機械学習と理論物理

Akio Tomiya (RIKEN-BNL)

オーガナイザーの皆様
ありがとうございます。

自己紹介
Who and what am I?

Akio Tomiya

兵庫県立大(物性) →PhD in 2015 (阪大、素粒子論)

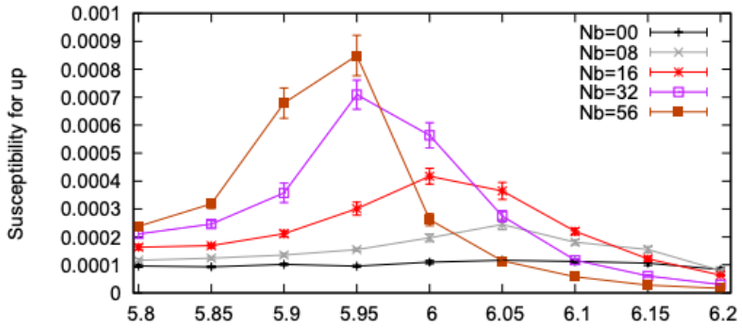
→華中師範大学（武漢、ポスドク)

→理化学研究所/ブルックヘブン国立研究所(NY、ポスドク) 今もアメリカ。

研究分野

1. QCD熱力学 (2008.00493, …)

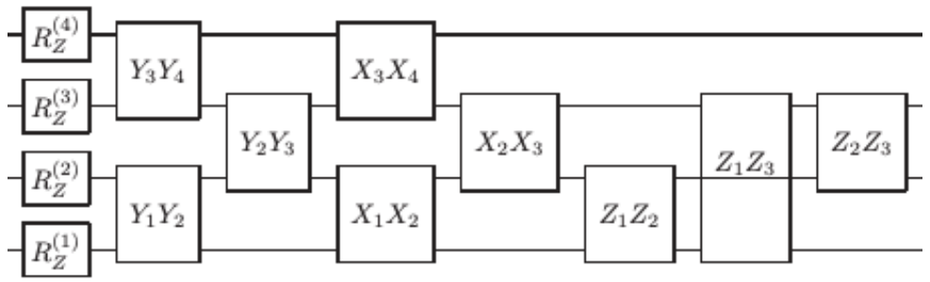
U(1)A とか磁場とか。



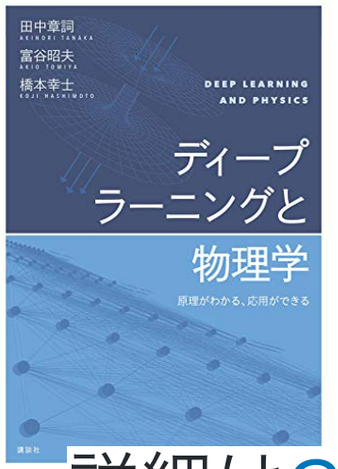
(c.f. 深谷さんのトーク)

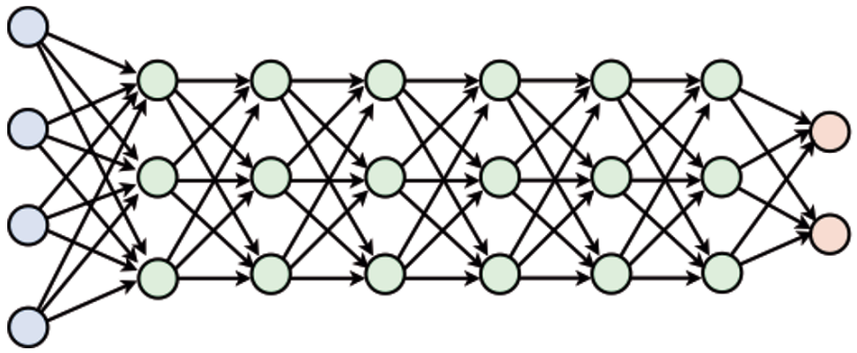
2. 量子コンピュータ (2001.00485)

シュウィンガー模型、再現しました



3. 機械学習 (本日)





詳細はGoogle scholarへ

話すこと

1. 機械学習とは何なのか
2. 物理への応用
3. 2020年代にあたって

※ 何をどう話すか迷いましたが、研究分野の紹介をすることにしました
(ほとんどレビューです)

Machine learning?

Machine learning?

最新情報

- Physics ∩ ML: http://www.physicsmeetsml.org/
  - アメリカ東海岸中心, 英語
  - 日本時間深夜

- DLAP2020 https://cometscome.github.io/DLAP2020/
  - 日本, 日本語
  - 隔週の木曜朝
  - 過去の研究会ページ(2017~)もあり。

どちらでも過去のスライドが見れます。

Machine learning?

教科書、色々ありますが...

理論系

- これならわかる機械学習入門 瀧雅人(標準的)
- ディープラーニングと物理学、田中 橋本 富谷(物理屋フレンドリー)
- 入門パターン認識と機械学習 後藤 小林 (ニューラルネット以外、色々載ってる)
- 数学的: ベイズ統計の理論と手法、渡辺澄夫、統計的学習理論、金森敬文
- パターン認識と機械学習(場の理論の教科書で言うItzykson-Zuberっぽい印象)

実装系

- PythonとKerasによるディープラーニング、F. Chollet et al(実装)
- Pythonユーザのための Jupyter [実戦] 入門、池内et al (Jupyterの使いかた)
- Pythonで機械学習入門、大関 真之 (物理屋が書いたGAN の実装)

Machine learning?

Data (empirical knowledge) determine output

計算機学者のサミュエル による機械学習の古い “定義”

「明示的なプログラムをすることなく、機械が学習できる能力を持つようにする」

(1959)

「経験だけから、その背後の構造を抽出し未知の
状況にも適用できる能力を機械に自動的に獲得させる」

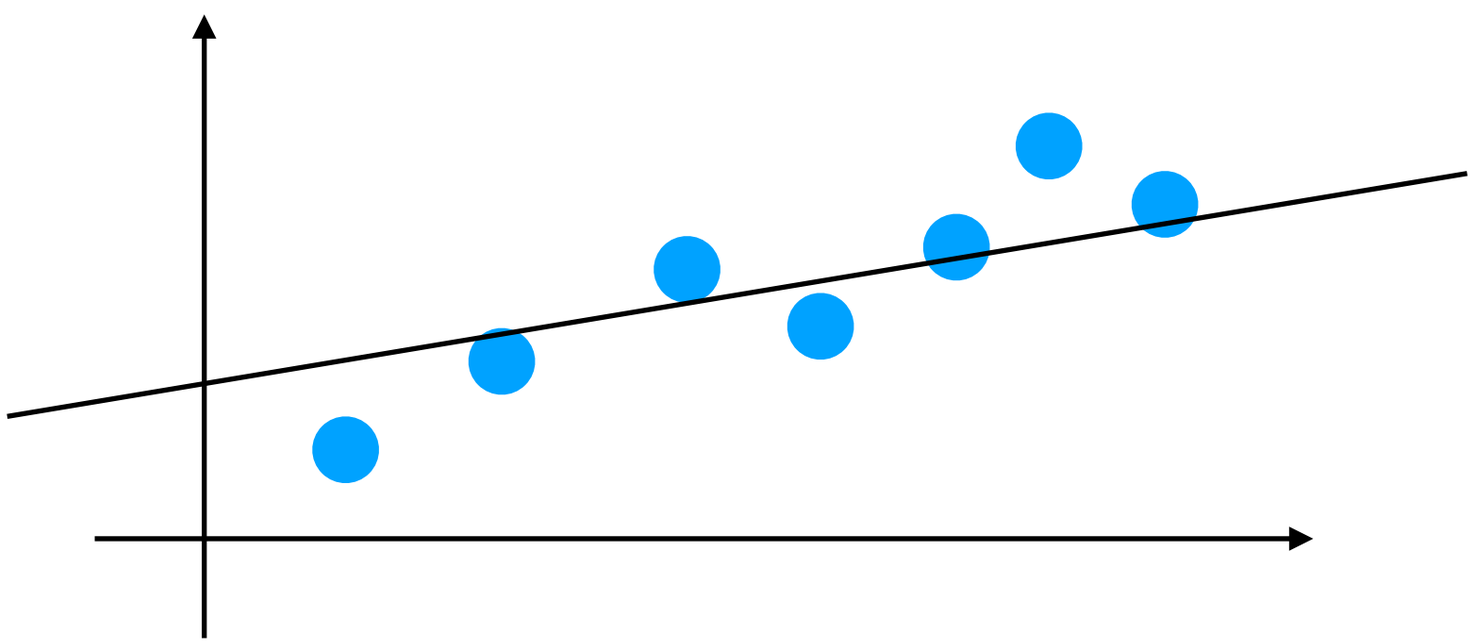
「フィットでは？」

Machine learning?

Akio Tomiya

Data (empirical knowledge) determine output

線形回帰



x: 入力データ

y: 教師ラベル

フィット関数 ~ 機械学習のアーキテクチャ

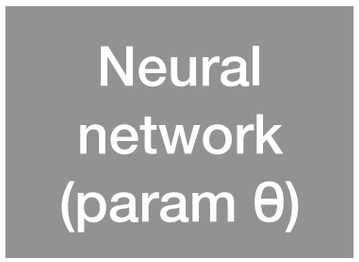
c.f. PPP 2019 の佐藤さんのスライド

Machine learning?

Akio Tomiya

Data (empirical knowledge) determine output

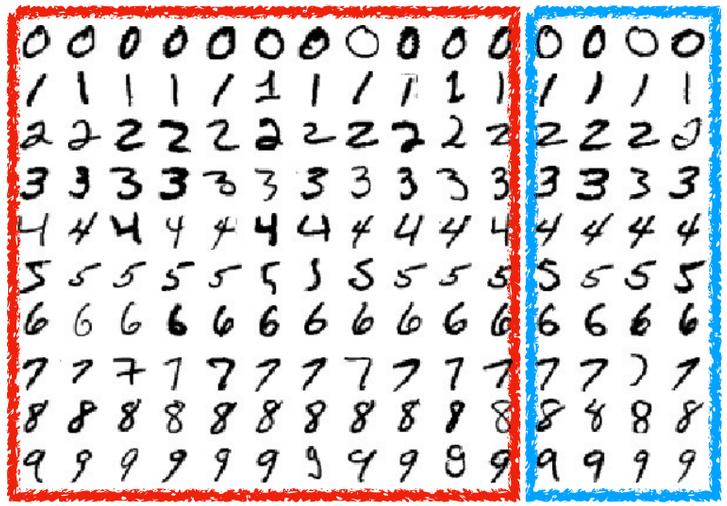
E.g. Supervised learning using neural networks



Machine learning?

Data (empirical knowledge) determine output

E.g. Supervised learning using neural networks



Training data

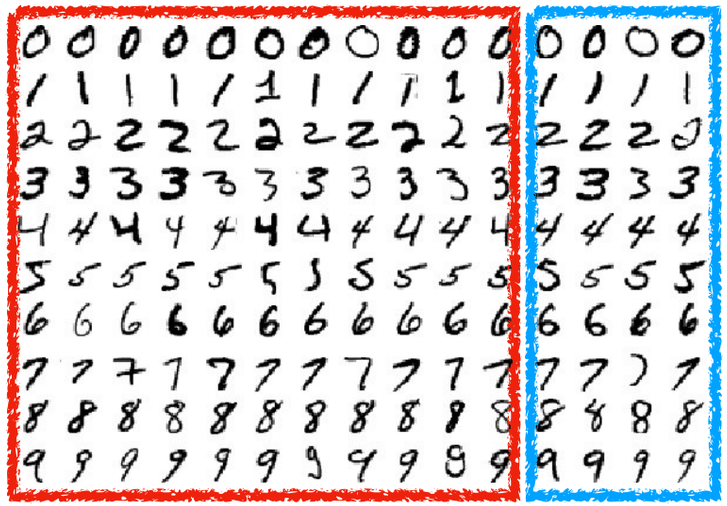
Neural network (param θ)

Machine learning?

Data (empirical knowledge) determine output

E.g. Supervised learning using neural networks

Sample



Training data

Feed

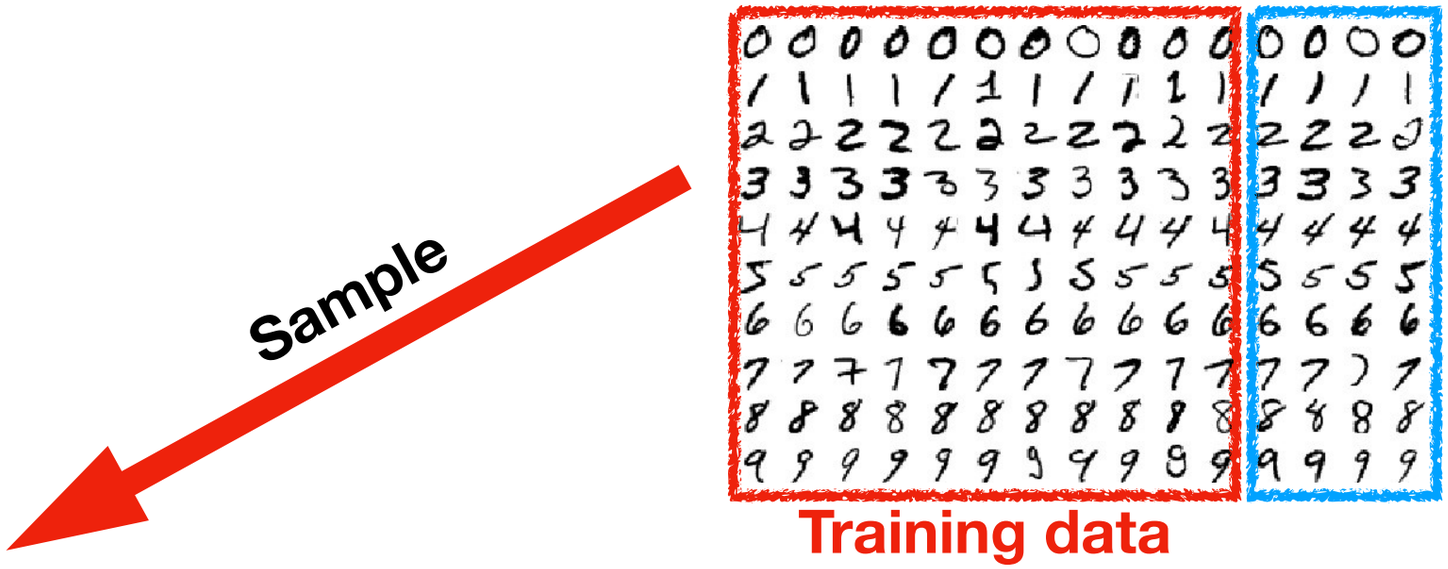
Neural network (param θ)

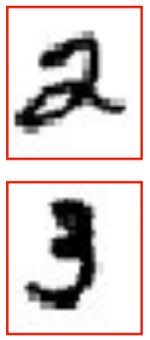
Machine learning?

Akio Tomiya

Data (empirical knowledge) determine output

E.g. Supervised learning using neural networks





Feed

Neural network (param θ)

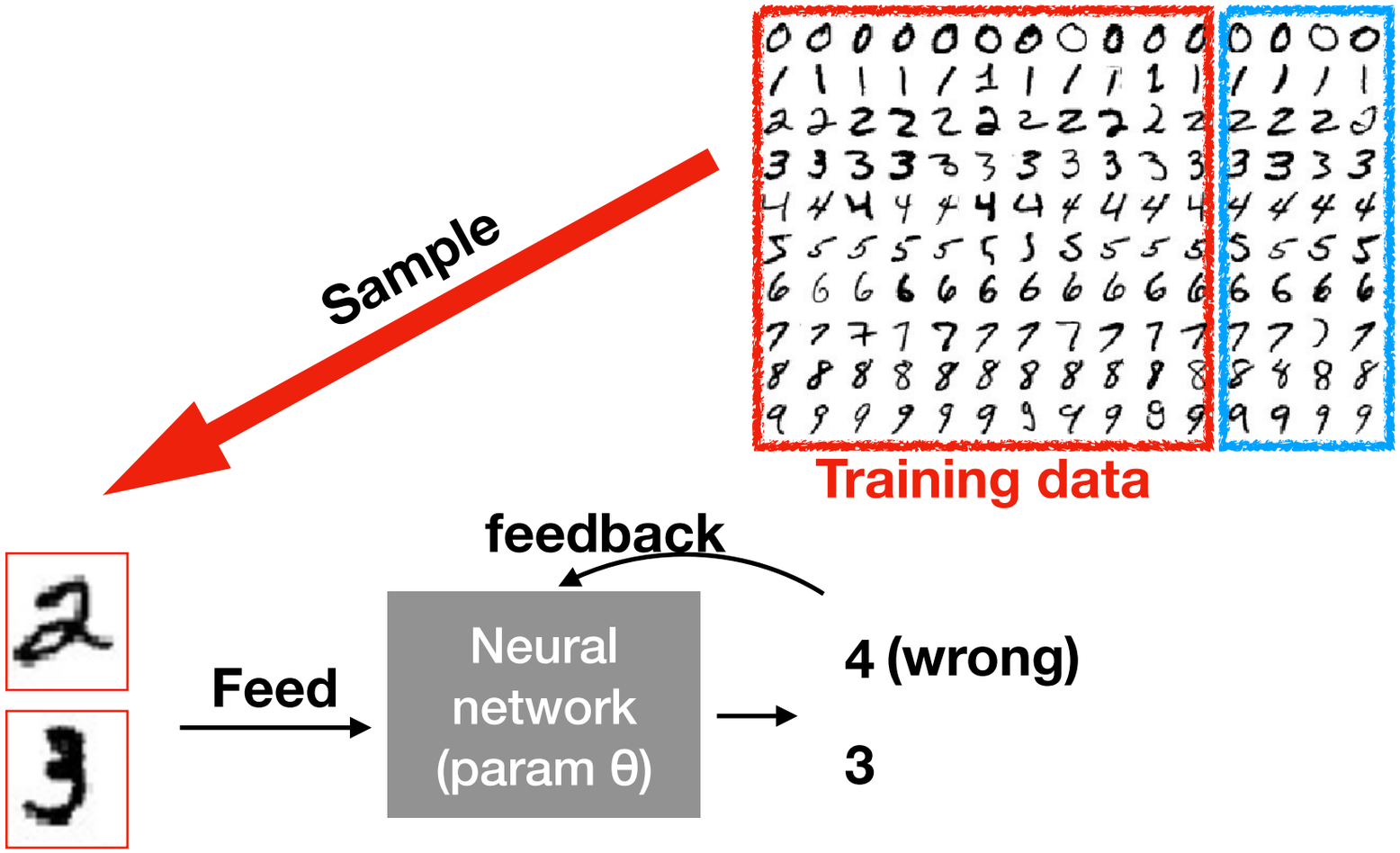
4 (wrong)

3

Machine learning?

Data (empirical knowledge) determine output

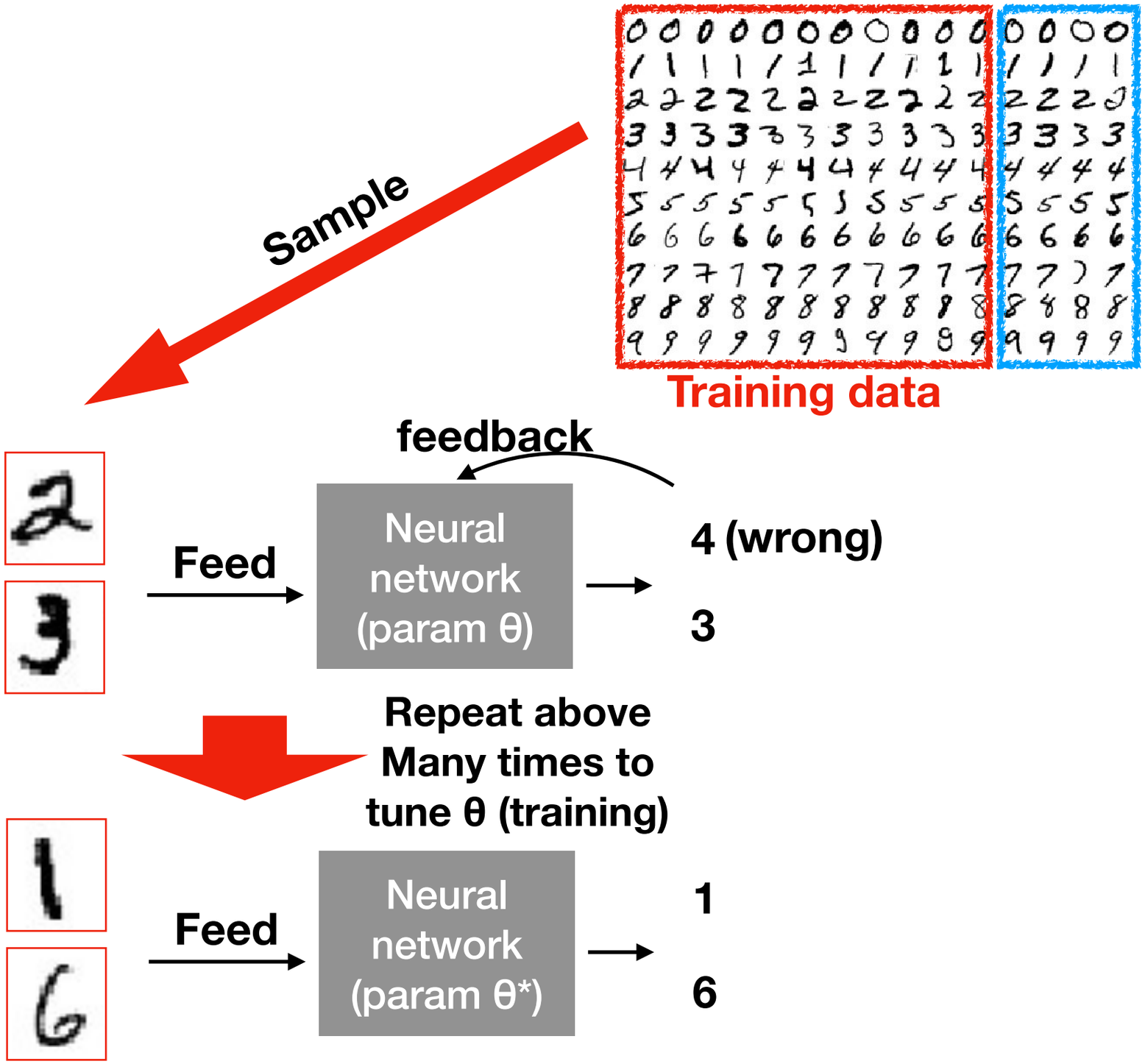
E.g. Supervised learning using neural networks



Machine learning?

Data (empirical knowledge) determine output

E.g. Supervised learning using neural networks

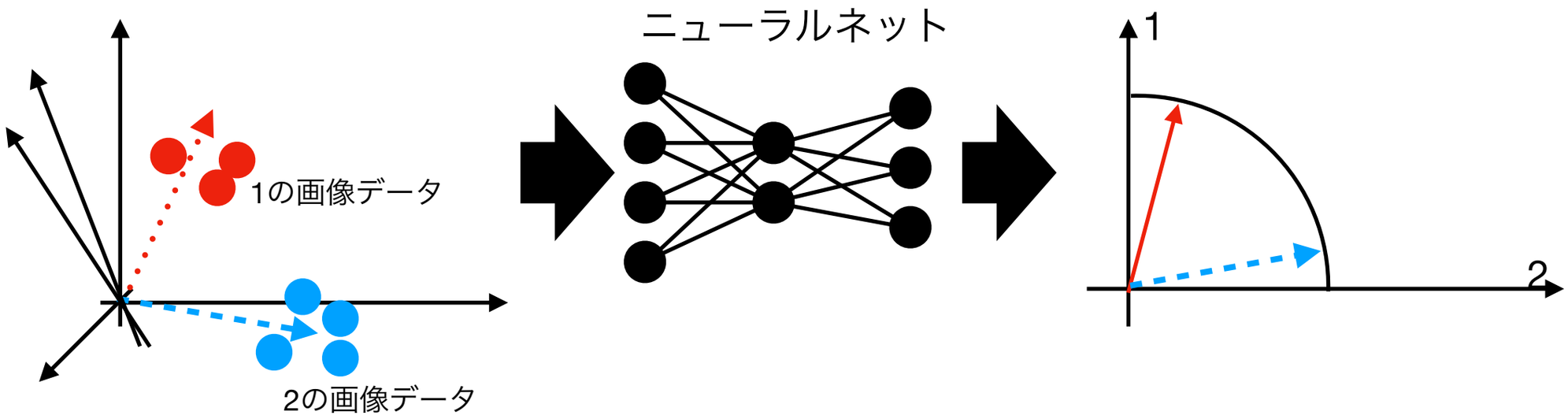


Machine learning?

Data (empirical knowledge) determine output

$$= \begin{pmatrix} 0.624 \\ 0.21 \\ 0.3434 \\ 0.756 \\ 0.3456 \\ 0.64 \\ 0.251 \\ 0.11 \\ 0.23 \\ \vdots \end{pmatrix}$$

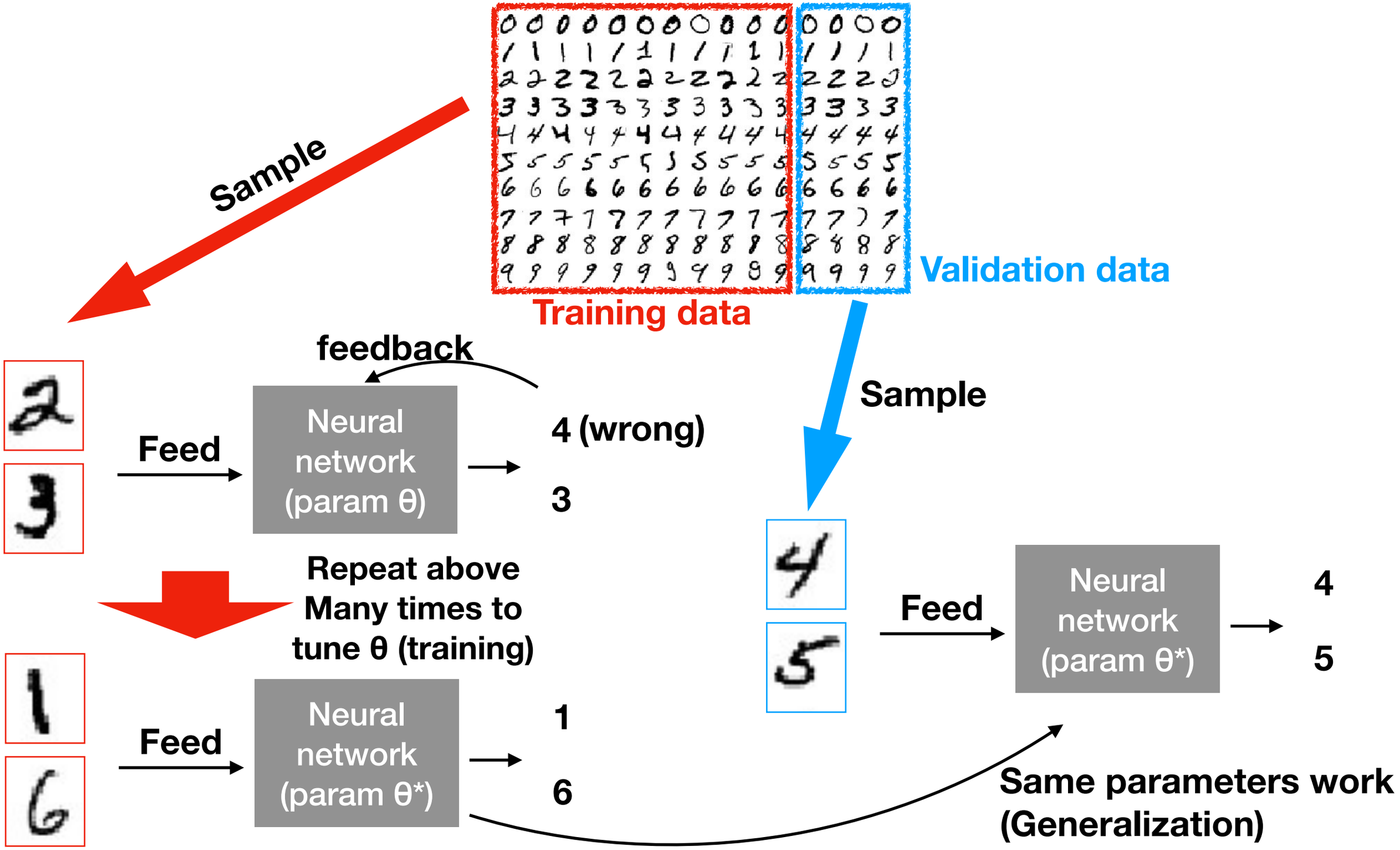
画像はベクトル



Machine learning?

Data (empirical knowledge) determine output

E.g. Supervised learning using neural networks



Category of machine learning

Akio Tomiya

Unsupervised learning is one of class of machine learning

- 教師あり学習
    - 画像認識など、教師ラベルありのもの。
- 教師なし学習
    - 教師ラベルなしでの分類。データの分布を理解する。
- 強化学習
    - e.g. Alpha GOなど

物理への応用
Too many to introduce…

格子QCDへの応用に話を限ります

1. 格子QCD
    1. Configuration generation in LQCD
    2. Reduction of cost in measurements
    3. Application to QCD thermodynamics
2. Phenomenology
3. String, AdS/CFT
4. Condensed matter

[ディープラーニングと物理学2020]のホームページなどを見てください

物理量の決定にまつわる話

物理量の決定にまつわる話

Akio Tomiya

3点関数の決定と輸送係数の決定

格子QCDでは、ゲージ配位を作ったあと、測定をする必要がある

たまに測定で問題：

1. 計算コストが大変
2. 格子上でどうやって計算していいかわからない

1. の例として3点関数
2. の例として輸送係数

があり、それぞれ機械学習を使った解決法が提案されている

3点関数の予言

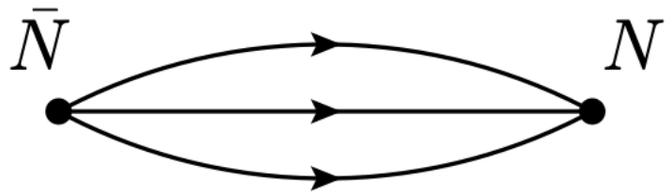
核子の行列要素を計算したいがコストが...

Akio Tomiya

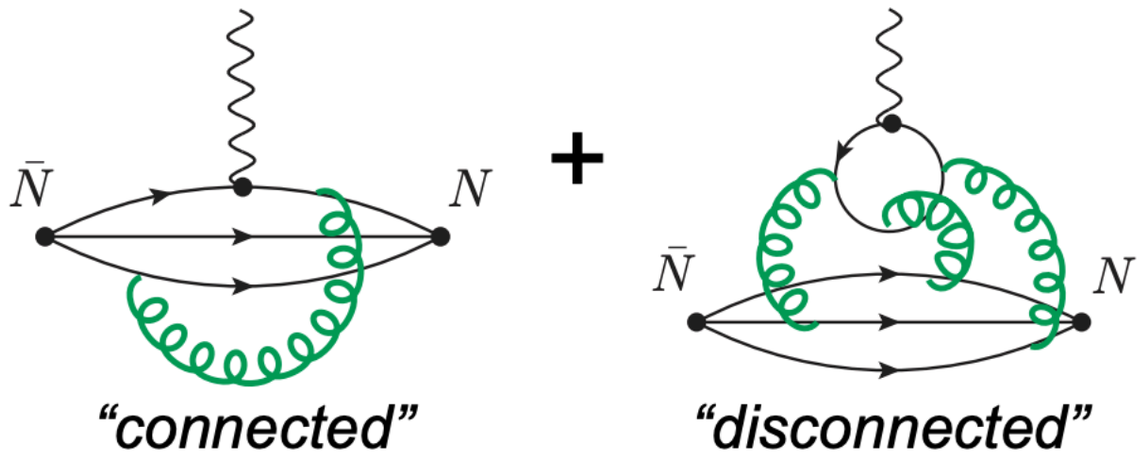
B Yoon et al 1807.05971, 1909.10990

$$R_{\mathcal{O}}(T,\tau;P,P') = \frac{\langle N(T)\mathcal{O}(\tau)\bar{N}(0)\rangle}{\langle N(T)\bar{N}(0)} \xrightarrow[T,\tau,(T-\tau)\to\infty]{} \langle P'|\mathcal{O}|P\rangle$$

$$C_{2\mathrm{pt}}(T) = \langle N(T)\bar{N}(0)\rangle =$$



$$C^{\mathcal{O}}_{3\mathrm{pt}}(T) = \langle N(T)\mathcal{O}(\tau)\bar{N}(0)\rangle =$$



同じ配位から2点関数と3点関数を計算すると相関がある

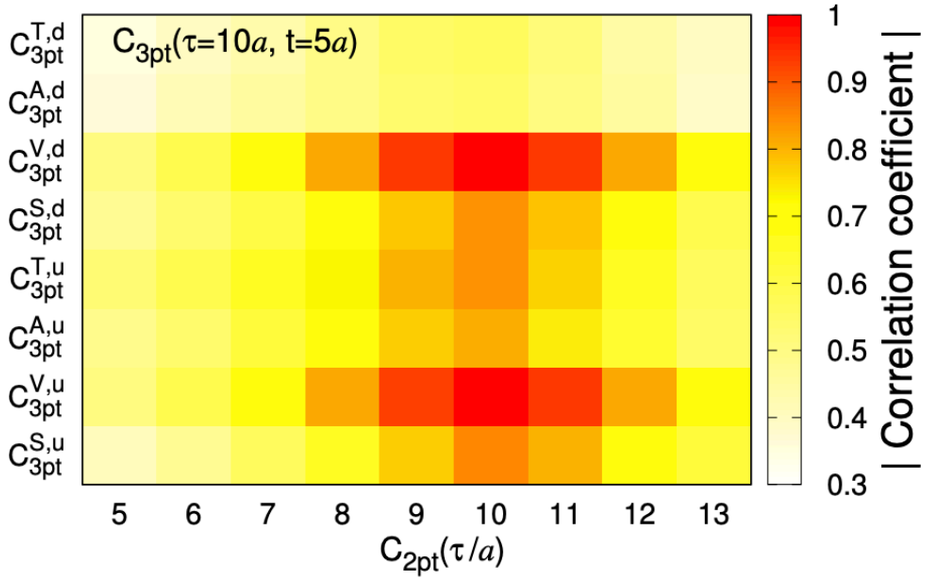
→なんか予言できる？

http://www.int.washington.edu/talks/WorkShops/int_14_57W/People/Syritsyn_S/Syritsyn.pdf

3点関数の予言

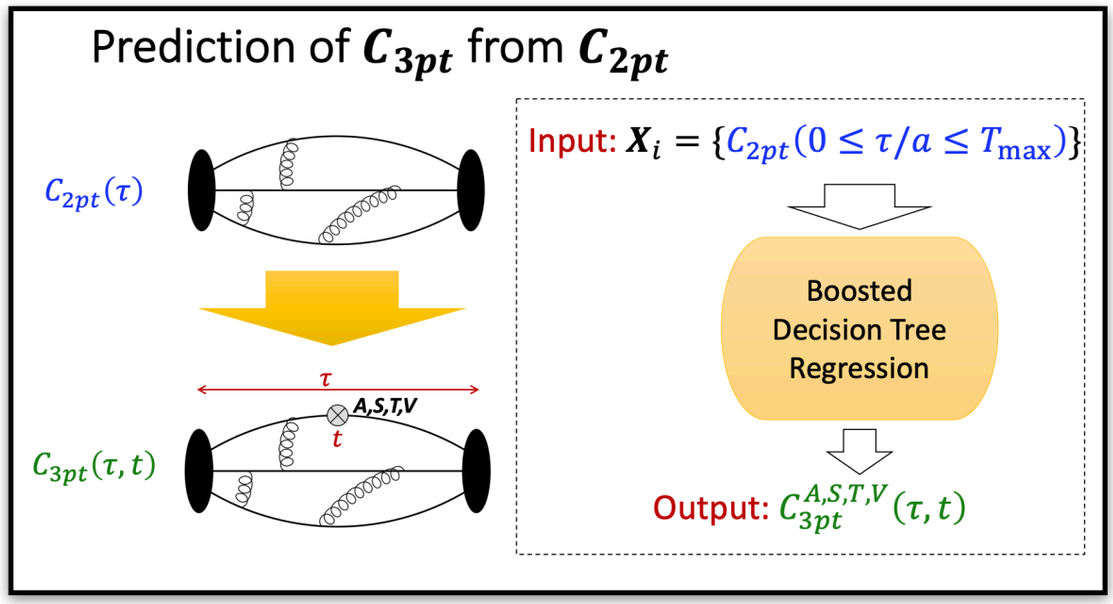
行列要素を計算したいがコストが...

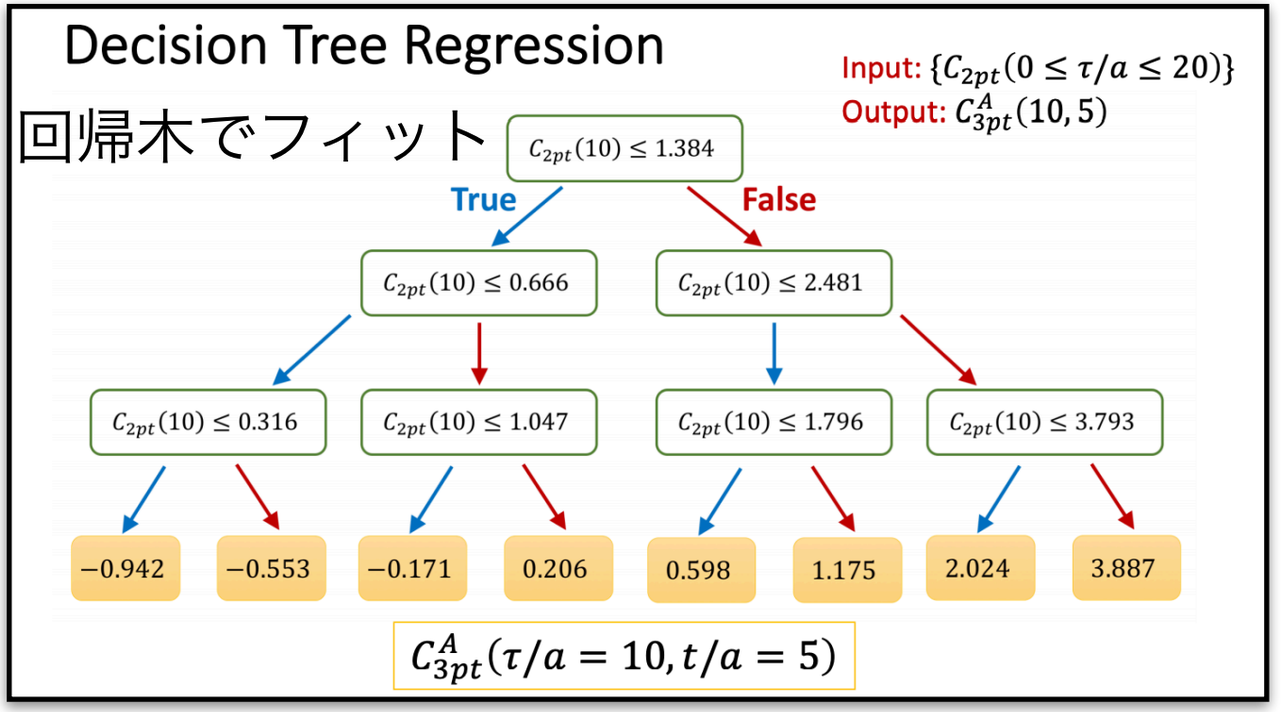
Akio Tomiya



Conf#	3pt @ t=5, tau=10	2pt(tau=10)
1	-0.8	0.1
2	0.2	1.2
3	2.0	3.2

B Yoon et al 1807.05971, 1909.10990





From Boram's slides in Lattice 2018

3点関数の予言

行列要素を計算したいがコストが...

とは言え、系統誤差が気になる

よく知られたAll mode averaging (AMA) と似た感じで補正できる

$$\overline{O} = \frac{1}{M-N}\sum_{i\in\{UD\}} O_i^{\mathrm{P}} + \frac{1}{N_b}\sum_{i\in\{BC\}} (O_i - O_i^{\mathrm{P}}),$$

第一項: 機械学習の予言

第二項: 補正項

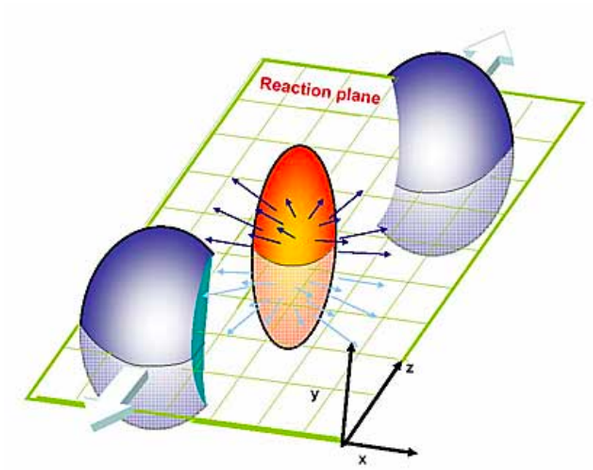
	直接測定	$\mathcal{VP}2(\tau\to\infty)$ $[C_{2\mathrm{pt}}, C_{3\mathrm{pt}}(12)]$	Cost(%)
g_S	0.989(18)	0.981(20)	80
g_A	1.2303(51)	1.2304(61)	93
g_T	1.0311(51)	1.0326(54)	74
g_V	1.0443(19)	1.0440(21)	78

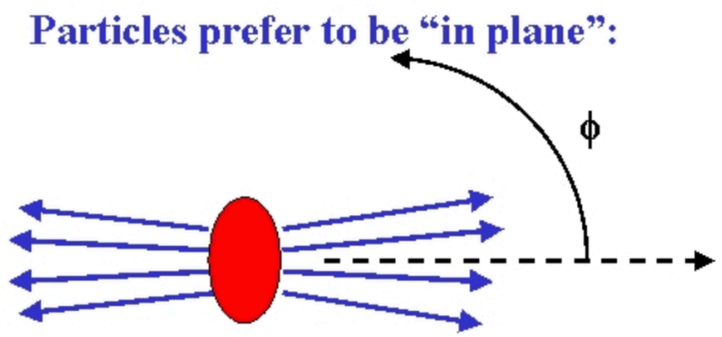
うまく行ってそう

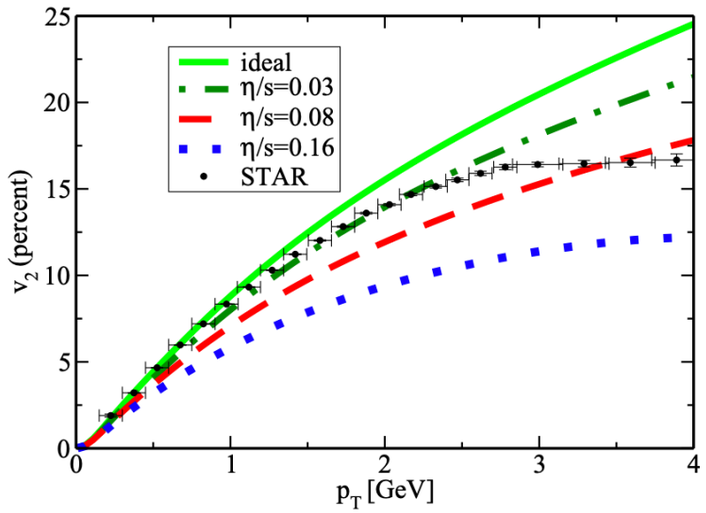
輸送係数の決定(1/3)

Akio Tomiya

格子QCDでスペクトル関数ρ(ω)を決めるのは難しい







eta/s < 0.1?

s: エントロピー密度、これは格子QCDで計算できていた

eta: ずり粘性。むずかしい。

$$\eta(T) = \pi \frac{d\rho(\omega)}{d\omega}\bigg|_{\omega=0}$$

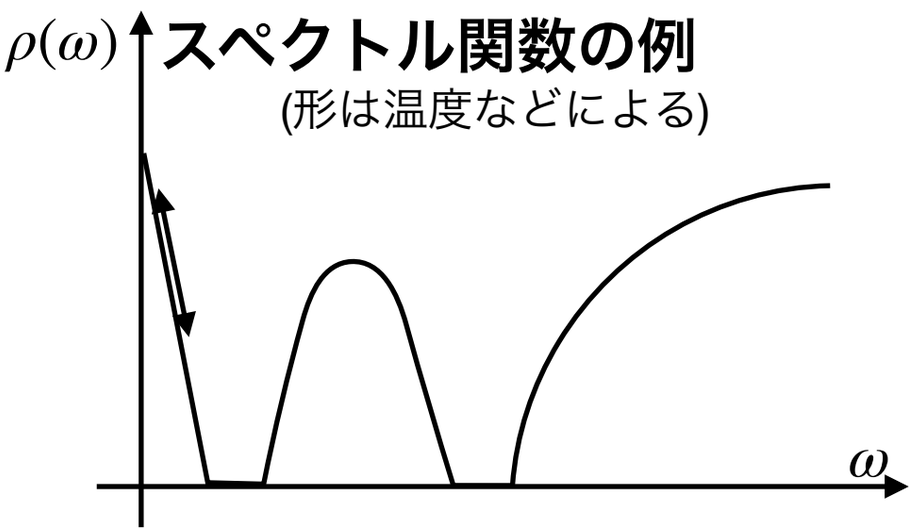
K~tanh

↓未知

$$\frac{1}{T^5}\int d\vec{x}\left\langle T^R_{12}(0)T^R_{12}(x)\right\rangle = \int_{-\infty}^{+\infty} d\omega K(\tau,\omega)\rho(\omega)$$

格子で測れる、x=O(10)点　　**ω=O(1000)点**

ρを決めるのに情報が足りない (ill-posed)



[https://www.bnl.gov/rhic/news/061907/story2.asp, https://www.bnl.gov/phobos/Presentations/paris01/sld025.htm, 0706.1522]

輸送係数の決定(2/3)
スパースモデリング法

Akio Tomiya

E. Itou, Y. Nagai
2004.02426

$$\frac{1}{T^5}\int d\vec{x}\left\langle T^R_{12}(0)T^R_{12}(x)\right\rangle = \int_{-\infty}^{+\infty} d\omega K(\tau,\omega)\rho(\omega)$$

スケマティックには

$$\begin{pmatrix}\langle TT\rangle_1\\ \langle TT\rangle_2\\ \langle TT\rangle_3\end{pmatrix} = \begin{pmatrix}* & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & *\end{pmatrix}\begin{pmatrix}\rho_1\\ \rho_2\\ \rho_3\\ \rho_4\\ \rho_5\end{pmatrix}$$

Kの対角基底

$$\begin{pmatrix}\langle TT\rangle'_1\\ \langle TT\rangle'_2\\ \langle TT\rangle'_3\end{pmatrix} = \begin{pmatrix}s_1 & 0 & 0 & 0 & 0\\ 0 & s_2 & 0 & 0 & 0\\ 0 & 0 & s_3 & 0 & 0\end{pmatrix}\begin{pmatrix}\rho'_1\\ \rho'_2\\ \rho'_3\\ \rho'_4\\ \rho'_5\end{pmatrix}$$

データによらない。

「ρ’ の内、データから決まるのは一部で決まらないのは0」という
拘束条件をいれたら解ける。
Kの対角基底 + 拘束条件 + フィット = スパースモデリング法

輸送係数の決定(3/3)
スパースモデリング法

Akio Tomiya

E. Itou, Y. Nagai
2004.02426

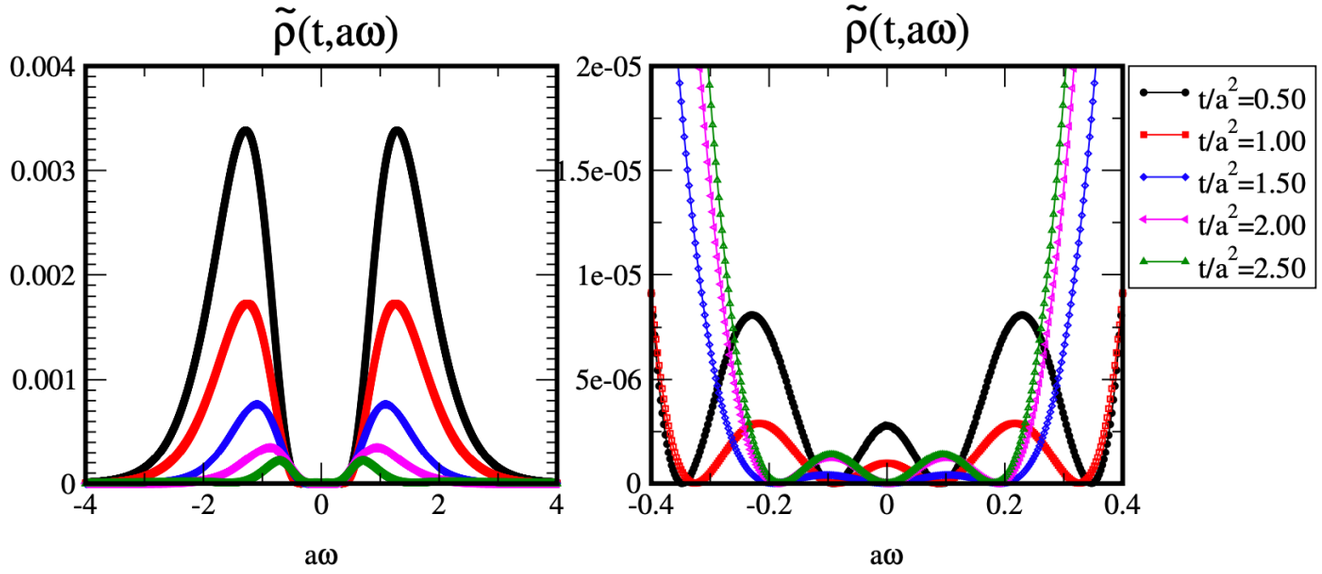
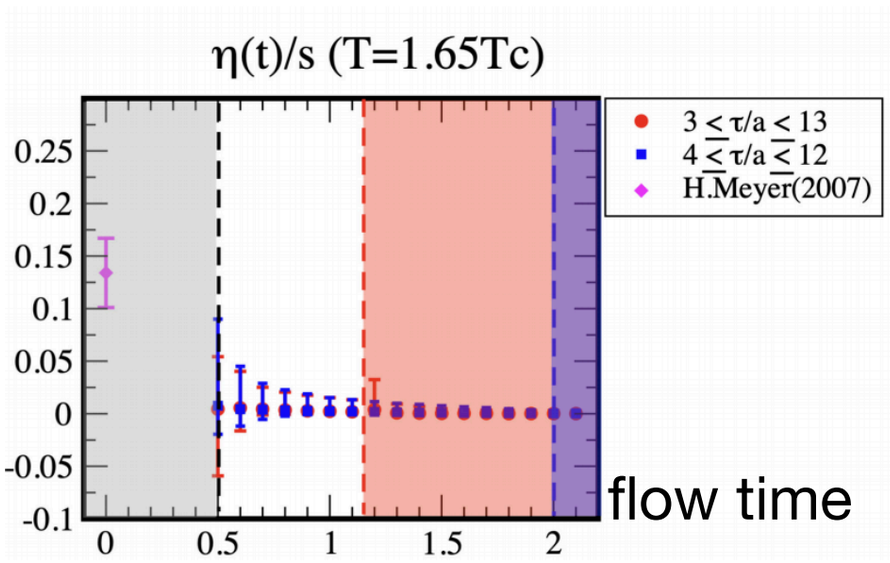


Figure 6. The obtained spectral function $\tilde{\rho}(t, a\omega)$ as a function of $a\omega$. The right panel is an enlarged plot in $\omega \approx 0$ regime.

L = 64^3 x 16, Plaquette gauge, beta = 6.3

s/T^3 = 4.98(24)



Even for small statistics, it gives consistent results with small error

Summary for detection

Akio Tomiya

3点の予言と輸送係数

- 数値計算のコスト削減の１種として機械学習がつかえる。
  核子のチャージの計算は良さそう。

- スパースモデリング (& flow) で誤差少なく、輸送係数の計算

- スパースモデリングは一般的な手法で拘束条件も色々いれれるのでできることが多そう。

格子ゲージ理論の配位生成

ゲージ場配位の生成
サイコロを振って経路積分する

Akio Tomiya

- **格子QCDの経路積分は有限次元積分**

$$\langle O[\phi] \rangle = \frac{1}{Z} \int \underline{\mathscr{D}\phi e^{-S[\phi]}}\, O[\phi]$$

$$\mathscr{D}\phi = \prod_{i \in \{\mathbb{Z}/L\}^4} d\phi_i$$: ~ 10000 次元、台形法など絶望的

表 5.1　誤差と次元と計算時間の関係

次元 s	誤差 10^{-1}	10^{-2}	10^{-3}	10^{-4}
4	10^2	10^4	10^6	10^8
20	10^{10}	10^{20}	10^{30}	10^{40}
100	10^{50}	10^{100}	10^{150}	10^{200}
500	10^{250}	10^{500}	10^{750}	10^{1000}

(確率的シミュレーションの基礎、手塚 集)

ゲージ場配位の生成
サイコロを振って経路積分する

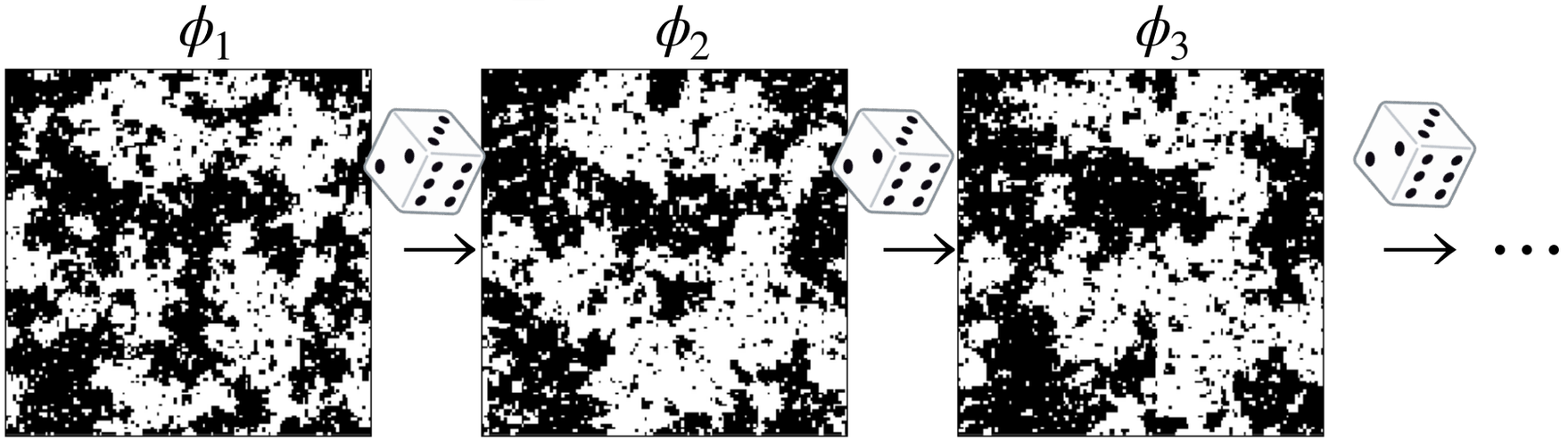
Akio Tomiya

- **格子QCDの経路積分は有限次元積分**

$$\langle O[\phi] \rangle = \frac{1}{Z} \int \mathscr{D}\phi e^{-S[\phi]} O[\phi]$$

$$\mathscr{D}\phi = \prod_{i \in \{\mathbb{Z}/L\}^4} d\phi_i$$: ~ 10000 次元、台形法など絶望的

代わりに $P[\phi] = \frac{1}{Z} e^{-S[\phi]}$ **でマルコフ連鎖モンテカルロ法！**



配位の列(アンサンブル)をつくり、統計平均で期待値を計算する

ゲージ場配位の生成
サイコロを振って経路積分する

Akio Tomiya

- **格子QCDの経路積分は有限次元積分**

$$\langle O[\phi] \rangle = \frac{1}{Z} \int \underline{\mathscr{D}\phi e^{-S[\phi]}}\, O[\phi]$$

$$\mathscr{D}\phi = \prod_{i \in \{\mathbb{Z}/L\}^4} d\phi_i$$: ~ 10000 次元、台形法など絶望的

$$= \frac{1}{N} \sum_{k}^{N} O[\phi_k] \ \pm \ O(\frac{1}{\sqrt{N}})$$

マルコフ連鎖: $P[\phi] = \frac{1}{Z} e^{-S[\phi]}$

- 台形法などは積分の次元が高いときに誤差が大きい、時間かかる
- モンテカルロ法は、次元に依存しないので使える
- モンテカルロ法の誤差は、 独立なサンプル数 で決定されている。

ゲージ場配位の生成

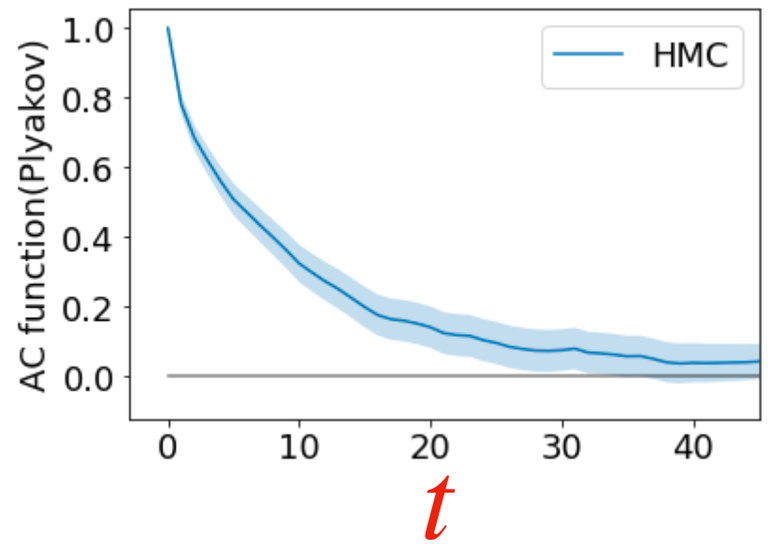
サンプル間に相関→コスパ悪くなる

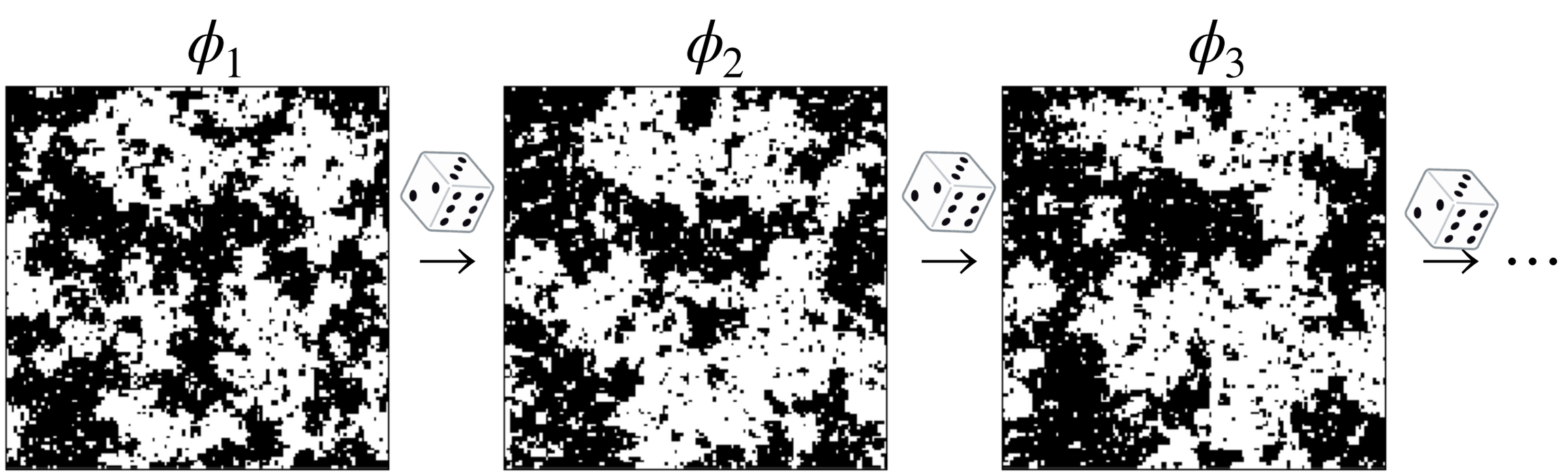
Akio Tomiya

$$\langle O[\phi] \rangle = \frac{1}{N}\sum_{k}^{N} O[\phi_k] \ \pm \ O(\frac{1}{\sqrt{N_{indep}}})$$

$$N_{indep} = \frac{N}{2\tau_{ac}}$$

$$\bar{\Gamma}(t) = \frac{1}{N-t}\sum_{k} (O[\phi_{k+t}] - \bar{O})(O[\phi_k] - \bar{O}) \sim e^{-t/\tau_{ac}}$$





Γ や τ_{ac} は、配位間の類似度を測定している

長い自己相関はいつ起こる？

Akio Tomiya

自己相関は相転移があると長くなる

Data from
Nf=3, standard staggered
with magnetic field

$$L^3 \times N_t = 16^3 \times 4$$
$$ma = 0.03$$

β	N_{conf}	τ_{ac}	N_{indep}
5.166	15k	47	160
5.167	20k	224	45
5.168	20k	656	15
5.169	**20k**	2940	3
5.170	15k	1306	6
5.171	14k	58	116
5.172	10k	48	106

k=1000

Critical temp.

$$N_{indep} = \frac{N_{conf}}{2\tau_{ac}}$$

$$\langle O[\phi] \rangle = \frac{1}{N_{conf}} \sum_{k}^{N_{conf}} O[\phi_k] \ \pm \ O(\frac{1}{\sqrt{N_{indep}}})$$

$$\tau_{ac} \sim \xi^z \sim L^z$$

z : Dynamic critical exponent (see 1703.03136)

τ_{ac}: アルゴリズム依存 (N. Madras et. al 1988)

もし z が小さい(or shorter τ_{ac}) アルゴリズムをみつければ、

相転移点 (~ 連続極限) に近くても計算コストが削減できる。

自己相関は減らせるか？

機械学習で頑張る

Akio Tomiya

$$\langle O[\phi] \rangle = \frac{1}{N} \sum_{k}^{N} O[\phi_k] \ \pm \ O(\frac{1}{\sqrt{N_{indep}}})$$

$$N_{indep} = \frac{N}{2\tau_{ac}}$$

$$\bar{\Gamma}(t) = \frac{1}{N-t} \sum_{k} (O[\phi_{k+t}] - \bar{O})(O[\phi_k] - \bar{O}) \sim e^{-t/\tau_{ac}}$$

τ_{ac} is given by an update algorithm (N. Madras et. al 1988)

- 自己相関時間 τ_{ac} は配位の類似度を与える
- τ_{ac} はアルゴリズムに依存する
- たとえば、τ_{ac}が半分になれば、同じ時間で倍の計算ができる

機械学習をつかって減らせないか？

Markov chain Monte-Carlo?

Akio Tomiya

If detailed balance satisfied, we can sample using it

A key concept is the detailed balance condition:

If an update algorithm P(.|.) satisfies

$$P(\phi_{k'}|\phi_k)e^{-S[\phi_k]} = P(\phi_k|\phi_{k'})e^{-S[\phi_{k'}]}$$

it will converge in a desired distribution (skip proof, + some condition)

$$P_{eq}(\phi) = \frac{1}{\int \mathscr{D}\phi' e^{-S[\phi']}} e^{-S[\phi]}$$

Recent progress

Self-learning Monte Carlo

J.Liu, Y.Qi, Z.Meng, L.Fu (arXiv:1610.03137)

Accept/Reject **Proposing part**

$$P(S_{k'}|S_k) = \min\left(1, \frac{e^{-\beta(H[S_{k'}]-H_{eff}[S_{k'}])}}{e^{-\beta(H[S_k]-H_{eff}[S_k])}}\right) Q_{eff}(S_{k'}|S_k)$$

Corrected by modified Metropolis test

Update using effective model (Cheap)

This is an exact algorithm:
if the effective model far from the system, acceptance is zero.

Recent progress

Self-learning Monte Carlo

J.Liu, Y.Qi, Z.Meng, L.Fu (arXiv:1610.03137)

Accept/Reject **Proposing part**

$$P(S_{k'}|S_k) = \min\left(1, \frac{e^{-\beta(H[S_{k'}]-H_{eff}[S_{k'}])}}{e^{-\beta(H[S_k]-H_{eff}[S_k])}}\right) Q_{eff}(S_{k'}|S_k)$$

Corrected by modified Metropolis test

Update using effective model (Cheap)

This is an exact algorithm:
if the effective model far from the system, acceptance is zero.

Testcase

$$H = -J\sum_{\langle ij\rangle} S_iS_j - K\sum_{ijkl\in\square} S_iS_jS_kS_l,$$

No global update because of 2nd term

$$H_{\text{eff}} = E_0 - \tilde{J}_1\sum_{\langle ij\rangle_1} S_iS_j$$

$$S_i = \pm 1$$

Ising model with parameter $\tilde{J}_1$, which is determined by fitting!
(no fancy update is needed!)
This has global update(cheap)

Recent progress

Self-learning Monte Carlo

J.Liu, Y.Qi, Z.Meng, L.Fu (arXiv:1610.03137)

Accept/Reject **Proposing part**

$$P(S_{k'}|S_k) = \min\left(1, \frac{e^{-\beta(H[S_{k'}]-H_{eff}[S_{k'}])}}{e^{-\beta(H[S_k]-H_{eff}[S_k])}}\right) Q_{eff}(S_{k'}|S_k)$$

Corrected by modified Metropolis test

Update using effective model (Cheap)

This is an exact algorithm.

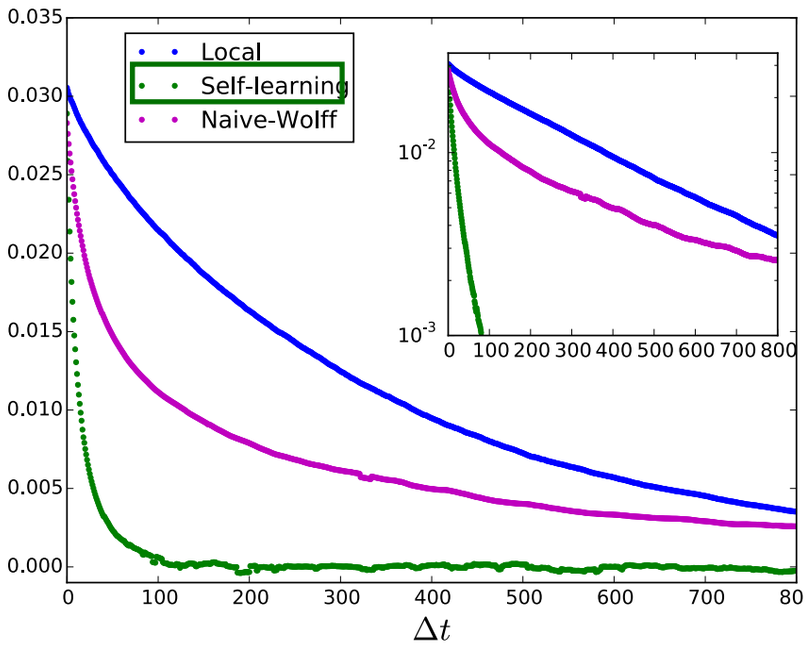
Testcase

$$H = -J\sum_{\langle ij\rangle} S_iS_j - K\sum_{ijkl\in\square} S_iS_jS_kS_l,$$

$$H_{\rm eff} = E_0 - \tilde{J}_1\sum_{\langle ij\rangle_1} S_iS_j$$

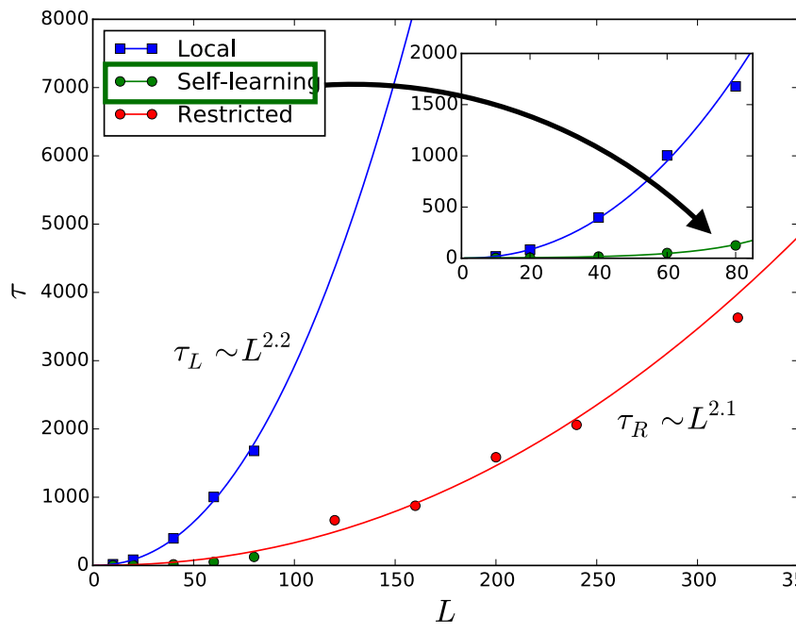
$$S_i = \pm 1$$

Autocorrelation function



24 time efficient

Dynamic Critical exponent



Very mild scaling

Recent progress

QCD with Self-learning Monte Carlo

work in progress

Collaborate with
Akinori Tanaka (Riken iTHENS)
Yuki Nagai (JAEA/ RIKEN AIP)

$$P(U_{k'}|\,U_k) = \min\left(1, \frac{e^{-(S[U_{k'}]-S_{eff}[U_{k'}])}}{e^{-(S[U_k]-S_{eff}[U_k])}}\right) Q_{eff}(U_{k'}|\,U_k)$$

Setup: SU(2) plaquette action + staggered quarks + 4dim

Effective action = hopping parameter expanded action, heatbath

Recent progress

QCD with Self-learning Monte Carlo

work in progress

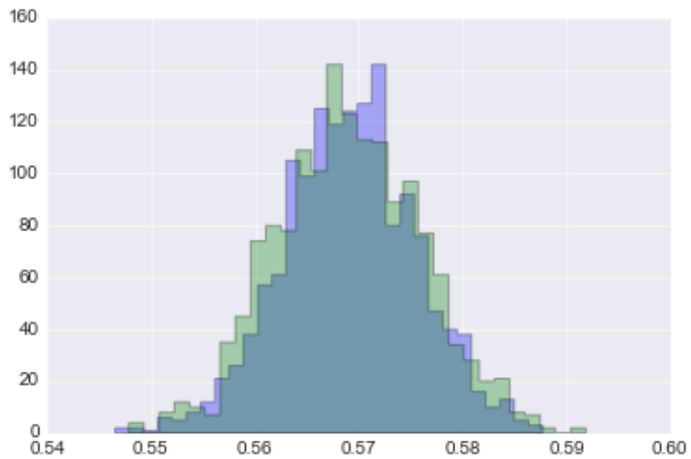
Collaborate with
Akinori Tanaka (Riken iTHENS)
Yuki Nagai (JAEA/ RIKEN AIP)

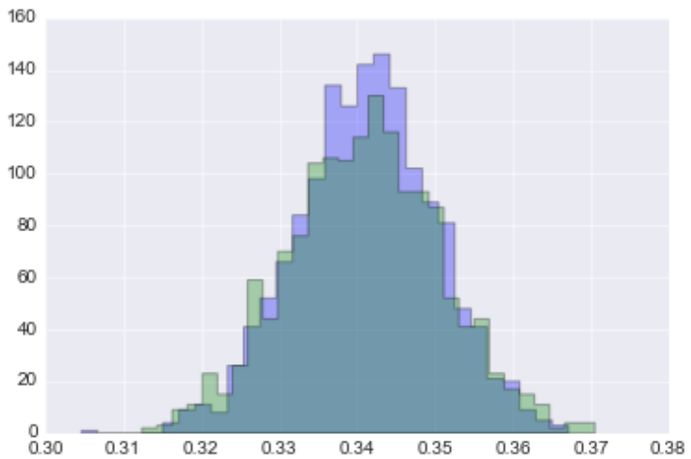
$$P(U_{k'}|\,U_k) = \min\left(1, \frac{e^{-(S[U_{k'}]-S_{eff}[U_{k'}])}}{e^{-(S[U_k]-S_{eff}[U_k])}}\right) Q_{eff}(U_{k'}|\,U_k)$$

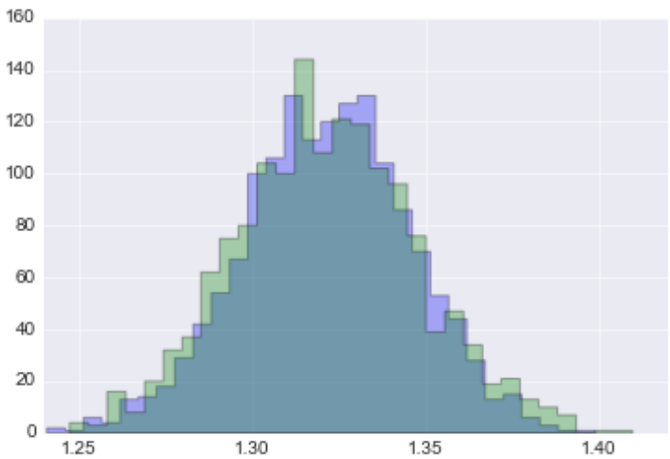
Setup: SU(2) plaquette action + staggered quarks + 4dim

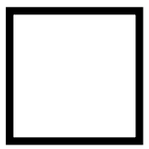
Effective action = hopping parameter expanded action, heatbath

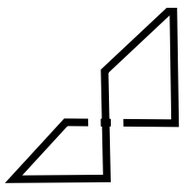
Observables （■=HMC, ■=SLMC）











So far so good

Recent progress

QCD with Self-learning Monte Carlo

work in progress

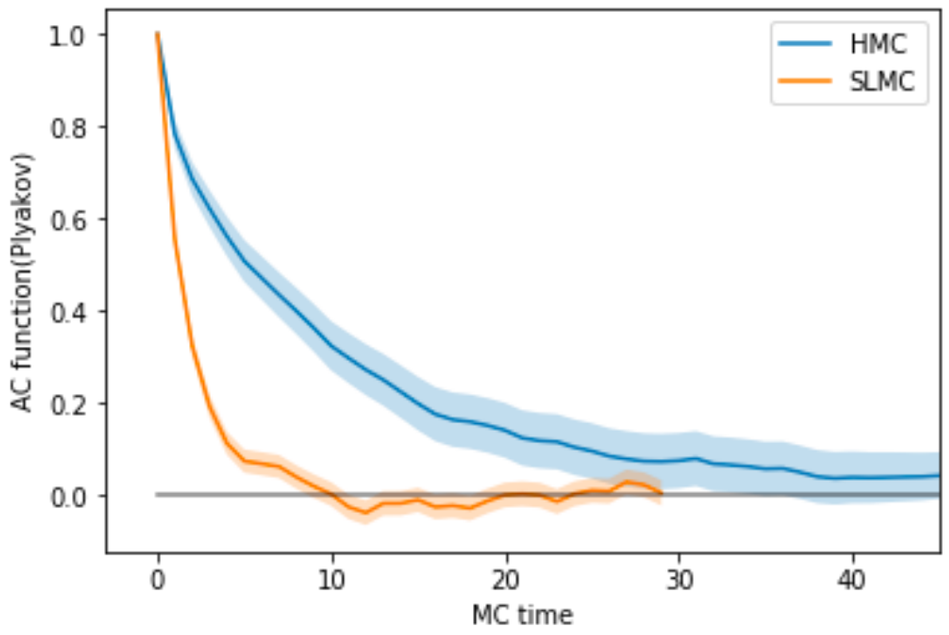
Collaborate with
Akinori Tanaka (Riken iTHENS)
Yuki Nagai (JAEA/ RIKEN AIP)

$$P(U_{k'}|\,U_k) = \min\left(1, \frac{e^{-(S[U_{k'}]-S_{eff}[U_{k'}])}}{e^{-(S[U_k]-S_{eff}[U_k])}}\right) Q_{eff}(U_{k'}|\,U_k)$$

Setup: SU(2) plaquette action + staggered quarks + 4dim

Effective action = hopping parameter expanded action, heatbath

Autocorrelation for Polyakov loop （■=HMC, ■=SLMC）



Recent progress

QCD with Self-learning Monte Carlo

work in progress

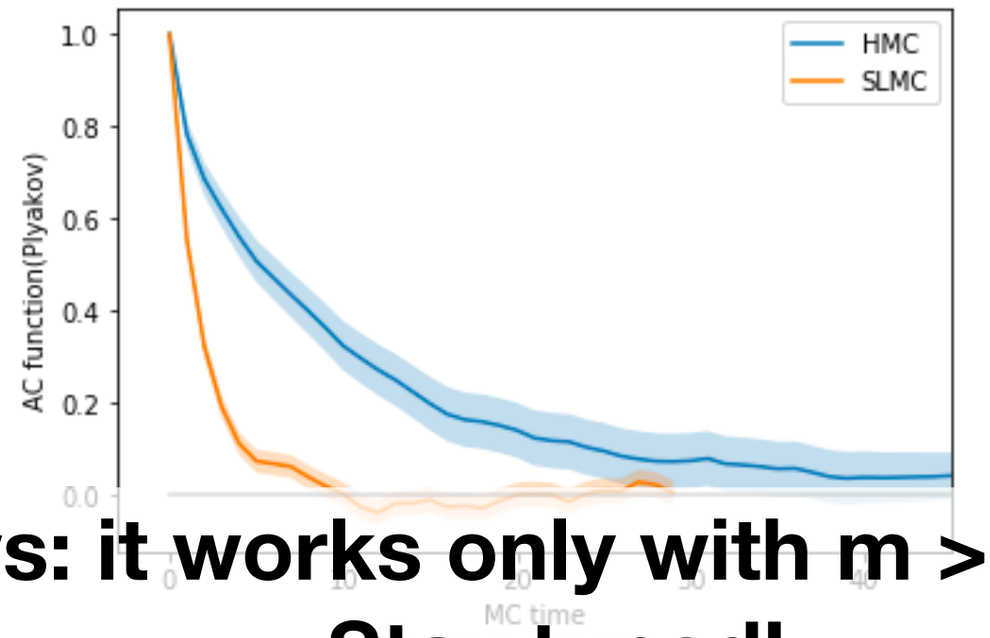
Collaborate with
Akinori Tanaka (Riken iTHENS)
Yuki Nagai (JAEA/ RIKEN AIP)

$$P(U_{k'}|\,U_k) = \min\left(1, \frac{e^{-(S[U_{k'}]-S_{eff}[U_{k'}])}}{e^{-(S[U_k]-S_{eff}[U_k])}}\right) Q_{eff}(U_{k'}|\,U_k)$$

Setup: SU(2) plaquette action + staggered quarks + 4dim

Effective action = hopping parameter expanded action, heatbath

Autocorrelation for Polyakov loop （■=HMC, ■=SLMC）



Bad news: it works only with m > 0.1 for now

Stay tuned!

Flow based algorithm

(un)-Trivializing map

Akio Tomiya

Luscher の夢 (0907.5491)

$$U \xrightarrow{\mathcal{F}^{-1}} V \xrightarrow{\text{HMC}} V' \xrightarrow{\mathcal{F}} U'$$

Fig. 1. The proposed simulation algorithm for lattice QCD updates the gauge field U in three steps, following the arrows in this diagram, where the Hamilton function used in the HMC step has the standard kinetic term and includes the Jacobian of the field transformation $\mathcal{F}$ (cf. subsect. 2.4).

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathrm{D}[U]\, \mathcal{O}(U)\, \mathrm{e}^{-S(U)}$$

$U = \mathcal{F}(V)$ ~ Flow equation

もし $S(\mathcal{F}(V)) - \ln \det \mathcal{F}_*(V) = \text{constant},$ ならば、

$$\langle \mathcal{O} \rangle = \int \mathrm{D}[V]\, \mathcal{O}(\mathcal{F}(V)).$$

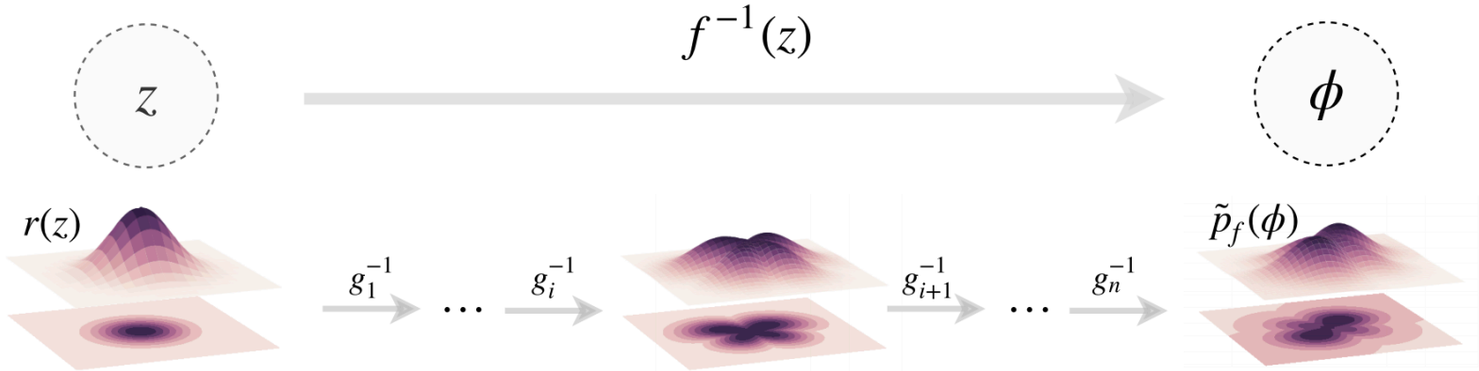
実際には、Wilson flow だとJacobian が…

Flow based algorithm

(un)-Trivializing map

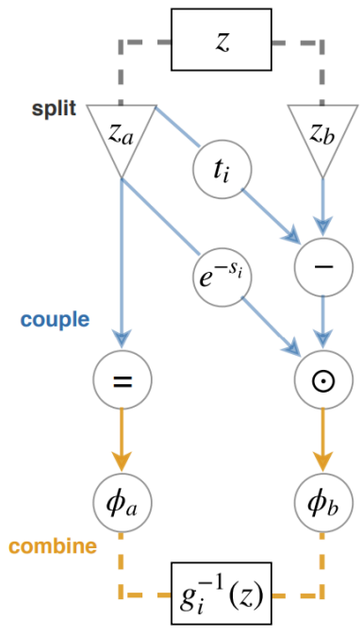
Akio Tomiya

2d scalar



(a) Normalizing flow between prior and output distributions

MIT + Google brain



(b) Inverse coupling layer

FIG. 1: In (a), a normalizing flow is shown transforming samples z from a prior distribution $r(z)$ to samples ϕ distributed according to $\tilde{p}_f(\phi)$. The mapping $f^{-1}(z)$ is constructed by composing inverse coupling layers g_i^{-1} as defined in Eq. (10) in terms of neural networks s_i and t_i and shown diagrammatically in (b). By optimizing the neural networks within each coupling layer, $\tilde{p}_f(\phi)$ can be made to approximate a distribution of interest, $p(\phi)$.

ニューラルネットで可逆なTrivializing map を再現

もし作れれば、自由場でサンプルして逆にたどればOK

Normalizing flow という仕組みでできる

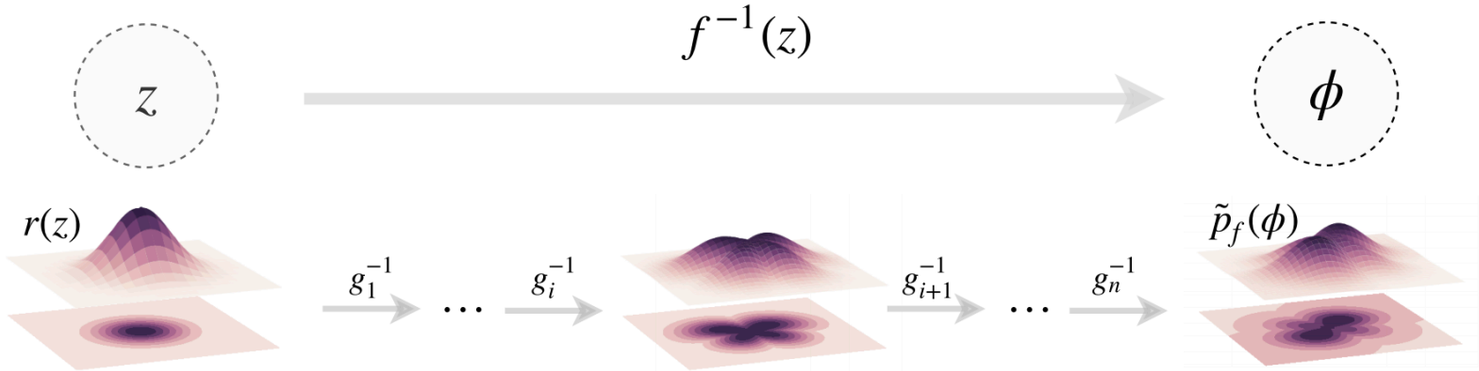
1904.12072, 2003.06413, 2008.05456

Flow based algorithm

(un)-Trivializing map

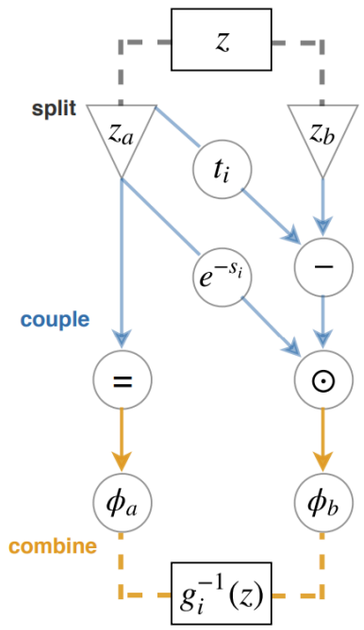
Akio Tomiya

2d scalar



(a) Normalizing flow between prior and output distributions

MIT + Google brain



(b) Inverse coupling layer

FIG. 1: In (a), a normalizing flow is shown transforming samples z from a prior distribution $r(z)$ to samples ϕ distributed according to $\tilde{p}_f(\phi)$. The mapping $f^{-1}(z)$ is constructed by composing inverse coupling layers g_i^{-1} as defined in Eq. (10) in terms of neural networks s_i and t_i and shown diagrammatically in (b). By optimizing the neural networks within each coupling layer, $\tilde{p}_f(\phi)$ can be made to approximate a distribution of interest, $p(\phi)$.

格子上のz をガウシアンでサンプル→inverse trivializing map

で場の理論の作用にあう様に変形していく。Jacobian も計算できる。

アンサンブルを作った後にメトロポリステスト→厳密！

1904.12072, 2003.06413, 2008.05456

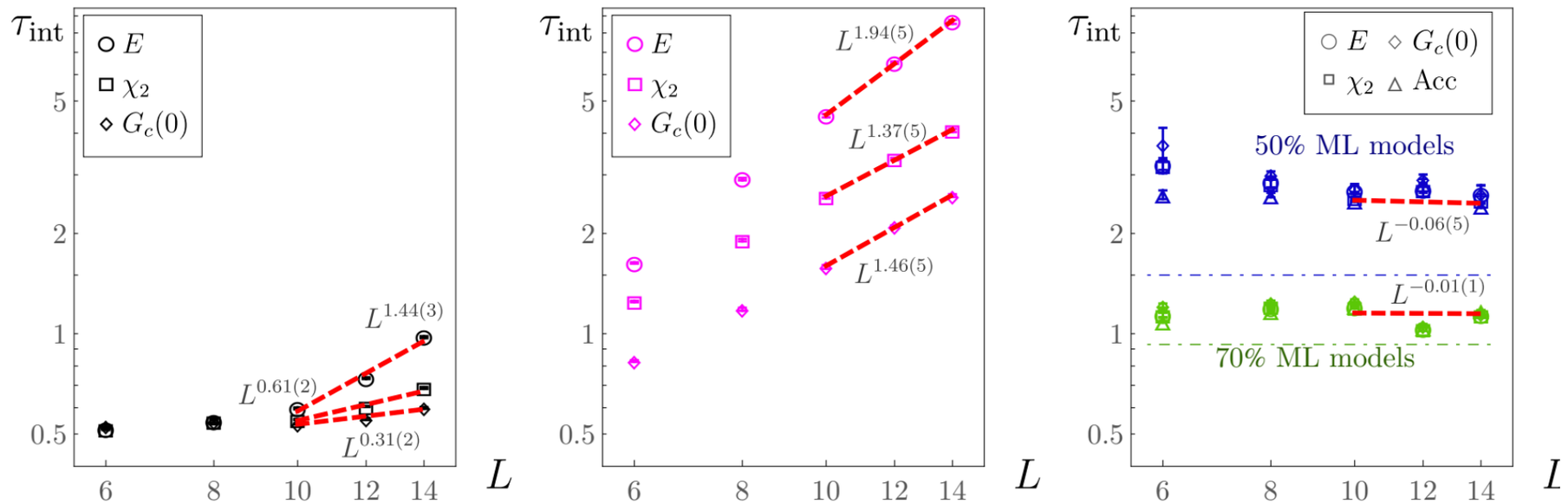
Flow based algorithm

(un)-Trivializing map

Akio Tomiya

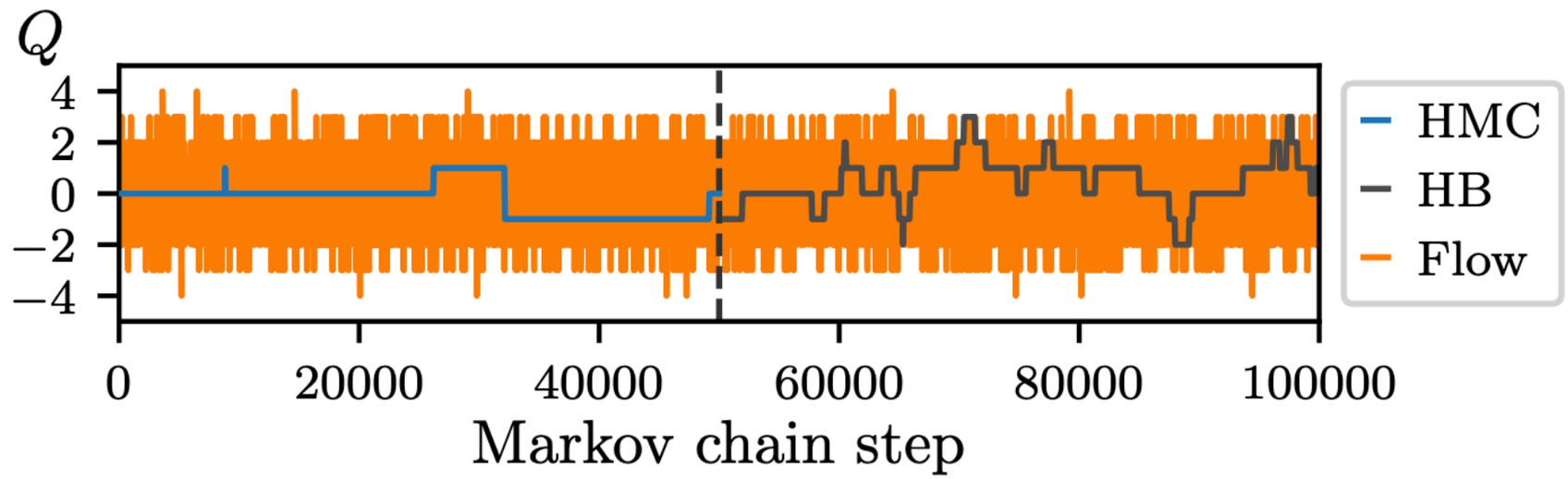
MIT + Google brain

2d scalar



(a) HMC ensembles (b) Local Metropolis ensembles (c) Flow-based MCMC ensembles

2d pure U(1)



2d pure SU(N) の結果もある。

1904.12072, 2003.06413, 2008.05456

Summary and outlook for here

Machine learning + MCMC = efficient(?)

- Markov chain Monte-Carlo enables us to evaluate expectation values of QFT but inefficiency comes from algorithm in practice = autocorrelation

- Autocorrelation can be reduced by using RBM but convergence is not guaranteed. We need to monitor distribution of observables

- Other works with GAN share same problem (worse?) for convergence. It is not guaranteed.

- Self-learning Monte-Carlo(SLMC) for LQCD might work well, at least it is a converging algorithm and **it can treat gauge field** but a lot of effort is needed.

- FLOW based model is OK, application in low dimension, quenched

2020年代にあたって

2020年代にあたって

2016年以降、色々方向性が固まってきた。次は？

機械学習を物理(格子QCD)に応用できることはわかった。

ほんとに良いの？

- 配位生成はOKとしても物理量は信頼できない。エラーバーは？
  → 出力の不定性を評価できれば…

- 見知らぬデータに関する誤差？
  → 汎化の理解が進めば…

- アルゴリズムはスケールするの？
  → 商用の応用があるのでこれは多分OK

- ほんとに信用できたり使える道具になるためにはこれらをなんとかしないと。

- もしこれらが乗り越えられたらもっと強力な道具に。

Summary

Machine learning provides us new techniques

- 通常の格子QCDの歴史でたとえるなら、1980年代前半から後半の領域に入ってきた？ (できないことも多いが、挑戦できること多そう)

- 分野としての課題 (2020年代?)

- ニューラルネットの汎化の理解 (物理→機械学習の応用？)

- 系統誤差のない/系統誤差を評価できる アルゴリズム・枠組みの開発 (工学的な道具から科学で使える道具へ)