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

Yuki Nagai,[1,2,*] Akinori Tanaka,[3,1,4,†] and Akio Tomiya[5,‡]

[1] *Mathematical Science Team, RIKEN Center for Advanced Intelligence Project (AIP),*
1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan
[2] *CCSE, Japan Atomic Energy Agency, 178-4-4, Wakashiba, Kashiwa, Chiba 277-0871, Japan*
[3] *Interdisciplinary Theoretical and Mathematical Sciences Program (iTHEMS) RIKEN 2-1,*
Hirosawa, Wako, Saitama 351-0198, Japan
[4] *Department of Mathematics, Faculty of Science and Technology, Keio University,*
3-14-1 Hiyoshi, Kouhoku-ku, Yokohama 223-8522, Japan
[5] *RIKEN/BNL Research center, Brookhaven National Laboratory, Upton, New York 11973, USA*

(Received 28 January 2021; accepted 23 November 2022; published 8 March 2023)

In this paper, we develop the self-learning Monte-Carlo (SLMC) algorithm for non-Abelian gauge theory with dynamical fermions in four dimensions to resolve the autocorrelation problem in lattice QCD. We perform simulations with the dynamical staggered fermions and plaquette gauge action by both in the hybrid Monte-Carlo (HMC) and SLMC for zero and finite temperature to examine the validity of SLMC. We confirm that SLMC can reduce autocorrelation time in non-Abelian gauge theory and reproduce results from HMC. For finite temperature runs, we confirm that SLMC reproduces correct results with HMC, including higher-order moments of the Polyakov loop and the chiral condensate. Besides, our finite temperature calculations indicate that four flavor QC$_2$D with $\hat{m} = 0.5$ is likely in the crossover regime in the Colombia plot.

DOI: 10.1103/PhysRevD.107.054501

I. INTRODUCTION

For more than 40 years, numerical calculations of lattice QCD have been an established technique to calculate QCD observables including nonperturbative effects [1]. Lattice QCD is based on the Markov chain Monte-Carlo (MCMC) algorithms with the detailed balance condition, which guarantees the MCMC process's convergence, and it is crucial for dealing with the dimensionality of the path integral for lattice QCD. From a practical point of view, convergence is necessary because it allows for uses of the computational result in precise phenomenology. (See [2,3] and references therein, for example).

A MCMC algorithm for lattice QCD is required following three conditions. The first condition is convergence of the Markov chain update to the equilibrium distribution. Typically, one requires the detailed balance condition because it is one of sufficient conditions for the convergence.[1] The second condition is applicability for a non-Abelian gauge system with dynamical fermions because QCD consists of gluons and quarks, and we must guarantee the gauge invariance. The third condition is no-bias in calculations. For example, the hybrid algorithm or R algorithm can deal with dynamical fermions, but it has a bias [4], and it is not favored. Another example is the molecular dynamics, which can generate configurations for lattice QCD by itself, but its ergodicity is not guaranteed because it only sweeps one energy constant surface. Moreover, it has bias from the finite step size.

De facto standard algorithms for gauge theories with dynamical fermions are the hybrid Monte-Carlo (HMC) [5] and its variant, rational hybrid Monte-Carlo (RHMC) [6] because they satisfy the three conditions above. The basic idea of (R)HMC is based on the Metropolis algorithm with fictitious hamiltonian dynamics, including Gaussian momentum, along with fictitious time. Generally speaking, a Metropolis algorithm shows high acceptance if the theory's energy function does not change so much during the update process.[2] (R)HMC uses the molecular dynamics with a reversible symplectic integrator, and it preserves the

[*] nagai.yuki@jaea.go.jp
[†] akinori.tanaka@riken.jp
[‡] akio.tomiya@riken.jp

[1] To guarantee it, we need additionally and ergodicity, which is explained later on.

[2] From a viewpoint of efficiency of mixing, it depends on how far-reaching the proposals are.

energy function mostly and acceptance rate is controlled by the violation of energy. The Gaussian momentum guarantees ergodicity of the algorithm. In addition to this, update process allows us to include fermions through the pseudofermion trick. Despite these advantages, (R)HMC suffers from unavoidable critical slowing down problem with several parameter regimes [7–9], and thus to resolve the problem, a new algorithm is demanded.

On the other hand, recent progress of machine (ML) learning drives application it to lattice field theory [10–30]. There are three typical usages of machine learning in this context. The first is making a detector of phase boundaries [11,12,31]. The second is calculation for observable to reduce numerical cost [13–16]. The third is configuration generation for field theory [17–22]. These applications are inspired by similarities between configuration generation and image generation but except for flow-based algorithm [20–22], they do not enjoy the convergence theorem. This means that one has to compare observables one by one with observables generated by a "legal algorithm" like HMC.

Self-learning Monte-Carlo(SLMC) is a machine-learning based configuration generation algorithm [32–38]. Originally it has been developed for a classical spin model [32] but it works widely among quantum model in condensed matter physics and quantum chemistry [33,36]. SLMC employs updates using a tunable action with an efficient update algorithm and the Metropolis—Hastings test, and then, it becomes an exact algorithm. One of the advantages of SLMC is interpretability; we can use our insight of the original theory to construct the tunable action. Besides, one can also use neural networks to evaluate the tunable action in the algorithm [34,36] by giving up interpretability in order to achieve better acceptance. We develop SLMC for non-Abelian gauge field with dynamical fermions.

From the 1990s, systems with dynamical fermions using hopping parameter expanded action have been investigated [39–44]. They used truncated determinant to perform simulations for the Schwinger model and zero temperature QCD. Here we clarify the difference between the present work and these studies. First, we perform an extensive study for volume dependence and action dependence, and we find that the Polyakov loop in the effective action improves acceptance, for example, see Fig. 9 or Fig. 10. Second, we (re)formulated the algorithm in the context of ML application. It clarifies the mathematical meanings and possible extension of the algorithm. Third, we confirm our algorithm reduce autocorrelation for a finite temperature system. As an similar idea, the multiboson algorithm [45,46] is known. It uses the Metropolis algorithm with polynomially expanded fermion action. But we do not use pseudofermion field to update the gauge field in the present work. We leave that idea for future study.

In this work, we perform simulations with two-color QCD with dynamical fermions in four dimensions (QC$_2$D) using HMC and SLMC. Besides, we apply our algorithm to investigate QC$_2$D with 4 flavors phase diagram associated the Polyakov loop in heavy mass regime at finite temperature.[3] In that regime, HMC is suffered from long autocorrelation problem of the Polyakov loop. We find that, both in zero and finite temperature, SLMC has smaller autocorrelation time than HMC and gives consistent results with correct cumulants.

This paper is organized as follows. In Sec. II, we review self-learning Monte-Carlo from Metropolis—Hasting algorithm. In Sec. II, we explain target system and effective action in our calculations. In Sec. IV, we introduce our results at zero temperature runs. In Sec. V, we show results for an application of SLMC to a finite temperature system. In Sec. VI, we summarize our results.

II. SELF-LEARNING MONTE-CARLO

Self-learning Monte-Carlo (SLMC) algorithm is a general purposed sampling scheme based on Metropolis–Hastings algorithm. It consists of two phases: (a) Training phase and (b) Application phase (Fig. 1), and (c) hybrid.

First of all, let us emphasize here that SLMC is a sampling algorithm for a fixed probability density. In other words, it is a sampling method for a target lattice action S with fixed lattice size, gauge coupling, fermion contents, fermion mass in lattice gauge theory. In the latter part of the present paper, we compare results of various SLMC runs for S defined in (7), four-dimensional $SU(2)$ plaquette and log det term from 4 tastes standard staggered fermions, under different lattice sizes, fermion masses, gauge couplings.

A. Training phase

In the training phase [Fig. 1(a)], we prepare multiple samples $\{U_i\}_{i=1,2,\ldots,N}$ from the probability density $e^{-S[U]}/Z$, a fitting model $S^{\theta}_{\mathrm{eff}}$ with a simple update probability P_θ. In our case, each U_i is a pair of gauge configurations, and $S^{\theta}_{\mathrm{eff}}$ is a sum of certain loop operators with tunable coupling constants defined in (9), and P_θ is heatbath update.

After sampling from the target density, the fitting parameters θ are determined by minimizing a loss function, which we choose averaged least squared loss L_2 between the target action value $S[U_i]$ and the model action value $S^{\theta}_{\mathrm{eff}}[U_i]$ over all (finite) samples:

$$L_2 = \frac{1}{N}\sum_{i=1}^{N}(S[U_i] - S^{\theta}_{\mathrm{eff}}[U_i])^2. \tag{1}$$

[3] The one-loop beta function for $\mathrm{SU}(N_c = 2)$ gauge theory with $n_f = 4$ fundamental matters is $\beta(g) = -\frac{g^3}{(4\pi)^2}(\frac{11}{3}N_c - \frac{2}{3}n_f) = -\frac{g^3}{(4\pi)^2}4.6 < 0$, thus, this theory is asymptotic free. In addition, this theory is not infrared conformal [47]. Thus, QC$_2$D with $n_f = 4$ has qualitatively same nature to conventional QCD.

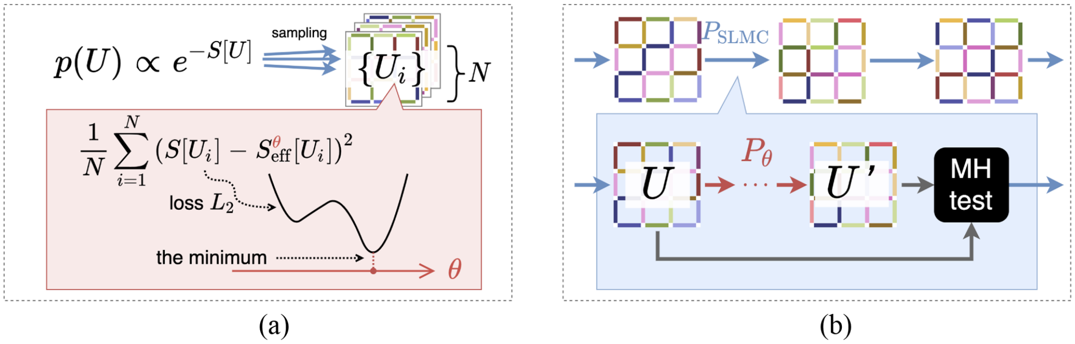


FIG. 1. Two phases of SLMC. We apply this method under 4D links with fermion contribution. (a) Training phase of SLMC and (b) application phase of SLMC.

One can choose other loss functions: KL divergence, or squared loss with certain regularization terms, etc. The choice of loss function and parameters θ only affect the acceptance probability of the Metropolis–Hastings test in Fig. 1(b), that we explain next.

B. Application phase

In the application phase [Fig. 1(b)], we run the Metropolis–Hastings algorithm with the fitted effective parameters θ. More precisely, we use the θ dependent update $P_\theta(U'|U)$ to generate proposal sample U' for given configuration U, that satisfies the detailed balance condition with the fitted model $S_{\rm eff}$. But our purpose is sampling from the target probability $e^{-S[U']}/Z$, so after generating the proposal sample U', we conduct the Metropolis–Hastings test with acceptance probability

$$\min\left(1, \frac{e^{-(S[U']-S^{\theta}_{\rm eff}[U'])}}{e^{-(S[U]-S^{\theta}_{\rm eff}[U])}}\right), \tag{2}$$

then the Markov-chain satisfies the detailed balance condition with S. If the readers are not familiar with this algorithm, please see Appendix A.

C. Hybrid: Self-learning phase

The Metropolis–Hastings update algorithm satisfies the detailed balance condition, and one can get the samples from the target probability based on the corresponding Markov chain. We can utilize these samples to improve parameters θ further by repeating the training phase. In the later experiments, we apply this scheme, i.e., we begin with initial parameters, as explained in simulation setup, and keep adjusting parameters θ by minimizing (1) in each Metropolis–Hastings update.[4]

III. NUMERICAL SETUP

In this section, we introduce our numerical setup for simulation of lattice gauge theory. Throughout this paper, we show every quantity in lattice unit. We perform simulations with $SU(2)$ plaquette action with four tastes standard staggered fermions for implementational simplicity. Our target system is described by the action $S[U,\bar{\chi},\chi]$,

$$S[U,\bar{\chi},\chi] = S_g[U] + \sum_n \bar{\chi}(n)(M[U]\chi)(n), \tag{3}$$

where $S_g[U]$ is the Wilson plaquette action,

$$S_g[U] = \beta\sum_n\sum_{\mu=1}^{4}\sum_{\nu>\mu}\left(1-\frac{1}{2}{\rm tr}U_{\mu\nu}(n)\right), \tag{4}$$

$$U_{\mu\nu}(n) = U_\mu(n)U_\nu(n+\hat{\mu})U^\dagger_\mu(n+\hat{\nu})U^\dagger_\nu(n), \tag{5}$$

and $M[U]$ is a massive staggered Dirac operator,

$$M[U]\chi(n) = \frac{1}{2}\sum_{\mu=1}^{4}\eta_\mu(n)[U_\mu(n)\chi(n+\hat{\mu}) - U^\dagger_\mu(n-\hat{\mu})\chi(n-\hat{\mu})] + \hat{m}\chi(n), \tag{6}$$

and $\eta_\mu(n)$ is the staggered factor. $U_\mu(n)$ is a link variable for $SU(2)$ gauge field, and $\chi(n)$ is a single component spinor field. $\hat{m}$ is a quark mass in the lattice unit. Summation of n indicates that summation over four dimensional coordinate: $n=(n_1,n_2,n_3,n_4)$. Here $\eta_\mu(n)$ is the staggered factor,

$$\eta_\mu(n) = \begin{cases} 1 & (\mu=1), \\ (-1)^{n_1} & (\mu=2), \\ (-1)^{n_1+n_2} & (\mu=3), \\ (-1)^{n_1+n_2+n_3} & (\mu=4), \end{cases}$$

which enables us to reconstruct the Dirac spinor structure in the continuum limit.

[4] The parameters' change gradually "thermalized." So, we can regard the later part of the Markov chain as usual chain by MH–test, that would guarantee exactness. We leave showing its exactness as a future work.

All numerical calculations for HMC and SLMC are performed by Julia [48] and the code is developed by ourselves. In addition, we use heatbath method, which is reviewed in Appendix B, to implement the proposal probability P_θ in the equation Fig. 1(b) and we implement automatic generation of heatbath code which generates staples from given loop operators [49]. This enables us to investigate effective action with various types of extended loops. However, in the present work, we employ the effective action (9) and slightly generalized effective actions with other loops for simplicity, and leave the study with complicated loops as a future work.

A. SLMC setup

1. Target action

First, we reformulate our target action in purely bosonic language which is given by

$$S[U] = S_g[U] + S_f[U], \tag{7}$$

where $S_g[U]$ is the plaquette gauge action and,

$$S_f[U] = -\log\ \det M^\dagger M, \tag{8}$$

is fermion contribution. The Dirac operator M is defined in (6). We use exact diagonalization to evaluate the Dirac operator in this paper for simplicity and this can be improved by using stochastic estimator in the reweighting technique [50,51].

2. Fitting model (effective action)

In practice, we employ heavy mass expanded fermion action with truncation [52] for the fitting model S^θ_{eff} in SLMC. Our effective action consists of the plaquette, rectangular loops and the Polyakov loop for $\mu = 1$, 2, 3, 4 directions. Let us denote $\vec{n} = (n_1, n_2, n_3)$ is spatial coordinate and n_4 is the temporal direction, and let lattice size be N_1, N_2, N_3 and N_4 for spatial and temporal extent, respectively. The effective action is,

$$\begin{aligned} S^\theta_{\mathrm{eff}}[U] = \sum_n \left[\beta_{\mathrm{plaq}} \sum_{\mu=1}^{4} \sum_{\nu>\mu} \left(1 - \frac{1}{2}\mathrm{tr} U_{\mu\nu}(n)\right) + \beta_{\mathrm{rect}} \sum_{\mu=1}^{4} \sum_{\nu\neq\mu} \left(1 - \frac{1}{2}\mathrm{tr} R_{\mu\nu}(n)\right)\right] \\ + \beta^{\mu=1}_{\mathrm{Pol}} \sum_{n_2,n_3,n_4} \mathrm{tr}\left[\prod_{n_1=0}^{N_1-1} U_1(\vec{n}, n_4)\right] + \beta^{\mu=2}_{\mathrm{Pol}} \sum_{n_1,n_3,n_4} \mathrm{tr}\left[\prod_{n_2=0}^{N_2-1} U_2(\vec{n}, n_4)\right] \\ + \beta^{\mu=3}_{\mathrm{Pol}} \sum_{n_1,n_2,n_4} \mathrm{tr}\left[\prod_{n_3=0}^{N_3-1} U_3(\vec{n}, n_4)\right] + \beta^{\mu=4}_{\mathrm{Pol}} \sum_{n_1,n_2,n_3} \mathrm{tr}\left[\prod_{n_4=0}^{N_4-1} U_4(\vec{n}, n_4)\right] + \beta_{\mathrm{const}}, \end{aligned} \tag{9}$$

where $n = (\vec{n}, n_4)$, $R_{\mu\nu}(n)$ is a rectangular Wilson loop,

$$\begin{aligned} R_{\mu\nu}(n) = U_\mu(n) U_\mu(n+\hat{\mu}) U_\nu(n+2\hat{\mu}) U^\dagger_\mu(n+\hat{\mu}+\hat{\nu}) \\ \times U^\dagger_\mu(n+\hat{\nu}) U^\dagger_\nu(n), \end{aligned} \tag{10}$$

and $\theta = \{\beta_{\mathrm{plaq}}, \beta_{\mathrm{rect}}, \beta^{\mu=1}_{\mathrm{Pol}}, \beta^{\mu=2}_{\mathrm{Pol}}, \beta^{\mu=3}_{\mathrm{Pol}}, \beta^{\mu=4}_{\mathrm{Pol}}, \beta_{\mathrm{const}}\}$ are determined by a linear regression with prior HMC run.[5] In general, one can include additional extended loops to improve the acceptance rate. This algorithm is based on interpretable ML. Namely, coefficients in the effective action have physical meaning. Note that SLMC gives an exact Markov chain procedure regardless of which terms are included. Even if the action is far from the target action, SLMC does not have bias but the acceptance rate approaches to zero.

We choose $P^\theta(U'|U)$ as the heatbath algorithm with whole extended loops in $S^\theta_{\mathrm{eff}}[U]$. Our strategy in the present work to overcome the critical slowing down is increasing the number of heatbath updates or overrelaxation in the SLMC update process. The critical slowing down depends on both of the update algorithm and the criticality of the system, and criticality is unavoidable. We overcome the critical slowing down by somewhat brute force way; repeating cheap updates.[6]

The acceptance rate can be estimated *a priori* by a loss of the regression[7]:

$$\text{Acceptance rate} \sim \exp(-\sqrt{L_2}), \tag{11}$$

[5] Prior HMC runs are not mandatory. One can improve the parameters within the SLMC runs as we commented in Sec. II C. Most of our simulations in this paper are done in this way. If the effective action is far from the target system, a prior HMC run is necessary to avoid inefficiency.

[6] This strategy is the same spirit of the all mode averaging (AMA) [53]. AMA reduces statistical error using many "sloppy" (cheap) calculation, and bias from sloppiness corrected by a few costly bias correction term.

[7] This estimation assumes the unbiased condition $0 = \langle S[U] - S^\theta_{\mathrm{eff}}[U]\rangle_S$ where the expectation is taken by the target probability $e^{-S[U]}/Z$, see Appendix I of [34] for more details.

where L_2 is the loss of the regression (1) for the effective action. It is similar to Karsch formula [39,54]. Namely, the acceptance rate can be reduced controlled by adding more and more extended loops as improving the linear regression.

3. How to compare HMC and SLMC

Generally, it is not fair to compare two different algorithms in terms of the elapsed time since the elapsed time depends on the architecture of machines and technical details of implementations. Here, we enumerate the number of most costly expensive parts in each algorithm. In HMC case, taking inversion of the Dirac operator is the expensive part. It occurs each molecular dynamics step and the Metropolis step, and the conjugation gradient (CG) method is usually used, which has many matrix-vector operations. On the other hand, in our SLMC case, fermion determinant calculation for the target action is the expensive part, which occurs each Metropolis step. As we mentioned before, this can be replaced by a stochastic estimator and it reduces numerical cost, but it still has the highest cost. A stochastic estimator includes the calculation of the inversion of the Dirac operator, which is similar to the CG method. In this work, we count the number of Metropolis test both in HMC and SLMC for simplicity. This comparison is not fair for SLMC, but still, it gives better results than HMC in terms of the autocorrelation.[8]

B. Observables

In the present work, we measure plaquette, rectangular Wilson loop, and the Polyakov loop to check the consistency of the algorithm. In our finite temperature application, we calculate a susceptibility (second-order cumulant) and the Binder cumulant, which is a fourth order moment, for the plaquette, rectangular Wilson loop, Polyakov loop, and chiral condensates as a function of β to check the consistency of the algorithm for possible biases in higher moments in addition to the mean values.

1. Polyakov loop

The Polyakov loop along with the temporal direction is a good indicator of the confinement-deconfinement transition,

$$\langle L\rangle = \frac{1}{N_\sigma^3}\left\langle \sum_{\vec{n}} \mathrm{Tr} \prod_{n_4} U_4(n_4, \vec{n})\right\rangle, \tag{12}$$

where we take $N_1 = N_2 = N_3 = N_\sigma$ as the spatial size of the lattice. The Polyakov loop susceptibility is,

$$\chi_L = \langle L^2\rangle - \langle L\rangle^2. \tag{13}$$

We also analyze the Binder cumulant B_L^4 [55], which is defined by,

$$B_L^4(\beta) = \frac{\langle(\delta L)^4\rangle}{\langle(\delta L)^2\rangle^2}, \tag{14}$$

where $\delta L = L - \langle L\rangle$. Binder cumulant is an indicator of the order of phase transition. If it takes $B^4 = 3.0$, that point does not have any singularity (crossover). If it takes $1 < B^4 < 3.0$, that point is the second-order phase transition, and the value is related to the universality class. If it takes $B^4 = 1.0$, that point is the first-order phase transition.

2. Chiral condensate

In addition to the Polyakov loop, we calculate four tastes chiral condensate,

$$\langle\bar{\psi}\psi\rangle = \frac{1}{N_\sigma^3 N_\tau}\left\langle \mathrm{Tr}\frac{1}{D+m}\right\rangle, \tag{15}$$

where $N_4 = N_\tau$, Tr indicates trace over all index in the Dirac operator, and its higher order moments as well as for the Polyakov loop.

3. Autocorrelation

Autocorrelation time is a measure of correlations between configurations, which quantifies the inefficiency of an MCMC algorithm. The decay of the autocorrelation function gives autocorrelation time, but the autocorrelation function itself is a statistical object, so we cannot determine the autocorrelation exactly. Instead, we calculate the approximated autocorrelation function [56,57] defined by,

$$\bar{\Gamma}(\tau) = \frac{1}{N_{\mathrm{trj}} - \tau}\sum_{c}^{N_{\mathrm{trj}}}(O_c - \bar{O})(O_{c+\tau} - \bar{O}), \tag{16}$$

where $O_c = O[U^{(c)}]$ is the value of operator O for the cth configuration $U^{(c)}$ and τ is fictitious time of HMC. N_{trj} is the number of trajectories. Conventionally, the normalized autocorrelation function $\bar{\rho}(\tau) = \bar{\Gamma}(\tau)/\bar{\Gamma}(0)$ is used.

The integrated autocorrelation time τ_{int} approximately quantifies effects of autocorrelation. This is given by,

$$\tau_{\mathrm{int}} = \frac{1}{2} + \sum_{\tau=1}^{W}\bar{\rho}(\tau). \tag{17}$$

We regard two configurations separated by $2\tau_{\mathrm{int}}$ as independent ones. In practice, we determine a window size W as a first point $W = \tau$ where $\bar{\Gamma}(\tau) < 0$ for the smallest τ. The statistical error of integrated autocorrelation time is estimated by the Madras–Sokal formula [57,58],

[8] It should depend on the scaling of the autocorrelation time to volume too. We left a more rigorous comparison for future work.

$$\langle \delta\tau_{\rm int}^2 \rangle \simeq \frac{4W+2}{N_{\rm trj}} \tau_{\rm int}^2. \qquad (18)$$

We use the square root of (18) to estimate the error on the autocorrelation time. It is obvious that the autocorrelation is observable dependent, and we focus on the autocorrelation from the Polyakov loop since we perform simulations with rather coarse lattices.[9]

IV. ALGORITHM ANALYSIS

A. Simulation setup

We compare results from SLMC and HMC with a baseline parameter set: $N_\sigma^3 \times N_\tau = 6^3 \times 6$, $\hat{m} = 0.5$, $\beta = 2.5$. The fictitious time for the leapfrog integration is taken to $\tau = 1$. Our effective action contains plaquette, rectangular, Polyakov loops for every direction as we explained in (9). We discard $O(100)$ trajectory from the analysis for thermalization, and the number of analyzed configurations[10] is $O(1000) - O(10000)$. We initialized $\beta_{\rm plaq} = \beta$, the coupling in the gauge action in target theory, and $\beta_{\rm others}$ are sampled as small random variables.

We perform several variations of SLMC runs by following reasons. The first purpose is, to confirm that results are independent from choice of effective actions. In addition, the statistical property with effective actions have to be investigated. The second purpose is, to see change of acceptance rate. Our effective action is inspired from the hopping parameter expansion, so expect that the acceptance is increased if we add higher order loops.

We estimate statistical error using the jackknife method. The binning size of the jackknife method is taken to be the number of jackknife samples 10 in the histogram, and others are taken to be larger than the autocorrelation time for each observable.

B. Simulation results

1. HMC vs SLMC

Before detail comparison, we compare HMC and SLMC for $\beta = 2.5$, $N_\sigma^3 \times N_\tau = 6^4$, $\hat{m} = 0.5$ (Fig. 2). The horizontal axis (MC time) is counted as the number of the Metropolis test as we explained above. The integrated autocorrelation time for HMC and SLMC are $\tau^{\rm HMC} = 62(24)$ and $\tau^{\rm SLMC} = 4.0(4)$, respectively. We choose the number of overrelaxation as 10 and the number of heatbath as 100 in this case.

We show results from HMC and SLMC for plaquette, rectangular Wilson loop, and Polyakov loop in histogram with statistical error in Fig. 3. One can see that all quantities from SLMC are consistent with ones from HMC. The shape of the histogram for the Polyakov loop indicates the effects of dynamical fermions because if it is quenched, the Polyakov loop is symmetric under $\mathbb{Z}_2$ reflection if the statistics are large enough.[11]

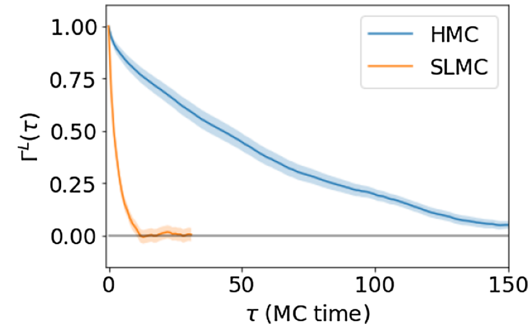


FIG. 2. Autocorrelation function for the Polyakov loop for $\beta = 2.5$, $N_\sigma^3 \times N_\tau = 6^4$, $\hat{m} = 0.5$. We measure the Monte-Carlo time τ as a unit of the Metropolis test. The bands indicate one sigma error bar. Please see main text for meaning of the horizontal axis.

2. Extensive study

Other quantities and setups are summarized the results in Table I. ID 0–7 in the table are results from HMC and various SLMC with $N_\sigma^3 \times N_\tau = 6^4$, $\beta = 2.5$, and $\hat{m} = 0.5$. Each SLMC entry means:

(i) SLMC_nup100: 100 times heatbath update is used,
(ii) SLMC_all: effective action including 3 × 1-rectangular, chair, and crown operators is used,
(iii) SLMCnor01 and SLMCnor20: 1 and 20 time overrelaxations after the heatbath update are used, respectively,
(iv) SLMCplq: effective action, which includes only plaquette term, is used,
(v) SLMCplqrct: effective action, which includes plaquette and the rectangular term is used.

All of the results from various SLMC are consistent with ones from HMC, except for ID6.[12] Besides, SLMCplq contains only one term, but it achieves roughly 60% acceptance and with consistent results. Comparing to the acceptance rate for SLMCplqrct and SLMC_all, improvement of effective action, namely adding more loop operators, the data shows that adding loops improves the acceptance rate.

[9] We checked the topological charge and its autocorrelation, but that is not relevant for our lattice spacing and system size.

[10] All of our measurements are done on the fly and performed every trajectory.

[11] More quantitative signal to distinguish quenched and unquenched runs is mass splitting of pseudoscalar and vector channel. However, we leave it for future study.

[12] Here we comment on the weird statistical behavior of $\langle R \rangle$ for ID 6 in Table I. We evaluated the statistical error using the jackknife method, and the binning size is taken as larger than the autocorrelation time, which is estimated using Eq. (17). However, the determination of the window size W can bring systematic error in it, and we might underestimate the autocorrelation time. We believe it should have a larger error if we perform the more rigorous statistical operation, but we leave this for future work.

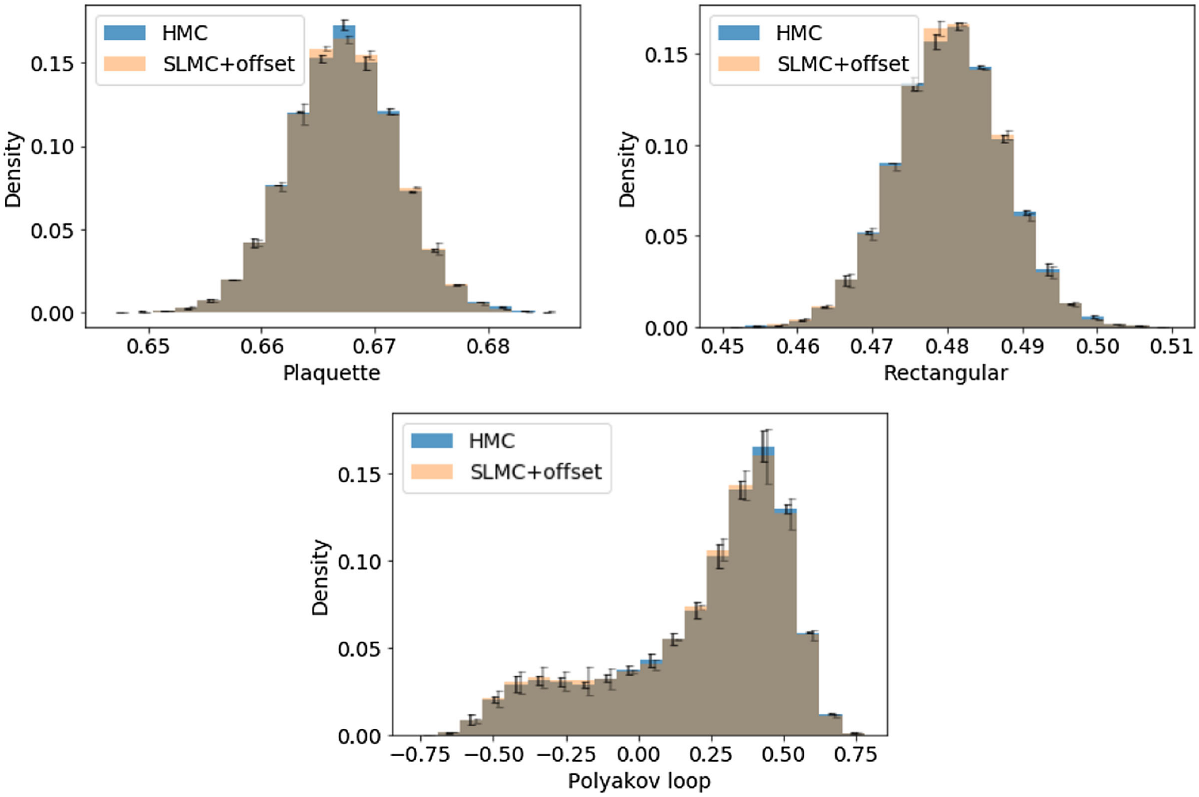


FIG. 3. Histogram of results from HMC and SLMC for $\beta = 2.5$, $N_\sigma^3 \times N_\tau = 6^4$, $\hat{m} = 0.5$. To distinguish these results, we shift results for SLMC to the right. Error bar is estimated by the Jackknife method. Left-top: plaquette. Right-top: rectangular Wilson loop. Middle: Polyakov loop.

Beta dependence are summarized in ID 8–13 in Table I. We vary β as 0.8, 1.2, 4.0 both in HMC and SLMC. The acceptance rate in SLMC for $\beta = 0.8$ is slightly low, but it gives consistent results. β dependence is correctly reproduced.

Results from lighter mass $\hat{m} = 0.05$ are summarized in ID 14–16 in Table I. In this case, for SLMC, acceptance is low (30%–40%), but it still gives consistent results to the ones in HMC. The acceptance rate for SLMC with higher order loops:

(i) SLMC_polys: effective action with chair, crown, and 3×1 Wilson loop, $\Omega_{1,2}$ in [59] notation,

is improved from SLMC. ID 17–18 in the table are results for $\hat{m} = 0.1$. The tendency of acceptance rate is the same to $\hat{m} = 0.05$ but slightly better as expected.

We examine volume dependence in the table (ID 19–21) and the volume is taken to $N_\sigma^3 \times N_\tau = 8^4$. Acceptance rate is lower than the SLMC with $N_\sigma^3 \times N_\tau = 6^4$. This is similar to what happens for the reweighting, but in our case, thanks to the tunable parameters, inefficiency is not drastic. Examinations for smaller volume $N_\sigma^3 \times N_\tau = 4^4$ are summarized in ID 22–23. The acceptance rate reaches to 80% for SLMC, and this is also expected.

In summary, SLMC can reproduce results from HMC even with plaquette effective action. The number of terms in the effective action affects to acceptance rate. For larger volumes and small mass, the acceptance rate becomes small. This can be improved by adding more and more terms to the effective action.

V. APPLICATION TO FINITE TEMPERATURE

A. Simulation setup

Here we present an application of SLMC algorithm to a finite temperature system. We perform simulation for four flavor QC$_2$D with $\hat{m} = 0.5$ in $N_\sigma^3 \times N_\tau = 8^3 \times 4$ lattice. As we will mention later, quarks are not decoupled from the theory. Our β range is $\beta = 1$–2.4, which contains a transition (crossover) point $\beta_c \sim 2.1$. We employ the Wilson plaquette gauge action and the standard staggered fermion. In SLMC update, 20 times heatbath updates are used except for $\beta = 2.1$ while 100 times heatbath for $\beta = 2.1$. The number of trajectory for HMC are 1000 for $\beta \neq 2.1$ and 20000 for $\beta = 2.1$ to see behavior of the Binder cumulant in detail. The number of trajectory for SLMC are O(5000)–O(40000) and please find details in Table II. We initialized $\beta_{\mathrm{plaq}} = \beta$, the coupling in the gauge action in target theory, and β_{others} are sampled as small random variables as well.

At the heavy quark mass regime, the system expected to show confinement/deconfinement transition (or crossover) associated with the Polyakov loop. The Polyakov

TABLE I. Simulation results for zero temperature with HMC and SLMC. ALG indicates different algorithms, and please see the main text for details. $\hat{m}$ indicates dimensionless quark mass. Acceptance means the acceptance rate. N_{trj} is the number of trajectories except for the thermalization. $\langle P\rangle$, $\langle R\rangle$, and $\langle L\rangle$ mean expectation value of plaquette, rectangular Wilson loop, and the Polyakov loop, respectively. We plot these values in Appendix C.

ID	ALG	N_σ	N_τ	β	m	Acceptance	N_{trj}	$\langle P\rangle$	$\langle R\rangle$	$\langle L\rangle$
0	HMC	6	6	2.5	0.50	0.65	50000	0.66718(5)	0.48037(9)	0.23(1)
1	SLMC_nup100	6	6	2.5	0.50	0.72	48850	0.66711(1)	0.48021(3)	0.197(3)
2	SLMC	6	6	2.5	0.50	0.73	50000	0.66718(3)	0.48031(5)	0.22(1)
3	SLMC_all	6	6	2.5	0.50	0.77	50000	0.66719(3)	0.48034(5)	0.19(1)
4	SLMCnor01	6	6	2.5	0.50	0.74	50000	0.66717(3)	0.48032(5)	0.23(2)
5	SLMCnor20	6	6	2.5	0.50	0.73	50000	0.66731(3)	0.48054(4)	0.191(9)
6	SLMCplq	6	6	2.5	0.50	0.57	50000	0.66746(4)	0.48074(5)	0.18(1)
7	SLMCplqrct	6	6	2.5	0.50	0.69	50000	0.66727(5)	0.48046(7)	0.211(8)
8	HMC	6	6	0.8	0.50	0.85	50000	0.20764(4)	0.04444(3)	0.0025(3)
9	HMC	6	6	1.2	0.50	0.84	50000	0.30356(4)	0.09382(3)	0.0034(2)
10	HMC	6	6	4.0	0.50	0.73	50000	0.80346(3)	0.68011(4)	0.65(1)
11	SLMC	6	6	0.8	0.50	0.58	50000	0.20781(5)	0.04454(5)	0.0032(4)
12	SLMC	6	6	1.2	0.50	0.59	50000	0.30365(4)	0.09387(3)	0.0033(7)
13	SLMC	6	6	4.0	0.50	0.87	50000	0.8035(2)	0.68016(3)	0.58(4)
14	HMC	6	6	2.5	0.05	0.82	50000	0.67774(4)	0.49772(5)	0.437(6)
15	SLMC	6	6	2.5	0.05	0.34	50000	0.67813(8)	0.4982(1)	0.436(7)
16	SLMC_polys	6	6	2.5	0.05	0.43	36350	0.67793(4)	0.49798(5)	0.446(7)
17	HMC	6	6	2.5	0.10	0.73	50000	0.6771(4)	0.49666(5)	0.428(6)
18	SLMC	6	6	2.5	0.10	0.37	50000	0.67749(7)	0.49732(9)	0.438(5)
19	HMC	8	8	2.5	0.50	0.77	50000	0.66659(2)	0.47916(2)	0.01(1)
20	SLMC	8	8	2.5	0.50	0.54	6630	0.66682(10)	0.4795(2)	−0.03(4)
21	SLMC_all	8	8	2.5	0.50	0.62	5300	0.66678(9)	0.4794(1)	0.04(3)
22	HMC	4	4	2.5	0.50	0.66	50000	0.6706(1)	0.4878(2)	0.64(2)
23	SLMC	4	4	2.5	0.50	0.84	50000	0.67073(7)	0.48792(9)	0.656(7)

TABLE II. Same table with Table I but for finite temperature with HMC and SLMC. We plot these values in Appendix C.

ID	ALG	N_σ	N_τ	β	m	Acceptance	N_{trj}	$\langle P\rangle$	$\langle R\rangle$	$\langle L\rangle$
0	HMC	8	4	1.0	0.5	0.88	1000	0.256(2)	0.067(1)	0.028(2)
1	HMC	8	4	1.2	0.5	0.89	1000	0.3037(2)	0.0939(1)	0.031(2)
2	HMC	8	4	1.4	0.5	0.87	1000	0.353(2)	0.1269(2)	0.036(1)
3	HMC	8	4	1.6	0.5	0.87	1000	0.4048(3)	0.1674(3)	0.039(2)
4	HMC	8	4	1.8	0.5	0.85	1000	0.4625(3)	0.2204(3)	0.055(3)
5	HMC	8	4	1.9	0.5	0.85	1000	0.4948(3)	0.2541(4)	0.078(3)
6	HMC	8	4	2.0	0.5	0.85	1000	0.5297(5)	0.2942(7)	0.137(7)
7	HMC	8	4	2.1	0.5	0.85	20200	0.5684(2)	0.3439(3)	0.327(4)
8	HMC	8	4	2.2	0.5	0.84	1000	0.6041(5)	0.3932(8)	0.553(6)
9	HMC	8	4	2.3	0.5	0.85	1000	0.6302(4)	0.4296(6)	0.679(4)
10	HMC	8	4	2.4	0.5	0.85	1000	0.6507(4)	0.4579(6)	0.757(3)
11	SLMC	8	4	1.0	0.5	0.44	990	0.2559(3)	0.0671(3)	0.028(3)
12	SLMC	8	4	1.2	0.5	0.47	41020	0.30368(6)	0.09393(7)	0.0319(5)
13	SLMC	8	4	1.4	0.5	0.48	41930	0.35296(7)	0.12676(7)	0.0347(5)
14	SLMC	8	4	1.6	0.5	0.46	41970	0.40492(5)	0.16744(6)	0.0424(5)
15	SLMC	8	4	1.8	0.5	0.47	41570	0.46257(7)	0.2205(8)	0.0578(7)
16	SLMC	8	4	1.9	0.5	0.47	5590	0.4949(3)	0.2544(4)	0.081(2)
17	SLMC	8	4	2.0	0.5	0.46	41200	0.53001(7)	0.2947(1)	0.14(1)
18	SLMCup100	8	4	2.1	0.5	0.41	11920	0.5682(2)	0.3437(3)	0.324(3)
19	SLMC	8	4	2.2	0.5	0.50	41180	0.6049(8)	0.3944(1)	0.559(1)
20	SLMC	8	4	2.3	0.5	0.55	34670	0.63037(4)	0.42958(6)	0.6775(6)
21	SLMC	8	4	2.4	0.5	0.60	34550	0.65066(3)	0.45782(4)	0.7558(5)

loop operator, along with the imaginary time direction, has the center symmetry, but it is broken at high temperatures. It means that the system around or above the (pseudo)critical temperature, the system could be affected by long autocorrelation for the Polyakov loop. We attempt to solve this autocorrelation by employing SLMC. Meanwhile, we show that SLMC with effective action can treat detail of phase transition without bias.

B. Simulation results

Here we show results at finite temperature. At the heavy mass regime, Polyakov loop is a central observable. Polyakov loop as a function of β is shown in the left panel of Fig. 4. One can see that a transition point is around $\beta \sim 2.1$. The right panel of Fig. 4 is the difference of results for the Polyakov loop from HMC and SLMC and the values are consistent with 0. Central values for other gluonic observables can be found in Table II.

Next, we show results for the susceptibility for Polyakov loop (left panel of Fig. 5). One can find a peak of around $\beta \sim 2.1$. Besides, results from HMC and SLMC are consistent with each other. The right panel of Fig. 5 is the Binder cumulant for Polyakov loop. Results from HMC and SLMC are consistent with each other. Besides, the Binder cumulant suggest that our quark mass $\hat{m} = 0.5$ is not in the quenched regime because in the quenched case, it takes a value for the second-order phase transition $B_L^4 \approx 1.6$ [60,61].

Next, we show results for the chiral condensates as a function of β (Left panel of Fig. 6). The transition point is located $\beta \sim 2.1$. The right panel of Fig. 6 is difference of results for the chiral condensates from HMC and SLMC. Results show that central values are consistent with each other.

Results of higher cumulants can be found in Fig. 7. The left panel of Fig. 7 shows the susceptibility for the chiral condensates while the right panel is Binder cumulant for the chiral condensates. Results from HMC and SLMC are consistent with each other.

Our results of the Binder cumulant for the Polyakov loop and chiral condensates (Figs. 5 and 7) at $\hat{m} = 0.5$ do not

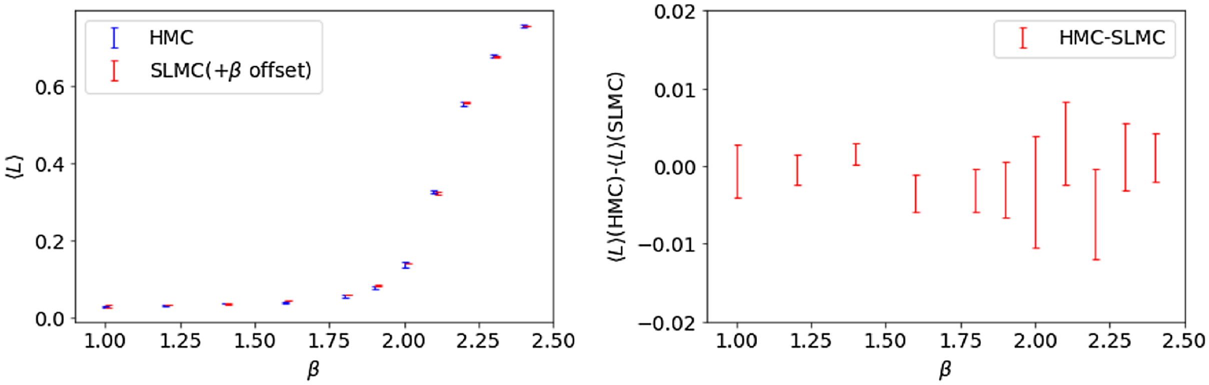


FIG. 4. Polyakov loop as a function of β. To avoid overlapping of symbols, we slightly shift in β for SLMC to the right in plots. Left: comparison plot for HMC and SLMC. Right: difference between Polyakov loop by HMC and SLMC.

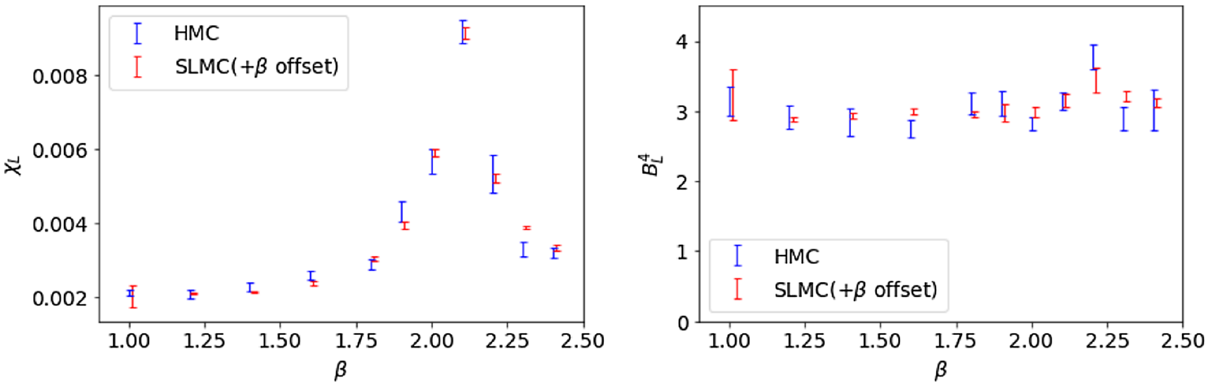


FIG. 5. Susceptibility and the Binder cumulant for Polyakov loop as a function of β. To avoid overlapping symbols, we shift β for SLMC in plots. Left; Polyakov loop susceptibility. It has a peak $\beta \sim 2.1$. Right: the Binder cumulant. It takes values about 3 for all β regime.

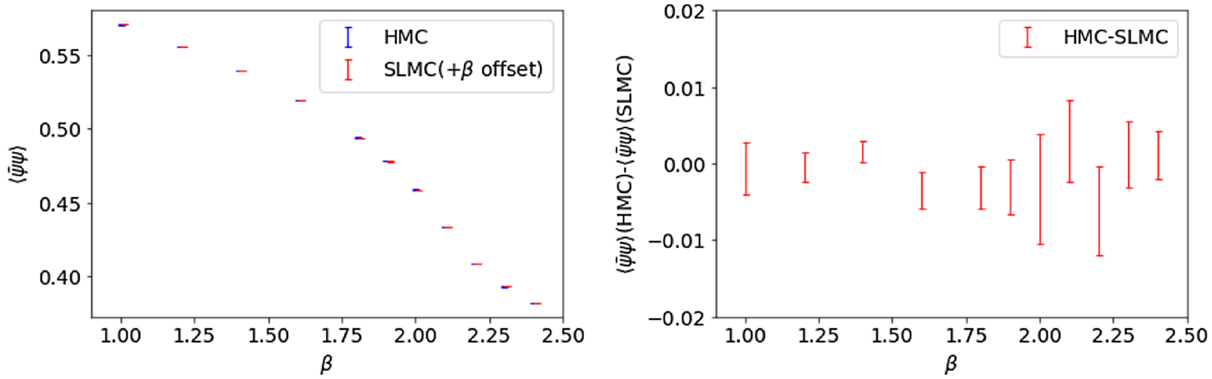


FIG. 6. Same plots of Fig. 4 but for the chiral condensate.

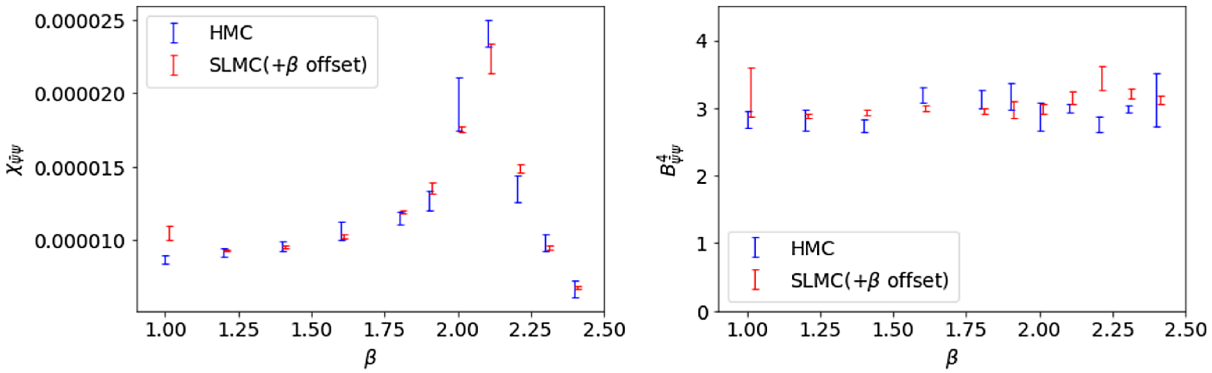


FIG. 7. Same plots of Fig. 5 but for the chiral condensate.

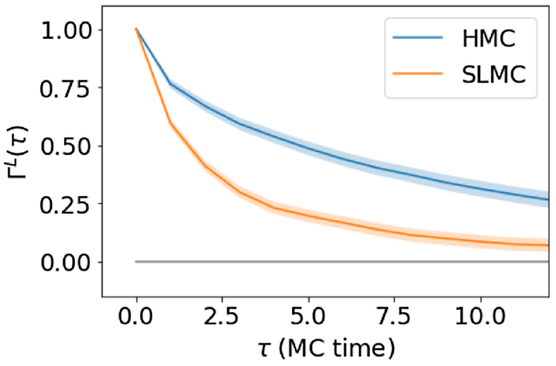


FIG. 8. Autocorrelation function for the Polyakov loop for $\beta = 2.1$, $N_\sigma^3 \times N_\tau = 8^3 \times 4$, $\hat{m} = 0.5$. the number of the heatbath is 100 in SLMC.

show any indication of the second order phase transition (with three-dimensional Ising universality class[13]). This can be confirmed by performing a systematic study for large volumes, a finer scan of beta, and increasing statistics, but we leave this issue for a future study.

As we have seen, $\beta = 2.1$ is the closest coupling to the critical (crossover) point (Fig. 8). We choose the number of heatbath updates as 100 in this case.

VI. CONCLUSION

In this work, we develop self-learning Monte-Carlo algorithm for lattice Yang-Mills theory with dynamical fermions in four dimensions. We work with QC$_2$D with $n_f = 4$ as an example with zero and finite temperature.

We confirm that SLMC works for zero temperature runs even for the out of expansion radius of the hopping parameter expansion because of the exactness. This is expected since SLMC itself is free from the choice of effective action. The acceptance rate becomes low for small β and large volume, but it can be fixed by adding more extended loops. Our code for automatic generation of heatbath code will be published in another paper.

For finite temperature runs, we confirm that SLMC reproduces correct results with HMC, including higher-order moments of the Polyakov loop and the chiral condensate. Our calculations indicate that QC$_2$D with $\hat{m} = 0.5$ is in the crossover regime, and we leave the precise determination of the order of phase transition to

[13] Svetitsky and Yaffe have conjectured the confinement/deconfinement transition for pure SU(2) in $3+1$ dimension is the second order phase transition with 3D-Ising universality class [62] while the chiral phase transition is first order for $n_f = 4$ light quarks by Pisarski and Wilczek [47,63].

future study since the critically of the phase transition is smeared if the system is small.

Our current algorithm uses heavy mass expanded effective action with the linear regression, which induces low efficiency of simulation for lighter mass. This would be fixed by employing a neural network like [34].

ACKNOWLEDGMENTS

A. T. would like to thank to P. S. Bedaque W. Detmold, K. Orginos during a workshop “A. I. for Nuclear Physics Workshop” at Jefferson Lab for notifying references and comments, and S. Valgushev for fruitful discussion. The work of Akio T. was supported by the RIKEN Special Postdoctoral Researcher program and partially supported by JSPS KAKENHI Grant No. JP20K14479. The calculations were partially performed by the supercomputing systems SGI ICE X at the Japan Atomic Energy Agency. The numerical calculations were partially carried out on XC40 at YITP in Kyoto University. Y. N. was partially supported by JSPS-KAKENHI Grants No. 18K11345, No. 18K03552, No. 22K03539, and No. 22H05111. The work of Akinori T. was partially supported by JSPS KAKENHI Grants No. 18K13548 and No. 22H05116.

APPENDIX A: REVIEW OF METROPOLIS–HASTINGS ALGORITHM AND SLMC

1. Metropolis–Hastings update

For given variable U, MH update step is composed of the following two steps:

(1) sampling a candidate variable U' by a fixed conditional probability $P(U'|U)$

(2) accepting the candidate with probability $p_{\mathrm{acc}}(U', U) = \min\left(1, \frac{e^{-S[U']}P(U|U')}{e^{-S[U]}P(U'|U)}\right)$, otherwise rejecting the candidate U' and keeping U.

This conditional sampling step can be understood as sampling from the conditional probability below:

$$P_{\mathrm{MH}}(U'|U) = p_{\mathrm{acc}}(U', U)P(U'|U) + \delta(U'-U)\int dU_{\mathrm{rej}}(1-p_{\mathrm{acc}}(U_{\mathrm{rej}}, U)) \times P(U_{\mathrm{rej}}|U), \tag{A1}$$

where the δ represents Dirac’s delta function for continuous stochastic variable. The first term corresponds to acceptance, and the second term represents the rejection process, which happens with probability $1-p_{\mathrm{acc}}$ [64]. The MH probability (A1) satisfies two important conditions explained below.

a. Normalization condition

This complete form satisfies the normalization condition of the conditional probability density:

$$\begin{aligned}\int dU' P_{\mathrm{MH}}(U'|U) &= \int dU' p_{\mathrm{acc}}(U', U)P(U'|U) + \int dU'\delta(U'-U)\int dU_{\mathrm{rej}}(1-p_{\mathrm{acc}}(U_{\mathrm{rej}}, U))P(U_{\mathrm{rej}}|U)\\ &= \int dU' p_{\mathrm{acc}}(U', U)P(U'|U) + \int dU_{\mathrm{rej}}(1-p_{\mathrm{acc}}(U_{\mathrm{rej}}, U))P(U_{\mathrm{rej}}|U)\\ &= \int dU_{\mathrm{rej}}P(U_{\mathrm{rej}}|U) = 1\end{aligned}$$

If the variable U is discrete valued, the integral is replaced by the summation and the Dirac’s delta to Kronecker’s delta.

b. Detailed balance condition

The probability (A1) is sum of two terms, and it is sufficient to show detailed balance condition of each term. First, let us focus on the first term $p_{\mathrm{acc}}(U', U)P(U'|U) =: P_{\text{MH-acc}}(U'|U)$. By using a simple formula, $\min(1, x) = \min(x^{-1}, 1)x$, we can show the detailed balance condition for $P_{\text{MH-acc}}$ as follows,

$$\begin{aligned}P_{\text{MH-acc}}(U'|U) &= \min\Bigg(1, \underbrace{\frac{e^{-S[U']}P(U|U')}{e^{-S[U]}P(U'|U)}}_{x}\Bigg)P(U'|U)\\ &= \min(x^{-1}, 1)xP(U'|U)\\ &= \frac{e^{-S[U']}}{e^{-S[U]}}P_{\text{MH-acc}}(U|U').\end{aligned}$$

We need to show the detailed balance condition for the second term also. To make the argument clearer, let us call the second term $\delta(U'-U)\int dU_{\mathrm{rej}}(1-p_{\mathrm{acc}}(U_{\mathrm{rej}}, U))P(U_{\mathrm{rej}}|U)$

as $P_{\text{MH-rej}}(U'|U)$, then we can multiply $\frac{e^{-S[U']}}{e^{-S[U]}}$ and replace U by U' because of the insertion of delta function:

$$\begin{aligned}P_{\text{MH-rej}}(U'|U) &= \frac{e^{-S[U']}}{e^{-S[U]}}\delta(U-U') \\ &\quad\times\int dU_{\text{rej}}(1-p_{\text{acc}}(U_{\text{rej}},U'))P(U_{\text{rej}}|U') \\ &= \frac{e^{-S[U']}}{e^{-S[U]}}P_{\text{MH-rej}}(U|U'),\end{aligned}$$

and it completes the proof, i.e.,

$$\begin{aligned}P_{\text{MH}}(U'|U) &= P_{\text{MH-acc}}(U'|U)+P_{\text{MH-rej}}(U'|U) \\ &= \frac{e^{-S[U']}}{e^{-S[U]}}P_{\text{MH-acc}}(U|U')+\frac{e^{-S[U']}}{e^{-S[U]}}P_{\text{MH-rej}}(U|U') \\ &= \frac{e^{-S[U']}}{e^{-S[U]}}P_{\text{MH}}(U|U').\end{aligned}$$

2. Simplification in SLMC

In SLMC, we prepare the model S^{θ}_{eff} with fitting parameters θ with simple update $P_\theta(U'|U)$ that satisfies detailed balance condition for S^{θ}_{eff}:

$$P_\theta(U'|U) = \frac{e^{-S^{\theta}_{\text{eff}}[U']}}{e^{-S^{\theta}_{\text{eff}}[U]}}P_\theta(U|U'), \tag{A2}$$

and use it as the proposal probability $P(U'|U)$ in (A1). In this case, we can utilize the above detailed balance condition (A2) to simplify the calculation of acceptance probability in each step:

$$\begin{aligned}p_{\text{acc}}(U',U) &= \min\left(1,\frac{e^{-S[U']}P_\theta(U|U')}{e^{-S[U]}P_\theta(U'|U)}\right) \\ &= \min\left(1,\frac{e^{-(S[U']-S^{\theta}_{\text{eff}}[U'])}}{e^{-(S[U]-S^{\theta}_{\text{eff}}[U])}}\right).\end{aligned}$$

We use the heatbath update protocol as P_θ in this paper, that is explained in Appendix B.

APPENDIX B: HEATBATH UPDATE AND THE ROLE OF STAPLE

In this section, we consider the following action:

$$S_{\text{loop}}[U] = \text{const} + \text{tr}(U_1U_2\cdots U_{\spadesuit}), \tag{B1}$$

where the set of link variables $U_1, U_2, \ldots, U_{\spadesuit}$ should correspond to a closed loop. Note that we take simplified notation for the link variables, i.e., it should be written like $U_1 = U_\mu(n)$ for certain μ and n in the main text notation. We focus on the loop not self-intersected, then for an arbitrary U_i, $S_{\text{loop}}[U]$ is factorized as

$$S_{\text{loop}}[U] = \text{const} + \text{tr}(U_iV_i), \tag{B2}$$

where $V_i = U_{i+1}\cdots U_{\spadesuit}U_1U_2\cdots U_{i-1}$ is called staple. Heatbath update is the update based on the conditional probability

$$P_{\text{HB}}(U_i|\hat{U}_i) = \frac{e^{-\text{tr}(U_iV_i)}}{z_{\text{loop}}(\hat{U}_i)}, \tag{B3}$$

where $\hat{U}_i$ means the set of link variables except for U_i, $\hat{U}_i = \{U_1, U_2, \ldots, U_{i-1}, U_{i+1}, \ldots U_{\spadesuit}\}$. Note that the staple V_i only depends on $\hat{U}_i$, and we name the normalization factor as $z_{\text{loop}}(\hat{U}_i) = \int dU_i e^{-\text{tr}(U_iV_i)}$. Now let us consider the following transition probability for update:

$$P(U'|U) = \delta(\hat{U}'_i - \hat{U}_i)P_{\text{HB}}(U'_i|\hat{U}_i), \tag{B4}$$

then the staple is invariant $V'_i = V_i$ under the update, and it is straightforward to show the detailed balance condition for the action (B1):

$$P(U'|U) = \frac{e^{-S_{\text{loop}}[U']}}{e^{-S_{\text{loop}}[U]}}P(U|U').$$

So, if we can implement the sampling from the conditional probability (B3), it provides a "legal" update transition probability defined by (B4). Once the matrix value of the staple V_i is given, sampling from (B3) is immediate. In addition, we can repeat same procedure for a more generic action

$$\begin{aligned}S^{\theta}_{\text{loops}}(U) &= S_{\text{loop1}}(U) + S_{\text{loop2}}(U) + \cdots \\ &= \text{const} + \beta_{\text{loop1}}\text{tr}(U_iV^{(1)}_i) \\ &\quad + \beta_{\text{loop2}}\text{tr}(U_iV^{(2)}_i) + \cdots,\end{aligned} \tag{B5}$$

where $\theta = \{\beta_{\text{loop1}}, \beta_{\text{loop2}}, \ldots\}$. In this case, the corresponding conditional probability is

$$\begin{aligned}P_\theta(U_i|\hat{U}_i) &= \frac{e^{-\beta_{\text{loop1}}\text{tr}(U_iV^{(1)}_i)-\beta_{\text{loop2}}\text{tr}(U_iV^{(2)}_i)-\cdots}}{\int dU_ie^{-\beta_{\text{loop1}}\text{tr}(U_iV^{(1)}_i)-\beta_{\text{loop2}}\text{tr}(U_iV^{(2)}_i)-\cdots}} \\ &= \frac{e^{-\text{tr}(U_iV^{\text{loops}}_i)}}{\int dU_ie^{-\text{tr}(U_iV^{\text{loops}}_i)}},\end{aligned} \tag{B6}$$

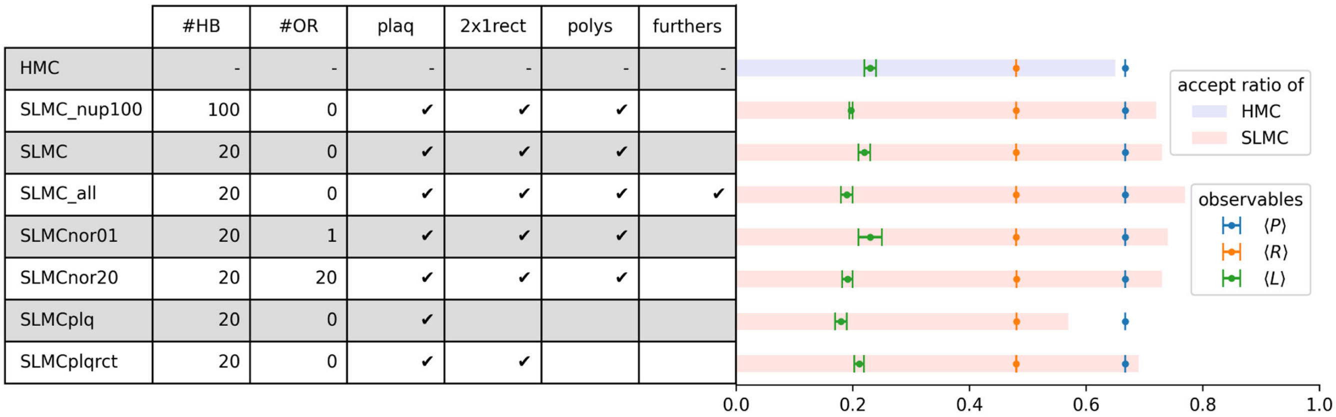
where $V^{\text{loops}}_i = \beta_{\text{loop1}}V^{(1)}_i + \beta_{\text{loop2}}V^{(2)}_i + \cdots$, and reduces to the sampling from (B3) for a given staple $V_i = V^{\text{loops}}_i$.

APPENDIX C: PLOTS

In this section, we show plots for Tables I and II and additional information on SLMC parameters in the numerical simulations.

1. Plots for Tables I and II

We plot accept ratio as bar graph with blue color for HMC, and with red colors for SLMC. For observables, $\langle P\rangle =$ plaquette, $\langle R\rangle =$ rectangular Wilson loop, $\langle L\rangle =$ Polyakov loop, are plotted with error bar.



	#HB	#OR	plaq	2x1rect	polys	furthers
HMC	-	-	-	-	-	-
SLMC_nup100	100	0	✔	✔	✔	
SLMC	20	0	✔	✔	✔	
SLMC_all	20	0	✔	✔	✔	✔
SLMCnor01	20	1	✔	✔	✔	
SLMCnor20	20	20	✔	✔	✔	
SLMCplq	20	0	✔			
SLMCplqrct	20	0	✔	✔		

FIG. 9. Plots for ID 0–7 in Table I. All simulations are executed by using $N_\sigma = N_\tau = 6$, $\beta = 2.5$, $m = 0.50$. First two columns mean: #HB = number of heatbath updates, #OR = number of overrelaxations. Remaining columns are showing $S^\theta_{\rm eff}$ including summation of: plaq = plaquettes, 2 × 1rect = rectangulars with size 2 × 1, polys = Polyakov loops along x, y, z, t directions, furthers = rectangular with size 3 × 1 + chairs + crowns, for SLMC effective action.

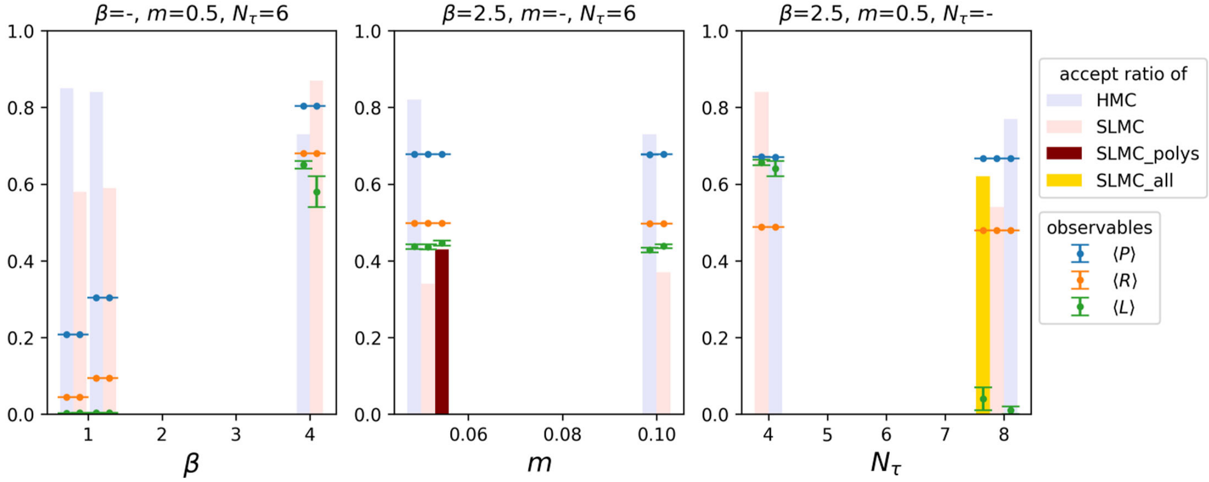


FIG. 10. Plots for ID 8–23 in Table I. SLMC_polys executed SLMC with $S^\theta_{\rm eff}$ including basic terms [plaq, 2 × 1rect, polys] and additional terms including [3 × 1rect, 4 × 1rect, chair, crown] and bended polyakov loops Ω_1, Ω_2 which are defined in [59]. The setups of SLMC and SLMC_all are same as shown in Fig. 9.

a. Table I (ID 0–7)

In Table I, we have 24 entries for each column. We plot the first 8 entries in Fig. 9. The 1st entry (ID 0) is for HMC with leapfrog integration $N_{\rm int} = 10$ times, and we treat it as a baseline. The remaining entries are for SLMCs.

b. Table I (ID 8–23)

We plot the remaining 16 entries in Fig. 10. As one can see, adding higher loop actions to $S^\theta_{\rm eff}$ increases the acceptance, e.g., SLMC_polys in middle, SLMC_all in right, without changing the values of observables within error bars.

c. Table II

We plot the all entries in Fig. 11.

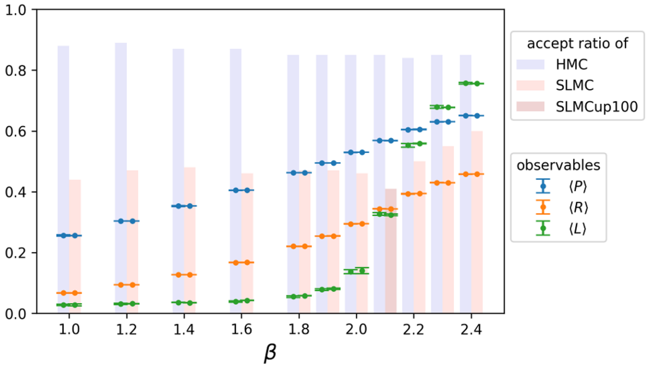


FIG. 11. Plots for ID 0–21 in Table II. The setups of SLMC and SLMCup100 are same as shown in Fig. 9.

2. Effective action parameters

In this section, we show some determined set of parameters θ of $S_{\mathrm{eff}}^{\theta}$ for SLMC in Tables I and II with some comments.

a. Basic setup for SLMC in zero temperature

In Table III, we show the determined parameters with the effective action defined in (9). As one can see, each fitted value β_{plaq} takes closer value to β. This is plausible from physical perspectives. For example, one can execute the hopping parameter expansion at heavy mass regime, and in that case, the effective action takes similar form, and $\beta_{\mathrm{plaq}} = \beta$. In addition, one can see the sign of $\beta_{\mathrm{Pol}}^{\mu=4}$ is opposite to the sign of $\beta_{\mathrm{Pol}}^{\mu=1,2,3}$. This is also consistent with the hopping parameter expansion because of antiperiodicity of the fermion along time direction. Let us emphasize that we can determine each parameter even if the fermion mass is relatively small.

b. SLMC with additional loops in zero temperature

In Tables IV and V, we show the determined parameters with effective actions with more terms. In the table, we use the same terminology used in [59].

c. SLMC in finite temperature

We summarize parameters in Table VI. In this case, we can obsreve that $|\beta_{\mathrm{Pol}}^{\mu=4}| > |\beta_{\mathrm{Pol}}^{\mu=1,2,3}|$. This is also consistent with the hopping parameter expansion.

TABLE III. The determined parameters with the effective action for zero temperature.

ID	ALG	N_τ	β	m	β_{plaq}	β_{rect}	$\beta_{\mathrm{Pol}}^{\mu=1}$	$\beta_{\mathrm{Pol}}^{\mu=2}$	$\beta_{\mathrm{Pol}}^{\mu=3}$	$\beta_{\mathrm{Pol}}^{\mu=4}$	β_{const}
11	SLMC	6	0.8	0.50	0.848	0.0195	0.00247	0.00155	0.00180	−0.00468	0.508
12	SLMC	6	1.2	0.50	1.25	0.0197	0.00189	0.00166	0.00183	−0.00456	0.909
13	SLMC	6	4.0	0.50	4.04	0.0173	0.00660	0.00658	0.00657	−0.00653	3.70
18	SLMC	6	2.5	0.10	2.55	0.0551	0.0243	0.0242	0.0241	−0.0238	1.72
20	SLMC	8	2.5	0.50	2.54	0.0264	0.00167	0.00143	0.00175	−0.00175	2.20
23	SLMC	4	2.5	0.50	2.55	0.0159	0.0396	0.0395	0.0396	−0.0396	2.20

TABLE IV. The determined parameters with the effective action in for ID16 SLMC_polys ($N = 6, \beta = 2.5, m = 0.05$).

β_{plaq}	$\beta_{2\times 1\mathrm{rect}}$	$\beta_{\mathrm{Pol}}^{\mu=1}$	$\beta_{\mathrm{Pol}}^{\mu=2}$	$\beta_{\mathrm{Pol}}^{\mu=3}$	$\beta_{\mathrm{Pol}}^{\mu=4}$	$\beta_{3\times 1\mathrm{rect}}$	$\beta_{4\times 1\mathrm{rect}}$	β_{chair}	
2.52	0.0284	−0.0264	−0.0294	−0.0261	0.0284	0.00689	0.00136	0.00279	
β_{crown}	$\beta_{\Omega_1}^{\mu=1}$	$\beta_{\Omega_1}^{\mu=2}$	$\beta_{\Omega_1}^{\mu=3}$	$\beta_{\Omega_1}^{\mu=4}$	$\beta_{\Omega_2}^{\mu=1}$	$\beta_{\Omega_2}^{\mu=2}$	$\beta_{\Omega_2}^{\mu=3}$	$\beta_{\Omega_2}^{\mu=4}$	β_{const}
0.00555	−0.000938	−0.00106	−0.000945	0.00105	−0.000201	−0.000141	−0.000200	0.000122	1.48

TABLE V. ID21 SLMC_all ($N = 8, \beta = 2.5, m = 0.50$).

β_{plaq}	$\beta_{2\times 1\mathrm{rect}}$	$\beta_{\mathrm{Pol}}^{\mu=1}$	$\beta_{\mathrm{Pol}}^{\mu=2}$	$\beta_{\mathrm{Pol}}^{\mu=3}$	$\beta_{\mathrm{Pol}}^{\mu=4}$	$\beta_{3\times 1\mathrm{rect}}$	β_{chair}	β_{crown}	β_{const}
2.52	0.0155	0.00161	0.00172	0.00171	−0.00152	0.00274	0.000863	0.00263	2.20

TABLE VI. The determined parameters with the effective action for finite temperature.

ID	ALG	N_σ	N_τ	β	m	β_{plaq}	β_{rect}	$\beta_{\mathrm{Pol}}^{\mu=1}$	$\beta_{\mathrm{Pol}}^{\mu=2}$	$\beta_{\mathrm{Pol}}^{\mu=3}$	$\beta_{\mathrm{Pol}}^{\mu=4}$	β_{const}
12	SLMC	8	4	1.2	0.5	1.25	0.0212	0.000288	−0.000940	0.000238	−0.0329	0.909
14	SLMC	8	4	1.6	0.5	1.65	0.0225	0.000585	0.000212	0.00121	−0.0364	1.31
16	SLMC	8	4	1.9	0.5	1.95	0.0252	0.000475	0.000357	0.000620	−0.0420	1.61
18	SLMCup100	8	4	2.1	0.5	2.14	0.0303	0.000731	0.000511	0.000563	−0.0505	1.80
20	SLMC	8	4	2.3	0.5	2.34	0.0266	0.00115	0.00117	0.00135	−0.0456	2.00

[1] M. Creutz, Monte Carlo study of quantized SU(2) gauge theory, Phys. Rev. D **21**, 2308 (1980).

[2] S. Aoki *et al.*, FLAG review 2019: Flavour lattice averaging group (FLAG), Eur. Phys. J. C **80**, 113 (2020).

[3] Kohtaroh Miura, Review of lattice QCD studies of hadronic vacuum polarization contribution to muon $g-2$, Proc. Sci. LATTICE2018 (**2019**) 010.

[4] A. D. Kennedy, Algorithms for dynamical fermions, 2006.

[5] Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth, Hybrid Monte Carlo, Phys. Lett. B **195**, 216 (1987).

[6] M. A. Clark, The rational hybrid Monte Carlo algorithm, Proc. Sci. LAT2006 (**2006**) 004.

[7] Stefan Schaefer, Rainer Sommer, and Francesco Virotta, Critical slowing down and error analysis in lattice QCD simulations, Nucl. Phys. **B845**, 93 (2011).

[8] B. A. Berg and T. Neuhaus, Multicanonical Ensemble: A New Approach to Simulate First Order Phase Transitions, Phys. Rev. Lett. **68**, 9 (1992).

[9] Heng-Tong Ding, Christian Schmidt, Akio Tomiya, and Xiao-Dan Wang, Chiral phase structure of three flavor QCD in a background magnetic field, Phys. Rev. D **102**, 054505 (2020).

[10] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G. R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab, A high-bias, low-variance introduction to machine learning for physicists, Phys. Rep. **810**, 1 (2019).

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

[12] Stefan Blücher, Lukas Kades, Jan M. Pawlowski, Nils Strodthoff, and Julian M. Urban, Towards novel insights in lattice field theory with explainable machine learning, Phys. Rev. D **101**, 094507 (2020).

[13] Boram Yoon, Estimation of matrix trace using machine learning, 2016.

[14] Boram Yoon, Tanmoy Bhattacharya, and Rajan Gupta, Machine learning estimators for lattice QCD observables, Phys. Rev. D **100**, 014504 (2019).

[15] Rui Zhang, Zhouyou Fan, Ruizi Li, Huey-Wen Lin, and Boram Yoon, Machine-learning prediction for quasiparton distribution function matrix elements, Phys. Rev. D **101**, 034516 (2020).

[16] Kim A. Nicoli, Christopher J. Anders, Lena Funcke, Tobias Hartung, Karl Jansen, Pan Kessel, Shinichi Nakajima, and Paolo Stornati, On estimation of thermodynamic observables in lattice field theories with deep generative models, 2020.

[17] Akinori Tanaka and Akio Tomiya, Towards reduction of autocorrelation in hmc by machine learning, 2017.

[18] Jan M. Pawlowski and Julian M. Urban, Reducing autocorrelation times in lattice simulations with generative adversarial networks, Mach. Learn. Sci. Technol. **1**, 045011 (2020).

[19] Kai Zhou, Gergely Endrodi, Long-Gang Pang, and Horst Stocker, Regressive and generative neural networks for scalar field theory, Phys. Rev. D **100**, 011501 (2019).

[20] M. S. Albergo, G. Kanwar, and P. E. Shanahan, Flow-based generative models for Markov Chain Monte Carlo in lattice field theory, Phys. Rev. D **100**, 034515 (2019).

[21] Gurtej Kanwar, Michael S. Albergo, Denis Boyda, Kyle Cranmer, Daniel C. Hackett, Sebastien Racaniere, Danilo Jimenez Rezende, and Phiala E. Shanahan, Equivariant Flow-Based Sampling for Lattice Gauge Theory, Phys. Rev. Lett. **125**, 121601 (2020).

[22] Denis Boyda, Gurtej Kanwar, Sebastien Racaniere, Danilo Jimenez Rezende, Michael S. Albergo, Kyle Cranmer, Daniel C. Hackett, and Phiala E. Shanahan, Sampling using $su(n)$ gauge equivariant flows, 2020.

[23] Andrei Alexandru, Paulo F. Bedaque, Henry Lamm, and Scott Lawrence, Deep learning beyond lefschetz thimbles, Phys. Rev. D **96**, 094505 (2017).

[24] Phiala E. Shanahan, Daniel Trewartha, and William Detmold, Machine learning action parameters in lattice quantum chromodynamics, Phys. Rev. D **97**, 094506 (2018).

[25] Lukas Kades, Jan M. Pawlowski, Alexander Rothkopf, Manuel Scherzer, Julian M. Urban, Sebastian J. Wetzel, Nicolas Wink, and Felix Ziegler, Spectral reconstruction with deep neural networks, 2019.

[26] Boram Yoon, Tanmoy Bhattacharya, and Rajan Gupta, Machine learning estimators for lattice QCD observables, 2019.

[27] Sam Offler, Gert Aarts, Chris Allton, Jonas Glesaaen, Benjamin Jager, Seyong Kim, Maria Paola Lombardo, Sinead M. Ryan, and Jon-Ivar Skullerud, News from bottomonium spectral functions in thermal QCD, 2019.

[28] Rui Zhang, Carson Honkala, Huey-Wen Lin, and Jiunn-Wei Chen, Pion and kaon distribution amplitudes in the continuum limit, 2020.

[29] D. L. Boyda, M. N. Chernodub, N. V. Gerasimeniuk, V. A. Goy, S. D. Liubimov, and A. V. Molochkov, Machine-learning physics from unphysics: Finding deconfinement temperature in lattice Yang-Mills theories from outside the scaling window, 2020.

[30] Yarin Gal, Vishnu Jejjala, Damian Kaloni Mayorga Pena, and Challenger Mishra, Baryons from mesons: A machine learning perspective, 2020.

[31] Sebastian Johann Wetzel and Manuel Scherzer, Machine learning of explicit order parameters: From the Ising model to SU(2) lattice gauge theory, Phys. Rev. B **96**, 184410 (2017).

[32] Junwei Liu, Yang Qi, Zi Yang Meng, and Liang Fu, Self-learning Monte Carlo method, Phys. Rev. B **95**, 041101 (2017).

[33] Yuki Nagai, Huitao Shen, Yang Qi, Junwei Liu, and Liang Fu, Self-learning Monte Carlo method: Continuous-time algorithm, Phys. Rev. B **96**, 161102 (2017).

[34] Huitao Shen, Junwei Liu, and Liang Fu, Self-learning Monte Carlo with deep neural networks, Phys. Rev. B **97**, 205140 (2018).

[35] Chuang Chen, Xiao Yan Xu, Junwei Liu, George Batrouni, Richard Scalettar, and Zi Yang Meng, Symmetry-enforced self-learning Monte Carlo method applied to the Holstein model, Phys. Rev. B **98**, 041102 (2018).

[36] Yuki Nagai, Masahiko Okumura, Keita Kobayashi, and Motoyuki Shiga, Self-learning hybrid Monte Carlo: A first-principles approach, Phys. Rev. B **102**, 041124 (2020).

[37] Junwei Liu, Huitao Shen, Yang Qi, Zi Yang Meng, and Liang Fu, Self-Learning Monte Carlo Method in Fermion Systems, Phys. Rev. B **95**, 241104 (2017).

[38] Yuki Nagai, Masahiko Okumura, Keita Kobayashi, and Motoyuki Shiga, Self-learning hybrid Monte Carlo: A first-principles approach, Phys. Rev. B **102**, 041124 (2020).

[39] Alan C. Irving and James C. Sexton, Approximate actions for lattice QCD simulation, Phys. Rev. D **55**, 5456 (1997).

[40] Alan C. Irving, James C. Sexton, and Eamonn Cahill, Approximate actions for dynamical fermions, Nucl. Phys. B, Proc. Suppl. **63**, 967 (1998).

[41] A. Duncan, E. Eichten, and H. Thacker, Efficient algorithm for QCD with light dynamical quarks, Phys. Rev. D **59**, 014505 (1998).

[42] A. Duncan, E. Eichten, and H. Thacker, Truncated determinant approach to light dynamical quarks, Nucl. Phys. B, Proc. Suppl. **73**, 837 (1999).

[43] A. Duncan, E. Eichten, R. Roskies, and H. Thacker, Loop representations of the quark determinant in lattice QCD, Phys. Rev. D **60**, 054505 (1999).

[44] Martin Hasenbusch, Exploiting the hopping parameter expansion in the hybrid Monte Carlo simulation of lattice QCD with two degenerate flavors of Wilson fermions, Phys. Rev. D **97**, 114512 (2018).

[45] A. Borrelli, Ph. de Forcrand, and A. Galli, Non-hermitian exact local bosonic algorithm for dynamical quarks, Nucl. Phys. **B477**, 809 (1996).

[46] C. Alexandrou, Ph. de Forcrand, M. D'Elia, and H. Panagopoulos, Improved multiboson algorithm, Nucl. Phys. B, Proc. Suppl. **83–84**, 765 (2000).

[47] O. Kaczmarek, F. Karsch, and E. Laermann, Thermodynamics of two-colour QCD, Nucl. Phys. B, Proc. Suppl. **73**, 441 (1999).

[48] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah, Julia: A fresh approach to numerical computing, SIAM Rev. **59**, 65 (2017).

[49] A. Tomiya Y. Nagai and A. Tanaka, Julia code for LQCD (to be published).

[50] Anna Hasenfratz, Roland Hoffmann, and Stefan Schaefer, Reweighting towards the chiral limit, Phys. Rev. D **78**, 014515 (2008).

[51] Thomas DeGrand, Reweighting QCD simulations with dynamical overlap fermions, Phys. Rev. D **78**, 117504 (2008).

[52] H. Saito, S. Ejiri, S. Aoki, T. Hatsuda, K. Kanaya, Y. Maezawa, H. Ohno, and T. Umeda, Phase structure of finite temperature QCD in the heavy quark region, Phys. Rev. D **84**, 054502 (2011).

[53] Thomas Blum, Taku Izubuchi, and Eigo Shintani, New class of variance-reduction techniques using lattice symmetries, Phys. Rev. D **88**, 094503 (2013).

[54] Sourendu Gupta, A. Irback, F. Karsch, and B. Petersson, The acceptance probability in the hybrid Monte Carlo method, Phys. Lett. B **242**, 437 (1990).

[55] Kurt Binder, Critical Properties from Monte Carlo Coarse Graining and Renormalization, Phys. Rev. Lett. **47**, 693 (1981).

[56] Ulli Wolff, Monte Carlo errors with less errors, Comput. Phys. Commun. **156**, 143 (2004); **176**, 383(E) (2007).

[57] Neal Madras and Alan D. Sokal, The Pivot algorithm: A highly efficient Monte Carlo method for selfavoiding walk, J. Stat. Phys. **50**, 109 (1988).

[58] Martin Luscher, Schwarz-preconditioned HMC algorithm for two-flavour lattice QCD, Comput. Phys. Commun. **165**, 199 (2005).

[59] Naoki Wakabayashi, Shinji Ejiri, Kazuyuki Kanaya, and Masakiyo Kitazawa, Scope and convergence of the hopping parameter expansion in finite-temperature quantum chromodynamics with heavy quarks around the critical point, Prog. Theor. Exp. Phys. **2022**, 033B05 (2022).

[60] J. Engels, J. Fingberg, and M. Weber, Finite size scaling analysis of SU(2) lattice gauge theory in $(3+1)$-dimensions, Nucl. Phys. **B332**, 737 (1990).

[61] A. Denbleyker, Yuzhi Liu, Y. Meurice, and A. Velytsky, Finite size scaling and universality in su(2) at finite temperature, Proc. Sci. LAT2009 **2010**, 197.

[62] Benjamin Svetitsky and Laurence G. Yaffe, Critical behavior at finite temperature confinement transitions, Nucl. Phys. **B210**, 423 (1982).

[63] Robert D. Pisarski and Frank Wilczek, Remarks on the chiral phase transition in chromodynamics, Phys. Rev. D **29**, 338 (1984).

[64] Martin Luscher, Computational strategies in lattice QCD, in *Les Houches Summer School: Session 93: Modern Perspectives in Lattice QCD: Quantum Field Theory and High Performance Computing*, Lecture Notes of the Les Houches Summer School (Oxford Academic, Oxford, 2012), https://academic.oup.com/book/25536.