(格子場の理論向けの) 機械学習

東京女子大学

富谷 昭夫

MLPhYs Foundation of "Machine Learning Physics"
Grant-in-Aid for Transformative Research Areas (A)

Program for Promoting Researches on the Supercomputer Fugaku
Large-scale lattice QCD simulation and development of AI technology

akio_at_yukawa.kyoto-u.ac.jp

富谷昭夫 (とみやあきお)

AI技術 + 格子QCD

何者? 格子QCDでも主に有限温度QCD

機械学習を格子QCDの応用

主な論文 https://scholar.google.co.jp/citations?user=LKVqy_wAAAAJ

Detection of phase transition via convolutional neural networks
A Tanaka, A Tomiya
Journal of the Physical Society of Japan 86 (6), 063001

AIで相転移検出

Digital quantum simulation of the schwinger model with topological term via adiabatic state preparation
B Chakraborty, M Honda, T Izubuchi, Y Kikuchi, A Tomiya
arXiv preprint arXiv:2001.00485

素粒子の計算を量子コンピュータで

経歴

2006-2010	: 兵庫県立大学 (超伝導の理論)
2015	: 大阪大学大学院・博士 (素粒子論)　深谷さんの学生
2015 - 2018	: 中国・武漢でポスドク H-T Ding教授
2018 - 2021	: アメリカ・ニューヨークでポスドク(理研基礎特)
2021 - 2024	: 大阪国際工科専門職大学で助教
2024 -	: 東京女子大学で専任講師

研究費

MLPhYs Foundation of "Machine Learning Physics"
Grant-in-Aid for Transformative Research Areas (A)

+ 量子計算

Program for Promoting Researches on the Supercomputer Fugaku
Large-scale lattice QCD simulation and development of AI technology

田中章詞
富谷昭夫
橋本幸士
DEEP LEARNING AND PHYSICS
ディープラーニングと物理学
Mathematical Physics Studies
Akinori Tanaka
Akio Tomiya
Koji Hashimoto
Deep Learning and Physics
Springer
橋本幸士
物理学者、機械学習を使う
機械学習・深層学習の物理学への応用
Using Machine Learning for Physics
朝倉書店
Understanding Machine Learning from Scratch
富谷昭夫
Akio Tomiya
これならわかる機械学習入門

Organizing “Deep Learning and physics”

https://cometscome.github.io/DLAP2020/

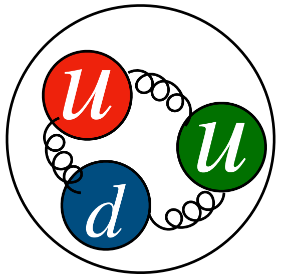
その他:

2023 シン・仮面ライダー監修

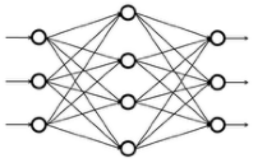
2024 第29回 日本物理学会 論文賞

2019 第14回 素粒子メダル奨励賞

もくじ



格子QCD



ニューラルネット

格子QCDのための
機械学習と応用

1. フローベース法
2. 完全作用
3. 自己学習モンテカルロ法
4. CASK: Gauge symmetric transformer
5. バイアス補正近似

MLPhys?

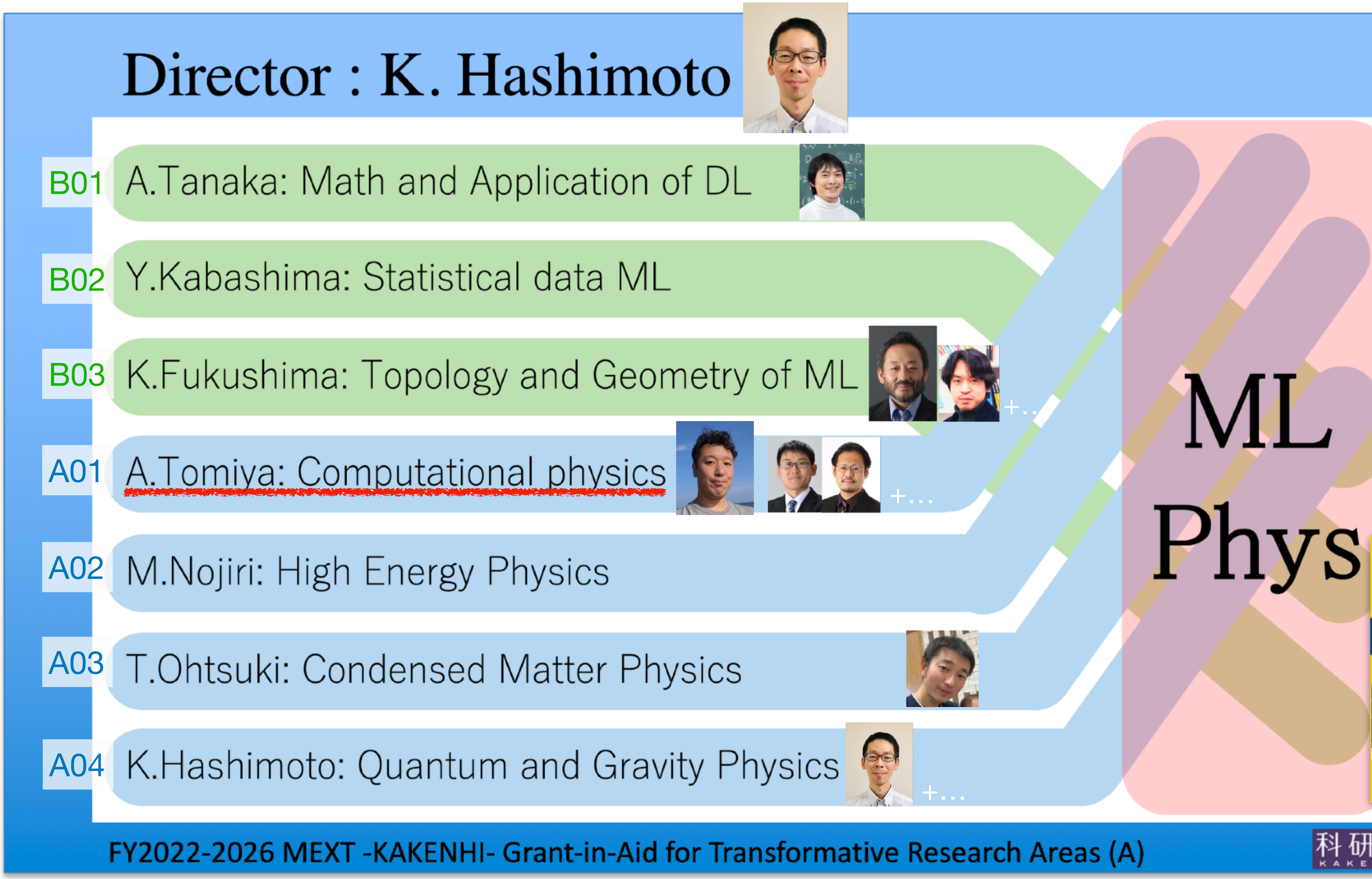
MLPhys Foundation of "Machine Learning Physics"
Grant-in-Aid for Transformative Research Areas (A)

My team: LQCD + ML

"Machine Learning Physics Initiative"

MLPhYs

2022-2027, 10M USD, 70 researchers



Director : K. Hashimoto

B01 A.Tanaka: Math and Application of DL

B02 Y.Kabashima: Statistical data ML

B03 K.Fukushima: Topology and Geometry of ML +...

A01 A.Tomiya: Computational physics +...

A02 M.Nojiri: High Energy Physics

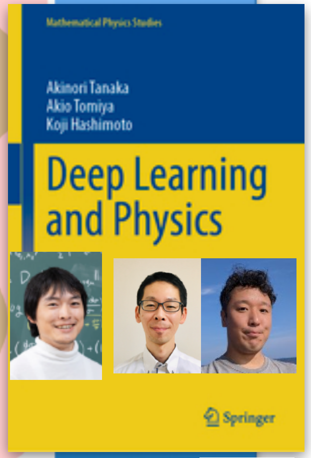
A03 T.Ohtsuki: Condensed Matter Physics

A04 K.Hashimoto: Quantum and Gravity Physics +...

ML Phys

FY2022-2026 MEXT -KAKENHI- Grant-in-Aid for Transformative Research Areas (A)

科研費 KAKENHI



Mathematical Physics Studies

Akinori Tanaka
Akio Tomiya
Koji Hashimoto

Deep Learning and Physics

Springer

2021

https://mlphys.scphys.kyoto-u.ac.jp/en/

String data 2024

https://mlphys.scphys.kyoto-u.ac.jp/stringdata2024/

2024, Dec.10-12
Yukawa Institute, Kyoto University, Japan

The deadline for the registration for the on-site participants is October 31st. 2024.

Confirmed invited speakers

- Yago Bea (University of Barcelona)
- Gabriel Lopes Cardoso (Lisbon, IST)
- François Charton (META AI)
- Sergei Gukov (Caltech)
- James Halverson (Northeastern University)
- Song He (Jilin University / Max Planck Institute Potsdam)
- Edward Hirst (Queen Mary, University of London)
- Vishunu Jejjala (University of the Witwatersrand in Johannesburg)
- Hyun-Sik Jeong (Institute for Theoretical Physics UAM-CSIC in Madrid)
- Keun-Young Kim (GIST)
- Sven Krippendorf (Arnold Sommerfeld Center for Theoretical Physics, LMU Munich)
- Anindita Maiti (Perimeter Institute)
- Fabian Ruehle (Northeastern University)
- Matthew Schwartz (Harvard University)
- Rak-Kyeong Seong (UNIST)
- Eva Silverstein (Stanford)

Organizers

Hashimoto, Koji (Kyoto University, chair)
Yoshida, Kentaroh (Saitama University)
Murata, Masaki (Saitama Institute of Technology)
Sugishita, Sotaro (Kyoto University)
Hirono, Yuji (Osaka University)
Sannai, Akiyoshi (Kyoto University)
Yoda, Takuya (Kyoto University)
Hikida, Yasuyuki (Kyoto University)
Tanahashi, Norihiro (Kyoto University)

My team (A01): LQCD + ML

MLPhYs

PI: Akio Tomiya (Me)
TWCU
LQCD, ML

Kouji Kashiwa
Fukuoka Institute of Technology
LQCD, ML

Hiroshi Ohno
U. of Tsukuba
LQCD

Tetsuya Sakurai
U. of Tsukuba
Computation

Yasunori Futamura
U. of Tsukuba
Computation

post-docs
& external members

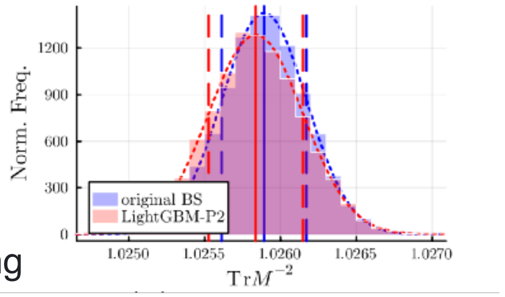
B. J. Choi
U. of Tsukuba

J. Takahashi
Meteorological College

Y. Nagai
U. of Tokyo

- **Apply machine learning techniques on LQCD (To increase what we can do)**
- **Find physics-oriented ML architecture**
- **Making codes for LQCD + ML**

measurement with BDT



On going

LatticeQCD.jl

Open source

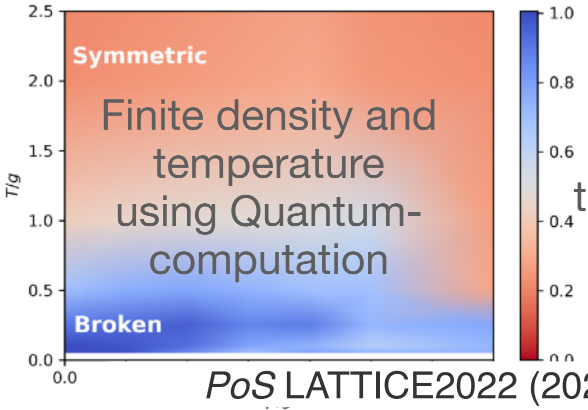
LQCD (+ML) with julia

This covers most of modern tech

https://github.com/akio-tomiya/LatticeQCD.jl

(and associated sub-libraries)

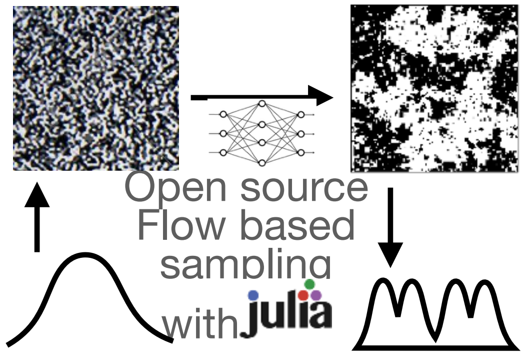
arXiv: 2409.03030



Finite density and temperature using Quantum-computation

PoS LATTICE2022 (2023) 039

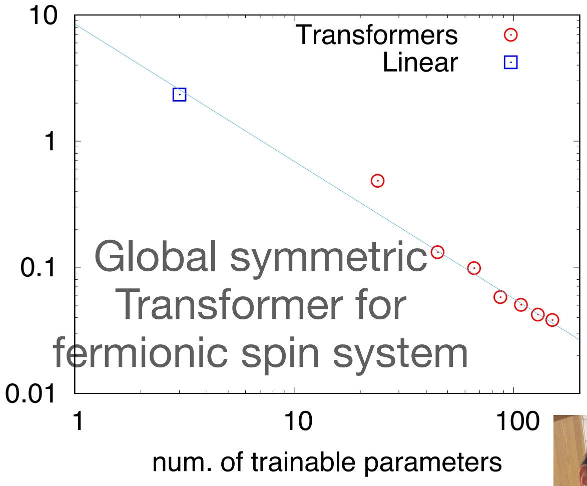
ML + QC:
Quantum thermodynamics using Density matrix and MADE



Open source
Flow based sampling
with julia

arXiv: 2208.08903

ML Phys A01



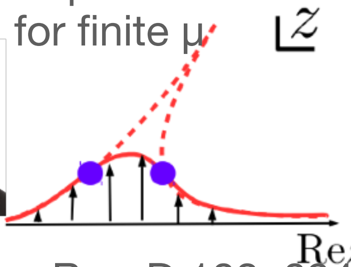
Global symmetric Transformer for fermionic spin system

2310.13222
2306.11527

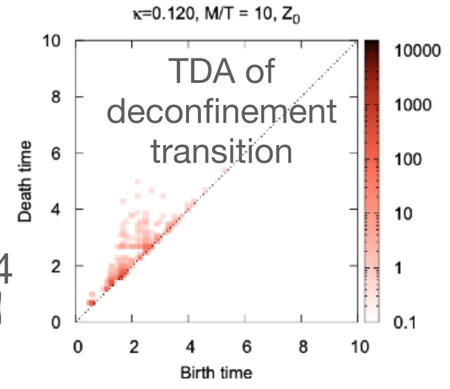
$$\frac{dU_\mu^{(t)}(n)}{dt} = \mathcal{G}^{\bar{\theta}}(U_\mu^{(t)}(n))$$

Gauge covariant neural net
arXiv: 2103.11965

Path optimization for finite μ



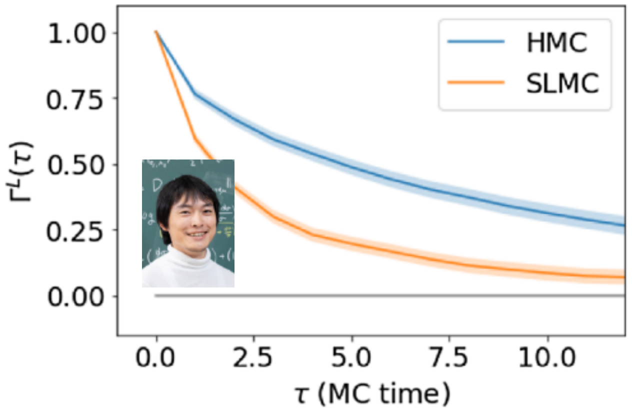
Phys. Rev. D 108, 094504
(Figure from 1812.11506)



TDA of deconfinement transition

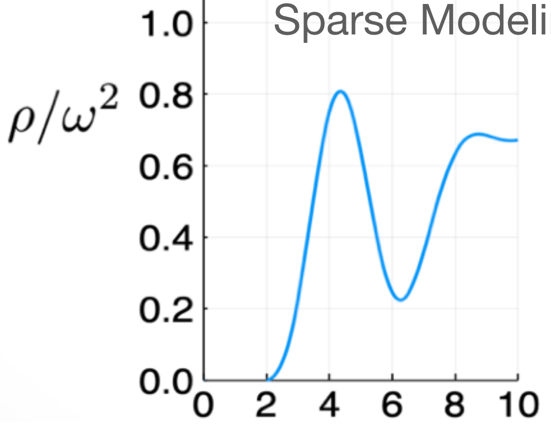
1810.07635

Gauge invariant self-learning MC



Phys. Rev. D 107, 054501

Spectral function with Sparse Modeling

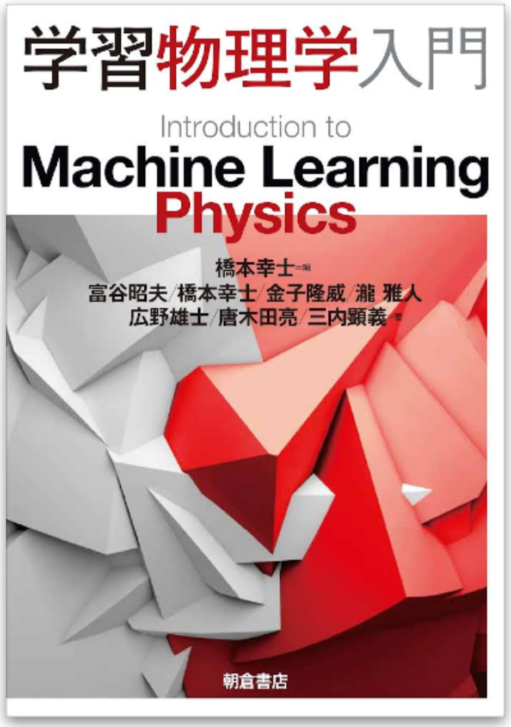


(arXiv: 2311.15233)
+ on going

MLPhYs

2冊、本がでます。

- 学習物理学入門
  - 2024年11月発売 https://www.asakura.co.jp/detail.php?book_code=13152&srsltid=AfmBOooFKDHyAwIo46LHKRhvYc9gCX8ukX0pGhUxlusQMDkdi8P026SX
  - 内容 統計的推定、ニューラルネットの基礎 トランスフォーマー、言語モデル、ニューラルネットの平均場理論、ニューラルネットの波動関数、...



- 格子QCD入門（柏、大野、富谷）
  - 来年刊行

準備中

CM
僕が関わってるものでいくつかお知らせがあります。

- 富岳・研究支援パートタイマー(共同研究者)
  - → 詳しくはチラシ 富谷 akio-tomiya@lab.twcu.ac.jp まで。
- JuliaQCDのコントリビューター募集
  - 新しく何かを付け足したりバグを見つけたら気軽にプルリクエストください。
- 東京女子大学・大学院生募集 (修士女性のみ。博士課程から男性もOK)
  - 尾田欣也 odakin@lab.twcu.ac.jp まで。

格子QCDの数値計算の諸問題

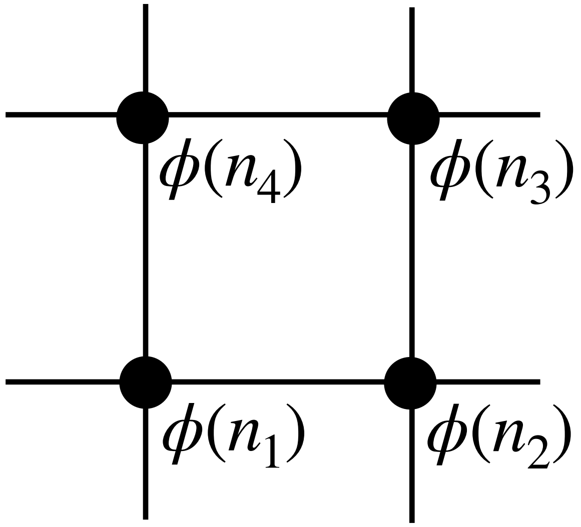
格子QCDの数値計算の諸問題

場の理論は難しい。何もわからん。

$$\int D\phi e^{-S[\phi]} O[\phi]$$

$$D\phi = \prod_{n \in \mathrm{lat}} \mathrm{d}\phi(n)$$

格子上のスカラー場



経路積分 = 多次元の積分

L = 10 -> L^4で1万次元の多重積分

これができれば場の理論が「解ける」

すべての物理量がわかる

(cf 水素原子の波動関数**)**

ミレニアム懸賞金問題

任意のコンパクトな単純ゲージ群 **G** に対して、非自明な量子ヤン・ミルズ理論が **{R} ^{4}}** 上に存在し、質量ギャップ **Δ > 0** を持つことを証明せよ。存在とは**…(**略**)…**。 **[1]**

とにか解ければ良い。

道具は、何を使っても良い。

[1] https://ja.wikipedia.org/wiki/%E3%83%A4%E3%83%B3%E2%80%93%E3%83%9F%E3%83%AB%E3%82%BA%E6%96%B9%E7%A8%8B%E5%BC%8F%E3%81%A8%E8%B3%AA%E9%87%8F%E3%82%AE%E3%83%A3%E3%83%83%E3%83%97%E5%95%8F%E9%A1%8C

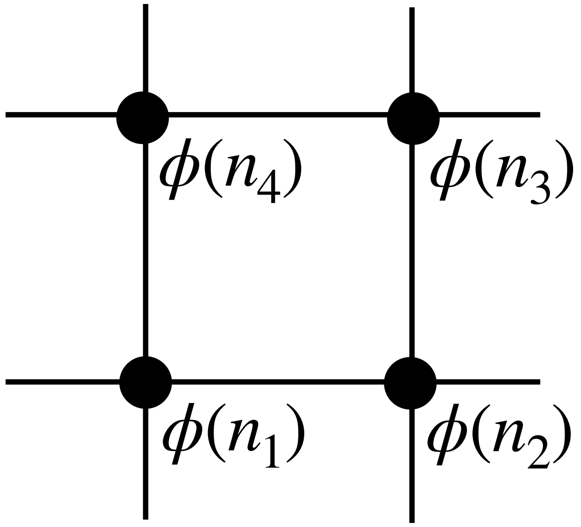
格子QCDの数値計算の諸問題

経路積分は難しい

$$\int D\phi e^{-S[\phi]}O[\phi]$$

$$D\phi = \prod_{n\in\text{lat}} \mathrm{d}\phi(n)$$

格子上のスカラー場



経路積分 = 多次元の積分

L = 10 -> L^4で1万次元の多重積分

これができれば場の理論が「解ける」

すべての物理量がわかる

(cf 水素原子の波動関数**)**

別の例

イジング模型**:** 状態和を閉じた式で書ければ良いが、**3**次元では未だにわからない。

$$Z = \sum_{\sigma_1=\pm 1}\cdots\sum_{\sigma_n=\pm 1} \mathrm{e}^{-\beta H[\sigma]}$$

問題としては同質

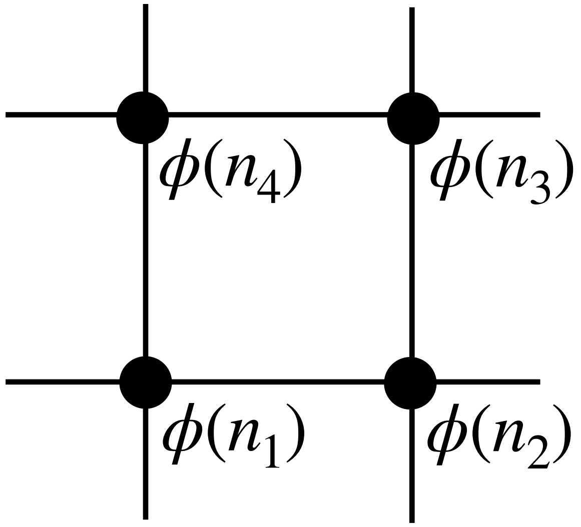
格子QCDの数値計算の諸問題

経路積分は難しい

$$\int D\phi e^{-S[\phi]}O[\phi]$$

$$D\phi = \prod_{n\in \mathrm{lat}} \mathrm{d}\phi(n)$$

格子上のスカラー場



経路積分 = 多次元の積分

L = 10 -> L^4で1万次元の多重積分

これができれば場の理論が「解ける」

すべての物理量がわかる

(cf 水素原子の波動関数)

積分は大変。ゲージ場なら

4(方向)x8(自由度)x2(複素) = 64倍

の64万次元積分。手では無理そう。

計算機でも台形法(≒区分求積法)では無理。

モンテカルロ法で頑張る。

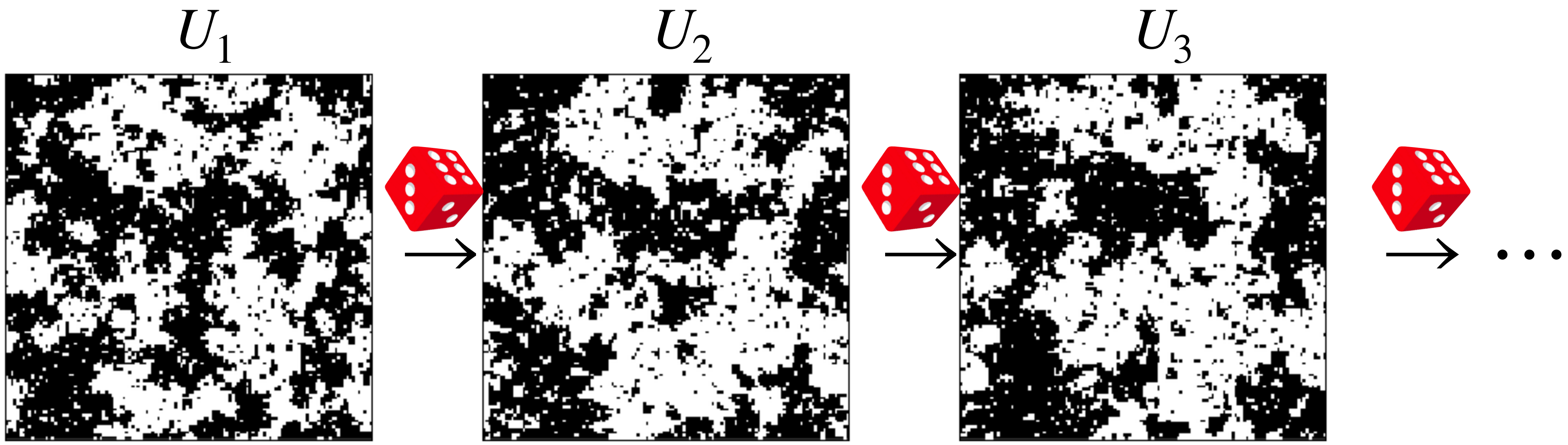
Introduction

Monte-Carlo integration is available

M. Creutz 1980

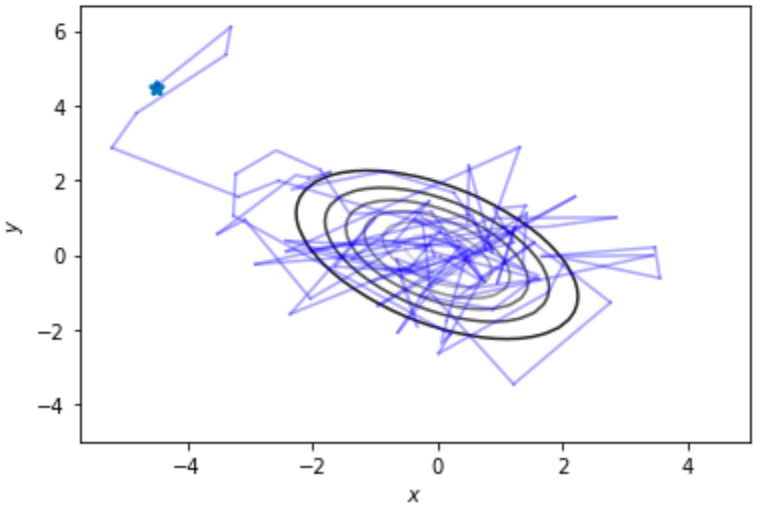
$$\langle \mathcal{O} \rangle = \frac{1}{Z}\int \mathcal{D}U e^{-S_{\rm eff}[U]}\mathcal{O}(U) \qquad S_{\rm eff}[U] = S_{\rm gauge}[U] - \log\det(\not{D}[U]+m)$$

Monte-Carlo: Generate field configurations with “$P[U] = \frac{1}{Z}e^{-S_{\rm eff}[U]}$”**. It gives expectation value**



HMC: Hybrid (Hamiltonian) Monte-Carlo
De-facto standard algorithm

$$S(x,y) = \frac{1}{2}(x^2 + y^2 + xy)$$

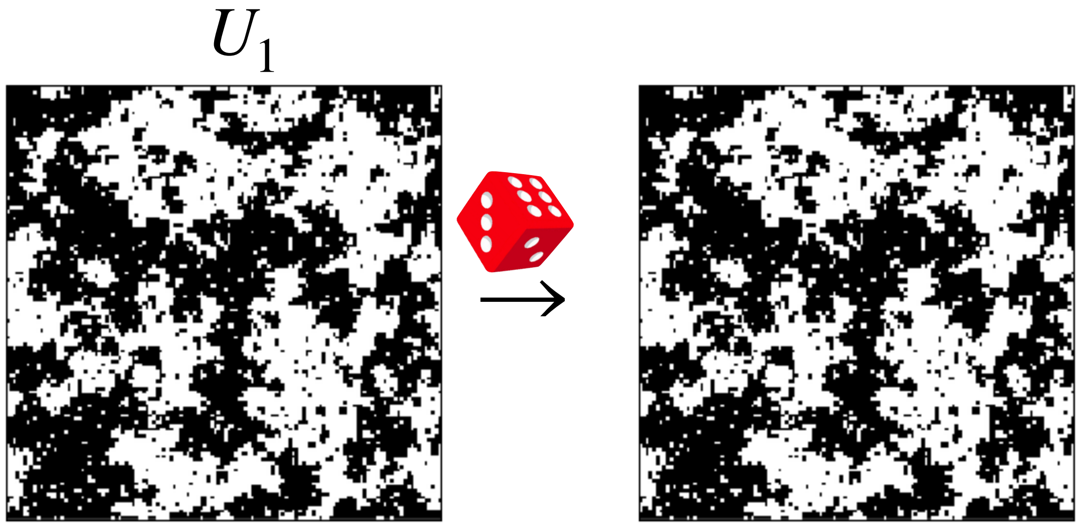


Duane 1987

Introduction

Monte-Carlo integration is available

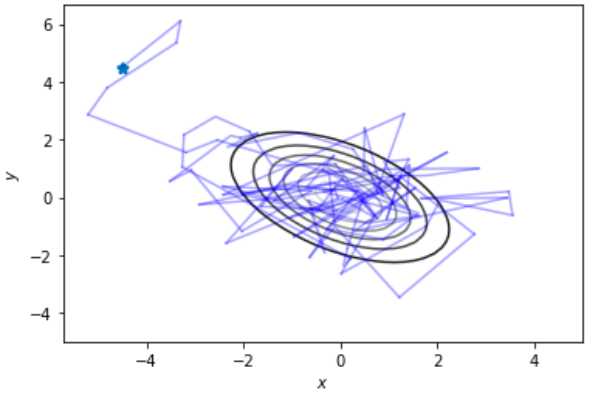
Duane 1987
[1] ref in N. Christ https://arxiv.org/abs/1103.1810



HMC: Hybrid (Hamiltonian) Monte-Carlo
De-facto standard algorithm

$$S(x,y)=\frac{1}{2}(x^2+y^2+xy)$$

ノリとしては、ランジュバンだと思って良い。



HMC

1 $S(x,y)$ を(x,y)に対するポテンシャルだと思い直す。

2 架空の力学系$H=\frac{1}{2}p_x^2+\frac{1}{2}p_y^2+S(x,y)$をでっち上げる。

3 p_x,p_y の初期値をガウシアン乱数で振る (ランダム力)。

4 時間発展をハミルトン方程式に従って解く。$\dot{p}_x=-\frac{\partial S}{\partial x}$
(古典力学 + ランダム力、確率的量子化[1])

5 メトロポリス法で時間発展の誤差を消去。量子論として正しい分布

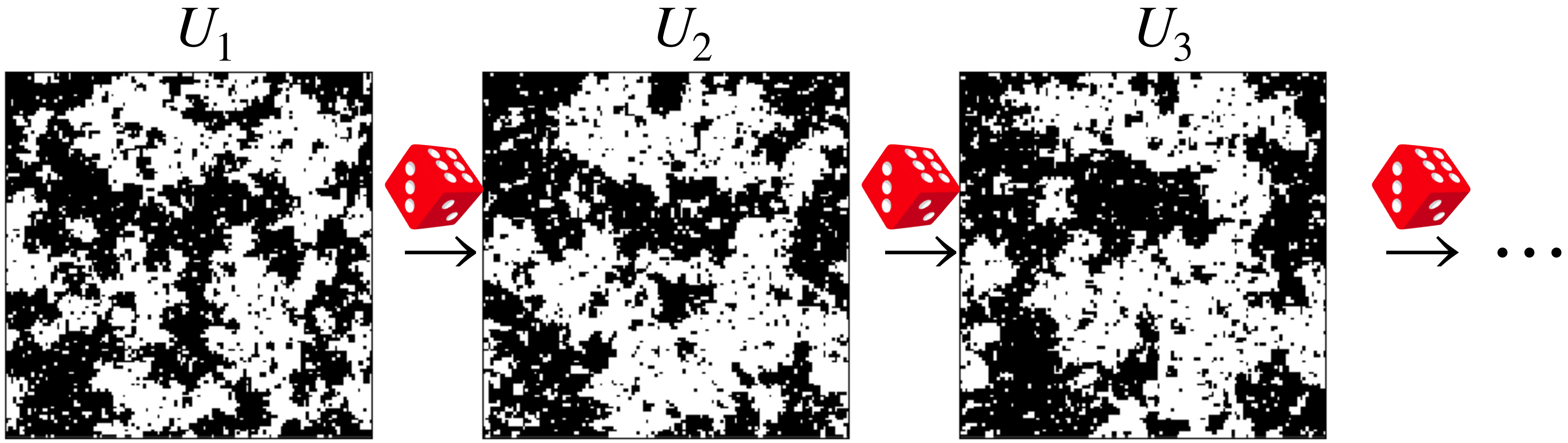
Introduction

Monte-Carlo integration is available

M. Creutz 1980

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}U e^{-S_{\mathrm{eff}}[U]} \mathcal{O}(U) \qquad S_{\mathrm{eff}}[U] = S_{\mathrm{gauge}}[U] - \log \det(\not{D}[U] + m)$$

Monte-Carlo: Generate field configurations with “$P[U] = \frac{1}{Z} e^{-S_{\mathrm{eff}}[U]}$”**. It gives expectation value**



Error of integration is determined by the number of sampling

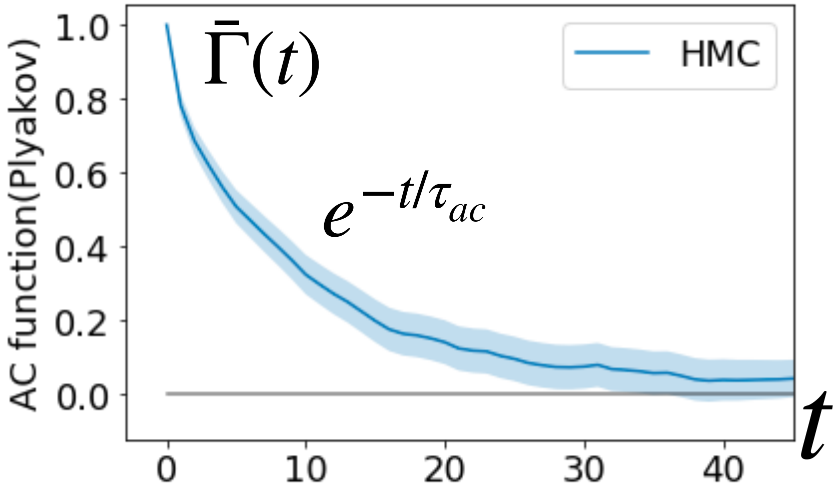
$$\langle \mathcal{O} \rangle = \frac{1}{N_{\mathrm{sample}}} \sum_{k}^{N_{\mathrm{sample}}} \mathcal{O}[U_k] \pm O\left(\frac{1}{\sqrt{N_{\mathrm{sample}}}}\right)$$

Introduction

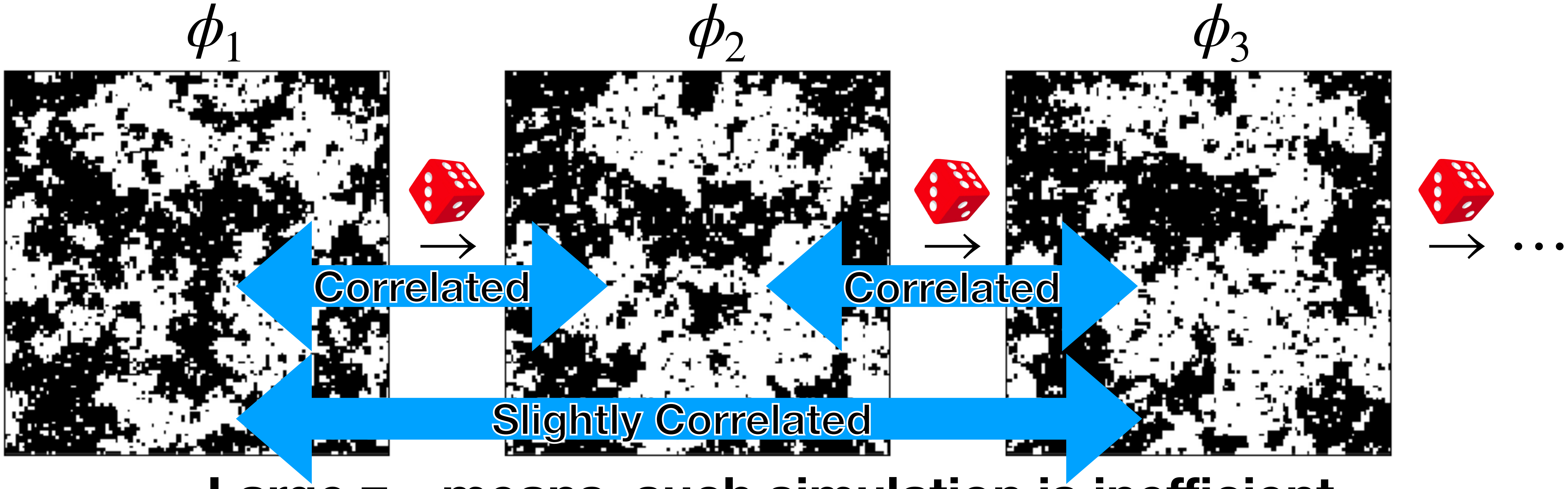
Correlation between samples = inefficiency of calculation

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

$$N_{\text{indep}} = \frac{N_{\text{sample}}}{2\tau_{ac}}$$



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



Large τ_{ac} means, such simulation is inefficient

Introduction

Long autocorrelation around the critical temperature

実際の格子QCD計算

arXiv:2006.13422
Nf=3, **dynamical staggered**
with magnetic field

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

β	N_{sample}	τ_{ac}	N_{indep}
5.166	15,000	47	160
5.167	20,000	224	45
5.168	20,000	656	15
5.169	**20,000**	2940	3
5.170	15,000	1306	6
5.171	14,000	58	116
5.172	10,000	48	106

$$N_{\text{indep}} = \frac{N_{\text{sample}}}{2\tau_{ac}}$$

相転移温度

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

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

z : 動的臨界指数 (see 1703.03136)

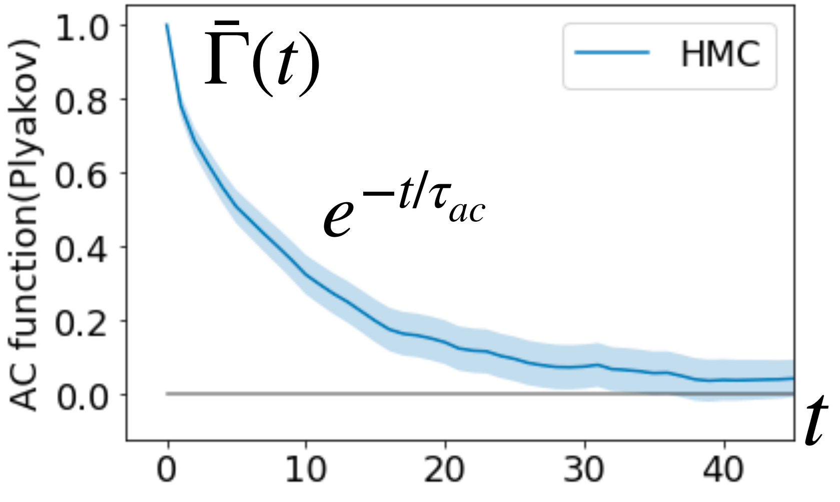
τac: アルゴリズム・物理量による

(N. Madras et. al 1988)

Introduction

Summary for now: long autocorrelation = inefficiency

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



$$N_{\text{indep}} = \frac{N_{\text{sample}}}{2\tau_{ac}}$$

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

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

- Autocorrelation time τ_{ac} quantifies similarity between samples
- τ_{ac} is algorithm dependent quantity
- If τ_{ac} becomes half, we can get doubly precise results in the same time cost

Can we make this mild using machine learning?

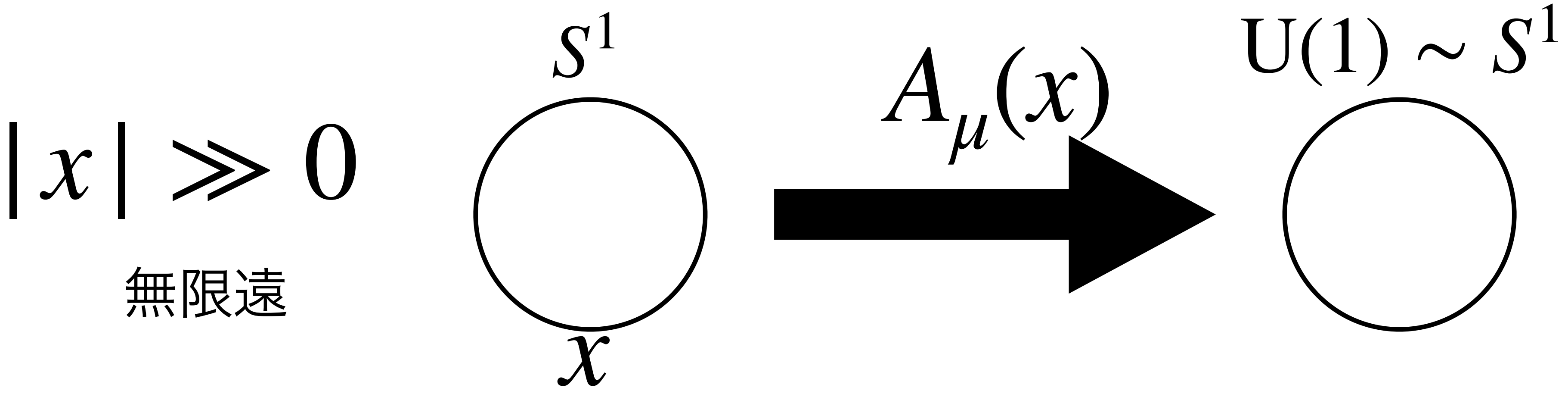
トポロジー凍結

ゲージ場のトポロジー

ゲージ場には、トポロジーがある(整数で区別できる配位がある)。連続理論の古典解で

2次元、U(1)ゲージ理論での巻き付き

4次元、SU(N) のインスタントン



何回も巻き付いても良い -> 巻き付き数。連続変形(運動方程式)では巻き付きを変えれない

量子論でも経路積分の鞍点を与えるため重要。QCDでも。

トポロジカル電荷 (~巻き付き数) Q の2乗はアクシオンの質量と関係

$$m_a \propto \langle Q^2 \rangle$$

軸性U(1)アノマリーとも深い関係。カイラル対称性と深く結びついている。

指数定理。

トポロジー凍結

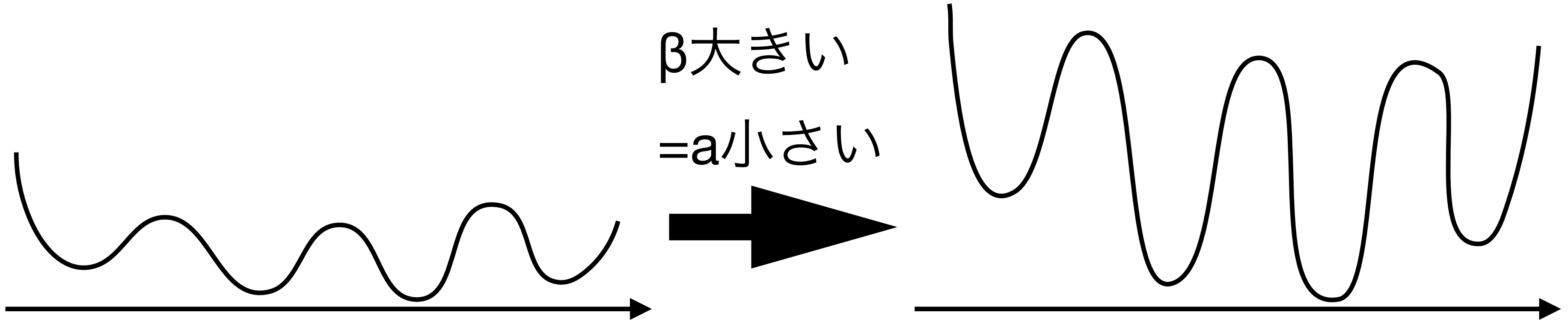
ゲージ場のトポロジー

$$Z(\theta) = \sum_{Q} \int DA^{(Q)}\ \mathrm{e}^{-S+\mathrm{i}\theta Q}$$

トポロジカル電荷Qに対するグランドカノニカルが自然な量子論 (**クラスター分解性)。Qについて、足し上げる必要がある。自然界ではθはなぜか0 (強いCP問題)。

大きなβ (= 小さな格子間隔a) でHMCを実行すると、ゲージ場のトポロジーが変化しなくなってしまう。トポロジー凍結問題。長い自己相関の種にもなる。

理由: HMC は、$\dot{p} = -\partial H/\partial U = -\partial S/\partial U$ という古典運動方程式とランダム力で計算を進めるアルゴリズム。連続理論ではトポロジカルセクターは移りあえないので宿命。運動方程式に基づかない、HMC以外のアルゴリズムが必要(?)



格子アーティファクトを削減するには、小さなa がほしいのでかなり悩ましい。

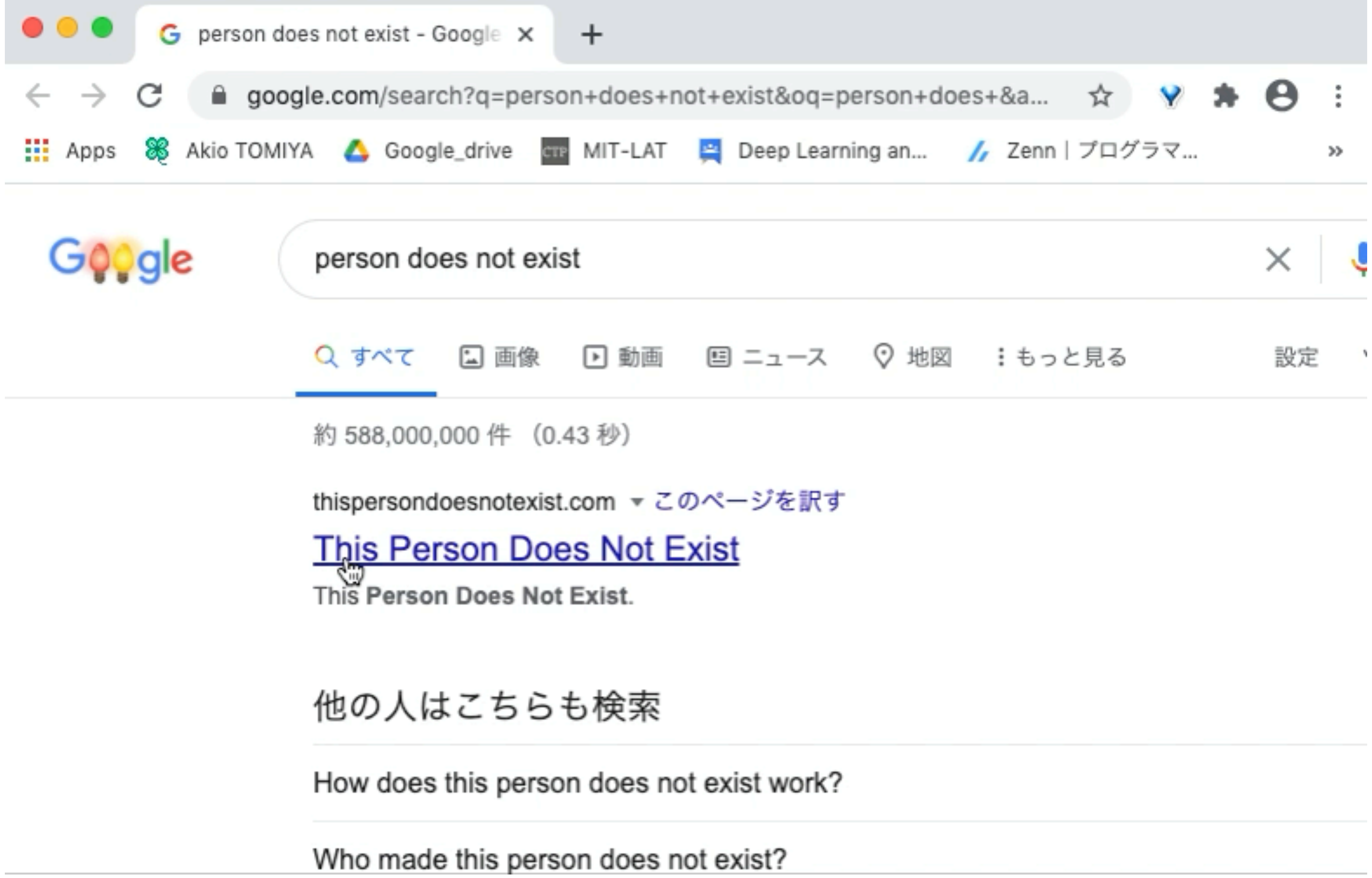
https://arxiv.org/abs/0710.4469, https://arxiv.org/abs/0707.0396 ** https://doi.org/10.1016/0370-2693(76)90277-X

Introduction
他にも問題がある

- 離散化誤差
- CG ソルバーの加速
- 測定の効率化
- 符号問題

Introduction

Neural net can make human face images



person does not exist

約 588,000,000 件 （0.43 秒）

thispersondoesnotexist.com ▾ このページを訳す

This Person Does Not Exist

This **Person Does Not Exist**.

他の人はこちらも検索

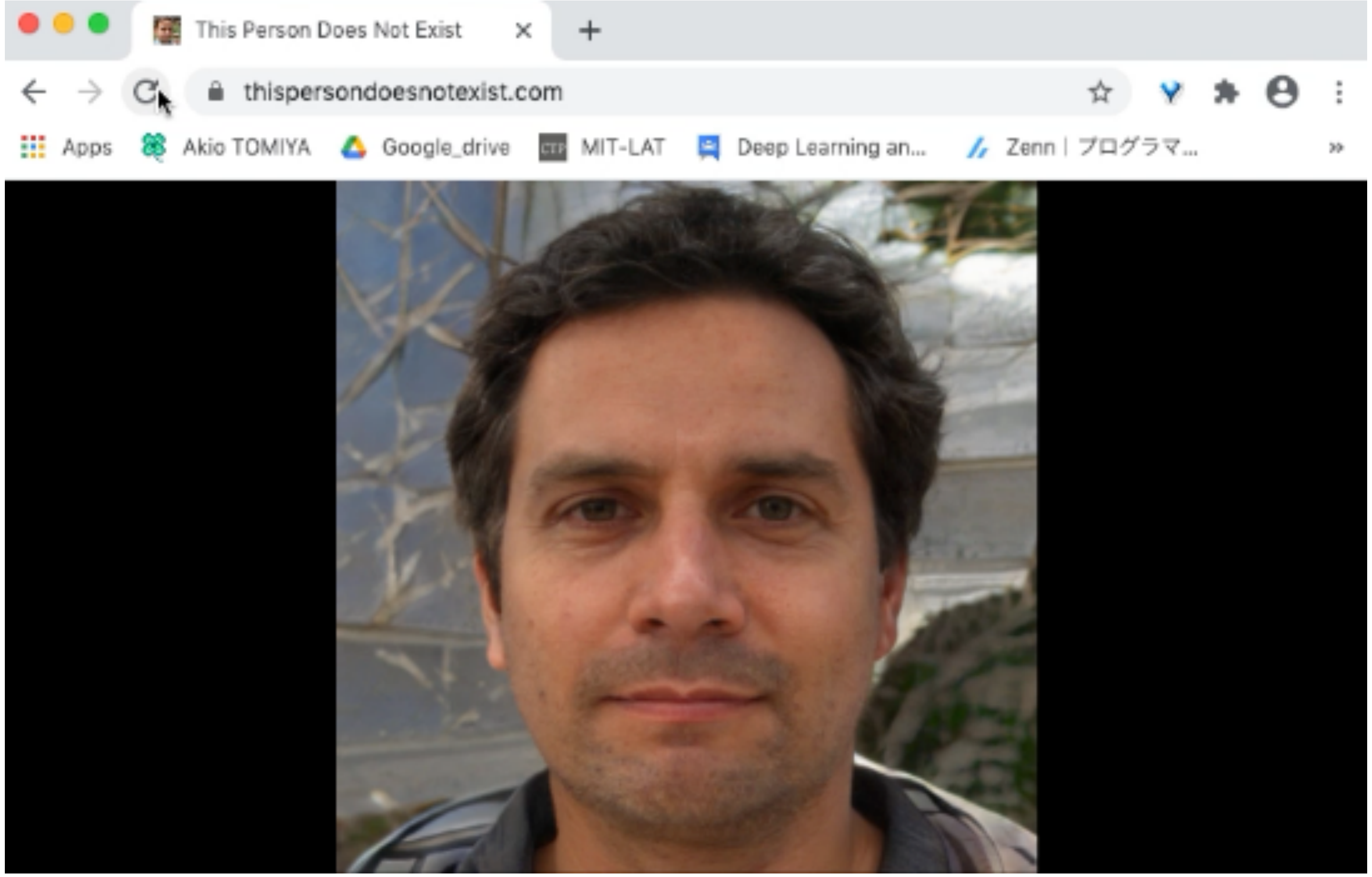
How does this person does not exist work?

Who made this person does not exist?

Introduction

Neural net can make human face images

Neural nets can generate realistic human faces (Style GAN2)

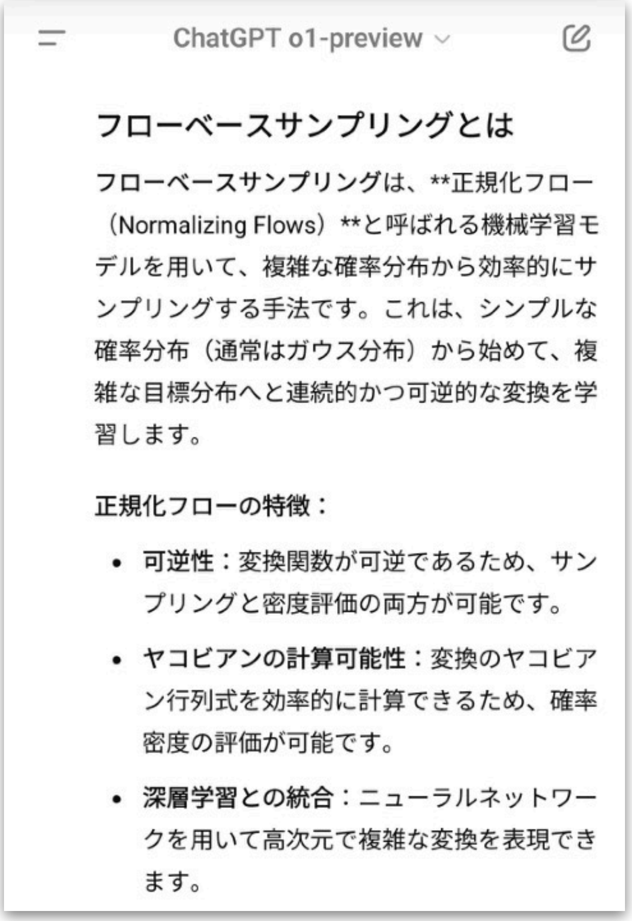


Realistic Images can be generated by machine learning!
Configurations as well? (configuration ~ images?)

ところで...

Open AI o1 出ましたね

- 2024/09/13、ChatGPT の新しいのでましたね。
- 試したところすごそう。Transformer すごい。

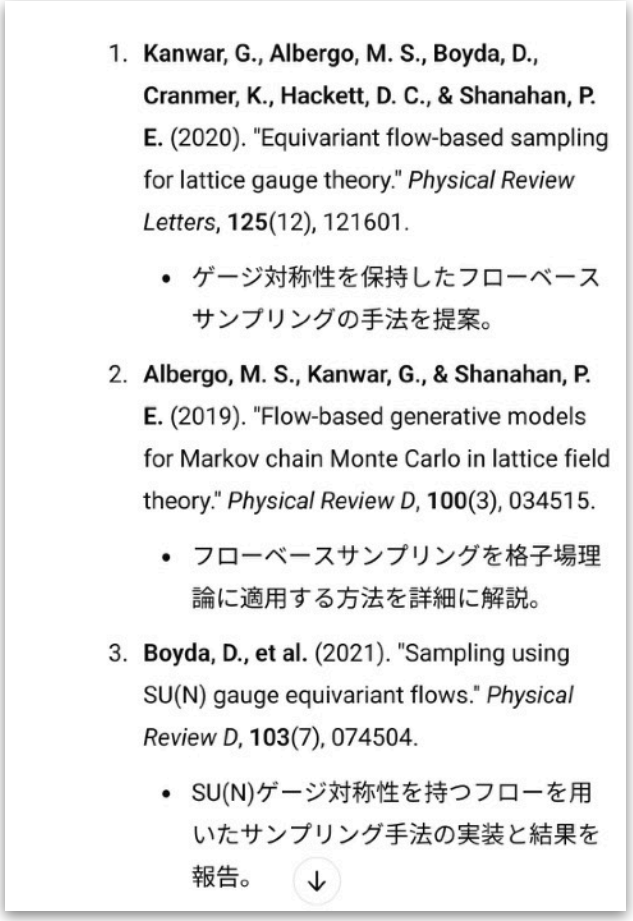


フローベースサンプリングとは

フローベースサンプリングは、**正規化フロー（Normalizing Flows）**と呼ばれる機械学習モデルを用いて、複雑な確率分布から効率的にサンプリングする手法です。これは、シンプルな確率分布（通常はガウス分布）から始めて、複雑な目標分布へと連続的かつ可逆的な変換を学習します。

正規化フローの特徴：

- **可逆性**：変換関数が可逆であるため、サンプリングと密度評価の両方が可能です。
- **ヤコビアンの計算可能性**：変換のヤコビアン行列式を効率的に計算できるため、確率密度の評価が可能です。
- **深層学習との統合**：ニューラルネットワークを用いて高次元で複雑な変換を表現できます。

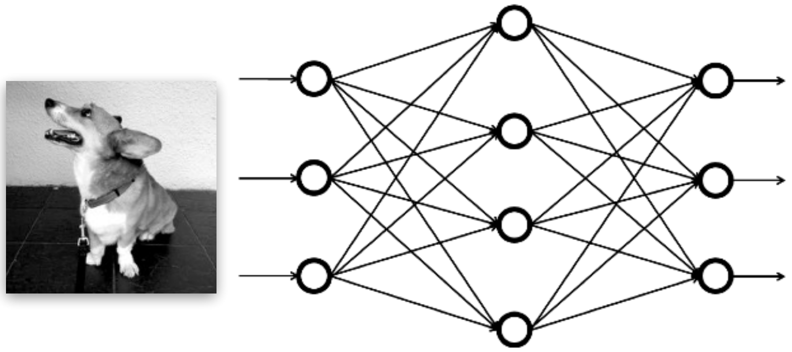


1. **Kanwar, G., Albergo, M. S., Boyda, D., Cranmer, K., Hackett, D. C., & Shanahan, P. E.** (2020). "Equivariant flow-based sampling for lattice gauge theory." *Physical Review Letters*, **125**(12), 121601.
   - ゲージ対称性を保持したフローベースサンプリングの手法を提案。
2. **Albergo, M. S., Kanwar, G., & Shanahan, P. E.** (2019). "Flow-based generative models for Markov chain Monte Carlo in lattice field theory." *Physical Review D*, **100**(3), 034515.
   - フローベースサンプリングを格子場理論に適用する方法を詳細に解説。
3. **Boyda, D., et al.** (2021). "Sampling using SU(N) gauge equivariant flows." *Physical Review D*, **103**(7), 074504.
   - SU(N)ゲージ対称性を持つフローを用いたサンプリング手法の実装と結果を報告。

https://x.com/TomiyaAkio/status/1834389267097006474/photo/1

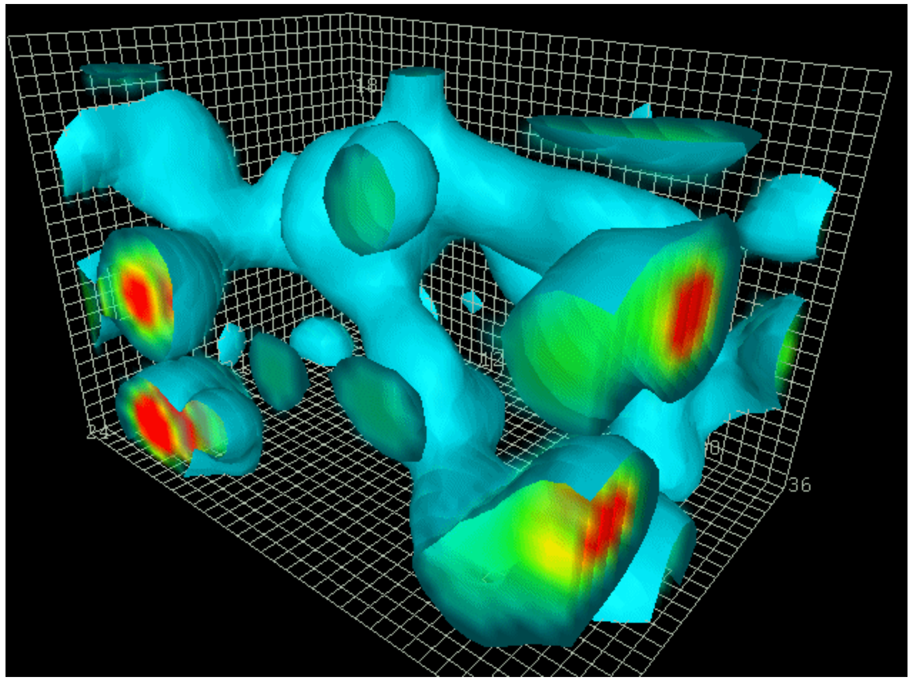
ML for LQCD is needed

- Machine learning/ Neural networks
  - data processing techniques for 2d/3d data in the real world (pictures)
  - (Variational) Approximation (∼ fitting)



$\begin{pmatrix}1\\0\\0\end{pmatrix}$ = **Dog**

- Lattice QCD is more complicated than pictures
  - 4 dimension
  - Non-abelian gauge symmetry
  - Fermions
  - Exactness is necessary
- Q. How can we deal with?



http://www.physics.adelaide.edu.au/theory/staff/leinweber/VisualQCD/QCDvacuum/

Introduction

Configuration generation with machine learning is developing

Year	Group	ML	Dim.	Theory	Gauge sym	Exact?	Fermion?	Lattice2021/ref
2017	AT+	RBM + HMC	2d	Scalar	-	No	No	arXiv: 1712.03893
2018	K. Zhou+	GAN	2d	Scalar	-	No	No	arXiv: 1810.12879
2018	J. Pawlowski +	GAN +HMC	2d	Scalar	-	Yes?	No	arXiv: 1811.03533
2019	MIT+	Flow	2d	Scalar	-	Yes	No	arXiv: 1904.12072
2020	MIT+	Flow	2d	U(1)	Equivariant	Yes	No	arXiv: 2003.06413
2020	MIT+	Flow	2d	SU(N)	Equivariant	Yes	No	arXiv: 2008.05456
2020	AT+	SLMC	**4d**	SU(N)	Invariant	Yes	Partially	arXiv: 2010.11900
2021	M. Medvidovic´+	A-NICE	2d	Scalar	-	No	No	arXiv: 2012.01442
2021	S. Foreman	L2HMC	2d	U(1)	Yes	Yes	No	
2021	**AT+**	SLHMC	**4d**	QCD	Covariant	Yes	YES!	
2021	L. Del Debbio+	Flow	2d	Scalar, O(N)	-	Yes	No	
2021	MIT+	Flow	2d	Yukawa	-	Yes	Yes	
2021	**S. Foreman, AT+**	Flowed HMC	2d	U(1)	Equivariant	Yes	No but compatible	arXiv: 2112.01586
2021	XY Jing	Neural net	2d	U(1)	Equivariant	Yes	No	
2022	J. Finkenrath	Flow	2d	U(1)	Equivariant	Yes	Yes (diagonalization)	arxiv: 2201.02216
2022	MIT+	Flow	2d	U(1)	Equivariant	Yes	Yes (diagonalization)	arXiv:2202.11712

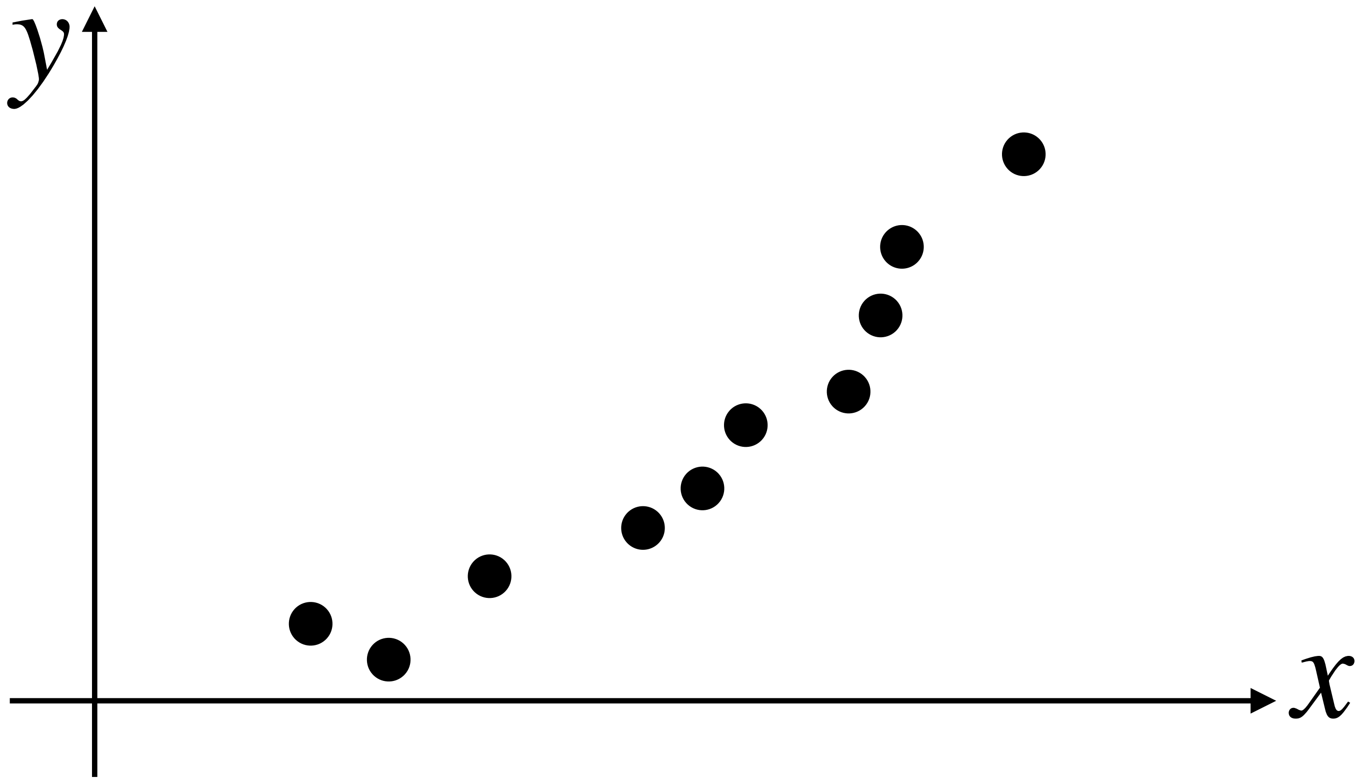
+…

ニューラルネット

What is machine learning?

E.g. Linear regression ∈ Supervised learning

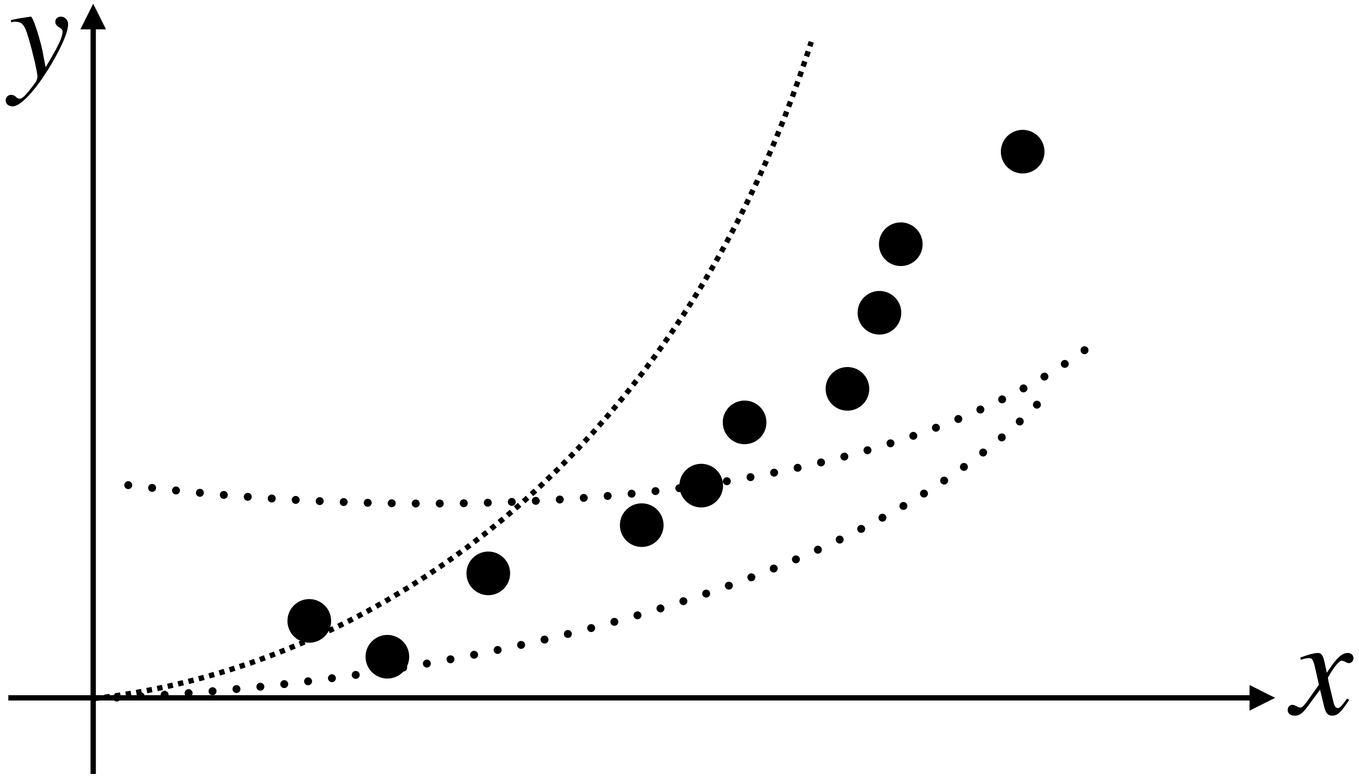
Data: $D = \left\{ (x^{(1)}, y^{(1)}),\ (x^{(2)}, y^{(2)}),\ \cdots \right\}$



What is machine learning?

E.g. Linear regression ∈ Supervised learning

Data: $D = \left\{ (x^{(1)}, y^{(1)}),\ (x^{(2)}, y^{(2)}),\ \cdots \right\}$



$f_{\{a,b,c\}}(x) = ax^2 + bx + c$

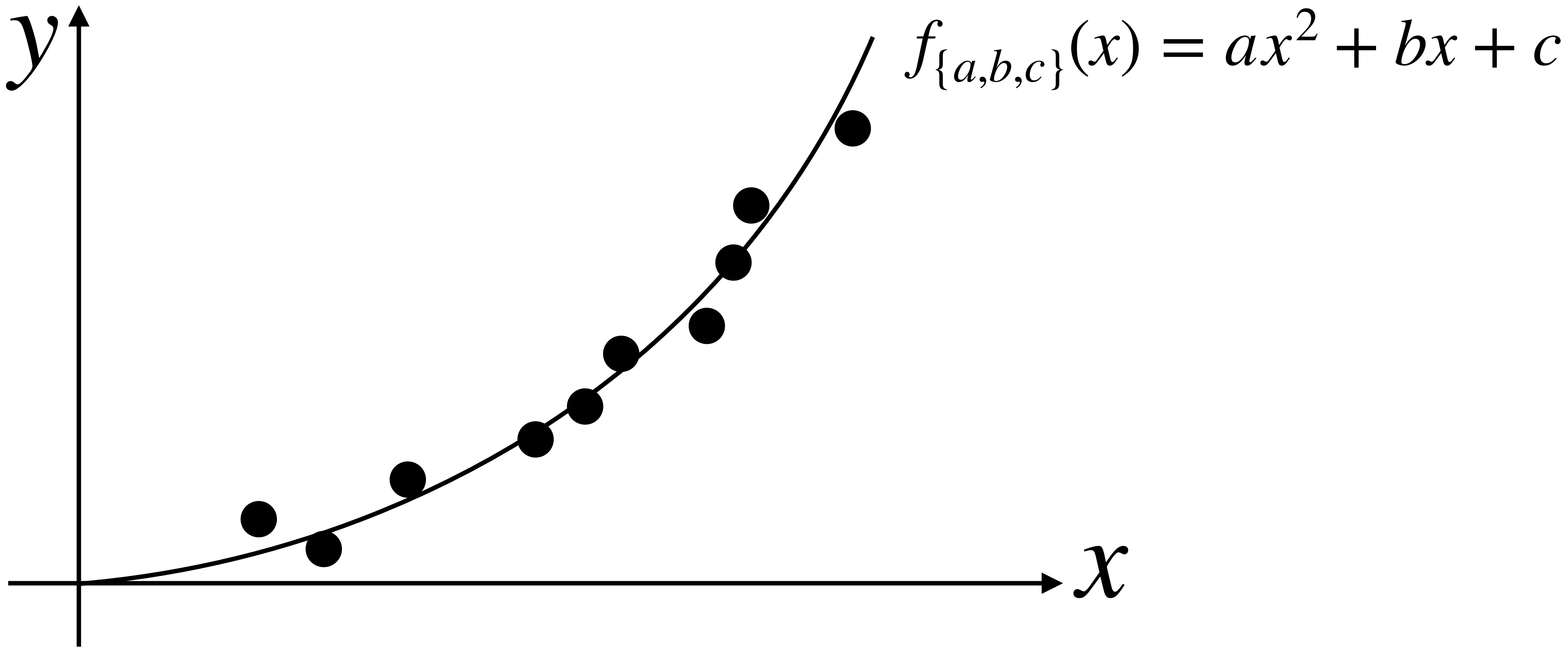
$$E = \frac{1}{2} \sum_d \left| f_{\{a,b,c\}}(x^{(d)}) - y^{(d)} \right|^2$$

$a, b, c,$ are determined by minimizing E
(training = fitting by data)

What is machine learning?

E.g. Linear regression ∈ Supervised learning

Data: $D = \left\{(x^{(1)}, y^{(1)}),\ (x^{(2)}, y^{(2)}),\ \cdots\right\}$



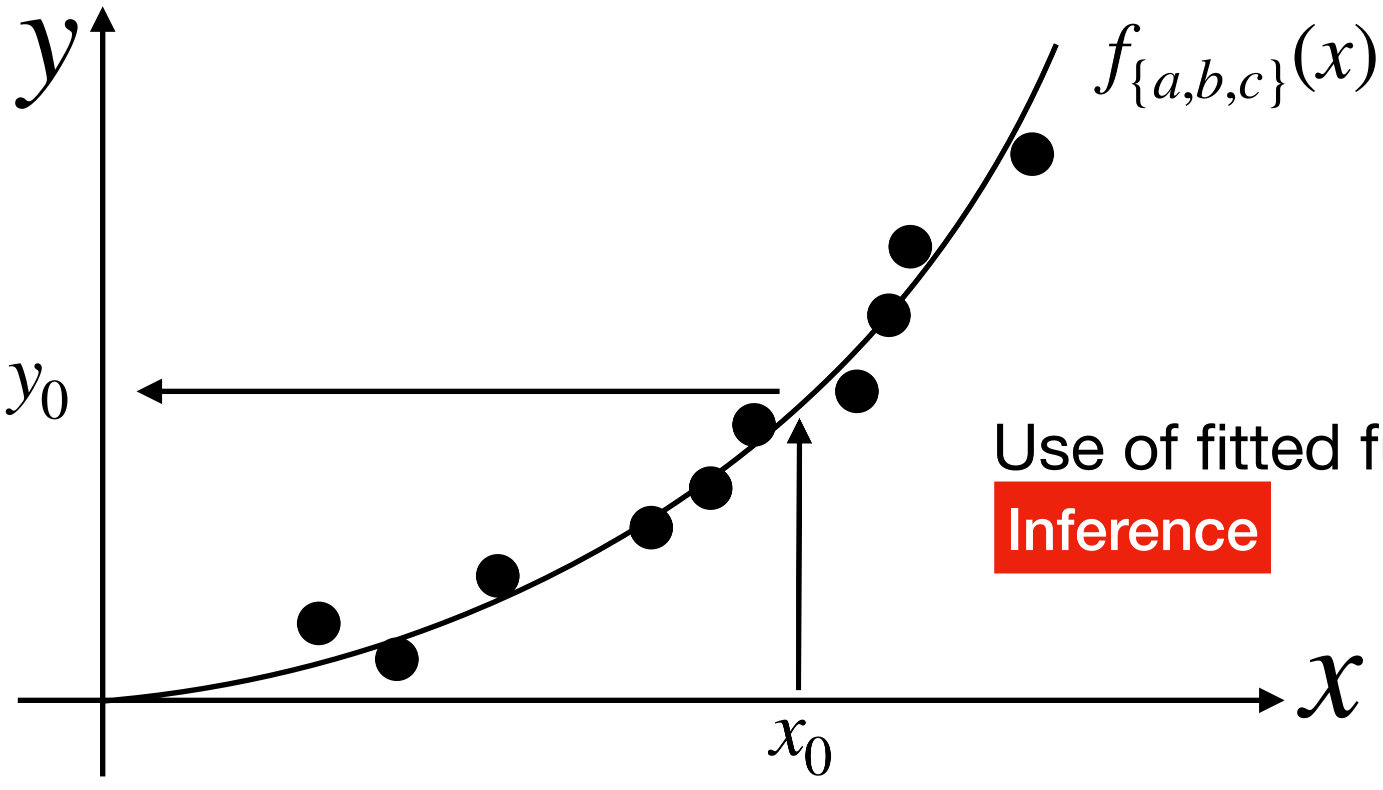
$f_{\{a,b,c\}}(x) = ax^2 + bx + c$ $\qquad E = \dfrac{1}{2}\sum_d \left| f_{\{a,b,c\}}(x^{(d)}) - y^{(d)} \right|^2$

$a, b, c,$ are determined by minimizing E
(training = fitting by data)

What is machine learning?

E.g. Linear regression ∈ Supervised learning

Data: $D = \left\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots\right\}$



$$f_{\{a,b,c\}}(x) = ax^2 + bx + c$$

Use of fitted function = **Inference**

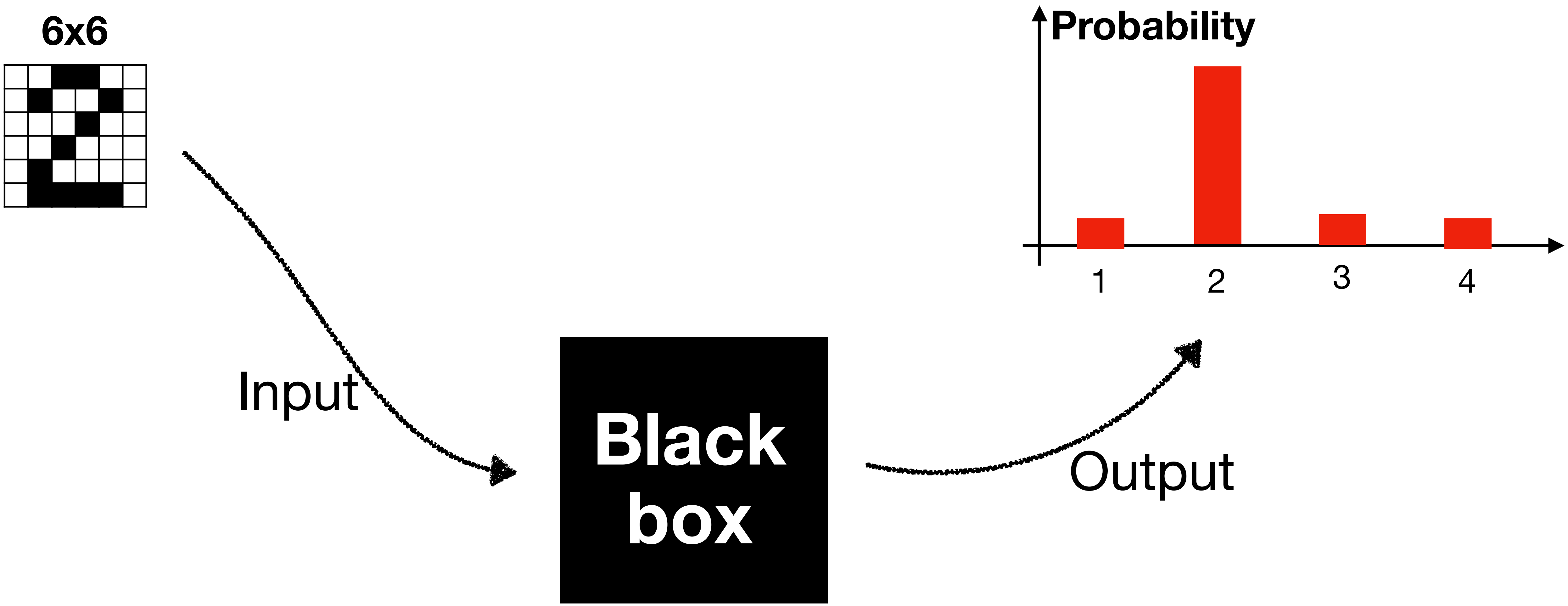
Now we can predict y value which not in the data

In physics language, variational method

What is the neural networks?

Neural network is a *universal* approximation function

Example: Recognition of hand-written numbers (0-9)



6x6

Input

Black box

Output

How can we formulate this “Black box”?
Ansatz?

What is the neural networks?

Neural network is a *universal* approximation function

Example: Recognition of hand-written numbers (0-9)

6x6

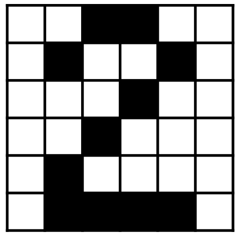
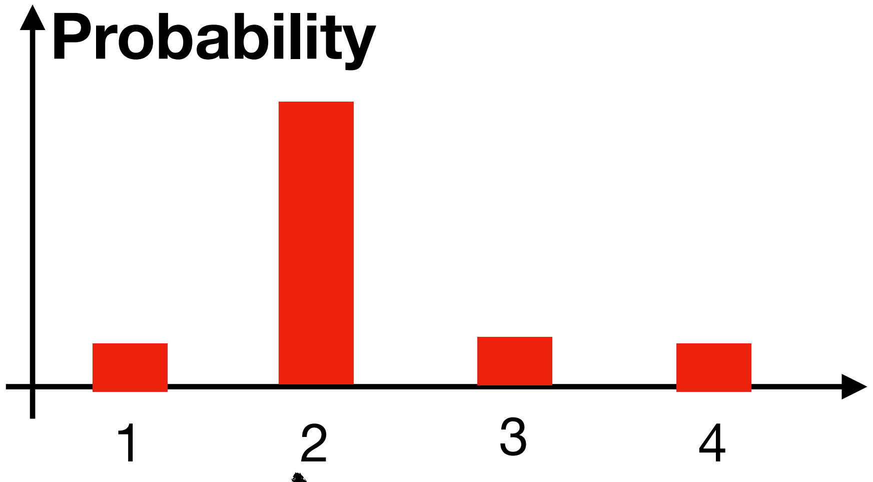


Image is a vector (6x6=36 dim)

$$= \begin{pmatrix} 0.000 \\ 0.000 \\ 0.8434 \\ 0.756 \\ 0.3456 \\ 0.64 \\ 0.251 \\ \vdots \end{pmatrix} = \vec{x}$$

Regard



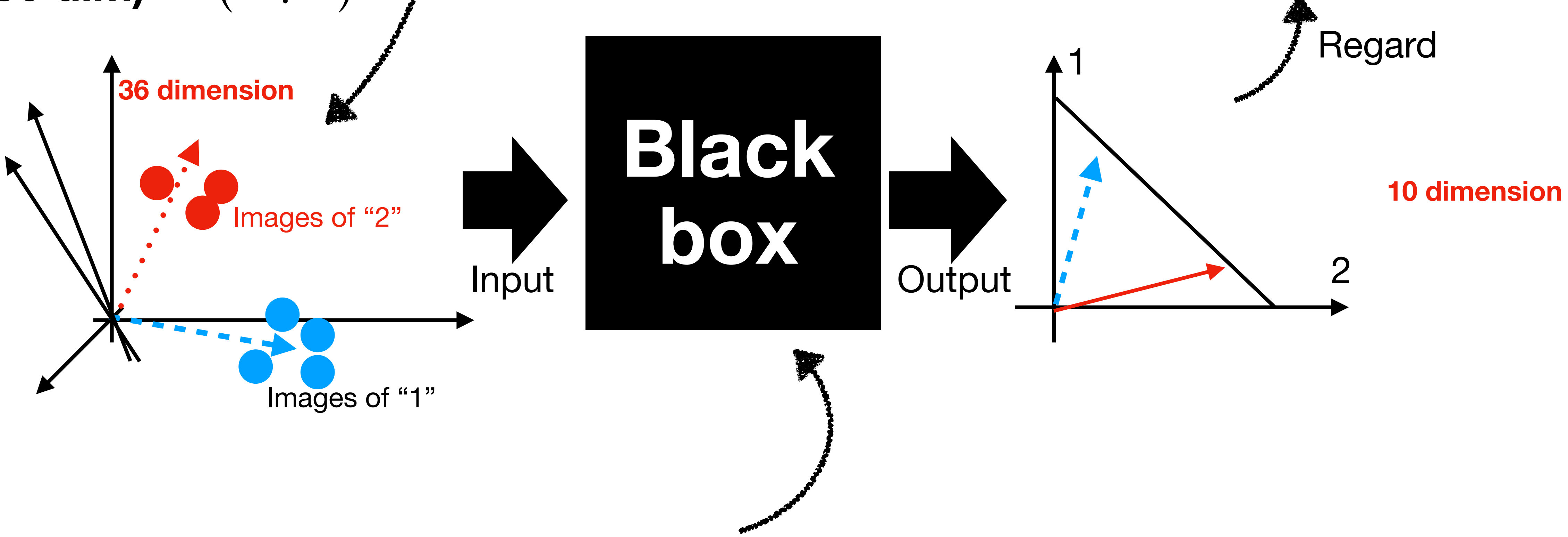


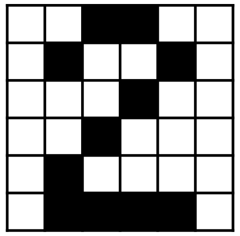
Image recognition = Find a map between two vector spaces

What is the neural networks?

Neural network is a *universal* approximation function

Example: Recognition of hand-written numbers (0-9)

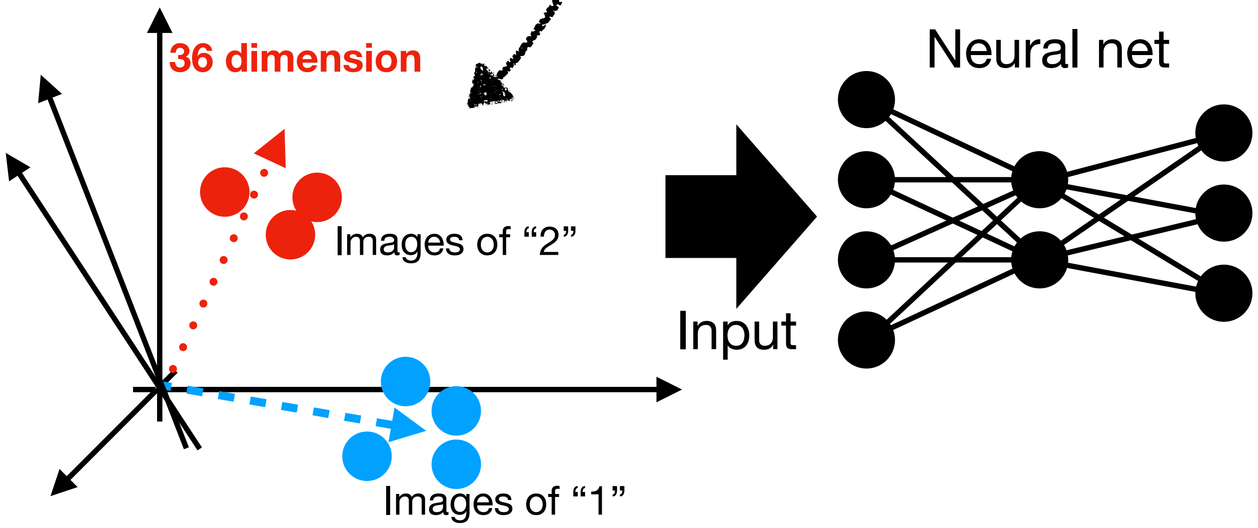
6x6



$$= \begin{pmatrix}0.000\\0.000\\0.8434\\0.756\\0.3456\\0.64\\0.251\\\vdots\end{pmatrix} = \vec{x}$$

Image is a vector (6x6=36 dim)

Regard



What is the neural networks?

Affine transformation + element-wise transformation

Layers of neural nets $l = 2,3,\cdots, L,\ \vec{u}^{(1)} = \vec{x}$ $W^l,\ \vec{b}^{(l)}$ are fit parameters

$$\begin{cases} \vec{z}^{(l)} = W^{(l)}\vec{u}^{(l-1)} + \vec{b}^{(l)} \\ u_i^{(l)} = \sigma^{(l)}(z_i^{(l)}) \end{cases}$$

Affine transformation
(b=0 called linear transformation)

Element-wise (local) non-linear.
hyperbolic tangent-ish function

A fully connected neural net = composite function (Linear&non-linear)

$$f_\theta(\vec{x}) = \sigma^{(3)}(W^{(3)}\sigma^{(2)}(W^{(2)}\vec{x} + \vec{b}^{(2)}) + \vec{b}^{(3)})$$

θ is a set of parameters: $w_{ij}^{(l)}, b_i^{(l)}, \cdots$

- Input = vectors, output = vectors
- Neural net = a nested function with a lot of parameters (W, b)
- Parameters (W, b) are determined from data (fitting/training)

Neural network = map between vectors and vectors

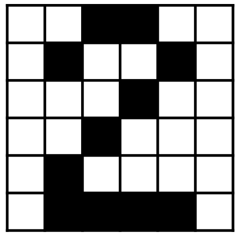
Physicists terminology: Variational ansatz

What is the neural networks?

Neural network is a *universal* approximation function

Example: Recognition of hand-written numbers (0-9)

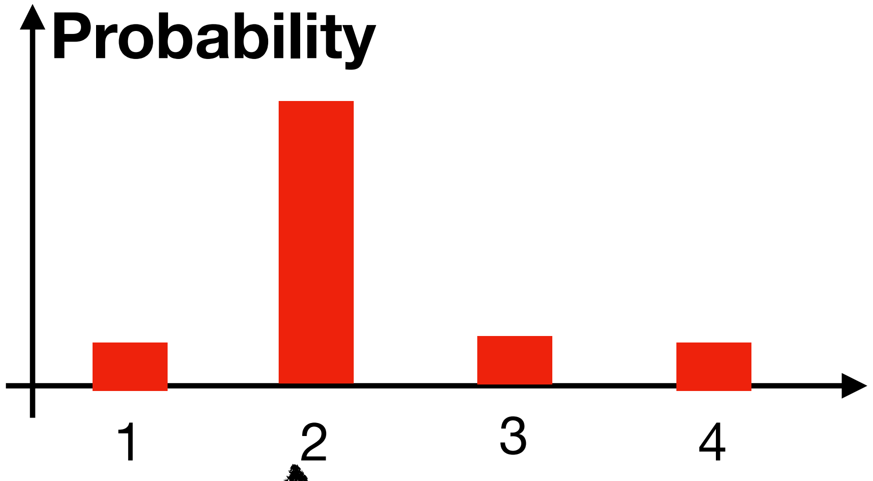
6x6

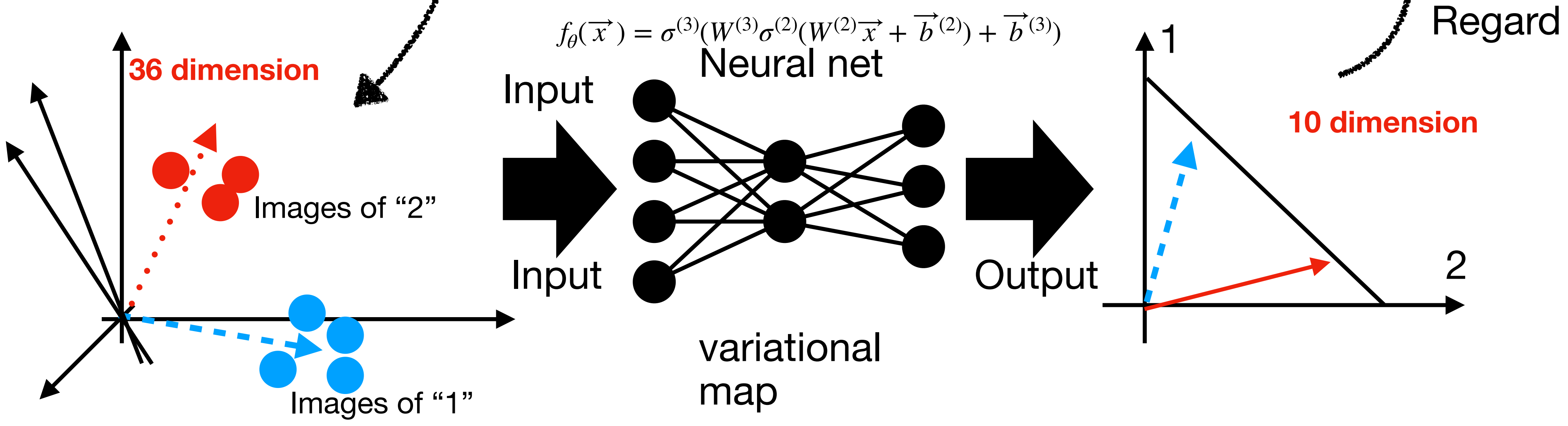


$$= \begin{pmatrix} 0.000 \\ 0.000 \\ 0.8434 \\ 0.756 \\ 0.3456 \\ 0.64 \\ 0.251 \\ \vdots \end{pmatrix} = \vec{x}$$

Image is a vector (6x6=36 dim)

Regarding

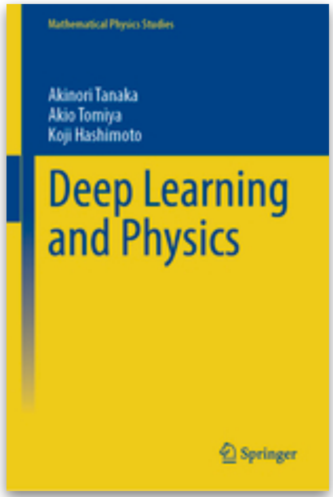



$f_\theta(\vec{x}) = \sigma^{(3)}(W^{(3)}\sigma^{(2)}(W^{(2)}\vec{x} + \vec{b}^{(2)}) + \vec{b}^{(3)})$
36 dimension
Images of "2"
Images of "1"
Input
Input
Neural net
variational map
Output
1
2
10 dimension
Regard

"0" = (1,0,0,…)
"1" = (0,1,0,…)
"2" = (0,0,1,…)
…
"9" = (0,0,…,1)

Fact: Neural network can mimic any function = A systematic variational function.

In this example, NN mimics image (36-dim vector) and label (10-dim vector)

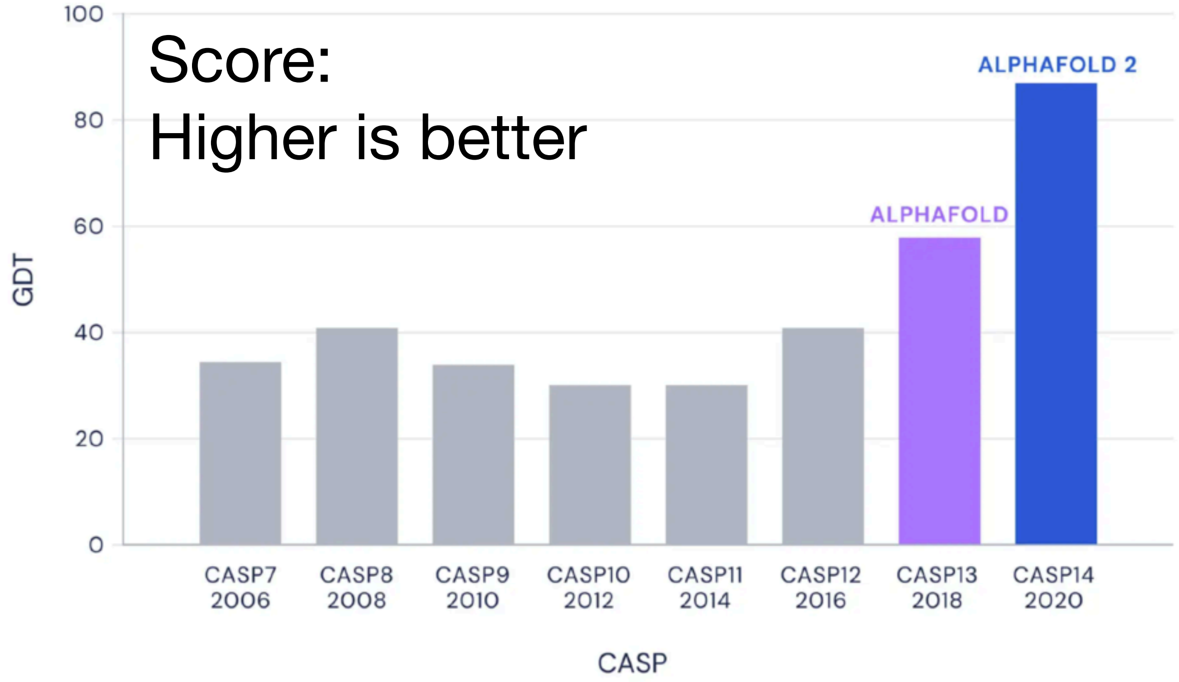


What is the neural networks?

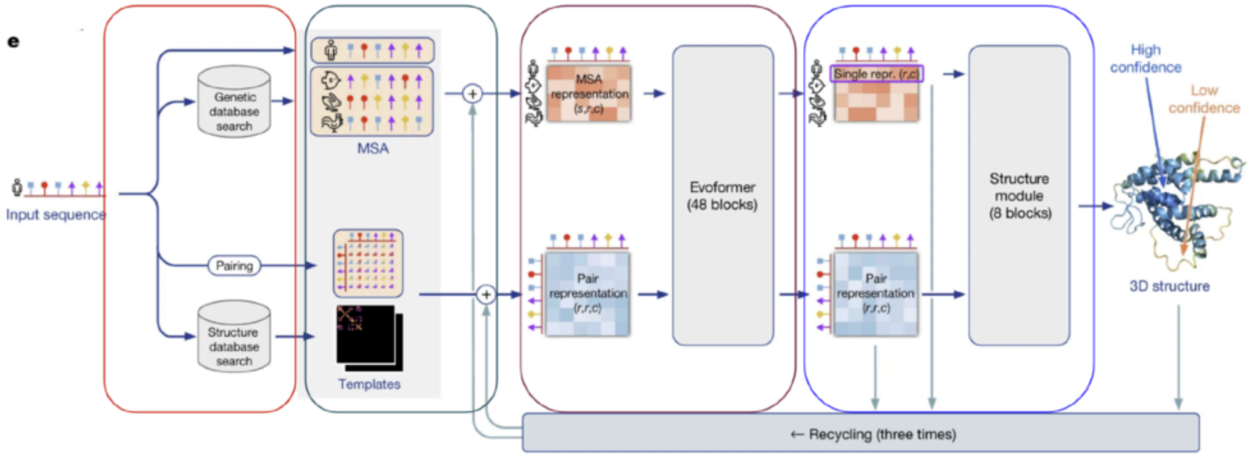
Akio Tomiya

Neural network have been good job

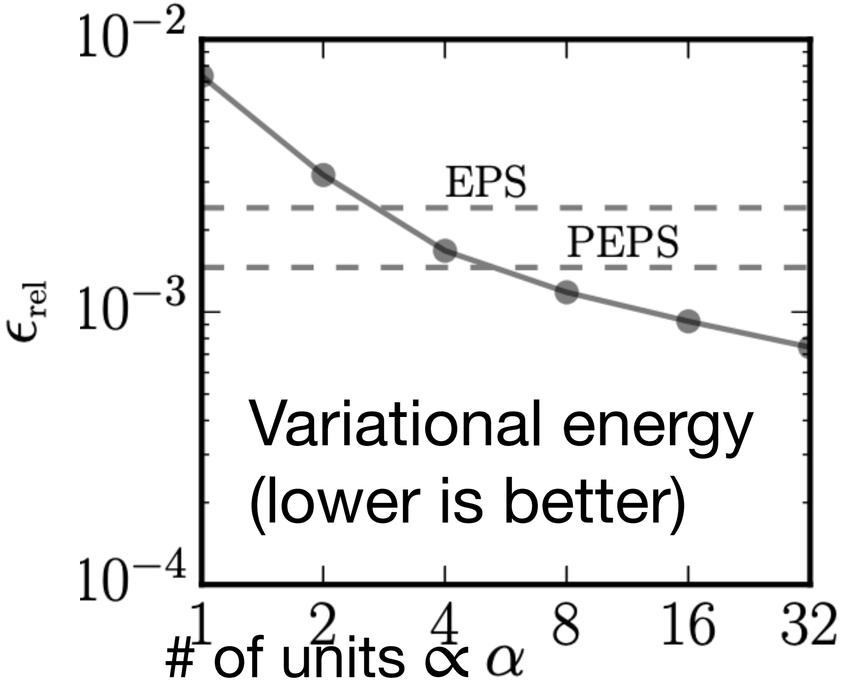
Protein Folding (AlphaFold2, John Jumper+, Nature, 2020+), Transformer neural net



Score:
Higher is better

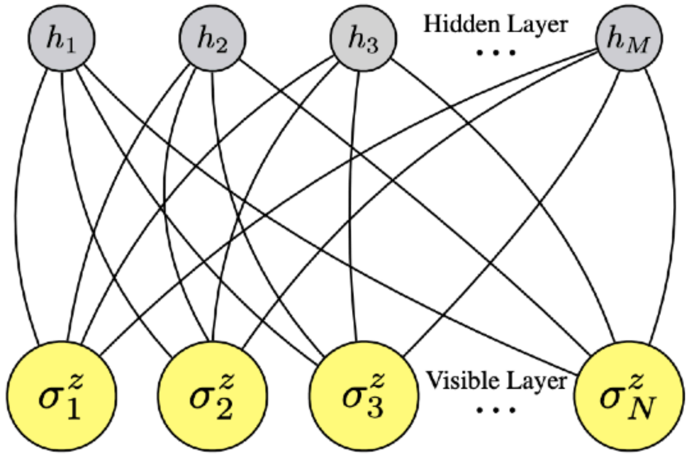

e
Input sequence
Genetic database search
Pairing
Structure database search
MSA
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)

Neural network wave function for many body (Carleo Troyer, Science 355, 602 (2017))



Variational energy
(lower is better)

# of units ∝ α



Neural net + “Expert knowledge” → Best performance

https://horomary.hatenablog.com/entry/2021/10/01/194825

https://www.aist.go.jp/aist_j/press_release/pr2020/pr20201111/pr20201111.html

What is the neural networks?

サロゲートモデル

- ニューラルネットは万能な関数近似
- 微分方程式 (運動方程式)の解を模倣させればよさそう。
  - = サロゲートモデル
  - 真面目に解かず、解を表現。初期条件に関する関数として近似
  - (偏微分方程式なら境界条件)
- 真面目に解くより速い(場合もある)
  - 物理系の特性を入れていて変なことが起こらないようにする。

フローベース法

(配位生成法)

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

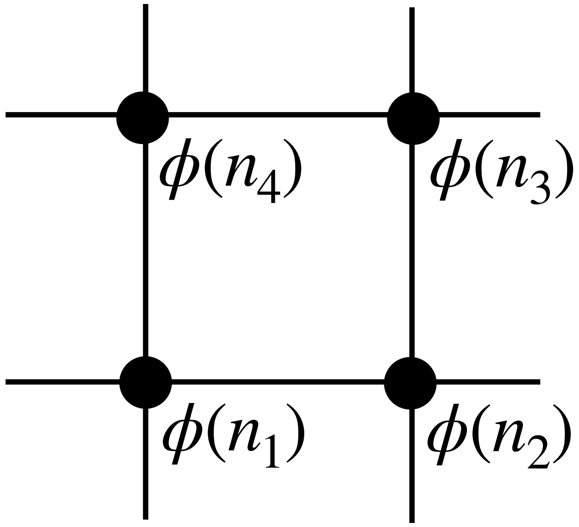
フローベース法

経路積分は難しい

$$\int D\phi e^{-S[\phi]} O[\phi]$$

$$D\phi = \prod_{n \in \mathrm{lat}} \mathrm{d}\phi(n)$$

格子上のスカラー場



経路積分 = 多次元の積分

L = 10 -> L^4で1万次元の積分

□ 部分積分

☑ 置換積分

いい感じに置換積分できないか?

(正確にはサンプリングに有利に

なるように変数変換できないか)

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

変数変換で積分・サンプリングが簡単になる例

ボックス・ミュラー法

$$\begin{cases} z = e^{-\frac{1}{2}(x^2+y^2)} \\ \tan\theta = y/x \end{cases}$$

変数変換

$$\int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy \, e^{-\frac{1}{2}x^2 - \frac{1}{2}y^2} = \frac{1}{2} \underline{\int_0^{2\pi} d\theta \int_0^1 dz}$$

難しい積分

簡単な積分(重みが平ら)

変数変換を行うと、積分を簡単化することができる(右辺は長方形の面積)

右辺を使うと、サンプリングも簡単
長方形から一様にサンプル

$$\begin{cases} \xi_1 \sim (0,2\pi) \\ \xi_2 \sim (0,1) \end{cases}$$

ガウス乱数 x, y
は変数変換で構成できる

$$\begin{cases} x = r\cos\theta & \theta = \xi_1 \\ y = r\sin\theta & r = \sqrt{-2\log\xi_2} \end{cases}$$

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

置換積分(変数変換)

$$\int D\phi e^{-S[\phi]} O[\phi] = \int Dz \underbrace{\left| \det \frac{\partial \phi}{\partial z} \right|}_{=\text{Jacobian}=J} e^{-S[\phi[z]]} O[\phi[z]]$$

経路積分

$z(n)$は変数変換した場

$$S_{\text{eff}}[z] = S[\phi[z]] - \log J[z]$$

$$= \int Dz e^{-S_{\text{eff}}[z]} O[\phi[z]] \qquad Dz = \prod_{n \in \text{lat}} \text{d}z(n)$$

もし、こいつに運動項$(\partial\phi)^2$がなければ積分は劇的に簡単

-> 自明化写像 (Trivializing map)

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

置換積分(変数変換)

$$\int D\phi e^{-S[\phi]} O[\phi] = \int Dz e^{-S_{\text{eff}}[z]} O[\phi[z]]$$

経路積分　　　　**変数変換後(Trivial)**

仮に、 $S_{\text{eff}}[z] = "\sum_n \phi^2(n)"$ となってたとする。

$$\int Dz e^{-S_{\text{eff}}[z]} O[\phi[z]] = \int \prod_{n \in \text{lat}} \mathrm{d}z(n)\ e^{-S_{\text{eff}}[z]} O[\phi[z]]$$

$$= \int \prod_{n \in \text{lat}} \mathrm{d}z(n)\ e^{-\sum_n \phi^2(n)} O[\phi[z]]$$

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

クロスタームなし、変数を独立に積分できる

フローベース法

置換積分(変数変換)

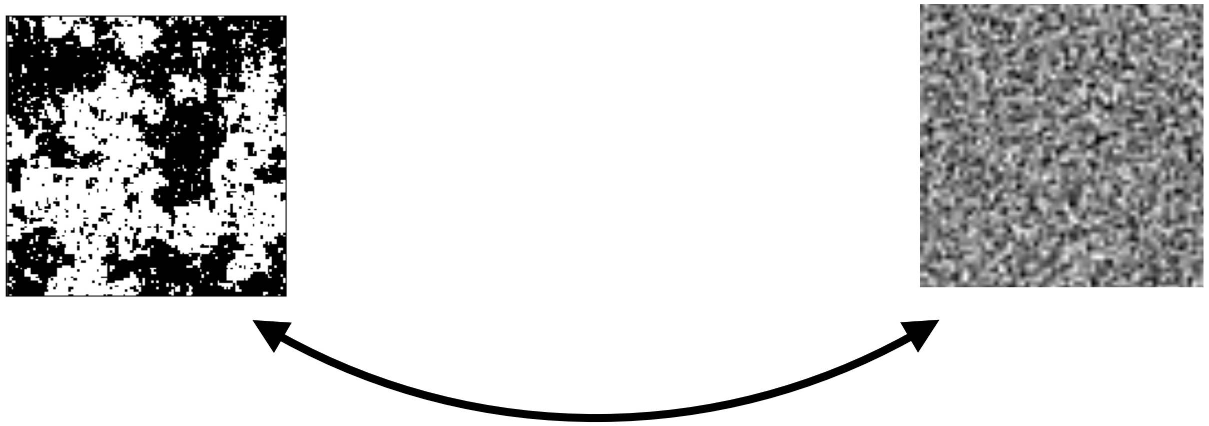
$$\int D\phi e^{-S[\phi]} O[\phi] = \int Dz e^{-S_{\text{eff}}[z]} O[\phi[z]]$$

場の理論

変数変換後(Trivial)

異なる点に相関

異なる点は無相関



これをつなぐ変換?

存在するかは一旦棚上げし、具体的に
変数変換がどうなってるのかを見てみる

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

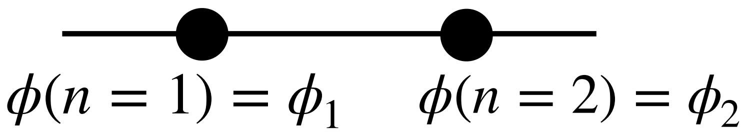
置換積分(変数変換)

$$\int D\phi e^{-S[\phi]} O[\phi] = \int Dz e^{-S_{\rm eff}[z]} O[\phi[z]]$$

場の理論　　　　**変数変換後(Trivial)**

多変数関数で変数変換(すべての座標で変数(場)を変換)

例として2点しかない格子上の場の理論を考えてみる。



$$S = (\phi_1 - \phi_2)^2 + \phi_1^2 + \phi_2^2 + \cdots$$

高次の項

$$\int \mathrm{d}\phi_1 \mathrm{d}\phi_2 e^{-S[\phi]} O(\phi_1, \phi_2)$$

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

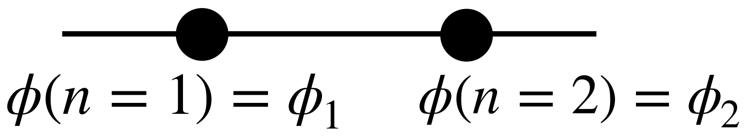
置換積分(変数変換)

$$\int D\phi e^{-S[\phi]}O[\phi] = \int Dz e^{-S_{\text{eff}}[z]}O[\phi[z]]$$

場の理論　　**変数変換後(Trivial)**

多変数関数で変数変換(すべての座標で変数(場)を変換)

例として2点しかない格子上の場の理論を考えてみる。



$$S = (\phi_1 - \phi_2)^2 + \phi_1^2 + \phi_2^2 + \underbrace{\cdots}_{\text{高次の項}}$$

$$\int \mathrm{d}\phi_1 \mathrm{d}\phi_2 e^{-S[\phi]}O(\phi_1, \phi_2)$$

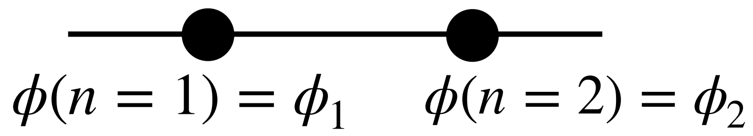
変数変換

$$\begin{cases} z_1 = f_1(\phi_1, \phi_2) \\ z_2 = f_2(\phi_1, \phi_2) \end{cases}$$

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

置換積分(変数変換)



$$S = (\phi_1 - \phi_2)^2 + \phi_1^2 + \phi_2^2 + \cdots$$

変数変換
$$\begin{cases} z_1 = f_1(\phi_1, \phi_2) \\ z_2 = f_2(\phi_1, \phi_2) \end{cases}$$

変数変換のヤコビアン

$$\begin{bmatrix} \mathrm{d}z_1 \\ \mathrm{d}z_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \dfrac{\partial f_1}{\partial \phi_1} & \dfrac{\partial f_1}{\partial \phi_2} \\ \dfrac{\partial f_2}{\partial \phi_1} & \dfrac{\partial f_2}{\partial \phi_2} \end{bmatrix}}_{= J} \begin{bmatrix} \mathrm{d}\phi_1 \\ \mathrm{d}\phi_2 \end{bmatrix}$$

$$\mathrm{d}z_1 \mathrm{d}z_2 = \det J \mathrm{d}\phi_1 \mathrm{d}\phi_2$$

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

置換積分(変数変換)

$$S_{\mathrm{eff}}[z] = S[\phi[z]] - \log J[z]$$

$$\int D\phi e^{-S[\phi]} O[\phi] = \int Dz e^{-S_{\mathrm{eff}}[z]} O[\phi[z]]$$

これを繰り返せば、4次元時空の場の理論も変数変換できる

格子ゲージ理論でやってみると、自明化できる (Luscher、存在証明)

→ 第一次近似はGradient flow (場の拡散方程式)

しかし、det J の評価は、O((L^4)^3) = O(L^12) かかる。

これは実際上、無理。

M. Luscher arXiv:0907.5491

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

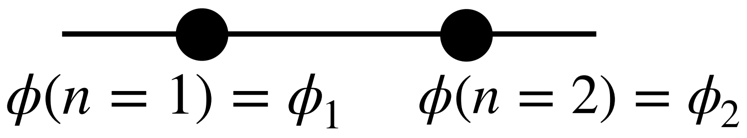
ここまでのまとめ

- 格子場の理論での経路積分は多重積分
- 積分の変数変換は、ヤコビアンを生む
  - 座標依存性が消せる (かも)
  - (証明は飛ばすが、実はtrivializing map の存在を証明できる)
- とはいえ、ヤコビアンの計算が大変なので実用的でない
- 機械学習でなんとかしましょう。

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

賢い変数変換 (シンプレクティック積分と同様)



$$S=(\phi_1-\phi_2)^2+\phi_1^2+\phi_2^2+\cdots$$

変数変換(even)
$$\begin{cases}\phi_1'=\phi_1\\ \phi_2'=\mathrm{e}^{s(\phi_1)}\phi_2\end{cases}$$

$s(\phi_1)$ **は適当な関数**
(ニューラルネット)

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

賢い変数変換 (シンプレクティック積分と同様)

Akio Tomiya

$\phi(n=1)=\phi_1$ $\phi(n=2)=\phi_2$

$$S=(\phi_1-\phi_2)^2+\phi_1^2+\phi_2^2+\cdots$$

変数変換(even)
$$\begin{cases}\phi_1'=\phi_1\\ \phi_2'=\mathrm{e}^{s(\phi_1)}\phi_2\end{cases}$$

$s(\phi_1)$ は適当な関数
(ニューラルネット)

$$\begin{bmatrix}\mathrm{d}\phi_1'\\ \mathrm{d}\phi_2'\end{bmatrix}=\begin{bmatrix}\frac{\partial\phi_1'}{\partial\phi_1} & \frac{\partial\phi_1'}{\partial\phi_2}\\ \frac{\partial\phi_2'}{\partial\phi_1} & \frac{\partial\phi_2'}{\partial\phi_2}\end{bmatrix}\begin{bmatrix}\mathrm{d}\phi_1\\ \mathrm{d}\phi_2\end{bmatrix}=\begin{bmatrix}1 & 0\\ * & \mathrm{e}^{s(\phi_1)}\end{bmatrix}\begin{bmatrix}\mathrm{d}\phi_1\\ \mathrm{d}\phi_2\end{bmatrix}$$

log のヤコビアンが簡単 $\log\det J=\log\det\begin{bmatrix}1 & 0\\ * & \mathrm{e}^{s(\phi_1)}\end{bmatrix}=s(\phi_1)$

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

賢い変数変換 (シンプレクティック積分と同様)

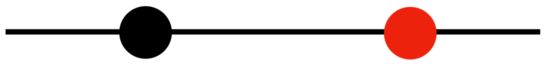
$\phi(n=1)=\phi_1$ $\phi(n=2)=\phi_2$

$$S = (\phi_1 - \phi_2)^2 + \phi_1^2 + \phi_2^2 + \cdots$$

変数変換(even)

$$\phi_1^{(1)} = \phi_1$$

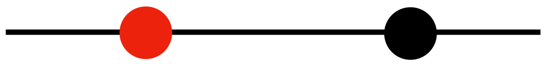
$$\phi_2^{(1)} = \mathrm{e}^{s^{(1)}(\phi_1)}\phi_2$$



変数変換(odd)

$$\phi_1^{(2)} = \mathrm{e}^{s^{(2)}(\phi_2^{(1)})}\phi_1^{(1)}$$

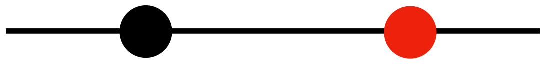
$$\phi_2^{(2)} = \phi_2^{(1)}$$



変数変換(even)

$$\phi_1^{(3)} = \phi_1^{(2)}$$

$$\phi_2^{(3)} = \mathrm{e}^{s^{(3)}(\phi_1^{(2)})}\phi_2^{(2)}$$



…

…

$$z_1 = \phi_1^{(n)}$$

$$z_2 = \phi_2^{(n)}$$

trivial

$$\log\det\left(\frac{\partial(z_1, z_2)}{\partial(\phi_1, \phi_2)}\right) = s^{(1)}(\phi_1^{(1)}) + s^{(2)}(\phi_2^{(1)}) + \cdots$$

ヤコビアンは、関数の値(計算結果)

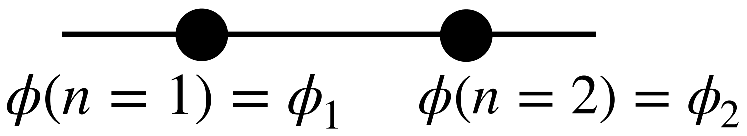
ニューラルネットの出力した数字

だけの足し算で書ける

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

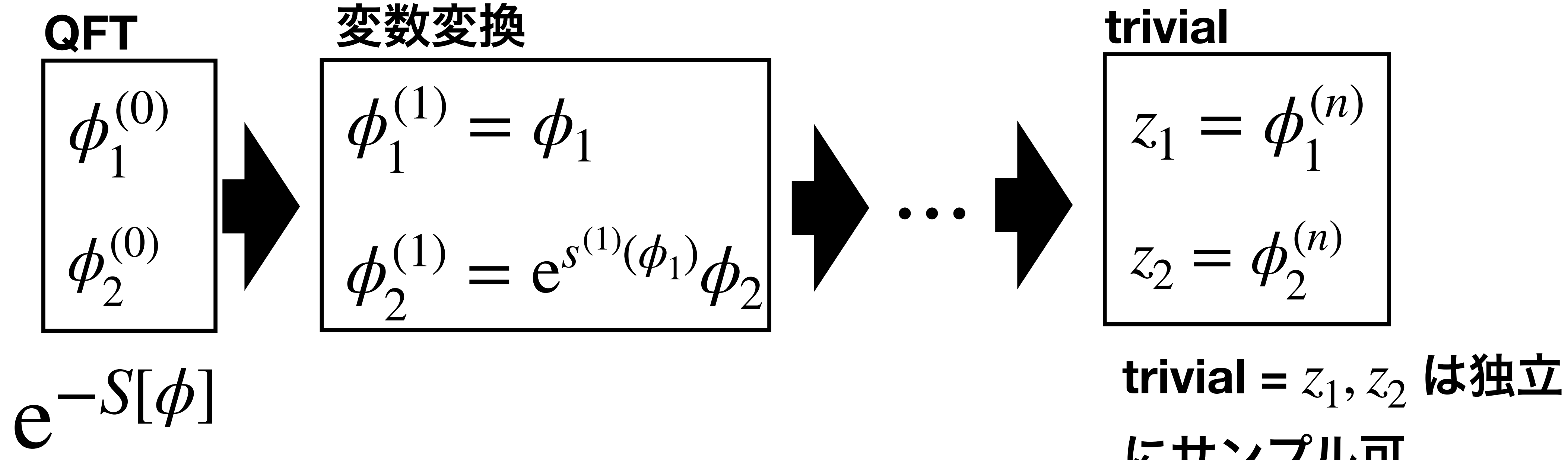
フローベース法

賢い変数変換 (シンプレクティック積分と同様)



$\phi(n=1)=\phi_1$ $\phi(n=2)=\phi_2$

$$S=(\phi_1-\phi_2)^2+\phi_1^2+\phi_2^2+\cdots$$



QFT

$\phi_1^{(0)}$

$\phi_2^{(0)}$

$e^{-S[\phi]}$

変数変換

$\phi_1^{(1)}=\phi_1$

$\phi_2^{(1)}=e^{s^{(1)}(\phi_1)}\phi_2$

$\cdots$

trivial

$z_1=\phi_1^{(n)}$

$z_2=\phi_2^{(n)}$

trivial = z_1, z_2 は独立にサンプル可

逆に、右から出発して左に行けないか?

もし、変数変換が可逆なら可能

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

賢い変数変換 (シンプレクティック積分と同様)

$\phi(n=1)=\phi_1$ $\phi(n=2)=\phi_2$

$$S=(\phi_1-\phi_2)^2+\phi_1^2+\phi_2^2+\cdots$$

変数変換(even)
$$\begin{cases}\phi_1'=\phi_1\\ \phi_2'=\mathrm{e}^{s(\phi_1)}\phi_2\end{cases}$$

$s(\phi_1)$ は適当な関数

※ この変換は、全単射 (逆解きできる)

逆変換(even)
$$\begin{cases}\phi_1=\phi_1'\\ \phi_2=\mathrm{e}^{-s(\phi_1')}\phi_2'\end{cases}$$

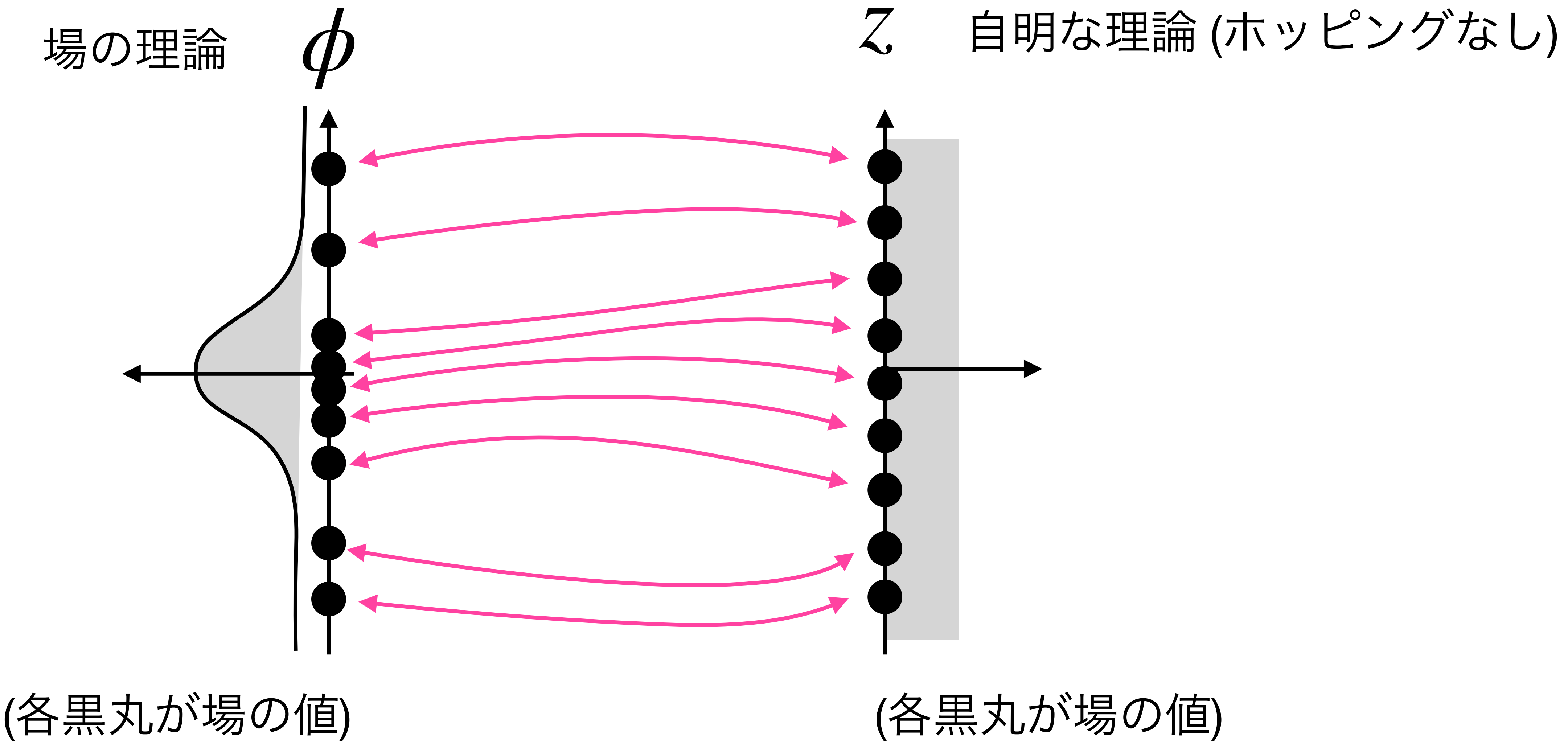
s^{-1} は必要なし

ということで、話を逆転させ、un-trivializing map (非自明化写像)を作る

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

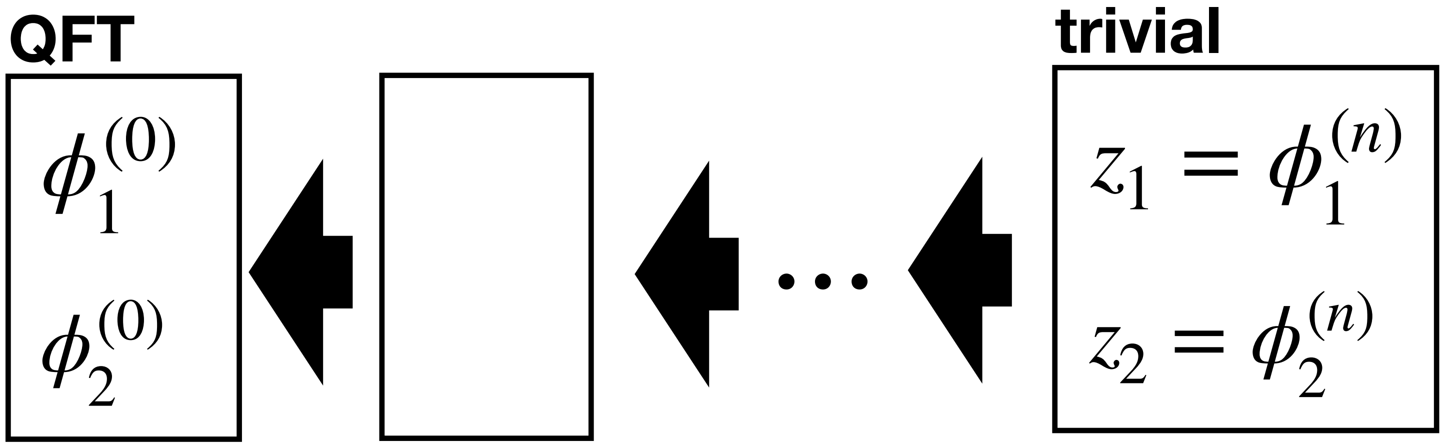
全単射、逆に戻れる　(イメージ図)



フローベース法

賢い変数変換 (シンプレクティック積分と同様)

各点独立に
サンプルで
きない



各点独立に
サンプルできる

$$P[\phi] = \mathrm{e}^{-S[\phi]}$$

$$R[z] \propto \mathrm{e}^{-V_1(z_1)}\mathrm{e}^{-V_2(z_2)}$$

$$Q[\phi] = \mathrm{e}^{-S_{\mathrm{eff}}[\phi[z]]} \equiv J_\theta^{-1}[z]R[z]$$

PとQの「距離」(正確にはKLダイバージェンス)を

最小にするように関数$s^{(k)}$ のパラメータを調整(学習すればよい)

KLダイバージェンス
(カルバック・ライブラー)、相対エントロピー

$$D_{\mathrm{KL}}[P\,||\,Q_\theta] = \int \underline{D\varphi P[\varphi]}\log \frac{P[\varphi]}{Q_\theta[z^{-1}[\varphi]]}$$

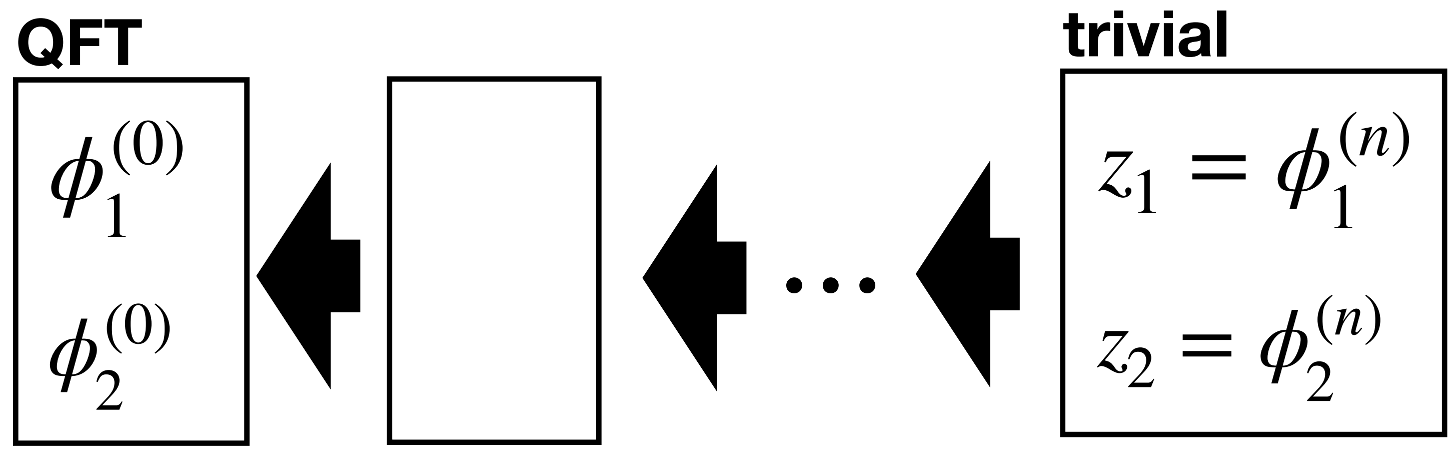
大変な経路積分が残っている

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

賢い変数変換 (シンプレクティック積分と同様)

各点独立に
サンプルで
きない



QFT

$\phi_1^{(0)}$

$\phi_2^{(0)}$

$\cdots$

trivial

$z_1 = \phi_1^{(n)}$

$z_2 = \phi_2^{(n)}$

各点独立に
サンプルできる

$$P[\phi] = \mathrm{e}^{-S[\phi]}$$

$$R[z] \propto \mathrm{e}^{-V_1(z_1)}\mathrm{e}^{-V_2(z_2)}$$

$$Q[\phi] = \mathrm{e}^{-S_{\mathrm{eff}}[\phi[z]]} \equiv J_\theta^{-1}[z]R[z]$$

逆KLダイバージェンス

$$D_{\mathrm{KL}}[Q_\theta || P] = \int DzQ_\theta[z]\log\frac{Q_\theta[z]}{P[\phi[z]]}$$

$$= \int DzR[z]J_\theta^{-1}[z]\log\left(\frac{R[z]J_\theta^{-1}[z]}{P[\phi[z]]}\right)$$

各点独立にサンプルできる

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

賢い変数変換 (シンプレクティック積分と同様)

逆変換(even)

$$\begin{cases} \phi_1 = \phi'_1 \\ \phi_2 = \mathrm{e}^{-s(\phi'_1)}\phi'_2 \end{cases}$$

ステップ1

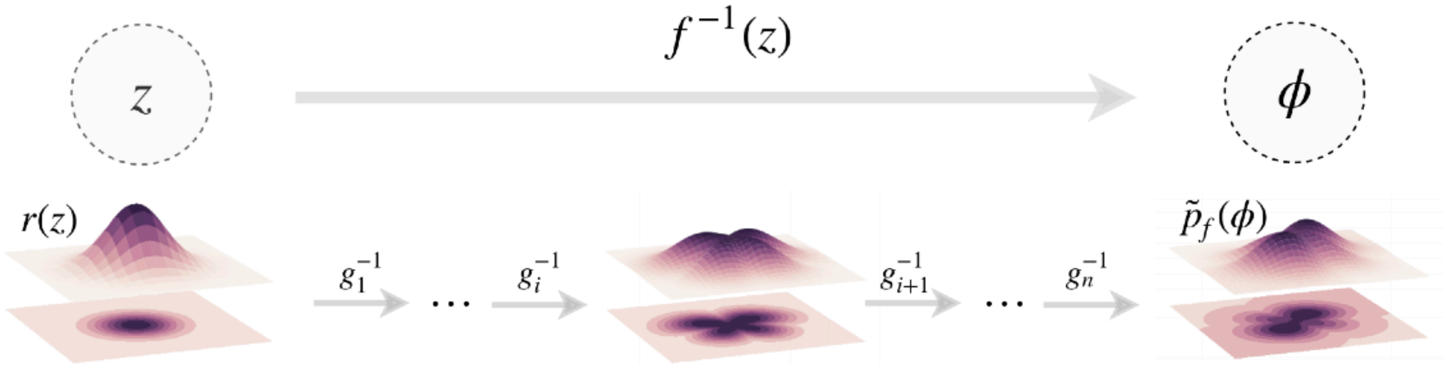
z_n を独立にサンプルする。

ステップ2

座標を偶奇にわけ、偶数点と奇数点を交互にニューラルネットト$s(\cdot)$で変換する (右式)
繰り返して作用させ、$\phi_n^{(\mathrm{app})}$ を得る。

ステップ3

出てきた結果は、$Q[\phi^{\mathrm{app}}] = J_\theta^{-1}[z]R[z]$ のように分布している。$Q[\phi^{\mathrm{app}}]$ と$P[\phi^{(\mathrm{app})}] = \mathrm{e}^{-S}/Z$ の距離(KL divergence)を測って最小化



(a) Normalizing flow between prior and output distributions

ここで出てきた分布は
少しずれてる
(近似なので)

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

賢い変数変換 (シンプレクティック積分と同様)

ステップ1

z_n を独立にサンプルする。

ステップ2

座標を偶奇にわけ、偶数点と奇数点を交互

にニューラルネット$s(\cdot)$で変換する (右式)

繰り返して作用させ、$\phi_n^{(\mathrm{app})}$ を得る。

ステップ3

出てきた結果は、$Q[\phi^{\mathrm{app}}] = J_\theta^{-1}[z]R[z]$ のように分布している。$Q[\phi^{\mathrm{app}}]$

と$P[\phi^{(\mathrm{app})}] = \mathrm{e}^{-S}/Z$ の距離(KL divergence)を測って最小化

ステップ4

メトロポリス・ヘイスティングステストを実施して、

近似による分布のズレを補正 (厳密アルゴリズム!)

逆変換(even)

$$\begin{cases} \phi_1 = \phi'_1 \\ \phi_2 = \mathrm{e}^{-s(\phi'_1)}\phi'_2 \end{cases}$$

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

フローベース法

Flow based ML for QFT

Akio Tomiya

MIT + Deepmind + …

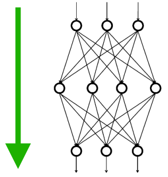
$$\int D\phi e^{-S[\phi]}O[\phi] \approx \underline{\prod_n \int \mathrm{d}z_n e^{-V(z_n)} J^{-1}[z] O[f[z]]}$$

Original integral: hard

Easy

フローベース法

自己相関なし
各点の相関なし
(ホットスタート)

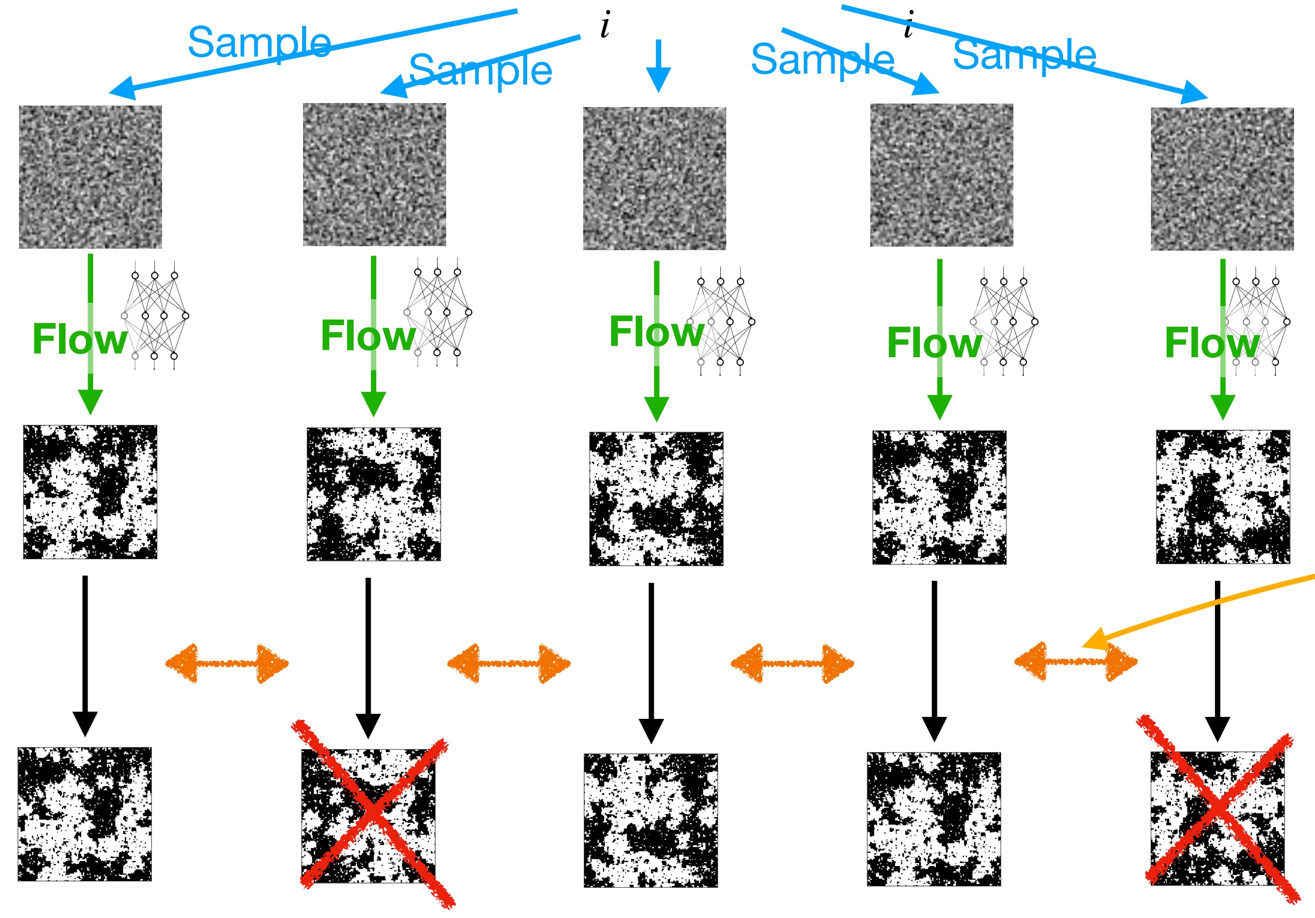


自己相関なし
各点の相関は近似的

正しい相関

配位間の自己相関は
かなり小さい

$$\prod_i^{Vol} e^{-V(\varphi_i)} = \prod_i^{Vol} r(\varphi_i)$$



Reject
(Use left conf.)

Reject
(Use left conf.)

①バカパラ 自明な理論から
(運動項なし、
~ T>>0のイジング模型、ランダム)

②非自明化写像
「冷却＝変数変換
学習済みニューラルネット

③メトロポリスヘイスティングス

$$e^{-S}/e^{-V(\varphi_i)}J^{-1}[\varphi]$$

シーケンシャル

自己相関時間

~ Rejection rate

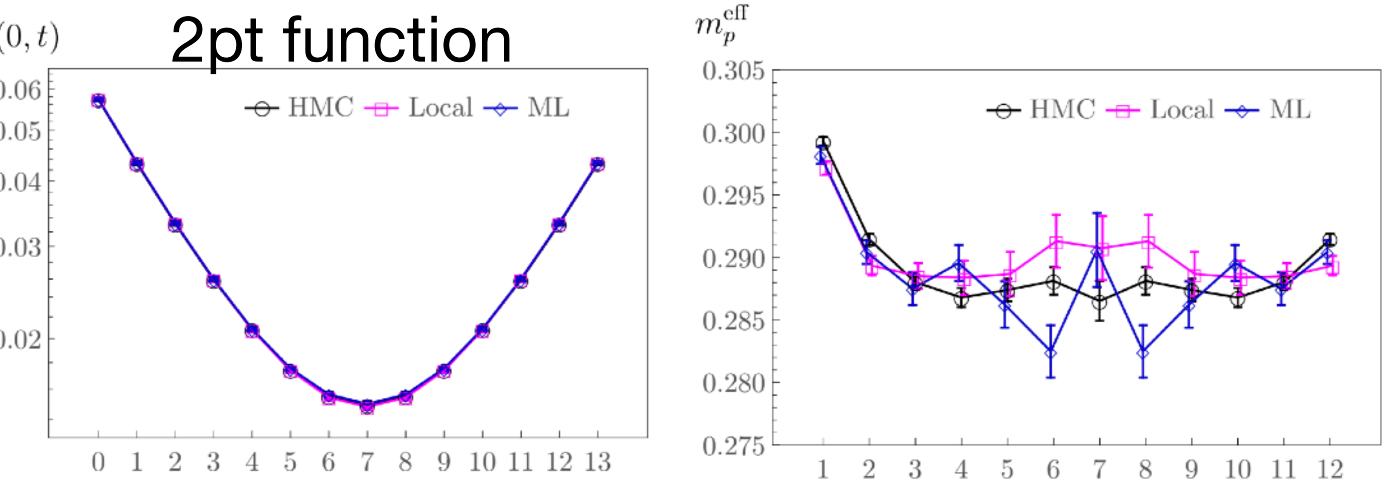
Configuration generation in LQCD

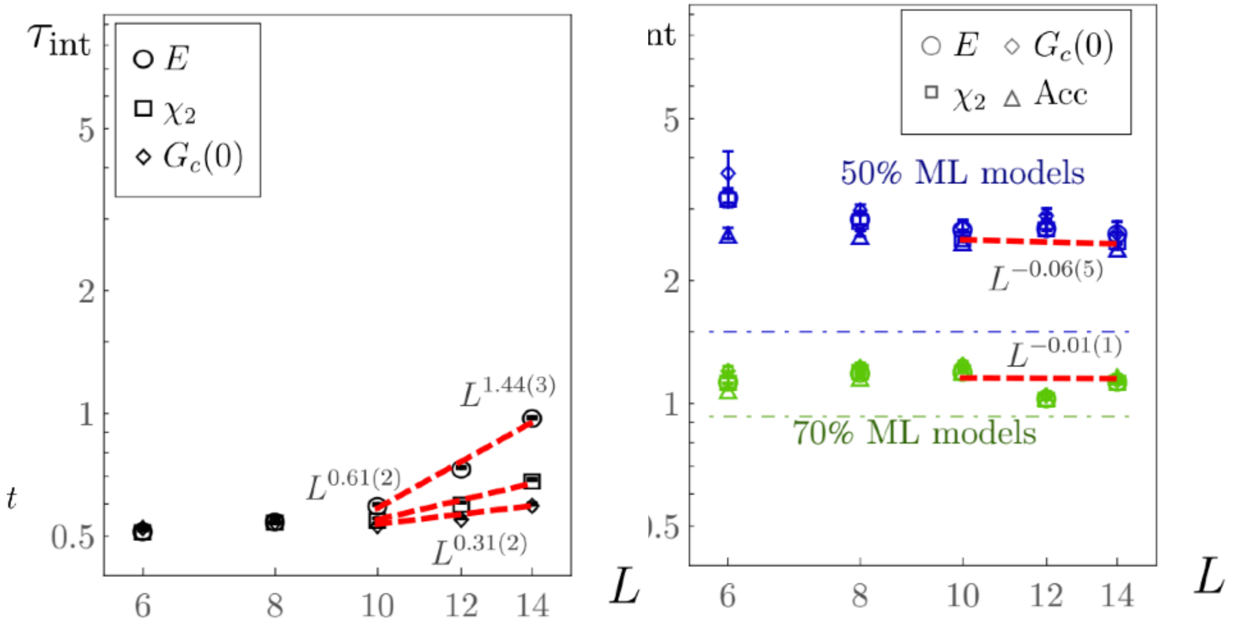
Flow based algorithm = neural net represented flow algorithm

Real scalar in 2 dimension

MIT + DeepMind 2019~

2pt function





(a) HMC ensembles

(c) Flow-based MCMC ensembles

物理量は一致 (厳密アルゴリズム)

自己相関は短い (?)

arxiv 1904.12072, 2003.06413, 2008.05456

フローベース法
ゲージ場のトポロジー

Akio Tomiya

Abelian chiral gauge theories on the lattice with exact gauge invariance
Martin Luscher(DESY)
Nucl.Phys.B 549 (1999) 295-334 • e-Print: hep-lat/9811032 • DOI: 10.1016/S0550-3213(99)00115-7
Weyl fermions on the lattice and the nonAbelian gauge anomaly
Martin Luscher(DESY)
Nucl.Phys.B 568 (2000) 162-179 • e-Print: hep-lat/9904009 • DOI: 10.1016/S0550-3213(99)00731-2

ゲージ場には、トポロジーがある。

アドミッシビリティ条件 ~ プラケットが1に十分近ければ
連続理論と同様にトポロジカルセクターに分割できる
(ゲージ場の配位に対して整数(トポロジカル電荷)を付与できる)

大きなβ (= 小さな格子間隔a) でHMCを実行すると、ゲージ場のトポロジーが
変化しなくなってしまう。トポロジー凍結問題

格子アーティファクトを削減するには、小さなa がほしいのでかなり悩ましい。

フローベース法は、β = 0からβ>> 0 の配位をつくる。
アドミッシビリティを強く破ってサンプル、アドミッシビリティを守る領域へ変形。
→ トポロジー凍結には有用(?)

とはいえ、不安もあるので厳密解がある2d でcheckしましょう

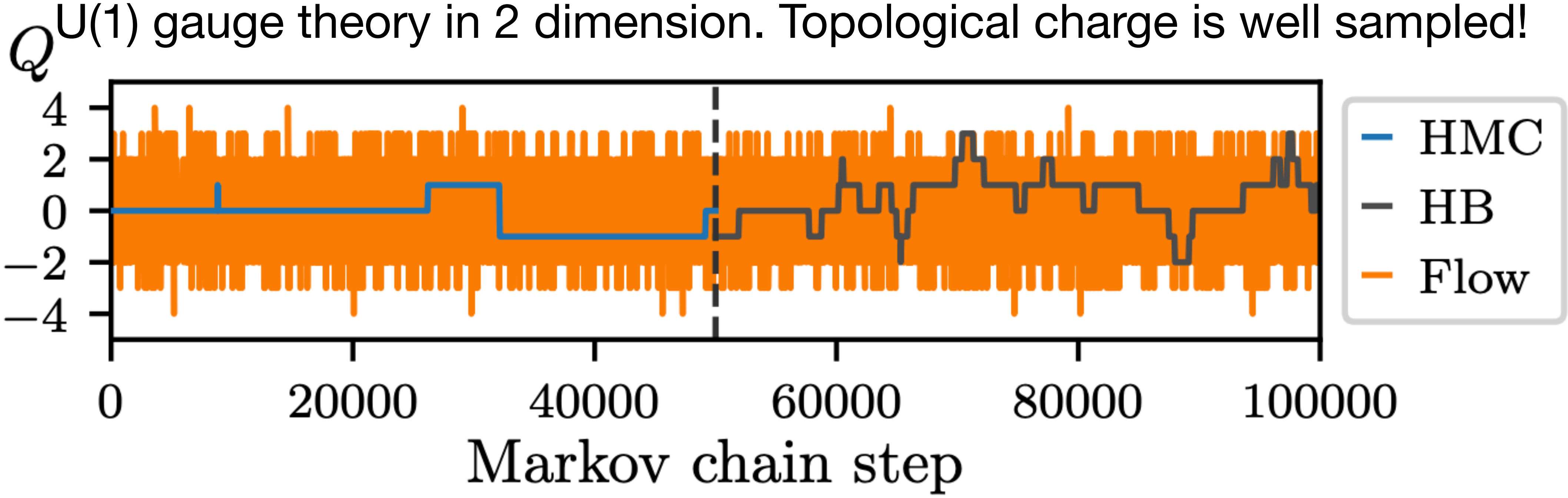
arXiv: 2101.08176 and ref therein

Configuration generation in LQCD

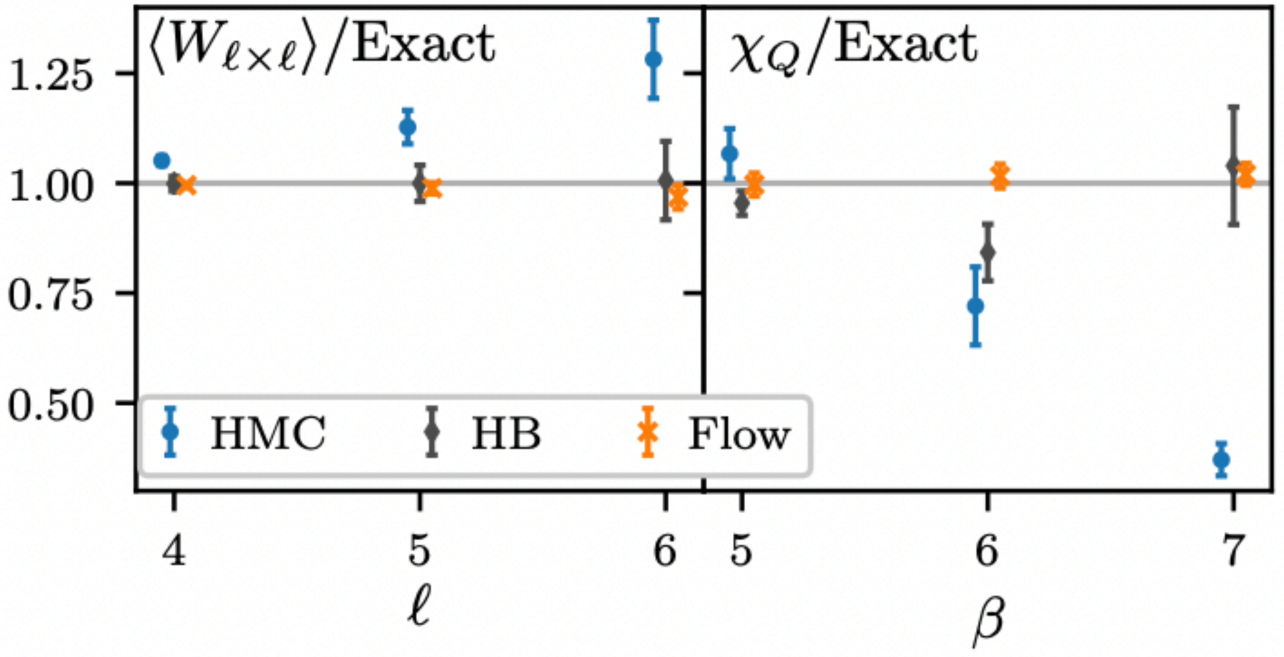
Akio Tomiya

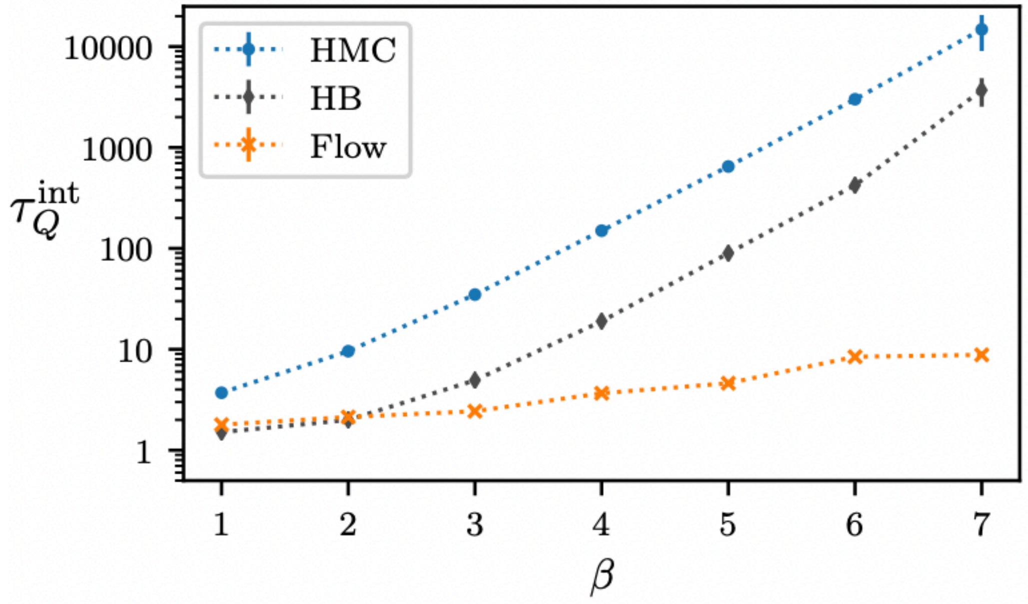
Flow based algorithm = neural net represented flow algorithm

U(1) gauge theory in 2 dimension. Topological charge is well sampled!



物理量もバッチリあう。



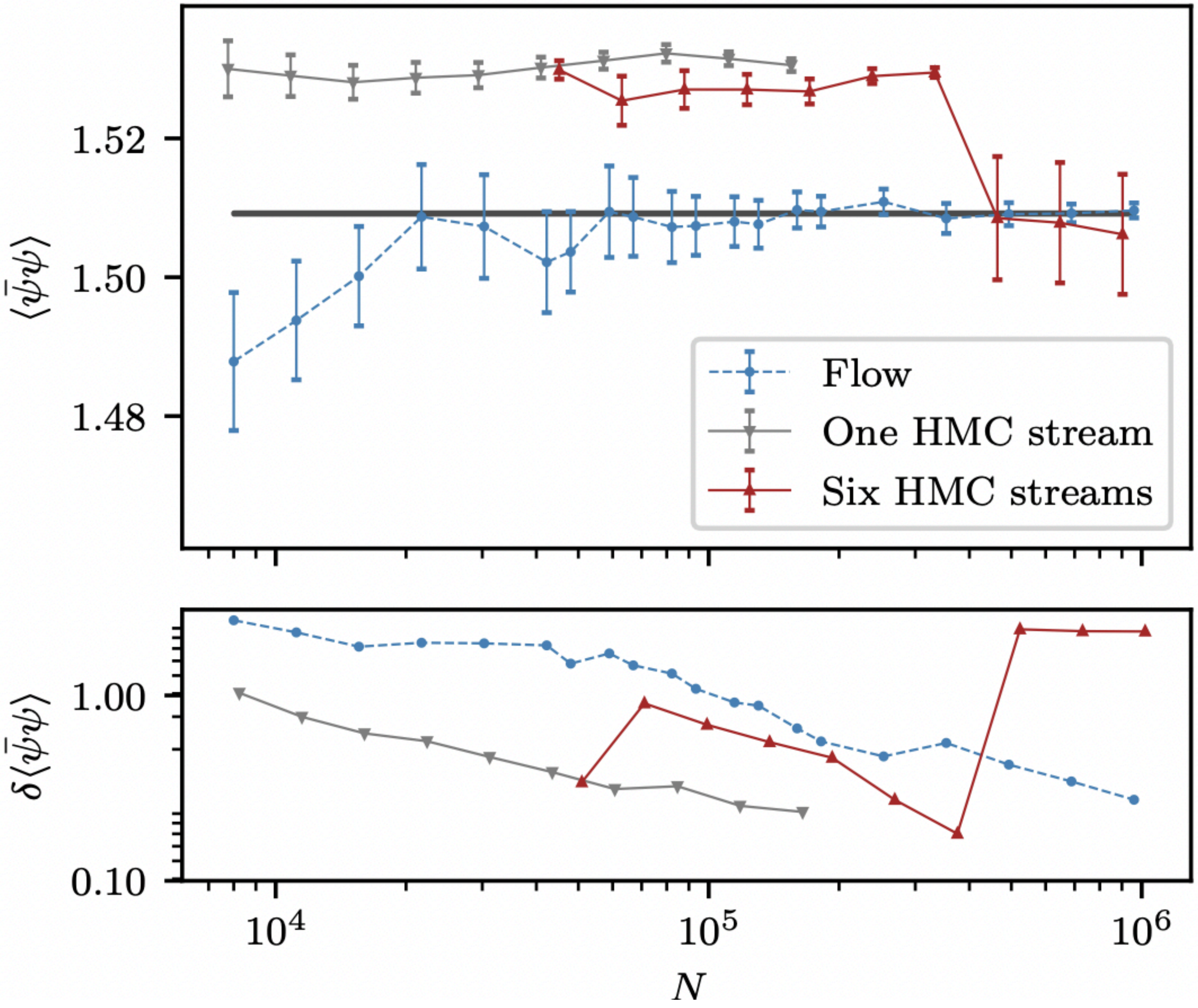


arxiv 1904.12072, 2003.06413, 2008.05456

Configuration generation in LQCD

Schwinger model, 2d QED (Toy model of QCD in 4d)

https://arxiv.org/abs/2202.11712



Compare to the exact solution

HMC fails to sample
-> strongly auto-correlated

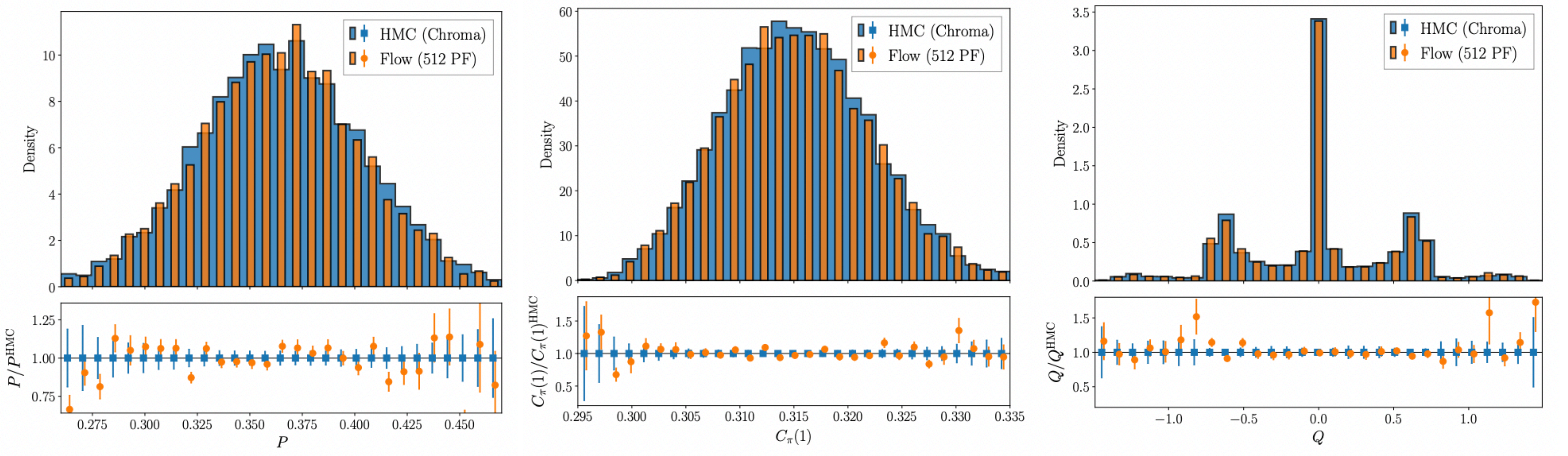
Low-laying modes in D affects a lot & source of auto-correlation in HMC

arxiv 1904.12072, 2003.06413, 2008.05456

Configuration generation in LQCD

4d QCD

[1] MIT + Deepmind +



(a) Plaquette

(c) Pion correlation function at $x_0 = 1$

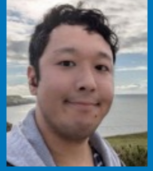
(d) Topological charge at $t/a^2 = 4$

- Results for full QCD, four dimensional SU(3), Wilson fermions
  - Lattice volume 4^4, $\beta = 1$, and $\kappa = 0.1$, $Nf = 2$
- Larger volume? Scaling?
- Which kind of neural net?

[1] https://arxiv.org/abs/2208.03832, https://arxiv.org/abs/2211.07541

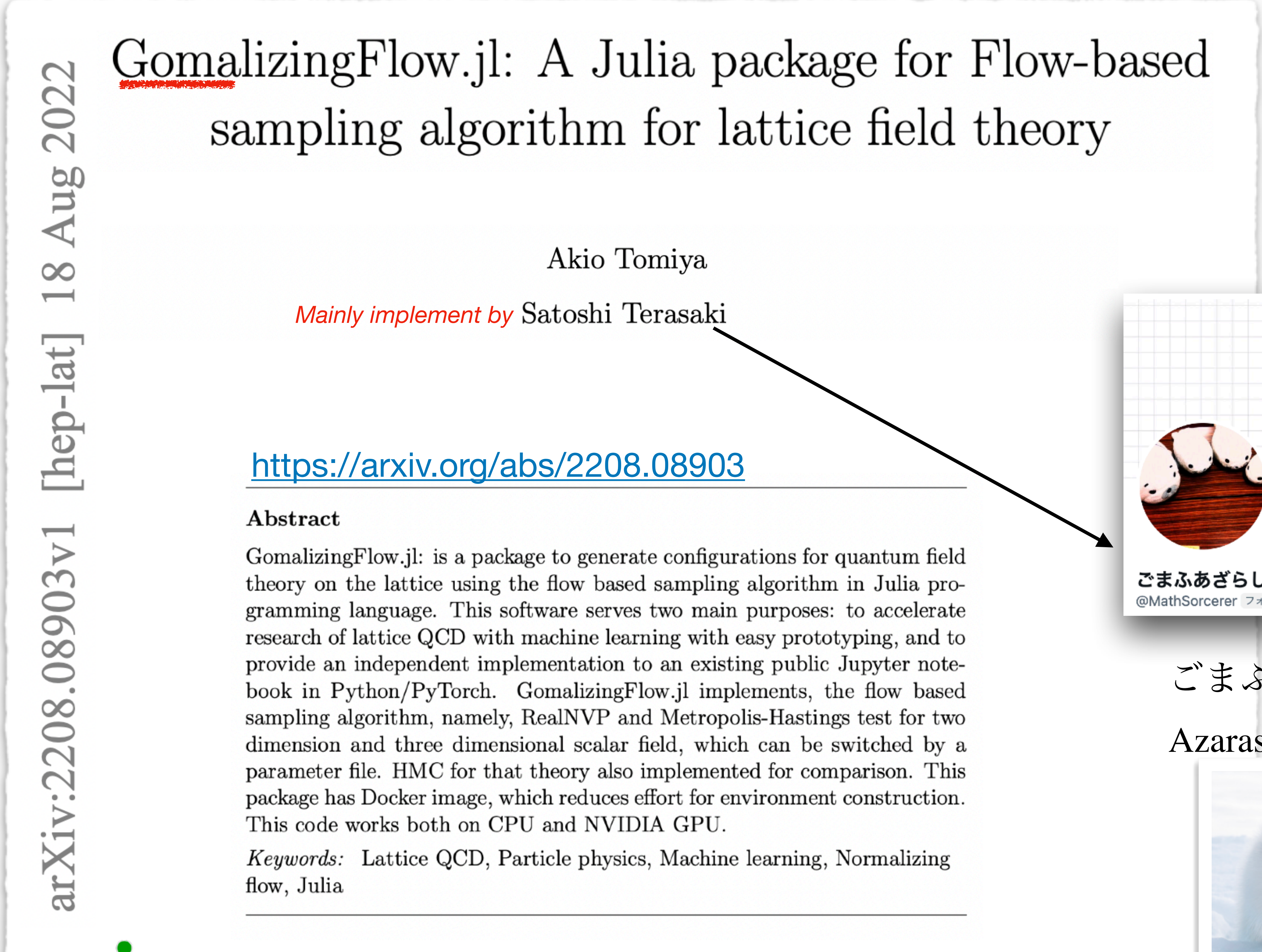
Normalizing flow in Julia

We made a public code in Julia Language



Akio Tomiya

AT+ 2022



arXiv:2208.08903v1 [hep-lat] 18 Aug 2022

GomalizingFlow.jl: A Julia package for Flow-based sampling algorithm for lattice field theory

Akio Tomiya

Mainly implement by Satoshi Terasaki

https://arxiv.org/abs/2208.08903

Abstract

GomalizingFlow.jl: is a package to generate configurations for quantum field theory on the lattice using the flow based sampling algorithm in Julia programming language. This software serves two main purposes: to accelerate research of lattice QCD with machine learning with easy prototyping, and to provide an independent implementation to an existing public Jupyter notebook in Python/PyTorch. GomalizingFlow.jl implements, the flow based sampling algorithm, namely, RealNVP and Metropolis-Hastings test for two dimension and three dimensional scalar field, which can be switched by a parameter file. HMC for that theory also implemented for comparison. This package has Docker image, which reduces effort for environment construction. This code works both on CPU and NVIDIA GPU.

Keywords: Lattice QCD, Particle physics, Machine learning, Normalizing flow, Julia

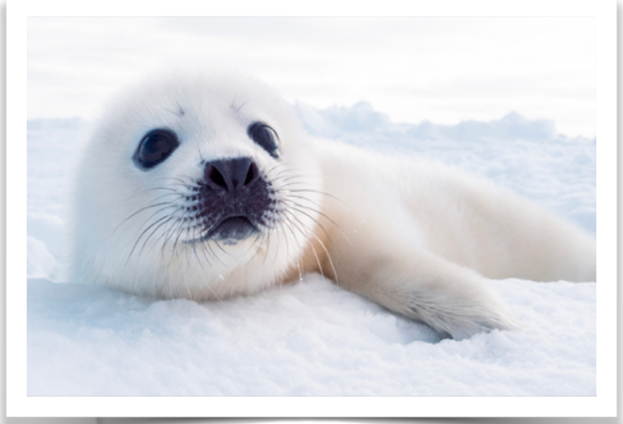
julia

ごまふあざらし(GomahuAzarashi)

@MathSorcerer フォローされています

ごまふあざらし = GOMAfu

Azarashi (spotted seal)



↓ economic convolution for flow

https://github.com/AtelierArith/GomalizingFlow.jl https://ml4physicalsciences.github.io/2022/files/NeurIPS_ML4PS_2022_31.pdf

完全作用 (Perfect action)

完全作用

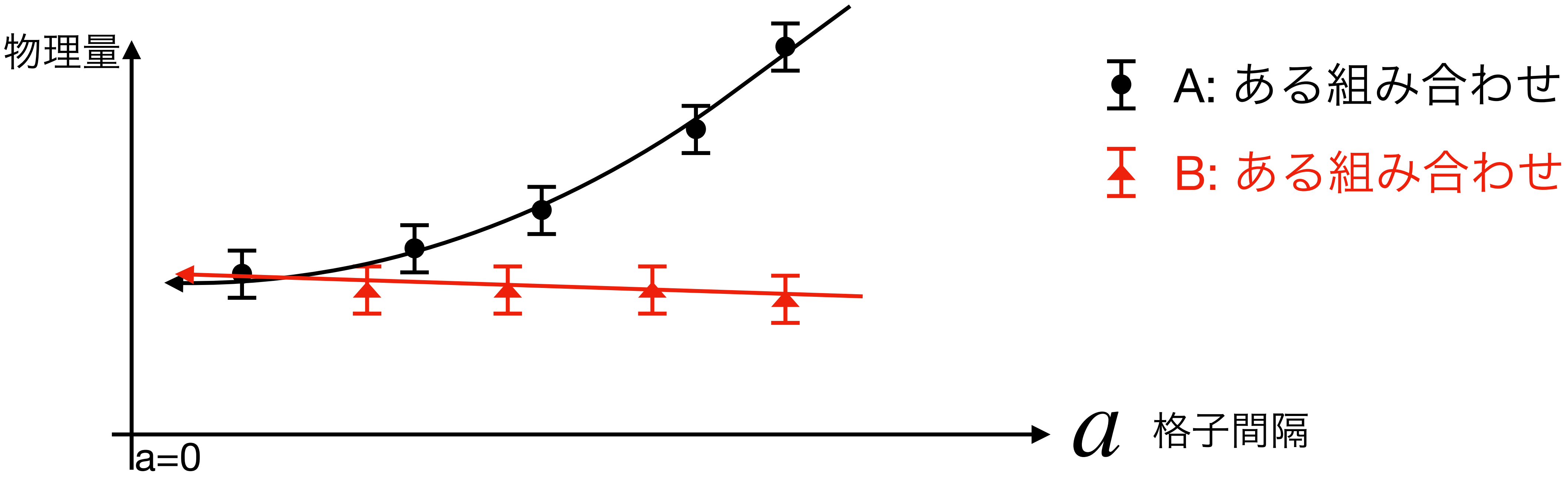
同じ物理量を計算するにしても離散化の任意性

$$\langle O^{(\mathrm{lat})}\rangle_{(\mathrm{lat})} = \frac{1}{Z}\int D\phi \mathrm{e}^{-S^{(\mathrm{lat})}} O^{(\mathrm{lat})}$$

Kieran Holland+
https://arxiv.org/abs/2401.06481

$$\begin{cases} \text{格子作用 } S^{(\mathrm{lat})} = \text{連続作用 } S^{(\mathrm{cnt})} + (a\text{の高次}) \\ \text{格子上の物理量 } O^{(\mathrm{lat})} = \text{連続理論の物理量 } O^{(\mathrm{cnt})} + (a\text{の高次}) \end{cases}$$

➡ **格子上の物理量の期待値 $\langle O^{(\mathrm{lat})}\rangle_{(\mathrm{lat})}$ = 連続理論の物理量 $\langle O^{(\mathrm{cnt})}\rangle$ + (aの高次)**

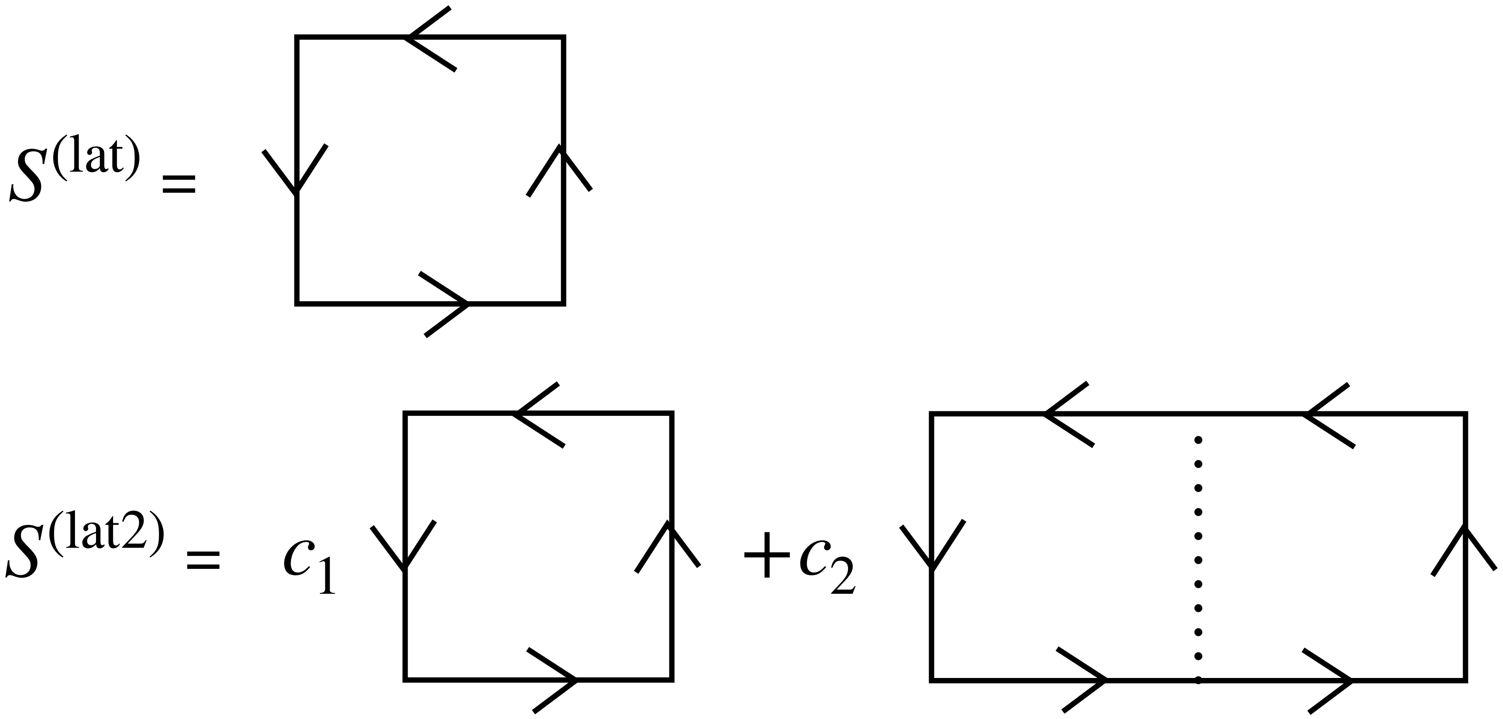


組み合わせB のほうが良い。なぜならば格子間隔依存性がマイルド

(最初から連続極限に近い). 物理量はおいておいて、こんな作用を作れるか。

完全作用

Symanzik improvement program



$S^{(\mathrm{lat})} =$

$S^{(\mathrm{lat2})} = \ c_1 \qquad + c_2$

格子化による影響を減らすようにc_i を決める。

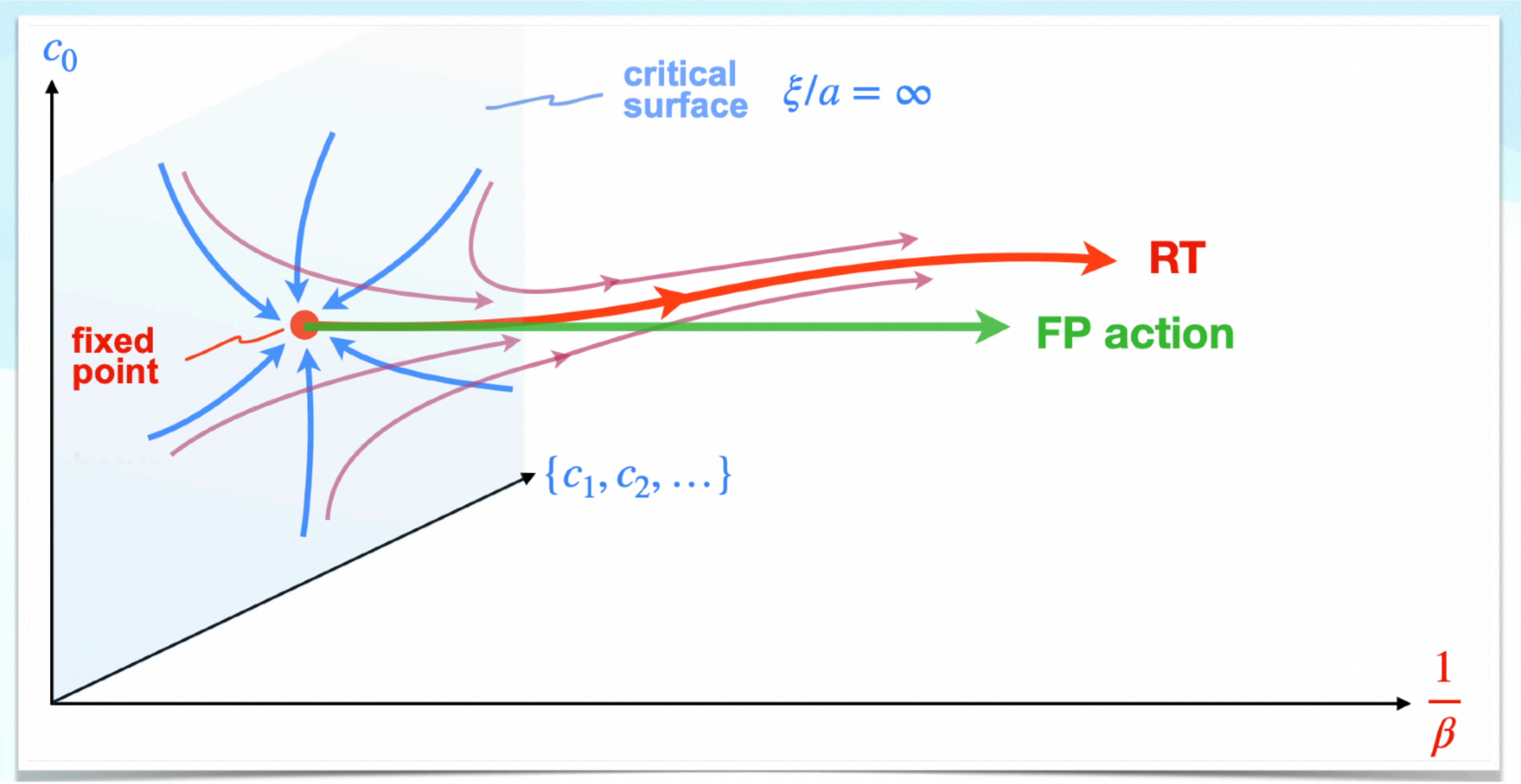
- たとえば、$U_\mu(n) = \exp(\mathrm{i}aA_\mu(n))$ として、作用が連続極限に行くようにする

  (tree level improved symanzik action)
- くりこみ群から決める (岩崎作用)
- 数値的なブロックスピン変換から決める (DBW2作用)

もっと良いのはないのか?

完全作用

RT軌道上の作用なら離散化によるズレが少ない



Kieran Holland+
https://arxiv.org/abs/2401.06481

P. Hasenfratz, F. Niedermayer [Nucl. Phys. B414 (1994) 785, hep-lat/9308004]

本当に固定点近く、

くりこみ群軌道上に近い作用で良い?

→ 完全作用

アンザッツとして、ゲージ同変ニューラルネット(スメアリング)

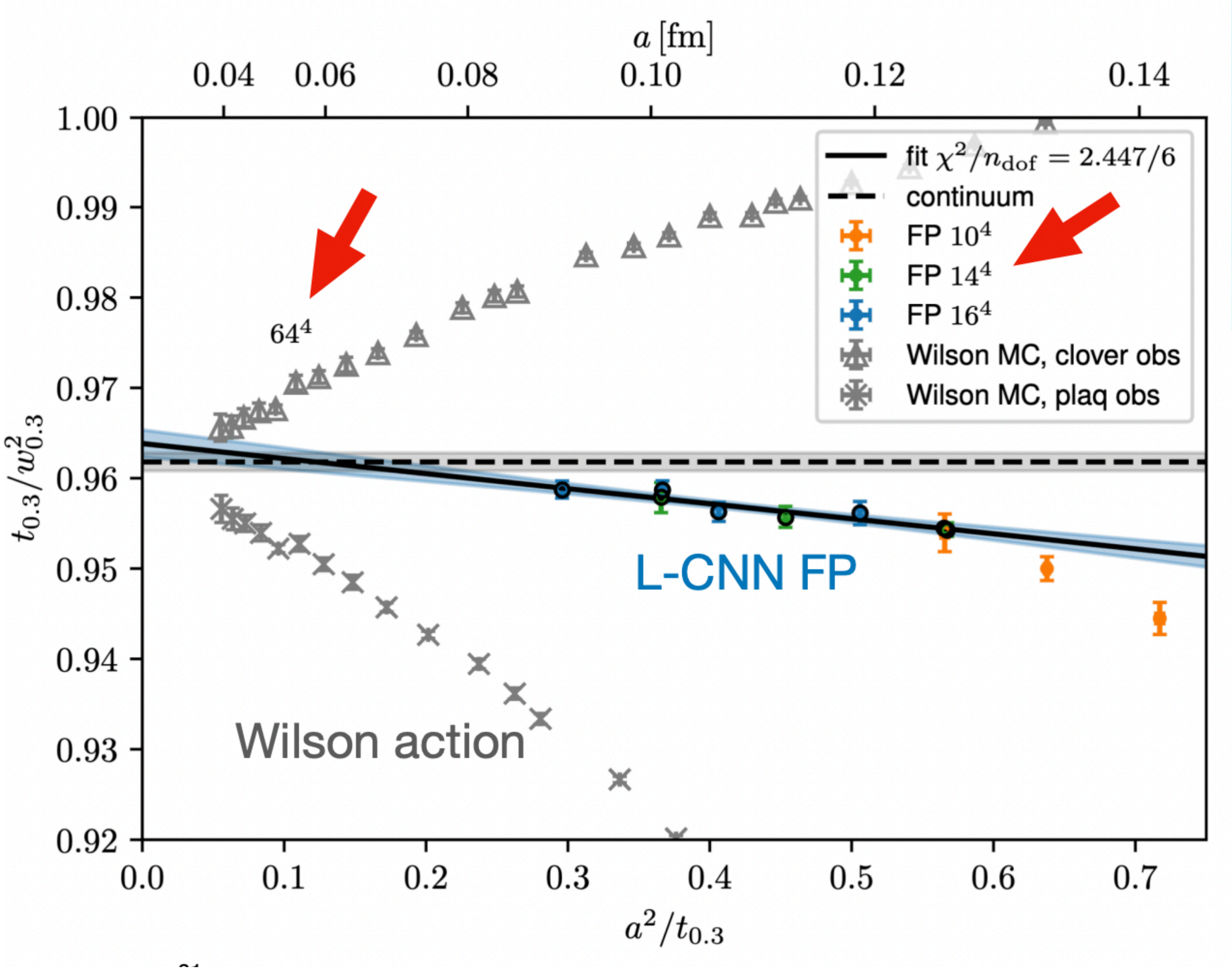
を用いて作用を書く (沢山のimprovement terms)

http://pyweb.swan.ac.uk/~aarts/ml-lft-files/david.pdf

完全作用

RT軌道上の作用なら離散化によるズレが少ない

Kieran Holland+
https://arxiv.org/abs/2401.06481



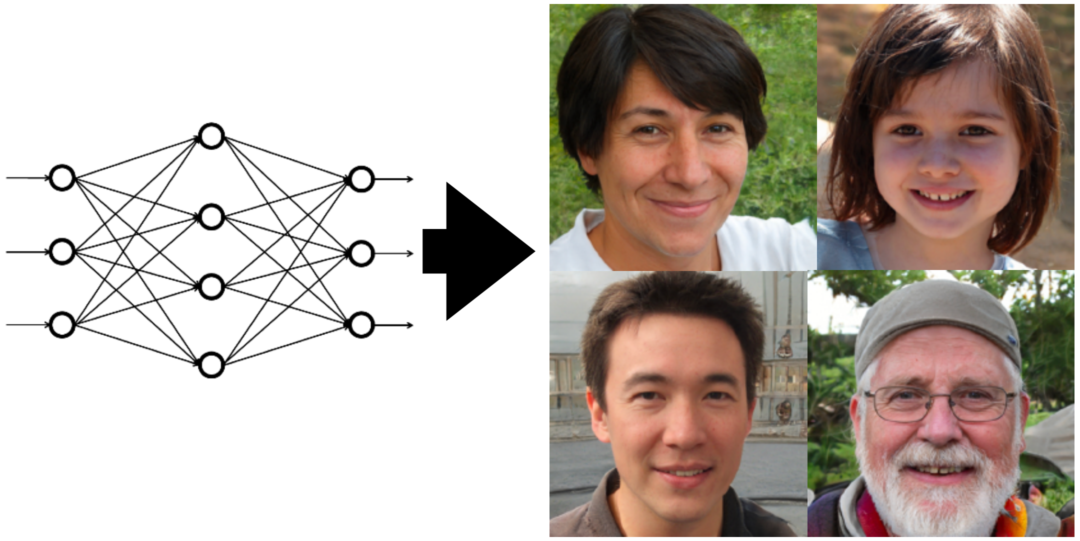
Two previous works to realize Gauge symmetric Transformer for LQCD

1. Gauge covariant net arXiv: 2103.11965 AT+
2. Transformer for fermion-spin systems

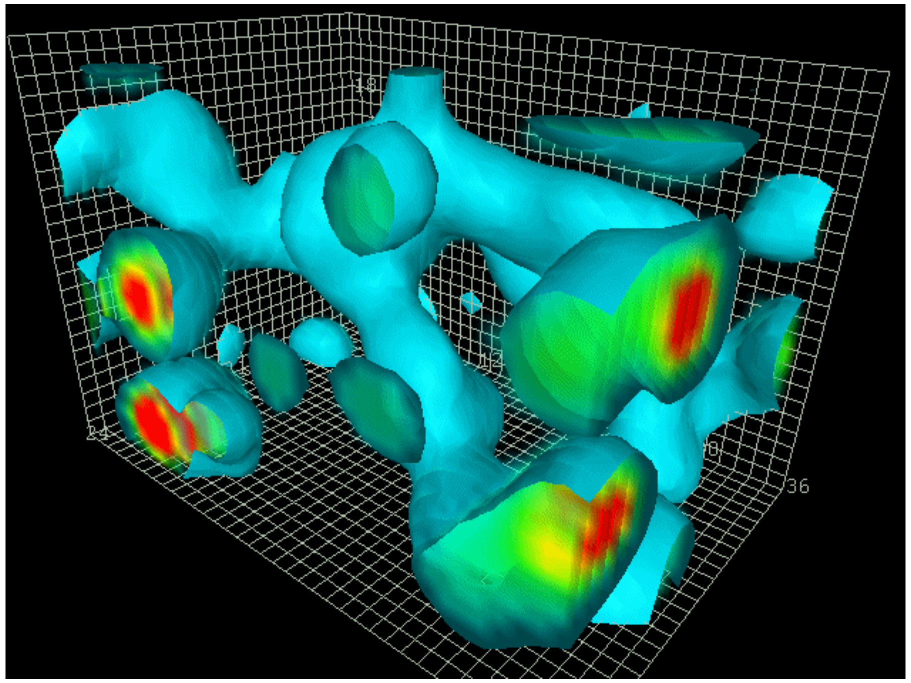
2310.13222 AT+
2306.11527 AT+

ML for LQCD is needed

- Neural networks
  - Data processing techniques mainly for 2d image (a picture = pixels = a set of real #)
  - Neural network helps data processing e.g. AlphaFold3
- Lattice QCD requires numerical effort but is more complicated than pictures
  - 4 dimension
  - **Non-abelian gauge d.o.f. and symmetry**
  - Fermions (Fermi-Dirac statistics)
  - Exactness of algorithm is necessary
- Q. How can we deal with neural nets?



thispersondoesnotexist.com



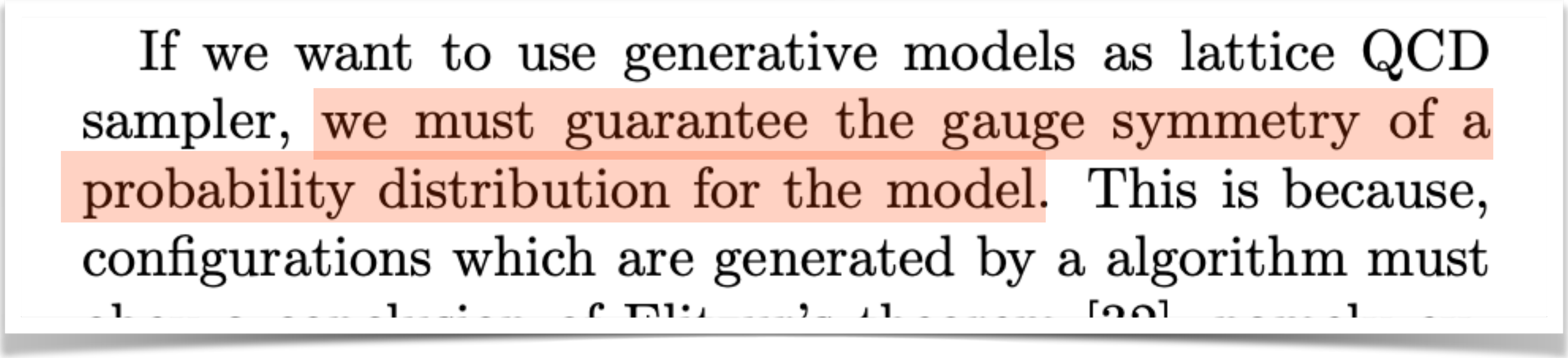
http://www.physics.adelaide.edu.au/theory/staff/leinweber/VisualQCD/QCDvacuum/

What is the neural networks?

Attempts to gauge symmetry

7,8 years! 😯

In my paper for fields generation using ML (1712.03893),



If we want to use generative models as lattice QCD sampler, we must guarantee the gauge symmetry of a probability distribution for the model. This is because, configurations which are generated by a algorithm must

We have created several architectures:

2010.11900, AT+: Gauge *invariant* self-learning MC for 4d LQCD

2103.11965, AT+ Gauge *covariant* self-learning HMC for 4d LQCD
(Covariant NN = adaptive gradient flow = adaptive stout)

(2310.13222, AT+: Global symmetric transformer for fermion-spin system)

This work, AT+: *Gauge* symmetric transformer for 4d LQCD

There are several realization of gauge covariant maps arXiv:2012.12901 arXiv: 2305.02402

Overview/outline

Gauge covariant transformer for LQCD

Two conditions/restrictions in LQCD:

Gauge symmetry
U(x, x+μ)

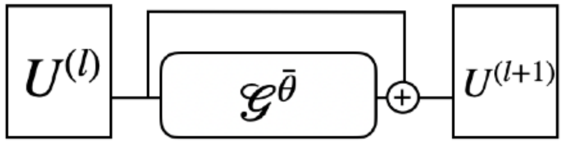
Non-locality from
pseudo-fermions
(1/D) ~ non-local

(I want to mimic
this by NN)

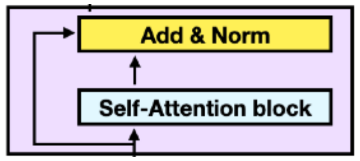
Solutions in neural net:

arXiv: 2103.11965 AT+

1. Gauge covariant net
(adaptive stout)



2. Transformer with global symmetry
(Heisenberg spin + electron)



2310.13222 AT+
2306.11527 AT+

3. Gauge symmetric Transformer for LQCD

This talk

Overview/outline

Gauge covariant transformer for LQCD

Two conditions/restrictions in LQCD:

Gauge symmetry
U(x, x+μ)

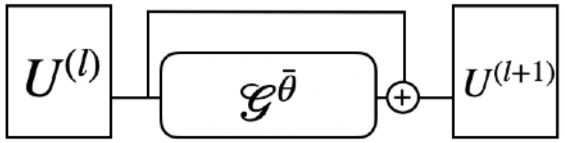
Non-locality from
pseudo-fermions
(1/D) ~ non-local

(I want to mimic
this by NN)

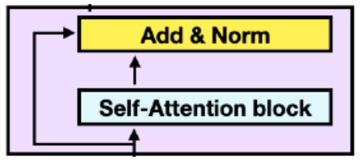
Solutions in neural net:

arXiv: 2103.11965 AT+

1. **Gauge covariant net**
(adaptive stout)



2. Transformer with global symmetry
(Heisenberg spin + electron)



2310.13222 AT+
2306.11527 AT+

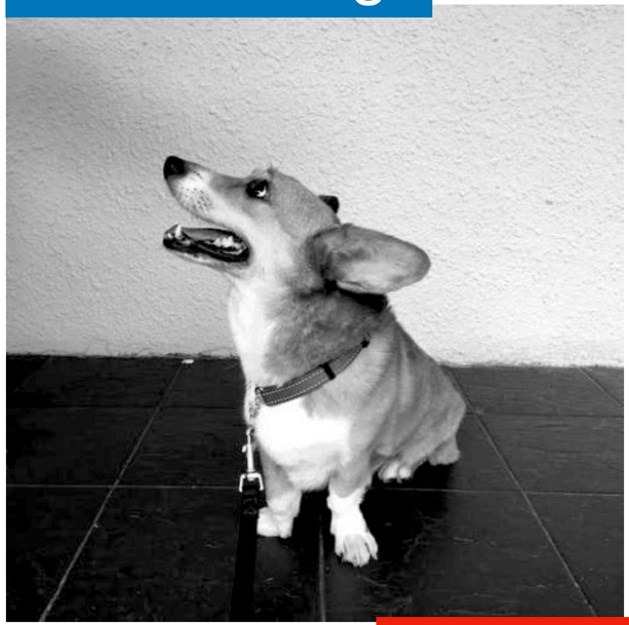
3. Gauge symmetric Transformer for LQCD

This talk

What is conv. neural networks?

The convolution layer can treat a translation transformation

Filter on image



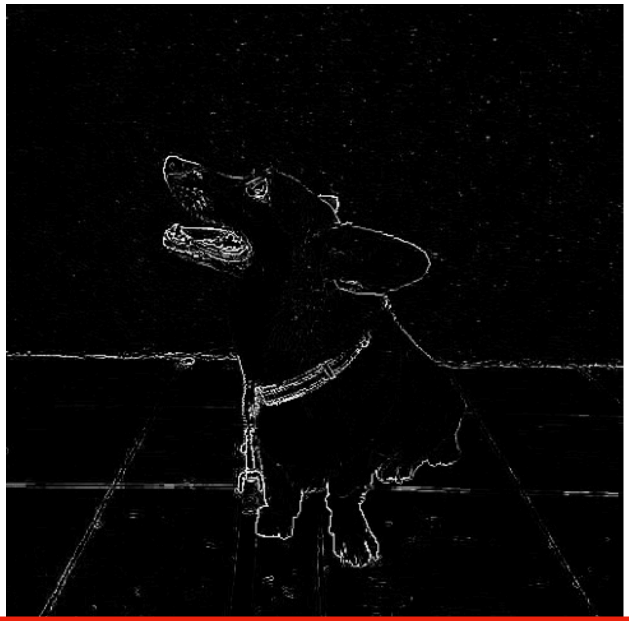
*

Laplacian filter

0	1	0
1	-2	1
0	1	0

(Discretization of ∂^2)

=



Edge detection

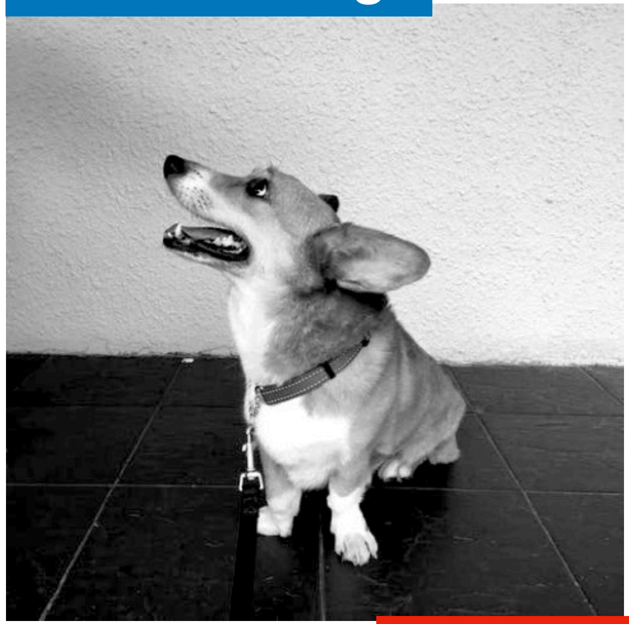
IMPORTANT: If inputs are shifted to right, outputs are shifted to right

= translationally equivaliant (similar to covariance, operation just commute)

What is conv. neural networks?

Convolution layer = trainable filter

Filter on image



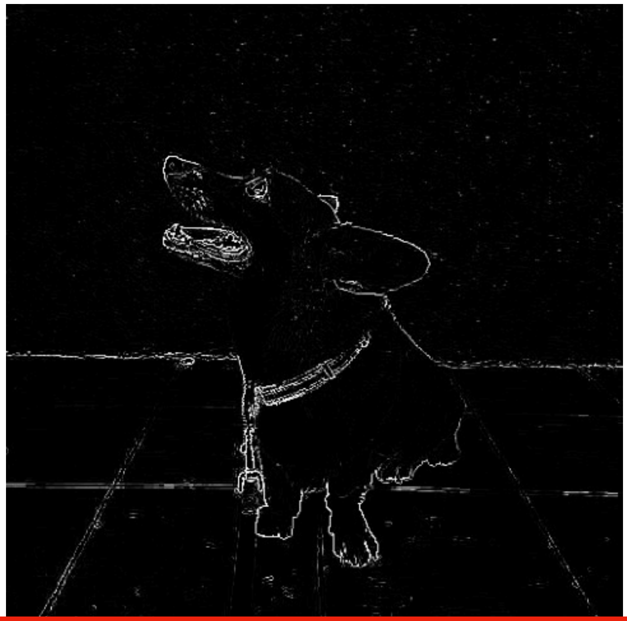
*

Laplacian filter

0	1	0
1	-2	1
0	1	0

(Discretization of ∂^2)

=



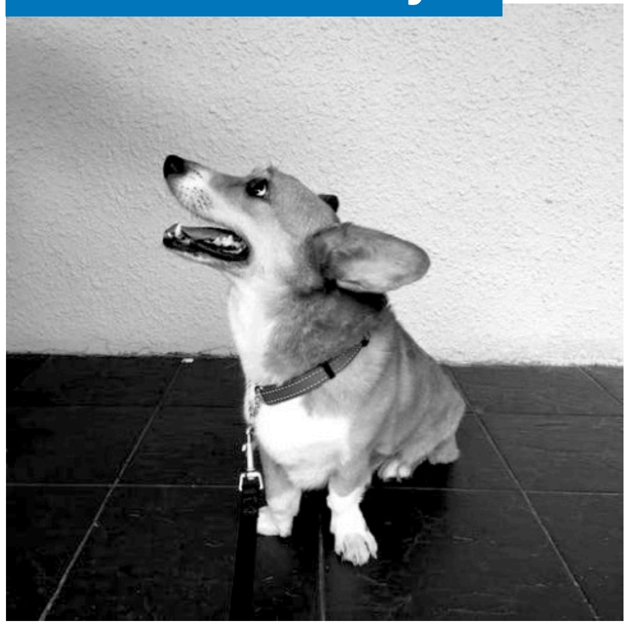
Edge detection

IMPORTANT: If inputs are shifted to right, outputs are shifted to right

= translationally equivaliant (similar to covariance, operation just commute)

Convolution layer

Fukushima, Kunihiko (1980)
Zhang, Wei (1988) + a lot!



*

Trainable filter

W_{11}	W_{12}	W_{13}
W_{21}	W_{22}	W_{23}
W_{31}	W_{32}	W_{33}

→

Edge detection

Smoothing
(Gaussian filter)

...

Gaussian filter

$\frac{1}{16}$

1	2	1
2	4	2
1	2	1

This can be any filter which helps feature extraction but still transitionally equivariant!

Smearing

Smoothing improves global properties

Eg.

Coarse image

Numerical derivative is unstable

Gaussian filter

$$\frac{1}{16}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 1 \\ \hline 1 & 2 & 1 \\ \hline\end{array} \longrightarrow$$

Smoothened image

Numerical derivative is stable

We want to smoothen gauge configurations with keeping gauge symmetry

Two types:

APE-type smearing

Stout-type smearing

M. Albanese+ 1987
R. Hoffmann+ 2007
C. Morningster+ 2003

Smearing

Smoothing with gauge symmetry, APE type

M. Albanese+ 1987
R. Hoffmann+ 2007

APE-type smearing

Covariant sum

$$U_\mu(n) \to U_\mu^{\text{fat}}(n) = \mathcal{N}\left[(1-\alpha)U_\mu(n) + \frac{\alpha}{6}V_\mu^\dagger[U](n)\right]$$

Normalization

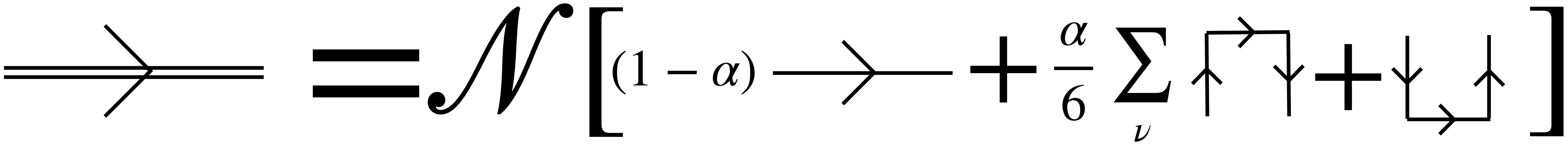
$$\mathcal{N}[M] = \frac{M}{\sqrt{M^\dagger M}}$$

Or projection

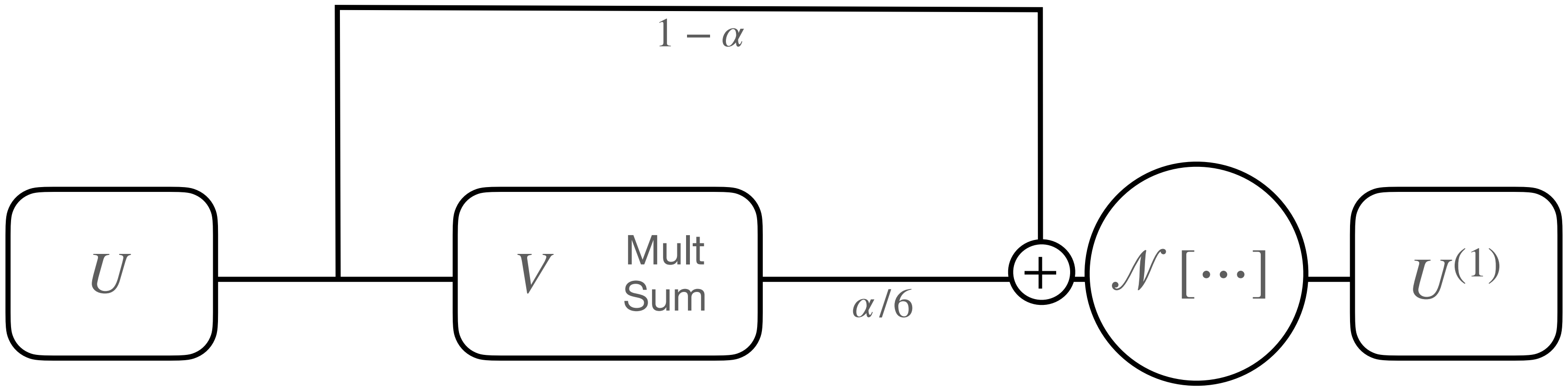
$$V_\mu^\dagger[U](n) = \sum_{\mu\neq\nu} U_\nu(n)U_\mu(n+\hat{\nu})U_\nu^\dagger(n+\hat{\mu}) + \cdots$$

$V_\mu^\dagger[U](n)$& $U_\mu(n)$ shows same transformation
→$U_\mu^{\text{fat}}[U](n)$ is as well

Schematically,

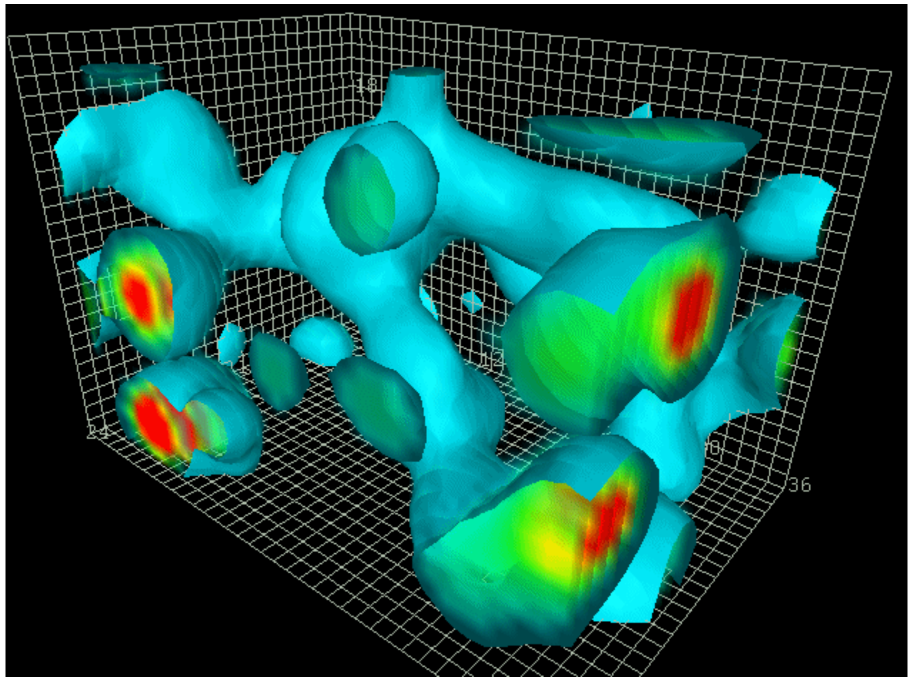


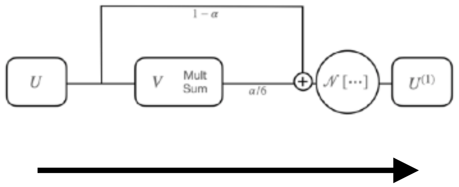
In the calculation graph,

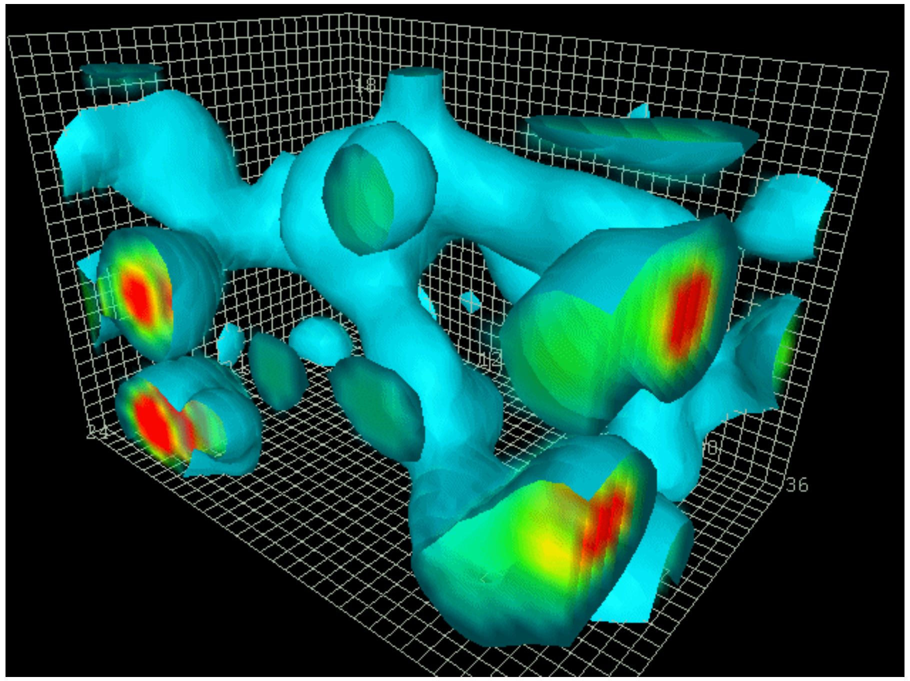


Smearing is a map with gauge covariance

Smearing makes a map between configurations, works as a *filter*







Gauge covariant neural network

Akio Tomiya

= trainable smearing (= residual flow)

AT Y. Nagai arXiv: 2103.11965

Smearing = gauge covariant way of transform gauge configurations

Covariant sum

$$U_\mu(n) \to U_\mu^{\mathrm{smr}}(n) = \mathcal{N}\left[(1-\alpha)U_\mu(n) + \frac{\alpha}{6}V_\mu^\dagger[U](n)\right]$$

staple

$$V_\mu^\dagger[U](n) = \sum_{\mu\neq\nu} U_\nu(n)U_\mu(n+\hat\nu)U_\nu^\dagger(n+\hat\mu) + \cdots$$

Normalization

$$\mathcal{N}[M] = \frac{M}{\sqrt{M^\dagger M}}$$

Or projection

Gauge covariant neural network = smearing with tunable parameters w

$$\begin{cases} z_\mu^{(l)}(n) = w_1 U_\mu(n) + w_2 V_\mu^\dagger[U](n) \\ U_\mu^{(l+1)}(n) = \mathcal{N}(z_\mu^{(l)}(n)) \end{cases}$$

Trainable param

link-wise projection/normalization (local)

Gauge covariant NN: $U_\mu^{\mathrm{NN}}(n)[U] = U_\mu^{(4)}(n)\left[U_\mu^{(3)}(n)\left[U_\mu^{(2)}(n)\left[U_\mu(n)\right]\right]\right]$

Gauge covariant variational map: $U_\mu(n) \mapsto U_\mu^{\mathrm{NN}}(n) = U_\mu^{\mathrm{NN}}(n)[U]$

Stout type can be constructed in the same way

There are several realization of gauge covariant maps arXiv:2012.12901 arXiv: 2305.02402

Gauge covariant neural network

= trainable smearing (= residual flow)

AT Y. Nagai arXiv: 2103.11965

Stout-type

$$U_\mu(n) \to U^{\mathrm{smr}}_\mu(n) = \mathrm{e}^{\sum_i \rho_i L_i[U]} U_\mu(n)$$

staple

$$V^\dagger_\mu[U](n) = \sum_{\mu \neq \nu} U_\nu(n) U_\mu(n+\hat{\nu}) U^\dagger_\nu(n+\hat{\mu}) + \cdots$$

Trainable param

Training done by the back-prop
(extension to the stout paper [1])

Following results using this stout type

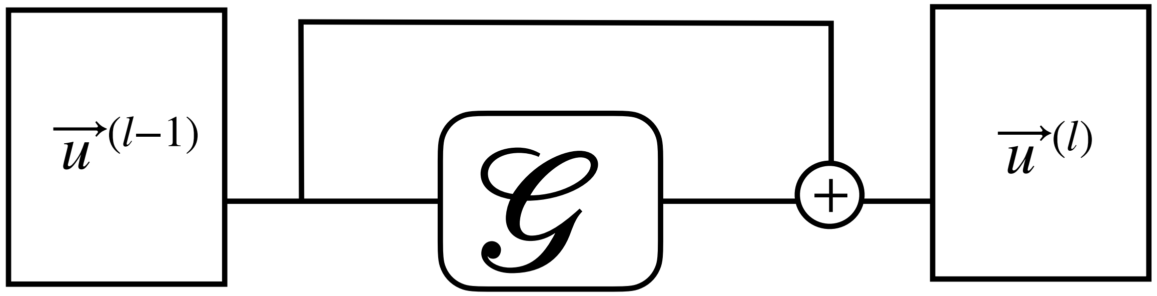
[1] C. Morningster+ 2003

There are several realization of gauge covariant maps arXiv:2012.12901 arXiv: 2305.02402

Gauge covariant neural network

Neural ODE of Cov-Net = “gradient flow”

ResNet



arXiv: 1512.03385

Continuum
Layer
Limit

Neural ODE

$$\frac{d\overrightarrow{u}^{(t)}}{dt} = \mathscr{G}(\overrightarrow{u}^{(t)})$$

arXiv: 1806.07366
(Neural IPS 2018 best paper)

Gauge covariant neural network

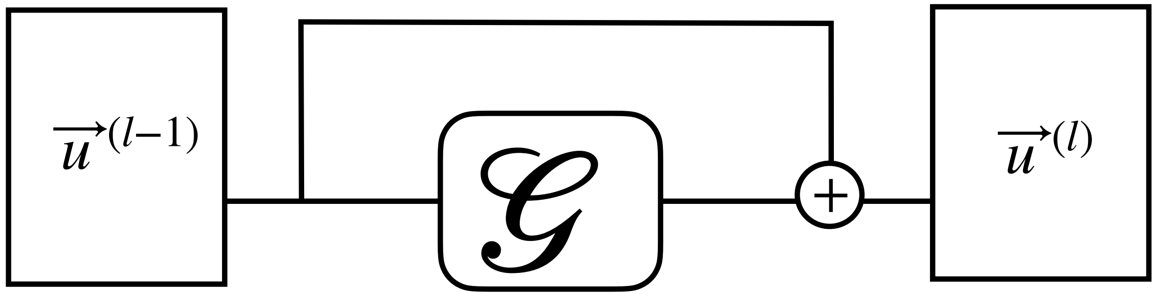
Akio Tomiya

Neural ODE of Cov-Net = “gradient flow”

ResNet

Continuum
Layer
Limit

Neural ODE



arXiv: 1512.03385

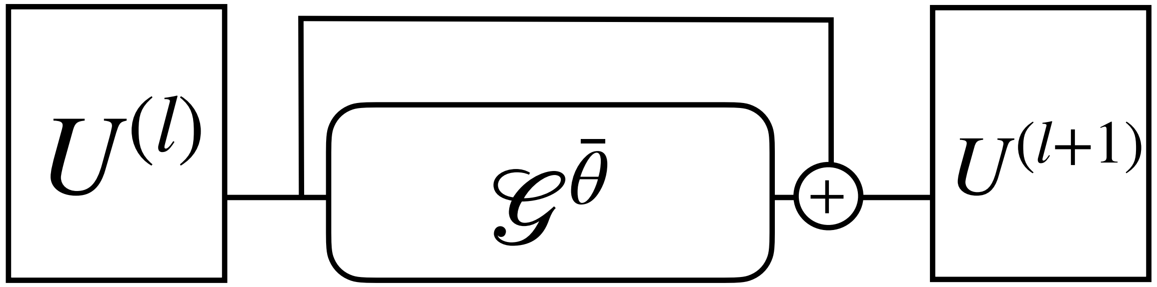
$$\frac{d\vec{u}^{(t)}}{dt} = \mathscr{G}(\vec{u}^{(t)})$$

arXiv: 1806.07366
(Neural IPS 2018 best paper)

Gauge-cov net

Continuum
Layer
Limit

Neural ODE
for Gauge-cov NN



AT Y. Nagai arXiv: 2103.11965

$$\frac{dU_\mu^{(t)}(n)}{dt} = \mathscr{G}^{\bar{\theta}}(U_\mu^{(t)}(n))$$

“Gradient” flow
(not has to be gradient of S)

“Continuous stout smearing is the Wilson flow”

2010 M. Luscher
AT Y. Nagai arXiv: 2103.11965
cf. 2212.11387 AT+

Gauge covariant neural network

= trainable smearing

AT Y. Nagai arXiv: 2103.11965

Dictionary	(convolutional) Neural network	Gauge Covariant Neural network
Input	Image (2d data, structured)	gauge config (4d data, structured)
Output	Image (2d data, structured)	gauge config (4d data, structured)
Symmetry	Translation	Translation, rotation(90°), Gauge sym.
with Fixed param	Image filter	(APE/stout …) Smearing
Local operation	Summing up nearest neighbor with weights	Summing up staples with weights
Activation function	Tanh, ReLU, sigmoid, …	projection/normalization in Stout/HYP/HISQ
Formula for chain rule	Backprop	“Smeared force calculations” (Stout)
Training?	Backprop + Delta rule	AT Nagai 2103.11965

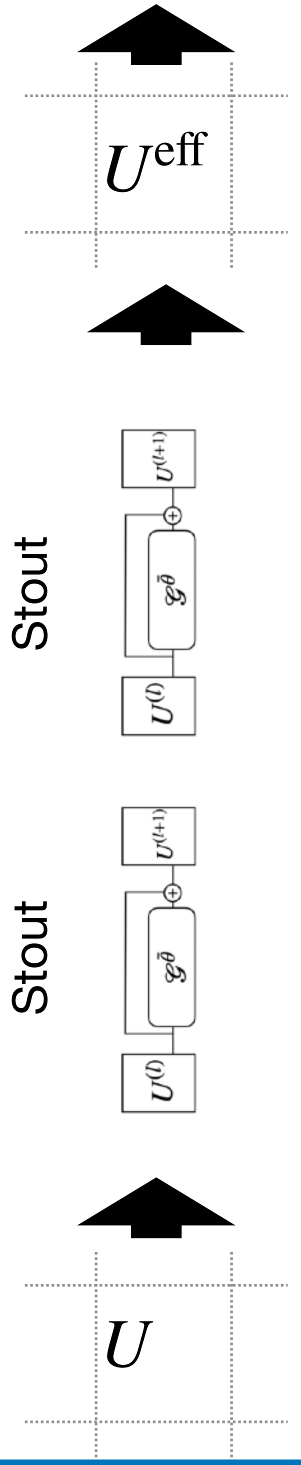
Well-known

(Index i in the neural net corresponds to n & μ in smearing. Information processing with NN is evolution of scalar field)

Gauge covariant neural net

Simulation parameter

Construct effective action using operators with U^{eff}



- Self-learning HMC (1909.02255, 2021 AT+), an exact algorithm
  - Exact Metropolis test and MD with effective action
- Target S : $m = 0.3$, dynamical staggered fermion, Nf=2, $L^4 = 4^4$, SU(2), $\beta = 2.7$
- Effective action in MD (S^{eff})
  - Same gauge action
  - $m_{\mathrm{eff}} = 0.4$ dynamical staggered fermion, Nf=2
  - U links are replaced by U^{eff} in D_{stag}
- “Adaptively reweighted HMC”

Details (skip)

Network: trainable stout (plaq+poly)

arXiv: 2103.11965

Structure of NN

(Polyakov loop+plaq in the stout-type)

$$\Omega_\mu^{(l)}(n) = \rho_{\mathrm{plaq}}^{(l)} O_\mu^{\mathrm{plaq}}(n) + \begin{cases} \rho_{\mathrm{poly},4}^{(l)} O_4^{\mathrm{poly}}(n) & (\mu = 4), \\ \rho_{\mathrm{poly,s}}^{(l)} O_i^{\mathrm{poly}}(n), & (\mu = i = 1, 2, 3) \end{cases}$$

All ρ is weight

O meas an loop operator

$$Q_\mu^{(l)}(n) = 2[\Omega_\mu^{(l)}(n)]_{\mathrm{TA}}$$

TA: Traceless, anti-hermitian operation

$$U_\mu^{(l+1)}(n) = \exp(Q_\mu^{(l)}(n)) U_\mu^{(l)}(n)$$

$$U_\mu^{\mathrm{NN}}(n)[U] = U_\mu^{(2)}(n)\left[U_\mu^{(1)}(n)\left[U_\mu(n)\right]\right]$$

2- layered stout
with 6 trainable parameters

Neural network Parametrized action:

$$S_\theta[U] = S_{\mathrm{g}}\left[U\right] + S_{\mathrm{f}}\left[\phi, U_\theta^{\mathrm{NN}}[U]; m_{\mathrm{h}} = 0.4\right],$$

Action for MD is built by gauge covariant NN

Loss function:

$$L_\theta[U] = \frac{1}{2}\left| S_\theta[U, \phi] - S[U, \phi] \right|^2,$$

Invariant under,
rot, transl, gauge trf.

Training strategy:

1.Train the network in prior HMC (online training+stochastic gr descent)
2.Perform SLHMC with fixed parameter

Details (skip)

Results: Loss decreases along with the training

arXiv: 2103.11965

Loss function:

$$L_\theta[U] = \frac{1}{2}\left| S_\theta[U,\phi] - S[U,\phi] \right|^2,$$

Intuitively, e^(-L) is understood as Boltzmann weight or reweighting factor.

Prior HMC run (training)

$$\frac{\partial S}{\partial \rho_i^{(l)}} = 2\,\mathrm{Re}\sum_{\mu',m}\mathrm{tr}\left[U_{\mu'}^{(l)\dagger}(m)\Lambda_{\mu',m}\frac{\partial C}{\partial \rho_i^{(l)}}\right]$$

$$\theta \leftarrow \theta - \eta\frac{\partial L_\theta(\mathcal{D})}{\partial \theta},$$

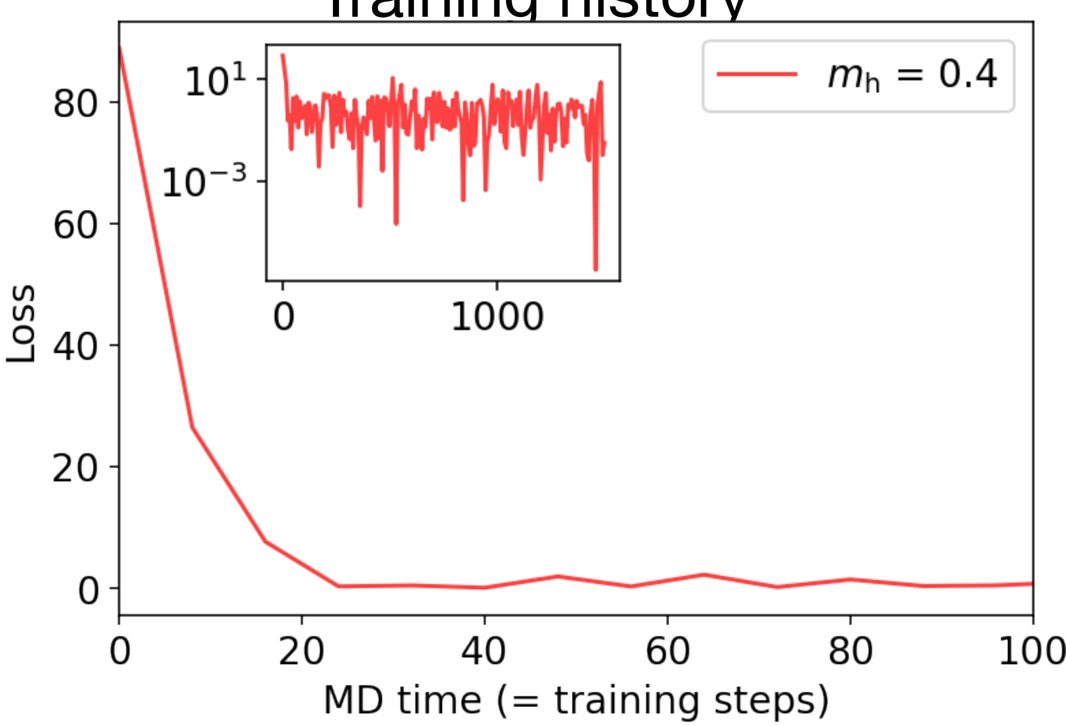
$$\frac{\partial L_\theta(\mathcal{D})}{\partial w_i^{(L-1)}} = \frac{\partial L_\theta(\mathcal{D})}{\partial S_\theta}\frac{\partial S_\theta}{\partial w_i^{(L-1)}}$$

Ω: sum of un-traced loops

C: one U removed Ω

Λ: A polynomial of U. (Same object in stout)

Training history



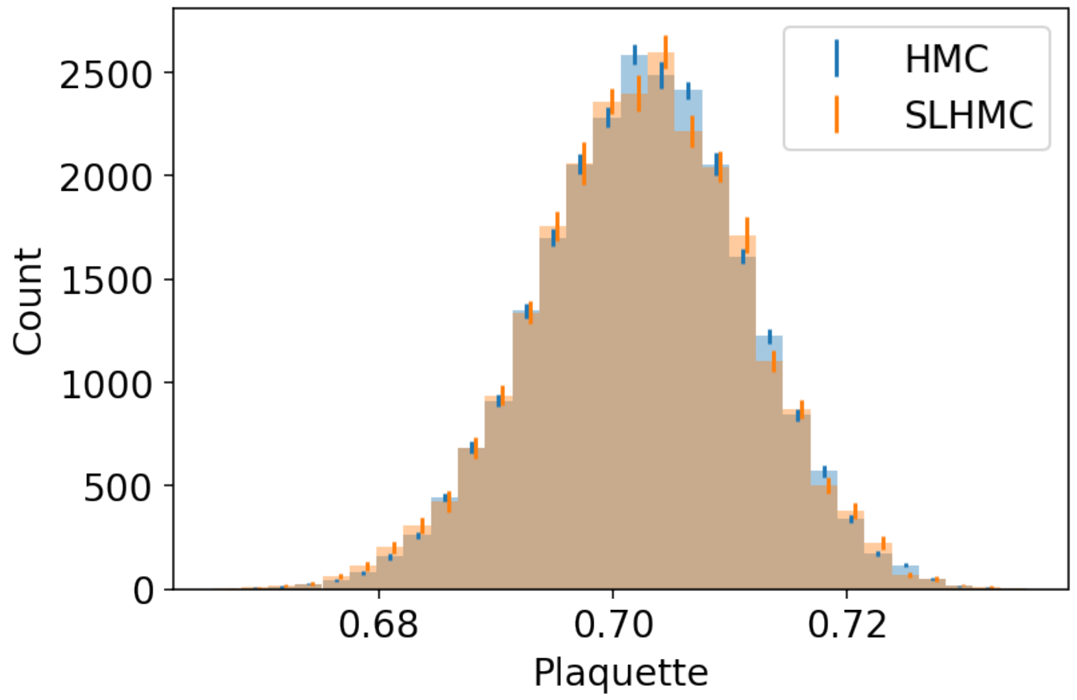
Without training, e^(-L)<< 1,
this means that candidate with approximated action never accept.
After training, e^(-L) ~1, and we get practical acceptance rate!

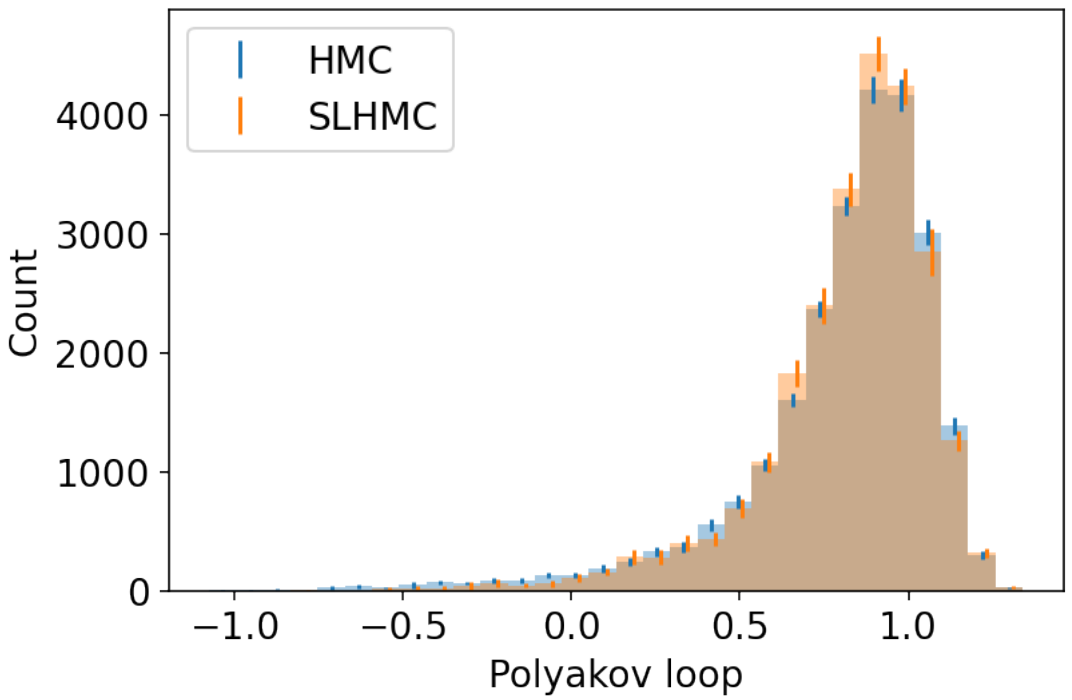
We perform SLHMC with these values!

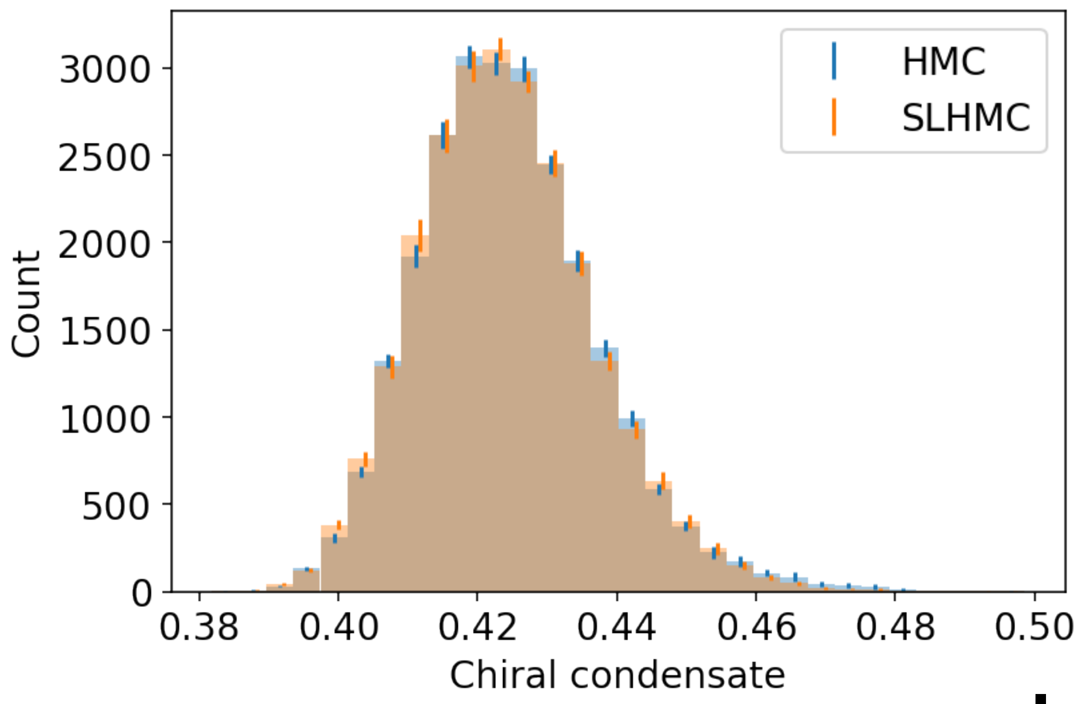
Application for the staggered in 4d

Results are consistent with each other (stout-type used)

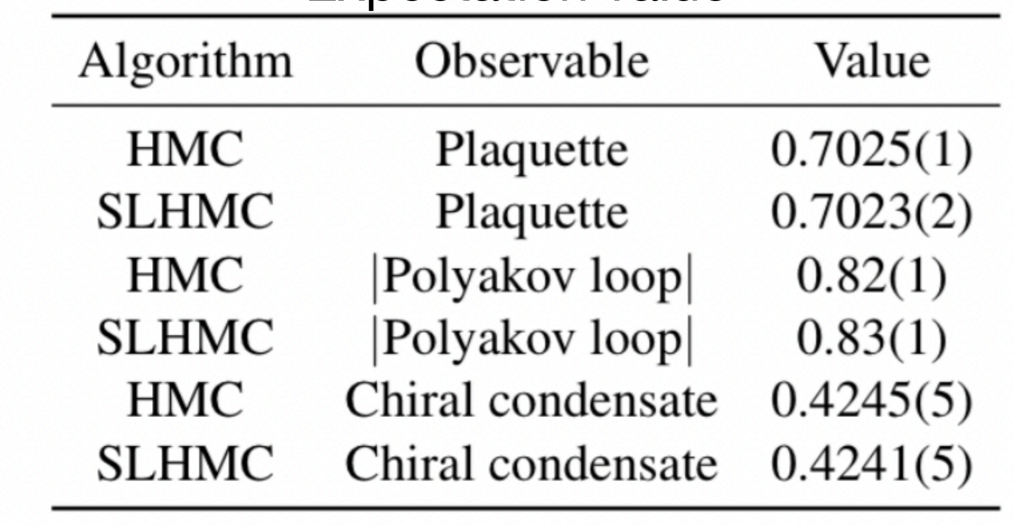
arXiv: 2103.11965







Expectation value



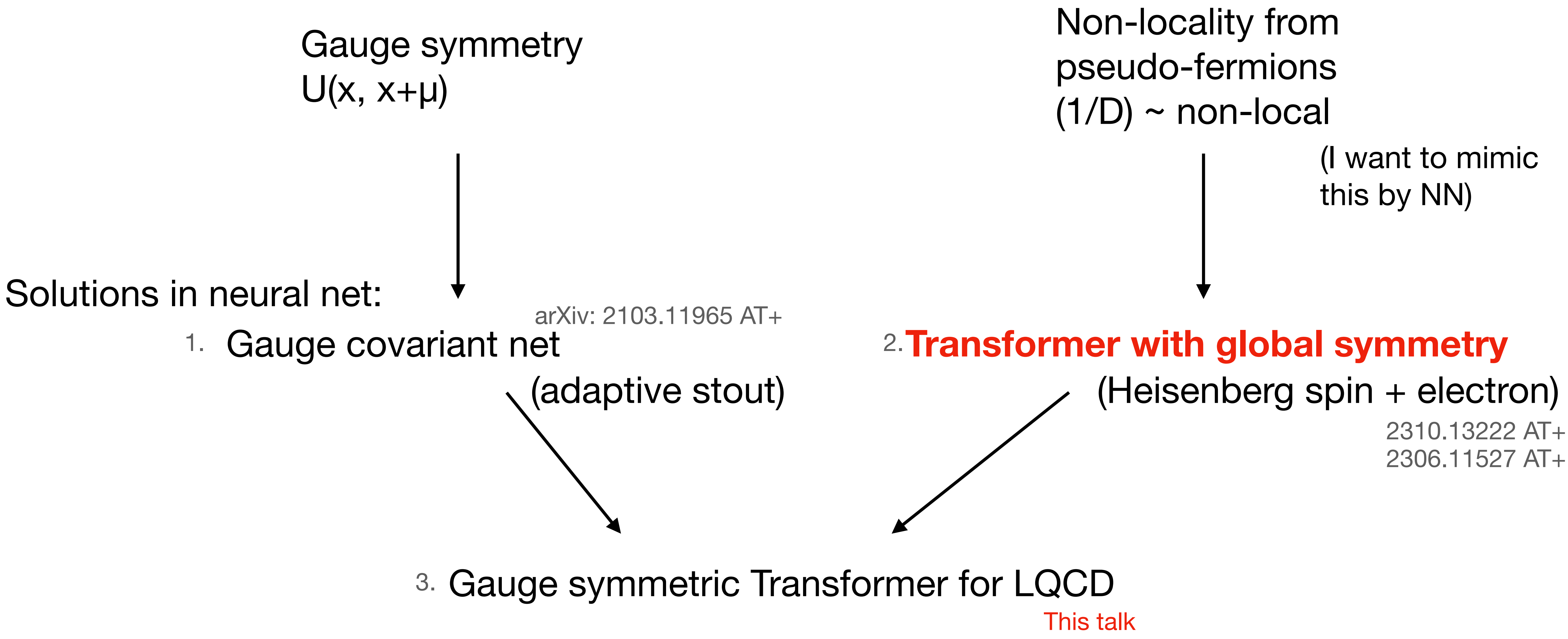
Algorithm	Observable	Value
HMC	Plaquette	0.7025(1)
SLHMC	Plaquette	0.7023(2)
HMC	|Polyakov loop|	0.82(1)
SLHMC	|Polyakov loop|	0.83(1)
HMC	Chiral condensate	0.4245(5)
SLHMC	Chiral condensate	0.4241(5)

Implemented by LatticeQCD.jl julia

Overview/outline

Gauge covariant transformer for LQCD

Two conditions/restrictions in LQCD:



Gauge symmetry
U(x, x+μ)

Non-locality from
pseudo-fermions
(1/D) ~ non-local

(I want to mimic
this by NN)

Solutions in neural net:

arXiv: 2103.11965 AT+

1. Gauge covariant net
(adaptive stout)

2. **Transformer with global symmetry**
(Heisenberg spin + electron)

2310.13222 AT+
2306.11527 AT+

3. Gauge symmetric Transformer for LQCD

This talk

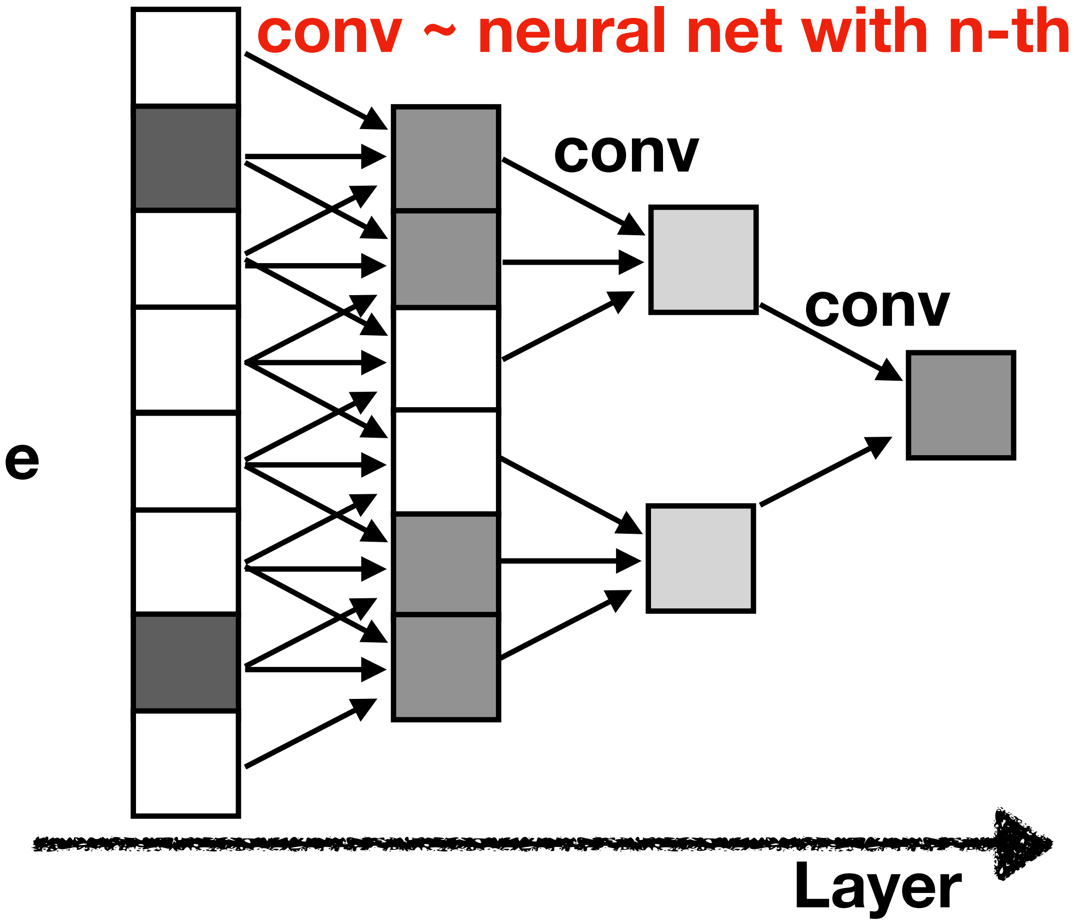
Equivariance and convolution

Convolutional Neural network have been good job but local

Convolutional neural layers in neural networks keep translational symmetry,
it can be generalized to any continuous/discrete symmetry in the theory. It helps generalization.

conv ~ neural net with n-th nearest neighbor connections (local)

e.g.
1d Input image



Distant correlations here can be captured by 3 steps of convolutional operation (Repetition of local operation)

However, 1 step of convolutional layer can pick up only local correlation and representability of neural networks is limited. Global correlations are sometimes important.
How can we overcome these difficulties?

Transformer and Attention

Attention layer used in Transformers (GPT, Bard)

Akio Tomiya

arXiv:1706.03762

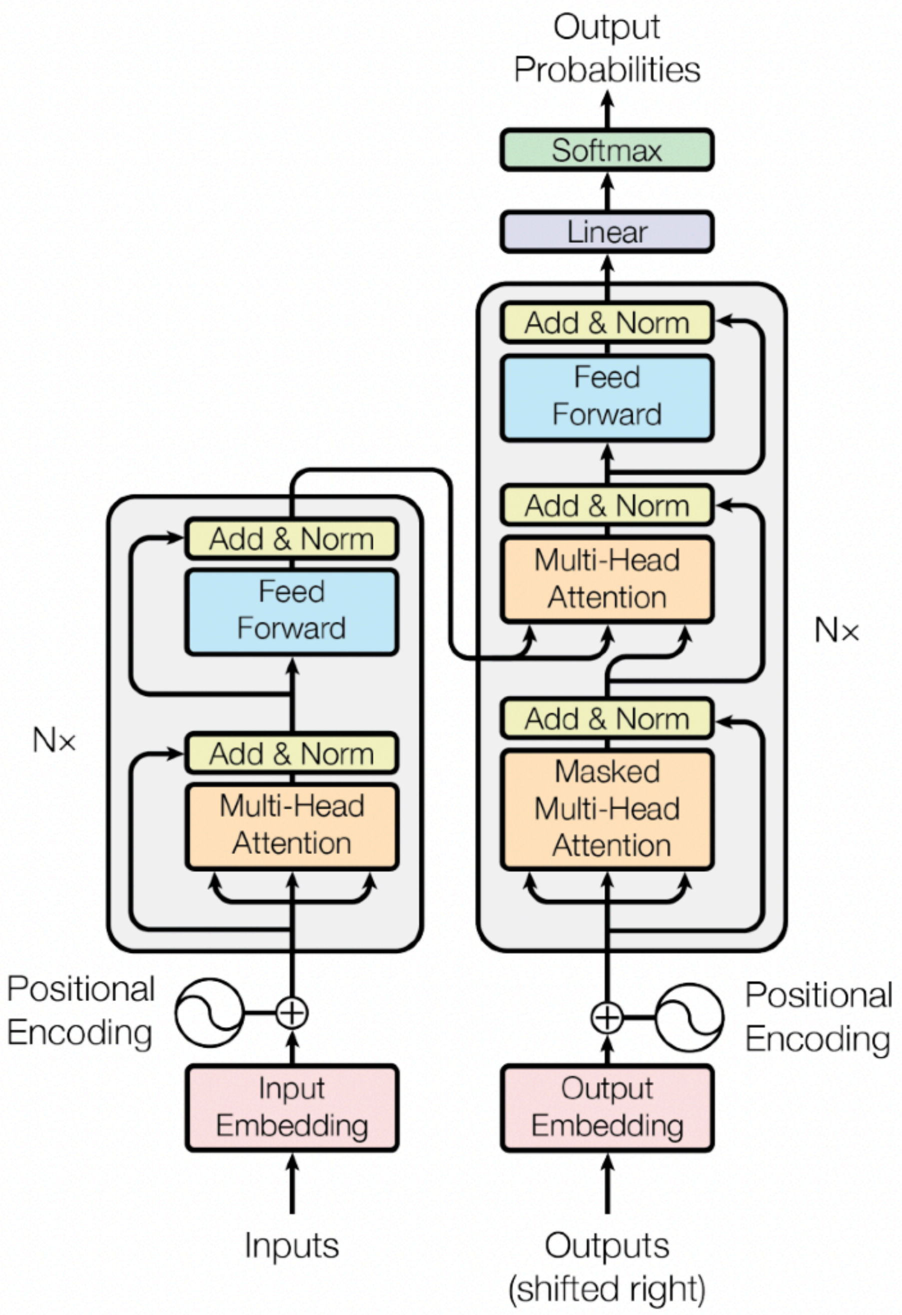


Figure 1: The Transformer - model architecture.

OpenAI ChatGPT

Gemini

DeepL

Attention layer (in transformer model) has been introduced in a paper titled
“Attention is all you need” (1706.03762)
State of the art architecture of language processing.
Attention layer is essential.

Transformer and Attention

Attention layer can capture non-local correlations

arXiv:1706.03762

Modifier in language can be non-local

Eg. I am Akio Tomiya living in Japan, who studies machine learning and physics

In physics terminology, this is **non local correlation.**

The attention layer enables us to treat non-local correlation with a neural net!

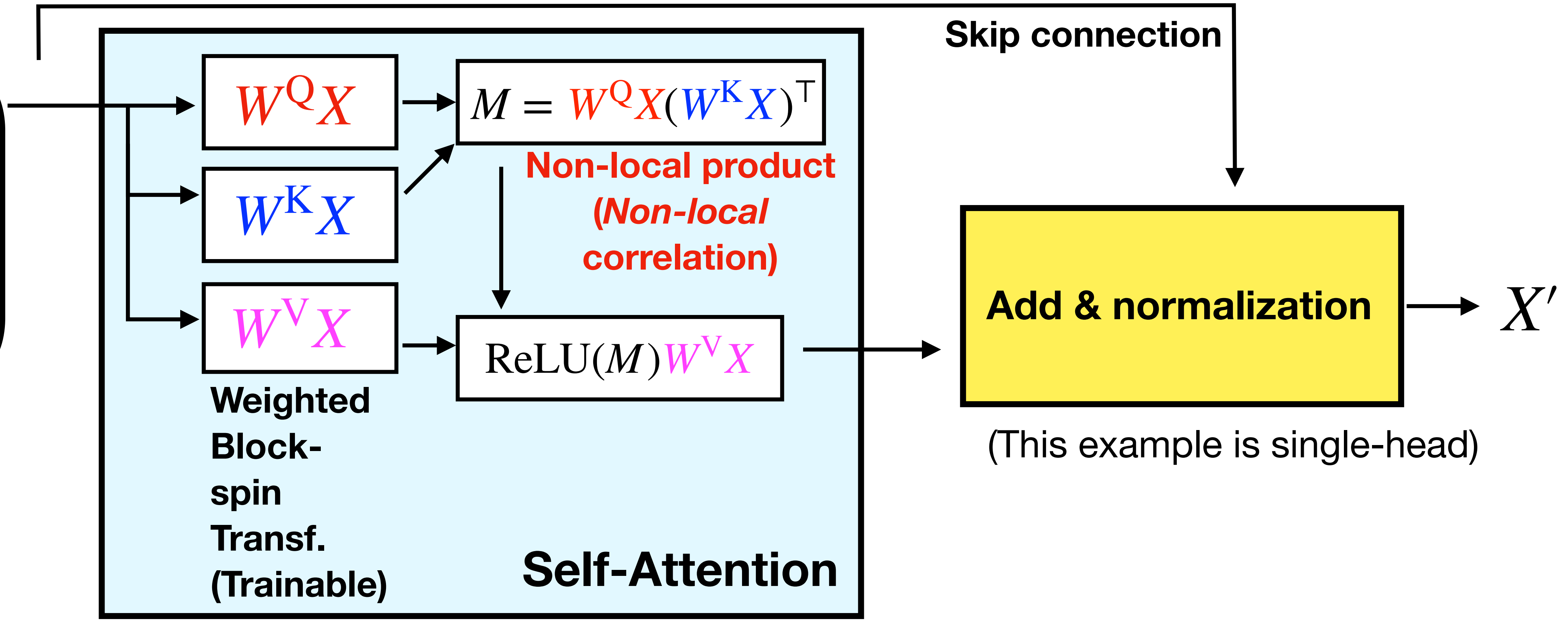
Simplified version of Attention/Transformer

$$X = \begin{pmatrix} \mathrm{I} \\ \mathrm{am} \\ \mathrm{Akio} \\ \vdots \end{pmatrix}$$

Array of word vectors

Word~vector

X: matrix


Skip connection
$W^{\mathrm{Q}}X$
$W^{\mathrm{K}}X$
$W^{\mathrm{V}}X$
$M = W^{\mathrm{Q}}X(W^{\mathrm{K}}X)^{\top}$
Non-local product (Non-local correlation)
$\mathrm{ReLU}(M)W^{\mathrm{V}}X$
Weighted Block-spin Transf. (Trainable)
Self-Attention
Add & normalization
X'
(This example is single-head)

These can be repeated

Transformer and Attention

Transformer shows scaling lows (power law)

arXiv: 2001.08361

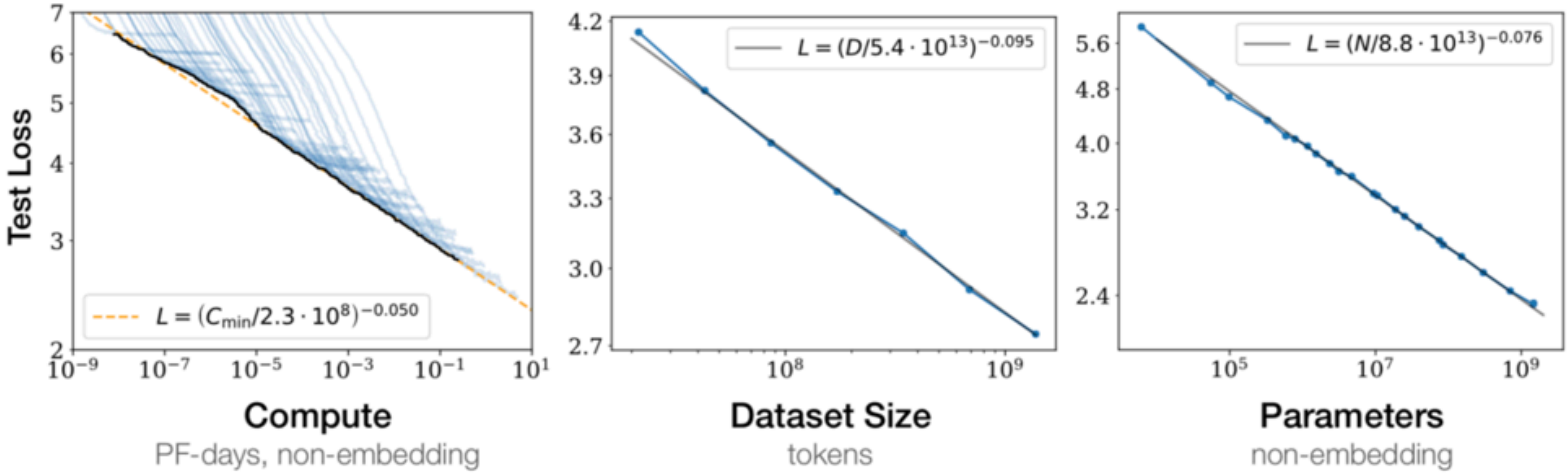


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

- **Transformers requires huge data**
  (e.g. GPT uses all electric books in the world)
  Because it has few inductive bias (no equivariance)
- **It can be improved systematically**

Transformer and Attention

Physically symmetric Attention layer

Attention layer can capture global correlation
Equivariance reduces data demands for training

	Equivariance	**Capturable correlation**	**Data demmands**	**Applications**
Convolution (∈ equivariant layers)	Yes 👍	Local 😲	Low 👍	Image recognition VAE, GAN Normalizing flow
Standard Attention layer	No 😲	Global 👍	Huge 😲	ChatGPT GEMINI Vision Transformer arXiv:1706.03762
Physically *Equivariant* attention layer	Yes 👍	Global 👍	?	Kondo system (this work) arXiv: 2306.11527

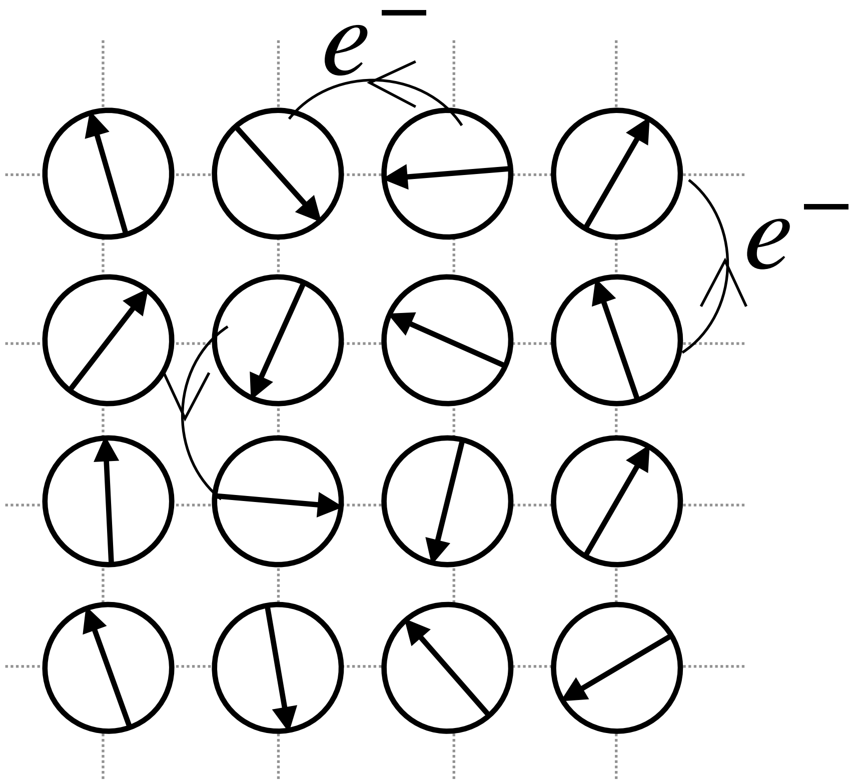
Self-learning Monte-Carlo

Target: Double exchange model

Target system: Classical Heisenberg spin $\mathbf{S}_i$+ Fermion on 2d lattice

$$H = -t \sum_{\alpha,\langle i,j \rangle} (\hat{c}^\dagger_{i\alpha} \hat{c}_{j\alpha} + \mathrm{h.c.}) + \frac{J}{2} \sum_i \mathbf{S}_i \cdot \hat{\sigma}_i$$

(Kondo model)



3d vectors on 2d lattice
Anti-ferro magnet

Two different phases
- Anti-ferromagnet (~staggered mag)
- Paramagnet (~normal metal)

(This system is similar to lattice QCD but easier)

Self-learning Monte-Carlo

Previous work

Target system: Classical Heisenberg spin $\mathbf{S}_i$+ Fermion on 2d lattice

$$H = -t \sum_{\alpha,\langle i,j \rangle} (\hat{c}^\dagger_{i\alpha} \hat{c}_{j\alpha} + \mathrm{h.c.}) + \frac{J}{2} \sum_i \mathbf{S}_i \cdot \hat{\sigma}_i$$ (Kondo model)

Naive effective model:

$$H^{\mathrm{Linear}}_{\mathrm{eff}} = - \sum_{\langle i,j \rangle_n} J^{\mathrm{eff}}_n \mathbf{S}_i \cdot \mathbf{S}_j + E_0$$

J^{eff}_n: n-th nearest neighbor

J^{eff}_n is determined by regression (training) to improve approximation

Self-learning Monte-Carlo:

Update with H_{eff} , and Metropolis-Hastings with H & H_{eff}

Cancel inexactness. This is an exact algorithms

Self-learning Monte-Carlo

Previous work

Target system: Classical Heisenberg spin S_i+ Fermion on 2d lattice

$$H = -t \sum_{\alpha,\langle i,j\rangle} (\hat{c}^\dagger_{i\alpha}\hat{c}_{j\alpha} + \mathrm{h.c.}) + \frac{J}{2}\sum_i \mathbf{S}_i \cdot \hat{\sigma}_i$$

(Kondo model)

This has fermion det

Naive effective model:

$$H^{\mathrm{Linear}}_{\mathrm{eff}} = -\sum_{\langle i,j\rangle_n} J^{\mathrm{eff}}_n \mathbf{S}_i \cdot \mathbf{S}_j + E_0$$

J^{eff}_n: n-th nearest neighbor

$$\longrightarrow \quad H_{\mathrm{eff}} = -\sum_{\langle i,j\rangle_n} J^{\mathrm{eff}}_n \mathbf{S}^{\mathrm{NN}}_i \cdot \mathbf{S}^{\mathrm{NN}}_j + E_0$$

We replace this by “translated” spin S^{NN}_i with a transformer and used in self-learning MC

mimics effects from fermions with smeared spins

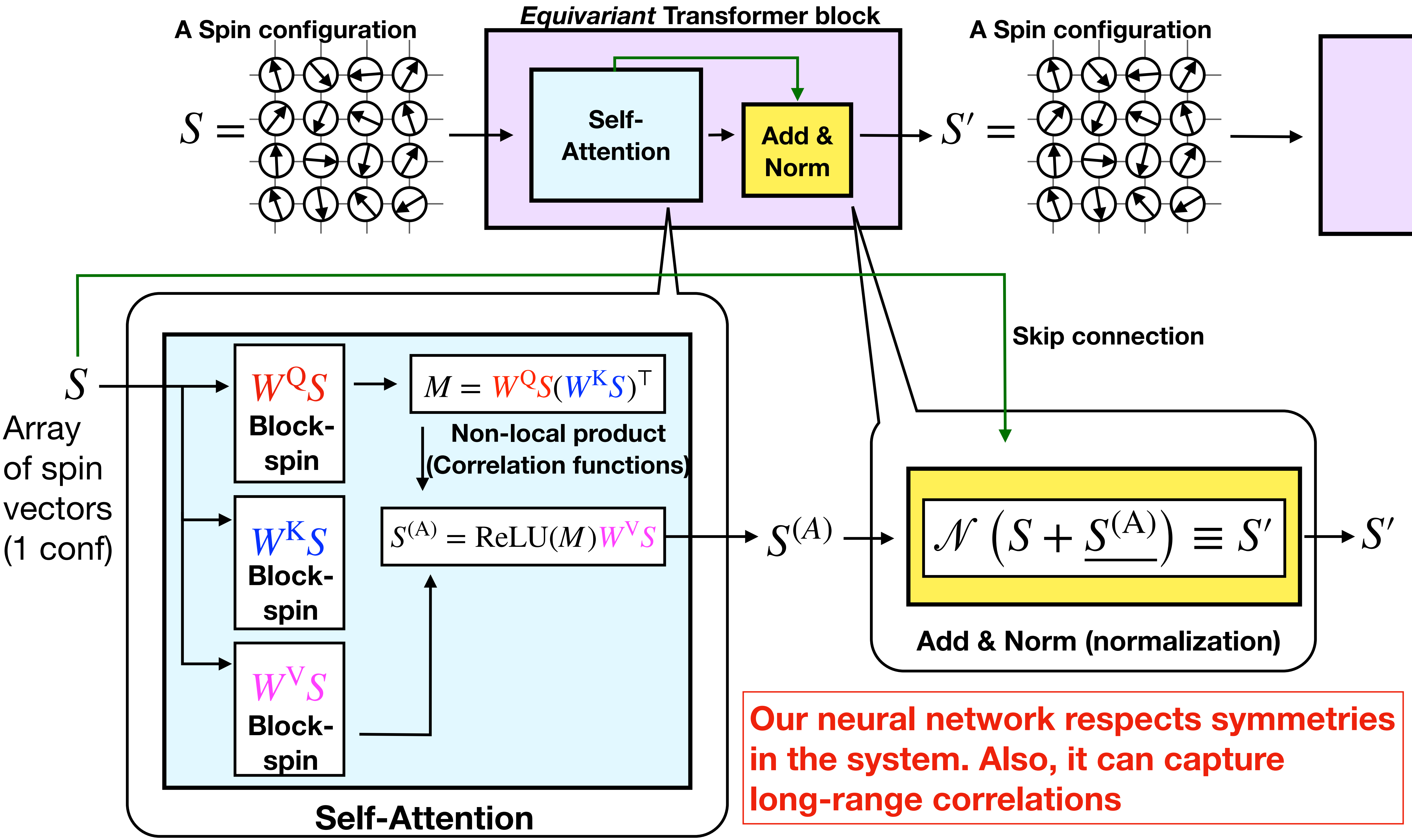
This doesn’t have fermion det

Self-learning Monte-Carlo

Physically equivariant Attention layer/Transformer

Akio Tomiya

arXiv: 2306.11527.



A Spin configuration

$S =$

Equivariant Transformer block

Self-Attention

Add & Norm

A Spin configuration

$S' =$

S

Array of spin vectors (1 conf)

$W^{\mathrm{Q}}S$ Block-spin

$W^{\mathrm{K}}S$ Block-spin

$W^{\mathrm{V}}S$ Block-spin

$M = W^{\mathrm{Q}}S(W^{\mathrm{K}}S)^{\top}$

Non-local product (Correlation functions)

$S^{(\mathrm{A})} = \mathrm{ReLU}(M)W^{\mathrm{V}}S$

Self-Attention

$S^{(A)}$

Skip connection

$\mathcal{N}\left(S + \underline{S^{(\mathrm{A})}}\right) \equiv S'$

S'

Add & Norm (normalization)

Our neural network respects symmetries in the system. Also, it can capture long-range correlations

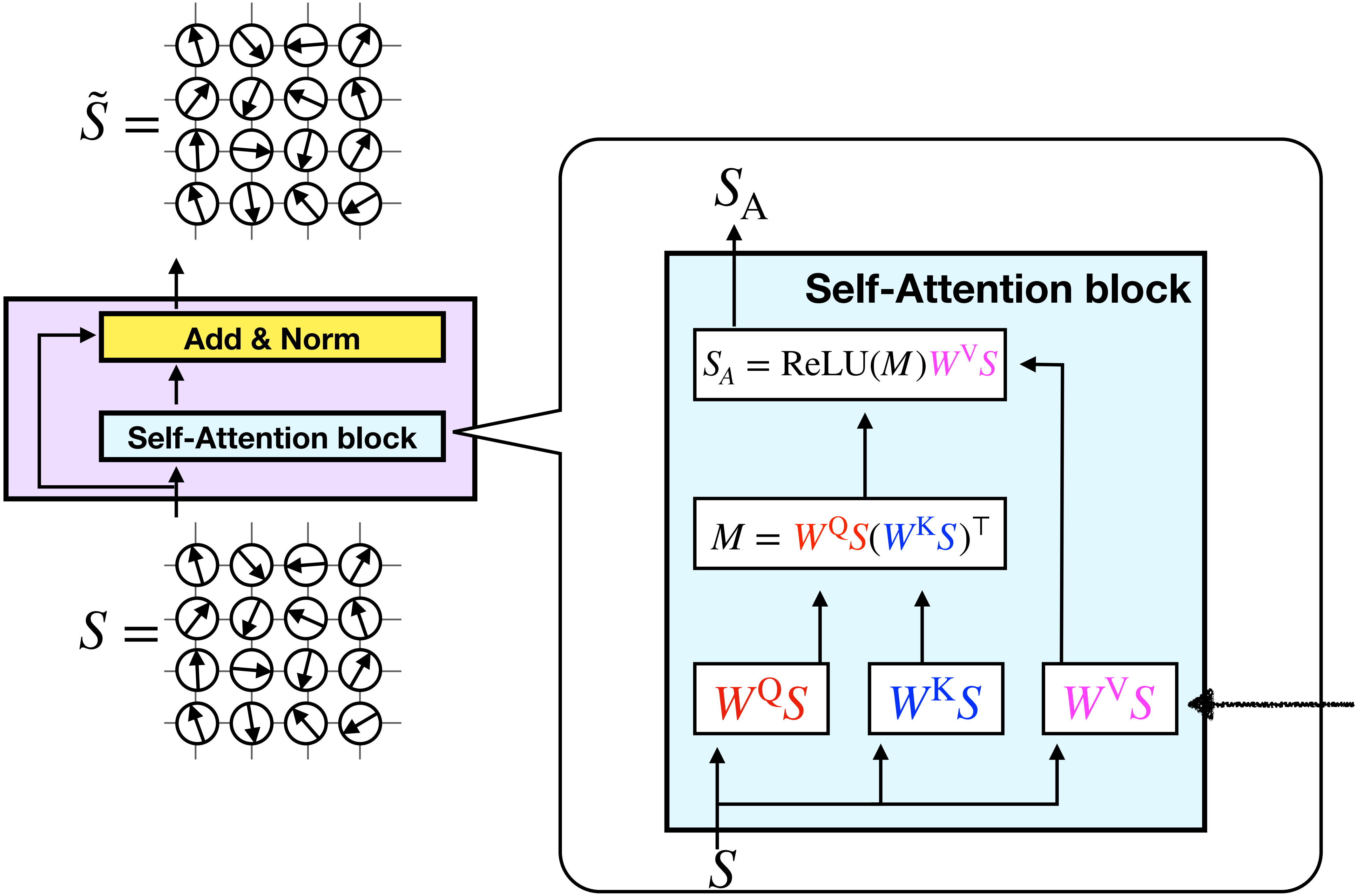
Equivariant attention

Self-learning Monte-Carlo

Akio Tomiya

arXiv: 2306.11527.

Attention block makes effective spin field with non-local BST



$\tilde{S} =$

Add & Norm

Self-Attention block

S_{A}

Self-Attention block

$S_A = \mathrm{ReLU}(M) W^{\mathrm{V}} S$

$M = W^{\mathrm{Q}} S (W^{\mathrm{K}} S)^{\top}$

$W^{\mathrm{Q}} S$

$W^{\mathrm{K}} S$

$W^{\mathrm{V}} S$

S

$S =$

Smearing (BST)

Rot. equivariant

Trsl. equivariant

trainable!

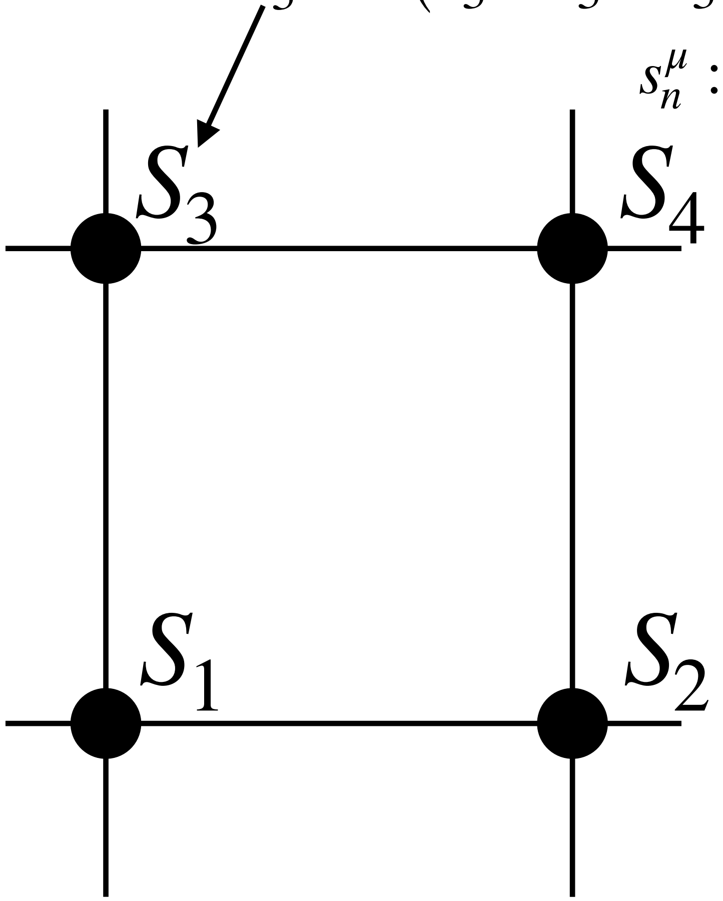
Self-learning Monte-Carlo

Equivariant under spin-rotation & translation

arXiv: 2306.11527.

$S_3^\top = \begin{pmatrix} s_3^1 & s_3^2 & s_3^3 \end{pmatrix}^\top$

s_n^μ : O(3) vector



$$\mathbf{S} = \begin{pmatrix} S_1^\top & S_2^\top & S_3^\top & S_4^\top \end{pmatrix}^\top$$

- Local weighted sum over neighbors
= "Smeared spin" with parameters
~ "Block spin sum" with parameters

$$\tilde{S}_i^\alpha = \sum_{l=0} w_l^\alpha S_{i+l} \qquad \alpha = \mathrm{Q, K, V}$$

$w_l^\alpha \in \mathbb{R}$: trainable

Translationally equivariant
Rotationally equivariant

$$S_i^\top = \begin{pmatrix} s_i^1 & s_i^2 & s_i^3 \end{pmatrix}^\top$$

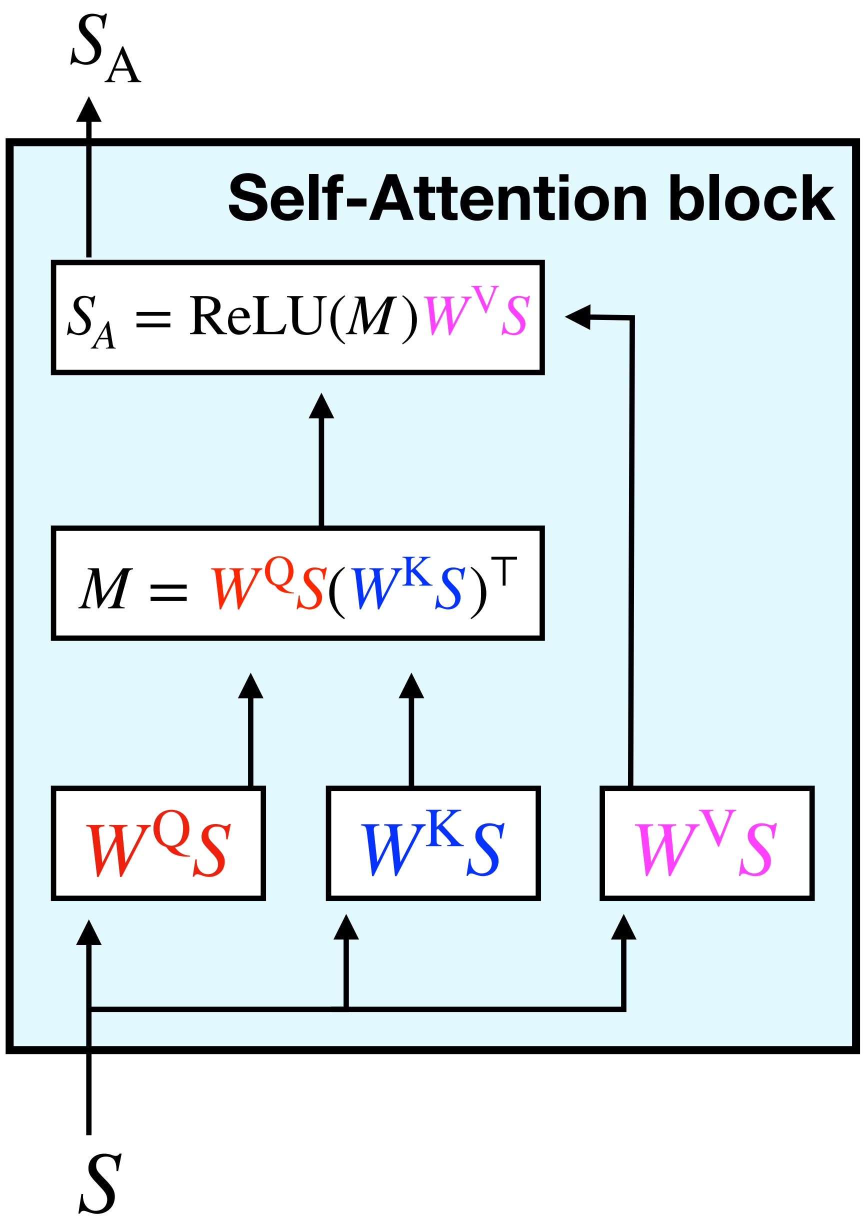
$$|S_i| = \sqrt{(s_i^1)^2 + (s_i^2)^2 + (s_i^3)^2}$$
$$= 1$$

3 component scalar, normalized

Self-learning Monte-Carlo

Equivariant under spin-rotation & translation

arXiv: 2306.11527.



S_{A}

Self-Attention block

$S_A = \mathrm{ReLU}(M) W^{\mathrm{V}} S$

$M = W^{\mathrm{Q}} S (W^{\mathrm{K}} S)^\top$

$W^{\mathrm{Q}} S$ $W^{\mathrm{K}} S$ $W^{\mathrm{V}} S$

S

$$\mathbf{S} = \begin{pmatrix} S_1^\top & S_2^\top & S_3^\top & S_4^\top \end{pmatrix}^\top$$

$$S_i^\top = \begin{pmatrix} s_i^1 & s_i^2 & s_i^3 \end{pmatrix}^\top$$

s_n^μ : O(3) vector

$$\tilde{S}_i^\alpha = W^\alpha S = \sum w_l^\alpha S_{i+l}$$

"averaged spin" by neighbors

Gram matrix with averaged spin

$$M = \tilde{G}^\alpha \equiv (\tilde{\mathbf{S}}^\alpha)^\top \tilde{\mathbf{S}}^\alpha \quad \alpha = \mathrm{Q}, \mathrm{K}, \mathrm{V}$$

$$G \equiv \mathbf{S}^\top \mathbf{S} = \begin{pmatrix} S_1^\top S_1 & S_1^\top S_2 & S_1^\top S_3 & S_1^\top S_4 \\ S_2^\top S_1 & S_2^\top S_2 & S_2^\top S_3 & S_2^\top S_4 \\ S_3^\top S_1 & S_3^\top S_2 & S_3^\top S_3 & S_3^\top S_4 \\ S_4^\top S_1 & S_4^\top S_2 & S_4^\top S_3 & S_4^\top S_4 \end{pmatrix}$$

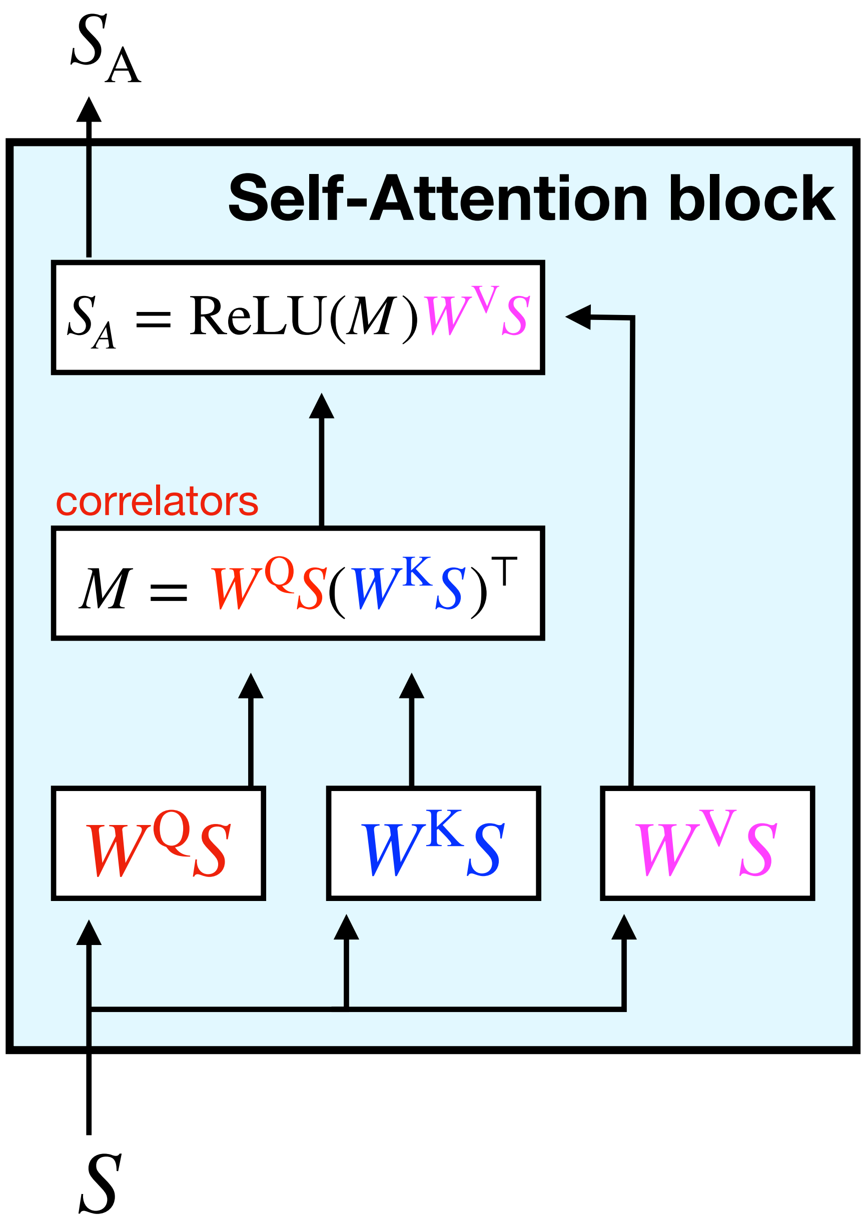
Translationally covariant, Rotationally invariant

A set of correlators

Self-learning Monte-Carlo

Equivariant under spin-rotation & translation

arXiv: 2306.11527.



$$\mathbf{S} = \begin{pmatrix} S_1^\top & S_2^\top & S_3^\top & S_4^\top \end{pmatrix}^\top$$

$$S_i^\top = \begin{pmatrix} s_i^1 & s_i^2 & s_i^3 \end{pmatrix}^\top$$

s_n^μ : O(3) vecto

$$\tilde{S}_i^\alpha = W^\alpha S = \sum w_l^\alpha S_{i+l}$$

"averaged spin" by neighbors

Gram matrix with averaged spin

$$M = \tilde{G}^\alpha \equiv (\tilde{\mathbf{S}}^\alpha)^\top \tilde{\mathbf{S}}^\alpha \quad \alpha = \mathrm{Q}, \mathrm{K}, \mathrm{V}$$

Translationally covariant
Rotationally invariant

$$S_A = \mathrm{ReLU}(M) W^{\mathrm{V}} S$$

$$= \mathrm{ReLU}(M) \tilde{S}^{\mathrm{V}}$$

A set of correlators

Self-learning Monte-Carlo

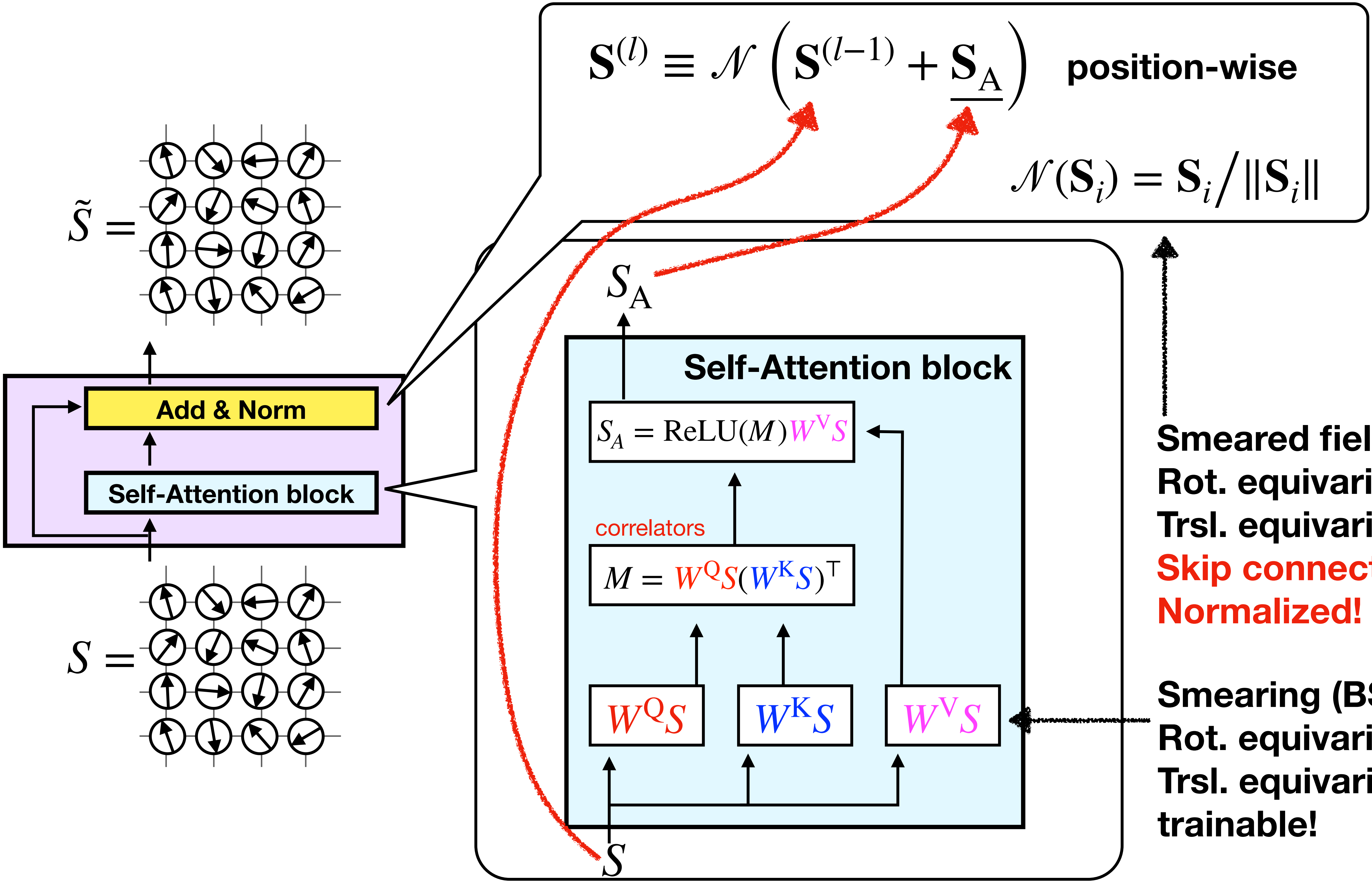
Akio Tomiya

arXiv: 2306.11527.

Attention block makes effective spin field with non-local BST

$$\mathbf{S}^{(l)} \equiv \mathcal{N}\left(\mathbf{S}^{(l-1)} + \underline{\mathbf{S}_{\mathrm{A}}}\right)$$ position-wise

$$\mathcal{N}(\mathbf{S}_i) = \mathbf{S}_i / \|\mathbf{S}_i\|$$



Smeared fields
Rot. equivariant
Trsl. equivariant
Skip connection
Normalized!

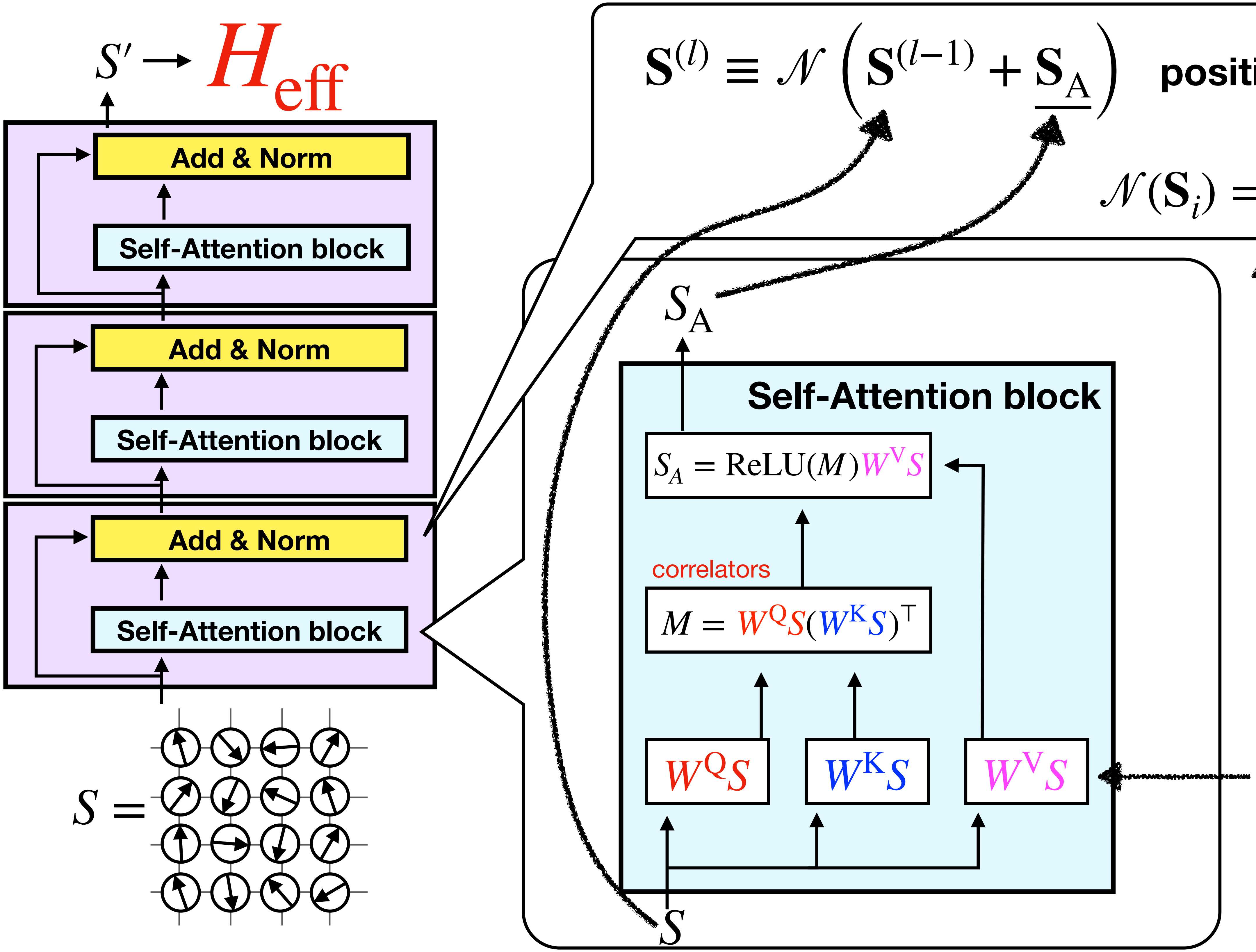
Smearing (BST)
Rot. equivariant
Trsl. equivariant
trainable!

Self-learning Monte-Carlo

Akio Tomiya

arXiv: 2306.11527.

Variational Hamiltonian with Equivariant Attention layers



$$\mathbf{S}^{(l)} \equiv \mathcal{N}\left(\mathbf{S}^{(l-1)} + \underline{\mathbf{S}_{\rm A}}\right)$$

position-wise

$$\mathcal{N}(\mathbf{S}_i) = \mathbf{S}_i / \|\mathbf{S}_i\|$$

Smeared fields
Rot. equivariant
Trsl. equivariant
Skip connection
Normalized!

Smearing (BST)
Rot. equivariant
Trsl. equivariant
trainable!

Self-learning Monte-Carlo

SLMC = MCMC with an effective model/ Adaptive rew

arXiv:1610.03137+

For statistical spin system, we want to calculate expectation value with

$$W(\{\mathbf{S}\}) \propto \exp[-\beta H(\{\mathbf{S}\})]$$

On the other hand, an effective model $H_{\text{eff}}(\{\mathbf{S}\})$ can compose in MCMC,

$\{\mathbf{S}\} \xrightarrow{\text{eff}} \{\mathbf{S}\} \xrightarrow{\text{eff}} \{\mathbf{S}\} \xrightarrow{\text{eff}} \{\mathbf{S}\}$ this distributes $W_{\text{eff}}(\{\mathbf{S}\}) \propto \exp[-\beta H_{\text{eff}}(\{\mathbf{S}\})]$

if the update「→」 satisfies the detailed balance. We can employ Metropolis test like

$$A_{\text{eff}}(\{\mathbf{S}'\}, \{\mathbf{S}\}) = \min\left(1, W_{\text{eff}}(\{\mathbf{S}'\})/W_{\text{eff}}(\{\mathbf{S}\})\right).$$

Self-learning Monte-Carlo

Akio Tomiya

SLMC = MCMC with an effective model/ Adaptive rew

arXiv:1610.03137+

For statistical spin system, we want to calculate expectation value with

$$W(\{\mathbf{S}\}) \propto \exp[-\beta H(\{\mathbf{S}\})]$$

On the other hand, an effective model $H_{\text{eff}}(\{\mathbf{S}\})$ can compose in MCMC,

$\{\mathbf{S}\} \xrightarrow{\text{eff}} \{\mathbf{S}\} \xrightarrow{\text{eff}} \{\mathbf{S}\} \xrightarrow{\text{eff}} \{\mathbf{S}\}$ this distributes $W_{\text{eff}}(\{\mathbf{S}\}) \propto \exp[-\beta H_{\text{eff}}(\{\mathbf{S}\})]$

if the update「→」 satisfies the detailed balance. We can employ Metropolis test like

$$A_{\text{eff}}(\{\mathbf{S}'\},\{\mathbf{S}\}) = \min\left(1, W_{\text{eff}}(\{\mathbf{S}'\})/W_{\text{eff}}(\{\mathbf{S}\})\right).$$

SLMC: Self-learning Monte-Carlo

We can construct *double* MCMC with $H(\{\mathbf{S}\})$ and $H_{\text{eff}}(\{\mathbf{S}\})$

$$\{\mathbf{S}\} \xrightarrow{\text{eff}} \{\mathbf{S}\} \xrightarrow{\text{eff}} \{\mathbf{S}\} \xrightarrow{\text{eff}} \{\mathbf{S}\} \longrightarrow \{\mathbf{S}\} \xrightarrow{\text{eff}} \{\mathbf{S}\} \xrightarrow{\text{eff}} \{\mathbf{S}\} \xrightarrow{\text{eff}} \{\mathbf{S}\} \longrightarrow$$

with Metropolis-*Hastings* test: $A(\{\mathbf{S}'\},\{\mathbf{S}\}) = \min\left(1, \dfrac{W(\{\mathbf{S}'\})}{W(\{\mathbf{S}\})}\dfrac{W_{\text{eff}}(\{\mathbf{S}\})}{W_{\text{eff}}(\{\mathbf{S}'\})}\right).$

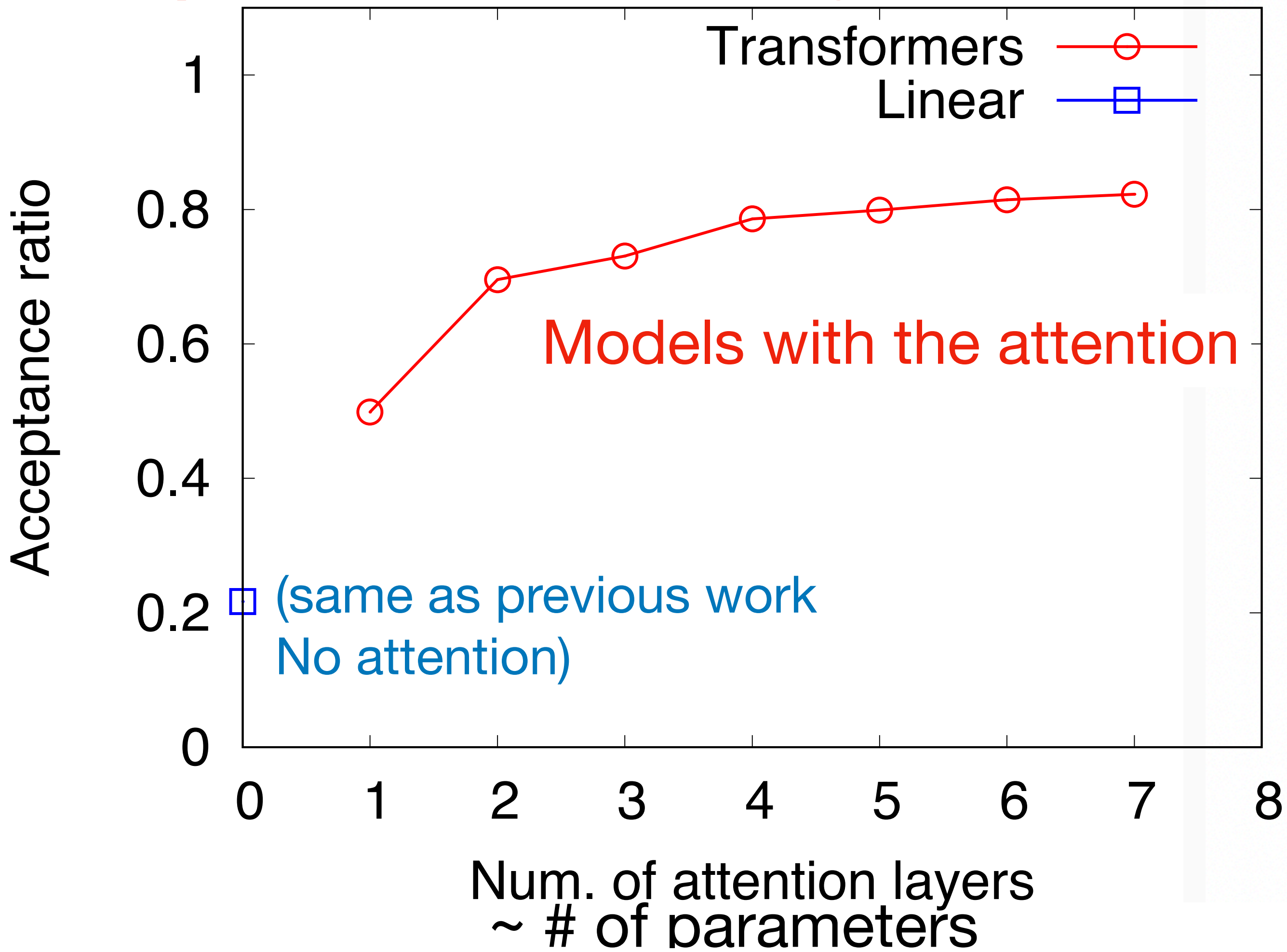
- **Effective model can have fit parameters**
- **Exact! It satisfies detailed balance with $W(\{\mathbf{S}\})$ (exact)**
- **It has been used for full QCD too (arXiv: 2010.11900, 2103.11965)**

Transformer and Attention

Akio Tomiya
arXiv: 2306.11527 + update

Application to O(3) spin model with fermions

Acceptance rate ~ efficiency



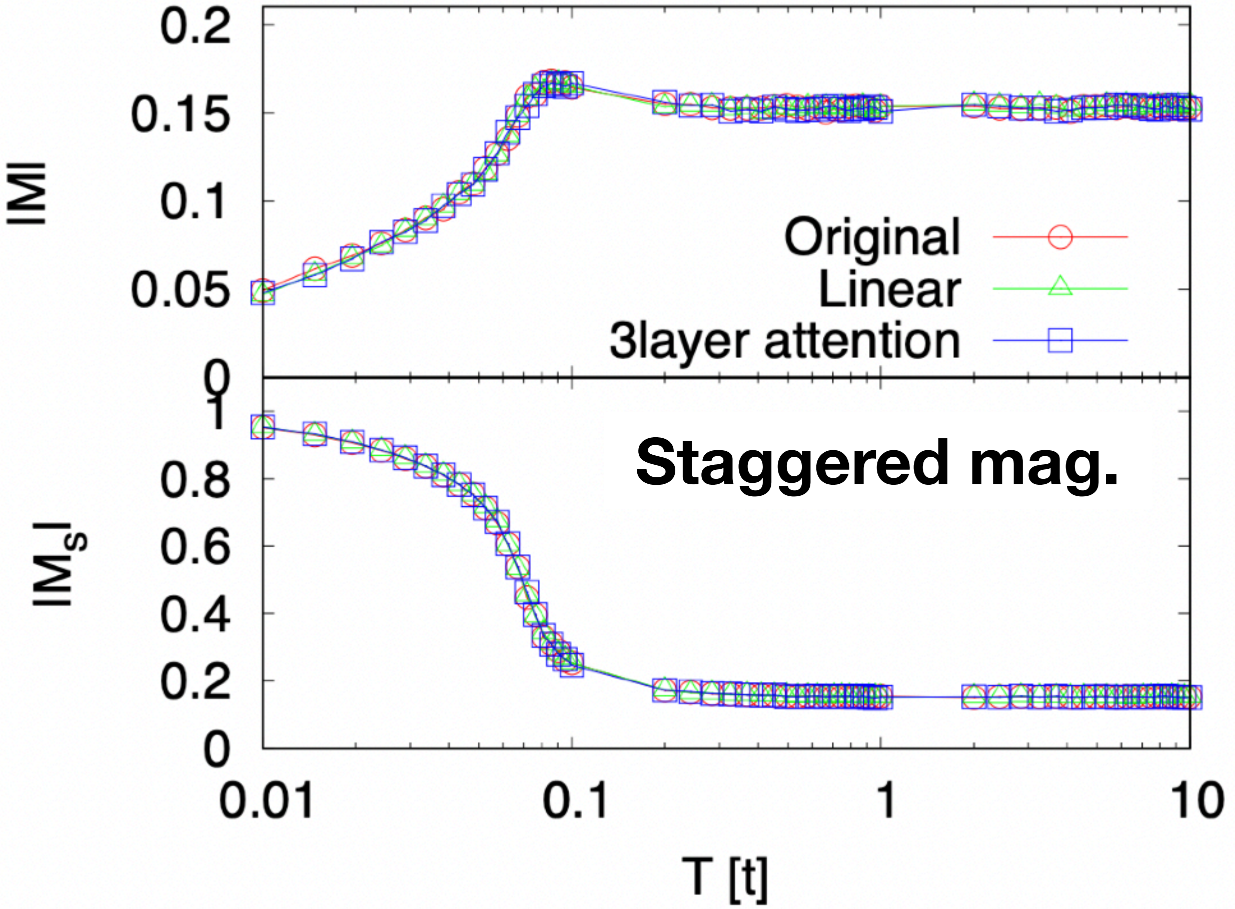
Models with the attention

(same as previous work
No attention)

Note: As far as we tested,
CNN-type does not work in this case.

No improvements with increase of layers.
(Global correlations of fermions from
Fermi-Dirac statistics make acceptance bad?)

Observables



Physical values are consistent
(as we expected)

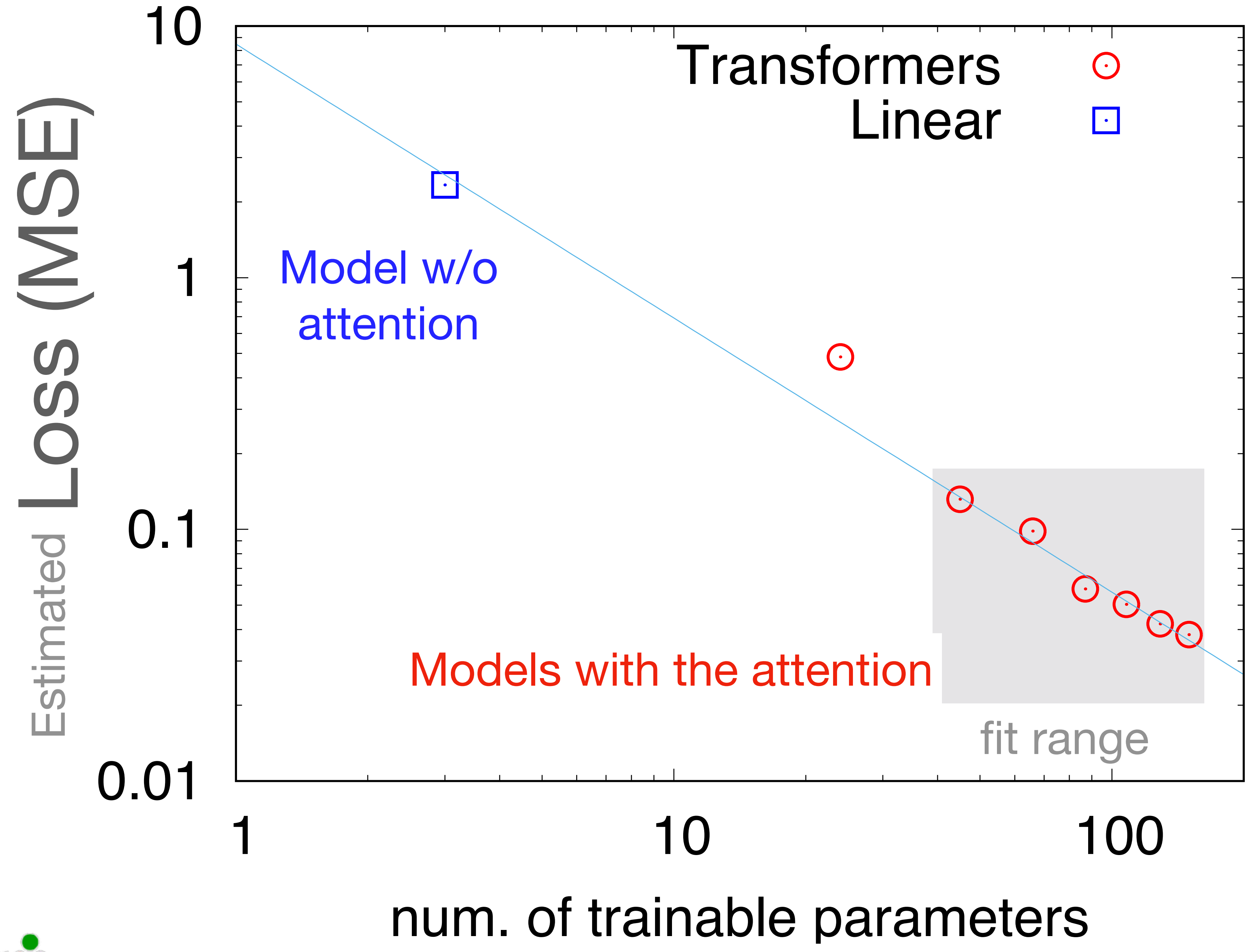
Nx=Ny=6
(Lattice sites)

Transformer and Attention

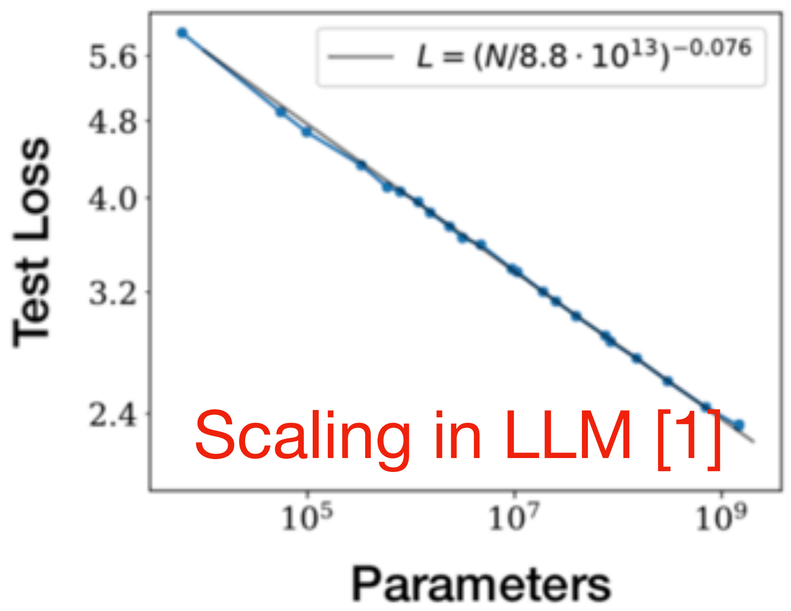
Loss function shows Power-type scaling law as LLM

arXiv: 2306.11527 + update

$$\text{Acceptance rate} = \exp\left(-\sqrt{\text{MSE}}\right)$$



(1 layer ~ 30 parameters)



Scaling in LLM [1]

Line is just for guiding eyes (no meaning)

fit ~(7.1/x)^(1.1)

[1] arXiv:2001.08361

Gauge covariant transformer (*CASK*)

Work in progress

A. Tomiya, H. Ohno, Y. Nagai

Lattice 2024
Jul 29 (Mon), 2024, 11:55 AM
Algorithms and artificial intelligence

Overview/outline

Gauge covariant transformer for LQCD

Two conditions/restrictions in LQCD:

Gauge symmetry
U(x, x+μ)

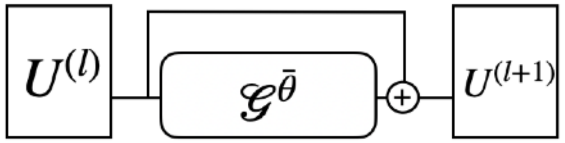
Non-locality from
pseudo-fermions
(1/D) ~ non-local

(I want to mimic
this by NN)

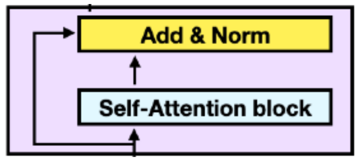
Solutions in neural net:

arXiv: 2103.11965 AT+

1. Gauge covariant net
(adaptive stout)



2. Transformer with global symmetry
(Heisenberg spin + electron)



2310.13222 AT+
2306.11527 AT+

3. **Gauge symmetric Transformer for LQCD**

This talk

Gauge covariant transformer

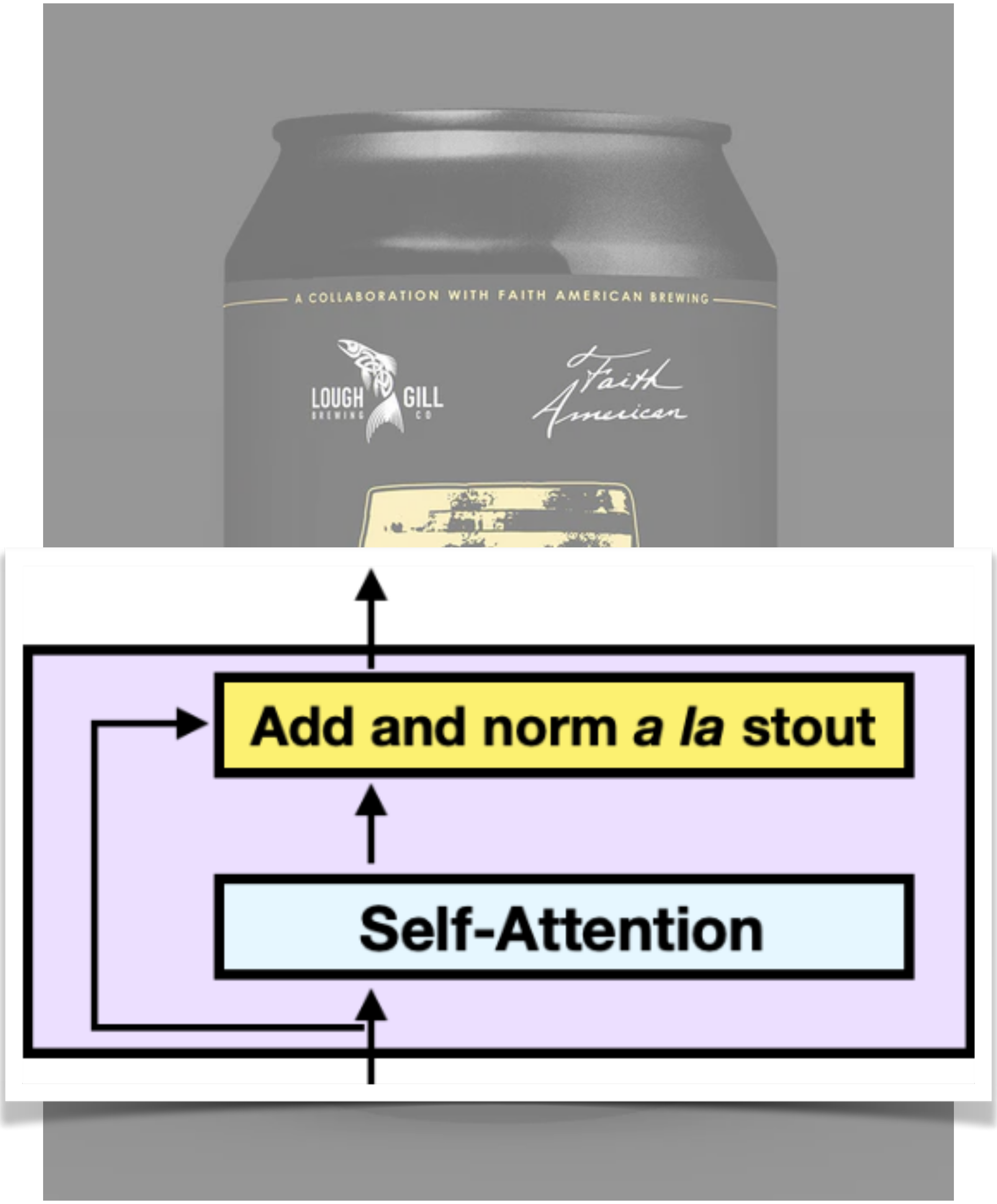
CASK?

A COLLABORATION WITH FAITH AMERICAN BREWING
LOUGH GILL
BREWING CO
Faith American
CASK
BARREL AGED IMPERIAL IRISH COFFEE STOUT
12.0%
ALC BY VOL
330 ML
AGED IN BARRELS FROM CLONAKILTY DISTILLERY

Cask stout
(Whisky Barrel-Aged Stout beer)
= stout beer in a cask

https://loughgillbrewery.com/products/cask-irish-coffee-stout

Gauge covariant transformer
= CASK


A COLLABORATION WITH FAITH AMERICAN BREWING
LOUGH GILL
BREWING CO
Faith American
Add and norm *a la* stout
Self-Attention

Cask stout
(Whisky Barrel-Aged Stout beer)
= stout beer in a cask

Covariant attention block
CASK = Covariant Attention
with Stout Kernel

It is named in an obvious reason😜

https://loughgillbrewery.com/products/cask-irish-coffee-stout

Gauge covariant transformer

Collection of ML/LQCD

Lattice	ML(Framework)	ML/Lattice
· Demon method (inverse MC) arXiv1508.04986 AT+ · Hopping parameter	Linear regression	Phys. Rev. D 107, 054501 AT+ Gauge inv. SLMC Trivializing with SD eq a la Luscher 2212.11387 AT+
Stout & Flow	CNN/Equivariant NN	Gauge covariant net 2021 AT+
(nothing. mean field?)	Transformer - GPT	· Global symmetric Transformer 2306.11527 AT+ · CASK (this talk)

https://loughgillbrewery.com/products/cask-irish-coffee-stout

Gauge covariant transformer

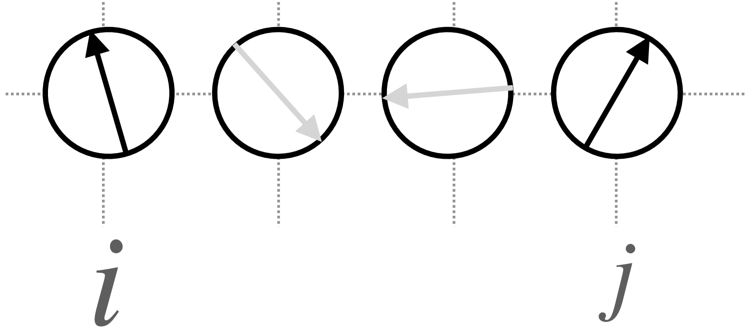
Idea: Attention must be covariant

Attention matrix in transformer ~ correlation function (with block-spin transformed spin)

-> we replace it with “correlation function for links” in a **covariant** way

$$a_{ij} \sim \vec{S}_i \cdot \vec{S}_j$$

(with activation)

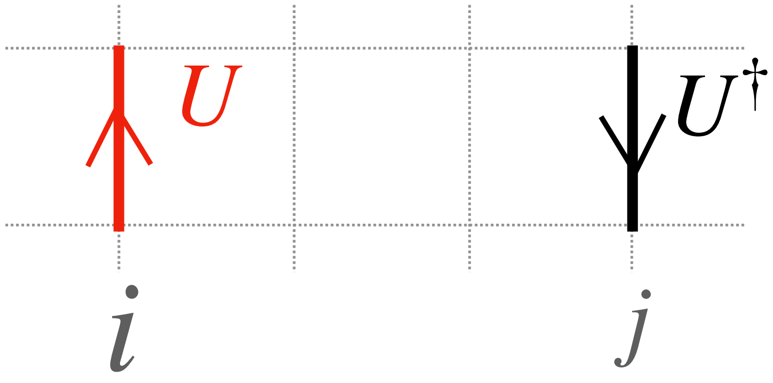


invariant
under global O(3)

$$a_{ij} \sim (R\vec{S})_i^\top R\vec{S}_j = \vec{S}_i^\top \vec{S}_j$$

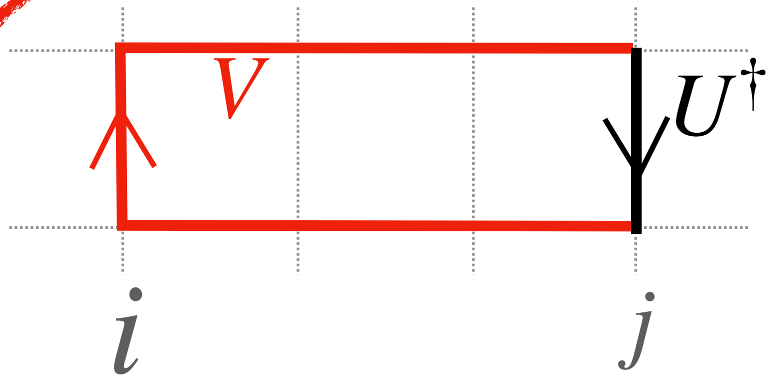
In total, output is covariant

✗ $a_{i\mu,j\mu} \sim \mathrm{Re\ tr}\ U_\mu(i) U_\mu^\dagger(j)$



not invariant
(cannot be used)

✓



invariant under
local SU(N)

$$a_{i\mu,j\mu} \sim \mathrm{Re\ tr}\ V_\mu(i) U_\mu^\dagger(j)$$

(with activation)

In total, output is covariant

Gauge covariant transformer

Structure of gauge symmetric attention using stout

[1] 2021 AT+

Procedure in three steps:

0. U^{in} : Input configuration/Links

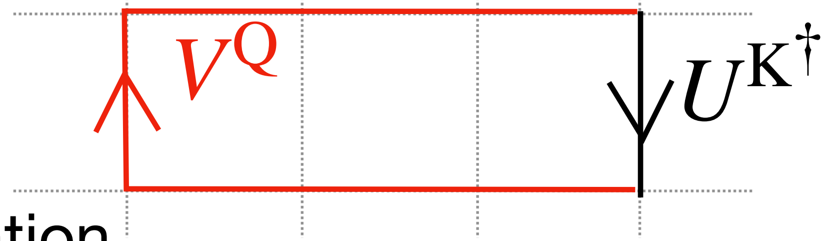
1. 3 types of (trainable) stout [1] -> $U^{\mathrm{Q}}, U^{\mathrm{K}}, U^{\mathrm{V}}$ (they have different weights)

$$U^{\alpha} = \exp[\rho^{\alpha} L[U^{\mathrm{in}}]]U^{\mathrm{in}} \qquad \alpha = \mathrm{Q}, \mathrm{K}, \mathrm{V}$$

weights

Loop operator
projected on Lie algebra

2. Construct attention matrix (Rectangular Wilson loop) using $U^{\mathrm{Q}}, U^{\mathrm{K}}$ -> $a_{(*,*)}$



$$\sim a_{(*,*)}$$ (with activation)

cf. sparse attention

3. Construct "stout smeared" [1] link with weight $a_{(*,*)}$ and U^{V}, U (as matrix mult)

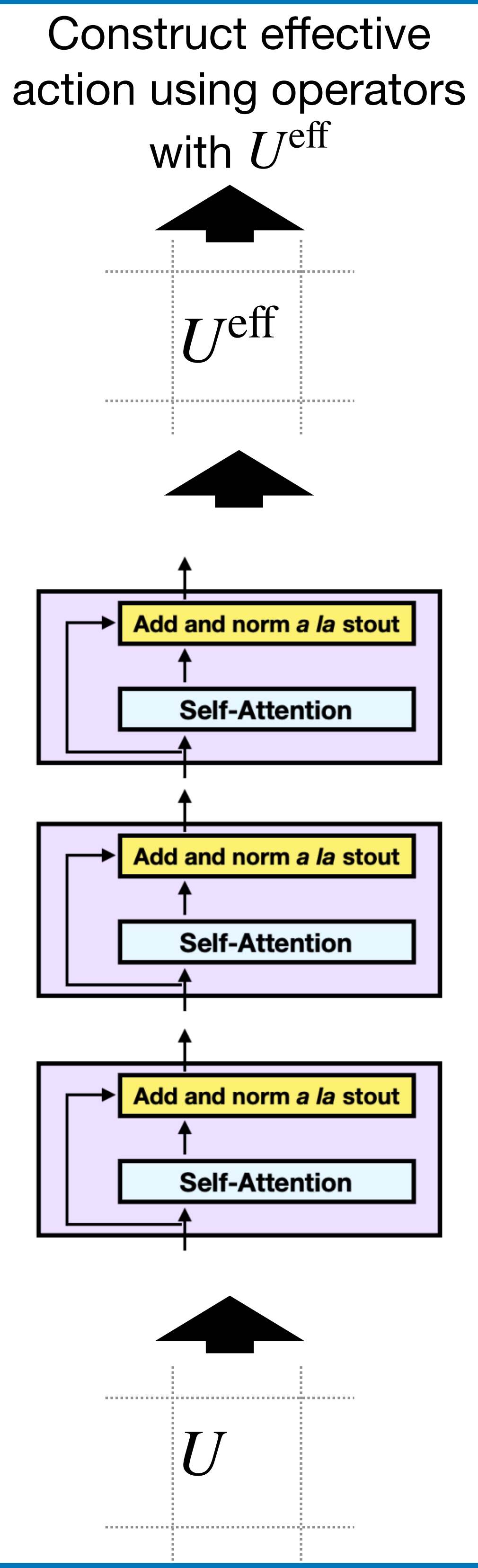
$$U^{\mathrm{out}} = \exp[a_{(*,*)} L[U^{\mathrm{V}}]]U^{\mathrm{in}}$$ Covariant

(This can be extend to have multi-head trivially)

Loop operator
projected on Lie algebra

Gauge covariant transformer

Simulation parameter



Construct effective action using operators with $U^{\rm eff}$

$U^{\rm eff}$

Add and norm *a la* stout

Self-Attention

Add and norm *a la* stout

Self-Attention

Add and norm *a la* stout

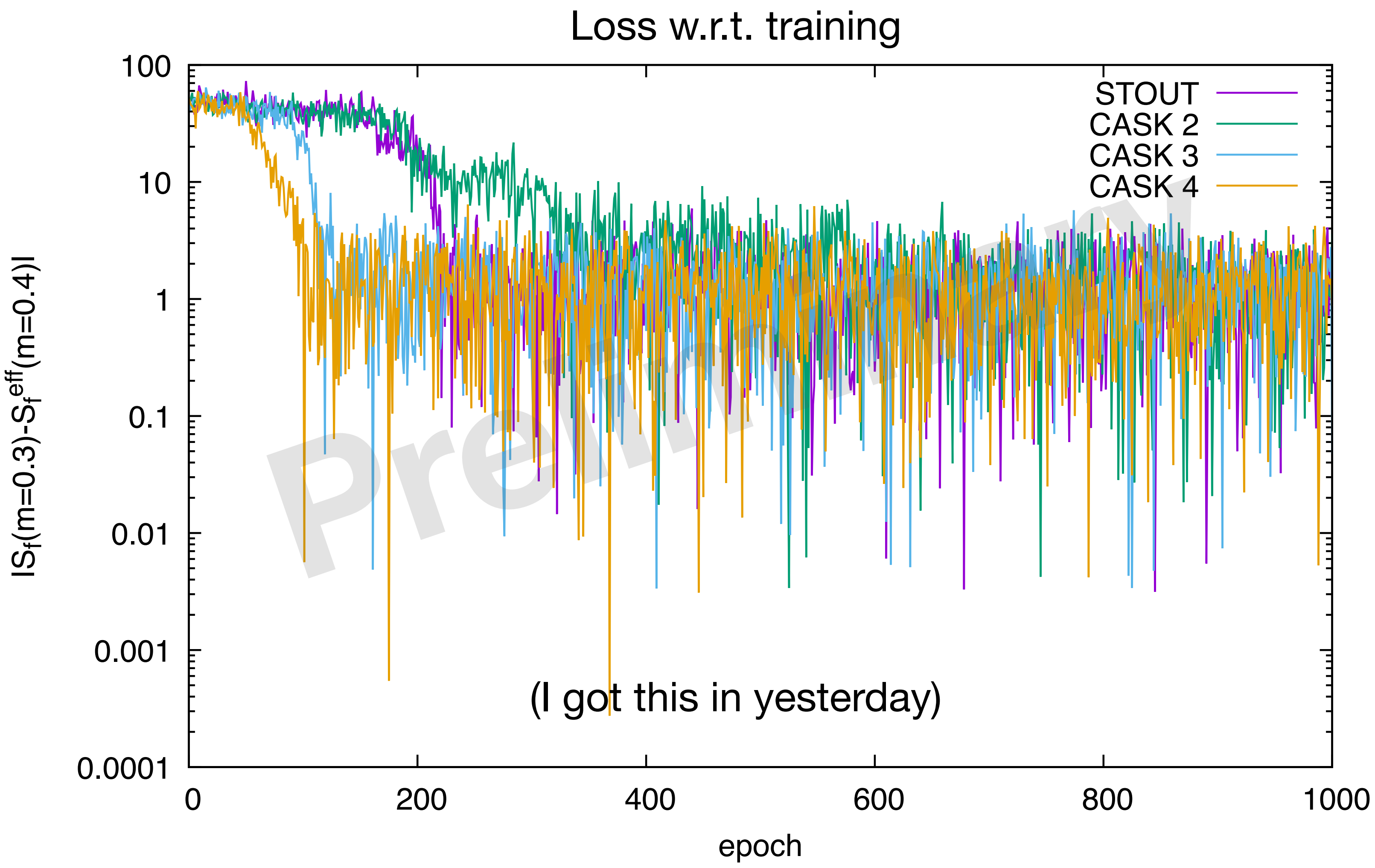
Self-Attention

U

- Self-learning HMC (1909.02255, 2021 AT+), an exact algorithm
  - Exact Metropolis test and MD with effective action
- Target S : $m = 0.3$, dynamical staggered fermion, Nf=2, $L^4 = 4^4$, SU(2), $\beta = 2.7$
- Effective action in MD ($S^{\rm eff}$)
  - Same gauge action
  - $m_{\rm eff} = 0.4$ dynamical staggered fermion, Nf=2
  - CASK with plaquette covariant kernel
    - Attention = 7-links rect staple (=3 plaq)
  - U links are replaced by $U^{\rm eff}$ in $D_{\rm stag}$
- “Adaptively reweighted HMC”

Gauge covariant transformer

Loss = difference of action



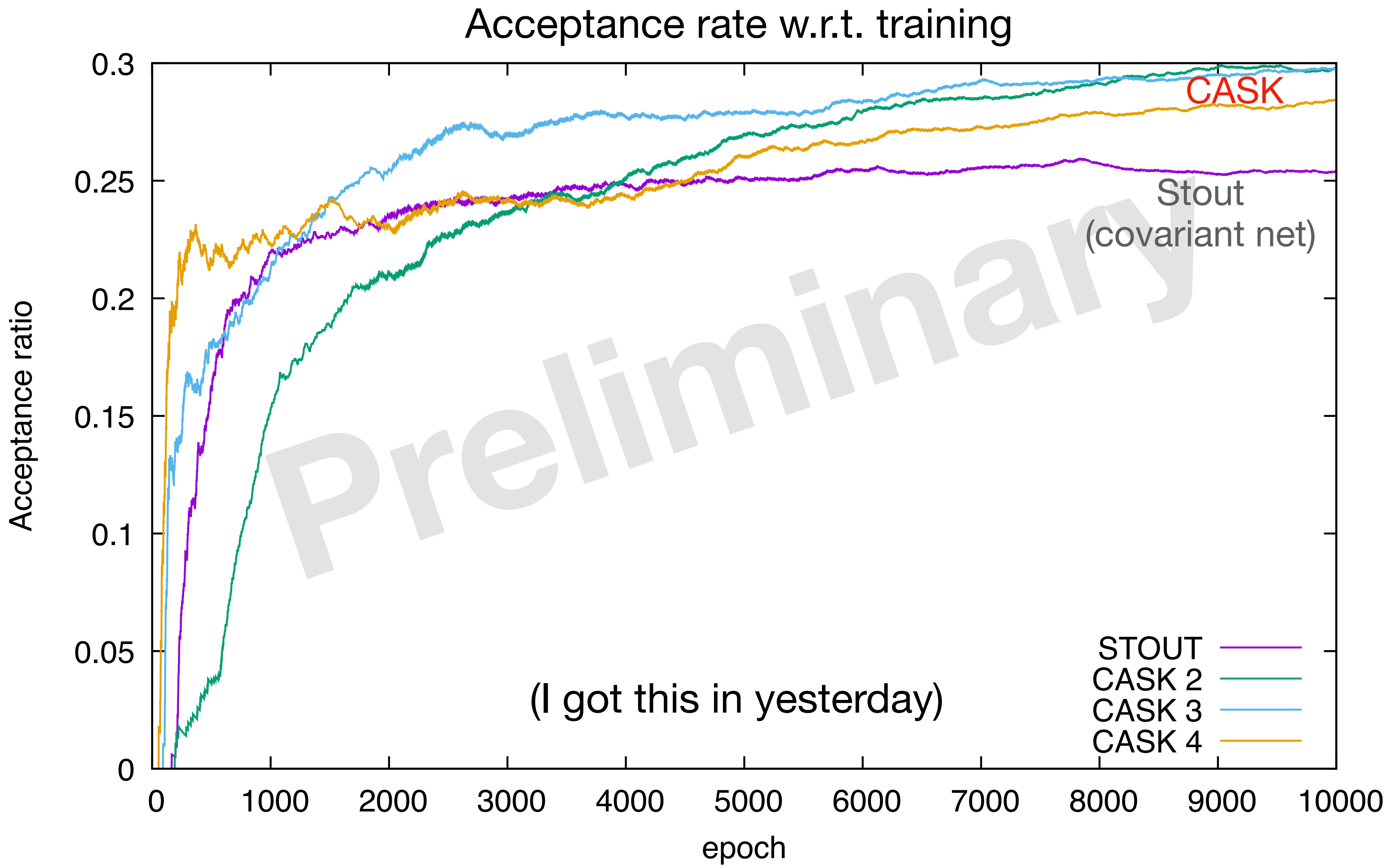
(I got this in yesterday)

= MC steps

- Loss decreases along with the training steps
- it works as same as the stout (covariant net)
- No gain?

Gauge covariant transformer

Some gain



= MC steps

- In terms of acceptance, CASK has some gain
- It is still improving

バイアス補正近似法

Work in progress

Reduction of cost in measurements

Costly observables

Measurements are needed! Some observables are numerically expensive.

Measurement: determination of quark propagator
For a given gauge configuration $[A_\mu]$, $1/(D[A]+m)$ can be calculated (*)

Machine learning can help?

Concern:
- Machine learning are approximation, can you remove bias?

(*) Precisely speaking, we need to fix the gauge.

Reduction of cost in measurements

Cost reduction via machine learning a la AMA

G S. Bali+ 0910.3970, Blum, Izubuchi, Shintani 2012
B. Yoon+ 1807.05971, 1909.10990
H. Wettig+ [1] , B. Choi+ WIP [2]

All mode averaging (AMA) technique can reduce statistical error using approximation.
Approximation can be biased but it can be corrected.

$$\langle O \rangle = \underline{\langle O^{(\mathrm{Appox})} \rangle} + \underline{\langle (O - O^{(\mathrm{Appox})}) \rangle}$$

evaluate

$$\frac{1}{N_{\mathrm{conf}}} \sum_{c=1}^{N_{\mathrm{conf}}} O^{(\mathrm{Appox})}[U_c]$$

Cheap
A lot of statistics

evaluate

$$\frac{1}{N_{\mathrm{bc}}} \sum_{c'=1}^{N_{\mathrm{bc}}} (O[U_{c'}] - O^{(\mathrm{Appox})}[U_{c'}])$$

Expensive
Small amount

AMA has been developed **without** machine learning,
but it can be used with machine learning

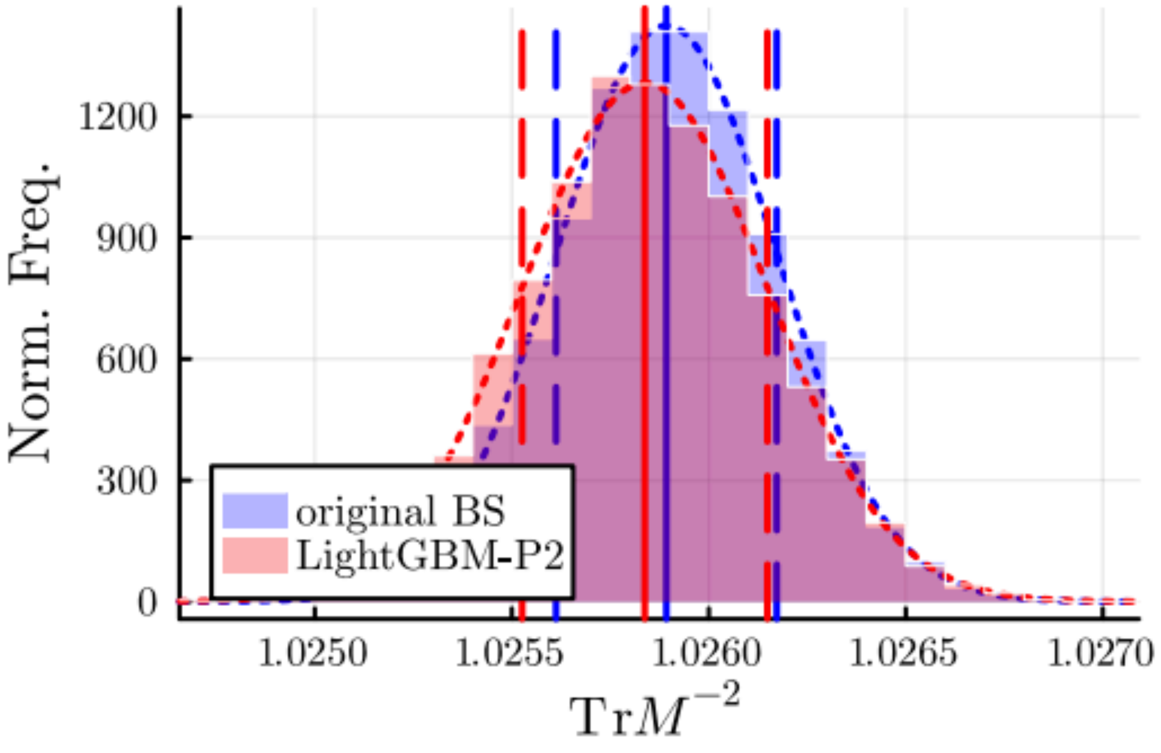
[1] https://conference.ippp.dur.ac.uk/event/1265/contributions/7450/attachments/5874/7758/lat24.pdf
[2] https://conference.ippp.dur.ac.uk/event/1265/contributions/7582/attachments/5706/7462/benji.pdf

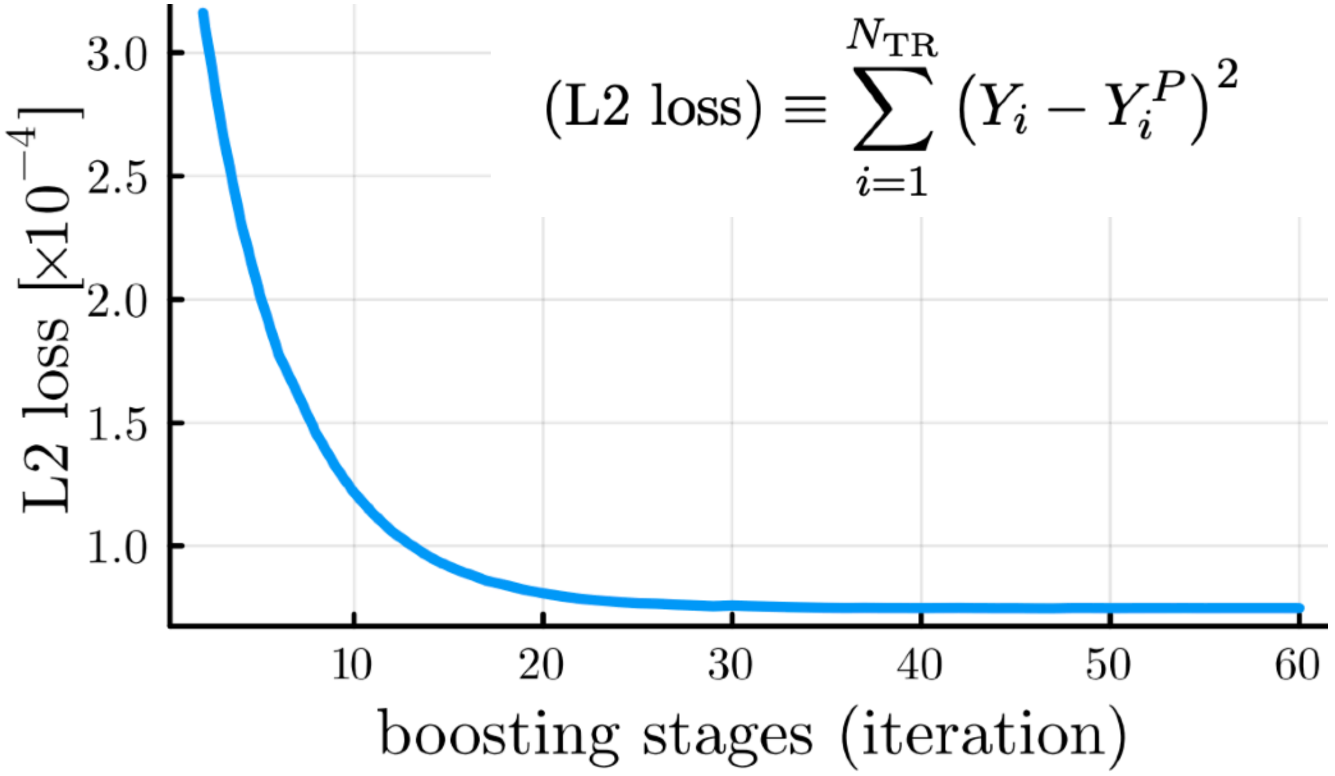
Reduction of cost in measurements

Reducing cost without bias from ML

B. Yoon+ 1807.05971, 1909.10990
H. Wettig+ [1] , B. Choi+ WIP [2]

Plaq to $\mathrm{tr}[D^{-2}]$





$$\langle O \rangle \simeq \frac{1}{N_{\text{conf}}} \sum_{c=1}^{N_{\text{conf}}} O^{(\text{Appox})}[U_c] + \frac{1}{N_{\text{bc}}} \sum_{c'=1}^{N_{\text{bc}}} (O[U_{c'}] - O^{(\text{Appox})}[U_{c'}])$$

So far so good (I skipped details),
and details can be found in [2]

[1] https://conference.ippp.dur.ac.uk/event/1265/contributions/7450/attachments/5874/7758/lat24.pdf
[2] https://conference.ippp.dur.ac.uk/event/1265/contributions/7582/attachments/5706/7462/benji.pdf

まとめ

格子QCD + 機械学習

- 格子QCDには実用上、問題もある
- 機械学習が有用 (かもしれない)
    - 僕が始めた2016年からは隔世の感がある
    - ゲージ場OK、厳密性OK
    - できないこともある、頑張りましょう。
    - 本当に速くなる? コードによる。目的による。
- 機械学習 + 格子QCDのコード (JuliaQCD)
    - 応用先は色々。頑張りましょう。

MLPhys Foundation of "Machine Learning Physics"
Grant-in-Aid for Transformative Research Areas (A)

Program for Promoting Researches on the Supercomputer Fugaku
Large-scale lattice QCD simulation and development of AI technology

KAKENHI: 20K14479, 22H05112, 22H05111, 22K03539

Thanks!