jh240028

CPU と GPU を同時利用する自己学習モンテカルロ法の開発

永井佑紀（東京大学情報基盤センター学際情報科学研究部門）

本研究は、東京大学情報基盤センターの Wisteria/BDEC-01 を用いて、CPU と GPU を同時に利用する自己学習モンテカルロ法（SLMC）の開発を目指したものである。分子動力学法 PIMD に対して、CPU と GPU の並列実行を可能にするコード改修を行い、aenet-PyTorch を導入して高速な機械学習ポテンシャルを実現した。さらに、同変トランスフォーマーを導入し、有効模型の精度向上と長距離相関の捕捉に成功した。加えて、水や重水などの軽元素に対する核量子効果を再現するために、自己学習ハイブリッドモンテカルロ法（SLHMC-MIX）も開発した。この手法は、第一原理ポテンシャル（FP）と機械学習ポテンシャル（MLP）を線形結合した混合ポテンシャルを用いることで、従来の FP-PIMD に比べて大幅な計算効率向上と高精度な構造再現を実現した。これにより、広範な材料系への応用が期待される。

1. 共同研究に関する情報

(1) 共同利用・共同研究を実施している拠点名

東京大学 情報基盤センター

(2) 課題分野

大規模計算科学課題分野

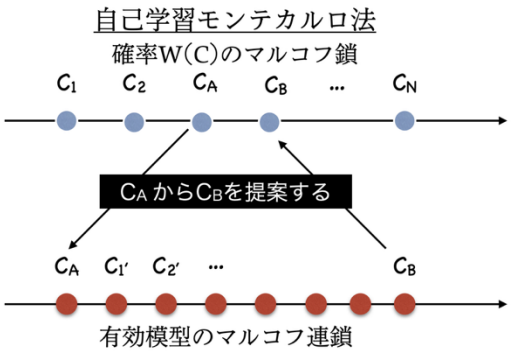
(3) 参加研究者一覧と役割分担

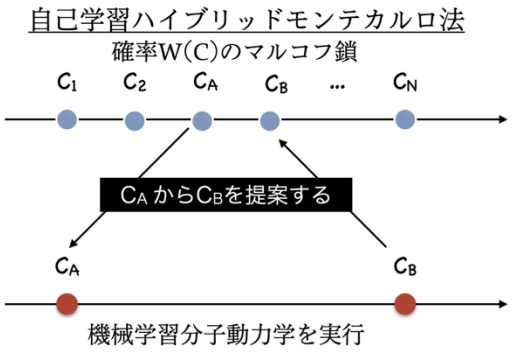
永井佑紀：研究計画の立案および遂行
奥村雅彦：SLHMC の物理学的観点からの議論
住元真司、中島研吾、荒川隆：Wisteria の CPU+GPU 連携パートの技術的観点からの情報提供および計算効率化に関する議論

2. 研究の目的と意義

自己学習モンテカルロ法(Self-learning Monte Carlo, SLMC)は、マルコフ連鎖モンテカルロ法の一種であり、マルコフ連鎖において次の配位の候補を生成する際、有効模型によるマルコフ連鎖（自己学習モンテカルロ法）や分子動力学法（自己学習ハイブリッドモンテカルロ法）を用いて、候補を生成する手法である（図参照）。この方法は、「機械学習を使っているにも関わらずシミュレーションの精度が元のシミュレーションによるものと等しい」という著しい特徴を持っている。この特徴のおかげで、構成する有効模型の精度に左右されずにシミュレーションを安心して遂行でき、物理量の精度を保証しつつ劇的な高速化に成功している。また、「シミュレーションの最中に学習が実行でき、逐次的にモデルを改善することができる」という特徴もあり、シミュレーションと機械学習を同時に実行できるという意味で東京大学のスーパーコンピュータ Wisteria/BDEC-01 の「「計算・データ・学習」融合スーパーコンピュータシステム」のコンセプトとの親和性が高い。

この手法自体は MCMC を使う様々なシミュレーションに適用することができ、これまで、原子・分子系における機械学習分子動力学法、高エネルギー物理学分野における格子量子色力学、強相関電子系における量子モンテカルロ法など、様々な分野に適用してきた。また、機械学習分子動力学分野においては、オ





ープンソースの分子動力学法パッケージ PIMD に SLMC 機能を追加し、多くの人が SLMC を用いたシミュレーションが可能になるように研究開発を行ってきた。そして、PIMD は、現在、大学・企業問わず様々なグループが研究開発に実際に利用している状況である。一方、SLMC は、シミュレーションの最中に有効模型を徐々に改良していく手法であるため、「オリジナルの重たいコードを実行するシミュレーションパート」と「有効模型を機械学習するパート」の二つの異なる種類の計算パートが存在している。また、「有効模型を機械学習するパート」に関しては、世界中の様々な研究グループが研究開発を行っており、その際に GPU を利用することで高速に学習が実現できることが知られている。そして、第一原理分子動力学を始めとするシミュレーションパートは、マルチ CPU コアによる並列計算が効力を発揮することが知られている。しかし、二つの異なる計算パートを組み合わせて使用するコードを開発することは、計算物理学者だけでは難しく、計算科学者との協力が不可欠である。そこで、本申請では、Wisteria/BDEC-01 の Odyssey と Aquarius の 2 つの計算機を同時使用する SLMC コードを開発することを目標とする。その際、Odyssey と Aquarius 間の通信を利用して、マルチコア CPU シミュレーションと GPU による機械学習を両立させ、一つの SLMC シミュレーションとして統合されたコードを開発する。

特に、最も利用者が多いと見込まれる機械学習分子動力学シミュレーションに着目し、既存の PIMD パッケージに CPU-GPU 通信を実装しつつ、GPU に対応した様々なオープンソース機械学習有効模型作成コードを連携させる。このコードの開発が実現できれば、SLMC による効率的な機械学習シミュレーションを高速に実施できるようになり、より多種多様な物質群（電池材料、熱電材料、その他高機能物性材料）の理論設計とスクリーニングが可能となり、研究開発が加速されることになる。

3. 当拠点の公募型共同研究として実施した意義

SLMC は全ての MCMC に原理的には適用可能であり、効率的な機械学習を行う手段としても有用である。そのため、物理学分野に限らず幅広い範囲に適用可能である。また、SLMC は本質的に二つの計算パートから成り立っていることから、シミュレーションと機械学習を融合させるコードを開発することが必要不可欠であり、コード開発の知見はこれら「シミュレーションと機械学習の融合」関連の研究を発展させる可能性がある。

上記の観点から研究を遂行した結果、Wisteria における CPU と GPU 連携に関する知見が得られた。連携のために開発したコードは今後コードを整理してオープンソースソフトウェアとして公開予定である。

4. 前年度までに得られた研究成果の概要

該当なし

5. 今年度の研究成果の詳細

オープンソースの分子動力学法パッケージである PIMD (Path Integral Molecular Dynamics)は、主に量子力学的効果を取り入れたシミュレーションを実施するためのもので、CPU と GPU を連携させて効率的に動作させることは元々想定されていない。特に、PIMD の SLMC (Self-Learning Monte Carlo) コードに関しては、以下のような CPU ベースのシステムとして構築されている。

・ オリジナルの重たい計算を実行するシミュレーションパート

系全体の量子運動やエネルギー計算を行う部分。計算負荷が大きい。

・有効模型 (effective model) を機械学習するパート

シミュレーションデータを元に、より効率的なポテンシャルモデルを学習する部分。こちらも CPU で動作するよう設計されている。

[コードの解析と修正] このように、PIMD の従来のコード構造では、シミュレーションパートと機械学習パートが同一の CPU リソース上で逐次実行される形となっており、これが GPU を活用した効率的な計算を妨げている要因となっていた。そこで、以下の 2 つのステップでコードの解析と修正を行った。

1. コードの分離
   まず、PIMD のコードを解析し、シミュレーションパートと機械学習パートを明確に分離するようにコード構成を修正した。これにより、各パートが独立して実行されるようになり、CPU と GPU の並列活用が可能となるようにした。
2. 簡易連結コードの作成
   次に、WaitIO を用いた簡易的な連携コードを作成。これにより、Odyssey (CPU クラスタ)と Aquarius (GPU クラスタ)間で協調的にジョブを実行できるようにし、計画されていた**「CPU-GPU 連携テスト」**を実施した。

[aenet-PyTorch との連携実装] PIMD には元々、aenet (The Atomic Energy Network) との連携が組み込まれている。しかし、この従来の SLMC コードは、aenet の CPU 版のみを対象としており、GPU での高速な機械学習を行うことができなかった。そこで、PyTorch を基盤とする aenet-PyTorch (https://github.com/atomisticnet/aenet-pytorch) を PIMD に統合するために、以下のような拡張と修正を行った。

・PIMD コードの改変

PIMD と aenet-PyTorch の双方のソースコードを改変し、相互にデータを効率的にやり取りできるように修正した。特に、機械学習ポテンシャルの学習部分を GPU で高速に処理することを目指した。

・非同期実行の実現

以前は、機械学習パートの計算中はシミュレーションパートが待機する構造であったが、これを改良し、シミュレーションパートが機械学習パートを待たずに動作し続けるようにした。

ファイルの有無で実行状態を判別するシンプルな仕組みを導入し、以下のような形で実装した：

1. シミュレーションパート: 指定されたファイルの存在を監視し、その有無で機械学習パートの終了を判断。
2. 機械学習パート: 別のトリガーファイルの存在により、学習の開始を判断。

・協調動作の確認

この新しい実装により、Odyssey (CPU)と Aquarius (GPU)で、SiO_2の SLMC が協調して動作することを確認した。この方法は非常にシンプルながら、基本的な CPU+GPU 連携を達成するための有効な手法であることが示された。

[WaitIO による連携] 今回はファイル有無による簡単な実装で CPU-GPU 連携を行った。今後、WaitIO によるファイル書き込みなしの連携も試みる予定である。しかし、今回の連

携の実装において新たに注意しなければいけない点も見つけることができた。それは、CPU 計算と GPU 計算のロードバランスである。CPU と GPU は二つのジョブとして投げられており、連携が行われる。その際、どちらのジョブも常に計算が回っているようにすべきである。例えば、CPU がシミュレーションパートを実行している際にもしデータが集まっていない場合、GPU は待機状態になっており、何も実行せずに無駄に計算時間を消費することになってしまう。そのため、最初はシミュレーションパートを短めにするなど両者が両方とも動作するように全体を調整する必要がある。これは、シミュレーションパートで得られたデータを増やす（テイラー展開による方法 M. Cooper et al., npj Comput Mater 6, 54 (2020)などが考えられる）ことで対応できる可能性がある。

[自己学習モンテカルロ法の精緻化と同変トランスフォーマー] 自己学習モンテカルロ法および自己学習ハイブリッドモンテカルロ法でシミュレーションを行う際に重要な要素の一つが、有効模型をどのように構築するかである。有効模型が元の模型をよりよく模倣すればするほど、モンテカルロ法におけるアクセプト率が改善される。

一方、昨今の生成 AI では、Transformer や拡散モデルといった高度な生成モデルが急速に発展しており、これらの技術は物理系のモデリングにも応用が進んでいる。特に、Transformer は長距離の相関を捉える能力に優れており、格子模型やスピン系、原子分子系のような複雑な相関を持つ物理系に対して有効なモデリング手法となり得る。Transformer を用いた機械学習ポテンシャルとしてはすでに TorchMD(arXiv:2202.02541)が提案されているが、ネットワーク構造がやや複雑であり、パラメータ数が数十万～数百万である。従来の Behler-Parrinelo 型（1000 パラメータ程度）と比べて著しく多く、MD として実行する際の機械学習ポテンシャル計算速度が遅くなってしまう。そのため、パラメータの多い優秀なモデルは訓練と推論が遅くなるという欠点を持つため、高速な MD 実行という観点ではよりよいモデルを探索する必要があった。

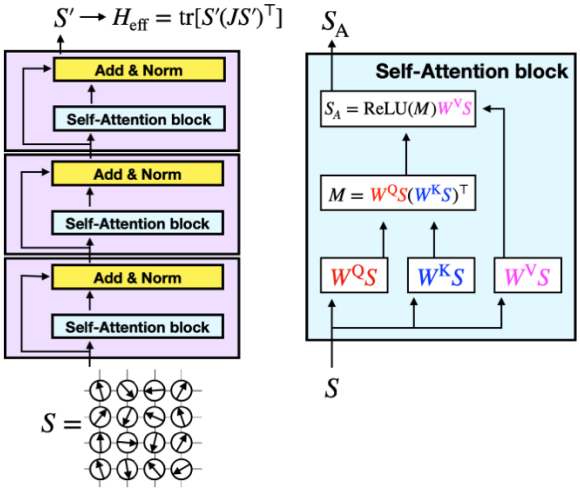
これらを踏まえ、同変トランスフォーマーを開発した(Y. Nagai and A. Tomiya, J. Phys. Soc. Jpn. 93, 114007 (2024))。本研究では、二次元格子上の古典スピン・フェルミオンモデル（二重交換模型）を対象として、以下の手法を導入した。

・対称性を持つ Transformer の設計

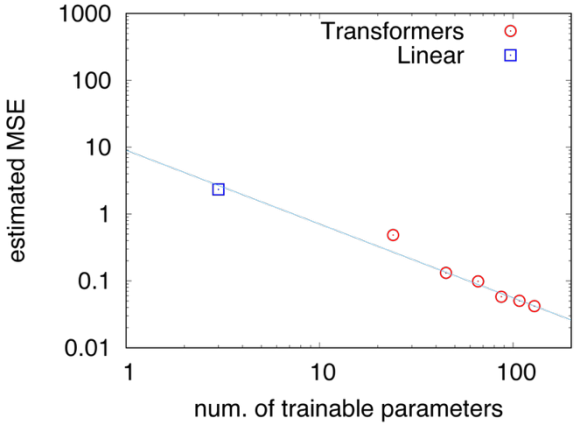
スピン回転対称性や並進対称性を保つため、自己注意機構（Self-Attention）を持つ Transformer を拡張し、モデル内で入力データの対称性を保持するように設計した。

・有効スピン場の構築

対称性を持つモデルを構築するために、Gram 行列を用いて回転不変性を持つ Attention Matrix を導入し、この Matrix をスピンを表す 3 次元ベクトルの集合に作用させるモデルを設計した。その結果、全体として同変性を持つモデルを設計することができた。このモデルは入力をスピン集団、出力もスピン集団とするモデルである（下図参照）。最終的には得られた”有効スピン”を入力として物理的な模型（古典ハイゼンベルグ模型）のエネルギーを計算し、有効模型が提案するエネルギーとした。



図：同変トランスフォーマーの模式図



図：精度のパラメータ数依存性

・ＳＬＭＣとの結合

構築した有効模型を SLMC に組み込み、この模型がモンテカルロ法のアクセプト率を向上させること示した。

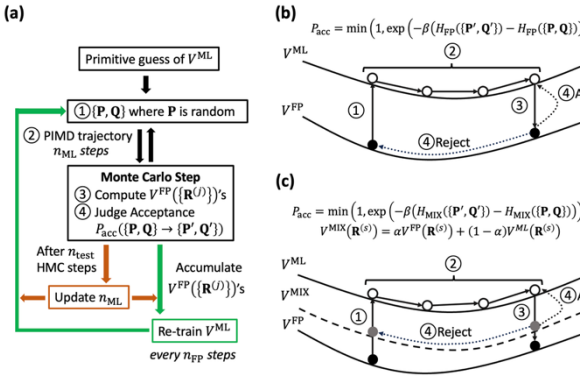
上記の有効模型は Attention 層の枚数を変更することができる。そして、層１枚あたりのパラメータは数十と極めて少ない。パラメータが非常に少ないにも関わらず、層の数を増やしていくと精度が逐次的に改善されることがわかった（右上図参照）。このような、パラメータ数を増やせば増やすほど精度が改善される現象は生成 AI（大規模言語モデル）におけるスケーリング則として知られている。本研究は、パラメータ数を著しく減らしたにも関わらず大規模言語モデルと同様のスケーリング則が現れることを示しており、正しく Transformer としての性能を持っていることを示すことができている。

以上の結果から、対称性を正しく考慮して注意深く構築することにより、大規模言語モデルと類似の傾向を持つ有効模型を物理系においても作ることができた。そして、スピン系の模型ではあるが、自己学習モンテカルロ法とトランスフォーマーを組み合わせることに成功することができたことを意味する。この研究によって、機械学習ポテンシャル構築においても、パラメータ数の少ないトランスフォーマーモデルを設計できる可能性が生じた。今後は MD の実行性能の高い（推論性能の良い）トランスフォーマーモデルを構築する予定である。

[自己学習ハイブリッドモンテカルロ法の拡張：SLHMC-MIX の開発]　水や重水（D_2O）などの軽元素を含む物質において、核量子効果（Nuclear Quantum Effects: NQEs）は構造や熱力学的性質に大きな影響を及ぼす。これらの効果を高精度に再現するためには、経路積分分子動力学（Path Integral Molecular Dynamics: PIMD）などの量子統計的手法が用いられるが、第一原理計算（First-Principles: FP）に基づく PIMD は計算コストが非常に高く、大規模系や長時間スケールのシミュレーションには適していない。　本研究では、第一原理計算と機械学習ポテンシャルを組み合わせた自己学習経路積分ハイブリッドモンテカルロ法（Self-Learning

Path Integral Hybrid Monte Carlo with Mixed Potentials: SL-PIHMC-MIX)を開発し、水の核量子効果を高効率かつ高精度に再現する手法を提案した（B. Thomsen et al. J. Chem. Phys. 161, 204109 (2024) ）。

SL-PIHMC-MIX は、第一原理ポテンシャル(FP)と機械学習ポテンシャル(MLP)を線形結合した混合ポテンシャルを用いて、経路積分ハイブリッドモンテカルロ(PIHMC)シミュレーションを行う手法である(下図参照)。この手法では、初期段階で少数の FP 計算を行い、その結果を用いて MLP を訓練する。訓練された MLP は、PIHMC シミュレーション中に再評価され、必要に応じて再訓練されることで、シミュレーションの精度と効率を両立する。そして、SL-PIHMC-MIX では、リウェイト法を用いて FP ポテンシャルに基づく統計量を再現することが可能であり、MLP の導入によるバイアスを補正することができる。



図：S L—PIHMC-MIX の模式図

SLHMC-MIX では以下の成果を得ることができた。

・計算効率の大幅な向上: 従来の FP-PIMD では、32 ビーズ（経路積分 MD で必要なコピーされた原子構造数）の構造評価に約 10 万回のエネルギー評価が必要であったが、SL-PIHMC-MIX では約 5,000 回の評価で同等の精度を達成した。

・高精度な構造再現: SL-PIHMC-MIX によって得られた水の構造は、FP-PIMD と一致し、核量子効果を高精度に再現できることが確認された。

・重水（D_2O）への適用: 本手法は、重水に対しても適用可能であり、同様の精度と効率を示した。

・異なる DFT 機能の比較: RPBE-D3、SCAN、rev-vdW-DF2、optB88-vdW など、複数の密度汎関数理論(DFT)機能を用いたシミュレーションを行い、SL-PIHMC-MIX の汎用性と精度を検証した。

本研究で開発した SL-PIHMC-MIX は、第一原理計算の精度を維持しつつ、計算コストを大幅に削減することが可能であり、水や重水などの軽元素を含む物質の核量子効果を高効率にシミュレーションする新たな手法として有望である。今後は、他の水素結合系や生体分子、材料科学への応用が期待される。

6. 進捗状況の自己評価と今後の展望

［進捗状況の自己評価］

CPU と GPU を連携させた自己学習モンテカルロ法の開発は PIMD の機能拡張という形で果たされている。その際、機械学習ポテンシャルは aenet-PyTorch に対応している。一方、当初の予定である、機械学習ポテンシャル作成ソフトウェア NequIP および Allegro との連携に関しては、aenet-PyTorch における CPU と GPU のロードバランスの調整という課題に直面したため十分に行うことができなかった。そして、SLHMC-MIX という新しい自己学習ハイブリッドモンテカルロ手法の開発と、新しい同変トランスフォーマーによるモデルの構築は、自己学習モンテカルロ法の精緻化という意味では本プロジェクトの一部とみなすことができる。

これらを踏まえて、達成度は 80 パーセントとする。

［今後の展望］

Aenet-PyTorch への連携を実装するにあたって、さまざまな細かな技術的課題が存在していた。これらを解決するために Aenet-PyTorch を改修するという方法をとってきた。

そして、aenet の中身を改修する過程において、書き方が少し古く、速度の面でも改善点があることが判明した。そこで、これらの点を解消した自前のコードを開発することにする。これらとこれまでの研究計画を踏まえ、次年度以降は以下のような計画を予定している。

[最新の知見を利用可能な機械学習ポテンシャルソフトウェアの新規開発] 次年度は、Fortran2008 と Julia 言語で書かれたオープンソースソフトウェア BPnet(仮称)を開発する。これは、2 つの理由がある。CPU-GPU 連携を実装する際に、CPU 及び GPU で実行可能な環境を整備する必要があるが、最新の CPU 及び GPU に対応しているかどうかはオープンソースソフトウェアの開発状況に依存してしまっており、ソースコード改変が困難となることがある。また、aenet や n2p2 などのソフトウェアは開発がほぼ停止しており、最新の機械学習の知見が入らない。そのため、進展著しい機械学習業界での手法を入れることができない。そこで、我々が最近開発している Kolmogorov-Arnold networks (Y. Nagai and M. Okumura, arXiv:2407.17774) も実装した新しいソフトウェアを開発する。そして、Julia 言語を通して GPU を使うことで、最新のアーキテクチャに容易に対応可能なソフトウェアを開発する。

[PIMD-NequIP 実装] 本年度は Wisteria 上で CPU-GPU 連携をテストし、[PIMD-aenet-PyTorch 連携実装]は完了している。一方、別のオープンソースソフトウェア NequIP の連携開発は途中となってしまっていた。そこで、2025 年に稼働を開始した Miyabi システム上での動作も可能となるように実装を行うこととする。

[PIMD-Allegro 連携実装と評価]大規模 GPU 学習が可能な Allegro を実装し、進展次第では、「大規模 HPC チャレンジ」に応募し，更に大規模な計算機資源を利用した評価を実施する。

Allegro はマルチ GPU において並列化効率が高いと言われている。これまでのパートでは基本的には 1GPU で調査する予定であるが、Allegro は可能であればマルチ GPU での訓練が可能となるように実装する。

改良した PIMD を利用者向けにオープンソースで提供する。PIMD は現在オープンソースで公開されているが、PIMD の開発者とコンタクトを取りつつ、PIMD の本体にマージするか別の派生パッケージとして公開するかを検討する。