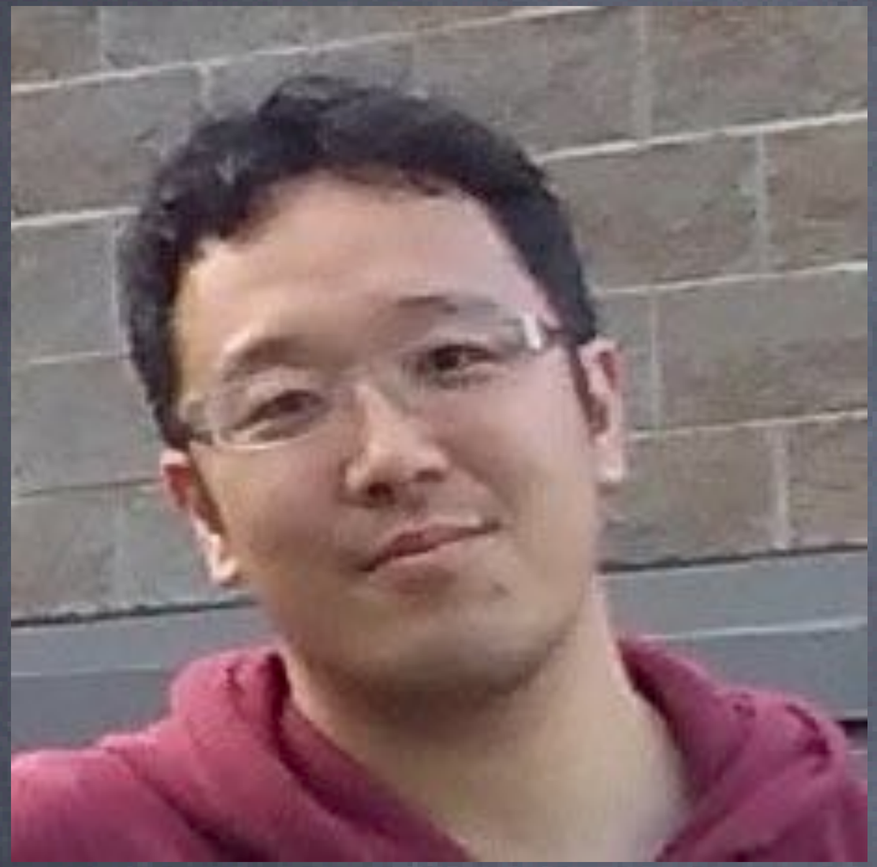
Self-learning Monte-Carlo Method for non-abelian gauge theory with dynamical fermions

Japan Atomic Energy Agency
RIKEN AIP

Yuki Nagai

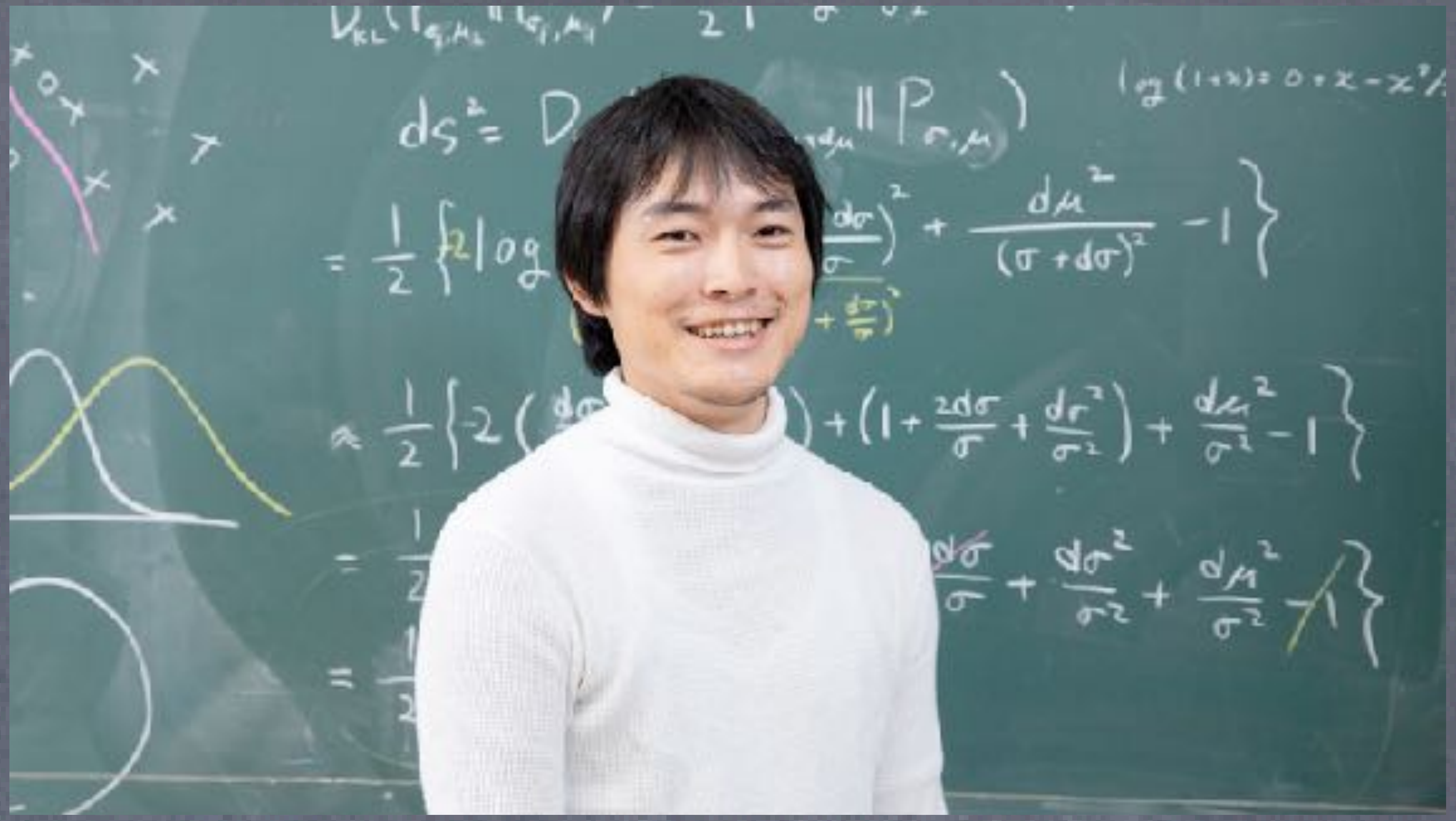
Yuki Nagai, Akinori Tanaka, Akio Tomiya,
“Self-learning Monte-Carlo for non-abelian gauge theory with dynamical fermions”,
arXiv:2010.11900

Collaborators



Akio Tomiya
RIKEN BNL

Machine learning, Lattice QCD



Akinori Tanaka
RIKEN AIP,
RIKEN ITHEMS

Machine learning, Physics and Mathematics

Yuki Nagai

Machine learning, Condensed matter

Yuki Nagai, Akinori Tanaka, Akio Tomiya, "Self-learning Monte-Carlo for non-abelian gauge theory with dynamical fermions", arXiv:2010.11900

About me

Superconductivity and condensed matter theory

2010: Ph.D in Univ. of Tokyo

2010-2019 researcher in Japan Atomic Energy Agency

2016-2017 visiting researcher in MIT

Machine learning and physics

2018- visiting researcher in RIKEN AIP

2019- senior researcher in Japan Atomic Energy Agency

Yuki Nagai

What machine and condensed matter physicist do

Lagrangian → Analytical calculations / Numerical calculations → Physical observables

Lagrangian →

me

Machine

→ Physical observables

condensed matter physicist and machine might not understand the Lagrangian...

Today's talk

Self learning Monte Carlo method

High-speed method with making an effective Hamiltonian/Lagrangian

configuration → heavy numerical calculation → Boltzmann weight

configuration → Effective model → Boltzmann weight

Spin systems

strongly correlated electron systems

atomic/molecular systems

Lattice QCD

Exact method: physical observables are statistically exact

Outline

- Machine learning and physics
- Self learning Monte Carlo method
- Examples in condensed matters
- Self learning Monte Carlo method for lattice QCD simulations
- Summary

Machine learning and physics

What is this?

This is a cat

You saw many cats so you know this is a cat

What is this?

This is a saiga antelope

a critically endangered antelop

https://ja-jp.facebook.com/9GAGCute/photos/saiga-antelope-a-priority-species-for-conservationthe-break-up-of-the-former-uss/816439751883761/

You did not learn this, yet.

What is a machine-learning?

Cats and others

function

f(x)

y

cat?

yes or no

x

Training process

cat!

Using many input data x, f(x)=y is obtained

Self learning Monte Carlo method

Configuration

Boltzmann weight

Simulate with use of this

What is a machine-learning?

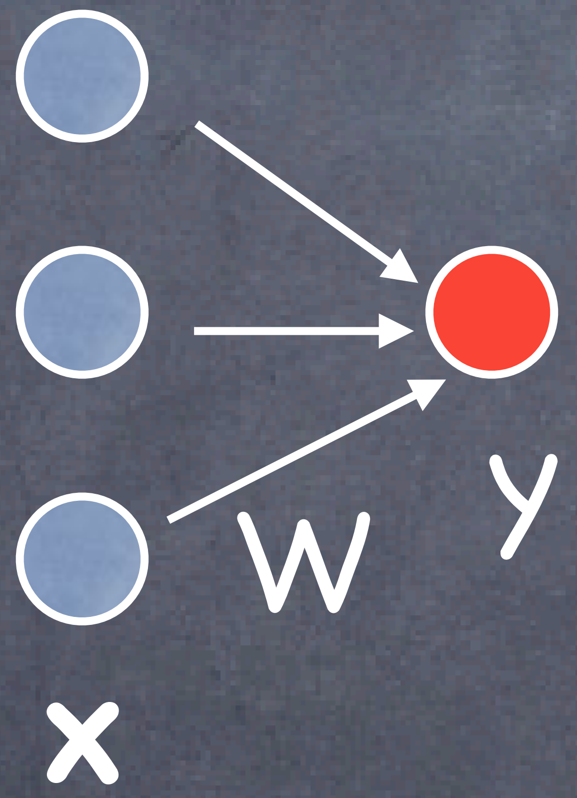
Supervised learning

Using many input data x and output data y, a function f(x)=y is obtained

Simplest case $y = ax + b$ Linear regression

Multi inputs→vector x

$y = W\mathbf{x} + b$



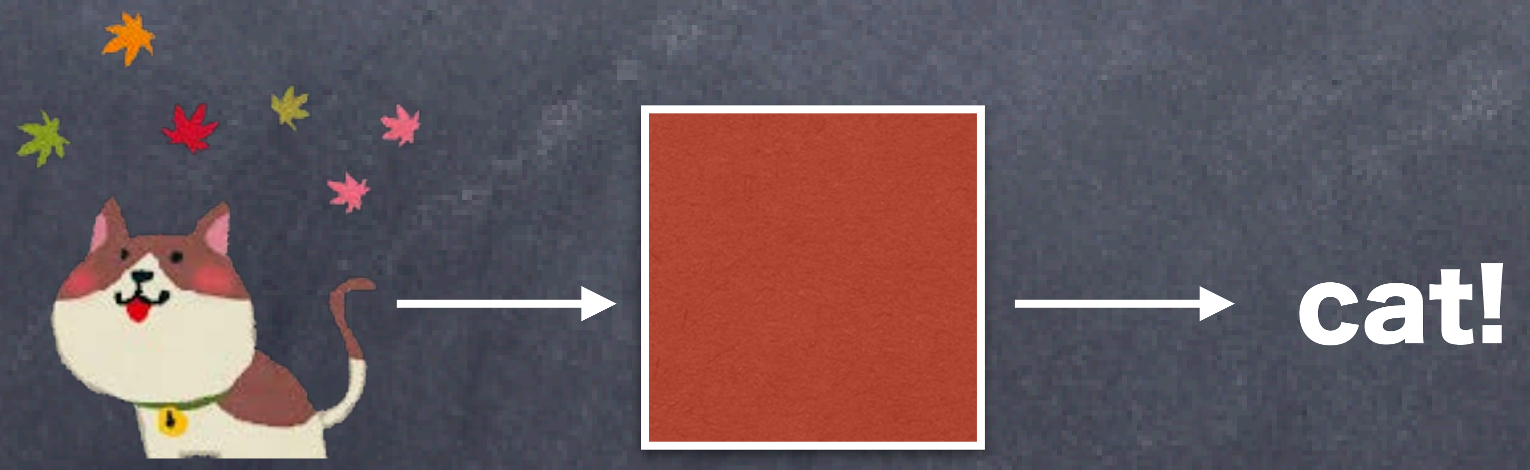
Not enough?

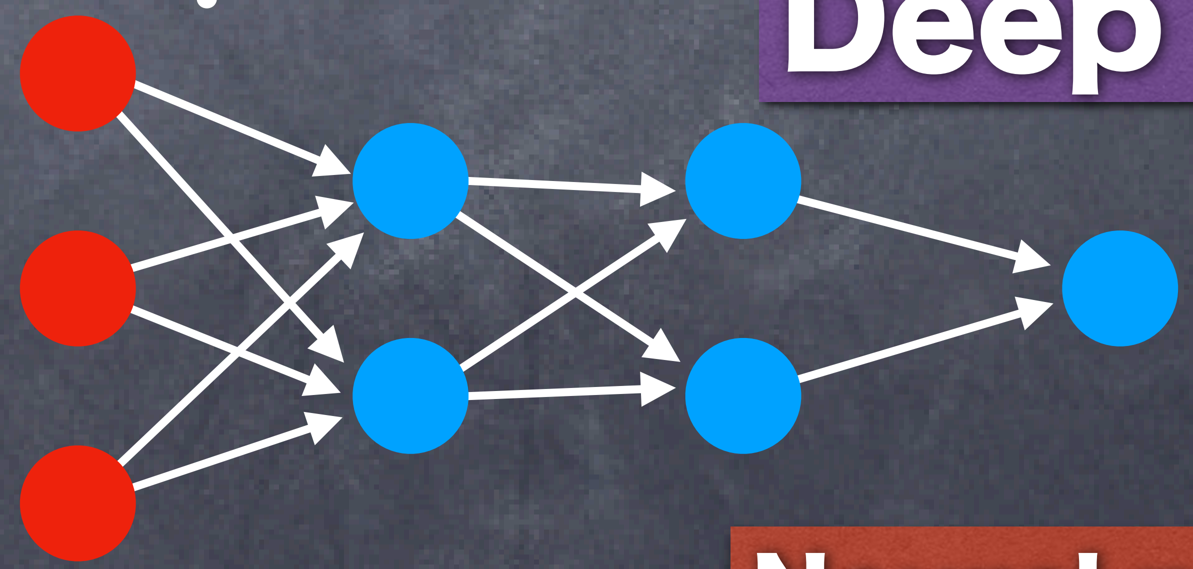
$y = W_2\,\mathbf{f}(W_1\mathbf{x} + \mathbf{b}_1) + b$ **f:non linear func.**

$y = W_3\mathbf{f}(W_2\,\mathbf{f}(W_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + b$

⋮

Deep learning





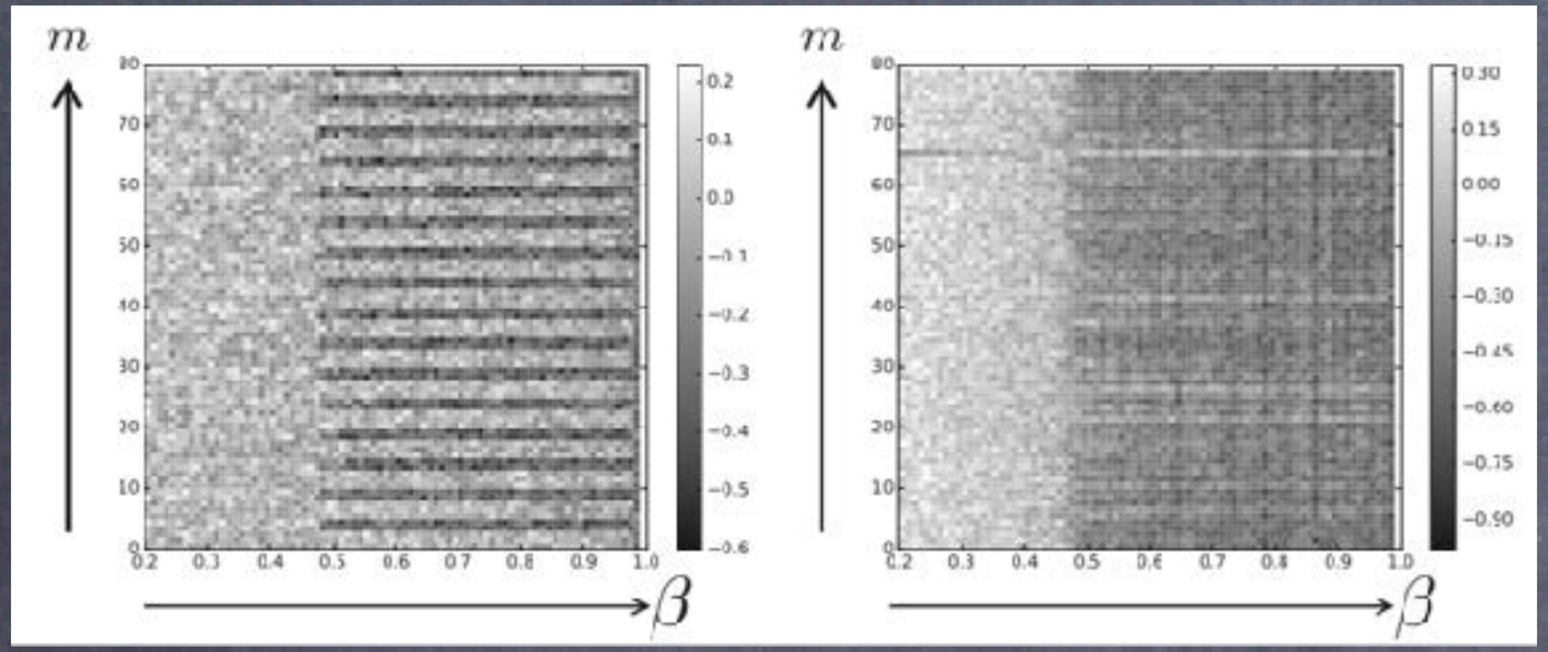
Neural networks

Applications to physics

Detecting phase transitions

Phase transition in the Ising model

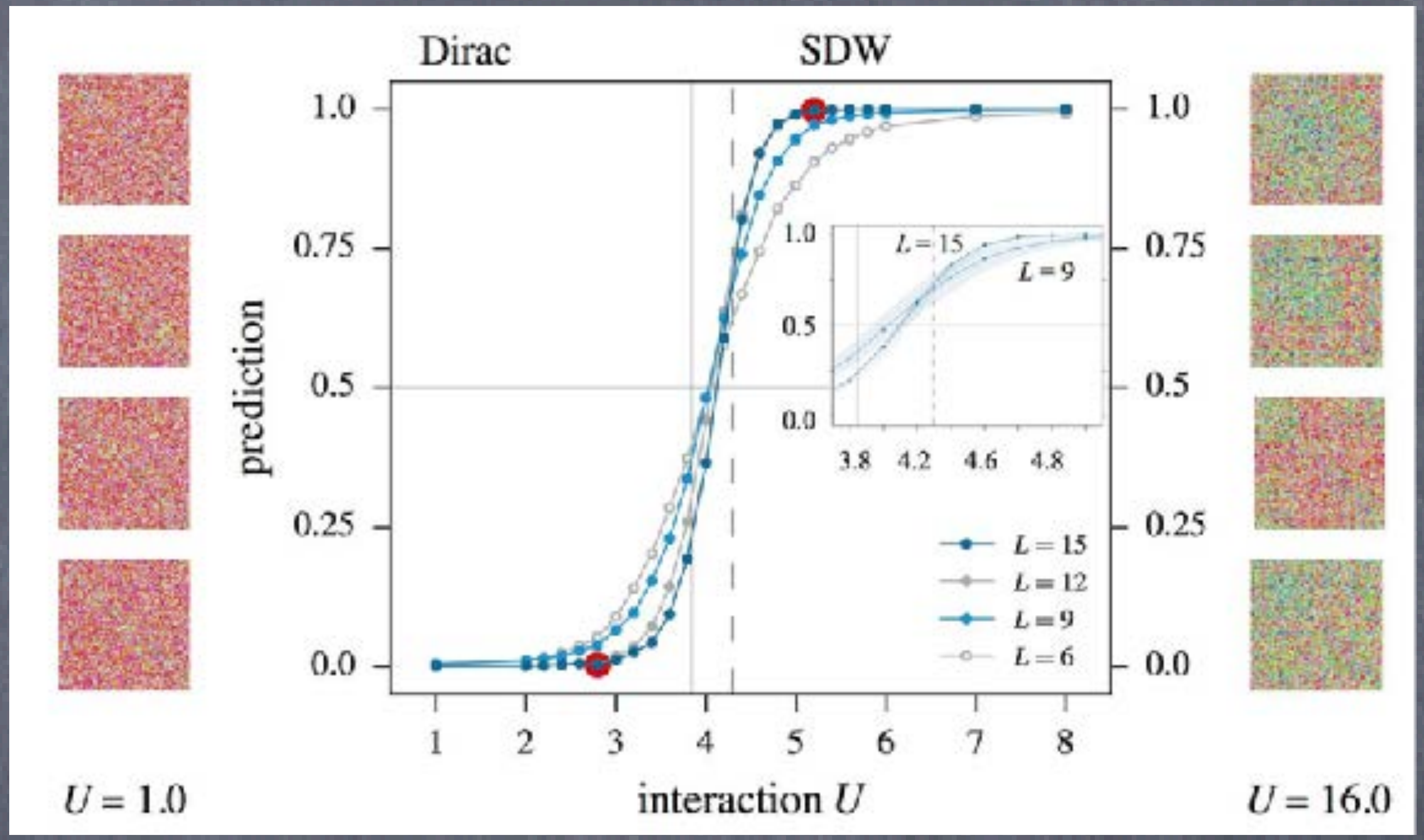
A.Tanaka and Y. Tomita, J. Phys. Soc. Jpn. 86, 063001 (2017)



Analysis of spin distribution

phase detection in the quantum system with the Fermion sign-problem

P. Broecker et al. Scientific Reports 7 8823 (2017)



Honeycomb Hubbard model

Analysis of the equal-time Green's function

Many papers => image recognitions

Application to physics

Image recognition : Detection of the phase transition etc.

Other approach?

Nature is too complicated

Build a simplest model and analyze it

Example: Throwing a ball. Where does the ball fall?

Neglecting a wind -> Not so bad

Done is better than perfect

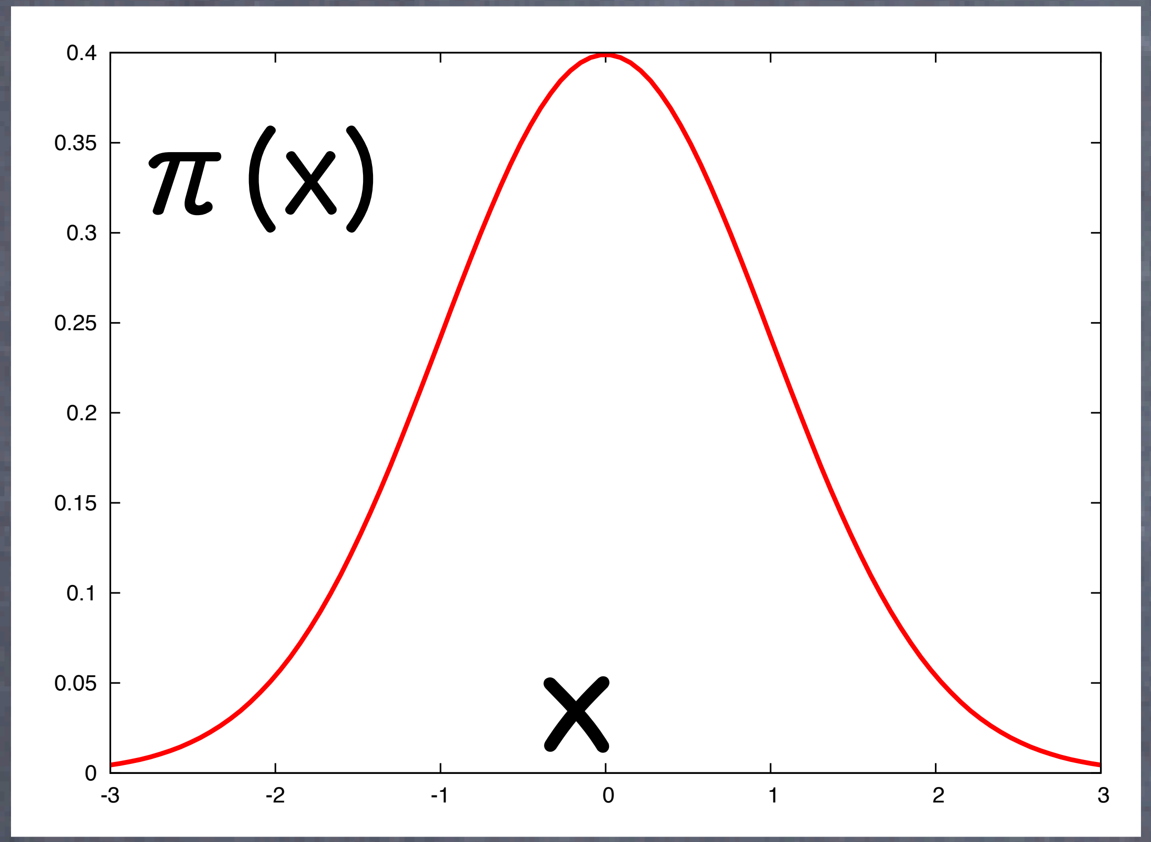
What physicists did : Build a model describing phenomena

Self-learning Monte Carlo method : A Machine builds a model

Self learning Monte Carlo method

Purpose

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



multi-dimensional integrals

$$I = E_\pi\left[h(\mathbf{x})\right] = \int_{\mathcal{X}} h(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$$

Regarding π (x) as a probability distribution function,

$$\hat{I} = \frac{1}{n}\sum_{i=1}^{n} h(\mathbf{x}^{(i)})$$

Numerical approximations of multi-dimensional integrals

Bayesian statistics, computational physics, quantum chemistry, and computational biology, etc.

To use machine learning techniques in Monte Calro method

MCMC in physics

For example: Classical spin systems

Partition function

$$Z = \sum_{i=0}^{\infty} e^{-\beta E_i}$$

Excepted value for A

$$\langle A \rangle = \frac{1}{Z} \sum_{i=0} e^{-\beta E_i} A_i$$

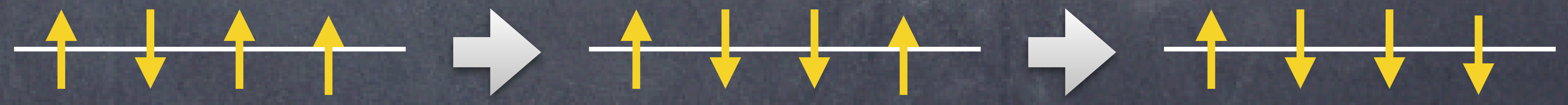
$w_i = \exp(-\beta E_i)$ as a probability distribution

→Monte Carlo simulation

$$I = \frac{1}{n} \sum_{i=1}^{n} w(\boldsymbol{x}_i)$$

i: spin configuration

$w(x_1)$ $w(x_2)$ $w(x_3)$



$x_1=(1,-1,1,1)$ $x_2=(1,-1,-1,1)$ $x_3=(1,-1,-1,-1)$

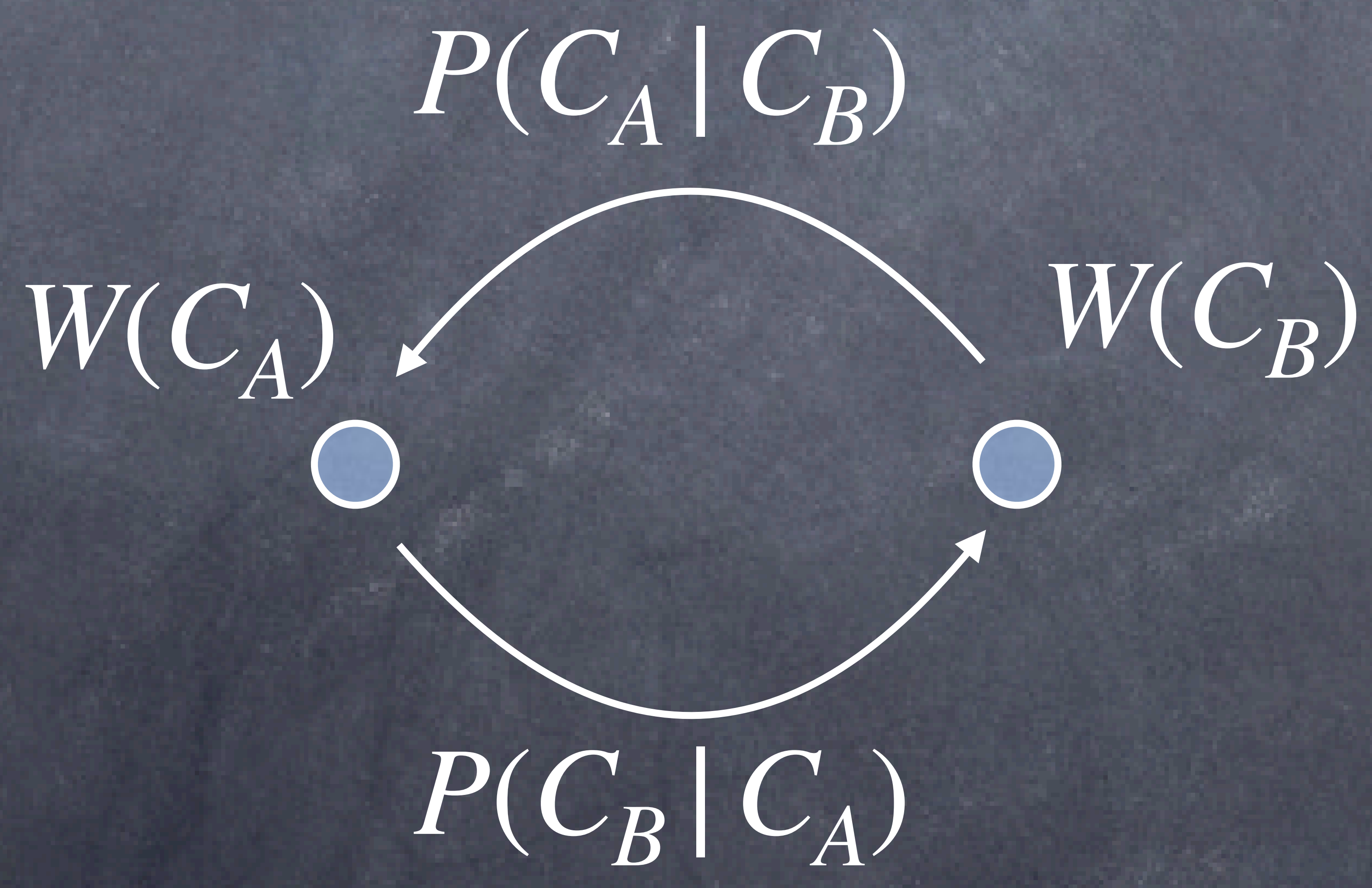
Details of MCMC

$$\int dx_1 \cdots dx_N W(x_1, \cdots, x_N) f(x_1, \cdots, x_N) \sim \sum_C f(C)$$

$C = (x_1, \cdots, x_N)$ is randomly generated with the probability $W(C)$

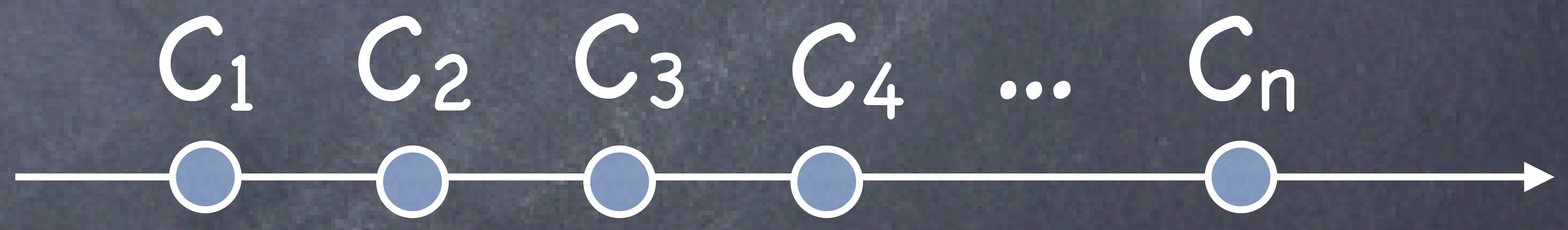
How to generate $W(C)$

We use Markov chain that has a desired distribution as its equilibrium distribution



Detailed balance condition

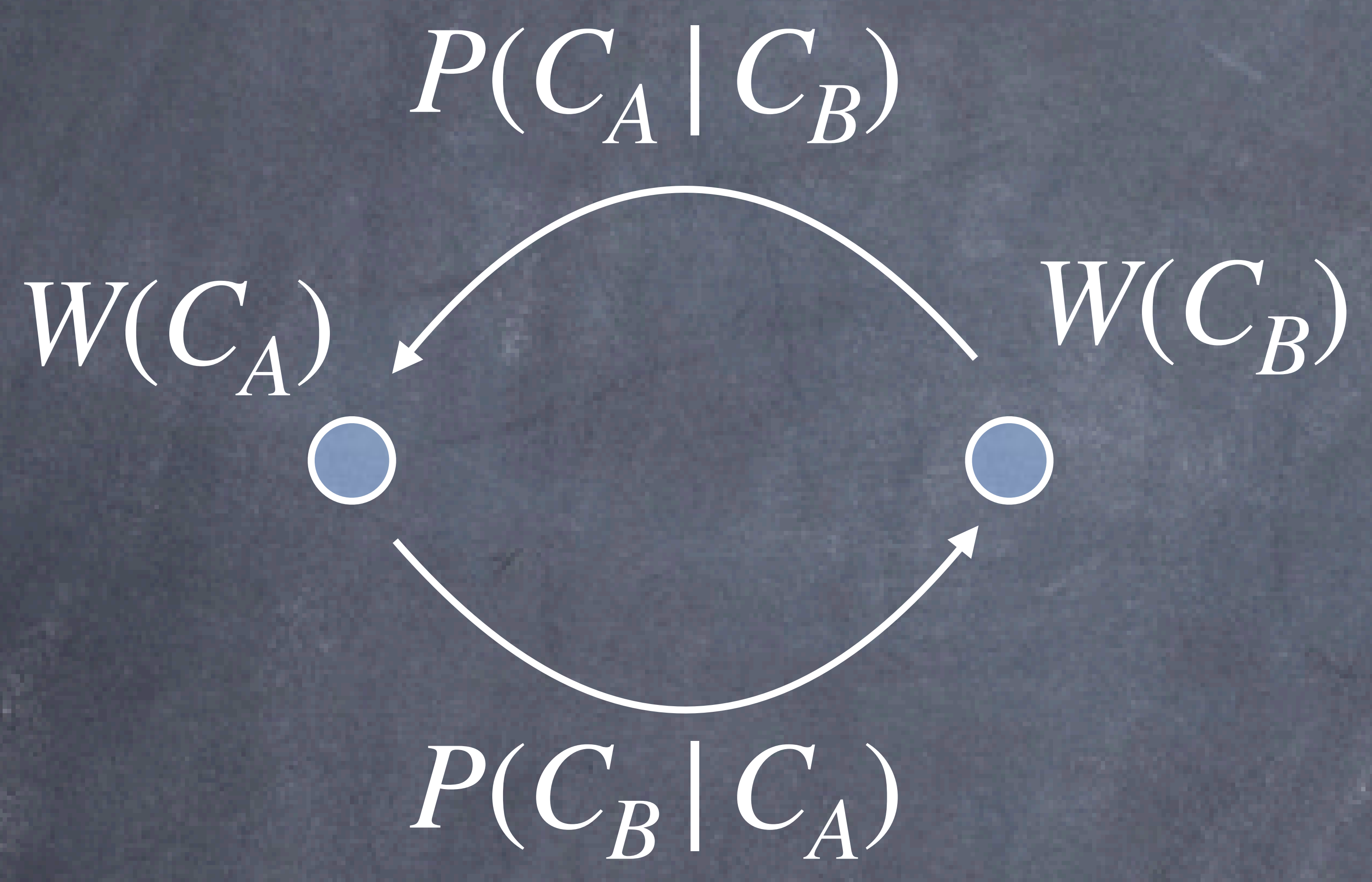
$$P(C_A | C_B) W(C_B) = P(C_B | C_A) W(C_A)$$



We can design $P(C_A|C_B)$

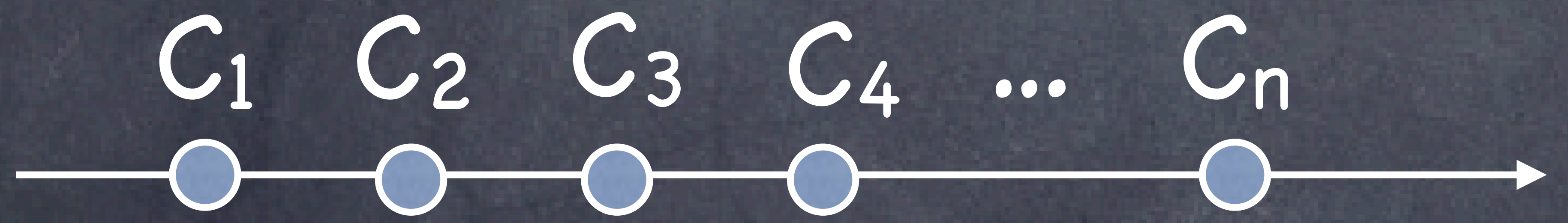
Details of MCMC

We can design P(C_A|C_B)



Detailed balance condition

$$P(C_A | C_B) W(C_B) = P(C_B | C_A) W(C_A)$$



Metropolis-Hastings algorithm

$$P(C_B | C_A) = g(C_B | C_A) A(C_B, C_A)$$

$g(C_B|C_A)$:Proposal probability

$A(C_B,C_A)$:Acceptance probability

$$\frac{A(C_B, C_A)}{A(C_A, C_B)} = \frac{W(C_B)}{W(C_A)} \frac{g(C_A | C_B)}{g(C_B | C_A)}$$

Acceptance ratio from C_A to C_B

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

This acceptance ratio should be high!

How to improve MCMC?

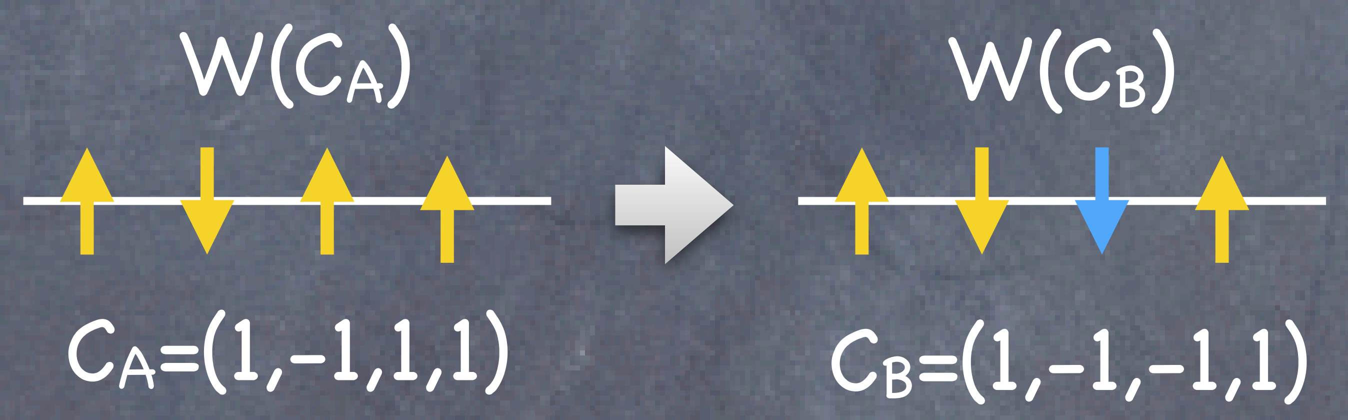
Acceptance ratio from C_A to C_B

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

We have to choose C_B with high acceptance ratio $A(C_B, C_A)$

Solution 1: Local updates

If C_B is similar to C_A, $W(C_B)$ might be similar to $W(C_A)$



One randomly chooses a **single** site and proposes a new configuration by changing the variable on this site

Good: general

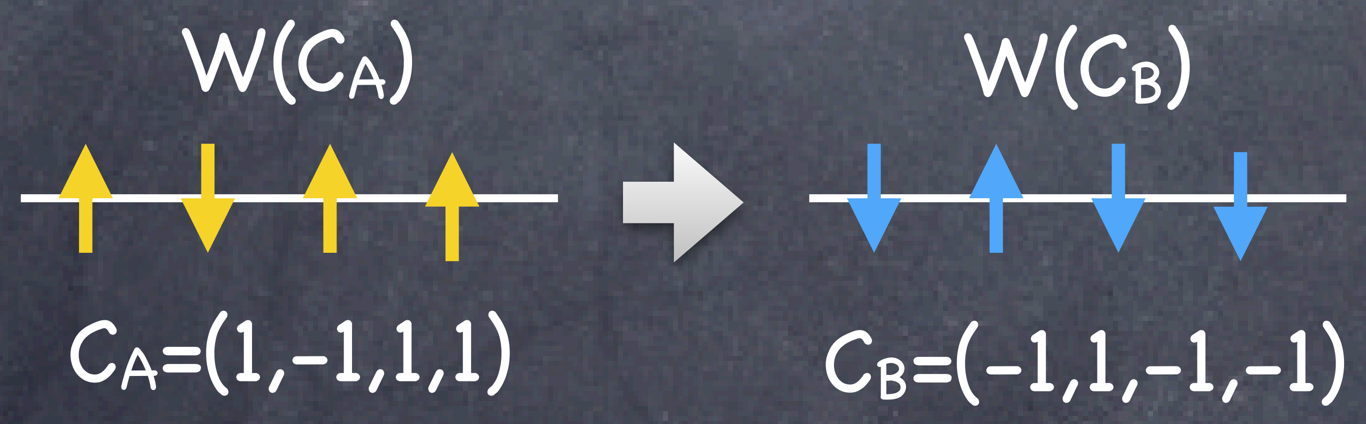
Bad: difference between C_A and C_B is small

Long autocorrelation time

Solution 2: Global updates

We choose C_B with the use of knowledge of a system

Swendsen-Wang, Wolff, worm, etc.



Variables on an extensive number of sites are **simultaneously** changed in a single MC update

Good: difference between C_A and C_B is not small

Bad: it is hard to find it

Self-learning MC

Acceptance ratio from C_A to C_B

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

We have to choose C_B with high acceptance ratio $A(C_B, C_A)$

Solution 3: Self-learning updates

Usually, ratio of the proposal probability is one $\frac{g(C_A|C_B)}{g(C_B|C_A)} = 1$

we can change this!

Another Markov chain with the probability W'(C)

$C_A \quad C_2 \quad C_3 \quad C_4 \quad \ldots \quad C_B$

$P'(C_B|C_A)$: probability from CA to CB

This Markov chain proposes C_B from C_A!

Detailed balance condition

$$P'(C_A|C_B)W'(C_B) = P'(C_B|C_A)W'(C_A)$$

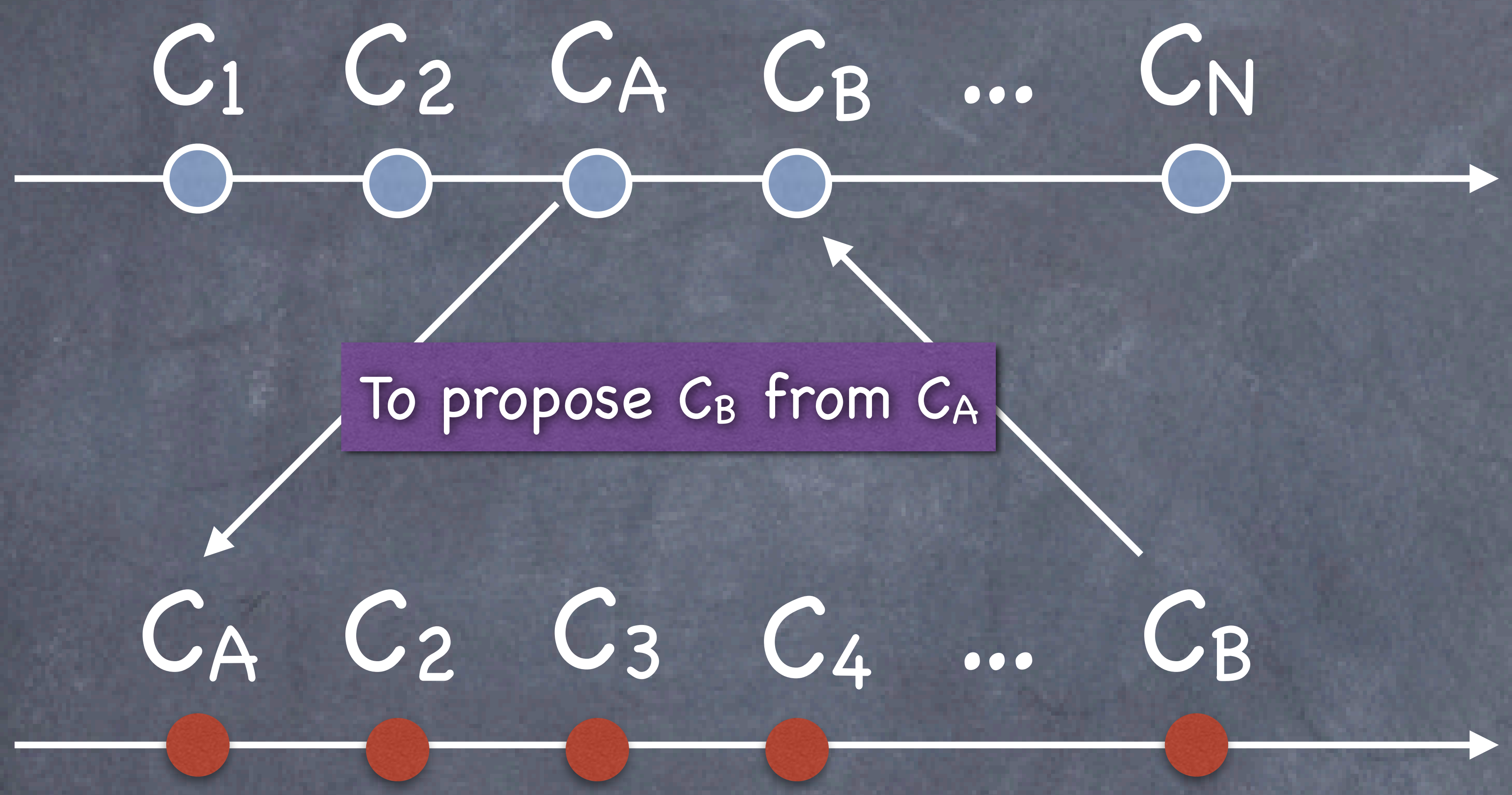
$P'(C_B|C_A)=g(C_B|C_A)$

$$\frac{g(C_A|C_B)}{g(C_B|C_A)} = \left(\frac{W'(C_B)}{W'(C_A)}\right)^{-1}$$

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

Concept of SLMC

Markov chain with the probability W(C)



Another Markov chain with the probability W'(C)

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

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

If W'(C)=W(C),

the acceptance ratio is **one**!

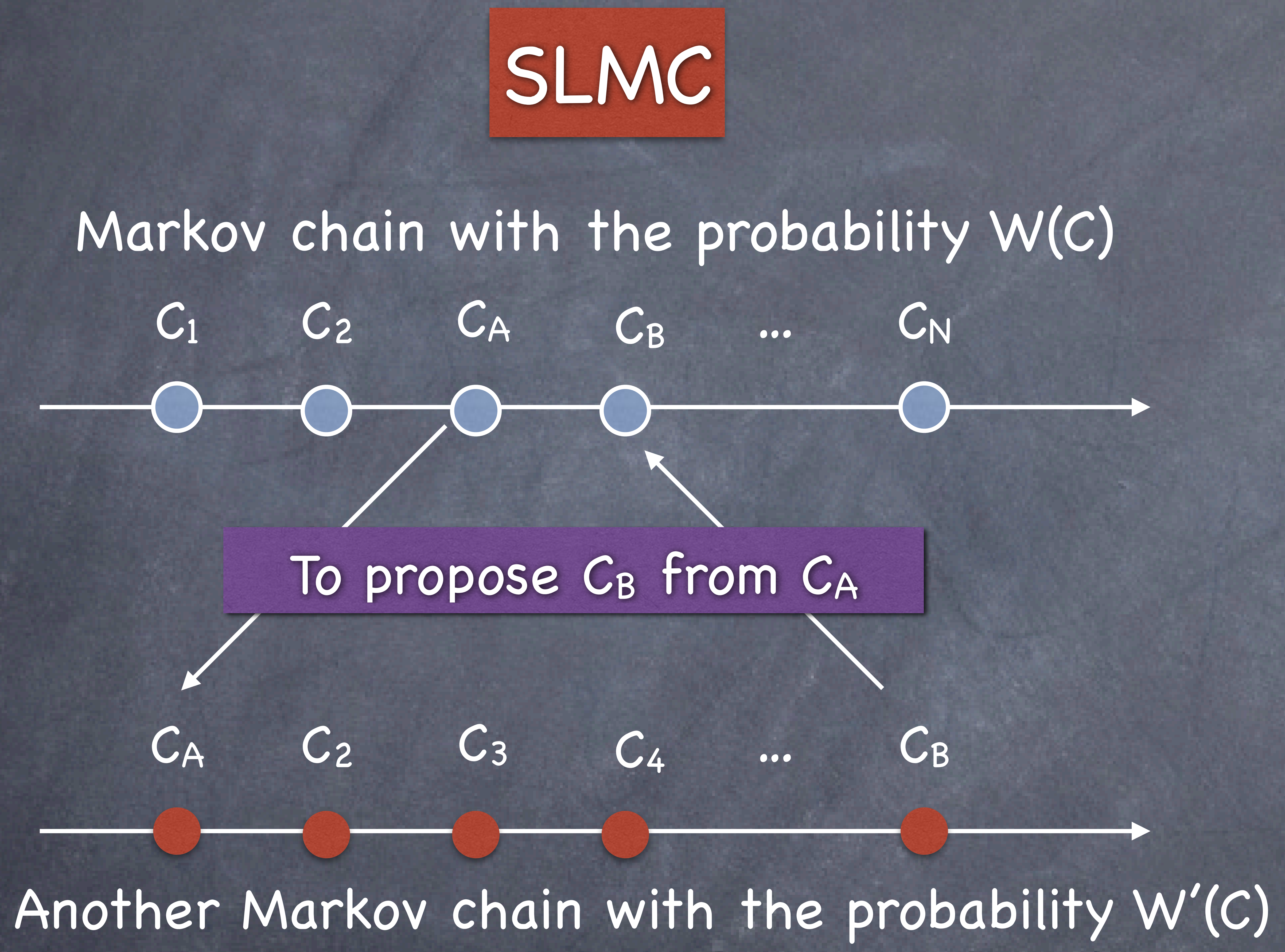
If the computational cost of the proposal Markov chain is small, we can speed up the simulation

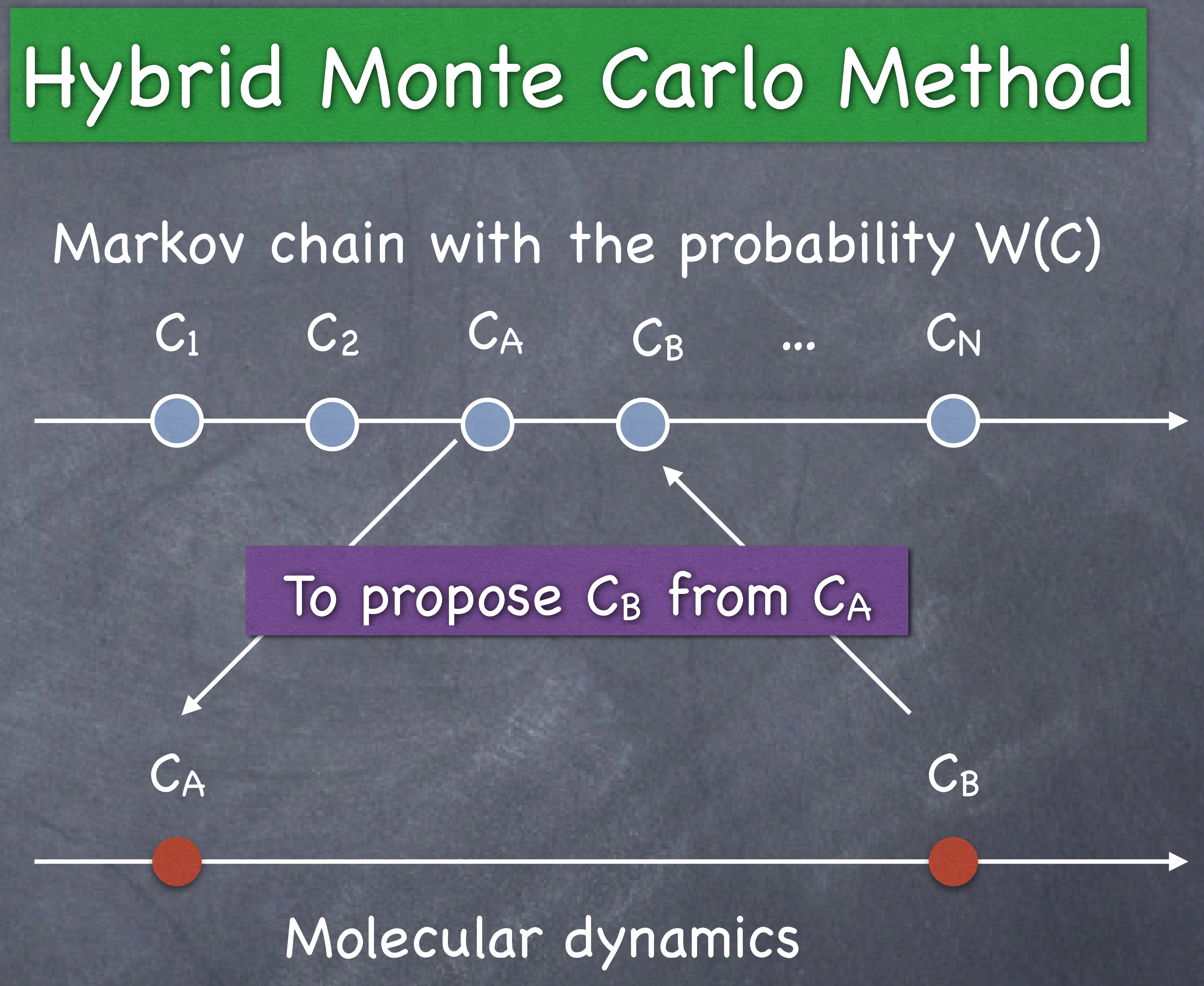
How to construct the Markov chain with W'(C)?

->Machine learning technique!

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

SLMC and HMC





These two are exact!

Examples in condensed matters

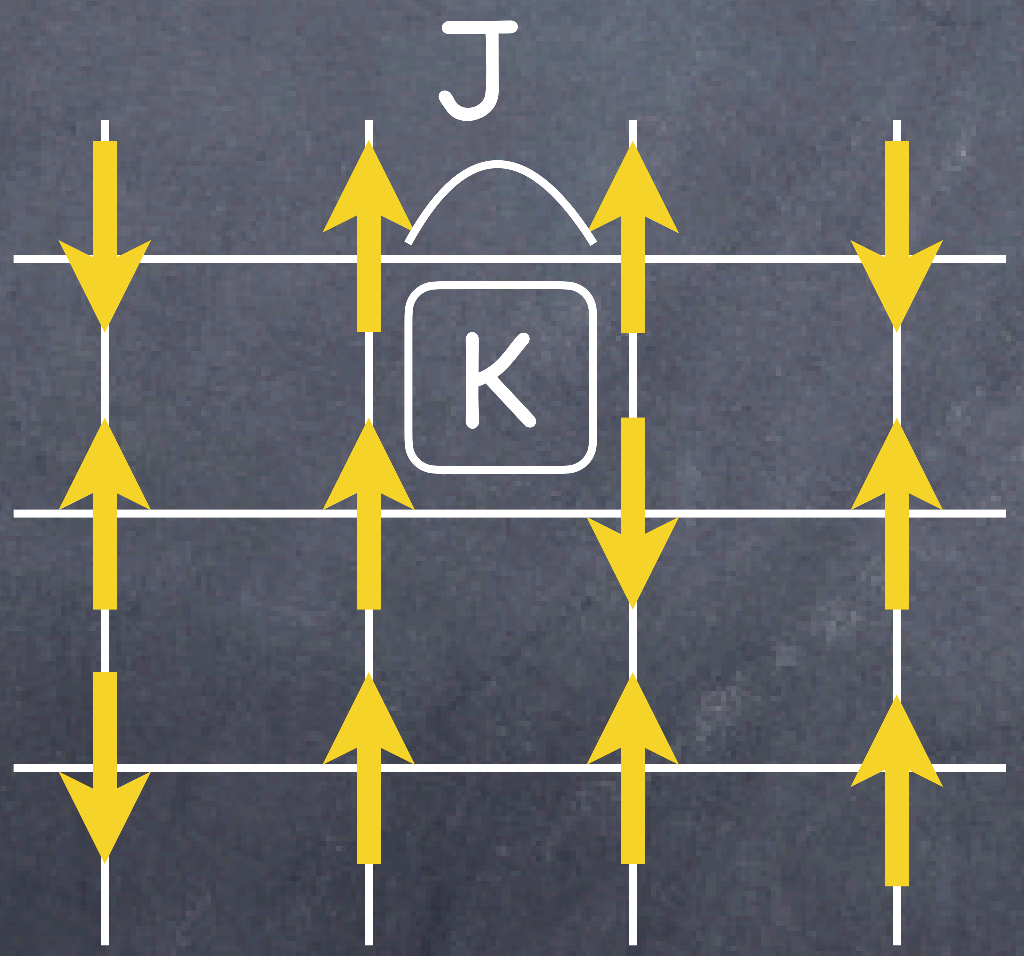
Classical spin system

Hamiltonian: classical model on a two-dimensional square lattice

Original model

$$H = -J \sum_{\langle ij \rangle} S_i S_j - K \sum_{ijkl \in \square} S_i S_j S_k S_l,$$

Four-body interaction:
No efficient global update method



Log w=-β H(S_i)

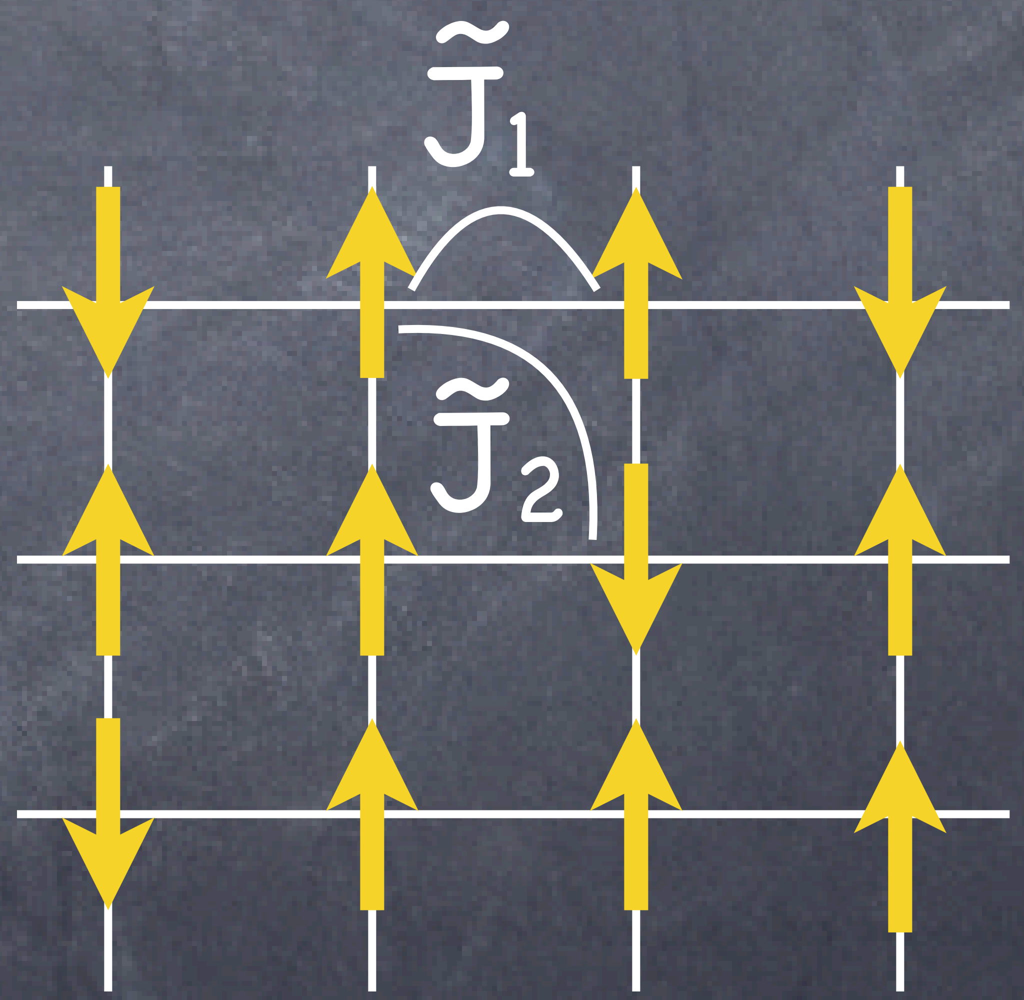
Effective model

$$H_{\text{eff}} = E_0 - \tilde{J}_1 \sum_{\langle ij \rangle_1} S_i S_j - \tilde{J}_2 \sum_{\langle ij \rangle_2} S_i S_j - \cdots,$$

Only two-body interactions:
Wolff global update method

Gathering the configurations and their weights

Linear regression



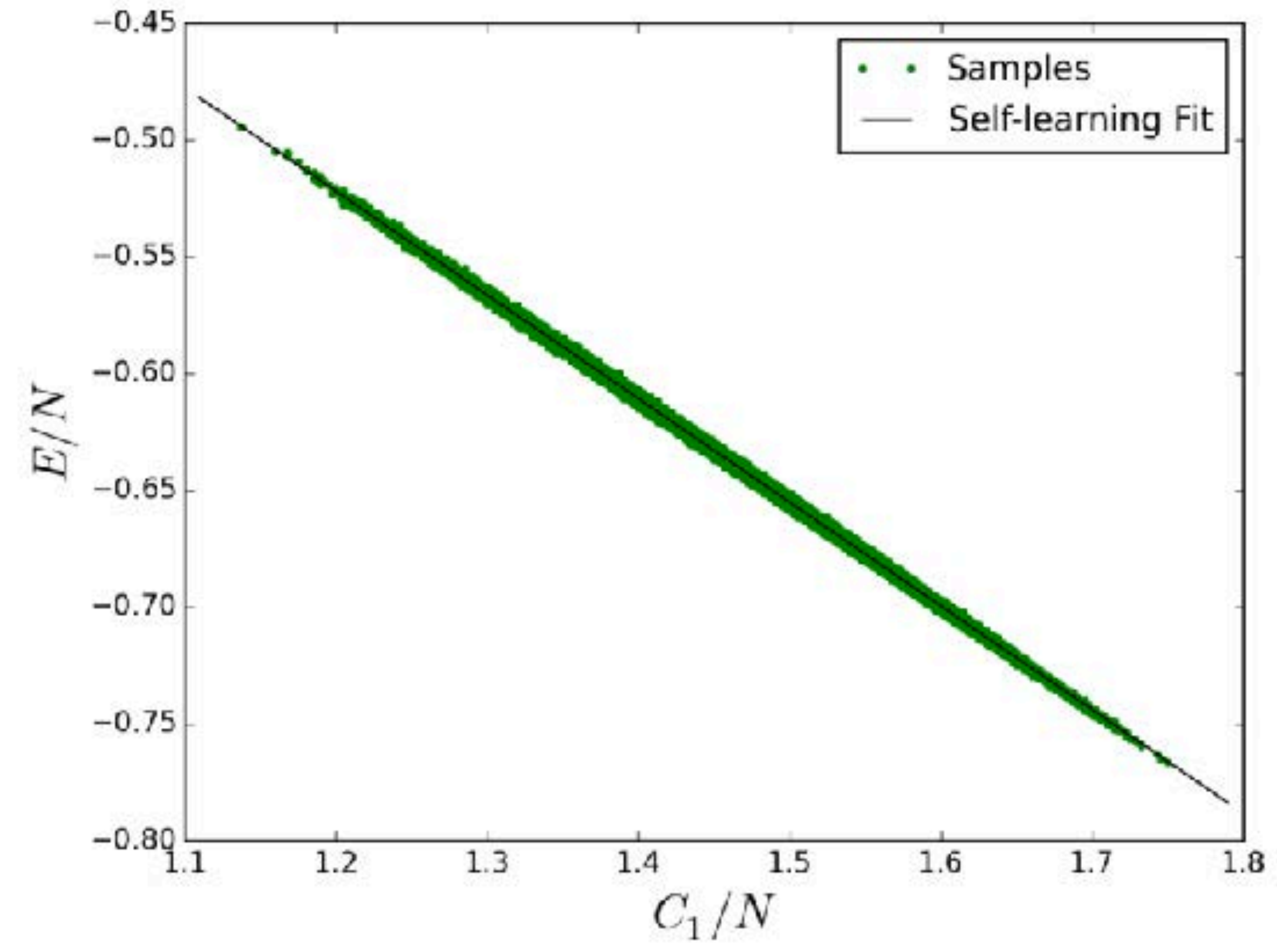
Log w_{eff}=-β H_{eff}(S_i)

Classical spin system

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

Original model

$$H = -J \sum_{\langle ij \rangle} S_i S_j - K \sum_{ijkl \in \square} S_i S_j S_k S_l,$$

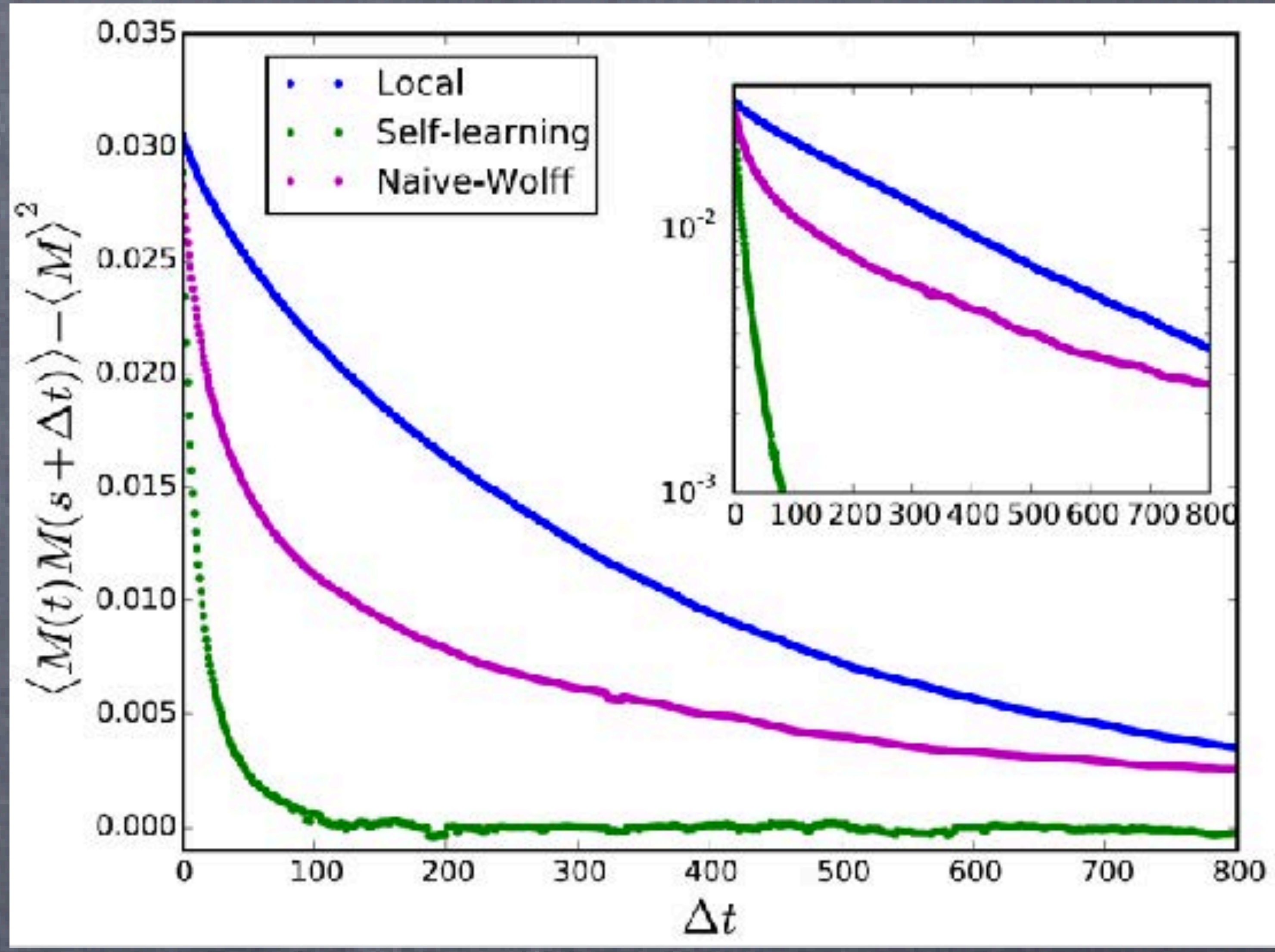


C_1:nearest neighbor spin-spin correlation

Effective model

$$H_{\mathrm{eff}} = E_0 - \tilde{J}_1 \sum_{\langle ij \rangle_1} S_i S_j - \tilde{J}_2 \sum_{\langle ij \rangle_2} S_i S_j - \cdots,$$

Decay of the autocorrelation function



	$\tilde{J}_1$	$\tilde{J}_2$	$\tilde{J}_3$	Mean error
Train 1	1.2444	−0.0873	−0.0120	0.0009
Train 2	1.1064			0.0011

Only one parameter J_1 reproduces the original weights!

Double exchange model

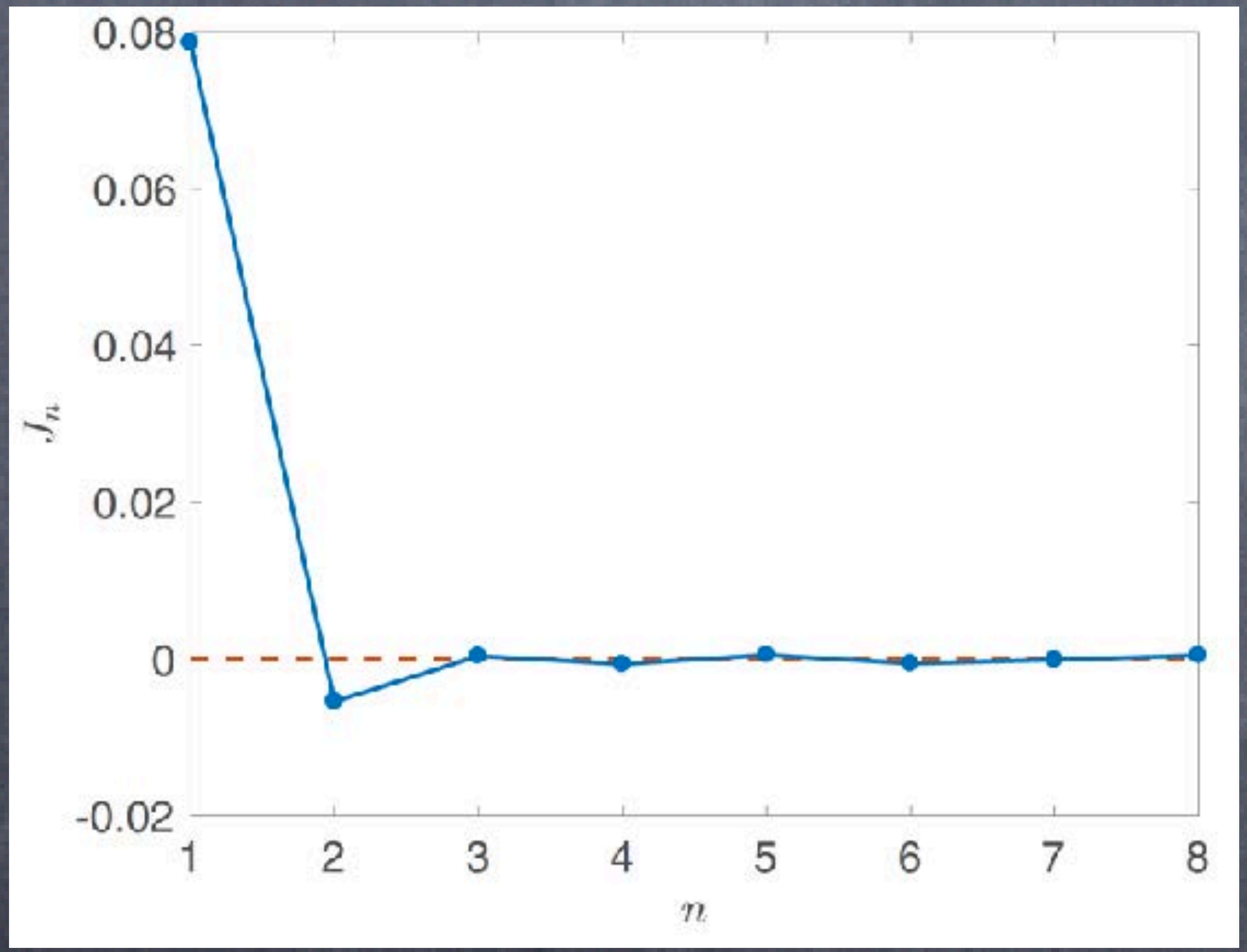
J. Liu, H. Shen, Y. Qi, Z. Y. Meng, and L. Fu, Phys. Rev. B 95, 241104(R)(2017)

Original model

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

$$Z = \sum_{\phi} \det[\mathbf{I} + e^{-\beta H_f[\phi]}] \equiv \sum_{\phi} W[\phi].$$

matrix



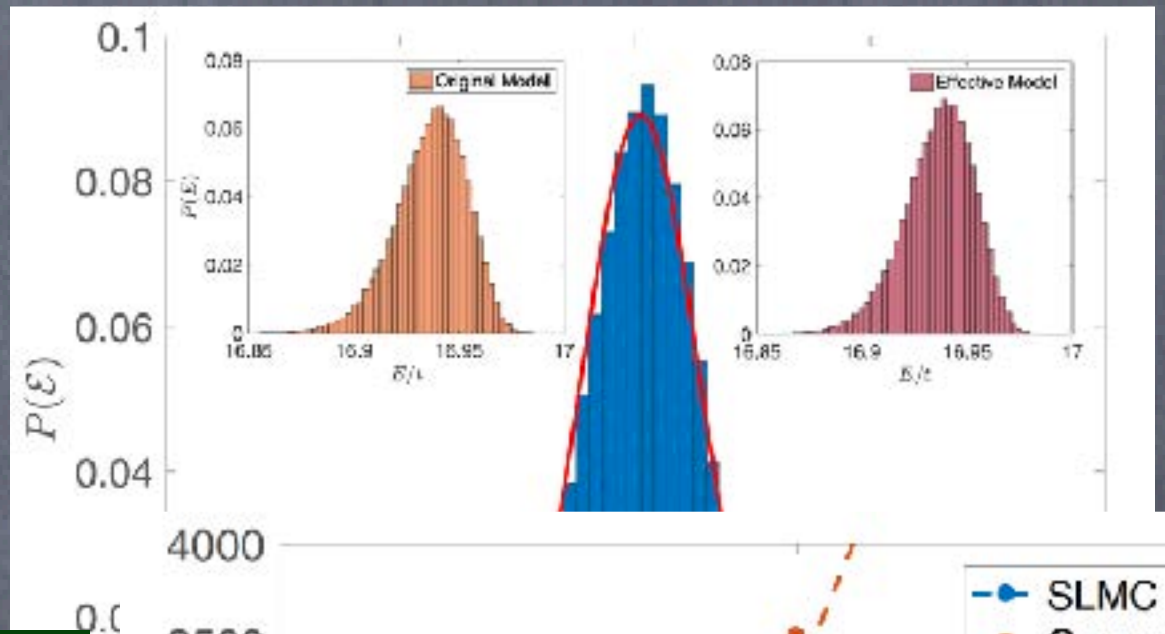
Oscillating function-> RKKY interaction

Effective model

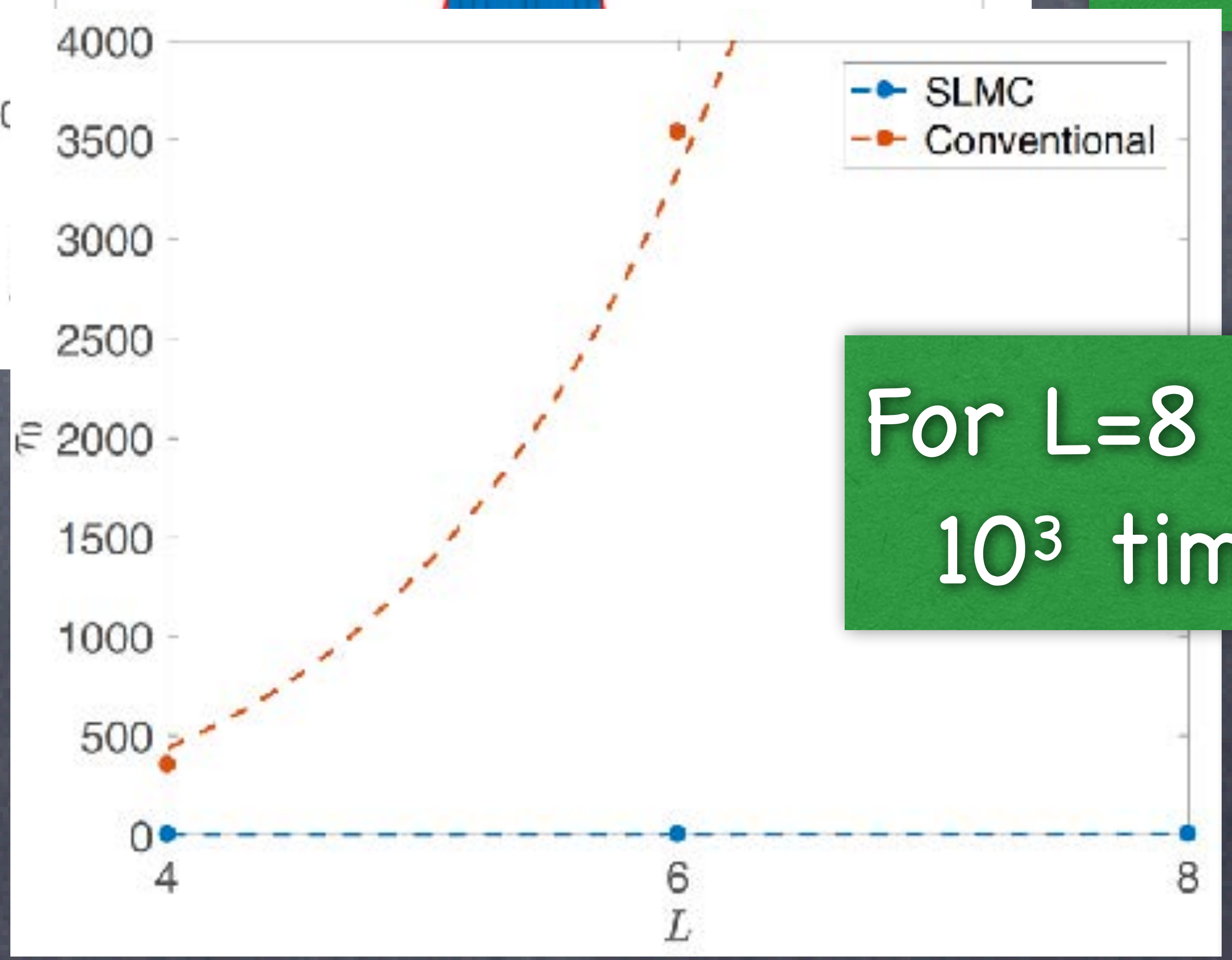
$$H_{\text{eff}} = E_0 - J_1 \sum_{\langle ij \rangle_1} \vec{S}_i \cdot \vec{S}_j - J_2 \sum_{\langle ij \rangle_2} \vec{S}_i \cdot \vec{S}_j - \cdots,$$

$$W[\phi] \simeq e^{-\beta H_{\text{eff}}[\phi]}$$

scalar



Almost same weight distribution



Autocorrelation time

System size

For L=8 (8x8x8 cubic)
10^3 times speedup!

Continuous-time quantum Monte Carlo method

Anderson impurity model

continuous-time auxiliary-field method (CTAUX)

$$H = H_0 + H_1$$
$$H_0 = -(\mu - U/2)(n_\uparrow + n_\downarrow) + \sum (V c_\sigma^\dagger a_{p,\sigma} + h.c.) + \sum_{\sigma,p} \epsilon_p a_{p,\sigma}^\dagger a_{p,\sigma} + K/\beta,$$
$$H_1 = U(n_\uparrow n_\downarrow - (n_\uparrow + n_\downarrow)/2) - K/\beta,$$

$$H_1 = -\left(\frac{K}{2\beta}\right) \sum_{s=\pm1} e^{\gamma s(n_\uparrow - n_\downarrow)}$$

Ising-like auxiliary fields

Partition function

$$\frac{Z}{Z_0} = \mathrm{Tr}\left[e^{-\beta H_0} T_\tau e^{-\int_0^\beta d\tau H_1(\tau)}\right],$$
$$= \sum_{n=0} \int_0^\beta d\tau_1 \cdots \int_{\tau_{n-1}}^\beta d\tau_n \left(\frac{K}{2\beta}\right)^n \frac{Z_n(\{s_i,\tau_i\})}{Z_0}.$$

$$Z_n(\{s_i,\tau_i\})/Z_0 \equiv \prod_{\sigma=\uparrow,\downarrow} \det N_\sigma^{-1}(\{s_i,\tau_i\}),$$
$$N_\sigma^{-1}(\{s_i,\tau_i\}) \equiv e^{V_\sigma\{s_i\}} - G_{0\sigma}^{\{\tau_i\}}(e^{V_\sigma\{s_i\}} - 1)$$

$$Z_0 \equiv \mathrm{Tr}\ e^{-\beta H_0}$$ $$(G_{0\sigma}^{\{\tau_i\}})_{ij} = g_\sigma(\tau_i - \tau_j)$$

Markov chain on different Feynman diagrams

configurations

Number of spins changes on continuous imaginary-time axis

What is an effective model?

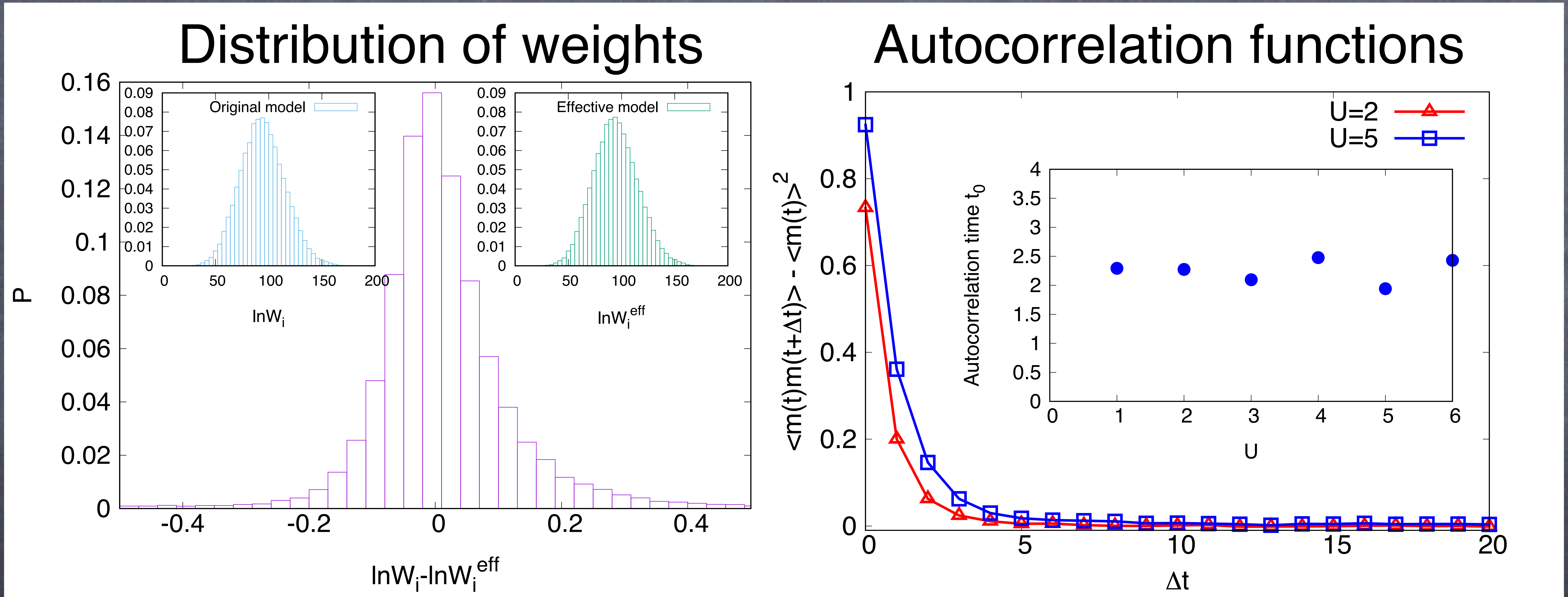
Self-learning continuous-time QMC

YN, H. Shen, Y. Qi, J. Liu and L. Fu, Phys. Rev. B 96, 161102(R) (2017)

Effective model

$$-\beta H_n^{\mathrm{eff}}(\{s_i,\tau_i\}) \equiv \frac{1}{n}\sum_{i,j} J(\tau_i-\tau_j)s_is_j + \frac{1}{n}\sum_{i,j} L(\tau_i-\tau_j) + f(n)$$

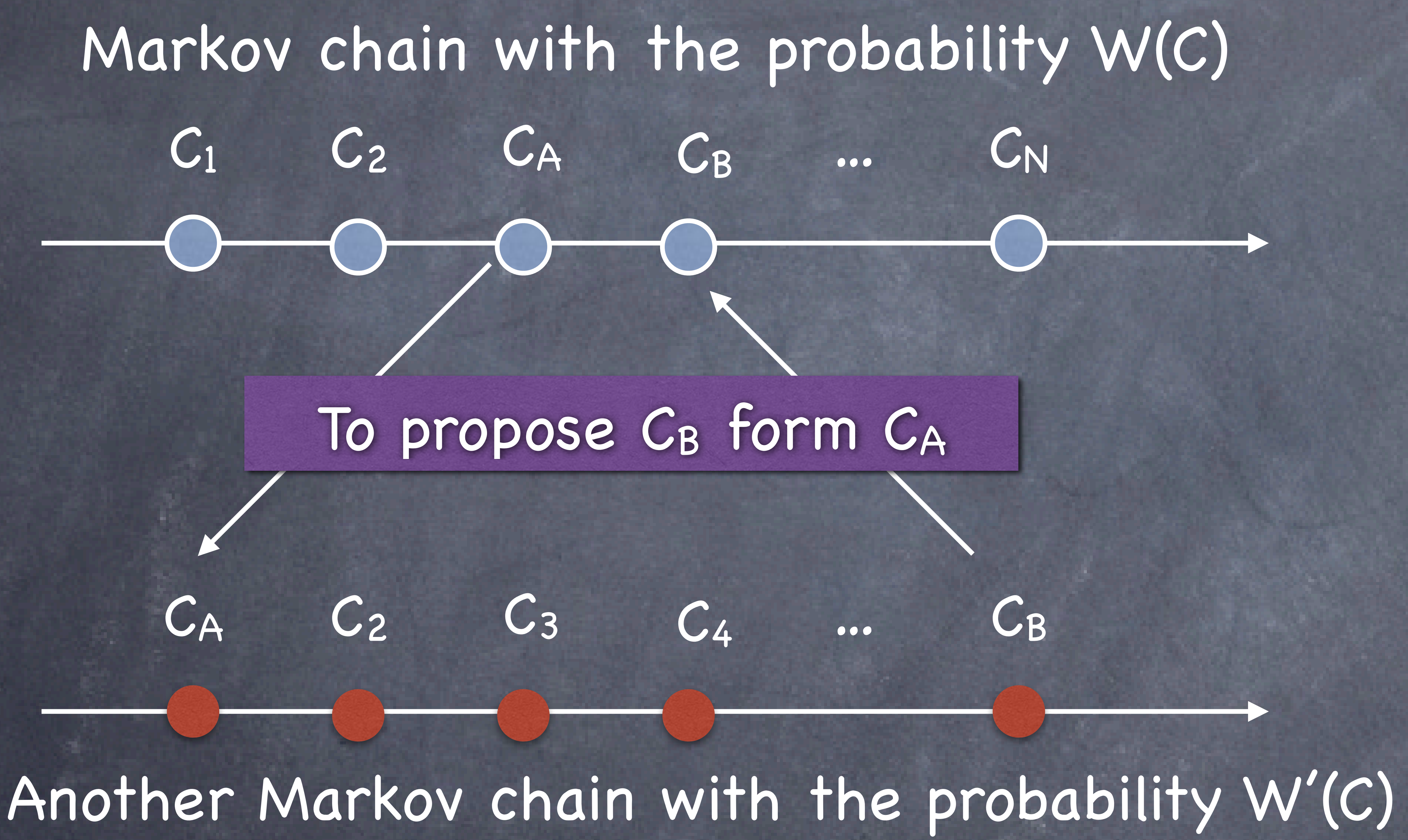
We call this the diagram generating function (DGF)



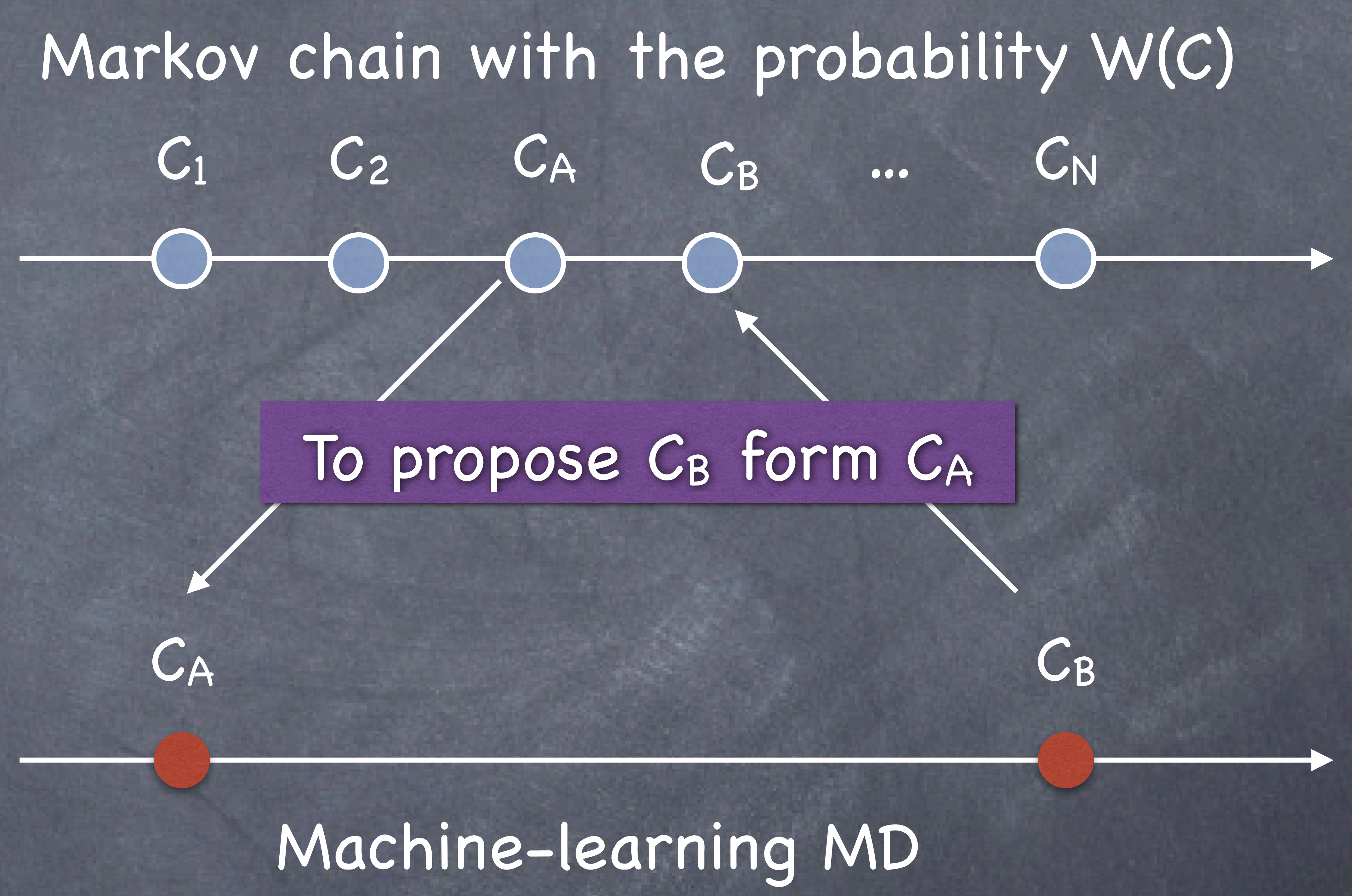
The DGF can propose good configurations!

Self-learning Hybrid Monte Carlo method (SLHMC)

SLMC



SLHMC



We developed the SLHMC for molecular simulations

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

When does the SLMC become better?

1. Long autocorrelation time

    Near critical point, one needs to get many configurations

2. Heavy computational cost for calculating the Boltzmann weight

    Calculation of the Fermion determinant is heavy

    one can integrate out the fermion with the use of the SLMC

    Calculation based on the density functional theory is heavy

    one can use the neural networks to imitate Hamiltonian

Self learning Monte Carlo method for Lattice QCD simulations

Lattice QCD to me

Formula One in computational physics

Lattice QCD

Many cutting edge technologies

supercomputer, hybrid Monte Carlo, parallel computing, GPU computing...

We can learn many things from the Lattice QCD

?

Condensed matter physics

Many body physics

Lattice QCD package

You can start it in 10 minutes

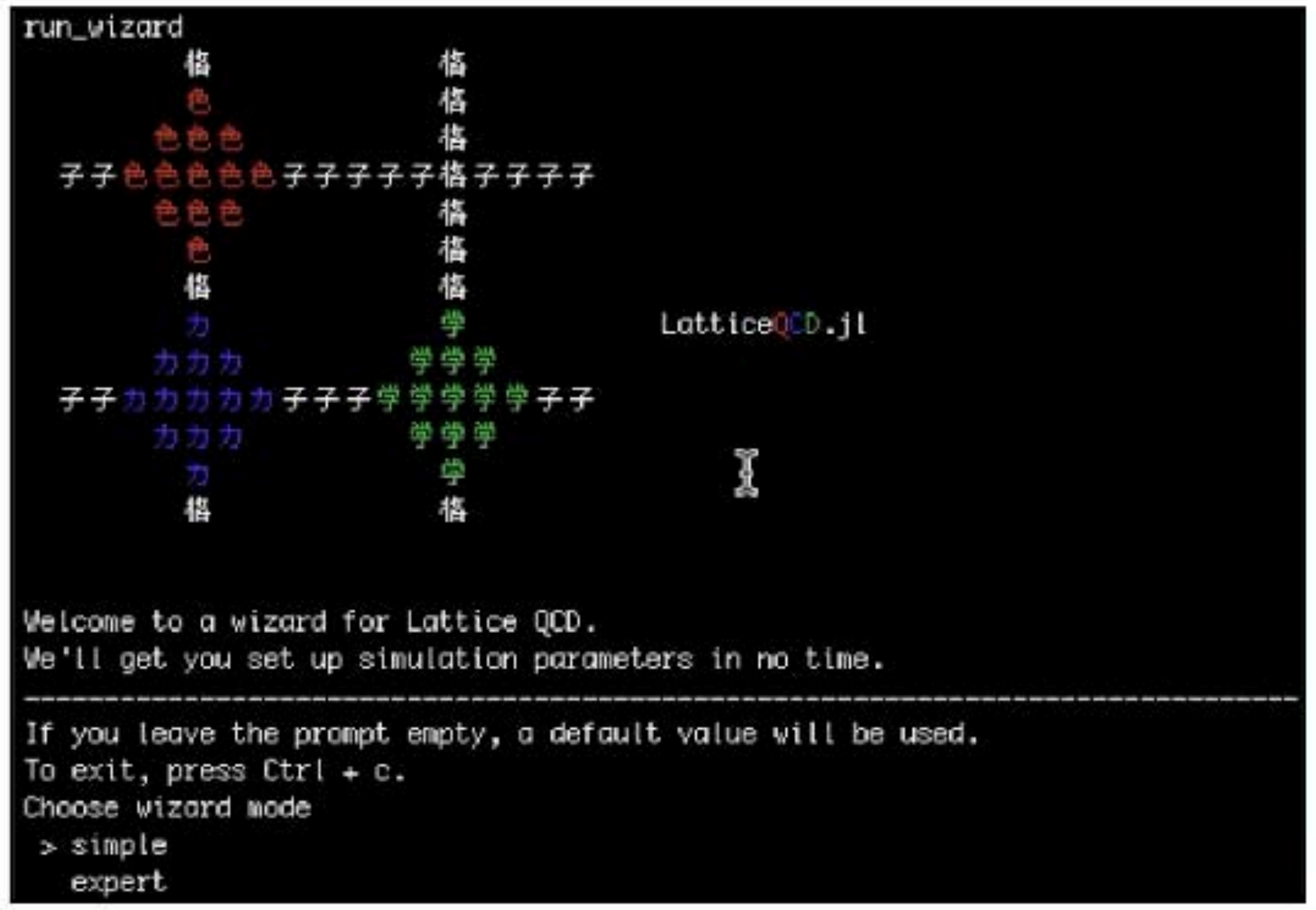
We have made a public LQCD code by Julia language:
https://github.com/akio-tomiya/LatticeQCD.jl

Easy and quick start on laptop/desktop: HMC/heatbath/SLMC + Measurements

Compatible speed with a Fortran code

You can start in 3 steps

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



julia

LatticeQCD.jl

Lattice QCD

Eqs given from Dr. Tomiya

QCD in 3 + 1 dimension

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

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

QCD in Euclidean 4 dimension

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

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

Lattice regularization

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

a is lattice spacing (cutoff)

They are "same" up to irreverent operators $\quad \mathrm{Re}\ U_{\mu\nu} \sim \frac{-1}{2} g^2 a^4 F_{\mu\nu}^2 + O(a^6)$

Physical observables

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

$$\langle \mathcal{O} \rangle = \frac{1}{Z}\int \mathcal{D}U e^{-S_{\mathrm{eff}}[U]}\mathcal{O}(U) \qquad S_{\mathrm{eff}}[U] = S_{\mathrm{gauge}}[U] - \log\det(D\!\!\!/[U] + m)$$

We can use the MCMC!

Effective action

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

Actions about Gauge field　　　Actions about fermions

SLMC in condensed matters

classical spins + fermions -> classical spins

There is a small coupling expansion
The SLMC can reach to large coupling region

SLMC in lattice QCD

gauge fields + fermions -> gauge fields

There is a large mass expansion

Can the SLMC reach to small mass region?

SLMC for gauge systems

Acceptance ratio

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

$g(C_B|C_A)$:Proposal probability

$$A(U_A \to U_B) = \min\left(1, \frac{e^{-(S(U_B)-S_{\mathrm{eff}}(U_B))}}{e^{-(S(U_A)-S_{\mathrm{eff}}(U_A))}}\right)$$

S(U): Two color QCD (gluons + 4 of quarks), 4dim

S_{eff}(U): QCD action without fermions (1/m expansion)

Plaquette + Rect + Polyakov loop + (crown, chair, etc.)

Couplings are determined from linear regression

parameters are tuned to make acceptance high.

Update method: heatbath. Any update, which satisfies the detailed balance is fine

(non-reversible update is fine)

Target system

System

Two color QCD (plaquette + staggered(not rooted))

HMC and SLMC, T=0 & T>0

Parameters

Ls = 4,6,8, Lt=Ls(T=0) or 4 (T>0)

m=0.5,...,0.05

beta = [0.8-4.0] including a phase transition for T>0

Effective action

Plaquette+rect+Polyakov loop +(chair,crown,Bended polyakov)

(automatic code generator is used)

Observables

Plaquette (P), Polyakov loop (L)

Chiral condensate

and their Binder cumulant (4th order cumulant)

Topological charge is not relevant for this volume

$$B_L^4(\beta) = \frac{\left\langle (\delta L)^4 \right\rangle}{\left\langle (\delta L)^2 \right\rangle^2}$$

$$\delta L = L - \langle L \rangle$$

Code

Fully written in Julia lang.

(Different from the public one (old ver is used in the paper))

How to compare different algorithms?

In general, different algorithms can not compare

There are several ways.

Measure elapsed time?
Count by operation by operation?
Most numerically expensive part?

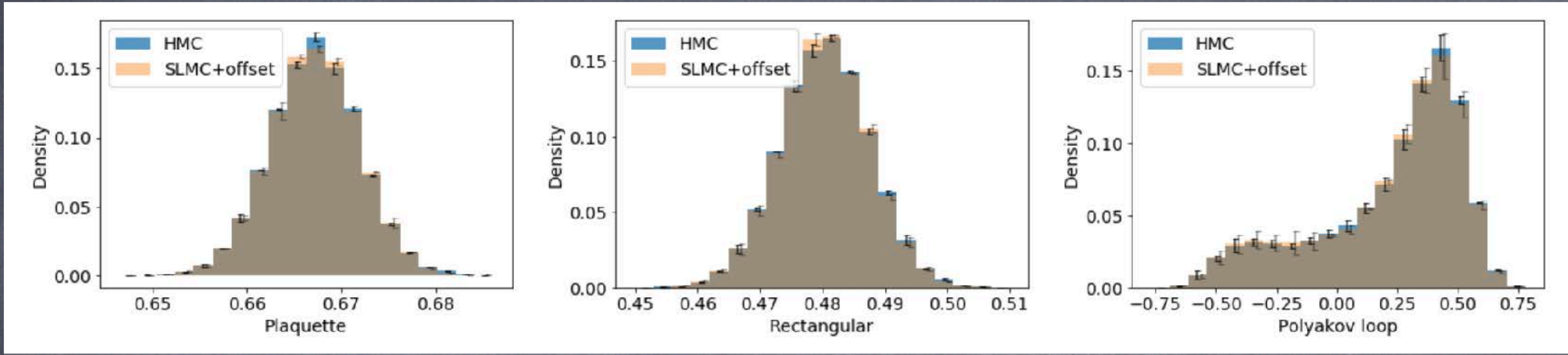
implementation dependent
(hardware and software)

<u>We count it by the number of Metropolis test</u>

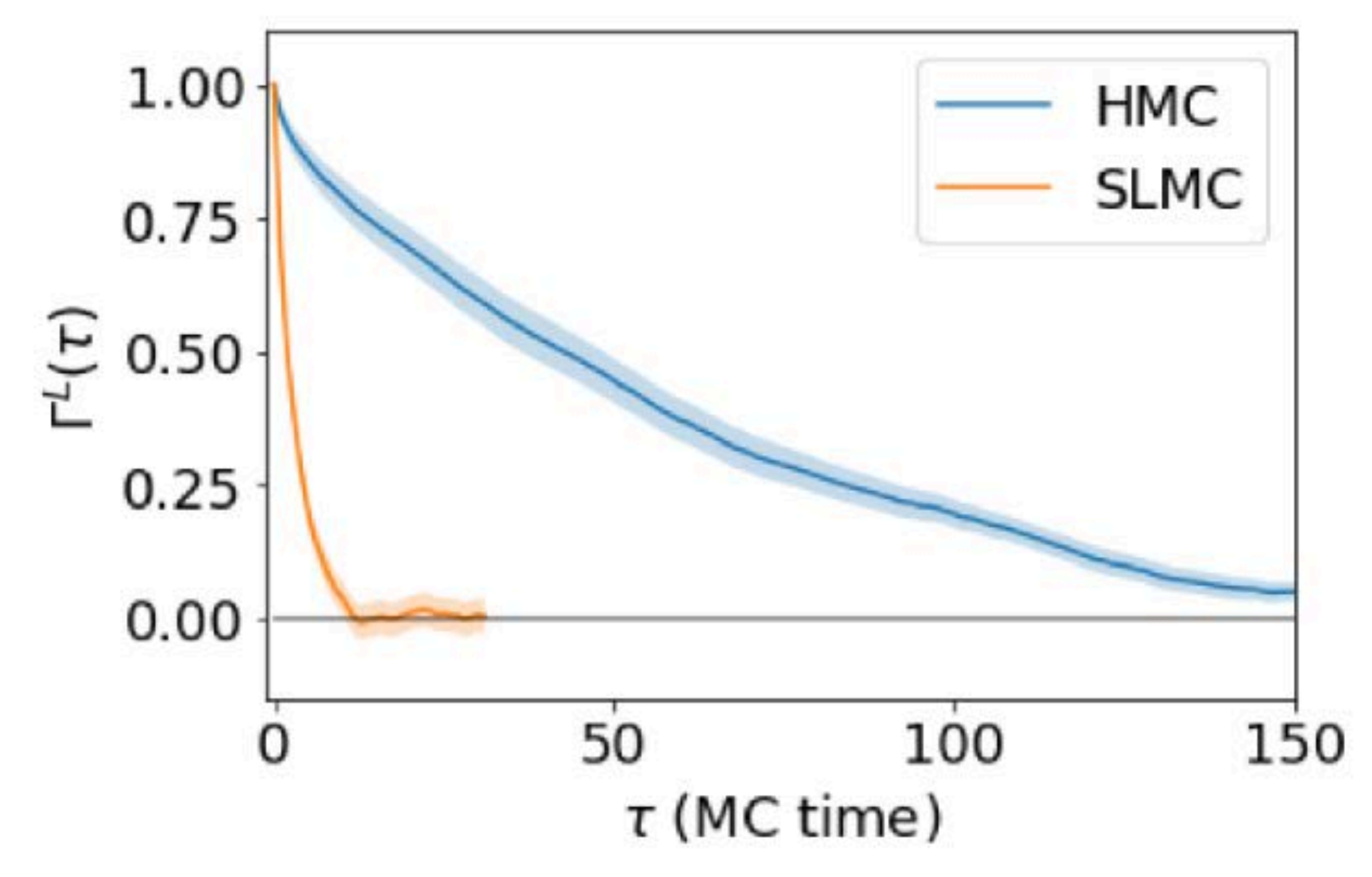
Namely our "MD time" is counted by the number of Metropolis test

T=0 results

YN, A. Tomiya, and A. Tanaka, arXiv:2010.11900



Observables are consistent (expected)

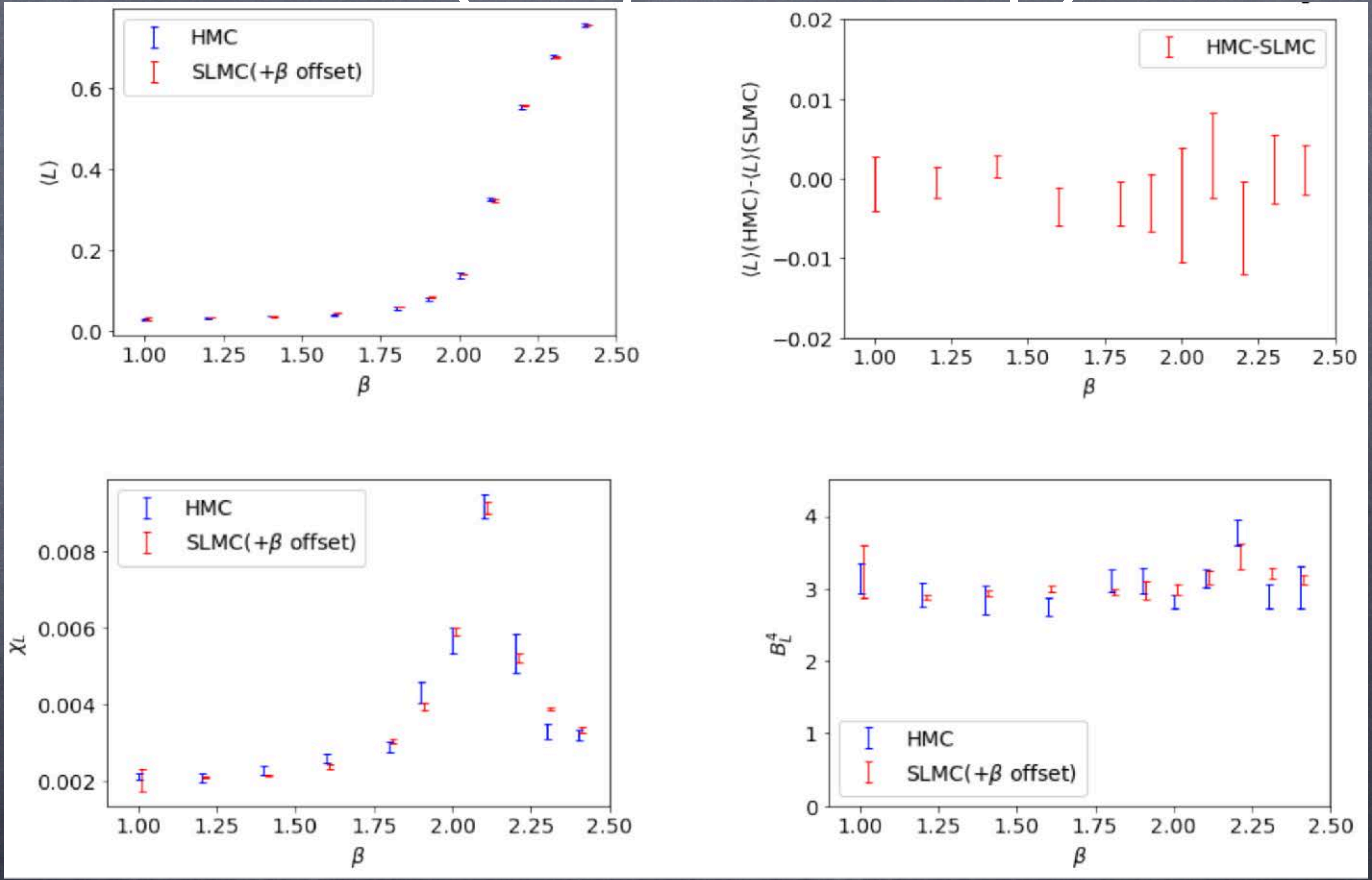


Autocorrelation for Polyakov loop

Very short

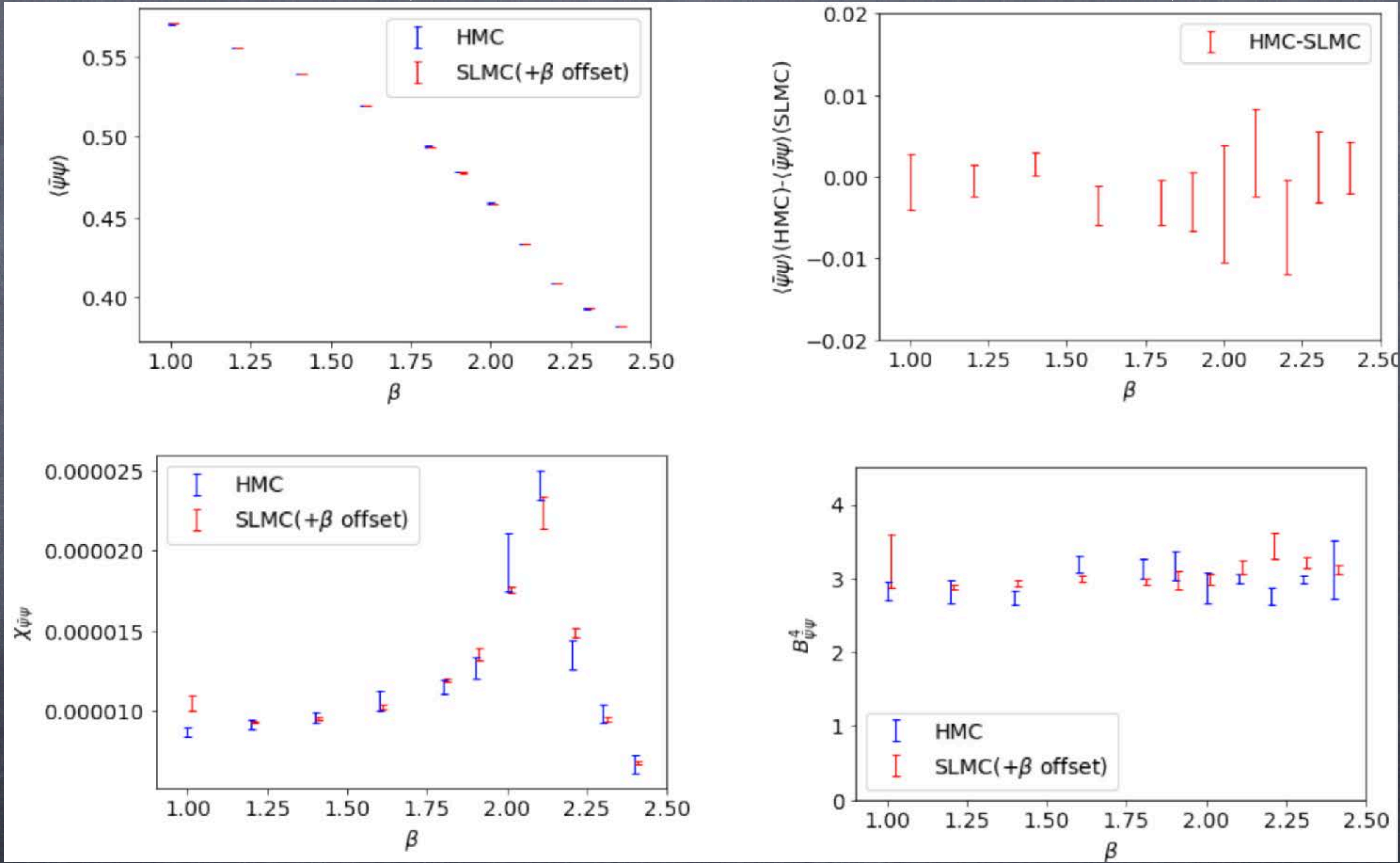
YN, A. Tomiya, and A. Tanaka, arXiv:2010.11900

T>0 results (Polyakov loop)



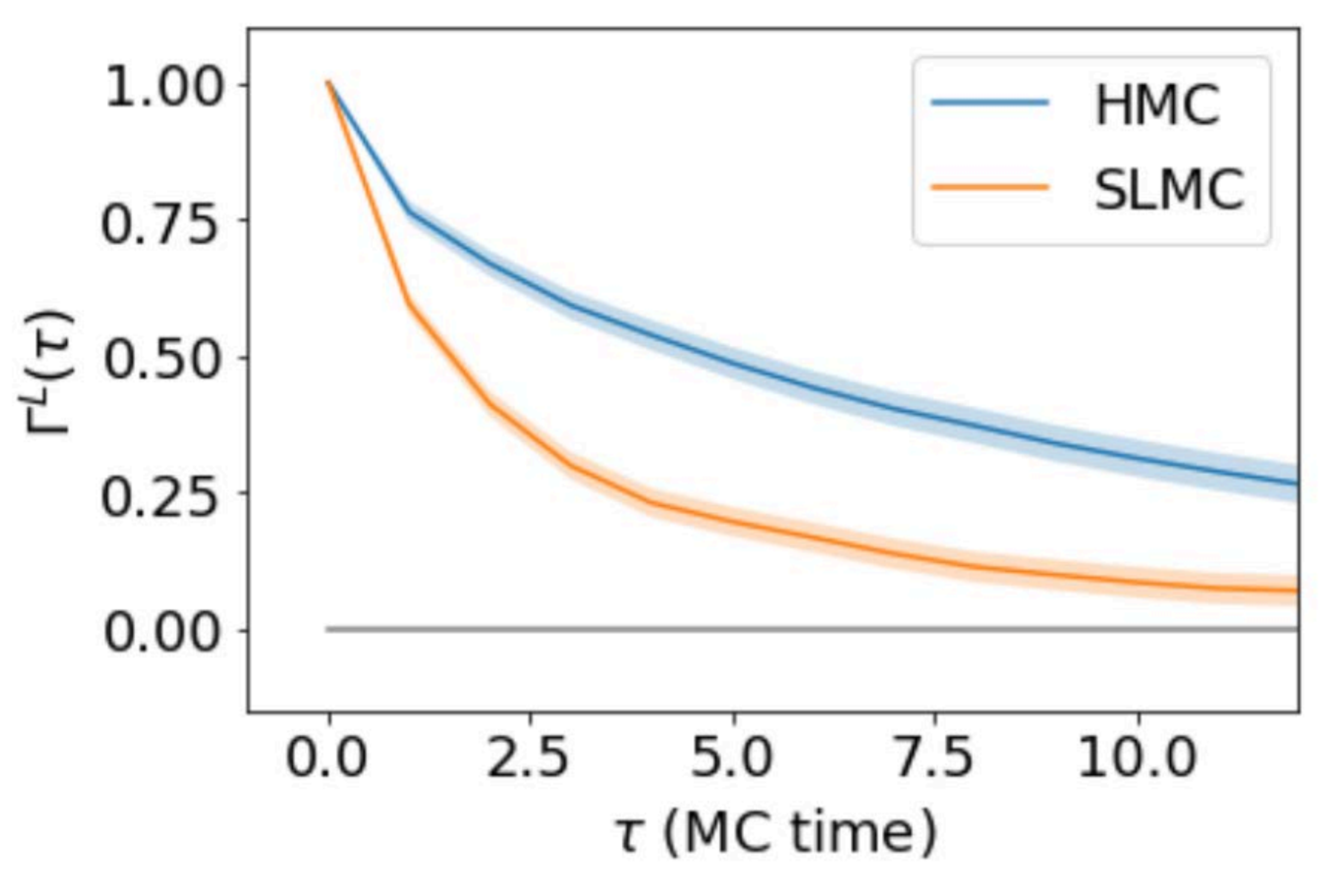
YN, A. Tomiya, and A. Tanaka, arXiv:2010.11900

T>0 results (Chiral condensate)



YN, A. Tomiya, and A. Tanaka, arXiv:2010.11900

T>0 results



Autocorrelation around the critical regime. it is slightly better

Quantitive way to calculate QFT observables

Acceptance goes down for lighter mass...(expected)

for m=0.5, roughly 80-60%

ALG	N_σ	N_τ	β	m	Acceptance	N_{trj}	$\langle P\rangle$	$\langle R\rangle$	$\langle L\rangle$
HMC	6	6	2.5	0.05	0.82	50000	0.67774(4)	0.49772(5)	0.437(6)
SLMC	6	6	2.5	0.05	0.34	50000	0.67813(8)	0.4982(1)	0.436(7)
HMC	6	6	2.5	0.10	0.73	50000	0.6771(4)	0.49666(5)	0.428(6)
SLMC	6	6	2.5	0.10	0.37	50000	0.67749(7)	0.49732(9)	0.438(5)

We have to use more expressible effective action!

neural net?

Issues to be solved

Extend to SU(3) -> straight forward
Nf = 2+1 -> straight forward
Determinant -> stochastic estimator (A. Hasenfratz's work)
Lighter mass -> using neural network effective action? (SLMC+NN)
Topological charge for large system
Systematic study of critical scaling

H. Shen, J. Liu and L. Fu, Phys. Rev. B 97, 205140 (2018)
YN, M. Okumura and A. Tanaka, Phys. Rev. B 101, 115111 (2020)
YN, M. Okumura, K. Kobayashi, and M. Shiga, Phys. Rev. B 102, 041124 (2020)

Can neural networks mimic log det(D[U]+m)?

Heatbath or MD with an action including neural networks

Summary

Summary

Machine learning is a tool to construct functions

machine for cats

→ → **cat!**

Self-learning Monte Carlo method

Configuration

→ → Boltzmann weight

Self-learning Monte Carlo method including fermions in LQCD is just started

A lot of things to do