Statistical Mechanics of Deep Linear Neural Networks: The Backpropagating Kernel Renormalization

Qianyi Li[1,2] and Haim Sompolinsky[2,3,4]

[1]*The Biophysics Program, Harvard University, Cambridge, Massachusetts 02138, USA*
[2]*Center for Brain Science, Harvard University, Cambridge, Massachusetts 02138, USA*
[3]*Racah Institute of Physics, Hebrew University, Jerusalem 91904, Israel*
[4]*Edmond and Lily Safra Center for Brain Sciences, Hebrew University, Jerusalem 91904, Israel*

The groundbreaking success of deep learning in many real-world tasks has triggered an intense effort to theoretically understand the power and limitations of deep learning in the training and generalization of complex tasks, so far with limited progress. In this work, we study the statistical mechanics of learning in deep linear neural networks (DLNNs) in which the input-output function of an individual unit is linear. Despite the linearity of the units, learning in DLNNs is highly nonlinear; hence, studying its properties reveals some of the essential features of nonlinear deep neural networks (DNNs). Importantly, we exactly solve the network properties following supervised learning using an equilibrium Gibbs distribution in the weight space. To do this, we introduce the backpropagating kernel renormalization (BPKR), which allows for the incremental integration of the network weights layer by layer starting from the network output layer and progressing backward until the first layer's weights are integrated out. This procedure allows us to evaluate important network properties, such as its generalization error, the role of network width and depth, the impact of the size of the training set, and the effects of weight regularization and learning stochasticity. BPKR does not assume specific statistics of the input or the task's output. Furthermore, by performing *partial* integration of the layers, the BPKR allows us to compute the emergent properties of the neural representations across the different hidden layers. We propose a heuristic extension of the BPKR to nonlinear DNNs with rectified linear units (ReLU). Surprisingly, our numerical simulations reveal that despite the nonlinearity, the predictions of our theory are largely shared by ReLU networks of modest depth, in a wide regime of parameters. Our work is the first exact statistical mechanical study of learning in a family of deep neural networks, and the first successful theory of learning through the successive integration of degrees of freedom in the learned weight space.

Subject Areas: Statistical Physics

I. INTRODUCTION

Gradient-based learning in multilayered neural networks has achieved surprising success in many real-world problems including machine vision, speech recognition, natural language processing, and multi-agent games [1–4]. Deep learning (DL) has been applied successfully to basic and applied problems in physical, social, and biomedical sciences and has inspired new neural circuit models of information processing and cognitive functions in animals and humans [5,6]. These exciting developments have generated a widespread interest in advancing the theoretical understanding of the success and limitations of DL and, more generally, in computation with deep neural networks (DNNs). Nevertheless, many fundamental questions remain unresolved, including the puzzling ability of gradient-based optimization to avoid being trapped in poor local minima, and the surprising ability of complex networks to generalize well despite the fact that they are usually heavily overparametrized; namely, the number of learned weights far exceeds the minimal number required for perfectly fitting the training data [7,8]. These problems have fascinating ramifications for statistical mechanics, such as energy landscapes in high dimensions, glassy dynamics, and the role of degeneracy, symmetry, and invariances [9–12]. Indeed, statistical mechanics has been one of the most fruitful theoretical approaches to learning in neural networks [13–16]. However, its classical phenomenology of capacity, learning curves, and phase transitions was formulated largely in the context of single-layer or shallow architectures.

In this work, we develop a new statistical mechanical theory, appropriate for learning in deep architectures.

We focus on the statistical mechanics of weight space in deep linear neural networks (DLNNs) in which single neurons have a linear input-output transfer function. DLNNs do not possess superior computational power over a single-layer linear perceptron [17]. However, because the input-output function of the network depends on products of weights, learning is a highly nonlinear process, and it exhibits some of the salient features of the nonlinear networks. Indeed, in very interesting recent work [18,19], the authors investigated the nonlinear gradient descent dynamics of DLNNs. These studies focused on the properties of the dynamic trajectories of gradient-based learning. To tackle the problem analytically, they had to rely on restrictive assumptions about initial weights and on simplifying assumptions about the data statistics. In contrast, we focus on the equilibrium properties of the distribution in weight space induced by learning, allowing us to address some of the fundamental problems in DL, such as the features determining the DNN's ability to generalize despite overparametrization, the role of depth and width, as well as the size of the training set, and the effect of regularization and learning stochasticity (akin to temperature).

To analyze the property of the weight distribution, we consider the posterior probability distribution in the weight space after learning with a Gaussian prior under a Bayesian framework [20–22]. As introduced in Sec. II, the posterior distribution of the weights can also be formulated as a Gibbs distribution with a cost function consisting of the training error and an L_2 weight regularization term. The Bayesian formulation and the Gibbs distribution of the weights have become standard framework for analyzing statistical properties of neural network models and have been applied in various studies on the statistical mechanics of

learning [14,23–27]. In most of our analysis, we constrain ourselves to the zero-temperature limit, in which case, the network attains zero training error when operating below capacity, and the Gaussian prior introduces bias to the weight distribution to favor weights with smaller L_2 norms within the weight space that yields zero training error.

We evaluate statistical properties in weight space induced by DL by successive *backward* integration of the weights layer by layer starting from the output layer. As shown in Fig. 1, each stage of the successive integration of a layer of weights yields an effective Hamiltonian of the remaining upstream weights. As Sec. II shows, this effective Hamiltonian is expressed in terms of a renormalized kernel matrix K_l, which is the $P \times P$ matrix of the overlaps of all pairs of vectors of activations of the lth layer induced by the P inputs of the training set. This matrix is a function of all upstream weights, and in the successive integration process, it is renormalized by a scalar renormalization variable that summarizes the effect of the integrated downstream weights on the effective Hamiltonian of the remaining weights. Therefore, we refer to the successive backward integration process as backpropagating kernel renormalization (BPKR). Using mean field techniques, this scalar renormalization variable can be evaluated by a self-consistent equation, which is exact in the thermodynamic limit. Thus, our theory is the first exact statistical mechanical study of the weight space properties of DNNs and the first discovery of kernel renormalization of the learned degrees of freedom (d.o.f.). Our BPKR is schematically explained in Fig. 1 and described in detail in Sec. II.

Our work is closely related to the interesting recent research on infinitely wide DNNs [28,29]. It is well known that the ensemble of input-output functions implemented
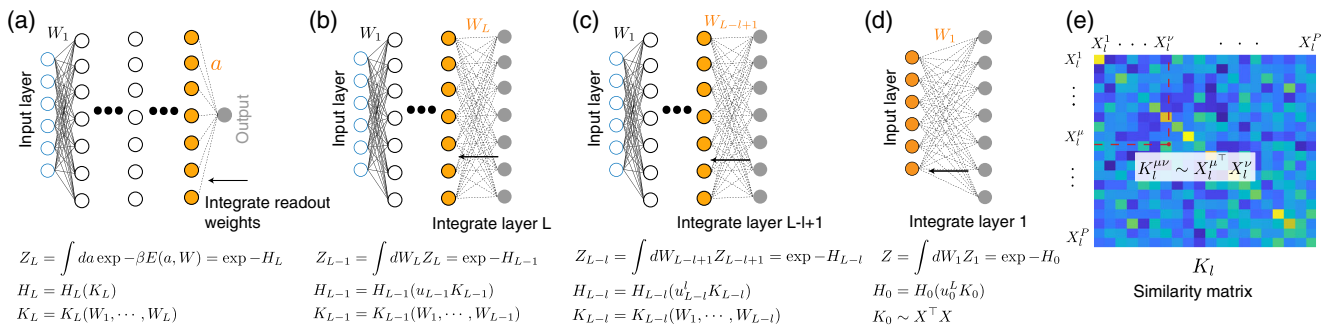

FIG. 1. Schematics of the backpropagating kernel renormalization. (a) Integrating out the readout weights of the network yields a partial partition function Z_L in the weight space with an effective Hamiltonian H_L, which is a function of all the hidden-layer weights through its dependence on the Lth layer kernel matrix [see panel (e)] (with an additional L_2 regularization on the remaining weights neglected here). (b) Integrating out layer L yields a partial partition function Z_{L-1} in the remaining weight space with an effective Hamiltonian H_{L-1}, which has the same structure as H_L except that the $(L-1)$th layer kernel is multiplied by an order parameter u_{L-1}, a scalar renormalization variable that summarizes the effect of the Lth layer weights on the effective Hamiltonian. (c) Similarly, integrating out all weights downstream of layer $L-l$ yields a partial partition function Z_{L-l} with an effective Hamiltonian with an order parameter u_{L-l}, summarizing the effect of all l upstream integrated layers. (d) Integrating out all the weights in the network yields the total partition function Z with the total free energy of the system H_0 with an order parameter u_0, which depends only on the training inputs $X^\mu, \mu = 1, \ldots, P$. (e) The lth layer kernel matrix is the similarity matrix of this layer's responses to the P training inputs up to a normalization factor, and it is a function of all upstream weights.

by infinitely wide networks is equivalent to a Gaussian process (GP) in function space with a covariance matrix defined by a Gaussian kernel, which is the kernel matrix averaged over weights sampled from the Gaussian distribution. This GP limit holds when the network width (the number of neurons in each layer, N) approaches infinity while the size of the training data, P, is held constant, severely limiting its applicability to most realistic conditions. In contrast, our theory holds in the thermodynamic limit assumed in most statistical mechanical studies of neural computation [24,25,30–32], namely, letting both N and P approach infinity while keeping the load $\alpha = P/N$ fixed. As we show here, the behavior of the system at finite α is often qualitatively different from the infinite width limit, and our theory correctly predicts network behavior in a much wider range of parameters that are more relevant in real learning tasks, providing much richer insight into the complex properties of DL in large networks and into the role of the training data in shaping the network performance and emergent representations. Our theory applies to the entire range of $0 \leq \alpha \leq \infty$. We introduce the new notions of wide and narrow networks depending on whether α is smaller or larger than 1. Narrow networks, in particular, deviate qualitatively from the $\alpha \to 0$ limit of the GP theory. This is because when $\alpha > 1$, a zero training error solution cannot be achieved just by optimizing the readout weights but necessarily requires appropriate changes in the hidden-layer weights.

In Sec. III, we apply our theory to derive the network generalization performance and its dependence on the architectural parameters of width and depth as well as the size and statistics of the training data set. We calculate the system's phase diagram and show the important role of the L_2 weight regularization parameter σ. In Sec. IV, we present two important extensions. First, we extend our network architecture to include multiple outputs (denoted as $m > 1$) and show that, in this case, the BPKR is characterized by an $m \times m$ kernel renormalization matrix. Interestingly, their m eigenvalues are independent and obey self-consistent equations similar to the single-output case. While most of our study focuses on the zero-temperature limit of the weight space Gibbs distribution, taking into account only the portion of weight space that yields zero training error, we show in Sec. IV that our BPKR is readily applicable to the finite-temperature case and discuss the way kernel renormalization affects the effect of temperature for networks of different depths.

The power of the BPKR is that it allows the computation of not only the system's performance as a whole but also the representation of the data at each layer, readily captured by the statistics of the mean layerwise kernel matrices. We show in Sec. V how both the input and the task statistics affect these representations, for instance, revealing underlying block structures of the task. In Sec. VI, we present a heuristic extension of the BPKR to nonlinear deep

networks and numerically test its predictions for rectified linear units (ReLU) networks. Surprisingly, we find that this approximation nicely predicts the behavior of ReLU networks with modest depth and not too small width, N. Our results are discussed in the last section.

II. BACKPROPAGATING KERNEL RENORMALIZATION FOR DLNNs

A. Statistical mechanics of learning in deep networks

We consider a multilayer network with L hidden layers whose input-output mapping is given by

$$f(x, \Theta) = \frac{1}{\sqrt{N_L}} \sum_{i=1}^{N_L} a_i \phi_i(x, W), \qquad (1)$$

where $x \in R^{N_0}$ is an input vector of dimension N_0, and ϕ_i is the response of a neuron in the top hidden layer (of size N_L) to that input. In general, ϕ is a nonlinear function of x and the network's hidden weights, denoted by W. The output of the network is a scalar that linearly sums the top layer activations weighted by the readout weights a_i. We denote all network weights by $\Theta = (a, W)$.

We assume supervised learning with the following cost function,

$$E(\Theta) = \frac{1}{2} \sum_{\mu=1}^{P} (f(x^\mu, \Theta) - y^\mu)^2 + \frac{T}{2\sigma^2} \Theta^\top \Theta. \qquad (2)$$

The first term is the mean-squared deviation of the network outputs on a set of P training input vectors x^μ, $\mu = 1, \ldots, P$, from their target labels y^μ. The second term, with amplitude $T\sigma^{-2}$, is a regularization term that favors weights with small L_2 norm. The temperature parameter T in this term means that the L_2 regularization acts as an entropic term. In particular, in the regime we are mostly interested in, $T \to 0$, the first term will enforce minimization of the training error while the L_2 term shapes the statistical measure of the weight vectors that minimize the training error, biasing it in favor of weights with small norms. Without this term, all weights that minimize the error would have the same probability. On the other hand, the L_2 term is irrelevant at low T if the minimum of the error is unique. We call the parameter σ the weight *noise* parameter as it controls the amount of fluctuation in the weights at zero T.

We investigate the properties of the equilibrium distribution of the weights, defined by the Gibbs distribution, $P(\Theta) = Z^{-1} \exp(-E/T)$, where Z is the partition function, $Z = \int d\Theta \exp(-E/T)$. The Gibbs distribution is equivalent to the posterior distribution of the weights with a Gaussian prior.

The fundamental statistical mechanical properties of the system can be derived from the partition function and its

extensions as shown below. However, calculating Z exactly is intractable, but integrating out the readout weights is straightforward; by doing this, we write $Z = \int dW Z_L(W)$, $Z_L(W) = \int da \exp(-E/T) = \exp[-H_L(W)]$, where $H_L(W)$ is the effective Hamiltonian of the hidden-layer weights W after integrating out the readout weights a [Fig. 1(a), see details in the Appendix A],

$$H_L(W) = \frac{1}{2\sigma^2} \text{Tr} W^\top W + \frac{1}{2} Y^\top (K_L(W) + TI)^{-1} Y$$
$$+ \frac{1}{2} \log \det(K_L(W) + TI), \qquad (3)$$

where Y is the $P \times 1$ column vector of the training target labels. The matrix K_L is a $P \times P$ kernel matrix of the top layer. We assume, for simplicity, that all the hidden layers have equal width N. For each layer, we define its kernel matrix by

$$K_l = \frac{\sigma^2}{N} X_l^\top X_l, \qquad (4)$$

where X_l is the $N \times P$ matrix of activation of the lth layer in response to the training inputs, $X_{i,l}^\mu = \phi_i^l(x^\mu, W')$, and $W' = \{W_k\}_{k<l+1}$ denotes all the weights upstream of l. [Importantly, unlike other uses of kernels (e.g., in SVMs and DNNs [28,29]), here we define kernels as simply the unaveraged dot products of the representations at the corresponding layers; hence, the lth kernel matrix is a function of all the weights upstream of X_l, and, in particular, K_L depends on all the hidden-layer weights W.]

The results we derive throughout this work are exact in the thermodynamic limit, which is defined as $N, N_0, P \to \infty$, while $\alpha = P/N$ and $\alpha_0 = P/N_0$ remain finite. Aside from these limits, we do not make any assumptions about the training inputs x^μ or the target outputs Y.

Although our theory is developed for all temperatures (Sec. IV B and Appendix A), our primary focus is on the limit of zero temperature, exploring the statistical properties of the solution weight space, namely, the space of all Θ that yields zero training error. The zero T theory is particularly simple for $\alpha < 1$, in which case the kernel matrices in Eq. (4) are full rank. Substituting the zero-temperature limit, $H_L(W)$ reduces to

$$H_L(W) = \frac{1}{2\sigma^2} \text{Tr} W^\top W + \frac{1}{2} Y^\top K_L(W)^{-1} Y$$
$$+ \frac{1}{2} \log \det[K_L(W)]. \qquad (5)$$

We call networks with $N > P$ wide networks. Narrow networks ($\alpha > 1$) will be discussed at the end of this section.

Integrating over the weight matrices $\{W_k\}_{k\leq L}$ is an intractable problem, in general. Here, we focus on the simple case where all the input-output functions ϕ are linear, so $x_{i,l} = (1/\sqrt{N}) w_l^{i\top} x_{l-1}$.

B. Backpropagating kernel renormalization

Even in the linear case, the Hamiltonian Eq. (5) is not quadratic in the weights; thus, integrating out the weights is highly nontrivial. Instead, we compute the full partition function Z by successive integrations; in each of them, only a single-layer weight matrix is integrated, and it yields a partial partition function of the remaining d.o.f. in the "weight space" (see schematics in Fig. 1). Starting from the top-layer W_L, we can move backward until all weights are integrated out. Integrating the top hidden-layer weight matrix W_L yields a partial partition function, $Z_{L-1} = \int dW_L Z_L(W) = \exp[-H_{L-1}]$,

$$H_{L-1}(W', u_{L-1}) = \frac{1}{2\sigma^2} \text{Tr} W'^\top W' + \frac{1}{2u_{L-1}} Y^\top K_{L-1}^{-1} Y$$
$$+ \frac{1}{2} \log \det(K_{L-1} u_{L-1}) - \frac{N}{2} \log u_{L-1}$$
$$+ \frac{1}{2\sigma^2} N u_{L-1}, \qquad (6)$$

where $W' = \{W_k\}_{k<L}$ denotes all the weights upstream of W_L [schematically shown in Fig. 1(b)]. The first three terms are similar in form to H_L, Eq. (5), with $K_{L-1}(W')$ denoting the $P \times P$ kernel matrix of the $L-1$ layer [Eq. (4) with $l = L - 1$], which is now a function of W'. The kernel terms in H_{L-1} are renormalized by a scalar u_{L-1}, representing the effect of the integrated W_L on the effective Hamiltonian of W'. While u_{L-1} originally appears as an auxiliary integration variable (Appendix A), in the thermodynamic limit, it is an order parameter determined self-consistently by minimizing H_{L-1},

$$1 - \sigma^{-2} u_{L-1} = \alpha(1 - u_{L-1}^{-1} r_{L-1}), \qquad (7)$$

where we have denoted, for general l,

$$r_l = \frac{1}{P} Y^\top K_l^{-1} Y. \qquad (8)$$

We call this quantity the lth layer mean-squared readout since it equals the squared mean of the vector of output weights that read out the target labels directly from layer l (after training the full network) (see Appendix A). As will be shown, the mean-squared readouts are key parameters that capture the effect of the task on the properties of the trained network.

To proceed, we note that apart from scaling by the order parameter u_{L-1}, Eq. (6) has exactly the same form as Eq. (5); hence, the steps of integration of the remaining weights can be repeated layer by layer. After the lth iteration, we obtain [Fig. 1(c)]

$$H_{L-l}(W') = \frac{1}{2\sigma^2} \mathrm{Tr}\, W'^{\top} W' + \frac{1}{2u_{L-l}^l} Y^{\top} K_{L-l}^{-1} Y$$
$$+ \frac{1}{2} \log \det(K_{L-l} u_{L-l}^l) - \frac{lN}{2} \log u_{L-l}$$
$$+ \frac{lN}{2\sigma^2} u_{L-l}, \tag{9}$$

which is a function of $W' = \{W_k\}_{k<L-l+1}$ [note that the superscript in u_{L-l}^k denotes a power, $u_{L-l}^k \equiv (u_{L-l})^k$]. Thus, at each stage, a scalar kernel renormalization appears, u_{L-l}, summarizing the effect of the downstream layers that have been integrated out. This order parameter obeys the mean field equation

$$1 - \sigma^{-2} u_{L-l} = \alpha(1 - u_{L-l}^{-l} r_{L-l}), \tag{10}$$

the solution of which depends on the remaining weights W' and the task, through the layerwise mean-squared readout, Eq. (8).

Finally, integrating out all the weights yields an equation for the network scalar renormalization factor u_0,

$$1 - \sigma^{-2} u_0 = \alpha(1 - u_0^{-L} r_0), \tag{11}$$

with the input layer's mean-squared readout

$$r_0 = \frac{1}{P} Y^{\top} K_0^{-1} Y. \tag{12}$$

Here, $K_0 = (\sigma^2/N_0) X^{\top} X$ is the input kernel, where X is the $N \times P$ input data matrix.

Standard techniques using the partition function as a generating functional allow for the derivation of important statistics of the system, in particular, its generalization performance. While the statistics of the performance of the system are evaluated by completing the integration over all weights, i.e., $l = L$ as in Eqs. (11) and (12) above, the results of partial weight integration are important in evaluating the properties of the representations in individual layers (Sec. V).

We compare our results to that of the GP theory [28] for infinitely wide networks. In the GP theory, the kernels K_l, Eq. (4), are self-averaged, and furthermore, the weight distribution is Gaussian, so the weight-dependent kernels can be replaced by their average over Gaussian weights (with variances σ^2/N or σ^2/N_0 for the first layer). For a linear network, this amounts to having the kernel of layer l being simply σ^2 times the kernel of layer $l - 1$; hence, $K_l = \sigma^{2l} K_0$.

Importantly, unlike the predictions of the GP theory, we show below that the statistics of the kernel matrices induced by learning are complex, and the relation between kernel statistics at one stage of integration and the next cannot be fully captured by a simple renormalization of the entire matrix by a scalar factor. Instead, different statistics change differently upon integrating the degrees of freedom.

Nevertheless, several quantities depending on the kernel undergo a simple renormalization. In particular, the layerwise mean-squared readout r_l undergoes a simple scaling under weight averaging, i.e.,

$$r_{l-1} = u_{l-1} \langle r_l \rangle_l \tag{13}$$

for all $1 \le l \le L$. The subscript in $\langle \cdot \rangle_l$ denotes averaging over the upstream weights from l to L so that $l - 1$ is the top unintegrated layer.

Likewise, upon successive integrations of all weights, we have $r_0 = u_0^l \langle r_l \rangle$.

This relation is important as it provides an operational definition of the order parameters u_l, which can be used for their direct evaluation in numerical simulations. Also, we show below that the mean and variance of the network predictor transform simply by renormalizing the associated kernels with u_0 [see Eqs. (21) and (20)].

As can be seen from Eq. (11), our results hold when the input kernel is full rank, which implies $\alpha_0 = P/N_0 < 1$. This condition is understandable since for $\alpha_0 > 1$ there is no W that achieves zero training error (in the linear networks). We denote $\alpha_0 = 1$ as the interpolation threshold of our network (below which the training data can be exactly matched). This threshold holds for generic input [i.e., such that the rank of K_0 is $\min(P, N_0)$] and for a target function that is not perfectly realized by a linear input-output mapping (otherwise zero error can be achieved for all α_0). In most of our work, we focus on the properties of zero error solution space; i.e., we assume $\alpha_0 < 1$.

C. BPKR for narrow architectures

As stated above, the BPKR at finite temperatures is well defined for all α. However, the zero-temperature limit is subtle when $\alpha > 1$ since the $P \times P$ kernel matrices, Eq. (4), of the hidden layers are of rank $N < P$, while we have assumed above that the kernel matrices are invertible. Indeed, the above results, Eqs. (6)–(10) and (13) hold only for $\alpha < 1$.

This difference between wide and narrow architectures reflects the difference in the impact of learning on W in the two regimes. While in the wide regime, even for generic untrained W, the training data can be perfectly fit by an appropriate choice of readout weights a, in the narrow regime, a perfect learning of the task cannot be achieved without an appropriate modification of W. Specifically, at every stage of the integration, after averaging out the weights upstream from the lth layer, the remaining weights must ensure that Y is in the N-dimensional subspace of $\mathbb{R}^P$ spanned by the N (P-dimensional) vectors $X_{i,l}, i = 1, \ldots, N$, induced by the P training inputs.

As shown in Appendix A, these constraints lead to replacement for Eq. (10) by

$$u_{L-l}^{l+1} = \alpha \sigma^2 r_{L-l}, \tag{14}$$

where the layer mean-squared readout at zero temperature, r_{L-l}, is given by

$$r_{L-l} = \frac{1}{P} Y^\top K_{L-l}^+ Y \tag{15}$$

for $1 \leq l < L$. Here, K_{L-l} is the kernel of the $(L-l)$th layer for a set of weights W_{L-l} that yields zero training error, and K_{L-l}^+ denotes the pseudo-inverse of K_{L-l}. In addition, the scaling relationship, Eq. (13), which provides an operational definition of the kernel renormalization OPs, still holds for $1 \leq l < L$. However, for the average of r_l over *all* hidden weights, the relation with r_0 is given by

$$\langle r_l \rangle = u_0^{-l} r_0 - 1 + \frac{1}{\alpha}, \tag{16}$$

where r_0 is given by Eq. (12) (see Appendix A and Sec. IA in the Supplemental Material [33] for details). Thus, $\langle r_l \rangle$ has a cusp as a function of α at $\alpha = 1$ (see Fig. 1 in Ref. [33]).

Importantly, Eqs. (11) and (12) between u_0 and r_0 hold for $0 \leq \alpha < \infty$, as K_0 is full rank as long as $\alpha_0 < 1$. Hence, many important system properties, such as the generalization error and the predictor statistics which depend on α through u_0, are *smooth* functions of α for all α (Sec. III).

D. Predictor statistics

The generalization performance is closely related to the learning-induced statistics of the predictor, Eq. (1), for a new input vector x. First, we note that when the hidden-layer weights W are fixed, the predictor $f(x)$ obeys Gaussian statistics (from the fluctuations in the readout weights a) with

$$\langle f(x) \rangle_a = k_L^\top(x_L) K_L^{-1} Y, \tag{17}$$

$$\langle [\delta f(x)]^2 \rangle_a = K(x_L, x_L) - k_L^\top(x_L) K_L^{-1} k_L(x_L), \tag{18}$$

where x_L is the vector of top-layer activations in response to the new input x; $k_L(x_L)$ is the $P \times 1$ vector given by $k_L^\mu(x_L) = K(x_L, x_L^\mu)$, where for any two vectors x, y,

$$K(x, y) = N^{-1} \sigma^2 x^\top y. \tag{19}$$

The subscript a in Eqs. (17) and (18) denotes averaging with respect to a only. Thus, the moments of the predictor depend on W through the $P \times P$ kernel matrix K_L and through x_L and x_L^μ. Evaluating its first two moments with respect to the full averaging (over Θ), we find (Appendix B)

$$\langle f(x) \rangle = k_0^\top(x) K_0^{-1} Y, \tag{20}$$

with k_0 the $P \times 1$ vector given by $k_0^\mu = K_0(x, x^\mu)$, where for any two input vectors x, y, $K_0(x, y) = N_0^{-1} \sigma^2 x^\top y$. Thus, at zero temperature, the mean predictor is independent of the network architecture or noise level σ and retains its value predicted by the GP limit. This makes sense as, at zero temperature, multiplying the kernels in the numerator and denominator by a scalar cancels out. The variance of the predictor takes into account the W average of Eq. (18), which is the mean contribution from the fluctuations in a as well as the variance of the conditioned mean Eq. (17). These contributions produce the following simple result,

$$\langle [\delta f(x)]^2 \rangle = u_0^L [K_0(x, x) - k_0^\top(x) K_0^{-1} k_0(x)] \tag{21}$$

(Appendix B). Thus, the predictor variance equals the variance of the $L = 0$ network [Eq. (18) for $L = 0$] scaled by the kernel renormalization factor u_0^L, which makes sense since the variance scales linearly with the kernel. This variance renormalization due to the presence of hidden layers has an important impact on the generalization error, which depends on both moments of f and can be written as $\varepsilon_g(x) = \langle [f(x) - y(x)]^2 \rangle = [\langle f(x) \rangle - y(x)]^2 + \langle [\delta f(x)]^2 \rangle$, where $y(x)$ is the target label of the new input x.

The network properties after integrating the weights of all hidden layers depend on the kernel renormalization factor u_0 but not the intermediate renormalization factors u_l ($1 \leq l < L$). We summarize below several important expressions for the renormalization factor u_0 and the equations for the predictor statistics that depend on u_0, which holds for $0 \leq \alpha < \infty$. These results are the main conclusions from this section, and they will be useful for analyzing the generalization performance and how it depends on various network parameters in Sec. III.

Kernel renormalization factor u_0:

u_0 relates the average of the top-layer mean-squared readout r_L (over all L weight matrices) to the input-layer mean-squared readout r_0

$$r_L = \frac{1}{P} Y^\top K_L^+ Y \tag{22}$$

and

$$r_0 = \frac{1}{P} Y^\top K_0^{-1} Y \tag{23}$$

through

$$r_0 = u_0^L \left[\langle r_L \rangle - \max \left(1 - \frac{1}{\alpha}, 0 \right) \right]. \tag{24}$$

The matrix K_L^+ is the (psuedo) inverse of the top-layer representation, $K_L = (\sigma^2/N) X_L^\top X_L$, and K_0 is the input-layer kernel.

The predictor statistics for an input x is

$$\langle f(x) \rangle = k_0^\top(x) K_0^{-1} Y, \qquad (25)$$

$$\langle [\delta f(x)]^2 \rangle = u_0^L [K_0(x,x) - k_0^\top(x) K_0^{-1} k_0(x)], \quad (26)$$

where $k_0(x) = K_0(x, X_0)$ and X_0 stands for the P training vectors.

The self-consistent equation for u_0 is

$$1 - \sigma^{-2} u_0 = \alpha(1 - u_0^{-L} r_0). \qquad (27)$$

E. Qualitative differences between wide and narrow architectures

To highlight the qualitative differences between wide and narrow architectures, we use the equation for u_0 given by Eq. (27) and show in Figs. 2(a) and 2(b) u_0 vs σ for different α (see Sec. IIA in Ref. [33]). Note because K_0 scales with σ^2, when we vary σ, we hold $\sigma^2 r_0$ constant. The limit of infinite width corresponds to $\alpha \to 0$. In this limit, Eq. (27) yields $u_0 = \sigma^2$, which is the prediction of the GP theory for a linear network. First, in contrast to the GP theory, for finite α, u_0 attains a nonzero value for $\sigma \to 0$. Furthermore, the dependence of u_0 on σ is qualitatively different in the wide and narrow regimes. For $\alpha < 1$, u_0
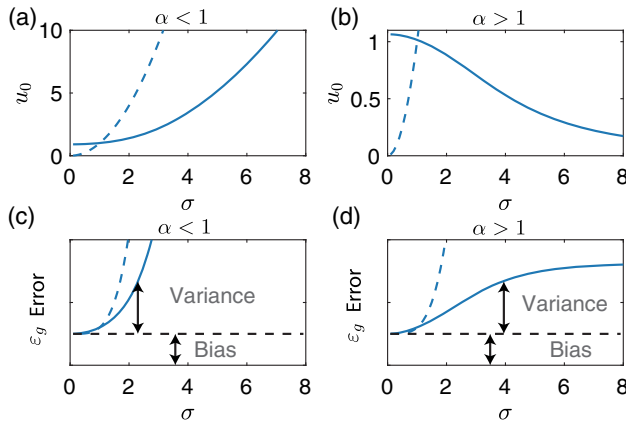

FIG. 2. (a,b) Dependence of the order parameter u_0 on the noise parameter σ in wide [(a); $\alpha = 0.8$] and narrow networks [(b); $\alpha = 1.1$]. Blue solid lines: theory. Blue dashed lines: the prediction of GP theory ($u_0 = \sigma^2$). In both panels (a) and (b), u_0 is finite for low σ. Additionally, in panel (a), ($\alpha < 1$) u_0 diverges as $\sigma^2(1 - \alpha)$ for $\sigma \to \infty$, slower than in the GP theory. In panel (b), for $\alpha > 1$, u_0 vanishes as $\sigma \to \infty$, which is drastically different from the GP. (c,d) Dependence of the generalization error on σ for wide ($\alpha = 0.8$) and wide ($\alpha = 1.1$) regimes. The change with σ is due to the change in variance, which scales as $\sigma^2 u_0$ [Eq. (26)]. The bias contribution is independent of σ, Eq. (25) and black dashed lines. (c) Generalization error diverging slower than in GP theory for $\alpha < 1$ as $\sigma \to \infty$. (d) Generalization error increasing and approaching a finite limit as $\sigma \to \infty$ for $\alpha > 1$, in stark contrast to the divergence predicted by the GP theory.

increases monotonically with σ, diverging for large σ, $u_0 \to \sigma^2(1 - \alpha)$. Importantly, the behavior is reversed for $\alpha > 1$. Here, u_0 decreases monotonically with σ and vanishes for large σ as $u_0^L \to \sigma^{-2}[\alpha\sigma^2 r_0/(\alpha - 1)]$. This difference in behavior, particularly for large σ, reflects the differences in the effect of learning on the weight space, as discussed at the beginning of Sec. II C. Learning imposes more constraints on W in the narrow regime, and only a small fraction of W have nonzero Gibbs probability, deviating strongly from the predictions of the GP limit.

III. GENERALIZATION

In a linear network, the mapping between input and output is given by an N_0-dimensional effective weight vector proportional to $a^\top W_L W_{L-1} \cdots W_1$. As mentioned above, here we assume the system is below the interpolation threshold, i.e., $\alpha_0 = P/N_0 < 1$; hence, our network perfectly learns the training input-output relations as $T \to 0$, even without hidden layers (i.e., $L = 0$). Thus, our deep network [with $L \geq 1$ and $\alpha = P/N$ of $\mathcal{O}(1)$] is always in the heavily overparametrized regime, where the number of modifiable parameters is much larger than the number of parameters needed to satisfy the training data. Naively, this would imply that the system is extremely poor in generalization. However, as we will show, this is not necessarily the case due to the presence of "inductive bias" in the form of L_2 regularization. In this section, we discuss how the generalization error depends on various network parameters, including the noise parameter σ, the network width N, and the network depth L, which may provide helpful insights for selecting network parameters during training.

A. Dependence of generalization on noise

From the predictor statistics, Eqs. (25) and (26), we conclude that the contribution of the squared bias $[\langle f(x) \rangle - y(x)]^2$ to ε_g is constant, independent of the network parameters N, L, and σ. As for the contribution from the variance, this tracks the behavior of $u_0^L \sigma^2$ (the factor σ^2 stems from the noise dependence of the kernels). Thus, from our previous analysis of u_0 in Sec. II C, we can predict the generalization error's dependence on the noise, as shown schematically in Fig. 2(c) for wide and in Fig. 2(d) for narrow networks. In both regimes, the variance grows monotonically with noise, but for large noise, in the narrow regime, the generalization error does not diverge but saturates to a finite value $\alpha r_0 \sigma^2/(\alpha - 1)$ (r_0 scales as σ^{-2}), while in the wide regime, the generalization error diverges as $\sigma^4(1 - \alpha)$ (see Ref. [33], Sec. IIA).

B. Dependence of generalization on width

We now consider in detail the dependence of ε_g on α for different levels of noise. A detailed analysis (see Ref. [33],

Sec. IIB) shows that when other parameters are fixed, the generalization error varies monotonically with width, increasing if

$$\sigma^{2(L+1)} > \sigma^2 r_0. \tag{28}$$

Otherwise, it decreases with width. The latter case is an example where, despite increasing model complexity through increasing N, generalization performance improves, as shown in Fig. 3. In the example shown, we use normally distributed training input vectors and training labels Y generated by a noisy linear teacher. The generalization error is measured here on the network outputs generated by inputs, which are corrupted versions of the training vectors (detailed in Appendix E). Thus, this example corresponds to the case where the training data play the role of P templates and the test inputs are sampled from Gaussian noise around these templates [34]. The model introduces a relation between target outputs and data statistics, which is a more realistic situation than vanilla, normally distributed test data, where there is no inherent relation between inputs and outputs (aside from weak correlation induced by the random teacher weights). We emphasize that although we choose a specific example to present our numerics, our theory is not limited to any specific type of input-output distribution.

C. Dependence of generalization on depth

First, we discuss the limit of large L, analyzing the fixed point of Eq. (11), i.e., the solution for $u_0(L \to \infty)$.

We recall that in the GP limit, $u_0(L) = u_0 = \sigma^{2L}$; hence, if $\sigma < 1$, $u_0 \to 0$, and the entire deep network collapses to a single-layer network, or if $\sigma > 1$, u_0 diverges. Thus, in order to obtain a nontrivial behavior, the noise needs to be

fine-tuned to $\sigma = 1$. Similar fine-tuning is required, in the GP theory, in nonlinear networks in the limit of large L. As we show below, the behavior of our networks is strikingly different. In fact, we show that in the low-noise regime, the system is self-tuned in that u_0 approaches a finite fixed-point value.

Specifically, Eq. (11) predicts that, in the low-noise regime, defined by

$$\sigma^2(1 - \alpha) < 1, \tag{29}$$

$u_0 \to u_\infty = 1$, independent of σ. Furthermore, the approach to this limit is inversely proportional to L, $u_0 \approx 1 - (v_0/L)$, where the prefactor v_0 obeys

$$\exp -v_0 = \frac{\alpha\sigma^2 r_0}{[1 - \sigma^2(1 - \alpha)]}. \tag{30}$$

Thus, there are two subregimes. If $\alpha\sigma^2 r_0 < 1 - \sigma^2(1 - \alpha)$, then $v_0 > 0$, implying that $u_0(L) < 1$ and increases with L toward its fixed-point value 1. If this inequality does not hold, u_0 decreases with L towards its fixed point. Note that narrow networks in which $\alpha > 1$ are always in the low-noise regime given by Eq. (29), for all values of σ.

In contrast, in the high-noise regime $\sigma^2(1 - \alpha) > 1$, the fixed-point value is $u_0 \to u_\infty = \sigma^2(1 - \alpha) > 1$. These results have important implications for the predictor variance and the generalization error. Inspecting Eq. (21), we conclude that in the low-noise regime, the predictor variance and the generalization error reach a finite value since $u_0^L \to \exp -v_0$, which depends on α, σ, and r_0. On the other hand, in the high-noise limit, u_0^L diverges exponentially with L, yielding a divergent generalization error.
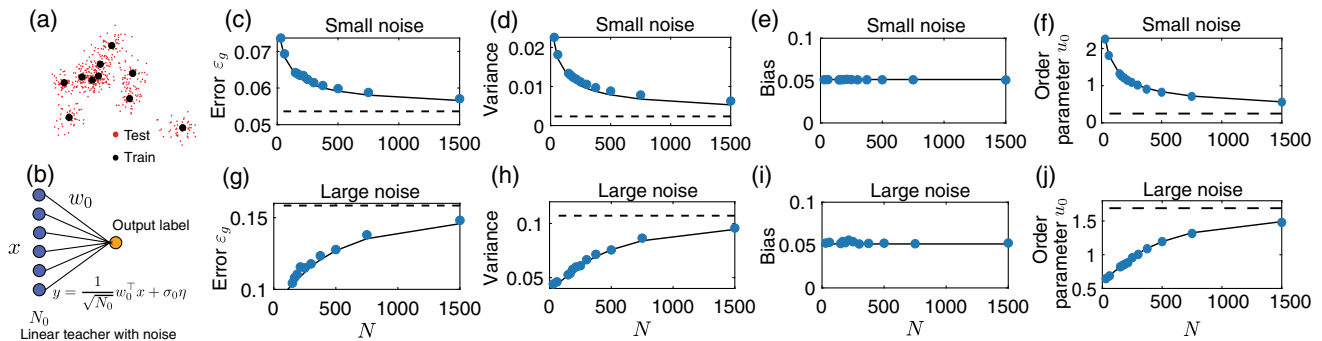

FIG. 3. Dependence of network behavior on hidden-layer width N. (a) Schematics of inputs in the template model: N_0-dimensional vectors clustered around P templates (shown as black dots), which are used as training vectors. The testing data (shown as red dots) are sampled from the P clusters by adding Gaussian noise to the templates (see Appendix E). (b) Target labels are outputs of a single-layer linear teacher network with weight vector w_0 and additive output noise. (c)–(j) Behavior of a linear network with a single hidden layer ($L = 1$) vs hidden-layer size (width) N: generalization error (c,g), variance (d,h), and bias (e,i) of the predictor averaged over the test data, and normalized by the amplitude of the labels, and the order parameter u_0 (f,j). We performed the same averaging and normalization for all the following results. Black solid lines: theory. Blue dots: simulation. Black dashed lines: GP limit ($N = \infty$). Top row: parameter regime with small noise where the generalization error increases with α (i.e., decreases with N). Bottom row: parameter regime with large noise where the generalization error decreases with α (i.e., increases with N). (See detailed parameters for the simulation in Appendix E.)
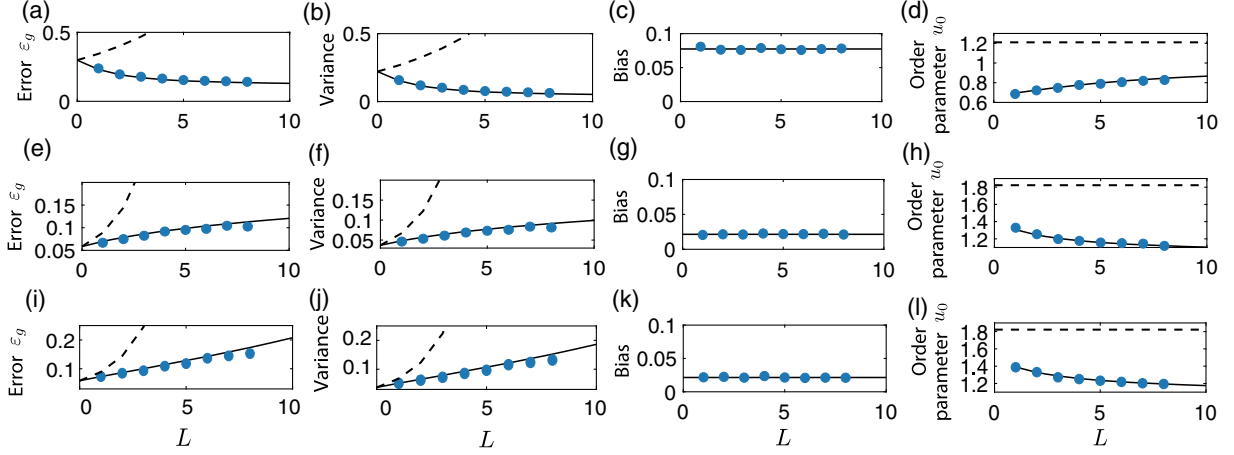
FIG. 4.   Dependence of the generalization error on network depth L. (a,e,i) Generalization error, (b,f,j) variance, (c,g,k) bias of the predictor, and (d,h,l) order parameter u_0 as a function of L. Black solid lines: theory. Blue dots: simulation. Black dashed lines: GP limit ($N \to \infty$). (Details of the example model and parameters are described in Appendix E.) (a)–(d) Subregime where the generalization error decreases with L. (e)–(h) Subregime where the generalization error increases with L approaching a finite limit. (i)–(l) High-noise regime where the generalization error increases with L and diverges as $L \to \infty$.

Further analysis shows that when all other parameters are held fixed, the generalization error is monotonic with L (see Ref. [33], Sec. IIC). In the low-noise regime, it saturates to a finite value, decreasing when $\nu_0 > 0$ (because u_0^L is inversely related to u_0) and increasing otherwise. These three behaviors are illustrated in Fig. 4 for the same template model and the noisy linear teacher task as illustrated in Fig. 3. The phase diagram of the generalization error depicting its different behaviors is shown in Fig. 5. In Fig. 5(a), we show the trend with respect to the width (equivalently, α) in the plane of noise (σ^2) and input mean-squared readout ($\sigma^2 r_0$) (scaled so that it is independent of noise), where the boundaries are given by $\sigma^{2(L+1)} = \sigma^2 r_0$. In Fig. 5(b), we show the behaviors with respect to the depth in the plane of σ^2 and α (for
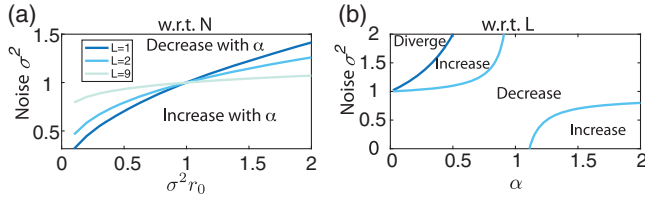

FIG. 5.   Summary plot showing different parameter regimes for the dependence of the generalization error on width (N) and depth (L). (a) Plane of noise σ^2 and $\sigma^2 r_0$ (the input readout parameter r_0 is normalized by σ^2 so that $\sigma^2 r_0$ is independent of σ). Above the boundary lines, the generalization error decreases with α; below it, the generalization error increases with α. The three different curves are for $L = 1$, $L = 2$, and $L = 9$. The separating line becomes flatter as L increases. (b) For a fixed $\sigma^2 r_0$, in the plane of α and σ^2, there are three types of behaviors of ε_g vs L: It can decrease, increase, or diverge. Note that in the narrow regime ($\alpha > 1$), ε_g never diverges, consistent with Eq. (29).

$\sigma^2 r_0 = 0.8$). Note that in the GP limit ($\alpha \to 0$), the behavior of the generalization error with respect to L is either decreasing and goes to 0 or it is divergent.

D. Varying the size of the training set

Until now, we have considered the dependence of ε_g on network parameters for a fixed training set, in particular, fixed training set size P. Here, we consider the effect of varying P. In addition to varying α, α_0, changes in the training set affect r_0, as well as the kernels appearing in Eqs. (20) and (21). The exact effect of changing P depends, of course, on the details of the input and output data. Here, we address what happens near and above the interpolation threshold, $\alpha_0 = 1$. For $\alpha_0 > 1$, the minimal training error is nonzero. However, there is a huge degeneracy of weights that minimize this error, defined by all values of Θ that yield the same input-output linear mapping, given by input-output effective weights (see Ref. [33], Sec. IID) that obey

$$\frac{1}{\sqrt{N_0}} \frac{1}{\sqrt{N^L}} a^\top W_L W_{L-1} \cdots W_1 = (XX^\top)^{-1} XY, \quad (31)$$

which implies that the predictor is uniquely given by

$$f(x) = x^\top (XX^\top)^{-1} XY \quad (32)$$

for all x (whether belonging to the training set or not). Hence, the training and the generalization error in the deep network is identical to that of a single-layer network. The generalization error is given by the bias component of the error, as the predictor variance is zero.

The singularity of the input kernel at $\alpha_0 = 1$ gives rise to a simple example of a double descent in the generalization error. The divergence at the interpolation threshold is

known to be suppressed by the addition of L_2 regularizers [16]. The reason for the persistence of this divergence in our theory is the fact that our L_2 regularization term is scaled by the temperature T [see Eq. (2), which means that, at zero T, it does not lift the degeneracy of the solutions (and the concentration of the solution space in large norm weights at $\alpha_0 = 1$). Indeed, this degeneracy is halted, in our theory, only by finite temperature, as shown in Sec. IV B.

The singularity may drastically affect the behavior near it on both sides of the interpolation threshold. For instance, if the input is sampled from a standard Gaussian independent identically distribution (IID), then r_0 diverges when $\alpha_0 \to 1$ as $\propto |1 - \alpha_0|^{-1}$, leading to vanishing of the variance of the predictor (averaged over the testing example) as $(1 - \alpha_0)^{1/L}$. The sample average of the squared mean, $\langle f(x) \rangle^2$, diverges as $\propto |1 - \alpha_0|^{-1}$ (see Appendix B for the assumptions made here and additional analysis in Ref. [33], Sec. IID). Hence, the generalization error on both sides of $\alpha_0 = 1$ is dominated by the bias and diverges as $|1 - \alpha_0|^{-1}$. This nonmonotonicity of ε_g is reminiscent of double descent [35,36]. However, genuine double descent, namely, nonmonotonicity of ε_g when increasing the number of network parameters for a fixed training set does not occur in our system since the error is always monotonic with α (as shown in Fig. 3). In Fig. 6, we show these results for the template model where the inputs are clustered, with two rules for the labels: a noisy linear teacher as in Fig. 3 and random labels where the label for each cluster is binary and drawn randomly (both detailed in Appendix E).

For the noisy linear teacher task, the minimum generalization error is achieved on the right-hand side (rhs) of the interpolation threshold. Because of the linearity of the task, we do not need a large number of parameters (i.e., small α_0) to generalize well. However, for the random labeling task, the minimum generalization error is achieved on

the left-hand side (lhs) of the interpolation threshold. Because of the nonlinearity of the task itself, having $N_0 > P$ is required for good performance.

IV. EXTENSIONS

A. Multiple outputs

We now consider the case where there are $m > 1$ linear outputs, with the $N \times m$ readout weight matrix A. The rest of the architecture is the same as above, with L hidden layers of width N. Extending our theory, we obtain, instead of scalar renormalization factors $u_l(m = 1)$, $m \times m$ renormalization matrices $\mathcal{U}_l$ per layer (see details in Appendix C). Here, we focus on the zero-temperature limit. We first consider $\alpha < 1$. We define the hidden-layer $m \times m$ readout covariance matrix $\mathcal{R}_l = (1/P)Y^\top K_l^{-1} Y$. This matrix is diagonalized via

$$\mathcal{R}_l = \frac{1}{P} Y^\top K_l^{-1} Y = V_l \mathrm{diag}(r_{1l}, \ldots, r_{kl}, \ldots, r_{ml}) V_l^\top. \quad (33)$$

Here, $\mathrm{diag}(r_{1l}, \ldots, r_{kl}, \ldots, r_{ml})$ denotes a diagonal matrix with components $\{r_{kl}\}_{k=1,\ldots,m}$, and V_l is the unitary matrix diagonalizing $\mathcal{R}_l$ and it depends on the specific realization of $\{W_k\}_{k<l+1}$. We find that the matrix $\mathcal{U}_l$ is diagonalized by the same unitary matrix V_l, i.e., $\mathcal{U}_l = V_l \mathrm{diag}(u_{1l}, \ldots, u_{kl}, \ldots, u_{ml}) V_l^\top$. Each layerwise renormalization factor u_{kl} obeys the same equation as single-output scalar renormalization,

$$1 - \alpha = \sigma^{-2} u_{kl} - \alpha u_{kl}^{-(L-l)} r_{kl}, \qquad 1 \le k \le m. \quad (34)$$

Similarly, the matrix $\mathcal{U}_0$ is diagonalized by the unitary matrix V_0, which diagonalizes the input-layer $m \times m$ readout covariance matrix,
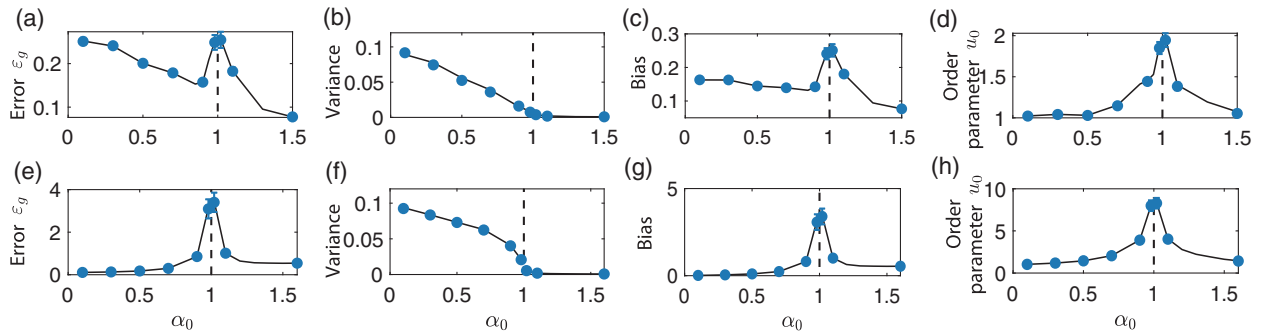

FIG. 6. Dependence of the generalization error on the training set size P for a single hidden-layer ($L = 1$) network. (a,e) Generalization error, (b,f) variance, (c,g) bias of the predictor, and (d,h) the order parameter u_0 as a function of α_0. Note that u_0, $\langle f(x) \rangle^2$ and ε_g all diverge as $\alpha_0 \to 1$ from both sides. The variance of the predictor vanishes as $\alpha_0 \to 1^-$ and remains zero for $\alpha_0 > 1$. Black solid lines: theory. Black dashed lines: interpolation threshold at $\alpha_0 = 1$. Blue dots with error bars: simulations. Error bars show the standard deviation of the mean across different realizations of the noise in the training data labels. (a)–(d) Network trained on the template model and the noisy linear teacher task with parameters described in Appendix E. (e)–(h) Network trained on the template model with random labeling of each cluster, with parameters described in Appendix E. [In this figure, we use the finite T result (Sec. IV B) for the theory curves since, as $\alpha_0 \to 1$, increasingly lower T needs to be used if we want to approximate the zero T limit (see details in Appendix E and Ref. [33], Sec. IIE).]

$$\mathcal{R}_0 = \frac{1}{P} Y^\top K_0^{-1} Y = V_0 \text{diag}(r_{10}, \ldots, r_{k0}, \ldots, r_{m0}) V_0^\top. \quad (35)$$

The renormalization matrix $\mathcal{U}_0$ obeys $\mathcal{U}_0 = V_0 \text{diag}(u_{10}, u_{20}, \ldots, u_{ml}) V_0^\top$, with u_{k0} obeying similar equations as Eq. (11), i.e.,

$$1 - \alpha = \sigma^{-2} u_{k0} - \alpha u_{k0}^{-L} r_{k0}, \quad 1 \le k \le m. \quad (36)$$

Similarly to the single-output case, $\mathcal{R}_l = Y^\top K_l^{-1} Y$ undergoes a matrix product renormalization under weight averaging (Appendix C and Ref. [33], Sec. IIIA), i.e.,

$$\mathcal{R}_{l-1} = \mathcal{U}_{l-1} \langle \mathcal{R}_l \rangle_l. \quad (37)$$

The average of the m-dimensional vector $f(x)$ has the same form as in the single-output case, $\langle f(x) \rangle = k_0^\top(x) K_0^{-1} Y$. The covariance matrix of the predictor is

$$\langle \delta f(x) \delta f(x)^\top \rangle = \mathcal{U}_0^L [K_0(x, x) - k_0^\top(x) K_0^{-1} k_0(x)]$$
$$= V_0 \text{diag}(u_{10}^L, \ldots, u_{k0}^L, \ldots, u_{m0}^L)$$
$$\times V_0^\top [K_0(x, x) - k_0^\top(x) K_0^{-1} k_0(x)]. \quad (38)$$

Thus, the fluctuations in the predictor of each mode (eigenvectors in V_0) are independent and are given as in the scalar output case. However, the overall behavior may be different than in the $m = 1$ case since individual outputs typically consist of contributions from multiple modes. For instance, the generalization error may not be monotonic with either α or with L.

Similar to BPKR for the single-output case, for $\alpha > 1$, the zero-temperature limit is affected by the singularity of the kernel matrices K_l. Equations (33), (34), and (37) hold only for $\alpha < 1$, and they need to be modified for $\alpha > 1$, as we show here. We find (Appendix C and Ref. [33], Sec. IIIB) that the renormalization matrices $\mathcal{U}_l$ are still diagonalized by the unitary matrix V_l, which diagonalizes the $m \times m$ readout covariance readout matrices, $\mathcal{R}_l$; for $\alpha > 1$, they are given as

$$\mathcal{R}_l = \frac{1}{P} Y^\top K_l^+ Y = V_l \text{diag}(r_{1l}, \ldots, r_{kl}, \ldots, r_{ml}) V_l^\top, \quad (39)$$

where the r_{kl}'s are the eigenvalues of $\mathcal{R}_l$ and K_l^+ is the pseudo-inverse of K_l. The renormalization eigenvalues u_{kl} of the matrix $\mathcal{U}_l$ are related to the eigenvalues of $\mathcal{R}_l$ by

$$u_{kl}^{L-l+1} = \alpha \sigma^2 r_{kl}. \quad (40)$$

The relation given in Eq. (37), which provides an operational definition of the matrix kernel renormalization OPs, still holds for $1 \le l < L$. However, for the full average over weights, it is replaced by

$$\langle \mathcal{R}_l \rangle = \mathcal{U}_0^{-l} \mathcal{R}_0 - \left(1 - \frac{1}{\alpha}\right) I \quad (41)$$

(see Appendix C and Ref. [33], Sec. IIIC for details).

As in the single-output case, Eqs. (35) and (36) hold for $0 \le \alpha < \infty$, as K_0 is full rank as long as $\alpha_0 < 1$. Hence, $\mathcal{U}_0$ and other quantities, such as the generalization error, are smooth functions of α for all α.

Similarly as for the single-output case in Sec. II, the network properties after integrating the weights of all hidden layers depend on the kernel renormalization matrix $\mathcal{U}_0$ but not the intermediate renormalization factors $\mathcal{U}_l$ $(1 \le l < L)$. We summarize below several important expressions for the renormalization matrix $\mathcal{U}_0$ and the equations for the predictor statistics that depend on $\mathcal{U}_0$. These expressions hold for $0 \le \alpha < \infty$.

Kernel renormalization $m \times m$ matrix $\mathcal{U}_0$ for a network with m outputs:

The renormalization matrix $\mathcal{U}_0$ relates the average mean-squared top-layer and the input-layer readout covariance matrices via

$$\mathcal{R}_0 = \mathcal{U}_0^L \left[\langle \mathcal{R}_L \rangle - \max\left(1 - \frac{1}{\alpha}, 0\right) I \right], \quad (42)$$

where $\mathcal{R}_L = (1/P) Y^\top K_L^+ Y$ and $\mathcal{R}_0 = (1/P) Y^\top K_0^{-1} Y$. The predictor statistics is

$$\langle f(x) \rangle = k_0^\top(x) K_0^{-1} Y \langle \delta f(x) \delta f(x)^\top \rangle$$
$$= \mathcal{U}_0^L [K_0(x, x) - k_0^\top(x) K_0^{-1} k_0(x)]. \quad (43)$$

Using the diagonal form of $\mathcal{R}_0$, $\mathcal{R}_0 = V_0 \text{diag}(r_{10}, \ldots, r_{k0}, \ldots, r_{m0}) V_0^\top$, the self-consistent equation of $\mathcal{U}_0$ is

$$\mathcal{U}_0 = V_0 \text{diag}(u_{10}, \ldots, u_{k0}, \ldots, u_{m0}) V_0^\top,$$
$$1 - \alpha = \sigma^{-2} u_{k0} - \alpha u_{k0}^{-L} r_{k0}. \quad (44)$$

B. Finite temperature

Until now, we focused on the limit of zero temperature. We now consider briefly the effect of finite temperature, i.e., when the training error is not strictly minimized (see details in Appendix A). Our BPKR framework holds for general temperatures as well, where the sole effect of temperature is to add to the renormalized W-dependent kernel matrix, a regularizing diagonal term, TI [see Eq. (3)]. In particular, after l successive integration of layer weights, the effective Hamiltonian becomes

$$H_{L-l}(W') = \frac{1}{2\sigma^2} \text{Tr} W'^\top W' + \frac{1}{2} Y^\top [u_{L-l}^l K_{L-l} + TI]^{-1} Y$$
$$+ \frac{1}{2} \log \det(u_{L-l}^l K_{L-l} + TI)$$
$$- \frac{lN}{2} \log u_{L-l} + \frac{lN}{2\sigma^2} u_{L-l}, \quad (45)$$

where the kernel still undergoes kernel renormalization with the scalar renormalization factor u_{L-l}^l.

After integration of all the weights, the equations for the kernel renormalization factor u_0 become

$$1 - \sigma^{-2}u_0 = -\frac{1}{N}Y^\top(u_0^L K_0 + TI)^{-2}u_0^L K_0 Y$$
$$+ \frac{1}{N}\text{Tr}((u_0^L K_0 + TI)^{-1}u_0^L K_0). \quad (46)$$

Furthermore, at finite T, the predictor mean and variances are given as

$$\langle f(x) \rangle = u_0^L k_0^\top(x)(u_0^L K_0 + TI)^{-1}Y, \quad (47)$$

$$\langle [\delta f(x)]^2 \rangle = u_0^L[K_0(x,x) - u_0^L k_0^\top(x)(u_0^L K_0 + TI)^{-1}k_0(x)]. \quad (48)$$

Thus, at finite temperature, both the mean and the variance of the predictor differ from their GP counterparts, by renormalization of all kernels by the factor u_0^L.

Although the predictor value for x^μ in the training set is not y^μ [as is evident from Eq. (47)], the regularization provided by finite temperature may improve the generalization error, in particular, in the neighborhood of $\alpha_0 = 1$ where otherwise it would diverge. Indeed, for any specific task, there is an optimal temperature that minimizes the generalization, as shown in the example of Fig. 7(a). Depending on the specific training task and parameters, there exist cases where the optimal temperature is zero and also where the optimal temperature is above zero, in which case the training error for optimal generalization is nonzero. The existence of an optimal T with minimum generalization error may provide guidance for choosing an appropriate regularization strength during training.

The effect of temperature is analogous to the effect of early stopping in the gradient descent dynamics (see Ref. [16] and Ref. [33], Sec. VIII for more details). Finally, we discuss how the effect of temperature changes with the depth of the network. From Eq. (21), we observe that the effect of temperature on generalization performance is controlled by the relative strength between the temperature term TI and the renormalized kernel $u_0^L K_0$. For finite temperature, both TI and K_0 are of $\mathcal{O}(1)$, and the relative strength between the temperature term and the renormalized kernel is thus controlled by $\lambda \equiv u_0^{-L}$; if λ is small, the effect of temperature is also small. The finite-temperature order parameter u_0 behaves, in the limit of large L, similarly to that at zero temperature. In the low-noise regime $\sigma^2(1 - \alpha) < 1$, u_0 approaches unity for large L as $u_0 \approx 1 - (v_0/L)$; hence, λ approaches the finite value $T \exp v_0$, as shown in Fig. 7(b) [see Eq. (A34) in Appendix A]. On the other hand, in the high-noise regime, $\sigma^2(1 - \alpha) > 1$, u_0 approaches a limit larger than 1; hence, λ goes to zero for deep networks for all finite T [Fig. 7(c)], as $L \to \infty$, implying that the temperature term given by TI can be neglected compared to the renormalized kernel, and the behavior of the network becomes similar to the zero-temperature behavior. In Fig. 7(d), we look at the effect of temperature on the generalization performance for networks with different depth L in the large-noise regime. Since the temperature term TI becomes more negligible compared to the renormalized kernel as L increases, we see that the curve becomes shallower as L increases, suggesting the behavior at finite T becomes more similar to $T = 0$; also for larger L, larger T is required to compensate for the small λ to achieve optimal performance.
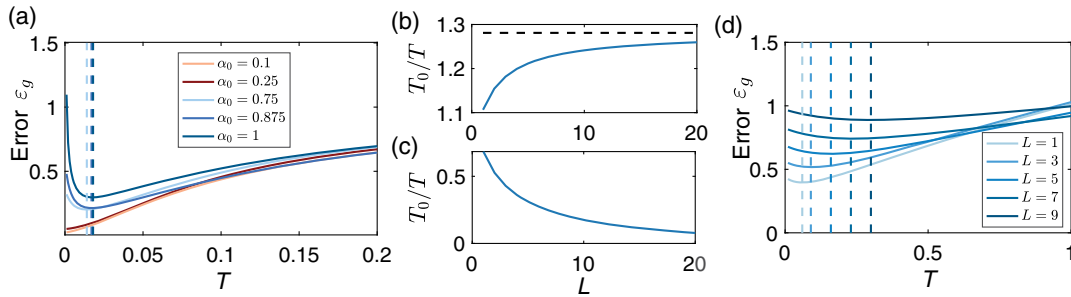

FIG. 7. Finite-temperature theory according to Eqs. (47) and (48) for a network trained on the template model with random labels (see parameters in Appendix E) with finite temperature. (a) Generalization error against the temperature for three different α_0's. Dashed lines: optimal temperature for each α_0. For this specific type of input data, for small α_0 ($\alpha_0 = 0.1, 0.25$), the optimal temperature is at $T = 0$; for α_0 close to 1 ($\alpha_0 = 0.75, 0.875, 1$), the minimum generalization error is achieved at $T > 0$. (b) For $\sigma^2(1 - \alpha) < 1$, u_0 approaches 1 and $\lambda = u_0^{-L}$ approaches the finite value $\exp v_0$ (black dashed line). (c) In another regime where $\sigma^2(1 - \alpha) > 1$, u_0 approaches a limit larger than 1 and $\lambda \to 0$ as $L \to \infty$. (d) Generalization error vs T for five different L's. Dashed lines: optimal temperature for each L. In the regime $\sigma^2(1 - \alpha) > 1$, since λ goes to 0 and the temperature term TI becomes more negligible compared to the renormalized kernel $u_0^L K_0$ as L increases, larger T is required to compensate for the small λ to obtain an optimal generalization error.

V. CHANGES IN REPRESENTATIONS ACROSS LAYERS

Until now, we have discussed the behavior of the network output for different parameter regimes. We now turn to how the representation of the data changes across the different layers.

A. Layerwise mean kernels

The kernel matrices of the hidden layers are an important indicator of the stimulus features represented by these layers, similar to the role of similarity matrices [37,38]. Hence, it is interesting to consider the statistics of the layerwise kernels in our system. We find that the weight-averaged kernel matrices are, to leading order in N, identical to those resulting from Gaussian weights (i.e., as in the GP limit). The non-Gaussianity appears in mean kernels only in the $\mathcal{O}(1/N)$ corrections. Specifically, for the single-output case,

$$\langle K_l \rangle_l = \sigma^2 \left(1 - \frac{1}{N}\right) K_{l-1} + \frac{\sigma^2}{N u_{l-1}^{L-l+1}} YY^\top. \quad (49)$$

The subscript l emphasizes that the average is over W_l and the upstream weights, i.e., $\{W_k\}_{k>l-1}$. Proceeding to successively integrate all upstream weights, the fully averaged kernel of the lth layer ($1 \leq l \leq L$) is given by

$$\langle K_l \rangle = \sigma^{2l} \left(1 - \frac{1}{N}\right)^l K_0 + \frac{m_l}{N} YY^\top, \quad (50)$$

where the amplitudes m_l consist of the sum of the geometric series with terms such as in Eq. (49), yielding $m_l = \sigma^{2l} u_0^{-L} [(u_0^l \sigma^{-2l} - 1)/(u_0 \sigma^{-2} - 1)]$ (see Appendix D and Ref. [33], Sec. VA). Thus, the correction term in Eq. (50) encodes the output task via the output similarity matrix $YY^\top$ (and u_0). In the following analysis and examples, we consider the regime $\sigma \ll 1$, where the second term in Eq. (50) becomes evident compared to the first term. The shape of this correction is the same in all layers; however, its relative strength compared to the GP term increases with the depth of the layer (see Appendix D and Ref. [33], Sec. VB).

The situation is richer in the case of multiple outputs. Here, the layerwise mean kernels are ($1 \leq l \leq L$)

$$\langle K_l \rangle = \sigma^{2l} \left(1 - \frac{m}{N}\right)^l K_0 + \frac{1}{N} Y V_0 M_l V_0^\top Y^\top, \quad (51)$$

where M_l is the diagonal matrix M_l whose kth eigenvalue is $m_{kl} = \sigma^{2l} u_{k0}^{-L} [(\sigma^{-2l} u_{k0}^l - 1)/(\sigma^{-2} u_{k0} - 1)]$. For all l, the maximal eigenvalue of M_l corresponds to the mode with the smallest eigenvalue of R_0, and this mode may dominate the correction to the mean kernel matrix (Appendix D and Ref. [33], Sec. VB). Note that with multiple outputs, the

corrections represented by the last term in Eq. (51) are not simply proportional to the output similarity matrix as in the single-output case. The difference is pronounced if the spectrum of U_0 (or, equivalently, that of R_0) departs substantially from uniformity.

A synthetic example is shown in Fig. 8, in which the P input vectors x_μ are linear combinations of P orthogonal vectors z_i with $z_i^T z_j = \delta_{ij}$,

$$x_\mu = \sum_{i=1}^{P-1} w_i^{\mu\top} z_i + w_P^{\mu\top} z_P. \quad (52)$$

The linear coefficients w_i^μ are sampled IID from $\mathcal{N}(0, I)$ for $i \leq P - 1$ but from $\mathcal{N}(0, \frac{1}{10} I)$ for $i = P$. The output is two dimensional, classifying the inputs according to the sign of their projections on the $(P-1)$th and the Pth basis vectors, respectively [i.e., $Y = \text{sgn}([z_{P-1}, z_P]^\top X) + \sigma_0 \eta$]. As a result, the output similarity matrix $(1/m) YY^\top$ shows four blocks corresponding to the four categories [Fig. 8(b)]. However, because the input is not fully aligned with the output, and the output direction corresponding to z_P has smaller variance than that corresponding to z_{P-1}, the block corresponding to the Pth classification direction is suppressed in the non-GP correction to the kernel of the hidden
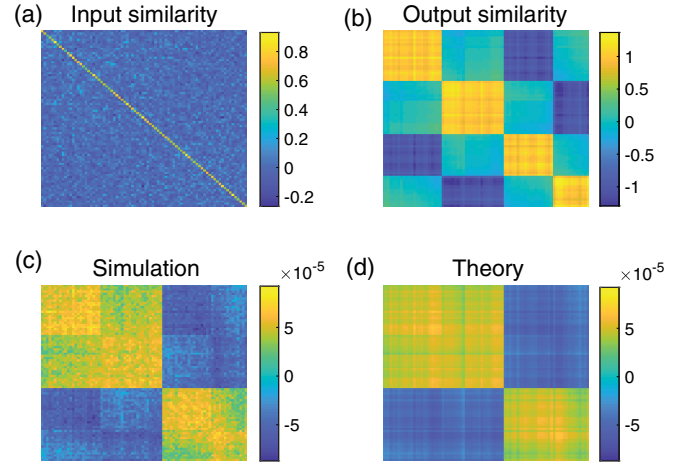

FIG. 8. Hidden representations with multiple outputs. Results for a single hidden-layer network ($L = 1$) trained on the synthetic example described by Eq. (52). In this figure and Fig. 9, to enhance the relative strength of the task-relevant structure, we choose small σ, corresponding to strong regularization. (a) Input similarity matrix. (b) Output similarity matrix, showing four diagonal blocks corresponding to the four categories of the two labels [positive or negative projection onto the $(P-1)$th or Pth dimension]. (c) Simulation result corresponding to the non-GP correction term in Eq. (51), showing only two blocks corresponding to the classification into two categories along the $(P-1)$th direction. The structure of blocks corresponding to the Pth classification direction as shown in panel (b) is now suppressed. (d) Theory for the non-GP correction term, Eq. (51). (See Appendix E for details.)
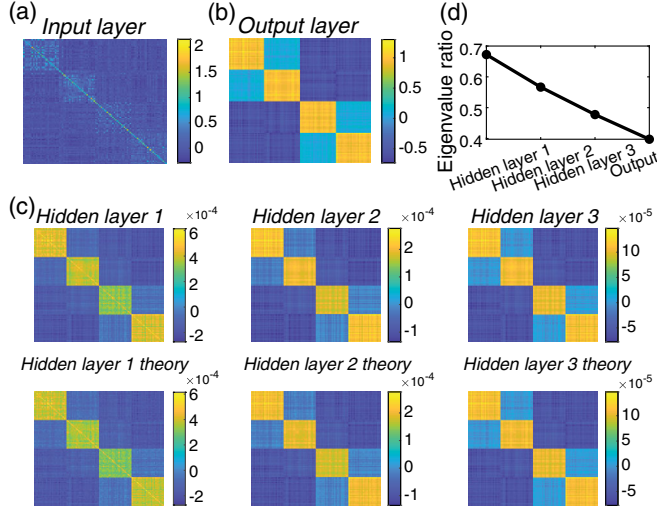
FIG. 9. Simulation and theory for the mean kernel for binary classification tasks on MNIST. The network is trained on four different MNIST digits, which are grouped into two higher-order categories (see Appendix E). The output of the network is six dimensional: Four of the output units are one-hot representations of the four digits; the other two outputs label the inputs according to their high-order category. (a) Input similarity matrix. (b) Output similarity matrix. (c) Average kernel of the hidden layer for $l = 1$, 2, 3. Top: simulation. Bottom: theory. (d) Ratio between the mean of the second and third largest eigenvalues (corresponding to the magnitude of the four smaller blocks) and the largest eigenvalue (corresponding to the magnitude of the two larger blocks) of the non-GP correction terms in the layerwise mean kernel, monotonically decreasing with l.

layer, Eq. (51) [Figs. 8(c) and (d)]. Note that the observed similarity pattern is not a linear combination of the input similarity $(1/N)X^\top X$ matrix [which is almost structureless, Fig. 8(a)] and the output similarity matrix [Fig. 8(b)].

Another example shown in Fig. 9 considers a linear network with three hidden layers and six output units each performing a binary classification task on MNIST input images of four digits (see details of this task in Appendix E). Here, the input similarity matrix $(1/N)X^\top X$ shows a weak but noticeable four-block structure [Fig. 9(a)] corresponding to four different digits. The output similarity matrix $(1/m)YY^\top$ exhibits a pronounced hierarchical block structure [Fig. 9(b)]. We look at how the block structure is modified in the different hidden-layer kernels (Fig. 9).

We observe three major effects of the changes in the average kernels across layers. First, the magnitude of the task-related contribution to the mean kernel increases as l increases, as expected from the theory (Appendix D and Ref. [33], Sec. VB). Second, in this example, we find that the finer-scale structure becomes more pronounced in the mean layer kernel than in the output similarity matrix, as can be seen from comparing Figs. 9(b) and 9(c). Third, the contributions from finer-scale structure become less

pronounced for deeper layers, as seen in Fig. 9(c). The second and third points can also be observed more straightforwardly in Fig. 9(d), where we show the ratio between the mean of the second and third largest eigenvalues (corresponding to the four smaller blocks) and the largest eigenvalue (corresponding to the two larger blocks) of the non-GP correction terms in the layerwise mean kernels. This ratio decreases with l, suggesting that the finer structure becomes less pronounced for large l, and this ratio for all hidden layers is larger than that for the output layer. The origin for agreement of the mean kernels with their GP limit to leading order in N is that the second-order statistics of hidden-layer weights are just their GP values to leading order in N. Their renormalization appears only in the $\mathcal{O}(1/N)$ corrections to their covariance matrix (Appendix D) because the learning-induced terms in the effective Hamiltonians, such as Eq. (9), are of order P (as there are P training constraints), which is of the order of N, while the L_2 Gaussian term is of the order of the number of weights in each layer, which is N^2. On the other hand, the leading term and the correction terms scale differently with σ, such that in the low-noise limit, the strength of the correction relative to the GP term grows as σ^{-2l}/N (see Ref. [33], Sec. VB).

B. Mean inverse kernels

While the average kernel retains, to leading order, its GP value, the average inverse kernel does not. In fact, to leading order in N, we obtain

$$\langle K_l^{-1} \rangle = \frac{1}{\sigma^{2l}(1-\alpha)^l} K_0^{-1}. \tag{53}$$

However, similar to the mean, the average inverse kernels encode the target outputs only in the correction terms. Equation (53) implies that the mean inverse kernel matrix diverges as $\alpha \to 1$ at zero temperature (Fig. 10). In fact, its trace for all $\alpha > 1$ is proportional to $1/T$ for small T (see Fig. 10 and Ref. [33], Sec. VI). This divergence of the zero-temperature mean inverse kernels in narrow networks is expected, as discussed in Sec. II. Note that because of averaging over weights, the mean kernel, Eq. (50), has a rank of P, even when $N < P$. Furthermore, the divergence of the mean inverse kernels when $\alpha \geq 1$ does not lead to divergence of the mean-squared readout parameters r_l and the renormalization scalars u_l, as observed above (and explained in Appendix A and Ref. [33], Secs. IA and IB).

Concluding this section, we note that even though the second-order weight statistics and the related mean kernel are, to leading order, equal to their GP limit, other statistics of the weights and kernels, and, in particular, the predictor statistics and the generalization error, deviate from the GP limit already to leading order, for all $\alpha = \mathcal{O}(1)$, as shown here and in Sec. III.
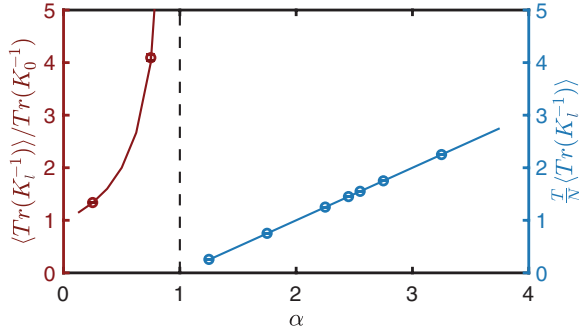
FIG. 10. Trace of the average inverse kernel of a single hidden-layer network. Lines: theory. Circles: simulations. Simulations are done with the same model and parameters as Fig. 6, described in Appendix E. Left: The mean inverse kernel diverges as $\alpha \to 1$ at zero temperature. Right: The trace of the average inverse kernel for the single hidden-layer network multiplied by T, showing the divergence of the trace for all $\alpha > 1$ as $1/T$ as $T \to 0$.

VI. BPKR IN DEEP RELU NETWORKS

A. Approximate BPKR for ReLU networks

Our theory applies to deep networks with linear units, which are limited in their expressive power. To enhance the system's expressivity, one might adopt an architecture comprising a fixed (nonlearned) nonlinear mapping of the input to a shallow layer that then projects to the deep linear networks with learned synapses, as has been studied extensively in recent years (e.g., inputs projecting to a nonlinear kernel representation or to a layer of nonlinear neurons via random weights [36,39,40]). Since our theory does not rely on specific assumptions about the input statistics, our BPKR applies readily to this architecture, with the input vectors and the associated input kernel defined by the nonlinear representation of the shallow layer.

Our theory is not expected to hold for architectures where the learned weights project to nonlinear units, as is the case in most applications of DNNs. In such cases, integration of even one layer of synapses is hard. Here, we ask to what extent our theory can be adapted to nonlinear networks to yield a reasonable approximation in some parameter regimes. For simplicity, we assume a single linear output unit.

We recall that in the GP limit, the properties of DNNs are accounted for by the GP kernels appropriate for the chosen nonlinearity [29]. For example, infinitely wide deep networks with ReLU nonlinearity, which will be studied here, yield GP kernels for the lth layer of the form

$$\langle K_l^{GP}(x,y) \rangle = \frac{\sigma^2}{2\pi}\sqrt{\langle K_{l-1}^{GP}(x,x) \rangle \langle K_{l-1}^{GP}(y,y) \rangle} J(\theta_{l-1}),$$

$$J(\theta_{l-1}) = \sin(\theta_{l-1}) + (\pi - \theta_{l-1})\cos(\theta_{l-1}), \quad (54)$$

$$\theta_{l-1} = \arccos\left(\frac{\langle K_{l-1}^{GP}(x,y) \rangle}{\sqrt{\langle K_{l-1}^{GP}(x,x) \rangle \langle K_{l-1}^{GP}(y,y) \rangle}}\right), \quad (55)$$

where θ_{l-1} represents the angle between the $l-1$ representations of x and y, and the superscript GP in K_l^{GP} specifies the GP kernel of the lth layer, differentiating from the K_l we previously defined in Eq. (4), which represents the dot product of activations of the network. These equations can be solved by iteration from the initial condition $K_0^{GP}(x,y) = K_0(x,y) = (\sigma^2/N_0)x^\top y$, for a pair of input vectors x and y.

The average symbol is the result of (self-)averaging with respect to Gaussian weights. In the GP limit, the predictor statistics of a network with L layers are given in terms of these kernels as

$$\langle f(x) \rangle = \langle k_L^{GP\top}(x) \rangle \langle K_L^{GP} \rangle^{-1} Y, \quad (56)$$

$$\langle [\delta f(x)]^2 \rangle = \langle K_L^{GP}(x,x) \rangle - \langle k_L^{GP\top}(x) \rangle \langle K_L^{GP} \rangle^{-1} \langle k_L^{GP}(x) \rangle, \quad (57)$$

where $\langle k_L^{GP\mu}(x) \rangle = \langle K_L^{GP}(x,x^\mu) \rangle$.

To extend the BPKR to ReLU networks with finite α, we make the ansatz that the weight statistics are modified relative to their GP value by a scalar kernel renormalization u_0. Because in the ReLU nonlinearity K_l is a linear function of the amplitude of K_{l-1}, we reason that the iterative equation has a similar structure to the linear network case, culminating in

$$1 - \sigma^{-2}u_0 = \alpha(1 - u_0^{-L}r_0), \quad (58)$$

$$r_0 = \frac{\sigma^{2L}}{P} Y^\top \langle K_L^{GP} \rangle^{-1} Y, \quad (59)$$

and, consequently, the mean predictor is unchanged from Eq. (56), while the variance is given by

$$\langle [\delta f(x)]^2 \rangle = u_0^L \sigma^{-2L}(\langle K_L^{GP}(x,x) \rangle - \langle k_L^{GP\top}(x) \rangle \langle K_L^{GP} \rangle^{-1} \langle k_L^{GP}(x) \rangle). \quad (60)$$

Note that in the linear case, the Gaussian-averaged $\langle K_L^{GP} \rangle = \sigma^{2L}K_0$, which reduces the above equations to the exact BPKR equations [see Eqs. (20) and (21)]. Also, for $\alpha = 0$, Eq. (58) yields $u_0 = \sigma^2$, and the theory reduces to the GP limit for ReLU networks.

B. Generalization in ReLU networks

Our approximate BPKR predicts that the generalization error increases with α for low σ and decreases for high σ. We have checked these predictions for a ReLU network of a single hidden-layer network trained for the noisy linear
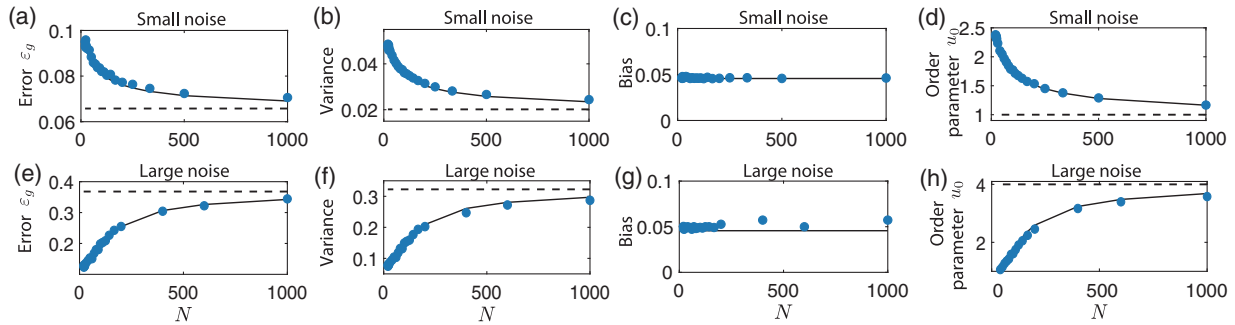
FIG. 11.   Single hidden-layer ($L = 1$) ReLU network trained on the template model with labels generated by a noisy linear teacher with details of parameters in Appendix E. (a,e) Generalization error, (b,f) variance, (c,g) bias of the predictor, and (d,h) the order parameter u_0 as a function of N. Black solid lines: theory. Blue dots: simulation. Black dashed lines: GP limit ($N = \infty$). (a)–(d) Results in the small-noise regime where the generalization error decreases with N. (e)–(h) Results in the large-noise regime where the generalization error increases with N.
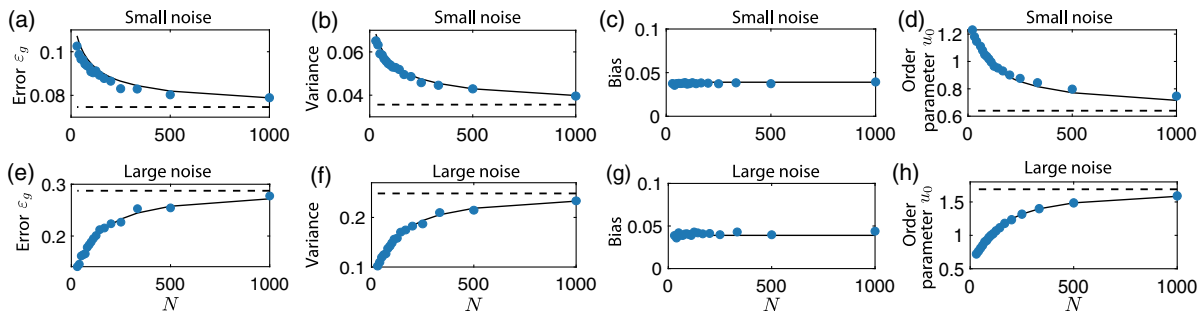
FIG. 12.   Single hidden-layer ($L = 1$) ReLU network trained on MNIST binary classification of two digits (0 and 1) with details of the parameters in Appendix E. (a,e) Generalization error, (b,f) variance, (c,g) squared bias of the predictor, and (d,h) order parameter u_0 as a function of N. Black solid lines: theory. Blue dots: simulation. Black dashed lines: GP limit ($N \to \infty$). (a)–(d) Results in the small-noise regime where the generalization error decreases with N. (e)–(h) Results in the large-noise regime where the generalization error increases with N.

teacher task described in Appendix E. Results are shown in Fig. 11. We have also checked these predictions for a ReLU network of a single hidden layer trained for the MNIST binary classification task (see details in Appendix E) as shown in Fig. 12.

The simulation behaves qualitatively the same as predicted by the theory; ε_g in the ReLU network increases with α for small noise and decreases at high noise. Furthermore, surprisingly, there is also a good quantitative agreement between the simulations and the approximate BPKR for ReLU networks, even for small N (i.e., $\alpha \sim 10$). The mean predictor contributing to the bias component of the generalization error is constant, and its value fits the prediction given by Eq. (56). The predictor variance, as well as ε_g, varies with N, in close agreement with the approximate BPKR prediction. Furthermore, the order parameter u_0 defined by $u_0^L = [(\sigma^{2L} Y^\top \langle K_L^{GP} \rangle^{-1} Y)/(Y^\top \langle K_L^{-1} \rangle Y)]$ varies with N, in close agreement with Eq. (58).

In the examples above, $\alpha_0 < 1$. In the linear network, we found a divergence of the bias and the generalization error at $\alpha_0 = 1$ and vanishing of the predictive variance for $\alpha_0 > 1$. These features are not expected to hold for the

nonlinear network because of the stronger expressivity contributed by the trained nonlinear hidden layer. We look at whether our ansatz also serves as a good approximation in the regime of $\alpha_0 > 1$, where the nonlinearity plays a crucial role in allowing for zero training error. The results are shown in Fig. 13. Surprisingly, even here, results for both the predictor statistics and the order parameter are in good agreement with the theory.

In all previous examples, we did not observe double descent in ε_g because we are in the regime where the network achieves zero training error for all $N \geq 2$ (because our N_0 is sufficiently large). We therefore test our results with small N_0, pushing the network closer to its interpolation threshold, which is roughly when $N \sim \alpha_0$, i.e., $P \sim NN_0$ (i.e., the number of learned parameters equals the number of training data [41]). Indeed, we see significant deviation from the approximate BPKR as N decreases and approaches α_0, which suggests that the scalar renormalization of the kernel becomes inadequate as the network approaches its expressivity capacity. While the simulation shows a double-descent behavior, our theoretical ansatz does not (Fig. 14). The theoretical results agree with the
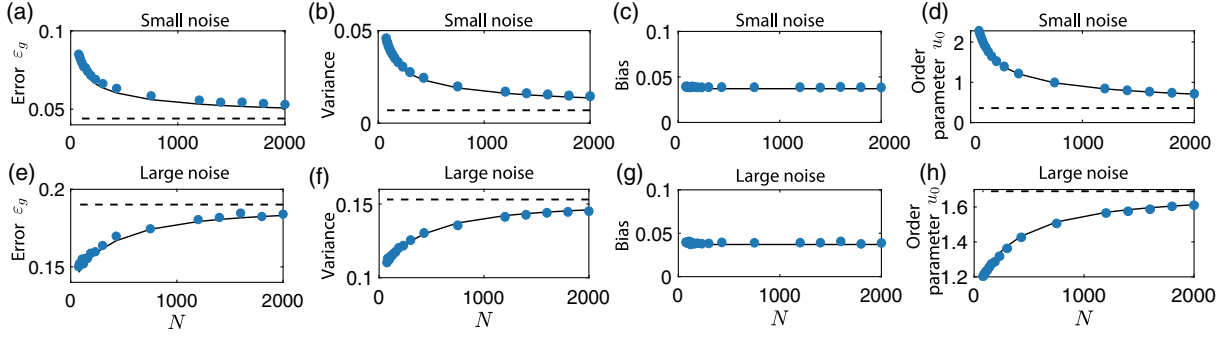
FIG. 13. Single hidden-layer ($L = 1$) ReLU network trained on the template model linear teacher example with details of the parameters in Appendix E, in the $\alpha_0 > 1$ regime. (a,e) Generalization error, (b,f) variance, (c,g) squared bias of the predictor, and (d,h) the order parameter u_0 as a function of N. Black solid lines: theory. Blue dots: simulation. Black dashed lines: GP limit ($N = \infty$). (a)–(d) Results in the small-noise regime where the generalization error decreases with N. (e)–(h) Results in the large-noise regime where the generalization error increases with N.

simulations only on the rhs of the interpolation threshold, i.e., larger N, and they fit the simulations significantly better than the GP approximation, as shown in Figs. 14(b), 14(g), and 14(l). Incidentally, it is interesting to compare the generalization behavior in the three tasks, which differ in their complexity. In the linear teacher task (Appendix E), the *minimum* generalization error is on the lhs of the interpolation threshold. However, for the random labeling task (Appendix E) and classification of MNIST data (Appendix E), because of the nonlinearity of the task, a large number of network parameters are required in order to generalize well, and the minimum generalization error is achieved on the rhs of the interpolation threshold, which is

similar to the linear network (Fig. 6). Importantly, for N below the interpolation threshold while the training error is nonzero, the minimal training error solution is not unique, and this degeneracy in the weights induces variability in the input-output mapping of the network, as shown by the nonvanishing of the predictor variance in the left side of the peak in Fig. 14, except at $N = 1$. This case is different from the linear case, where for $\alpha_0 > 1$ the predictor variance vanishes [see Eq. (21) and Fig. 6].

All the examples above were for a single hidden layer. We also test our ansatz against the simulation results for the ReLU network with multiple hidden layers. As we see in Fig. 15, our approximate BPKR agrees reasonably well
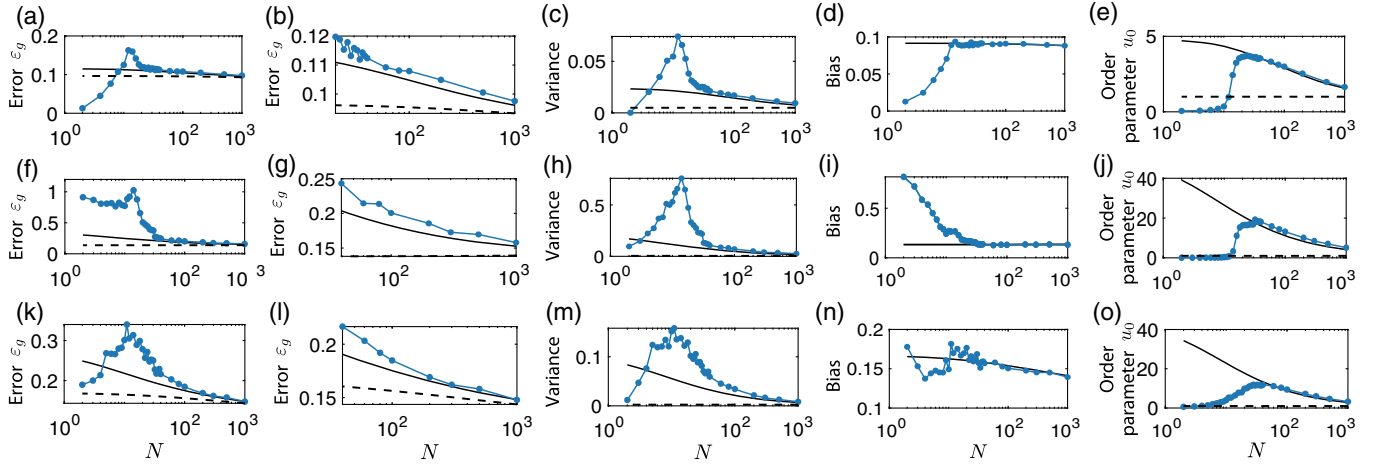

FIG. 14. Single hidden-layer ReLU network with smaller N_0 to push the network closer to its capacity. Black solid lines: theory. Blue lines: simulation. Black dashed lines: GP limit. (a,f,k) Generalization error [(b,g,l) zooming in on the large N part of (a,f,k) to show the convergence to the theory and to compare our ansatz with the GP approximation], (c,h,m) variance, and (d,i,n) bias of the predictor, and (e,j,o) the order parameter u_0. (a)–(e) Trained on the linear teacher task with detailed parameters in Appendix E. The minimum generalization error is observed at small N, on the lhs of the interpolation threshold due to the linearity of the task. (f)–(j) Trained on the template model with random labeling of each cluster (see detailed parameters in Appendix E). The task itself requires nonlinearity, and the minimum generalization error is achieved in the overparametrized regime. (k)–(o) Trained on the randomly projected MNIST data of two digits (see Appendix E). This task also requires nonlinearity, and the minimum generalization error is achieved in the overparametrized regime. [In this figure, we also use the finite T ansatz (see Ref. [33], Sec. VII) for the theory curves.]
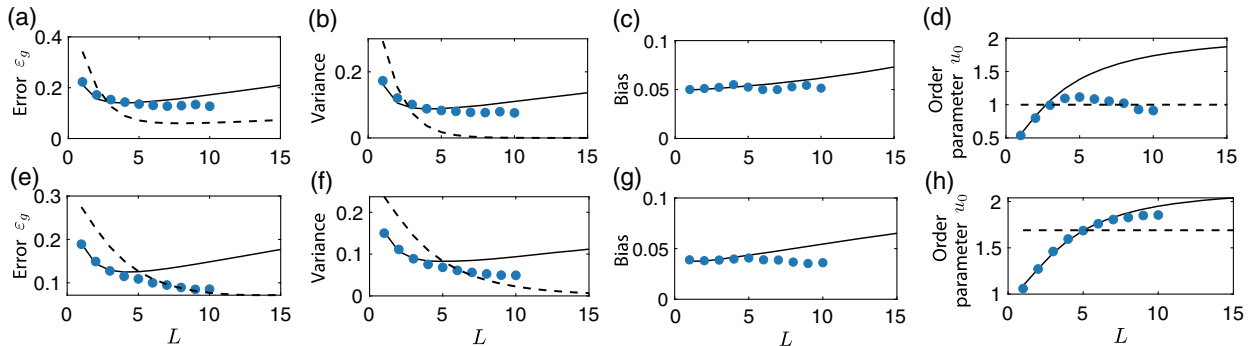
FIG. 15. Generalization error of deep ReLU networks as a function of depth L. Blue dots: simulation. Black solid lines: theoretical approximation. Black dashed lines: GP limit. (a,e) Generalization error, (b,f) variance, and (c,g) bias of the predictor, and (d,h) the order parameter u_0. (a)–(d) Results for the template model with noisy linear teacher labels, with the parameters in Appendix E. (e)–(h) Results for a binary MNIST classification task with parameters in Appendix E.

with the simulation for $L = 1$–5 and is significantly better than the predictions of the GP limit, but the agreement fails for large L. Thus, for finite α, when L becomes larger, renormalization of the kernel just by a scalar becomes inadequate.

VII. DISCUSSION

A. Summary

Since the seminal work of Gardner [42,43], statistical mechanics has served as one of the major theoretical frameworks for understanding the complexity of supervised learning. However, so far, it has focused mostly on shallow architectures and addressed the classical bias-variance trade-off where calculated learning curves displayed improvement of generalization when the number of examples was large compared to the system size [44,45]. Statistical mechanics has also focused on phase transitions, local minima, and spin-glass properties due to the underlying nonlinearity of the learning cost function and the quenched randomness of the training data [26,31]. It is well known that deep learning challenges many of the above intuitions, calling for a new theory of learning in deep architectures. In this work, we have developed a new statistical mechanics framework of learning in a family of networks with deep architectures. To make the theory analytically tractable, we have focused on networks with linear units. Despite their limited expressive power, they do share important aspects of nonlinear deep networks, as is highlighted in our work. Importantly, unlike most previous statistical mechanical theories of learning, which resorted to extremely simplifying assumptions about the input statistics and the target labels, our theory general fully exposed the relation between network properties and task details.

DLNNs have been the focus of several studies. References [46,47] prove the absence of suboptimal local minima under mild conditions, a result that is consistent with our results. A very interesting work [18] studied the

gradient descent dynamics of learning in DLNNs, with results that depended critically on the initial conditions (small random weights) and only became tractable with simplifying assumptions about the data ($XX^\top = I$ and $P > N$). Keeping N and P fixed for most of the simulation and analysis, Ref. [18] addressed the changes in representation during training and across different layers. Under the restricted assumptions, they found that (when there are multiple outputs) the learning dynamics can be decomposed into multiple modes, which evolve independently, qualitatively similar to the multiple modes found in our analysis (see further below). However, they did not address the basic question of the system's performance such as the predictor statistics and the generalization error and its critical dependence on various network parameters. Here, we studied the nature of the Gibbs distribution in the weight space induced by learning with the training mean-squared error as the Hamiltonian. We have focused mainly (but not exclusively) on the properties of the feasible weight space consisting of weight vectors that yield zero training error, which is the case in many real-world applications of DNNs, with the well-known L_2 regularization (with an amplitude parametrized by inverse noise, σ^{-2}).

Because of the highly nonlinear nature of the training Hamiltonian, evaluating the statistical mechanical properties of DLNNs seems intractable. Here, we developed the BPKR method to integrate out the weight matrices layer by layer, allowing us to derive equations for the system's properties that are exact in the thermodynamic limit. Importantly, in contrast to most kernel-based theories of deep networks, our thermodynamic limit is defined by letting both the width N and training size P diverge while the load $\alpha = P/N$ remains of order one, extending the well-known thermodynamic limit of statistical mechanics of learning to deep architectures [16,25]. We have shown that the effect of the finite load is to change the effective Hamiltonians through an α-dependent kernel renormalization at each successive step of weight integration.

In addition to load α, depth L, and weight noise parameter σ^2. Inputs and their labels in the training data affect the properties of the system through the mean-squared input-layer readout parameter, $\sigma^2 r_0$, Eq. (12). Our results yield rich phase diagrams specifying the dependence of the generalization error on the width and the depth of the network, Figs. 3–5. Importantly, depending on σ^2 and $\sigma^2 r_0$, the generalization error may decrease upon increasing width (i.e., decreasing α) and increasing depth (i.e., increasing L). Since this occurs within the overparametrized regime where the training error is zero throughout, our results prove that, in an exactly solvable deep network, increasing network complexity may lead to a substantial improvement in generalization. We were also able to identify the parameter regimes where this improvement happens.

Importantly, the BPKR also enabled us to evaluate the posterior properties of each layer's weights imposed by learning. We leveraged this to explore the effect of input and output data on the layerwise similarity matrices induced through learning and showed that because of different renormalization strengths, amplification of modes in the layer representations is not uniform, as demonstrated in the examples in Figs. 8 and 9. Recent studies have analyzed the similarity matrices of neuronal activities for structured tasks and compared them with the representations in the hidden layers of DNNs [48–50]. Therefore, our work may provide a theoretical understanding of how neuronal representations are constrained by the task structure.

B. BPKR and GD learning

In the case of multiple outputs, we show in Sec. IV A that the layerwise renormalization order parameters are not scalars but matrices. The renormalization order parameter after full averaging is diagonalized by the unitary matrix, which diagonalizes the input-layer readout covariance matrix, Eq. (33). Different eigenvalues corresponding to different modes obey an independent set of equations, analogous to Ref. [18], which shows modes evolving independently with time during GD learning. However, our renormalization modes are defined by diagonalizing input-layer mean-squared readout matrices and not by a fixed input-output covariance matrix as in Ref. [18]. As stated above, our results rely on the equilibrium assumption but not on the special structure of the data nor on an initial condition.

It is interesting to explore the similarity of the behavior of our system with the properties of gradient descent, with implicit regularization induced by early stopping, starting from random initial conditions [16,51,52]. As we show in Sec. VIII of Ref. [33], the generalization properties of the early stopping dynamics may exhibit qualitatively similar features to those predicted by our theory for Gibbs learning, with the initial variance of the weights in the early stopping dynamics playing the role of our noise parameter σ^2. For example, the generalization error of weights learned

through the early stopping dynamics increases with the network width for large initial weight variance and decreases with the network width for small initial weight variance, which qualitatively agrees with the behavior of the generalization error in our BPKR theory in different regimes of the noise parameter σ. Extending our theory to the learning dynamics is an interesting ongoing study.

C. Nonlinear DNNs

We have extended our theory to ReLU networks by applying a scalar kernel renormalization scheme on the GP nonlinear kernels [Sec. VI, Eqs. (56)–(59)]. Testing this approximation against numerical simulations of a few learning tasks with ReLU networks with a moderate number of layers revealed strikingly good qualitative and quantitative agreement regarding the width and noise dependencies of the predictor statistics and the generalization error, as well as the layerwise mean-squared readout order parameters, with much greater accuracy than the GP theory.

Importantly, this BPKR approximate theory for ReLU networks holds, even in cases where, for a linear network, the system would be in a highly underparametrized regime (Fig. 13), such that the neuronal nonlinearity plays a crucial role in the ability of the system to yield zero training and low generalization error (Fig. 15). The failure of the approximation for deeper networks ($L \geq 5$) is expected. The GP theory for nonlinear networks predicts that, as $L \to \infty$, not only does the magnitude of the kernel matrices converge to a (finite or infinite) fixed point, but its matrix structure also converges to a fixed point, implying the loss of information about the structure of the inputs in deep networks with infinite width. Thus, when the width is finite, i.e., $\alpha = \mathcal{O}(1)$, we expect to see a renormalization not only in the kernels' magnitudes (as in our scalar renormalization) but also in their shape. In addition, in nonlinear networks with finite width, the basic description of the system may depend on higher-order statistics than the kernel matrices, as suggested by the recent work of Refs. [53,54].

Even for shallow nonlinear networks, the approximate nonlinear BPKR breaks down in the underparametrized regime, on the left side of the interpolation threshold at $N \approx \alpha_0$. Thus, if α_0 is large, there is a substantial range of small N for which the system is in the underparametrized regime, and this gives rise to a peak in the generalization error (as a function of N) near the interpolation threshold, a genuine "double-descent" phenomena as studied in Refs. [35,36]. Naturally, our approximate theory predicts monotonic dependence in N; hence, it is valid only on the right side of the double-descent peak, i.e., in the overparametrized regime, Fig. 14.

D. Relation to other methods

Successive integration of random variables of joint distributions is used in belief propagation algorithms [15,55–60]. However, despite the Markovian property of

the distribution of the deep network activations, the posterior distribution of the weights takes a complicated form, as described in Sec. II A [Eqs. (3) and (5)], rendering the layerwise weight integration intractable in general, and even in linear networks, it can be performed only in the thermodynamic limit, as shown here. Therefore, although Bayesian inference algorithms such as message passing are commonly applied to study the distribution of hidden-layer activations of Bayesian neural networks [61–63], they are not directly applicable for computing the posterior distribution of the weights. Recent works on inference of the posterior weight distribution have proposed to extend backprop learning algorithms to also update the variances of the weights, by approximating the weight distribution as an independent Gaussian distribution [64,65]. As our work shows, the posterior distribution is far from being IID Gaussian. Importantly, backprop learning algorithms do not necessarily provide insight into the final solutions. In contrast, our work is a theoretical study of the properties of the posterior distribution of weights after learning.

Our BPKR also has some analogy with the renormalization group (RG) approach in physics. Similar to BPKR, RG evaluates properties of high-dimensional systems by successive integration of subsets of the systems' d.o.f. [66]. However, the analogy is limited because in contrast to RG, here there is no obvious notion of coarse graining of d.o.f. Our system combines properties of layered physical systems [67–70] with the mean field aspect arising from the full layer-to-layer connectivity. The latter is demonstrated by the fact that the behavior at the critical point $\sigma^2(1 - \alpha) = 1$ is mean-field-like [see Eq. (30)].

E. Extensions of present work

There are several paths for extending our theory to deeper nonlinear networks. Exact mean-field equations are possible for specific forms of nonlinearities. For a generic nonlinearity, approximate methods might be possible. These methods would likely involve renormalization not only of kernels but also of other terms in the effective Hamiltonian, such as fourth-order kernels [53]. These are topics of ongoing work.

Our theory applies to fully connected networks without additional constraints on the network structure, while in practice, other types of neural networks such as convolutional neural networks (CNNs) are commonly used for image processing, speech recognition, and various tasks. Recent work discussed extension of the GP theory to CNNs [71–73]. Incorporating such architectural restrictions into our theory induces shape renormalization of the kernel (i.e., not simply renormalization by a scalar) and is a topic of ongoing work. Other extensions of our theory include loss functions other than MSE and regularization terms other than L_2.

ACKNOWLEDGMENTS

APPENDIX A: BACKPROPAGATING KERNEL RENORMALIZATION FOR DLNNS

We begin with the partition function

$$Z = \int d\Theta \exp\left[-\frac{\beta}{2N}\sum_{\mu=1}^{P}\left(\sum_{i=1}^{N}a_i\phi_i(x^\mu, W) - y^\mu\right)^2 - \frac{1}{2\sigma^2}\Theta^T\Theta\right] \tag{A1}$$

and introduce P auxiliary integration variables, $t^\mu (\mu = 1, \ldots, P)$, to linearize the quadratic training error,

$$Z = \int d\Theta \int \Pi_\mu^P dt_\mu \exp\left[-\frac{1}{2\sigma^2}\Theta^\top\Theta - \sum_{\mu=1}^{P} it_\mu\left(\frac{1}{\sqrt{N}}\sum_{i=1}^{N}a_i\phi_i(x^\mu, W) - y^\mu\right) - \frac{T}{2}t^\top t\right]. \tag{A2}$$

Integrating over a, we have $Z = \int dW Z_L(W)$ with

$$Z_L(W) = \int dt \exp\left[-\frac{1}{2}t^\top(K_L + TI)t + it^\top Y - \frac{1}{2\sigma^2}\text{Tr}(W^\top W)\right], \tag{A3}$$

where the kernel matrix K_L is defined in Eq. (4) with $l = L$. Integrating over t yields

$$Z_L = \exp\left[-\frac{1}{2}Y^\top(K_L + TI)^{-1}Y - \frac{1}{2}\log\det(K_L + TI) - \frac{1}{2\sigma^2}\mathrm{Tr}(W^\top W)\right]. \tag{A4}$$

To make further progress, we assume all the units are linear, so the hidden units are $x_{i,l} = (1/\sqrt{N})w_l^{i\top}x_{l-1}$ (and the first layer units are $x_{i,1} = (1/\sqrt{N_0})w_1^{i\top}x$). We evaluate $Z = \int dW Z_L(W)$ by successive integrations of weight matrices one at a time, starting from the top layer. Integrating the top hidden-layer weights to compute $Z_{L-1}(W') = \int dW_L Z_L(W) = \exp[-H_{L-1}]$, where the weights W' consist of all weight matrices upstream of W_L, $W' = \{W_k\}_{k<L}$, we obtain

$$Z_{L-1}(W') = \int \Pi_{i=1}^{N_L} dw_L^i \int dt \exp\left[-\frac{1}{2}t^\top(K_L + TI)t - \frac{1}{2\sigma^2}\mathrm{Tr}(W^\top W) + it^\top Y - \frac{1}{2\sigma^2}\mathrm{Tr}(W'^\top W')\right]$$
$$= \int dt \exp\left[it^\top Y + NG(t) - \frac{T}{2}t^\top t - \frac{1}{2\sigma^2}\mathrm{Tr}(W'^\top W')\right], \tag{A5}$$

$$G(t) = \log\left\langle \exp-\frac{1}{2N}t^\top K_w^L t\right\rangle_w, \tag{A6}$$

where the average is with respect to a *single* N-dimensional weight vector w_L^i with IID $\mathcal{N}(0,\sigma)$ components, and $K_w^{L,\mu\nu} = \sigma^2 x_{i,L}^\mu x_{i,L}^\nu = (\sigma^2/N)x_{L-1}^{\mu\top}w_L^i w_L^{i\top}x_{L-1}^\nu$. Performing the average in Eq. (A6) yields $G(t) = -\frac{1}{2}\log(1 + h_{L-1})$, where

$$h_{L-1} = \frac{\sigma^2}{N}t^\top K_{L-1}t. \tag{A7}$$

To integrate over t, we enforce the identity Eq. (A7), by Fourier representation of the delta function, introducing the auxiliary variable u_{L-1},

$$Z_{L-1} = \int du_{L-1}\int_{-1} dh_{L-1}\int dt \exp\left[it^\top Y - \frac{N}{2}\log(1 + h_{L-1}) + \frac{N_L}{2\sigma^2}u_{L-1}h_{L-1}\right.$$
$$\left. - \frac{1}{2}t^\top(u_{L-1}K_{L-1} + TI)t - \frac{1}{2\sigma^2}\mathrm{Tr}(W'^\top W')\right], \tag{A8}$$

and integrating over t,

$$Z_{L-1} = \int du_{L-1}\int_{-1} dh_{L-1}\exp\left[-\frac{N}{2}\log(1 + h_{L-1}) + \frac{1}{2\sigma^2}Nu_{L-1}h_{L-1} - \frac{1}{2}Y^T(u_{L-1}K_{L-1} + TI)^{-1}Y\right.$$
$$\left. - \frac{1}{2}\log\det(u_{L-1}K_{L-1} + TI) - \frac{1}{2\sigma^2}\mathrm{Tr}(W'^\top W')\right]. \tag{A9}$$

In the limit of $N \to \infty$, $P \to \infty$, and fixed α, we solve this integral with the saddle-point method. One of the saddle-point equations yields $u_{L-1} = [(\sigma^2)/(1 + h_{L-1})]$; plugging back in Eq. (A9), we obtain

$$Z_{L-1}(W') = \int du_{L-1}\exp-H_{L-1}(W', u_{L-1}) \tag{A10}$$

with the effective Hamiltonian

$$H_{L-1}(W', u_{L-1}) = \frac{1}{2}Y^\top(u_{L-1}K_{L-1} + TI)^{-1}Y - \frac{N}{2}\log u_{L-1} + \frac{1}{2}\log\det(K_{L-1}u_{L-1} + TI) + \frac{1}{2\sigma^2}\mathrm{Tr}W'^\top W' + \frac{1}{2\sigma^2}Nu_{L-1}. \tag{A11}$$

Thus, integrating over W_L, results in the presence of an auxiliary scalar DOF, u_{L-1}. Finally, we eliminate u_{L-1} through a saddle-point equation,

$$N(1 - \sigma^{-2}u_{L-1}) = -Y^T(u_{L-1}K_{L-1} + TI)^{-2}u_{L-1}K_{L-1}Y + \text{Tr}(u_{L-1}K_{L-1} + TI)^{-1}u_{L-1}K_{L-1}. \tag{A12}$$

At the $T \to 0$ limit, we obtain Eq. (7).

This procedure can be iterated layer by layer. We demonstrate it by computing $H_{L-2}(W')$ defined via $Z_{L-2}(W'') = \int dW_{L-1}Z_{L-1}(W') = \exp[-H_{L-2}(W'')]$, where W'' denotes all weight matrices upstream of W_{L-1},

$$Z_{L-2}(W'') = \int du_{L-1} \int dW_{L-1} \exp\left[\frac{N}{2}\log u_{L-1} - \frac{1}{2\sigma^2}Nu_{L-1} - \frac{1}{2}Y^T(u_{L-1}K_{L-1} + TI)^{-1}Y\right.$$
$$\left. - \frac{1}{2}\log \det(u_{L-1}K_{L-1} + TI) - \frac{1}{2\sigma^2}\text{Tr}(W'^\top W')\right]$$
$$= \int du_{L-1} \int dt \exp\left[it^\top Y + NG(t) - \frac{T}{2}t^\top t - \frac{1}{2\sigma^2}\text{Tr}(W''^\top W'') + \frac{N}{2}\log u_{L-1} - \frac{1}{2\sigma^2}Nu_{L-1}\right], \tag{A13}$$

$$G(t) = \log\left\langle \exp -\frac{u_{L-1}}{2N}t^\top K_w^{L-1}t\right\rangle_w, \tag{A14}$$

where the average is with respect to a single N-dimensional Gaussian vector with IID $\mathcal{N}(0, \sigma)$ components.

Performing this average yields $G(t) = -\log(1 + h_{L-2})$ with

$$h_{L-2} = \frac{\sigma^2 u_{L-1}}{N_{L-1}}t^\top K_{L-2}t. \tag{A15}$$

Similar to above, we introduce two additional scalar integration variables u_{L-2} and h_{L-2},

$$Z_{L-2} = \int du_{L-2} \int_{-1} dh_{L-2} \int du_{L-1} \int dt \exp\left[it^\top Y - \frac{N}{2}\log(1 + h_{L-2}) + \frac{N}{2\sigma^2}u_{L-2}h_{L-2}\right.$$
$$\left. - \frac{1}{2}t^\top(u_{L-1}u_{L-2}K_{L-1} + TI)t + \frac{N}{2}\log u_{L-1} - \frac{1}{2\sigma^2}Nu_{L-1} - \frac{1}{2\sigma^2}\text{Tr}(W''^\top W'')\right]. \tag{A16}$$

Integrating over t and plugging in the saddle point of h_{L-2} [$u_{L-2} = (\sigma^2)/(h_{L-2} + 1)$], we have the effective Hamiltonian

$$H_{L-2}(W'', u_{L-1}, u_{L-2})$$
$$= -\frac{N}{2}\log u_{L-1} + \frac{1}{2\sigma^2}\text{Tr}W''^\top W'' + \frac{1}{2\sigma^2}Nu_{L-1}$$
$$+ \frac{1}{2}Y^\top(u_{L-1}u_{L-2}K_{L-2} + TI)^{-1}Y - \frac{N}{2}\log u_{L-2}$$
$$+ \frac{1}{2}\log \det(K_{L-2}u_{L-1}u_{L-2} + TI) + \frac{1}{2\sigma^2}Nu_{L-2}. \tag{A17}$$

Finally, u_{L-1} and u_{L-2} are computed via saddle-point equations

$$N(1 - u_{L-1}\sigma^{-2})$$
$$= -u_{L-1}u_{L-2}Y^T(u_{L-1}u_{L-2}K_{L-2} + TI)^{-2}K_{L-2}Y$$
$$+ u_{L-1}u_{L-2}\text{Tr}(u_{L-1}u_{L-2}K_{L-2} + TI)^{-1}K_{L-2} \tag{A18}$$

$$N(1 - u_{L-2}\sigma^{-2})$$
$$= -u_{L-1}u_{L-2}Y^T(u_{L-1}u_{L-2}K_{L-2} + TI)^{-2}K_{L-2}Y$$
$$+ u_{L-1}u_{L-2}\text{Tr}(u_{L-1}u_{L-2}K_{L-2} + TI)^{-1}K_{L-2}. \tag{A19}$$

The solution obeys $u_{L-1} = u_{L-2}$, and we now have

$$Z = \int dW'' \int du_{L-2} \exp -H_{L-2}(W'', u_{L-2}) \tag{A20}$$

with

$$H_{L-2}(W'', u_{L-2}) = -N\log u_{L-2} + \frac{1}{2\sigma^2}\text{Tr}W''^\top W''$$
$$+ \frac{1}{\sigma^2}Nu_{L-2} + \frac{1}{2}Y^\top(u_{L-2}^2K_{L-2} + TI)^{-1}Y$$
$$+ \frac{1}{2}\log \det(K_{L-1}u_{L-2}^2 + TI). \tag{A21}$$

Evaluating u_{L-2} via the saddle-point equation yields

$$N(1 - u_{L-2}\sigma^{-2}) = -Y^T(u_{L-2}^2 K_{L-2} + TI)^{-2} u_{L-2}^2 K_{L-2} Y$$
$$+ \mathrm{Tr}[(u_{L-2}^2 K_{L-2} + TI)^{-1} u_{L-2}^2 K_{L-2}],$$

$$\text{(A22)}$$

where the kernel K_{L-2} is renormalized by u_{L-2}^2. Note that in the integration of W_{L-2}, both u_{L-1} and u_{L-2} are auxiliary integration variables (hence, independent of weights) and are determined at the last step by the new saddle-point equation (A22) as functions of W''. In contrast, the saddle-point value of u_{L-1} in the first renormalization step, Eq. (A12), is a function of W'. In fact, the average of u_{L-1} of the first renormalization step over W_{L-1} obeys $u_{L-1} = u_{L-2}$ of the second renormalization step (see paragraph below on renormalization of order parameters).

Similarly, iterating this renormalization l times yields Eqs. (45) and (46).

1. Narrow network at zero temperature

At finite temperature, the above derivation holds for all α. However, in the zero-temperature limit, we need to address the singularity of the hidden layers' kernel matrices when $\alpha > 1$. We begin with the partition function after integrating the readout layer at zero temperature,

$$Z_L(W) = \int dt \exp\left[-\frac{1}{2} t^\top K_L t + it^\top Y - \frac{1}{2\sigma^2} \mathrm{Tr}(W^\top W)\right].$$

$$\text{(A23)}$$

With eigenvalue decomposition of K_L, $K_L = V\Sigma V^\top$, where V is a unitary $P \times P$ matrix, and Σ is a $P \times P$ diagonal matrix with elements $(\Sigma_1, \ldots, \Sigma_N, 0, \ldots, 0)$, and orthogonal transformation of variables $V^\top t \to t$, we have

$$Z_L(W) = \int dt \exp\left[-\frac{1}{2} t^\top \Sigma t + it^\top V^\top Y - \frac{1}{2\sigma^2} \mathrm{Tr}(W^\top W)\right].$$

$$\text{(A24)}$$

We introduce notations $t_\| = [t_1, \cdots, t_N]^\top \in \mathbb{R}^N$, $t_\perp = [t_{N+1}, \ldots, t_P]^\top \in \mathbb{R}^{N-P}$, $V_\| = [V_1, \ldots, V_N] \in \mathbb{R}^{P \times N}$, $V_\perp = [V_{N+1}, \ldots, V_P] \in \mathbb{R}^{P \times (P-N)}$, and $\Sigma_\| = \mathrm{diag}(\Sigma_1, \ldots, \Sigma_N) \in \mathbb{R}^{N \times N}$. With these notations, we can write

$$Z_L(W) = \int dt_\| \int dt_\perp \exp\left[-\frac{1}{2} t_\|^\top \Sigma_\| t_\| + it_\|^\top V_\|^\top Y\right.$$
$$\left. + it_\perp^\top V_\perp^\top Y - \frac{1}{2\sigma^2} \mathrm{Tr}(W^\top W)\right].$$

$$\text{(A25)}$$

Integrating over $t_\perp$ yields $\delta(V_\perp^\top Y)$. The δ function forces the projection of Y onto the directions perpendicular to X_L to vanish. In the zero-temperature limit, this constraint on the weights ensures zero training error; therefore, Y must lie in the subspace spanned by X_L. We next integrate $t_\|$ and obtain $Z_L(W) = \delta(V_\perp Y) \exp[-H_L(W)]$, with

$$H_L(W) = \frac{1}{2} Y^\top K_L^+ Y + \frac{1}{2} \log \det(C_L) + \frac{1}{2\sigma^2} \mathrm{Tr}(W^\top W),$$

$$\text{(A26)}$$

where $K_L^+ = V_\| \Sigma_\|^{-1} V_\|^\top$ is the pseudo-inverse of K_L, and $C_L = (\sigma^2/N) X_L X_L^\top$ has the same determinant as $\Sigma_\|$.

Similarly, we have $Z_{L-l}(W) = \delta(V_\perp Y) \exp[-H_{L-l}(W)]$; here, $V_\perp$ are the eigenvectors of K_{L-l} spanning its null space, and

$$H_{L-l}(W') = \frac{1}{2u_{L-l}^l} Y^\top K_{L-l}^+ Y + \frac{1}{2} \log \det(u_{L-l}^{-l} C_{L-l})$$
$$+ \frac{1}{2\sigma^2} \mathrm{Tr}(W'^\top W').$$

$$\text{(A27)}$$

Differentiating Eq. (A27) with respect to u_{L-l}, we obtain Eqs. (14) and (15).

2. Properties of the order parameters

Here, we show that the order parameters u_l undergo a trivial renormalization upon averaging. For any function of u_l, we can write

$$\langle f(u_l) \rangle_l = \frac{1}{Z_{l-1}} \int du_l f(u_l) \int dW_l \int dt \exp\left[it^\top Y - \frac{1}{2} t^\top (u_l^{L-l} K_l + TI)t + \frac{N(L-l)}{2} \log u_l\right.$$
$$\left. - \frac{N(L-l)}{2\sigma^2} u_l + \frac{1}{2\sigma^2} \mathrm{Tr}(W'^\top W')\right]$$
$$= \frac{1}{Z_{l-1}} \int du_l \int du_{l-1} \int dt f(u_l) \exp\left[it^\top Y - \frac{1}{2} t^\top (u_{l-1} u_l^{L-l} K_{l-1} + TI)t + \frac{N(L-l)}{2} \log u_l\right.$$
$$\left. - \frac{N(L-l)}{2\sigma^2} u_l + \frac{1}{2\sigma^2} \mathrm{Tr}(W''^\top W'') + \frac{N}{2} \log u_{l-1} - \frac{N}{2\sigma^2} u_{l-1}\right],$$

$$\text{(A28)}$$

which is equal to the saddle-point value of $f(u_l)$. Since $u_l = u_{l-1}$ at the saddle point, where u_{l-1} obeys the saddle point Eq. (46) appropriate for $L - l + 1$ iterations, we have

$$\langle f(u_l)\rangle_l = f(u_{l-1}), \tag{A29}$$

which holds for all $0 \le \alpha < \infty$ and all T.

The order parameters u_l have a simple interpretation given by Eq. (13) for $1 \le l \le L$ at zero temperature for $\alpha < 1$ (see Ref. [33], Sec. IVA for the derivation at finite temperature). We evaluate

$$\frac{1}{P}\langle Y^\top K_l^{-1} Y\rangle_l = \frac{1}{Z_{l-1}} \int dW_l \frac{1}{P} Y^\top K_l^{-1} Y \int dt \exp\left[it^\top Y - \frac{1}{2}t^\top u_l^{L-l} K_l t + \frac{1}{2\sigma^2}\mathrm{Tr}(W'^\top W')\right]$$

$$= -\frac{1}{Z_{l-1}} \int dW_l \frac{1}{P} Y^\top \int dt\, it u_l^{L-l} \exp\left[it^\top Y - \frac{1}{2}t^\top u_l^{L-l} K_l t + \frac{1}{2\sigma^2}\mathrm{Tr}(W'^\top W')\right]. \tag{A30}$$

Performing integration over W_l with the same approach we used to compute the partition function Z_{l-1} above, and introducing the same order parameter u_{l-1}, we reduce the above expression to

$$-\frac{1}{Z_{l-1}} \int du_{l-1} \frac{1}{P} Y^\top \int dt\, it u_{l-1}^{L-l}$$

$$\exp\left[it^\top Y + \frac{N(L-l+1)}{2}\log u_{l-1}\right.$$

$$\left. -\frac{1}{2}t^\top u_{l-1}^{L-l+1} K_{l-1} t - \frac{N(L-l+1)}{2\sigma^2} u_{l-1}\right]$$

$$= \frac{1}{P} u_{l-1}^{-1} Y^\top K_{l-1}^{-1} Y. \tag{A31}$$

At zero temperature for $\alpha < 1$, the expression leads to Eq. (13) for all l ($1 \le l \le L$). Note that this result is conditioned on the upstream weights $\{W_k\}_{k<l}$.

For $\alpha > 1$ at zero temperature, the OP obeys Eq. (13) for $1 \le l < L$ (partial averaging of the weights), but the relation is replaced by Eq. (16) for averaging over all hidden weights. The details of these results are delegated to Sec. IA of Ref. [33].

From Eq. (A2), it follows that the W-dependent average of the readout weights is

$$\langle a\rangle = -\frac{\sigma^2}{\sqrt{N}} i\Phi^\top \langle t\rangle. \tag{A32}$$

The statistics of t can be obtained from Eq. (A3),

$$\langle t\rangle = i(K_L + TI)^{-1} Y. \tag{A33}$$

Therefore, we have $\langle a\rangle = (\sigma^2/\sqrt{N})\Phi^\top(K_L + TI)^{-1} Y$, and $\langle a\rangle^\top \langle a\rangle = \sigma^2 P r_L = \sigma^2 Y^\top (K_L + TI)^{-2} K_L Y$. In the zero-temperature limit, for $\alpha < 1$, $\langle a\rangle^\top \langle a\rangle = \sigma^2 Y^\top K_L^{-1} Y$, and for $\alpha > 1$, $\langle a\rangle^\top \langle a\rangle = \sigma^2 Y^\top K_L^+ Y$.

Similarly, we can define a_l as the readout weight vector trained with inputs from the lth layer of the trained network to produce the target output Y, and we can obtain the statistics of a_l by simply replacing the K_L in the above equations with K_l. At zero temperature, we have $\langle a_l\rangle^\top \langle a_l\rangle = \sigma^2 Y^\top K_l^{-1} Y$ for $\alpha < 1$, and

$\langle a_l\rangle^\top \langle a_l\rangle = \sigma^2 Y K_l^+ Y$ for $\alpha > 1$. In Eqs. (8) and (15), the definition of r_l is equivalent to $r_l = (\sigma^{-2}/P)\langle a_l\rangle^\top \langle a_l\rangle$; therefore, we name r_l as the mean-squared layer readout.

The second-order statistics of a_l, including its variance and its norm, are discussed further in Ref. [33], Sec. IB.

Earlier in this section, we introduced the detailed derivation of the self-consistent equation for the order parameter at finite temperature, given by Eq. (46). At the $L \to \infty$ limit, in the low-noise regime $\sigma^2(1-\alpha) < 1$, u_0 approaches 1. We assume that u_0 goes to 1 as $u_0 \approx 1 - (v_0/L)$, as we discussed in Sec. III for zero temperature. Plugging in Eq. (46), we have

$$1 - \sigma^{-2} = -\frac{1}{N} Y^\top (\exp(-v_0) K_0 + TI)^{-2} \exp(-v_0) K_0 Y \tag{A34}$$

$$+ \frac{1}{N}\mathrm{Tr}((\exp(-v_0) K_0 + TI)^{-1} \exp(-v_0) K_0). \tag{A35}$$

This self-consistent equation determines $\exp(v_0)$, which is the limit of λ as $L \to \infty$, as we present in Figs. 7(b) and 7(c) in Sec. IV B.

APPENDIX B: GENERALIZATION

The mean-squared generalization error depends only on the mean and variance of the predictor, and they can be computed using the following generating function:

$$Z(t_{P+1}) = \int D\Theta \exp\left[-\frac{\beta}{2}\sum_{\mu=1}^{P}\left(\frac{1}{\sqrt{N}}\sum_{i=1}^{N} a_i \phi_i(x^\mu, W) - y^\mu\right)^2\right.$$

$$\left. + it_{P+1}\frac{1}{\sqrt{N}}\sum_{i=1}^{N} a_i \phi(W, x) - \frac{T}{2\sigma^2}\Theta^\top\Theta\right], \tag{B1}$$

where x is an arbitrary new point. The statistics of the predictor are given by

$$\langle f(x)\rangle = \partial_{it_{P+1}} \log Z|_{t_{P+1}=0}, \tag{B2}$$

$$\langle \delta^2 f(x)\rangle = \partial^2_{it_{P+1}} \log Z|_{t_{P+1}=0}. \tag{B3}$$

The integral can be performed similarly as in Appendix A by introducing P auxiliary integration variables, $t^\mu (\mu = 1, ..., P)$, integrating over W, and introducing order parameters u_l layer by layer.

After integrating the weights of the entire network, we obtain

$$Z(t_{p+1}) = \int du_0 \exp\left[\frac{NL}{2}\log u_0 - \frac{NL}{2\sigma^2}u_0 + \frac{1}{2}[iY + t_{P+1}^\top u_0^L k_0(x)]^\top (u_0^L K_0 + TI)^{-1}[iY + t_{P+1}^\top u_0^L k_0(x)]\right.$$

$$\left. - \frac{1}{2}\log\det(u_0^L K_0 + TI) - \frac{1}{2}t_{P+1}^\top u_0^L K_0(x,x)t_{P+1}\right], \tag{B4}$$

where $T_0 = u_0^{-L}T$, as defined in Sec. IV B. Differentiating Z, we obtain

$$\langle f(x)\rangle = \partial_{it_{p+1}}\log Z|_{t_{p+1}=0} = u_0^L k_0^\top(x)(u_0^L K_0 + TI)^{-1}Y. \tag{B5}$$

Because the derivative is evaluated at $t_{P+1} = 0$, the saddle point u_0 satisfies the same equation as Eq. (46) for $l = L$. Similarly, we calculate the second-order statistics

$$\langle\delta^2 f(x)\rangle = \partial_{it_{p+1}}^2 \log Z|_{t_{p+1}=0}$$
$$= u_0^L K_0(x,x) - u_0^L k_0^\top(x)(u_0^L K_0 + TI)^{-1}u_0^L k_0(x). \tag{B6}$$

Taking the $T \to 0$ limit, we obtain Eqs. (20) and (21).

The dependence of the generalization error with respect to σ, N, and L is determined by the behavior of $\sigma^2 u_0$ with respect to σ, N, and L, which is shown in Secs. IIA–IIC of Ref. [33]. The dependence on P (which affects both α and α_0) hinges on the specific statistics of input and output. Here, we analyze the relatively simple case of input sampled from IID Gaussian distribution and target output generated by a linear teacher with additive noise, and we focus on the behavior near $\alpha_0 = 1$.

For $P < N_0$, we first consider r_0, averaged over the linear teacher noise. Near $\alpha_0 = 1$, since $\text{Tr}K_0^{-1}$ diverges as $(1 - \alpha_0)^{-1}$ (see Ref. [74]), r_0 is dominated by the contribution from the noise term in the target noisy teacher output Y, and it yields $r_0 \sim \sigma_0^2(1 - \alpha_0)^{-1}$, where σ_0 denotes the amplitude of the teacher's noise.

Since r_0 is divergent as $\alpha_0 \to 1$, keeping the dominant terms in Eq. (11), we obtain $u_0 \sim (\sigma^2 \alpha r_0)^{1/L+1} \sim \alpha^{1/L+1}(1 - \alpha_0)^{-1/L+1}$; thus, u_0^L diverges as $\alpha^{L/L+1}(1 - \alpha_0)^{-L/L+1}$.

The contribution of the squared mean predictor $\langle f(x)\rangle^2$ to ϵ_g averaged over the test sample x and the linear teacher noise is given by the corresponding averages of $Y^\top K_0^{-1}k_0 k_0^\top K_0^{-1}Y$. Assuming $\langle xx^\top\rangle = \gamma I$, then the divergent contribution, similar to r_0, comes from the noise in the linear teacher and is given by $\gamma\sigma_0^2\alpha_0(1 - \alpha_0)^{-1}$.

Since $\text{Tr}(\sigma^2 N_0^{-1}XK_0^{-1}X^\top)$ scales with the rank of K_0 and grows as α_0, the $K_0(x,x) - k_0^\top K_0^{-1}k_0$ term in the variance of

the predictor vanishes as $1 - \alpha_0$, and thus, the variance vanishes as $\alpha^{L/L+1}(1 - \alpha_0)^{1/L+1}$, as $\alpha_0 \to 1$. Therefore, the generalization error is dominated by the divergent bias as $\alpha_0 \to 1$, and it diverges as $\alpha_0/(1 - \alpha_0)$.

For $P > N_0$, because now the network cannot achieve zero training error, we replace the Y in r_0 with $X^\top(XX^\top)^{-1}XY$, which is the output the network actually learns on the training data. Near $\alpha_0 = 1$, since $(1/P)\text{Tr}((XX^\top)^{-1})$ diverges as $\alpha_0^{-1}(\alpha_0 - 1)^{-1}$ [74], r_0 is also dominated by the contribution from the noise term in Y, and it is given by $\sigma_0^2\alpha_0^{-1}(\alpha_0 - 1)^{-1}$. Similarly, we obtain $u_0 \sim (\sigma^2[N_0/N]r_0)^{1/L+1} \sim [\alpha_0(\alpha_0 - 1)]^{-1/L+1}$ and $\langle\langle f(x)\rangle^2\rangle \sim \gamma\sigma_0^2(\alpha_0 - 1)^{-1}$. The generalization error is dominated by the divergent bias as $\alpha_0 \to 1$, and it diverges as $(\alpha_0 - 1)^{-1}$.

The case of clustered inputs, as in our template model, is treated analytically in Sec. IID of Ref. [33].

APPENDIX C: MULTIPLE OUTPUTS

1. BPKR for multiple outputs

Here, we extend the calculations in Appendix A to multiple outputs ($m > 1$) in the zero-temperature limit for $\alpha < 1$. For $m > 1$, we introduce the integration variables t from a $P \times m$ matrix; hence,

$$Z_{L-1} = \int \Pi_{i=1}^{N_L}dw_L^i \int dt \exp\left[-\frac{1}{2}\text{Tr}(t^\top K_L t)\right.$$

$$\left. + i\text{Tr}(t^\top Y) - \frac{1}{2\sigma^2}\text{Tr}(W^\top W)\right]$$

$$= \int dt \exp\left[i\text{Tr}(t^\top Y) + NG(t) - \frac{1}{2\sigma^2}\text{Tr}(W'^\top W')\right] \tag{C1}$$

$$G(t) = \log\left\langle\exp - \frac{1}{2N}\text{Tr}(t^\top K_w t)\right\rangle_w. \tag{C2}$$

Integrating over w yields $G(t) = -\frac{1}{2}\log\det(I + \mathcal{H}_{L-1})$, where the $m \times m$ dim matrix is $\mathcal{H}_{L-1} = (\sigma^2/N)t^\top K_{L-1}t$, a relation that is enforced by an auxiliary matrix variable $\mathcal{U}_{L-1}$. With $\hat{t} = K_{L-1}^{1/2}t$, we have

$$Z_{L-1} = \int d\mathcal{U}_{L-1} \int d\mathcal{H}_{L-1} \int dt \exp\left[i\mathrm{Tr}(t^\top Y) - \frac{N}{2}\log\det(I + \mathcal{H}_{L-1}) + \frac{N}{2\sigma^2}\mathrm{Tr}(\mathcal{U}_{L-1}\mathcal{H}_{L-1})\right.$$
$$\left. - \frac{1}{2}\mathrm{Tr}(\mathcal{U}_{L-1}t^\top K_{L-1}t) - \frac{1}{2\sigma^2}\mathrm{Tr}(W'^\top W')\right]$$
$$= \int d\mathcal{U}_{L-1} \int d\mathcal{H}_{L-1} \int d\hat{t} \exp\left[i\hat{t}^\top K_{L-1}^{-1/2}Y - \frac{N}{2}\log\det(I + \mathcal{H}_{L-1}) + \frac{N}{2\sigma^2}\mathrm{Tr}(\mathcal{U}_{L-1}\mathcal{H}_{L-1})\right.$$
$$\left. - \frac{m}{2}\log\det K_{L-1} - \frac{1}{2}\mathrm{Tr}(\hat{t}\mathcal{U}_{L-1}\hat{t}^\top) - \frac{1}{2\sigma^2}\mathrm{Tr}(W'^\top W')\right]. \tag{C3}$$

For $\alpha < 1$, we can integrate over $\hat{t}$, yielding

$$Z_{L-1} = \int d\mathcal{U}_{L-1} \int d\mathcal{H}_{L-1} \exp\left[-\frac{N}{2}\log\det(I + \mathcal{H}_{L-1})\right.$$
$$+ \frac{N}{2\sigma^2}\mathrm{Tr}(\mathcal{U}_{L-1}\mathcal{H}_{L-1}) - \frac{1}{2}\mathrm{Tr}(\mathcal{U}_{L-1}^{-1}Y^\top K_{L-1}^{-1}Y)$$
$$- \frac{m}{2}\log\det K_{L-1} - \frac{P}{2}\log\det(\mathcal{U}_{L-1})$$
$$\left. - \frac{1}{2\sigma^2}\mathrm{Tr}(W'^\top W')\right]. \tag{C4}$$

Again substituting the saddle point of $\mathcal{H}_{L-1}$, i.e., $I + \mathcal{H}_{L-1} = \sigma^2\mathcal{U}_{L-1}^{-1}$, yields

$$Z_{L-1} = \int d\mathcal{U}_{L-1} \exp\left[\frac{N}{2}\log\det\mathcal{U}_{L-1} - \frac{N}{2\sigma^2}\mathrm{Tr}(\mathcal{U}_{L-1})\right.$$
$$- \frac{1}{2}\mathrm{Tr}(\mathcal{U}_{L-1}^{-1}Y^\top K_{L-1}^{-1}Y) - \frac{m}{2}\log\det K_{L-1}$$
$$\left. - \frac{P}{2}\log\det(\mathcal{U}_{L-1}) - \frac{1}{2\sigma^2}\mathrm{Tr}(W'^\top W')\right]. \tag{C5}$$

Differentiating with respect to $\mathcal{U}_{L-1}$, we obtain the self-consistent equation for $\mathcal{U}_{L-1}$,

$$I - \sigma^{-2}\mathcal{U}_{L-1} = \alpha\left(I - \frac{1}{P}Y^\top K_{L-1}^{-1}Y\mathcal{U}_{L-1}^{-1}\right). \tag{C6}$$

A similar conclusion can be extended to the following integration steps, and we have

$$I - \sigma^{-2}\mathcal{U}_{L-l} = \alpha\left(I - \frac{1}{P}Y^\top K_{L-l}^{-1}Y\mathcal{U}_{L-l}^{-l}\right). \tag{C7}$$

From these equations, it follows that for all l, $\mathcal{U}_{L-l}$ can be diagonalized with the eigenvectors of the mean-squared readout matrix. Writing the eigenvalue matrix of the readout matrix as $\mathrm{diag}(r_{1,L-l}, \ldots, r_{k,L-l}, \ldots, r_{m,L-l}) = V_{L-l}^\top([1/P]Y^\top K_{L-l}^{-1}Y)V_{L-l}$, the renormalization eigenvalue matrix can be written as $\mathrm{diag}(u_{1,L-l}, \ldots, u_{k,L-l}, \ldots,$

$u_{m,L-l}) = V_{L-l}^\top \mathcal{U}_{L-l}V_{L-l}$, and Eq. (C7) can be reduced to independent equations for the eigenvalues $u_{k,L-l 1 \leq k \leq m}$, as given by Eq. (34). However, Eq. (C7) holds only for $\alpha < 1$; because of the singularity of K_{L-l} for $l < L$ at $\alpha > 1$, the equation for the eigenvalues of $\mathcal{U}_{L-l}$ is replaced by Eq. (14) for narrow networks. (See Sec. IIIB of Ref. [33] for details.)

For wide networks, we can calculate the statistics of $Y^\top K_l^{-1}Y$, with an approach similar to that for the single-output case, by relating it to the average of t (see Sec. IIIA of Ref. [33] for details), obtaining Eq. (37). For narrow networks, we calculate the statistics of $Y^\top K_l^+Y$; for the same reason as in the single-output calculations, we need to relate the quantity to the second-order moment of t; the procedure is also similar to the single-output case in Appendix A, and we obtain Eqs. (37) and (41) (see details in Sec. IIIC of Ref. [33]).

Iterating the integration steps until all weights are integrated, we obtain the equation for the eigenvalues of $\mathcal{U}_0$,

$$I - \sigma^{-2}u_{k0} = \alpha(1 - u_{k0}^{-L}r_{k0}), \tag{C8}$$

where $\mathcal{U}_0 = V_0\mathrm{diag}(u_{10}, \ldots, u_{k0}, \ldots, u_{m0})V_0^\top$, V_0 is defined through the input readout covariance matrix, $(1/P)Y^\top K_0^{-1}Y = V_0\mathrm{diag}(r_{10}, \ldots, r_{k0}, \ldots, r_{m0})V_0^\top$, proving Eq. (36).

Equation (C8) holds for all α as long as $\alpha_0 < 1$. We also note that a straightforward generalization of Eq. (A29) leads to

$$\langle f(\mathcal{U}_l)\rangle_l = f(\mathcal{U}_{l-1}), \tag{C9}$$

which also holds for all $0 \leq \alpha < \infty$.

2. Generalization for multiple outputs

The generalization error and related quantities can be calculated similarly as in Appendix B, replacing the scalar order parameter with a matrix,

$$Z_{L-l}(t_{P+1}) = \int d\mathcal{U}_{L-l} \int dt \exp\left[i\mathrm{Tr}(t^\top Y) + \frac{(N-P)l}{2}\log\det\mathcal{U}_{L-l} - \frac{Nl}{2\sigma^2}\mathrm{Tr}(\mathcal{U}_{L-l}) - \frac{1}{2}\mathrm{Tr}(\mathcal{U}_{L-l}^l t^\top K_{L-l}t)\right.$$

$$\left. - \frac{1}{2\sigma^2}\mathrm{Tr}(W'^\top W') + \mathrm{Tr}(\mathcal{U}_{L-l}^l t_{P+1}^\top k_{L-l}^\top t) - \frac{1}{2}\mathrm{Tr}(\mathcal{U}_{L-l}^l t_{P+1}^\top K_{L-l}(x,x)t_{P+1})\right]. \tag{C10}$$

Integrating over t and taking $l = L$, we obtain

$$Z(t_{p+1}) = \int d\mathcal{U}_0 \exp\left[\frac{(N-P)L}{2}\log\det\mathcal{U}_0 - \frac{NL}{2\sigma^2}\mathrm{Tr}(\mathcal{U}_0) - \frac{1}{2}\mathrm{Tr}(\mathcal{U}_0^L t_{P+1}^\top K_0(x,x)t_{P+1})\right.$$

$$\left. + \frac{1}{2}\mathrm{Tr}[(iK_0^{-1/2}Y + K_0^{-1/2}k_0(x)t_{P+1}\mathcal{U}_0^L)\mathcal{U}_0^{-L}(iK_0^{-1/2}Y + K_0^{-1/2}k_0(x)t_{P+1}\mathcal{U}_0^L)^\top]\right]. \tag{C11}$$

Taking the derivative with respect to t_{P+1},

$$\langle f(x)\rangle = \partial_{it_{p+1}}\log Z|_{t_{p+1}=0} = \frac{\partial\mathrm{Tr}(it_{P+1}^\top k_0^\top(x)K_0^{-1}Y)}{\partial it_{P+1}} = k_0^\top(x)K_0^{-1}Y, \tag{C12}$$

$$\langle\delta f_i(x)\delta f_j(x)\rangle = \partial_{it_{P+1}^i}\partial_{it_{P+1}^j}\log Z|_{t_{p+1}=0} = \frac{\frac{1}{2}\partial\mathrm{Tr}(U_0^L t_{P+1}^\top t_{P+1})}{\partial_{t_{P+1}^i}\partial_{t_{P+1}^j}}[K_0(x,x) - k_0^\top(x)K_0^{-1}k_0(x)]$$

$$= \mathcal{U}_{0i,j}^L[K_0(x,x) - k_0^\top(x)K_0^{-1}k_0(x)], \tag{C13}$$

we obtain the predictor statistics as described in Sec. IV A.

3. Multiple outputs at finite temperature

In Sec. IV A, we focused on results for multiple outputs at zero temperature. Here, we introduce the results for multiple outputs at finite T (see Sec. IV B of Ref. [33] for detailed derivations). The partition function after integrating over l layers is given by

$$Z_{L-l} = \int d\mathcal{U}_{L-l}\exp\left[\frac{Nl}{2}\log\det\mathcal{U}_{L-l} - \frac{Nl}{2\sigma^2}\mathrm{Tr}(\mathcal{U}_{L-l}) - \frac{1}{2}\hat{Y}^\top(\mathcal{U}_{L-l}^l \otimes K_{L-l} + TI)^{-1}\hat{Y}\right.$$

$$\left. - \frac{1}{2}\log\det(\mathcal{U}_{L-l}^l \otimes K_{L-l} + TI) - \frac{1}{2\sigma^2}\mathrm{Tr}(W'^\top W')\right], \tag{C14}$$

where $\hat{Y}$ is an mP-dimensional vector that denotes the vectorized Y. The corresponding saddle-point equation for $\mathcal{U}_{L-l}$ for $1 \le l \le L$ is given as

$$I - \sigma^{-2}\mathcal{U}_{L-l} = \frac{1}{N}\mathrm{Tr}_P[(\mathcal{U}_{L-l}^l \otimes K_{L-l} + TI)^{-1}(\mathcal{U}_{L-l}^l \otimes K_{L-l})]$$

$$+ \frac{1}{N}\mathrm{Tr}_P[(\mathcal{U}_{L-l}^l \otimes K_{L-l} + TI)^{-1}\hat{Y}\hat{Y}^\top(\mathcal{U}_{L-l}^l \otimes K_{L-l} + TI)^{-1}(\mathcal{U}_{L-l}^l \otimes K_{L-l})], \tag{C15}$$

where the matrices on the rhs before taking the trace are of dimension $mP \times mP$, and Tr_P denotes summing the P diagonal blocks of size $m \times m$. Unlike the zero-temperature case, we cannot obtain saddle-point equations for each of the eigenvalues of $\mathcal{U}_{L-l}$. The kernel undergoes renormalization in the form of a Kronecker product with the renormalization matrix $\mathcal{U}_l$.

The predictor statistics of the multiple output case at finite temperature is given by

$$\langle f(x_{P+1})\rangle = [\mathcal{U}_0^L \otimes k_0^\top(x)](\mathcal{U}_0^L \otimes K_0 + TI)^{-1}\hat{Y}, \tag{C16}$$

$$\langle\delta f(x_{P+1})\delta f(x_{P+1})^\top\rangle = \mathcal{U}_0^L K_0(x,x) - [\mathcal{U}_0^L \otimes k_0^\top(x)](\mathcal{U}_0^L \otimes K_0 + TI)^{-1}[\mathcal{U}_0^L \otimes k_0(x)]. \tag{C17}$$

See another formulation of the results without the Kronecker product in Sec. IV B of Ref. [33].

APPENDIX D: WEIGHT COVARIANCE AND MEAN LAYER KERNELS

We derive the mean layer kernels for the multiple output network, starting from calculating $\langle w_L w_L^\top \rangle_L$, where w_L is the weight vector corresponding to a single node in the Lth hidden layer, conditioned on the weights of the previous $L-1$ layers. Using Eqs. (C1) and (C2), this quantity can be expressed as

$$\langle w_L w_L^\top \rangle_L = \frac{1}{Z_{L-1}} \int dt A(t) \exp\left[i\mathrm{Tr}(t^\top Y) + NG(t) \right.$$
$$\left. - \frac{1}{2\sigma^2} \mathrm{Tr}(W'^\top W') \right], \tag{D1}$$

where

$$A(t) = \frac{\langle w_L w_L^\top \exp -\frac{1}{2N}\mathrm{Tr}(t^\top K_w^L t) \rangle_{w_L}}{\langle \exp -\frac{1}{2N}\mathrm{Tr}(t^\top K_w^L t) \rangle_{w_L}}$$

$$= \frac{1}{z}\int dw_L w_L w_L^\top$$

$$\times \exp\left[-\frac{1}{2\sigma^2} w_L^\top\left(I + \frac{\sigma^4}{NN_{L-1}} X_{L-1} t t^\top X_{L-1}^\top \right) w_L \right]$$

$$= \sigma^2\left[I + \frac{\sigma^4}{NN_{L-1}} X_{L-1} t t^\top X_{L-1}^\top \right]^{-1}$$

$$= \sigma^2\left[I - \frac{\sigma^4}{NN_{L-1}} X_{L-1} t\left[I + \frac{\sigma^2}{N} t^\top K_{L-1} t \right]^{-1} t^\top X_{L-1}^\top \right]. \tag{D2}$$

Plugging $A(t)$ back in, we have

$$\langle w_L w_L^T \rangle_L = \sigma^2 I - \frac{\sigma^6}{NN_{L-1}Z_{L-1}} \int dt \exp\left[i\mathrm{Tr}(t^\top Y) \right.$$
$$\left. - \frac{N}{2}\log\det\left(I + \frac{\sigma^2}{N} t^\top K_{L-1} t \right) - \frac{1}{2\sigma^2}\mathrm{Tr}(W'^\top W') \right]$$
$$\times X_{L-1} t\left[I + \frac{\sigma^2}{N} t^\top K_{L-1} t \right]^{-1} t^\top X_{L-1}^\top. \tag{D3}$$

The term $\mathrm{Tr}(W'^\top W')$ does not depend on t; therefore, we ignore it for simplicity below.

We first compute the integral over t by introducing OPs $\mathcal{U}_{L-1}$ and $\mathcal{H}_{L-1}$ as in Appendix C, and with a change of variable $\hat{t} = K_{L-1}^{1/2} t$, and the saddle-point relation $\mathcal{U}_{L-1}(I + \mathcal{H}_{L-1}) = \sigma^2 I$, we write the integration over t as

$$\int d\hat{t} \exp\left[i\mathrm{Tr}(\hat{t}^\top K_{L-1}^{-1/2} Y) - \frac{1}{2}\mathrm{Tr}(\hat{t}\mathcal{U}_{L-1}\hat{t}^\top) \right]$$
$$\times K_{L-1}^{-1/2}\hat{t}[I + \mathcal{H}_{L-1}]^{-1} t^\top K_{L-1}^{-1/2}$$
$$= K_{L-1}^{-1/2}[\mathrm{Tr}(\mathcal{U}_{L-1}^{-1}(I + \mathcal{H}_{L-1})^{-1})I$$
$$- \sigma^{-2}(K_{L-1}^{-1/2} Y \mathcal{U}_{L-1}^{-1})\mathcal{U}_{L-1}(K_{L-1}^{-1/2} Y \mathcal{U}_{L-1}^{-1})^\top]K_{L-1}^{-1/2}$$
$$= \sigma^{-2}(m K_{L-1}^{-1} - K_{L-1}^{-1} Y \mathcal{U}_{L-1}^{-1} Y^\top K_{L-1}^{-1}). \tag{D4}$$

Plugging back in Eq. (D3) yields

$$\langle w_L w_L^\top \rangle_L = \sigma^2 I - \frac{\sigma^4}{NN_{L-1}}[m X_{L-1} K_{L-1}^{-1} X_{L-1}^\top$$
$$- X_{L-1} K_{L-1}^{-1} Y \mathcal{U}_{L-1}^{-1} Y^\top K_{L-1}^{-1} X_{L-1}^\top]. \tag{D5}$$

In particular, the weight variance $\langle w_L^\top w_L \rangle_L$ equals

$$\langle w_L^\top w_L \rangle_L = \mathrm{Tr}\langle w_L w_L^\top \rangle = \sigma^2 N - \sigma^2\alpha(m - u_{L-1}^{-1} r_{L-1}), \tag{D6}$$

implying that while the GP term is of $\mathcal{O}(N)$ as expected, the non-GP correction term is of $\mathcal{O}(1)$.

Using similar methods, we can derive, for all layers,

$$\langle w_{L-l} w_{L-l}^\top \rangle_{L-l} = \sigma^2 I - \frac{\sigma^4}{NN_{L-l-1}}[m X_{L-l-1} K_{L-l-1}^{-1} X_{L-l-1}^\top$$
$$- X_{L-l-1} K_{L-l-1}^{-1} Y \mathcal{U}_{L-l-1}^{-(l+1)} Y^\top K_{L-l-1}^{-1} X_{L-l-1}^\top]. \tag{D7}$$

Since $\langle K_{L-l} \rangle = (\sigma^2/N_{L-l-1}) X_{L-l-1}^\top \langle w_{L-l} w_{L-l}^\top \rangle X_{L-l-1}$,

$$\langle K_{L-l} \rangle = \sigma^2\left[\left(1 - \frac{m}{N} \right) K_{L-l-1} + \frac{1}{N} Y \mathcal{U}_{L-l-1}^{-(l+1)} Y^\top \right]. \tag{D8}$$

Using Eqs. (C9) and (D8), we derive Eqs. (50) and (51) through iterations. The above result holds for all α. For derivation of this result in narrow networks, see Sec. VA of Ref. [33].

APPENDIX E: DETAILS OF NUMERICAL STUDIES

1. Examples

a. Template model

Instead of having the standard input statistics in many synthetic models, namely, sampled IID from a normal distribution, we assume a template model, in which inputs are clustered and the target rule largely obeys this structure, in order to introduce a strong correlation between input structure and target outputs, as explained below. These types of examples are common in practice, for instance, in MNIST and CIFAR-10, where the network is trained on clustered input data and the target labels exhibit

significant correlations with the cluster structure [see Figs. 9, 12, 14(k)–14(o) for results on the MNIST example]. Because of these input-output correlations, the template model can yield good generalization performance even well below the interpolation threshold. (The predicted dependence of the system's properties on the various parameters is general.) In principle, both training and test data would be sampled from the same cluster statistics. To simplify the analysis, instead we assume that training inputs consist of the cluster centers, or what we call the templates. The test inputs are sampled by adding Gaussian noise to the training inputs, such that the test data are clustered around the training data, as illustrated in Fig. 3(a),

$$x_{\text{test}}^{\mu} = \sqrt{1 - \gamma} x^{\mu} + \sqrt{\gamma} \eta. \qquad (E1)$$

We consider two types of labeling of the data, which differ in task complexity. One is a noisy linear teacher task, where the labels are generated by a noisy linear teacher network, $y = (1/\sqrt{N_0}) w_0^{\top} x + \sigma_0 \eta$. Here, $w_0 \sim \mathcal{N}(0, \sigma_w^2 I)$, $\eta \sim \mathcal{N}(0, I)$ are both Gaussian IID, σ_w represents the amplitude of the linear teacher weights, and σ_0 represents the noise level of the linear teacher. Here, the optimal weights of the linear DNNs are those that yield a linear input-output mapping identical to that given by the teacher weights. Because of the linearity of the rule, if σ_0 is small the system can yield a small training error even when $P > N_0$. If γ is also small, the system will again yield a good generalization error approaching its minimum on the rhs of the interpolation threshold ($\alpha_0 > 1$), when the network is in the underparametrized regime, converging to the optimal error for $\alpha_0 \gg 1$ [see Figs. 6(a)–6(d)]. The second task is a random labeling of the data clusters that are centered around the templates. We assign random binary labels to the data x^{μ}, $Y^{\mu} \in \{-1, 1\}$. For the test data, we assign label Y^{μ} to it if it is generated by adding noise to the training data x^{μ}. Here, for $P > N_0$, the task is inherently nonlinear (even for small γ), and the minimum generalization error is achieved on the lhs of the interpolation threshold ($\alpha_0 < 1$) because small P implies not only small sizes of training data but also an easier task.

Parameters for the noisy linear teacher.—In Fig. 3, the parameters are $N_0 = 1000$, $P = 300$, $\gamma = 0.05$, $\sigma_0 = 0.1$, and $\sigma_w = 1$. The top panels (c)–(f) show the small-noise regime where the generalization error decreases with N; here, we choose $\sigma = 0.5$. The bottom panels (g)–(j) show the larger noise regime where the generalization error increases with N, with $\sigma = 1.3$.

In Fig. 4, the parameters are $N_0 = 200$, $P = 100$, $\gamma = 0.05$, and $\sigma_0 = 0.1$. The top panels (a)–(d) show the subregime where the generalization error decreases with L; here, the parameters are $\sigma_w = 0.3$, $\sigma = 1.1$, and $\alpha = 0.5$. The middle panels (e)–(h) show the subregime where the generalization error increases with L but goes to a finite limit as $L \to \infty$; here, the parameters are $\sigma_w = 0.9$,

$\sigma = 1.35$, and $\alpha = 0.5$. The bottom panels (i)–(l) show the high-noise regime where the generalization error increases with L and diverges as $L \to \infty$; here, the parameters are $\sigma_w = 0.9$, $\sigma = 1.35$, and $\alpha = 0.4$.

In Figs. 6(a)–6(d), the parameters are $N_0 = 500$, $N = 200$, $\gamma = 0.1$, $\sigma_0 = 0.3$, $\sigma_w = 1$, and $\sigma = 1$.

In Fig. 11, the parameters are $N_0 = 400$, $P = 100$, $\gamma = 0.05$, $\sigma_0 = 0.1$, and $\sigma_w = 1$. The top panels (a)–(d) show the small-noise regime where the generalization error decreases with N; here, $\sigma = 1$. The bottom panels (e)–(h) show the high-noise regime where the generalization error increases with N; here, $\sigma = 2$.

In Fig. 13, the parameters are $N_0 = 100$, $P = 200$, $\gamma = 0.05$, $\sigma_0 = 0.1$, and $\sigma_w = 1$. The top panels (a)–(d) show the small-noise regime where the generalization error decreases with N; here, $\sigma = 1$. The bottom panels (e)–(h) show the high-noise regime where the generalization error increases with N; here, $\sigma = 1.3$.

In Figs. 14(a)–(e), the simulation parameters are $N_0 = 20$, $P = 300$, $\gamma = 0.05$, $\sigma_0 = 0.3$, $\sigma_w = 1$, and $\sigma = 1$.

In Figs. 15(a)–(d), the simulation parameters are $N_0 = 200$, $P = 100$, $\gamma = 0.05$, $\alpha = 0.7$, $\sigma_0 = 0.1$, $\sigma_w = 0.3$, and $\sigma = 1$.

Parameters for the random cluster labeling.—In Figs. 6(e)–6(h), the parameters are $N_0 = 500$, $N = 200$, $\gamma = 0.1$, and $\sigma = 1$.

In Fig. 7(a), the parameters are $N_0 = 400$, $N = 800$, $\gamma = 0.1$, and $\sigma = 0.5$. In Fig. 7(d), the parameters are $N_0 = 400$, $P = 300$, $N = 600$, $\gamma = 0.1$, and $\sigma = 1.5$.

In Figs. 14(e)–14(h), the simulation parameters are $N_0 = 20$, $P = 300$, $\gamma = 0.1$, and $\sigma = 1$.

b. Synthetic example with block structure

In Fig. 8, we present the layerwise mean kernels trained on a synthetic example with an output similarity matrix that exhibits a block structure. The parameters used in the simulation are $N = N_0 = 100$, $P = 80$, and $\sigma_0 = 0.1$. We choose $\sigma = 0.1$, so the non-GP correction term ($\sim \sigma^2 / N$) is of the same order as the input term ($\sim \sigma^4$) for the single hidden-layer network we consider. In Figs. 8(c) and 8(d), we show the simulation and theoretical results for the non-GP correction given by $(\sigma^2 / N) Y V U_1 V^T Y^T$.

c. Binary classification of randomly projected MNIST data

The example we show in Fig. 9 is trained on a binary classification task on a subset of randomly projected MNIST data. For MNIST, the input dimension is fixed to 784, the number of pixels in the images. In the example we show, we first appropriately normalize and center the data, such that it has zero mean and standard deviation 1; then, we randomly project the MNIST data to N_0 dimensions with a Gaussian IID weight matrix $W_0 \in \mathbb{R}^{N_0 \times 784}$, $W_0 \sim \mathcal{N}(0, I)$, and add a ReLU nonlinearity to the projected data,

$$x = \text{ReLU}\left(\frac{1}{\sqrt{784}} W_0 x_{\text{MNIST}}\right). \tag{E2}$$

We then further train the network with the input $x^\mu (\mu = 1, \ldots, P)$ and their corresponding labels.

In Fig. 9, the network is trained on a subset of the MNIST data with four different digits (1,5,6,7). The output of the network is six dimensional ($y \in \{1, -1\}^6$), designed to have hierarchical block structure; four of the binary outputs are one-hot vectors each encoding one digit. The four digits are divided into two categories [(1,7) and (5,6)]; the other two binary outputs each classify one of the two categories. The parameters are $N = N_0 = 1000$ and $P = 100$. We again choose small $\sigma (\sigma = 0.1)$ so that the non-GP correction term becomes evident in the layerwise mean kernels.

In Fig. 14, we use the same example to train a ReLU network with a single output to perform binary classification on two digits, 0 and 1. The parameters are $N_0 = 20$, $P = 300$, and $\sigma = 1$.

d. Binary classification on subsets of the MNIST data

We show in Fig. 12 the result for a ReLU network trained on a binary classification task on a subset of MNIST data directly. The MNIST data determine the input dimension $N_0 = 784$. We properly normalize and center the data, such that they have zero mean and standard deviation 1. We train on a subset of MNIST data with digits 0 and 1, and the network outputs $y \in 1, -1$. The parameters are $N_0 = 784$ and $P = 100$. The top panels (a)–(d) show the small-noise regime where the generalization error decreases with N; here, $\sigma = 0.8$. The bottom panels (e)–(h) show the large-noise regime where the generalization error increases with N, and $\sigma = 1.3$.

In Figs. 15(e)–15(h), we use the same example with parameters $N_0 = 784$, $P = 100$, $\alpha = 0.5$, and $\sigma = 1.3$.

2. Langevin dynamics

We run simulations to sample from the Gibbs distribution corresponding to the energy E given by Eq. (2), defined in Sec. II, and compute the statistics from the distribution to compare with our theory. We use the well-known result, that the Langevin dynamics

$$\Delta\Theta = -\epsilon\partial_\Theta E + \sqrt{2\epsilon T}\eta \tag{E3}$$

generates a time-dependent distribution on the state space that converges at long times to the Gibbs distribution. We perform the Langevin dynamics; at each iteration, we compute the predictor on a set of new points and r_l for the current weight, and we obtain samples of the predictor and r_l from the underlying Gibbs distribution. We can then calculate statistics of the predictor, including $\langle f(x) \rangle$,

$\langle \delta f(x)^2 \rangle$, and $\langle r_l \rangle / r_0$, and compare them with our theoretical results. Because we focus mostly on $T \to 0$, in our simulations we also choose small T [$T = 0.001$ for results presented in all figures except for Figs. 8 and 9, where because of the small σ values, we need to use $T = 0.0001$].

3. Finite T effects

In the simulations of Langevin dynamics, we choose small T in order to compare with the $T \to 0$ theoretical results. In most cases presented in this paper, choosing $T = 0.001$ (or, in Sec. V A, with $T = 0.0001$) is sufficient to approximate the $T \to 0$ limit. However, in Fig. 6 where we consider the dependence of the generalization error on P, as $\alpha_0 \to 1$, the kernel K_0 becomes singular, and the closer α_0 is to $\alpha_0 = 1$, the lower the temperature needs to be to approximate the zero T limit. For this reason, in Fig. 6, the solid curves show the theory for finite T with $T = 0.001$ as in the simulation, with the mean predictor and the variance $\langle \delta f(x)^2 \rangle$ as given by Eqs. (47) and (48), and the order parameter u_0 given by Eq. (46) with $l = L$. In Sec. IIE of Ref. [33], we show the same results as in Fig. 6 but now with an extra curve plotting the theoretical result for zero temperature. We see that $T = 0.001$ is a good approximation to the zero T theory when α_0 is not close to the interpolation threshold $\alpha_0 = 1$, but it deviates from it as α_0 approaches this threshold.

---

[1] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, *Learning to Communicate with Deep Multi-Agent Reinforcement Learning*, in *Advances in Neural Information Processing Systems* (2016), pp. 2145–2153.

[2] Y. Goldberg, *Neural Network Methods for Natural Language Processing*, Synth. Lect. Hum. Lang. Technol. **10.1**, 1 (2017).

[3] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, *Object Recognition with Gradient-Based Learning*, in *Shape, Contour and Grouping in Computer Vision* (Springer, New York, 1999), pp. 319–345.

[4] L. Deng, G. Hinton, and B. Kingsbury, *New Types of Deep Neural Network Learning for Speech Recognition and Related Applications: An Overview*, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (IEEE, Vancouver, BC, Canada, 2013), pp. 8599–8603.

[5] A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, A. Pritzel, M. J. Chadwick, T. Degris, J. Modayil *et al.*, *Vector-Based Navigation Using Grid-like Representations in Artificial Agents*, Nature (London) **557**, 429 (2018).

[6] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, *Deep Learning for Visual Understanding: A Review*, Neurocomputing; Variable Star Bulletin **187**, 27 (2016).

[7] T. Poggio, A. Banburski, and Q. Liao, *Theoretical Issues in Deep Networks*, Proc. Natl. Acad. Sci. U.S.A. **117**, 30039 (2020).

[8] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, *Understanding Deep Learning (Still) Requires Rethinking Generalization*, Commun. ACM **64**, 107 (2021).

[9] M. Baity-Jesi, L. Sagun, M. Geiger, S. Spigler, G. B. Arous, C. Cammarota, Y. LeCun, M. Wyart, and G. Biroli, *Comparing Dynamics: Deep Neural Networks versus Glassy Systems*, in *International Conference on Machine Learning* (2018), pp. 314–323.

[10] A. J. Ballard, R. Das, S. Martiniani, D. Mehta, L. Sagun, J. D. Stevenson, and D. J. Wales, *Energy Landscapes for Machine Learning*, Phys. Chem. Chem. Phys. **19**, 12585 (2017).

[11] S. Becker, Y. Zhang, and A. A. Lee, *Geometry of Energy Landscapes and the Optimizability of Deep Neural Networks*, Phys. Rev. Lett. **124**, 108301 (2020).

[12] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, *Contractive Auto-Encoders: Explicit Invariance During Feature Extraction*, in *International Conference on Learning Representations* (2011).

[13] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, *Machine Learning and the Physical Sciences*, Rev. Mod. Phys. **91**, 045002 (2019).

[14] A. Engel and C. Van den Broeck, *Statistical Mechanics of Learning* (Cambridge University Press, Cambridge, England, 2001).

[15] M. Mezard and A. Montanari, *Information, Physics, and Computation* (Oxford University Press, New York, 2009).

[16] M. S. Advani, A. M. Saxe, and H. Sompolinsky, *High-Dimensional Dynamics of Generalization Error in Neural Networks*, Neural Netw. **132**, 428 (2020).

[17] G.-X. Yuan, C.-H. Ho, and C.-J. Lin, *Recent Advances of Large-Scale Linear Classification*, Proc. IEEE **100**, 2584 (2012).

[18] A. M. Saxe, J. L. McClelland, and S. Ganguli, *A Mathematical Theory of Semantic Development in Deep Neural Networks*, Proc. Natl. Acad. Sci. U.S.A. **116**, 11537 (2019).

[19] A. M. Saxe, J. L. Mcclelland, and S. Ganguli, *Exact Solutions to the Nonlinear Dynamics of Learning in Deep Linear Neural Network*, in *International Conference on Learning Representations* (Citeseer, Banff, AB, Canada, 2014).

[20] N. Tishby, E. Levin, and S. A. Solla, *Consistent Inference of Probabilities in Layered Networks: Predictions and Generalization*, in *International Joint Conference on Neural Networks* (1989), Vol. 2, pp. 403–409.

[21] D. J. MacKay, *A Practical Bayesian Framework for Backpropagation Networks*, Neural Comput. **4**, 448 (1992).

[22] R. M. Neal, *Bayesian Learning for Neural Networks* (Springer Science & Business Media, New York, 2012), Vol. 118.

[23] Y. Bahri, J. Kadmon, J. Pennington, S. S. Schoenholz, J. Sohl-Dickstein, and S. Ganguli, *Statistical Mechanics of Deep Learning*, Annu. Rev. Condens. Matter Phys. **11**, 501 (2020).

[24] D. J. Amit, H. Gutfreund, and H. Sompolinsky, *Statistical Mechanics of Neural Networks Near Saturation*, Ann. Phys. (N.Y.) **173**, 30 (1987).

[25] M. Advani, S. Lahiri, and S. Ganguli, *Statistical Mechanics of Complex Neural Systems and High Dimensional Data*, J. Stat. Mech. (2013) P03014.

[26] H. S. Seung, H. Sompolinsky, and N. Tishby, *Statistical Mechanics of Learning from Examples*, Phys. Rev. A **45**, 6056 (1992).

[27] T. L. Watkin, A. Rau, and M. Biehl, *The Statistical Mechanics of Learning a Rule*, Rev. Mod. Phys. **65**, 499 (1993).

[28] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, *Deep Neural Networks as Gaussian Processes*, in *International Conference on Learning Representations* (2018), arXiv:1711.00165.

[29] Y. Cho and L. K. Saul, *Kernel Methods for Deep Learning*, in *Advances in Neural Information Processing Systems* (2009), pp. 342–350.

[30] S. Chung, D. D. Lee, and H. Sompolinsky, *Classification and Geometry of General Perceptual Manifolds*, Phys. Rev. X **8**, 031003 (2018).

[31] S. Ganguli and H. Sompolinsky, *Statistical Mechanics of Compressed Sensing*, Phys. Rev. Lett. **104**, 188701 (2010).

[32] S. Ganguli and H. Sompolinsky, *Compressed Sensing, Sparsity, and Dimensionality in Neuronal Information Processing and Data Analysis*, Annu. Rev. Neurosci. **35**, 485 (2012).

[33] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevX.11.031059 for further details of the theory and the simulations.

[34] B. Babadi and H. Sompolinsky, *Sparseness and Expansion in Sensory Representations*, Neuron **83**, 1213 (2014).

[35] S. Mei and A. Montanari, *The Generalization Error of Random Features Regression: Precise Asymptotics and Double Descent Curve*, Comm. Pure Appl. Math. (2021) https://doi.org/10.1002/cpa.22008.

[36] M. Belkin, D. Hsu, S. Ma, and S. Mandal, *Reconciling Modern Machine-Learning Practice and the Classical Bias–Variance Trade-off*, Proc. Natl. Acad. Sci. U.S.A. **116**, 15849 (2019).

[37] J. Shawe-Taylor, N. Cristianini *et al.*, *Kernel Methods for Pattern Analysis* (Cambridge University, Cambridge, England, 2004).

[38] T. Hofmann, B. Schölkopf, and A. J. Smola, *Kernel Methods in Machine Learning*, Ann. Stat. **36**, 1171 (2008).

[39] K. Cutajar, E. V. Bonilla, P. Michiardi, and M. Filippone, *Random Feature Expansions for Deep Gaussian Processes*, in *International Conference on Machine Learning* (PMLR, Sydney, NSW, Australia, 2017), pp. 884–893.

[40] A. Rahimi and B. Recht, *Random Features for Large-Scale Kernel Machines*, in *Advances in Neural Information Processing Systems* (2007), pp. 1177–1184.

[41] R. Vershynin, *Memory Capacity of Neural Networks with Threshold and Rectified Linear Unit Activations*, SIAM J. Math. Data Sci. **2**, 1004 (2020).

[42] E. Gardner, *The Space of Interactions in Neural Network Models*, J. Phys. A **21**, 257 (1988).

[43] E. Gardner and B. Derrida, *Optimal Storage Properties of Neural Network Models*, J. Phys. A **21**, 271 (1988).

[44] P. Domingos, *A Unified Bias-Variance Decomposition for Zero-One and Squared Loss*, AAAI/IAAI **2000**, 564 (2000).

[45] S. Geman, E. Bienenstock, and R. Doursat, *Neural Networks and the Bias/Variance Dilemma*, Neural Comput. **4**, 1 (1992).

[46] T. Laurent and J. Brecht, *Deep Linear Networks with Arbitrary Loss: All Local Minima Are Global*, in *International Conference on Machine Learning* (PMLR, Stockholm, Sweden, 2018), pp. 2902–2907.

[47] H. Lu and K. Kawaguchi, *Depth Creates No Bad Local Minima*, arXiv:1702.08580.

[48] D. L. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo, *Performance-Optimized Hierarchical Models Predict Neural Responses in Higher Visual Cortex*, Proc. Natl. Acad. Sci. U.S.A. **111**, 8619 (2014).

[49] N. Kriegeskorte, M. Mur, and P. A. Bandettini, *Representational Similarity Analysis-Connecting the Branches of Systems Neuroscience*, Front. Syst. Neurosci. **2**, 4 (2008).

[50] A. Messinger, L. R. Squire, S. M. Zola, and T. D. Albright, *Neuronal Representations of Stimulus Associations Develop in the Temporal Lobe During Learning*, Proc. Natl. Acad. Sci. U.S.A. **98**, 12239 (2001).

[51] R. Dodier, *Geometry of Early Stopping in Linear Networks*, Adv. Neural Inf. Process. Syst. 365 (1995).

[52] M. Li, M. Soltanolkotabi, and S. Oymak, *Gradient Descent with Early Stopping Is Provably Robust to Label Noise for Overparametrized Neural Networks*, in *International Conference on Artificial Intelligence and Statistics* (PMLR, Palermo, Sicily, Italy, 2020), pp. 4313–4324.

[53] G. Naveh, O. Ben-David, H. Sompolinsky, and Z. Ringel, *Predicting the Outputs of Finite Networks Trained with Noisy Gradients*, arXiv:2004.01190.

[54] J. M. Antognini, *Finite Size Corrections for Neural Network Gaussian Processes*, arXiv:1908.10030.

[55] J. S. Yedidia, W. T. Freeman, Y. Weiss, *Generalized Belief Propagation*, in *Advances in Neural Information Processing Systems* (2000), Vol. 13, pp. 668–674.

[56] J. S. Yedidia, W. T. Freeman, Y. Weiss *et al.*, *Understanding Belief Propagation and Its Generalizations*, in *Exploring Artificial Intelligence in the New Millennium* (2003), Vol. 8, pp. 236–239.

[57] J. S. Yedidia, W. T. Freeman, and Y. Weiss, *Constructing Free-Energy Approximations and Generalized Belief Propagation Algorithms*, IEEE Trans. Inf. Theory **51**, 2282 (2005).

[58] J. Pearl, *Fusion, Propagation, and Structuring in Belief Networks*, Artif. Intell. **29**, 241 (1986).

[59] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Elsevier, New York, 2014).

[60] Y. Weiss and J. Pearl, *Belief Propagation: Technical Perspective*, Commun. ACM **53**, 94 (2010).

[61] J. Winn, C. M. Bishop, and T. Jaakkola, *Variational Message Passing*, J. Mach. Learn. Res. **6**, 661 (2005).

[62] T. Parr, D. Markovic, S. J. Kiebel, and K. J. Friston, *Neuronal Message Passing Using Mean-Field, Bethe, and Marginal Approximations*, Sci. Rep. **9**, 1889 (2019).

[63] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, *Factor Graphs and the Sum-Product Algorithm*, IEEE Trans. Inf. Theory **47**, 498 (2001).

[64] J. M. Hernández-Lobato and R. Adams, *Probabilistic Back-propagation for Scalable Learning of Bayesian Neural Networks*, in *International Conference on Machine Learning* (PMLR, Lille, France, 2015), pp. 1861–1869.

[65] A. Graves, *Practical Variational Inference for Neural Networks*, in *Advances in Neural Information Processing Systems* (Citeseer, Granada, Spain, 2011), pp. 2348–2356.

[66] N. Goldenfeld, *Lectures on Phase Transitions and the Renormalization Group* (CRC Press, Boca Raton, FL, 2018).

[67] L.-Y. Chen, N. Goldenfeld, and Y. Oono, *Renormalization Group and Singular Perturbations: Multiple Scales, Boundary Layers, and Reductive Perturbation Theory*, Phys. Rev. E **54**, 376 (1996).

[68] S. W. Pierson and O. T. Valls, *Renormalization-Group Study of a Layered-Superconductor Model*, Phys. Rev. B **45**, 13076 (1992).

[69] S. W. Pierson, *Critical Behavior of Vortices in a Layered System*, Phys. Rev. Lett. **73**, 2496 (1994).

[70] S.-H. Li and L. Wang, *Neural Network Renormalization Group*, Phys. Rev. Lett. **121**, 260601 (2018).

[71] M. Van der Wilk, C. E. Rasmussen, and J. Hensman, *Convolutional Gaussian Processes*, in *Advances in Neural Information Processing Systems* (2017), pp. 2845–2854.

[72] A. Garriga-Alonso, C. E. Rasmussen, and L. Aitchison, *Deep Convolutional Networks as Shallow Gaussian Processes*, in *International Conference on Learning Representations* (2018), pp. 1–16.

[73] R. Novak, L. Xiao, J. Lee, Y. Bahri, G. Yang, J. Hron, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein, *Bayesian Deep Convolutional Networks with Many Channels are Gaussian Processes*, in *International Conference on Learning Representations* (2019), arXiv:1810.05148.

[74] Z. Bai and J. W. Silverstein, *Sample Covariance Matrices and the Marčenko-Pastur Law*, in *Spectral Analysis of Large Dimensional Random Matrices* (Springer, New York, 2010), pp. 39–58.