# Data variability in neural network Bayesian inference

Javed Lindner [1,2,3,*] David Dahmen [1] Michael Krämer,[3] and Moritz Helias [1,2]
[1]*Institute for Advanced Simulation (IAS-6), Computational and Systems Neuroscience, Jülich Research Centre, Jülich, Germany*
[2]*Department of Physics, Faculty 1, RWTH Aachen University, Aachen, Germany*
[3]*Institute for Theoretical Particle Physics and Cosmology, RWTH Aachen University, Aachen, Germany*

(Received 7 December 2023; accepted 5 August 2025; published 25 March 2026)

Bayesian inference and kernel methods are well established in machine learning. The neural network Gaussian process in particular provides a concept to investigate neural networks in the limit of infinitely wide hidden layers by using kernel and inference methods. Here, we build upon this limit and provide a field theoretic formalism that covers the generalization properties of infinitely wide networks. We systematically compute generalization properties of linear, nonlinear, and deep nonlinear networks for kernel matrices with heterogeneous entries. In contrast to currently employed spectral methods, we derive the generalization properties from the statistical properties of the input, elucidating the interplay of input dimensionality, size of the training dataset, and variability of the data. We show that data variability leads to a non-Gaussian action reminiscent of a $\varphi^3 + \varphi^4$ theory. Using our formalism on a synthetic task and on MNIST, we obtain a homogeneous kernel matrix approximation for the learning curve as well as corrections due to data variability that allow the estimation of the generalization properties and exact results for the bounds of the learning curves in the case of infinitely many training data points.

DOI: 10.1103/r1s3-lr3g

## I. INTRODUCTION

Machine learning and in particular deep learning continues to influence all areas of science. Employed as a scientific method, explainability, a defining feature of any scientific method, however, is still largely missing. This is also important to provide guarantees and to guide educated design choices to reach a desired level of accuracy. The reason is that the underlying principles by which artificial neural networks (ANNs) reach their unprecedented performance are largely unknown. There is, up to date, no complete theoretical framework that fully describes the behavior of artificial neural networks so that it would explain the mechanisms by which neural networks operate. Such a framework would also be useful to support architecture search and network training.

Investigating the theoretical foundations of artificial neural networks on the basis of statistical physics dates back to the 1980s. Early approaches to investigate neural information processing were mainly rooted in the spin-glass literature and included the computation of the memory capacity of the perceptron, collective network dynamics [1], and investigations of the energy landscape of attractor network [2–4].

As in the thermodynamic limit in solid state physics, some modern approaches deal with ANNs with an infinite number of hidden neurons to simplify calculations. This leads to a relation between ANNs and Bayesian inference on Gaussian processes [5,6], known as the neural network Gaussian process (NNGP) limit: The prior distribution of network outputs across realizations of network parameters here becomes a Gaussian process that is uniquely described by its covariance function, also known as its kernel. This approach has been used to obtain insights into the relation of network architecture and trainability [7–11]. Other works have investigated training by gradient descent as a means to shape the corresponding kernel [12]. A series of recent studies also captures networks at finite width, including adaptation of the kernel due to feature learning effects [13–19]. Even though training networks with gradient descent is the most abundant setup, different schemes such as Bayesian deep learning [20] provide an alternative perspective on training neural networks. Rather than finding the single-best parameter realization to solve a given task, the Bayesian approach aims to find the optimal parameter distribution.

In this work, we adopt the Bayesian approach and investigate the effect of variability in the training data on the generalization properties of wide neural networks. We do so in the limit of infinitely wide linear and nonlinear networks. To obtain analytical insights, we apply tools from statistical field theory to derive approximate expressions for the predictive distribution in the NNGP limit (see Fig. 1). The remainder of this work is structured in the following way: In Sec. II, we describe the setup of supervised learning in shallow and deep networks in the framework of Bayesian inference and we introduce a synthetic dataset that allows us to control the degree of pattern separability, dimensionality, and variability of the resulting overlap matrix. In Sec. III, we develop the field theoretical approach to learning curves and its application to the synthetic dataset as well as to MNIST [21]: Sec. III A presents the general formalism and shows that data variability

*Contact author: ja.lindner@fz-juelich.de

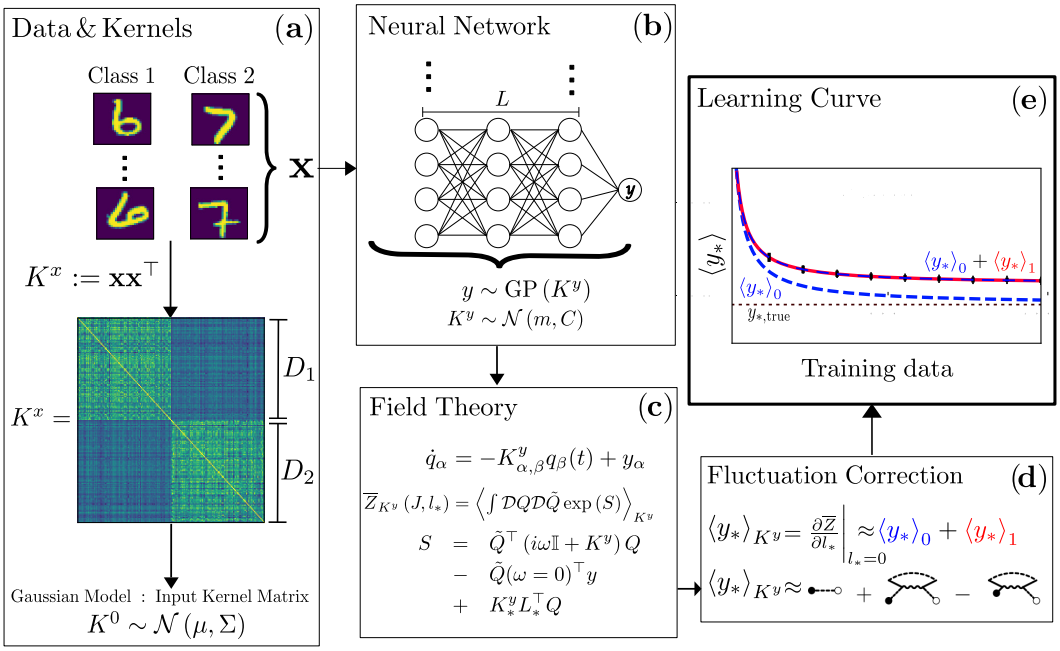



FIG. 1. Field theory of generalization in Bayesian inference. (a) A binary classification task, such as distinguishing pairs of digits in MNIST, can be described with the help of an overlap matrix $K^x$ that represents similarity across the $c = c_1 + c_2$ images of the training set of two classes, 1 and 2 with $D_1$ and $D_2$ samples, respectively. Entries of the overlap matrix are heterogeneous. Different drawings of $c$ example patterns each lead to different realizations of the overlap matrix; the matrix is stochastic. We here describe the matrix elements by a correlated multivariate Gaussian. (b) The data are fed through a feedforward neural network to produce an output $y$. In the case of infinitely wide hidden layers and under Gaussian priors on the network weights, the output of the network is a Gaussian process with the kernel $K^y$, which depends on the network architecture and the input kernel $K^x$. (c) To obtain statistical properties of the posterior distribution, we compute its disorder-averaged moment-generating function $\overline{Z}(J, l_*)$ diagrammatically. (d) The leading-order contribution from the homogeneous kernel $\langle y^* \rangle_0$ is corrected by $\langle y^* \rangle_1$ due to the variability of the overlaps; both follow as derivatives of $\overline{Z}(J, l_*)$. (e) Comparing the mean network output on a test point $\langle y^* \rangle$, the zeroth-order theory $\langle y_* \rangle_0$ (blue dashed), the first-order approximation in the data variability $\langle y_* \rangle_{0+1}$ (blue-red dashed), and empirical results (black crosses) as a function of the amount of training data (learning curve) shows how variability in the dataset limits the network performance and validates the theory.

in general leads to a non-Gaussian process. Here, we also derive perturbative expressions to characterize the posterior distribution of the network output. We first illustrate these ideas on the simplest but nontrivial example of linear Bayesian regression and then generalize them first to linear and then to nonlinear deep networks. In Sec. II C, we show results for the synthetic dataset to obtain interpretable expressions that allow us to identify how data variability affects generalization; we then generalize the theory to be applicable to real-world datasets and illustrate its application with the MNIST dataset. In Sec. IV, we summarize our findings, discuss them in the light of the literature, and provide an outlook.

## II. SETUP

In this background section, we outline the relation between neural networks, Gaussian processes, and Bayesian inference. We further present an artificial binary classification task that allows us to control the degree of pattern separation and variability and test the predictive power of the theoretical results for the network generalization properties.

### A. Neural networks, Gaussian processes, and Bayesian inference

The advent of parametric methods such as neural networks is preceded by nonparametric approaches such as Gaussian processes. There are, however, clear connections between the two concepts that allow us to borrow from the theory of Gaussian processes and Bayesian inference to describe the seemingly different neural networks. We will here give a short recap on neural networks, Bayesian inference, Gaussian processes, and their mutual relations.

#### *1. Background: neural networks*

In general, a feedforward neural network maps inputs $x_\alpha \in \mathbb{R}^{N_{\mathrm{dim}}}$ to outputs $y_\alpha \in \mathbb{R}^{N_{\mathrm{out}}}$ via the transformations

$$h_\alpha^{(l)} = \mathbf{W}^{(l)} \phi^{(l)}\left(h_\alpha^{(l-1)}\right), \quad \text{with} \quad h_\alpha^0 = \mathbf{V} x_\alpha,$$
$$y_\alpha = \mathbf{U} \phi^{(L+1)}\left(h_\alpha^{(L)}\right), \tag{1}$$

where $\phi^{(l)}(x)$ are activation functions, $\mathbf{V} \in \mathbb{R}^{N_h \times N_{\mathrm{dim}}}$ are the read-in weights, $N_{\mathrm{dim}}$ is the dimension of the input, $\mathbf{W}^{(l)} \in \mathbb{R}^{N_h \times N_h}$ are the hidden weights, $N_h$ denotes the number of hidden neurons, and $\mathbf{U} \in \mathbb{R}^{N_{\mathrm{out}} \times N_h}$ are the readout weights. Here, $l$ is the layer index $1 \leqslant l \leqslant L$ and $L$ the number of layers of the network; we here assume layer-independent activation functions $\phi^{(l)} = \phi$. The collection of all weights are the model parameters $\Theta = \{\mathbf{V}, \mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}, \mathbf{U}\}$. The goal of training a neural network in a supervised manner is to find a set of parameters $\hat{\Theta}$ that reproduces the input-output relation $(x_{\mathrm{tr},\alpha}, y_{\mathrm{tr},\alpha})_{1 \leqslant \alpha \leqslant D}$ for a set of $D$ pairs of inputs and outputs as accurately as possible, while also maintaining the ability to

generalize. Hence, one partitions the data into a training set $\mathcal{D}_{\rm tr}$, $|\mathcal{D}_{\rm tr}| = D$, and a test set $\mathcal{D}_{\rm test}$, $|\mathcal{D}_{\rm test}| = D_{\rm test}$. The training data are given in the form of the matrices $\mathbf{x}_{\rm tr} \in \mathbb{R}^{N_{\rm dim} \times N_{\rm tr}}$ and $\mathbf{y}_{\rm tr} \in \mathbb{R}^{N_{\rm out} \times N_{\rm tr}}$. The quality of how well a neural network is able to model the relation between inputs and outputs is quantified by a task-dependent loss function $\mathcal{L}(\Theta, x_\alpha, y_\alpha)$. Starting with a random initialization of the parameters $\Theta$, one tries to find an optimal set of parameters $\hat{\Theta}$ that minimizes the loss $\sum_{\alpha=1}^{D} \mathcal{L}(\Theta, x_{{\rm tr},\alpha}, y_{{\rm tr},\alpha})$ on the training set $\mathcal{D}_{\rm tr}$. The parameters $\hat{\Theta}$ are usually obtained through methods such as stochastic gradient descent. The generalization properties of the network are quantified after the training by computing the loss $\mathcal{L}(\hat{\Theta}, x_\alpha, y_\alpha)$ on the test set $(x_{{\rm test},\alpha}, y_{{\rm test},\alpha}) \in \mathcal{D}_{\rm test}$, which are data samples that have not been used during the training process. Neural networks hence provide, by definition, a parametric modeling approach, as the goal is to a find an optimal set of parameters $\hat{\Theta}$.

#### 2. *Background: Bayesian inference and Gaussian processes*

The parametric viewpoint in Sec. II A 1 that yields a point estimate $\hat{\Theta}$ for the optimal set of parameters can be complemented by considering a Bayesian perspective [20,22,23]: For each network input $x_\alpha$, the network equations (1) yield a single output $y(x_\alpha|\Theta)$. One typically considers a stochastic output $y(x_\alpha|\Theta) + \xi_\alpha$ where $\xi_\alpha$ are Gaussian independently and identically distributed (i.i.d.) with variance $\sigma^2_{\rm reg}$ [24]. This regularization allows us to define the probability distribution $p(y|x_\alpha, \Theta) = \langle \delta[y_\alpha - y(x_{{\rm tr},\alpha}|\Theta) - \xi_\alpha] \rangle_{\xi_\alpha} = \mathcal{N}(y_\alpha; y(x_\alpha|\Theta), \sigma^2_{\rm reg})$. An alternative interpretation of $\xi_\alpha$ is a Gaussian noise on the labels. Given a particular set of the network parameters $\Theta$, this implies a joint distribution $p(\mathbf{y}|\mathbf{x}_{\rm tr}, \Theta) := \prod_{\alpha=1}^{D} \langle \delta[y_\alpha - y(x_{{\rm tr},\alpha}|\Theta) - \xi_\alpha] \rangle_{\{\xi_\alpha\}} = \prod_{\alpha=1}^{D} p(y_\alpha|x_\alpha, \Theta)$ of network outputs $\{y_\alpha\}_{1 \leqslant \alpha \leqslant D}$, each corresponding to one network input $\{x_{{\rm tr},\alpha}\}_{1 \leqslant \alpha \leqslant D}$. One aims to use the training data $\mathcal{D}_{\rm tr}$ to compute the posterior distribution for the weights $\mathbf{V}, \mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}, \mathbf{U}$ by conditioning on the network outputs to agree to the desired training values. Concretely, we here assume as a prior for the model parameters that the parameter elements $V_{ij}$, $W^{(l)}_{ij}$, and $U_{ij}$ are i.i.d. according to centered Gaussian distributions $V_{ij} \sim \mathcal{N}(0, \sigma_v^2/N_{\rm dim})$, $W^{(l)}_{ij} \sim \mathcal{N}(0, \sigma_w^2/N_h)$, and $U_{ij} \sim \mathcal{N}(0, \sigma_u^2/N_h)$.

The posterior distribution of the parameters $p(\Theta|\mathbf{x}_{\rm tr}, \mathbf{y}_{\rm tr})$ then follows from Bayes's theorem as

$$p(\Theta|\mathbf{x}_{\rm tr}, \mathbf{y}_{\rm tr}) = \frac{p(\mathbf{y}_{\rm tr}|\mathbf{x}_{\rm tr}, \Theta)\, p(\Theta)}{p(\mathbf{y}_{\rm tr}|\mathbf{x}_{\rm tr})}, \tag{2}$$

with the likelihood $p(\mathbf{y}_{\rm tr}|\mathbf{x}_{\rm tr}, \Theta)$, the weight prior $p(\Theta)$, and the model evidence $p(\mathbf{y}_{\rm tr}|\mathbf{x}_{\rm tr}) = \int d\Theta\, p(\mathbf{y}_{\rm tr}|\mathbf{x}_{\rm tr}, \Theta) p(\Theta)$, which provides the proper normalization. The posterior parameter distribution $p(\Theta|\mathbf{x}_{\rm tr}, \mathbf{y}_{\rm tr})$ also determines the distribution of the network output $y_*$ corresponding to a test point $x_*$ by marginalizing over the parameters $\Theta$:

$$p(y_*|x_*, \mathbf{x}_{\rm tr}, \mathbf{y}_{\rm tr}) = \int d\Theta\, p(y_*|x_*, \Theta) p(\Theta|\mathbf{x}_{\rm tr}, \mathbf{y}_{\rm tr}), \tag{3}$$

$$= \frac{p(y_*, \mathbf{y}_{\rm tr}|x_*, \mathbf{x}_{\rm tr})}{p(\mathbf{y}_{\rm tr}|\mathbf{x}_{\rm tr})}. \tag{4}$$

One can understand this intuitively: The distribution in Eq. (2) provides a set of viable parameters $\Theta$ based on the training data. An initial guess for the correct choice of parameters via the prior $p(\Theta)$ is refined, based on whether the choice of parameters accurately models the relation of the training data, which is encapsulated in the likelihood $p(\mathbf{y}_{\rm tr}|\mathbf{x}_{\rm tr}, \Theta)$. This viewpoint of Bayesian parameter selection is also equivalent to what is known as Bayesian deep learning [20]. The distribution $p(y_*, \mathbf{y}_{\rm tr}|x_*, \mathbf{x}_{\rm tr})$ describes the joint network outputs for all training points and the test point. In the case of wide networks, where $N_h \to \infty$ [5,6], the distribution of network outputs $p(y_*, \mathbf{y}_{\rm tr}|x_*, \mathbf{x}_{\rm tr})$ approaches a Gaussian process $y \sim \mathcal{N}(0, K^y)$, where the covariance $\langle y_\alpha y_\beta \rangle = K^y_{\alpha\beta}$ is also denoted as the kernel. This is beneficial, as the inference for the network output $y_*$ for a test point $x_*$ then also follows a Gaussian distribution with mean and covariance given by [25]

$$\langle y_* \rangle = K^y_{*\alpha} (K^y)^{-1}_{\alpha\beta}\, y_{{\rm tr},\beta}, \tag{5}$$

$$\langle (y_* - \langle y_* \rangle)^2 \rangle = K^y_{**} - K^y_{*\alpha} (K^y)^{-1}_{\alpha\beta}\, K^y_{\beta *}, \tag{6}$$

where summation over repeated indices is implied. There has been extensive research in relating the outputs of wide neural networks to Gaussian processes [5,26,27], including recent work on corrections due to finite-width effects $N_h \gg 1$ [13–18,28,29].

### B. Our contribution

A fundamental assumption of supervised learning is the existence of a joint distribution $p(x_{\rm tr}, y_{\rm tr})$ from which the set of training data as well as the set of test data are drawn. In this work, we follow the Bayesian approach and investigate the effect of variability in the training data on the generalization properties of wide neural networks. We do so in the kernel limit of infinitely wide linear and nonlinear networks. Variability here has two meanings: First, for each drawing of $D$ pairs of training samples $(x_{{\rm tr},\alpha}, y_{{\rm tr},\alpha})_{1 \leqslant \alpha \leqslant D}$ one obtains a $D \times D$ kernel matrix $K^y$ with heterogeneous entries, so in a single instance of Bayesian inference, the entries of the kernel matrix vary from one entry to the next. Second, each such drawing of $D$ training data points and one test data point $(x_*, y_*)$ leads to a different kernel $\{K^y_{\alpha\beta}\}_{1 \leqslant \alpha, \beta \leqslant D+1}$, which follows some probabilistic law $K^y \sim p(K^y)$.

Our work builds upon previous results for the NNGP limit to formalize the influence of such stochastic kernels. We here develop a field theoretic approach to systematically investigate the influence of the underlying kernel stochasticity on the generalization properties of the network, namely, the learning curve, the dependence of $\langle y_* \rangle$ on the number of training samples $D = |\mathcal{D}_{\rm tr}|$. As we assume Gaussian i.i.d. priors on the network parameters, the output kernel $K^y_{\alpha\beta}$ solely depends on the network architecture and the input overlap matrix:

$$K^x_{\alpha\beta} = \sum_{i=1}^{N_{\rm dim}} x_{\alpha i} x_{\beta i}, \quad x_\alpha, x_\beta \in \mathcal{D}_{\rm tr} \cup \mathcal{D}_{\rm test}, \tag{7}$$

with $\alpha, \beta = 1, \ldots, D+1$. We next define a data model that allows us to approximate the probability measure for the data variability.

### C. Definition of a synthetic dataset

To investigate the generalization properties in a binary classification task, we introduce a synthetic stochastic binary classification task. This task allows us to control the statistical properties of the data with regard to the dimensionality of the patterns, the degree of separation between patterns belonging to different classes, and the variability in the kernel. Moreover, it allows us to construct training datasets $\mathcal{D}_{\text{tr}}$ of arbitrary sizes and we will show that the statistics of the resulting kernels is indeed representative for more realistic datasets such as MNIST. The dataset consists of pattern realizations $x_\alpha \in \{-1, 1\}^{N_{\text{dim}}}$ with dimension $N_{\text{dim}}$ even. We denote the entries $x_{\alpha i}$ of this $N_{\text{dim}}$-dimensional vector for data point $\alpha$ as pixels that randomly take either of two values $x_{\alpha i} \in \{-1, 1\}$ with respective probabilities $p(x_{\alpha i} = 1)$ and $p(x_{\alpha i} = -1)$ that depend on the class $c(\alpha) \in \{1, 2\}$ of the pattern realization and whether the index $i$ is in the left half ($i \leqslant N_{\text{dim}}/2$) or the right half ($i > N_{\text{dim}}/2$) of the pattern: For class $c(\alpha) = 1$, each pixel $x_{\alpha i}$ with $1 \leqslant i \leqslant N_{\text{dim}}$ is realized independently as a binary variable as

$$x_{\alpha i} = \begin{cases} 1, & \text{with } p, \\ -1, & \text{with } (1-p), \end{cases} \quad \text{for } i \leqslant \frac{N_{\text{dim}}}{2}, \tag{8}$$

$$x_{\alpha i} = \begin{cases} 1, & \text{with } (1-p), \\ -1, & \text{with } p, \end{cases} \quad \text{for } i > \frac{N_{\text{dim}}}{2}. \tag{9}$$

For a pattern $x_\alpha$ in the second class $c(\alpha) = 2$, the pixel values are distributed independently of those in the first class with a statistics that equals the negative pixel values of the first class, which is $P(x_{\alpha i}) = P(-x_{\beta i})$ with $c(\beta) = 1$ and $c(\alpha) = 2$. There are two limiting cases for $p$ that illustrate the construction of the patterns: In the limit $p = 1$, each pattern $x_\alpha$ in $c = 1$ consists of a vector, where the first $N_{\text{dim}}/2$ pixels have the value $x_{\alpha i} = 1$, whereas the second half consists of pixels with the value $x_{\alpha,i} = -1$. The opposite holds for patterns in the second class $c = 2$. This limiting case is shown in Fig. 2 (right column). In the limit case $p = 0.5$, each pixel assumes the value $x_{\alpha,i} = \pm 1$ with equal probability, regardless of the pattern class membership or the pixel position. Hence, one cannot distinguish the class membership of any of the training instances. This limiting case is shown in Fig. 2 (left column). If $c(\alpha) = 1$, we set $y_{\text{tr},\alpha} = -1$ and for $c(\alpha) = 2$ we set $y_{\text{tr},\alpha} = 1$. We now investigate the description of this task in the framework of Bayesian inference. The hidden variables $h^0_\alpha$ [Eq. (1)] in the input layer under a Gaussian prior on $V_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_v^2/N_{\text{dim}})$ follow a Gaussian process with kernel $K^{(0)}$ given by

$$K^0_{\alpha\beta} = \left\langle h^0_\alpha h^0_\beta \right\rangle_{V \sim \mathcal{N}\left(0, \frac{\sigma_v^2}{N_{\text{dim}}}\right)} \tag{10}$$

$$= \frac{\sigma_v^2}{N_{\text{dim}}} \sum_{i=1}^{N_{\text{dim}}} x_{\alpha i} x_{\beta i}. \tag{11}$$

Separability of the two classes is reflected in the structure of this input kernel $K^0$ as shown in Fig. 2: In the cases with $p = 0.8$ and $p = 1$, one can clearly distinguish blocks; the diagonal blocks represent intraclass overlaps, the off-diagonal blocks interclass overlaps. This is not the case for $p = 0.5$, where no clear block structure is visible. In the case of $p = 0.8$, one can further observe that the blocks are not as clear-cut as in the case $p = 1$, but rather noisy, similar to $p = 0.5$. This is due to the probabilistic realization of patterns, which induces stochasticity in the blocks of the input kernel $K^0$ [Eq. (10)]. To quantify this effect, based on the distribution of the pixel values [Eq. (9)], we compute the distribution of the entries of $K^0$ for the binary classification task. The mean of the overlap elements $\mu_{\alpha\beta}$ and their covariances $\Sigma_{(\alpha\beta)(\gamma\delta)}$ are defined via

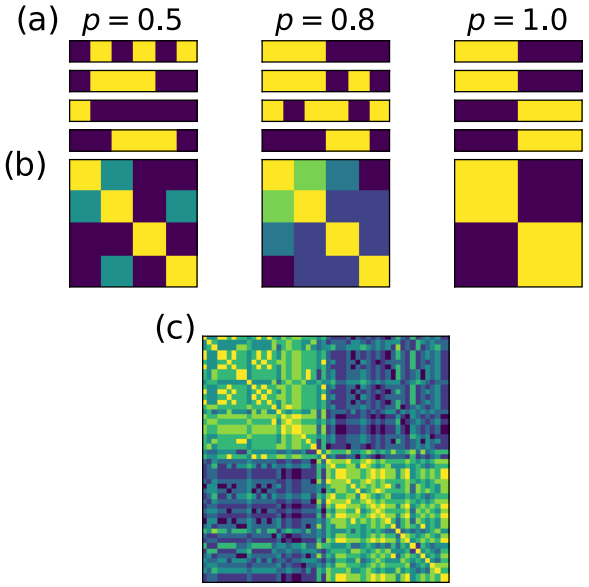



FIG. 2. Synthetic dataset. (a) Two sample vectors $x^{(1\leqslant\alpha\leqslant 4)}$ for each of the two classes (upper and lower). The values for the pixels can be $x_i^{(\alpha)} \in \{-1, 1\}$, where yellow indicates $x_i^{(\alpha)} = 1$ and blue $x_i^{(\alpha)} = -1$. The three columns correspond to three different settings of the pixel probability $p \in \{0.5, 0.8, 1\}$. (b) Empirical overlap matrices $K^x$ [Eq. (10)] for $D/2 = 2$ patterns of class 1 and $D/2 = 2$ patterns of class 2. The entries of the overlap matrices are $K^x_{\alpha\beta} \in [-1, 1]$. Darker colors indicate $K^x_{\alpha\beta} \approx -1$ and brighter colors correspond to $K^x_{\alpha\beta} \approx 1$. (c) Empirical overlap matrix $K^x$ for the same task as in panels (a) and (b) with $D = 50$. Other parameters: $N_{\text{dim}} = 6$.

$$\mu_{\alpha\beta} = \left\langle K^0_{\alpha\beta} \right\rangle, \tag{12}$$

$$\Sigma_{(\alpha\beta)(\gamma\delta)} = \left\langle \delta K^0_{\alpha\beta}\, \delta K^0_{\gamma\delta} \right\rangle, \tag{13}$$

$$\delta K^0_{\alpha\beta} = K^0_{\alpha\beta} - \mu_{\alpha\beta}, \tag{14}$$

where the expectation value $\langle\cdot\rangle$ is taken over drawings of $D$ training samples each. By construction, we have $\mu_{\alpha\beta} = \mu_{\beta\alpha}$. The covariance is further invariant under the exchange of $(\alpha, \beta) \leftrightarrow (\gamma, \delta)$ and, due to the symmetry of $K^0_{\alpha\beta} = K^0_{\beta\alpha}$, also under swapping $\alpha \leftrightarrow \beta$ and $\gamma \leftrightarrow \delta$ separately. In the artificial task setting, the parameter $p$, the pattern dimensionality $N_{\text{dim}}$, and the variance $\sigma_v^2/N_{\text{dim}}$ of each read-in weight $V_{ij}$ define the elements of $\mu_{\alpha\beta}$ and $\Sigma_{(\alpha\beta)(\gamma\delta)}$, which read (see the

Supplemental Material [30])

$$\mu_{\alpha\beta} = \sigma_v^2 \begin{cases} 1, & \alpha = \beta, \\ u, & c_\alpha = c_\beta, \\ -u, & c_\alpha \neq c_\beta, \end{cases}$$

$$\Sigma_{(\alpha\beta)(\alpha\beta)} = \frac{\sigma_v^4}{N_{\text{dim}}}\kappa,$$

$$\Sigma_{(\alpha\beta)(\alpha\delta)} = \frac{\sigma_v^4}{N_{\text{dim}}} \begin{cases} \nu, & \text{for } \begin{cases} c_\alpha = c_\beta = c_\delta, \\ c_\alpha \neq c_\beta = c_\delta, \end{cases} \\ -\nu, & \text{for } \begin{cases} c_\alpha = c_\beta \neq c_\delta, \\ c_\alpha = c_\delta \neq c_\beta, \end{cases} \end{cases} \tag{15}$$

with

$$\kappa := 1 - u^2,$$
$$\nu := u\,(1-u),$$
$$u := 4p(p-1) + 1.$$

In addition to this, the tensor elements of $\Sigma_{(\alpha\beta)(\gamma\delta)}$ are zero for the following index combinations because we fixed the value of $K^0_{\alpha\alpha}$ by construction:

$$\begin{aligned} \Sigma_{(\alpha\beta)(\gamma\delta)} &= 0, \quad \text{with } \alpha \neq \beta \neq \gamma \neq \delta, \\ \Sigma_{(\alpha\alpha)(\beta\gamma)} &= 0, \quad \text{with } \alpha \neq \beta \neq \gamma, \\ \Sigma_{(\alpha\alpha)(\beta\beta)} &= 0, \quad \text{with } \alpha \neq \beta, \\ \Sigma_{(\alpha\alpha)(\alpha\beta)} &= 0, \quad \text{with } \alpha \neq \beta, \\ \Sigma_{(\alpha\alpha)(\alpha\alpha)} &= 0, \quad \text{with } \alpha \neq \beta. \end{aligned} \tag{16}$$

The expressions for $\Sigma_{(\alpha\beta)(\alpha\beta)}$ and $\Sigma_{(\alpha\beta)(\alpha\delta)}$ in Eq. (15) show that the magnitude of the fluctuations is controlled through the parameter $p$ and the pattern dimensionality $N_{\text{dim}}$: The covariance $\Sigma$ is suppressed by a factor of $1/N_{\text{dim}}$ compared to the mean values $\mu$. Hence, we can use the pattern dimensionality $N_{\text{dim}}$ to investigate the influence of the strength of fluctuations. As illustrated in Fig. 1(a), the elements $\Sigma_{(\alpha\beta)(\alpha\beta)}$ denote the variance of individual entries of the kernel, while $\Sigma_{(\alpha\beta)(\alpha\gamma)}$ are covariances of entries across elements of a given row $\alpha$, visible as horizontal or vertical stripes in the color plot of the kernel.

Equation (15) implies, by construction, a Gaussian distribution of the elements $K^0_{\alpha\beta}$ as it only provides the first two cumulants. One can show that the higher-order cumulants of $K^0_{\alpha\beta}$ scale subleading in the pattern dimension and are hence suppressed by a factor $\mathcal{O}(1/N_{\text{dim}})$ compared to $\Sigma_{(\alpha\beta)(\gamma\delta)}$ (see the Supplemental Material [30]). The suppression of higher-order cumulants such as the variance with factors $\mathcal{O}(1/N_{\text{dim}})$ does not depend on the task setting and is, assuming the general structure of the kernel [Eq. (10)], general if one assumes independently distributed pixels.

## III. RESULTS

In this section, we derive the field theoretic formalism that allows us to compute the statistical properties of the inferred network output in Bayesian inference with a stochastic kernel. We show that the resulting process is non-Gaussian and reminiscent of a $\varphi^3 + \varphi^4$ theory. Specifically, we compute the mean of the predictive distribution of this process conditioned on the training data. This is achieved by employing systematic approximations with the help of Feynman diagrams.

Subsequently, we show that our results provide an accurate bound on the generalization capabilities of the network. We further discuss the implications of our analytic results for neural architecture search.

### A. Field theoretic description of Bayesian inference

#### 1. Bayesian inference with stochastic kernels

In general, a network implements a map from the inputs $x_\alpha$ to corresponding outputs $y_\alpha$. In particular, a model of the form in Eq. (1) implements a nonlinear map $\psi : \mathbb{R}^{N_{\text{dim}}} \to \mathbb{R}^{N_h}$ of the input $x_\alpha \in \mathbb{R}^{N_{\text{dim}}}$ to a hidden state $h_\alpha \in \mathbb{R}^{N_h}$. This map may also involve multiple hidden layers, biases, and nonlinear transformations. The readout weight $\mathbf{U} \in \mathbb{R}^{1\times N_h}$ links the scalar network output $y_\alpha \in \mathbb{R}$ and the transformed inputs $\psi(x_\alpha)$ with $1 \leqslant \alpha \leqslant D_{\text{tot}} = D + D_{\text{test}}$ that yields

$$y_\alpha = \mathbf{U}\,\psi(x_\alpha) + \xi_\alpha, \tag{17}$$

where $\xi_\alpha \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2_{\text{reg}})$ is a regularization noise in the same spirit as in Ref. [24]. We assume that the prior on the readout vector elements is a Gaussian $\mathbf{U}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_u^2/N_h)$. The distribution of the set of network outputs $y_{1\leqslant\alpha\leqslant D_{\text{tot}}}$ is then in the limit $N_h \to \infty$, a multivariate Gaussian [5]. The kernel matrix of this Gaussian is obtained by taking the expectation value with respect to the readout vector, which yields

$$\langle y_\alpha\, y_\beta\rangle_{\mathbf{U}} =: K^y_{\alpha\beta} = \sigma_u^2\, K^\psi_{\alpha\beta} + \delta_{\alpha\beta}\,\sigma^2_{\text{reg}}, \tag{18}$$

$$K^\psi_{\alpha\beta} = \frac{1}{N_h}\sum_{i=1}^{N_h} \psi_i(x_\alpha)\,\psi_i(x_\beta). \tag{19}$$

The kernel matrix $K^y_{\alpha\beta}$ describes the covariance of the network's output and hence depends on the kernel matrix $K^\psi_{\alpha\beta}$. The additional term $\delta_{\alpha\beta}\,\sigma^2_{\text{reg}}$ acts as a regularization term, which is also known as a ridge regression [31] or Tikhonov regularization [32]. In the context of neural networks, one can motivate the regularizer $\sigma^2_{\text{reg}}$ by using the $L^2$ regularization in the readout layer. This is also known as weight decay [33]. Introducing the regularizer $\sigma^2_{\text{reg}}$ is necessary to ensure that one can properly invert the matrix $K^y_{\alpha\beta}$, ensuring that the expressions [Eqs. (5) and (6)] are numerically stable.

Different drawings of sets of training data $\mathcal{D}_{\text{tr}}$ lead to different realizations of kernel matrices $K^\psi$ and $K^y$. The network output $y_\alpha$ hence follows a multivariate Gaussian with a stochastic kernel matrix $K^y$. A more formal derivation of the Gaussian statistics, including an argument for its validity in deep neural networks, can be found in Ref. [27]. A consistent derivation using field theoretical methods and corrections in terms for the width of the hidden layer $N_h$ for deep and recurrent networks has been presented in Ref. [28].

In general, the input kernel matrix $K^0$ [Eq. (10)] and the output kernel matrix $K^y$ are related in a nontrivial fashion, which depends on the specific network architecture at hand. From now on, we make an assumption on the stochasticity of $K^0$ and assume that the input kernel matrix $K^0$ is distributed

according to a multivariate Gaussian

$$K^0 \sim \mathcal{N}(\mu,\ \Sigma), \tag{20}$$

where $\mu$ and $\Sigma$ are given by Eqs. (12) and (13), respectively.

In the limit of large pattern dimensions $N_{\text{dim}} \gg 1$, this assumption is warranted for the kernel matrix $K^0$. This structure further assumes that the overlap statistics are unimodal, which is indeed mostly the case for data such as MNIST (see Appendix 4). Furthermore, we assume that this property holds for the output kernel matrix $K^y$ as well and that we can find a mapping from the mean $\mu$ and covariance $\Sigma$ of the input kernel to the mean $m$ and covariance $C$ of the output kernel $(\mu_{\alpha\beta},\ \Sigma_{(\alpha\beta)(\gamma\delta)}) \to (m_{\alpha\beta},\ C_{(\alpha\beta)(\gamma\delta)})$ so that $K^y$ is also distributed according to a multivariate Gaussian

$$K^y \sim \mathcal{N}(m,\ C). \tag{21}$$

For each realization $K^y_{\alpha\beta}$, the joint distribution of the network outputs $y_{1\leqslant\alpha\leqslant D_{\text{tot}}}$ corresponding to the training and test data points $\mathbf{x}$ follow a multivariate Gaussian

$$p(\mathbf{y}|\mathbf{x}) \sim \mathcal{N}(0, K^y). \tag{22}$$

The kernel allows us to compute the conditional probability $p(y_*|\mathbf{x}_{\text{tr}}, \mathbf{y}_{\text{tr}}, x_*)$, as defined in Eq. (3), for a test point $(x_*, y_*) \in \mathcal{D}_{\text{test}}$ conditioned on the data from the training set $(\mathbf{x}_{\text{tr}}, \mathbf{y}_{\text{tr}}) \in \mathcal{D}_{\text{tr}}$. This distribution is Gaussian with mean and variance given by Eqs. (5) and (6), respectively. It is our goal to take into account that $K^0$ is a stochastic quantity, which depends on the particular draw of the training and test datasets $(\mathbf{x}_{\text{tr}}, \mathbf{y}_{\text{tr}}) \in \mathcal{D}_{\text{tr}}$, $(x_*, y_*) \in \mathcal{D}_{\text{test}}$. The labels $\mathbf{y}_{\text{tr}}$ and $y_*$ are, by construction, deterministic and take either one of the values $\pm 1$. In the following, we investigate the dependence of the mean of the predictive distribution on the number of training samples, which we call the learning curve. A common assumption is that this learning curve is rather insensitive to the very realization of the chosen training points. Thus, we assume that the learning curve is self-averaging. The mean computed for a single draw of the training data is hence expected to agree well to the average over many such drawings. Under this assumption, it is sufficient to compute the data-averaged mean-inferred network output, which reduces to computing the disorder average of the following quantity:

$$\langle y_* \rangle_{K^y} = \left\langle K^y_{*\alpha} [K^y]^{-1}_{\alpha\beta} \right\rangle_{K^y} y_\beta. \tag{23}$$

To perform the disorder average and to compute perturbative corrections, we will follow these steps:

(1) construct a suitable dynamic moment-generating function $Z(K^y)$,

(2) propagate the input stochasticity to the network output $K^0_{\alpha\beta} \to K^y_{\alpha\beta}$,

(3) disorder average the functional using the model $K^y_{\alpha\beta} \sim \mathcal{N}(m_{\alpha\beta},\ C_{(\alpha\beta)(\gamma\delta)})$,

(4) and finally perform the computation of perturbative corrections using diagrammatic techniques.

#### 2. Constructing the dynamic moment-generating function

Our ultimate goal is to compute learning curves. Therefore, we want to evaluate the disorder-averaged mean-inferred network output [Eq. (23)]. Both the presence of two correlated random matrices and the fact that one of the matrices appears as an inverse complicate this process. One alternative route is to define the moment-generating function

$$Z(l_*) = \int dy_*\, \exp(l_* y_*) p(y_*|x_*, \mathbf{x}_{\text{tr}}, \mathbf{y}_{\text{tr}}) \tag{24}$$

$$= \frac{\int dy_*\, \exp(l_* y_*) p(y_*, \mathbf{y}_{\text{tr}}|x_*, \mathbf{x}_{\text{tr}})}{p(\mathbf{y}_{\text{tr}}|\mathbf{x}_{\text{tr}})} \tag{25}$$

$$=: \frac{\mathcal{Z}(l_*)}{\mathcal{Z}(0)}, \tag{26}$$

with joint Gaussian distributions $p(y_*, \mathbf{y}_{\text{tr}}|x_*, \mathbf{x}_{\text{tr}})$ and $p(\mathbf{y}_{\text{tr}}|\mathbf{x}_{\text{tr}})$ that can each be readily averaged over $K^y$. Equation (23) is then obtained as

$$\langle y_* \rangle_{K^y} = \frac{\partial}{\partial l_*} \left\langle \frac{\mathcal{Z}(l_*)}{\mathcal{Z}(0)} \right\rangle_{K^y} \Bigg|_{l_*=0}. \tag{27}$$

A complication of this approach is that the numerator and denominator cofluctuate. We here solve this problem by constructing a dynamic moment-generating function, where this normalization drops out [34]. Our goal is hence to design a dynamic process where a time-dependent observable is related to $y_*$, our mean-inferred network output. We hence define the linear process in the auxiliary variables $q_\alpha$:

$$\frac{\partial q_\alpha(t)}{\partial t} = -K^y_{\alpha\beta}\, q_\beta(t) + y_\alpha, \tag{28}$$

for $(x_\alpha, y_\alpha) \in \mathcal{D}_{\text{tr}}$. It follows that $q_\alpha(t \to \infty) = [K^y]^{-1}_{\alpha\beta}\, y_\beta$ is a fixed point. The fact that $K^y_{\alpha\beta}$ is a covariance matrix ensures that it is positive semidefinite and hence implies the convergence to a fixed point. We can obtain [Eq. (5)] $\langle y_* \rangle = K^y_{*\alpha} [K^y]^{-1}_{\alpha\beta}\, y_\beta$ from Eq. (28) as a linear readout of $q_\alpha(t \to \infty)$ with the matrix $K^y_{*\alpha}$. Using the Martin-Siggia-Rose-deDominicis-Janssen formalism [35–38], one can express this as the first functional derivative of the moment-generating function $Z(L_*, K^y)$ in frequency space:

$$Z(L_*, K^y) = \int \mathcal{D}\{Q, \tilde{Q}\} \exp(S(Q, \tilde{Q}, L_*)), \tag{29}$$

$$S(Q, \tilde{Q}, L_*) = \tilde{Q}^\top_\alpha (i\omega \mathbb{I} + K^y)_{\alpha\beta} Q_\beta - \tilde{Q}^0_\alpha y_\alpha + K^y_{*\alpha} L^\top_* Q_\alpha, \tag{30}$$

where $\tilde{Q}^\top_\alpha (\cdots) Q_\beta = \frac{1}{2\pi} \int d\omega \tilde{Q}_\alpha(-\omega) (\cdots) Q_\beta(\omega)$ and $\tilde{Q}_\alpha(\omega = 0) = \tilde{Q}^0_\alpha$. As $Z(L_*, K^y)$ is normalized such that $Z(0, K^y) = 1\ \forall\ K^y$, we can compute Eq. (23) by evaluating the functional derivative of the disorder-averaged moment-generating function $\overline{Z}(L_*)$ (see Appendix 1) at $t \to \infty$:

$$\overline{Z}(L_*) = \left\langle \int \mathcal{D}\{Q, \tilde{Q}\} \exp\left(S(Q, \tilde{Q}, L_*)\right) \right\rangle_{K^y}, \tag{31}$$

$$\langle y_* \rangle_{K^y} = \lim_{t\to\infty} \int d\omega \exp(i\omega t) \frac{\delta \overline{Z}(L_*)}{\delta L_*(-\omega)} \Bigg|_{L_*(-\omega)=0}. \tag{32}$$

By construction, the distribution of the kernel matrix entries $K^y_{\alpha\beta}$ is a multivariate Gaussian [Eq. (20)]. In the following, we will treat the stochasticity of $K^y_{\alpha\beta}$ perturbatively to gain insights into the influence of input stochasticity. Opposed to this approach, a common alternative route around the problem of fluctuating normalizations $\mathcal{Z}(0)$ is to consider the

cumulant-generating function $W(l_*) = \ln \mathcal{Z}(l_*)$ and to obtain $\langle y_* \rangle_{K^y} = \frac{\partial}{\partial l_*} \langle W(l_*) \rangle_{K^y}$, which, however, requires averaging the logarithm instead. This is commonly done with the replica trick [39,40].

### 3. Perturbative treatment of the disorder-averaged moment-generating function

To compute the disorder-averaged mean-inferred network output [Eq. (23)], we need to compute the disorder average of the dynamic moment-generating function $\overline{Z}(L_*) = \langle Z(L_*, K^y) \rangle_{K^y}$ and its functional derivative at $L_*(\omega) = 0$. Due to the linear appearance of $K^y$ in the action [Eq. (30)] and the Gaussian distribution for $K^y$ [Eq. (21)], we perform the disorder average directly and obtain the action

$$\overline{Z}(L_*) = \langle Z(L_*, K^y) \rangle_{K^y} \tag{33}$$

$$:= \int \mathcal{D}\{Q, \tilde{Q}\} \exp(\overline{S}(Q, \tilde{Q}, L_*)), \tag{34}$$

$$\begin{aligned} \overline{S}(Q, \tilde{Q}, L_*) = {} & \tilde{Q}_\alpha^\top [i\omega \mathbb{I}_{\alpha\beta} + m_{\alpha\beta}] Q_\beta \\ & - \tilde{Q}_\alpha^0 y_\alpha + m_{*\alpha} L_*^\top Q_\alpha \\ & + \tfrac{1}{2} C_{(\alpha\beta)(\gamma\delta)} \tilde{Q}_\alpha^\top Q_\beta \tilde{Q}_\gamma^\top Q_\delta \\ & + C_{(*\alpha)(\beta\gamma)} L_*^\top Q_\alpha \tilde{Q}_\beta^\top Q_\gamma + \mathcal{O}(L_*^2), \end{aligned} \tag{35, 36}$$

with $\tilde{Q}^0 := \tilde{Q}(\omega = 0)$ (for details, see Appendix 1). As we ultimately aim to obtain corrections for the mean-inferred network output $\langle y_* \rangle_{K^y}$, we utilize the action in Eq. (36) and established results from field theory to derive the leading-order terms as well as perturbative corrections diagrammatically. The presence of the variance and covariance terms in Eq. (36) introduces corrective factors, which cannot appear in the zeroth-order approximation, which corresponds to the homogeneous kernel that neglects fluctuations in $K^y$ by setting $C_{(\alpha\beta)(\gamma\delta)} = 0$. This will provide us with the tools to derive an asymptotic bound for the mean-inferred network output $\langle y^* \rangle$ in the case of an infinitely large training dataset. This bound is directly controlled by the variability in the data. We provide empirical evidence for our theoretical results for linear, nonlinear, and deep-kernel settings and show how the results could serve as indications to aid neural architecture search based on the statistical properties of the underlying dataset.

### 4. Field theoretic elements to compute the mean-inferred network output $\langle y_* \rangle_{K^y}$

The field theoretic description of the inference problem in the form of an action [Eq. (36)] allows us to derive perturbative expressions for the statistics of the inferred network output $\langle y_* \rangle_{K^y}$ in a diagrammatic manner. This diagrammatic treatment for perturbative calculations is a powerful tool and is standard practice in statistical physics [41], data analysis, and signal reconstruction [42], and more recently in the investigation of artificial neural networks [43,44].

Comparing the action [Eq. (36)] to prototypical expressions from classical statistical field theory such as the $\varphi^3 + \varphi^4$ theory [38,41], one can similarly associate the elements of a field theory:

(1) $-\tilde{Q}_\alpha^0 y_\alpha \doteq$  is a monopole term,

(2) $m_{*\alpha} L_*^\top Q_\alpha \doteq$  is a source term,

(3) $\Delta_{\alpha\beta} := (-i\omega \mathbb{I} - m)^{-1}_{\alpha\beta} \doteq$  is a propagator that connects the fields $Q_\alpha(\omega)$ and $\tilde{Q}_\beta(-\omega)$,

(4) $C_{(*\alpha)(\beta\gamma)} L_*^\top Q_\alpha \tilde{Q}_\beta^\top Q_\gamma \doteq$  is a three-point vertex,

(5) $\frac{1}{2} C_{(\alpha\beta)(\gamma\delta)} \tilde{Q}_\alpha^\top Q_\beta \tilde{Q}_\gamma^\top Q_\delta \doteq$  is a four-point vertex.

The following rules for Feynman diagrams simplify calculations:

(1) To obtain corrections to first order in $C \sim \mathcal{O}(1/N_{\text{dim}})$, one has to compute all diagrams with a single vertex (three-point or four-point) [38]. This approach assumes that the interaction terms $C_{(\alpha\beta)(\gamma\delta)}$ that stem from the variability of the data are small compared to the mean $m_{\alpha\beta}$. In the case of strong heterogeneity, one cannot use a conventional expansion in the number of vertices $C_{(\alpha\beta)(\gamma\delta)}$ and would have to resort to other methods.

(2) Vertices, source terms, and monopoles have to be connected with one another using the propagator $\Delta_{\alpha\beta} = (-i\omega \mathbb{I} - m)^{-1}_{\alpha\beta}$ that couples $Q_\alpha(\omega)$ and $\tilde{Q}_\beta(-\omega)$ with each other.

(3) We only need diagrams with a single external source term $L_*$ because we seek corrections to the mean-inferred network output.

(4) The structure of the integrals appearing in the four-point and three-point vertices containing $C_{(\alpha\beta)(\gamma\delta)}$ with contractions by $\Delta_{\alpha\beta}$ or $\Delta_{\gamma\delta}$ within a pair of indices $(\alpha\beta)$ or $(\gamma\delta)$ yield vanishing contributions; such diagrams are known as closed response loops [38]. This is because the propagator $\Delta_{\alpha\beta}(t - s)$ in time domain vanishes for $t = s$, which corresponds to the integral $\int d\omega\, \Delta_{\alpha\beta}(\omega)$ over all frequencies $\omega$. A detailed explanation is given in Appendix 1.

(5) Due to frequency conservation at the vertices in the $\frac{1}{2} \tilde{Q}_\alpha^\top Q_\beta\, C_{(\alpha\beta)(\gamma\delta)}\, \tilde{Q}_\gamma^\top Q_\delta$ and since by point (4) above we only need to consider contractions by $\Delta_{\beta\gamma}$ or $\Delta_{\delta\alpha}$ by attaching the external legs, for $t \to \infty$ all frequencies are constrained to $\omega = 0$. Therefore propagators within a loop are replaced $\Delta_{\alpha\beta} = (-i\omega \mathbb{I} - m)^{-1}_{\alpha\beta} \to -(m^{-1})_{\alpha\beta}$.

These rules directly yield that the corrections for the disorder-averaged mean-inferred network to first order in $C_{(\alpha\beta)(\gamma\delta)}$ can only include the diagrams (see Appendix 1)

$\langle y_* \rangle \doteq$ 


$$+ \mathcal{O}\left(C^2\right), \tag{37}$$

which translate to our main result

$$\begin{aligned} \langle y_* \rangle_{0+1} = {} & m_{*\alpha} m^{-1}_{\alpha\beta}\, y_\beta + m_{*\epsilon} m^{-1}_{\epsilon\alpha} C_{(\alpha\beta)(\gamma\delta)} m^{-1}_{\beta\gamma}\, m^{-1}_{\delta\rho}\, y_\rho \\ & - C_{(*\alpha)(\beta\gamma)} m^{-1}_{\alpha\beta} m^{-1}_{\gamma\delta}\, y_\delta + \mathcal{O}(C^2). \end{aligned} \tag{38}$$

We here define the first line as the zeroth-order approximation $\langle y_* \rangle_0 := m_{*\alpha} m^{-1}_{\alpha\beta}\, y_\beta$, which has the same form as Eq. (5), and the latter two lines as perturbative corrections $\langle y_* \rangle_1 = \mathcal{O}(C)$ that are of linear order in $C$.

The mean [Eq. (38)] of the predictive distribution is a central quantity that controls the generalization error; concretely, it enters in the bias term of a bias-variance decomposition [Eq. (A108)], as shown in Appendix 7. An extension of this

approach to also obtain the posterior variance on an unseen test point is presented in that Appendix as well. This allows one to also compute the expected mean squared error loss on a test point, a commonly used quantity to assess generalization.

#### 5. *Evaluation of expressions for block-structured overlap matrices*

To evaluate the first-order correction $\langle y_* \rangle_1$ in Eq. (38), we make use of the fact that Bayesian inference is insensitive to the order in which the training data are presented. We are hence free to assume that all training samples of one class are presented en bloc. Moreover, supervised learning assumes that all training samples are drawn from the same distribution. As a result, the statistics is homogeneous across blocks of indices that belong to the same class. The propagators $-m^{-1}_{\alpha\beta}$ and interaction vertices $C_{(\alpha\beta)(\gamma\delta)}$ and $C_{(*\alpha)(\beta\gamma)}$, correspondingly, have a block structure. To obtain an understanding on how variability of the data and hence heterogeneous kernels affect the ability to make predictions, we consider the simplest yet nontrivial case of binary classification where we have two such blocks.

In this section, we focus on the overlap statistics given by the artificial dataset described in Sec. II C. This dataset entails certain symmetries. Generalizing the expressions to a less symmetric task is straightforward, but lengthy, and is deferred to Appendix 5. For the classification task, with two classes $c_\alpha \in \{1, 2\}$, the structure for the mean overlaps $\mu_{\alpha\beta}$ and their covariance $\Sigma_{(\alpha\beta)(\gamma\delta)}$ at the read-in layer of the network given by Eq. (15) are inherited by the mean $m_{\alpha\beta}$ and the covariance $C_{(\alpha\beta)(\gamma\delta)}$ of the overlap matrix at the output of the network. In particular, all quantities can be expressed in terms of only four parameters $a, b, K$, and $v$ whose values, however, depend on the network architecture and will be given for linear and nonlinear networks below. For four indices $\alpha, \beta, \gamma$, and $\delta$ that are all different,

$$m_{\alpha\alpha} = a, \tag{39}$$

$$m_{\alpha\beta} = \begin{cases} b, & c_\alpha = c_\beta, \\ -b, & c_\alpha \neq c_\beta, \end{cases} \tag{40}$$

$$C_{(\alpha\alpha),(\gamma\delta)} = 0, \tag{41}$$

$$C_{(\alpha\beta)(\alpha\beta)} = K, \tag{42}$$

$$C_{(\alpha\beta)(\alpha\delta)} = \begin{cases} v, & c_\alpha = c_\beta = c_\delta, \quad c_\alpha \neq c_\beta = c_\delta, \\ -v, & c_\alpha = c_\beta \neq c_\delta, \quad c_\alpha = c_\delta \neq c_\beta. \end{cases} \tag{43}$$

This symmetry further assumes that the network does not have biases and utilizes point-symmetric activation functions $\phi(x)$ such as $\phi(x) = \mathrm{erf}(x)$. In general, all tensors are symmetric with regard to swapping $\alpha \leftrightarrow \beta$ as well as $\gamma \leftrightarrow \delta$ and the tensor $C_{(\alpha\beta)(\gamma\delta)}$ is invariant under swaps of the index pairs $(\alpha\beta) \leftrightarrow (\gamma\delta)$. We further assume that the class label for class 1 is $y$ and that the class label for class 2 is $-y$. In subsequent calculations and experiments, we consider the prediction for the class $y = -1$. In particular, we hence set $y_\alpha = -1 \; \forall \; \alpha \in c_1$ and $y_\alpha = 1 \; \forall \; \alpha \in c_2$. For the test point, we always choose $* \in c_1$ and hence the true label for the test point always reads $y_* = -1$. As the choice of the labels $y$ is in fact arbitrary, we settle for a symmetric choice in order to simplify analytical calculations.

This setting is quite natural, as it captures the presence of differing mean intraclass and interclass overlaps. Further $K$ and $v$ capture two different sources of variability. Whereas $K$ is associated with the presence of i.i.d. distributed variability on each entry of the overlap matrix separately, $v$ corresponds to variability stemming from correlations between different patterns. Using the properties in Eq. (43), one can evaluate Eq. (38) for the inference of test points $*$ within class $c_1$ on a balanced training set with $D$ samples explicitly to (see the Supplemental Material [30])

$$\langle y_* \rangle_0 = Dgy, \tag{44}$$

$$\begin{aligned} \langle y_* \rangle_1 = {} & v g \hat{y} (q_1 + 3q_2)(D^3 - 3D^2 + 2D) \\ & + K g \hat{y} (q_1 + q_2)(D^2 - D) - v \hat{y} (q_1 + q_2)(D^2 - D) \\ & + \mathcal{O}\big(C^2_{(\alpha\beta)(\gamma\delta)}\big), \quad \text{for} \quad * \in c_1, \end{aligned} \tag{45}$$

with the additional variables

$$g = \frac{b}{(a-b) + bD}, \tag{46}$$

$$q_2 = -\frac{b}{(a-b) + bD}, \tag{47}$$

$$q_1 = \frac{1}{a-b} + q_2, \tag{48}$$

$$\hat{y} = \frac{y}{(a-b) + bD}, \tag{49}$$

which stem from the analytic inversion of block matrices (using the approach from Ref. [45]; see the Supplemental Material [30]). Carefully treating the dependencies of the parameters in Eqs. (45) and (49), one can compute the limit $D \gg 1$ and show that the $\mathcal{O}(1)$ behavior of Eq. (45) for test points $* \in c_1$ for the zeroth-order approximation, $\lim_{D\to\infty} \langle y_* \rangle_0 := \langle y_* \rangle_0^{(\infty)}$, and the first-order correction, $\lim_{D\to\infty} \langle y_* \rangle_1 := \langle y_* \rangle_1^{(\infty)}$, is given by

$$\langle y_* \rangle_0^{(\infty)} = y, \tag{50}$$

$$\langle y_* \rangle_1^{(\infty)} = \frac{y}{(a-b)b} \left( (K - 4v) - v \frac{a-b}{b} \right). \tag{51}$$

This result implies that regardless of the amount of training data $D$, the lowest value of the limiting behavior is controlled by the data variability represented by $v$ and $K$. Due to the symmetric nature of the task setting, neither the limiting behavior [Eq. (51)] nor the original expression [Eq. (45)] explicitly show the dependence on the relative number of training samples in the two respective classes $c_{1,2}$. This is due to the fact that the task setup in Eq. (43) is symmetric.

In the case of asymmetric statistics as they typically appear in real-world datasets, one can still derive exact analytical results for the mean of the predictive distribution such as Eq. (45), which is done in Appendix 5. Concretely, Eq. (A102) is the general expression. The asymmetric statistics appear in the form of a larger set of distinct elements of the tensors $m_{\alpha\beta}$ and $C_{(\alpha\beta)(\gamma\delta)}$. The results are hence structurally similar, yet more verbose. Moreover, the difference between variance $a$ and covariance $b$ enters the expression in a nontrivial manner.

Using those results, we will investigate the implications for linear, nonlinear, and deep kernels using the artificial dataset

(Sec. II C) as well as real-world data. In the case of the Ising task setting, it can be shown that the first-order correction allows one to compute meaningful limits for the task performance in the sense that they are not lower than the Bayes optimal error. If, on the other hand, one solely considers the limiting behavior of the zeroth-order approximation, one may falsely conclude that the network predictor would be better than the Bayes optimal one (see Appendix 10).

### B. Applications to linear, nonlinear, and deep nonlinear NNGP kernels

#### 1. Linear kernel

Before going to the nonlinear case, let us investigate the implications of Eqs. (45) and (51) for a simple one-layer linear network. We assume that our network consists of a read-in weight $\mathbf{V} \in \mathbb{R}^{1\times N_{\text{dim}}}$; $\mathbf{V}_i \sim \mathcal{N}(0, \sigma_v^2/N_{\text{dim}})$, which maps the $N_{\text{dim}}$-dimensional input vector to a one-dimensional output space. Including a regularization noise, the output hence reads

$$y_\alpha = \mathbf{V} x_\alpha + \xi_\alpha. \tag{52}$$

In this particular case, the read-in, readout, and hidden weights in the general setup [Eq. (1)] coincide with each other. Computing the average with respect to the weights $\mathbf{V}$ yields the kernel

$$K^y_{\alpha\beta} = \langle y_\alpha y_\beta \rangle_{\mathbf{V}} = K^0_{\alpha\beta} + \delta_{\alpha\beta}\, \sigma^2_{\text{reg}}, \tag{53}$$

where $K^0_{\alpha\beta}$ is given by Eq. (10); it is hence a rescaled version of the overlap of the input vectors and the variance of the regularization noise.

We now assume that the matrix elements of the input-data overlap [Eq. (53)] are distributed according to a multivariate Gaussian [Eq. (20)].

As the mean and the covariance of the entries $K^y_{\alpha\beta}$ are given by the statistics [Eq. (15)], we evaluate Eqs. (45) and (51) with

$$\begin{aligned} a^{(\text{Lin})} &= \sigma_v^2 + \sigma^2_{\text{reg}}, \quad b^{(\text{Lin})} = \sigma_v^2\, u, \\ K^{(\text{Lin})} &= \frac{\sigma_v^4}{N_{\text{dim}}}\,(1-u^2), \\ v^{(\text{Lin})} &= \frac{\sigma_v^4}{N_{\text{dim}}}\,u\,(1-u), \\ u &:= 4p(p-1)+1. \end{aligned} \tag{54}$$

The asymptotic result for the first-order correction, assuming that $\sigma_v^2 \neq 0$, can hence be evaluated, assuming $p \neq 0.5$, as

$$\langle y_* \rangle_1^{(\infty)} = \frac{y_1 \sigma_v^2 \frac{(1-u)}{N_{\text{dim}}}}{\left(\sigma_v^2(1-u)+\sigma^2_{\text{reg}}\right)u}\left(-2u - \frac{\sigma^2_{\text{reg}}}{\sigma_v^2}\right). \tag{55}$$

Using this explicit form of $\langle y_* \rangle_1^{(\infty)}$, one can see

(1) as $u \in [0, 1]$ the corrections are always negative and hence provide a less optimistic estimate for the generalization compared to the zeroth-order approximation;

(2) in the limit $\sigma_v^2 \to \infty$, the regularizer in Eq. (55) becomes irrelevant and the matrix inversion becomes unstable.

(3) Taking $\sigma_v^2 \to 0$ yields a setting where constructing the limiting formula [Eq. (55)] is not useful, as all relevant quantities [Eq. (45)] like $g, v, K \to 0$ vanish; hence, the inference yields zero that is consistent with our intuition: $\sigma_v^2 \to 0$ implies that only the regularizer decides, which is unbiased with regard to the class membership of the data. Hence, the kernel cannot make any prediction that is substantially informed by the data.

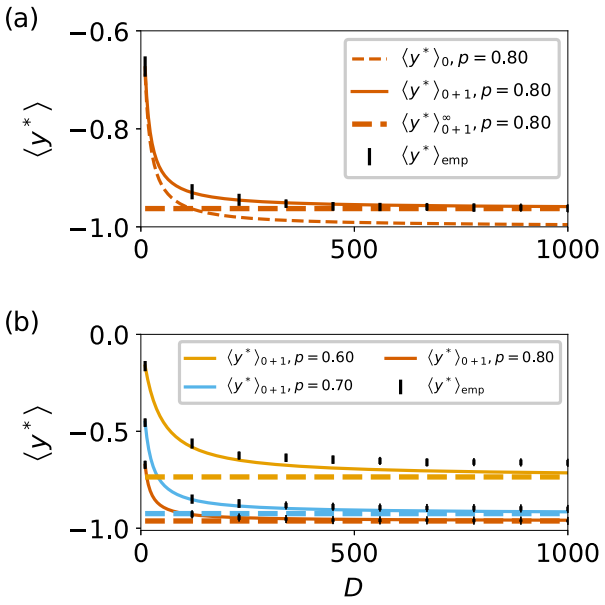



FIG. 3. Predictive mean in linear regression with heterogeneous kernel. (a) Comparison of empirical data (bars, see Appendix 8 for details), zeroth-order approximation $\langle y^* \rangle_0$ (dashed), first-order corrections $\langle y^* \rangle_{0+1}$ (solid), and the asymptotic value $\langle y^* \rangle^\infty_{0+1}$ in the case of infinite training data (dashed horizontal line) for a one-layer linear network with $\sigma_v^2 = 1$, $\sigma^2_{\text{reg}} = 0.8$, $N_{\text{dim}} = 50$, and $p = 0.8$. (b) Comparison of empirical inference data (bars) with first-order results $\langle y^* \rangle_{0+1}$ (solid line) and asymptotic values $\langle y^* \rangle^\infty_{0+1}$ for $p = 0.6$ (orange), $p = 0.7$ (blue), and $p = 0.8$ (red). Empirical results display mean and standard deviation over 50 trials with 2000 test points per trial.

Figure 3 shows that the zeroth-order approximation $\langle y^* \rangle_0$, even though it is able to capture some dependence on the amount of training data, is indeed too optimistic and predicts a mean-inferred network output closer to its negative target value $y = -1$ than numerically obtained. The first-order correction on the other hand is able to reliably predict the results. Furthermore, the limiting results $D \to \infty$ match the numerical results for different task settings $p$. These limiting results are consistently higher than the zeroth-order approximation $\langle y_* \rangle_0$ and depend on the level of data variability. Deviations of the empirical results from the theory in the case $p = 0.6$ could stem from the fact that for $p = 0.5$ the fluctuations are maximal and our theory assumes small fluctuations.

Empirical results are here and in the following obtained from the known expressions of the mean of the predictive distribution [Eq. (5)] of Bayesian inference on the Gaussian process, applied to datasets of sizes given by the $x$ axis in Eq. (14). The mean is averaged over multiple realizations of the training and test sets, as described in detail in Appendix 8.

#### 2. Nonlinear kernel

We will now investigate how the nonlinearities $\phi$ present in typical network architectures [Eq. (1)] influence our results for the learning curve [Eqs. (45) and (51)].

As the ansatz in Sec. III A does not make any assumption, apart from Gaussianity, on the overlap matrix $K^y$, the results presented in Sec. III A 5 are general. One can use the knowledge of the statistics of the overlap matrix in the read-in layer $K^0$ in Eq. (15) to extend the result [Eq. (45)] to both nonlinear and deep feedforward neural networks.

As in Sec. III B 1, we start with the assumption that the input kernel matrix is distributed according to a multivariate Gaussian: $K^0_{\alpha\beta} \sim \mathcal{N}(\mu_{\alpha\beta}, \Sigma_{(\alpha\beta)(\gamma\delta)})$. In the nonlinear case, we consider a read-in layer $\mathbf{V} \in \mathbb{R}^{N_h \times N_{\text{dim}}}; V_{i,j} \sim \mathcal{N}(0, \sigma_v^2/N_{\text{dim}})$, which maps the inputs to the hidden-state space, and a separate readout layer $\mathbf{W} \in \mathbb{R}^{1\times N_h}; W_i \sim \mathcal{N}(0, \sigma_w^2/N_h)$, obtaining a neural network with a single hidden layer

$$h^{(0)}_\alpha = \mathbf{V} x_\alpha,$$
$$y_\alpha = \mathbf{W}\phi\left(h^{(0)}_\alpha\right) + \xi_\alpha \tag{56}$$

and network kernel

$$\langle y_\alpha y_\beta \rangle_{\mathbf{V},\mathbf{W}} = \frac{\sigma_w^2}{N_h} \sum_{i=1}^{N_h} \left\langle \phi\left(h^{(0)}_{\alpha i}\right) \phi\left(h^{(0)}_{\beta i}\right) \right\rangle_{\mathbf{V}} + \delta_{\alpha\beta}\, \sigma^2_{\text{reg}}. \tag{57}$$

As we consider the limit $N_h \to \infty$, one can replace the empirical average $\frac{1}{N_h}\sum_{i=1}^{N_h} \cdots$ with a distributional average $\frac{1}{N_h}\sum_{i=1}^{N_h} \cdots \to \langle \cdots \rangle_{\mathbf{h}^{(0)}}$ [7,46]. This yields the following result for the kernel matrix $K^y_{\alpha\beta}$ of the multivariate Gaussian

$$K^y_{\alpha\beta} \underbrace{\to}_{N_h\to\infty} \sigma_w^2 \left\langle \phi^0_\alpha \phi^0_\beta \right\rangle_{\mathbf{h}^{(0)},\mathbf{V}} + \delta_{\alpha\beta}\, \sigma^2_{\text{reg}}, \tag{58}$$

where we introduced the shorthand $\langle \phi(h^{(0)}_\alpha)\phi(h^{(0)}_\beta)\rangle := \langle \phi^0_\alpha \phi^0_\beta \rangle$. The expectation over the hidden states $h^{(0)}_\alpha$ and $h^{(0)}_\beta$ is with regard to the Gaussian

$$\begin{pmatrix} h^{(0)}_\alpha \\ h^{(0)}_\beta \end{pmatrix} \sim \mathcal{N}\left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} K^0_{\alpha\alpha} & K^0_{\alpha\beta} \\ K^0_{\beta\alpha} & K^0_{\beta\beta} \end{pmatrix} \right), \tag{59}$$

with the variance $K^0_{\alpha\alpha}$ and the covariance $K^0_{\alpha\beta}$ given by Eq. (10). Evaluating the Gaussian integrals in Eq. (58) is analytically possible in certain limiting cases [26,47]. For an erf-activation function, as a prototype of a saturating activation function, this average yields

$$\left\langle \phi^0_\alpha \phi^0_\alpha \right\rangle_{\mathbf{h}^{(0)}} = \frac{4}{\pi} \arctan\left(\sqrt{1 + 4K^0_{\alpha\alpha}}\right) - 1, \tag{60}$$

$$\left\langle \phi^0_\alpha \phi^0_\beta \right\rangle_{\mathbf{h}^{(0)}} = \frac{2}{\pi} \arcsin\left( \frac{2K^0_{\alpha\beta}}{1 + 2K^0_{\alpha\alpha}} \right). \tag{61}$$

Equation (58) hence provides information on how the mean overlap $m_{\alpha\beta}$ changes due to the application of the nonlinearity $\phi(\cdot)$, fixing the parameters $a$, $b$, $K$, and $v$ of the general form [Eq. (43)] as

$$a^{(\text{Non-lin})} = K^y_{\alpha\alpha} = \sigma_w^2 \left\langle \phi^0_\alpha \phi^0_\alpha \right\rangle_{\mathbf{h}^{(0)}} + \sigma^2_{\text{reg}}, \tag{62}$$

$$b^{(\text{Non-lin})} = K^y_{\alpha\beta} = \sigma_w^2 \left\langle \phi^0_\alpha \phi^0_\beta \right\rangle_{\mathbf{h}^{(0)}}, \tag{63}$$

where the averages over $h^{(0)}$ are evaluated with regard to the Gaussian [Eq. (59)] for $\phi(x) = \text{erf}(x)$. We further require in Eq. (63) that $\alpha \neq \beta$, $c(\alpha) = c(\beta)$.

To evaluate the corrections in Eq. (45), we also need to understand how the presence of the nonlinearity $\phi(x)$ shapes the parameters $K$ and $v$ that control the variability. Under the assumption of small covariance $\Sigma_{(\alpha\beta)(\gamma\delta)}$, one can use Eq. (61) to compute $C_{(\alpha\beta)(\gamma\delta)}$ using linear response theory. As $K^0_{\alpha\beta}$ is stochastic and provided by Eq. (20), we decompose $K^0_{\alpha\beta}$ into a deterministic kernel $\mu_{\alpha\beta}$ and a stochastic perturbation $\eta_{\alpha\beta} \sim \mathcal{N}(0, \Sigma_{(\alpha\beta)(\gamma\delta)})$. Linearizing Eq. (63) around $\mu_{\alpha\beta}$ via Price's theorem [48,49], the stochasticity in the readout layer yields (see Appendix 1)

$$C_{(\alpha\beta)(\gamma\delta)} = \sigma_w^4\, K^{(\phi')}_{\alpha\beta}\, K^{(\phi')}_{\gamma\delta}\, \Sigma_{(\alpha\beta)(\gamma\delta)}, \tag{64}$$

$$K^{(\phi')}_{\alpha\beta} := \left\langle \phi'\left(h^{(0)}_\alpha\right) \phi'\left(h^{(0)}_\beta\right) \right\rangle, \tag{65}$$

where $h^{(0)}$ is distributed as in Eq. (59). This clearly shows that the variability simply transforms with a prefactor

$$K^{(\text{Non-lin})} = \sigma_w^4\, K^{(\phi')}_{\alpha\beta}\, K^{(\phi')}_{\alpha\beta}\, \kappa, \tag{66}$$

$$v^{(\text{Non-lin})} = \sigma_w^4 K^{(\phi')}_{\alpha\beta}\, K^{(\phi')}_{\alpha\delta}\, v, \tag{67}$$

with $\kappa$ and $v$ defined as in Eq. (15). Evaluating the integral in $\langle \phi'(h^{(0)}_\alpha)\phi'(h^{(0)}_\beta)\rangle$ is hard in general. In fact, the integral that occurs is equivalent to the one in Ref. [50] for the Lyapunov exponent and, equivalently, in Refs. [7,8] for the susceptibility in the propagation of information in deep feedforward neural networks. This is consistent with the assumption that our treatment of the nonlinearity follows a linear response approach as in Ref. [50]. For the erf activation, we can evaluate the kernel $K^{(\phi')}_{\alpha\beta}$ as

$$K^{(\phi')}_{\alpha\beta} = \frac{4}{\pi(1 + 2a^{(0)})} \left( 1 - \left( \frac{2b^{(0)}}{1 + 2a^{(0)}} \right)^2 \right)^{-\frac{1}{2}}, \tag{68}$$

$$a^{(0)} = \sigma_v^2, \quad b^{(0)} = \sigma_v^2 u, \tag{69}$$

$$u = 4p(p-1) + 1, \tag{70}$$

which allows us to evaluate Eq. (67). Already in the one hidden-layer setting, we can see that the behavior is qualitatively different from a linear setting: $K^{(\text{Non-lin})}$ and $v^{(\text{Non-lin})}$ scale with a linear factor THAT now also involves the parameter $\sigma_v^2$ in a nonlinear manner.

#### 3. Multilayer kernel

So far, we considered single-layer networks. However, in practice the application of multilayer networks is often necessary. One can straightforwardly extend the results from the nonlinear case (Sec. III B 2) to the deep nonlinear case. We consider the architecture introduced in Eq. (1), where the variable $L$ denotes the number of hidden layers and $1 \leqslant l \leqslant L$ is the layer index. Similar to the computations in Sec. III B 2, one can derive a set of relations to obtain $K^y_{\alpha\beta}$:

$$K^0_{\alpha\beta} = \frac{\sigma_v^2}{N_{\text{dim}}} K^x_{\alpha\beta}, \tag{71}$$

$$K^{(\phi)l}_{\alpha\beta} = \sigma_w^2 \left\langle \phi\left(h^{(l-1)}_\alpha\right) \phi\left(h^{(l-1)}_\beta\right) \right\rangle, \tag{72}$$

$$K^y_{\alpha\beta} = \sigma_u^2 \left\langle \phi\left(h^{(L)}_\alpha\right) \phi\left(h^{(L)}_\beta\right) \right\rangle + \delta_{\alpha\beta}\, \sigma^2_{\text{reg}}, \tag{73}$$

with the setting $K^0 \sim \mathcal{N}(\mu, \Sigma)$. As Refs. [7,8,51] showed for feedforward networks, deep nonlinear networks strongly alter both the variance and the covariance. So, we expect them to

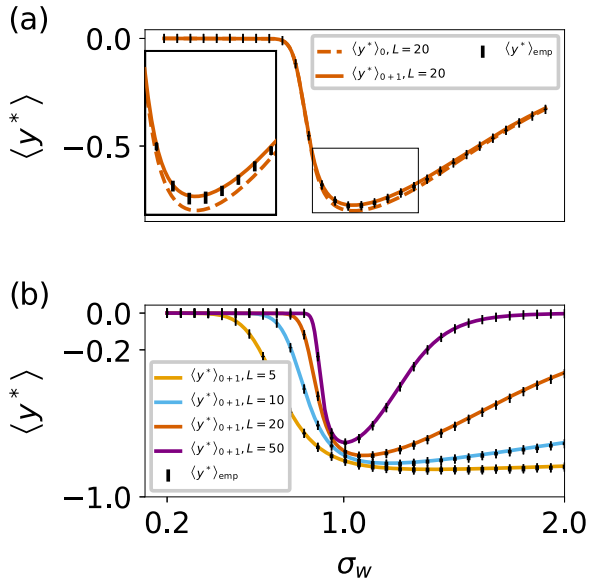



FIG. 4. Predictive mean in a deep nonlinear feedforward network with heterogeneous kernel. (a) Comparison of mean-inferred network output for nonlinear network with $\phi(x) = \mathrm{erf}(x)$, $L = 20$ for different values of the gain $\sigma_w$. The figure displays numerical results (bars), zeroth-order approximation (dashed), and first-order corrections (solid). (b) Similar comparison as in panel (a) for different network depths $L = 5$, 10, 20, and 50. In all settings, we used $N_{\mathrm{dim}} = 50$ for $D = 100$, $p = 0.8$, $\sigma_v^2 = 1$, and $\sigma_{\mathrm{reg}}^2 = 1$. Empirical results display mean and standard deviation over 1000 trials with 1000 test points per trial.

influence the generalization properties. In order to understand how the fluctuations $\Sigma_{(\alpha\beta)(\gamma\delta)}$ transform through propagation, one can employ the chain rule to linearize Eq. (73) and obtain

$$C_{(\alpha\beta)(\gamma\delta)} = \sigma_u^4 \prod_{l=1}^{L+1} \left[ K_{\alpha\beta}^{(\phi')l} K_{\gamma\delta}^{(\phi')l} \right] \Sigma_{(\alpha\beta)(\gamma\delta)}. \tag{74}$$

A systematic derivation of this result as the leading-order fluctuation correction in $N_h^{-1}$ is found in the Appendix of Ref. [28], whereas a derivation using a linear response approach is part of Appendix 1.

Equations (73) and (74) show that the kernel performance will depend on the nonlinearity $\phi$, the variances $\sigma_v^2$, $\sigma_w^2$, $\sigma_u^2$, and the network depth $L$.

Figure 4(a) shows the comparison of the mean-inferred network output $\langle y^* \rangle$ for the true test label $y = -1$ between empirical results and the first-order corrections. The regime ($\sigma_w^2 < 1$) in which the kernel vanishes leads to a poor performance. The marginal regime ($\sigma_w^2 \simeq 1$) provides a better choice for the overall network performance. Figure 4(b) shows that the maximum absolute value for the predictive mean is achieved slightly in the supercritical regime $\sigma_w^2 > 1$. With larger number of layers, the optimum becomes more and more pronounced and approaches the critical value $\sigma_w^2 = 1$ from above. The optimum for the predictive mean to occur slightly in the supercritical regime may be surprising with regard to the expectation that network trainability peaks precisely at $\sigma_w^2 = 1$ [7]. In particular at shallow depths, the optimum becomes very wide and shifts to $\sigma_w > 1$. For few layers, even at $\sigma_w^2 > 1$ the increase of variance $K_{\alpha\alpha}^y$ per layer remains moderate and stays within the dynamical range of the activation function. Thus, differences in covariance are faithfully transmitted by the kernel and hence allow for accurate predictions. The theory including corrections to linear order throughout matches the empirical results and hence provides good estimates for choosing the kernel architecture.

As the depth of the network increases, the absolute value of the mean of the predictive distribution becomes smaller. This tendency is already visible in the curves without fluctuation corrections and can hence can be understood based on the level of the mean kernels: It is essentially controlled by the overlap within a class compared to the overlap across classes. This difference progressively declines with depth, where the decline is slowest close to the critical point and faster above and below it. A quantitative analysis is given in Appendix 9. This behavior is qualitatively different from what one would expect from finite-sized neural networks [17,19,52]: Here, feature learning enables networks to adapt the layerwise kernels to the structure of the task [53]. It also appears that deeper networks are able to adapt better, which typically reduces the training error. We here, instead, study the lazy regime, which hence may differ from the rich regime found in finite-sized networks.

#### *4. Experiments on nonsymmetric task settings and MNIST*

In contrast to the symmetric setting in the previous subsections, real datasets such as MNIST exhibit asymmetric statistics so that the different blocks in $m_{\alpha\beta}$ and $C_{(\alpha\beta)(\gamma\delta)}$ assume different values in general. All theoretical results from Sec. III A still hold. However, as the tensor elements of $m_{\alpha\beta}$ and $C_{(\alpha\beta)(\gamma\delta)}$ change, one needs to reconsider the evaluation in Sec. III A 5 in the most general form that yields a more general version of the result.

*Finite MNIST dataset.* First, we consider a setting, where we work with the pure MNIST dataset for two distinct labels 0 and 4. In this setting, we estimate the class-dependent tensor elements $m_{\alpha\beta}$ and $C_{(\alpha\beta)(\gamma\delta)}$ directly from the data. We define the dataset size per class, from which we sample the theory as $D_{\mathrm{base}}$. The training points are also drawn from a subset of these $D_{\mathrm{base}}$ data points. To compare the analytical learning curve for $\langle y_* \rangle$ at $D$ training data points to the empirical results, we need to draw multiple samples of training datasets of size $D < D_{\mathrm{base}}$. As the amount of data in MNIST is limited, these samples will generally not be independent and therefore violate our assumption. Nevertheless, we can see in Fig. 5 that if $D$ is sufficiently small compared to $D_{\mathrm{base}}$, the empirical results and theoretical results match well.

*Gaussianized MNIST dataset.* To test whether deviations in Fig. 5 at large $D$ stem from correlations in the samples of the dataset, we construct a generative scheme for MNIST data. This allows for the generation of infinitely many training points and hence the assumption that the training data are i.i.d. is fulfilled. We construct a pixelwise Gaussian distribution for MNIST images from the samples. We use this model to sample as many MNIST images as necessary for the evaluation of the empirical learning curves. Based on the class-dependent statistics for the pixel means and the pixel covariances in the input data, one can directly compute the elements of the

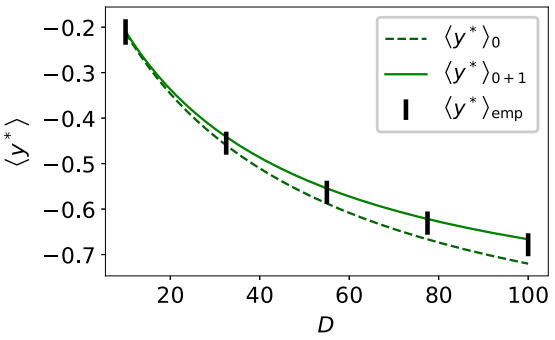



FIG. 5. Predictive mean for a linear network with MNIST data: Comparison of mean-inferred network output for a linear network with one layer for different training set sizes $D$. The figure displays numerical results (bars), zeroth-order prediction (dashed), and first-order corrections (solid). Settings: $N_{\text{dim}} = 784$, $\sigma^2_{\text{reg}} = 2$, and $D_{\text{base}} = 4000$. MNIST classes: $c_1 = 0$, $c_2 = 4$, $y_{c_1} = -1$, and $y_{c_2} = 1$; balanced dataset in $D_{\text{base}}$ and at each $D$. Empirical results display mean and standard deviation over 1000 trials with 1000 test points per trial.

mean $\mu_{\alpha\beta}$ and the covariance $\Sigma_{(\alpha\beta)(\gamma\delta)}$ for the distribution of the input kernel matrix $K^0_{\alpha\beta}$. We see in Fig. 6 that our theory describes the results well for this dataset also for large numbers of training samples.

Furthermore, we can see that in the case of an asymmetric dataset the learning curves depend on the balance ratio of training data $\rho = D_1/D_2$. The bias toward class one in Fig. 6(b) is evident from the curves with $\rho > 0.5$ predicting a lower mean-inferred network output, closer to the target label $y = -1$ of class 1.

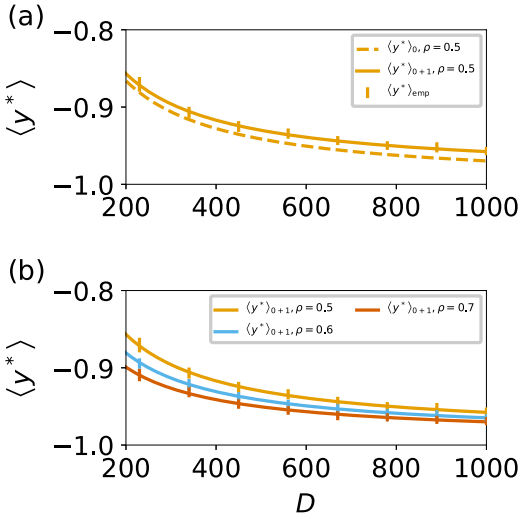



FIG. 6. Predictive mean for an error function (erf) network with Gaussianized MNIST data: (a) Mean-inferred network output for MNIST classification with $\phi(x) = \text{erf}(x)$. Figure shows zeroth-order (dashed line), first-order (solid line), and empirical results (bars). (b) Mean-inferred network output in first-order approximation (solid lines) and empirical results (bars) for MNIST classification with different ratios $\rho = D_1/D_2$ between numbers of training samples per class $D_1$ and $D_2$, respectively; $\rho = 0.5$ (yellow), $\rho = 0.6$ (blue), and $\rho = 0.7$ (red). Empirical results display mean and standard deviation over 50 trials with 1000 test points per trial.

## IV. DISCUSSION

In this work, we investigate the influence of data variability on the performance of Bayesian inference. The probabilistic nature of the data manifests itself in a heterogeneity of the entries in the block-structured kernel matrix of the corresponding Gaussian process. We show that this heterogeneity for a sufficiently large number of $D$ of data samples can be treated as an effective non-Gaussian theory. By employing a time-dependent formulation for the mean of the predictive distribution, this heterogeneity can be treated as a disorder average that circumvents the use of the replica trick. This was the basis of previous investigations of analytical learning curves such as Ref. [54], which also treated the data variability on the level of data samples. In contrast, our approach treats data variability on the kernel level. This, in conjunction with a dynamical approach, allows us to circumvent replica calculations as well as variational approximations. Further, our approach also yields expressions that are valid away from the limit of large datasets and therefore does not rely on the use of the grand canonical partition function. The perturbative treatment of the variability yields first-order corrections to the mean in the variance of the heterogeneity that always push the mean of the predictive distribution toward zero. In particular, we obtain limiting expressions that accurately describe the mean in the limit of infinite training data, qualitatively correcting the zeroth-order approximation corresponding to homogeneous kernel matrices, which is overconfident in predicting the mean to perfectly match the training data in this limit. This finding shows how variability fundamentally limits predictive performance and provides not only a quantitative but also a qualitative difference. Moreover, at a finite number of training data the theory explains the empirically observed performance accurately. We show that our framework captures predictions in linear, nonlinear shallow and deep networks. In nonlinear networks, we show that the optimal value for the variance of the prior weight distribution is achieved in the supercritical regime. The optimal range for this parameter is broad in shallow networks and becomes progressively more narrow in deep networks. These findings support that the optimal initialization is not at the critical point where the variance is unity, as previously thought [7], but that supercritical initialization may have an advantage when considering input variability. An artificial dataset illustrates the origin and the typical statistical structure that arises in heterogeneous kernels, while the application of the formalism to MNIST [21] demonstrates potential use to predict the expected performance in real-world applications. The presented formalism may be extended to also compute the posterior variance. This allows the computation of the expected mean squared error loss on an unseen data point, a frequently used measure of generalization. Further, one can use the results of this manuscript on the mean-inferred network to correct the observational bias in regression tasks (see Appendix 11).

The field theoretical formalism can be combined with approaches that study the effect of fluctuations due to the finite width of the layers [16,28,55,56]. In fact, in the large $N_h$ limit the NNGP kernel is inert to the training data, the so called

lazy regime. At finite network width, the kernel itself receives corrections that are commonly associated with the adaptation of the network to the training data, thus representing what is known as feature learning. The interplay of heterogeneity of the kernel with such finite-size adaptations is a fruitful future direction.

Another approach to study learning in the limit of large width is offered by the neural tangent kernel (NTK) [12], which considers the effect of gradient descent on the network output up to linear order in the change of the weights. A combination of the approach presented here with the NTK instead of the NNGP kernel seems possible and would provide insights into how data heterogeneity affects training dynamics.

The analytical results presented here are based on the assumption that the variability of the data is small and can hence be treated perturbatively. In the regime of large data variability, it is conceivable to employ self-consistent methods instead, which would technically correspond to the computation of saddle points of certain order parameters, which typically leads to an infinite resummation of the perturbative terms that dominate in the large $N_h$ limit. Such approaches may be useful to study and predict the performance of kernel methods for data that show little or no linear separability and are thus dominated by variability. Another direction of extension is the computation of the variance of the Bayesian predictor, which in principle can be treated with the same set of methods as presented here. Finally, since the large width limit as well as finite-size corrections, which in particular yield the kernel response function that we employed here, can be obtained for recurrent and deep networks in the same formalism [28] as well as for residual networks (ResNets) [57], the theory of generalization presented here can straightforwardly be extended to recurrent networks and to ResNets.

## ACKNOWLEDGMENTS

We thank Claudia Merger, Bastian Epping, Kai Segadlo, Alexander van Meegen, and Noah Schürholz for helpful discussions. This work was partly supported by the German Federal Ministry for Education and Research (BMBF Grant No. 01IS19077A to Jülich and BMBF Grant No. 01IS19077B to Aachen) and funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—368482240/GRK2416, the Excellence Initiative of the German federal and state governments (ERS PF-JARA-SDS005), and the Helmholtz Association Initiative and Networking Fund under Project No. SO-092 (Advanced Computing Architectures, ACA). Open access publication funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—491111487. The authors gratefully acknowledge the computing time granted by the JARA Vergabegremium and provided on the JARA Partition part of the supercomputer JURECA at Forschungszentrum Jülich (computation Grant No. JINB33).

## DATA AVAILABILITY

The source code of the manuscript has been uploaded to [60].

## APPENDIX

### 1. The dynamic moment-generating function $Z(L_*, K^y)$

Ultimately, we want to compute the disorder average of

$$y_*(K_{*\cdot}, K_{\cdot\cdot}) = K_{*\alpha} K^{-1}_{\alpha\beta} y_\beta, \tag{A1}$$

$$\langle y_* \rangle_{K_{*\cdot}, K_{\cdot\cdot}} = \left\langle K_{*\alpha} K^{-1}_{\alpha\beta} y_\beta \right\rangle_{K_{*\cdot}, K_{\cdot\cdot}}, \tag{A2}$$

where $K_{\alpha\beta}$ and $K_{*\alpha}$ indicate the kernel matrix of training samples and the kernel matrix between test and training samples, respectively. As the matrices $K_{*\alpha}$ and $K_{\alpha\beta}$ are correlated (due to the presence of the same training data in both matrices) and $K_{\alpha\beta}$ occurs as an inverse, the average is not straightforward. One way to evaluate Eq. (A2) is to compute the average of the moment-generating function $Z(j)$ of the predictive distribution and to compute its derivatives. By Eq. (3), the predictive distribution is given by the ratio of $p(y_*, \mathbf{y}_{\text{tr}} | x_*, \mathbf{x}_{\text{tr}})$ and $p(\mathbf{y}_{\text{tr}} | \mathbf{x}_{\text{tr}})$, which each describe a Gaussian process with a simple dependence on $K$ [28]. Consequently, the moment-generating function of the predictive distribution is also given as a ratio

$$\langle Z(j) \rangle_K = \left\langle \frac{\mathcal{Z}(j, K)}{\mathcal{Z}(0, K)} \right\rangle_K, \tag{A3}$$

which requires an average over the numerator and the denominator, which are also correlated as the normalization $\mathcal{Z}(0, K)$ also depends on the realization of $K$. We hence introduce a dynamic approach, where the normalization $\mathcal{Z}(0, K)$ is fixed by construction to $\mathcal{Z}(0, K) = 1 \; \forall \; K$ [34]. We define the dynamics of an auxiliary variable $q_\alpha(t)$ via

$$\frac{\partial}{\partial t} q_\alpha(t) = -K_{\alpha\beta} q_\beta(t) + y_\alpha. \tag{A4}$$

This inhomogeneous differential equation in the long time limit converges to

$$q_\alpha(t \to \infty) = K^{-1}_{\alpha\beta} y_\beta. \tag{A5}$$

We used the fact that $K_{\alpha\beta}$ is positive semidefinite, as it is a covariance matrix. It is hence safe to assume that a fixed point $q_\alpha(t \to \infty)$ exists. Further, we make the assumption that we start the dynamics at $t \to -\infty$ with an arbitrary initial condition. Hence, we can assume that the fixed point is achieved. A readout of $q_\alpha(t \to \infty)$ using the test-train kernel matrix $K_{*\alpha}$ yields the mean-inferred network output [Eq. (A1)] for a fixed realization of $K_{*\alpha}$ and $K_{\alpha\beta}$. We want to compute the disorder average of $K_{*\alpha} q_\alpha(t \to \infty)$ . Using the Martin-Siggia-Rose-DeDominicis-Jannsen formalism, we construct the dynamic moment-generating function $Z$ for the dynamics given in Eq. (A4):

$$Z(l_*, K) = \int \prod_{\alpha=1}^{D} d\tilde{q}_\alpha \exp\left(S(q, \tilde{q}, K_{*\cdot}, K_{\cdot\cdot}, l_*)\right), \tag{A6}$$

$$\begin{aligned} &S(q, \tilde{q}, K_{*\cdot}, K_{\cdot\cdot}, l_*) \\ &:= \int dt \tilde{q}_\alpha(t) \left[ \frac{\partial}{\partial t} \mathbb{I}_{\alpha\beta} + K_{\alpha\beta} \right] q_\beta(t) - \int dt \tilde{q}_\alpha(t) y_\alpha \\ &\quad + \int dt l_*(t) K_{*\alpha} q_\alpha(t). \end{aligned} \tag{A7}$$

We now go to Fourier space in the variables $q_\alpha(t), \tilde{q}_\alpha(t) \to Q_\alpha(\omega), \tilde{Q}_\alpha(\omega)$ using the definition

$$q_\alpha(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} d\omega \exp(i\omega t) Q_\alpha(\omega), \tag{A8}$$

$$Q_\alpha(\omega) = \int_{-\infty}^{\infty} dt \exp(-i\omega t) q_\alpha(t). \tag{A9}$$

Consequently, products in time domain transform as

$$\int dt q_\alpha(t) p_\beta(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} d\omega Q_\alpha(\omega) P_\beta(-\omega) := Q_\alpha^\top P_\beta, \tag{A10}$$

where we defined the inner product in Fourier domain; we here used that $(2\pi)^{-1}\int dt \exp[it(\omega+\omega')] = \delta(\omega+\omega')$. Likewise, the time derivative $\partial_t$ transforms as

$$\int dt q_\alpha(t) \frac{\partial}{\partial t} q_\beta(t) = \frac{1}{2\pi}\int d\omega\, Q_\alpha(-\omega) i\omega Q_\beta(\omega) = Q_\alpha^\top(i\omega) Q_\beta. \tag{A11}$$

Finally, the product $\int dt \tilde{q}_\alpha(t) y_\alpha$ transforms as

$$\int dt \tilde{q}_\alpha(t) y_\alpha = \int dt \frac{1}{2\pi}\int d\omega \exp(i\omega t)\tilde{Q}_\alpha(\omega) y_\alpha \tag{A12}$$

$$= \int d\omega \delta(\omega)\tilde{Q}_\alpha(\omega) y_\alpha \tag{A13}$$

$$= \tilde{Q}_\alpha(\omega = 0) y_\alpha. \tag{A14}$$

Those definitions transform the action of the dynamic moment-generating functional Eq. (A6) to

$$S(Q, \tilde{Q}, K_{*\cdot}, K_{\cdot\cdot}, L_*) = \frac{1}{2\pi}\int d\omega d\omega' \delta(\omega+\omega')\tilde{Q}_\alpha(\omega)[i\omega' \mathbb{I}_{\alpha\beta} + K_{\alpha\beta}]Q_\beta(\omega') - \tilde{Q}_\alpha(\omega=0)y_\alpha + \frac{1}{2\pi}\int d\omega\, L_*(\omega) K_{*\alpha} Q_\alpha(-\omega). \tag{A15}$$

We now want to perform the disorder average. As in the main text, we assume Gaussian disorder for $K_{\alpha\beta}$ and $K_{*\alpha}$ according to the statistics [Eq. (21)] and compute the disorder-averaged moment-generating function

$$\langle Z(L_*, K)\rangle_K = \int \mathcal{D}\{Q\tilde{Q}\} \langle \exp(S(Q, \tilde{Q}, K_{*\cdot}, K_{\cdot\cdot}, L_*))\rangle_K. \tag{A16}$$

As $K_{*\alpha}$ and $K_{\alpha\beta}$ appear linearly in the action and both are Gaussian variables, the average is straightforward and only affects the term

$$\left\langle \exp\left(\tilde{Q}_\alpha^\top Q_\beta K_{\alpha\beta} + K_{*\alpha} L_*^\top Q_\alpha\right)\right\rangle_K = \exp(\overline{S}), \tag{A17}$$

with

$$\overline{S} = m_{\alpha\beta}\tilde{Q}_\alpha^\top Q_\beta + m_{*\alpha} L_*^\top Q_\alpha \tag{A18}$$

$$+ \tfrac{1}{2} C_{(\alpha\beta)(\gamma\delta)} \tilde{Q}_\alpha^\top Q_\beta \tilde{Q}_\gamma^\top Q_\delta + C_{(*\alpha)(\beta\gamma)} L_*^\top Q_\alpha \tilde{Q}_\beta^\top Q_\gamma + \mathcal{O}(L_*^2). \tag{A19}$$

In the main text and in this Appendix, we only consider perturbative corrections toward the mean-inferred network output. This corresponds to computing the derivative of $\langle Z(L_*, K)\rangle_K$ and subsequently setting $L_* = 0$. Because of this, we only need to consider terms that are linear in $L_*$ and hence drop any $\mathcal{O}(L_*^2)$ in Eq. (A19). Hence, the full disorder-averaged action reads

$$\langle \exp(S(Q, \tilde{Q}, K_{*\cdot}, K_{\cdot\cdot}, L_*))\rangle_K = \exp\left(\tilde{Q}_\alpha^\top [i\omega \mathbb{I}_{\alpha\beta} + m_{\alpha\beta}] Q_\beta - \tilde{Q}_\alpha(\omega=0) y_\alpha + m_{*\alpha} L_*^\top Q_\alpha \right.$$
$$\left. + \tfrac{1}{2} C_{(\alpha\beta)(\gamma\delta)} \tilde{Q}_\alpha^\top Q_\beta \tilde{Q}_\gamma^\top Q_\delta + C_{(*\alpha)(\beta\gamma)} L_*^\top Q_\alpha \tilde{Q}_\beta^\top Q_\gamma \right). \tag{A20}$$

We can decompose this action into a free theory with added perturbative vertices. The free theory is simply a quadratic theory in $Q$ and $\tilde{Q}$

$$\langle \exp(S(Q, \tilde{Q}, K_{*\cdot}, K_{\cdot\cdot}, L_*))\rangle_K^{(0)} = \exp(\tilde{Q}_\alpha^\top [i\omega \mathbb{I}_{\alpha\beta} + m_{\alpha\beta}] Q_\beta - \tilde{Q}_\alpha(\omega=0) y_\alpha + m_{*\alpha} L_*^\top Q_\alpha) \tag{A21}$$

$$= \exp\left( -\frac{1}{2}\begin{pmatrix}\tilde{Q}_\alpha \\ Q_\alpha\end{pmatrix}^\top \begin{pmatrix} 0 & -(i\omega \mathbb{I}_{\alpha\beta} + m_{\alpha\beta}) \\ -(-i\omega \mathbb{I}_{\alpha\beta} + m_{\alpha\beta}) & 0 \end{pmatrix} \begin{pmatrix}\tilde{Q}_\alpha \\ Q_\alpha\end{pmatrix} \right.$$
$$\left. - \tilde{Q}_\alpha(\omega=0) y_\alpha + m_{*\alpha} L_*^\top Q_\alpha \right), \tag{A22}$$

whereas the perturbative part consists of expressions reminiscent of the $\varphi^3 + \varphi^4$ theory

$$\langle \exp(S(Q, \tilde{Q}, K_{*\cdot}, K_{\cdot\cdot}, L_*)) \rangle_K^{(1)} = \exp\left(\tfrac{1}{2}\tilde{Q}_\alpha^\top Q_\beta C_{(\alpha\beta)(\gamma\delta)} \tilde{Q}_\gamma^\top Q_\delta + C_{(*\alpha)(\beta\gamma)} L_*^\top Q_\alpha \tilde{Q}_\beta^\top Q_\gamma + \mathcal{O}(L_*^2)\right). \tag{A23}$$

The fixed point dynamics in Eqs. (A4) and (A5) can also be treated in the Fourier domain. From the inverse Fourier transform, we obtain our limit results via

$$q_\alpha(t) = \frac{1}{2\pi} \int d\omega \exp(i\omega t) Q_\alpha(\omega), \tag{A24}$$

$$\langle y_* \rangle_K = \lim_{t\to\infty} \left\langle \frac{1}{2\pi} \int d\omega \exp(i\omega t) K_{*\alpha} Q_\alpha(\omega) \right\rangle_L, \tag{A25}$$

$$\langle y_* \rangle_K = \lim_{t\to\infty} \frac{1}{2\pi} \int d\omega \exp(i\omega t) 2\pi \frac{\delta}{\delta L_*(-\omega)} [\langle Z(L_*, K) \rangle_K] \Big|_{L_*(\omega)=0}. \tag{A26}$$

### *a. The free case*

In the free case, the action implies that we can contract the monopoles $-\tilde{Q}_\alpha(\omega = 0) y_\alpha$ and $m_{*\alpha} L_*^\top Q_\alpha$ with the propagator

$$\Delta_{\alpha\beta}(\omega, \omega') = \begin{pmatrix} \langle \tilde{Q}_\alpha(\omega) \tilde{Q}_\beta(\omega') \rangle & \langle Q_\alpha(\omega) \tilde{Q}_\beta(\omega') \rangle \\ \langle \tilde{Q}_\alpha(\omega) Q_\beta(\omega') \rangle & \langle Q_\alpha(\omega) Q_\beta(\omega') \rangle \end{pmatrix} \tag{A27}$$

$$= \left[ \frac{1}{2\pi} \delta(\omega + \omega') \begin{pmatrix} 0 & -(i\omega\mathbb{I} + m)_{\alpha\beta} \\ -(-i\omega\mathbb{I} + m)_{\alpha\beta} & 0 \end{pmatrix} \right]^{-1} \tag{A28}$$

$$= 2\pi \delta(\omega + \omega') \begin{pmatrix} 0 & -(-i\omega\mathbb{I} + m)^{-1}_{\alpha\beta} \\ -(i\omega\mathbb{I} + m)^{-1}_{\alpha\beta} & 0 \end{pmatrix}. \tag{A29}$$

The propagator $\Delta_{\alpha\beta}$ can hence contract $\tilde{Q}_\alpha(\omega)$ and $Q_\beta(-\omega)$ with each other. $\delta(\omega + \omega')$ implicitly corresponds to frequency conservation in the propagator. From the equation above, we see that effectively we get the propagator

$$\Delta_{\alpha\beta}(\omega, \omega') = \langle Q_\alpha(\omega) \tilde{Q}_\beta(\omega') \rangle_0 = -2\pi \delta(\omega + \omega')(+i\omega\mathbb{I} + m)^{-1}_{\alpha\beta}. \tag{A30}$$

Hence, we can construct the diagram contracting the monopoles $-\tilde{Q}_\alpha(\omega = 0) y_\alpha$ and $m_{*\alpha} L_*^\top Q_\alpha$ using $\Delta_{\alpha,\beta}$:

$$\frac{\delta}{\delta L_*(-\omega')} [\langle Z(l_*, K) \rangle_{0,K}]|_{L_*(\omega)=0} = \frac{\delta}{\delta L_*(-\omega')} \left[ -m_{*\alpha} \frac{1}{2\pi} \int d\omega L_*(-\omega) \langle Q_\alpha(\omega) \tilde{Q}_\beta(\omega = 0) \rangle_0 y_\beta \right] \Big|_{L_*(\omega)=0} \tag{A31}$$

$$= \frac{\delta}{\delta L_*(-\omega')} \left[ m_{*\alpha} \frac{1}{2\pi} 2\pi \int d\omega L_*(-\omega)(i\omega\mathbb{I} + m)^{-1}_{\alpha\beta} \delta(\omega) y_\beta \right] \Big|_{L_*(\omega)=0} \tag{A32}$$

$$= \delta(\omega') m_{*\alpha} (i\omega'\mathbb{I} + m)^{-1}_{\alpha\beta} y_\beta. \tag{A33}$$

This result yields the zeroth-order approximation for the mean-inferred network output as

$$\langle y_* \rangle_{0,K} = \lim_{t\to\infty} \frac{1}{2\pi} \int d\omega \exp(i\omega t) 2\pi \frac{\delta}{\delta L_*(-\omega)} [\langle Z(L_*, K) \rangle_{0,K}]|_{L_*(\omega)=0} \tag{A34}$$

$$= \lim_{t\to\infty} \int d\omega \exp(i\omega t) \delta(\omega') m_{*\alpha} (+i\omega' \mathbb{I}_{\alpha\beta} + m_{\alpha\beta})^{-1} y_\beta \tag{A35}$$

$$= m_{*\alpha} m^{-1}_{\alpha\beta} y_\beta, \tag{A36}$$

which is the well-known result of Gaussian process inference, as it has to be. Diagrammatically, this corresponds to evaluating the contraction of a monopole with a source term via the propagator leading to the zeroth-order contribution

$\langle y_* \rangle_0 \doteq$  $= m_{*\alpha} m^{-1}_{\alpha\beta} y_\beta$ . (A37)

### *b. Vanishing response loop contributions*

The perturbative expressions, which contribute to the first-order approximation of the mean-inferred network output $\langle y_+ \rangle_{0+1}$, can also be translated into Feynman diagrams. In this subsection, we elaborate on vanishing response loops that limit the set of admissible Feynman diagrams. This is related to the functional formulation of the starting point for our dynamical approach [Eq. (A4)].

In order to go from Eq. (A4) to the functional formulation in Eq. (A6), we follow along the lines of Ref. [38] and start by discretizing the differential equation in the Ito convention. Assuming that $t \in [0, T]$ and discretizing the interval in $N$ steps of width $h = T/N$, we get in the Ito convention

$$q_{\alpha,t+1} - q_{\alpha,t} = -K_{\alpha\beta} q_{\beta,t} h + y_\alpha h, \tag{A38}$$

with

$$h = \frac{T}{N}, \quad t = ih, \quad i \in \{0, 1, \ldots, N\}. \tag{A39}$$

We enforce the relation between $q_{\alpha,t+1}$ and $q_{\alpha,t}$ explicitly using the Fourier transform of the Dirac delta distribution

$$\delta(q) = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} d\tilde{q} \, \exp(\tilde{q} q) \tag{A40}$$

yielding the action

$$Z(\{j_{\alpha,t}, \tilde{j}_{\alpha,t}\}) = \int \mathcal{D}q \mathcal{D}\tilde{q} \exp(S), \tag{A41}$$

$$\int \mathcal{D}q \mathcal{D}\tilde{q} = \prod_{\alpha,t} \int_{-\infty}^{\infty} dq_{\alpha,t} \prod_{\alpha,t} \int_{-i\infty}^{i\infty} \frac{d\tilde{q}_{\alpha,t}}{2\pi i}, \tag{A42}$$

$$S = \sum_{\alpha,t} \tilde{q}_{\alpha,t+1} (q_{\alpha,t+1} - q_{\alpha,t} + K_{\alpha\beta} q_{\beta,t} h) - \sum_{\alpha,t} \tilde{q}_{\alpha,t+1} y_\alpha h + \sum_{\alpha,t} j_{\alpha,t} q_{\alpha,t}, \tag{A43}$$

where we introduced source terms for $q_{\alpha,t}$ as $j_{\alpha,t}$. A natural way to introduce source terms for the auxiliary variables $\tilde{q}_{\alpha,t}$ is to insert a source $\tilde{j}_{\alpha,t}$ on the right-hand side of Eq. (A38) as $q_{\alpha,t+1} - q_{\alpha,t} = -K_{\alpha\beta} q_{\beta,t} h + y_\alpha h + \tilde{j}_{\alpha,t}$. Hence, the action with both sources reads

$$S = \sum_{\alpha,t} \tilde{q}_{\alpha,t+1} (q_{\alpha,t+1} - q_{\alpha,t} + K_{\alpha\beta} q_{\beta,t} h) - \sum_{\alpha,t} \tilde{q}_{\alpha,t+1} y_\alpha h + \sum_{\alpha,t} j_{\alpha,t} q_{\alpha,t} - \sum_{\alpha,t} \tilde{q}_{\alpha,t+1} \tilde{j}_{\alpha,t}. \tag{A44}$$

This shows that one can interpret the correlator between $q_{\alpha,t}$ and $\tilde{q}_{\alpha,s}$ as the linear response of an infinitesimal perturbation at time $s$ on the state at time $t$. Assuming the free theory, we simply replace $K_{\alpha\beta} \to m_{\alpha\beta}$. The correlator reads

$$\left. \frac{\partial Z}{\partial j_{\alpha,t} \partial \tilde{j}_{\beta,s}} \right|_{j=0,\tilde{j}=0} = \langle q_{\alpha,t} \tilde{q}_{\beta,s+1} \rangle_0 |_{j=0,\tilde{j}=0}. \tag{A45}$$

Going back to a continuous time setting, this corresponds to introducing a small regulator $\epsilon \to 0$ into the correlator between the variable $q_\alpha(t)$ and the response field $\tilde{q}_\beta(s)$:

$$\left. \frac{\delta Z}{\delta j_\alpha(t) \delta \tilde{j}_\beta(s)} \right|_{j=0,\tilde{j}=0} = \lim_{\epsilon \searrow 0} \langle q_\alpha(t) \tilde{q}_\alpha(s+\epsilon) \rangle_0. \tag{A46}$$

On a technical level, the presence of this regulating term $\epsilon$ enforces causality in the response function. It further leads to vanishing equal time responses:

$$\left. \frac{\delta Z}{\delta j_\alpha(t) \delta \tilde{j}_\beta(t)} \right|_{j=0,\tilde{j}=0} = \lim_{\epsilon \searrow 0} \langle q_\alpha(t) \tilde{q}_\alpha(t+\epsilon) \rangle_0 = 0. \tag{A47}$$

The reason for this is that one can show that the response propagator is $\langle q_\alpha(t) \tilde{q}_\alpha(s) \rangle_0 \propto H(t-s)$, where $H(t-s)$ is the Heaviside function [38]. This has the direct consequence that integrals of the form

$$\int dt \, \langle q_\alpha(t) \tilde{q}_\beta(t) \rangle_0 = 0 \tag{A48}$$

vanish trivially. Further, because of Eq. (A50), the propagator in the frequency domain now includes an additional factor $\exp(-i\omega\epsilon)$, $\epsilon > 0$. Hence, integrals of the form

$$\frac{1}{2\pi} \int d\omega \left. \frac{\delta Z}{\delta J_\alpha(-\omega) \delta \tilde{J}_\beta(\omega)} \right|_{J=0,\tilde{J}=0} \tag{A49}$$

vanish as well. One can see this by starting from the definition

$$\begin{aligned}\frac{1}{2\pi}\int d\omega \frac{\delta Z}{\delta J_\alpha(-\omega)\delta \tilde{J}_\beta(\omega)}\bigg|_{J=0,\tilde{J}=0} &= \frac{1}{2\pi}\int d\omega \langle Q_\alpha(\omega)\tilde{Q}_\beta(-\omega)\rangle e^{-i\omega\epsilon} = -\frac{1}{2\pi}\int_{-\infty}^{\infty} d\omega\, (i\omega\mathbb{I}+m)^{-1}_{\alpha\beta} e^{-i\omega\epsilon} \\ &= -\frac{1}{2\pi}\int_{-\infty}^{\infty} d\omega \sum_\gamma (i\omega\mathbb{I}+m)^{-1}_{\alpha\gamma}\mathbb{I}_{\gamma\beta} e^{-i\omega\epsilon} \\ &= -\frac{1}{2\pi}\int_{-\infty}^{\infty} d\omega \sum_{\gamma,n} (i\omega\mathbb{I}+m)^{-1}_{\alpha\gamma} v^{(n)}_\gamma v^{(n)}_\beta e^{-i\omega\epsilon} \\ &= -\frac{1}{2\pi}\int_{-\infty}^{\infty} d\omega \sum_{n} (i\omega\mathbb{I}+\lambda^{(n)})^{-1} v^{(n)}_\alpha v^{(n)}_\beta e^{-i\omega\epsilon}, \end{aligned} \tag{A50}$$

where we utilized the fact that the matrix $m$ is symmetric and hence the eigenvectors $v^{(n)}$ corresponding to the eigenvalues $\lambda^{(n)}$ of $m$ form a complete and orthonormal basis set. Further, we know that $m$ is a covariance matrix and hence positive semidefinite and therefore $\lambda^{(n)} \in \mathbb{R}^+$. We treat the terms of $\sum_n$ in Eq. (A50) individually. Integrals of the form

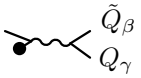
$$\frac{1}{2\pi}\int_{-\infty}^{\infty} d\omega (i\omega\mathbb{I}+\lambda^{(n)})^{-1} e^{-i\omega\epsilon} = 0 \tag{A51}$$

vanish due to the regulator by the residue theorem: The term $(i\omega\mathbb{I}+\lambda^{(n)})^{-1}$ creates a pole in the upper complex half plane of $\omega$, whereas the regulator $\exp(-i\omega\epsilon)$ requires one to close the contour integration in the lower half plane, where no pole is present. Hence, the integral vanishes and response loops vanish in frequency space as well. The consequence for the set of Feynman diagrams is that diagrams, which contract a pair of legs of the same vertex, will vanish; this is sometimes referred to as closed response loops. This greatly reduces the number of allowed Feynman diagrams, as we will see in the next section.

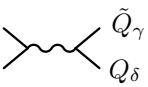
#### *c. The interacting case: first-order corrections for the mean-inferred network output* $\langle y_* \rangle_{0+1}$

Taking into account perturbations, we now need to deal with contractions of the higher-order terms:

$$\text{three}-\text{point Vertex}: C_{(*\alpha)(\beta\gamma)} L_*^\top Q_\alpha \tilde{Q}_\beta^\top Q_\gamma = \tag{A52}$$

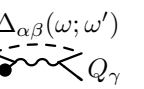



$$\text{four}-\text{point Vertex}: \frac{1}{2}\tilde{Q}_\alpha^\top Q_\beta C_{(\alpha\beta)(\gamma\delta)} \tilde{Q}_\gamma^\top Q_\delta = \tag{A53}$$



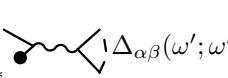

Taking the three-point vertex as an example: The propagator $\Delta_{\alpha\beta}$ can only contract auxiliary fields $\tilde{Q}_\alpha$ and fields $Q_\beta$ with each other. We are not able to contract two auxiliary fields or two real fields with each other. In the case of the three-point vertex, one could hence either contract $Q_\alpha$, $\tilde{Q}_\beta$ or $\tilde{Q}_\beta$, $Q_\gamma$, which corresponds to

$$C_{(*\alpha)(\beta\gamma)} \int d\omega d\omega' L_*(-\omega) Q_\gamma(\omega') \Delta_{\alpha\beta}(\omega;\omega') = \tag{A54}$$




,

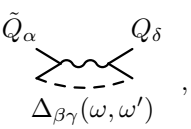
$$C_{(*\alpha)(\beta\gamma)} \int d\omega L_*(-\omega) Q_\alpha(\omega) \int d\omega' \Delta_{\alpha\beta}(\omega',\omega') = \tag{A55}$$




$=0\,,$

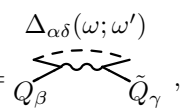
where the second expression vanishes, as we close a response loop. Similarly, the four-point vertex could be within a pair ($\tilde{Q}_\alpha^\top Q_\beta$ or $\tilde{Q}_\gamma^\top Q_\delta$) or between pairs ($\tilde{Q}_\alpha^\top Q_\delta$ or $Q_\beta \tilde{Q}_\gamma^\top$). Again, we see that the first option yields 0, as we close a response loop. Diagrammatically, this reads

$$\frac{1}{2} C_{(\alpha\beta)(\gamma\delta)} \int d\omega d\omega' \Delta_{\beta\gamma}(\omega;\omega') \tilde{Q}_\alpha(-\omega) Q_\delta(\omega') = \tag{A56}$$



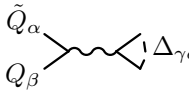

,

$$\frac{1}{2} C_{(\alpha\beta)(\gamma\delta)} \int d\omega d\omega' \Delta_{\beta\gamma}(\omega;\omega') \tilde{Q}_\alpha(-\omega) Q_\delta(\omega') = \tag{A57}$$



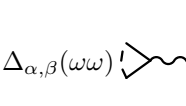

,

$$\frac{1}{2}C_{(\alpha\beta)(\gamma\delta)}\int d\omega\Delta_{\gamma\delta}(\omega,\omega)\int d\omega'\tilde{Q}_\alpha(-\omega')Q_\beta(\omega') =$$




$$\Delta_{\gamma\delta}(\omega,\omega) \quad =0 \quad , \tag{A58}$$

$$\frac{1}{2}C_{(\alpha\beta)(\gamma\delta)}\int d\omega\Delta_{\alpha\beta}(\omega,\omega)\int d\omega'\tilde{Q}_\gamma(-\omega')Q_\delta(\omega') = \Delta_{\alpha,\beta}(\omega\omega)$$




$$= 0 \quad , \tag{A59}$$

#### 2. Diagrams for the mean of inferred network output to first order in $C_{(*\beta)(\gamma\delta)}$, $C_{(\alpha\beta)(\gamma\delta)}$

As we elaborated in Sec. III A 4, we use a field theoretic framework to evaluate the perturbative corrections to the mean-inferred network output, based on the averaged moment-generating function, which reads

$$\begin{aligned}\langle Z(L_*,K)\rangle_K &= \int \mathcal{D}Q\mathcal{D}\tilde{Q}\langle\exp\left(S(Q,\tilde{Q},K_{*\cdot},K_{\cdot\cdot},L_*)\right)\rangle_K := \int \mathcal{D}Q\mathcal{D}\tilde{Q}\exp(\bar{S}(Q,\tilde{Q},L_*)),\\ \bar{S}(Q,\tilde{Q},L_*) &= \tilde{Q}_\alpha^\top[i\omega\mathbb{I}_{\alpha\beta}+m_{\alpha\beta}]Q_\beta - \tilde{Q}_\alpha(\omega=0)y_\alpha + m_{*\alpha}L_*^\top Q_\alpha\\ &\quad + \frac{1}{2}C_{(\alpha\beta)(\gamma\delta)}\tilde{Q}_\alpha^\top Q_\beta\tilde{Q}_\gamma^\top Q_\delta + C_{(*\alpha)(\beta\gamma)}L_*^\top Q_\alpha\tilde{Q}_\beta^\top Q_\gamma,\end{aligned} \tag{A60}$$

where we introduced the notation $1/(2\pi)\int d\omega\,\tilde{Q}(-\omega)Q(\omega) := \tilde{Q}^\top Q$. We use the Einstein convention for the Greek indices $\alpha$, $\beta$, $\gamma$, and $\delta$ that run over the training data exclusively and $*$ denotes the test point that is by definition not part of the training set. From the way that we constructed our theory, we can obtain the disorder-averaged mean-inferred network output by computing derivatives with respect to $L_*$

$$\langle y_*\rangle_K = \lim_{t\to\infty}\frac{1}{2\pi}\int d\omega\exp(i\omega t)2\pi\frac{\delta}{\delta L_*(-\omega)}[\langle Z(L_*,K)\rangle_K]\Big|_{L_*(\omega)=0}. \tag{A61}$$

As we are not able to solve the integral occurring in Eq. (A60) in a closed form, we treat the third- and fourth-order terms in Eq. (A60) perturbatively. We want to study the perturbative terms up to linear order in the expression for $\langle y_*\rangle$. A systematic way to do this is to introduce Feynman diagrams. We there introduce the diagrammatic notation shown in Sec. III A 4.

#### 3. The mean and covariance in the synthetic dataset

As presented in the main text, we need to consider the statistical properties of the overlaps between patterns. In the case of the synthetic task settings, one can evaluate those properties directly. The patterns in the synthetic task have length $N_{\text{dim}}$, where $N_{\text{dim}}$ is even. Each of the $N_{\text{dim}}$ pixels can take the value $\pm1$. The value of each pixel is drawn independently. The probability to be $\pm1$ is given by the parameter $p\in[0,1]$, the class membership of the pattern, and the relative position of the pixel in the pattern according to

$$x_{\alpha i} = \begin{cases}1, & \text{with } p,\\ -1, & \text{with } (1-p),\end{cases} \quad \text{for } i \leqslant \frac{N_{\text{dim}}}{2}, \tag{A62}$$

$$x_{\alpha i} = \begin{cases}1, & \text{with } (1-p),\\ -1, & \text{with } p,\end{cases} \quad \text{for } i > \frac{N_{\text{dim}}}{2}. \tag{A63}$$

For a pattern $x^{(\alpha)}$ in the second class $c=2$, the pixel values are distributed according to

$$x_{\alpha i} = \begin{cases}-1, & \text{with } p,\\ 1, & \text{with } (1-p),\end{cases} \quad \text{for } i \leqslant \frac{N_{\text{dim}}}{2}, \tag{A64}$$

$$x_{\alpha i} = \begin{cases}-1, & \text{with } (1-p),\\ 1, & \text{with } p,\end{cases} \quad \text{for } i > \frac{N_{\text{dim}}}{2}. \tag{A65}$$

We aim to compute the mean $\mu_{\alpha\beta}$ and the covariances $\Sigma_{(\alpha\beta)(\gamma\delta)}$ of the overlaps

$$K_{xx}(\alpha,\beta) = \sum_{i=1}^{N_{\text{dim}}} x_{\alpha i}x_{\beta i}. \tag{A66}$$

Following the calculations in the Supplemental Material [30], this yields

$$\mu_{\alpha\beta} = N_{\text{dim}}\begin{cases}1, & \alpha=\beta,\\ \hat{\mu}, & c_\alpha = c_\beta,\\ -\hat{\mu}, & c_\alpha \neq c_\beta.\end{cases} \tag{A67}$$

$$\Sigma_{(\alpha\beta)(\alpha\delta)} = N_{\text{dim}}\begin{cases}\hat{\mu}(1-\hat{\mu}), & \text{for } \begin{cases}c_\alpha = c_\beta = c_\delta,\\ c_\alpha \neq c_\beta = c_\delta,\end{cases}\\ -\hat{\mu}(1-\hat{\mu}), & \text{for } \begin{cases}c_\alpha = c_\beta \neq c_\delta,\\ c_\alpha = c_\delta, \neq c_\beta,\end{cases}\end{cases} \tag{A68}$$

with $\hat{\mu} := 4p(p-1)+1$.

#### 4. Distribution of overlap-matrix elements for MNIST and FashionMNIST

As stated in the main text, we make the assumption that the elements of the input kernel $K^x_{\alpha\beta} = x_\alpha x_\beta^\top$ are distributed according to multivariate Gaussian distributions. It is, *a priori*, not clear whether this is the case for real datasets such as MNIST, FashionMNIST, or CIFAR-10 (see Fig. 7). Upon investigating the distribution of the pixel values for the input kernel $K^x_{\alpha\beta}$, we can see that a Gaussian approximation provides a good description. One can, however, also see that the distribution of CIFAR-10 values is centered around 0

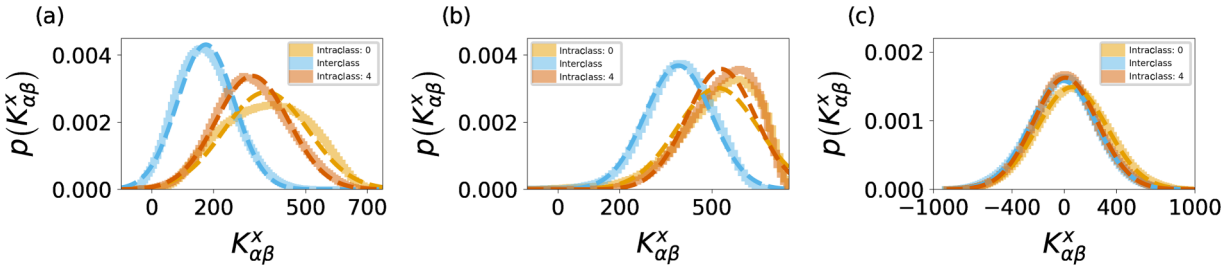



FIG. 7. Distribution of pixel values in input kernels for MNIST, FashionMNIST, and CIFAR-10. Normalized interclass (blue) and intraclass (red, yellow) distributions of the pixel values in the input kernel $K^x_{\alpha\beta}$. Solid lines indicate histograms and dashed lines indicate Gaussian approximations. Results are displayed for (a) MNIST, (b) FashionMNIST, and (c) CIFAR-10. Results are displayed for $N_{\text{dat}} = 2000$ training samples with a balance ratio of 0.5. Class labels: $c_1 = 0$ and $c_2 = 4$.

indicating that the overlap between two patterns $x_{\alpha i}$ and $x_{\beta i}$ is close to orthogonal and hence a simple dot-product kernel might not be informative for this task setting.

### 5. General expressions for the inference formula in asymmetric block-structured task settings

We here treat the most general case for the statistics of overlaps as they occur in real-world datasets, such as MNIST. As in the main text, we assume that the elements of $K^y_{*\alpha}$ and $K^y_{\alpha\beta}$ are distributed according to a multivariate Gaussian distribution

$$K^y_{\alpha\beta} \sim \mathcal{N}(m_{\alpha\beta}, C_{(\alpha\beta)(\gamma\delta)}), \tag{A69}$$

where $\alpha, \beta, \gamma$, and $\delta$ can be either train or test points. The most general choice is to assume that the statistics only depends on the class membership of $a, b, c$, and $d$. We assume a binary classification task with the classes $c_1$ and $c_2$. Hence, the mean of the kernel matrices reads

$$m_{\alpha\beta} = \begin{cases} a, & \alpha = \beta, \\ b, & \alpha \neq \beta, \quad c_\alpha = c_\beta = c_1, \\ d, & \alpha \neq \beta, \quad c_\alpha = c_\beta = c_2, \\ c, & c_\alpha \neq c_\beta, \end{cases} \tag{A70}$$

which is a block matrix. Both the variance $C_{(\alpha\beta)(\alpha\beta)}$ and the covariance $C_{(\alpha\beta)(\gamma\delta)}$ inherit the block structure as well:

$$C_{(\alpha\beta)(\alpha\beta)} = \begin{cases} K_1, & c_\alpha = c_\beta = c_1, \\ \overline{K}_1, & c_\alpha = c_\beta = c_2, \\ K_2 = \overline{K}_2, & c_\alpha \neq c_\beta, \end{cases} \tag{A71}$$

$$C_{(\alpha\beta)(\alpha\delta)} = \begin{cases} v_1, & c_\alpha = c_1, c_\beta = c_1, c_\delta = c_1, \\ v_2, & c_\alpha = c_1, c_\beta = c_2, c_\delta = c_2, \\ v_3, & c_\alpha = c_1, c_\beta = c_1, c_\delta = c_2, \\ v_4 = v_3, & c_\alpha = c_1, c_\beta = c_2, c_\delta = c_1, \\ \overline{v}_1, & c_\alpha = c_2, c_\beta = c_2, c_\delta = c_2, \\ \overline{v}_2, & c_\alpha = c_2, c_\beta = c_1, c_\delta = c_1, \\ \overline{v}_3, & c_\alpha = c_2, c_\beta = c_2, c_\delta = c_1, \\ \overline{v}_4 = \overline{v}_3, & c_\alpha = c_2, c_\beta = c_1, c_\delta = c_2. \end{cases} \tag{A72}$$

Further, the tensor elements $C_{(\alpha\beta)(\gamma\delta)}$ for the case where all indices are different $\alpha \neq \beta \neq \gamma \neq \delta$ yield zero. This follows directly from

$$\begin{aligned} C_{(\alpha\beta)(\gamma\delta)} &= \left\langle \left(K^y_{\alpha\beta} - \left\langle K^y_{\gamma\delta} \right\rangle\right)\left(K^y_{\gamma\delta} - \left\langle K^y_{\gamma\delta} \right\rangle\right) \right\rangle \\ &= \left\langle K^y_{\alpha\beta} K^y_{\gamma\delta} \right\rangle - \left\langle K^y_{\alpha\beta} \right\rangle \left\langle K^y_{\gamma\delta} \right\rangle, \end{aligned} \tag{A73}$$

$$\text{i.i.d samples } \alpha, \beta, \gamma, \delta : = \left\langle K^y_{\alpha\beta} \right\rangle \left\langle K^y_{\gamma\delta} \right\rangle - \left\langle K^y_{\alpha\beta} \right\rangle \left\langle K^y_{\gamma\delta} \right\rangle = 0. \tag{A74}$$

Hence, the tensor $C_{(\alpha\beta)(\gamma\delta)}$ is sparse by construction due to the assumption of independent and identically distributed training data samples. One additional assumption we make in the main text is that the diagonal elements of the kernel matrix $K^y_{\alpha\alpha}$ are deterministic, $K^y_{\alpha\alpha} = \langle K^y_{\alpha\alpha} \rangle = a$. From this follows directly

$$\begin{aligned} C_{(\alpha\alpha)(\beta\gamma)} &= \left\langle \left(K^y_{\alpha\alpha} - \left\langle K^y_{\alpha\alpha} \right\rangle\right)\left(K^y_{\beta\gamma} - \left\langle K^y_{\beta\gamma} \right\rangle\right) \right\rangle \\ &= \left\langle (a - a)\left(K^y_{\beta\gamma} - \left\langle K^y_{\beta\gamma} \right\rangle\right) \right\rangle = 0, \end{aligned} \tag{A75}$$

$$\rightarrow C_{(\alpha\alpha)(\alpha\beta)} = C_{(\alpha\alpha)(\alpha\alpha)} = 0. \tag{A76}$$

As $m^{-1}_{\alpha\beta}$ corresponds to the propagator in the expressions in our main text, we state the matrix elements of the inverse of a bipartite block-structured matrix [Eq. (A70)] in Appendix 5. Based on the action [Eq. (A20)], we now want to compute the corrections at first order to the mean-inferred network output that is given by

$$\begin{aligned} \langle y_* \rangle_{0+1} = {} & g_{*\alpha} y_\alpha + g_{*\alpha} C_{(\alpha\beta)(\gamma\delta)} m^{-1}_{\beta\gamma} m^{-1}_{\delta\delta'} y_{\delta'} \\ & - C_{(*\alpha)(\beta\gamma)} m^{-1}_{\alpha\beta} m^{-1}_{\gamma\gamma'} y_{\gamma'}. \end{aligned} \tag{A77}$$

This can be rewritten as

$$\langle y^* \rangle_{0+1} = g_{*\alpha} y_\alpha + V_4(*) - V_3(*), \tag{A78}$$

$$V_3(*) = \sum_{\alpha,\beta,\gamma} C_{(*\alpha)(\beta\gamma)} m^{-1}_{\alpha\beta} \hat{y}_\gamma, \tag{A79}$$

$$V_4(*) = \sum_{\alpha,\beta,\gamma,\delta} g_{*\alpha} C_{(\alpha\beta)(\gamma\delta)} m^{-1}_{\beta\gamma} \hat{y}_\delta, \tag{A80}$$

$$g_{*,\alpha} = \sum_{\alpha'} m_{*\alpha'} m^{-1}_{\alpha'\alpha}, \tag{A81}$$

$$\hat{y}_\gamma = \sum_{\gamma'} m^{-1}_{\gamma\gamma'} y_{\gamma'}, \tag{A82}$$

where we introduced the shorthand notation $g_{*\alpha}, \hat{y}, V_3(*)$, and $V_4(*)$ to simplify subsequent calculations in this part of the Appendix. Evaluating the expression [Eq. (A77)] is numerically expensive, because it requires

computing contractions over $D$ training examples in up to five indices and hence scales as $\mathcal{O}(D^5)$. However, we know from the definition of our problem setting that the tensors $C_{(\alpha\beta)(\gamma\delta)}$ and $C_{(*\alpha)(\beta\gamma)}$ are sparse and have block structure. We can therefore simplify the problem and compute the contractions analytically. We will do so by computing $V_3(*)$ and $V_4(*)$ individually and combine the results later. It is important to note that even though the approach below is general, it is practically restricted to a binary classification problem. The reason for this is that in our calculation we exploit the block structure in the tensors by splitting the contractions over the indices $\alpha, \beta, \ldots$ into two parts, assuming that the data are presented in an ordered fashion:

$$\sum_{\alpha} = \sum_{c_\alpha = c_1} + \sum_{c_\beta = c_2}. \tag{A83}$$

In principle, an extension to more classes is possible in an analogous way, but it requires the inversion of block matrices with more than two blocks. In the case of $V_3(*)$, the contractions over the three indices $\alpha$, $\beta$, and $\gamma$ hence produce eight terms:

$$\begin{aligned}\left(\sum_{\alpha,c(\alpha)=1} + \sum_{\beta,c(\alpha)=2}\right)\left(\sum_{\beta,c(\beta)=1} + \sum_{\beta,c(\beta)=2}\right)\left(\sum_{\gamma,c(\gamma)=1} + \sum_{\gamma,c(\gamma)=2}\right) \\ = \sum_{c(\alpha)=c(\beta)=c(\gamma)=1} + \sum_{c(\alpha)=c(\beta)=1,c(\gamma)=2} + \cdots \sum_{c(\alpha)=c(\beta)=c(\gamma)=2},\end{aligned} \tag{A84}$$

whereas $V_4(*)$ requires the evaluation of 16 individual terms. As the number of terms grows exponentially with the number of classes, we restrict ourselves to binary classification. In the remainder of this section, we denote $D_1$ as the number of training samples in class 1 and $D_2$ as the number of training samples in class 2.

#### *a. Three-point term $V_3(*)$*

We evaluate $V_3(*)$ by splitting the contractions in eight terms. However, due to the structure of $C_{(*\alpha)(\beta,\gamma)}$ two of those terms vanish. The reason for this is that these terms contain tensor elements $C_{(*\cdot)(\cdot,\cdot)}$ that vanish by construction. Take

$$\sum_{c_\alpha=1,c_\beta=2,c_\gamma=2} C_{(*,\alpha)(\beta,\gamma)} m^{-1}_{\alpha,\beta} \hat{y}_\gamma \tag{A85}$$

for example. We know that $C_{(*\alpha)(\beta\gamma)} = 0$ if neither $\alpha = \beta$ nor $\alpha = \gamma$ and because the test point $* \notin \{\alpha, \beta, \gamma\}$. However, as $c_\alpha = 1$ and both $c_\beta = c_\gamma = 2$ and the set of the two patterns are distinct, there is no index combination of $\alpha\beta\gamma$ that yields $C_{(*\alpha)(\beta\gamma)} \neq 0$. Hence, this term vanishes. Due to symmetry, the equivalent term with $c_\alpha = 2, c_\beta = c_\gamma = 1$ vanishes as well. In this Appendix, we will present the calculation for the first term. The calculations for the remaining seven terms can be found in the Supplemental Material [30] .

*Example calculation for case 1:* $c_\alpha = 1, c_\beta = 1, c_\gamma = 1$. For the first term, we will spell out the calculation steps explicitly. The calculations for the subsequent terms follow along similar lines. We start by introducing the notation

$$C^{c_\alpha,c_\beta,c_\gamma,c_\delta}_{(\alpha\beta)(\gamma\delta)}, \quad \text{with} \quad c_\alpha, c_\beta, c_\gamma, c_\delta \in \{1, 2\}. \tag{A86}$$

From Ref. [30], we know that the expressions $\hat{y}_\gamma$ have block structure as well and $\hat{y}_\gamma = \hat{y}_1 \; \forall \; \gamma$ with $c_\gamma = 1$. To evaluate the first sum

$$\sum_{\alpha,\beta,\gamma}^{c_\alpha=c_\beta=c_\gamma=c_1} C_{(*,\alpha)(\beta,\gamma)} m^{-1}_{\alpha,\beta} \hat{y}_\gamma. \tag{A87}$$

We can start by recognizing that the expression $C_{(*\alpha)(\beta\gamma)}$ is only unequal to zero if either $\alpha = \beta$ or $\alpha = \gamma$. Hence, we can split the sum

$$\begin{aligned}\sum_{\alpha,\beta,\gamma}^{c_\alpha=c_\beta=c_\gamma=c_1} C_{(*,\alpha)(\beta,\gamma)} m^{-1}_{\alpha,\beta} \hat{y}_\gamma = \sum_{\alpha=\beta,\gamma}^{c_\alpha=c_\beta=c_\gamma=c_1} C_{(*,\alpha)(\alpha,\gamma)} m^{-1}_{\alpha,\alpha} \hat{y}_\gamma \\ + \sum_{\alpha=\gamma,\beta}^{c_\alpha=c_\beta=c_\gamma=c_1} C_{(*,\alpha)(\beta,\alpha)} m^{-1}_{\alpha,\beta} \hat{y}_\gamma.\end{aligned} \tag{A88}$$

From this, and exploiting the block structure in $C_{(*\alpha)(\beta\gamma)}$, $\hat{y}_\gamma$, $m^{-1}_{\alpha\beta}$, we get

$$\begin{aligned}&\sum_{\alpha,\beta,\gamma}^{c_\alpha=c_\beta=c_\gamma=c_1} C_{(*,\alpha)(\beta,\gamma)} m^{-1}_{\alpha,\beta} \hat{y}_\gamma \\ &= \sum_{\alpha=\beta,\gamma}^{c_\alpha=c_\beta=c_\gamma=c_1} C_{(*,\alpha)(\alpha,\gamma)} q_1 \hat{y}_\gamma + \sum_{\alpha=\gamma,\beta}^{c_\alpha=c_\beta=c_\gamma=c_1} C_{(*,\alpha)(\beta,\alpha)} q_2 \hat{y}_\gamma \\ &= C^{*,1,1,1}_{(*\alpha)(\alpha,\gamma)} D_1(D_1 - 1)\hat{y}_1(q_1 + q_2).\end{aligned} \tag{A89}$$

As the subsequent calculations are analogous, we will simply state the results for the sake of brevity if no additional considerations need to be taken care of.

*Total expression for $V_3(*)$.* From the symmetries of $C$ and by combining the results from the eight terms, we can write down the full expressions for $V_3(*)$ as

$$\begin{aligned}V_3(*) = &C^{*,1,1,1}_{(*\alpha)(\alpha,\gamma)} N_1(N_1 - 1)\hat{y}_1(q_1 + q_2) \\ &+ C^{*,1,1,2}_{(*\alpha)(\alpha,\gamma)} N_1 N_2 q_1 \hat{y}_2 \\ &+ C^{*,1,2,1}_{(*\alpha)(\beta,\alpha)} N_1 N_2 \hat{y}_1 r \\ &+ C^{*,2,1,2}_{(*\alpha,)(\beta,\alpha)} N_1 N_2 r \hat{y}_2 \\ &+ C^{*,2,2,1}_{(*\alpha)(\alpha,\gamma)} N_1 N_2 q_1' \hat{y}_1 \\ &+ N_2(N_2 - 1)\hat{y}_2 C^{*,2,2,2}_{(*\alpha)(\alpha,\gamma)}(q_1' + q_2'),\end{aligned} \tag{A90}$$

which reduces after some calculations (see the Supplemental Material [30]) to

$$\begin{aligned}V_3(* \in c_1) = &v_1 N_1(N_1 - 1)(q_1\hat{y}_1 + q_2\hat{y}_1) \\ &+ v_3 N_1 N_2(q_1\hat{y}_2 + \hat{y}_1 r) \\ &+ \overline{v}_2 N_1 N_2(r\hat{y}_2 + q_1'\hat{y}_1) \\ &+ \overline{v}_3 N_2(N_2 - 1)(q_1'\hat{y}_2 + q_2'\hat{y}_2),\end{aligned} \tag{A91}$$

$$\begin{aligned}V_3(* \in c_2) = &v_3 N_1(N_1 - 1)(q_1\hat{y}_1 + q_2\hat{y}_1) \\ &+ v_2 N_1 N_2(q_1\hat{y}_2 + \hat{y}_1 r) \\ &+ \overline{v}_3 N_1 N_2(r\hat{y}_2 + q_1'\hat{y}_1) \\ &+ \overline{v}_1 N_2(N_2 - 1)(q_1'\hat{y}_2 + q_2'\hat{y}_2).\end{aligned} \tag{A92}$$

### b. Four-point expression $V_4(*)$

The calculations in this subsection are analogous to the calculations for $V_3(*)$, the difference being that instead of eight terms we now have four contractions over the training data indices $\alpha, \beta, \gamma$, and $\delta$ and hence we need to consider 16 terms. Here, we present the calculation for the first term in detail. The remaining 15 terms are part of the Supplemental Material [30] .

*Example calculation for term 1:* $c_\alpha = 1, c_\beta = 1, c_\gamma = 1, c_\delta = 1$. For the first term, we will spell out the calculation steps explicitly. The calculations for the subsequent terms follow along similar lines. We use the same notation as in the previous subsection for tensor elements:

$$C^{c_\alpha,c_\beta,c_\gamma,c_\delta}_{(\alpha\beta)(\gamma\delta)}, \quad \text{with} \quad c_\alpha, c_\beta, c_\gamma, c_\delta \in \{1,2\}. \tag{A93}$$

For the first term, we get

$$\sum_{c_\alpha=1,c_\beta=1,c_\gamma=1,c_\delta=1} g_{*,\alpha} C_{(\alpha,\beta)(\gamma,\delta)} m^{-1}_{\beta,\gamma} \hat{y}_\delta = g_{*1}\hat{y}_1 D_1(D_1-1) C^{1,1,1,1}_{(\alpha,\beta)(\alpha,\beta)}(q_1+q_2) + g_{*1}\hat{y}_1 D_1(D_1-1)(D_1-2)) C^{1,1,1,1}_{(\alpha,\beta)(\alpha,\delta)}(q_1+3q_2). \tag{A94}$$

The occurring terms have different origins. First, we can state that $g_{*\alpha} = g_{*1}$ and $\hat{y}_\delta = \hat{y}_1$ because of the block structure. Next, we can decompose the sum into relevant expressions. One of them covers the cases where either $\alpha = \gamma, \beta = \delta$ and another $\alpha = \delta, \beta = \gamma$ that produce the following sums:

$$\sum_{\alpha=\gamma,\beta=\delta}^{c_\alpha=c_\beta=c_1} g_{*,\alpha} C_{(\alpha,\beta)(\alpha,\beta)} m^{-1}_{\beta,\alpha} \hat{y}_\beta + \sum_{\alpha=\delta,\beta=\gamma}^{c_\alpha=c_\beta=c_1} g_{*,\alpha} C_{(\alpha,\beta)(\beta,\alpha)} m^{-1}_{\beta,\beta} \hat{y}_\alpha. \tag{A95}$$

In both sums, the tensor element yields $C^{1,1,1,1}_{(\alpha\beta)(\alpha\beta)}$ because of the exchange symmetry in the tensor $C$. Both sums simplify to

$$g_{*1} C^{1,1,1,1}_{(\alpha\beta)(\alpha\beta)} \hat{y}_1 \left( \sum_{\alpha=\gamma,\beta=\delta}^{c_\alpha=c_\beta=c_1} m^{-1}_{\beta,\alpha} + \sum_{\alpha=\delta,\beta=\gamma}^{c_\alpha=c_\beta=c_1} m^{-1}_{\beta,\beta} \right). \tag{A96}$$

Because of the block structure in the propagator $m^{-1}_{\alpha\beta}$, we can also replace $m^{-1}_{\alpha\beta} = q_2$ and $m^{-1}{}_{\beta\beta} = q_1$ according to our definitions in Ref. [30]. The sums evaluate to

$$\sum_{\alpha=\gamma,\beta=\delta}^{c_\alpha=c_\beta=c_1} m^{-1}_{\beta,\alpha} + \sum_{\alpha=\delta,\beta=\gamma}^{c_\alpha=c_\beta=c_1} m^{-1}_{\beta,\beta} = \sum_{\alpha=\gamma,\beta=\delta}^{c_\alpha=c_\beta=c_1} q_2 + \sum_{\alpha=\delta,\beta=\gamma}^{c_\alpha=c_\beta=c_1} q_1 = D_1(D_1-1)(q_2+q_1). \tag{A97}$$

Here, we counted the terms in the sums in the following way: We know that we need to enforce $\alpha \neq \beta$ because otherwise the tensor would yield $C_{(\alpha\alpha)(\alpha\alpha)} = 0$ and hence the terms would vanish. We can choose any of the $D_1$ training examples for the index $\alpha$. Because $\alpha \neq \beta$, we are left with $D_1 - 1$ choices for $\beta$ that produces the corresponding prefactor. In addition to the sums [Eq. (A95)], one also needs to consider the subcases that only two indices in $C_{(\alpha,\beta)(\gamma,\delta)}$ are equal. This produces the sums

$$\sum_{c_\alpha=1,c_\beta=1,c_\gamma=1,c_\delta=1} g_{*,\alpha} C_{(\alpha,\beta)(\gamma,\delta)} m^{-1}_{\beta,\gamma} \hat{y}_\delta = \sum_{c_\alpha=1,c_\beta=1,c_\delta=1} g_{*,\alpha} C_{(\alpha,\beta)(\beta,\delta)} m^{-1}_{\beta,\beta} \hat{y}_\delta + \sum_{c_\alpha=1,c_\beta=1,c_\delta=1} g_{*,\alpha} C_{(\alpha,\beta)(\alpha,\delta)} m^{-1}_{\beta,\alpha} \hat{y}_\delta + \sum_{c_\alpha=1,c_\beta=1,c_\gamma=1} g_{*,\alpha} C_{(\alpha,\beta)(\gamma,\alpha)} m^{-1}_{\beta,\gamma} \hat{y}_\alpha + \sum_{c_\alpha=1,c_\beta=1,c_\gamma=1} g_{*,\alpha} C_{(\alpha,\beta)(\gamma,\beta)} m^{-1}_{\beta,\gamma} \hat{y}_\beta. \tag{A98}$$

As above, we can extract the terms $g_{*1}, C$, and $\hat{y}_1$ due to the block structure in the tensors and we can insert the explicit expressions for the propagator elements $m^{-1}_{\beta\beta}$ and $m^{-1}_{\alpha\beta}$:

$$\sum_{c_\alpha=1,c_\beta=1,c_\gamma=1,c_\delta=1} g_{*,\alpha} C_{(\alpha,\beta)(\gamma,\delta)} m^{-1}_{\beta,\gamma} \hat{y}_\delta = g_{*1}\hat{y}_1 C^{1,1,1,1}_{(\alpha\beta)(\alpha\delta)} \left( \sum_{c_\alpha=1,c_\beta=1,c_\delta=1} q_1 + \sum_{c_\alpha=1,c_\beta=1,c_\delta=1} q_2 + \sum_{c_\alpha=1,c_\beta=1,c_\gamma=1} q_2 + \sum_{c_\alpha=1,c_\beta=1,c_\gamma=1} q_2 \right). \tag{A99}$$

Again, we need to consider the admissible elements for the sums: We assume that the index $\alpha$ can take any of the $D_1$ values. The index $\beta$ can hence take $D_1 - 1$ values as we require $\alpha \neq \beta$. Likewise, $\gamma$ can take $D_1 - 2$ values as we have $\alpha \neq \gamma$ and $\beta \neq \gamma$. If this were not the case, we would over count elements of the sum as cases where $\gamma = \beta$ are already covered in Eq. (A95). Combining all of these results, we get the first term

$$\sum_{c_\alpha=1,c_\beta=1,c_\gamma=1,c_\delta=1} g_{*,\alpha} C_{(\alpha,\beta)(\gamma,\delta)} m^{-1}_{\beta,\gamma} \hat{y}_\delta = g_{*1}\hat{y}_1 D_1(D_1-1) C^{1,1,1,1}_{(\alpha,\beta)(\alpha,\beta)}(q_1+q_2) = +g_{*1}\hat{y}_1 D_1(D_1-1)(D_1-2)) C^{1,1,1,1}_{(\alpha,\beta)(\alpha,\delta)}(q_1+3q_2). \tag{A100}$$

*Total expression for $V_4(*)$.* Considering the symmetries and the definition of $C$ (see the Supplemental Material [30]) and by combining all 16 terms, the total expression for $V_4(*)$ hence reads

$$\begin{aligned}V_4(*) = {} & g_{*1}\hat{y}_1 D_1(D_1-1) * (K_1(q_1+q_2) + v_1(D_1-2)(q_1+3q_2)) \\ & + g_{*2}\hat{y}_2 D_2(D_2-1) * (\overline{K}_1(q_1'+q_2') + \overline{v}_1(D_2-2)(q_1'+3q_2')) \\ & + D_1D_2(D_1-1)(g_{*1}(q_1+q_2)\hat{y}_2 + g_{*1}4r\hat{y}_1 + g_{*2}\hat{y}_1(q_1+q_2))v_3 \\ & + D_2(D_2-1)D_1(g_{*2}(q_2'+q_1')\hat{y}_1 + g_{*2}4r\hat{y}_2 + g_{*1}\hat{y}_2(q_1'+q_2'))\overline{v}_3 \\ & + \overline{v}_2 D_1D_2(D_1-1)(r * g_{*1}\hat{y}_2 + g_{*1}\hat{y}_1 q_1' + g_{*2}\hat{y}_2 q_2 + g_{*2}\hat{y}_1 r) \\ & + v_2 D_1D_2(D_2-1)(g_{*1}\hat{y}_2 r + g_{*1}\hat{y}_1 q_2' + g_{*2}\hat{y}_2 q_1 + g_{*2}\hat{y}_1 r) \\ & + K_2 D_1D_2(g_{*1}\hat{y}_2 r + g_{*2}\hat{y}_1 r + g_{*1}\hat{y}_1 q_1' + g_{*2}\hat{y}_2 q_1). \end{aligned} \tag{A101}$$

#### *c. Full result*

Taking the full results from $V_3(*)$ and $V_4(*)$, we get

$$\begin{aligned}\langle y_* \rangle_{0+1} = {} & g_{*1}D_1 y_1 + g_{*2}D_2 y_2 + K_1(q_1+q_2)g_{*1}\hat{y}_1 D_1(D_1-1) + \overline{K}_1(q_1'+q_2')g_{*2}\hat{y}_2 D_2(D_2-1) \\ & + v_1\hat{y}_1 D_1(D_1-1)[(D_1-2)(q_1+3q_2) * g_{*1} - (q_1+q_2)] \\ & + \overline{v}_1(D_2-2)(q_1'+3q_2') * g_{*2}\hat{y}_2 D_2(D_2-1) \\ & + v_3 D_1D_2[(D_1-1)(g_{*1}(q_1+q_2)\hat{y}_2 + g_{*1}4r\hat{y}_1 + g_{*2}\hat{y}_1(q_1+q_2)) - (q_1\hat{y}_2 + \hat{y}_1 r)] \\ & + \overline{v}_3 D_2(D_2-1)[D_1(g_{*2}(q_2'+q_1')\hat{y}_1 + g_{*2}4r\hat{y}_2 + g_{*1}\hat{y}_2(q_1'+q_2')) - (q_1'\hat{y}_2 + q_2'\hat{y}_2)] \\ & + \overline{v}_2 D_1D_2[(D_1-1)(r * g_{*1}\hat{y}_2 + g_{*1}\hat{y}_1 q_1' + g_{*2}\hat{y}_2 q_2 + g_{*2}\hat{y}_1 r) - (r\hat{y}_2 + q_1'\hat{y}_1)] \\ & + v_2 D_1D_2(D_2-1)(g_{*1}\hat{y}_2 r + g_{*1}\hat{y}_1 q_2' + g_{*2}\hat{y}_2 q_1 + g_{*2}\hat{y}_1 r) \\ & + K_2 D_1D_2(g_{*1}\hat{y}_2 r + g_{*2}\hat{y}_1 r + g_{*1}\hat{y}_1 q_1' + g_{*2}\hat{y}_2 q_1). \end{aligned} \tag{A102}$$

#### *d. Symmetric case*

The expression in Eq. (A102) can be greatly simplified if one considers a symmetric task setting such as in the Ising task of the main text. Following the calculations in the Supplemental Material [30] and exploiting that in the symmetric case, we have

$$K_1, \overline{K}_1, K_2 \to K, \tag{A103}$$

$$v_1, v_2, \overline{v}_1, \overline{v}_2 \to v, \tag{A104}$$

$$v_3, v_4, \overline{v}_3, \overline{v}_4 \to -v; \tag{A105}$$

the expression reduces to the statement in Eqs. (44) and (45).

### 6. Limiting value for $D \to \infty$ of $\langle y_* \rangle_{0+1}$ in the symmetric setting

We can now compute expansions of the mean of the predictive distribution in $1/D \ll 1$ in order to obtain the leading-order expressions for large number of training samples and hence get the asymptotic value for $\lim_{D\to\infty}\langle y_* \rangle_{0+1}$. Starting from the full expression [Eqs. (44) and (45)], we compute the limiting value for $D \to \infty$. Following along the lines of the Supplemental Material [30], we recover in $\mathcal{O}(1)$ the result from the main text:

$$\langle y \rangle_{0+1}(D \to \infty) = y_1 + \frac{y_1}{b}\left(\frac{1}{a-b}(K-4v) - \frac{v}{b}\right). \tag{A106}$$

### 7. The relation between the posterior mean and the generalization error

Generalization in machine learning is typically measured in terms of the loss on a test dataset $\mathcal{D}_{\text{test}} = \{(x_*, \hat{y}_*)\}$, which is distinct from the training dataset $\mathcal{D}_{\text{train}} = \{(x_\alpha, \hat{y}_\alpha)\}_{1\leqslant\alpha\leqslant D}$. In the case of the squared error loss, we evaluate

$$\mathcal{L}_{\text{test}} = (\hat{y}_* - y_*)^2, \tag{A107}$$

where $y_*$ denotes the network output on a test point $x_*$ with the corresponding label $\hat{y}_*$. Generalization is typically defined in terms of the mean $\langle \cdots \rangle_{p(y_*|\mathcal{D}_{\text{train}},\mathcal{D}_{\text{test}})} := \langle \cdots \rangle$ of the generalization error over the posterior distribution of $y_*$ on a fixed set of training and test points. The goal of this Appendix is to show

(1) that the posterior mean computed in the main text enters the bias term of the generalization error,

(2) and that the approach of the main text can be extended to also obtain the remaining terms of the generalization error. To see these two points, we first express Eq. (A107) in terms of the cumulants of $y_*$ by inserting the mean $\langle y_* \rangle_p$:

$$\begin{aligned}\langle \mathcal{L}_{\text{test}} \rangle_p &= \langle (\hat{y}_* - y_*)^2 \rangle_p = \langle ((\hat{y}_* - \langle y_* \rangle_p) - (y_* - \langle y_* \rangle_p))^2 \rangle_p \\ &= \langle (\hat{y}_* - \langle y_* \rangle_p)^2 \rangle_p - 2\langle \hat{y}_* - \langle y_* \rangle_p \rangle \underbrace{\langle y_* - \langle y_* \rangle_p \rangle_p}_{=0} + \langle (y_* - \langle y_* \rangle_p)^2 \rangle_p \\ &= (\hat{y}_* - \langle y_* \rangle_p)^2 + \langle (y_* - \langle y_* \rangle_p)^2 \rangle_p, \end{aligned} \tag{A108}$$

where the first term denotes the bias and the second term denotes the variance contribution, which reflects the common notion of the bias-variance decomposition in machine learning. Contemporary work, such as Cohen *et al.* [13] AND Li *et al.* [58], point out both contributions. If we assume that the network outputs are distributed according to a Gaussian process with a kernel $K$, we can utilize established expressions for the mean and variance and obtain

$$\langle \mathcal{L}_{\text{test}} \rangle_p = \left(\hat{y}_* - K_{*\alpha} K^{-1}_{\alpha\beta} y_\beta\right)^2 + K_{**} - K_{*\alpha} K^{-1}_{\alpha\beta} K_{\beta *}. \tag{A109}$$

As in the main text, we now consider the disorder average over realizations of $K_{*\circ}$ and $K_{\circ\circ}$ of $\langle \mathcal{L}_{\text{test}} \rangle_p$, assuming Gaussian fluctuations of the form

$$K_{ab} = m_{ab} + \eta_{ab}, \tag{A110}$$

$$\langle \eta_{ab} \eta_{cd} \rangle_\eta := C_{(ab)(cd)} \quad a, b, c, d \in [*, 1, \ldots, D]. \tag{A111}$$

Hence, we can write the disorder-averaged mean generalization loss as

$$[\langle \mathcal{L}_{\text{test}} \rangle_p]_K = \left[\left(\hat{y}_* - K_{*\alpha} K^{-1}_{\alpha\beta} y_\beta\right)^2\right]_K + \left[K_{**} - K_{*\alpha} K^{-1}_{\alpha\beta} K_{\beta *}\right]_K. \tag{A112}$$

Starting from the bias contribution, we obtain

$$\begin{aligned}\left[\left(\hat{y}_* - K_{*\alpha} K^{-1}_{\alpha\beta} y_\beta\right)^2\right]_K &= \left[\hat{y}_*^2 - 2\hat{y}_*\left(K_{*\alpha} K^{-1}_{\alpha\beta} y_\beta\right) + \left(K_{*\alpha} K^{-1}_{\alpha\beta} y_\beta K_{*\gamma} K^{-1}_{\gamma\delta} y_\delta\right)\right]_K \\ &= \hat{y}_*^2 - 2\hat{y}_*\left[K_{*\alpha} K^{-1}_{\alpha\beta} y_\beta\right]_K + \left[\left(K_{*\alpha} K^{-1}_{\alpha\beta} y_\beta K_{*\gamma} K^{-1}_{\gamma\delta} y_\delta\right)\right]_K. \end{aligned} \tag{A113}$$

The mean expression $[K_{*\alpha} K^{-1}_{\alpha\beta} y_\beta]_K$ yields the results from Eq. (38), demonstrating point 1 mentioned above.

We now proceed to show point 2, namely, that the remaining terms in Eq. (A112) may be obtained in a similar manner. We start with the remaining term appearing in the bias term [Eq. (A113)]

$$V_{\alpha\beta,\gamma\delta} := \left[\left(K_{*\alpha} K^{-1}_{\alpha\beta} y_\beta K_{*\gamma} K^{-1}_{\gamma\delta} y_\delta\right)\right]_K = \left[(m_{*\alpha} + \eta_{*\alpha})(m+\eta)^{-1}_{\alpha\beta}(m_{*\gamma} + \eta_{*\gamma})(m+\eta)^{-1}_{\gamma\delta} y_\beta y_\delta\right]_K.$$

Now rewriting the two terms $(m+\eta)^{-1} = m^{-1}\,(\mathbb{I} + \eta m^{-1})^{-1}$ and expanding the latter factor in a von Neumann series up to linear order $(\mathbb{I} + \eta m^{-1})^{-1} = \mathbb{I} - \eta m^{-1} + \eta m^{-1} \eta m^{-1} + \mathcal{O}(\eta^3)$, we get

$$\begin{aligned} V_{\alpha\beta,\gamma\delta} &= \left[(m_{*\alpha} + \eta_{*\alpha})(m+\eta)^{-1}_{\alpha\beta}(m_{*\gamma} + \eta_{*\gamma})(m+\eta)^{-1}_{\gamma\delta} y_\beta y_\delta\right]_K \\ &= \left[(m_{*\alpha'} + \eta_{*\alpha'}) m^{-1}_{\alpha'\alpha} (\mathbb{I} - \eta m^{-1} + \eta m^{-1} \eta m^{-1})_{\alpha\beta}\, (m_{*\gamma'} + \eta_{*\gamma'}) m^{-1}_{\gamma'\gamma} (\mathbb{I} - \eta m^{-1} + \eta m^{-1} \eta m^{-1})_{\gamma\delta} y_\beta y_{\delta K} + \mathcal{O}(\eta^3)\right]_K. \end{aligned}$$

We now keep terms up to linear order in $C$ and hence at most quadratic in $\eta$:

$$\begin{aligned} V =& \big[(m_{*\alpha'} m_{*\gamma'} + \eta_{*\alpha'} m_{*\gamma'} + m_{*\alpha'} \eta_{*\gamma'} + \eta_{*\alpha'} \eta_{*\gamma'}) m^{-1}_{\alpha'\alpha} m^{-1}_{\gamma'\gamma} y_\beta y_\delta (\delta_{\alpha\beta}\delta_{\gamma\delta} - [\eta m^{-1}]_{\alpha\beta}\delta_{\gamma\delta} - [\eta m^{-1}]_{\gamma\delta}\delta_{\alpha\beta} \\ &+ [\eta m^{-1}]_{\alpha\beta}[\eta m^{-1}]_{\gamma\delta} + [\eta m^{-1} \eta m^{-1}]_{\alpha\beta}\delta_{\gamma\delta} + [\eta m^{-1} \eta m^{-1}]_{\gamma\delta}\delta_{\alpha\beta})\big]_K + \mathcal{O}(C^2) \\ =& m_{*\alpha'} m_{*\gamma'} m^{-1}_{\alpha'\alpha} m^{-1}_{\gamma'\gamma} y_\beta y_\delta \delta_{\alpha\beta}\delta_{\gamma\delta} + m_{*\alpha'} m_{*\gamma'} m^{-1}_{\alpha'\alpha} m^{-1}_{\gamma'\gamma} y_\beta y_\delta [[\eta m^{-1}]_{\alpha\beta}[\eta m^{-1}]_{\gamma\delta}]_K \\ &+ 2 m_{*\alpha'} m_{*\gamma'} m^{-1}_{\alpha'\alpha} m^{-1}_{\gamma'\gamma} y_\beta y_\delta [[\eta m^{-1} \eta m^{-1}]_{\alpha\beta}]_K \delta_{\gamma\delta} - 2 m^{-1}_{\alpha'\alpha} m^{-1}_{\gamma'\gamma} y_\beta y_\delta [\eta_{*\alpha'} m_{*\gamma'} [\eta m^{-1}]_{\alpha\beta}\delta_{\gamma\delta}]_K \\ &- 2 m^{-1}_{\alpha'\alpha} m^{-1}_{\gamma'\gamma} y_\beta y_\delta [\eta_{*\alpha'} m_{*\gamma'} [\eta m^{-1}]_{\gamma\delta}\delta_{\alpha\beta}]_K + m^{-1}_{\alpha'\alpha} m^{-1}_{\gamma'\gamma} y_\beta y_\delta [\eta_{*\alpha'} \eta_{*\gamma'}]_K \delta_{\alpha\beta}\delta_{\gamma\delta}. \end{aligned} \tag{A114}$$

Utilizing the fact that $\eta$ follows a centered Gaussian distribution, we can evaluate the averages $[\eta_{\alpha\beta}]_K \equiv 0$, $[\eta_{\alpha\rho}\eta_{\gamma\zeta}]_K \equiv C_{(\alpha\rho),(\gamma\zeta)}$, rewriting terms such as $[[\eta m^{-1}]_{\alpha\beta}[\eta m^{-1}]_{\gamma\delta}]_K = m^{-1}_{\rho\beta} m^{-1}_{\zeta\delta} [\eta_{\alpha\rho}\eta_{\gamma\zeta}]_K = m^{-1}_{\rho\beta} m^{-1}_{\zeta\delta}\, C_{(\alpha\rho),(\gamma\zeta)}$, which yields

$$\begin{aligned} V_{\alpha\beta,\gamma\delta} =& \left(m_{*\alpha'} m^{-1}_{\alpha'\alpha} y_\alpha\right)\left(m_{*\gamma'} m^{-1}_{\gamma'\gamma} y_\gamma\right) + m_{*\alpha'} m_{*\gamma'} m^{-1}_{\alpha'\alpha} m^{-1}_{\gamma'\gamma} y_\beta y_\delta m^{-1}_{\rho\beta} m^{-1}_{\zeta\delta} C_{(\alpha\rho)(\gamma\zeta)} \\ &+ 2\left(m_{*\alpha'} m^{-1}_{\alpha'\alpha} m^{-1}_{\rho\zeta} C_{(\alpha\rho)(\zeta\nu)} m^{-1}_{\nu\beta} y_\beta - m^{-1}_{\alpha'\alpha} C_{(*\alpha')(\alpha\rho)} m^{-1}_{\rho\beta} y_\beta\right)\left(m_{*\gamma'} m^{-1}_{\gamma'\gamma} y_\gamma\right) \\ &- 2\left(m^{-1}_{\alpha'\beta} y_\beta\right)\left(m_{*\gamma'} m^{-1}_{\gamma'\gamma}\right)\left(y_\delta m^{-1}_{\rho\delta}\right) C_{(*\alpha')(\gamma\rho)} + \left(m^{-1}_{\alpha'\alpha} y_\alpha\right)\left(m^{-1}_{\gamma'\gamma} y_\gamma\right) C_{(*\alpha')(*\gamma')} + \mathcal{O}(C^2). \end{aligned} \tag{A115}$$

Inserting this into Eq. (A113) yields

$$\left[\left(\hat{y}_* - K_{*\alpha}K^{-1}_{\alpha\beta}y_\beta\right)^2\right]_K = \hat{y}_*^2 - 2\hat{y}_*\left(m_{*\alpha}m^{-1}_{\alpha\beta}y_\beta + m_{*\epsilon}m^{-1}_{\epsilon\alpha}C_{(\alpha\beta)(\gamma\delta)}m^{-1}_{\beta\gamma}\,m^{-1}_{\delta\rho}\,y_\rho - C_{(*\alpha)(\beta\gamma)}m^{-1}_{\alpha\beta}m^{-1}_{\gamma\delta}\,y_\delta\right) + V_{\alpha\beta,\gamma\delta} + \mathcal{O}(C^2). \tag{A116}$$

We now proceed to evaluate the variance term $[K_{**} - K_{*\alpha}K^{-1}_{\alpha\beta}K_{\beta*}]_K$ in Eq. (A112), which we can do in a similar manner. Again, we start by decomposing $K$ into a deterministic $m$ and stochastic component $\eta$ and expand the term $K^{-1} = (m+\eta)^{-1} = m^{-1}\,(\mathbb{I} + \eta\, m^{-1})^{-1} = m^{-1}\,(\mathbb{I} - \eta\, m^{-1} + \eta\, m^{-1}\eta\, m^{-1}) + \mathcal{O}(\eta^3)$ to get

$$\begin{aligned}\left[K_{**} - K_{*\alpha}K^{-1}_{\alpha\beta}K_{\beta*}\right]_K &= \left[m_{**} - (m_{*\alpha}+\eta_{*\alpha})m^{-1}_{\alpha\alpha'}(\mathbb{I} - \eta m^{-1} + \eta m^{-1}\eta m^{-1})_{\alpha\beta}(m_{\beta*}+\eta_{\beta*}) + \mathcal{O}(\eta^3)\right]_K \\ &= \Big[m_{**} - m_{*\alpha}m^{-1}_{\alpha\beta}m_{\beta*} - m_{*\alpha}m^{-1}_{\alpha\alpha'}(-\eta m^{-1})_{\alpha'\beta}\eta_{\beta*} - m_{*\alpha}m^{-1}_{\alpha\alpha'}(\eta m^{-1}\eta m^{-1})_{\alpha'\beta}m_{\beta*} \\ &\quad - \eta_{*\alpha}m^{-1}_{\alpha\alpha'}\delta_{\alpha'\beta}\eta_{\beta*} - \eta_{*\alpha}m^{-1}_{\alpha\alpha'}(-\eta m^{-1})_{\alpha'\beta}m_{\beta*} + \mathcal{O}(\eta^3)\Big]_K.\end{aligned} \tag{A117}$$

Again, rewriting terms such as $(-\eta m^{-1})_{\alpha'\beta}\eta_{\beta*} = \eta_{\alpha'\beta'}\eta_{\beta*}\,m^{-1}_{\beta'\beta}$, and then using $[\eta_{\alpha'\beta'}\eta_{\beta*}]_K = C_{(\alpha'\beta'),(\beta*)}$, we get

$$\begin{aligned}\left[K_{**} - K_{*\alpha}K^{-1}_{\alpha\beta}K_{\beta*}\right]_K &= m_{**} - m_{*\alpha}m^{-1}_{\alpha\beta}m_{\beta*} + 2\,m_{*\alpha}m^{-1}_{\alpha\alpha'}C_{(\alpha'\beta')(\beta*)}m^{-1}_{\beta'\beta} \\ &\quad - m_{*\alpha}m^{-1}_{\alpha\alpha'}C_{(\alpha'\beta')(\gamma'\delta')}m^{-1}_{\beta'\gamma'}m^{-1}_{\delta'\beta}m_{\beta*} - m^{-1}_{\alpha\beta}C_{(*\alpha)(*\beta)} + \mathcal{O}(C^2),\end{aligned} \tag{A118}$$

where we combined two terms $m_{*\alpha}m^{-1}_{\alpha\alpha'}C_{(\alpha'\beta')(\beta*)}m^{-1}_{\beta'\beta}$ and $m^{-1}_{\alpha\alpha'}C_{(*\alpha)(\alpha'\beta')}m^{-1}_{\beta'\beta}m_{\beta*}$ that are identical due to the implicit summation over $\alpha$ and $\beta$ and the symmetry of $C$.

Combining Eqs. (A116) and (A118) of those results together, we obtain the full perturbative expression for the mean of the generalization loss [Eq. (A112)] to first order in $C$:

$$\begin{aligned}[\langle\mathcal{L}_{\text{test}}\rangle_p]_K &= \left[\left(\hat{y}_* - K_{*\alpha}K^{-1}_{\alpha\beta}y_\beta\right)^2\right]_K + \left[K_{**} - K_{*\alpha}K^{-1}_{\alpha\beta}K_{\beta*}\right]_K \\ &= \langle y_*\rangle_0^2 + m_{*\alpha'}m_{*\gamma'}m^{-1}_{\alpha'\alpha}m^{-1}_{\gamma'\gamma}y_\beta y_\delta m^{-1}_{\rho\beta}m^{-1}_{\zeta\delta}C_{(\alpha\rho)(\gamma\zeta)} + 2\langle y_*\rangle_1\langle y_*\rangle_0 - 2\left(m^{-1}_{\alpha'\beta}y_\beta\right)\left(m_{*\gamma'}m^{-1}_{\gamma'\gamma}\right)\left(y_\delta m^{-1}_{\rho\delta}\right)C_{(*\alpha')(\gamma\rho)} \\ &\quad + \left(m^{-1}_{\alpha'\alpha}y_\alpha\right)\left(m^{-1}_{\gamma'\gamma}y_\gamma\right)C_{(*\alpha')(*\gamma')} + \mathcal{O}(C^2) + m_{**} - m_{*\alpha}m^{-1}_{\alpha\beta}m_{\beta*} + 2\,m_{*\alpha}m^{-1}_{\alpha\alpha'}C_{(\alpha'\beta')(\beta*)}m^{-1}_{\beta'\beta} \\ &\quad - m_{*\alpha}m^{-1}_{\alpha\alpha'}C_{(\alpha'\beta')(\gamma'\delta')}m^{-1}_{\beta'\gamma'}m^{-1}_{\delta'\beta}m_{\beta*} - m^{-1}_{\alpha\beta}C_{(*\alpha)(*\beta)} + \mathcal{O}(C^2),\end{aligned} \tag{A119}$$

where we introduced the shorthand $\langle y_*\rangle_0 = m_{*\alpha'}m^{-1}_{\alpha'\alpha}y_\alpha$ and $\langle y_*\rangle_1 = m_{*\alpha'}m^{-1}_{\alpha'\alpha}m^{-1}_{\rho\zeta}C_{(\alpha\rho)(\zeta\nu)}m^{-1}_{\nu\beta}y_\beta - m^{-1}_{\alpha'\alpha}C_{(*\alpha')(\alpha\rho)}m^{-1}_{\rho\beta}y_\beta$. We can reorder the terms in such a way that we can clearly distinguish the perturbative corrections in $C$:

$$\begin{aligned}[\langle\mathcal{L}_{\text{test}}\rangle_p]_K &= \left[\left(\hat{y}_* - K_{*\alpha}K^{-1}_{\alpha\beta}y_\beta\right)^2\right]_K + \left[K_{**} - K_{*\alpha}K^{-1}_{\alpha\beta}K_{\beta*}\right]_K \\ &= \langle y_*\rangle_0^2 + m_{**} - m_{*\alpha}m^{-1}_{\alpha\beta}m_{\beta*} + \left(m_{*\alpha'}m_{*\gamma'}m^{-1}_{\alpha'\alpha}m^{-1}_{\gamma'\gamma}y_\beta y_\delta m^{-1}_{\rho\beta}m^{-1}_{\zeta\delta}\right)C_{(\alpha\rho)(\gamma\zeta)} \\ &\quad - m_{*\alpha}m^{-1}_{\alpha\alpha'}m^{-1}_{\beta'\gamma'}m^{-1}_{\delta'\beta}m_{\beta*}C_{(\alpha'\beta')(\gamma'\delta')} + 2\,m_{*\alpha}m^{-1}_{\alpha\alpha'}m^{-1}_{\beta'\beta}C_{(\alpha'\beta')(\beta*)} \\ &\quad - 2\left(m^{-1}_{\alpha'\beta}y_\beta\right)\left(m_{*\gamma'}m^{-1}_{\gamma'\gamma}\right)\left(y_\delta m^{-1}_{\rho\delta}\right)C_{(*\alpha')(\gamma\rho)} + 2\langle y_*\rangle_1\langle y_*\rangle_0 \\ &\quad + \left(m^{-1}_{\alpha'\alpha}y_\alpha\right)\left(m^{-1}_{\gamma'\gamma}y_\gamma\right)C_{(*\alpha')(*\gamma')} - m^{-1}_{\alpha\beta}C_{(*\alpha)(*\beta)} + \mathcal{O}(C^2).\end{aligned} \tag{A120}$$

We can see that diagrammatically the same structure as in the main for $\langle y_*\rangle_0$ and $\langle y_*\rangle_1$ appear. In addition to this, expressions appear that correspond to additional diagrams from the same diagrammatic elements such as the mean-corrected expressions, like $2\,m_{*\alpha}m^{-1}_{\alpha\alpha'}m^{-1}_{\beta'\beta}C_{(\alpha'\beta')(\beta*)}$. In addition, further diagrammatic building blocks such as $C_{(*\alpha)(*\beta)}$ appear, that are currently not part of our action and would need an extension of our formalism which we leave for future work.

### 8. Experimental details on the numerical simulations

In this Appendix, we outline details to obtain the numerical results in the figures of the main text. As Neal [5] established, the distribution of neural networks in the limit of infinite hidden layer widths is described by a Gaussian process. Using predictions in Bayesian training using Gaussian processes with the kernels $K$, one can describe the generalization properties of the Neural Network Gaussian process with the mean and the variance of the inferred network output on an unseen test point:

$$\langle y_*\rangle = K_{*\alpha}K^{-1}_{\alpha\beta}y_\beta, \tag{A121}$$

$$\langle(y_* - \langle y_*\rangle)^2\rangle = K_{**} - K_{*\alpha}K^{-1}_{\alpha\beta}K_{\beta*}. \tag{A122}$$

To obtain the numerical results for the data-averaged mean-inferred network output, displayed in the figures of the main text, we perform the following steps:

(1) Establish a network architecture by choosing a set of hyperparameters such as network depth $L$, activation function $\phi(x)$, regularization noise $\sigma^2_{\text{reg}}$, and the variance of the hidden weights $\sigma^2_w$.

(2) Draw a balanced set of training data points $\mathcal{D}_{\text{tr}} = \{(x_\alpha, y_\alpha)\}_{1\leqslant\alpha\leqslant D}$ and a set of test data points that are not contained in the training set $\mathcal{D}_{\text{test}} = \{(x_{*i}, y_{*i})\}_{1\leqslant i\leqslant M}$.

(3) Compute the input kernels for the train-train kernel $K^0_{\alpha\beta}$ and the test-train kernel $K^0_{*\alpha}$ according to Eq. (10).

(4) Propagate the input kernels for the chosen network architecture from step 1 using the analytical results [Eqs. (60) and (61)].

(5) Utilize the propagated kernels in the expression for the mean-inferred network output [Eq. (A121)].

(6) Average the resulting mean-inferred network outputs over the training set and test-set realizations.

### 9. Degradation of the mean-inferred network output with network depth

In Fig. 4, we observe that the mean-inferred network output degrades with increasing network depth for all settings of $\sigma_w^2$, while maintaining a local minimum close to the critical value $\sigma_w^2 \approx 1$. One can understand this phenomenon by considering the mean-inferred network output in the case where there is no variability across kernel elements. This is indeed sufficient as we can see from Fig. 4 that there are no qualitative differences to the case with variability. Expressing the mean-inferred network output and assuming a symmetric task setting such as in the figure, we obtain from Eq. (44)

$$\langle y_* \rangle_0 = \frac{D\, b}{(a-b) + D\, b}\, y_1, \tag{A123}$$

where $D$ denotes the amount of training data, $y_1$ denotes the class label for class 1, $a$ denotes the diagonal value, and $b$ denotes the intraclass overlap of the propagated kernel. We note that in the symmetric setting the interclass overlap $c$ is given by $c = -b$ and that the intraclass overlap is the same for both classes $c_1$ and $c_2$. From the expression, we see that the mean-inferred network output will degrade to $\langle y_* \rangle_0 \to 0$, if $b \to 0$. Note that $a$ cannot go to 0 as we introduce a regularizing constant $\sigma^2_{\text{reg}} \neq 0$ [see Eq. (58)]. In Fig. 8, we show that indeed the intraclass overlap $b$ for different values of $\sigma_w^2$ approaches 0 as the depth increases. However, both above and below the critical value $\sigma_w \approx 1$ this approach is much faster compared to the critical value $\sigma_w^2 \approx 1$. This explains the observation in Fig. 4 that the mean-inferred network output degrades for all settings of $\sigma_w^2$, while settings close to the critical value of $\sigma_w^2$ still perform better than settings away from the critical value.

### 10. Ising task: Bayes optimal error

In order to compare the results from the main text to the baseline results of the Ising task setting, we want to compare the performance of the network to the Bayes optimal estimator Fukunaga [59] that is given by

$$\langle y_{*\in c_1} \rangle = \mathbb{E}_{x \in c_1}(f(x)), \tag{A124}$$

$$f(x) = \text{argmax}_y(p(x|c(y))) = \begin{cases} y_1, & p(x|c_1) > p(x|c_2), \\ y_2, & p(x|c_2) > p(x|c_1), \end{cases} \quad \text{with} \quad c(y) = \begin{cases} c_1, & \text{for } y = y_1, \\ c_2, & \text{for } y = y_2. \end{cases} \tag{A125}$$

This is based on the following the intuition: We need to average over all possible pattern realizations that correspond to class 1 ($x \in c_1$), the class that we would like to infer. The Bayes optimal choice of the class, given any possible pattern $x$, is the class where the conditional probability for this particular pattern $p(x|c)$ is the highest. Based on this, we choose the corresponding label to be either $y_{c_1}$, if $p(x|c_1)$ is the highest, or $y_{c_2}$, if $p(x|c_2)$ is the highest.

#### a. Task setting for the Bayes optimal error

To obtain the Bayes optimal error, we investigate a computationally equivalent task: Each pattern is drawn from one of two classes $c_1$ or $c_2$. Each pattern consists of $N$ pixels. If the pattern comes from class $c_1$, each pixel of the pattern is $+1$ with a probability $p$ and $-1$ with a probability $1-p$ with $p > 0.5$; vice versa for class 2. Class $c_1$ has the labels $y_1$ and class $c_2$ comes with the labels $y_2$. Each pixel in each pattern is set i.i.d.

#### b. Computing the Bayes optimal error for the Ising task

To obtain the Bayes optimal error, we parametrize the patterns by the amount of pixels that are $+1$. This number is denoted by $k$. Hence, the conditional probabilities of obtaining $k$ pixels with $+1$ reads

$$p(k|c_1) = \binom{N}{k} p^k (1-p)^{N-k}, \tag{A126}$$

$$p(k|c_2) = \binom{N}{k} (1-p)^k p^{N-k}. \tag{A127}$$

Computing the average in Eq. (A124), we have

$$\mathbb{E}_{x\in c_1}(\cdots) = \sum_{k=0}^{N} p(k|c_1)(\cdots) = \sum_{k=0}^{N} \binom{N}{k} p^k (1-p)^{N-k} (\cdots). \tag{A128}$$

To obtain $\text{argmax}_y(p(k|c(y)))$, we compute the fraction

$$\frac{p(k|c_1)}{p(k|c_2)} = \frac{p^k (1-p)^{N-k}}{p^{N-k}(1-p)^k} = \left(\frac{p}{1-p}\right)^{2k-N}, \tag{A129}$$

which for our case with $p > 0.5$ yields $p/(1-p) > 1$ and hence

$$\text{argmax}_y(p(k|c(y))) = \begin{cases} y_1, & k > N/2, \\ y_2, & k < N/2. \end{cases} \tag{A130}$$

So, it yields class $c_1$ if more than half the patterns are $+1$ and class $c_2$ otherwise. For the Bayes optimal error, this reads

$$\langle y_{*\in c_1} \rangle = \sum_{k=0}^{N} p(k|c_1) \text{argmax}_y(p(k|c(y))) \tag{A131}$$

$$= \sum_{k=0}^{N/2} p(k|c_1) y_2 + \sum_{k=N/2}^{N} p(k|c_1) y_1 \tag{A132}$$

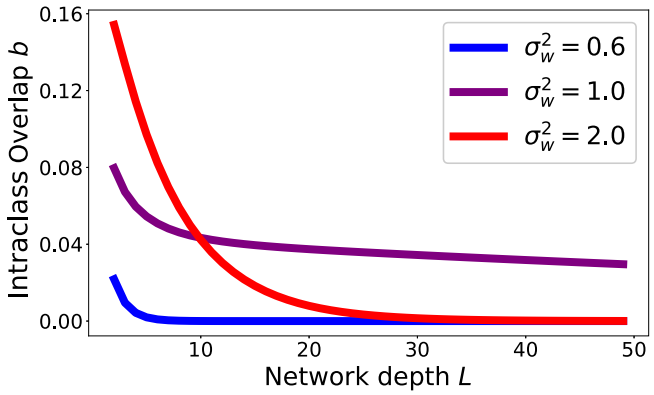



FIG. 8. Intraclass overlap declines with network depths. Mean intraclass overlap $b$ for a nonlinear network with activation function $\phi(x) = \mathrm{erf}(x)$ for three different values of the weight variance $\sigma_w^2 = \{0.6, 1.0, 2.0\}$. In all cases, $b$ converges to zero with increasing network depth. Below the critical value (blue) and above the critical value (red), the convergence is faster than close to the critical value (purple). Results obtained for the Ising task with $p = 0.8$, for a balanced dataset with $D = 4$ data points and a regularization noise of variance $\sigma_{\mathrm{reg}}^2 = 1$.

$$= y_1 \left( \sum_{k=0}^{N/2} \binom{N}{k} p^k (1-p)^{N-k} - \sum_{k=N/2}^{N} \binom{N}{k} p^k (1-p)^{N-k} \right). \tag{A133}$$

Comparing this result to the NNGP results from the main text, we find that even in the case of infinite training data $D \to \infty$, the predictive mean of the network does not coincide with the Bayes optimal predictor, which indicates a performance gap due to the choice of classification via NNGP (see Fig. 9). Further, we find that the zeroth-order results are always overly optimistic, yielding predictions with $\langle y_* \rangle_{D\to\infty} = y_1$ that are better than the Bayes optimal result. This shows that the zeroth-order results produce unreasonable performance beyond the statistical limit. For small values of the task parameter $p$, the fluctuation corrected theory overestimates the corrections, yielding positive values for the predictive mean. This is because only the leading-order fluctuation corrections are incorporated in the theory.

### 11. Bias correction using block formula

Utilizing the expressions for the mean-inferred network output in the zeroth-order approximation for $D$ training data points in the Ising task setting:

$$\langle y_* \rangle_0 = \frac{D\,b}{(a-b) + D\,b}\, y_1, \tag{A134}$$

for the kernel inference

$$\langle y_* \rangle = K_{*\alpha} K^{-1}_{\alpha\beta} y_\beta = \langle y_* \rangle_0 + \text{fluct.} \tag{A135}$$

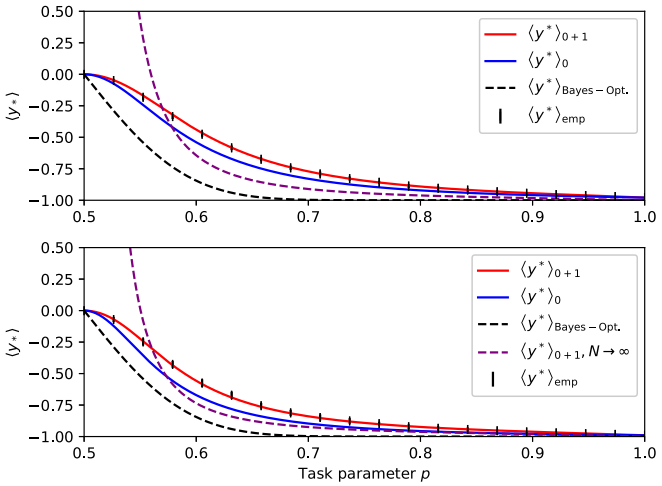



FIG. 9. Bayes optimal error: Comparing the Bayes optimal regression result (black dashed) in linear (top) and erf (bottom) single-layer NNGPs with numerical results (black vertical scatter), mean field theory (blue), and corrected (red) results for different task parameters $p$ with dimension $N = 50$ and amount of training data $D = 100$. The plots also indicate results for the limiting value of infinite training data (purple dashed), $D \to \infty$. Further parameters: $\sigma_v^2 = \sigma_w^2 = 1$ and $\sigma_{\mathrm{reg}}^2 = 1$.

one can try to correct for the bias by estimating the parameters $a$ and $b$ from data. Assuming that we perform inference using the maximum a posteriori (MAP) estimate, we approximate the relation between the true label $y_1$ and the inferred label of $y_*$

$$y_* \approx \langle y_* \rangle = \langle y_* \rangle_0 + \text{fluct.} \tag{A136}$$

$$\to y_1 \approx \frac{(a-b) + Db}{Db} \langle y_* \rangle. \tag{A137}$$

Within this notation, we obtain $\langle y_* \rangle$ from kernel inference. In general, the terms $a$ and $b$ are unknown for a given task, so one needs to sample them from the training dataset.

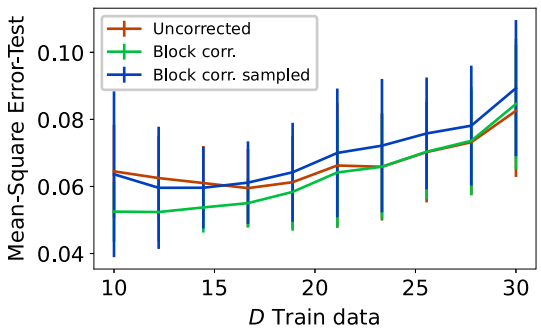



FIG. 10. Bias correction with zeroth-order results: Using the mean-inferred network output $\langle y_* \rangle$ as an MAP estimate, the squared test error MSE $:= \sum_*(y_* - \langle y_* \rangle)^2$ for the corrected expression (green) is consistently lower than for the uncorrected expression (red). Sampling the parameters in the correction from the dataset seemingly removes the benefits from bias correction (blue). Parameters for the experiment: linear network, $p = 0.8, N = 50, \sigma_{\mathrm{reg}}^2 = 10^{-2}, N_{\mathrm{trials}} = 500$, and $N_{\mathrm{test}} = 2000$.

Comparing the results for the corrected MAP estimate

$$\langle y_* \rangle_{\text{corr.}} := \gamma K_{*\alpha} K^{-1}_{\alpha\beta} y_\beta \tag{A138}$$

to the uncorrected estimate, we can observe that indeed the corrected result obtains better performance on the squared error difference (see Fig. 10):

$$\text{MSE} := \sum_* (y_* - \langle y_* \rangle)^2. \tag{A139}$$

However, we also observe that this is the case for small amounts of training data, indicating that this correction does provide a benefit in correcting for regression tasks in the small data regime, whereas corrections by fluctuation results would not yield a benefit due to the increased requirements of training data to estimate the kernel covariance elements.


[1] H. Sompolinsky, A. Crisanti, and H. J. Sommers, Chaos in random neural networks, Phys. Rev. Lett. **61**, 259 (1988).

[2] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Storing infinite numbers of patterns in a spin-glass model of neural networks, Phys. Rev. Lett. **55**, 1530 (1985).

[3] E. Gardner, The space of interactions in neural network models, J. Phys. A: Math. Gen. **21**, 257 (1988).

[4] E. Gardner and B. Derrida, Optimal storage properties of neural network models, J. Phys. A: Math. Gen. **21**, 271 (1988).

[5] R. M. Neal, *Bayesian Learning for Neural Networks* (Springer, New York, 1996).

[6] C. Williams, Computing with infinite networks, in *Advances in Neural Information Processing Systems*, edited by M. Mozer, M. Jordan, and T. Petsche (MIT Press, Cambridge, MA, USA, 1996), Vol. 9.

[7] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli, Exponential expressivity in deep neural networks through transient chaos, in *Advances in Neural Information Processing Systems*, Vol. 29 (Curran Associates, Inc., Barcelona, Spain, 2016).

[8] S. S. Schoenholz, J. Gilmer, S. Ganguli, and J. Sohl-Dickstein, Deep information propagation, in *5th International Conference on Learning Representations* (2017).

[9] G. Yang, Wide feedforward or recurrent neural networks of any architecture are Gaussian processes, in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., Vancouver, British Columbia, Canada, 2019), Vol. 32.

[10] H. Cui, B. Loureiro, F. Krzakala, and L. Zdeborová, Generalization error rates in kernel regression: The crossover from the noiseless to noisy regime*, J. Stat. Mech. (2022) 114004.

[11] B. S. Ruben and C. Pehlevan, *Learning Curves for Heterogeneous Feature-Subsampled Ridge Ensembles* (New Orleans, Louisiana, USA, 2023).

[12] A. Jacot, F. Gabriel, and C. Hongler, *Neural Tangent Kernel: Convergence and Generalization in Neural Networks* (Long Beach, CA, USA, 2018).

[13] O. Cohen, O. Malka, and Z. Ringel, Learning curves for overparametrized deep neural networks: A field theory perspective, Phys. Rev. Res. **3**, 023034 (2021).

[14] G. Naveh and Z. Ringel, A self consistent theory of Gaussian processes captures feature learning effects in finite CNNs, in *Advances in Neural Information Processing Systems*, Vol. 34, edited by A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (Curran Associates, Inc., Virtual, 2021), pp. 21352–21364.

[15] J. A. Zavatone-Veth and C. Pehlevan, Exact marginal prior distributions of finite Bayesian neural networks, in *Advances in Neural Information Processing Systems*, Vol. 34, edited by A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (Curran Associates, Inc., Virtual, 2021), pp. 3364–3375.

[16] J. A. Zavatone-Veth, A. Canatar, B. Ruben, and C. Pehlevan, Asymptotics of representation learning in finite Bayesian neural networks, in *Advances in Neural Information Processing Systems*, Vol. 34, edited by A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (NeurIPS, virtual, 2021).

[17] Q. Li and H. Sompolinsky, Statistical mechanics of deep linear neural networks: The backpropagating kernel renormalization, Phys. Rev. X **11**, 031059 (2021).

[18] D. A. Roberts, S. Yaida, and B. Hanin, *The Principles of Deep Learning Theory* (Cambridge University Press, 2022).

[19] B. Bordelon and C. Pehlevan, Self-consistent dynamical field theory of kernel evolution in wide neural networks, in *Advances in Neural Information Processing Systems*, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Curran Associates, Inc., New Orleans, Louisiana, USA, 2022), pp. 32240–32256.

[20] K. P. Murphy, *Probabilistic Machine Learning: Advanced Topics* (MIT Press, 2023).

[21] Y. LeCun, C. Cortes, and C. J. Burges, The MNIST database of handwritten digits (1998).

[22] D. J. MacKay, *Information Theory, Inference and Learning Algorithms* (Cambridge University Press, 2003).

[23] K. P. Murphy, *Probabilistic Machine Learning: An Introduction* (MIT Press, 2022).

[24] C. K. I. Williams and D. Barber, Bayesian classification with Gaussian processes, IEEE Trans. Pattern Anal. Mach. Intel. **20**, 1342 (1998).

[25] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, *Adaptive Computation and Machine Learning* (MIT Press, Cambridge, MA, 2006), p. 248.

[26] Y. Cho and L. Saul, Kernel methods for deep learning, in *Advances in Neural Information Processing Systems*, edited by Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (Curran Associates, Inc., 2009), Vol. 22.

[27] J. Lee, J. Sohl-Dickstein, J. Pennington, R. Novak, S. Schoenholz, and Y. Bahri, Deep neural networks as Gaussian processes, in *International Conference on Learning Representations* (OpenReview.net, Vancouver, British Columbia, Canada, 2018).

[28] K. Segadlo, B. Epping, A. van Meegen, D. Dahmen, M. Krämer, and M. Helias, Unified field theoretical approach to deep and recurrent neuronal networks, J. Stat. Mech. (2022) 103401.

[29] S. Ariosto, R. Pacelli, M. Pastore, F. Ginelli, M. Gherardi, and P. Rotondo, Statistical mechanics of deep learning beyond the infinite-width limit, Nat. Mach. Intell. **5**, 1497 (2023).

[30] See Supplemental Material at http://link.aps.org/supplemental/10.1103/r1s3-lr3g for detailed versions of the calculations performed in the main text and Appendix.
[31] A. E. Hoerl and R. W. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, Technometrics **42**, 80 (2000).
[32] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, 1st ed. (MIT Press, Cambridge, 2006).
[33] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning (MIT Press, 2016).
[34] C. De Dominicis, Dynamics as a substitute for replicas in systems with quenched random impurities, Phys. Rev. B **18**, 4913 (1978).
[35] P. Martin, E. Siggia, and H. Rose, Statistical dynamics of classical systems, Phys. Rev. A **8**, 423 (1973).
[36] H.-K. Janssen, On a Lagrangean for classical field dynamics and renormalization group calculations of dynamical critical properties, Z. Phys. B **23**, 377 (1976).
[37] J. Stapmanns, T. Kühn, D. Dahmen, T. Luu, C. Honerkamp, and M. Helias, Self-consistent formulations for stochastic nonlinear neuronal dynamics, Phys. Rev. E **101**, 042124 (2020).
[38] M. Helias and D. Dahmen, *Statistical Field Theory for Neural Networks* (Springer International, 2020), p. 203.
[39] K. Fischer and J. Hertz, *Spin Glasses* (Cambridge University Press, 1991).
[40] M. Mezard and A. Montanari, *Information, Physics and Computation* (Oxford University Press, Oxford, UK, 2009).
[41] J. Zinn-Justin, *Quantum Field Theory and Critical Phenomena* (Clarendon Press, Oxford, 1996).
[42] T. Ensslin and M. Frommert, Reconstruction of signals with unknown spectra in information field theory with parameter uncertainty, Phys. Rev. D **83**, 105014 (2011).
[43] E. Dyer and G. Gur-Ari, Asymptotics of wide networks from Feynman diagrams, arXiv:1909.11304
[44] K. Fischer, A. René, C. Keup, M. Layer, D. Dahmen, and M. Helias, Decomposing neural networks as mappings of correlation functions, Phys. Rev. Res. **4**, 043143 (2022).
[45] K. Segadlo, Theory of learning and prediction by deep and recurrent networks in gaussian process approximation, Master's thesis, RWTH Aachen University, 2021.
[46] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, Deep neural networks as Gaussian processes, arXiv:1711.00165.
[47] C. K. Williams, Computation with infinite neural networks, Neural Comput. **10**, 1203 (1998).
[48] R. Price, A useful theorem for nonlinear devices having Gaussian inputs, IRE Trans. Inf. Theory **4**, 69 (1958).
[49] J. Schuecker, S. Goedeke, D. Dahmen, and M. Helias, Functional methods for disordered neural networks, arXiv:1605.06758.
[50] L. Molgedey, J. Schuchhardt, and H. Schuster, Suppressing chaos in neural networks by noise, Phys. Rev. Lett. **69**, 3717 (1992).
[51] L. Xiao, Y. Bahri, J. Sohl-Dickstein, S. S. Schoenholz, and J. Pennington, Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks, arXiv:1806.05393.
[52] I. Seroussi, G. Naveh, and Z. Ringel, Separation of scales and a thermodynamic description of feature learning in some CNNs, Nat. Commun. **14**, 908 (2023).
[53] K. Fischer, J. Lindner, D. Dahmen, Z. Ringel, M. Krämer, and M. Helias, Critical feature learning in deep neural networks, arXiv:2405.10761.
[54] D. Malzahn and M. Opper, A variational approach to learning curves, in *Advances in Neural Information Processing Systems*, edited by T. Dietterich, S. Becker, and Z. Ghahramani (MIT Press, 2001), Vol. 14.
[55] G. Naveh, O. Ben-David, H. Sompolinsky, and Z. Ringel, Predicting the outputs of finite networks trained with noisy gradients, Phys. Rev. E **104**, 064301 (2021).
[56] S. Yaida, Non-Gaussian processes and neural networks at finite widths, in *Proceedings of The First Mathematical and Scientific Machine Learning Conference*, *Proceedings of Machine Learning Research*, edited by J. Lu and R. Ward (PMLR, 2020), Vol. 107, pp. 165–192.
[57] K. Fischer, D. Dahmen, and M. Helias, Field theory for optimal signal propagation in ResNets, arXiv:2305.07715.
[58] Q. Li, B. Sorscher, and H. Sompolinsky, Representations and generalization in artificial and brain neural networks, Proc. Nat. Acad. Sci. USA **121**, e2311805121 (2024).
[59] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. (Academic Press, 2009).
[60] J. Lindner, D. Dahmen, M. Krämer, and M. Helias, Data variability in neural network Bayesian inference, Zenodo, 2025, https://zenodo.org/records/17529999.