# Mitigating the Hubbard sign problem with complex-valued neural networks

Marcel Rodekamp [1,2,3,4] Evan Berkowitz [1,2,3] Christoph Gäntgen [1,3,4] Stefan Krieg [1,2,3,4]
Thomas Luu [1,4,5] and Johann Ostmeyer [6]

[1]*Institute for Advanced Simulation, Forschungszentrum Jülich, 54245 Jülich, Germany*
[2]*JARA & Jülich Supercomputing Center, Forschungszentrum Jülich, 54245 Jülich, Germany*
[3]*Center for Advanced Simulation and Analytics (CASA), Forschungszentrum Jülich, 52425 Jülich, Germany*
[4]*Helmholtz-Institut für Strahlen- und Kernphysik, Rheinische Friedrich-Wilhelms-Universität Bonn, 53115 Bonn, Germany*
[5]*Institut für Kernphysik, Forschungszentrum Jülich, 54245 Jülich, Germany*
[6]*Department of Mathematical Sciences, University of Liverpool, Liverpool L69 7ZL, United Kingdom*

(Received 21 March 2022; revised 26 August 2022; accepted 2 September 2022; published 22 September 2022)

Monte Carlo simulations away from half filling suffer from a sign problem that can be reduced by deforming the contour of integration. Such a transformation, which induces a Jacobian determinant in the Boltzmann weight, can be implemented using neural networks. This additional determinant cost for a generic neural network scales cubically with the volume, preventing large-scale simulations. We implement an architecture, based on complex-valued affine coupling layers, which reduces this to linear scaling. We demonstrate the efficacy of this method by successfully applying it to systems of different size, the largest of which is intractable by other Monte Carlo methods due to its severe sign problem.

DOI: 10.1103/PhysRevB.106.125139

## I. INTRODUCTION

The computational sign problem encumbers successful importance sampling from complex-valued distributions with Markov chain Monte Carlo algorithms such as hybrid Monte Carlo (HMC). Sampling from the configuration space of a wide variety of interesting physical systems suffers such a difficulty, ranging from lattice quantum chromodynamics (QCD) at finite baryon chemical potential and doped condensed matter systems in equilibrium to the real-time evolution of quantum systems.

By deforming the real manifold of integration for a path integral of interest into complex variables, one may reduce the sign problem substantially [1–4]. In the last few years, new formal developments have inspired investigation into leveraging Lefschetz thimbles [5–11]—high-dimensional analogs of contours of steepest descent which can be located by holomorphic flow. In Ref. [6], for example, fluctuations about the saddle point of each thimble were sampled to simulate the $(0+1)$-dimensional Thirring model, something much akin to the method of steepest descent. In practice the determination of the precise location of each thimble's saddle point, or critical point, as well as the relevant sampling "direction" about these points, is numerically costly and prohibitive. An alternative method is to train neural networks to learn the map from some starting manifold to any beneficial manifold, including one that approximates the thimbles that contribute to the integral [12–14].

In our previous work [14] we were limited by the computational cost of incorporating the Jacobian determinant of this map into our importance sampling. In this paper we leverage complex-valued neural networks built of affine coupling layers to reduce the scaling of the Jacobian determinant cost. We focus on the Hubbard model on a honeycomb lattice away from half filling and compare methods by computing single-particle correlation functions.

This paper is organized in the following way. In Sec. II, a brief recap of the Hubbard model and basic notation is given. After that, some prior methods to alleviate or remove the sign problem and usage within HMC are discussed. In Sec. III, we describe our neural network architecture. In Sec. IV, we show a numerical test of the network on three systems where we can exactly diagonalize the Hamiltonian, and one larger system beyond our ability to exactly diagonalize.

## II. FORMALISM

The Hubbard model [15] describes a fixed spatial lattice $X$ on which particles can move and interact. In the particle-hole basis it is described by Hamiltonian

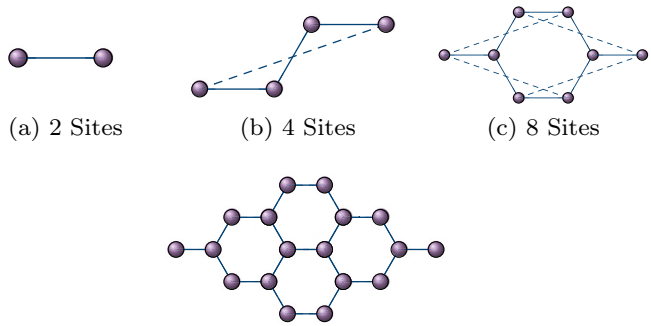
$$\mathcal{H}[K,V,\mu] = -\sum_{x,y\in X}\left(p_x^\dagger K^{xy} p_y - h_x^\dagger K^{xy} h_y\right) + \frac{1}{2}\sum_{x,y\in X}\rho_x V^{xy}\rho_y + \mu\sum_{x\in X}\rho_x, \tag{1}$$

where the amplitudes in $K$ encode the hopping of fermionic particles $p$ and holes $h$, the potential $V$ describes the interactions between charges

$$\rho_x = p_x^\dagger p_x - h_x^\dagger h_x, \tag{2}$$

and the chemical potential $\mu$ incentivizes charge. By adjusting $K$ and $V$ this model can describe a wide variety of physical systems. We restrict our attention to the case where $K$ encodes



(a) 2 Sites (b) 4 Sites (c) 8 Sites

(d) 18 Sites (boundary suppressed)

FIG. 1. (a)–(d) Graphical representation of the arrangement of ions considered in the numerical investigation. Each node corresponds to an ion, while each edge indicates an allowed particle or hole hopping. The dashed lines represent the periodic boundary.

a honeycomb structure with nearest-neighbor hopping and the interaction $V$ is local,

$$K = \kappa\delta_{\langle xy\rangle}, \quad V = U\delta_{xy}; \tag{3}$$

the bipartiteness of the honeycomb permits a signed sublattice transformation that flips the sign of the hopping of holes. As we are focusing on algorithmic issues we focus on only the four systems displayed in Fig. 1. These—the 2-, 4-, 8-, and 18-site models—are examples of the honeycomb lattice with periodic boundary conditions.

Our aim is to compute observables $\mathcal{O}$ according to the thermal trace

$$\langle\mathcal{O}\rangle = \frac{1}{\mathcal{Z}}\mathrm{Tr}[\mathcal{O}e^{-\beta H}], \tag{4}$$

where the partition function $\mathcal{Z}$ is the trace without the observable and $\beta$ is the inverse temperature, the Euclidean time extent. Trotterizing into $N_t$ time slices, inserting Grassmannian resolutions of the identity, and linearizing the interaction via the Hubbard-Stratonovich transformation [16] leads to the action

$$S[\Phi\,|\,K,V,\mu] = -\ln\det M[\Phi\,|\,K,\mu]\cdot M[-\Phi|\,-K,-\mu] + \frac{1}{2}\sum_t\sum_{x,y\in X}\Phi_{tx}(\delta V^{-1})^{xy}\Phi_{ty}, \tag{5}$$

where $\Phi\in\mathbb{R}^{|\Lambda|}$ is an auxiliary field on the space-time lattice $\Lambda = [0, N_t - 1]\otimes X$ and $\delta = \beta/N_t$. We use the exponential discretization [17] for the fermion matrices

$$M[\Phi\,|\,K,\mu]_{x't',xt} = \delta_{x'x}\delta_{t't} - (e^{\delta(K+\mu)})_{x'x}e^{+i\Phi_{xt}}\mathcal{B}_{t'}\delta_{t'(t+1)}, \tag{6}$$

where $\mathcal{B}$ encodes the antiperiodic boundary conditions in time. On a bipartite lattice we may replace the $-K$ in the holes' fermion matrix with $+K$; then when $\mu = 0$, the determinant may be made manifestly positive semidefinite. When $\mu$ is finite, $S$ is complex; a great deal of recent effort has been made in the computational physics community to understand this case [18–21].

The transformation of the thermal average (4) leads to the path integral

$$\langle\mathcal{O}\rangle = \frac{1}{\mathcal{Z}}\int\mathcal{D}\Phi\; e^{-\beta S[\Phi]}\mathcal{O}[\Phi] \equiv \int\mathcal{D}\Phi\; p_S[\Phi]\mathcal{O}[\Phi], \tag{7}$$

where the partition function $\mathcal{Z}$ is the integral without the observable $\mathcal{O}$. When the action is real, importance-sampling methods draw $N_{\mathrm{conf}}$ configurations according to the Boltzmann distribution

$$p_S[\Phi] = \frac{1}{\mathcal{Z}}e^{-S[\Phi]} \tag{8}$$

and estimate observables (7) by an unweighted average. Any practical calculation samples only finitely many configurations $N_{\mathrm{conf}}$, and the resulting statistical uncertainties scale as $N_{\mathrm{conf}}^{-1/2}$ as long as the configurations are independent.

At finite $\mu$ a complex-valued action yields an oscillating integrand, and $p_S$ (8) can no longer be interpreted as a standard probability density, rendering a straightforward application of importance sampling impossible.

To recover an importance-sampling algorithm, we can separate the real and imaginary parts of the action $S = \mathrm{Re}S + i\,\mathrm{Im}S$ and rewrite the partition function

$$\mathcal{Z} = \int\mathcal{D}\Phi\; e^{-S} = \int\mathcal{D}\Phi\; e^{-\mathrm{Re}S}e^{-i\,\mathrm{Im}S} \propto \langle e^{-i\,\mathrm{Im}S}\rangle_{\mathrm{Re}S} \equiv \Sigma, \tag{9}$$

where the expectation value is with respect to the real part of the action and we call $\Sigma$ the statistical power. So, by sampling according to $p_{\mathrm{Re}S}$ we can estimate

$$\langle\mathcal{O}\rangle = \frac{\langle e^{-i\,\mathrm{Im}S}\mathcal{O}\rangle_{\mathrm{Re}S}}{\langle e^{-i\,\mathrm{Im}S}\rangle_{\mathrm{Re}S}} = \frac{1}{\Sigma}\langle e^{-i\,\mathrm{Im}S}\mathcal{O}\rangle_{\mathrm{Re}S}. \tag{10}$$

When the statistical power $\Sigma$ (9) cannot be reliably distinguished from zero the sign problem is too strong, and the whole procedure fails [6,14,22,23]. Reference [22] showed that the effective number of configurations

$$N_{\mathrm{conf}}^{\mathrm{eff}} = |\Sigma|^2\cdot N_{\mathrm{conf}} \tag{11}$$

controls the scaling of statistical errors $\sim(N_{\mathrm{conf}}^{\mathrm{eff}})^{-1/2}$.

It is widely expected that the statistical power shrinks exponentially with space-time volume $\beta|X|$. Because the power is the ratio of the full and phase-quenched partition functions, it should be exponential in a difference of free energies, which is extensive in the space-time volume [24]. For small nonbipartite examples we have previously confirmed the exponential dependence on $\beta$ [14].

A promising alternative to simple reweighting is to complexify the domain of integration and transform $\phi\in\mathcal{M}_{\mathbb{R}} = \mathbb{R}^{|\Lambda|}$ to a manifold $\Phi\in\mathcal{M}\subset\mathbb{C}^{|\Lambda|}$. As long as $\mathcal{M}$ is in the same homology class, the analog of the Cauchy integral theorem ensures that the partition function is unchanged [2],

$$\mathcal{Z} = \int_{\mathcal{M}}\mathcal{D}\Phi\; e^{-S[\Phi]}. \tag{12}$$

Parametrizing the manifold $\mathcal{M}$ by the real fields induces a Jacobian determinant, yielding [6]

$$\mathcal{Z} = \int_{\mathcal{M}_{\mathbb{R}}}\mathcal{D}\phi\; e^{-S[\Phi(\phi)]+\ln\det J[\Phi(\phi)]}, \tag{13}$$

and observables are computed on the manifold $\mathcal{O}[\Phi(\phi)]$.

A judicious choice of the manifold $\mathcal{M}$ can diminish or completely remove the sign problem [2,25]. Even when sampling according to $p_{\mathrm{Re}S^{\mathrm{eff}}}$ with an imperfect manifold with a complex effective action

$$S^{\mathrm{eff}}[\phi] = S[\Phi(\phi)] - \ln\det J[\Phi(\phi)], \quad J_{ij} = \frac{\partial \Phi_i}{\partial \phi_j}, \tag{14}$$

if the statistical power $\Sigma$ (9) is sufficiently improved, we can reweight (10) with the imaginary part $\mathrm{Im}S^{\mathrm{eff}}$.

There are many strategies for picking target manifolds [26]. One choice is to try to approximate the Lefschetz thimbles—high-dimensional manifolds analogous to contours of steepest descent, which have constant imaginary action and therefore have a much-reduced sign problem [25]. Each thimble contains a critical point $\Phi_c$ that satisfies

$$\left.\frac{\partial S[\Phi]}{\partial \Phi}\right|_{\Phi=\Phi_c} = 0 \tag{15}$$

and is therefore a fixed point of the holomorphic flow

$$\frac{d\Phi(\tau)}{d\tau} = \left(\frac{\partial S[\Phi(\tau)]}{\partial \Phi(\tau)}\right)^* \tag{16}$$

as a function of the fictitious flow time $\tau$ and initial condition $\Phi(0) = \phi$. We can trace trajectories under the flow using the integrator

$$\mathcal{J}_\tau^{\pm}[\phi] \equiv \int_0^{\pm\tau} \left(\frac{\partial S[\Phi(\tau_f)]}{\partial \Phi(\tau_f)}\right)^* d\tau_f. \tag{17}$$

A thimble is the set of complexified configurations that flow to a critical point under downward flow $\mathcal{J}_\infty^-$.

There may be many thimbles in $\mathbb{C}^{|\Lambda|}$, and only some might contribute. The upward flow $\mathcal{J}_\infty^+$ discovers these thimbles automatically. After enough flow time $\tau$ the integrator $\mathcal{J}_\tau^+$ drives any $\Phi(0)$ either to a place on a thimble or to neverland—any place where thimbles of different imaginary action meet and therefore must have zero weight. When $\Phi$ starts on a valid integration manifold, its image under $\mathcal{J}_\infty^+$ is on a thimble that contributes to the integral or is in neverland. For an approachable discussion and proof, see the recent review in Ref. [2].

Therefore we can try to evaluate the path integral (13) on the manifold given by $\Phi(\phi) = \mathcal{J}_\infty^+[\phi]$ for each $\phi$ on any valid starting manifold $\mathcal{M}_0$, such as $\mathcal{M}_{\mathbb{R}}$. Though this seems to make sign-problem-free simulations possible, two issues remain. While integrating the flow (17) is cheap, performing molecular dynamics integration on the thimbles at first glance involves the costly computation of the Hessian $\partial_{\Phi_i}\partial_{\Phi_j} S[\Phi]$ due to the appearance of the Jacobian determinant of the flow in the effective action (14), though some ideas for quickly estimating the Jacobian have been proposed [27] and recent work [28] shows how to accelerate this for sparse, local (bosonic) actions. The Jacobian determinant has to be evaluated at any accept-reject step with computational cost scaling as $|\Lambda|^3$.

Second, because thimbles only touch at places of zero weight, algorithms such as HMC [29] which use a smooth update of the fields $\Phi$ would be encumbered by an ergodicity problem. The severity of this issue is ameliorated in two ways. As any practical integrator $\mathcal{J}_\tau^+$ necessarily approximates the flow, the resulting integration manifold is only approximately the union of contributing thimbles. Additionally, we do not need to flow for very much time. Both of these mean that the important configurations are smoothly connected, though the imaginary part of the action is not perfectly piecewise constant. In practice, picking a $\tau$ is a trade-off between reducing the computational cost of the flow and an improvement of the statistical power.

The cost of the flow and the associated Jacobian determinant is such that it is beneficial to train a neural network to learn the map $\mathcal{J}_\tau^+ : \mathcal{M}_0 \to \tilde{\mathcal{M}}$. In the next section we explain our network's architecture.

Of course, understanding neural networks as general function approximators yields an interpretation of any (numerical) integrator as a network, though it is parameter-free and needs no training—its layers, given by some discretization of the flow equations (16), are exactly known. Just as we can produce training configurations closer to the thimbles with a more precise integrator, by adding additional layers we may train the network to reproduce the integrated flow more accurately. So, one expects a trade-off between the nearness to the thimbles (thinking of the number of layers as a proxy) and the effort required to train. The algorithm we describe is exact, even in the case where the network does not offer an acceleration, since the network produces a manifold with the correct homology class regardless of its fidelity to the thimbles.

Because we can integrate on any manifold in the same homology class as $\mathbb{R}^{|\Lambda|}$, it may be beneficial to find simple manifolds that can improve the statistical power without the computational cost of flowing [30,31]. One such manifold is the tangent (hyper)plane $\Phi \in \mathcal{M}_T$ [2,6,14], a hyperplane parallel to the real manifold offset by a constant imaginary piece so that it intersects the critical-point image of the zero configuration $i\Phi_c^0 = \mathcal{J}_\infty^+(0)$

$$\Phi(\phi) = \phi + i\Phi_c^0 \tag{18}$$

for all $\phi \in \mathcal{M}_{\mathbb{R}}$. For many smaller systems this transformation already reduces the sign problem enough that reweighting can be applied. However, in our larger examples the tangent plane gives no appreciable statistical power. Nevertheless, we can reduce the cost and potentially increase the potency of flowing if we start from the tangent plane [14].

One obvious approach to constructing an HMC-like algorithm is to attempt molecular dynamics on the target manifold $\tilde{\mathcal{M}}$ given by $\tilde{\Phi}$: in our case, an approximation of the thimbles. However, remaining on the manifold is not so simple [28,32–34].

In contrast, performing HMC on the tangent plane *is* simple: When integrating molecular dynamics trajectories, simply neglect the imaginary part of the force. Because the real plane suffers from a severe sign problem in the examples we study, we use this tangent-plane HMC as a benchmark. In the remainder of this paper we refer to it simply as "HMC" unless clarification is needed.

For further improvement we do molecular dynamics on the tangent plane $\mathcal{M}_T$ and perform the Metropolis-Hastings accept-reject step on the target manifold $\tilde{\mathcal{M}}$ according to the effective action (14). We track both the configuration on the integration manifold $\mathcal{M}_0$ and its image on the target manifold $\tilde{\mathcal{M}}$ to avoid paying the computational cost of applying or

inverting the transformation $\tilde{\Phi}$ more than needed. Assuming the numerical implementation of the map $\tilde{\Phi}$ is invertible, proof that this algorithm has detailed balance is provided in Ref. [14]. One can use a reversible integrator or an invertible neural network to satisfy this requirement.

## III. MACHINE-LEARNING METHOD

To accelerate the transformation to the target manifold $\tilde{\mathcal{M}}$, reducing computational complexity, it is possible to define a neural network trained to approximate the integrator (17) $\mathcal{NN} \approx \mathfrak{I}_\tau^+$.

One approach is to learn the imaginary part of any configuration on the target manifold $\tilde{\mathcal{M}}$ given its real part [12,14]

$$\text{shift} : \mathcal{M}_0 \to \tilde{\mathcal{M}}, \quad \Phi \mapsto \Phi + iNN(\text{Re}\Phi). \tag{19}$$

This ansatz has two advantages. First, the ergodicity issue, induced by potential trapping on individual thimbles, is removed [12]. Second, the network can use the well-established methods of real-valued neural networks. Computational costs due to flowing are reduced as the application of the neural network is much cheaper then any numerical integration. However, a major disadvantage is the computational effort and severe volume scaling of the Jacobian determinant [14].

In this paper we use complex-valued neural networks—networks with complex parameters—to instead learn the map from the integration manifold $\mathcal{M}_0$ to the target manifold $\tilde{M}$,

$$\tilde{\Phi} = \mathcal{NN}(\phi) \approx \mathfrak{I}_\tau^+(\phi). \tag{20}$$

This approach enjoys a significant advantage over the shift network (19): Given the right network architecture, the Jacobian may be evaluated very quickly. Below we will explain our use of affine coupling layers to reduce the scaling of the Jacobian determinant from a general cubic scaling down to a linear scaling in the volume $|\Lambda|$.

For a recent overview of complex-valued networks, see Ref. [35]. Typical automatic differentiation algorithms can be applied to complex-valued neural networks in a manner similar to that in which real-valued ones are applied [36–38] by switching the differentiation rule to Wirtinger derivatives [36]

$$\begin{aligned}\frac{\partial f(z)}{\partial z} &= \frac{1}{2}\left(\frac{\partial f(z)}{\partial\, \text{Re}z} - i\frac{\partial f(z)}{\partial\, \text{Im}z}\right),\\ \frac{\partial f(z)}{\partial z^*} &= \frac{1}{2}\left(\frac{\partial f(z)}{\partial\, \text{Re}z} + i\frac{\partial f(z)}{\partial\, \text{Im}z}\right).\end{aligned} \tag{21}$$

The Wirtinger derivatives have the advantage that they coincide with complex derivatives for holomorphic functions while also extending to nonholomorphic ones. This generalization is required for two reasons. First, loss functions typically are not holomorphic and are not differentiable in the complex sense. Second, Liouville's theorem, stating that bounded entire functions are constant, reduces the usability of any complex-valued neural network if only holomorphic components can be used. As automatic differentiation is possible through backpropagation using Wirtinger derivatives, these restrictions can be overcome, and a neural network $\mathcal{NN} : \mathbb{C}^m \to \mathbb{C}^n$ with complex-valued weights can be defined [35]. We want to emphasize that a nonholomorphic network can approximate the thimbles even though their definition is manifestly holomorphic. This can be understood by utilizing the universal approximation theorem [39] and realizing that the change of variable requires an embedding which is at least twice differentiable in the Wirtinger sense. It is expected that such networks have an improved expressivity compared with real-valued networks of twice the size—mimicking the real and imaginary parts—as complex networks do not have to learn complex arithmetic [35].

Special care has to be taken when evaluating the Jacobian induced by the parametrization of $\tilde{M}$. The Jacobian in the effective action (14) is defined by the derivative of the transformation according to its real parameters—a derivative in the real sense. When applying a nonholomorphic neural network to parametrize the manifold, the Wirtinger derivatives force us to reexpress the derivative in the real sense by combining the two equations of (21) and the transformation on the tangent plane (18)

$$J_{ij} \equiv \frac{\partial \mathcal{NN}\left(\phi + i\Phi_c^0\right)_i}{\partial \phi_j} = \frac{\partial \mathcal{NN}(\Phi)_i}{\partial \Phi_j} + \frac{\partial \mathcal{NN}(\Phi)_i}{\partial \Phi_j^*}. \tag{22}$$

To identify an architecture with an efficiently computable Jacobian determinant, split the network into $L$ constituent layers:

$$\begin{aligned}\Phi_0(\phi) &= \phi + i\Phi_c^0,\\ \Phi_{\ell>0}(\phi) &= \mathcal{NN}_\ell(\Phi_{\ell-1}(\phi))\\ &= (\mathcal{NN}_\ell \circ \mathcal{NN}_{\ell-1} \circ \cdots \circ \mathcal{NN}_1)(\phi),\\ \tilde{\Phi}(\phi) = \Phi_L(\phi) &= \mathcal{NN}_L(\phi) \equiv \mathcal{NN}(\phi).\end{aligned} \tag{23}$$

The Jacobian determinant of the neural network [40] is then given as the product of the Jacobian determinants of each layer

$$\det J = \prod_{\ell=1}^{L} \det J_{\mathcal{NN}_\ell}. \tag{24}$$

Consequently, we focus on layers with computationally simple Jacobian determinants. Coupling layers

$$\mathcal{NN}_\ell(\Phi) = \begin{cases} c_\ell[\Phi_A,\ \Phi_B] & A_\ell \text{ components} \\ \Phi_B & B_\ell \text{ components}\end{cases} \tag{25}$$

fulfill this requirement [41,42]. Here, $A$ and $B$ are layer-specific partitions of the input vector $\Phi$ of equal cardinality $\frac{1}{2}|\Lambda|$, and $\Phi_{A,B}$ are the components of the input belonging to the indicated partition. If the coupling layer $c_\ell[\Phi_A, \Phi_B]$ acts in an elementwise manner and is holomorphic in the components $\Phi_A$

$$\frac{\partial c_\ell[\Phi_A, \Phi_B]}{\partial \Phi_A^*} = 0, \tag{26}$$

the Jacobian determinant of each layer is given by

$$\det J_{\mathcal{NN}_\ell}(\Phi) = \prod_{i=0}^{|A|-1} \frac{\partial c_\ell[\Phi_A, \Phi_B]}{\partial (\Phi_A)_i}. \tag{27}$$

Furthermore, using an affine coupling [41]

$$c_\ell[\Phi_A, \Phi_B] = e^{m_\ell(\Phi_B)} \odot \Phi_A + a_\ell(\Phi_B), \tag{28}$$

with arbitrary differentiable functions $m_\ell, a_\ell : \mathbb{C}^{|\Lambda|/2} \to \mathbb{C}^{|\Lambda|/2}$ acting on the $B$ indices of the input configuration $\Phi$, yields a

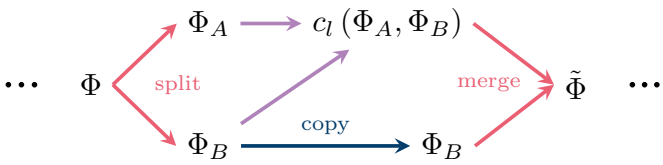



FIG. 2. Pictorial representation of one coupling layer (25). First the input configuration $\Phi$ is split into two partitions, $\Phi_A$ and $\Phi_B$. The corresponding $A$ components are then changed according to the prescribed coupling $c_l(\Phi_A, \Phi_B)$, while the $B$ components are untouched. We utilize an affine transformation (28) for the coupling $c_l$. The resulting output vector $\tilde{\Phi}$ is then constructed from the transformed $A$ components and unchanged $B$ components.

computationally cheap (log) Jacobian determinant

$$\ln\det J_{\mathcal{NN}}(\phi) = \sum_{\ell=1}^{L}\sum_{i=0}^{|A|-1} m_\ell(\Phi_{\ell-1}(\phi)_B)_i. \tag{29}$$

The expressivity of the neural network is controlled by the trainable parameters in the coupling functions $m_\ell, a_\ell$. If $f$ denotes an affine transformation

$$f(\Phi) = \omega\cdot\Phi + b \tag{30}$$

and $g$ denotes the nonlinear "softsign" function

$$g(z) = \frac{z}{1+|z|}, \tag{31}$$

we take the coupling functions to be

$$a_\ell, m_\ell = g\circ f\circ g\circ f \tag{32}$$

with independent complex weight matrices $\omega$ and bias vectors $b$. The softsign function is nonholomorphic, requiring us to consider the Jacobian in the Wirtinger sense (22). Due to the structure of the Jacobian matrix, the nonzero nonholomorphic components $\partial_{\Phi_B^*} c_\ell$ do not contribute to the determinant (29). A graphical representation of this architecture is displayed in Fig. 2.

We add layers in pairs so that $L$ is even. Each pair shares their partitioning. In each pair the first layer modifies the $A$ partition (25) and the next modifies the $B$ partition using the same ansatz with independent weights and biases. Notice that the Jacobian determinant can be implemented so that it is evaluated during the forward pass [41], which reduces the required additional cost to only the sums of Eq. (29). Consequently, the Jacobian determinant in the effective action (14) only adds a computational complexity linear in the volume $|\Lambda|$.

The training setup was kept simple, allowing for further improvements in the future. A standard $L_1$ loss function and the ADAM algorithm implemented in PYTORCH [43] were used to train the network. We kept the ADAM-specific hyperparameters—running average coefficients $\beta_i = (0.9, 0.999)$, denominator shift $\varepsilon = 1\times10^{-8}$, and weight decay $w = 0$—at the standard values. The training data comprised 10 000 (16 000 for the 18-sites problem) configurations drawn from normal distributions $\phi \sim \mathcal{N}_{0,\sigma}$, with $\sigma$ uniformly sampled between $\sqrt{U/(1+16/N_t)}$ and $\sqrt{U}$ [14], as input. The "labels" consist of the corresponding flowed configurations $\mathfrak{I}_\tau^+(\phi)$, where the integration is performed using an adaptive Runge-Kutta method of fourth order. A similar setup is used for the validation and testing data but only for 2000 configurations each. To avoid learning features of the thimbles irrelevant to the integral [6,10,14,23], only configurations that did not flow to neverland are included in the training.

The network $\mathcal{NN}$ with two pairs of coupling layers was initialized to the identity so that before training it reproduced the tangent-plane configurations which were fed into it. We experimented with learning $\mathfrak{I}_\tau^+$ different flow times $\tau \in \{1\times10^{-6}, 1\times10^{-5}, 1\times10^{-3}, 1\times10^{-2}, 1\times10^{-1}\}$. We computed both the statistical power and measured correlators (as shown later in Fig. 4). If we flow too much, most configurations flow to neverland, and training becomes expensive; if we flow too little, the statistical power hardly improves. The results shown in the next section have a flow time $\tau = 1\times10^{-1}$.

Unfortunately picking a fixed flow time of this size was not feasible for the 18-sites problem. Instead, we defined a window of flow times $\tau \in [0, 0.1]$ on which the flow is performed, as was originally done in Ref. [14]. In this manner, fixed flow-time configurations which would have flowed to neverland and thus have been rejected could still be used if their configurations remained valid within the window of flow times. It was found in Ref. [14] that this method greatly decreased the cost of generating training data. In future work we will continue to investigate more efficient ways of generating training data, and performing the training process itself, including by sampling one training point from the steps along a holomorphic flow to $\tau = \infty$ according to the real part of the step's action.

## IV. RESULTS

We simulate the Hubbard model on the honeycomb lattices of 2, 4, 8, and 18 sites shown in Fig. 1, using configurations obtained on the tangent plane and via our neural network $\mathcal{NN}$, at inverse temperature $\beta = 4$, $N_t = 32$ time slices, on-site coupling $U = 4$, and chemical potential $\mu = 3$. To compare the machine-learning-enhanced (ML) HMC with other implementations such as the real-plane (standard) HMC with molecular dynamics on $\mathcal{M}_\mathbb{R}$ and the tangent-plane HMC on $\mathcal{M}_T$, we consider the statistical power $\Sigma$. A suitable algorithm will have $|\Sigma|$ close to 1, whereas low values indicate a less suitable algorithm, since considerably more statistics would be required (11). Figure 3 shows estimates of $|\Sigma|$ with different numbers of configurations for the three mentioned HMC variants. The ML HMC is shown in blue, the tangent-plane HMC is shown in orange, and the real-plane HMC is shown in red. The ML HMC outperforms the two other algorithms in every case. Moreover, in the case of eight sites, enormous statistics are required to even get a reasonable estimate of the statistical power for the real- and tangent-plane HMCs, while the power of the ML HMC stabilizes with far fewer samples. For 18 sites it was not feasible to simulate with the real-plane HMC, and thus it is not shown here. We can see that the tangent-plane HMC does not get any reliable value for the statistical power while the ML HMC converges relatively fast.

We show the efficacy of our method by computing Euclidean-time correlators for a single particle or single hole


FIG. 3. The statistical power $|\Sigma| = |\langle e^{-i\,\mathrm{Im}S}\rangle|$ is plotted against the number of configurations. Three different algorithms are compared: the real-plane (standard) HMC (orange), the tangent-plane HMC (red), and the ML HMC (blue). For 18 sites the real plane was totally noisy, and it is left out here. It can be seen that the ML HMC outperforms both real- and tangent-plane HMCs.

created at time 0 and site $y$ and destroyed at time $t$ and site $x$.

$$C^p_{xy}(t) = \langle p^\dagger_x(t) p_y(0)\rangle = \big\langle M[+\Phi| + \mu]^{-1}_{xt;y0}\big\rangle, \tag{33}$$

$$C^h_{xy}(t) = \langle h^\dagger_x(t) h_y(0)\rangle = \big\langle M[-\Phi| - \mu]^{-1}_{xt;y0}\big\rangle. \tag{34}$$

To improve our signal, we average on time slices in $t \in [\delta, \beta - \delta]$,

$$C_{xy}(t) = \tfrac{1}{2}\big(C^p_{xy}(t) + C^{h*}_{xy}(\beta - t)\big); \tag{35}$$

the addends are equal by symmetry even when $\mu \neq 0$. We then project both spatial indices to the same momentum $k$ to construct $C_k(t)$ for each momentum allowed by the lattice, and average correlators whose momenta are equal by rotational symmetry.

The match of our correlators in Fig. 4 with the exact results demonstrates that our algorithm is sampling the correct distribution. Each row of the figure corresponds to one of the exactly diagonalizable system sizes, and each column restricts the number of configurations $N_{\mathrm{conf}}$ used to estimate the correlators. The red correlators are determined using a tangent-plane HMC, and the blue ones are determined using the ML HMC. Finally, the black dashed lines correspond to the correlators obtained by an exact diagonalization procedure. For the smaller examples the statistical errors of the ML HMC are much smaller, especially with fewer samples, as is expected from their respective statistical powers shown in Fig. 3. The worst sign problem can be found in the eight-site case. Here, the tangent-plane HMC fails even for $N_{\mathrm{conf}} = 100\,000$ and the statistical uncertainty in the correlators is essentially 100%. The ML HMC obtains a weak signal at $N_{\mathrm{conf}} = 50\,000$ configurations and improves with greater statistics.

Finally, we compute correlators for a system with 18 sites and the same parameters but with $U = 3$ which is not tractable by exact diagonalization. As shown in the statistical power plot in Fig. 3 this model has a severe sign problem which could not be previously overcome. Again comparing the tangent-plane HMC and ML HMC in Fig. 5, it can be seen that the ML HMC outperforms the tangent-plane HMC and with the 100 000 measurements quite a good signal is obtained.

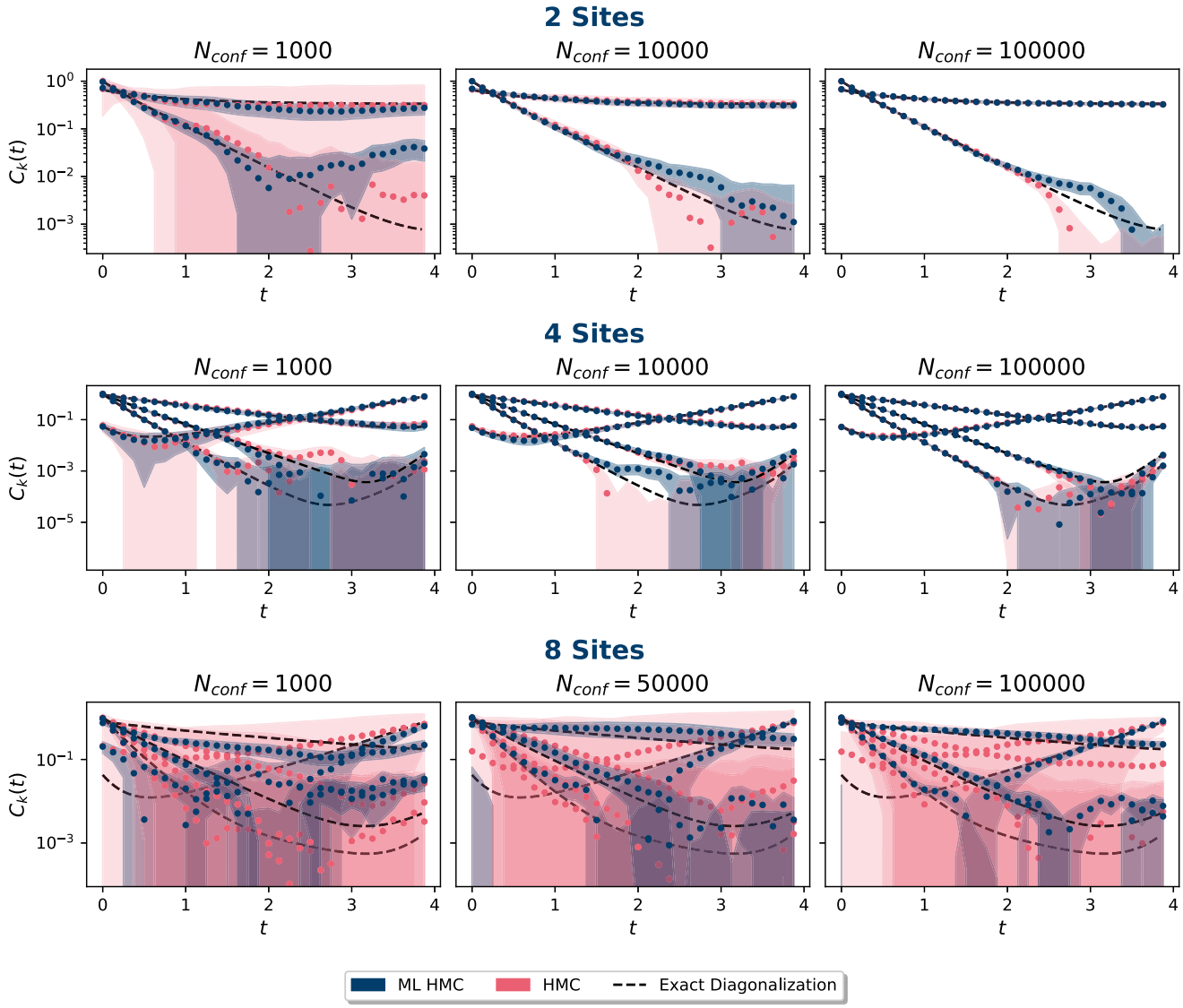



FIG. 4. Momentum-projected correlation functions measured using tangent-plane HMC and ML HMC are shown in red and blue, respectively. These correlators were calculated with an inverse temperature $\beta = 4$, $N_t = 32$ time slices, on-site coupling $U = 4$, and chemical potential $\mu = 3$. The dashed black lines were determined by exact diagonalization. Each row corresponds to a different number of ions (as in Fig. 1) increasing from top to bottom. Each column uses a different number of configurations $N_{\text{conf}}$ to estimate the correlators, increasing from left to right. Comparing the statistical power per $N_{\text{conf}}$ from Fig. 3 suggests that we use $N_{\text{conf}} = 1000, 10\,000, 100\,000$ for two and four sites as the uncertainty strongly differs. However, for eight sites there is not much difference in the uncertainty of $|\Sigma|$ between $N_{\text{conf}} = 1000$ and $10\,000$; we show $N_{\text{conf}} = 1000, 50\,000, 100\,000$ instead. The ML HMC's improved statistical power shown in Fig. 3 is reflected in the accuracy and uncertainty of these correlators. The sign problem of the two-site and four-site models is mild enough such that the tangent plane gives fairly good results, but the ML HMC gives more precise results with fewer configurations. For eight sites the tangent-plane HMC completely fails even at $N_{\text{conf}} = 100\,000$, while the ML HMC succeeds.

In all cases we measured on every tenth configuration such that no appreciable autocorrelation is found. All these simulations indicate that the neural network improves the statistical power and uncertainty in observables quite drastically even when using a simple architecture. We anticipate further improvements of our network by incorporating additional layers or incorporating knowledge of the problem's symmetries using equivariant layers [44–46].

The main advantage of our complex architecture lies in the efficiency of the Jacobian determinant (29) calculation. The form of the determinant (29) shows that it can be computed during the forward pass, reusing intermediate results from the application of the network, and is linear in the volume $|\Lambda|^{\alpha_{\mathcal{NN}}}$

$$\alpha_{\mathcal{NN}} = 1. \tag{36}$$

The calculation of the determinant using a shift layer [12,14] with the implementation of PYTORCH [43] [through lower-upper (LU) decomposition] scales with the third power of the volume $|\Lambda|^{\alpha_{\text{shift}}}$, i.e., $\alpha_{\text{shift}} = 3$. Measurements

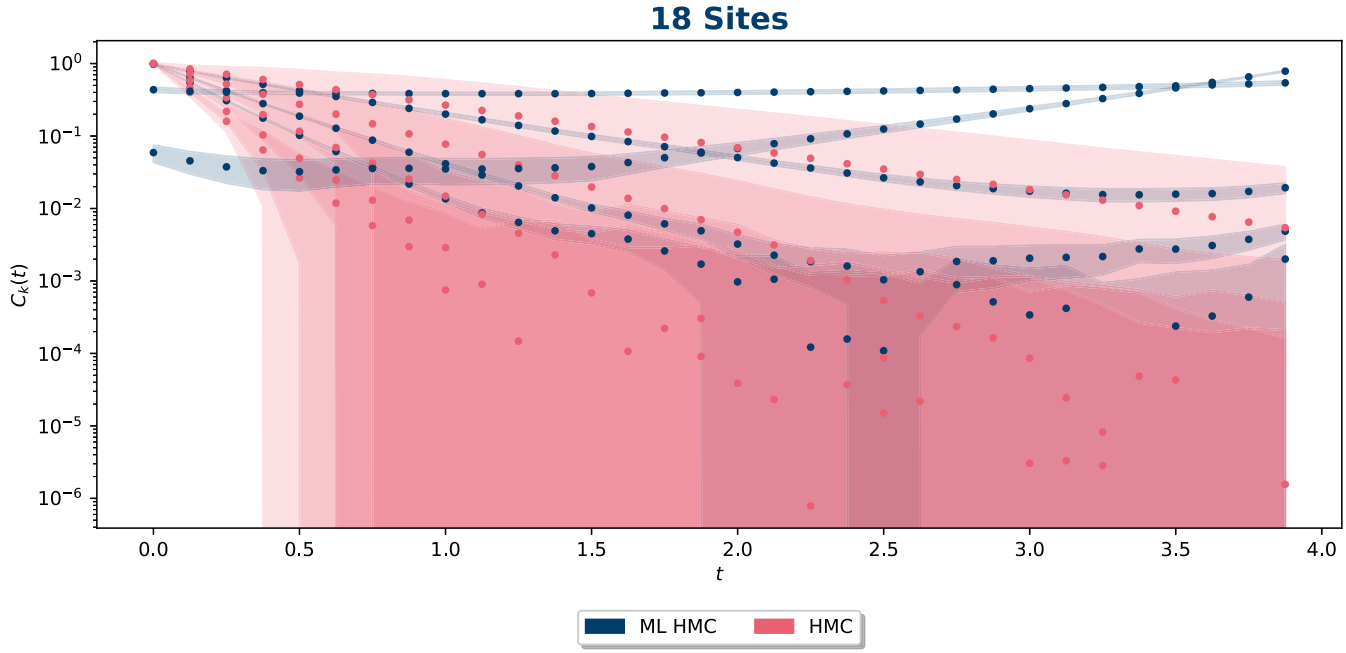



FIG. 5. The single-particle correlators are displayed for the 18-site model at an inverse temperature $\beta = 4$, with $N_t = 32$ the number of time slices and with $U = 3$ and $\mu = 3$. The correlators have been measured with $N_{\text{conf}} = 100\,000$ configurations. Again, the tangent-plane HMC (red) does not provide any insight, while the ML HMC resolves the correlators well. Assuming similarity to the smaller $U = 4$ examples, the ML HMC clearly determines the low-energy correlator, while the tangent-plane HMC fails to find it at all.

of the execution times of the determinant for the two neural network architectures are compared in Fig. 6. The left panel shows the execution time per layer of $\ln\det J$ for different artificial system volumes. These volumes define the size of the configuration $\Phi$ which is randomly sampled and then passed to the networks. On the log-log plot the linear behavior in the region $|\Lambda| > 1 \times 10^7$—for $\alpha_{\mathcal{NN}}$—and $|\Lambda| > 7 \times 10^2$—for $\alpha_{\text{shift}}$—determines the algorithms'

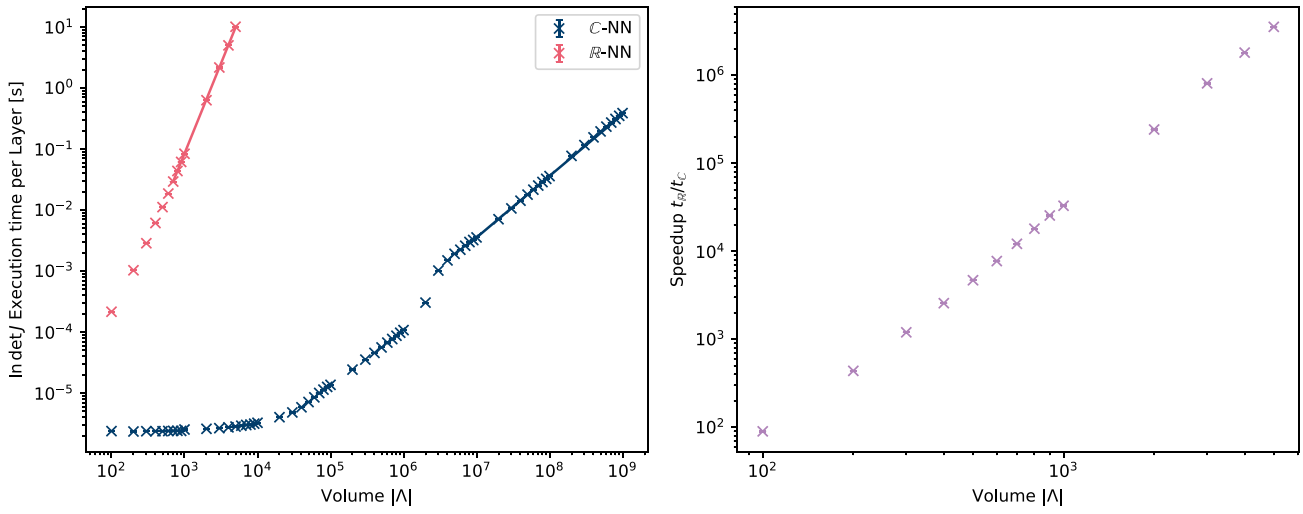



FIG. 6. The left panel shows the scaling behavior per layer of $\ln\det J$ for the previously used shift neural network, $\tilde{\Phi} = \Phi + iNN(\Phi)$ (red), and for the complex-valued paired affine coupling neural network, $\tilde{\Phi} = \mathcal{NN}(\Phi)$ (blue). In the right panel, we show the speedup for the different system volumes. Theoretically, the Jacobian determinant scaling of the shift network is expected to be cubic in the system volume, while the $\mathcal{NN}$ is expected to scale linearly, resulting in a quadratic speedup. The solid lines, in the left panel, represent log-log fits whose slopes determine the measured scaling orders. We find for the slopes of the shift layer (red) a value of 2.955(1) and for $\mathcal{NN}$ (blue) a value of 1.008(1), resulting in a scaling improvement of power 1.947(2). The timing measurements were performed on JURECA [47] using one AMD EPYC CPU.

scaling. A simple least-squares fit provides the scaling exponents

$$\alpha_{\text{shift}} = 2.955(1), \quad \alpha_{\mathcal{NN}} = 1.008(1), \tag{37}$$

confirming our expected scaling behavior. We then calculate the speedup achieved with the complex over the shift network architecture in the right panel of Fig. 6. The expected quadratic speedup is confirmed by the benchmark result of

$$\alpha_{\text{shift}} - \alpha_{\mathcal{NN}} = 1.947(2). \tag{38}$$

## V. CONCLUSIONS

Mitigating the sign problem induced by a complex action is a major target of algorithmic development for simulating quantum-mechanical systems. The application of neural networks approximating Lefschetz thimbles has shown great promise in the past. We show that the supervised training of a simple complex-valued neural network architecture—paired affine coupling layers with complex weights and biases—allows for the successful simulation of systems with increasingly severe sign problems. Our ML HMC approach reduces the sign problem sufficiently and enjoys a statistical power much greater than vanilla real-plane or tangent-plane HMC, as shown in Fig. 3, improving the reliability of the correlator estimators in Fig. 4. We demonstrated the fidelity and correctness of our method by simulating two-, four- and eight-site models and comparing our results with those obtained from direct diagonalization, obtaining excellent agreement. We then applied our method to the 18-sites problem, where direct diagonalization is not realizable. Our results here thus represent predictions for this system in a regime where standard Monte Carlo methods are not possible due to the severity of the sign problem.

Our results were made possible due to the favorable volume scaling of our method. Compared with previous methods, we drastically reduced the computational cost of the Jacobian determinant from a general cubic scaling down to linear in the volume. This has been numerically tested and demonstrated in Fig. 6. The computational complexity of our method is therefore dominated by the application of the neural network itself and can be further improved by using sparse methods, convolutional layers, or other layer architectures. We are actively investigating such possibilities.

## ACKNOWLEDGMENTS

We thank Jan-Lukas Wynen for many helpful discussions. This work was funded in part by the NSFC and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through the funds provided to the Sino-German Collaborative Research Center "Symmetries and the Emergence of Structure in QCD" (NSFC Grant No. 12070131001, DFG Project No. 196253076–TRR110) as well as STFC Consolidated Grant No. ST/T000988/1. M.R. was supported under RWTH Exploratory Research Space (ERS) Grant No. PF-JARA-SDS005. We gratefully acknowledge the computing time granted by the JARA Vergabegremium and provided on the JARA Partition part of the supercomputer JURECA at Forschungszentrum Jülich.

[1] K. Kashiwa, Y. Mori, and A. Ohnishi, Phys. Rev. D **99**, 014033 (2019).
[2] A. Alexandru, G. Başar, P. F. Bedaque, and N. C. Warrington, Rev. Mod. Phys. **94**, 015006 (2022).
[3] W. Detmold, G. Kanwar, M. L. Wagman, and N. C. Warrington, Phys. Rev. D **102**, 014514 (2020).
[4] W. Detmold, G. Kanwar, H. Lamm, M. L. Wagman, and N. C. Warrington, Phys. Rev. D **103**, 094517 (2021).
[5] S. Lefschetz, Trans. Am. Math. Soc. **22**, 327 (1921).
[6] A. Alexandru, G. Başar, and P. Bedaque, Phys. Rev. D **93**, 014504 (2016).
[7] M. Cristoforetti, F. Di Renzo, G. Eruzzi, A. Mukherjee, C. Schmidt, L. Scorzato, and C. Torrero, Phys. Rev. D **89**, 114505 (2014).
[8] M. Cristoforetti, F. Di Renzo, A. Mukherjee, and L. Scorzato, Phys. Rev. D **88**, 051501(R) (2013).
[9] A. Mukherjee, M. Cristoforetti, and L. Scorzato, Phys. Rev. D **88**, 051502(R) (2013).
[10] T. Kanazawa and Y. Tanizaki, J. High Energy Phys. 03 (2015) 044.
[11] Y. Tanizaki, Y. Hidaka, and T. Hayata, New J. Phys. **18**, 033002 (2016).
[12] A. Alexandru, P. F. Bedaque, H. Lamm, and S. Lawrence, Phys. Rev. D **96**, 094505 (2017).
[13] Y. Mori, K. Kashiwa, and A. Ohnishi, PTEP **2018**, 023B04 (2018).
[14] J.-L. Wynen, E. Berkowitz, S. Krieg, T. Luu, and J. Ostmeyer, Phys. Rev. B **103**, 125153 (2021).
[15] J. Hubbard, Proc. R. Soc. London, Ser. A **276**, 238 (1963).
[16] J. Hubbard, Phys. Rev. Lett. **3**, 77 (1959).
[17] J.-L. Wynen, E. Berkowitz, C. Körber, T. A. Lähde, and T. Luu, Phys. Rev. B **100**, 075141 (2019).
[18] A. Mukherjee and M. Cristoforetti, Phys. Rev. B **90**, 035134 (2014).
[19] M. Fukuma, N. Matsumoto, and N. Umeda, Phys. Rev. D **100**, 114510 (2019).
[20] M. Ulybyshev, C. Winterowd, and S. Zafeiropoulos, Phys. Rev. D **101**, 014508 (2020).
[21] M. Schneider, J. Ostmeyer, K. Jansen, T. Luu, and C. Urbach, Phys. Rev. B **104**, 155118 (2021).
[22] C. E. Berger, L. Rammelmüller, A. C. Loheac, F. Ehmann, J. Braun, and J. E. Drut, Phys. Rep. **892**, 1 (2021).
[23] Y. Mori, K. Kashiwa, and A. Ohnishi, Phys. Lett. B **781**, 688 (2018).
[24] K. Splittorff and J. J. M. Verbaarschot, Phys. Rev. Lett. **98**, 031601 (2007).

[25] M. Cristoforetti, F. Di Renzo, and L. Scorzato (AuroraScience Collaboration), Phys. Rev. D **86**, 074506 (2012).
[26] Y. Tanizaki, H. Nishimura, and J. J. M. Verbaarschot, J. High Energy Phys. 10 (2017) 100.
[27] A. Alexandru, G. Başar, P. F. Bedaque, G. W. Ridgway, and N. C. Warrington, Phys. Rev. D **93**, 094514 (2016).
[28] G. Fujisawa, J. Nishimura, K. Sakai, and A. Yosprakob, J. High Energy Res. 04 (2022) 179.
[29] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth, Phys. Lett. B **195**, 216 (1987).
[30] A. Alexandru, P. F. Bedaque, H. Lamm, S. Lawrence, and N. C. Warrington, Phys. Rev. Lett. **121**, 191602 (2018).
[31] N. C. Warrington, Taming the sign problem in lattice field theory with deformed path integral contours, Ph.D. thesis, University of Maryland, College Park, 2019.
[32] H. Fujii, D. Honda, M. Kato, Y. Kikukawa, S. Komatsu, and T. Sano, J. High Energy Phys. 10 (2013) 147.
[33] H. Fujii, S. Kamata, and Y. Kikukawa, J. High Energy Phys. 11 (2015) 07; **2016**, 36(E) (2016).
[34] M. Fukuma, N. Matsumoto, and N. Umeda, arXiv:1912.13303 [hep-lat].
[35] J. Bassey, L. Qian, and X. Li, arXiv:2101.12249 [stat.ML].
[36] P. Bouboulis, arXiv:1005.5170 [cs.LG].
[37] D. Brandwood, Opt. Antennas **130**, 11 (1983).
[38] K. Kreutz-Delgado, arXiv:0906.4835 [math.OC].
[39] F. Voigtlaender, arXiv:2012.03351 [math.FA].
[40] Note that this requires the input dimension and output dimension of each layer to be equal.
[41] M. S. Albergo, D. Boyda, D. C. Hackett, G. Kanwar, K. Cranmer, S. Racanière, D. J. Rezende, and P. E. Shanahan, arXiv:2101.08176 [hep-lat].
[42] S. Foreman, T. Izubuchi, L. Jin, X.-Y. Jin, J. C. Osborn, and A. Tomiya, arXiv:2112.01586 [cs.LG].
[43] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai *et al.*, in *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Red Hook, NY, 2019), pp. 8024–8035.
[44] M. Favoni, A. Ipp, D. I. Müller, and D. Schuh, Phys. Rev. Lett. **128**, 032003 (2022).
[45] D. Luo, G. Carleo, B. K. Clark, and J. Stokes, Phys. Rev. Lett. **127**, 276402 (2021).
[46] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan, Phys. Rev. Lett. **125**, 121601 (2020).
[47] D. Krause and P. Thörnig, J. Large-Scale Res. Facil. **4**, A132 (2018).