A mathematical theory for understanding when abstract representations emerge in neural networks

Bin Wang,* W. Jeffrey Johnston, and Stefano Fusi†
Center for Theoretical Neuroscience, Columbia University, New York, NY
(Dated: October 14, 2025)

Recent experiments reveal that task-relevant variables are often encoded in approximately orthogonal subspaces of the neural activity space. These disentangled low-dimensional representations are observed in multiple brain areas and across different species, and are typically the result of a process of abstraction that supports simple forms of out-of-distribution generalization. The mechanisms by which such geometries emerge remain poorly understood, and the mechanisms that have been investigated are typically unsupervised (e.g., based on variational auto-encoders). Here, we show mathematically that abstract representations of latent variables are guaranteed to appear in the last hidden layer of feedforward nonlinear networks when they are trained on tasks that depend directly on these latent variables. These abstract representations reflect the structure of the desired outputs or the semantics of the input stimuli. To investigate the neural representations that emerge in these networks, we develop an analytical framework that maps the optimization over the network weights into a mean-field problem over the distribution of neural preactivations. Applying this framework to a finite-width ReLU network, we find that its hidden layer exhibits an abstract representation at all global minima of the task objective. We further extend these analyses to two broad families of activation functions and deep feedforward architectures, demonstrating that abstract representations naturally arise in all these scenarios. Together, these results provide an explanation for the widely observed abstract representations in both the brain and artificial neural networks, as well as a mathematically tractable toolkit for understanding the emergence of different kinds of representations in task-optimized, feature-learning network models.

I. INTRODUCTION

How task structure shapes the geometry of neural representations has long been a central question in neuroscience and machine learning. The ability to learn appropriate representations from training data is fundamental to a system's capacity for generalization. In neuroscience, recent experiments have shown that, following task training, neural responses across various brain regions often exhibit a characteristic low-dimensional geometry, referred to as an abstract representation [8, 13, 17, 24, 61, 65, 70, 91] (Fig. 1). In these representations, task-relevant variables are represented in distinct, approximately orthogonal subspaces in the firing rate space of neurons (Fig. 1A, right). Each of these variables is represented in an abstract format because when the activity is projected along its coding direction, it becomes invariant with respect to all the other variables ('dissociated from specific instances'). These representations are based on specialized or linearly mixed selectivity neurons and allow for a simple form of out-of-distribution generalization. Importantly, high-dimensional representations (Fig. 1B, right) do not have this property, although they would allow a linear decoder to report the value of task-relevant variables accurately [29, 44]. For these representations, the variables are nonlinearly mixed with each other [29, 78].

In machine learning, a similar type of representation geometry is called a disentangled representation [6, 16, 93, 95]. The variables represented in an abstract format in neuroscience experiments can be viewed as latent factors underlying the training data, which would be disentangled in the learned representation. Variational autoencoders and their variants have been the standard approach to obtain disentangled representations [14]. However, due to identifiability issues, it has been shown that learning disentangled representations in an entirely unsupervised manner is difficult if not impossible [35, 53, 54]. For this reason, other approaches have been proposed, which often include additional regularization [3, 25, 32, 51] or supervision [2, 40, 80].

Here we focus on the supervised approach proposed in [2, 40], in which a feedforward network is trained to output multiple labels that depend on latent variables. Abstract representations of the latent variables emerge in the last hidden layer of the network after training. We study analytically the emergence of abstract representations in the simplest nonlinear network model that exhibits representation learning: a feedforward network with a single hidden layer and a nonlinear activation function (Fig. 2A, left). For the first time, we prove that the process of minimization of a simple mean square error with l^2-weight regularization in the multi-task setting of [2, 40] is guaranteed to generate abstract representations. We show that this result is robust to the choice of nonlinear activation function. To achieve his goal, we develop an analytical framework to characterize the optimal neural representation in such networks trained on any task. This analytical framework

* bw2841@columbia.edu.
† sf2237@columbia.edu.

is applicable to a broader class of tasks and network architectures, providing a powerful tool for characterizing the structure of neural representation in task-optimized networks [20, 43, 77, 81, 100]. Our work therefore advances existing analytical methods for studying task-optimized neural networks, and more broadly, in those nonlinear models exhibiting permutation symmetry.

The paper is organized as follows. In Section II, we define the task and network model and then introduce the analytical framework for characterizing the optimal neural representations in a two-layer nonlinear network trained on a given task. This framework establishes an exact mapping from the original network model to an effective model whose degrees of freedom are the neural preactivation patterns on training data (Fig. 2A). Solving for the optimal neural representation in the original model is then reduced to analyzing a corresponding mean-field theory in this effective model. In Section III, we apply this framework to finite-width ReLU networks and derive the optimal neural representation for the corresponding task model (Fig. 2B). In Section IV, we further use this analytical framework to show that the abstract representation remains optimal for two broad classes of nonlinear activation functions. Finally, in Section V, we extend the framework to study additional tasks and deep network architectures.

II. THE ANALYTICAL FRAMEWORK

A. Data and network model

We consider feedforward network models trained through supervised learning, with the training dataset

$$\mathcal{D} = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^{P} \subseteq \mathbb{R}^{d_X} \times \mathbb{R}^{d_Y}. \tag{1}$$

Here, P is the number of training samples (input-output mappings) in the task. d_X and d_Y are the input and output dimensions.

Here we assume that the input patterns are basically unstructured, whereas the interesting structure is in the desired outputs, or the labels used to train the network (similarly to [40, 83]). In general, these labels might reflect the structure of a latent space used to generate the complex, seemingly unstructured inputs [40]. However, here we ignore this possibility and we just assume that all the structure that we are going to study is in the output patterns, $\mathbf{y}^i$.

More specifically, we assume that the output associated with a particular input $\mathbf{y}^i \in \{\pm 1\}^{d_Y}$ consists of $d_Y \in \mathbb{N}_+$ binary labels. We denote the input and output matrices as

$$X_{data} = (\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^P) \in \mathbb{R}^{d_X \times P},$$
$$Y = (\mathbf{y}^1, \mathbf{y}^2, ..., \mathbf{y}^P) \in \{\pm 1\}^{d_Y \times P}. \tag{2}$$

Based on their output labels, all the training data form 2^{d_Y} distinct classes. We further assume that each class has the same number of training data, $n \in \mathbb{N}_+$. Namely, all the classes are balanced, and the total number of training data points P satisfies

$$P = n \cdot 2^{d_Y}. \tag{3}$$

Denote the ith row vector of Y as $\mathbf{v}_i \in \{\pm 1\}^P$, whose components are the ith binary labels for all training data, $i = 1, 2, ..d_Y$. Under the above assumption, we find (SI §1.1) that (i) $\mathbf{v}_i \cdot \mathbf{v}_j = P\delta_{ij}$; (ii) $\mathbf{1} \cdot \mathbf{v}_i = 0$, for $\mathbf{1} = (1, 1, ..., 1)^T \in \mathbb{R}^P$, i.e. exactly half of the components of $\mathbf{v}_i$ is $+1$ and the other half is -1, and (iii) $\mathbf{v}_i$'s are all the eigenvectors of the output kernel matrix $K_Y \equiv Y^T Y$ with non-vanishing eigenvalues.

We are interested in the neural representation in a minimal two-layer network model trained to produce these input-output mappings (Fig.2). A two-layer network defines a map

$$f_{W_1, W_2, \mathbf{b}}(\mathbf{x}) = W_2 \phi(W_1 \mathbf{x} + \mathbf{b}),$$

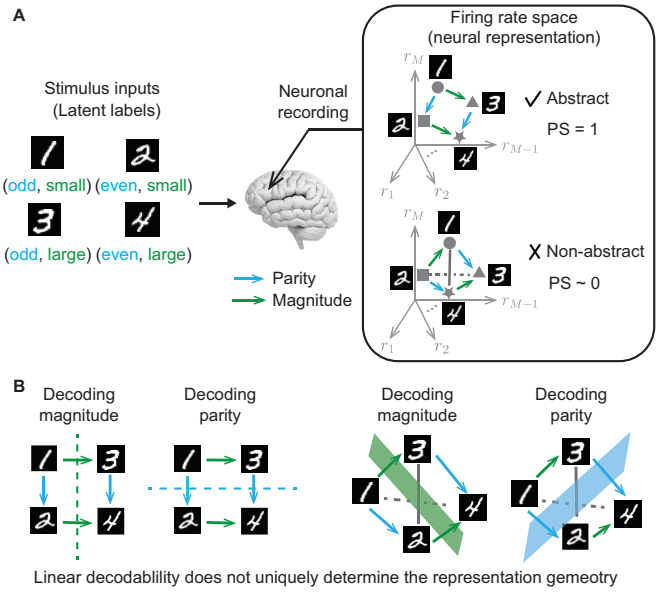

FIG. 1. Abstract representations. (A) Each stimulus input in the task (e.g. images of handwritten digits) is associated with several binary variables (e.g. parity and magnitude of the digits). An abstract representation is one in which each binary variable is represented along a single axis in the population activity space. This is shown in the top plot inside the frame. This geometric property can be quantified using the parallelism score (PS), which measures how parallel the coding directions for one variable remain when the other variables vary. For example, it would measure the parallelism of the parity coding direction for small and large digits (two values of the other variable, magnitude). Abstract representations are low-dimensional and have $PS = 1$. In an alternative neural representation, the points representing the different digits are arranged on a tetrahedron shape, which is the highest dimensional representation (bottom in the frame). This would correspond to a non-abstract representation and has $PS \sim 0$. (B) Magnitude and parity are equally decodable for both geometries.

where ϕ is a component-wise nonlinear activation function. The width of the hidden layer is M. $W_1 \in \mathbb{R}^{M\times d_X}$ and $W_2 \in \mathbb{R}^{d_Y\times M}$ are the weight matrices of the 1st and 2nd layer. $\mathbf{b} \in \mathbb{R}^M$ is the bias parameter. When M is large, the above functional form can approximate any continuous function [18].

The weight and bias parameters in the network are optimized on the training dataset $\mathcal{D}$ via the loss function

$$\begin{aligned} &E(W_1, W_2, \mathbf{b}) \\ &= \sum_{i=1}^{P} \left[\mathbf{y}^i - W_2\phi(W_1\mathbf{x}^i + \mathbf{b})\right]^2 + \lambda_1 \|W_1\|_F^2 \\ &\qquad + \lambda_1 \|\mathbf{b}\|_2^2 + \lambda_2 \|W_2\|_F^2 \\ &\equiv \|Y - W_2\phi(WX)\|_F^2 + \lambda_1 \|W\|_F^2 + \lambda_2 \|W_2\|_F^2 . \end{aligned} \quad (4)$$

Here $\|\cdot\|_F$ is the Frobenius norm of a matrix. In the last line, we introduced the augmented input matrix X and weight matrix W to incorporate the bias parameter $\mathbf{b}$,

$$W \equiv (W_1, \mathbf{b}), \quad X \equiv \begin{pmatrix} X_{data} \\ \mathbf{1}^T \end{pmatrix}. \quad (5)$$

$\lambda_{1,2} > 0$ in Eq. (4) are the strengths of l^2-weight regularization and are typically small.

For a data point $(\mathbf{x}, \mathbf{y})$, we call $\mathbf{r} = \phi(W_1\mathbf{x} + \mathbf{b})$ the population *firing rate vector* or *neural representation* of the data in the hidden layer. We assume that after the task training, the neural network has reached a global minimum of the loss [Eq. (4)]. And we will investigate the neural representations at these global minima. Our main analysis and results will be presented in this setting of two-layer neural networks, and the extensions to deep neural networks are deferred to Section V.

B. Parallelism score measures the abstractness of the neural representation

Previous work uses *parallelism score* (PS) as a measure of the abstractness of a neural representation [2, 8]. We define PS below using our notation. Given a network with the parameters (W_1, W_2, b), its representation of the training data $(\mathbf{x}^i, \mathbf{y}^i)$ is $\mathbf{r}^i = \phi(W_1\mathbf{x}^i + \mathbf{b})$. Since all the training data form 2^{d_Y} distinct classes based on their output labels, we define the prototype representation of each class as the mean firing rate vector of the n data points [Eq.(3)] belonging to this class,

$$\mathbf{r}(\mathbf{y}) = \frac{1}{n} \sum_{i:\mathbf{y}^i=\mathbf{y}} \mathbf{r}^i. \quad (6)$$

where $\mathbf{y} = (y_1, .., y_{d_Y}) \in \{\pm 1\}^{d_Y}$ is the class label.

An abstract representation is characterized by the property that each latent binary label is encoded along a specific direction in the population firing rate space, independently of the other latent labels. To measure this, we examine how the neural representation changes when varying only the kth latent label y_k while keeping other labels $\mathbf{y}_{\setminus k} = \alpha \in \{\pm 1\}^{d_Y-1}$ fixed,

$$\Delta\mathbf{r}(k;\alpha) = \mathbf{r}(y_k = +1, \mathbf{y}_{\setminus k} = \alpha) - \mathbf{r}(y_k = -1, \mathbf{y}_{\setminus k} = \alpha).$$

For different labels $\alpha_1, \alpha_2 \in \{\pm 1\}^{d_Y-1}$, we quantify how consistent the direction of representation changes are, using cosine similarity

$$PS_k(\alpha_1, \alpha_2) \equiv \frac{\Delta\mathbf{r}(k;\alpha_1) \cdot \Delta\mathbf{r}(k;\alpha_2)}{\|\Delta\mathbf{r}(k;\alpha_1)\|_2 \, \|\Delta\mathbf{r}(k;\alpha_2)\|_2} \in [-1, 1].$$

From this definition, if the representation change for the kth latent label is independent of other latent labels $[\Delta\mathbf{r}(k;\alpha_1) = \Delta\mathbf{r}(k;\alpha_2)]$, then $PS_k(\alpha_1, \alpha_2) = 1$. So any deviation from 1 would indicate an interdependence between different latent labels.

The parallelism score for the kth latent label is defined as the average cosine similarity for all distinct pairs (α_1, α_2),

$$PS_k = \frac{1}{2^{d_Y-1} \cdot (2^{d_Y-1} - 1)} \sum_{\substack{\alpha_1,\alpha_2 \in \{\pm 1\}^{d_Y-1} \\ \alpha_1 \neq \alpha_2}} PS_k(\alpha_1, \alpha_2).$$

The overall PS for the neural representation is the average over all latent labels

$$PS = \frac{1}{d_Y} \sum_{k=1}^{d_Y} PS_k. \quad (7)$$

.

By definition, PS of a neural representation only depends on the inner product between the neural representation of all data pairs, $\mathbf{r}^i \cdot \mathbf{r}^j$. So we introduce the representation kernel matrix $K \in \mathbb{R}^{P\times P}$ whose element is

$$K_{ij} \equiv \mathbf{r}^i \cdot \mathbf{r}^j = \phi(W_1\mathbf{x}^i + \mathbf{b}) \cdot \phi(W_1\mathbf{x}^j + \mathbf{b}) \quad (8)$$

This kernel matrix fully determines the PS for a neural representation. Moreover, following the definition, PS does not change when (1) adding a constant to all elements of K, or (2) multiplying K by a positive number.

If the neural representation of data is completely random, $PS \sim 0$. A neural representation is called an abstract representation if the PS is large and close to 1 [8, 17]. We emphasize that the PS of a neural representation is not intrinsically tied to the (linear) decodability of the latent labels (Fig. 1A).

C. The effective mean-field energy

We investigate the global minima of the loss function for the two-layer network models $E(W_1, W_2, \mathbf{b})$ [Eq. (4)]. These minima can also be thought of as the ground states of a physical system with a Hamiltonian

$E(W_1, W_2, \mathbf{b})$. The corresponding Gibbs measure with inverse-temperature parameter β is

$$p(W_1, W_2, \mathbf{b}) = Z_\beta^{-1} \exp\left[-\beta E(W_1, W_2, \mathbf{b})\right], \quad (9)$$

where $Z_\beta \equiv \int_{W_1,W_2,\mathbf{b}} \exp\left[-\beta E(W_1, W_2, \mathbf{b})\right] dW_1 dW_2 d\mathbf{b}$ is the corresponding partition function and the free energy is $F_\beta \equiv \beta^{-1} \ln Z_\beta$. Specifically, the ground state of the system is related to the zero-temperature free energy $\lim_{\beta\to+\infty} F_\beta$.

We introduce some notations to state our main result for the zero-temperature free energy. Denote the augmented input and output kernel matrices as

$$K_X = X^T X \in \mathbb{R}^{P\times P}, \quad K_Y = Y^T Y \in \mathbb{R}^{P\times P}. \quad (10)$$

Since up to a rotation, the input and output data X, Y can be reconstructed from these kernel matrices, we say that these kernel matrices capture the input and output *geometry* (Fig. 2B). Denote RangeK_X as the column space of the matrix K_X

The neural preactivations for all P input data can be summarized in the preactivation matrix (Fig. 3AB)

$$H \equiv X^T W^T = (\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_M) \in \mathbb{R}^{P\times M}, \quad (11)$$

where the kth column vector $\mathbf{h}_k \in \mathbb{R}^P$ represents the preactivation pattern of the kth hidden neuron for all P data points (or stimulus conditions). We note that H is closely related to the response matrix commonly used in neuroscience [44], whose rows correspond to all the stimulus conditions and columns to all the neurons.

With these notations, we find that (SI §1.2) the zero-temperature free energy can be obtained via the following optimization problem over the neural preactivations,

$$\lim_{\beta\to+\infty} F_\beta = \min_{\mathbf{h}_k \in \text{Range}K_X} E(\mathbf{h}_1, \mathbf{h}_2, .., \mathbf{h}_M), \quad (12)$$

where $E(\mathbf{h}_1, \mathbf{h}_2, .., \mathbf{h}_M)$ is

$$E(\mathbf{h}_1, \mathbf{h}_2, .., \mathbf{h}_M) = \lambda_1 \sum_{k=1}^{M} \mathbf{h}_k^T K_X^\dagger \mathbf{h}_k + \text{tr}\left(\frac{\lambda_2}{\lambda_2 + \sum_{k=1}^{M} \phi(\mathbf{h}_k)\phi(\mathbf{h}_k)^T} K_Y\right) \quad (13)$$

Here $K_X^\dagger$ is the Moore-Penrose inverse of K_X. Eq. (12)-(13) shows that the zero-temperature free energy is determined by the global minima of the effective energy function [Eq. (13)] over the neural preactivation patterns in the hidden layer, subject to the constraint $\mathbf{h}_k \in$ RangeK_X.

This maps the original system, whose energy function [Eq. (4)] is over its parameter space $(W_1, W_2, \mathbf{b})$, into an effective system with energy function described by Eq. (13). The effective system consists of M neurons, each of which has a P-dimensional state variable $\mathbf{h} \in \mathbb{R}^P$, lying in the data-dependent subspace RangeK_X (Fig. 3DE). Note that Eq. (12)-(13) are valid for any network width M and input/output kernel matrix K_X/K_Y.

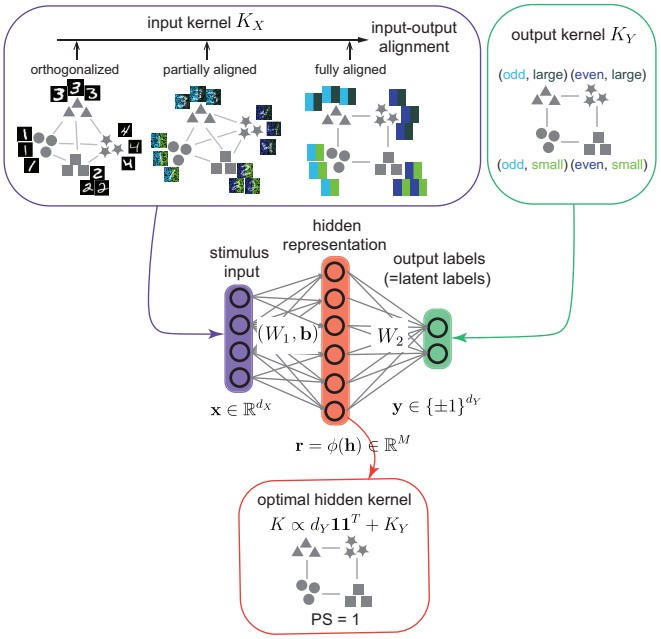

FIG. 2. Model set-up and summary of the main results. (Middle) The two-layer nonlinear network models are trained on tasks related to abstract representation. The weight and bias parameters in each layer are optimized for the task. The hidden layer has width M. For a range of input geometries (characterized by different input kernel matrices K_X) and the specified output geometry (where each output label is exactly the latent label), the optimal hidden representation is always abstract. (Top right) The output labels for each stimulus input are exactly its associated binary latent labels. (Top left) The range of input geometry changes smoothly from a fully orthogonalized input where different stimuli are represented by orthogonal vectors, to an input geometry that is fully aligned with the outputs. Here to illustrate the orthogonalized input in the 4-dimensional space, we draw the 3D projection of it that has a tetrahedron shape.

An immediate consequence of the above equation is that the optimal preactivation matrix $H_* = (\mathbf{h}_{1*}, ..., \mathbf{h}_{M*})$ [Eq. (11)] and the corresponding optimal representation kernel $K_* = \phi(H_*)\phi(H_*)^T$ [Eq. (8)] only depend on the input and output kernel matrices K_X and K_Y, rather than depending directly on the data matrices X, Y. This invariance is a general property for any rotationally-symmetric energy function such as Eq. (4).

Eq.(13) is invariant under permutation over neurons. So we introduce the empirical measure of the preactivations $\rho_M = \sum_{k=1}^{M} \delta_{\mathbf{h}_k}$ (the unnormalized empirical distribution of $\mathbf{h}_k$'s) and rewrite it as

$$\int \lambda_1 \mathbf{h}^T K_X^\dagger \mathbf{h}\, d\rho_M(\mathbf{h}) + \text{tr}\left(\frac{\lambda_2}{\lambda_2 + \int \phi(\mathbf{h})\phi(\mathbf{h})^T d\rho_M(\mathbf{h})} K_Y\right) \quad (14)$$

Note that the representation kernel can be written as $K[\rho_M] = \int \phi(\mathbf{h})\phi(\mathbf{h})^T d\rho_M(\mathbf{h})$. Since all the

preactivation patterns $\mathbf{h}_k$'s lie in RangeK_X, so is the support of the measure, supp$\rho_M \subseteq$ RangeK_X [66].

Eq. (14) shows that the energy function of the effective M-neuron system only depends on the global statistics of neural activity ρ_M. The measure ρ_M can be regarded as the order parameter of the (finite-size) effective M-neuron system, which is permutation-invariant and reflects the symmetry in the original system [Eq. (4)]. This is similar to many other models in statistical physics [1, 42, 46, 47, 63, 69] where the order parameter is a (probability) measure rather than simple scalars.

D. The optimality condition

To compute the PS [Eq. (7)] corresponding to the ground state of effective system, we want to find the representation kernel $K[\rho_M^*]$ corresponding to the global minima ρ_M^* of Eq. (14), with the constraint that ρ_M^* is a sum of M Dirac-delta measures and supp$\rho_M^* \subseteq$ RangeK_X. This optimization is challenging to perform directly, as the space of finite M-sum Dirac delta measures is not convex. To address this, we relax the constraint by enlarging the optimization domain to include all finite positive measures supported on Range$K_X \subseteq \mathbb{R}^P$, which also includes continuous measures. We denote this new space of measures [67], as $M_+(K_X)$. By definition, $M_+(K_X)$ is convex.

Now for $\rho \in M_+(X)$, we consider the energy functional defined by Eq. (14),

$$E[\rho] \equiv \lambda_1 \int \mathbf{h}^T K_X^\dagger \mathbf{h}\, d\rho(\mathbf{h}) + \mathrm{tr}\left[\frac{\lambda_2}{\lambda_2 + \int \phi(\mathbf{h})\phi(\mathbf{h})^T\, d\rho(\mathbf{h})} K_Y\right]. \tag{15}$$

This is a convex functional on $M_+(X)$: since $M_+(X)$ is a convex set (equipped with the usual sum between measures), $E[\rho]$ is a convex function on $M_+(X)$ if and only if $\forall \rho_1, \rho_2 \in M_+(X)$, $f(t) \equiv E[t\rho_1 + (1-t)\rho_2]$ is a convex function for $t \in [0,1]$ [12]. The latter can be checked by computing the 2nd derivative of $f(t)$ (SI §1.3),

$$f''(t) = 2\lambda_2 \mathrm{tr}\left(\frac{1}{\lambda_2 + K_t}\delta K \frac{1}{\lambda_2 + K_t} K_Y \frac{1}{\lambda_2 + K_t}\delta K\right) \geq 0. \tag{16}$$

See SI §1.3 for the expressions of K_t and δK.

For this convex optimization problem, the Karush–Kuhn–Tucker (KKT) condition for the optimal solution ρ_* is (SI §1.4)

$$\lambda_1 \mathbf{h}^T K_X^\dagger \mathbf{h} - \lambda_2 \phi(\mathbf{h})^T \frac{1}{\lambda_2 + K[\rho_*]} K_Y \frac{1}{\lambda_2 + K[\rho_*]} \phi(\mathbf{h}) \geq 0, \forall\, \mathbf{h} \in \mathrm{Range}K_X,$$
$$\text{" = " holds } \Rightarrow\ \forall\, \mathbf{h} \in \mathrm{supp}(\rho_*), \tag{17}$$

where $K[\rho_*] = \int \phi(\mathbf{h})\phi(\mathbf{h})^T\, d\rho_*(\mathbf{h})$ is the representation kernel for ρ_*. Because of convexity, any solution ρ_* satisfying the KKT condition is automatically a global minimum of Eq. (15).

This KKT condition shows that any global minimum ρ_* of Eq. (15) must satisfy the inequality in Eq. (17) and its support is confined to those vectors $\mathbf{h}$ for which the equal sign is attained. The two conditions can be interpreted as a mean-field description [69] of the effective M-neuron system (Fig. 3F). To see this, we define the single-neuron mean-field energy,

$$E(\mathbf{h};\rho) \equiv \lambda_1 \mathbf{h}^T K_X^\dagger \mathbf{h} - \phi(\mathbf{h})^T \frac{\lambda_2}{\lambda_2 + K[\rho]} K_Y \frac{1}{\lambda_2 + K[\rho]} \phi(\mathbf{h}). \tag{18}$$

And the KKT conditions [Eq. (17)] are equivalent to

$$\min_{\mathbf{h} \in \mathrm{Range}K_X} E(\mathbf{h};\rho_*) = 0,$$
$$\mathrm{supp}(\rho_*) \subseteq \operatorname*{argmin}_{\mathbf{h} \in \mathrm{Range}K_X} E(\mathbf{h};\rho_*), \tag{19}$$

for any optimal solution ρ_*. The support of ρ_* contains the optimal preactivation patterns (or single-neuron tuning) for training data, which we denote as $A(K_X, K_Y)$.

From statistical physics perspective, Eq.(19) show that the individual neuron's preactivation $\mathbf{h}$ is trying to minimize the single-neuron mean-field energy $E(\mathbf{h};\rho)$, where the "mean-field" is generated by the statistics of all neurons' activity ρ_* (Fig. 3C). The two terms in $E(\mathbf{h};\rho)$ [Eq.(18)] have interesting interpretations: the 1st term pushes all the neurons' activities to align with the largest principal component of the input kernel, while the 2nd term encourages the transformed neuronal activity $\phi(\mathbf{h})$ to align with the largest principal component of the output-induced mean-field, $\frac{1}{\lambda_2 + K[\rho_*]} K_Y \frac{1}{\lambda_2 + K[\rho_*]}$.

These mean-field equations [Eq. (19)] need to be solved in a self-consistent manner [46, 47, 63, 69] (See SI §1.4). For arbitrary input and output data K_X, K_Y, and nonlinear activation function ϕ, these equations are systems of nonlinear equations that need to be solved numerically. Fortunately, for training data related abstract representation (Section II A), the mean-field equations [Eq. (19)] can be solved exactly as we will present below.

As a final note, although the above optimization is

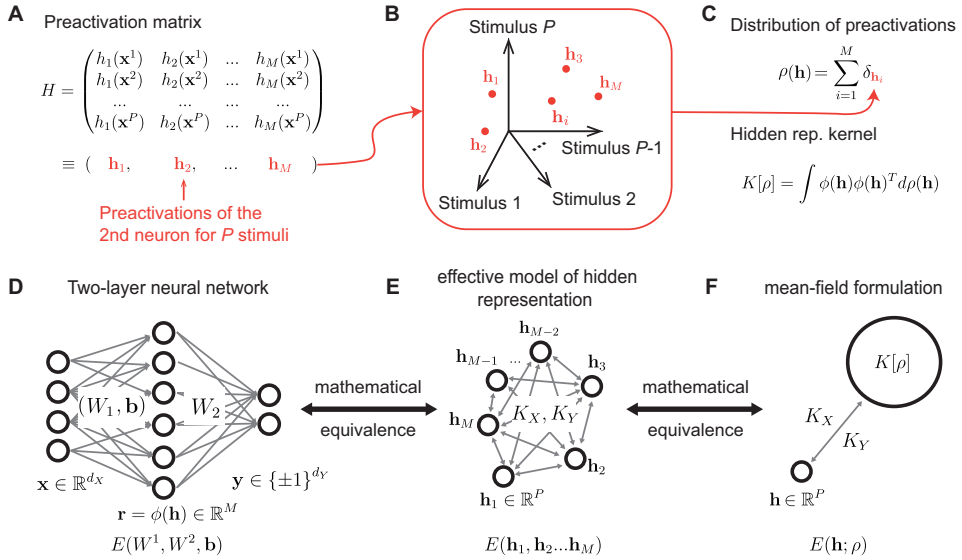

FIG. 3. The analytical framework. (A) The neural preactivation patterns for all P stimuli can be captured in the preactivation matrix H [Eq. (11)]. Each row represents the M hidden neurons' preactivations for a specific stimulus. Each column represents the preactivations of a specific hidden neuron for all P stimuli. (B) The column vectors of H can be plotted in a P-dimensional space that encodes each hidden neuron's tuning for all P stimuli. (C) The statistics of preactivations of hidden neurons can be captured by the empirical (unnormalized) measure. The hidden representation kernel matrix is a linear function of such an empirical measure. (D-F) Mathematically, finding the optimal network reduces to determining the ground state of an effective M-neuron system whose interactions are governed by the input and output kernel matrices [Eq. (13)], which is further equivalent to a mean-field problem where a single representative neuron interacts with the statistics of the neural activity in the network [Eq. (18)].

performed over all positive measures in $M_+(X)$ and in general, some of these measures cannot be attained by a finite-size system, in many cases, the global minimizer ρ_* is a finite sum of Dirac-delta measures. As we show in the next section, this fact allows us to study the optimal neural representations in a *finite-width* network by optimizing Eq. (15) over the infinite-dimensional space $M_+(X)$. And the latter problem is a convex problem [Eq. (14)].

III. WHITENED AND TARGET-ALIGNED INPUTS LEAD TO ABSTRACT REPRESENTATION IN RELU NETWORK

We investigate the optimal neural representations for the data and network model in Section II A. Throughout this section, we assume a ReLU activation function: $\phi(z) = [z]_+$. We focus on two types of inputs: (i) whitened inputs, and (ii) inputs exhibiting stronger alignment with the outputs than the whitened case (hereafter "target-aligned inputs", illustrated in Fig. 4).

For these input geometries, we find that all the solutions of the mean-field problem [Eq. (19)] correspond to abstract hidden representations. We summarize the key steps of the argument here, with detailed derivations provided in SI §2. We start with the scenario where each binary class contains a single data point ($n = 1$) and then extend the results to multi-element classes ($n \geq 2$).

A. Whitened input + single-element class ($n = 1$)

For whitened (or orthogonalized) inputs, $X_{data}^T X_{data} = I_P$. The augmented input kernel matrix and its psudo-inverse are

$$K_X = I_P + \mathbf{1}\mathbf{1}^T, \qquad K_X^\dagger = I_P - \frac{1}{P+1}\mathbf{1}\mathbf{1}^T.$$

Both matrices are positive definite. So the constraint on $\mathbf{h}$ in Eq. (19) is trivial, Range$K_X = \mathbb{R}^P$.

The key result that helps us solve the mean-field equation [Eq.(19)] is the following lower bound for single-neuron mean-field energy (SI §2.1),

$$\begin{aligned} E(\mathbf{h};\rho) &\geq \lambda_1 \mathbf{h}_+^T K_X^\dagger \mathbf{h}_+ - \lambda_2 \mathbf{h}_+^T \frac{1}{\lambda_2 + K[\rho]} K_Y \frac{1}{\lambda_2 + K[\rho]} \mathbf{h}_+ \\ &\equiv E_r(\mathbf{h}_+;\rho), \qquad \forall \mathbf{h} \in \mathbb{R}^P, \forall \rho \in M_+(K_X). \end{aligned} \tag{20}$$

where $\mathbf{h}_+ = [\mathbf{h}]_+$ and $\mathbf{h}_- = [-\mathbf{h}]_+$ are the positive and negative components of the vector $\mathbf{h}$. Using this inequality, we can find the minimum of $E(\mathbf{h};\rho)$ [as required in Eq.(19)] via minimizing $E_r(\mathbf{h}_+;\rho)$ over all the nonnegative vectors $\mathbf{h}_+ \geq 0$.

Minimizing $E_r(\mathbf{h}_+;\rho)$ turns out to be equivalent to determining the copositivity of a matrix [85] and is generally a non-convex problem. Using the properties of the output kernel K_Y, we solve this non-convex problem

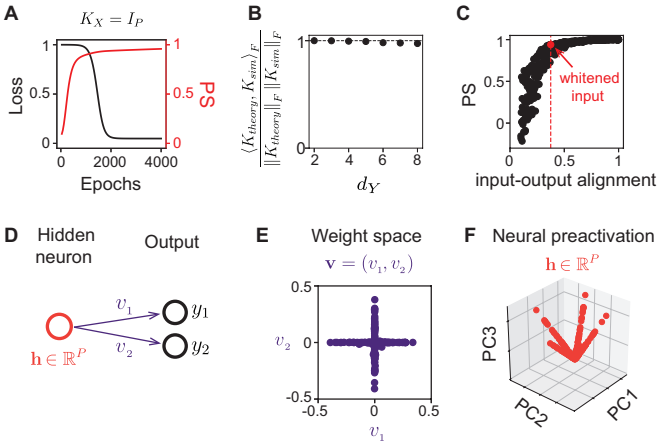

FIG. 4. Task-optimized ReLU network exhibits abstract representation for whitened and target-aligned inputs. (A) The training loss and the parallelism score of the hidden representation are plotted against the number of training epochs for the whitened input. The training is through a gradient descent algorithm. After training, the network performs the task perfectly with zero training loss and has an abstract hidden representation ($PS \to 1$). (B) The optimal hidden kernel predicted by theory (K_{theory} given by Eq. (21)) is aligned with the one found in numerical simulation (K_{sim}) for different output dimensions d_Y. (C) The parallelism score of the hidden representation after training as a function of the input-output alignment. See SI §5 for the definition of input-output alignment. Each point in the plot represents a specific input geometry with a randomly sampled input kernel. The red dot indicates the point for the whitened input kernel. For inputs that are more aligned to the output than the whitened one, the PS is close to 1. (D-E) Modularity of the single-neuron tuning in the hidden layer is captured by the preactivation vector $\mathbf{h}$ and weight vector $\mathbf{v}$ to the output layer for each hidden neuron. Here $d_Y = 2$. (E) Each hidden neuron only has nonzero output weights to a single output unit, suggesting that different neurons in the hidden layer are used to "read out" different output labels. (F) The first three principal components of the neural preactivation space (the same space as Fig. 3B). The preactivation vectors of the hidden neurons concentrate along $P = 4$ distinct directions, as predicted by Eq. (23).

and show that any solution ρ_* must have the hidden representation kernel (SI §2.2)

$$K[\rho_*] = b_*(d_Y \mathbf{1}\mathbf{1}^T + K_Y). \tag{21}$$

where

$$b_* = \sqrt{\frac{\lambda_2}{\lambda_1}\frac{P+1}{P(P+2)}} - \frac{\lambda_2}{P}. \tag{22}$$

Here $\lambda_{1,2}$ are assumed to be small to ensure $b_* > 0$, i.e. $\lambda_1\lambda_2 < \frac{P+2}{P(P+1)}$. Interestingly, the solutions of the mean-field equation ρ_* [Eq.(19)], or equivalently the global minimizers of the loss [Eq.(15)], are not unique. But they all correspond to the identical representation kernel given by Eq.(21).

The support of ρ_* gives the set of optimal preactivation vectors in the hidden layer $A(K_X, K_Y)$, which is found to consist of $2d_Y$ line rays in the nonnegative orthorant $\mathbb{R}^P_{\geq 0}$ (SI §2.2)

$$\mathbf{h} = \mathbf{h}_+ = \alpha(\mathbf{1} + \mathbf{v}_i) \text{ or } \alpha(\mathbf{1} - \mathbf{v}_i), \quad \text{for some } \alpha \geq 0 \text{ and } i \in \{1, 2, .., d_Y\}. \tag{23}$$

As half of the components in $\mathbf{v}_i$ are $+1$ and the other ones are -1, these optimal preactivation vectors have exactly half of the components to be 2 and the other ones are zero.

Finally, this optimal solution ρ_* can be attained in a network with $M \geq 2d_Y$ hidden neurons where each neuron simply takes the preactivation $\mathbf{h}_i^{\pm} = \sqrt{b_*}(\mathbf{1} \pm \mathbf{v}_i)$ (SI §2.2).

Altogether, when $M \geq 2d_Y$ and the input is whitened, we find the optimal hidden representation as in Eq.(21) and that the neurons in the hidden layer clusters into $2d_Y$ groups given by Eq.(23).

B. Whitened input + multi-element class ($n \geq 2$)

When each class contains multiple data points ($n \geq 2$ in Eq. (3)), whitened input has input kernel matrix $X_{data}^T X_{data} = I_P$. Namely, both the within-class and between-class correlations are zero. Here we consider a slightly more general form of input kernel that would allow within-class correlation,

$$X_{data}^T X_{data} = \underbrace{\begin{pmatrix} \underbrace{C_1}_{n\times n} & & & \\ & C_2 & & \\ & & .. & \\ & & & C_{2^{d_Y}} \end{pmatrix}}_{n\cdot 2^{d_Y} \times n\cdot 2^{d_Y}} = \sum_{i=1}^{2^{d_Y}} C_i \otimes \mathbf{e}_i\mathbf{e}_i^T, \tag{24}$$

where $\otimes$ is the Kronecker product between two matrices. $\mathbf{e}_i$, $i = 1, 2, .., 2^{d_Y}$ is the standard basis in $\mathbb{R}^{2^{d_Y}}$. The two terms in the Kronecker product naturally represent the between-class and within-class correlations.

The within-class correlation matrices C_i are assumed to satisfy the following conditions:

(1): C_i is positive definite for $i = 1, 2, 3, .., 2^{d_Y}$;

(2): All the C_i's have the same largest eigenvalue $c > 0$ with the same eigenvector $\mathbf{1} \in \mathbb{R}^n$, i.e. $C_i\mathbf{1} = c\mathbf{1}$.

Under this assumption, the augmented input kernel matrix is $K_X = \sum_{i=1}^{2^{d_Y}} C_i \otimes \mathbf{e}_i\mathbf{e}_i^T + \mathbf{1}\mathbf{1}^T \otimes \mathbf{1}\mathbf{1}^T$. And the output kernel becomes $K_Y = K_Y^0 \otimes \mathbf{1}\mathbf{1}^T$. Here K_Y^0 denotes the $d_Y \times d_Y$ output kernel matrix in the single-element case.

In SI §2.3, we solve the mean-field problem [Eq. (19)] and find the optimal representation kernel is indeed similar as before

$$K[\rho_*] = b_*(d_Y \mathbf{1}\mathbf{1}^T + K_Y^0) \otimes \mathbf{1}\mathbf{1}^T, \tag{25}$$

and b_* is

$$b_* = \sqrt{\frac{\lambda_2}{\lambda_1}\frac{P+c}{(P+2c)P}} - \frac{\lambda_2}{P}. \tag{26}$$

Assume that $\lambda_{1,2}$ are small enough to have $b_* > 0$.

Similarly, the set of optimal neural preactivations $A(K_X, K_Y)$ is given by $2d_Y$ directions

$$A(K_X, K_Y) = \left\{ \mathbf{h} \in \mathbb{R}^{2^{d_Y}} \otimes \mathbb{R}^n \,\middle|\, \mathbf{h} = \alpha(\mathbf{1} + \mathbf{v}_i^0) \otimes \mathbf{1} \text{ or } \alpha(\mathbf{1} - \mathbf{v}_i^0) \otimes \mathbf{1}, \alpha \geq 0 \text{ and } i = 1, 2, .., d_Y \right\}, \tag{27}$$

where $\mathbf{v}_i^0$'s are the eigenvectors of K_Y^0 with nonvanishing eigenvalues.

The optimal kernel [Eq. (25)] (and the associated optimal measure ρ_*) can be attained when the number of hidden neurons $M \geq 2d_Y$. Thus, despite having nonzero within-class correlations, the optimal kernel matrix is similar to the single-element case [Eq.(21)].

This optimal representation kernel [Eq. (25)] has an intriguing geometric interpretation: the between-class correlation matrix $b_*(d_Y \mathbf{1}\mathbf{1}^T + K_Y^0)$ is the same as for the single-element class case [Eq. (21)] and the within-class correlation matrix is given by $\mathbf{1}\mathbf{1}^T$. This form of within-class correlation means the neural representation of all data points from the same binary class "collapse" into a single point in the hidden layer, as also indicated in the set of optimal preactivations [Eq. (27)]. Such a property closely resembles the neural collapse phenomena reported in previous studies [37, 72, 88, 89, 92, 101]. In SI §5.1, we discuss the similarity and difference between our results and the work related to neural collapse. For ReLU nonlinearity, this "within-class collapse" property can be directly proved via Jensen's inequality (SI §2.4). The fact that the optimal preactivation patterns in the hidden layer clusters into only a few directions [Eq. (23) and Eq. (25)] is also known as the quantization phenomenon for ReLU networks [55].

C. Target-aligned input

The above results for whitened input can be generalized to inputs more aligned to the output than the whitened case (Fig. 4C). We will state our main results for single-element ($n = 1$) and multi-element ($n \geq 2$) classes here.

For the single-element class ($n = 1$), we consider the augmented input kernel matrix of the form

$$K_X = \frac{c_0}{P}\mathbf{1}\mathbf{1}^T + \frac{c_Y}{P}K_Y + \sum_{j=d_Y+1}^{P-1} c_j \mathbf{u}_j \mathbf{u}_j^T, \tag{28}$$

where $K_Y = \sum_{i=1}^{d_Y} \mathbf{v}_i \mathbf{v}_i^T$ is the output kernel and $\mathbf{u}_i$'s are the orthonormal basis of span$\{\mathbf{1}, \mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{d_Y}\}^\perp$ in $\mathbb{R}^P$ (Section II A).

The 2nd and 3rd terms in the above equation can be considered input components that are aligned and orthogonal to the output. For whitened inputs considered in Section III A, $c_0 = P + 1, c_Y = c_j = 1$, $j = d_Y + 1.., P - 1$. Here we consider those inputs whose output-aligned component is greater than or equal to the orthogonal component, meaning that

$$c_0 > c_Y \geq c_j > 0, \qquad j = d_Y + 1, .., P - 1. \tag{29}$$

Unlike the fully orthogonalized input before, the input here has positive (homogeneous) correlations if c_0 is large. Interestingly, we find (SI §2.6) that this introduced correlation does not change the form of the optimal kernel $K[\rho_*] = b_*(d_Y \mathbf{1}\mathbf{1}^T + K_Y)$, and only modifies the coefficient b_*

$$b_* = \sqrt{\frac{\lambda_2 c_0 c_Y P}{\lambda_1 [P(c_0 - c_Y) + 2c_Y]}} - \sqrt{\frac{\lambda_2}{P}} \tag{30}$$

The same result holds for multi-element classes ($n \geq 2$, $P = n \cdot 2^{d_Y}$). Now the augmented input kernel has the tensor product form

$$K_X = K_X^0 \otimes C + \mathbf{1}\mathbf{1}^T \otimes \mathbf{1}\mathbf{1}^T \tag{31}$$

where K_X^0 is the input kernel in the single-element class scenario [Eq. (28)] and satisfies Eq. (29). For simplicity, we also assume that all the classes have the same within-class correlation matrix C, which satisfies the same properties as before (Section III A). This input kernel recovers the one for orthogonalized classes (Section III B) if $c_0 = 1, c_Y = c_j = 1$, and recovers single-element class [Eq. (28)] if every class has a single element ($n = 1$). An example dataset with the above input kernel is when the within-class correlation is c_{in} and between-class correlation is c_{out}, with $c_{out} < c_{in}$.

The optimal kernel in this case has the same form as for orthogonalized inputs [Eq. (25)], $K[\rho_*] = b_*(d_Y \mathbf{1}\mathbf{1}^T + K_Y^0) \otimes \mathbf{1}\mathbf{1}^T$, but with an optimal coefficient (SI §2.6)

$$b_* = \sqrt{\frac{\lambda_2}{\lambda_1}\frac{cc_Y(P + cc_0)}{(P + cc_0 + cc_Y)P}} - \frac{\lambda_2}{P}.$$

This optimal kernel (and the associated optimal measures ρ_*) can be attained when the number of hidden neurons $M \geq 2d_Y$. The set of optimal preactivations $A(K_X, K_Y)$ is the same as before [Eq. (23) or Eq. (27)].

Altogether, this shows that for target-aligned inputs (Eq.(29)), the optimal representation kernel [Eq. (28)] does not depend on the magnitude of orthogonal components c_j for all $j = d_Y + 1, .., P - 1$. However, this property does not hold if the target-aligned condition [Eq.(29)] is not satisfied, i.e. when the orthogonal component is large, $c_j > c_Y$ (Fig.5C). The underlying intuition is that for nonlinear networks, the output-aligned component and the output-orthogonal component in the input "compete" with each other to form the hidden representation. In contrast, a linear network only uses the output-aligned component to form the hidden representation.

D. PS, single-neuron tuning and weight matrices for the optimal solution

We have shown that for orthogonalized [Eq. (24)] or target-aligned [Eq. (31)] inputs, when $M \geq 2d_Y$, all global minima of the loss have the representation kernel $K[\rho_*] = b_* \left(d_Y \mathbf{1}\mathbf{1}^T \otimes \mathbf{1}\mathbf{1}^T + K_Y\right)$. In this section, we investigate other properties of these optimal solutions.

From the optimal representation kernel, we can compute its PS, denoted $PS(K[\rho_*])$. By translation- and scale-invariance of PS (Section II B), $PS(K[\rho_*]) = PS(K_Y)$. By the definition of K_Y, the prototype representation [Eq. (6)] of each binary class corresponds to each vertex of a d_Y-dimensional hypercube (Fig. 2). So the representation change for the kth binary latent label is aligned with the kth axis of the hypercube (Fig.4), independent of other latent labels: $PS_k(\alpha_1, \alpha_2) \equiv \frac{\Delta \mathbf{r}(k;\alpha_1) \cdot \Delta \mathbf{r}(k;\alpha_2)}{\|\Delta \mathbf{r}(k;\alpha_1)\|_2 \|\Delta \mathbf{r}(k;\alpha_2)\|_2} = 1$. Therefore, the overall parallelism score $PS(K_Y) = 1 = PS(K[\rho_*])$. And the optimal representation kernel [Eq. (24) and (31)] corresponds to an abstract representation.

Next, we look at the responses of individual neurons in the optimal solution ρ_*. From Eq. (23) and Eq. (27), the optimal preactivation of each neuron has the form $\mathbf{h} = \alpha(1 \pm \mathbf{v}_i^0) \otimes \mathbf{1}$ for some $i \in \{1, 2, 3.., d_Y\}$ and $\alpha > 0$ (we focus on neurons having nonzero preactivation patterns). The P components of $\mathbf{h} \in \mathbb{R}^P \simeq \mathbb{R}^{2^{d_Y}} \otimes \mathbb{R}^n$ are this neuron's preactivations for the P training data points [Eq.(11)]. Based on the definition of $\mathbf{v}_i^0$ (Section II A and SI §1.1), the neuron with $\mathbf{h} = \alpha(1 + \mathbf{v}_i^0) \otimes \mathbf{1}$ (or $\mathbf{h} = \alpha(1 - \mathbf{v}_i^0) \otimes \mathbf{1}$) has nonzero preactivations exactly for those training data points whose ith output label are positive (or negative). So all the neurons in the hidden layer are divided into $2d_Y$ groups: neurons in each group respond only based on a single output label (Fig.5D-F). Note that a random network would not have a modular response property like this, and such modularity is a consequence of task training.

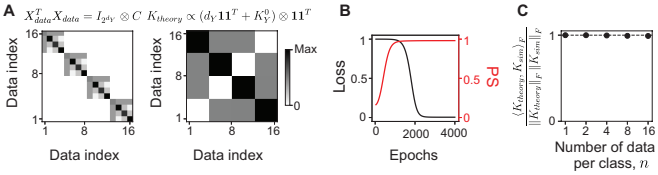

FIG. 5. Task-optimized ReLU network for multi-element classes ($n \geq 2$). (A) An example of input kernel with nonzero within-class correlation but zero between-class correlation (left). For illustration, this example assumes the same within-class correlation matrix C for all classes but our theory also works for class-specific correlation [Eq.(24)]. The theory predicts that the optimal hidden kernel K_{theory} is proportional to the output kernel up to a positive shift (right). In particular, the hidden kernel has a block structure because all the data within each binary class has the same hidden representation [Eq.(27)]. (B) Training loss and PS are plotted against the number of training epochs. (C) The predicted hidden kernel (K_{theory}) always aligns with simulation (K_{sim}) when the number of data per class n changes.

Since the preactivation of every neuron is a linear combination of $\mathbf{1} \otimes \mathbf{1}$ and $\mathbf{v}_i \otimes \mathbf{1}$ for some k, the optimal preactivation matrix H_* [Eq. (11)] has the following form

$$H_* = \underbrace{(\mathbf{1} \otimes \mathbf{1}, \mathbf{v}_1 \otimes \mathbf{1}, \mathbf{v}_2 \otimes \mathbf{1}, .., \mathbf{v}_{d_Y} \otimes \mathbf{1})}_{P \times (d_Y + 1) \text{ matrix}} \mathbf{A},$$

where $\mathbf{A} \in \mathbb{R}^{(d_Y+1) \times M}$ is the matrix encoding the coefficients α for every neuron. Importantly, $H_* \in \mathbb{R}^{P \times M}$ has rank $d_Y + 1$. The corresponding optimal weight matrices for Eq. (4) can be solved as $W_* = H_* X^\dagger$ (since X is of full rank) and $W_{2,*} = Y H_*^T [\lambda_2 + H_* H_*^T]^{-1}$ (SI §1.2). So these matrices also have rank $d_Y + 1$, which could be much smaller than the hidden layer width M and input dimensions d_X. The fact that the optimal weight matrices are low-rank indicates that the neural network learns the task-relevant low-dimensional structures in the input, rather than constructs a large set of fixed features from the inputs as in kernel machine [23, 97].

IV. ABSTRACT REPRESENTATION EMERGES IN THE HIDDEN LAYER, INDEPENDENT OF SINGLE-NEURON NONLINEARITY

We show that the results from the previous section can be extended to other activation functions ϕ beyond ReLU. In particular, the emergence of abstract representation is robust to single-neuron nonlinearity and is mainly determined by the task structure.

We consider two broad classes of nonlinear activation functions. The 1st class is the threshold nonlinear functions of the form

$$\phi(z) = \begin{cases} \phi_+(z) & z \geq 0 \\ 0 & z < 0 \end{cases} \tag{32}$$

Here we require the nonzero function $\phi_+ : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ to satisfy the following properties:

(1): ϕ_+ is continuous, non-decreasing and $\phi_+(0) = 0$;

(2): For any $z > 0$, the slope function $B(z) \equiv \frac{\phi_+(z)}{z} \geq 0$ is non-increasing, and $B_0 \equiv \lim_{z \to 0^+} B(z) \in (0, +\infty)$.

The property **(2)** can be viewed as a saturation effect in the neural responses. Examples in this function class include ReLU, hard Sigmoid, and functions that are concave for positive inputs (Fig. 6A).

The 2nd class of nonlinear activations is those odd functions,

$$\phi(z) = \begin{cases} \phi_+(z) & z \geq 0 \\ -\phi_+(-z) & z < 0 \end{cases}, \tag{33}$$

where ϕ_+ satisfies the same properties **(1)** and **(2)** as above. This class of functions includes linear functions, tanh, and any odd function that is concave for positive inputs (Fig. 7A).

In SI §3, we find that for orthogonalized and target-aligned inputs, the optimal representation kernel converges to the form $K[\rho_*^M] \to a_* \mathbf{1}\mathbf{1}^T \otimes \mathbf{1}\mathbf{1}^T + b_* K_Y$ as $M \to +\infty$. As noted before, this optimal kernel corresponds to an abstract neural representation.

Our central strategy is to (i) prove these results for a perturbed version of the nonlinear function, and then (ii) extend the result via a continuity argument. We outline these key arguments for the single-element class case ($n = 1$) with whitened input. Detailed derivations, as well as the scenarios with multi-element classes ($n \geq 2$) and target-aligned inputs, are provided in SI §3.

A. Threshold nonlinear activation

For any $\delta > 0$ and a 1st class nonlinearity ϕ, we introduce the perturbed activation function

$$\phi^\delta(z) \equiv \begin{cases} B_0 \delta - \phi_+(\delta) + \phi_+(z) & z \geq \delta \\ B_0 z & 0 \leq z \leq \delta \\ 0 & z < 0 \end{cases},$$

where $B_0 = \lim_{z \to 0^+} \frac{\phi_+(z)}{z} > 0$ is the slope of ϕ at the origin. Here ϕ is simply replaced by its linear tangent on $[0, \delta]$. Moreover, it can be checked that if ϕ is a 1st class nonlinearity, so is the perturbed one ϕ^δ.

For this perturbed activation ϕ^δ and whitened input $K_X = I_P + \mathbf{1}\mathbf{1}^T$ (single-element class $n = 1$), we find (SI §3.1-3.2) that a similar result as Eq. (20) holds

$$E(\mathbf{h}; \rho) \geq E_r(\phi^\delta(\mathbf{h}); \rho), \quad \forall \mathbf{h} \in \mathbb{R}^P, \forall \rho \in M_+(K_X), \tag{34}$$

where $E_r(\cdot; \rho)$ is given by Eq.(20) except that the regularization strength λ_1 is rescaled $\lambda_1 \to \lambda_1 B_0^{-2}$. This result is proved via the method of majorization and Schur-convexity [9, 58] (SI §3.1-3.2).

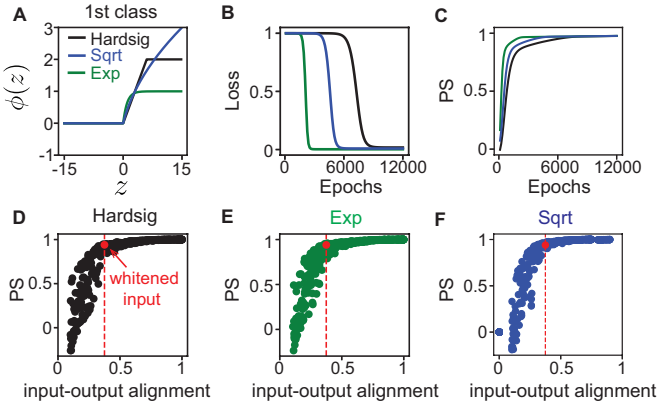

FIG. 6. Task-optimized network with threshold nonlinearity. (A) Three examples of nonlinear activation functions from the 1st class. See detailed expressions for the three functions in SI §5. (B,C) Training loss and PS of the hidden representation are plotted against the number of training epochs, for the three activation functions. (D-F) PS of the optimal hidden representation when varying the input-output alignments. This plot appears to be insensitive to the specific shape of nonlinearity used in the network.

Note that the argument $\phi^\delta(\mathbf{h})$ in $E_r(\phi^\delta(\mathbf{h}); \rho)$ takes values from a hypercube $\phi^\delta(\mathbf{h}) \in [0, \phi_+(+\infty)]^P$ within the nonnegative orthant $\mathbb{R}^P_{\geq 0}$. Minimizing $E_r(\phi^\delta(\mathbf{h}); \rho)$ (SI §3.2), we find that the optimal representation kernel is still of the form $K^\delta[\rho_*] = b_*(d_Y \mathbf{1}\mathbf{1}^T + K_Y)$, with b_*

$$b_* = \sqrt{\frac{\lambda_2 B_0^2}{\lambda_1} \frac{P+1}{P(P+2)}} - \frac{\lambda_2}{P},$$

where $\lambda_{1,2}$ are assumed to be small enough such that $b_* > 0$. So the effect of the nonlinearity ϕ^δ is to simply scale λ_1 by B_0^{-2}, the inverse square of the slope of ϕ_+ at the origin.

The set of optimal preactivations in this case is the same as the ReLU case [Eq. (23)] but with an additional constraint $\mathbf{h} = B_0^{-1} \phi^\delta(\mathbf{h})$ (SI §3.2). Furthermore, this optimal solution is attained when the hidden layer has a width $M > 2 d_Y \lceil \sqrt{b_*} \delta^{-1} \rceil$ (SI §3.2).

Since the optimal kernel (and b_*) is independent of $\delta > 0$, we can carry out the limiting argument $\delta \downarrow 0^+$ and find that this optimal kernel $K^\delta[\rho_*] = b_*(d_Y \mathbf{1}\mathbf{1}^T + K_Y)$ indeed remains unchanged even if $\delta = 0$ (SI §3.4).

While these results show that for whitened and target-aligned inputs, the optimal representation for any 1st class nonlinear activation function ϕ is always abstract, interestingly, we find in simulations that the PS of the optimal representation for other input geometries seems to be also robust to the choice of nonlinearity for a wide network (Fig.6D-F).

B. Odd-symmetric nonlinear activation

Similarly, for the 2nd class of nonlinearity, the perturbed activation function is

$$\phi^\delta(z) \equiv \begin{cases} B_0\delta - \phi_+(\delta) + \phi_+(z) & z \geq \delta \\ B_0 z & -\delta \leq z \leq \delta \\ -B_0\delta + \phi_+(\delta) - \phi_+(-z) & z \leq -\delta \end{cases}$$

where B_0 is the slope at $z \to 0^+$. If ϕ is a 2nd class nonlinearity, so is the perturbed one ϕ^δ.

For single-element class ($n = 1$) and whitened inputs, a similar bound as Eq. (34) holds (SI §3.3)

$$E(\mathbf{h}; \rho) \geq E_r(\phi^\delta(\mathbf{h}); \rho), \quad \forall \mathbf{h} \in \mathbb{R}^P, \forall \rho \in M_+(K_X),$$

where $E_r(\cdot; \rho)$ is given by Eq. (20) except that the regularization strength λ_1 is rescaled $\lambda_1 \to \lambda_1 B_0^{-2}$.

Note that there is one crucial difference for the 2nd class nonlinearity compared to the 1st class nonlinearity: the argument in $E_r(\phi^\delta(\mathbf{h}); \rho)$, $\phi^\delta(\mathbf{h}) \in [-\phi_+(+\infty), \phi_+(+\infty)]^P$ takes values within the hypercube region that is symmetric to the origin. Particularly, it can take negative values. Due to this difference, minimizing $E_r(\phi^\delta(\mathbf{h}); \rho)$ becomes a convex quadratic programming problem rather than a nonconvex copositive programming problem.

Specifically, the unique optimal kernel is found to be

$$K^\delta[\rho_*] = b_* K_Y. \tag{35}$$

where

$$b_* = \sqrt{\frac{\lambda_2 B_0^2}{\lambda_1 P}} - \frac{\lambda_2}{P}.$$

$\lambda_{1,2}$ are small enough to ensure $b_* \geq 0$. We note that there is no constant shift in the kernel.

Moreover, the set of optimal preactivations (SI §3.3) can now vary continuously within a subspace, having the form

$$\mathbf{h} = \sum_{i=1}^{d_Y} \alpha_i \mathbf{v}_i, \; \alpha_i \in \mathbb{R} \text{ and } i \in \{1, 2, .., d_Y\}, \tag{36}$$

with the additional constraint $\mathbf{h} = B_0^{-1}\phi^\delta(\mathbf{h})$. Furthermore, this optimal solution can be attained when the network width $M > d_Y \lceil b_* \delta^{-1} \rceil$.

Despite the differences, the optimal kernel in Eq. (35) still represents an abstract representation. Finally, repeating the limiting argument as before shows that this optimal representation kernel remains unchanged when taking $\delta \downarrow 0^+$ (SI §3.4).

C. Differences in single-neuron tuning

For both classes of nonlinearity [Eq. (32)-(33)], we have shown that the optimal hidden representation in the network is abstract and has PS equal to 1. While the abstractness of the representation is robust to different choices of nonlinearity, the single-neuron tuning properties are different for two classes of nonlinearity (Fig.7D-F).

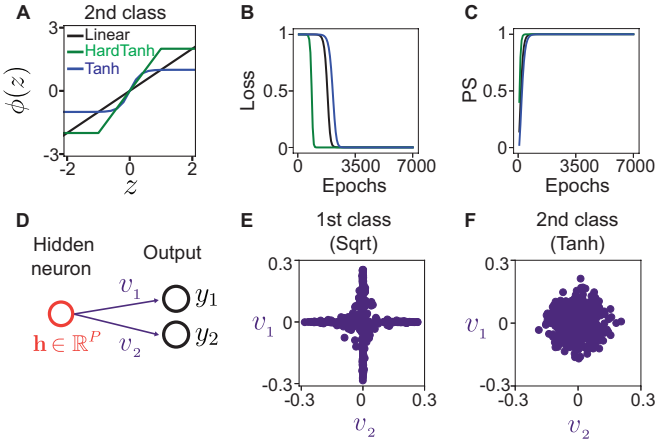

FIG. 7. Task-optimized network with odd-symmetric nonlinearity. (A) Three examples of nonlinear activation functions from the 2nd class. See detailed expressions for the three functions in SI §5. (B,C) Training loss and PS of the hidden representation as a function of the number of training epochs, for the three activation functions in (A). (D-F) Although the optimal hidden representations are always abstract for the two classes of nonlinearity, the hidden neurons have different tuning properties. For 1st class nonlinearity, each hidden neuron is specifically connected to a single output unit (E), while this connectivity appears unstructured for hidden neurons when using the 2nd class nonlinearity (F).

When the activation function is ReLU (which belongs to the 1st class of nonlinearity [Eq. (32)]), we have shown that the optimal neural representation consists only of neurons that are either positively or negatively tuned to a single output label (Section III D), yielding $2d_Y$ groups of neurons. For general 1st class nonlinearity ϕ, since it shares the same set of optimal preactivations [Eq. (23)] as ReLU, this modular tuning remains [68] (Fig. 7E).

On the other hand, for odd activation functions [Eq. (33)] (including the linear activation function), the optimal preactivations [Eq. (36)] can freely rotate in a subspace (when $\mathbf{h}$ is small). So in this case, neurons in the hidden layer generally exhibit mixed selectivity [29, 78] (Fig. 7F).

Therefore, even if both classes of nonlinearity generate the same optimal population geometry (characterized by the same representation kernel $K[\rho_*]$ up to a global shift), they lead to different single-neuron tuning properties. This suggests that the optimal tuning curves on the single-neuron level are not only impacted by the task structure, but also by the biophysical properties of individual neurons (which determine the response nonlinearity).

V. EXTENSIONS

In the previous sections, we use our analytical framework to study optimal neural representations for two-layer networks trained on the data model in Section II A. However, the framework is indeed applicable to additional data models and deep neural networks. For illustrative purposes, we present these extensions here, with a focus on the single-element class ($n = 1$) with ReLU nonlinear activation. These results can be extended to any nonlinearity as before (Section IV).

We first derive some general properties of the optimal representation kernel derived using our analytical framework (Section II). Given any input and output matrices X, Y, denote the set of optimal representation kernels as $S(X, Y) \subseteq \mathbb{R}^{P\times P}$. According to Eq. (13), the optimal preactivation matrix H and thus the optimal representation kernel only depend on the input and output kernels, $S(X, Y) \equiv S(K_X, K_Y)$. Moreover, this set is invariant under global scaling factors of the input and output kernels (for ReLU), $S(\alpha_X K_X, \alpha_Y K_Y) = S(K_X, K_Y)$ if $\alpha_X, \alpha_Y > 0$.

(1) Denote the set of optimal measures of Eq. (15) as $\mathcal{P}(K_X, K_Y) \subseteq M_+(X)$. Since the problem is a convex optimization problem, the optimal solution set, $\mathcal{P}(K_X, K_Y)$ is a convex set and in particular, is simply connected. Because the kernel $K[\cdot]$ is a linear map on $M_+(X)$, the set of optimal representation kernel matrices $S(K_X, K_Y) = K[\mathcal{P}(K_X, K_Y)]$ is also convex and simply connected. For ReLU nonlinearity, $S(K_X, K_Y)$ is a subset of the set of completely positive matrices and can be attained when the network width $M \geq 1+P(P+1)/2$ [85]. This property is usually called mode connectivity of the loss landscape [4, 28, 64, 86], and our analytical framework provides an alternative way to uncover this property.

(2) For input kernel K_X and any optimal kernel $K[\rho_*] \in S(K_X, K_Y)$, denote the set of optimal preactivations as $A(K_X, K_Y)$. Now if a rank one perturbation is added to the input kernel, $K_X^{new} = K_X - a\mathbf{v}\mathbf{v}^T$, $a \in \mathbb{R}, \mathbf{v} \in \text{Range}K_X$, and moreover,

$$a\mathbf{v}^T K_X^\dagger \mathbf{v} < 1 \quad \text{and} \quad K_X^\dagger \mathbf{v} \in A(K_X, K_Y)^\perp,$$

then the same kernel $K[\rho_*]$ is also optimal for this perturbed input (SI §4.1). This property shows that the optimal hidden representation is robust to input perturbation along certain directions.

(3) If the pseudo-inverse of the input kernel $K_X^\dagger$ has negative off-diagonals, we can show that the optimal set of neural preactivations lies entirely within the nonnegative orthorant, $A(K_X, K_Y) \subseteq \mathbb{R}^P_{\geq 0}$. And this property holds independently of the output kernel K_Y. Under this condition, the problem of training two-layer network [Eq. (4)] is equivalent to a dictionary learning problem with positive representations. Input kernels K_X satisfying such properties are known as inverse M-matrix [19, 27, 38, 39]. (A notable special case are strictly ultrametric matrices [26, 62], see details in SI §4.2).

(4) For any optimal measure $\rho_* \in \mathcal{P}(K_X, K_Y)$, the output of the network for a new data point $\mathbf{x} \in \mathbb{R}^{d_X}$ is (SI §4.3),

$$\mathbf{y}_*(\mathbf{x}) = Y \frac{1}{\lambda_2 + K[\rho_*]} \int \phi(\mathbf{h})\phi(\mathbf{h}^T K_X^\dagger X^T \mathbf{x}) d\rho_*(\mathbf{h}),$$

where X is the augmented input matrix [Eq. (5)] and Y is the output matrix [Eq. (2)]. This result thus allows us to investigate the generalization property of the optimal network solution. As an illustration, we apply it to simple compositional-generalization tasks (see SI §4.4).

A. Anisotropic input-output geometry

In this section, we consider a scenario where the input and output kernels are

$$K_X = \frac{c_0}{P}\mathbf{1}\mathbf{1}^T + \sum_{i=1}^{d_Y} \frac{c_i}{P}\mathbf{v}_i\mathbf{v}_i^T + \sum_{j=d_Y+1}^{P-1} c_j \mathbf{u}_j\mathbf{u}_j^T,$$

$$K_Y = \sum_{i=1}^{d_Y} d_i \mathbf{v}_i\mathbf{v}_i^T.$$

According to the definition of $\mathbf{v}_i$ (Section II A), this would correspond to anisotropic output labels where the ith output direction is scaled by a factor $d_i > 0$ for different i's, yielding a hierarchical structure in the output kernel (Fig. 8A). We assume that the inputs are also anisotropic ($c_i > 0$'s are different) and are target-aligned,

$$c_0 > \max_{i=1,..d_Y} c_i, \quad \text{and} \quad \min_{i=1,..d_Y} c_i \geq \max_{j=d_Y+1,..P-1} c_j.$$

This recovers scenarios in previous sections if all c_i's and d_i's are equal to each other.

We solve the mean-field problem [Eq. (19)] in this case (SI §4.5). And this yields the unique optimal hidden representation kernel

$$K[\rho_*] = \sum_{i=1}^{d_Y} b_i^* \left(\mathbf{1}\mathbf{1}^T + \mathbf{v}_i\mathbf{v}_i^T\right)$$

where the coefficient b^* is

$$b_i^* = \sqrt{\frac{\lambda_2 d_i c_0 c_i P}{\lambda_1(c_0 + c_i)}} - \sqrt{\frac{\lambda_2}{P}}.$$

When $d_i = 1, c_i = c_Y$, this solution recovers Eq. (30). We see that up to a global translation, the effect of anisotropic input and output just scales the ith direction of the hidden representation by a factor b_i^* that depends on d_i. The hidden representation still has a hypercube (or hyper-rectangular) geometry (Fig. 8C) and is abstract. However, the anisotropy in the training data induces a more pronounced stage-like transition in the learning dynamics (Fig. 8C).

B. Deep neural network

The analytical framework presented in Section II generalizes naturally to deep networks (Fig.8D). We consider a deep feedforward network

$$f_{\{W^l\}}(\mathbf{x}) = W^L\phi(W^{L-1}\phi(..\phi(W^1\mathbf{x}))),$$

For convenience, we have incorporated the bias parameter of the 1st layer into the input $\mathbf{x}$, and the rest of the layers are bias-free. The loss function is

$$E(\{W^l\}) \equiv \left\|Y - f_{\{W^l\}}(X)\right\|_F^2 + \sum_{l=1}^{L} \lambda_l \left\|W^l\right\|_F^2 .$$

We are interested in when the last layer in this network exhibits abstract representations (Fig.8D) given small regularization parameters $\lambda_l \ll 1, l = 1, 2, .., L$.

Following similar procedures as in Section II C, we can introduce the empirical measure for preactivations for each layer $\rho^l = \sum_{k=1}^{M} \delta_{\mathbf{h}_k^l}$ and derive the KKT conditions for the optimal solutions (SI §4.6).

For the data model in Section II A with whitened input $K_X = I_P + \mathbf{1}\mathbf{1}^T$, we solve these KKT conditions to get an optimal representation kernel of the form

$$K[\rho_*^l] = b_*^l(d_Y\mathbf{1}\mathbf{1}^T + K_Y), \qquad l = 1, 2, .., L.$$

In the limit $\lambda_l \equiv \lambda \downarrow 0^+$, $l = 1, 2, .., L$, these coefficients are

$$b_*^1 = \gamma_* \frac{(d_Y + 1)(P + 1)}{d_Y P(P + 2)},$$
$$b_*^l = (\gamma_*)^{l-1} b_*^1, \quad l = 2, ..., L - 1.$$

where $\gamma_* = \sqrt[L]{\frac{d_Y^2 P(P+2)}{(d_Y+1)^2(P+1)}} + O(\lambda_l)$. The above result gives an optimal network that exhibits abstract representation in the last layer (and all the other layers, Fig. 8D) and can be attained when the width of every layer $M \geq 2d_Y$ (SI §4.6). In general, the effective energy function for deep networks ($L \geq 3$) is not convex over the space of measures [unlike Eq.(16)]. But the above solution is always a strict local minimum of the loss for any number of hidden layers $L \geq 2$ (SI §4.6).

Finally, the analytical framework is extended to analyze the optimal representation in recurrent neural networks (SI §4.7). The loss functional in this case can be written as a functional over the space of measures of the temporal trajectory of the neural preactivations. As in the deep feedforward network case, this loss functional is generally not a convex functional. The KKT condition for a recurrent network depends on the full temporal trajectory of preactivations (SI §4.7), rather than factorizing cleanly at each layer as in the feedforward case (SI §4.6). Nevertheless, we find that for whitened inputs and the data model in Section II A, the learned representation at the last timestep remains abstract (Fig. 8E).

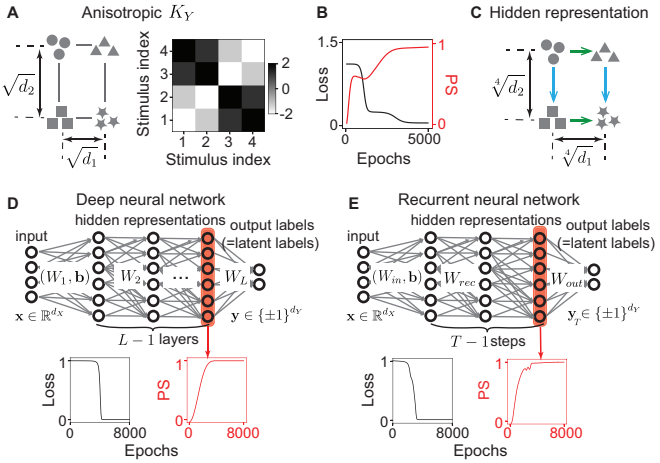

FIG. 8. Extensions of the analytical framework to anisotropic input-output geometry, deep feedforward network, and recurrent network models. (A) For anisotropic output geometry, different output dimensions are scaled by a different factor $\sqrt{d_i}$. The corresponding output kernel matrix is shown to have a hierarchical block structure. (B) Training loss and PS of hidden representation are plotted against the number of training epochs. The training dynamics has more prominent stage-like transitions than the case with isotropic outputs in Figure 4A. (C) The optimal hidden representation is aligned with the output geometry in (A), except that the axis representing each latent label is rescaled. It has PS equal to 1. (D) A deep neural network trained on related tasks develops an abstract representation in its last layer. (E) A recurrent neural network trained on the related tasks develops an abstract representation at the last timestep.

VI. DISCUSSION

The low-dimensional disentangled abstract representations of task-relevant variables have been observed in multiple brain areas of multiple species (Fig.1, [8, 24, 61, 70, 91]).Despite the ubiquity of this type of representational geometry, we still do not know the network mechanisms that lead to its formation. Here, we showed that abstract representations naturally emerge in simple feedforward neural networks when the networks are optimized on tasks that depend on the variables that should be represented in an abstract format. In these networks, the geometry of the hidden layer reflects the geometry of the outputs (labels, Fig.1-2). The input geometry or input encoding also has an important effect on the representational geometry learned in the hidden layer (Fig.4C, Fig.6D-F and [2, 41]): when the input and output geometries are aligned and abstract, it is not surprising that the hidden representation will also be abstract. It is less obvious that whitened inputs — despite not being aligned with any specific low-dimensional representation geometry — nonetheless facilitate the emergence of low-dimensional abstract neural representations. This facilitation arises from the maximal dimensionality of the whitened data,

which allows the hidden-layer neural representation to "move around more freely" in the high-dimensional space and inherit the low-dimensional structure from the output. In the brain, these high-dimensional representations are likely the result of some form of 'recoding' [56, 57] which is probably implemented in the hippocampus[7, 13, 30, 75, 90]. Earlier work has shown that even random dimensionality expansion increases the separability between different stimuli and helps discrimination [5, 52, 59]. Here, we suggest that these expansion layers offer another benefit: they facilitate better representation learning in downstream brain areas. These learned low-dimensional representations would support sample-efficient generalization for novel tasks [6, 93].

In order to investigate the optimal neural representation in the network models, we developed a mathematical framework that maps the original problem of weight space optimization to an optimization problem over the neural preactivations [Fig.3A-C, Eq. (13)]. The new problem corresponds to an effective model where neurons interact with each other through a "mean-field" induced by the overall distribution of neural activity [Eq. (14), Fig. 3]. We derive the KKT condition for the optimal distribution of neural activity [Eq. (17)]. For a large class of input kernels, the KKT condition is equivalent to a nonnegative quadratic programming problem, which we solve in closed form when the outputs exhibit cubic symmetry (SI §2.2). Crucially, because the new optimization problem is convex, any solution of the KKT condition is automatically a global minimum of the loss function. We show that, in many cases, all such solutions correspond to an abstract neural representation. To our knowledge, this provides the first theoretical result showing the robust emergence of abstract representation in nonlinear neural networks, when trained to binary decision tasks closely related to prior neuroscience experiments (Section II A, [8, 17]). These results, together with the analytical framework developed to derive them, constitute the main innovations of this work.

Our work thus provides insights into why abstract representations appear in the brain [8, 24, 61, 70, 91]. Our analytical framework is also a more general tool for analyzing the optimal representations for various tasks used in the machine learning literature. Our framework is a complementary approach to many existing works on analyzing learning in two-layer neural networks. Here we highlight and compare it with two sets of related work (see a more comprehensive discussion in SI §5):

A substantial body of prior work has investigated two-layer networks from the perspective of Bayesian neural networks. Leveraging tools from statistical physics, these studies have shown that certain scaling limits of the model lead to Gaussian equivalence properties [31, 49, 71], wherein the neural preactivations follow a joint Gaussian distribution—a regime closely linked to the kernel limit or lazy regime [23, 36, 97]. More recent work has explored alternative scaling limits [10, 48, 94, 98, 99] that are connected to the mean-field limit of deep networks. While these results primarily concern the (infinite-width or infinite-input-dimension) scaling limits, our framework enables direct analysis of optimal solutions in finite-width networks with finite-dimensional inputs. These results for finite-width networks allow us to make statements about the optimal representation in various infinite-width limits (SI §5.1). Furthermore, whereas Bayesian approaches typically average the network properties over the posterior distribution, our method provides insights into the structure of individual global minima of the loss function. We comment that when deriving the scaling limit from our finite-width results, we first take the infinite temperature limit $\beta \to +\infty$ followed by infinite-width limit $M \to +\infty$. As a consequence, the resulting network always learns the low-dimensional features in the data. This is different from the usual Bayesian network setting where $M \to +\infty$ is taken first and then $\beta \to +\infty$, and feature learning only happens when using a special weight scaling in the loss function.

Another series of work examined the learning dynamics of two-layer networks. Earlier works focused on single-layer perceptron [22, 73, 84] and deep linear networks [82, 83]. Recently, the learning dynamics of two-layer nonlinear networks were also studied both in the mean-field regime [15, 60, 79, 87] and in finite-width settings [11]. These analyses on nonlinear networks are typically only tractable for a one-dimensional output. However, abstract representations (Fig. 1) concern the relationship between the hidden representations of different output dimensions (see Section II B). So we adopt a different approach here: rather than tracking the entire training trajectory, we directly analyze global minima of the loss function [Eq. (4)]. An intriguing future direction would be to extend the existing results for learning dynamics with one-dimensional outputs, to those tasks in Section II A that require multi-dimensional outputs, and investigate how the hidden representation kernel evolves during training.

A byproduct of our analysis is a characterization of single-neuron selectivity in the optimal network solution. This is determined by the set of optimal preactivation vectors $A(K_X, K_Y)$ [Eq.(23), (25) and (36)], whose components specify the neural response to each stimulus in the training set. A longstanding question in neuroscience is whether neurons exhibit "interpretable" tuning—activity explained by a single task-relevant variable—or "mixed" tuning, where responses depend on combinations of multiple variables [29]. Experimental data and models have suggested that the tuning type depends on details of task structure [21, 41, 96], brain regions [74], and species. For the task and network architecture studied here (Section II A), we find that the nonlinearity in the hidden layer plays a key role (see also [2]): depending on its form, the task optimization yields either distinct neural modules tuned to individual binary

latent variables or neurons with linear mixed selectivity to multiple latent variables. In the brain, the neuron nonlinearity varies across regions due to differences in single-neuron biophysical and morphological properties [76].

Despite such variability on the single-neuron level, we demonstrated that in the large-network limit, the emergence of abstract representations on the population level is robust to the specific form of single-neuron nonlinearities. This universality result offers a potential explanation for recent experimental observations of abstract representations across diverse brain areas. Previous studies have shown that neural network models, under certain scaling limits, exhibit Gaussian universality or Gaussian equivalence properties, typically attributed to central limit theorem-like mechanisms [31, 33]. In contrast, the optimal preactivation distributions (ρ_*) in our model are non-Gaussian, suggesting a different origin of universality. We believe that the shared task structure [Section II A] on which all these models are trained is the underlying reason for this universal abstract representation. This insight aligns with recent empirical findings that networks with different architectures, when trained on similar tasks, often converge to similar neural representations—a phenomenon referred to as the Platonic representation hypothesis [34, 45, 50]. Our results thus provide a tractable mathematical model for this hypothesis and illustrate that neural representations for a task on the population level (in terms of representation kernel), can be relatively insensitive to single-neuron-level response details.

Finally, the order parameter ρ—defined as the distribution over neural preactivations—provides a powerful tool for exploiting the permutation symmetry inherent in these network models [4, 15, 60, 87]. We anticipate that this formulation will not only aid in analyzing the optimal solutions in the networks studied here but may also be extended to investigate a broader class of permutation-symmetric models, including ResNet and transformer models in modern AI systems, as well as to analyze learning dynamics in network models trained with biological learning rules.

VII. ACKNOWLEDGEMENT

This work is supported by NSF DBI-2229929 (ARNI), NIH NINDS K99NS138578 (WJJ), the Simons Foundation (542983SPI, SF), the Swartz Foundation, the Gatsby Charitable Foundation, and the Kavli Foundation.

[1] The order parameter for spin glasses: a function on the interval 0-1. *Journal of Physics A: Mathematical and General*, 13(3):1101, 1980.

[2] Matteo Alleman, Jack W Lindsey, and Stefano Fusi. Task structure and nonlinearity jointly determine learned representational geometry. *arXiv preprint arXiv:2401.13558*, 2024.

[3] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks (2017). *arXiv preprint arXiv:1701.04862*, 2017.

[4] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017.

[5] Omri Barak, Mattia Rigotti, and Stefano Fusi. The sparseness of mixed selectivity neurons controls the generalization–discrimination trade-off. *Journal of Neuroscience*, 33(9):3844–3856, 2013.

[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[7] Marcus K Benna and Stefano Fusi. Place cells may simply be memory cells: Memory compression leads to spatial tuning and history dependence. *Proceedings of the National Academy of Sciences*, 118(51):e2018422118, 2021.

[8] Silvia Bernardi, Marcus K Benna, Mattia Rigotti, Jérôme Munuera, Stefano Fusi, and C Daniel Salzman. The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell*, 183(4):954–967, 2020.

[9] Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

[10] Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. *Advances in Neural Information Processing Systems*, 35:32240–32256, 2022.

[11] Etienne Boursier, Loucas Pillaud-Vivien, and Nicolas Flammarion. Gradient flow dynamics of shallow relu networks for square loss and orthogonal inputs. *Advances in Neural Information Processing Systems*, 35:20105–20118, 2022.

[12] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[13] Lara M Boyle, Lorenzo Posani, Sarah Irfan, Steven A Siegelbaum, and Stefano Fusi. Tuned geometries of hippocampal representations meet the computational demands of social memory. *Neuron*, 112(8):1358–1371, 2024.

[14] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in \beta-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[15] Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in neural information processing systems*, 31, 2018.

[16] Chi-Ning Chou, Royoung Kim, Luke A Arend, Yao-Yuan Yang, Brett D Mensh, Won Mok Shim, Matthew G Perich, and SueYeon Chung. Geometry linked to untangling efficiency reveals structure and computation in neural populations. *bioRxiv*, pages

2024–02, 2025.

[17] Hristos S Courellis, Juri Minxha, Araceli R Cardenas, Daniel L Kimmel, Chrystal M Reed, Taufik A Valiante, C Daniel Salzman, Adam N Mamelak, Stefano Fusi, and Ueli Rutishauser. Abstract representations emerge in human hippocampal neurons during inference. *Nature*, 632(8026):841–849, 2024.

[18] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[19] Claude Dellacherie, Servet Martinez, Jaime San Martin, et al. *Inverse M-matrices and ultrametric matrices*, volume 2118. Springer, 2014.

[20] Adrien Doerig, Rowan P Sommers, Katja Seeliger, Blake Richards, Jenann Ismael, Grace W Lindsay, Konrad P Kording, Talia Konkle, Marcel AJ Van Gerven, Nikolaus Kriegeskorte, et al. The neuroconnectionist research programme. *Nature Reviews Neuroscience*, 24(7):431–450, 2023.

[21] Will Dorrell, Kyle Hsu, Luke Hollingsworth, Jin Hwa Lee, Jiajun Wu, Chelsea Finn, Peter E Latham, Tim EJ Behrens, and James CR Whittington. Range, not independence, drives modularity in biologically inspired representations. *arXiv preprint arXiv:2410.06232*, 2024.

[22] Andreas Engel. *Statistical mechanics of learning*. Cambridge University Press, 2001.

[23] Matthew Farrell, Stefano Recanatesi, and Eric Shea-Brown. From lazy to rich to exclusive task representations in neural networks and neural codes. *Current opinion in neurobiology*, 83:102780, 2023.

[24] Valeria Fascianelli, Aldo Battista, Fabio Stefanini, Satoshi Tsujimoto, Aldo Genovesio, and Stefano Fusi. Neural representational geometries reflect behavioral differences in monkeys and recurrent neural networks. *Nature Communications*, 15(1):6479, 2024.

[25] Jorge Fernandez-de Cossio-Diaz, Simona Cocco, and Rémi Monasson. Disentangling representations in restricted boltzmann machines without adversaries. *Physical Review X*, 13(2):021003, 2023.

[26] Miroslav Fiedler. Some characterizations of symmetric inverse m-matrices. *Linear algebra and its applications*, 275:179–187, 1998.

[27] Miroslav Fiedler and Vlastimil Pták. On matrices with non-positive off-diagonal elements and positive principal minors. *Czechoslovak Mathematical Journal*, 12(3):382–400, 1962.

[28] C Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization. *arXiv preprint arXiv:1611.01540*, 2016.

[29] Stefano Fusi, Earl K Miller, and Mattia Rigotti. Why neurons mix: high dimensionality for higher cognition. *Current opinion in neurobiology*, 37:66–74, 2016.

[30] Mark A Gluck and Catherine E Myers. Psychobiological models of hippocampal function in learning and memory. *Neurobiology of learning and memory*, pages 417–448, 1998.

[31] Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová. Modeling the influence of data structure on learning in neural networks: The hidden manifold model. *Physical Review X*, 10(4):041044, 2020.

[32] Daniella Horan, Eitan Richardson, and Yair Weiss. When is unsupervised disentanglement possible? *Advances in Neural Information Processing Systems*, 34:5150–5161, 2021.

[33] Hong Hu and Yue M. Lu. Universality laws for high-dimensional learning with random features. *IEEE Transactions on Information Theory*, 69(3):1932–1964, 2023.

[34] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Forty-first International Conference on Machine Learning*, 2024.

[35] Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

[36] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[37] Arthur Jacot, Peter Súkeník, Zihan Wang, and Marco Mondelli. Wide neural networks trained with weight decay provably exhibit neural collapse. *arXiv preprint arXiv:2410.04887*, 2024.

[38] Charles R Johnson. Inverse m-matrices. *Linear Algebra and its Applications*, 47:195–216, 1982.

[39] Charles R Johnson and Ronald L Smith. Inverse m-matrices, ii. *Linear algebra and its applications*, 435(5):953–983, 2011.

[40] W Jeffrey Johnston and Stefano Fusi. Abstract representations emerge naturally in neural networks trained to perform multiple tasks. *Nature Communications*, 14(1):1040, 2023.

[41] W Jeffrey Johnston and Stefano Fusi. Modular representations emerge in neural networks trained to perform context-dependent tasks. *bioRxiv*, 2024.

[42] Robert O Jones. Density functional theory: Its origins, rise to prominence, and future. *Reviews of modern physics*, 87(3):897–923, 2015.

[43] Nancy Kanwisher, Meenakshi Khosla, and Katharina Dobs. Using artificial neural networks to ask 'why'questions of minds and brains. *Trends in Neurosciences*, 46(3):240–254, 2023.

[44] Matthew T Kaufman, Marcus K Benna, Mattia Rigotti, Fabio Stefanini, Stefano Fusi, and Anne K Churchland. The implications of categorical and category-free mixed selectivity on representational geometries. *Current opinion in neurobiology*, 77:102644, 2022.

[45] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.

[46] Yoshiki Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In *International symposium on mathematical problems in theoretical physics: January 23–29, 1975, kyoto university, kyoto/Japan*, pages 420–422. Springer, 1975.

[47] Yoshiki Kuramoto and Yoshiki Kuramoto. *Chemical turbulence*. Springer, 1984.

[48] Clarissa Lauditi, Blake Bordelon, and Cengiz Pehlevan. Adaptive kernel predictors from feature-learning infinite limits of neural networks. *arXiv preprint arXiv:2502.07998*, 2025.

[49] Qianyi Li and Haim Sompolinsky. Statistical mechanics of deep linear neural networks: The backpropagating kernel renormalization. *Physical Review X*, 11(3):031059, 2021.

[50] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent learning: Do different neural networks learn the same representations? *arXiv preprint arXiv:1511.07543*, 2015.

[51] Qiyao Liang, Daoyuan Qian, Liu Ziyin, and Ila Fiete. Compositional generalization via forced rendering of disentangled latents. *arXiv preprint arXiv:2501.18797*, 2025.

[52] Ashok Litwin-Kumar, Kameron Decker Harris, Richard Axel, Haim Sompolinsky, and LF Abbott. Optimal degrees of synaptic connectivity. *Neuron*, 93(5):1153–1164, 2017.

[53] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[54] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. A sober look at the unsupervised learning of disentangled representations and their evaluation. *Journal of Machine Learning Research*, 21(209):1–62, 2020.

[55] Hartmut Maennel, Olivier Bousquet, and Sylvain Gelly. Gradient descent quantizes relu network features. *arXiv preprint arXiv:1803.08367*, 2018.

[56] David Marr and W Thomas Thach. A theory of cerebellar cortex. *From the retina to the neocortex: selected papers of David Marr*, pages 11–50, 1991.

[57] David Marr, David Willshaw, and Bruce McNaughton. *Simple memory: a theory for archicortex*. Springer, 1991.

[58] Albert W Marshall, Ingram Olkin, and Barry C Arnold. Inequalities: theory of majorization and its applications. 1979.

[59] James L McClelland and Nigel H Goddard. Considerations arising from a complementary learning systems perspective on hippocampus and neocortex. *Hippocampus*, 6(6):654–665, 1996.

[60] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

[61] Karyna Mishchanchuk, Gabrielle Gregoriou, Albert Qü, Alizée Kastler, Quentin JM Huys, Linda Wilbrecht, and Andrew F MacAskill. Hidden state inference requires abstract contextual representations in the ventral hippocampus. *Science*, 386(6724):926–932, 2024.

[62] Reinhard Nabben and Richard S Varga. A linear algebra proof that the inverse of a strictly ultrametric matrix is a strictly diagonally dominant stieltjes matrix. *SIAM Journal on Matrix Analysis and Applications*, 15(1):107–113, 1994.

[63] John W Negele. The mean-field theory of nuclear structure and dynamics. *Reviews of Modern Physics*, 54(4):913, 1982.

[64] Quynh Nguyen. On connected sublevel sets in deep learning. In *International conference on machine learning*, pages 4790–4799. PMLR, 2019.

[65] Ramon Nogueira, Chris C Rodgers, Randy M Bruno, and Stefano Fusi. The geometry of cortical representations of touch in rodents. *Nature Neuroscience*, 26(2):239–250, 2023.

[66] Here the support of a measure suppρ can be intuitively thought of as the largest region where the measure has nonzero mass, and it is formally defined as the largest closed set $A \subseteq \mathbb{R}^P$ such that $\rho(A^c) = 0$.

[67] To be technically correct, we also require all 2nd-order moments of ρ are finite in $M_+(X)$. This would ensure all the integrals in $E[\rho]$ are finite.

[68] More precisely, we show that in SI §3.1-3.2, the perturbed nonlinearity shares the same set of optimal preactivations as ReLU. By continuity, this form of modularity persists when $\delta \downarrow 0^+$.

[69] Manfred Opper and David Saad. *Advanced mean field methods: Theory and practice*. MIT press, 2001.

[70] Pia-Kelsey O'Neill, Lorenzo Posani, Jozsef Meszaros, Phebe Warren, Carl E Schoonover, Andrew JP Fink, Stefano Fusi, and C Daniel Salzman. The representational geometry of emotional states in basolateral amygdala. *BioRxiv*, pages 2023–09, 2023.

[71] R Pacelli, S Ariosto, Mauro Pastore, F Ginelli, Marco Gherardi, and Pietro Rotondo. A statistical mechanics framework for bayesian deep neural networks beyond the infinite-width limit. *Nature Machine Intelligence*, 5(12):1497–1507, 2023.

[72] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

[73] Nishil Patel, Sebastian Lee, Stefano Sarao Mannelli, Sebastian Goldt, and Andrew Saxe. Rl perceptron: Generalization dynamics of policy learning in high dimensions. *Physical Review X*, 15(2):021051, 2025.

[74] Lorenzo Posani, Shuqi Wang, Samuel P Muscinelli, Liam Paninski, and Stefano Fusi. Rarely categorical, always high-dimensional: how the neural code changes along the cortical hierarchy. *bioRxiv*, pages 2024–11, 2025.

[75] James B Priestley, John C Bowler, Sebi V Rolotti, Stefano Fusi, and Attila Losonczy. Signatures of rapid plasticity in hippocampal ca1 representations during novel experiences. *Neuron*, 110(12):1978–1992, 2022.

[76] Alexander Rauch, Giancarlo La Camera, Hans-Rudolf Luscher, Walter Senn, and Stefano Fusi. Neocortical pyramidal cells respond as integrate-and-fire neurons to in vivo–like input currents. *Journal of neurophysiology*, 90(3):1598–1612, 2003.

[77] Blake A Richards, Timothy P Lillicrap, Philippe Beaudoin, Yoshua Bengio, Rafal Bogacz, Amelia Christensen, Claudia Clopath, Rui Ponte Costa, Archy de Berker, Surya Ganguli, et al. A deep learning framework for neuroscience. *Nature neuroscience*, 22(11):1761–1770, 2019.

[78] Mattia Rigotti, Omri Barak, Melissa R Warden, Xiao-Jing Wang, Nathaniel D Daw, Earl K Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):585–590, 2013.

[79] Grant M Rotskoff and Eric Vanden-Eijnden. Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error. *stat*, 1050:22, 2018.

[80] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[81] Andrew Saxe, Stephanie Nelli, and Christopher Summerfield. If deep learning is the answer, what is the question? *Nature Reviews Neuroscience*, 22(1):55–67, 2021.

[82] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[83] Andrew M Saxe, James L McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019.

[84] Hyunjune Sebastian Seung, Haim Sompolinsky, and Naftali Tishby. Statistical mechanics of learning from examples. *Physical review A*, 45(8):6056, 1992.

[85] Naomi Shaked-Monderer and Abraham Berman. *Copositive and completely positive matrices*. World Scientific, 2021.

[86] Berfin Simsek, François Ged, Arthur Jacot, Francesco Spadaro, Clément Hongler, Wulfram Gerstner, and Johanni Brea. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In *International Conference on Machine Learning*, pages 9722–9732. PMLR, 2021.

[87] Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics*, 80(2):725–752, 2020.

[88] Peter Súkeník, Christoph Lampert, and Marco Mondelli. Neural collapse vs. low-rank bias: Is deep neural collapse really optimal? In *38th Annual Conference on Neural Information Processing Systems*, volume 38, 2024.

[89] Peter Súkeník, Marco Mondelli, and Christoph H Lampert. Deep neural collapse is provably optimal for the deep unconstrained features model. *Advances in Neural Information Processing Systems*, 36:52991–53024, 2023.

[90] Weinan Sun, Johan Winnubst, Maanasa Natrajan, Chongxi Lai, Koichiro Kajikawa, Arco Bast, Michalis Michaelos, Rachel Gattoni, Carsen Stringer, Daniel Flickinger, et al. Learning produces an orthogonalized state machine in the hippocampus. *Nature*, pages 1–11, 2025.

[91] Wenbo Tang, Justin D Shin, and Shantanu P Jadhav. Geometric transformation of cognitive maps for generalization across hippocampal-prefrontal circuits. *Cell reports*, 42(3), 2023.

[92] Tom Tirer and Joan Bruna. Extended unconstrained features model for exploring deep neural collapse. In *International Conference on Machine Learning*, pages 21478–21505. PMLR, 2022.

[93] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[94] Alexander van Meegen and Haim Sompolinsky. Coding schemes in neural networks learning classification tasks. *Nature Communications*, 16(1):3354, 2025.

[95] Albert J Wakhloo, Will Slatton, and SueYeon Chung. Neural population geometry and optimal coding of tasks with shared latent structure. *arXiv preprint arXiv:2402.16770*, 2024.

[96] James CR Whittington, Will Dorrell, Surya Ganguli, and Timothy EJ Behrens. Disentanglement with biological constraints: A theory of functional cell types. *arXiv preprint arXiv:2210.01768*, 2022.

[97] Blake Woodworth, Suriya Gunasekar, Jason D. Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 3635–3673. PMLR, 09–12 Jul 2020.

[98] Greg Yang. Tensor programs ii: Neural tangent kernel for any architecture. *arXiv preprint arXiv:2006.14548*, 2020.

[99] Greg Yang and Edward J Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, pages 11727–11737. PMLR, 2021.

[100] Guangyu Robert Yang and Xiao-Jing Wang. Artificial neural networks for neuroscientists: a primer. *Neuron*, 107(6):1048–1070, 2020.

[101] Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features. *Advances in Neural Information Processing Systems*, 34:29820–29834, 2021.