

Molekularna dinamika na strojno naučenoj plohi potencijalne energije

Ovčar, Juraj

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:000337>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-19**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

Juraj Ovčar

MOLEKULARNA DINAMIKA NA STROJNO
NAUČENOJ PLOHI POTENCIJALNE ENERGIJE

Diplomski rad

Zagreb, 2019.

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

INTEGRIRANI PREDDIPLOMSKI I DIPLOMSKI SVEUČILIŠNI STUDIJ
FIZIKA; SMJER ISTRAŽIVAČKI

Juraj Ovčar

Diplomski rad

**Molekularna dinamika na strojno
naučenoj plohi potencijalne energije**

Voditelj diplomskog rada: dr. sc. Ivor Lončarić

Ocjena diplomskog rada: _____

Povjerenstvo: 1. _____

2. _____

3. _____

Datum polaganja: _____

Zagreb, 2019.

Zahvaljujem se prije svega mentoru dr. sc. Ivoru Lončariću na razjašnjavanju potrebnih fizikalnih i računalnih koncepata te na ukazanom strpljenju i savjetima pri rješavanju različitih problema koji su se pojavili tijekom izrade ovog diplomskog rada.

Također se zahvaljujem svojim prijateljima, djevojci i posebno obitelji, bez čije bezuvjetne podrške izrada ovog diplomskog rada ne bi bila moguća.

Sažetak

Za modeliranje materijala iz prvih principa danas se predominantno koristi teorija funkcionala gustoće, koja pruža dobar omjer preciznosti i računalne efikasnosti. Međutim, zbog trenutnih računalnih ograničenja, obično je prezahtjevno modelirati dinamiku sustava s desecima atoma. Jedna od obećavajućih metoda za približavanje simulacija realnim sustavima i povećanje skala na kojima je moguće provoditi dinamiku je interpolacija potencijala dobivenog s teorijom funkcionala gustoće pomoću neuronskih mreža. U ovom diplomskom radu smo simulirali sustav molekula ugljičnog monoksida adsorbiranih na površini paladija ozračenoj femtosekundnim laserskim pulsom. Prikazali smo proces strojnog učenja te implementirali molekularno-dinamičke simulacije u modelu Langevinove dinamike s vremenski ovisnim temperaturama.

Ključne riječi: teorija funkcionala gustoće, strojno učenje, neuronske mreže, ploha potencijalne energije, heterogena kataliza, laserski inducirana dinamika, Langevinova dinamika, model dviju temperatura, molekularno-dinamičke simulacije.

Molecular dynamics on machine-learned potential energy surface

Abstract

Nowadays, density functional theory is the predominant tool used in first-principles material modeling, providing an adequate ratio between precision and computational efficiency. However, because of current computational limitations, it is usually too demanding to simulate dynamics of systems consisting of tens of atoms. A promising method for reducing the gap between simulations and realistic systems and extending the scales across which it is possible to simulate dynamics is interpolation of potentials calculated with density functional theory via neural networks. In this master's thesis we have simulated a system of carbon monoxide molecules adsorbed on palladium irradiated with a femtosecond laser pulse. We have shown the process of machine-learning and implemented molecular-dynamics simulations in the Langevin dynamics model with time dependent temperatures.

Keywords: Density Functional Theory, Machine Learning, Neural Networks, Potential Energy Surface, Heterogeneous Catalysis, Laser-Induced Dynamics, Langevin Dynamics, Two Temperature Model, Molecular Dynamics Simulations

Sadržaj

1	Uvod	1
2	Uvod u strojno učenje	3
2.1	'Feed-forward' neuronske mreže	3
2.2	Simetrijske funkcije	6
2.3	Programski paket Amp	8
3	Desorpcija dvoatomskih molekula s metalnih površina inducirana femto-	
	sekundnim laserskim pulsevima	10
3.1	Teorijska razmatranja	10
3.2	O podacima korištenim za strojno učenje	12
4	Uvježbavanje neuronskih mreža	14
4.1	Preciznost neuronskih mreža u ovisnosti o strukturi mreže i <i>cutoff</i> pa-	
	rametru	14
4.2	Komentar o <i>cutoff</i> -parametru	17
4.3	Uvježbavanje na skupovima s više slika	18
4.4	Testiranje mreže na (3×3) geometriji	19
5	Molekularna dinamika	22
5.1	NVE dinamika	22
5.2	Vremenski ovisna Langevinova dinamika	25
6	Zaključak	29
	Dodaci	31
A	Povjesna crtica o heterogenoj katalizi	31
B	Python kod za vremenski ovisnu Langevinovu dinamiku	33
	Literatura	42

1 Uvod

Modeliranje materijala iz prvih principa, posebice modeliranje pomoću teorije funkcionala gustoće (DFT¹), je u današnjici vrijedan komplement eksperimentalnim istraživanjima materijala [1]. S praktične strane, takvo modeliranje se može koristiti u svrhu ubrzanja karakterizacije i potrage za novim materijalima, a s konceptualne omogućava razumijevanje svojstava materijala preko ponašanja na atomskoj razini [2]. Usprkos tome što za mnoge primjene DFT pruža najbolji omjer preciznosti i računalne efikasnosti, proračuni dinamičkih svojstava materijala pomoću DFT-a su često računalno prezahtjevni i stoga se tradicionalno pribjegava korištenju klasičnih/empiričkih potencijala (polja sila) manje preciznosti u odnosu na DFT. Međutim, korištenjem DFT-a u kombinaciji s metodama strojnog učenja mogu se produžiti dužinske i vremenske skale preko kojih se izvršavaju simulacije atomističke dinamike [3]. Osnovna ideja je konstruirati plohu potencijalne energije prilagodbom DFT proračuna na vrlo fleksibilnu funkciju s velikim brojem parametara. Jednom kad je ploha konstruirana, energije i sile se mogu brzo izvrijednjivati te se tako omogućavaju dulje simulacije na većim sistemima [4]. Efikasna prilagodba podataka se može izvršiti pomoću metoda strojnog učenja, a u ovom radu se koriste umjetne neuronske mreže za koje je već pokazano da ostvaruju visoku preciznost pri prilagodbi potencijalnih ploha [5–9].

U ovom radu ćemo najprije prikazati uvježbavanje² neuronskih mreža, a zatim ćemo prikazati korištenje dobivenog potencijala za molekularnu dinamiku. Sustav čiju smo dinamiku proučavali sastoji se od molekula ugljičnog monoksida (CO) adsorbiranih na površini paladija (Pd(111)). Razumijevanje dinamike ovakvih sustava je od praktičnog (tehnološkog) interesa za, primjerice, katalitičke konvertere u automobilima, ili Fischer-Tropsch proces, odnosno u širem smislu za razumijevanje procesa heterogene katalize; u Dodatku A opisani su rani razvoji teorijskog opisa takvih fenomena. Kako bi se što preciznije simulirala dinamika ovakvih sustava, često je potrebno uzeti u obzir interakciju među adsorbiranim molekulama pa se posljedično u jediničnu ćeliju simulacije mora uključiti velik broj čestica; simulacije dinamike takvih sustava s velikim brojem atoma i stupnjeva slobode na razini teorije funk-

¹Eng. Density Functional Theory.

²Eng. training.

cionala gustoće su računalno vrlo zahtjevne. Kao što ćemo vidjeti u ovom radu, moguće je umanjiti te računalne zahtjeve tako da se pomoću DFT-a izračuna potencijalna energija različitih konfiguracija i potom se interpolacijom konstruira glatka ploha potencijalne energije (postupak interpolacije se naziva uvježbavanjem mreže - vidi poglavlja 2 i 4). Na tako konstruiranom potencijalu smo pomoću vremenski ovisne Langevinove dinamike modelirali dinamiku molekula induciranih femtosekundnim laserskim pulsom (vidi odjeljak 3.1). Dakle, kao rezime, u ovom radu smo pomoću strojno naučene plohe potencijalne energije simulirali dinamiku molekula CO na površini Pd(111) obasjanoj femtosekundnim laserskim pulsom.

2 Uvod u strojno učenje

2.1 'Feed-forward' neuronske mreže

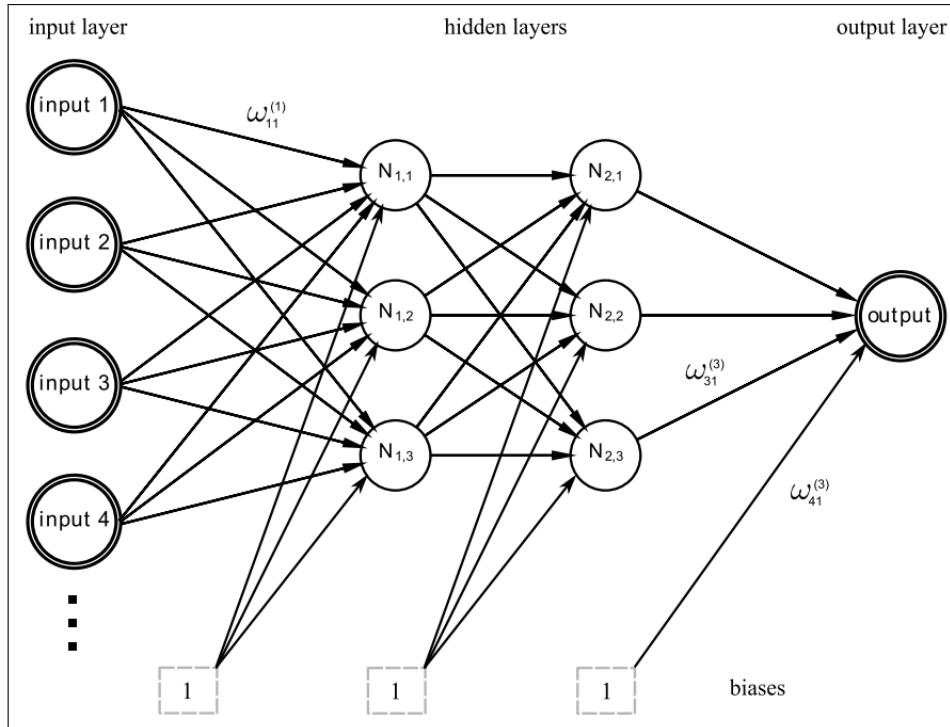
Zbog trenutnih ograničenja računalnih mogućnosti računala, korištenje *ab initio* metoda elektronske strukture za molekularnu dinamiku [10] je moguće samo za sustave sačinjene od maksimalno nekoliko stotina atoma. Od takvih *ab initio* metoda, zbog dobrog omjera preciznosti i efikasnosti dominantno se koristi teorija funkcionala gustoće (DFT). Međutim, broj simuliranih atoma i vrijeme simulacije su često nedovoljni za adekvatan opis realnih sistema. Stoga su razvijene daljnje aproksimativne metode čiji je cilj konstrukcija potencijalne plohe kao funkcije atomskih koordinata (često se te plohe skraćeno nazivaju PES³) na kojoj je molekularna dinamika računalno provediva. Elektronski stupnjevi slobode u dinamičkim procesima se u takvim modelima ne uzimaju eksplicitno, nego primjerice uvođenjem fluktuacijsko-disipacijskih sila. Ideja je izračunati energiju pojedinih atomskih konfiguracija nekom metodom, primjerice DFT-om, a onda ih interpolirati u PES. Za interpolaciju tako dobivenih točaka razvijene su mnoge metode, čiji je pregled dan u referenci [11]. Mi smo točke dobivene DFT-om interpolirali koristeći neuronsku mrežu (skraćeno NN) [3].

Shema tipične 'feed-forward' neuronske mreže prikazana je na Slici 2.1. Sastoji se od ulaznog sloja, proizvoljnog broja slojeva skrivenih neurona i izlaznog čvora. Neuronska mreža iz ulaznih podataka (input, u našem slučaju atomske koordinate) učita N_0 brojeva, gdje N_i predstavlja broj neurona u i -tom sloju. Ti brojevi se šalju skrivenim neuronima koji generiraju niz N_1 brojeva koje prosljeđuju drugom sloju skrivenih neurona i tako dalje sve do izlaznog neurona u kojem se generira konačni rezultat (output, u našem slučaju potencijalna energija). Formula prema kojoj se računa output j -tog skrivenog neurona u i -tom čvoru je:

$$o_j^{(i)} = f \left(\sum_{k=1}^{N_i-1} \omega_{kj}^{(i)} o_k^{(i-1)} + b_j^{(i)} \right), \quad (2.1)$$

gdje je $\omega_{kj}^{(i)}$ težina (weight) s kojom output iz k -tog neurona iz prethodnog sloja ulazi u promatrani skriveni neuron, $b_j^{(i)}$ pristranost (bias) promatranog neurona, a

³Potential Energy Surface



Slika 2.1: Shematski prikaz 'feed-forward' neuronske mreže s dva sloja skrivenih neurona. U tekstu su pristranosti označene s b umjesto s ω kako se ne bi pomiješale s težinama. Preuzeto iz [12].

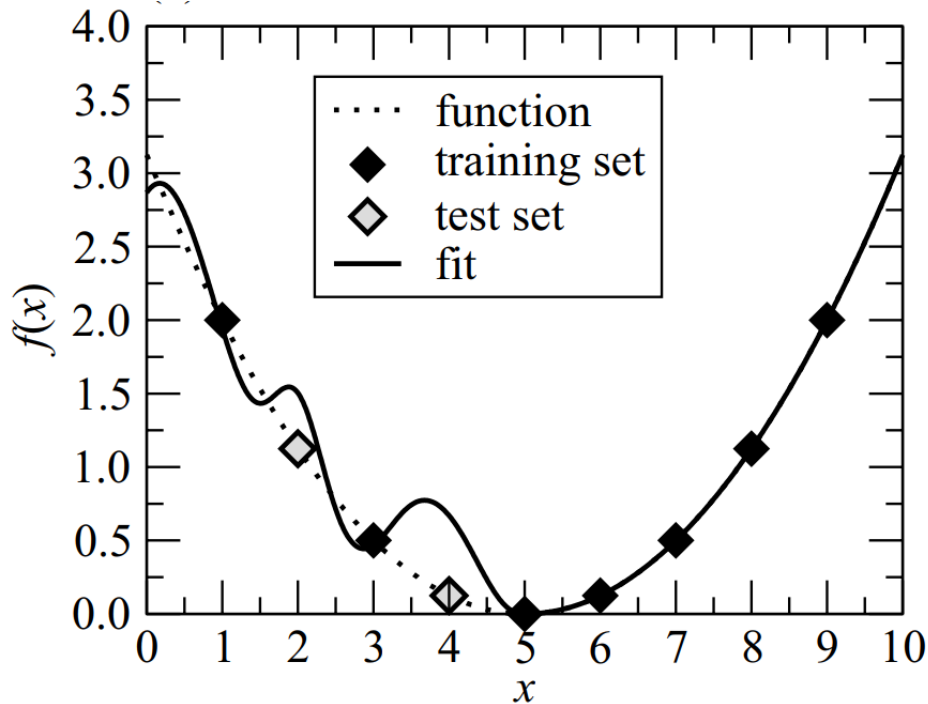
f njegova aktivacijska funkcija. Često korištene aktivacijske funkcije su tangens hiperbolni, sigmoidalna, Gaussova i linearna funkcija. Pokazano je da dovoljno velike 'feed-forward' neuronske mreže mogu proizvoljnom točnošću aproksimirati bilo koju kontinuiranu funkciju pod uvjetom da je f lokalno omeđena po dijelovima kontinuirana funkcija koja nije polinom [13]. Primjer 1D interpolacije neuronskom mrežom prikazan je na Slici 2.2. Skup točaka koje treba interpolirati naziva se skupom za uvježbavanje⁴, a proces variranja parametara mreže (težina i pristranosti) uvježbavanjem. Parametri mreže se variraju tako da minimiziraju neku zadanu "funkciju gubitka"⁵ koja mjeri odstupanje predviđenih vrijednosti od onih zadanih u skupu za uvježbavanje. Mi smo za funkciju gubitka koristili korijen srednjeg kvadratnog odstupanja ukupne energije sustava:

$$\Gamma = \sqrt{\frac{1}{M} \sum_{j=1}^M \left(\hat{E}_j - E_j \right)^2}, \quad (2.2)$$

gdje je M broj zadanih (training) atomskih konfiguracija, N_j broj atoma u j -toj

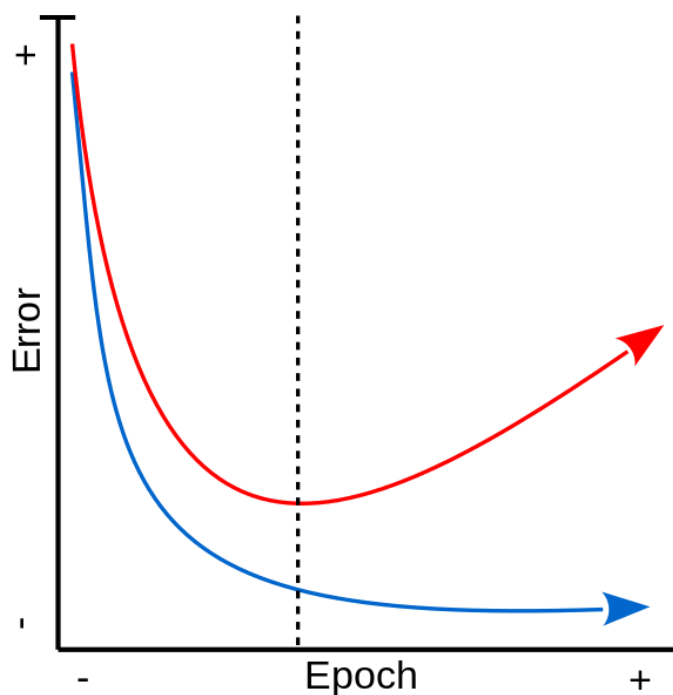
⁴Eng. training set.

⁵U stranoj literaturi "Loss function" ili "Cost function".



Slika 2.2: Primjer interpolacije 1D funkcije pomoću neuronske mreže. Preuzeto iz [3].

konfiguraciji, E_j energije konfiguracija izračunate DFT-om, a \hat{E}_j je predviđanje NN-a za j -tu konfiguraciju. Smatra se da je mreža konvergirala kad funkcija gubitka padne ispod neke zadane vrijednosti. Nakon konvergencije, uspoređuju se predviđanja NN-a za neke atomske konfiguracije različite od onih iz skupa za uvježbavanje (taj skup podataka naziva se testnim setom). *Overfitting* predstavlja jedan od značajnijih problema interpolacije neuronskim mrežama. Ako se zada presnažan uvjet konvergencije, NN će vrlo precizno interpolirati točke skupa za uvježbavanje tako da deformira interpolacijsku krivulju između tih točaka, povećavajući grešku na testnom setu. Takvi NN-ovi se ne generaliziraju dobro na nepoznate podatke. Rješenje problema *overfitting*-a ilustrirano je na slici 2.3 i često se naziva *early stopping*. Uvede se treći skup podataka različitih od onih iz skupova za uvježbavanje i testiranje, tzv. validacijski set. Svakih nekoliko iteracija uvježbavanja se računa funkcija gubitka na validacijskom setu. Za uvježbanu mrežu se onda uzme ona za koju je validacijska funkcija gubitka dosegla globalni minimum. Performanse tako uvježbane mreže se onda provjeravaju na testnom setu.



Slika 2.3: Ilustracija *overfitting*-a. Crvena linija predstavlja grešku na validacijskom setu, a plava grešku na skupu za uvježbavanje. Crtkanom linijom označena je iteracija uvježbavanja pri kojoj je validacijska funkcija gubitka dosegla minimum. Preuzeto s [14].

2.2 Simetrijske funkcije

U odsustvu vanjskih polja broj nezavisnih stupnjeva slobode sistema N atoma je $3N - 6$, gdje se oduzima 6 stupnjeva slobode koji odgovoraju translacijama i rotacijama čitavog sustava. Tih 6 stupnjeva slobode se može uzeti u obzir tako da se sustav umjesto s $3N$ prostornih koordinata opiše pomoću $3N - 6$ "unutarnjih" koordinata (primjerice duljine veza i valentnih ili dihedralnih kuteva). Međutim, tako definirane koordinate nisu dobar input za neuronsku mrežu ako, primjerice, u sustavu postoji više atoma iste vrste. Razlog je to što atomske koordinate u NN ulaze kao vektorstupac, gdje svaka komponenta odgovara jednom neuronu u ulaznom sloju pa, ako se koriste prostorne koordinate i permutiraju se dva identična atoma, NN za fizički istu konfiguraciju atoma može izračunati različitu energiju (budući da težine na pojedinim neuronima ulaznog sloja mogu varirati). Stoga se vektor atomskih koordinata \mathbf{R} preslikava *simetrijskim funkcijama* [3] \mathbf{G} u vektor $\mathbf{G}(\mathbf{R})$ invarijantan na operacije koje čuvaju simetriju sistema. Simetrijske funkcije koje smo mi koristili pripadaju klasi *atomski-centriranih* simetrijskih funkcija. Polazišna pretpostavka je da je ukupna energija sustava zbroj lokalnih doprinosa interakcija pojedinih atoma

sa njihovom neposrednom okolinom. U toj aproksimaciji neuronska mreža ukupnu energiju konfiguracije N atoma računa kao:

$$\hat{E} = \sum_{i=1}^N \hat{E}_i(\mathbf{G}_i(\mathbf{R})) \quad (2.3)$$

gdje je \mathbf{R} 3N-dimenzionalni vektor kartezijevih koordinata atoma, \mathbf{G}_i simetrijska funkcija i -tog atoma, a \hat{E}_i predviđeni energijski doprinos i -tog atoma (i njegove okoline). Ovaj pristup je koristan, između ostalog, jer je energija ekstenzivna u broju atoma, pa se NN može koristiti za sisteme različitih veličina. Mi smo koristili Gaussove simetrijske funkcije. U toj shemi se svaki vektor $\mathbf{G}_i(\mathbf{R})$ sastavlja unijom dva pomoćna podvektora: $\mathbf{G}_i(\mathbf{R}) = \mathbf{G}_i^I(\mathbf{R}) \cup \mathbf{G}_i^{II}(\mathbf{R})$. $\mathbf{G}_i^I(\mathbf{R})$ predstavlja interakcije i -tog atoma s pojedinačnim atomima, a $\mathbf{G}_i^{II}(\mathbf{R})$ interakcije i -tog atoma s parovima atoma (interakcije tri tijela). Komponente vektora $\mathbf{G}_i^I(\mathbf{R})$ se računaju za različite vrijednosti parametara η i R_C pomoću sljedeće formule:

$$G_{i,\eta R_S}^I(\mathbf{R}) = \sum_{j \neq i} e^{-\eta(R_{ij}-R_S)^2/R_C^2} f_C(R_{ij}), \quad (2.4)$$

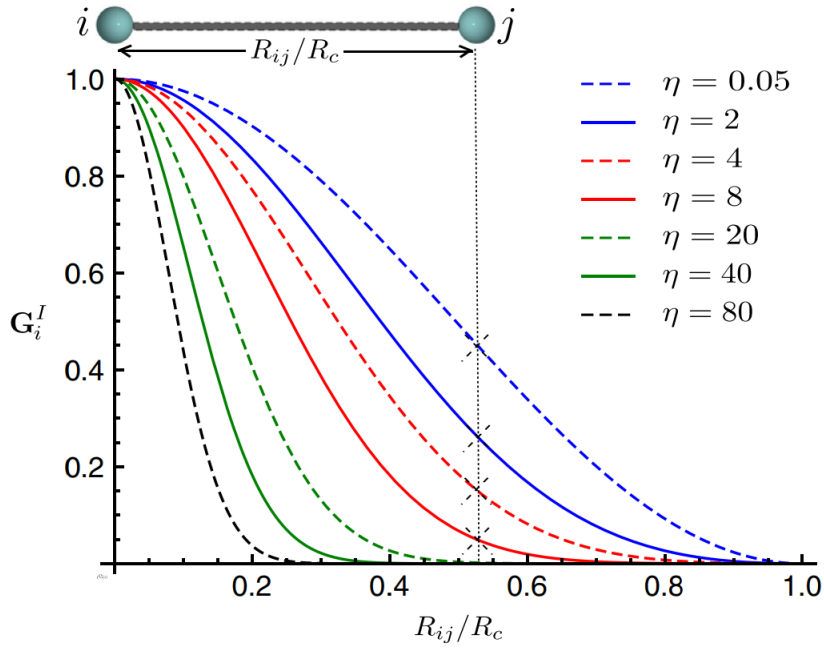
gdje su parametri η i R_S širina i centar gausijana, $R_{ij} = \|\mathbf{R}_j - \mathbf{R}_i\|$ je udaljenost između dva atoma, a $f_C(r)$ glatka funkcija na $r \leq R_C$ te vrijedi $f_C(r \geq R_C) = 0$. $f_C(r)$ se naziva *cutoff* funkcijom. Eksplicitno za $r \leq R_C$:

$$f_C(r) = 0.5(1 + \cos(\pi r/R_C)). \quad (2.5)$$

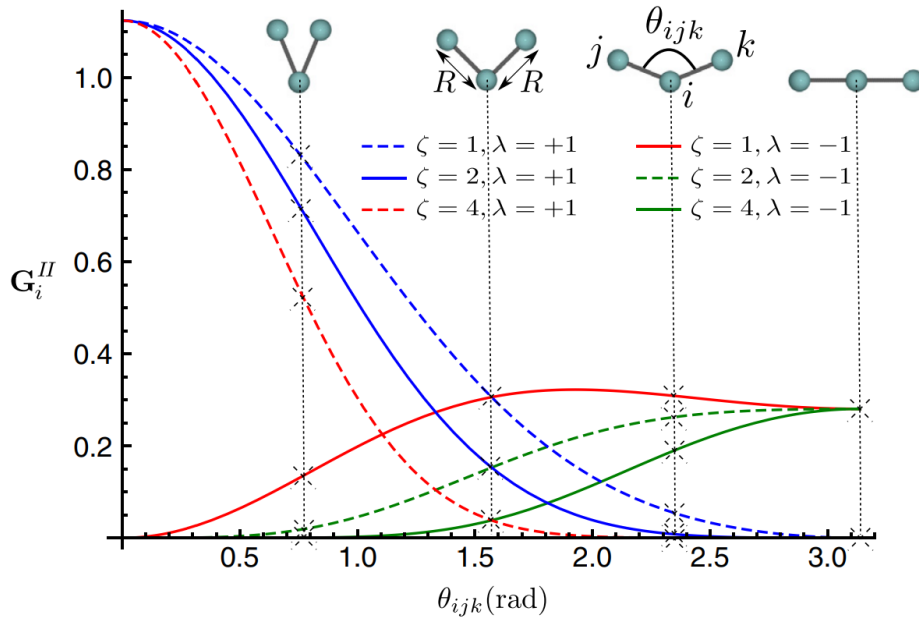
Komponente $\mathbf{G}_i^{II}(\mathbf{R})$ računaju se prema:

$$G_{i,\eta\zeta\lambda}^{II}(\mathbf{R}) = 2^{1-\zeta} \sum_{\substack{j,k \neq i \\ (j \neq k)}} (1 + \lambda \cos(\theta_{ijk}))^\zeta e^{-\eta(R_{ij}^2 + R_{ik}^2 + R_{jk}^2)/R_C^2} f_C(R_{ij}) f_C(R_{ik}) f_C(R_{jk}) \quad (2.6)$$

za različite vrijednosti parametara η , ζ i λ . θ_{ijk} je kut među tri atoma (vidi sliku 2.5). Na slikama 2.4 i 2.5 prikazani su grafovi komponenti $\mathbf{G}_i^I(\mathbf{R})$ i $\mathbf{G}_i^{II}(\mathbf{R})$.



Slika 2.4: Ovisnost komponenti $\mathbf{G}_i^I(\mathbf{R})$ o udaljenosti atoma za različite η , uz $R_S = 0$. Preuzeto iz [12].



Slika 2.5: Ovisnost komponenti $\mathbf{G}_i^{II}(\mathbf{R})$ o kutu među atomima za različite ζ i λ , uz $\eta = 0.005$ i $R = R_C/3$. Preuzeto iz [12].

2.3 Programski paket Amp

ASE⁶ [15] je skup alata i Python modula za konstrukciju, manipulaciju, vizualizaciju te vršenje atomističkih simulacija. ASE može služiti kao sučelje za mnoge pakete za proračun elektronske strukture (GPAW, Dacapo, VASP, Quantum ESPRESSO, ...).

⁶The Atomic Simulation Environment

Amp ⁷ [12], otvoreni (open-source) programski paket kojeg smo koristili za strojno učenje, je dizajniran kao modul ASE-a, odnosno proračuni elektronske strukture u nekom drugom paketu (u našem slučaju VASP-u) se učitavaju preko ASE-a u Amp. Uvježbane neuralne mreže se mogu koristiti u ASE-u za atomističke simulacije. Amp je većinom pisan u Pythonu, a kompjutacijski zahtjevni procesi su pisani u Fortranu.

⁷Atomistic Machine-learning Package

3 Desorpcija dvoatomskih molekula s metalnih površina inducirana femtosekundnim laserskim pulsevima

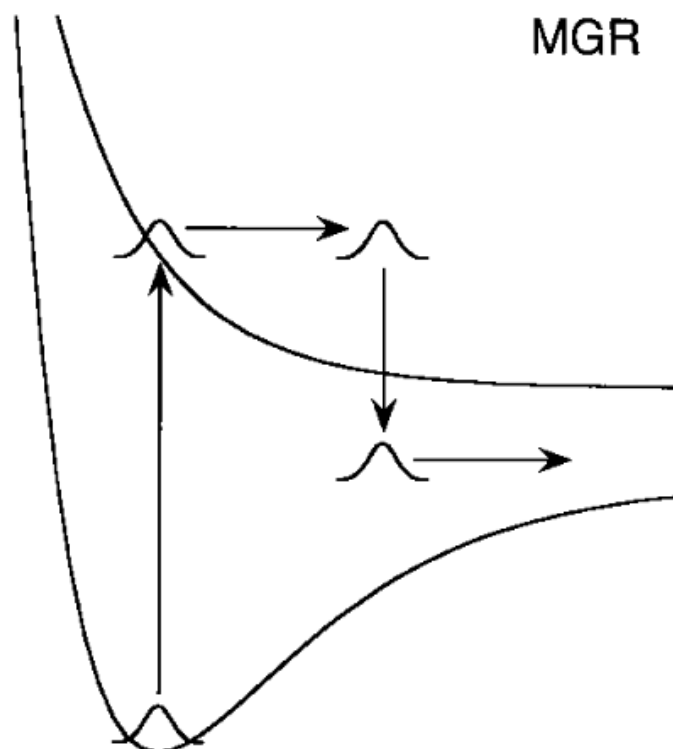
3.1 Teorijska razmatranja

Poznato je da ozračivanje površina intenzivnim i kratkovalnim (UV i vidljivim) laserskim zračenjem može izazvati difuziju, desorpciju, disocijaciju i kemijske reakcije adsorbiranih molekula (admolekula) [16]. Na izolatorskim površinama među supstratom i adsorbatom najčešće prevladavaju Van der Waalsove interakcije koje slabo perturbiraju elektronsku strukturu admolekula. Desorpcijska energija je mala (kaže se da su molekule fizisorpcirane), a admolekule energiju iz laserskog pulsa primaju direktno, pobuđenjem elektrona valentne ljuske. S druge strane, metali jako apsorbiraju UV zračenje, pa u većini fotokemijskih procesa na vodljivim površinama dominira indirektno pobuđenje admolekula preko jakih interakcija s elektronskih i fononskim podsustavom. Odziv metalne površine na lasersko pobuđenje može se aproksimirati pomoću makroskopskog modela "dviju temperatura" (2TM) [17]: elektronski i fononski podsustavi se modeliraju kao dva vezana toplinska rezervoara izbačena iz ravnoteže laserskim pulsom, pa je ovisnost njihovih temperatura o vremenu dana jednadžbama sačuvanja energije:

$$\begin{aligned} C_e \frac{\partial T_e}{\partial t} &= \frac{\partial}{\partial z} \kappa \frac{\partial T_e}{\partial z} - g(T_e - T_l) + S(z, t), \\ C_l \frac{\partial T_l}{\partial t} &= g(T_e - T_l), \end{aligned} \quad (3.1)$$

gdje su C_e i C_l toplinski kapaciteti elektrona i fonona, T_e i T_l njihove temperature, g je konstanta elektron-fononskog vezanja, κ je termalna vodljivost elektrona, z je okomita udaljenost od površine, a $S(z, t)$ apsorbirana energija lasera u jediničnom vremenu i volumenu. Za mnoge sustave je značajan mehanizam prijenosa energije s vodljive površine na molekulu formiranjem kratkoživućih ionskih vibracijskih stanja apsorpcijom elektrona s repa "zagrijane" Fermijeve distribucije (ili netermaliziranog elektrona) u molekularnu orbitalu (DIET⁸ procesi). Takvi procesi su prvi put kvalitativno opisani Menzel-Gomer-Redhead (MGR) neadijabatskim modelom (Slika 3.1). Promatra se jedna admolekula u osnovnom vibracijskom stanju potencijala Morse-

⁸Eng. Desorption Induced by Electronic Transitions



Slika 3.1: Shematski prikaz MGR modela za srednja vremena τ . Preuzeto iz [16].

ovog tipa. Apsorpcija laserski pobuđenog elektrona u molekularnu orbitalu modelira se naglom (instantnom) promjenom potencijalne plohe te ion admolekule biva projiciran u neko stanje novog (odbojnog) potencijala. Ovisno o vremenu života τ tog metastabilnog stanja, moguća su tri ishoda. Ako je τ dugo, s površine desorbira ion admolekule. Ako je τ kratko, admolekula padne u vezano stanje originalne potencijalne plohe i relaksacijom u osnovno stanje disipira energiju u metal. Takva disipacija se obično odvija stvaranjem elektron-šupljinskih parova na površini [18]. Između ta dva ekstrema postoji τ za koje admolekula ima vremena evoluirati u neko stanje iz kojeg se relaksira u stanje pozitivne energije originalne plohe te desorbira s površine. Postoje DIMET⁹ modeli, generalizacije MGR modela koji dozvoljavaju mogućnost niza realnih i virtualnih prijelaza između energetskih stanja osnovne i (kontinuumu) pobuđenih ploha potencijalne energije. Ovom problemu se može pristupiti iz drugog gledišta, koristeći generalizaciju klasične adijabatske (Langevinove) molekularne dinamike [19]. Molekularna dinamika se, umjesto na kontinuumu potencijalnih ploha, vrši u Born-Oppenheimerovoj aproksimaciji na jednoj efektivnoj plohi $V(\mathbf{r}_1, \dots, \mathbf{r}_N)$ ovisnoj o položajima atoma. Elektronski stupnjevi slobode u model

⁹Eng. Desorption Induced by Multiple Electronic Transitions

ne ulaze eksplicitno, već se disipacije uzimaju u obzir kroz silu trenja, a neelastična pobuđenja molekula na višu potencijalnu plohu stohastičkom silom. Obje ove sile ovise o temperaturi i povezane su fluktuacijsko-disipacijskim teoremom. Slike Langevinove dinamike i DIMET modela su ekvivalentne ako su u promatranom režimu oblici osnovne i pobuđenih ploha dovoljno slični te dolazi do mnogo oscilacija među njima. Dvije slike mogu unificirati jedinstvenim formalizmom [20]. Označavajući koeficijent trenja s $\eta_{e,i}$, a fluktuacijsku silu s $\mathbf{R}_{e,i}$, jednadžbe gibanja za pojedine atome u molekulama glase:

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = -\nabla_{\mathbf{r}_i} V(\mathbf{r}_1, \dots, \mathbf{r}_N) - \eta_{e,i}(T_e(t), \mathbf{r}_i) \frac{d\mathbf{r}_i}{dt} + \mathbf{R}_{e,i}(T_e(t), \eta_{e,i}(\mathbf{r}_i)). \quad (3.2)$$

Realni sustavi mogu pokazivati znatno kompleksnije ponašanje ili može dominirati direktna interakcija ad molekula s fononskim podsustavom [21] koju ovdje korišteni model također uzima u obzir.

3.2 O podacima korištenim za strojno učenje

Pomoću AIMDEF¹⁰ [22] sheme simuliran je sustav adsorbiranih CO molekula na površini Pd(111) ozračenoj UV laserskim pulsom u slici Langevinove dinamike [10]. Za sve atome u molekulama CO rješava se Langevinova jednadžba 3.2. Sve adijabat-ske sile (derivacije potencijala) se računaju pomoću DFT-a i Hellman-Feynmanovog teorema. Vremenska ovisnost temperatura fononskog i elektronskog podsustava se računa rješavajući sistem jednadžbi 3.1. Stohastička sila $\mathbf{R}_{e,i}$ je modelirana kao gausijanski bijeli šum s varijancom danom fluktuacijsko-disipacijskim teoremom:

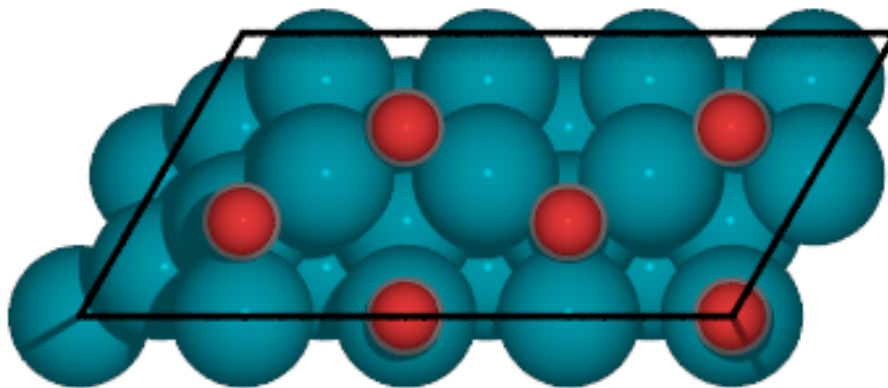
$$\text{Var}(\mathbf{R}_{e,i}(T_e, \eta_{e,i})) = \frac{2k_B T_e(t) \eta_{e,i}(\mathbf{r}_i)}{\Delta t}, \quad (3.3)$$

gdje je Δt vremenski korak simulacije. U svakom koraku simulacije koeficijent trenja $\eta_{e,i}(\mathbf{r}_i)$ se računa unutar LDFA¹¹ [23] aproksimacije, što znači da ovisi o lokalnoj elektronskoj gustoći gole površine na položaju adsorbata. Računa se kao onaj koeficijent trenja kojeg bi imao atom (C ili O) koji se giba u slobodnom elektronskom plinu iste gustoće. Simulirani sustav se sastoji od četiri sloja po 8 atoma Pd(111) (ćelija (4×2)) i 6 adsorbiranih molekula CO (konfiguracija korištena za uvježbavanje mreža) ili

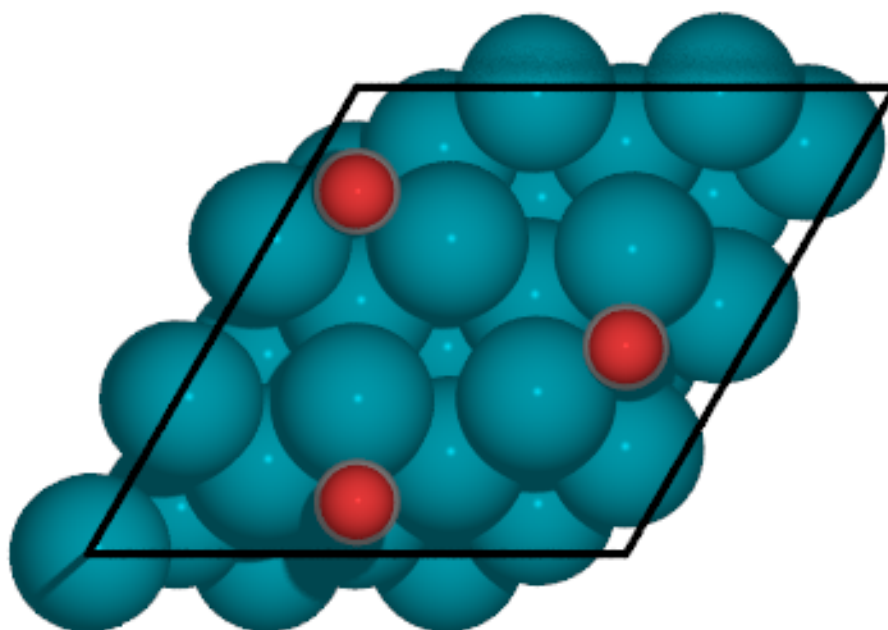
¹⁰Eng. Ab Initio Molecular Dynamics with Electronic Friction

¹¹Eng. Local Density Functional Approximation

četiri sloja po 9 atoma Pd (ćelija (3×3)) s 3 adsorbirane molekule CO. Ove dvije konfiguracije odgovaraju različitim prekrivenostima površine (0.75 ML i 0.33 ML). Sustavi su prikazani na Slikama 3.2 i 3.3. Četvrti sloj Pd (onaj najdalje od površine)



Slika 3.2: Sustav molekula CO adsorbiranih na površinu Pd(111) u (4×2) geometriji.



Slika 3.3: Sustav molekula CO adsorbiranih na površinu Pd(111) u (3×3) geometriji.

se tokom simulacija držao "smrznutim", za trećeg se koristila klasična Newtonova jednadžba gibanja u adijabatskoj aproksimaciji, a efekt lasera na prvi i drugi sloj je uzet spajanjem slojeva na Nosé-Hooverov termostat [24, 25] s vremenski ovisnom temperaturom T_i izračunatom u 2TM modelu [26]. Kalkulacija je implementirana u programskom paketu VASP koristeći DFT i vdW-DF funkcional izmjene i korelacije [27]. Interakcija elektrona i jezgara je tretirana PAW¹² metodom [28].

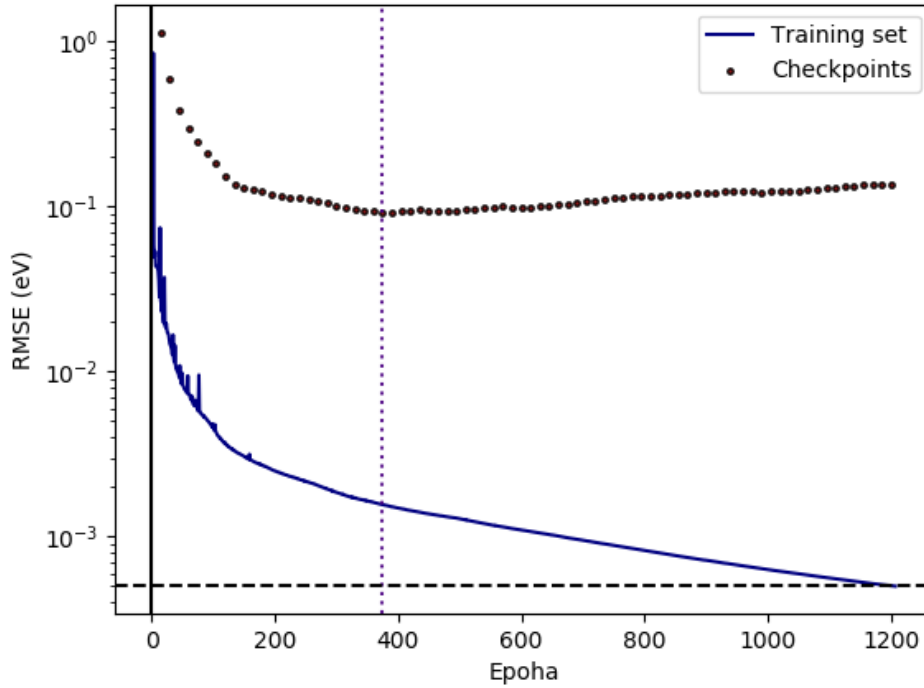
¹²Projector Augmented-Wave

4 Uvježbavanje neuronskih mreža

4.1 Preciznost neuronskih mreža u ovisnosti o strukturi mreže i cutoff parametru

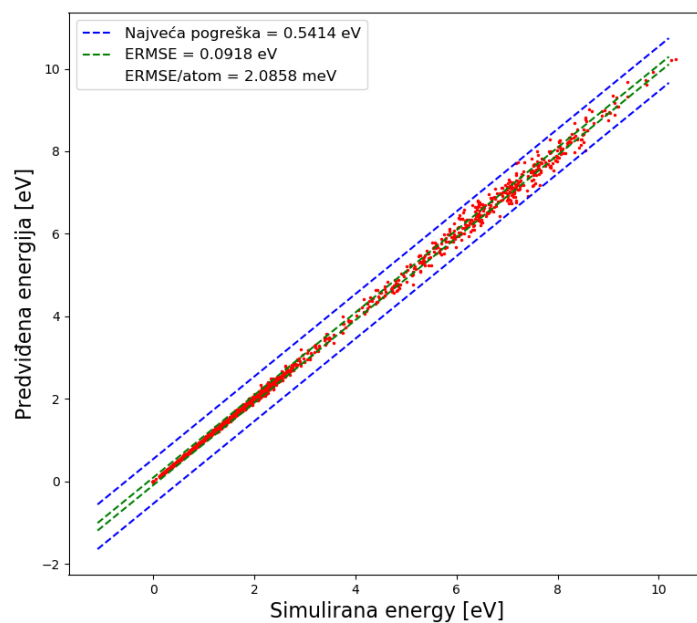
Prvo smo provjerili preciznost predviđanja neuronskih mreža o njihovoj strukturi i *cutoff*-parametru. Dobiveni DFT podaci sadržavali su 465 simulacija (trajektorija) u trajanju od 3.5 ps za različite temperature i početne konfiguracije sustava. Vremenski korak simulacije bio je 1 fs, pa većina trajektorija sadrži 3500 "slika" sustava. Ukupni broj slika je nešto više od $1.6 \cdot 10^6$. Nasumično smo poredali trajektorije, pa iz prvih 300 trajektorija odabrali svaku 201. sliku za skup za uvježbavanje, iz sljedećih 100 trajektorija smo odabrali svaku 201. za validacijski set, a za testni set smo iz 200 nasumičnih trajektorija odabrali svaku 251. sliku, pazeći pritom da ne odaberemo konfiguraciju koju smo već uzeli za skup za uvježbavanje ili validaciju. Tako smo dobili skup za uvježbavanje od 5200 slika, skup za validaciju od 1759 slika te skup za testiranje od 1200 slika. S obzirom na to da dinamika iz koje su podaci dobiveni sadrži nasumične fluktuacijske sile, ovakav odabir rezultira nasumično odabranim slikama koje dobro istražuju konfiguracijski prostor, ali, kao što ćemo kasnije vidjeti, postoje "netipične" točke u konfiguracijskom prostoru čijih slika ima relativno male te je mala vjerojatnost da budu obuhvaćene ovakvim odabirom.

Prije uvježbavanja odabiru se struktura mreže (broj skrivenih slojeva i neurona), simetrijske funkcije (u Amp-u se nazivaju *deskriptorima*), cutoff-funkcija i pripadajući parametri. Koristili smo gausijanske deskriptore (2.4) i (2.6) s kosinusnom cutoff-funkcijom (2.5). Odabrali smo parametre $\eta = (0.05, 0.23, 1.08, 5.0)$ za deskriptore (2.4) te $\eta = 0.005$, $\gamma = (1, -1)$ i $\zeta = (1.0, 4.0)$ za deskriptore (2.6). R_C smo varirali između 2 Å i 6.5 Å. Optimalnu mrežu za zadane ulazne parametre smo pronašli tako da svakih 15 koraka uvježbavanja (epoha) generiramo kopiju trenutne mreže ("*checkpoint*") kojom računamo korijen srednjih kvadratnih odstupanja predviđenih energije od simuliranih (ERMSE), a za optimalnu mrežu uzimamo onu za koju se postigne globalni minimum ERMSE na validacijskom setu. Na slici 4.1 prikazan je primjer uvježbavanja za mrežu 30 – 20 – 10 – 10, (30 skrivenih neurona u prvom skrivenom sloju, 20 u drugom, ...) uz $R_C = 4.25$ Å. Usporedbe predviđenih i simuliranih energija na validacijskom i testnom setu za checkpoint na kojem je postignut

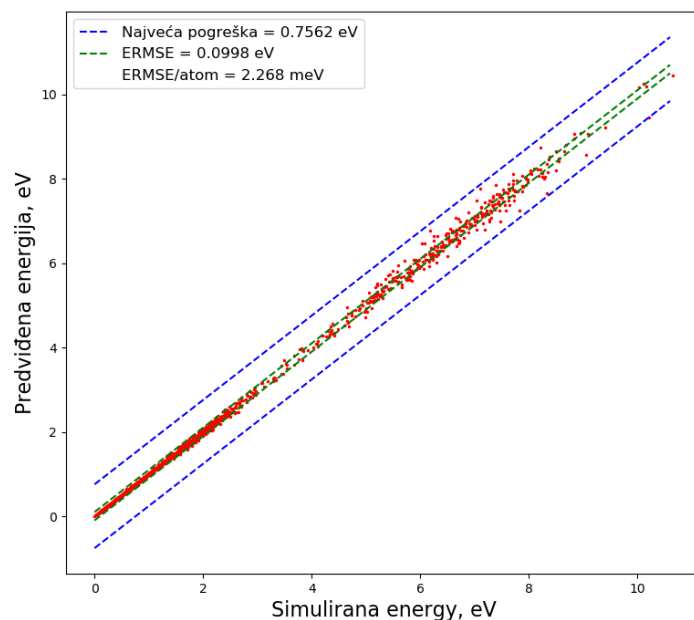


Slika 4.1: Uvježbavanje mreže $30 - 20 - 10 - 10$ uz $R_C = 4.25 \text{ \AA}$. Uvjet konvergencije smo postavili na ERMSE od $0.5 \cdot 10^{-3} eV$ na skupu za uvježbavanje (horizontalna crkana linija). Globalni minimum ERMSE na validacijskom setu označen je točkastom vertikalnom linijom na epohi 375. Ta epoha odgovara ERMSE od $1.56 \cdot 10^{-3} eV$ na skupu za uvježbavanje.

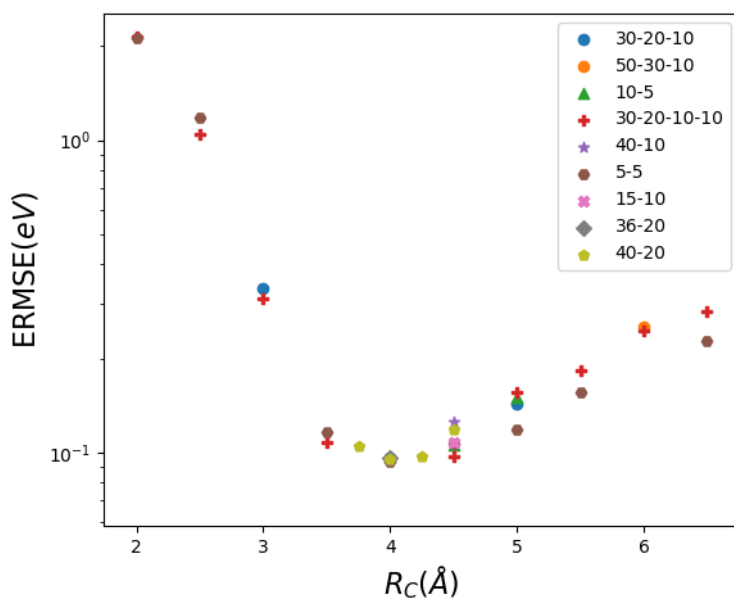
globalni minimum validacijskog seta prikazani su na Slikama 4.2 i 4.3. Dobiva se da je za testni set ERMSE po atomu $2.26 meV/atom$. Istu proceduru ponovili smo za različite mreže i vrijednosti parametra R_C . Rezultati na skupu za testiranje su prikazani na Slici 4.4. U okolini $R_C \approx 4 \text{ \AA}$ sve mreže se ponašaju slično. Najmanji pronađeni ERMSE s ovim skupovima za uvježbavanje, validaciju i testiranje iznosi $2.11 meV/atom$ za mrežu 5-5 i $R_C = 4.0 \text{ \AA}$. S obzirom na to da postoji globalni minimum za R_C , zaključujemo da ovako odabrani deskriptori ne opisuju dobro interakcije među atomima na udaljenosti većoj od $R_C \approx 4 \text{ \AA}$. Bitan zaključak razmatranja u ovom paragrafu je da preciznost uvježbane mreže ne ovisi jako o njoj strukturi, nego je samo važno da je mreža dovoljno velika (sadržava dovoljan broj slobodnih parametara) da zadovolji univerzalni aproksimacijski teorem.



Slika 4.2: Usporedba točnih (simuliranih) i predviđenih energija na validacijskom setu za mrežu $30 - 20 - 10 - 10$ uz $R_C = 4.25 \text{ \AA}$ i uvjet konvergencije $1.56 \cdot 10^{-3} \text{ eV}$. Nultočka energije je za potrebe crteža odabrana tako da energije padaju u interval $0 - 11 \text{ eV}$.



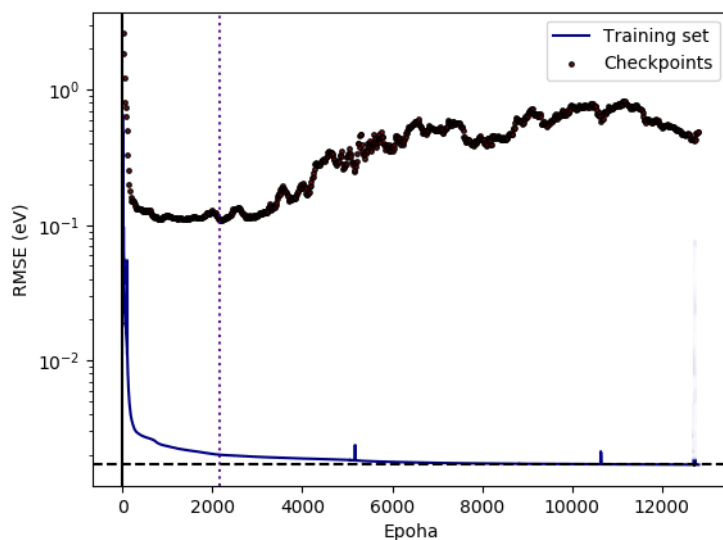
Slika 4.3: Usporedba točnih (simuliranih) i predviđenih energija na testnom setu za mrežu $30 - 20 - 10 - 10$ uz $R_C = 4.25 \text{ \AA}$ i uvjet konvergencije $1.56 \cdot 10^{-3} \text{ eV}$. Nultočka energije je za potrebe crteža odabrana tako da energije padaju u interval $0 - 11 \text{ eV}$.



Slika 4.4: Ovisnost ERMSE o R_C za različite mreže.

4.2 Komentar o cutoff-parametru

Ako prilikom odabira ulaznih parametara odaberemo premalen *cutoff*, validacija mreže će izgledati kao na slici 4.5, odnosno validacijska krivulja će imati više lokalnih minimuma. Ovaj efekt je najizraženiji za male mreže, ali je prisutan bez obzira na



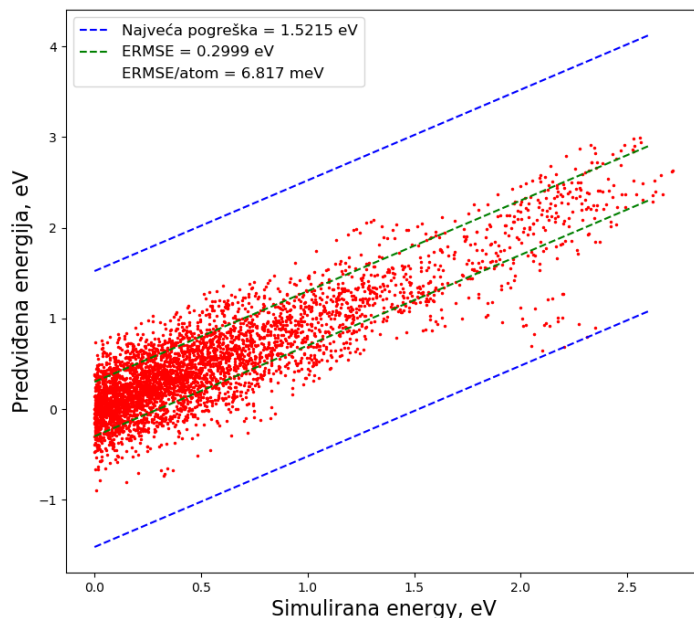
Slika 4.5: Validacija mreže 5 – 5 s $R_C = 3.5 \text{ \AA}$. Okomitom točkastom linijom označen je globalni minimum greške na validacijskom setu. NN ne može umanjiti grešku na skupu za uvježbavanje ispod vrijednosti označenoj vodoravnom crtkanom linijom.

veličinu mreže. Ovo se zapravo događa za mreže za koje ne vrijedi univerzalni aprok-

simacijski teorem: mreža nije u stanju proizvoljno umanjiti pogrešku na skupu za uvježbavanje i varijacija parametara dovodi do nasumičnih fluktuacija validacijske krivulje.

4.3 *Uvježbavanje na skupovima s više slika*

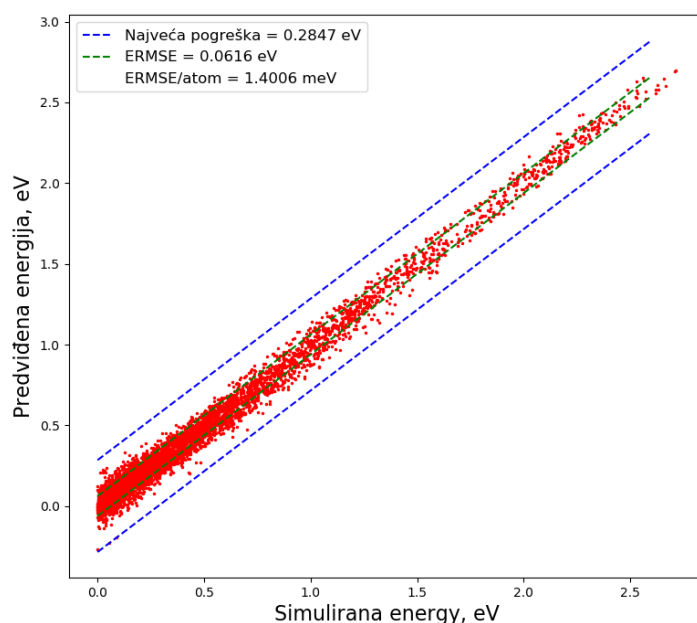
Prilikom odabirom slika na kojima se mreže uvježbavaju i testiraju, valja biti oprezan da odabrane slike dobro obuhvaćaju što veći dio konfiguracijskog prostora, inače se može dogoditi da mreža vrlo loše predviđa energije u režimu koji nije reprezentiran u skupu za uvježbavanje. Primjer toga prikazan je na Slici 4.6; korištena je mreža uvježbana kako je opisano u odjeljku 4.1 te su testirana njena predviđanja na konfiguracijama potencijalne energije veće od -2.32 eV/atom. Takvih slika ima samo 4265, odnosno sačinjavaju manje od 0.3% konfiguracija izračunatih DFT-om pa je mala vjerojatnost da ćemo ih nasumičnim odabirom slika uključiti u skupove za uvježbavanje. Kako bi doskočili tom problemu, povećali smo skupove za uvježbavanje i validaciju



Slika 4.6: Testiranje mreže 40 – 20 s $R_C = 4.25$ Å uz uvjet konvergencije $1.3769 \cdot 10^{-03}$ na slikama energije veće od -2.32 eV/atom.

na sljedeći način: iz prvih 200 trajektorija uzeli smo svaku 39. sliku, iz 300. – 350. trajektorije smo uzeli svaku 7. sliku te smo iz svih trajektorija izabrali sve slike potencijalne energije veće od -2.34 eV/atom, budući da su dosadašnje mreže pokazivale

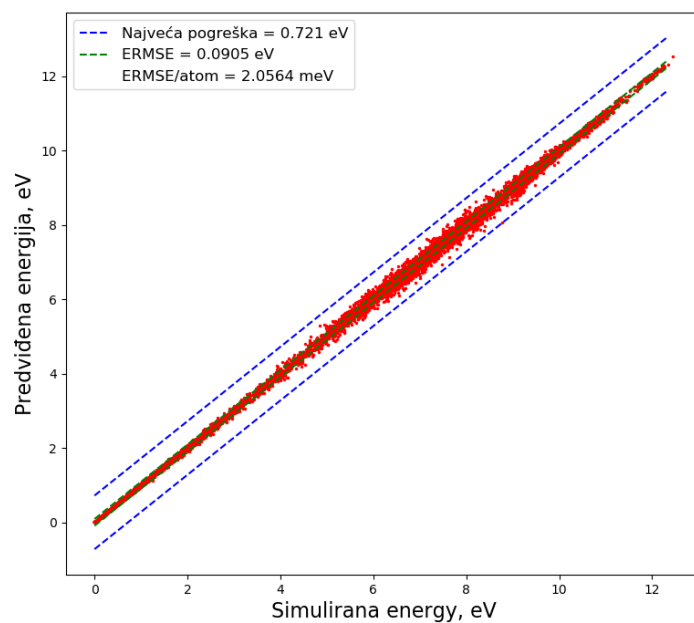
najlošije performanse na konfiguracijama visoke energije. Ovakav odabir rezultirao je s ukupno 54800 slika. Dobivene konfiguracije smo nasumično poredali te smo uvježbali mrežu na skupu od prvih 35200 slika, validirali smo je pomoću sljedećih 8800 slika, a testirali na skupu od 10800 slika. Performanse na istom skupu kao na Slici 4.6 prikazane su na Slici 4.7 (treba napomenuti da su realne performanse vjerojatno nešto lošije jer novi skup za uvježbavanje možda sadrži neke od ovih slika pa ovakav test nije sasvim tehnički ispravan). Test iste mreže na skupu od 10800 slika prikazan je na Slici 4.8. Ovu mrežu smo koristili u nastavku rada.



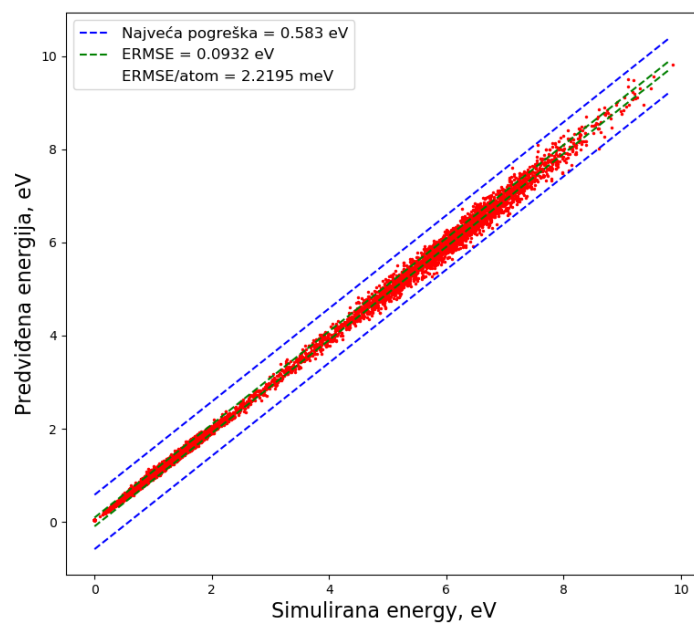
Slika 4.7: Testiranje mreže $40 - 20$ s $R_C = 4.25 \text{ \AA}$ uz uvjet konvergencije $1.9268 \cdot 10^{-3}$ na slikama energije veće od -2.32 eV/atom . U odnosu na Sliku 4.6 su skupovi za uvježbavanje i validaciju povećani.

4.4 Testiranje mreže na (3×3) geometriji

Za uvježbavanje mreža koristili smo samo konfiguracije u (4×2) geometriji. U ovom odjeljku ćemo prikazati predviđanja neuronskih mreža na geometriji (3×3) . Od dostupnih 362 trajektorije od kojih svaka sadržava prosječno 3500 slika odabrali smo svaku 151. sliku, rezultirajući sa skupom za testiranje od 8688 slika. Predviđanje je prikazano na Slici 4.9. Usprkos tome što u skupu za uvježbavanje nije korištena nijedna slika (3×3) geometrije, predviđanje je precizno zato što smo koristili atomski



Slika 4.8: Testiranje mreže $40 - 20$ s $R_C = 4.25 \text{ \AA}$ uz uvjet konvergencije $1.9268 \cdot 10^{-3}$ na 10800 slika. Vidljivo je poboljšanje ERMSE/atom u odnosu na optimalnu mrežu iz odjeljka 4.1.



Slika 4.9: Predviđanje energija na (3×3) geometriji.

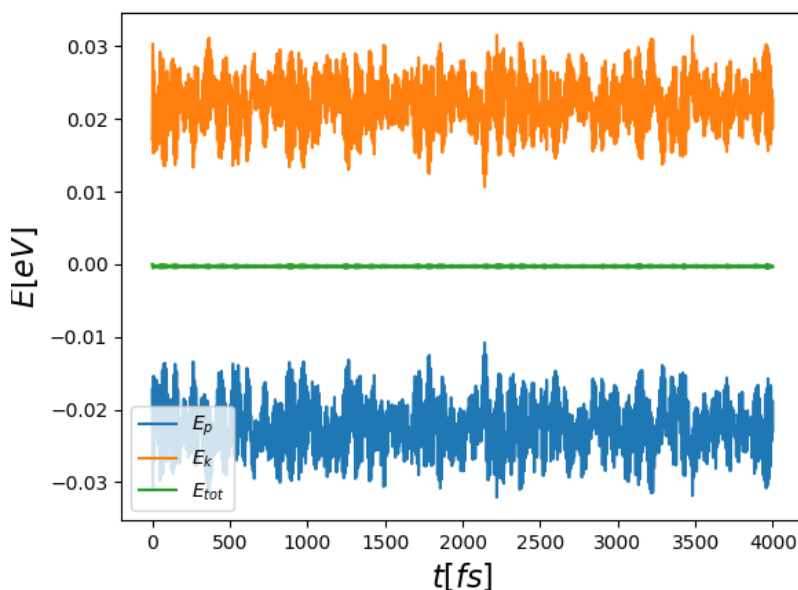
centrirane simetrijske funkcije (jednadžba 2.3). Treba napomenuti da bi se reproducirale DFT energije za ovu geometriju atomskim energijama treba dodati 19.4017 eV .

Razlog je što u DFT proračunima napravljenima u VASP-u nula potencijalne energije definirana u ovisnosti o veličini sustava. Budući da izbor nule potencijalne energije nema fizikalnog značaja, neuronska mreža zadovoljavajuće predviđa energije za sustave različitog broja čestica.

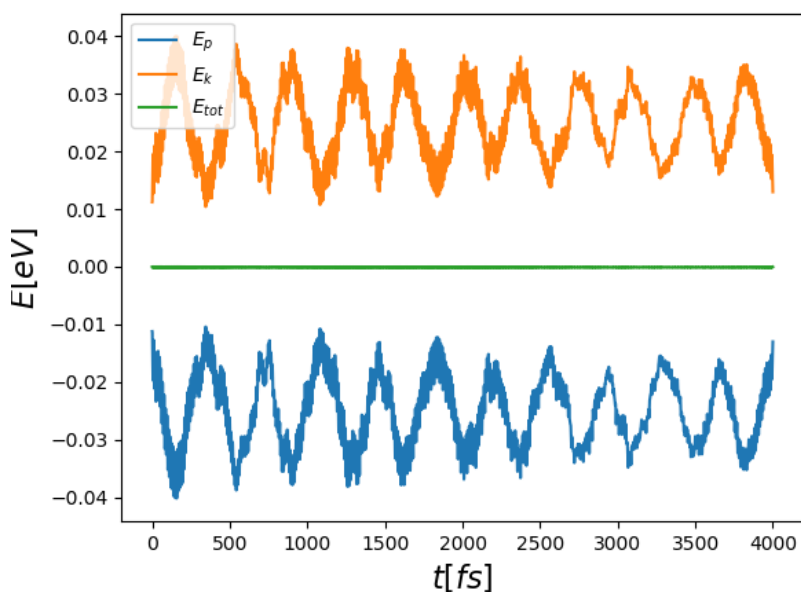
5 Molekularna dinamika

5.1 NVE dinamika

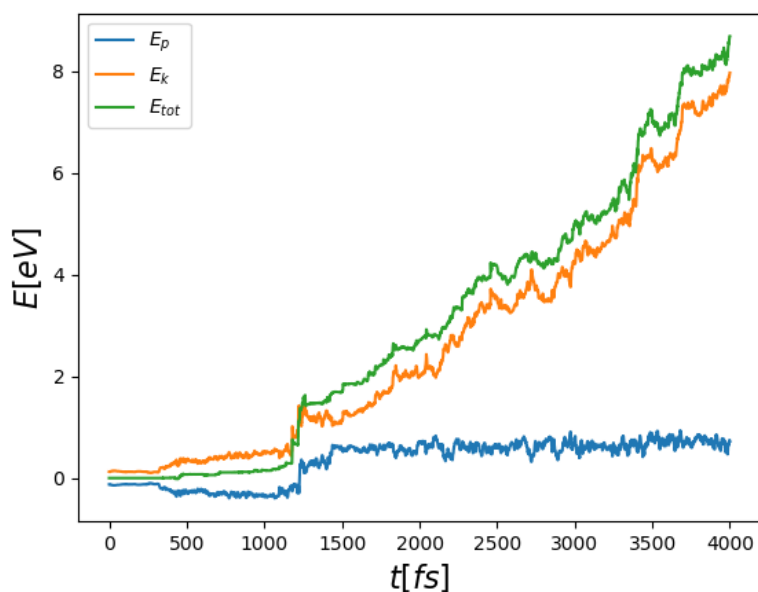
Prvo smo testirali konzistentnost energija i sila koje predviđa neuronska mreža provodeći simulaciju pri konstantnom broju čestica, volumenu i energiji sustava (tzv. NVE dinamika). Koristili smo implementaciju NVE dinamike u programskom paketu ASAP [29] koji se koristi kao kalkulator za molekularno-dinamičke simulacije unutar ASE-a. Prednost ASAP-a je to što omogućava paralelizaciju proračuna pa se simulacije mogu znatno ubrzati. Dinamika je implementirana pomoću *Velocity Verlet* algoritma [30], metode za numeričku integraciju klasičnih Newtonovih jednadžbi gibanja. Simulacije smo napravili za različite temperature (izračunate kao da su atomi idealni plin) i geometrije. Vremenski korak dinamike bio je $\Delta t = 1$ fs. Na Slici 5.1 i 5.2 vidimo da je energija očuvana, kako i treba biti. Nasuprot tome, na Slici 5.3, a donekle i na Slici 5.4 energija nije očuvana. Razlog tome je što su s povećanom temperaturom i fluktuacije kinetičke energije veće pa greška brzine (koja se u *Velocity Verlet* algoritmu skalira s Δt^2) postaje značajna; posljedica toga je da umjesto da vidimo male oscilacije oko ukupne energije, vidimo njeno gotovo konstantno povećavanje. Smanjenjem koraka simulacije na $\Delta t = 0.1$ fs možemo umanjiti ovaj efekt, kao što vidimo na Slici 5.5 u usporedbi sa Slikom 5.3.



Slika 5.1: NVE dinamika za (4×2) geometriju. Početne brzine atoma odgovaraju temperaturama $T_{ads} = 300.5$ K i $T_{latt} = 94.8$ K.

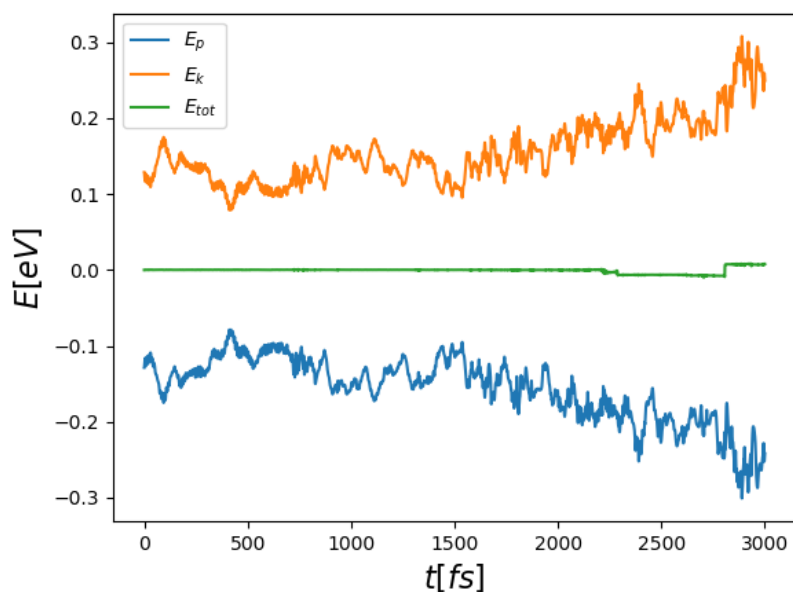


Slika 5.2: NVE dinamika za (3×3) geometriju. Početne brzine atoma odgovaraju temperaturama $T_{ads} = 199.88$ K i $T_{latt} = 87.2$ K.

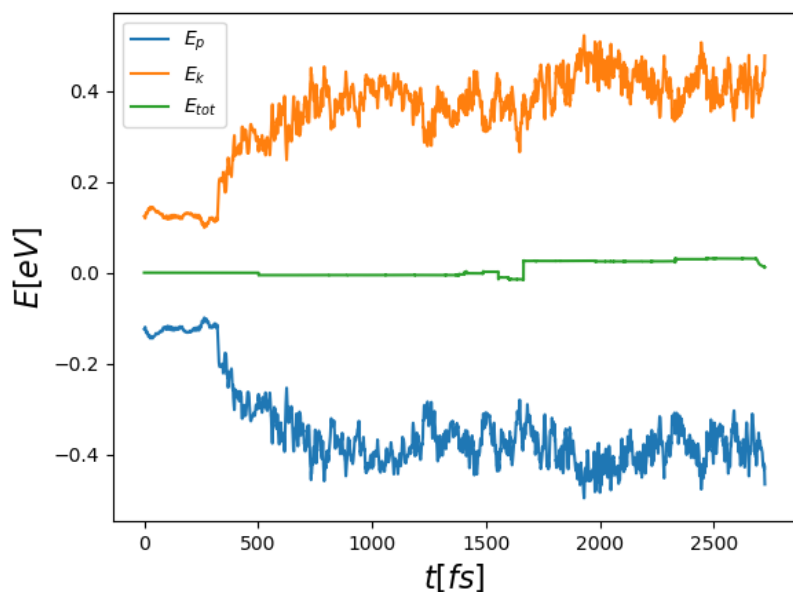


Slika 5.3: NVE dinamika za (4×2) geometriju. Početne brzine atoma odgovaraju temperaturama $T_{ads} = 1215.32$ K i $T_{latt} = 1155.22$ K.

Iako neuronska mreža predviđa konzistentne energije i sile, neka ograničenja uvježbane neuronske mreže možemo lako pronaći. Primjerice, na Slici 5.6 prikazana je ovisnost potencijala o udaljenosti centra mase jedne molekule CO od površine. Očekivali



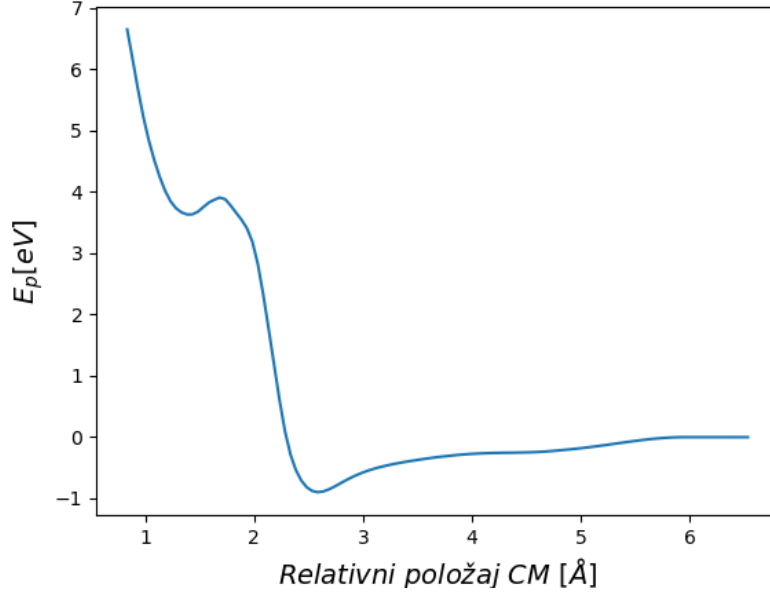
Slika 5.4: NVE dinamika za (3×3) geometriju. Početne brzine atoma odgovaraju temperaturama $T_{ads} = 1219.3$ K i $T_{latt} = 1231.69$ K.



Slika 5.5: NVE dinamika za (4×2) geometriju. Početne brzine atoma odgovaraju temperaturama $T_{ads} = 1215.32$ K i $T_{latt} = 1155.22$ K (isto kao i Slika 5.3, ali je vremenski korak $\Delta t = 0.1$ fs.)

bismo potencijal tipa Lennard-Jones potencijala, odnosno odbojan potencijal na malim udaljenostima, ali, kao što se vidi na slici, postoji i privlačni dio potencijala za udaljenosti $1.3 - 1.7$ Å. Pri visokim temperaturama, ako molekule imaju dovoljno energije, može se dogoditi ulazak u ovakav nefizikalni lokalni minimum. Rješenje

ovog problema je izračunati energiju takvih konfiguracija u DFT-u te ih dodati u skup za uvježbavanje neuronske mreže.



Slika 5.6: Potencijalna energija u ovisnosti o udaljenosti centra mase jedne molekule CO od površine. Sve atome osim onih u promatranoj molekuli zadržali smo fiksiranima.

5.2 Vremenski ovisna Langevinova dinamika

Simulirali smo dinamiku molekula na površini ozračenoj femtosekundnim laserskim pulsom kako je opisano u odjeljku 3.1. Duljina simulacija bila je 3500 fs s vremenskim korakom od 1 fs. Rješavanjem sustava jednadžbi 3.1 dobiven je niz vremenski ovisnih fononskih i elektronskih temperatura T_{ph} i T_{el} koje su interpolirane *interp1d* metodom iz SciPy paketa [31]. Na isti način je interpolirana i prostorno ovisna elektronska gustoća koju stvara atom Pd (ta gustoća nam biti potrebne za izračun koeficijenta trenja adsorbata). Navedena elektronska gustoća je izračunata pomoću DFT-a. Četvrti sloj atoma Pd (onaj najdalje od površine) tijekom simulacije držali smo fiksiranim. Za preostala tri sloja rješavana je Langevinova jednadžba:

$$M_i \frac{d^2 \mathbf{r}_i}{dt^2} = -\nabla_{\mathbf{r}_i} V(\mathbf{r}_1, \dots, \mathbf{r}_N) - \eta_{ph} \frac{d\mathbf{r}_i}{dt} + \mathbf{R}_{ph,i}(T_{ph}(t), \eta_{ph}), \quad (5.1)$$

gdje je N ukupni broj atoma u ćeliji, η_{ph} je konstantni koeficijent trenja za čiju smo vrijednost uzeli 0.2 u ASE-ovim jedinicama [32], a stohastička sila $\mathbf{R}_{ph,i}(T_{ph}(t), \eta_{ph})$

je modelirana kao gausijanski bijeli šum s varijancom analognom jednadžbi 3.3. Za atome u molekulama CO rješavana je jednadžba 3.2, uz napomenu da smo koeficijent trenja računali u svakom koraku simulacije računali prema izrazima:

$$\begin{aligned}\eta_C &= 22.654 \cdot r_S^{-1.8284} e^{-3.134 \cdot r_S} + 2.497 \cdot r_S^{-2.061} \cdot e^{0.0793 \cdot r_S} \\ \eta_O &= 1.36513 \cdot r_S^{-1.8284} e^{-0.0820301 \cdot r_S} + 50.342 \cdot r_S^{0.490785} \cdot e^{-2.70429 \cdot r_S},\end{aligned}\quad (5.2)$$

gdje je r_S Wigner-Seitzov radijus:

$$r_S = \left(\frac{3}{4\pi n(\mathbf{r})} \right)^{1/3}, \quad (5.3)$$

a $n(\mathbf{r})$ je elektronska gustoća na položaju \mathbf{r} atoma C ili O čiji su izvor svi atomi Pd. Dakle, zamišlja se da su adsorbati uronjeni u elektronsku gustoću stvorenu od atoma površine te da je ta elektronska gustoća izvor disipacija (fluktuacija) adsorbata. Sami izrazi 5.2 dobiveni su prilagodbom funkcije s pozitivnih šest slobodnih parametara a_1, \dots, a_6 :

$$\eta(r_S) = a_1 r_S^{a_2} e^{-a_3 r_S} + a_4 r_S^{-a_5} e^{a_6 r_S} \quad (5.4)$$

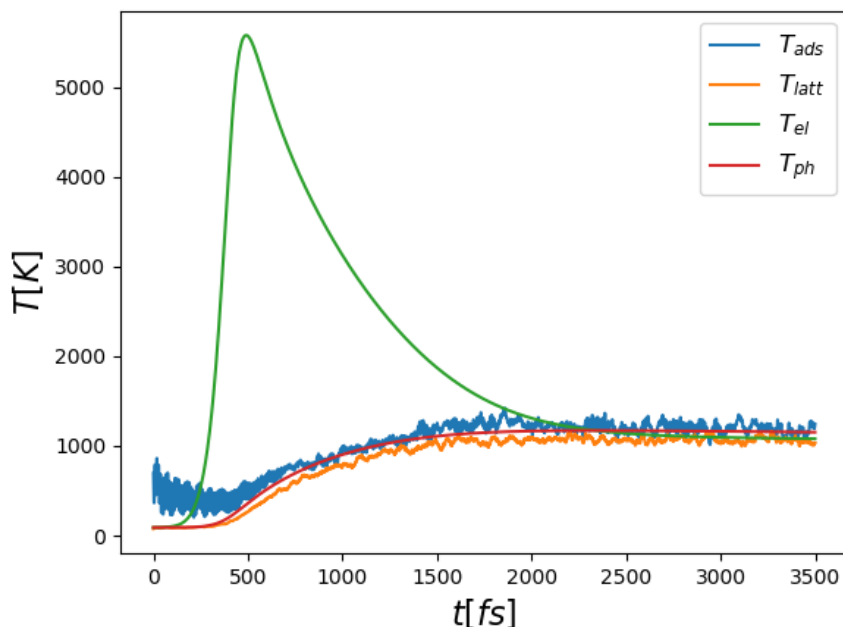
na točke izračunate za različite vrijednosti n_0 prema jednadžbi:

$$\eta = \frac{4\pi n_0}{k_F} \sum_{l=0}^{\infty} (l+1) \sin^2[\delta_l(k_F) - \delta_{l+1}(k_F)], \quad (5.5)$$

gdje opet imamo $r_S = [3/(4\pi n_0)]^{1/3}$, a k_F je Fermijev impuls te je $\delta_l(k_F)$ fazni pomak u raspršenju na Fermijevom nivou [23]. Jednadžbe gibanja su integrirane prema shemi danoj u referenci [33]. Početne položaje i brzine smo preuzeli iz POSCAR datoteka korištenih u DFT proračunima opisanima u odjeljku 3.2. Brzine atoma površine su fiksirane pomoću Nosé-Hooverovog termostata tako da odgovaraju temperaturi od 90 K (kad bi atome promatrali kao idealni plin). Iz te termalizacijske dinamike su odabrani nasumične konfiguracije te su njihovi početni položaji i brzine zapisani u POSCAR datoteke. Za atome adsorbata izračunate su frekvencije vibracijskih normalnih modova te im je potom zadana energija nulte točke¹³ uzduž svakog moda. Dinamiku smo računalno implementirali u ASE-u tako što smo modificirali klasu za

¹³Eng. Zero Point Energy

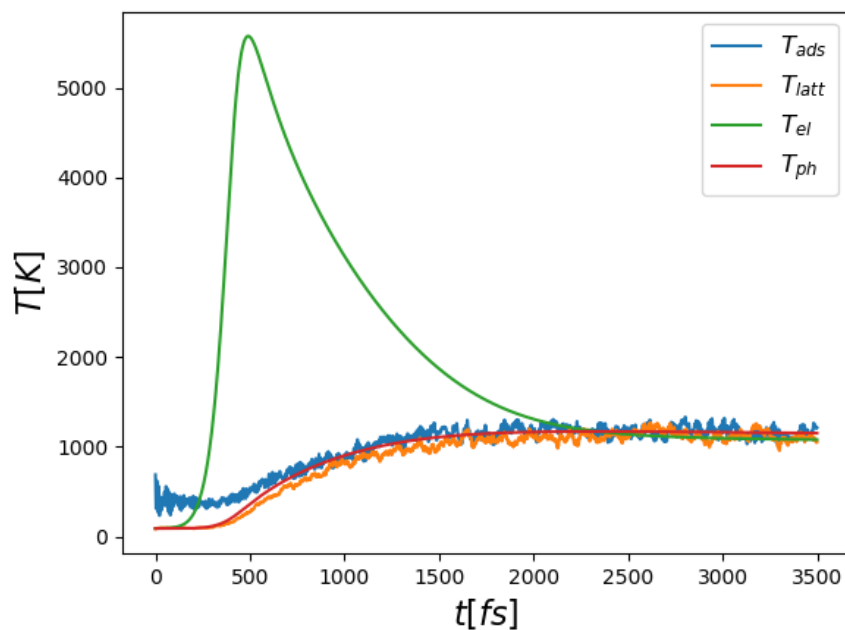
Langevinovu molekularnu dinamiku pri konstantnoj temperaturi i koeficijentima trenja (vidi Dodatak B).



Slika 5.7: Prosječna temperatura adsorbiranih molekula CO, površine Pd(111) te elektronskog i fononskog podsustava za (3×3) geometriju.

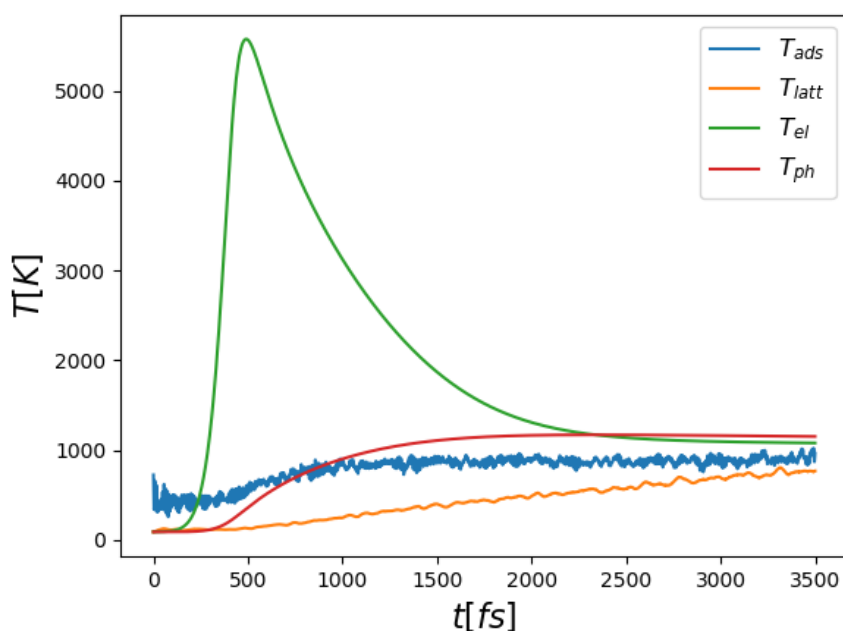
Prema razmatranjima iz prethodnog odjeljka vidimo da je s vremenskim korakom $\Delta t = 1$ fs pouzdanije dinamiku možemo provoditi za (3×3) geometriju. Hipoteza (koju treba egzaktno provjeriti) je da je razlog tome to što u (4×2) geometriji više do izražaja dolaze interakcije među adsorbiranim molekulama (zbog veće prekrivenosti površine) te da postoji nedostatak u naučenom potencijalu, sličan onom prikazanom na Slici 5.6, pa postoji vjerojatnost ulaska u nefizikalni režim, veća nego za (3×3) geometriju. Na Slici 5.7 prikazana je prosječna temperatura adsorbiranih molekula CO, površine Pd(111) te elektronskog i fononskog podsustava usrednjena preko 30 simulacija za geometriju (3×3) .

Slika 5.8 je analogna, samo se odnosi na (4×2) geometriju i usrednjena je preko 19 simulacija. U oba slučaja vidimo da temperature adsorbata i površine prate fononsku krivulju, to jest je vezanje adsorbiranih molekula s pobuđenim elektronskim podsustavom slabo te dominira interakcija adsorbiranih molekula s vibracijama površine. Vrijedi promotriti i simulaciju s nižim faktorom trenja za atome površine $\eta_{ph} = 3 \cdot 10^{-3}$. Prosjek preko 19 takvih simulacija prikazan je na Slici 5.9: vidimo da



Slika 5.8: Prosječna temperatura adsorbiranih molekula CO, površine Pd(111) te elektronskog i fononskog podsustava za (4×2) geometriju.

se površina vrlo sporo termalizira s fononskim podsustavom te je kinetička energija adsorbata manja u odnosu na simulacije s većim faktorom trenja površine; to smo i očekivali. Slika je sljedeća: η_{ph} je mjera vezanja površinskih atoma na laserski puls, pa oni zadobivaju manje kinetičke energije iz pobuđenog fononskog podsustava i ne prilaze blizu atomima adsorbata te je tako smanjena elektronska gustoća koju vide molekule. Efektivno smo znatno smanjili vezanje adsorbata na pobuđeni fononski podsustav metalne površine.



Slika 5.9: Prosječna temperatura adsorbiranih molekula CO, površine Pd(111) te elektronskog i fononskog podsustava za (4×2) geometriju i nizak faktor trenja površine.

6 Zaključak

Promotрили smo sustav molekula CO adsorbiranih na površinu Pd(111). Uvježbavanjem neuronskih mreža različitih struktura došli smo do zaključka da preciznost predviđanja slabo ovisi o njihovoj veličini, nego je jedino važno da su neuronske mreže dovoljno velike da zadovoljavaju univerzalni aproksimacijski teorem. Potvrdili smo da je potencijalna ploha strojno naučena pomoću atomski centriranih simetrijskih funkcija adekvatna i za opis sustava različitog broja čestica od onog u skupovima za uvježbavanje. Napisali smo klasu unutar ASE-a za vremenski ovisnu Langevinovu dinamiku koja modelira dinamiku sustavu induciranu femtosekundnim laserskim pulsom te za izvršavanje energija i sila koristili strojno naučenu potencijalnu plohu. Zaključili smo da se adsorbirane molekule slabo vežu na pobuđeni elektronski podsustav te da energiju iz laserskog pulsa primaju prvenstveno preko interakcije s molekulama površine.

Zaključno, neuronske mreže omogućavaju znatno povećanje vremenskih i prostornih skala preko kojih se mogu izvršavati dinamike kompleksnih sustava. Njihov glavni nedostatak je što obično loše rade u djelu konfiguracijskog prostora "neis-

traženom” u skupovima za uvježbavanje. Bez obzira na to, s pravilnim odabirom konfiguracijskog prostora, moguće je znatno skratiti ukupno vrijeme potrebno za proučavanje dinamike nekog sustava, budući je da pri takvim dinamikama većina vremena ”potrošena” na DFT proračune, a njih broj za iste konačne rezultate se korištenjem neuronskih mreža može umanjiti za nekoliko redova veličine; u našem slučaju, s nekoliko milijuna na tek nekoliko desetaka tisuća.

Dodaci

Dodatak A Povjesna crtica o heterogenoj katalizi

Fizikalna istraživanja klase fenomena danas poznatih pod nazivom *heterogene katalize* započeta su tokom 19. stoljeća. Radi se o procesima u kojima prisustvo neke tvari (katalizatora) potiče ili ubrzava kemijsku reakciju među nekim drugim tvarima (reaktantima). Riječ heterogena znači da su agregatne faze katalizatora i reaktanta različite. Jedno od prvih značajnih otkrića je ono Humphrya Davya 1817. godine [34]: prilikom istraživanja zapaljivih plinovitih smjesa u svrhu unaprjeđenja rudarske sigurnosne svjetiljke¹⁴, Davy je otkrio da se umetanjem platinaste žice u komoru ispunjenu ugljenim plinom i zrakom na temperaturi puno manjoj od temperature zapaljenja ugljenog plina, žica zagrije do usijanja, a plin izgubi svojstvo zapaljivosti. Značajan korak prema unificiranom objašnjenju Davyevog i sličnih eksperimenata napravio je Michael Faraday [35]. Faraday je 1847. sustavno proveo niz eksperimenata konverzije plinova vodika i kisika u dodiru s površinom platine u vodenu paru. Utvrdio je da je jedini nužan uvjet za aktivnost platine u katalizi reakcije to da je njena površina *”savršeno čista i metalična”* te je pronašao slično ponašanje u poroznim spužvastim platinama i drugim metalima, primjerice zlatu i paladiju. Tako je naslutio da se radi o nekom *strukturnom* svojstvu, inherentnom površini, koje u određenoj mjeri posjeduju svi metali i općenito kruta tijela. Postulirao je interakciju među česticama plinova te među česticama plina i površine koja uzrokuje stalnu izmjenu molekula između plina i adsorbiranog sloja, to jest da molekule plinova bivaju privučene vrlo blizu metalnoj površini, na kojoj se, zahvaljujući privlačnoj međusobnoj interakciji, pronađu i kombiniraju, a nastala se molekula zbog slabe interakcije s površinom u konačnici desorbira u vodenu paru. Na tragu Faradayeve intuicije, mikroskopski mehanizam iza heterogene katalize prvi je ispravno razumio Irving Langmuir 1922. godine u okviru kinetičke teorije plinova [36]. Bilo je poznato da je brzina reakcije $2\text{CO} + \text{O}_2 = 2\text{CO}_2$ na površini platine proporcionalna tlaku kisika i temperaturi platine, a obrnuto proporcionalna tlaku ugljikovog monoksida. Ključna Langmuirova misao je da je metalna površina uređena struktura, na kojoj postoji prebrojiv skup točaka u koje se mogu adsorbirati molekule plina. Ako

¹⁴Danas poznata kao Davyjeva svjetiljka.

se u njih adsorbira molekula CO, ona se na površini "orijentira" nepovoljno za reakciju s molekulama plina O₂. Ovaj mikroskopski mehanizam djelovanja adsorbiranog monoatomskog sloja CO kao katalitičkog otrova objašnjava ovisnost brzine reakcije o tlaku plina CO i temperaturi platine, jer je s višim tlakom i nižom temperaturom u prosjeku više molekula CO adsorbirano na površini. S druge strane, adsorbirani kisik je vrlo reaktivan s plinom CO te formira tako čvrste veze s atomima platine da ne desorbira s površine čak ni na 1500 kelvina. Langmuir je 1932. dobio Nobelovu nagradu za otkrića u površinskoj kemiji. Tako su otvorena vrata modernim teorijskim i eksperimentalnim istraživanjima površina i adsorbiranih filmova, koja nastoje razumjeti cjelovitu kvantno-mehaničku sliku dinamike sustava uzimajući u obzir što više relevantnih mikroskopskih stupnjeva slobode.

Dodatak B Python kod za vremenski ovisnu Langevinovu dinamiku

```
1 """LangevinLaser dynamics class."""
2
3 from ase.md.md import MolecularDynamics
4 from ase.parallel import world
5 from ase import units
6 from ase.constraints import FixAtoms
7
8 from scipy import interpolate
9 import numpy as np
10
11
12 class LangevinLaser(MolecularDynamics):
13     """Class for Langevin molecular dynamics with time dependent
14     temperatures and adsorbate frictions.
15     Adapted from the ASE Langevin class (see https://wiki.fysik.dtu.dk/ase/).
16
17     Usage: Langevin(atoms, timestep, temperature, density,
18     lattice_friction, simulation_number)
19
20     atoms
21         The list of atoms.
22     timestep
23         The time step.
24     temperature
25         String; name of file which contains time versus electronic
26     temperatures versus phonon temperatures.
27     density
28         String; name of file which contains distance from lattice atom
29     versus electron density.
30     lattice_friction
31         Constant friction for lattice atoms
32     simulation_number
33         A label which will be used in naming the temperature log file
34     fixcm
35         If True, the position and momentum of the center of mass is
```



```

32     kept unperturbed.  Default: True.
33     rng
34     Random number generator, by default numpy.random.  Must have a
35     standard_normal method matching the signature of
36     numpy.random.standard_normal.
37
38     RATTLE constraints can be used with these propagators, see:
39     E. V.-Eijnden, and G. Ciccotti, Chem. Phys. Lett. 429, 310 (2006)
40     The propagator is Equation 23 (Eq. 39 if RATTLE constraints are used)
41     of the above reference.  That reference also contains another
42     propagator in Eq. 21/34; but that propagator is not quasi-symplectic
43     and gives a systematic offset in the temperature at large time steps.
44
45     This dynamics accesses the atoms using Cartesian coordinates.”””
46
47     # Helps Asap doing the right thing.  Increment when changing stuff:
48     _lgv_version = 3
49
50     def __init__(self, atoms, timestep, temperature, density,
51     lattice_friction, simulation_number, fixcm=True,
52     trajectory=None, logfile=None, loginterval=1,
53     communicator=world, rng=np.random):
54
55         """ The number of POSCAR file with initial conditions """
56         self.simulation_number = str(simulation_number)
57
58         """ How many lattice atoms are frozen """
59         self.frozen_lattice = 8.
60
61         """ How often should the height of molecules be checked, on what
62         height to freeze """
63
64         self.freeze_interval = 50
65         self.freeze_height = 17.0
66         self.freeze_counter = 0
67
68         """ Load timestep and C and O masses in atomic units """
69
70         self.timestep = timestep
71         self.C_mass = 12.

```

```

70     self.O_mass = 16.
71
72     """ Read from the temperature file into 1D arrays and generate
interpolated functions """
73
74     temperature_file = np.loadtxt(temperature)
75     temp_t = temperature_file[:, 0] * units.fs
76     temp_el = temperature_file[:, 1] * units.kB
77     temp_ph = temperature_file[:, 2] * units.kB
78
79     self.interpolated_el_temp = interpolate.interp1d(temp_t, temp_el,
kind='cubic')
80     self.interpolated_ph_temp = interpolate.interp1d(temp_t, temp_ph,
kind='cubic')
81
82     """ Read from the density file into 1D arrays, generate
interpolated function and get cutoff """
83
84     density_file = np.loadtxt(density)
85
86     r_array = density_file[:, 0]
87     n_array = density_file[:, 1]
88
89     self.interpolated_density = interpolate.interp1d(r_array, n_array
, kind='cubic')
90
91     self.cutoff = r_array[-1]
92
93     """Make empty arrays for the coefficients """
94
95     self.natoms = atoms.get_number_of_atoms()
96     self.c1 = self.c2 = self.c3 = self.c4 = self.c5 = np.zeros(self.
natoms)
97
98     """ Get indices of adsorbate and surface atoms """
99
100    self.symbols = np.asarray(atoms.get_chemical_symbols(), dtype=st
r
)
101
102    self.C_indices = np.flatnonzero(self.symbols == 'C')

```

```

103     self.O_indices = np.flatnonzero(self.symbols == 'O')
104     self.lattice_indices = np.flatnonzero(self.symbols == 'Pd')
105     self.adsorbate_indices = np.concatenate((self.C_indices, self.
O_indices))
106     self.adsorbate_number = len(self.adsorbate_indices)
107     self.lattice_number = len(self.lattice_indices)
108
109     """ Make empty arrays for atom temperatures and atom frictions.
"""
110
111     self.T = np.zeros(self.natoms)
112     self.friction = np.zeros(self.natoms)
113
114     """ Set friction for surface atoms immediately since it is
constant during the simulation """
115
116     np.put(self.friction, self.lattice_indices, lattice_friction)
117
118     """ Get atomic masses """
119
120     self.mass = atoms.get_masses()
121
122     """ Friction conversion factor from a.u. to ase units """
123
124     self.conversion_factor = 0.0267902937/units.fs
125
126     self.fixcm = fixcm # will the center of mass be held fixed?
127     self.communicator = communicator
128     self.rng = rng
129     MolecularDynamics.__init__(self, atoms, timestep, trajectory,
logfile, loginterval)
130     self.updatevars()
131
132     def todict(self):
133         d = MolecularDynamics.todict(self)
134         d.update({'temperature': self.temp,
135                 'friction': self.fr,
136                 'fix-cm': self.fixcm})
137     return d
138

```

```

139     def friction_c(self, rs_c):
140         fric_c = np.where(rs_c > 0., (22.654*rs_c**(2.004)*np.exp(-3.134*
rs_c)+2.497*rs_c**(-2.061)*np.exp(0.0793*rs_c))*self.conversion_factor
/self.C_mass, 0.)
141         return fric_c
142
143     def friction_o(self, rs_o):
144         fric_o = np.where(rs_o > 0., (1.36513 * rs_o ** (-1.8284) * np.
exp(-0.0820301 * rs_o) + 50.342 * rs_o ** (0.490785) * np.exp(-2.70429
* rs_o))*self.conversion_factor/self.O_mass, 0.)
145         return fric_o
146
147     def calculate_friction(self):
148
149         distances = self.atoms.get_distances_list(self.adsorbate_indices ,
self.lattice_indices , mic=True)
150         distances[distances >= self.cutoff] = np.nan
151         density = np.nansum(self.interpolated_density(distances), axis=1)
152
153         rs = np.where(density > 0., (3. / (4. * np.pi * density) ) **
(1./3.), 0.)
154
155         np.put(self.friction , self.C_indices , self.friction_c(rs[self.
C_indices]))
156         np.put(self.friction , self.O_indices , self.friction_o(rs[self.
O_indices]))
157
158     def calculate_temperature(self):
159
160         current_time = self.get_time()
161
162         np.put(self.T, self.adsorbate_indices , self.interpolated_el_temp(
current_time))
163         np.put(self.T, self.lattice_indices , self.interpolated_ph_temp(
current_time))
164
165     def freeze(self):
166
167         freeze_indices = np.where(self.atoms[self.adsorbate_indices].
get_positions()[:,2]>=self.freeze_height)

```

```

168
169     if freeze_indices[0].size != 0:
170         all_indices = np.append(freeze_indices , np.arange(36,44,1))
171         c = FixAtoms(all_indices)
172         self.atoms.set_constraint(c)
173         self.freeze_counter = freeze_indices[0].size
174
175     def print_temperatures(self):
176
177         """Calculate and log the temperatures (as if the system consisted
178         of free gas particles)"""
179
180         current_time = self.get_time()/units.fs
181
182         velocities = np.sqrt(np.sum(self.atoms.get_velocities()**2, axis
183         =1))
184         kinetic_energies = 0.5 * self.mass * velocities**2
185         adsorbate_kinetic_average = 1./self.adsorbate_number * np.sum(
186         kinetic_energies[self.adsorbate_indices])
187         lattice_kinetic_average = 1./(self.lattice_number - self.
188         frozen_lattice) * np.sum(kinetic_energies[self.lattice_indices])
189         C_kinetic_average = 1./(self.adsorbate_number/2.) * np.sum(
190         kinetic_energies[self.C_indices])
191         O_kinetic_average = 1./(self.adsorbate_number/2.) * np.sum(
192         kinetic_energies[self.O_indices])
193
194         adsorbate_temperature = 2./(3. * units.kB) *
195         adsorbate_kinetic_average
196         lattice_temperature = 2./(3. * units.kB) *
197         lattice_kinetic_average
198         C_temperature = 2./(3. * units.kB) * C_kinetic_average
199         O_temperature = 2./(3. * units.kB) * O_kinetic_average
200
201         with open('temperatures_' + self.simulation_number, 'a') as
202         temperatures_output:
203             temperatures_output.write(str(current_time) + '\t' + str(
204             adsorbate_temperature) + '\t' + str(lattice_temperature) + '\t' + str(
205             self.T[0]/units.kB) + '\t' + str(self.T[35]/units.kB) + '\t' + str(
206             adsorbate_kinetic_average) + '\t' + str(lattice_kinetic_average) + '\n
207             ')

```

```

195
196 def updatevars(self):
197
198     if(self.get_number_of_steps() % self.freeze_interval == 0):
199         self.freeze()
200
201     dt = self.timestep
202
203     """Get electronic and phonon temperatures at current time"""
204
205     self.calculate_temperature()
206
207     """Calculate adsorbate friction at current time"""
208
209     self.calculate_friction()
210
211     sigma = np.sqrt(2. * self.T * self.friction / self.mass)
212     self.c1 = (dt / 2. - dt * dt * self.friction / 8.)
213     self.c2 = (dt * self.friction / 2 - dt * dt * self.friction *
self.friction / 8.)
214     self.c3 = (np.sqrt(dt) * sigma / 2. - dt ** 1.5 * self.friction *
sigma / 8.)
215     self.c5 = (dt ** 1.5 * sigma / (2 * np.sqrt(3.)))
216     self.c4 = (self.friction / 2. * self.c5)
217
218     # Works in parallel Asap, #GLOBAL number of atoms:
219     self.natoms = self.atoms.get_number_of_atoms()
220
221 def step(self, f):
222
223     self.updatevars()
224     print(self.get_time()/units.fs)
225
226     atoms = self.atoms
227     natoms = len(atoms)
228
229     # This velocity as well as xi, eta and a few other variables are
stored
230     # as attributes, so Asap can do its magic when atoms migrate
between

```

```

231     # processors .
232     self.v = atoms.get_velocities()
233
234     self.xi = self.rng.standard_normal(size=(natoms, 3))
235     self.eta = self.rng.standard_normal(size=(natoms, 3))
236
237     if self.communicator is not None:
238         self.communicator.broadcast(self.xi, 0)
239         self.communicator.broadcast(self.eta, 0)
240
241     # First halfstep in the velocity.
242     self.v += (self.c1[:, None] * f / self.masses - self.c2[:, None]
243 * self.v + self.xi * self.c3[:, None] - self.c4[:, None] * self.eta)
244
245     # Full step in positions
246     x = atoms.get_positions()
247     if self.fixcm:
248         old_cm = atoms.get_center_of_mass()
249         # Step:  $x^n \rightarrow x^{(n+1)}$  - this applies constraints if any.
250         atoms.set_positions(x + self.dt * self.v + self.c5[:, None] *
251 self.eta)
252     if self.fixcm:
253         new_cm = atoms.get_center_of_mass()
254         d = old_cm - new_cm
255         # atoms.translate(d) # Does not respect constraints
256         atoms.set_positions(atoms.get_positions() + d)
257
258     # recalc velocities after RATTLE constraints are applied
259     self.v = (self.atoms.get_positions() - x -
260 self.c5[:, None] * self.eta) / self.dt
261     f = atoms.get_forces(md=True)
262
263     # Update the velocities
264     self.v += (self.c1[:, None] * f / self.masses - self.c2[:, None]
265 * self.v +
266 self.c3[:, None] * self.xi - self.c4[:, None] * self.
267 eta)
268
269     if self.fixcm: # subtract center of mass vel
270         v_cm = self._get_com_velocity()

```

```
267         self.v -= v_cm
268
269         # Second part of RATTLE taken care of here
270         atoms.set_momenta(self.v * self.masses)
271
272         # Write to output
273         self.print_temperatures()
274
275         return f
276
277     def _get_com_velocity(self):
278         """Return the center of mass velocity.
279         Internal use only. This function can be reimplemented by Asap.
280         """
281         return np.dot(self.masses.flatten(), self.v) / self.masses.sum()
```


Literatura

- [1] Robert O Jones. Density functional theory: Its origins, rise to prominence, and future. *Reviews of modern physics*, 87(3):897, 2015.
- [2] Reinhard J Maurer, Christoph Freysoldt, Anthony M Reilly, Jan Gerit Brandenburg, Oliver T Hofmann, Torbjörn Björkman, Sébastien Lebègue, and Alexandre Tkatchenko. Advances in density-functional calculations for materials modeling. *Annual Review of Materials Research*, 49, 2019.
- [3] Jörg Behler. Representing potential energy surfaces by high-dimensional neural network potentials. *Journal of Physics: Condensed Matter*, 26(18):183001, 2014.
- [4] Andreas Singraber, Jörg Behler, and Christoph Dellago. A library-based lammps implementation of high-dimensional neural network potentials. *Journal of chemical theory and computation*, 2019.
- [5] Khosrow Shakouri, Jörg Behler, Jörgg Meyer, and Geert-Jan Kroes. Accurate neural network description of surface phonons in reactive gas–surface dynamics: N₂⁺ ru (0001). *The journal of physical chemistry letters*, 8(10):2131–2136, 2017.
- [6] Brian Kolb, Xuan Luo, Xueyao Zhou, Bin Jiang, and Hua Guo. High-dimensional atomistic neural network potentials for molecule–surface interactions: Hcl scattering from au (111). *The journal of physical chemistry letters*, 8(3):666–672, 2017.
- [7] Nick Gerrits, Khosrow Shakouri, Jörg Behler, and Geert-Jan Kroes. Accurate probabilities for highly activated reaction of polyatomic molecules on surfaces using a high-dimensional neural network potential: Chd₃⁺ cu (111). *The Journal of Physical Chemistry Letters*, 2019.
- [8] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chemical science*, 8(4):3192–3203, 2017.

- [9] Nongnuch Artrith, Alexander Urban, and Gerbrand Ceder. Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species. *Physical Review B*, 96(1):014112, 2017.
- [10] R. Car and M. Parrinello. Unified approach for molecular dynamics and density-functional theory. *Phys. Rev. Lett.*, 55:2471–2474, Nov 1985.
- [11] Lionel Raff, Ranga Komanduri, and Martin Hagan. *Neural networks in chemical reaction dynamics*. OUP USA, 2012.
- [12] Alireza Khorshidi and Andrew A Peterson. Amp: A modular approach to machine learning in atomistic simulations. *Computer Physics Communications*, 207:310–324, 2016.
- [13] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.
- [14] <https://bioinfo.irc.ca/overfitting-and-regularization>, Oct 2017.
- [15] <https://wiki.fysik.dtu.dk/ase/>.
- [16] Hua Guo, Peter Saalfrank, and Tamar Seideman. Theory of photoinduced surface reactions of ad molecules. *Progress in Surface Science*, 62:239–303, 12 1999.
- [17] SI Anisimov, BL Kapeliovich, and TL Perelman. Electron emission from metal surfaces exposed to ultrashort laser pulses. *Zh. Eksp. Teor. Fiz*, 66(2):375–377, 1974.
- [18] Adolfo G. Eguiluz. Electron-hole-pair damping of an excited state of a molecule absorbed on a metal surface. *Phys. Rev. B*, 30:4366–4371, Oct 1984.
- [19] Martin Head-Gordon and John C Tully. Molecular dynamics with electronic frictions. *The Journal of chemical physics*, 103(23):10137–10145, 1995.
- [20] Mads Brandbyge, Per Hedegård, T. F. Heinz, J. A. Misewich, and D. M. Newns. Electronically driven adsorbate excitation mechanism in femtosecond-pulse laser desorption. *Phys. Rev. B*, 52:6042–6056, Aug 1995.

- [21] M Bonn, Stephan Funk, Ch Hess, Daniel N Denzler, Catherine Stampfl, Matthias Scheffler, Martin Wolf, and Gerhard Ertl. Phonon-versus electron-mediated desorption and oxidation of co on ru (0001). *Science*, 285(5430):1042–1045, 1999.
- [22] D. Novko, M. Blanco-Rey, M. Alducin, and J. I. Juaristi. Surface electron density models for accurate ab initio molecular dynamics with electronic friction. *Phys. Rev. B*, 93:245435, Jun 2016.
- [23] J. I. Juaristi, M. Alducin, R. Díez Muiño, H. F. Busnengo, and A. Salin. Role of electron-hole pair excitations in the dissociative adsorption of diatomic molecules on metal surfaces. *Phys. Rev. Lett.*, 100:116102, Mar 2008.
- [24] Shuichi Nosé. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of chemical physics*, 81(1):511–519, 1984.
- [25] Dino Novko, Ivor Lončarić, María Blanco-Rey, J. Iñaki Juaristi, and Maite Alducin. Energy loss and surface temperature effects in ab initio molecular dynamics simulations: N adsorption on ag(111) as a case study. *Phys. Rev. B*, 96:085437, Aug 2017.
- [26] Ivor Lončarić, M. Alducin, P. Saalfrank, and J. I. Juaristi. Femtosecond-laser-driven molecular dynamics on surfaces: Photodesorption of molecular oxygen from ag(110). *Phys. Rev. B*, 93:014301, Jan 2016.
- [27] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, and B. I. Lundqvist. Van der waals density functional for general geometries. *Phys. Rev. Lett.*, 92:246401, Jun 2004.
- [28] P. E. Blöchl. Projector augmented-wave method. *Phys. Rev. B*, 50:17953–17979, Dec 1994.
- [29] <https://wiki.fysik.dtu.dk/asap/>. [Pristupljeno 26.6.2019.].
- [30] William C Swope, Hans C Andersen, Peter H Berens, and Kent R Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *The Journal of Chemical Physics*, 76(1):637–649, 1982.

- [31] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.
- [32] <https://wiki.fysik.dtu.dk/ase/ase/units.html>. [Pristupljeno 27.6.2019.].
- [33] Eric Vanden-Eijnden and Giovanni Ciccotti. Second-order integrators for Langevin equations with holonomic constraints. *Chemical physics letters*, 429(1-3):310–316, 2006.
- [34] H. Davy. Some new experiments and observations on the combustion of gaseous mixtures; with an account of a method of preserving a continued light in mixtures of inflammable gases and air without flame. *Proceedings of the Royal Society of London*, 2:77, Jan 1815.
- [35] Michael Faraday. Experimental researches in electricity. sixth series. *Philosophical Transactions of the Royal Society of London*, 124:55–69, 1834.
- [36] Irving Langmuir. The mechanism of the catalytic action of platinum in the reactions $2CO + O_2 = 2CO_2$ and $2H_2 + O_2 = 2H_2O$. *Trans. Faraday Soc.*, 17:621–654, 1922.