X}, p_G)$. Alors, si l'on dispose d'une famille de fonctions paramétrées $\{f_\omega\}_{\omega\in\mathcal{W}}$, qui est dans ce cas un réseau de neurones, la résolution du problème

$$\max_{\omega\in\mathcal{W}} \mathbb{E}_{x\sim p_\mathcal{X}}[f_\omega(x)] - \mathbb{E}_{z\sim p(z)}[f_\omega(g_\theta(z))] \tag{3.19}$$

résulte en un calcul de $W(p_\mathcal{X}, p_G)$ à une constante multiplicative près. Ceci implique que si on dispose d'un discriminateur (appelé un *critique* dans le cas d'un WGAN du fait qu'il doit nécessairement avoir une couche de sortie linéaire pour que $f_\omega : \mathbf{X} \to \mathbb{R}$) f_ω qui est K-Lipschitz et d'un générateur g_θ, l'équation 3.19 définit l'objectif du critique qui est donc entraîné à estimer la distance de Wasserstein entre $p_\mathcal{X}$ et p_G.

La formulation des fonctions objectif du WGAN est alors

$$\mathcal{L}_f(\omega,\theta) = \mathbb{E}_{z\sim p(z)}[f_\omega(g_\theta(z))] - \mathbb{E}_{x\sim p_\mathcal{X}}[f_\omega(x)] \tag{3.20}$$

$$\mathcal{L}_g(\omega,\theta) = -\mathbb{E}_{z\sim p(z)}[f_\omega(g_\theta(z))]\,. \tag{3.21}$$

Ainsi, le critique est entraîné à établir une frontière de décision entre les données *fausses* (négatives) et *vraies* (positives) au lieu d'estimer les probabilités correspondantes comme dans la formulation originale du GAN. Comme le critique approxime une la distance de Wasserstein, il doit avoir une sortie linéaire (c'est-à-dire une fonction de transfert linéaire à sa dernière couche).

Il reste cependant à développer une stratégie pour s'assurer que le critique f_ω soit Lipschitz. Une solution est simplement d'imposer aux paramètres ω d'être contenus dans un intervalle compact $[-c,c]^{\dim\omega}$, puisque toute fonction définie sur un intervalle compact est Lipschitz. Cependant, d'autres stratégies plus efficaces ont été développées, comme l'ajout d'un terme de régularisation à $\mathcal{L}_f$

$$\lambda\,(\max\{0,\|\nabla f_\omega(\hat{\mathbf{x}})\|-1\})^2\,, \tag{3.22}$$

où λ est un hyper-paramètre contrôlant le poids relatif de la régularisation et $\hat{\mathbf{x}}$ est une instance issue de l'interpolation entre un exemple d'entraînement et un exemple généré. Cette pénalité force directement la condition sur f_ω d'être Lipschitz.

Le WGAN présente empiriquement un entraînement et une convergence plus stables et de plus la distance de Wasserstein estimée par le critique est corrélée empiriquement avec la qualité des observations générées par g_θ. Ces propriétés en font donc une architecture de choix pour le développement d'un modèle génératif sur une base de données complexes, et il sera donc appliqué à la modélisation de la structure du silicium amorphe. L'implémentation du WGAN utilisé dans ce travail est présentée dans l'algorithme 2 de l'annexe II.

3.6 Auto-Encodeurs Variationnels (VAE)

Les Auto-Encodeurs Variationnels [30] sont des modèles génératifs profonds qui permettent d'apprendre à découpler les facteurs explicatifs de la distribution des données $p_{\mathcal{X}}$. Ils sont entraînés à maximiser la vraisemblance de l'ensemble de données sous le processus génératif, c'est-à-dire que si le générateur est un réseau de neurones paramétré g_θ de distribution $G_\theta(\mathbf{x}|\mathbf{z})$, le VAE cherche à maximiser $\forall \mathbf{x} \in \mathbb{D}$

$$p(\mathbf{x}) = \int_{\mathcal{Z}} g_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) \mathrm{d}\mathbf{z} \,, \tag{3.23}$$

où ici encore la distribution des variables latentes $p_{\mathcal{Z}} = p(\mathbf{z})$ est choisie comme étant une distribution simple, typiquement Gaussienne. L'idée de base du VAE est de définir encore une fois deux réseaux de neurones dans le but trouver une description latente $\mathbf{z}$ des données $\mathbf{x}$ suivant la loi $p_{\mathcal{Z}}$.

L'encodeur est un réseau de neurones q_ϕ de distribution $Q_\phi(\mathbf{z}|\mathbf{x})$ et de paramètres ϕ qui encode une observation $\mathbf{x}$ de $\mathbb{D}$ vers une variable latente $\mathbf{z}$.

$$q_\phi : \mathcal{X} \to \mathcal{Z} \tag{3.24}$$

Il est entraîné à faire correspondre la distribution des variables latentes à une distribution normale $\mathcal{N}(\mu, \sigma)^2$, où μ et σ sont les vecteurs de sortie de q_ϕ et où σ est une matrice diagonale ou un scalaire. Ceci implique que la description latente est forcée à être décorrélée. Il ne s'agit pas d'une approximation considérable du fait qu'il est toujours possible, au travers d'une fonction suffisamment complexe, de transformer des variables corrélées en de nouvelles variables décorrélées.

Le décodeur est un réseau de neurones g_θ de distribution $G_\theta(\mathbf{x}|\mathbf{z})$ qui décode le vecteur latent $\mathbf{z}$ en une reconstruction $\tilde{\mathbf{x}}$.

$$g_\theta : \mathcal{Z} \to \mathcal{X} \tag{3.25}$$

Il est entraîné à reconstruire les données $\mathbf{x}$ à partir des variables latentes $\mathbf{z}$ le mieux pos-

sible. Il est donc nécessaire de disposer d'une mesure de distance sur $\mathcal{X}$ pour pouvoir estimer l'erreur de reconstruction du décodeur.

A la fin de l'entraînement, l'encodeur transformera effectivement les données $\mathbf{x}$ vers une distribution Gaussienne et le décodeur sera en mesure de reconstruire celles-ci à partir du vecteur latent $\mathbf{z}$. Il est alors possible de générer de nouvelles observations appartenant à $p_{\mathcal{X}}$ en fournissant au décodeur un vecteur latent issu de $p_{\mathcal{Z}}$. Comme les variables latentes sont décorrélées, les VAE permettent de découpler efficacement les caractéristiques des données d'entrée de manière automatique. Par exemple, un VAE entraîné sur des images de visages apprend à découpler efficacement des caractéristiques comme le genre ou l'expression.

La fonction objectif que minimise le VAE est une borne inférieure variationnelle de la vraisemblance des données $p(\mathbf{x})$. En effet, considérons la divergence de Kullback-Leiber (KL) du modèle d'inférence $Q_\phi(\mathbf{z}|\mathbf{x})$ sur la distribution postérieure du modèle génératif $G_\theta(\mathbf{z}|\mathbf{x})$)

$$D_{KL}(Q_\phi(\mathbf{z}|\mathbf{x})||G_\theta(\mathbf{z}|\mathbf{x})) = \mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}\left[\log Q_\phi(\mathbf{z}|\mathbf{x}) - \log G_\theta(\mathbf{z}|\mathbf{x})\right] . \tag{3.26}$$

Alors, en appliquant la loi de Bayes et en réarrangeant les termes on obtient

$$\log p(\mathbf{x}) = \mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}\left[\log G_\theta(\mathbf{x}|\mathbf{z})\right] - D_{KL}(Q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) + D_{KL}(Q_\phi(\mathbf{z}|\mathbf{x})||G_\theta(\mathbf{z}|\mathbf{x})) \tag{3.27a}$$

$$\geq \mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}\left[\log G_\theta(\mathbf{x}|\mathbf{z})\right] - D_{KL}(Q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) = \mathcal{L}(\theta,\phi,\mathbf{x}) . \tag{3.27b}$$

Ainsi, la maximisation de la log-vraisemblance se fait par le biais d'une approche variationnelle consistant à minimiser la borne inférieure $\mathcal{L}(\theta,\phi,\mathbf{x})$. Le terme de différence $D_{KL}(Q_\phi(\mathbf{z}|\mathbf{x})||G_\theta(\mathbf{z}|\mathbf{x}))$ pousse Q_ϕ à produire des variables latentes $\mathbf{z}$ qui peuvent reproduire l'entrée $\mathbf{x}$. Bien qu'il soit absent dans la fonction objectif du VAE, il est petit si Q_ϕ est de suffisamment grande capacité.

Le terme $\mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}\left[\log G_\theta(\mathbf{x}|\mathbf{z})\right]$ est le terme de **reconstruction** en ce qu'il est minimal lorsque les données générées par P_θ à partir des variables latentes inférées par Q_ϕ sont égales aux entrées $\mathbf{x}$.
Le terme $-D_{KL}(Q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$ est le terme de **régularisation** puisqu'il pousse la distribution des variables latentes $Q_\phi(\mathbf{z}|\mathbf{x})$ à suivre la loi $p(\mathbf{z})$. L'architecture du VAE est illustrée à la figure 3.5.

Pour pouvoir appliquer la descente de gradient stochastique sur l'objectif 3.27b, il est nécessaire d'estimer les espérances en jeu dans sa définition.

En choisissant $Q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu(\mathbf{x};\phi),\Sigma(\mathbf{x};\phi))$ où μ,Σ sont les vecteurs de sortie d'un réseau de neurones profond, le terme de régularisation devient alors une divergence KL entre deux distributions Gaussiennes multivariées qui a une forme close

$$D_{KL}(\mathcal{N}(\mu,\Sigma)||\mathcal{N}(0,1)) = \frac{1}{2}\left(\mathrm{Tr}\,\Sigma + \mu^T\mu - \log\det\Sigma - \dim\mu\right). \tag{3.28}$$

Le terme de reconstruction est approximé en remplaçant l'espérance sur $\mathbf{x}$ par la moyenne de Q_ϕ sur un mini-lot de données, comme il est habituel lorsque l'on effectue la descente de gradient stochastique. Cependant, il est impossible d'effectuer la rétropropagation des gradients à travers la distribution stochastique $\mathcal{N}(\mu,\Sigma)$ puisque le gradient d'un processus aléatoire est indéfini. Une technique utilisée pour contrer ce problème, et rendre l'entraînement du VAE possible, consiste à échantillonner de $\mathcal{N}(\mu,\Sigma)$ en prenant $\mathbf{z} = \mu + \varepsilon\Sigma$ où $\varepsilon \sim \mathcal{N}(0,1)$. Ce reparamétrage permet d'appliquer la rétropropagation des gradients à travers l'auto-encodeur et rend donc l'entraînement du VAE possible.

L'objectif à minimiser du VAE est donc

$$\mathbb{E}_{\mathbf{x}\sim\mathbb{D}}\left[\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,1)}[\log G_\theta(\mathbf{x}|\mathbf{z}=\mu+\varepsilon\Sigma)] - D_{KL}(Q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))\right]. \tag{3.29}$$

L'implémentation du VAE est présentée dans l'algorithme 3 de l'annexe II.

L'Auto-Encodeur de Wasserstein (WAE) [57] est une extension du VAE utilisant une

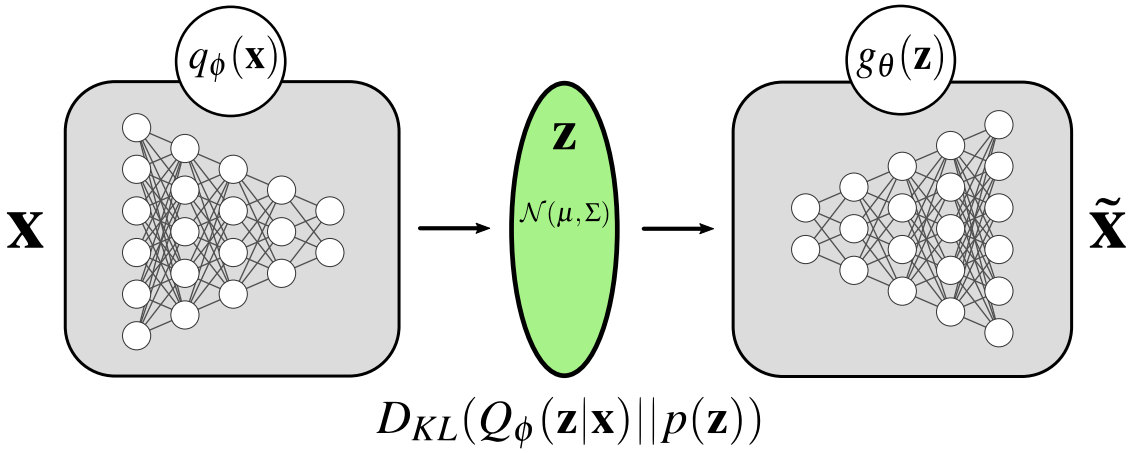

FIGURE 3.5 : Architecture du VAE. L'encodeur $q_\phi(\mathbf{x})$ projette les données $\mathbf{x}$ vers une description latente $\mathbf{z}$. Il est entraîné à faire correspondre la distribution de ces dernières $Q_\phi(\mathbf{z}|\mathbf{x})$ à une distribution antérieure $p(\mathbf{z})$ (typiquement $\mathcal{N}(0,1)$. Le décodeur $g_\theta(\mathbf{z})$ reconstruit ces données à partir de $\mathbf{z}$. Il est entraîné à minimiser l'erreur de reconstruction entre $\mathbf{x}$ et $\tilde{\mathbf{x}}$.

régularisation qui minimise une forme pénalisée de la distance de Wasserstein entre la distribution antérieure $p(\mathbf{z})$ et la distribution latente du modèle $Q_\phi(\mathbf{z}|\mathbf{x})$. Cette minimisation est effectuée par le biais d'un processus accusatoire comme dans le cas du WGAN.

L'architecture du WAE est composée d'un encodeur q_ϕ et d'un décodeur g_θ comme dans le VAE ainsi que d'un discriminateur latent d_γ.
Le discriminateur est entraîné à discerner les variables latentes du modèle $Q_\phi(\mathbf{z}|\mathbf{x})$ des variables issues de la distribution antérieure $\mathbf{z} \sim p(\mathbf{z})$, c'est-à-dire à classifier les premières comme *fausses* et les secondes comme *vraies* précisément comme dans un GAN.

$$\mathcal{L}_D = \frac{\lambda}{N}\sum_{i=1}^{N} \log d_\gamma(\mathbf{z}_i) + \log\left(1 - d_\gamma\left(q_\phi(\mathbf{x}_i)\right)\right) . \tag{3.30}$$

L'auto-encodeur est à son tour entraîné à minimiser l'erreur de reconstruction, qui dans ce cadre peut prendre une forme générale $c(\mathbf{x},\mathbf{y})$, ainsi qu'à tromper le discriminateur de manière à faire correspondre la distribution latente à $p(\mathbf{z})$.

$$\mathcal{L}_{AE} = \frac{1}{N}\sum_{i=1}^{N} c\left(\mathbf{x}_i, G_\theta\left(q_\phi(\mathbf{x}_i)\right)\right) - \lambda \log D_\gamma\left(q_\phi(\mathbf{x}_i)\right) . \tag{3.31}$$

Le facteur λ permet de contrôler l'amplitude de régularisation du jeu accusatoire entre d_γ et (q_ϕ, g_θ).

La fonction objectif du WAE s'obtient à partir d'une relaxation du transport optimal entre la distribution des données $p_\mathcal{X}$ et la distribution des reconstructions p_G tel que mesuré par la fonction de coût $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$. Celui-ci s'écrit

$$W_c(p_\mathcal{X}, p_G) = \inf_{\Gamma \in P(\mathbf{x} \sim p_\mathcal{X}, \mathbf{y} \sim p_G)} \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim \Gamma}[c(\mathbf{x}, \mathbf{y})] \,, \tag{3.32}$$

où $P(\mathbf{x} \sim p_\mathcal{X}, \mathbf{y} \sim p_G)$ est l'ensemble de toutes les distributions jointes de $(\mathbf{x}, \mathbf{y})$ dont les marginales sont $p_\mathcal{X}$ et p_G respectivement. Lorsque p_G est défini à travers un modèle à variables latentes, on a

$$p_G(\mathbf{x}) = \int_\mathcal{Z} \mathrm{d}\mathbf{z}\, p_G(\mathbf{x}|\mathbf{z}) p_z(\mathbf{z}) \,. \tag{3.33}$$

Dans ce cas, le transport optimal 3.32 devient

$$\inf_{Q_\phi : Q_z = p_z} \mathbb{E}_{p_\mathcal{X}} \mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}[c(\mathbf{x}, G_\theta(\mathbf{z}))] \,, \tag{3.34}$$

où Q_z est la distribution marginale de $\mathbf{z}$ lorsque $\mathbf{x} \sim p_\mathcal{X}$ et $\mathbf{z} \sim Q_\phi(\mathbf{z}|\mathbf{x})$. Ce résultat permet d'optimiser à travers les encodeurs stochastiques $Q_\phi(\mathbf{z}|\mathbf{x})$ au lieu d'optimiser à travers tous les couplages $\mathbf{x}$ et $\mathbf{y}$. La relaxation de la contrainte $Q_\phi : Q_z = p_z$ mène à la fonction objectif du WAE

$$\mathcal{L}_{WAE} = \inf_{Q_\phi(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}} \mathbb{E}_{p_\mathcal{X}} \mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}[c(\mathbf{x}, g_\theta(\mathbf{z}))] + \lambda \mathcal{D}_\mathcal{Z}(Q_z, p_z) \,, \tag{3.35}$$

où $\mathcal{Q}$ est un ensemble d'encodeurs stochastiques et $\mathcal{D}_\mathcal{Z}$ une divergence arbitraire entre Q_z et p_z.

L'implémentation qui sera considérée dans ce travail consiste à choisir $\mathcal{D}_\mathcal{Z} = D_{JS}$ et d'appliquer un entraînement accusatoire pour l'estimer, à l'aide du discriminateur d_γ introduit précédemment. L'algorithme 4 de l'annexe II présente cette implémentation.

CHAPITRE 4

APPRENTISSAGE DE LA SURFACE D'ÉNERGIE POTENTIELLE

4.1 La surface d'énergie potentielle

La qualité des modèles de structure atomique réalisés par Monte Carlo ou Dynamique Moléculaire dépend grandement de la description du paysage énergétique utilisée, c'est-à-dire de la surface d'énergie potentielle (SEP). Elle est une quantité cruciale pour la description des systèmes atomiques en ce qu'elle permet le calcul de toutes leurs propriétés en plus de servir de base pour toute procédure d'optimisation comme la recherche de structures d'énergie minimale. Bien que les méthodes *ab initio* décrivant la structure électronique comme la théorie de la fonctionnelle de la densité (DFT) fournissent une description réaliste de la SEP, elles sont computationnellement très prohibitives : les applications les plus poussées ne peuvent dépasser les quelques centaines d'atomes ou les dizaines de picosecondes de dynamique moléculaire.

Les potentiels empiriques constituent une alternative efficace grâce à la description effective des interactions au travers d'une forme fonctionnelle paramétrée physiquement plausible, dont les paramètres sont ajustés pour reproduire des données expérimentales ou *ab initio* [51, 54, 55]. Ils ouvrent ainsi la voie à des applications de grande échelle pouvant aller jusqu'au million d'atomes ; et bien que leur portabilité soit raisonnable, des circonstances où la forme fonctionnelle considérée est inappropriée peuvent mener à des résultats qualitativement mauvais [8].

Cependant l'ajustement des paramètres et la conception de la forme fonctionnelle de ces potentiels, nécessitant parfois des corrections *ad hoc* pour un système particulier [61], limitent leur précision et leur portabilité. Les algorithmes d'apprentissage permettent au contraire une approche non biaisée et purement statistique pour l'approximation des interactions atomiques. Au travers de l'exploitation de grands ensembles de

données, ils rendent possible l'évaluation de la SEP à la précision voulue, déterminée par le niveau de théorie utilisé pour générer les énergies de l'ensemble de données. Leur grande flexibilité pour l'approximation de fonctions en haute dimension ainsi que leur bonne performance (souvent linéaire avec le nombre d'atomes) en font une classe d'algorithmes de choix pour l'approximation de la SEP. Ainsi les algorithmes d'apprentissage généralisent le cadre sur lequel les potentiels empiriques sont construits, ouvrant alors une nouvelle voie pour la construction de potentiels efficaces et précis.

Le calcul de la configuration d'énergie minimale pour un système de N atomes, dont le Hamiltonien H est complètement représenté par l'ensemble des positions et des charges atomiques $\{Z_i, \mathbf{r}_i\}$, est une procédure d'optimisation[1] $H[\Psi] \rightarrow E_0[\Psi]$ difficile et computationnellement lourde. Ce problème peut être contourné par l'approximation de la SEP par un réseau de neurones sur un nombre fini d'observations. La SEP apprise permet alors d'effectuer cette même procédure d'optimisation à une vitesse jusqu'à 5 ordres de grandeur plus grande, ouvrant alors la voie à l'évaluation efficace du paysage énergétique au niveau de théorie voulue.

La surface d'énergie potentielle est une fonction scalaire réelle des degrés de liberté du système, son apprentissage est donc une tâche de régression. La représentation naturelle des degrés de liberté d'un système atomique est l'ensemble des coordonnées cartésiennes $\{\mathbf{r}_i\}_{i\in[1,\ldots,N]} \in \mathbb{R}^{N\times 3}$ ainsi que l'ensemble des charges électriques $\{Z_i\}_{i\in[1,\ldots,N]}$ si différents éléments sont présents (les vitesses sont omises puisqu'on considère des systèmes à l'équilibre). Cependant, il ne s'agit pas d'une représentation pratique pour l'apprentissage automatique.

En effet si chaque degré de liberté doit correspondre à un nœud d'entrée d'un réseau de neurones, celui-ci deviendrait inutilement grand pour des systèmes de grande échelle. Le nombre de nœuds d'un réseau de neurones ne pouvant être changé après l'entraînement, il faudrait développer un nouveau modèle pour chaque nombre de de-

[1] Sauf dans les rares cas où une diagonalisation exacte est possible.

grés de liberté. Ceci est clairement inefficace, et il convient donc d'adapter l'architecture du réseau de neurones ainsi que la représentation du système pour avoir une méthode satisfaisante.

De plus, l'énergie d'un système atomique est invariante par toute transformation orthogonale (translation, rotation, isométrie) ainsi que par toute permutation d'atomes équivalents. Mais les coordonnées cartésiennes ne présentent aucune de ces symétries : il y a alors une infinité de représentations numériquement distinctes correspondant à des configurations énergétiquement équivalentes. Ceci entrave tout processus d'apprentissage puisque la fonction apprise ne possédera pas cette invariance pourtant requise, à moins d'augmenter l'ensemble de données avec une série finie de représentations équivalentes. Cette solution n'est cependant pas efficace puisque bien souvent le nombre d'opérations de symétrie à prendre en compte est très grand. De plus il y aura toujours une erreur nette d'échantillonnage des différentes représentations puisqu'il est impossible de rendre compte de symétries continues avec un nombre fini d'opérations.

Ainsi la question cruciale pour établir un algorithme de régression de la SEP est celle de choisir une représentation du système englobant naturellement toutes ses symétries de manière à réduire la dimension du problème. Il est désirable que cette représentation satisfasse les propriétés suivantes [6, 24]

1. **Invariance par isométrie** : L'énergie est invariante par toute transformation orthogonale du système.

2. **Invariance par permutation** : L'énergie est invariante par toute permutation de l'indexage des atomes.

3. **Différentiabilité** : L'énergie est différentiable par rapport aux distances interatomiques, notamment pour permettre le calcul des forces.

Une vaste gamme de représentations de systèmes atomiques ou moléculaires a été développée dans les récentes années, comme par exemple les matrices de Coulomb, les

transformations en ondelettes, les *bag of bonds*, le chevauchement lisse de positions atomiques ou encore des fonctions de symétrie généralisées. Selon les cas, ces représentations sont traitées soit par des modèles d'apprentissage à noyau ou par des réseaux de neurones profonds. Les sections suivantes discutent des représentations pertinentes à l'apprentissage de la SEP par des réseaux de neurones profonds.

4.2 Représentation par des fonctions de symétrie généralisées

Les formes fonctionnelles des potentiels empiriques ont déjà démontré leur capacité à représenter de manière satisfaisante les environnements locaux des atomes lors de leur utilisation pour représenter l'énergie potentielle. C'est en partant de cette idée que Behler et Parrinello [8] ont utilisé des fonctionnelles similaires à celles des potentiels empiriques de Tersoff et Stillinger-Weber pour former une représentation adéquate des configurations atomiques, du fait qu'elles possèdent déjà toutes les propriétés requises pour une bonne représentation, telles qu'énoncées auparavant.

Les fonctions de symétrie sont des fonctions à plusieurs corps des coordonnées atomiques qui permettent de représenter l'environnement structurel local des atomes de manière à prendre en compte les symétries du système. Le point de départ pour leur construction consiste à exprimer, comme dans la plupart des potentiels empiriques, l'énergie totale comme la somme des énergies atomiques individuelles

$$E = \sum_i E_i \,. \tag{4.1}$$

Chaque énergie atomique est calculée à partir d'une somme sur tous les atomes, qui est restreinte à un voisinage de l'atome considéré à l'aide d'une fonction de coupure donnée par

$$f_c(r_{ij}) = \begin{cases} \tanh^3(1-\frac{r_{ij}}{r_c}) & r_{ij} \leq r_c \\ 0 & r_{ij} > r_c \end{cases} .$$

Cette fonction de coupure possède des premières et secondes dérivées continues, et se démarque ainsi des fonctions utilisées dans les potentiels de Tersoff ou Stillinger-Weber qui ont souvent une discontinuité dans leur deuxième dérivée en $r_{ij} = r_c$.

L'introduction d'une fonction de coupure sommée sur tous les atomes permet alors de garder un nombre constant de fonctions de symétrie quel que soit l'environnement local d'un atome, ce qui est une condition nécessaire pour que le réseau de neurones ait un nombre constant d'unités. Les atomes situés à $r > r_c$ ne contribuent donc pas à l'énergie de l'atome en question. Les fonctions de symétrie définies par Behler se composent alors de trois fonctions radiales et deux fonctions angulaires, qui correspondent respectivement aux contributions à deux et trois corps au potentiel

$$\mathcal{G}_i^1 = \sum_{j \neq i} f_c(r_{ij}) \tag{4.2a}$$

$$\mathcal{G}_i^2 = \sum_{j \neq i} e^{-\eta (r_{ij} - r_s)^2} f_c(r_{ij}) \tag{4.2b}$$

$$\mathcal{G}_i^3 = \sum_{j \neq i} \cos(\kappa r_{ij}) f_c(r_{ij}) \tag{4.2c}$$

$$\mathcal{G}_i^4 = 2^{1-\zeta} \sum_{j,k \neq i} (1 + \lambda \cos\theta_{ijk})^\zeta e^{-\eta (r_{ij}^2 + r_{ik}^2 + r_{jk}^2)} f_c(r_{ij}) f_c(r_{ik}) f_c(r_{jk}) \tag{4.2d}$$

$$\mathcal{G}_i^5 = 2^{1-\zeta} \sum_{j,k \neq i} (1 + \lambda \cos\theta_{ijk})^\zeta e^{-\eta (r_{ij}^2 + r_{ik}^2)} f_c(r_{ij}) f_c(r_{ik}) \,. \tag{4.2e}$$

La fonction $\mathcal{G}_1$ est simplement la somme des fonctions de coupure sur tous atomes autres que l'atome de référence. La fonction $\mathcal{G}_2$ représente une couche sphérique Gaussienne autour de l'atome considéré, dont la largeur et le centre sont contrôlés par les paramètres η et r_s respectivement. Lorsque $r_s = 0$ et η est très petit, elle se réduit à la fonction $\mathcal{G}_1$. La fonction $\mathcal{G}_3$ représente un cosinus amorti dont la période est contrôlée par le paramètre κ, semblable à une série de Fourier. Cependant, ce caractère oscillatoire peut mener différentes contributions atomiques à s'annuler, $\mathcal{G}_3$ ne dois donc jamais être utilisée seule pour représenter un environnement local. Les fonctions radiales étant toutes définies comme une somme sur les atomes voisins, elles décrivent donc la coordination effective a différentes distances de l'atome central. La figure 4.1 présente les trois fonc-

tions radiales pour différents paramètres.

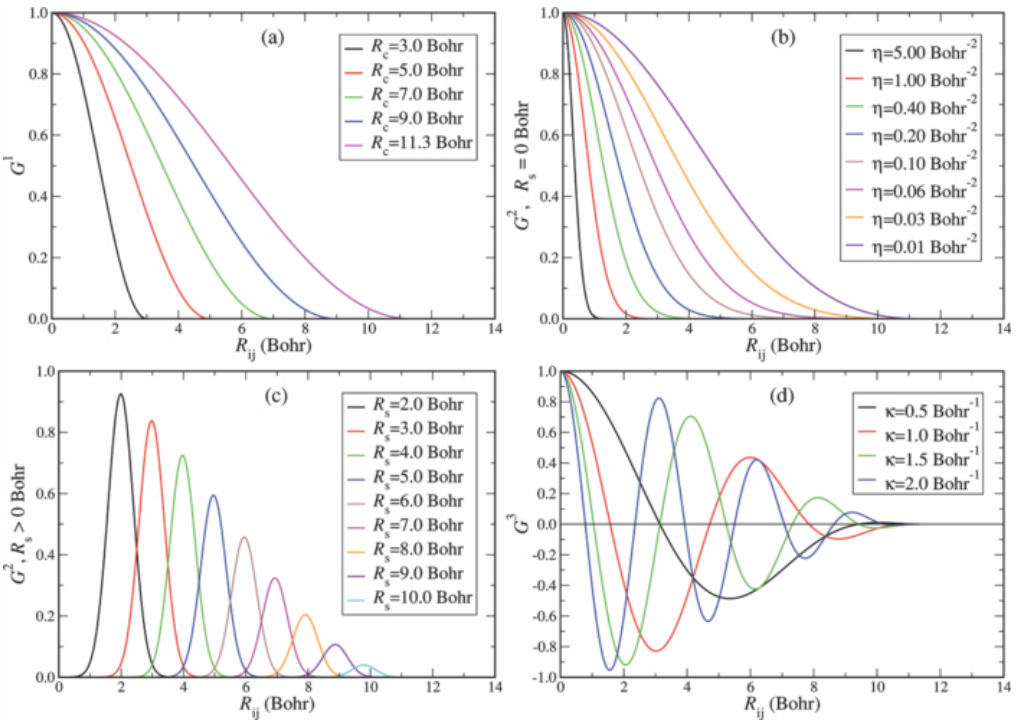

FIGURE 4.1 : Fonctions de symétrie radiale $\mathcal{G}^1$,$\mathcal{G}^2$ et $\mathcal{G}^3$ en fonction de la distance interatomique entre deux atomes quelconques [6]. Pour les fonctions $\mathcal{G}^2$ et $\mathcal{G}^3$ un rayon de coupure $r_c = 11.3$ Bohr a été utilisé et le paramètre η en c) a été fixé à 3.0Bohr^{-2}.

Les fonctions angulaires $\mathcal{G}_4$ et $\mathcal{G}_5$ partagent la même contribution angulaire, une somme pondérée par des Gaussiennes des cosinus des angles θ_{ijk} (l'angle entre $\mathbf{r}_{ij}$ et $\mathbf{r}_{ik}$), représentée à la figure 4.2, mais diffèrent dans leur contribution radiale. La résolution angulaire est contrôlée par le paramètre ζ, qui permet une gamme plus ou moins large d'angles considérer dans la description de l'environnement local. La partie radiale est contrôlée de manière similaire aux fonctions radiales par les paramètres η, r_c. Cependant, la fonction $\mathcal{G}_5$ n'imposant aucune contrainte sur $\mathbf{r}_{jk}$ elle contient naturellement un plus grand nombre de termes dans la somme. Ceci implique que $\mathcal{G}_5$ a des valeurs non nulles sur une plus grande plage d'angles.

L'invariance de ces fonctions par toute isométrie est claire : une isométrie ne mo-

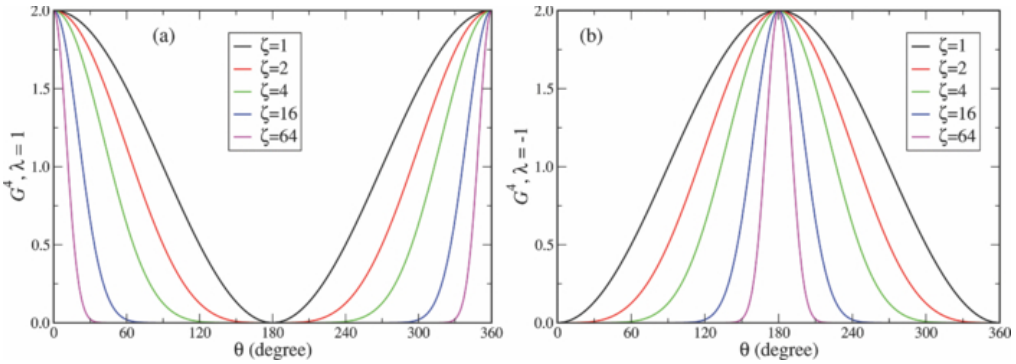

FIGURE 4.2 : Contribution angulaire aux fonctions de symétrie $\mathcal{G}_4$ et $\mathcal{G}_5$ pour un système triatomique. La symétrie par rapport à $\theta = 180°$ est bien présente, et le maximum de la fonction cosinus est contrôlé par le paramètre $\lambda = \pm 1$.

difie pas, par définition, les distances et les angles entre les vecteurs. L'invariance par permutation de chaque fonction individuelle[2] est obtenue au travers des sommes sur l'ensemble des atomes et la différentiabilité du fait qu'il s'agit d'une somme de compositions de fonctions différentiables. Remarquons que la présence de fonctions à deux corps (les fonctions radiales $\mathcal{G}_1$, $\mathcal{G}_2$ et $\mathcal{G}_3$) et de fonctions à trois corps (les fonctions angulaires $\mathcal{G}_4$ et $\mathcal{G}_5$) permet de décrire l'environnement structurel local de chaque atome sur plusieurs échelles.

Ainsi, la représentation d'une configuration atomique par des fonctions de symétrie généralisées est un changement de base englobant toutes les symétries du système qui est donc choix approprié pour la régression de l'énergie potentielle totale. Notons cependant que les fonctions angulaires $\mathcal{G}_4$ et $\mathcal{G}_5$ sont computationnellement lourdes à cause de la double somme appliquée sur le tenseur angulaire Θ.

4.3 Applications spécifiques du potentiel généralisé

L'apprentissage de la SEP à l'aide des fonctions de symétrie a été appliqué avec succès à différents problèmes de la physique du solide. Dans un premier article définis-

[2]Il est nécessaire d'ordonner la représentation totale $\mathcal{G}$ pour qu'elle soit globalement invariante par permutation, même si chaque fonction $\mathcal{G}_i$ l'est déjà.

sant cette représentation, Behler et Parrinello [8] ont utilisé 48 fonctions de symétrie de type $\mathcal{G}_2$ et $\mathcal{G}_4$ avec différents paramètres pour représenter un ensemble configurations de 64 atomes silicium liquide à 3000 K. Ainsi, l'ensemble de coordonnées cartésiennes $\mathbf{X} \in \mathbb{R}^{64\times3}$ est représenté par un ensemble de fonctions de symétrie $\mathcal{G} \in \mathbb{R}^{64\times48}$. L'énergie potentielle constituant les données de sortie a été calculée à partir de la théorie de la fonctionnelle de la densité sous l'approximation de la densité locale.

L'architecture du réseau de neurones, représentée à la figure 4.3, consiste en un simple perceptron multicouche avec 40 unités cachées et deux couches. La couche cachée est non-linéaire avec une fonction d'activation tanh tandis que la couche de sortie (avec 1 unité cachée) est linéaire. La dimension correspondant aux différents atomes est utilisée comme dimensions des mini-lots, permettant ainsi le traitement parallèle des atomes par le réseau. La sortie de la couche linéaire est alors interprétée comme un vecteur d'énergies atomiques, dont la somme constitue l'énergie potentielle totale.
Comme la description en fonctions de symétrie des configurations n'entre pas en jeu dans l'entraînement du réseau de neurones, celles-ci sont précalculées avant d'être fournies au modèle.

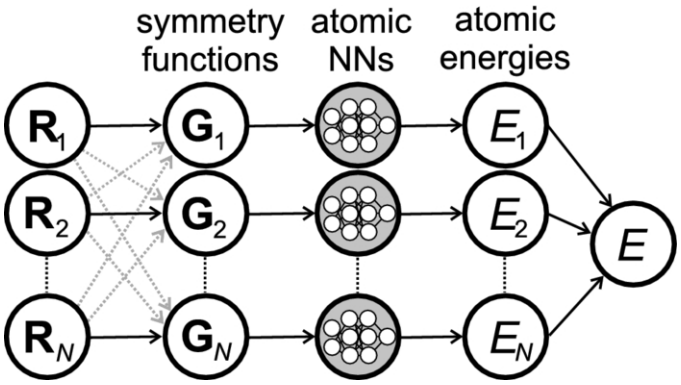

FIGURE 4.3 : Architecture du réseau de neurones pour la régression du potentiel interatomique [6]. La dimension des mini-lots correspond aux différents atomes de la configuration. Les coordonnées cartésiennes sont premièrement transformées en fonctions de symétrie puis ces dernières sont traitées par un perceptron multicouche. Les énergies sont finalement sommées pour obtenir l'énergie totale.

Puisque les configurations de l'ensemble d'entraînement ainsi que leurs énergies

peuvent être calculées à souhait, le modèle a été entraîné de manière auto-cohérente. L'ensemble de données initial comporte des structures cristallines incluant des phases à haute pression, et des simulations de dynamique moléculaire à différentes conditions de température et de pression. Lorsque l'erreur quadratique moyenne sur des nouvelles configurations est plus grande que l'erreur d'entraînement, lesdites configurations sont ajoutées à l'ensemble d'entraînement et la procédure est réitérée. Au terme de ces itérations, l'ensemble de données contient 9000 exemples de configurations et d'énergies.

Les résultats montrent que la racine de l'erreur quadratique moyenne (RMSE) sur l'ensemble d'entraînement est de 4 meV par atome et de 5 meV par atome sur l'ensemble de test, ce qui correspond à $0.15-0.18\%$ de l'énergie de lien du silicium, tandis que les forces calculées par le réseau de neurones présentent une erreur de 0.2 eV/Å par rapport aux calculs de DFT. Par la suite, différentes simulations de dynamique moléculaire avec un pas de temps de 20 ps à 3000 K ont été effectuées en utilisant des potentiels empiriques, des calculs de DFT et ledit réseau de neurones entraîné. La figure 4.4 montre que les FDR des configurations issues des simulations utilisant la DFT et le réseau de neurones sont très similaires tandis que celles issues de simulations utilisant des potentiels empiriques sont sensiblement différentes.

Une autre mesure de la fiabilité de ce réseau de neurones a été faite en comparant les énergies issues de la DFT avec les prédictions du modèle sur un système suivant une simulation de méta-dynamique à 300 K et 15 GPa avec un pas de temps de 2 ps. Malgré les grands changements structuraux induits par la simulation, le réseau de neurones entraîné effectue des prédictions de l'énergie potentielle avec une fiabilité comparable à celle atteinte sur l'ensemble de test, c'est-à-dire avec une RMSE l'ordre de 5-10 meV par atome par rapport aux énergies DFT.

Dans un article subséquent, Behler [6] introduit les fonctions de symétrie $\mathcal{G}_1$,$\mathcal{G}_3$ et $\mathcal{G}_5$ pour une description plus complète de l'environnement local d'un atome. Il montre en qualité d'exemple l'apprentissage par un simple perceptron multicouche d'une fonction

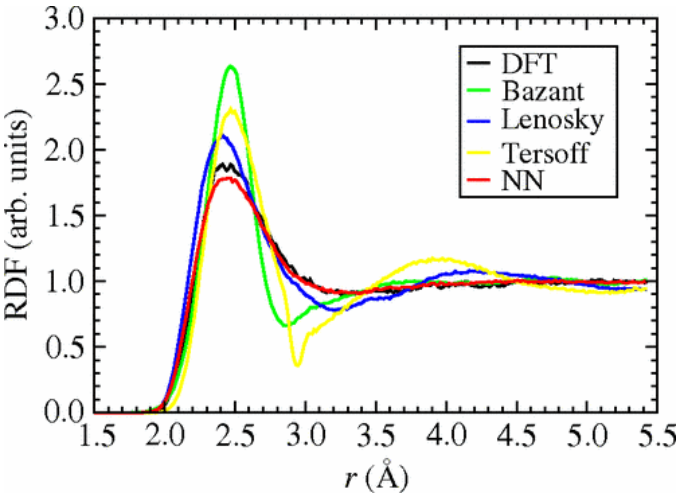

FIGURE 4.4 : Fonctions de distribution radiale d'une cellule de 64 atomes de silicium liquide à 3000 K, obtenues à partir de simulations de dynamique moléculaire à 3000 K pendant 20 ps (8 ps pour la DFT) [8]. On remarque une bonne correspondance entre la FDR issue du potentiel DFT (noir) et celle obtenue à partir du potentiel appris par le réseau de neurones (rouge). Les FDR issues d'autres potentiels empiriques s'en écartent notablement.

périodique présentant beaucoup de minima locaux

$$f(r) = \frac{\cos(5\frac{r}{r_0}) + (\frac{r}{r_0} - 4.5)^2}{5} - 1 \,, \tag{4.3}$$

lorsque la coordonnée r est représentée par 3 fonctions de symétrie de type $\mathcal{G}_2$, avec pour paramètres $\eta_1 = 1.0\text{Bohr}^{-2}$, $\eta_1 = 0.10\text{Bohr}^{-2}$, $\eta_1 = 0.01\text{Bohr}^{-2}$ et $r_c = 12.0\text{Bohr}$, $r_s = 0.0\text{Bohr}$ pour les trois fonctions. Cet exemple permet de vérifier la capacité de la transformation de coordonnées $r \rightarrow \{\mathcal{G}_2^{\eta_1,r_c,r_s}, \mathcal{G}_2^{\eta_2,r_c,r_s}, \mathcal{G}_2^{\eta_3,r_c,r_s}\}$ à représenter correctement, sans perte d'information, une fonction non convexe.

La figure 4.5 montre une excellente description de la fonction et de son gradient. Bien sûr, les fonctions de symétrie utilisées sont plus complexes que la fonction même qu'il faut reproduire ; cependant cette expérience montre que les nouvelles variables $\{\mathcal{G}_i^{\mu}\}$ ont au moins la même capacité de représentation que les coordonnées cartésiennes.

Une autre application de cette représentation porte sur les transitions structurelles en

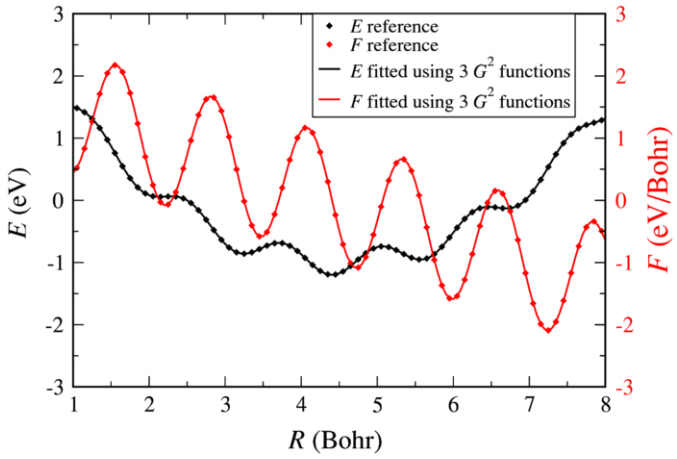

FIGURE 4.5 : Comparaison de la fonction définie en 4.3 et de la force associée $F = -\nabla E$ avec l'ajustement associé appris par le réseau de neurones [6].

fonction de la pression du silicium, qui présentent une phénoménologie riche du fait de la vaste gamme de phases observées expérimentalement.

Dans les conditions de pression et de température ambiantes le silicium se conforme dans la structure diamant. À une pression proche de 10.3 GPa il subit alors une transition vers la structure β-Sn qui se transforme ensuite en une structure Imma à 13.2 GPa puis en hexagonale simple aux alentours de 15.6 GPa. À une pression de 38 GPa le silicium adopte la structure Cmca puis à 42 GPa une structure hexagonale compacte est observée. Finalement, vers 78 GPa le silicium adopte une structure cubique à face centrées.

Bien que la DFT fournit une classification énergétique de ces phases en accord avec les observations, les potentiels empiriques sont typiquement moins fiables. Ainsi, Behler et al. [25] ont utilisé un réseau de neurones avec la même architecture qu'à la figure 4.3 avec chaque couche comportant 35 nœuds. La même procédure auto-cohérente d'entraînement et d'augmentation de l'ensemble de données décrite précédemment a été utilisée, en combinant des simulations de dynamique moléculaire à différentes températures et pressions partant de différentes structures cristallines, des structures amorphes et liquides, des structures issues de simulations Monte Carlo hybride ainsi que des simulations de méta-dynamique. Au total, un ensemble de 17144 configurations de 64 atomes

ainsi que leurs énergies respectives a été formé, dont 1907 ont été séparées pour former l'ensemble de test. L'erreur quadratique moyenne sur l'ensemble d'entraînement est de 5.9 meV par atome et de 7.1 meV par atome sur l'ensemble de test tandis que l'erreur absolue moyenne est de 4.2 et 4.7 meV par atome pour l'ensemble d'entraînement et de test respectivement.

Le potentiel ainsi obtenu est applicable à des pressions allant de 0 à 100 GPa, et les prédictions d'énergie en fonction du volume pour les différentes phases du silicium sont très fidèlement reproduites par celui-ci comme le montre la figure 4.6.

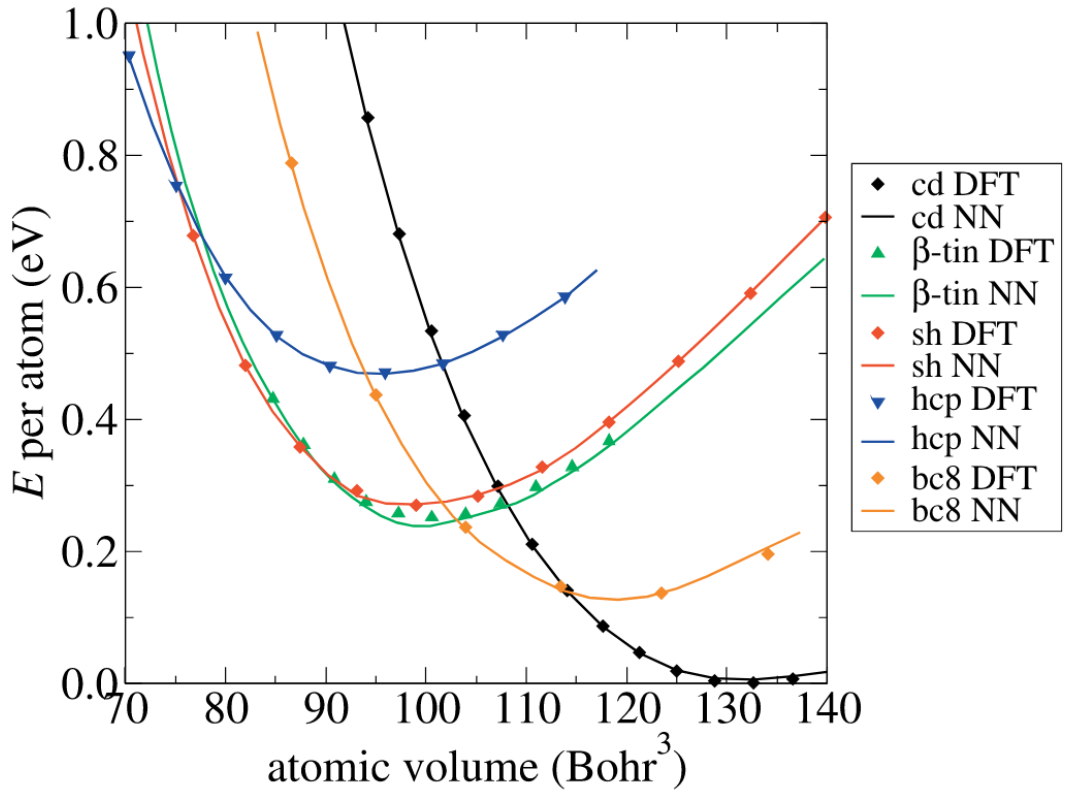

FIGURE 4.6 : Énergie en fonction du volume atomique pour différentes phases du silicium calculées à l'aide de la DFT et l'approximation de la densité locale et calculées avec le potentiel issu du réseau de neurones [25].

Toutes les transitions structurelles citées plus haut sont correctement prédites à l'aide

de simulations de méta-dynamique utilisant le potentiel appris par le réseau de neurones. En particulier, la phase intermédiaire β-Sn se présentant lors de la transition entre la structure diamant et la structure hexagonale simple est correctement prédite par le potentiel tandis que l'utilisation d'un potentiel basé sur la méthode des liaisons fortes résulte en une phase métastable.

Un fait remarquable est que ledit potentiel prédit correctement la phase Imma lors de la transition de la structure β-Sn à la structure hexagonale simple alors que cette structure n'a pas été intégrée dans l'ensemble d'entraînement. Ceci souligne le fait que lorsque le réseau de neurones est entraîné sur un grand nombre d'observations (i.e. d'environnements atomiques) suffisamment variées, il devient très transférable. En effet, plus d'un million d'environnements atomiques ont été utilisés pour l'entraînement, ainsi même si un certain environnement atomique n'a jamais été observé lors de la construction du potentiel, celui-ci peut être proche d'autres environnements connus. La flexibilité et la différentiabilité du réseau de neurones permettent alors de caractériser correctement ce nouvel environnement.

Il est important de noter cependant que des structures très différentes de celles présentes dans l'ensemble d'entraînement ne pourront pas être décrites correctement. Il est par exemple difficile de penser qu'un réseau de neurones entraîné sur des structures avec des conditions aux frontières périodiques soit capable de caractériser correctement les effets de surface sur des échantillons finis.

En conclusion, la représentation d'une configuration atomique par des fonctions de symétrie permet l'apprentissage de la surface d'énergie potentielle en ce qu'elle intègre correctement les symétries du système tout en décrivant de manière complète l'environnement local atomique. Au travers de cette représentation, un réseau de neurones permet d'approximer la SEP issue des calculs de DFT précisément et peut par la suite être utilisé comme fonction d'énergie potentielle dans des simulations de dynamique moléculaire avec une performance comparable à celle des potentiels empiriques tout en gardant le

réalisme issu des calculs *ab initio*. Cette stratégie prédit avec succès les transitions de phase structurelles du silicium, mais a aussi été appliquée au calcul des propriétés structurelles et vibrationnelles du sodium ainsi que les phases de coexistence du diamant et du graphite [28, 29].

Les fonctions de symétrie sont donc une représentation bien adaptée à la description des positions atomiques pour l'apprentissage profond et c'est pourquoi les modèles développés dans le prochain chapitre se basent sur celle-ci.

CHAPITRE 5

MODÈLES GÉNÉRATIFS PROFONDS POUR LA STRUCTURE ATOMIQUE DU SILICIUM AMORPHE

Les réseaux de neurones profonds permettent d'approximer de manière efficace la SEP, ce qui permet leur utilisation comme fonction d'énergie potentielle dans des simulations de dynamique moléculaire. Ces dernières simulent l'évolution temporelle du système physique considéré en intégrant les équations du mouvement, et sont donc intrinsèquement itératives. Toutefois dans le cadre de la modélisation de la structure atomique d'un matériau, une telle procédure est computationnellement exigeante du fait des grandes barrières énergétiques mises en jeu dans une déformation structurelle du système. La méthode d'activation et de relaxation (ART) permet de contourner ce problème en explorant directement le paysage énergétique du système pour visiter séquentiellement les minima locaux, correspondant à des structures réalistes, par le biais des points de selle.

Dans cette optique, les modèles génératifs profonds contournent ce problème en approximant la distribution de probabilité des configurations atomiques de l'ensemble d'entraînement ce qui permet par la suite d'en tirer des réalisations de manière très efficace. En effet, lors de la génération d'un échantillon par exemple par un GAN ou un VAE, aucun calcul d'énergie ni procédure itérative ne sont effectués : l'exemple est généré par une simple propagation avant d'un vecteur aléatoire de l'espace latent à travers le générateur.

Le développement d'un modèle génératif pour la structure atomique du silicium amorphe nécessite avant toute chose la formation d'un ensemble de données sur lequel entraîner le réseau de neurones. Cette procédure, dont il sera question dans la première section, est d'une importance capitale pour la performance de l'apprentissage. Dans le but de valider la performance et la viabilité des modèles considérés, de confirmer la

possibilité d'effectuer un apprentissage sur l'ensemble de données généré et de former une intuition pour la tâche d'apprentissage en question, la deuxième section traite de l'apprentissage de la SEP à partir de l'ensemble de données formé, en analogie avec les travaux de Behler et al. exposés au chapitre précédent. Finalement, la dernière section présente les résultats de l'entraînement d'un WGAN et d'un WAE conditionnels pour la modélisation de la structure atomique du silicium amorphe et discute l'analyse structurelle des configurations générées.

5.1 Construction de l'ensemble de données

L'apprentissage statistique nécessite absolument un ensemble de données représentant un bon échantillonnage de la distribution de probabilité que l'on souhaite approximer. Définir ce qu'est un bon échantillonnage pour l'apprentissage est donc difficile en pratique et peut dépendre de la distribution considérée. Bien souvent, il s'agit de disposer d'un ensemble d'observations suffisamment grand et varié de manière à capturer les caractéristiques principales de la distribution, comme la moyenne et la variance, ou dans notre cas les différentes propriétés structurelles du a-Si.

Pour les configurations atomiques du silicium amorphe, il est nécessaire de déterminer quels sont les facteurs qui déterminent ce qu'est une configuration acceptable, c'est-à-dire quelles sont les caractéristiques de la distribution de probabilité $p(\mathbf{x})$ associant à un ensemble de coordonnées $\mathbf{x} \in \mathbb{R}^{n\times 3}$ la probabilité qu'il s'agisse d'une configuration atomique de a-Si. D'un point de vue énergétique, ces configurations représentent des minima locaux de la SEP, le minimum global correspondant à la structure cristalline. Ainsi, un bon échantillonnage de la distribution $p(\mathbf{x})$ que l'on cherche à approximer se traduit par un bon échantillonnage des minima de la SEP.

Les algorithmes développés ici pouvant être appliqués à n'importe quel choix de SEP, le choix le plus pratique pour le développement d'une preuve de concept est celui

du potentiel empirique de Stillinger-Weber modifié, dont les paramètres sont exposés dans le tableau 5.I. Ce potentiel permet de générer efficacement un grand ensemble de données à l'aide des diverses implémentations disponibles dans la vaste littérature de modèles qui en font l'usage.

ε	A	λ	σ	a	B	p	γ
1.64833	7.049556277	31.5	2.095037426	1.8	0.6022245584	4	1.2

TABLEAU 5.I : Paramètres du potentiel de SW modifié [61].

Dans le but d'obtenir un ensemble de données varié, les configurations ont été générées par deux méthodes différentes.

La première partie de l'ensemble de données a été formée à l'aide de la technique d'activation et de relaxation ART décrite auparavant, pour des configurations de 216 atomes (27 cellules unitaires) dans une boîte de simulation fixe de $[-8.141, 8.141]^3 \text{Å}^3$ avec des conditions aux frontières périodiques (CFP). Chaque nouveau minimum est pris comme un nouvel exemple, et l'ensemble des minima est ensuite filtré pour éliminer les configurations structurellement équivalentes, c'est-à-dire les configurations qui sont liées par une opération de symétrie de la boîte de simulation ou par une permutation d'indices atomiques. De plus, toutes les configurations ayant une énergie positive ont été rejetées.

Les configurations générées par ART sont donc fortement corrélées, mais leur nombre est suffisant pour qu'une grande variété de structures soit présente dans le jeu de données. La présence de configurations hautement corrélées dans l'ensemble de données n'est pas a priori problématique est pourrait même être bénéfique à l'apprentissage, bien qu'une étude détaillée de l'impact de ces corrélations est manquante.

Par la suite, l'ensemble de données a été étendu à l'aide de simulations de dynamique moléculaire avec le même potentiel à une température de 900 K et avec un pas de temps de 1 fs. À chaque intervalle régulier de 10 fs, une nouvelle configuration est ajoutée à l'ensemble de données. Les configurations équivalentes ou d'énergie positive ont aussi

été rejetées.

Au total, l'ensemble est donc constitué de 238071 configurations de silicium amorphe et autant d'énergies potentielles totales de SW, constituant respectivement les entrées et les cibles de l'ensemble de données. Celui-ci a été ensuite séparé en ensemble d'entraînement, de validation et de test constitués respectivement de 230000, 4071 et 4000 exemples. Les propriétés principales de cet ensemble de données (noté $\mathbb{D}$) sont présentées au tableau 5.II, tandis que la figure 5.1 montre la FDR de quatre configurations prises aléatoirement de $\mathbb{D}$.

N	ρ (g/cm^3)	$\langle E \rangle$ (eV)	σ_E (eV)	$\langle r_0 \rangle$ Å	$\langle \theta_0 \rangle$ (°)
238071	2.334	-686.4332	14.777609	2.357	108.97

TABLEAU 5.II : Propriétés de l'ensemble de données $\mathbb{D}$: nombre d'exemples N, valeur moyenne de l'énergie potentielle totale (cibles) $\langle E \rangle$, déviation standard de l'énergie potentielle totale σ_E, valeur moyenne de la distance entre premiers voisins $\langle r_0 \rangle$, valeur moyenne de l'angle de liaison $\langle \theta_0 \rangle$.

5.2 Apprentissage de la SEP

Maintenant que nous disposons d'un ensemble de données $\mathbb{D} = \{\mathbf{x}_i, E_i\}_{i=1}^N$, la première tâche consiste à valider la méthodologie qui sera utilisée par la suite pour la modélisation de la structure du silicium amorphe. Il s'agit en particulier de confirmer l'efficacité de la représentation par des fonctions de symétrie, de valider la possibilité d'effectuer une tâche d'apprentissage sur $\mathbb{D}$ et de comparer la performance de différents choix d'architectures de réseaux de neurones pour l'apprentissage de la SEP.

Remarquons ici qu'aucune tentative d'entraîner un réseau de neurones pour l'apprentissage de la SEP en utilisant directement les coordonnées cartésiennes des configurations n'a pu réussir. La représentation par des fonction de symétrie, où toute autre représentation décrivant de manière invariante l'environnement local des atomes, est donc

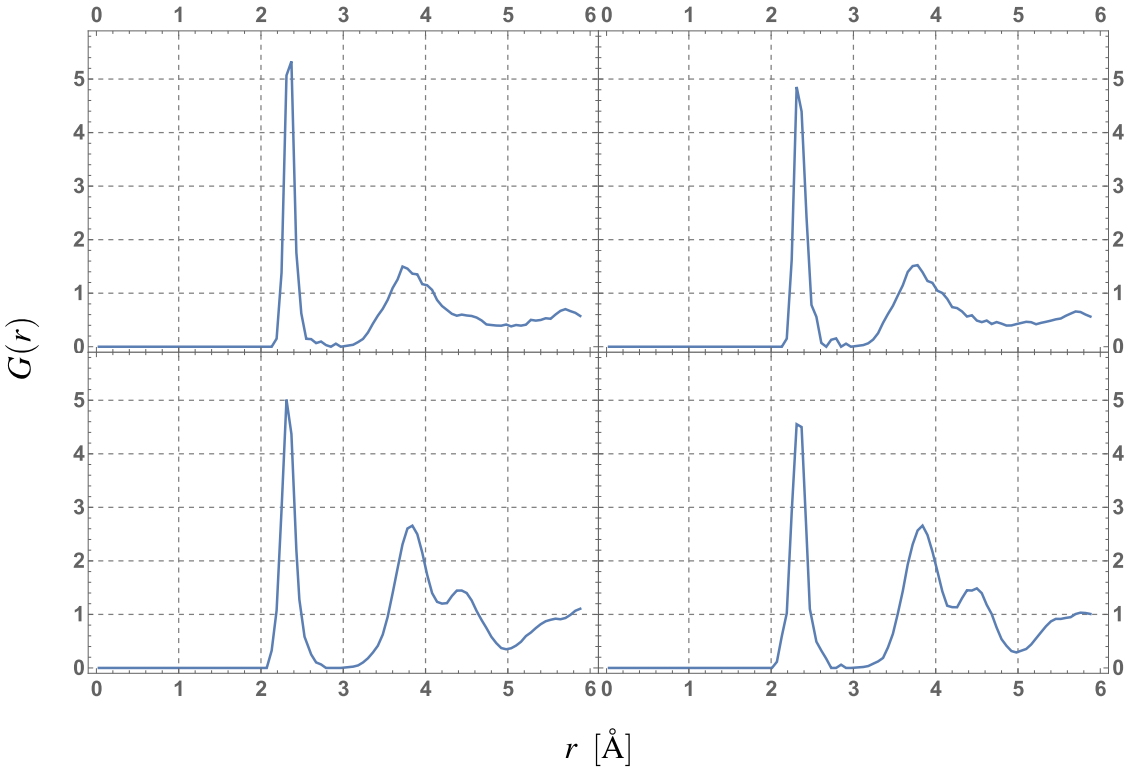

FIGURE 5.1 : Fonctions de distribution radiale de 4 éléments de l'ensemble de données.

nécessaire pour mener à bien une tâche d'apprentissage sur des systèmes atomiques [6]. Pour représenter de manière complète une configuration atomique, 18 fonctions de symétrie de chaque type $\mathcal{G}_1, \mathcal{G}_2$ et $\mathcal{G}_3$ ont été utilisées pour un total de 54 fonctions de symétrie. Les fonctions de symétrie angulaire $\mathcal{G}_4$ et $\mathcal{G}_5$ n'ont pas été utilisées car le calcul et la double somme du tenseur angulaire $\Theta_{ijk} \in \mathbb{R}^{N^3}$ inclus dans leur définition sont computationnellement inaccessibles, leur intégration aux modèles ralentissant l'entraînement d'un facteur 100.

Ainsi, une configuration atomique représentée par une matrice de coordonnées cartésiennes $\mathbf{x} \in \mathbb{R}^{216\times 3}$ est transformée en 54 fonctions de symétrie $\mathbf{g} \in \mathbb{R}^{216\times 54}$. Contrairement aux travaux de Behler et al., leur calcul ne constitue plus un prétraitement avec des paramètres fixes. Le calcul des fonctions de symétrie est ici implémenté comme une couche de neurones de manière à pouvoir intégrer les paramètres des fonctions à l'apprentissage au même titre que les autres paramètres du réseau de neurones. Une telle

implémentation permet aussi d'accélérer leur calcul en prenant avantage du calcul tensoriel optimisé sur les GPU.

La première architecture de réseau de neurones considérée pour la régression de énergies potentielles totales est un perceptron multicouche (MLP), similairement aux travaux de Behler [8]. Chaque atome est traité en parallèle par le réseau de neurones, c'est-à-dire que chaque configuration est considérée comme un mini-lot de 216 exemples : le réseau effectue donc un pas de gradient par configuration. L'énergie totale est ensuite calculée comme la somme des énergies individuelles $E = \sum_{i=1}^{N} E_i$. L'architecture du MLP utilisé est présentée en détail dans la figure 5.2.

La deuxième architecture diffère de l'approche de Behler, elle consiste en un réseau de neurones convolutif (CNN) à une dimension, où la dimension des mini-lots est aussi utilisée pour traiter en parallèle chaque atome et la dimension spatiale correspond aux différentes fonctions de symétrie. Celles-ci n'ayant pas de structure ni d'ordre significatif toutes les couches convolutives utilisent des filtres de dimension 1 (une dimension plus grande faciliterait le sur-apprentissage). Dans ce cas, la convolution de k filtres $\mathbf{w}$ avec un vecteur $\mathbf{x}$ de taille n correspond simplement au produit dyadique $(\mathbf{w} \otimes \mathbf{x})_{ij} = w_i x_j$.

De plus chaque couche convolutive est remplacée par un bloc résiduel, un choix d'architecture qui exhibe un gain de performance notable pour plusieurs tâches d'apprentis-

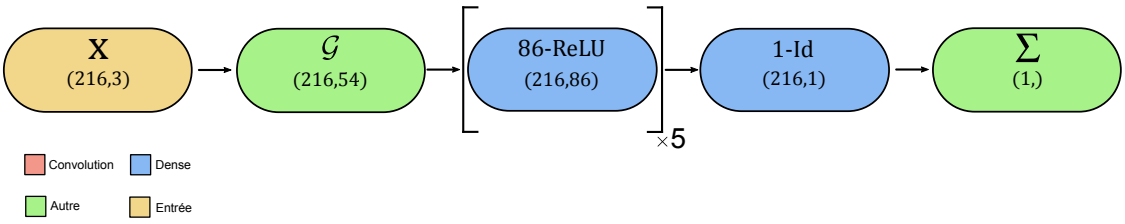

FIGURE 5.2 : Architecture du MLP pour l'apprentissage de la SEP. Chaque bloc représente une couche de neurones dont la couleur spécifie le type, le nom indique le nombre de paramètres et la fonction de transfert, et les dimensions du tenseur de sortie de la couche sont indiquées. La première dimension correspond toujours à celle des mini-lots : chaque élément dans cette dimension est traité en parallèle par le réseau. Le bloc $\mathcal{G}$ correspond au calcul des fonctions de symétrie tandis que le bloc Σ effectue une somme.

sage supervisé standard [22]. L'idée d'un bloc résiduel est de définir des courts-circuits entre différentes couches du réseau de neurones qui seraient le cas échéant séparées par d'autres couches. Ces courts-circuits permettent de limiter le problème récurrent de l'évaporation des gradients dans les réseaux profonds en facilitant la circulation des gradients au travers du réseau, en plus d'encourager la réutilisation des caractéristiques apprises par une couche plus loin dans le réseau. L'architecture du CNN utilisé ici est présentée à la figure 5.3.

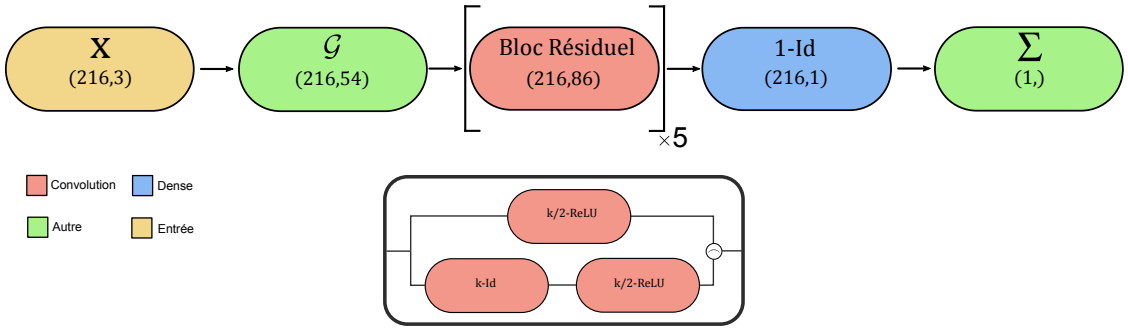

FIGURE 5.3 : Architecture du CNN pour l'apprentissage de la SEP, et structure des blocs résiduels utilisés. L'opération $\frown$ correspond à la concaténation sur la dimension des canaux du tenseur. Chaque bloc représente une couche de neurones dont la couleur spécifie le type, le nom indique le nombre de paramètres et la fonction de transfert, et les dimensions du tenseur de sortie de la couche sont indiquées. La première dimension correspond toujours à celle des mini-lots : chaque élément dans cette dimension est traité en parallèle par le réseau. Les convolutions ont toutes des filtres de taille 1, le nombre de paramètres indiqué correspond donc au nombre de filtres de la couche.

Ces modèles (MLP et CNN) sont entraînés à minimiser l'erreur quadratique entre l'énergie estimée $\tilde{E}(\mathbf{x})$ et l'énergie cible y, c'est-à-dire la fonction objectif

$$\mathcal{L}(\mathbf{x}, y) = (\tilde{E}(\mathbf{x}) - y)^2 \,. \tag{5.1}$$

Ils sont entraînés avec ADAM(η,β_1,β_2,λ) où η est le pas de gradient, β_1 le facteur d'inertie, β_2 le facteur d'inertie pour l'estimé du second moment du gradient et λ le facteur de l'échéancier des paramètres tel que décrit dans l'annexe I. Leur performance est mesurée par la racine de l'erreur quadratique moyenne sur N configurations (de valida-

tion ou de test), c'est-à-dire

$$\mathrm{RMSE}(\tilde{E}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\tilde{E}(\mathbf{x}_i) - y_i)^2}\,. \tag{5.2}$$

Les résultats de l'entraînement de ces deux architectures sur $\mathbb{D}$ sont exposés au tableau 5.III. Chaque modèle possède un nombre similaire de paramètres, et a été entraîné pendant 8 époques. La figure 5.4 présente l'évolution de l'erreur d'entraînement pendant l'entraînement du MLP et du CNN. Lorsque l'erreur de validation atteint un minimum, les paramètres du modèle sont sauvegardés.

Le CNN présente une meilleure performance que le MLP avec une RMSE de généralisation de 1.101 eV (5.095 meV/atome) contre 1.507 eV (6.978 meV / atome), pour les paramètres correspondant à la meilleure performance de validation. Ces valeurs sont très similaires à celles atteintes par Behler [8] pour des configurations de 64 atomes, malgré que la description utilisée ici soit défaillante du fait de l'omission des fonctions de symétrie G_4 et G_5 dans la description des configurations atomiques, puisque leur intégration

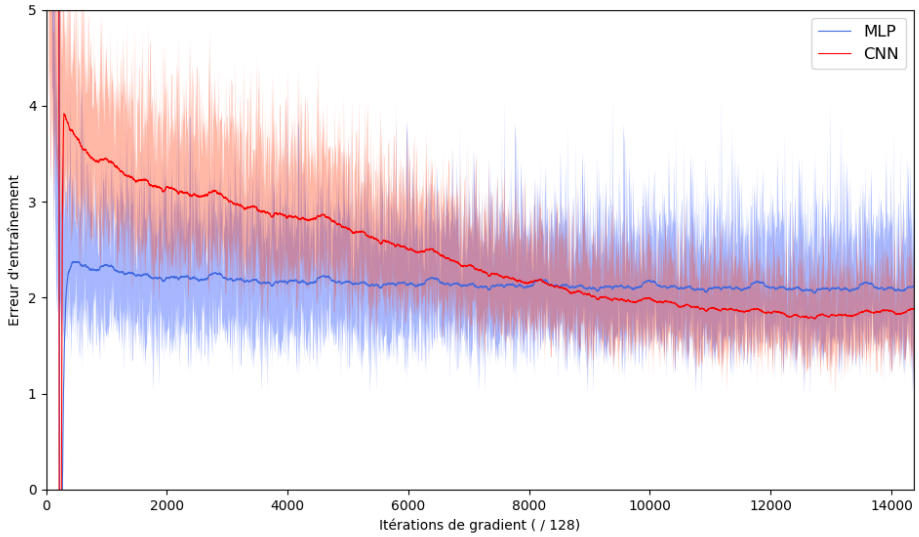

FIGURE 5.4 : Erreur quadratique moyenne (sur 128 exemples) d'entraînement du MLP et du CNN. Les courbes pleines sont le résultat d'un filtrage de hautes fréquences de type Savitzky-Golay.

s'est révélée computationnellement inaccessible.

On remarque aussi que le nombre de paramètres des réseaux de neurones utilisés est plus grand : cette plus grande capacité se traduit en une convergence plus rapide. Cependant, chaque atome étant traité en parallèle par le modèle, il ne devrait pas y avoir de dépendance significative de la capacité optimale avec le nombre d'atomes. Il est donc attendu qu'une performance similaire puisse être atteinte avec moins de paramètres mais un entraînement prolongé.

Ces résultats indiquent que l'utilisation de convolutions et de blocs résiduels présente un avantage net par rapport aux couches denses. En particulier, lors des différents essais effectués, l'introduction de blocs résiduels a toujours manifesté une convergence plus rapide, malgré la faible profondeur du réseau.

	Nombre de paramètres	Époques	RMSE$(\mathbb{D}_{\text{ent}})$	RMSE$(\mathbb{D}_{\text{test}})$	
			(eV)	(eV)	(meV/at.)
MLP	34853	8	0.99833	1.507	6.978
CNN	34761	8	0.97121	1.101	5.095

TABLEAU 5.III : Comparaison de la performance des deux architectures MLP et CNN pour la régression des énergies potentielles. Chaque modèle est formé d'une première couche appliquant la transformation en fonctions de symétrie. Le MLP est composé de 5 couches denses avec des fonctions de transfert ReLU et une couche de sortie linéaire. Le CNN est composé de 5 blocs résiduels convolutifs avec des filtres scalaires, des fonctions de transfert ReLU et une couche de sortie linéaire. L'entraînement a été fait avec ADAM dans les deux cas avec les hyper-paramètres $\eta = 10^{-4}, \beta_1 = 0.5, \beta_2 = 0.9, \lambda = 10^{-6}$.

La performance de généralisation obtenue avec chacune des architectures est satisfaisante et permettrait l'utilisation de la SEP apprise dans des simulations la requérant. Mais il est important de remarquer que l'apprentissage ne s'est fait que sur des configurations réalistes, proches d'un minimum local de la vraie surface d'énergie potentielle. Ceci implique qu'il n'est pas possible de s'attendre à un bon ajustement de la SEP pour

des configurations substantiellement différentes que celles sur lesquelles s'est entraîné le modèle.

En effet, une procédure de minimisation de la SEP apprise a été effectuée dans le but de vérifier la qualité de celle-ci. L'algorithme d'optimisation utilisé est la méthode de Newton tronquée, ou méthode des gradients conjugués avec imposition des CFP[1]. Lorsque la configuration initiale est aléatoire, la procédure d'optimisation révèle que la SEP apprise possède effectivement des minima fictifs correspondant à des structures irréalistes ayant une énergie de 10^9 eV.

Pour combler ce problème, il est possible d'ajouter à $\mathbb{D}$ des configurations de haute énergie en ajoutant du bruit (uniforme ou Gaussien) aux configurations et en modifiant l'énergie potentielle cible conformément. Mais un fait remarquable est qu'il a été impossible d'atteindre une performance acceptable avec les modèles exposés ici lorsque de telles configurations sont intégrées dans l'ensemble d'entraînement. Même lorsque seule une faible fraction de l'ensemble d'entraînement est bruitée, le modèle ne converge pas et stagne à une performance très basse. Il s'agit de la raison pour laquelle les configurations de haute énergie générées parfois par ART ont été retirées lors de la formation de $\mathbb{D}$.

Une explication possible pour ce comportement est que les configurations atomiques de haute énergie ou aléatoires se trouvent dans un sous-espace de $\mathbb{R}^{N\times 3}$ très différent que celui des configurations à l'équilibre ou proches de l'équilibre. Ainsi, si le modèle apprend rapidement la structure de cet ensemble, l'ajout de nouvelles observations en dehors dudit sous-espace perturbe l'entraînement. Il est possible que l'entraînement d'un GAN semi-supervisé puisse alléger ce problème, du fait que le générateur prendrait le rôle de fournir des configurations corrompues au discriminateur, qui serait alors entraîné à les rejeter correctement. Ceci résulterait en une SEP plus robuste vis-à-vis des configurations structurellement différentes que celles présentes dans $\mathbb{D}$.

[1] Cette minimisation a été effectuée avec l'algorithme TNC de la librairie logicielle SciPy, en Python.

5.3 Modèles génératifs profonds

Il est maintenant clair qu'il est possible d'appliquer une tâche d'apprentissage pour des configurations atomiques de silicium amorphe à l'aide de la représentation par des fonctions de symétrie. Naturellement, l'application des modèles génératifs profonds présentés en 3.5 et 3.6 pour la modélisation de la structure du silicium amorphe nécessite donc aussi l'utilisation de cette représentation.

Ces modèles ont étés développés spécifiquement pour l'apprentissage non supervisé. Cependant, il est certainement avantageux d'utiliser efficacement toute l'information contenue dans $\mathbb{D}$ pour l'entraînement. Comme nous disposons en plus de l'énergie potentielle totale de chaque configuration, il est nécessaire d'adapter les architectures du WAE et du WGAN pour en intégrer l'usage. Une stratégie commune est de conditionner le modèle par les énergies cibles $\mathbf{y}$.

Pour le WGAN, ceci correspond à fournir $\mathbf{y}$ à la fois au générateur g_θ et au critique f_ω ce qui modifie leurs distributions en $G_\theta(\mathbf{x}|\mathbf{z},\mathbf{y})$ et $F_\omega(p|\mathbf{x},\mathbf{y})$ respectivement i.e. les distributions sont conditionnelles en $\mathbf{y}$.

Pour le WAE, ceci correspond à fournir $\mathbf{y}$ à la fois à l'encodeur q_ϕ et au décodeur g_θ ce qui modifie leurs distributions en $Q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$ et $G_\theta(\mathbf{x}|\mathbf{z},\mathbf{y})$ respectivement. Bien que dans d'autres travaux seul le décodeur est conditionnel [34], le modèle développé ici a présenté une performance légèrement accrue lorsque tout l'auto-encodeur est conditionnel.

L'entraînement d'un WGAN sur $\mathbb{D}$ s'est avéré plus instable et difficile à faire converger que celui du WAE. Lors des différents essais effectués le WAE présentait un entraînement plus stable et générait systématiquement des configurations de plus basse énergie, avec des architectures et des hyper-paramètres semblables. Ainsi, seuls les résultats du WAE seront présentés en détail.

Un des problèmes possibles qui freinent la performance de génération du WGAN

est que le générateur ne possède pas, malgré l'ajout d'un terme de reconstruction dans sa fonction objectif, de représentation sensible d'un ensemble d'atomes de manière à pouvoir générer des configurations réalistes. Le WAE traite naturellement ce problème en apprenant une représentation latente des données d'entraînement à partir de laquelle le générateur, i.e. le décodeur, peut générer efficacement.

De plus, l'architecture du WAE incorpore naturellement un objectif de reconstruction. Il est donc important de choisir la fonction mesurant l'erreur de reconstruction avec précaution, puisqu'elle est responsable de fournir l'information de gradient nécessaire à la bonne convergence de l'encodeur et du décodeur.

Définir une métrique, en termes de propriétés structurelles, entre deux configurations atomiques n'est pas trivial. Celle-ci doit préférablement avoir les propriétés d'une distance, être calculable efficacement pour permettre l'implémentation sur les GPU et la rétropropagation des gradients, et posséder des bonnes propriétés de continuité de manière à résulter en une information de gradient bien définie partout. Finalement, une distance entre deux configurations atomiques doit aussi être naturellement invariante par toutes les transformations de symétrie qui laissent lesdites structures invariantes, comme la translation, la rotation, la réflexion et la permutation d'indices atomiques.

Les fonctions de symétrie $\mathcal{G}_\mu(\mathbf{x})$ possèdent naturellement ces propriétés et permettent donc la définition d'une distance entre deux configurations. En considérant le changement de variables induit par celles-ci comme un espace Euclidien, il est alors convenable de définir la distance entre deux configurations comme la distance Euclidienne entre leurs fonctions de symétrie. Celles-ci ne sont cependant pas invariantes par permutation d'indices, et il est donc nécessaire de forcer un ordre d'indices en ordonnant les vecteurs $\mathcal{G}_\mu(\mathbf{x})$. Ainsi, la distance entre deux configurations atomiques est

$$d_{\mathcal{G}}(\mathbf{u},\mathbf{v}) = \sum_{i=1}^{N}\sum_{\mu=1}^{K}\left(\mathrm{sort}\mathcal{G}_\mu^i(\mathbf{u}) - \mathrm{sort}\mathcal{G}_\mu^i(\mathbf{v})\right)^2 . \tag{5.3}$$

Cette fonction possède en effet les propriétés d'une distance puisqu'il est aisé de vérifier

qu'elle est non-négative, symétrique par rapport à ses arguments, qu'elle vaut 0 lorsque $\mathbf{u} \sim \mathbf{v}$ (où $\sim$ signifie l'équivalence de deux configurations), et qu'elle vérifie l'inégalité du triangle.

Le coût de reconstruction du WAE est donc défini comme la valeur moyenne de la distance $d_{\mathcal{G}}$ entre un mini-lot de configurations reconstruites et le mini-lot de configurations originales. Ainsi, les fonctions objectif du discriminateur et de l'auto-encodeur sont, respectivement

$$\mathcal{L}_D = \frac{\lambda}{B} \sum_{i=1}^{B} \log f_\gamma(\mathbf{z}_i) + \log\left(1 - f_\gamma\big(q_\phi(\mathbf{x}_i, \mathbf{y}_i)\big)\right) \tag{5.4}$$

$$\mathcal{L}_{AE} = \frac{1}{B} \sum_{i=1}^{B} [d_{\mathcal{G}}(\mathbf{x}_i, g_\theta(q_\phi(\mathbf{x}_i, \mathbf{y}_i), \mathbf{y}_i))] - \lambda \log f_\gamma(q_\phi(\mathbf{x}_i, \mathbf{y}_i)) \,, \tag{5.5}$$

où B est la taille des mini-lots. La figure 5.5 illustre l'architecture du WAE conditionnel utilisé.

L'Auto-Encodeur est essentiellement convolutif, puisque ce choix permet la formation de réseaux plus profonds avec un nombre réduit de paramètres et s'est accompagné d'une meilleure performance que des couches denses pour l'apprentissage de la SEP.

Les premières couches de l'encodeur consistent à calculer les fonctions de symétrie à partir des coordonnées atomiques et à traiter les énergies potentielles totales par une couche dense. Les sorties de ces deux couches sont concaténées avant d'être traitées par 5 couches convolutives. Une dernière couche dense linéaire, nécessaire pour générer des vecteurs latents arbitraires, est appliquée pour générer les vecteurs de sortie μ, Σ de l'encodeur. Ces derniers sont utilisés pour tirer le mini-lot de vecteurs latents de manière stochastique $\mathbf{z} = \mu + \varepsilon\Sigma$ où $\varepsilon \sim \mathcal{N}(0,1)$. Ce calcul permet de tirer de $\mathcal{N}(\mu, \Sigma)$ tout en gardant des gradients bien définis par rapport à μ et Σ, ce qui rend possible la rétropropagation des gradients à travers l'encodeur[2]. Il est important de noter que le choix d'un encodeur stochastique, plutôt que déterministe (i.e. qui produit directement le vecteur latent $\mathbf{z}$), est important pour la bonne convergence du modèle.

[2]En effet, les expressions $\nabla_\mu \mathcal{N}(\mu, \Sigma)$ et $\nabla_\Sigma \mathcal{N}(\mu, \Sigma)$ sont indéfinies.

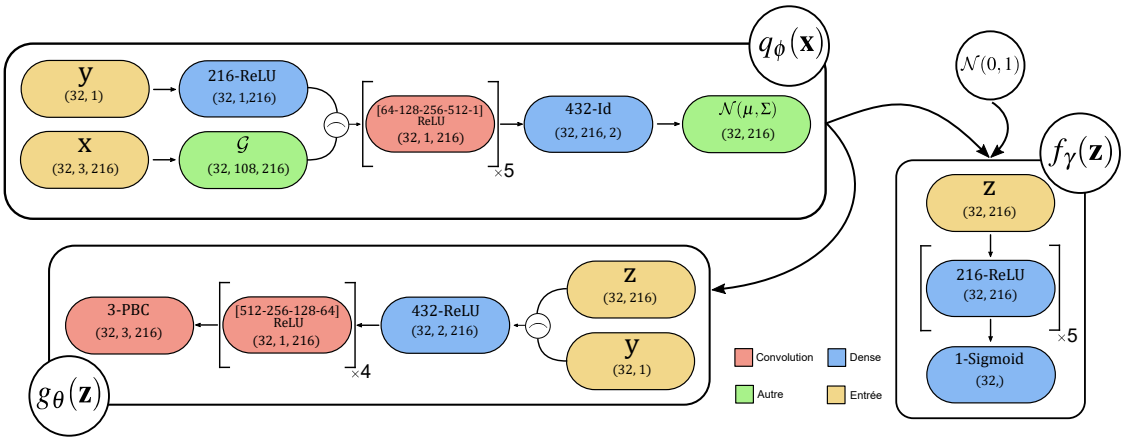

FIGURE 5.5 : Architecture du WAE utilisé pour la modélisation de la structure atomique du a-Si. Chaque bloc représente une couche de neurones, dont la couleur spécifie le type, le nom indique le nombre de paramètres et la fonction de transfert (ou la nature de la couche) et les dimensions du tenseur de sortie de la couche sont indiquées. Les couches d'entrée **x** et **y** représentent respectivement l'approvisionnement d'un mini-lot de coordonnées atomiques et d'énergies potentielles. La couche d'entrée **z** représente l'approvisionnement d'un vecteur latent aléatoire tiré d'une distribution normale $\mathcal{N}(0,1)$ ou $\mathcal{N}(\mu,\Sigma)$, où μ,Σ sont les vecteurs de sortie de l'encodeur. La couche $\mathcal{G}$ représente le calcul des 108 fonctions de symétrie. La première dimension des tenseurs est celle des mini-lots, la deuxième celle des canaux et la troisième est la dimension spatiale sur laquelle les convolutions sont appliquées. L'opération $\frown$ symbolise la concaténation sur la deuxième dimension des tenseurs entrants.

La première couche du décodeur est dense et permet de projeter la concaténation des vecteurs latents et des énergies potentielles totales vers une dimension plus grande, avant d'être traités par 5 couches convolutives. La fonction de transfert de la dernière couche du décodeur, permettant d'assurer les CFP, est

$$\mathrm{pbc}(x) = (x - L) \mathrm{mod} (2L) - L \,, \tag{5.6}$$

où L est la moitié de la longueur de la boîte de simulation, qui est fixée à 8.141 dans notre cas. Cette fonction et non-linéaire et permet de générer automatiquement des configurations dans la boîte de simulation. De plus, l'encodeur et le décodeur sont construits de manière à être le plus symétriques possible dans leur architecture, une heuristique souvent employée à la fois dans les VAE que dans les GAN pour stabiliser l'entraînement.

Le discriminateur est un perceptron multicouche, puisque les variables latentes n'ont pas de structure spatiale particulière, avec un nombre de paramètres semblable à ceux de l'encodeur et du décodeur, dans le but d'équilibrer les capacités de tous les réseaux de neurones en jeu dans le modèle. Finalement, une heuristique utilisée dans [57] consiste à aider le discriminateur à discerner la distribution latente du modèle de $\mathcal{N}(0,1)$ en sachant que le discriminateur optimal (pour la divergence de Jensen-Shannon D_{JS}) entre les deux distributions $P_{\mathcal{Z}}$ et Q_{ϕ} est $D^{*} = \log dP_{\mathcal{Z}}(\mathbf{z}) - \log dQ(\mathbf{z}|\mathbf{x},\mathbf{y})$. Comme $p_{\mathcal{Z}}$ est connu, on ne fait apprendre que $\log dQ(\mathbf{z}|\mathbf{x},\mathbf{y})$ au discriminateur soustrayant à sa sortie le terme

$$\log dP_{\mathcal{Z}} = \frac{1}{2}\big(\log(2\pi) + \sum_{i=1}^{\dim z} z_i^2\big) \ . \tag{5.7}$$

Le WAE considéré ici est donc constitué d'un encodeur stochastique avec 274657 paramètres formé de 5 couches convolutives et une couche de sortie linéaire et dense, d'un décodeur déterministe avec 268387 paramètres formé d'une couche d'entrée dense et de 5 couches convolutives, et d'un discriminateur de 234577 paramètres formé de 5 couches denses et d'une sortie sigmoïde. Sauf spécification, toutes les fonctions de transfert sont des rectifieurs linéaires. Au total, le modèle comporte 777621 paramètres d'optimisation. Le modèle a été entraîné pendant 20 époques avec ADAM et les hyper-paramètres présentés au tableau 5.IV, et le pas de gradient pour tous les réseaux de neurones a suivi de plus un échéancier

$$\eta(t) = \frac{\eta}{t^{\frac{1}{4}}} \ , \tag{5.8}$$

où t est le nombre d'itérations de l'AGS. Cet échéancier s'est montré favorable à la bonne convergence du modèle.

L'évolution de l'erreur d'entraînement (excluant le terme de reconstruction) en fonction des itérations est présentée à la figure 5.6. L'entraînement s'est effectué en 10 heures sur un GPU Tesla K80. Après cinq époques, l'entraînement est stable, ce qui signifie que

B	η_{AE}	η_D	β_1	β_2	w	λ	$\dim z$
32	5×10^{-4}	10^{-3}	0.5	0.999	10^{-8}	1	216

TABLEAU 5.IV : Hyper-paramètres utilisés pour l'entraînement du WAE : taille des mini-lots B, pas de gradient pour l'auto-encodeur et le discriminateur η_{AE}, η_D, premier et deuxième facteur d'inertie pour ADAM β_1, β_2, facteur de l'échéancier des paramètres w et dimension de l'espace latent $\dim z$.

l'encodeur a correctement appris à projeter les configurations de $\mathbb{D}$ vers une représentation latente Gaussienne. Cet aspect est très important puisqu'il témoigne de la capacité de l'encodeur à correctement décorréler des variables latentes permettant de représenter le système. De plus si l'encodeur manquait à sa tâche, le décodeur ne pourrait pas apprendre à reconstruire correctement les configurations à partir de variables latentes issues de $\mathcal{N}(0,1)$ et ne pourrait donc pas générer de configurations réalistes à partir de cette distribution.

La stabilité de l'entraînement est cependant contre-balancée par la sensibilité du modèle aux hyper-paramètres d'optimisation. Faute d'un choix approprié de ceux-ci, l'entraînement est souvent instable ou divergent. La réduction du pas de gradient permet d'empêcher au modèle de sortir du minimum local, ce qui permet la génération de structures de basse énergie. Un échéancier trop rapide sur les pas de gradient ralentit ou empêche la convergence du modèle tandis qu'un échéancier trop lent fait le fait immanquablement sortir du minimum.

Une valeur de β_1 trop grande pénalise fortement l'entraînement, il est donc souhaitable d'en garder la valeur dans la plage $[0.5, 0.7]$. Les valeurs idéales de β_2 sont entre 0.9 et 0.999. Les pas de gradient de l'auto-encodeur et du discriminateur ont été ajustés dans la plage $[10^{-4}, 10^{-3}]$, des valeurs plus grandes menant systématiquement à un entraînement divergent et des valeurs plus petites à un entraînement trop long à converger. Finalement, une dimension latente $\dim z$ plus basse qu'environ 200 mène souvent à de fortes instabilités dans les premières époques de l'entraînement accusatoire qui se caractérisent pas de très fortes oscillations des fonctions de coût liées au discriminateur. Le modèle se stabilise éventuellement et a pu générer des configurations réalistes avec

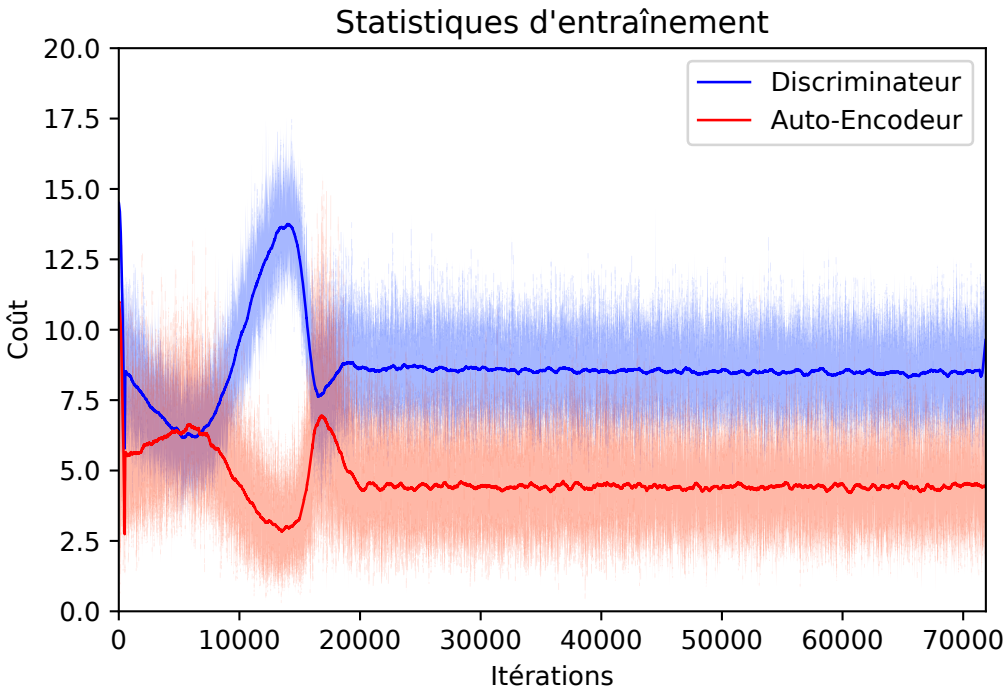

FIGURE 5.6 : Erreur d'entraînement du WAE en fonction des itérations de gradient. Une époque correspond à 3593 itérations. Seule la partie accusatoire de la fonction objectif de l'auto-encodeur $\mathbb{E}_{\mathbf{x}\sim\mathbb{D}} \log f_\gamma(q_\phi(\mathbf{x},\mathbf{y}))$ est illustrée. Les courbes pleines sont le résultat d'un filtrage de hautes fréquences de Savitzky-Golay.

$\dim z = 150$. Des valeurs plus grandes réduiraient la compression d'information devant être apprise par l'encodeur, ce qui pénaliserait la qualité des variables latentes et le modèle en général. Au vu des symétries présentes dans le a-Si, qui soulignent la redondance d'information présente dans les coordonnées atomiques, il est souhaitable de garder un facteur de compression d'information assez grand.

Comme le montre la figure 5.7, l'erreur de reconstruction pour des configurations issues de l'ensemble de validation décroît de manière monotone, malgré la présence de pics épars. Celle-ci semble converger par la suite vers une valeur comparable à distance typique entre les configurations de $\mathbb{D}$.

À la fin de l'entraînement, le modèle génératif est formé uniquement par le décodeur,

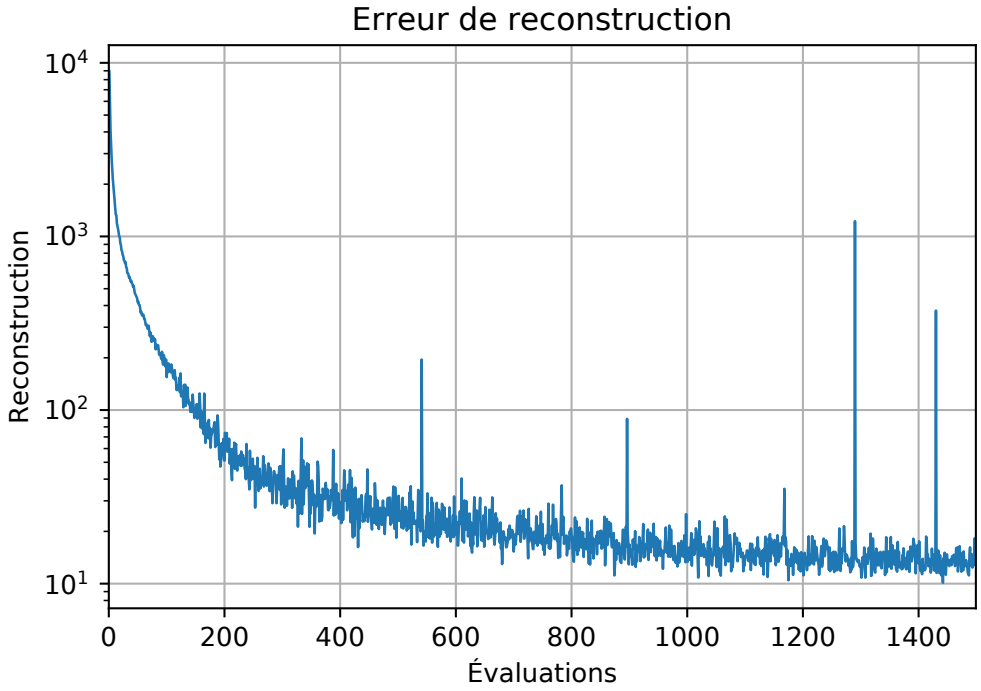

FIGURE 5.7 : Évolution de l'erreur de reconstruction évaluée sur des mini-lots de l'ensemble de validation, en échelle logarithmique. L'erreur a été évaluée à chaque 128 itérations de gradient.

ou générateur, g_θ. Le processus de génération consiste à fournir à g_θ un mini-lot de vecteurs latents $\mathbf{z} \sim \mathcal{N}(0,1)$ ainsi que d'énergies $\mathbf{y} \sim \mathcal{N}(\mu_E, \sigma_E^2)$ où μ_E, σ_E sont la valeur moyenne et l'écart-type des énergies sur tout l'ensemble de données, respectivement.

Un mini-lot de 32 nouvelles configurations est ainsi généré en 122 ms $\pm$ 682 μs, sur une machine locale. La valeur moyenne de l'énergie potentielle de SW des configurations générées est de -226.8 eV avec un écart-type de 48.48 eV. La figure 5.8 montre une configuration générée avec une énergie potentielle de -259.28 eV. Bien que d'énergie supérieure à celle des configurations de l'ensemble de données, la structure est qualitativement satisfaisante. On remarque qu'il n'y a pas d'atomes excessivement rapprochés et que la structure est désordonnée. De plus, le modèle n'est certainement pas en régime de sur-apprentissage puisque dans ce cas il générerait des structures semblables, structurellement et énergétiquement, à celles de l'ensemble de données. Ceci signifie que le

modèle génératif a correctement appris, de manière non supervisée, à produire des structures présentant un ordre local bien défini et une énergie potentielle totale négative. Il s'agit donc d'un signe clair de la convergence partielle de la distribution du générateur vers la distribution des données $p_{\mathcal{X}}$.

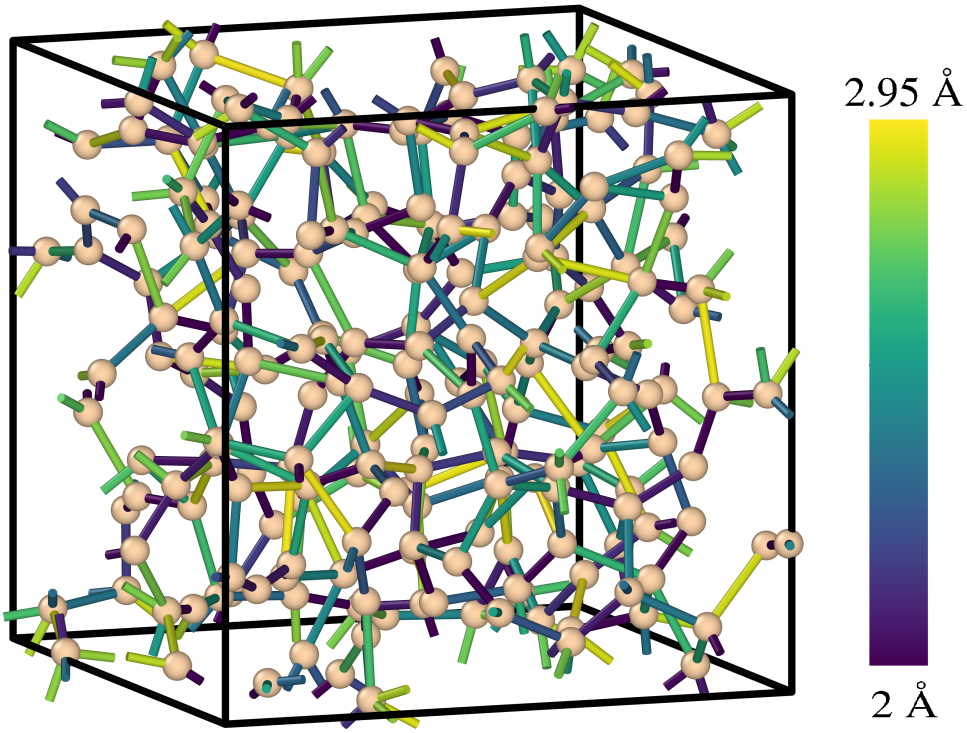

FIGURE 5.8 : Configuration atomique générée par le WAE d'énergie -259.28 eV. Le code de couleur des liens décrit une échelle allant de 2Å à 2.95Å. On remarque un désordre prononcé dans la distribution des longueurs de liaison, mais aucun défaut structurel majeur. La figure a été conçue à l'aide de OVITO [52].

Remarquons cependant que malgré que l'énergie potentielle totale des configurations générées soit significativement plus grande que celle des configurations de $\mathbb{D}$, la distance $d_{\mathcal{G}}$ entre ces dernières est tout à fait comparable à celle entre deux configurations de $\mathbb{D}$.

Ceci est un signe que la distance $d_{\mathcal{G}}$ est défaillante pour ce qui est de discerner précisément deux configurations ayant pourtant des propriétés structurelles différentes. Une raison à cela pourrait être l'absence des fonctions de symétrie $\mathcal{G}_4$ et $\mathcal{G}_5$ dans la description des configurations.

Le code de couleur des liens interatomiques indique cependant qu'une grande partie de voisins se trouvent à des distances plus grandes que 2.35 Å(bleu). Une analyse plus détaillée peut être faite à partir de la FDR de cet échantillon présentée à la figure 5.9. L'absence totale d'atomes voisins à des distances inférieures à 1.6 Åconfirme que le générateur a correctement appris ce critère crucial de la structure atomique du silicium de manière qualitative bien que quantitativement encore imprécise.

Le chevauchement conséquent des deux premiers pics de la FDR est notable et traduit un désordre structurel local trop élevé. Ce chevauchement rend arbitraire la délimitation des premiers et deuxièmes voisins. En effet, si la FDR ne s'annule pas après son premier pic, il n'y a pas de délimitation naturelle entre la première et la deuxième couche de voisins et leur définition n'est plus univoque puisque certains deuxièmes voisins pourraient se retrouver plus proches que certains premiers voisins. Ainsi, le rayon de coupure séparant ces deux couches doit être fixé manuellement, et l'heuristique utilisée ici est celle de considérer la distance r_c s'approchant au mieux à la fois du minimum de la FDR entre les deux premiers pics et du croisement des deux meilleurs ajustements Gaussiens de ces derniers ce qui résulte en $r_c = 2.95$Å.

Les propriétés structurelles issues de ce choix de r_c sont exposées au tableau 5.V. La configuration possède une coordination de premiers voisins C_1 légèrement défaillante de 3.592. Le premier pic est centré à 2.403 Ået l'écart relatif de la distance interatomique est de 14.12%, ce qui illustre un grand désordre dans la première couche de voisins. La deuxième couche de voisins est à une distance moyenne de 4.045 Å, avec un écart-type de 0.795 Ået une coordination $C_2 = 13.55$, assez proche de celle des configurations d'entraînement. Dans cette analyse, les deuxièmes voisins sont définis comme les voi-

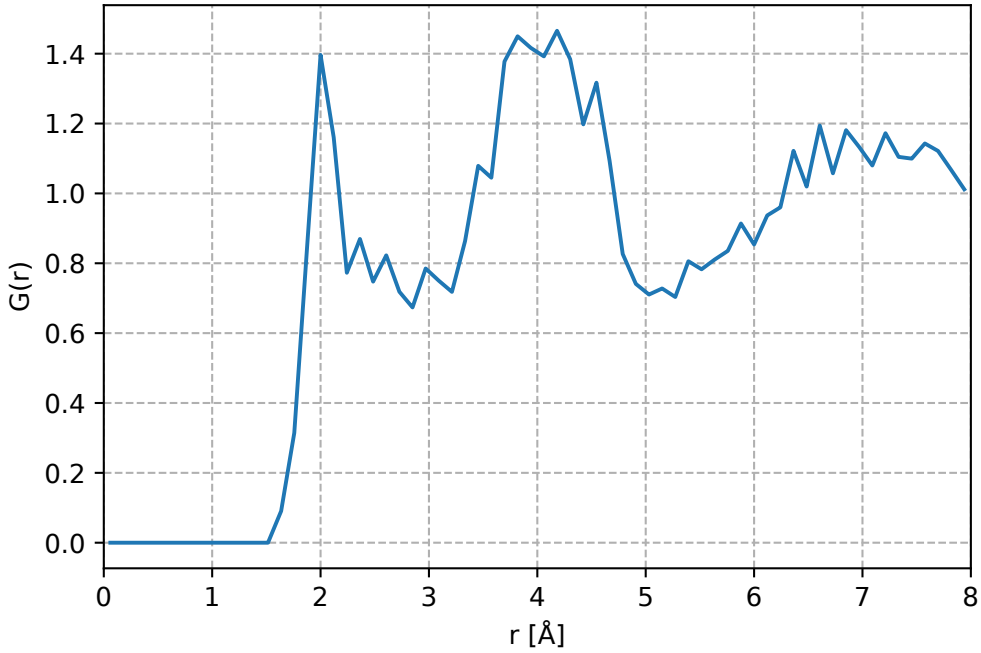

FIGURE 5.9 : Fonction de distribution radiale (FDR) d'un échantillon généré par le WAE après 10 époques. La FDR est normalisée par la densité de l'échantillon (5.00e-2 atomes/Å^3).

	ρ (g/cm^3)	C_1 (at.)	r_1 (Å)	σ_1 (Å)	C_2 (at.)	r_2 (Å)	σ_2 (Å)
WAE	2.334	3.592	2.403	0.339	13.55	4.04	0.795
$\mathbb{D}$	2.334	3.917	2.362	0.074	13.87	3.94	0.446
Exp.	2.285	3.881	2.352	0.065	12.43	3.81	0.238

TABLEAU 5.V : Comparaison des propriétés structurelles des échantillons générés par le WAE, de l'ensemble de données $\mathbb{D}$ et de l'échantillon expérimental recuit [31] : densité (ρ), coordination (C), position des pics (r) et déviation standard (σ) des deux premières couches de voisins, telles que définies par $r_c = 2.95$.

sins des premiers voisins (et donc aucun second rayon de coupure n'a été utilisé). La distribution des angles de liaison qui en résulte est présentée à la figure 5.10. L'angle de liaison moyen est de 108.42° et sont écart-type est de 28.59°. La distribution angulaire présente donc un désordre trop prononcé mais est correctement centrée proche de l'angle

tétraédrique. Remarquons cependant que cette caractérisation structurelle dépend fortement du choix de r_c, et peut donc changer quantitativement en fonction de celui-ci.

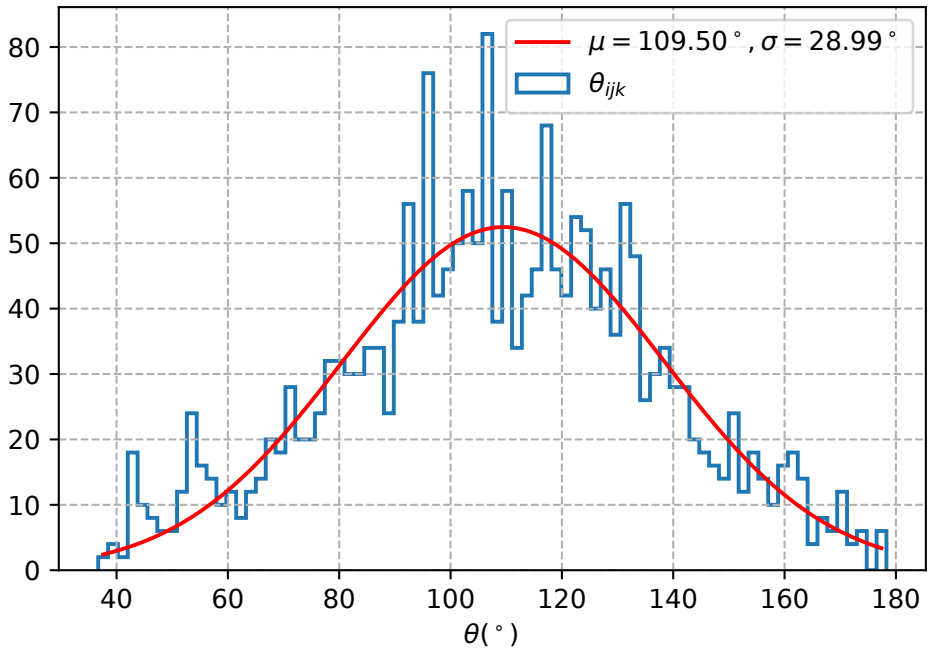

FIGURE 5.10 : Distribution des angles de liaison entre premiers voisins et ajustement Gaussien de celle-ci. La valeur moyenne et l'écart-type de l'ajustement sont comparables à celles de la distribution i.e. 108.42° et 28.59° respectivement.

Ainsi les configurations générées par le modèle présentent une structure qualitativement réaliste avec des valeurs moyennes des distances interatomiques et des angles de liaison proches de celles du silicium amorphe, bien que trop désordonnée par rapport aux configurations de $\mathbb{D}$, qui se traduit par un écart-type des distances interatomiques et des angles de liaison trop élevé.

Plusieurs raisons peuvent expliquer ces considérations. La plus importante est l'absence de description angulaire, fournie par $\mathcal{G}_4$ et $\mathcal{G}_5$, dans la représentation par les fonctions de symétrie. En effet, la défaillance de $d_{\mathcal{G}}$ quant à discerner des configurations

énergétiquement distantes de presque 400 eV, comme c'est le cas entre les configurations générées par le modèle et les configurations d'entraînement, souligne un problème dans la description des structures atomiques. L'intégration de fonctions de symétrie angulaires de type $\mathcal{G}_4$ et $\mathcal{G}_5$ dans le modèle se traduit malheureusement par une augmentation drastique du temps de calcul qui rend l'entraînement du WAE très difficile à mettre en oeuvre en pratique.

Un deuxième enjeu possible est celui de la richesse de l'ensemble de données $\mathbb{D}$. Les configurations générées par ART sont fortement corrélées et peu de configurations issues de simulations de dynamique moléculaire sont présentes dans le jeu de données comparé aux travaux de Behler et Parrinello. En particulier, l'ajout d'une plus vaste gamme de structures à différentes températures et pressions serait éventuellement bénéfique à la variété de l'ensemble de données et aurait un impact bénéfique sur l'apprentissage. La forte corrélation entre les configurations générées par ART n'est pas a priori problématique tant qu'un assez grand nombre de celles-ci est présent dans l'ensemble de manière à garantir une bonne variété de structures.

Finalement, deux autres facteurs peuvent influer sur la qualité du modèle. La convergence optimale de l'entraînement est difficile à atteindre puisqu'elle dépend sensiblement du choix des hyper-paramètres, comme l'heuristique utilisée pour l'échéancier des pas de gradient. Notamment, il est aisé de faire sortir le modèle d'un bon minimum local avec un entraînement trop prolongé ou un échéancier trop lent.

De plus, notons que le WAE n'approxime en aucun cas la SEP. Bien que l'information des énergies potentielles ait été intégrée au modèle, il est possible qu'elle ne soit pas suffisamment exploitée et que ceci impacte négativement la convergence de celui-ci, puisqu'il est bien connu que l'apprentissage non supervisé est un tâche vastement plus ardue que l'apprentissage supervisé.

Pour conclure, l'entraînement d'un WAE exploitant l'information des énergies potentielles totales et utilisant une mesure d'erreur de reconstruction basée sur la distance

euclidienne entre les fonctions de symétrie des configurations a été mené a bien. L'entraînement accusatoire se révèle être stable ce qui signifie que le modèle ne présente pas de difficultés à encoder les configurations atomiques en un vecteur latent suivant une distribution Gaussienne unitaire. Cependant, la mesure de distance utilisée ne permet pas de distinguer des configurations structurellement différentes ayant pourtant des énergies potentielles totales sensiblement éloignées. Le modèle génératif profond ainsi entraîné est cependant prometteur du fait que les configurations générées présentent une structure qualitativement correcte et une énergie potentielle totale du même ordre de grandeur que les configurations générées par ART dans l'ensemble d'entraînement. Des pistes d'amélioration sont considérées, comme une description plus complète de la structure atomique ou l'utilisation d'un ensemble de données plus riche.

CONCLUSION ET PERSPECTIVES

Le développement de modèles alternatifs à la simulation de la structure atomique du silicium amorphe basés sur des algorithmes d'apprentissage profond a été exploré dans ce travail. Le problème consistait à développer un modèle génératif profond, combinant efficacité computationnelle et réalisme physique, pour fournir une preuve de concept de la viabilité de cette approche. En premier lieu, un ensemble de configurations de 216 atomes générées par la méthode ART et la dynamique moléculaire avec le potentiel de Stillinger-Weber a été formé pour l'entraînement de différents réseaux de neurones. L'impossibilité de mener à bien une tâche d'apprentissage en se servant directement des coordonnées cartésiennes des configurations atomiques a été soulignée, et une représentation convenable de celles-ci a été présentée. La représentation par des fonctions de symétrie, introduite par Behler et Parrinello, permet de décrire l'environnement local des atomes de manière naturellement invariante aux opérations de symétrie inhérentes aux configurations atomiques comme les transformations orthogonales et les permutations d'indices atomiques.

L'efficacité de cette représentation a été validée en effectuant l'apprentissage de la surface d'énergie potentielle du potentiel de SW sur l'ensemble de données généré. Un réseau de neurones convolutif modérément profond a présenté une RMSE de 1.101 eV par configuration, soit 5.095 meV par atome, une performance qui se compare bien à celle des travaux de Behler et Parinnello pour des systèmes de 64 atomes.

Par la suite un WAE conditionnel a été entraîné pour la modélisation de la structure atomique du silicium amorphe. Les configurations générées ont une énergie potentielle moyenne de -226.8 eV, soit presque 400 eV de plus que l'énergie potentielle moyenne de l'ensemble de données. Cette différence en énergie explicite le bon accord qualitatif des structures générées avec celles de l'ensemble d'entraînement. Dans le but de comparer précisément les caractéristiques structurelles des configurations obtenues, un échantillon généré d'énergie de -259.28 eV a été analysé.

La FDR de l'échantillon présente un chevauchement conséquent de ses deux premiers pics, et un rayon de coupure a été déterminé à $r_c = 2.95$. La distribution des distances interatomiques a une moyenne $r_1 = 2.403$ Å et un écart-type $\sigma_1 = 0.339$ Å, ce qui signifie que la première couche de voisins est trop éloignée et éparse. Celle-ci est également légèrement sous-coordonnée avec une coordination $C_1 = 3.592$. La deuxième couche de voisins est centrée en $r_2 = 4.04$ Å avec un écart-type de $\sigma_2 = 0.795$ Å. La distributions des angles de liaison, centrée en $\theta_0 = 108.42^\circ$ avec un écart-type de $\sigma_\theta = 28.59^\circ$ est légèrement décentrée et trop large. La deuxième couche de voisins aussi légèrement sous-coordonnée avec une coordination $C_2 = 13.55$.

Un premier facteur pouvant expliquer la convergence de l'algorithme vers cette distribution de configurations provient du fait que la mesure de distance permettant de calculer l'erreur de reconstruction lors de l'entraînement du WAE discerne mal la différence entre ces configurations structurellement peu relaxées et une configuration de l'ensemble de données. Ceci peut s'expliquer du fait de l'omission, pour des raisons de performance, des fonctions de symétrie angulaires dans la description des environnements locaux atomiques utilisée dans la définition de ladite distance.

Un deuxième facteur est celui de la richesse et de la qualité de l'ensemble de données utilisé pour l'entraînement du modèle. Celui-ci est formé majoritairement de configurations générées par ART fortement corrélées et en partie de simulations de dynamique moléculaire à 900 K avec un pas de temps de 1 fs. La grande corrélation entre les configurations issues de ART n'est pas a priori un problème pour l'apprentissage. Le petit nombre de configurations issues de la MD et leur pauvre variété en termes de température et de pression peut cependant défavoriser la richesse du jeu de données et donc la qualité de l'apprentissage.

En conclusion, cette étude constitue une preuve de concept pour l'application des modèles génératifs profonds à la simulation de la structure atomique du silicium amorphe. L'approche s'avère être viable, et bien que les structures générées ne soient pas aussi relaxées et réalistes que celles de l'ensemble de données utilisé pour l'entraînement, ce

travail a démontré des signes clairs qu'une meilleure convergence est possible.

Dans un travail futur, ces modèles génératifs pourraient être modifiés de manière à être indépendants du nombre d'atomes. Comme la structure des matériaux désordonnés est principalement locale, cette méthode ouvrirait la voie à l'apprentissage sur des petites configurations pour la génération de configurations de plus grande taille au travers d'une procédure auto-cohérente. Un modèle initialement entraîné sur des configurations de tailles variées pourrait apprendre progressivement à générer des configurations de taille toujours plus grande, ce qui ouvrirait la voie à la génération efficace et réaliste de très grands systèmes. Bien sûr, rendre le modèle indépendant du nombre d'atomes tout en restant efficace pour la génération de configurations est une tâche difficile, mais certainement intéressante à explorer.

BIBLIOGRAPHIE

[1] John Aldrich. R.a. fisher and the making of maximum likelihood 1912-1922. *Statist. Sci.*, 12(3) :162–176, 09 1997. doi : 10.1214/ss/1030037906. URL https://doi.org/10.1214/ss/1030037906.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *ArXiv e-prints*, January 2017.

[3] G. T. Barkema and Normand Mousseau. Event-based relaxation of continuous disordered systems. *Phys. Rev. Lett.*, 77 :4358–4361, Nov 1996. doi : 10.1103/PhysRevLett.77.4358. URL https://link.aps.org/doi/10.1103/PhysRevLett.77.4358.

[4] G. T. Barkema and Normand Mousseau. High-quality continuous random networks. *Phys. Rev. B*, 62 :4985–4990, Aug 2000. doi : 10.1103/PhysRevB.62.4985. URL https://link.aps.org/doi/10.1103/PhysRevB.62.4985.

[5] Robert Bassett and Julio Deride. Maximum a posteriori estimators as a limit of bayes estimators. *Mathematical Programming*, Jan 2018. ISSN 1436-4646. doi : 10.1007/s10107-018-1241-0. URL https://doi.org/10.1007/s10107-018-1241-0.

[6] Jörg Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of Chemical Physics*, 134(7) :074106, 2011. doi : 10.1063/1.3553717. URL https://doi.org/10.1063/1.3553717.

[7] Jörg Behler. Neural network potential-energy surfaces in chemistry : a tool for large-scale simulations. *Phys. Chem. Chem. Phys.*, 13 :17930–17955, 2011. doi : 10.1039/C1CP21668F. URL http://dx.doi.org/10.1039/C1CP21668F.

[8] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98 :146401, Apr

2007. doi : 10.1103/PhysRevLett.98.146401. URL https://link.aps.org/doi/10.1103/PhysRevLett.98.146401.

[9] R. Bellman, Rand Corporation, and Karreman Mathematics Research Collection. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957. ISBN 9780691079516.

[10] R. Bellman, R.E. Bellman, and Karreman Mathematics Research Collection. *Adaptive Control Processes : A Guided Tour*. Princeton Legacy Library. Princeton University Press, 1961.

[11] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1) :1–127, January 2009. ISSN 1935-8237. doi : 10.1561/2200000006. URL http://dx.doi.org/10.1561/2200000006.

[12] Parthapratim Biswas, Raymond Atta-Fynn, and D Drabold. Reverse monte carlo modeling of amorphous silicon. 69, 01 2004.

[13] N. Cusack. *The physics of structurally disordered matter : an introduction*. Graduate Student Series in Physics Series. A. Hilger, 1987. ISBN 9780852745915.

[14] Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, et al. Lasagne : First release., August 2015. URL http://dx.doi.org/10.5281/zenodo.27878.

[15] P. Engel. *Geometric Crystallography : An Axiomatic Introduction to Crystallography*. Springer Netherlands, 1986. ISBN 9789027723413.

[16] A. Filipponi, F. Evangelisti, M. Benfatto, S. Mobilio, and C. R. Natoli. Structural investigation of a-si and a-si :h using x-ray-absorption spectroscopy at the si k edge. *Phys. Rev. B*, 40 :9636–9643, Nov 1989. doi : 10.1103/PhysRevB.40.9636. URL https://link.aps.org/doi/10.1103/PhysRevB.40.9636.

[17] Julian H. Gibbs and Edmund A. DiMarzio. Nature of the glass transition and the glassy state. *The Journal of Chemical Physics*, 28(3) :373–383, 1958. doi : 10.1063/1.1744141. URL https://doi.org/10.1063/1.1744141.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[20] Ian J. Goodfellow. NIPS 2016 tutorial : Generative adversarial networks. *CoRR*, abs/1701.00160, 2017. URL http://arxiv.org/abs/1701.00160.

[21] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[23] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017. URL http://arxiv.org/abs/1706.08500.

[24] Matthew Hirn, Nicolas Poilvert, and Stéphane Mallat. Quantum energy regression using scattering transforms. *CoRR*, abs/1502.02077, 2015. URL http://arxiv.org/abs/1502.02077.

[25] Behler Jörg, Martoňák Roman, Donadio Davide, and Parrinello Michele. Pressure-induced phase transitions in silicon studied by neural network-based metadynamics simulations. *physica status solidi (b)*, 245(12) :2618–2629. doi : 10.1002/pssb.200844219. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/pssb.200844219.

[26] Walter. Kauzmann. The nature of the glassy state and the behavior of liquids at low temperatures. *Chemical Reviews*, 43(2) :219–256, 1948. doi : 10.1021/cr60135a002. URL https://doi.org/10.1021/cr60135a002.

[27] P. N. Keating. Effect of invariance requirements on the elastic strain energy of crystals with application to the diamond structure. *Phys. Rev.*, 145 :637–645, May 1966. doi : 10.1103/PhysRev.145.637. URL https://link.aps.org/doi/10.1103/PhysRev.145.637.

[28] Rustam Z. Khaliullin, Hagai Eshet, Thomas D. Kühne, Jörg Behler, and Michele Parrinello. Graphite-diamond phase coexistence study employing a neural-network mapping of the ab initio potential energy surface. *Phys. Rev. B*, 81 :100103, Mar 2010. doi : 10.1103/PhysRevB.81.100103. URL https://link.aps.org/doi/10.1103/PhysRevB.81.100103.

[29] Rustam Z. Khaliullin, Hagai Eshet, Thomas D. Kühne, Jörg Behler, and Michele Parrinello. Graphite-diamond phase coexistence study employing a neural-network mapping of the ab initio potential energy surface. *Phys. Rev. B*, 81 :100103, Mar 2010. doi : 10.1103/PhysRevB.81.100103. URL https://link.aps.org/doi/10.1103/PhysRevB.81.100103.

[30] D. P Kingma and M. Welling. Auto-Encoding Variational Bayes. *ArXiv e-prints*, December 2013.

[31] Khalid Laaziri, S. Kycia, S. Roorda, M. Chicoine, J. L. Robertson, J. Wang, and S. C. Moss. High-energy x-ray diffraction study of pure amorphous silicon. *Phys.*

Rev. B, 60 :13520–13533, Nov 1999. doi : 10.1103/PhysRevB.60.13520. URL https://link.aps.org/doi/10.1103/PhysRevB.60.13520.

[32] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. URL http://arxiv.org/abs/1711.05101.

[33] P. Luc, C. Couprie, S. Chintala, and J. Verbeek. Semantic Segmentation using Adversarial Networks. *ArXiv e-prints*, November 2016.

[34] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial Autoencoders. *ArXiv e-prints*, November 2015.

[35] Rachid Malek and Normand Mousseau. Dynamics of lennard-jones clusters : A characterization of the activation-relaxation technique. *Phys. Rev. E*, 62 :7723–7728, Dec 2000. doi : 10.1103/PhysRevE.62.7723. URL https://link.aps.org/doi/10.1103/PhysRevE.62.7723.

[36] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least Squares Generative Adversarial Networks. *ArXiv e-prints*, November 2016.

[37] Masakazu Matsugu, Katsuhiko Mori, Yusuke Mitari, and Yuji Kaneda. Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, 16(5) : 555 – 559, 2003. ISSN 0893-6080. doi : https://doi.org/10.1016/S0893-6080(03)00115-1. URL http://www.sciencedirect.com/science/article/pii/S0893608003001151. Advances in Neural Networks Research : IJCNN '03.

[38] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4) :115–133, Dec 1943. ISSN 1522-9602. doi : 10.1007/BF02478259. URL https://doi.org/10.1007/BF02478259.

[39] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072.

[40] Normand Mousseau and G. T. Barkema. Traveling through potential energy landscapes of disordered materials : The activation-relaxation technique. *Phys. Rev. E*, 57 :2419–2424, Feb 1998. doi : 10.1103/PhysRevE.57.2419. URL https://link.aps.org/doi/10.1103/PhysRevE.57.2419.

[41] K.P. Murphy and F. Bach. *Machine Learning : A Probabilistic Perspective*. Adaptive Computation and Machine learning. MIT Press, 2012. ISBN 9780262018029.

[42] D.R. Nelson. *Defects and Geometry in Condensed Matter Physics*. Cambridge University Press, 2002. ISBN 9780521004008.

[43] S. Nowozin, B. Cseke, and R. Tomioka. f-GAN : Training Generative Neural Samplers using Variational Divergence Minimization. *ArXiv e-prints*, June 2016.

[44] Kyoung-Su Oh and Keechul Jung. Gpu implementation of neural networks. *Pattern Recognition*, 37(6) :1311 – 1314, 2004. ISSN 0031-3203. doi : https://doi.org/10.1016/j.patcog.2004.01.013. URL http://www.sciencedirect.com/science/article/pii/S0031320304000524.

[45] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12 :145 – 151, 1999. ISSN 0893-6080. doi : https://doi.org/10.1016/S0893-6080(98)00116-6. URL http://www.sciencedirect.com/science/article/pii/S0893608098001166.

[46] A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *ArXiv e-prints*, November 2015.

[47] L. J. Ratliff, S. A. Burden, and S. Shankar Sastry. On the Characterization of Local Nash Equilibria in Continuous Games. *ArXiv e-prints*, November 2014.

[48] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, page 400 – 407, 1951.

[49] F. Rosenblatt. The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

[50] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Parallel distributed processing : Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986. ISBN 0-262-68053-X. URL http://dl.acm.org/citation.cfm?id=104279.104293.

[51] Frank H. Stillinger and Thomas A. Weber. Computer simulation of local order in condensed phases of silicon. *Phys. Rev. B*, 31 :5262–5271, Apr 1985. doi : 10.1103/PhysRevB.31.5262. URL https://link.aps.org/doi/10.1103/PhysRevB.31.5262.

[52] Alexander Stukowski. Visualization and analysis of atomistic simulation data with ovito–the open visualization tool. *Modelling and Simulation in Materials Science and Engineering*, 18(1) :015012. URL http://stacks.iop.org/0965-0393/18/i=1/a=015012.

[53] R. Ramprasad T. Mueller, A. G. Kusne. Machine learning in materials science : Recent progress and emerging applications. *Reviews in Computational Chemistry*, 2016.

[54] J. Tersoff. New empirical approach for the structure and energy of covalent systems. *Phys. Rev. B*, 37 :6991–7000, Apr 1988. doi : 10.1103/PhysRevB.37.6991. URL https://link.aps.org/doi/10.1103/PhysRevB.37.6991.

[55] J. Tersoff. Empirical interatomic potential for silicon with improved elastic properties. *Phys. Rev. B*, 38 :9902–9905, Nov 1988. doi : 10.1103/PhysRevB.38.9902. URL https://link.aps.org/doi/10.1103/PhysRevB.38.9902.

[56] The Theano Development Team, R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky, Y. Bengio, A. Bergeron, J. Bergstra, V. Bisson, J. Bleecher Snyder, N. Bouchard, N. Boulanger-Lewandowski, X. Bouthillier, A. de Brébisson, O. Breuleux, P.-L. Carrier, K. Cho, J. Chorowski, P. Christiano, T. Cooijmans, M.-A. Côté, M. Côté, A. Courville, Y. N. Dauphin, O. Delalleau, J. Demouth, G. Desjardins, S. Dieleman, L. Dinh, M. Ducoffe, V. Dumoulin, S. Ebrahimi Kahou, D. Erhan, Z. Fan, O. Firat, M. Germain, X. Glorot, I. Goodfellow, M. Graham, C. Gulcehre, P. Hamel, I. Harlouchet, J.-P. Heng, B. Hidasi, S. Honari, A. Jain, S. Jean, K. Jia, M. Korobov, V. Kulkarni, A. Lamb, P. Lamblin, E. Larsen, C. Laurent, S. Lee, S. Lefrancois, S. Lemieux, N. Léonard, Z. Lin, J. A. Livezey, C. Lorenz, J. Lowin, Q. Ma, P.-A. Manzagol, O. Mastropietro, R. T. McGibbon, R. Memisevic, B. van Merriënboer, V. Michalski, M. Mirza, A. Orlandi, C. Pal, R. Pascanu, M. Pezeshki, C. Raffel, D. Renshaw, M. Rocklin, A. Romero, M. Roth, P. Sadowski, J. Salvatier, F. Savard, J. Schlüter, J. Schulman, G. Schwartz, I. Vlad Serban, D. Serdyuk, S. Shabanian, É. Simon, S. Spieckermann, S. Ramana Subramanyam, J. Sygnowski, J. Tanguay, G. van Tulder, J. Turian, S. Urban, P. Vincent, F. Visin, H. de Vries, D. Warde-Farley, D. J. Webb, M. Willson, K. Xu, L. Xue, L. Yao, S. Zhang, and Y. Zhang. Theano : A Python framework for fast computation of mathematical expressions. *ArXiv e-prints*, May 2016.

[57] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. Wasserstein Auto-Encoders. *ArXiv e-prints*, November 2017.

[58] F. P. Cantelli V. Glivenko. Sulla determinazione empirica delle leggi di probabilità. *Giorn. Ist. Ital. Attuari*, 1933.

[59] Chervonenkis Vapnik. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, pages 264–280, 1971.

[60] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, New

York, NY, USA, 2 edition, 2000. ISBN 978-0-387-98780-4. doi : 10.1007/978-1-4757-3264-1.

[61] R.L.C. Vink, G.T. Barkema, W.F. van der Weg, and Normand Mousseau. Fitting the stillinger–weber potential to amorphous silicon. *Journal of Non-Crystalline Solids*, 282(2) :248 – 255, 2001. ISSN 0022-3093. doi : https://doi.org/10.1016/S0022-3093(01)00342-8. URL http://www.sciencedirect.com/science/article/pii/S0022309301003428.

[62] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *Trans. Evol. Comp*, 1(1) :67–82, April 1997. ISSN 1089-778X. doi : 10.1109/4235.585893. URL https://doi.org/10.1109/4235.585893.

[63] F. Wooten, K. Winer, and D. Weaire. Computer generation of structural models of amorphous si and ge. *Phys. Rev. Lett.*, 54 :1392–1395, Apr 1985. doi : 10.1103/PhysRevLett.54.1392. URL https://link.aps.org/doi/10.1103/PhysRevLett.54.1392.

[64] W. H. Zachariasen. The atomic arrangement in glass. *Journal of the American Chemical Society*, 54(10) :3841–3851, 1932. doi : 10.1021/ja01349a006. URL https://doi.org/10.1021/ja01349a006.

[65] R. Zallen. *The Physics of Amorphous Solids*. Wiley-Interscience publication. John Wiley & Sons, 2008.

Annexe I

Algorithme du gradient avec inertie et ADAM

Il est commun de visualiser l'hypersurface de la fonction objectif $\mathcal{L}$ en $k+1$ dimensions comme une surface sur laquelle se déplace le modèle. Ce déplacement est alors caractérisé par un vecteur $\theta \in \mathbb{R}^k$ dont la dynamique est régie par l'algorithme d'optimisation utilisé, qui correspond aux lois physiques du mouvement du système.

L'algorithme du gradient présente des problèmes lorsque l'hypersurface présente un ravin abrupte (comme il arrive souvent près d'un minimum local) : le système se met à osciller sur les bords de celui-ci en ne faisant que peu de progrès vers le minimum. De plus, lorsque l'hypersurface présente des plateaux de faible pente l'algorithme du gradient converge très lentement, n'ayant pas d'équivalent d'une accélération. L'algorithme du gradient avec inertie [45] aide à contrer ce problème en ajoutant à la règle de mise à jour un terme correspondant à l'impulsion du système

$$\mathbf{v}_t = \gamma \mathbf{v}_{t-1} + \eta \nabla_\theta \mathcal{L}(\theta) \tag{I.1}$$

$$\theta_{t+1} = \theta_t - \mathbf{v}_t \ . \tag{I.2}$$

Le paramètre γ a une valeur de 0.9 par défaut dans la plupart des implémentations. Le terme d'inertie ajoute donc une fraction du pas de gradient précédent au pas actuel, qui correspond à une moyenne mobile exponentielle de tous les gradients précédents. Ceci provoque une accélération dans les régions de plateaux de $\mathcal{L}$ et permet d'amortir les oscillations autour des ravins pour recentrer le mouvement vers le minimum. L'algorithme du gradient avec inertie est en fait équivalent au mouvement discret d'une particule de massive dans un champ de force conservatif avec un temps discret si on applique la

correspondance

$$\eta = \frac{(\Delta t)^2}{m + \mu \Delta t} \tag{I.3a}$$

$$\gamma = \frac{m}{m + \mu \Delta t} \,, \tag{I.3b}$$

où m est la masse de la particule, μ est le coefficient de friction et le champ de forces est donné par $\nabla_\theta \mathcal{L}$.

Un autre algorithme du gradient très populaire est ADAM (*Adaptive Moment Estimation*), qui calcule un estimé du premier (moyenne) et second moment (variance non centrée) des gradients ce qui revient à attribuer un pas de gradient différent pour chaque paramètre. Ces estimés sont donnés respectivement par

$$\mathbf{m}_t = \frac{1}{1-\beta_1^t}\left(\beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t\right) \tag{I.4a}$$

$$\mathbf{v}_t = \frac{1}{1-\beta_2^t}\left(\beta_2 \mathbf{v}_{t-1} + (1-\beta_2)\mathbf{g}_t^2\right) \,, \tag{I.4b}$$

où $g_t = \nabla_{\theta_t} \mathcal{L}$. Les paramètres β_1 et β_2 contrôlent le taux d'amortissement des moyennes mobiles exponentielles des gradients et des gradients carrés respectivement. Comme m_t et v_t sont initialisés à zéro, ces moments sont biaisés vers zéro lors des premiers pas de temps surtout lorsque β_1 et β_2 sont proches de 1. Ce biais est contré par les termes $\frac{1}{1-\beta_1}$ et $\frac{1}{1-\beta_2}$ dans la définition des estimés des moments.
La règle de mise à jour de ADAM est alors

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\mathbf{v}_t} + \varepsilon}\mathbf{m}_t \,, \tag{I.5}$$

où ε est un facteur de stabilité numérique. ADAM peut aussi être interprété comme le mouvement discret d'une boule massive dans un champ de force conservatif avec friction. En effet cette dynamique est décrite par l'équation

$$\ddot{\theta}_t + a(t)\dot{\theta}_t + \nabla f(\theta_t) = \mathbf{0} \,. \tag{I.6}$$

Si $r(n) = r$ ou $r(n) = r/\sum_{l=1}^{n} a(l)$ et $a(n) = an^{-\tau}$ pour $\tau \in (0,1]$, la correspondance est alors donnée par $\beta_1 = 1 - a(n+1)r(n)$, $\beta_2 = 1 - \alpha a(n+1)r(n)$ et $f = \mathcal{L}$ si on considère de plus que les gradients ont un second moment stationnaire i.e. $\mathbf{v} = \mathbb{E}[(\nabla f)^2]$ [23].

Il est commun de définir un échéancier sur le pas de gradient des algorithmes du gradient de manière à assurer une convergence stable vers un minimum. L'échéancier doit respecter les conditions [48]

$$\sum_{i=1}^{\infty} \eta_i = \infty \tag{I.7a}$$

$$\sum_{i=1}^{\infty} \eta_i^2 < \infty \,, \tag{I.7b}$$

de manière à remplir les conditions de convergence dans le cas de l'optimisation convexe. Un échéancier très commun, utilisé avec succès sur ADAM est la règle

$$\eta(t) = \frac{\eta}{\sqrt{t}} \,, \tag{I.8}$$

qui respecte effectivement les conditions I.7.

Une autre modification qui peut améliorer la performance de ADAM est l'incorporation d'un échéancier sur les paramètres. Bien qu'un échéancier sur les paramètres soit équivalent à la régularisation L^2 pour le cas de l'algorithme du gradient standard, ce n'est plus le cas pour ADAM [32]. La règle de mise à jour de ADAM avec un échéancier sur le pas de gradient et sur les paramètres devient alors

$$\theta_{t+1} = \theta_t - \eta_t \left(\frac{\mathbf{m}_t}{\sqrt{\mathbf{v}_t} + \varepsilon} + \lambda \theta_{t-1} \right) . \tag{I.9}$$

En particulier, [32] montrent que l'ajout de l'échéancier sur les paramètres plutôt qu'un implémentation correcte de la régularisation L^2 est préférable pour les algorithmes adap-

tatifs comme ADAM du fait que le premier est une version normalisée de manière adaptative du second. C'est cette version de l'algorithme de ADAM qui est utilisée tout au long de ce travail.

Annexe II

Algorithmes du GAN, WGAN, VAE et WAE

Algorithme 1 : GAN.

Require : Hyper-paramètres : Taille des mini-lots $m = 32$, itérations du discriminateur $k = 1$, pas de gradient $\eta = 0.001$, premier facteur d'inertie $\beta_1 = 0.5$, deuxième facteur d'inertie $\beta_2 = 0.999$.

Require : θ_0, les paramètres initiaux du générateur. ϕ_0, les paramètres initiaux du discriminateur.

for *k itérations* **do**

  Tirer un mini-lot de m échantillons de la loi antérieure $\{\mathbf{z}^{(i)}\}_{i=1}^{i=m} \sim p_{\mathbb{Z}}$

  Tirer un mini-lot de m exemples $\{\mathbf{x}^{(i)}\}_{i=1}^{i=m} \sim p_{\mathbb{D}}$

  $\Phi \leftarrow \nabla_{\phi} \frac{1}{m} \sum_{i=1}^{i=m} \left[\log f_{\phi}\left(\mathbf{x}^{(i)}\right) + \log\left(1 - f_{\phi} \circ g_{\theta}(\mathbf{z}^{(i)})\right)\right]$

  $\phi \leftarrow \phi + \eta \mathrm{ADAM}_{\beta_1,\beta_2}(\phi, \Phi)$

  Tirer un mini-lot de m échantillons de la loi antérieure $\{\mathbf{z}^{(i)}\}_{i=1}^{i=m} \sim p_{\mathbb{Z}}$

  $\theta \leftarrow \nabla_{\theta} \frac{1}{m} \sum_{i=1}^{i=m} \left[\log\left(1 - f_{\phi} \circ g_{\theta}(\mathbf{z}^{(i)})\right)\right]$

  $\theta \leftarrow \theta - \eta \mathrm{ADAM}_{\beta_1,\beta_2}(\theta, \Theta)$

end

Algorithme 2 : WGAN.

Le critique f_ω doit avoir une couche de sortie linéaire.

Require : Hyper-paramètres : Taille des mini-lots $m = 32$, itérations du discriminateur $k = 5$, pas de gradient $\eta = 0.0001$, premier facteur d'inertie $\beta_1 = 0.5$, deuxième facteur d'inertie $\beta_2 = 0.999$, le paramètre de bruit $c = 0.5$, le paramètre de régularisation $\lambda = 10$.

Require : θ_0, les paramètres initiaux du générateur. ω_0, les paramètres initiaux du critique.

for *k itérations* **do**

- Tirer un mini-lot de m échantillons de la loi antérieure $\{\mathbf{z}^{(i)}\}_{i=1}^{i=m} \sim p_{\mathbb{Z}}$
- Tirer un mini-lot de m exemples $\{\mathbf{x}^{(i)}\}_{i=1}^{i=m} \sim p_{\mathbb{D}}$ et calculer sa déviation standard σ
- Tirer $\alpha \sim U(0,1)$ et $\varepsilon \sim U(0,1)^m$
- Former un mini-lot de bruit $\{\delta^{(i)} = c\sigma\varepsilon^{(i)}\}_{i=1}^{i=m}$
- Former un mini-lot bruité $\{\hat{\mathbf{x}}^{(i)} = \alpha\mathbf{x}^{(i)} + (1-\alpha)(\mathbf{x}^{(i)} + \delta^{(i)})\}_{i=1}^{i=m}$
- $\Omega \leftarrow \nabla_\omega \frac{1}{m} \sum_{i=1}^{i=m} \left[f_\omega\left(g_\theta(\mathbf{z}^{(i)}) - f_\omega\left(\mathbf{x}^{(i)}\right)\right) + \lambda \max\left\{0, \nabla_{\hat{\mathbf{x}}} \| f_\omega(\hat{\mathbf{x}}^{(i)}) \| - 1\right\}^2 \right]$
- $\omega \leftarrow \omega - \eta \mathrm{ADAM}_{\beta_1,\beta_2}(\omega, \Omega)$
- Tirer un mini-lot de m échantillons de la loi antérieure $\{\mathbf{z}^{(i)}\}_{i=1}^{i=m} \sim p_{\mathbb{Z}}$
- $\Theta \leftarrow \nabla_\theta - \frac{1}{m} \sum_{i=1}^{i=m} \left[f_\omega\left(g_\theta(\mathbf{z}^{(i)})\right) \right]$
- $\theta \leftarrow \theta - \eta \mathrm{ADAM}_{\beta_1,\beta_2}(\theta, \Theta)$

end

Algorithme 3 : VAE.

Require : Hyper-paramètres : Taille des mini-lots $m = 32$, dimension de l'espace latent z, pas de gradient $\eta = 0.001$, premier facteur d'inertie $\beta_1 = 0.5$, deuxième facteur d'inertie $\beta_2 = 0.999$.

Require : ϕ_0, les paramètres initiaux de l'encodeur. θ_0, les paramètres initiaux du décodeur.

for *k itérations* **do**

- Tirer un mini-lot de m exemples $\{\mathbf{x}^{(i)}\}_{i=1}^{i=m} \sim \mathbb{D}$
- Propager à travers l'encodeur $\{\mu^{(i)}, \Sigma^{(i)} = q_\phi(\mathbf{x}^{(i)})\}_{i=1}^{i=m}$
- Tirer un mini-lot $\{\varepsilon^{(i)} \sim \mathcal{N}(0,1)\}_{i=1}^{i=m}$
- Échantillonner les variables latentes $\{\mathbf{z}^{(i)} = \mu^{(i)} + \varepsilon^{(i)}\Sigma^{(i)}\}_{i=1}^{i=m}$
- Propager à travers le décodeur $\{\tilde{\mathbf{x}}^{(i)} = g_\theta(\mathbf{z}^{(i)})\}$
- Calculer $\mathcal{L}_{VAE}(\phi,\theta) = \sum_{i=1}^{m}\left[\log\tilde{\mathbf{x}}^{(i)} - \frac{1}{2}\left(\mathrm{Tr}\Sigma^{(i)} + \mu^{(i),T}\mu^{(i)} - \log\det\Sigma^{(i)}\right)\right]$
- $\Phi \leftarrow \nabla_\phi \mathcal{L}_{VAE}(\phi,\theta)$
- $\Theta \leftarrow \nabla_\theta \mathcal{L}_{VAE}(\phi,\theta)$
- $\phi \leftarrow \phi - \eta\,\mathrm{ADAM}_{\beta_1,\beta_2}(\phi,\Phi)$
- $\theta \leftarrow \theta - \eta\,\mathrm{ADAM}_{\beta_1,\beta_2}(\theta,\Theta)$

end

Algorithme 4 : WAE.

Require : Hyper-paramètres : Taille des mini-lots $m = 32$, dimension de l'espace latent z, pas de gradient $\eta_q, \eta_g, \eta_d = 0.001$, premier facteur d'inertie $\beta_=0.5$, second facteur d'inertie $\beta_2 = 0.999$, facteur de régularisation $\lambda = 1$, échéancier des paramètres $w = 10^{-5}$.

Require : ϕ_0, les paramètres initiaux de l'encodeur q_ϕ. θ_0, les paramètres initiaux du décodeur g_θ. γ_0 les paramètres initiaux du discriminateur d_γ.

for *k itérations* **do**

- Tirer un mini-lot de m exemples $\{\mathbf{x}^{(i)}\}_{i=1}^{i=m} \sim \mathbb{D}$
- Échantillonner de la distribution antérieure $\{\tilde{\mathbf{z}}^{(i)} \sim \mathcal{N}(0,1)\}_{i=1}^{i=m}$
- Propager à travers l'encodeur $\{\mu^{(i)}, \Sigma^{(i)} = q_\phi(\mathbf{x}^{(i)})\}_{i=1}^{i=m}$
- Tirer un mini-lot $\{\varepsilon^{(i)} \sim \mathcal{N}(0,1)\}_{i=1}^{i=m}$
- Échantillonner les variables latentes $\{\mathbf{z}^{(i)} = \mu^{(i)} + \varepsilon^{(i)}\Sigma^{(i)}\}_{i=1}^{i=m}$
- $\Phi \leftarrow \nabla_\phi \sum_{i=1}^{m} \nabla_\phi [c(\mathbf{x}, g_\theta(\mathbf{z})) - \lambda \log d_\gamma(\mathbf{z})]$
- $\Theta \leftarrow \nabla_\theta \sum_{i=1}^{m} \nabla_\theta [c(\mathbf{x}, g_\theta(\mathbf{z})) - \lambda \log d_\gamma(\mathbf{z})]$
- $\Gamma \leftarrow \nabla_\gamma \frac{\lambda}{m} \sum_{i=1}^{m} \nabla_\gamma [\log d_\gamma(\tilde{\mathbf{z}}) + \log(1 - d_\gamma(\mathbf{z})]$
- $\phi \leftarrow \phi - \mathrm{ADAM}_{\beta_1,\beta_2}(\phi, \Phi)$
- $\theta \leftarrow \theta - \eta \mathrm{ADAM}_{\beta_1,\beta_2}(\theta, \Theta)$
- $\gamma \leftarrow \gamma + \eta \mathrm{ADAM}_{\beta_1,\beta_2}(\gamma, \Gamma)$

end