Supplemental material:
Self-Learning Monte Carlo Method: Continuous-Time Algorithm

Yuki Nagai,[1,2] Huitao Shen,[2] Yang Qi,[2] Junwei Liu,[2] and Liang Fu[2]

[1]*CCSE, Japan Atomic Energy Agency, 178-4-4, Wakashiba, Kashiwa, Chiba, 277-0871, Japan*

[2]*Department of physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

I. SELF LEARNING UPDATES

The probability of moving from a configuration c_i to a configuration c_j can be split into the probability of proposing the move and the probability of accepting it, $p(c_i \to c_j) = p^{\rm prop}(c_i \to c_j)p^{\rm acc}(c_i \to c_j)$. Then the detailed balance principle implies

$$\frac{p^{\rm acc}(c_i \to c_j)}{p^{\rm acc}(c_j \to c_i)} = \frac{p^{\rm prop}(c_j \to c_i)}{p^{\rm prop}(c_i \to c_j)} \frac{w_{c_j}}{w_{c_i}}. \tag{1}$$

In the self-learning CT-AUX, the new configuration c_j is proposed based on the effective weight $w^{\rm eff}_{c_j}$. The probability to propose the move $p^{\rm prop}(c_j \to c_i)/p^{\rm prop}(c_i \to c_j) = w^{\rm eff}_{c_i}/w^{\rm eff}_{c_j}$. Combined with Eq. (1), we obtain the desired acceptance rate. This result can be understood intuitively in the limit that the effective weight $w^{\rm eff}_c$ is equal to the original weight w_c for all configurations. Then we are as if doing the MC update on exactly the original model. Therefore the "global update" from configuration c_i to configuration c_j should always be accepted, i.e., $p^{\rm acc}(c_i \to c_j)/p^{\rm acc}(c_j \to c_i) = w^{\rm eff}_{c_i} w_{c_j}/(w^{\rm eff}_{c_j} w_{c_i}) = 1$.

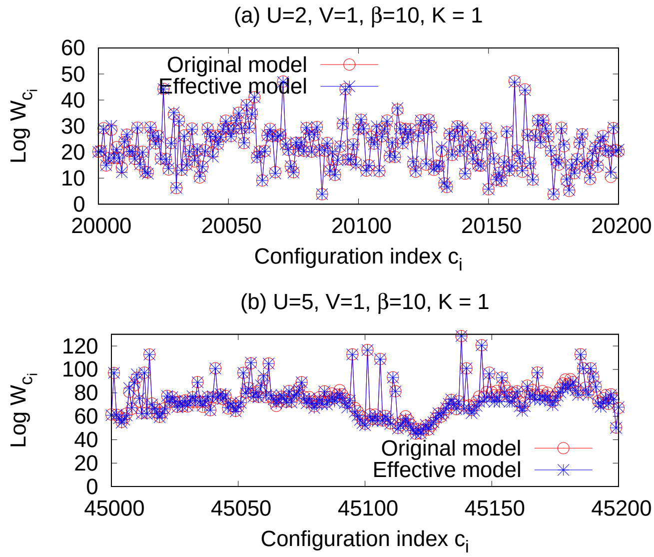

FIG. 1. (Color online) Comparison between the original and effective weights with $m_c = 12$. We set $V = 1$, $\beta = 10$, and $K = 1$. The configurations are generated by the Markov process in the original CTAUX simulation. The expansion order n changes in the simulation.

II. COMPARISON BETWEEN THE ORIGINAL AND EFFECTIVE WEIGHTS

To show the efficiency of the trained DGF, we plot the original and effective weights w_{c_i} and $w^{\rm eff}_{c_i}$. We set $m_s = 12$ $V = 1$, $\beta = 10$, and $K = 1$. The configurations are generated by the Markov process in the original CTAUX simulation. The expansion order n changes in the simulation. As shown in Fig. 1, one can clearly find that the weights between these two methods are quite similar.

III. LOCAL UPDATES IN SL-CTAUX

We show that the calculation cost of the local updates is $O(\langle n \rangle)$. We consider the configuration with the expansion order n and the insertion of a vertex with the auxiliary spin s at the imaginary time τ. The weight $w^{\rm eff}_{n+1}$ is expressed as

$$\ln w^{\rm eff}_{n+1} = \frac{1}{n+1}\sum_{i,j}^{n+1} g(\tau_i - \tau_j)s_i s_j + \frac{1}{n+1}\sum_{i,j}^{n+1} h(\tau_i - \tau_j) + f(n+1), \tag{2}$$

$$= \frac{1}{n+1}\left(n w^{\rm eff}_n + 2s\sum_{j=1}^{n} g(\tau - \tau_j)s_j + g(0) + 2\sum_{j=1}^{n} h(\tau - \tau_j) + h(0)\right) + f(n+1). \tag{3}$$

Thus, the ratio $w^{\rm eff}_{n+1}/w^{\rm eff}_n$ is rewritten as

$$\ln \frac{w^{\rm eff}_{n+1}}{w^{\rm eff}_n} = \frac{1}{n+1}\left(2s\sum_{j=1}^{n} g(\tau - \tau_j)s_j + g(0) + 2\sum_{j=1}^{n} h(\tau - \tau_j) + h(0)\right) - \frac{w^{\rm eff}_n}{n+1} + f(n+1) - f(n). \tag{4}$$

In the case of removal update, The ratio $w^{\mathrm{eff}}_{n-1}/w^{\mathrm{eff}}_{n}$ becomes

$$\ln \frac{w^{\mathrm{eff}}_{n-1}}{w^{\mathrm{eff}}_{n}} = \frac{1}{n-1}\left(-2s\sum_{j=1}^{n} g(\tau-\tau_j)s_j + g(0) - 2\sum_{j=1}^{n} h(\tau-\tau_j)\right.$$
$$\left. + h(0)\right) + \frac{w^{\mathrm{eff}}_{n}}{n-1} + f(n-1) - f(n). \tag{5}$$

Thus, the calculation cost of the local updates is $O(\langle n\rangle)$.