第34回CCSEワークショップ

「原子力材料研究開発におけるDX推進の現状と将来：原子力材料研究開発の革新と新展開」

Neural Network力場のGPUによる高速化

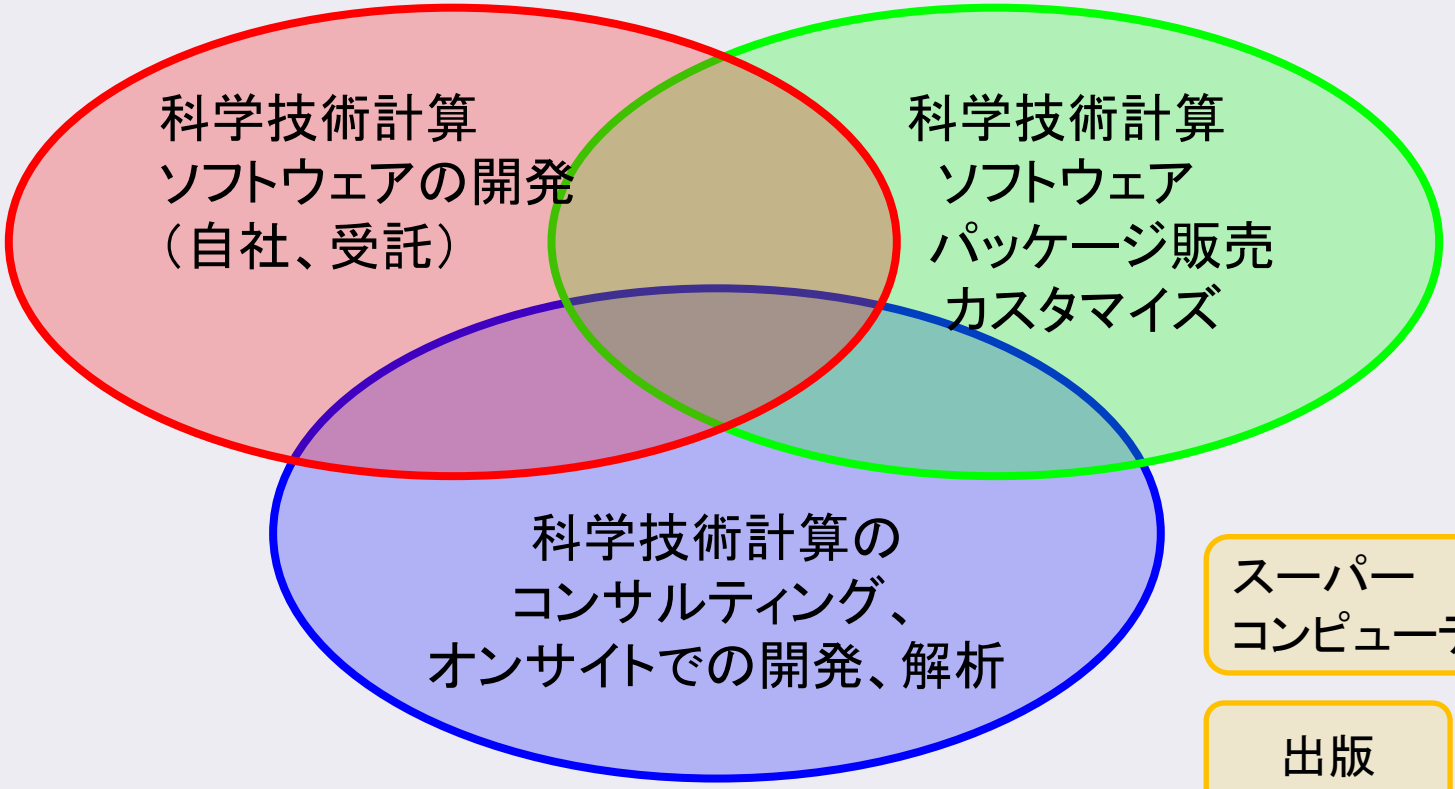
アドバンスソフト株式会社

西原　慧径

[14:45～15:15]

アドバンスソフトについて

アドバンスソフトがご提供するサービス



スーパー
コンピューティング

出版

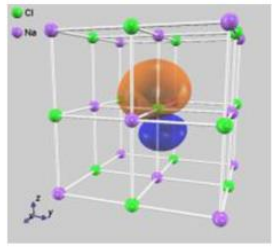
科学技術計算ソフトウェアの開発を基礎とした、
科学技術計算に関する様々なソリューションをご提供します。

アドバンスソフトについて

開発・販売しているソフトウェア製品

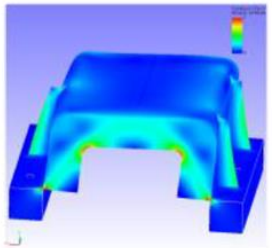
ナノ

Advance/PHASE
Advance/NanoLabo



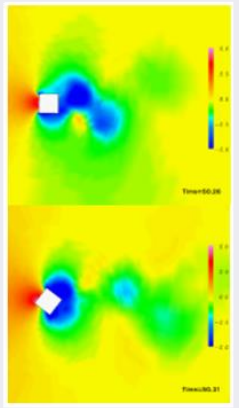
構造・音響

Advance/FrontSTR
Advance/FrontNoise

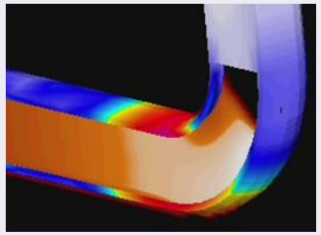


流体

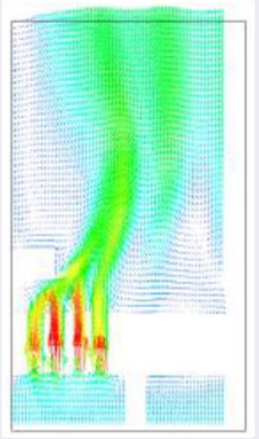
Advance/FrontFlow/red



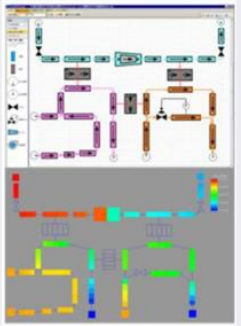
Advance/FrontFlow/FOCUS



Advance/FrontFlow/MP

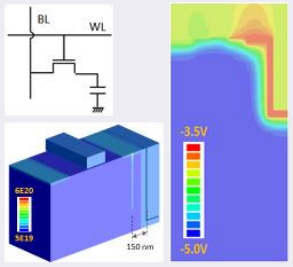


Advance/FrontNetシリーズ



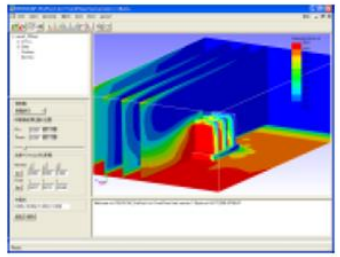
半導体・光／電磁波

Advance/TCAD
Advance/ParallelWave



プリポスト

Advance/REVOCAP



アウトライン

1. Neural Network力場の製品化
2. Neural Network力場のGPU化
3. Neural Network力場とヤコビの梯子

1. Neural Network力場の製品化

Advance/NeuralMD

- 当社では**Neural Network力場**を作成して運用するためのソフトウェアである**Advance/NeuralMD**という製品を開発および販売している。

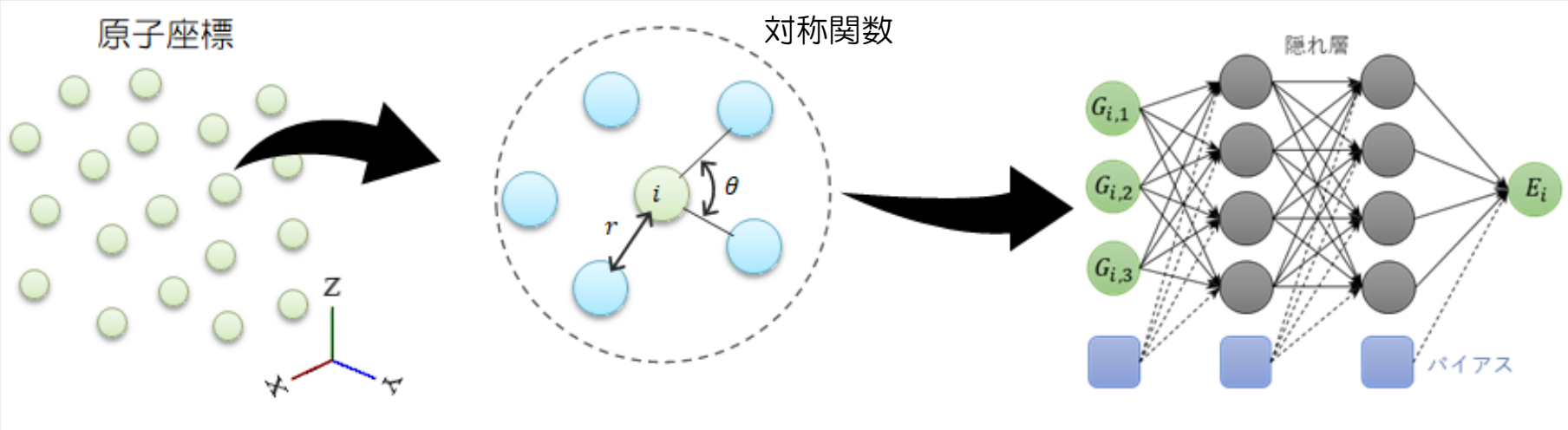
- GUIから操作可能で、簡単に力場を作成することが出来る。

- 計算エンジンを自社開発しており、種々の特徴的な機能を搭載。

    1. Δ-NNP法による効率的な学習
    2. Metropolis法による未知教師データの生成
    3. 自己学習ハイブリッドモンテカルロ法

Neural Network力場について

- 分子動力学計算にて使用する分子力場をNeural Networkで表現したもの。
- 第一原理計算(DFT)の結果を教師データとしてNeural Networkを学習させる。
- 既存の分子力場(OPLS-AA、ReaxFF、Tersoffなど)に比べて高精度(ただし計算コストも高い)。
- 原子座標から各原子の化学環境を表現した対称関数を生成して多層パーセプトロンに入力、エネルギーを出力する。逆方向の伝搬で力(エネルギーの微分)も計算できる。



Neural Network力場について

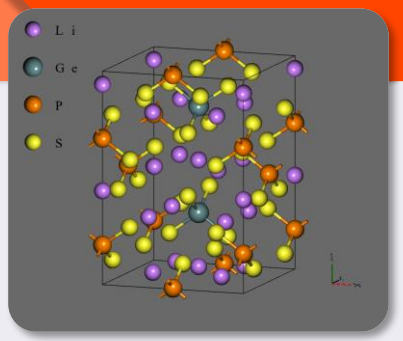
長所

- 第一原理の精度を再現しつつ、数千原子系での数ns以上の分子動力学計算が可能。
- 既存の分子力場が存在しない系につても、ユーザーが自ら力場を作成できる。

短所

- 既存の分子力場よりも計算コストが高く(ReaxFFの5倍~)、一般的な計算サーバーでは数千原子系の計算が限界。
  ⇒ GPU化により大幅に高速化され、数万～数十万原子系での計算も可能
- 力場の性能や精度が、教師データの品質・作成手順・ユーザーの練度などに依存する。
  ⇒ 自己学習ハイブリッドモンテカルロ法などにより解決

Δ-NNP法



- Δ-NNP法では、全エネルギーE_{tot}を古典力場にて計算されたエネルギーE_{C}とNNPにて計算されたエネルギーE_{NNP}の和で表現する： $E_{\mathrm{tot}} = E_{\mathrm{C}} + E_{\mathrm{NNP}}$

- 古典力場としては、Lennard-Jones様の2体ポテンシャルを使用する：

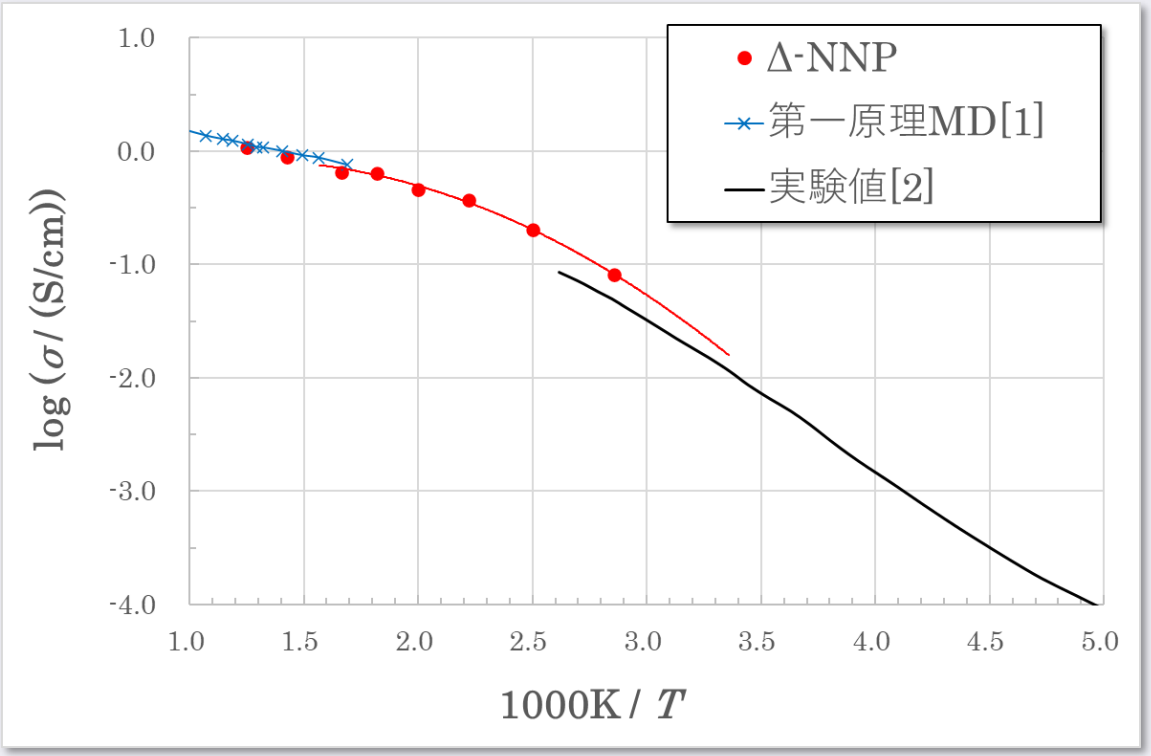
$$E_{\mathrm{C}} = \frac{1}{2}\sum_{i,j}\frac{A}{r_{ij}^{12}} + \frac{B}{r_{ij}^{10}} + \frac{C}{r_{ij}^{8}} + \frac{D}{r_{ij}^{6}}$$

- Neural Networkの教師データにはDFTで計算されたエネルギーE_{DFT}そのものではなく、DFTと古典力場の差分 $E_{\mathrm{DFT}} - E_{\mathrm{C}}$ を使用する。

- ポテンシャル地形の第0近似を古典力場として、余剰のエネルギーをNNPで補完するというアプローチ。

- (1)Neural Network学習過程の収束性が向上し、且つ、(2)少ない教師データでも有用な力場を作成できるというメリットがある。

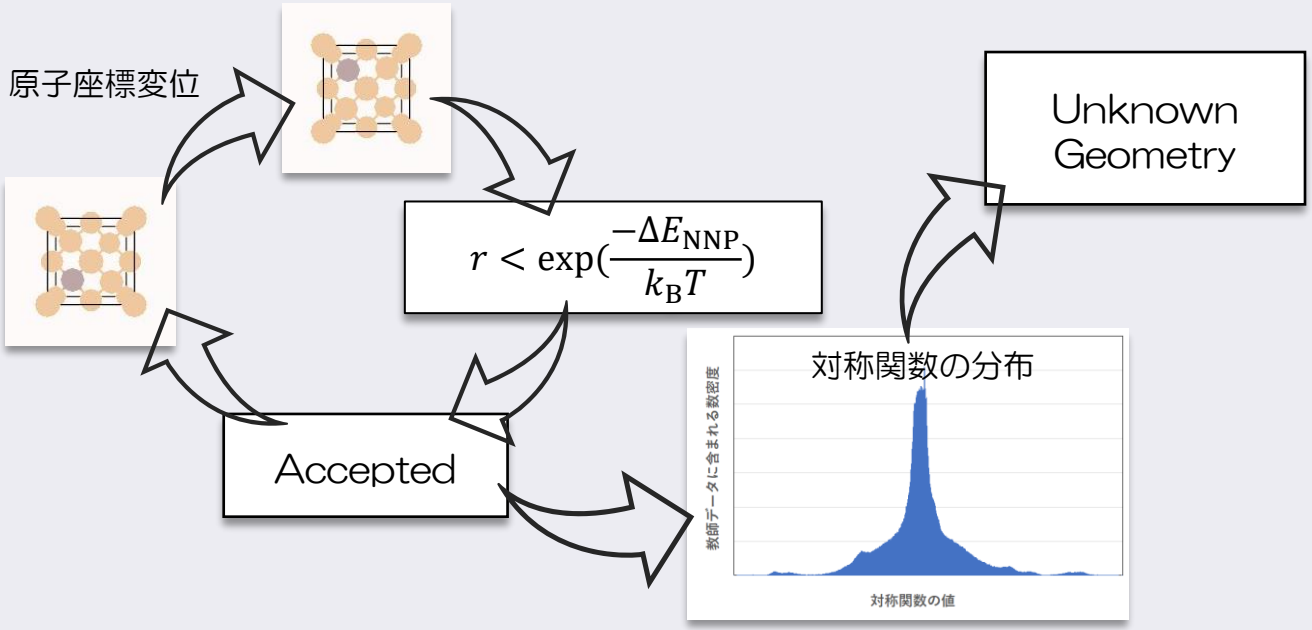
$Li_{10}GeP_2S_{12}$のイオン伝導率を再現することに成功

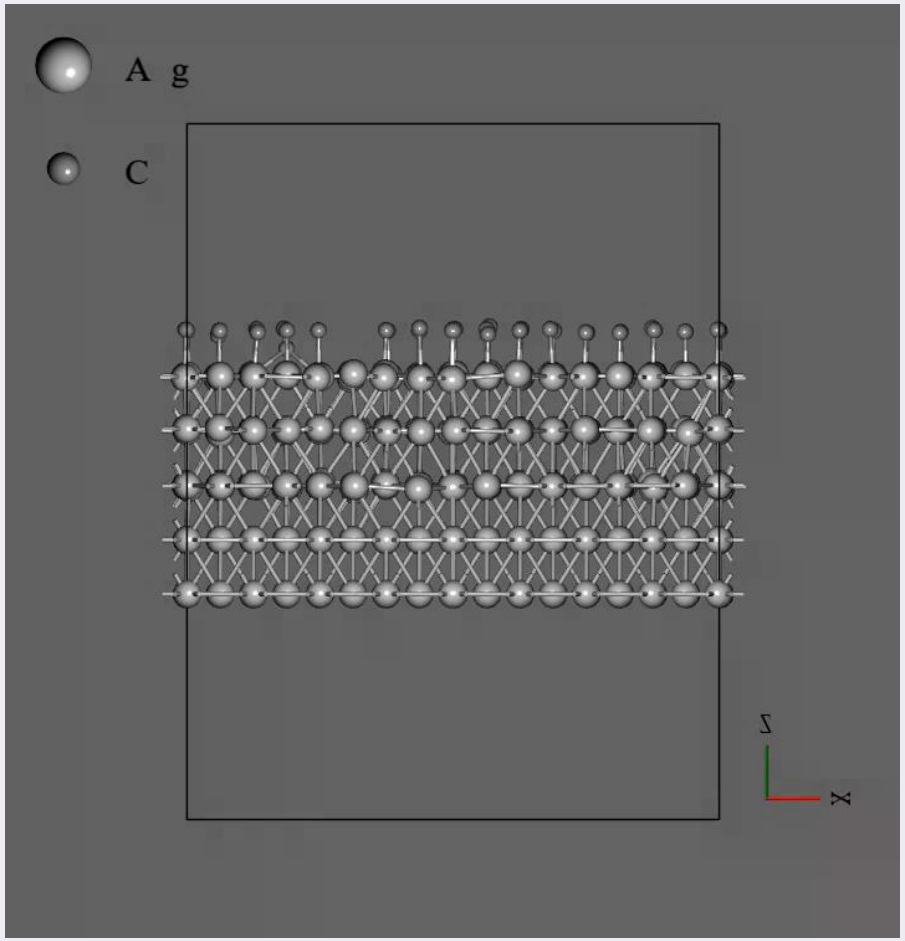


[1] A.Marcolongo, ea al., https://arxiv.org/abs/1910.10090
[2] 菅野了次, Electrochemistry, 85(9), 591–596 (2017)

Metropolis法による未知教師データの生成

- 作成済みのNeural Network力場を利用して、Metropolis法によるモンテカルロ・シミュレーションが実施可能。
  - 原子の並進移動、原子種の入れ替え、空孔位置の入れ替えの操作が利用可能。
- シミュレーションの過程で生成された多数の構造を教師データに追加して、Neural Networkの再学習に使用できる。
  - 対称関数の分布(ヒストグラム)を利用して、未知の構造のみを再学習用に抽出する。





※京都大学、黒川先生よりご提供頂いたデータ。
Ag表面でのC原子の拡散・成長過程。

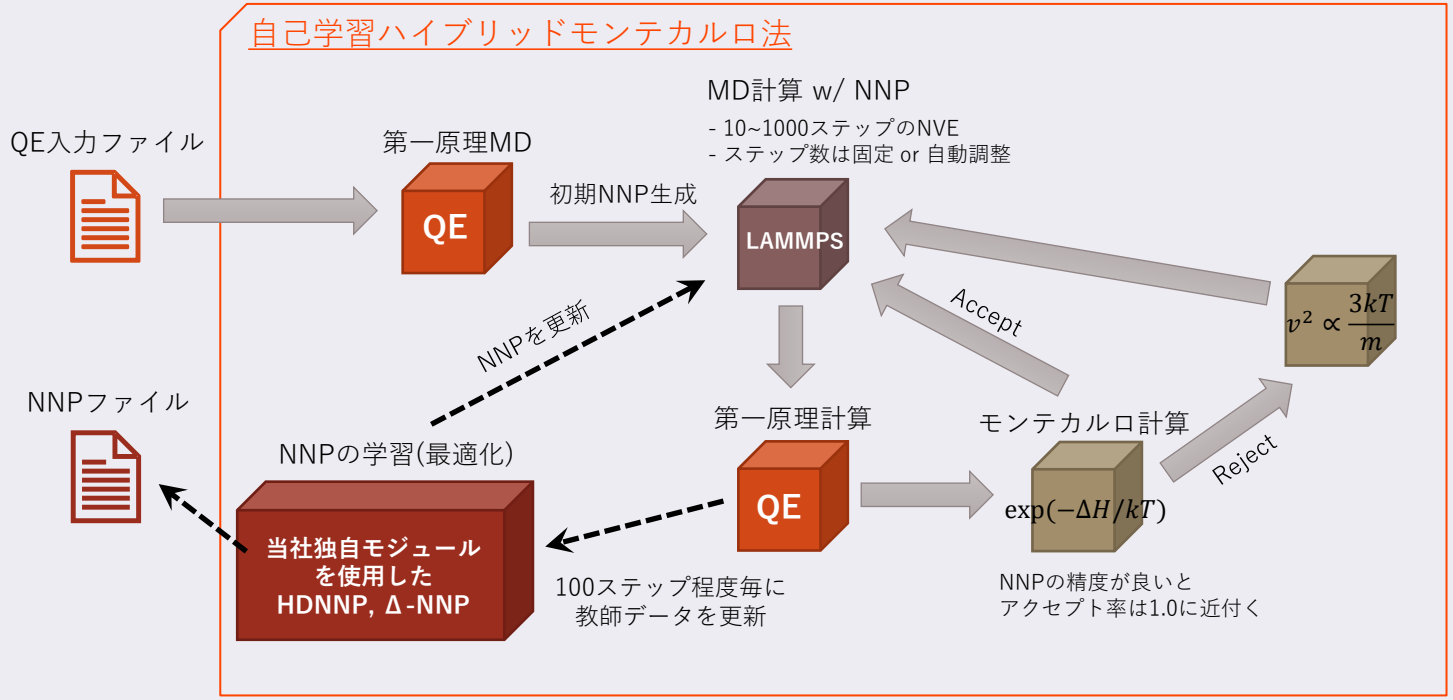
【特徴機能③】

自己学習ハイブリッドモンテカルロ法

“Self-learning hybrid Monte Carlo: A first-principles approach”, Y.Nagai, et al., Phys. Rev. B 102 (2020) 041124

Advance/NeuralMDにおける実装

- 第一原理計算には、Quantum ESPRESSO (オープンソース)を使用
- 分子動力学計算には、LAMMPS (オープンソース)を使用
- Neural Networkの学習については、当社独自実装のモジュールを使用
- モンテカルロ法におけるアンサンブルには、NVTおよびNPTに対応
- アクセプト率に応じて、MD計算のステップ数を自動調整する機能を搭載
- 全ての機能をGUIの画面から操作可能



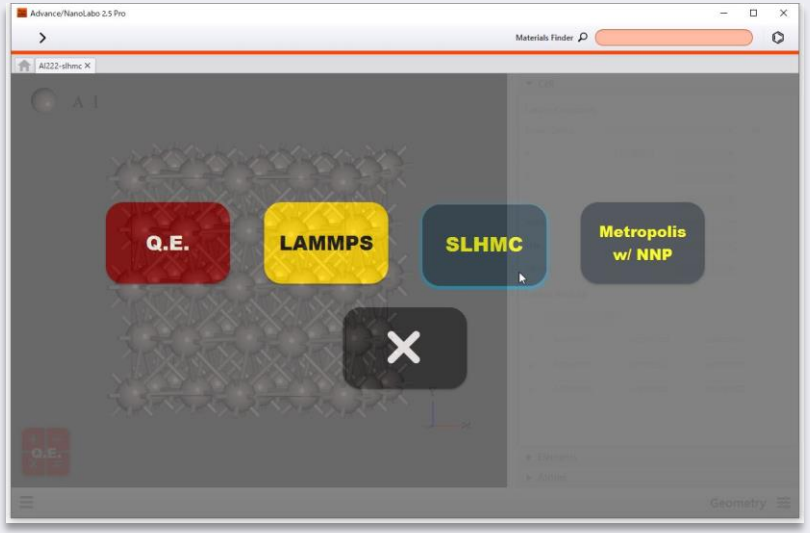
自己学習ハイブリッドモンテカルロ法の計算スキーム。
ユーザーは、QEの入力ファイルを用意するのみでよい。

【特徴機能③】

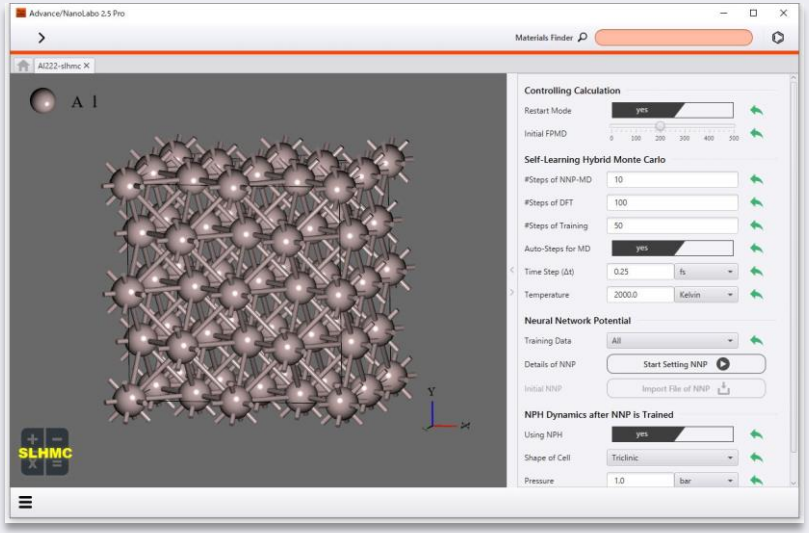
自己学習ハイブリッドモンテカルロ法

Advance/NanoLabo(GUI)からの操作手順

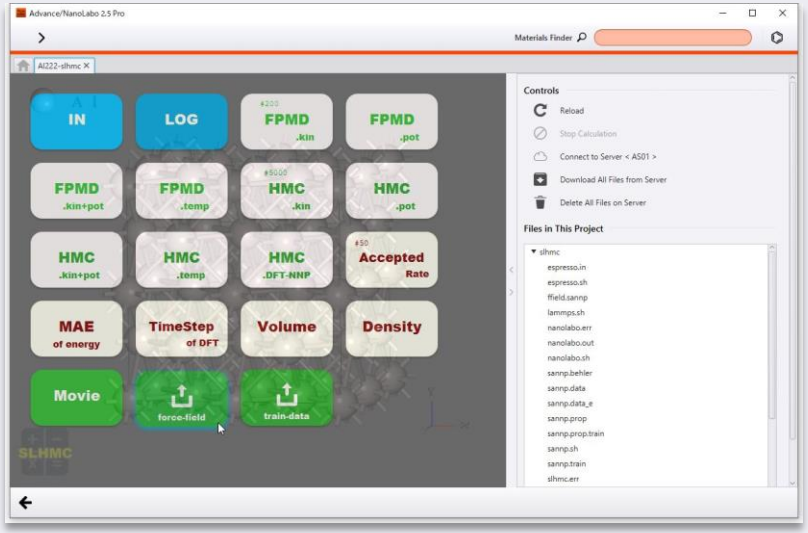
①計算モードを自己学習ハイブリッド
モンテカルロ法(SLHMC)に変更



②計算条件などを設定して計算実行



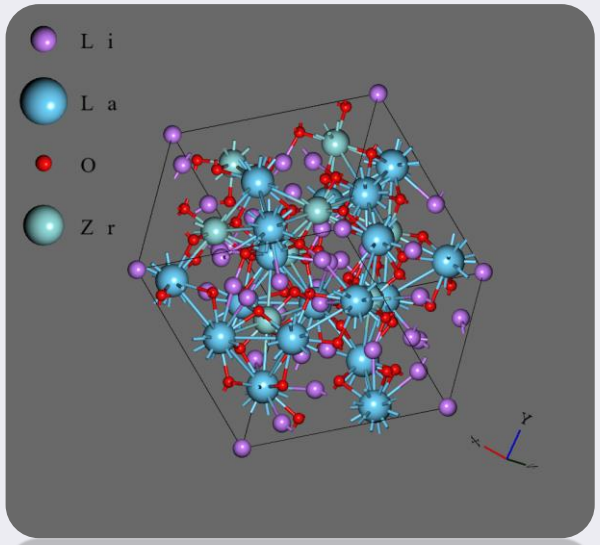
③計算完了後に力場ファイルをエクスポート



計算事例

- 酸化物系Liイオン伝導体： $Li_7La_3Zr_2O_{12}$ (LLZO)
  - NPTアンサンブルで自己学習ハイブリッドモンテカルロ法を実行。
  - 教師データの総数は、2500個
  - 結晶構造は、BCC構造のプリミティブセル
  - 温度は、800K (構造のバリエーションを得るために高めの温度に設定)
  - Intel Xeon Gold 5218 (2CPU, 32コア)を使用して、約50時間で力場作成が完了

LLZOのプリミティブセル



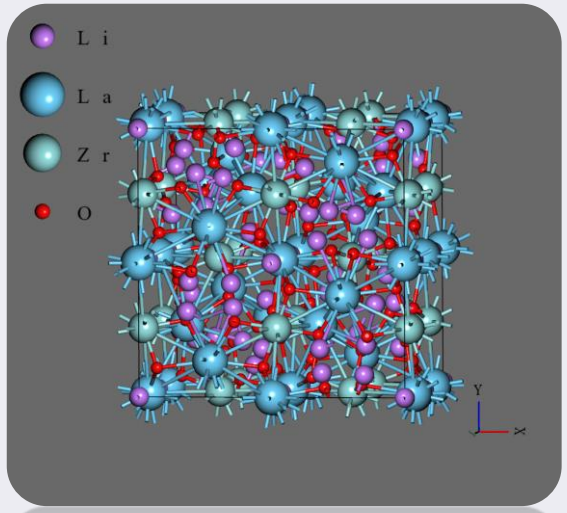
計算条件	
原子数	96
擬ポテンシャル	USPP
カットオフ	36Ry / 324Ry
k点サンプリング	Γ 点のみ
XC汎関数	GGA-PBE
第一原理MD	200ステップ
NNP-MD	10~80ステップ (2.5~20.0fs)
対称関数	Chebyshev (80個, Rc=5 Å)
ニューラルネットワーク	40ノード x 2層

計算事例

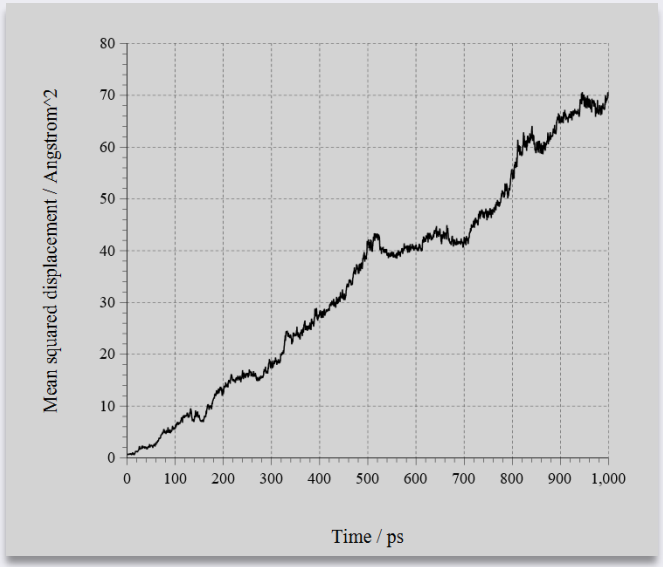
- ユニットセル(192原子系)にて、NNP-MDを実施して拡散係数D(イオン伝導率σ)を評価
- NPTアンサンブル(isotropic)にて、600K、1.0nsのシミュレーション
- BCC構造(立方晶)とBCT(正方晶)でのイオン伝導率の差異を再現

$$\sigma = \frac{\rho e^2 D}{kT}$$

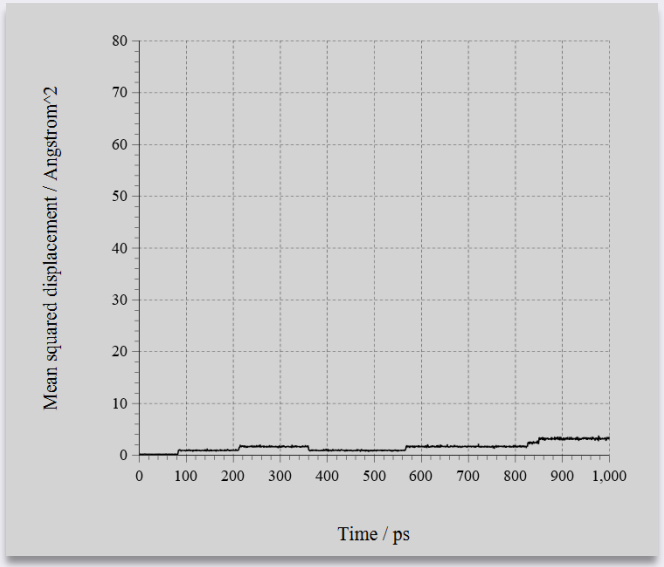
LLZOのユニットセル



BCC構造でのLiのMSD



BCT構造でのLiのMSD



BCC構造の σ / (S/cm)	
Buckingham	8.6 x 10^{-4} †
NNP-MD	9.4 x 10^{-2}
実験値	2.5 x 10^{-1} ‡

BCT構造の σ / (S/cm)	
NNP-MD	4.2 x 10^{-3}
実験値	2.4 x 10^{-2} ‡

† A.Kuhn et al., J. Phys. Chem., 2012, 226, 252
‡ F.Chen et al., J. Phys. Chem., 2018, 122(4), 1963

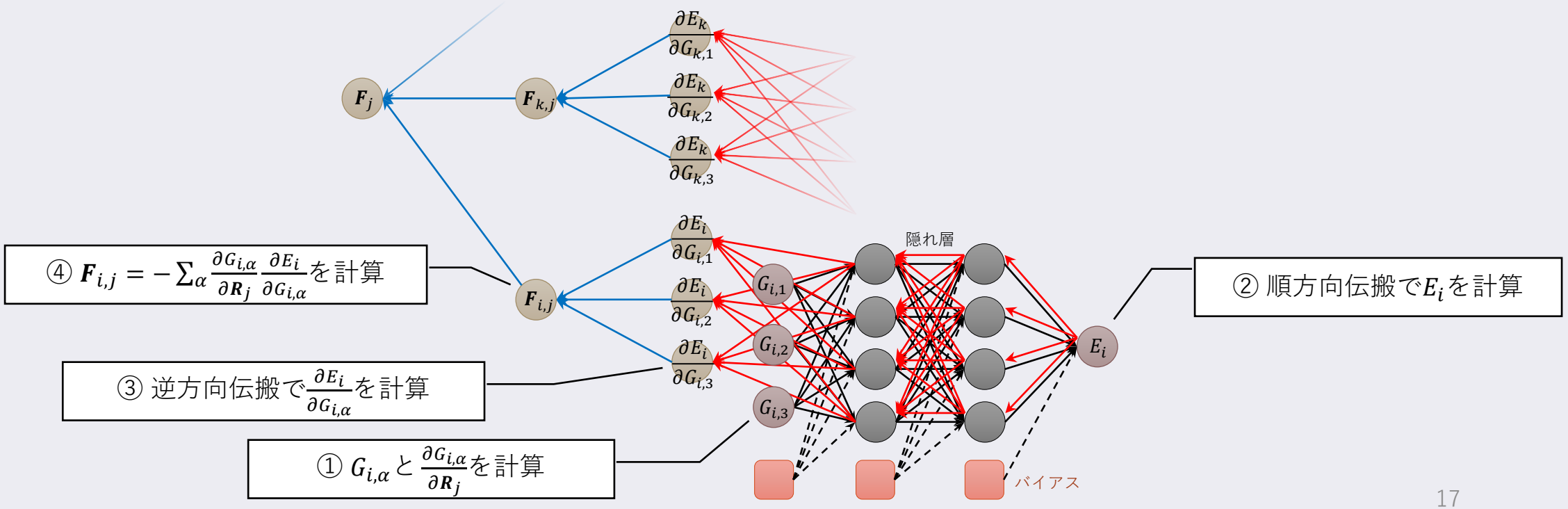
2. Neural Network力場のGPU化

Neural Network力場のGPU化

- Neural Network力場は既存の古典力場に比べて計算コストが高く、大規模系の計算にはGPU化が必須
- Neural Networkの学習過程およびLAMMPSでの分子動力学計算をGPU化
- MPIと併用することで複数のGPUを搭載したマシン and/or GPUを搭載した複数のマシンノードに対応
- GPU１デバイス当たり２～４つのMPIプロセスを起動することで、GPUとCPUの双方の稼働率を常に高い状態に保持できるように設計
- NVDIA A100 x 8 を使用して、Intel Xeon Platinum の約60倍の高速化を実現
- 原子数の制限が無いので、複数GPUを使って数万～10万原子系の計算が可能

Neural Network力場の計算手続き

Neural Network力場の計算では、下図のような多層パーセプトロンを用いた手続きにてエネルギーおよび力を計算する。先ずは、①対称関数$G_{i,\alpha}$および隣接原子座標$\boldsymbol{R}_j$による微分$\frac{\partial G_{i,\alpha}}{\partial \boldsymbol{R}_j}$を計算します。これを②順方向に伝搬させてエネルギー$\boldsymbol{E}_i$を計算した後、③逆方向への伝搬にて微分$\frac{\partial E_i}{\partial G_{i,\alpha}}$を計算します。最後に、④隣接原子間で$\sum_\alpha \frac{\partial G_{i,\alpha}}{\partial \boldsymbol{R}_j}\frac{\partial E_i}{\partial G_{i,\alpha}}$を計算して原子$i$が$j$に及ぼす力$\boldsymbol{F}_{i,j} = -\frac{\partial E_i}{\partial \boldsymbol{R}_j}$を計算します。



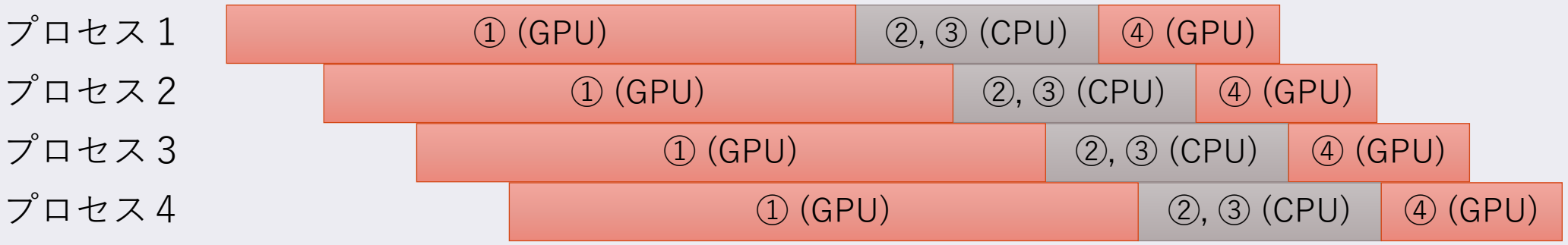
Neural Network力場の計算手続き

さらに、Neural Networkの学習過程においては、⑤力$\boldsymbol{F}_i$の誤差を計算する必要がある。力はNeural Networkの一階微分として計算されるため、その誤差を計算するには二階微分が要求される。この二階微分の計算には多数回のLevel-2およびLevel-3 BLASが実行されるため、④以上の計算コストが掛かり、学習過程における計算時間の90%以上を占める。

	計算プロセス	計算方法	計算コスト	学習過程	LAMMPSでのMD
①	対称関数とその微分	明示的にコーディング	**最大**	最初に1度だけ計算	**MDステップ毎に計算**
②	順方向伝搬	Level-3 BLAS	小	**エポック毎に計算**	**MDステップ毎に計算**
③	逆方向伝搬	Level-3 BLAS	小	**エポック毎に計算**	**MDステップ毎に計算**
④	隣接原子間での力の縮約	Level-2 BLAS	中	**エポック毎に計算**	**MDステップ毎に計算**
⑤	二階微分	Level-2,3 BLAS	大	**エポック毎に計算**	計算しない

LAMMPSによるMD計算のGPU化

- LAMMPSでのMD計算では、各MDステップにおいて①～④の処理を実行する。
- ②と③は比較的に計算量が少ないため、CPUで処理する。
- 対称関数の計算(①)と力の計算(④)をGPU化する。
- MD計算では①が最大のボトルネック(計算時間の95%以上)であるが、対称関数の計算はGPUとの相性が良く大幅な高速化が期待できる。１デバイス当たり２～４プロセスのMPI並列を適用することで計算効率が向上する。
- Neural Network力場の学習過程も、概ね同様の手続を適用している。



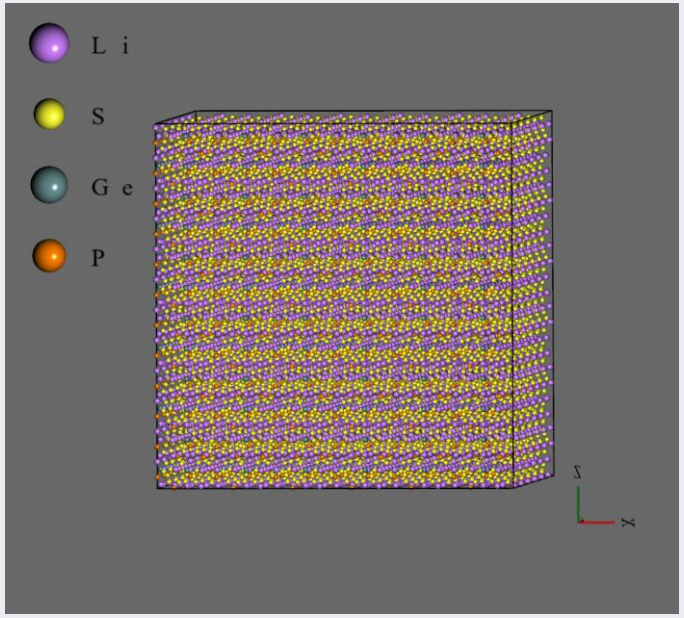
Mat3raでのベンチマーク

Mat3raはExabyte.io社によって提供されているクラウド環境で、第一原理計算や分子動力学計算などのシミュレーションに特化した計算リソースが用意されています(https://mat3ra.com)。今回はAWSクラウド環境にてGPU化されたAdvance/NeuralMDのベンチマークを実施しました。計算リソースの手配にはMat3raの代理店である伊藤忠テクノソリューションズ(CTC)にご協力頂いております(https://ctc-mi-solution.com/exabyte-io/)。

MATERIALS R&D CLOUD

Mat3raでのベンチマーク

Advance/NeuralMDで作成したNeural Network力場を使って、LAMMPSによる分子動力学計算のベンチマークを実施しました。LAMMPSはGCC11.2.0, OpenBLAS, OpenMPI4.1.1, CUDA11.5でコンパイルしています。計算に使用した系は、硫化物リチウムイオン伝導体$Li_{10}GeP_2S_{12}$のスーパーセルモデルです(下図)。原子数は21,600個であり、Neural Network力場を適用するには比較的に大きな系になります。Neural Network力場および分子動力学計算の計算条件は下表の通りです。100ステップのMD計算実施後に計算時間を計測しています。



計算条件	設定値
対称関数の種類	Chebyshev多項式
対称関数の動径成分	50個
対称関数の角度成分	30個
カットオフ半径	6.0 Å
Δ-NNP法	適用有り
NNの構造	2層 x 40ノード (twisted tanh)
アンサンブル	NVT (T=500K)
時間刻み	0.5fs
MDステップ数	100

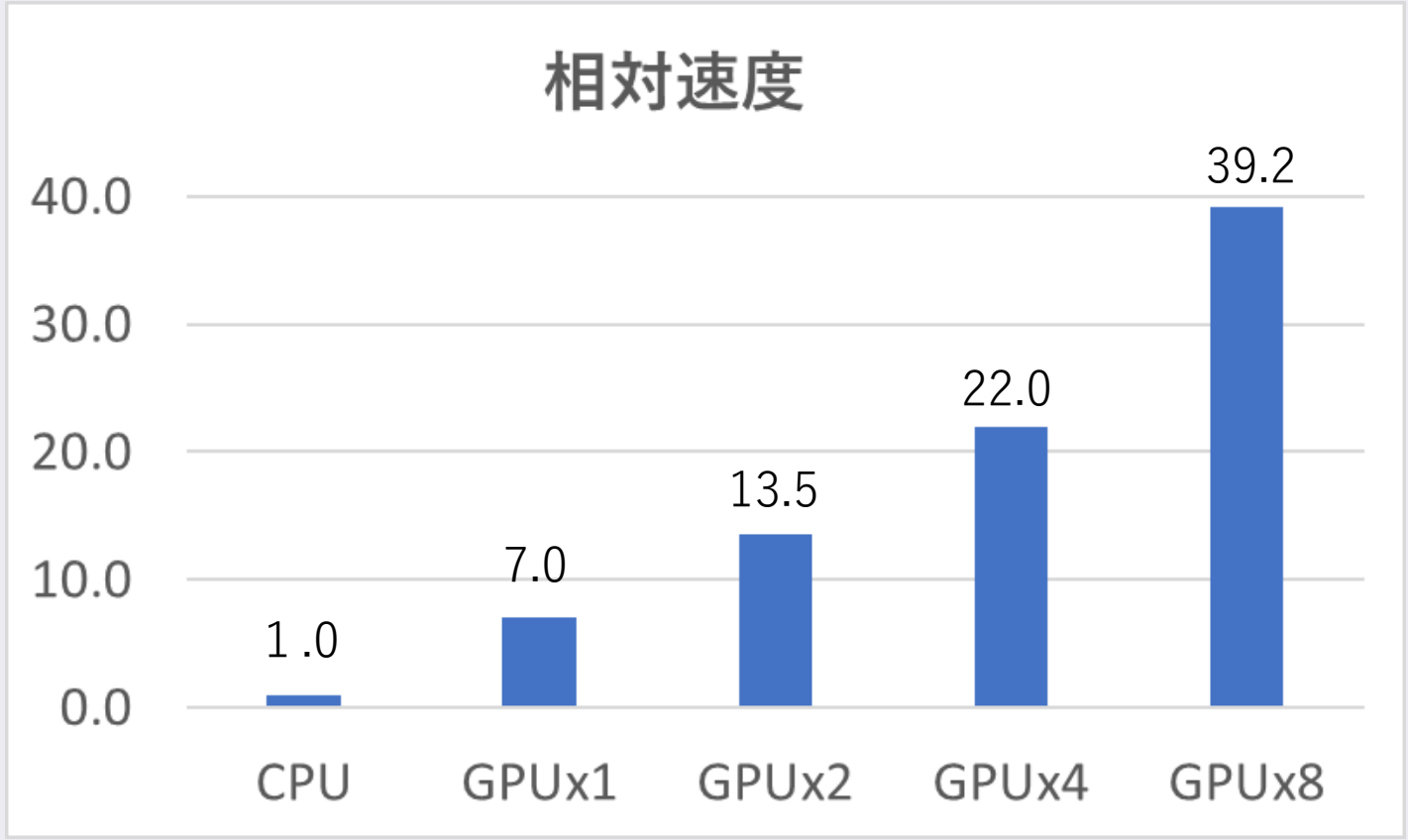
Mat3raでのベンチマーク

AWSでの計算条件および計算時間は下表の通りです。CPUのみ、および、GPUを１～８デバイス使用した場合の5ケースにて計算しています。１GPUデバイス当たり、4つのMPIプロセスを起動しています。

	CPUのみ	GPU x 1	GPU x 2	GPU x 4	GPU x 8
ジョブQueue	OFplus	GOF	G4OF	G4OF	G8OF
CPU種別	**Intel Xeon Platinum / 72core**	Intel Xeon E5-2686-v4			
GPU種別	-	**NVIDIA V100**			
MPIプロセス数	72	4	8	16	32
OpenMPスレッド数	1	2	2	2	2
GPUデバイス数	-	1	2	4	8
計算時間 / sec	150.8	21.6	11.2	6.9	3.9

Mat3raでのベンチマーク

CPUでの計算時間を１とした相対計算時間を下図に示す。

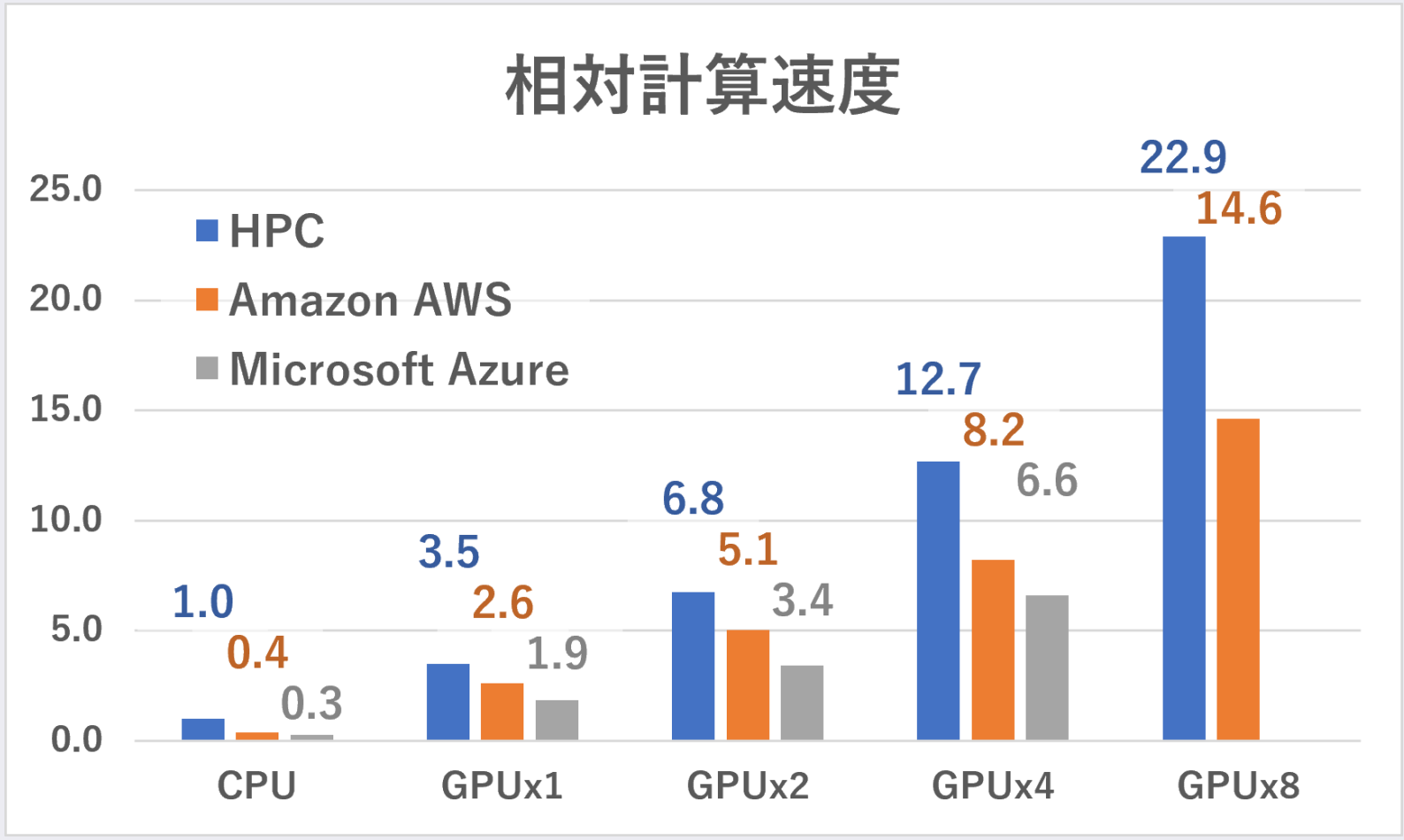


NVIDIA A100搭載マシンでのベンチマーク

- HPCシステムズ(https://www.hpc.co.jp)にご提供頂いた、NVIDIA A100 x 8 搭載マシンにて同様のベンチマークを実施
- 計算対象および条件は、先ほどと同様のLGPSスーパーセルモデル(21,600原子系)

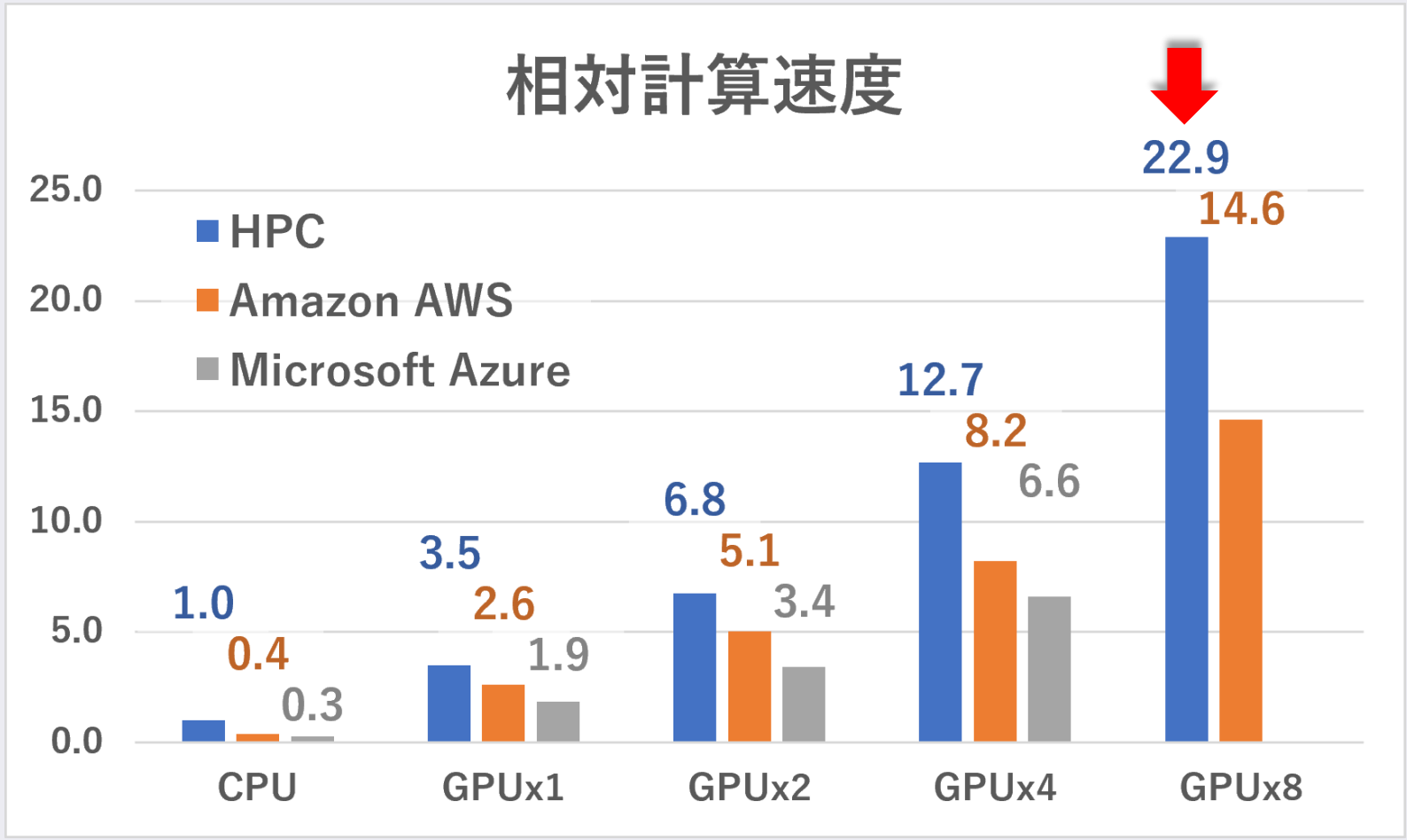
	CPUのみ	GPU x 1	GPU x 2	GPU x 4	GPU x 8
CPU種別	AMD EPYC 7742 / 64core	AMD EPYC 7742 / 64core			
GPU種別	-	NVIDIA A100			
MPIプロセス数	64	4	8	16	32
OpenMPスレッド数	1	2	2	2	2
GPUデバイス数	-	1	2	4	8
計算時間 / sec	56.4	16.0	8.3	4.4	2.5

NVIDIA A100搭載マシンでのベンチマーク



NVIDIA A100搭載マシンでのベンチマーク

AWSのIntel Xeon Platinumと比較すると
NVIDIA A100 x 8 により**57.3倍**の高速化を実現！



ベンチマーク結果のまとめ

- AWSではGPU(NVIDA V100)１デバイスで、Intel Xeon Platinumの7.0倍ほど高速化された。
- GPUのデバイス数を増やすと計算速度は十分に良くスケールすることが確認された。
- 今回のベンチマークに使用した2万原子系ではGPUが4デバイスほどあれば十分に実用的な計算が実施できることが確認できた。原子数が5000以下であれば、GPU１デバイスでも十分高速にMD計算が実施できるものと推測される。
- A100を使用すれば、さらに1.5倍ほど高速化される。A100を多数搭載した計算環境があれば、数万～10万原子系の計算も十分に可能。

3. Neural Network力場とヤコビの梯子

密度汎関数理論 (DFT)

- 密度汎関数理論(DFT : Density Functional Theory)とは、
  基底状態エネルギーEを電子密度$\rho(\mathbf{r})$の汎関数として定義すること。
  多体波動関数を明示的に含まないので、旧来の波動関数理論よりも計算コストが低い

$$E = E[\rho]$$

- 汎関数による記述可能性については理論的に厳密に正しいのだが、
  汎関数の具体的な関数形が未知である。

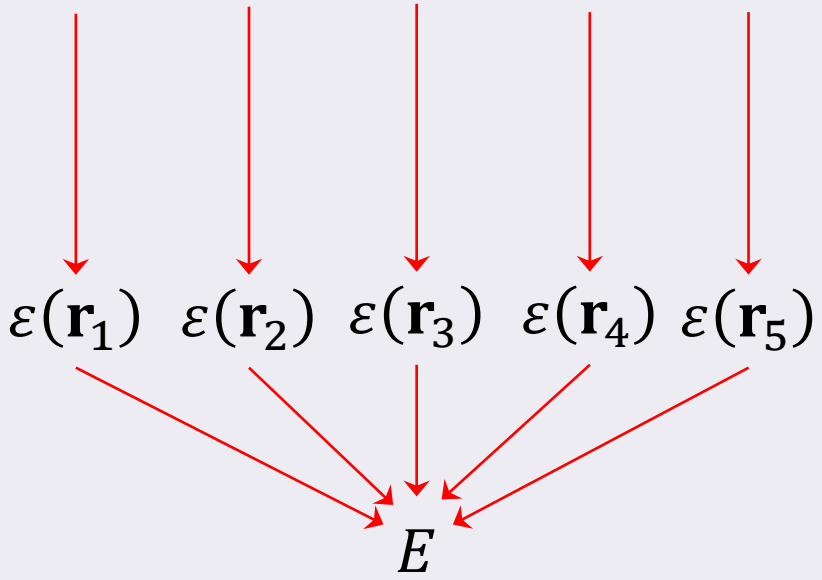
- エネルギー汎関数に近似的な関数形を適用する。
  局所密度近似LDA、一般化密度勾配近似GGA、ハイブリッド汎関数など

密度汎関数理論 (DFT)

- 局所密度近似LDA

$$E = E[\rho] = E(\rho(\mathbf{r}_1), \rho(\mathbf{r}_2), \rho(\mathbf{r}_3), \rho(\mathbf{r}_4), \rho(\mathbf{r}_5), \cdots)$$

$\rho(\mathbf{r}_1)$のみから$\varepsilon(\mathbf{r}_1)$が決まる

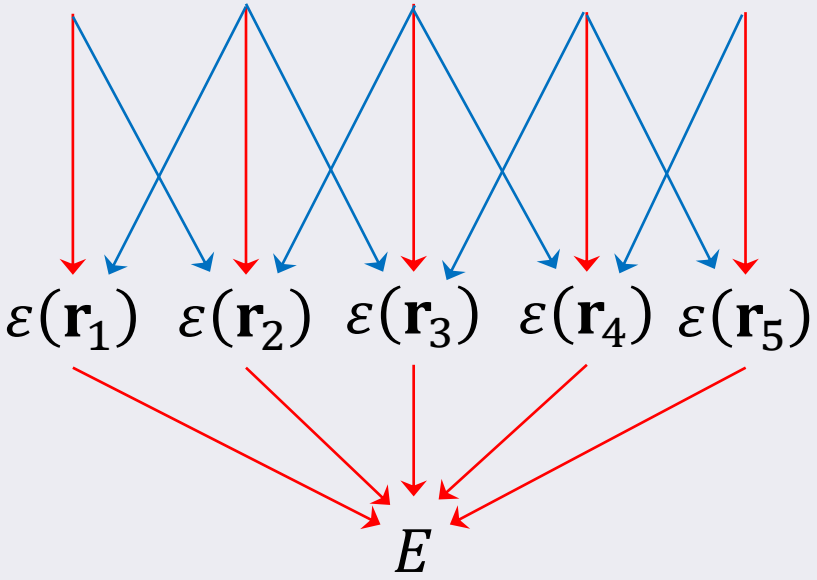


密度汎関数理論 (DFT)

- 一般化密度勾配近似GGA

$$E = E[\rho] = E(\rho(\mathbf{r}_1), \rho(\mathbf{r}_2), \rho(\mathbf{r}_3), \rho(\mathbf{r}_4), \rho(\mathbf{r}_5), \cdots)$$

$\rho(\mathbf{r}_2)$と隣接する$\rho(\mathbf{r}_1), \rho(\mathbf{r}_3)$のみ
から$\varepsilon(\mathbf{r}_2)$が決まる

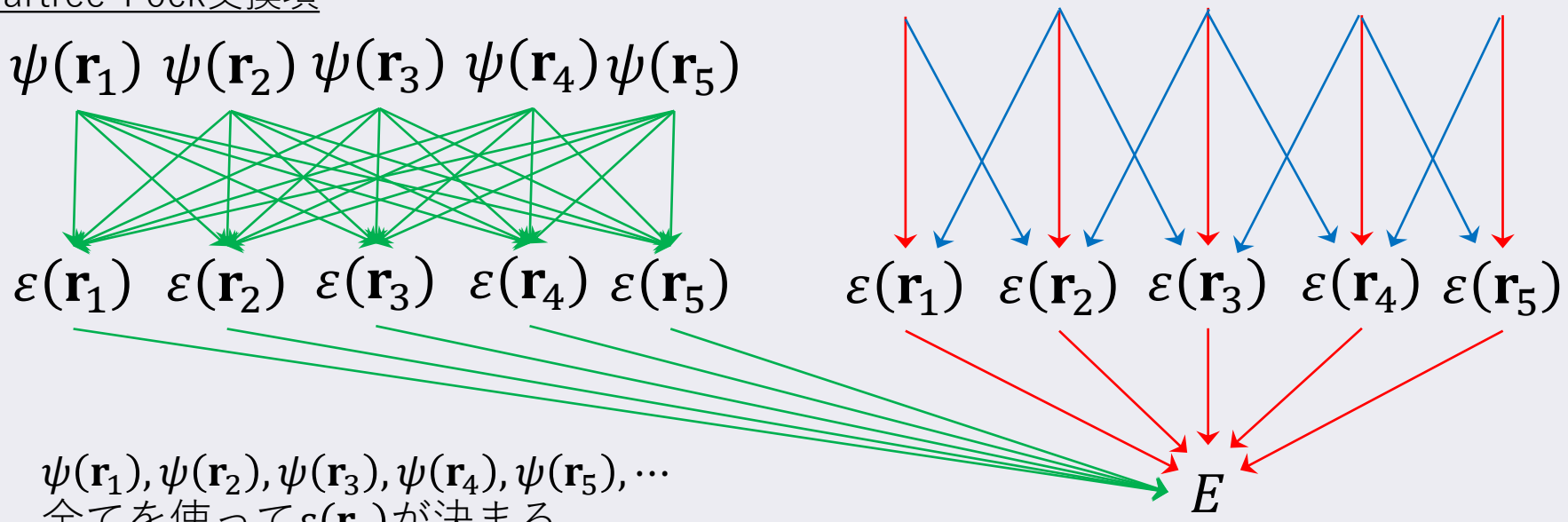


密度汎関数理論 (DFT)

- ハイブリッド汎関数

$$E = E[\rho] = E(\rho(\mathbf{r}_1), \rho(\mathbf{r}_2), \rho(\mathbf{r}_3), \rho(\mathbf{r}_4), \rho(\mathbf{r}_5), \cdots)$$

Hartree-Fock交換項



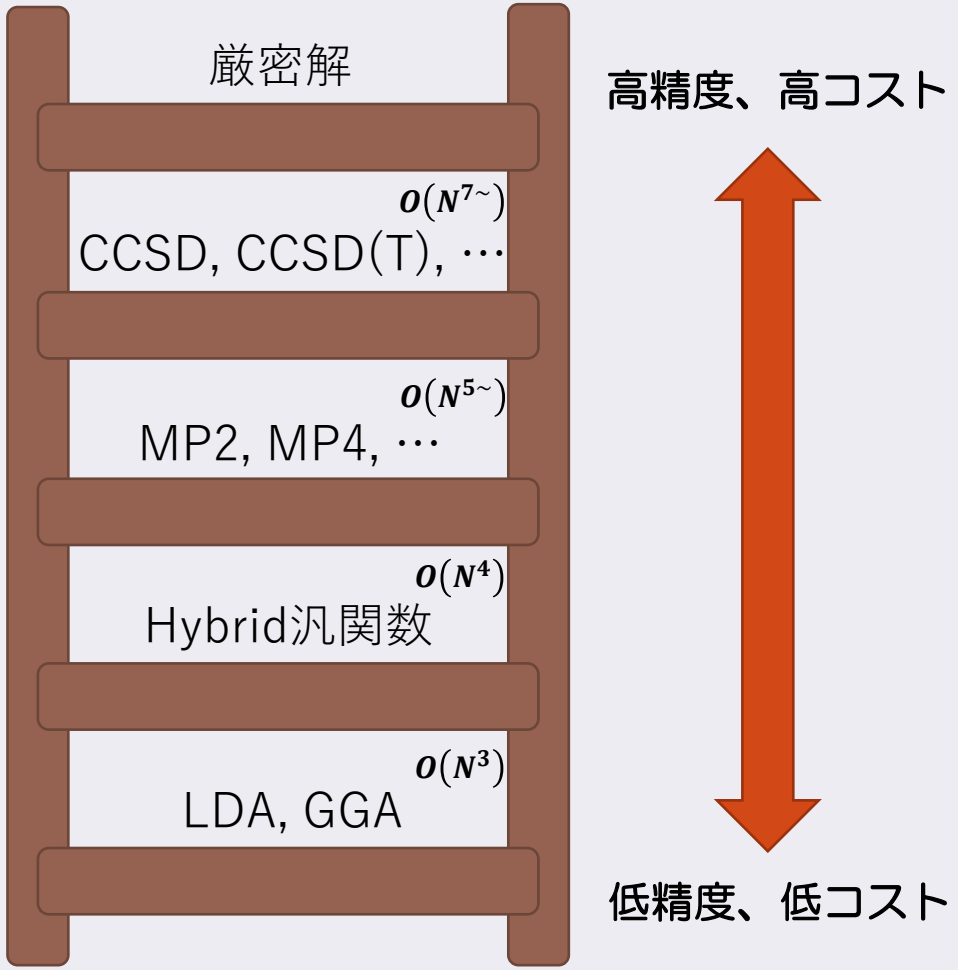
$\psi(\mathbf{r}_1), \psi(\mathbf{r}_2), \psi(\mathbf{r}_3), \psi(\mathbf{r}_4), \psi(\mathbf{r}_5), \cdots$
全てを使って$\varepsilon(\mathbf{r}_1)$が決まる

エッジを1/r や erfc/r に固定した
グラフ畳み込みと見なせる。

ヤコビの梯子

- LDAやGGAは計算コストが低い。ただし計算精度も低い(バンドギャップの過小評価など)。
- Hybrid汎関数を適用すると計算精度は向上するものの、計算コストも増大する。
- さらに高精度化しようとすると、MP2やCCSDなどの波動関数理論で計算されたエネルギーを混ぜ合わせたDouble-Hybrid汎関数となる。
- DFTの強みは、波動関数理論よりも低コストであることだったはず。しかしながら、精度を追求することで波動関数理論と何ら変わらない計算コストに達する。
- DFTも波動関数理論も計算機上のアルゴリズムの１つである。本来$O(N!)$の厳密解を、精度を保ったまま、より低いコストで解くことはできない。

  ⇒　とどのつまり P≠NP問題



Neural Network力場の場合

- DFT vs 波動関数理論
  - 高コストな波動関数理論を低コストなDFTで代替しようとした。
  - 計算精度を犠牲にすることで、DFTによる高速化を実現した。
  - Hybrid汎関数で高精度化を図るも、計算コストも上昇。
  - 精度とコストの両立は不可能であった。

- NN力場 vs DFT
  - 高コストなDFTを低コストなNN力場で代替しようとしている。
  - 計算精度または汎用性を犠牲にすることで、NN力場による高速化を実現している。
  - グラフニューラルネットワークを応用することで高性能化を図る手法も登場しているが、その分だけ計算コストも増大している。
  - 機械学習を用いたとしても、精度とコストの両立は不可能か？

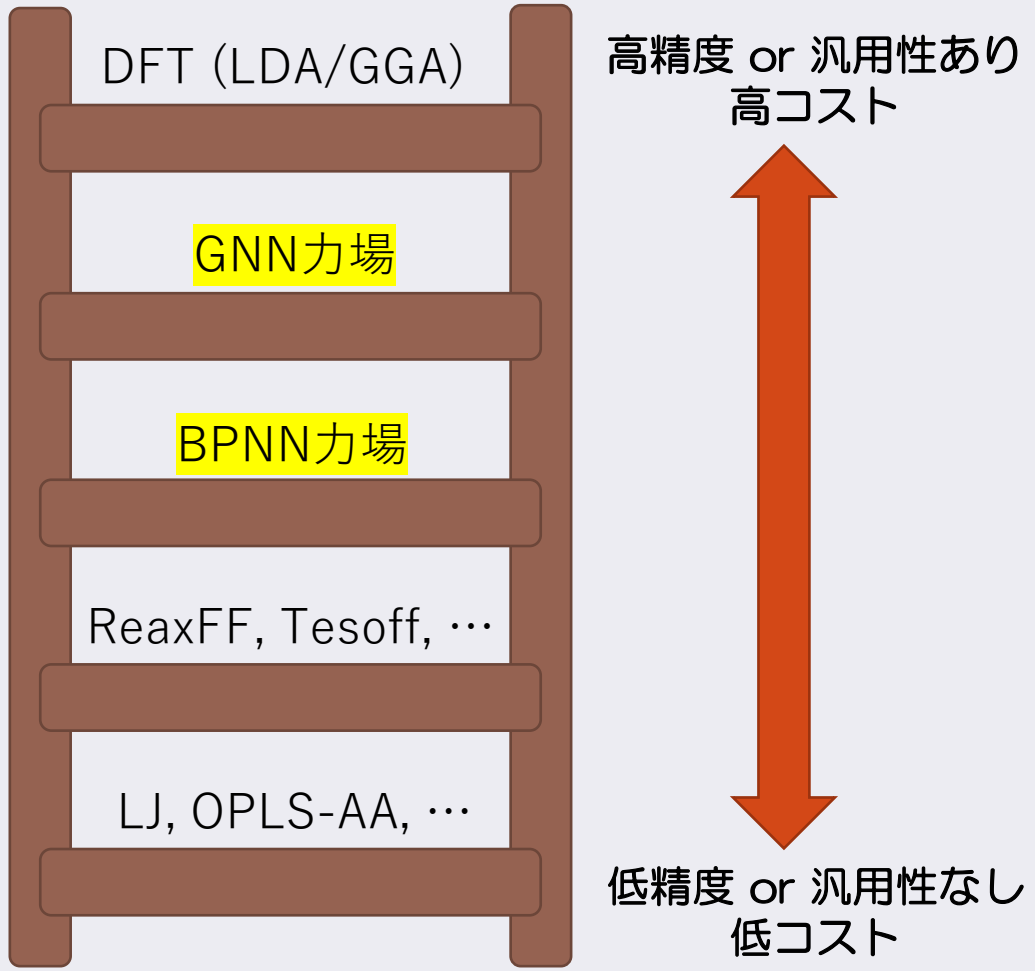
Neural Network力場の場合

- DFTにおけるヤコビの梯子の問題を、力場法まで拡張して解釈してみる。

- これまではDFTと古典力場の間には大きな隔たりがあったが、この状況を補間する手法してNN力場を位置付けることができる。
  - DFT：1～100原子
  - NN力場：100～10,000原子
  - 古典力場：1,000～1,000,000原子

- 力場の場合には、精度とコストに加えて**汎用性**も重要なファクターとなる。

- BPNN力場とGNN力場で状況が異なる。



BPNN力場とGNN力場

- Behler-Parrinello Neural Network (BPNN)力場
  - 原子の化学環境を表現した対称関数を多層パーセプトロンに入力して、原子エネルギーを出力する。
  - 実装の容易さからC++やFortranで開発されることが多い。
  - MD計算実行時にMPI並列が利用可能で、100～10,000原子系の計算が可能。
  - 個別の系に対してユーザーが力場を適宜作成する必要があり、汎用性はない。
  - **aenet**, **n2p2**, **DeepMD**, **Advance/NeuralMD** など

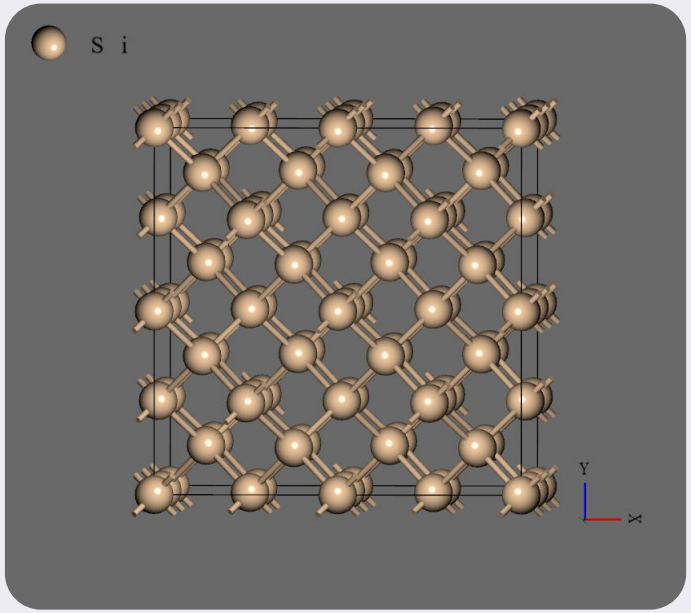
- Graph Neural Network (GNN)力場
  - 原子間の結合の情報をグラフニューラルネットワークとして表現する。原子エネルギーに加えて、結合エネルギーを露わに表現できるためBPNN力場に比べて非常に高性能。
  - Pythonの自動微分を活用しなければ実装は困難。
  - MD実行時もPythonを使用しており、MPI並列は利用不可。GPU 1 デバイスのみでの高速化。
  - 最新のGPUを使えば 100～1,000原子程度の計算は可能。
  - 事前に大量の教師データ(数百万～数億)で学習済みのものが公開されており、ユーザーが自身で力場を作成する必要がない。⇒ **Open Catalyst**, **M3GNet** など

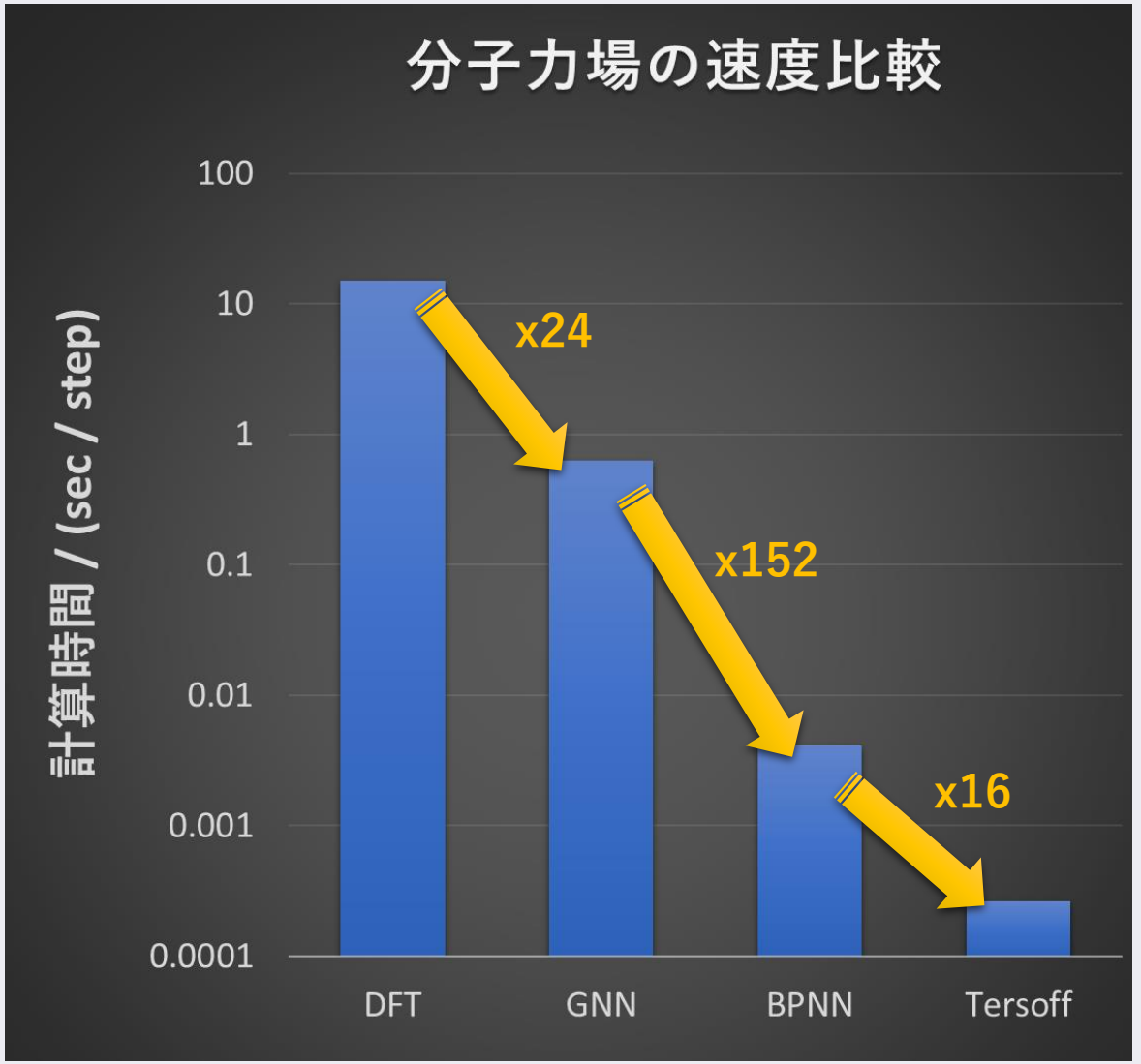
BPNN力場とGNN力場

- 計算速度のベンチマーク

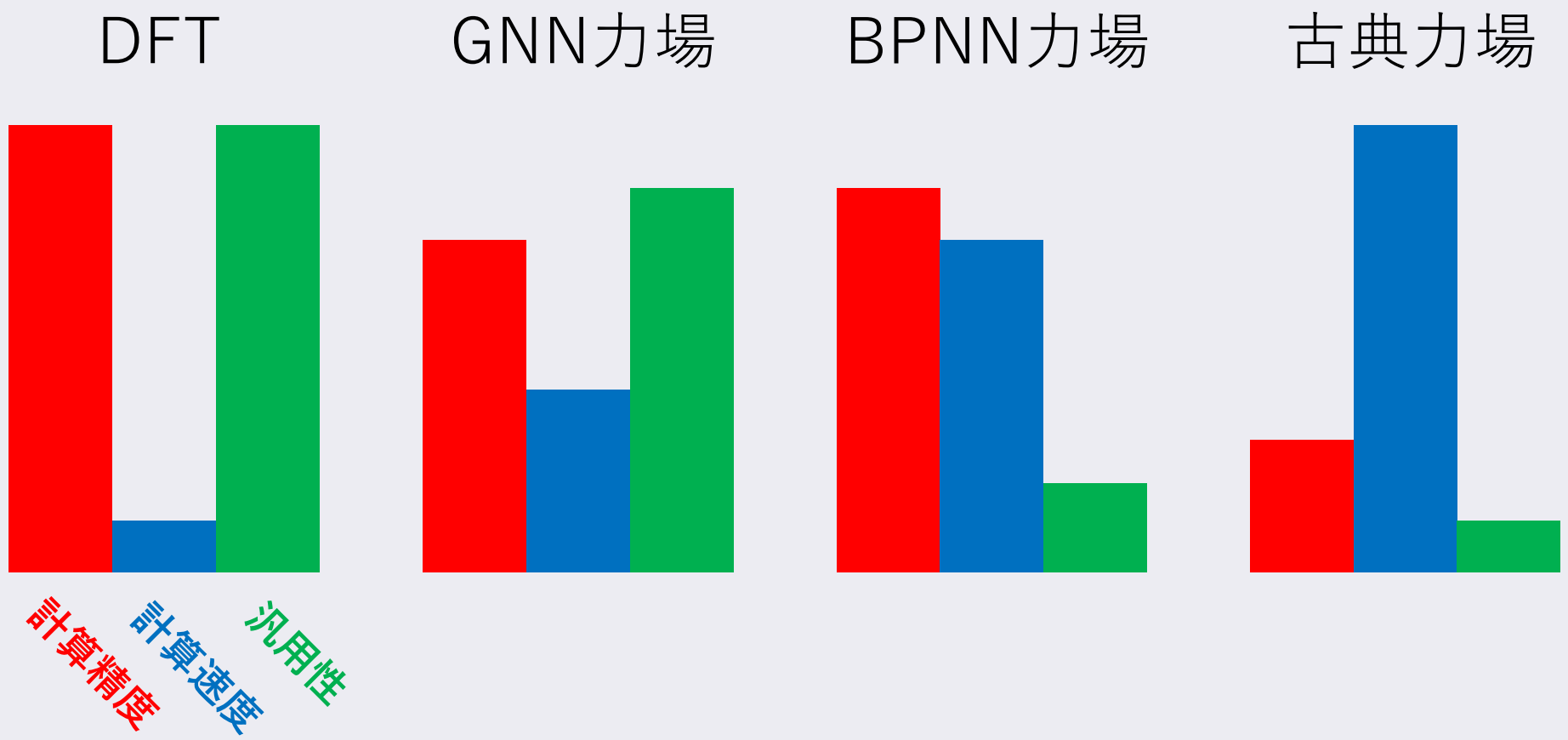
  ✓ 計算対象
  Siスーパーセルモデル (64原子系)

  ✓ マシンスペック
  CPU：Inter® Core™ i7-10700 @ 2.90GHz (8コア)
  OS　：Windows10





計算精度／計算速度／汎用性



計算精度／計算速度／汎用性

- BPNN力場は、古典力場の高精度化と見なせる。
  ⇒　従来の古典力場では不可能であった
  定量的な物性値の予測に適している。

- GNN力場は、DFTの高速化と見なせる。
  ⇒　DFTによる大量のデータ生成が必要なMIにおいて、
  GNN力場で代替することで大きな高速化が期待できる。

今後の予測

- 最近では電子状態を模倣したNN力場も考案されつつあるが、DFTに匹敵する計算コストになることが予測される。現行のGNN程度に留めておくのがベストである。
- Python(特に自動微分)のおかげで、当該分野の研究開発のスピードが極めて速い。1～2年のうちにGNN力場の決定版的なものが公開されるはず。公開された力場をMIなど活用するためのプラットフォーム作りが重要。
- BPNN力場は、MDシミュレーション技術の一部として定着する。効率的な力場作成の手法、および、力場作成～MD計算までの工程を使い易くしたユーザーインターフェースが重要。
- GNN力場でMD計算を行い、そのトラジェクトリを教師データとしてBPNN力場に焼き直す手法の開発。

ご清聴ありがとうございました。