格子QCDとその周辺における機械学習の活用

富谷昭夫 (IPUT Osaka)

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

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 it.

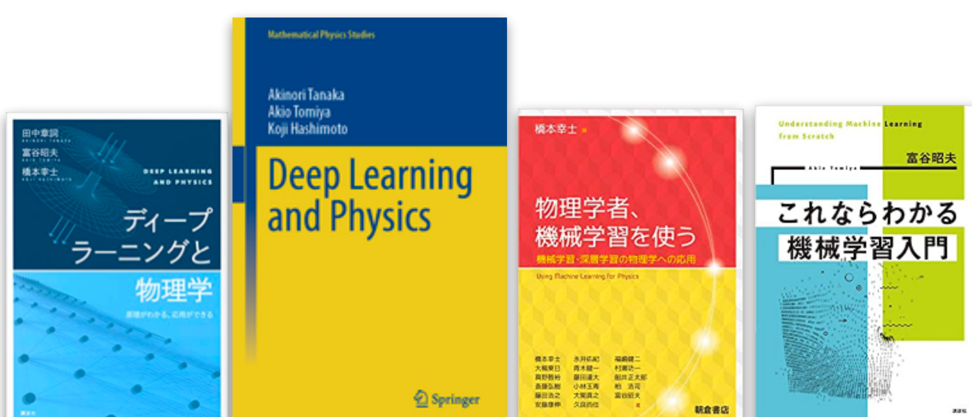
My papers https://scholar.google.co.jp/citations?user=LKVqy_wAAAAJ

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

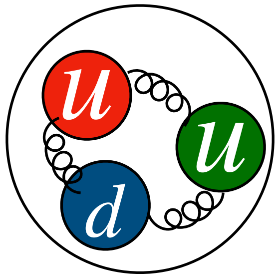
+quantum computer

Others:

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

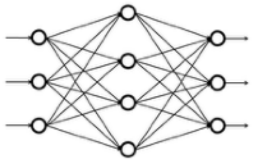
Outline of my talk

Lattice QCD?



Problem and Goal

Transformer for Physics

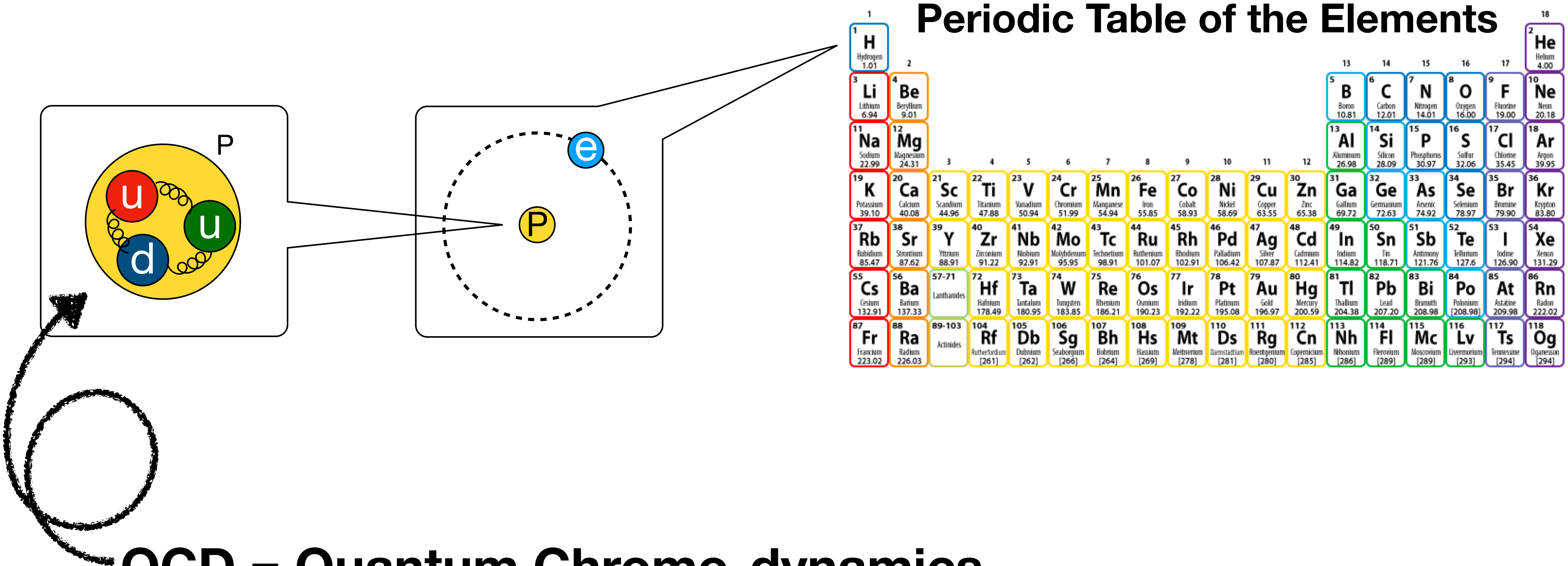


What is Lattice QCD?

Introduction

What is QCD?

Akio Tomiya



QCD = Quantum Chromo-dynamics

= A fundamental theory for particles inside of nuclei

Quantum many body, relativistic, strongly correlated

Introduction

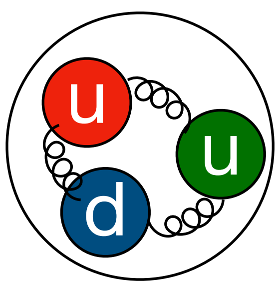
Lattice QCD = QCD on discretized spacetime = calculable

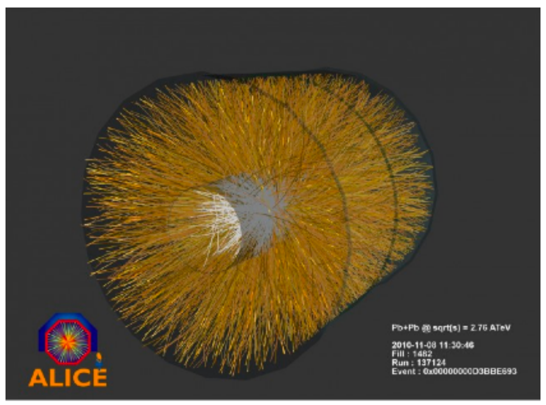
QCD (Quantum Chromo-dynamics) in 3 + 1 dimension

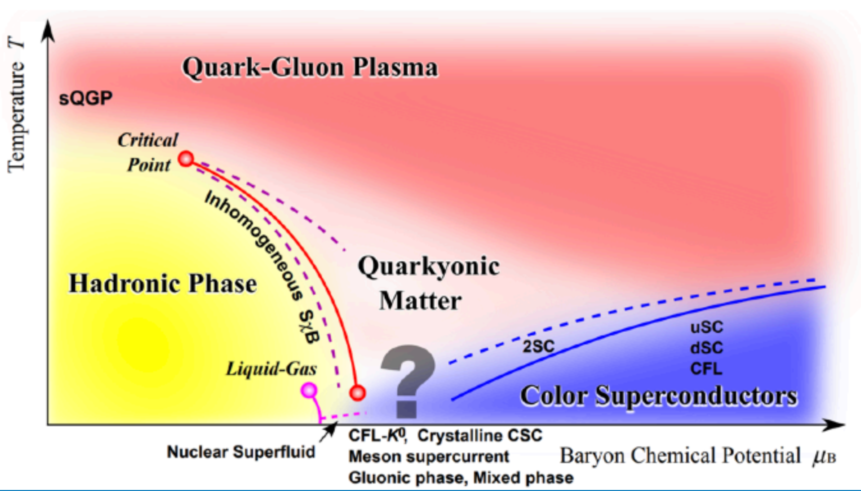
$$S = \int d^4x \left[-\frac{1}{2} \mathrm{tr}\, F_{\mu\nu}F^{\mu\nu} + \bar{\psi}\left(i\partial\!\!\!/ + gA\!\!\!/ - m\right)\psi \right]$$

$$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu - \mathrm{i}g[A_\mu, A_\nu]$$

Non-commutable version of (quantum) electro-magnetism







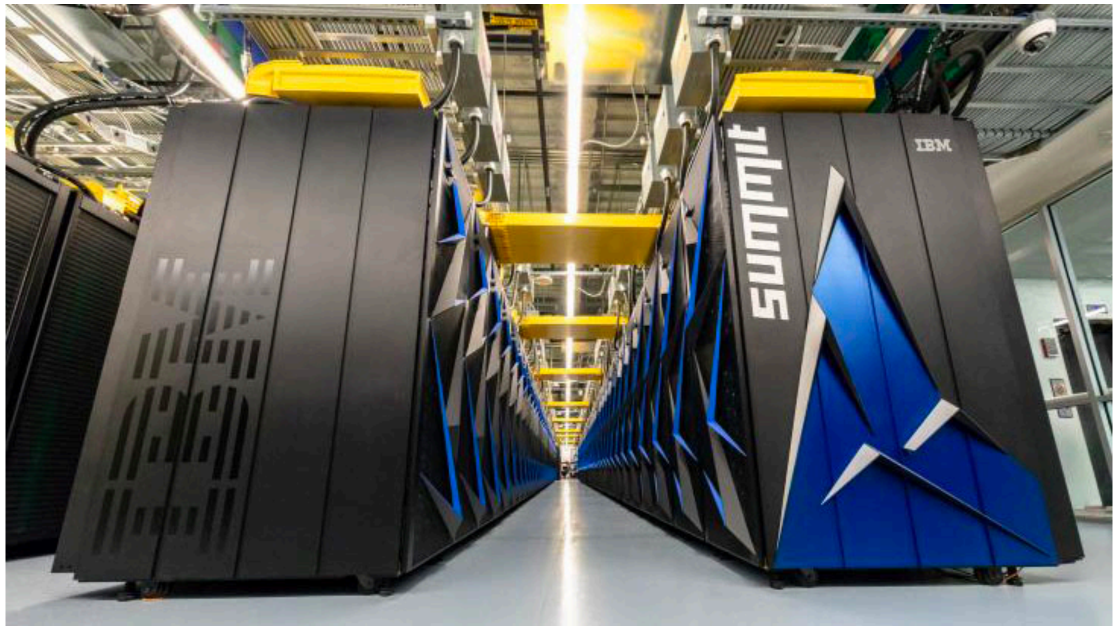
- This describes inside of nuclei& mass of hadrons, equations of states etc
- If we discretized the system, it becomes like spin-glass + fermions system
- **We want to evaluate expectation values with following integral,**

$$\langle O \rangle \sim \int \mathcal{D}A\mathcal{D}\bar{\psi}\mathcal{D}\psi e^{\mathrm{i}S} O$$

- We can use Markov Chain Monte-Carlo

物理の道具, 既存手法の問題点

格子QCDの計算はスパコンで！(1980年~)



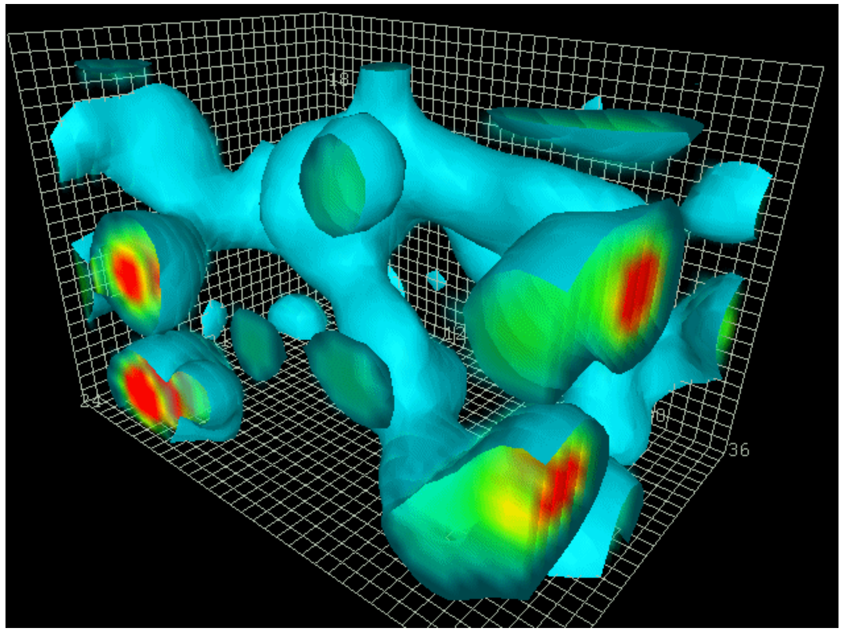
スパコンで計算して何がわかるの？

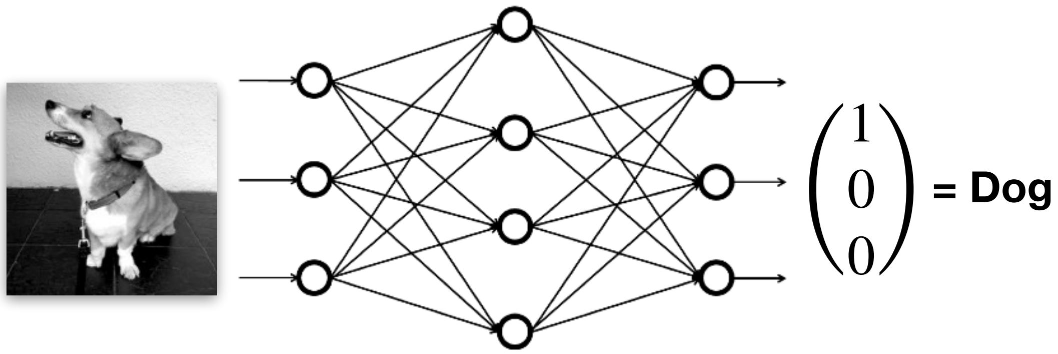
- 陽子/中性子の仲間の質量 (前述の通り)
- 原子核同士の引力/斥力の様子 (星の生まれて死ぬまでを理解するのに必要)
- 高温での陽子/中性子等の溶解の様子 (宇宙の歴史に関わる)
- ダークマターの候補の性質(実験で見つけるには性質を知っておく必要あり)
- 陽子/中性子内のクォークの様子
- 手計算で計算できない各種係数(素粒子の標準理論が実験と整合性チェックに必要)

などなど...

Introduction

What is our final goal for our research field?





<u>What we want to solve?</u>

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

<u>Restrictions (problems) to use ML:</u>

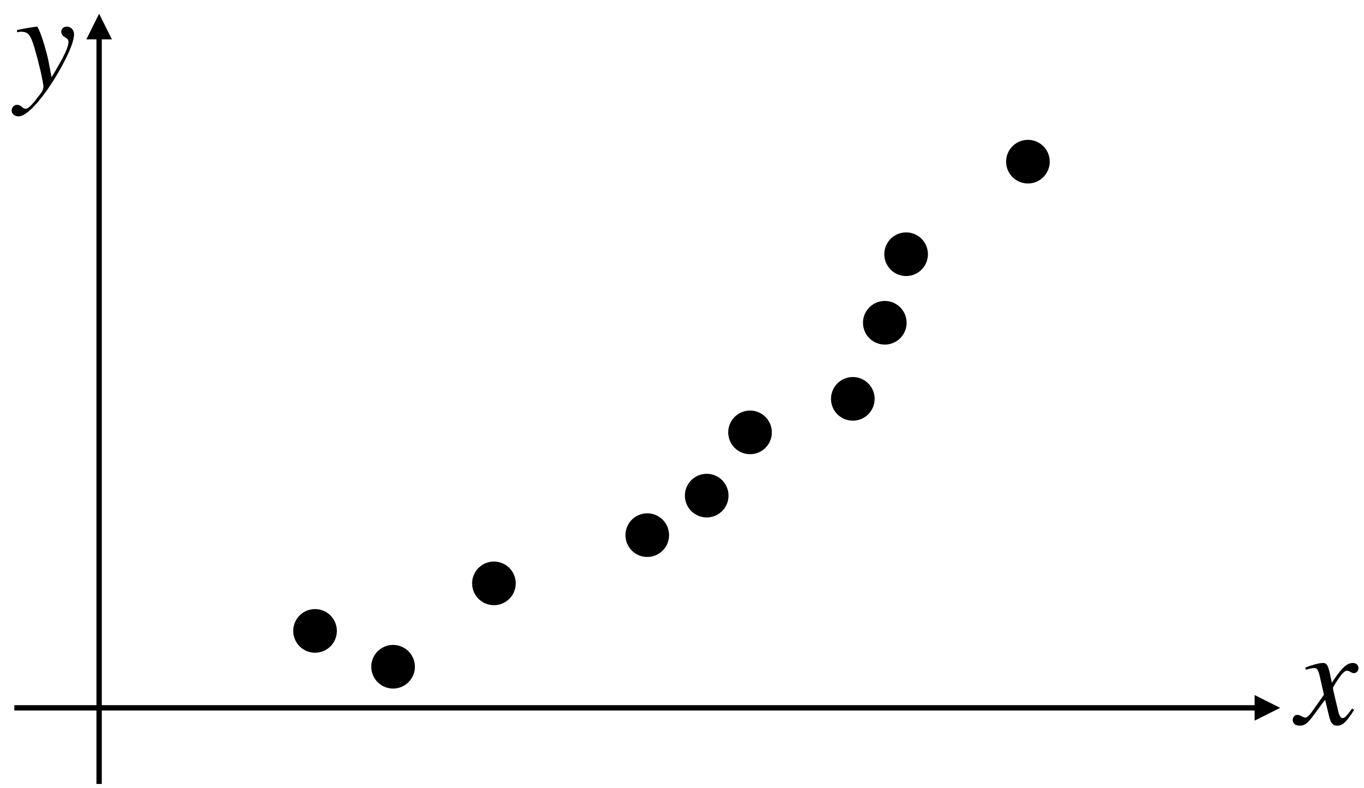
- Exactness & quantitative. Machine learning is an approximator
- **Gauge symmetry**, global symmetry is essential. While ML is not for physics
- Code. How can we make neural nets w/ HPC? (not showing in this talk)

Machine learning?

What is machine learning?

E.g. Linear regression ∈ Supervised learning

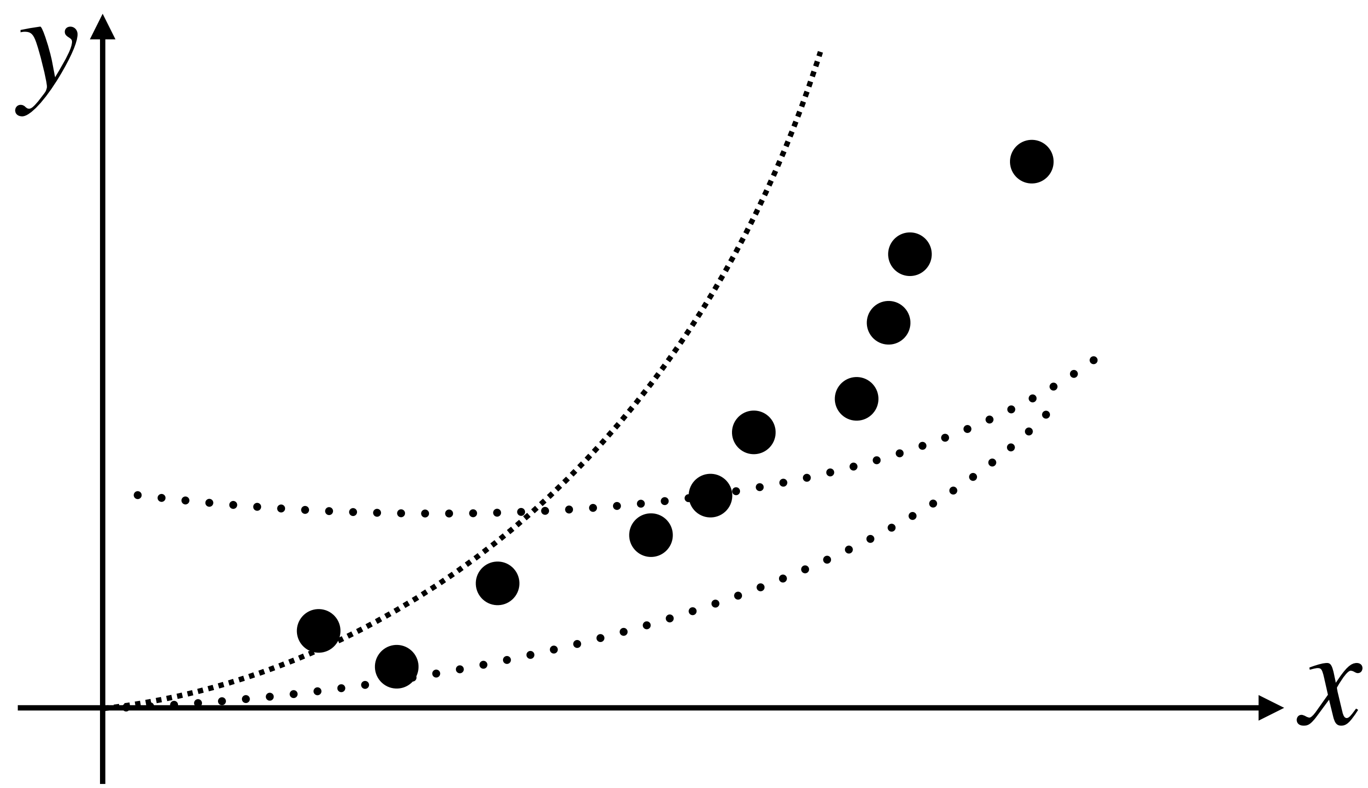
Data: $D = \left\{(x^{(1)}, y^{(1)}),\ (x^{(2)}, y^{(2)}),\ \cdots\right\}$



What is machine learning?

E.g. Linear regression $\in$ Supervised learning

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



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

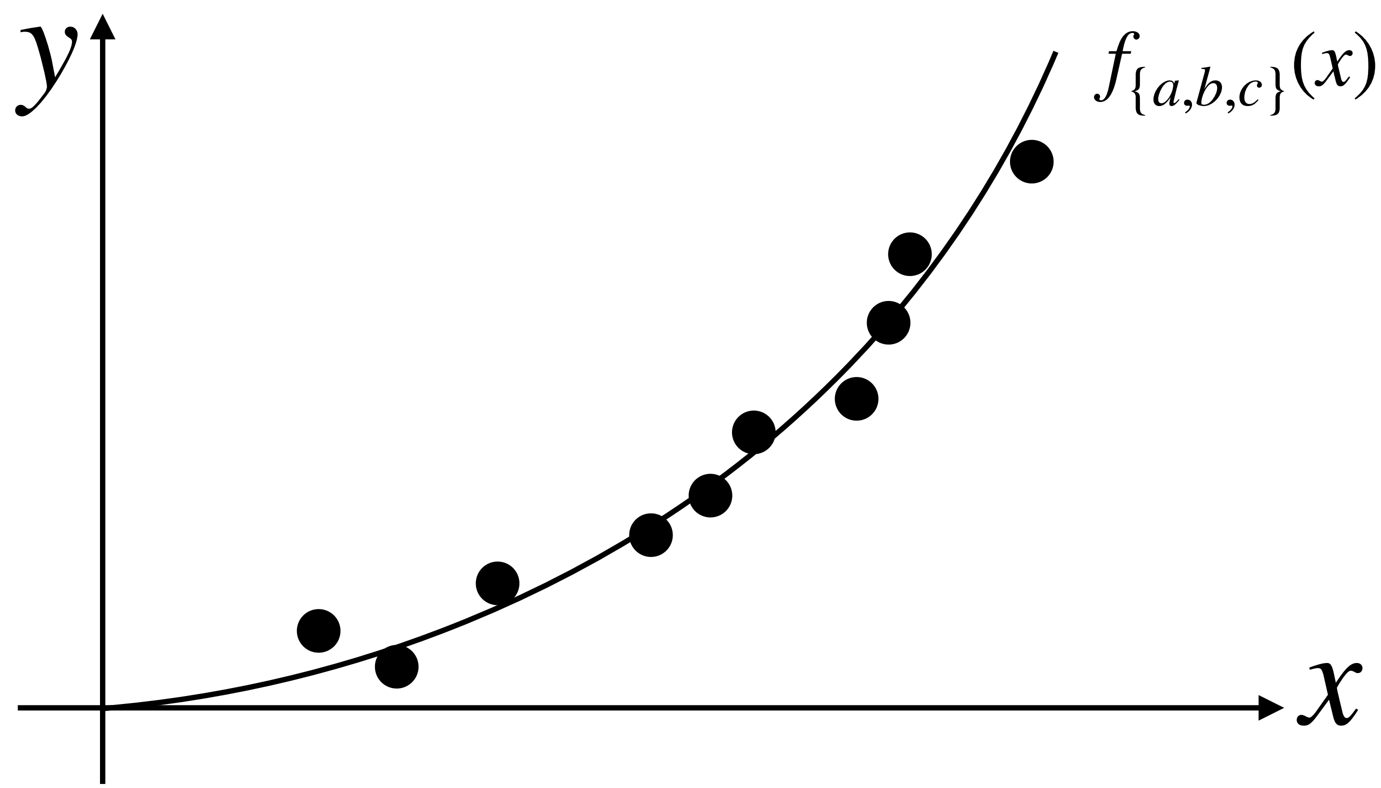
$$E = \frac{1}{2} \sum_d \left| f_{\{a,b,c\}}(x^{(d)}) - y^{(d)} \right|^2$$

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

What is machine learning?

E.g. Linear regression ∈ Supervised learning

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



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

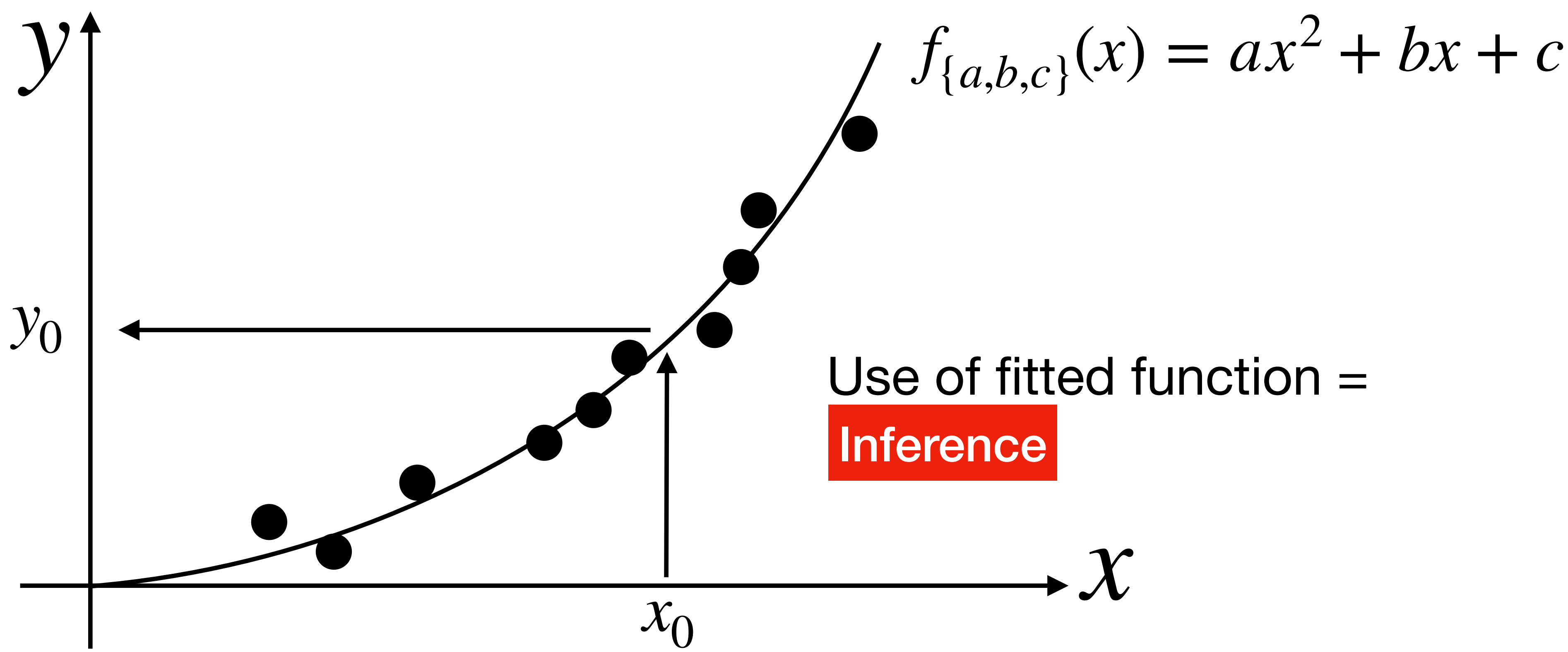
$f_{\{a,b,c\}}(x) = ax^2 + bx + c$ $\qquad E = \frac{1}{2} \sum_d \left| f_{\{a,b,c\}}(x^{(d)}) - y^{(d)} \right|^2$

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

What is machine learning?

E.g. Linear regression ∈ Supervised learning

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



Use of fitted function =
Inference

Now we can predict y value which not in the data

In physics language, variational method

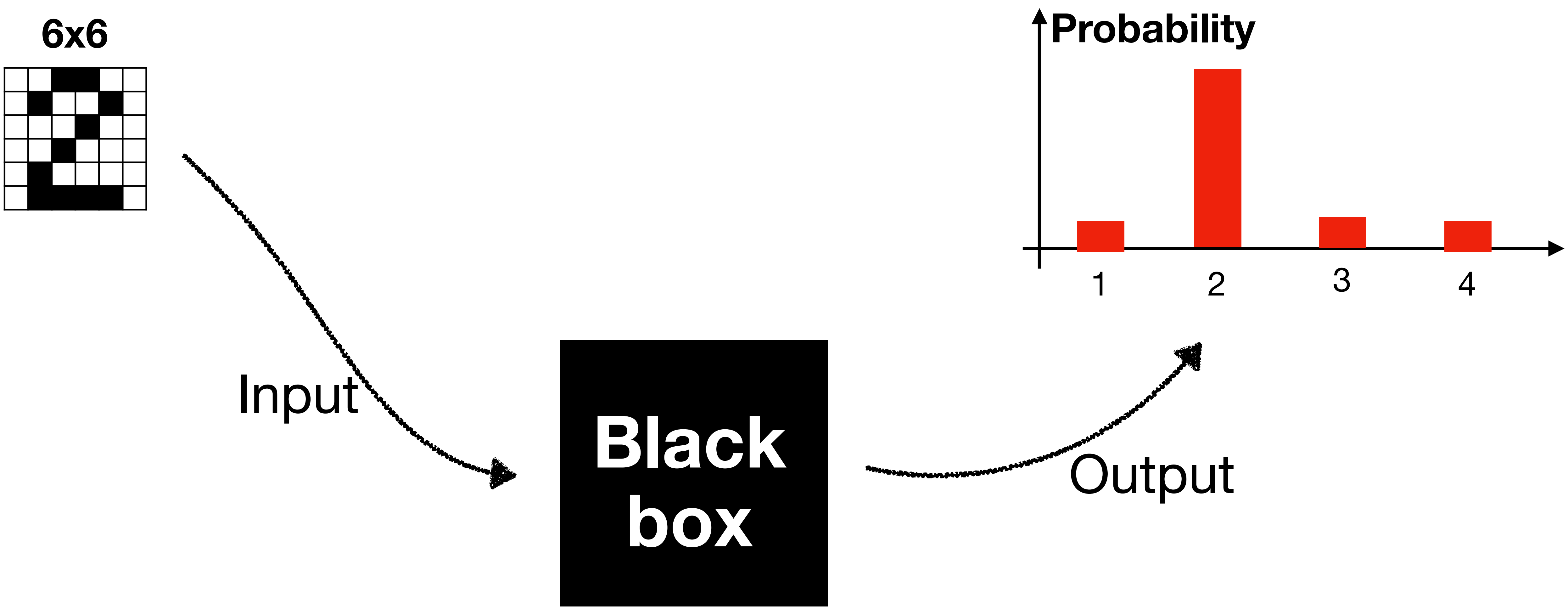
Akio Tomiya

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?

Akio Tomiya

Neural network is a *universal* approximation function

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

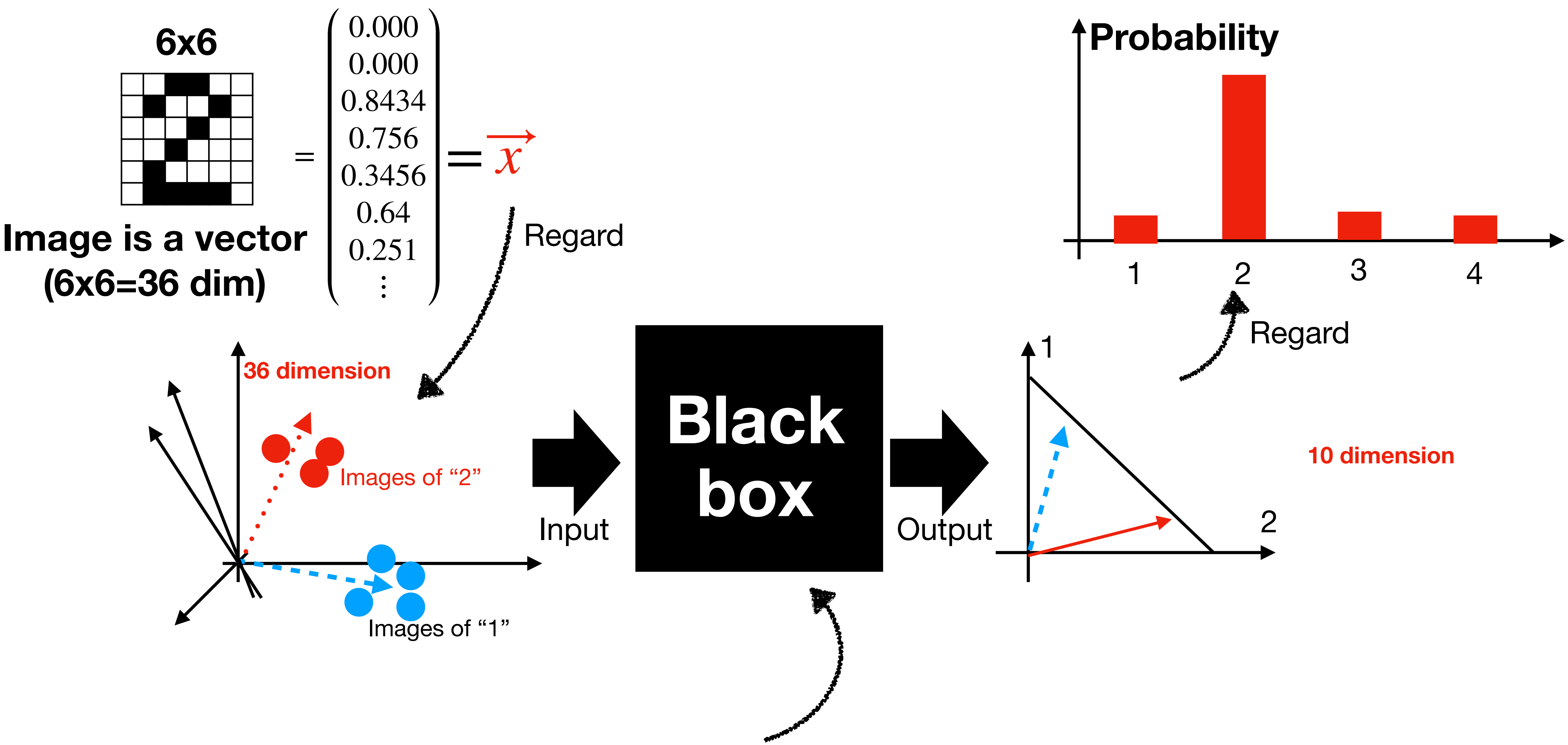


Image recognition = Find a map between two vector spaces

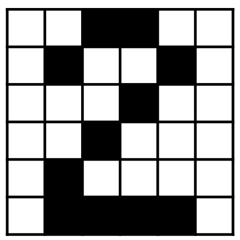
Akio Tomiya

What is the neural networks?

Neural network is a *universal* approximation function

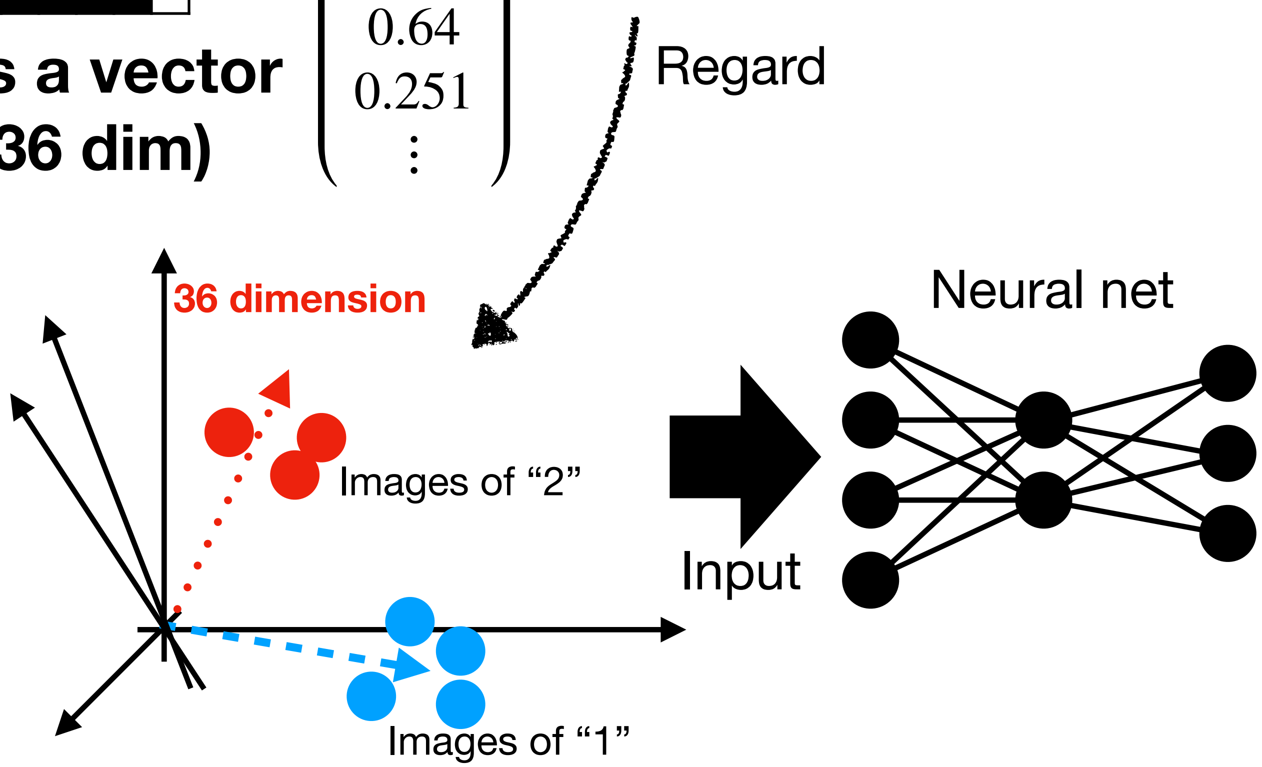
Example: Recognition of hand-written numbers (0-9)

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)



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:

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

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

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

Neural network = map between vectors and vectors

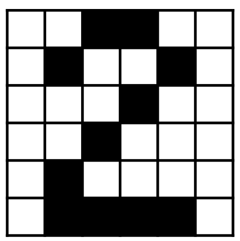
Physicists terminology: Variational ansatz

What is the neural networks?

Neural network is a *universal* approximation function

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

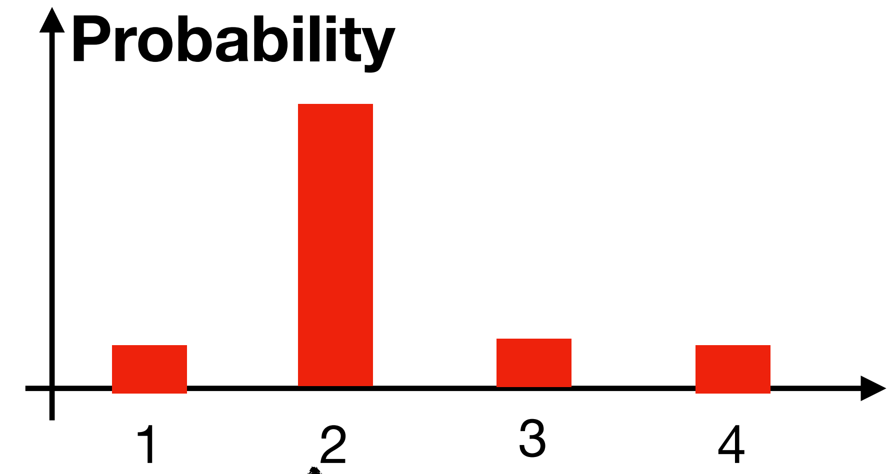
6x6



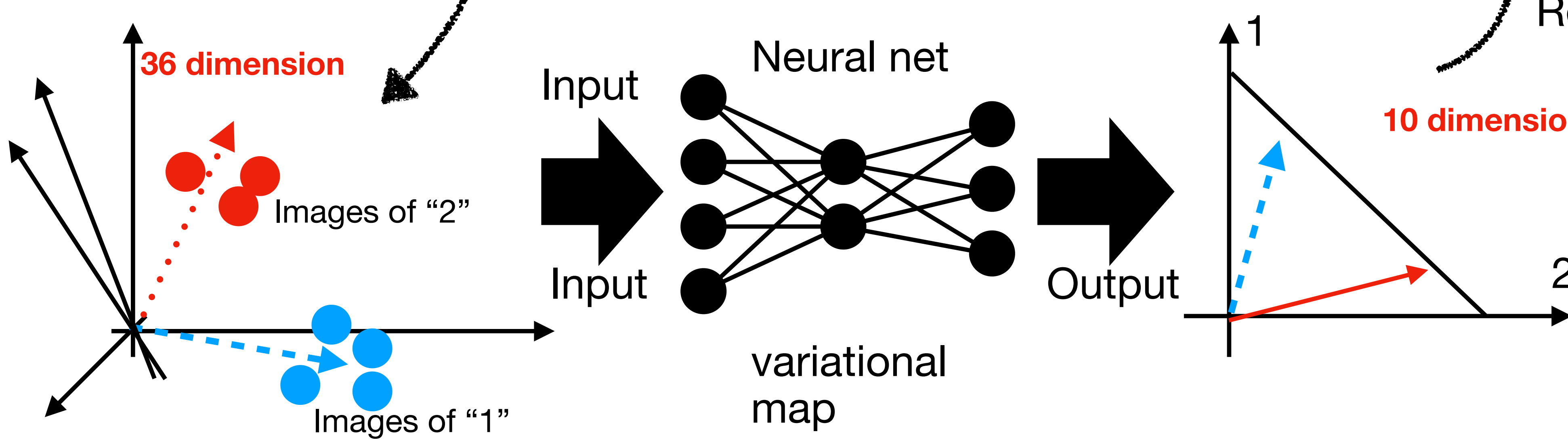
$$= \begin{pmatrix} 0.000 \\ 0.000 \\ 0.8434 \\ 0.756 \\ 0.3456 \\ 0.64 \\ 0.251 \\ \vdots \end{pmatrix} = \vec{x}$$

Image is a vector (6x6=36 dim)

Regarding



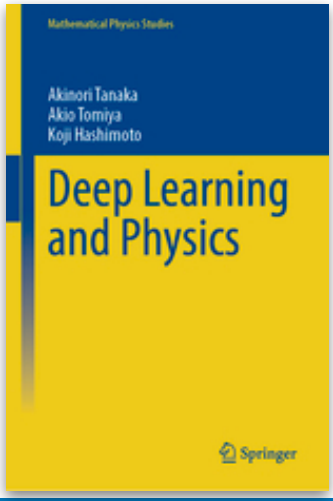
Regard



“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)

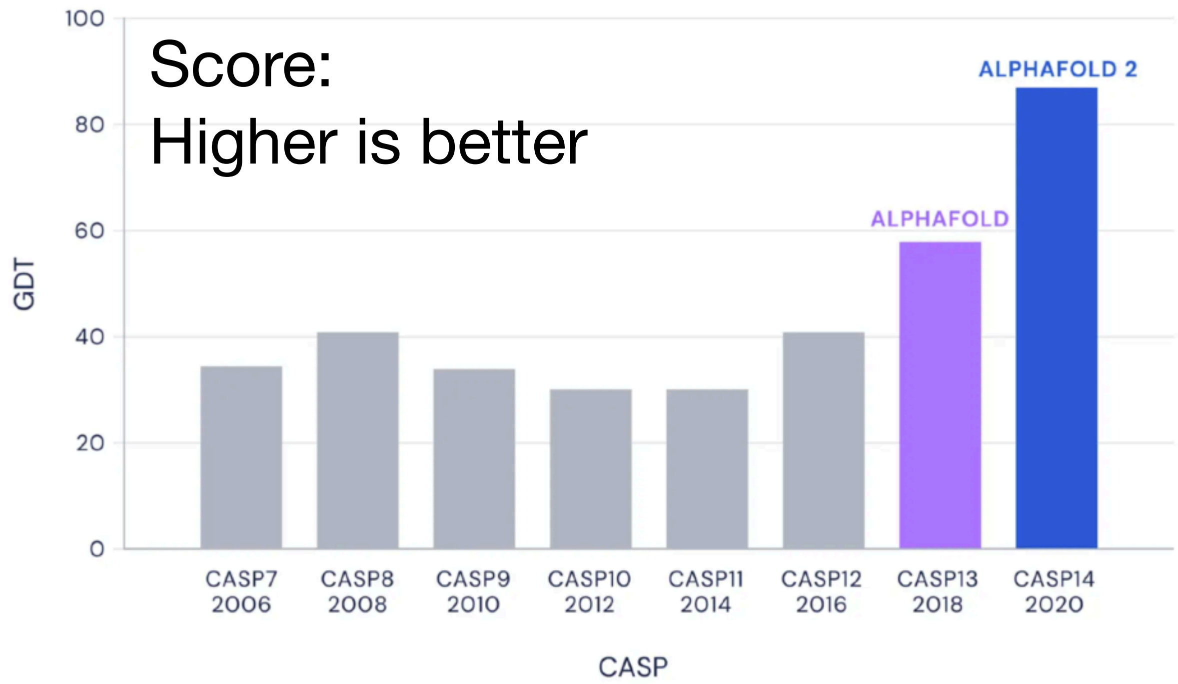


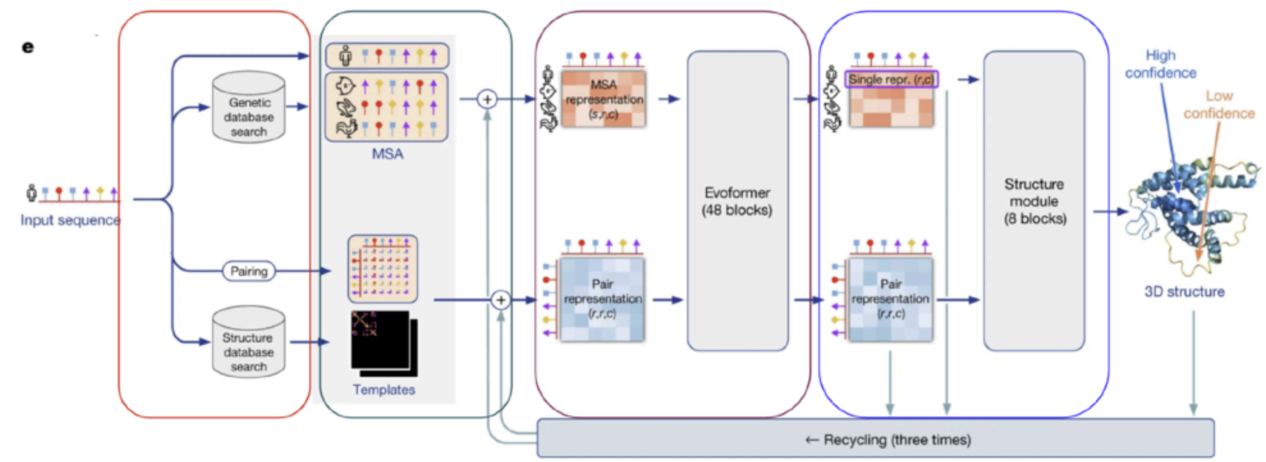
What is the neural networks?

Akio Tomiya

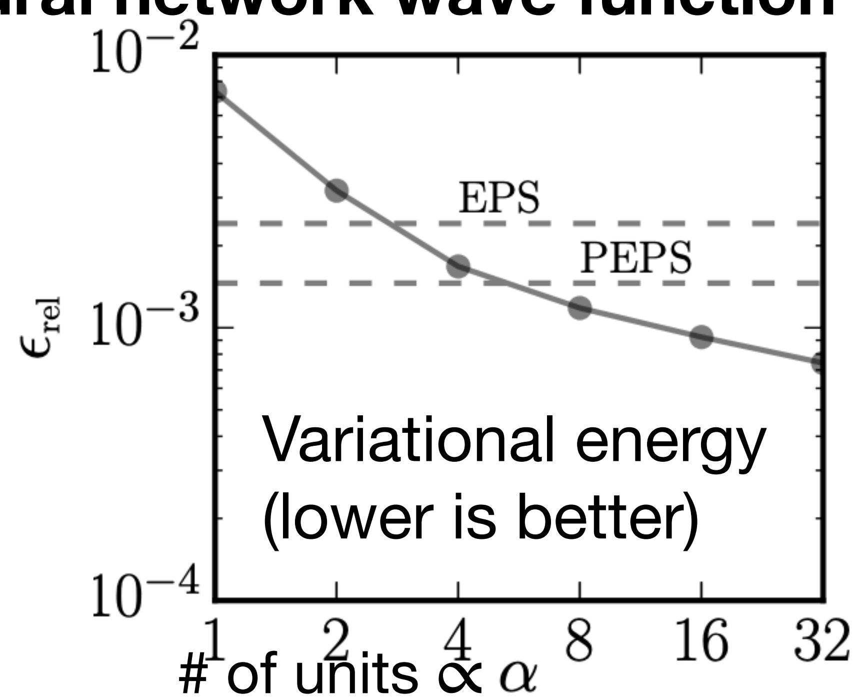
Neural network have been good job

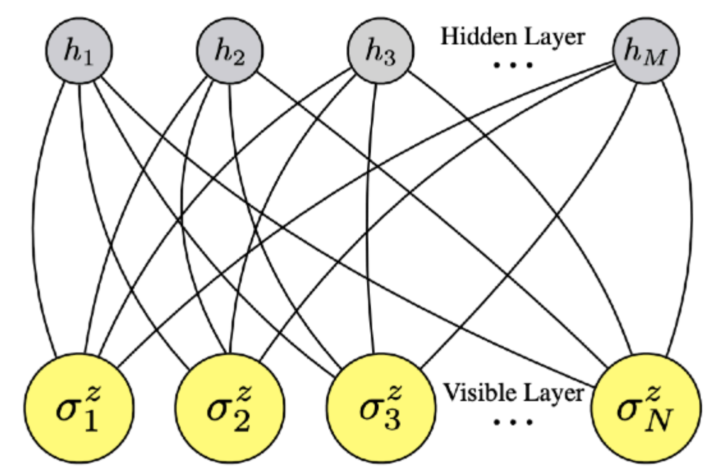
Protein Folding (AlphaFold2, John Jumper+, Nature, 2020+), Transformer neural net





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

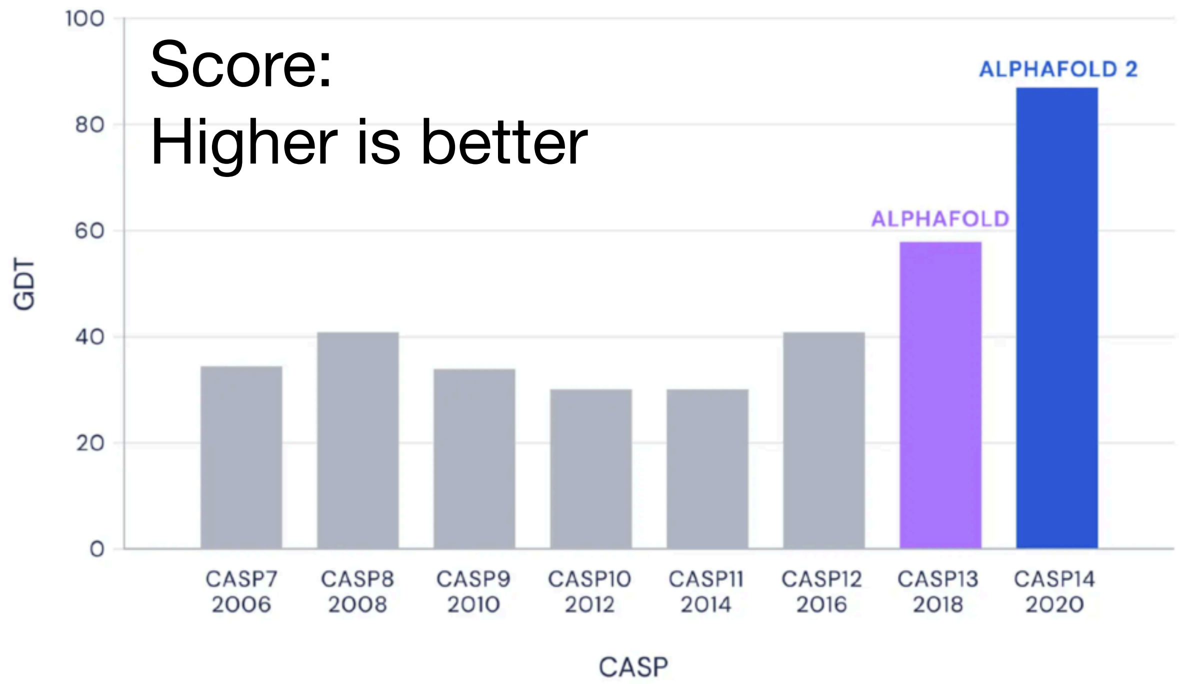
Problem and Goal

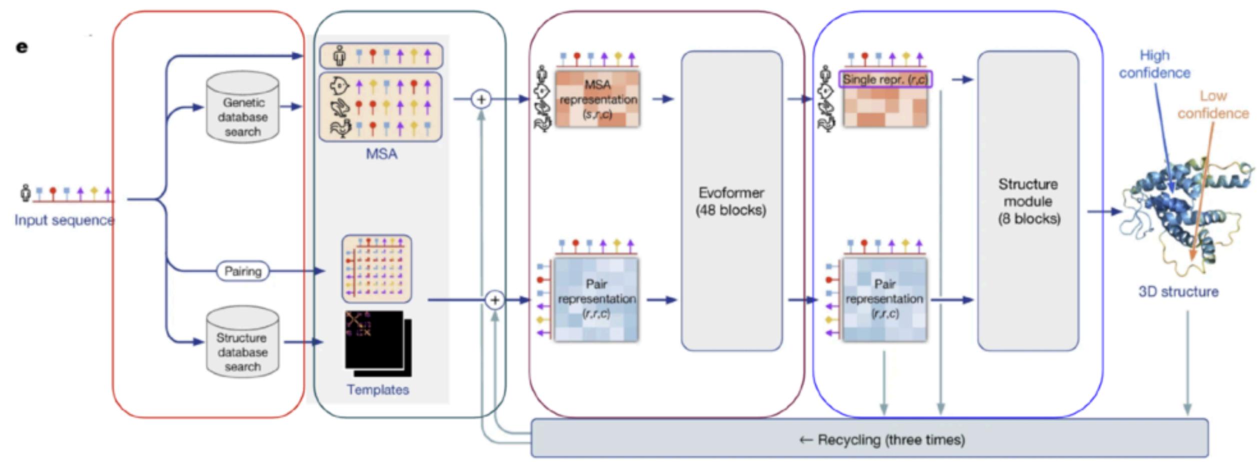
Akio Tomiya

Equivariance and convolution

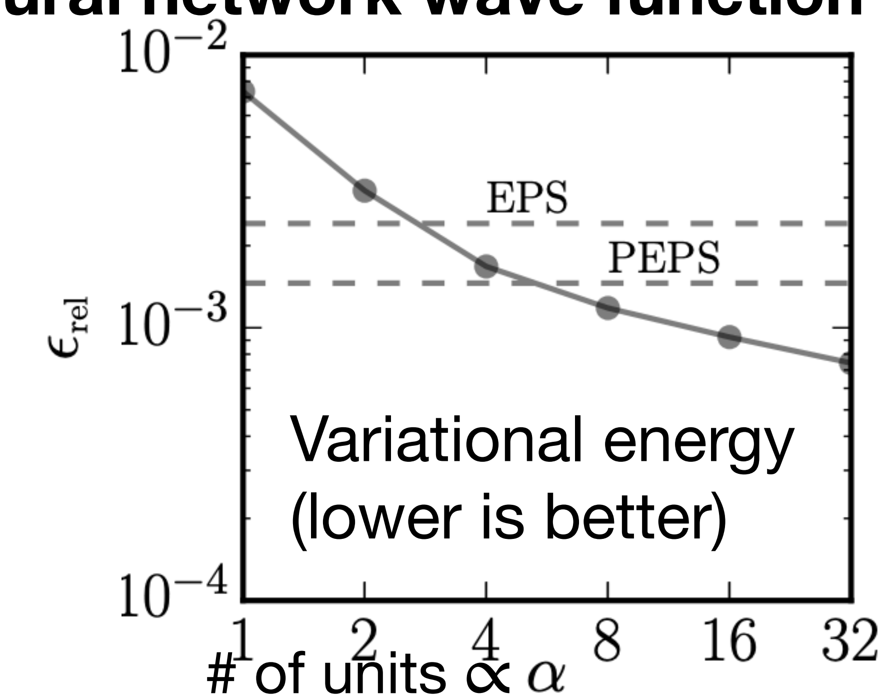
Neural network works quite well in natural science

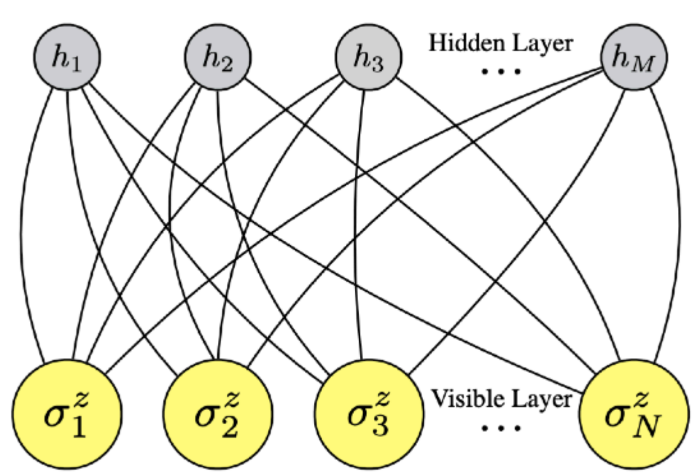
Protein Folding problem (AlphaFold2, John Jumper+, Nature, 2020+), Transformer





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





Neural net + "Expert knowledge" → Best performance

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

Motivation

Akio Tomiya

Monte-Carlo integration is available, but still expensive!

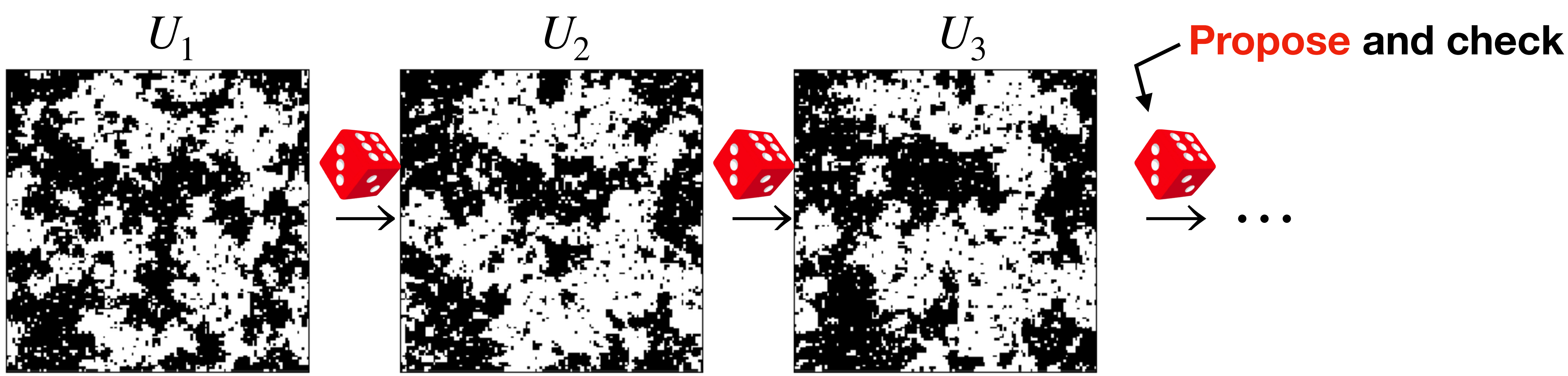
M. Creutz 1980

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

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

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

Markov-Chain



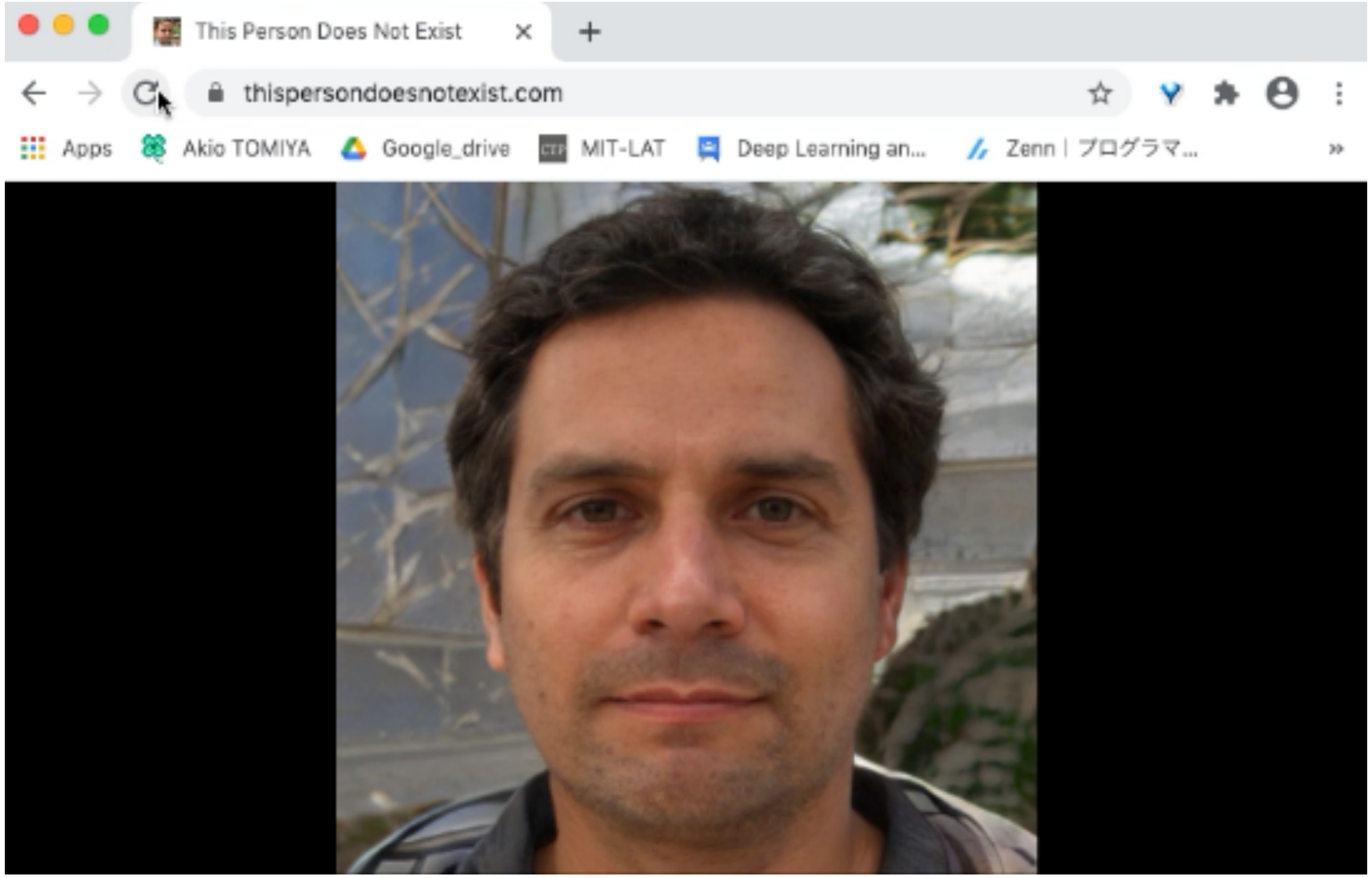
$$\langle \mathcal{O} \rangle \approx \frac{1}{N_{\text{sample}}} \sum_{k=1}^{N_{\text{sample}}} \mathcal{O}[U_k]$$

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

Introduction

Generative neural net can make human face images

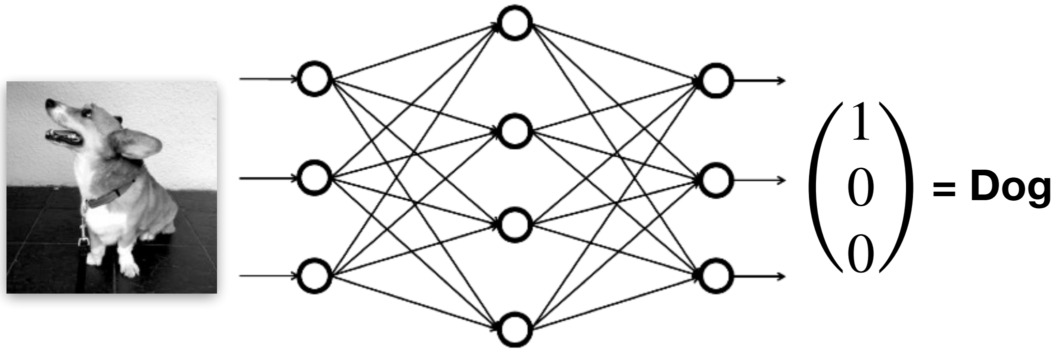
Neural nets can generate realistic human faces (Style GAN2)

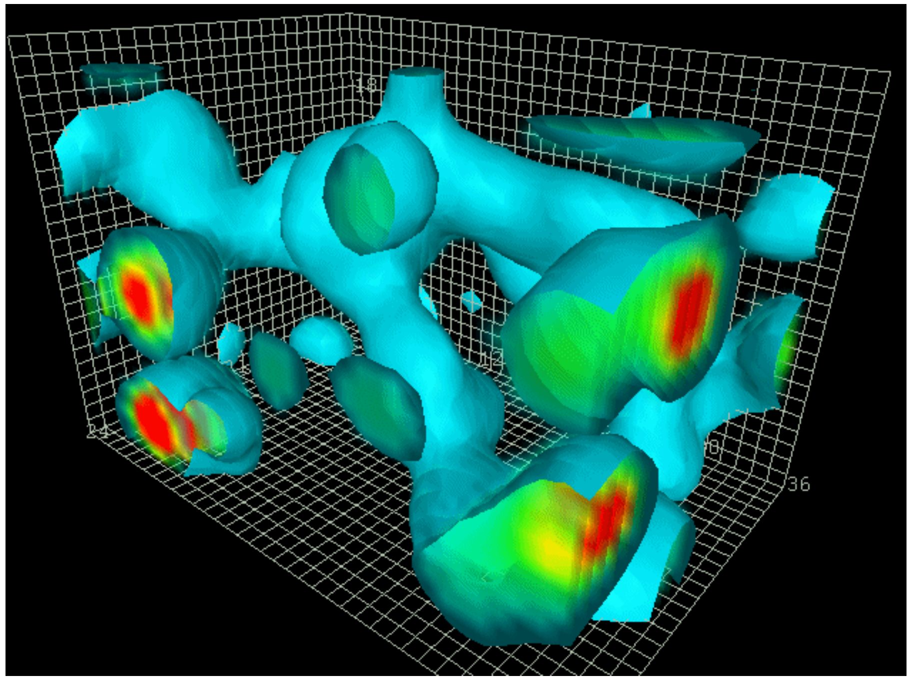


Realistic Images can be generated by machine learning!
Configurations as well? (proposals ~ images?)

Introduction
ML for LQCD is needed

- 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

Akio Tomiya

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

2 cases in lattice theory:

Configuration generation

1. Flow-based sampling
2. Transformer (Not gauge theory)

Flow-based sampling

Flow based sampling algorithm

Akio Tomiya

Change of variables makes problem easy

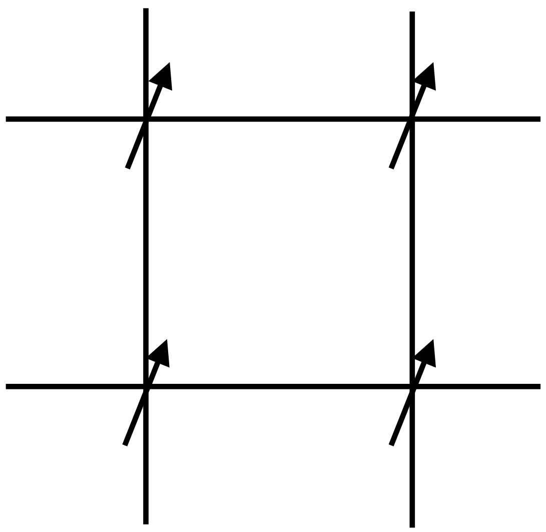
Ising model

$$\sum_{\{s\}} \phi e^{-\beta H[s]} O[s]$$

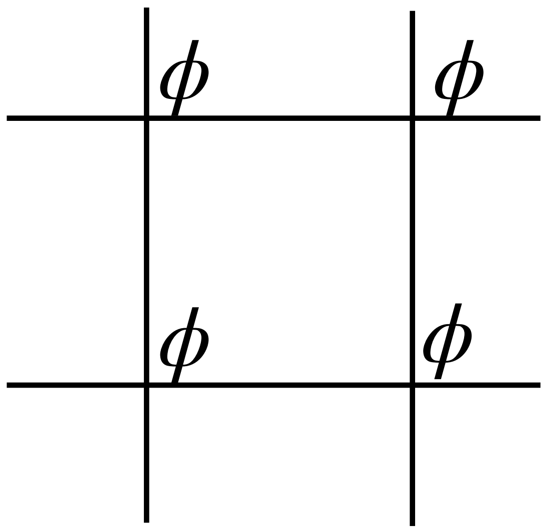
QFT

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

Ising Model



Ising Model



$$\phi_i \in \mathbb{R}$$

Energy function (Hamiltonian)

$$H = -J \sum s_i s_j$$

Energy function (Euclidean action)

$$S = -\sum_i [\sum_\mu \phi_i(\phi_{i+\mu} + \phi_{i-\mu} - 2\phi_i) + \phi_i^2]$$

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

Flow based sampling algorithm

Change of variables makes problem easy

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

We want this (Green's function)

Evaluation is hard
(1M dimension integration)

Back to high school,

- Integration by parts
- Change of variables

Are there any good "Change of variables" for QFT?

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

Flow based sampling algorithm

Akio Tomiya

Change of variables makes problem easy

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

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

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

If **this is easy to sample (or integrate), we are happy**

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

Flow based sampling algorithm

Akio Tomiya

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

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

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.

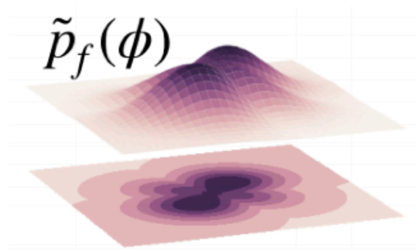
Flow based sampling algorithm

Akio Tomiya

Trivialization is attractive

QFT probability:
Propagating modes
~ correlations

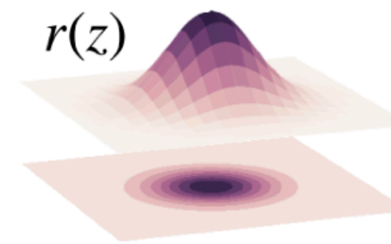
$$P[\phi] = \frac{1}{Z} e^{-S[\phi]} = P(\phi_1, \phi_2, \cdots, \phi_{L^4})$$



Can we find a change of variable?

Point-wise prob. dist.
Trivial theory
No propagation
(Not the Gaussian FP)

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



$r(z_i)$ probability for 1 variable

- Correlations in $P[\phi]$ makes theory non-trivial and it makes MCMC harder.
- $P^{\rm tri}[z] = r(z_1) r(z_2) \cdots r(z_{L^4})$ 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

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)

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!!

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

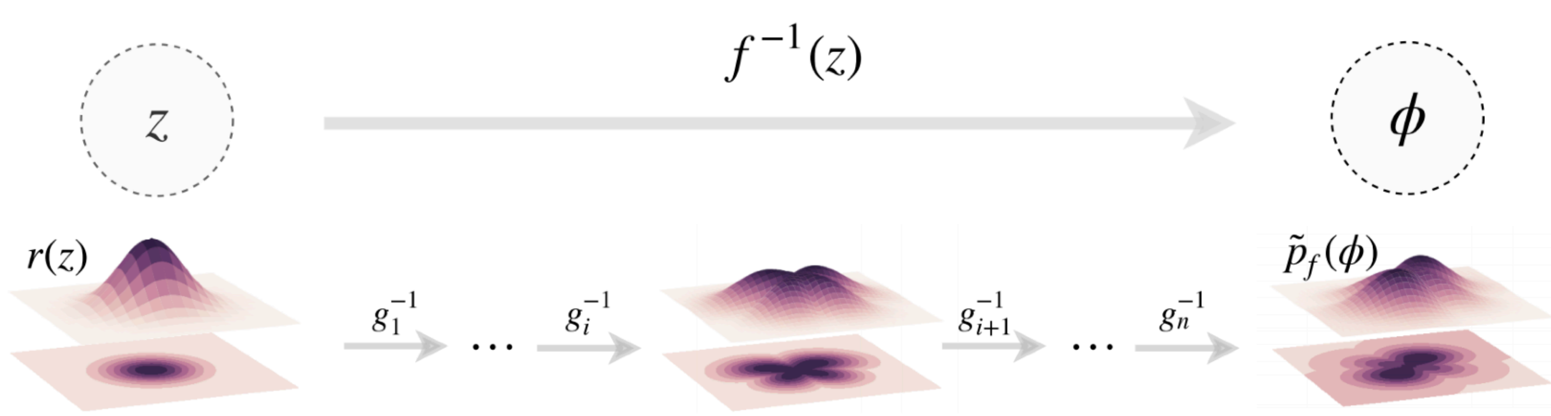
Related works

Akio Tomiya

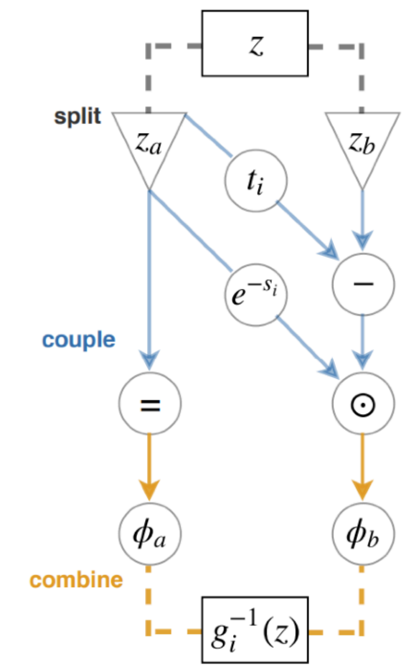
Flow based algorithm = neural net represented flow algorithm

Real scalar in 2 dimension

MIT + Google brain 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" $\tilde{\phi} = \mathscr{F}(\phi)$

If it is well represented, we can sample from a Gaussian

It can be done "Normalizing flow" (Real Non-volume preserving map)

Moreover, Jacobian is tractable!

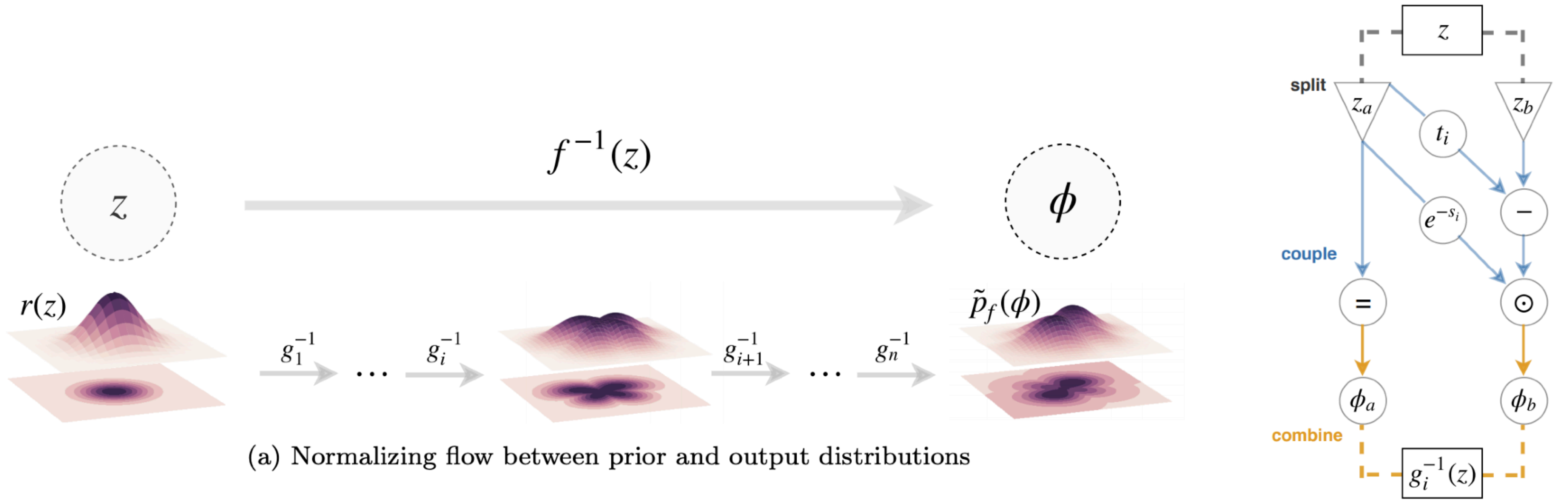
arxiv 1904.12072, 2003.06413, 2008.05456

Related works

Flow based algorithm = neural net represented flow algorithm

Real scalar in 2 dimension

MIT + Google brain 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

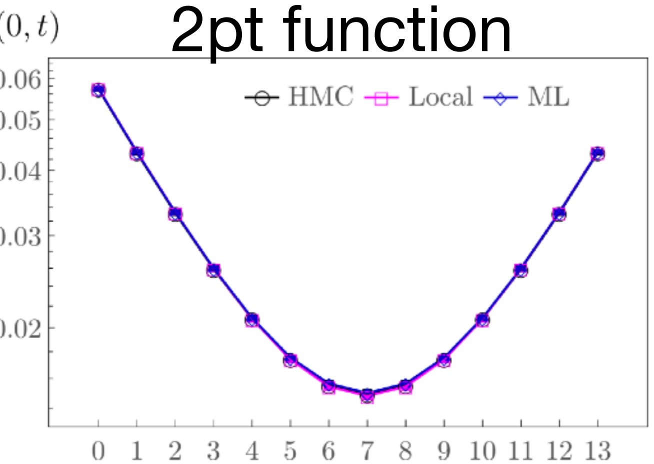
Related works

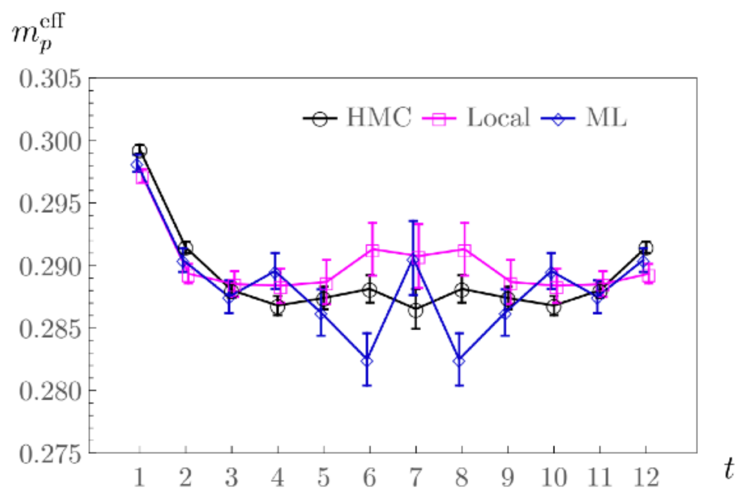
Flow based algorithm = neural net represented flow algorithm

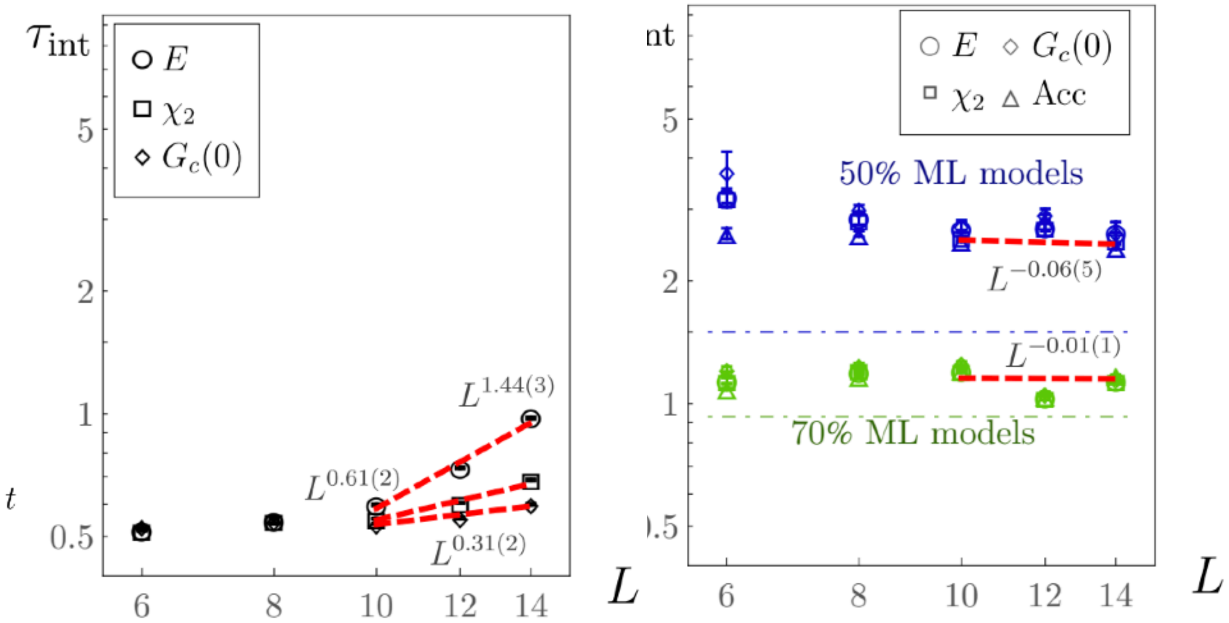
Real scalar in 2 dimension

MIT + Google brain 2019~

2pt function



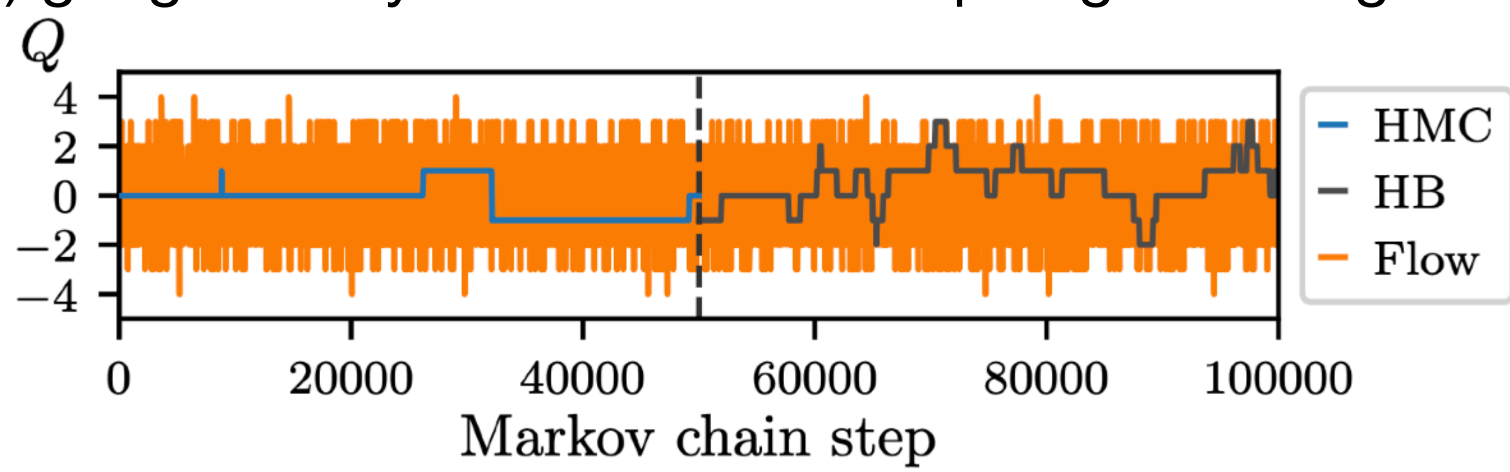




(a) HMC ensembles

(c) Flow-based MCMC ensembles

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



Applied already on SU(N)

4d? Fermions? -> OK

arxiv 1904.12072, 2003.06413, 2008.05456

Transformer for spin + fermion system as a test case for Lattice QCD

Transformer and Attention

Akio Tomiya

Attention layer used in Transformers (GPT, Bard)

arXiv:1706.03762

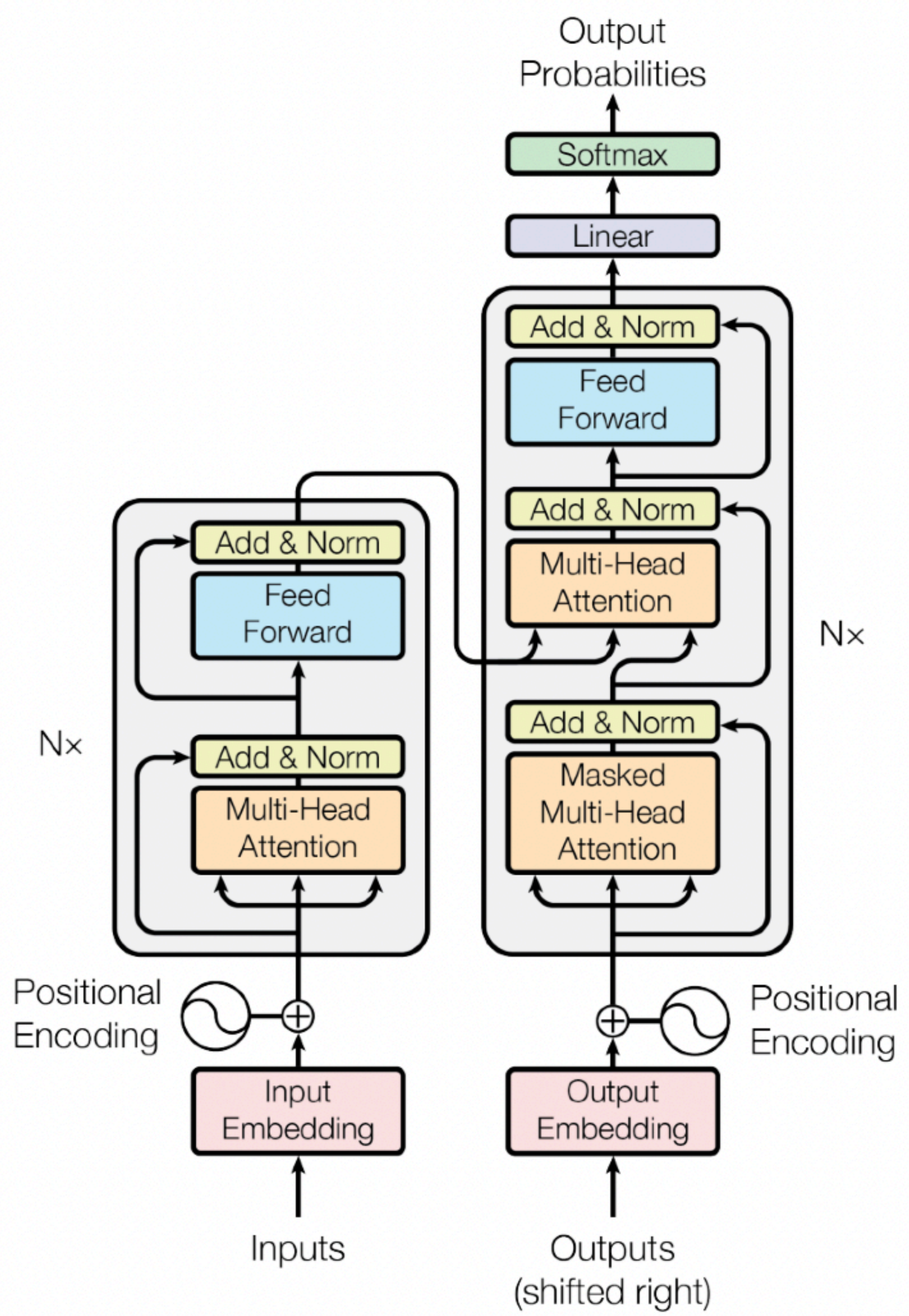


Figure 1: The Transformer - model architecture.

OpenAI ChatGPT

Bard

DeepL

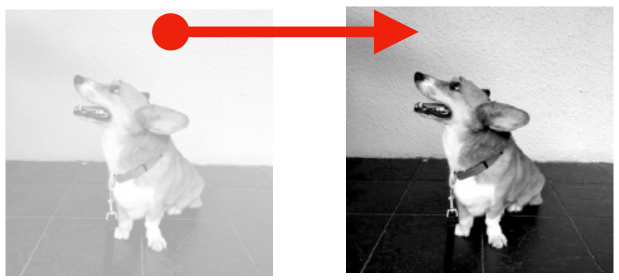
Attention layer (in transformer model) has been introduced in a paper titled
“Attention is all you need” (1706.03762)
State of the art architecture of language processing.
Attention layer is essential.

Equivariance and convolution

Akio Tomiya

Knowledge ∋ Convolution layer = trainable filter, Equivariant

Filter on image



shift to right

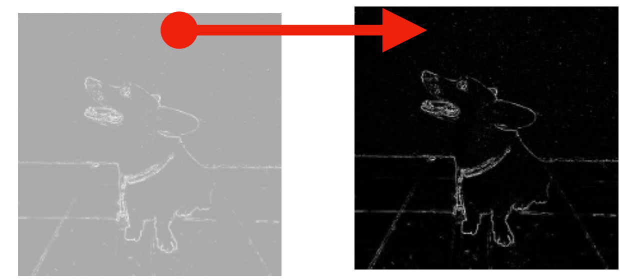
$*$

Laplacian filter

0	1	0
1	-2	1
0	1	0

(Discretization of ∂^2)

$=$

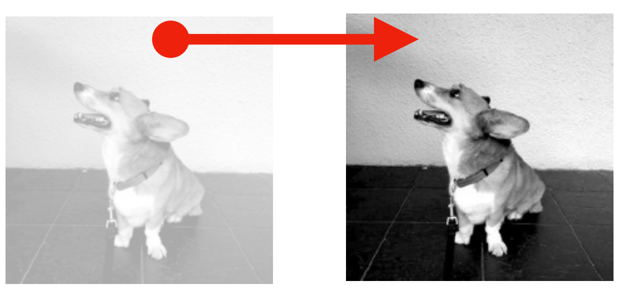


shift to right

Edge detection

Translational operation is *commutable* with filtering (equivariant)

Convolution layer



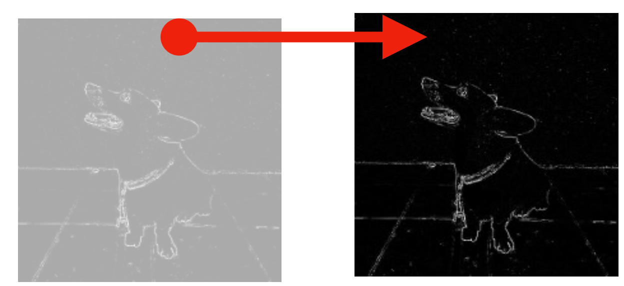
shift to right

$*$

Trainable filter

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

$=$



shift to right

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

Translational operation is *commutable* with convolutional neurons (equivariant)

This can be any filter which helps feature extraction (minimizing loss)

Equivariance reduces data demands. Ensuring symmetry (plausible Inference)

Many of convolution are needed to capture global structures

Transformer and Attention

Attention layer can capture non-local correlations

arXiv:1706.03762

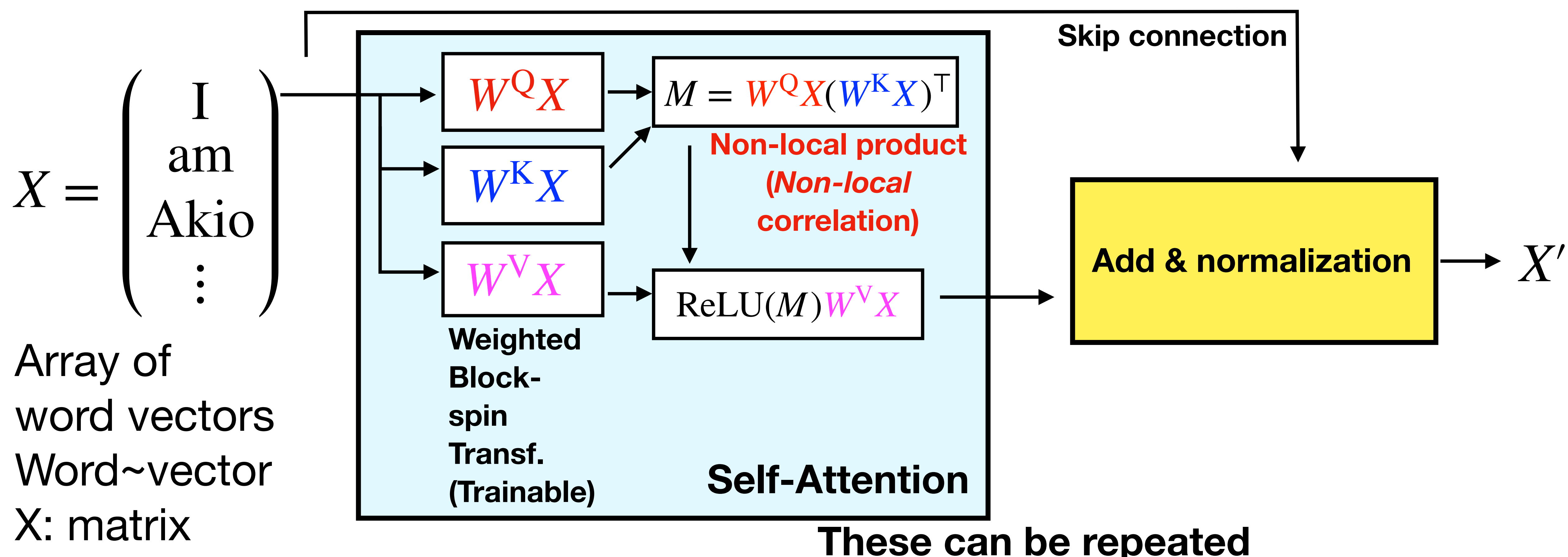
Modifier in language can be non-local

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



Akio Tomiya

Transformer and Attention

Transformer shows scaling lows (power law)

arXiv: 2001.08361

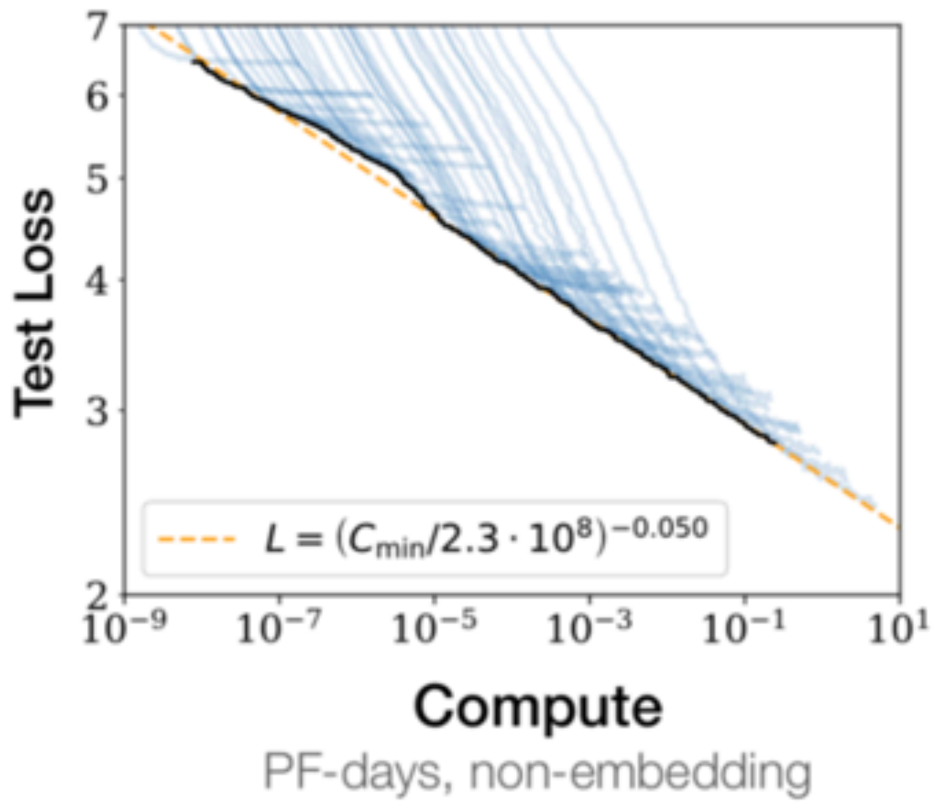


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when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

Akio Tomiya

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 Bard Vision Transformer arXiv:1706.03762
(This work) Physically *Equivariant* attention	Yes 👍	Global 👍	?	This work arXiv: 2306.11527

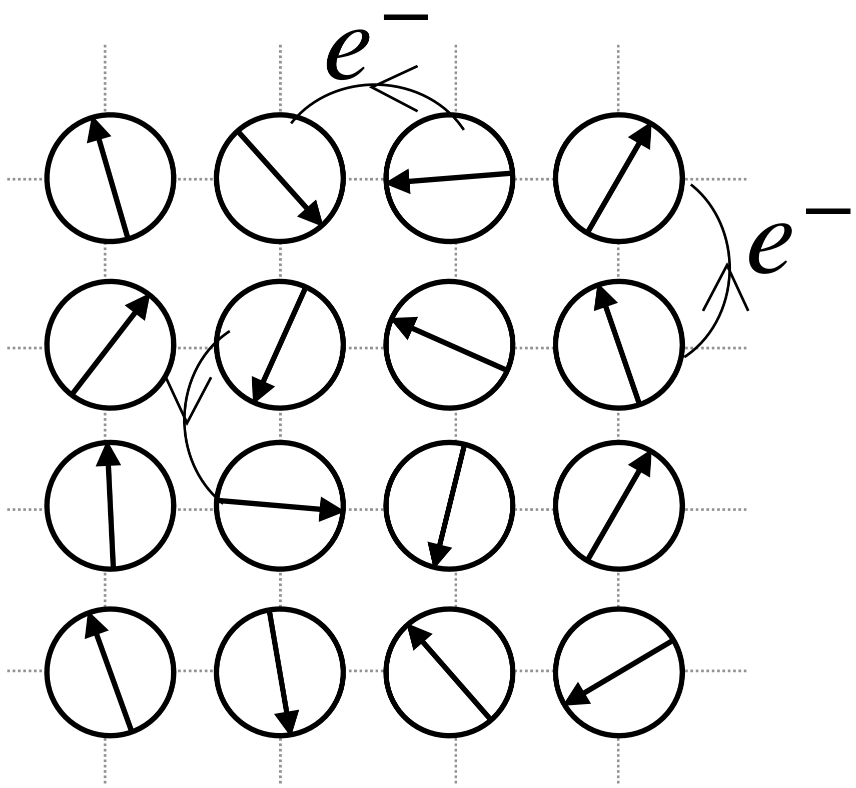
Self-learning Monte-Carlo

Akio Tomiya

Target: Double exchange model

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

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



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

(This system is similar to lattice QCD but easier)

Self-learning Monte-Carlo

Akio Tomiya

Previous work

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

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

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

Self-learning Monte-Carlo:

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

This is an exact algorithms

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

Self-learning Monte-Carlo

Previous work

Akio Tomiya

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

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

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**

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

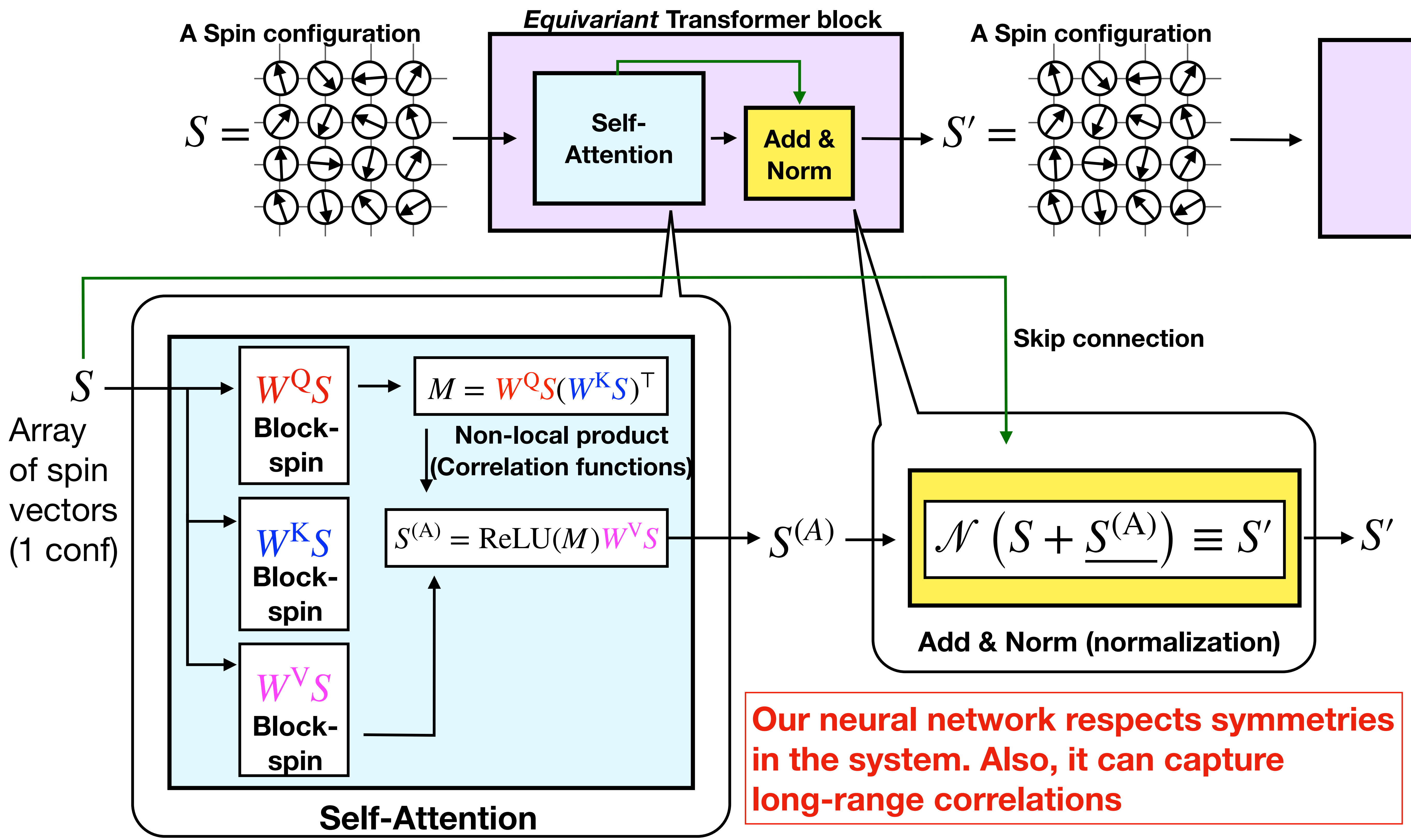
We replace this by
“translated” spin S_i^{NN}
with a transformer
and used in self-learning MC

Self-learning Monte-Carlo

Akio Tomiya

Physically equivariant Attention layer/Transformer

arXiv: 2306.11527.



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

Self-learning Monte-Carlo

SLMC = MCMC with an effective model

Akio Tomiya

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 $H_{\rm eff}(\{\mathbf{S}\})$ can compose in MCMC,

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

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

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

Self-learning Monte-Carlo

Akio Tomiya

SLMC = MCMC with an effective model

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 $H_{\rm eff}(\{\mathbf{S}\})$ can compose in MCMC,

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

if the update「$\rightarrow$」 satisfies the detailed balance. We can employ Metropolis test like

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

SLMC: Self-learning Monte-Carlo

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

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

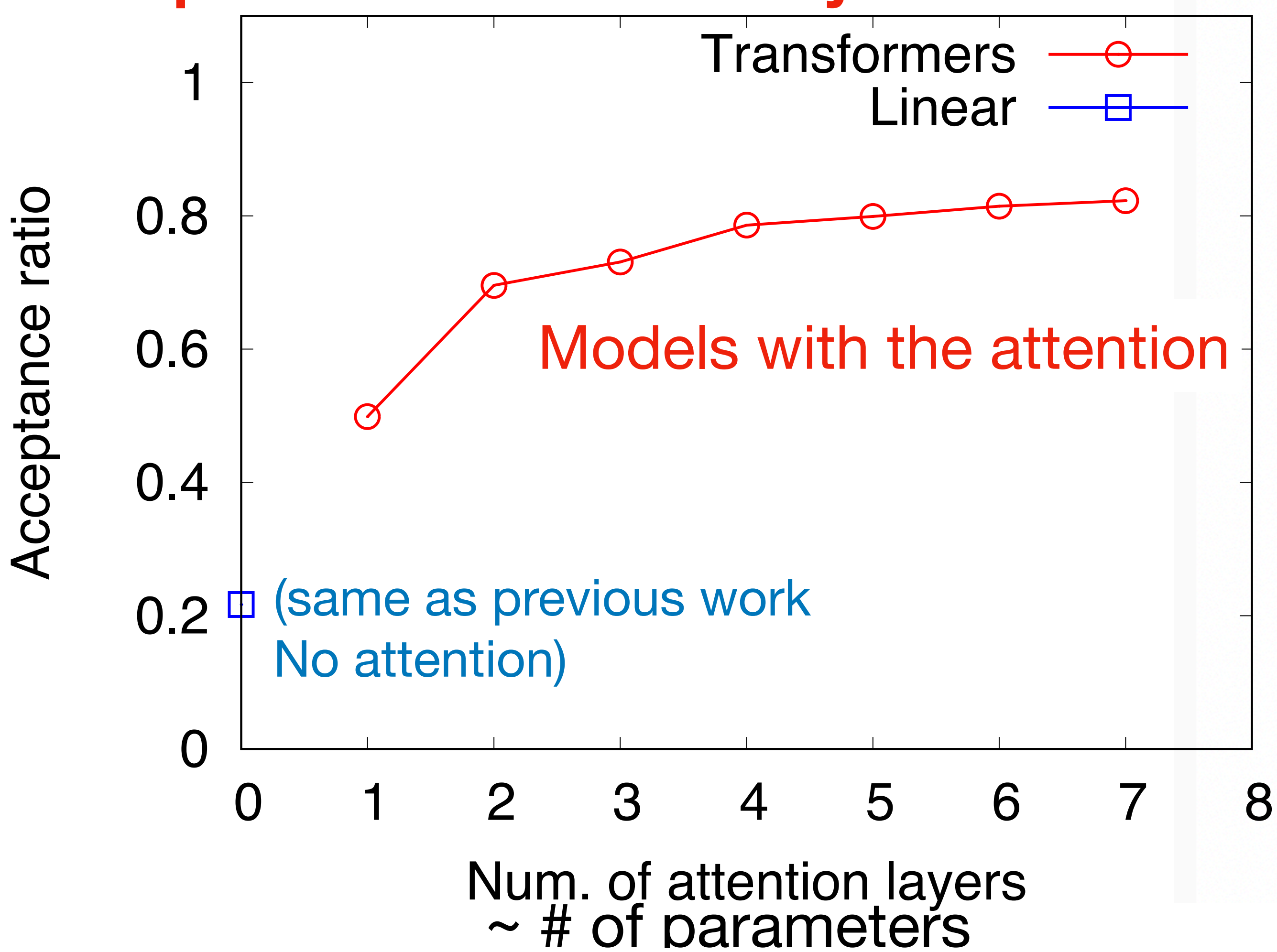
with Metropolis-*Hastings* test: $A(\{\mathbf{S}'\},\{\mathbf{S}\}) = \min\left(1, \dfrac{W(\{\mathbf{S}'\})}{W(\{\mathbf{S}\})}\dfrac{W_{\rm eff}(\{\mathbf{S}\})}{W_{\rm eff}(\{\mathbf{S}'\})}\right).$

- **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

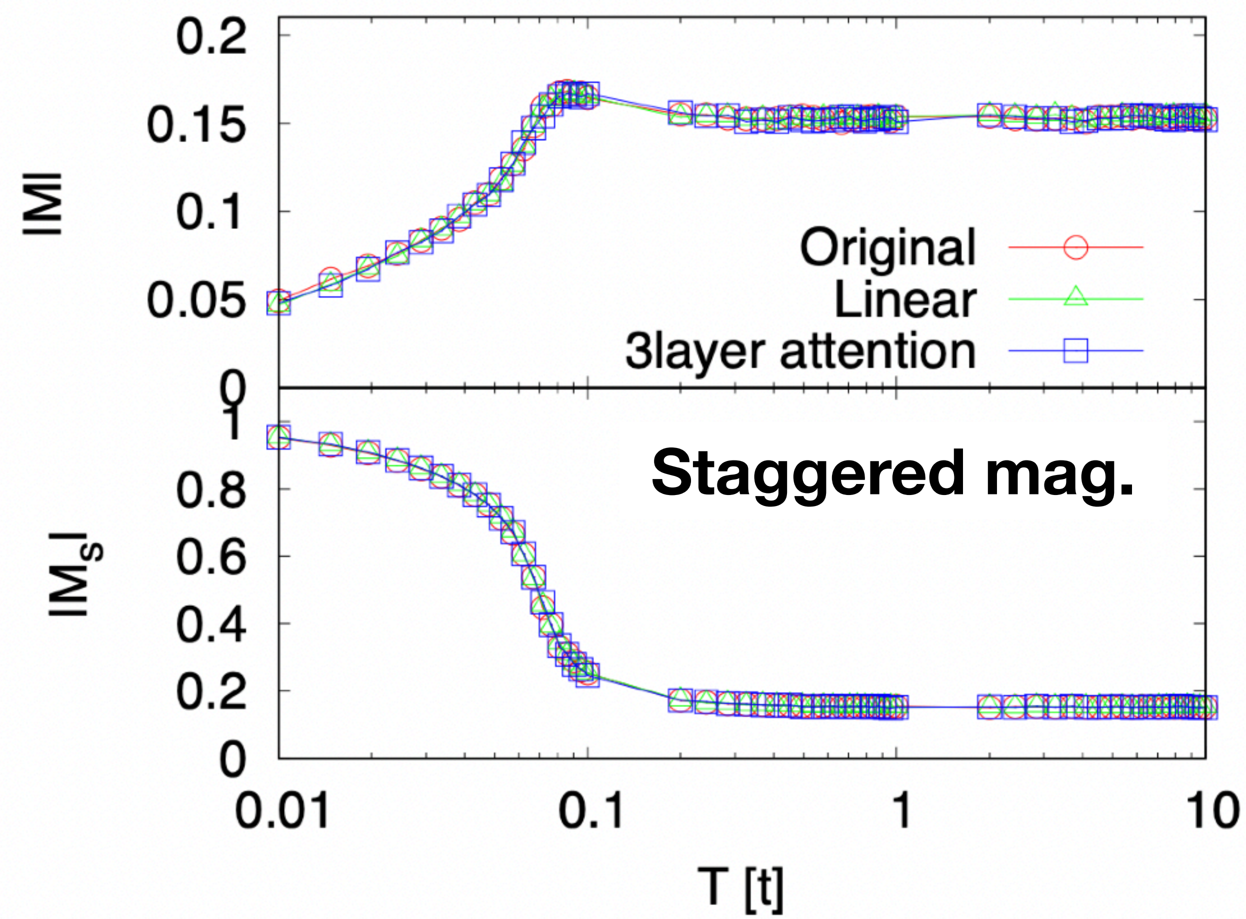
Application to O(3) spin model with fermions

Acceptance rate ~ efficiency



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)

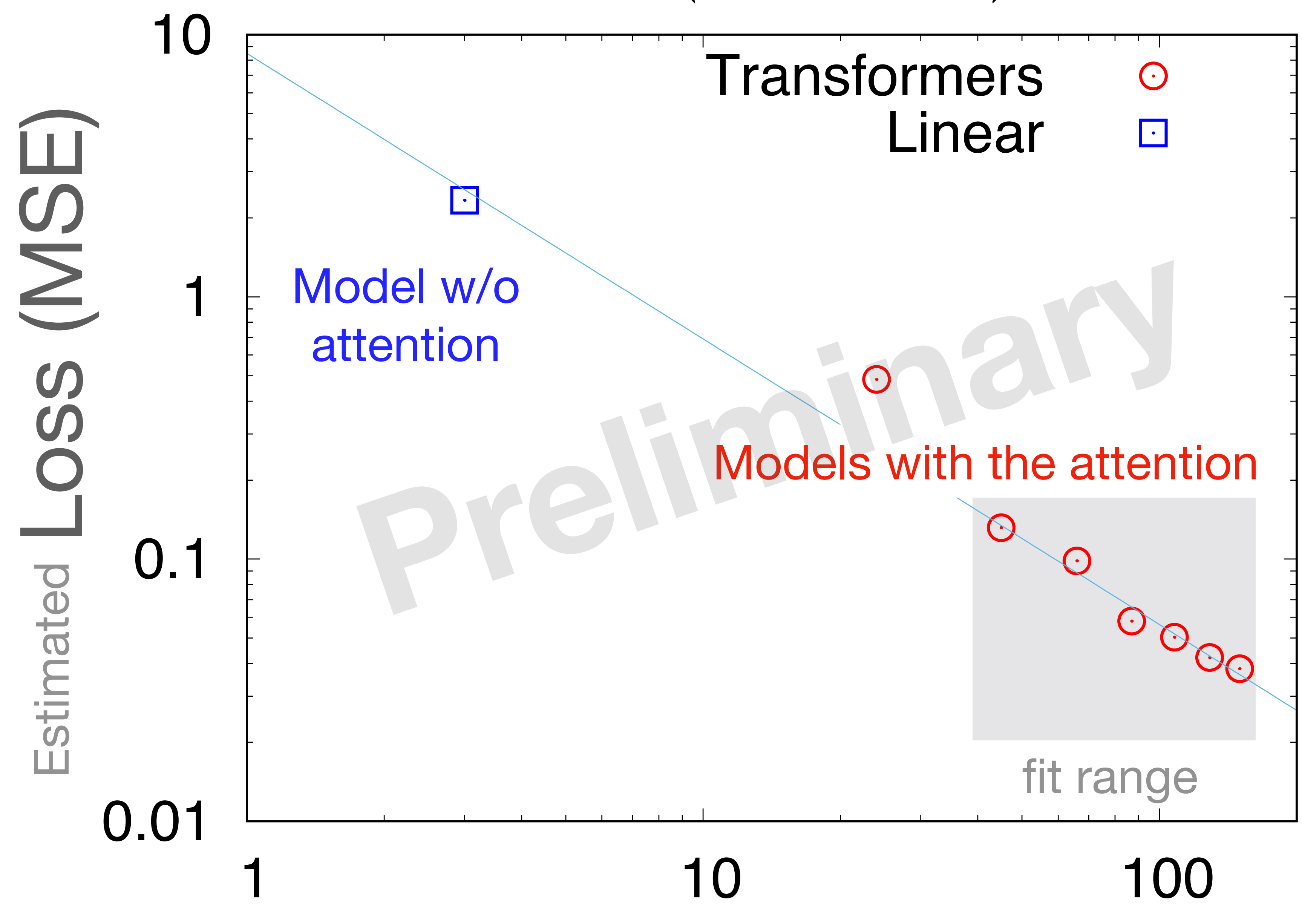
Transformer and Attention

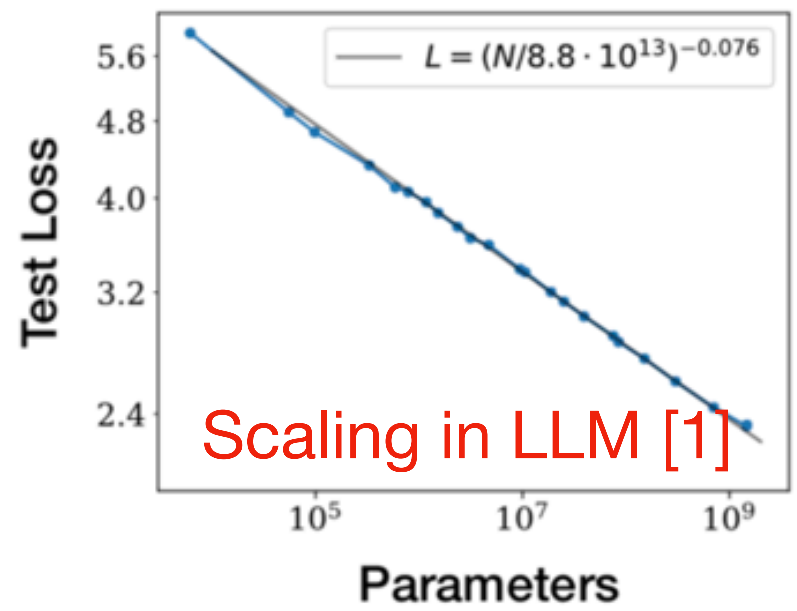
Akio Tomiya

Loss function shows Power-type scaling law as LLM

arXiv: 2306.11527 + update

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





Line is just for guiding eyes (no meaning)

fit ~(7.1/x)^(1.1)

[1] arXiv:2001.08361

総合的に速くするには?

AI(ML) + Science
速く・正確に

AI (機械学習手法) は、なんとかできることが増えてきた

- 既存手法で担保すべき「厳密性」
- 他にもAMAのような系統誤差補正手法もある

結局、どこに向かうのか

- 既存手法 + AI手法のハイブリッド (eg 自己学習モンテカルロや前処理)
- 速くなる ≠ 処理が高速
  -> 例えば遅くても、MCなら単位時間あたりに独立な配位をいくつ作れる?
- 機械学習自体の高速化、量子化、スパースアテンション

HPC系の技術とAIは相性が悪いことも多い

- 量子化 (低精度表現)
- データ量
- (今のところ)ML は低精度GPUで出来てしまう、今後もおそらく続く

AI(ML) + Science
速く・正確に

AI (機械学習手法) は、なんとかできることが増えてきた

- 既存手法で担保すべき「厳密性」
- 他にもAMAのような系統誤差補正手法もある

結局、どこに向かうのか

- 既存手法 + AI手法のハイブリッド (eg 自己学習モンテカルロや前処理)
- 速くなる ≠ 処理が高速
  -> 例えば遅くても、MCなら単位時間あたりに独 … ?
- 機械学習自体の高速化、量子化、スパースアテン …

HPC系の技術とAIは相性が悪いことも多い

- 量子化 (低精度表現)
- データ量
- (今のところ)ML は低精度GPUで出来てしまう、今後もおそらく続く

新しいアーキテクチャの提案

(ドメイン知識を生かしたNNの設計)

AIコード自身の高速化

アルゴリズム改良

コードをどうするか

AI手法との共存

AI用ハードウェア

Summary

Machine learning + lattice field theory

- Machine learning is useful for natural science/physics/Lattice QCD
    - Multi-dimensional integration is done by MCMC
- MCMC proposals can be made by Machine learning
    - Transformer for a spin+fermion system
        - Scaling law for a Transformer for physical system
- Future work: Transformer for lattice gauge theory
- Combining ML and expert knowledge (e.g. symmetry) of computational physics/LatticeQCD is important
- How can we use AI for science (open question)
    - We know AI is useful for data generation though

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!

9/9 - 9/13

格子QCDのサマースクールを行います

筑波大学東京キャンパス

場の理論の基礎から格子QCDの最先端まで

4月くらいに正式情報がでます。