

分子動力学におけるニューラルネットワークの利用

西原 慧径*

Neural Network Techniques for Molecular Dynamics

Satomichi Nishihara*

材料シミュレーションの分野において近年注目されつつある、ニューラルネットワーク力場に関する弊社の取り組みを紹介する。弊社ではニューラルネットワーク力場を運用するための製品である Advance/NeuralMD を開発および販売しており、本記事では当該製品を活用してリチウムイオン伝導体の力場を作成してイオン伝導率を解析した事例を示す。また、2020 年より Facebook 社とカーネギーメロン大学が共同で実施している Open Catalyst Project では、触媒系の解析を主たる目的としてグラフニューラルネットワークを利用した汎用力場の開発が試みられている。2021 年 12 月に当該プロジェクトの成果物として学習済みの力場情報が公開されており、弊社ではこの汎用力場を自社開発の GUI システムである Advance/NanoLabo に搭載した。汎用力場の運用方法などについても紹介する。

Keywords: 分子動力学、第一原理計算、ニューラルネットワーク、ニューラルネットワーク力場、グラフニューラルネットワーク、汎用力場、Open Catalyst Project、リチウムイオン伝導体、イオン伝導率、LAMMPS、Quantum ESPRESSO、Advance/NanoLabo、Advance/NeuralMD

1. はじめに

分子動力学計算を実施する際には、力場は必須である。力場は与えられた原子座標におけるポテンシャルエネルギーを定義するものであり、一階微分することで原子に働く力が得られる。この力を使用して運動方程式を解くことで、分子動力学計算が遂行される。従来の力場では、簡単な関数形を用いてポテンシャルエネルギーを表現していた。例えば、有機分子系における共有結合はバネ（二次関数）で表現され、金属や半導体などの無機材料における原子間相互作用は Lennard-Jones ポテンシャルや Buckingham ポテンシャルなどの 2 体関数で表現される。このような簡単な関数形を使用することで、少ない計算コストで効率的なシミュレーションが実施できる。しかしながら、その計算精度は関数形の中で使用されるパラメータに大きく依存する（バネ定数など）。また、既報文献などで公開されているパラメータが存

在すれば、その値を参照することで直ちに分子動力学計算を実施することができるのだが、逆に既知のパラメータがなければ適正なパラメータを自作する必要がある。パラメータの作成には第一原理計算の結果を参照する必要があり、極めて多くの手間が掛かる。計算精度の問題、および、パラメータを自作する必要があるという問題から、分子動力学においてシステムティックに力場を作成する方法や汎用的な力場を開発することは極めて重要な課題であった。残念ながら、この課題は長年に渡って解決されずにいた。しかしながら、近年における機械学習やニューラルネットワークの技術の発展に伴い、材料シミュレーションの分野においてもニューラルネットワークの活用が進み、2010 年代に入ってからニューラルネットワークにてポテンシャルエネルギーを表現するというニューラルネットワーク力場が開発された。ニューラルネットワーク力場の登場により、第一原理計算と同程度の計算精度にて分子動力学計算が実施できるようになり、かつ、未知の系に対してもシステムティックに力場の作成が可能とな

*アドバンスソフト株式会社 第 1 事業部

1st Computational Science and Engineering Group,
AdvanceSoft Corporation

った。これにより力場における課題が解決され、分子動力学法が種々の系に適用可能となり、当該分野は大きな発展を遂げつつある。

2. ニューラルネットワーク力場

ニューラルネットワーク力場にはいくつかのバリエーションが存在するのだが、現在最もよく使われているのが Behler によって提案された High-Dimensional Neural Network Potential (HDNNP) である[1]。弊社製品である Advance/NeuralMD においても、この HDNNP に基づいた機能が実装されている。

2.1. HDNNP

HDNNP では、系のポテンシャルエネルギー E_{tot} を各原子のエネルギー E_i の総和で表現する：

$$E_{tot} = \sum_{i \in \{\text{all atoms}\}} E_i \quad (1)$$

各原子のエネルギーは、ニューラルネットワークにて計算される：

$$E_i = E_{NN}(G_{i,1}, G_{i,2}, G_{i,3}, \dots) \quad (2)$$

E_{NN} は E_i を出力する多層パーセプトロンであり、隠れ層の数は 2~3 程度、各層のノード数は 20~50 程度である。 E_{NN} の入力層は、対称関数 G_i である。対称関数は、 i 番目の原子の周辺にある他の原子の立体構造(化学環境)を表現したものであり、原子間距離や結合角などを引数に含む関数である。

対称関数にも多くのバリエーションが存在するのだが、Advance/NeuralMD では原子番号で重み付けされた Behler 関数[2]：

$$G_i^{rad} = \sum_j e^{-\eta(R_{ij}-R_s)^2} f_c(R_{ij}) Z_j \quad (3)$$

$$G_i^{ang} = 2^{1-\zeta} \sum_{j,k} (1 + \lambda \cos(\theta_{ijk}))^\zeta \times e^{-\eta[(R_{ij}-R_s)^2 + (R_{ik}-R_s)^2 + (R_{jk}-R_s)^2]} \times \quad (4)$$

$$f_c(R_{ij}) f_c(R_{ik}) f_c(R_{jk}) \sqrt{Z_j Z_k}$$

および、Chebyshev 関数[3]：

$$c_{i,\alpha}^{(2)} = \sum_j T_\alpha\left(\frac{2R_{ij}}{R_c} - 1\right) f_c(R_{ij}) Z_j \quad (5)$$

$$c_{i,\alpha}^{(3)} = \sum_{j,k} T_\alpha\left(\frac{2\theta_{ijk}}{\pi} - 1\right) \times f_c(R_{ij}) f_c(R_{ik}) \sqrt{Z_j Z_k} \quad (6)$$

を採用している。 Z_j は原子番号、 $f_c(R_{ij})$ カットオフ関数である。いずれの対称関数についても、原子番号で重み付けがなされているため、元素種数が増えても対称関数の数が増大せずに計算コストを抑えることができるというメリットがある。実用上は 7 元素種程度であれば、十分な精度の力場を作成可能であることを確認済みである。

力場の関数形にニューラルネットワークを使用しているため、従来の力場におけるパラメータに相当するものはニューラルネットワークにおける重みとバイアスとなる。従って、重みとバイアスを系に応じて最適化する必要がある。この最適化のプロセスこそが、ニューラルネットワークの学習である。学習には教師データが必要であるのだが、今回の場合には何らかの手法にて事前に計算された高精度なポテンシャルエネルギーが教師データとなり得る。この目的においては第一原理計算が適任である。すなわち、種々の原子構造において第一原理計算を行って全エネルギーを得る。原子構造と全エネルギーの組を教師データとすれば良い。また、ニューラルネットワークを順方向に伝搬させれば関数値(エネルギー)を得るのだが、逆方向に伝搬させると微分値すなわち力を得ることができる。つまり、第一原理にて計算された原子に働く力についても教師データに含めることができるのである。この場合には、ニューラルネットワークの学習過程において二階微分が要求されることになる。二階微分は計算コストが高いため、学習におけるボトルネックとなる。

十分な精度の力場を得るためには、教師データとして数万以上の原子構造について第一原理計算を実施する必要がある。多数の教師データの作成には、多大な計算コストを要する。このため、できるだけ教師データの数を少なくするための工夫が必要となる。弊社では HDNNP を改良した Δ -Neural Network Potential (Δ -NNP)を開発し、教

師データの数を削除しても十分な精度の力場を作成できることを実証した。 Δ -NNP は Advance/NeuralMD に実装済みである。

2.2. Δ -NNP

通常の HDNNP では、式(1-2)のように、ポテンシャルエネルギー全体をニューラルネットワークで表現する。一方、 Δ -NNP ではポテンシャルエネルギー全体 E_{tot} を従来の古典力場で計算されたポテンシャルエネルギー E_C とニューラルネットワークで計算されたポテンシャルエネルギー $\sum E_i$ の足し合わせで表現する：

$$E_{tot} = E_C + \sum_{i \in \{\text{all atoms}\}} E_i \quad (7)$$

ここで、古典力場としては Lennard-Jones-like な 2 体ポテンシャル

$$E_C = \frac{1}{2} \sum_{i,j} \frac{A}{R_{ij}^{12}} + \frac{B}{R_{ij}^{10}} + \frac{C}{R_{ij}^8} + \frac{D}{R_{ij}^6} \quad (8)$$

を使用する。式(8)右辺に含まれるパラメータ A, B, C, D は教師データ(第一原理計算の結果)を使用して事前に最適化されたものを使用する。ポテンシャルエネルギー全体を式(7)のように定義すると、ニューラルネットワークの学習に使用する教師データは第一原理にて計算された全エネルギー E_{DFT} そのものではなく、第一原理と古典力場との差分値 $E_{DFT} - E_C$ となる。差分値を教師データとするため、“ Δ ”-NNP と名付けた。

Δ -NNP は無機材料に対して極めて有用である。無機結晶においては、HDNNP ほどの精度はないにせよ、Lennard-Jones や Buckingham などの 2 体関数にてポテンシャル地形の概ねの構造を表現できることが知られている。そこで、ポテンシャル地形の第 0 近似として 2 体関数を適用し、2 体関数では表現しきれない余剰の部分をニューラルネットワークで補完するというアプローチが Δ -NNP の基本的な考え方である。その結果、ニューラルネットワークは凹凸の少ない地形のみを表現することになり、ニューラルネットワークの真価を十分に発揮することなく結晶系を取り扱うことができる。つまり、結晶系という平時においては、ニューラルネットワークは余力を十分に

残した状態にある。逆に、欠陥や不純物、アモルファス状態、表面や界面、イオンが激しく運動するといった特殊な状況においては、このニューラルネットワークの余力が機能するのである。

ニューラルネットワークの研究では往々にして「アルゴリズム or ニューラルネットワーク」という形で既存のアルゴリズムとの対比にてニューラルネットワークの優位性を示すことがある。 Δ -NNP は逆に「アルゴリズム and ニューラルネットワーク」という形で既存のアルゴリズム(古典力場)と組み合わせてニューラルネットワークを運用することで、両者の長所を同時に活かすことのできる手法と言える。

また、 Δ -NNP の適用により、ニューラルネットワーク学習過程の収束性が向上し、かつ、少ない教師データでも有用な力場が作成できるというメリットが得られた。リチウムイオン伝導体に適用した実例を、次章で示す。

3. リチウムイオン伝導体への応用

リチウムイオン伝導体である $\text{Li}_{10}\text{GeP}_2\text{S}_{12}$ (LGPS)に Δ -NNP を適用して、イオン伝導率を計算した。LGPS は室温で 10^{-2}S/cm 以上の高いイオン伝導率を有しており、全固体リチウムイオン電池における固体電解質としての利用が期待されている。正方晶のユニットセルに、四面体構造の $[\text{GeS}_4]^{4-}$ を 2 つ、 $[\text{PS}_4]^{3-}$ を 4 つ、 Li^+ を 20 個含んでいる(図 1)。

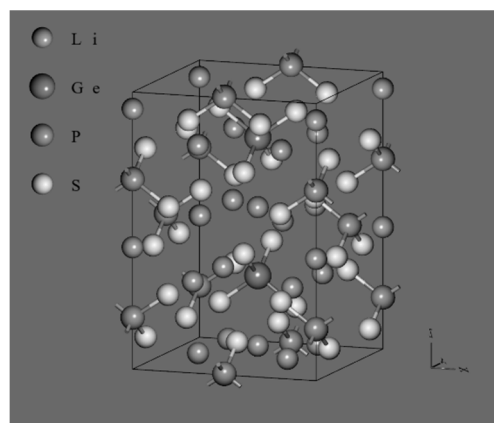


図 1 LGPS の結晶構造

3.1. 初期力場の作成

まずは、少ない教師データで初期力場を作成す

る。その後、強化学習により教師データ数を増やす。初期力場用の教師データとしては、LGPS のユニットセル(50 原子系)を使用した。結晶構造には Materials Project[4] より取得したもの(mp-696128)をそのまま使用している。取得した結晶構造を基に、原子座標をランダムに $0.1\sim 0.2\text{\AA}$ だけ変位させて多数の構造を生成し、各構造に対して第一原理計算を実施してその結果を教師データとした。SCF 計算が収束したもののみを教師データとして採用しており、採用された構造数は 714 個である。第一原理計算には Quantum ESPRESSO[5]を使用しており、擬ポテンシャルにはウルトラソフト型、カットオフエネルギーは 40Ry 、k 点サンプリングは Γ 点のみとした。

作成した教師データを用いて、力場を最適化した。まずは 2 体関数(式(8))におけるパラメータを最適化して、その後にニューラルネットワークの学習を行った。比較のために、2 体関数を使わない通常の HDNNP も作成した。 Δ -NNP および HDNNP とともに、 $\text{RMSE}(\text{Energy}) \simeq 1.4\text{meV/atom}$ 、 $\text{RMSE}(\text{Force}) \simeq 0.08\text{eV/\AA}$ まで収束させた。ニューラルネットワーク力場における対称関数には、80 個の Chebyshev 関数を使用した。カットオフ半径は $6\text{\AA}$ で、多層パーセプトロンの構造は 2 層 $\times$ 40 ノード、活性化関数は twisted tanh である。

3.2. 初期力場を使った試計算

作成した初期力場を使用して、簡易な分子動力学計算を試みた。分子動力学計算には LAMMPS[6]を使用した。ただし、弊社にて HDNNP および Δ -NNP を使用できるように改修したものである[7]。計算対象は LGSP のユニットセルで、NVT アンサンブルにて 100ps のシミュレーションを実施した。温度は 800K 、時間刻み幅は 0.5fs である。

Δ -NNP にて計算された 100ps 後の LGPS の構造を図 2 に示す。また、このときの動径分布関数を図 3 に示す。 800K という高温での分子動力学シミュレーションであるにもかかわらず、構造が破綻することなく $[\text{GeS}_4]^{4-}$ および $[\text{PS}_4]^{3-}$ の四面体構造が保持されている。原子座標をランダムに変

位させただけの 700 個程度の少数の教師データのみで作成した力場ではあるが、極めて高い安定性の実現できている。もちろん、当該力場では教師データの数少な過ぎるため、このまま定量的な解析を行うことはできない。しかしながら、この初期力場を使って分子動力学計算やモンテカルロ計算を実施することで多数の構造を新たに生成することができる。生成された構造を教師データに追加することで強化学習を実施して、定量性が担保できるまでに力場の精度を向上させることができる。

Δ -NNP において最適化された 2 体関数の形状を図 4 に示す。 Ge-S および P-S の共有結合が正しく表現できていることが分かる。これにより構造が破綻することなく、安定に保たれるのである。また、 Li-S については引力相互作用ではあるものの、その結合は比較的弱い。つまり、 Li イオンが比較的自由に動くことができ、LGPS がイオン伝導体として機能することを示唆している。

一方、HDNNP にて計算された 100ps 後の構造および動径分布関数を、図 5 および図 6 に示す。構造が大きく破綻しており、 Li イオンが凝集していることが分かる。HDNNP では 2 体関数による骨格が存在しないため、約 700 個という少ない教師データでは十分な性能の力場を作ることができなかったのである。すなわち、教師データの範囲を逸脱する構造が出現すると直ちに力場が破綻して、外挿がほとんど機能しないのである。このように、HDNNP では初期力場を使った分子動力学計算やモンテカルロ計算にて教師データを増やして強化学習を実施することができないのである。

3.3. 強化学習

Δ -NNP にて作成した初期力場を使用して、メトロポリス法に基づくモンテカルロ計算を実施した。モンテカルロ計算の機能は Advance/NeuralMD に実装されており、事前に生成済みの対称関数の分布を利用して未知の構造のみを抽出することができる。温度を $300\sim 1000\text{K}$ に設定して多数の構造を生成した。最終的な教師データ

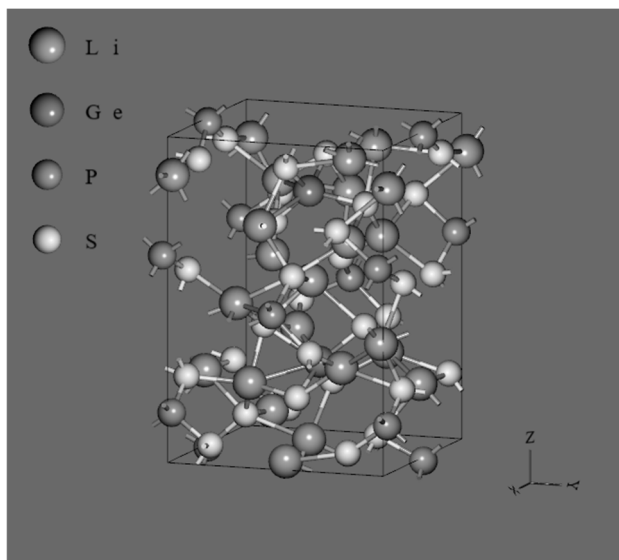


図2 Δ-NNP で計算した 100ps 後の構造

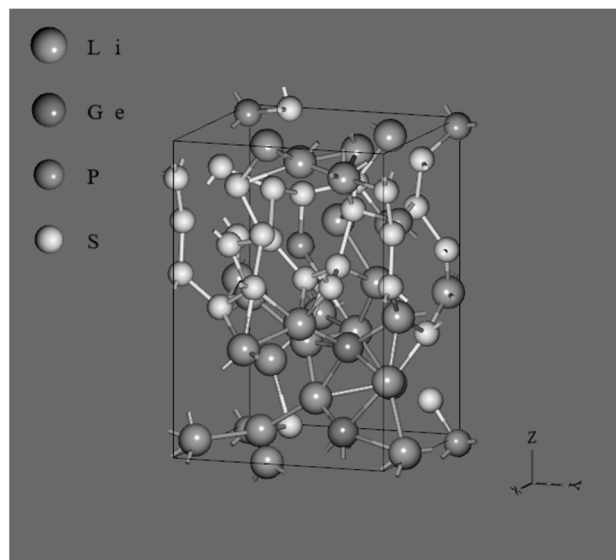


図5 HDNNP で計算した 100ps 後の構造

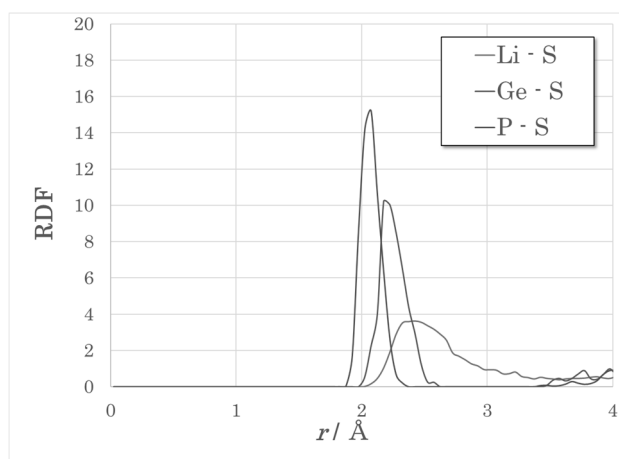
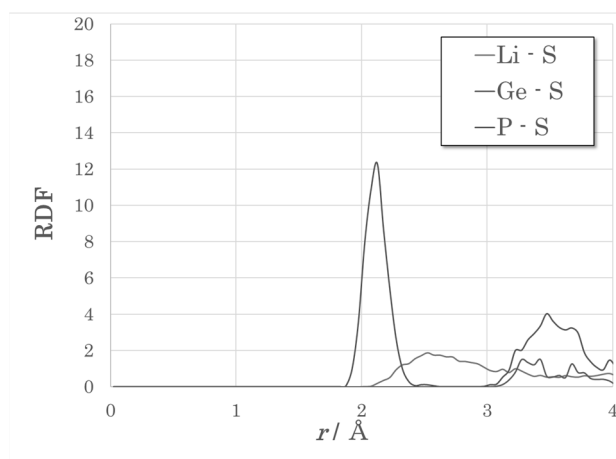
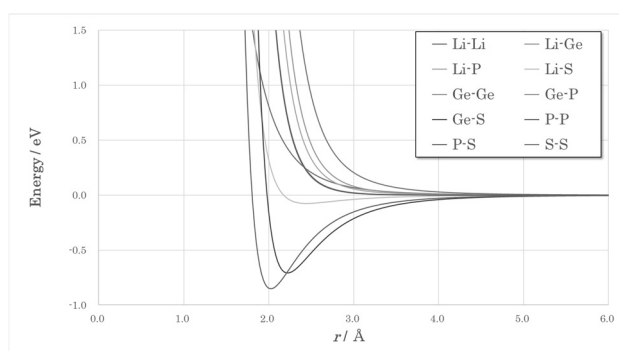
図3 Δ-NNP で計算された
100ps 後の動径分布関数図6 HDNNP で計算された
100ps 後の動径分布関数

図4 Δ-NNP における 2 体関数

の数は、6,914 個である。これらの教師データを用いて、L-BFGS 法にて 10,000 エポックの学習を実施した。RMSE(Energy) = 3.4meV/atom、RMSE(Force)=0.077eV/Åまで収束した。学習過程

における RMSE の変化を図 7 および図 8 に示す。エネルギーおよび原子に働く力ともに、HDNNP よりも Δ-NNP の方が収束性が良いことが確認できる。

3.4. イオン伝導率の計算

作成した力場を使用して Li イオンの拡散係数を計算した。拡散係数の計算方法には、Mean Square Displacement (MSD)の傾きから算出する方法を採用した。この方法では、十分に大きな系にて分子動力学計算を実施する必要がある。今回は 1600 原子系のスーパーセルモデルを使用した(図 9)。NVT アンサンブルにて時間刻み幅を 0.5fs とし、0.5~1.0ns のシミュレーションを実施して

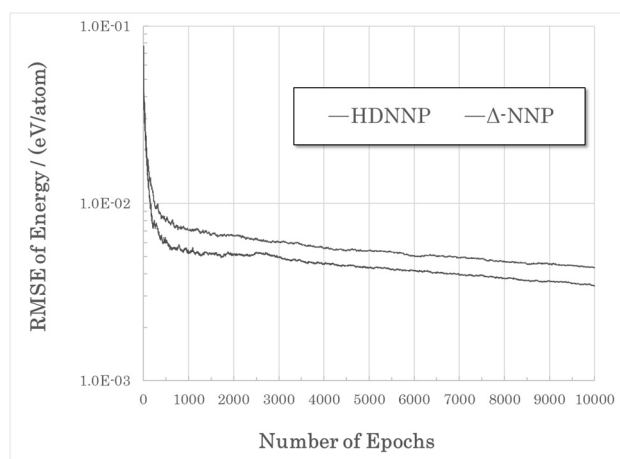


図7 エネルギーの RMSE

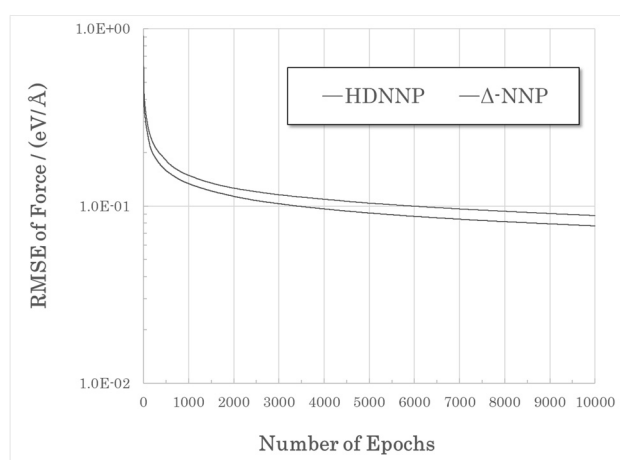


図8 原子に働く力の RMSE

MSD の傾きを評価した。温度は 350～800K の範囲の種々の値とした。一般に、高温では原子の運動が大きく、モデルサイズが小さくても十分な精度で拡散係数を評価できる。逆に、低温ではモデルサイズを十分に大きくしないと十分な統計精度を得ることができない。このため、第一原理分子動力学法でも高温領域での拡散係数は計算できるものの、低温（常温）領域では第一原理分子動力学法を使うことはできない。従って、比較的広い温度範囲の拡散係数を評価するという目的において、第一原理に近い精度で 1000 原子以上のシミュレーションが可能なニューラルネットワーク力場は極めて有用な手法と言える。

実際に計算された Li イオンの拡散係数 D は、下式にてイオン伝導率 σ に換算できる。

$$\sigma = \frac{\rho e^2 D}{kT} \quad (9)$$

ρ は Li イオンの濃度、 e は素電荷、 k はボルツマン定数、 T は温度である。各温度について計算されたイオン伝導率を図 10 に示す。ただし、横軸は逆温度、縦軸は σ の log スケールである。高温領域においては既報の第一原理計算の結果[8]をよく再現しており、かつ、第一原理では計算できない 300K 近傍の低温領域も計算できている。300K 近傍においては、実験値[9]をよく再現できている。 $T = 298.15\text{K}$ におけるイオン伝導率を 2 次関数で補外して予測すると、 $\sigma = 1.6 \times 10^{-2} \text{S/cm}$ であった。実験値 $\sigma = 1.2 \times 10^{-2} \text{S/cm}$ にかかなり近い数値が得られた。

このように、 Δ -NNP を活用すると第一原理計算よりも低コストに高精度なシミュレーションが可能であることに加え、既存の HDNNP よりも少ない教師データの数で力場作成ができるため大幅な時間短縮につながる。さらに、最新のクラウド環境と組み合わせて運用することでより効率的なシミュレーションが実現できる[10]。

ここまでで紹介したニューラルネットワーク力場の作成から分子動力学計算の実施までの一連の計算は、弊社にて開発および販売している GUI システムである Advance/NanoLabo の画面上で実施することができる。ユーザーは入力ファイルの記法や煩わしいデータ管理を意識することなく、ニューラルネットワーク力場の作成や運用ができるようになっている。

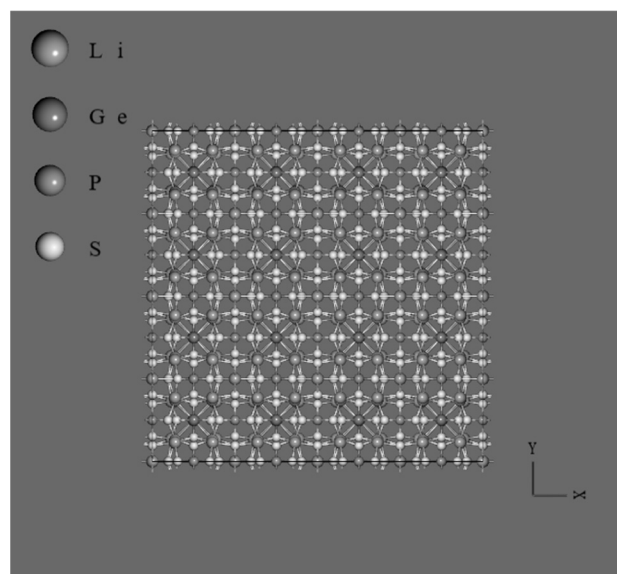


図9 LGPS のスーパーセルモデル

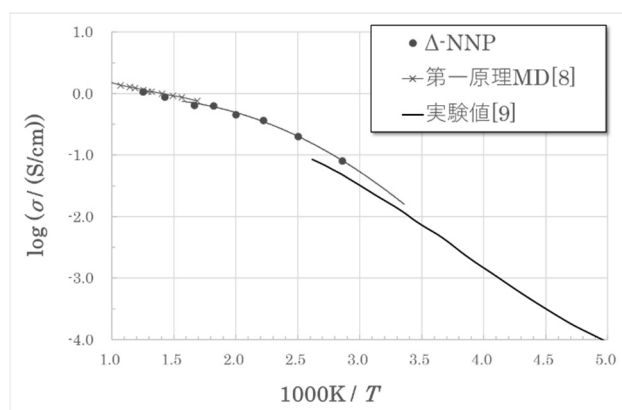


図 10 Li イオンのイオン伝導率

4. Open Catalyst Project

2020 年より Facebook 社とカーネギーメロン大学が共同で、Open Catalyst Project と実施している[11]。当該プロジェクトでは、最新のグラフニューラルネットワークを応用して触媒系の効率的なシミュレーションを実施するためのシステムを開発することを目的としている。プロジェクトの中ではいくつかの課題が設けられており、その中の一つ“S2EF”というものが指定された構造におけるエネルギーと力を予測するという内容であり、ニューラルネットワーク力場に相当する。S2EF では 2 億構造以上の教師データを使用してニューラルネットワーク力場を最適化しており、学習済みのニューラルネットワークの情報は GitHub にて公開されている[12]。2 億以上の教師データの中には多種多様な構造が含まれており、触媒系のみならず、多くの結晶や分子に適用可能な“汎用力場”となっている。弊社では世界に先駆けて、この汎用力場のインターフェースを開発して LAMMPS および Advance/NanoLabo に搭載した。

4.1. 汎用力場

前章までで紹介してきたニューラルネットワーク力場は、Behler-Parinello 型[1]と呼ばれる。Behler-Parinello 型の力場は種々の系に適用され、多くの成果を挙げている。しかしながら、ユーザーが自ら教師データを作成して力場を最適化する必要があるため、即座に分子動力学シミュレーションを実施することができないというデメリットがある。一方、Open Catalyst Project にて開発

されたニューラルネットワーク力場は、莫大な教師データのもとで事前に学習済みであるため、ユーザーが自身で力場を作成する必要がなく、簡単にシミュレーションが行える。また、種々の系に適用可能な汎用力場である。このような汎用性はグラフニューラルネットワークを利用していることに起因する。Behler-Parinello 型力場では注目している原子の置かれている化学環境を対称関数にて直接に表現する。一方、汎用力場では対称関数を直接使用するのではなく、原子間の結合を考慮したグラフニューラルネットワークによる複数回の畳み込みを実施することでより精密な化学環境の表現が可能となる。特にこの畳み込み過程はスルーボンドでの化学情報の伝搬を実現する。

Open Catalyst Project で開発された汎用力場があるのなら Behler-Parinello 型のような個別の系に特化した力場は不要ではないのか、と思われるかもしれない。しかしながら、必ずしもそうではない。汎用力場は決して全ての問題を解決し得る万能な手法ではない。汎用力場の計算コストは $O(N)$ である(N は原子数)。一方、第一原理計算の計算コストは $O(N^{3-7})$ であり、多体シュレーディンガー方程式の厳密解は $O(N^\infty)$ である。もし仮に任意の系に対して、汎用力場にて第一原理計算や厳密解を完全に再現可能であるのなら $P=NP$ となるわけだが、おそらくこれは間違いである。現実の問題として、現行の汎用力場では物理量の定量的な評価には問題があることを確認している。すなわち、汎用力場は汎用性が高いものの計算精度は若干低くなっている。逆に、Behler-Parinello 型力場については、汎用性は低いのだが特定の問題に対して高い精度を担保できる。

実際の運用方法としては、汎用力場にて大雑把にシミュレーションを実施して計算条件やモデルの妥当性を見積もった後、より定量的な評価をしたいのであれば Behler-Parinello 型力場に切り替えるのが良さそうである。その際、汎用力場で実施した分子動力学計算のトラジェクトリーを教師データの作成に使用することで、より効率的に Behler-Parinello 型力場の作成が実現できる。

4.2. LAMMPS インターフェース

Open Catalyst Project では PyTorch[13]を使って各種ツールが実装されており、S2EF の汎用力場についても Python および PyTorch にて実装されている。この Python のモジュールは与えられた原子構造におけるエネルギーと力を出力するようになっているのだが、分子動力学計算の機能自体は実装されていない。そこで弊社にて、当該モジュールと LAMMPS のインターフェースを開発して、GitHub にて公開した[7]。当該インターフェースは“OC20DRIVER”という名称のパッケージであり、LAMMPS で利用可能な多体力場(pair_style)として汎用力場を提供している。これにより、LAMMPS における分子動力学法の種々の機能と汎用力場を組み合わせ運用できる。ただし、リアル応力が計算できない、MPI 並列に非対応であるという制約がある。OpenMP によるスレッド並列および GPU には対応している。大規模系を計算したい場合には GPU の使用が推奨である。

汎用力場の利用方法としては、入力ファイルにて下記のコマンドを設定する。

```
pair_style oc20
```

GPU を使用する場合には、

```
pair_style oc20/gpu
```

である。Open Catalyst Project にて提供されている汎用力場にはいくつかのバリエーションが存在するのだが、OC20DRIVER では、

- ✓ GemNet-dT
- ✓ DimeNet++
- ✓ CGCNN
- ✓ SchNet
- ✓ SpinConv

の5つが利用可能である。使用する力場の種類は、以下のコマンドで設定する。

```
pair_coeff ** GemNet-dT 元素名リスト
```

上記は、GemNet-dT を使用する場合の例である。また、GemNet-dT については順方向伝搬にて力を計算しているため、力がエネルギーの微分値に完全には合致しない。エルゴード性が担保されない

ため、その運用には留意が必要である。

また、Advance/NanoLabo には OC20DRIVER のインターフェースが搭載されているため、LAMMPS を介して Advance/NanoLabo から Open Catalyst Project の汎用力場を利用することも可能となっている。

参考文献

- [1] J. Behler, IJQC 115, 1032 (2015).
- [2] M. Gastegger, et al., J. Chem. Phys. 148, 241709 (2018)
- [3] N. Artrith, et al., Phys. Rev. B 96, 014112 (2017)
- [4] <https://materialsproject.org>
- [5] <https://www.quantum-espresso.org>
- [6] <https://www.lammps.org>
- [7] <https://github.com/advancesoftcorp/lammps>
- [8] A. Marcolongo, et al., <https://arxiv.org/abs/1910.10090>
- [9] 菅野了次, Electrochemistry, 85(9), 591-596 (2017)
- [10] <https://www.engineering-eye.com/EXABYTE/>
- [11] <https://opencatalystproject.org>
- [12] <https://github.com/Open-Catalyst-Project/ocp>
- [13] <https://pytorch.org>

※ 技術情報誌アドバンスシミュレーションは、アドバンスソフト株式会社 ホームページのシミュレーション図書館から、PDF ファイル（カラー版）がダウンロードできます。（ダウンロードしていただくには、アドバンス/シミュレーションフォーラム会員登録が必要です。）