

ビッグデータとHPCを活用した アプリケーション事例のご紹介

2023年6月21日

白幡 晃一

富士通株式会社 富士通研究所

コンピューティング研究所

イノベティブコンピューティング コアプロジェクト



- ビッグデータとHPCを扱うアプリケーションの事例として、遺伝子分析と材料探索について紹介
 - 遺伝子分析では、データから新たな発見の手掛かりを提示する因果発見技術を「富岳」上に実装し、1,000兆通りの可能性から未知の因果を発見できる技術を開発
 - 材料探索では、アンモニアをクリーンに合成するための触媒探索において、量子化学シミュレーション高速化技術を開発し、因果発見技術と組み合わせることで探索期間を半減

スーパーコンピュータ「富岳」と「発見するAI」で、がんの薬剤耐性に関わる未知の因果メカニズムを高速に発見する新技術を開発

1,000兆通りの可能性から患者の遺伝子の特徴を一日以内で絞り込み、効率的な薬剤開発に寄与

○薬剤耐性とは？

- 薬剤の投与を続けているうちに、その薬剤の効果が弱まってしまう現象

○薬剤の臨床試験の課題

- 薬剤効果が期待される患者のおおまかな同定が重要だが、患者の特性（個人や臓器における遺伝子やその発現量など）により薬剤効果は異なる
- 患者の特性として複数の遺伝子発現量の組合せを考えると、特性の総数は1000兆通りを超える

○今回の取り組みで目指すブレークスルー

1. 患者の特性を網羅的に調べて、薬剤効果との関係を示唆する因果メカニズムを抽出
2. 抽出した特性とメカニズムを検証し、薬剤効果が期待できる患者を同定
3. 同定した患者だけを対象として臨床試験を行うことで、経費を大きく削減し、成功確率を高める → **臨床試験の効率化！**

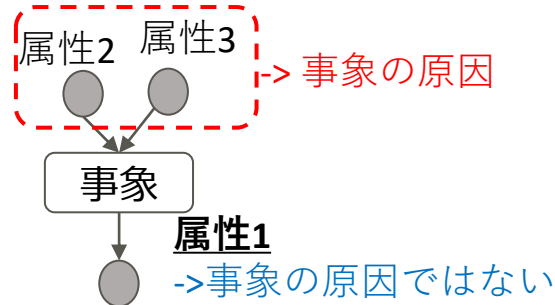
- 現在のほとんどのAI技術は属性と事象の**相関関係**を学習
- 高い相関を持つ場合に必ずしも**因果関係**があるとは限らない



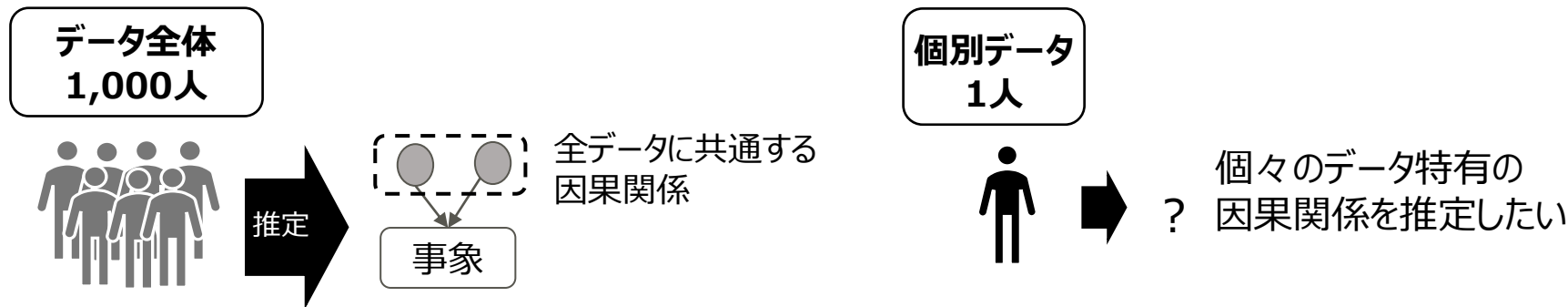
相関関係

属性	相関関係
属性1	高
属性2	高
属性3	中
属性4	-

因果関係



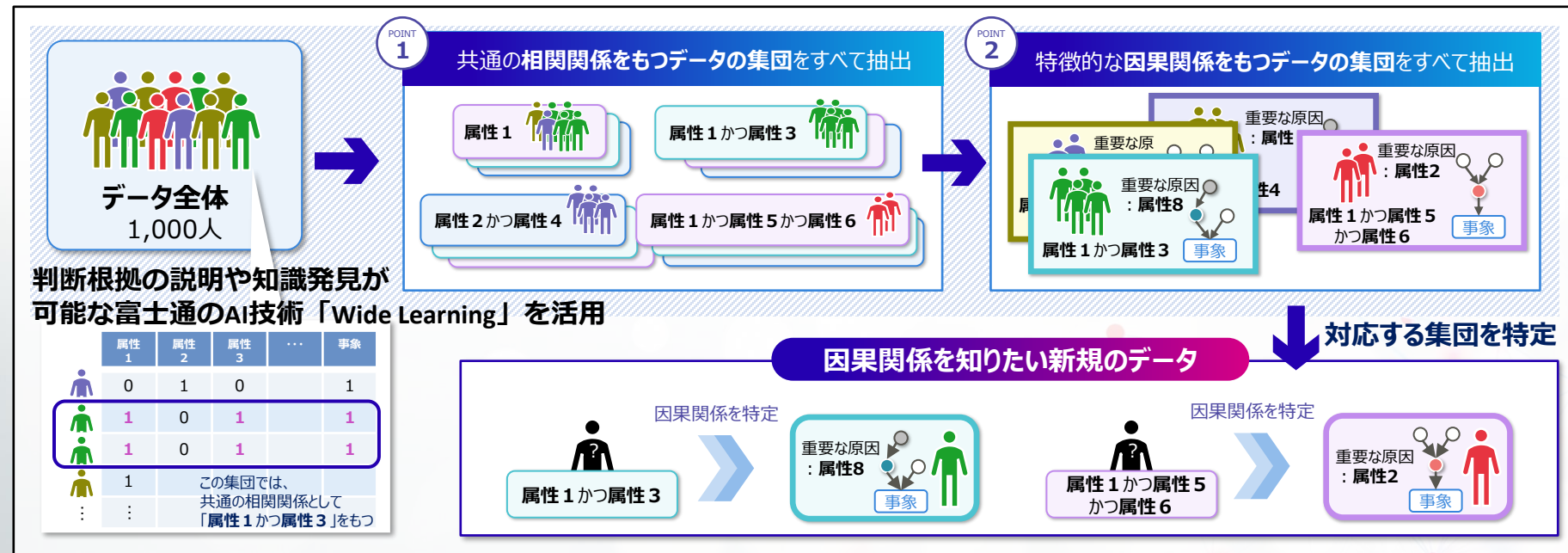
- 現実には因果関係は**状況によって異なる因果メカニズム**により生じることが多い
- 様々な因果メカニズムに影響を受ける個々のデータの因果関係をどのように推定するか？



- 薬剤開発以外の現場においても、様々な状況によって異なる因果メカニズムを発見し、業務改善につなげることが可能

分野	発見したい因果メカニズム	期待される業務改善
マーケティング	顧客の特性ごとのサービス解約に至ったルートと根本原因	サービス離れしようとしている顧客を特定し、解約の根本原因を明らかにすることで解約防止にむけた効果的な施策を策定
システム運用	複雑なシステムにおける障害発生箇所とその原因	障害箇所と原因をリアルタイムで提示し、早期復旧を支援
生産現場	稼働環境に応じて不良品発生を引き起こす制御パラメータ	現在の稼働環境に適したパラメータの自動設定により、不良品の発生率を減らす

データ全体に共通する因果関係だけではうまく因果メカニズムを推定できないことがある
→ 富士通の「発見するAI」では個々のデータに特徴的な因果関係を網羅的に抽出



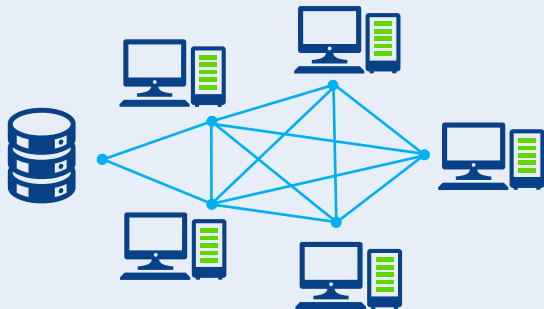
※2020年12月17日プレスリリース「ヒトやモノなどのデータの一つひとつが持つ特徴的な因果関係を発見する技術を開発」

<https://pr.fujitsu.com/jp/news/2020/12/17.html>

従来では数百変数に対する因果探索が限界だった
→ **数万変数に対する因果探索が実行可能に。**

大規模な因果探索のための処理の高速化

因果探索アルゴリズムの並列化



スーパーコンピュータ上での計算処理



提供：理化学研究所



提供：理化学研究所

スーパーコンピュータ「富岳」

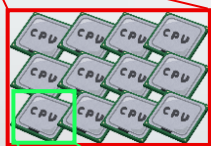
少数のコンピュータからスーパーコンピュータまで、計算能力を引き出す
機械学習環境を大規模因果探索に適用

計算回路を使い切る技術



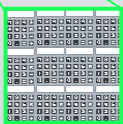
コンピュータ間並列

通信を抑え、効率的に実行できる並列性を抽出



CPUコア並列(～50倍)

プロセッサ内の処理回路をすべて活用

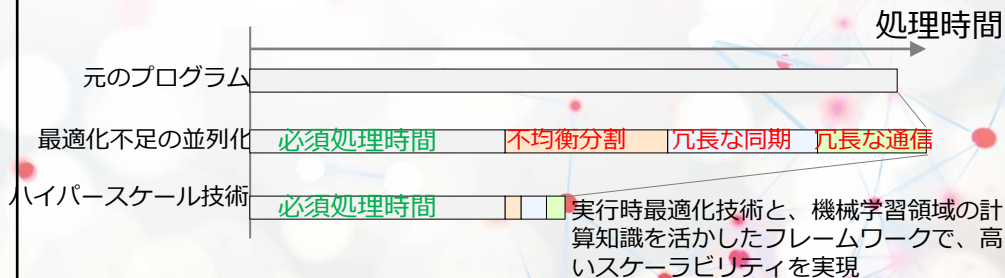


演算器並列(～10倍)

プロセッサに合わせて、実行時に最適なプログラムを生成

計算を止めない技術

並列化の際に必要な通信が、計算を止めないように最適化



事例

ゲノム医療への応用

薬剤耐性の原因究明

がんへの薬剤応答データから、
薬剤耐性メカニズムに関わる遺伝子関係を抽出。

- ヒトの全遺伝子（20000種類）に対して
重要な因果関係を網羅的に抽出

従来技術では4000年かかっていた → 1日で完了

- 肺がんの治療薬として知られる「ゲフィチニブ」の
耐性獲得メカニズムに関して、
専門家が認める新仮説の抽出に成功

共同研究：東京医科歯科大）宮野先生



スーパーコンピュータ「富岳」成果創出加速プログラム
「大規模データ解析と人工知能技術によるがんの起源と多様性
の解明」 研究代表

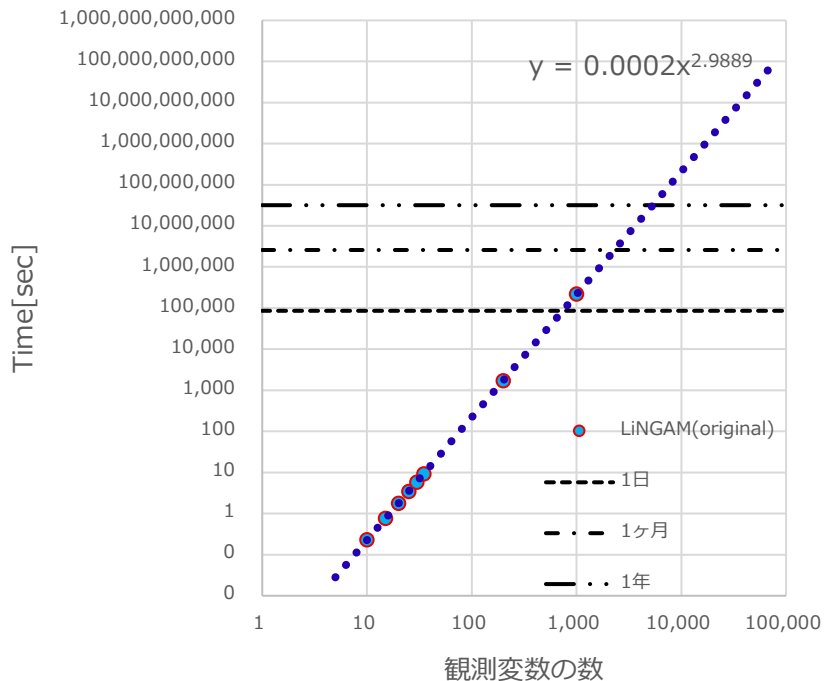
因果関係の仮説と新発見の可能性



既存の文献通り

PRRX1の上皮間葉転換(EMT)制御についてはいくつかの研究がある。EMTは肺がんにおけるゲフィチニブ耐性のメカニズムの一つであることが多くの研究で示唆されている。

栗原ら, IPSJ-HPC-180 (SWoPP2021)



○評価環境

- CPU:Xeon Gold 6148
- シングルスレッド動作

○評価結果

- 200変数で約1,700秒、1,000変数で約60時間
- 計算量は約 $O(x^3)$
- ターゲット1万変数→5.73年かかる

As-isのLiNGAM実装では、数百変数以下が実用限界(1日以内)
目標:1万変数で1日以下を実現するためにはLiNGAMの高速化

- 高速化対象の関数は、配列の要素に対して算術演算を繰り返す処理を含む関数

LiNGAMのPythonコード

栗原ら, IPSJ-HPC-180 (SWoPP2021)

```
def _entropy(self,u)
# uはfp32の1次元配列
return (1 + np.log(2 * np.pi)) / 2 - ¥
    k1 * (np.mean(np.log(np.cosh(u))) - gamma)**2 - ¥
    k2 * (np.mean(u * np.exp((-u**2) / 2))**2
```

疑似Cコードによる動作イメージ

```
float entropy(float* array, int N) { // Nは配列の要素数
    float retVal;
    float temp1 = 0.0f;
    float temp2 = 0.0f;
    for (int i = 0; i < N; i++) {
        temp1 += log(cosh(array[i]));
        temp2 += array[i] * exp(-1 * pow(array[i], (double)2) / 2.0);
    }

    retVal = (1.0 + log(2.0 * M_PI)) / 2.0 - k1 * pow(((temp1 / size) - gamma), 2.0) - k2 * pow((temp2 / size),
2.0);

    return retVal;
}
```

提案：JITで複数算術関数の処理をフュージョン FUJITSU

栗原ら, IPSJ-HPC-180 (SWoPP2021)

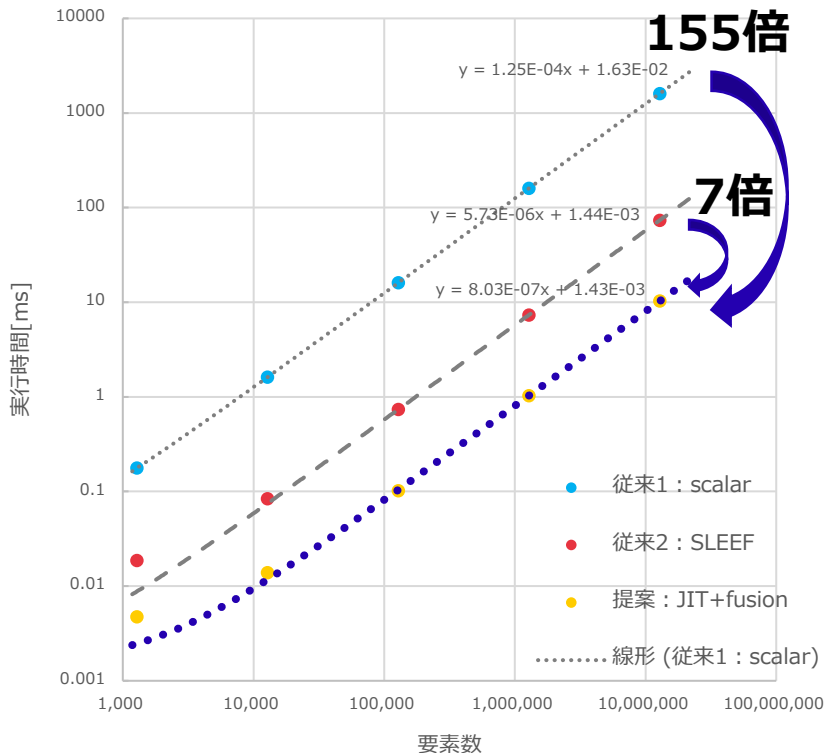
- 配列の各要素に対して、複数の算術関数を繰り返す処理を**1回の関数call**で実現する関数を、Just-In-Time(JIT)に生成するライブラリを開発
 - Just-In-Timeで生成することにより、算術関数(sin, cos, tan, log, exp, asin, acos, atn, sinh, cosh, tanhなど)の任意の組み合わせに対応可能
 - M種類の算術関数に対し、k個の組合せは m^k 通りとなり、事前に実装を用意するのは不可
 - JITコード生成機能として、富士通が開発したXbyak_aarch64*を使用
 - Xbyak_aarch64はArmv8-A + SVEアーキテクチャ向けの機械語列をメモリ上に書き込むことができるOSS

*Xbyak_aarch64

https://github.com/fujitsu/xbyak_aarch64

<https://connect.linaro.org/resources/lvc21/lvc21-203/>

栗原ら, IPSJ-HPC-180 (SWoPP2021)



○ 評価環境

○ Fujitsu PRIME HPC FX700

- CPU: A64FX
- 1コア1スレッド動作

○ 使用ライブラリ

- 従来技術1: libm
- 従来技術2: libsleef
- 提案技術: exp(exp)の処理に限定してJITコードを生成できるように開発したサンプルプログラム

評価に使用した処理の疑似コード表現

```
float a[N];
float b[N];

/* 配列aには何かしらの入力データが入っているものとする */
for(int i=0; i<N; i++) {
    b[i] = exp(exp(a[i]));
}
```

○ 評価結果

○ 従来技術1に比べて約**155倍**、従来技術2に比べて約**7倍**の性能改善

- 提案技術のコード生成のオーバーヘッドは約0.15msecで非常に小さかった

材料探索効率化技術

HPCとAIを活用した量子化学シミュレーション高速化技術を開発し、クリーンなアンモニア合成に向けた触媒候補探索期間を大幅に削減

HPCとAIとデジタルアニーラで材料探索プロセスを革新する

本ご紹介

高速シミュレーション (HPC/AI)

高速な原子レベル
シミュレーション

- 原子分子レベルの相互作用を
求める量子化学計算を高精度化
- 現状スパコンで数10年を要する
計算を数時間で実現

因果発見技術

化学反応における新たな
知見を得る

- 原子レベルのシミュレーションと実験
データから材料候補と化学反応特
性間の因果関係を網羅的に分析
- 専門家の勘だけでは見いだせな
かった新しい反応経路を発見

デジタルアニーラ

高速な化学空間探索

- 材料特性予測モデルをデジタル
アニーラの最適化問題に変換
- 従来の方法よりも迅速に広い
化学空間からより良い材料候
補を発見

Computing Workload Broker (HPC/量子)

自動的なハードウェア資源割り当て最適化

HPCとAI技術を活用し、カーボンフリー素材開発を促進



Atmonia

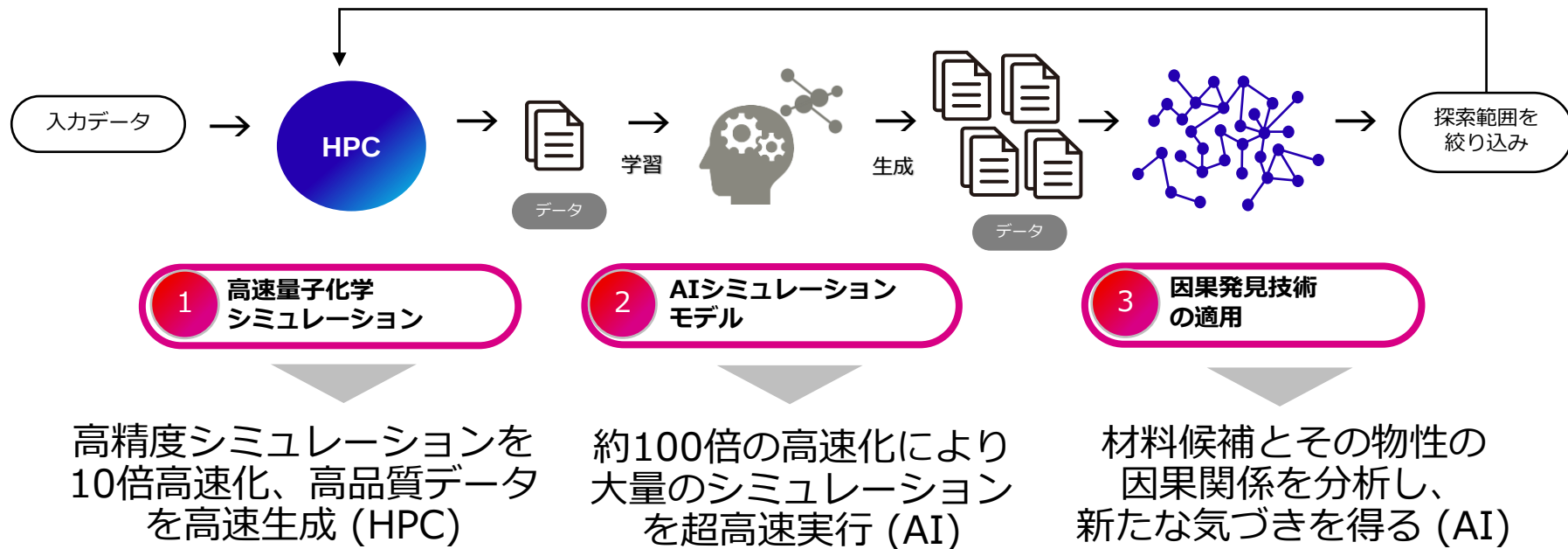
Atmonia社との共同研究

(アンモニア合成触媒を開発するアイスランドのベンチャー企業)

カーボンフリー物質であるアンモニアの触媒探索をHPCとAIで加速

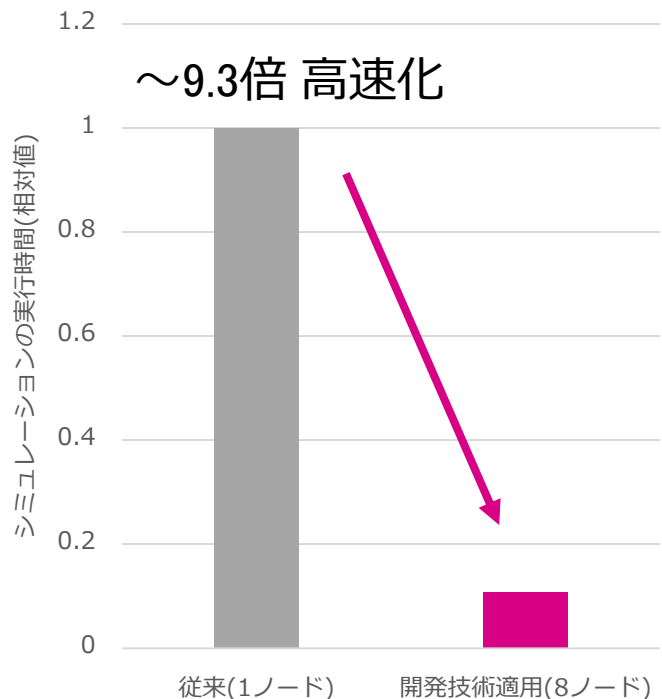
- アンモニアはCO₂の排出が無く、水素に比べて輸送が容易なため次世代エネルギーとして注目
- しかし現在主流のアンモニア合成手法では化石燃料から原料の水素を作るためCO₂が大量に排出されてしまう課題がある
- 富士通のHPC/AI技術を活用しAtmonia社と 水と空気（窒素）と電気のみでアンモニアを取り出せる革新的な新触媒を研究中

今回開発・適用した技術： HPC & AI の融合による高速材料探索技術



HPC/AI を組み合わせた探索効率化技術により
新材料候補の探索期間の大幅な短縮を可能に

①高速量子化学シミュレーション



- 量子化学計算の一つの手法である密度汎関数法 (DFT) の高速化技術を開発
- 計算量の削減と並列計算の効率化 (HPC)
 - DFT 計算のアルゴリズムを改良し、計算量を削減
 - 「富岳」を活用した並列計算で計算と通信を効率化

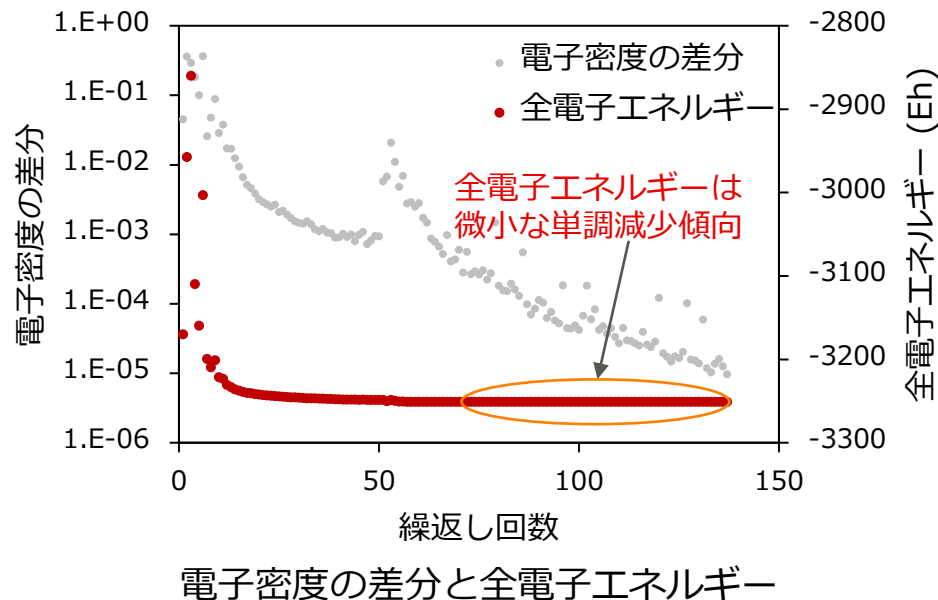


約10倍の高速化を実現

大量のシミュレーションによる高速データ生成、探索の高効率化

大田ら, JCPE Journal 21 (2), 52-54, 2022

○提案：全電子エネルギーの安定状態により収束条件を緩和



SCF計算の
収束条件

- 全電子エネルギーの変化率*1
 - 電子密度の差分の変化率*1
- を追加

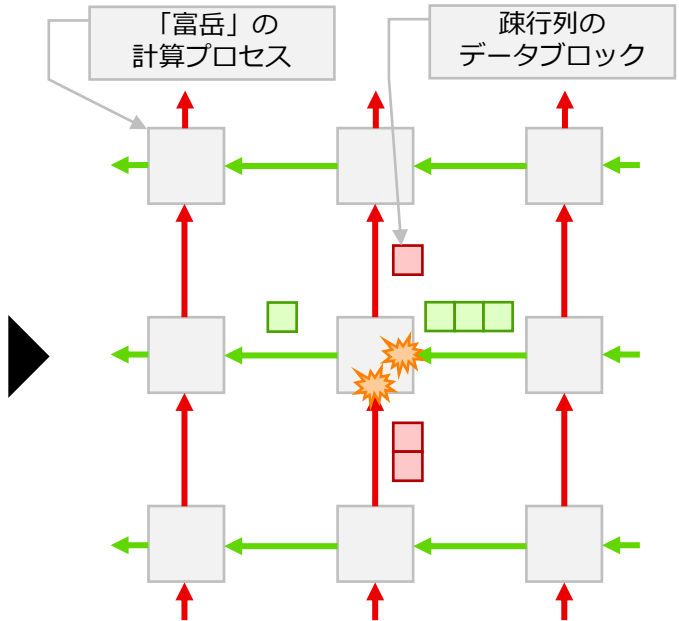
1.2~2.5倍高速化

*1 変化率_n = |(現在値_n - 過去m回の平均値)/過去m回の平均値|

Oyama et al., IEEE/ACM CCGrid 2023

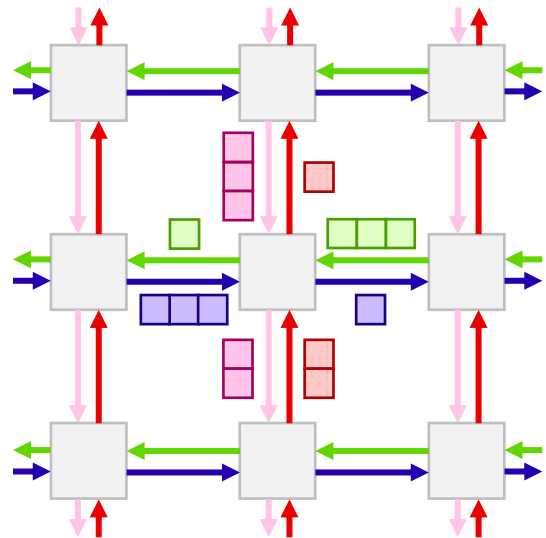
従来技術

材料開発アプリにおける
疎行列積を「富岳」で計算



通信される行列データ   の量が不均衡
→ 通信待ち時間  が発生

新規開発した技術



二次元トラス上で双方向に通信することで
通信をオーバーラップし、待ち時間を削減

Proposal: Optimized electron integrals

Oyama et al., IEEE/ACM CCGrid 2023

We promote compiler's SIMD conversion by manual refactoring

```
1  IF (jatom >= latom) THEN
2    i = 1
3    j = offset_bd
4    DO md = 1, mdm
5      DO mb = 1, mbm
6        ks(j) = js(j)+kbd(i)
7        i = i+1
8        j = j+1
9      END DO
10   END DO
11 ELSE
12   i = 1
13   offset_bd = ...
14   DO md = 1, mdm
15     j = offset_bd+md-1
16     DO mb = 1, mbm
17       ks(j) = ks(j)+kbd(i)
18       i = i+1
19       j = j+mdm
20     END DO
21   END DO
22 END IF
```

Existing code



Refactoring

```
1  IF (jatom >= latom) THEN
2    i = (offset_bd+mbm*mdm-1)
3    ks(offset_bd:i) = ks(offset_bd:i)+ \
4      kbd(1:(mbm*mdm))
5  ELSE
6    offset_bd = ...
7    DO md = 1, mdm
8      j = offset_bd+md-1
9      ks(j:(j+mdm*(mbm-1)):mdm)= \
10        ks(j:(j+mdm*(mbm-1)):mdm)+ \
11        kbd(1+(md-1)*mbm:md*mbm)
12    END DO
13 END IF
```

Optimized code

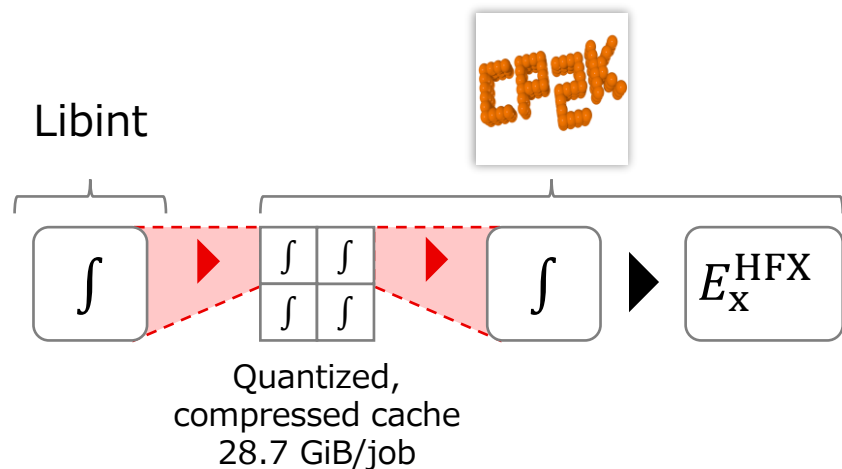
SIMD enabled

SIMD disabled

Proposal: Optimized electron integrals

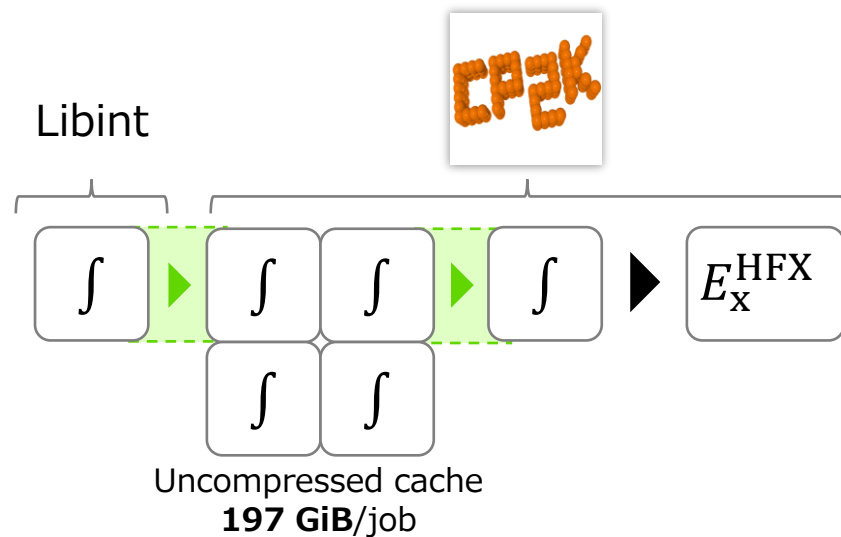
Oyama et al., IEEE/ACM CCGrid 2023

We keep CP2K's integral cache uncompressed for speedup



Existing method

✗ Compression/decompression overhead



Optimized method

- ✓ No overhead but in-memory copy
- Multi-node are required

Evaluation: Optimization methods

The two optimization methods improve Fugaku performance

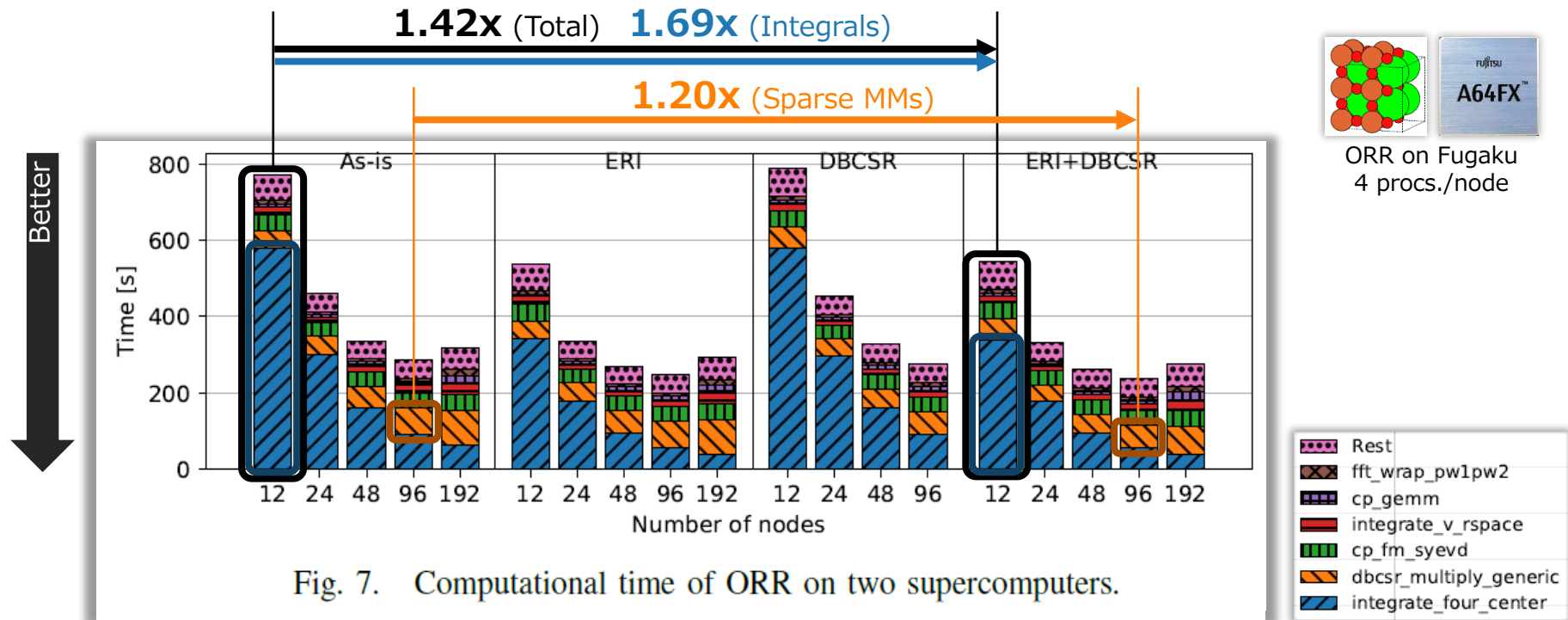
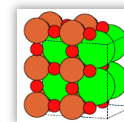
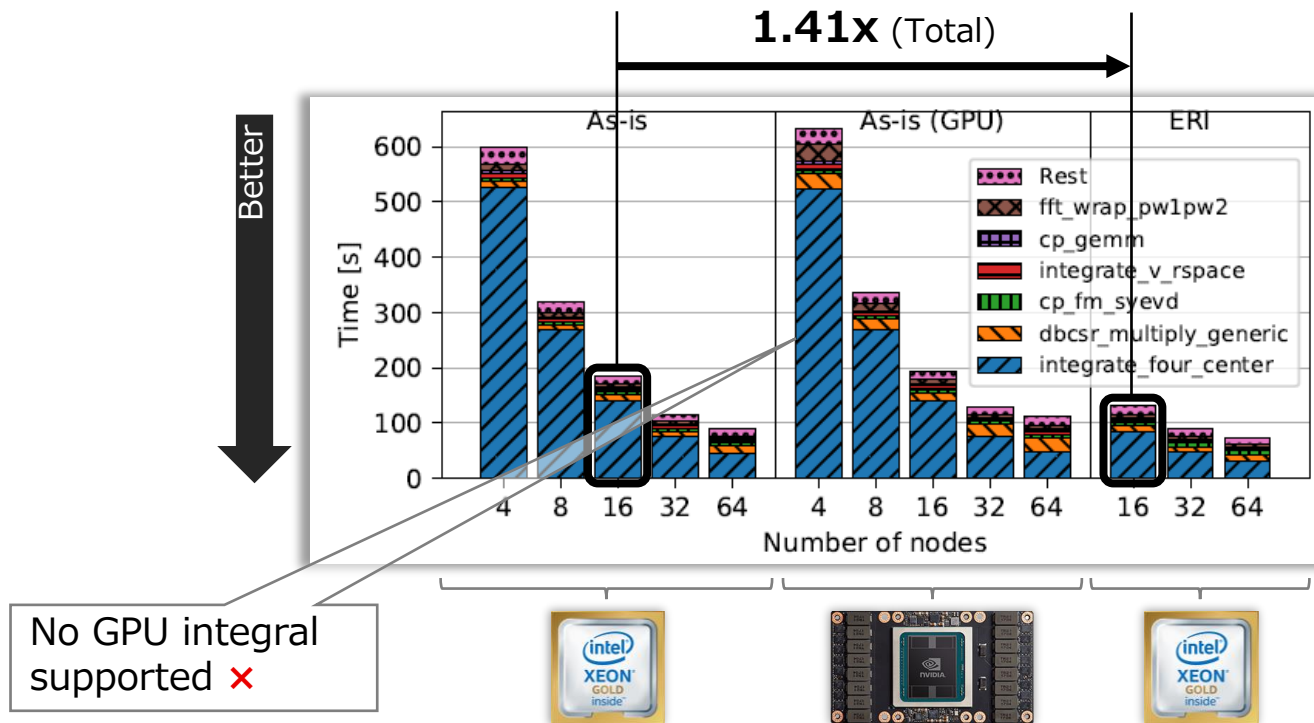


Fig. 7. Computational time of ORR on two supercomputers.

Evaluation: Optimization methods

The integral optimization also improves ABCI performance

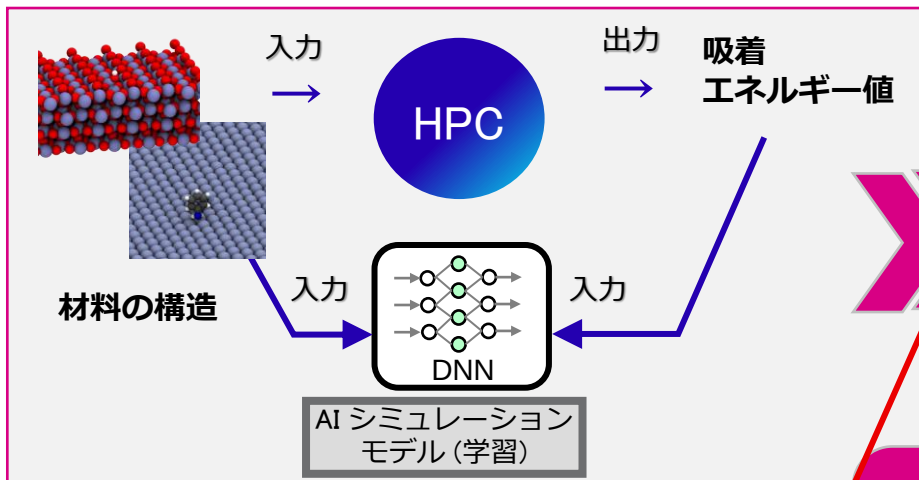


ORR on ABCI
2 procs./node

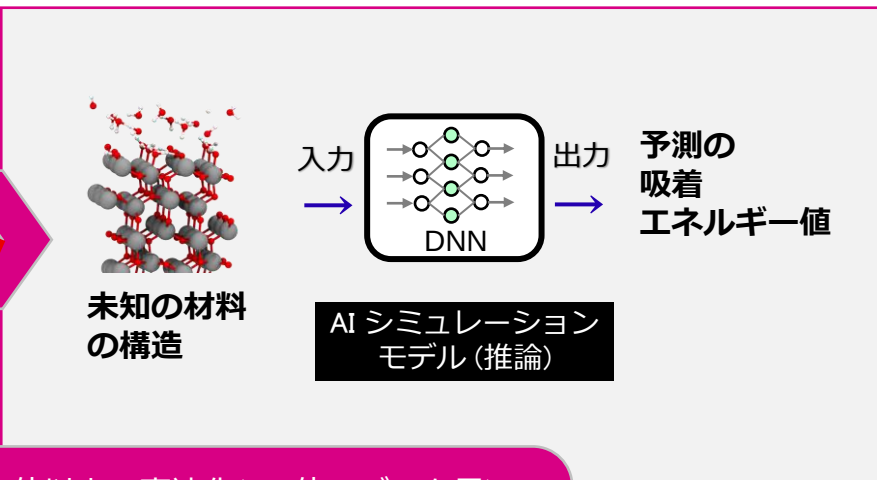
② AI シミュレーションモデル

- HPCで大量に実行したDFTシミュレーションデータをAIシミュレーションモデルに学習させ、未知の材料構造の吸着エネルギー値を瞬速で推論
→ **約100倍以上のデータ入手が可能になる**
- 大量データ生成により、探索範囲の拡大、因果発見技術の精度向上が図れる

量子化学(DFT)シミュレーション



DFTシミュレーションの AI シミュレーションモデルへの置き換え



100倍以上の高速化(100倍のデータ量)

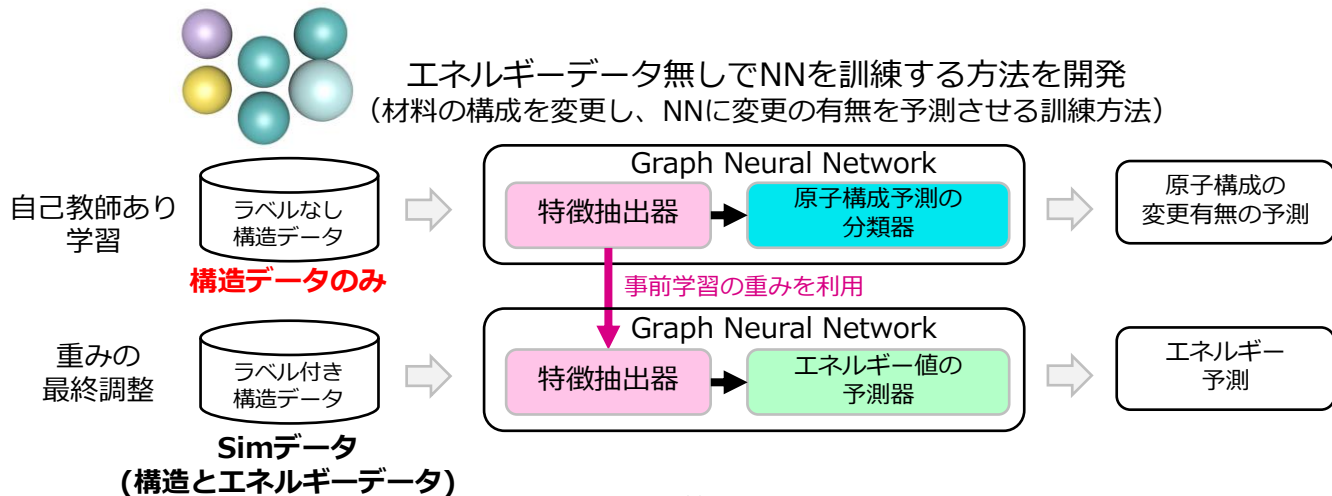
Sakai et al., IEEE IJCNN 2023 Workshop

○ 背景・従来の課題

- 材料探索には、材料の「エネルギー」が性能指標に用いられる
- エネルギー予測精度の高いNNの実現に必要な大量のエネルギーデータを用意するには数十～数千時間かかる

○ 開発技術

- エネルギーデータを増やさなくてもNNを高精度化する技術(触媒探索向け自己教師あり学習)を開発
 - データを増やさなくてもよい → データを用意する時間・費用が抑えられる
 - NNが高精度化 → より効率的な材料探索が可能となる



Experimental conditions: Data & GNN model

Dataset

Open Catalyst 2022 (OC22) [1]

- Train data: 45890, validation data: 2624

Poisoned Catalyst [2]

- Train data: 3811, validation data: 900
- 5-fold cross validation

Expanded Poisoned Catalyst [2]

- Train data: 13453, validation data: 1000
- 5-fold cross validation

GNN model

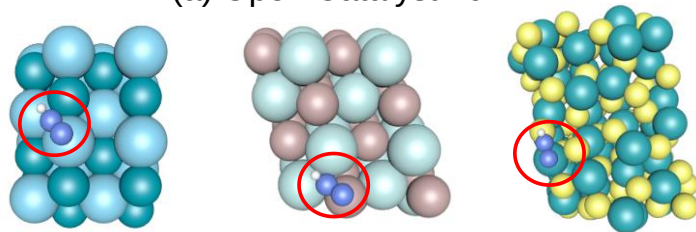
CGCNN

PaiNN

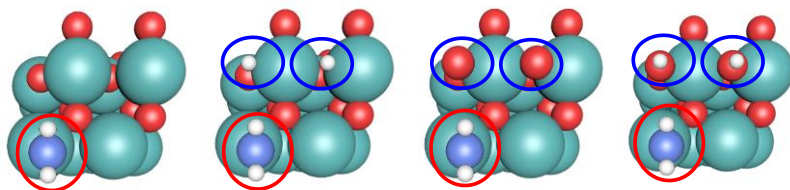
Sakai et al., IEEE IJCNN 2023 Workshop

Red circle : Adsorbate
Blue circle : Poisoning atom

(a) Open Catalyst 2022



(b) Poisoned Catalyst



OC22 has clean surface.

Poisoned Catalyst has poisoned surface.

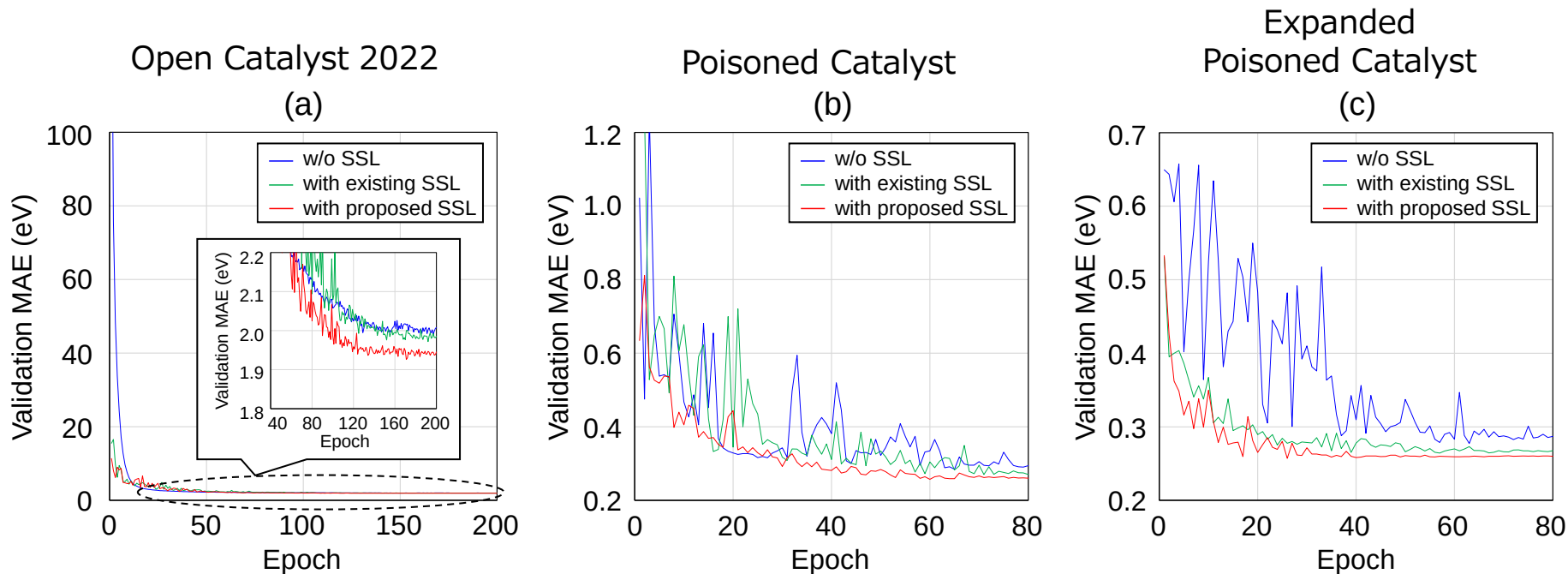
[1] Open Catalyst Project, <https://github.com/Open-Catalyst-Project/ocp>

[2] A. B. Hoskuldsson, et al., "Computational screening of rutile oxides for electrochemical ammonia formation", ACS Sustainable Chemistry & Engineering, 2017.

Results: Learning curve during fine-tuning

Sakai et al., IEEE IJCNN 2023 Workshop

○ Faster convergence when using our proposed SSL



③ 因果発見技術

従来のAI

膨大なデータの中の相関関係を利用して予測

因果発見技術

データから因果関係の法則を発見

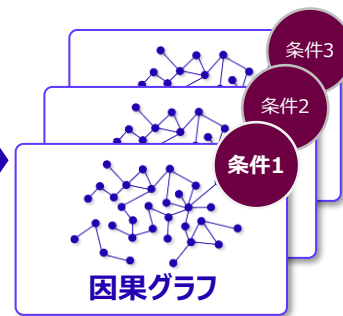
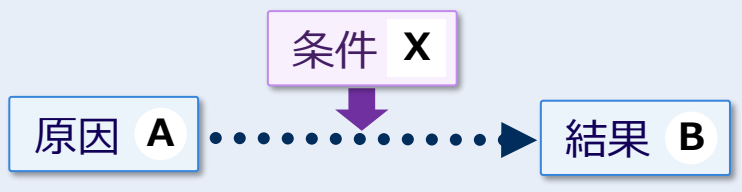
データ



入力

因果関係の法則

条件Xのとき因果関係A→Bが生じる



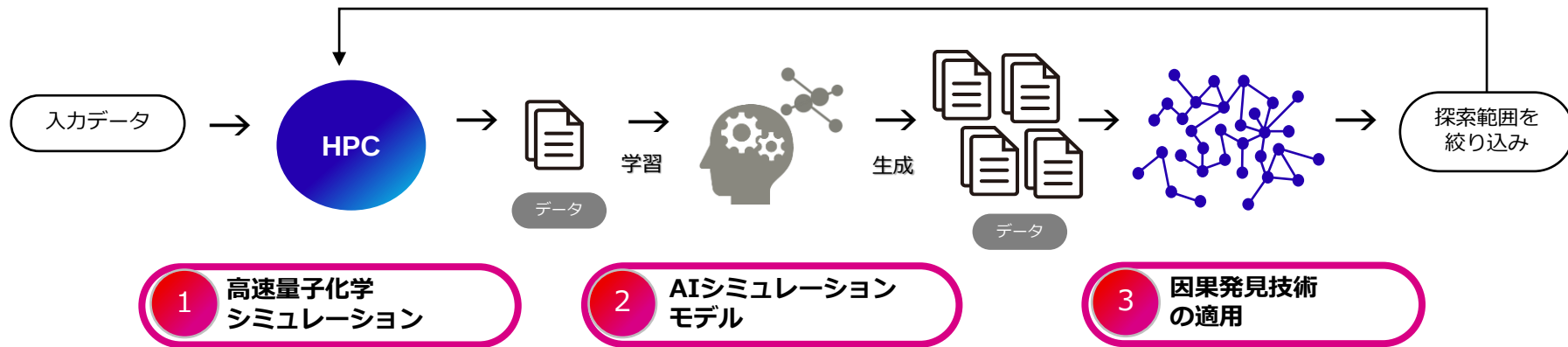
アクション
(改善・証明)



<https://pr.fujitsu.com/jp/news/2020/12/17.html>
<https://pr.fujitsu.com/jp/news/2022/03/7.html>

○技術①＋②で生成したシミュレーションデータに対して、③の因果発見技術を適用することにより、触媒候補の探索期間を半分以下に削減することに成功

- 因果発見技術への入力データ項目：触媒原子の種類、位置関係、反応エネルギー量など
- 因果発見で得た知見：「触媒候補の合金のベースとなる金属は族番号が小さいものが適している」など



- 遺伝子分析では、データから新たな発見の手掛かりを提示する因果発見技術を「富岳」上に実装し、1,000兆通りの可能性から未知の因果を発見できる技術を開発
 - 富士通のWide Learningを活用して集団ごとの因果を抽出
 - 相関・因果関係から大量の集団が得られるため、「富岳」で大量の計算を実行
- 材料探索では、アンモニアをクリーンに合成するための触媒探索において、量子化学シミュレーション高速化技術を開発し、因果発見技術と組み合わせることで探索期間を半減
 - DFT 計算のアルゴリズム改良、並列計算の効率化、AIシミュレーションモデル
 - 約10000パターンの触媒候補に対してシミュレーションを実行し、化学データベースと組み合わせで因果分析

Thank you

