






Autoencoder-based analytic continuation method for strongly correlated quantum systems

Maksymilian Kliczkowski ¹, Lauren Keyes ², Sayantan Roy,² Thereza Paiva ³, Mohit Randeria,² Nandini Trivedi,² and Maciej M. Maška ¹

¹*Institute of Theoretical Physics, Wrocław University of Science and Technology, 50-370 Wrocław, Poland*

²*Department of Physics, The Ohio State University, Columbus, Ohio 43210, USA*

³*Instituto de Física, Universidade Federal do Rio de Janeiro, 21941-972 Rio de Janeiro RJ, Brazil*

 (Received 28 February 2024; revised 28 July 2024; accepted 27 August 2024; published 11 September 2024)

Solving ill-posed problems is central to a variety of scientific investigations. We focus here on the analytic continuation of imaginary-time data obtained from quantum Monte Carlo (QMC) simulations to the real frequency axis, which involves the numerical inversion of a Laplace transform, a well-known ill-posed problem. We propose an unsupervised autoencoder-type neural network to address this problem, and we show that our encoder-decoder approach can extract high-quality real frequency spectral functions from imaginary-time Green's functions $G(\tau)$. With a deeply tunable architecture we demonstrate, for artificial test data with noise added, that the autoencoder neural network can locate sharp features of spectral functions, which may be lost using maximum entropy (MaxEnt) methods currently in use. We demonstrate the strength of the autoencoder approach by applying it to QMC results for a single-band Hubbard model as a function of density, and we show that it is more robust against noise in the input $G(\tau)$ compared to MaxEnt. The proposed method is general and can also be applied to other ill-posed inverse problems.

DOI: [10.1103/PhysRevB.110.115119](https://doi.org/10.1103/PhysRevB.110.115119)

I. INTRODUCTION

Solving ill-posed inverse problems is critical across scientific disciplines, where the reconstruction of functions from indirect or noisy observations presents inherent challenges [1–3]. Prominent ill-posed inverse problems include, e.g., recovering the signal from convoluted or blurred versions [4–7], reconstructing a function from its Laplace transform [8–11], determining object properties from scattered waves [12,13], finding the initial temperature distribution within a material based on temperature measurements at its surface [14], or reconstructing a function from a finite set of data. In this paper, we propose an autoencoder-type neural network as a universal tool for solving ill-posed inverse problems. To demonstrate its effectiveness, we apply it to perform analytic continuation, transforming imaginary-time quantum Monte Carlo (QMC) data to real frequency. Given its universality, the same scheme can address various ill-posed inverse problems, e.g., it can be used instead of the Lucy-Richardson iterative deconvolution technique [15,16] for spectroscopy data.

Powerful nonperturbative techniques to investigate strongly correlated quantum systems at finite temperatures, such as QMC simulations, as well the use of QMC impurity solvers in dynamical mean-field theory (DMFT) [17] and cluster-DMFT [18], allow one to compute Green's functions $G(\tau)$ and other dynamical correlation functions in the imaginary-time τ domain. To make contact with experiments, one needs to transform the imaginary-time data $G(\tau)$ to the real-frequency spectral function (SF) $A(\omega)$.

Given $A(\omega)$, one can readily find the corresponding $G(\tau)$ via a Fredholm integral of the first kind [19–21],

$$G(\tau) = \int_{-\infty}^{\infty} \mathcal{K}(\tau, \omega) A(\omega) d\omega, \quad (1)$$

where for fermions the kernel \mathcal{K} is given by

$$\mathcal{K}(\tau, \omega) = -\frac{e^{-\omega\tau}}{1 + e^{-\beta\omega}} \quad (2)$$

for $0 \leq \tau \leq \beta = 1/T$, the inverse temperature. Although it is easy to calculate the integral in Eq. (1), the reverse transformation—required to relate QMC data to experiments—is ill-posed, exemplifying an inverse problem. It is highly susceptible to statistical and numerical errors. Even slight fluctuations in the input can result in significant discrepancies in the final result [22]. This makes the problem of analytic continuation extremely challenging, and due to its importance, for, e.g., quantum many-body physics, tremendous effort is being invested into its solution [23].

By discretizing imaginary time τ_i and frequencies ω_j and introducing notation $G_i \equiv G(\tau_i)$, $A_j \equiv A(\omega_j)$, Eq. (1) can be rewritten in a matrix form,

$$\vec{G} = \mathcal{K}\vec{A}, \quad (3)$$

where $\vec{G} = (G_1, \dots, G_N)$, $\vec{A} = (A_1, \dots, A_M)$, and \mathcal{K} is an $N \times M$ kernel matrix. The direct approach to determining $A(\omega)$ would be to calculate a generalized inverse of Eq. (3) [24,25]. However, the difficulties mentioned above usually render this simple idea unusable. Therefore, different methods have been proposed, including Padé approximants [26], stochastic analytic continuation [23,27], sparse modeling [28–30], spectrum averaging [31], stochastic pole expansion [32], genetic algorithms [33] and maximum entropy (MaxEnt) methods [34,35], with the last being the most widely used. Significant effort has been put into their enhancement, but the search for a fully reliable method continues. Recently, a new approach has been introduced using the “Nevanlinna”

structure of Green's function [36,37]. This approach undertakes interpolation rather than searching for a fit that matches multiple predefined conditions. A robust extension of this method, called PES, related to projection, estimation, and semidefinite relaxation, has been proposed in Ref. [38].

In this paper, we demonstrate the efficiency and accuracy of a different strategy, involving rapidly advancing machine learning (ML) techniques to tackle this issue. We compare it with MaxEnt. Neural Networks (NNs) have already been proposed to obtain $A(\omega)$ from $G(\tau)$ [22,39–41], producing remarkable results compared to MaxEnt. These approaches were based on a popular type of ML, *supervised learning*, where a NN is trained to provide an expected output for a given input. Since supervised learning requires labeled data, a large number of SFs must first be artificially generated in a physically meaningful way. These SFs are then the “labels,” i.e., the expected outputs. The corresponding $G(\tau)$'s are calculated according to Eq. (1), and pairs $[G(\tau), A(\omega)]$ are used to train the NN. Because this type of NN is trained on “artificial” SFs, it does not necessarily perform well for real QMC data. Our approach incorporates supervised learning as a first (pretraining) step, but ultimately the NN trains on real QMC data.

The rest of the paper is organized as follows. In Sec. II, we outline our autoencoder approach, with details of the neural network architecture, its training, and testing described in Appendix A. We focus in Sec. III on comparing our autoencoder results with MaxEnt, first for synthetic data and then for determinantal QMC (DQMC) [42,43] data. Details of the MaxEnt procedure are relegated to Appendix B, and DQMC details can be found in Appendix C. In Sec. IV we describe how we can include custom regularization, such as imposing sum rules, in our autoencoder approach. We conclude in Sec. V by summarizing how the autoencoder approach is more robust against noise compared to MaxEnt, the method traditionally used for analytic continuation of imaginary-time data to real frequency spectral functions.

II. AUTOENCODER APPROACH

Our proposed approach uses an autoencoder, schematically illustrated in Fig. 1, a *feed-forward*-type NN in which all information travels from the input layer to the output layer through the latent layer. The *encoder* consists of a NN that takes as input the imaginary time $\vec{G} = (G_1, \dots, G_N)$ and produces the spectral functions SF $\vec{A} = (A_1, \dots, A_M)$ in the latent layer. This in turn serves as the input to the *decoder*, which simply uses Eq. (3) to produce the output $\vec{G}' = (G'_1, \dots, G'_N)$. We can succinctly write this as

$$\vec{G} \xrightarrow{\text{EN}} \vec{A} \xrightarrow{\text{DE}} \vec{G}', \quad (4)$$

where the symbols EN and DE represent the *encoder* and *decoder* parts of the NN, respectively. The NN is then trained to minimize the difference between $G(\tau)$ and $G'(\tau)$, without requiring any prior knowledge of $A(\omega)$, as described below.

The details of the NN architecture used as the encoder are described in Appendix A; see especially Fig. 8 and associated discussion. We should note an important point of difference with typical applications where the encoder compresses

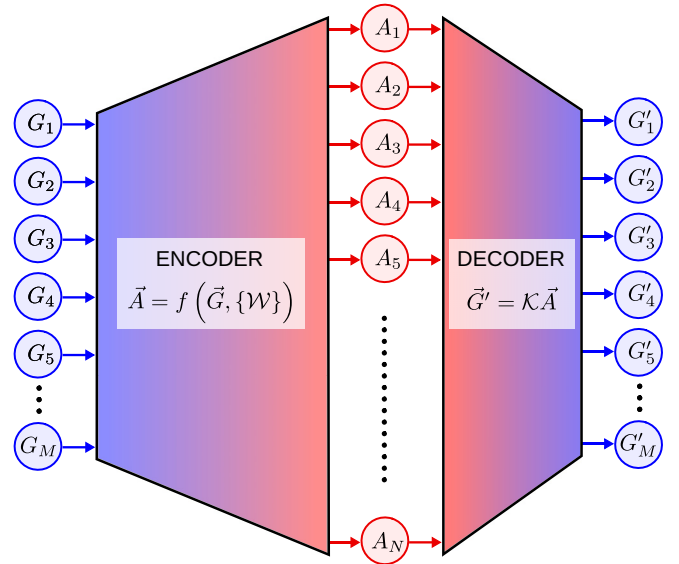


FIG. 1. Illustration of the structure of the autoencoder (AE), where G_i , G'_i , and A_i are elements of vectors defined in Eq. (3) and used in Eqs. (4) and (5). The function f that performs the inverse transformation $f: \vec{G} \rightarrow \vec{A}$ is represented by the *encoder* NN defined by trainable neuron weights $\{\mathcal{W}\}$ for fixed metaparameters, such as the number of layers in the neural network and the type of activation function.

information, and thus the latent layer has a size smaller than the size of the input and output layers. In our implementation, the size of the latent space is in fact larger than in the input and output layers, since we want good resolution in frequency space; however, we will still use the terms *encoder* and *decoder* for the first and second parts of the AE.¹

The only trainable layers in this architecture are in the encoder, where the NN performs the inverse transformation $\vec{A} = f(\vec{G})$ (see Fig. 1). This is possible due to the *universal approximation theorem*, asserting that a NN of sufficient size can accurately approximate any piecewise continuous function [44–46]. To implicitly define function f , the weights $\{\mathcal{W}\}$ in the NN are determined by requiring that the reconstructed Green's function \vec{G}' be as close to the original Green's function \vec{G} as possible. Since the decoder part performs a well-posed forward transformation (3), the encoder is trained to perform the inverse transformation with the SF \vec{A} as its output. The natural choice for the loss function to be minimized

¹The actual structure is richer, as shown in Fig. 8 and described in Appendix A 2. In particular, the encoder itself has a narrow layer with $\mathcal{N}_L = 5, \dots, 200$ neurons that acts similarly to the latent layer in AEs. In analogy to denoising, e.g., photos, this narrow layer helps to extract important features of Green's functions from QMC data with statistical noise. This AE-based extraction method achieves higher efficiency compared to averaging multiple “snapshots.” Consequently, the proposed approach allows shorter QMC runs to produce SFs that match the quality of MaxEnt SFs obtained from significantly longer runs. The influence of the width of this layer on the quality of the SFs is discussed in Appendix A 4.

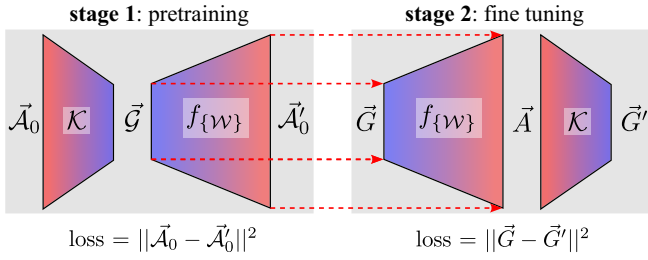


FIG. 2. Schematic illustration of the two-stage procedure. At *stage 1* artificial spectral functions (SFs) \vec{A}_0 are used to determine approximate values of the weights $\{W\}$, which at *stage 2* are used as initial values of the *encoder* part of the autoencoder (AE). At *stage 2* real QMC Green's functions \vec{G} are used to fine-tune $\{W\}$. While *stage 1* can be formally represented by an AE, since the direct transformation is just a matrix-vector multiplication, this stage falls into the *supervised learning* category. *Stage 2* is *unsupervised learning*.

during training is the squared distance between \vec{G} and \vec{G}' ,

$$\chi^2 = \|\vec{G} - \vec{G}'\|^2, \quad (5)$$

or its obvious generalization in terms of the covariance matrix [47].

Similar to the MaxEnt method, minimization of χ^2 suffers from the lack of uniqueness. One possibility is to introduce, as in MaxEnt, an additional entropy term, which favors the similarity of $A(\omega)$ to a chosen default model. However, this introduces an ambiguity in the choice of the default model and in the competition between χ^2 with the entropy term. Therefore, we propose a different approach that is described next.

Supervised pretraining and unsupervised learning

Since the forward problem (1) at fixed β does not depend on a particular physical model, the inverse transformation should be unique and independent of the nature of Green's functions. However, this would require infinite numerical precision. Moreover, since our approach represents the transformation as a NN, and too many neurons would make training infeasible, we have to work with an approximate form of the inverse transformation. We also emphasize that in the absence of *a priori* knowledge of the spectral functions $A(\omega)$, the problem is not amenable to supervised learning.

We thus propose the following two-stage procedure for training the encoder NN illustrated in Fig. 2:

(i) First, the $\overline{\text{EN}}$ learns general characteristics of SFs using artificially generated spectral functions. We call this the *pretraining stage*.

(ii) Second, we fine-tune the NN weights using $G(\tau)$'s obtained directly from QMC data. This stage corresponds to *unsupervised learning*.

The second stage uses the methodology already outlined above in (5) to optimize the NN weights starting with initial weights obtained from the first step. The pretraining stage, however, needs further discussion, and we turn to that next.

In the pretraining stage, we follow the method of Refs. [22,39–41] and use “artificial” SFs \vec{A}_0 modeled as the

sum of a random number of Gaussian peaks. For each SF, we calculate the corresponding Green's function \vec{G}_0 according to Eq. (3). Then, we use a large set of pairs $(\vec{G}_0^n, \vec{A}_0^n)$ to train $\overline{\text{EN}}$. In the language typically used for supervised learning, we use a Green's function \vec{G}_0 as a “data point” and the corresponding spectral function \vec{A}_0 as a “label,” which is expected at the output layer of the NN. We use calligraphic letters to distinguish the “artificial” SFs and the corresponding Green's functions from the ones obtained in QMC simulations. Formally, by analogy to Eq. (4), the procedure described above can be illustrated as

$$\vec{A}_0 \overline{\text{DE}} \vec{G}_0 \overline{\text{EN}} \vec{A}'_0, \quad (6)$$

where $\overline{\text{EN}}$ now plays the role of a decoder and is trained to minimize [48]

$$\eta^2 = \|\vec{A}_0 - \vec{A}'_0\|^2. \quad (7)$$

However, since $\overline{\text{DE}}$ is given by Eq. (3) and we know \vec{A} , this is a standard problem of supervised learning, which usually has a well-defined unique solution.

We call this *pretraining*, since at this stage the NN does not learn how to transform real QMC data. Instead, the network learns general features of SFs. This step is also useful for addressing the “curse of dimensionality” [49]. Namely, the *pretrained* weights and biases of $\overline{\text{EN}}$ provide suitable initial conditions for the second stage to prevent backpropagation from becoming trapped in a suboptimal local minimum. Thus, in actual training with QMC $G(\tau)$'s according to Eq. (4), we only adjust already pretrained weights and biases of $\overline{\text{EN}}$.

In the MaxEnt approach, prior knowledge of the spectra is introduced by defining the default model, which is used to select one of the many solutions to minimization of χ^2 in Eq. (5). In the present approach, the pretraining plays a similar role. Here, instead, it forces the SFs to be close to a realistic multipeak structure, as opposed to simple default models typically used in MaxEnt. Once acquired, the weights can be employed subsequently without the need for recalculation.

While pretraining is the main way to ensure a lack of ambiguity of the solution, it is easy to apply other techniques to improve the quality of the SFs obtained. We impose a limited number of physical assumptions about the SFs, such as their normalization and non-negativity. The flexibility of NNs enables the incorporation of any desired regularization. For instance, we can introduce the entropy term, typically maximized in the MaxEnt procedure, as a separate layer within the NN framework. We explore other possibilities, including simple $L1$ and $L2$ penalties (absolute and squared values of weights, respectively), as well as more sophisticated regularizations based on first- and second-derivative constraints to ensure the SF's smoothness. We further follow the idea of White [50] to incorporate constraints set by spectral moments that are known analytically. Using the flexibility and power of NNs, we demonstrate the ability to tackle the challenges posed by statistical noise and the absence of prior knowledge of the SF in a more universal manner.

Comparison of the *pretraining* stage with methods, where NNs are used to produce SFs, reveals similarities. Indeed, this stage is what has been proposed in Refs. [22,39–41] in the

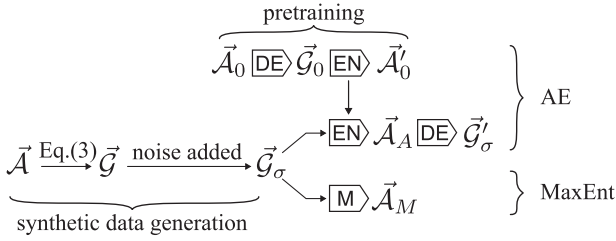


FIG. 3. Testing the robustness of our approach against statistical errors (details in text). \boxed{M} represents the MaxEnt procedure; \bar{A}_A (\bar{A}_M) denote the SFs obtained within the AE (MaxEnt) methods. The part denoted as “autoencoder” includes the pretraining and the actual AE training. We use index “0” for the SFs and Green’s functions used at pretraining stage to distinguish them from SFs used in the actual error estimation procedure.

context of *supervised learning*. However, we claim that the second stage, based on *unsupervised learning*, is crucial to obtain reliable results for QMC data. In Appendix A 4 we show that errors are much larger for the SFs obtained with \boxed{EN} trained only at the first stage (pretrained only) than after the full procedure (see Figs. 9 and 10).

III. COMPARING PERFORMANCE

Figure 3 shows how the proposed approach has been validated and compared with the MaxEnt method.

A. Synthetic data

To demonstrate the advantages of the proposed method, we need measures to evaluate its efficiency and accuracy. The general scheme of testing the proposed approach is illustrated in Fig. 3. There is no obvious way to assess the accuracy of the AE approach. The true SFs corresponding to $G(\tau)$ ’s produced by QMC are *a priori* unknown, and thus we cannot determine whether the result produced by the AE or by MaxEnt is more accurate. To overcome this difficulty, we compare the performance of the AE and MaxEnt on synthetic noisy Green’s functions, for which the exact SF is known. Namely, we generate “artificial” SFs of the same nature as those used in pretraining. Subsequently, we use Eq. (1) to calculate corresponding $G(\tau)$ ’s. Finally, we add noise σ that mimics averaging over different lengths of QMC runs. Various magnitudes of noise σ are added to $G(\tau)$ to produce a set \vec{G}_σ (for details, see Appendix A). The resulting set of \vec{G}_σ ’s is used to perform the second training stage of the NN [Eq. (4)]. Once training is complete, we test the AE (MaxEnt) by applying it on sets \vec{G}_σ independent of those used to train the NN and produce SFs \bar{A}_A (\bar{A}_M). This process is illustrated in Fig. 3. Since we know the original SF \bar{A} , we explicitly calculate the errors introduced by both the AE and MaxEnt. We define

$$\delta_A^2 = \|\bar{A} - \bar{A}_A\|^2, \quad \delta_M^2 = \|\bar{A} - \bar{A}_M\|^2, \quad (8)$$

where δ_A^2 and δ_M^2 measure the AE and MaxEnt errors, respectively.

Panels (a)–(d) of Fig. 4 show examples of SFs obtained with the AE and MaxEnt for two different noise magnitudes,

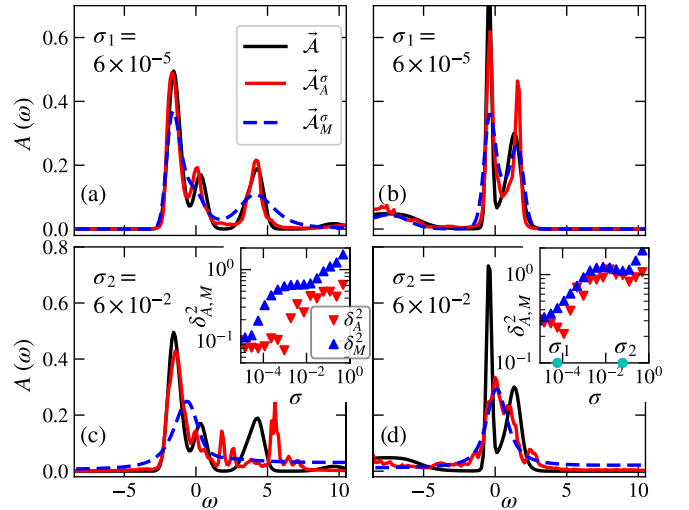


FIG. 4. Testing the spectral functions (SFs) predicted by the autoencoder (AE) (red solid line \bar{A}_A) and MaxEnt (blue dashed line \bar{A}_M) from Green’s functions \mathcal{G} calculated from a known SF \bar{A}_0 (solid black line). Results are shown for two different known SFs [one in (a) and (c), another in (b) and (d)] and for two values of the noise σ imposed on artificial Green’s functions [$\sigma_1 = 6 \times 10^{-5}$ in (a) and (b), $\sigma_2 = 6 \times 10^{-2}$ in (c) and (d)]. The insets show the σ -dependence of the reconstruction errors $\delta_{A,M}^2$ defined in Eq. (8). The cyan circles indicate the values of σ_1 and σ_2 . We always find the reconstruction error $\delta_A^2 \leq \delta_M^2$.

σ_1 and σ_2 . Figures 4(e) and 4(f) show the corresponding dependence of δ_A^2 and δ_M^2 on σ . It can be seen that δ_A^2 is smaller than or equal to δ_M^2 for all values of σ , indicating that shorter QMC runs can be used with the AE method to produce SFs with the same or greater accuracy than those produced by MaxEnt. We further demonstrate the impact of noise in Fig. 5. In panel (d) [(b)] we show the SFs $\mathcal{A}_A(\omega)$ [$\mathcal{A}_M(\omega)$] obtained by using AE [MaxEnt] on the $G_\sigma(\tau)$ ’s given in panel (c) for various σ . Comparison of $\mathcal{A}_{A,M}(\omega)$ with the original SF $\mathcal{A}(\omega)$ [Fig. 5(a)] directly highlights the robustness of the AE to noise. By evaluating the AE’s performance across σ , we examine the resulting SF’s susceptibility to statistical errors. We show that, having trained on a broad set of noisy Green’s functions, the AE adeptly captures crucial SF characteristics compared to simple averaging. This ability is the main reason why AEs are used in image denoising [51]. During training, in addition to learning to transform data to the frequency domain, the AE learns to distinguish between inherent statistical errors in QMC and key features that define the quantum system. It can be noted in Fig. 5 that while for $\sigma = 7 \times 10^{-6}$ (the cyan line) both MaxEnt and AE give accurate results, for $\sigma = 10^{-3}$ (the pink line) the AE reproduces much better the three-peak structure of the original SF. Therefore, one can suspect that significantly shorter QMC runs are needed to generate SFs of similar quality for the AE compared to MaxEnt. This problem is studied in the next section.

B. QMC data

Next, we test our approach on real QMC data. To do this, we use single-particle imaginary-time Green’s functions, obtainable in determinant quantum Monte Carlo (DQMC) simulation, which are defined as $G_{\vec{k}}(\tau) =$

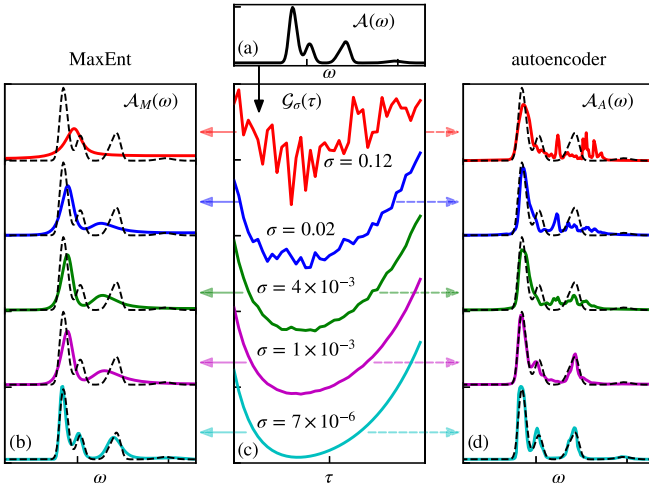


FIG. 5. Comparison of the effectiveness of the autoencoder and MaxEnt to reproduce “artificial” SF. (a) Example of “artificial” SF $\mathcal{A}(\omega)$ used to calculate $\mathcal{G}(\tau)$ using Eq. (3). Noisy $\mathcal{G}_\sigma(\tau)$'s with various σ values are shown in panel (c). MaxEnt results $\mathcal{A}_M(\omega)$ are shown in panel (b) and autoencoder results $\mathcal{A}_A(\omega)$ in panel (d). Both methods work well for the smallest value of $\sigma = 7 \times 10^{-6}$, but at $\sigma = 1 \times 10^{-3}$ (pink curve) we see that AE performs much better than MaxEnt in capturing the three peaks in $\mathcal{A}(\omega)$.

$-\sum_\sigma \langle \hat{c}_{\vec{k}\sigma}^\dagger(\tau) \hat{c}_{\vec{k}\sigma}^\dagger(0) \rangle$, where \vec{k} stands for the particle momentum vector, and $\hat{c}_{\vec{k}\sigma}^\dagger(\tau)$, $\hat{c}_{\vec{k}\sigma}(\tau)$ denote fermionic creation and annihilation operators at imaginary time τ . We study the Fermi-Hubbard model both at half-filling ($\mu = 0$) where there is no sign problem and away from half-filling ($\mu = 1.5$) at intermediate temperatures where the average sign is $\langle s \rangle \approx 0.3$ and away from zero so the DQMC is controlled [52–54]. In particle-hole symmetric form, the Hamiltonian is given by

$$\hat{H} = -t \sum_{(i,j)\sigma} (\hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} + \text{H.c.}) + U \sum_i \left(\hat{n}_{i\uparrow} - \frac{1}{2} \right) \left(\hat{n}_{i\downarrow} - \frac{1}{2} \right) - \mu \hat{N}, \quad (9)$$

where $\hat{c}_{i\sigma}^\dagger$ ($\hat{c}_{i\sigma}$) denotes fermionic creation (annihilation) operators, t is the hopping integral, and U denotes the on-site Coulomb repulsion. Fermionic number operators are defined as $\hat{n}_{i\sigma} = \hat{c}_{i\sigma}^\dagger \hat{c}_{i\sigma}$, $\hat{n}_i = \hat{n}_{i\uparrow} + \hat{n}_{i\downarrow}$, $\hat{N} = \sum_i \hat{n}_i$.

We begin the discussion of the DQMC results by showing SFs $A_{\vec{k}}(\omega)$ calculated on a half-filled 16×16 system at high symmetry points of the Brillouin zone in Fig. 6. Both AE and MaxEnt results are presented, with relatively good agreement between these two approaches. Similarly, in Appendix C we show the comparison of the AE and MaxEnt density of states. However, since in this case the exact SF is not known, it is not possible to determine which of these results is more accurate. Therefore, we propose another method to estimate the accuracy in the proposed approach. To this end, we compare the convergence rate of the AE with that of the MaxEnt method. In analogy to Eq. (8), we define

$$\Delta_A^2(\sigma) = \|\vec{A}_A^\infty - \vec{A}_A^\sigma\|^2, \quad \Delta_M^2(\sigma) = \|\vec{A}_M^\infty - \vec{A}_M^\sigma\|^2, \quad (10)$$

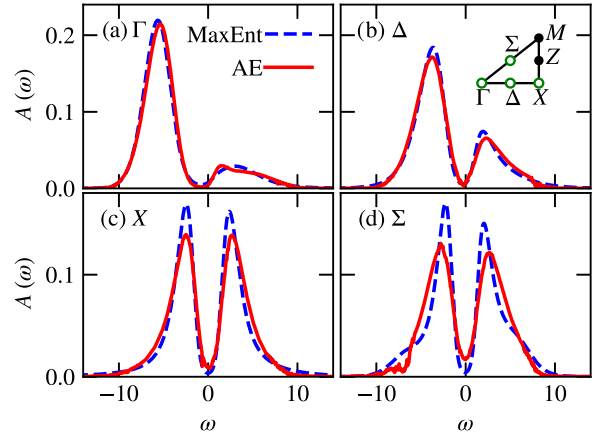


FIG. 6. (a)–(d) Spectral functions obtained from DQMC $G(\tau)$ using AE (red) and MaxEnt (blue dashed) for the 2D Hubbard model with $U/t = 8$, $\mu = 0$, and $\beta = 2t$. Various panels correspond to momenta marked in panel (b) with open green circles.

where \vec{A}_A^∞ (\vec{A}_M^∞) denotes the SF obtained within the AE (MaxEnt) method for the number of $G(\tau)$'s sufficiently large to ensure the SF's convergence. \vec{A}_A^σ and \vec{A}_M^σ represent SFs analogous to \vec{A}_A^∞ and \vec{A}_M^∞ , but they are calculated from a smaller number of $G(\tau)$'s, which is chosen to give a statistical error equal to σ . Figure 7 shows the convergence rates for AE and MaxEnt for half-filling and away from it.

For both fillings, the AE SFs converge much faster than the MaxEnt SFs, i.e., the AE method can produce the same quality SFs from more noisy $G(\tau)$'s than the MaxEnt method. Here, by convergence we mean the noise level σ that leads to a certain value of the error between A^σ and A^∞ . This value of σ is higher for most of the points displayed in Figs. 7(c) and 7(f), thus suggesting that shorter QMC runs are sufficient for achieving a desired output quality. We note, however, that at large σ , the performance might degrade in the AE framework. Nevertheless, our observation indicates that such values introduce extreme noise levels that lead to the nonphysical scenario of $G(\tau)$'s changing the sign [see Fig. 5(c)]. Thus, we conclude that, in practical applications, shorter QMC runs would be required when using the AE.

IV. CUSTOM REGULARIZATION: SUM RULES

In this section, we elaborate on defining the regularization for the NN. The $\langle \text{EN} \rangle$ part of the AE has to solve the problem of transforming the input function into a function that, when integrated, gives the same input function. The transformation is determined by weights and biases, which are adjusted during the training stage. Since this is an ill-posed problem, it is reasonable to expect that the final values of these weights and biases will not be optimal, resulting in divergence in the output of AE from the input.

Training is an iterative process that starts with random (or pretrained) initial values of the weights and biases, and therefore the final parameters of the neural network can also depend on initialization. To minimize this dependence, we employ a technique to improve the results of a *single initialization*. Since integration “smooths out” irregularities in the

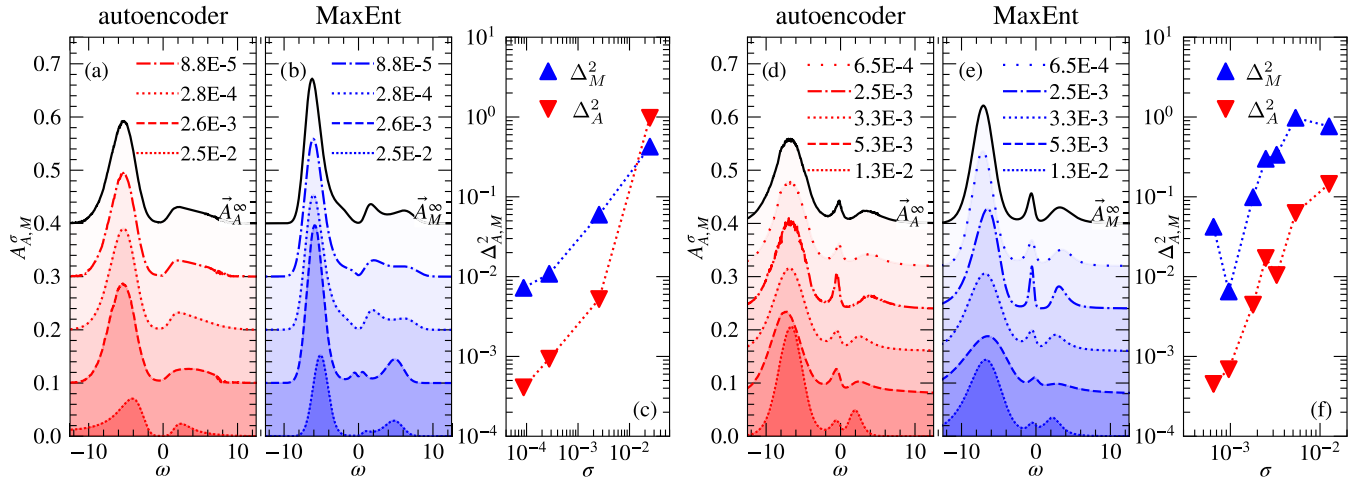


FIG. 7. Comparisons of SFs obtained by AE and MaxEnt at half-filling $\mu = 0$ [panels (a)–(c)] and away from half-filling $\mu = 1.5$ [panels (d)–(f)] at the Γ point in the Brillouin zone. The SFs are overlaid, starting from the shortest QMC runs (highest σ). $A_{A,M}^\infty$ are indicated with black lines. The convergence rates for half-filling and away from it are shown in (c) and (f), respectively [dotted lines in panels (c) and (f) are guides to the eye]. The errors $\Delta_{M,A}$ are defined in Eq. (10). The AE (MaxEnt) results are marked with red inverse triangles (blue triangles). We average over the errors in $G(\tau)$'s at all τ to obtain σ . We see from panels (c) and (f) that the errors for AE (Δ_A^2) are generally smaller than the errors for MaxEnt (Δ_M^2).

integrand function, the output of the DE part is highly insensitive to high-frequency oscillations or noise in the output of the encoder part. Consequently, to extract the essential features of SFs, we incorporate regularization methods commonly used in inverse problems. These regularization techniques help stabilize and improve the reconstruction process. To this end, the total cost function is defined as

$$\mathcal{L} = \chi^2(\vec{G}, \vec{G}') + \sum_m \alpha_m \eta^m(\vec{G}, \vec{G}'; \vec{A}), \quad (11)$$

where the regularization strength α_m describes the significance of the specific regularization function $\eta^m(\vec{G}, \vec{G}'; \vec{A})$. The interplay between different penalty terms enables a more meaningful solution, especially in situations where the transformation is inherently ambiguous. In the MaxEnt procedure, as described in the preceding section, the loss \mathcal{L} is accompanied by an *entropy* penalty, which is used to minimize spurious correlations between the data by introducing a default reference model. Typically, a stronger regularization leads to a smoother output during the procedure. A notable advantage of the AE approach lies in its flexibility to incorporate arbitrary regularization terms. In particular, we consider physically relevant terms that penalize deviations from the *sum rules*, which can be analytically calculated for the Hubbard model up to the second moment [50]. They are defined as

$$m_0 = \int_{-\infty}^{\infty} d\omega A(k, \omega) = 1, \quad (12a)$$

$$m_1 = \int_{-\infty}^{\infty} d\omega \omega A(k, \omega) = \varepsilon_k - \mu + \frac{U}{2}(n-1), \quad (12b)$$

$$m_2 = \int_{-\infty}^{\infty} d\omega \omega^2 A(k, \omega) = \left(\varepsilon_k - \mu - \frac{U}{2} \right)^2 + U \left(\varepsilon_k - \mu - \frac{U}{2} \right) n + \frac{1}{2} U^2 n, \quad (12c)$$

where $\varepsilon_k = -2t(\cos k_x + \cos k_y)$ is the tight binding dispersion on the square lattice with only nearest-neighbor hopping t . In Table I, we compare the deviations of the three lowest moments of the AE and MaxEnt SFs from the exact values. While the zeroth moment (normalization of the SF) is rather precise in both approaches, the errors for the remaining moments are significantly smaller for the AE.

The moments are enforced by applying penalties that increase as the computed moment for the resulting SF deviates from its anticipated value. However, this introduces additional parameters that determine the priority of satisfying the sum rules in relation to the importance of similarity between \vec{G} and \vec{G}' . If the penalties are too small, the AE might neglect the sum rules; conversely, if the penalties are too large, it could generate a random function that gives the correct moments.

For some of our results, we have employed hyperparameter optimization techniques such as grid search or Bayesian search [55]. It should be noted that there may be potential improvements in the quality of the results by additionally applying *ensemble learning* techniques [56].

V. SUMMARY

Due to the completely different schemes used to carry out the analytical continuation with the proposed method and the

TABLE I. Absolute errors of m th-order moments for various high-symmetry points in the Brillouin zone, obtained with AE and MaxEnt from the QMC SFs.

m	AE				MaxEnt			
	Γ	Δ	X	Σ	Γ	Δ	X	Σ
0	0.0073	0.0053	0.0009	0.003	0.002	0.0014	0.002	0.0011
1	0.02	0.0012	0.0001	0.0021	0.07	0.024	0.02	0.0055
2	0.61	0.16	0.71	0.12	1.6	1.5	3.6	0.7

MaxEnt method, it is not possible to make an unambiguous comparison of the efficiency of these methods in terms of, for example, calculation time, computer resources required, or accuracy achieved. Both the AE and MaxEnt methods require fine-tuning to obtain accurate results. In the case of the MaxEnt method, we must at least choose the default model and the “strength” of the entropy term. For the AE method, we have to decide on the metaparameters (number of layers in the neural network, the type of the activation function, etc.). And this process of fine-tuning can depend on models, on model parameters, on temperature, etc. Additionally, in the case of QMC data, the target spectral function is unknown and therefore it is not possible to directly compare the absolute accuracy of the two methods.

However, despite these difficulties, several advantages of the AE approach can be clearly identified, which may prove crucial for obtaining accurate results for highly correlated systems, especially in areas where massive QMC simulations are needed to reduce statistical errors. Among these, we can highlight the following strengths of the proposed method:

(i) In contrast to MaxEnt, where one has to define the default model, no prior knowledge about the physical nature of the SFs is needed.

(ii) In the standard approach, MaxEnt is applied to smooth $G(\tau)$'s obtained by averaged QMC runs. In the AE method, the NN aims to recognize the meaningful characteristics of $G(\tau)$'s and distinguish them from statistical noise. Unlike basic averaging, which requires many noise realizations for effective denoising, a properly trained variational AE can efficiently yield results with a few or even just a single image [51,57]. That is why, in most cases, for larger σ the AE outperforms MaxEnt. The ability to obtain the same precision in QMC through shorter simulations is crucial, especially when the fermion sign problem yields noisier data at lower temperatures.

(iii) By introducing custom regularization, imposing physical constraints on the SFs within the AE approach is straightforward. In addition to non-negativity or smoothness requirements, one can impose, e.g., the sum rules.

(iv) The AE requires only one training for a given set of model parameters, such as U or filling, and all momenta can be trained at the same time, in principle. Further, the AE does not need to be retrained for small changes in the model parameters. After the training stage, a few $G(\tau)$'s are sufficient to produce a reliable SF.

In conclusion, our comparison of the performance of the AE with the MaxEnt method finds that AE outperforms MaxEnt in obtaining the specific features in the spectral function accurately. The AE's and MaxEnt's predictions show similar features; however, we find that the autoencoder approach exhibits greater robustness to statistical noise. Remarkably, as a result, the AE can be applied to data from notably shorter simulations, enabling the study of larger systems. Our method offers valuable insights into the spectral properties of quantum systems, particularly when QMC simulations are computationally demanding.

This work highlights and upholds the potential of using the AE as a powerful tool for extracting SFs from Green's functions, paving the way for advances in condensed-matter physics and other fields where such spectral analysis is

crucial. We anticipate that the advancement of sophisticated networks within the rapidly evolving field of ML will incorporate and build on the methodology proposed in this study, leading to a deeper understanding of strongly correlated systems. We notice that numerous diverse ideas, recently introduced, can be integrated into the framework of this approach. Moreover, the proposed AE method can also be applied to other inverse problems, such as those mentioned in the Introduction. Recently, a similar approach was proposed as a tool to determine the separability of quantum states [58].

ACKNOWLEDGMENTS

L.K., M.R., and N.T. were supported by the NSF Materials Research Science and Engineering Center Grant No. DMR-2011876. S.R. was supported by NSF Grant No. GR126818. T.P. acknowledges financial support from Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro Grants No. E-26/200.959/2022 and No. E-26/210.100/2023; CNPq Grants No. 403130/2021-2 and No. 308335/2019-8; and also INCT-IQ. Numerical calculations have been partly carried out using high-performance computing resources provided by the Wrocław Centre for Networking and Supercomputing and the Unity cluster at Ohio State.

APPENDIX A: DETAILS OF THE AUTOENCODER NEURAL NETWORK AND ITS TRAINING AND TESTING

In this Appendix, we describe the methods we use to perform the *analytic continuation* for both artificially generated Green's functions and those obtained from the QMC simulation. We briefly describe the NN architecture in terms of technical details of the trainable part (see Fig. 8). We begin the discussion by providing the form of the input data. The SFs presented in the scope of this work have two separate origins. We introduce both of them in the following subsections. Namely, we establish the procedure of generating the *pretraining* SFs data set in order to validly perform *training* on the Green's functions obtained from the QMC.

1. Pretraining procedure

During the *pretraining*, defined as a stage of training the initial weights of the $\overline{\text{EN}}$, we introduce a substantial number of “artificial” SFs $\{\vec{A}^n\}$. These are modeled to feature multiple Gaussian peaks randomly located at various frequencies, each with random weights,

$$\mathcal{A}_{\text{Gauss}}(\omega) = \sum_i \lambda_i \frac{1}{\sqrt{2\pi}\zeta_i} \exp\left[-\frac{1}{2\zeta_i^2}(\omega - \omega_i)^2\right], \quad (\text{A1})$$

where λ_i , ω_i , and ζ_i are random parameters that determine the shape of the “spectral function” ($\sum_i \lambda_i = \pi^{-1}$).

As the choice of Gaussian peaks may not be pertinent and unique, we additionally perform the pretraining using a random combination of Lorentzian peaks of the form

$$A(\omega) = \sum_i \lambda_i \frac{\gamma_i}{(\omega - \omega_i)^2 + \gamma_i^2}, \quad (\text{A2})$$

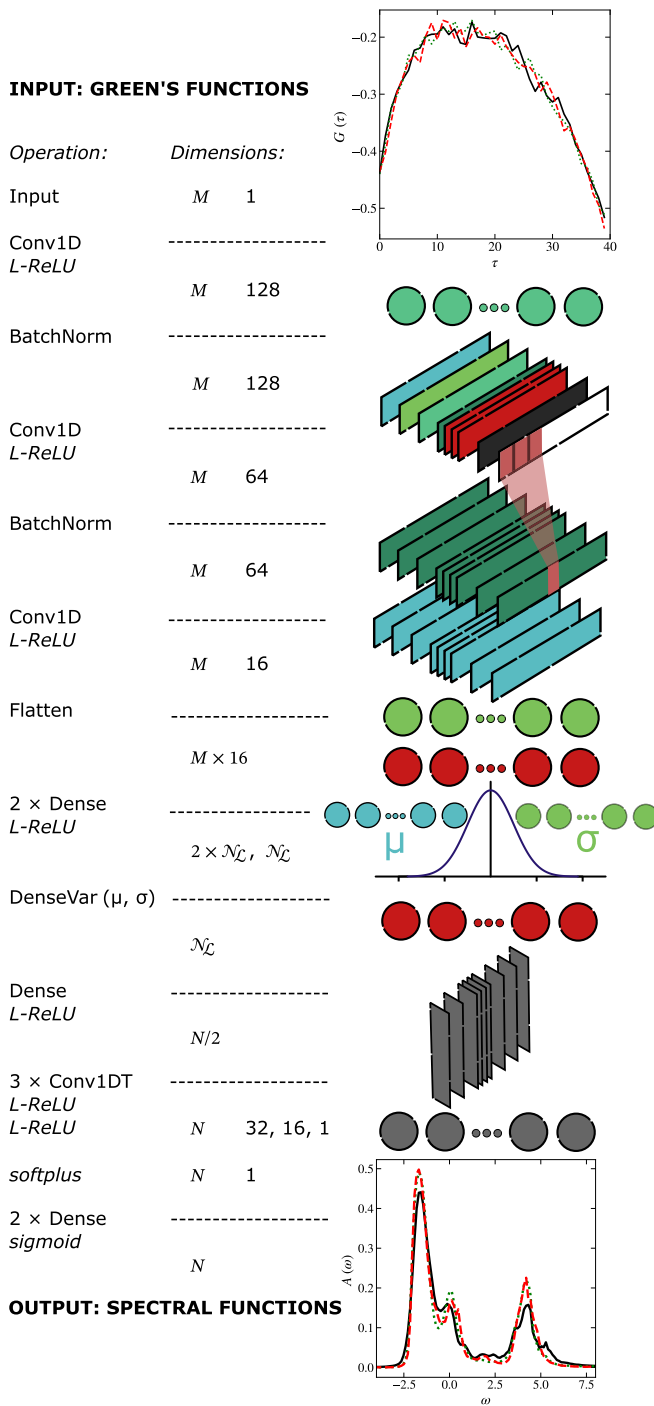


FIG. 8. An example architecture of the $\boxed{\text{EN}}$ part of the NN with N frequency units and M imaginary Matsubara times. Layers are shown graphically with their corresponding dimensions. Reshape layers have been omitted for simplicity. The layer names are according to Keras API [59]. The network was implemented in the TensorFlow framework [60].

where the meaning of λ_i , ω_i , and γ_i is analogous to that of the parameters in Eq. (A1). Spectral functions of this form are often observed in physical systems, in which the parameter γ_i indicates a finite relaxation time [17,61]. In Fig. 10 we show examples of results obtained with AE pretrained

with Lorentzian spectral functions. The procedure will be described more thoroughly in the following sections.

Although the pretraining procedure alone may strongly depend on the choice of the particular form of $\mathcal{A}(\omega)$, we argue that the AE performs well on both of them. Potentially, the most general approach is to consider a variety of forms and random parameters for the pretraining data set. This approach could additionally equip the network with the ability to work adequately well for distinct input Green's functions. We note, however, that the generalization of the network is not the main goal of this work. In terms of inverse problems, one should balance between having a general and a highly accurate prediction, focusing on the latter.

After generating a large set of artificial SFs, for each of them we solve the forward problem, as outlined in Eq. (1), to compute corresponding Green's functions $\{\tilde{\mathcal{G}}^n\}$ at a specified inverse temperature β . To ensure that these SFs are physically meaningful, we adhere to the recently proposed generation procedure by Zhang *et al.* [22], a method that has been shown to effectively optimize the weights of the NN. To perform the *pretraining* process, we employ pairs $(\tilde{\mathcal{G}}^n, \tilde{\mathcal{A}}_0^n)$, as elaborated in the main text. We introduce statistical noise as random normal variables $\mathcal{N}(0, \sigma^2)$ for each imaginary-time point τ within $\{\tilde{\mathcal{G}}^n\}$'s. We note that after a binning procedure, the noise obtained during QMC shall have such a form [62,63]. The variance, denoted as σ , characterizing Green's function, which is typically obtained by statistical means from the QMC, serves as a metric in Fig. 4 to quantify the resilience of the AE and MaxEnt approaches in the presence of deviations resulting from finite numerical simulations.

2. Network architecture

In this work, we propose the *autoencoder* approach to *analytic continuation*, as described in Eq. (4). While the number of imaginary times, denoted as M ($M = 40$ in most cases) in Green's functions, and $N = 250, 1000$ frequencies in the SFs, impose constraints on the outer layers of the $\boxed{\text{EN}}$ part of the AE (specifically, the *input* and the *output*), the internal structure of the network architecture is by no means restricted. Moreover, the $\boxed{\text{DE}}$ part is strictly determined by the Fredholm integral [Eq. (1)] and can be implemented as a fully connected, nontrainable dense layer, featuring weights defined by Eq. (2) and zero bias. The presence of statistical noise originating from QMC simulations suggests that it is reasonable to utilize network architectures that are capable of discriminating between noise and relevant data. The choice of a network architecture is significant for the autoencoder performance, and the simplest networks may fail to reproduce the Green's functions with high quality. In this context, we choose to implement a variational autoencoder (VAE) type network.

VAEs represent a class of NNs that are characterized by their generative capabilities [64,65]. That is, they model the distribution of the true data generation by introducing a latent dimension. In this context, there resides a single VAE within $\boxed{\text{EN}}$, since the true $\boxed{\text{DE}}$ is determined *a priori*. VAEs have been shown to be effective across a spectrum of optimization problems, including anomaly detection [66,67], text

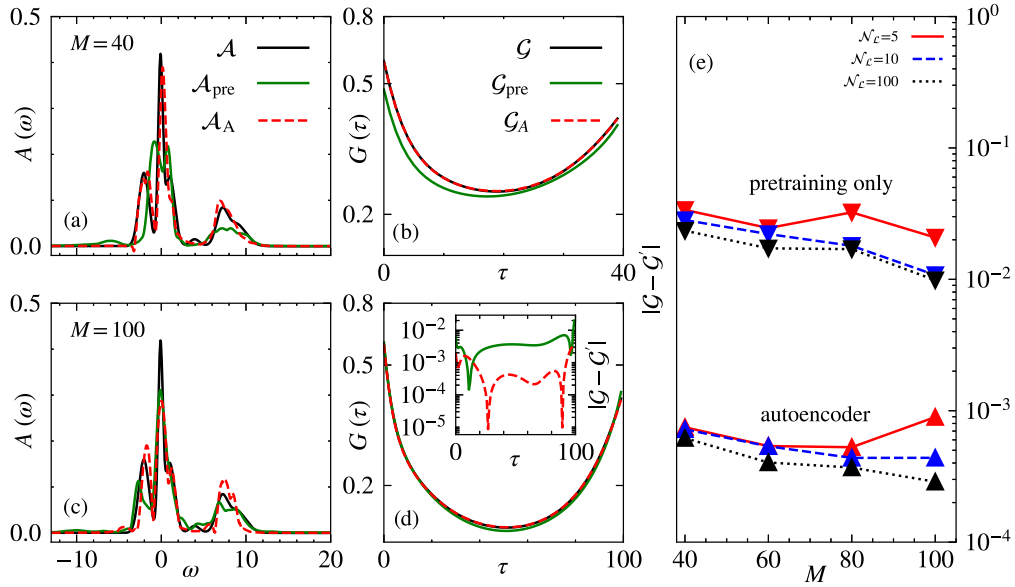


FIG. 9. Benchmarks of the AE approach in terms of the number of variational parameters using Gaussian artificial SFs [see Eq. (A1)]. (a),(c) SFs \mathcal{A}_{pre} obtained with pretrained models only (green lines) (see the upper procedure in Fig. 3 in the main text) and SFs \mathcal{A}_A obtained using the proposed unsupervised AE technique (red dashed lines) for the number of imaginary times $M = 40$ (a) and $M = 100$ (c). (b),(d) The corresponding Green's functions. In panels (a)–(d) the “true” Green's functions \mathcal{G} and SFs \mathcal{A} are shown with black solid lines. (e) Reconstruction error for Green's functions. Inverse triangles are for the pretrained-only model and triangles for AE as a function of imaginary times number M for different sizes of the AE latent space $\mathcal{N}_{\mathcal{L}}$ (see Fig. 8). The procedures are performed for $N = 1000$ frequencies. Inset: Reconstruction error for single realization of Green's functions at each imaginary time τ ($M = 100$). The Green's functions are taken akin to panel (d).

classification [68], illness diagnosis and classification [69], and physical problems [70–73]. The ability of VAEs to faithfully characterize each element of the input set through the latent probability distribution makes them a potentially powerful tool to address inverse problems [74].

In the initial part of $\overline{\text{EN}}$, we opt for the use of one-dimensional convolutional (Conv1D) layers featuring a kernel of size 3 and a stride of size 1. The application of one-dimensional kernels allows the network to grasp the monotonic patterns within the input data. This further contributes to diminishing the impact of noise on Green's

functions. We adopt the initialization of the weights for the *pretraining* procedure akin to He *et al.* [75]. Batch normalization is applied between Conv1D layers with activation as a *leaky* implementation of rectifier linear units (L-ReLU) in order to avoid overfitting to a single noise realization. The latent space representation comprises $\mathcal{N}_{\mathcal{L}} = 120$ fully connected neurons. We tested various $\mathcal{N}_{\mathcal{L}}$ values ranging from 5 to 200 [see Figs. 9(e) and 10(e)]. We found that those sizes are sufficient to extract all the important features of the resulting SFs. To transform the encoded values within the latent dimension, we employ multiple transposed Conv1D (Conv1DT) layers.

The final output of $\overline{\text{EN}}$ consists of two fully connected feed-forward layers, each comprising N units, with the sigmoid activation function facilitating the normalization condition. Both *pretraining* and *training* procedures are executed using the Adam optimizer [76] with *ams-grad* enhancement [77]. For efficient learning, we use the early stopping condition on the validation data set [78]. An example of the architecture used for the generation of QMC SFs for the Hubbard model from Eq. (9) is visualized in Fig. 8.

In principle, the architecture allows for performing the analytic continuation for various momenta \vec{k} simultaneously. As we wanted to make the presentation more accessible, we only experimented with this idea, but we plan to return to it for future development. Additionally, to further control the noise level and input dimensionality, one can approximate the input Green's functions using Legendre polynomials, with the flexibility to make adjustments by manipulating their order [79]. This approach would allow us to perform the analytic

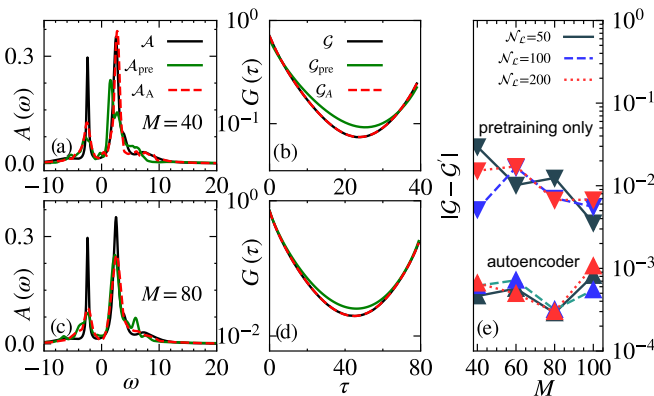


FIG. 10. The same as in Fig. 9, but for Lorentzian SFs [see Eq. (A2)]. The procedures are performed for the number of frequencies $N = 2000$. The number of imaginary times is (a),(b) $M = 40$ and (c),(d) $M = 80$.

continuation regardless of the imaginary-time dimension in the input Green's functions. Although we do not currently employ this technique, it can be easily applied within our framework. Different architectures and activation functions remain to be explored in the future.

3. Testing procedure

We illustrate the procedure that we used to compare the effectiveness of the MaxEnt and AE approaches. Panel (a) in Fig. 5 in the main text shows an example of an "artificial" SF. Then, Eq. (1) is used to generate the corresponding Green's function. In the next step, a series of noisy Green's functions \mathcal{G}_σ is generated by introducing random disorder of various magnitudes σ . Examples are shown in panel (c). An increase in the noise makes the inverse problem harder to optimize. This can be seen in panels (b) and (d), which show the results of the analytic continuation performed with the help of the MaxEnt and AE methods, respectively. As expected, reducing the noise level improves the ability of both methods to find the corresponding spectral function more accurately from the noisy set $\{\tilde{\mathcal{G}}_\sigma^n\}$. However, the AE method gives more accurate results for large and intermediate noise levels. This confirms that shorter QMC runs are needed for AE to obtain the same accuracy as for the MaxEnt method.

4. Parameter dependence and benchmarks

In this section, we explore the performance of the AE method as we vary a number of its variational parameters. Within the architecture used in this work (see Fig. 8), the natural choices for these parameters are (i) the latent dimension \mathcal{N}_L , (ii) the number of imaginary times M , and (iii) the number of frequencies N . Following the same approach as in the main text (and as denoted in this Appendix), we conduct benchmarks using both artificial SFs and those obtained from real QMC simulation.

In Fig. 9, results obtained using two different NN approaches (supervised and unsupervised) for the reconstruction procedure on Gaussian artificial SFs [Eq. (A1)] are compared. Figures 9(a) and 9(c) shows the true SFs, marked with solid black lines, along with the reconstructions obtained using the ANN according to Eq. (2). The corresponding Green's functions \mathcal{G} are presented in Figs. 9(b) and 9(d) with $M = 40$ ($M = 100$) imaginary times. Additionally, the red dashed lines represent the SFs and $\mathcal{G}(\tau)_A$, \mathcal{G}_{pre} 's obtained with the AE approach \mathcal{A}_A , while the solid yellow lines show the results obtained using a network with pretrained weights only, \mathcal{A}_{pre} . Namely, in the latter, the model has been pretrained on some other $\mathcal{G}'(\tau)$'s and tested on a new data set of $\mathcal{G}(\tau)$ without additional steps, facilitating supervised learning. In this comparison, we used $\mathcal{N}_L = 50$ and $N = 1000$. Notably, the proposed AE approach, incorporating an unsupervised training phase, produces Green's functions \mathcal{G}_A that are closer to the true \mathcal{G} than those produced by the only pretrained network.

To further support this claim, in Fig. 9(e) we illustrate the distance $|\mathcal{G} - \mathcal{G}'|$ between the true and reconstructed Green's functions for various latent dimensions $\mathcal{N}_L = 5, 10$, and 100. The error naturally decreases with an increasing number of latent neurons \mathcal{N}_L . To account for variations, we average the

latter over $N_r = 10$ artificial SFs. Similar results are also presented in Fig. 10 for Lorentzian artificial SFs [Eq. (A2)]. The presentation is analogous to that of Fig. 9. However, there we used a wider latent layer $\mathcal{N}_L = 50, 100, 200$ and $N = 2000$. Interestingly, we demonstrate that for such a relatively high complexity of the network, the reconstruction errors appear to reach a plateau as \mathcal{N}_L increases, sometimes resulting in marginally higher errors.

To additionally evaluate the performance of the proposed AE framework, in Fig. 11 we use the real QMC data of the Hubbard model at $U = 3, \mu = 0$ with an increasing number of imaginary times M . We do this by fixing the inverse temperature $\beta = 3$, that is, reducing the error introduced due to the Trotter decomposition $d\tau$ [42]. Since the number of imaginary times M determines both the input and output size of the neural network, varying it effectively alters the complexity of the network. Figures 11(a), 11(b) and 11(c) present the SFs $A(\omega)$ obtained through Green's functions from QMC simulations, depicted in Figs. 11(d), 11(e) and 11(f) at Γ , Δ , and X points of the Brillouin zone, respectively. The training was performed using $N_r = 100$ snapshots from the QMC. With different colors and lines, we indicate the number of imaginary times M . Notably, for all values of M , the resulting SFs are very similar. This suggests that the AE can effectively perform analytic continuation regardless of the size of the input data. In Figs. 11(g), 11(h) and 11(i), we additionally present the deviations of the spectral moments \tilde{m}_i of the reconstructed spectral functions from true m_i for $i = 0, 1, 2$. The definition of the spectral moments can be found in Eq. (12).

5. Noninteracting spectral functions

The proposed method for analytic continuation is tailored for interacting systems, for which a challenge comes from the difficulty of obtaining the spectral response for large systems, which remain beyond the reach of current computer clusters. To address this, we have developed a benchmarking strategy using the artificial SFs of an arbitrary complication level, as described in detail in the previous subsections. However, to additionally validate the proposed method, we complement the aforementioned benchmarks by using the SFs obtained directly from the diagonalization procedure of the Hamiltonian matrix of a well-defined, noninteracting system. For such a system, it is straightforward to compute the retarded Green's function using Lehmann's representation,

$$G(z) = \frac{1}{z - H} = U \frac{1}{-D + \mathbf{z}} U^T, \quad (\text{A3})$$

where U is a unitary matrix that diagonalizes the Hamiltonian $H = UDU^T$ and \mathbf{z} is a diagonal matrix with each element equal to z . Subsequently, the above Green's function is transformed to the momentum space

$$G(\vec{k}, \vec{k}', z) = \sum_{\vec{R}_i} \sum_{\vec{R}_j} e^{i(\vec{k} \cdot \vec{R}_i - \vec{k}' \cdot \vec{R}_j)} G_{ij}(z), \quad (\text{A4})$$

where the summation goes over the real-space vectors of a specific lattice. Note that the dependence on a single momentum is valid for translationally invariant systems only. Since our interest in this work does not extend to

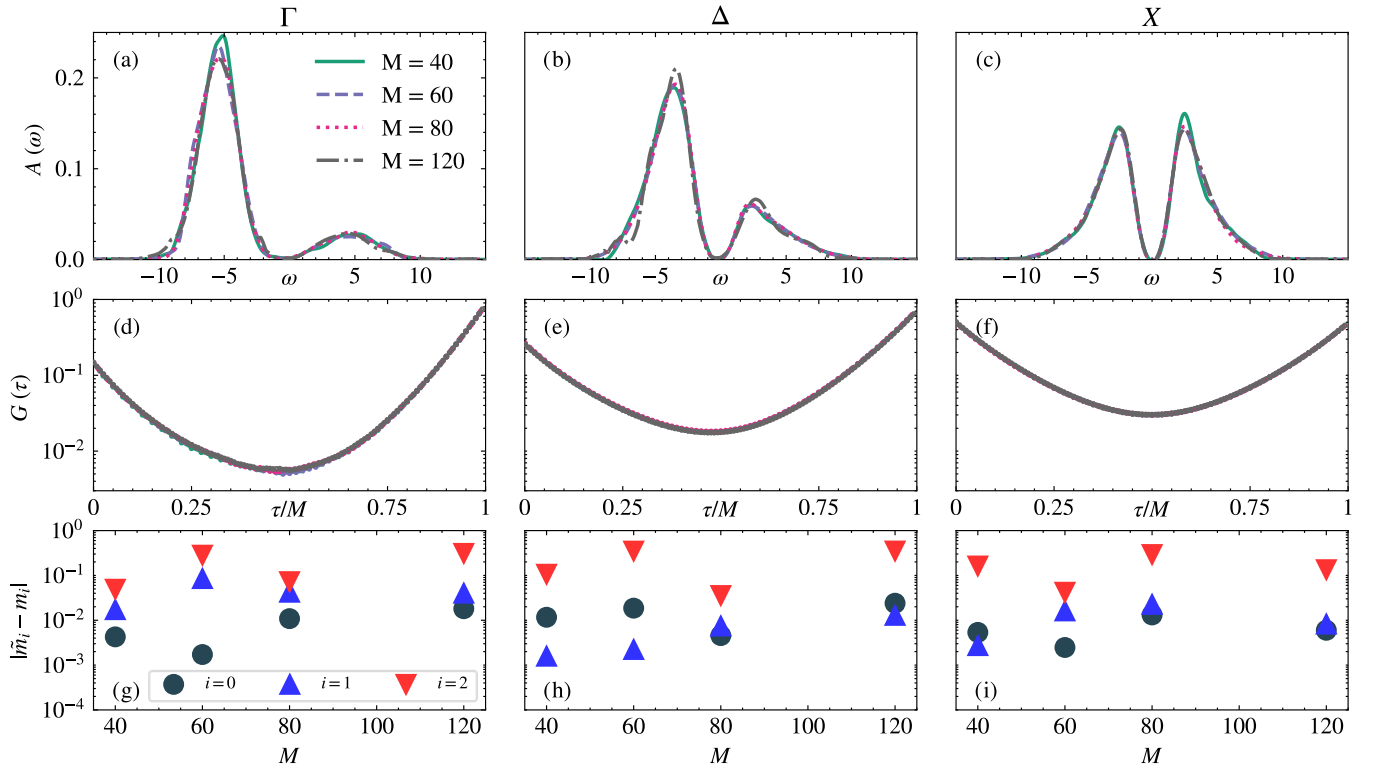


FIG. 11. (a),(b),(c) SFs $A(\omega)$ calculated using the AE method. They are obtained from Green's functions at high-symmetry points of the Brillouin zone Γ , Δ , and X , shown in panels (d), (e), and (f), respectively. Line styles and colors correspond to different numbers of imaginary times M . Here, the inverse temperature $\beta = 3$ is fixed, thus effectively only the Trotter step size $d\tau$ changes. (g),(h),(i) Deviations of the spectral moments of the reconstructed SFs \tilde{m}_i , for $i = 0, 1, 2$, from the true moments m_i given by Eq. (12).

resolving all properties of the system, we exclusively focus on equal-momentum Green's functions, i.e., $\vec{k} = \vec{k}'$. The spectral function $A(\vec{k}, \omega)$ is determined from the standard formula

$$A(\vec{k}, \omega) = - \lim_{\epsilon \rightarrow 0^+} \frac{1}{\pi} \text{Im} G(\vec{k}, \omega + i\epsilon). \quad (\text{A5})$$

The spectral functions are then transformed according to Eq. (3) to obtain the Green's functions $G(\tau)$. Although the notion of temperature broadening is insignificant for noninteracting systems (the Boltzmann factors cancel out due to the lack of interactions), we still require input to the NN in the form of imaginary-time vectors. Therefore, in what follows we assume $\beta = 2$ and $d\tau = 0.05$ for simplicity. Similarly to the artificial SF approach, we impose noise $\sigma = 10^{-5}$ on the $G(\tau)$'s. For this benchmark, we choose the Aubry-André (AA) model [80] in two dimensions [81,82],

$$\hat{H} = -t \sum_{nm} [\hat{c}_{n,m}^\dagger (\hat{c}_{n+1,m} + \hat{c}_{n,m+1}) + \text{H.c.}] + \lambda \sum_{nm} [\cos \alpha(n+m) + \cos \alpha(n-m)] \hat{c}_{nm}^\dagger \hat{c}_{nm}, \quad (\text{A6})$$

in which the potential is a periodic function with a generally irrational frequency parameter α [set to $\alpha = (\sqrt{5} - 1)/2$ throughout this study] and the potential strength λ . We assume the hopping integral $t = 1$. Typically, a random phase ϕ is added to the periodic function, but here it is set to zero. The AA model describes the incommensurability between the length of the space-modulation of the on-site potential and the

lattice constant. There exists a self-duality between momentum and real space at a transition point $\lambda = 2$ [83]. Below this value, all wave functions are localized in the momentum space. In contrast, when $\lambda > 2$, the eigenvectors are localized in the real space. Quasiperiodic systems like this are of great interest, possessing multiple fascinating topological properties that they can inherit from higher-dimensional "parent" models [84]. They also tend to exhibit multifractal spectra [85]. The AA model is also considered one of the toy models for understanding the localization phenomenon.

Figure 12 shows the SFs for the AA model resolved at different momenta \vec{k} . The path in the Brillouin zone is selected similarly to Fig. 6. Figures 12(a) and 12(c) show the SFs obtained by exact diagonalization of the Hamiltonian via Eq. (A5), while Figs. 12(b) and 12(d) show the SFs $A(\vec{k}, \omega)$ calculated using the AE method. The upper (bottom) row shows the result for the potential strength $\lambda = 0.5$ ($\lambda = 2.5$), as marked additionally in each panel. This choice exemplifies two distinct phases of the AA model.

To simulate the effect of interactions, a relatively high $\epsilon = 0.05$ is chosen in Eq. (A5). However, the value is low enough for the spectral response to remain close to δ -like sharp peaks. Although we are able to study the noninteracting Hamiltonian for much larger systems, we fixed the number of lattice sites $N_s = 400$ ($L_x = L_y = 20$) for clarity. No size effects are visible in the reconstruction performance. The number of frequencies is set to $N = 1000$.

We observe a relatively high similarity of the SFs calculated by the AE to the ones computed exactly, though the

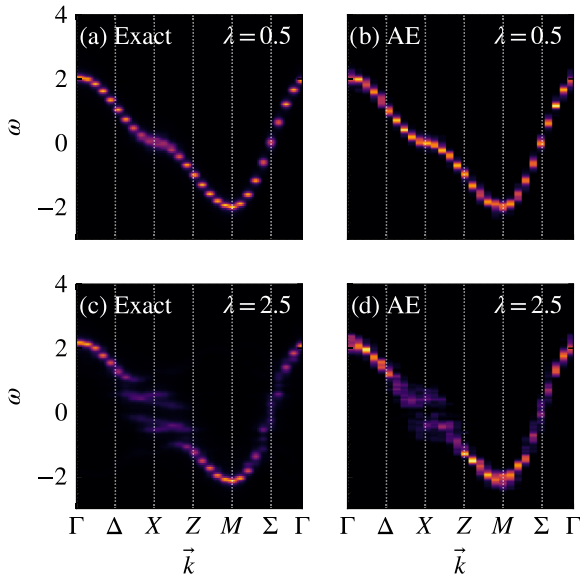


FIG. 12. Momentum-resolved spectra of the Aubry-André model [Eq. (A6)] on a square lattice. Panels (a) and (c) show SFs obtained by exact diagonalization of the Hamiltonian [Eq. (A5)], while panels (b) and (d) show SFs calculated by the AE method. $\lambda = 0.5$ (extended states) is assumed in panels (a) and (b) and $\lambda = 2.5$ (localized states) is assumed in panels (c) and (d). Brighter colors indicate a higher magnitude of the SFs. The white dotted lines indicate high-symmetry points in the Brillouin zone.

former SFs show a slightly higher broadening. This suggests that the pretraining procedure could be improved by including a wider set of artificial SFs that encompass δ -like peaks with a lower spectral weight. Noninteracting SFs included in the pretraining set could further enhance the efficiency of the AE framework. The limitation induced by a variety of considerably sharp peaks in the spectrum requires further study.

APPENDIX B: MAXENT PROCEDURE

The maximum entropy method for analytic continuation uses Bayesian principles to identify the spectral function which, given some Green’s function data, minimizes the functional

$$Q = \frac{1}{2} \chi^2 - \alpha S[A(\omega)]. \quad (\text{B1})$$

Q is the canonical symbol used for this functional, and its role is analogous to the loss \mathcal{L} in machine-learning techniques. Minimizing Q is equivalent to performing a chi-squared fitting, regularized by the Shannon entropy term $S[A(\omega)] = -\int d\omega A(\omega) \ln[A(\omega)/d(\omega)]$. The function $d(\omega)$ is known as the default model, which must be chosen by using prior knowledge about the physical nature of the SF. The parameter α controls the relative strength between the χ^2 and the entropy terms in Q . If $\alpha \gg 1$, then the MaxEnt method would give $d(\omega)$ as the solution for SF [86].

The primary hurdles in using MaxEnt are the identification of the appropriate value of α and the appropriate choice of $d(\omega)$. There have been various approaches to choosing α , including Bryan’s method, which finds the SF that minimizes Eq. (B1) for a range of α values, then gives an average SF,

weighted by the probability of each α [86]. It is difficult to choose a default model, since the structure of the spectral function is *a priori* unknown. One can use general properties, such as whether the system is in a conducting or insulating state, but even these basic properties may be unknown when studying a model with an incomplete phase diagram.

In this paper, we use the MaxEnt code implemented by Levy [35]. We treat α with Bryan’s method, using 60 equally spaced α values in the range $0.1 \leq \alpha \leq 20$. The SF is calculated with 250 equally spaced frequencies, ranging from $-15 < \omega t < 15$, where t sets the scale of the hopping energy. The choice of default model is described in the following sections.

1. MaxEnt for artificial spectral functions

We perform the MaxEnt procedure for Green’s functions (Fig. 4), obtained from artificially generated spectral functions using a uniform default model. This is because there is no physical intuition that allows us to choose a different model.

2. MaxEnt for DQMC at half-filling

For the Green’s functions produced by DQMC at $U/t = 8$, $\mu = 0$, and $\beta = 0.5$ (Fig. 7), the default model is chosen to consist of two Gaussian peaks, centered at $\omega t = \pm 5$ and both with a standard deviation of 1. This is due to the repulsive Hubbard model being likely in a gapped state for these parameters.

3. MaxEnt for DQMC away from half-filling

A consequence of doping the repulsive Hubbard model away from the Mott insulating limits is an asymmetry in the local density of states (LDOS) [87]. With increasing particle doping, the LDOS spectrum shifts to the left to satisfy the sum rules. Since the choice of default model is ambiguous, we perform the analytic continuation with uniform, Lorentzian, and Gaussian default models, while varying the width of the last two models. The choice of optimal spectral functions is based on which default model reproduces the first three moments of the spectral function the closest, as defined in Eq. (12).

APPENDIX C: DQMC

Green’s functions shown in this work are produced with the “bandmott” version of the determinant quantum Monte Carlo code [88] for the cases with a mild average sign. The simulation is performed on a square lattice of side length $L_x = L_y = 16$. Because of symmetries of the lattice, only the lower triangular half of the first quadrant is computed. The inverse temperature is kept at $\beta = 2t$ (mild average sign) and $\beta = 3t$ ($\langle s \rangle \approx 0.3$) and the Hubbard interaction at $U/t = 8$. The Trotter error is fixed at $\Delta\tau = 0.05$ ($\Delta\tau = 0.075$) for the simulations at $\beta = 2t$ ($\beta = 3t$).

1. DQMC at half-filling

The results at half-filling were obtained using 1000 warm-up sweeps, performed before the measurement sweeps. Every next measurement is taken after 10 sweeps through the

TABLE II. Average particle dopings n and average sign $\langle s \rangle$ resulting from the DQMC of the Hubbard model from Eq. (9) away from half-filling ($\mu \neq 0$).

μ	$\langle s \rangle$	n
1.00	0.9714	1.0237
4.00	0.9324	1.3988

auxiliary fields. With the exception of Fig. 7, all data are produced from 40 measurements and 100 different trials ($m_s = 4 \times 10^3$ measurements in total). A “trial” refers to one instance of the DQMC system produced by a random initial seed.

In Fig. 7, the number of measurements is varied, but all other parameters are kept the same. Separate trials are produced by different random seeds. The largest error bars, labeled “ $\sigma = 2.5E - 2$ ” in the figure, result from $m_s = 15$ measurements on one trial, “ $\sigma = 2.6E - 3$ ” is the result of $m_s = 40$ measurements on one trial, “ $\sigma = 2.8E - 4$ ” is produced by performing 40 measurements on 100 different trials ($m_s = 4 \times 10^3$ measurements in total), and “ $\sigma = 8.8E - 5$ ” is produced by performing 40 measurements on 1002 different trials ($m_s \approx 4 \times 10^4$ measurements in total). The $\tilde{A}_{M,A}^\infty$ is obtained from the Green’s functions originating from 40 measurements on 3000 different random seeds, so $m_s = 1.2 \times 10^5$ measurements in total and $\sigma = 4.7 \times 10^{-5}$. The error bar on each $G_{\vec{k}}(\tau)$ is given by the standard error over the measurements.

2. DQMC away from half-filling

The QMC simulations away from half-filling were performed with 2000 warm-up sweeps and $m_s = 5000$ measurement sweeps. The measurement of Green’s functions was done after every 10 measurement sweeps. The Green’s functions were averaged over runs from 20 independent initial configurations of the Hubbard-Stratonovich fields. The MaxEnt was performed at dopings outlined in Table II. The values are obtained directly from the DQMC simulation. For the Lorentzian default model, widths of $\Gamma \in \{0.5, 1.0, 2.0\}$ were used, whereas for Gaussian default model, widths of $\sigma \in \{1.0, 2.0\}$ were used. Out of these, the Gaussian default model with $\sigma = 2.0$ gave the closest results to the sum rules [Eq. (12)]. We compare both AE and MaxEnt methods for different dopings in Fig. 13 by showing the density of states (DOS), i.e., the SFs summed over all momenta, for different electron concentrations. The validity of the analytically continued spectral functions used in Fig. 13 was checked by calculating the imaginary-time Green’s function by the forward integration [Eq. (1)], which matched Green’s functions obtained directly from QMC simulations within at most 1% error.

According to Table I, in Table III we compare the absolute errors of the spectral moments [Eq. (12)] using the Hubbard

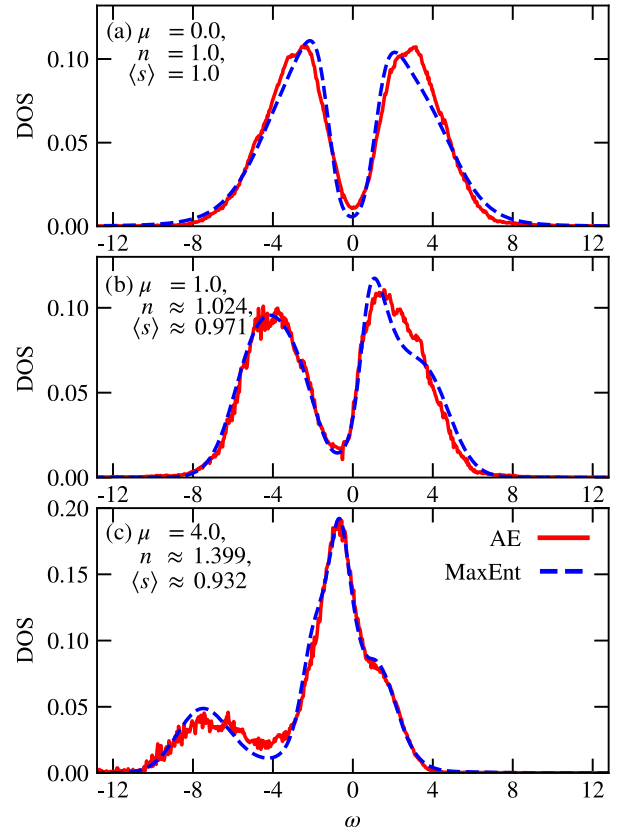


FIG. 13. Density of states calculated with the help of the AE (red lines) and MaxEnt (blue lines) methods for the Hubbard model from Eq. (9) at $U = 8$ on a 16×16 square lattice. The procedures were carried out in three different particle fillings: (a) $n = 1.0$, (b) $n \approx 1.024$, and (c) $n \approx 1.399$.

model away from half-filling $\mu = 1.5$ and $\beta = 3$, $U = 8$ for the MaxEnt and AE frameworks. Again, the AE method outperforms the MaxEnt approach in reconstructing spectral moments. Nevertheless, the errors accumulated for the second moment m_2 are larger in this case compared to the half-filled scenario. This suggests that performing the analytic continuation might be generally more challenging for systems away from $\mu = 0$.

TABLE III. Absolute errors of m th-order moments at the Γ point in the Brillouin zone, obtained with AE and MaxEnt from the QMC SFs away from half-filling at $\mu = 1.5$, $\beta = 3$, and $U = 8$.

m	AE	MaxEnt
0	0.0047	0.0226
1	0.03	0.3130
2	3.19	5.064

[1] S. I. Kabanikhin, *Inverse and Ill-posed Problems* (De Gruyter, Berlin, 2011).

[2] A. Tanaka, A. Tomiya, and K. Hashimoto, in *Deep Learning and Physics*, Mathematical Physics Studies, edited by A. Tanaka,

- A. Tomiya, and K. Hashimoto (Springer, Singapore, 2021), pp. 129–138.
- [3] A. Tarantola, *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation* (Elsevier, Amsterdam, 1987).
- [4] E. Jacquelin, A. Bennani, and P. Hamelin, *J. Sound Vib.* **265**, 81 (2003).
- [5] M. Asim, F. Shamshad, and A. Ahmed, *Blind Image Deconvolution Using Deep Generative Priors* (IEEE Transactions on Computational Imaging, 2020), pp. 1493–1506.
- [6] D. Adke, A. Karnik, H. Berman, and S. Mathi, in *2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST)* (IEEE, Yogyakarta, Indonesia, 2021), pp. 79–83.
- [7] S. Hörmann and F. Jammoul, *J. Multivariate Anal.* **189**, 104886 (2022).
- [8] F. R. De Hoog, J. H. Knight, and A. N. Stokes, *SIAM J. Sci. Stat. Comput.* **3**, 357 (1982).
- [9] J. van Iseghem, *Appl. Numer. Math.* **3**, 529 (1987).
- [10] Y.-K. Kwok and D. Barthez, *Inverse Probl.* **5**, 1089 (1989).
- [11] H. Gzyl, *Appl. Math. Comput.* **73**, 181 (1995).
- [12] T. Scotti and A. Wirgin, *Inverse Probl.* **11**, 1097 (1995).
- [13] T. Scotti and A. Wirgin, *Inverse Probl.* **12**, 1027 (1996).
- [14] M. A. Kant and P. Rudolf Von Rohr, *Int. J. Heat Mass Transf.* **99**, 1 (2016).
- [15] L. B. Lucy, *Astron. J.* **79**, 745 (1974).
- [16] W. H. Richardson, *J. Opt. Soc. Am.* **62**, 55 (1972).
- [17] A. Georges, G. Kotliar, W. Krauth, and M. J. Rozenberg, *Rev. Mod. Phys.* **68**, 13 (1996).
- [18] T. Maier, M. Jarrell, T. Pruschke, and M. H. Hettler, *Rev. Mod. Phys.* **77**, 1027 (2005).
- [19] A. D. Polianin and A. V. Manzhirov, *Handbook of Integral Equations* (CRC, Boca Raton, FL, 1998).
- [20] E. Zappala, A. H. d. O. Fonseca, J. O. Caro, and D. van Dijk, *Nat. Mach. Intell.* (2024), doi: [10.1038/s42256-024-00886-8](https://doi.org/10.1038/s42256-024-00886-8).
- [21] Y. Guan, T. Fang, D. Zhang, and C. Jin, *Int. J. Appl. Comput. Math.* **8**, 87 (2022).
- [22] R. Zhang, M. E. Merkel, S. Beck, and C. Ederer, *Phys. Rev. Res.* **4**, 043082 (2022).
- [23] H. Shao and A. W. Sandvik, *Phys. Rep.* **1003**, 1 (2023).
- [24] C. R. Rao and S. K. Mitra, *Generalized Inverse of Matrices and Its Applications*, Wiley Series in Probability and Mathematical Statistics (Wiley, New York, 1971).
- [25] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, 2nd ed., CMS Books in Mathematics No. 15 (Springer, New York, 2003).
- [26] J. Schött, I. L. M. Loch, E. Lundin, O. Granas, O. Eriksson, and I. Di Marco, *Phys. Rev. B* **93**, 075104 (2016).
- [27] A. W. Sandvik, *Phys. Rev. B* **57**, 10287 (1998).
- [28] J. Otsuki, M. Ohzeki, H. Shinaoka, and K. Yoshimi, *Phys. Rev. E* **95**, 061302(R) (2017).
- [29] K. Yoshimi, *Comput. Phys. Commun.* **244**, 319 (2019).
- [30] Y. Motoyama, K. Yoshimi, and J. Otsuki, *Phys. Rev. B* **105**, 035139 (2022).
- [31] S. R. White, in *Computer Simulation Studies in Condensed Matter Physics III*, Springer Proceedings in Physics Vol. 53, edited by H. K. V. Lotsch, D. P. Landau, K. K. Mon, and H.-B. Schüttler (Springer, Berlin, 1991), pp. 145–153.
- [32] L. Huang and S. Liang, *Phys. Rev. B* **108**, 235143 (2023).
- [33] E. Vitali, M. Rossi, L. Reatto, and D. E. Galli, *Phys. Rev. B* **82**, 174510 (2010).
- [34] E. Pavarini, E. Koch, F. Anders, and M. Jarrell, *Correlated Electrons: From Models to Materials Modeling and Simulation* (Forschungszentrum Jülich, 2012), Vol. 2.
- [35] R. Levy, J. P. F. LeBlanc, and E. Gull, Implementation of the maximum entropy method for analytic continuation, *Comput. Phys. Commun.* **215**, 149 (2017).
- [36] J. Fei, C.-N. Yeh, and E. Gull, *Phys. Rev. Lett.* **126**, 056402 (2021).
- [37] S. Iskakov, A. Hampel, N. Wentzell, and E. Gull, *Comput. Phys. Commun.* **304**, 109299 (2024).
- [38] Z. Huang, E. Gull, and L. Lin, *Phys. Rev. B* **107**, 075151 (2023).
- [39] R. Fournier, L. Wang, O. V. Yazyev, and Q. S. Wu, *Phys. Rev. Lett.* **124**, 056401 (2020).
- [40] H. Yoon, J.-H. Sim, and M. J. Han, *Phys. Rev. B* **98**, 245101 (2018).
- [41] J. Yao, C. Wang, Z. Yao, and H. Zhai, *Machine Learn.: Sci. Technol.* **3**, 025010 (2022).
- [42] R. Blankenbecler, D. J. Scalapino, and R. L. Sugar, *Phys. Rev. D* **24**, 2278 (1981).
- [43] S. R. White, D. J. Scalapino, R. L. Sugar, E. Y. Loh, J. E. Gubernatis, and R. Scalettar, *Phys. Rev. B* **40**, 506 (1989).
- [44] K. Hornik, M. Stinchcombe, and H. White, *Neural Netw.* **2**, 359 (1989).
- [45] H. T. Siegelmann and E. D. Sontag, in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (ACM, Pittsburgh, PA, 1992), pp. 440–449.
- [46] A. Barron, *IEEE Trans. Inf. Theor.* **39**, 930 (1993).
- [47] In the actual calculations, we use the covariance matrix to take into account different uncertainties of different Green’s functions.
- [48] Since there are no uncertainties in the artificially generated SFs, contrary to minimizing the loss function defined in Eq. (5), here we do not use a covariance matrix.
- [49] R. Bellman and R. Bellman, *Dynamic Programming*, Rand Corporation Research Study (Princeton University Press, Princeton, NJ, 1957).
- [50] S. R. White, *Phys. Rev. B* **44**, 4670 (1991).
- [51] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, *J. Mach. Learn. Res.* **11**, 3371 (2010).
- [52] M. Troyer and U.-J. Wiese, *Phys. Rev. Lett.* **94**, 170201 (2005).
- [53] D. Hangleiter, I. Roth, D. Nagaj, and J. Eisert, *Sci. Adv.* **6**, eabb8341 (2020).
- [54] G. Pan and Z. Y. Meng, *Encyc. Condens. Matter Phys.* **1**, 879 (2024).
- [55] M. Claesens and B. De Moor, Hyperparameter search in machine learning, *MIC 2015: The XI Metaheuristics International Conference* (2015).
- [56] M. Ganaie, M. Hu, A. Malik, M. Tanveer, and P. Suganthan, *Eng. Appl. Artif. Intell.* **115**, 105151 (2022).
- [57] A. Buades, B. Coll, and J. M. Morel, *Multiscale Model. Simul.* **4**, 490 (2005).
- [58] M. Krawczyk, J. Pawłowski, M. M. Maška, and K. Roszak, *Phys. Rev. A* **109**, 022405 (2024).
- [59] F. Chollet *et al.*, Keras (2015), <http://keras.io>.
- [60] M. Abadi *et al.*, TensorFlow developers, TensorFlow (2015), <https://tensorflow.org>.

- [61] M. Rigol, V. Dunjko, and M. Olshanii, *Nature (London)* **452**, 854 (2008).
- [62] P. Werner, A. Comanac, L. de' Medici, M. Troyer, and A. J. Millis, *Phys. Rev. Lett.* **97**, 076405 (2006).
- [63] Y. Yu, A. F. Kemper, C. Yang, and E. Gull, Denoising of imaginary time response functions with Hankel projections, *Phys. Rev. Res.* **6**, L032042 (2024).
- [64] D. P. Kingma and M. Welling, *Found. Trends Machine Learn.* **12**, 307 (2019).
- [65] L. Pinheiro Cinelli, M. Araújo Marins, E. A. Barros Da Silva, and S. Lima Netto, *Variational Methods for Machine Learning with Applications to Deep Networks* (Springer International, Cham, 2021), pp. 111–149.
- [66] H. Xu, Y. Feng, J. Chen, Z. Wang, H. Qiao, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, and D. Pei, in *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18* (ACM, Lyon, France, 2018), pp. 187–196.
- [67] R. Yao, C. Liu, L. Zhang, and P. Peng, in *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)* (IEEE, San Francisco, 2019), pp. 1–7.
- [68] W. Xu, H. Sun, C. Deng, and Y. Tan, *Proc. AAAI Conf. Artif. Intell.* **31**, 1 (2017).
- [69] R. F. Mansour, J. Escorcia-Gutierrez, M. Gamarra, D. Gupta, O. Castillo, and S. Kumar, *Pattern Recog. Lett.* **151**, 267 (2021).
- [70] S. J. Wetzels, *Phys. Rev. E* **96**, 022140 (2017).
- [71] A. Khoshaman, W. Vinci, B. Denis, E. Andriyash, H. Sadeghi, and M. H. Amin, *Quantum Sci. Technol.* **4**, 014001 (2018).
- [72] C. Miles, M. R. Carbone, E. J. Sturm, D. Lu, A. Weichselbaum, K. Barros, and R. M. Konik, *Phys. Rev. B* **104**, 235111 (2021).
- [73] A. Baul, N. Walker, J. Moreno, and K.-M. Tam, *Phys. Rev. E* **107**, 045301 (2023).
- [74] H. Goh, S. Sheriffdeen, J. Wittmer, and T. Bui-Thanh, in *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*, Proceedings of Machine Learning Research Vol. 145, edited by J. Bruna, J. Hesthaven, and L. Zdeborova (PMLR, 2022), pp. 386–425.
- [75] K. He, X. Zhang, S. Ren, and J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in *2015 IEEE International Conference on Computer Vision (ICCV)* (Santiago, Chile, 2015), pp. 1026–1034.
- [76] D. P. Kingma and J. Ba, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) [cs].
- [77] S. J. Reddi, S. Kale, and S. Kumar, [arXiv:1904.09237](https://arxiv.org/abs/1904.09237) [cs, math, stat].
- [78] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Adaptive Computation and Machine Learning (MIT Press, Cambridge, MA, 2016).
- [79] L. Boehnke, H. Hafermann, M. Ferrero, F. Lechermann, and O. Parcollet, *Phys. Rev. B* **84**, 075145 (2011).
- [80] S. Aubry and G. André, *Ann. Isr. Phys. Soc.* **3**, 18 (1980).
- [81] H.-j. Li, J.-p. Dou, and G. Huang, *Phys. Rev. A* **89**, 033843 (2014).
- [82] A. Szabo and U. Schneider, *Phys. Rev. B* **101**, 014205 (2020).
- [83] H. Hiramoto and M. Kohmoto, *Phys. Rev. B* **40**, 8225 (1989).
- [84] Y. E. Kraus, Y. Lahini, Z. Ringel, M. Verbin, and O. Zeitlinger, *Phys. Rev. Lett.* **109**, 106402 (2012).
- [85] A.-K. Wu, [arXiv:2109.07062](https://arxiv.org/abs/2109.07062) [cond-mat].
- [86] M. Jarrell and J. Gubernatis, *Phys. Rep.* **269**, 133 (1996).
- [87] M. Randeria, R. Sensarma, N. Trivedi, and F.-C. Zhang, *Phys. Rev. Lett.* **95**, 137001 (2005).
- [88] Z. Bai, W. Chen, R. Scalettar, and I. Yamazaki, in *Proceedings of the 4th International Congress of Chinese Mathematician (ICCM)*, edited by L. Ji, K. Liu, L. Yang, S.-T. Yau III (Higher Education Press : International Press, Beijing, Somerville, MA, 2008), p. 253.