Self-learning Monte Carlo method and all optical neural network

Junwei Liu (劉軍偉)

Department of Physics

Hong Kong University of Science and Technology

Deep learning and Physics, YITP, Kyoto, Japan, 2019

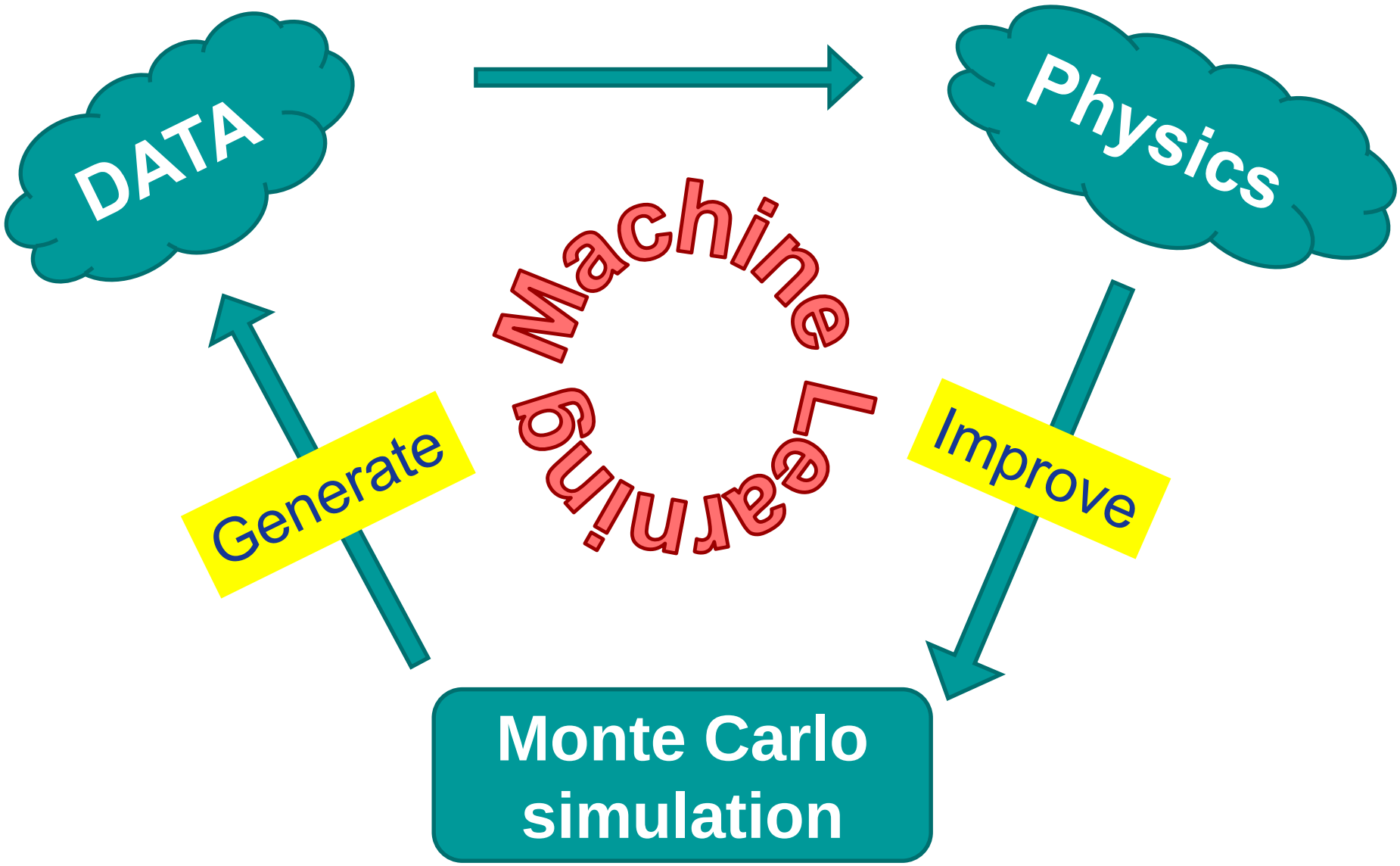
The general motivation

- As well known, there are great developments in machine learning techniques such as principle component analysis, deep neural networks, convolutional neural networks, generative neural networks, reinforcement learning and so on.

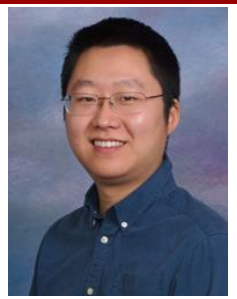
- By using these methods, there are also many great achievements such as image recognition, auto-pilot cars and Alpha-GO.

- Can we use these methods in physics to solve some problems? If so, what kind of problems can we solve and how?

Part 1: Self-learning Monte Carlo (ML→ Physics)



Collaborators and References


Liang Fu
(MIT)


Yang Qi
(Fudan)


Ziyang Meng
(IOP)


Xiaoyan Xu
(UCSD)


Yuki Nagai
(JAEA)


Huitao Shen
(MIT)

References

1. Self-Learning Monte Carlo Method, PRB 95, 041101(R) (2017)
2. Self-Learning Monte Carlo Method in Fermion Systems, PRB 95, 241104(R) (2017)
3. Self-Learning Quantum Monte Carlo Method in Interacting Fermion Systems, PRB 96, 041119(R) (2017)
4. Self-learning Monte Carlo method: Continuous-time algorithm, PRB 96, 161102 (2017)
5. Self-learning Monte Carlo with Deep Neural Networks, PRB 97, 205140 (2018)

Related work from Prof Lei Wang's group at IOP

1. Accelerated Monte Carlo simulations with restricted Boltzmann machines, PRB 95, 035105 (2017)
2. Recommender engine for continuous-time quantum Monte Carlo methods, PRE 95, 031301(R) (2017)

Monte Carlo methods

- Consider a statistical mechanics problem

$$Z = \sum_C e^{-\beta H[C]} = \sum_C W(C)$$

$$\langle O \rangle = \sum_C O[C] e^{-\beta H[C]} / Z = \sum_C O[C] W(C) / Z$$

- Pick up N configurations (samples) in the configuration space $\{C\}$ based on the importance $W(C_i)/Z$. Then we can estimate observables $\langle O \rangle$ as

$$\langle O \rangle \approx \sum_{i=1}^{N} O[C_i] \Big/ N$$

- The statistical error is proportional to $1/\sqrt{N}$. In high dimension (d>8), Monte Carlo is the most important and sometimes the only available method to perform the integral/summation.

Quantum Monte Carlo methods

- Consider a quantum system characterized by H

$$Z = \sum_{\psi} \langle \psi | e^{-\beta H} | \psi \rangle$$

Method 1: Trotter decomposition

$$Z = \sum_{\psi} \langle \psi | e^{-\beta H} | \psi \rangle$$

$$= \sum_{\psi \psi_1 \psi_2 \ldots \psi_M} \langle \psi_1 | e^{-\beta H/M} | \psi_2 \rangle \langle \psi_2 | e^{-\beta H/M} | \psi_3 \rangle \ldots \langle \psi_M | e^{-\beta H/M} | \psi_1 \rangle$$

Method 2: serial expansion

$$Z = \sum_{\psi} \langle \psi | e^{-\beta H} | \psi \rangle = \sum_{\psi} \langle \psi | \sum_n (-\beta H)^n / n! \, | \psi \rangle$$

- Map the *N*-dimensional quantum model to be a (*N+1*)-dimensional “classical” model, and then use Monte Carlo method to simulate this “classical” model.

Markov chain Monte Carlo (MCMC)

- MCMC is a way to do important sampling based on the distribution $W(C)$. Configurations are generated one by one by the following approach:

1. first propose the next trial configuration C_t
2. If $Rand() \leq \boldsymbol{T(C_i \rightarrow C_t)}$, then the next configuration $C_{i+1} = C_t$ Otherwise, $C_{i+1} = C_i$
3. Repeating step 1 and step 2

$$\cdots \rightarrow C_{i-1} \rightarrow C_i \rightarrow C_{i+1} \rightarrow \cdots$$

- It is clearly that the next step only depends on the last step and **transition matrix** T, and we can demonstrate that the **detailed balance principle** guarantees that the Markov process will converge to the desired distribution

$$\frac{T(C \rightarrow D)}{T(D \rightarrow C)} = \frac{W(D)}{W(C)}$$

N. Metropolis, J. Chem. Phys. 21, 1087 (1953)

Metropolis-Hastings algorithm

- In Metropolis Algorithm, transition matrix is chosen as

$$T(C \to D) = \min\left\{1, \frac{W(D)}{W(C)}\right\}$$

N. Metropolis, et. al J. Chem. Phys. 21, 1087 (1953)

- The transition probability can be further written as

$$T(C \to D) = S(C \to D)p(C \to D)$$

$S(C \to D)$:proposal probability of conf. C from conf. D

$p(C \to D)$:acceptance probability of configuration C

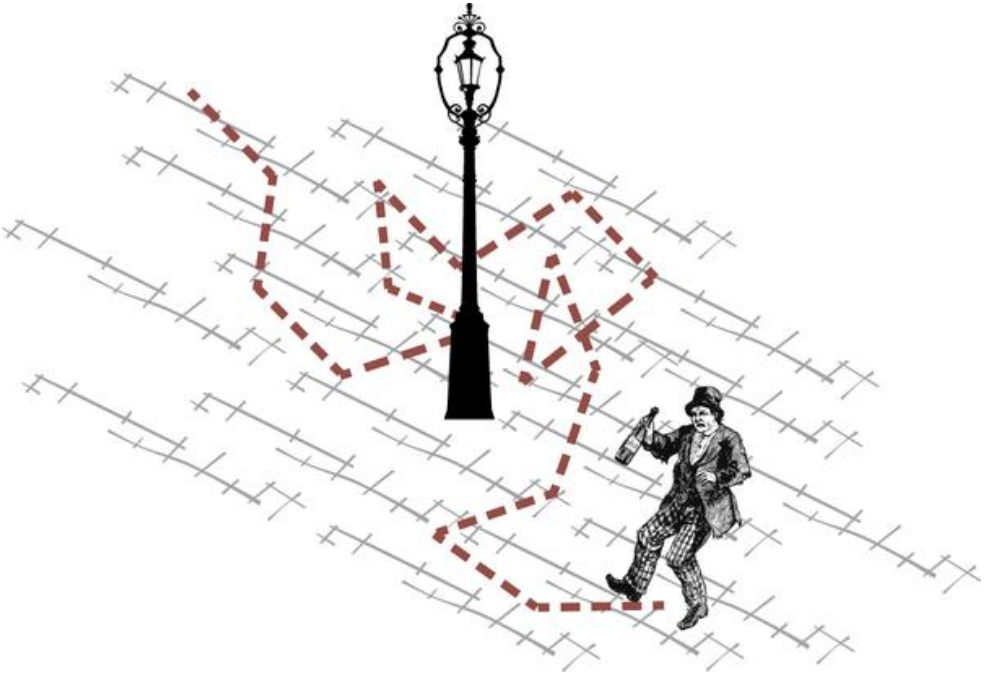
- In Metropolis-Hastings Algorithm, the acceptance ratio is

$$p(C \to D) = \min\left\{1, \frac{W(D)S(D \to C)}{W(C)S(C \to D)}\right\}$$

W. H. Hastings, Biometrika 57, 97 (1970)

Independent Samples

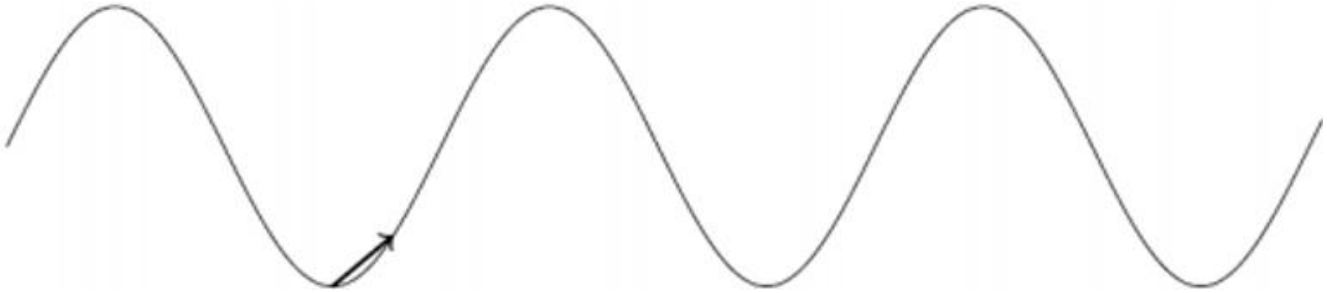
- As well known, for the statistic measurements, only the ***independent*** samples matter.



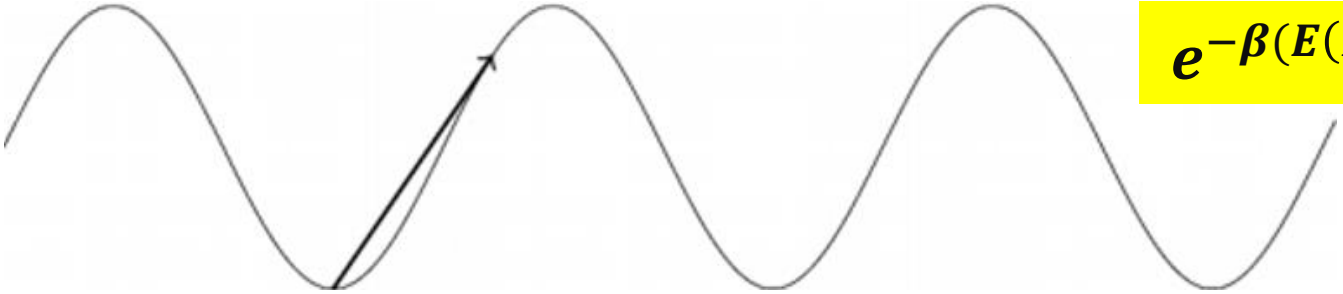
- However, since the configurations are generated sequentially one by one as random walk in configuration space, it is inevitable that the configurations in the Markov chain are correlated with each other and not independent.
- The configurations generated by different Monte Carlo methods have different autocorrelations.

Different update algorithms

- Too small step length: small difference, high acceptance.

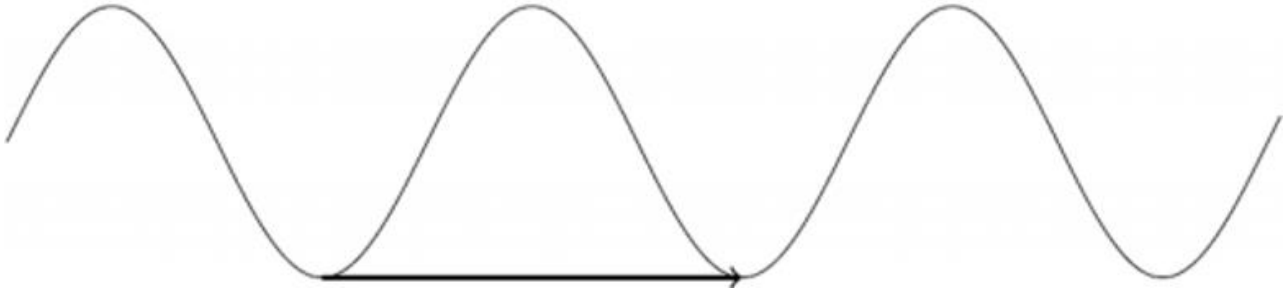


- Too large step length: big difference, low acceptance.



$$e^{-\beta(E(D)-E(C))}$$

- Ideal step length: big difference and high acceptance, exploring the low-energy configurations.



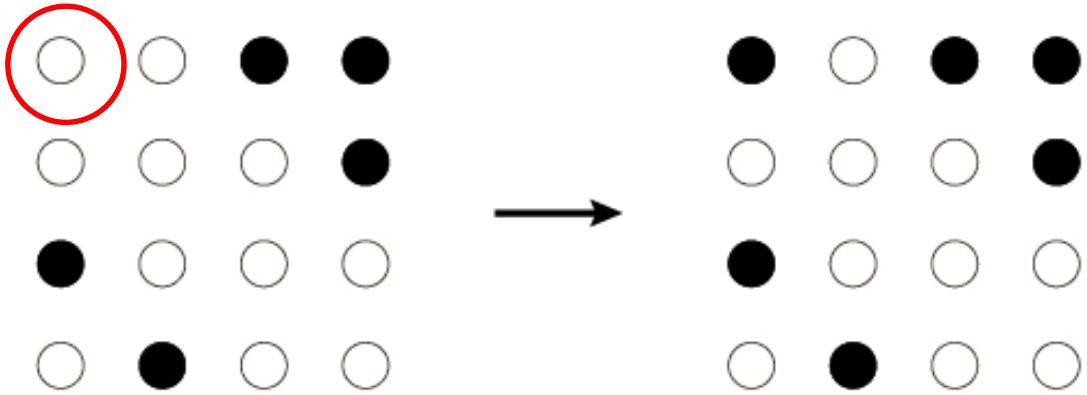
How to justify different Monte Carlo methods?

Time consumption tl_c to get two ***statistically independent*** configurations

- **Auto-correlation time** l_c: the number of steps to get independent configurations
  - ✓ Bigger differences → independent, but low acceptance ratio
  - ✓ Similar energies → high acceptance ratio, but not independent
- Time consumption t to get one configuration (mainly for the calculation of weight)

Self-learning Monte Carlo methods are designed to improve **both** parts, thus can speed up the calculations dramatically.

Local update



- Local Update $S(C \to D) = S(D \to C) = \frac{1}{N}$

- Acceptance ratio

$$\alpha\,(C \to D) = \min\left\{1, \frac{W(D)S(D \to C)}{W(C)S(C \to D)}\right\} = \min\left\{1, e^{-\beta(E(D)-E(C))}\right\}$$

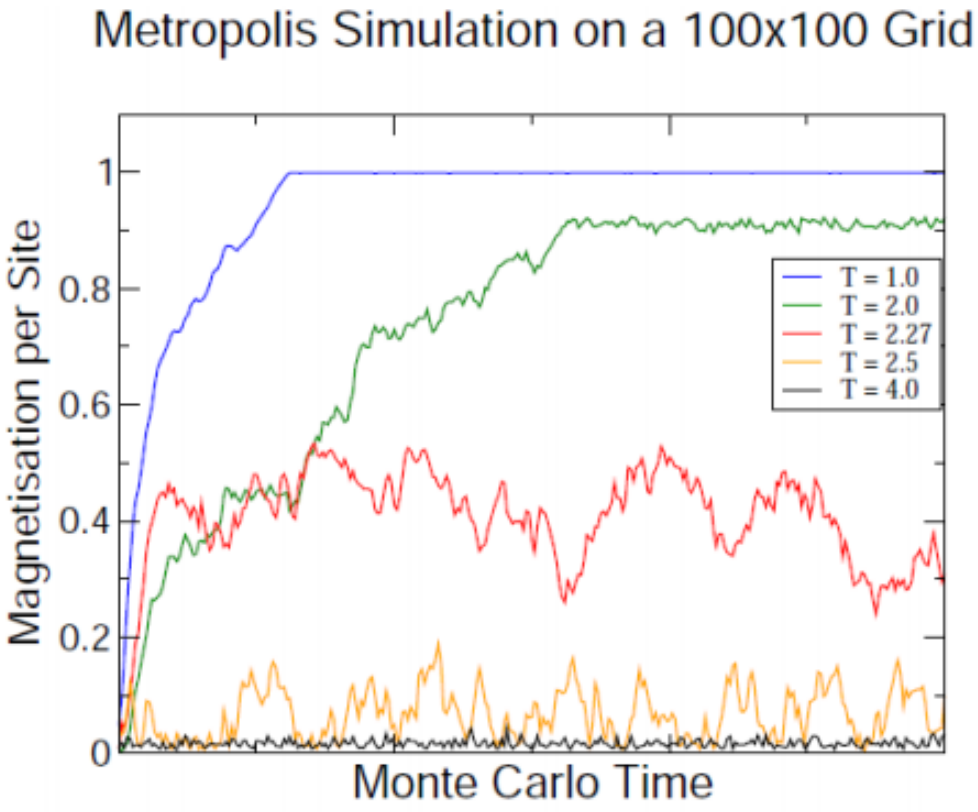
- Very general: applies to any model

N. Metropolis, et al J. Chem. Phys. 21, 1087 (1953)

Critical slowing down

- Dynamical relaxation time diverges at the critical point: convergence is very slow in the critical system.

- For 2D Ising model, autocorrelation time $\tau \propto L^z, z = 2.125$

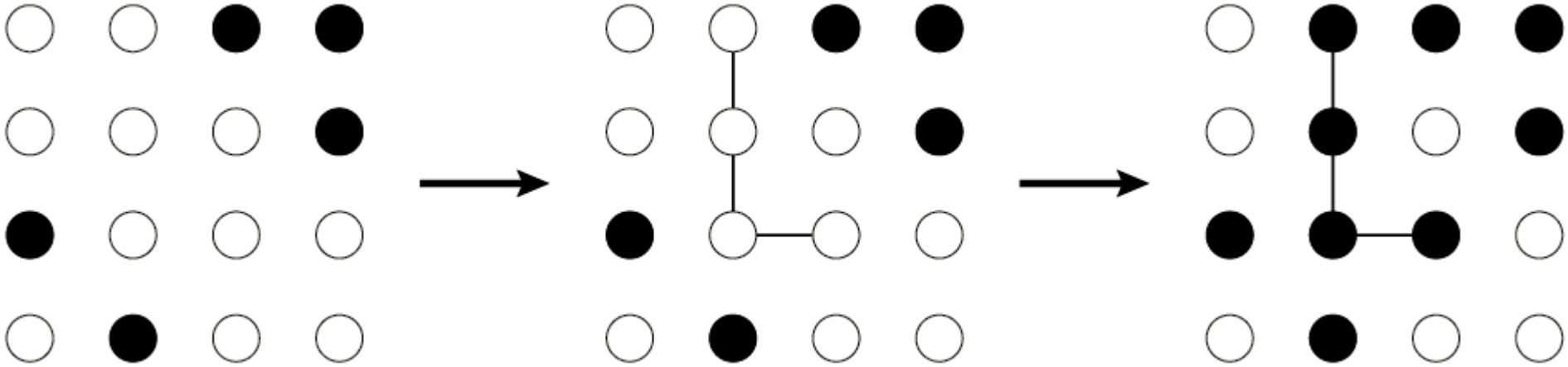


How to get high acceptance ratio?

$$\alpha\,(C \to D) = \min\left\{1, \frac{W(D)S(D \to C)}{W(C)S(C \to D)}\right\}$$

$$\frac{W(D)}{W(C)} = \frac{S(C \to D)}{S(D \to C)}$$

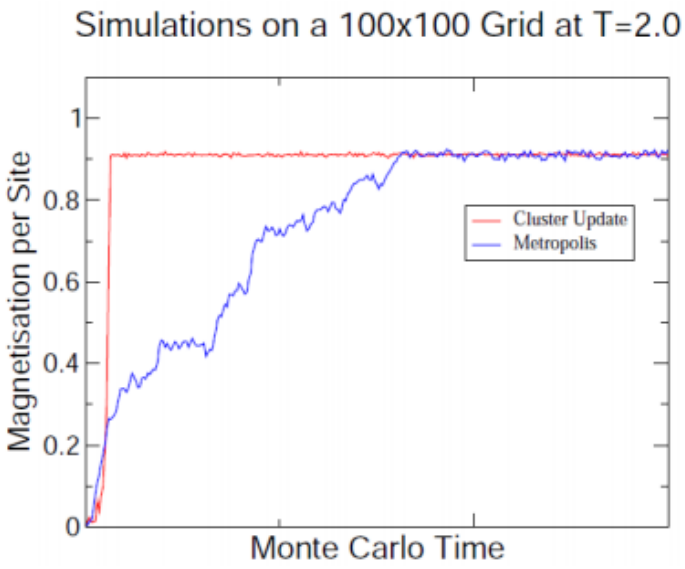
Global update -- Wolff algorithm in Ising model

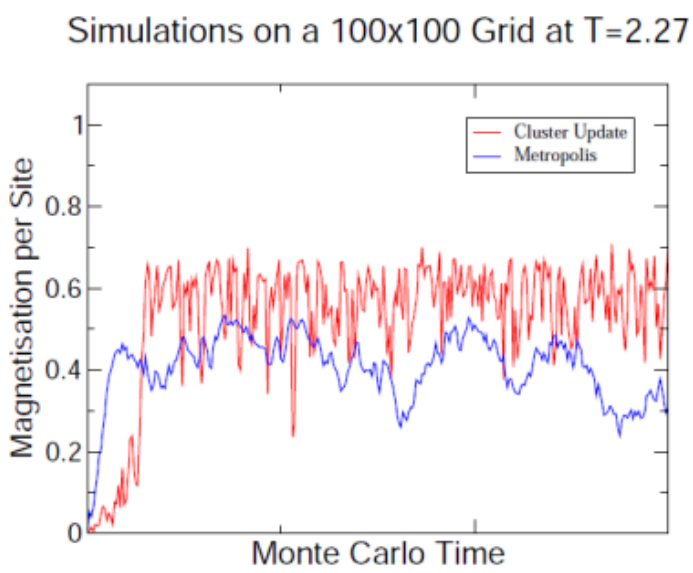


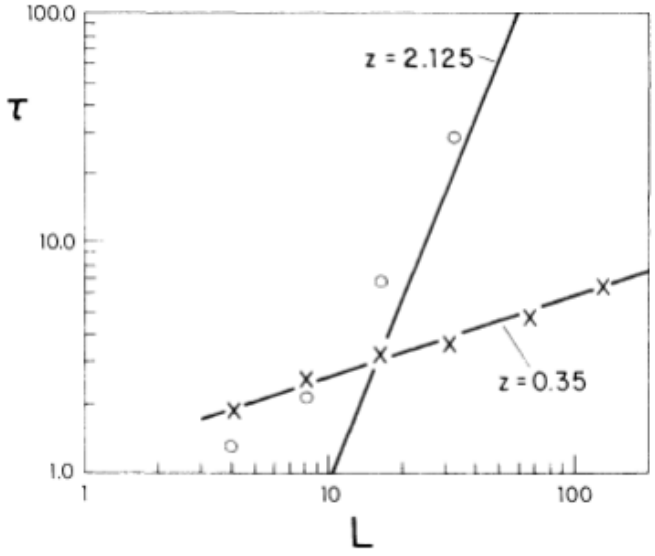
1. Randomly choose one site i
2. If the adjacent sites have the same status, then add them in the cluster C with the probability $S(i \to j) = 1 - e^{-2\beta J}$
3. Repeat step 2 for all the sites in the cluster C
4. Change the status of all the sites in the cluster C

Swendsen and Wang, Phys. Rev. Lett. 58, 86 (1987)
U. Wolff, Phys. Rev. Lett. 62, 361 (1989)

Reduce critical slowing down

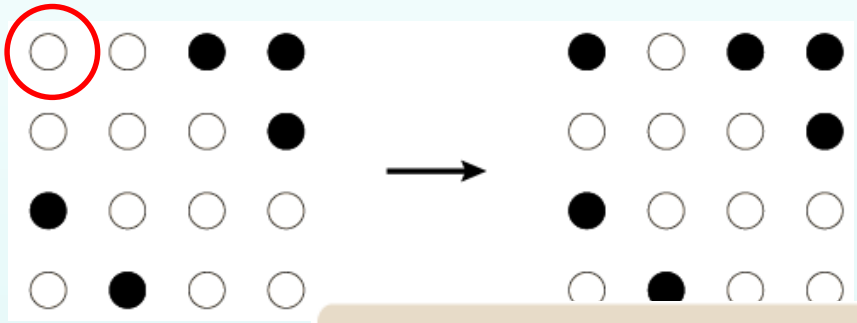


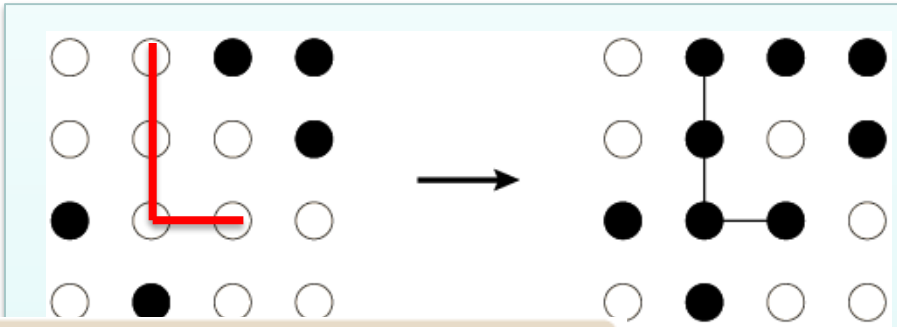




Swendsen and Wang, Phys. Rev. Lett. 58, 86 (1987)

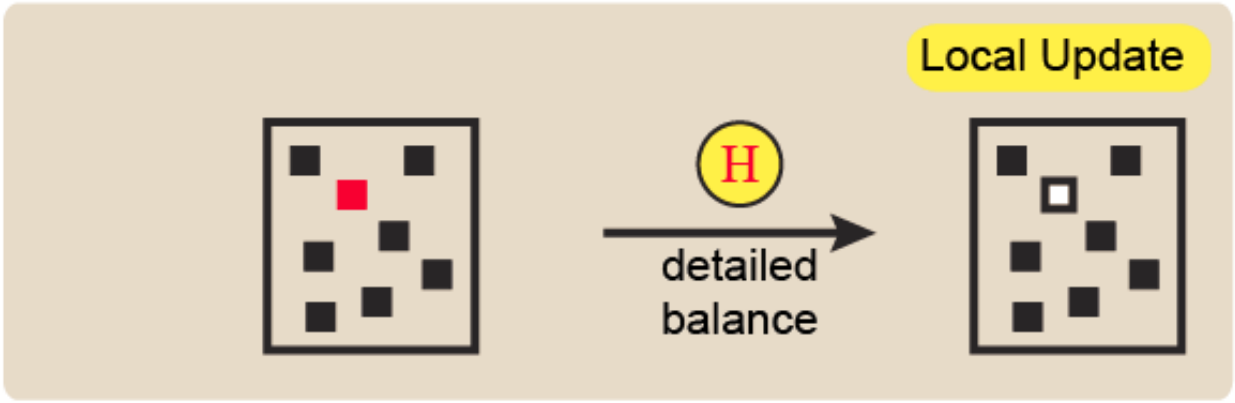
Local update and global update

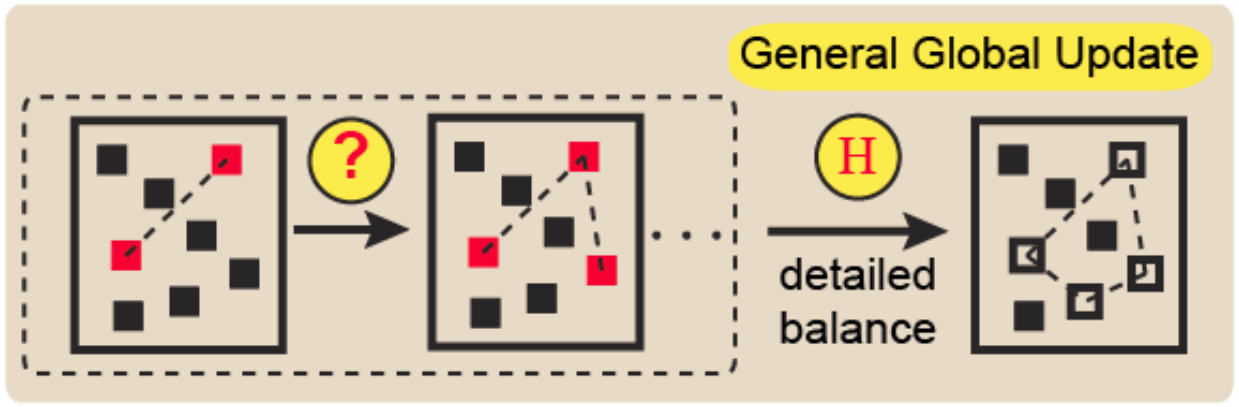




Local updat

- Locally u … nfiguration by chang … changing step … step
- Very gen …
- Inefficien … fic transition … be slowing … r models





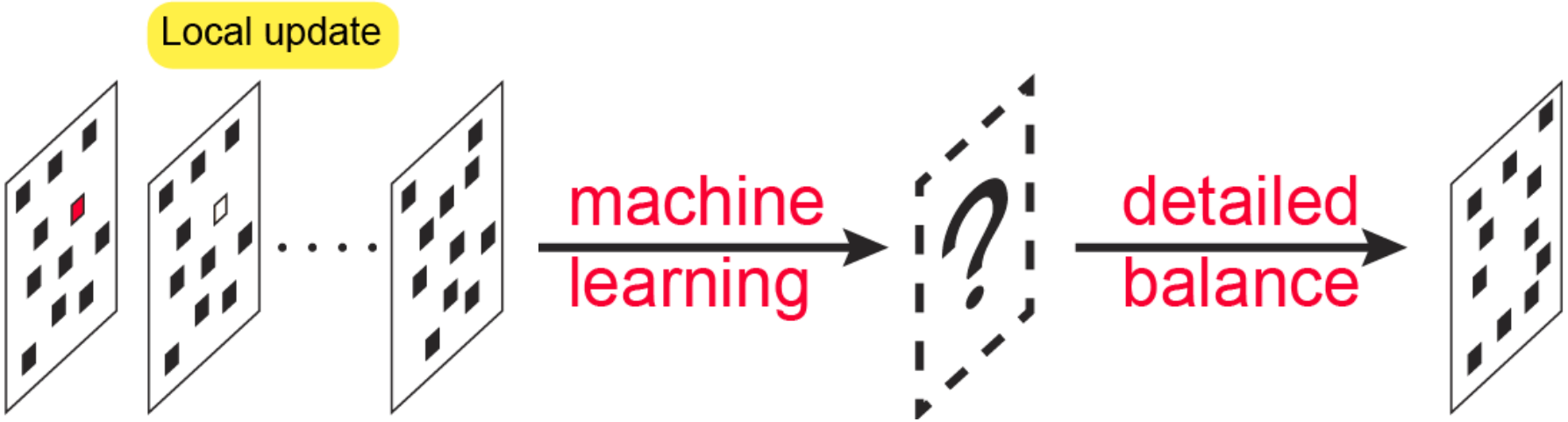
How to get high acceptance ratio?

$$\alpha\,(C \to D) = \min\left\{1, \frac{W(D)S(D \to C)}{W(C)S(C \to D)}\right\}$$

$$\frac{W(D)}{W(C)} = \frac{S(C \to D)}{S(D \to C)}$$

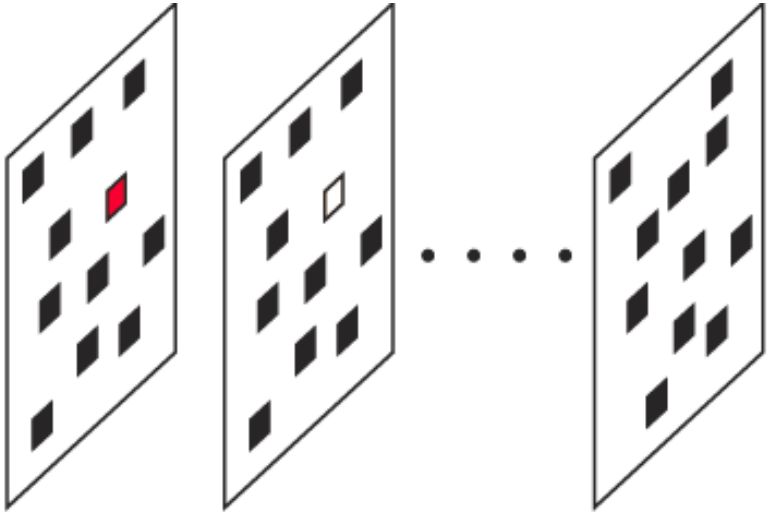
$$\frac{W(D)}{W(C)} \approx \frac{S(C \to D)}{S(D \to C)}$$

My initial naïve idea



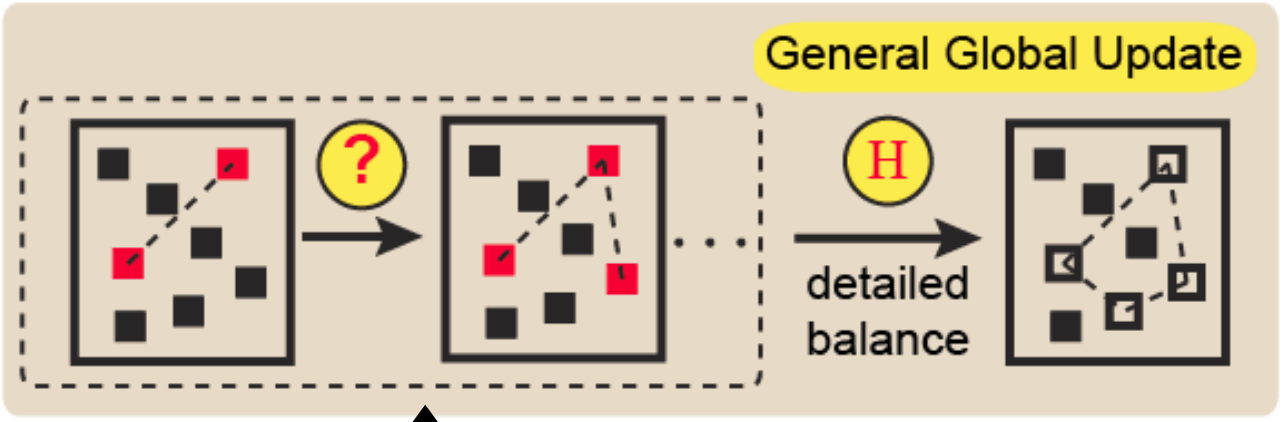
- Use machine learning to learn some **common features** for these “important” configurations, and then generate new configurations based on these learned features.
- It seems to be good, but it does not work, because we don’t know $\frac{S(C\to D)}{S(D\to C)}$, and we cannot calculate the right acceptance probability.
- However, this idea tells us that there are **other important information** in the generated configurations, besides $\langle O\rangle \approx \sum_{i=1}^{N} O[C_i]/N$ based on them.

Hidden information in generated configurations



- The generated configurations have a similar distribution close to the original distribution !!!
- It is too obvious and seems to be too trivial, but we do not use this seriously besides calculating the average of operators.
- Can we further use it and how?
- The answer is YES and we can do it in self-learning Monte Carlo method.

The right way to use the hidden information in generated configurations



Yang Qi

(Fudan)

Critical properties

Learn

Guide

Universality class

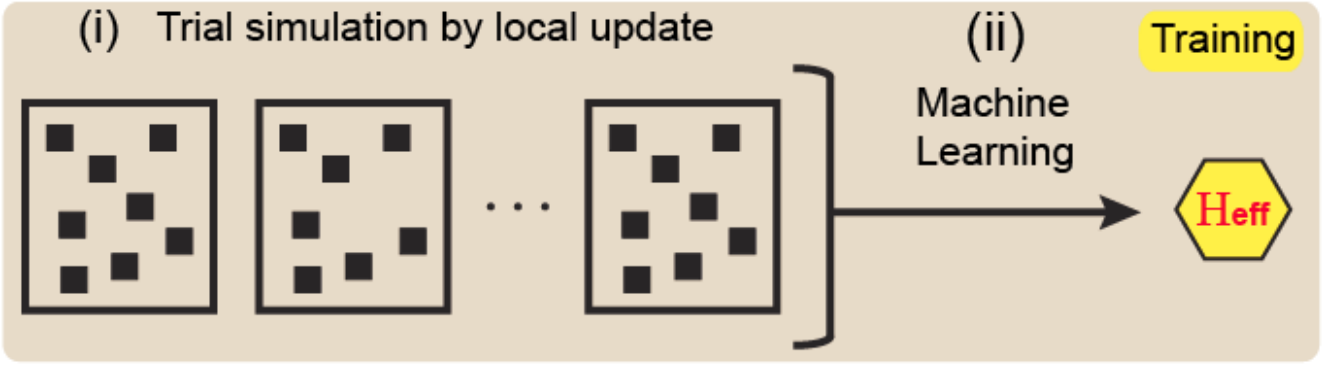
Effective Model

Core ideas of self-learning Monte Carlo

- Learn an approximated ***simpler*** model
  - ✓ Having efficient global update methods
  - ✓ Evaluating the weights faster

First learn

(a)

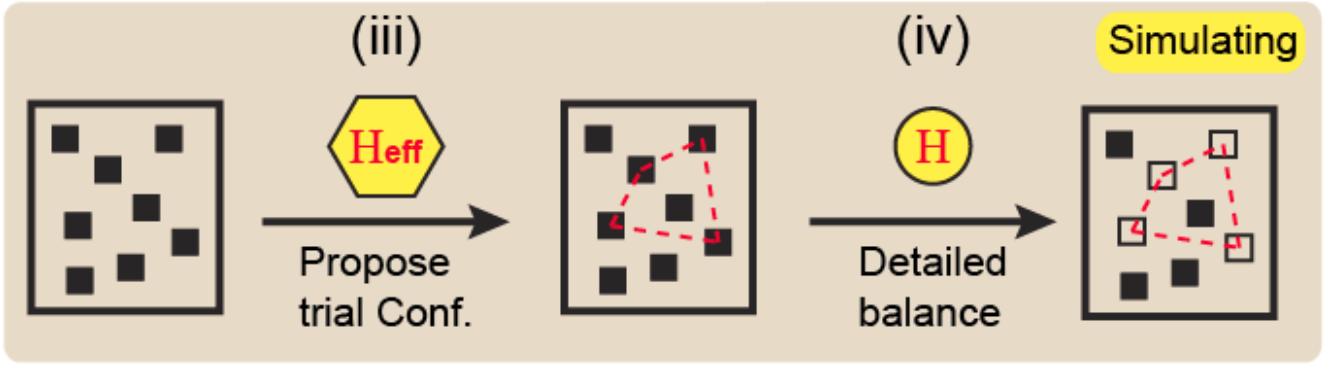


How to train?

- Use the **simpler model** to guide us to simulate the original hard model

Then earn

(b)



How to propose?

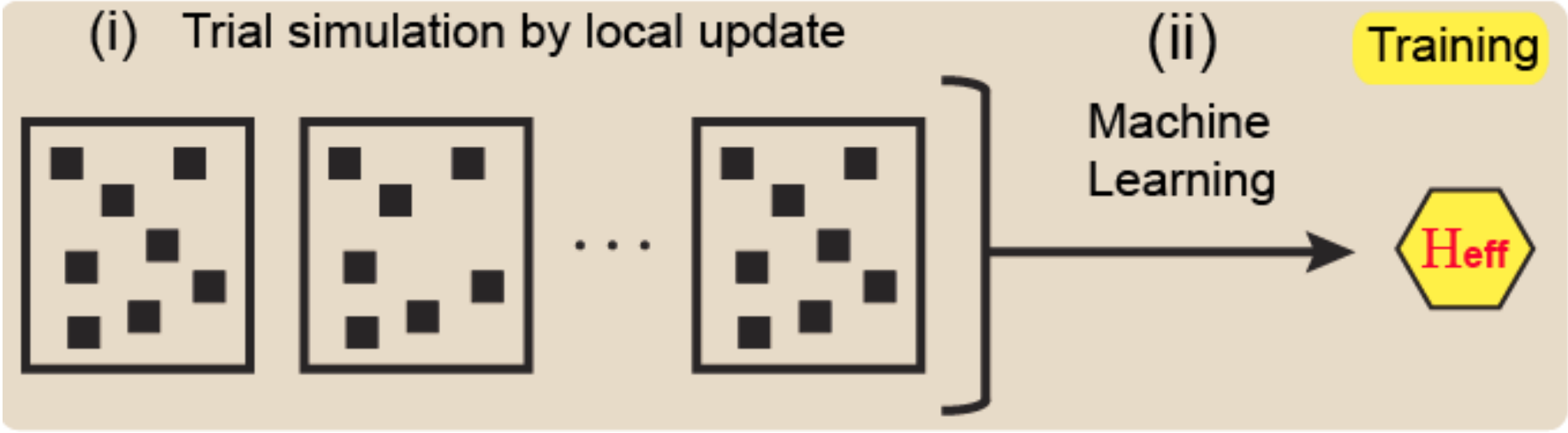
J. Liu, et al PRB 95, 041101(R) (2017)

SLMC in Boson system

- Original Hamiltonian has both two-body and four-body interactions, and we do not have global update method.

$$H = -J \sum_{\langle ij \rangle} S_i S_j - K \sum_{ijkl \in \square} S_i S_j S_k S_l \qquad K/J = 0.2$$

Ising transition with $T_c = 2.493$

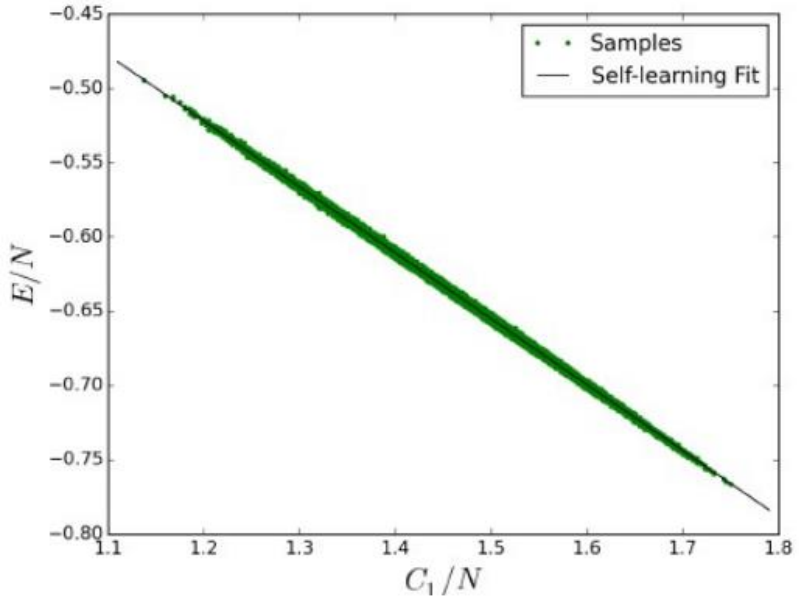


Fit the parameters in H_{eff}

- Effective Hamiltonian

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

- Generate configurations with local update at T=5>Tc, away from the critical points
- Perform linear regression to fit the parameters in H_{eff}

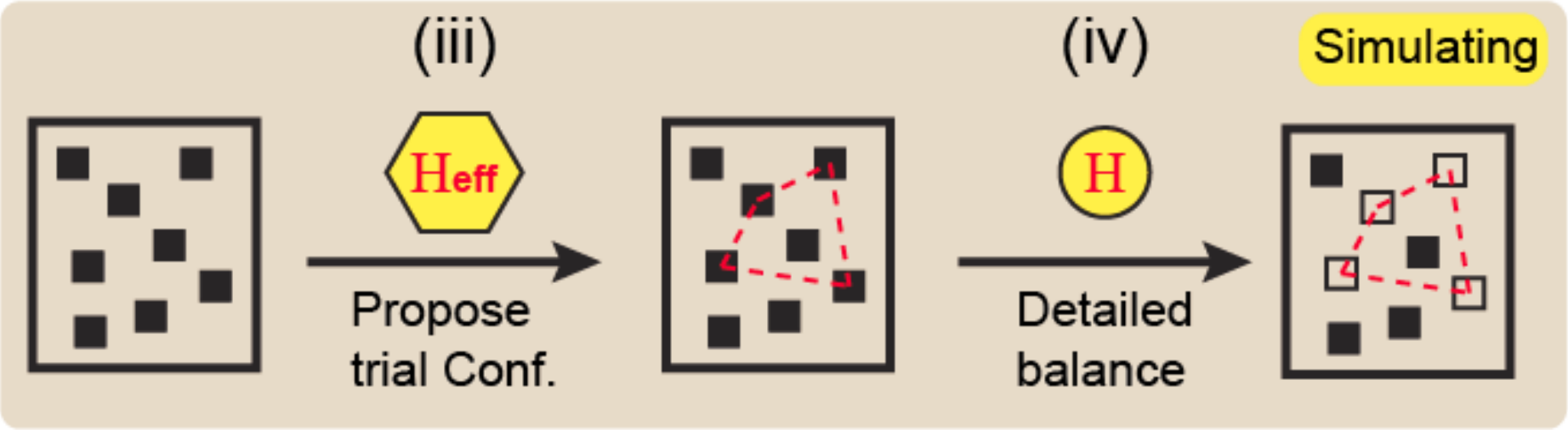


$$C_1 = \sum_{\langle i,j \rangle} S_i S_j$$

	J_1	J_2	J_3
Train 1	1.2444	-0.0873	-0.0120
Train 2	1.1064	-	-

- Generate configurations with reinforced learning at Tc

Build the cluster based on H_{eff}



- Self-learning update: cluster is constructed using Wolff update obeying detailed balance principle of effective model $\boldsymbol{H}_{\text{eff}}$

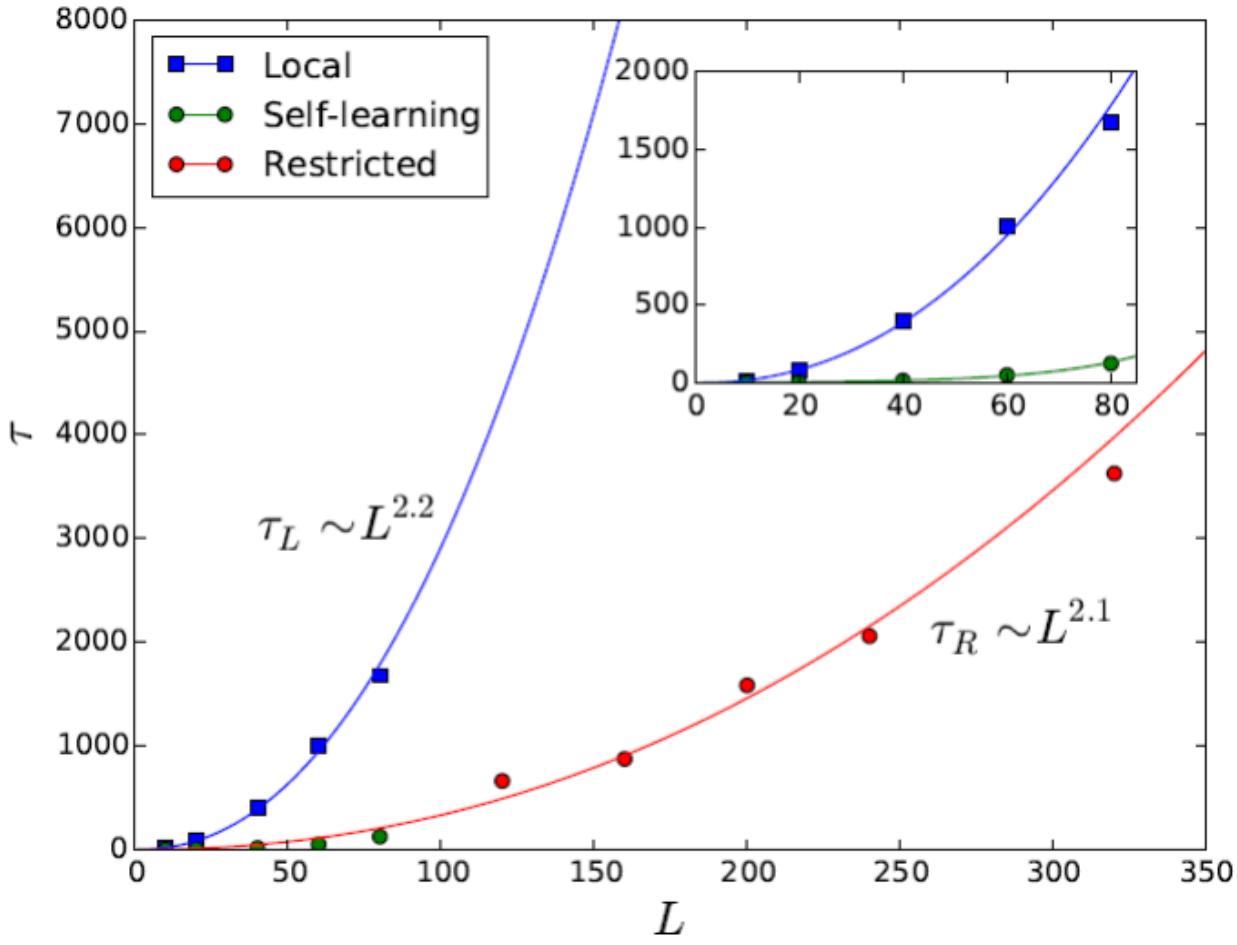
$$\frac{S(D \to C)}{S(C \to D)} = \frac{W_{\text{eff}}(C)}{W_{\text{eff}}(D)}$$

- Acceptance ratio:

$$\alpha(C \to D) = \min\left\{1, \frac{W(D)S(D \to C)}{W(C)S(C \to D)}\right\} = \min\left\{1, \frac{W(D)W_{\text{eff}}(C)}{W(C)W_{\text{eff}}(D)}\right\}$$

$$= \min\left\{1, e^{-\beta\left[\left(E(D)-E_{\text{eff}}(D)\right)-\left(E(C)-E_{\text{eff}}(C)\right)\right]}\right\}$$

For different sizes $L \times L$



- SLMC is 20~40 fold faster than the local update.
- We can easily get the result for 320 × 320.

J. Liu, et al PRB 95, 041101(R) (2017)

How to justify different Monte Carlo methods?

Time consumption tl_c to get two ***statistically independent*** configurations

- **Auto-correlation time** l_c: the number of steps to get independent configurations
  - ✓ Bigger differences → independent, but low acceptance ratio
  - ✓ Similar energies → high acceptance ratio, but not independent
- Time consumption t to get one configuration (mainly for the calculation of weight)

Self-learning Monte Carlo methods are designed to improve **either or both** parts depending on models, thus can speed up the calculations dramatically.

Weight (Free energy) in Fermion systems

- In general fermion systems, the partition function can be calculated based on the following form

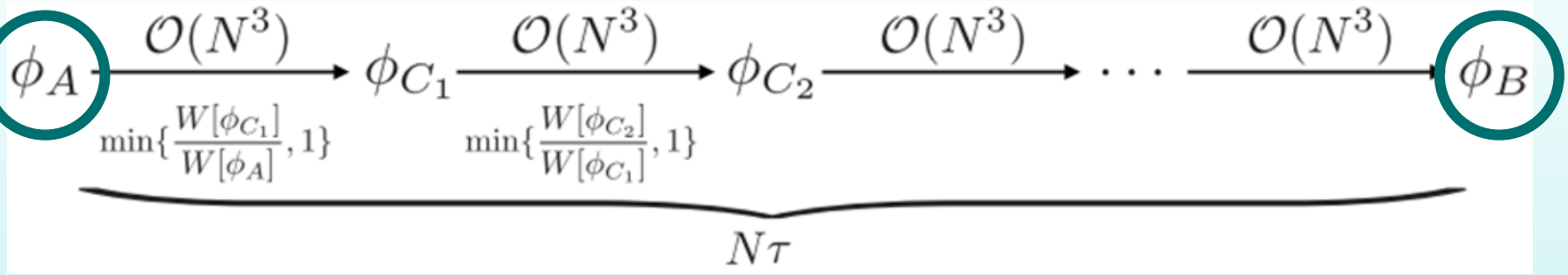
$$Z = \sum_{\phi(\tau)} \det \left[\mathbf{I} + \prod_{\tau} e^{-\Delta\tau H_f[\phi(\tau)]} \right] \equiv \sum_{\phi(\tau)} W[\phi(\tau)]$$

- We have to deal with heavy matrix operations to calculate the weight (free energy), which is very time-consuming $O(N^3)$.

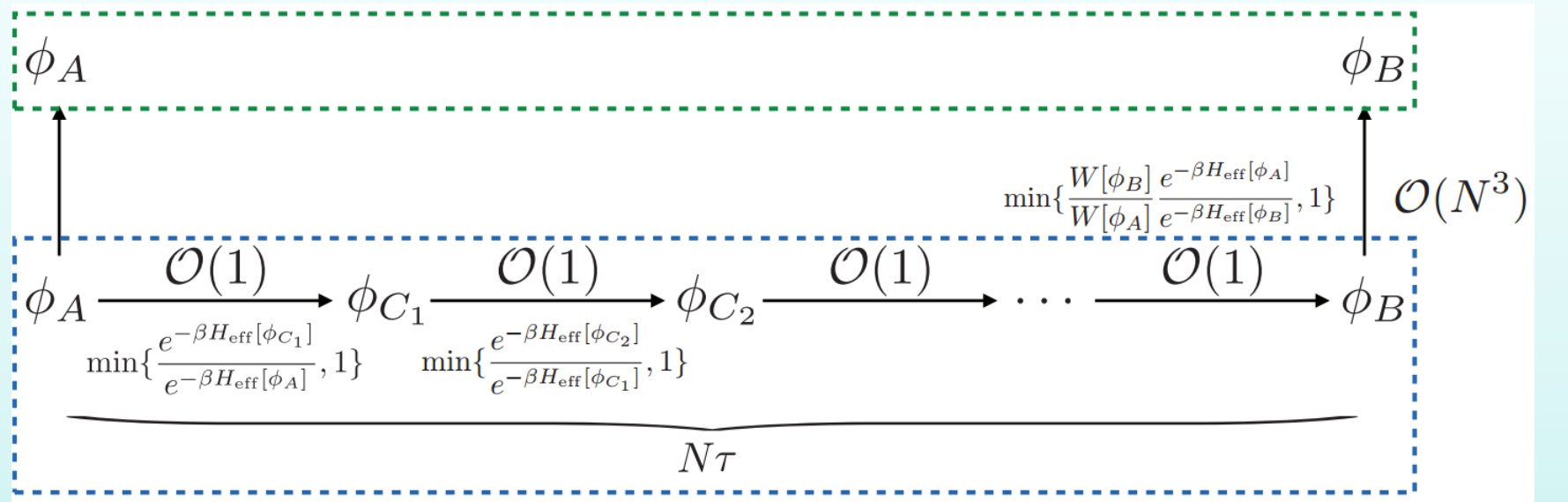
- However, the weight ***only*** depends on the configuration of bosonic field, which can be described by **pure Boson models**.

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

Complexity of SLMC



- Complexity to obtain two uncorrelated configurations $N\tau \times N^3 = \tau N^4$



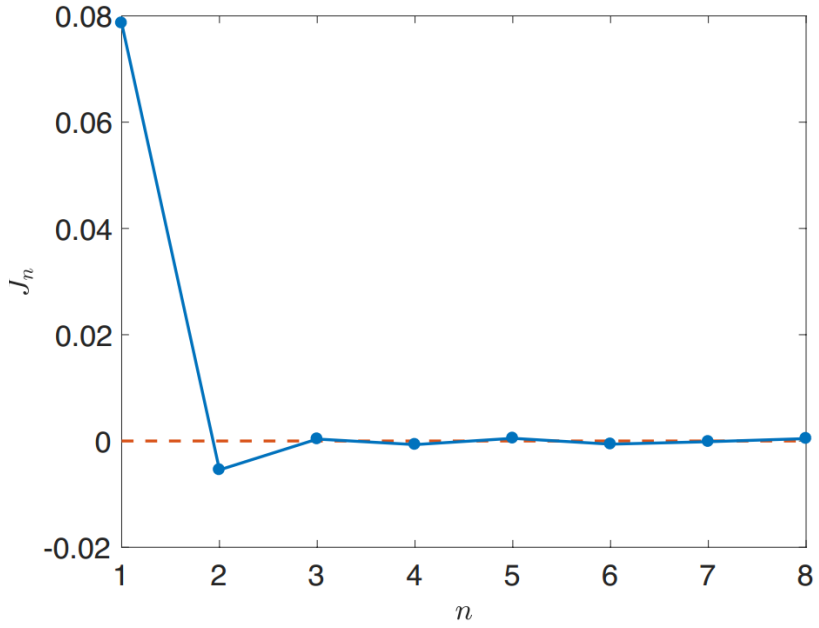
- Complexity to obtain two uncorrelated configurations $\left(N\tau + N^3\right)/p$

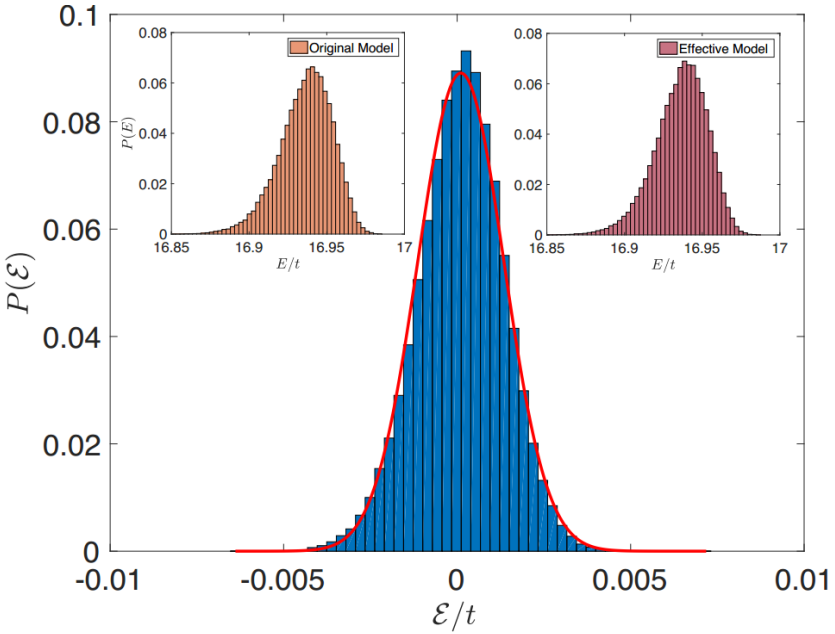
Example: double exchange model

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

- Magnetic order in metal
- Free Fermions are coupled to classical Heisenberg spins
- Weight $\prod_n (1 + e^{-\beta E_n(\phi)})$
- The E_n are obtained by ED of H for given configurations ϕ
- Computational complexity in conventional method: $O(\tau L^d L^{3d})$

Learning the effective model





- Effective model

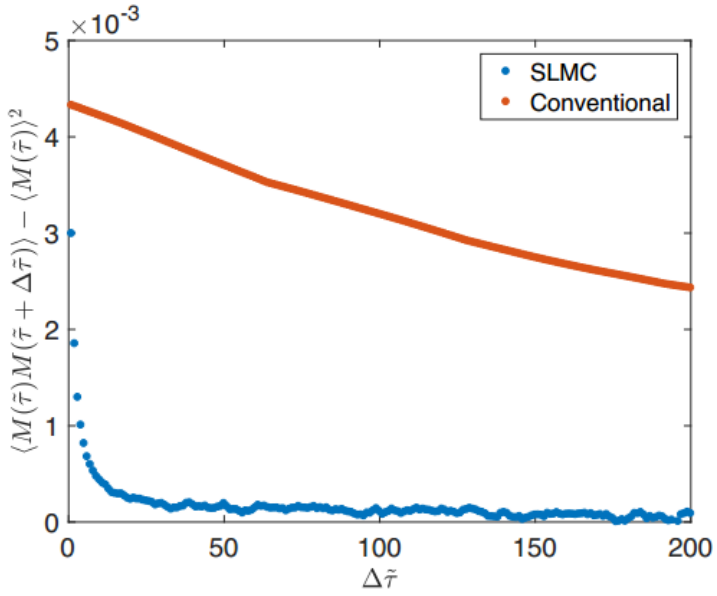
$$H_{\text{eff}} = E_0 - J_1 \sum_{\langle ij \rangle_1} \vec{S}_i \cdot \vec{S}_j - J_2 \sum_{\langle ij \rangle_2} \vec{S}_i \cdot \vec{S}_j - \cdots$$

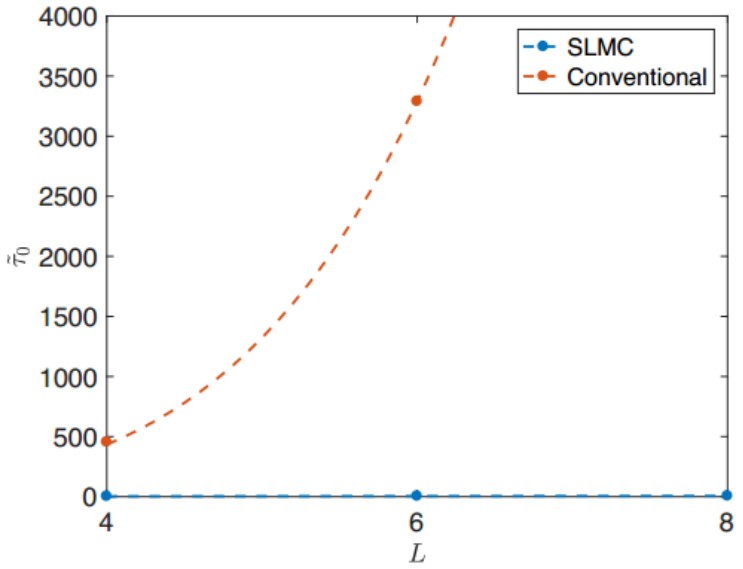
- Reproduce the RKKY feature

- Reproduce the right Boltzmann distribution

J. Liu, et al PRB 95, 241104(R) (2017)

No critical slowing down





- Autocorrelation time is much shorter in SLMC than conventional method
- There is no critical slowing down in SLMC

- Computational complexity in conventional method: $O(\tau L^d L^{3d})$
- Computational complexity in SLMC: $O\left(l_c \approx \tau L^d\right) + O(L^{3d})$
- SLMC reduces the complexity by $O(\tau L^d)$, and can be more than 1000 times faster

J. Liu, et al PRB 95, 241104(R) (2017)

Example II: SLMC in DQMC (Interacting Fermion)

- In conventional DQMC, the computational complexity in conventional method: $O(\tau\beta N^3)$

Ziyang Meng

(IOP)

- In SLMC with cumulative update, the complexity is $O(\gamma\beta N l_c + N^3 + \beta N^2)$
  a. Cumulative update $O(\gamma\beta N l_c)$
  b. Detail balance $O(N^3)$
  c. Sweep Green's function $O(\beta N^2)$

Xiaoyan Xu

(UCSD)

X. Y. Xu, et al PRB 96, 041119(R) (2017)

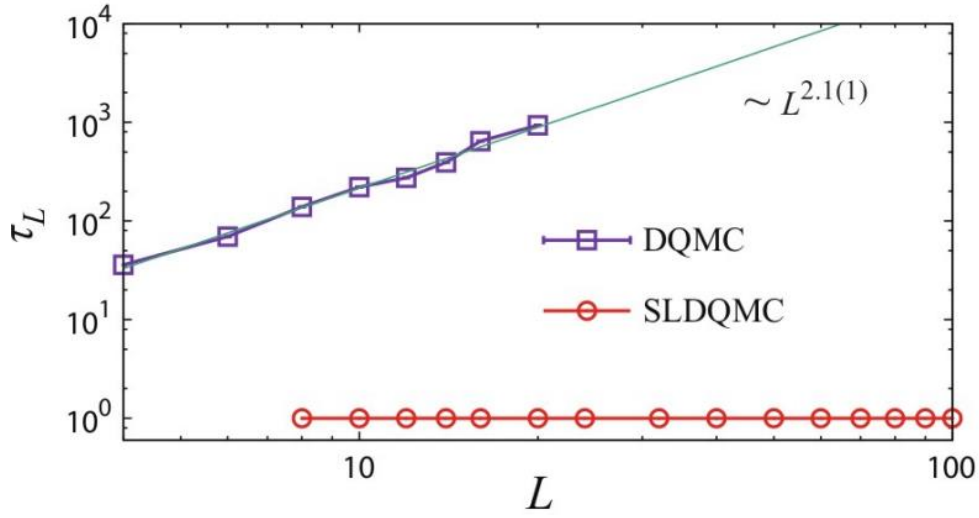
Results for a particular model

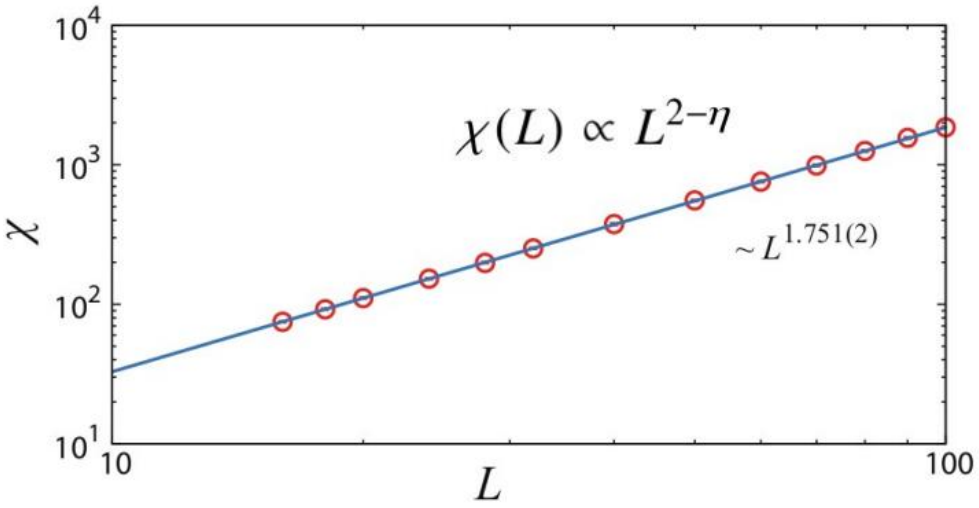
$$H = H_f + H_{sf} + H_s,$$

$$H_f = -t \sum_{\langle ij\rangle\lambda\sigma} c^{\dagger}_{i\lambda\sigma} c_{j\lambda\sigma} + h.c. - \mu \sum_{i\lambda\sigma} n_{i\lambda\sigma}$$

$$H_{sf} = -\xi \sum_i s^z_i (\sigma^z_{i1} - \sigma^z_{i2}),$$

$$H_s = -J \sum_{\langle ij\rangle} s^z_i s^z_j - h \sum_i s^x_i,$$





For the first time, we can achieve large system size as **100×100** in two dimension

X. Y. Xu, et al PRB 96, 041119(R) (2017)

Example III: SLMC in CT-AUX

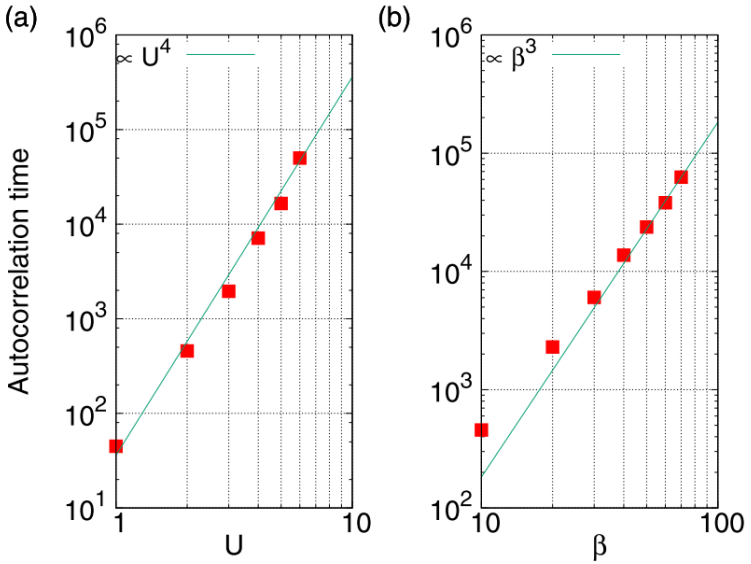
Single impurity Anderson model

$$H = H_0 + H_1,$$

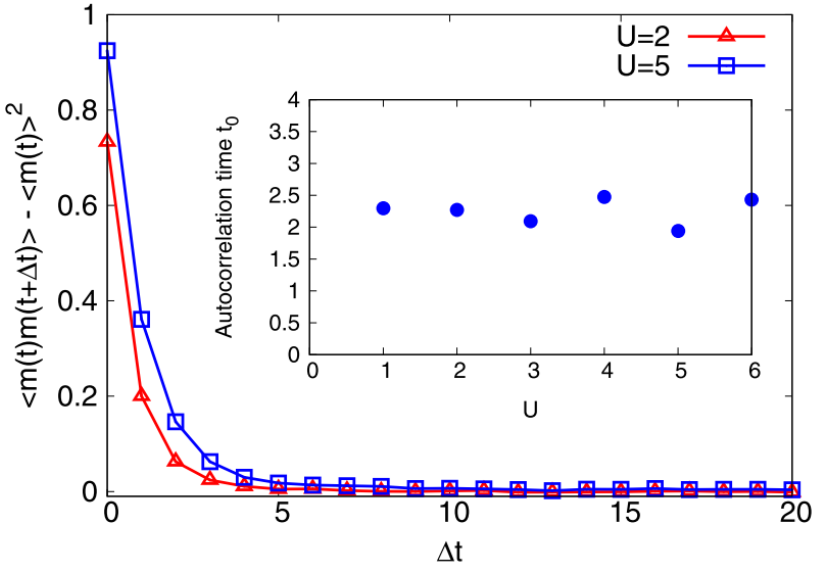
$$H_0 = -(\mu - U/2)(n_\uparrow + n_\downarrow) + \sum_{\sigma,p}(V c_\sigma^\dagger a_{p,\sigma} + \text{H.c.})$$

$$+ \sum_{\sigma,p} \epsilon_p a_{p,\sigma}^\dagger a_{p,\sigma} + K/\beta,$$

$$H_1 = U(n_\uparrow n_\downarrow - (n_\uparrow + n_\downarrow)/2) - K/\beta,$$



Yuki Nagai
(JAEA)

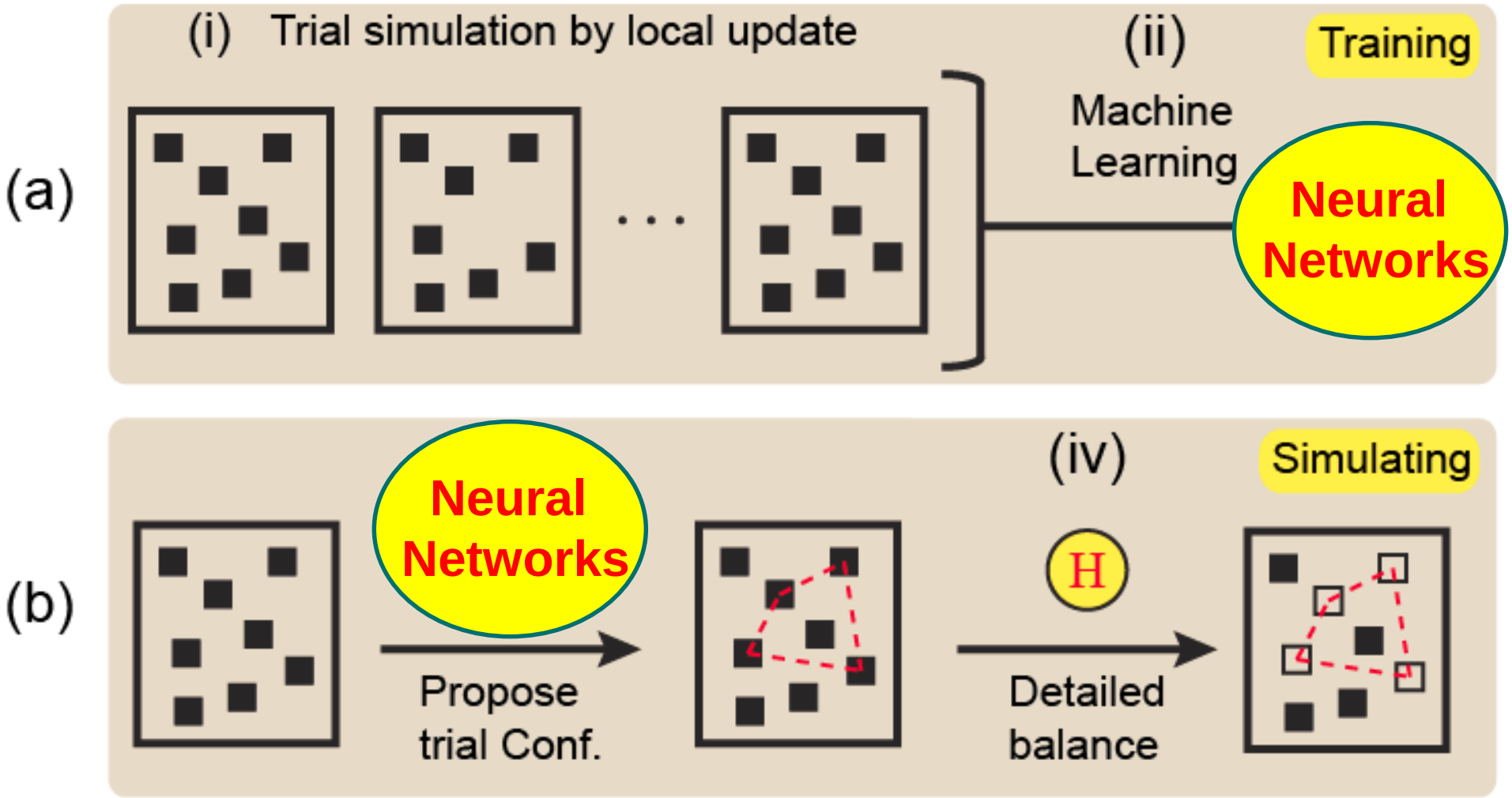


For the single impurity Anderson model in low temperature or large interaction, one can achieve speedup

$$O\left(\frac{\langle n \rangle}{m_c \tau_{SL}}\right) \approx O(\beta U)$$

Yuki Nagai, et al PRB 96, 161102 (2017)

SLMC with Deep Neural Networks



H. Shen, J. Liu, L. Fu, PRB 97, 205140 (2018)

Asymmetric Anderson model with a single impurity

$$\hat{H} = \hat{H}_0 + \hat{H}_1,$$

$$\hat{H}_0 = \sum_{\mathbf{k}} \varepsilon_{\mathbf{k}} \hat{c}^{\dagger}_{\mathbf{k}} \hat{c}_{\mathbf{k}} + V \sum_{\mathbf{k}\sigma} (\hat{c}^{\dagger}_{\mathbf{k}} \hat{d}_{\sigma} + \text{h.c.}) + \mu \hat{n}_d,$$

$$\hat{H}_1 = U \left(\hat{n}_{d,\uparrow} - \frac{1}{2} \right) \left(\hat{n}_{d,\downarrow} - \frac{1}{2} \right),$$

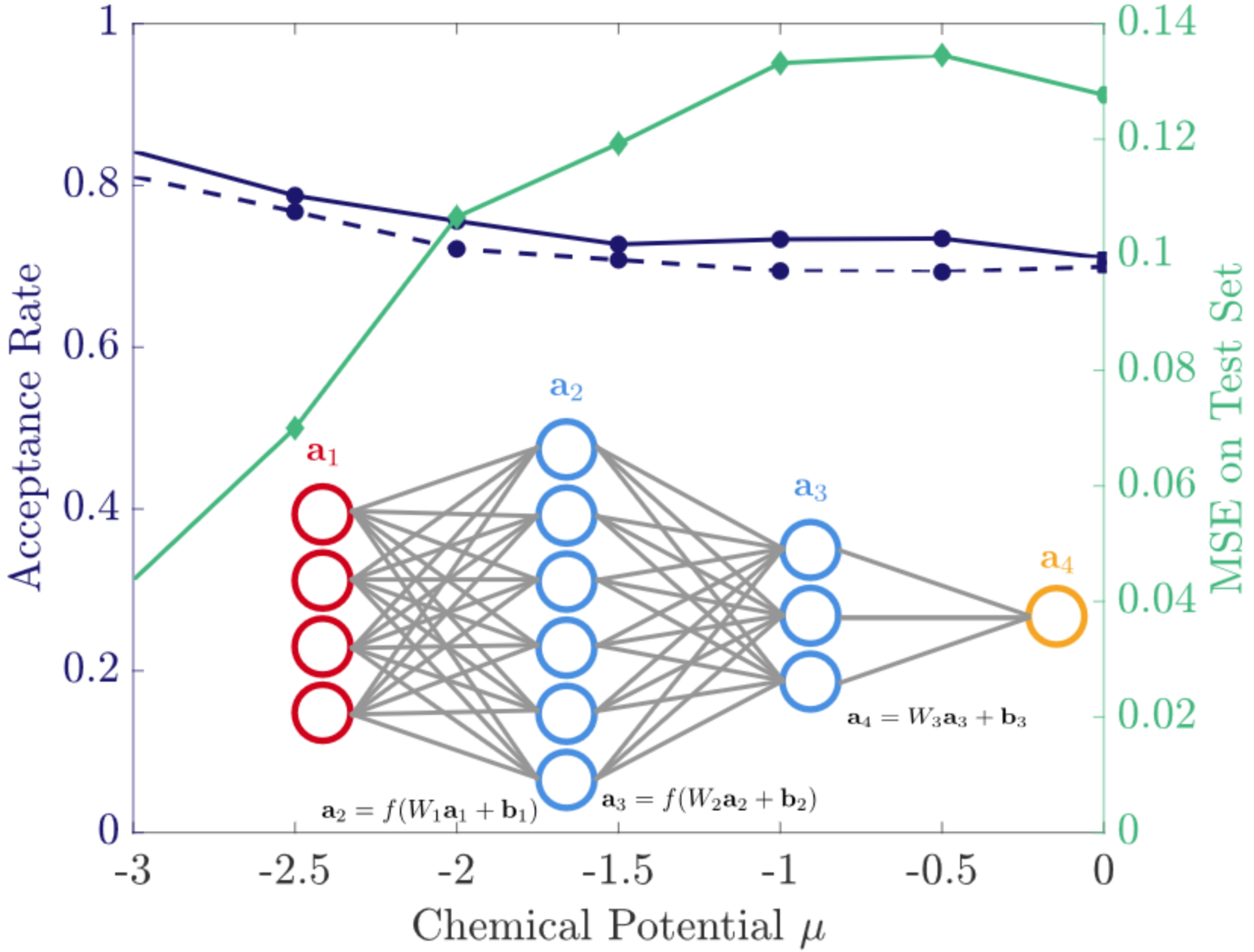
- $\hat{d}_{\sigma}$ and $\hat{c}_k$ are the fermion annihilation operator for the impurity and for the conduction electrons, and $\hat{n}_{d,\sigma} \equiv \hat{d}^{+}_{\sigma} \hat{d}_{\sigma}$, $\hat{n}_d = \hat{n}_{d\uparrow} + \hat{n}_{d\downarrow}$

- By Hubbard-Stratonovich transformation, we can decouple the interaction parts as an auxiliary bosonic field coupled to free fermions $e^{-\Delta\tau \hat{H}_1} = \frac{1}{2} \sum_{s=\pm 1} e^{\lambda s (\hat{n}_{d,\uparrow} - \hat{n}_{d,\downarrow})}$

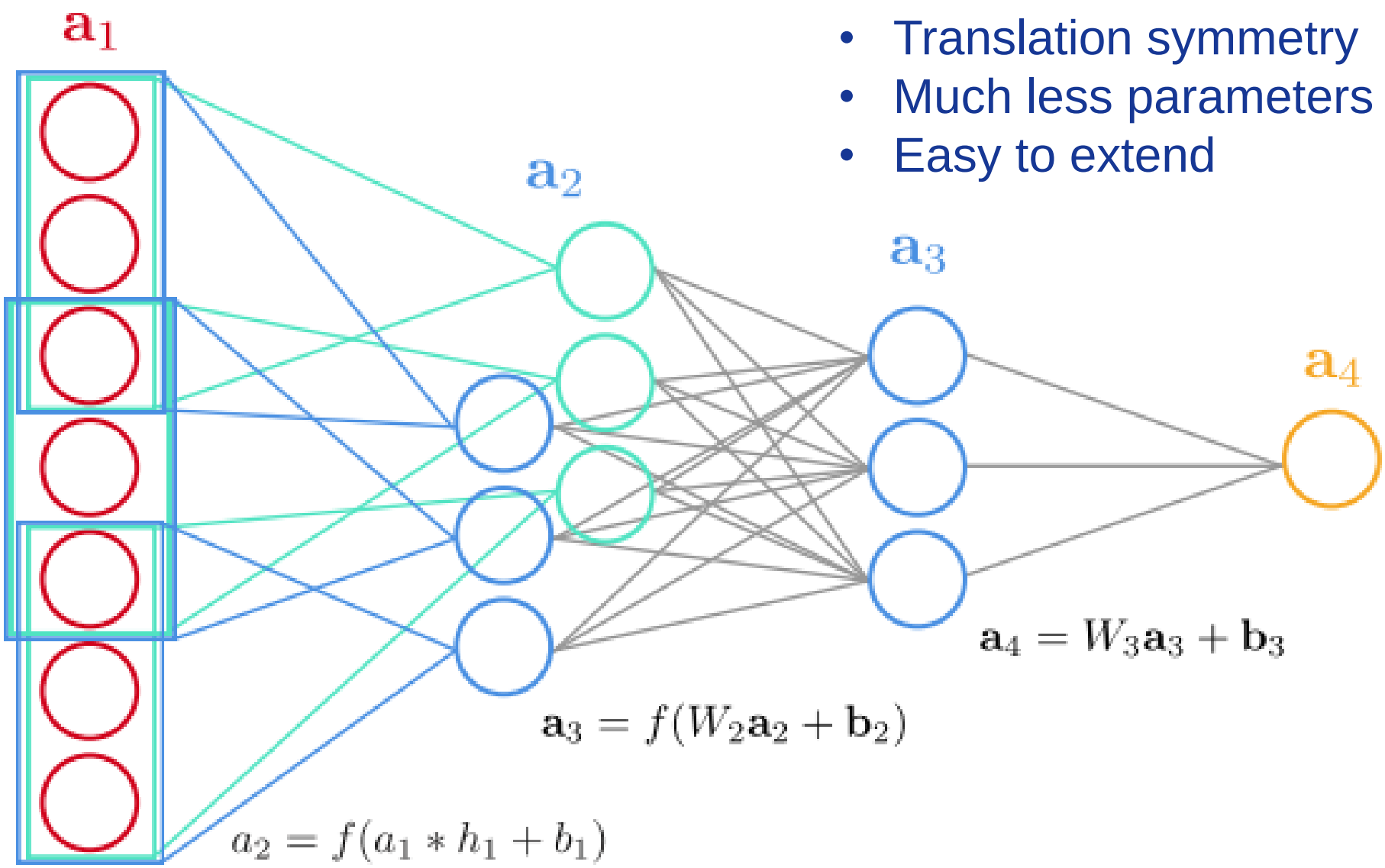
- Then we can use Monte Carlo to sample the auxiliary field configuration space, thus simulating the original asymmetric Anderson model.

- It is not easy to write an explicit effective model to describe the bosonic action here, but we use **neural networks** to do it.

Fully connected neural network



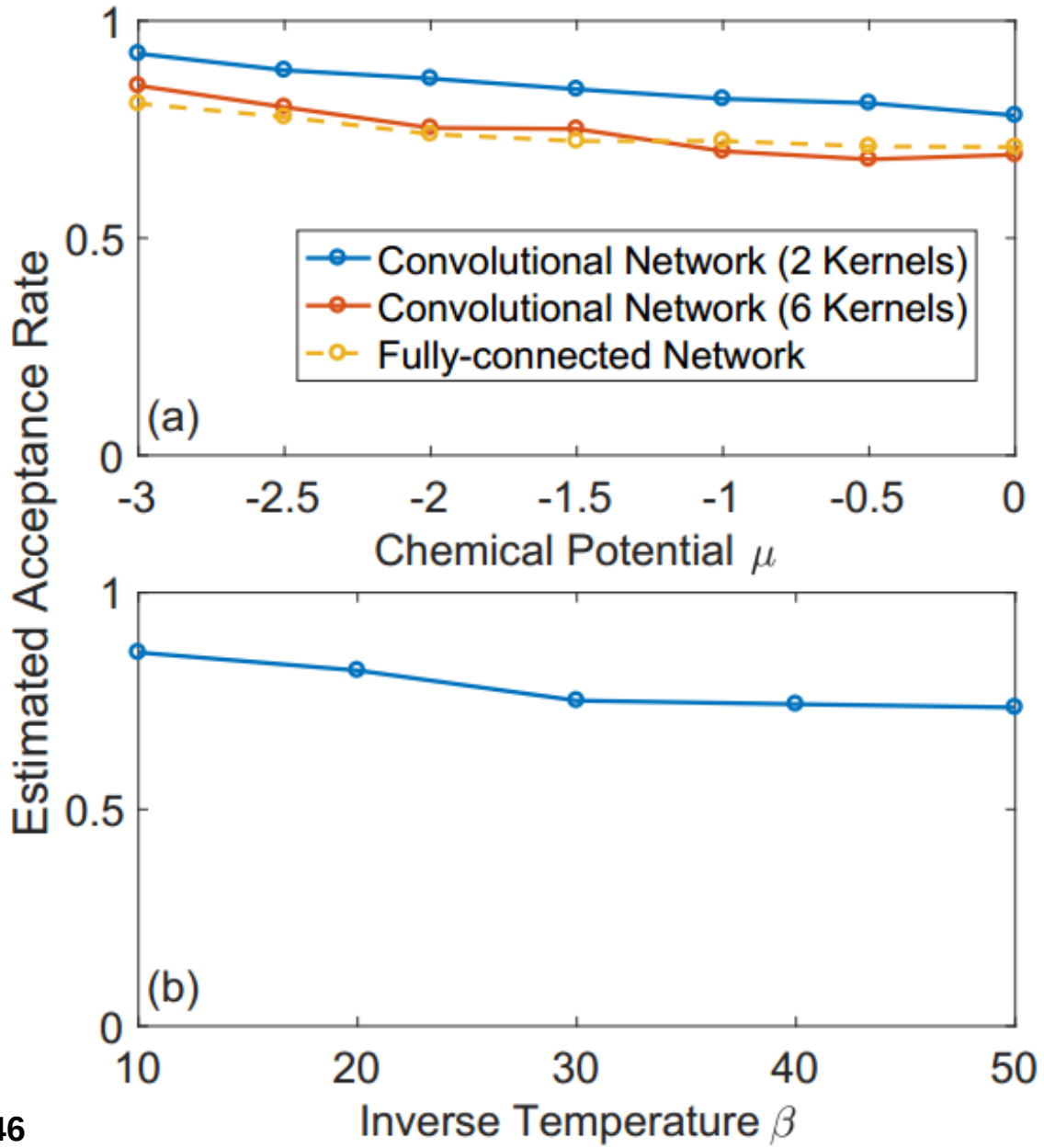
Insights from fully connected NN



- Translation symmetry
- Much less parameters
- Easy to extend

H. Shen, J. Liu, L. Fu, PRB 97, 205140 (2018)

Results of convolutional network



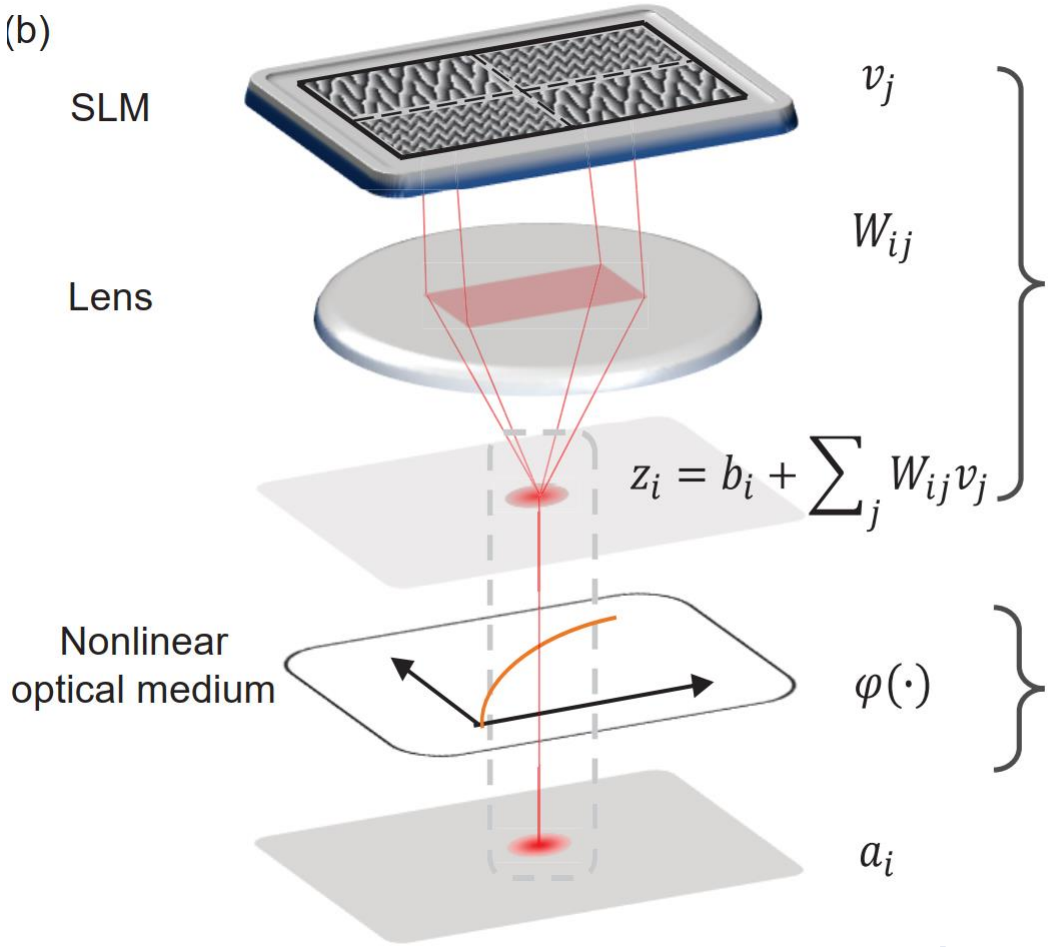
Huitao Shen

(MIT)

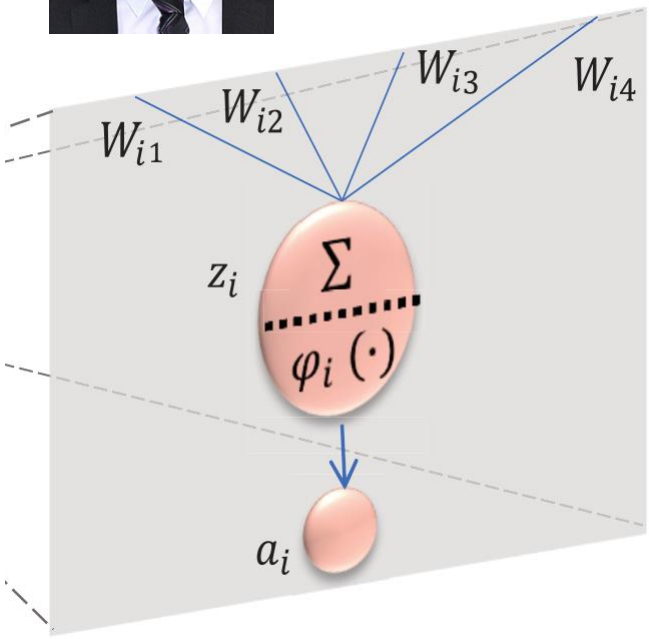
- DNNs can generally work for different chemical potential and temperature.

H. Shen, J. Liu, L. Fu, PRB 97, 205140 (2018)

Part 2: All optical neural network (Physics→ ML)

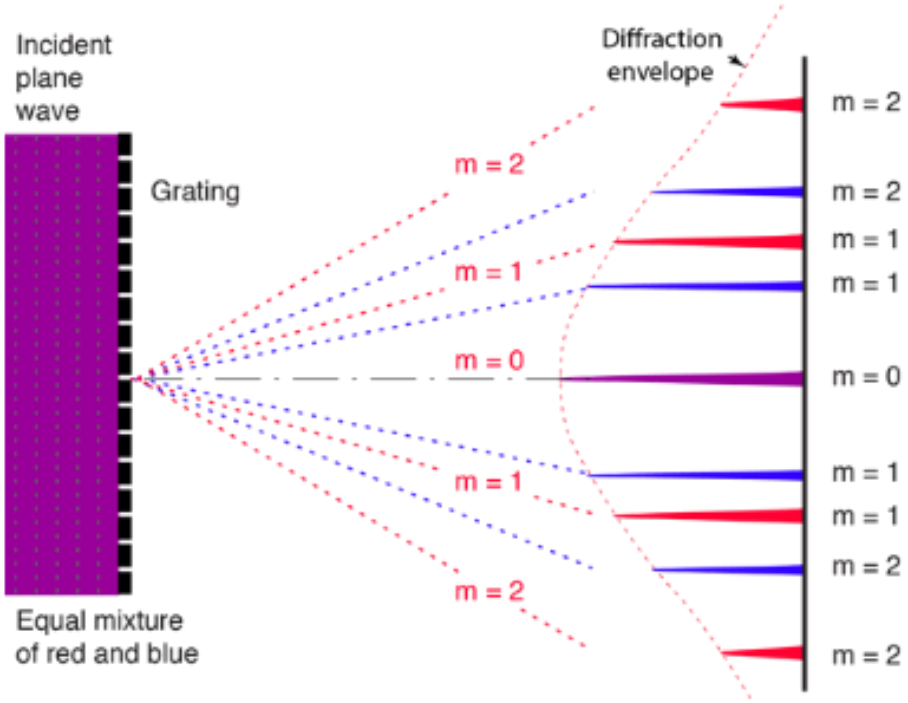


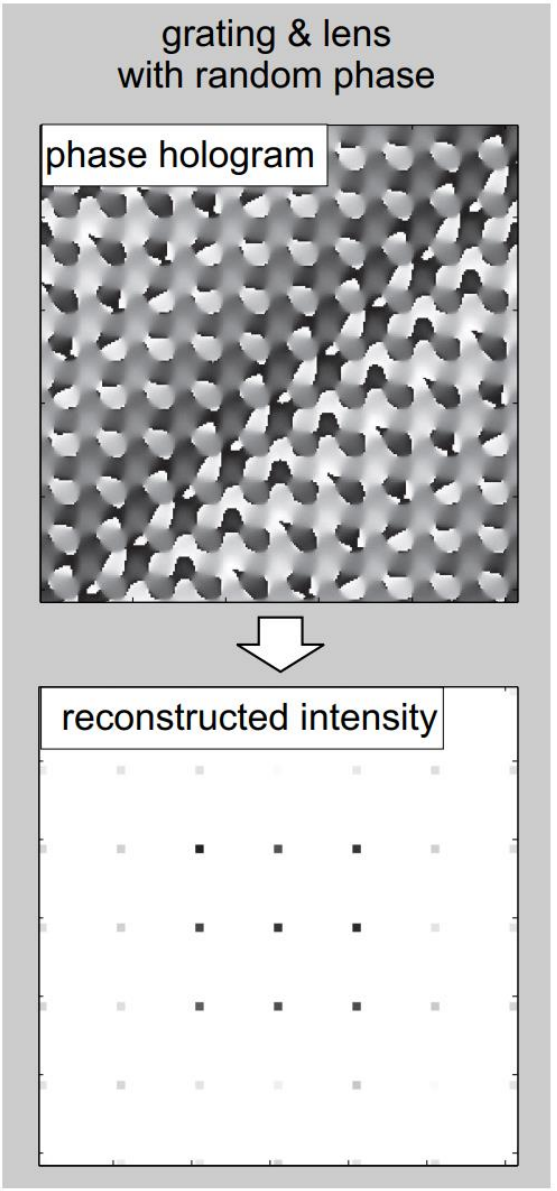
Shengwang Du (HKUST)



- At the speed of light
- Intrinsic infinite parallel calculations
- Robust against local errors

Spatial light modulator: linear transformation



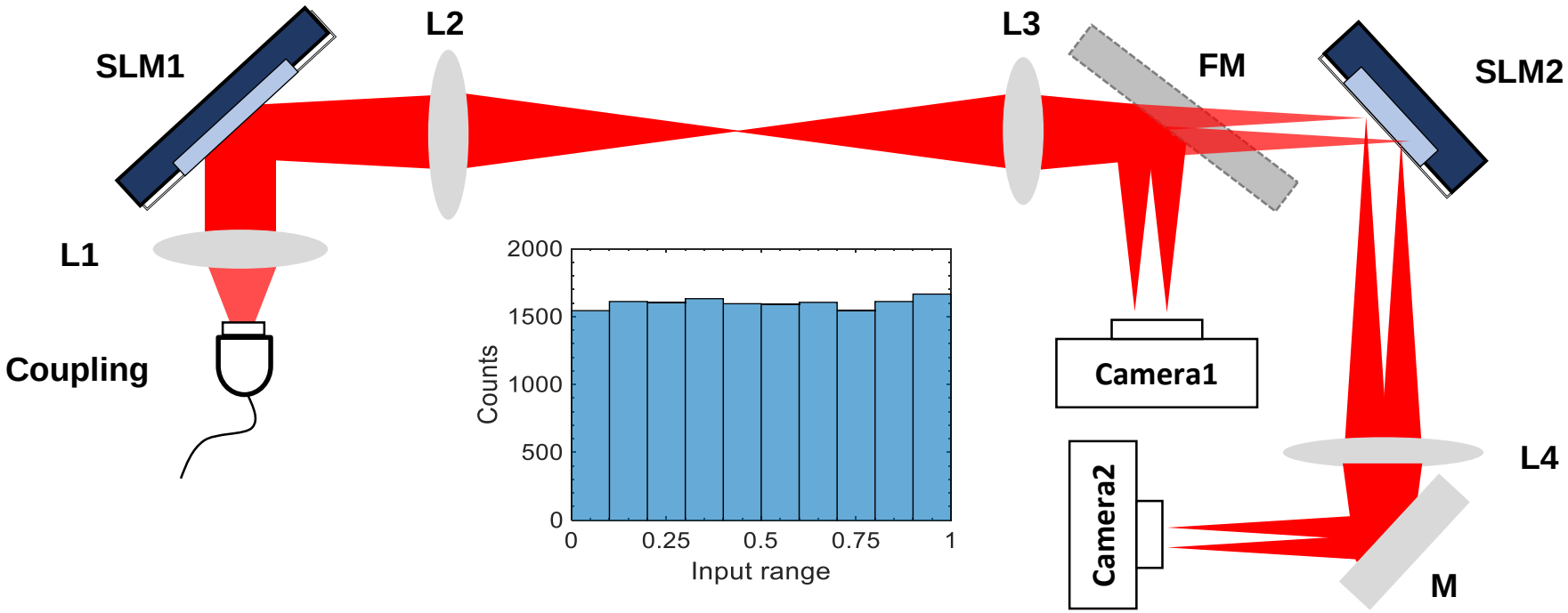


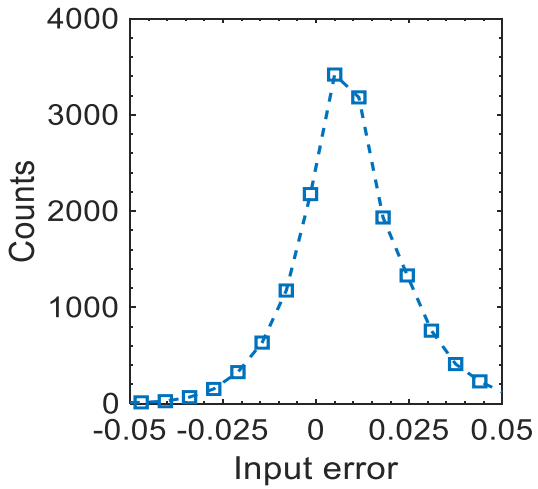
- Use grating to control the relation between the input light and the output light

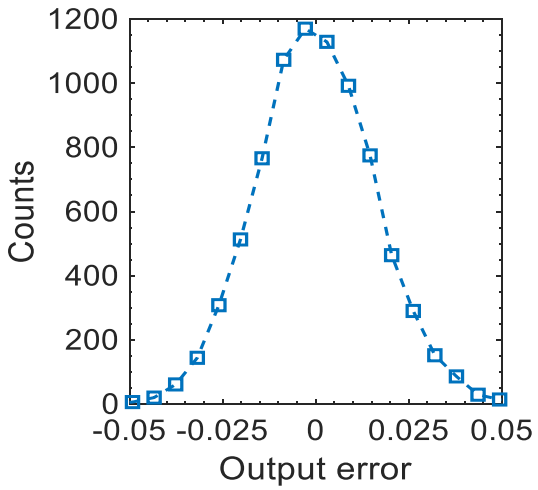
→ linear transformation

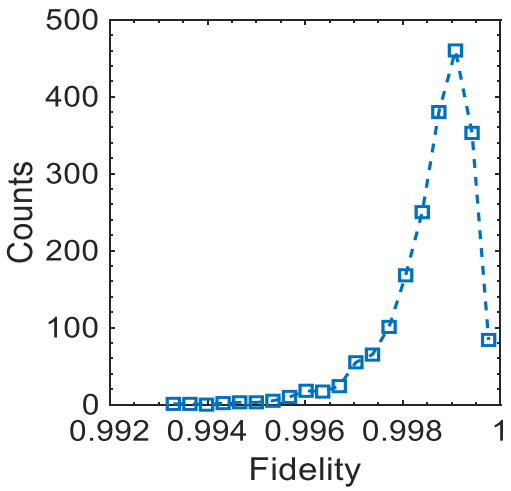
$$I_{out} = W I_{in}$$

Test the linear transformation

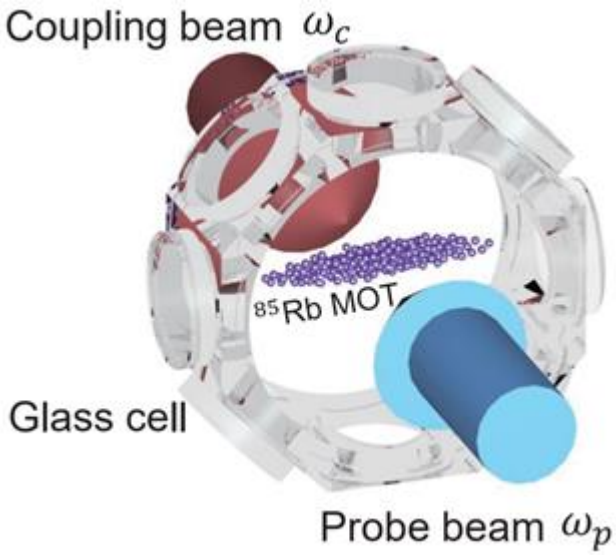


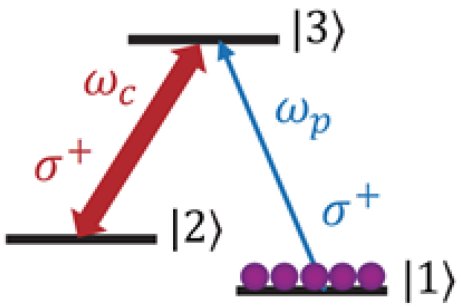




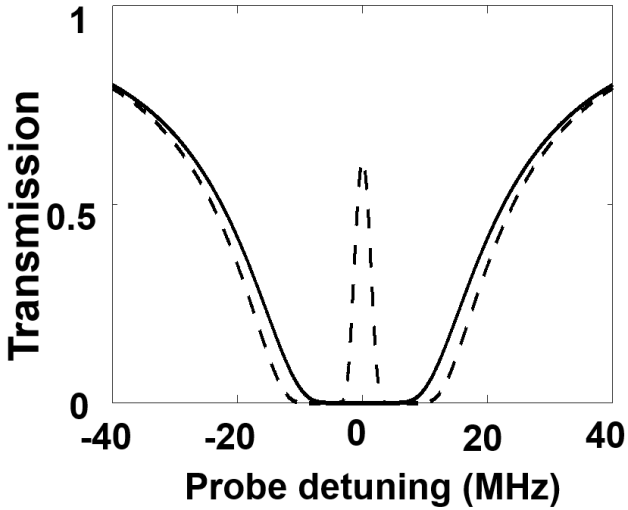


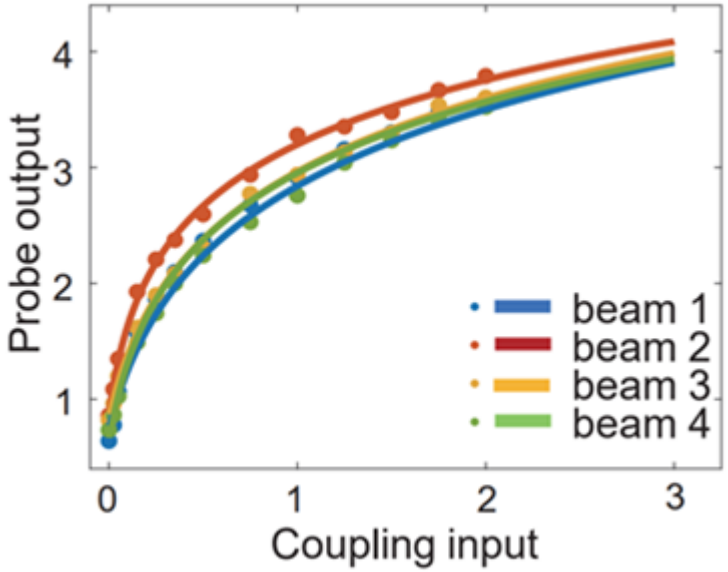
Non-linear function in optics

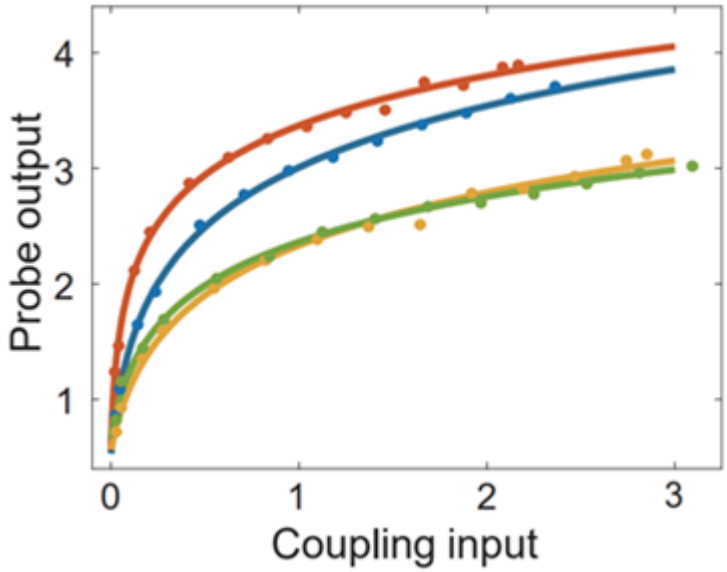




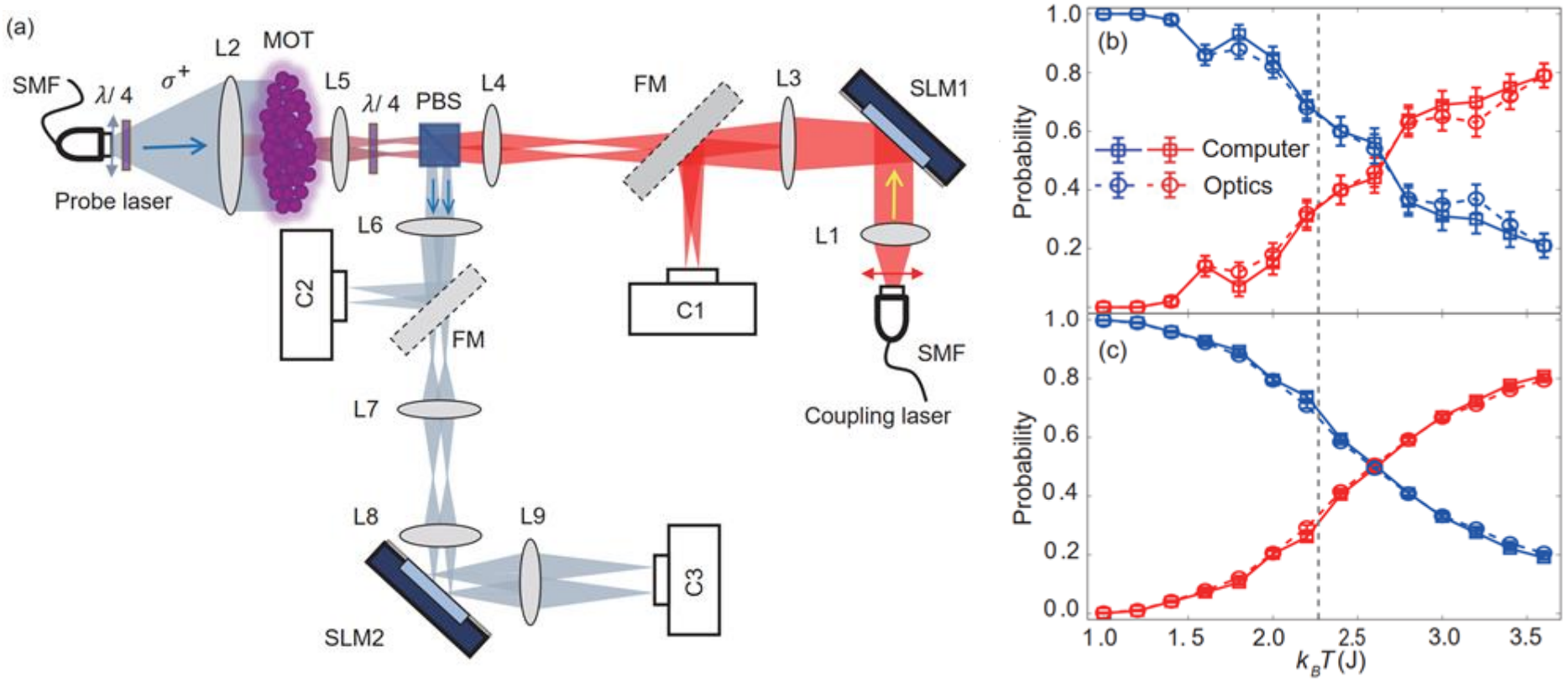
Electromagnetically induced transparency







Test the fully connected neural network



- We can reproduce well the results from the fully connected neural networks $(16, 4, 2)$ in computer

Ying Zuo, et al. Optica 6, 1132 (2019)

Summary

1. SLMC can be generally applied in both bosonic and fermionic systems to speed up MC simulations.

2. SLMC is a **bridge** between the numerical simulation and analytic studies for many-body physics:

   - Better understanding from analytic studies can give us better effective model and better efficiency.

   - The learned effective Model in SLMC is a good starting point, and can give us more insights for analytic studies.

3. SLMC also offers us a **framework** to integrate machine learning techniques into Monte Carlo simulations.

4. We have already realized the AONN in experiments.

Thank you for your attention