

Tudományos Diákköri Konferencia

— ELTE Kémiai Intézet —

Szabó András

Reakciók minimum energia útvonalának optimalizálása gépi tanulás felhasználásával

Témavezetők:

Daru János és Mészáros Bence Balázs

ELTE TTK, Szerves Kémiai Tanszék



Eötvös Loránd

Tudományegyetem

Természettudományi Kar

Tartalomjegyzék

1. Bevezetés	2
1.1. Elektronszerkezet számítás módszerek	3
1.1.1. A Hartree-Fock (HF) módszer	3
1.1.2. DFT (Density Functional Theory)	5
1.1.3. TB (Tight Binding)	6
1.1.4. xTB (Extended Tight Binding)	6
1.2. Minimum energia út optimalizálására szolgáló módszerek	7
1.2.1. String módszerek	7
1.2.2. Nudged Elastic Band módszerek	7
1.3. Machine Learning módszerek	9
1.4. ML módszerek a PES közelítésére	11
1.5. ML-PES alkalmazása a geometria és MEP optimalizációs algoritmusokban	13
1.6. Célkitűzés	13
2. Módszerek és Eredmények	14
2.1. Módszerek	14
2.2. Alkalmazott programok és hardver	16
2.3. Eredmények	17
2.4. Összefoglalás	30
2.5. Köszönetnyilvánítás	30

1. fejezet

Bevezetés

A számítógépes kémia lehetővé teszi kémiai rendszerek modellezését és megértését az elméleti kémiai modellek hatékony számítógépes algoritmusainak felhasználásával. A modern tudományos kutatás szerves részévé vált, és számos tudományágban van gyakorlati alkalmazása. A gyakorlatban a numerikus matematikán, statisztikus fizikán és kvantumkémian alapuló numerikus megközelítéseket használják kémiai rendszerek viselkedésének szimulálására és előrejelzésére például szerves és szervetlen kémia, gyógyszerkutatás és anyagtudomány területeken.

A számítógépes kémia reaktív rendszerekkel foglalkozó területén a reakcióutak feltérképezése kulcsfontosságú és összetett feladat, melyben többek között a NEB (Nudged Elastic Band) módszer lehet segítségünkre. A eljárás jelentősége abban áll, hogy ez a triviálisan párhuzamosítható algoritmus nem csak az átmeneti állapotot, és annak tulajdonságait tudja meghatározni hanem a teljes minimum energia útvonalat (Minimum Energy Path, MEP) egyszerre optimalizálja. Hátulütője azonban a nagy számítási kapacitás igénye. Munkánkkal ezt a problémát kívánjuk orvosolni, a számítások gépi tanulással végzett gyorsításával.

Az utóbbi évtized új, gépi tanuláson (Machine Learning, ML) alapuló módszerei az elméleti kémiában is egyre nagyobb teret hódítanak. A koncepció lényege, hogy míg a hagyományos módszerek csupán az alkalmazott közelítések javításával konvergáltathatóak, a gépi tanulási algoritmusok tanulnak az általuk feldolgozott adatokból és hibákból. Így egy iteratív folyamat eredményeként biztosíthatnak egyre pontosabb eredményeket a feldolgozott adatokkal arányosan anélkül, hogy az újabb számítások költségét növelnék. Kutatásom során egy ML-NEB módszert fejlesztettem tovább, amely egy gépi

tanuláson alapuló segéd potenciális energia felület (potential energy surface, PES) segítségével gyorsítja fel a MEP felderítését. Az általam javasolt módszer egy költséghatékony erőter és a célul kitűzött PES különbségét közelíti ML regresszióval (Delta Learning [1]), ezáltal hatékonyabb, robosztusabb konvergenciát eredményez a NEB szimulációk során.

1.1. Elektronszerkezet számítás módszerek

Az elméleti kémia alappillére az atomok és azok egymással való kölcsönhatásának kvantumfizikai leírása a Schrödinger-egyenlet segítségével. Azonban az egyenlet túlságosan bonyolult ahhoz, hogy analitikusan megoldható legyen, illetve véges számítási kapacitások miatt a legtöbb esetben numerikusan sem megoldható, ezért közelítő módszereket kell használnunk. A közelítések első lépése általában a Born–Oppenheimer közelítés, melynek alapja, hogy az elektronokra felírt mozgás-egyenletekben az atommagokat álló helyzetűnek tekintjük jelentősen nagyobb tömegükből kifolyólag. Ez lehetővé teszi az elektronszerkezet jellemzésénél az atommagok és az elektronok szabadságfokainak szétválasztását. Mivel alapállapotú rendszerek esetén ez egy kiváló közelítés, a dolgozat során támaszkodni fogok a Born–Oppenheimer közelítésre.

1.1.1. A Hartree-Fock (HF) módszer

A HF az elektronszerkezet számítás egyik legalapvetőbb elméleti megközelítése, amely fontos eredményeket szolgáltat az elektronok viselkedésének leírására molekulákban [2]. A módszer fő célja az elektronok térbeli és energetikai elrendezésének meghatározása egy adott rendszerben, valamint a közelítő alapállapotú hullámfüggvény kiszámítása. A komplexitás csökkentése érdekében, a módszer felírása során azt is feltételezzük, hogy az elektronok egymástól függetlenül mozgó részecskék. A Pauli-elv következtében az N elektront tartalmazó rendszer hullámfüggvényét a

$$\Phi_{SD} = \frac{1}{\sqrt{N!}} \begin{pmatrix} \Phi_1(1) & \Phi_2(1) & \cdots & \Phi_N(1) \\ \Phi_1(2) & \Phi_2(2) & \cdots & \Phi_N(2) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_1(N) & \Phi_2(N) & \cdots & \Phi_N(N) \end{pmatrix}; \langle \Phi_i | \Phi_j \rangle = \delta_{ij}$$

Slater determinánssal közelítsük, ahol N a részecske számot, Φ_i pedig az i -edik molekula pályát jelöli.

A HF közelítés során az egzakt elektronokra vonatkozó nem-relativisztikus Hamilton-operátort alkalmazzuk, amely egy-elektron és két-elektron tagokra bontható. A h_i egy-elektron operátor gyűjti össze az elektron kinetikus energiáját, valamint a mag-elektronok közti Coulomb kölcsönhatást.

$$\mathbf{h}_i = -\frac{1}{2}\nabla_i^2 - \sum_A^{N_{mag}} \frac{Z_A}{|\mathbf{R}_A - \mathbf{r}_i|}$$

A második tag két elektron koordinátáitól függ, és az elektron-elektron taszítás leírásáért felelős.

$$\mathbf{g}_{ij} = \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}$$

Ezek alapján felírjuk a Hamilton-operátort, mely kiegészül még a mag-mag taszítással:

$$\mathbf{H}_e = \sum_i^{N_{el}} \mathbf{h}_i + \sum_{j>i}^{N_{el}} \mathbf{g}_{ij} + \mathbf{v}_{nn}$$

A fenti jelöléseket felhasználva az a HF teljes energia a következő alakban áll elő:

$$E = \sum_{i=1}^{N_{el}} \langle \phi_i(1) | \mathbf{h}_1 | \phi_i(1) \rangle \quad (1.1)$$

$$+ \sum_{i=1}^{N_{el}} \sum_{i>j}^{N_{el}} (\langle \phi_i(1)\phi_j(2) | \mathbf{g}_{12} | \phi_i(1)\phi_j(2) \rangle - \langle \phi_i(1)\phi_j(2) | \mathbf{g}_{12} | \phi_j(1)\phi_i(2) \rangle) + V_{nn} \quad (1.2)$$

A fenti összefüggés egyszerűbb alakra hozható, a J_{ij} és K_{ij} jelölések bevezetésével a Coulomb taszításra illetve a kicserélődési energiára. A szummázás szintén egyszerűsíthető J és K mátrók azonosságainak felhasználásával:

$$E = \sum_{i=1}^{N_{el}} h_i + \frac{1}{2} \sum_{i=1}^{N_{el}} \sum_{j=1}^{N_{el}} (J_{ij} - K_{ij}) + V_{nn}$$

A variációs elvből adódóan egy iteratív eljárásban azt a Slater-determinánst keressük, amely a teljes energiára adott fenti kifejezést minimalizálja. Az eljárás végén, megkapjuk az úgynevezett kanonikus pályákat és a hozzájuk tartozó ε_i pályae energiákat, melyek segítségével az energiára a következő képlet adódik.

$$E = \sum_i^{N_{el}} \epsilon_i - \frac{1}{2} \sum_{ij}^{N_{el}} (J_{ij} - K_{ji}) + V_{nn}$$

Fontos megjegyezni, hogy a Hartree-Fock módszer egy elsőrendű közelítés, amely nem veszi figyelembe az elektron-korrelációt. Ennek eredményeként a módszer annak ellenére, hogy a teljes energia 99%-át képes visszaadni, a legtöbbször nem elégséges a kémiai rendszerek értelmezéséhez. Az elektronkorrelációt figyelembe vételéhez más módszerekre lesz szükségünk. Bár ismert az adott bázisban értelmezett egzakt megoldást szolgáltató eljárás, a Full Configuration Interaction (Full CI) [3], de sajnos a módszer rendkívül költséges, használata kis méretű rendszerekre korlátozódik. A magas számítási igény annak köszönhető, hogy a Full CI az összes lehetséges elektronkonfigurációt (gerjesztett determinánst) figyelembe veszi, és a hullámfüggvényt ezek lineáris kombinációjaként közelíti.

A Full CI-t, illetve az ezzel ekvivalens Full CC-t (ahol CC a Coupled Clustert jelöli) a gyakorlatban a költségek csökkentése érdekében közelítjük, a megfelelő sorozatok véges levágásán keresztül. A CC módszer nem variációs, viszont a sorfejtés levágásával (pl. csupán az egyszeres, kétszeres) gerjesztések figyelembe vételével nem okoz méretkonzisztencia problémát a CI módszerrel ellentétben. Leggyakrabban a CCSD (Coupled Cluster Singles Doubles) illetve a CCSD(T) (Coupled Cluster Singles Doubles (Triples)) módszereket használják napjainkban.

1.1.2. DFT (Density Functional Theory)

A DFT [4] az elektronsűrűsége felírt funkcionált használja a hullámgömbök közelítésére és molekuláris tulajdonságok számítására. Ez egy másik megközelítés, ahol szintén csak egyetlen determinánst használnak, de a Hartree-Fock módszerrel ellentétben ebben az esetben ez nem a valós, hanem egy nem-kölcsönható segéd rendszer hullámfüggvénye. Ennek megfelelően a módszer a HF-hoz nagyon hasonló egyenletekre vezet, csupán egy külső, úgynevezett kicserélődési-korrelációs potenciálfüggvényben tér el. A DFT alapja a Hohenberg-Kohn tételek [5], amelyek kimondják, hogy az elektronok sűrűségfüggvénye meghatározza a potenciálfüggvényt és fordítva. Az "lokális" funkcionálok csak a helyi (lokális) elektronsűrűség-függvényre támaszkodnak, míg a nem lokális funkcionálok a teljes elektronsűrűség-függvényt használják. Bár a DFT módszerek ma már rutinszerűen alkalmazhatók nagyméretű rendszerekre is (100-200 atom), számos esetben ezeknek a számításoknak az elvégzése továbbra is időigényes, így gyorsabb számításokra van szükség. Ezekben az esetekben egy jó alternatíva lehet a TB (Tight

Binding) módszerek alkalmazása.

1.1.3. TB (Tight Binding)

egy olyan DFT alapú szemiempirikus módszer [6], amely minimális bázison operál. Ez a módszer-család a teljes rendszer energiáját atomok kölcsönhatásán keresztül írja le, és mindehhez egy közelítő Hamilton-operátort alkalmaz. A módszer előnye, hogy viszonylag költséghatékony, és alkalmazható nagyobb rendszerekre is, például makromolekulák, nanorészecskék és szilárdtestek elektronszerkezetének vizsgálatára. Azonban a TB módszer komoly közelítésekkel dolgozik, és a fizikai paraméterek becslésére és a kísérleti adatok illesztésére is támaszkodik.

1.1.4. xTB (Extended Tight Binding)

egy elektronszerkezet számítási módszer, amely az eredeti Tight Binding (TB) módszert bővíti ki [7]. A számolás során az eljárás az atomi töltések függvényében harmadrendű tagokat is alkalmaz. Ezen módszer nagy előnye, hogy csupán atomi paramétereket tartalmaz, azokból származtatja az atompárokra vonatkozóakat. Így parametrizálás általános, és elérhető a teljesen periodikus rendszerre. Az xTB számítási eljárásoknak több alfajtája van fejlettségüket tekintve GFN0-tól GFN2-ig. A legegyszerűbb és leggyorsabb módszer a GFN0-xTB nem önkonzisztens, míg a legfejlettebb GFN2-xTB módszer már iteratív és önkonzisztens eljárás, az s bázisfüggvényeken túl alkalmaz p bázisfüggvényeket is, valamint a fejlett D4 korrekciót alkalmazza a diszperzió leírására [8]. A GFN módszereket geometriákra, frekvenciákra és nemkötő kölcsönhatásokra optimalizálták, és nem energiákra. Ennek köszönhetően ezek a módszerek jobban képesek modellezni a kémiai kötéseket, a reakciókat és a molekuláris tulajdonságokat. Az xTB módszer gyors és hatékony, így alkalmazható nagyobb rendszerekre is és már kellően pontos ahhoz, hogy hasznos kémiai információkat nyújtson.

1.2. Minimum energia út optimalizálására szolgáló módszerek

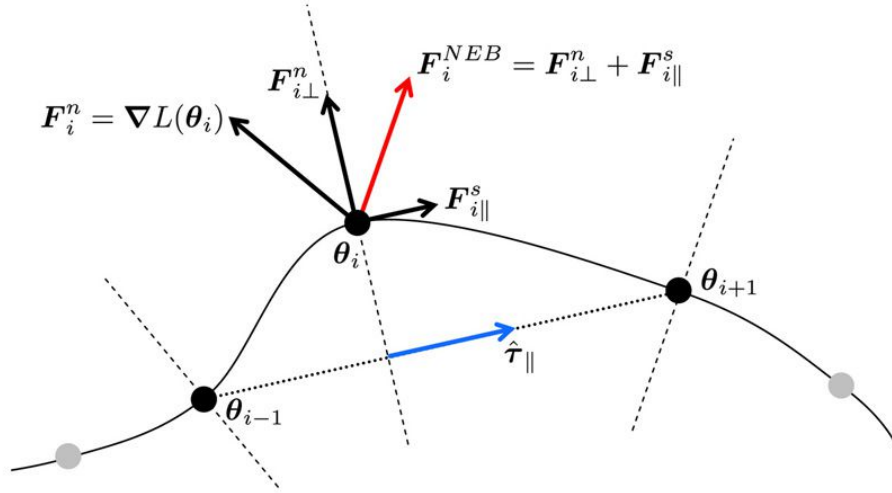
Az elméleti kémia reaktív rendszerekkel foglalkozó területén a reakcióutak feltérképezése rutinszerűen felmerülő feladat, melyben többek között a NEB (Nudged Elastic Band) módszer [9] lehet segítségünkre. A módszer egy útvonalat optimalizál a fázistérben, melynek pontjaihoz jól-definiált energiát rendel a potenciális energia felületet (PES) ami egy skalár mező az atomi koordináták függvényében. A kiindulási geometriák, az átmeneti állapotok és az esetleges intermedierek illetve reakciótermékek mind az energiafelületen azonosítható kritikus pontok, melyek azonosításából reakció dinamikai információk is kinyerhetők. Egy elemi reakció esetében két lokális minimum reprezentálja a kiindulási és a végtermék geometriáját. A kettő között a legideálisabb, azaz legalacsonyabb energiákon át fog vezetni a reakció „útvonal”, tehát a közöttük lévő legkisebb energiákhoz tartozó geometriák fogják reprezentálni a kémiai folyamatot. Ezt a trajektóriát hívjuk Minimum Energy Path (MEP, minimum energia útvonal)-nak, amelynek maximuma, az elemi lépést leíró PES nyeregpontja, reprezentálja az átmeneti állapotot. Bár a MEP megismerése kulcsfontosságú lenne a kémiai reakciókban, meghatározása korántsem egyszerű feladat, ezért számos módszert dolgoztak ki erre a feladatra. A legjelentősebbek közülük a string és NEB módszer családjához tartoznak.

1.2.1. String módszerek

A legegyszerűbb string módszer (magyar fordításban húr-módszer) a Zero Temperature String (ZTS) [10] esetén a MEP-et egy spline-al reprezentáljuk, melyet az egyenlő távolságokban található reakcióképek definiálnak. Growing String Method (GSM) [11], a ZTS-nek egy továbbgondolt változata ahol a kiindulási Stringbe újabb pontokat illesztünk-be a számunkra releváns helyen, általában a nyeregpont közelében, ezzel javítva a pontosságot. A módszert ugyan kisebb költségekkel lehet számolni, azonban a hátulütője, hogy nem paralellizálható megfelelően.

1.2.2. Nudged Elastic Band módszerek

A Nudged Elastic Band (NEB) [9], magyar fordításban „lökösött rugalmas szalag”, szintén diszkretizálja a MEP-et, de az egyes állapotokat harmonikus potenciálokkal köti össze. Az így kapott rugalmas sáv energiája (TNEB) meghatározható az összes kép energiája és harmonikus tagok összegeként.



1.1. ábra: Az erőkomponensek felbontása a NEB algoritmus során [13]

$$T_{NEB}(\mathbf{R}, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M, \mathbf{P}) = \sum_{i=1}^M E(\mathbf{x}_i) + \sum_{i=1}^{M-1} \frac{1}{2} k (\mathbf{x}_{i+1} - \mathbf{x}_i)^2$$

A string algoritmushoz hasonlóan ennek a konstrukciónak a célja az egyenletes eloszlás biztosítása a teljes útvonalon. Ezen T_{NEB} energia minimalizálása pedig “kirajzolja” a legkedvezőbb útvonalat a PES-en. Az optimalizáció során azonban problémák merülhetnek fel. Például, ha a rugóállandó túl kicsi, a rendszer a minimumok felé “csúszik”, és a nyeregpontnál nagy hibát fog adni, míg ha túl erős, akkor hajlamos a MEP-től eltérni és a PES magasabb energiájú pontjain áthaladni. Megoldást jelenthet, ha sok képet használunk fel, illetve az úgynevezett IT-NEB módosítás. Az Improved Tangent NEB (IT-NEB) [12] egy továbbfejlesztett változata a Nudged Elastic Band (NEB) módszernek. Az IT-NEB célja a a korábbi NEB változatok során tapasztalt numerikus instabilitások és szingularitások kezelése. Az algoritmus a képek közötti elmozdulásvektorok kiszámítására összpontosít, és a rugóerőknek csak az út érintőjével párhuzamos, a rendszer valós gradiensének pedig az arra merőleges komponensét veszi figyelembe, így elérhető, hogy a rugók egyedül a lánc menti eloszlásra hassanak. Mivel a reakcióutat diszkrét képek halmaza reprezentálja, így az utat egy adott kép környezetében a szomszédos képekből kell megbecsülni. A NEB algoritmus az adott pontbeli érintőt a két szomszédos képének különbség-vektoraként határozza meg. A Climbing Image (CI-NEB: „mászó kép”) [14] algoritmusnál a

legmagasabb energiájú képnek megengedjük, hogy elmozduljon a többihez képest a NEB-lánc mentén, azzal a céllal, hogy tökéletesen a nyeregpontra legyen. A gyakorlatban ez a párhuzamos erőkomponens invertálásával érhető el, mely mindig felfelé “húzza” majd a kiválasztott képet. Ez azt jelenti, hogy a nyeregponthoz közelebbi képeket kapunk javítva a módszer pontosságát az átmeneti állapot közelében. Azonban ezzel plusz számítási terhet is eredményez, mellyel lassítja a konvergenciát. Az Energy-Weighted NEB (Energia-súlyozott NEB) [14, 15] az egyes pontokat súlyozza azok energiájának függvényében, ellentétben az eredeti módszerrel ahol a pontokat azonos mértékben vesszük figyelembe. Így a magasabb energiájú területek nagyobb arányban szerepelnek a számításban. Ezzel segítve a nyeregpontra környékének feltérképezését és pontosítja az átmeneti állapot leírását anélkül, hogy újabb erőforrásokat vonna be a számításba, ellenben a CI-NEB módszerrel.

1.3. Machine Learning módszerek

Az utóbbi évtizedben jelentős teret és figyelmet nyert a számítástechnikában a gépi tanulás (ML), mely egy olyan számítási módszer, amely lehetővé teszi a számítógépek számára, hogy adatokból tanuljanak és önállóan döntéseket hozzanak vagy előrejelzéseket készítsenek. Mindezek mellett a tanulási folyamat után a program futási költségei igen alacsonyak és a becslt értékek pontosak. Általánosságban az ML folyamat a következő lépésekből áll:

1. Adatgyűjtés: Először adatokat gyűjtünk a problémával kapcsolatban. Ezek lehetnek strukturált adatok, táblázatok vagy adatbázisok formájában, de akár képek, hangok vagy szövegek is.
2. Adatok előkészítése: Az adatokat előkészítjük a további elemzésekhez. Ez magában foglalhatja a hiányzó adatok pótlását, a zaj eltávolítását, az adatok normalizálását vagy kódolását megfelelő formátumra.
3. Modellválasztás és tanítás: Kiválasztunk egy megfelelő ML modellt a probléma jellegétől függően. Ez lehet például neurális hálózat, döntési fa, klaszterezési algoritmus stb. Ezután a modellt tanítjuk a rendelkezésre álló adatokkal, ahol a modell beállítja a belső paramétereit, hogy megtalálja a mintázatokat és összefüggéseket az adatokban.
4. Modell kiértékelése: A modell hatékonyságát kiértékeljük egy tesztadat készleten. Mérőszámokat

használunk, mint például a pontosság, hogy értékeljük a modell teljesítményét.

5. Modell finomhangolása: Ha a modell nem megfelelő teljesítményt nyújt, tovább finomhangolhatjuk a paramétereit vagy kipróbálhatunk más modell-architektúrákat.
6. Előrejelzés: Ha elégedettek vagyunk a modell teljesítményével, alkalmazhatjuk a modellt új adatokra, hogy előrejelzéseket készítsünk vagy döntéseket hozzunk az adott probléma alapján.

Napjainkban az egyik leggyakrabban alkalmazott módszercsalád ML regresszióra a neurális hálózatok (Neural Networks, NN). Ezek olyan gépi tanulás modellek, melyeket az emberi agy idegrendszere inspirált. Ezek a modellek több egymással összekapcsolt egységből, ún. neuronokból épülnek fel, amelyek információt dolgoznak fel és továbbítanak. A neurális hálózatoknak gyakran több, egymás után következő rétege van. Az első réteg a bemeneti réteg, amely fogadja az adatokat, az utolsó pedig a kimeneti réteg, amely a végeredményt szolgáltatja. A rétegek között találhatók ún. rejtett rétegek, melyekben a neuronok megtanulják a bemeneti adatok közötti összefüggéseket és mintázatokat. A neuronok a bemeneti adatokat súlyokkal és aktivációs függvényekkel kombinálják, majd a kimenetüket továbbítják a következő rétegbe. A neurális hálózatokban az információ előre halad a rétegeken keresztül. A tanítási folyamat során úgy optimalizálják a súlyokat, hogy a kívánt eredményt adja a modell. Ezen modellek lehetővé teszik komplex összefüggések tanulását és az adatokban rejlő mintázatok alapján való általánosítást is. A neurális hálózatok sokféle feladatban alkalmazhatók, beleértve a képfelismerést, beszéd felismerést, természetes nyelvfeldolgozást, a data mining-et, előrejelzést, vagy akár atomi rendszerek PES-ének reprezentációját is.

Egy másik megközelítés a Gaussian Process Regression (GPR), amelyik egy statisztikai gépi tanulás módszer. A GPR bayesi megközelítésen alapul, amely a kernel trükk segítségével vezeti vissza a nem-lineáris illesztési problémát egy (regularizált ez a szó kimaradhat) lineáris regresszióra. A GPR a kívánt függő változó és a hozzá tartozó bemeneti változók közötti összefüggések modellezésére szolgál.

$$y(\mathbf{x}) = \sum_{m=1}^M c_m k(\mathbf{x}, \mathbf{x}_m)$$

A tanulás folyamán az eljárás a c_m -ek segítségével illeszti meg az adathalmazt, majd készíti el azt a függvényt, ami reprezentálja a bemeneti és kívánt adatok közötti összefüggést. Egy $\mathbf{y}$ adathalmaz

esetén a $\mathbf{c}$ vektor Σ diagonális regularizáció mellett következő alakban adódik:

$$\mathbf{c} = [\mathbf{K}_{NN} + \Sigma]^{-1} \mathbf{y}$$

. A módszer alapvetően egy valószínűségi eloszlást rendel a bemeneti változókhoz, és az adatpontok közötti korrelációt kernel függvényekkel reprezentálja. Bár többféle kernel ismeretes, a leggyakrabban alkalmazott a Gauss-függvény:

$$k(\mathbf{x}, \mathbf{x}_m) = \exp \left(-\frac{|\mathbf{x} - \mathbf{x}_m|^2}{2\sigma^2} \right)$$

A fenti kernel függvény definíciójában x és x_m bemeneti adatok, míg egy fontos paraméter a σ , ami az illesztett függvény szélességét befolyásolja. A predikció során a fentebbi kernel függvények lineáris kombinációjaként közelítjük a kívánt függvényt. A GPR előnye, hogy az bayesi alapú, így képes a predikciókon túl azoknak bizonytalanságát is analitikusan megadni. Ez hasznos pl. olyan adaptív algoritmusok esetén, ahol az új adatpontoknak a predikciók megbízhatósága alapján mintavételezzük.

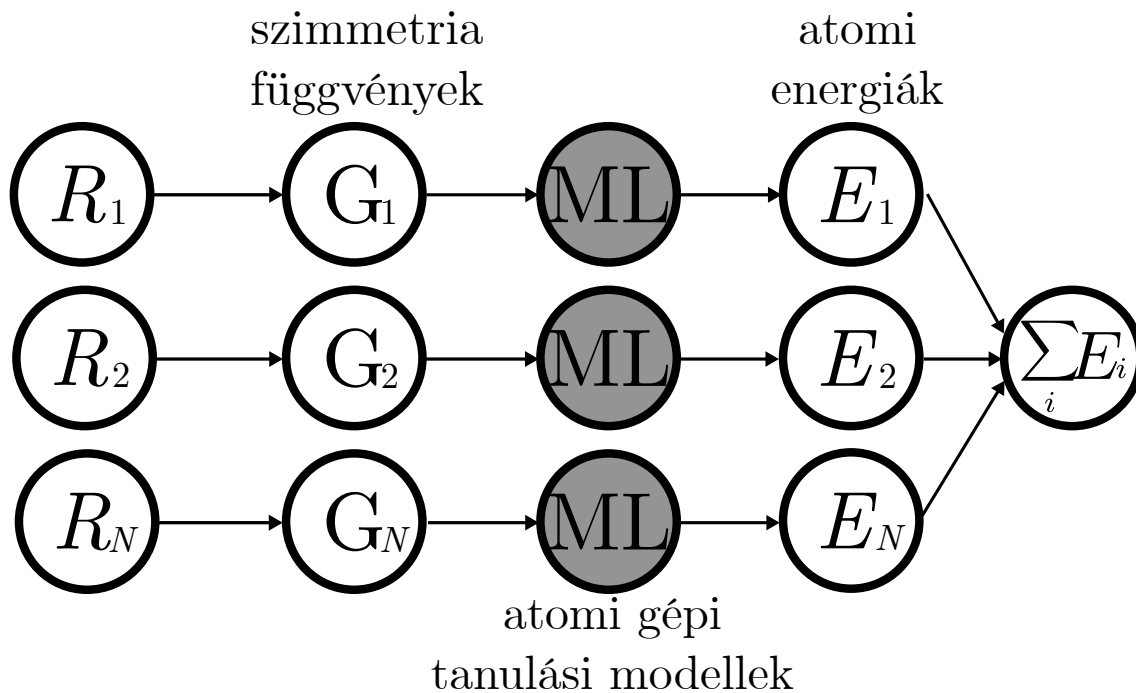
1.4. ML módszerek a PES közelítésére

A Behler–Parrinello (BP) [16] egy olyan ML módszer, amely neurális hálókat használ fel az atomi energia megtanuláshoz. Bemeneti információként ez a modell atomi szimmetria függvényeket alkalmaz, amely letapogatja az atomok környezetét olyan függvények segítségével, amelyek az adott atom és szomszédjai távolságaitól, valamint két szomszédos atomhoz mutató vektor által bezárt szögtől függ. A teljes rendszer energiája az atomi energiák összegeként adódik (lásd a 1.2 ábrát).

A modell tanítás után alkalmazható az új konfigurációk potenciális energiájának pontos és gyors előrejelzésére, amely lehetővé teszi akár kiterjedt rendszerek dinamikájának nagy pontosságú szimulációját [17] és a különböző tulajdonságok, például a szerkezetek, az energia vagy a reakcióutak vizsgálatát.

A Gaussian Approximation Potentials (GAP) [18] a Behler–Parrinello módszerrel analóg módon a rendszer energiáját atomi energiák összegeként közelíti, azonban neurális hálók helyett GPR gépi tanulás módszert alkalmaz. A GPR tulajdonságainak köszönhetően kevesebb adatponttal is tanítható modell, amelyek esetén adaptív tanulási algoritmusok alkalmazása is könnyen megvalósítható.

A Direct Machine Learning (Direct-ML) módszernél a bemeneti adatokhoz, például molekulageometriákhoz rendeljük az általuk adott PES-t. Ehhez előbb a molekula geometriákhoz tartozó poten-



1.2. ábra: Az atomisztikus gépi tanulási módszerek működési sémája

ciális energiákat egy referencia módszer segítségével meghatározzuk. Majd egy ML módszer alkalmazásával tanuljuk a bemeneti és a kapott adatok (PES) közötti összefüggést. Ez alapján kívánjuk a Direct-ML-lel közvetlenül meghatározni az újabb geometriához tartozó energia felületeket.

A Delta Machine Learning (Delta-ML) [1] egy összetett tanulási módszer, melynek során két adathalmaz közötti különbséget tanuljuk meg a bemeneti adatok függvényében. A számítógépes kémiában ezt a módszert arra tudjuk felhasználni, hogy egy pontos és egy közelítő, de gyorsabban számolható módszer közötti különbséget megtanuljunk, így később az ML egy korrekcióként használható a gyorsan számolható módszer javítására. Amennyiben a közelítő módszerünk a konstans függvény, a Delta-ML visszaadja a Direct-ML eljárást.

Az eljárás során kiszámítjuk a két módszer által szolgáltatott energiák és erők közötti különbséget (deltát) a kívánt konfigurációkon. Ezután használjuk a gépi tanulási algoritmusokat (például: GPR), hogy megtanuljuk a delta függvényét a bemeneti adatok (atomi ujjlenyomatok) függvényében. A predikció során pedig a költséghatékony módon „on-the-fly” számoljuk, és hozzáadjuk az illesztett korrekciós tagot. Majd az így kapott eredő PES-t alkalmazzuk a későbbi szimulációk során, amely

így gyorsabb és költséghatékonyabb, mint a magasabb számításigényű referencia számítás. A módszer különösen előnyös azokban az esetekben, amikor magas költséggel járnak a referencia számítások, de elérhető egy gyorsabb, közelítő módszer, hiszen általában kevesebb referencia számítást igényel, mint a Direct-ML. Cserébe viszont költségesebb lesz a predikció, mivel a közelítő módszert és az ML predikciót is ki kell számítani a kérdéses konfigurációkra.

1.5. ML-PES alkalmazása a geometria és MEP optimalizációs algoritmusokban

Bár a ML-PES szimulációk többségében a szimulációkat egy már konvergáltatott felületen végzik, vannak irodalmi példák arra is, amikor egy kezdetleges felületet alkalmaznak arra, hogy egy a konfigurációs térben folytatott optimalizációs feladatot felgyorsítsanak. Ezen munkákban iteratíván fejlesztik a felületet az optimumhoz közeledve és a rendszer koordinátáit illetve a közelítő PES szimultán optimalizálódik. A fenti megközelítést alkalmazták már geometria optimalizációra [19, 20] és az átmeneti állapotok lokalizációjára [21, 22], valamint NEB szimulációk felgyorsítására is [23, 24]

1.6. Célkitűzés

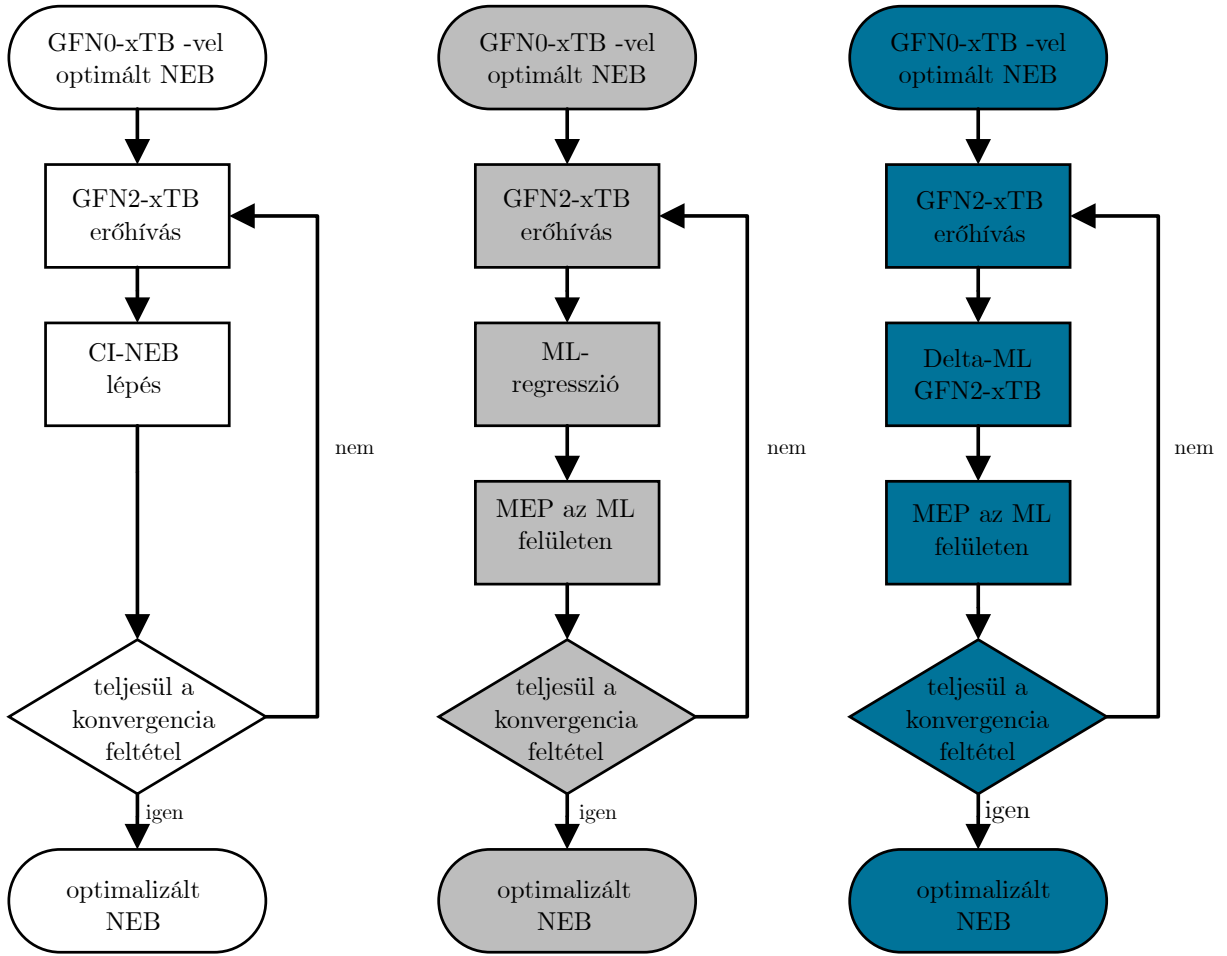
Diákköri kutatásom során együttműködő partnereink, Andreas W. Hauser és munkatársai által kifejlesztett Direkt-ML-en alapuló NEB algoritmus [24] tesztelését tűztük ki realisztikus kémiai reakciókon, valamint a módszer robusztusságának javítása érdekében célul tűztük ki egy Delta-ML alapú variáns kifejlesztését. Célunk, hogy egy általánosan alkalmazható NEB algoritmust fejlesszünk ki, amely képes a rutin elméleti kémiai számítások során a reakció utak optimalizációjára költséges (pl. DFT, MP2) szinten, a tradicionális NEB algoritmusok számításigényének töredékének felhasználása mellett.

2. fejezet

Módszerek és Eredmények

2.1. Módszerek

Diákköri kutatásom során a Meyer et al. által kifejlesztett Direct-ML-en alapuló ML-NEB módszert [24], valamint az általunk kifejlesztett Delta-ML-NEB algoritmust teszteltük tradicionális NEB algoritmussal összehasonlításban azonos reakciókból álló adatbázison. A példareakciókat irodalmi adatbázisokból [15, 25] választottuk ki véletlenszerűen (lásd a 2.5 összesítő ábrát). A mi Delta-ML algoritmusunk abban különbözik az eredeti algoritmustól, hogy nem közvetlenül illeszti meg a referencia módszer energiáját és erőit ML regresszióval, hanem egy kis költségű módszer hibáját illeszti meg a cél metodológiához képest. A két eltérő költségű elméleti szintnek számos módszert választhatunk, például szemimpiikus és DFT, GGA és Hybrid funkcionálok, vagy MP2 és CCSD(T). Mi a módszer paramétereinek optimalizálása érdekében két alacsony költségű szemimpiikus módszert, a GFN0-xTB-t és a GFN2-xTB-t alkalmaztunk. A három eljárást (GFN2-xTB alapú tradicionális NEB (továbbiakban GFN2-xTB-vel jelölve), valamint a Direct-ML és a Delta-ML NEB algoritmus) a 2.1. ábra szemlélteti.



2.1. ábra: A három vizsgált algoritmus folyamatábrája. A fekete, szürke és kék ábrák a GFN2-xTB, Direkt-ML és Delta-ML algoritmusokat jelölik.

Az összehasonlítás során GFN0-xTB segítségével generáljunk a kiindulási NEB-et, mivel a gyakori eljárás az, hogy a NEB számításokat egy alacsonyabb szintű erőtérral konvergáltatott NEB-ből indítják. Ezt követően a három algoritmust konvergenciáig futtatjuk, vagy addig amíg az eljárás valamilyen hiba miatt le nem állt. Vizsgálatunk során a gépi tanulás nélkül tisztán GFN2-xTB erőkön alapuló NEB számítást tekintettük referenciának, és a másik két algoritmust, ezek eredményeihez hasonlítottuk.

A Direct és Delta ML eljárások során a GAP módszert alkalmaztuk, Artrith és Kolpak féle szimmetriafüggvények [26] felhasználásával. Ahogy a 2.1. ábra szemlélteti, az ML alapú NEB algoritmusok

esetén a FIRE NEB-relaxációs lépés [27] helyett a rendelkezésre álló referencia számításokat felhasználva egy ML-PES-t illesztettünk, majd ezen a felületen konvergáltattuk a NEB algoritmust. Ennek megfelelően minden GFN2-xTB számításra egy konvergáltatott ML alapú NEB számítás jut.

Összehasonlításunkban a számítások során végzett erőhívások számát követtük figyelemmel, mivel ezek mennyisége arányos az eljárás hatékonyságával és futási idejével, abban az esetben, ha a gépi tanulás költsége elhanyagolható az erőszámításokhoz képest. Továbbá a futási idővel szemben nem függenek a számítógép hardverétől és egyéb külső körülményektől. Ezenfelül még hosszabb számítási igényű módszerekkel való összehasonlításra is enged következtetni. Továbbá feltüntettük az egyes molekulák szabadsági fokainak számát is, ami tájékoztatást nyújt a vizsgált rendszer komplexitására.

2.2. Alkalmazott programok és hardver

Bár az erőszámítások számát alkalmazzuk az összehasonlítás során, a teljesség kedvéért közöljük az általunk használt számítógép specifikációját. Számításaink egy node-on futottak, amelyek 18 magos Intel Xeon Gold 6240 processzorral és 90 GB RAM-mal rendelkeznek. Számításokhoz az Atomic Simulation Environment (ASE)-t használtunk keretrendszerként, melyben a GFN-xTB és az mlpot kalkulátorokat használtuk. A futtatásnál a GFN0-xTB és a GFN2-xTB esetén is a maximális iterációszámot 200-ra míg az pontosságot 0,0001-re állítottuk. A reakció dinamikai szimulációk futtatásához a FIRE-nevű algoritmust használtuk [27]. A FIRE egy NEB optimáló algoritmus [27], ahol az eljárás lényege, hogy a NEB következő lépését a NEB elmozdulásának sebessége alapján határozza meg. Azonban nem vesz minden irányú sebességet azonos súllyal figyelembe. Az algoritmus lényegét az alábbi egyenlet adja, ahol P a rendszer teljesítménye, $\mathbf{F}$ a NEB lánc adott pontjában ható erő, és $\mathbf{v}$ a NEB egy adott pontjának az elmozdulása.

$$P = \mathbf{F} \cdot \mathbf{v}$$

Ezen számítási eljárás miatt bizonyos esetekben megeshet, hogy a NEB nem konvergál, hanem oszcillál egy adott végállapot körül. Ez akkor fordulhat elő, ha a PES nem elég pontos, vagy egyenletlen.

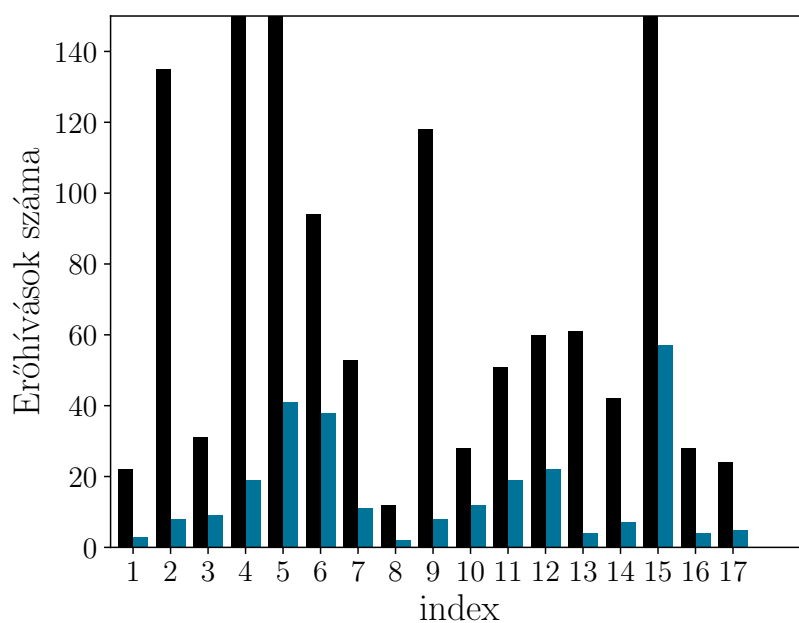
2.3. Eredmények

A GFN2-xTB, direkt-ML, valamint delta-ML módszereket alkalmazó NEB futások erőhívásainak számát összehasonlítva képet kaphatunk azoknak számítási költségéről. Az egyes reakciókra kapott eredmények összefoglalását az 2.1. táblázatban láthatjuk szabadsági fokuk szerint sorba rendezve, mely mennyiséggel a reakció komplexitását reprezentáljuk. Látható, hogy a delta-ML legtöbb esetben kevesebb erőhívásból konvergált, mint a direct-ML. Habár szembetűnő, hogy a direct-ML sok esetben abortál konvergencia feltétel előtt, mégis hamarabb ad megoldást a 15-ös reakció esetén. A 4-es, 5-ös és 15-es reakcióknál az „on-the-fly” GFN2-xTB NEB nem érte el a konvergencia feltételt és a minimum körül oszcillált. Érdekes, hogy ezekben az esetekben nem csak a Delta-ML hanem a Direct-ML is konvergált. Ez azzal magyarázható, hogy a folytonos ML felületeken történő optimalizáció átsegíti a rendszert a fázistér olyan pontjain, ahol a GFN2-xTB SCF konvergencia problémákat mutat. A GFN2-xTB és a Delta-ML erőhívásainak számát vizuálisan is összehasonlítottuk az 1. ábrán. Jól látható, hogy minden esetben jelentősen csökkent a szükséges erőhívások száma. A mindhárom esetben lefutott számítások közül a 2-es reakció esetén legkirívóbb a módszer gyorsaságának növekedése Delta-ML alkalmazásával: a GFN2-xTB módszerhez képest 16-szor kevesebb erőhívásra volt szükség. Kiemelendő továbbá a 9-es, 13-as, 14-es és a 16-os reakció is, ahol szintén remek teljesítményt figyelhetünk meg. Átlagosan az erőhívások száma a delta-ML módszerben 22,8%-a volt hagyományos NEB-ének, ami több, mint négyszeres megtakarítást jelent.

A Direct-ML és a Delta-ML eredmények összehasonlítása esetén látványos, hogy az általunk javasolt módszer jóval robosztusabb. Amint a 2-es ábrán látható, hogy a direkt-ML csupán 29,5%-ban fut le eredményesen, addig a delta-ML remek arányban, 94,1%-ban. A megnövekedett stabilitás arra vezethető vissza, hogy az „on-the-fly” számolt alacsony költségű felület megakadályozza a kémiaiailag irreleváns konfigurációk meglátogatását, ami konvergencia problémákat eredményez az új referenciapontok számítása esetén. Továbbá a Delta-ML-hez általában kevesebb referenciapont szükséges, illetve azonos számú referenciapont esetén pontosabb PES-t eredményez. A NEB konvergencia-problémája egyes esetekben nem az elektronszerkezet számítási problémához köthető, hanem magához a NEB algoritmushoz. A köztes adatokból az látszik, hogy a NEB a minimum körül oszcillál, azonban a PES nem elég pontos, hogy a minimumba jussunk. Ezzel a továbbfejlesztett módszerrel a vizsgált rendszereken háromszor gyakrabban ad eredményt, mint az eredeti Direct-ML algoritmus. Ezen felül a Delta-ML

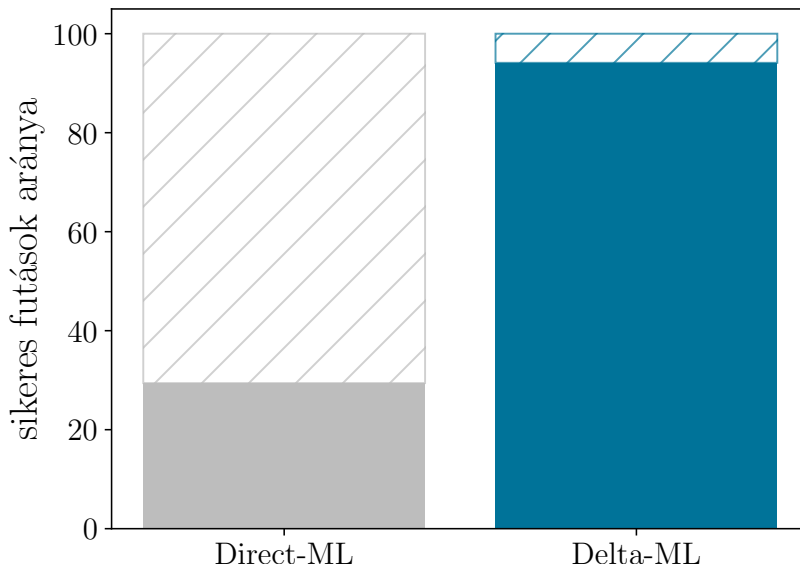
reakció sorszáma	szabadsági fokok száma	GFN2-xTB	direct-ML	delta-ML
1	3	22	$(34)^B$	3
2	6	135	25	8
3	9	31	53	9
4	15	$(12823)^A$	37	19
5	24	$(77317)^A$	$(100)^A$	41
6	24	94	$(41)^A$	38
7	24	53	$(66)^A$	11
8	24	12	$(4)^B$	2
9	27	118	$(4)^B$	8
10	27	28	$(100)^A$	12
11	36	53	$(0)^B$	19
12	36	60	$(0)^B$	22
13	42	61	$(5)^B$	$(4)^B$
14	42	42	73	7
15	42	$(45558)^A$	26	57
16	54	28	$(0)^B$	4
17	60	42	$(1)^B$	5
Átlag:		55,6	42,5	16,6

2.1. táblázat. Az erőhívások száma az általunk vizsgált módszereknél. (A =nem konvergált a NEB; B =GFN2-xTB nem konvergált)



2.2. ábra: Az erőhívások száma a példareakció függvényében kék = Delta-ML, fekete = GFN2-xTB-NEB (Ábrázolásakor az erőhívások számát 150-ig ábrázoltuk az Y-on tengelyen, azonban a 4, 5 és 15-ös példa esetén ennél jóval magasabb értékek adódtak)

módszer egyetlen eset kivételével kevesebb erőszámítást igényelt és gyorsabban futott a Direct-ML-nél.



2.3. ábra: Konvergáló futások százalékos aránya a Direkt-ML (szürke) és a Direct-ML (kék) alapú NEB futások esetén. Színezett rész –szürke = Direct-ML, kék = Delta-ML konvergált szimulációk, csíkozott rész –hibás futások.

Mivel az ML-NEB algoritmusok során egyszerre optimalizáljuk a PES-t és a közelítő MEP-et, oszcilláció léphet fel. Így előfordulhat, hogy az ML-NEB hamar egy jó közelítő MEP-t ad, mégsem konvergál az algoritmus. Hogy feltérképezzük ezt a lehetőséget, megvizsgáltuk, hogy az ML alapú algoritmusok kicserélése tradicionális, tisztán GFN2-xTB erőkon alapú NEB futásokra 25, 50 és 75 lépés után, eredményez-e gyorsabb konvergenciát. A 2. táblázatban összefoglalt eredményekből látható, hogy a kettes reakció esetén már 25 iteráció után a módszer az GFN2-xTB-nek megfelelő energiát adja vissza. Azonban a 75. iterációnál a GFN2-xTB energiáját csak másik három példánál éri el. Érdekes eredmény, hogy az 5-ös reakció esetén a 25 iterációnál még a GFN2-xTB konvergál addig a további lépések után már nem képes konvergenciára jutni. Ideális esetben a látott változás ugyanúgy kellene, hogy viselkedjen mint a 1, 3, 10, 14 -es esetekben, ahol a futási idővel növekszik a pontosság, azaz csökken az erőhívások száma amit a GFN2-xTB-nek kell elvégezni az itt tekintett pontosságú energiához. Ezzel szemben láthatunk példákat ahol a ellenkezőre mint például a 6-os reakció, ahol

reakció sorszáma	szabadsági fokok száma	Direct-ML	Direct-ML	Direct-ML
		25	50	75
1	3	62	45	45
2	6	0	0	0
3	9	17	1	0
4	15	64	0	0
5	24	416	(25031) ^A	(22503) ^A
6	24	178	107	215
7	24	63	76	62
8	24	31	31	30
9	27	136	136	136
10	27	42	18	3
11	36	53	53	53
12	36	60	60	60
13	42	6030	6430	6124
14	42	28	9	0
15	42	(19399) ^A	(18811) ^A	(20050) ^A
16	54	28	28	28
17	60	(10740) ^A	(11020) ^A	(11296) ^A
Átlag:		478,7	499,5	450,4

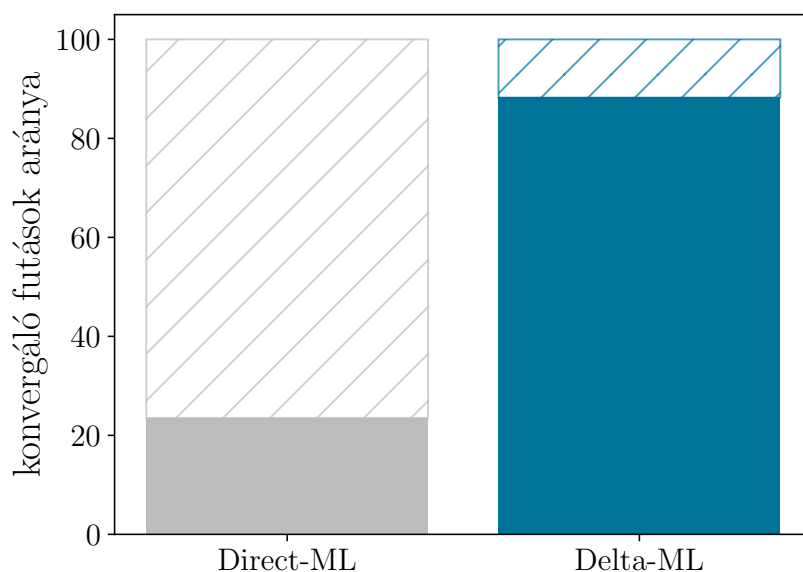
2.2. táblázat. A NEB GFN2-xTB módszerrel történő konvergáltatásához szükséges lépések száma, ha 25, 50 illetve 75 lépés hosszú Direct-ML szimuláció eredményéből indultunk ki (A =GFN2-xTB nem konvergált)

ugyan kezdetben javul az eredmény de a 75. iteráció esetén pontatlanabb választ ad a módszer. Érdekes még megemlíteni a 7-es reakciót, ahol az 50. iteráció ugyan pontatlanabb eredményt ad mint a 25-ös viszont ebben az esetben is pontosabb a 75. iteráció esetén a program. Viszont érdekes kérdést vet fel mivel a 75. iteráció esetén és a 25. iteráció esetén a program mindössze 1 extra erőhívással adott megoldást, vajon szükség van-e minden esetben az összes lépésre a ML módszer során. A fentiek alapján érdemes lehet az algoritmust úgy továbbfejleszteni, hogy a segéd ML felületen végzett NEB optimalizálás előtt megvizsgáljuk, hogy az eredeti algoritmus FIRE lépése konvergálna-e, valamint úgy, hogy az ML alapú optimalizálást bizonyos lépés szám, vagy maximum erő után kikapcsoljuk.

A 3. táblázat esetében is tovább futtattuk az eredményeinket GFN2-xTB-vel, a pontos módszerrel. Ennek során azt vizsgálva hány lépést tesz a GFN2-xTB a konvergenciáig, amiből következtethetünk, a Delta-ML pontosságára. Továbbá figyelembe vettük, hogy hányadik tanulási iteráció volt amit tovább futtattunk. A táblázatból látszik, hogy a 25. iteráció változó eredményeket ad. A NEB-ek 64,7%-a a pontos megoldást adja. Ez az érték a 50-es iterációnál már 82,4%, 75. iterációnál már 88,2%. Érdekes megemlíteni, hogy a 15-ös reakciónál a GFN2-xTB nem konvergál a 25. és az 50. iteráció részeredményéből, ennek ellenére a végső 75. iteráció utáni NEB-től nulla lépést tesz a GFN2-xTB azaz pontosnak találja azt. A 12-es és 13-as reakció során a több ML-iteráció nem pontosítja az eredményt. Míg az 3, 4 és 5-ös reakciók esetén az 50. iterációra visszkapjuk a GFN2-xTB által konvergált NEB-et. A 2. és a 3. táblázatot összevetve láthatjuk, hogy a Delta-ML jelentősen jobban közelíti meg a GFN2-xTB-t amit az alábbi ábrán láthatunk.

reakció sorszáma	szabadsági fokok száma	Delta-ML	Delta-ML	Delta-ML
		25	50	75
1	3	0	0	0
2	6	0	0	0
3	9	0	0	0
4	15	57	0	0
5	24	57	0	0
6	24	73	0	0
7	24	0	0	0
8	24	0	0	0
9	27	0	0	0
10	27	0	0	0
11	36	0	0	0
12	36	1	1	1
13	42	141	141	141
14	42	0	0	0
15	42	$(21010)^A$	$(20473)^A$	0
16	54	0	0	0
17	60	0	0	0
Átlag:		20,6	8,8	8,3

2.3. táblázat. A NEB GFN2-xTB módszerrel történő konvergáltatásához szükséges lépések száma, ha 25, 50 illetve 75 lépés hosszú Delta-ML szimuláció eredményéből indultunk ki (A =GFN2-xTB nem konvergált)



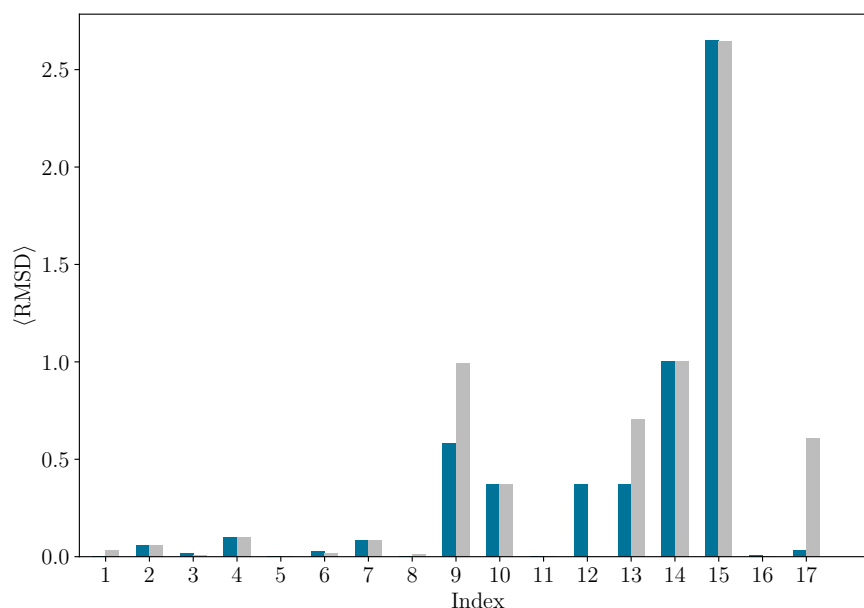
2.4. ábra: Konvergáló futások százalékos aránya a Direkt-ML (szürke) és a Direct-ML (kék) alapú NEB futások esetén. Színezett rész –szürke = Direct-ML, kék = Delta-ML konvergált szimulációk, csíkozott rész –hibás futások.

A fentebbi ábrán ábrázoltam a Direct- és Delta-ML által adott pontos energiák arányát. Amiket a tovább futtatás esetén kaptunk a fentebbi táblázatokban. Az energiákat akkor tekinthetjük pontosnak jelen esetben ha a GFN2-xTB-vel nem kellene további lépéseket végeznünk a PES-en hanem teljesíthük a konvergencia feltételt, így 0 lépésből konvergál a NEB-ünk

reakció sorszám	szabadsági fokok száma	RMSD-átlag Delta-ML	RMSD-max Delta-ML	RMSD-átlag Direct-ML	RMSD-max Direct-ML
1	3	0.00	0.02	0.03	0.11
2	6	0.06	0.16	0.06	0.16
3	9	0.02	0.04	0.01	0.03
4	15	0.10	0.19	0.10	0.20
5	24	B	B	B	B
6	24	0.03	0.04	0.02	0.05
7	24	0.09	0.24	0.09	0.22
8	24	0.00	0.01	0.01	0.05
9	27	0.58	0.94	0.99	1.82
10	27	0.37	0.68	0.37	0.68
11	36	C	C	C	C
12	36	0.37	0.68	A	A
13	42	0.37	0.76	0.70	1.00
14	42	01.01	1.77	01.01	1.77
15	42	2.65	2.75	2.64	2.75
16	54	0.01	0.02	A	A
17	60	0.03	0.07	0.61	0.84

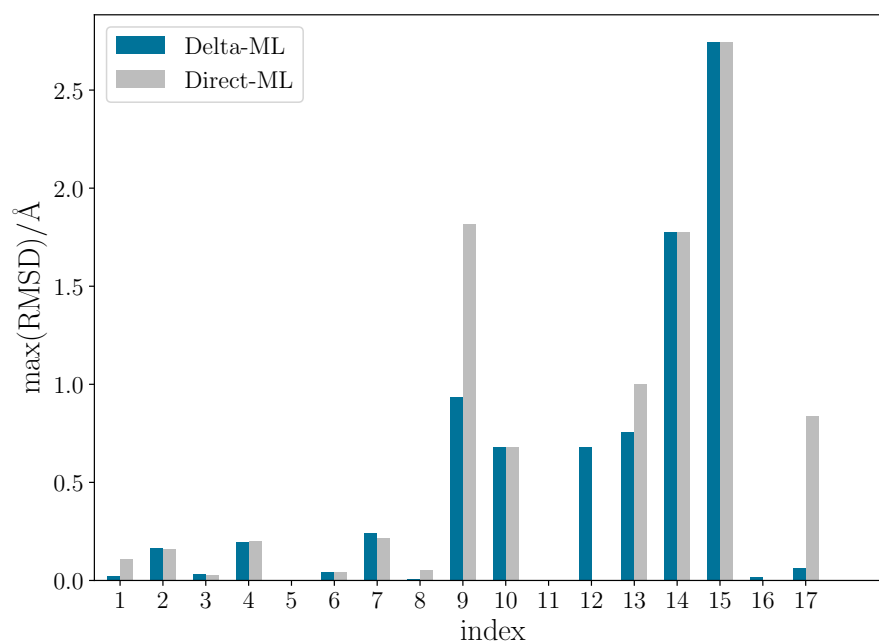
2.4. táblázat. RMSD értékek táblázata A = a módszer az első lépésben nem konvergált, B = GFN2-xTB nem konvergált, C = az RMSD számolás során hiba történt

Az alábbi RMSD értékeket úgy számítottuk ki, hogy a GFN2-xTB által végzett futtatásokat vettük referenciaként. Ehhez viszonyítottuk a két ML módszer által létrehozott NEB-eket. Ezt úgy végeztük el, hogy egy script segítségével azonos állásba forgattuk a molekulákat és kiszámoltuk közöttük az RMSD értéket.



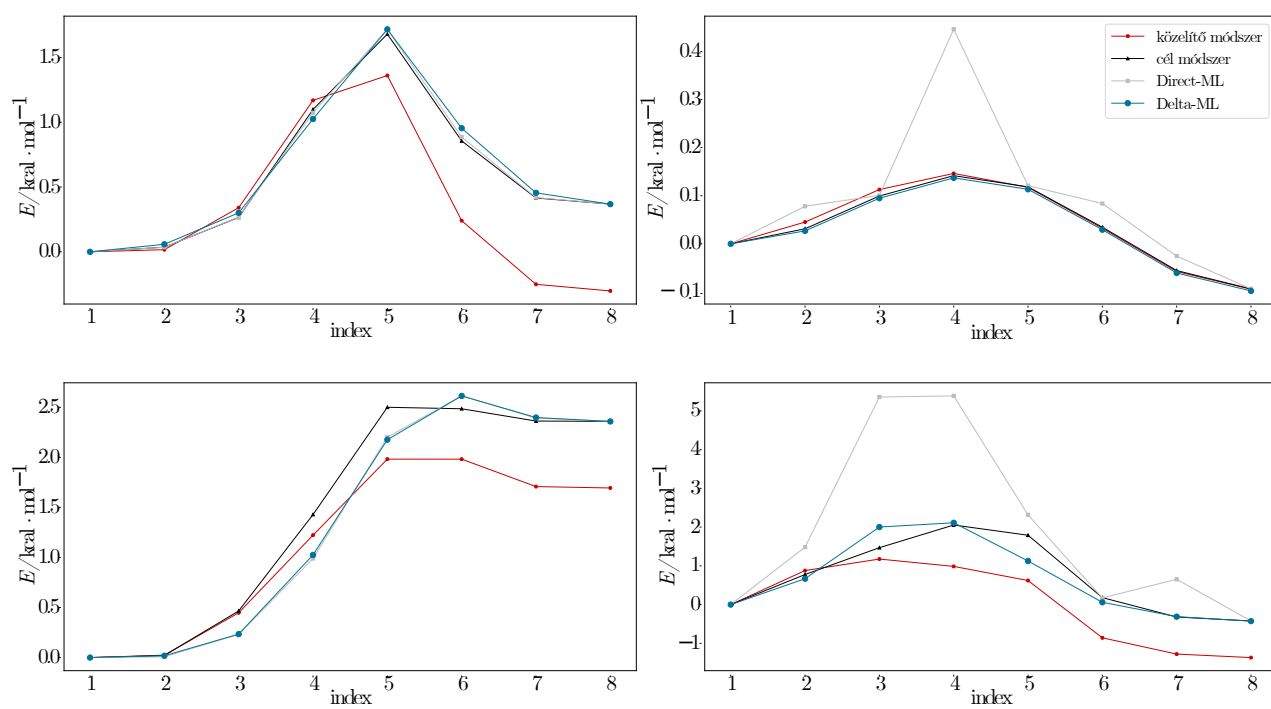
2.5. ábra: Átlag RMSD értékek ábrázolva a reakció sor számának függvényében szürke – Direct-ML, kék – Delta-ML

A fentebbi ábrán a teljes NEB átlagos RMSD-t ábráztuk, mind a Direct- és Delta-ML-re. Az eltérés nem minden esetben látványos azonban a 9-es és a 17-es esetekben látható, hogy jelentősen kisebb, ami azt reprezentálja, hogy az eredményünk pontosabb, azaz jobban közelíti a GFN2-xTB által elértet. A teljese grafikonon megfigyelhető az a tendencia, hogy a Delta-ML kisebb RMSD értékeket ad így nagyobbak tekinthető a pontossága.

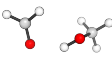
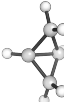
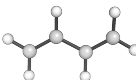
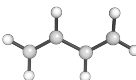
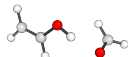
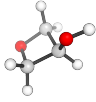
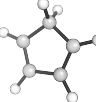
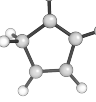
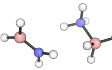
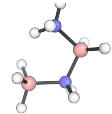
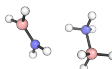
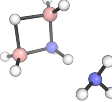
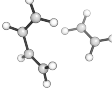
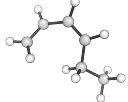


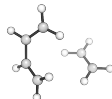
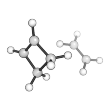
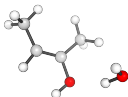
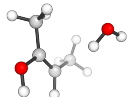
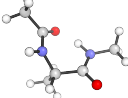
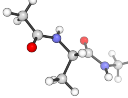
2.6. ábra: Maximális RMSD értékek ábrázolva a reakció sor számának függvényében szürke – Direct-ML, kék – Delta-ML

Az ábrán a maximális RMSD értékeket hasonlítottuk össze a két ML módszernél a GFN2-xTB függvényében. Ebben az esetben is jól kivehető a tendencia, hogy a Delta-ML RMSD értékei kisebbek. Ez azt jelenti, hogy akkurátusabb eredményre jut mint a Direct-ML. Akár csak az RMSD-k átlagánál itt is kiemelkedően jó eredményt mutat a Delta-ML a 9-es és 17-es reakció esetén, azonban érdekes, hogy a 7-es reakció esetén a Delta-ML ad nagyobb maximális értéket, ez azért is különös mivel az átlagos RMSD-énél a reakciónál nem jelenik meg.



2.7. ábra: A reakciók energiái relatíve a kiindulási állapot energiájához képest. Piros – GFN0-xTB, fekete – GFN2-xTB, szürke – Direct-ML, kék – Delta-ML

reakció sorszáma	szabadsági fokok száma	Reaktáns	Termék	Ref.
1	3			A
2	6			B
3	9			B
4	15			B
5	24			A
6	24			B
7	24			B
8	24			B
9	27			B
10	27			A
11	36			A
12	36			A
13	42			A

reakció sorszáma	szabadsági fokok száma	Reaktáns	Termék	Ref.
14	42			A
15	42			A
16	54			A
17	60			C

2.5. táblázat. a vizsgált kémiai reakciók reaktánsai és termékei. A=ref [15] B=ref [25] C=ref [28]

2.4. Összefoglalás

Kutatásaink során célul tűztük ki, hogy az eredeti direct-ML-en alapuló ML-NEB módszert pontosabbá és robosztusabbá tegyük. Eredményeink alapján kijelentjük, hogy a vizsgált rendszerek esetén az új algoritmus háromszor robosztusabb, mint az eredeti ML-NEB eljárás. Az általunk készített Delta-ML ráadásul átlagosan négyszer kevesebb erőhívás után konvergál, mint az eredeti NEB szimulációk. TDK dolgozatomban két DFTB módszer különbségét illesztettük eljárásunkkal, azonban további kutatásaink során szeretnénk magasabb szintű (pl. DFT, MP2) módszerekre is alkalmazni. Továbbá szeretnénk még vizsgálni más gépi tanuláson alapuló módszereket is például egy Performant Atomic Cluster Expansion-ön alapulót[29], amelyet később szintén kiegészítenénk Delta-ML-el.

2.5. Köszönetnyilvánítás

Köszönöm Kubicskó Károlynak a dolgozat alapos átolvasását és értékes javaslatait.

Irodalomjegyzék

- [1] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Big data meets quantum chemistry approximations: the δ -machine learning approach. *Journal of Chemical Theory and Computation*, 11(5):2087–2096, 2015.
- [2] Frank Jensen. *Introduction to computational chemistry*. John Wiley & Sons, 2017.
- [3] I Ge Ross. Calculations of the energy levels of acetylene by the method of antisymmetric molecular orbitals, including σ - π interaction. *Transactions of the Faraday Society*, 48:973–991, 1952.
- [4] Walter Kohn, Axel D Becke, and Robert G Parr. Density functional theory of electronic structure. *The Journal of Physical Chemistry*, 100(31):12974–12980, 1996.
- [5] Pierre Hohenberg and Walter Kohn. Inhomogeneous electron gas. *Physical Review*, 136(3B):B864, 1964.
- [6] Fernand Spiegelman, Nathalie Tarrat, Jérôme Cuny, Leo Dontot, Evgeny Posenitskiy, Carles Martí, Aude Simon, and Mathias Rapacioli. Density-functional tight-binding: basic concepts and applications to molecules and clusters. *Advances in Physics: X*, 5(1):1710252, 2020.
- [7] Christoph Bannwarth, Eike Caldeweyher, Sebastian Ehlert, Andreas Hansen, Philipp Pracht, Jakob Seibert, Sebastian Spicher, and Stefan Grimme. Extended tight-binding quantum chemistry methods. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 11(2):e1493, 2021.
- [8] Eike Caldeweyher, Jan-Michael Mewes, Sebastian Ehlert, and Stefan Grimme. Extension and evaluation of the d4 london-dispersion model for periodic systems. *Physical Chemistry Chemical Physics*, 22(16):8499–8512, 2020.

- [9] Gregory Mills, Hannes Jónsson, and Gregory K Schenter. Reversible work transition state theory: application to dissociative adsorption of hydrogen. *Surface Science*, 324(2-3):305–337, 1995.
- [10] Weiqing Ren, Eric Vanden-Eijnden, et al. Simplified and improved string method for computing the minimum energy paths in barrier-crossing events. *The Journal of Chemical Physics*, 126(16):164103, 05 2007.
- [11] Paul Zimmerman. Reliable transition state searches integrated with the growing string method. *Journal of Chemical Theory and Computation*, 9(7):3043–3050, 2013.
- [12] Nikolai A Zarkevich and Duane D Johnson. Nudged-elastic band method with two climbing images: Finding transition states in complex energy landscapes. *The Journal of Chemical Physics*, 142(2), 2015.
- [13] Christian Tönsing, Jens Timmer, and Clemens Kreutz. Optimal paths between parameter estimates in non-linear ode systems using the nudged elastic band method. *Frontiers in Physics*, 7:149, 2019.
- [14] Graeme Henkelman, Blas P Uberuaga, and Hannes Jónsson. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *The Journal of Chemical Physics*, 113(22):9901–9904, 2000.
- [15] Vilhjálmur Ásgeirsson, Benedikt Orri Birgisson, Ragnar Björnsson, Ute Becker, Frank Neese, Christoph Riplinger, and Hannes Jónsson. Nudged elastic band method for molecular reactions using energy-weighted springs combined with eigenvector following. *Journal of Chemical Theory and Computation*, 17(8):4929–4945, 2021.
- [16] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical Review Letters*, 98(14):146401, 2007.
- [17] János Daru, Harald Forbert, Jörg Behler, and Dominik Marx. Coupled cluster molecular dynamics of condensed phase systems enabled by machine learning potentials: Liquid water benchmark. *Physical Review Letters*, 129(22):226001, 2022.

- [18] Volker L Deringer, Albert P Bartók, Noam Bernstein, David M Wilkins, Michele Ceriotti, and Gábor Csányi. Gaussian process regression for materials and molecules. *Chemical Reviews*, 121(16):10073–10141, 2021.
- [19] Yilin Yang, Omar A Jiménez-Negrón, and John R Kitchin. Machine-learning accelerated geometry optimization in molecular simulation. *The Journal of Chemical Physics*, 154(23), 2021.
- [20] Ralf Meyer and Andreas W Hauser. Geometry optimization using gaussian process regression in internal coordinate systems. *The Journal of Chemical Physics*, 152(8), 2020.
- [21] Alexander Denzel and Johannes Kästner. Gaussian process regression for transition state search. *Journal of Chemical Theory and Computation*, 14(11):5777–5786, 2018.
- [22] Alexander Denzel and Johannes Kästner. Hessian matrix update scheme for transition state search based on gaussian process regression. *Journal of Chemical Theory and Computation*, 16(8):5083–5089, 2020.
- [23] José A Garrido Torres, Paul C Jennings, Martin H Hansen, Jacob R Boes, and Thomas Bligaard. Low-scaling algorithm for nudged elastic band calculations using a surrogate machine learning model. *Physical Review Letters*, 122(15):156001, 2019.
- [24] Ralf Meyer, Klemens S Schmuck, and Andreas W Hauser. Machine learning in computational chemistry: An evaluation of method performance for nudged elastic band calculations. *Journal of Chemical Theory and Computation*, 15(11):6513–6523, 2019.
- [25] Sebastian Dohm, Markus Bursch, Andreas Hansen, and Stefan Grimme. Semiautomated transition state localization for organometallic complexes with semiempirical quantum chemical methods. *Journal of Chemical Theory and Computation*, 16(3):2002–2012, 2020.
- [26] Nongnuch Artrith and Alexie M Kolpak. Understanding the composition and activity of electrocatalytic nanoalloys in aqueous solvents: a combination of dft and accurate neural network potentials. *Nano letters*, 14(5):2670–2676, 2014.
- [27] Erik Bitzek, Pekka Koskinen, Franz Gähler, Michael Moseler, and Peter Gumbsch. Structural relaxation made simple. *Physical Review Letters*, 97(17):170201, 2006.

- [28] Davide Mandelli and Michele Parrinello. A modified nudged elastic band algorithm with adaptive spring lengths. *The Journal of Chemical Physics*, 155(7), 2021.
- [29] Yu Lysogorskiy, Cas Oord, Anton Bochkarev, Sarath Menon, Matteo Rinaldi, Thomas Hammerschmidt, Matous Mrovec, Aidan Thompson, Gábor Csányi, Christoph Ortner, and Ralf Drautz. Performant implementation of the atomic cluster expansion (pace) and application to copper and silicon. *npj Comput Mater*, 7(97), 2021.