


Emergence of hierarchical modes from deep learning

Chan Li¹ and Haiping Huang^{1,2,*}

¹PMI Laboratory, School of Physics, Sun Yat-sen University, Guangzhou 510275, People's Republic of China

²Guangdong Provincial Key Laboratory of Magnetolectric Physics and Devices, Sun Yat-sen University, Guangzhou 510275, People's Republic of China

 (Received 24 August 2022; revised 27 February 2023; accepted 29 March 2023; published 11 April 2023)

Large-scale deep neural networks consume expensive training costs, but the training results in less-interpretable weight matrices constructing the networks. Here, we propose a mode decomposition learning that can interpret the weight matrices as a hierarchy of latent modes. These modes are akin to patterns in physics studies of memory networks, but the least number of modes increases only logarithmically with the network width and even becomes a constant when the width grows further. The mode decomposition learning not only saves a significant large amount of training costs but also explains the network performance with the leading modes, displaying a striking piecewise power-law behavior. The modes specify a progressively compact latent space across the network hierarchy, making a more disentangled subspace compared to standard training. Our mode decomposition learning is also studied in an analytic online learning setting, which reveals multiple stages of learning dynamics with a continuous specialization of hidden nodes. Therefore the proposed mode decomposition learning points to a cheap and interpretable route towards the magical deep learning.

DOI: [10.1103/PhysRevResearch.5.L022011](https://doi.org/10.1103/PhysRevResearch.5.L022011)

Introduction. Deep neural networks are dominant tools with a broad range of applications, not only in image and language processing but also scientific research [1,2]. These networks are parameterized by a huge amount of trainable weight matrices, thereby consuming expensive training costs. However, these weight matrices are hard to interpret, and thus mechanisms underlying the macroscopic performance of the networks remain a big mystery in theoretical studies of neural networks [3,4].

To save the computational cost, previous studies of deep networks applied singular value decomposition to the weight matrices [5–8]. This decomposition requires the orthogonality condition for the singular vectors and positive singular values. The training also involves a carefully-designed structure for the trainable decomposition scheme [7,8]. These constraints and designs make the training process complicated, and thus a concise physics interpretation is still lacking. In addition, previous studies of recurrent memory networks showed that the network weight can be decomposed into separate random orthogonal patterns with corresponding importance scores [9,10]. Inspired by these studies, we conjecture that the learning in deep networks is shaped by a hierarchy of latent modes, which are not necessarily orthogonal, and the weight matrix can be expressed by these modes.

The mode decomposition learning (MDL) leads to a progressively compact latent mode space across the network

hierarchy, and meanwhile, the subspaces corresponding to different types of input are strongly disentangled, facilitating discrimination. The least number of latent modes achieving comparable performance with the costly standard methods grows only logarithmically with the network width and could even be a constant, thereby significantly reducing the training cost. The mode spectrum exhibits an intriguing piecewise power-law behavior. In particular, these properties do not depend on details of the training setting. Therefore our proposed MDL calls for a rethinking of conventional weight-based deep learning through the lens of cheap and interpretable mode-based learning.

Model. To show the effectiveness of the MDL scheme, we train a deep network to implement a classification task of handwritten digits [11]. The deep network has L layers ($L - 2$ hidden layers) with N_l neurons in the l th layer. The weight value of the connection from neuron i at the upstream layer l to neuron j at the downstream layer $l + 1$ is specified by w_{ij}^l . The activation of the neuron j at the downstream layer $h_j^{l+1} = f(z_j^{l+1}) = \max(0, z_j^{l+1})$, where the preactivation $z_j^{l+1} = \sum_i w_{ij}^l h_i^l$. For the output layer, the softmax function, $h_k = e^{z_k} / \sum_i e^{z_i}$, is chosen to specify the probability over all classes of the input images. The cross entropy $\mathcal{C} = -\sum_i \hat{h}_i \ln h_i$ is used as the cost function for the supervised learning, and \hat{h}_i is the target label (one-hot form). After training (the cross entropy is repeatedly averaged over minibatches of training examples), we evaluate the generalization performance of the network on an unseen test dataset.

Single weight values are not interpretable. According to our hypothesis, latent patterns would emerge from training in each layer. We call these patterns hierarchical modes for deep learning. Therefore the relationship between the modes and weight values is expressed by the following mode

*huanghp7@mail.sysu.edu.cn

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

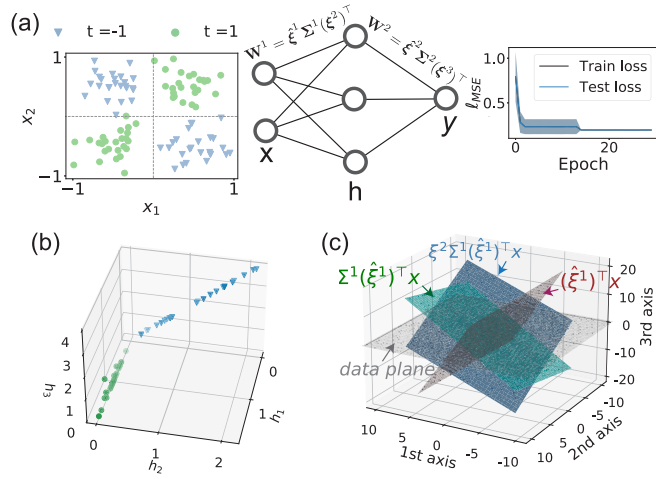


FIG. 1. A simple illustration of the mode decomposition learning. (a) A deep neural network of three layers, including one hidden layer with three hidden nodes, for a classification task of nonlinearly separable data. The weight matrix $w_{ij}^l = \sum_{\alpha=1}^p \hat{\xi}_{i,\alpha}^l \Sigma_{\alpha}^l \xi_{j,\alpha}^l$, where $p = 3$. The distribution of input data is modeled as a Gaussian mixture (see the main text) from which samples are assigned to labels $t = \pm 1$ based on the corresponding mixture component. The training performance is measured by the mean-squared-error loss function $\ell_{\text{MSE}}(y, t) = \|y - t\|^2/2$. (b) The representation of hidden neurons \mathbf{h} plotted in the 3D space, displaying the geometric separation. (c) The successive mappings from input sample \mathbf{x} (gray) to $(\hat{\xi}^1)^T \mathbf{x}$ (dark red), followed by $\Sigma^1 (\hat{\xi}^1)^T \mathbf{x}$ (green), and finally, $\xi^2 \Sigma^1 (\hat{\xi}^1)^T \mathbf{x}$ (blue).

decomposition:

$$\mathbf{w}^l = \hat{\xi}^l \Sigma^l (\xi^{l+1})^T, \quad (1)$$

where there are p^l upstream modes $\hat{\xi}^l \in \mathbb{R}^{N_l \times p^l}$ and the same number of downstream modes $\xi^{l+1} \in \mathbb{R}^{N_{l+1} \times p^l}$. The importance of each pair of adjacent modes is specified by the diagonal of the importance matrix $\Sigma^l \in \mathbb{R}^{p^l \times p^l}$, which is a diagonal matrix here. These modes may not be orthogonal with each other, and the importance score can take a real value. This setting allows for more degrees of freedom for learning features of input-output mappings. We will detail their geometric and physical interpretations below.

A geometric interpretation of Eq. (1) in a simple learning task is shown in Fig. 1. We use a three-layer network with three hidden neurons. The input data is sampled from a four-component Gaussian mixture [12],

$$\mathbb{P}(\mathbf{x}, t) = P(t) \sum_{\pm} P_{\pm} \mathcal{N}(\mathbf{x} | \mu_x^{t,\pm}, \Sigma_x^{t,\pm}), \quad (2)$$

where $\mathcal{N}(\mathbf{x} | \mu_x^{t,\pm}, \Sigma_x^{t,\pm})$ denotes a Gaussian distribution with mean $\mu_x^{t,\pm}$ and covariances $\Sigma_x^{t,\pm}$, and $P(t) = P_{\pm} = \frac{1}{2}$. For the label $t = +1$, $\mu_x^{t=+1,\pm} = \pm(0.5, 0.5)^T$, while for $t = -1$, $\mu_x^{t=-1,\pm} = \pm(-0.5, 0.5)^T$. Covariances are isotropic throughout with $\Sigma_x^{t,\pm} = 0.05\mathbf{1}$. The input samples $\mathbf{x} \in \mathbb{R}^2$ are first projected to the input pattern space spanned by $(\hat{\xi}^1)_i^T$ ($i = 1, 2, 3$). Then all three directions of this projection get expanded or contracted via $\Sigma^1 (\hat{\xi}^1)^T \mathbf{x}$. Finally the geometrically modified representation is remapped to the downstream representation space of a higher dimensionality, as $\xi^2 \Sigma^1 (\hat{\xi}^1)^T \mathbf{x}$

[Fig. 1(c)]. The nonlinearity of the transfer function is then applied to the last linear transformation, leading to the geometric separation [Fig. 1(b)]. We conclude that the MDL provides rich angles to look at the geometric transformation of the input information along the hierarchy of deep networks.

Rather than the conventional weight values in standard backpropagation (BP) algorithms [1], the trainable parameters are latent patterns in the MDL. The training is implemented by stochastic gradient descent in the mode space $\theta^l = (\hat{\xi}^l, \Sigma^l, \xi^{l+1})$ [13],

$$\begin{aligned} \Delta \xi_{j\alpha}^{l+1} &\equiv -\eta \frac{\partial \mathcal{L}}{\partial \xi_{j\alpha}^{l+1}} = -\eta \mathcal{K}_j^{l+1} \Sigma_{\alpha}^l \sum_i \hat{\xi}_{i\alpha}^l h_i^l, \\ \Delta \Sigma_{\alpha}^l &\equiv -\eta \frac{\partial \mathcal{L}}{\partial \Sigma_{\alpha}^l} = -\eta \sum_j \mathcal{K}_j^{l+1} \xi_{j\alpha}^{l+1} \sum_i \hat{\xi}_{i\alpha}^l h_i^l, \\ \Delta \hat{\xi}_{i\alpha}^l &\equiv -\eta \frac{\partial \mathcal{L}}{\partial \hat{\xi}_{i\alpha}^l} = -\eta \Sigma_{\alpha}^l h_i^l \sum_j \mathcal{K}_j^{l+1} \xi_{j\alpha}^{l+1}, \end{aligned} \quad (3)$$

where \mathcal{L} denotes the cost function (e.g., cross-entropy or mean-squared error) over a minibatch of training data, η denotes the learning rate, and $\mathcal{K}_j^{l+1} \equiv \partial \mathcal{L} / \partial z_j^{l+1}$ denotes the error term, which could backpropagate from the top layer where $\mathcal{K}_j^l = -\dot{h}_j^l (1 - h_j^l)$ for $\mathcal{L} = \mathcal{C}$ (cross entropy). Based on the chain rule, the error backpropagation equation can be derived as $\mathcal{K}_i^l = \sum_j \mathcal{K}_j^{l+1} \sum_{\alpha} \xi_{j\alpha}^{l+1} \Sigma_{\alpha}^l \hat{\xi}_{i\alpha}^l f'(z_i^l)$ [13]. To ensure the preactivation is independent of the upstream-layer width, we take the initialization scheme that $[\xi^{l+1} \Sigma^l (\hat{\xi}^l)^T]_{ij} \sim \mathcal{O}(\frac{1}{\sqrt{N_i}})$ [9]. To avoid the ambiguity of choosing patterns (e.g., scaled by a factor), we impose an identical regularization with strength 10^{-4} for all trainable parameters. However, our result does not change qualitatively with the specific values of regularization [13].

We remark that for each hidden layer there exist two types of pattern ($\xi^l \neq \hat{\xi}^l$). Equation (3) is used to learn these patterns. We call this case 1L2P. If we assume $\xi^l = \hat{\xi}^l$, the training can be further simplified as in [13], and we call this case 1L1P. The nature of this mode-based computation can be understood as an expanded linear-nonlinear layered computation, as $f(z_j^{l+1}) = f(\sum_{\alpha} c_{\alpha j} \kappa_{\alpha})$, where the linear field $\kappa_{\alpha} = \sum_i \hat{\xi}_{i\alpha}^l h_i^l$ and the equivalent weight $c_{\alpha j} = \xi_{j\alpha}^{l+1} \Sigma_{\alpha}^l$. Therefore the number of modes acts as the linear-layer width. We leave a systematic exploration of this linear-nonlinear structure by statistical mechanics in forthcoming works.

Online learning dynamics in a shallow network. The MDL can be analytically understood in an online learning setting, where we consider a one-hidden-layer architecture. The online learning can be considered as a special case of the above minibatch learning (i.e., the batch size is set to 1, and the sample is visited by the learning only once). The training dataset consists of n pairs $\{\mathbf{x}^v, y^v\}_{v=1}^n$. Each training example is independently sampled from a probability distribution $\mathbb{P}(\mathbf{x}, y) = \mathbb{P}(y|\mathbf{x})\mathbb{P}(\mathbf{x})$, where $\mathbb{P}(\mathbf{x})$ is a standard Gaussian distribution, and the scalar label y^v is generated by the neural network of k hidden neurons, (i.e., teacher, indicated by the symbol * below). Given an input $\mathbf{x}^v \in \mathbb{R}^d$, the corresponding

label is created by

$$y^v = \frac{1}{k} \sum_{r=1}^k \sigma \left(\frac{[\xi^* \Sigma^* (\hat{\xi}^*)^T]_r \mathbf{x}^v}{\sqrt{d}} \right) = \frac{1}{k} \sum_{r=1}^k \sigma(\lambda_r^{*v}), \quad (4)$$

where $[\xi^* \Sigma^* (\hat{\xi}^*)^T]_r$ denotes the r th row of the matrix $\xi^* \Sigma^* (\hat{\xi}^*)^T$, and $\lambda_r^{*v} = [\xi^* \Sigma^* (\hat{\xi}^*)^T]_r \mathbf{x}^v / \sqrt{d}$ represents the r th element of the teacher local field vector $\lambda^{*v} \in \mathbb{R}^k$. The teacher network is quenched as $[\xi^* \Sigma^* (\hat{\xi}^*)^T]_{ij} \sim \mathcal{O}(1)$. Here we focus on the nonlinear transfer function $\sigma(x) = \text{erf}(x/\sqrt{2})$. In addition, we train the other shallow network, called the student network, by minimizing the loss function $\mathcal{L}(y, \hat{f}(\mathbf{x}, \Theta))$ over the training data (labels are given by the teacher network), where Θ denotes the trainable parameters. The student's prediction for a fresh sample \mathbf{x} is given by

$$\hat{f}(\mathbf{x}, \hat{\xi}, \Sigma, \xi) = \frac{1}{m} \sum_{r=1}^m \sigma \left(\frac{[\xi \Sigma (\hat{\xi})^T]_r \mathbf{x}}{\sqrt{d}} \right) = \frac{1}{m} \sum_{r=1}^m \sigma(\lambda_r), \quad (5)$$

where λ_r denotes the r th component of the student local field $\lambda = \xi \Sigma (\hat{\xi})^T \mathbf{x}$, and the student has m hidden neurons. The student is supplied with data samples in sequence (one sample each time step). We next use v to indicate the time step as well.

The mean-squared error can be evaluated as

$$\ell_{\text{MSE}}(\Omega) = \frac{1}{2} \mathbb{E}_{\lambda, \lambda^* \sim \mathcal{N}(\lambda, \lambda^* | 0, \Omega)} [(\hat{f}(\lambda) - f(\lambda^*))^2], \quad (6)$$

where $f(\cdot)$ indicates the teacher's output, and we have replaced the expectation $\mathbb{E}_{\mathbf{x}, y \sim \mathbb{P}(\mathbf{x}, y)}[\cdot]$ by $\mathbb{E}_{\lambda, \lambda^* \sim \mathcal{N}(\lambda, \lambda^* | 0, \Omega)}[\cdot]$, because of the central-limit theorem and the i.i.d. setting we consider [14–16]. The covariance of the local field $\Omega^v \in \mathbb{R}^{(k+m) \times (k+m)}$ can be specified as follows,

$$\Omega^v \equiv \begin{bmatrix} \mathbf{Q}^v & \mathbf{M}^v \\ (\mathbf{M}^v)^T & \mathbf{P} \end{bmatrix}, \quad (7)$$

where $\mathbf{Q}^v \equiv \mathbb{E}_{\mathbf{x}, y \sim \mathbb{P}(\mathbf{x}, y)}[\lambda^v (\lambda^v)^T]$, $\mathbf{M}^v \equiv \mathbb{E}_{\mathbf{x}, y \sim \mathbb{P}(\mathbf{x}, y)}[\lambda^v (\lambda^{*v})^T]$, and $\mathbf{P} \equiv \mathbb{E}_{\mathbf{x}, y \sim \mathbb{P}(\mathbf{x}, y)}[\lambda^{*v} (\lambda^{*v})^T]$. By definition, \mathbf{P} is fixed, while \mathbf{Q}^v and \mathbf{M}^v evolve according to the gradient updates following a set of deterministic ordinary differential equations (ODEs) as the input dimension $d \rightarrow \infty$ [13]. These matrices are exactly the order parameters in physics. For simplicity, we consider $\xi = \xi^*$ and $\Sigma = \Sigma^*$, i.e., only the upstream patterns are learned.

Results. MDL can reach a similar test accuracy with that of BP performed in the weight space when p is sufficiently large [Fig. 2(a)]. The computational cost of the BP scales with N_l^2 . In contrast, MDL works in the mode space, requiring a training cost of only the order of pN_l . Note that p is much smaller than N_l (or $\lim_{N_l \rightarrow \infty} p^l/N_l = 0$), and our MDL does not need any additional training constraints (compared to other matrix factorization algorithms [13]). Remarkably, when $p = 30$ the performance of MDL almost matches that of BP [Fig. 2(b)], but it only utilizes 40% of the full sets of parameters that are consumed by the BP. In fact, each hidden layer can have two different types of latent pattern (1L2P) due to the mode decomposition. But if we assume that $\xi^l = \hat{\xi}^l$, i.e., each layer shares a single type of pattern (1L1P), we can further reduce the computational cost by an amount of $\sum_l p^l N_l$, without sacrificing the test accuracy [Fig. 2(b)]. Varying the network

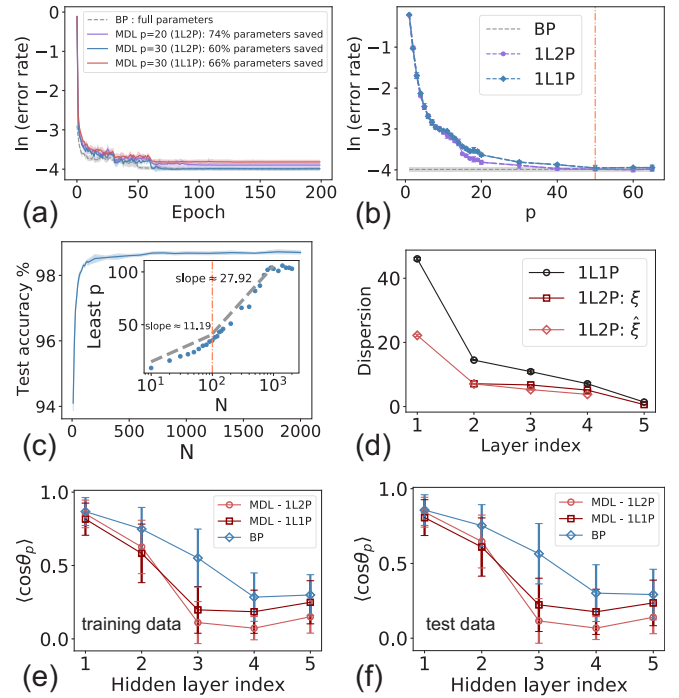


FIG. 2. Test performance and mode hierarchy of MDL in deep neural networks. Error rate is defined as the fraction of misclassified examples. (a) Training trajectories of a four-layer network, indicated by 784-100-100-10, where each number indicates the corresponding layer width. The number of modes $p^l = p$ for layer l , where $l = 1, \dots, L$. $p = 20$, or $p = 30$. Networks are trained on the full MNIST dataset (6×10^4 images) and tested on an unseen dataset containing 10^4 images. The fluctuation is computed over five independent runs. (b) Testing accuracy vs p (the number of modes is the same for all layers). The same architecture as (a) is used. The error bar characterizes the fluctuation across five independently trained networks, and each marker denotes the average result. The least number of modes is indicated by the dash-dot line. (c) The performance changes with the network width. The inset shows the least number of modes vs the layer width N (in the logarithmic scale). The network architecture is given by 784- N - N -10. The dash-dot line in the inset separates the piecewise logarithmic increase ($\propto \ln N$) regions. The result is obtained from five independent runs. (d) The averaged Euclidean distance (dispersion) from the pattern-cloud center ($\frac{1}{p} \sum_a \xi_a^l$) as a function of layer index. The network architecture is specified by 784-100-100-100-10 ($p = 30$). (e)–(f) Subspace overlap (principal angle) vs layer. The overlap is averaged with five independent runs, and seven-layer networks with hidden-layer width 100 are trained ($p = 30$).

width, we reveal a *logarithmic* increase of the least number of modes [Fig. 2(c)], which is a novel property of deep learning in the mode space, in stark contrast to a linear number of memory patterns in previous studies [9]. When the network width further grows, the least number can even become a constant. We argue that this manifests three separated phases of poor-good-saturated performance with increasing layer width (see Fig. S9 in [13]).

To see how the latent patterns are transformed in geometry along the network hierarchy, we first calculate the center of the pattern space; then the Euclidean distance from this center to each pattern is analyzed. We find that the pattern space

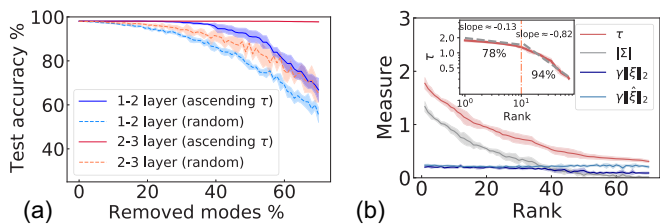


FIG. 3. The robustness properties of well-trained four-layer MDL models with the architecture 784-100-100-10. The case of 1L1P is considered with $p = 70$ in the hidden layers. (a) Effects of removing modes through two protocols: removing modes with weak measure τ first (solid line) and removing modes randomly (dashed line). The fluctuation is computed over ten independent runs. (b) The rescaled ℓ_2 norms $\gamma \|\hat{\xi}_\alpha\|_2$, $\gamma \|\hat{\xi}_\alpha\|_2$ and the absolute values of Σ vs their rank (in descending order) in the hidden layers, where $\gamma = \sum_\alpha |\Sigma_\alpha| / \sum_\alpha (\|\hat{\xi}_\alpha\|_2 + \|\hat{\xi}_\alpha\|_2)$. The inset shows a log-log plot of the τ measure, displaying a piecewise power-law behavior. The error bar is computed over five independent runs. The marked percentage indicates the generalization accuracy after removing the corresponding side of modes.

becomes progressively compact when going to deep layers [Fig. 2(d)]. To further characterize the geometric details, we define the subspace spanned by the principal eigenvectors of the layer neural responses to one type of inputs. Then the subspace overlap is calculated as the cosine of the principal angle between two subspaces corresponding to two types of inputs [13,17]. We find that the hidden-layer representation becomes more disentangled with layer in comparison with BP [Figs. 2(e) and 2(f)]. MDL shows great computational benefits of representation disentanglement, thereby facilitating discrimination. A slight increase of the overlap is observed for deeper layers, which is caused by the saturation of the test performance (see more analyses in [13]).

Compared to other matrix factorization methods, MDL has no additional constraints for the modes and importance scores, therefore being flexible for feature extraction. We find that the interlayer patterns (e.g., $\hat{\xi}^l$ vs $\hat{\xi}^{l+1}$) are more orthogonal than the intralayer (patterns belonging to each layer) ones. The geometric transformation carried out by these latent pattern matrices is not strictly a rotation for which the ℓ_2 norm is preserved. This flexibility is likely the key to make our method better than other matrix factorization methods in both training cost and learning performance (see details in [13]).

We next ask whether some modes are more important than the others. Therefore we rank the modes according to the measure $\tau_\alpha = \gamma \|\hat{\xi}_\alpha\|_2 + \gamma \|\hat{\xi}_\alpha\|_2 + |\Sigma_\alpha|$, where $\gamma = \sum_\alpha |\Sigma_\alpha| / \sum_\alpha (\|\hat{\xi}_\alpha\|_2 + \|\hat{\xi}_\alpha\|_2)$ to make comparable the magnitudes of the pattern and importance (Σ) score. Removing modes with weak values of τ first yields much higher accuracy than the random removal protocol [Fig. 3(a)], suggesting the existence of leading modes. Moreover, deeper layers are more robust. Figure 3(b) shows the measure as a function of rank in descending order, which can be approximately captured by piecewise power-law behavior (a transition point at the rank 10). Ranking with only the importance scores yields similar behavior [13]. A small exponent is observed for the leading measures, while the remaining measures bear a large

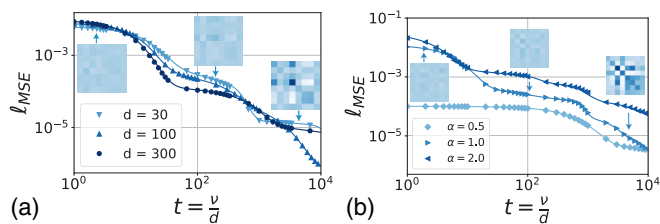


FIG. 4. Mean-squared error dynamics in terms of $t = \frac{\nu}{d}$, where ν denotes the online sample index, and d is the input dimension. The teacher and student networks share the same number of hidden neurons ($m = k = 8$). Markers represent results of the simulation, while the solid lines denote the theoretical predictions from solving the mean-field ODEs. The number of modes $p^* = p = \alpha \ln d$ (α denotes the mode load here). (a) Fixed $\alpha = 1$. (b) Fixed $d = 100$. The color deepens as α or d increases. The insets display the evolving \mathbf{M} matrix for $d = 30$ and $\alpha = 1.0$, respectively.

exponent, thereby revealing the coding hierarchy of latent modes in the deep networks. This intriguing behavior does not change with the regularization strength or the hidden-layer width [13].

Finally, the online mean-squared error dynamics of our model can be predicted perfectly in a teacher-student setting. The number of modes strongly affects the shape of the learning dynamics, and a large mode load can make the plateau disappear (Fig. 4). Moreover, during learning, the alignment between receptive fields of the student’s hidden nodes and the teacher’s modes continuously emerges, which is called the specialization transition [16,18].

Conclusion. In this Letter we propose a mode decomposition learning that works in the mode space rather than the conventional weight space. This learning scheme has three-fold technical and conceptual advances. First, the learning can achieve the comparable performance with standard methods, with a significant reduction of training costs. We also find that the least number of modes grows only logarithmically with the network width and becomes even independent of larger width, which is in stark contrast to a linear number of patterns in recurrent memory networks. Second, the learning leads to progressively compact pattern spaces, which promotes highly disentangled hierarchical representations. The upstream pattern maps the activity into a low-dimensional space, and then the resulting embedding is further expanded or contracted. After that, the modified embedding is remapped into the high-dimensional activity space. This sequence of geometric transformation can be understood as a linear-nonlinear hidden structure. Third, all modes are not equally important to the generalization ability of the network, showing an intriguing piecewise power-law behavior. Finally, the mode learning dynamics can be predicted by the mean-field ODEs, revealing the mode specialization transition. Therefore the MDL inspires a rethinking of conventional deep learning, offering a faster, more interpretable training framework. Future works along this direction will be inspired. For example, the impact of other structured dataset, transformer, or convolutional network structures, mode dynamics in overparameterized or recurrent networks, and the origin of adversarial vulnerability of deep networks in terms of geometry of the mode space.

Acknowledgments. This research was supported by the National Natural Science Foundation of China for Grant No. 12122515, Guangdong Provincial Key Labora-

tory of Magnetoelectric Physics and Devices (Grant No. 2022B1212010008), and Guangdong Basic and Applied Basic Research Foundation (Grant No. 2023B1515040023).

-
- [1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016).
- [2] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, *Rev. Mod. Phys.* **91**, 045002 (2019).
- [3] H. Huang, *Statistical Mechanics of Neural Networks* (Springer, Singapore, 2022).
- [4] D. A. Roberts, S. Yaida, and B. Hanin, *The Principles of Deep Learning Theory: An Effective Theory Approach to Understanding Neural Networks* (Cambridge University Press, Cambridge, 2022).
- [5] M. Jaderberg, A. Vedaldi, and A. Zisserman, Speeding up convolutional neural networks with low rank expansions, [arXiv:1405.3866](https://arxiv.org/abs/1405.3866).
- [6] H. Yang, M. Tang, W. Wen, F. Yan, D. Hu, A. Li, H. Li, and Y. Chen, Learning low-rank deep neural networks via singular vector orthogonality regularization and singular value sparsification, in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (IEEE, New York, 2020), pp. 2899–2908.
- [7] L. Giambagli, L. Buffoni, T. Carletti, W. Nocentini, and D. Fanelli, Machine learning in spectral domain, *Nat. Commun.* **12**, 1330 (2021).
- [8] L. Chicchi, L. Giambagli, L. Buffoni, T. Carletti, M. Ciavarella, and D. Fanelli, Training of sparse and dense deep neural networks: Fewer parameters, same performance, *Phys. Rev. E* **104**, 054312 (2021).
- [9] Z. Jiang, J. Zhou, T. Hou, K. Y. M. Wong, and H. Huang, Associative memory model with arbitrary Hebbian length, *Phys. Rev. E* **104**, 064306 (2021).
- [10] J. Zhou, Z. Jiang, T. Hou, Z. Chen, K. Y. M. Wong, and H. Huang, Eigenvalue spectrum of neural networks with arbitrary Hebbian length, *Phys. Rev. E* **104**, 064307 (2021).
- [11] Y. LeCun, The MNIST database of handwritten digits, retrieved from <http://yann.lecun.com/exdb/mnist>.
- [12] K. Fischer, A. René, C. Keup, M. Layer, D. Dahmen, and M. Helias, Decomposing neural networks as mappings of correlation functions, *Phys. Rev. Res.* **4**, 043143 (2022).
- [13] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevResearch.5.L022011> for technical and experimental details. Codes are available at <https://github.com/Chan-Li/MDL-model>.
- [14] M. Biehl and H. Schwarze, Learning by on-line gradient descent, *J. Phys. A: Math. Gen.* **28**, 643 (1995).
- [15] D. Saad and S. A. Solla, Exact Solution for On-Line Learning in Multilayer Neural Networks, *Phys. Rev. Lett.* **74**, 4337 (1995).
- [16] S. Goldt, M. S. Advani, A. M. Saxe, F. Krzakala, and L. Zdeborová, Generalisation dynamics of online learning in over-parameterised neural networks, [arXiv:1901.09085](https://arxiv.org/abs/1901.09085).
- [17] A. Björck and G. H. Golub, Numerical methods for computing angles between linear subspaces, *Math. Comput.* **27**, 579 (1973).
- [18] H. Schwarze, Learning a rule in a multilayer neural network, *J. Phys. A: Math. Gen.* **26**, 5781 (1993).