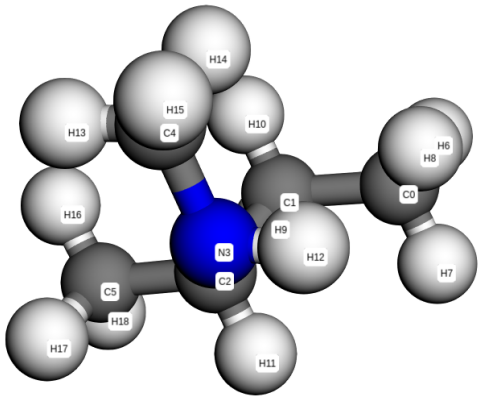
Apprentissage sous contraintes physiques

Molecule Energy Prediction



Hanna Bekkare - Maxime Moshfeghi
Team *The Cats*

Toulouse INP - ENSEEIHT
France

27 juin 2025

Table des matières

Introduction

Dans le domaine de la chimie quantique computationnelle, la prédiction de l'énergie d'atomisation des molécules représente un défi majeur, en particulier en raison des contraintes physiques complexes que ces systèmes imposent. Ce projet s'inscrit dans cette problématique, en cherchant à modéliser la surface d'énergie potentielle de petites molécules organiques à partir de leur structure géométrique tridimensionnelle.

L'objectif est de prédire l'énergie $E(r)$ d'une configuration moléculaire r, tout en respectant les invariances fondamentales liées aux translations, rotations et permutations des atomes. Cette tâche de régression non linéaire en haute dimension nécessite la construction de représentations géométriques adaptées, capables de capturer la complexité des interactions atomiques tout en intégrant ces contraintes physiques.

Pour y parvenir, nous avons principalement utilisé l'approche scattering 3D, fondée sur les ondelettes harmoniques solides. Cette méthode permet d'extraire des descripteurs invariants aux transformations géométriques, qui servent ensuite de base à une régression multilinéaire pour estimer l'énergie moléculaire. Nous avons également envisagé d'autres méthodes d'apprentissage reposant sur des features telles que les matrices de Coulomb afin de comparer leur performance dans ce contexte.

Chapitre 1

Présentation des données

1.1 Structures des données

Comme dit dans l'introduction, l'objectif est de prédire l'énergie $E(\mathbf{r})$, où $\mathbf{r} = (r_1, \ldots, r_N)$ avec $\forall i \in [\![1, N]\!], r_i \in \mathbb{R}^3$ correspond aux coordonnées de chaque atome de la molécule. Bien sûr, on dispose aussi pour chacun des atomes la nature de cet atome (d'où l'on peut décrire, entre autres, son numéro atomique et des propriétés qui lui sont donc intrinsèques). En résumé, le jeu d'entraînement comprends les données suivantes :

— coordonnées (x, y, z) de chaque atome de la molécule
— la nature de l'atome à chacune de ces coordonnées
— l'énergie de la molécule étudiée

Les données sur les structures des molécules sont stockées dans des fichiers `.xyz`.

Exemple d'un fichier .xyz

```
16
Properties=species:S:1:pos:R:3 pbc="F F F"
C       -0.66765700      -1.71208400       1.74041400
N       -0.98793200       0.20794300       0.20618800
...     ...               ...              ...
H        2.49897500       2.35500300      -1.14965300
H        2.23706700       1.26773100      -2.49871400
```

Chaque fichier `id.xyz` du jeu d'entraînement à une correspondance avec une énergie. Ces données sont disponibles dans un fichier `.csv` à part :

Fichier.csv avec les énergies à prédire

```
id,energy
1,-90.10787994300517
2,-69.92764655700375
...
```

Pour le test, les données dont l'on dispose sont les mêmes, à l'exception, bien sûr, de l'énergie qui est la donnée que l'on cherche à prédire.

Dans l'idée, le modèle que l'on souhaite faire est résumé par le schéma ci-dessous :

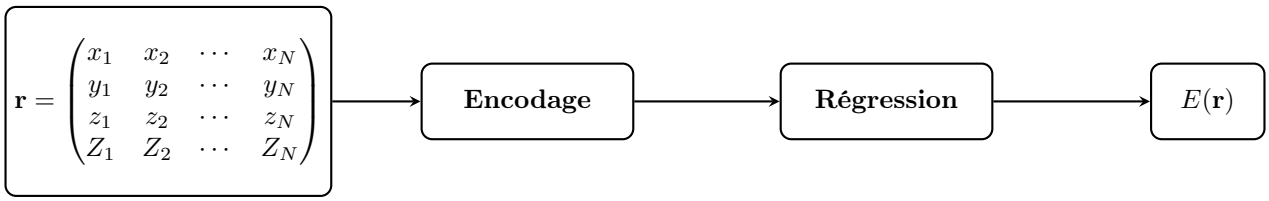


FIGURE 1.1 – Schéma de la prédiction de l'énergie d'une molécule

Une molécule peut être aussi représentée en 3D, grâce à la bibliothèque `py3Dmol` (d'autres fonctions existent également dans d'autres packages, mais il s'agit de la représentation visuelle la plus intéressante, entre autres car la vue 3D est en rotation à l'exécution).

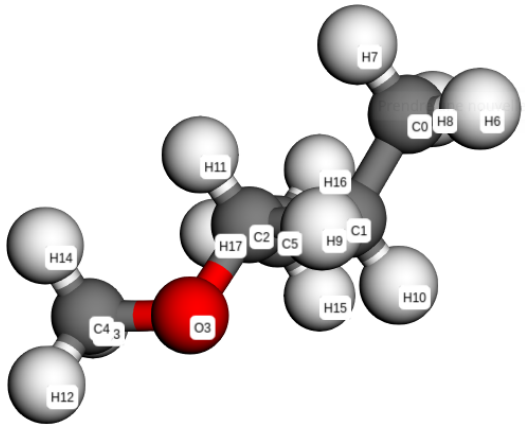


FIGURE 1.2 – Capture d'écran de la vue `py3Dmol` d'une molécule

1.2 Premières statistiques descriptives

Une première statistique brute intéressante à exploiter est la population d'atomes présentes au sein des molécules de notre dataset.

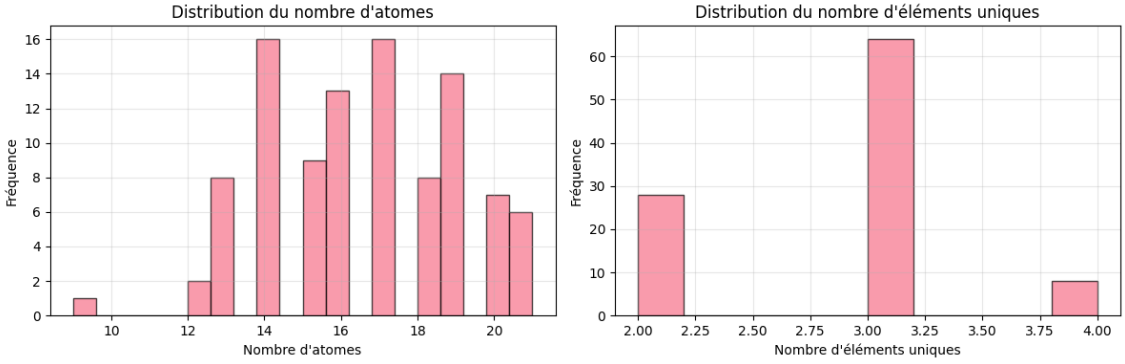


FIGURE 1.3 – Distribution des atomes au sein des molécules

De ces graphes, on constate que la majorité des molécules contient 3 atomes différents.

Au passage, on peut d'ailleurs étudier la fréquence des 4 atomes :

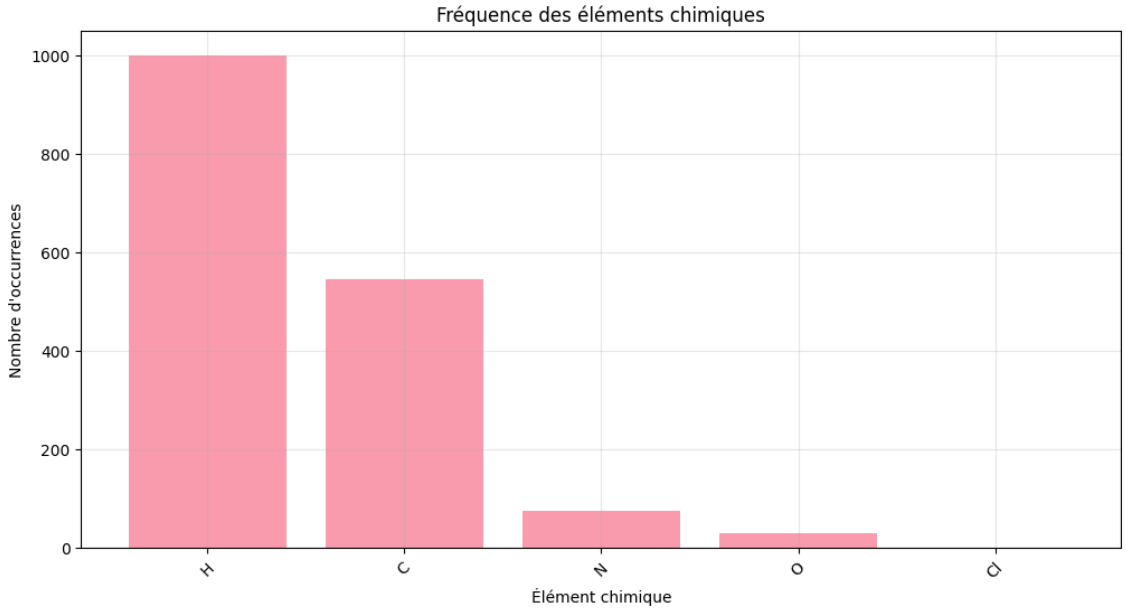


FIGURE 1.4 – Fréquences de présence des atomes au sein des molécules

Bien sûr, l'hydrogène apparaît comme étant l'atome le plus fréquent.
On peut aussi faire une analyse statistique rapide sur la distribution des énergies des molécules :

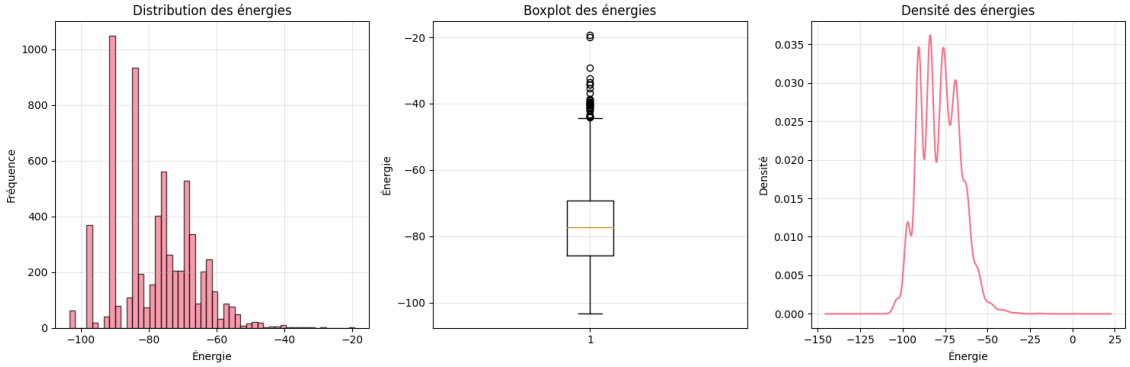


FIGURE 1.5 – Distribution des énergies des molécules

On constate que 95% des énergies se retrouvent entre -45 et -102.

Chapitre 2

Régression linéaire sur des descripteurs simples

2.1 Feature : masse atomique moyenne

Le premier descripteur utilisé est la moyenne des masses atomiques d'une molécule. Ce choix correspond à l'approche la plus simple pour prédire l'énergie, en n'utilisant qu'une seule information : la masse totale. Ce modèle est rapide à mettre en place, facile à entraîner, et très lisible : on suppose qu'une molécule plus lourde a besoin de plus d'énergie pour être formée. Il sert ainsi de *baseline* — une référence minimale pour évaluer les méthodes plus avancées.

2.1.1 Implémentation

Dans cette section et pour le reste du projet, on utilise la bibliothèque `ase` qui nous permet de lire les fichiers au format `.xyz`. Afin de créer la base d'entrainement, on implémente une fonction `xyz_to_mass` qui nous permet de calculer les masses des molécules. La fonction `csv_mass` nous permet de créer un fichier csv contenant les attributs `id`, `energy` et `mass` qui peuvent être utilisés pour l'entraînement. Comme le problème de prédiction est simple, on utilise le modèle `LinearRegression()` de la librairie `sklearn.linear_model`.

2.1.2 Analyse des résultats

On observe sur la Table 2.1 que les résultats obtenus avec cette méthode simple de prédiction ne sont pas satisfaisants.

En effet, la simplicité du modèle repose sur une hypothèse physico-chimique naïve : l'idée que plus une molécule contient de matière, plus son énergie d'atomisation est élevée. Si cette corrélation peut sembler intuitive (davantage d'atomes impliquant plus de liaisons à rompre), elle néglige de nombreux aspects fondamentaux de la chimie moléculaire.

Les limites du modèle sont immédiates. Il ignore totalement la nature des atomes impliqués, néglige la géométrie et la structure moléculaire, et ne tient pas compte des types de liaisons chimiques. De plus, l'hypothèse d'une relation linéaire entre masse et énergie est peu réaliste : deux molécules de même masse peuvent avoir des énergies de formation radicalement différentes. En somme, ce modèle constitue une approche volontairement simplifiée, utile pour fixer un point de départ, mais rapidement insuffisante pour une modélisation fidèle du phénomène étudié.

Méthode des moyennes	**RMSE**
Entraînement	7.059
Test	7.003

TABLE 2.1 – Précision obtenue pour la méthode des moyennes

2.2 Feature : angles de liaison

Dans cette seconde approche, nous explorons l'utilisation des angles de liaison comme descripteurs géométriques pour prédire l'énergie des molécules. Contrairement à la méthode précédente fondée uniquement sur la masse atomique moyenne, cette stratégie cherche à capturer davantage d'informations structurelles, tout en restant simple et interprétable.

2.3 Implémentation

À l'aide de la bibliothèque `ase`, nous extrayons les angles locaux formés entre triplets d'atomes liés, en tenant compte uniquement des voisins situés dans un rayon de $3,5$ autour de chaque atome central. Cette contrainte permet de ne considérer que les interactions chimiques pertinentes.

Pour chaque molécule, quatre statistiques descriptives sont calculées à partir des angles obtenus : la moyenne, l'écart-type, le minimum et le maximum. Ces valeurs condensent l'essentiel de la géométrie locale dans un vecteur de faible dimension (4 features par molécule), ce qui permet un apprentissage efficace et rapide. Ces données sont fusionnées avec les énergies d'atomisation, puis utilisées pour entraîner un modèle `RandomForestRegressor` de la bibliothèque `sklearn.ensemble`, choisi pour sa robustesse aux non-linéarités et sa faible sensibilité aux outliers.

2.3.1 Analyse des résultats

Les résultats obtenus à l'aide des descripteurs angulaires sont présentés dans la Table 3.6. Bien que les performances sur l'ensemble d'entraînement soient très bonnes (RMSE = 1.69), les erreurs sur l'ensemble de test sont nettement plus élevées (RMSE = 9.95). Cette différence importante indique un phénomène de surapprentissage : le modèle s'adapte trop étroitement aux données d'entraînement, au détriment de sa capacité à généraliser sur des exemples nouveaux.

Ce comportement suggère que les descripteurs fondés uniquement sur les angles de liaison, bien qu'intuitifs et physiquement motivés, ne suffisent pas à capturer toute la complexité du problème. Le modèle parvient à exploiter efficacement les variations internes de la base d'entraînement, mais il échoue à généraliser ces relations à d'autres molécules dont la structure ou la composition diffère.

Malgré tout, cette méthode présente plusieurs avantages : elle repose sur une notion chimique simple (les angles de liaison), réduit considérablement la dimensionnalité (4 valeurs par molécule), et capte partiellement la géométrie locale. Toutefois, elle ne tient pas compte de la nature chimique des atomes, ignore les interactions à longue portée ou les effets de torsion, et dépend fortement du choix du rayon de coupure 3,5 , qui pourrait ne pas convenir à toutes les topologies moléculaires.

En somme, cette approche illustre bien l'intérêt et les limites du *feature engineering* basé uniquement sur la géométrie. Elle met en lumière la nécessité d'introduire des représentations plus riches et physico-chimiquement complètes pour améliorer la robustesse et la capacité de généralisation du modèle.

Méthode des angles de liaison	**RMSE**
Entraînement	1.6908
Test	9.952

TABLE 2.2 – Précision obtenue pour la méthode fondée sur les angles de liaison

Chapitre 3

Régression linéaire sur des descripteurs complexes

3.1 Feature : matrices de Coulomb

La matrice de coulomb est un descripteur moléculaire "complexe". En effet, il s'agit d'une matrice de la taille $nb_{atomes} \times nb_{atomes}$, et définit de la manière suivante :

$$M_{ij} = \begin{cases} 0.5 Z_i^{2.4} & \text{si } i = j \\ \dfrac{Z_i Z_j}{R_{ij}} & \text{si } i \neq j \end{cases}$$

où Z_k correspond au numéro atomique de l'atome k et R_{kl} correspond à la distance entre l'atome k l'atome l.

Ce descripteur est invariant par translation et par rotation. En effet, les seules données exploitées ici sont le numéro atomique, intrinsèque à chaque atome et qui ne dépend aucunement de la position de l'atome dans l'espace, et les distances inter-atomes, qui font disparaître les valeurs des angles au moment du passage à la norme.

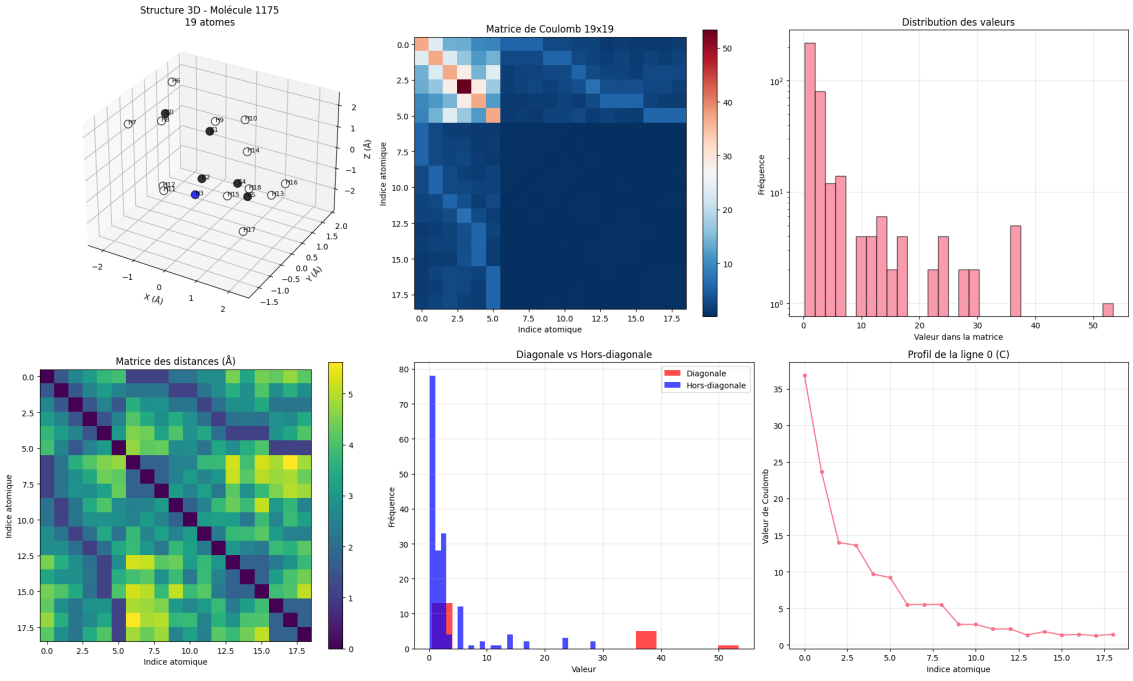


FIGURE 3.1 – Analyse d'une matrice de Coulomb pour la molécule C5H13N

Dans la figure ci-dessus, on a une visualisation en 3D de la molécule étudié. Juste à sa droite, on trouve une

visualisation de la matrice de Coulomb. Le bloc 6×6 en haut à gauche de la matrice présentant des valeurs fortes correspond aux atomes de carbone et à l'atome d'azote, qui sont fortement liés aux autres atomes. Le reste de la matrice correspond aux interactions des atomes d'hydrogènes avec les autres atomes, et on remarque bien que les atomes d'hydrogène sont très peu, pour ne pas dire pas du tout, liés aux autres atomes d'hydrogènes.
Remarque : On obtient de l'invariance par permutation en imposant par exemple une méthode de classement des colonnes par leur norme 2 au sein de la matrice, ou bien en récupérant uniquement les valeurs propres de la matrice et en les ordonnant dans l'ordre croissant ou décroissant.

3.1.1 Implémentation de l'apprentissage

Dans un premier temps, on calcule les matrices de Coulomb des molécules à l'aide de la fonction `CoulombMatrix` de la librairie `dscribe`. Ceci constitue le premier descripteur sur lequel on entraîne notre régresseur pour prédire l'énergie moléculaire. On utilise différents régresseurs afin de les comparer et d'essayer d'obtenir le score de RMSE le plus bas sur la platforme Kaggle.

On décide de jouer sur le paramètre **permutation** de la fonction `CoulombMatrix` parmis les trois options suivantes : **"sorted_l2"**, **"eigenspectrum"** et, **"random"** avec le paramètre **"sigma"** $= 0.1$.

- **sorted_l2** : trie les lignes/colonnes de la matrice en fonction de la norme L^2 de chaque ligne, ce qui permet une représentation canonique tout en conservant une grande partie de l'information chimique.
- **eigenspectrum** : remplace la matrice complète par son spectre (vecteur contenant uniquement les valeurs propres), ce qui garantit une invariance parfaite à la permutation des atomes, mais au prix d'une perte d'information structurale.
- **random** : applique une permutation aléatoire des lignes et colonnes à chaque appel. Cette méthode nécessite généralement une moyenne sur plusieurs réalisations ou l'introduction de bruit (paramètre σ) pour éviter un sur-apprentissage à des structures particulières.

Ensuite, pour réduire encore plus la RMSE de test, on tente d'ajouter deux descripteurs supplémentaires : `SOAP` et `ACSF`.

- **SOAP** (Smooth Overlap of Atomic Positions) encode l'environnement local de chaque atome à l'aide de fonctions de base sphériques et radiales. Il est particulièrement adapté pour capturer les interactions inter-atomiques dans des systèmes chimiques, tout en étant invariant par rotation, translation et permutation des atomes.
- **ACSF** (Atom-Centered Symmetry Functions) est un descripteur inspiré de la physique, construit à partir de fonctions radiales et angulaires centrées sur chaque atome. Il permet de modéliser les environnements atomiques locaux avec un certain degré de finesse tout en respectant les symétries fondamentales.

3.1.2 Analyse des résultats

La Table 3.7 présente les résultats obtenus lors de l'entraînement des modèles précedemment décrits.

Regressor	Permutation	RMSE Train	RMSE Test
LinearRegressor	sorted_l2	1.392	1.781
LinearRegressor	eigenspectrum	2.613	2.614
LinearRegressor	random	1.425	1.490
XGBRegressor	sorted_l2	0.148	0.467
XGBRegressor	eigenspectrum	0.320	1.033
XGBRegressor	random	0.152	0.484
XGBRegressor + SOAP	random	0.115	0.482
XGBRegressor + SOAP	**sorted_l2**	**0.110**	**0.466**
XGBRegressor + SOAP + ACSF	sorted_l2	0.108	0.472

TABLE 3.1 – Comparaison des performances en RMSE des différents modèles testés

On constate que l'ajout de modèles non linéaires comme `XGBRegressor` améliore significativement les performances par rapport à la régression linéaire. Par exemple, le RMSE passe de **1.781** à **0.467** en test avec le même descripteur (CoulombMatrix avec `sorted_l2`).

Concernant le paramètre de permutation dans `CoulombMatrix`, on remarque que :

— L'option `sorted_l2` donne généralement les meilleurs résultats, probablement car elle fournit une représentation plus stable et cohérente des molécules.

— L'option `eigenspectrum`, bien qu'invariante par permutation, semble moins expressive et donne de moins bonnes performances.

— `random` donne des résultats intermédiaires, mais reste compétitif avec des régresseurs non linéaires.

L'ajout de descripteurs supplémentaires comme `SOAP` permet une amélioration marginale mais réelle. Par exemple, avec `XGBRegressor` et la permutation `sorted_l2`, le RMSE test passe de **0.467** à **0.466** avec SOAP, et à **0.472** avec SOAP + ACSF. Bien que l'ajout de `ACSF` ne réduise pas le RMSE test de manière significative, il améliore légèrement le RMSE d'entraînement, ce qui indique que le modèle capte un peu plus de complexité.

Ces résultats suggèrent que l'utilisation combinée de plusieurs descripteurs peut potentiellement aider, mais que le choix du modèle et des paramètres associés joue un rôle tout aussi crucial.

3.2 Feature : Harmonic Scattering 3D

3.2.1 Théorie

Cette section reprend en partie les travaux fait dans la ressource [1]. L'idée du Scattering Harmonique 3D est de modéliser une molécule par les densités électroniques présentes en son sein (par exemple liées aux électrons de valence, aux liaisons, etc,...). Ces densités sont agrégées au sein d'une même fonction ρ, définie pour une molécule x de la manière suivante :

$$\rho_x(u) = \sum_k \gamma_k g(u - r_k)$$

où γ_k correspond aux nombres d'électrons de l'atome k, r_k la position de ce même atome dans l'espace, et g est une gaussienne (qui pour chaque k est donc centrée sur l'atome).

Cette valeur est invariante par permutation (par commutativité de la somme).

On définit ensuite la valeur suivante, invariante par rotation :

$$U[j,l]\rho(u) = \left(\sum_{-l}^{l} |\rho * \psi_{j,l}^m(u)|^2 \right)^{1/2}$$

Il s'agit d'une norme dans la base des ondelelettes $(\psi_{j,l}^m)$, qui sont définit de la manière suivante :

$$\psi_\ell^m(u) = \frac{1}{\left(\sqrt{2\pi}\right)^3} e^{-|u|^2/2} |u|^\ell Y_\ell^m \left(\frac{u}{|u|} \right)$$

L'invariant est ensuite intégré sur tous l'espace, on obtient ainsi un invariant par rotation et par translation :

$$S\rho[j,l,q] = \int_{\mathbb{R}^3} |U[j,l]\rho(u)|^q du$$

Ces coefficients correspondent à ceux du scattering d'un molécule x.

3.2.2 Application

Supposition pour le dénombrement des coordonnées à l'ordre 0 :

Dans le code, on déclare les variables suivantes :

Data processing and vizualisation.ipynb

```
...
J = 1
L = 2
...
M, N, O = 32, 32, 32
...
```

Cela signifie que j prendra ses valeurs dans $[\![0,1]\!]$ et l prendra ses valeurs dans $[\![0,2]\!]$. D'autre part, dans l'article sur le scattering ([1]), q prend sa valeur dans $[\![1,2]\!]$. On peut donc dénombrer, dans notre cas, le nombre de coefficients κ que l'on va obtenir pour le scattering $(S\rho[j,l,q])_{j\in[\![0,1]\!],l\in[\![0,2]\!],q\in[\![1,2]\!]}$:

$$\kappa = \underbrace{|[\![0,1]\!]|}_{2} \times \underbrace{|[\![0,2]\!]|}_{3} \times \underbrace{|[\![1,2]\!]|}_{2} = 12$$

On s'attend donc à ce que le scattering d'une molécule soit un vecteur de taille 12. On peut le vérifier en affichant le scattering de quelques molécules parmi notre jeu d'entraînement :

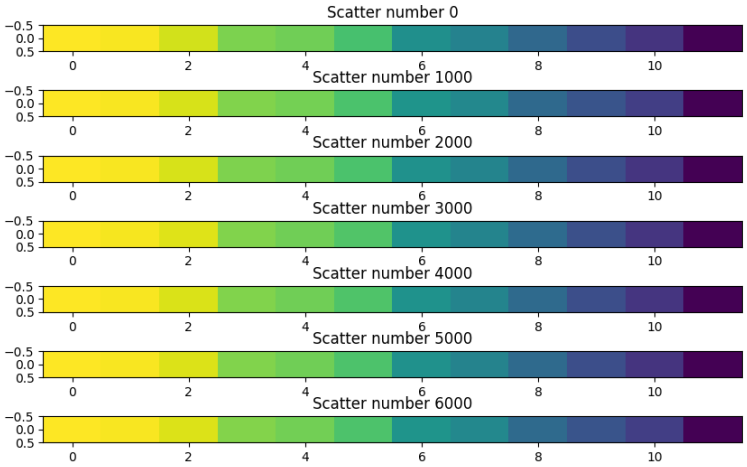


FIGURE 3.2 – Analyse du scattering à l'ordre 0 de 7 molécules

On voit effectivement que le scattering a une taille de 12, mais cependant les vecteurs semblent assez proches. Pour visualiser le scattering, on préférera donc centrer ces valeurs :

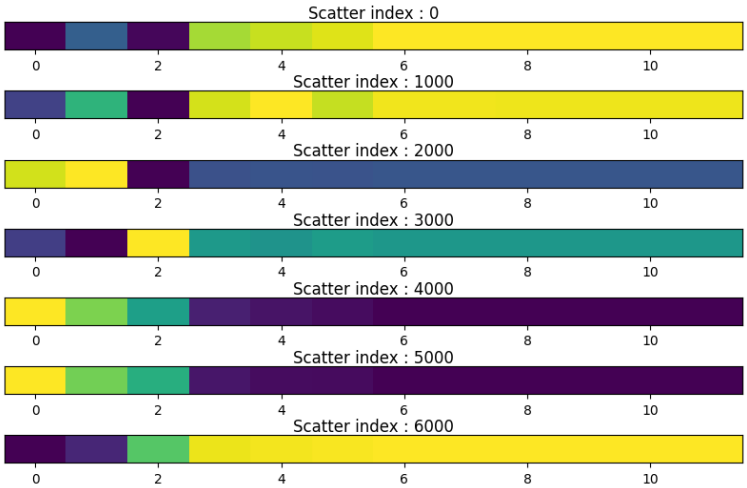


FIGURE 3.3 – Analyse du scattering à l'ordre 0 centré de 7 molécules

Remarque : dans notre cas, le centrage est en fait fait par un Scaler, en amont de l'entraînement pour le cas de base implémenté dans le notebook donné comme exemple (avec la régression Ridge). Nous avons aussi implémenté une partie avec XGBoost, et on centre cette fois-ci les données manuellement.

3.2.3 Analyse des résultats

Ridge	**RMSE**
Entraînement	6.735
Test	6.997

TABLE 3.2 – Précision obtenue sur la méthode de régression ridge avec `alpha` = 0.01

XGBoost	**RMSE**
Entraînement	4.705
Test	7.488

TABLE 3.3 – Précision obtenue sur la méthode XGBoost appliquée aux scatterings, pour `learning_rate` = 0.1, `max_depth` = 6

XGBoost	**RMSE**
Entraînement	0.0146
Test	8.366

TABLE 3.4 – Précision obtenue sur la méthode XGBoost appliquée aux scatterings, pour `learning_rate` = 0.1, `max_depth` = 20

Cas d'overfitting

XGBoost	**RMSE**
Entraînement	0.525
Test	11.841

TABLE 3.5 – Précision obtenue sur la méthode XGBoost appliquée aux scatterings, pour `learning_rate` = 1.2, `max_depth` = 6

Ces résultats n'étant pas exceptionnels, nous avons décidé d'augmenter les paramètres du scattering :

Data processing and vizualisation.ipynb

```
...
J = 3
L = 3
...
M, N, O = 160, 112, 80
...
```

Cette fois-ci, on obtient de bien meilleurs résultats. On constate que les coefficients d'ordre 1 et 2 sont bien plus nombreux (on passe de 108 à 480). On obtient dans le meilleur des cas :

Ridge	**RMSE**
Entraînement	0.108
Test	0.108

TABLE 3.6 – Précision obtenue sur la méthode de régression ridge avec `alpha` = 0.01

Conclusion

Au terme de ce projet, nous avons exploré différentes approches pour prédire l'énergie d'atomisation de petites molécules organiques en respectant les contraintes physiques inhérentes à leur représentation. L'objectif était de développer des modèles de régression capables d'estimer cette énergie à partir de la structure tridimensionnelle des molécules, tout en intégrant des invariances essentielles (rotation, translation, permutation).

Nous avons d'abord établi des bases simples, à travers l'utilisation de descripteurs élémentaires comme la masse atomique moyenne ou les angles de liaison. Ces méthodes, bien que rapides à implémenter et interprétables, se sont révélées insuffisantes pour capter la complexité des interactions atomiques, comme en témoignent leurs performances limitées, particulièrement en phase de test.

L'introduction de descripteurs plus sophistiqués, tels que les matrices de Coulomb, les signatures SOAP et ACSF, ainsi que le Scattering Harmonique 3D, a permis d'améliorer significativement la précision des prédictions. Les modèles non linéaires, notamment XGBoost, ont montré un fort potentiel en exploitant efficacement ces représentations complexes. Cependant, les risques de surapprentissage demeurent, en particulier lorsqu'on pousse les paramètres du modèle à l'extrême.

Le bilan des meilleures approches est résumé dans le tableau suivant :

Regressor	**Méthode d'encodage**	**RMSE Train**	**RMSE Test**
XGBRegressor + SOAP	Coulomb matrix with `sorted_l2`	0.110	0.466
Ridge, `alpha` = 0.001	**Scattering, J = L = 3, (M, N, O)=(160, 112, 80)**	**0.108**	**0.108**

TABLE 3.7 – Comparaison des performances en RMSE des différents modèles testés

Ces résultats mettent en lumière l'importance cruciale du choix des descripteurs et de la régularisation dans les tâches d'apprentissage supervisé en chimie.

On notera aussi que les modèles qui fonctionnent le mieux sont ceux contraints par la physique.

Bibliographie

[1] Matthew Hirn Stéphane Mallat Michael Eickenberg, Georgios Exarchakis and Louis Thiry. Solid harmonic wavelet scattering for predictions of molecule properties, 2018.