

PIMDを用いた機械学習分子シミュレーション

精度保証された機械学習シミュレーション手法：自己学習ハイブリッドモンテカルロ法の紹介
機械学習ポテンシャルの精度に依らずに第一原理計算精度を保証する計算手法

原子力機構システム計算科学センターシミュレーション技術開発室

理研 革新知能統合研究センター (AIP)

永井佑紀

アウトライン

- イントロダクション
- 原子・分子シミュレーションと機械学習
- 機械学習分子シミュレーションの問題点と解決策：自己学習ハイブリッドモンテカルロ法
- PIMDによる原子分子シミュレーション： SiO_2 と $\text{YNi}_2\text{B}_2\text{C}$ を例にして
- まとめ

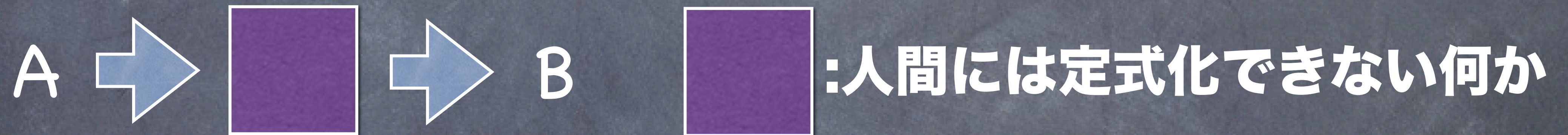
イントロダクション

今日の講演

機械学習を何に使うか？

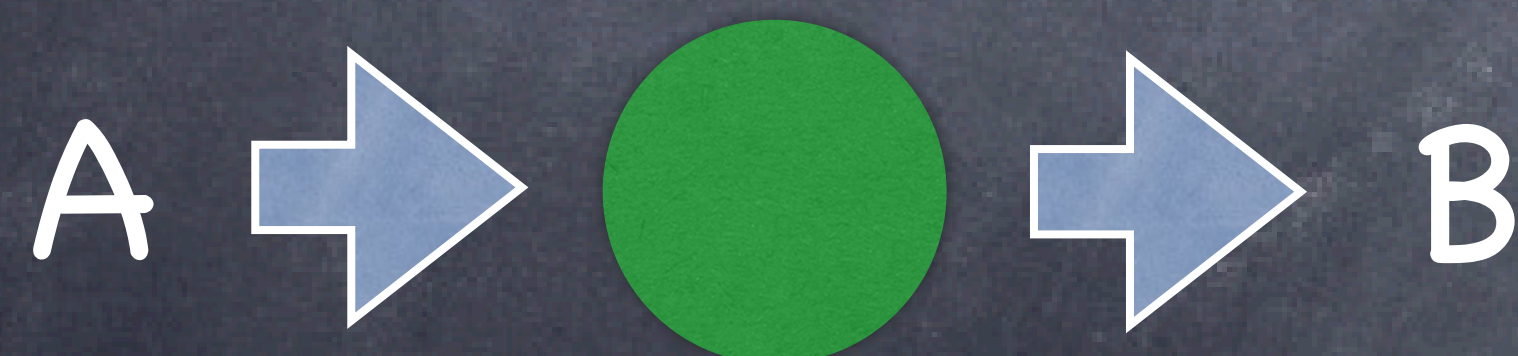
猫画像判別、自動運転、囲碁、自動お絵描きetc

何かを判断したり、推測したりを機械が行う -> **代わりにやってもらう**

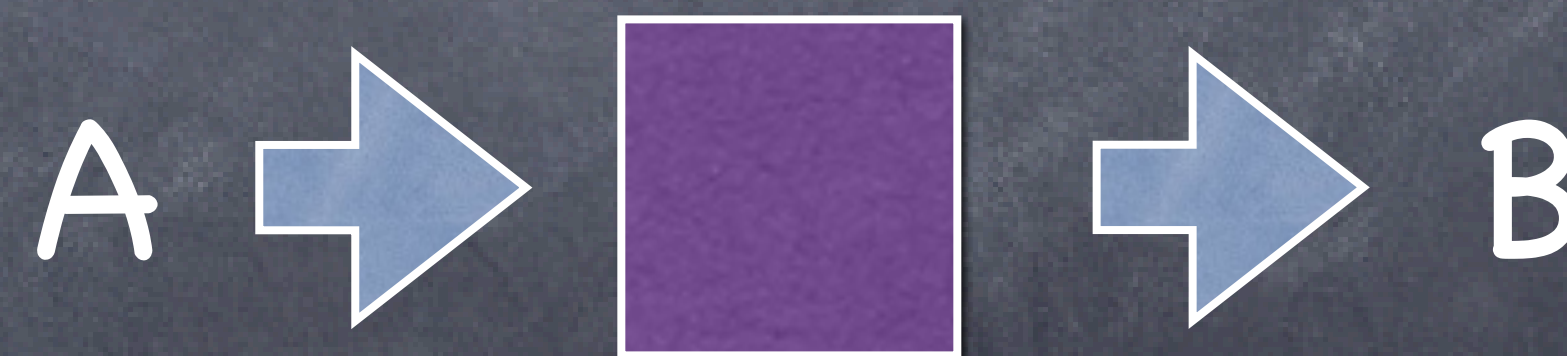


シミュレーションに機械学習を使うには？

これまで計算が大変だった部分を機械が行う



定式化済みだけど計算コスト大



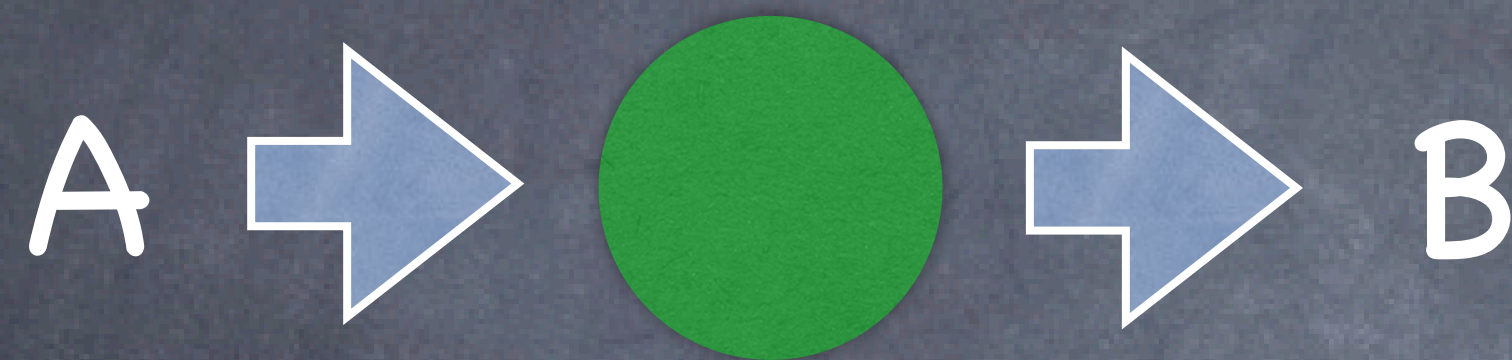
計算コストの軽い機械

シミュレーションの高速化に機械学習を用いる

今日の講演

シミュレーションに機械学習を使うには？

これまで計算が大変だった部分を機械が行う

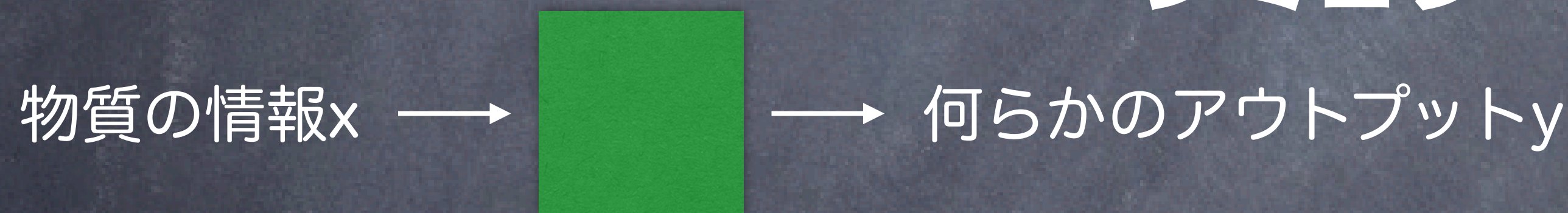


定式化済みだけど計算コスト大

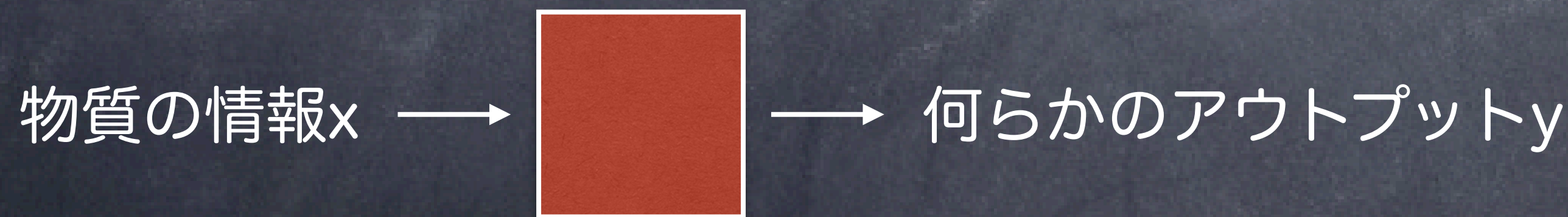


計算コストの軽い機械

シミュレーションの高速化に機械学習を用いる



$f(x)$: 場の量子論、密度汎関数理論、流体力学etc

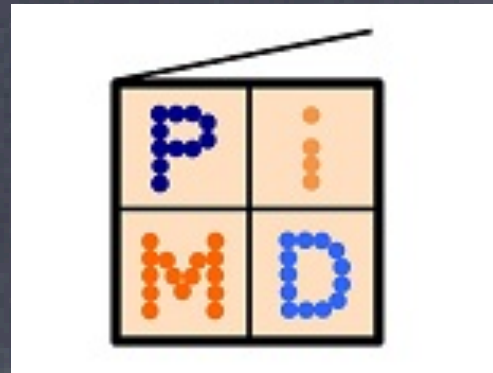


$g(x)$: ニューラルネットワーク

機械学習を使ったシミュレーション
の妥当性はどう検証するのか？

**-> 自己学習ハイブリッドモンテカル
口法の提案**

今日の講演



PIMD : CCSEが開発している原子分子シミュレーション

パッケージ(メイン開発者 : 志賀基之氏)

オープンソース:無料

原子同士に働く力を計算する様々な方法に対応している

古典ポテンシャル、第一原理計算、機械学習ポテンシャル

量子効果も考慮できる(経路積分MD)ので、水素、重水素、三重水素の違いも扱える

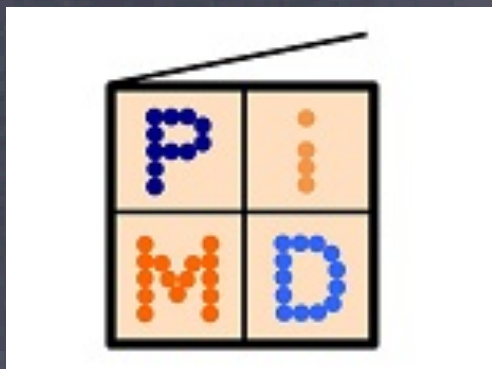
自己学習ハイブリッドモンテカルロ法が実装済み

1. 機械学習ポテンシャルをシミュレーションしながら学習できる

液体、固体、合金、様々な系で「最適な」ポテンシャルを構築できる

2. 機械学習ポテンシャルの精度に依らずに、第一原理MDの精度を達成できる

今日の講演



PIMD : CCSEが開発している原子分子シミュレーション
パッケージ(メイン開発者 : 志賀基之氏)

- ・ 構造最適化、基準振動解析、フォノン計算、弾性定数
- ・ 反応経路探索 : ストリング法、最急降下法、最緩上昇法
- ・ 古典統計 : レプリカ交換法
- ・ 量子統計 : 経路積分分子動力学法
- ・ 古典力学 : 分子動力学法、ハイブリッド・モンテカルロ法
- ・ 半古典動力学 : セントロイドおよびリング・ポリマー分子動力学法
- ・ 非断熱動力学 : 平均場動力学法、サーフェス・ホッピング法
- ・ 自由エネルギー計算 : メタ・ダイナミクス法、拘束分子動力学法、平均力ダイナミクス
- ・ 機械学習 : 自己学習・ハイブリッドモンテカルロ法(new!)

アンサンブル

- ・ エネルギー一定 (NVE)、温度一定 (NVT)、圧力一定 (NPT)、応力一定 (NtT)

境界条件

- ・ 自由境界、周期境界 (立方セル、平行六面体セル)

ポテンシャル

- ・ 第一原理的手法 : SMASH
- ・ 古典力場 : ADP(new!), AMBER、CHARMM、CLAYFF(new!), EAM、OPLS、Tersoff(new!)
- ・ 分極力場 : 誘起双極子とTholeダンピング補正 (new!)
- ・ 他のコードとのインターフェース (第一原理的手法) : ABINIT-MP、CP2K(new!)、GAMESS、GAUSSIAN、MOLPRO、NTCHEM、ORCA、QUANTUM ESPRESSO、TURBOMOLE、VASP
- ・ 他のコードとのインターフェース (半経験的手法) : DFTB、MOPAC
- ・ マルチスケール法 : ONIOM、QM/MM
- ・ 機械学習ポテンシャル : AENET
- ・ ユーザー定義のサブルーチン

物理学と機械学習は相性が良い

物理学では何をするか

観測する



ボールの落下地点を
記録する

法則を見つける

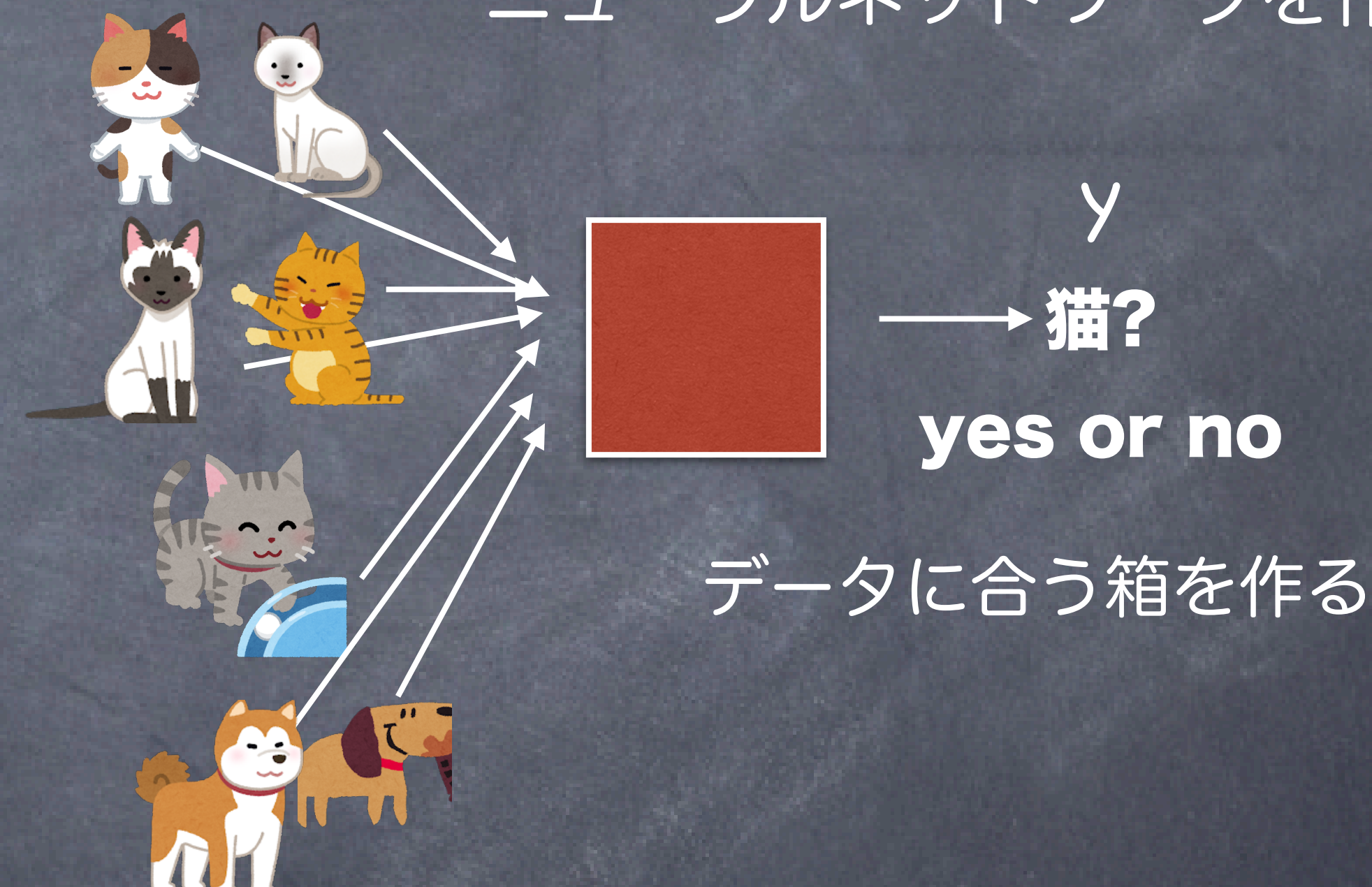
ニュートンの法則？
空気抵抗を無視してみる
大体合う？

もっと精緻に考える？
形状も考える？

理論を構築し、
解いて予測

機械学習では何をするか

ニューラルネットワークを作る



データに合う箱を作る

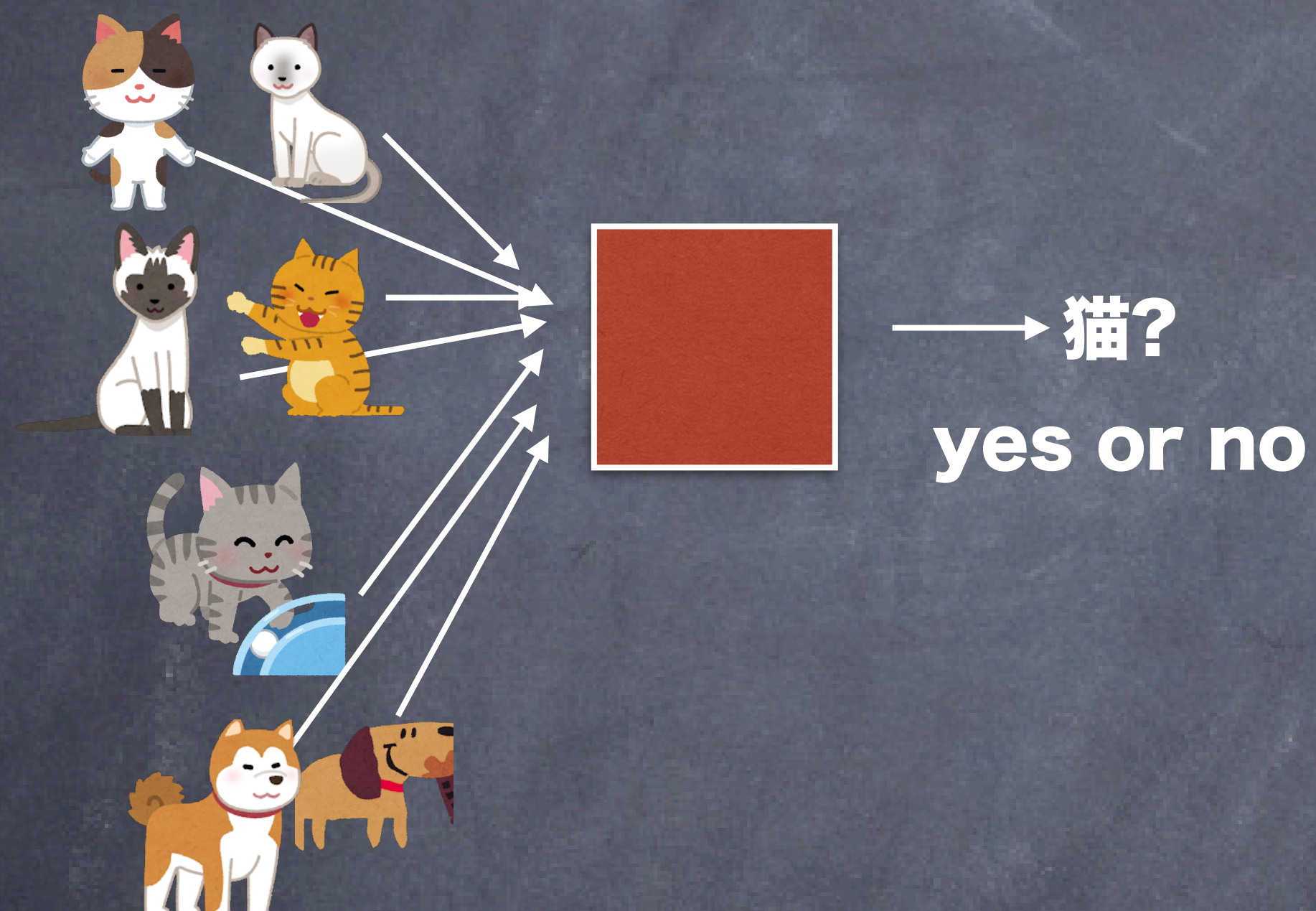
データを集める

箱を作り、
入れて予測

機械学習をどう使うか

機械学習

学習によって作られたニューラルネットワーク



人間は「猫を識別する理論」を知らない

が「猫を識別する関数」は作れる

知っている理論に適用すると何が起こる？

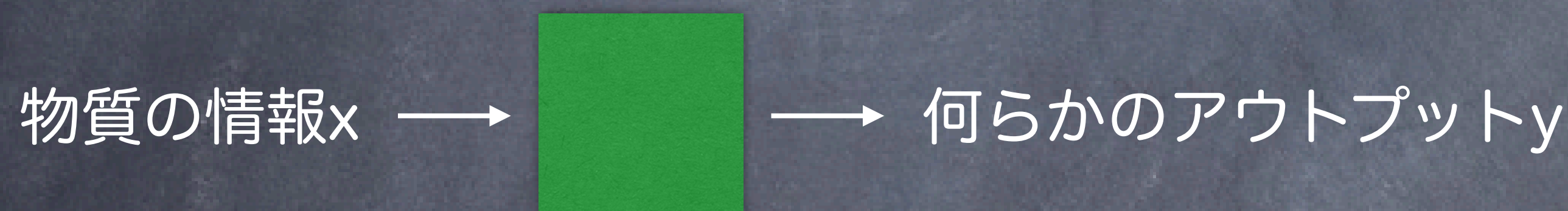
物理学で、「インプット x から何かを計算する理論 $f(x)$ 」を知っているとする
「インプット x から何かを計算する関数 $g(x)$ 」をニューラルネットで作る

$f(x)=g(x)$ となる $g(x)$ を作ると何が嬉しいか？

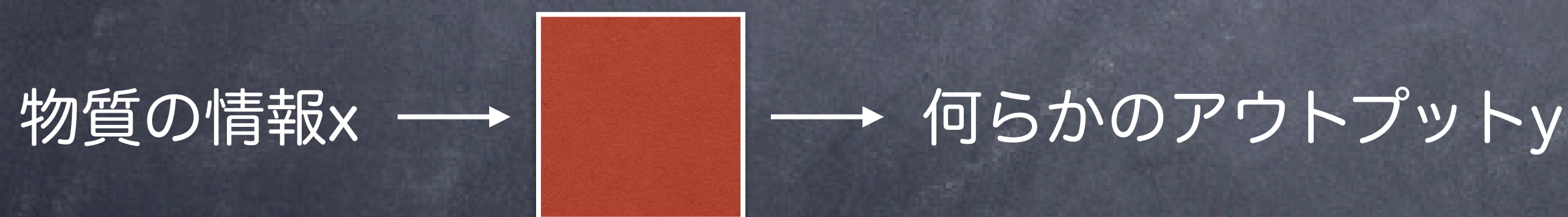
機械学習をどう使うか

物理学で、「インプット x から何かを計算する理論 $f(x)$ 」を知っているとする
 「インプット x から何かを計算する関数 $g(x)$ 」をニューラルネットで作る
 $f(x)=g(x)$ となる $g(x)$ を作ると何が嬉しいか？

物理学

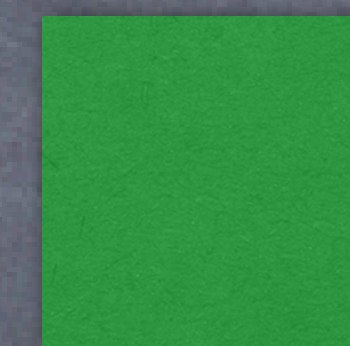


$f(x)$: 場の量子論、密度汎関数理論、流体力学etc



$g(x)$: ニューラルネットワーク

違いは何か



$f(x)$: 非常に計算が大変な場合がある



$g(x)$: 学習が終われば計算は一瞬

計算の高速化に使うことができる！

原子・分子シミュレーションと機械学習

量子力学と物質

我々は量子力学を知っている

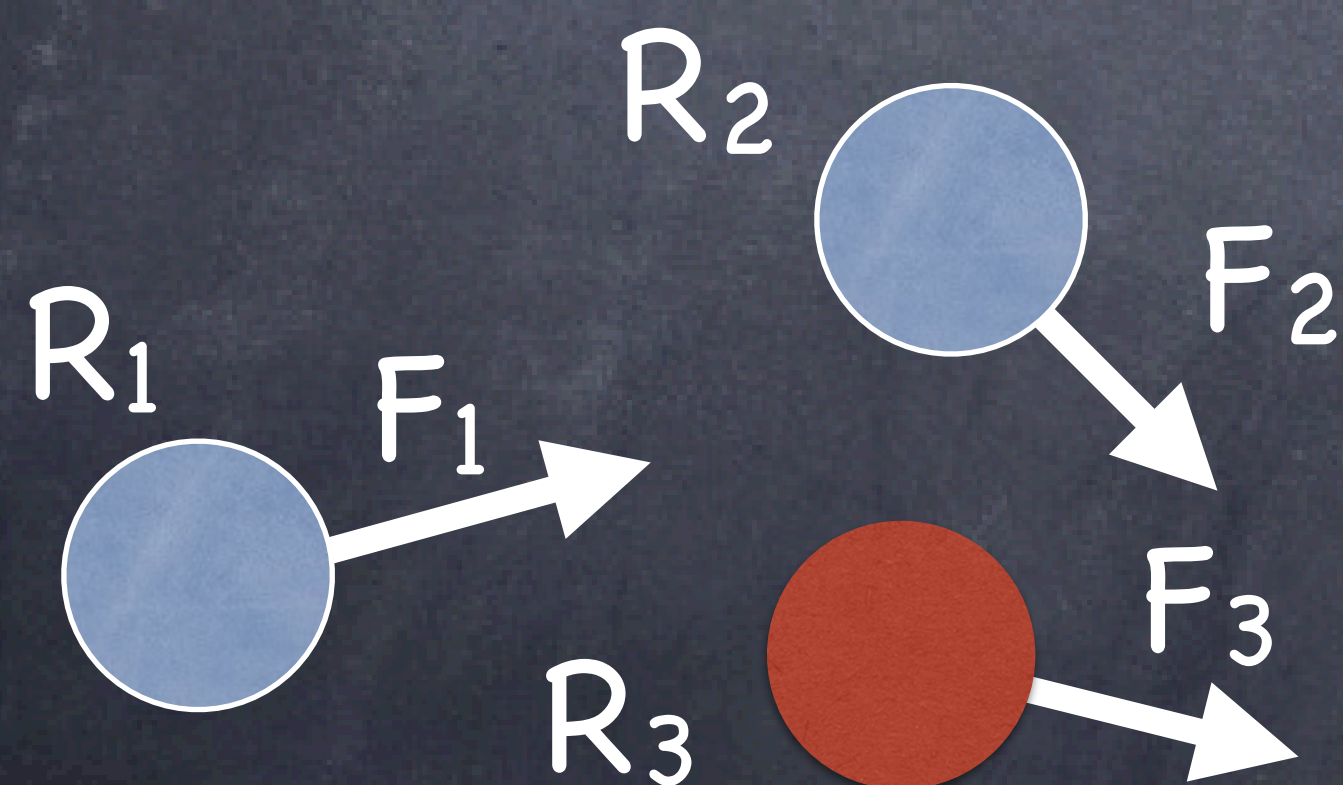
$$\hat{H}\psi(\vec{r}_1, \dots, \vec{r}_N, \vec{R}_1, \dots, \vec{R}_M) = E\psi(\vec{r}_1, \dots, \vec{r}_N, \vec{R}_1, \dots, \vec{R}_M) \quad \text{注：時間依存がない場合}$$

電子位置 r と原子核の位置 R に関する多体のシュレーディンガー方程式を知っている

これが解ければ物質に関する殆どのがわかる (はず)

密度汎関数理論の登場(1998年ノーベル化学賞Walter Kohn)によって

$E^{\vec{R}_1, \dots, \vec{R}_M}$ は計算できるようになった (原子核位置 R を固定し電子の量子性を考慮)



あらゆるものは原子からできている

固体、液体、気体、電池、生体、創薬、ウイルスetc.
計算ができればなんでもシミュレーションできる？

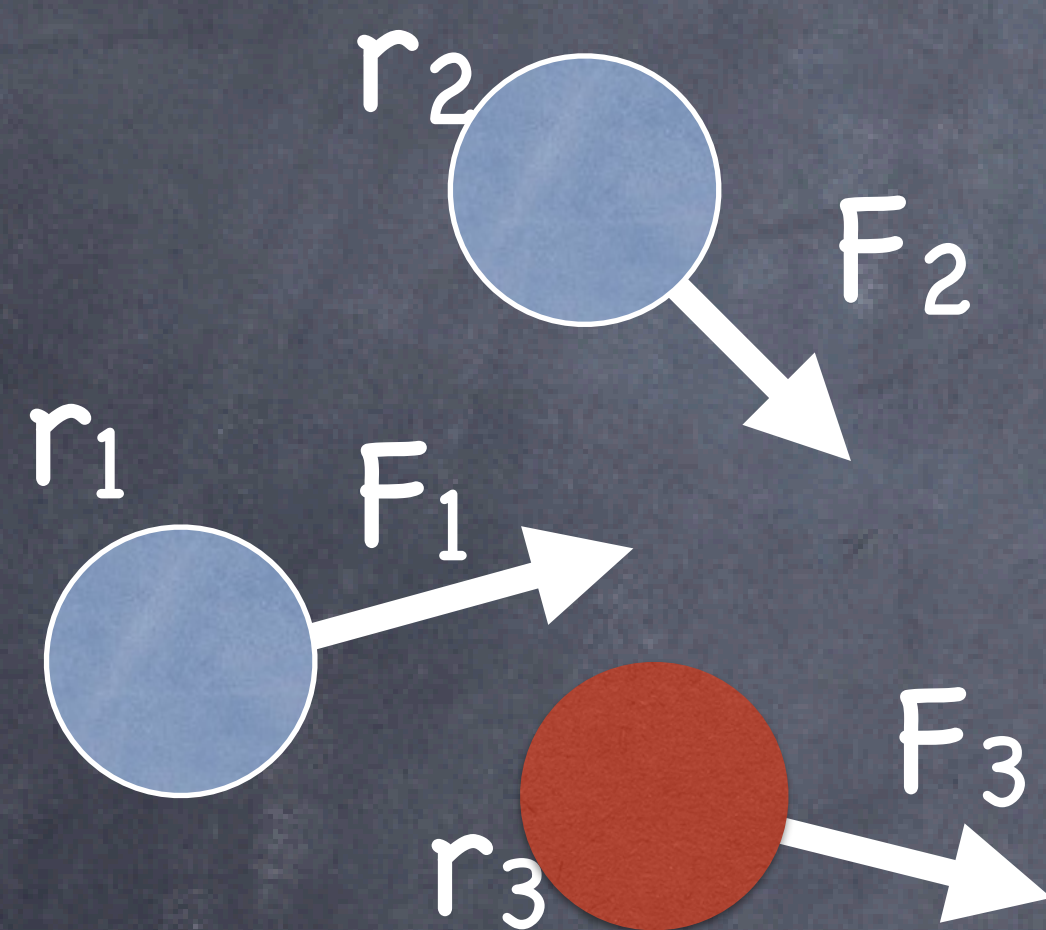
-> 計算が重たい!

分子動力学法 (MD)

原子や分子間に働く力がわかればシミュレーションできる

Lennard-Jonesポテンシャル等

エネルギーから力は計算できる



$$F_1 = \frac{\partial E(r_1, \dots, r_N)}{\partial r_1} \quad F_2 = \frac{\partial E(r_1, \dots, r_N)}{\partial r_2} \quad F_3 = \frac{\partial E(r_1, \dots, r_N)}{\partial r_3}$$

精度の良いエネルギー評価ができれば精度の良い力がわかる

第一原理計算 -> 精度の良いエネルギーの評価が可能

密度汎関数理論(DFT)

各時刻ごとにDFT計算を行えば、精度の良い力が計算できる

典型的には、1fsごとにDFT計算を行う

f -> p -> n -> μ -> m -> 1s

ニューラルネットに置き換えよう

Behler-Parrinello型ニューラルネットワーク

第一原理計算 -> 高精度な力の評価 とても重たい -> ニューラルネットワークでフィット

“Neural network models of potential energy surfaces”
T. B. Blank et al., J. Chem. Phys. **103**, 4129 (**1995**)

2007 年 革新的な論文が登場

“Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces”, J. Behler and M. Parrinello, Phys. Rev. Lett. **98**, 146401 (2007)



1000倍以上高速に

やってみたい！そんな時には

原子・分子シミュレーション用の
ニューラルネットワークを作るソフトウェア

オープンソースソフトウェア



The Atomic Energy Network (aenet)

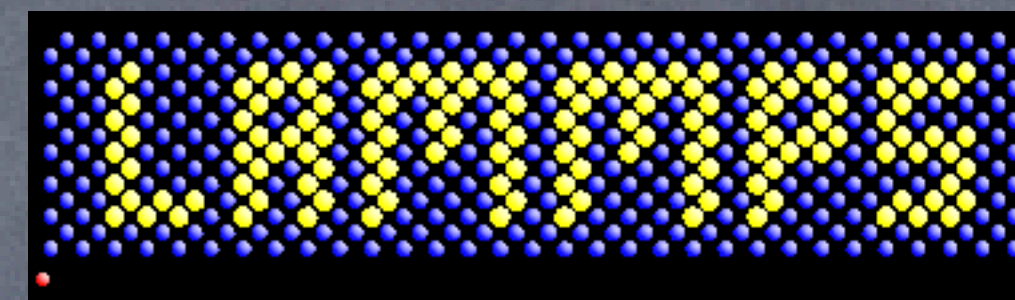
A neural network potential package (n2p2)

市販ソフトウェア

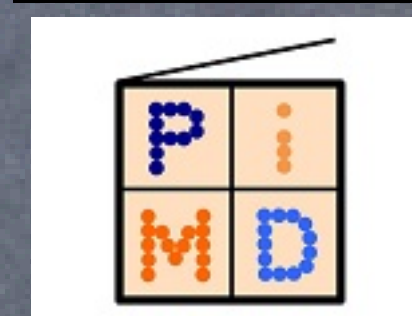
ニューラルネットワーク分子動力学システム Advance/NeuralMD

汎用原子レベルシミュレータ Matlantis

MDを行うソフトウェア



LAMMPS



PIMD：我々（原子力機構）が開発

新たなプログラミングなしに機械学習分子シミュレーションが
できる環境が整っている

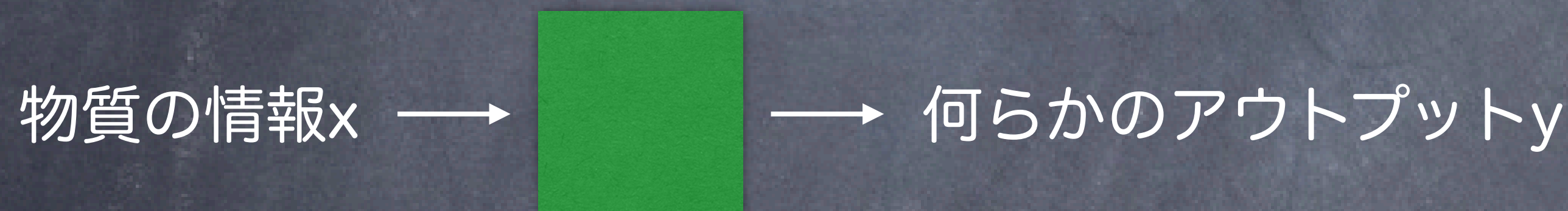
機械学習シミュレーションの問題点と解決策： 自己学習ハイブリッドモンテカルロ法

機械学習の(本質的な?)問題点

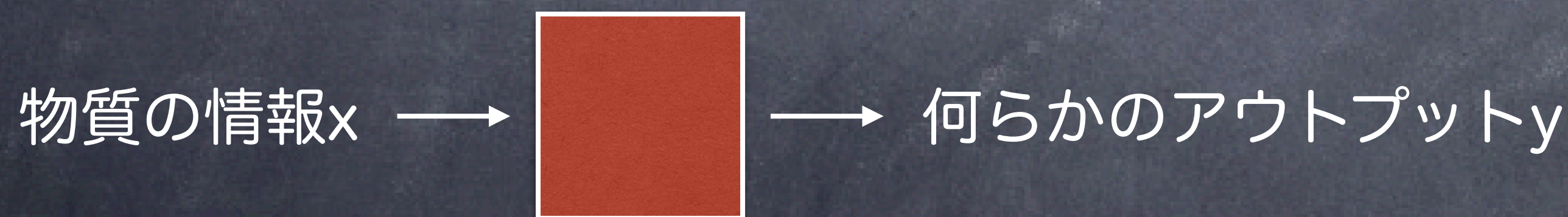
機械学習は何かを模倣する $g(x) \sim f(x)$

与えられたインプットから、オリジナルによく似たアウトプットを出力する

機械学習をシミュレーションに使う場合



$f(x)$: 場の量子論、密度汎関数理論、流体力学etc



$g(x)$: ニューラルネットワーク

ちゃんと計算できる部分を機械学習に置き換えた事による誤差は？

機械学習を使ったシミュレーションの妥当性はどう検証するのか？

完全に一致することはないが、どこまで精度を高めれば良いのか？

機械学習MDの問題点について

訓練データはどうやって集めるのか？

1. 圧縮したり引き伸ばしたり歪めたりして作った色々な構造のデータを集める

様々な構造で第一原理計算を行う → 訓練データが集まる

問題点 1：どんな構造を入れるべき？（構造候補は無限にある）

問題点 2：どのくらいの数の訓練データを集めれば、やりたいシミュレーションを再現するに足る十分な数だと結論づけられるのか？

1000 ? 10000 ? 100000 ?

2. 第一原理MDを行なって得られる構造データを集める

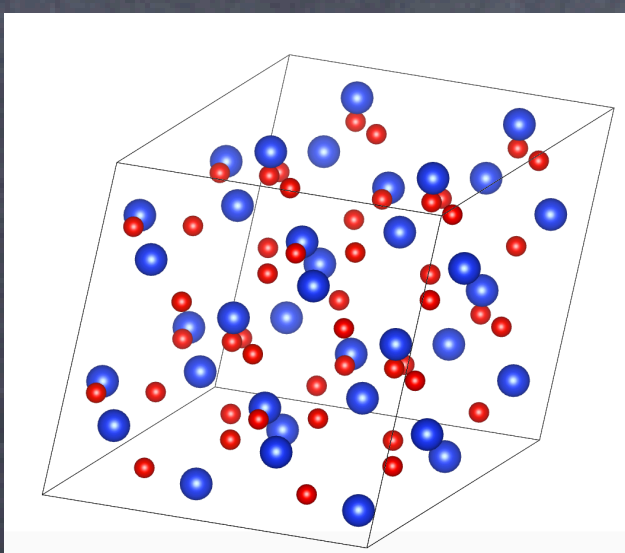
毎ステップごとに新しい構造データが得られる

問題点 1：どのくらいの長さのMDを行えばいいのか？どのくらいの数の訓練データを集めればやりたいシミュレーションを再現するに足る十分な数だと結論づけられるのか？

1000steps? 10000steps? 100000steps?

問題点 2：そもそも第一原理MDの計算コストが大きいために機械学習MDを使いたいわけで

第一原理MDでデータを集めていたら時間がかかってしまう



機械学習MDの問題点について

構築したニューラルネットワークの精度はどうやって評価するのか

機械学習ポテンシャルはDFTポテンシャルの近似
悪いニューラルネットワークは悪い力を生む

大きな系を取り扱いたい

大きな系は教師データがないが、使っているNNの精度をどうやって知ればいいのか？
(第一原理MDができないような大きな系で使いたないので、教師データがない)

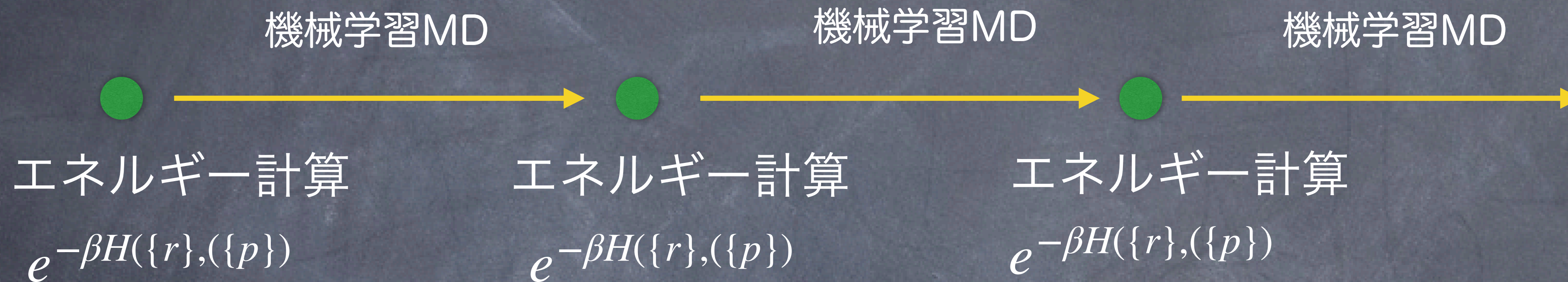
機械学習MDは原子種の数が増える（4原子種以上）と不安定になりがち

複雑なインプットから作ったニューラルネットワークの精度は悪くなりがち

**ニューラルネットワークの精度の良し
悪しにに振り回されたくない！**

->自己学習ハイブリッドモンテカルロ法を使おう

自己学習ハイブリッドモンテカルロ法



従来のHMC: MD部分は第一原理MD

自己学習HMC: MD部分を機械学習MDに置き換えておく

MLMDとMCを合わせた手法 自己学習HMCの計算精度は第一原理MDと等しい

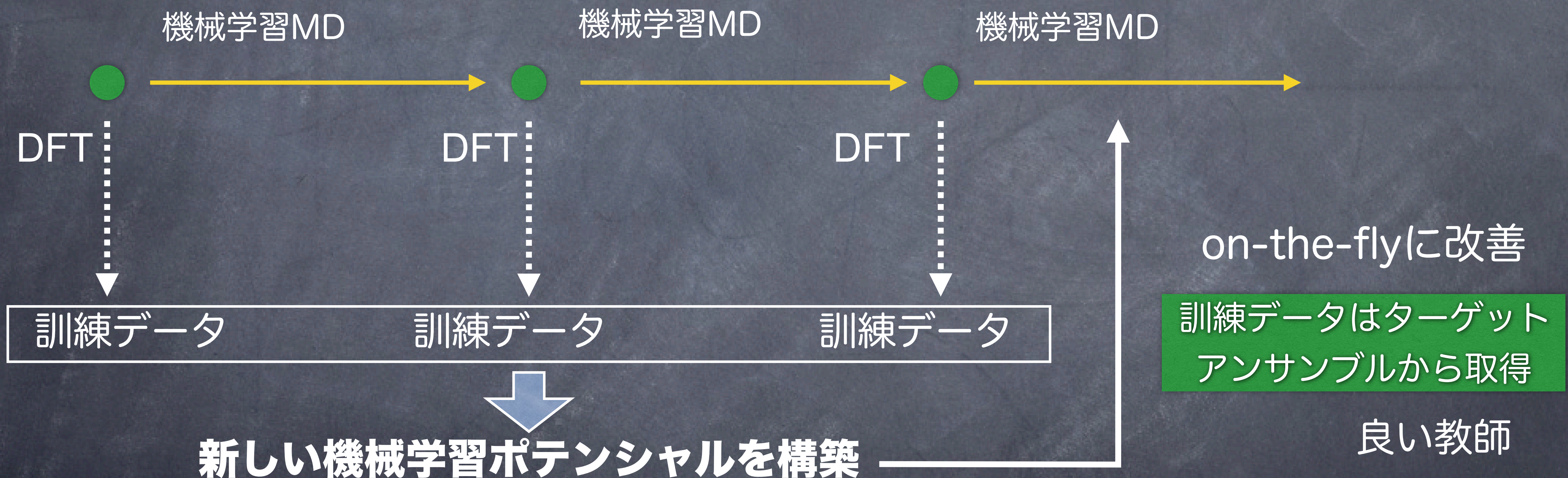
ニューラルネットワークを使っているのに、計算精度が元の手法と同じになっている！

ニューラルネットワークの置き換えによる高速化と、第一原理計算の精度を両立

SLHMC法の計算について

機械学習MD: 近似されたポテンシャルを利用 -> 厳密ではない

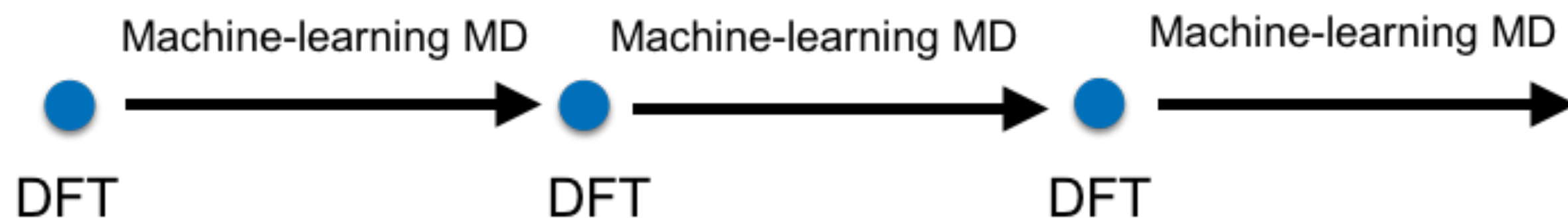
SLHMC: 近似されたポテンシャルを利用-> それでも厳密! Metropolis-Hastings法的一种なので



機械学習ポテンシャルがシミュレーション中に逐次的に改善される

自己学習ハイブリッドモンテカルロ法 (SLHMC)

1. 厳密: 第一原理MDと同じ精度を持つが高速に計算ができる
2. On-the-fly: 機械学習ポテンシャルはシミュレーションの最中に自動的に改善される



1. The trial configuration is generated by the machine-learning MD

$$\dot{\mathbf{p}}_i = -\frac{\partial V_{\text{ML}}(\{\mathbf{r}\}, t)}{\partial \mathbf{r}_i}, \quad \dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{m_i},$$

2. The trial configuration is accepted by the following probability

$$P_{\text{acc}}(\{\mathbf{p}, \mathbf{r}\} \rightarrow \{\mathbf{p}', \mathbf{r}'\}) = \min \left(1, e^{-\beta(H_{\text{DFT}}(\{\mathbf{p}', \mathbf{r}'\}) - H_{\text{DFT}}(\{\mathbf{p}, \mathbf{r}\}))} \right)$$

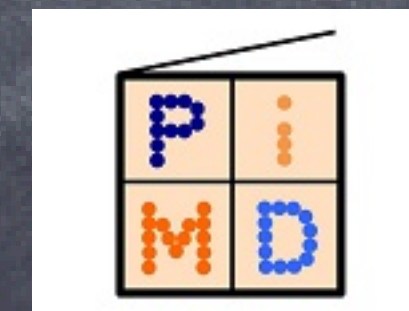
$$H_{\text{DFT}} = \sum_i^N \frac{|\mathbf{p}_i|^2}{2m_i} + V_{\text{DFT}}(\{\mathbf{r}\}).$$

DFTポテンシャルに対するメトロポリス-ヘイスティングス法

ハイブリッドモンテカルロ法の一つ
統計的に厳密な結果

以後の計算には我々が開発したPIMDを使用

PIMD



<https://ccse.jaea.go.jp/software/PIMD/index.en.html>

ニューラルネットにはaenetを使用

PIMDによる原子分子シミュレーション

様々な分子シミュレーション手法

	Accuracy	Computational cost
Classical MD	low	very low
First-principles MD	very high	very high
Machine-learning MD	high	low
<u>Self-learning Hybrid Monte Carlo</u>	very high	middle

SLHMCを使うと嬉しい時：

1. DFTレベルの精度の計算をしたいがDFT-MDの計算コストが大きすぎて断念してしまっている時
2. 機械学習MDのための高精度なニューラルネットワークを構築したい時

どちらの場合でも、ニューラルネットワークは自動的に改善される

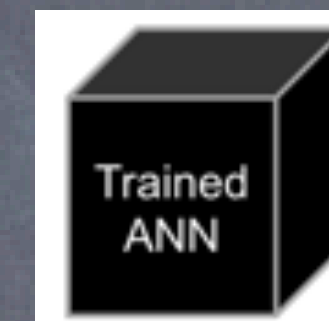
逐次的訓練と厳密な結果

α -quartz (SiO_2)

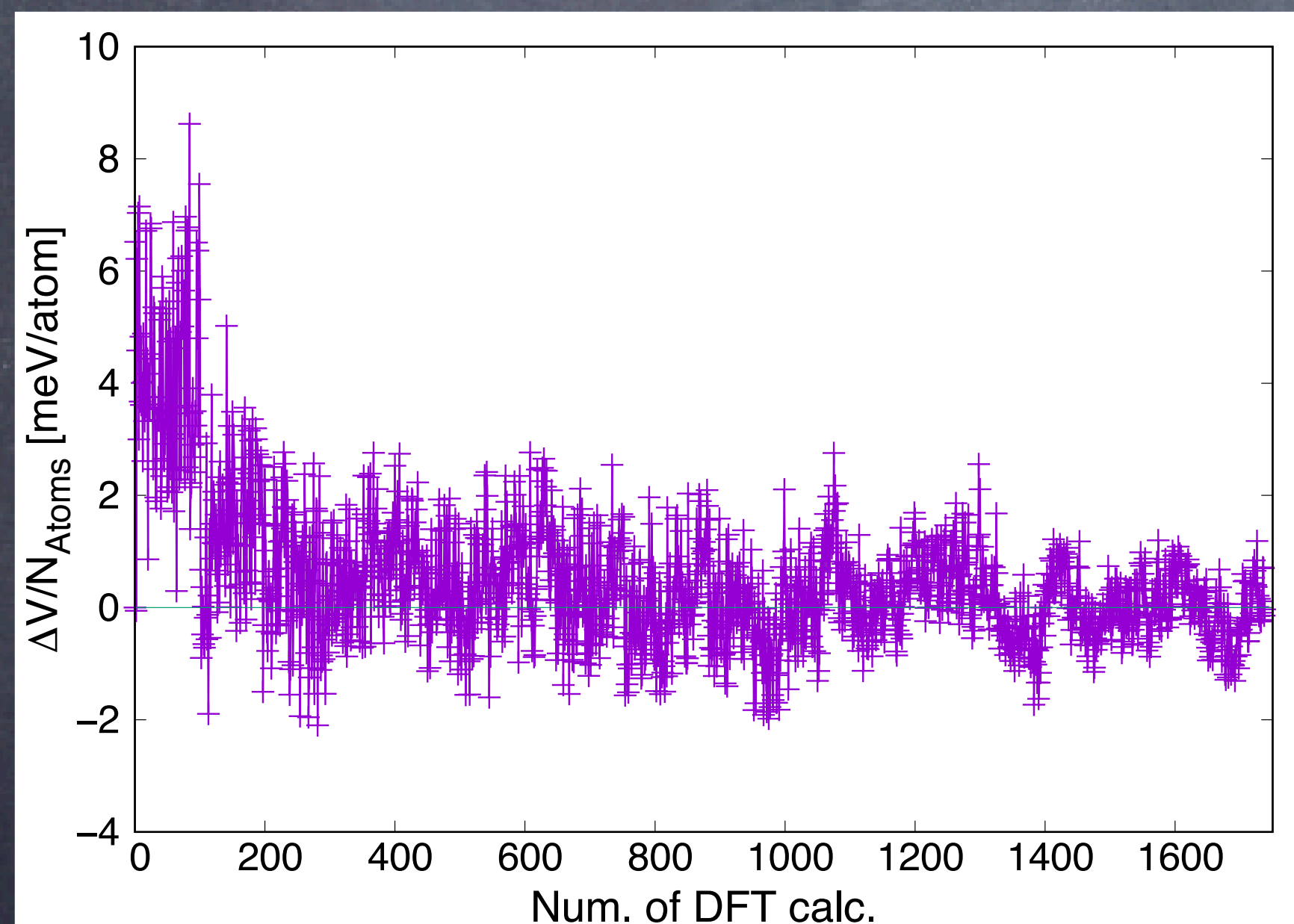
2x2x2 スーパーセル 300K NVT

DFT : VASP

ニューラルネットワーク：隠れ層2層、各15ユニット

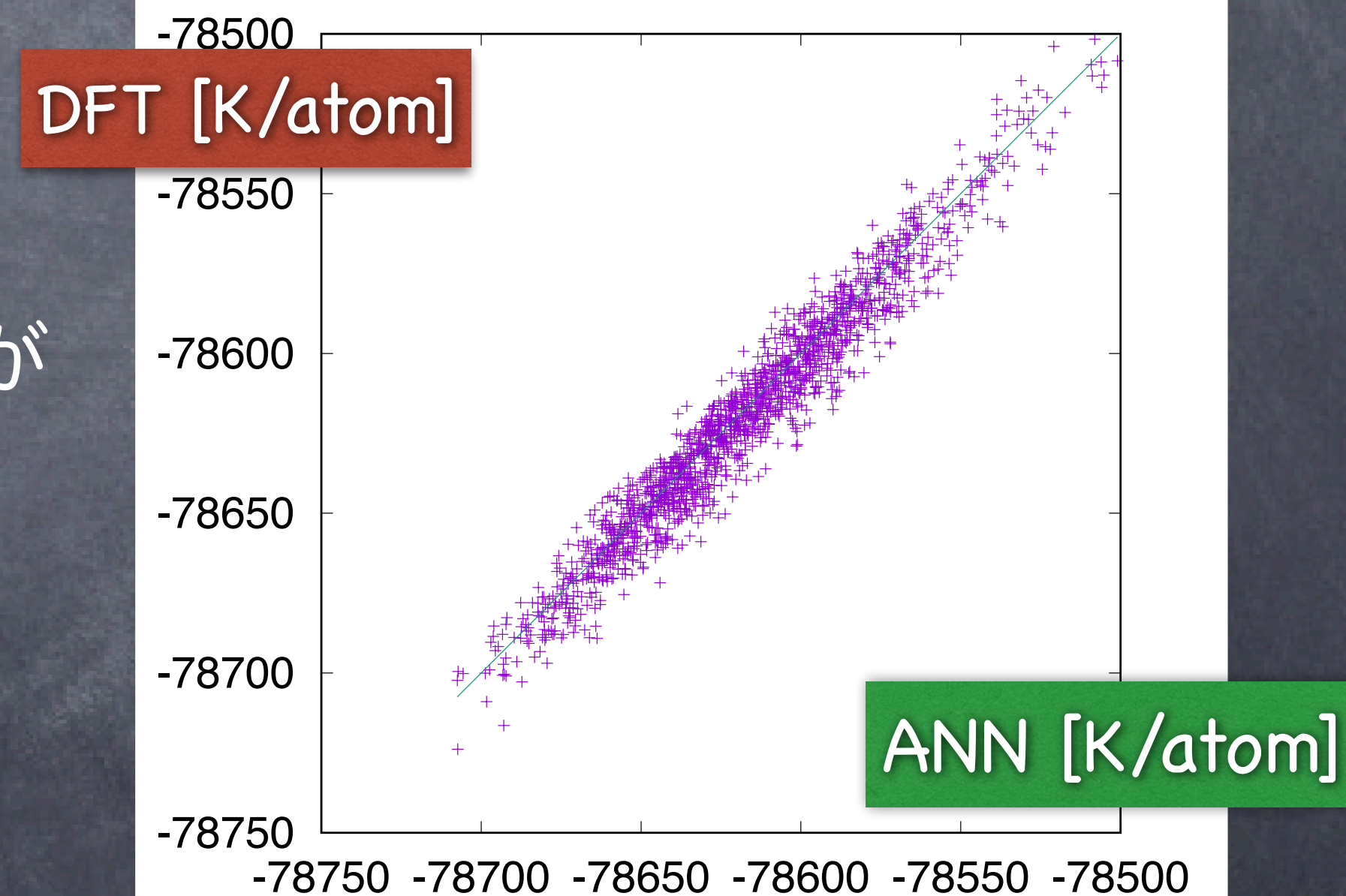


PIMD with aenetを使用



1 meV/atomの精度

ニューラルネットワークが自動的に改善されていく



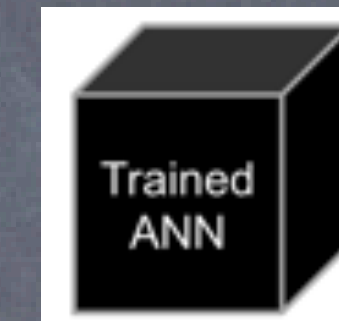
逐次的訓練と厳密な結果

α -quartz (SiO_2)

2x2x2 スーパーセル 300K NVT

DFT : VASP

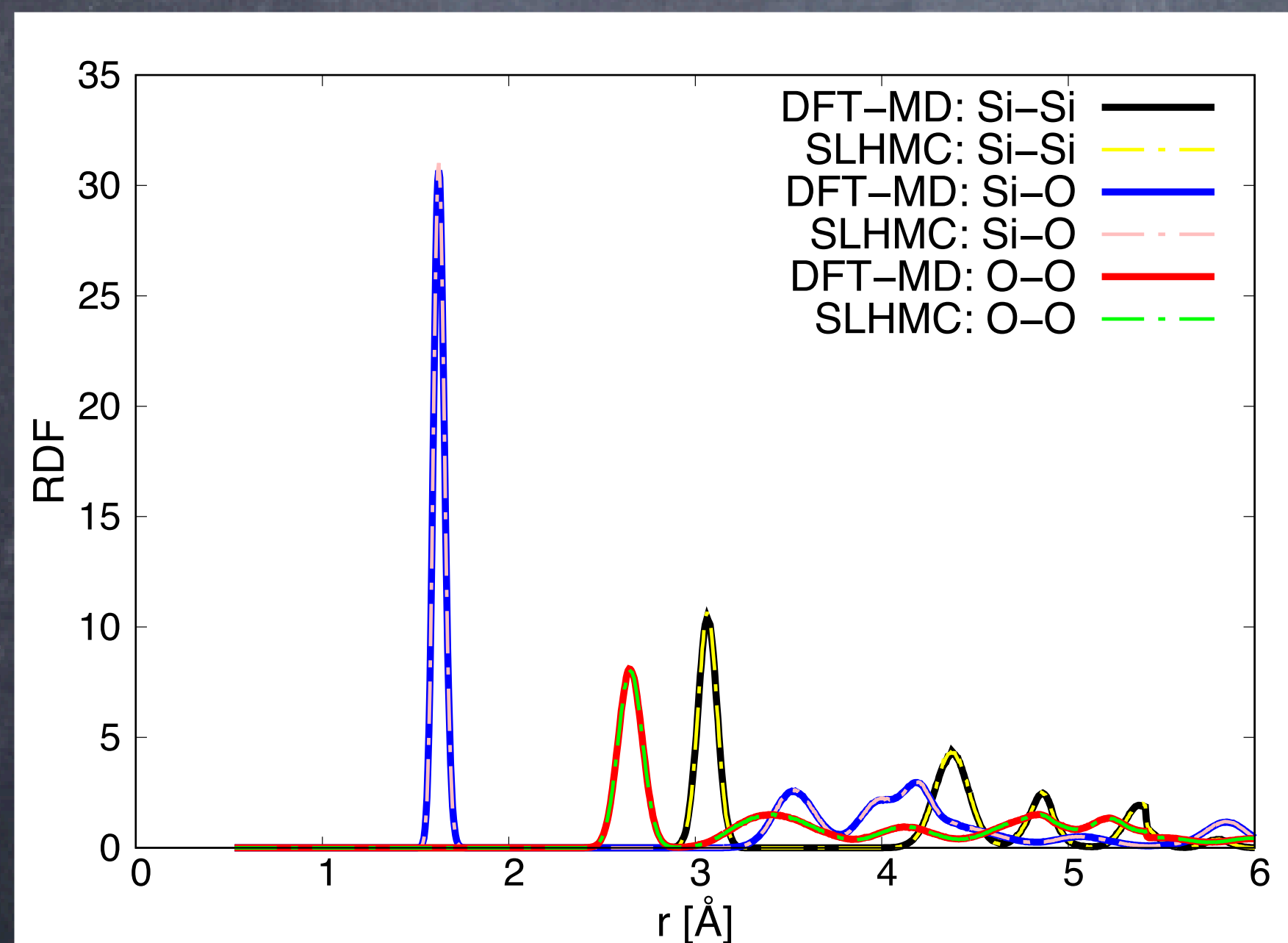
ニューラルネットワーク：隠れ層2層、各15ユニット



PIMD with aenetを使用

SLHMCはDFTポテンシャルに対する
Metropolis-Hastings法

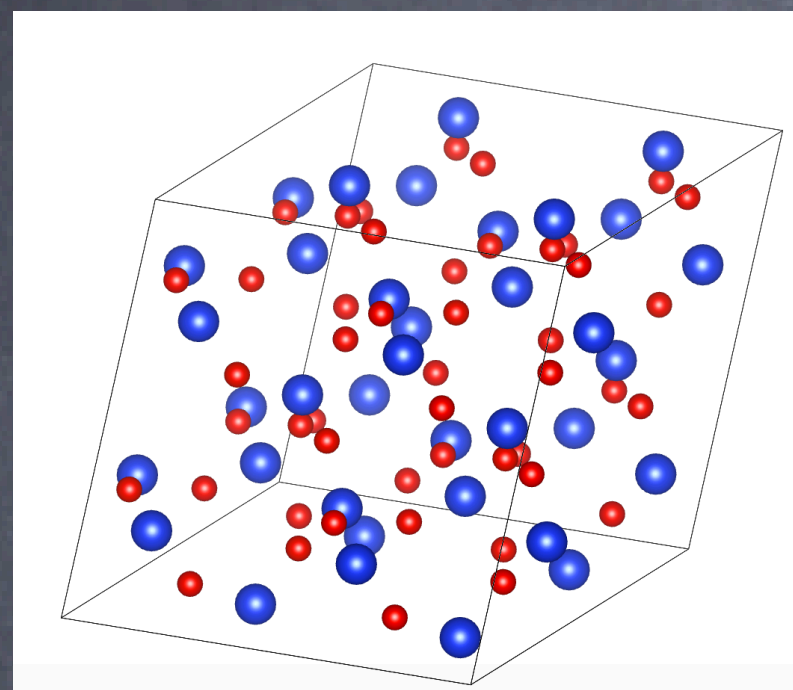
1. 訓練データ：対象とするアンサンブルそのもの
効率的なサンプリングと訓練が可能
2. ニューラルネットワークの精度に依らず、常に厳密
結果の精度が保証されている！



動径分布関数

逐次的訓練と厳密な結果

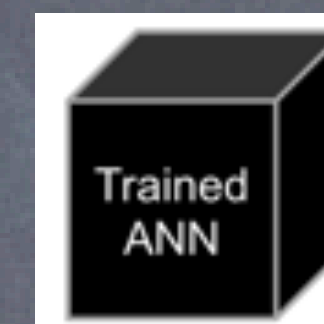
α -quartz (SiO_2)



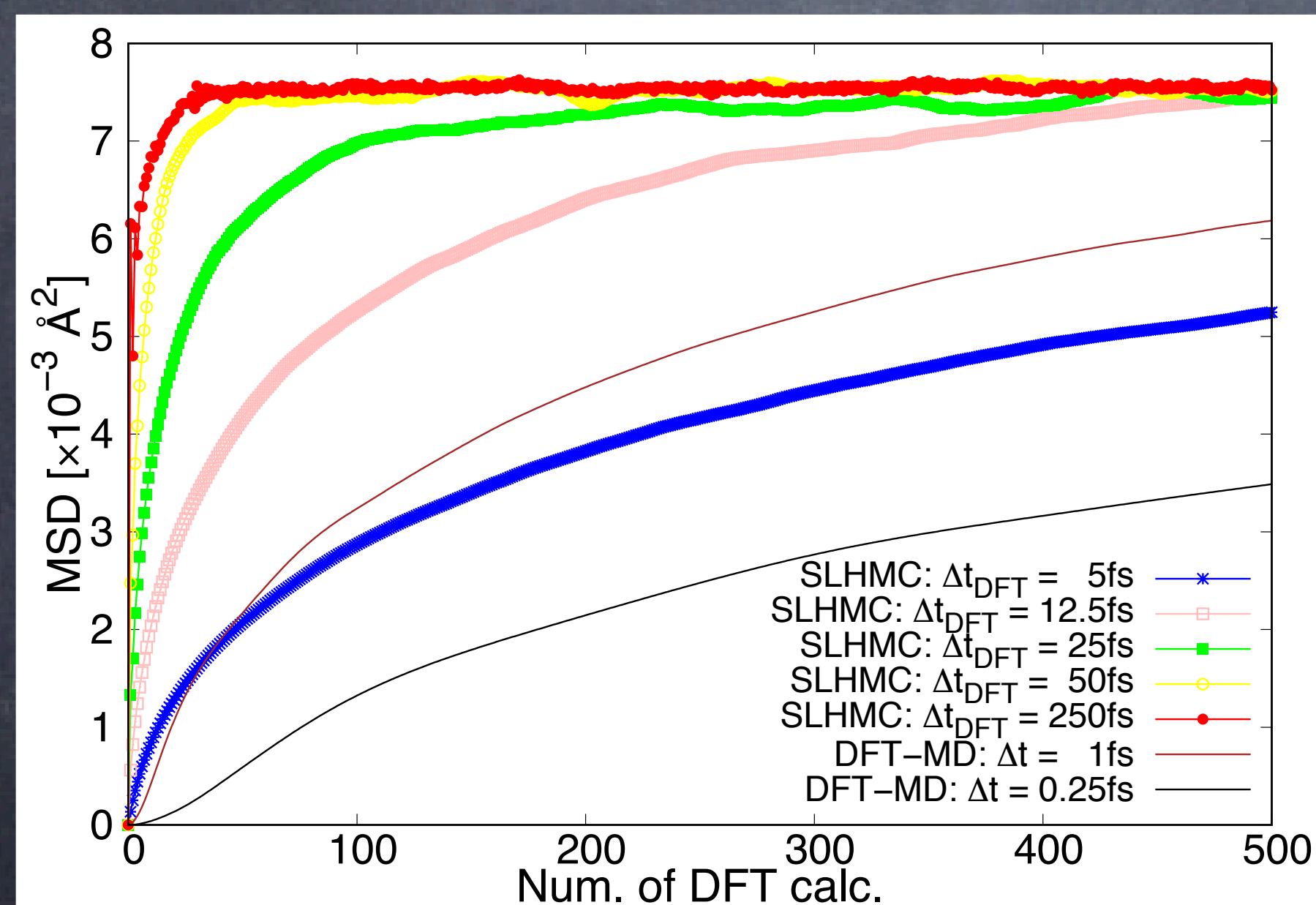
2x2x2 スーパーセル 300K NVT

DFT : VASP

ニューラルネットワーク：隠れ層2層、各15ユニット

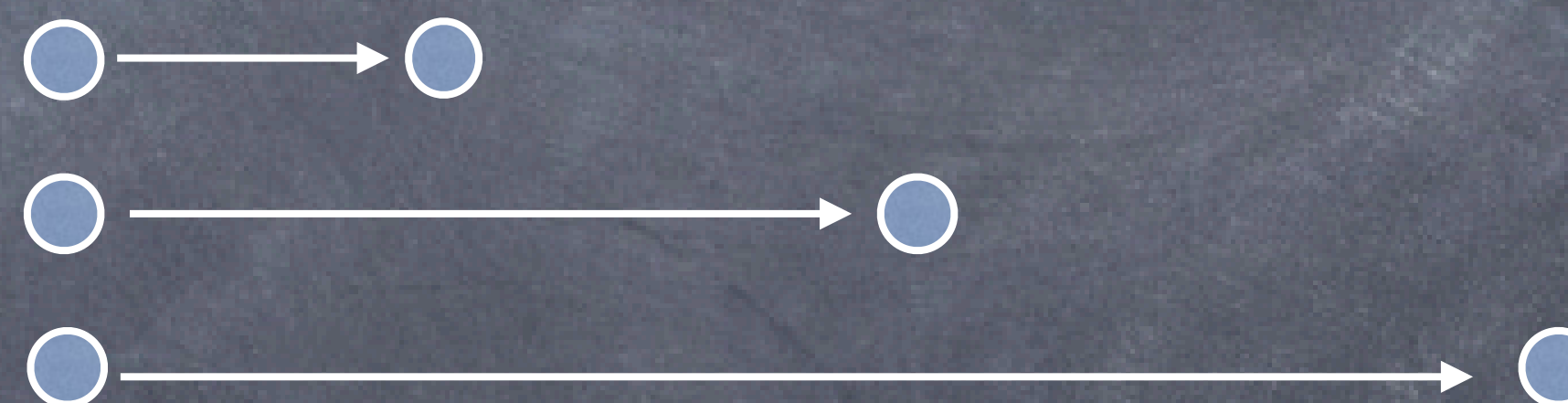


PIMD with aenetを使用



平均二乗変位

計算コストがDFT-MDと比べて劇的に減少

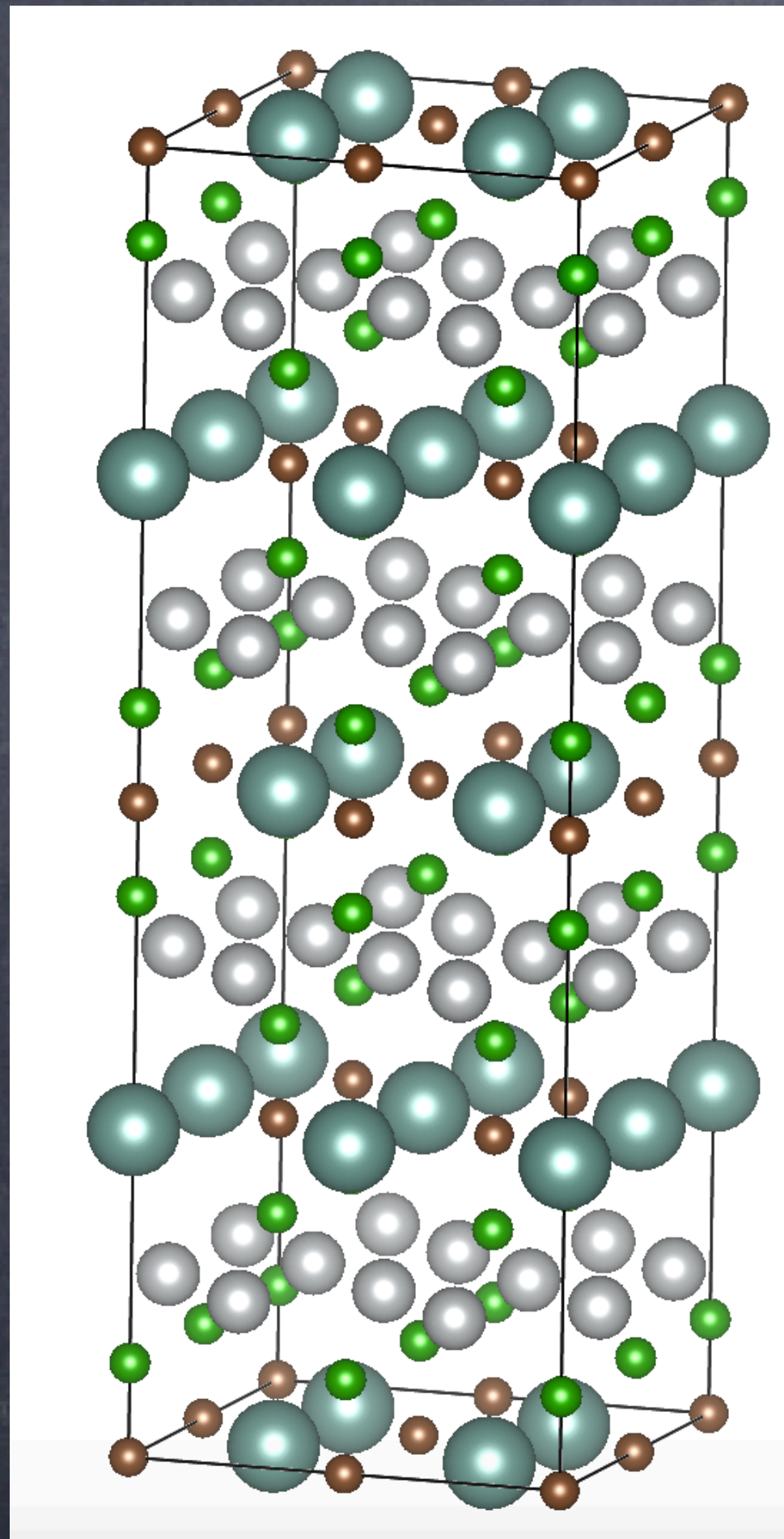


機械学習MD

DFT計算の回数が減っている

高精度な機械学習ポテンシャルによる長時間MD

SLHMCで機械学習ポテンシャルを訓練



2x2x2

YNi₂B₂C:

フォノン由来の非従来型s波超伝導体($T_c \sim 15K$)

DFT: VASP

PIMD with aenetを使用

1000K HMC -> 高精度な機械学習ポテンシャルを構築

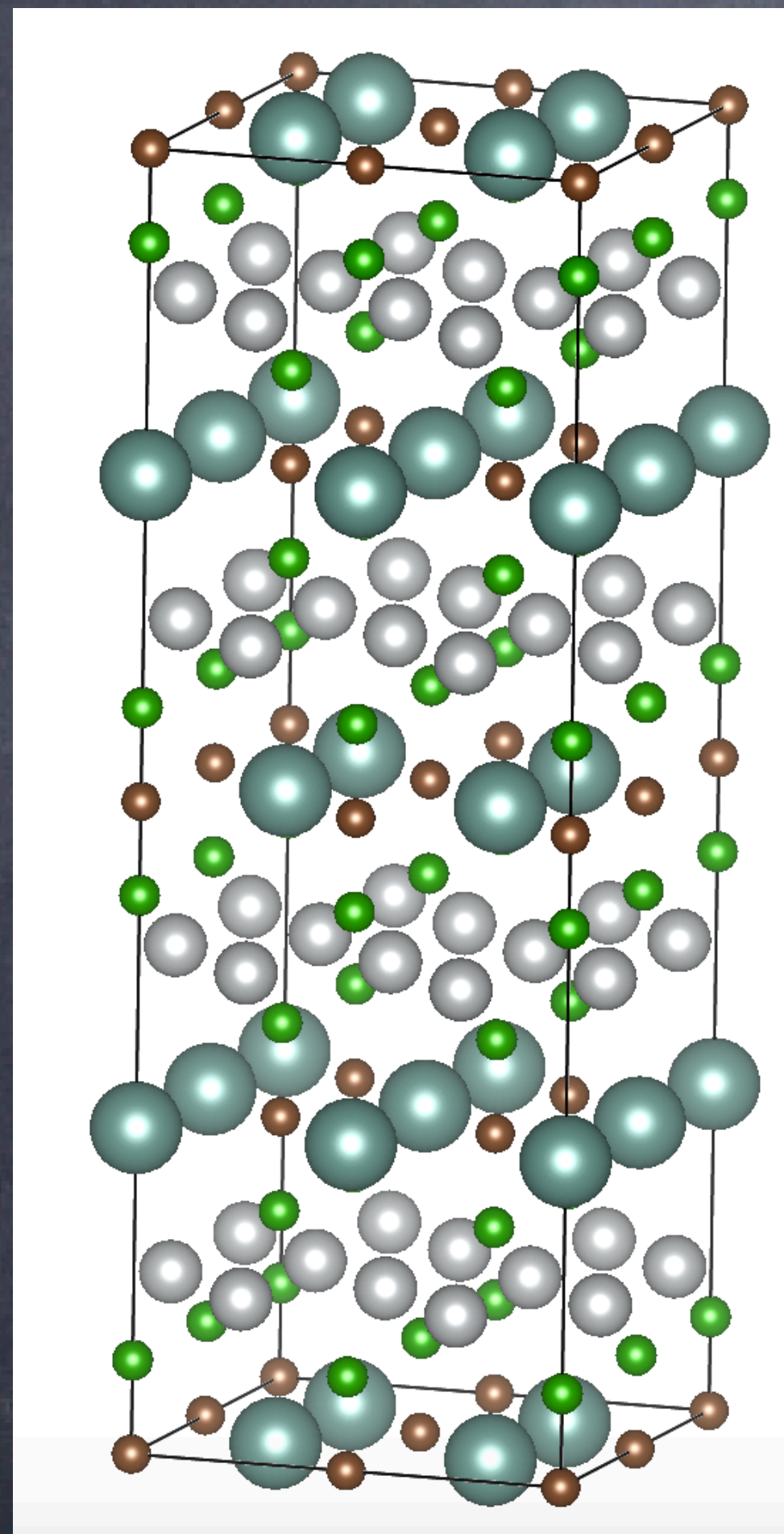
1 fs 間隔で100,000 ステップで MD x 10

通常のDFT-MDではこの長さの計算は大変

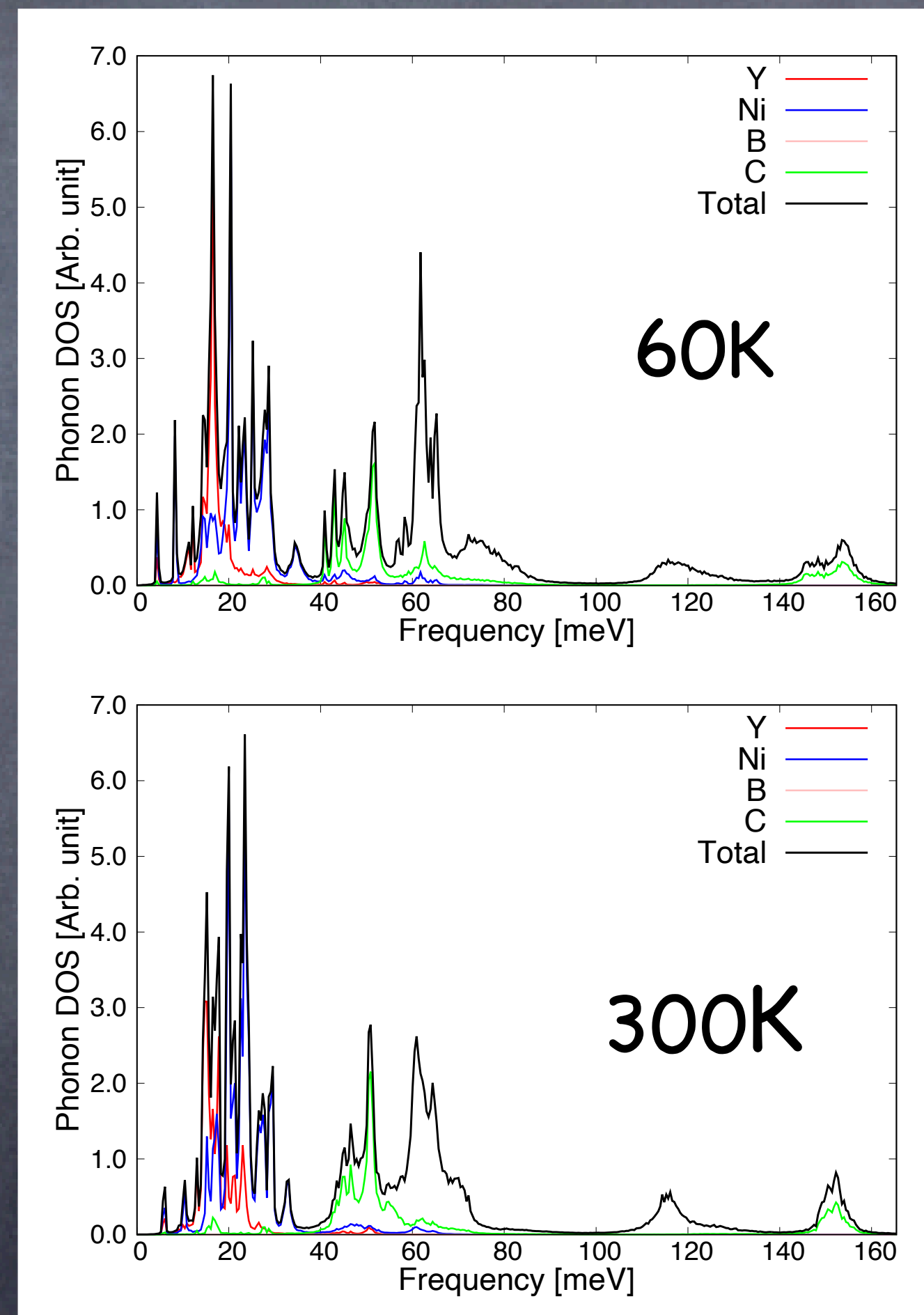
高精度な機械学習ポテンシャルによる長時間MD

SLHMCで機械学習ポテンシャルを訓練

フォノン状態密度



2x2x2



1. SLHMCで訓練
2. ML-MDを実行

1000K HMC -> 高精度な機械学習ポテンシャルを構築

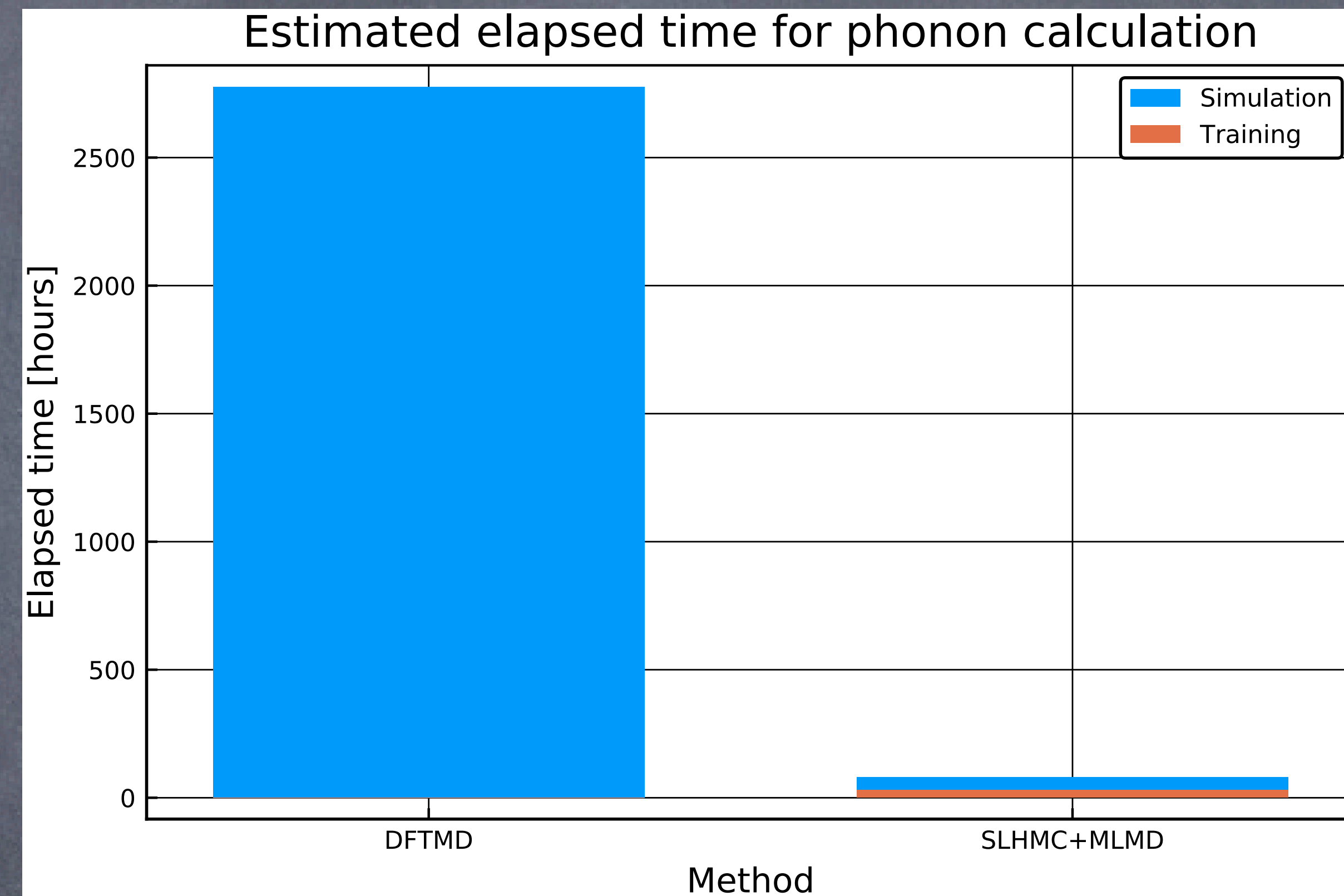
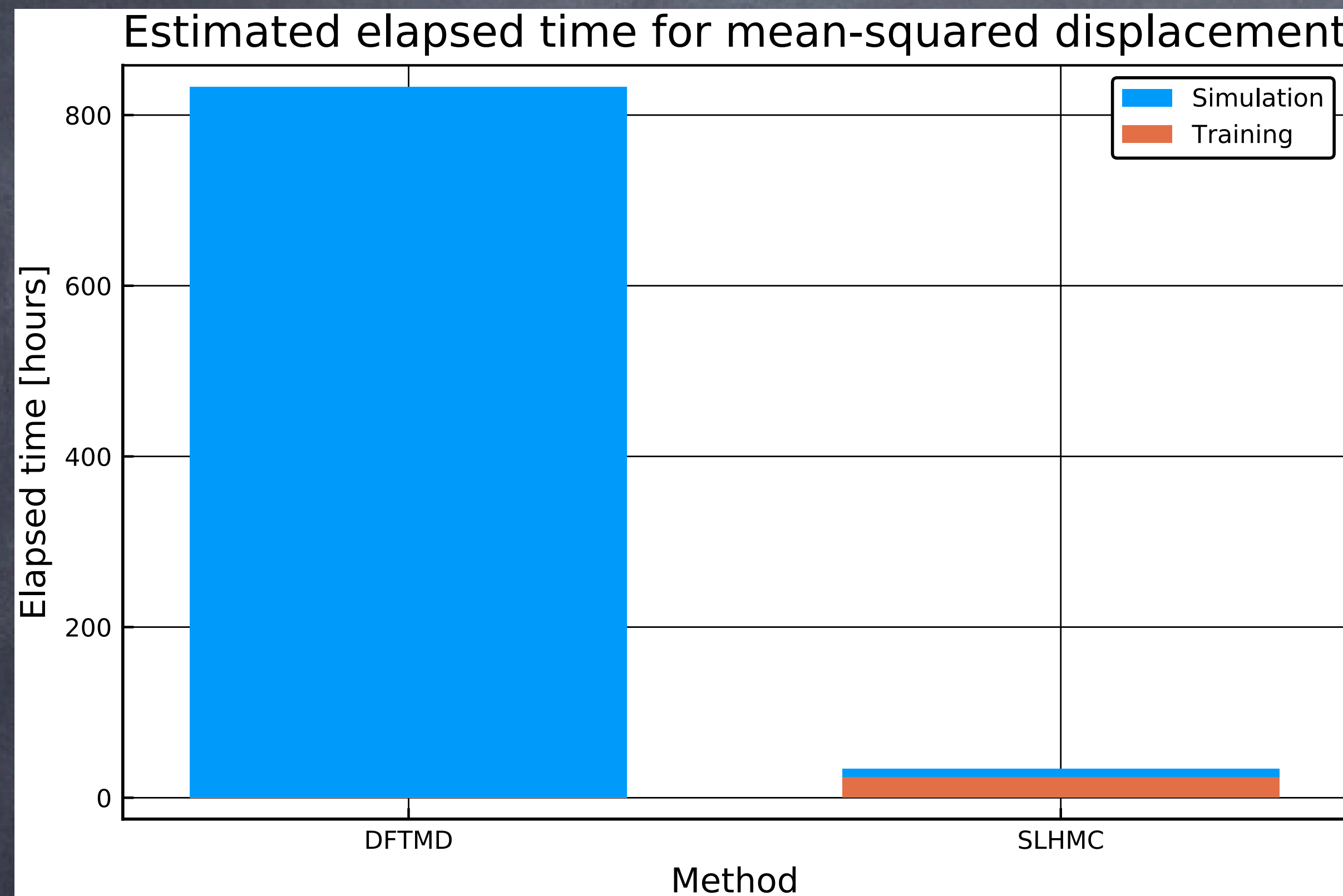
1fs 間隔で100,000 ステップで MD x 10

強い温度依存性を発見

実際の計算時間

SLHMC: SiO_2

SLHMC+MD: $\text{YNi}_2\text{B}_2\text{C}$



240CPUコア並列：JAEAスパコンを使用

SLHMCは訓練時間込みでも速い！

液体シリカ

NVTアンサンブル（粒子数、体積、温度一定）以外でも使いたい

NPTアンサンブル（粒子数、圧力、温度一定）の手法を開発

TABLE I. Acceptance ratio P_{ac} and mean MCMC interval dt_m of SLHMC for liquid silica with 72 and 216 atoms. N means the number of atoms. t_{eff} is the product of P_{ac} , dt_m , and the number of MCMC steps (20 000).

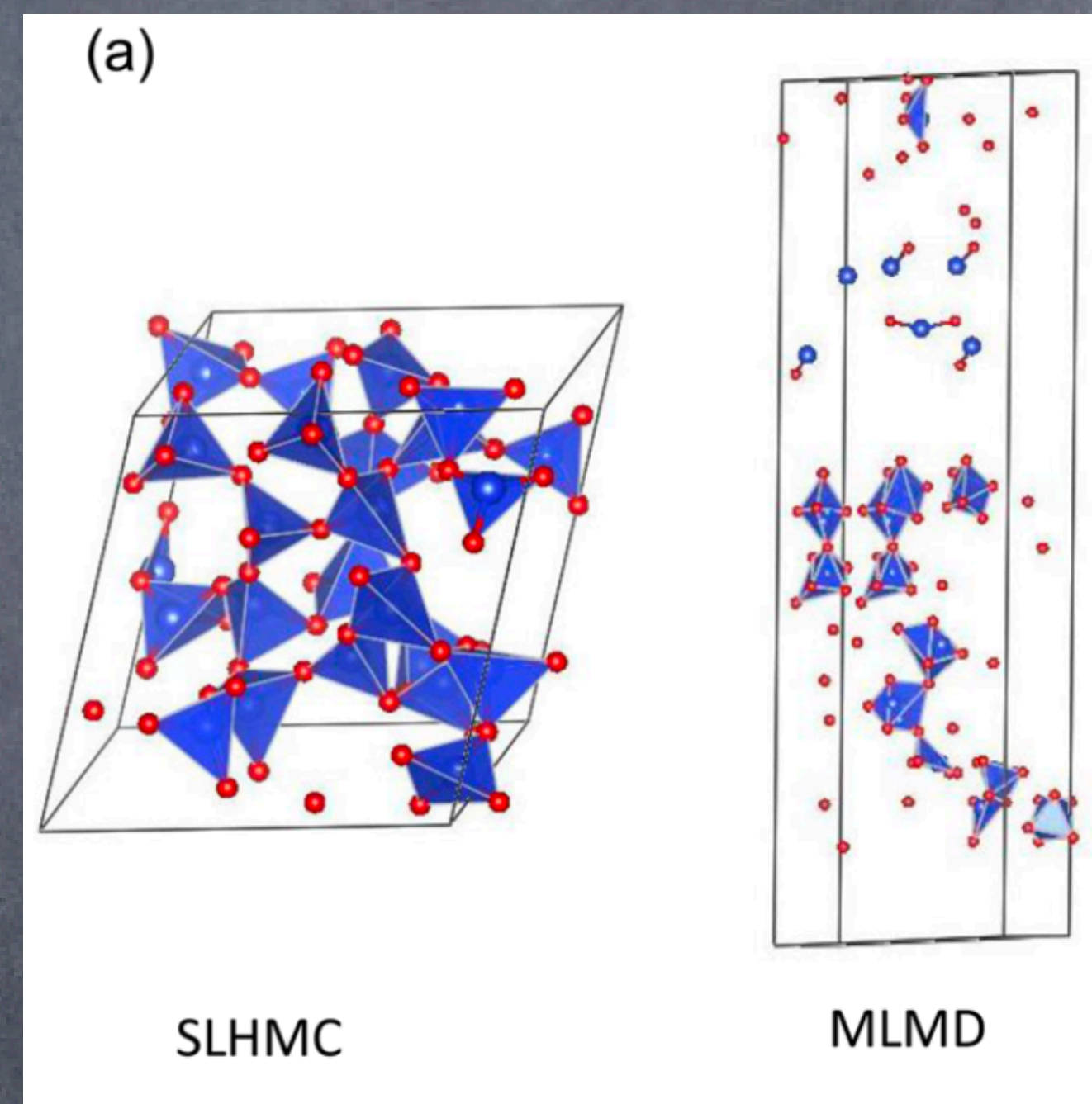
N	Temperature (K)	P_{ac} (%)	dt_m (fs)	t_{eff} (ps)
72	2500	34.1	242	1646
	3000	31.0	217	1343
	3500	33.8	235	1590
	2373	27.2	210	1142
216	3000	23.8	188	893
	3500	17.4	111	388

MD部分を可変にした

約200fsに一回だけ第一原理計算を行う

第一原理計算精度を保った1ns オーダーの
シミュレーションが可能に

72原子、4000Kで、50ps後の構造



MLMD

8600構造で学習
->結晶構造が壊れる

SLHMC

悪い構造はreject
しているため安定

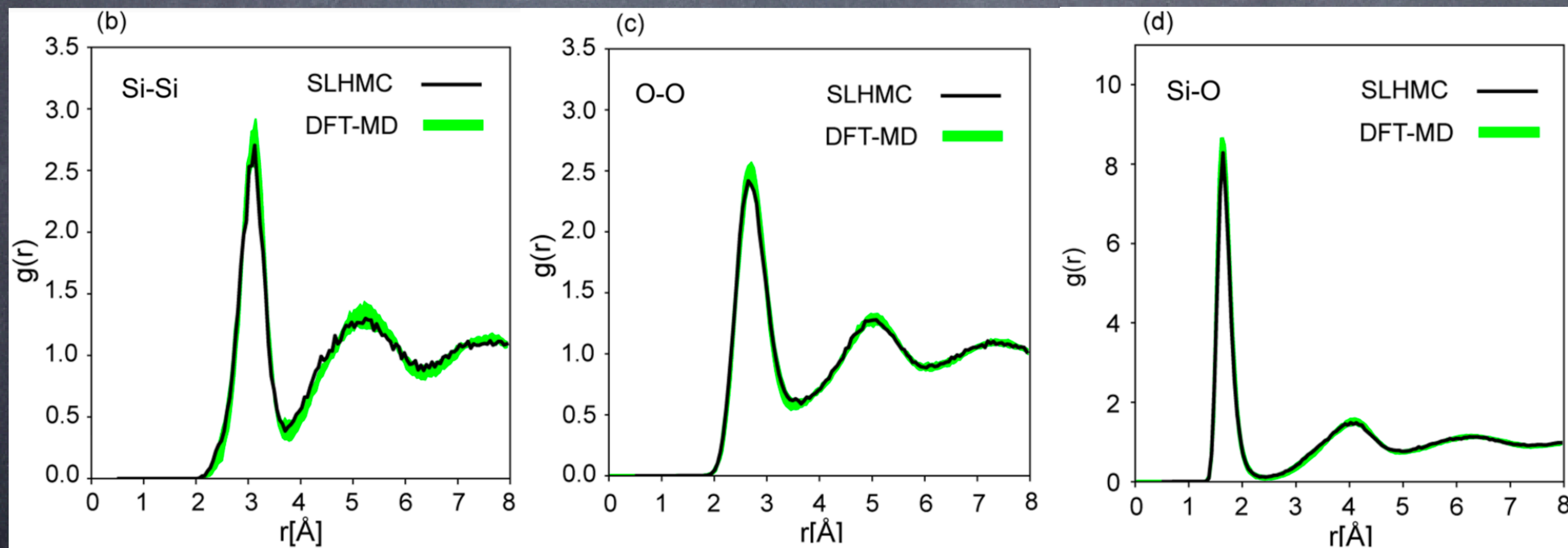
->安定なシミュレーションが可能

液体シリカ

NVTアンサンブル（粒子数、体積、温度一定）以外でも使いたい

NPTアンサンブル（粒子数、圧力、温度一定）の手法を開発

80 psの第一原理MDの結果との比較 (4000K)



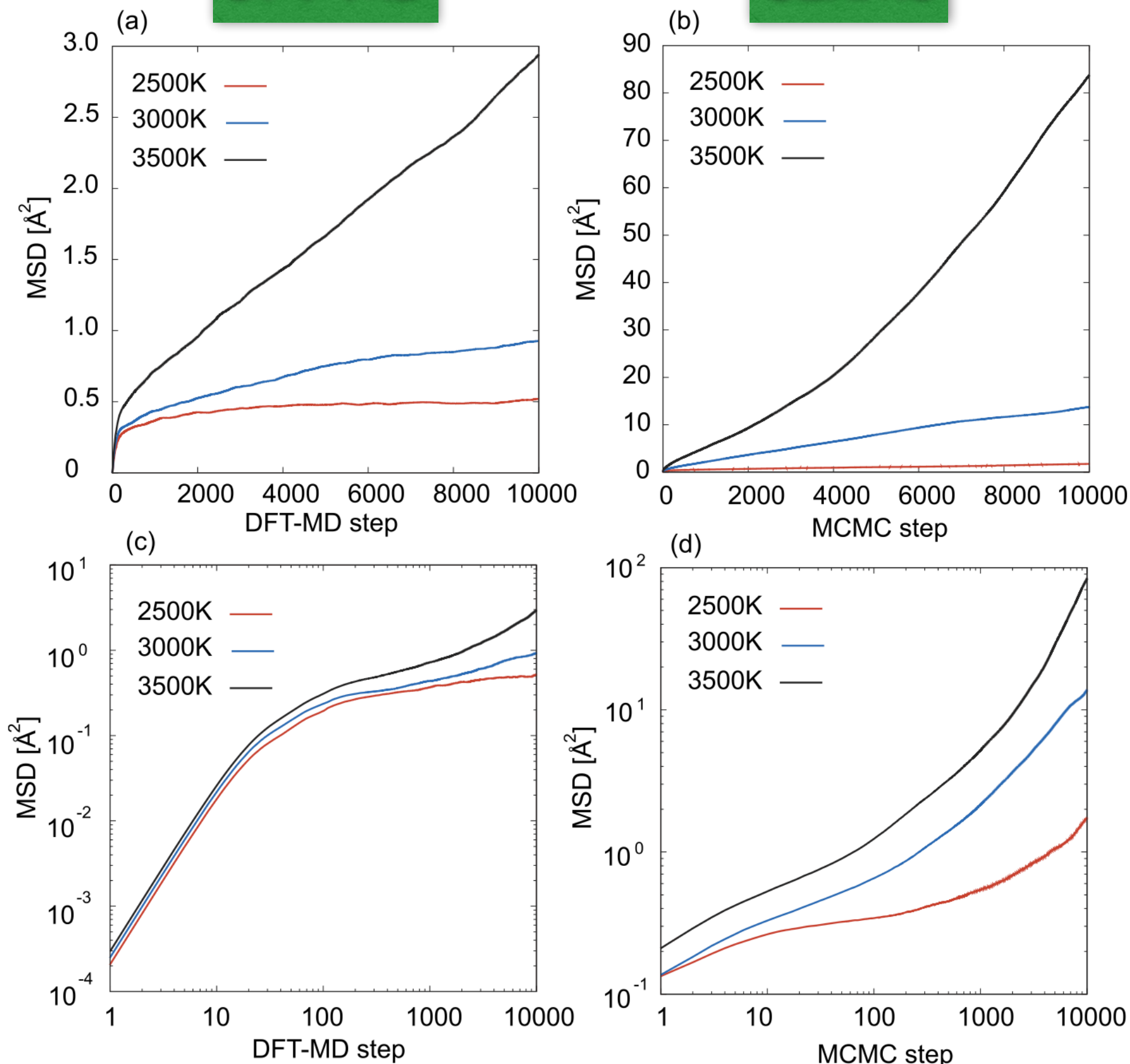
ポテンシャルが完全に一致していなくても、物理量は同じ精度

液体シリカ

NPTアンサンブル（粒子数、圧力、温度一定）の手法を開発

DFT-MD

SLHMC



平均二乗変位：三つの時間領域がある

ballistic, plateau and diffusive regimes

実験での融点は1983K

DFT-MDでは3500Kでもdiffusive regime
がよく見えない

**SLHMCだとMDを長くすることで実効的に
長時間平均が計算可能 -> diffusive
regimeが観測できる**

液体シリカ

NPTアンサンブル（粒子数、圧力、温度一定）の手法を開発

平均二乗変位：三つの時間領域がある

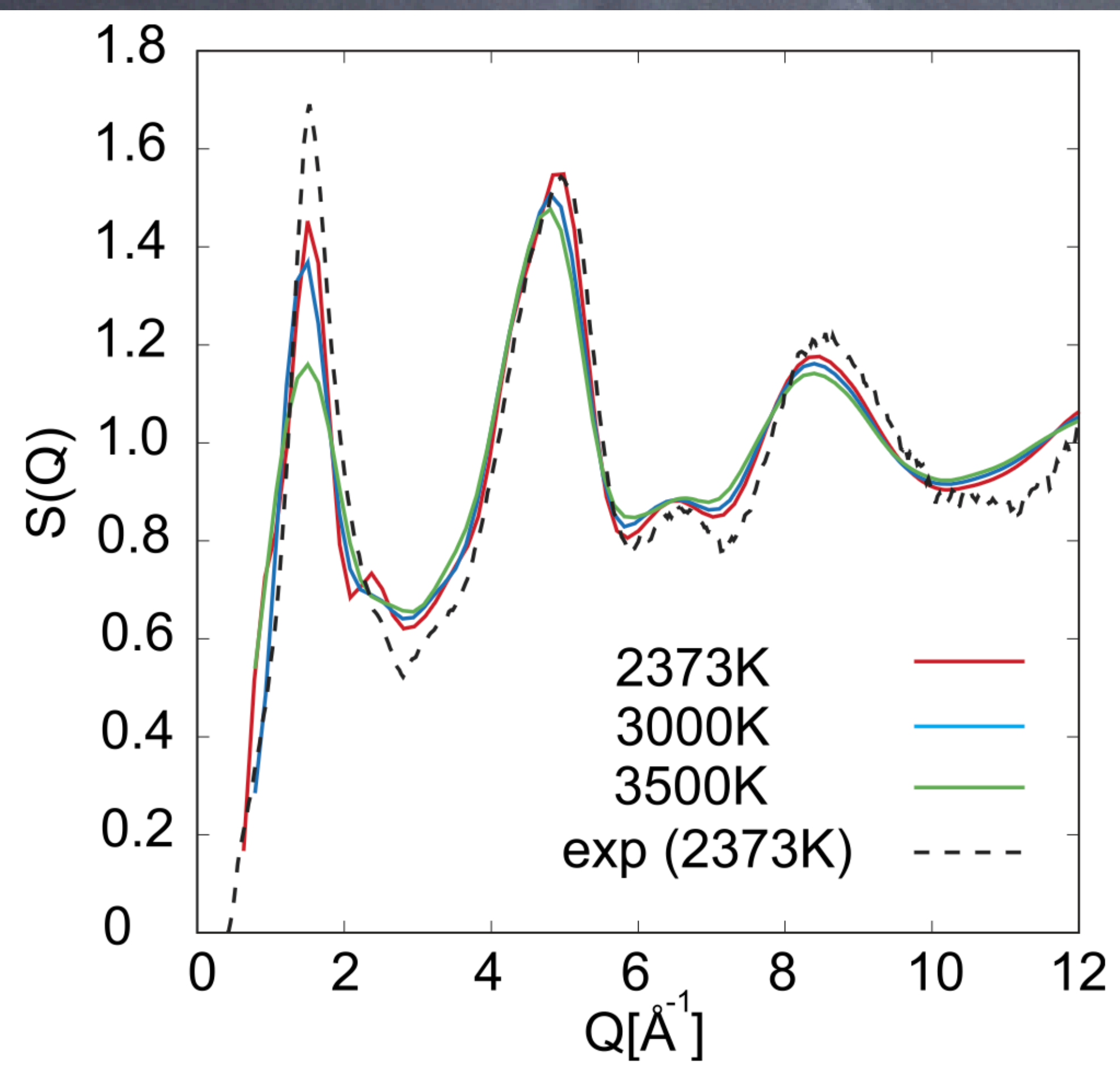
ballistic, plateau and diffusive regimes

実験での融点は1983K

DFT-MDでは3500Kでもdiffusive regime
がよく見えない

**SLHMCだとMDを長くすることで実行的に
長時間平均が計算可能 -> diffusive
regimeが観測できる**

-> 実験値をよく再現する結果が得られる

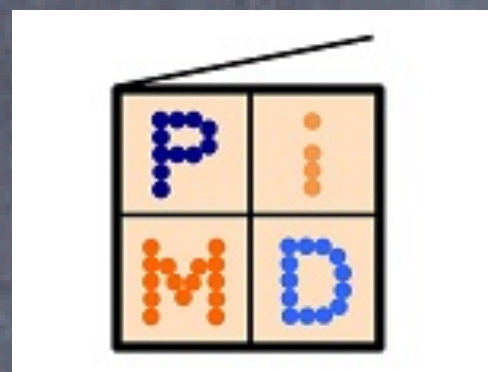


x線による構造因子

まとめ

まとめ

精度保証された機械学習シミュレーション手法：自己学習ハイブリッドモンテカルロ法の紹介
 機械学習ポテンシャルの精度に依らずに第一原理計算精度を保証する計算手法



PIMD：CCSEが開発している原子分子シミュレーション
 パッケージ(メイン開発者：志賀基之氏)

オープンソースソフトウェア

機械学習をシミュレーションに使う不安

模倣なので結果が正しいか不安 → 自己学習ハイブリッドモンテカルロ(SLHMC)法

YN et al., Phys. Rev. B 102, 041124(R) (2020)

SLHMCを使うと嬉しい時：

1. DFTレベルの精度の計算をしたいがDFT-MDの計算コストが大きすぎて断念
 してしまっている時
2. 機械学習MDのための高精度なニューラルネットワークを構築したい時