Editors' Suggestion

Memorizing without overfitting: Bias, variance, and interpolation in overparameterized models

Jason W. Rocks[1] and Pankaj Mehta[1,2,*]
[1]*Department of Physics, Boston University, Boston, Massachusetts 02215, USA*
[2]*Faculty of Computing and Data Sciences, Boston University, Boston, Massachusetts 02215, USA*

(Received 29 October 2020; revised 9 June 2021; accepted 22 February 2022; published 15 March 2022)

The bias-variance trade-off is a central concept in supervised learning. In classical statistics, increasing the complexity of a model (e.g., number of parameters) reduces bias but also increases variance. Until recently, it was commonly believed that optimal performance is achieved at intermediate model complexities which strike a balance between bias and variance. Modern deep learning methods flout this dogma, achieving state-of-the-art performance using "overparameterized models" where the number of fit parameters is large enough to perfectly fit the training data. As a result, understanding bias and variance in overparameterized models has emerged as a fundamental problem in machine learning. Here, we use methods from statistical physics to derive analytic expressions for bias and variance in two minimal models of overparameterization (linear regression and two-layer neural networks with nonlinear data distributions), allowing us to disentangle properties stemming from the model architecture and random sampling of data. In both models, increasing the number of fit parameters leads to a phase transition where the training error goes to zero and the test error diverges as a result of the variance (while the bias remains finite). Beyond this threshold, the test error of the two-layer neural network decreases due to a monotonic decrease in *both* the bias and variance as opposed to the classical bias-variance trade-off. We also show that in contrast with classical intuition, overparameterized models can overfit even in the absence of noise and exhibit bias even if the student and teacher models match. We synthesize these results to construct a holistic understanding of generalization error and the bias-variance trade-off in overparameterized models and relate our results to random matrix theory.

DOI: 10.1103/PhysRevResearch.4.013201

I. INTRODUCTION

Machine learning (ML) is one of the most exciting and fastest-growing areas of modern research and application. Over the last decade, we have witnessed incredible progress in our ability to learn statistical relationships from large data sets and make accurate predictions. Modern ML techniques have now made it possible to automate tasks such as speech recognition, language translation, and visual object recognition, with wide-ranging implications for fields such as genomics, physics, and even mathematics. These techniques—in particular the deep learning methods that underlie many of the most prominent recent advancements—are especially successful at tasks that can be recast as supervised learning problems [1]. In supervised learning, the goal is to learn statistical relationships from labeled data (e.g., a collection of pictures labeled as containing a cat or not containing a cat). Common examples of supervised learning tasks include classification and regression.

A fundamental concept in supervised learning is the bias-variance trade-off. In general, the out-of-sample, generalization, or test error, of a statistical model can be decomposed into three sources: bias (errors resulting from erroneous assumptions which can hamper a statistical model's ability to fully express the patterns hidden in the data), variance (errors arising from oversensitivity to the particular choice of training set), and noise. This bias-variance decomposition provides a natural intuition for understanding how complex a model must be in order to make accurate predictions on unseen data. As model complexity (e.g., the number of fit parameters) increases, bias decreases as a result of the model becoming more expressive and better able to capture complicated statistical relationships in the underlying data distribution. However, a more complex model may also exhibit higher variance as it begins to overfit, becoming less constrained and therefore more sensitive to the quirks of the training set (e.g., noise) that do not generalize to other data sets. This trade-off is reflected in the generalization error in the form of a classical "U-shaped" curve: the test error first decreases with model complexity until it reaches a minimum before increasing dramatically as the model overfits the training data. For this reason, it was commonly believed until recently that optimal performance is achieved at intermediate model complexities which strike a balance between bias (underfitting) and variance (overfitting).

Modern deep learning methods defy this understanding, achieving state-of-the-art performance using "overparameterized models" where the number of fit parameters is so large—often orders of magnitude larger than the number of data points [2]—that one would expect a model's accuracy to be overwhelmed by overfitting. In fact, empirical experiments show that convolutional networks commonly used in image

[*] pankajm@bu.edu

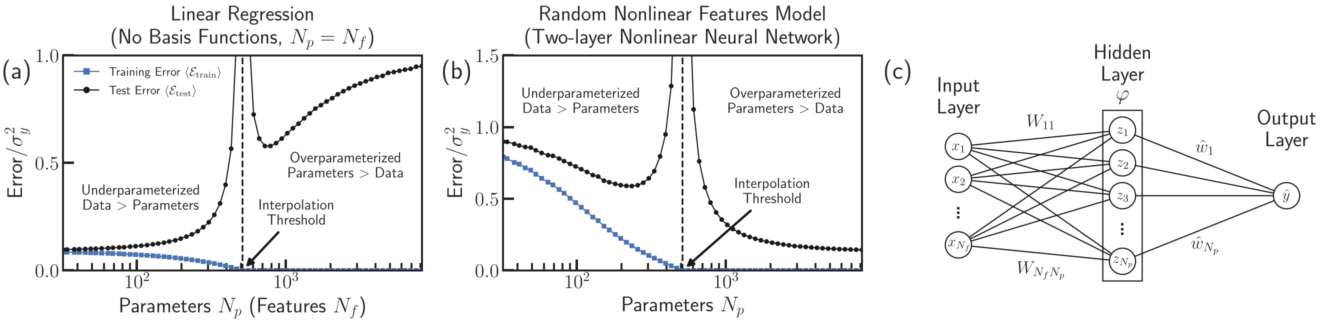

FIG. 1. Double-descent phenomenon. [(a) and (b)] Examples of the average training error (blue squares) and test error (black circles) for two different models calculated via numerical simulations. In both models, the test error diverges when the training error reaches zero at the interpolation threshold, located where the number of parameters N_p matches the number of points in the training data set M (indicated by a black dashed vertical line). (a) In linear regression without basis functions, the number of features in the data N_f matches the number of fit parameters N_p. (b) The random nonlinear features model (two-layer neural network where the parameters of the middle layer are random but fixed) decouples the number of features N_f and the number of fit parameters N_p by incorporating an additional "hidden layer" and transforms the data using a nonlinear activation function (e.g., ReLU), resulting in the canonical double-descent behavior. (c) Schematic of the model architecture for the random nonlinear features model. Numerical results are shown for a linear teacher model $y(\vec{\mathbf{x}}) = \vec{\mathbf{x}} \cdot \vec{\boldsymbol{\beta}} + \varepsilon$, a signal-to-noise ratio of $\sigma_\beta^2\sigma_X^2/\sigma_\varepsilon^2 = 10$, and a small regularization parameter of $\lambda = 10^{-6}$. The y axes have been scaled by the variance of the training set labels $\sigma_y^2 = \sigma_\beta^2\sigma_X^2 + \sigma_\varepsilon^2$. Each point is averaged over at least 1000 independent simulations trained on $M = 512$ data points with small error bars indicating the error on the mean. In (b), there are less features than data points $N_f = M/4$. See Sec. II for precise definitions and Sec. S4 of Ref. [5] for additional simulation details.

classification are so overly expressive that they can easily fit training data with randomized labels, or even images generated from random noise, with almost perfect accuracy [3]. Despite the apparent risks of overfitting, these models seem to perform at least as well as, if not better than, traditional statistical models. As a result, modern best practices in deep learning recommend using highly overparameterized models that are expressive enough to achieve zero error on the training data [4].

Clearly, the classical picture provided by the bias-variance trade-off is incomplete. Classical statistics largely focuses on underparameterized models which are simple enough that they have a nonzero training error. In contrast, modern deep learning methods push model complexity past the *interpolation threshold*, the point at which the training error reaches zero [6–10]. In the classical picture, approaching the interpolation threshold coincides with a large increase, or even divergence, in the test error via the variance. However, numerical experiments suggest that the predictive performance of overparameterized models is better described by "double-descent" curves which extend the classic U-shape past the interpolation threshold to account for overparameterized models with zero training error [6,11,12]. Surprisingly, if model complexity is increased past the interpolation threshold, the test error once again decreases, often resulting in overparameterized models with even better out-of-sample performance than their underparameterized counterparts [see Fig. 1(b)].

This double-descent behavior stands in stark contrast with the classical statistical intuition based on the bias-variance trade-off; both bias *and* variance appear to decrease past the interpolation threshold. Therefore understanding this phenomenon requires generalizing the bias-variance decomposition to overparameterized models. More broadly, explaining the unexpected success of overparameterized models represents a fundamental problem in ML and modern statistics.

A. Relation to previous work

In recent years, many attempts have been made to understand the origins of this double-descent behavior via numerical and/or analytical approaches. While much of this work has relied on well constructed numerical experiments on complex deep learning models [3,6,8,13], many theoretical studies have focused on a much simpler setting: the so-called "lazy training" regime. Previously, it was observed that in the limit of an infinitely wide network, the learning process appears to mimic that of an approximate kernel method in which the kernel used by the model to express the data—the neural tangent kernel (NTK)—remains fixed [14,15]. This stands in contrast to the so-called "feature training" regime in which the kernel evolves over time as the model learns the most informative way to express the relationships in the data [16].

Making use of the observation that the kernel remains approximately fixed in the lazy regime, many analytical studies have derived training and test errors for neural networks where the top layer is trained, but the middle layer(s) remained fixed, effectively reducing these models to linearized versions of regression or classification with various types of nontrivial basis functions [17–44]. Furthermore, some of these studies have considered nonlinear data distributions [28,40]. Importantly, such studies have typically combined a fixed-kernel approach with specific choices of loss functions (e.g., mean-squared error) to guarantee convexity, while the loss landscapes of neural networks in practical settings are often highly nonconvex [45]. Despite being limited to the lazy regime and convex loss landscapes, the closed-form solutions for the training and test error obtained for these models exhibit the double-descent phenomenon, demonstrating that many of the key features of more complex deep learning architectures can arise in much simpler settings.

A smaller subset of these studies have also attempted to extend these calculations to compute the bias-variance decomposition [17,19,25–27,30,31,34,38,40,42,44]. However, this literature is rife with qualitative and quantitative disagreements, and as a result, a consensus has not formed regarding many of the basic properties of bias and variance in overparameterized models. Underlying these disagreements is the ubiquitous use of nonstandard and varying definitions of bias and variance (see Sec. V F for an in-depth discussion).

For example, some studies consider a fixed design matrix for the training data set in the definitions of bias and variance, but not for the test set, resulting in an effective mismatch in their data distributions [17,19,27,30,31,34,40]. As a result, these studies do not even reproduce the classical bias-variance trade-off expected in the underparameterized regime. Meanwhile, other studies do not distinguish between sources of randomness stemming from the model architecture (e.g., due to initialization) and sampling of the training data set, inadvertently leading these analyses to derive the bias of ensemble models rather than the models actually under investigation [17,25,26,31,38,44]. Consequently, these studies have found that in the absence of regularization, the bias reaches a minimum at the interpolation threshold and then remains constant into the overparameterized regime.

In fact, of these studies, closed-form expressions using the standard definitions of bias and variance have only been obtained for the simple case of linear regression without basis functions and a linear data distribution [42]. While this setting captures some qualitative aspects of the double-descent phenomenon [see Fig. 1(a)], it requires a one-to-one correspondence between features in the data and fit parameters and a perfect match between the data distribution and model architecture, making it difficult to understand, if and how these results generalize to more complicated statistical models.

In line with previous studies, in this work, we also focus on the lazy regime with a convex loss landscape, considering two different linear models that emulate many properties of more complicated neural network architectures (see next section). However, our approach differs in that we utilize the traditional definitions of bias and variance, allowing us to clear up much of the confusion surrounding the bias-variance decomposition in overparameterized models. In this way, we connect modern deep learning to the statistical literature of the last century and in doing so, gain proper intuition for the origins of the double-descent phenomenon.

B. Overview of approach

In this work, we use methods from statistical physics to derive analytic results for bias and variance in the overparameterized regime for two minimal model architectures. These models, whose behavior is depicted in Fig. 1, are *linear regression* (ridge regression without basis functions and in the limit where the regularization parameter goes to zero—often called "ridge-less regression" in the statistics and ML literature) and the *random nonlinear features model* (a two-layer neural network with an arbitrary nonlinear activation function where the top layer is trained and parameters for the intermediate layer are chosen to be random but fixed). We generate the data used to train both models using a nonlinear "teacher model" where the labels are related to the features through a nonlinear function (usually with additive noise). Using similar terminology, we often refer to the details of a model's architecture as the "student model." Crucially, the differences between the two models we consider allow us to disentangle the effects of model architecture on bias and variance versus effects arising from randomly sampling the underlying data distribution.

Linear regression is one of the simplest models in which the test error diverges at the interpolation threshold but then decreases in the overparameterized regime [Fig. 1(a)]. Because this model uses the features in the data directly without modification (i.e., it lacks a hidden layer), it provides evidence that the process of randomly sampling the data itself plays an integral part in the double-descent phenomena. The random nonlinear features model [Figs. 1(b) and 1(c)] provides insight into the effects of filtering the features through an additional transformation, in effect, changing the way the model views the data. This disconnect between features and their representations in the model is crucial for understanding bias and variance in more complex overparameterized models.

To treat these models analytically, we make use of the "zero-temperature cavity method" which has a long history in the physics of disordered systems and statistical learning theory [46–48]. In particular, our calculations follow the style of Ref. [49] and assume that the solutions can be described using a replica-symmetric ansatz, which we confirm numerically. Our analytic results are exact in the thermodynamic limit where the number of data points M, the number of features in the data N_f, and the number of fit parameters (hidden features) N_p all tend towards infinity. Crucially, when taking this limit, the ratios between these three quantities are assumed to be fixed and finite, allowing us to ask how varying these ratios and other model properties (such as linear versus nonlinear activation functions) affect generalization. We confirm that all our analytic expressions agree extremely well with numerical results, even for relatively small system sizes ($N_f, N_p, M \sim$ 10–1000).

C. Summary of major results

Before proceeding further, we briefly summarize our major results.

(1) We derive analytic expressions for the test (generalization) error, training error, bias, and variance for both models using the zero-temperature cavity method.

(2) We show that both models exhibit a phase transition at an interpolation threshold to an interpolation regime where the training error is zero.

(3) In the underparameterized regime, we find that the variance diverges as it approaches the interpolation threshold, leading to extremely large generalization error, while the bias either remains constant (linear regression) or decreases monotonically in a classical bias-variance trade-off (random nonlinear features model).

(4) In the overparameterized regime, we find that the test error either decreases nonmonotonically due to a decrease in variance and an increase in bias (linear regression) or decreases monotonically due to a monotonic decrease in both

variance and bias, even in the absence of regularization (random nonlinear features model).

(5) We show that bias in overparameterized models has two sources: error resulting from mismatches between the student and teacher models (i.e., the model is incapable of fully capturing the full data distribution) and incomplete sampling of the data's feature space. We show that as a result of the latter case, one can have nonzero bias even if the student and teacher models are identical. We also show that bias decreases in the interpolation regime only if the number of features in the data remains fixed.

(6) We show that biased models can overfit even in the absence of noise. In other words, biased models can interpret signal as noise.

(7) We show that the zero-temperature susceptibilities that appear in our cavity calculations measure the sensitivity of a fitted model to small perturbations. We discuss how these susceptibilities can be used to identify phase transitions and are each related to different aspects of the double-descent phenomena.

(8) We combine these observations to provide a comprehensive intuitive explanation of the double-descent curves for test error observed in overparameterized models, making connections to random matrix theory. In particular, we discuss how diverging test error stems from small eigenvalues in the Hessian, corresponding to poorly sampled directions in the space of input features for linear regression or the space of hidden features for the random nonlinear features model.

(9) We discuss why using the standard definitions of bias and variance are necessary to properly connect the double-descent phenomenon to the classical bias-variance trade-off.

D. Organization of paper

In Sec. II, we start by providing the theoretical setup for both models and briefly summarize the methods we use to derive analytic expressions. In Sec. III, we provide precise definitions of bias and variance, taking great care to distinguish between different sources of randomness. In Sec. IV, we report our analytic results and compare them to numerical simulations. In Sec. V, we use these analytic expressions to understand how bias and variance generalize to overparameterized models and also discuss the roles of the susceptibilities that arise as part of our cavity calculations. Finally, in Sec. VI, we conclude and discuss the implications of our results for modern ML methods.

II. THEORETICAL SETUP

A. Supervised learning task

In this work, we consider data points $(y, \vec{\mathbf{x}})$, each consisting of a continuous label y paired with a set of N_f continuous features $\vec{\mathbf{x}}$. To distinguish the features in the data from those in the model, we refer to $\vec{\mathbf{x}}$ as the "input features." We frame the supervised learning task as follows: using the relationships learned from a training data set, construct a model to accurately predict the labels y of new data points based on their input features $\vec{\mathbf{x}}$.

B. Data distribution (teacher model)

We assume that the relationship between the input features and labels (the data distribution or teacher model) can be expressed as

$$y(\vec{\mathbf{x}}) = y^*(\vec{\mathbf{x}}; \vec{\boldsymbol{\beta}}) + \varepsilon, \tag{1}$$

where ε is the label noise and $y^*(\vec{\mathbf{x}}; \vec{\boldsymbol{\beta}})$ is an unknown function representing the "true" labels. This function takes the features as arguments and combines them with a set of N_f "ground truth" parameters $\vec{\boldsymbol{\beta}}$ which characterize the correlations between the features and labels.

We draw the input features for each data point independently and identically from a normal distribution with zero mean and variance σ_X^2/N_f. Normalizing the variance by N_f ensures that the magnitude of each feature vector is independent of the number of features and that a proper thermodynamic limit exists. Note that in this work, we consider features that do not contain noise. We also choose each element of the ground truth parameters $\vec{\boldsymbol{\beta}}$ and the label noise ε to be independent of the input features and mutually independent from one another, drawn from normal distributions with zero mean and variances σ_β^2 and σ_ε^2, respectively.

In this work, we restrict ourselves to a teacher model of the form

$$y^*(\vec{\mathbf{x}}; \vec{\boldsymbol{\beta}}) = \frac{\sigma_\beta \sigma_X}{\langle f' \rangle} f\left(\frac{\vec{\mathbf{x}} \cdot \vec{\boldsymbol{\beta}}}{\sigma_X \sigma_\beta}\right), \tag{2}$$

where the function f is an arbitrary nonlinear function and $\langle f' \rangle = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dh e^{-\frac{h^2}{2}} f'(h)$ is a normalization constant chosen for convenience with prime notation used to indicate a derivative (see Sec. S1D of Ref. [5]). We place a factor of $1/(\sigma_X \sigma_\beta)$ inside the function f so that its argument has unit variance, while the prefactor $\sigma_\beta \sigma_X$ ensures that y^* reduces to a linear teacher model $y^*(\vec{\mathbf{x}}) = \vec{\mathbf{x}} \cdot \vec{\boldsymbol{\beta}}$ when $f(h) = h$. Furthermore, we assume both the labels and input features are centered so that f has zero mean with respect to its argument. While the results we report hold for a general f of this form, all figures show numerical simulations for a linear teacher model unless otherwise specified.

C. Model architectures (student models)

We consider two different student models that we discuss in detail below: linear regression and the random nonlinear features model. A schematic of the network architecture for the latter model is depicted in Fig. 1(c). Both student models take the general form

$$\hat{y}(\vec{\mathbf{x}}) = \vec{\mathbf{z}}(\vec{\mathbf{x}}) \cdot \hat{\mathbf{w}}, \tag{3}$$

where $\hat{\mathbf{w}}$ is a vector of fit parameters and $\vec{\mathbf{z}}(\vec{\mathbf{x}})$ is a vector of "hidden" features which each may depend on a combination of the input features $\vec{\mathbf{x}}$. The hidden features $\vec{\mathbf{z}}$ are effectively the representations of the data points from the perspective of the model.

Since we only fit the top layer of the network in both models, the number of fit parameters N_p equals the number of hidden features, and Eq. (3) is equivalent to a linear model with basis functions. Despite its simplicity, we will show that this model reproduces much of the interesting behaviors observed in more complicated neural networks.

1. Linear regression

For linear regression without basis functions [Fig. 1(a)], the representations of the features from the perspective of the model are simply the input features themselves so that $\vec{\mathbf{z}}(\vec{\mathbf{x}}) = \vec{\mathbf{x}}$. In other words, the hidden and input features are identical, leading to exactly one fit parameter for each feature, $N_p = N_f$.

2. Random nonlinear features model

In the random nonlinear features model [Figs. 1(b) and 1(c)], the hidden features for each data point are related to the input features via a random matrix W of size $N_f \times N_p$ and a nonlinear activation function φ,

$$\vec{\mathbf{z}}(\vec{\mathbf{x}}) = \frac{1}{\langle\varphi'\rangle}\frac{\sigma_W\sigma_X}{\sqrt{N_p}}\varphi\left(\frac{\sqrt{N_p}}{\sigma_W\sigma_X}W^T\vec{\mathbf{x}}\right), \tag{4}$$

where φ acts separately on each element of its input and $\langle\varphi'\rangle = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\mathrm{d}he^{-\frac{h^2}{2}}\varphi'(h)$ is a normalization constant chosen for convenience (see Sec. S1D of Ref. [5]). We take each element of W to be independent and identically distributed, drawn from a normal distribution with zero mean and variance σ_W^2/N_p. The normalization by N_p is chosen so that the magnitude of each hidden feature vector only depends on the ratio of the number of input features to parameters which we always take to be finite. We place a factor of $\sqrt{N_p}/(\sigma_W\sigma_X)$ inside the activation function so that its argument has approximately unit variance, and a second pre-factor of $\sigma_W\sigma_X/\sqrt{N_p}$ in front of φ to ensure that $\vec{\mathbf{z}}$ reduces to a linear model $\vec{\mathbf{z}}(\vec{\mathbf{x}}) = W^T\vec{\mathbf{x}}$ when the activation function is linear [i.e., $\varphi(h) = h$]. We note that while this model is technically a linear model with a specific choice of basis functions, it is equivalent to a two-layer neural network where the weights of the middle layer are chosen to be random and only the top layer is trained. Although our analytic results hold for an arbitrary nonlinear activation function φ, all figures show results for ReLU activation where $\varphi(h) = \max(0, h)$.

D. Fitting procedure

We train each model on a training data set consisting of M data points, $\mathcal{D} = \{(y_a, \vec{\mathbf{x}}_a)\}_{a=1}^M$. For convenience, we organize the vectors of input features in the training set into an observation matrix X of size $M \times N_f$ and define the length-M vectors of training labels $\vec{\mathbf{y}}$, training label noise $\vec{\boldsymbol{\varepsilon}}$, and label predictions for the training set $\hat{\mathbf{y}}$. We also organize the vectors of hidden features evaluated on the input features of the training set, $\{\vec{\mathbf{z}}(\vec{\mathbf{x}}_a)\}_{a=1}^M$, into the rows of a hidden feature matrix Z of size $M \times N_p$.

Given a set of training data $\mathcal{D}$, we solve for the optimal values of the fit parameters $\hat{\mathbf{w}}$ by minimizing the standard loss function used for ridge regression,

$$L(\hat{\mathbf{w}}; \mathcal{D}) = \frac{1}{2}\|\Delta\vec{\mathbf{y}}\|^2 + \frac{\lambda}{2}\|\hat{\mathbf{w}}\|^2, \tag{5}$$

where the notation $\|\cdot\|$ indicates an L_2 norm and $\Delta\vec{\mathbf{y}} = \vec{\mathbf{y}} - \hat{\mathbf{y}}$ is the vector of residual label errors for the training set. The first term is simply the mean squared error between the true labels and their predicted values, while the second term imposes standard L_2 regularization with regularization parameter λ. We will often work in the "ridge-less limit" where we take the limit $\lambda \to 0$.

E. Model evaluation

To evaluate each model's prediction accuracy, we measure the training and test (generalization) errors. We define the training error as the mean squared residual label error of the training data,

$$\mathcal{E}_{\text{train}} = \frac{1}{M}\|\Delta\vec{\mathbf{y}}\|^2. \tag{6}$$

We define the interpolation threshold as the model complexity at which the training error becomes exactly zero (in the thermodynamic and ridge-less limits). Analogously, we define the test error as the mean squared error evaluated on a test data set, $\mathcal{D}' = \{(y_a', \vec{\mathbf{x}}_a')\}_{a=1}^{M'}$, composed of M' new data points drawn independently from the same data distribution as the training data,

$$\mathcal{E}_{\text{test}} = \frac{1}{M'}\|\Delta\vec{\mathbf{y}}'\|^2, \tag{7}$$

where $\Delta\vec{\mathbf{y}}' = \vec{\mathbf{y}}' - \hat{\mathbf{y}}'$ is a length-M' vector of residual label errors between the vector of labels $\vec{\mathbf{y}}'$ and their predicted values $\hat{\mathbf{y}}'$ for the test set.

F. Exact solutions

To solve for unique optimal solution, we set the gradient with respect to the fit parameters to zero, giving us a set of N_p equations with N_p unknowns,

$$0 = \frac{\partial L(\hat{\mathbf{w}})}{\partial\hat{\mathbf{w}}} = -Z^T\Delta\vec{\mathbf{y}} + \lambda\hat{\mathbf{w}}. \tag{8}$$

Solving this set of equations results in a unique solution for the fit parameters,

$$\hat{\mathbf{w}} = \left[\lambda I_{N_p} + Z^TZ\right]^{-1}Z^T\vec{\mathbf{y}}. \tag{9}$$

For simplicity, we also take the ridge-less limit where λ is infinitesimally small ($\lambda \to 0$). While our calculations do provide exact solutions for finite λ, the solutions in the limits of small λ are much more insightful. In this limit, Eqs. (9) and (3) approximate to

$$\hat{\mathbf{w}} \approx Z^+\vec{\mathbf{y}}, \quad \hat{y}(\vec{\mathbf{x}}) \approx \vec{\mathbf{z}}(\vec{\mathbf{x}})\cdot Z^+\vec{\mathbf{y}}, \tag{10}$$

where $^+$ denotes a Moore-Penrose inverse or pseudoinverse.

G. Hessian matrix

We note that the solution for the fit parameters in Eq. (9) depends on the matrix Z^TZ, which we refer to in the ridge-less limit as the Hessian matrix. The matrix Z^TZ/M can be interpreted as an empirical covariance matrix of the hidden features sampled by the training set when the hidden features are centered. The authors of Ref. [50] showed that for ridge regression, the divergence of the test error at the interpolation threshold can be naturally understood in terms of the spectrum of the Hessian. Inspired by this observation, we also explore the relationship between the eigenvalues of the Hessian and the double-descent phenomenon in our more general setting.

To do so, we reproduce the known eigenvalue distribution for the Hessian for both models. We note that in linear regression (no basis functions), the Hessian is simply a Wishart matrix whose eigenvalues follow the Marchenko-Pastur distribution [51]. The eigenvalue spectrum for the Hessian for the random nonlinear features model was explored in Ref. [52]. For both models, we provide an alternative derivation of these spectra using the zero-temperature cavity method, allowing us to directly relate the eigenvalues of the Hessian matrix to the double-descent phenomenon.

H. Derivation of closed-form solutions

While the expressions in the previous section are quite general, they hide much of the complexity of the problem and are difficult to analyze carefully. For this reason, we make use of the zero-temperature cavity method to find closed-form solutions for all quantities of interest. The zero-temperature cavity method has a long history in the physics of disordered systems and statistical learning theory and has been used to analyze the Hopfield model [53] and more recently, compressed sensing [20,54]. The cavity method is an alternative to the more commonly used replica method or analyses based on random matrix theory.

Like the replica method, finding closed-form solutions requires some additional assumptions. In particular, we assume that the solutions satisfy a replica symmetric ansatz (an assumption we confirm numerically by showing remarkable agreement between our analytic results and simulations). Furthermore, we work in the thermodynamic limit, where $N_f, M, N_p \to \infty$ and keep terms to leading order in these quantities. Our results are exact under these assumptions.

To apply the zero-temperature cavity method, we start by defining the ratio of the number of input features to training data points $\alpha_f = N_f/M$ and the ratio of fit parameters to training data points $\alpha_p = N_p/M$. Next, we take the thermodynamic limit $N_f, M, N_p \to \infty$, while keeping the ratios α_f and α_p finite. The essence of the cavity method is to expand the solutions of Eq. (8) with $M+1$ data points, N_f+1 features and N_p+1 parameters about the solutions where one quantity of each type has been removed: $(M+1, N_f+1, N_p+1) \to (M, N_f, N_p)$. These two solutions are then related using generalized susceptibilities. The result is a set of algebraic self-consistency equations that can be solved for the distributions of the removed quantities. The central limit theorem then allows us to approximate any quantity defined as a sum over a large number of random variables (e.g., the training and test errors) using just distributions for the removed quantities. Furthermore, using the procedure described in Ref. [55], we use the susceptibilities resulting from the cavity method to reproduce the known closed-form solutions for the eigenvalues spectra of the Hessian matrices for both models. We refer the reader to Ref. [5] for further details on these calculations.

III. BIAS-VARIANCE DECOMPOSITION

The bias-variance decomposition separates test error into components stemming from three distinct sources: bias, variance, and noise. Informally, bias captures a model's tendency to underfit, reflecting the erroneous assumptions made by a model that limit its ability to fully express the relationships underlying the data. On the other hand, variance captures a model's tendency to overfit, capturing characteristics of the training set that are not a reflection of the data's true relationships, but rather a by-product of random sampling (e.g., noise). As a result, a model with high variance may not generalize well to other data sets drawn from the same data distribution. Noise simply refers to an irreducible error inherent in generating a set of test data (i.e., the label noise in the test set).

Formally, bias represents the extent to which the label predictions $\hat{y}(\vec{\mathbf{x}})$ differs from the true function underlying the data distribution $y^*(\vec{\mathbf{x}})$ when evaluated on an arbitrary test data point $\vec{\mathbf{x}}$ and averaged over all possible training sets $\mathcal{D}$ [56,57],

$$\mathrm{Bias}[\hat{y}(\vec{\mathbf{x}})] = \mathrm{E}_{\mathcal{D}}[\hat{y}(\vec{\mathbf{x}})] - y^*(\vec{\mathbf{x}}). \tag{11}$$

Likewise, variance formally measures the extent to which solutions of $\hat{y}(\vec{\mathbf{x}})$ for individual training sets $\mathcal{D}$ vary around the average [56,57],

$$\mathrm{Var}[\hat{y}(\vec{\mathbf{x}})] = \mathrm{E}_{\mathcal{D}}\left[\hat{y}^2(\vec{\mathbf{x}})\right] - \mathrm{E}_{\mathcal{D}}[\hat{y}(\vec{\mathbf{x}})]^2. \tag{12}$$

Finally, the noise is simply the mean squared label noise associated with an arbitrary test data point $\vec{\mathbf{x}}$,

$$\mathrm{Noise} = \mathrm{E}[\varepsilon^2] = \sigma_\varepsilon^2. \tag{13}$$

The standard bias-variance decomposition relates these three quantities to the test error (averaged over all possible training sets $\mathcal{D}$). In addition, we must take into account the fact that the test error is evaluated on M' test data points, while the bias and variance only consider a single test point. Since each test point is drawn from the same distribution, averaging the test error over all possible test sets $\mathcal{D}'$ is equivalent to averaging the bias and variance over the point $\vec{\mathbf{x}}$. This gives us the canonical bias-variance decomposition [56,57],

$$\mathrm{E}_{\mathcal{D}',\mathcal{D}}[\mathcal{E}_{\mathrm{test}}] = \mathrm{E}_{\vec{\mathbf{x}}}[\mathrm{Bias}^2[\hat{y}(\vec{\mathbf{x}})]] + \mathrm{E}_{\vec{\mathbf{x}}}[\mathrm{Var}[\hat{y}(\vec{\mathbf{x}})]] + \sigma_\varepsilon^2. \tag{14}$$

In this work, we also consider other sources of randomness (e.g., $\vec{\boldsymbol{\beta}}$ and W). To incorporate these random variables, we define the more general ensemble-averaged squared bias and variance, respectively, as

$$\begin{aligned}\langle \mathrm{Bias}^2[\hat{y}]\rangle &= \mathrm{E}_{\vec{\boldsymbol{\beta}},W,\vec{\mathbf{x}}}[\mathrm{Bias}[\hat{y}(\vec{\mathbf{x}})]^2]\\ &= \mathrm{E}_{\vec{\boldsymbol{\beta}},W,\vec{\mathbf{x}}}[(\mathrm{E}_{X,\vec{\boldsymbol{\varepsilon}}}[\hat{y}(\vec{\mathbf{x}})] - y^*(\vec{\mathbf{x}}))^2],\end{aligned} \tag{15}$$

$$\begin{aligned}\langle \mathrm{Var}[\hat{y}]\rangle &= \mathrm{E}_{\vec{\boldsymbol{\beta}},W,\vec{\mathbf{x}}}[\mathrm{Var}[\hat{y}(\vec{\mathbf{x}})]]\\ &= \mathrm{E}_{\vec{\boldsymbol{\beta}},W,\vec{\mathbf{x}}}[\mathrm{E}_{X,\vec{\boldsymbol{\varepsilon}}}[\hat{y}^2(\vec{\mathbf{x}})] - \mathrm{E}_{X,\vec{\boldsymbol{\varepsilon}}}[\hat{y}(\vec{\mathbf{x}})]^2],\end{aligned} \tag{16}$$

where we have explicitly included all random variables considered in this work. All analytic expressions we report are ensemble-averaged (denoted by angle brackets $\langle\cdot\rangle$) and utilize the ensemble-averaged bias-variance decomposition of the test error,

$$\langle \mathcal{E}_{\mathrm{test}}\rangle = \langle \mathrm{Bias}^2[\hat{y}]\rangle + \langle \mathrm{Var}[\hat{y}]\rangle + \sigma_\varepsilon^2. \tag{17}$$

By fixing any parameters that do not pertain to the random sampling process of the test or training data (in this case, $\vec{\boldsymbol{\beta}}$ and W), this formula properly reduces to the canonical bias-variance decomposition in Eq. (14).

IV. RESULTS

In this section, we provide analytic results for the training error, test error, bias, and variance, along with partial comparisons to numerical results. We limit ourselves to simply discussing major features of our closed-form solutions, deferring a discussion of the implications of these results to the next section. Analytic derivations and complete comparisons to numerical results are left to Ref. [5].

A. General solutions

We first report the forms of the solutions for arbitrary student and teacher models. We find that the training error, test error, bias, and variance take the general forms

$$\langle \mathcal{E}_{\text{train}} \rangle = \langle \Delta y^2 \rangle, \tag{18}$$

$$\langle \mathcal{E}_{\text{test}} \rangle = \sigma_X^2 \langle \Delta\beta^2 \rangle + \sigma_{\delta z}^2 \langle \hat{w} \rangle^2 + \sigma_{\delta y^*}^2 + \sigma_\varepsilon^2, \tag{19}$$

$$\langle \text{Bias}^2[\hat{y}] \rangle = \sigma_X^2 \langle \Delta\beta_1 \Delta\beta_2 \rangle + \sigma_{\delta z}^2 \langle \hat{w}_1 \hat{w}_2 \rangle + \sigma_{\delta y^*}^2, \tag{20}$$

$$\langle \text{Var}[\hat{y}] \rangle = \sigma_X^2 [\langle \Delta\beta^2 \rangle - \langle \Delta\beta_1 \Delta\beta_2 \rangle] + \sigma_{\delta z}^2 [\langle \hat{w}^2 \rangle - \langle \hat{w}_1 \hat{w}_2 \rangle], \tag{21}$$

which depend on five key ensemble-averaged quantities: $\langle \Delta y^2 \rangle$, $\langle \hat{w}^2 \rangle$, $\langle \Delta\beta^2 \rangle$, $\langle \hat{w}_1 \hat{w}_2 \rangle$, and $\langle \Delta\beta_1 \Delta\beta_2 \rangle$ (see Sec. S1E of Ref. [5] for detailed derivation). The first two quantities are the average of the squared training label errors $\langle \Delta y^2 \rangle$ and the average of the squared fit parameters $\langle \hat{w}^2 \rangle$. The third quantity $\langle \Delta\beta^2 \rangle$ measures a model's accuracy in identifying the ground truth parameters $\vec{\boldsymbol{\beta}}$. To see this, we note that an estimate of the ground truth parameters for each model can be constructed from the fit parameters via the expression $\hat{\boldsymbol{\beta}} \equiv W\hat{\mathbf{w}}$, with residual parameter errors $\Delta\vec{\boldsymbol{\beta}} \equiv \vec{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}$. Given these definitions, $\langle \Delta\beta^2 \rangle$ is then the average of the squared residual parameter errors. Finally, the quantities $\langle \hat{w}_1 \hat{w}_2 \rangle$ and $\langle \Delta\beta_1 \Delta\beta_2 \rangle$ measure the average covariances of a pair fit parameters or residual parameter errors, respectively, that have the same index but derive from models trained on different training sets drawn independently from the same data distribution.

In addition to these five ensemble averages, the above expressions also depend on the quantities $\sigma_{\delta y^*}^2$ and $\sigma_{\delta z}^2$ which characterize the degree of nonlinearity of the labels and hidden features, respectively. To define these quantities, we note that the nonlinear labels and hidden features considered in this work can each be decomposed into linear and nonlinear parts.

First, we decompose the true labels in Eq. (2) into two components that are statistically independent with respect to the distribution of input features, allowing us to express the teacher model as

$$y(\vec{\mathbf{x}}) = \vec{\mathbf{x}} \cdot \vec{\boldsymbol{\beta}} + \delta y_{\text{NL}}(\vec{\mathbf{x}}) + \varepsilon. \tag{22}$$

The first term $\vec{\mathbf{x}} \cdot \vec{\boldsymbol{\beta}}$ captures the linear correlations between the input features $\vec{\mathbf{x}}$ and the true labels y^* via the ground truth parameter $\vec{\boldsymbol{\beta}}$, while the second term $\delta y_{\text{NL}}^*(\vec{\mathbf{x}})$ captures the nonlinear behavior of y^* [defined as $\delta y_{\text{NL}}^*(\vec{\mathbf{x}}) \equiv y^*(\vec{\mathbf{x}}) - \vec{\mathbf{x}} \cdot \vec{\boldsymbol{\beta}}$]. The nonlinear component has zero mean (since the labels are centered with zero mean) and we define its variance as $\sigma_{\delta y^*}^2$. Previously, this decomposition was implemented by noting that $\delta y_{\text{NL}}^*(\vec{\mathbf{x}})$ behaves like an independent Gaussian process [28,40]. Here, we note that this approximation follows naturally in the thermodynamic limit from the relationship $\vec{\boldsymbol{\beta}} = \Sigma_{\vec{\mathbf{x}}}^{-1}\text{Cov}_{\vec{\mathbf{x}}}[\vec{\mathbf{x}}, y^*(\vec{\mathbf{x}})]$ where $\Sigma_{\vec{\mathbf{x}}} \equiv \text{Cov}_{\vec{\mathbf{x}}}[\vec{\mathbf{x}}, \vec{\mathbf{x}}^T]$ is the covariance matrix of the input features (see Sec. S1D of Ref. [5]).

We also decompose the hidden features in Eq. (4) into three statistically independent components with respect to the distribution of input features,

$$\vec{\mathbf{z}}(\vec{\mathbf{x}}) = \frac{\mu_z}{\sqrt{N_p}}\vec{\mathbf{1}} + W^T\vec{\mathbf{x}} + \delta\vec{\mathbf{z}}_{\text{NL}}(\vec{\mathbf{x}}). \tag{23}$$

The first term $\mu_Z/\sqrt{N_p}\vec{\mathbf{1}}$ is the mean of each hidden feature where $\vec{\mathbf{1}}$ is a length-N_p vector of ones. Analogously to the label decomposition, the second term $W^T\vec{\mathbf{x}}$ captures the linear correlations between the input features $\vec{\mathbf{x}}$ and the hidden features $\vec{\mathbf{z}}(\vec{\mathbf{x}})$ via the matrix of parameters W, while the third term $\delta\vec{\mathbf{z}}_{\text{NL}}(\vec{\mathbf{x}})$ captures the remaining nonlinear behavior of $\vec{\mathbf{z}}(\vec{\mathbf{x}})$ [defined as $\delta\vec{\mathbf{z}}_{\text{NL}}(\vec{\mathbf{x}}) \equiv \vec{\mathbf{z}}(\vec{\mathbf{x}}) - \mu_z\vec{\mathbf{1}}/\sqrt{N_p} - W^T\vec{\mathbf{x}}$]. The nonlinear component has zero mean and we define its total variance as $\sigma_{\delta z}^2$. Like the nonlinear teacher model, it was previously observed that the nonlinear component of the hidden features behaves like an independent Gaussian process [58], and this decomposition has since been used as a common trick to obtain closed-form solutions for nonlinear models. Here, we again note that this approximation follows naturally in the thermodynamic limit from the relationship $W = \Sigma_{\vec{\mathbf{x}}}^{-1}\text{Cov}_{\vec{\mathbf{x}}}[\vec{\mathbf{x}}, \vec{\mathbf{z}}(\vec{\mathbf{x}})^T]$ (see Sec. S1D of Ref. [5]).

We find the variance of the nonlinear components of the labels and hidden features, respectively, to be

$$\sigma_{\delta y^*}^2 = \sigma_\beta^2 \sigma_X^2 \Delta f, \quad \Delta f = \frac{\langle f^2 \rangle - \langle f' \rangle^2}{\langle f' \rangle^2}, \tag{24}$$

$$\sigma_{\delta z}^2 = \sigma_W^2 \sigma_X^2 \Delta\varphi, \quad \Delta\varphi = \frac{\langle \varphi^2 \rangle - \langle \varphi \rangle^2 - \langle \varphi' \rangle^2}{\langle \varphi' \rangle^2}, \tag{25}$$

where the quantities $\langle f^2 \rangle$, $\langle f' \rangle$, $\langle \varphi^2 \rangle$, $\langle \varphi \rangle$, and $\langle \varphi' \rangle$ are integrals of the form

$$\langle g \rangle = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dh e^{-\frac{1}{2}h^2} g(h) \tag{26}$$

with derivatives indicated via prime notation [e.g., $f' = df(h)/dh$]. The differences Δf and $\Delta\varphi$ measure the ratio of the variances of each nonlinear component to its linear counterparts and go to zero in the linear limit. For ReLU activation, $\varphi(h) = \max(h, 0)$, we find $\langle \varphi^2 \rangle = 1/2$, $\langle \varphi \rangle = 1/\sqrt{2\pi}$, and $\langle \varphi' \rangle = 1/2$, resulting in $\Delta\varphi = 1 - 2/\pi$.

We derive Eqs. (19)–(21) by decomposing the labels and hidden features of the test data (see Sec. S1E in Ref. [5]). As a result, we can identify what elements of the test data lead to each term in these expressions. We observe that terms proportional to σ_X^2 capture error arising from the linear components of the test data's labels and hidden features. More precisely, these terms measure error due to the mismatch between the linear components of the test labels and a model's predictions based solely on the linear components of the test data's hidden features $\vec{\mathbf{x}} \cdot \Delta\beta = \vec{\mathbf{x}} \cdot \vec{\boldsymbol{\beta}} - \vec{\mathbf{x}} \cdot W\hat{\mathbf{w}}$. In contrast, terms proportional to $\sigma^2_{\delta y^*}$ and $\sigma^2_{\delta z}$ represent errors due to the nonlinear components of the test data's labels and hidden features, respectively. Since the model is linear in the fit parameters, it cannot fully capture nonlinear relationships in the test data that deviate from those observed in the training set.

Finally, we note that the decomposition of the labels in Eq. (22) suggests that each key quantity, along with total train error, test error, bias, and variance, decompose into contributions from the different parts of the training labels. Since each term is statistically independent, the contribution of each is proportional to its respective variance, allowing us to simply read off the sources of each type of error from our analytic results. In particular, the linear and nonlinear components of the labels give rise to terms proportional to $\sigma^2_\beta\sigma^2_X$ and $\sigma^2_{\delta y^*}$, respectively, while the label noise gives rise to terms proportional to σ^2_ε.

B. Linear regression

Here, we present results for linear regression (no basis functions). Generally, our solutions are most naturally expressed in terms of $\alpha_f = N_f/M$, the ratio of input features to training data points, and $\alpha_p = N_p/M$, the ratio of fit parameters to training data points. However, in this case, the input and hidden features coincide ($N_f = N_p$), so all expressions depend only on α_f. The ensemble-averaged training error, test error, bias, and variance for linear regression are

$$\langle \mathcal{E}_{\text{train}} \rangle = \begin{cases} \left(\sigma^2_\varepsilon + \sigma^2_{\delta y^*}\right)(1-\alpha_f) & \text{if} \quad N_f < M \\ 0 & \text{if} \quad N_f > M \end{cases}, \tag{27}$$

$$\langle \mathcal{E}_{\text{test}} \rangle = \begin{cases} \left(\sigma^2_\varepsilon + \sigma^2_{\delta y^*}\right)\frac{1}{(1-\alpha_f)} & \text{if} \quad N_f < M \\ \sigma^2_\beta\sigma^2_X\frac{(\alpha_f-1)}{\alpha_f} + \left(\sigma^2_\varepsilon + \sigma^2_{\delta y^*}\right)\frac{\alpha_f}{(\alpha_f-1)} & \text{if} \quad N_f > M \end{cases}, \tag{28}$$

$$\langle \text{Bias}^2[\hat{y}] \rangle = \begin{cases} \sigma^2_{\delta y^*} & \text{if} \quad N_f < M \\ \sigma^2_\beta\sigma^2_X\frac{(\alpha_f-1)^2}{\alpha_f^2} + \sigma^2_{\delta y^*} & \text{if} \quad N_f > M \end{cases}, \tag{29}$$

$$\langle \text{Var}[\hat{y}] \rangle = \begin{cases} \left(\sigma^2_\varepsilon + \sigma^2_{\delta y^*}\right)\frac{\alpha_f}{(1-\alpha_f)} & \text{if} \quad N_f < M \\ \sigma^2_\beta\sigma^2_X\frac{(\alpha_f-1)}{\alpha_f^2} + \left(\sigma^2_\varepsilon + \sigma^2_{\delta y^*}\right)\frac{1}{(\alpha_f-1)} & \text{if} \quad N_f > M \end{cases}. \tag{30}$$

In writing these expressions, we have taken the ridge-less limit, $\lambda \to 0$ (when a quantity is reported as zero, leading terms of order λ^2 are reported in Sec. S1F of the Ref. [5]).

In Fig. 2(a), we plot the expressions for the training and test error in Eqs. (27) and (28) with comparisons to numerical results for a linear teacher model $\sigma^2_{\delta y^*} = 0$. We find that the model's behavior falls into two broad regimes, depending on whether $\alpha_f > 1$ or $\alpha_f < 1$ (or equivalently, $\alpha_p > 1$ or $\alpha_p < 1$). In Fig. 2(a), we observe that below $\alpha_f = 1$, the training error is finite, decreasing monotonically as α_f increases until reaching zero at $\alpha_f = 1$. Beyond this threshold, the addition of extra features/parameters has no effect and the training error remains pinned at zero. Thus, $\alpha_f = 1$ corresponds to the interpolation threshold, separating the regions where the model has zero and nonzero training error, i.e., the under and overparameterized regimes. At the interpolation threshold, the test error diverges [Fig. 2(a)], indicative of a phase transition based on the divergence of the corresponding susceptibilities in the cavity equations (see Sec. V E and Sec. S1F of Ref. [5]).

The bias and variance, reported in Eqs. (29) and (30), are plotted in Fig. 2(b). When $\alpha_f \leqslant 1$, the bias is zero for a linear teacher model. This can be understood by noting that the teacher and student models match in this case and there are more data points than parameters (i.e., we are working in a regime where intuitions for classical statistics apply). In this regime, the variance increases monotonically as α_f is increased, diverging at the interpolation threshold as the model succumbs to overfitting. However, when $\alpha_f > 1$, the variance exhibits the opposite behavior, decreasing monotonically as α_f is increased. The bias, on the other hand, *increases* monotonically towards the limit $\sigma^2_\beta\sigma^2_X + \sigma^2_{\delta y^*}$ as α_f goes to infinity. Consequently, the test error in the overparameterized regime is characterized by a surprising "inverted bias-variance trade-off" where the bias increases with model complexity while the variance decreases.

In the solutions for the training error, test error, and variance, we observe that the error due to the nonlinear label components (proportional to $\sigma^2_{\delta y^*}$) always appears as an additive component to the errors stemming from the label noise (proportional to σ^2_ε). However, unlike the label noise, the nonlinear label variance $\sigma^2_{\delta y^*}$ also appears in the bias as an additional constant irreducible term that arises as a result of attempting to fit a nonlinear data distribution with a model that is linear in the fit parameters.

Finally, in Fig. 2(c), we report the minimum nonzero eigenvalue σ^2_{min} of the Hessian matrix Z^TZ, with examples of the full eigenvalue spectrum shown in Figs. 2(i)–2(iii). Since $Z^TZ = X^TX$ for our model of linear regression, the eigenvalue spectrum is simply the Marchenko-Pastur distribution (see Sec. S2A of Ref. [5] for derivation). Importantly, we find the interpolation threshold $\alpha_f = 1$ coincides with the point at which σ^2_{min} goes to zero. In the underparameterized regime, there is a finite gap in the eigenvalue spectrum, with no small eigenvalues. In the overparameterized, there is also a finite gap, but instead between the bulk of the spectrum

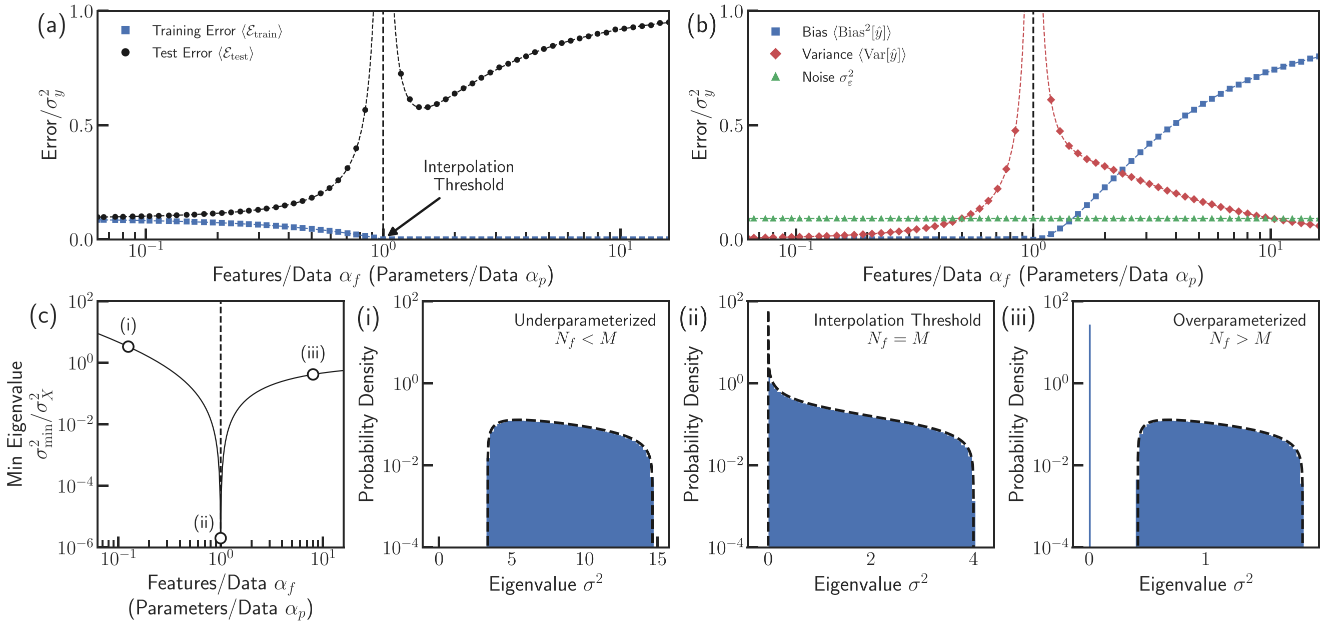

FIG. 2. Linear regression (no basis functions). Analytic solutions for the ensemble-averaged (a) training error (blue squares) and test error (black circles), and (b) bias-variance decomposition of test error with contributions from the squared bias (blue squares), variance (red squares), and test set label noise (green triangles), plotted as a function of $\alpha_f = N_f/M$ (or equivalently, $\alpha_p = N_p/M$). Analytic solutions are indicated as dashed lines with numerical results shown as points with small error bars indicating the error on the mean. In each panel, a black dashed vertical line marks the interpolation threshold $\alpha_f = 1$. (c) Analytic solution for the minimum eigenvalue σ^2_{min} of the Hessian matrix Z^TZ. Examples of the eigenvalue distributions are shown (i) in the underparameterized regime with $\alpha_f = 1/8$, (ii) at the interpolation threshold, $\alpha_f = 1$, and (iii) in the overparameterized regime with $\alpha_f = 8$. Analytic solutions for the distributions are depicted as black dashed curves with numerical results shown as blue histograms. See Sec. S4 of Ref. [5] for additional simulation details.

and a buildup of eigenvalues at exactly zero. We discuss the implications of these findings later in Sec. V B.

C. Random nonlinear features model

Unlike the solutions for linear regression, the analytic expressions for the random nonlinear features model are not so simple, so we defer these expressions to the Appendix. In Fig. 3(a) and 3(b), we plot the training error, test error, bias, and variance as a function of $\alpha_p = N_p/M$ for fixed $\alpha_f < 1$ (more data points M than input features N_f), while in Figs. 3(c)–3(f), we plot all quantities as a function of both α_p and α_f. In all plots, we depict the special case of a linear teacher model $\sigma^2_{\delta y^*} = 0$ and ReLU activation $\varphi(h) = \max(h, 0)$. Analogously to linear regression, the nonlinear model has two distinct regimes separated by the line $\alpha_p = 1$. In Fig. 3(c), we find that the training error is finite when $\alpha_p < 1$ and goes to zero when $\alpha_p \geqslant 1$, marking the boundary $\alpha_p = 1$ as the interpolation threshold. Figure 3(d) shows that the test error diverges at each point along this boundary and no longer diverges when $\alpha_f = 1$ as in the linear case [Fig. 2(a)]. However, we do still find that the divergence in the test error is associated with a phase transition indicated by diverging susceptibilities (see Sec. V E and Appendix).

In addition, the test error only displays a small qualitative difference between the regimes where $\alpha_f < 1$ and $\alpha_f > 1$. We find that the test error only shows a canonical double-descent behavior when $\alpha_f < 1$ [Fig. 3(d)]. As in linear regression, the variance [Fig. 3(f)], accounts for the divergence of the test error at the phase boundaries, while the bias [Fig. 3(e)] remains finite, decreasing monotonically for all α_p. However, unlike linear regression, the bias of the nonlinear model never reaches zero, even for a linear teacher model. Furthermore, the closed-form solutions show that the nonlinear components of the labels $\sigma^2_{\delta y^*}$ contribute in the same way as in the two linear models, adding a small constant irreducible bias [Eq. (20)], along with an additive component to the label noise (see Appendix).

Finally, in Fig. 3(g), we report the minimum nonzero eigenvalue σ^2_{min} of the Hessian matrix Z^TZ as a function of both α_p and α_f (see Sec. S2B of Ref. [5] for derivation of analytic results). We find that σ^2_{min} approaches zero along the entire interpolation boundary $\alpha_p = 1$. In Figs. 3(i)–3(iii), we show examples of the full eigenvalue spectrum for $\alpha_f < 1$ for the under and overparameterized regimes, along with the interpolation threshold. We find that the spectrum in the underparameterized regime displays a finite gap that goes to zero near the interpolation threshold. In the overparameterized regime, we find that although the gap between the buildup of eigenvalues at zero and the nonzero eigenvalues is much smaller, it is still finite. Interestingly, we also find that additional gaps can appear in the eigenvalue distribution between sets of finite-valued eigenvalues, which likely reflects the fact that ReLU activation functions result in a large fraction of zero-valued entries in Z.

V. UNDERSTANDING BIAS AND VARIANCE IN OVERPARAMETERIZED MODELS

Having presented our analytic results, we now discuss the implications of our calculations for understanding bias and

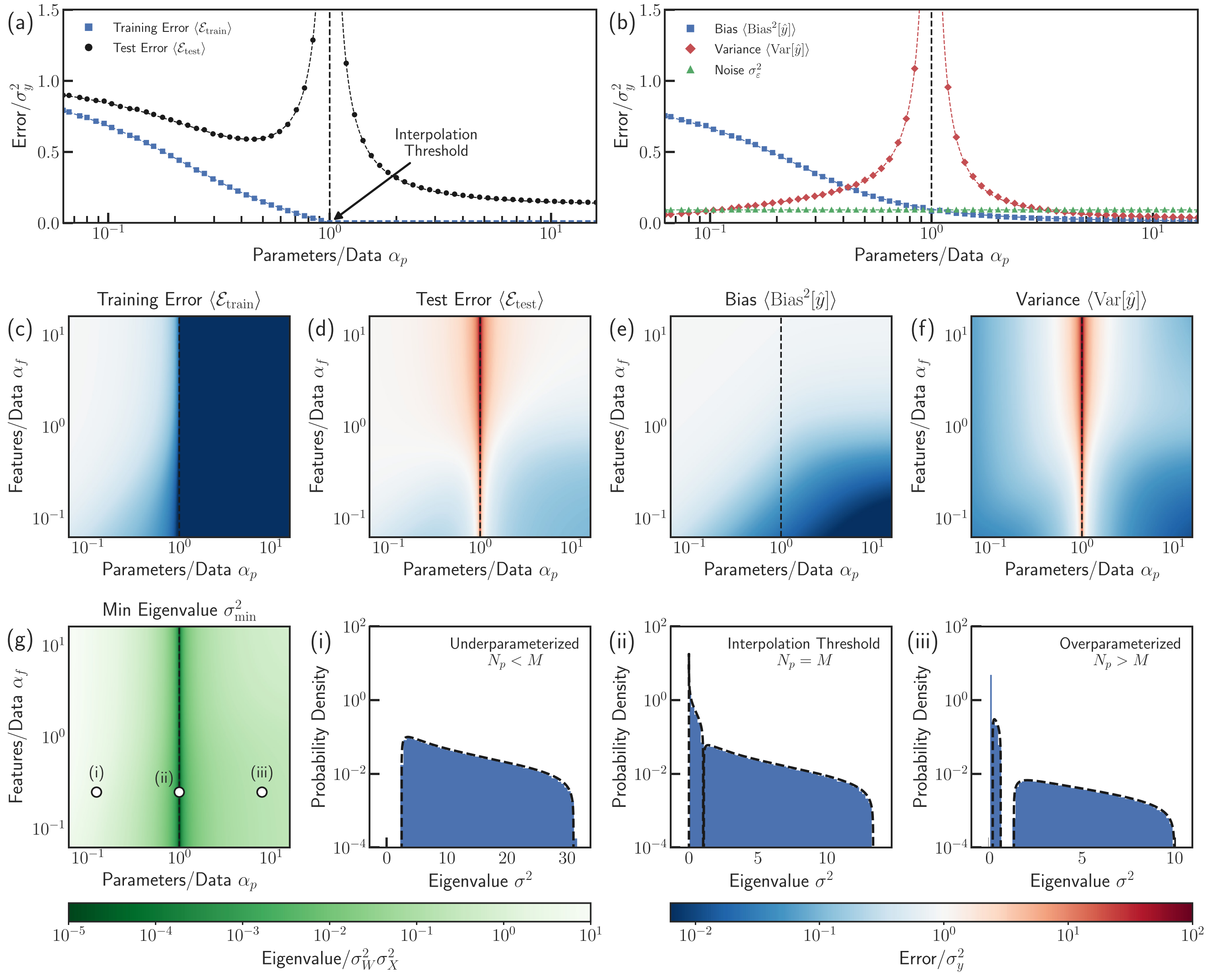

FIG. 3. Random nonlinear features model (two-layer neural network). Analytic solutions for the ensemble-averaged (a) training error (blue squares) and test error (black circles) and (b) bias-variance decomposition of test error with contributions from the squared bias (blue squares), variance (red squares), and test set label noise (green triangles), plotted as a function of $\alpha_p = N_p/M$ for fixed $\alpha_f = N_f/M = 1/4$. Analytic solutions are indicated as dashed lines with numerical results shown as points. Analytic solutions as a function of both α_p and α_f are also shown for the ensemble-averaged (c) training error, (d) test error, (e) squared bias, and (f) variance. In all panels, a black dashed line marks the boundary between the under and overparameterized regimes at $\alpha_p = 1$. (g) Analytic solution for the minimum eigenvalue $\sigma^2_{\rm min}$ of the Hessian matrix Z^TZ. Examples of the eigenvalue distributions are shown (i) in the underparameterized regime with $\alpha_p = 1/8$, (ii) at the interpolation threshold, $\alpha_p = 1$, and (iii) in the overparameterized regime with $\alpha_p = 8$, all for $\alpha_p = 1/4$. Analytic solutions for the distributions are shown as blacked dashed curves with numerical results shown as blue histograms. See Sec. S4 of Ref. [5] for additional simulation details.

variance in a more general setting. Our discussion emphasizes the qualitatively new phenomena that are present in overparameterized models.

A. Two sources of bias: imperfect models and incomplete exploration of features

Traditionally, bias is viewed as a symptom of a model making incorrect assumptions about the data distribution (a mismatch between the teacher and student models). However, our calculations show that this description of the origin of bias is incomplete. A striking feature of our results is that overparameterized models can be biased even if our statistical models are expressive enough to fully capture all relationships underlying the data. In fact, linear regression shows us that one can have a nonzero bias even if the student and teacher models are identical [e.g., $f(h) = h$]. Even when the student and teacher model are the same, the bias is nonzero if there are more input features N_f than data points M [see $\alpha_f > 1$ region of Fig. 2(b)].

To better understand this phenomenon, it is helpful to think of the input features as spanning an N_f-dimensional space. The training data can be embedded in this N_f-dimensional input feature space by considering the eigenvectors, or principal components, of the empirical covariance matrix of input features X^TX/M, with X defined as the $M \times N_f$ design matrix whose rows correspond to training data points and columns to input features (see Sec. II). When there are more data points than input features ($M > N_f$), the training data will typically span the entire N_f-dimensional input feature space (i.e., the principal components of X^TX generically span all of input feature space). In contrast, when there are fewer training data points than input features ($M < N_f$), the training data will typically span only a fraction of the entire input feature space (i.e., the principal components will span a subspace of the full N_f-dimensional input feature space). For this reason, when $M < N_f$ the model is "blind" to data that varies along these unsampled directions. Consequently, any predictions the model makes about these directions will reflect assumptions implicitly or explicitly built into the model rather than relationships learned from the training data set. This can result in a nonzero bias even when the teacher and student models are identical.

In the random nonlinear features model, the nonlinear activation function makes it impossible to perfectly represent the input features via the hidden features, even for a linear teacher model. While increasing the number of hidden features does reduce bias, the nonlinear model is never able to perfectly capture the linear nature of the data distribution and is always biased. Similarly, neither model is able to express the nonlinear components of the labels for a nonlinear teacher model, resulting in a constant, irreducible bias in both cases.

Finally, we wish to point out that, unlike some previous studies, we find that the bias never diverges [17,19,40], including at the interpolation transition, nor does it reach a minimum at the transition, remaining constant into the overparameterized regime in the ridge-less limit [17,25,26,38]. Instead, we find that the bias remains finite and decreases monotonically, even in the absence of regularization.

B. Variance: overfitting stems from poorly sampled direction in space of feature

Variance measures the tendency of a model to overfit, or attribute too much significance to, aspects of the training data that do not generalize to other data sets. Even when all data is drawn from the same distribution, the predictions of a trained model can vary depending on the details of each particular training set. More specifically, a model may exhibit high variance when a direction in feature space is present in the training data, but not sampled well enough to reflect its true nature in the underlying data distribution. When presented with new data that has a significant contribution along this undersampled direction, the model is forced to extrapolate (often incorrectly) based on the little information it can glean from the training set.

In linear regression, the empirical variance along each principal direction is explicitly measured by its associated eigenvalue in the empirical covariance matrix X^TX/M. Generally, a model's variance will be dominated by the most poorly sampled principal direction, or minimum component $\hat{\mathbf{h}}_{\min}$, corresponding to the smallest nonzero eigenvalue $\sigma^2_{\min}$ of X^TX. The projection of an arbitrary data point $\vec{\mathbf{x}}$ onto $\hat{\mathbf{h}}_{\min}$ can be found by taking a dot product, $\hat{\mathbf{h}}_{\min} \cdot \vec{\mathbf{x}}$. The observed variance of $\hat{\mathbf{h}}_{\min} \cdot \vec{\mathbf{x}}$ for a given training set is $\sigma^2_{\text{train}} = \sigma^2_{\min}/M$. For comparison, we define the true, or expected, variance of $\hat{\mathbf{h}}_{\min} \cdot \vec{\mathbf{x}}$ for an average test set as σ^2_{test}, representing data points drawn from the full data distribution (see Sec. S3 of Ref. [5]).

The first row of Fig. 4 shows how observing a small variance along a particular direction sampled by the training data can lead to overfitting in linear regression. In Fig. 4(a), we plot the average ratio $\sigma_{\text{train}}/\sigma_{\text{test}}$ as a function of α_f for simulated data. In Figs. 4(a-i)–4(a-iii), we then plot the labels y versus $\hat{\mathbf{h}}_{\min} \cdot \vec{\mathbf{x}}$ for the training set (orange points) and an equally-sized test data set (blue points), representative of the full data distribution. In each panel, the relationship between the labels and $\hat{\mathbf{h}}_{\min} \cdot \vec{\mathbf{x}}$ as predicted by the model based on the training set is depicted as an orange line. For comparison, we also show the expected relationship for an average test set as a blue line, representing the true relationship underlying the data (see Sec. S3 of Ref. [5] for explicit formulas).

In Fig. 4(a-i), we see that when the model is underparameterized ($N_f < M$), the spread of the training set along the minimum component is comparable to that of the test set ($\sigma_{\text{train}}/\sigma_{\text{test}} = 0.7$). Because there are more data points than input features, many of the data points are likely to contain significant contributions from each direction, including the minimum component, corresponding to a finite gap in the corresponding eigenvalue distribution depicted in Fig. 2(c-i). As a result, the training set will provide the model with an accurate representation of the data distribution along this direction in feature space. In this case, we see that the model is able to closely approximate the true relationship in the data even in the presence of noise.

However, at the interpolation threshold when the number of input features equals the number of data points ($N_f = M$), Fig. 4(a-ii) shows that the spread of the training data points along the minimum component is very narrow compared to the test data ($\sigma_{\text{train}}/\sigma_{\text{test}} \approx 0.006$), while in Fig. 2(c-ii), we observe that the gap in the eigenvalue distribution disappears. In this case, the training set contains a very small, but insufficient, amount of information about the data distribution along this direction. This poor sampling causes the model to overfit the noise of the training set, resulting in a slope that is much larger than that of the true relationship. When presented with a new data point with a significant contribution along $\hat{\mathbf{h}}_{\min}$, the model will be forced to extrapolate beyond the narrow range of $\hat{\mathbf{h}}_{\min} \cdot \vec{\mathbf{x}}$ observed in the training set. This extrapolation will hamper the model's ability to generalize, leading to inaccurate predictions that are highly dependent on the precise details of the noise sampled by the training set.

Surprisingly, we find in Fig. 4(a-iii) that further increasing the number of features so that the model becomes overparameterized ($N_f > M$) actually *increases* the spread in the training data along the minimum component ($\sigma_{\text{train}}/\sigma_{\text{test}} \approx 1.9$), *reducing* the effects of overfitting. When there are more features than data points, each data point is likely to explore a never-before-seen combination of features. Naively, one would expect this to leave many of the directions poorly sampled. However, because the norm of each data point is approximately the same in the thermodynamic limit, the fact that each data point is different means that all are likely to make independent contributions of different sizes to the sampled directions, including $\hat{\mathbf{h}}_{\min}$. Even if this means only a single data point contributes to a particular component, this contribution must be of significant size for the data point to be independent of the rest. So while some directions are not represented in the training set at all, the ones that are present are typically

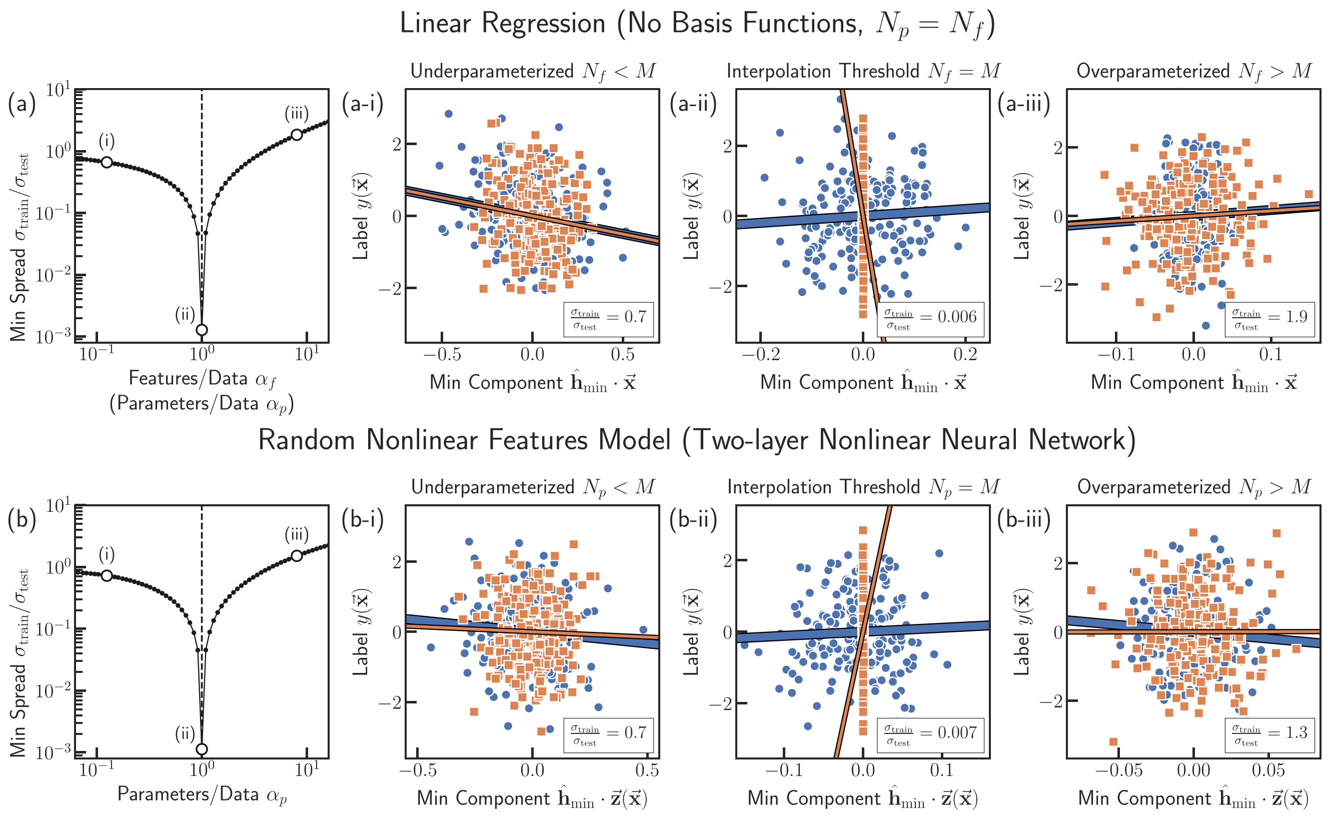

FIG. 4. Poorly sampled directions in space of features lead to overfitting. Demonstrations of this phenomenon are shown for (a) linear regression and (b) the random nonlinear features model. Columns (i), (ii), and (iii) correspond to models which are underparameterized, exactly at the interpolation threshold, or overparameterized, respectively. In each example, the relationship between the labels and the projection of their associated input or hidden features onto the minimum principal component $\hat{\mathbf{h}}_{\min}$ of $Z^T Z$ is depicted for a set of training data (orange squares) and a test set (blue circles). Orange lines indicate the relationship learned by a model from the training set, while the expected relationship for an average test set is shown as a blue line. In the left-most column, the spread (standard deviation) of an average training set along the x axis, $\sigma^2_{\mathrm{train}} = \sigma^2_{\min}/M$, is plotted relative to the spread that would be expected for an average test set, σ^2_{test}, for simulated data as a function of α_p. Smaller values are associated with lower prediction accuracy on out-of-sample data, coinciding with small eigenvalues in $Z^T Z$. All results are shown for a linear teacher model. See Ref. [5] for analytic derivations of learned and expected relationships and spreads along minimum principal components (Sec. S3), along with additional details of numerical simulations (Sec. S4).

well-represented by at least one—if not many—data points, providing a sufficient amount of signal (or spread) to reveal relationships in the underlying distribution. Consequently, the model is able to learn the true relationship between the labels and features, just as in the underparameterized case. We observe this phenomenon directly in the eigenvalue distribution in Fig. 2(c-iii), with a buildup of eigenvalues at exactly zero corresponding to unsampled directions accompanied by a finite gap separating these eigenvalues from the rest of the distribution. This is the underlying reason that the variance decreases with model complexity beyond the interpolation threshold. A similar observation was made in Ref. [18] in the context of ridge regression using methods from random matrix theory.

In the second row of Fig. 4, we demonstrate that the same patterns also lead to overfitting in the random nonlinear features model, indicating that the intuition gained from linear regression translates directly to more complex settings. In this case, the model can be interpreted as indirectly sampling the data distribution via the empirical covariance matrix of hidden features $Z^T Z/M$. We calculate the minimum component $\hat{\mathbf{h}}_{\min}$ as the principal component of $Z^T Z$ with the smallest eigenvalue. In Figs. 4(b-i)–4(b-iii), we plot the labels y versus the projection of each data point's hidden features $\vec{\mathbf{z}}$ onto this minimum component $\hat{\mathbf{h}}_{\min} \cdot \vec{\mathbf{z}}$, with the ratio $\sigma_{\mathrm{train}}/\sigma_{\mathrm{test}}$ shown in Fig. 4(b). In contrast to linear regression, we see that overfitting results from poorly sampling—or observing very limited spread along – a direction in the space of hidden features rather than input features. In the random nonlinear features model, overfitting is most pronounced when the number of hidden features matches the number of data points at the interpolation threshold ($N_p = M$), where the gap in the eigenvalue distribution at zero disappears [Fig. 3(g-ii)].

C. Biased models can interpret signal as noise

Typically, variance is attributed to overfitting inconsistencies in the labels due to noise in the training set. Indeed, we observe that the contribution to the variance due to noise is nonzero in each model. Surprisingly, we also find that overfitting can occur in the absence of noise when a model is biased. In each model, we observe a direct correspondence between each source of bias and a source of variance. In other words,

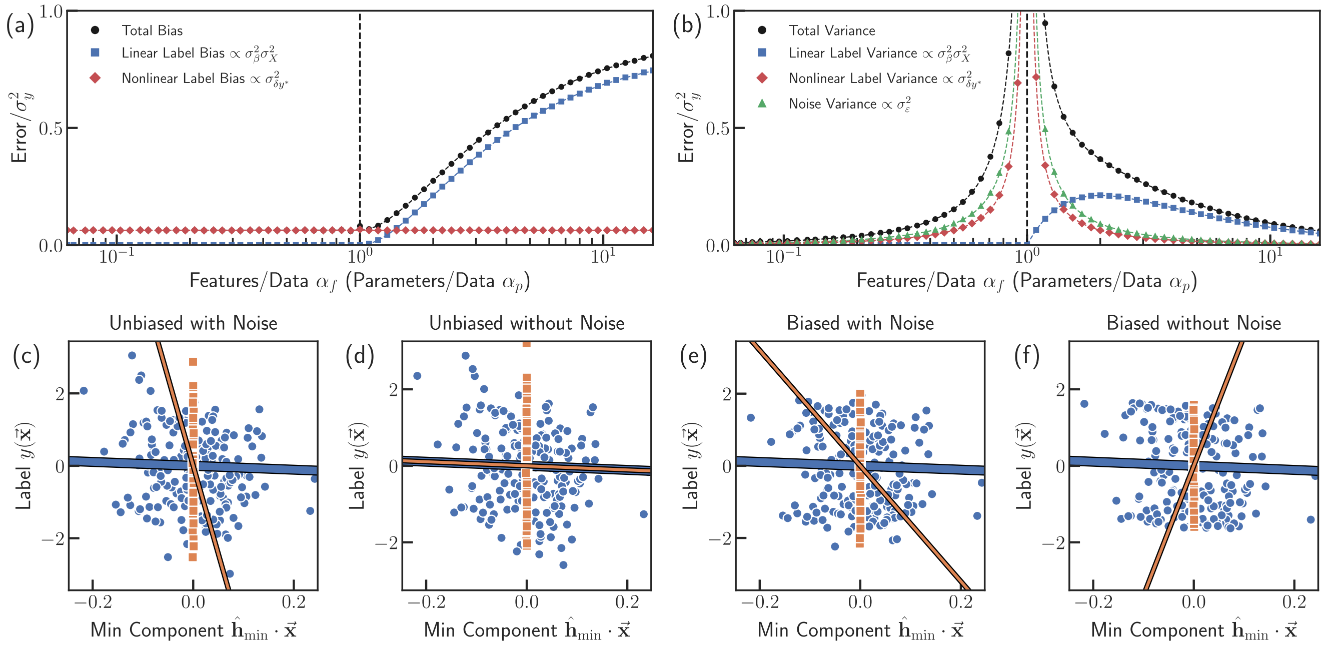

FIG. 5. Biased models can interpret signal as noise. (a) The total bias (black circles) with contributions from the linear label components (blue squares) and nonlinear label components (red diamonds), and the (b) the total variance (black circles) with contributions from the linear label components (blue squares), nonlinear label components (red diamonds), and the label noise (green triangles) are shown for linear regression with a nonlinear teacher model $f(h) = \tanh(h)$ [see Eq. (2)]. Analytic solutions are indicated as dashed lines with numerical results shown as points. Contributions from the linear label components, nonlinear label components, and label noise are found by identifying terms in the analytic solutions proportional to $\sigma_\beta^2\sigma_X^2$, $\sigma_{\delta y^*}^2$, and σ_ε^2, respectively. Each source of bias acts as effective noise, giving rise to a corresponding source of variance. The effects of this phenomenon on the relationships learned by a linear regression model are depicted at the interpolation threshold for an unbiased model with linear data, $f(h) = h$, (c) with noise and (d) without noise, and for a biased model with nonlinear data, $f(h) = \tanh(h)$, (e) with noise and (f) without noise. In each example, the relationship between the labels and the projection of their associated input features onto the minimum principal component $\hat{\mathbf{h}}_{\min}$ of X^TX is depicted for a set of training data (orange squares) and a test set (blue circles). Orange lines indicate the relationship learned by a model from the training set, while the expected relationship for an average test set is shown as a blue line. See Ref. [5] for analytic derivations of learned and expected relationships (Sec. S3), along with additional details of numerical simulations (Sec. S4).

in the absence of noise, the variance is zero only when the bias is zero.

To illustrate this, in Figs. 5(a) and 5(b), we plot the contributions to the bias and variance, respectively, from the different statistically independent components of the labels in Eq. (22) for our model of linear regression with a nonlinear teacher model of the form $f(h) = \tanh(h)$ [see Eq. (2)]. In this case, note that our model can never fully represent the true data distribution and hence will always be biased. We find that both contributions to the bias from the linear (blue) and nonlinear (red) components of the training labels, proportional to $\sigma_\beta^2\sigma_X^2$ and $\sigma_{\delta y^*}^2$, respectively, in Eq. (29) (see Sec. IV A), have a corresponding contribution to the variance for all values of α_f in Eq. (30). This suggests the following interpretation: a model with nonzero bias gives rise to variance by interpreting part of the training set's signal y^* as noise. In other words, a model which cannot fully express the relationships underlying the data distribution may inadvertently treat this unexpressed signal as noise.

We demonstrate this phenomena in Figs. 5(c)–5(f) using our model of linear regression trained at the interpolation threshold ($N_f = M$), where the only contribution to the bias stems from the nonlinear components of the training labels. In each panel, we plot the labels as a function of the projection of each data point's input features onto the minimum principal component $\hat{\mathbf{h}}_{\min} \cdot \vec{\mathbf{x}}$ for a set of training data (orange) and a set of test data (blue). We then compare the resulting model (orange line) to the expected relationship for an average test set (blue line), representing the underlying relationship in the data distribution.

To confirm that bias is necessary for this phenomenon, Figs. 5(c) and 5(d) show simulations for a linear teacher model $f(h) = h$ with and without label noise. In this case, the student and teacher models match and the model is unbiased. As expected, we find that with label noise, the model overfits the training data and the resulting slope does not accurately reflect the true relationship underlying the data, while without label noise, the model is able to avoid overfitting and provides a good approximation of the true relationship

In Figs. 5(e) and 5(f), we performed the same simulations for a nonlinear teacher model $f(h) = \tanh(h)$ with and without label noise. In this case, the student and teacher models do not match and the model is always biased. With label noise, the model overfits the training data. However, even in the absence of label noise, the model *still* overfits the training data. Collectively, our simulations indicate that even if the label noise is zero, any finite amount of bias can result in overfitting, especially if the training set severely undersamples the data along a particular direction in feature space.

We find that this observation holds for both models for every observed source of bias, including contributions stemming from the linear and nonlinear components of the labels (proportional to $\sigma_\beta^2\sigma_X^2$ and $\sigma_{\delta y*}^2$, respectively) and the linear and nonlinear components of the hidden features in the test data [proportional to σ_X^2 and $\sigma_{\delta z}^2$, respectively, in Eqs. (20) and (21)]. As a result, in the absence of label noise, the test error can only diverge at an interpolation threshold when a model is biased. Finally, we note that this behavior also manifests in contributions to the training error in the underparameterized regime for each model, with each source of bias corresponding to an additional source of training error below the interpolation threshold.

D. Interpolating is not the same as overfitting

Our results make clear that interpolation (zero training error) occurs independently from overfitting (poor generalization) in overparameterized models. Interpolation occurs when the number of independent directions in the space of hidden features (or equivalently, input features in linear regression) sampled by the training set is sufficient to account for the variations in the labels. In both models, the interpolation threshold is located where the number of principal components (measured via the rank of Z^TZ) matches the number of data points. On the other hand, the test error diverges as a result of the variance diverging at the interpolation threshold. These larges variances result from poor sampling along directions in Z^TZ (small eigenvalues), resulting in very little spread of data along these directions in the training set relative to the full distribution.

In underparameterized models, interpolation and overfitting coincide. Increasing the number of fit parameters results in a greater number of sampled directions in the space of features, but makes it more likely to poorly sample any particular direction, resulting in large variance. The result is that the interpolation threshold always coincides with a divergence in the test error. In contrast, interpolation and overfitting occur independently in overparameterized models. Once the interpolation threshold is reached, further increasing the number of fit parameters cannot improve the training error since it is already at a minimum. However, increasing the model complexity can reduce the effects of overfitting and decrease the variance by allowing for better sampling along the directions captured by the training set (Fig. 4). For this reason, increasing model complexity past the interpolation threshold can actually result in an *increase* in model performance without succumbing to overfitting.

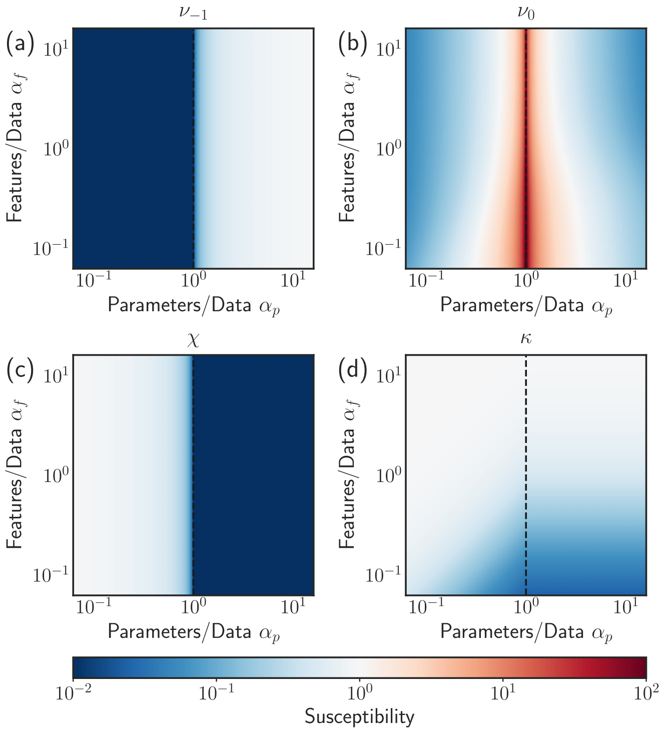

FIG. 6. Susceptibilities for random nonlinear features model. Analytic solutions for three key susceptibilities as a function of $\alpha_p = N_p/M$ and $\alpha_f = N_f/M$. [(a) and (b)] The susceptibility ν measures the sensitivity of the fit parameters to small perturbations in the gradient. In the small λ limit, $\nu \approx \lambda^{-1}\nu_{-1} + \nu_0$. (a) The coefficient ν_{-1} characterizes overparameterization, equal to the the fraction of fit parameters in excess of that needed to achieve zero training error, (b) while ν_0 characterizes overfitting, diverging at the interpolation threshold when Z^TZ has a small eigenvalue. (c) The susceptibility χ measures the sensitivity of the residual label errors of the training set to small perturbations in the label noise. As a result, χ characterizes interpolation, equal to the fraction of data points that would need to be removed from the training set to achieve zero training error. (d) The susceptibility κ measures the sensitivity of the residual parameter errors to small perturbations in the ground truth parameters. We observe that κ decreases as a model becomes less biased, indicating that the model is better able to express the relationships underlying the data.

E. Susceptibilities measure sensitivity to perturbations

Here, we discuss the roles of the susceptibilities that naturally arise as part of our cavity calculations. In many physical systems, susceptibilities are quantities of interest that measure the effects on a system due to small perturbations. In particular, the susceptibilities in our models each characterize a different type of perturbation and in doing so, a different aspect of the double-descent phenomenon. Setting the gradient equation in Eq. (8) equal to a small nonzero field $\vec{\boldsymbol{\eta}}$, such that $\partial L/\partial\hat{\mathbf{w}} = \vec{\boldsymbol{\eta}}$, each of the key susceptibilities in our derivations can be expressed as the trace of a corresponding susceptibility matrix,

$$\nu = \frac{1}{N_p}\operatorname{Tr}\frac{\partial\hat{\mathbf{w}}}{\partial\vec{\boldsymbol{\eta}}},\qquad \chi = \frac{1}{M}\operatorname{Tr}\frac{\partial\Delta\vec{\mathbf{y}}}{\partial\vec{\boldsymbol{\varepsilon}}},\qquad \kappa = \frac{1}{N_f}\operatorname{Tr}\frac{\partial\Delta\vec{\boldsymbol{\beta}}}{\partial\vec{\boldsymbol{\beta}}}. \tag{31}$$

In Fig. 6, we plot each of these quantities as a function of α_p and α_f for the random nonlinear features model (see Appendix for analytic expressions).

The susceptibility ν measures perturbations to the fit parameters $\hat{\mathbf{w}}$ due to small changes in the gradient $\vec{\boldsymbol{\eta}}$. In the small λ limit, we make the approximation $\nu \approx \lambda^{-1}\nu_{-1} + \nu_0$ and find that the coefficient of each term has a different interpretation. The first coefficient ν_{-1}, shown in Fig. 6(a), characterizes overparameterization, counting the fraction of fit parameters in excess of that needed to achieve zero training error. Since these degrees of freedom are effectively unconstrained in the small λ limit, this term diverges as λ approaches zero. The second coefficient ν_0, shown in Fig. 6(b), characterizes overfitting and diverges at the interpolation threshold in concert with the variance when Z^TZ has a small eigenvalue. We note that ν is actually the trace of the inverse Hessian of the loss function in Eq. (5), or is equivalently the Green's function, and can be used to extract the eigenvalue spectrum of the Hessian matrix [55].

The second susceptibility χ, shown in Fig. 6(c), measures the sensitivity of the residual label errors of the training set $\Delta\mathbf{\vec{y}}$ to small changes in the label noise $\vec{\boldsymbol{\varepsilon}}$. We observe that χ goes to zero at the interpolation threshold and remains zero in the interpolation regime. Accordingly, χ characterizes interpolation by measuring the fractions of data points that would need to be removed from the training set to achieve zero training error.

Finally, κ, shown in Fig. 6(d), measures the sensitivity of the residual parameter errors $\Delta\vec{\boldsymbol{\beta}}$ to small changes in the underlying ground truth parameters $\vec{\boldsymbol{\beta}}$. We observe that κ decreases as the model becomes less biased, indicating that the model is better able to express the relationships underlying the data (the relationship of κ to the bias is explored in more detail Ref. [59]).

F. Nonstandard bias-variance decompositions lead to incorrect interpretations of double-descent

The analytical results for bias and variance for the random nonlinear features model extend the classical understanding of generalization into a modern setting. While the model exhibits a classical bias-variance trade-off in the underparameterized regime, in the overparameterized regime the test error decreases monotonically due to a monotonic reduction in both bias and variance, even in the absence of regularization. In other words, the benefits of overparameterization are twofold: it can reduce the likelihood of overfitting the training data, while simultaneously improving a model's ability to capture trends hidden in the data.

The alternative and varying interpretations of the double-descent phenomenon found in previous studies are a direct result of the use of nonstandard bias-variance decompositions, highlighting the importance of using the historical definitions when using these quantities to interpret double-descent. Much of this confusion can be attributed to the precise definition of what we call the *sampling average* in our definitions for bias and variance, which captures the randomness associated with sampling the training data $\mathcal{D}$. Previous studies have deviated from these standard definitions in two ways (see Sec. S6 of Ref. [5] for numerical comparisons of these alternatives with the standard definitions).

The first is the so-called fixed-design setting in which the design matrix X is not included in the sampling average [17,19,27,30,31,34,40]. By holding the design matrix fixed for the training set, but not the test set, an effective mismatch arises between their respective data distributions, introducing an additional source of bias. As a result, one finds that the bias of the random nonlinear features model diverges at the transition, suggesting the model does not display a classical bias-variance trade-off in the underparameterized regime, despite exhibiting a U-shaped test error [17,19,40].

In the second nonstandard formulation, the random initialization of the hidden layer is included as part of the sampling average [17,25,26,31,38,44]. Consequently, the bias in this setting can be interpreted as measuring the bias of an ensemble model, $\hat{y}_{\text{ens}}(\vec{\mathbf{x}}) = \mathrm{E}_W[\hat{y}(\vec{\mathbf{x}})]$, composed of an average over all possible models with different matrices W, rather than the actual model under consideration. In this setting, the bias misses a contribution that would normally be incurred due to the reality that the model only utilizes a single instance of the matrix W, rather than averaging over the entire ensemble. In this setting, one finds that the bias of the random nonlinear features model decreases to a minimum at the interpolation threshold and then remains constant into the overparameterized regime—paradoxically suggesting that the ability of a model to express complex relationships stops increasing once one reaches the interpolation transition [17,25,26,38].

In contrast to these two scenarios, we find that the bias of the nonlinear random features model monotonically decreases with the number of parameters in both the under *and* overparameterized regimes, suggesting that adding parameters while holding the number of input features fixed always increases the ability a model to capture trends in the data. This means that while there is a trade-off between bias and variance in the underparameterized regime, this trade-off disappears in the overparameterized regime where both bias and variance decrease as one adds fit parameters.

VI. CONCLUSIONS

Understanding how the bias-variance trade-off manifests in overparameterized models where the number of fit parameters far exceeds the number of data points is a fundamental problem in modern statistics and machine learning. Here, we have used the zero-temperature cavity method, a technique from statistical physics, to derive exact analytic expressions for the training error, test error, bias, and variance in the thermodynamic limit for two minimal model architectures: linear regression (no basis functions) and the random nonlinear features model (a two-layer neural network with nonlinear activation functions where only the top layer is trained). These analytic expressions, when combined with numerical simulations, help explain one of the most puzzling features of modern ML methods: the ability to generalize well while simultaneously achieving zero error on the training data.

We observe this phenomenon of "memorizing without overfitting" in both models. Importantly, our results show that this ability to generalize is not unique to modern ML methods such as those employed in deep learning; both models we consider here are convex. We also note that we do not employ commonly used methods such as stochastic gradient descent to train our models. Instead, we use a straightforward regularization procedure based on an L_2 penalty and even work in the limit where the strength of the regularization is sent to zero. This shows that the ability to generalize while achieving zero training error, sometimes referred to as interpolation, seems to be a generic property of even the simplest overparameterized models such as linear regression and does not require any special training, regularization, or initialization methods.

Our results show that in stark contrast with the kinds of models considered in classical statistics, the variance of overparameterized models reaches a maximum at the interpolation threshold (the model complexity at which one can achieve zero error on the training data set) and then surprisingly decreasing with model complexity beyond this threshold, giving rise to the double-descent phenomenon. These large variances at the interpolation threshold are directly tied to the existence of small eigenvalues in the Hessian matrix, which can be interpreted as a symptom of poor sampling of the data distribution by the training set when viewed by the model through the hidden features. In addition, overparameterized models can introduce new sources of bias. Bias can arise not only from a mismatch between the model and the underlying data distribution, but also from training data sets that span only a subset of the data's feature space. Overparameterized models with bias can also mistake signal for noise, resulting in a nonzero variance even in the absence of noise. This shows that the properties overparameterized models are governed by a subtle interplay between model architecture and random sampling of the data distribution via the training data set.

We note that our models are limited in two significant ways: (i) we focus on the "lazy regime" in which the kernel remains fixed during optimization and (ii) we consider convex loss landscapes containing unique solutions. In contrast, deep learning models in practical settings often exhibit highly nonconvex loss landscapes and exist in the "feature regime" where their kernels evolve to better express the data. Many questions remain regarding the relationship between these two properties: how do neural networks learn "good" sets of features via their kernels and how do such choices relate to different local minima in the overall loss landscape? Recent work suggests that in this more complex setting, generalization error may be improved by looking for wider, more representative minima in the landscape [60–62]. Understanding bias, variance, and generalization in the context of nonconvex fitting functions and the relationship of these quantities to the width and local entropy of minima represents an important future area of investigation.

One possible direction for exploring these ideas may be to exploit the relationship between wide neural nets and Gaussian processes [14,15,63] and explore how the spectrum changes with the properties of various minima. Alternatively, one could apply our analytical approach to study fixed kernel methods in nonconvex settings. For example, the perceptron exhibits a nonconvex loss landscape by including negative constraint cutoffs and can be solved analytically by utilizing a Replica Symmetry Breaking ansatz [64]. In principle, it should be possible to extend these calculations to compute the bias-variance decomposition, eigenvalue spectrum, and susceptibilities with and without basis functions.

Finally, our analysis suggests that our conclusions may be tested directly in practical settings in two ways. First, it would be instructive to compute the eigenvalue spectra of the Hessian of deep learning models. It is known that the eigenvalue spectra of neural networks in the overparameterized regime exhibit a gap with a large number of eigenvalues clustered around zero and the rest located in a nonzero bulk [65]. However, there has not been a comprehensive study of how the spectrum evolves as one advances through the interpolation threshold. Second, it would be interesting to compute the relevant susceptibilities, such as those in Eq. (31). While we do not expect the susceptibilities of deep learning models to quantitatively match those computed here, we do expect them to follow the qualitative behavior exhibited in Fig. 6. These susceptibilities could be computed by utilizing their matrix forms (e.g., ν is the trace of the inverse Hessian), or by calculating the linear response directly via efficient differentiation techniques originally developed for computing gradients for metalearning [66]. Examining such susceptibilities may also prove useful in understanding the nature of deep learning models in the feature regime and nonconvex loss landscapes.

ACKNOWLEDGMENTS

We would like to thank Robert Marsland III for extremely useful discussions. This work was supported by NIH NIGMS Grant No. 1R35GM119461 and a Simons Investigator in the Mathematical Modeling of Living Systems (MMLS) award to P.M. The authors also acknowledge support from the Shared Computing Cluster administered by Boston University Research Computing Services.

APPENDIX: ANALYTIC EXPRESSIONS FOR RANDOM NONLINEAR FEATURES MODEL

For the random nonlinear features model, the solutions for the five averages used to express the solutions in Eqs. (18)–(21) are in turn related to a set of five scalar susceptibilities that are a natural result of the cavity method, ν, χ, κ, ω, and ϕ. Each of these susceptibilities is defined as the ensemble average of the trace of a different susceptibility matrix which measures the responses of quantities such as the residual label error, residual parameter error, fit parameter values, etc., to small perturbations (see Sec. V E). Collectively, the ensemble-averaged quantities satisfy the equations

$$\begin{pmatrix} \langle \hat{w}^2 \rangle \\ \langle \hat{u}^2 \rangle \\ \langle \Delta y^2 \rangle \\ \langle \Delta \beta^2 \rangle \end{pmatrix} = \begin{pmatrix} 1 & -\sigma_W^2 \frac{\alpha_f}{\alpha_p} \nu^2 & -\sigma_{\delta z}^2 \alpha_p^{-1} \nu^2 & 0 \\ -\sigma_W^2 \omega^2 & 1 & -\sigma_X^2 \alpha_f^{-1} \kappa^2 & 0 \\ -\sigma_{\delta z}^2 \chi^2 & 0 & 1 & -\sigma_X^2 \chi^2 \\ -\sigma_W^2 \kappa^2 & 0 & -\sigma_X^2 \alpha_f^{-1} \phi^2 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ \sigma_\beta^2 \omega^2 \\ (\sigma_\varepsilon^2 + \sigma_{\delta y^*}^2) \chi^2 \\ \sigma_\beta^2 \kappa^2 \end{pmatrix}, \tag{A1}$$

$$\langle \hat{w}_1 \hat{w}_2 \rangle = \frac{\sigma_\beta^2}{\sigma_W^2} \frac{\sigma_W^4 \frac{\alpha_f}{\alpha_p} \omega^2 \nu^2}{\left(1 - \sigma_W^4 \frac{\alpha_f}{\alpha_p} \omega^2 \nu^2\right)}, \quad \langle \Delta \beta_1 \Delta \beta_2 \rangle = \sigma_\beta^2 \frac{\kappa^2}{\left(1 - \sigma_W^4 \frac{\alpha_f}{\alpha_p} \omega^2 \nu^2\right)}. \tag{A2}$$

The quantity $\langle \hat{u}^2 \rangle$ is the mean squared average of the length-N_f vector quantity $\hat{\mathbf{u}} = X^T \Delta \vec{\mathbf{y}}$ obtained as a byproduct of the cavity derivation. The solutions for linear regression can be formulated similarly in terms of a pair of scalar susceptibilities that are analogous to χ and ν (see Sec. S1F of Ref. [5]).

In each model, a subset of the scalar susceptibilities diverges wherever two different sets of solutions meet, indicating the existence of a second-order phase transition. For the random nonlinear features model, these susceptibilities are (to leading order in small λ)

$$\chi = \begin{cases} 1 - \alpha_p & \text{if} \quad N_p < M \\ \frac{\lambda}{2\sigma_{\delta_z}^2} \frac{\alpha_p}{(\alpha_p - 1)}\left[1 - (1 + \Delta\varphi)\alpha_f + \sqrt{[1 - (1 + \Delta\varphi)\alpha_f]^2 + 4\Delta\varphi\alpha_f}\right] & \text{if} \quad N_p > M \end{cases}, \tag{A3}$$

$$\nu = \begin{cases} \frac{1}{2\sigma_{\delta_z}^2} \frac{1}{(1-\alpha_p)}\left[\alpha_p - (1 + \Delta\varphi)\alpha_f + \sqrt{[\alpha_p - (1 + \Delta\varphi)\alpha_f]^2 + 4\Delta\varphi\alpha_f\alpha_p}\right] & \text{if} \quad N_p < M \\ \frac{1}{\lambda}\frac{(\alpha_p - 1)}{\alpha_p} + \frac{1}{2\sigma_{\delta_z}^2} \frac{1}{(\alpha_p - 1)}\left[1 - (1 + \Delta\varphi)\alpha_f + \sqrt{[1 - (1 + \Delta\varphi)\alpha_f]^2 + 4\Delta\varphi\alpha_f}\right] & \text{if} \quad N_p > M \end{cases}, \tag{A4}$$

$$\kappa = \frac{1}{1 + \sigma_X^2 \sigma_W^2 \alpha_f^{-1} \chi \nu}, \quad \omega = \sigma_X^2 \alpha_f^{-1} \chi \kappa, \quad \phi = -\sigma_W^2 \nu \kappa. \tag{A5}$$

We plot these analytic forms for χ, ν and κ in Fig. 6.

[1] Y. Lecun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[2] A. Canziani, A. Paszke, and E. Culurciello, An analysis of deep neural network models for practical applications, arXiv:1605.07678.

[3] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, Understanding deep learning requires re-thinking generalization, international conference on learning representations (ICLR) (2017).

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

[5] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevResearch.4.013201 for complete analytic derivations and additional numerical results.

[6] M. Belkin, D. Hsu, S. Ma, and S. Mandal, Reconciling modern machine-learning practice and the classical bias-variance tradeoff, Proc. Natl. Acad. Sci. USA **116**, 15849 (2019).

[7] M. Geiger, S. Spigler, S. D'Ascoli, L. Sagun, M. Baity-Jesi, G. Biroli, and M. Wyart, Jamming transition as a paradigm to understand the loss landscape of deep neural networks, Phys. Rev. E **100**, 012115 (2019).

[8] S. Spigler, M. Geiger, S. D'Ascoli, L. Sagun, G. Biroli, and M. Wyart, A jamming transition from under to overparametrization affects generalization in deep learning, J. Phys. A: Math. Theor. **52**, 474001 (2019).

[9] M. Geiger, A. Jacot, S. Spigler, F. Gabriel, L. Sagun, S. d'Ascoli, G. Biroli, C. Hongler, and M. Wyart, Scaling description of generalization with number of parameters in deep learning, J. Stat. Mech.: Theory Exp. (2020) 023401.

[10] 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).

[11] D. Kobak [@hippopedoid], Twitter Thread: twitter.com/hippopedoid/status/1243229021921579010, (2020).

[12] M. Loog, T. Viering, A. Mey, J. H. Krijthe, and D. M. J. Tax, A brief prehistory of double descent, Proc. Natl. Acad. Sci. USA **117**, 10625 (2020).

[13] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, Deep double descent: Where bigger models and more data hurt, international conference on learning representations (ICLR) (2020).

[14] A. Jacot, F. Gabriel, and C. Hongler, Neural tangent kernel: Convergence and generalization in neural networks, Advances in Neural Information Processing Systems (NeurIPS) **31** (2018).

[15] J. Lee, L. Xiao, S. S. Schoenholz, Y. B. R. Novak, J. Sohl-Dickstein, and J. Pennington, Wide neural networks of any depth evolve as linear models under gradient descent, Advances in Neural Information Processing Systems (NeurIPS) **32** (2019).

[16] M. Geiger, S. Spigler, A. Jacot, and M. Wyart, Disentangling feature and lazy training in deep neural networks, J. Stat. Mech.: Theory Exp. (2020) 113301.

[17] B. Adlam and J. Pennington, Understanding double descent requires a fine-grained bias-variance decomposition, Advances in Neural Information Processing Systems (NeurIPS) **33**, 11022 (2020).

[18] M. S. Advani, A. M. Saxe, and H. Sompolinsky, High-dimensional dynamics of generalization error in neural networks, Neural Networks **132**, 428 (2020).

[19] J. Ba, M. Erdogdu, T. Suzuki, D. Wu, and T. Zhang, Generalization of Two-layer Neural Networks: An Asymptotic Viewpoint, International Conference on Learning Representations (ICLR) (2020).

[20] J. Barbier, F. Krzakala, N. Macris, L. Miolane, and L. Zdeborová, Optimal errors and phase transitions in high-dimensional generalized linear models, Proc. Natl. Acad. Sci. USA **116**, 5451 (2019).

[21] P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler, Benign overfitting in linear regression, Proc. Natl. Acad. Sci. USA **117**, 30063 (2020).

[22] M. Belkin, D. Hsu, and J. Xu, Two models of double descent for weak features, SIAM Journal on Mathematics of Data Science **2**, 1167 (2020).

[23] K. Bibas, Y. Fogel, and M. Feder, A new look at an old problem: A universal learning approach to linear regression, *IEEE*

International Symposium on Information Theory (ISIT) (IEEE, Piscataway, NJ, 2019).

[24] Z. Deng, A. Kammoun, and C. Thrampoulidis, A Model of Double Descent for High-Dimensional Logistic Regression, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4267 (2020).

[25] S. D'Ascoli, L. Sagun, and G. Biroli, Triple descent and the two kinds of overfitting: Where & why do they appear? Advances in Neural Information Processing Systems (NeurIPS) **33** (2020).

[26] S. D'Ascoli, M. Refinetti, G. Biroli, and F. Krzakala, Double Trouble in Double Descent: Bias and Variance(s) in the Lazy Regime, Proceedings of the 37th International Conference on Machine Learning (ICML), PMLR **119**, 2280 (2020).

[27] M. Dereziński, F. Liang, and M. W. Mahoney, Exact expressions for double descent and implicit regularization via surrogate random design, Advances in Neural Information Processing Systems (NeurIPS) **33**, 5152 (2020).

[28] O. Dhifallah and Y. M. Lu, A Precise Performance Analysis of Learning with Random Features, arXiv:2008.11904.

[29] F. Gerace, B. Loureiro, F. Krzakala, M. Mézard, and L. Zdeborová, Generalisation error in learning with random features and the hidden manifold model, Proceedings of the 37th International Conference on Machine Learning (ICML), PMLR **119**, 3452 (2020).

[30] T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani, Surprises in high-dimensional ridgeless least squares interpolation, arXiv:1903.08560.

[31] A. Jacot, B. Şimşek, F. Spadaro, C. Hongler, and F. Gabriel, Implicit regularization of random feature models, Proceedings of the 37th International Conference on Machine Learning (ICML), PMLR **119**, 4631 (2020).

[32] G. R. Kini and C. Thrampoulidis, Analytic study of double descent in binary classification: The impact of loss, *IEEE International Symposium on Information Theory (ISIT)* (IEEE, Piscataway, NJ, 2020).

[33] A. K. Lampinen and S. Ganguli, An analytic theory of generalization dynamics and transfer learning in deep linear networks, International Conference on Learning Representations (ICLR) (2019).

[34] Z. Li, W. J. Su, and D. Sejdinovic, Benign overfitting and noisy features, arXiv:2008.02901.

[35] T. Liang and A. Rakhlin, Just interpolate: Kernel "Ridgeless" regression can generalize, Ann. Stat. **48**, 1329 (2020).

[36] T. Liang, A. Rakhlin, and X. Zhai, On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels, Proceedings of Thirty Third Conference on Learning Theory, PMLR **125**, 2683 (2020).

[37] Z. Liao, R. Couillet, and M. W. Mahoney, A random matrix analysis of random Fourier features: beyond the Gaussian kernel, a precise phase transition, and the corresponding double descent, Advances in Neural Information Processing Systems (NeurIPS) **33** (2020).

[38] L. Lin and E. Dobriban, What causes the test error? Going beyond bias-variance via ANOVA, J. Mach. Learn. Research **22**, 1 (2021).

[39] P. P. Mitra, Understanding overfitting peaks in generalization error: Analytical risk curves for l_2 and l_1 penalized interpolation, arXiv:1906.03667.

[40] S. Mei and A. Montanari, The generalization error of random features regression: Precise asymptotics and the double descent curve, Commun. Pure Appl. Math. **75**, 667 (2021).

[41] V. Muthukumar, K. Vodrahalli, and A. Sahai, Harmless interpolation of noisy data in regression, *IEEE International Symposium on Information Theory (ISIT)* (IEEE, Piscatawa, NJ, 2019).

[42] P. Nakkiran, More data can hurt for linear regression: Sample-wise double descent, arXiv:1912.07242.

[43] J. Xu and D. Hsu, On the number of variables to use in principal component regression, Advances in Neural Information Processing Systems (NeurIPS) **32** (2019).

[44] Z. Yang, Y. Yu, C. You, J. Steinhardt, and Y. Ma, Rethinking bias-variance trade-off for generalization of neural networks, Proceedings of the 37th International Conference on Machine Learning (ICML), PMLR **119**, 10767 (2020).

[45] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, Identifying and attacking the saddle point problem in high-dimensional non-convex optimization, Advances in Neural Information Processing Systems (NeurIPS) **27** (2014).

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

[47] M. Mézard and G. Parisi, The cavity method at zero temperature, J. Stat. Phys. **111**, 1 (2003).

[48] M. Ramezanali, P. P. Mitra, and A. M. Sengupta, The cavity method for analysis of large-scale penalized regression, arXiv:1501.03194.

[49] P. Mehta, W. Cui, C.-H. Wang, and R. Marsland, Constrained optimization as ecological dynamics with applications to random quadratic programming in high dimensions, Phys. Rev. E **99**, 052111 (2019).

[50] M. Advani, G. Bunin, and P. Mehta, Statistical physics of community ecology: a cavity solution to MacArthur's consumer resource model, J. Stat. Mech.: Theory Exp. (2018) 033406.

[51] V. A. Marčenko and L. A. Pastur, Distribution of eigenvalues for some sets of random matrices, Mathematics of the USSR-Sbornik **1**, 457 (1967).

[52] J. Pennington and P. Worah, Nonlinear random matrix theory for deep learning, J. Stat. Mech.: Theory Exp. (2019) 124005.

[53] M. Mézard, Mean-field message-passing equations in the Hopfield model and its generalizations, Phys. Rev. E **95**, 022117 (2017).

[54] M. Ramezanali, P. P. Mitra, and A. M. Sengupta, Critical behavior and universality classes for an algorithmic phase transition in sparse reconstruction, J. Stat. Phys. **175**, 764 (2019).

[55] W. Cui, J. W. Rocks, and P. Mehta, The perturbative resolvent method: Spectral densities of random matrix ensembles via perturbation theory, arXiv:2012.00663.

[56] S. Geman, E. Bienenstock, and R. Doursat, Neural networks and the bias/variance dilemma, Neural Comput. **4**, 1 (1992).

[57] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006).

[58] S. Goldt, M. Mézard, F. Krzakala, and L. Zdeborová, Modeling the Influence of Data Structure on Learning in Neural Networks: The Hidden Manifold Model, Phys. Rev. X **10**, 041044 (2020).

[59] J. W. Rocks and P. Mehta, The geometry of overparameterized regression and adversarial perturbations, arXiv:2103.14108.

[60] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina, Entropy-SGD: biasing gradient descent into wide valleys, J. Stat. Mech.: Theory Exp. (2019) 124018.

[61] C. Baldassi, E. M. Malatesta, M. Negri, and R. Zecchina, Wide flat minima and optimal generalization in classifying high-dimensional Gaussian mixtures, J. Stat. Mech.: Theory Exp. (2020) 124012.

[62] F. Pittorino, C. Lucibello, C. Feinauer, E. M. Malatesta, G. Perugini, C. Baldassi, M. Negri, E. Demyanenko, and R. Zecchina, Entropic gradient descent algorithms and wide flat minima, International Conference on Learning Representations (ICLR) (2021).

[63] S. Yaida, Non-Gaussian processes and neural networks at finite widths, Proceedings of The First Mathematical and Scientific Machine Learning Conference, PMLR **107**, 165 (2020).

[64] S. Franz, G. Parisi, M. Sevelev, P. Urbani, and F. Zamponi, Universality of the SAT-UNSAT (jamming) threshold in non-convex continuous constraint satisfaction problems, SciPost Phys. **2**, 019 (2017).

[65] L. Sagun, U. Evci, V. U. Güney, Y. Dauphin, and L. Bottou, Empirical analysis of the hessian of over-parametrized neural networks, arXiv:1706.04454.

[66] C. Finn, P. Abbeel, and S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, Proceedings of the 34th International Conference on Machine Learning (ICML), PMLR **70**, 1126 (2017).