Self-learning Monte Carlo Method

Zi Yang Meng

（孟子杨）

Institute of Physics, Chinese Academy of Sciences

Know thyself

"Know thyself" (Greek: γνῶθι σεαυτόν, gnothi seauton)

one of the Delphic maxims and was inscribed in the pronaos (forecourt) of the Temple of Apollo at Delphi

Delphic Maxims

"Know thyself" (Greek: γνῶθι σεαυτόν, gnothi seauton). Thales of Miletus (c. 624 – c. 546 BC)

"nothing in excess" (Greek: μηδέν άγαν). Solon of Athens (c. 638 – 558 BC)

"make a pledge and mischief is nigh" (Greek: Ἐγγύα πάρα δ'ἄτη).

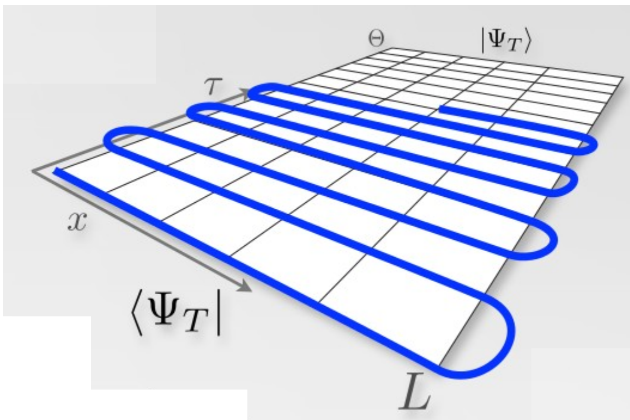
Collaborators and Referencs

- Xiao Yan Xu, IOP, CAS
- Jiuwei Liu, Massachusetts Institute of Technology
- Yang Qi, Massachusetts Institute of Technology
- Liang Fu, Massachusetts Institute of Technology

- Self-Learning Monte Carlo Method, arXiv:1610.08376
- Self-Learning Monte Carlo Method in Fermion Systems, arXiv:1611.
- Self-Learning Determinantal Quantum Monte Carlo Method, arXiv:1612.

Quantum Monte Carlo simulation

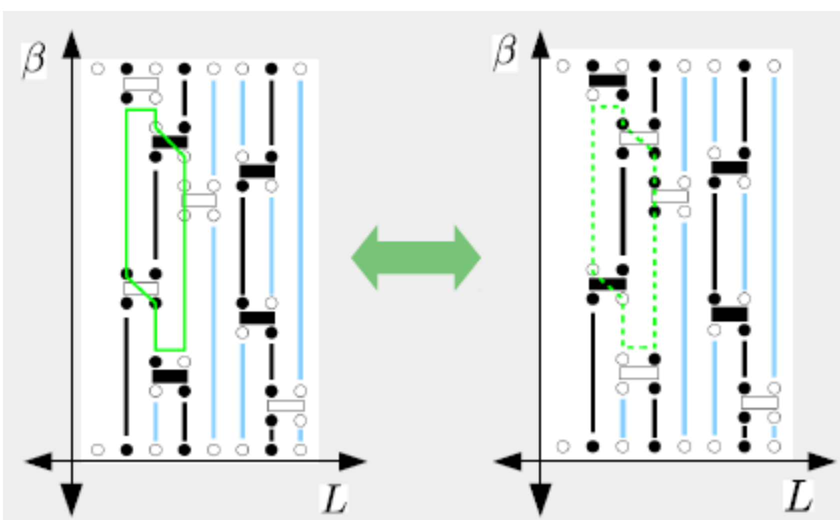
- Determinantal QMC for fermions



Hubbard-like models:
- Metal-Insulator transition
- Unconventional superconductivity
- Non-Fermi-liquid
- Interaction effects on topological state of matter

……

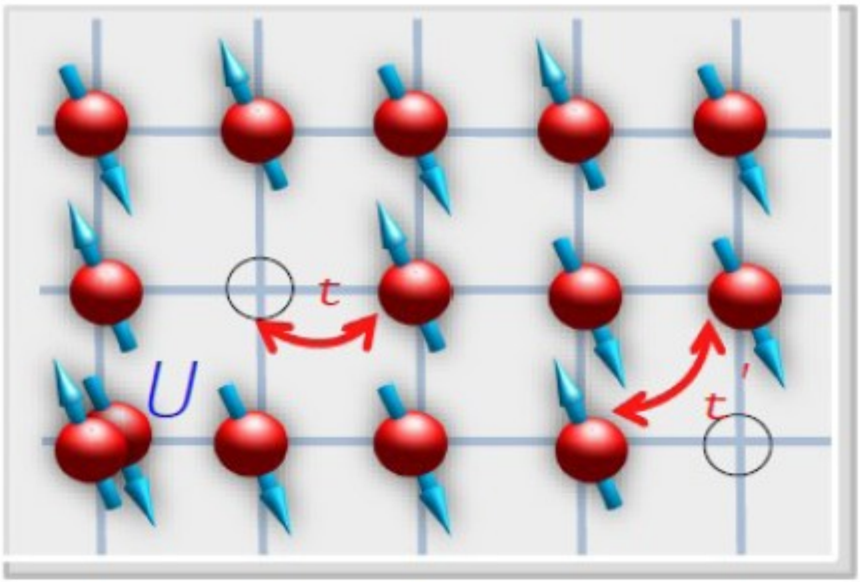
- World-line QMC for bosons

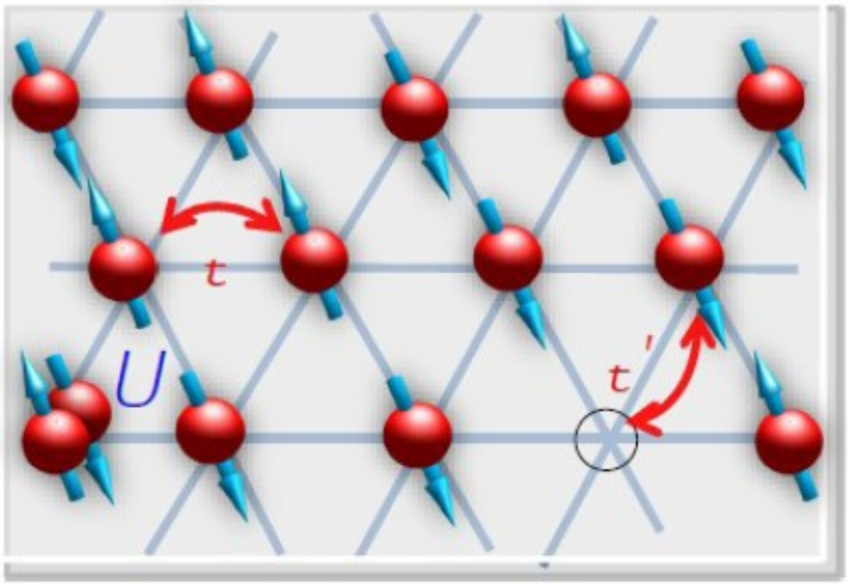


Heisenberg-like models:
- Quantum magnetism
- Phase transition and critical phenomena
- Quantum spin liquids
- Quantum spin ice

……

Basic problem





Partition function: $$Z = \mathrm{Tr}\left[e^{-\beta(\hat{H}-\mu\hat{N})}\right] = \sum_n \langle n|e^{-\beta(\hat{H}-\mu\hat{N})}|n\rangle$$

Observables: $$\langle\hat{A}\rangle = \frac{\mathrm{Tr}\left[\hat{A}e^{-\beta(\hat{H}-\mu\hat{N})}\right]}{\mathrm{Tr}\left[e^{-\beta(\hat{H}-\mu\hat{N})}\right]} = \frac{\sum_n \langle n|\hat{A}e^{-\beta(\hat{H}-\mu\hat{N})}|n\rangle}{\sum_n \langle n|e^{-\beta(\hat{H}-\mu\hat{N})}|n\rangle}$$

Fock space: $$\{|n\rangle\} \sim 2^{N_e}\ \left(e^{N_e \ln(2)}\right) \qquad 4^{N_e}\ \left(e^{N_e \ln(4)}\right)$$

Monte Carlo simulation

- Widely used in statistical and quantum many-body physics
- Unbiased: statistical error $1/\sqrt{N}$
- Universal: applies to any model without sign problem

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

- Markov chain Monte Carlo is a way to do important sampling

$$\cdots \to \mathcal{C}_{i-1} \to \mathcal{C}_i \to \mathcal{C}_{i+1} \to \cdots$$

- Distribution of $\mathcal{C}$ converges to the Boltzmann distribution $W(\mathcal{C})$
- Observable can be measured from a Markov chain

$$\langle O \rangle = \frac{\sum_{\mathcal{C}} O(\mathcal{C}) W(\mathcal{C})}{\sum_{\mathcal{C}} W(\mathcal{C})} = \frac{1}{\mathcal{N}} \sum_i O(\mathcal{C}_i)$$

Autocorrelation time

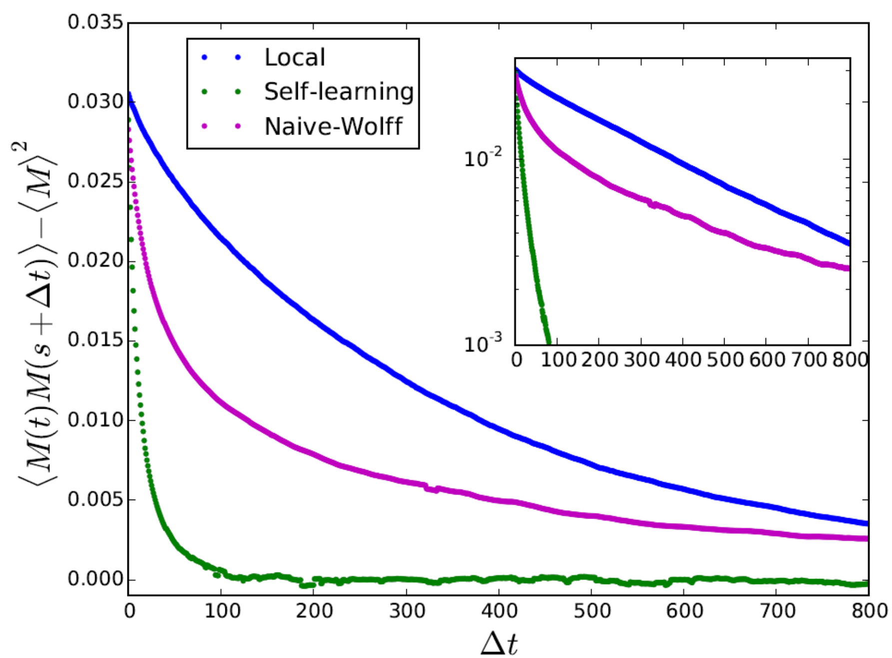
- Markov process, Monte Carlo time sequence

$$\cdots \to O(t-1) \to O(t) \to O(t+1) \to \cdots$$

$$O(t) = O[\mathcal{C}(t)]$$

- Autocorrelation function

$$A_O(\Delta t) = \langle O(t) O(t+\Delta t)\rangle - \langle O(t)\rangle^2 \propto e^{-\Delta t/\tau}$$



Monte Carlo simulation

$$\cdots \to \mathcal{C}_{i-1} \to \mathcal{C}_i \to \mathcal{C}_{i+1} \to \cdots$$

- Detailed balance guarantees the Markov process converges to desired distribution

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

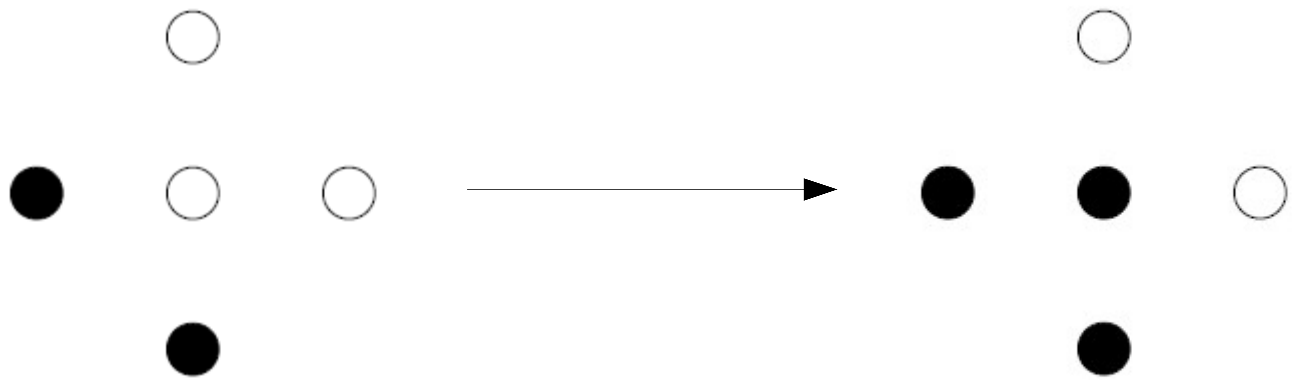
- Metropolis-Hastings algorithm: proposal – acceptance/rejection

$$p(\mathcal{C} \to \mathcal{D}) = q(\mathcal{C} \to \mathcal{D})\alpha(\mathcal{C} \to \mathcal{D})$$

$$\alpha(\mathcal{C} \to \mathcal{D}) = \min\{1, \frac{W(\mathcal{D})q(\mathcal{D} \to \mathcal{C})}{W(\mathcal{C})q(\mathcal{C} \to \mathcal{D})}\}$$

- N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953)
- W. H. Hastings, Biometrika **57**, 97 (1970)

Metropolis algorithm: local update



- Local update $$q(\mathcal{C} \to \mathcal{D}) = q(\mathcal{D} \to \mathcal{C}) = \frac{1}{N}$$
- Acceptance ratio $$\alpha(\mathcal{C} \to \mathcal{D}) = \min\{1, \frac{W(\mathcal{D})}{W(\mathcal{C})}\}$$

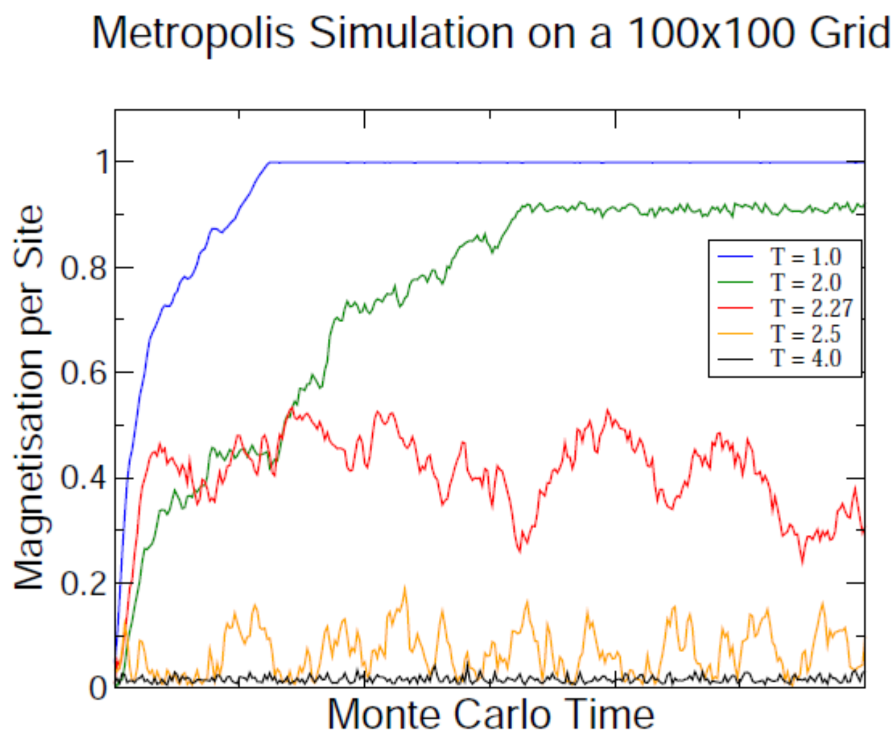
$$\frac{W(\mathcal{D})}{W(\mathcal{C})} = e^{-\beta(E(\mathcal{D})-E(\mathcal{C}))} = e^{-\beta([-1-1+1+1]-[1+1-1-1])} = 1$$

➢ N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953)

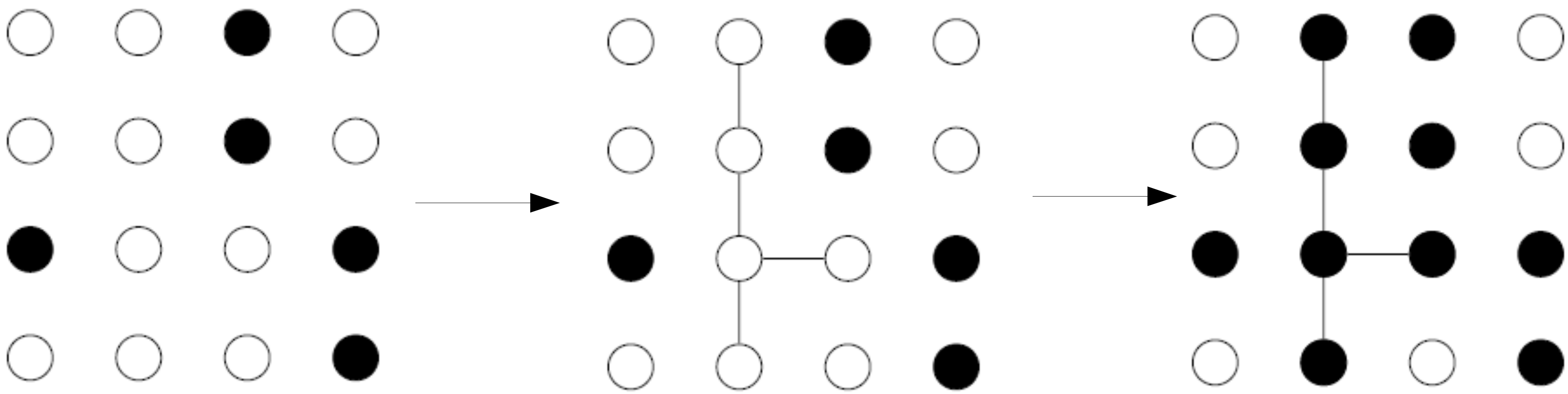
Critical slowing down

- Dynamical relaxation time diverges at the critical point: critical system is slow to equilibrate.

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



Wolff algorithm: cluster update



- A cluster is built from bonds
- Probability of activating a bond is cleverly designed

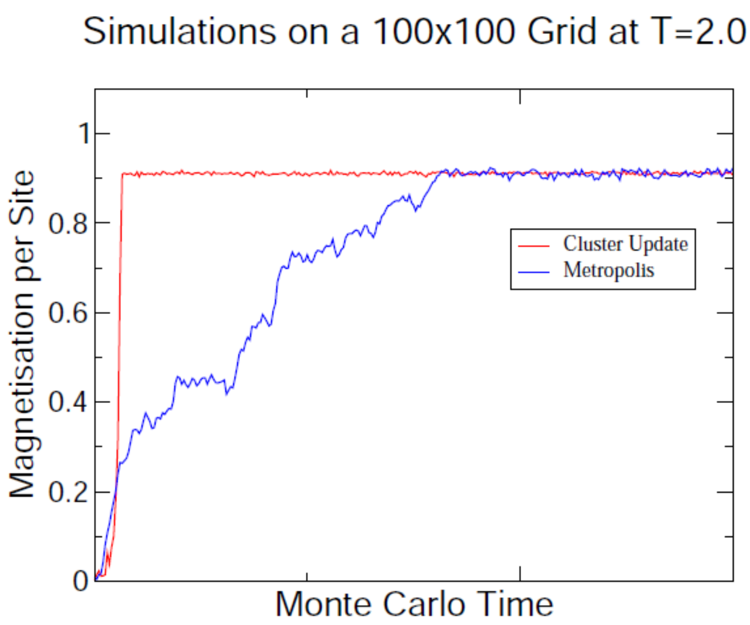
$$q(i \to j) = 1 - e^{\min\{0, -2\beta S_i S_j\}}$$

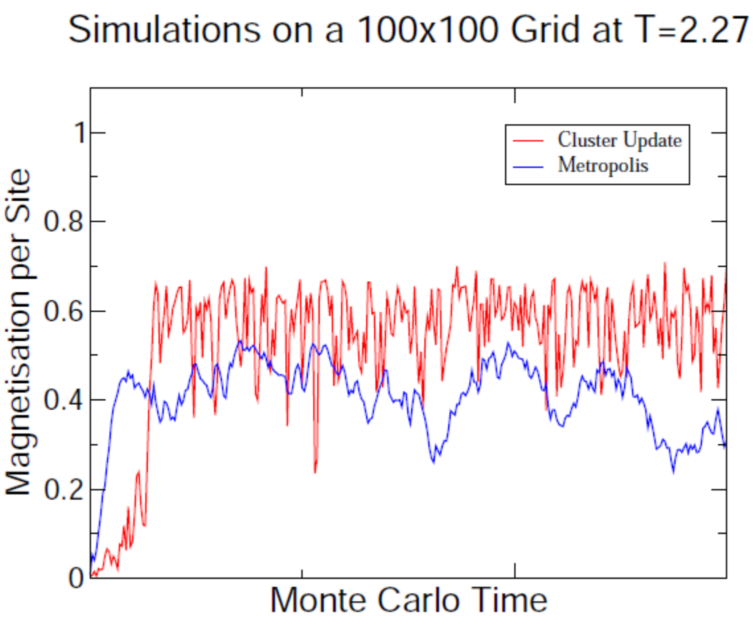
$$\frac{q(\mathcal{A} \to \mathcal{B})}{q(\mathcal{B} \to \mathcal{A})} = \prod_{\langle i,j \rangle, i \in c, j \notin c} \frac{1 - q(i \to j)_{\mathcal{A}}}{1 - q(i \to j)_{\mathcal{B}}} = \prod_{\langle i,j \rangle, i \in c, j \notin c} e^{-2\beta (S_i^{\mathcal{A}} S_j^{\mathcal{A}} - S_i^{\mathcal{B}} S_j^{\mathcal{B}})} = \frac{W(\mathcal{B})}{W(\mathcal{A})}$$

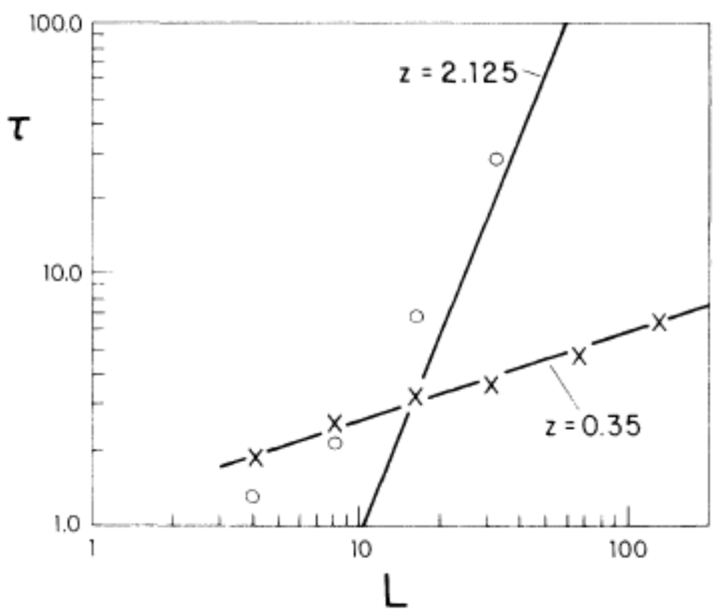
- an ideal acceptance ratio $\alpha(\mathcal{A} \to \mathcal{B}) = 1$

➢ U. Wolff, Phys. Rev. Lett. **62**, 361 (1989)

Reduce critical slowing down



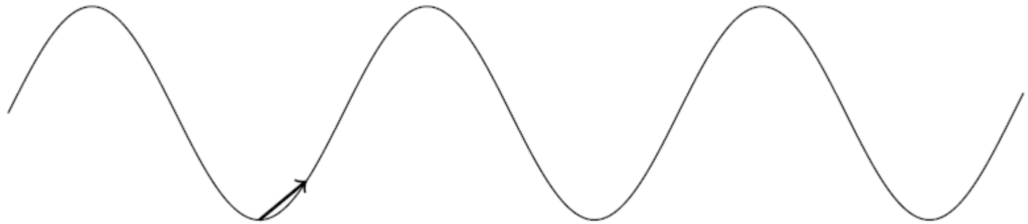




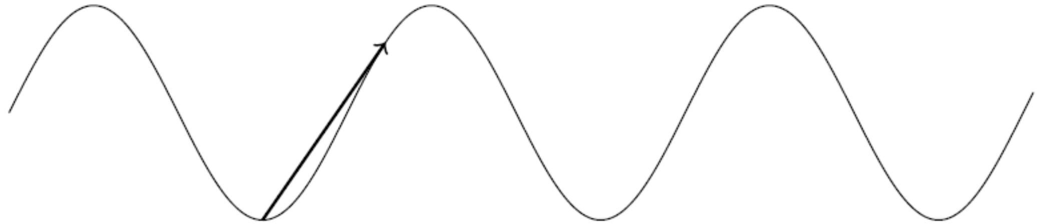
➢ Swendsen and Wang, Phys. Rev. Lett. **58**, 86 (1987)

Learn thyself

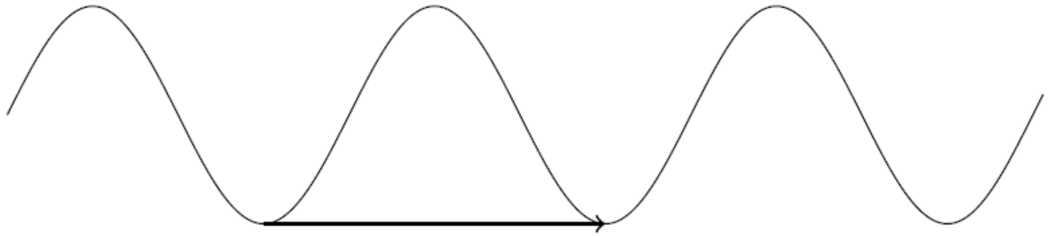
- Step too small: small difference, high acceptance



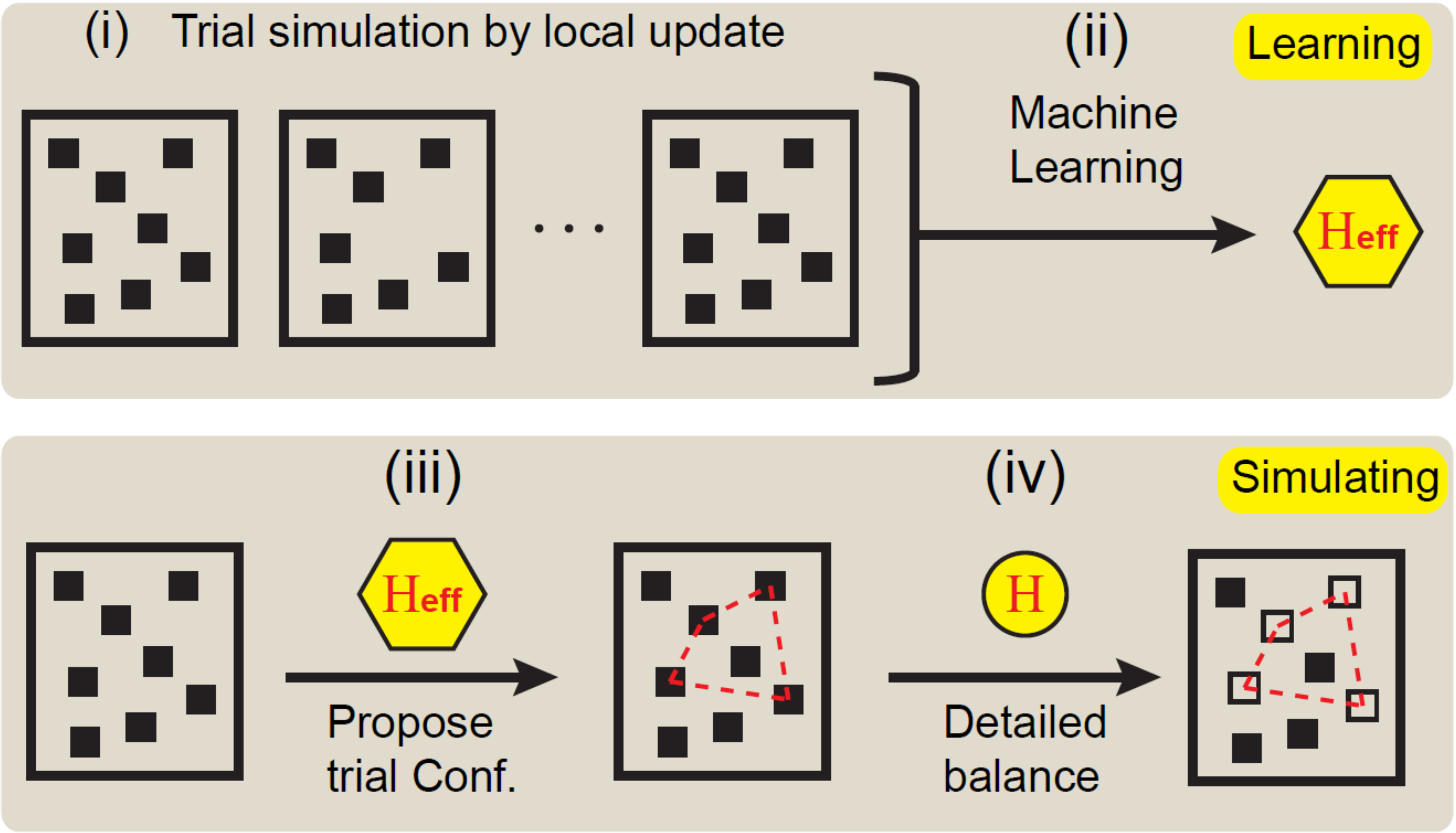
- Step too large: big difference, low acceptance



- Global update: explore the low-energy configurations



Self-Learning



Example I: SLMC for Bosons

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

Ising transition with $T_c = 2.493$

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

- The self-learning update: cluster is constructed using the effective model

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

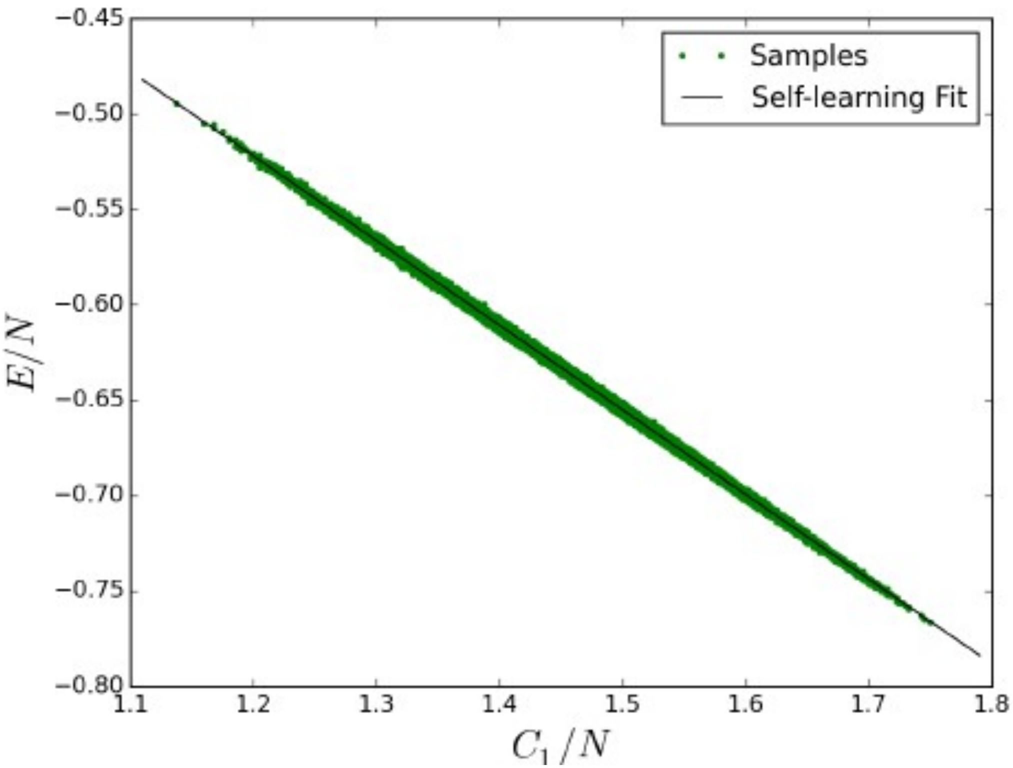
- The acceptance ratio:

$$\alpha(\mathcal{C}\to\mathcal{D}) = \min\{1, \frac{W(\mathcal{D})W_{\text{eff}}(\mathcal{D})}{W(\mathcal{C})W_{\text{eff}}(\mathcal{D})}\} = \min\{1, e^{-\beta[(E(\mathcal{D})-E_{\text{eff}}(\mathcal{D}))-(E(\mathcal{C})-E_{\text{eff}}(\mathcal{C}))]}\}$$

- The acceptance ratio can be very high, autocorrelation time can be very short
- effective model capture the low-energy physics

Example I: SLMC for Bosons

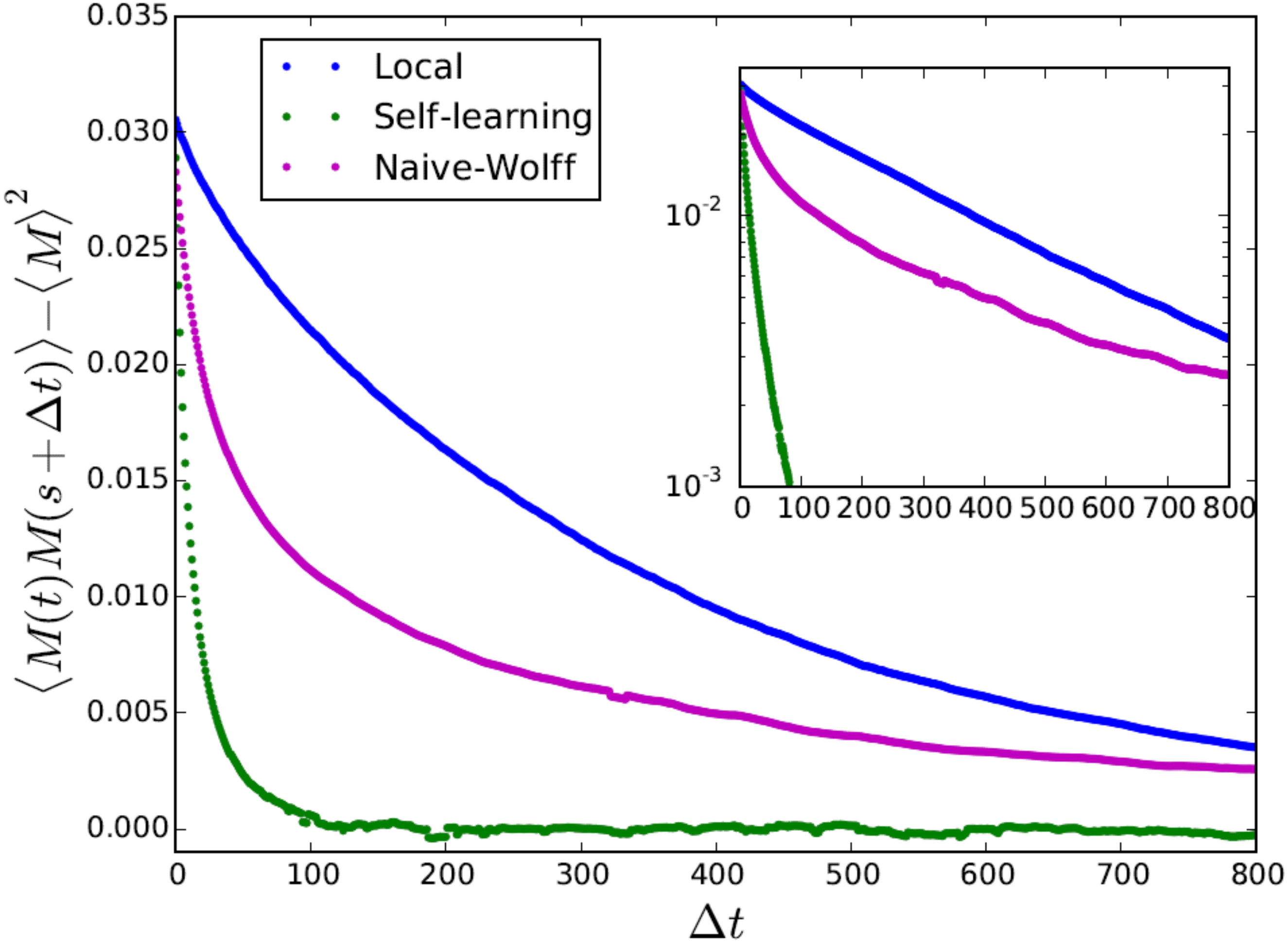
- Generate configurations with local update, at T=5 > Tc.
- Perform linear regression
- Generate configurations with reinforced learning at Tc



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

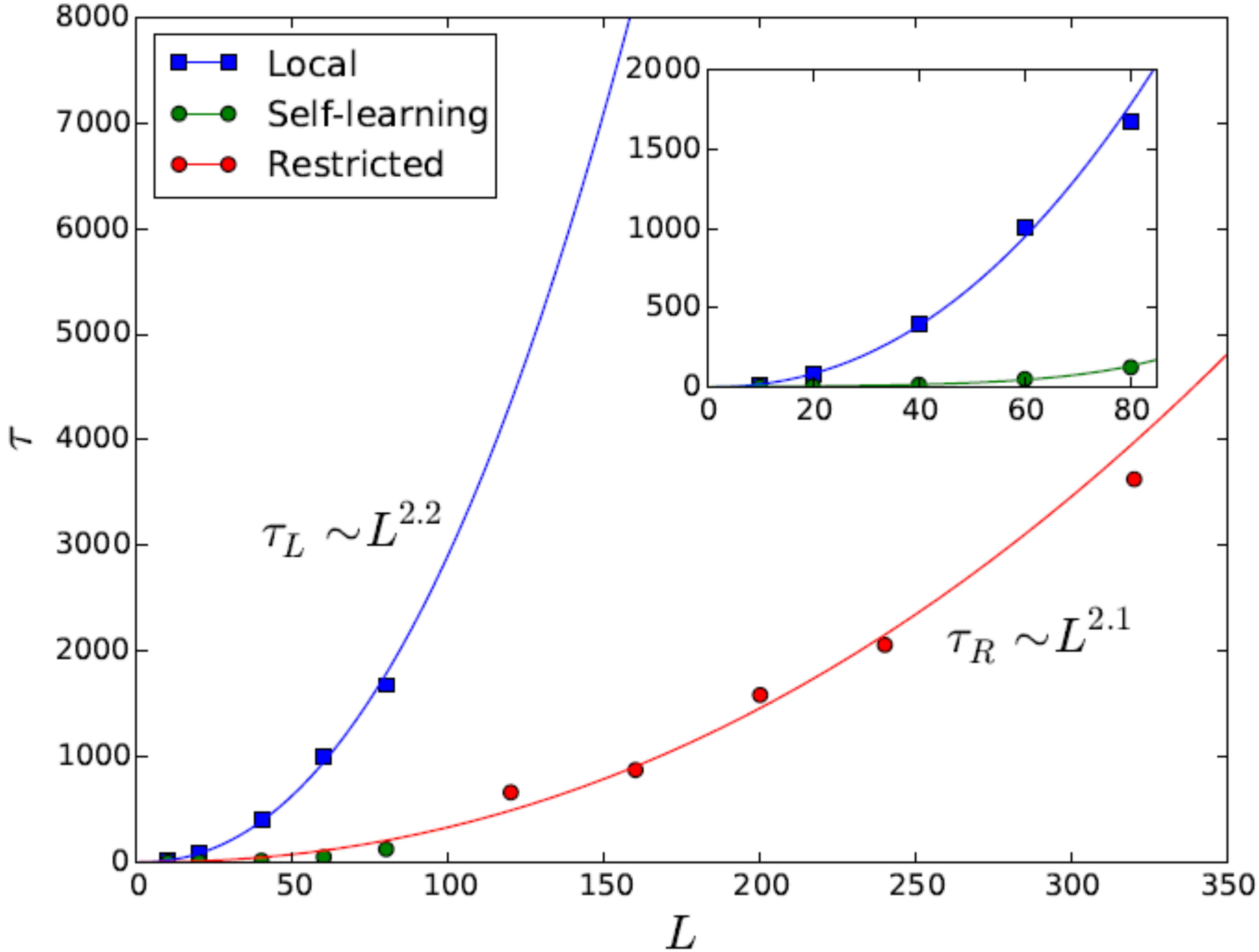
	J_1	J_2	J_3
Train 1	1.2444	-0.0873	-0.0120
Train 2	1.1064	-	-

Example I: SLMC for Bosons



System size 40x40 at Tc

Example I: SLMC for Bosons



- Speedup of 10~20 times

Example II: SLMC for Fermions

- Double exchange model

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

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

- Computational complexity

$$O(\tau_0 \times L^{3d} \times L^d) = O(\tau_0 \times L^{4d})$$

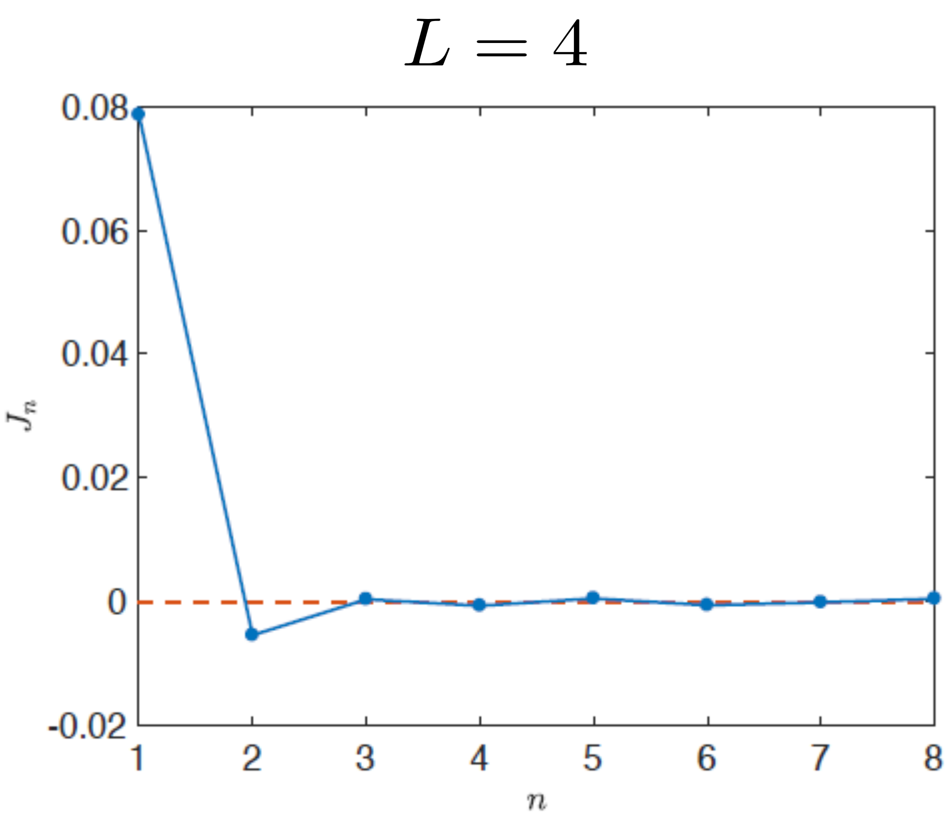
- Fit effective model

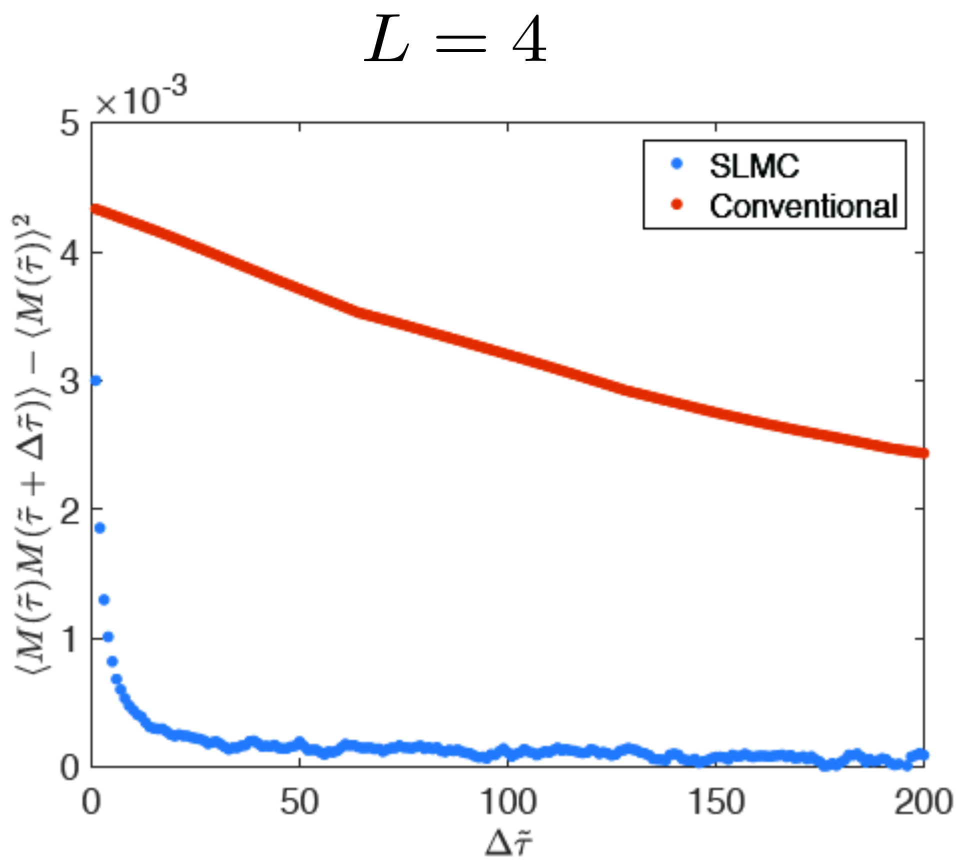
$$W[\phi] \simeq e^{-\beta H_{\text{eff}}[\phi]}$$

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

Example II: SLMC for Fermions

- effective model captures the low-energy physics, RKKY interaction.
- only need to learn from small system sizes





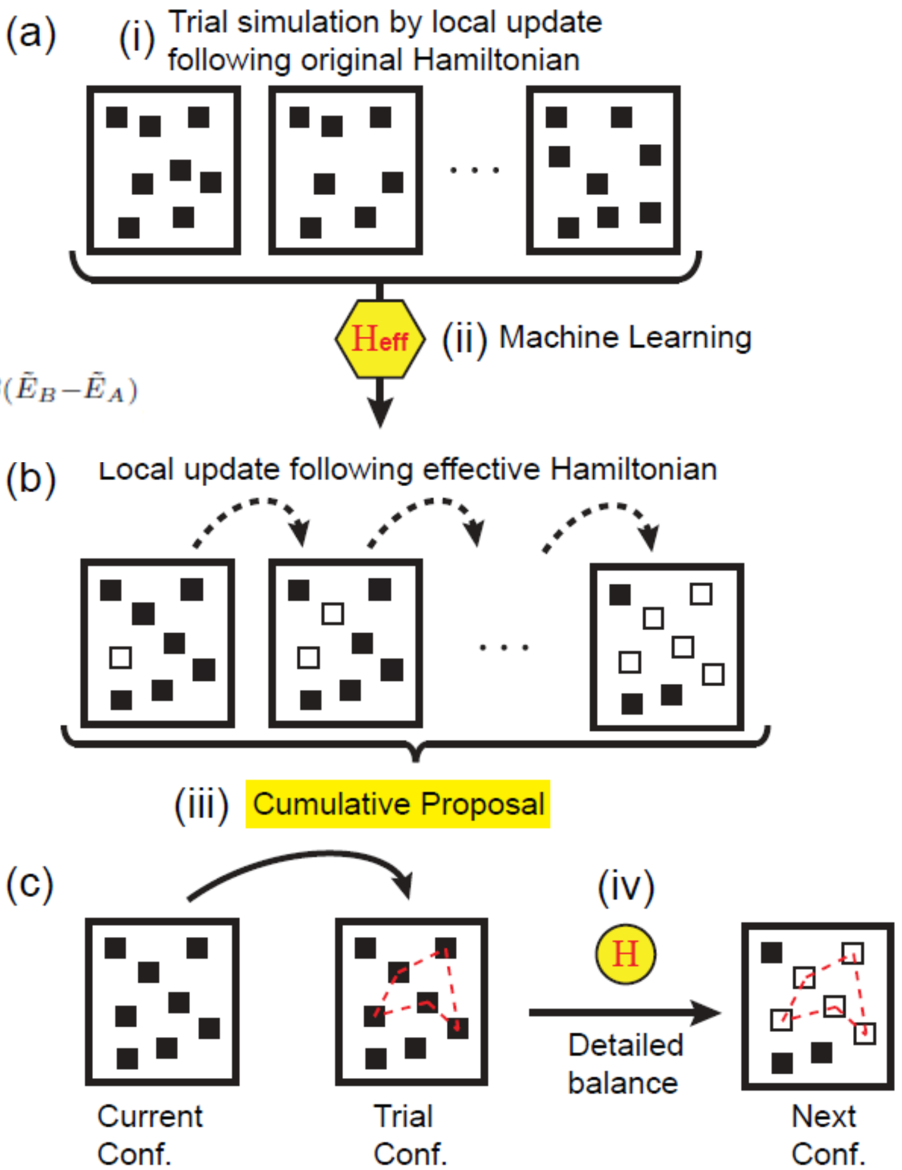
Example II: SLMC for Fermions

- Cumulative update

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

$$\frac{\mathcal{S}(A\to B)}{\mathcal{S}(B\to A)}=\prod_{i=0}^{n_c-1}\frac{\tilde{P}(C_i\to C_{i+1})}{\tilde{P}(C_{i+1}\to C_i)}=\prod_{i=0}^{n_c-1}e^{-\beta(\tilde{E}_{i+1}-\tilde{E}_i)}=e^{-\beta(\tilde{E}_B-\tilde{E}_A)}$$

$$p(A\to B)=\min\{1,e^{-\beta(E_B-\tilde{E}_B)-(E_A-\tilde{E}_A)}\}$$

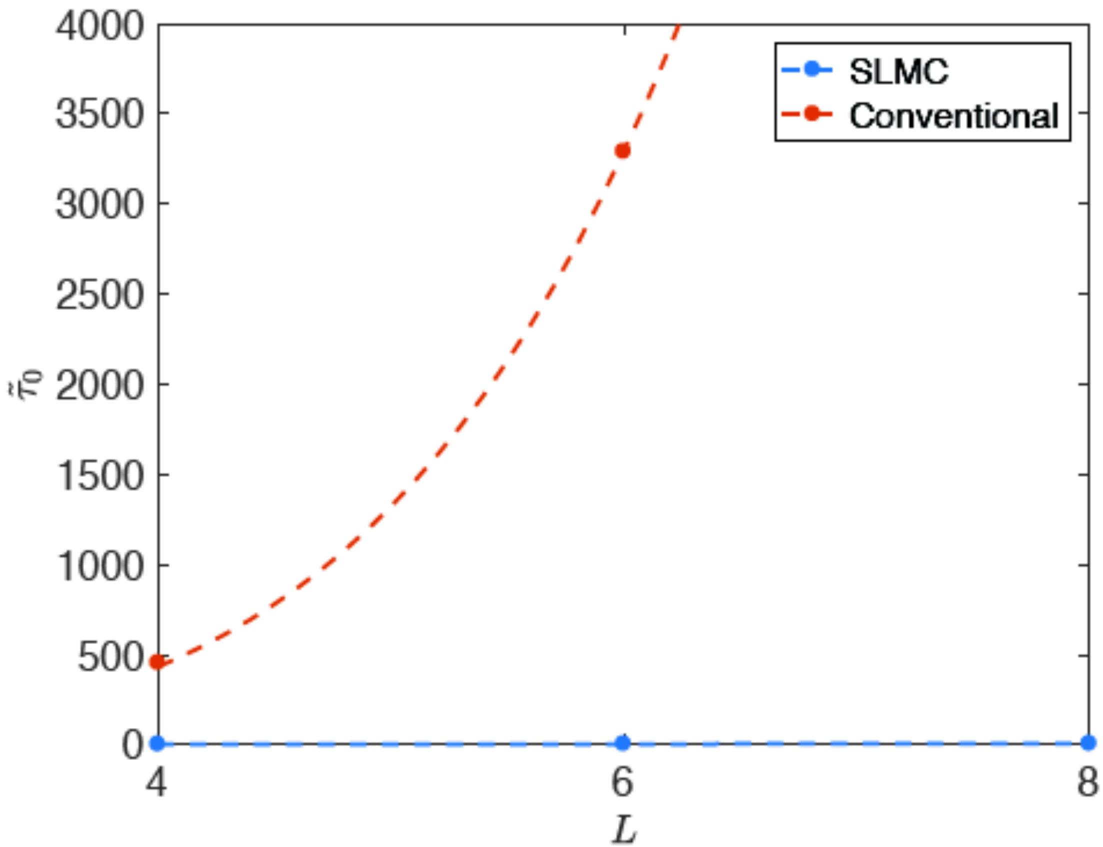


Example II: SLMC for Fermions

- Computation complexity at most

$$O(l_c) + O(\tau_0 \times L^{3d}) = O(\tau_0 \times L^{3d})$$

- Speedup of $O(L^z + L^d) = O(L^{d+z})$



- L=4,6,8, at L=8, 10^3 times faster.