Scalar field restricted Boltzmann machine as an ultraviolet regulator

Gert Aarts,[1,2,*] Biagio Lucini,[3,†] and Chanju Park[1,‡]

[1] *Department of Physics, Swansea University, Swansea SA2 8PP, United Kingdom*
[2] *European Centre for Theoretical Studies in Nuclear Physics and Related Areas (ECT*)*
and Fondazione Bruno Kessler, 38123 Villazzano (TN), Italy
[3] *Department of Mathematics, Swansea University, Swansea, SA2 8PP, United Kingdom*

(Received 23 October 2023; accepted 30 January 2024; published 29 February 2024)

Restricted Boltzmann machines (RBMs) are well-known tools used in machine learning to learn probability distribution functions from data. We analyze RBMs with scalar fields on the nodes from the perspective of lattice field theory. Starting with the simplest case of Gaussian fields, we show that the RBM acts as an ultraviolet regulator, with the cutoff determined by either the number of hidden nodes or a model mass parameter. We verify these ideas in the scalar field case, where the target distribution is known, and explore implications for cases where it is not known using the MNIST dataset. We also demonstrate that infrared modes are learnt quickest.

DOI: 10.1103/PhysRevD.109.034521

I. INTRODUCTION

In recent years machine learning (ML) has gained tremendous popularity in the physical sciences [1]. In theoretical nuclear and high-energy physics, ML is applied to a wide range of problems, see e.g. the reviews [2,3]. In lattice field theory (LFT), there are applications to all aspects of LFT computations [4], with the development of normalizing flows to generate field configurations a particularly active area of research [5,6]. From a theoretical perspective, it is of interest to explore synergies between ML on the one hand and statistical physics and LFT on the other hand, as many ML problems can be studied using the tools of the latter, see e.g. Ref. [7]. The connection between neural networks, Markov random fields and (Euclidean) lattice field theory have indeed not gone unnoticed, leading to the notions of quantum field-theoretic machine learning (QFT/ML) [8] and neural network/QFT correspondence [9,10]. Further exploration of this connection may be fruitful in both directions, providing potential insights relevant to both the ML and the LFT/QFT communities.

In this paper, we take a step in this direction by considering one of the simplest generative ML models, the restricted Boltzmann machine (RBM) [11,12]. We analyze the RBM with continuous fields as degrees of freedom from the perspective of a Euclidean LFT and give a complete understanding in the case of Gaussian fields. We verify our analytical insights using simple scalar field theories in one and two dimensions, for which the target distribution is known, and also the Modified National Institute of Standards and Technology database (MNIST) dataset, to demonstrate that our findings are indeed relevant for typical ML datasets without known target distributions. We are in particular interested in the choice of “architecture,” which admittedly is quite straightforward for an RBM, namely the number of hidden nodes as well as the choice of certain hyperparameters. Our main conclusion is that the scalar field RBM acts as an ultraviolet regulator, with the cutoff determined by either the number of hidden nodes or a model mass parameter. We will make clear what this implies for the MNIST data set, but note here already that in QFT language the MNIST dataset is ultraviolet divergent and infrared safe.

The paper is organized as follows. In Sec. II we introduce scalar field RBMs from the perspective of LFT and give some exact solutions for the Gaussian case. The standard equations to train an RBM are summarized in Sec. III. In Sec. IV we analyze these equations analytically and work out some simple examples in detail. The findings of this section will be further explored in the two following sections. First, we consider as target theories free scalar fields in one and two dimensions in Sec. V, for which the target distribution is known. In Sec. VI we validate our findings for a dataset with an unknown distribution, namely the MNIST dataset. Options to add interactions are discussed in Sec. VII. A summary is given in the final section. Appendix A contains some more details on the algorithm

[*] g.aarts@swansea.ac.uk
[†] b.lucini@swansea.ac.uk
[‡] c.j.park@swansea.ac.uk

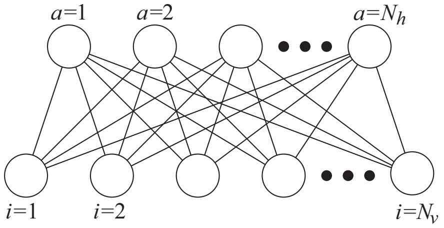


FIG. 1. Bipartite graph, with N_v (N_h) nodes in the visible (hidden) layer.

employed, while in Appendix B the Kullback-Leibler divergence is evaluated in the Gaussian case.

II. SCALAR FIELDS ON A BIPARTITE GRAPH

RBMs are defined on a bipartite graph, consisting of one visible layer (with N_v nodes) and one hidden layer (with N_h nodes), see Fig. 1. Importantly, there are no connections within each layer, only between the two layers. The degrees of freedom living on the nodes can be discrete, as in an Ising model, continuous or mixed; Ref. [13] is a useful review.

In this section, we consider an RBM from the viewpoint of lattice field theory. We consider continuous fields and denote these as ϕ_i ($i = 1, \ldots, N_v$) for the visible layer and h_a ($a = 1, \ldots, N_h$) for the hidden layer. The layers are coupled via bilinear terms and involve the $N_v \times N_h$ weight matrix W, as

$$\phi^T W h = \sum_{i=1}^{N_v} \sum_{a=1}^{N_h} \phi_i W_{ia} h_a. \tag{1}$$

The aim is to describe a probability distribution $p(\phi)$ on the visible layer, constructed by integrating over the hidden nodes in the joint probability distribution $p(\phi, h)$, as follows:

$$p(\phi) = \int Dh\, p(\phi, h), \qquad p(\phi, h) = \frac{\exp(-S(\phi, h))}{Z}, \tag{2}$$

where we have denoted the "energy" in the exponential as an action (following LFT notation) and the partition function reads

$$Z = \int D\phi Dh \exp(-S(\phi, h)). \tag{3}$$

The integrals are over all nodes,

$$\int D\phi = \prod_{i=1}^{N_v} \int_{-\infty}^{\infty} d\phi_i, \qquad \int Dh = \prod_{a=1}^{N_h} \int_{-\infty}^{\infty} dh_a. \tag{4}$$

Due to the absence of intralayer connections, the action takes a simple form in general:

$$S(\phi, h) = V_\phi(\phi) + V_h(h) - \phi^T W h, \tag{5}$$

where the two potentials can be any function (as long as the integrals are well defined) and be node dependent, i.e.,

$$V_\phi(\phi) = \sum_i V_i^{(\phi)}(\phi_i), \quad V_h(h) = \sum_a V_a^{(h)}(h_a). \tag{6}$$

Since there is no coupling between nodes within a layer, there is no "kinetic" or nearest-neighbor term; these are only generated via the coupling to the other layer.

To proceed, a natural starting point is to consider quadratic potentials, i.e., free fields (we discuss interactions in Sec. VII). We hence consider as action,

$$\begin{aligned} S(\phi, h) &= \sum_i \frac{1}{2}\mu^2 \phi_i^2 + \sum_a \frac{1}{2\sigma_h^2}(h_a - \eta_a)^2 - \sum_{i,a} \phi_i W_{ia} h_a, \\ &= \frac{1}{2}\mu^2 \phi^T \phi + \frac{1}{2\sigma_h^2}(h - \eta)^T (h - \eta) - \phi^T W h. \end{aligned} \tag{7}$$

A few comments are in order. We have denoted the prefactor as a mass term (μ^2) in the case of ϕ and as a variance ($1/\sigma_h^2$) in the case of h; this is inessential, but emphasizes that the model on the visible layer is ultimately the one we are interested in. Both μ^2 and σ_h^2 are independent of the node; this is sufficient, as node dependence can be introduced via the weight matrix W, as we will see shortly. Finally, a source (or bias) η_a is introduced in the hidden layer but not in the visible layer; again this is sufficient, as a nonzero bias breaks both symmetries, $h \to -h$, $\phi \to -\phi$.

Integrating out the hidden nodes then leads to the following distribution on the visible layer,

$$\begin{aligned} p(\phi) &= \int Dh\, p(\phi, h), \\ &= \frac{1}{Z} \exp\left(-\frac{1}{2}\phi^T K \phi + \phi^T J\right), \end{aligned} \tag{8}$$

with

$$K \equiv \mu^2 1\!\!1 - \sigma_h^2 W W^T, \qquad J \equiv W\eta, \tag{9}$$

and where Z now reads

$$Z = \int D\phi \exp\left(-\frac{1}{2}\phi^T K \phi + \phi^T J\right). \tag{10}$$

We note therefore that the distribution on the visible layer resembles a generating function for a scalar field theory, with the possibility of all-to-all bilinear interactions between the fields via the nonlocal kernel K, and the bias resulting in a source term J coupled to ϕ. The connected two-point function or propagator is given by

$$\langle\phi_i\phi_j\rangle - \langle\phi_i\rangle\langle\phi_j\rangle = K^{-1}_{ij}. \tag{11}$$

The hidden layer has provided auxiliary degrees of freedom to establish correlations between the visible nodes.

To continue the discussion we now assume the target probability distribution $p_{\text{target}}(\phi)$ is known and Gaussian, such that we can solve the RBM explicitly, i.e., we give explicit expressions for the weight matrix W and the bias η. We denote the target kernel as K^ϕ and consider the symmetric case ($\phi\to-\phi$, $\eta=J=0$) for simplicity. Since K^ϕ is a real and symmetric matrix, it can be diagonalized; for the theory to exist, all its eigenvalues are assumed to be semipositive. The RBM is then solved by equating the two kernels, $K^\phi = K$, i.e.,

$$K^\phi = \mu^2 1\!\!1 - \sigma_h^2 WW^T, \tag{12}$$

which implies

$$WW^T = \frac{1}{\sigma_h^2}(\mu^2 1\!\!1 - K^\phi) \equiv \mathcal{K}. \tag{13}$$

Since WW^T is semipositive, we find conditions on the parameter μ^2, namely

$$\begin{aligned}\mu^2/\sigma_h^2 &\geq \max\,[\text{eigenvalues}(WW^T)],\\ \mu^2 &\geq \max\,[\text{eigenvalues}(K^\phi)].\end{aligned} \tag{14}$$

Consider now the case that $N_h = N_v$. It is then easy to find some solutions for W, given that the rhs of Eq. (13) is symmetric and positive:

(1) The rhs of Eq. (13) can be decomposed in a Cholesky decomposition, $\mathcal{K} = LL^T$, where L is a lower triangular matrix with real and positive diagonal entries. The solution is then simply $W = L$. The triangular structure means that hidden node a connects to visible nodes with $a \leq i$ only.

(2) The rhs of Eq. (13) can be diagonalized via an orthogonal transformation,

$$\mathcal{K} = ODO^T = O\sqrt{D}O^T O\sqrt{D}O^T, \tag{15}$$

yielding the symmetric solution

$$W = W^T = O\sqrt{D}O^T. \tag{16}$$

Hence we have found two explicit solutions. Additional solutions are found from either of the above by a right multiplication of W by an orthogonal transformation, rotating the hidden nodes,

$$W \to WO_R^T, \qquad h \to O_R h, \qquad O_R^T O_R = 1\!\!1, \tag{17}$$

since O_R drops out of the combination WW^T.

We conclude therefore that an infinite number of solutions is present. These can be constrained by imposing further conditions on W, as in the first two cases above. We will discuss this degeneracy further below.

Next, we may consider the case where $N_h < N_v$. From Eq. (13) it is clear that the accuracy of reproducing the target distribution depends on the ranks of the matrices involved. We find

$$\text{rank}(WW^T) \leq \min\,(N_v, N_h), \qquad \text{rank}(\mathcal{K}) \leq N_v. \tag{18}$$

Only when the ranks are equal will the target distribution be reproducible; this is particularly relevant when choosing $N_h \ll N_v$. Below we will consider in detail what happens of either of the two conditions found so far, i.e., Eq. (14) and $\text{rank}(WW^T) = \text{rank}(\mathcal{K})$ is not valid.

III. TRAINING RBM PARAMETERS

The exact solutions above are only useful when the target model is a known Gaussian model and $N_h = N_v$. In general, the target distribution is not known and one has to learn from a finite dataset. The training of the model is then done by maximizing the log-likelihood function $\mathcal{L}(\theta|\phi)$. The learnable parameters are collectively indicated as $\theta = \{W, \eta, \mu^2\}$. Note that we will consider the case of unbroken symmetry and hence the bias is taken to be zero throughout, $\eta_a = 0$. We are hence concerned with determining the weight matrix W and the mass parameter μ^2.

The model distribution is given by Eq. (8), with $J = 0$. Given data consisting of N_{conf} configurations, labeled as $\phi^{(d)}, d = 1, \ldots, N_{\text{conf}}$, the log-likelihood function of the model is written as

$$\begin{aligned}\mathcal{L}(\phi|\theta) &= \frac{1}{N_{\text{conf}}}\sum_{d=1}^{N_{\text{conf}}} \log p_{\text{model}}(\phi^{(d)};\theta),\\ &= -\frac{1}{N_{\text{conf}}}\sum_{d=1}^{N_{\text{conf}}}\left(\frac{1}{2}\phi^{(d)T}K\phi^{(d)} + \ln Z\right).\end{aligned} \tag{19}$$

This log-likelihood function can be optimized with gradient ascent algorithms, where the gradient is taken with respect to the coupling matrix W and the mass parameter μ^2. Explicitly,

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial W_{ia}} &= \frac{1}{N_{\text{conf}}}\sum_d\sum_j \sigma_h^2\phi_i^{(d)}W_{aj}\phi_j^{(d)} - \sum_j \sigma_h^2\langle\phi_i W_{aj}\phi_j\rangle_{\text{model}},\\ &= \sigma_h^2\sum_j\left(\frac{1}{N_{\text{conf}}}\sum_d \phi_i^{(d)}W_{aj}\phi_j^{(d)} - \langle\phi_i W_{aj}\phi_j\rangle_{\text{model}}\right)\\ &= \sigma_h^2\sum_j(C_{ij}^{\text{target}} - C_{ij}^{\text{model}})W_{ja},\end{aligned} \tag{20}$$

where the two-point correlation matrices for the data (i.e., the target) and the model are given, respectively, by

$$C_{ij}^{\mathrm{target}} = \frac{1}{N_{\mathrm{conf}}}\sum_{d=1}^{N_{\mathrm{conf}}} \phi_i^{(d)}\phi_j^{(d)} = \langle \phi_i\phi_j\rangle_{\mathrm{target}} \equiv K_{\phi ij}^{-1},$$
$$C_{ij}^{\mathrm{model}} = \langle \phi_i\phi_j\rangle_{\mathrm{model}} = K_{ij}^{-1}. \tag{21}$$

Similarly, for μ^2 one finds

$$\frac{\partial\mathcal{L}}{\partial\mu^2} = -\frac{1}{2}\sum_i(\langle\phi_i^2\rangle_{\mathrm{target}} - \langle\phi_i^2\rangle_{\mathrm{model}}). \tag{22}$$

When all the data is available, one is able to evaluate the two-point function by summing over configurations before training the RBM. This would yield the target two-point function, computed via the data. In the numerical implementations below, we will analyze the properties of this two-point function further, since the matrix sizes are such that this is feasible. Alternatively, we may consider the case where the target distribution $p_{\mathrm{target}}(\phi)$ is known and the correlation matrix C_{ij}^{target} of the target theory is obtainable. In that case, there is no need to use data but one can use the correlation function directly. It should be noted that in general the correlation matrix C_{ij}^{target} is not directly accessible due to computational complexity, even if the analytical form of the target distribution is known.

If the target distribution is known, then the same equations can also be derived by extremizing the Kullback-Leibler (KL) divergence,

$$\mathrm{KL}(p_{\mathrm{target}}||p_{\mathrm{model}}) = \int D\phi\, p_{\mathrm{target}}(\phi)\log\frac{p_{\mathrm{target}}(\phi)}{p_{\mathrm{model}}(\phi,\theta)}, \tag{23}$$

keeping in mind that only the model distribution depends on the learnable parameters θ. With the distribution given by Eq. (8) and the θ dependence contained in the kernel K only (recall that $J = 0$), extremizing with respect to θ then yields

$$\frac{\partial}{\partial\theta}\mathrm{KL}(p_{\mathrm{target}}||p_{\mathrm{model}}) = \frac{1}{2}\left\langle\phi^T\frac{\partial K}{\partial\theta}\phi\right\rangle_{\mathrm{target}} - \frac{1}{2}\left\langle\phi^T\frac{\partial K}{\partial\theta}\phi\right\rangle_{\mathrm{model}}, \tag{24}$$

which yields the same equations for W and μ^2 as above, but with the opposite sign, as the KL divergence is minimized.

In actual applications, the gradients are used in a discretized update of the form

$$\theta_{n+1} = \theta_n + \eta_n\frac{\partial\mathcal{L}}{\partial\theta}, \tag{25}$$

where η_n is the, possibly time-dependent, learning rate. Details of the commonly used persistent contrastive divergence algorithm and time-dependent learning rate can be found in Appendix A.

IV. SEMIANALYTICAL SOLUTION

A. Singular value decomposition

Before solving the RBM numerically, we aim to gain analytical insight in the update equations using a singular decomposition for the weight matrix in the continuous time limit [13].

The update equations for the weight matrix W and the mass term μ^2, in the continuous time limit, read [see Eqs. (20)–(22)],

$$\dot{W} = \sigma_h^2[K_\phi^{-1} - K^{-1}]W, \tag{26}$$

$$\dot{\mu}^2 = -\frac{1}{2}\mathrm{tr}K_\phi^{-1} + \frac{1}{2}\mathrm{tr}K^{-1}, \tag{27}$$

with the two-point functions (or propagators)

$$K_{\phi ij}^{-1} = \langle\phi_i\phi_j\rangle_{\mathrm{target}}, \quad \mathrm{tr}K_\phi^{-1} = \sum_{i=1}^{N_v}\langle\phi_i\phi_i\rangle_{\mathrm{target}}, \tag{28}$$

$$K_{ij}^{-1} = \langle\phi_i\phi_j\rangle_{\mathrm{model}}, \qquad \mathrm{tr}K^{-1} = \sum_{i=1}^{N_v}\langle\phi_i\phi_i\rangle_{\mathrm{model}}. \tag{29}$$

Recall that $\langle\phi_i\rangle = 0$. The dot denotes the time derivative. We remind the reader that both K and K_ϕ are symmetric $N_v \times N_v$ matrices and that the weight matrix W is of size $N_v \times N_h$. We assume $N_h \leq N_v$. The RBM (model) kernel is

$$K = \mu^2 1\!\!1 - \sigma_h^2 WW^T, \tag{30}$$

where σ_h^2 is the variance of the hidden nodes.

We use the singular value decomposition to write W as

$$W = U\Xi V^T, \quad UU^T = 1\!\!1_{N_v\times N_v}, \quad VV^T = 1\!\!1_{N_h\times N_h}, \tag{31}$$

where U is an orthogonal $N_v \times N_v$ matrix, V is an orthogonal $N_h \times N_h$ matrix, and Ξ is the rectangular $N_v \times N_h$ matrix with the (ordered) singular values ξ_a $(a = 1, \ldots, N_h)$ on the diagonal. The RBM kernel then takes the form

$$K = \mu^2 1\!\!1 - \sigma_h^2 U\Xi\Xi^T U^T,$$
$$= U[\mu^2 1\!\!1 - \sigma_h^2\Xi\Xi^T]U^T \equiv UD_K U^T, \tag{32}$$

with the diagonal matrix

$$D_K = \mathrm{diag}(\underbrace{\mu^2 - \sigma_h^2\xi_1^2, \mu^2 - \sigma_h^2\xi_2^2, \ldots, \mu^2 - \sigma_h^2\xi_{N_h}^2}_{N_h}, \underbrace{\mu^2, \ldots, \mu^2}_{N_v - N_h}). \tag{33}$$

Note that the existence of the model requires that $\mu^2 > \sigma_h^2\xi_1^2$, with ξ_1 the largest singular value of W. Equation (33) demonstrates that only the first N_h eigenvalues can potentially be learnt, with the remaining $N_v - N_h$ eigenvalues frozen at the higher scale μ^2.

The symmetric target kernel K_ϕ and the corresponding two-point function K_ϕ^{-1} can be diagonalized via an orthogonal transformation as

$$K_\phi = O_\phi D_\phi O_\phi^T, \quad K_\phi^{-1} = O_\phi D_\phi^{-1} O_\phi^T, \quad O_\phi O_\phi^T = 11_{N_v \times N_v}, \tag{34}$$

where the eigenvalues of K_ϕ are assumed to be positive again.

The rhs of Eq. (26) can now be written as

$$\sigma_h^2[K_\phi^{-1} - K^{-1}]W = U\sigma_h^2[U^T O_\phi D_\phi^{-1} O_\phi^T U - D_K^{-1}]\Xi V^T. \tag{35}$$

The term within the brackets will be encountered frequently below and hence we honor it with a new symbol,

$$\Lambda \equiv U^T O_\phi D_\phi^{-1} O_\phi^T U - D_K^{-1} = \Lambda^T. \tag{36}$$

The evolution equation for W can then be compactly written as

$$\dot{W} = \sigma_h^2 U\Lambda\Xi V^T, \qquad \dot{W}^T = \sigma_h^2 V\Xi^T\Lambda U^T. \tag{37}$$

We note that Λ drives the evolution in the learning process: it vanishes when the basis on the visible layer is aligned with the basis for the data $(U \to O_\phi)$ and the eigenvalues, or widths of the Gaussians, are correctly determined $(D_K \to D_\phi)$. One may note that Λ does not depend on V, which acts on the hidden nodes, resulting in the degeneracy discussed in Sec. II: any rotation of the hidden nodes leaves the solution on the visible layer invariant and the learning stops when $\Lambda \to 0$, irrespective of what V is.

B. Learning dynamics

Having defined the needed quantities, we are now in a position to determine the learning dynamics of W in detail, i.e., the evolution of U, V, and the singular values Ξ. We consider separately

$$WW^T = U\Xi\Xi^T U^T, \qquad W^T W = V\Xi^T\Xi V^T. \tag{38}$$

Taking the derivative of the first product gives

$$\frac{d}{dt}(WW^T) = \dot{U}\Xi\Xi^T U^T + U\Xi\Xi^T\dot{U}^T + U\frac{d}{dt}(\Xi\Xi^T)U^T. \tag{39}$$

On the other hand, Eq. (37) gives

$$\begin{aligned}\frac{d}{dt}WW^T &= \dot{W}W^T + W\dot{W}^T,\\ &= \sigma_h^2 U\Lambda\Xi\Xi^T U^T + \sigma_h^2 U\Xi\Xi^T\Lambda U^T.\end{aligned} \tag{40}$$

Conjugating both equations with U^T and U then yields

$$U^T\dot{U}\Xi\Xi^T + \Xi\Xi^T\dot{U}^T U + \frac{d}{dt}(\Xi\Xi^T) = \sigma_h^2\Lambda\Xi\Xi^T + \sigma_h^2\Xi\Xi^T\Lambda. \tag{41}$$

Since $U^T\dot{U} = -\dot{U}^T U$ is skew symmetric (due to U being orthogonal) and $\Xi\Xi^T$ is diagonal, it is easy to see that

$$U^T\dot{U}\Xi\Xi^T + \Xi\Xi^T\dot{U}^T U = U^T\dot{U}\Xi\Xi^T - \Xi\Xi^T U^T\dot{U} \tag{42}$$

is a symmetric matrix with zeroes on the diagonal. Equation (41) then decomposes into one equation for the diagonal elements, determining the singular values, and one for the off-diagonal ones, determining U, namely

$$\frac{d}{dt}(\Xi\Xi^T) = \sigma_h^2\Lambda_d\Xi\Xi^T + \sigma_h^2\Xi\Xi^T\Lambda_d = 2\sigma_h^2\Lambda_d\Xi\Xi^T, \tag{43}$$

$$U^T\dot{U}\Xi\Xi^T - \Xi\Xi^T U^T\dot{U} = \sigma_h^2(\Lambda - \Lambda_d)\Xi\Xi^T + \sigma_h^2\Xi\Xi^T(\Lambda - \Lambda_d), \tag{44}$$

where

$$\Lambda_d = \mathrm{diag}(\Lambda). \tag{45}$$

Repeating the same analysis for $W^T W$ gives nearly identical equations in the $N_h \times N_h$ subspace, namely

$$\frac{d}{dt}(\Xi^T\Xi) = 2\sigma_h^2\Xi^T\Lambda_d\Xi, \tag{46}$$

$$V^T\dot{V}\Xi^T\Xi - \Xi^T\Xi V^T\dot{V} = 2\sigma_h^2\Xi^T(\Lambda - \Lambda_d)\Xi. \tag{47}$$

Note that

$$\Xi\Xi^T = \mathrm{diag}(\xi_1^2, \xi_2^2, \ldots, \xi_{N_h}^2, 0, \ldots, 0), \tag{48}$$

$$\Xi^T\Xi = \mathrm{diag}(\xi_1^2, \xi_2^2, \ldots, \xi_{N_h}^2). \tag{49}$$

The equations for $\Xi\Xi^T$ and $\Xi^T\Xi$ yield identical equations for the N_h singular values.

The equation for μ^2 finally reads, in this notation,

$$\dot{\mu}^2 = -\frac{1}{2}\mathrm{tr}\Lambda = -\frac{1}{2}\mathrm{tr}\Lambda_d. \tag{50}$$

Keeping μ^2 fixed, it is easy to see that σ_h^2 can be absorbed in the time parameter ($\tilde{t} = t\sigma_h^2$) and the singular values, see Eq. (33); hence it does not add any freedom to the model. When μ^2 is learnt as well, its time evolution will depend on σ_h^2, after rescaling time as $t \to \tilde{t}$.

As noted, V does not appear in the driving term Λ. Hence V simply follows the evolution, until $\Lambda - \Lambda_d \to 0$, see Eq. (46). For square matrices, $N_h = N_v$, this redundancy can be removed by choosing W to be symmetric ($V = U$) or by enforcing W to be of the lower (or upper) triangular form (Cholesky decomposition of WW^T), see Sec. II.

C. Simple examples

1. $N_v = N_h = 2$

The simple example of two visible and two hidden nodes can be worked out in detail. We will note a number of characteristics which remain relevant also for larger systems.

First we note that U, V, and O_ϕ are all 2×2 rotation matrices; we denote the angles as θ_U, θ_V, and θ_0, respectively. Then one notes that

$$U^T\dot{U} = \dot{\theta}_U \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \tag{51}$$

and the same for $V^T\dot{V}$, with $\dot{\theta}_V$. Finally, the combination $O_\phi^T U$ is a rotation over an angle $\Delta\theta = \theta_U - \theta_0$.

We denote the two eigenvalues of the target kernel K_ϕ with $\kappa_{1,2}$ and of the RBM kernel K with $\lambda_{1,2} = \mu^2 - \sigma_h^2\xi_{1,2}^2$. This yields the driving term,

$$\Lambda = \begin{pmatrix} \frac{1}{\kappa_1}\cos^2\Delta\theta + \frac{1}{\kappa_2}\sin^2\Delta\theta - \frac{1}{\lambda_1} & \left(\frac{1}{\kappa_2} - \frac{1}{\kappa_1}\right)\cos\Delta\theta\sin\Delta\theta \\ \left(\frac{1}{\kappa_2} - \frac{1}{\kappa_1}\right)\cos\Delta\theta\sin\Delta\theta & \frac{1}{\kappa_2}\cos^2\Delta\theta + \frac{1}{\kappa_1}\sin^2\Delta\theta - \frac{1}{\lambda_2} \end{pmatrix}. \tag{52}$$

Putting everything together then gives the following equations:

$$\dot{\xi}_1 = \sigma_h^2\left(\frac{1}{\kappa_1}\cos^2\Delta\theta + \frac{1}{\kappa_2}\sin^2\Delta\theta - \frac{1}{\mu^2 - \sigma_h^2\xi_1^2}\right)\xi_1, \tag{53}$$

$$\dot{\xi}_2 = \sigma_h^2\left(\frac{1}{\kappa_2}\cos^2\Delta\theta + \frac{1}{\kappa_1}\sin^2\Delta\theta - \frac{1}{\mu^2 - \sigma_h^2\xi_2^2}\right)\xi_2, \tag{54}$$

and

$$\dot{\Delta\theta} = \sigma_h^2\frac{\xi_1^2 + \xi_2^2}{\xi_1^2 - \xi_2^2}\rho, \tag{55}$$

$$\dot{\theta}_V = 2\sigma_h^2\frac{\xi_1\xi_2}{\xi_1^2 - \xi_2^2}\rho, \tag{56}$$

$$\dot{\mu}^2 = \frac{1}{2}\left(\frac{1}{\mu^2 - \sigma_h^2\xi_1^2} + \frac{1}{\mu^2 - \sigma_h^2\xi_2^2} - \frac{1}{\kappa_1} - \frac{1}{\kappa_2}\right), \tag{57}$$

where

$$\rho = \left(\frac{1}{\kappa_2} - \frac{1}{\kappa_1}\right)\cos\Delta\theta\sin\Delta\theta. \tag{58}$$

These equations have several fixed points. The difference of angles is given by $\Delta\theta = 0, \pi/2$. Which of these is selected depends on which eigenvalue $\kappa_{1,2}$ is smaller. Note that the SVD decomposition orders the singular values, $\xi_1 > \xi_2$. The equations have fixed points at $\sigma_h^2\xi_{1,2}^2 = \mu^2 - \kappa_{1,2}$ and at $\xi_{1,2}^2 = 0$. We consider first the case of fixed μ^2. The actual realization depends on the ordering of $\kappa_{1,2}$ and μ^2. We find

$$\begin{aligned} &\mu^2 > \kappa_2 > \kappa_1: \ \Delta\theta = 0, \\ &\mu^2 - \sigma_h^2\xi_1^2 = \kappa_1, \qquad \mu^2 - \sigma_h^2\xi_2^2 = \kappa_2, \end{aligned} \tag{59}$$

$$\begin{aligned} &\mu^2 > \kappa_1 > \kappa_2: \ \Delta\theta = \pi/2, \\ &\mu^2 - \sigma_h^2\xi_1^2 = \kappa_2, \qquad \mu^2 - \sigma_h^2\xi_2^2 = \kappa_1. \end{aligned} \tag{60}$$

(The fixed points at $\xi_{1,2}^2 = 0$ are unstable.) This is illustrated in Fig. 2 (top row). In this case, both eigenvalues are learnt correctly. If μ^2 is smaller than an eigenvalue, then it cannot be reproduced and is replaced by μ^2,

$$\kappa_2 > \mu^2 > \kappa_1: \Delta\theta = 0, \quad \mu^2 - \sigma_h^2\xi_1^2 = \kappa_1, \quad \xi_2 = 0, \tag{61}$$

$$\kappa_1 > \mu^2 > \kappa_2: \Delta\theta = \pi/2, \quad \mu^2 - \sigma_h^2\xi_1^2 = \kappa_2, \quad \xi_2 = 0, \tag{62}$$

see Fig. 2 (middle row). In this case, only the smallest eigenvalue is learnt, while the other one evolves to μ^2 [see also Eq. (32)].

In case μ^2 is smaller than all eigenvalues, $\mu^2 < \kappa_{1,2}$, the eigenmodes cannot be reproduced and are replaced by μ^2, with $\xi_1 = \xi_2 = 0$. Finally, we remark again that θ_V simply evolves until $\rho \to 0$, but it does not influence the learning of the other parameters.

The actual eigenvalues may not be known, and one may choose μ^2 to be too low, as in the second example above. This can be evaded by learning μ^2 itself, using Eq. (57). The system is now overparametrized, with $\xi_{1,2}$ and μ^2 being learnt to reproduce $\kappa_{1,2}$. In this case one finds that the

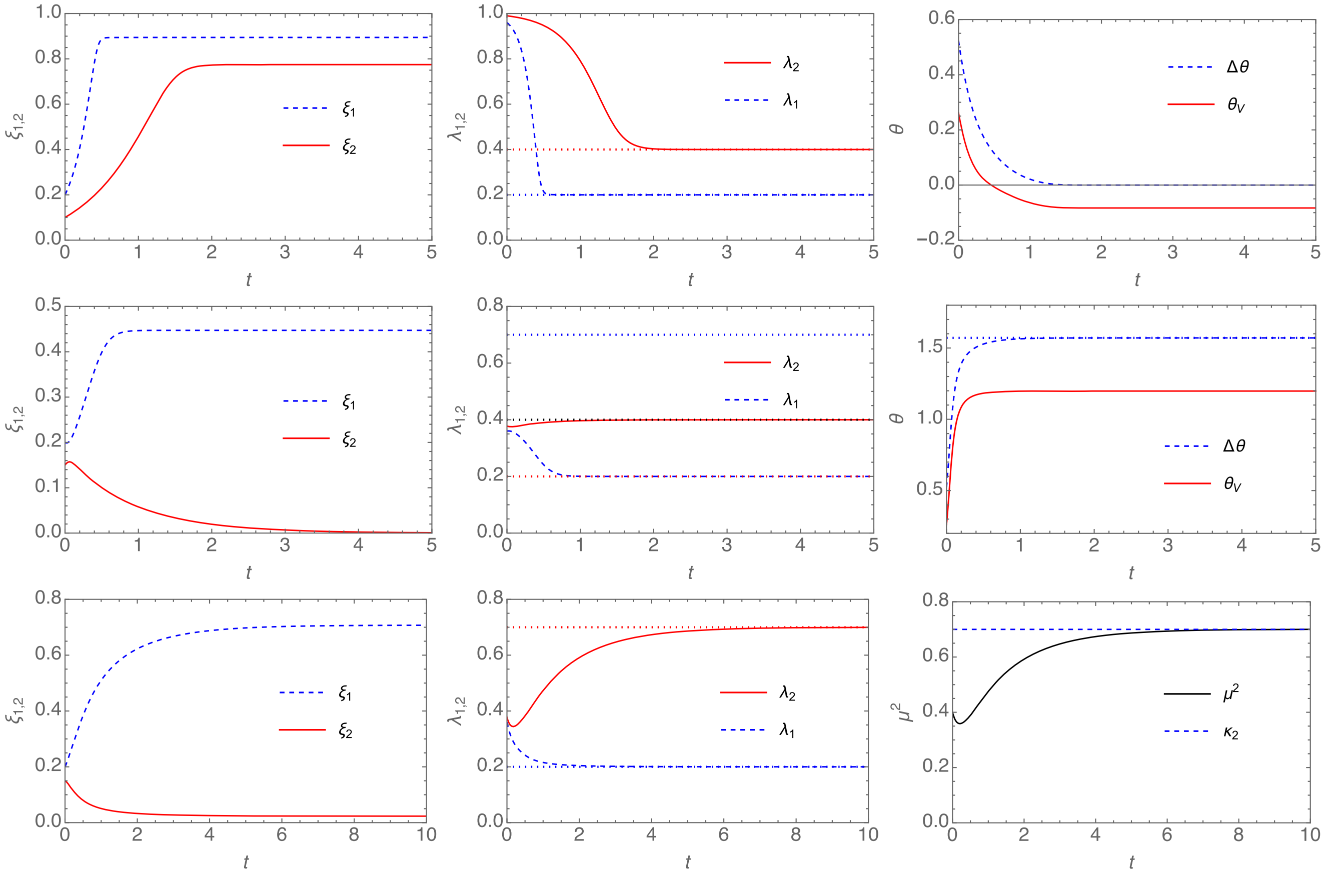


FIG. 2. Top row: learning evolution for the case $\mu^2 > \kappa_2 > \kappa_1$, specifically $\kappa_1 = 0.2$, $\kappa_2 = 0.4$, $\mu^2 = 1$, of the singular values (left), eigenvalues (middle), and angles (right). Middle row: as above, for the case $\kappa_1 > \mu^2 > \kappa_2$, specifically $\kappa_1 = 0.7$, $\kappa_2 = 0.2$, $\mu^2 = 0.4$. Bottom row: as above, including a time dependent μ^2 (right), for the case $\kappa_2 > \mu^2 > \kappa_1$, specifically $\kappa_1 = 0.2$, $\kappa_2 = 0.7$, $\mu^2(0) = 0.4$. In all cases, $\sigma_h^2 = 1$.

eigenvalues are reproduced, irrespective of the initial value of μ^2, see Fig. 2 (bottom row). Note that one of the singular values decreases since μ^2 itself increases towards the largest eigenvalue.

2. Approach to the fixed point

To understand the evolution towards the fixed point, a simple linearization suffices. We consider the case of fixed μ^2. Taking concretely case (59) above, we expand about the fixed point and write

$$\sigma_h^2\xi_i^2 = \mu^2 - \kappa_i + x_i, \qquad (\Delta\theta)^2 = 0 + y. \tag{63}$$

Expanding Eqs. (53)–(55) in x_i and y and absorbing σ_h^2 in the time parameter (denoting the derivative with respect to $\tilde{t} = \sigma_h^2 t$ with a prime) then yields the linearized equations

$$x_1' = -2(\mu^2 - \kappa_1)\left[\frac{x_1}{\kappa_1^2} + \left(\frac{1}{\kappa_1} - \frac{1}{\kappa_2}\right)y\right], \tag{64}$$

$$x_2' = -2(\mu^2 - \kappa_2)\left[\frac{x_2}{\kappa_2^2} + \left(\frac{1}{\kappa_2} - \frac{1}{\kappa_1}\right)y\right], \tag{65}$$

$$y' = -2\frac{2\mu^2 - \kappa_1 - \kappa_2}{\kappa_1\kappa_2}y. \tag{66}$$

We hence find exponential convergence, controlled by the relaxation rates

$$\gamma_i = \frac{\mu^2 - \kappa_i}{\kappa_i^2}, \qquad \gamma_{\Delta\theta} = \frac{\kappa_1}{\kappa_2}\gamma_1 + \frac{\kappa_2}{\kappa_1}\gamma_2. \tag{67}$$

The angle $\Delta\theta(\tilde{t})$ relaxes with $\gamma_{\Delta\theta}$, whereas the singular values $\xi_i(\tilde{t})$ decay with $\min(\gamma_i, \gamma_{\Delta\theta})$. Interestingly, the relaxation rates are set by the difference between the RBM mass parameter μ^2 and the eigenvalues κ_i in the spectrum. Irrespective of the actual values of μ^2 and the eigenvalues κ_i, the mode corresponding to the higher eigenvalue relaxes the slowest. We hence conclude the following:

(1) infrared modes, i.e., those corresponding to the smallest eigenvalues will converge faster, this can indeed be observed in Fig. 2 (top row);
(2) increasing the value of μ^2 will lead to more rapid convergence for all modes. This will be explored below in more realistic cases.

3. $N_v = 2$, $N_h = 1$

The case of one hidden node serves to demonstrate what happens when $N_h < N_v$. It is particularly simple as V is replaced by $v = 1$ and we only need to consider one angle and one singular value, determined by the following equations:

$$\dot{\xi}_1 = \sigma_h^2\left(\frac{1}{\kappa_1}\cos^2\Delta\theta + \frac{1}{\kappa_2}\sin^2\Delta\theta - \frac{1}{\mu^2 - \sigma_h^2\xi_1^2}\right)\xi_1 \tag{68}$$

and

$$\dot{\Delta\theta} = \sigma_h^2\rho, \tag{69}$$

$$\dot{\mu}^2 = \frac{1}{2}\left(\frac{1}{\mu^2 - \sigma_h^2\xi_1^2} + \frac{1}{\mu^2} - \frac{1}{\kappa_1} - \frac{1}{\kappa_2}\right), \tag{70}$$

where

$$\rho = \tilde{\rho}\cos\Delta\theta\sin\Delta\theta, \qquad \tilde{\rho} = \left(\frac{1}{\kappa_2} - \frac{1}{\kappa_1}\right). \tag{71}$$

The equation for the angle is now decoupled and can be solved, as

$$\tan\left[\Delta\theta(t)\right] = \tan\left[\Delta\theta(0)\right]e^{\sigma_h^2\tilde{\rho}t}, \tag{72}$$

such that

$$\kappa_2 > \kappa_1 \Leftrightarrow \tilde{\rho} < 0 \Leftrightarrow \lim_{t\to\infty}\Delta\theta(t) = 0, \tag{73}$$

$$\kappa_2 < \kappa_1 \Leftrightarrow \tilde{\rho} > 0 \Leftrightarrow \lim_{t\to\infty}\Delta\theta(t) = \frac{\pi}{2}. \tag{74}$$

Using this in Eq. (68) confirms that the smallest eigenvalue of K_ϕ is reproduced (for constant μ^2). If μ^2 is learnt as well, then Eq. (70) ensures it becomes equal to the largest of the two eigenvalues.

To summarize, we note the following observations: if either the number of hidden nodes or the mass parameter μ^2 is chosen too small, the infrared part of the spectrum (lowest eigenvalue) is reproduced, while the ultraviolet part (highest eigenvalue) evolves to μ^2; making μ^2 a learnable parameter yields one more degree of freedom to correctly reproduce the next eigenvalue; infrared modes are learnt quicker than ultraviolet modes. These observations for the simple case considered here remain relevant for more interesting systems, as we will demonstrate now.

V. LEARNING GAUSSIAN DISTRIBUTIONS

We continue with the case for which the target distribution is known and Gaussian, namely free scalar fields discretized on a lattice in one and two dimensions. The continuum action reads, in n Euclidean dimensions,

$$S(\phi) = \int d^nx\,\frac{1}{2}(\partial_\mu\phi\partial_\mu\phi + m^2\phi^2). \tag{75}$$

The simplest lattice-discretized equivalent is, on a one-dimensional lattice with N_v nodes and with periodic boundary conditions,

$$S(\phi) = \frac{1}{2}\sum_{i,j=1}^{N_v}\phi_i K_{ij}^{\phi}\phi_j, \tag{76}$$

where

$$K_{ij}^{\phi} = (2 + m^2)\delta_{ij} - \delta_{i,j+1} - \delta_{i,j-1}. \tag{77}$$

We use "lattice units," $a = 1$, throughout. The spectrum of the target kernel K^ϕ is easy to compute analytically and reads

$$\kappa_k = m^2 + p_{\mathrm{lat},k}^2 = m^2 + 2 - 2\cos\left(\frac{2\pi k}{N_v}\right), \tag{78}$$

with $-N_v/2 < k \leq N_v/2$. Each eigenvalue is doubly degenerate, except the minimum $(k = 0, \kappa_{\min} = m^2)$ and the maximal $(k = N_v/2, \kappa_{\max} = m^2 + 4)$ ones. Referring back to Sec. II, the exact spectrum can only be learnt when $N_h = N_v$ and when the RBM mass parameter

$$\mu^2 > \kappa_{\max} = m^2 + 4. \tag{79}$$

Since the target theory is known, we can train the model directly from the correlation matrix of the target theory without the need for pregenerated training data. Then each term of the gradient is estimated by persistent contrastive divergence (PCD) to train the RBM, see Appendix A for details. The scalar field mass parameter is chosen as $m^2 = 4$ and the variance on the hidden layer equals $\sigma_h^2 = 1$ throughout.

A. Initialization with an exact solution

We start with the case of a constant RBM mass parameter $\mu^2 = 9 > \kappa_{\max} = 8$, with $N_v = N_h = 10$. To test the numerical code, we may initialize the weight matrix W according to one of the exact solutions found in Sec. II: the Cholesky (lower triangular) solution and the symmetric solution. The results are shown in Figs. 3 and 4. Here and throughout we denote the exact eigenvalues of the target distribution with κ_α $(\alpha = 1, \ldots, N_v)$ and the eigenvalues of the model kernel K with $\lambda_\alpha = \mu^2 - \sigma_h^2\xi_\alpha^2$. We will refer to these as the RBM eigenvalues. The latter depends on the training stage, indicated by epochs, see Appendix A. As can be seen in Figs. 3 and 4 (left), the RBM eigenvalues are correctly initialized for both choices and fluctuate around the correct values during training.

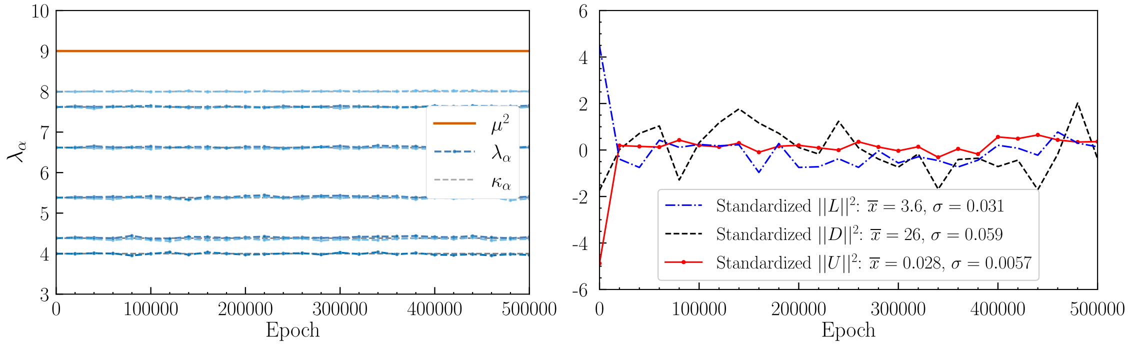


FIG. 3. Cholesky initialization. Left: evolution of RBM eigenvalues λ_α during training. Note that adjacent eigenvalues are colored alternatively. Exact eigenvalues κ_α are shown with horizontal dashed lines and the RBM mass parameter μ^2 with the horizontal full line. After the Cholesky initialization, the RBM eigenvalues fluctuate around the correct values. Right: the L_2 norm of each part of the coupling matrix, diagonal (D), upper (U) triangular, and lower (L) triangular. Values are standardized, with $\bar{x}$ (σ) the mean value (standard deviation) along the training interval. Each part fluctuates around its average value.

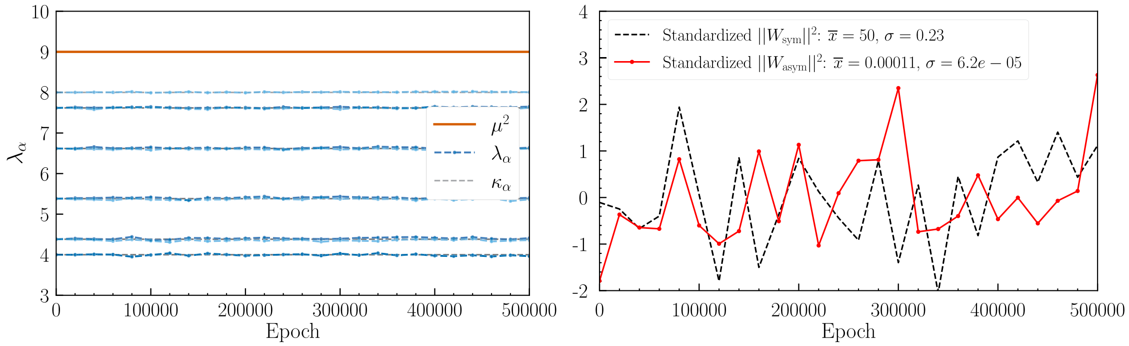


FIG. 4. Left: as in Fig. 3, for the symmetric initialization. Right: standardized L_2 norm of symmetric and asymmetric parts of the coupling matrix. The latter remains small during updates.

To indicate the size of the fluctuations, we do the following. In the Cholesky case, we consider separately the L_2 norm of the lower triangular elements, of the upper triangular elements (which are initialized at zero) and of the elements on the diagonal. We then standard normalize these to compare the amplitudes of the fluctuations, see Fig. 3 (right). We observe that the sum of each part fluctuates around the average value during training, whose size is set by the initial value, demonstrating the stability of the PCD updates.

For the symmetric initialization, we show the L_2 norms of the symmetric and asymmetric parts, $W_{\text{sym}} = (W + W^T)/2$, $W_{\text{asym}} = (W - W^T)/2$. Since the initial coupling matrix W is symmetric, we expect the norm of the asymmetric part to remain significantly smaller during training. This can indeed be seen in Fig. 4 (right), where we show the evolution after standard normalization. The norm of the symmetric part of the coupling matrix is six orders of magnitude larger than that of the asymmetric part. As with the Cholesky initialization, we observe that the overall structure of the coupling matrix is approximately preserved. Note there is no reason for it to be *exactly* preserved, as this is neither imposed nor required.

B. Initialization with a random coupling matrix

In practical applications, the coupling matrix W is not initialized at an exact solution, but with random entries, drawn e.g. from a Gaussian distribution. In Fig. 5 we show the results obtained with elements of W sampled from a normal distribution $\mathcal{N}(0, 0.1)$. Other parameters are as above within particular $N_h = N_v$ and $\mu^2 > \kappa_{\text{max}}$; hence there are no obstructions to learning the target distribution exactly. This can indeed be seen in Fig. 5, where both the eigenvalues (left) and the action density (right) are seen to match. For the latter, configurations are generated using the trained RBM; the same number of Monte Carlo (Metropolis) generated configurations are shown, using binning to remove autocorrelations. The analytical result follows from the equipartition. It is noted that possible instabilities, due to λ_α turning negative either initially or during the learning stage, are not encountered with this initialization. If they are encountered, then they can be avoided by tuning the width of the initial coupling matrix and learning rate.

Since the elements of W are initially relatively small, the corresponding singular values ξ_α are small as well and the RBM eigenvalues $\lambda_\alpha = \mu^2 - \sigma_h^2 \xi_\alpha^2$ are close to μ^2 initially.

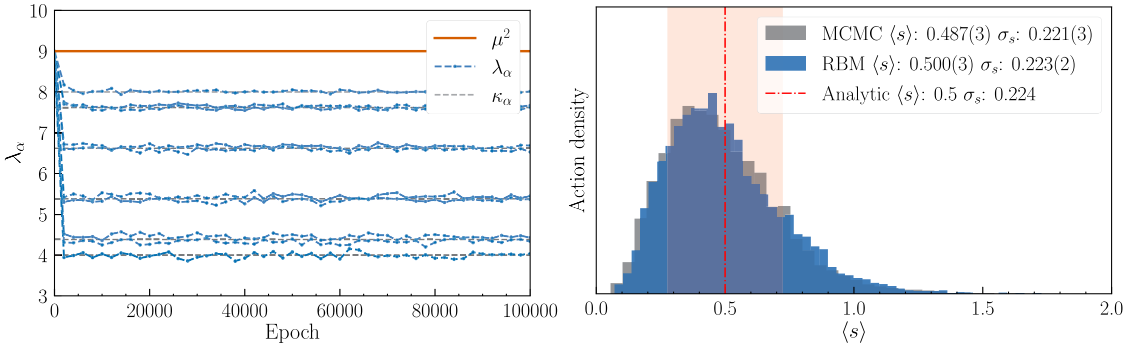


FIG. 5. Left: evolution of RBM eigenvalues λ_α during training, starting from a random coupling matrix W. Presentation as in Fig. 3 (left). Right: histogram density of action density from Monte Carlo generated and RBM generated samples.

They quickly evolve to the target values κ_α. The order in which the modes are learnt (or thermalized) can be understood easily. Referring back to Sec. IV, we consider Eq. (43) for the singular values and Eq. (36) for the driving term. Assuming we are on the correct eigenbasis, the latter reduces to

$$\Lambda = \Lambda_d = D_\phi^{-1} - D_K^{-1} = \mathrm{diag}(1/\kappa_\alpha - 1/\lambda_\alpha), \tag{80}$$

where $\lambda_\alpha = \mu^2 - \sigma_h^2\xi_\alpha^2$. Equation (43) then becomes [13]

$$\frac{d}{dt}\xi_\alpha^2 = 2\sigma_h^2\left(\frac{1}{\kappa_\alpha} - \frac{1}{\mu^2 - \sigma_h^2\xi_\alpha^2}\right)\xi_\alpha^2. \tag{81}$$

Note that this equation was encountered before (in a general basis) for $N_v = N_h = 2$, see Eqs. (53), (54). During the initial stages, the term within the brackets is negative and largest for the smallest eigenvalues. Hence the corresponding singular values evolve quickest. At late times, one may linearize around the fixed point. In Sec. IV we demonstrated for $N_v = 2$ nodes that the convergence in the linearized regime is exponentially fast and that the rate of convergence is set by $\gamma_\alpha = (\mu^2 - \kappa_\alpha)/\kappa_\alpha^2$. Hence the most infrared modes equilibrate fastest and the ultraviolet modes slowest. These aspects are demonstrated in Fig. 6, where we have shown the evolution of both the singular values (left) and the eigenvalues (right) during the initial stages of the training (the largest singular values correspond to the smallest eigenvalues). We note the similarity with the case of $N_v = 2$ modes in Sec. IV, see in particular Fig. 2 (top row).

So far we have kept the RBM mass parameter μ^2 fixed. However, it can also be treated as a learnable parameter using Eq. (22). This is particularly useful if details of the target spectrum are not known. It provides then an additional degree of freedom. In Fig. 7, the initial RBM mass parameter is initialized below $\kappa_{\max}$. It subsequently increases to match the largest eigenvalue, see Fig. 7 (left). Since the system is overparametrized, one of the singular values remains at the initial value, see Fig. 7 (right). Note the different timescale for equilibration compared to the case with a constant μ^2, as it takes time for μ^2 to find the correct value.

Up to now, we considered a scalar field in one dimension only. The generalization to higher dimensions is interesting since the RBM does not know about the dimensionality *a priori*, with the N_v visible nodes only connecting to the hidden nodes. We consider here two dimensions, using an $N_x \times N_y$ lattice. The eigenvalues of the target kernel are

$$\begin{aligned}\kappa_{\mathbf{k}} &= m^2 + p_{\mathrm{lat},\mathbf{k}}^2, \\ &= m^2 + 4 - 2\cos\left(\frac{2\pi k_x}{N_x}\right) - 2\cos\left(\frac{2\pi k_y}{N_y}\right),\end{aligned} \tag{82}$$

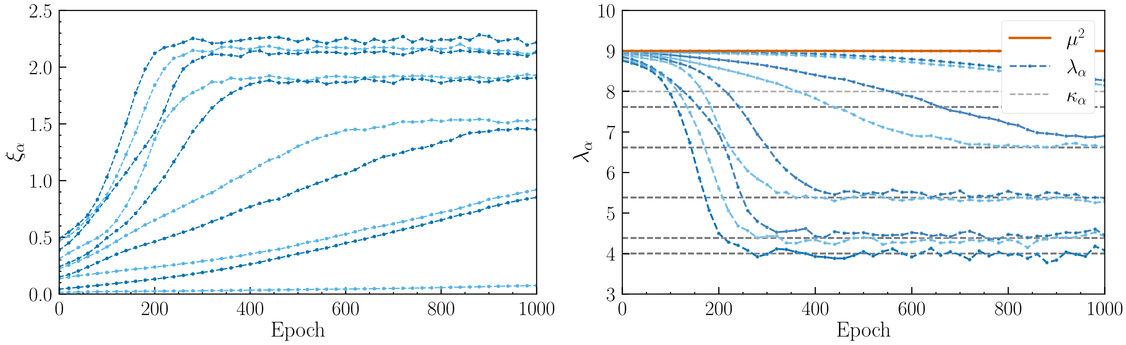


FIG. 6. Convergence of the singular values ξ_α (left) and the eigenvalues λ_α (right) for the system of Fig. 5. Infrared modes are learnt the quickest.

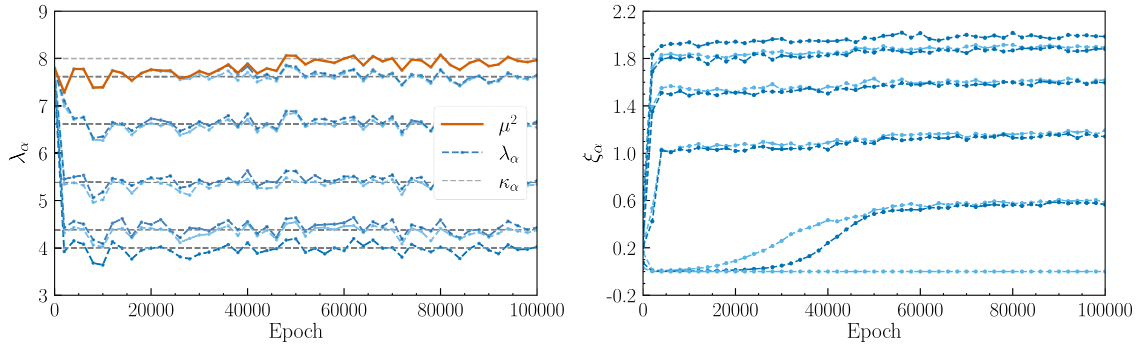


FIG. 7. Left: evolution of the RBM eigenvalues and mass parameter μ^2, with the latter initialized below $\kappa_{\max}$. Right: evolution of the singular values. Since the system is overparametrized, one of the singular values remains at the initial value.

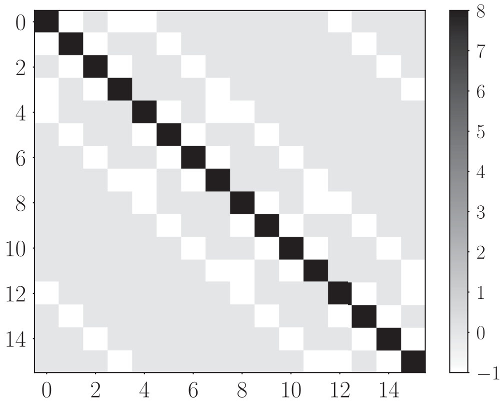


FIG. 8. Flattened kernel for a two-dimensional scalar field theory on a lattice with $N_x \times N_y = 4 \times 4$ sites and $m^2 = 4$. Each site has four nearest neighbors.

with $-N_{x,y}/2 < k_{x,y} \le N_{x,y}/2$. In this case, there is a larger degeneracy of eigenvalues. The RBM has $N_v = N_x \times N_y$ visible nodes. The dimensionality has to be learnt and encoded in the coupling matrix W. The (target) kernel and two-point functions are $(N_x \times N_y) \times (N_x \times N_y)$-dimensional tensors. This two-dimensional structure can be flattened in a one-dimensional representation, where the kinetic term is decomposed into a tensor product of two one-dimensional Laplacian operators,

$$\begin{aligned} K^{\phi,2d} &= m^2 \mathbb{1} + \Delta^{2d}, \\ &= m^2 \mathbb{1}_{N_x\times N_x} \otimes \mathbb{1}_{N_y\times N_y} + \Delta^{1d} \otimes \mathbb{1}_{N_y\times N_y} \\ &\quad + \mathbb{1}_{N_x\times N_x} \otimes \Delta^{1d}, \end{aligned} \tag{83}$$

where in the last expression $\otimes$ is the Kronecker product and the sizes of the identity matrices are given explicitly.

Figure 8 shows an example of a flattened scalar field kernel for the two-dimensional case with $N_x = N_y = 4$. Importantly, the spectrum of the flattened kernel and the original kernel are identical, since the boundary conditions are encoded correctly. The tensor product decomposition (83) allows one to see this explicitly.

In Fig. 9 (left), we show the evolution of the RBM eigenvalues. The RBM mass parameter is $\mu^2 = 16 > \kappa_{\max} = 12$. There should be four degenerate eigenvalues at 6 and 10, and six degenerate ones at 8. Yet it appears the eigenvalues only lie within a band close to the expected value. This is due to the fact that to obtain these results we have used a fixed learning rate (time step), which prevents the system from reaching high precision. This can be remedied by introducing an epoch dependent learning rate. This is explored in Appendix A. We multiply the learning rate by a factor close to one, $r = 0.99$, after a given number of epochs, $N_{\text{epoch}}^{\text{rate}} = 128$. The virtue of having a diminished learning rate in the later stages is that it allows

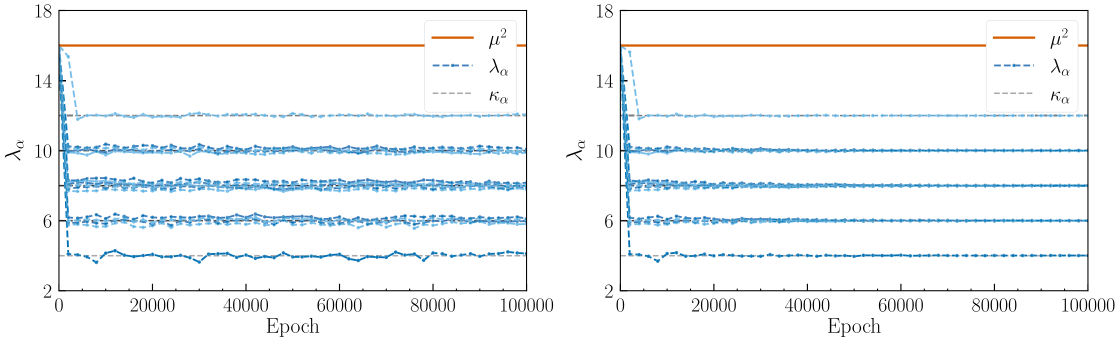


FIG. 9. Evolution of the eigenvalues in the two-dimensional case during training, with constant $\mu^2 = 16$ and $N_v = N_h = 16$, using a fixed (left) and a diminishing (right) learning rate.

the model to be finely trained, with less statistical fluctuations. The result is shown in Fig. 9 (right), where we indeed observe precise agreement with the target spectrum.

C. Ultraviolet regularization by the RBM mass parameter

Up to now, we have considered the ideal "architecture," namely $N_h = N_v$ and $\mu^2 > \kappa_{\max}$, for which Gaussian distributions can be learnt exactly, as we have demonstrated. In practice, one often chooses $N_h < N_v$ and the maximum eigenvalue may not be known. Here we determine what this implies.

We start with the case where $N_h = N_v$, but with μ^2 fixed and less than $\kappa_{\max}$. We refer to Eq. (81) for the evolution of the singular values in the eigenbasis. Take $\mu^2 < \kappa_\alpha$. In this case, the term inside the brackets is always negative and the only solution is $\xi_\alpha = 0$. The corresponding eigenvalue is then $\lambda_\alpha = \mu^2$. When $\mu^2 > \kappa_\alpha$, the solution is given by the fixed point, $\sigma_h^2 \xi_\alpha^2 = \mu^2 - \kappa_\alpha$, and $\lambda_\alpha = \mu^2 - \sigma_h^2 \xi_\alpha^2$. We hence conclude that the infrared part of the spectrum, with $\kappa_\alpha < \mu^2$, can be learnt, whereas the ultraviolet part, with $\kappa_\alpha > \mu^2$, cannot be learnt. Instead, the RBM eigenvalues take the value of the cutoff, μ^2 [14].

This is demonstrated in Fig. 10 for a one-dimensional scalar field theory with $N_v = N_h = 10$ nodes. As the condition for exact training is violated, the RBM model can no longer faithfully reproduce the target data and distribution. The impact of this depends on the importance of the ultraviolet modes, as we will see below for the MNIST dataset.

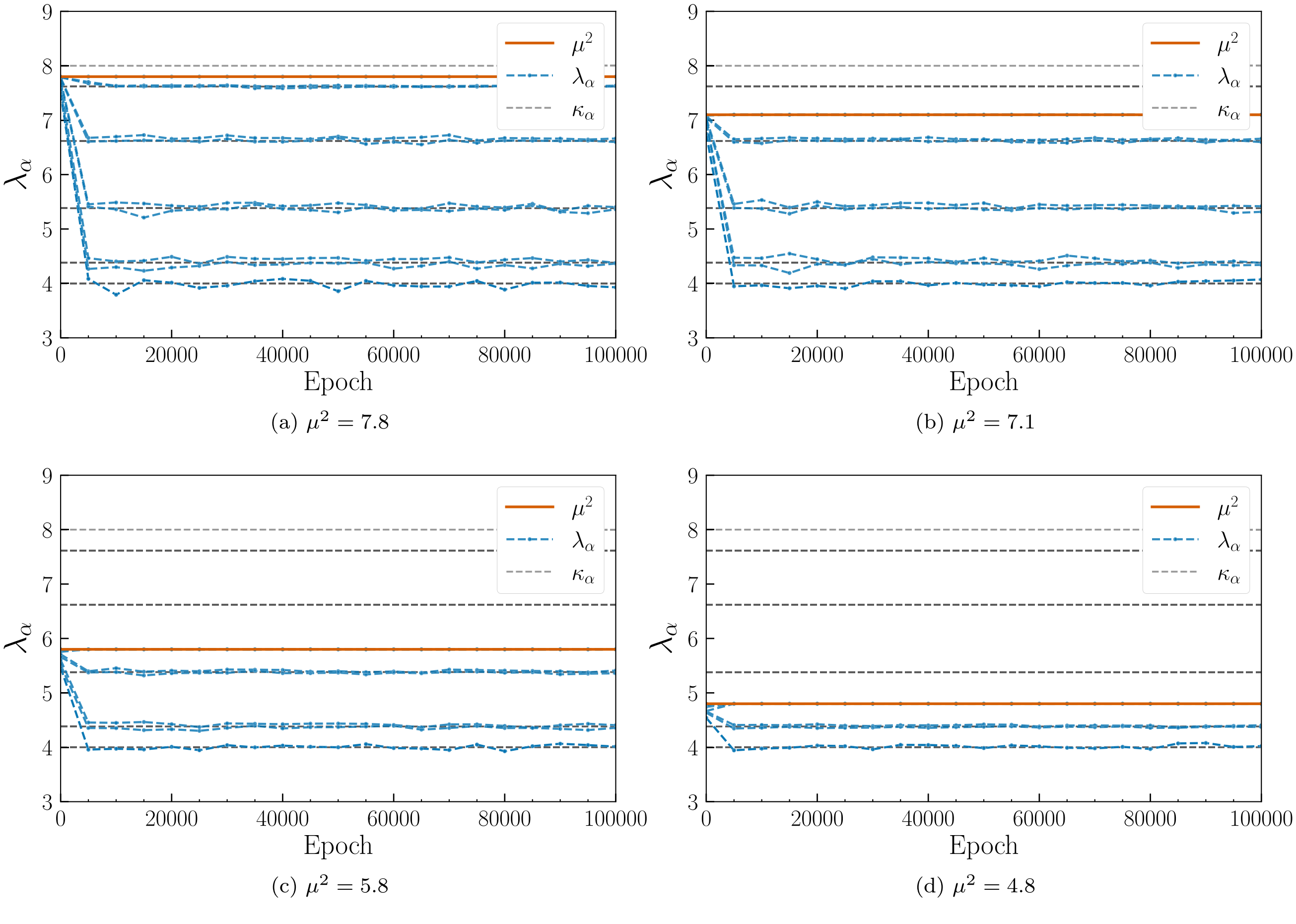
D. Ultraviolet regularization by the number of hidden nodes

Next, we consider the case with $N_h < N_v$. This is straightforward, as there are only N_h singular values, leading to the RBM eigenvalues

$$\lambda_\alpha = \begin{cases} \mu^2 - \sigma_h^2 \xi_\alpha^2 & 1 \le \alpha \le N_h \\ \mu^2 & N_h < \alpha \le N_v \end{cases}, \tag{84}$$

see e.g. Eq. (33). Again we note that the infrared part of the spectrum can be reproduced, whereas the ultraviolet part is fixed at μ^2, irrespective of the actual value of the target eigenvalue.

This is shown in Fig. 11 for the one-dimensional case with $N_v = 10$ and $N_h = 9, 8, 7, 6$. Note that all eigenvalues,



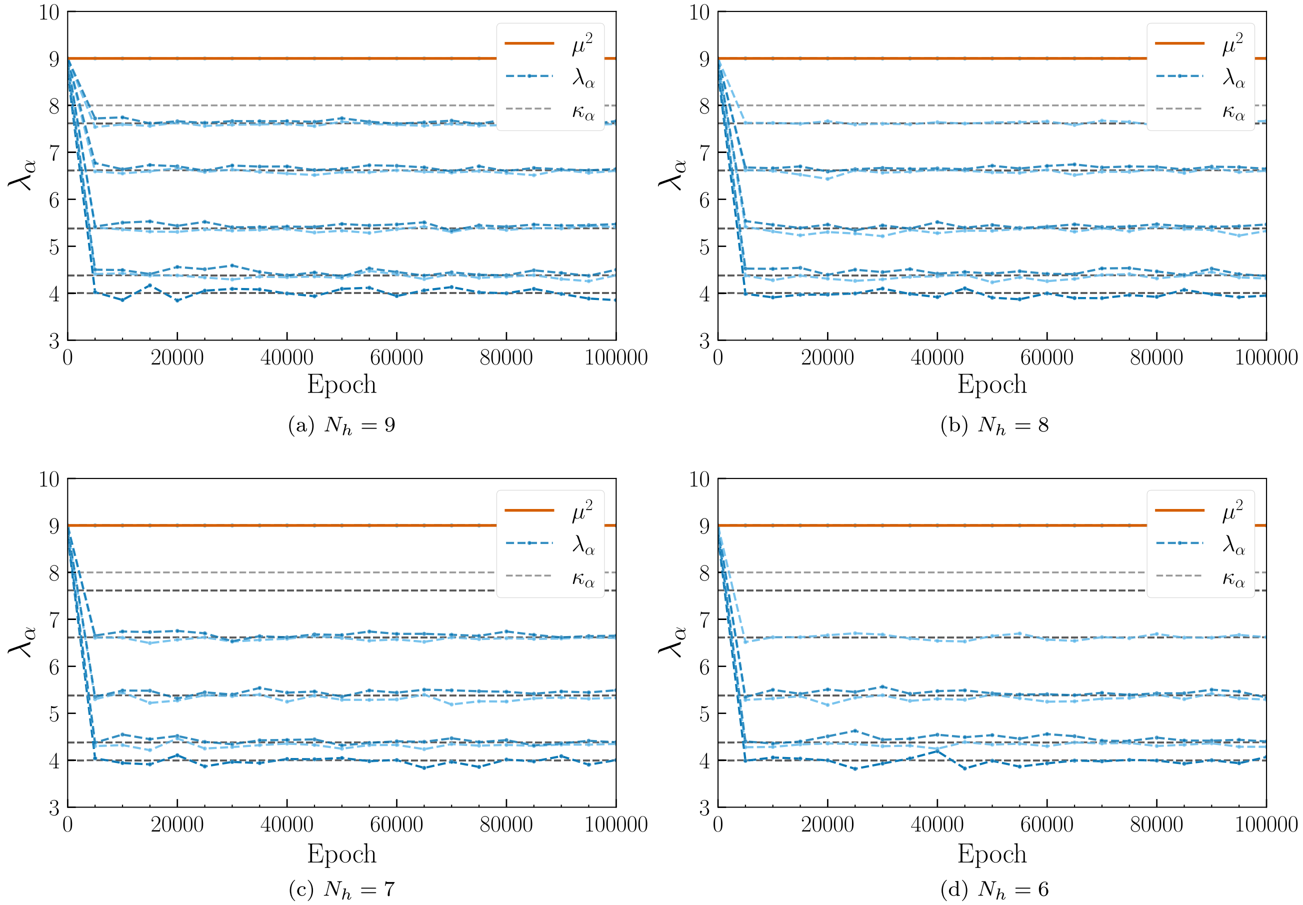
(a) $\mu^2 = 7.8$

(b) $\mu^2 = 7.1$

(c) $\mu^2 = 5.8$

(d) $\mu^2 = 4.8$

FIG. 10. Regularization by RBM mass parameter μ^2: evolution of the eigenvalues in the one-dimensional scalar field theory. Only the infrared part of the spectrum is reproduced.



(a) $N_h = 9$

(b) $N_h = 8$

(c) $N_h = 7$

(d) $N_h = 6$

FIG. 11. As above, but with regularization by the number of hidden nodes N_h.

except the minimal and maximal ones, are doubly degenerate. Hence in the case of $N_h = 8$ and 6, one of the degenerate eigenvalues remains and one is removed.

Finally, in Fig. 12 we give two examples in the two-dimensional scalar theory, using $\mu^2 = 9 < \kappa_{\max}$ on the left and $N_h = 8 < N_v = 16$ on the right.

VI. MNIST DATASET

It is important to ask whether the considerations above are also relevant for realistic datasets commonly used in ML. We consider the MNIST dataset [15], consisting of 60,000 28×28 images of digits. Hence $N_v = 784$, substantially larger than what we have considered so far.

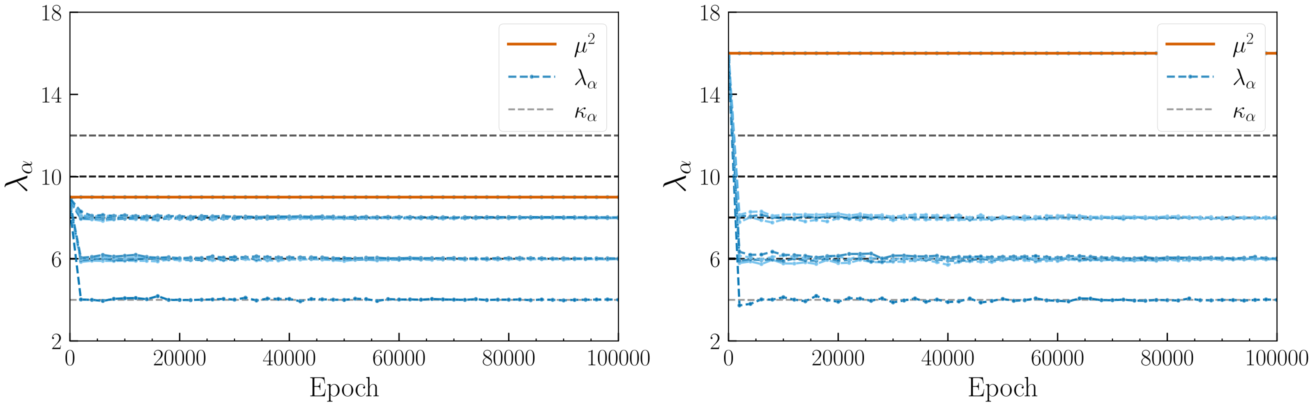


FIG. 12. Regularization by the RBM mass parameter $\mu^2 = 9$ (left) and by the number of hidden modes $N_h = 8$ (right) in the two-dimensional scalar field theory with $N_x \times N_y = 4 \times 4$.

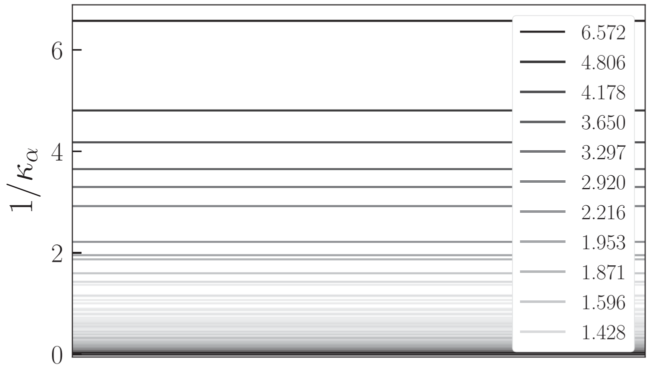


FIG. 13. Eigenvalues of the correlation matrix $\langle \phi_i \phi_j \rangle$ of the MNIST dataset. Note that many eigenvalues are close to zero. The values of the ten largest eigenvalues are indicated.

Unlike in the case of scalar fields, the probability distribution function is not known. However, we may still obtain the correlation matrix $\langle \phi_i \phi_j \rangle$ by summing over the 60,000 realizations. The MNIST (target) kernel is then given by its inverse,

$$K^{\mathrm{MNIST}} = \langle \phi_i \phi_j \rangle^{-1}_{\mathrm{MNIST}}. \tag{85}$$

The eigenvalues of the correlation matrix are the inverse of the eigenvalues of the kernels discussed so far and we hence denote them as $1/\kappa_\alpha$. The 784 eigenvalues are shown in Fig. 13. Many eigenvalues are close to zero. In the language of the previous sections, these correspond to the ultraviolet part of the spectrum of the quadratic kernel and hence the MNIST dataset can be said to be ultraviolet divergent. The values of the ten largest eigenvalues of the correlation matrix are listed explicitly on the right. These correspond to the infrared part of the spectrum of the quadratic kernel. Since these are finite, the MNIST dataset is infrared safe. This terminology reflects the ordering of the eigenvalues κ_α encoding the quadratic correlations in the MNIST data, from small (infrared) to large (ultraviolet). As in the two-dimensional scalar case, the eigenvalues do not depend on the flattening of the indices.

We will now train the scalar field RBM on the MNIST dataset, starting with $N_h = N_v$ and a fixed RBM mass parameter $\mu^2 = 100$. The result is shown in Fig. 14 (left). As before, the horizontal dashed black lines are the target eigenvalues, obtained from the MNIST correlation matrix. The blue lines are the RBM eigenvalues. The initial values are close to μ^2 and hence they become smaller during the learning stage. As above the infrared part of the spectrum is learnt quickest. This is further illustrated in Fig. 14 (right), where the evolution during the final 60,000 epochs are shown (out of one million). The smallest eigenvalues agree with the target values but the larger ones have essentially stopped learning before reaching the correct value, due to the reduced learning rate. We note that the RBM mass parameter $\mu^2 = 100$ regulates the number of modes here. In fact, there are 289 modes below the cutoff set by μ^2. Hence the number of hidden modes, $N_h = 784$, can be reduced without a loss of quality. We come back to this below.

Without knowledge of the target spectrum, the (constant) value for μ^2 may be chosen to be on the small side; as is obvious in Fig. 14 (left), there are many eigenvalues above $\mu^2 = 100$. This can be remedied by promoting μ^2 to a learnable parameter. This is demonstrated in Fig. 15, where μ^2 increases such that the target spectrum can be better learnt. The learning dynamics employs a diminishing learning rate, see Appendix A, which slows down the increase of μ^2 but also hinders the learning of the spectrum beyond the infrared modes. As stated, larger eigenvalues

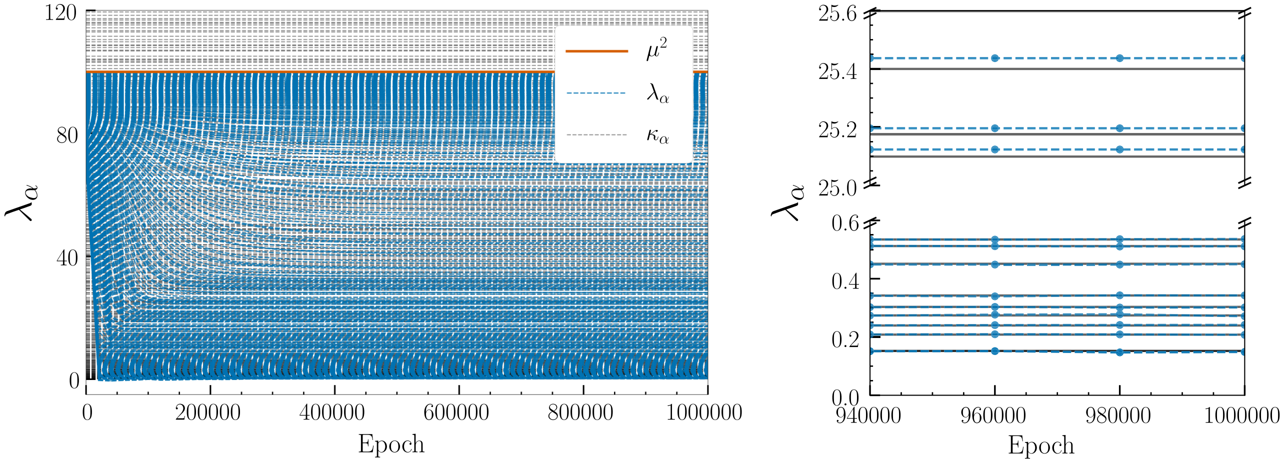


FIG. 14. Left: evolution of the eigenvalues for the MNIST dataset with fixed RBM mass parameter $\mu^2 = 100$, and $N_v = N_h = 784$. Right: evolution during the last few training epochs. The lowest eigenvalues have already matched their target values but the higher modes are still being trained, albeit at a very small learning rate.

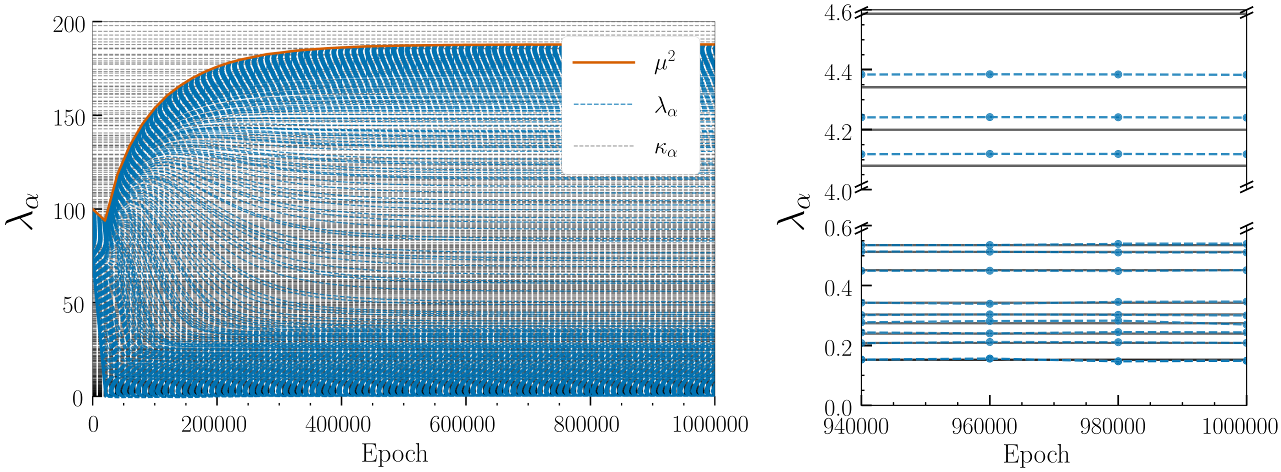


FIG. 15. As above, but with a learnable RBM mass parameter μ^2.

give smaller contributions to the update equations, leading a slower learning.

As in the scalar field case, we can regularize the MNIST data by choosing $N_h < N_v$. In this case, the number of modes that can be learnt depends on the number of modes in the spectrum with an eigenvalue less than μ^2, and the number of hidden nodes. Figure 16 shows the quality of regenerated images after one passes forward and backward through the trained RBM. Using the fixed RBM mass parameter $\mu^2 = 100$ limits the maximal number of modes to be included to $N_{\text{modes}}^{\text{max}} = 289$, the number of modes with an eigenvalue less than μ^2. We observe that one needs at least around 64 hidden nodes to have an acceptable generation, which is considerably smaller than the maximal possible number. This illustrates that the ultraviolet part of the spectrum can be ignored.

To give a more quantitative measure of the quality of regeneration, we have computed the data-averaged KL divergence for the trained model,

$$\begin{aligned}\mathrm{KL}(p_{\text{data}}||p_{\text{model}}) &= -\frac{1}{N_{\text{conf}}}\sum_{d=1}^{N_{\text{conf}}}\log p_{\text{model}}(\phi^{(d)},\theta^*) + \text{cst} \\ &= \frac{1}{N_{\text{conf}}}\sum_{d=1}^{N_{\text{conf}}}\frac{1}{2}\phi^{(d)T}K\phi^{(d)} + \log Z_{\text{model}} + \text{cst}, \qquad (86)\end{aligned}$$

where the constant "cst" term is independent of the model distribution. The result is shown in Fig. 17. We indeed observe the KL divergence between the target distribution and the model distribution starts to increase as more modes are excluded. Adding modes beyond the cutoff imposed by the choice of μ^2 does not increase the quality, as expected. As concluded "by eye" above, around 64–100 hidden nodes are required for a reasonable quality of regeneration.

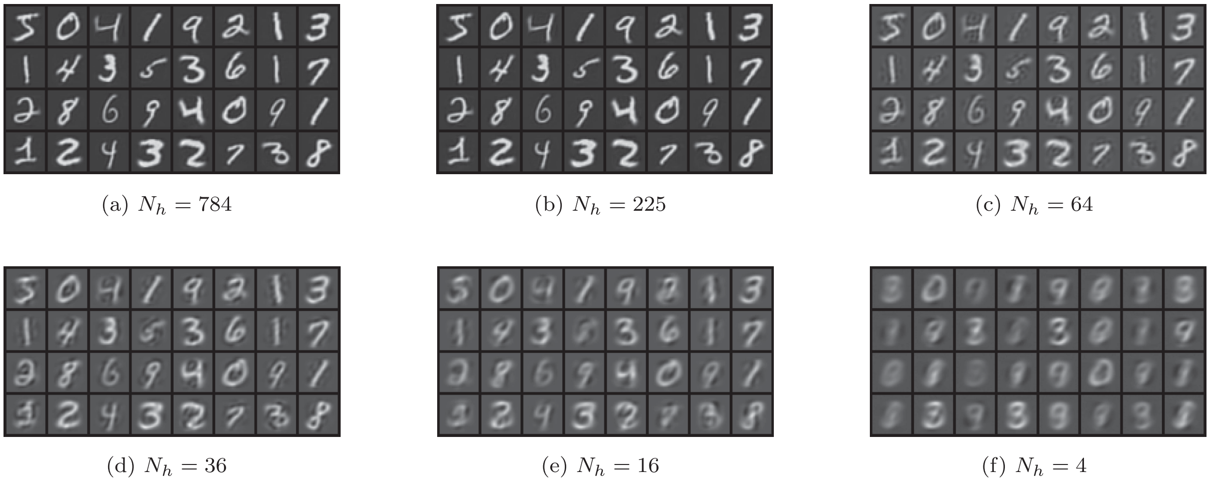

(a) $N_h = 784$
(b) $N_h = 225$
(c) $N_h = 64$
(d) $N_h = 36$
(e) $N_h = 16$
(f) $N_h = 4$

FIG. 16. Quality of regenerated images with different numbers of hidden nodes. As the number of hidden nodes decreases, the regeneration quality decreases.

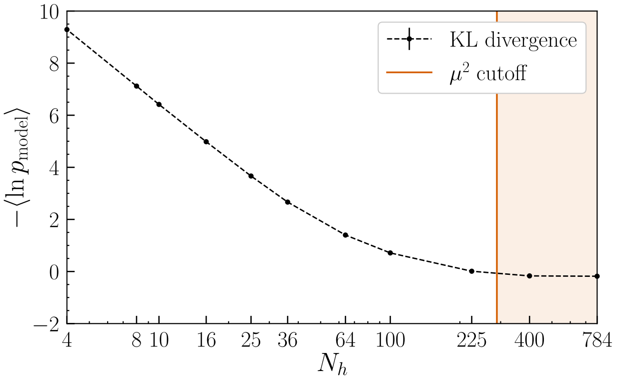


FIG. 17. Data-averaged KL divergence for the trained RBM, as a function of the number of hidden nodes N_h (on a logarithmic scale) at fixed $\mu^2 = 100$. For this value of μ^2, the maximal number of modes included is $N^{\rm max}_{\rm modes} = 289$ and hence increasing N_h above this value does not lead to additional improvement.

VII. INTERACTIONS

Strictly speaking, the Gaussian-Gaussian RBM can only be exact if the target distribution is Gaussian as well. To go beyond this, one needs to introduce interactions. There are (at least) two ways of doing so. Motivated by the notion of QFT-ML [8], one may add local interactions via potentials defined on nodes, see Eqs. (5), (6). A simple choice would be to add a $\lambda\phi^4$ term to each node on the visible layer since such systems are well understood and allow one to study e.g. spontaneous symmetry breaking in the context of the learning process. Of course, sampling the hidden layer requires a more costly sampling method than for the Gaussian case.

One may also change the nature of the hidden nodes from continuous to discrete, taking e.g. $h_a = \pm 1$. This leads to the Gaussian-Bernoulli RBM (see e.g. Ref. [13]), with the distribution and action

$$p(\phi, h) = \frac{1}{Z}\exp(-S(\phi, h)), \qquad (87)$$

$$S(\phi, h) = V_\phi(\phi) - \sum_{i,a}\phi_i W_{ia} h_a + \sum_a \eta_a h_a, \qquad (88)$$

where

$$Z = \int D\phi \prod_{a=1}^{N_h} \sum_{h_a=\pm 1} \exp(-S(\phi, h)). \qquad (89)$$

This gives the following induced distribution on the visible layer,

$$p(\phi) = \frac{1}{Z}\exp(-S(\phi)), \qquad (90)$$

with the effective action

$$S(\phi) = V_\phi(\phi) - \sum_a \ln\left(2\cosh(\psi_a)\right), \qquad (91)$$

where $\psi_a = \sum_i \phi_i W_{ia} - \eta_a$. A formal expansion in ψ_a then yields, up to a constant, the effective action on the visible layer,

$$\begin{aligned} S(\phi) &= V_\phi(\phi) - \sum_a \ln\left(1 + \sum_{n=1}^{\infty}\frac{\psi_a^{2n}}{(2n)!}\right), \\ &= V_\phi(\phi) + \sum_a \sum_{n=1}^{\infty}(-1)^n \frac{c_n}{(2n)!}\psi_a^{2n}, \end{aligned} \qquad (92)$$

with easily determined coefficients c_n. This is a highly nonlocal action.

To make the connection with the previous sections, it is straightforward to see that the quadratic ($n = 1$, $c_1 = 1$) term yields the same structure as above,

$$-\sum_a \frac{1}{2}\psi_a^2 = -\frac{1}{2}\phi^T W W^T \phi - \frac{1}{2}\eta^T \eta + \phi^T W \eta, \qquad (93)$$

which, when combined with the RBM mass parameter μ^2 on the visible layer, gives the same kernel, $K = \mu^2 \mathbb{1} - WW^T$, and source, $J = W\eta$.

Quartic interactions are generated at the next order. Taking for simplicity $\eta_a = 0$, such that only even terms in ϕ_i are present, one finds the $n = 2$ term ($c_2 = 2$),

$$\sum_a \frac{1}{12}\psi_a^4 = \frac{1}{12}\sum_{ijkl}\lambda_{ijkl}\phi_i\phi_j\phi_k\phi_l, \qquad (94)$$

with the coupling

$$\lambda_{ijkl} = \sum_a W_{ia}W_{ja}W_{ka}W_{la}. \qquad (95)$$

This is indeed a quartic term but with an *a priori* highly nonlocal coupling, generated by the all-to-all coupling to the hidden layer. From a QFT perspective, it would be of interest to study such a theory, which we postpone to the future.

VIII. CONCLUSION

In this paper, we have studied scalar field restricted Boltzmann machines from the perspective of lattice field theory. The Gaussian-Gaussian case can be understood completely. We have demonstrated, using analytical work and numerical experiments, that the scalar field RBM is an ultraviolet regulator, regulating the ultraviolet part of the spectrum of the quadratic operator of the target theory. This is also the case when the target probability distribution is not known, such as in the MNIST case, but where the spectrum can be extracted from the data-averaged correlation matrix. The cutoff is determined by the choice of the

RBM mass parameter or the number of hidden nodes. This provides a clear answer to generally difficult questions on the choice of ML "architecture," namely what are the consequences of choosing a particular setup. At least in this simple case the answer is straightforward and concerns the (in)sensitivity of the generative power of the RBM to the ultraviolet modes compared to the infrared modes.

We have also shown that infrared modes are learnt the quickest. This is of interest for models which suffer from critical slowing down, for which infrared modes are usually hard to handle. Indeed, many ML (inspired) generative approaches have surprisingly short autocorrelation times, which is worth exploring further.

As an outlook, we note that in the final section, we have indicated two ways to go beyond the Gaussian-Gaussian case. The QFT-ML approach, in which local potentials are added to nodes on e.g. the visible layer, is convenient for LFT practitioners since the resulting models are well understood. Replacing the continuous hidden degrees of freedom with binary ones (Gaussian-Bernoulli RBM) yields models of a very different character, involving highly nonlocal interaction terms to all orders. It would be of interest to understand these constructions further using QFT methods.

The data generated for this manuscript and simulation code are available from Ref. [16].

ACKNOWLEDGMENTS

G. A. and B. L. are supported by STFC Consolidated Grant No. ST/T000813/1. C. P. is supported by the UKRI AIMLAC CDT EP/S023992/1. We thank ECT* and the ExtreMe Matter Institute EMMI at GSI, Darmstadt, for support and the participants of the ECT*/EMMI workshop *Machine learning for lattice field theory and beyond* in June 2023 for discussion during the preparation of this paper.

APPENDIX A: DETAILS OF THE ALGORITHM

The training equations for the RBM parameters θ read, schematically,

$$\theta_{n+1} = \theta_n + \eta_n \frac{\partial \mathcal{L}}{\partial \theta},$$
$$\frac{\partial \mathcal{L}}{\partial \theta} = -\frac{1}{2}\left\langle \phi^T \frac{\partial K}{\partial \theta} \phi \right\rangle_{\text{target}} + \frac{1}{2}\left\langle \phi^T \frac{\partial K}{\partial \theta} \phi \right\rangle_{\text{model}}, \tag{A1}$$

where η_n is the learning rate. The first term on the rhs can be easily computed from the given data or target theory. The second term needs to be sampled from the model distribution, which is nontrivial. In most cases, this term is approximated by generating a Markov chain and truncating it after k steps, where k is empirically chosen. This is known as Contrastive Divergence (CD) [17]. For standard CD updates, the Markov chain is initialized from the input data and then the successive chains are sampled by Gibbs sampling. A more efficient update algorithm is Persistent Contrastive Divergence (PCD) [18] and is used in this paper. PCD initializes the Markov chain from the last state of the most recent update. Since this last state of the previous chain is already closer to the representative of the model distribution, the new Markov chain is initialized with a nearly thermalized state and only requires a small number of updates.

Alongside PCD, the gradient for each epoch is estimated by averaging over a minibatch. In the case of MNIST data, the training was done by using an effective correlation matrix obtained from the given dataset. Then 512 parallel PCD Markov chains were generated to form a minibatch. For the scalar field theory case, the training was done by directly using the analytical form of the kernel matrix of the target distribution without predefined training data. Then for each training epoch, 128 parallel PCD Markov chains were generated to be averaged and used to estimate the gradient.

The learning rate can be set to change during the training. For instance, one may multiply the learning rate by a factor of r after every $N_{\text{epoch}}^{\text{rate}}$ epochs (e.g. $r = 0.99$, $N_{\text{epoch}}^{\text{rate}} = 128$),

$$\eta_n = r\eta_{n-1} \quad \text{if} \quad \text{mod}\ (n, N_{\text{epoch}}^{\text{rate}}) = 0. \tag{A2}$$

Hence the learning rate becomes smaller as more epochs have passed. The virtue of having a small learning rate during the later part of the training is that it allows the model to be finely trained and that it reduces statistical fluctuations.

The effect of learning rate decay is shown in Fig. 18. Two models are trained with the same hyperparameters and initialization except for the learning rate decay parameters r and $N_{\text{epoch}}^{\text{rate}}$. The first model shown in Fig. 18 (left) is trained without learning rate decay. Fluctuation of the eigenvalues due to statistical noise remains. In contrast, the second model, shown in Fig. 18 (right), uses learning rate decay with $r = 0.99$, $N_{\text{epoch}}^{\text{rate}} = 128$. Statistical fluctuations die off in the end, leading to a precise result.

However, the values of r and $N_{\text{epoch}}^{\text{rate}}$ should be chosen in a delicate manner. If the decay rate r is too large, then the learning rate decreases too fast and the model freezes before it reaches the target destination. For example, in Fig. 19, the training flow of the scalar field RBM with the trainable mass parameter and $r = 0.99$, $N_{\text{epoch}}^{\text{rate}} = 128$ is shown (compare with Fig. 7). The model does not suffer when it is learning infrared modes, which are learnt quickest, but it fails to learn the highest mode of the target kernel. The model parameter freezes out before it reaches the target. One can decide the learning rate decay parameters by observing the regenerated samples and measuring the goodness of those. Since the ultraviolet modes are less relevant compared to the infrared ones, one can accept a

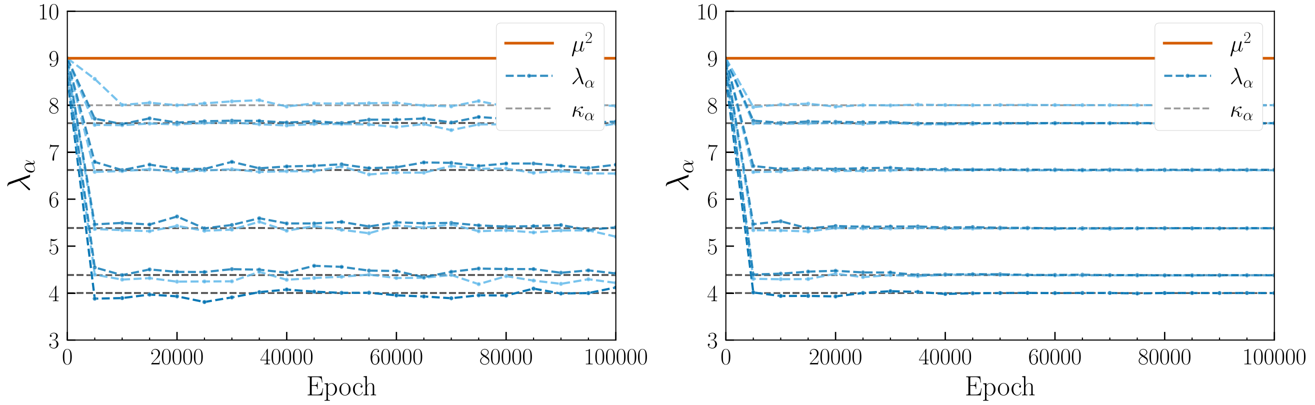


FIG. 18. Scalar field RBM trained without (left) and with (right) learning rate decay, using $r = 0.99$, $N_{\text{epoch}}^{\text{rate}} = 128$, and a fixed RBM mass parameter $\mu^2 = 9$.

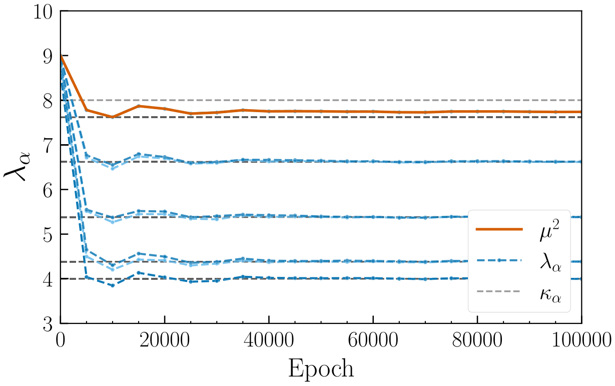


FIG. 19. Scalar field RBM with trainable RBM mass parameter μ^2 and learning rate decay as above. The model parameters are frozen before the RBM mass parameter reaches the (expected) largest eigenvalue of the target kernel.

truncation of the higher modes provided a target goodness is achieved. One can also employ an adaptive learning rate decay.

We have also looked at employing momentum optimization and L_2 regularization of the coupling matrix but have found no need for these.

APPENDIX B: KULLBACK-LEIBLER DIVERGENCE

For completeness, we evaluate here the KL divergence in the case that both the target theory and the model are Gaussian, without linear terms. This allows us to compare it with the log-likelihood in the main text. We consider the KL divergence,

$$\mathrm{KL}(p||q) = \int D\phi p(\phi) \log \frac{p(\phi)}{q(\phi,\theta^*)}, \tag{B1}$$

with $p(\phi)$ the target distribution and $q(\phi,\theta^*)$ the trained distribution (hence the asterisk on θ). We assume the learning process has found the correct eigenbasis, such that the distributions are

$$p(\phi) = \frac{1}{Z_p} e^{-\frac{1}{2}\sum_i a_i \phi_i^2}, \quad Z_p = \prod_i \int d\phi_i e^{-\frac{1}{2}a_i\phi_i^2}, \tag{B2}$$

$$q(\phi,\theta^*) = \frac{1}{Z_q} e^{-\frac{1}{2}\sum_i b_i \phi_i^2}, \quad Z_q = \prod_i \int d\phi_i e^{-\frac{1}{2}b_i\phi_i^2}, \tag{B3}$$

where all eigenvalues $a_i, b_i > 0$. To make the connection with the scalar theory with $N_h < N_v$ in Sec. V, we note that $i = 1, \ldots, N_v$, and that after training,

$$b_i = \begin{cases} \kappa_i & i \le N_h \\ \mu^2 & i > N_h \end{cases}. \tag{B4}$$

It is then straightforward to evaluate the KL divergence. In particular,

$$\log \frac{p(\phi)}{q(\phi,\theta^*)} = -\frac{1}{2}\sum_i (a_i - b_i)\phi_i^2 - \log \frac{Z_p}{Z_q}, \tag{B5}$$

with

$$\log \frac{Z_p}{Z_q} = \frac{1}{2}\sum_i \log \frac{b_i}{a_i}. \tag{B6}$$

Putting everything together, one finds

$$\mathrm{KL}(p||q) = \frac{1}{2}\sum_i \left(-1 + \frac{b_i}{a_i} - \log \frac{b_i}{a_i}\right) \ge 0. \tag{B7}$$

Each term is non-negative, and $\mathrm{KL}(p||q) \ge 0$, as it should be. The equality is achieved only when each eigenvalue is correctly determined. For the scalar theory in Sec. V, this becomes

$$\mathrm{KL}(p||q) = \frac{1}{2}\sum_{i=N_h+1}^{N_v} \left(-1 + \frac{\mu^2}{\kappa_i} - \log \frac{\mu^2}{\kappa_i}\right). \tag{B8}$$

[1] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. V.-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).

[2] P. Mehta, M. Bukov, C.-H. Wang, A. G. R. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, A high-bias, low-variance introduction to machine learning for physicists, Phys. Rep. **810**, 1 (2019).

[3] K. Zhou, L. Wang, L.-G. Pang, and S. Shi, Exploring QCD matter in extreme conditions with machine learning, Prog. Part. Nucl. Phys. **135**, 104084 (2024).

[4] D. Boyda *et al.*, Applications of machine learning to lattice quantum field theory, in *Snowmass 2021* (2022), Vol. 2 arXiv:2202.05838.

[5] K. Cranmer, G. Kanwar, S. Racanière, D. J. Rezende, and P. E. Shanahan, Advances in machine-learning-based sampling motivated by lattice quantum chromodynamics, Nat. Rev. Phys. **5**, 526 (2023).

[6] M. Gerdes, P. de Haan, C. Rainone, R. Bondesan, and M. C. N. Cheng, Learning lattice quantum field theories with equivariant continuous flows, SciPost Phys. **15**, 238 (2023).

[7] S. Ariosto, R. Pacelli, M. Pastore, F. Ginelli, M. Gherardi, and P. Rotondo, Statistical mechanics of deep learning beyond the infinite-width limit, arXiv:2209.04882.

[8] D. Bachtis, G. Aarts, and B. Lucini, Quantum field-theoretic machine learning, Phys. Rev. D **103**, 074510 (2021).

[9] J. Halverson, A. Maiti, and K. Stoner, Neural networks and quantum field theory, Mach. Learn. Sci. Tech. **2**, 035002 (2021).

[10] H. Erbin, V. Lahoche, and D. O. Samary, Non-perturbative renormalization for the neural network-QFT correspondence, Mach. Learn. Sci. Tech. **3**, 015027 (2022).

[11] P. Smolensky, Chapter 6: Information processing in dynamical systems: Foundations of harmony theory, in *Parallel Distributed Processing: Volume 1*, edited by D. Rumelhart and J. McLelland (MIT Press, Cambridge, 1986), pp. 194–281.

[12] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. **14**, 1771 (2002).

[13] A. Decelle and C. Furtlehner, Restricted Boltzmann machine: Recent advances and mean-field theory, Chin. Phys. B **30**, 040202 (2021).

[14] R. Karakida, M. Okada, and S. ichi Amari, Dynamical analysis of contrastive divergence learning: Restricted Boltzmann machines with gaussian visible units, Neural Netw. **79**, 78 (2016).

[15] Y. LeCun, C. Cortes, and C. Burges, *The MNIST Database of Handwritten Digits* (1998), http://yann.lecun.com/exdb/mnist/.

[16] G. Aarts, B. Lucini, and C. Park, chanjure/SRBM: v1.0.0, 10.5281/zenodo.10658564.

[17] M. A. Carreira-Perpiñán and G. Hinton, On contrastive divergence learning, Proc. Mach. Learn. Res. **5**, 33 (2005), https://proceedings.mlr.press/r5/carreira-perpinan05a.html.

[18] T. Tieleman, Training restricted Boltzmann machines using approximations to the likelihood gradient, in *Proceedings of the 25th International Conference on Machine Learning, ICML '08*, (New York, NY, USA) (Association for Computing Machinery, New York, 2008), pp. 1064–1071, 10.1145/1390156.1390290.