東京大学情報基盤センター
INFORMATION TECHNOLOGY CENTER, THE UNIVERSITY OF TOKYO

MLPhys 学術変革領域研究（A）学習物理学の創成
Foundation of "Machine Learning Physics"

Self-learning Monte Carlo method with equivariant transformer

The Information Technology Center,
The University of Tokyo

UTokyo

Yuki Nagai

Yuki Nagai and Akio Tomiya, “Self-Learning Monte Carlo with Equivariant Transformer”, J. Phys. Soc. Jpn. 93, 114007 (2024) Editors’ choice

Introduction

About me

Yuki Nagai

Born in Hokkaido, the northern island in Japan

2005/03 B. Eng. at Department of Applied Physics, Hokkaido University

2010/03 Ph.D at Department of Physics, the University of Tokyo

2010/04-2024/01 Scientist -> Senior Scientist, Japan Atomic Energy Agency

2024/02 Associate Professor in The University of Tokyo

2016/11-2017/10
Visiting Scholar, Department of Physics,
Massachusetts Institute of Technology, USA

2018-2023 Visiting researcher in RIKEN AIP

I have used supercomputers in JAEA and UTokyo

Condensed matter theory
Superconductivity,
Material science
Machine-learning and Physics

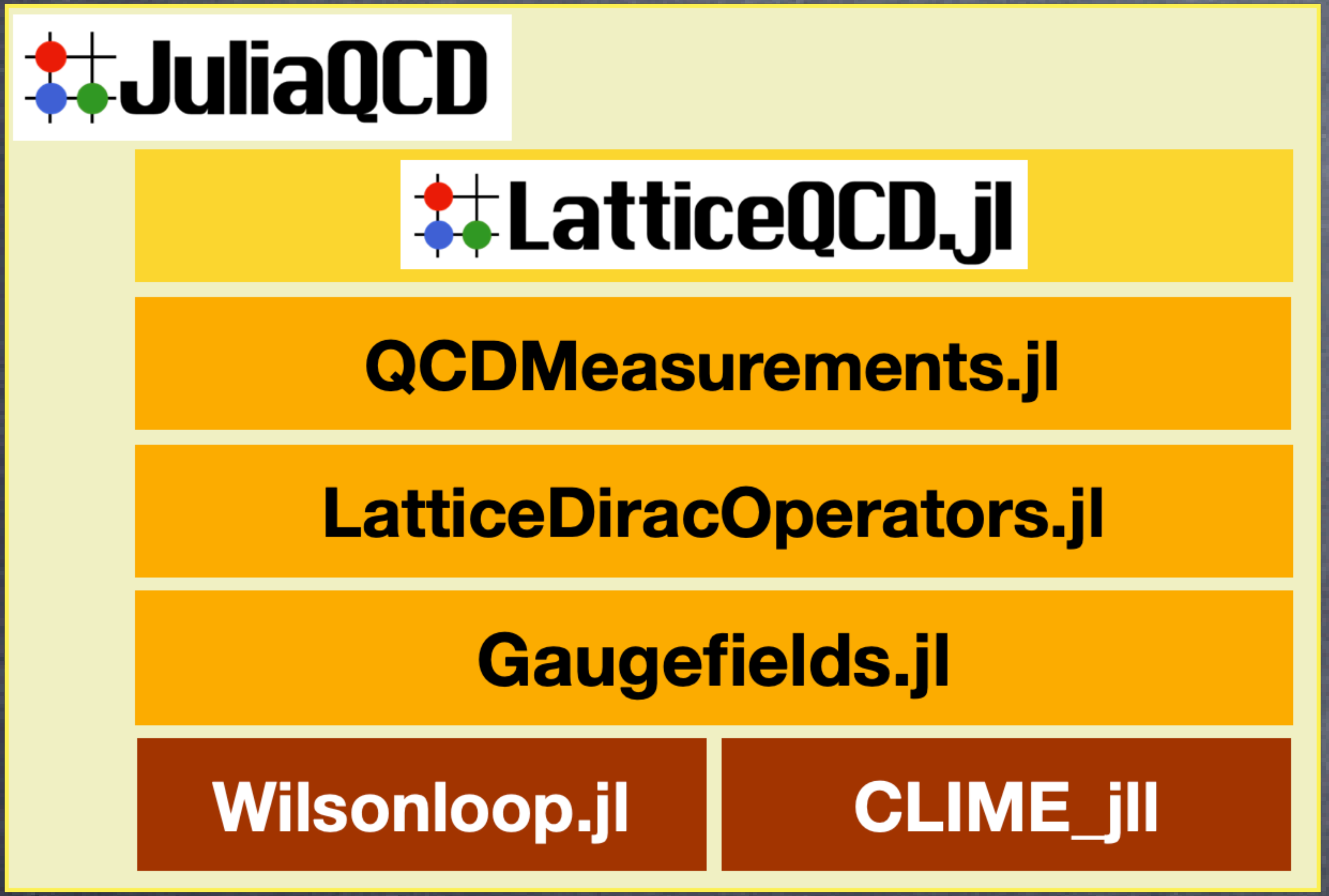
東京大学情報基盤センター
INFORMATION TECHNOLOGY CENTER, THE UNIVERSITY OF TOKYO

Lattice QCD code for generic purpose

Open source LQCD code in Julia Language

JuliaQCD

Akio Tomiya and YN


JuliaQCD

LatticeQCD.jl

QCDMeasurements.jl

LatticeDiracOperators.jl

Gaugefields.jl

Wilsonloop.jl

CLIME_jll

Machines: Laptop/desktop/Jupyter/Supercomputers

Functions: SU(Nc)-heatbath, (R)HMC, Self-learning HMC, SU(Nc) Stout
Dynamical Staggered, Dynamical Wilson, Dynamical Domain-wall
Measurements

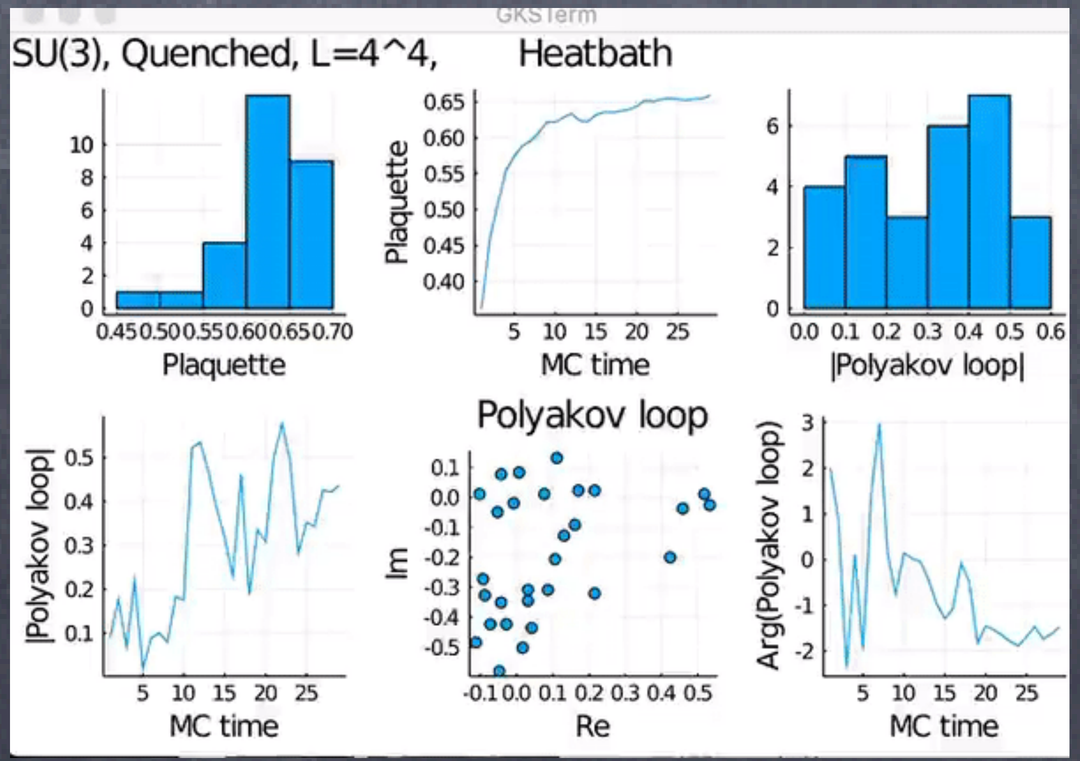
Start LQCD in **5 min**

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

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

arXiv:2409.03030

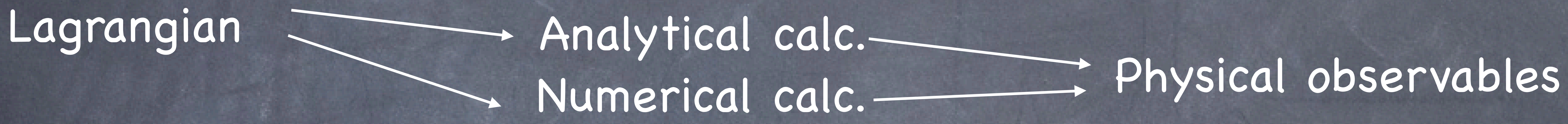
-> Next Dr. Akio Tomiya's talk



easy to run!

Julia looks like python
but fast like c or fortran

Machine and Condensed Matter physicists in high-energy physics



Lagrangian
me
machine
Physical observables

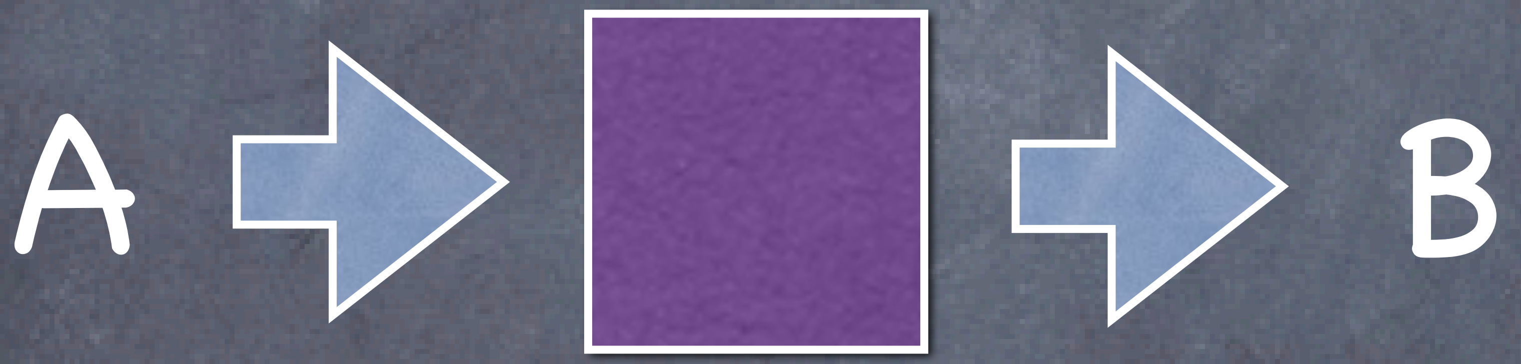
Machine and condensed matter physicists calculate physical observables without understanding any Lagrangian…?

Speedup with machine learning

In field of machine learning

Image recognition, AI chat etc.

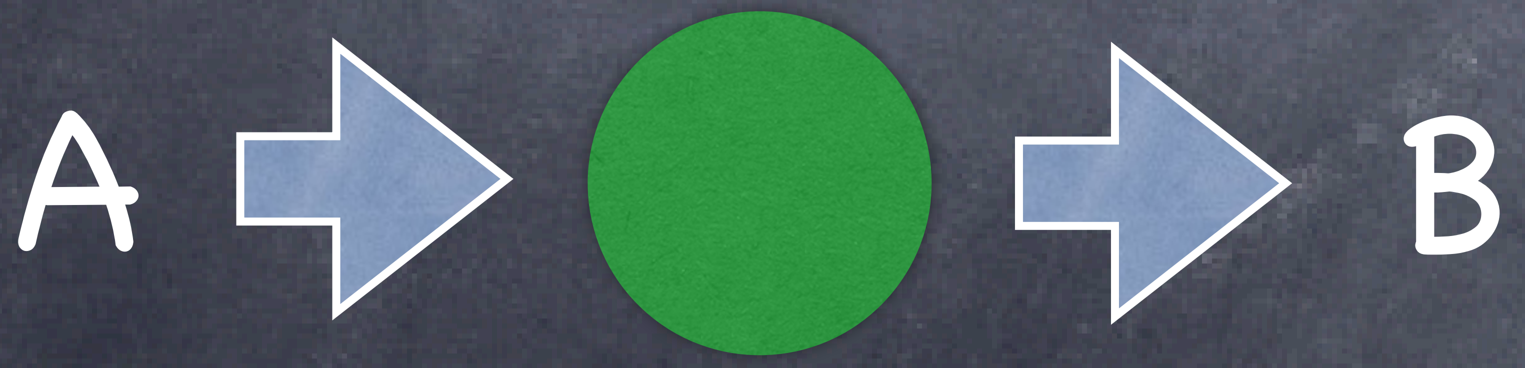
We do not have a theory of these. But the machine can imitate these



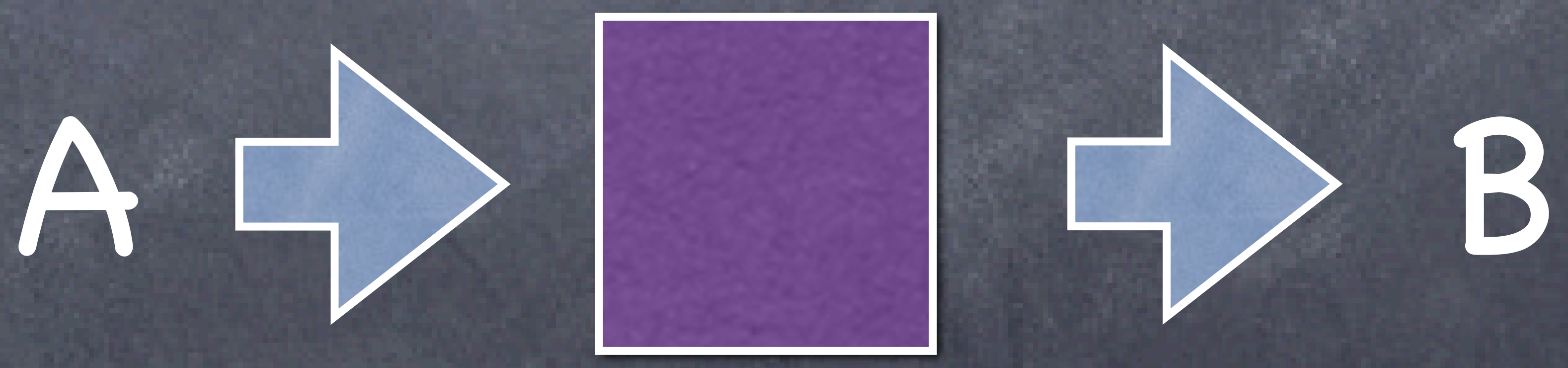
🟪 **:We do not have a theory**

How to use machine learning in simulations?

Known heavy task is replaced



heavy task from a concrete theory



effective model

We replace the heavy tasks by neural networks

Self learning Monte Carlo

Self-learning Monte Carlo

We calculate a partition function $Z=\int \exp(-S)$ or $\Sigma \exp(-\beta H)$

With the use of Monte Carlo method, we can calculate physical variables

Sometimes, the computational cost is heavy.

Configurations → **Heavy tasks** → Boltzmann weight

Configurations → **effective model** → Boltzmann weight

Spins **Electrons** **Atoms, molecules** **Lattice QCD**

To propose a new configuration, we use the effective model

Exact MCMC simulations

You want to do **MCMC** simulations with very heavy computational cost

The effective model to imitate the original model might be useful

A
B
heavy
A
B′
effective model

if the effective model is not good, B′ is not good

How long do you have to train the model?

By using the self-learning Monte Carlo method,

the output with an effective model becomes **exact**

What is the self-learning Monte Carlo?

common simulation with machine learning:

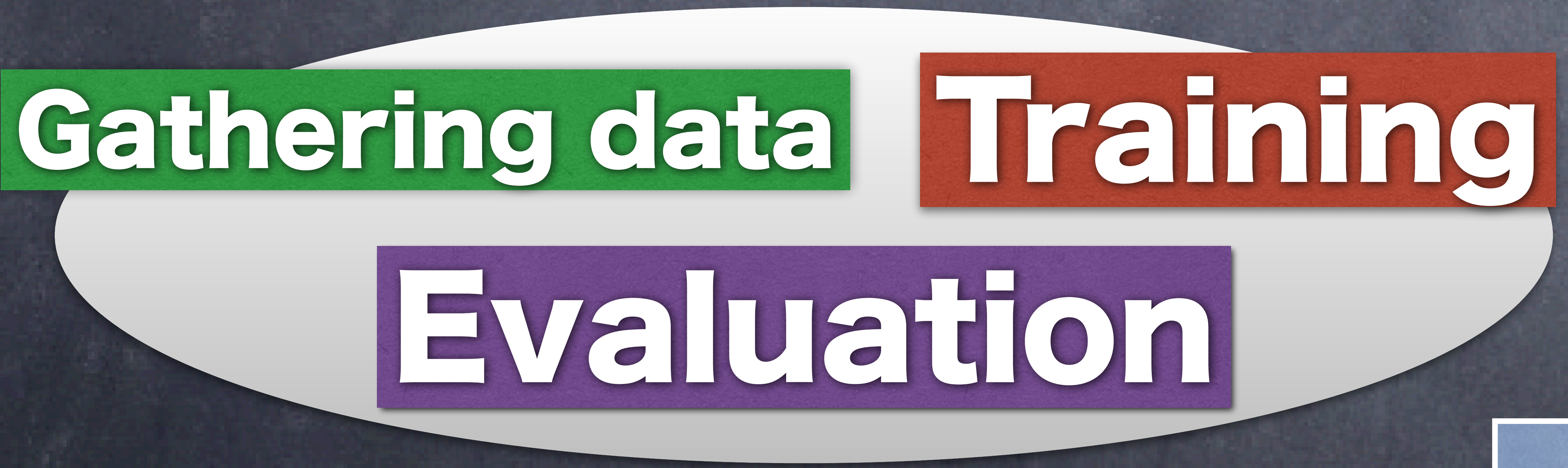
Machine learning

Simulation

Gathering data → Training → Evaluation

not good? gather more data

Self-learning Monte Carlo method



Gathering data

Training

Evaluation

We do three steps in same simulations

Num. of training data is drastically reduced (1/10) because of efficient sampling

Self-learning Monte Carlo

Spin systems

J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Phys. Rev. B 95, 041101(R) (2017)

H. Kohshiro and YN,,
“Effective Ruderman–Kittel–Kasuya–Yosida-like Interaction in Diluted Double-exchange Model: Self-learning Monte Carlo Approach”,
J. Phys. Soc. Jpn. 90, 034711 (2021)

YN and A. Tomiya, “Self-learning Monte Carlo with equivariant Transformer”, J. Phys. Soc. Jpn. 93, 114007 (2024)

Fermion+classical spins

Electrons

YN, H. Shen, Y. Qi, J. Liu, and L. Fu
“Self-learning Monte Carlo method: Continuous-time algorithm”,
Physical Review B 96, 161102(R) (2017) ***Editors’ Suggestion***

YN, M. Okumura, A. Tanaka
“Self-learning Monte Carlo method with Behler-Parrinello neural networks”,
Phys. Rev. B **101**, 115111 (2020)

Continuous time Quantum Monte Carlo

Atoms/molecules

Machine-learning MD

YN, M. Okumura, K. Kobayashi, and M. Shiga,
“Self-learning Hybrid Monte Carlo: A First-principles Approach”,
Phys. Rev. B **102**, 041124(R) (2020)

K. Kobayashi, YN, M. Itakura, and M. Shiga,
“Self-learning hybrid Monte Carlo method for isothermal–isobaric ensemble: Application to liquid silica”,
J. Chem. Phys. **155**, 034106 (2021)

YN, Yutaka Iwasaki, Koichi Kitahara, Yoshiki Takagiwa, Kaoru Kimura, Motoyuki Shiga, “High-Temperature Atomic Diffusion and Specific Heat in Quasicrystals”, Phys. Rev. Lett. 132, 196301 (2024)

Bo Thomsen, YN, Keita kobayashi, Ikutaro Hamada, and Motoyuki Shiga, “Self-learning path integral hybrid Monte Carlo with mixed ab initio and machine learning potentials for modeling nuclear quantum effects in water”, J. Chem. Phys. 161, 204109 (2024)

Lattice QCD

SU(N) Gauge theory on the lattice

YN, Akinori Tanaka, Akio Tomiya,
“Self-learning Monte-Carlo for non-abelian gauge theory with dynamical fermions”,
Phys. Rev. D 107, 054501 (2023)

YN and Akio Tomiya,
“Gauge covariant neural network for 4 dimensional non-abelian gauge theory”,
arXiv:2103.11965

Self-learning Monte Carlo

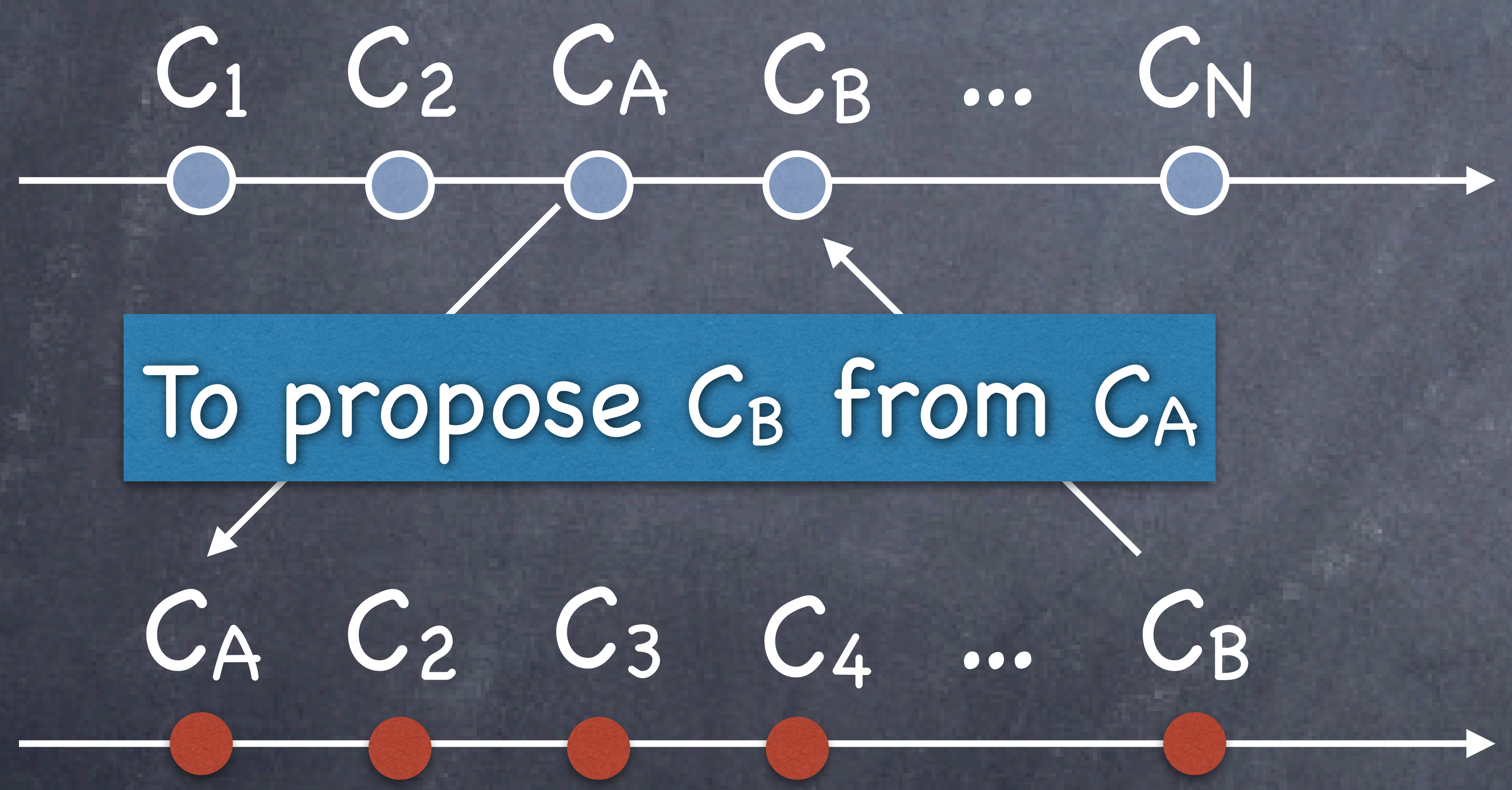
Self-learning Monte Carlo method (SLMC)

Self-learning Hybrid Monte Carlo method (SLHMC)

To speed up the Markov Chain Monte Carlo (MCMC) simulations

SLMC

Markov chain with the probability W(C)

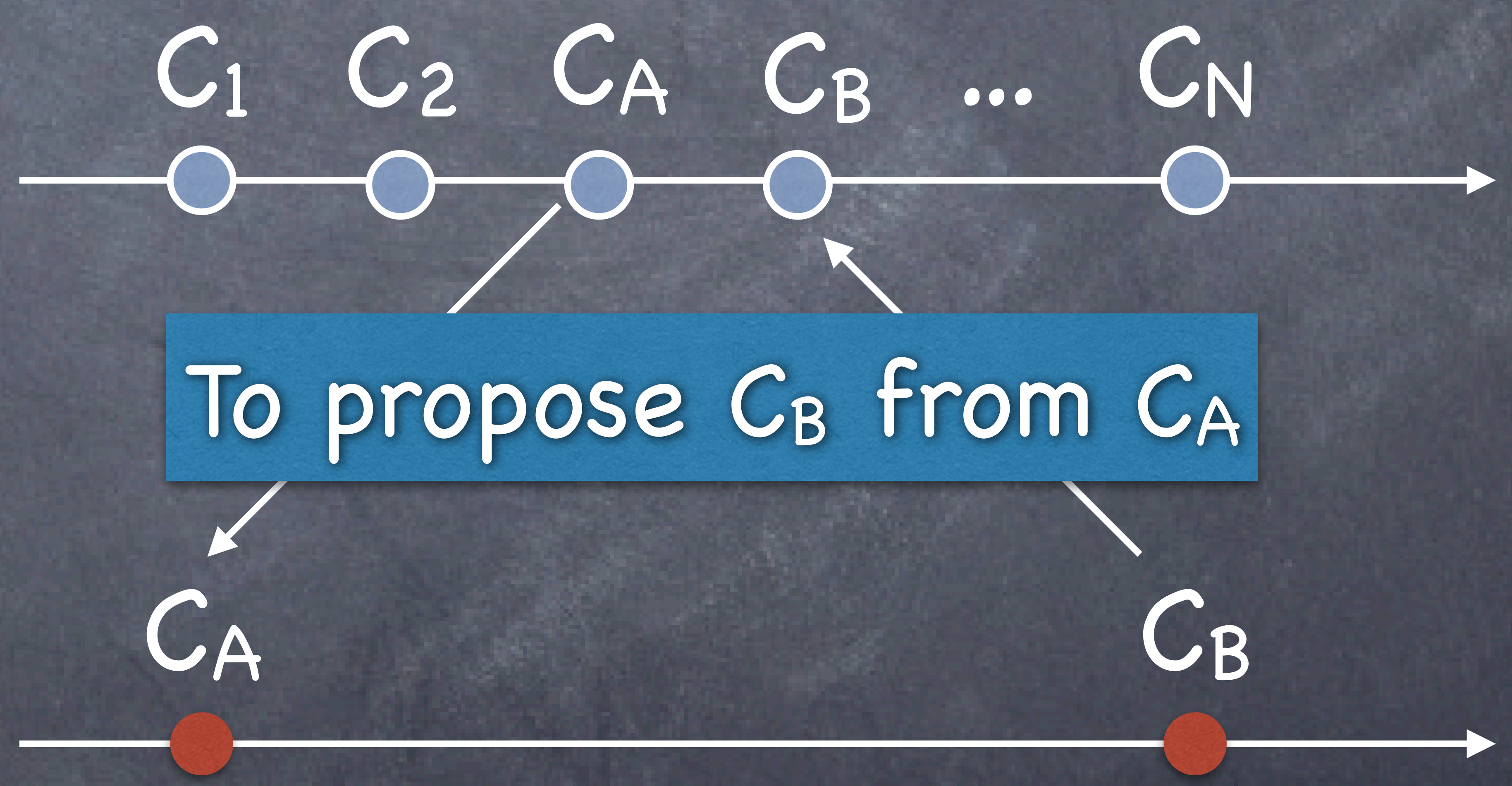


To propose C_B from C_A

Another Markov chain with the probability W'(C)

SLHMC

Markov chain with the probability W(C)



To propose C_B from C_A

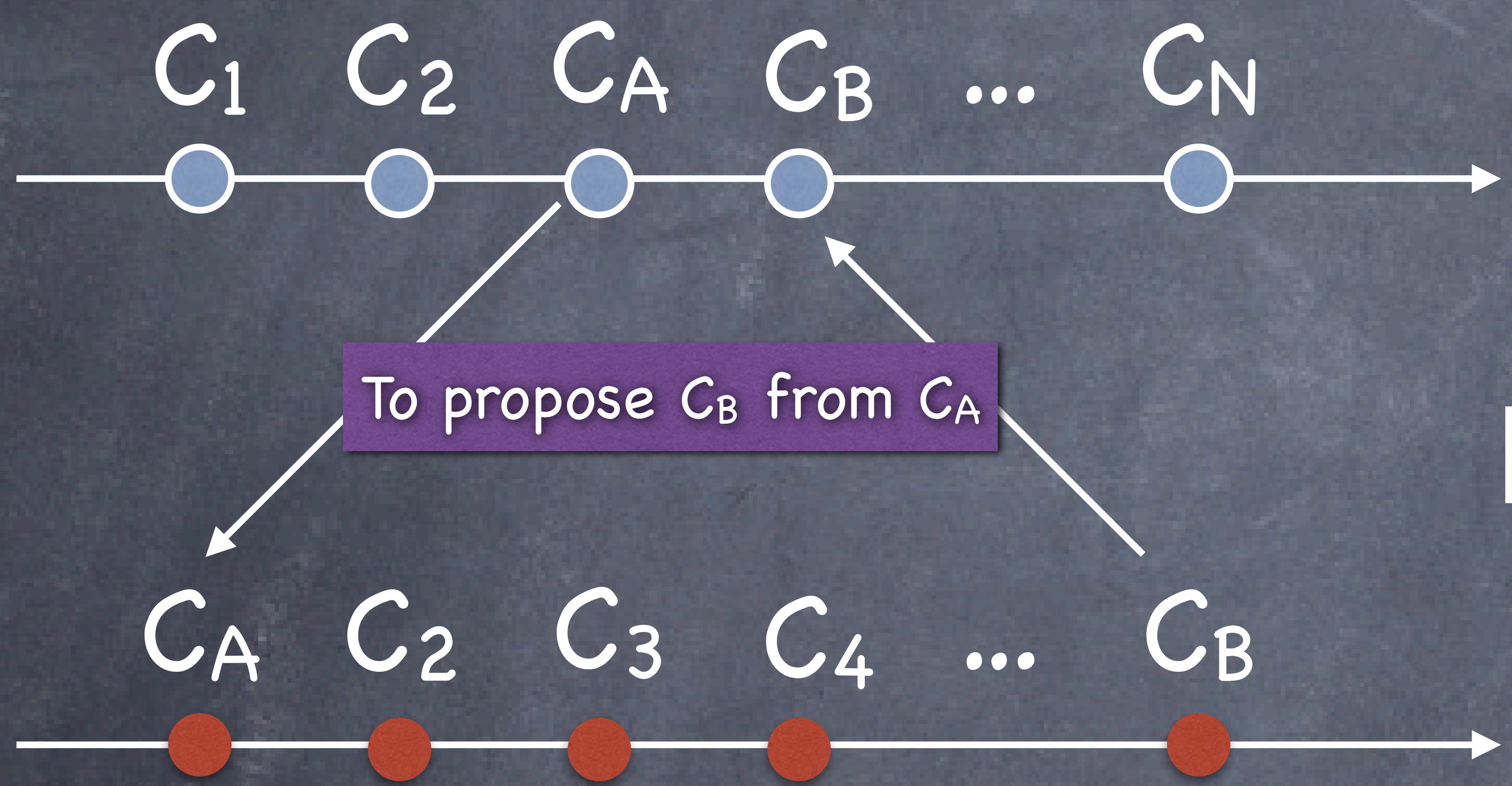
Machine learning molecular dynamics

Machine learning techniques are used for proposing new configuration!

J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Phys. Rev. B 95, 041101(R) (2017)

Concept of SLMC

Markov chain with the probability W(C)



To propose C_B from C_A

Another Markov chain with the probability W'(C)

Acceptance ratio for the Metropolis-Hastings algorithm

$$A(C_B, C_A) = \min\left(1, \frac{W(C_B)}{W(C_A)} \frac{g(C_A | C_B)}{g(C_B | C_A)}\right)$$

$g(C_B|C_A)$:Proposal probability

$$A(C_B, C_A) = \min\left(1, \frac{W(C_B)}{W(C_A)} \frac{W'(C_A)}{W'(C_B)}\right)$$

If W'(C)=W(C),
the acceptance ratio is **one**!

If the computational cost of the proposal Markov chain is small,
we can speed up the simulation

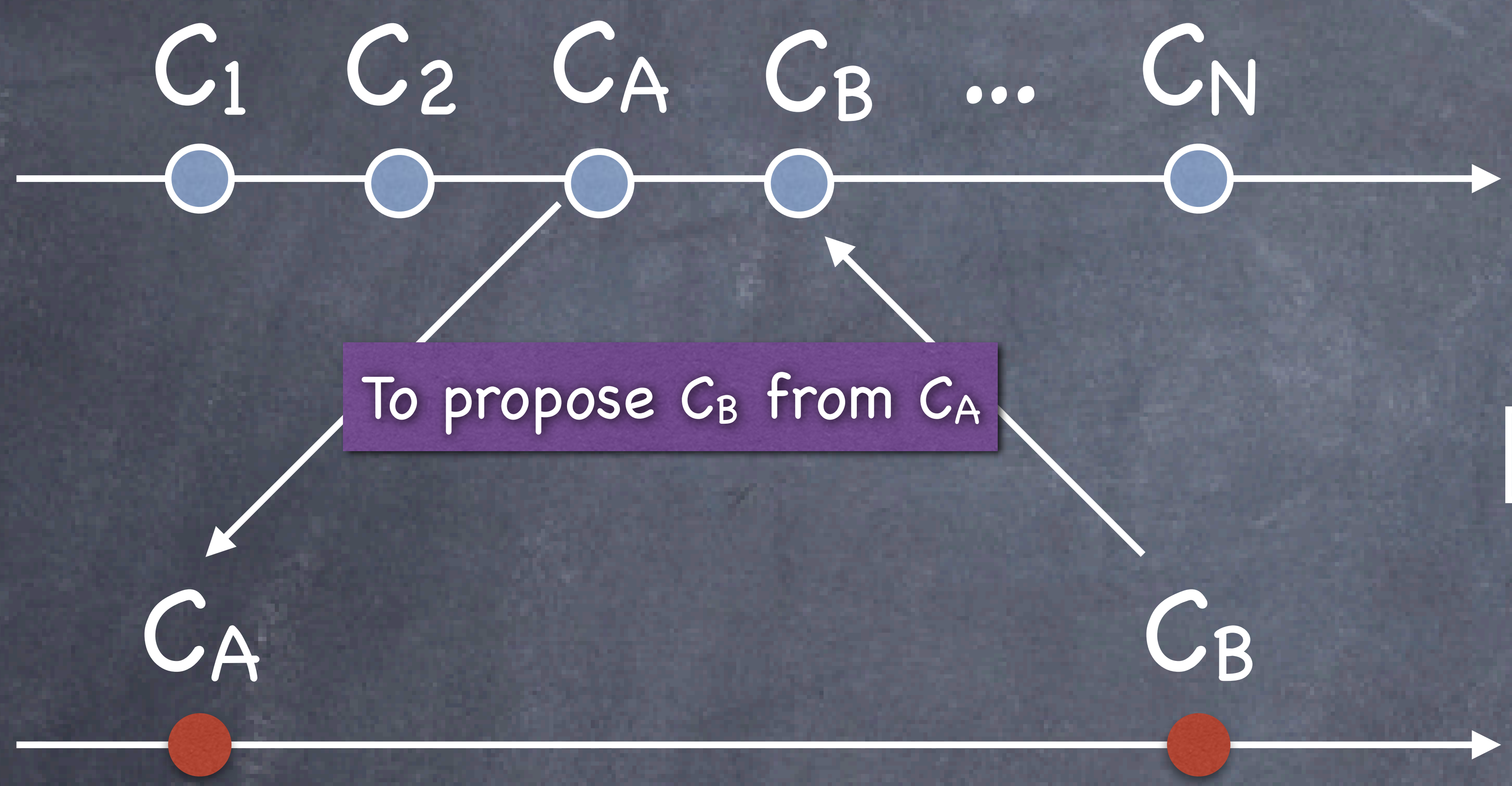
How to construct the Markov chain with W'(C)? ->Machine learning technique!

$W(C) = \exp(-\beta H(C)) \to W'(C) = \exp(-\beta H_{\mathrm{eff}}(C))$ We construct the effective Hamiltonian

YN, M. Okumura, K. Kobayashi, and M. Shiga, Phys. Rev. B 102, 041124(R) (2020)

Concept of SLHMC

Markov chain with the probability W(C)



To propose C_B from C_A

Machine learning molecular dynamics (MLMD)

Acceptance ratio for the Metropolis-Hastings algorithm in Hybrid Monte Carlo

$$A(C_B, C_A) = \min\left(1, \frac{W(C_B)}{W(C_A)}\right)$$

if the MD is time-reversal symmetric

MLMD conserves the energy of the effective model

MLMD DOES NOT conserve the energy of the original model

If the MD conserves the energy of the original model the acceptance ratio is **one**!

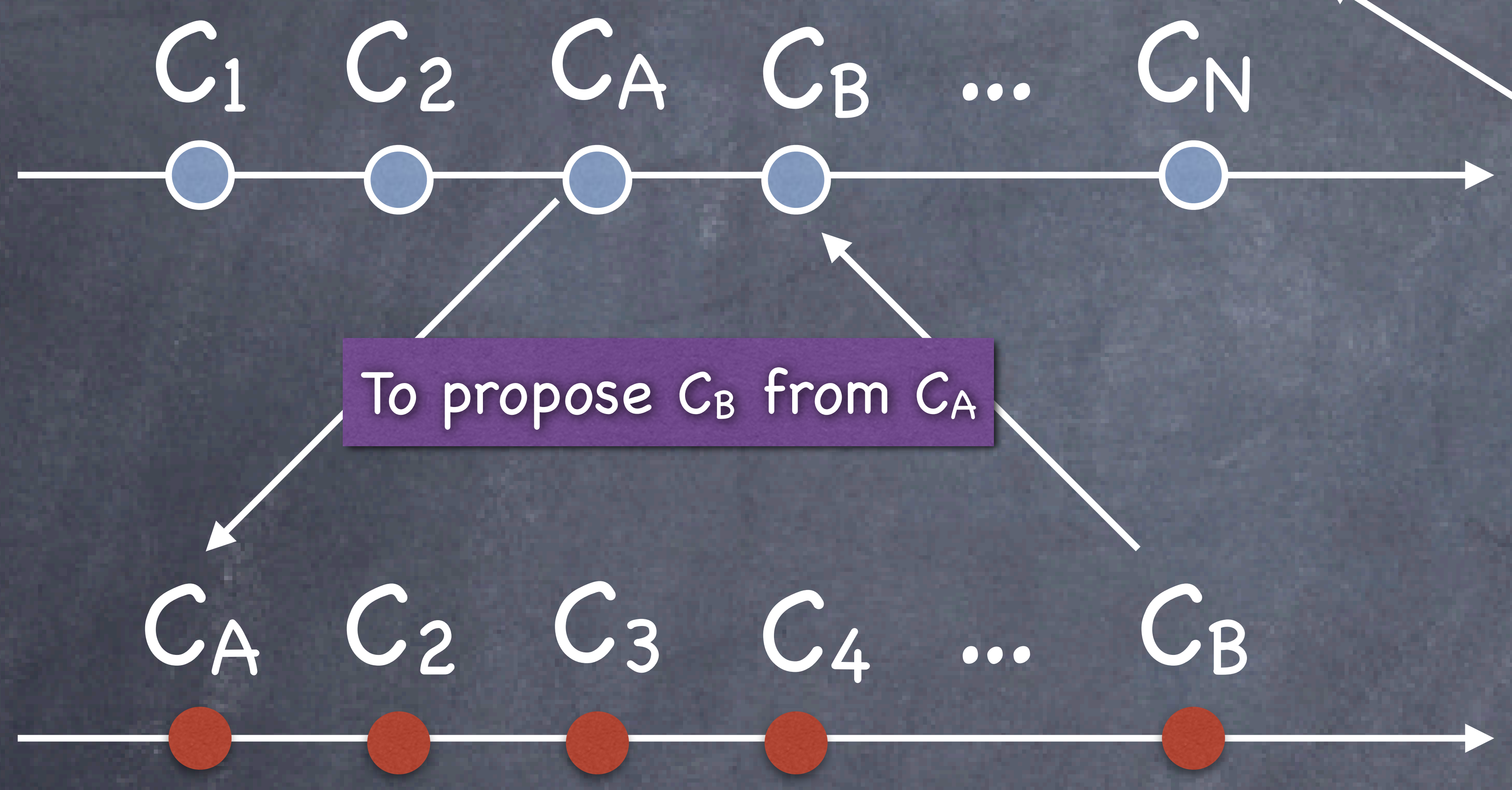
If the computational cost of the MLMD is small, we can speed up the simulation

In the field of atom and molecular systems, machine learning molecular dynamics was proposed in 2007

Self-learning Monte Carlo

for lattice QCD

Markov chain with the probability W(C)



Another Markov chain with the probability W'(C)

YN, Akinori Tanaka, Akio Tomiya,
"Self-learning Monte-Carlo for non-abelian gauge theory with dynamical fermions",
Phys. Rev. D 107, 054501 (2023)

$$S[U] = S_g[U] + S_f[U],$$

$$S_f[U] = -\log\ \det M^\dagger M,$$

integrated fermion action

effective model without fermion actions

$$S^\theta_{\rm eff}[U] = \sum_n \left[\beta_{\rm plaq}\sum_{\mu=1}^{4}\sum_{\nu>\mu}\left(1-\frac{1}{2}{\rm tr}U_{\mu\nu}(n)\right) + \beta_{\rm rect}\sum_{\mu=1}^{4}\sum_{\nu\neq\mu}\left(1-\frac{1}{2}{\rm tr}R_{\mu\nu}(n)\right)\right]$$
$$+\beta^{\mu=1}_{\rm Pol}\sum_{n_2,n_3,n_4}{\rm tr}\left[\prod_{n_1=0}^{N_1-1}U_1(\vec{n},n_4)\right] + \beta^{\mu=2}_{\rm Pol}\sum_{n_1,n_3,n_4}{\rm tr}\left[\prod_{n_2=0}^{N_2-1}U_2(\vec{n},n_4)\right]$$
$$+\beta^{\mu=3}_{\rm Pol}\sum_{n_1,n_2,n_4}{\rm tr}\left[\prod_{n_3=0}^{N_3-1}U_3(\vec{n},n_4)\right] + \beta^{\mu=4}_{\rm Pol}\sum_{n_1,n_2,n_3}{\rm tr}\left[\prod_{n_4=0}^{N_4-1}U_4(\vec{n},n_4)\right] + \beta_{\rm const},$$

Self-learning Monte Carlo

for lattice QCD

YN, Akinori Tanaka, Akio Tomiya,
“Self-learning Monte-Carlo for non-abelian gauge theory with dynamical fermions”,
Phys. Rev. D 107, 054501 (2023)

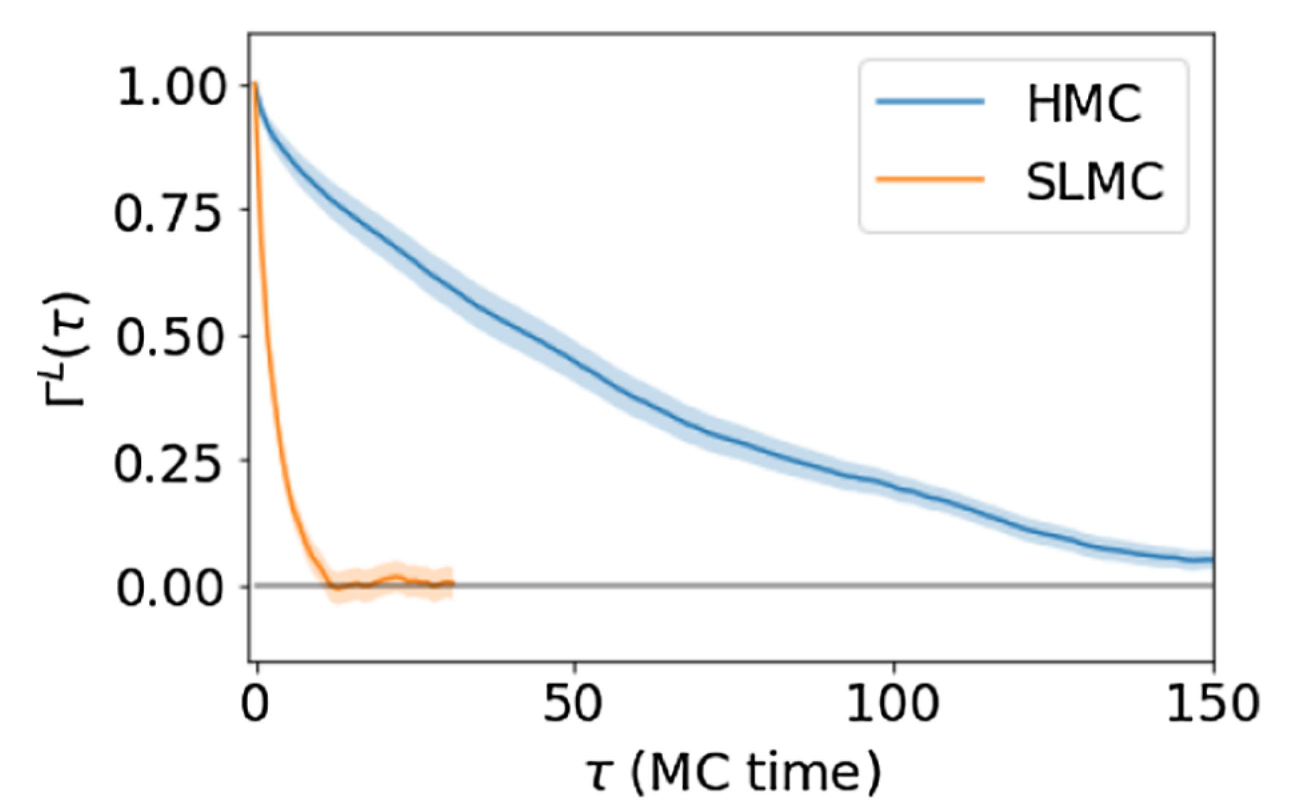
$$S[U] = S_g[U] + S_f[U],$$

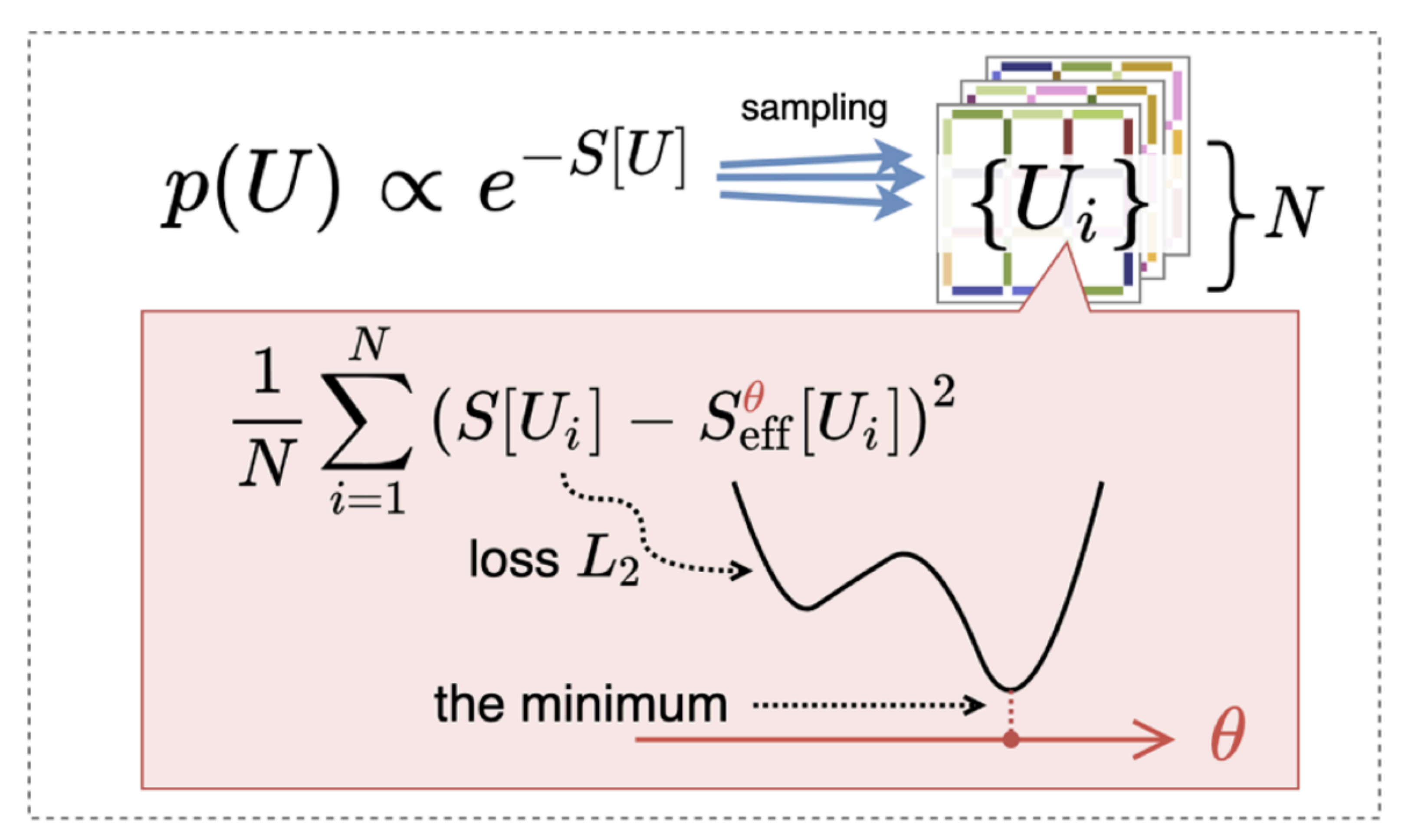
$$S_f[U] = -\log\ \det M^\dagger M,$$

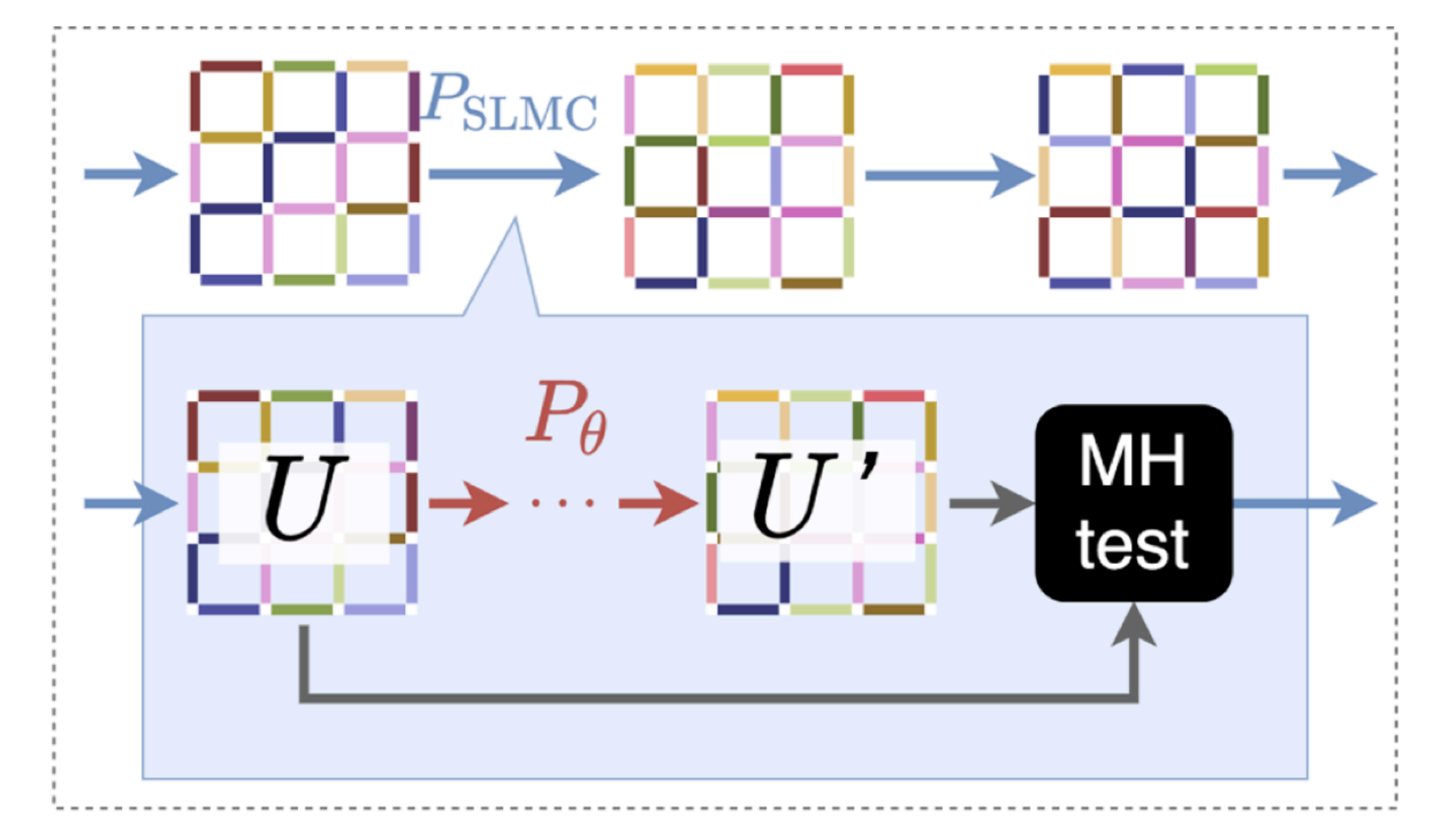
integrated fermion action

effective model without fermion actions

$$\begin{aligned} S^{\theta}_{\mathrm{eff}}[U] = \sum_n \Bigg[\beta_{\mathrm{plaq}} \sum_{\mu=1}^{4} \sum_{\nu>\mu} \left(1 - \frac{1}{2}\mathrm{tr} U_{\mu\nu}(n)\right) + \beta_{\mathrm{rect}} \sum_{\mu=1}^{4} \sum_{\nu\neq\mu} \left(1 - \frac{1}{2}\mathrm{tr} R_{\mu\nu}(n)\right)\Bigg] \\ + \beta^{\mu=1}_{\mathrm{Pol}} \sum_{n_2,n_3,n_4} \mathrm{tr}\left[\prod_{n_1=0}^{N_1-1} U_1(\vec{n}, n_4)\right] + \beta^{\mu=2}_{\mathrm{Pol}} \sum_{n_1,n_3,n_4} \mathrm{tr}\left[\prod_{n_2=0}^{N_2-1} U_2(\vec{n}, n_4)\right] \\ + \beta^{\mu=3}_{\mathrm{Pol}} \sum_{n_1,n_2,n_4} \mathrm{tr}\left[\prod_{n_3=0}^{N_3-1} U_3(\vec{n}, n_4)\right] + \beta^{\mu=4}_{\mathrm{Pol}} \sum_{n_1,n_2,n_3} \mathrm{tr}\left[\prod_{n_4=0}^{N_4-1} U_4(\vec{n}, n_4)\right] + \beta_{\mathrm{const}}, \end{aligned}$$







We use a linear interpolation

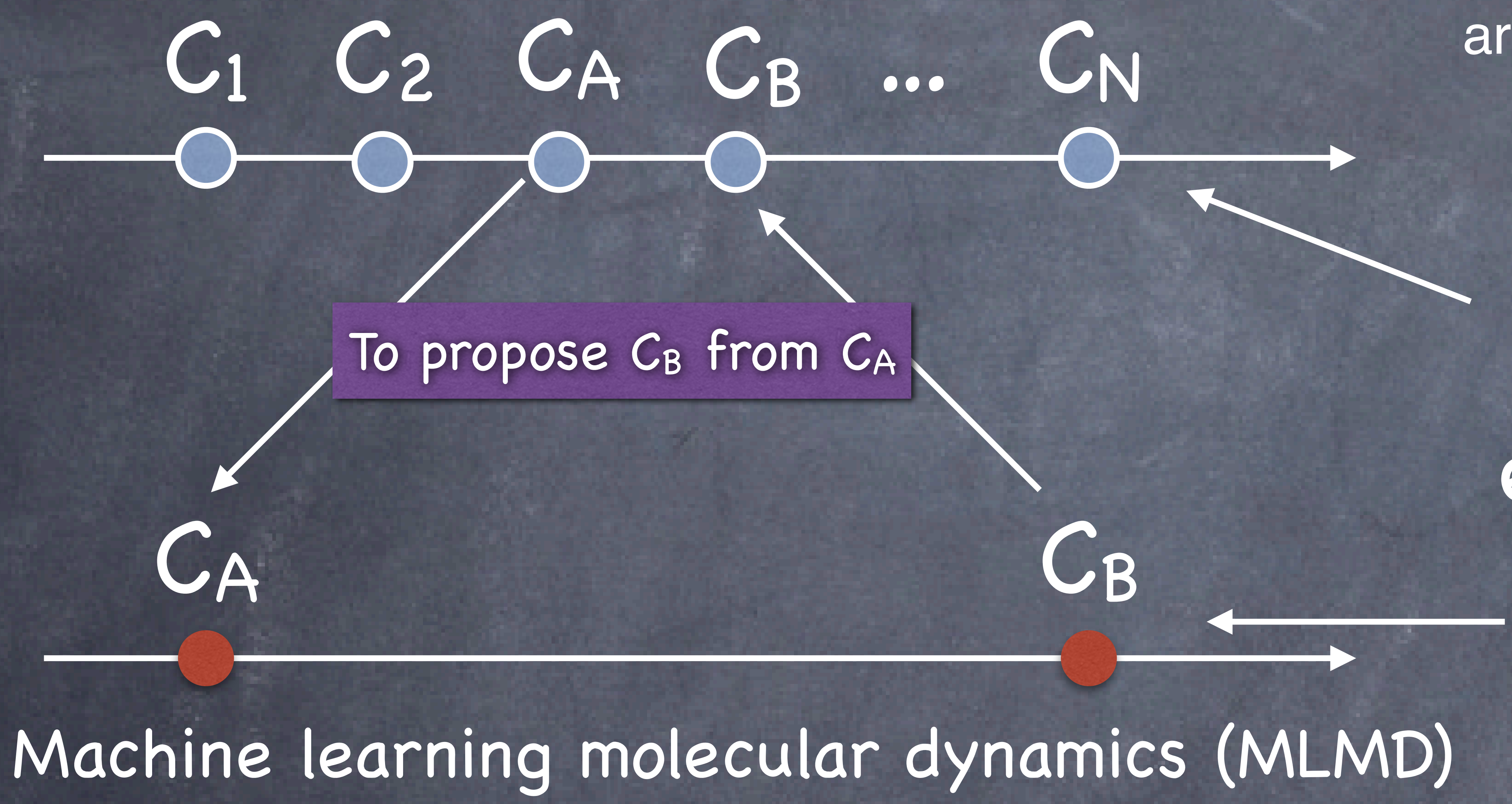
how to improve effective action?

Self-learning Hybrid Monte Carlo

for lattice QCD

Markov chain with the probability W(C)

YN and Akio Tomiya,
“Gauge covariant neural network for 4 dimensional non-abelian gauge theory”,
arXiv:2103.11965


C_1
C_2
C_A
C_B
…
C_N
To propose C_B from C_A
C_A
C_B
Machine learning molecular dynamics (MLMD)

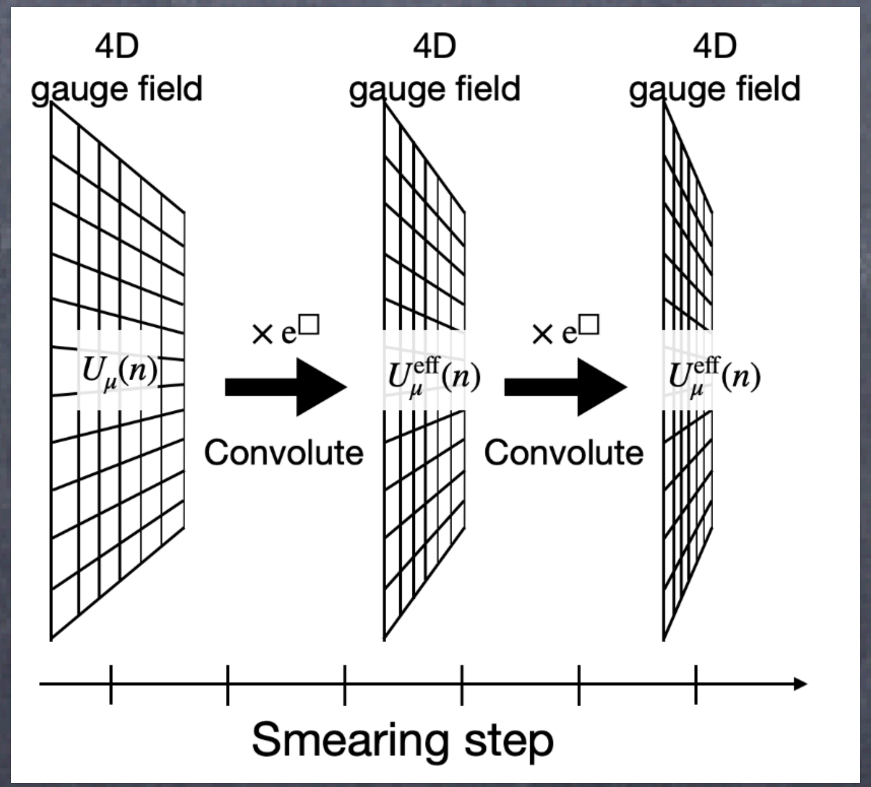
target action

$$S[U] = S_{\rm g}[U] + S_{\rm f}[\phi, U; m_{\rm l}],$$

effective action

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

if m1 < mh, the computational cost reduces



U^{NN}: trainable stout smearing

-> Next Dr. Akio Tomiya’s talk

Problems of SLMC

Configurations → **Heavy tasks** → Boltzmann weight

Configurations → **effective model** → Boltzmann weight

How to construct effective models?

Quality of the effective model is very important

In previous studies,

for example, a linear regression is used to construct the effective model inspired by the physical insight

Use Transformer!!

Transformer and Attention mechanism

Generative AIs

OpenAI ChatGPT

Gemini

DeepL

These AI have same architecture called Transformer

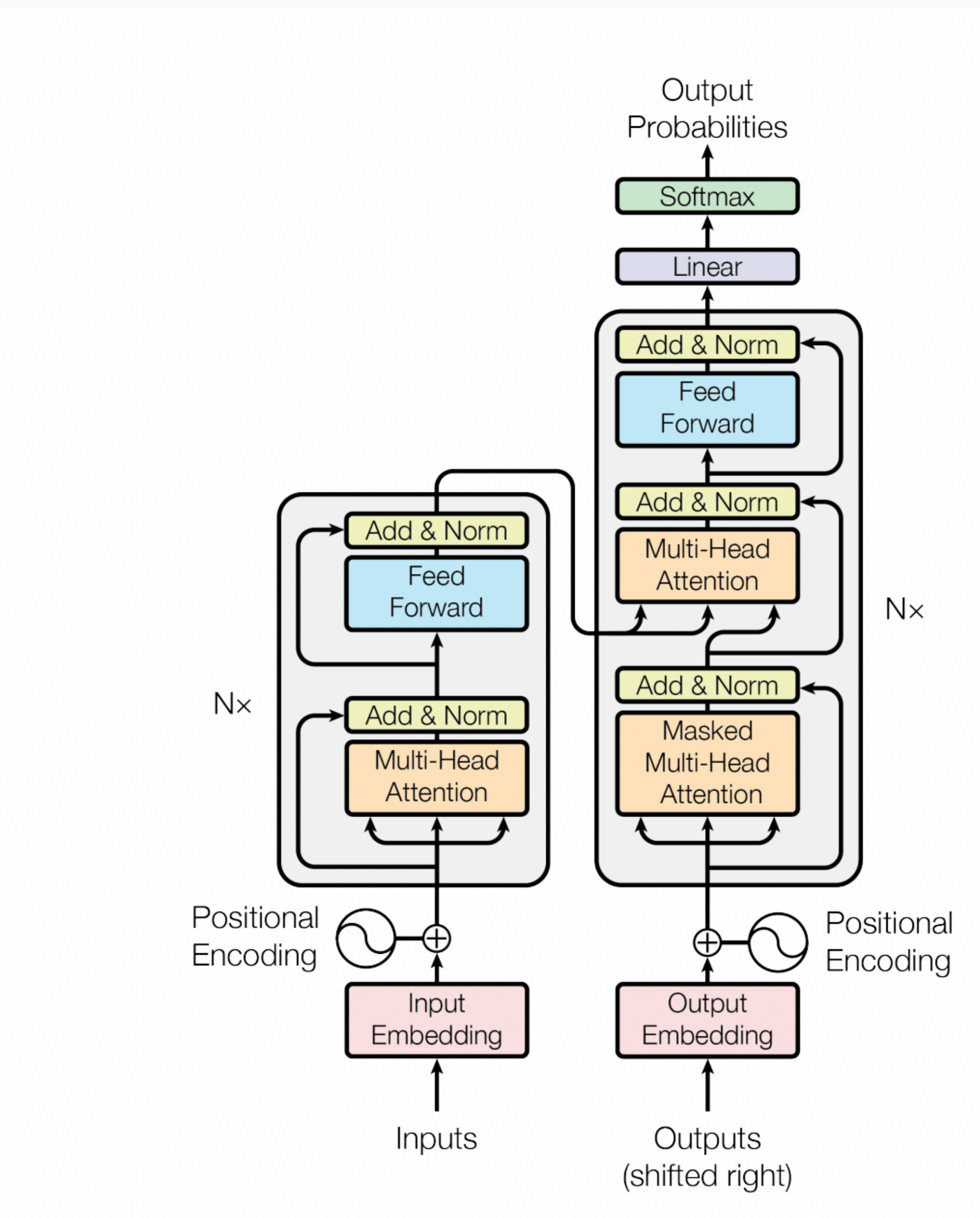


Figure 1: The Transformer - model architecture.

Transformer

AI Chat, Visualization, language translation

protein foldings etc.

Scaling lows of Transformer

https://arxiv.org/abs/2001.08361

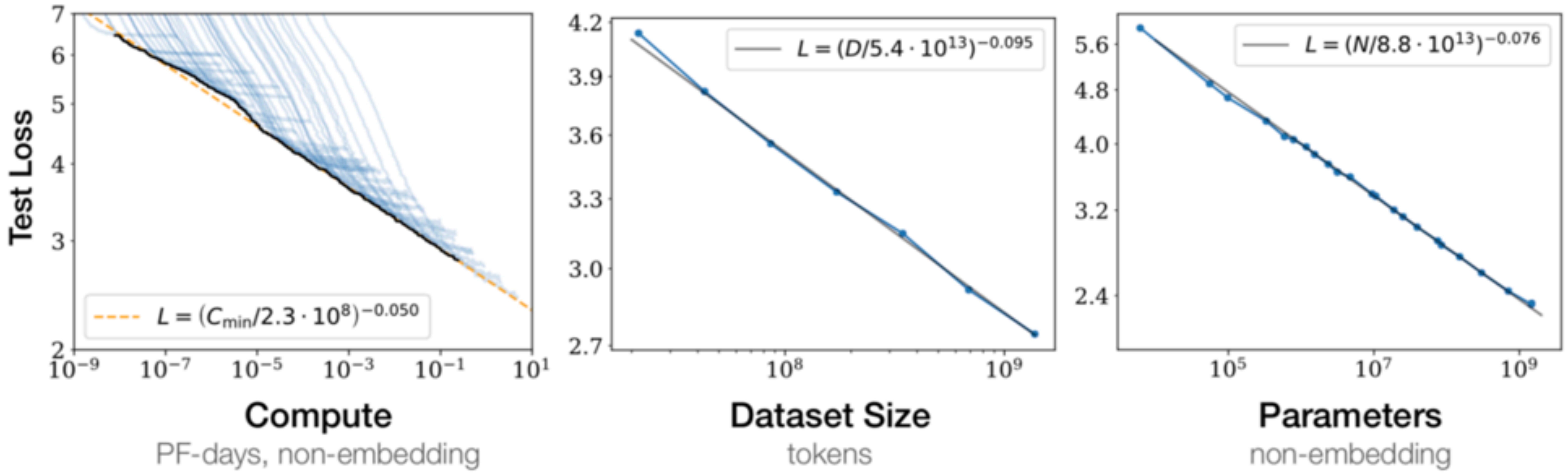
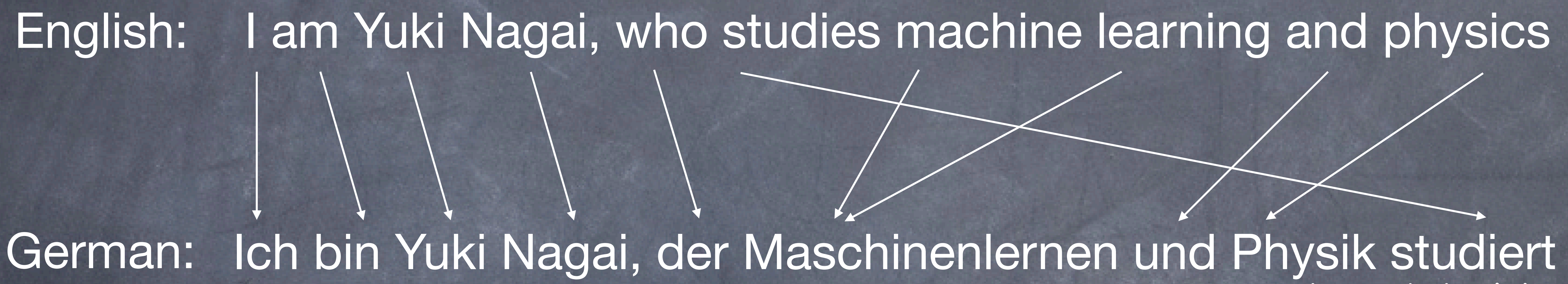


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when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

It requires huge data (e.g. GPT uses all electric books in the world)
= weak inductive bias, large data makes prediction better

Transformer and Attention

When we translate a sentence, we pay “attention” to words:



English: I am Yuki Nagai, who studies machine learning and physics

German: Ich bin Yuki Nagai, der Maschinenlernen und Physik studiert

translated by DeepL

Non-local dependencies can be treated by “Attention layer”

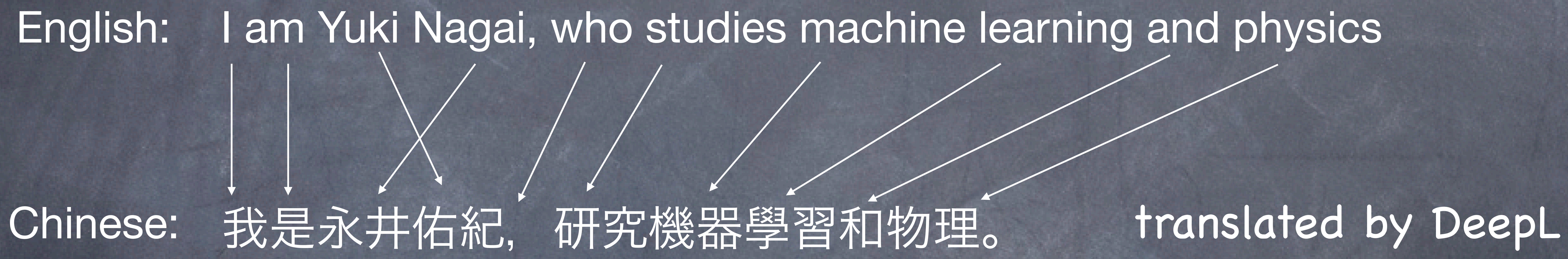
What are most important relations in words?

“Attention” layer can capture these relations

In physics terminology, this is **non local correlation.**
The attention layer enables us to treat it with a neural net!

Transformer and Attention

When we translate a sentence, we pay “attention” to words:



English: I am Yuki Nagai, who studies machine learning and physics

Chinese: 我是永井佑紀，研究機器學習和物理。

translated by DeepL

Non-local dependencies can be treated by “Attention layer”

What are most important relations in words?

“Attention” layer can capture these relations

In physics terminology, this is **non local correlation.**
The attention layer enables us to treat it with a neural net!

What is the attention mechanism?

There are many websites to explain the transformer and attention mechanism, in terms of language processing...

I try to explain the attention in terms of simple mathematics

This came from discussions with Dr. Tomiya

1. We consider a vector/matrix/tensor **A**   Ai or Aij or Aijk

2. We make three variables **K,Q,V** from **A**

   K = W^KA, Q = W^QA, V = W^VA   **W^k,W^Q,W^V:trainable parameters**

3. We generate new vector/matrix/tensor **B**

$$B_l = A_l + \sum_i P_i^l V_i \qquad P = \sigma(QK^T)$$

correlation between Q and K

l=i or ij or ijk

What is the attention mechanism?

$K = W^K A$, $Q = W^Q A$, $V = W^V A$

W^k,W^Q,W^V:trainable parameters

3. We generate new vector/matrix/tensor **B**

$$B_l = A_l + \sum_i P_i^l V_i$$

weighted sum

$$P = \sigma(QK^T)$$

σ :nonlinear funciton

correlation between Q and K

self-attention mechanism

This is most simplest architecture

In generative AIs, they use the multi-head attention

Simple mechanism but very effective!

How can we use this in physics?

Equivariant transformer

Problem in transformers

If we have many parameters (one Billion??), we can have a good model

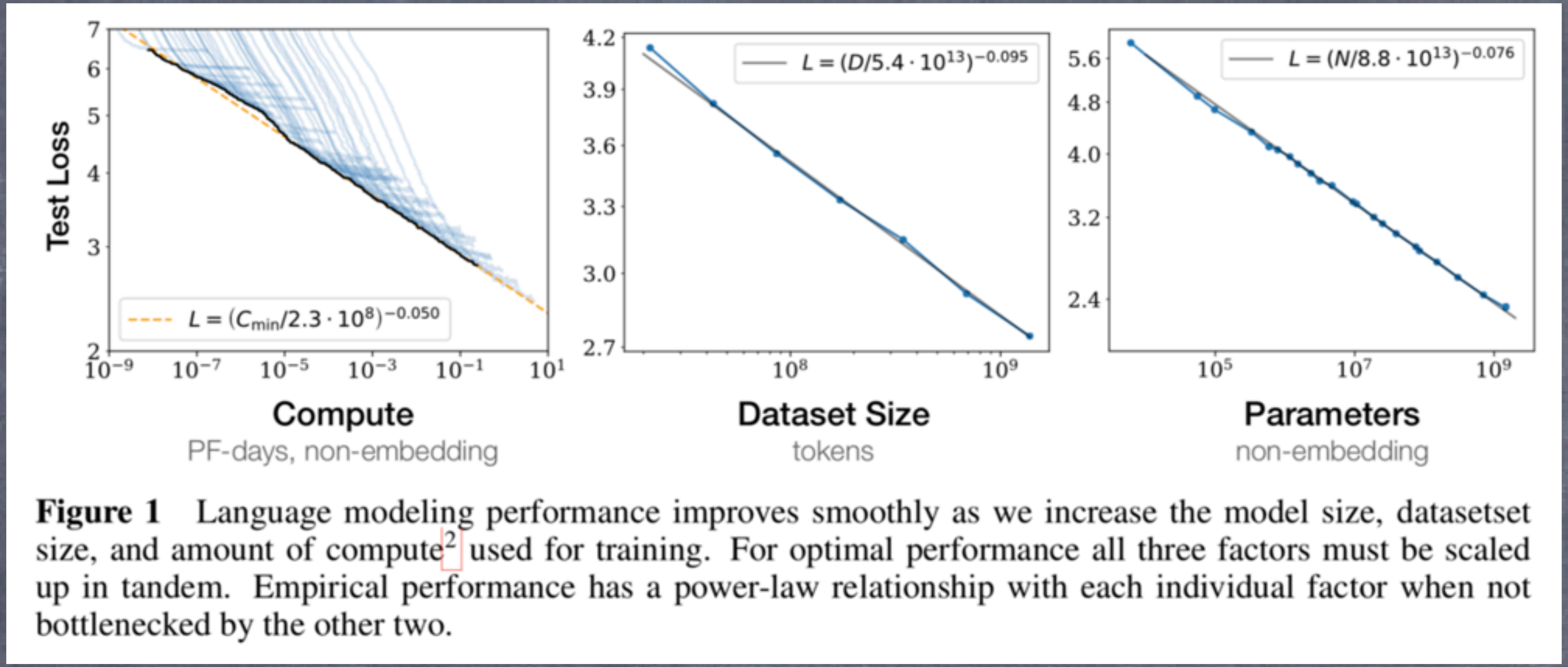


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when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

If a model in physics have billion parameters, the computational cost might be huge

-> We can not accelerate MCMC simulations!

We want to use transformers
We want to reduce num. of parameters

Let's use symmetry!!

Fermion and spin model

We want to focus on a simple lattice model

fermions and classical spins

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

called double exchange model
in condensed matter physics

Partition function:

$$Z = \sum_{\{\mathbf{S}\}}\prod_{n}(1 + e^{-\beta(\mu - E_n(\{\mathbf{S}\}))})$$

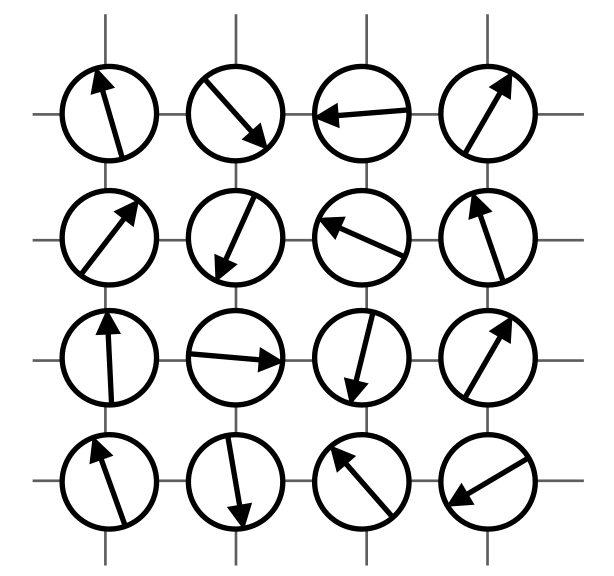
Input: spin configurations {S}

↓ diagonalization

Output: Boltzmann weight

Configurations: classical spins {S_i}

S_i: i-th three dimensional vector in spin space



We want to replace the diagonalization

Fermion and spin model

We want to focus on a simple lattice model

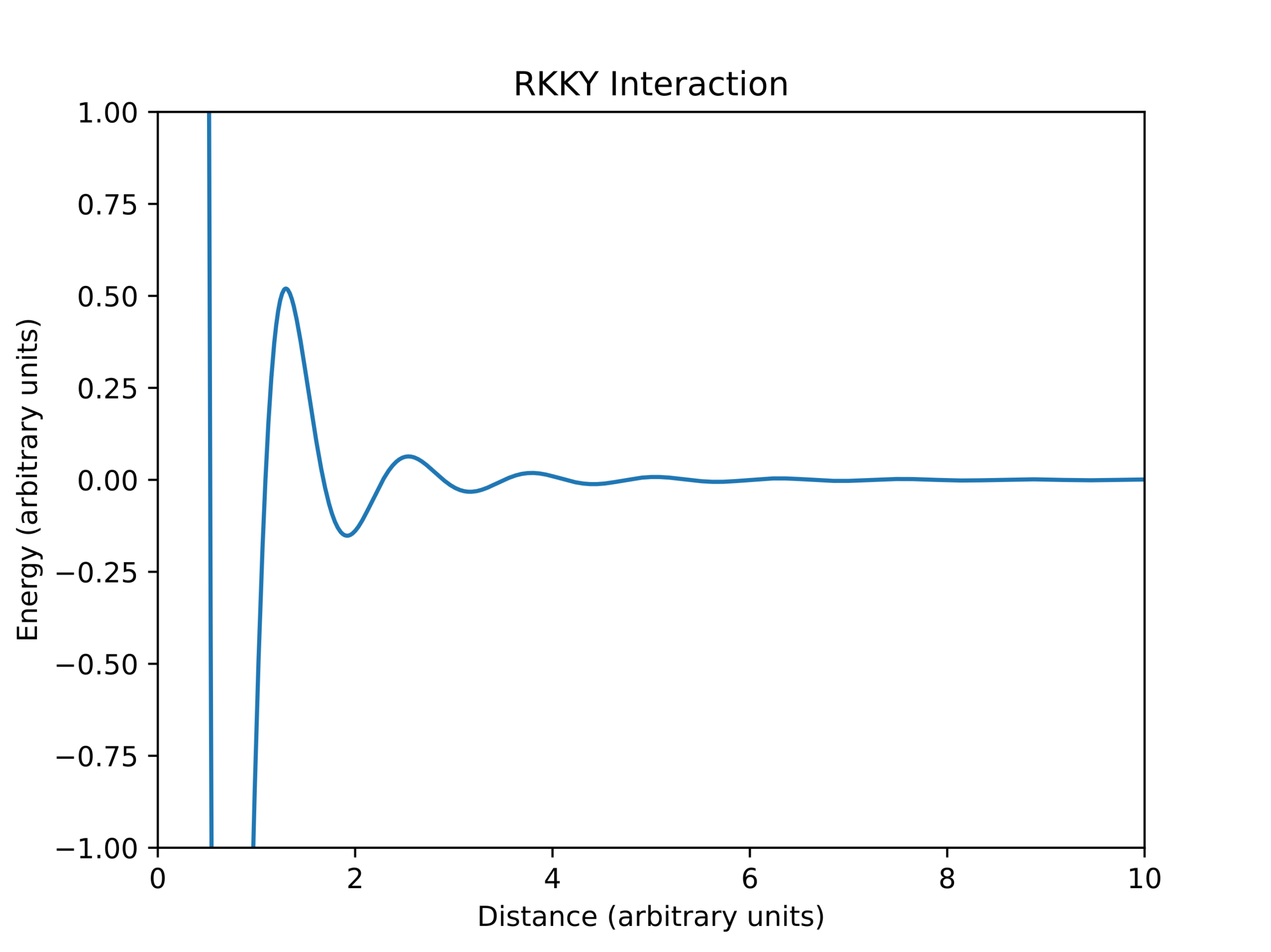
fermions and classical spins

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

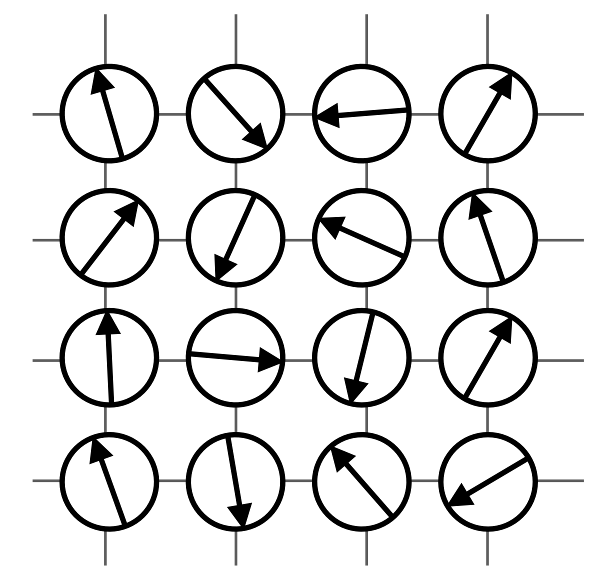
called double exchange model
in condensed matter physics

If J is small, we can use the perturbation theory

the **Ruderman–Kittel–Kasuya–Yosida** (**RKKY**) **interaction** models



$$H_{\mathrm{RKKY}} = -\sum_{\langle i,j \rangle_n} J_n \mathbf{S}_i \cdot \mathbf{S}_j$$



We can integrate out fermion degrees of freedom

fermion + spin -> spin

Fermion and spin model

We want to focus on a simple lattice model

fermions and classical spins

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

called double exchange model
in condensed matter physics

We want to consider large J region

Simple effective model

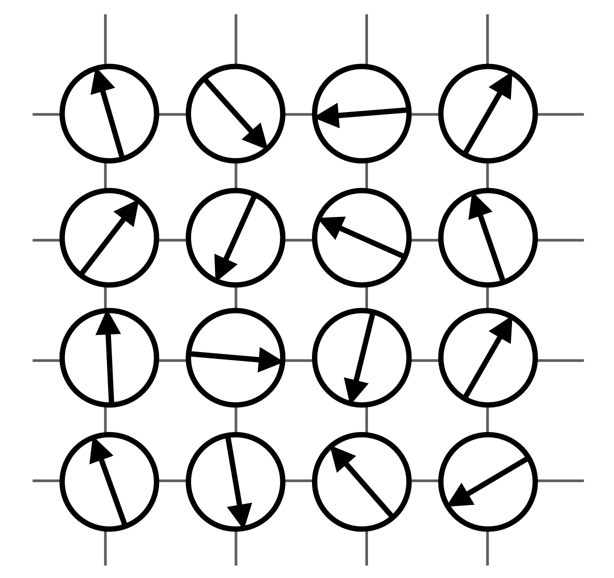
J. Liu, H. Shen, Y. Qi, Z. Y. Meng, and L. Fu, Phys. Rev. B 95, 241104(R)(2017)

J_n^{eff}: n-th nearest neighbor interaction

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

This is a linear model

by integrating out fermion degrees of freedom



similar to RKKY model
derived by physicist

There are only few parameters J_n^{eff}

Num. of parameters is too small! How to improve this model?

Fermion and spin model

fermions and classical spins

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

Simple effective model

J. Liu, H. Shen, Y. Qi, Z. Y. Meng, and L. Fu, Phys. Rev. B 95, 241104(R)(2017)

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

J_n^{eff}: n-th nearest neighbor interaction

This is a linear model

by integrating out fermion degrees of freedom

There are only few parameters J_n^{eff}

Effective model with a transformer

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

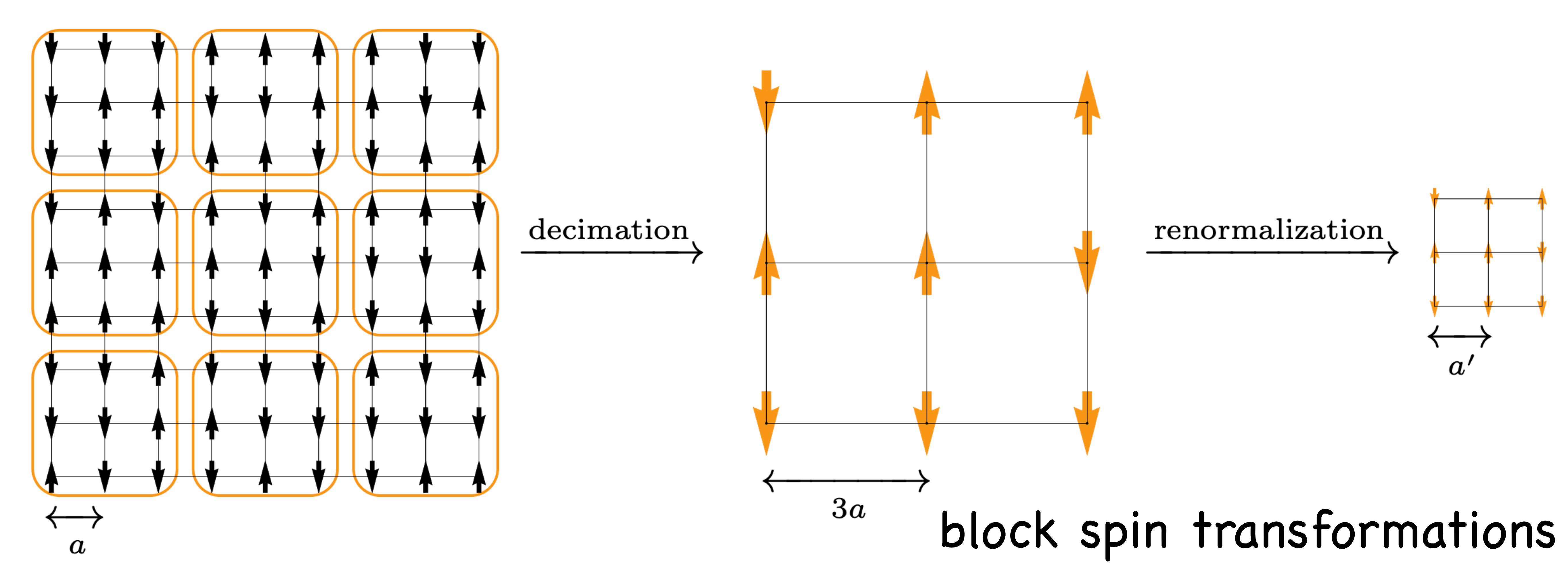
$$\mathbf{S}^{\mathrm{NN}}_i = f^{\mathrm{transformer}}(\{\mathbf{S}_i\})$$

We replace the spins with “translated” spin with a transformer

Fermion and spin model

How to construct model?

In physics, we know the renormalization group analysis



block spin transformations

Charlie Duclut., "Nonequilibrium critical phenomena :exact Langevin equations, erosion of tilted landscapes" Université Pierre et Marie Curie - Paris VI, 2017.

Spins are renormalized

Renormalized spin should have same symmetries

Spins become "effective" spins

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

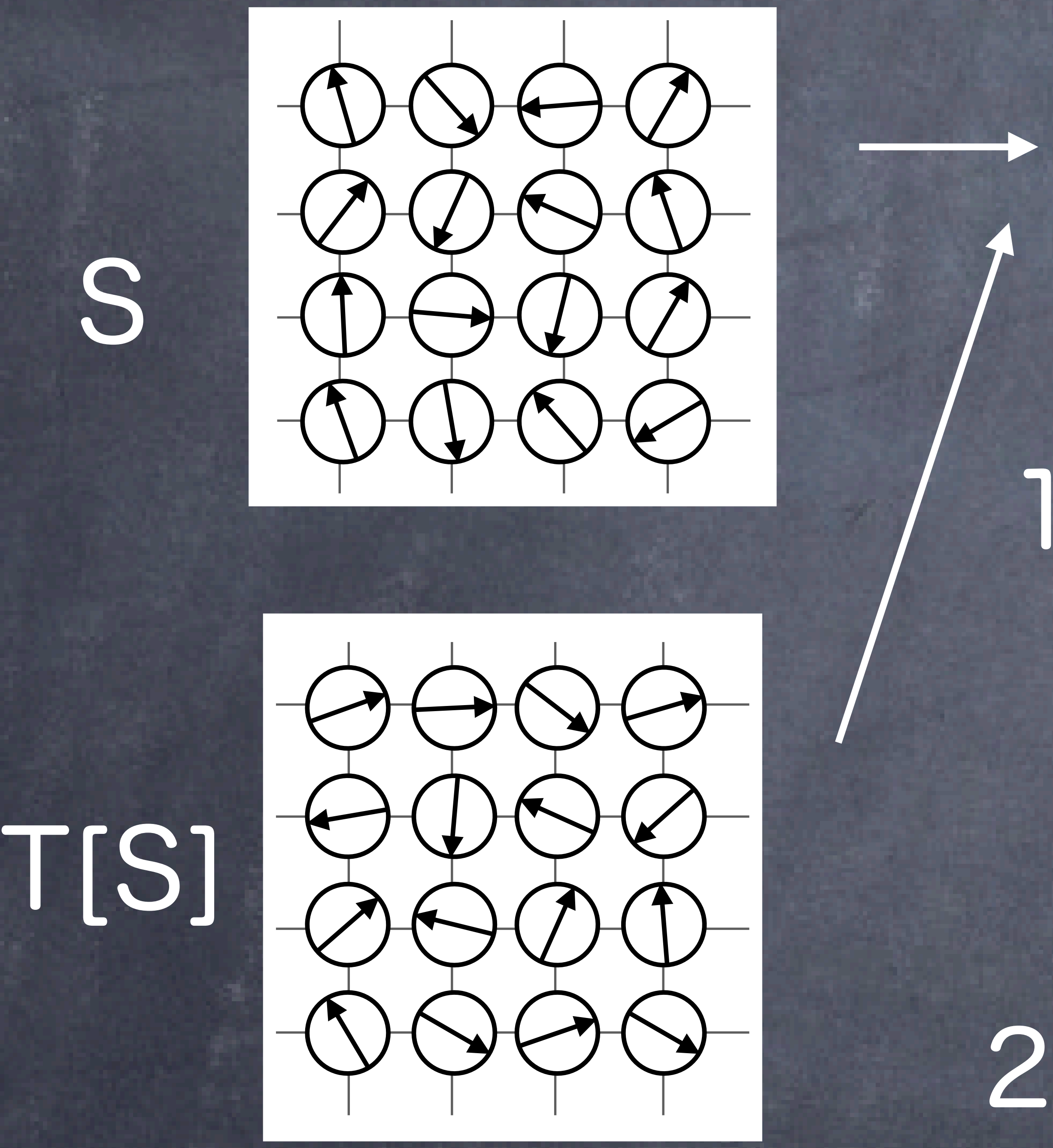
Heisenberg model for effective spins

If we can construct effective spins, we can construct effective model!

We need an equivariant model

Invariance and equivariance

Hamiltonian has a symmetry ->invariant with the symmetry operation T



$H(S) = H(T[S])$

symmetry invariant

We can consider two kinds of networks

1. make invariant input and put it into neural networks

$S \to C$

$T[S] \to C$ → $H = f(C)$

Conventional architecture can be used

2. make equivariant networks and make the output invariant

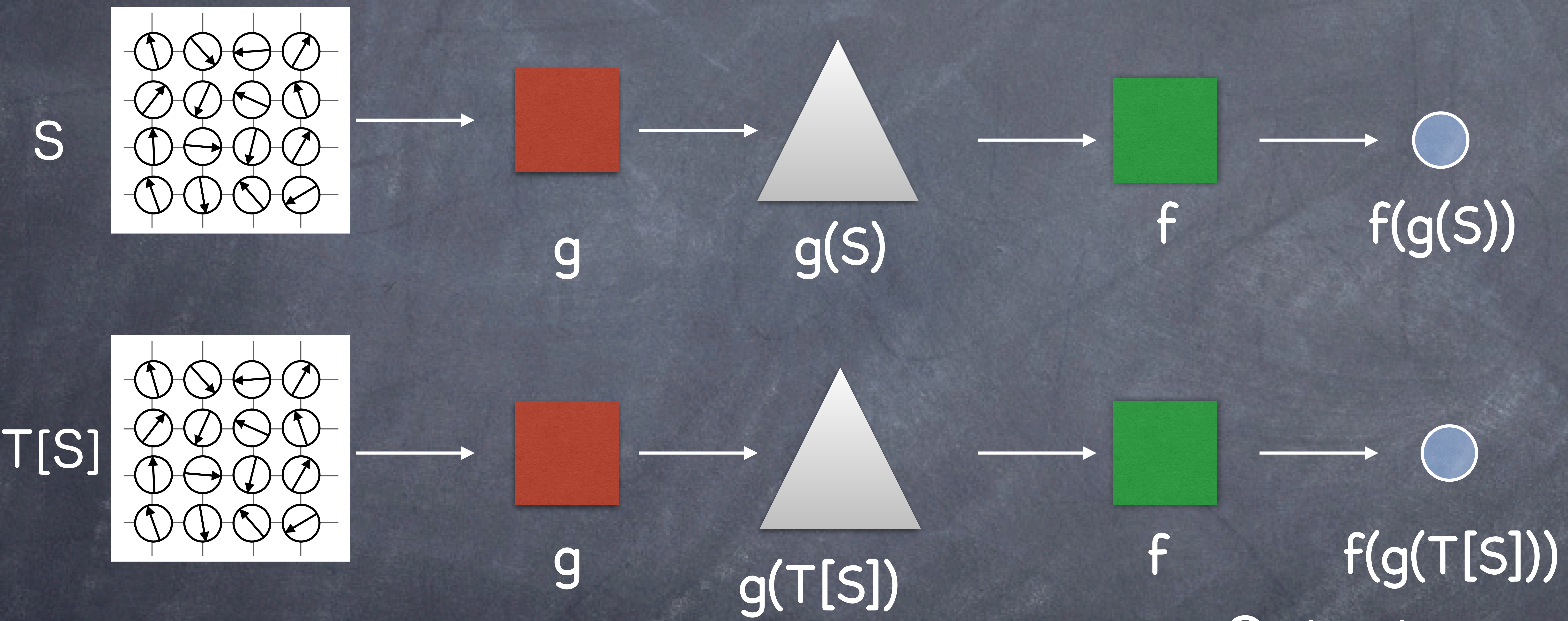
$T[g(S)] = g(T[S])$ $C = g(S)$ → $H = f(C)$

Equivariance

This network can keep a symmetry

Invariance and equivariance

2. make equivariant networks and make the output invariant



S

g

g(S)

f

f(g(S))

T[S]

g

g(T[S])

f

f(g(T[S]))

$f(T[S]) = f(S)$

Invariance

$T[g(S)] = g(T[S])$

Equivariance

Outputs are same

CNN uses equivariance

How to construct the attention layer

1. We consider a vector/matrix/tensor **A** Ai or Aij or Aijk

We consider spin "matrix" $[\hat{S}]_{i\mu} = [\vec{S}_i]_\mu$

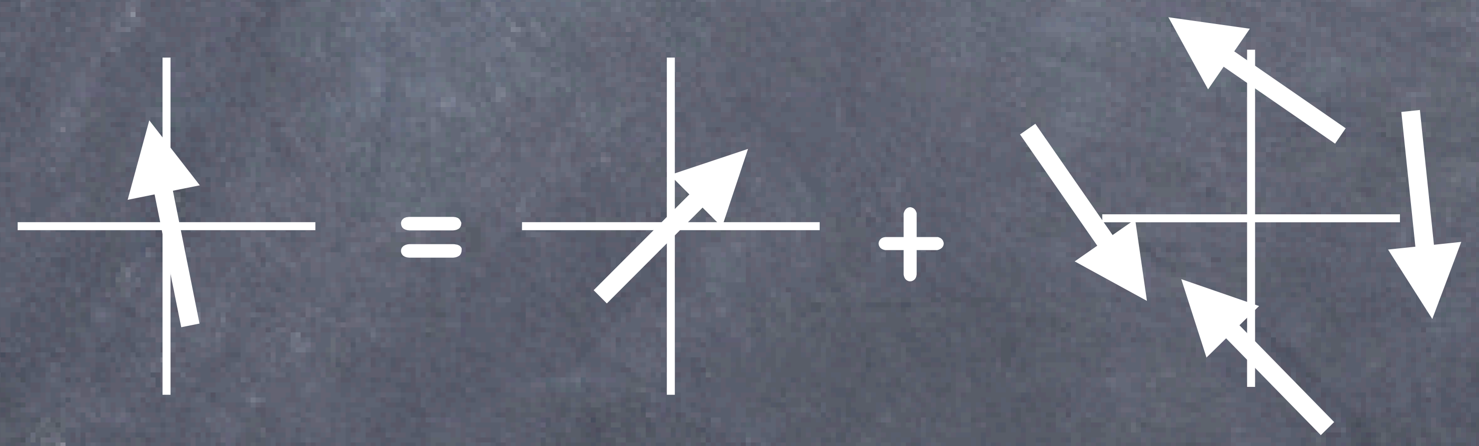
$\vec{S}_i$ is a classical spin on i-th site (vector)

2. We make three variables **K,Q,V** from **A**

K = W^K^A, Q = W^Q^A, V = W^V^A

We introduce "operators" $\hat{S}^{\mathrm{Q}} = \bar{W}^{\mathrm{Q}}\hat{S}$ $\hat{S}^{\mathrm{K}} = \bar{W}^{\mathrm{K}}\hat{S}$ $\hat{S}^{\mathrm{V}} = \bar{W}^{\mathrm{V}}\hat{S}$

$$[\bar{W}^\alpha \hat{S}]_{i\mu} \equiv \sum_{\langle i,j \rangle_n} W^\alpha_n \hat{S}_{j\mu}$$



W^k,W^Q,W^V:trainable parameters

W^k,W^Q,W^V do not depend on the site i (translational symmetry)

num. of parameters becomes a few

Yuki Nagai and Akio Tomiya, J. Phys. Soc. Jpn. 93, 114007 (2024)

How to construct the attention layer

3. We generate new vector/matrix/tensor **B**

$$B_l = A_l + \sum_i P_i^l V_i$$

⇨ $$\hat{S}^{(l)} \equiv \mathcal{N}(\hat{S}^{(l-1)} + \check{M}\hat{S}^{\mathrm{V}}) \qquad [\mathcal{N}(\hat{S})]_{i\mu} = [\vec{S}_i]_\mu / ||\vec{S}_i||$$

$P = \sigma(QK^T)$ correlation between Q and K

⇨ $$[\check{M}]_{ij} = \mathrm{ReLU}\left(\frac{1}{\sqrt{3}} \sum_{\mu=1}^{3} \hat{S}_{i\mu}^{\mathrm{Q}} \hat{S}_{j\mu}^{\mathrm{K}} \right)$$

$\hat{S}^{(l)}$ has spin-rotational equivariance

R[g(S)] = g(R[S])

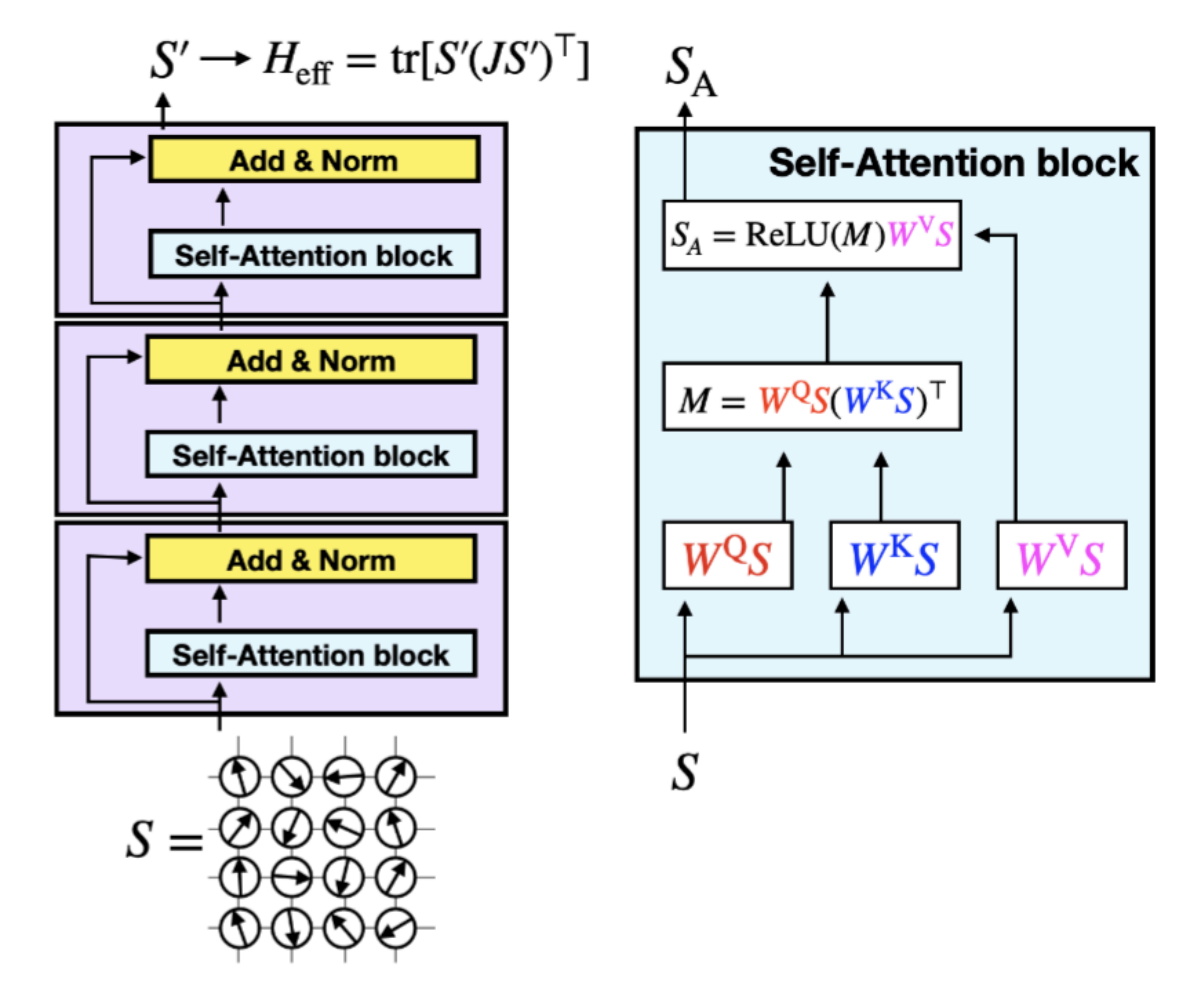
The "effective" spin $S^{(L)}$ can be regarded as a physical spin

-> renormalized spin

We can build a model!

Yuki Nagai and Akio Tomiya, J. Phys. Soc. Jpn. 93, 114007 (2024)

Equivariant Transformer for spin systems



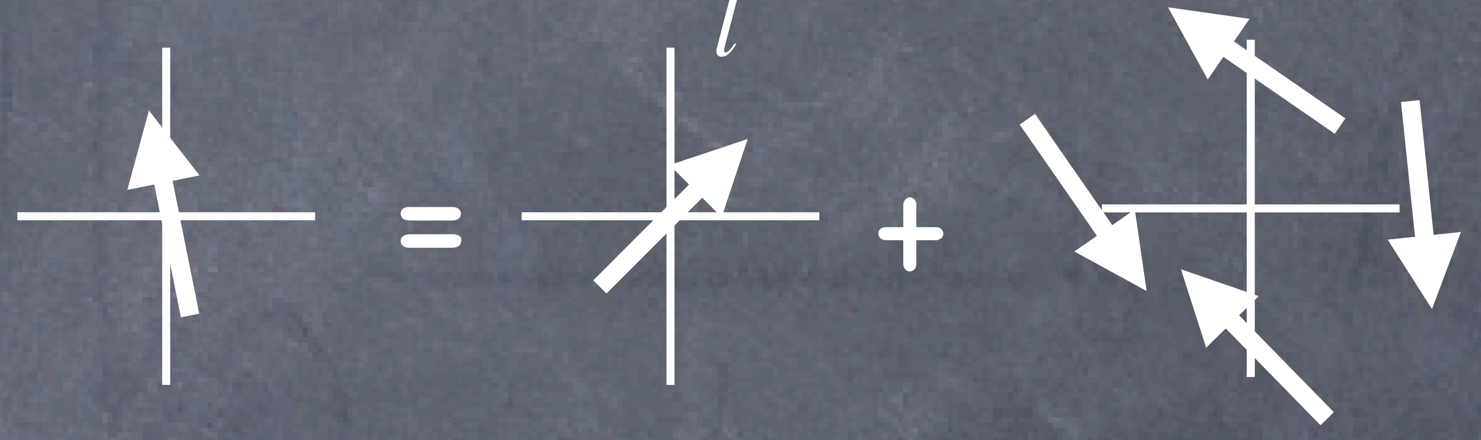
S → K = W^K S, Q = W^Q S, V = W^V S

$$K = W^{K}S$$

$$Q = W^{Q}S$$

$$V = W^{V}S$$

$$K_i = \sum_{l} W_l S_{i+l}$$



W only mixes neighbor spins
(short range interaction)

like block spin transformations

$$M = W^{Q}SS^{+}W^{K}$$

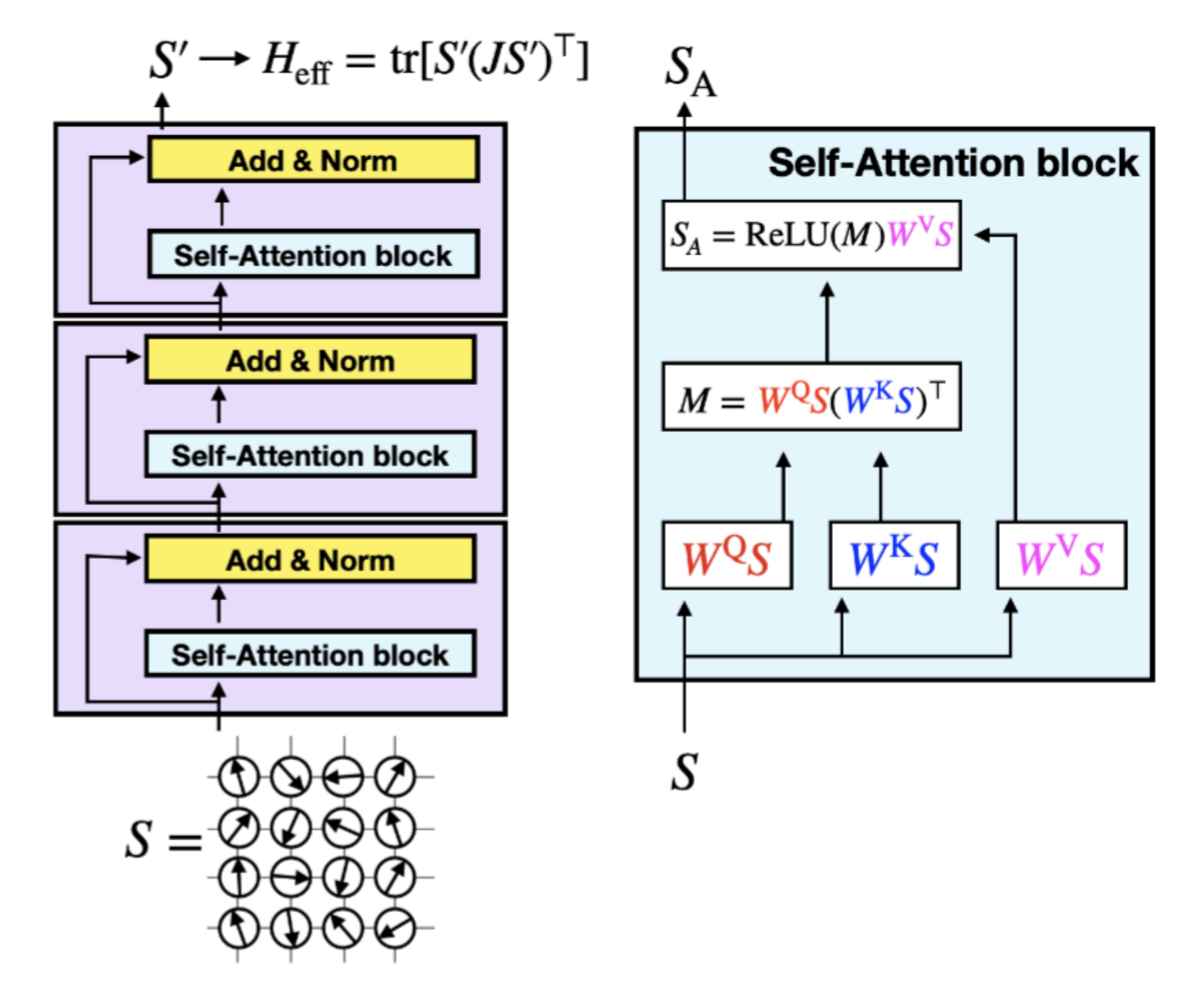
Rotational and translational invariant

$$S' = S + \mathrm{ReLU}(M)\, W^{V}S$$

Long range correlation is included

Yuki Nagai and Akio Tomiya, J. Phys. Soc. Jpn. 93, 114007 (2024)

Equivariant Transformer for spin systems



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

Layer 1

$$S_1 = \mathcal{N}(S + \mathrm{ReLU}(M^1(S))W^{V1}S)$$

Layer 2

$$S_2 = \mathcal{N}(S_1 + \mathrm{ReLU}(M^2(S_1))W^{V2}S_1)$$

Layer 3

$$S_3 = \mathcal{N}(S_2 + \mathrm{ReLU}(M^3(S_2))W^{V3}S_2)$$

Last Heisenberg model with effective spins

$$E = \sum_i \sum_l J_l \vec{S}_{3i} \cdot \vec{S}_{3i+l} + E_0$$

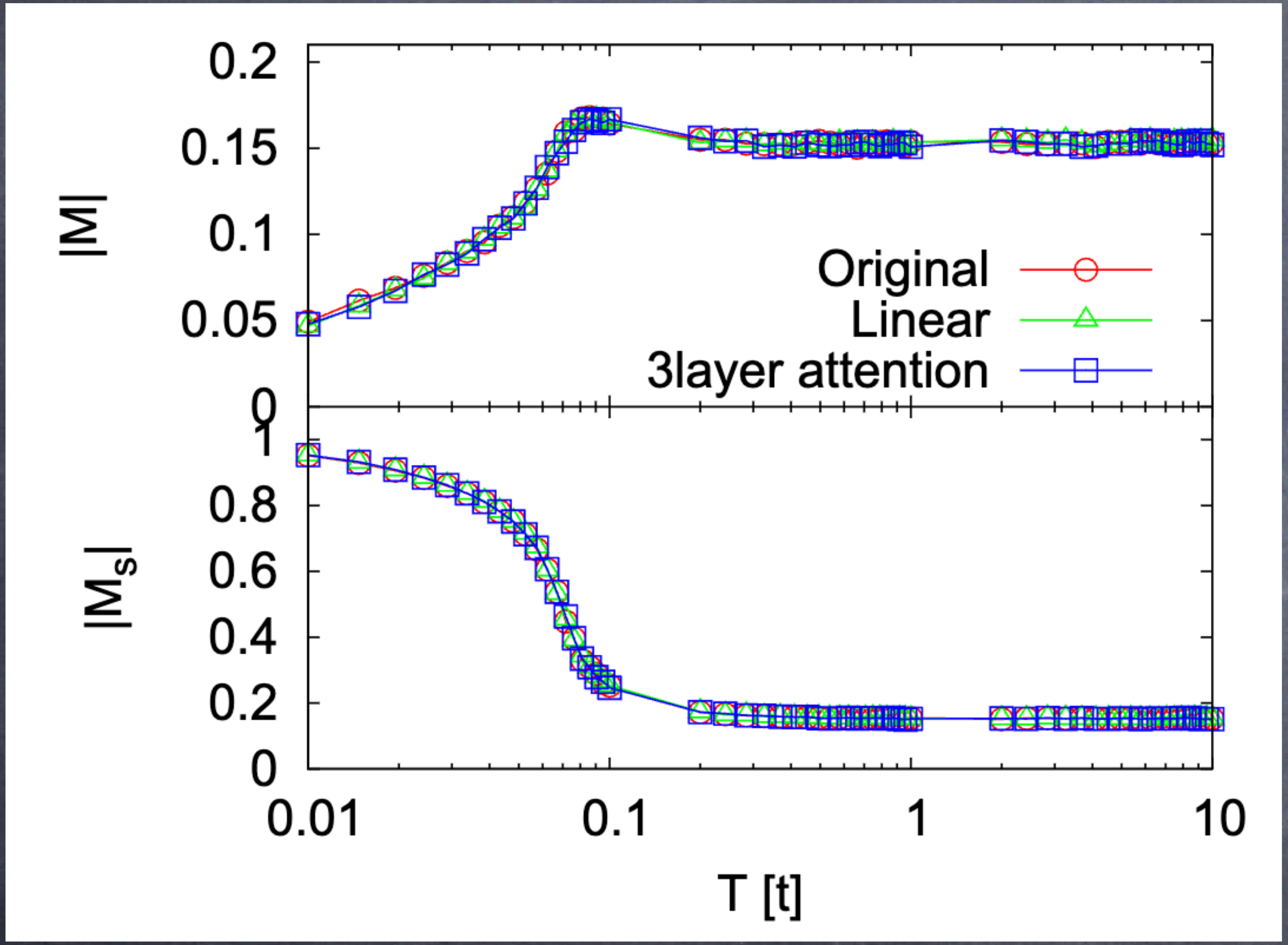
If the second term is zero

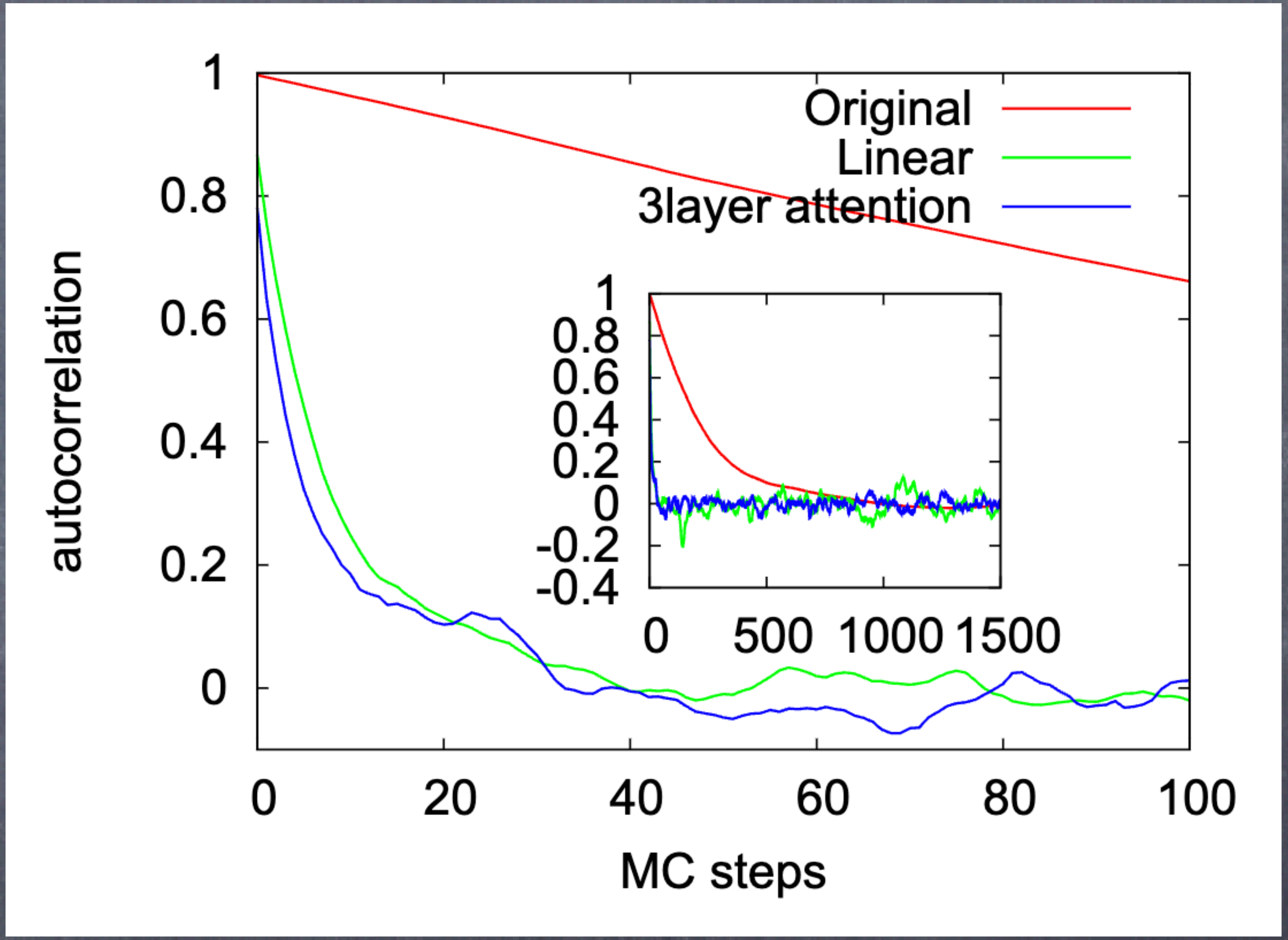
$$E = \sum_i \sum_l J_l \vec{S}_i \cdot \vec{S}_{i+l} + E_0$$ we get linearized model

Yuki Nagai and Akio Tomiya, J. Phys. Soc. Jpn. 93, 114007 (2024)

Results

2D double exchange model(fermion + classical spin)





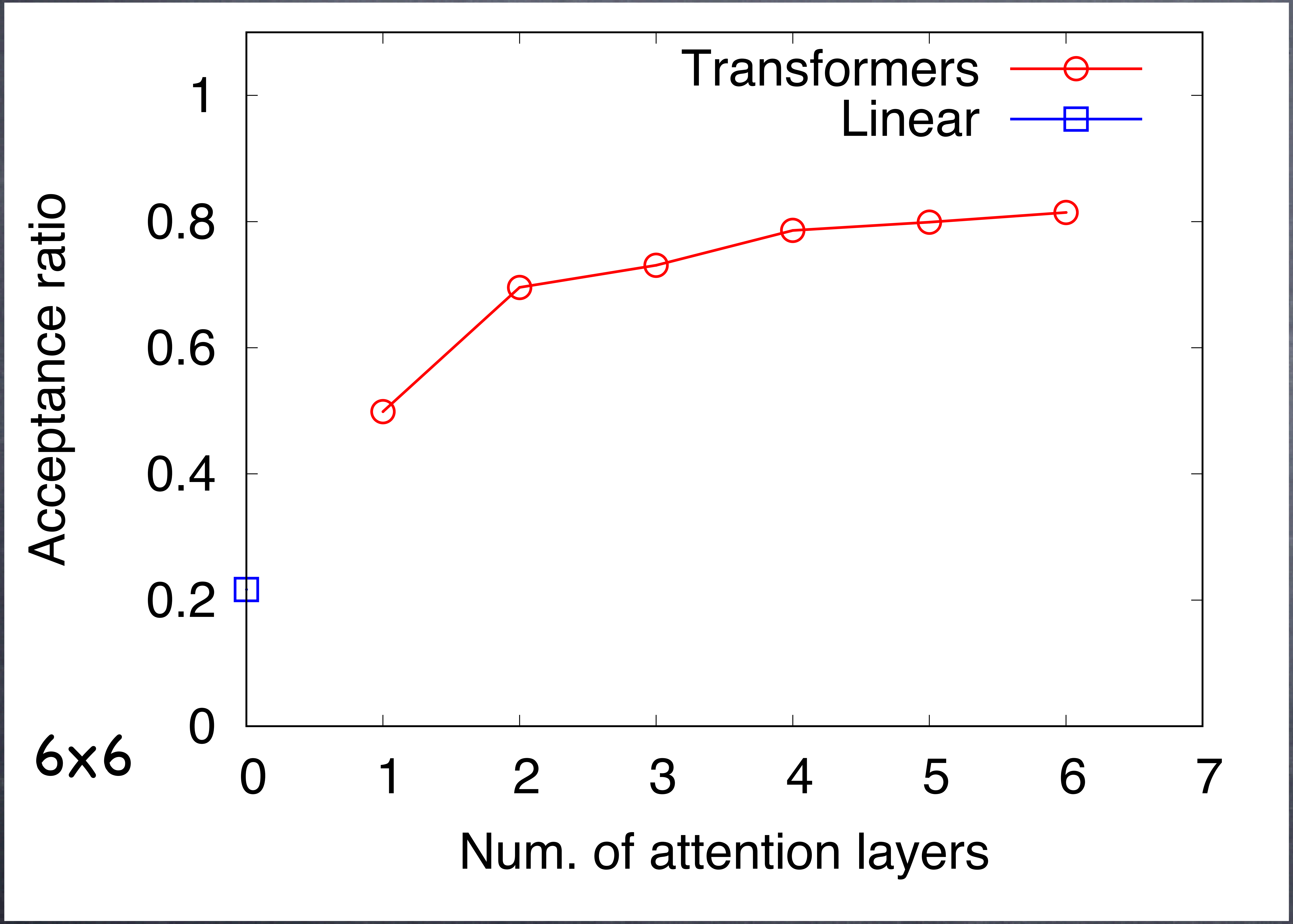
magnetization
and staggered magnetization

Autocorrelation time is reduced

Yuki Nagai and Akio Tomiya, J. Phys. Soc. Jpn. 93, 114007 (2024)

Results

N=6



6-th nearest neighbors

$$K_i = \sum_l W_l S_{i+l}$$

Num. of parameters per layer

7+7+7 = 21

Last layer: nearest neighbors

$$E = \sum_i \sum_l J_l \vec{S}_{3i} \cdot \vec{S}_{3i+l}$$

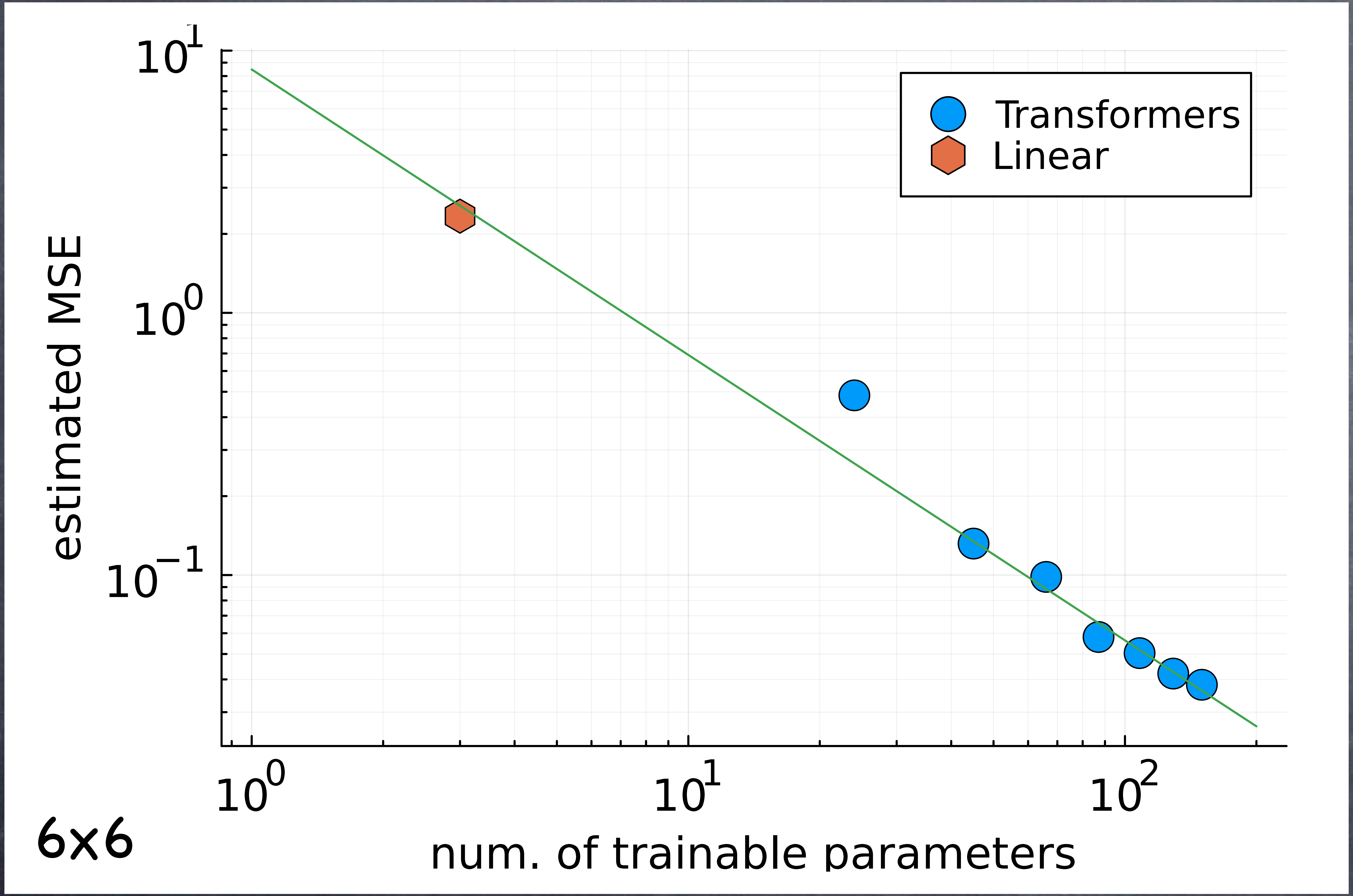
Num. of parameters is small

High acceptance ratio!

Yuki Nagai and Akio Tomiya, J. Phys. Soc. Jpn. 93, 114007 (2024)

Results

arXiv: 2306.11527



6-th nearest neighbors

$$K_i = \sum_l W_l S_{i+l}$$

Num. of parameters per layer

7+7+7 = 21

Scaling low?

This is like the scaling lows in Large Language Models

This is MC simulation

We generate data as we want

Application to LatticeQCD

2. We make three variables **K,Q,V** from **A**

$K = W^K A$, $Q = W^Q A$, $V = W^V A$

We introduced "operators" $\hat{S}^{\mathrm{Q}} = \bar{W}^{\mathrm{Q}}\hat{S}$ ⇒ Effective gauge field U^Q is needed

3. We generate new vector/matrix/tensor **B**

$$B_l = A_l + \sum_i P_i^l V_i \qquad P = \sigma(QK^T)$$ correlation between Q and K

We introduced inner product of spins ⇒ What is "inner product" in gauge field?

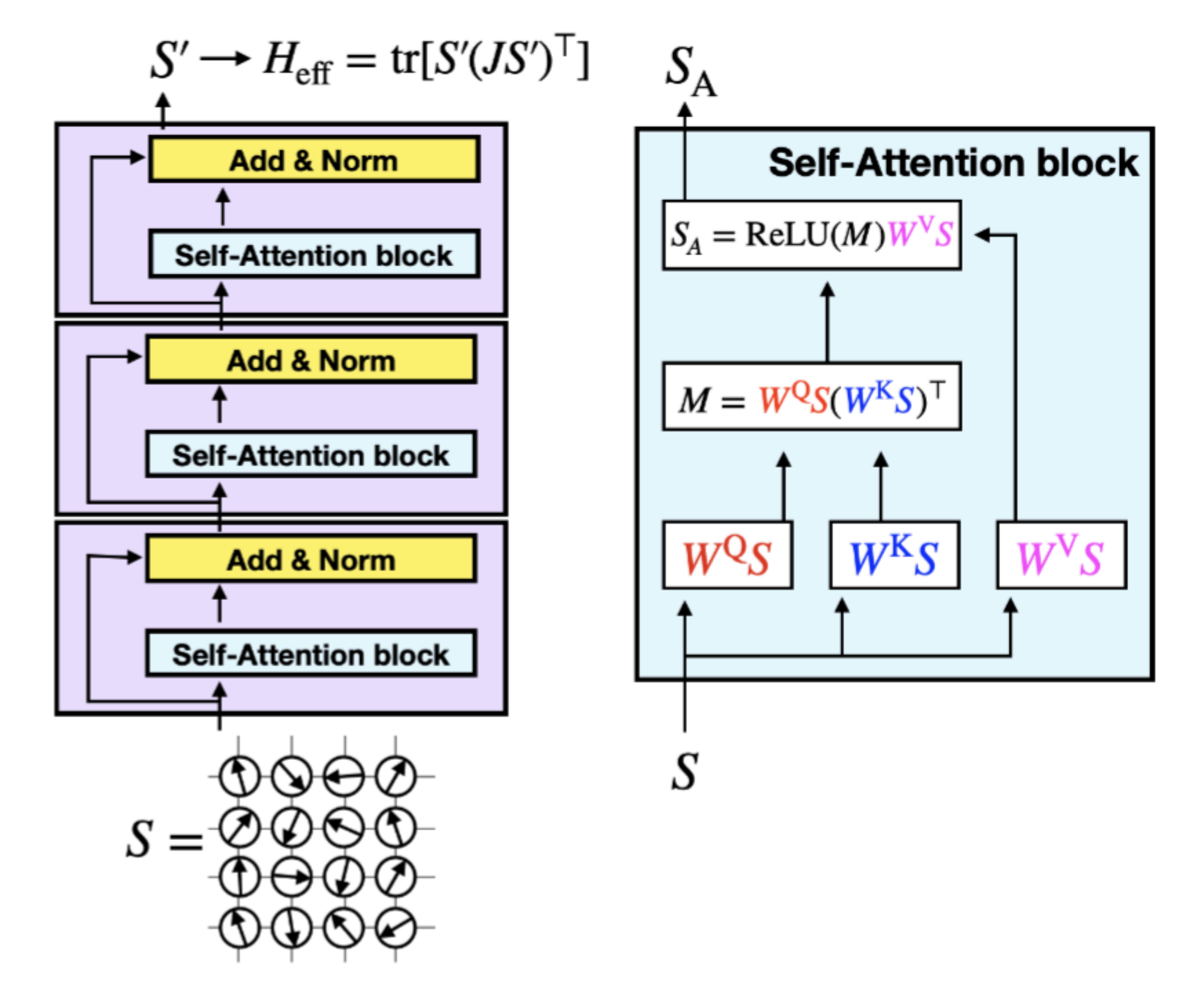
-> Next Dr. Akio Tomiya's talk

Summary

Summary

Yuki Nagai and Akio Tomiya, “Self-Learning Monte Carlo with Equivariant Transformer”, J. Phys. Soc. Jpn. 93, 114007 (2024)

Equivariant Transformer in spin systems



Equivariant with respect to spin-rotational and translational symmetries

We found the scaling low!

We can improve models with increasing num. of layers

“Transformer and Attention” is very useful!