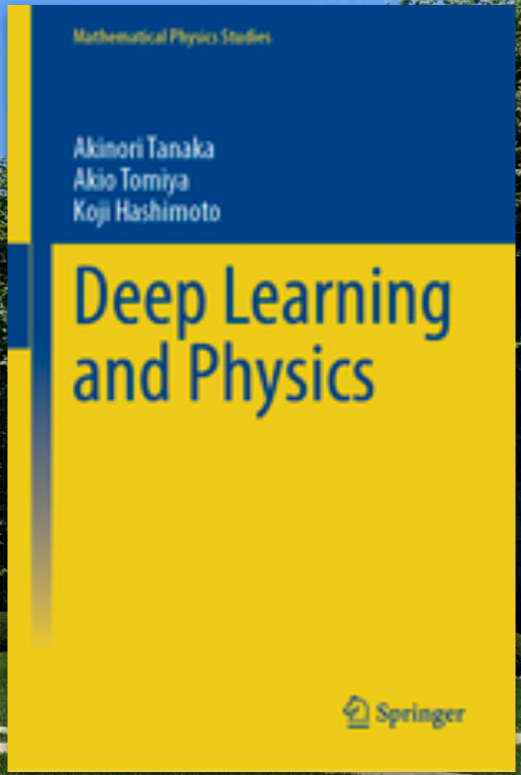
Confinement
2024/08/21 (Thu)
Karins, Australia

https://indico.cern.ch/event/1293041/contributions/6062384/

Machine Learning Methods in Lattice QCD

Akio Tomiya (Lecturer/Jr Associate prof)
Tokyo Woman's Christian University



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

富岳

akio_at_yukawa.kyoto-u.ac.jp

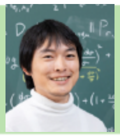
My team: LQCD + ML

"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  +...

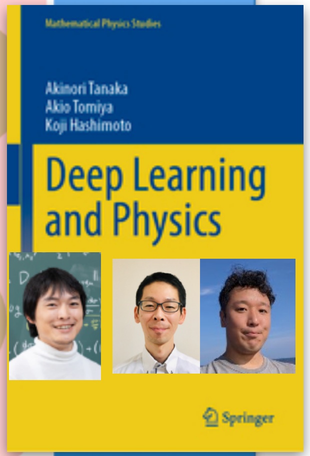
A01 A.Tomiya: Computational physics  +...

A02 M.Nojiri: High Energy Physics

A03 T.Ohtsuki: Condensed Matter Physics 

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

ML Phys



2021

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

科研費 KAKENHI

My team (A01): LQCD + ML

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

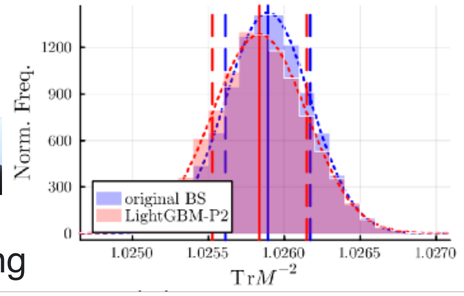
J. Takahashi
Meteorological College

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**

ML Phys A01

measurement with BDT



On going

LatticeQCD.jl

Open source

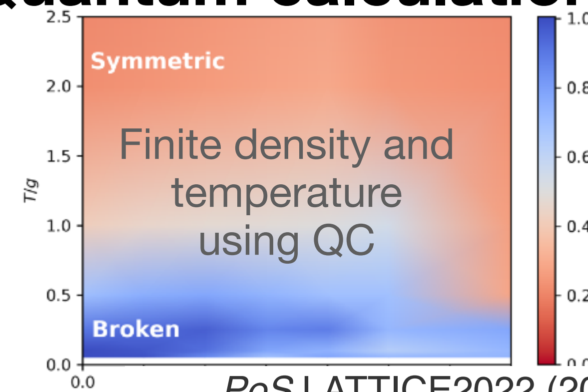
LQCD (+ML) with julia

This covers most of modern tech

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

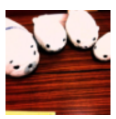
(and associated sub-libraries)

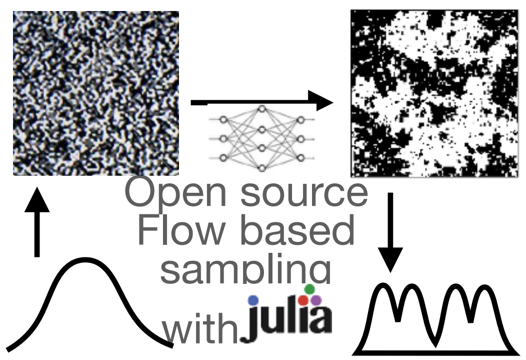
Quantum calculation



ML + QC: Quantum thermodynamics using Density matrix and MADE

PoS LATTICE2022 (2023) 039



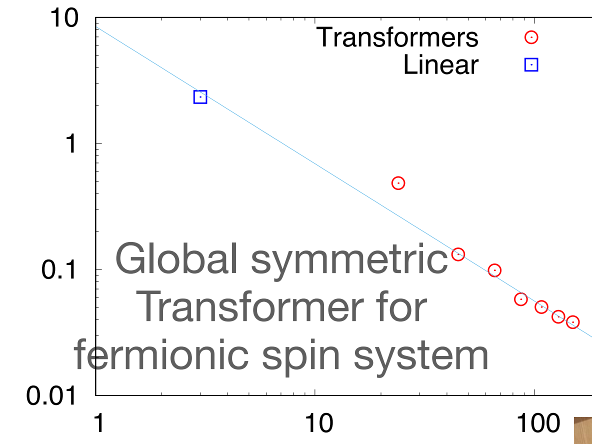


arXiv: 2208.08903

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

Gauge covariant neural net

arXiv: 2103.11965

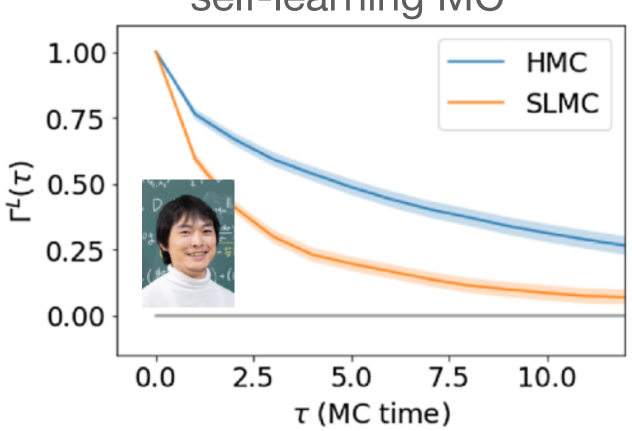


2310.13222

2306.11527

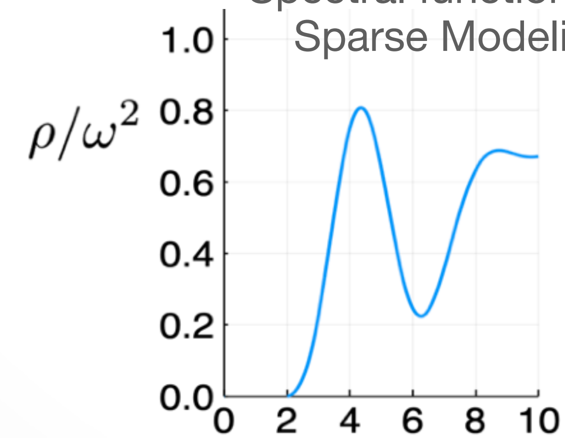
Gauge configuration

Gauge invariant self-learning MC



Phys. Rev. D 107, 054501

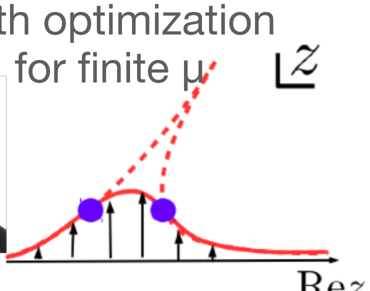
Spectral function with Sparse Modeling



(arXiv: 2311.15233)

+ on going

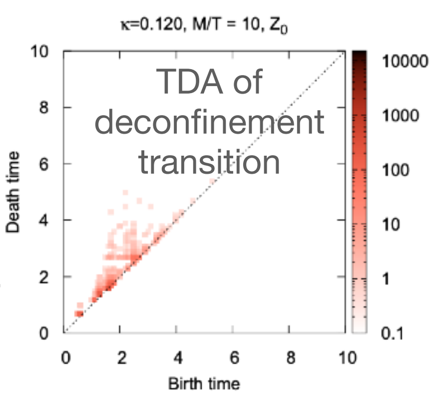
Path optimization for finite μ



Phys. Rev. D 108, 094504

(Figure from 1812.11506)

Sign problem

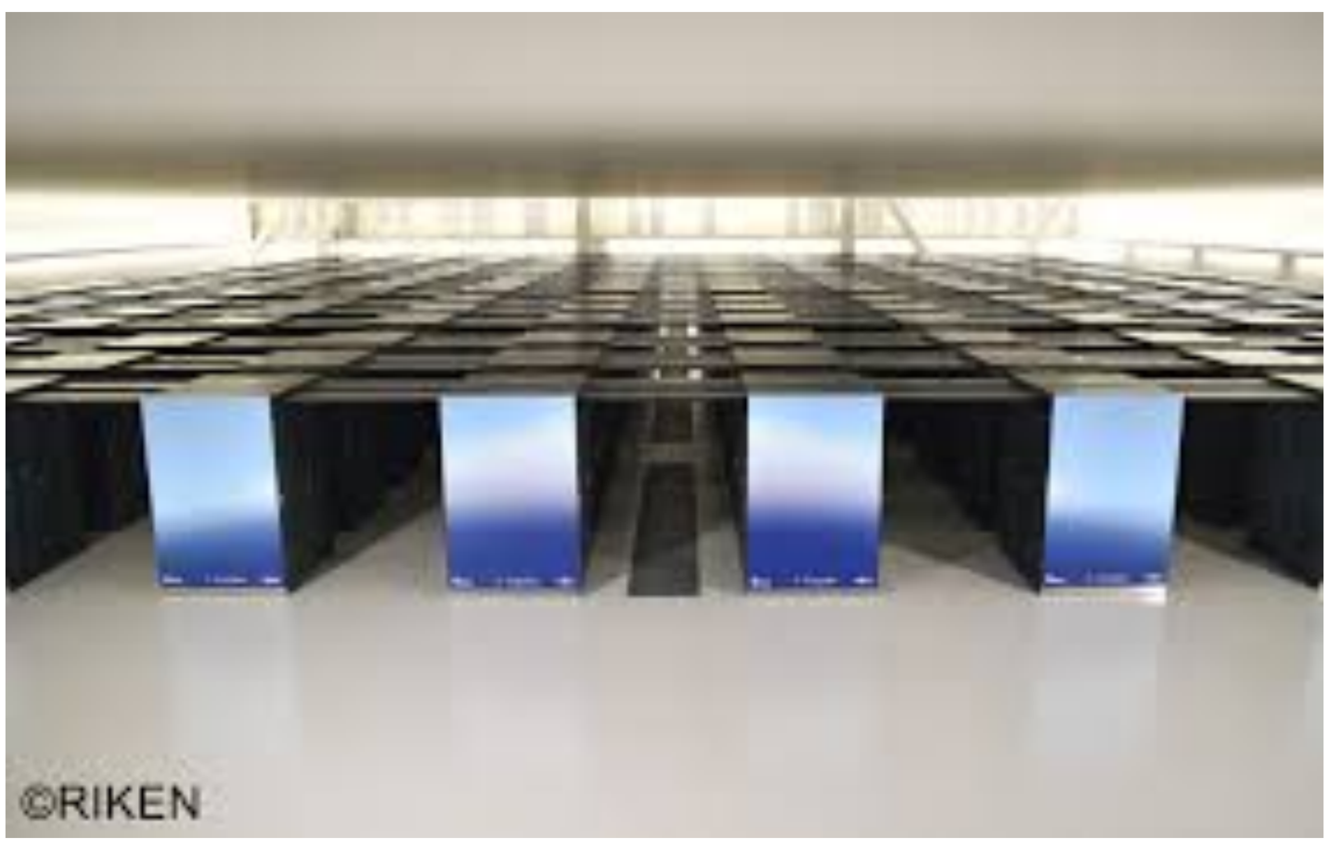


1810.07635

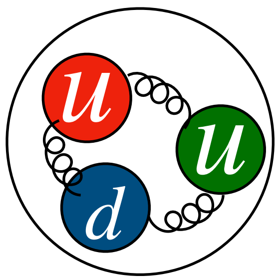
Other projects are going (with me)

“Program for Promoting Researches on the Supercomputer Fugaku”

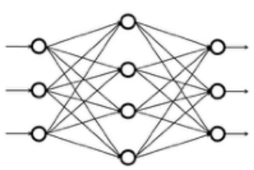
- Simulation for basic science: approaching the new quantum era
  - PI: Shoji Hashimoto
- Search for physics beyond the standard model using large-scale lattice QCD simulation and development of AI technology toward next-generation lattice QCD
  - PI: Takeshi Yamazaki



Outline of my talk



Lattice QCD?



Machine learning?

Machine learning
for Lattice QCD

1. Configuration generation in LQCD
    1. Self-learning MC with gauge covariant net
    2. CASK: Gauge symmetric transformer
    3. Flow based sampling
2. Reduction of cost in measurements
    4. Bias corrected approximation
    5. Control variates

- I am writing a review paper for machine learning applications in lattice QCD
  It should be appeared in JPSJ, Journal of the Physical Society of Japan, and arXiv(?) soon (I hope!)

What is Lattice QCD?

Introduction

Akio Tomiya

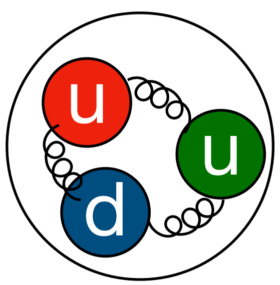
Lattice QCD = QCD on discretized spacetime = calculable

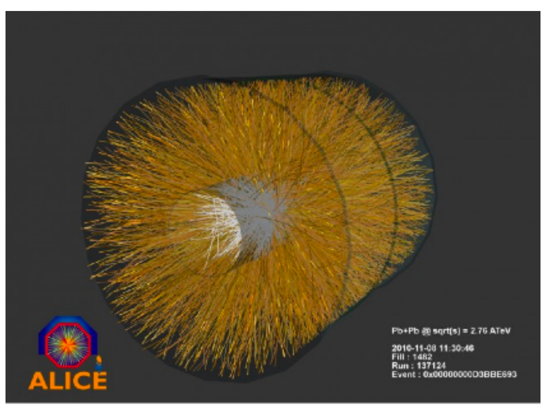
QCD (Quantum Chromo-dynamics) in 3 + 1 dimension

$$S = \int d^4x \left[-\frac{1}{2} \mathrm{tr}\, F_{\mu\nu}F^{\mu\nu} + \bar{\psi}\left(i\partial\!\!\!/ + gA\!\!\!/ - m\right)\psi \right]$$

$$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu - ig[A_\mu, A_\nu]$$

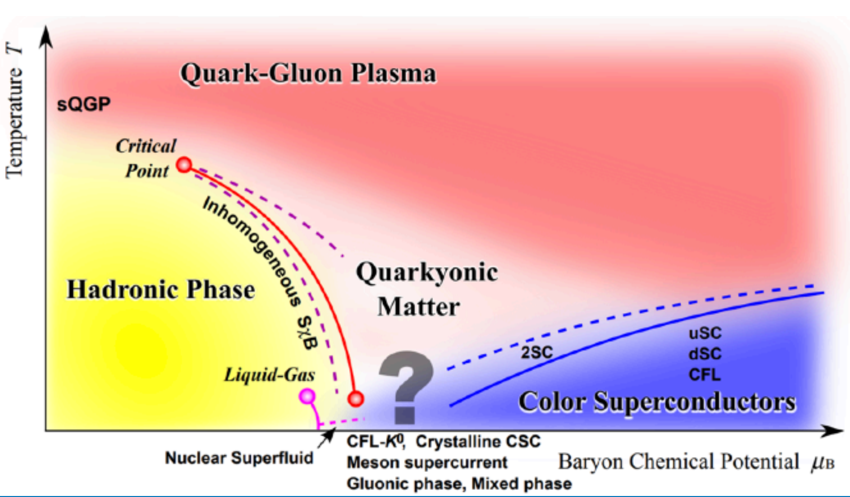
Non-commutable version of (quantum) electro-magnetism





- This describes inside of nuclei& mass of hadrons, equations of states etc
- **We want to evaluate expectation values with following integral,**

$$\langle O \rangle \sim \int \mathscr{D}A \mathscr{D}\bar{\psi} \mathscr{D}\psi e^{iS} O$$



- It is difficult. Let's use lattice!

Intro: Lattice QCD& Monte-Carlo

Numerical integral (via trapezoidal type) is impossible

$$S = \int d^4x \left[+ \frac{1}{2} \mathrm{tr}\, F_{\mu\nu} F_{\mu\nu} + \bar{\psi} \left(\partial\!\!\!/ - \mathrm{i} g A\!\!\!/ + m \right) \psi \right]$$

Lattice regularization

$$S[U, \psi, \bar{\psi}] = a^4 \sum_n \left[- \frac{1}{g^2} \mathrm{Re}\ \mathrm{tr}\ U_{\mu\nu} + \bar{\psi} \left(D\!\!\!/ + m \right) \psi \right]$$

They are ``same” up to irreverent operators
(= higher dimensional operators)

$$\mathrm{Re}\ U_{\mu\nu} \sim \frac{-1}{2} g^2 a^4 F_{\mu\nu}^2 + O(a^6)$$

= **cutoff,** a^{-1}

- Introduce cutoff a^{-1} =lattice regularization a,
- Results has discritized error $O(a)$ or $O(a^2)$
(These can be removed by extrapolation)
- We can use Monte Carlo for LQCD, as same as Ising model!

Intro: Lattice QCD& Monte-Carlo

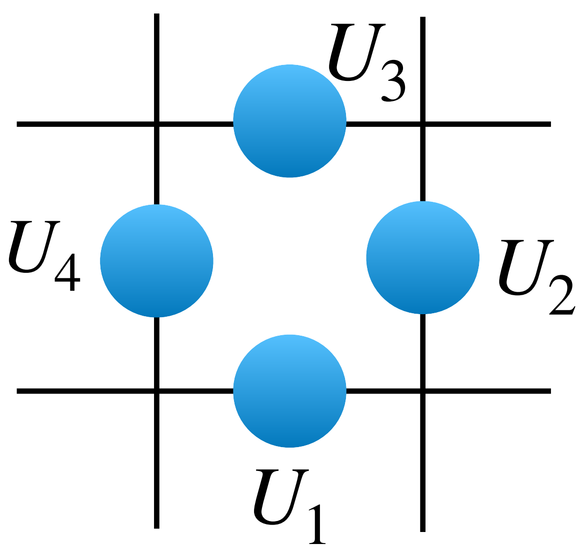
Akio Tomiya

LQCD = Non-perturbative calculation of QCD

QCD in Euclidean 4 dimension

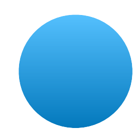
$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}U \mathcal{D}\bar{\psi} \mathcal{D}\psi e^{-S} \mathcal{O}(U) \quad = \frac{1}{Z} \int \mathcal{D}U e^{-S_{\text{gauge}}[U]} \det(D+m) \mathcal{O}(U)$$

LatticeQCD



SU(3) (3x3 unitary) on bonds

Importance sampling!

Sample  randomly according to e^{-S}

As same as Ising model

Lattice Gauge action for Imaginary time

$$S_{\text{gauge}} = \frac{6}{g^2} \sum_{vol} \left(1 - \frac{1}{3} \text{Re Tr}[U_1 U_2 U_3^\dagger U_4^\dagger] \right) \to \int \mathrm{d}^4 x \left[\text{tr } F_{\mu\nu} F_{\mu\nu} + O(a^2) \right]$$

Lattice QCD code for generic purpose

Open source LQCD code in Julia Language



Akio Tomiya
AT & Y. Nagai

LatticeQCD.jl

Q. julia Lang? A. Fast as fortran, easy as Python

Open source LQCD code (Julia Official package, v1.0)

Machines: Laptop/desktop/Jupyter/Supercomputers (almost everywhere)

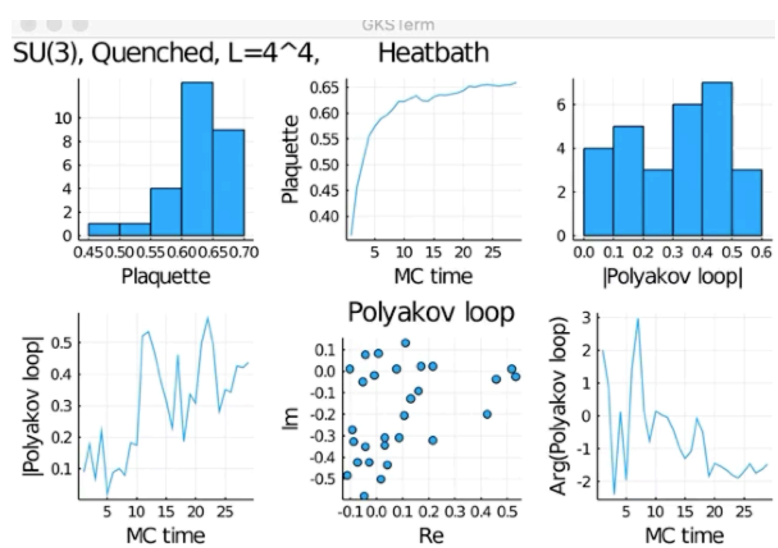
Advantage: Portability, no-explicit compile, fast, machine learning friendly

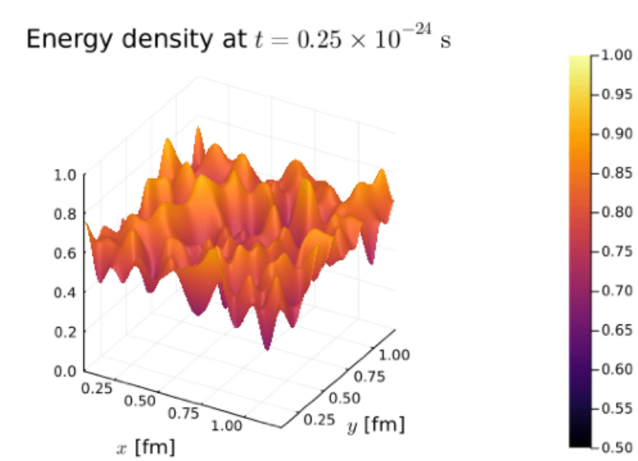
Functions: 4d, SU(Nc)-heatbath, (R)HMC, Self-learning HMC, SU(Nc) Stout, Z2 gauge, Dynamical Staggered, Dynamical Wilson, Dynamical Domain-wall Measurements (chiral condensate, topological charge, etc)

Start LQCD in **5 min**

1. Download Julia binary
2. Add the package through Julia package manager
3. Execute!

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





Introduction

Procedure of Lattice QCD, 3 steps

1.Production

- Fix lattice parameters, lattice size and cutoff a (=fix a coupling β), quark mass
- Sample gauge configurations from $e^{-S[U]}/Z$ (on **Supercomputers)**
- **large dimensional linear equations** have to be solved, many times

2.Measurement

- Calculate obsevables on each configuration
  - e.g. Quark condensate. $\bar{\psi}\psi \sim \mathrm{tr}\,(D+m)^{-1}$
- Some observable requires huge numerical resources
- Calculate on **Supercomputers**/workstation

3.Analysis, continuum limit

- Take average of observables, put statistical error bar (other errors as well)
- Extrapolate $a \to 0$
- (This is done on a laptop)

Introduction

Procedure of Lattice QCD, 3 steps

1.Production

- Fix lattice parameters, lattice size and cutoff a (=fix a coupling β), quark mass
- Sample gauge configurations from $e^{-S[U]}/Z$ (on **Supercomputers)**
- **large dimensional linear equations** have to be solved, many times

ML can be useful

2.Measurement

- Calculate obsevables on each configuration
  - e.g. Quark condensate. $\bar{\psi}\psi \sim \mathrm{tr}\,(D+m)^{-1}$
- Some observable requires huge numerical resources
- Calculate on **Supercomputers**/workstation

3.A

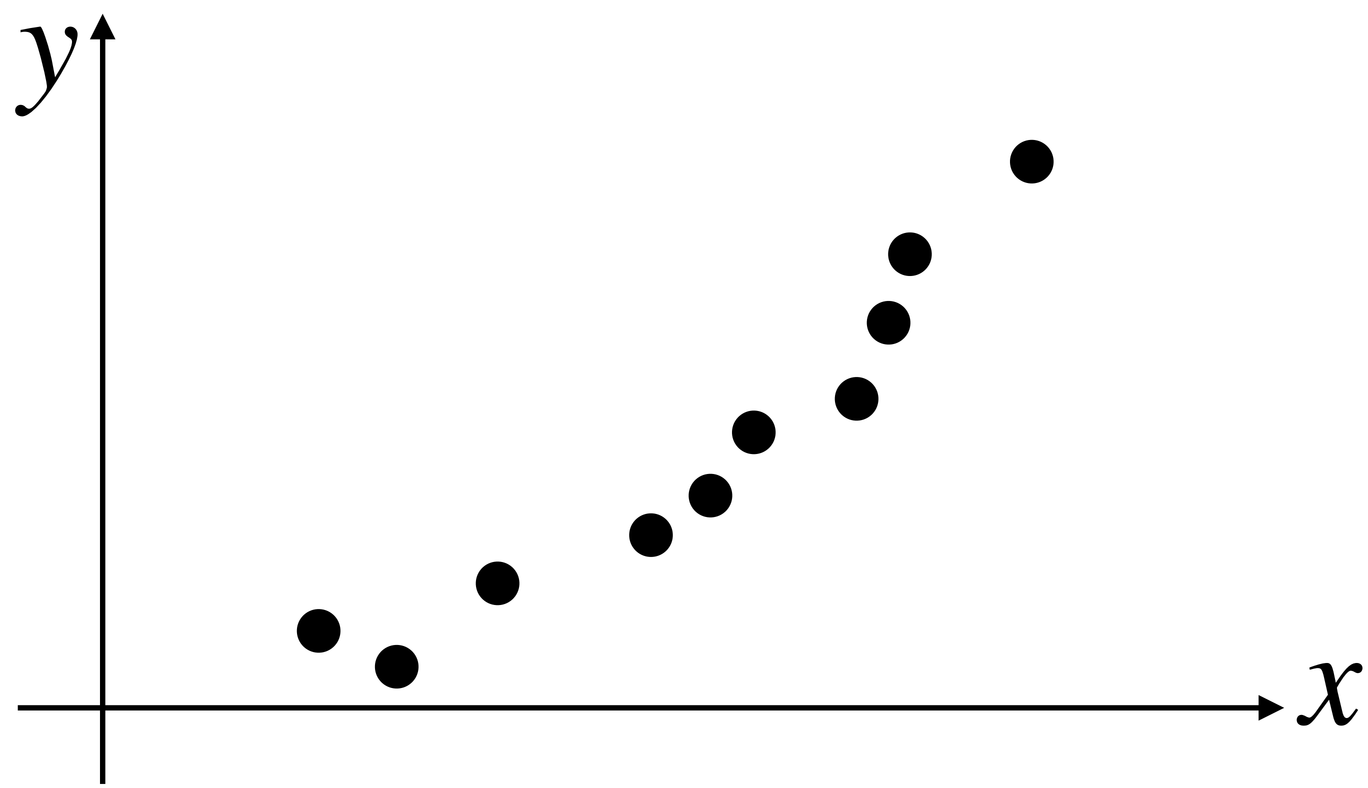
- Huge numerical cost
  - Production (solver, sampling topological sectors)
  - Measurements (sover)
- Machine learning can help both of them (probably)

Machine learning?

What is machine learning?

E.g. Linear regression ∈ Supervised learning

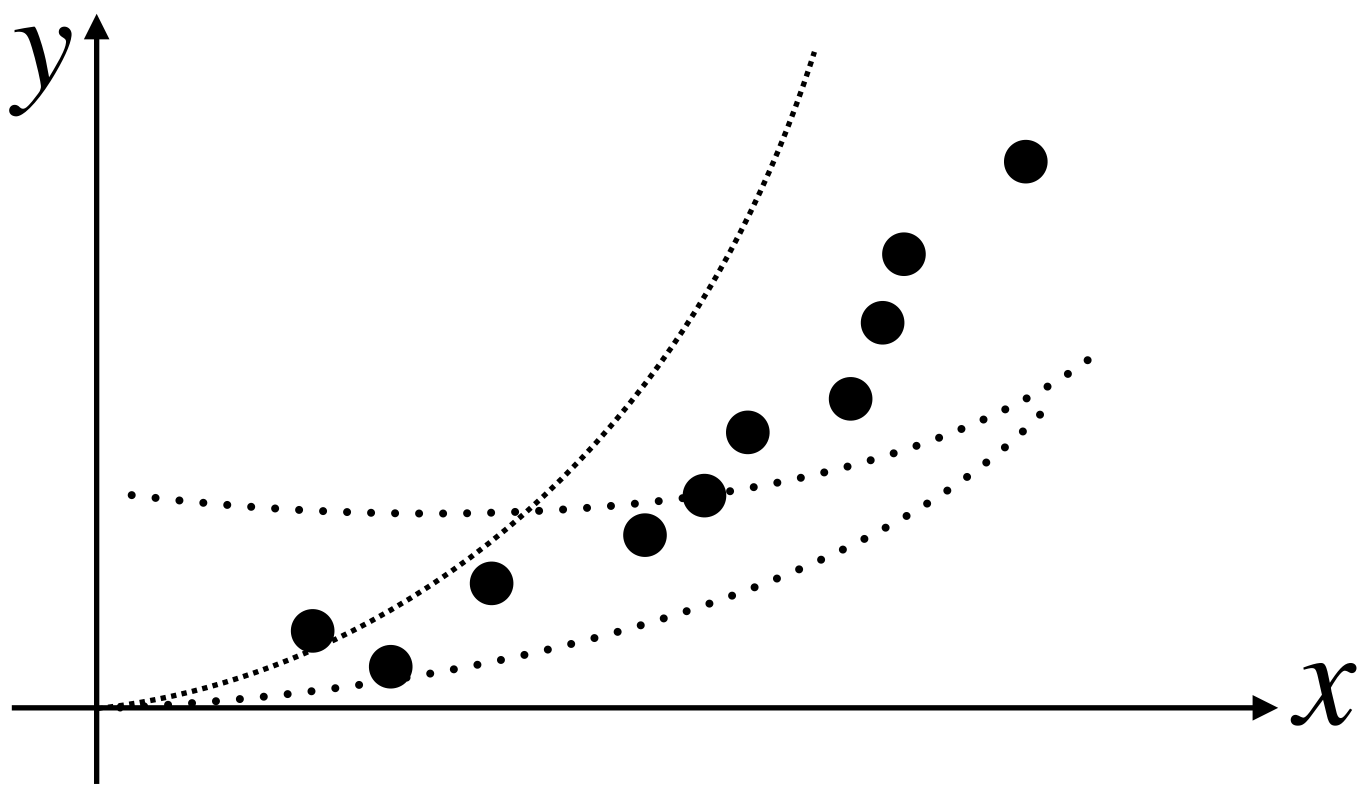
Data: $D = \left\{ (x^{(1)}, y^{(1)}),\ (x^{(2)}, y^{(2)}),\ \cdots \right\}$



What is machine learning?

E.g. Linear regression ∈ Supervised learning

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



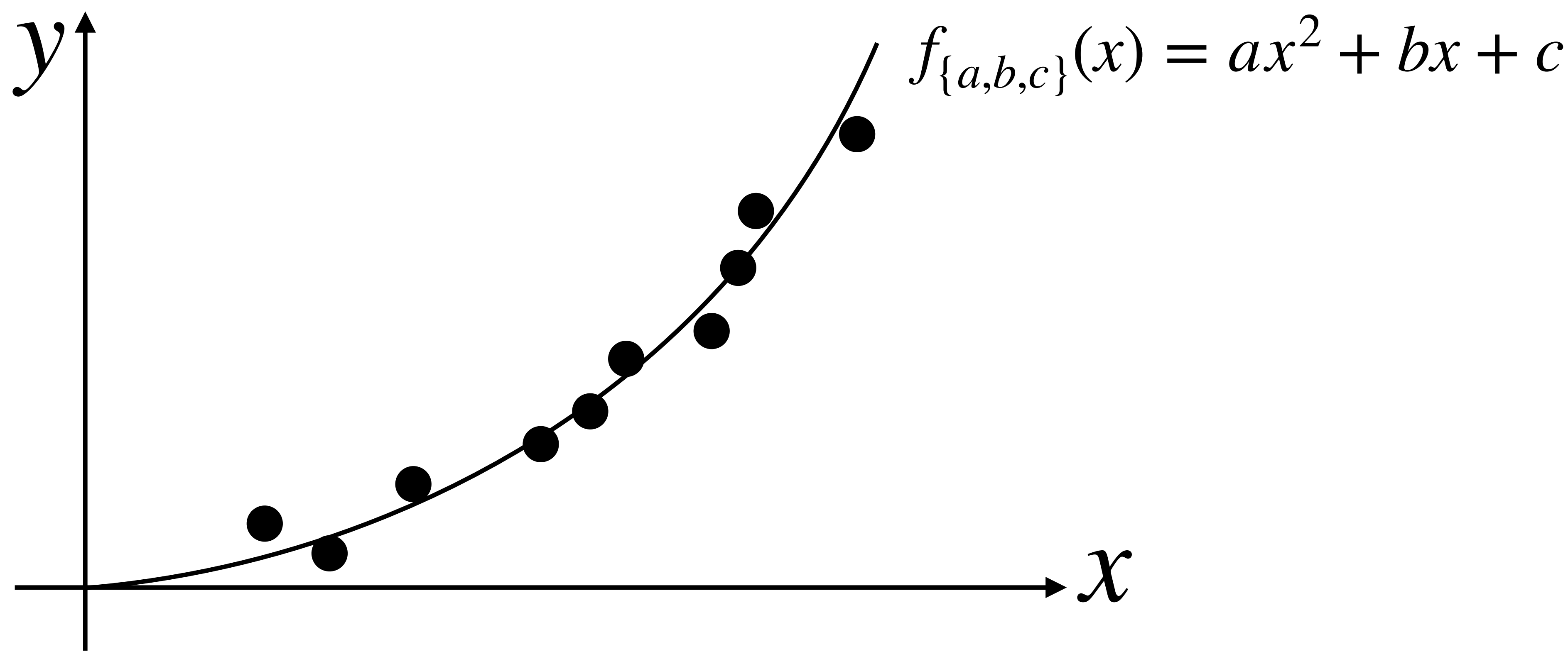
$$f_{\{a,b,c\}}(x) = ax^2 + bx + c \qquad E = \frac{1}{2}\sum_d \left| f_{\{a,b,c\}}(x^{(d)}) - y^{(d)} \right|^2$$

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

What is machine learning?

E.g. Linear regression ∈ Supervised learning

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



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

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

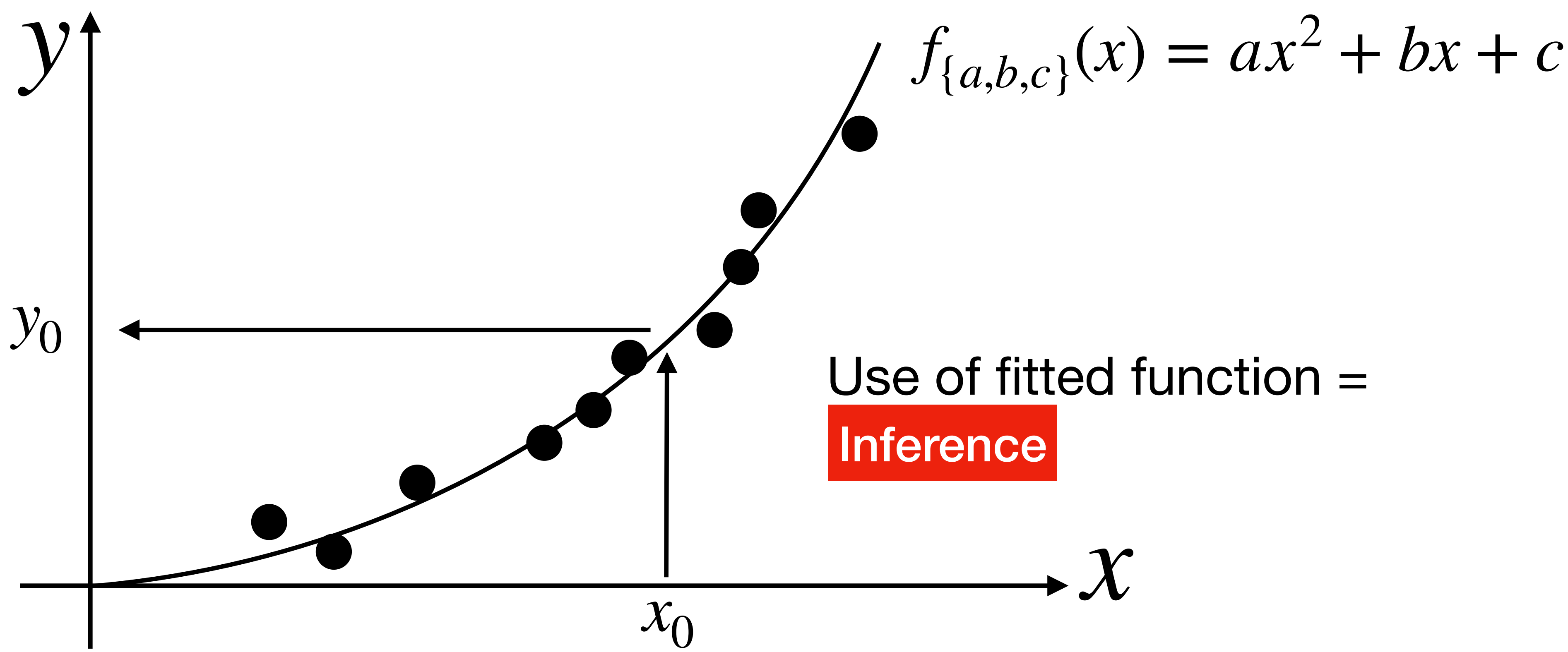
a, b, c, are determined by minimizing E

(training = fitting by data)

What is machine learning?

E.g. Linear regression ∈ Supervised learning

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



Use of fitted function = **Inference**

Now we can predict y value which not in the data

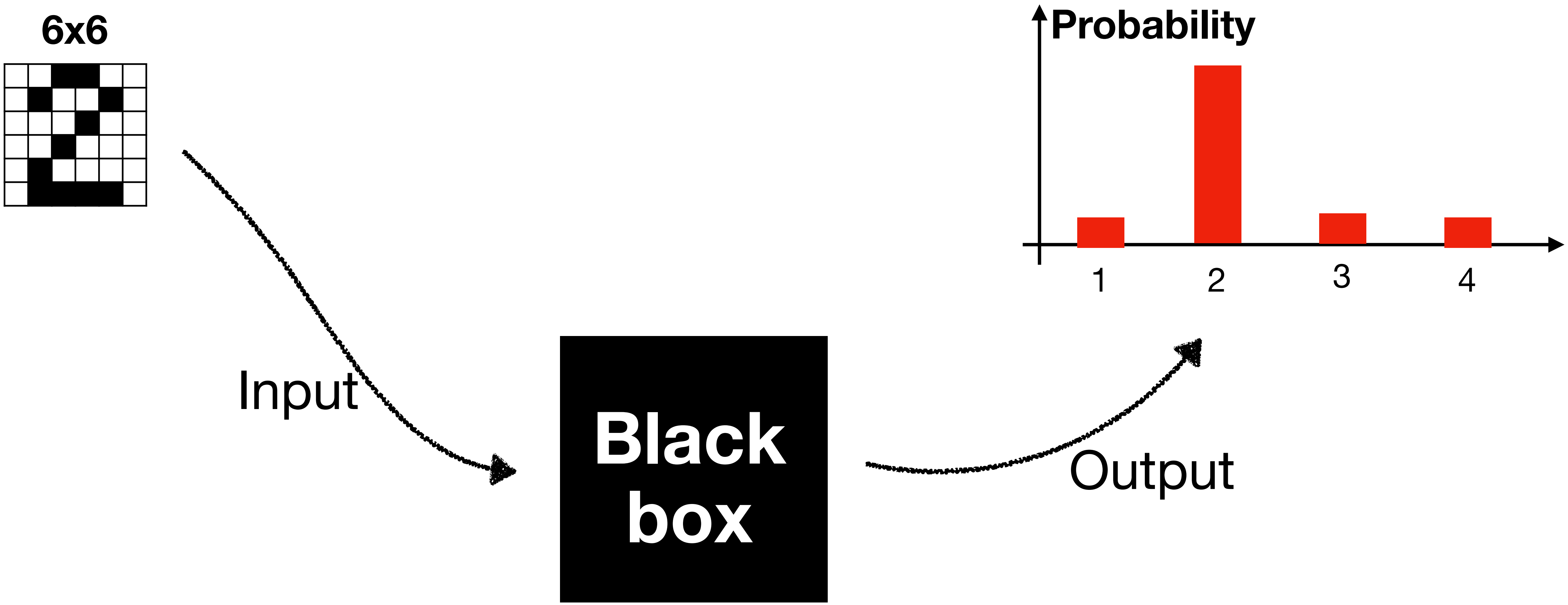
In physics language, variational method

What is the neural networks?

Akio Tomiya

Neural network is a *universal* approximation function

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



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)

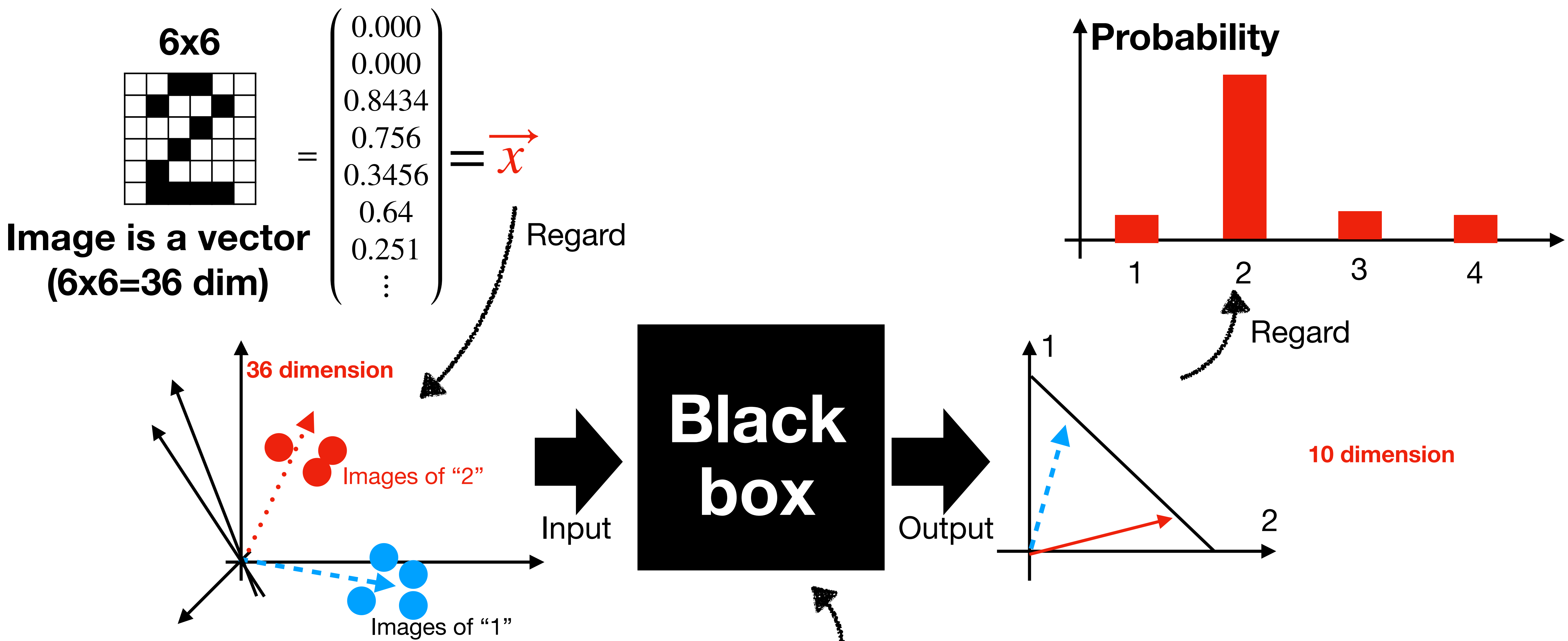


Image recognition = Find a map between two vector spaces

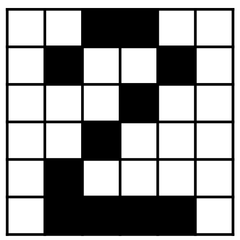
What is the neural networks?

Akio Tomiya

Neural network is a *universal* approximation function

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

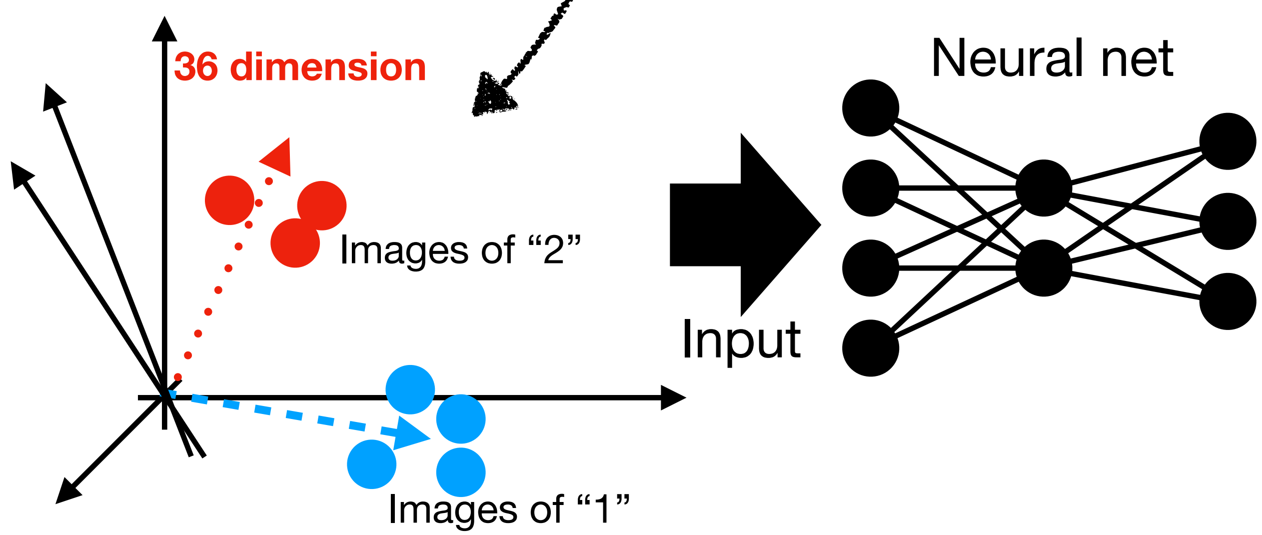
6x6



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

Image is a vector (6x6=36 dim)

Regard



What is the neural networks?

Akio Tomiya

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)

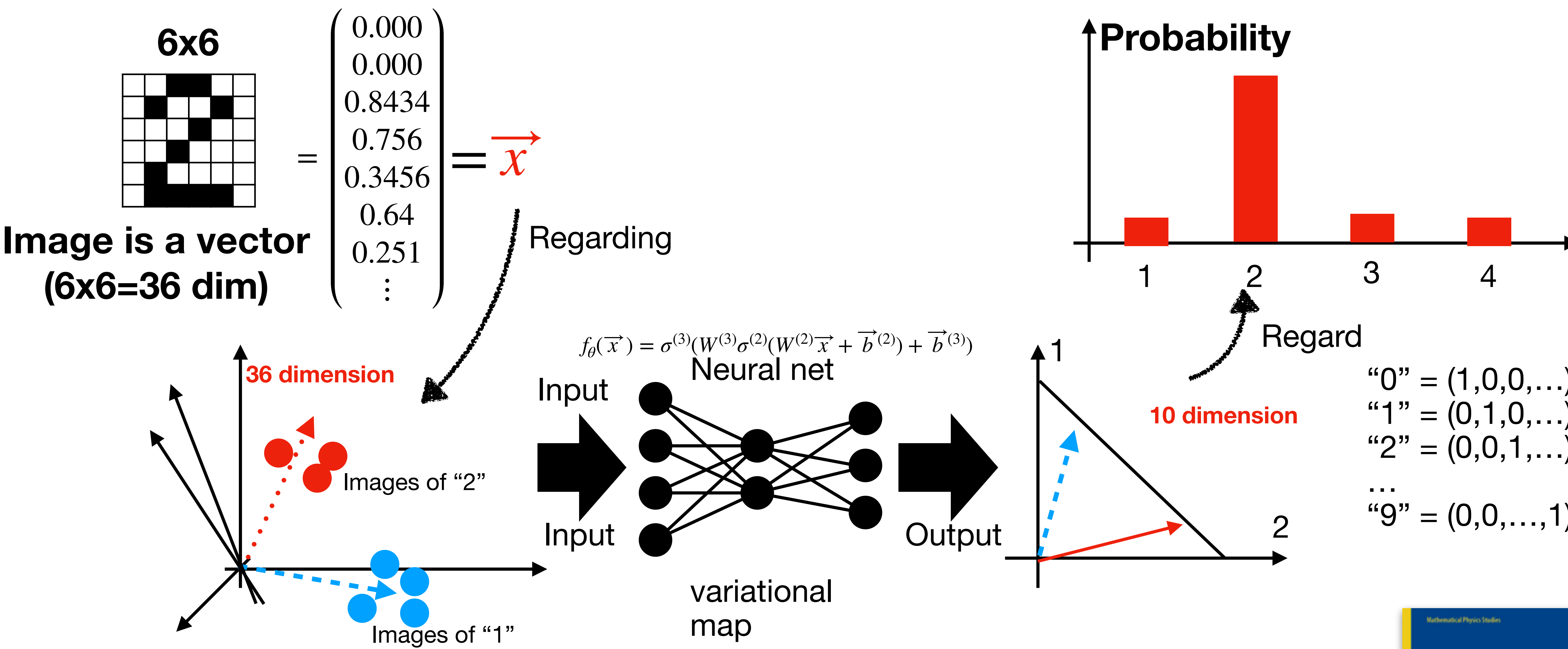
Neural network = map between vectors and vectors

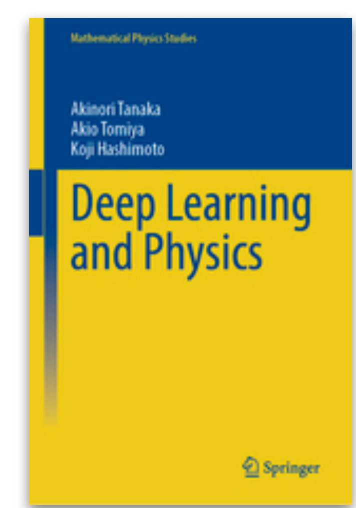
Physicists terminology: Variational ansatz

What is the neural networks?

Neural network is a *universal* approximation function

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





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

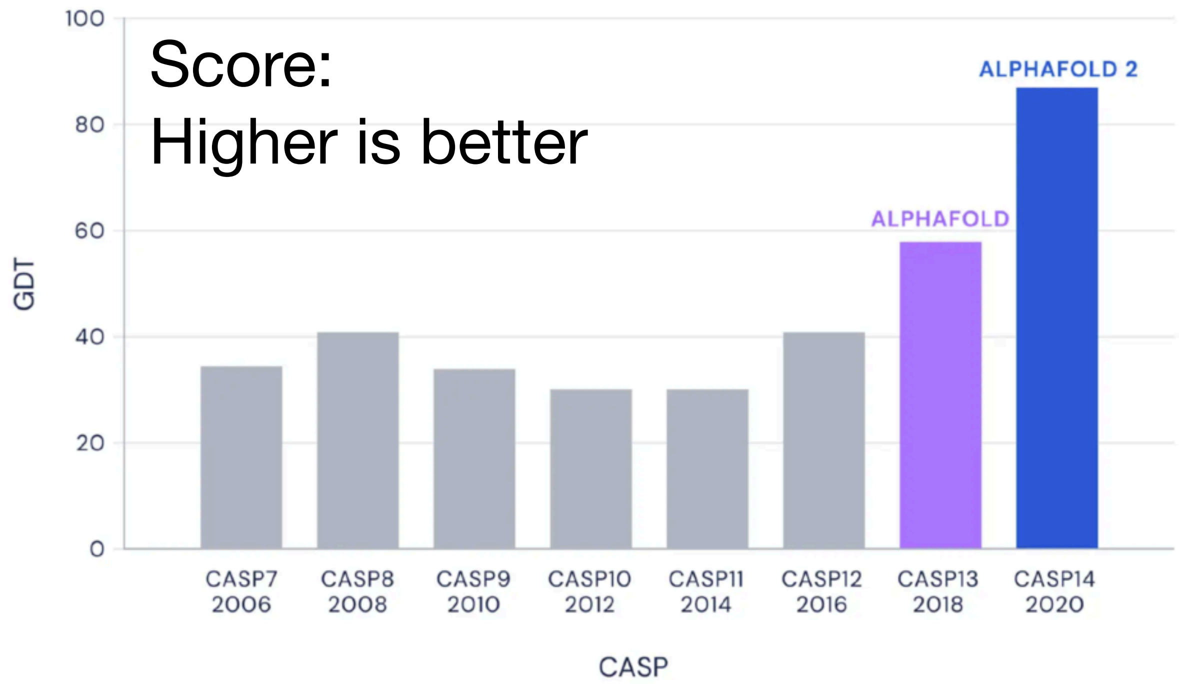
In this example, NN mimics image (36-dim vector) and label (10-dim vector)

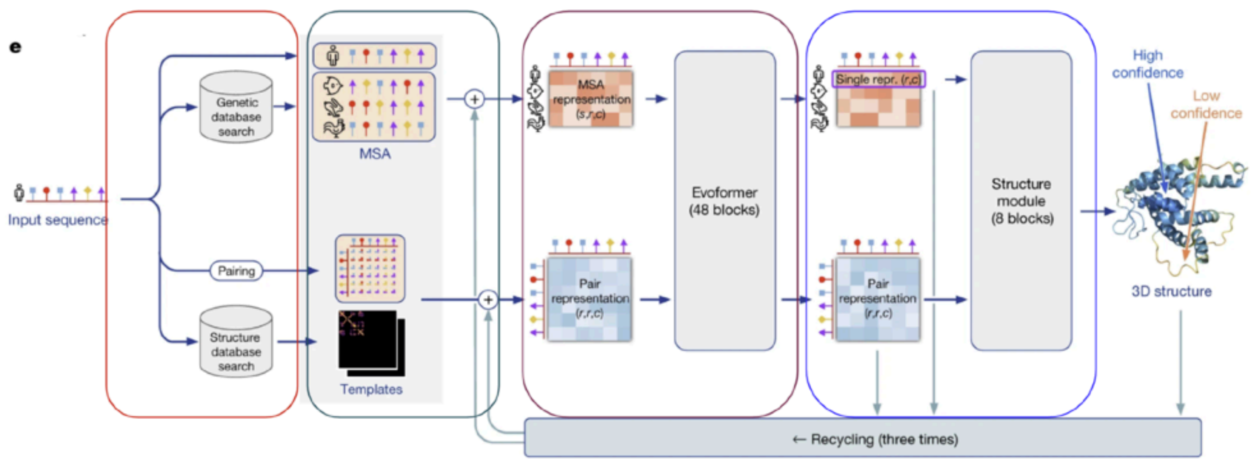
What is the neural networks?

Akio Tomiya

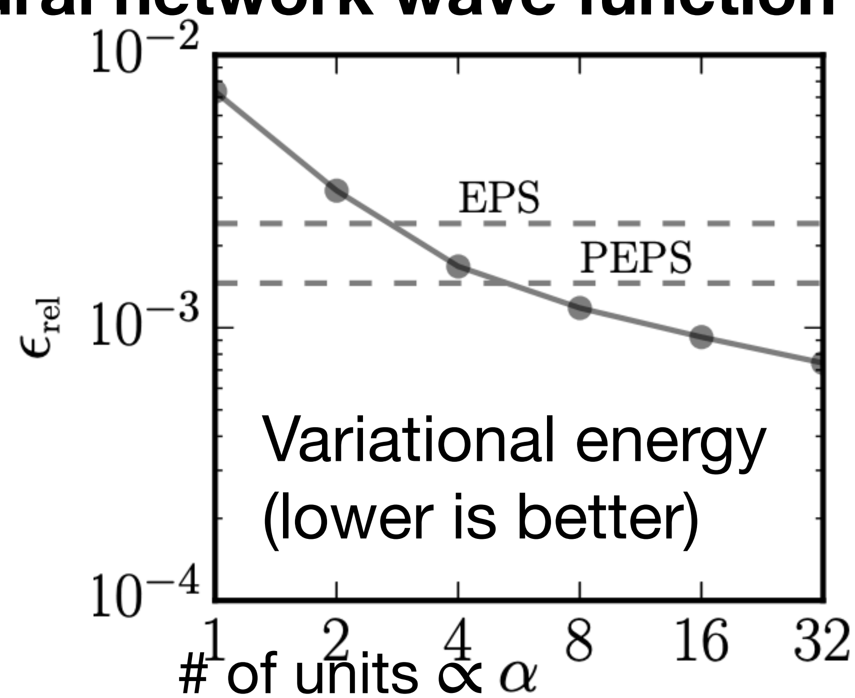
Neural network have been good job

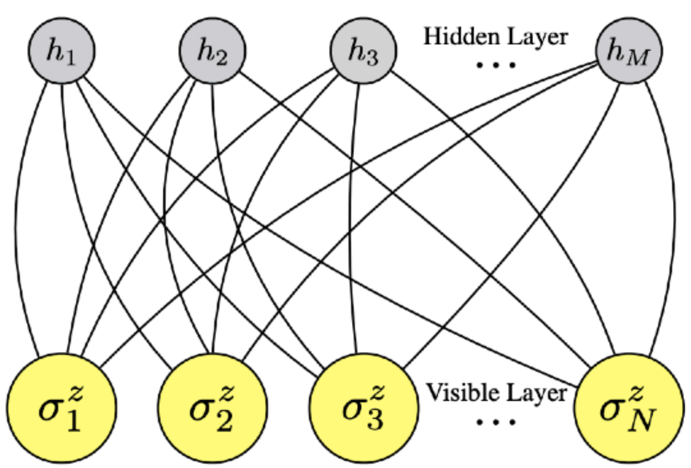
Protein Folding (AlphaFold2, John Jumper+, Nature, 2020+), Transformer neural net





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





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

Applications on LQCD

Machine learning for lattice QCD

1. Configuration generation in LQCD
    1. Self-learning MC with gauge covariant net
    2. CASK: Gauge symmetric transformer
    3. Flow based sampling
2. Reduction of cost in measurements
    4. Bias corrected approximation
    5. Control variates

I omitted a lot of important works due to the time limit (see [1])

[1] https://conference.ippp.dur.ac.uk/event/1265/contributions/7074/attachments/5662/7534/plenary.pdf

Machine learning + LQCD?

Introduction

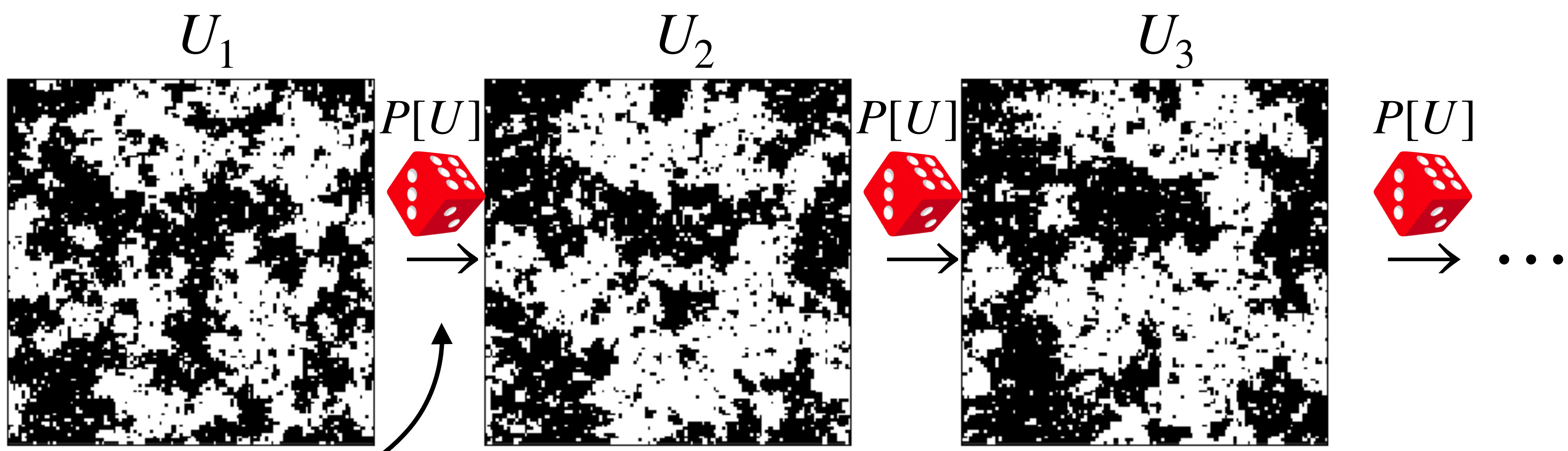
Monte-Carlo integration is available

M. Creutz 1980

Target integration
= expectation value $\langle \mathcal{O} \rangle = \frac{1}{Z}\int \mathcal{D}U e^{-S_{\text{eff}}[U]}\mathcal{O}(U)$

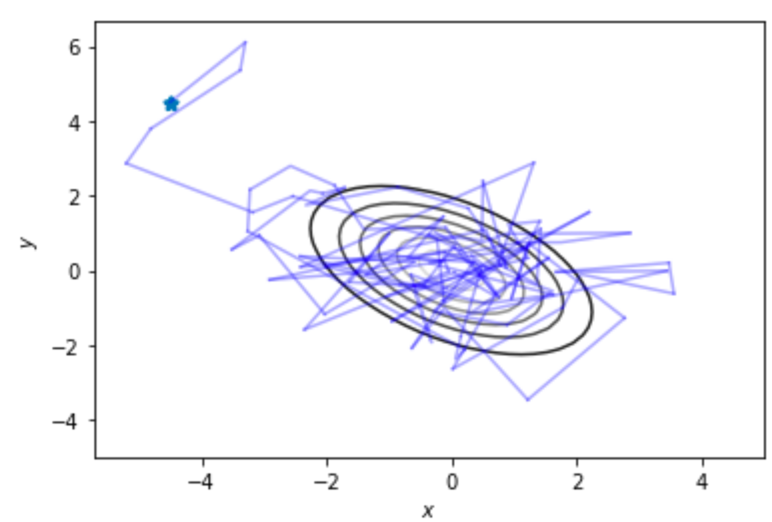
$S_{\text{eff}}[U] = S_{\text{gauge}}[U] - \log\det(\not{D}[U] + m)$

Monte-Carlo: Generate field configurations with "$P[U] \propto e^{-S_{\text{eff}}[U]}$". It gives expectation value



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

Random momentum + EOM
= Random walk like algorithm

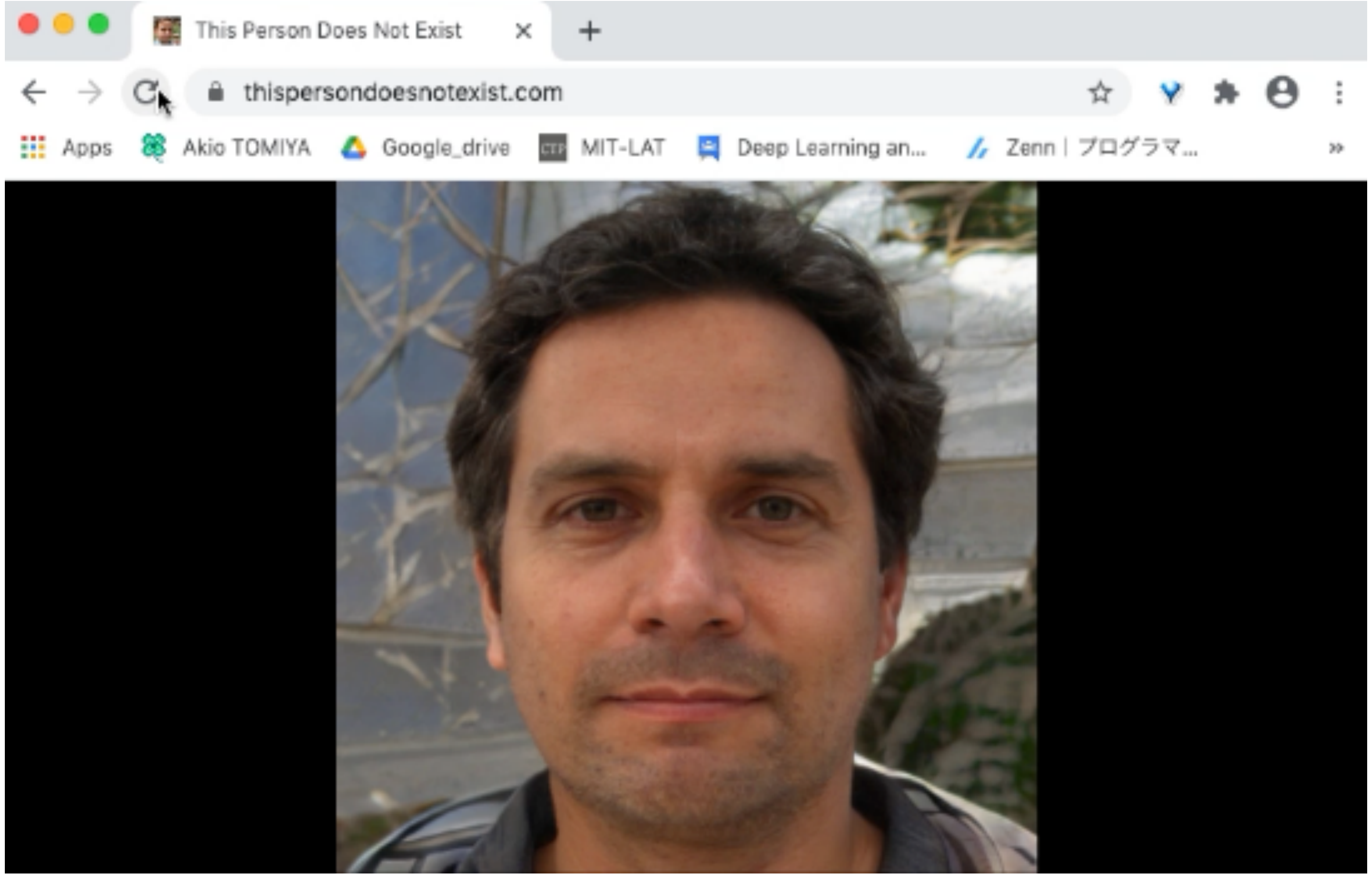


$S(x, y) = \frac{1}{2}(x^2 + y^2 + xy)$

Introduction

Generative neural net can make human face images

Neural nets can generate realistic human faces (Style GAN2)

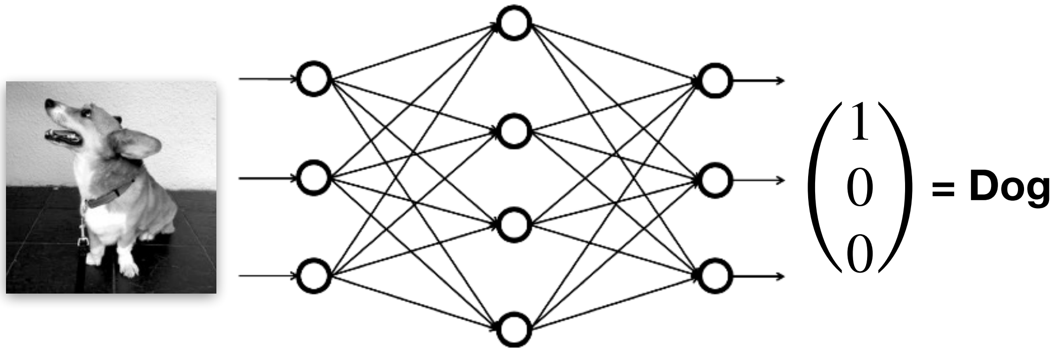


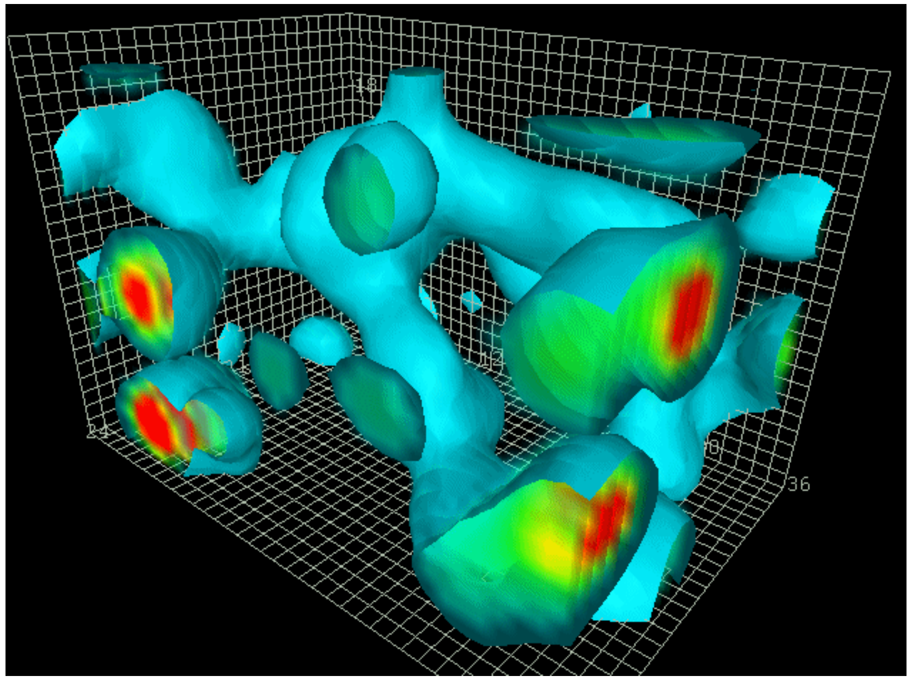
Realistic Images can be generated by machine learning!
Configurations as well? (proposals ~ images?)

Introduction

Machine learning for LQCD, LQCD with machine learning

- Machine learning/ Neural networks
  - data processing techniques for 2d/3d data in the real world (pictures)
  - (Variational) Approximation ($\sim$ fitting)
  - Generative NN can generate images/pictures
- Lattice QCD is more complicated than pictures
  - 4 dimension/relativistic
  - Non-abelian **gauge symmetry** (difficult)
  - Fermions (anti-commuting/fully quantum) -> Non-local effective correlation in gauge field
  - **Exactness** in MCMC 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

+…

This is not complete list. Related to lattice field theory and biased

Applications on LQCD

Machine learning for lattice QCD

1. Configuration generation in LQCD
    1. Self-learning MC with gauge covariant net
    2. CASK: Gauge symmetric transformer
    3. Flow based sampling
2. Reduction of cost in measurements
    4. Bias corrected approximation
    5. Control variates

I omitted a lot of important works due to the time limit (see [1])

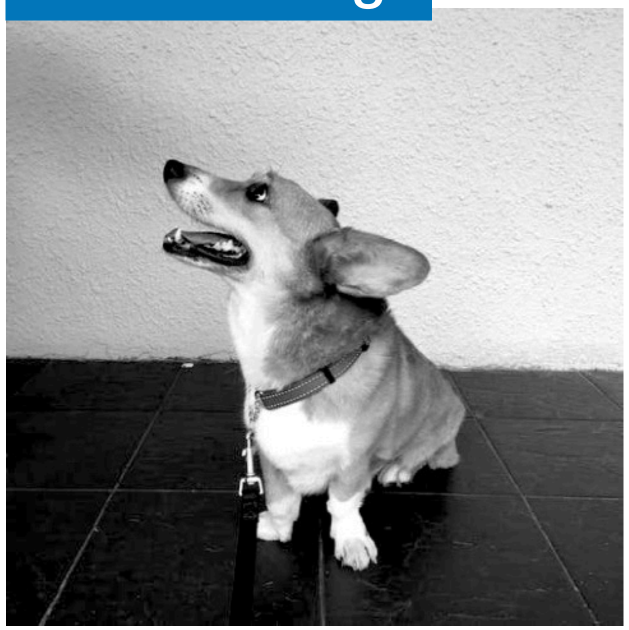
[1] https://conference.ippp.dur.ac.uk/event/1265/contributions/7074/attachments/5662/7534/plenary.pdf

Configuration generation in LQCD

Akio Tomiya

Convolution layer = trainable filter

Filter on image



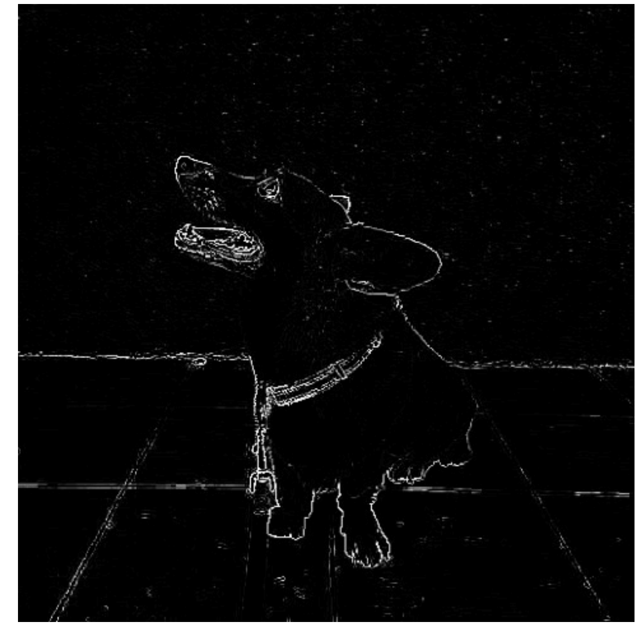
$*$

Laplacian filter

0	1	0
1	-2	1
0	1	0

(Discretization of ∂^2)

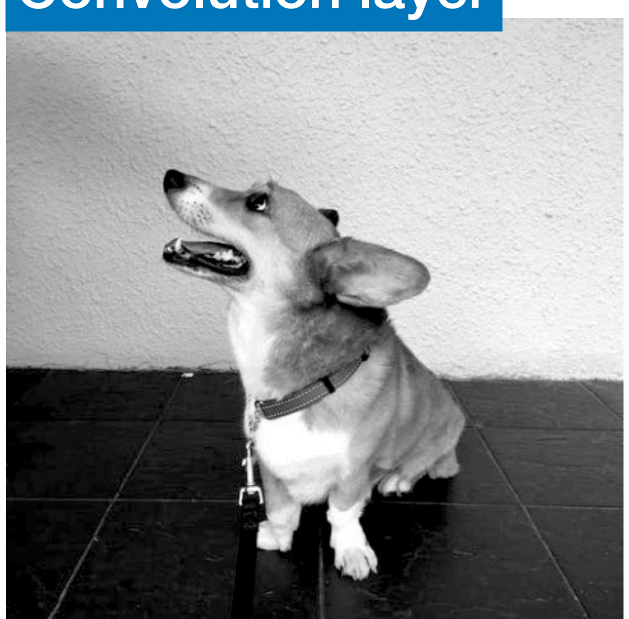
$=$



Edge detection

If input is shifted, output is shifted= respects transnational symmetry

Convolution layer



$*$

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)

...

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

Gaussian filter

$\frac{1}{16}$

1	2	1
2	4	2
1	2	1

(Training and data determines what kind of filter is realized)
Extract features

Convolution respects transnational symmetry as well

Configuration generation in LQCD

Smearing = Smoothing of gauge fields

Eg.

Coarse image

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

We want to smoothen *gauge* field configurations with keeping *gauge* symmetry

Two types:

APE-type smearing

Stout-type smearing

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

Configuration generation in LQCD

Akio Tomiya

Smearing ~ neural network with fixed parameter!

AT Y. Nagai arXiv: 2103.11965

General form of smearing (~smoothing, averaging in space)

$$\begin{cases} z_\mu(n) = w_1 U_\mu(n) + w_2 \mathcal{G}[U] & \text{Summation with gauge sym} \\ U_\mu^{\text{fat}}(n) = \mathcal{N}(z_\mu(n)) & \text{A local function (Projecting on the gauge group)} \end{cases}$$

It has similar structure with neural networks,

$$\begin{cases} z_i^{(l)} = \sum_j w_{ij}^{(l)} u_j^{(l-1)} + b_i^{(l)} & \text{Matrix product vector addition} \\ u_i^{(l)} = \sigma^{(l)}(z_i^{(l)}) & \text{element-wise (local) Non-linear transf.} \end{cases}$$

Typically σ ~ tanh shape

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

Multi-level smearing = Deep learning (with given parameters)

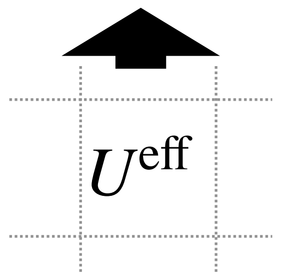
As same as the convolution, we can train weights.

Configuration generation in LQCD

Simulation parameter

Akio Tomiya

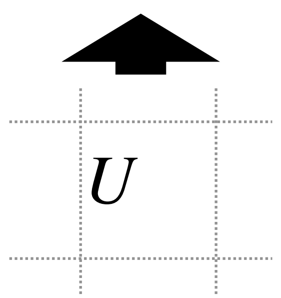
Construct effective action using operators with U^{eff}



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

Gauge covariant neural net (Adaptive smearing)

arXiv: 2103.11965



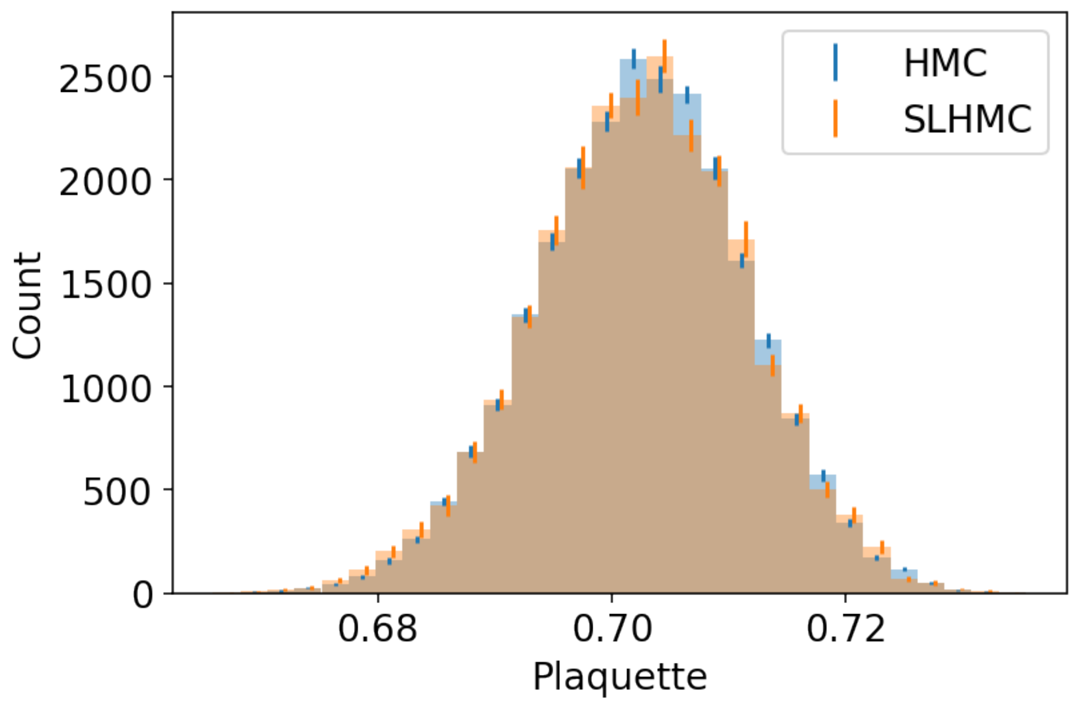
LatticeQCD.jl

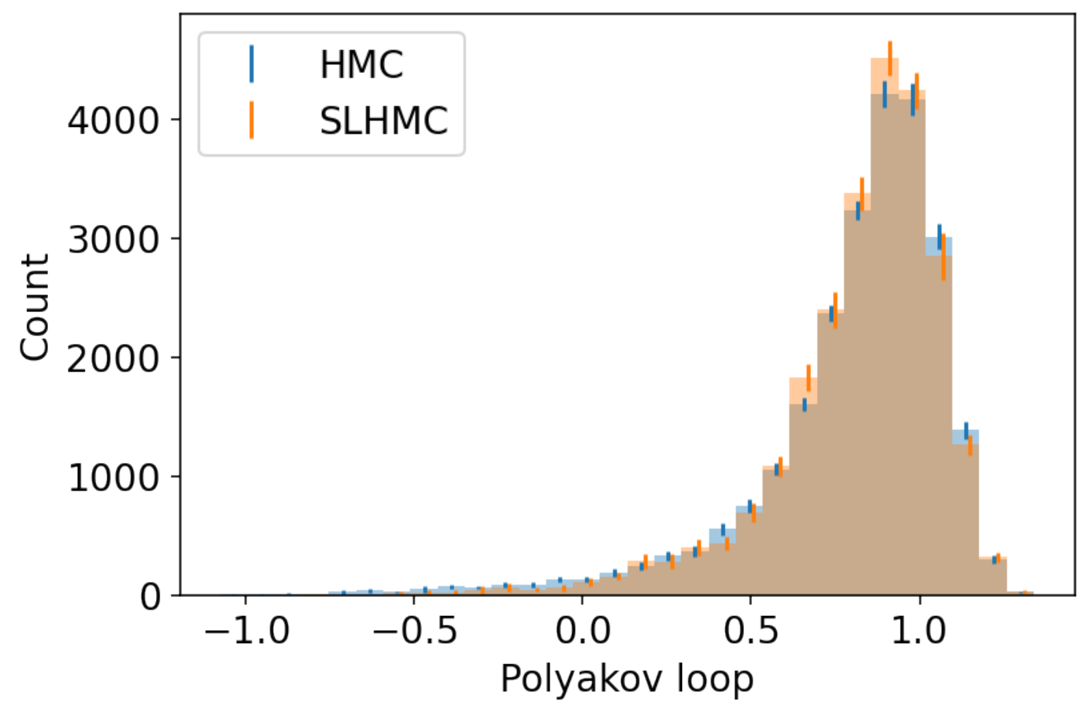
- Self-learning HMC (1909.02255, 2021 AT+), an exact algorithm
  - Exact Metropolis test and MD with effective action
- Target S : $m = 0.3$, dynamical staggered fermion, Nf=2, $L^4 = 4^4$, SU(2), $\beta = 2.7$. In Metropolis test
- Effective action S^{eff} in Molecular dynamics
  - Same gauge action
  - $m_{\text{eff}} = 0.4$ dynamical staggered fermion, Nf=2
    - Gauge covariant neural network (adaptive stout)
    - Bare U is fed, adaptively smeared U^{eff} is pop out
  - U **links are replaced by** U^{eff} **in** D_{stag}
- “Adaptively reweighted HMC”

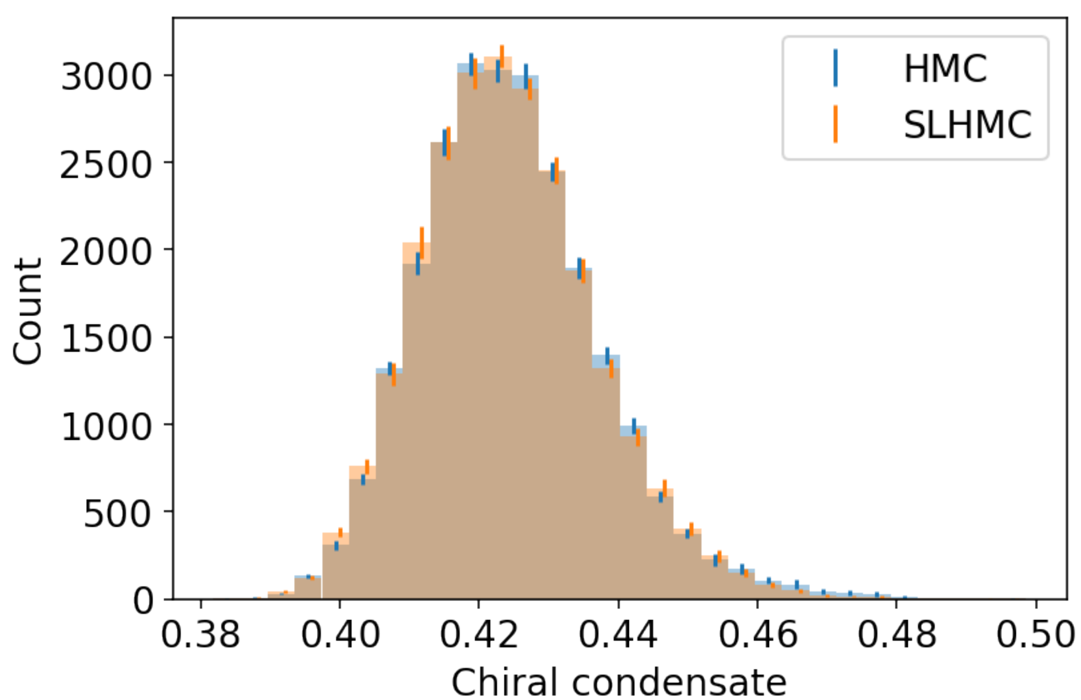
Configuration generation in LQCD

Application for the Full QCD in 4d

AT Y. Nagai arXiv: 2103.11965







Expectation value Acceptance = 40%

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)

What is showed?
Covariant net can mimic/absorb mass difference
SLHMC (~Adaptive reweighting) works

Applications on LQCD

Machine learning for lattice QCD

1. Configuration generation in LQCD
    1. Self-learning MC with gauge covariant net
    2. CASK: Gauge symmetric **transformer**
    3. Flow based sampling
2. Reduction of cost in measurements
    4. Bias corrected approximation
    5. Control variates

I omitted a lot of important works due to the time limit

[1] https://conference.ippp.dur.ac.uk/event/1265/contributions/7074/attachments/5662/7534/plenary.pdf

Configuration generation in LQCD

Akio Tomiya

Attention layer used in Transformers (GPT, Bard)

arXiv:1706.03762

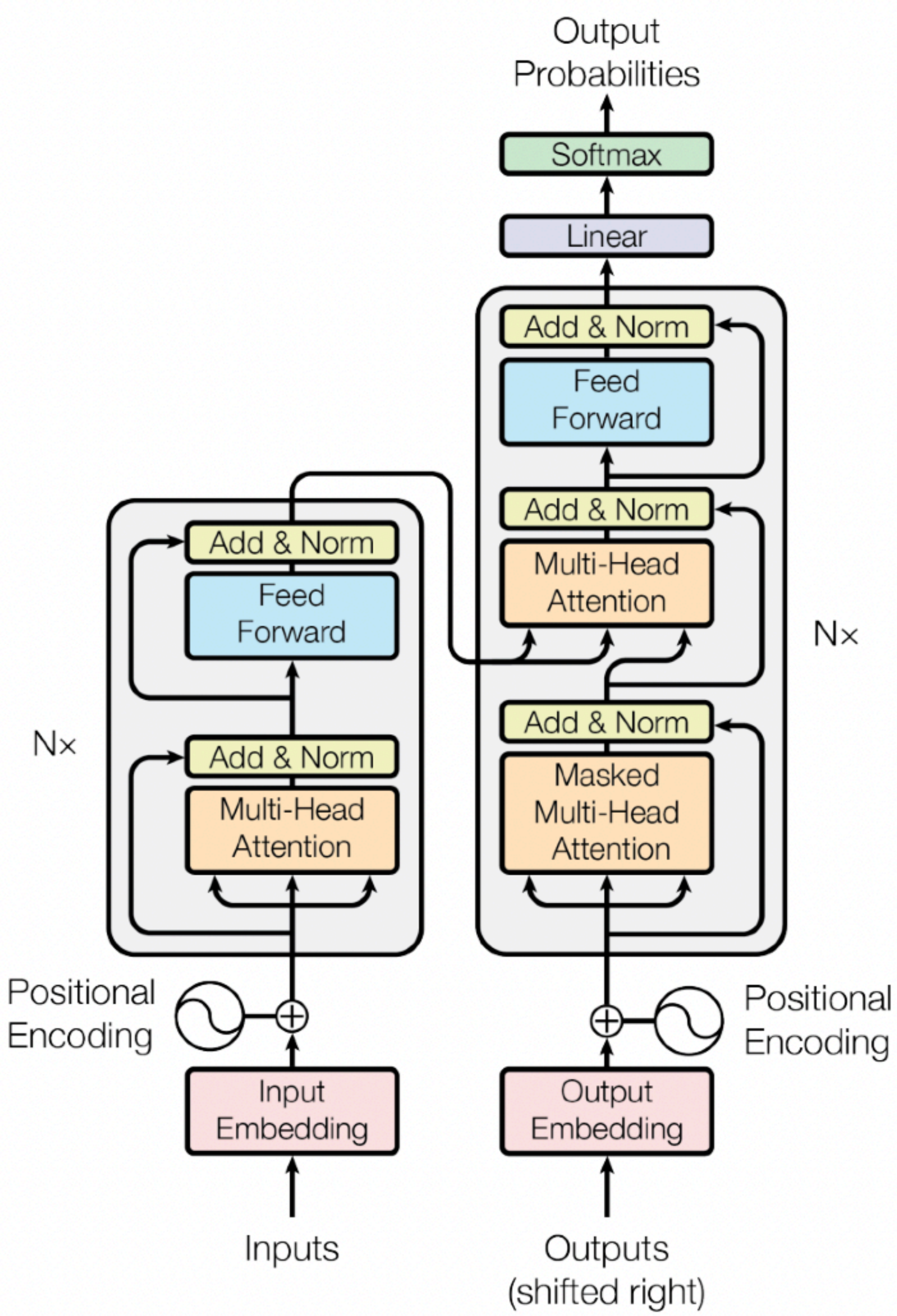


Figure 1: The Transformer - model architecture.

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.

Configuration generation in LQCD

Akio Tomiya

Attention layer can capture non-local correlations

arXiv:1706.03762

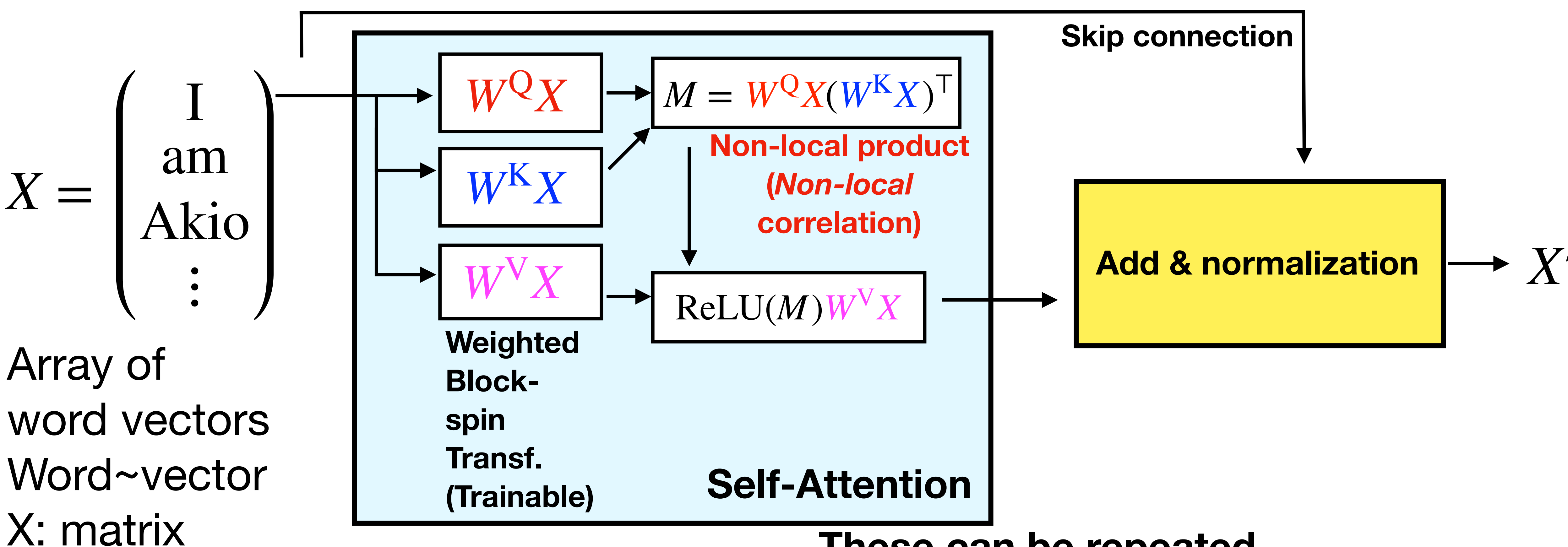
Modifier in language can be non-local

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

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

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

<u>Simplified version of Attention/Transformer</u>



These can be repeated

Configuration generation in LQCD

Akio Tomiya

Transformer shows scaling lows (power law)

arXiv: 2001.08361

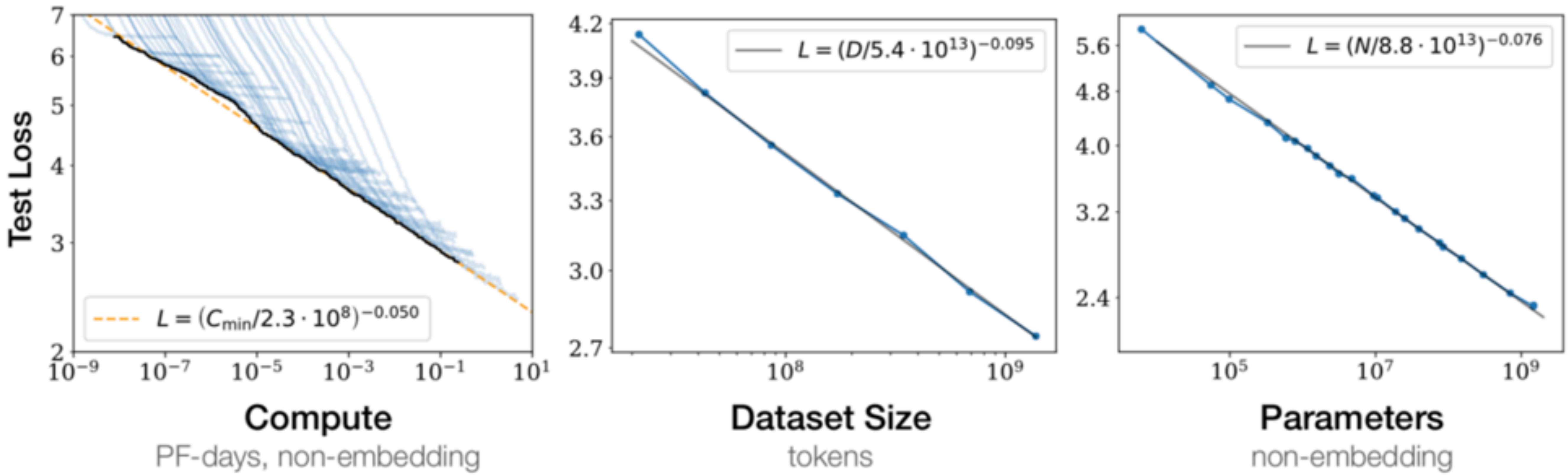


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when 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**

Configuration generation in LQCD

Akio Tomiya

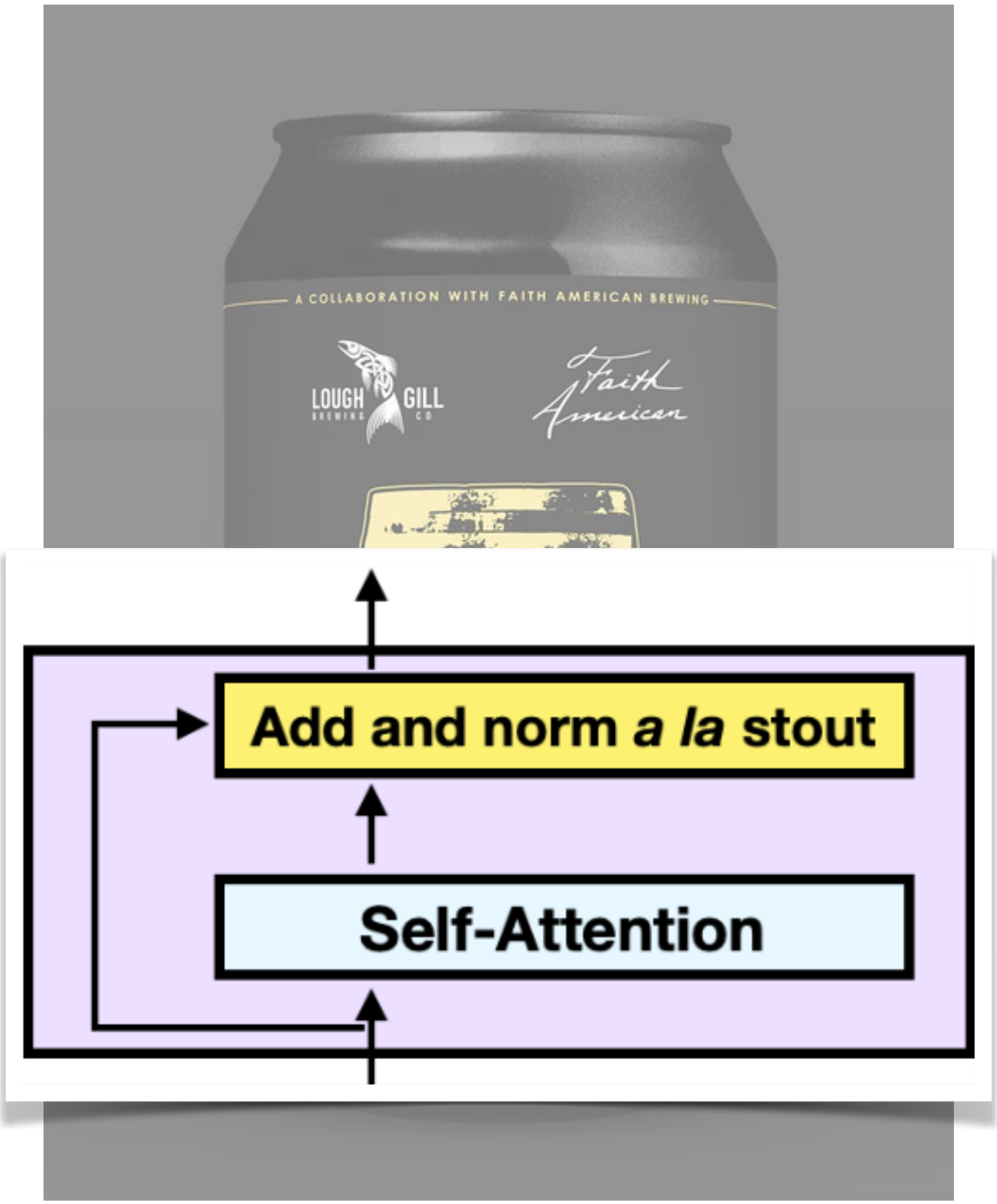
CASK?

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

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

Configuration generation in LQCD

CASK = Stout kernel, gauge covariant transformer for LQCD



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😜

Configuration generation in LQCD

Akio Tomiya

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

Configuration generation in LQCD

Akio Tomiya

Idea: Attention must be invariant

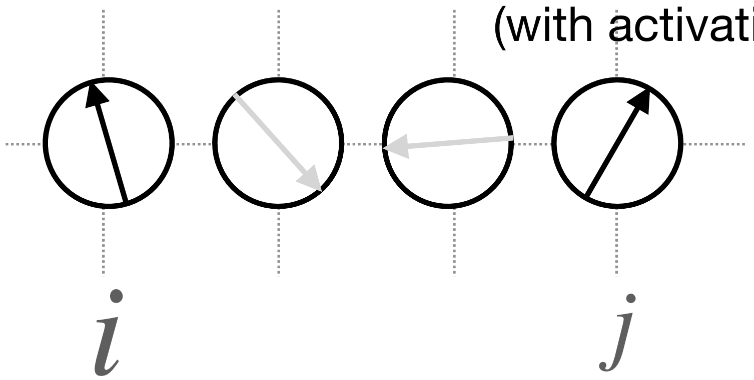
Attention matrix in transformer ~ correlation function (with block-spin transformed spin)

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

Transformer for Kondo spins

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

(with activation)



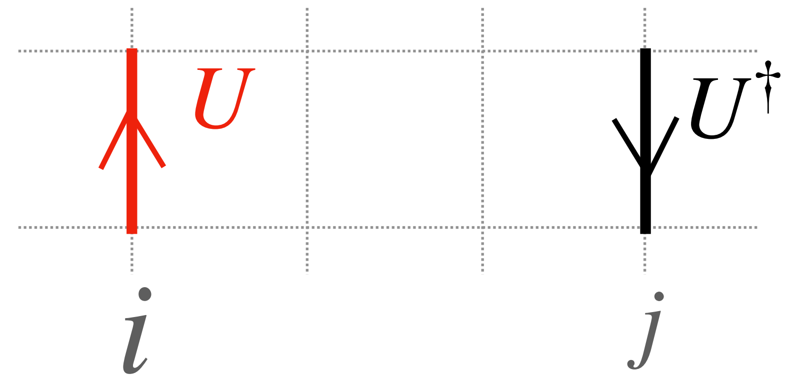
invariant
under global O(3)

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

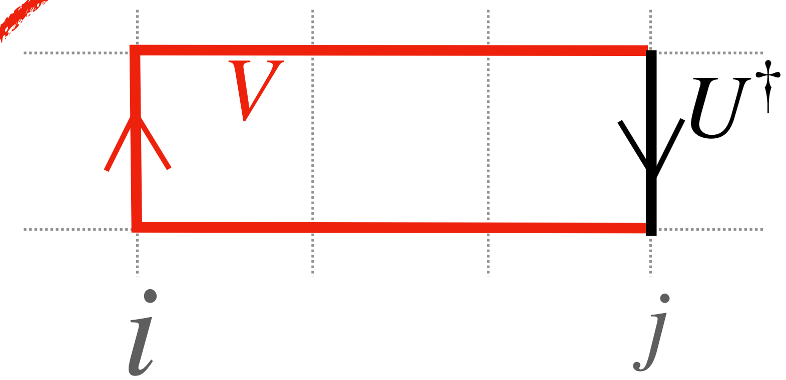
In total, output is covariant

2310.13222 AT+,2306.11527 AT+

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



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

WIP AT+

Configuration generation in LQCD

Structure of gauge symmetric attention using stout

Akio Tomiya

[1] 2021 AT+
WIP AT+

Procedure in three steps:

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

Loop operator projected on Lie algebra

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

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

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

cf. sparse attention, star attention

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

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

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

Loop operator projected on Lie algebra

Configuration generation in LQCD

Physically symmetric Attention layer for LQCD

[1] 2021 AT+
WIP AT+

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

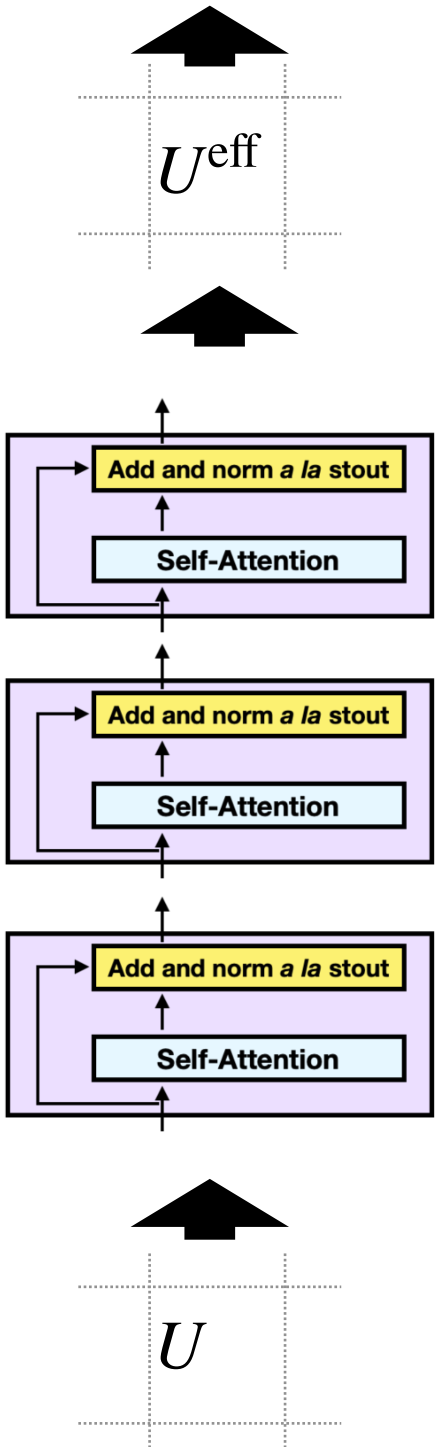
	Equivariance	**Gauge?**	**Capturable correlation**	**Data demmands**	**Applications**
Convolution (∈ equivariant layers)	Yes 👍	Yes 👍	Local 😲	Low 👍	VAE, GAN Normalizing flow SLHMC 2103.11965 AT+
Standard Attention layer arXiv:1706.03762	No 😲	No 😲	Global 👍	Huge 😲	ChatGPT GEMINI Vision Transformer
***Equivariant* attention for spin**	Yes 👍	No 😲	Global 👍	?	Kondo system (2310.13222 AT+ 2306.11527 AT+)
***Equivariant* attention for gauge**	Yes 👍	Yes 👍	Global 👍	?	WIP AT+

Configuration generation in LQCD

Simulation parameter

WIP AT+

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



LatticeQCD.jl

- 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), $\beta = 2.7$. In Metropolis test
- Effective action $S^{\rm eff}$ in Molecular dynamics
  - Same gauge action
  - $m_{\rm eff} = 0.4$ dynamical staggered fermion, Nf=2
  - CASK with plaquette covariant kernel
    - Attention = 7-links rect staple (=3 plaquette)
  - U links are replaced by $U^{\rm eff}$ in $D_{\rm stag}$
- “Adaptively reweighted HMC”

Configuration generation in LQCD

Loss = difference of action

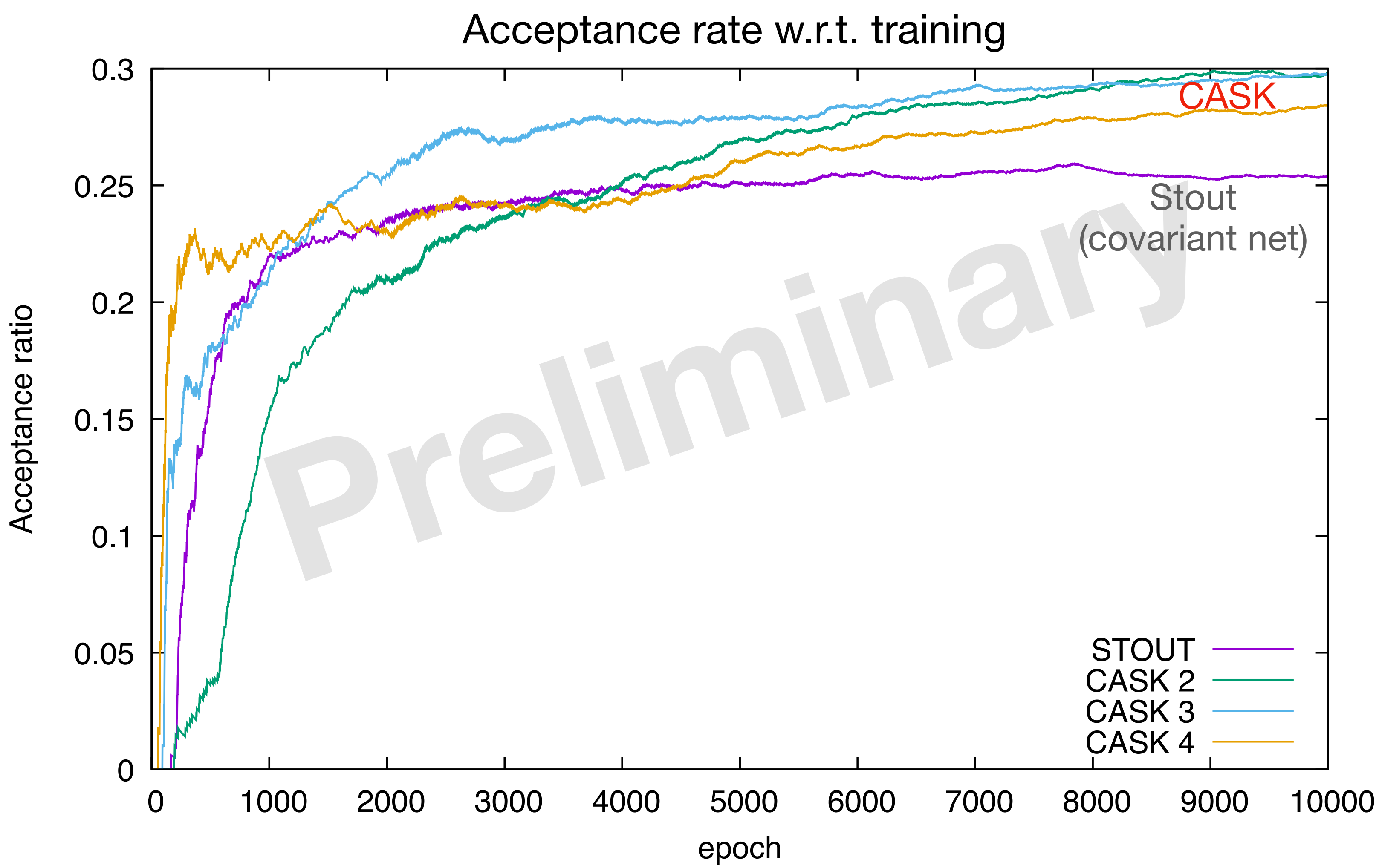
= MC steps

- Loss decreases along with the training steps
- it works as same as the stout (covariant net)
- Gain?

Configuration generation in LQCD

Attention blocks improve acceptance

WIP AT+



= MC steps

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

Applications on LQCD

Machine learning for lattice QCD

1. Configuration generation in LQCD
    1. Self-learning MC with gauge covariant net
    2. CASK: Gauge symmetric transformer
    3. Flow based sampling
2. Reduction of cost in measurements
    4. Bias corrected approximation
    5. Control variates

I omitted a lot of important works due to the time limit

[1] https://conference.ippp.dur.ac.uk/event/1265/contributions/7074/attachments/5662/7534/plenary.pdf

Configuration generation in LQCD

Akio Tomiya

Change of variables makes problem easy

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

Evaluation of Path integral
is hard
(1M dimensional integration)
a lot of cross terms

Back to high school,

- Integration by parts
- Change of variables

Are there any good "Change of variables" for QFT?
(to remove correlation)

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

Configuration generation in LQCD

Akio Tomiya

Change of variables makes problem easy

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

QFT

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

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

If **this does not have position dependence,**
integration is easy (trivially done on machine)
“Trivializing map”

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

Configuration generation in LQCD

Viewpoint: Change of variables makes problem easy

Simplest example: Box Muller

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

Change of variables

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

Target integral: hard

Easy

Change of variables sometimes problem easier (this case, it makes the measure flat)

RHS is flat measure
→We can sample like right eq.
(uniform)

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

We can reconstruct
a field config x, y
for original theory
like right eq.

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

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

Configuration generation in LQCD

Akio Tomiya

Gradient flow as a trivializing map

Trivializing map for lattice QCD has been demanded…

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \cdots \int \prod_{x \in 100} \prod_{y \in 100} \prod_{z \in 100} \prod_{t \in 100} d\phi_{x,y,z,t} \mathrm{e}^{-S(\phi)} \mathcal{O}[\phi_{x,y,z,t}]$$

$$\tilde{\phi} = \mathcal{F}_\tau(\phi)$$

Flow equation (change variable)
"Trivializing map"

If the solution satisfies $S(\mathcal{F}_\tau(\phi)) + \ln \det(\text{Jacobian}) = \sum_n \tilde{\phi}_n^2$,

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \cdots \int \prod_{x \in 100} \prod_{y \in 100} \prod_{z \in 100} \prod_{t \in 100} d\tilde{\phi} \mathcal{O}[\mathcal{F}_\tau(\phi)] \mathrm{e}^{-\sum \tilde{\phi}_n^2}$$

It becomes Gaussian integral! Easy to evaluate!! Position independent.

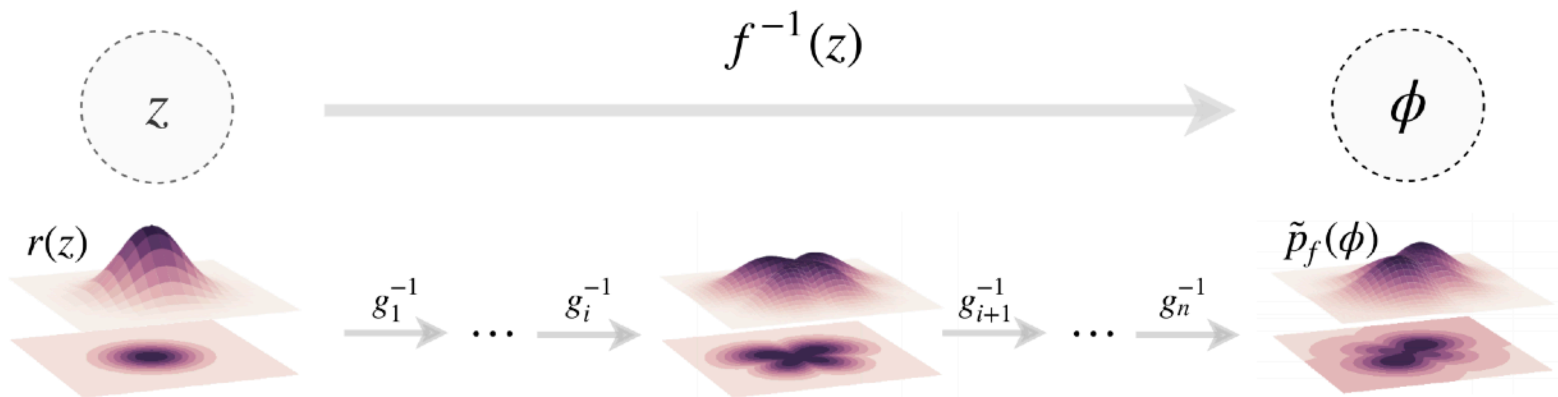
However, the Jacobian cannot evaluate easily, so it is not practical.
Life is hard.

M. Luscher arXiv:0907.5491
arxiv 1904.12072, 2003.06413, 2008.05456

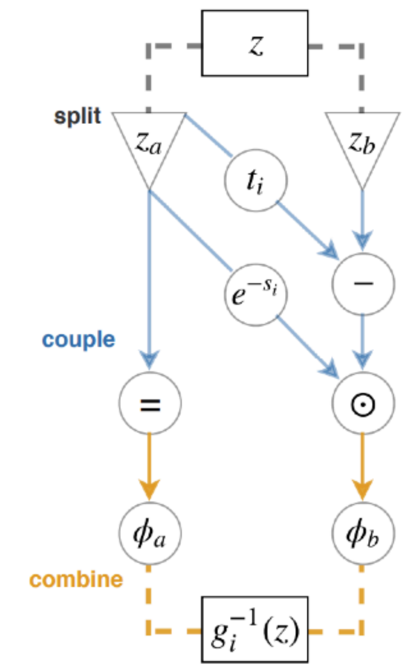
Configuration generation in LQCD

Flow based algorithm = neural net represented flow algorithm

MIT + DeepMind 2019~



(a) Normalizing flow between prior and output distributions



(b) Inverse coupling layer

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

Train a neural net as a "flow" $\tilde{\phi} = \mathscr{F}(\phi)$, Bijective.
If it is well represented, we can sample from a Gaussian
It can be done "Normalizing flow" (Real Non-volume preserving neural net)
Moreover, Jacobian is tractable!

arxiv 1904.12072, 2003.06413, 2008.05456

Configuration generation in LQCD

Akio Tomiya

Flow based ML for QFT

MIT + Deepmind + …

$$\int D\phi e^{-S[\phi]} O[\phi] \approx \underbrace{\prod_i \int d\varphi_i e^{-V(\varphi_i)} J[\varphi] O[F[\varphi]]}_{}$$

Original integral: hard

Easy

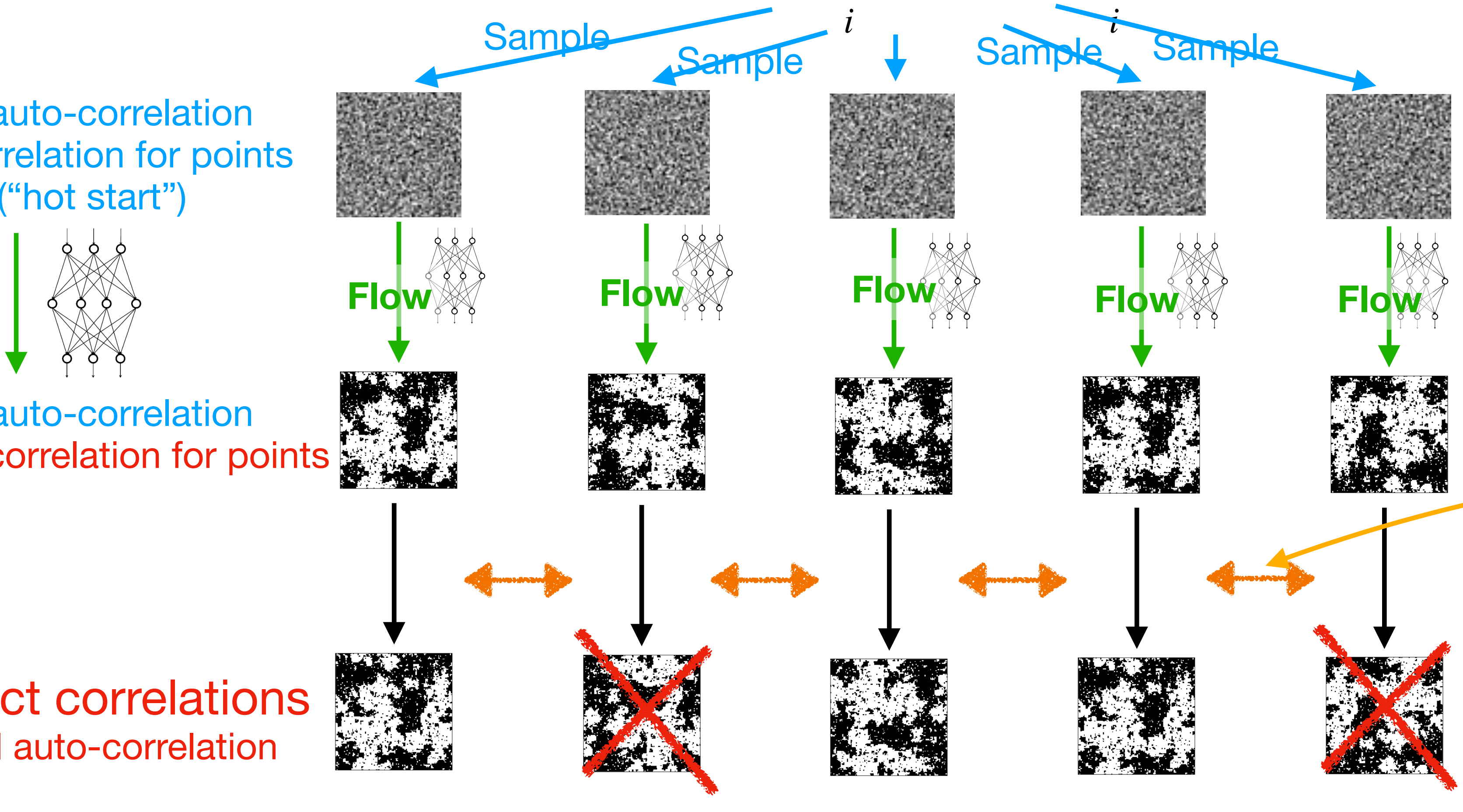
Flow-based sampling algorithm

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

No auto-correlation
No correlation for points
("hot start")

No auto-correlation
Approx.correlation for points

Correct correlations
Small auto-correlation



①Embarrassedly parallel sampling
from trivial theory
(no kinetic term, no topology,
~ Ising model at T>>0, random)

②"un-trivializing map"

"<u>Cooling</u> = change of variable"
via trained neural net

③Metropolis-*Hastings* with
$e^{-S}/e^{-V(\varphi_i)} J^{-1}[\varphi]$
Sequential

Auto-correlation
~ Rejection rate

Configuration generation in LQCD

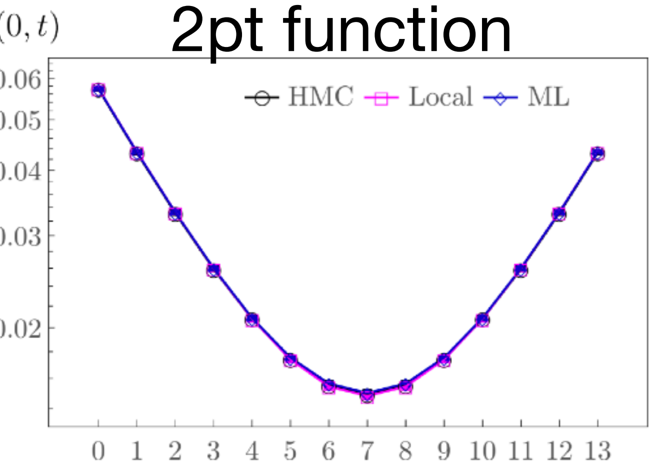
Akio Tomiya

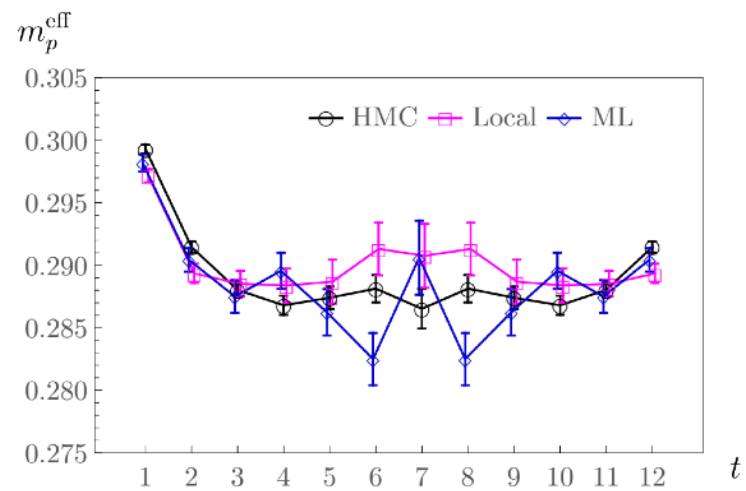
Flow based algorithm = neural net represented flow algorithm

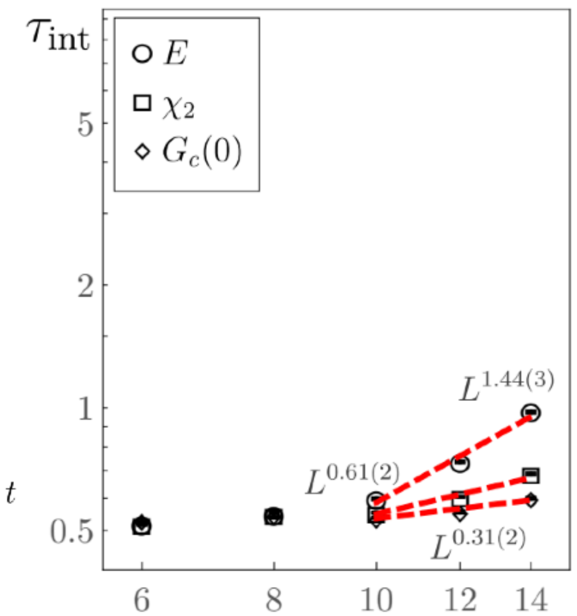
Real scalar in 2 dimension

MIT + DeepMind 2019~

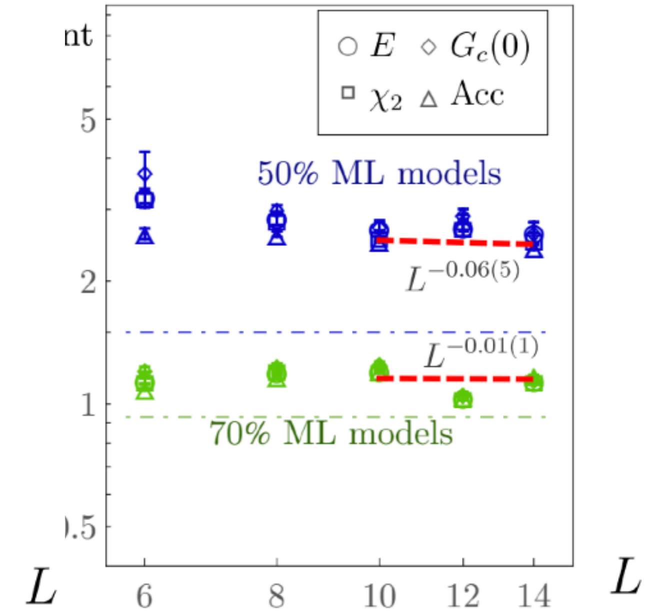
2pt function





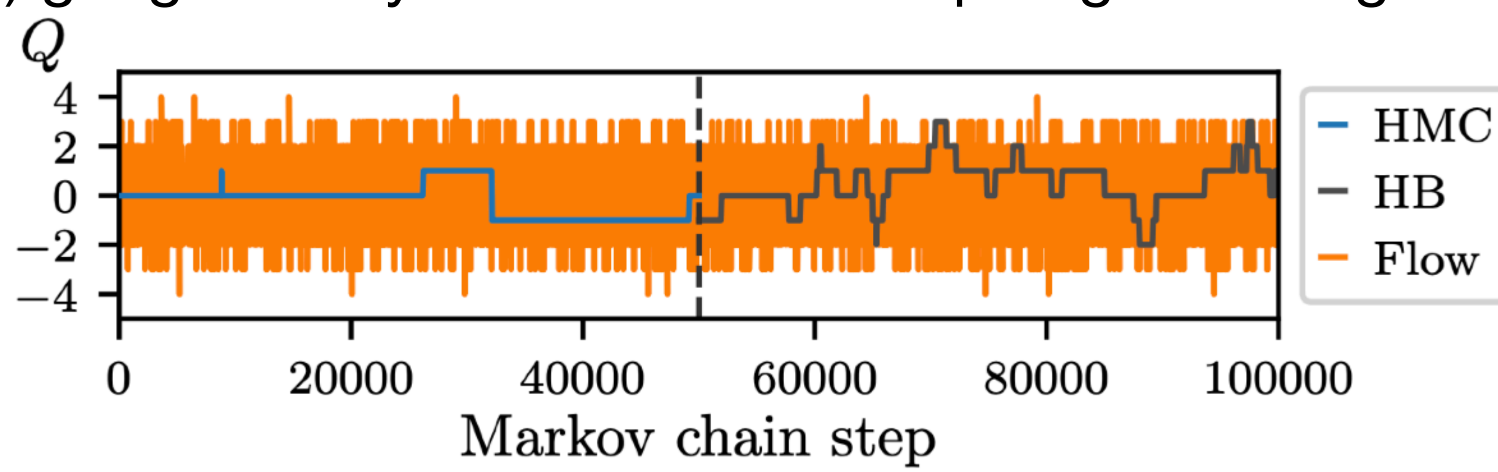


(a) HMC ensembles



(c) Flow-based MCMC ensembles

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



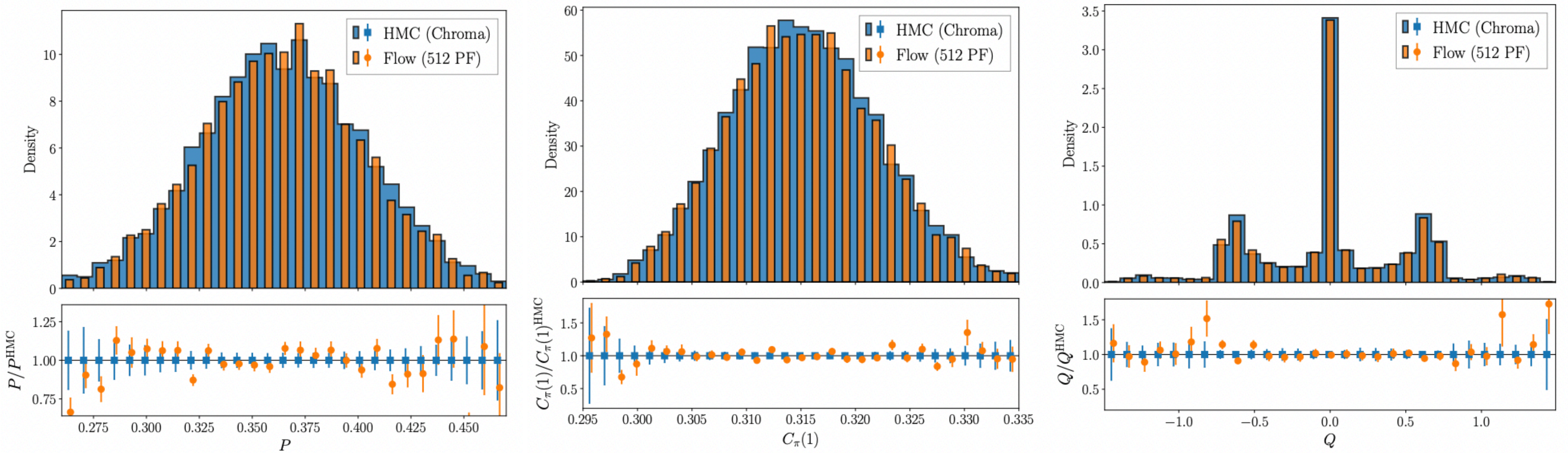
Applied already on SU(N) in 4d with dynamical fermions!

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?

[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

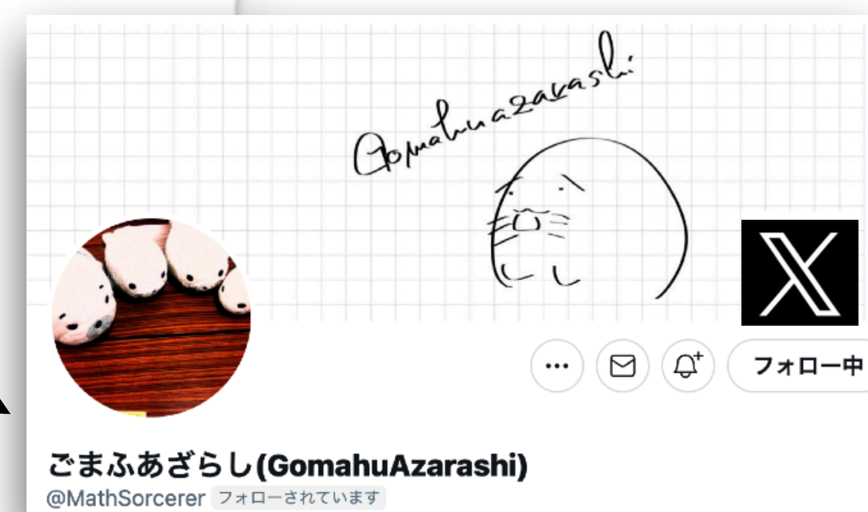
Mainly implement by Satoshi Terasaki

https://arxiv.org/abs/2208.08903

Abstract

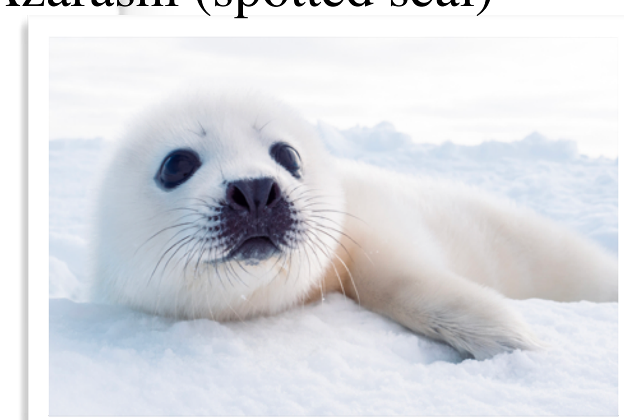
GomalizingFlow.jl: is a package to generate configurations for quantum field theory on the lattice using the flow based sampling algorithm in Julia programming language. This software serves two main purposes: to accelerate research of lattice QCD with machine learning with easy prototyping, and to provide 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



ごまふあざらし = GOMAfu

Azarashi (spotted seal)



↓ economic convolution for flow

Applications on LQCD

Machine learning for lattice QCD

1. Configuration generation in LQCD
   1. Self-learning MC with gauge covariant net:
   2. CASK: Gauge symmetric transformer
   3. Flow based sampling
2. Reduction of cost in measurements
   4. Bias corrected approximation
   5. Control variates

I omitted a lot of important works due to the time limit

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

Akio Tomiya

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

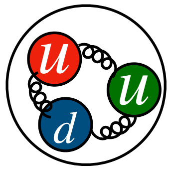
$$\frac{1}{N_{\mathrm{conf}}} \sum_{c=1}^{N_{\mathrm{conf}}} O^{(\mathrm{Appox})}[U_c]$$

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

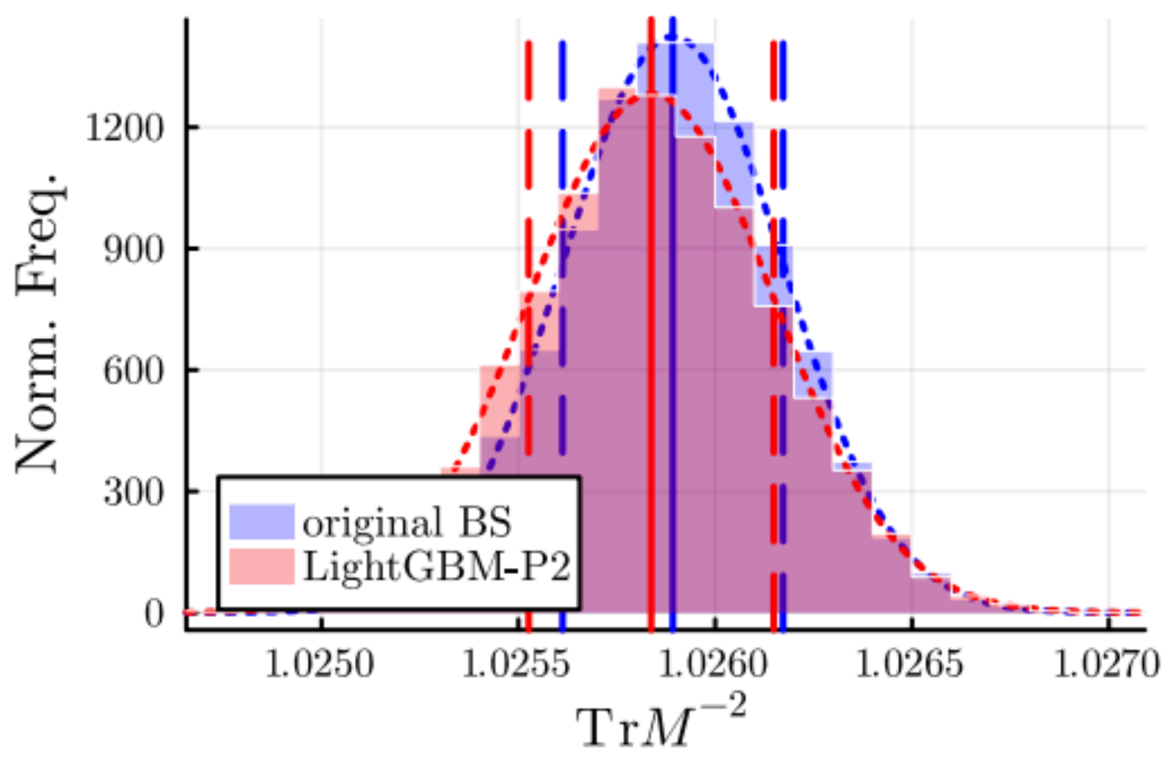
Akio Tomiya

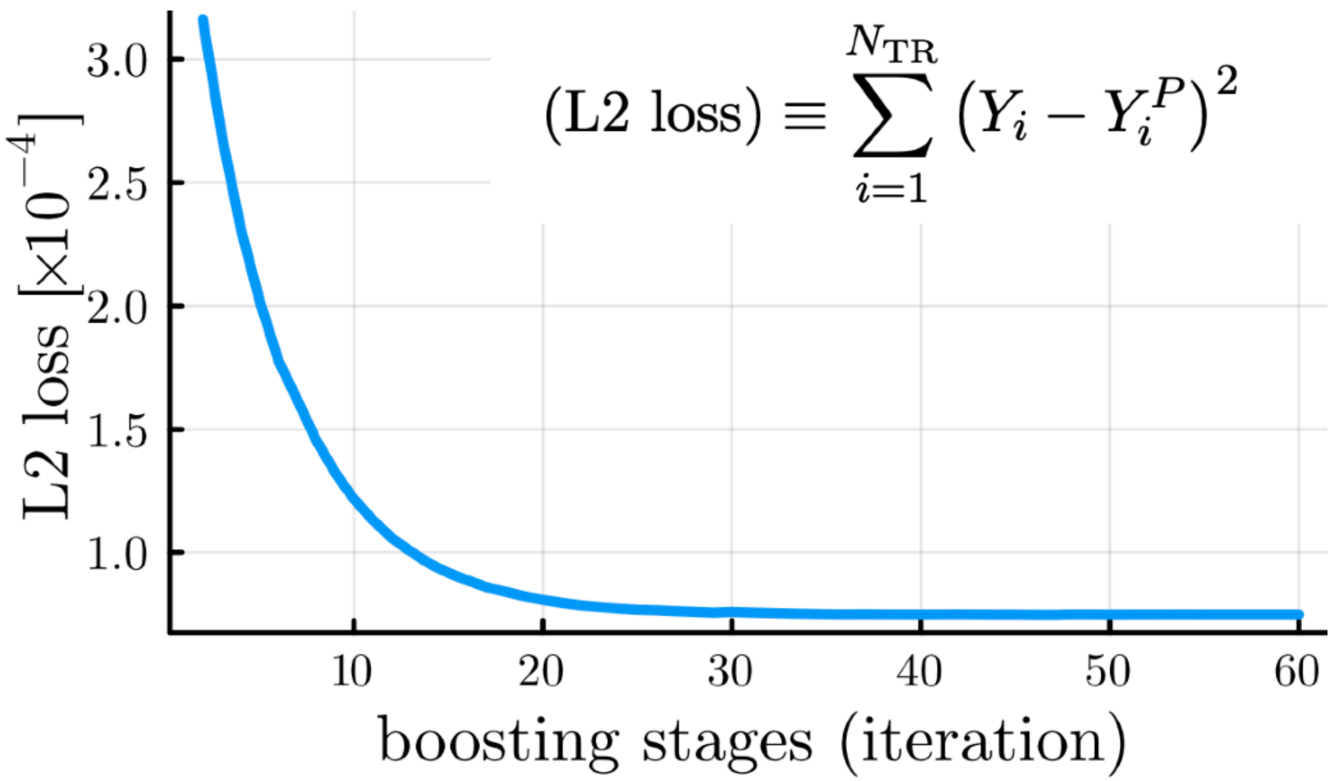
Reduction of cost in measurements

Reducing cost without bias from ML

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

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





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

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

Applications on LQCD
Machine learning for lattice QCD

1. Configuration generation in LQCD
    1. Self-learning MC with gauge covariant net:
    2. CASK: Gauge symmetric transformer
    3. Flow based sampling
2. Reduction of cost in measurements
    4. Bias corrected approximation
    5. Control variates

I omitted a lot of important works due to the time limit

Reduction of cost in measurements

Akio Tomiya

Control variates

See reference in [1]

Consider an observable on the lattice $\langle O \rangle$

$$\langle O \rangle = \lim_{N_{\rm conf}\to\infty} \frac{1}{N_{\rm conf}} \sum_{c=1}^{N_{\rm conf}} O[U_c]$$

For finite statistics, we have statistical error, which is propositional to the variance $\langle O^2 \rangle$

If we have, an estimator/observable f such that, …

$$\langle f \rangle = 0 \qquad \langle Of \rangle \gg 0$$

Then,

$$\langle (O - f) \rangle = \langle O \rangle \quad \text{and}$$

$$\langle (O - f)^2 \rangle = \langle O^2 \rangle + \langle f^2 \rangle - 2\underbrace{\langle Of \rangle}_{\text{Large}}$$

The operator $O - f$ has the same expectation value with O but it has small variance!

f is called control variates

[1] https://conference.ippp.dur.ac.uk/event/1265/contributions/7074/attachments/5662/7534/plenary.pdf
[2] https://conference.ippp.dur.ac.uk/event/1265/contributions/7596/attachments/5785/7609/lat2024_oh.pdf

Reduction of cost in measurements

Akio Tomiya

Control variates

See reference in [1]

$$\langle (O - f) \rangle = \langle O \rangle \quad \text{and} \quad \langle (O - f)^2 \rangle = \langle O^2 \rangle + \langle f^2 \rangle - 2\langle Of \rangle$$

- It sound too good to be true?
- How can we find control variates f ?
  - We can find it using Schwinger Dyson equation
  - or machine learning (next page)

[1] https://conference.ippp.dur.ac.uk/event/1265/contributions/7074/attachments/5662/7534/plenary.pdf
[2] https://conference.ippp.dur.ac.uk/event/1265/contributions/7596/attachments/5785/7609/lat2024_oh.pdf

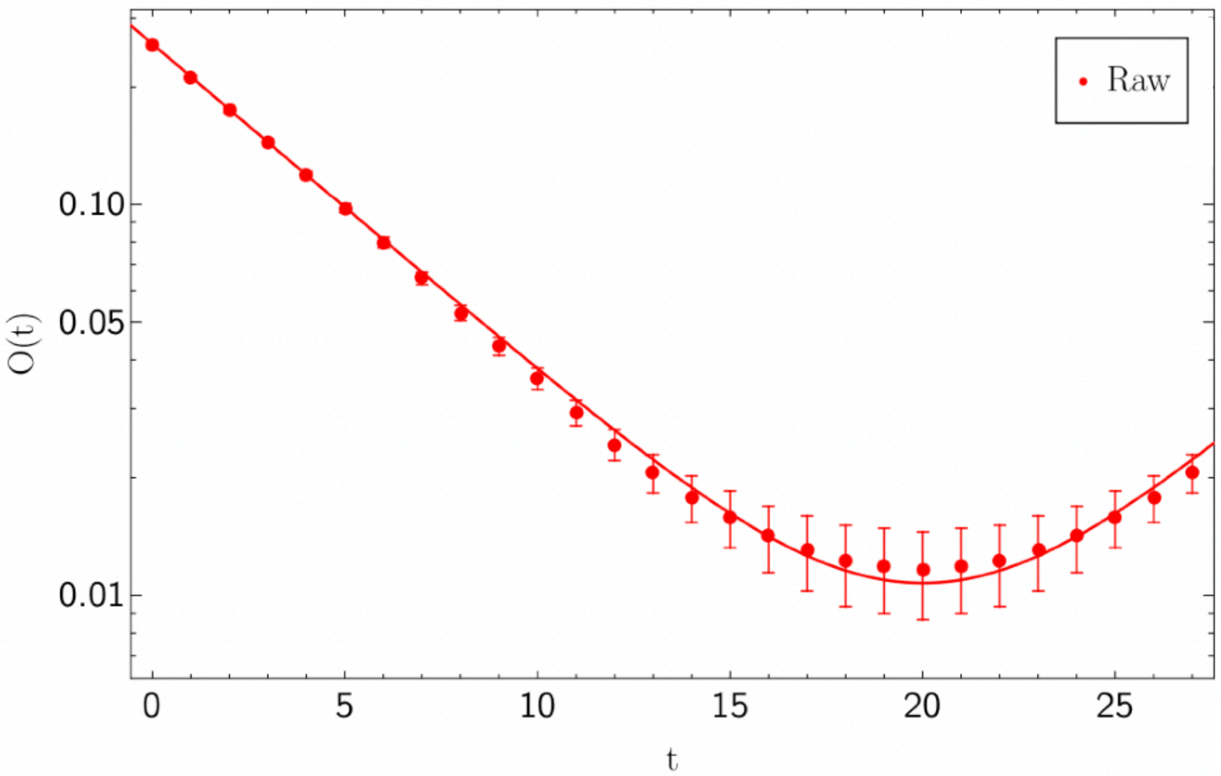
Reduction of cost in measurements

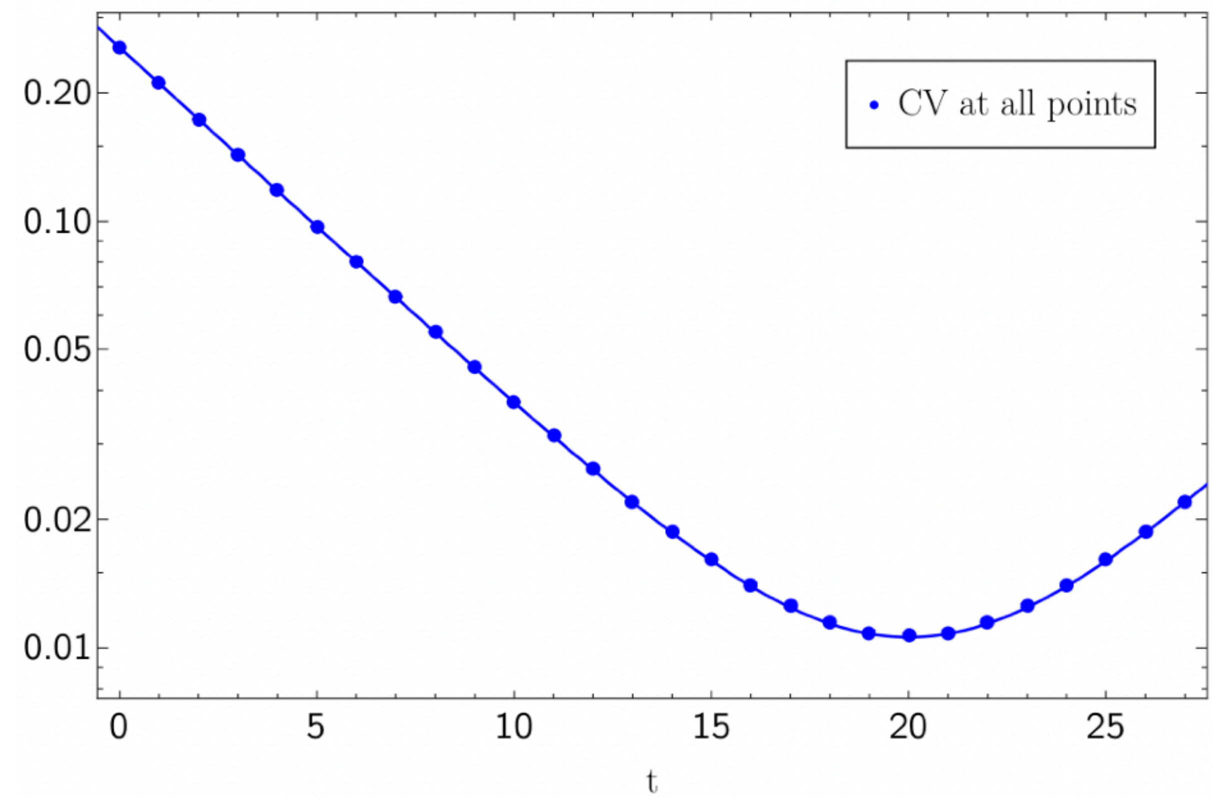
Control variates

See reference in [2]

Example: a scalar field theory in 1 + 1 dimensions.

$40 \times 10, m^2 = 0.01, \lambda = 0.1$





$$L(w) = \langle (O - f)^2 \rangle - \langle O - f \rangle^2$$

neural network parameters

$= \langle O \rangle^2$

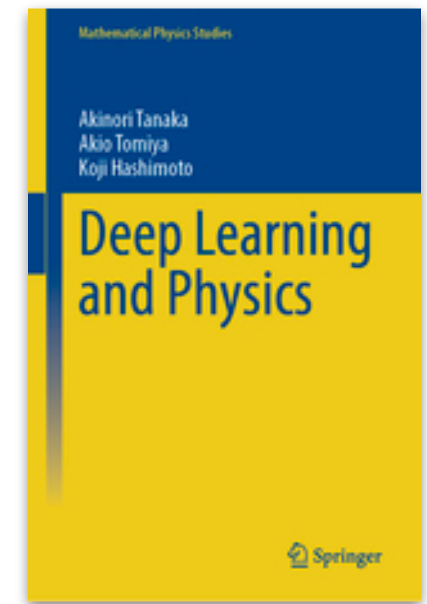
[1] https://conference.ippp.dur.ac.uk/event/1265/contributions/7074/attachments/5662/7534/plenary.pdf
[2] https://conference.ippp.dur.ac.uk/event/1265/contributions/7596/attachments/5785/7609/lat2024_oh.pdf

Akio Tomiya

Summary

Machine learning + lattice field theory

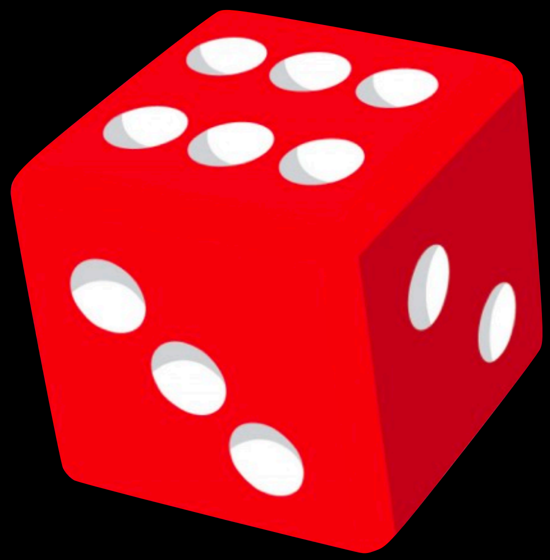
- Production and measurement need numerical cost
- Machine learning is useful for natural science/physics/Lattice QCD
  - to reduce cost in different ways
  - Supervised learning requires data ahead of training
  - Self-learning does not require it (SLHMC&Flow).
- Now, machine learning techniques are bias free
  - Gauge case, architectures are gauge covariant!
  - We can remove bias from ML
- Some results show better than existing algorithms (not all)
- Codes for LFT+ML are in Python but they are slow. We should go further.
  - Minimize code developing time + execution time.
- Julia might be good choice?



julia

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!



Akio Tomiya

Machine learning for theoretical physics

What am I?

I am a particle physicist, working on lattice QCD.

I want to apply machine learning on lattice QCD.

My papers 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

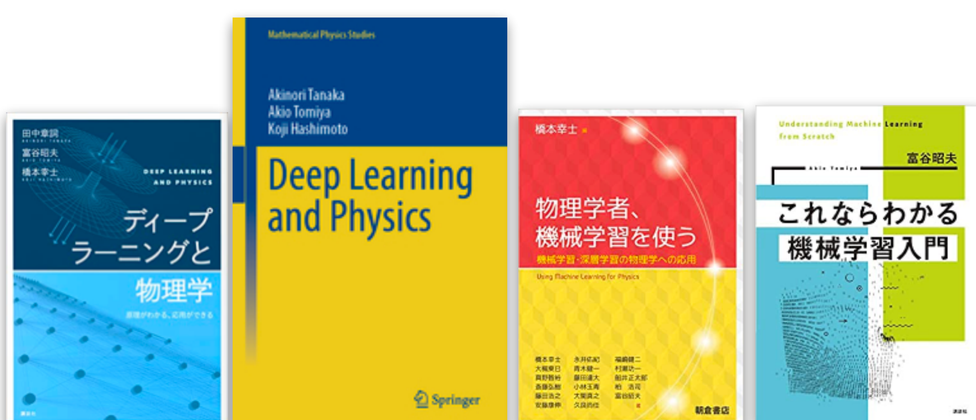
Detecting phase transition

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

Quantum computing for quantum field theory

Biography

2006-2010 : University of Hyogo (Superconductor)
2015 : PhD in Osaka university (Particle phys)
2015 - 2018 : Postdoc in Wuhan (China)
2018 - 2021 : SPDR in Riken/BNL (US)
2021 - : Assistant prof. in IPUT Osaka (ML/AI)



Organizing "Deep Learning and physics"

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

Kakenhi and others

Leader of proj A01 Transformative Research Areas, Fugaku

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

+quantum computer

Others:

Supervision of Shin-Kamen Rider
The 29th Outstanding Paper Award of the Physical Society of Japan
14th Particle Physics Medal: Young Scientist Award

Intro: Lattice QCD& Monte-Carlo

LQCD = Non-perturbative calculation of QCD

QCD in 3 + 1 dimension

$$S = \int d^4x \Big[-\frac{1}{2} \mathrm{tr}\, F_{\mu\nu}F^{\mu\nu} + \bar{\psi}\big(\mathrm{i}\partial\!\!\!/ + gA\!\!\!/ - m\big)\psi \Big]$$

$$Z = \int \mathscr{D}A\mathscr{D}\bar{\psi}\mathscr{D}\psi e^{\mathrm{i}S} \qquad F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu - \mathrm{i}g[A_\mu, A_\nu]$$

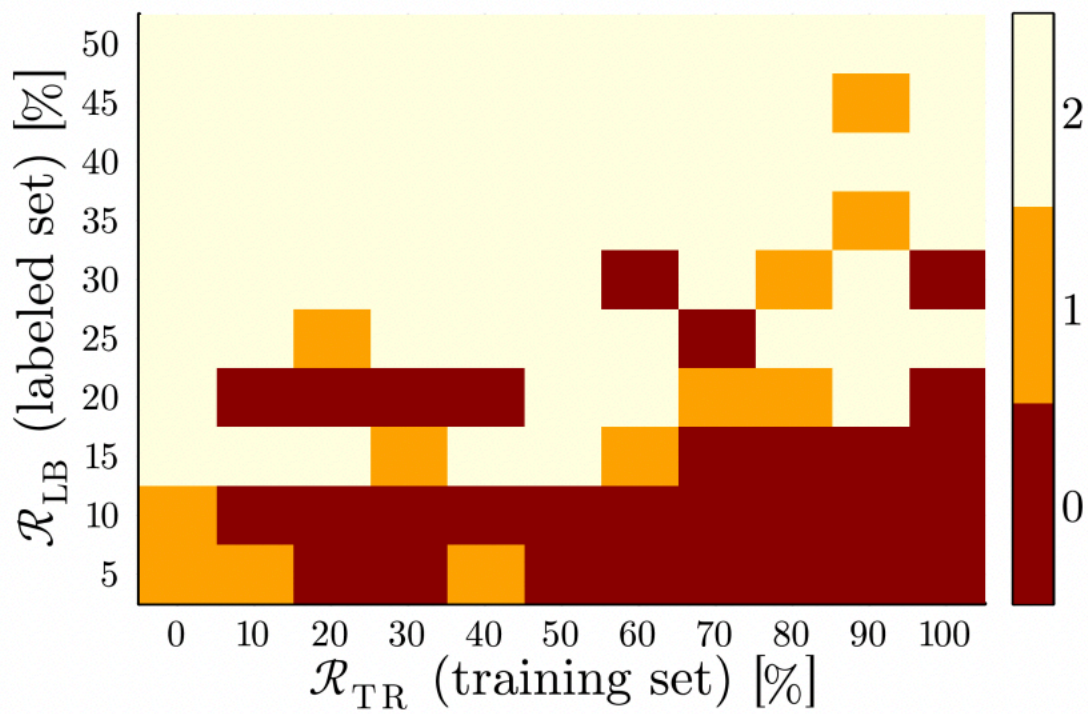
QCD in Euclidean 4 dimension (imaginary time)

$$S = \int d^4x \Big[+\frac{1}{2} \mathrm{tr}\, F_{\mu\nu}F_{\mu\nu} + \bar{\psi}\big(\partial\!\!\!/ - \mathrm{i}gA\!\!\!/ + m\big)\psi \Big]$$

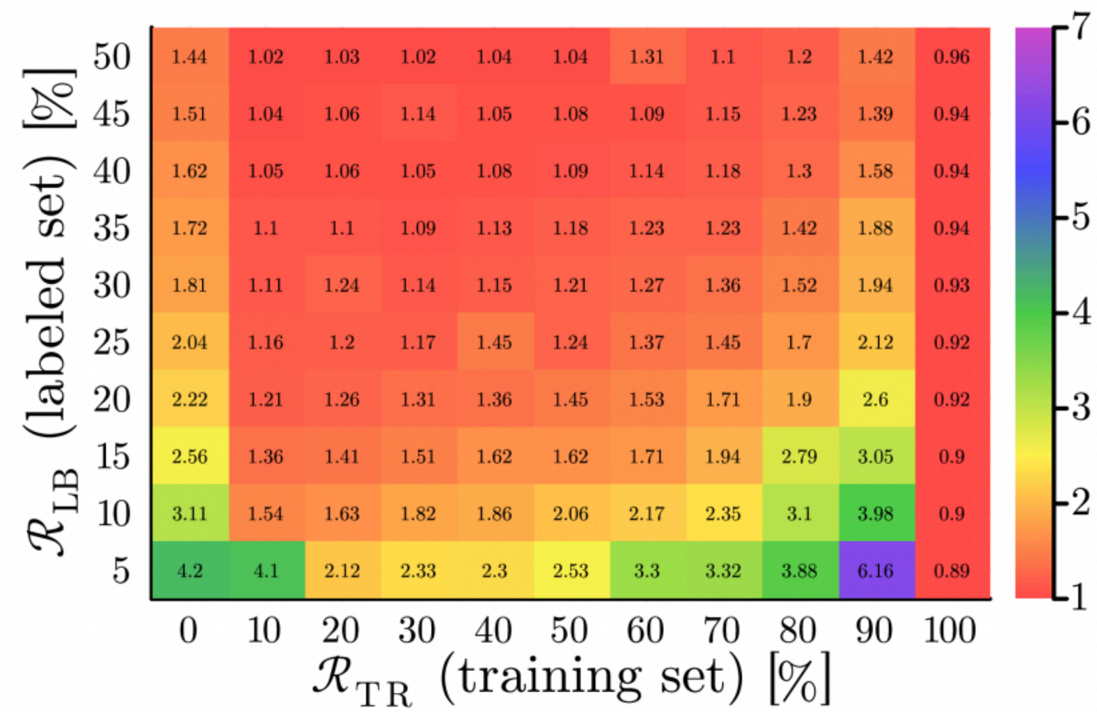
$$Z = \int \mathscr{D}A\mathscr{D}\bar{\psi}\mathscr{D}\psi e^{-S}$$

- Same Hamiltonian with real-time formalism
- Static property is the same (mass etc)
- How to calculate?

Example: Plaquette $\to \mathrm{Tr}M^{-3}$ estimation ($\mathcal{P}2$, ID-0)

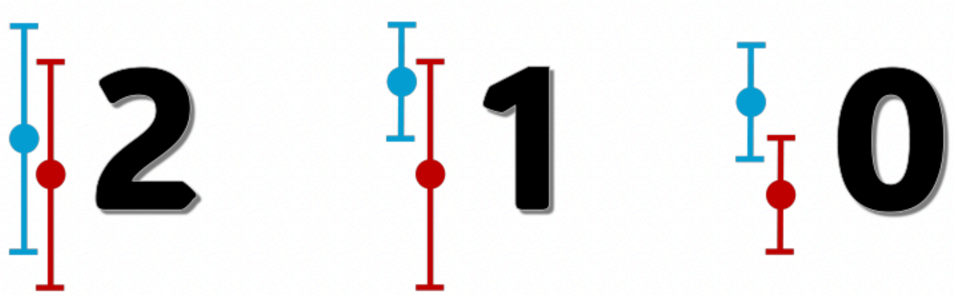


(a) $\bar{Y}$ (central value) check



(b) Magnitude of $\sigma_{\mathcal{P}2}/\sigma_{\mathrm{Orig.}}$

- $\bar{Y}$ score: white, orange, red



1 **Eval. 1:** central value check
➡ consistently white region
n $\mathcal{R}_{\mathrm{LB}} \geq 30\,\%$, $\mathcal{R}_{\mathrm{TR}} \leq 50\,\%$

2 **Eval. 2:** $\sigma_{\mathcal{P}2}/\sigma_{\mathrm{Orig.}}$ check
➡ Roughly $\boldsymbol{\sigma_{\mathcal{P}2} \lesssim 1.1\sigma_{\mathrm{Orig.}}}$
in $\mathcal{R}_{\mathrm{LB}} \geq 30\,\%$, $\mathcal{R}_{\mathrm{TR}} \leq 50\,\%$

Motivation

Monte-Carlo integration is available, but still expensive!

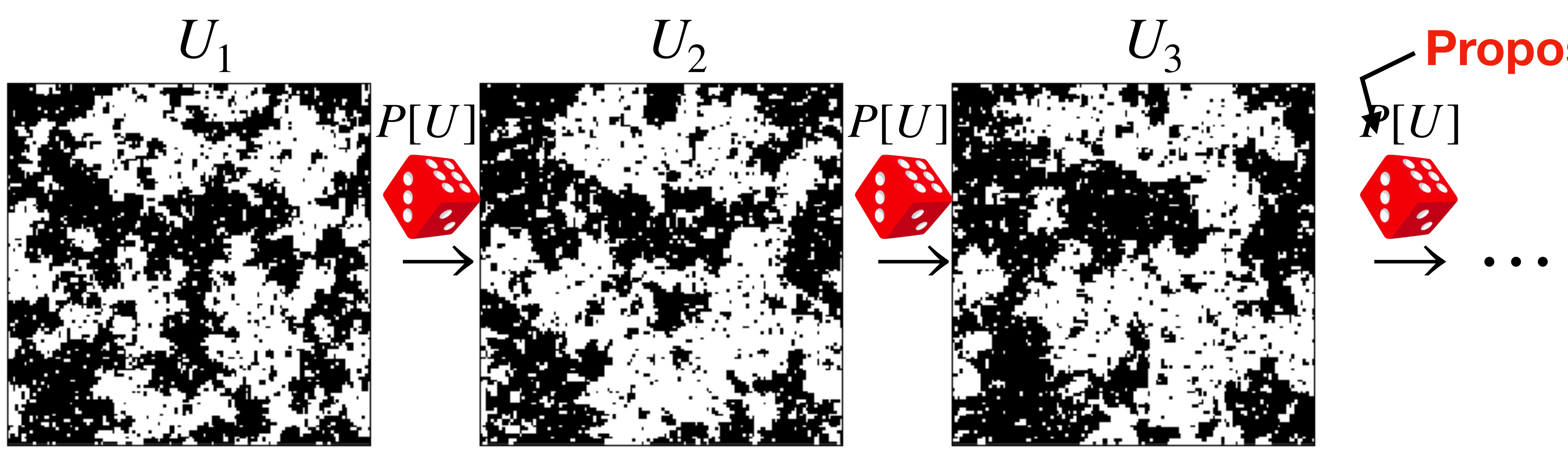
M. Creutz 1980

Target integration
= expectation value $\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}U e^{-S_{\text{eff}}[U]} \mathcal{O}(U)$

$S_{\text{eff}}[U] = S_{\text{gauge}}[U] - \log \det(\mathcal{D}[U] + m)$

Monte-Carlo: Generate field configurations with "$P[U] \propto e^{-S_{\text{eff}}[U]}$". It gives expectation value

Markov-Chain



$$\langle \mathcal{O} \rangle \approx \frac{1}{N_{\text{sample}}} \sum_{k=1}^{N_{\text{sample}}} \mathcal{O}[U_k]$$

Production with is numerically expensive
and how can we accelerate it? We use machine learning!

Introduction

Use of symmetry is crucial

Symmetries are essential for theoretical physics.
This is actually true as well in machine learning.
Equivariance/Covariance of symmetries helps generalization, and avoiding wrong extrapolation
(Symmetry restricts the function form)

Example in ML:

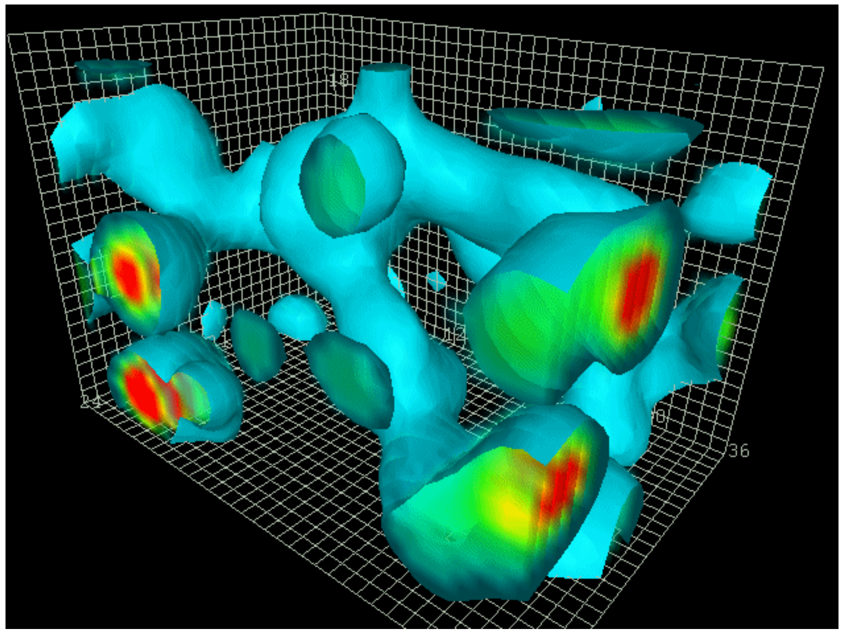
If data is translationally symmetric like photo images,
the frame work should respect this and one should implement with this translational symmetry in a neural network
= Convolutional neural net!

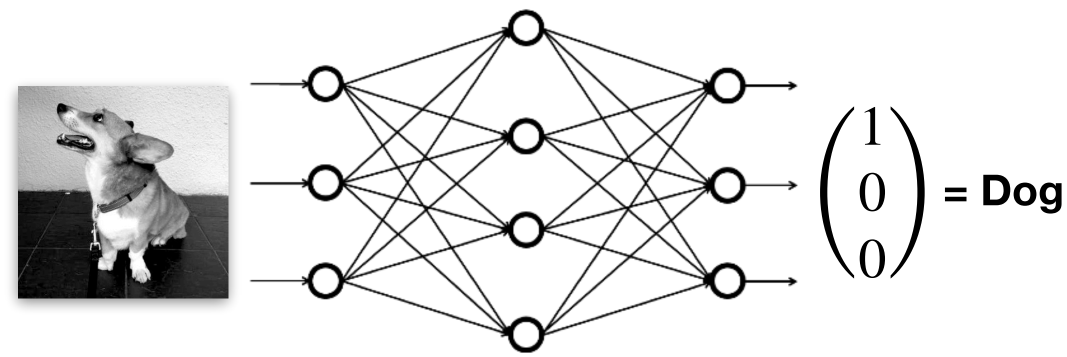
In physics + Machine learning,
= Physics embedded neural networks

We use symmetry in the system
as much as we can

Introduction

What is our final goal for QCD + Machine learning?





What we want to solve using machine learning?

- Reduction of numerical cost to beyond our current numerical limitations
  - Production and measurements
  - Use of machine learning may be useful

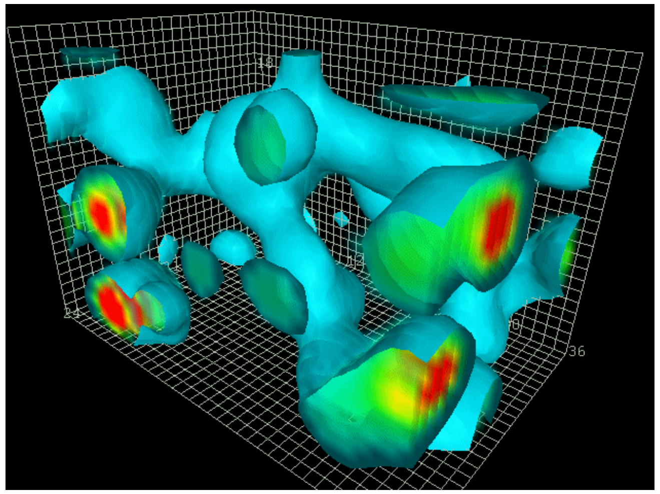
Restrictions (problems) to use ML:

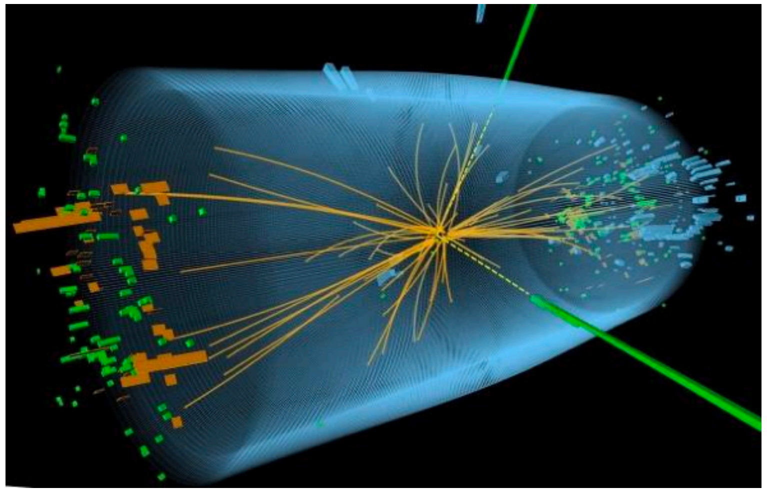
- Exactness & quantitative. Machine learning is an approximator
- **Gauge symmetry**, global symmetry is essential. While ML is not for physics
- Code. How can we make neural nets w/ HPC? (not showing in this talk)

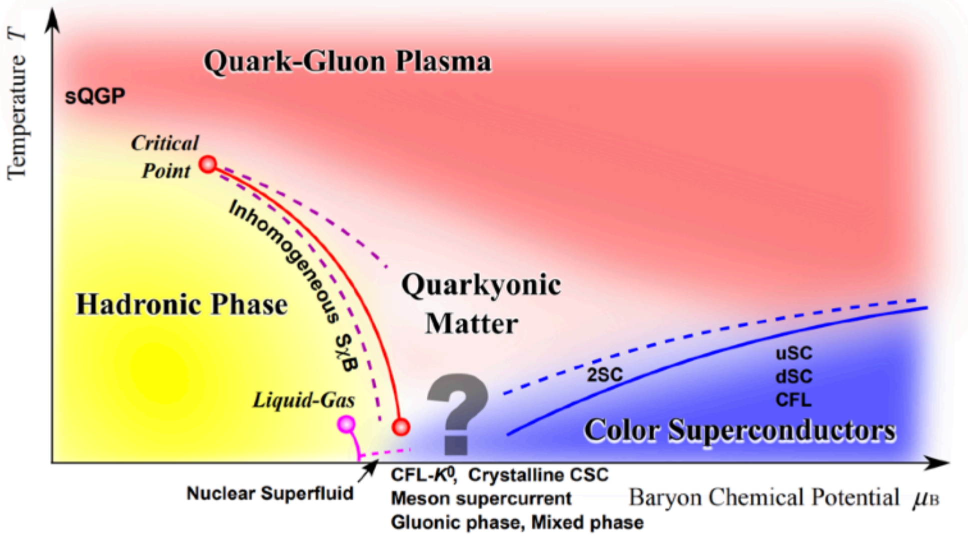
Introduction

What is our final goal for our research field?

Fukushima , Hatsuda
Rept.Prog.Phys.74:014001,2011







In short, we simulate of elementary particles in nuclei

Using super computers + Lattice QCD, we can understand…
- melting of protons/neutrons etc. at high temperatures
- attractive/repulsive forces between atomic nuclei
- candidate properties of dark matter

etc.

Intro: Lattice QCD& Monte-Carlo

Akio Tomiya

Numerical integral (via trapezoidal type) is impossible

$$S = \int d^4x \left[+ \frac{1}{2} \mathrm{tr}\, F_{\mu\nu} F_{\mu\nu} + \bar{\psi}\left(\partial\!\!\!/ - \mathrm{i}g A\!\!\!/ + m\right)\psi \right]$$

Lattice regularization

$$S[U, \psi, \bar{\psi}] = a^4 \sum_n \left[- \frac{1}{g^2} \mathrm{Re}\ \mathrm{tr}\ U_{\mu\nu} + \bar{\psi}\left(D\!\!\!/ + m\right)\psi \right]$$

a is lattice spacing (cutoff)

They are ``same" up to irreverent operators

$$\mathrm{Re}\ U_{\mu\nu} \sim \frac{-1}{2} g^2 a^4 F_{\mu\nu}^2 + O(a^6)$$

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}U \mathcal{D}\bar{\psi} \mathcal{D}\psi e^{-S} \mathcal{O}(U) = \frac{1}{Z} \int \mathcal{D}U e^{-S_{\mathrm{gauge}}[U]} \det(D + m) \mathcal{O}(U)$$

$$= \frac{1}{Z} \int \underline{\mathcal{D}U} e^{-S_{\mathrm{eff}}[U]} \mathcal{O}(U)$$

$$= \prod_{n \in \{\mathbb{Z}/L\}^4} \prod_{\mu=1}^{4} dU_\mu(n)$$

>1000 dim, no hope with
trapezoidal type numerical Integration -> use (Markov-chain) Monte Carlo

Flow based sampling algorithm

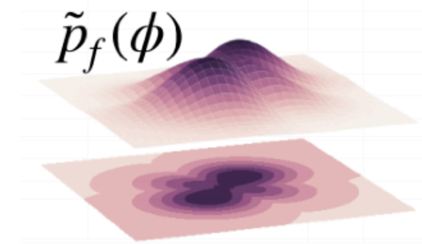
Akio Tomiya

Trivialization is attractive

Joint dist.

QFT probability:
Propagating modes
~ correlations

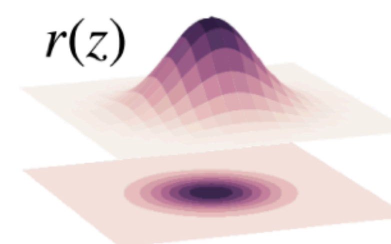
$$P[\phi] = \frac{1}{Z} \mathrm{e}^{-S[\phi]} = P(\phi_1, \phi_2, \cdots, \phi_{L^4})$$



Can we find a change of variable?

Trivial distribution
Trivial theory
No propagation, factorized
(Not the Gaussian FP)

$$P^{\mathrm{tri}}[z] = r(z_1) r(z_2) \cdots r(z_{L^4})$$



$r(z_i)$ probability for 1 variable
Easy to sample

- Correlations in $P[\phi]$ makes theory non-trivial and it makes MCMC harder.
- $P^{\mathrm{tri}}[z] = r(z_1) r(z_2) \cdots r(z_{L^4})$ has no correlation, sampling is trivial.
- Actually, there is a map between them. Trivializing map!
  - We can trivialize the target theory

Famous example: Nicolai map in SUSY. **Change of variable makes theory bilinear (~trivial)**. How about for non-SUSY?

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

Nicolai, H., Phys. Lett. 89B, 341 (1980); Nucl. Phys. B176, 419 (1980).

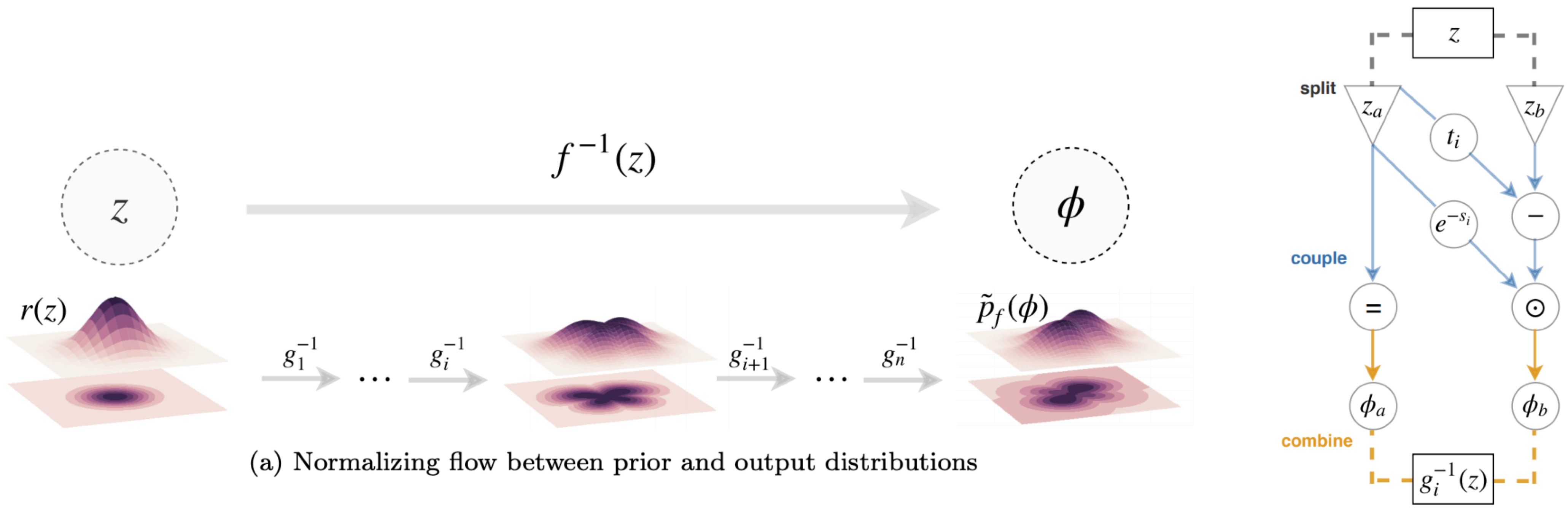
Related works

Akio Tomiya

Flow based algorithm = neural net represented flow algorithm

Real scalar in 2 dimension

MIT + DeepMind 2019~



(a) Normalizing flow between prior and output distributions

(b) Inverse coupling layer

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

Their sampling strategy

sample gaussian → inverse trivializing map → QFT configurations
Calculate Jacobian
After sampling, Metropolice-Hasting test (Detailed balance)→ exact!

arxiv 1904.12072, 2003.06413, 2008.05456