Self-learning Monte Carlo method: Continuous-time algorithm

PHYSICAL REVIEW B **96**, 161102(R) (2017)

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, Massachusetts 02139, USA*
(Received 23 May 2017; published 3 October 2017)

The recently introduced self-learning Monte Carlo method is a general-purpose numerical method that speeds up Monte Carlo simulations by training an effective model to propose uncorrelated configurations in the Markov chain. We implement this method in the framework of a continuous-time Monte Carlo method with an auxiliary field in quantum impurity models. We introduce and train a diagram generating function (DGF) to model the probability distribution of auxiliary field configurations in continuous imaginary time, at all orders of diagrammatic expansion. By using DGF to propose global moves in configuration space, we show that the self-learning continuous-time Monte Carlo method can significantly reduce the computational complexity of the simulation.

DOI: 10.1103/PhysRevB.96.161102

Quantum Monte Carlo (QMC) is an unbiased numerical method for studying quantum many-body systems. A standard QMC scheme for interacting fermion systems is the determinantal QMC method [1–4]. This method uses (1) the Hubbard-Stratonovich transformation to decompose the two-body fermion interaction, and (2) the Suzuki-Trotter decomposition of the partition function to discretize the imaginary-time interval into a large number of time slices. Monte Carlo sampling is performed in the space of auxiliary Hubbard-Stratonovich fields. Recently, a continuous-time modification of the fermionic QMC algorithm was developed [5–8]. In this algorithm, the partition function is expanded in the powers of interaction, and the Monte Carlo simulation is performed by the stochastic sampling of the diagrammatic expansion of interaction terms. Both the number and position of interaction terms on the imaginary-time interval change constantly during the simulation. For both determinantal and continuous-time QMC methods, to compute the weight of each configuration requires integrating out the fermions. This is very time consuming and in practice limits the size of fermion systems in QMC studies.

Recently, we introduced a general method, dubbed self-learning Monte Carlo (SLMC), which speeds up the MC simulation by designing and training a model to propose efficient global updates [9–11]. The philosophy behind SLMC is "first learn, then earn." In the learning stage, trial simulations are performed to generate a large set of configurations and their weights. These data are then used to train an effective model H_{eff}, whose Boltzmann weight $e^{-\beta H_{\text{eff}}}$ fits the probability distribution of the original problem. Next, in the actual simulation, H_{eff} is used as a guide to propose highly efficient global moves in configuration space. Importantly, the acceptance probability of such a global update is set by the detailed balance condition of the original Hamiltonian. This ensures the MC simulation is statistically exact.

The SLMC method is ideally suited for QMC simulation of fermion systems. In the determinantal QMC method, the weight of an auxiliary field configuration $\phi(x)$ is computed by integrating out fermions, which is numerically expensive. In contrast, the effective model $H_{\text{eff}}[\phi(x)]$ is an explicit functional of $\phi(x)$, and its Boltzmann weight can be computed fast. Therefore, the SLMC method has a far less computational cost than the original method, leading to a dramatic speedup, as we demonstrated in previous works [10].

In this Rapid Communication, we extend the SLMC to continuous-time quantum Monte Carlo algorithms for fermion systems. Based on a theoretical analysis and numerical study, we demonstrate that our continuous-time SLMC reduces the computational complexity of the simulation in the low-temperature or strong-coupling regime, where the autocorrelation time in the standard method becomes large. The key ingredient of our method is an effective model for the diagrammatic interaction expansion in continuous time, which we term the "diagram generating function" (DGF). The form of DGF is constrained by the symmetry of the Hamiltonian under study. The parameters in DGF are trained and optimized in the learning stage of SLMC. As an example, we implement SLMC to simulate the single impurity Anderson model [12], using the continuous-time auxiliary-field (CT-AUX) method [7,8,13,14]. The DGF for this model is found to take a remarkably simple form, and reproduce with very high accuracy the exact distribution of auxiliary fields in continuous time, to all orders of the diagrammatic expansion. We demonstrate the speedup of SLMC in comparison to the standard CT-AUX, and find the acceleration ratio increases with the average expansion order.

This Rapid Communication is organized as follows: We first briefly review the CT-AUX algorithm in the Anderson impurity model, after which we give a detailed introduction to the self-learning CT-AUX algorithm, and discuss the physical ideas behind the DGF. Then, we show the performance of our algorithm on the Anderson model. Finally, we analyze the complexity of the algorithm. The technical details are shown in the Supplemental Material [15].

While this work was being performed, a related work [16] also extending SLMC [9,10,17] to a continuous-time domain appeared. Unlike ours, that work uses an interaction

*Present address: Department of Physics, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China.

expansion without an auxiliary field, and does not analyze the computational complexity of continuous-time SLMC to demonstrate its speedup.

CT-AUX method. The Hamiltonian of the single impurity Anderson model is written as the combination of a free fermion part and an interaction part [8],

$$H = H_0 + H_1, \tag{1}$$

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

$$H_1 = U(n_\uparrow n_\downarrow - (n_\uparrow + n_\downarrow)/2) - K/\beta, \tag{3}$$

where $\sigma = \uparrow, \downarrow$, $c_\sigma^\dagger$ and $a_{p,\sigma}^\dagger$ are the fermion creation operators for an impurity electron with spin σ, and that for a bath electron with spin σ and momentum p, respectively. $n_\sigma = c_\sigma^\dagger c_\sigma$ is the fermion number operator. $\beta = 1/T$ is the inverse temperature. K is an arbitrary chosen parameter that controls the coupling strength of the auxiliary field and the average expansion order, which we will see below.

In the CT-AUX method, the density-density interaction in H_1 is decoupled by an auxiliary Ising field s as

$$H_1 = -\left(\frac{K}{2\beta}\right)\sum_{s=\pm1} e^{\gamma s(n_\uparrow - n_\downarrow)}. \tag{4}$$

γ is the coupling strength between the fermion density and the auxiliary field, and is determined by $\cosh(\gamma) \equiv 1 + (\beta U)/(2K)$. The partition function is expanded as

$$\frac{Z}{Z_0} = \text{Tr}\left[e^{-\beta H_0} T_\tau e^{-\int_0^\beta d\tau H_1(\tau)}\right] = \sum_{n=0}\int_0^\beta d\tau_1 \cdots \int_{\tau_{n-1}}^\beta d\tau_n \left(\frac{K}{2\beta}\right)^n \frac{Z_n(\{s_i,\tau_i\})}{Z_0}. \tag{5}$$

Here,

$$Z_n(\{s_i,\tau_i\})/Z_0 \equiv \prod_{\sigma=\uparrow,\downarrow} \det N_\sigma^{-1}(\{s_i,\tau_i\}),$$
$$N_\sigma^{-1}(\{s_i,\tau_i\}) \equiv e^{V_\sigma\{s_i\}} - G_{0\sigma}^{\{\tau_i\}}(e^{V_\sigma\{s_i\}} - 1), \tag{6}$$

where $Z_0 \equiv \text{Tr}\, e^{-\beta H_0}$, and $e^{V_\sigma\{s_i\}} \equiv \text{diag}(e^{\gamma(-1)^\sigma s_1}, \ldots, e^{\gamma(-1)^\sigma s_n})$ with the notations $(-1)^\uparrow \equiv 1$, $(-1)^\downarrow \equiv -1$, $(G_{0\sigma}^{\{\tau_i\}})_{ij} = g_\sigma(\tau_i - \tau_j)$ for $i \neq j$, and $(G_{0\sigma}^{\{\tau_i\}})_{ii} = g_\sigma(0^+)$. $g_\sigma(\tau) > 0$ is the free fermion Green's function at the impurity site. The configuration space for the MC sampling is hence the collection of all the possible auxiliary spin configurations on the imaginary-time interval and at all possible expansion orders $n = 0,1,...$, $c = \{\{\tau_1,s_1\}\cdots\{\tau_n,s_n\}\}$, where $0 \leqslant \tau_1 < \tau_2 < \cdots < \tau_n < \beta$ and $s_i = \uparrow, \downarrow$.

The corresponding weight w_c is given by Eq. (6). Then, a random walk $c_1 \to c_2 \to c_3 \to \cdots$ in configuration space is implemented usually by inserting/removing random spins at random imaginary times.

Self-learning CT-AUX. Here, we describe the self-learning continuous-time auxiliary-field method. As other SLMC methods, it consists of two parts: (1) Learn an effective model or DGF that approximates the probability distribution of auxiliary spins in the imaginary-time interval $\{\{\tau_1,s_1\}\cdots\{\tau_n,s_n\}\}$, and (2) propose a global move by executing a sequence of local updates in the effective model [10].

Since the number of auxiliary spins changes constantly with the expansion order n in the sampling process, one may expect that to reproduce the entire probability distribution at all orders requires a highly sophisticated model with a huge number of parameters. On the contrary, we introduce a DGF of a remarkably simple form which fits the probability distribution very accurately,

$$Z_n(\{s_i,\tau_i\})/Z_0 \simeq e^{-\beta H_n^{\text{eff}}(\{s_i,\tau_i\})}, \tag{7}$$

$$-\beta H_n^{\text{eff}}(\{s_i,\tau_i\}) \equiv \frac{1}{n}\sum_{i,j} J(\tau_i - \tau_j)s_i s_j + \frac{1}{n}\sum_{i,j} L(\tau_i - \tau_j) + f(n). \tag{8}$$

Several features of H_n^{eff} deserve attention:

(i) DGF serves as an approximation to Z_n in the weak-coupling expansion, as is indicated in Eq. (7), whose functional form could be obtained exactly if one could integrate out fermion degrees of freedom exactly. This is indeed what is done in the original CT-AUX algorithm. Here, in SLMC, the DGF is instead constructed by a series expansion and symmetry analysis. To be specific, Eq. (8) is the spin-spin interactions satisfying the spin-flip symmetry $s_i \to -s_i$ up to two-body terms. Since the performance of the DGF is already good enough at this stage, we did not include fourth-order interactions that are proportional to $s_i s_j s_k s_l$.

(ii) The interaction terms $J(\tau)$ and $L(\tau)$ are in principle allowed to be different functions of τ at different expansion orders n, which would result in vastly more parameters. Here, this predicament is avoided by choosing the same functions to all expansion orders.

(iii) The expansion-order-dependent factor $1/n$ in Eq. (7) is crucial. It can be justified by considering the atomic limit $V = 0$, where the interaction term $H_1(\tau) \equiv H_1$ in Eq. (5) becomes independent of τ_i, and hence $Z_n \propto \text{Tr}(H_1^n)$. For large n, $\text{Tr}(H_1^n) \simeq \epsilon_0^n$ is dominated by the contribution from the largest eigenvalue ϵ_0, hence $\ln Z_n/Z_0$ increases linearly with n. On the other hand, H_n^{eff} in Eq. (8) includes a summation of n^2 pairwise interactions at pairs of imaginary-time instances (τ_i,τ_j). Therefore, we must include the factor $1/n$ to match the two results.

As we will show later, this simple DGF performs remarkably well.

The training procedure goes as follows. Given a set of configurations $\{c_i\}$ taken from the Markov chain of a MC simulation, we minimize the mean square error $(\ln Z_n^{\text{eff}} - \ln Z_n/Z_0)$ on this training set by varying the functional form of $J(\tau)$, $L(\tau)$, and $f(n)$. In practice, we use Chebyshev polynomials $T_m(x) = \cos[m \arccos(x)]$ to expand functions J and L, $J(\tau) \equiv \sum_{m=0}^{m_{c,J}} a_m T_{2m}[x(|\tau|)]$ and $L(\tau) \equiv \sum_{m=0}^{m_{c,L}} b_m T_{2m}[x(|\tau|)]$ with $x(\tau) \equiv 2\tau/\beta - 1$ [18–22], and use a power series to expand the function f, $f(n) = \sum_{k=0}^{m_{c,f}} c_k n^k$. Here, $m_{c,J}$, $m_{c,L}$, and $m_{c,f}$ are the truncation orders for the respective functions. The rationale behind the choice of basis functions is that the Chebyshev polynomial is close to the

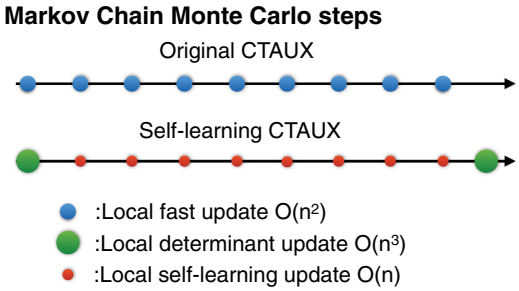

FIG. 1. Schematic figure for the Markov chains in the original and self-learning continuous-time Monte Carlo methods to obtain an uncorrelated configuration. n denotes the average expansion order that determines the size of the matrix $N_\sigma(\{s_i,\tau_i\})$, and further determines the complexity of the simulation. See the last section of this Rapid Communication for a detailed discussion.

minimax polynomial, minimizing the maximum error in the approximation. In other words, the Chebyshev polynomials approximate the original function uniformly [23]. In the simulation, we always increase the truncation order until the results converge. The total number of training parameters is thus $m_c \equiv m_{c,J} + m_{c,L} + m_{c,f} + 3$ (summation starts from 0) [24]. Since the DGF $H_n^{\rm eff}$ is a linear function of these parameters, they can be trained simply with a linear regression [25]. We have also exploited the iterative training procedure to improve the efficiency [9], whereby Monte Carlo configurations and weights generated by the self-learning algorithm are used as training data to further train the DGF. This procedure can be iterated until the DGF reproduces the exact probability distribution sufficiently well. We note that training the effective model can be regarded as supervised learning in a broader context of machine learning, which recently has many fruitful applications in physics [26–38].

After completing the training process, we use the trained DGF to propose highly efficient global moves on the Markov chain in actual simulations. Here, we adopt the general procedure of cumulative update introduced in Ref. [10]. Figure 1 illustrates how self-learning CT-AUX proposes global moves, in comparison with the original CT-AUX method. Starting from a configuration c_i, we perform a sufficiently large number (denoted by $M_{\rm eff}$) of local updates by inserting/removing random spins at random imaginary times based on the weights of the DGF, until reaching a configuration c_j that is sufficiently uncorrelated with c_i. The global move $c_i \to c_j$ is then proposed, and its acceptance rate p is calculated from the exact weight of the original model, $p = \min\{1,(w_{c_j} w_{c_i}^{\rm eff})/(w_{c_i} w_{c_j}^{\rm eff})\}$, where w_{c_i} and $w_{c_i}^{\rm eff}$ are weights of configuration c_i computed from the original model Eq. (5) and effective model Eq. (7), respectively. As shown previously [10], this cumulative update procedure fulfills the ergodicity condition and obeys the detailed balance principle. Since computing the weight of DGF is much faster than computing the fermion determinant in the original method, our method significantly reduces the computational cost of the simulation. A detailed discussion on the choice of the cumulative update length $M_{\rm eff}$ and the computational complexity of self-learning CT-AUX method is presented in the last section of this Rapid Communication.

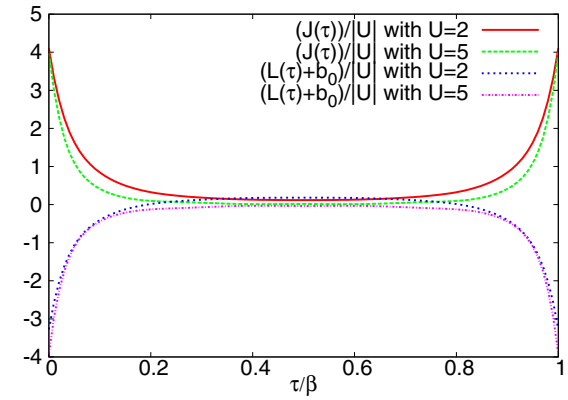

FIG. 2. Effective interactions for different U with $\beta = 10$, $V = 1$, and $K = 1$.

Performance on Anderson model. Now we are ready to show the performance of self-learning CT-AUX on the single impurity Anderson model. We consider a bath with a semicircular density of states $\rho_0(\epsilon) = [2/(\pi D)\sqrt{1-(\epsilon/D)^2}]$ and set the half bandwidth $D = 1$ as the energy unit. The chemical potential is set to be $\mu = U/2$ to maintain a half filling.

In the simulation, we use 5×10^4 configurations as the training data set. Throughout the parameter regime in our calculations, a total of 30 training parameters ($m_{c,J} = m_{c,L} = 12$, $m_{c,f} = 3$) is enough to guarantee the convergence of the DGF. After training, we obtain the interaction functions $J(\tau)$ and $L(\tau)$ in the DGF (8), as shown in Fig. 2. They become more localized at $\tau = 0$ and β with increasing U. To evaluate the accuracy of the DGF, we plot in Fig. 3 the distribution of the weights of the DGF and those of the original model exactly computed. The two distributions look very similar. To quantitatively measure the goodness of fit, we evaluate the quantity $R^2 \in [0,1]$ which is introduced as the "score" of the self-learning Monte Carlo method in general [10]. Here, we find the DGF for the Anderson impurity model (with $U = 5$, $\beta = 10$, $V = 1$, and $K = 1$) has a score of $R^2 = 99.9\%$.

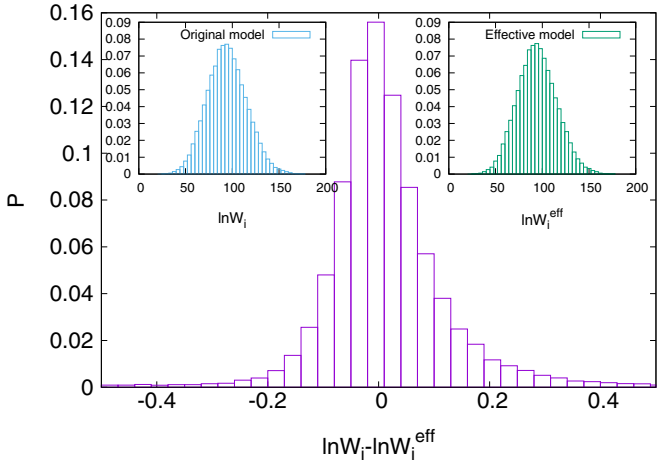

FIG. 3. For 5×10^4 independent configurations on the Markov chain of the original CT-AUX, the histogram is the distribution of the difference $\ln W_i - \ln W_i^{\rm eff}$. The upper-left and upper-right insets are distributions of $\ln W_i$ and $\ln W_i^{\rm eff}$, respectively. Here, $U = 5$, $V = 1$, $\beta = 10$, and $K = 1$.

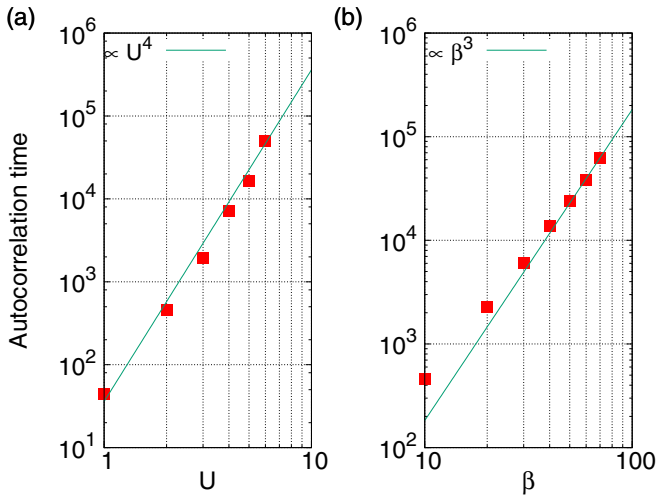

FIG. 4. Left: U dependence of the autocorrelation time of the original CT-AUX with $\beta = 10$. Right: β dependence of the autocorrelation time with $U = 2$. The other related parameters are $V = 1$ and $K = 1$. In both figures, the unit time is a local update (inserting/removing a auxiliary spin) in the original CT-AUX method.

Owing to the success of our DGF, a global move proposed by a cumulative update between two uncorrelated configurations has a very high average acceptance rate around 0.68.

To demonstrate the speedup of the self-learning CT-AUX method, we compute the autocorrelation function of the auxiliary spin polarization defined by $m \equiv (1/n)\sum_{i=1}^{n} s_i$. Figure 4 shows the autocorrelation time of the original CT-AUX method, defined in terms of the number of local updates. It is clear that the autocorrelation time increases rapidly with β and U, rendering the algorithm inefficient at low temperatures and in the strong-coupling regime. In contrast, the performance of the self-learning CT-AUX method is shown in Fig. 5. The autocorrelation function decays rapidly with the number of

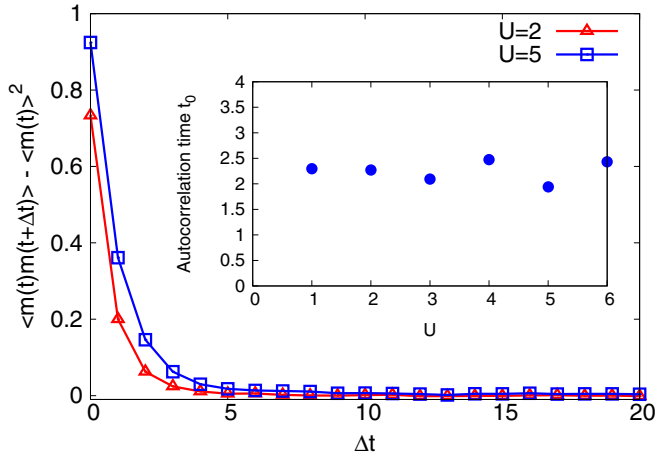

FIG. 5. Autocorrelation function of the auxiliary-spin magnetization for a system with $\beta = 10$, $V = 1$, and $K = 1$. The unit time is defined in the main text. Inset: U dependence of the autocorrelation time in the self-learning CT-AUX. We set the number of local updates on DGF to be $M_{\text{eff}} = 2 \times 10^3$ ($U = 1,2,3,4$) and $M_{\text{eff}} = 5 \times 10^4$ ($U = 5,6$).

global moves proposed by the DGF. This is because (1) a single global move is the cumulative outcome of M_{eff} local updates, where M_{eff} is taken to be so large that the proposed configuration is sufficiently uncorrelated from the current one, and (2) the average acceptance rates for such global moves are high enough—greater than 0.6 for all the data points in Fig. 5. The inset shows the U dependence of the autocorrelation time t_0, which is estimated from the initial slope of the autocorrelation function $\langle m(t)m(t+\Delta t)\rangle \sim e^{-\Delta t/t_0}$. It is worth noting that with increasing M_{eff}, the autocorrelation time of our self-learning algorithm saturates to a small value even for very large U.

Computational complexity. Finally, we discuss the actual calculation cost of the self-learning CT-AUX method. Figure 1 shows schematically the Markov chains to obtain two uncorrelated configurations. Roughly speaking, self-learning CT-AUX is faster than the original CT-AUX because the computational cost of each local move in the Markov chain is smaller than that in the CT-AUX. A detailed analysis is given as follows. In order to compare the two methods on an equal footing, we consider the cost to obtain an uncorrelated configuration from a given one. In this way, the two methods give the same error bar for the measured observables. The cost of inserting/removing a vertex with the use of fast updates is $O(\langle n\rangle^2)$ in the original CT-AUX simulation [8]. $\langle n\rangle$ is the average expansion order that determines the size of the matrix $N_\sigma(\{s_i,\tau_i\})$. To obtain an uncorrelated configuration, τ_{ori} such local updates are needed. (This is actually the definition of autocorrelation time in the original method.) Thus, the total cost is $O(\langle n\rangle^2\tau_{\text{ori}})$. On the other hand, the cost for inserting/removing a vertex is $O(m_c\langle n\rangle)$ in the effective model. Recall m_c is the number of the training parameters in the DGF. The scaling of $\langle n\rangle$ is different from that in the original CT-AUX because the weight of DGF is computed directly without calculating the fermion determinant. The number of local updates using DGF M_{eff} should be τ_{ori} in order to obtain an uncorrelated configuration. And we need one more calculation of the determinant to decide the weight of the proposed global move, whose computational cost is $O(\langle n\rangle^3)$. Note that the global move is not always accepted; there is additional τ_{SL} overhead, which is the autocorrelation time measured in Fig. 5. Thus the total calculation cost of the self-learning algorithm is $O[(\langle n\rangle^3 + m_c\langle n\rangle\tau_{\text{ori}})\tau_{\text{SL}}]$. Since $\langle n\rangle \sim \beta U$ [8] and the autocorrelation time τ_{ori} is approximately proportional to $U^4\beta^3$ as shown in Fig. 4, the second term in the parenthesis $m_c\langle n\rangle\tau_{\text{ori}}$ dominates. This is indeed the case shown in the inset in Fig. 5. In fact, in our computation $\langle n\rangle$ is less than 30 while the τ_{ori} can be up to of order 10^6. In this way, the actual speedup ratio t_s is expressed by

$$t_s \sim \frac{\langle n\rangle}{m_c\tau_{\text{SL}}}. \tag{9}$$

As long as the DGF is good enough, τ_{SL} is $O(1)$. Since m_c hardly scales with U and β, the self-learning CT-AUX method is generally faster than the original CT-AUX, especially in the low-temperature and strong-coupling regime when $\langle n\rangle \sim \beta U$ is large.

Conclusion. We developed the continuous-time version of the SLMC with an auxiliary field, which trains an effective model (DGF) to propose different uncorrelated configurations

in the Markov chain, with a high acceptance rate. The DGF for the Anderson impurity model is found to take a remarkably simple form, and reproduce very well the exact distribution of auxiliary fields in continuous time to all orders of the diagrammatic expansion. Our method reduces the computational complexity of the simulation in the low-temperature or strong-coupling regime, where the autocorrelation time in the standard method becomes large.

Our self-learning CT-AUX method has many potential applications. It can be used as an impurity solver for dynamical mean-field theory, and is ideal for studying systems near the critical point [39–41], where standard methods suffer from a severe critical slowing down. Our method can also be generalized straightforwardly to fermion lattice models.

The calculations were performed by the supercomputing system SGI ICE X at the Japan Atomic Energy Agency. The work at MIT was supported by DOE Office of Basic Energy Sciences, Division of Materials Sciences and Engineering under Award No. DE-SC0010526. Y.N. was supported by JSPS KAKENHI Grant No. 26800197, the "Topological Materials Science" (No. JP16H00995) KAKENHI on Innovative Areas from JSPS of Japan. H.S. is supported by MIT Alumni Fellowship Fund For Physics.

[1] R. Blankenbecler, D. J. Scalapino, and R. L. Sugar, Monte Carlo calculations of coupled boson-fermion systems. I, Phys. Rev. D **24**, 2278 (1981).

[2] J. E. Hirsch, Two-dimensional Hubbard model: Numerical simulation study, Phys. Rev. B **31**, 4403 (1985).

[3] J. E. Hirsch and R. M. Fye, Monte Carlo Method for Magnetic Impurities in Metals, Phys. Rev. Lett. **56**, 2521 (1986).

[4] S. R. White, D. J. Scalapino, R. L. Sugar, E. Y. Loh, J. E. Gubernatis, and R. T. Scalettar, Numerical study of the two-dimensional Hubbard model, Phys. Rev. B **40**, 506 (1989).

[5] A. N. Rubtsov, V. V. Savkin, and A. I. Lichtenstein, Continuous-time quantum Monte Carlo method for fermions, Phys. Rev. B **72**, 035122 (2005).

[6] P. Werner, A. Comanac, L. de Medici, M. Troyer, and A. J. Millis, Continuous-Time Solver for Quantum Impurity Models, Phys. Rev. Lett. **97**, 076405 (2006).

[7] E. Gull, A. J. Millis, A. I. Lichtenstein, A. N. Rubtsov, M. Troyer, and P. Werner, Continuous-time Monte Carlo methods for quantum impurity models, Rev. Mod. Phys. **83**, 349 (2011).

[8] E. Gull, P. Werner, O. Parcollet and M. Troyer, Continuous-time auxiliary-field Monte Carlo for quantum impurity models, Europhys. Lett. **82**, 57003 (2008).

[9] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Self-learning Monte Carlo method, Phys. Rev. B **95**, 041101(R) (2017).

[10] J. Liu, H. Shen, Y. Qi, Z. Y. Meng, and L. Fu, Self-learning Monte Carlo method in fermion systems, Phys. Rev. B **95**, 241104(R) (2017).

[11] X. Y. Xu, Y. Qi, J. Liu, L. Fu, and Z. Y. Meng, Self-Learning determinantal quantum Monte Carlo method, Phys. Rev. B **96**, 041119(R) (2017).

[12] P. W. Anderson, Localized Magnetic States in Metals, Phys. Rev. **124**, 41 (1961).

[13] P. Werner, E. Gull, O. Parcollet, and A. J. Millis, Momentum-selective metal-insulator transition in the two-dimensional Hubbard model: An 8-site dynamical cluster approximation study, Phys. Rev. B **80**, 045120 (2009).

[14] E. Gull, P. Staar, S. Fuchs, P. Nukala, M. S. Summers, T. Pruschke, T. C. Schulthess, and T. Maier, Submatrix updates for the continuous-time auxiliary-field algorithm, Phys. Rev. B **83**, 075122 (2011).

[15] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevB.96.161102 for the technical details of the self-learning method.

[16] L. Huang, Y.-F. Yang, and L. Wang, Recommender engine for continuous time quantum Monte Carlo methods, Phys. Rev. E **95**, 031301(R) (2017).

[17] L. Huang and L. Wang, Accelerated Monte Carlo simulations with restricted Boltzmann machines, Phys. Rev. B **95**, 035105 (2017).

[18] F. A. Wolf, I. P. McCulloch, O. Parcollet, and U. Schollwöck, Chebyshev matrix product state impurity solver for dynamical mean-field theory, Phys. Rev. B **90**, 115124 (2014).

[19] A. Braun and P. Schmitteckert, Numerical evaluation of Green's functions based on the Chebyshev expansion, Phys. Rev. B **90**, 165112 (2014).

[20] L. Covaci, F. M. Peeters, and M. Berciu, Efficient Numerical Approach to Inhomogeneous Superconductivity: The Chebyshev–Bogoliubov–de Gennes Method, Phys. Rev. Lett. **105**, 167006 (2010).

[21] Y. Nagai, Y. Ota, and M. Machida, Efficient numerical self-consistent mean-field approach for fermionic many-body systems by polynomial expansion on spectral density, J. Phys. Soc. Jpn. **81**, 024710 (2012).

[22] S. Sota and T. Tohyama, Low-temperature density matrix renormalization group using regulated polynomial expansion, Phys. Rev. B **78**, 113101 (2008).

[23] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, in *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd ed. (Cambridge University Press, Cambridge, U.K., 1992), Secs. 5.8, 5.9, and 5.10, pp. 184–188, 189–190, and 191–192.

[24] By convergence we mean the mean squared error in the linear regression, or the average acceptance rate/autocorrelation time of the cumulative update does not change significantly with m_c. The typical values of m_c could be roughly estimated by the "steepness" of functions $J(\tau)$ and $L(\tau)$ near $\tau = 0$ (Fig. 2). As will be pointed out later, these functions will become more localized (steeper) at $\tau = 0$ when the temperature is low or when U is large, in which case more high-order Chebyshev polynomials are needed to fit the interaction well.

[25] Here, we briefly comment on the numerical stability in the linear regression. The linear regression suffers from instability problems either when there is inherent collinearity between the samples or the sample size is very small. Usually in SLMC, the training data could be made sufficiently uncorrelated and could be generated as much as we want by doing a MC simulation. Therefore, the instability problem or overfitting is not likely to happen. However, in the scenario when generating training data is costly, this may become an issue. In this case, one may exploit techniques such as ridge regression to avoid overfitting.

[26] L. Zdeborova, Machine learning: New tool in the box, Nat. Phys. **13**, 420 (2017).

[27] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, Nat. Phys. **13**, 431 (2017).

[28] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, 602 (2017).

[29] A. Tanaka and A. Tomiya, Detection of phase transition via convolutional neural networks, J. Phys. Soc. Jpn. **86**, 063001 (2017).

[30] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, Nat. Phys. **13**, 435 (2017).

[31] Y. Zhang and E.-A. Kim, Quantum Loop Topography for Machine Learning, Phys. Rev. Lett. **118**, 216401 (2017).

[32] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang, On the equivalence of restricted Boltzmann machines and tensor network states, arXiv:1701.04831.

[33] D.-L. Deng, X. Li, and S. Das Sarma, Quantum Entanglement in Neural Network States, Phys. Rev. X **7**, 021021 (2017).

[34] Y. Huang and J. E. Moore, Neural network representation of tensor network and chiral states, arXiv:1701.06246.

[35] W. Hu, R. R. P. Singh, and R. T. Scalettar, Discovering phases, phase transitions, and crossovers through unsupervised machine learning: A critical examination, Phys. Rev. E **95**, 062122 (2017).

[36] Z. Cai, Approximating quantum many-body wave functions using artificial neural networks, arXiv:1704.05148.

[37] H. Fujita, Y. O. Nakagawa, S. Sugiura, and M. Oshikawa, Construction of Hamiltonians by machine learning of energy and entanglement spectra, arXiv:1705.05372.

[38] S. J. Wetzel and M. Scherzer, Machine learning of explicit order parameters: From the Ising model to SU(2) lattice gauge theory, arXiv:1705.05582.

[39] Y. Nagai, S. Hoshino, and Y. Ota, Critical temperature enhancement of topological superconductors: A dynamical mean-field study, Phys. Rev. B **93**, 220505(R) (2016).

[40] S. Hoshino and P. Werner, Superconductivity from Emerging Magnetic Moments, Phys. Rev. Lett. **115**, 247001 (2015).

[41] S. Hoshino and P. Werner, Electronic orders in multiorbital Hubbard models with lifted orbital degeneracy, Phys. Rev. B **93**, 155161 (2016).