CCS symposium, 15th symposium on Discovery, Fusion, Creation of New Knowledge by
Multidisciplinary Computational Sciences: Program of Parallel sessions
2 Oct 2023 CCS, Tsukuba university
https://www.ccs.tsukuba.ac.jp/sympo20231002en/

15th Symposium on
Discovery, Fusion, Creation of
New Knowledge by
Multidisciplinary Computational Sciences

Integrating Machine Learning into Lattice QCD

Akio Tomiya (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

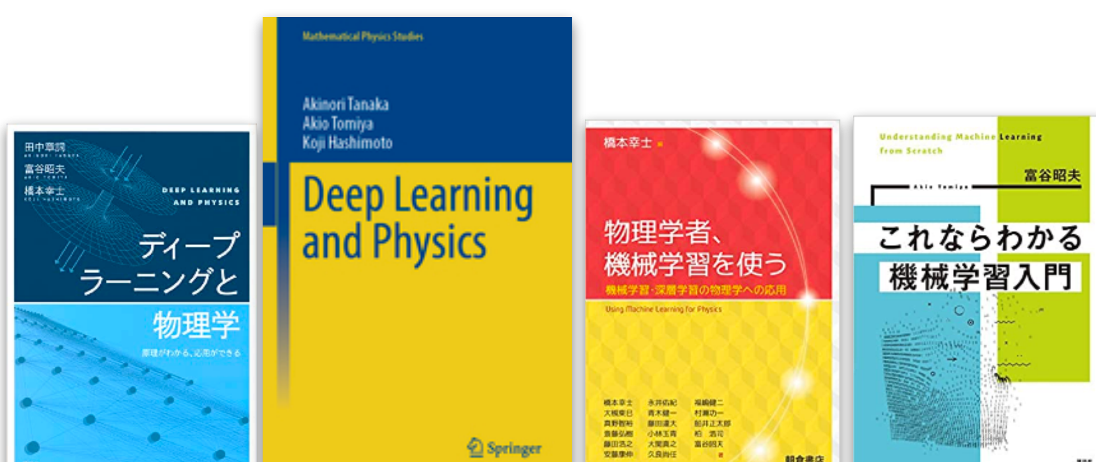
Detecting phase transition

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 (Cond. mat.)
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)



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

Kakenhi and others

Leader of proj A01 Transformative Research Areas

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:

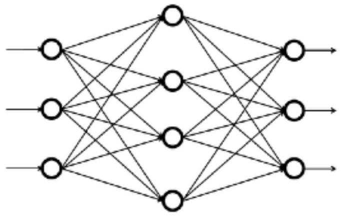
Organizing "Deep Learning and physics"

Supervision of Shin-Kamen Rider

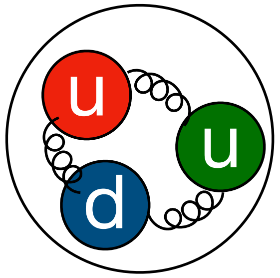
Outline of my talk

Message: I’ve been developing neural networks for lattice QCD

Machine learning?
Neural network?

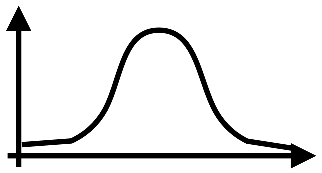


Lattice QCD?



How to calculate it
Problem

3 methods with ML

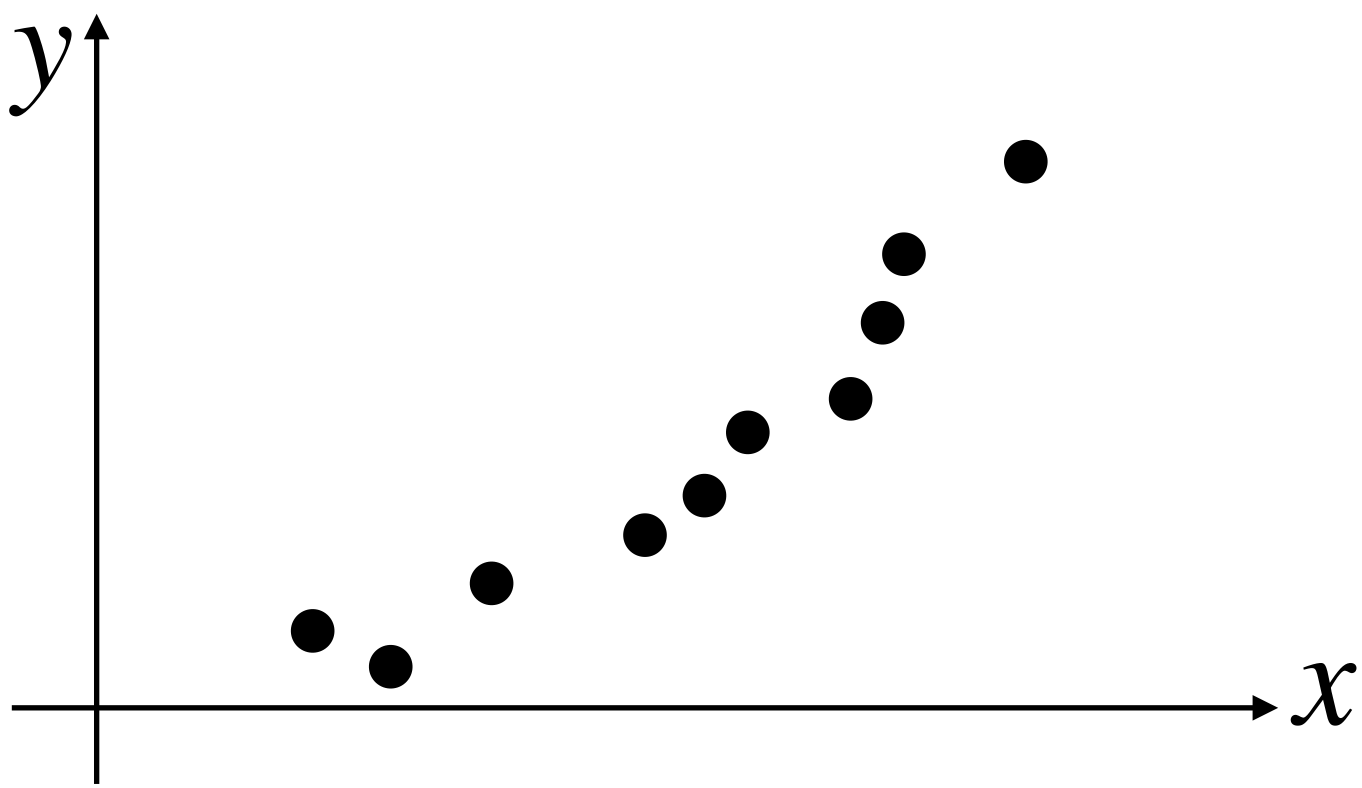


What is machine learning/AI?

What is machine learning?

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

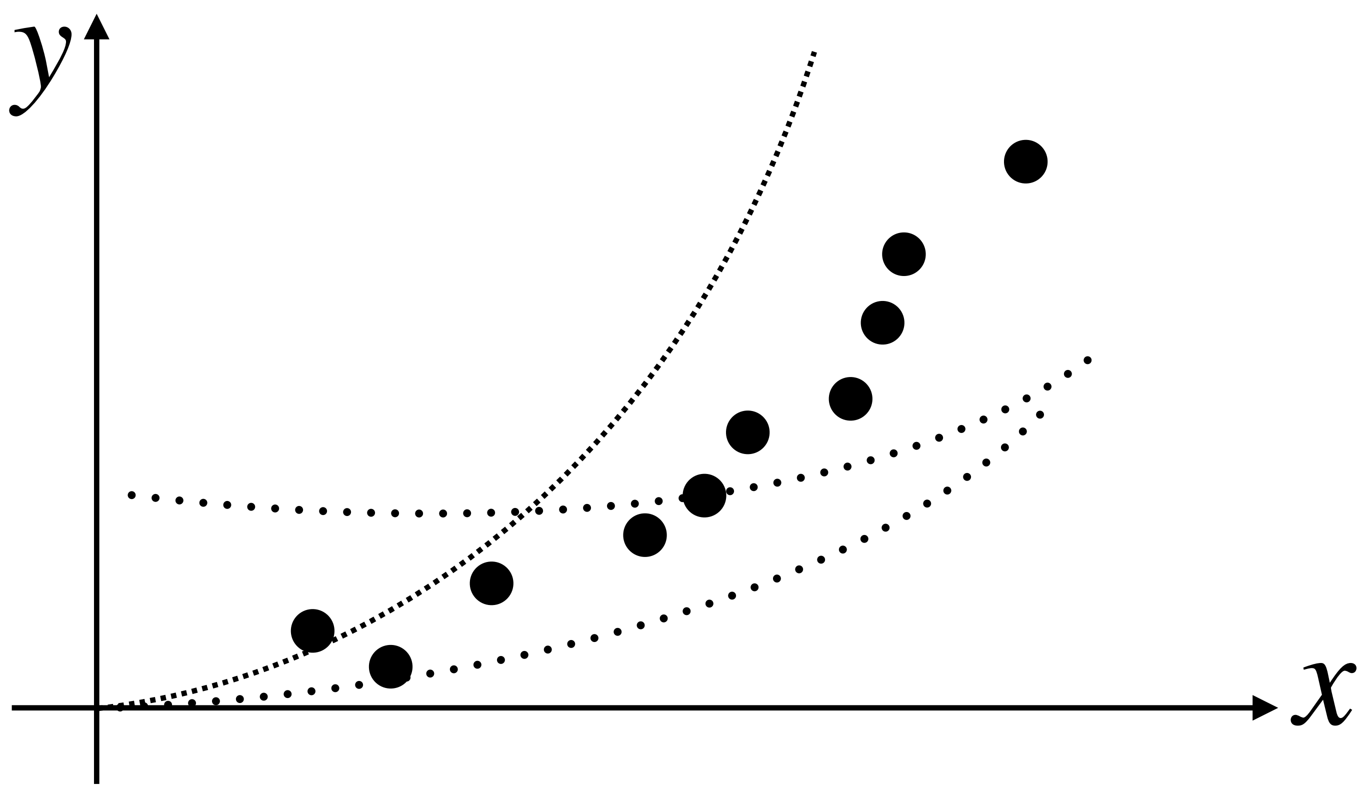
Data: $D = \left\{(x^{(1)}, y^{(1)}),\ (x^{(2)}, y^{(2)}),\ \cdots\right\}$



What is machine learning?

E.g. Linear regression ∈ Supervised learning

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



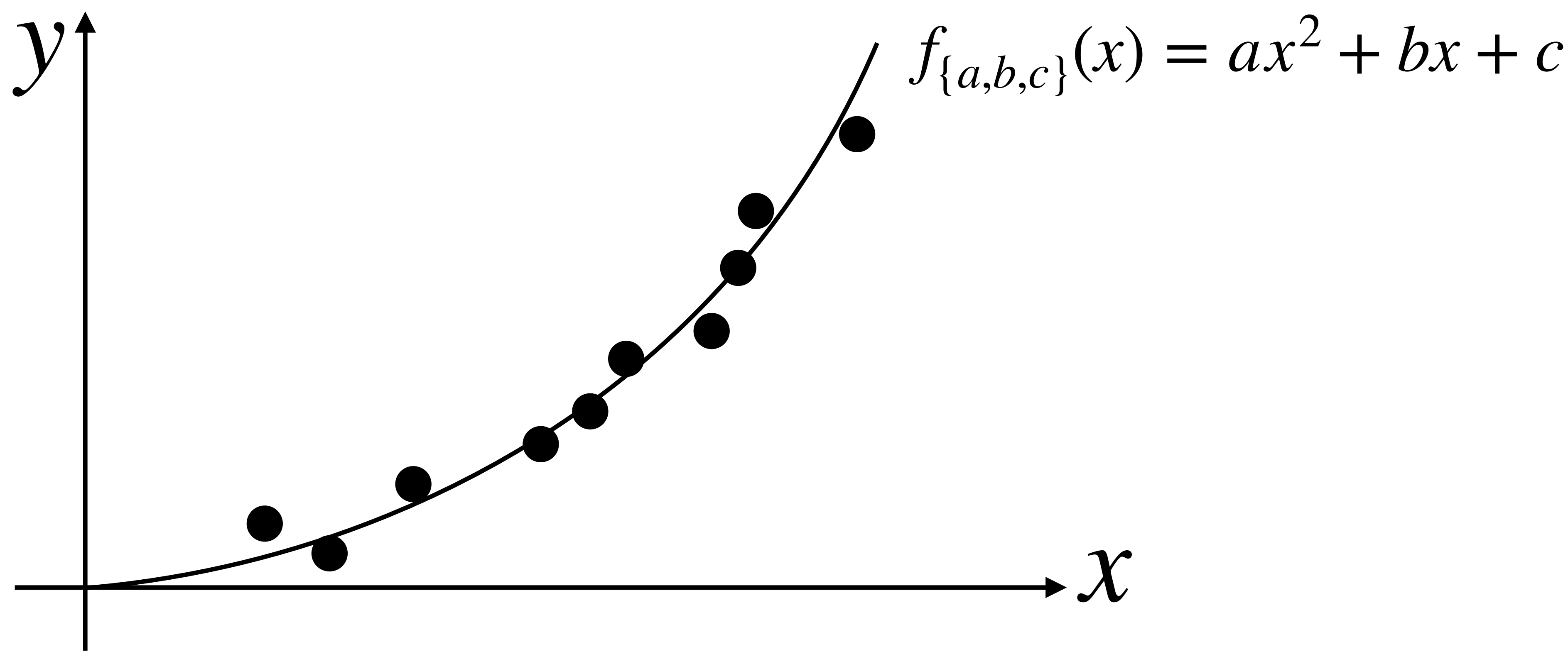
$$f_{\{a,b,c\}}(x) = ax^2 + bx + c \qquad E = \frac{1}{2}\sum_d \left| f_{\{a,b,c\}}(x^{(d)}) - y^{(d)} \right|^2$$

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

What is machine learning?

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

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



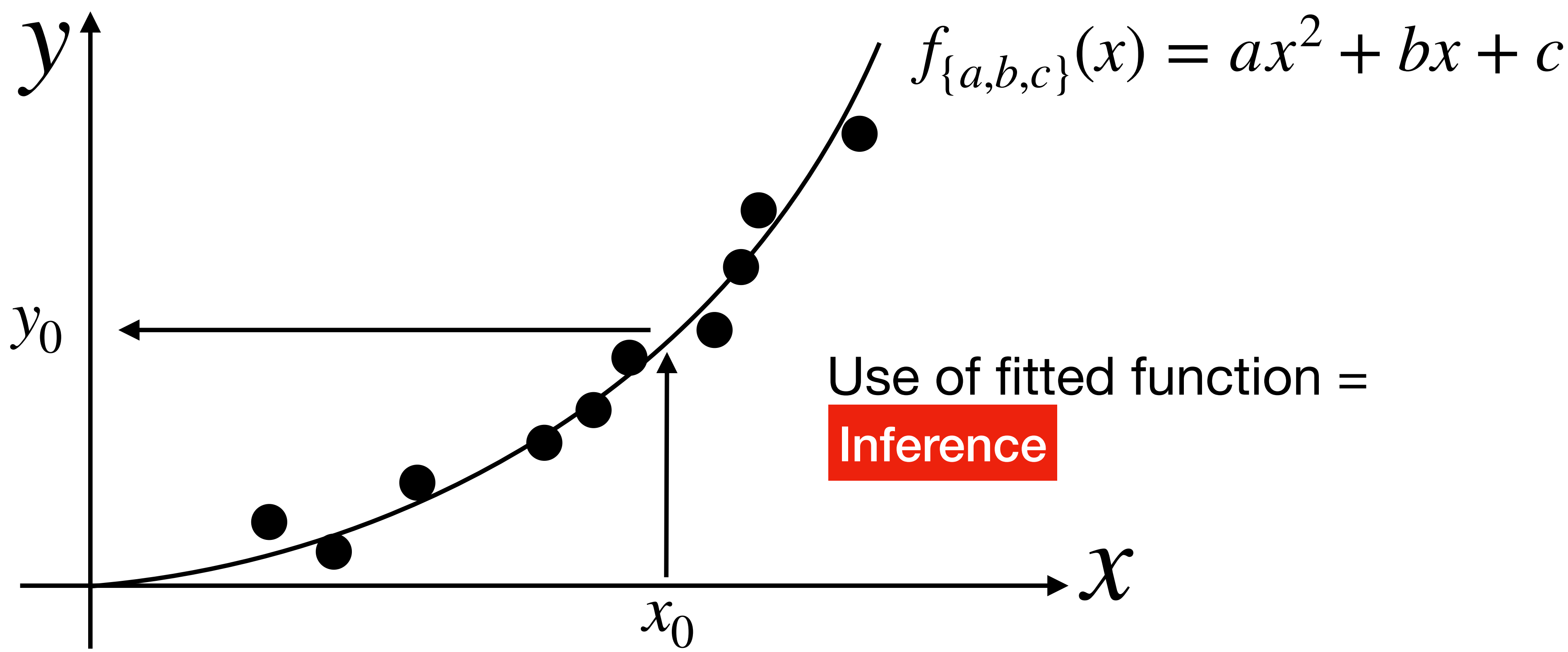
$f_{\{a,b,c\}}(x) = ax^2 + bx + c$ $\qquad$ $E = \frac{1}{2} \sum_d \left| f_{\{a,b,c\}}(x^{(d)}) - y^{(d)} \right|^2$

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

What is machine learning?

E.g. Linear regression ∈ Supervised learning

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



Use of fitted function = **Inference**

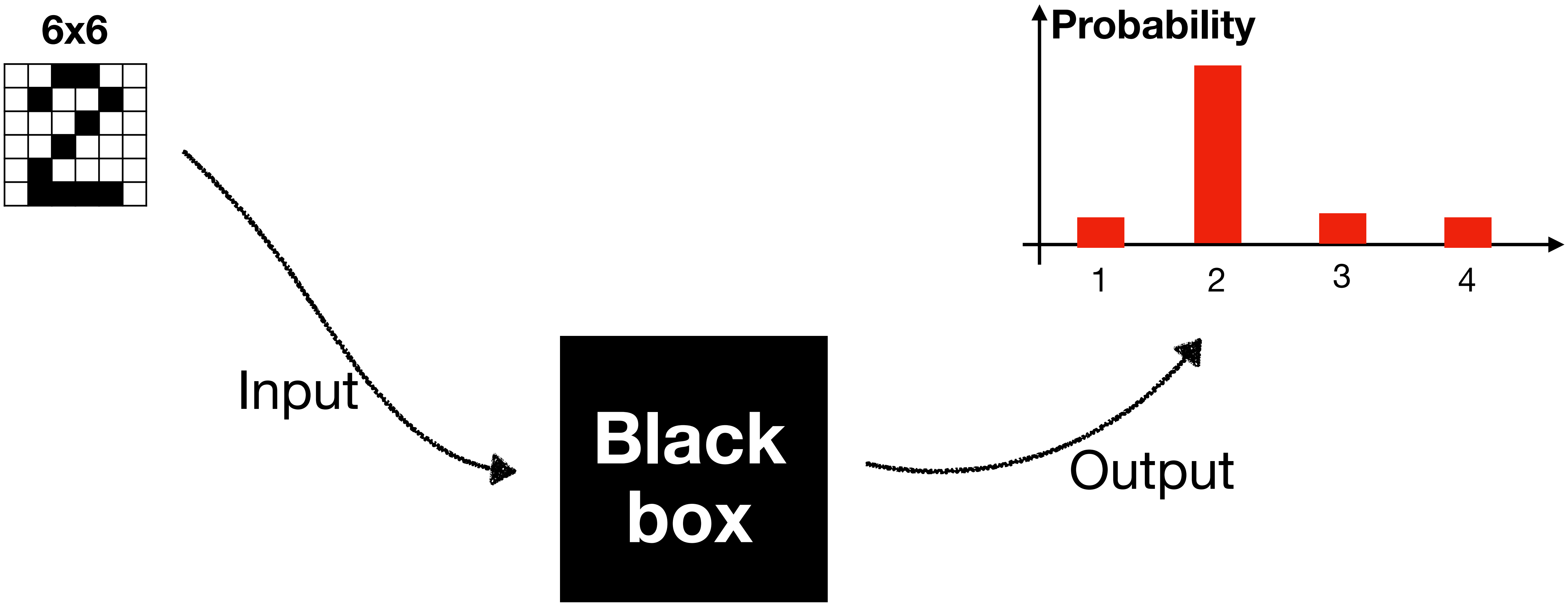
Now we can predict y value which not in the data

In physics language, variational method

What is the neural networks?

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)

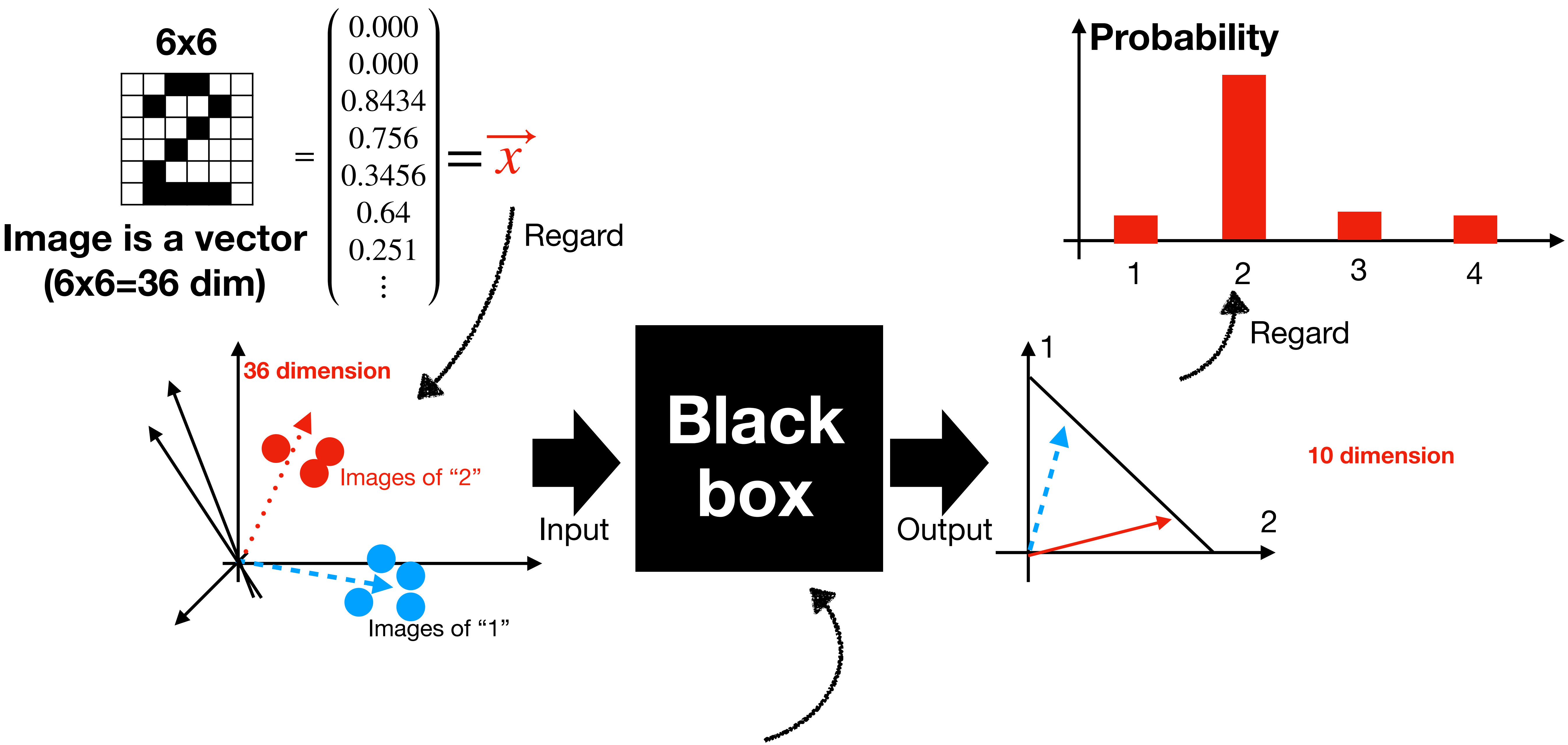


Image recognition = Find a map between two vector spaces

What is the neural networks?

Akio Tomiya

Neural network is a *universal* approximation function

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

6x6

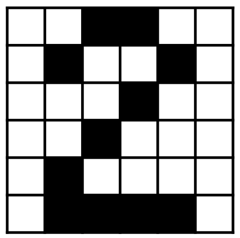
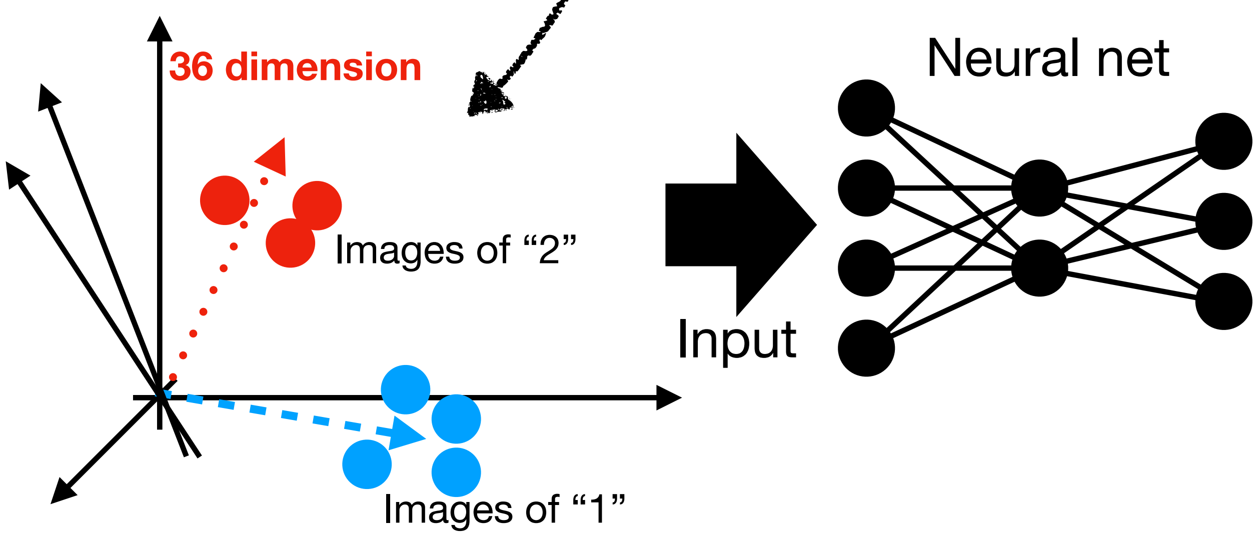


Image is a vector (6x6=36 dim)

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

Regard



What is the neural networks?

Akio Tomiya

Affine transformation + element-wise transformation

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

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

Affine transformation

(b=0 called linear transformation)

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

A fully connected neural net:

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

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

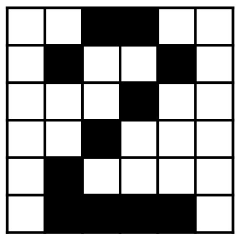
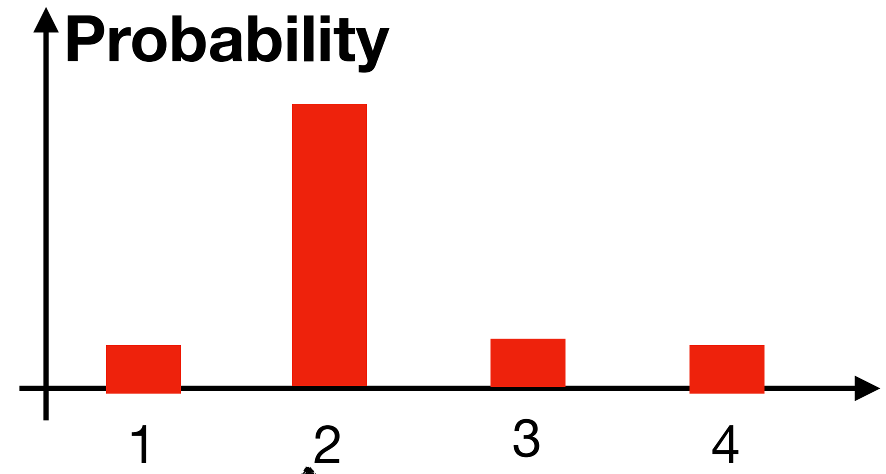
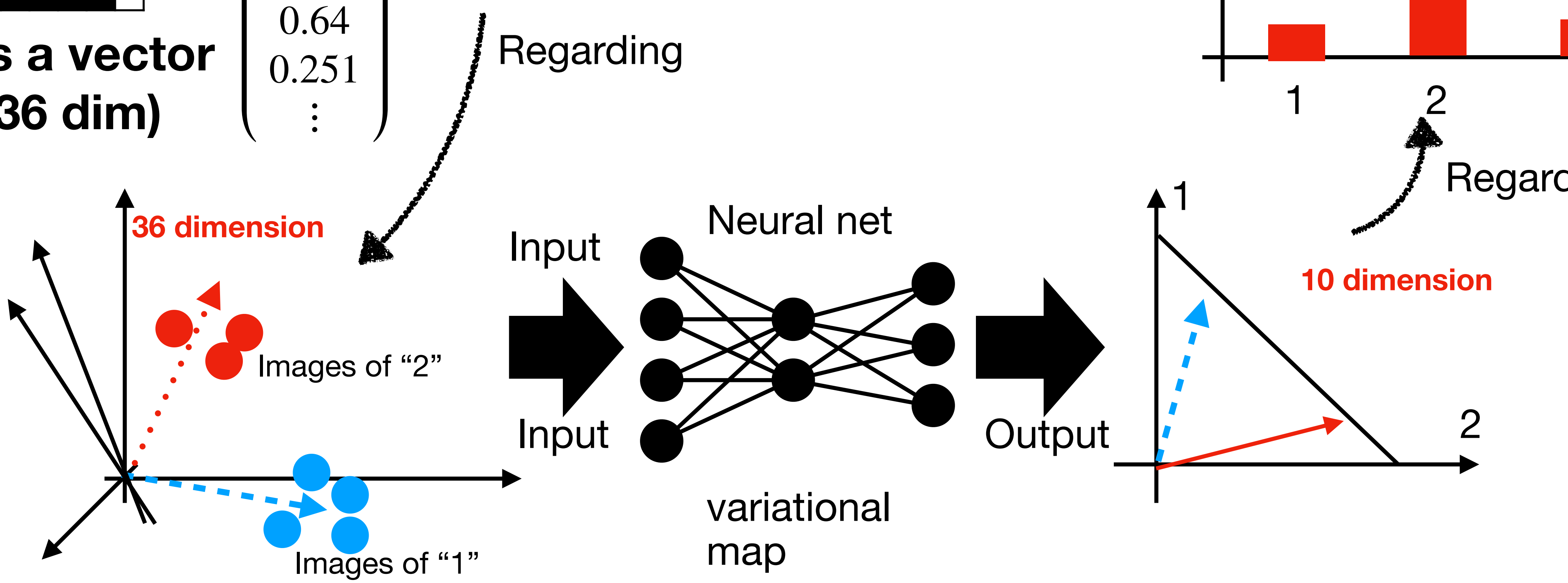


Image is a vector (6x6=36 dim)

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

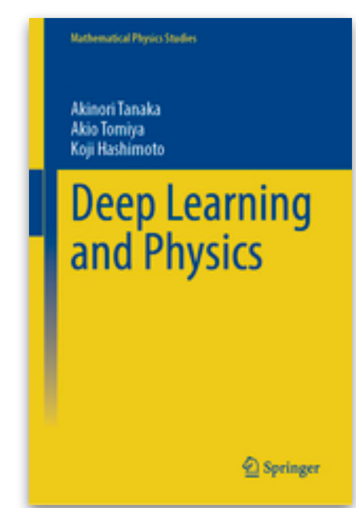




"0" = (1,0,0,…)
"1" = (0,1,0,…)
"2" = (0,0,1,…)
…
"9" = (0,0,…,1)

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

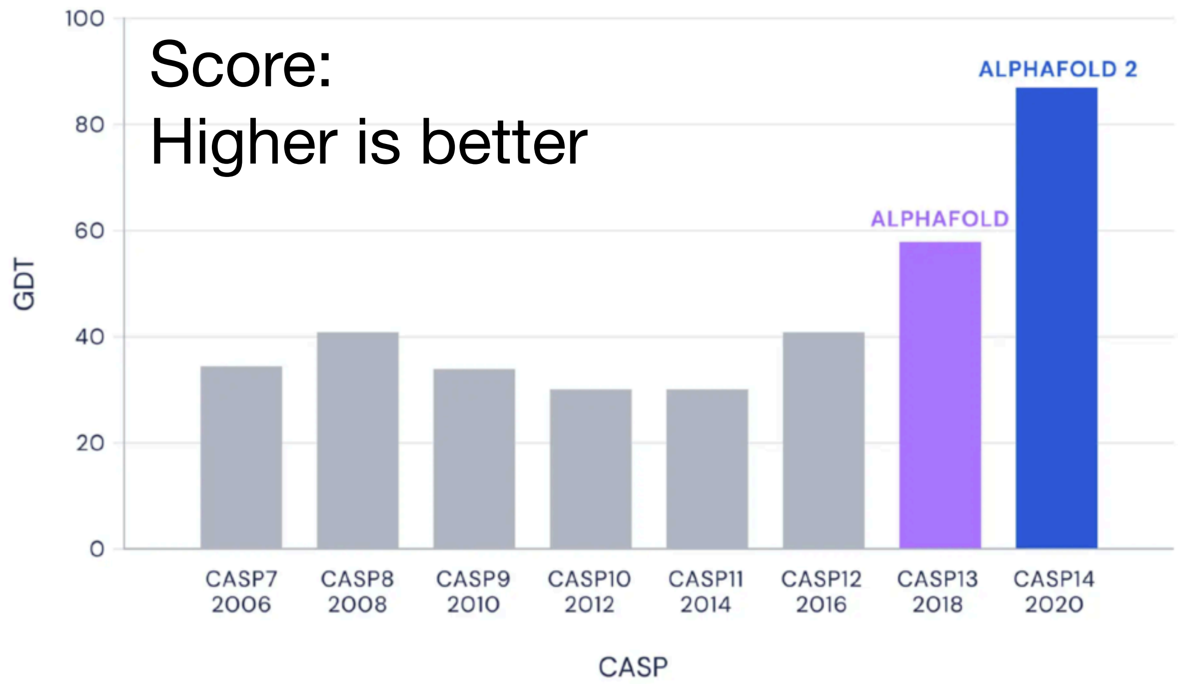
In this example, NN mimics image (36-dim vector) and label (10-dim vector)

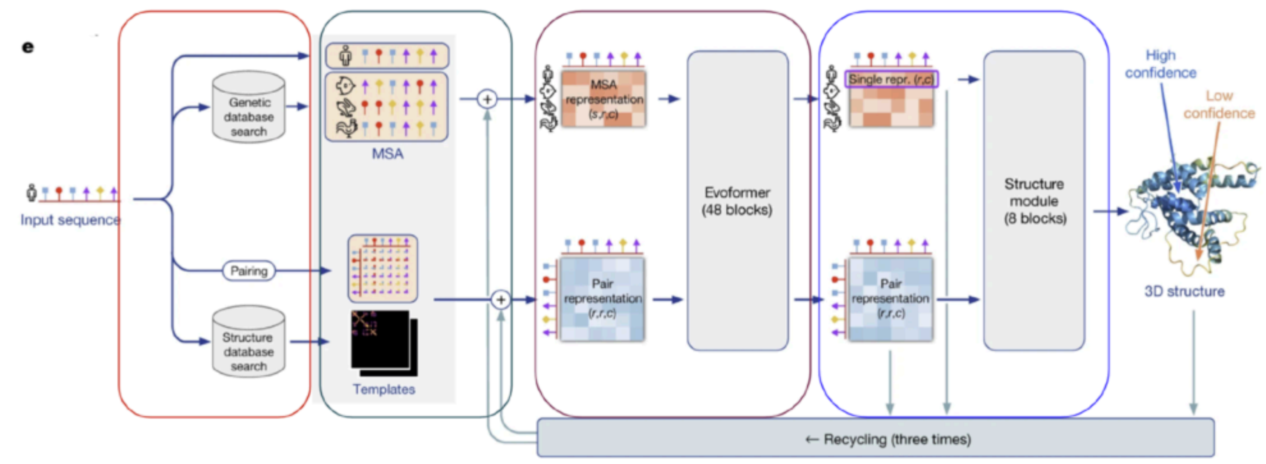


What is the neural networks?

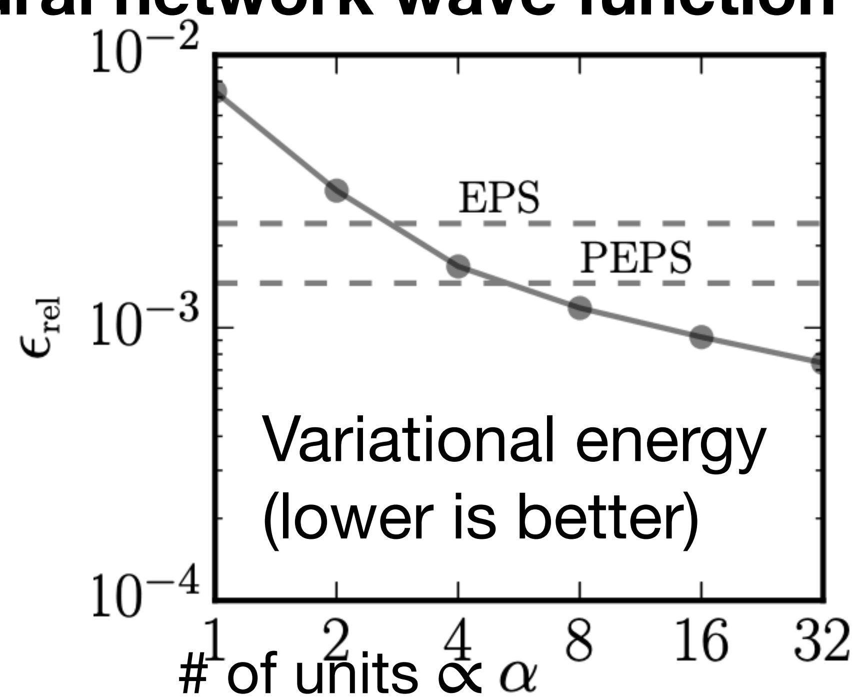
Neural network have been good job

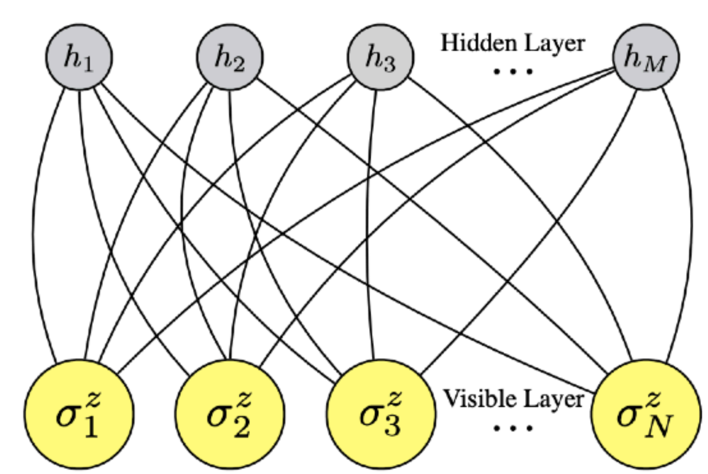
Protein Folding (AlphaFold2, John Jumper+, Nature, 2020+), Transformer neural net





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





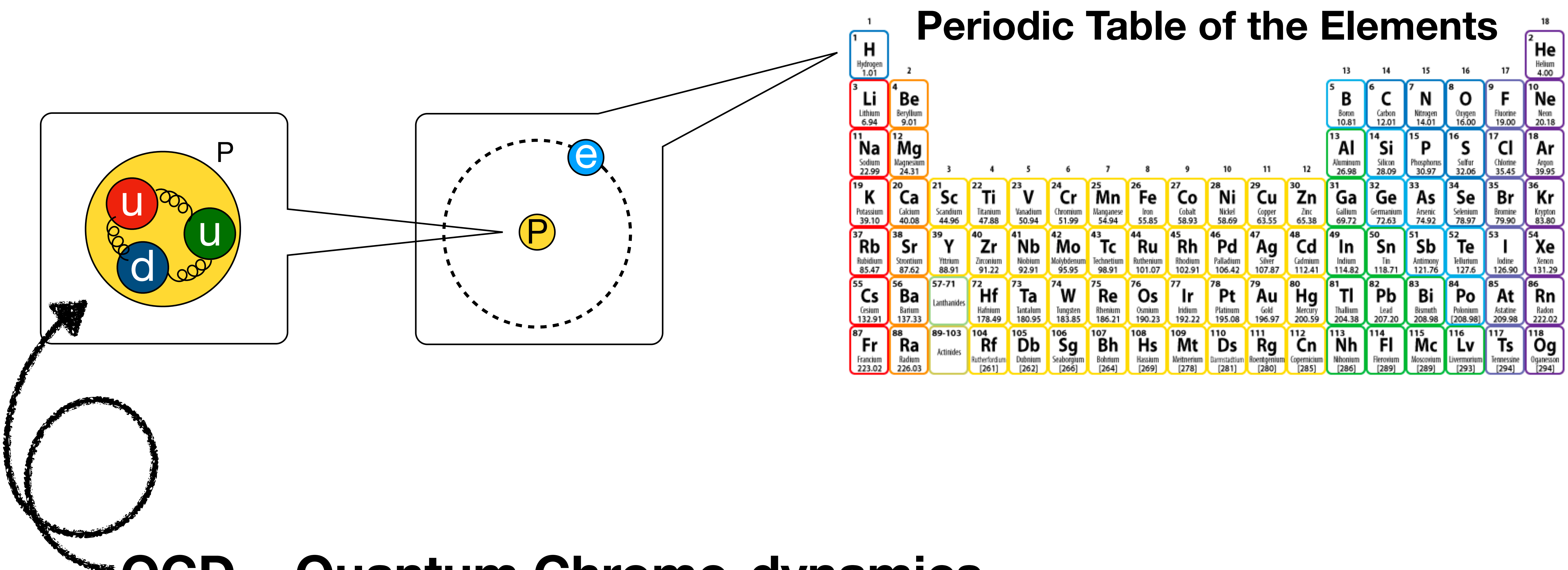
Neural net + “Expert knowledge” → Best performance

https://horomary.hatenablog.com/entry/2021/10/01/194825
https://www.aist.go.jp/aist_j/press_release/pr2020/pr20201111/pr20201111.html

What is Lattice QCD?

Introduction

What is QCD?



QCD = Quantum Chromo-dynamics

= A fundamental theory for particles in nuclei

Quantum many body, relativistic, strongly correlated

Introduction

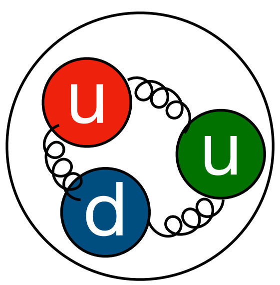
Lattice QCD = QCD on discretized spacetime = calculable

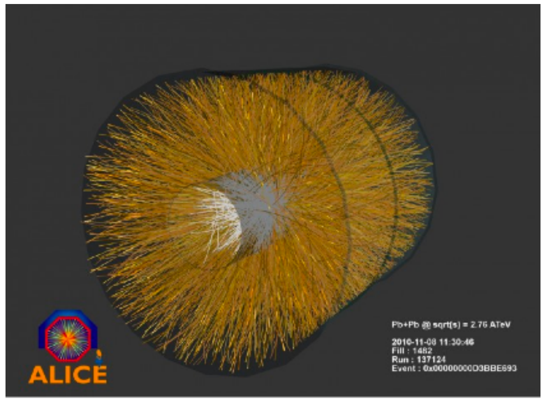
QCD (Quantum Chromo-dynamics) in 3 + 1 dimension

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

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

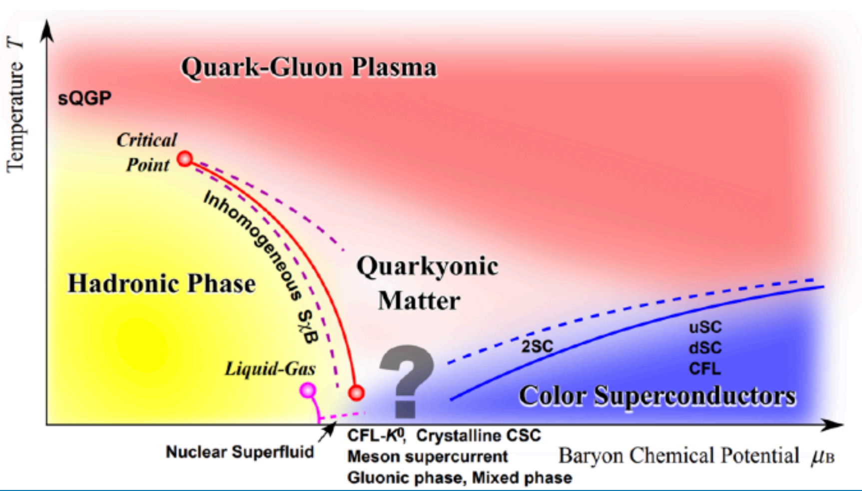
Non-commutable version of (quantum) electro-magnetism





- This describes…
  - inside of nuclei,& mass of hadrons
  - Equation of state of neutron stars, Heavy ion collisions, etc
- **We want to evaluate expectation values with following integral,**

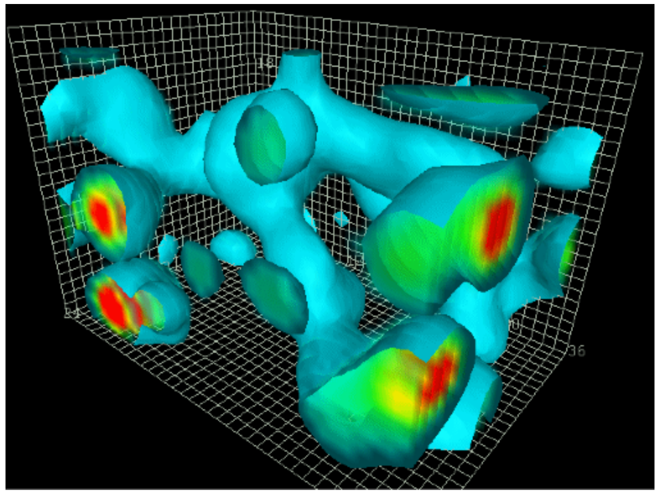
$$\langle O \rangle \sim \int \mathcal{D}A\mathcal{D}\bar{\psi}\mathcal{D}\psi e^{\mathrm{i}S}$$

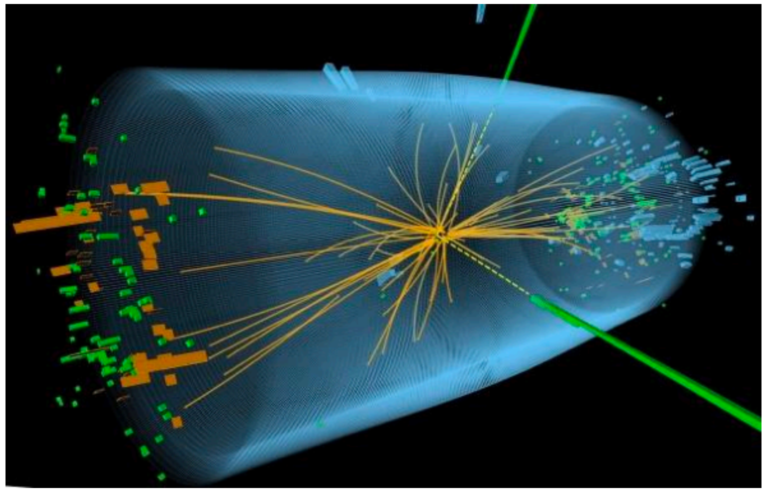


Lattice formulation enables us to do that

Introduction

What is our final goal for our research field?





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
  → related to the history of the universe
- attractive/repulsive forces between atomic nuclei
  → to understand how stars are born and die
- candidate properties of dark matter

etc.

We want to understand our universe from fundamental level!

Motivation

Monte-Carlo integration is available, but still expensive!

M. Creutz 1980

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

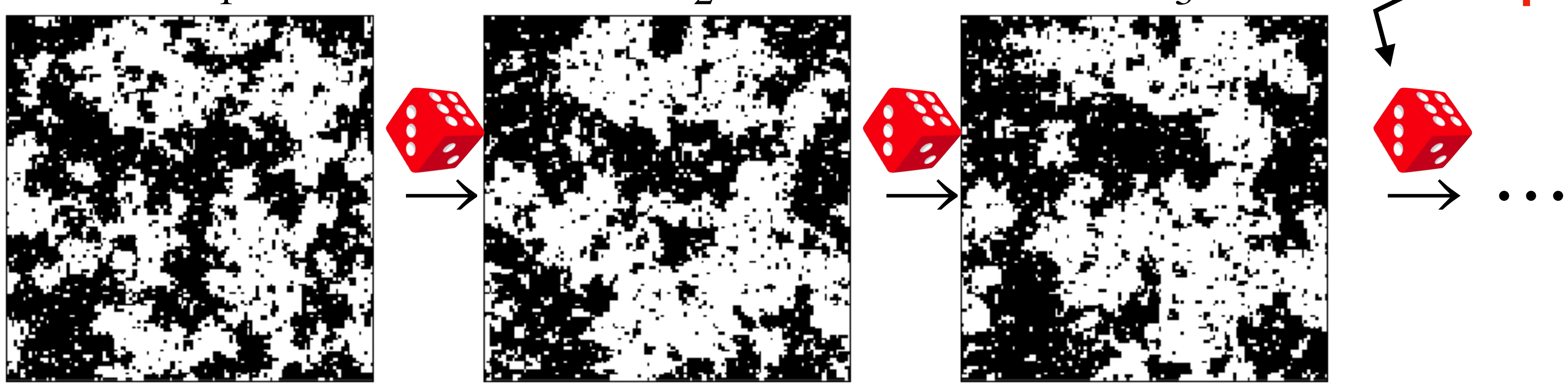
$$S_{\text{eff}}[U] = S_{\text{gauge}}[U] - \log \det(\mathbb{D}[U] + m)$$

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

U_1 U_2 U_3

Propose and check

Markov-Chain



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

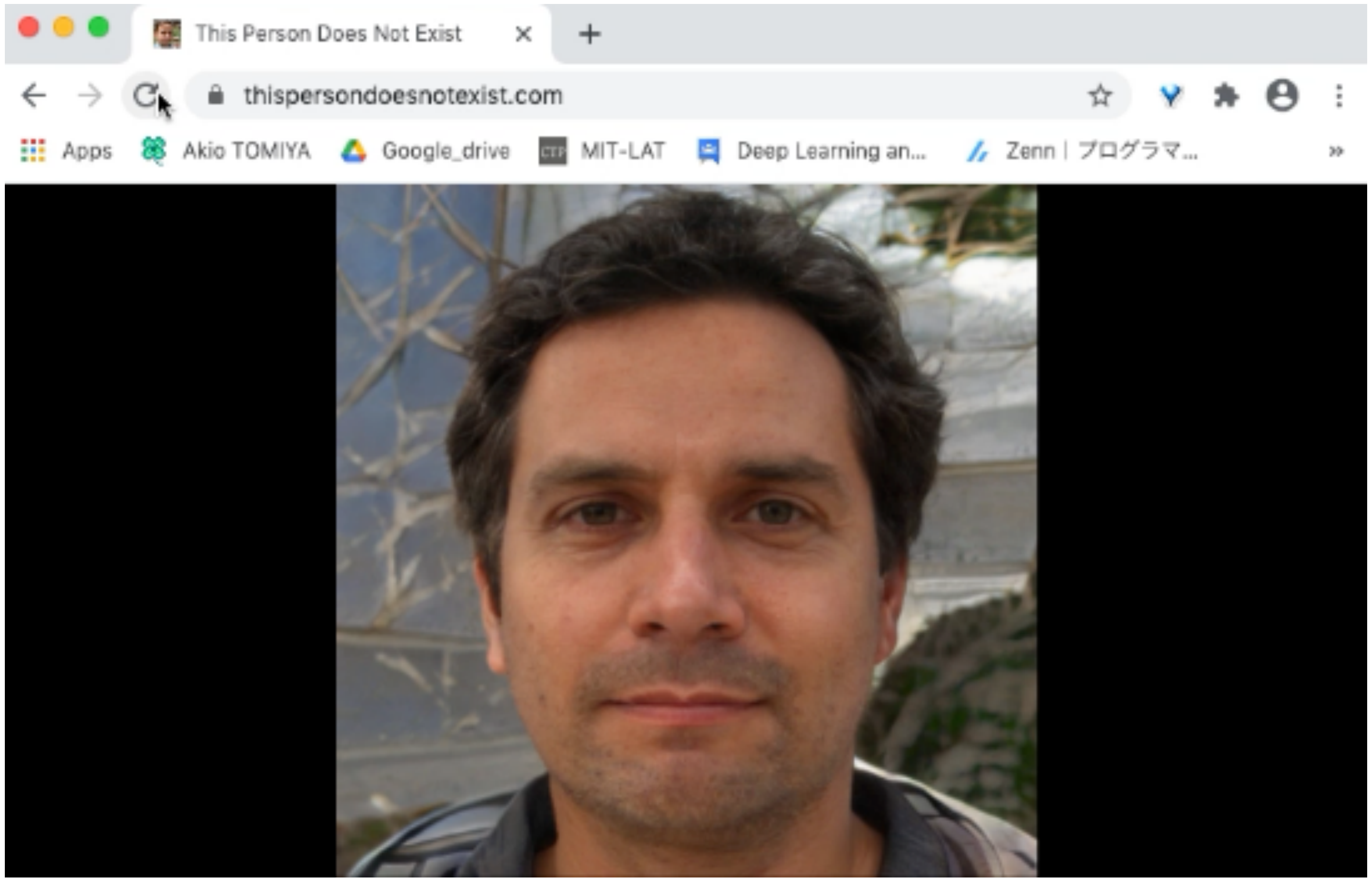
Numerically expensive (part) and how can we accelerate it?

We use machine learning!

Introduction

Neural net can make human face images

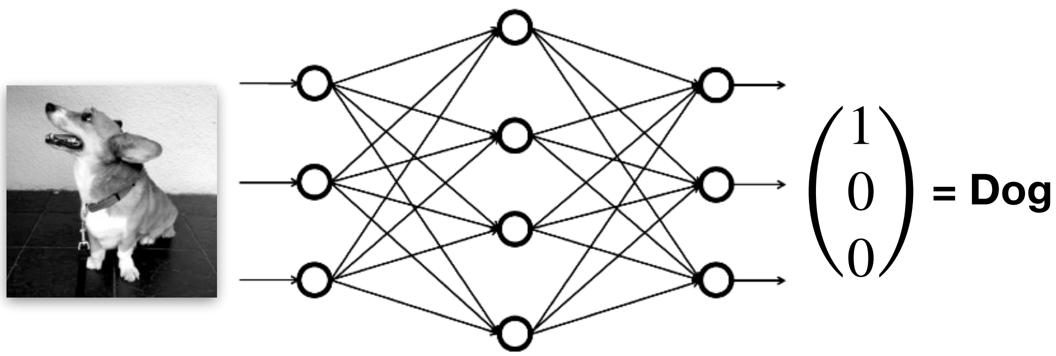
Neural nets can generate realistic human faces (Style GAN2)

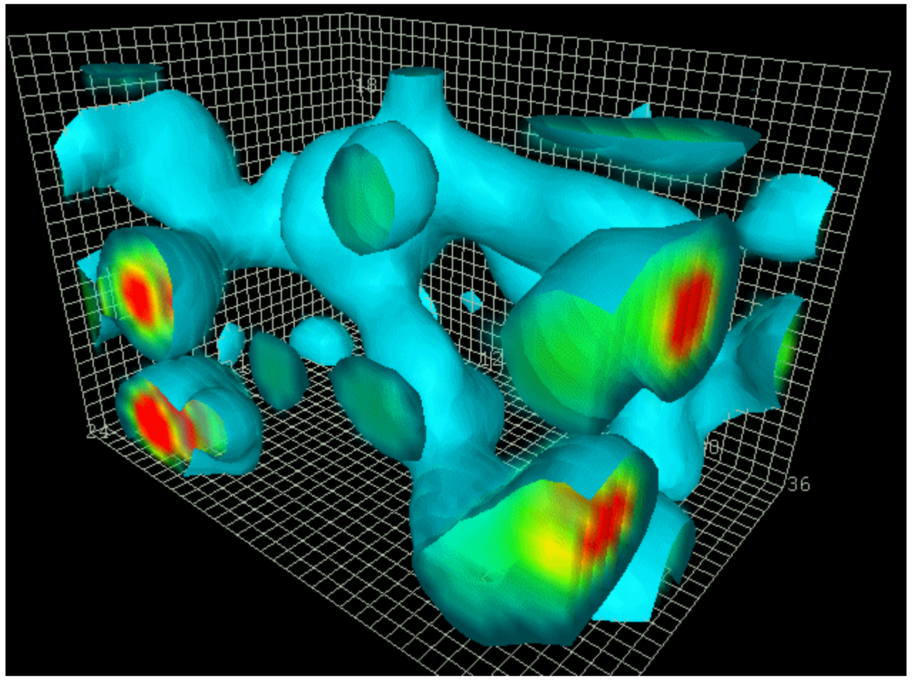


Realistic Images can be generated by machine learning!
Configurations as well? (proposals ~ images?)

ML for LQCD is needed

- Machine learning/ Neural networks
  - data processing techniques for 2d/3d data in the real world (pictures)
  - (Variational) Approximation (∼ fitting)
- Lattice QCD is more complicated than pictures
  - 4 dimension/relativistic
  - Non-abelian gauge symmetry (difficult)
  - Fermions (anti-commuting/fully quantum)
  - 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. Medvidović+	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

+...

Three methods with machine learning 1/3: Flow based sampling

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), like flat measure/Gaussian, 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

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

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.

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

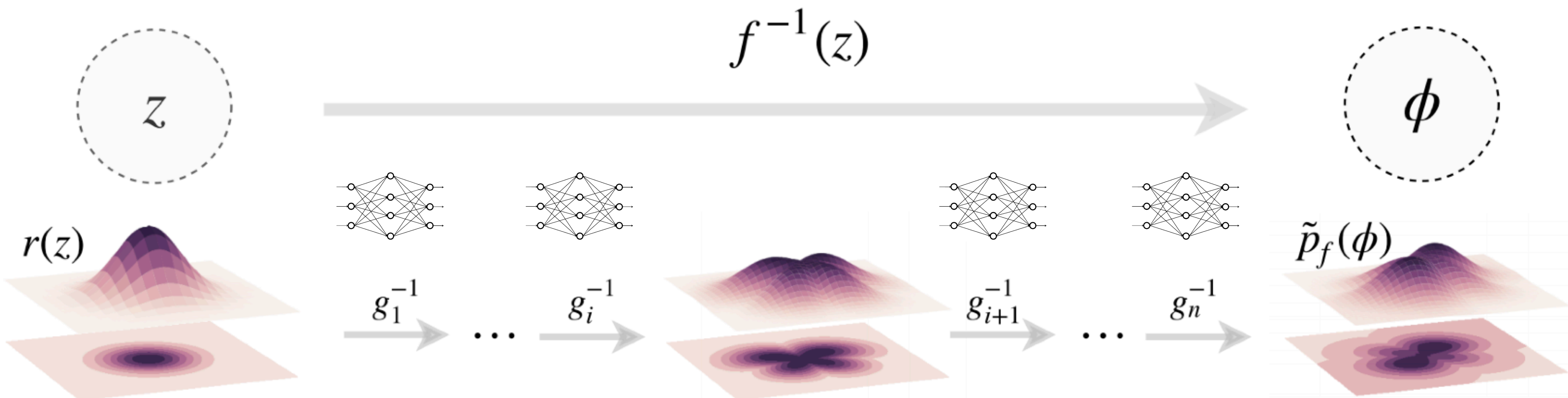
arxiv 1904.12072, 2003.06413, 2008.05456 and more.

Flow based sampling algorithm

Akio Tomiya

Trivializing map realized using neural network

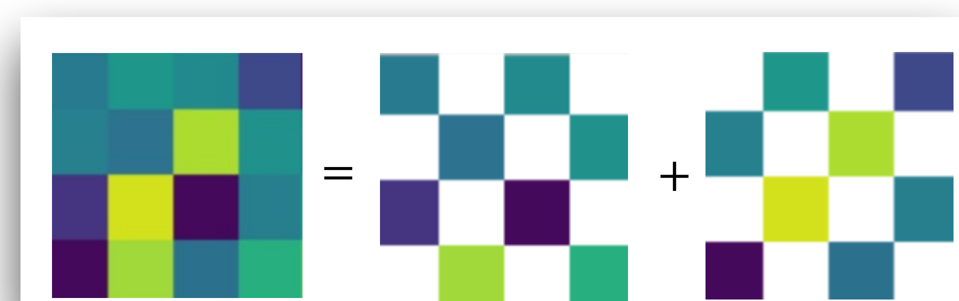
Normalizing flow? = Change of variable with **neural nets**
Tractable Jacobian is realized by checker-board technique



(a) Normalizing flow between prior and output distributions

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

Problem: Jacobian is difficult = O(V^3)
-> Introduce checker-board decomposition



Credit: Daniel Hacket (MIT)

arxiv 1904.12072, 2003.06413, 2008.05456 and more.

Flow based sampling algorithm

Akio Tomiya

Flow based ML for QFT

MIT + Deepmind + …

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

Original integral: hard

Easy

<u>Flow-based sampling algorithm</u>

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

①Embarrassedly parallel sampling
from trivial theory
(no kinetic term, no topology)

No auto-correlation
No correlation for points

②"un-trivializing map"

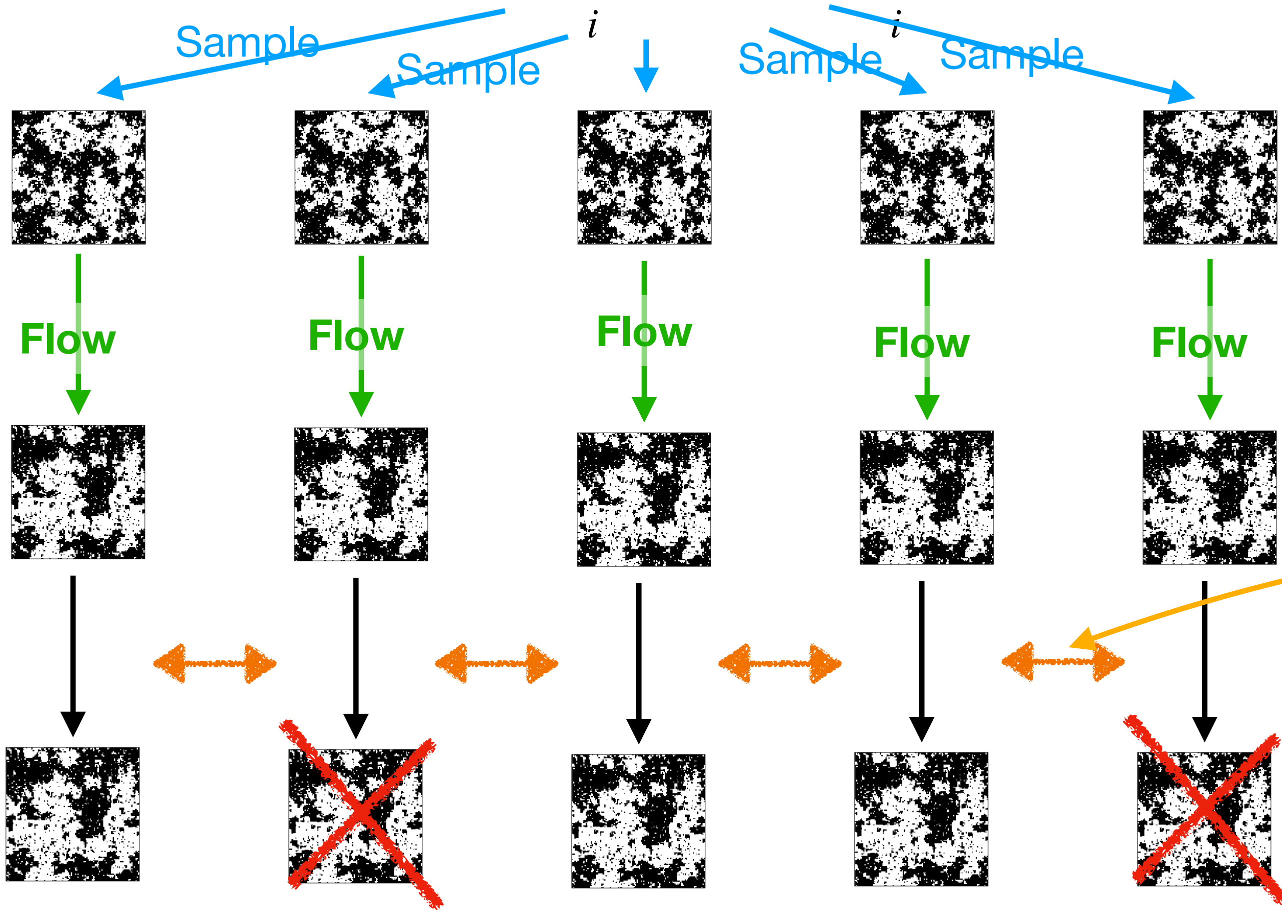
"Cooling = change of variable"
via trained neural net

No auto-correlation
Approx.correlation for points

③Metropolis-*Hastings* with

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

Only sequential



Correct correlations
Small auto-correlation

Reject
(Use left conf.)

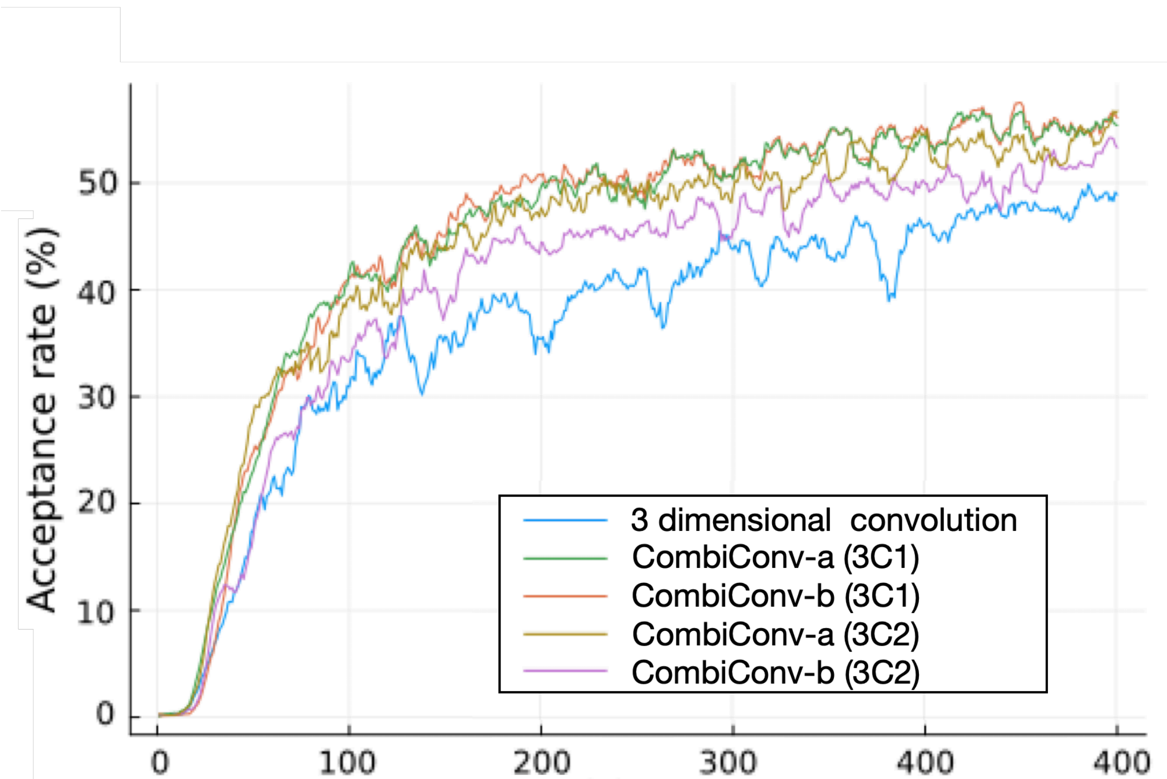
Reject
(Use left conf.)

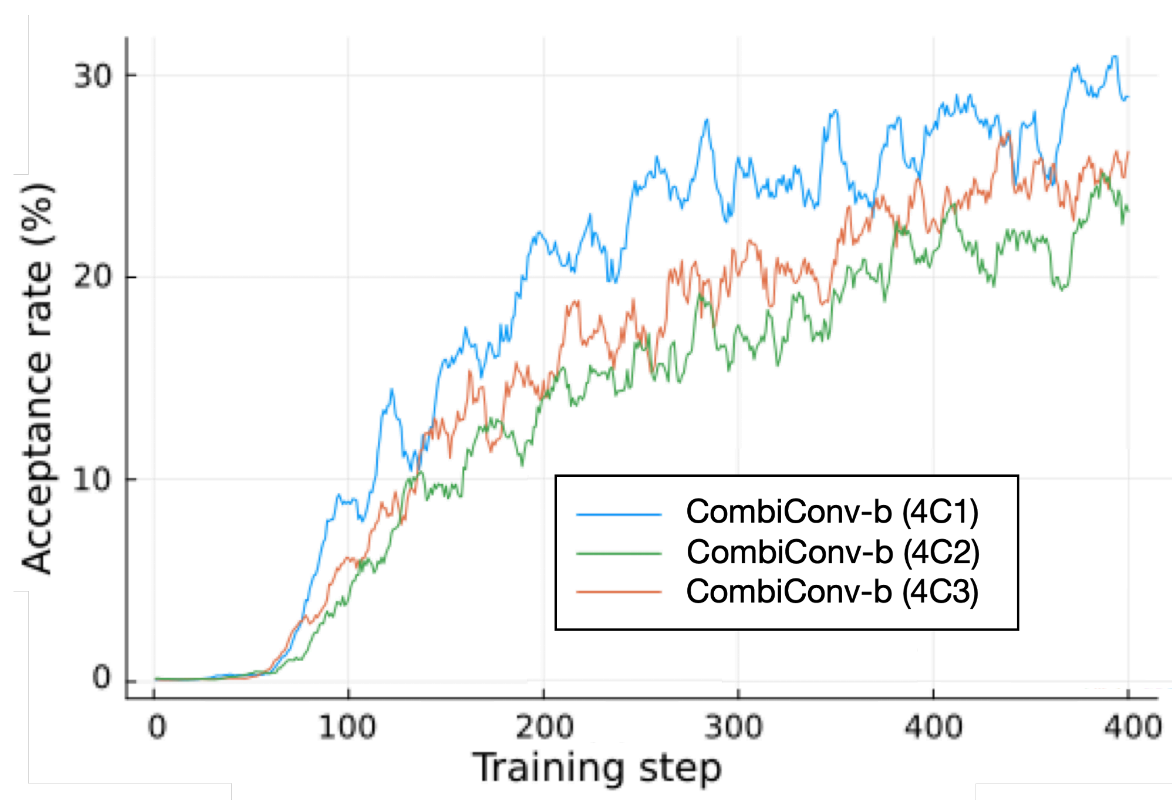
Auto-correlation
~ Rejection rate

Flow based sampling algorithm

Akio Tomiya

We make new convolutional layer for QFT in d-dim





- We implement CombiConv for flow-based sampling algorithm for d-dimensional scalar field theory on the lattice
- 3d convolution is available on GomalizingFlow.jl [1], open source implementation of flow-based sampling algorithm
- nCk $\equiv \dfrac{n!}{k!(n-k)!}$
- In 3d, the acceptance rate is improved for CombiConv compared with the conventional 3d convolution
- In 4d, it works well for any combination of lower dimensional convolution
- This works in any number of dimensions.

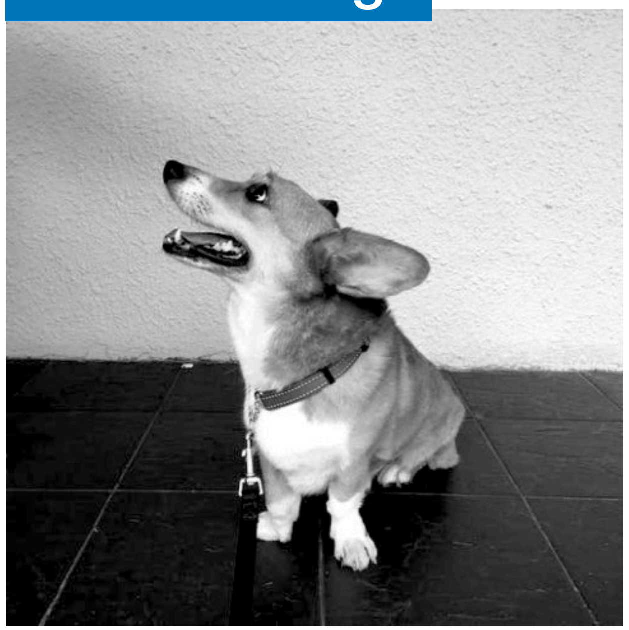
Three methods with machine learning 2/3: Gauge covariant NN + SLHMC

Convolution respects symmetry

Akio Tomiya

Convolution layer = trainable filter

Filter on image



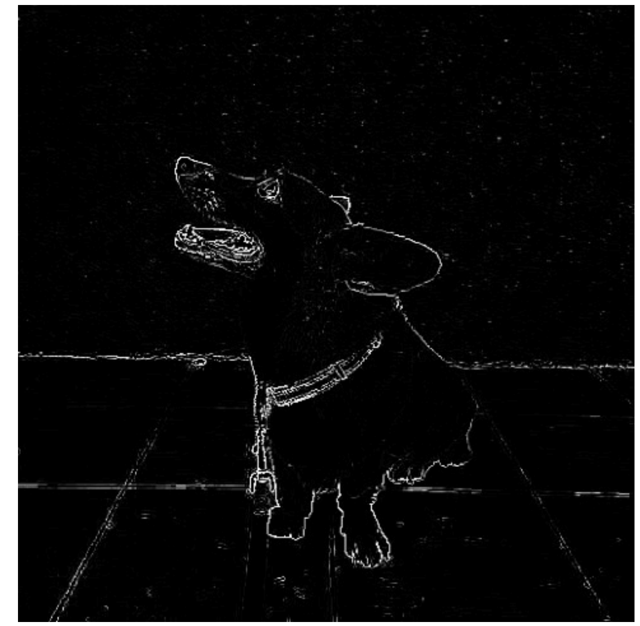
$*$

Laplacian filter

0	1	0
1	-2	1
0	1	0

(Discretization of ∂^2)

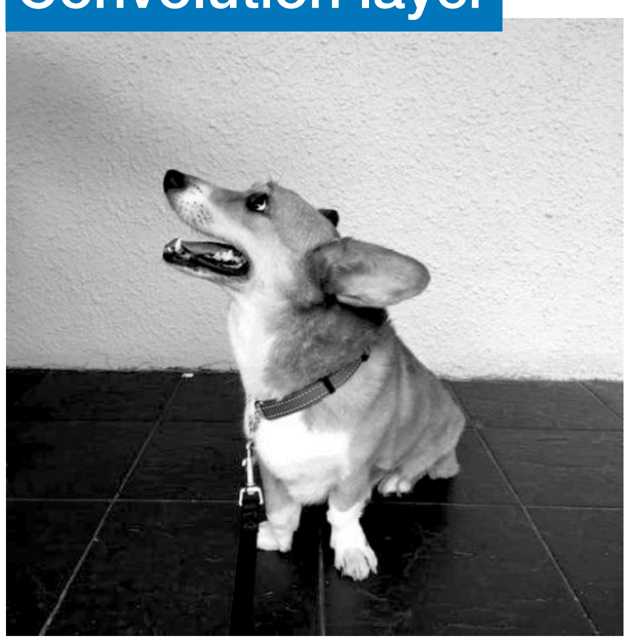
$=$



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)

...

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

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

Gaussian filter

$\frac{1}{16}$

1	2	1
2	4	2
1	2	1

Convolution respects transnational symmetry as well

Convolution respects symmetry

Akio Tomiya

Smearing = Smoothing of gauge fields

Eg.

Coarse image

Gaussian filter

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 1 \\ 1 & 2 & 1 \end{bmatrix}$$

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

Smearing ~ smoothing

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)

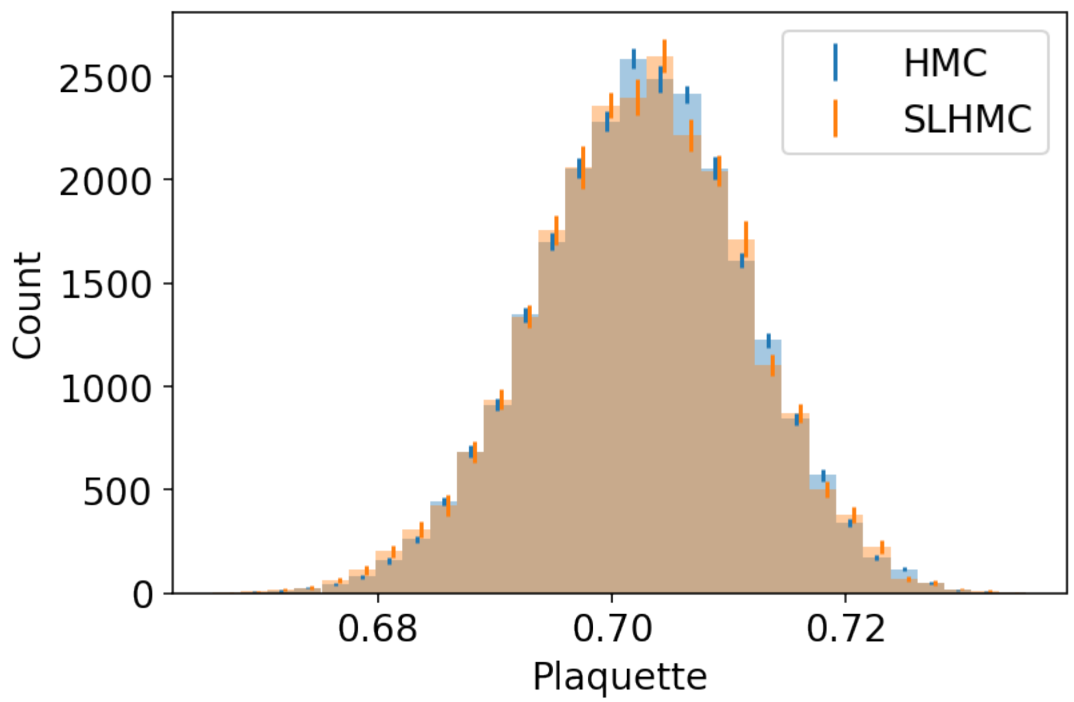
As same as the convolution, we can train weights.

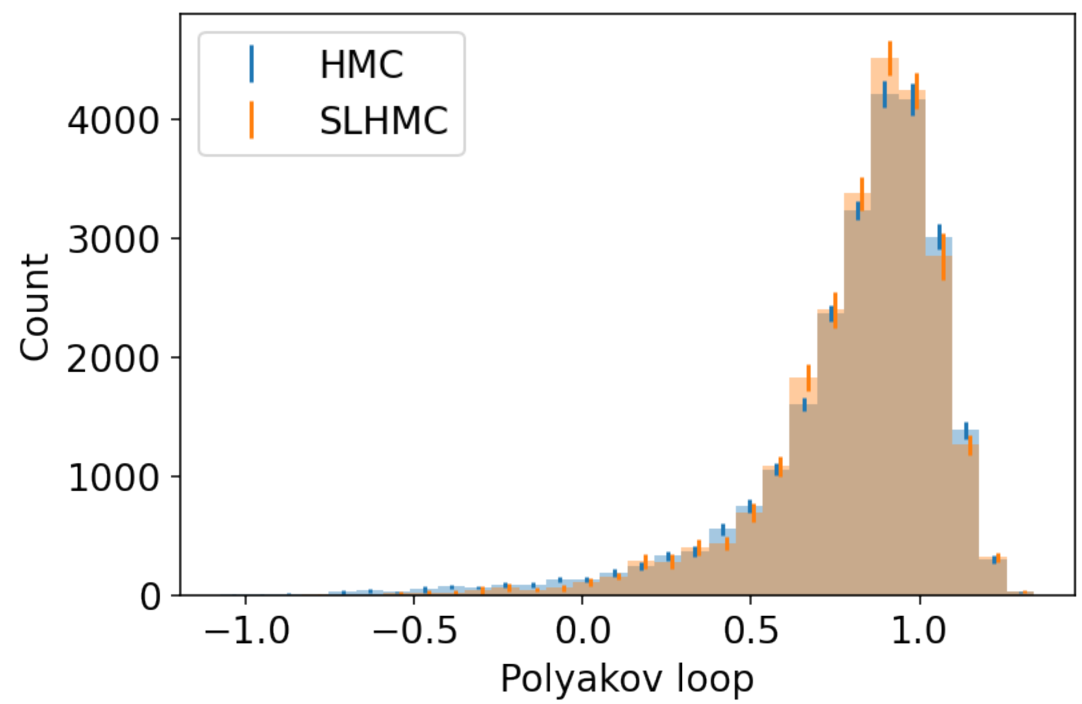
Application for the Full QCD in 4d

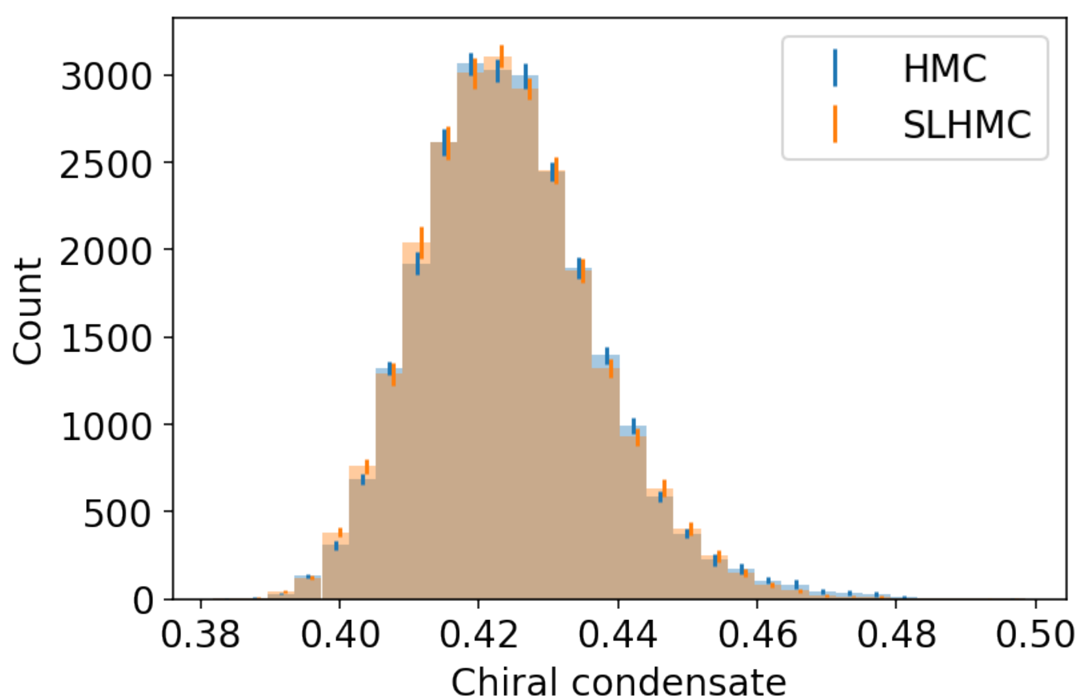
Akio Tomiya

Results are consistent with each other

AT Y. Nagai arXiv: 2103.11965







Expectation value

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

Acceptance = 40%

Three methods with machine learning 3/3: Transformer for physical system

Transformer and Attention

Akio Tomiya

Attention layer used in Transformers (GPT, Bard)

arXiv:1706.03762

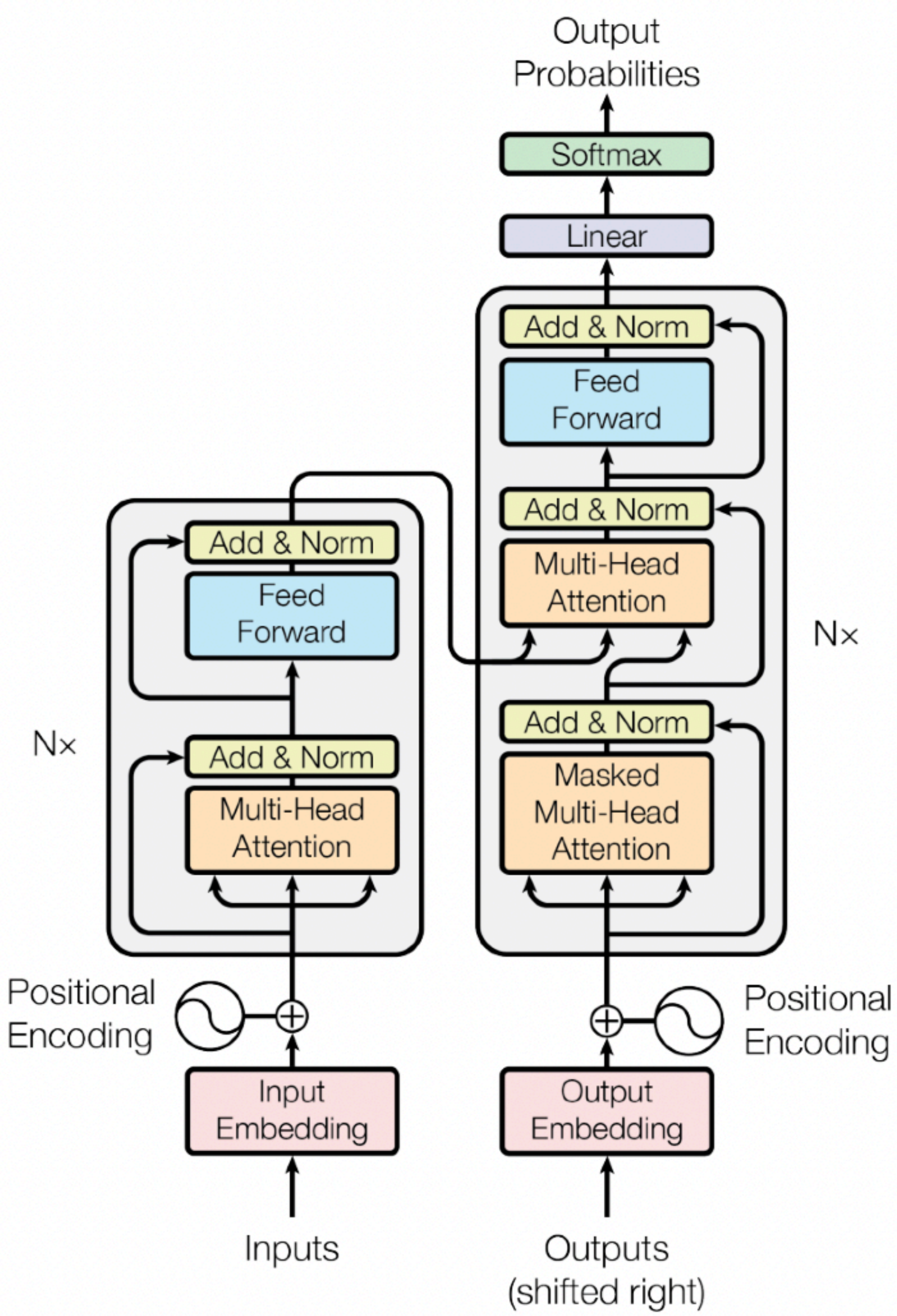


Figure 1: The Transformer - model architecture.

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

Transformer and Attention

Akio Tomiya

Attention layer can capture non-local correlations

arXiv:1706.03762

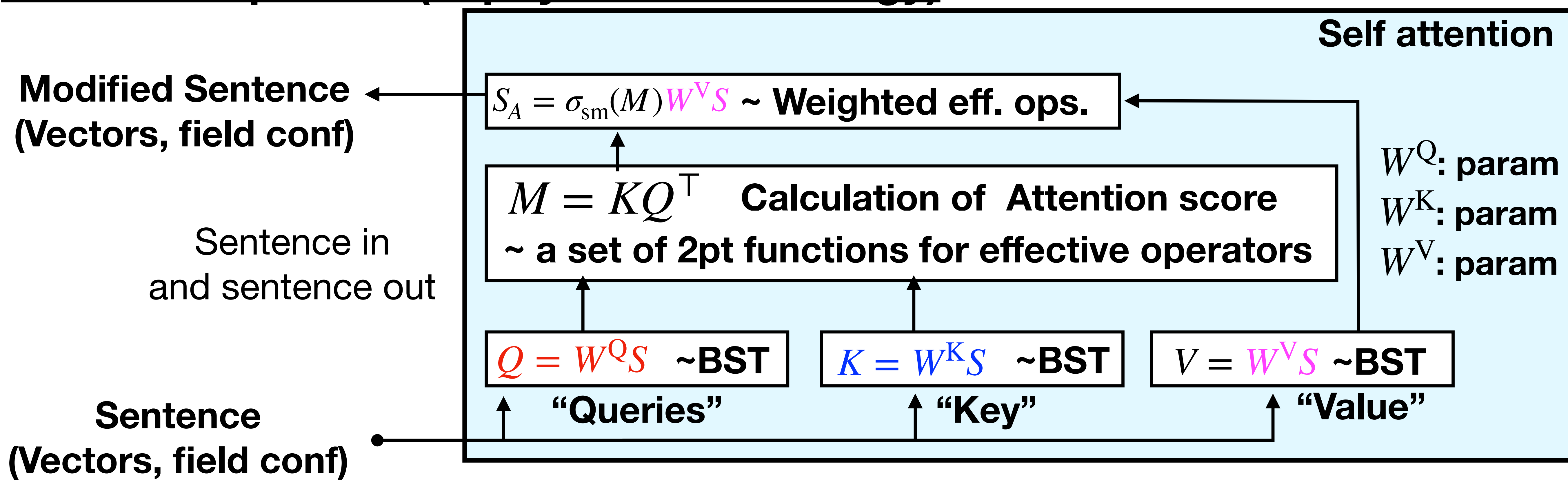
Modifier in language can be non-local

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

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

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

Schematic picture (in physics terminology)



Self-learning Monte-Carlo

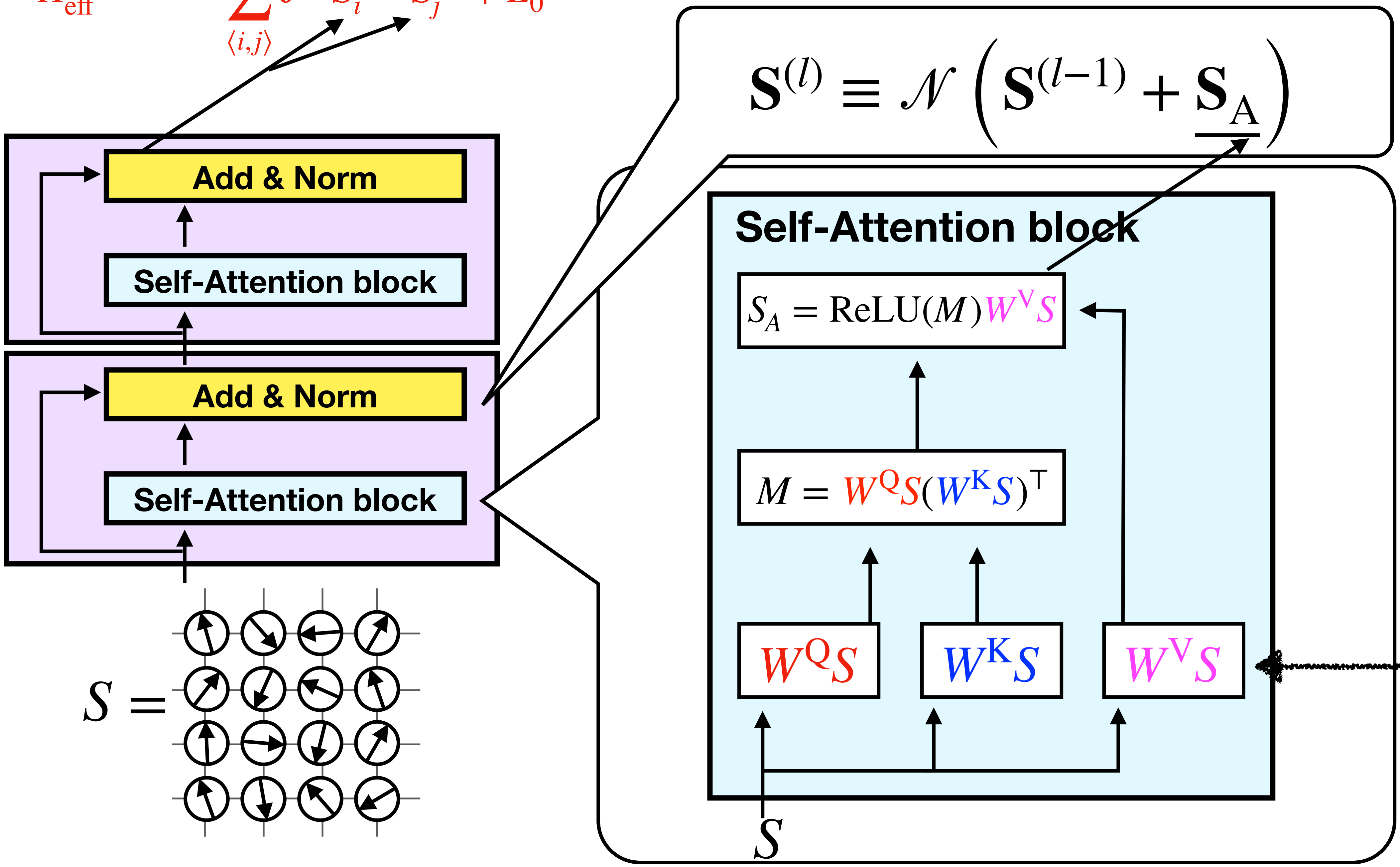
Physically equivariant Attention layer/Transformer

arXiv: 2306.11527.

We can construct effective hamiltonian with output of our Attention layer because "output of Attention = smoothened fields with non-local correlation"

$$H_{\text{eff}}^{\text{Linear}} = -\sum_{\langle i,j\rangle} J^{\text{eff}} \mathbf{S}_i^{\text{eff}} \cdot \mathbf{S}_j^{\text{eff}} + E_0$$

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



Averaged spins
Rot. equivariant
Trsl. equivariant
trainable!

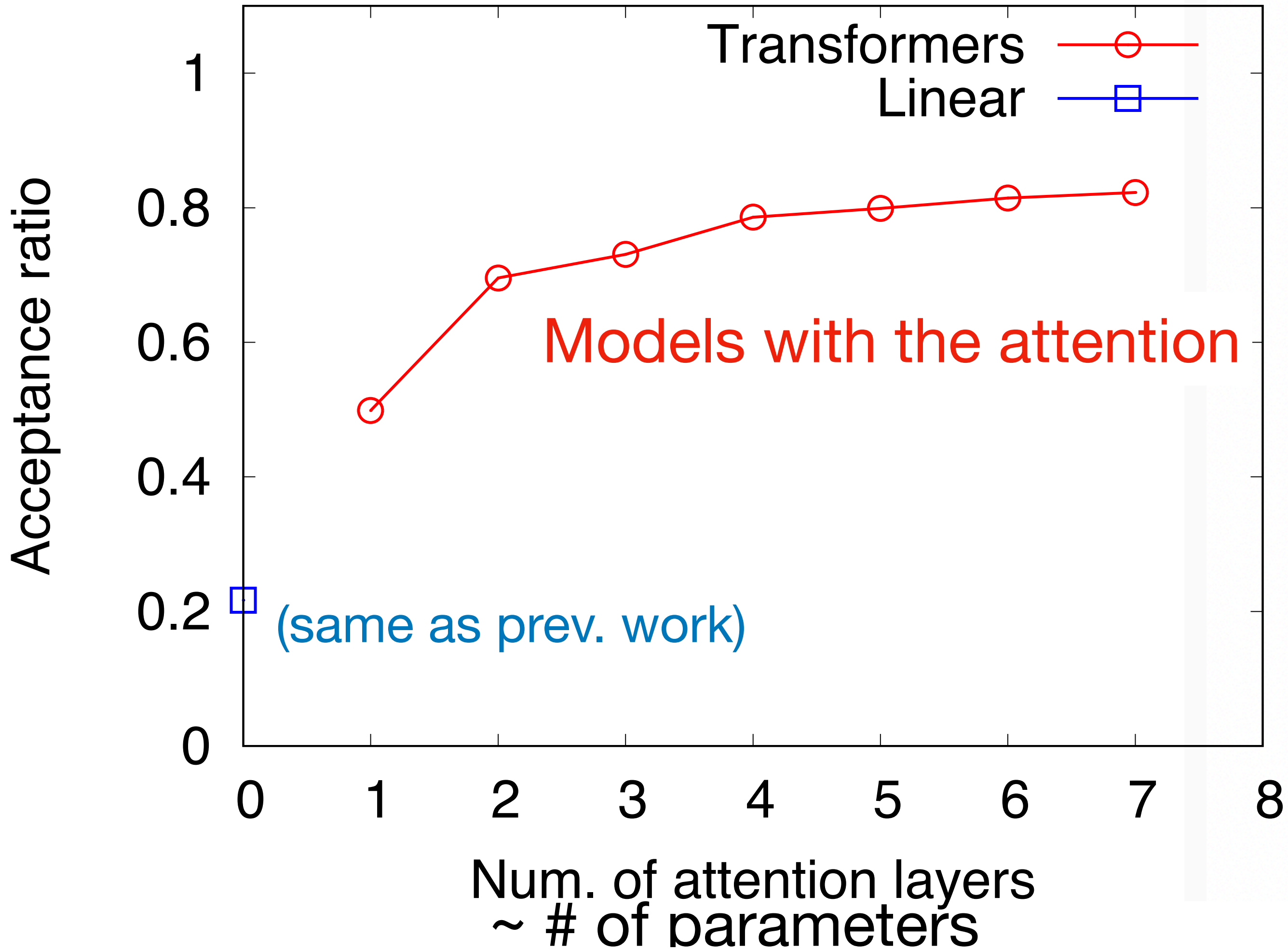
Averaged spins
Rot. equivariant
Trsl. equivariant
trainable!

Transformer and Attention

Akio Tomiya
arXiv: 2306.11527 + update

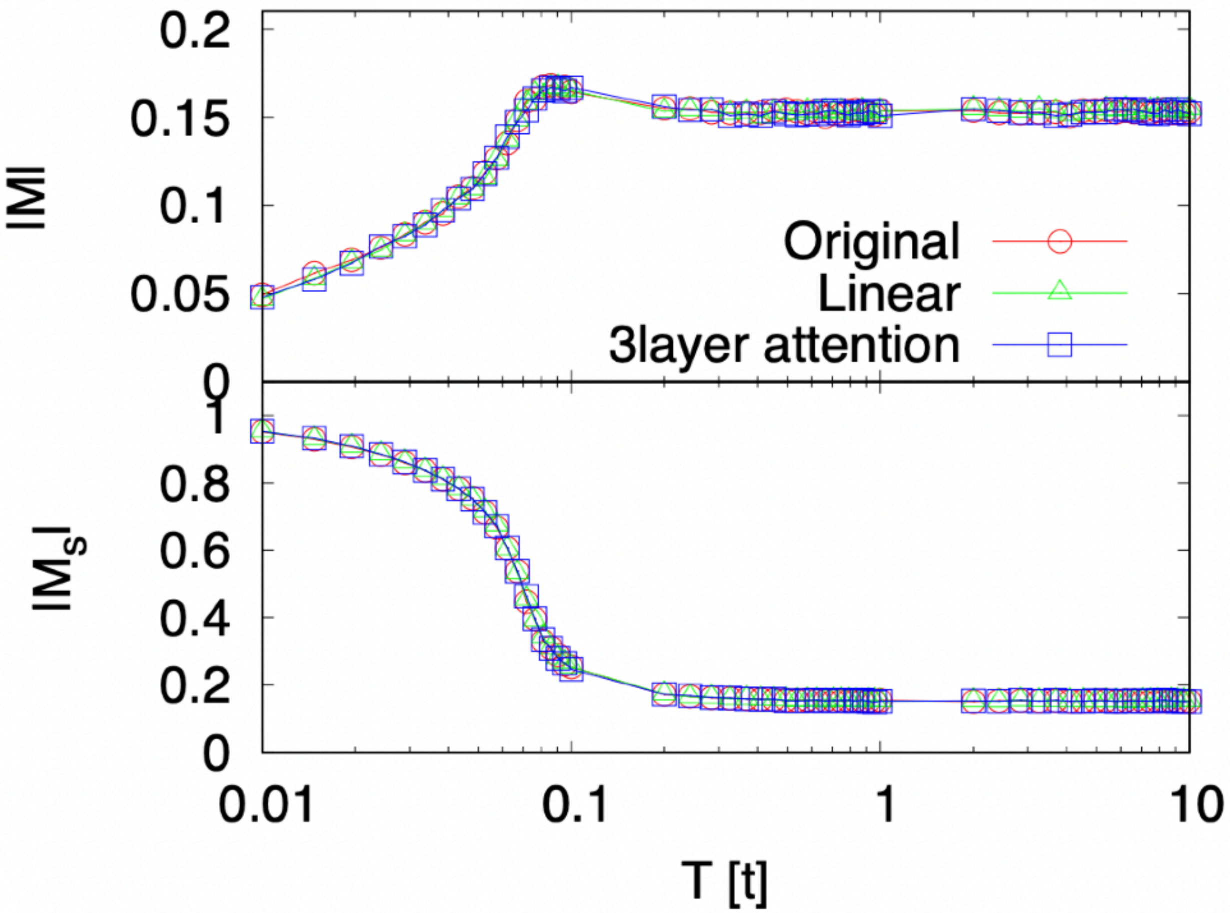
Application to O(3) spin model with fermions (Kondo model)

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)

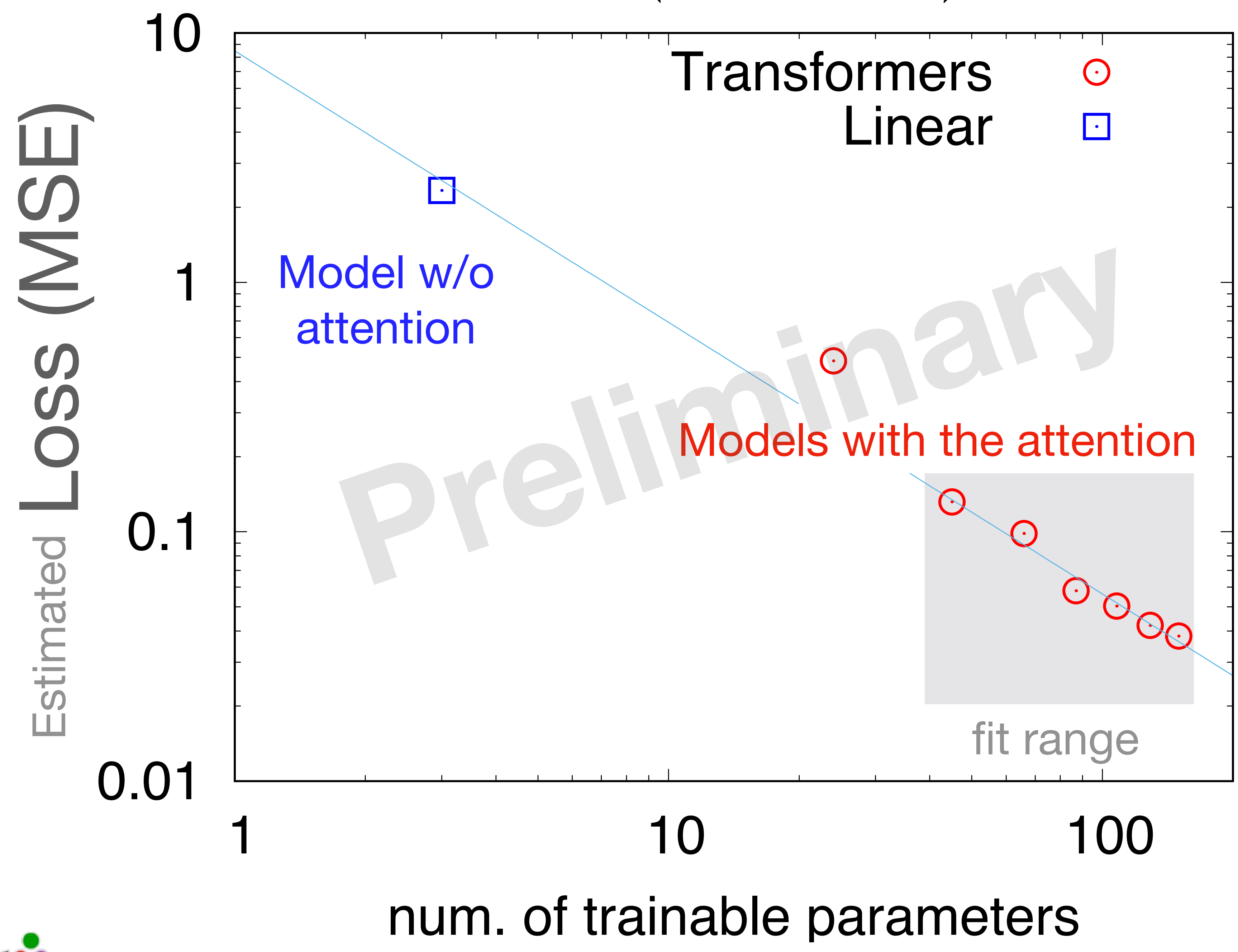
Nx=Ny=6
(Lattice sites)

Transformer and Attention

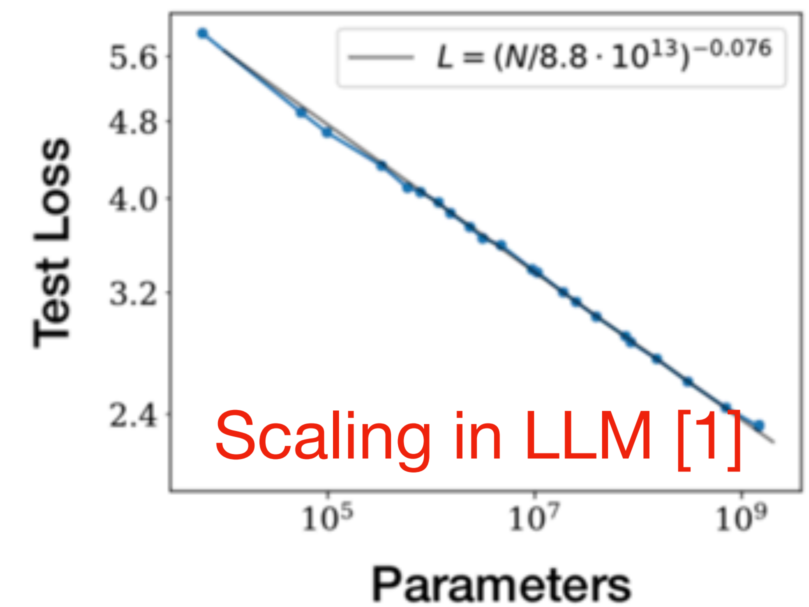
Loss function shows Power-type scaling law as LLM

arXiv: 2306.11527 + update

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



(1 layer ~ 30 parameters)



Line is just for guiding eyes (no meaning)

fit ~(7.1/x)^(1.1)

Summary

Machine learning + Computational Physics

- Machine learning is useful for natural science/physics/Lattice QCD
  - Multi-dimensional integration is done by MCMC
- MCMC candidate can be made by Machine learning
  - Flow-based sampling algorithm
  - Self-learning HMC + Gauge covariant neural network
  - Transformer for physical system (not gauge theory yet)
    - Scaling law for a Transformer for physical system
- ML + expert knowledge of computational physics/LatticeQCD is important

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!