PLACEHOLDER

Joan Tibau Terma

Resum — La simulació de dinàmica molecular (DM) és una tècnica computacional que estudia sistemes moleculars, proporcionant informació sobre l'estructura, l'energia, la dinàmica i altres propietats. Té aplicacions en la indústria farmacèutica, biologia, enginyeria de materials, física i química. S'utilitzen mètodes com Verlet per a resoldre les equacions del moviment simultàniament a mètodes de mecànica quàntica per a calcular les forces i energies d'interacció entre àtoms. Amb sistemes complexos, es requereixen tècniques d'optimització i paral·lelització. Les noves tècniques d'aprenentatge computacional, com les xarxes neuronals, ofereixen alternatives per abordar els problemes de complexitat de la DM. Aquest projecte té com a objectiu de l'estudi i l'aplicació de xarxes neuronals a simulacions físiques per a la predicció de propietats moleculars. S'utilitzat la toolbox SchNetPack2 que ptoporciona eines per a aplicar xarxes neuronals per a la predicció de propietats de bases de dades com ara QM9. S'ha realitzat una anàlisi detallada dels resultats obtinguts i s'han extret conclusions rellevants per a la millora del model.

Paraules clau — Analítica de Dades, xarxes neuronals, dinàmica molecular, base de dades QM9, SchNet, predicció de propietats, anàlisi de resultats, optimització.

Abstract — Molecular dynamics (MD) simulation is a computational technique that studies molecular systems, providing information about structure, energy, dynamics, and other properties. It has applications in the pharmaceutical industry, biology, materials engineering, physics, and chemistry. Numerical methods like Verlet or Gear are used to solve the equations of motion and calculate interaction forces. With complex systems, optimization and parallelization techniques are required. New computational learning techniques, such as neural networks, offer alternatives to address the complexity of MD. This project aims to study and apply neural networks to physical simulations for the prediction of molecular properties. The SchNetPack2 toolbox is used, providing tools for applying neural networks to predict properties from databases like QM9. A detailed analysis of the obtained results has been conducted, drawing relevant conclusions for model improvement.

Index Terms — Data Science, neural networks, molecular dynamics, QM9 database, SchNet, property prediction, result analysis, optimization.

1 Introducció:

La simulació de dinàmica molecular (DM) és una tècnica computacional amplament utilitzada per a l'estudi de sistemes moleculars. Proporciona informació clau sobre l'estructura, l'energia, la dinàmica i altres propietats de les molècules. Aquesta tècnica té una àmplia gamma d'aplicacions en àrees com la indústria farmacèutica, la biologia, l'enginyeria de materials, la física i la química. Per dur a terme aquestes simulacions, s'usen mètodes numèrics com l'algorisme de Verlet, que resol les equacions del moviment, juntament amb mètodes de mecànica quàntica per ~~a~~ calcular les forces i les energies d'interacció entre àtoms.

El document està organitzat amb els següents apartats:

- Exposició de les motivacions junt amb un breu context del camp de les Xarxes Neuronals (XN) aplicades a la DM i objectius del treball (punt 2).
- Revisió l'*state of the art* del camp de les XN aplicades a la DM, tant bases de dades com models, *frameworks* o *toolbox* existents (punt 3).
- Explicació de la metodologia emprada i exposició dels resultats obtinguts de les simulacions analitzats en detall (punt 4 i 5).
- Presentació de les conclusions rellevants i les possibles accions futures per millorar i ampliar aquest treball (punt 6).
- Agraïments (punt 7).
- Bibliografia i annexos (punt 8).

- *E-mail de contacte: jotite19@gmail.com*
- *Menció realitzada: Computació*
- *Treball tutoritzat per: Ramon Baldrich*
- *Tutoria externa per: Jordi Faraudo*
- *Curs 2022/23*

2 Motivació i objectius:

Un dels principals problemes del camp de la DM és que a mesura que els sistemes es fan més complexos augmenta la quantitat de dades a processar i, per tant, augmenta també el temps requerit per a cada pas de la simulació. Per exemple, en el cas d'una molècula orgànica relativament petita amb aproximadament 30 àtoms compostos exclusivament de Carboni (C), Hidrogen (H), Nitrogen (N) i Oxigen (O), el temps de càlcul per obtenir l'energia fonamental pot variar des de diverses hores fins a dies en equips informàtics estàndard. Aquesta estimació depèn de factors com la capacitat computacional, el mètode de càlcul usat i la complexitat de les interaccions entre àtoms. Això ha conduït a la necessitat de tècniques d'optimització i paral·lelització per accelerar el procés d'obtenció de mètriques. Recentment, s'han proposat noves estratègies basades en l'aprenentatge computacional, com l'ús de XN, per abordar els reptes de la complexitat de la DM.

En aquest projecte es plantegen els següents objectius per millorar l'estudi i l'aplicació de XN en simulacions físiques per a la predicció de propietats moleculars:

- **Objectiu 1**: Desenvolupar un coneixement profund dels fonaments teòrics i pràctics de les XN i la simulació de la DM.

- **Objectiu 2**: Desenvolupar una comprensió crítica dels avantatges i les limitacions de les XN en la simulació de la DM, comparant-les amb altres tècniques i abordatges existents.

- **Objectiu 3**: Estudiar els treballs més recents en el camp de la DM que apliquen XN, posant el focus en els següents aspectes: bases de dades utilitzades en els treballs recents relacionats amb la simulació de la DM i les XN; mètodes de representació dels sistemes de molècules usats en aquests treballs; arquitectures de les XN utilitzades en els treballs recents i la seva eficàcia i avaluar el rendiment i els resultats obtinguts amb els mètodes de simulació de la DM basats en les XN.

- **Objectiu 4**: Realitzar estudis específics per millorar l'eficiència i l'optimització dels models de simulació de la DM basats en les XN: reforçar els coneixements sobre les bases de dades utilitzades en el camp de la simulació de la DM; realitzar un estudi dels hiperparàmetres computacionals, com ara el batch size, el nombre d'epochs i el learning rate, per determinar la seva influència en els resultats de les simulacions i realitzar un estudi dels mòduls de les propietats físiques que es poden incloure en els models de simulació de la DM basats en les XN per millorar la predicció de propietats moleculars.

3 State of the Art:

3.1 Bases de dades:

En el camp del *machine learning* aplicat a la DM, són fonamentals les bases de dades com la MD17, la QM9 i d'altres. Aquestes bases proporcionen una gran quantitat de dades experimentals -obtingudes mitjançant simulacions quàntiques utilitzant mètodes tradicionals- que permeten entrenar models predictius per comprendre i simular el comportament de molècules i materials a escala atòmica.

3.2.1 QM9:

La base de dades QM9 és una col·lecció de dades moleculars que proporciona informació essencial sobre propietats químiques i físiques de diverses molècules orgàniques petites. Aquesta base de dades està basada en la base de dades GB17[1], que inclou 134.000 molècules orgàniques estables compostes pels elements C, H, O, N i F. Les dades provenen de simulacions quàntiques usant mètodes tradicionals i proporcionen informació detallada sobre propietats geomètriques, energètiques, electròniques i termodinàmiques de les molècules.

Les molècules de la base de dades QM9 corresponen a un subconjunt de 133.885 espècies amb un màxim de nou àtoms pesats (C, O, N, F) de "l'univers químic GDB-17", que conté 166 mil milions de molècules orgàniques. Es proporcionen les propietats com ara: geometries, freqüències harmòniques, moments de dipol, polaritzabilitats, l'energia fonamental, entalpies, entre d'altres. Totes les propietats es calculen utilitzant la química quàntica a nivell B3LYP/6-31G(2df,p).

Detalls tècnics de les dades

La base de dades QM9 disposa d'aproximadament 134.000 molècules orgàniques, les quals tenen entre 6 i 29 àtoms cadascuna, que poden ser C, H, O, N i F. En les figures 2 i 3 queda representada la distribució de les molècules respecte al nombre d'àtoms i els elements que les componen. Respecte a la distribució dels elements, mentre que els elements Carboni (C) i Hidrogen (H) apareixen en un 100% de les molècules i Oxigen (O) i Nitrogen (N) en un 80% i 60 % respectivament, el Fluor (F) apareix només en aproximadamente 3.000 molècules de les 134.000 totals.

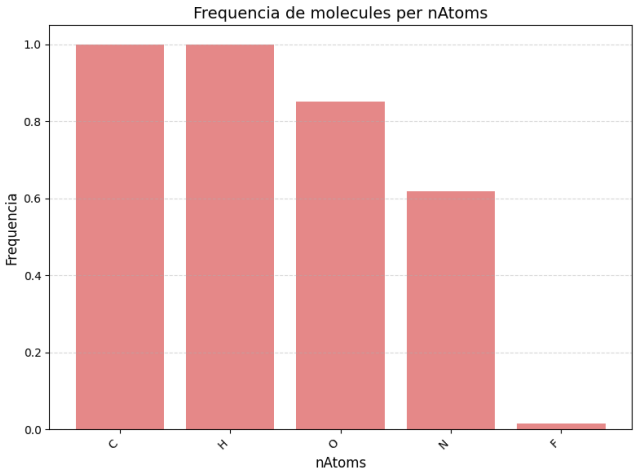

Figura 1: Histograma de la freqüència de molècules per element a la base de dades QM9. C i H apareixen a un 100% de les molècules i O i N un 80% i 60% respectivament, el F apareix només en 2,2% del total. *Gràfic d'autoria pròpia*.

En el cas de les dimensions de les molècules trobem una distribució normal on el número de molècules més freqüents són 17, 18 i 19 com es pot apreciar en la Figura 2 i 3.

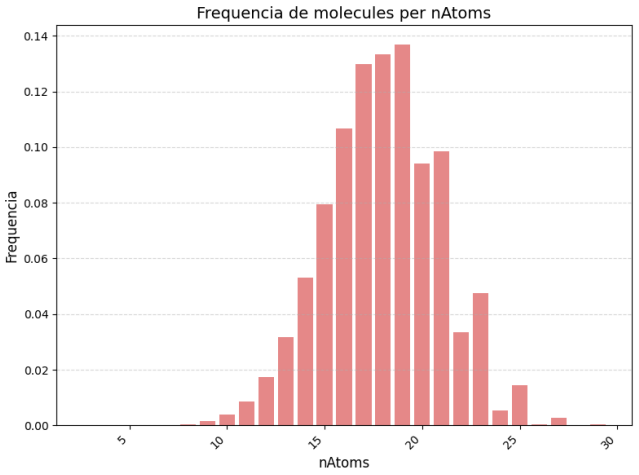

Figura 2: Histograma de la freqüència d'aparició per nombre d'àtoms a les molècules de la base de dades QM9. Les freqüències formen una distribució normal, amb un dèficit de molècules de més de 25 àtoms. *Gràfic d'autoria pròpia.*

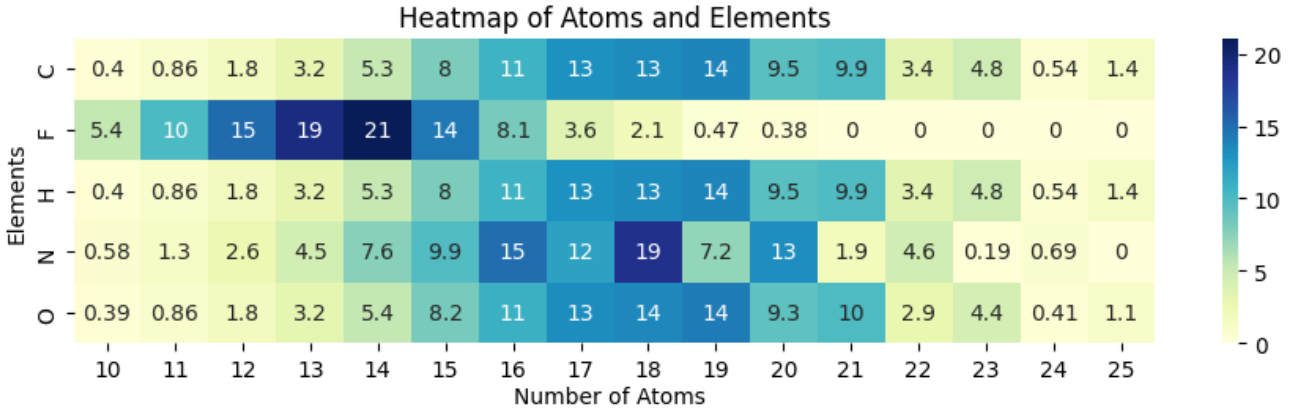

Figura 3: Heatmap de la distribució de les entrades de la base de dades QM9. En l'eix d'abscisses el nombre d'elements de les molècules i per a cada fila de l'heatmap la distribució de les molècules amb l'element corresponent. S'aprecia que les molècules amb Fluor present no segueixen la tendència general. *Gràfic d'autoria pròpia.*

Es pot concloure que la base de dades QM9 presenta alguns aspectes que poden afectar negativament l'entrenament de models de XN. En primer lloc, hi ha una desigualtat en la distribució dels elements, amb una major prevalença de Carboni i Hidrogen i una presència limitada de Fluor. Aquest desequilibri pot introduir biaixos en les prediccions per a molècules amb poca representació a la base de dades. A més, les dimensions de les molècules mostren una distribució normal, amb dimensions més freqüents que poden limitar la capacitat del model per generalitzar bé en altres dimensions. És important tenir en compte aquests factors negatius i reduir-ne l'impacte mitjançant un mostreig equilibrat de dades, tècniques de regularització i el disseny d'una arquitectura flexible.

3.2 Publicacions prèvies:

3.2.1 TorchMD:

TorchMD[3] és una eina avançada per a la simulació de DM que utilitza PyTorch com a base per a la implementació de models de XN. Amb una *data pipeline* flexible i modular, juntament amb les classes AtomisticModel i NeuralNetworkPotential, TorchMD permet tant l'ús de models predefinits com el desenvolupament de models personalitzats. A més, amb l'ajuda de PyTorch Lightning, se simplifiquen les tasques d'entrenament i avaluació dels models, proporcionant una interfície d'entrenament de nivell superior. Amb aquestes capacitats, TorchMD es converteix en una eina potent per a la investigació i l'avenç en l'àmbit de la descoberta de medicaments i materials.

3.2.2 SchNetPack:

SchNetPack 2.0[2] és una *toolbox* dissenyada per al desenvolupament i desplegament de XN per a simulacions de DM. Proporciona un *framework* flexible i modular per a la construcció de models complexos que poden predir diverses propietats de molècules, com ara forces d'interacció intermoleculars, energies fonamentals, entre d'altres. Les principals aportacions de SchNetPack 2.0 són: una *data pipeline* flexible, modularitat a l'hora de construir els models de XN, implementació de PyTorch per a DM, una interfície de comandes basada en Hydra per simplificar-ne l'ús i integració de PyTorch Lightning que permet gestionar i realitzar entrenaments fàcilment.

La *data pipeline* de SchNetPack 2.0, forma el marc de treball que permet processar i preparar les dades per als models de XN. Està composta per 2 components principals en forma de classe: ASEAtomsData, AtomsLoader.

- **ASEAtomsData** proporciona una interfície per carregar i manipular les dades. És un afegit a la interfície de la llibreria Atomic Simulation Environment (ASE)[10], proveïda per PyTorch. ASEAtomsData permet a l'usuari establir una sèrie de *preprocessing transforms* que s'apliquen a les dades individualment previ a què siguin agrupades i enviades als models. Aquestes operacions són importants per garantir que les dades estiguin en el format correcte i que puguin ser processades eficientment pel model. Algunes de les operacions més comunes inclouen el càlcul de llistes de veïns, l'eliminació d'*offsets* i el *casting* de propietats.

- **AtomsLoader** agrupa grans quantitats de dades per processar-les amb models. Concretament, AtomsLoader hereta una funcionalitat de Pytorch (DataLoader) per càrrega de dades, amb una funció *collate* que permet agrupar les dades de manera personalitzada per adaptar-se a les necessitats i aplicacions específiques. Permet carregar aquests lots de dades en paral·lel i així assolir un processament ràpid i eficient de grans conjunts de dades.

Amb SchNetPack, es pot treballar amb models predefinits o desenvolupar nous models personalitzats. Això ho fa possible l'ús de les classes AtomisticModel i NeuralNetworkPotential.

- **AtomisticModel** és la base de SchNetPack i hereta de la classe nn.Module de PyTorch. Utilitza mòduls predefinits i personalitzats per definir arquitectures de XN, aquests mòduls poden ser des de capes per a la representació de les dades a convolucions (llista

detallada dels mòduls disponibles en l'article[2]).

- **NeuralNetworkPotential** simplifica la creació de models MLP (Machine Learning Potentials). Aquesta subclasse aplica seqüencialment les funcions definides en AtomisticModel, amb l'afegit dels paràmetres *input_modules* i *output_modules*. El mètode *forward* és responsable de passar el diccionari d'inputs per les diferents funcions del model.

Una vegada els models han estat definits i les dades han estat processades, PyTorch Lightning entra en joc per simplificar el procés d'entrenament i avaluació dels models de SchNetPack. PyTorch Lightning proporciona una interfície d'entrenament de nivell superior que gestiona automàticament tasques com la configuració de l'entrenament, la gestió dels dispositius de càlcul, el càlcul dels gradients i l'actualització dels paràmetres del model.

4 METODOLOGIA:

Prenent de referència i simplificant la metodologia àgil utilitzada regularment en el camp del desenvolupament de software, se separarà el projecte en fases. En cada fase s'ha fet servir GitHub com a sistema de control de versions, podent establir cicles de treball de durada flexible (entre una i dues setmanes) per assegurar un seguiment adequat del progrés del projecte. Per a cada cicle, s'han predefinit objectius específics a assolir i s'han generat informes de cicle per verificar si s'han assolit tots els objectius desitjats i explicar les raons si no s'han assolit. També està contemplada la possibilitat de canviar l'ordre dels cicles de treball sempre que es justifiqui adequadament o no afecti negativament a altres tasques pendents. En finalitzar cada fase s'ha redactat un informe de progrés que recull els continguts dels informes dels cicles que componen la fase. Les fases es divideixen en dos grups: les dues primeres teòriques i la tercera pràctica.

- **Fase de formació**: En aquesta fase s'ha desenvolupat un coneixement profund dels fonaments teòrics i pràctics de les XN i la simulació de DM.
- **Fase d'exploració**: En aquesta fase s'han explorat els avantatges i les limitacions de les XN en la simulació de DM, comparant-les amb altres tècniques i abordatges existents.
- **Fase d'avaluació**: En aquesta fase s'ha usat la *toolbox* SchNetPack 2 amb l'objectiu de fer una anàlisi crítica del seu rendiment aplicant-la a la base de dades QM9.

4.1 Fase de formació:

Per a l'assoliment del primer objectiu, s'han fet servir diverses metodologies, incloent-hi tutories amb el tutor extern Jordi Faraudo (ICMAB) per al camp de la DM, i amb el tutor acadèmic Ramon Baldrich pel coneixement de XN; amb recerca pròpia a través de la lectura d'articles i altres fonts de documentació, així com l'aplicació d'assignatures relacionades com Aprenentatge Computacional (APC) per entendre els fonaments teòrics de les XN. A més, s'han realitzat reunions amb estudiants especialitzats en temes com Intel·ligència Artificial (IA) o les Matemàtiques Computacionals i Analítica de Dades (MatCAD) per obtenir una perspectiva diferent.

S'ha aconseguit desenvolupar un coneixement profund dels fonaments teòrics i pràctics de les XN i la simulació de DM. Tot plegat ha permès iniciar el treball del projecte amb més facilitat, així com entendre millor els objectius a aconseguir en les fases posteriors.

4.2 Fase d'exploració:

Per al desenvolupament d'una comprensió crítica dels avantatges i les limitacions de les XN en la simulació de DM, s'han dut a terme tutories amb el tutor extern Jordi Faraudo i lectures d'articles científics i treballs relacionats que comparaven les XN amb altres tècniques i abordatges existents. S'ha identificat que les XN tenen avantatges significatius en l'aplicació de tècniques de DM que involucren grans quantitats de dades i en l'extracció de característiques complexes. No obstant això, també s'ha detectat que presenten limitacions com la necessitat d'una gran quantitat de dades per al seu entrenament.

4.3 Fase d'avaluació:

La *pipeline* utilitzada per dur a terme l'última fase està representada en la ***Figura apèndix 1***, com mencionat anteriorment es tracta d'una fase pràctica i, per tant, els resultats d'aquesta es trobaran en l'apartat de resultats. Aquesta fase es divideix en 3 subfases:

- **Cerca d'hiperparàmetres**: per tal d'optimitzar els models proporcionats per la toolbox SchNetPack2 usant la base de dades QM9. Aquesta subfase s'ha dut a terme amb l'ajuda de Weights and Biases[4] (a partir d'ara WandB), una plataforma de monitoratge online que permet la fàcil visualització d'informació rellevant respecte l'entrenament, validació i resultats de models d'aprenentatge computacional, com ara corbes d'aprenentatge o consum de recursos. Especificament s'ha fet servir la funcionalitat ***sweep*** de WandB per a fer cerques d'hiperparàmetres. La primera etapa de l'optimització, s'ha centrat en la cerca d'hiperparàmetres estrictament computacionals com la taxa d'aprenentatge (learning rate), el nombre d'èpoques d'entrenament (epochs) i la mida del lot (batch size), amb l'objectiu de millorar el rendiment dels nostres models. Aquesta fase és fonamental, ja que els hiperparàmetres, com la taxa d'aprenentatge, el nombre d'èpoques d'entrenament i la mida del lot, tenen un impacte directe en el procés d'entrenament i poden afectar significativament el rendiment final del model. En la segona etapa, es canvia el focus de la cerca als hiperparàmetres interns de SchNetPack2, com la llista de veïns neighbor list i els paràmetres dels models de representació entre d'altres.
- **Anàlisi profunda dels resultats**: una vegada han quedat establerts els hiperparàmetres, s'han dut a terme una sèrie d'experiments amb l'objectiu de millorar els resultats dels models. Aquests experiments tenen tres fases: definició de la hipótesi, realització de tests i extracció de conclusions. S'han dut a terme 3 tests principals en els quals s'entra en més detall en l'apartat de resultats, els tests són: spliting i subsampling a la base

- **Test final**: com a ultim test s'ha entrenat un model tenint en compte les conclusions extretes dels anteriors experiments amb l'objectiu de comparar els resultats d'aquest model amb els resultats exposats en el paper de SchNetPack 2[2].

5 RESULTATS:

Abans de profunditzar en els resultats, és essencial establir alguns conceptes fonamentals. Tot i que en la toolbox SchNetPack2 s'hi tracten molts tipus de simulacions de DM, en aquest apartat es contemplaran exclusivament les simulacions de DM que busquen calcular l'energia de l'estat fonamental (U0) d'una molècula. L'atribut U0 indica l'energia total mínima d'una molècula en l'estat fonamental, és a dir l'estat d'energia més baixa que pot tenir el sistema. Aquesta energia es dona en electrons-volts (eV) o bé amb kcal/mol.

Per a realitzar una simulació de DM d'aquest tipus, és necessari definir un conjunt de condicions inicials, que inclouen les posicions (R) i el número atòmic (Z) dels àtoms del sistema. Aquestes condicions inicials es solen obtenir a partir d'una configuració estructural coneguda del sistema, com ara una estructura cristal·lina, o bé a partir d'una configuració aleatòria que segueixi les restriccions imposades per les interaccions del sistema.

Les posicions (R) són comunament representades per coordenades cartesianes, que són les més senzilles, consisteixen en la definició de la posició de cada àtom en un sistema de coordenades tridimensional, que es representa generalment en unitats de longitud com àngstroms (Å) o nanòmetres (nm). Així, una molècula es pot representar per la seva posició en l'espai, definida per les coordenades x, y i z de cada àtom que la conforma.

Tradicionalment, el càlcul de l'U0 es fa resolent les equacions de Schrödinger per aconseguir la funció d'ona electrònica i la seva energia associada[9]. Aquests mètodes són computacionalment costosos ja que han de tenir en compte la naturalesa quàntica dels electrons i el moviment dels nuclis atòmics. Això implica el càlcul de les funcions d'ona de tots els electrons de la molècula i, per tant, requereixen un gran nombre de càlculs matemàtics i computacionals. Com a resultat, aquests mètodes són limitats per la seva capacitat de resoldre problemes en molècules grans i complexes.

5.1 El Model:

Pel que fa al model usat, s'ha seguit un la guia subministrada en el git hub de SchNetPack2, s'ha escollit no modificar massa el model estàndard per poder comparar resultats amb que asseguren obtenir els creadors de la toolbox. Dit això en la ***Figura 4*** queda representat el model, a l'esquerra de la figura hi ha els mòduls pels quals passa un lot de dades (data batch) durant un pas de l'entrenament, la estructura es la següent:

- **Mòdul de representació**: ***SchNet*** a partir de les posicions (R) i els nombres atòmics (Z) i la llista de veïns (neighbour list) genera una llista de *features* que rep resenta la molècula, és un mòdul entrenable i per tant la representació es va millorant durant l'entrenament.

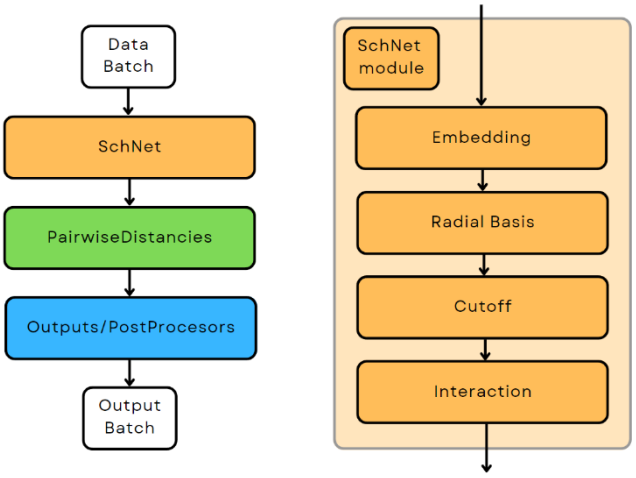

Figura 4: Diagrama dels mòduls del model usat. *Figura d'autoria pròpia.*

- **Mòdul predictor (input module)**: ***pairwiseDistancies.*** Es el mòdul responsable d'obtenir les prediccions de l'atribut objectiu a partir de les representacions realitzades per el mòdul de representació.
- **Mòdul de postprocessament:** s'ocupa de desfer les transformacions prèvies a l'entrenament**.**

A la dreta hi ha els mòduls interns del mòdul SchNet, siguen aquests, per més detalls de les capes internes d'aquest mòdul es pot trobar informació en la documentació original del treball SchNetPack v0[5], on s'explica el perquè de cada capa del model de representació.

5.2 Cerca d'hiperparàmetres:

Per aquest apartat s'ha decidit deixar els detalls fora de l'informe final i es troben en la secció d'apèndix A2, es proporciona una extensa explicació respecte al procés realitzat i les conclusions obtingudes. Seguidament es dona un resum d'aquest apèndix.

5.2.1 Hiperparametres Computacionals:

S'ha centrat en tres variables clau: batch_size, epochs i lr. El nombre d'èpoques (*epochs*) determina quantes vegades el model passa per tot el conjunt de dades d'entrenament, mentre que la mida del lot (*batch size*) especifica la quantitat de mostres que es processen en cada pas d'actualització dels pesos del model i per últim la taxa d'aprenentatge *learning rate* (lr) que indica la quantitat d'ajustament que es fa als pesos del model durant el procés d'entrenament.

Al final del procés de la cerca s'ha arribat a la següent conclusió: donada la situació que es compta amb temps limitat per fer l'estudi, s'escullen els valors que proporcionen major l'equilibri entre bons resultats i temps d'entrenament baixos.

- **Epochs**: 22
- **Batch_size**: 512
- **Lr**: 0.001

5.2.1 Hiperparàmetres Interns de SchNetPack2:

S'ha centrat en 3 paràmetres interns, *dataCutoff*, el número de veïns que té en compte a l'hora d'aplicar les transformacions prèvies a l'entrenament, trainingCutoff, el valor de Cutoff de la funció gaussiana utilitzada per representar les distàncies i n_atomBasis el nombre de *features* usat per representar la molècula.

Al final de la cerca s'ha arribat a la conclusió: els paràmetres escollits no han representat tanta variació en els resultats finals com s'esperava, tot i això s'han obtingut els següents paràmetres com als òptims:

- **DataCutoff**: 4
- **TrainingCutoff**: 5
- **N_atom_basis**: 38

5.3 Analisi profunda dels resultats:

Els resultats mostrats en aquest apartat s'han obtingut entrenant el model amb la base de dades QM9 default o modificada (indicat en cada resultat), i després validat el model amb la base de dades QM9 separada en *dataframes* per número d'àtoms de les molècules o separada per elements presents en les molècules. En el cas del número d'àtoms de la molècula s'ha decidit descartar els tamanys amb menys representació que un 0,1%, és a dir, s'utilitzaran per a la validació molècules d'entre 10 i 24 àtoms.

5.3.1 Baseline:

Abans d'entrar als tests individuals s'ha fet una *baseline* amb el model sense cap modificació per poder comparar els resultats dels experiments.

- **Hipòtesi 1:** com s'ha mencionat en l'apartat 2.3.1 es teoritza que mides de molècules poc representades en la base de dades donaran resultats pobres.
- **Hipòtesi 2:** seguint la hipòtesi 1, elements poc representats en la base de dades donaran resultats pobres

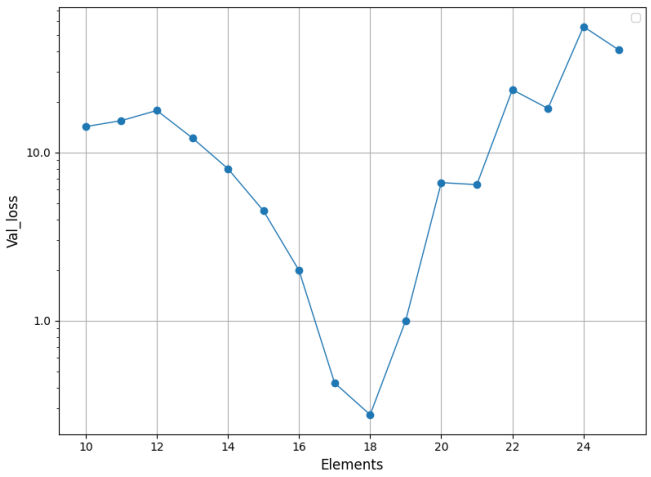

Figura 5: Model entrenat amb la base de dades QM9 default i validat amb els *dataframes* separats per número d'àtoms. L'escala de l'eix de les ordenades es logarítmic per visualitzar millor els resultats. Es pot veure la clara correlació inversa entre la freqüència d'aparició en la base de dades i la *validation loss*. **Confirma l'hipòtesi 1**

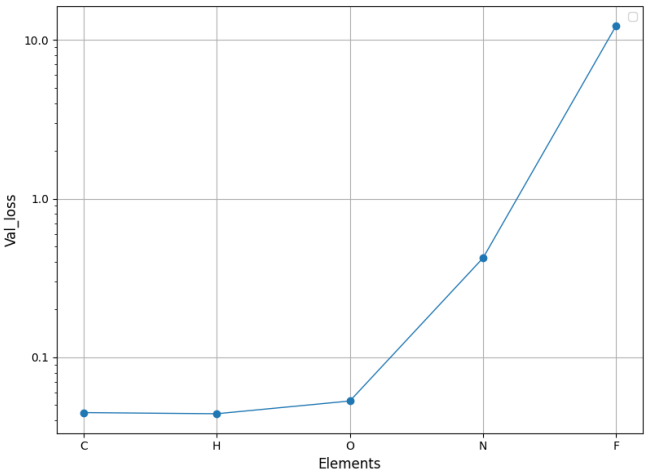

Figura 6: Model entrenat amb la base de dades QM9 default i validat amb els *dataframes* separats per elements. L'escala del eix de les ordenades es logarítmic per visualitzar millor els resultats. Es pot veure la clara correlació inversa entre la freqüència d'aparició en la base de dades dels elements i la *validation loss*. **Confirma l'hipòtesi 2.**

Es conclou que tant amb **la hipòtesi 1 i la hipòtesi 2 confirmades** hi ha molt marge de millora respecte al model base que queda completament esbiaixat a favor dels elements i mides de molècules més representats.

5.3.1 Database Subsampling:

La resposta més utilitzada en casos de bases de dades desequilibrades és el *subsampling/oversampling*, en el camp de la DM no és possible dur a terme *oversampling* així que s'han fet els tests amb *subsampling*.

- **Hipòtesi 3**: forçant que totes les mides de molècula tinguin la mateixa freqüència d'aparició s' equilibraran els resultats.

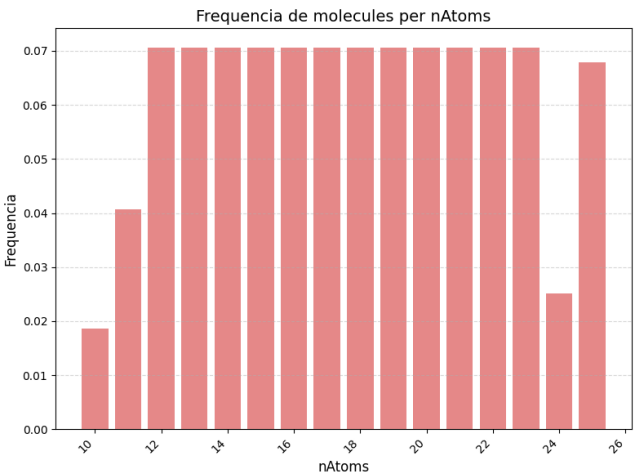

Figura 7: Histograma de la freqüència d'aparició per número d'àtoms en les molècules de la base de dades amb threshold de 2.000 molècules representades.

S'han fet 3 tests cada un amb un *threshold* de representació per número d'àtoms 2.000, 4.000 i 6.000 molècules.

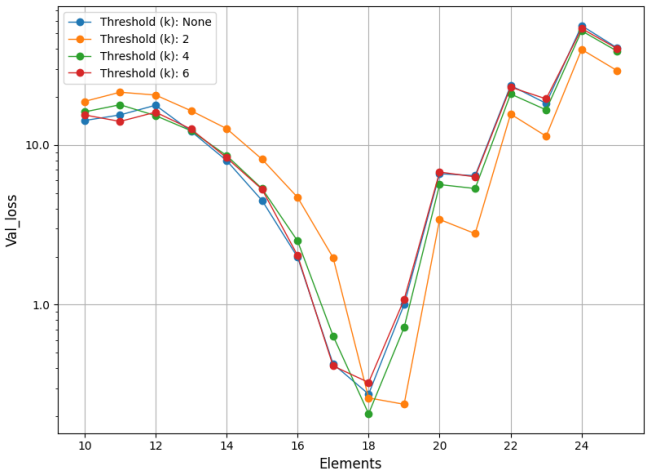

Figura 8: Models entrenats amb les bases de dades QM9 *subsampled* i validat amb els *dataframes* separats per número d'àtoms. L'escala del eix de les ordenades es logarítmic per visualitzar millor els resultats. Es pot visualitzar l'efecte del *subsampling* en els resultats en forma d'una millora per molècules més grans en el cas més allunyat de la base de dades *default* (Threshold de 2000).

Donat que l'efecte en els resultats no és l'esperat **la hipòtesi 3 queda descartada** el que porta a pensar que el principal causant del desequilibri en els resultats no és la representació en la base de dades.

5.3.1 Database Spliting:

A vista del poc impacte aconseguit amb el *subsampling* s'ha considerat necessari realitzar altres mètodes. El principi del mètode de *data base Spliting* és senzill, dividir la base de dades, entrenar amb una de les meitats i validar amb el total.

- **Hipòtesi 4**: els resultats obtinguts a partir de realitzar l'entrenament amb una meitat seran millors per a les molècules que pertanyin a aquesta meitat. Amb això quedarà demostrada la falta de capacitat de generalització del model.

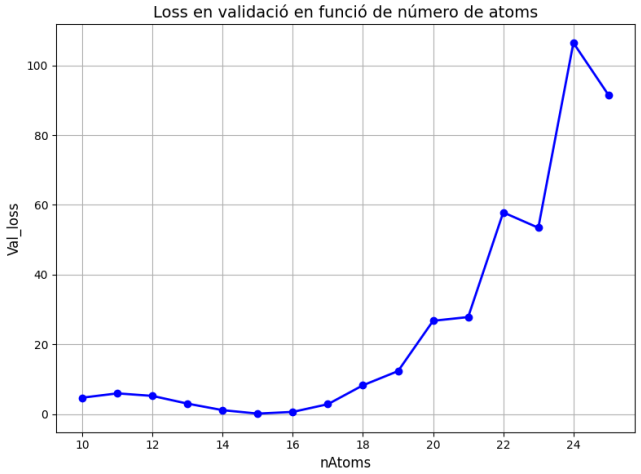

Figura 9: Models entrenats amb les bases de dades QM9 splited per valors inferiors a 18 número d'àtoms i validat amb els *dataframes* separats per número d'àtoms.

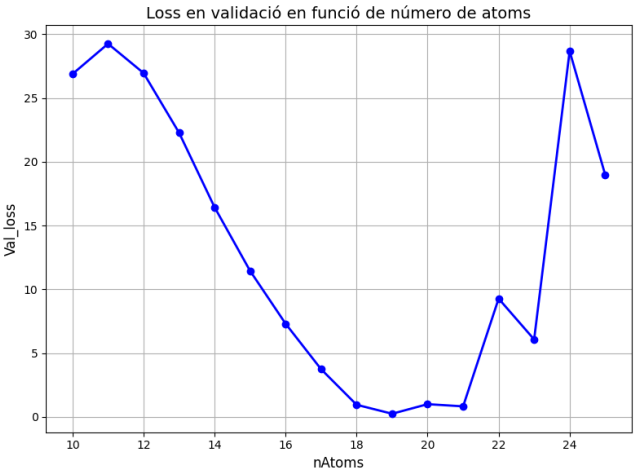

Figura 10: Models entrenats amb les bases de dades QM9 splited per valors superiors a 17 número d'àtoms i validat amb els *dataframes* separats per número d'àtoms.

Tant en la figura 9 com en la figura 10 s'aprecia el fenomen descrit en **la hipòtesi 4** que, per tant, queda **confirmada**. A part de les implicacions negatives mencionades en la hipòtesi que té aquest resultat també n'hi ha de positives, es redueix significativament l'error en molècules del rang d'àtoms amb el que s'ha entrenat el model, informació molt rellevant en cas de voler entrenar un model per a una mida de molècules específic.

5.3.4 Loss Rate Functions:

Un dels altres possibles mètodes per a compensar una base de dades desequilibrades es canviar l'impacte que tenen les dades en la funció del càlcul del loss rate durant l'entrenament que de base s'utilitza mean square error (MSE). S'han definit 2 posibles funcions de loss modificades, siguent **v** el factor per el qual es multiplica el valor de loss, **s** el numero d'atoms de la molecula, **f** la frequencia d'aparicio en la base de dades i **r** el rate d'impacte.

- **Size based**: amb l'objectiu de potenciar les molècules mes grans:

$$v = \frac{(s-7)^r}{10^{r+1}} + 1$$

- **Size and Frequency based**: busca un equilibri doentre el tamany i la frequencia:

$$v = log\left(\frac{\frac{s^r}{10^{r+1}}}{f} - \frac{r}{4.5}\right)$$

Per a visualitzar millor s'han realitzat les Figures Apendix 2 i 3 on queda representat l'evolucio del factor (v) per determinats números de rate (r).
Una vegada definides les funcions s'ha dut a terme un test entrenant amb la base de dades amb numero d'atoms entre 13 i 23 per obtener els millors rates per la funcio.

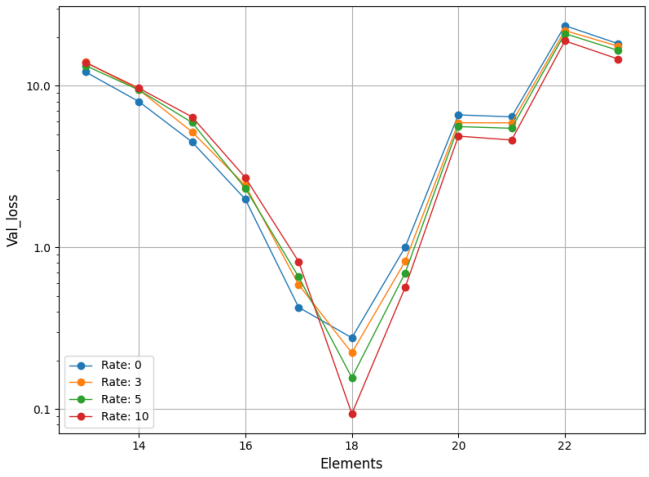

Figura 11: Models entrenats amb les bases de dades qm9 splited amb numero d'atoms entre 13 i 23 utilitzant la modificacio **Size Based** i validat amb els dataframes separats per numero de atoms. Per referencia la línea de rate = 0 representa la funcio default de loss del model.

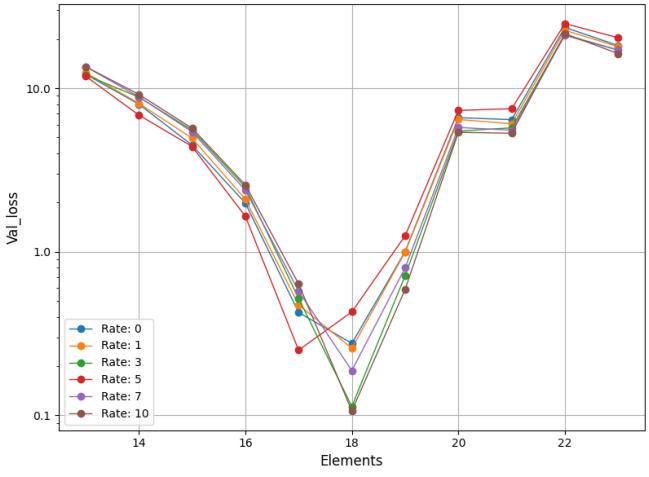

Figura 11: Models entrenats amb les bases de dades qm9 splited amb numero d'atoms entre 13 i 23 utilitzant la modificacio **Size and Frequency Based** i validat amb els dataframes separats per numero de atoms. Per referencia la línea de rate = 0 representa la funcio default de loss del model.

S'extreu de les figures 10 i 11 que els rates mes rellevants per minimitzar la loss en molècules amb numero d'atoms elevat son per ambdues funcions, el rate 10. Fet que te sentit considerant la naturalesa de les funcions.

5.3.4 Experiment final:

6 CONCLUSIÓ:

....

7 AGRAIMENTS:

....

8 BIBLIOGRAFIA:

[1] , Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum, and Jean-Louis Reymond, ***Journal of Chemical Information and Modeling*** *2012 52 (11), 2864-2875*

[2] **SchNetPack: A Deep Learning Toolbox For Atomistic Systems,** K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, and K.-R. Müller. ***Journal of chemical theory and computation***, 2019, 15, 448-455.

[3] **TorchMD: A Deep Learning Framework for Molecular Simulations**, Stefan Doerr, Maciej Majewski, AdriàPérez, Andreas Krämer, Cecilia Clementi, Frank Noe,Toni Giorgino, and Gianni De Fabritiis, ***Journal of chemical theory and computation***, 2021, 17, 2355−2363.

[4] **Weights & Biases Documentation**, (accedit el: 5/5/2023).

[5] **SchNet: A continuous-filter convolutional neural network for modeling quantum interactions**, Kristof T. Schütt, Pieter-Jan Kindermans, Huziel E. Sauceda, Stefan Chmiela, Alexandre Tkatchenko, Klaus-Robert Müller, ***arXiv***: 1706.08566v5

APÈNDIX

A1. FIGURES:

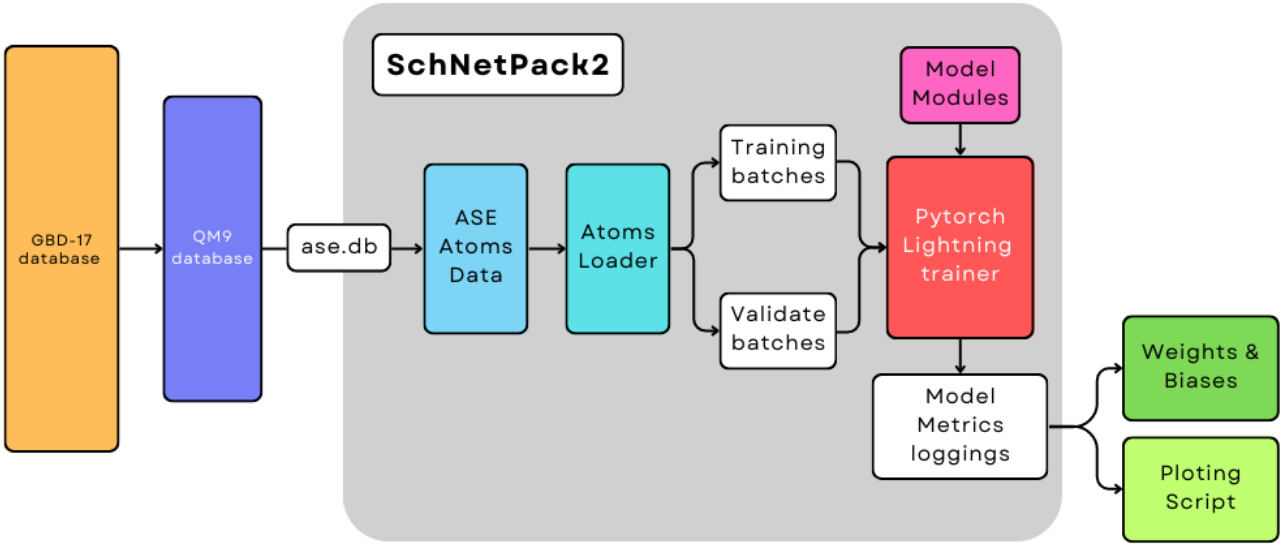

Figura Apendix 1:Diagrama de flux de la pipeline de la integracio de SchNetPack2 en el treball. *Figura d'autoria propia.*

A2. SECCIÓ D'APÈNDIX