Apprentissage sous contraintes physiques - Prédiction d'énergies moléculaire

Tristan Gay et Clément Gris - 5 ModIA

Juin 2025

1 Introduction

Ce projet s'inscrit dans le domaine de l'apprentissage automatique sous contraintes physiques, où l'objectif principal est de prédire l'énergie d'atomisation $E(\mathbf{r})$ d'une molécule en fonction de ses caractéristiques et des propriétés physiques qu'elle possède.

L'un des défis majeurs de ce projet réside dans le respect des contraintes de symétrie. En effet, l'énergie d'atomisation doit rester invariante sous des transformations de translation, de rotation et de permutation des atomes. Cela signifie que pour toute translation b, rotation U et permutation σ, l'énergie prédite doit satisfaire $E(T_b(\mathbf{r})) = E(T_U(\mathbf{r})) = E(T_\sigma(\mathbf{r})) = E(\mathbf{r})$.

Pour aborder ce problème, nous avons utilisé un sous-ensemble de la base de données QM7-X, qui contient 4739 structures de molécules avec un nombre variable d'atomes. Les données sont divisées en ensembles d'entraînement et de test, et sont fournies dans des formats spécifiques que nous avons appris à manipuler à l'aide de bibliothèques Python telles que ASE.

Ce rapport présente les différentes étapes de notre projet, depuis le préprocessing des données jusqu'à la mise en œuvre et l'évaluation de modèles d'apprentissage automatique. Nous détaillerons notamment les méthodes utilisées pour respecter les contraintes de symétrie, ainsi que les résultats obtenus et leur analyse. Nous conclurons par une discussion sur les perspectives d'amélioration et les travaux futurs possibles dans ce domaine. Tous les codes utilisés pour ce projet sont disponibles sur ce GitHub.

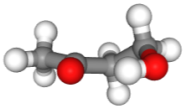
2 Data preprocessing

2.1 Analyse descriptives des molécules

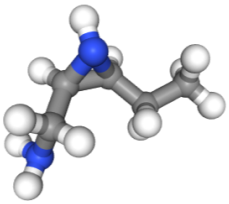
Dans un premier temps, nous procédons à l'importation des fichiers contenant les données moléculaires, ceux-ci étant au format `xyz`. Ce format est largement utilisé pour représenter des structures moléculaires en trois dimensions, car il permet de spécifier les coordonnées spatiales de chaque atome au sein de la molécule.

Afin de nous assurer de la bonne intégrité des données et de visualiser la structure des molécules, nous affichons quelques-unes de ces molécules en trois dimensions. Cette étape préliminaire est importante pour identifier d'éventuelles anomalies ou particularités structurales qui pourraient influencer les analyses ultérieures.

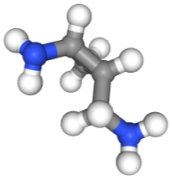
Pour illustrer cette étape, nous avons sélectionné aléatoirement plusieurs molécules issues du jeu de données. La figure 1 présente quelques exemples de ces structures moléculaires visualisées en 3D. Cette visualisation nous permet non seulement de confirmer la validité des données importées, mais également de nous familiariser avec les configurations spatiales des molécules étudiées.



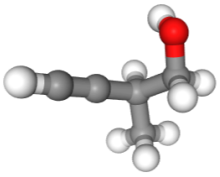
(a) Molécule 219



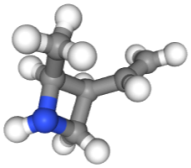
(b) Molécule 342



(c) Molécule 395



(d) Molécule 730



(e) Molécule 890

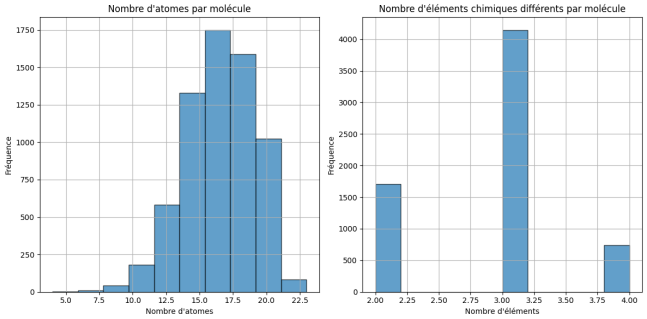
Figure 1: Cinq molécules aléatoires du jeu de données

Cette première visualisation nous permet d'observer la diversité présente dans notre jeu de données. Les molécules sont composées de différents types d'atomes, chacun représenté par une couleur spécifique : l'hydrogène (H) en blanc, le carbone (C) en gris, l'oxygène (O) en rouge, et l'azote (N) en bleu. Cette coloration facilite l'identification des atomes et permet une analyse visuelle rapide des structures moléculaires.

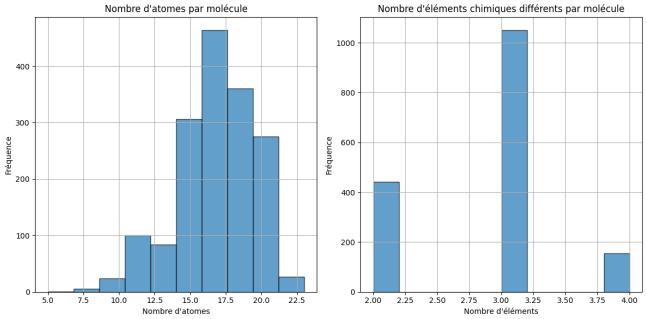
Ici, nous observons que les molécules de notre échantillon présentent une grande variété en termes de forme, de composition et de nombre d'atomes. En rejouant plusieurs fois le code de visualisation, et donc en observant différentes molécules, nous remarquons que cette diversité peut être généralisée à l'ensemble du jeu de données. Cela confirme la pertinence de disposer d'un jeu de données large et bien représentatif dans le cadre de notre étude.

La diversité structurelle et compositionnelle des molécules est un aspect nécessaire pour la robustesse de nos analyses ultérieures. En effet, un jeu de données varié permet de capturer un large éventail de propriétés moléculaires, ce qui est essentiel pour développer des modèles prédictifs fiables et généralisables.

Nous poursuivons ici nos analyses descriptives en affichant des statistiques sur nos ensembles de données d'entraînement et de test. La figure 2 présente les distributions du nombre d'atomes par molécule ainsi que le nombre d'éléments chimiques distincts présents dans chaque molécule.



(a) Statistiques de l'ensemble d'entraînement.



(b) Statistiques de l'ensemble de test.

Figure 2: Comparaison des statistiques des ensembles de données d'entraînement et de test.

Nous pouvons observer que le nombre d'atomes par molécule varie entre 5 et 23, et que le nombre d'éléments chimiques par molécule peut être de 2, 3 ou 4. La distribution du nombre d'atomes forme une courbe gaussienne dont la moyenne se situe autour de 16. Il est important de noter que les ensembles de données d'entraînement et de test présentent des statistiques similaires.

Cette similarité nous permet de conclure que notre ensemble de test est représentatif de notre ensemble d'entraînement, ce qui est essentiel pour la validité de nos analyses et la généralisation de nos modèles. Ainsi, nous pouvons être confiants dans l'utilisation de ces ensembles de données pour les étapes ultérieures de notre étude.

2.2 Analyse des energies

Comme expliqué en introduction, chaque molécule est associée à une énergie caractéristique, exprimée en électronvolts (eV). Les données utilisées sont stockées dans un fichier `.csv` contenant deux colonnes : la première correspond à l'identifiant unique de chaque molécule (de la forme `id_x`, où `x` est un entier), et la seconde indique l'énergie correspondante.

Avant de mettre en place des modèles de prédiction, il est pertinent d'observer la distribution des énergies afin de mieux comprendre la nature des données. Les figures ci-dessous présentent deux visualisations : un histogramme de la distribution des énergies, et un boxplot mettant en évidence les éventuels outliers.

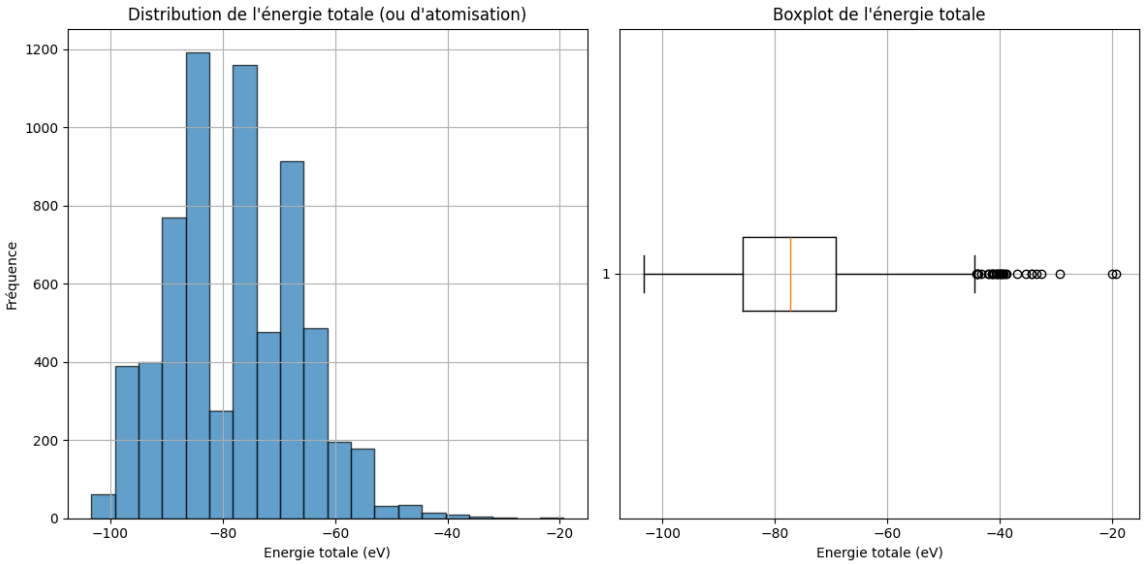


Figure 3: Distribution des énergies moléculaires : histogramme (haut) et boxplot (bas)

L'histogramme montre que les énergies sont comprises entre -110 eV et -20 eV. On observe que la majorité des molécules possèdent une énergie centrée autour de -80 eV, ce qui correspond à la valeur moyenne approximative de la distribution.

Le boxplot met en évidence une forte concentration des énergies entre -90 eV et -70 eV, ce qui représente environ 90 % des données. Les valeurs supérieures à -40 eV sont identifiées comme des outliers, suggérant des cas moléculaires particuliers.

Ces premières observations sont importantes car elles permettront de contextualiser les performances des modèles de prédiction. En effet, les erreurs de prédiction seront analysées non seulement en termes absolus, mais également relativement à l'échelle naturelle des énergies. Cela donnera un sens physique aux écarts obtenus et permettra de juger la pertinence des modèles développés.

3 Premiers modèles de prédiction

3.1 Régression linéaire

Une première approche consiste à mettre en place des modèles simples et interprétables afin de faire des prédictions. Dans un premier temps, nous avons donc mis en place une régression linéaire basée uniquement sur le nombre d'atomes. Les résultats sont présentés dans le tableau 1.

Métrique	Valeur
Coefficient (pente)	-4.0082
Intercept	-10.9544
Score R^2	0.9175
RMSE	3.3466

Table 1: Résultats de la régression linéaire basée sur le nombre d'atomes.

Les résultats sont intéressants comme on peut le voir sur la figure 4. Le score R^2 est élevé (0.91), montrant que la régression est performante. De plus, si l'on regarde la RMSE, qui est notre métrique de référence dans ce projet, elle est de 3.34. Cette valeur, au regard de l'étude descriptive des énergies effectuée dans la partie précédente, est relativement faible lorsque l'on sait que la grande majorité des énergies varient entre -90 et -70, donc une amplitude de 20 eV. Toutefois, il est clair que ce résultat peut être amélioré avec d'autres méthodes.

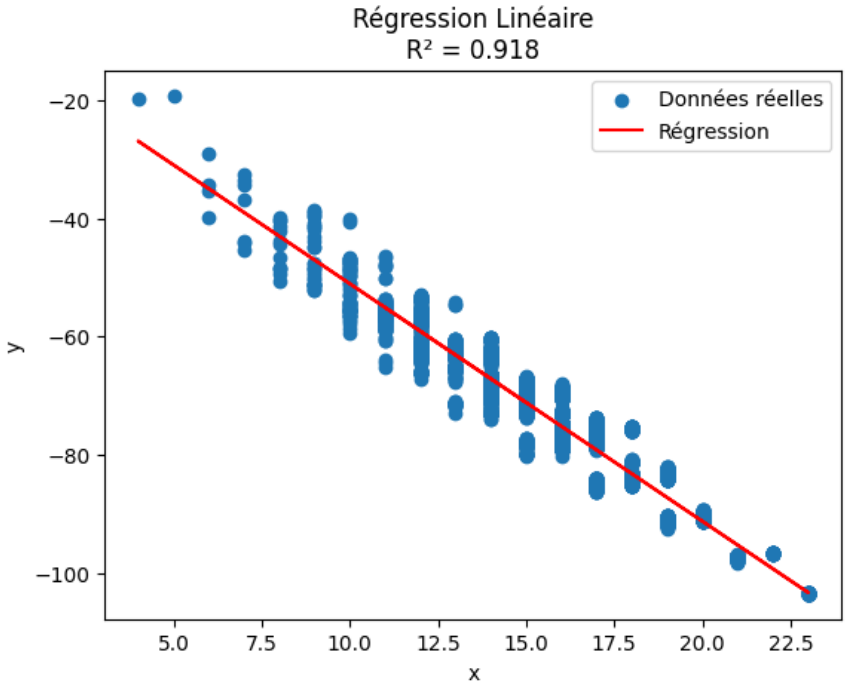


Figure 4: Régression linéaire basée sur le nombre d'atomes.

3.2 Mise en place d'un modèle Random Forest

Avant de nous pencher sur l'intégration de la physique des molécules dans nos prédictions, nous avons mis en place un modèle légèrement plus complexe qu'une régression linéaire : un Random Forest.

Pour exploiter ces fichiers .xyz, nous avons mis en place une panalyse utilisant la bibliothèque `ASE` (Atomic Simulation Environment). Chaque fichier est lu à l'aide de la fonction `read()`, qui retourne un objet `Atoms`. Cet objet contient toutes les informations dont nous avons besoin pour extraire des descripteurs géométriques importants. Nous pouvons depuis ces informations calculer d'autres caractéristiques.

Les caractéristiques calculées pour chaque molécule sont les suivantes :

- **Nombre d'atomes** (`num_atoms`) : taille de la molécule.
- **Coordonnées du centre de masse** (`center_of_mass_x/y/z`) : obtenues par moyenne pondérée des positions atomiques, selon leur masse.
- **Rayon de giration** (`radius_of_gyration`) : mesure de la compacité de la molécule autour de son centre de masse.
- **Distances interatomiques** : moyenne, minimum et maximum entre tous les couples d'atomes (`mean_distance`, `min_distance`, `max_distance`).

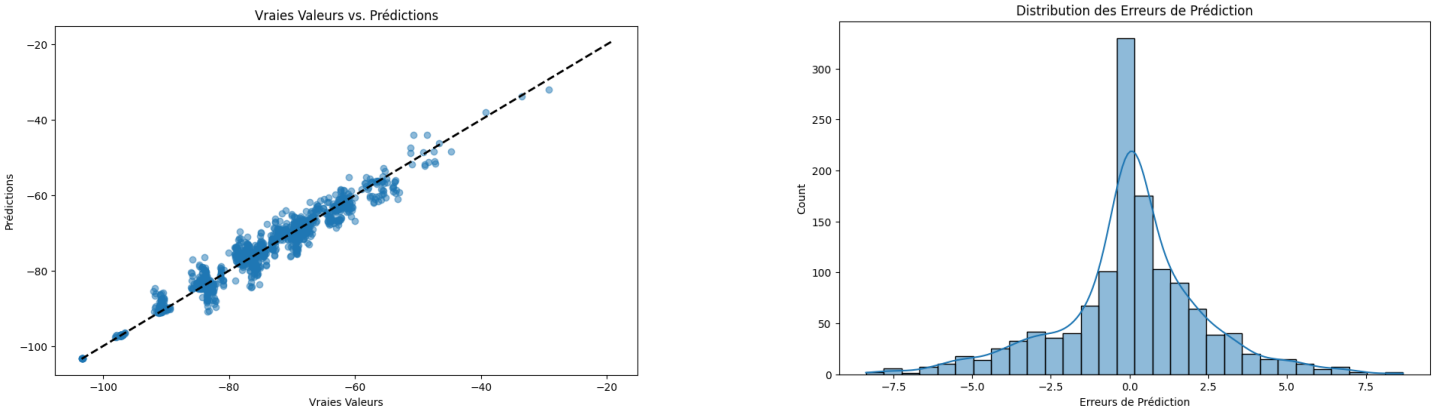
Le **rayon de giration** est défini par la formule suivante :

$$R_g = \sqrt{\frac{1}{M} \sum_{i=1}^{N} m_i \cdot \|\vec{r}_i - \vec{r}_{\text{com}}\|^2}$$

où m_i est la masse de l'atome i, $\vec{r}_i$ sa position, $\vec{r}_{\text{com}}$ le centre de masse de la molécule, et $M = \sum_i m_i$ la masse totale. Cette mesure donne une estimation quantitative de la dispersion des atomes autour du centre de masse.

Les caractéristiques extraites sont ensuite stockées dans un dictionnaire, puis regroupées dans un `DataFrame pandas`. Si un fichier `.csv` contenant les énergies est fourni, celles-ci sont associées à chaque molécule grâce à leur identifiant. Le résultat est une base de données structurée où chaque ligne correspond à une molécule représentée par un vecteur de descripteurs.

Nous avons ensuite mis en place notre modèle Random Forest qui prend en entrée un vecteur représentant les caractéristiques de chaque molécule. Nous obtenons une RMSE de 2.3099.



(a) Prédictions du modèle Random Forest.

(b) Histogramme des erreurs de prédiction.

Figure 5: Résultats du modèle Random Forest.

On peut voir que les résultats sont meilleurs que ceux de la régression linéaire, ce qui correspond à une RMSE plus faible. Cela est confirmé par la distribution de l'erreur, qui est concentrée autour de 0. Nous arrivons donc à améliorer les résultats en extrayant certaines caractéristiques physiques de nos données. Voyons maintenant comment extraire la physique d'une autre manière.

4 Matrice de Coulomb

La matrice de Coulomb M est une matrice symétrique définie par la relation suivante pour deux atomes i et j :

$$M_{ij} = \begin{cases} 0.5 Z_i^{2.4} & \text{si } i = j \\ \frac{Z_i Z_j}{\|\mathbf{R}_i - \mathbf{R}_j\|} & \text{si } i \neq j \end{cases}$$

où Z_i est le numéro atomique de l'atome i et $\mathbf{R}_i$ sa position spatiale. Les éléments diagonaux approximeraient l'énergie atomique en fonction du numéro atomique, tandis que les éléments hors diagonale modélisent la répulsion coulombienne entre paires d'atomes. On trie ensuite les lignes et les colonnes avec la norme L2. Cela nous assure que la matrice sera identique pour un même molécule quelque soit sa position, ou si elle a subi une rotation, etc ...

Cette matrice présente plusieurs propriétés intéressantes : elle est invariante par translation et rotation, et peut être rendue invariante par permutation des atomes via des techniques de tri (comme c'est notre cas). Elle permet ainsi de capturer les principales interactions structurelles dans une molécule tout en restant relativement simple à calculer.

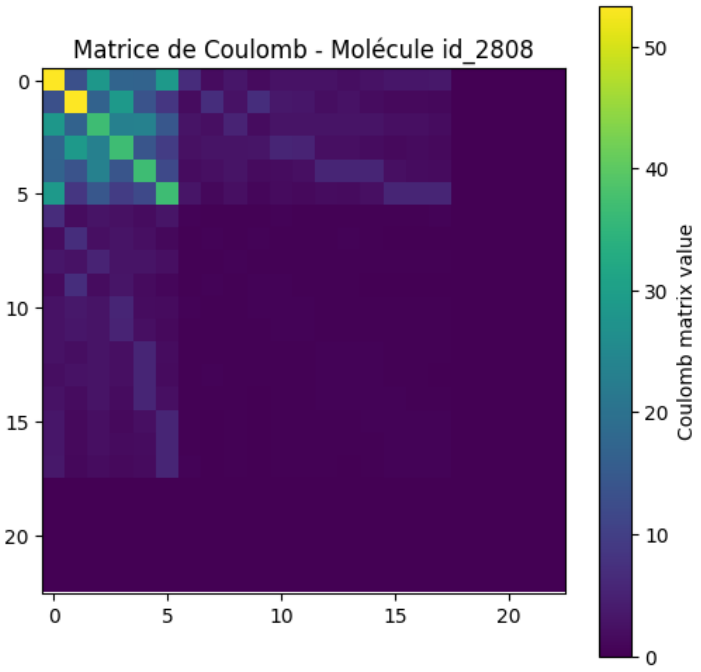


Figure 6: Exemple d'une matrices de Coulomb

Dans cette étude, la matrice de Coulomb a été utilisée comme descripteur global pour chaque molécule. Elle est ensuite étendue en une matrice de taille unique, avec un *padding* par zéros afin de garantir une taille fixe pour l'ensemble du jeu de données. Cette matrice est ensuite transformée sous forme de vecteur. Ce dernier sert d'entrée aux modèles de régression supervisée utilisés pour prédire l'énergie moléculaire.

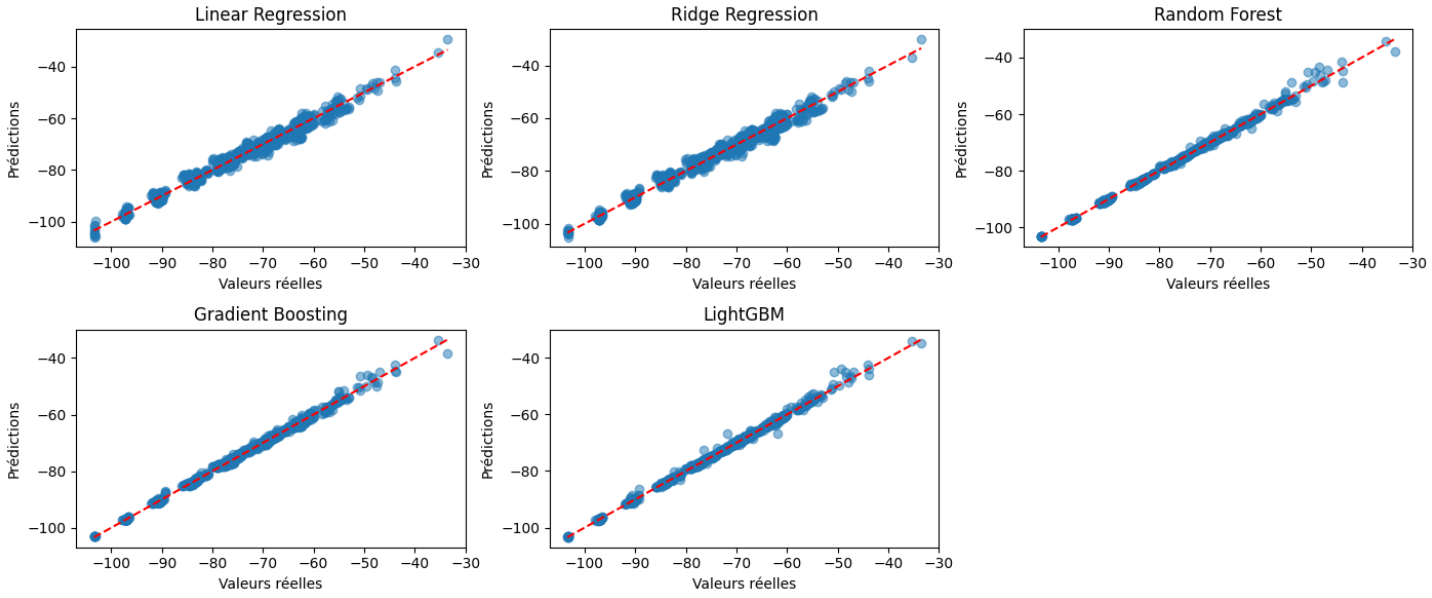


Figure 7: Prédictions de l'énergie depuis la matrice de Coulomb

Modèle	MAE	RMSE	R^2
Linear Regression	1.0795	1.4305	0.9852
Ridge Regression ($\alpha = 0.1$)	1.0783	1.4373	0.9850
Random Forest	0.2139	0.5083	0.9981
Gradient Boosting	0.4075	0.6002	0.9974
LightGBM	0.2480	0.5037	0.9982

Table 2: Résultats des différents modèles de régression

Nous pouvons observer que la matrice de Coulomb se révèle être un descripteur particulièrement efficace pour nos données, vérifiant les propriétés attendues telles que les invariances par translation, rotation et permutation.

Parmi les modèles testés, le LightGBM se distingue comme le plus performant, affichant une RMSE de 0,5037 sur le jeu de test. Cela montre que ce modèle arrive à prédire les résultats avec une erreur relativement faible.

D'autres modèles, comme Random Forest, sont également très efficaces, avec une RMSE de 0,5083, ce qui est à peine moins bien que le LightBGM, mais tout de même très proche en termes de précision.

Les modèles Gradient Boosting et Linear Regression affichent des RMSE respectivement de 0,6002 et 1,4305. Bien qu'ils soient moins performants que les deux modèles mentionnés précédemment, leurs résultats sont encore relativement bons par rapport à d'autres méthodes plus classiques.

La Régression Ridge présente la RMSE la plus élevée parmi les modèles évalués, avec une valeur de 1,4305, ce qui indique une erreur plus importante par rapport aux autres méthodes.

5 Scattering Harmonique pour l'extraction de caractéristiques

5.1 Explication du scattering

Dans cette section, nous nous sommes appuyés sur le code fourni en annexe du sujet du projet, qui utilise la bibliothèque Python `Kymatio`. Il nous a donc été nécessaire de comprendre son fonctionnement afin de pouvoir le modifier et l'adapter à notre problème spécifique.

5.1.1 Génération des Gaussiennes en 3D

Tout d'abord, nous chargeons nos molécules et calculons trois types de charges pour chaque molécule, qui seront utilisées par la suite : les charges nucléaires complètes, les charges de valence et les charges de cœur, qui sont la différence entre les charges nucléaires et de valence.

Ensuite, les distances entre les positions atomiques sont normalisées afin d'assurer un recouvrement optimal des Gaussiennes. Nous définissons alors la grille tridimensionnelle (M, N, O) sur laquelle nous projetons nos molécules.

Afin de pouvoir utiliser la transformée de scattering 3D sur nos molécules, il est nécessaire de construire une représentation continue de celles-ci dans l'espace tridimensionnel. Pour cela, nous avons projeté chaque molécule sur une grille 3D à l'aide d'une somme pondérée de fonctions de Gauss. Chaque atome est ainsi représenté par une gaussienne centrée sur sa position, pondérée par une charge atomique spécifique (nucléaire, de valence ou de cœur), et lissée selon un paramètre d'écart-type σ. Ce procédé permet de passer d'une représentation atomique discrète à un champ scalaire 3D continu, adapté aux méthodes d'analyse du signal comme la transformée de scattering. Nous assurons ainsi l'invariance par permutation.

Cette densité est ensuite convertie en un tenseur PyTorch afin de garantir la compatibilité avec le calcul GPU, puis deux types de descripteurs sont extraits :

- **Les intégrales d'ordre 0** : Ces dernières correspondent à des moments globaux de la densité. Elles fournissent des informations simples, globales et interprétables, mais ne sont pas directement capturées par la transformée de scattering.
- **Les coefficients de scattering** : Ces coefficients extraient des informations multi-échelles, locales et invariantes par translation et rotation sur la forme de la densité. Ils constituent des descripteurs plus riches, mais également plus abstraits. Nous détaillerons leur construction plus loin.

La transformée de scattering applique déjà des opérations de type intégrale (agrégations globales après filtrage), et elle capture directement les informations globales brutes au travers de son ordre 0. Ces informations, bien que simples, sont souvent très discriminantes entre différentes molécules.

5.1.2 Application du Scattering Harmonique et Calcul des Invariants

Dans le cadre de la transformée de scattering harmonique 3D, deux paramètres principaux contrôlent la richesse et la résolution des descripteurs extraits :

- J : nombre d'échelles analysées. Chaque échelle $j \in \{0, \ldots, J\}$ correspond à une dilatation 2^{-j}. Plus J est élevé, plus le signal est analysé à grande échelle, capturant ainsi des structures globales.
- L : nombre de directions angulaires, c'est-à-dire le nombre d'orientations sphériques différentes utilisées pour explorer la géométrie du signal. Plus L est élevé, plus les descripteurs capturent des détails directionnels fins.

Ces deux paramètres contrôlent la **résolution multi-échelle** et la **sensibilité directionnelle** des ondelettes utilisées pour le filtrage du signal.

La transformée scattering 3D est une méthode de traitement de signal qui permet d'extraire des descripteurs invariants à partir de signaux tridimensionnels. La méthode repose sur l'utilisation d'ondelettes harmoniques solides, qui sont des fonctions de base permettant de capturer des informations multi-échelles tout en préservant l'invariance géométrique.

La base du processus de scattering repose sur la construction des *ondelettes harmoniques solides*, définies à partir des harmoniques sphériques. Une ondelette $\psi_\ell^m(u)$ est donnée par la relation suivante :

$$\psi_\ell^m(u) \propto e^{-\frac{|u|^2}{2}} |u|^\ell Y_\ell^m \left(\frac{u}{|u|} \right)$$

Les ondelettes harmoniques solides sont ensuite dilatées pour couvrir plusieurs échelles de résolution. Cette dilatation est effectuée en modifiant l'échelle des ondelettes, selon un paramètre j qui contrôle la taille de l'ondelette :

$$\psi_{j,\ell}(u) = 2^{-3j} \psi_\ell^m(2^{-j} u)$$

Ici, j contrôle l'échelle de l'ondelette. Plus j est grand, plus l'ondelette couvre une large portion de l'espace. Le facteur 2^{-3j} permet de normaliser l'amplitude de l'ondelette à différentes échelles.

Une fois les ondelettes construites et dilatées, elles sont utilisées pour filtrer un signal $x(u)$ par convolution. Pour chaque échelle j et chaque orientation ℓ, on applique l'ondelette $\psi_{j,\ell}$ au signal $x(u)$.

Au premier ordre, on applique la convolution de $x(u)$ avec $\psi_{j,\ell}$:

$$U_{p_1} x(u) = \Big(\sum_{m=-\ell}^{\ell} \big| \big(x * \psi_{j,\ell}^m(u) \big) \big|^2 \Big)^{1/2}$$

Cette opération produit une première représentation du signal à une échelle donnée, capturant ainsi des informations locales sur le signal.

Ensuite, une deuxième couche de filtrage est appliquée en convoluant $U_{p_1} x(u)$ avec une ondelette dilatée $\psi_{j_2,\ell}$:

$$U_{p_2} x(u) = \Big(\sum_{m=-\ell}^{\ell} \big| \big(U_{p_1} x * \psi_{j_2,\ell}^m(u) \big) \big|^2 \Big)^{1/2}$$

Cela permet d'extraire des détails supplémentaires et plus fins sur la structure du signal à une échelle encore plus précise.

Les invariants sont extraits en intégrant les réponses filtrées $U_{p_1} x(u)$ et $U_{p_2} x(u)$ sur tout l'espace u, avec un exposant $q > 0$:

$$\Phi(x) = \Big\{ \int |U_{p_1} x(u)|^q \, du, \quad \int |U_{p_2} x(u)|^q \, du \Big\}$$

Ces intégrales mesurent l'intensité globale des réponses filtrées et fournissent des descripteurs invariants aux transformations géométriques telles que la translation et la rotation. Ainsi, $\Phi(x)$ est un ensemble de caractéristiques qui ne varient pas lorsque l'objet analysé est déplacé ou tourné dans l'espace tridimensionnel.

La méthode de scattering 3D permet d'extraire des descripteurs invariants à partir d'un signal tridimensionnel, en appliquant des ondelettes harmoniques solides à différentes échelles. Ces descripteurs sont à la fois multi-échelles et invariants aux translations et rotations.

Le code mis en place calcule donc les coefficients du scattering pour les différentes molécules et les sauvegarde. Les coefficients sont ensuite utilisés pour effectuer la régression afin de prédire les énergies d'atomisation.

5.1.3 Nombre de coefficients obtenus par la transformée de scattering

Regardons le nombre de coefficients produits par la transformée de scattering harmonique. Nous utilisons ici un scattering d'ordre 1 et 2, ainsi qu'une intégration sur plusieurs puissances. Soit q le nombre de puissances d'intégration utilisées (correspondant au paramètre `integral_powers`).

- Pour l'ordre 1, le nombre de coefficients produits est :

$$(J+1) \times (L+1) \times \mathrm{q}$$

- Pour l'ordre 2, le nombre de coefficients est :

$$\frac{J(J+1)}{2} \times (L+1) \times \mathrm{q}$$

Dans nos expériences, nous avons utilisé $J = 2$, $L = 3$, et $\mathrm{q} = 4$. Cela donne :

- Ordre 1 : $(2+1) \times (3+1) \times 4 = 3 \times 4 \times 4 = 48$ coefficients
- Ordre 2 : $\frac{2\times3}{2} \times 4 \times 4 = 3 \times 4 \times 4 = 48$ coefficients

En concaténant les deux ordres, on obtient $48 + 48 = 96$ coefficients. À cela s'ajoutent les coefficients de l'ordre 0, qui correspondent à $\mathrm{q} \times 1 = 4$ valeurs.

Au total, la transformée de scattering renvoie donc $96 + 4 = 100$ coefficients par molécule.

Comme nous faisons 3 scattering différents (sur toutes les charges, les charges de valence et celles de coeur), nous concaténons les descripteurs, ce qui donne $3 \times 100 = 300$ coefficients finaux par molécule.

5.2 Régression sur le scattering

Nous testons donc plusieurs valeurs de grille afin d'obtenir les meilleurs résultats :

Modèle	MAE	RMSE
Grille 16x16x16		
Ridge ($\alpha = 0.1$)	8.83	13.11
Ridge ($\alpha = 1$)	8.84	10.93
Ridge ($\alpha = 10$)	8.94	10.85
Lasso	9.48	11.48
ElasticNet	9.48	11.46
Random Forest	**8.74**	**10.79**
SVR	9.28	11.26
XGBoost	9.17	11.41
MLP	9.09	11.04

Modèle	MAE	RMSE
Grille 32x32x32		
Ridge ($\alpha = 0.1$)	**5.30**	**6.81**
Ridge ($\alpha = 1$)	5.52	7.00
Ridge ($\alpha = 10$)	5.73	7.28
Lasso	7.28	9.01
ElasticNet	7.09	8.81
Random Forest	6.07	7.76
SVR	6.57	8.38
XGBoost	6.11	7.89
MLP	5.96	7.58

Modèle	MAE	RMSE
Grille 64x64x64		
Ridge ($\alpha = 0.1$)	**0.28**	**0.55**
Ridge ($\alpha = 1$)	0.35	0.66
Ridge ($\alpha = 10$)	0.50	0.88
Lasso	2.31	3.05
ElasticNet	2.18	2.86
Random Forest	0.37	1.21
SVR	1.17	2.41
XGBoost	0.41	1.17
MLP	2.24	4.18

Modèle	MAE	RMSE
Grille 96x64x48		
Ridge ($\alpha = 0.1$)	**0.58**	**1.14**
Ridge ($\alpha = 1$)	0.71	1.33
Ridge ($\alpha = 10$)	0.90	1.63
Lasso	2.92	3.95
ElasticNet	2.76	3.76
Random Forest	0.75	1.86
SVR	1.79	3.26
XGBoost	0.81	1.77
MLP	2.13	3.44

Table 3: Comparaison des performances (MAE et RMSE) des modèles sur les différentes grilles. La meilleure RMSE est surlignée en vert, la pire en rouge.

Dans cette étude, nous avons comparé les performances de plusieurs modèles de régression sur quatre grilles de tailles croissantes : 16^3, 32^3, 64^3 et une grille $96 \times 64 \times 48$. L'évaluation a été réalisée à l'aide de la **Root Mean Squared Error (RMSE)**.

Grille 16×16×16

Sur la plus petite grille, le modèle **Random Forest** obtient la meilleure performance avec une RMSE de **10.79**, surpassant ainsi toutes les régressions linéaires et non linéaires. À l'inverse, la **régression ridge** avec un faible coefficient de régularisation ($\alpha = 0.1$) présente la plus mauvaise performance avec une RMSE de **13.11**. Cela suggère que pour des représentations peu résolues, les modèles non linéaires comme les forêts aléatoires sont plus capables de capturer les relations complexes dans les données.

Grille 32×32×32

Sur cette grille de taille moyenne, la tendance s'inverse. La **régression ridge avec** $\alpha = 0.1$ devient le modèle le plus performant avec une RMSE de **6.81**. Cela indique qu'à cette résolution, les régularisations légères des modèles linéaires peuvent capturer efficacement l'information sans sur-ajuster. Le pire résultat est obtenu par la **régression Lasso**, avec une RMSE de **9.01**, probablement en raison de son effet de sélection de variables trop agressif dans un espace de représentation encore peu structuré.

Grille 64×64×64

À cette résolution plus fine, les résultats confirment l'avantage des modèles linéaires bien régularisés. La **régression ridge avec** $\alpha = 0.1$ obtient la meilleure performance avec une RMSE remarquablement basse de **0.55**. Les modèles non linéaires tels que **MLP** ou **SVR** présentent des RMSE beaucoup plus élevées (jusqu'à **4.18**), ce qui laisse penser qu'ils peinent à généraliser dans un espace de très grande dimension sans optimisation poussée.

Grille 96×64×48

Enfin, sur cette grille, la régression ridge avec $\alpha = 0.1$ reste en tête avec une RMSE de **1.14**. Les performances des autres modèles, notamment des méthodes plus complexes comme **MLP** ou **SVR**, restent nettement inférieures avec des RMSE proches ou supérieures à **3.0**.

L'ensemble de ces résultats met en évidence que la performance des modèles dépend fortement de la résolution de la grille utilisée. À basse résolution, des modèles non linéaires comme Random Forest offrent les meilleures performances. En revanche, à mesure que la résolution augmente, les modèles linéaires régularisés, en particulier la régression ridge avec une faible valeur de α, deviennent systématiquement les plus performants. Les meilleurs résultats sont obtenus avec une grille 80x80x80.

5.3 Régression Multi-Linéaire sur le scattering

Le module `ElementwiseProd`, implémenté en `PyTorch`, est une architecture personnalisée conçue pour transformer un vecteur d'entrée en exploitant le produit élément par élément de plusieurs transformations linéaires activées.

Soit un vecteur d'entrée $x \in \mathbb{R}^d$, où d correspond à `input_dim`. Le module applique k transformations linéaires indépendantes $\{W_i x + b_i\}_{i=1}^k$, suivies d'une même fonction d'activation non linéaire ϕ, ici la fonction sigmoïde. Ces sorties activées sont ensuite combinées par un produit élément par élément (*Hadamard product*).

On définit k couches linéaires de sortie de dimension q, avec matrices de poids $W_i \in \mathbb{R}^{q \times d}$ et biais $b_i \in \mathbb{R}^q$ pour $i = 1, \ldots, k$. Pour une entrée $x \in \mathbb{R}^d$, chaque couche produit un vecteur activé :

$$z_i = \phi(W_i x + b_i) \in \mathbb{R}^q$$

Le vecteur de sortie final $y \in \mathbb{R}^q$ est obtenu par :

$$y = \prod_{i=1}^{k} z_i = z_1 \odot z_2 \odot \cdots \odot z_k$$

où $\odot$ désigne le produit élément par élément. Autrement dit, chaque composante y_j du vecteur final est donnée par :

$$y_j = \prod_{i=1}^{k} \phi\left((W_i x + b_i)_j\right), \quad \text{pour } j = 1, \ldots, q$$

Ce module peut être interprété comme une attention implicite et multiplicative : chaque transformation extrait une "vue" partielle de l'information, et seules les dimensions activées de façon cohérente par plusieurs couches contribuent fortement à la sortie finale. Il agit donc comme un filtre multiplicatif qui renforce ou supprime certaines dimensions en fonction de l'accord entre les différentes transformations linéaires.

L'idée est donc d'utiliser la sortie produite par le scattering dans un réseau simple effectuant ce produit expliqué précédemment. Nous reprenons donc les sorties produites par notre scattering avec la grille 64x64x64 et faisons une cross-validation sur notre modèle.
Nous obtenons les résultats suivants : **MAE: 0.4824701126275599, RMSE: 1.0871054336355932**.
Ainsi, cette méthode présentée en TP que nous avons mise en place dans ce projet propose des résultats intéressants qui sont meilleurs que certains présentés précédemment. Néanmoins, nous avons obtenu des meilleurs résultats en combinant le scattering et la régression Ridge.

5.4 Régression augmentée

5.4.1 Descripteurs extraits depuis les fichiers XYZ

Nous avons vu l'efficacité d'utiliser le scattering en preprocessing. Nous avons donc compris l'importance de cette étape de preprocessing. Nous avons donc décidé d'utiliser de nouvelles features afin d'obtenir de meilleurs résultats. Pour cela, la fonction `extract_features_from_xyz_inv` calcule un ensemble de caractéristiques physiques et géométriques à partir d'une molécule représentée sous forme de fichier `.xyz`. Ces descripteurs sont conçus pour être invariants aux translations, rotations, et permutations des atomes.

Voici la liste des descripteurs extraits :

1. Descripteurs géométriques et de masse

- `num_atoms` : nombre total d'atomes dans la molécule.
- `total_mass` : somme des masses atomiques.
- `mean_distance`, `min_distance`, `max_distance` : statistiques sur les distances interatomiques (hors diagonale).
- `bond_length_std`, `bond_length_moment_2`, `bond_length_moment_3` : écart-type et moments d'ordre 2 et 3 des longueurs de liaison.

2. Descripteurs invariants de forme

- `radius_of_gyration` : mesure la dispersion des masses autour du centre de masse (rayon de giration).
- `inertia_eig_0`, `inertia_eig_1`, `inertia_eig_2` : valeurs propres du tenseur d'inertie (triées par ordre croissant), représentant la distribution des masses autour des axes principaux.
- `mass_proj_var_0`, `mass_proj_var_1`, `mass_proj_var_2` : variances des projections des masses sur les axes principaux d'inertie.

3. Descripteurs basés sur la matrice de Coulomb La matrice de Coulomb est définie comme suit :

$$C_{ij} = \begin{cases} 0.5 Z_i^{2.4} & \text{si } i = j \\ \frac{Z_i Z_j}{\|R_i - R_j\|} & \text{si } i \neq j \end{cases}$$

où Z_i est le numéro atomique de l'atome i, et R_i sa position.

- `coulomb_mean`, `coulomb_std`, `coulomb_min`, `coulomb_max` : statistiques sur les éléments hors-diagonaux.
- `coul_spec_0` à `coul_spec_9` : 10 plus grandes valeurs propres de la matrice de Coulomb (spectre trié décroissant).

4. Descripteurs angulaires (angles triatomiques) Ces descripteurs mesurent les angles formés par des triplets d'atomes dans la molécule. Ils reflètent la géométrie locale des structures atomiques.

- `angle_mean`, `angle_std`, `angle_min`, `angle_max` : statistiques sur les angles triatomiques exprimés en degrés.

5. Skewness des coordonnées centrées Ces descripteurs mesurent l'asymétrie de la distribution spatiale des atomes centrée au barycentre :

- `skew_x`, `skew_y`, `skew_z` : asymétrie (skewness) sur chaque axe après centrage de la position des atomes.

6. Propriétés d'invariance Les descripteurs extraits ont été conçus pour respecter certaines invariances géométriques. Les caractéristiques telles que `total_mass`, `num_atoms`, ou encore `radius_of_gyration` dépendent uniquement de grandeurs globales ou relatives, ce qui les rend invariantes par translation et rotation. Le rayon de giration, les valeurs propres du tenseur d'inertie et les variances projetées sont invariants par rotation, car ils sont calculés à partir des positions centrées au barycentre, et ne dépendent que de la distribution relative des masses. Les distances interatomiques et les angles sont également invariants par rotation et translation.

Concernant la matrice de Coulomb, les valeurs propres (spectre) sont invariantes par permutation des atomes, et toutes les caractéristiques issues de la matrice de Coulomb sont invariantes par rotation et translation, car elles reposent sur les distances interatomiques.

Enfin, les descripteurs de skewness sont calculés à partir des coordonnées centrées, ce qui les rend invariants par translation.

Ces invariances sont essentielles pour garantir que les descripteurs capturent uniquement la structure intrinsèque de la molécule, indépendamment de sa position, son orientation ou de l'ordre des atomes dans le fichier.

5.4.2 Résultats

Nous obtenons au final 36 features en plus. Nous essayons plusieurs méthodes de régression, et nous calculons leur MAE et leur RMSE par cross-validation. Nous obtenons les résultats de la table 5. Le meilleur modèle est une régression avec une régularisation Ridge avec un $\alpha = 0.001$. Nous obtenons une RMSE de 0.163, ce qui est le meilleur résultat obtenu. Lors du test sur Kaggle, nous obtenons une RMSE sur le jeu de test de 0.146, montrant que nous avons pas d'overfitting.

Modèle	**MAE**	**RMSE**
Linear Regression	0.0865	0.4362
Ridge Regression ($\alpha = 10^{-6}$)	0.0852	0.3448
Ridge Regression ($\alpha = 10^{-5}$)	0.0846	0.2061
Ridge Regression ($\alpha = 10^{-4}$)	0.0881	0.1643
Ridge Regression ($\alpha = 10^{-3}$)	**0.0971**	**0.1633**
Ridge Regression ($\alpha = 10^{-2}$)	0.1127	0.1819
Ridge Regression ($\alpha = 10^{-1}$)	0.1349	0.2011
Ridge Regression ($\alpha = 1$)	0.1778	0.2534
Ridge Regression ($\alpha = 10$)	0.2685	0.3959
Lasso Regression	1.9281	2.3723
ElasticNet Regression	1.8457	2.3215
ElementwiseProdRegressor	1.5895	3.0488
Support Vector Regression	0.9168	2.1559
XGBoost Regression	0.2220	0.5326
MLP Regressor	1.6920	3.3666

Table 4: Comparaison des modèles de régression — MAE et RMSE

Grille 80×80×80

Nous décidons d'essayer une grille plus fine pour améliorer les résultats, tout en conservant les autres features. Comme pour toutes les autres grilles, le modèle donnant les meilleurs résultats est la régression de Ridge. La meilleure performance par validation croisée sur le jeu d'entraînement est obtenue pour $\alpha = 10^{-4}$, avec une RMSE de 0.0972. Sur le jeu de test partiel, on obtient une RMSE valant 0.140, et une RMSE de 0.092 sur le jeu de test complet.

Modèle	MAE	RMSE
Linear Regression	0.0438	0.2956
Ridge Regression ($\alpha = 10^{-6}$)	0.0455	0.2084
Ridge Regression ($\alpha = 10^{-5}$)	0.0458	0.1348
Ridge Regression ($\alpha = 10^{-4}$)	0.0484	**0.0972**
Ridge Regression ($\alpha = 10^{-3}$)	0.0595	0.2002
Ridge Regression ($\alpha = 10^{-2}$)	0.0787	0.2430
Ridge Regression ($\alpha = 10^{-1}$)	0.1059	0.1630
Ridge Regression ($\alpha = 1$)	0.1528	0.2159
Ridge Regression ($\alpha = 10$)	0.2478	0.3709

Table 5: Résultats des différents modèles Ridge et linéaire (MAE et RMSE)

6 Conclusion

Pour conclure ce rapport sur la prédiction d'énergies moléculaires sous contraintes physiques, nous avons exploré plusieurs approches visant à améliorer la précision de nos modèles tout en respectant les propriétés des molécules. Les défis posés par les invariances de translation, de rotation et de permutation ont été abordés à travers différentes méthodes.

Nous avons commencé par une analyse descriptive des données moléculaires, mettant en évidence la diversité structurelle et la composition des molécules dans notre jeu de données. Cette diversité s'est avérée importante pour la robustesse de nos modèles. L'analyse des énergies a révélé une distribution centrée autour de -80 eV, avec quelques outliers, ce qui a permis de contextualiser les performances des modèles et de comparer nos erreurs.

Les premiers modèles de prédiction, notamment la régression linéaire et le modèle Random Forest, ont montré des résultats prometteurs. La régression linéaire, bien que simple, a fourni une base solide avec un score R^2 élevé. Le modèle Random Forest a amélioré ces résultats en exploitant des caractéristiques physiques supplémentaires, réduisant ainsi la RMSE.

L'introduction de la matrice de Coulomb comme descripteur a montré certains avantages. Cette matrice, invariante par translation et rotation sur les molécules, a permis d'obtenir des résultats satisfaisants avec divers modèles de régression, le LightGBM se distinguant avec une RMSE de 0,5037.

L'utilisation de la transformée de scattering harmonique pour l'extraction de caractéristiques a également été explorée. Cette méthode a permis de capturer des informations multi-échelles et invariantes, améliorant ainsi la précision des modèles. Les résultats ont montré que les modèles linéaires bien régularisés, comme la régression ridge, étaient particulièrement efficaces à haute résolution. De plus, cette approche a non seulement démontré son efficacité dans notre étude, mais a également atteint un score de 0.092 sur le tableau de classement Kaggle. Ce résultat témoigne de la robustesse et de la précision de notre modèle.

Enfin, la régression multi-linéaire sur le scattering a apporté des résultats intéressants, bien que moins performants que ceux obtenus avec la régression ridge. Cette approche a néanmoins démontré le potentiel des architectures personnalisées pour transformer et combiner des caractéristiques de manière non linéaire.

En somme, ce projet a mis en lumière l'importance de combiner des descripteurs physiques pertinents avec des modèles d'apprentissage adaptés. Les résultats obtenus ouvrent la voie à des améliorations futures, notamment en explorant des architectures de réseaux de neurones plus complexes et en intégrant des connaissances physiques plus approfondies. Les perspectives d'amélioration incluent également l'utilisation de jeux de données plus larges et plus diversifiés, ainsi que l'exploration de nouvelles méthodes pour capturer les invariances et les symétries moléculaires.