Sublinearly Structured Deep Neural Networks Achieve Feature Learning Consistency for Compositional Functions

Abstract

Over the past decade, deep neural networks (DNNs) have achieved remarkable success on complex machine-learning tasks, yet the theoretical foundations of their performance remain incomplete. From a statistical viewpoint, a natural question is: *can DNNs attain feature-learning and prediction consistency comparable to that of classical models?* While a full characterization is open, we provide positive results for a broad subclass. We establish consistency guarantees for *sublinearly structured DNNs*—architectures whose input/output dimensions and number of hidden neurons grow sublinearly with the sample size—when learning features of hierarchically compositional target functions. Importantly, the consistency still holds even in the conventional "over-parameterized" regime where the total number of parameters exceeds the number of training samples. Empirically, sublinearly structured DNNs match or surpass wide DNNs in prediction. A structural audit further indicates that widely used convolutional neural networks (CNNs), including AlexNet, VGGNet, ResNet, GoogLeNet, are sublinearly structured on their image classification benchmarks. We further prove that the sublinearly structured DNNs achieve universal approximation for hierarchically compositional functions in the large-sample limit. Moreover, images exhibit an inherent hierarchical, compositional structure. Taken together, these results explain, through a statistical lens, why many large-scale deep learning models succeed after adequate training on massive image datasets.

Keywords: Compositional Function, Eigen Analysis, Feature Learning Consistency, Over Parameterization, Stochastic Neural Network

Mathematics Subject Classification (2020): 62M45, 62F12

1 Introduction

Over the past decade, DNNs have made major breakthroughs in many research domains, including image generation, protein folding, and language processing. The ability of these models to automatically learn the problem-specific features hidden in the training data is considered as a major factor contributing to their success, see e.g. Radhakrishnan et al. (2024), Shi et al. (2022), and Yang and Hu (2021). Therefore, understanding the mechanism of feature learning and, by extension, designing the network structure for ensuring the hidden features to be effectively learned has attracted much attention in recent literature.

Feature learning for linear models has been well studied in statistics, see e.g., Tibshirani (1996b), Fan and Li (2001), and Loh and Wainwright (2017), where one aims to identify important covariates through estimating their regression coefficients. For DNNs, we follow Radhakrishnan

et al. (2024) to define a neural feature as an eigenvector of $\boldsymbol{w}_l^T \boldsymbol{w}_l$, where $\boldsymbol{w}_l \in \mathbb{R}^{d_l \times d_{l-1}}$ denotes the weight matrix of hidden layer l, and d_l denotes the width of layer l. It is easy to see that the regression coefficient vector of the linear model can be viewed as a special case of this general definition (with $d_1 = 1$ and $d_0 = p$ for the number of covariates, and rescaled by $\boldsymbol{w}_1 \boldsymbol{w}_1^T$). Thus, if a method produces a consistent estimate of $\boldsymbol{w}_l$, it can also be said to be "feature learning consistent". As discussed later, since $\boldsymbol{w}_l$ is only identifiable up to certain loss-invariant transformations, the consistency of neural feature estimates serves as a reliable measure for the consistency of $\boldsymbol{w}_l$'s estimates. From a statistical perspective, a natural question is whether a DNN can achieve feature learning consistency similar to that of linear models.

To bridge the gap between linear models and deep learning, Sun and Liang (2022) and Liang et al. (2022) proposed a new type of neural network – stochastic neural network (abbreviated as StoNet). This network is formulated as a composition of many linear/logistic regressions and provides a framework for transferring theory and methods from linear models to DNNs. Additionally, the StoNet offers a convenient way for addressing many important problems encountered in modern data science, such as sufficient dimension reduction (Liang et al., 2022), uncertainty quantification (Sun and Liang, 2025), and causal effect estimation (Fang and Liang, 2024). These problems are otherwise hard to address using conventional DNNs. In this paper, based on the asymptotic equivalence between DNN and StoNet (Liang et al., 2022), we prove that the sublinearly structured DNN achieves consistency in parameter estimation (up to loss-invariant transformations) and, subsequently, consistency in feature learning for hierarchically compositional functions in which each constituent depends on at most a bounded number of variables.

Hierarchical compositional functions are multivariate maps built as compositions of low-arity modules arranged in a tree or directed acyclic graph (DAG). They also include conventional functions with a fixed input dimension as special cases. This structure is widespread in science and engineering (e.g., PDE operators, image processing pipelines) and captures rich multiscale interactions while keeping each module low-dimensional — precisely the regime where DNNs admit a sparse structure and avoid the curse of dimensionality.

We refer a DNN as *a sublinearly structured DNN* (or sublinear DNN for short) if its structure satisfies the constraints:

$$d_0 \prec n, \qquad d_{h+1} \prec n, \qquad \sum_{l=1}^{h} d_l \prec n, \tag{1}$$

where $d_0 = p$ denotes the input dimension, d_l denotes the width of layer l, h denotes the number of hidden layers, d_{h+1} denotes the output dimension, and n denotes the training sample size. Here, we denote $a_n \prec b_n$ if $\lim_{n\to\infty} \frac{a_n}{b_n} = 0$. In equation (1), the dependence of d_i's on n is implicit. This definition of sublinear DNN applies to fully connected neural networks and may require slight modification for convolutional neural networks (CNNs), where the filter size used at each convolutional layer must be considered in defining the corresponding StoNet (see Section 5). It is worth noting that the number of parameters in a sublinear DNN can still greatly exceed n; in other words, sublinear DNNs may be over-parameterized.

The parameter estimation consistency provides a theoretical guarantee for the asymptotic recovery of the underlying true system by sublinear DNNs. This, in turn, ensures consistent

predictions for future values. Additionally, we prove that sublinear DNNs achieve universal approximation for hierarchically compositional functions in the large-sample limit. We analyze the structures of many popular large-scale DNNs, such as AlexNet, VGGNet, ResNet, and GoogLeNet, used in image classification, and find that they are all sublinear on their image classification benchmarks. Furthermore, images exhibit an inherent hierarchical, compositional structure. Taken together, our results explain, through a statistical lens, why many large-scale DNNs succeed after adequate training on massive image datasets.

To our knowledge, this work represents the first theoretical result on feature learning consistency for DNNs in the over-parameterization regime, although restricted to the class of hierarchically compositional functions. This marks a notable distinction from existing studies, where consistency results have been established under the assumption that the total number of parameters or effective parameters is of lower order than n (Farrell et al., 2021; Sun et al., 2022). Our theory delineates a regime for deep learning in which the existence of an "optimal solution" is theoretically guaranteed. Our numerical experiments show that sublinear DNNs can achieve prediction accuracy comparable to, or even exceeding, that of wide DNNs for hierarchically compositional functions.

Related Works Motivated by the observation of benign overfitting (Bartlett et al., 2020), a line of work has emerged studying the properties of wide DNNs, see e.g., Yang and Hu (2021) and Woodworth et al. (2020). These studies typically rely on two key assumptions: (i) the wide DNNs are trained using gradient-descent-type methods, and (ii) the scale of initialization is appropriately chosen. For instance, Yang and Hu (2021) noted that the standard initialization of neural networks do not admit infinite-width limits that can learn features and proposed a modification to enable feature learning in the limit. Woodworth et al. (2020) showed how the scale of the initialization controls the transition between the kernel and feature learning regimes. Although our theory is restricted to the class of hierarchically compositional functions, it does not depend on the specific optimization algorithm used or the initialization scale adopted. On the restricted domain, our theory can be seen as complementary to those on wide DNNs, thereby providing a complete spectrum of theoretical insights on DNNs from narrow to wide.

Another line of related work investigates the intrinsic dimensionality of large-scale DNNs (see, e.g., Li et al., 2018; Aghajanyan et al., 2020). These studies show that such networks often have a very low intrinsic dimension that changes little as width or depth increase. Building on this observation, several low-rank adaptation methods have been proposed for fine-tuning large language models, such as LoRA (Hu et al., 2022) and QLoRA (Dettmers et al., 2023). This, in turn, suggests that large-scale DNNs admit effective low-dimensional reparameterizations, reflecting the hierarchical, compositional structure of the underlying target function and making sublinear DNNs a viable and efficient option.

2 DNN and Its Stochastic Surrogate

Consider a DNN model with h hidden layers defined as follows:

$$\begin{aligned} \tilde{\boldsymbol{Y}}_1 &= \boldsymbol{b}_1 + \boldsymbol{w}_1 \boldsymbol{X}, \\ \tilde{\boldsymbol{Y}}_i &= \boldsymbol{b}_i + \boldsymbol{w}_i \Psi(\tilde{\boldsymbol{Y}}_{i-1}), \quad i = 2, 3, \ldots, h, \\ \boldsymbol{Y} &= \boldsymbol{b}_{h+1} + \boldsymbol{w}_{h+1} \Psi(\tilde{\boldsymbol{Y}}_h) + \boldsymbol{e}_{h+1}, \end{aligned} \tag{2}$$

where $\boldsymbol{e}_{h+1} \sim N(0, \sigma^2_{h+1} I_{d_{h+1}})$ is Gaussian random error; $\boldsymbol{X} \in \mathbb{R}^{d_0}$; $\tilde{\boldsymbol{Y}}_i \in \mathbb{R}^{d_i}$ denotes the pre-activation of the i-th hidden layer; $\boldsymbol{w}_i \in \mathbb{R}^{d_i \times d_{i-1}}, \boldsymbol{b}_i \in \mathbb{R}^{d_i}$ denotes the weights and bias of the i-th layer; $\Psi(\cdot)$ is the element-wise activation function such that $\Psi(\tilde{\boldsymbol{Y}}_{i-1}) = (\Psi(\tilde{Y}_{i-1,1}), \Psi(\tilde{Y}_{i-1,2}), \ldots \Psi(\tilde{Y}_{i-1,d_{i-1}}))^T$. For simplicity, we consider only regression problems in (2). It can be easily extended to classification problems by replacing the third equation in (2) with a logit model.

The StoNet (Liang et al., 2022) is *a probabilistic deep learning model* defined by adding auxiliary noise to the pre-activation $\tilde{\boldsymbol{Y}}_i$'s in the DNN model (2):

$$\begin{aligned} \boldsymbol{Y}_1 &= \boldsymbol{b}_1 + \boldsymbol{w}_1 \boldsymbol{X} + \boldsymbol{e}_1, \\ \boldsymbol{Y}_i &= \boldsymbol{b}_i + \boldsymbol{w}_i \Psi(\boldsymbol{Y}_{i-1}) + \boldsymbol{e}_i, \quad i = 2, 3, \ldots, h, \\ \boldsymbol{Y} &= \boldsymbol{b}_{h+1} + \boldsymbol{w}_{h+1} \Psi(\boldsymbol{Y}_h) + \boldsymbol{e}_{h+1}, \end{aligned} \tag{3}$$

It can be viewed as a composition of many simple regressions, where $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_h$ are latent variables. For simplicity, we assume that $\boldsymbol{e}_i \sim N(0, \sigma_i^2 I_{d_i})$ for $i = 1, 2, \ldots, h, h+1$. Other distributions can also be assumed for $\boldsymbol{e}_i$'s. For instance, Sun and Liang (2022) assumed a modified double exponential distribution for $\boldsymbol{e}_1$ such that the first layer defines a series of support vector regressions (SVRs). The parameters $\{\sigma_1^2, \ldots, \sigma_h^2, \sigma_{h+1}^2\}$ control the variation of latent variables $\{\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_h\}$. For classification problems, σ_{h+1}^2 works as the temperature for the binomial or multinomial distribution formed at the output layer.

The property of the StoNet, as an approximator to the DNN, has been studied in Liang et al. (2022). A brief review for their theory is provided below, which forms the basis of this work. Let $\boldsymbol{\theta} = \{\boldsymbol{w}_1, \boldsymbol{b}_1, \ldots, \boldsymbol{w}_{h+1}, \boldsymbol{b}_{h+1}\}$ denote the collection of all parameters of the StoNet (3), let $\boldsymbol{\Theta}$ denote the space of $\boldsymbol{\theta}$, and let $\boldsymbol{Y}_{\text{mis}} = \{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_h\}$ denote the collection of all latent variables. Let $\pi(\boldsymbol{Y}, \boldsymbol{Y}_{\text{mis}} | \boldsymbol{X}, \boldsymbol{\theta})$ denote the joint density function of the pseudo-complete data $(\boldsymbol{Y}, \boldsymbol{Y}_{\text{mis}})$, and let $\pi_{\text{DNN}}(\boldsymbol{Y} | \boldsymbol{X}, \boldsymbol{\theta})$ denote the probability density function of the DNN model (2).

Regarding the network structure, activation function, and the variance of the latent variables, they made the following assumption:

Assumption 1. *(i) $\boldsymbol{\Theta}$ is compact, i.e., $\boldsymbol{\Theta}$ is contained in a d_θ-ball centered at 0 with radius r; (ii) $\mathbb{E}(\log \pi(\boldsymbol{Y}, \boldsymbol{Y}_{\text{mis}} | \boldsymbol{X}, \boldsymbol{\theta}))^2 < \infty$ for any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$; (iii) the activation function $\Psi(\cdot)$ is c-Lipschitz continuous for some constant c; (iv) the network's depth h and widths d_l's are both allowed to increase with n; (v) σ_{h+1} is a constant, and for every $k \in \{1, 2, \ldots, h\}$, $d_{h+1}(\prod_{i=k+1}^{h} d_i^2) d_k \sigma_k^2 \prec \frac{1}{h}$ and $d_k \sigma_k = o(1)$.*

Assumption 1-(i) & (ii) are regular and generally satisfied; Assumption 1-(iii) allows the StoNet to work with a wide range of Lipschitz continuous activation functions such as *tanh*,

sigmoid and *ReLU*; Assumption 1-(v) constrains the magnitude of noise added to each hidden neuron, where the factor $d_{h+1}(\prod_{i=k+1}^{h} d_i^2)d_k$ can be understood as the amplification factor of the noise $\boldsymbol{e}_k$ at the output layer. In general, the noise added to the first few hidden layers should be small to prevent large random errors propagated to the output layer. Under slightly weaker conditions than Assumption 1 (in particular, without requiring $d_k\sigma_k = o(1)$), they showed that StoNet (3) and the DNN (2) share the same asymptotic energy landscape, as stated in part (i) of Lemma 1. Imposing the slightly stronger scaling condition on σ_l further allows a refined analysis of the DNN parameter estimates (rather than focusing only on properties of the hidden neuron outputs).

Regarding the energy landscape of the DNN, they made Assumption 2. Let $Q^*(\boldsymbol{\theta}) = \mathbb{E}(\log \pi_{\text{DNN}}(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}))$ be the expected loss, taken with respect to the joint distribution $\pi(\boldsymbol{X}, \boldsymbol{Y})$. By Assumption 1-(i)&(ii) and the law of large numbers,

$$\frac{1}{n}\sum_{i=1}^{n} \log \pi_{\text{DNN}}(\boldsymbol{Y}^{(i)}|\boldsymbol{X}^{(i)}, \boldsymbol{\theta}) - Q^*(\boldsymbol{\theta}) \xrightarrow{p} 0 \tag{4}$$

holds uniformly over Θ, where the superscript i indexes observations of the dataset.

Assumption 2. *(i) The expected loss function $Q^*(\boldsymbol{\theta})$ is continuous in $\boldsymbol{\theta}$ and uniquely maximized (up to loss-invariant transformations) at $\boldsymbol{\theta}^*$; (ii) for any $\epsilon > 0$, $sup_{\boldsymbol{\theta}\in\Theta\setminus B(\epsilon)} Q^*(\boldsymbol{\theta})$ exists, where $B(\epsilon) = \{\boldsymbol{\theta} : \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| < \epsilon\}$, and $\delta = Q^*(\boldsymbol{\theta}^*) - sup_{\boldsymbol{\theta}\in\Theta\setminus B(\epsilon)} Q^*(\boldsymbol{\theta}) > 0$.*

Assumption 2 restricts the shape of $Q^*(\boldsymbol{\theta})$ around the global maximizer, ensuring that it is neither discontinuous nor too flat. Given nonidentifiability of the neural network model, Assumption 2 implicitly assumes that each $\boldsymbol{\theta}$ is unique up to loss-invariant transformations, such as reordering the hidden neurons within the same layer or simultaneously altering the signs or scales of certain weights and biases (see e.g., Liang et al. (2018a) and Sun et al. (2022) for further discussions). Alternatively, the optimal solutions can be considered as belonging to an equivalent class, subject to appropriate loss-invariant transformations, with the uniqueness assumption applying to this equivalent class. This nonidentifiability issue will be further discussed later, along with our demonstration of neural features learned with different neural networks. Under Assumptions 1 and 2, they proved part (ii) of Lemma 1.

Lemma 1 (Theorem 2.1 and 2.2 of Liang et al. (2022))**.** *Suppose Assumptions 1–2 hold, and $\pi(\boldsymbol{Y}, \boldsymbol{Y}_{\text{mis}}|\boldsymbol{X}, \boldsymbol{\theta})$ is continuous in $\boldsymbol{\theta}$. Then*

$$\begin{aligned} &(i) \quad \sup_{\boldsymbol{\theta}\in\Theta} \Big| \frac{1}{n}\sum_{i=1}^{n} \log \pi(\boldsymbol{Y}^{(i)}, \boldsymbol{Y}_{mis}^{(i)}|\boldsymbol{X}^{(i)}, \boldsymbol{\theta}) - \frac{1}{n}\sum_{i=1}^{n} \log \pi_{\text{DNN}}(\boldsymbol{Y}^{(i)}|\boldsymbol{X}^{(i)}, \boldsymbol{\theta}) \Big| \xrightarrow{p} 0, \\ &(ii) \quad \|\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*\| \xrightarrow{p} 0, \quad \text{as } n \to \infty, \end{aligned} \tag{5}$$

where $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\Theta} \mathbb{E}(\log \pi_{\text{DNN}}(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}))$ denotes the true parameters of the DNN model as specified in (2), and $\hat{\boldsymbol{\theta}}_n = \arg\max_{\boldsymbol{\theta}\in\Theta}\{\frac{1}{n}\sum_{i=1}^{n} \log \pi(\boldsymbol{Y}^{(i)}, \boldsymbol{Y}_{mis}^{(i)}|\boldsymbol{X}^{(i)}, \boldsymbol{\theta})\}$ denotes the maximum likelihood estimator of the StoNet model (3) with the pseudo-complete data.

Lemma 1 implies that the StoNet and DNN are asymptotically equivalent as n becomes large, which forms the basis for bridging many properties between the two types of neural networks. This asymptotic equivalence can be elaborated from two perspectives:

- Suppose the DNN model (2) is true, then it implies that when n becomes large, the parameters of the DNN can be learned by training a StoNet of the same structure with σ_i^2's satisfying Assumption 1-(v). Algorithmically, the StoNet can be viewed as a training algorithm of the DNN with latent variable augmentation when n is large.

- Suppose the StoNet (3) is true, then it implies that for any StoNet satisfying Assumptions 1-2, the parameters $\boldsymbol{\theta}$ can be learned by training a DNN of the same structure when n is large.

As shown later, this asymptotic equivalence leads to interesting results toward parameter estimation consistency for sublinear DNNs.

3 Feature Learning Consistency in sublinear DNNs

This section first describes the imputation-regularized optimization (IRO) algorithm (Liang et al., 2018b) for training sublinear StoNets, and then establishes parameter estimation consistency for them. This result, by Lemma 1, further implies parameter estimation consistency for the sublinear DNNs that are trained with an optimization algorithm such as stochastic gradient descent (SGD) or Adam (Kingma and Ba, 2015).

3.1 The IRO Algorithm

Notation: Let $D_n = \{\mathbb{Y}, \mathbb{X}\}$ denote a dataset of n observations, where $\mathbb{Y} \in \mathbb{R}^{n \times d_{h+1}}$ and $\mathbb{X} \in \mathbb{R}^{n \times p}$ contain the responses and covariates, respectively. Let $\boldsymbol{\sigma}_n^2 = (\sigma_1^2, \ldots, \sigma_{h+1}^2)$, where the dependence of σ_l on n is implicit as implied by Assumption 1.

Algorithm 1 IRO Algorithm for StoNet

Input: Dataset D_n, total iteration number T, and Monte Carlo step number t_{MC}.
Initialization: Randomly initialize the network parameters $\hat{\boldsymbol{\theta}}_n^{(0)}$.
for $t = 1$ **to** T **do**

- **Imputation step**: For each sample $(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)})$, draw $\boldsymbol{Y}_{\text{mis}}^{(i,t)}$ from $\pi(\boldsymbol{Y}_{\text{mis}}|\boldsymbol{Y}^{(i)}, \boldsymbol{X}^{(i)}, \hat{\boldsymbol{\theta}}_n^{(t-1)}, \boldsymbol{\sigma}_n^2)$ by running the SGLD or Metropolis-Hastings algorithm for t_{MC} steps.
- **Regularized optimization step**: Based on the pseudo-complete data $(\boldsymbol{Y}, \boldsymbol{Y}_{\text{mis}}^{(t)}, \boldsymbol{X})$, update $\hat{\boldsymbol{\theta}}_n^{(t-1)}$ by minimizing a penalized loss function, i.e., setting

$$\hat{\boldsymbol{\theta}}_n^{(t)} = \arg\min_{\boldsymbol{\theta}} \Big\{ -\frac{1}{n} \sum_{i=1}^{n} \log \pi(\boldsymbol{Y}^{(i)}, \boldsymbol{Y}_{\text{mis}}^{(i,t)} | \boldsymbol{X}^{(i)}, \boldsymbol{\theta}, \boldsymbol{\sigma}_n^2) + P_{\lambda_n}(\boldsymbol{\theta}) \Big\}, \tag{6}$$

where the penalty $P_{\lambda_n}(\boldsymbol{\theta})$ is chosen such that $\hat{\boldsymbol{\theta}}_n^{(t)}$ forms a consistent estimator of

$$\begin{aligned} \boldsymbol{\theta}_*^{(t)} &= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\theta}_n^{(t-1)}} \log \pi(\boldsymbol{Y}, \boldsymbol{Y}_{\text{mis}} | \boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{\sigma}_n^2) \\ &= \arg\max_{\boldsymbol{\theta}} \int \log \pi(\boldsymbol{Y}_{\text{mis}}, \boldsymbol{Y} | \boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{\sigma}_n^2) \pi(\boldsymbol{Y}_{\text{mis}} | \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{\theta}_n^{(t-1)}, \boldsymbol{\sigma}_n^2) \pi(\boldsymbol{Y} | \boldsymbol{X}, \boldsymbol{\theta}^*, \boldsymbol{\sigma}_n^2) d\boldsymbol{Y}_{\text{mis}} d\boldsymbol{Y}, \end{aligned}$$

where $\boldsymbol{\theta}_*^{(t)}$ is called the working true parameter at iteration t.

end for
Output: $\hat{\theta}_n^{(T)}$.

To facilitate theoretical development, we assume that for a given dataset D_n, the true model is a StoNet model with $\boldsymbol{\sigma}_n^2$ being known and satisfying Assumption 1-(v). By treating the latent variables $\boldsymbol{Y}_{\text{mis}}$ as missing, the IRO algorithm can be applied to train the StoNet. The key to the IRO algorithm is to find an estimator that is uniformly consistent for the working true parameter $\boldsymbol{\theta}_*^{(t)}$ over all iterations. For high-dimensional problems, as suggested by Liang et al. (2018b), such a uniformly consistent sparse estimator can typically be obtained by minimizing an appropriately penalized loss function as defined in (6). For a sublinear StoNet, such a penalty term is unnecessary. Therefore, we set $P_{\lambda_n}(\boldsymbol{\theta}) = 0$ for $\boldsymbol{\theta} \in \Theta$. Under this setting, solving (6) corresponds to solving a series of linear regressions by noting that the joint distribution $\pi(\boldsymbol{Y}_{\text{mis}}, \boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{\sigma}_n^2)$ can be decomposed in a Markovian structure:

$$\pi(\boldsymbol{Y}_{\text{mis}}, \boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{\sigma}_n^2) = \pi(\boldsymbol{Y}|\boldsymbol{Y}_h, \boldsymbol{\theta}, \boldsymbol{\sigma}_n^2)\pi(\boldsymbol{Y}_h|\boldsymbol{Y}_{h-1}, \boldsymbol{\theta}, \boldsymbol{\sigma}_n^2)\cdots\pi(\boldsymbol{Y}_1|\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{\sigma}_n^2), \tag{7}$$

and, furthermore, the components of $\boldsymbol{Y}_i \in \mathbb{R}^{d_i}$ are mutually independent conditional on $\boldsymbol{Y}_{i-1}$ for $i = 1, 2, \ldots, h+1$. For notational simplicity, we let $\boldsymbol{Y}_0 = \boldsymbol{X}$ and $\boldsymbol{Y}_{h+1} = \boldsymbol{Y}$.

We note that the IRO algorithm is reduced to the stochastic EM algorithm (Celeux and Diebolt, 1985; Nielsen, 2000) when no penalty is used in (6). However, the theoretical framework established in Liang et al. (2018b) still works for the sublinear StoNet models. For this reason, Algorithm 1 is still referred to as an IRO algorithm.

3.2 Feature Learning Consistency

For all theoretical results in this paper, the proofs are presented in the Appendix. Let $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denote, respectively, the minimum and maximum eigenvalues of the matrix A. For the inputs and network structure, we make the following assumption:

Assumption 3. *(i) The network width satisfies the condition given in (1); (ii)* $\boldsymbol{X} \in [0,1]^p$ *(i.e., in a bounded space); additionally, there exists a constant* $\kappa_{\min} > 0$ *such that* $\lambda_{\min}(\boldsymbol{\Sigma}_0) \geq \kappa_{\min}$*, where* $\boldsymbol{\Sigma}_0$ *denotes the covariance matrix of* $\boldsymbol{X}$*.*

Assumption 3-(i) restricts the structure of the DNN, while its universal approximation ability can still be established for some classes of functions under the large sample regime, detailed in Section 3.3. Assumption 3-(ii) is regular, which implies that the eigenvalues of the covariance matrix of $\boldsymbol{X}$ are uniformly bounded across sample sizes, i.e., $\kappa_{\min} \leq \lambda_{\min}(\boldsymbol{\Sigma}_0) \leq \lambda_{\max}(\boldsymbol{\Sigma}_0) \leq \kappa_{\max}$ for some constant $\kappa_{\max} > 0$ and all sample size $n > 0$.

Lemma 2. *For any layer* $l \in \{1, 2, \ldots, h\}$*, let* $\boldsymbol{\Sigma}_l^{(t)} \in \mathbb{R}^{d_l \times d_l}$ *denote the covariance matrix of the covariates of the regressions, which are formed for the neurons of layer* $l+1$ *at iteration* t *of Algorithm 1. If Assumptions 1 and 3 hold, then there exist constants* $c > 0$*,* $c' > 0$*, and* $\tilde{\kappa} > 0$ *such that* $c\sigma_l^2 \leq \lambda_{\min}(\boldsymbol{\Sigma}_l^{(t)}) \leq \lambda_{\max}(\boldsymbol{\Sigma}_l^{(t)}) \leq \tilde{\kappa} + c'\sigma_l + c'\sigma_l^2$ *holds for any iteration* t*.*

The proof of Lemma 2 is given in Appendix A.2. To study properties of the coefficient estimators for the regressions formed in the StoNet, we introduce the following two lemmas, one for linear regression and the other for multinomial logistic regression.

Lemma 3. *(Rencher and Schaalje, 2007, Theorem 7.3; Golub and Loan, 2013, Chapter 2) Consider a linear regression with* n *observations:* $\mathbb{Y} = \mathbb{X}\boldsymbol{\beta} + \sigma\boldsymbol{\epsilon}$*, where* $\mathbb{Y} \in \mathbb{R}^n$*,* $\mathbb{X} \in \mathbb{R}^{n \times p}$*,* $\boldsymbol{\beta} \in \mathbb{R}^p$*,*

$\sigma > 0$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I_p)$, *and* $p < n$. *If* $\lambda_{\min}(\mathbb{X}^T\mathbb{X}) \geq n\kappa_{\min}$, *then* $E\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|^2 = \sigma^2\|\mathbb{X}^T\mathbb{X}\|^{-1} \leq \frac{\sigma^2}{n\kappa_{\min}}$, *where* $\hat{\boldsymbol{\beta}}$ *denotes the ordinary least square (OLS) estimator of* $\boldsymbol{\beta}$.

Lemma 4. *Consider a multinomial logistic regression, which contains* $m+1$ *classes and* n *observations. Assume that (i) each covariate in* $\mathbb{X} \in \mathbb{R}^{n\times p}$ *is normally distributed with the variance decreasing with* n *at a rate of* $O(n^{-\alpha})$ *for some* $\alpha > 0$*; (ii)* $p \leq n$*; and (iii) the eigenvalues of* $\mathbb{X}^T\mathbb{X}$ *are positive and bounded, i.e., there exist constants* $\kappa_{\min} > 0$ *and* $\kappa_{\max} > 0$ *such that* $n\kappa_{\min} \leq \lambda_i(\mathbb{X}^T\mathbb{X}) \leq n\kappa_{\max}$ *holds for any* $i \in \{1, 2, \ldots, p\}$. *Let* $\vec{\boldsymbol{B}} = (\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T, \cdots, \boldsymbol{\beta}_m^T)^T$ *denote the vector of true regression coefficients of the model, where* $\boldsymbol{\beta}_i \in \mathbb{R}^p$, *and let* $\widehat{\vec{\boldsymbol{B}}}$ *denote the maximum likelihood estimator (MLE) of* $\vec{\boldsymbol{B}}$. *Then there exists a constant* ν_0 *such that* $\mathbb{E}_{\mathbb{Y}|\mathbb{X}}\|\widehat{\vec{\boldsymbol{B}}} - \vec{\boldsymbol{B}}\|^2 \leq \frac{1}{n\nu_0\kappa_{\min}}$ *holds with probability converging to 1, where the expectation is taken with respect to the conditional distribution* $\pi(\mathbb{Y}|\mathbb{X})$.

The proof of Lemma 4 is given in Appendix A.3. The conditions of Lemma 4 are specifically tailored to the auxiliary StoNet used in the proof. In this StoNet, the output of each hidden neuron is modeled as a Gaussian random variable with variance chosen by the user. As such, the conditions, including the rate at which the variance decreases, can be satisfied. Specifically, condition (i) aligns with Assumption 1-(v), where the variance of the random noise added to each hidden neuron tends to decrease as n increases; condition (ii) aligns with Assumption 3-(i); and condition (iii) aligns with Lemma 2.

Lemma 5. *Suppose that Assumptions 1–3 hold, and* $\sum_{l=1}^{h+1} d_l\sigma_l^2/\sigma_{l-1}^2 \prec n$ *holds as well, where* $\sigma_0^2 = \kappa_{\min}$, $\sigma_{h+1}^2 = \sigma_*^2$ *for normal regression in Lemma 3, and* $\sigma_{h+1}^2 = 1/\nu_0$ *for multinomial logistic regression in Lemma 4. Then there exist a constant* c *such that* $E\|\hat{\boldsymbol{\theta}}_n^{(t)} - \boldsymbol{\theta}_*^{(t)}\|^2 \leq \frac{c}{n}\sum_{l=1}^{h+1} d_l\frac{\sigma_l^2}{\sigma_{l-1}^2} := r_n \prec o(1)$.

The proof of Lemma 5 is given in Appendix A.4. Further, let's consider the mapping $M(\boldsymbol{\theta})$ as defined in (6), i.e., the EM update

$$M(\boldsymbol{\theta}) = \arg\max_{\tilde{\boldsymbol{\theta}}} \mathbb{E}_{\boldsymbol{\theta}} \log \pi(\boldsymbol{Y}, \boldsymbol{Y}_{\text{mis}}|\boldsymbol{X}, \tilde{\boldsymbol{\theta}}).$$

As argued in Liang et al. (2018b) and Nielsen (2000), it is reasonable to assume this mapping is a contraction. A recursive application of the mapping, i.e., setting $\hat{\boldsymbol{\theta}}_n^{(t+1)} = \boldsymbol{\theta}_*^{(t+1)} = M(\hat{\boldsymbol{\theta}}_n^{(t)})$, leads to a monotone increase of the target expectations $\mathbb{E}_{\hat{\boldsymbol{\theta}}_n^{(t)}} \log \pi(\boldsymbol{Y}, \boldsymbol{Y}_{\text{mis}}|\boldsymbol{X}, \hat{\boldsymbol{\theta}}_n^{(t+1)})$ for $t = 1, 2, \ldots, T$.

Assumption 4. *The mapping* $M(\boldsymbol{\theta})$ *is differentiable. Let* $\rho_{\max}(\boldsymbol{\theta})$ *be the largest singular value of* $\partial M(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$. *There exists a number* $\lambda^* < 1$ *such that* $\rho_{\max}(\boldsymbol{\theta}) \leq \lambda^*$ *for all* $\boldsymbol{\theta} \in \Theta$ *for sufficiently large* n *and almost every training dataset* D_n.

Note that the phrase "almost every training dataset D_n" is used in the probabilistic sense: since $M(\theta)$ (and hence $\rho_{\max}(\theta)$) depends on the realized sample D_n, the inequality $\rho_{\max}(\theta) \leq \lambda^*$ is understood to hold, for sufficiently large n, almost surely with respect to the sampling distribution of D_n (i.e., it may fail only on a set of datasets of probability zero). As discussed in Nielsen (2000), the differentiability of $M(\boldsymbol{\theta})$ ensures that the EM update is a contraction and subsequently converges to a fixed point of the mapping, and the continuity ensures that $\rho_{\max}(\boldsymbol{\theta}) < 1$ in a neighborhood of the fixed point. If the mapping has multiple fixed points, it is only locally a

contraction. When n is sufficiently large, we expect that each of the fixed points corresponds to an optimal solution equivalent to $\boldsymbol{\theta}^*$, up to certain loss-invariant transformations. Therefore, mathematically, we can treat the fixed point as unique. This leads to the following lemma.

Lemma 6. *Suppose Assumptions 1–4 hold and* $\sum_{l=1}^{h+1} d_l\sigma_l^2/\sigma_{l-1}^2 \prec n$, *where* σ_0 *is a constant. Then* $\|\hat{\boldsymbol{\theta}}_n^{(t)} - \boldsymbol{\theta}^*\| \xrightarrow{p} 0$ *as* $t \to \infty$ *and* $n \to \infty$.

Lemma 6 shows that the IRO algorithm provides a valid way for finding a consistent parameter estimator for a sublinear DNN. Our proof implies that with an appropriate choice of σ_l's, such a consistent estimator can also be obtained by directly maximizing the log-likelihood function of the complete data, i.e., setting

$$\widehat{\boldsymbol{\theta}}_n^* = \arg\max_{\boldsymbol{\theta}} \Big\{ \frac{1}{n} \log \pi(\mathbb{Y}, \mathbb{Y}_{\text{mis}} | \mathbb{X}, \boldsymbol{\theta}) \Big\}, \tag{8}$$

where $\mathbb{Y}_{\text{mis}}$ denotes imputed values of $\boldsymbol{Y}_{\text{mis}}$ for all n observations. Furthermore, it follows from Lemma 1 that a consistent estimator of $\boldsymbol{\theta}$ can also be obtained by directly maximizing the log-likelihood function of the DNN model, i.e., setting

$$\widehat{\boldsymbol{\theta}}_{\text{DNN,n}}^* = \arg\max_{\boldsymbol{\theta}} \Big\{ \frac{1}{n} \log \pi_{DNN}(\mathbb{Y} | \mathbb{X}, \boldsymbol{\theta}) \Big\}. \tag{9}$$

Theorem 1. *Suppose Assumptions 1–4 hold. (i) With an appropriate choice of* σ_l*'s such that* $d_{h+1}(\prod_{i=l+1}^{h} d_i^2) d_l\sigma_l^2 \prec \frac{1}{h}$ *for all* $l \in \{1, 2, \ldots, h\}$ *and* $\sum_{l=1}^{h+1} d_l\sigma_l^2/\sigma_{l-1}^2 \prec n$, *then the estimator (8) is consistent. (ii) If* $\sum_{l=1}^{h+1} d_l \prec n$, *then the estimator (9) is consistent, i.e.,* $\|\widehat{\boldsymbol{\theta}}_{\text{DNN,n}}^* - \boldsymbol{\theta}^*\| \xrightarrow{p} 0$ *as* $n \to \infty$.

By Theorem 1, one can set the width of the DNN to be of order $O(n^\gamma)$ for some $\frac{1}{2} < \gamma < 1$ and set its depth to be of order $O(n^{\gamma'})$ for some $0 < \gamma' < 1 - \gamma$, which ensures the total number of hidden and output neurons $\sum_{l=1}^{h+1} d_l \prec n$ and thus the parameter estimation consistency holds. Notably, the total number of parameters of the sublinear DNN can be much greater than n. In other words, *the sublinear DNN can be over-parameterized in the conventional sense, while still achieving consistency in parameter estimation.*

3.3 Approximation Power of Sublinear DNNs

Theorem 1 rests on the implicit assumption that the sublinear DNN can adequately approximate the target function. Given the model's structural constraints, a natural question arises: *can it still approximate common target classes, e.g., continuous functions on compact sets, arbitrarily well as the sample size* $n \to \infty$*?* While a complete characterization remains open, we establish positive results for several important classes of functions, as detailed below.

To understand the approximation power of DNNs, a line of work has analyzed compositional functions, see, e.g., Schmidt-Hieber (2020); Bauler and Kohler (2019); Poggio et al. (2017), motivated by the compositional structure of DNNs. Combining the approximation theory of Poggio et al. (2017) with Theorem 1, we obtain:

Theorem 2. *Let* $f(\boldsymbol{x})$ *be defined on a compact domain in* $\mathbb{R}^{d_0}$ *and admit a hierarchical compositional representation in which each constituent depends on at most* s *variables (with* $s \leq d_0$ *and being fixed).*

(i) If f is Lipschitz with the dimension $d_0 = O(n^\alpha)$ for some $0 < \alpha < 1$, then for any $\varepsilon = n^{-(1-\alpha-\delta)/s}$ with $0 < \delta < 1-\alpha$, there exists a sublinear ReLU DNN such that $\|f - f_\theta\| \leq \varepsilon$ as $n \to \infty$, where f_θ denotes the DNN function.

(ii) If f is continuously differentiable with $d_0 = O(n^\alpha)$ for some $0 < \alpha < 1$, then the same conclusion holds for sublinear DNNs that have a smooth activation function, such as sigmoid or tanh.

Refer to Appendix A.7 for the proof. Beyond compositional classes, the approximation theory of deep-ReLU networks established in Montanelli and Du (2019) yield analogous guarantees for functions in the Korobov space (denoted by $\mathcal{K}^{2,p}$ with an equipped ℓ^p-norm) — a subspace of the Sobolev space with dominating mixed smoothness, i.e., all mixed partial derivatives up to order 2 exist. The proof of Montanelli and Du (2019) leverages the ability of deep networks to approximate sparse grids (Zenger, 1991) via a binary tree structure, resembling the compositional structure used in Poggio et al. (2017).

Theorem 3. *If $f(\boldsymbol{x}) \in \mathcal{K}^{2,p}([0,1]^{d_0})$ (i.e., defined on a compact domain in $\mathbb{R}^{d_0}$) with $d_0 \prec n$, then for any $\varepsilon = n^{-(1/2-\delta)}$ with $0 < \delta < 1/2$, there exists a sublinear ReLU DNN such that $\|f - f_\theta\| \leq \varepsilon$ as $n \to \infty$.*

Additionally, we note that Theorem 1 complies with the neural scaling law. Both Hestness et al. (2017) and Kaplan et al. (2020) investigated scaling between model size (i.e., the number of parameters) and data size; the former found sub-linear scaling of model size with data size, whereas the latter found a super-linear scaling. Specifically, by Kaplan et al. (2020), the network width can increase with the data size at a polynomial rate of $d_l \prec n^{0.676} (\approx n^{0.5/0.74})$ for neural language models; and by Hestness et al. (2017), the scaling law $d_l \prec n^{0.5}$ holds for different model architectures in four deep learning domains: machine translation, language modeling, image processing, and speech recognition. For both scaling laws, the conditions of Theorem 1-(ii) can be satisfied by choosing an appropriate growth rate for the depth of the DNN, ensuring the parameter estimation consistency holds.

Sublinear DNNs can be trained effectively using SGD. For conventional nonlinear models, obtaining the exact maximum likelihood estimator (MLE) is often challenging, however, DNNs present a different picture: in practice, they often interpolate the training data (achieving essentially zero empirical risk), which coincides with attaining the MLE. This interpolation phenomenon has been widely documented in the deep-learning literature; for example, Zhou et al. (2019) noted that SGD, although considered as a randomized algorithm, converges in an intrinsically deterministic manner to a global minimum. See Section 1.3 of the supplement for an ablation study on SGD's sensitivity to learning rates. We find that sublinear DNNs maintain stable training and test error across a wide range of learning rates, whereas wide DNNs are markedly more sensitive.

4 Numerical Experiments

We first test the performance of the IRO algorithm 1 for StoNet training; see Supplement 1.2 for details. Our numerical experiments show that the StoNet trained with IRO and the DNN

trained with SGD perform similarly, which is consistent with the theory established in Lemma 6 and Theorem 1. In practice, the IRO algorithm requires solving a series of regressions on the entire dataset for each iteration, which can be slow for large datasets. Therefore, we use SGD in all subsequent experiments, while using StoNet with IRO as a bridge for transferring theory and methods from linear models to DNNs.

4.1 Feature Learning Consistency

To illustrate the consistency of feature learning in sublinear DNNs, we consider the following two-hidden-layer neural network model:

$$y_i = \boldsymbol{w}_3 \tanh(\boldsymbol{w}_2 \tanh(\boldsymbol{w}_1 \boldsymbol{x}_i + \boldsymbol{b}_1) + \boldsymbol{b}_2) + b_3 + \sigma\epsilon_i, \quad i = 1, 2, \ldots, n, \tag{10}$$

where ϵ_i's are *i.i.d.* standard Gaussian random errors. The neural network has a structure of p-5-5-1 with $p = 20$, $\boldsymbol{x}_i$'s are drawn independently from $N_p(0, I_p)$. The neural network parameters include $\boldsymbol{w}_1 \in \mathbb{R}^{5\times 20}$, $\boldsymbol{w}_2 \in \mathbb{R}^{5\times 5}$, $\boldsymbol{w}_3 \in \mathbb{R}^{1\times 5}$, $\boldsymbol{b}_1 \in \mathbb{R}^5$, $\boldsymbol{b}_2 \in \mathbb{R}^5$, and $b_3 \in \mathbb{R}$, and each of their elements is randomly drawn the set $\{-1, -0.5, 0, 0.5, 1\}$. Multiple datasets have been simulated from the model (10) under each setting: $n = 500$ and $50,000$. Obviously, this function belongs to the Korobov space and is also a hierarchical composition function, where p is considered as a fixed value.

We modeled the simulated data using 6 different DNNs with the respective structures given by p-5-5-1, p-1000-1000-1, p-10-10-1, p-10-10-10-1, p-10-10-10-10-1, and p-10-10-10-10-10-1. To demonstrate the consistency of neural feature learning, we calculate the canonical correlation (CC) coefficient $\rho_{k,1:k'} = \rho(\nu_k(\boldsymbol{w}_1^T\boldsymbol{w}_1), \nu_{1:k'}(\hat{\boldsymbol{w}}_1^T\hat{\boldsymbol{w}}_1))$, where $\nu_k(\boldsymbol{w}_1^T\boldsymbol{w}_1)$ denotes the k-th eigenvector of $\boldsymbol{w}_1^T\boldsymbol{w}_1$, $\nu_{1:k'}(\hat{\boldsymbol{w}}_1^T\hat{\boldsymbol{w}}_1)$ denotes top k' eigenvectors of $\hat{\boldsymbol{w}}_1^T\hat{\boldsymbol{w}}_1$, and $\hat{\boldsymbol{w}}_1$ denotes an estimator of $\boldsymbol{w}_1$. Under this setting, $\rho_{k,1:k'}$ is given by

$$\rho_{k,1:k'} = \max_{(c_1,\ldots,c_{k'})^T \in \mathbb{R}^{k'}} \mathrm{Corr}\left(\nu_k(\boldsymbol{w}_1^T\boldsymbol{w}_1), c_1\nu_1(\hat{\boldsymbol{w}}_1^T\hat{\boldsymbol{w}}_1) + \cdots + c_{k'}\nu_{k'}(\hat{\boldsymbol{w}}_1^T\hat{\boldsymbol{w}}_1)\right),$$

which measures the extent to which the neural feature $\nu_k(\boldsymbol{w}_1^T\boldsymbol{w}_1)$ is recovered by the learned neural network. As shown in Table A5, $\boldsymbol{w}_1^T\boldsymbol{w}_1$ contains three major eigenvalues. Table 1 presents the values of $\rho_{k,1:k'}$ for $k = 1, 2, 3$ and $k' = 3$.

A careful examination of Table 1 shows that the sublinear DNNs not only recover the neural features but also preserve their orders as n becomes large. Note that the network p-1000-1000-1 is considered wide when $n = 500$ but becomes sublinear in width for $n = 50,000$, and its results clearly highlight the importance of a sublinear structure for effective neural feature recovery. It is worth noting that this neural network has a total of 1,023,001 parameters, making it highly over-parameterized — a scenario commonly encountered in our deep learning practice.

Furthermore, the recovery of low-dimensional neural features by the network p-1000-1000-1 suggests that it contains a large number of redundant parameters, supporting the use of sublinear DNNs and the low-rank approximation method proposed in LoRA (Hu et al., 2022). The sublinear DNN actually provides loose, yet effective, upper bounds for the ranks that can be used for each hidden layer of the wide DNN in LoRA. Our results also indicate that the depth of

Table 1: Canonical correlations $\rho_{k,1:k'}$ $(k = 1, 2, 3)$ for different DNN structures, where the mean CC coefficient and its standard deviation (reported in parenthesis) are calculated by averaging over 5 independent datasets.

n	CC	p-5-5-1	p-1000-1000-1	p-10-10-1	p-10-10-10-1	p-10-10-10-10-1	p-10-10-10-10-10-1
500	$\rho_{1,1:1}$	0.95(0.02)	0.16(0.03)	0.60(0.15)	0.68(0.14)	0.58(0.14)	0.58(0.08)
500	$\rho_{1,1:2}$	0.97(0.02)	0.28(0.05)	0.84(0.06)	0.89(0.04)	0.77(0.12)	0.66(0.09)
500	$\rho_{1,1:3}$	0.99(0.00)	0.37(0.07)	0.89(0.04)	0.90(0.04)	0.79(0.11)	0.81(0.05)
50000	$\rho_{1,1:1}$	1.00(0.00)	0.88(0.03)	0.88(0.08)	0.88(0.07)	0.96(0.02)	0.88(0.08)
50000	$\rho_{1,1:2}$	1.00(0.00)	0.95(0.01)	0.96(0.03)	0.99(0.00)	0.97(0.01)	0.99(0.01)
50000	$\rho_{1,1:3}$	1.00(0.00)	0.97(0.01)	0.99(0.00)	1.00(0.00)	0.99(0.00)	1.00(0.00)
500	$\rho_{2:1:1}$	0.22(0.08)	0.22(0.04)	0.48(0.11)	0.49(0.11)	0.55(0.10)	0.59(0.08)
500	$\rho_{2,1:2}$	0.88(0.09)	0.47(0.02)	0.79(0.05)	0.81(0.04)	0.64(0.09)	0.76(0.03)
500	$\rho_{2,1:3}$	0.90(0.09)	0.52(0.04)	0.89(0.04)	0.87(0.05)	0.82(0.04)	0.83(0.03)
50,000	$\rho_{2,1:1}$	0.06(0.01)	0.29(0.10)	0.25(0.04)	0.33(0.12)	0.13(0.03)	0.27(0.07)
50,000	$\rho_{2,1:2}$	1.00(0.00)	0.94(0.01)	0.97(0.03)	0.99(0.00)	0.83(0.11)	0.89(0.08)
50,000	$\rho_{2,1:3}$	1.00(0.00)	0.97(0.01)	0.99(0.01)	0.99(0.00)	0.97(0.02)	1.00(0.00)
500	$\rho_{3,1:1}$	0.12(0.03)	0.24(0.07)	0.28(0.01)	0.19(0.02)	0.26(0.07)	0.27(0.09)
500	$\rho_{3,1:2}$	0.28(0.09)	0.34(0.06)	0.49(0.08)	0.38(0.08)	0.61(0.07)	0.38(0.07)
500	$\rho_{3,1:3}$	0.96(0.03)	0.42(0.07)	0.83(0.09)	0.62(0.14)	0.82(0.05)	0.63(0.08)
50,000	$\rho_{3,1:1}$	0.04(0.01)	0.14(0.05)	0.19(0.11)	0.10(0.03)	0.16(0.05)	0.17(0.12)
50,000	$\rho_{3,1:2}$	0.06(0.2)	0.28(0.05)	0.22(0.11)	0.13(0.02)	0.44(0.13)	0.29(0.15)
50,000	$\rho_{3,1:3}$	1.00(0.00)	0.93(0.04)	0.99(0.01)	0.99(0.00)	0.97(0.01)	0.99(0.01)

the DNN affects the recovery of neural features, but not significantly. The similar performance of all the different DNNs in neural feature recovery aligns well with the findings of Li et al. (2018), where it was observed that the intrinsic dimension of the DNN remains stable even as models grow in width and depth.

For a thorough comparison for the performance of sublinear and wide DNNs, we reported their training and test errors in Table 2. The comparison highlights the importance of consistent feature learning in DNN prediction. In particular, all the networks achieve oracle-level training and test errors when $n = 50,000$, where major neural features of the data have been successfully recovered as implied by Table 1. In contrast, when $n = 500$, the test errors appear to depend on the extent of neural feature recovery.

Table 2: Mean squared training and test errors produced by different DNN structures, where the mean and standard deviation (reported in parenthesis) are calculated by averaging over 5 independent datasets.

n		p-5-5-1	p-1000-1000-1	p-10-10-1	p-10-10-10-1	p-10-10-10-10-1	p-10-10-10-10-10-1
—	Model size	141	1,023,001	331	442	553	664
500	train	0.01(0.00)	0.00(0.00)	0.01(0.00)	0.01(0.00)	0.01(0.00)	0.00(0.00)
	test	0.05(0.02)	0.21(0.04)	0.08(0.02)	0.14(0.05)	0.15(0.03)	0.16(0.04)
50,000	train	0.01(0.00)	0.01(0.00)	0.01(0.00)	0.01(0.00)	0.01(0.00)	0.01(0.00)
	test	0.01(0.00)	0.01(0.00)	0.01(0.00)	0.01(0.00)	0.01(0.00)	0.01(0.00)

As mentioned earlier, $\boldsymbol{\theta}$ is unique up to loss-invariant transformations, such as reordering hidden neurons within the same layer or simultaneously altering the signs or scales of certain weights and biases. This invariance property makes it particularly challenging to demonstrate the consistency of DNN parameter estimation. To address this challenge, we present in Table A4 the canonical correlations $\rho_{4,1:k'}$ and $\rho_{5,1:k'}$ (for $k' = 1, 2, \ldots, 5$) achieved by the network p-5-5-1 with $n = 50,000$, and in Table A5 the eigenvalues. Based on the results shown in Table 1, Table

A4, and Table A5, we can conclude that for this network, the eigenvalues and eigenvectors of $\boldsymbol{w}_1^T\boldsymbol{w}_1$ have been asymptotically recovered by those of $\hat{\boldsymbol{w}}_1^T\hat{\boldsymbol{w}}_1$. Furthermore, since $d_1 \leq p$ in this network, the recovery of the eigenvalues and eigenvectors implies the recovery of $\boldsymbol{w}_1$ (up to elementary operations). Note that in the case where $d_1 \leq p$, each element of $\boldsymbol{w}_1$ can be uniquely determined by solving the equation

$$\boldsymbol{w}_1^T\boldsymbol{w}_1 = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{d_1}]\text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_{d_1}\}[\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{d_1}]^T,$$

where $\{\lambda_1, \ldots, \lambda_{d_1}\}$ and $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{d_1}\}$ denote the given eigenvalues and eigenvectors, respectively. In summary, we have provided an effective method to verify the consistency of parameter estimation for the DNNs with an upper triangular structure (i.e., where $d_i \leq d_{i-1}$ for $i = 1, 2, \ldots, h, h{+}1$). This eigen analysis-based method accounts for some loss-invariant transformations that $\boldsymbol{\theta}$ is subject to.

4.2 Double Descent and Beyond

Double descent is a surprising phenomenon in machine learning, which describes the observation that the test error of a model drops as the model grows ever larger into the highly overparameterized regime relative to the training sample size, see e.g., Belkin et al. (2019); Adlam and Pennington (2020); Schaeffer et al. (2023). This phenomenon will be explained at the end of this subsection from a perspective of neural feature learning.

MNIST As in Belkin et al. (2019), we worked with a subset of MNIST (with $n_{train} = 4000$, $p = 784$, and $K = 10$ classes) as training data. We trained a one-hidden-layer neural network: 784-L-10, where L is the hidden layer width, and measured its prediction performance on a test dataset with $n_{test} = 10,000$. Figure 1(a) shows the resulting training and test errors, where the second descent in test errors occurs with L ranging 50$\sim$1000. Notably, for each $L \in [50, 1000]$, the resulting DNN is sublinear in width, although its total number of parameters can be much greater than n_{train}. Our feature-learning consistency theory provides a principled explanation for the second descent phenomenon, as detailed below.

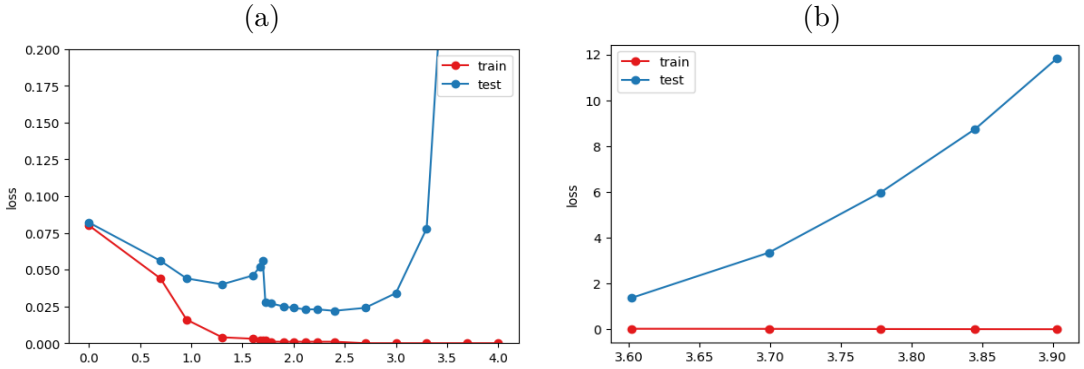

Figure 1: MNIST example, where the y-axis represents ℓ_2-loss and the x-axis represents $\log_{10}(L)$: (a) training both layer weights; (b) training the second layer weights only.

Importantly, Figure 1 shows that as L further increases, the test error increases again. That is, this example also exhibits a "double ascent" phenomenon. We would attribute the second

ascent to the lack of feature learning consistency. To make this point clearer, we trained the one-hidden-layer neural networks again with $L \in [4000, 8000]$, each of which forms a wide neural network. For each of the wide networks, we fixed the first layer's weights, initialized with $N(0, 0.5^2)$; and trained the second layer's weights only, which were initialized with $N(0, 0.1^2)$. Since $L > n_{train}$, the second layer forms K small-n-large-p logistic regressions and zero-training error solutions exist. As shown in Figure 1(b), these networks can still attain zero training errors, but their test errors are very large. By design, the features represented by the first hidden layer of the wide DNNs are purely noise; the resulting large test errors indicate the importance of feature learning consistency.

In what follows, we further explain the importance of feature learning consistency from two perspectives. First, let's understand why a wide DNN can predict well, if it is trained by a gradient descent method. Consider a StoNet model for nonlinear regression. Suppose that the StoNet model is true, and its hidden layer outputs $\boldsymbol{Y}_{\text{mis}}$ are observed. Training such a StoNet is reduced to solving a series of high-dimensional linear regressions. If a gradient descent method is used, then training the StoNet is equivalent to solving a series of ridge regressions with zero penalty, as the gradient descent method provides an implicit regularization for the models in training, see e.g., Gunasekar et al. (2017), Soudry et al. (2018), and Ji and Telgarsky (2019). By the recovery property of ridge regression (Kobak et al., 2020), each of the ridge regressions leads to consistent feature learning for its relevant variables as well as consistent estimation for its response. Therefore, the wide DNN can still predict well, if sufficiently trained with gradient decent. For classification problems, the explanation is similar.

Table 3: Training and test errors produced by a wide neural network with structure 784-5000-10, different learning rates and epochs, for the subset MNIST data.

Learning rate	#epoch	training loss	test loss
0.001	4,000	0.000	0.523
0.001	40,000	0.000	0.515
0.001	100,000	0.000	0.497
0.01	4,000	0.000	0.048

On the other hand, as pointed out by Soudry et al. (2018), the convergence of the gradient descent to its implicit regularization limit is very slow, only logarithmic in the convergence of the loss itself. This suggests that the "double ascent" phenomenon in Figure 1(a) might be due to insufficient training. To testify this, we re-trained a wide neural network with structure p-5000-10. We set the learning rate to 0.001 and increased the number of epochs from 4,000 to 40,000 and 100,000. We found that with lengthened runs, the test error of the wide DNN can be reduced, but at a very slow rate, see Table 3. We have also set the learning rate to 0.01 and re-trained the model for 4,000 epochs, which yielded low test errors and recovered the double descent phenomenon. Other than test errors, we compared the features learned in different runs. Figure 2 shows that consistent feature learning can be achieved by the wide network with a learning rate of 0.01, while a learning rate of 0.001 may require an extremely long time to achieve the same result. In summary, Table 3 and Figure 2 underscore the importance of feature learning consistency, reinforcing the evidence we observed in Section 4.1.

Additionally, we compared in Figure 3 the features learned by the sublinear and wide DNNs

with a learning rate of 0.001, where all the networks have been sufficiently trained to zero training errors. The comparison indicates that the sublinear DNN works better than the wide ones in feature extraction.

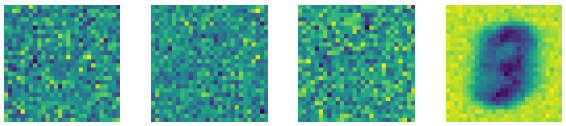

Figure 2: Features learned by a wide neural network p-5000-10 for the subset MNIST data under the settings (learning rate, epoch)=(0.001, 4,000), (0.001, 40,000), (0.001, 100,000), and (0.01, 4,000), from left to right, where the features were extracted from the first hidden layer using the method as described in Radhakrishnan et al. (2024).

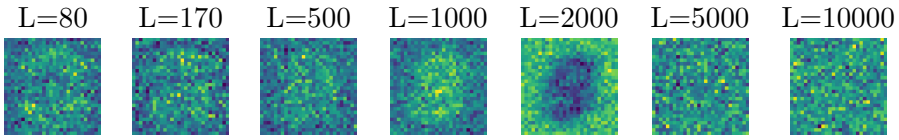

Figure 3: Features learned by the neural networks with structure p-L-10 for different values of L, where the features were extracted from the first hidden layer using the method as described in Radhakrishnan et al. (2024).

Other than classification, we have tried a nonlinear regression problem, see Supplement 1.3. As shown by Figure A2, if we fix the first layer weights at random values and trained the second layer weights only, the training error can be reduced to 0, while the test error is large. Again, this underscores the importance of feature learning consistency for prediction.

In summary, our experiments indicate that *feature learning consistency is crucial for DNNs to achieve accurate predictions*. Whether a network is narrow or wide, it can perform well as long as it effectively extracts the correct features. From a feature learning perspective, we provide the following conjectural explanation for the double descent phenomenon:

First of all, a large value of L allows more features to be extracted from the data; however, as mathematically shown by Chang (1983), the importance of features in separating data classes is not necessarily aligned with their eigenvalues. For instance, Chang (1983) constructed a two-component mixture Gaussian example, where the two components are only well-separated in the subspace of the first and last eigenvectors. In the context of DNNs, some useful features with large eigenvalues can be learned when L is small, leading to the first descent in test errors. As L increases to a moderate value, additional noisy features may be learned, resulting in an ascent in test errors. Finally, some useful features with small eigenvalues may only be learned when L is sufficiently large, leading to the second descent in test errors, where the network is highly over-parameterized.

4.3 More Examples: Sublinear or Wide?

In this subsection, we delve deeper into the choice of hidden layer widths from the perspective of feature learning. Theoretically, as shown in a series of papers (e.g., Gunasekar et al. (2017), Soudry et al. (2018), Ji and Telgarsky (2019)), the gradient descent method provides implicit regularization during model training. Specifically, for high-dimensional linear regression models

initialized at the origin, gradient descent converges to the solution with the minimum Euclidean norm. By applying this result to the StoNet model (3), we arrive at the following solution for $\boldsymbol{w}_i$ at each hidden layer:

$$\hat{\boldsymbol{w}}_i = \mathbb{Y}_i^T \Psi(\mathbb{Y}_{i-1})(\Psi(\mathbb{Y}_{i-1})^T \Psi(\mathbb{Y}_{i-1}) + \tilde{\sigma}^2 I)^{-1},$$

where $\tilde{\sigma}^2 \geq 0$ represents the implicit penalty coefficient, and $\mathbb{Y}_i \in \mathbb{R}^{n \times d_i}$ denotes imputed $\boldsymbol{Y}_i$ values for all n observations. This leads to the rank constraint:

$$\text{rank}(\hat{\boldsymbol{w}}_i^T \hat{\boldsymbol{w}}_i) \leq \min\{\text{rank}(\Psi(\mathbb{Y}_{i-1})), \text{rank}(\mathbb{Y}_i)\} \leq \min\{n, d_i, d_{i-1}) \leq n,$$

which indicates that an overly wide neural network will not learn more than n features at each hidden layer. By the single value decomposition of $\hat{\boldsymbol{w}}_i$, it is clear that features corresponding to zero eigenvalues will not affect the values of $\mathbb{Y}_i$. Therefore, to enable the DNN to extract more features from the data, one should set d_i's to be reasonably large, but not necessarily greater than n.

To illustrate this finding, we considered a simulation study, see Section 1.4 of the supplement, where the true regression function has a hierarchical composition structure and the networks $p-L-1$, $p-L-L-1$, and $p-L-L-L-L-1$ are trained. We set $n = 500$ and $L \in \{2, \ldots, 1000\}$. Additionally, we considered three UCI datasets: Boston housing, Yacht, and Energy. For each dataset, we tried DNNs with structure p-$L - \cdots - L$-1, with L ranging from 100 to 2,000 and h ranging from 2 to 7. For real-data problems, a slightly deeper architecture may better capture the unknown compositional structure of the true function. For evaluation, we performed five random train/test splits and trained a fresh network on each split. Refer to Tables A6–A8 for the results.

The results indicate that, with appropriate width and depth, sublinear DNNs can perform as well as or better than wide DNNs in prediction. Our findings recommend using sublinear architectures with a reasonably large width and suitable depth so that useful features are extracted and the compositional structure is captured. Moreover, to match the predictive performance of wider counterparts, the depth of a sublinear DNN may need to increase roughly inversely with its width.

4.4 CelebA

As another application of the sublinear DNN, we consider an example of feature extraction in classifying images from the CelebA dataset (Liu et al., 2015). As in Radhakrishnan et al. (2024), we employed a fully connected ReLU DNN for the task. The DNN we used has a structure of $3 \times 64 \times 64 - L - L - L - L - 2$ with $L = 1024$. Therefore, the DNN is still of sublinear width when applied to the CelebA data with a training sample size $n_{train} = 14,000$. We trained the fully connected DNN using SGD with a momentum parameter of 0.9, a learning rate of 0.05, a mini-batch size of 64, and 100 epochs. Figure 4 shows four features extracted in training, which indicate the success of feature learning by the sublinear DNN.

Figure 4: Four features extracted from the first hidden layer, as described in Radhakrishnan et al. (2024), in training a fully connected ReLU DNN with structure $3 \times 64 \times 64$-L-L-L-L-2 for the CelebA data: glass, smile, hat, and arched eyebrow, from left to right.

5 Structure Analysis for Large-Scale DNNs

It is worth noting that many large-scale DNNs, such as AlexNet (Krizhevsky et al., 2012), VGGNet (Simonyan and Zisserman, 2014), ResNet (He et al., 2016), and GoogLeNet (Szegedy et al., 2015), belong to the class of sublinear DNNs in their benchmark studies, despite containing a huge number of parameters. For any deep CNN, we can still randomize the feeding value of each node with incoming trainable connections as in (3), thereby enabling the construction of an asymptotically equivalent StoNet for it. For each node, the number of incoming connections, i.e., the dimension of explanatory variables of the corresponding regression, is calculated as $(s_l^{(1)} * s_l^{(2)} * d_{l-1} + 1)$, where $s_l^{(1)} * s_l^{(2)}$ denotes the filter size and corresponds to the fixed s value in the constituent map of the compositional function (see Theorem 2), and d_{l-1} denotes the number of filters in the previous layer, and '1' represents the bias term. For a deep CNN belonging to the class of sublinear DNNs, the following two conditions need to be satisfied: $\max_l (s_l^{(1)} * s_l^{(2)} * d_{l-1} + 1) \prec n$ and $\sum_l d_l \prec n$. The latter condition can also be interpreted as the total number of regressions formed in the stochastic deep CNN. The structures of the deep CNNs are analyzed in the following, based on the summary provided by Aqeel Anwar at `https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96`.

AlexNet is one of the earliest deep CNNs, which won the 2012 ImageNet LSVRC-2012 challenge. It comprises a total of 62.4 million trainable parameters, including 5 convolutional layers and 3 fully connected (FC) layers. In this network, the maximum number of incoming connections to a single node is 9,217, achieved at the first FC layer, and the total number of nodes with incoming trainable connections is 10,568. VGG16 has approximately 138.4 million parameters. In VGG16, the maximum number of incoming connections to a single node is 25,089, achieved at the first FC layer, and the total number of nodes with incoming trainable connections is 13,544. ResNets have many variants, e.g., ResNet18, ResNet50, and ResNet101. Let's consider ResNet18 as an example. It comprises approximately 11.5 million trainable parameters, its maximum number of incoming connections to a single node is 4,609, achieved at layers 15, 16 and 17, and its total number of nodes with incoming trainable connections is 4,904. GoogLeNet has about 6.4 million trainable parameters, in which the maximum number of incoming connections to a single node is 1,729, achieved in Inception 5b, and the total number of nodes with incoming trainable connections is 8,280.

In summary, all these networks are sublinear when trained on large-scale datasets such as ImageNet, CIFAR10, CIFAR100, and MNIST, each with $n \geq 50,000$ training samples. Moreover,

because images exhibit an inherent hierarchical, compositional structure, Theorems 1 and 2 apply to these sublinear networks. Taken together, these theoretical insights and the preceding analysis help explain why such large-scale networks achieve exceptional predictive performance after sufficient training on large-scale datasets.

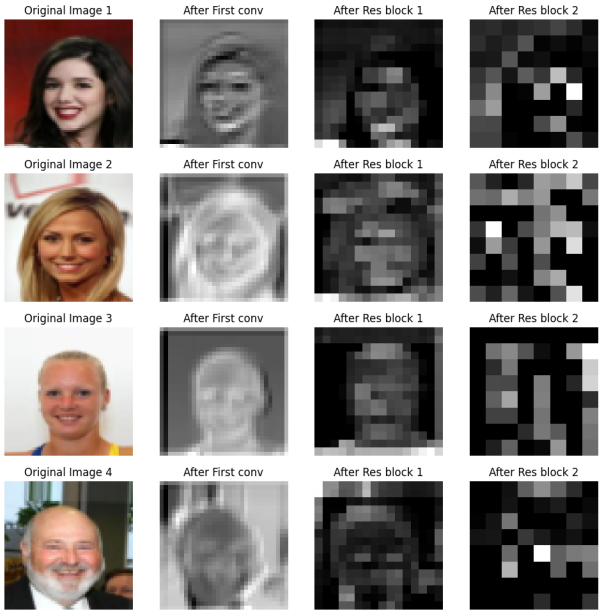

Figure 5: Activation maps of the smile features extracted from the first convolutional layer, first residual block, and second residual block.

We have tested feature consistency on CelebA using ResNet18. Instead of the original ResNet18, which has a kernel size configuration of (64, 128, 256, 512), we utilized a modified version of (16, 32, 64, 128). The modified ResNet18 was trained with SGD with momentum 0.9, a learning rate of 0.1, and a batch size of 64. For feature visualization, many methods are available, as listed at https://github.com/justinbellucci/cnn-visualizations-pytorch. We employed activation map visualization, processing images at each layer and visualizing the respective image representation. In other words, activation map visualization demonstrates what the image looks like after the application of each filter. Figure 5 shows that the sublinear CNNs works well for feature learning.

6 Conclusion

We study sublinear DNNs and prove that, in the large-sample limit, they achieve universal approximation and feature-learning consistency for hierarchically compositional functions. We also analyze AlexNet, VGGNet, and ResNet, showing that these deep CNNs are sublinear on their image-classification benchmarks. Because natural images are hierarchically compositional, our results offer a statistical explanation for the strong performance of large-scale deep learning models in image processing. Our theory identifies a regime in which consistent prediction is guaranteed for large-scale deep learning models despite possible over-parameterization. Empirically, sublinear DNNs match or outperform wide DNNs in prediction accuracy and are more robust to training hyperparameter settings.

The theoretical proof of this paper leverages StoNet as a surrogate for the DNN, creating a bridge between linear models and DNNs. Beyond sublinear DNNs, this approach can be applied to sparse deep learning by extending the sparse learning theory from linear models to DNNs. Additionally, we conjecture that the StoNet could enable the extension of benign overfitting theory from linear models to super-wide DNNs, leveraging its capability in sufficient dimension reduction (Liang et al., 2022).

In summary, this work validates the effectiveness of sublinear DNNs for learning features from hierarchically compositional functions and provides theoretical guidance for designing appropriate network architectures for tasks such as image processing, where hierarchical composition is intrinsic. The main takeaways are: (i) sublinear DNNs achieve feature-learning consistency for hierarchically compositional functions, even when the total number of parameters exceeds the sample size; (ii) although wide DNNs can drive training error near zero, their predictive performance can be sensitive to the optimization algorithms and hyperparameter settings, whereas sublinear DNNs are notably more robust; and (iii) sublinear DNNs comply with neural scaling laws, achieve universal approximation for hierarchically compositional functions in the large-sample limit, and may extend to other classes of functions, a direction that merits further study.

References

Ben Adlam and Jeffrey Pennington. Understanding double descent requires a fine-grained bias-variance decomposition. *NeurIPS*, 2020. URL https://api.semanticscholar.org/CorpusID:226278106.

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *ArXiv*, abs/2012.13255, 2020. URL https://api.semanticscholar.org/CorpusID:229371560.

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

Benedikt Bauler and Michael Kohler. On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *The Annals of Statistics*, 47(4):2261–2285, 2019.

Mikhail Belkin, Daniel J. Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849 – 15854, 2019. URL https://api.semanticscholar.org/CorpusID:198496504.

Dankmar Böhning. Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics*, 44:197–200, 1992. URL https://api.semanticscholar.org/CorpusID:118639142.

G. Celeux and J. Diebolt. The sem algorithm: a probabilistic teacher algorithm derived from the em algorithm for the mixture problem. *Computational Statistics Quarterly*, 2:73–82, 1985.

Wei-Chien Chang. On using principal components before separating a mixture of two multivariate normal distributions. *Applied Statistics*, 32:267–275, 1983. URL https://api.semanticscholar.org/CorpusID:116768253.

Jean Daunizeau. Semi-analytical approximations to statistical moments of sigmoid and softmax mappings of normal variables. *arXiv: Machine Learning*, 2017. URL https://api.semanticscholar.org/CorpusID:9191853.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *NeurIPS 2023*, 2023. URL https://api.semanticscholar.org/CorpusID:258841328.

Sören Dittmer, Emily J. King, and Peter Maass. Singular values for relu layers. *IEEE Transactions on Neural Networks and Learning Systems*, 31:3594–3605, 2018. URL https://api.semanticscholar.org/CorpusID:54446869.

Omar M. Eidous and Rima Abu-Shareefa. New approximations for standard normal distribution function. *Communications in Statistics - Theory and Methods*, 49:1357 – 1374, 2020. URL https://api.semanticscholar.org/CorpusID:126960195.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96:1348–1360, 2001.

Yaxin Fang and Faming Liang. Causal-stonet: Causal inference for high-dimensional complex data. *ICLR*, 2024.

M. Farrell, Tengyuan Liang, and S. Misra. Deep neural networks for estimation and inference. *Econometrica*, 89:181–213, 2021.

Manan S. Gandhi, Keuntaek Lee, Yunpeng Pan, and Evangelos A. Theodorou. Propagating uncertainty through the tanh function with application to reservoir computing. *ArXiv*, abs/1806.09431, 2018. URL https://api.semanticscholar.org/CorpusID:49415199.

Gene H. Golub and Charles Van Loan. *Matrix Computations (4th Edition)*. The Johns Hopkins University Press, Baltimore, 2013. URL https://api.semanticscholar.org/CorpusID:60523842.

Suriya Gunasekar, Blake E. Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Implicit regularization in deep matrix factorization. *2018 Information Theory and Applications Workshop (ITA)*, pages 1–10, 2017. URL https://api.semanticscholar.org/CorpusID:3909231.

Roger G. Hart. A formula for the approximation of definite integrals of the normal distribution function. *Mathematics of Computation*, 11:265–265, 1957. URL https://api.semanticscholar.org/CorpusID:44337841.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Frederick Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *ArXiv*, abs/1712.00409, 2017. URL https://api.semanticscholar.org/CorpusID:2222076.

Nicholas John Higham and Sheung Hun Cheng. Modifying the inertia of matrices arising in optimization. *Linear Algebra and its Applications*, pages 261–279, 1998. URL https://api.semanticscholar.org/CorpusID:7130134.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ICLR 2022*, 2022. URL https://api.semanticscholar.org/CorpusID:235458009.

Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In *Annual Conference Computational Learning Theory*, 2019. URL https://api.semanticscholar.org/CorpusID:195769437.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. Scaling laws for neural language models. *ArXiv*, abs/2001.08361, 2020. URL https://api.semanticscholar.org/CorpusID:210861095.

D.P. Kingma and J.L. Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Dmitry Kobak, Jonathan Lomond, and Benoit Sanchez. The optimal ridge penalty for real-world high-dimensional data can be zero or negative due to the implicit ridge regularization. *J. Mach. Learn. Res.*, 21:169:1–169:16, 2020. URL https://api.semanticscholar.org/CorpusID:263872969.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012. URL https://api.semanticscholar.org/CorpusID:195908774.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *ICLR 2018*, 2018. URL https://api.semanticscholar.org/CorpusID:13739955.

F. Liang, Q. Li, and L. Zhou. Bayesian neural networks for selection of drug sensitive genes. *Journal of the American Statistical Association*, 113:955–972, 2018a.

Faming Liang, Bochao Jia, Jingnan Xue, Qizhai Li, and Ye Luo. An imputation–regularized optimization algorithm for high dimensional missing data problems and beyond. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80:899–926, 2018b.

Siqi Liang, Yan Sun, and Faming Liang. Nonlinear sufficient dimension reduction with a stochastic neural network. *NeurIPS*, 2022.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015. URL https://api.semanticscholar.org/CorpusID:459456.

PL Loh and MJ Wainwright. Support recovery without incoherence: A case for nonconvex regularization. *The Annals of Statistics*, 45(6):2455–2482, 2017.

Hadrien Montanelli and Qiang Du. New error bounds for deep ReLU networks using sparse grids. *SIAM J. Math. Data Sci.*, 1(1):78–92, 2019.

S.F. Nielsen. The stochastic em algorithm: Estimation and asymptotic results. *Bernoulli*, 6: 457–489, 2000.

Dwight Nwaigwe and Marek Rychlik. Convergence rates for multi-classs logistic regression near minimum. *ArXiv*, abs/2012.04576, 2020. URL https://api.semanticscholar.org/CorpusID:227738566.

Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep–but not shallow–networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, 14(5):503–519, 2017. doi: 10.1007/s11633-017-1054-2.

Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism for feature learning in neural networks and backpropagation-free machine learning models. *Science*, 383:1461–1467, 2024.

Alvin C. Rencher and G. Bruce Schaalje. *Linear Models in Statistics (2nd Edition)*. Wiley, New Jersey, 2007. URL https://api.semanticscholar.org/CorpusID:118847613.

Rylan Schaeffer, Mikail Khona, Zachary Robertson, Akhilan Boopathy, Kateryna Pistunova, Jason W. Rocks, Ila Rani Fiete, and Oluwasanmi Koyejo. Double descent demystified: Identifying, interpreting & ablating the sources of a deep learning puzzle. *ArXiv*, abs/2303.14151, 2023. URL https://api.semanticscholar.org/CorpusID:257757313.

Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *Annals of Statistics*, 48(4):1875–1897, 2020.

Zhenmei Shi, Junyi Wei, and Yingyu Liang. A theoretical analysis on feature learning in neural networks: Emergence from inputs and advantage over fixed features. *ArXiv*, abs/2206.01717, 2022. URL https://api.semanticscholar.org/CorpusID:249375392.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL https://api.semanticscholar.org/CorpusID:14124313.

Daniel Soudry, Elad Hoffer, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *J. Mach. Learn. Res.*, 19:70:1–70:57, 2018. URL https://api.semanticscholar.org/CorpusID:3994909.

Y. Sun, Q. Song, and F. Liang. Consistent sparse deep learning: Theory and computation. *Journal of the American Statistical Association*, 117(540):1981–1995, 2022.

Yan Sun and Faming Liang. A kernel-expanded stochastic neural network. *Journal of the Royal Statistical Society Series B*, 84(2):547–578, 2022.

Yan Sun and Faming Liang. Uncertainty quantification for large-scale deep neural networks via post-stonet modeling. *Statistica Sinica*, page in press, 2025.

C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996a.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996b.

Blake E. Woodworth, Suriya Gunasekar, J. Lee, Edward Moroshko, Pedro H. P. Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. *Journal of Machine Learning Research*, 125:1–39, 2020. URL https://api.semanticscholar.org/CorpusID:211252492.

Greg Yang and J. Edward Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, 2021. URL https://api.semanticscholar.org/CorpusID:235825390.

G. Zenger. Sparse grids. In Wolfgang Hackbusch, editor, *Parallel Algorithms for Partial Differential Equations: Proceedings of the Sixth GAMM-Seminar, Kiel, January 19–21, 1990*, volume 31 of *Notes on Numerical Fluid Mechanics*. Friedr. Vieweg & Sohn, Braunschweig; Wiesbaden, 1991. ISBN 3-528-07631-3.

C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38:894–942, 2010.

Yi Zhou, Junjie Yang, Huishuai Zhang, Yingbin Liang, and Vahid Tarokh. SGD converges to global minimum in deep learning via star-convex path. *ICLR*, 2019. URL https://api.semanticscholar.org/CorpusID:57373762.

APPENDIX

A Theoretical Proofs

A.1 Useful Lemmas

Lemma A1. *Consider a random matrix* $\mathbb{U} \in \mathbb{R}^{n\times d}$ *with* $n \geq d$. *Suppose that the eigenvalues of* $\mathbb{U}^T\mathbb{U}$ *are upper bounded, i.e.,* $\lambda_{\max}(\mathbb{U}^T\mathbb{U}) \leq \kappa_{\max}$ *for some constant* $\kappa_{\max} > 0$. *Let* $\Psi(\mathbb{U})$ *denote*

an elementwise transformation of $\mathbb{U}$. *Then* $\lambda_{\max}\left((\Psi(\mathbb{U}))^T(\Psi(\mathbb{U}))\right) \leq \kappa_{\max}$ *for the tanh, sigmoid and ReLU transformations.*

Proof. For ReLU, the result follows from Lemma 5 of Dittmer et al. (2018). For *tanh* and *sigmoid*, since they are Lipschitz continuous with a Lipschitz constant of 1, Lemma 5 of Dittmer et al. (2018) also applies. □

Lemma A2 (First-order expansion with stochastic remainder)**.** *Let* $\Psi : \mathbb{R} \to \mathbb{R}$ *be an activation function, which act componentwise on vectors. Assume* $\Psi \in C^2(\mathbb{R})$ *with uniformly bounded second derivatives, i.e.,*

$$\|\Psi''\|_\infty := \sup_{x\in\mathbb{R}} |\Psi''(x)| < \infty.$$

Fix l *and write*

$$\boldsymbol{Y}_l = \tilde{\boldsymbol{Y}}_l + \boldsymbol{e}_l \in \mathbb{R}^{d_l},$$

where $\boldsymbol{e}_l$ *is independent of* $\tilde{\boldsymbol{Y}}_l$, *satisfies* $\mathbb{E}[\boldsymbol{e}_l] = 0$ *and* $\mathrm{Var}(\boldsymbol{e}_l) = \sigma_l^2 I_{d_l}$, *with* $\sigma_l \to 0$. *Then there exists a remainder vector* $\boldsymbol{r}_l \in \mathbb{R}^{d_l}$ *such that*

$$\Psi(\boldsymbol{Y}_l) = \Psi(\tilde{\boldsymbol{Y}}_l) + \nabla_{\tilde{\boldsymbol{Y}}_l}\Psi(\tilde{\boldsymbol{Y}}_l) \circ \boldsymbol{e}_l + \boldsymbol{r}_l, \tag{A1}$$

where $\nabla_{\tilde{\boldsymbol{Y}}_l}\Psi(\tilde{\boldsymbol{Y}}_l) = (\Psi'(\tilde{Y}_{l,1}), \ldots, \Psi'(\tilde{Y}_{l,d_l}))^\top$ *and* $\circ$ *denotes the Hadamard product. Moreover, the remainder satisfies the coordinatewise bound*

$$|r_{l,i}| \leq \frac{\|\Psi''\|_\infty}{2} e_{l,i}^2, \qquad i = 1, \ldots, d_l, \tag{A2}$$

where $r_{l,i}$ *denotes the* i*-th element of* $\boldsymbol{r}_l$ *(defined below). In particular, if* $d_l\sigma_l \to 0$, *then* $\|\boldsymbol{r}_l\|_2 = o_{\mathbb{P}}(\sigma_l)$.

Proof. For each coordinate i, apply Taylor's expansion to Ψ: there exists $\eta_{l,i} \in (0,1)$ such that

$$\Psi(\tilde{Y}_{l,i} + e_{l,i}) = \Psi(\tilde{Y}_{l,i}) + \Psi'(\tilde{Y}_{l,i})e_{l,i} + \frac{1}{2}\Psi''(\tilde{Y}_{l,i} + \eta_{l,i}e_{l,i})e_{l,i}^2.$$

Define $r_{l,i} := \frac{1}{2}\Psi''(\tilde{Y}_{l,i} + \eta_{l,i}e_{l,i})e_{l,i}^2$ and stack over i to obtain (A1). The bound (A2) follows immediately from $|\Psi''(\cdot)| \leq \|\Psi''\|_\infty$.

Next, by (A2),

$$\|\boldsymbol{r}_l\|_2 \leq \frac{\|\Psi''\|_\infty}{2}\Big(\sum_{i=1}^{d_l} e_{l,i}^4\Big)^{1/2} \leq \frac{\|\Psi''\|_\infty}{2}\sum_{i=1}^{d_l} e_{l,i}^2 = \frac{\|\Psi''\|_\infty}{2}\|\boldsymbol{e}_l\|_2^2,$$

where we used $(\sum a_i^2)^{1/2} \leq \sum |a_i|$ with $a_i = e_{l,i}^2$. Therefore, for any $\varepsilon > 0$,

$$\mathbb{P}(\|\boldsymbol{r}_l\|_2 > \varepsilon\sigma_l) \leq \mathbb{P}\Big(\|\boldsymbol{e}_l\|_2^2 > \frac{2\varepsilon}{\|\Psi''\|_\infty}\sigma_l\Big) \leq \frac{\mathbb{E}\|\boldsymbol{e}_l\|_2^2}{(2\varepsilon/\|\Psi''\|_\infty)\sigma_l} = \frac{\|\Psi''\|_\infty}{2\varepsilon}\, d_l\sigma_l,$$

by Markov's inequality and $\mathbb{E}\|\boldsymbol{e}_l\|_2^2 = \mathrm{tr}(\sigma_l^2 I_{d_l}) = d_l\sigma_l^2$. If $d_l\sigma_l \to 0$, the right-hand side tends to 0, proving $\|\boldsymbol{r}_l\|_2/\sigma_l \to 0$ in probability, i.e. $\|\boldsymbol{r}_l\|_2 = o_{\mathbb{P}}(\sigma_l)$. □

Define $\|A\|_{\mathrm{op}} = \sqrt{\lambda_{\max}(A^\top A)}$, where $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of a matrix.

Lemma A3 (Covariance expansion for $\boldsymbol{\Sigma}_l$)**.** *Assume the conditions of Lemma A2 hold, and define* $\boldsymbol{\Sigma}_l := Var(\Psi(\boldsymbol{Y}_l)) \in \mathbb{R}^{d_l \times d_l}$. *Assume in addition that* $\boldsymbol{e}_l$ *has independent coordinates with* $\mathbb{E}(e_{l,i}) = 0$, $Var(e_{l,i}) = \sigma_l^2$, *and* $\mathbb{E}(e_{l,i}^4) \le C_e \sigma_l^4$ *for some constant* $C_e < \infty$, *and that* $\|\Psi'\|_\infty < \infty$. *Let* $U_l = \mathbb{E}[\boldsymbol{r}_l | \tilde{\boldsymbol{Y}}_l]$ *and* $G(\tilde{\boldsymbol{Y}}_l) = \Psi(\tilde{\boldsymbol{Y}}_l) + U_l$. *If* $d_l \sigma_l \to 0$, *then there exists a remainder matrix* $R_{l,n} \in \mathbb{R}^{d_l \times d_l}$ *such that*

$$\boldsymbol{\Sigma}_l = Var(G(\tilde{\boldsymbol{Y}}_l)) + diag\Big\{\sigma_l^2 \, \mathbb{E}[(\nabla_{\tilde{\boldsymbol{Y}}_l} \Psi(\tilde{\boldsymbol{Y}}_l)) \circ (\nabla_{\tilde{\boldsymbol{Y}}_l} \Psi(\tilde{\boldsymbol{Y}}_l))]\Big\} + R_{l,n}, \tag{A3}$$

where $R_{l,n}$ *satisfies* $\|R_{l,n}\|_{\text{op}} = o(\sigma_l^2)$.

Proof. Fix l. By Lemma A2, there exists a measurable remainder $\boldsymbol{r}_l \in \mathbb{R}^{d_l}$ such that

$$\Psi(\boldsymbol{Y}_l) = \Psi(\tilde{\boldsymbol{Y}}_l) + A_l + \boldsymbol{r}_l = G(\tilde{\boldsymbol{Y}}_l) + A_l + \tilde{\boldsymbol{r}}_l, \tag{A4}$$

where $A_l := \nabla_{\tilde{\boldsymbol{Y}}_l} \Psi(\tilde{\boldsymbol{Y}}_l) \circ \boldsymbol{e}_l$ and $\tilde{\boldsymbol{r}}_l = \boldsymbol{r}_l - U_l$. Therefore, $\mathbb{E}[\tilde{\boldsymbol{r}}_l \mid \tilde{\boldsymbol{Y}}] = 0$.

By the law of total variance,

$$\boldsymbol{\Sigma}_l = \mathrm{Var}\Big(\mathbb{E}[\Psi(\boldsymbol{Y}_l) \mid \tilde{\boldsymbol{Y}}_l]\Big) + \mathbb{E}\Big(\mathrm{Var}(\Psi(\boldsymbol{Y}_l) \mid \tilde{\boldsymbol{Y}}_l)\Big). \tag{A5}$$

Since $\mathbb{E}(A_l \mid \tilde{\boldsymbol{Y}}_l) = 0$ and $\mathbb{E}(\tilde{\boldsymbol{r}}_l \mid \tilde{\boldsymbol{Y}}_l) = 0$, by (A4), the *conditional mean term* is

$$\mathrm{Var}\Big(\mathbb{E}[\Psi(\boldsymbol{Y}_l) \mid \tilde{\boldsymbol{Y}}_l]\Big) = \mathrm{Var}(G(\tilde{\boldsymbol{Y}}_l)). \tag{A6}$$

For the *conditional variance term*, conditioning on $\tilde{\boldsymbol{Y}}_l$ and using $\mathbb{E}(A_l \mid \tilde{\boldsymbol{Y}}_l) = 0$ and $\mathbb{E}(\tilde{\boldsymbol{r}}_l \mid \tilde{\boldsymbol{Y}}_l) = 0$,

$$\mathrm{Var}(\Psi(\boldsymbol{Y}_l) \mid \tilde{\boldsymbol{Y}}_l) = \mathrm{Var}(A_l + \tilde{\boldsymbol{r}}_l \mid \tilde{\boldsymbol{Y}}_l) = \mathrm{Var}(A_l \mid \tilde{\boldsymbol{Y}}_l) + \mathrm{Var}(\tilde{\boldsymbol{r}}_l \mid \tilde{\boldsymbol{Y}}_l) + 2\mathrm{Cov}(A_l, \tilde{\boldsymbol{r}}_l \mid \tilde{\boldsymbol{Y}}_l).$$

(i) For $\mathrm{Var}(A_l \mid \tilde{\boldsymbol{Y}}_l)$, taking expectations yields

$$\mathbb{E}[\mathrm{Var}(A_l \mid \tilde{\boldsymbol{Y}}_l)] = \mathrm{diag}\Big\{\sigma_l^2 \, \mathbb{E}[(\nabla_{\tilde{\boldsymbol{Y}}_l} \Psi(\tilde{\boldsymbol{Y}}_l)) \circ (\nabla_{\tilde{\boldsymbol{Y}}_l} \Psi(\tilde{\boldsymbol{Y}}_l))]\Big\}. \tag{A7}$$

(ii) By the law of iterated expectations and $\|\mathrm{Var}(Z)\|_{\text{op}} \le \mathbb{E}\|Z\|_2^2$,

$$\Big\|\mathbb{E}[\mathrm{Var}(\tilde{\boldsymbol{r}}_l \mid \tilde{\boldsymbol{Y}}_l)]\Big\|_{\text{op}} \le \mathbb{E}\|\tilde{\boldsymbol{r}}_l\|_2^2 \le \mathbb{E}\|\boldsymbol{r}_l\|_2^2.$$

Using the coordinatewise bound from Lemma A2, $|r_{l,i}| \le (\|\Psi''\|_\infty/2) e_{l,i}^2$, and the moment assumption $\mathbb{E}(e_{l,i}^4) \le C_e \sigma_l^4$, we get

$$\mathbb{E}\|\boldsymbol{r}_l\|_2^2 = \sum_{i=1}^{d} \mathbb{E}(r_{l,i}^2) \le \frac{\|\Psi''\|_\infty^2}{4} \sum_{i=1}^{d} \mathbb{E}(e_{l,i}^4) \le C \, d_l \, \sigma_l^4, \tag{A8}$$

hence

$$\Big\|\mathbb{E}[\mathrm{Var}(\tilde{\boldsymbol{r}}_l \mid \tilde{\boldsymbol{Y}}_l)]\Big\|_{\text{op}} = O(d_l \sigma_l^4) = o(\sigma_l^3) \qquad \text{if } d_l \sigma_l \to 0. \tag{A9}$$

(iii) Using $\|\mathrm{Cov}(U,V)\|_{\mathrm{op}} \le \mathbb{E}\|U\|_2\|V\|_2$ and then Cauchy–Schwarz,

$$\Big\|\mathbb{E}[\mathrm{Cov}(A_l, \tilde{\boldsymbol{r}}_l \mid \tilde{\boldsymbol{Y}}_l)]\Big\|_{\mathrm{op}} \le \mathbb{E}\|A_l\|_2\|\tilde{\boldsymbol{r}}_l\|_2 \le \sqrt{\mathbb{E}\|A_l\|_2^2}\,\sqrt{\mathbb{E}\|\tilde{\boldsymbol{r}}_l\|_2^2} \le \sqrt{\mathbb{E}\|A_l\|_2^2}\,\sqrt{\mathbb{E}\|\boldsymbol{r}_l\|_2^2}.$$

Moreover,

$$\mathbb{E}\|A_l\|_2^2 = \sum_{i=1}^{d} \mathbb{E}[\Psi'(\tilde{Y}_{l,i})^2 e_i^2] = \sigma_l^2 \sum_{i=1}^{d} \mathbb{E}[\Psi'(\tilde{Y}_{l,i})^2] \le d_l\,\sigma_l^2\,\|\Psi'\|_\infty^2.$$

Combining this with (A8) yields

$$\Big\|\mathbb{E}[\mathrm{Cov}(A_l, \tilde{\boldsymbol{r}}_l \mid \tilde{\boldsymbol{Y}}_l)]\Big\|_{\mathrm{op}} \le \|\Psi'\|_\infty \sqrt{d_l\sigma_l^2}\cdot\sqrt{C d_l \sigma_l^4} = O(d_l\sigma_l^3) = o(\sigma_l^2) \qquad \text{if } d_l\sigma_l \to 0. \tag{A10}$$

Plugging (A6) and the bounds (A7), (A9), (A10) into (A5) yields (A3), which concludes the proof. □

Remark 1. *Recall* $G(\tilde{\boldsymbol{Y}}_l) = \Psi(\tilde{\boldsymbol{Y}}_l) + U_l$ *with* $U_l = \mathbb{E}[\boldsymbol{r}_l \mid \tilde{\boldsymbol{Y}}_l]$. *Then*

$$Var(G(\tilde{\boldsymbol{Y}}_l)) = Var(\Psi(\tilde{\boldsymbol{Y}}_l)) + Var(U_l) + 2\,Cov(\Psi(\tilde{\boldsymbol{Y}}_l), U_l).$$

By Jensen's inequality, (A8)*, and the condition* $d_l\sigma_l \to 0$*,*

$$\mathbb{E}\|U_l\|_2^2 = \mathbb{E}\|\mathbb{E}[\boldsymbol{r}_l \mid \tilde{\boldsymbol{Y}}_l]\|_2^2 \le \mathbb{E}\|\boldsymbol{r}_l\|_2^2 \le C\,d_l\sigma_l^4 = o(\sigma_l^3).$$

Hence

$$\|\,Var(U_l)\|_{\mathrm{op}} \le \mathbb{E}\|U_l\|_2^2 = o(\sigma_l^3).$$

Next, using $\|\,Cov(X,Z)\|_{\mathrm{op}} \le \sqrt{\mathbb{E}\|X\|_2^2}\sqrt{\mathbb{E}\|Z\|_2^2}$ *and the boundedness of* Ψ *for* tanh *and sigmoid (say* $|\Psi(x)| \le B$*), we have*

$$\mathbb{E}\|\Psi(\tilde{\boldsymbol{Y}}_l)\|_2^2 \le d_l B^2, \qquad \mathbb{E}\|U_l\|_2^2 \le C d_l \sigma_l^4,$$

and thus

$$\|\,Cov(\Psi(\tilde{\boldsymbol{Y}}_l), U_l)\|_{\mathrm{op}} \le \sqrt{d_l B^2}\,\sqrt{C d_l \sigma_l^4} = O(d_l\sigma_l^2) = o(\sigma_l).$$

Therefore,

$$\|\,Var(G(\tilde{\boldsymbol{Y}}_l)) - Var(\Psi(\tilde{\boldsymbol{Y}}_l))\|_{\mathrm{op}} \le \|\,Var(U_l)\|_{\mathrm{op}} + 2\|\,Cov(\Psi(\tilde{\boldsymbol{Y}}_l), U_l)\|_{\mathrm{op}} = o(\sigma_l),$$

that is,

$$Var(G(\tilde{\boldsymbol{Y}}_l)) = Var(\Psi(\tilde{\boldsymbol{Y}}_l)) + o(\sigma_l) \quad \textit{in } \|\cdot\|_{\mathrm{op}}. \tag{A11}$$

A.2 Proof of Lemma 2

Proof. For simplicity of notation, we suppress the iteration index t. Let $\tilde{\boldsymbol{Y}}_1 = \boldsymbol{b}_1 + \boldsymbol{w}_1\boldsymbol{X}$, and let $\tilde{\boldsymbol{Y}}_l = \boldsymbol{b}_l + \boldsymbol{w}_l\Psi(\boldsymbol{Y}_{l-1})$ for $l = 2,\dots,h$. By the definition of the StoNet model (3), $\boldsymbol{Y}_l$ can be written as $\boldsymbol{Y}_l = \tilde{\boldsymbol{Y}}_l + \boldsymbol{e}_l$ for $l \in \{1,2,\dots,h\}$. Additionally, we rewrite the covariance matrix of $\boldsymbol{e}_l$ as $\sigma_l^2 I_{d_l}$ by depressing its dependence on the sample size n.

To establish this lemma, we consider two scenarios: (i) the activation function $\Psi(\cdot)$ is bounded and continuously differentiable and with uniformly bounded second derivatives, such as *tanh* or *sigmoid*; (ii) the ReLU activation function.

(i) *Activation functions: tanh and sigmoid*

Write $\boldsymbol{Y}_l = \tilde{\boldsymbol{Y}}_l + \boldsymbol{e}_l$, where the noise vector $\boldsymbol{e}_l \sim N(0, \sigma_l^2 I_{d_l})$ is independent of $\tilde{\boldsymbol{Y}}_l$. As implied by Assumption 1, σ_l decays as n increases, and $d_l\sigma_l \to 0$. By Lemma A2,

$$\Psi(\boldsymbol{Y}_l) \;=\; \Psi(\tilde{\boldsymbol{Y}}_l) + \nabla_{\tilde{\boldsymbol{Y}}_l}\Psi(\tilde{\boldsymbol{Y}}_l) \circ \boldsymbol{e}_l + \boldsymbol{r}_l, \qquad \|r_l\|_2 = o_{\mathbb{P}}(\sigma_l),$$

where "$\circ$" denotes elementwise (Hadamard) product. By Lemma A3, we obtain

$$\boldsymbol{\Sigma}_l = \mathrm{Var}\big(G(\tilde{\boldsymbol{Y}}_l)\big) + \mathrm{diag}\Big\{\sigma_l^2\,\mathbb{E}[(\nabla_{\tilde{\boldsymbol{Y}}_l}\Psi(\tilde{\boldsymbol{Y}}_l)) \circ (\nabla_{\tilde{\boldsymbol{Y}}_l}\Psi(\tilde{\boldsymbol{Y}}_l))]\Big\} + o(\sigma_l^2) \quad \text{in } \|\cdot\|_{\mathrm{op}}, \tag{A12}$$

where $G(\tilde{\boldsymbol{Y}}_l) = \Psi(\tilde{\boldsymbol{Y}}_l) + \mathbb{E}[\boldsymbol{r}_l|\tilde{\boldsymbol{Y}}_l]$; and for $\boldsymbol{v} \in \mathbb{R}^{d_l}$, $\mathrm{diag}\{\boldsymbol{v}\}$ denotes the $d_l \times d_l$ diagonal matrix with diagonal entries given by $\boldsymbol{v}$.

Since the connection weights take values in a compact space Θ, there exists a constant $0 < \tau_{\max} < \infty$ such that

$$\lambda_{\max}(\boldsymbol{w}_l\boldsymbol{w}_l^T) \le \tau_{\max},$$

for any $l = 1, 2, \ldots, h+1$, where $\boldsymbol{w}_l \in \mathbb{R}^{d_l \times d_{l-1}}$ is the weight matrix of the DNN at layer l.

Let $\mathbb{Y}_l \in \mathbb{R}^{n \times d_l}$ denote imputed $\boldsymbol{Y}_l$ values for all n observations. Building on Assumption 3-(ii), by induction, we can assume that the matrix $(\Psi(\mathbb{Y}_{l-1}))^T\Psi(\mathbb{Y}_{l-1})$ is positive definite and has the eigenvalues bounded by $n\kappa_{\min}$ and $n\kappa_{\max}$. By an extension of Ostrowski's theorem, see Theorem 3.2 of Higham and Cheng (1998), the eigenvalues of the matrix $\tilde{\mathbb{Y}}_l^T\tilde{\mathbb{Y}}_l = \boldsymbol{w}_l(\Psi(\mathbb{Y}_{l-1}))^T\Psi(\mathbb{Y}_{l-1})\boldsymbol{w}_l^T$ is bounded by

$$\lambda_{\max}(\tilde{\mathbb{Y}}_l^T\tilde{\mathbb{Y}}_l) \le n\kappa_{\max}\tau_{\max}.$$

By definition,

$$\lambda_{\max}(\tilde{\mathbb{Y}}_l^T\tilde{\mathbb{Y}}_l) = \max_{\|\boldsymbol{u}\|=1} \boldsymbol{u}^T\boldsymbol{w}_l(\Psi(\mathbb{Y}_{l-1}))^T\Psi(\mathbb{Y}_{l-1})\boldsymbol{w}_l^T\boldsymbol{u}.$$

By choosing $\boldsymbol{u}$ as a one-hot vector, it is easy to see that for any $i \in \{1, 2, \ldots, d_l\}$,

$$\sum_{j=1}^{n}(\tilde{Y}_l^{(j,i)})^2 \le \lambda_{\max}(\tilde{\mathbb{Y}}_l^T\tilde{\mathbb{Y}}_l) \le n\kappa_{\max}\tau_{\max}, \tag{A13}$$

where $\tilde{Y}_l^{(j,i)}$ denotes the (j,i)th element of $\tilde{\mathbb{Y}}_l \in \mathbb{R}^{n \times d_l}$. By the compactness of Θ and the boundedness of the activation function, $\tilde{Y}_l^{j,i}$ is uniformly bounded. By the strong law of large numbers, we have $\frac{1}{n}\sum_{j=1}^{n}(\tilde{Y}_l^{(j,i)})^2 \to \mathbb{E}(\tilde{Y}_l^{(j,i)})^2$ almost surely (a.s.), as $n \to \infty$. Therefore,

$$\mathbb{E}(\tilde{Y}_l^{(j,i)})^2 \le \kappa_{\max}\tau_{\max}. \tag{A14}$$

– (i-a) *tanh activation function.* Since $\tilde{Y}_l^{(j,i)} = \sum_{k=1}^{d_{l-1}} w_{ik}\Psi(Y_{l-1}^{(j,k)})$ is a linear combination of independent random variables with finite second moments, it is reasonable to assume

that $\tilde{Y}_l^{(j,i)}$ follows a Gaussian distribution $N(\mu_{l,i}, \varsigma_{l,i}^2)$. By (A14), $\tilde{Y}_l^{(j,i)}$ has bounded second moment, and therefore both $\mu_{l,i}$ and $\varsigma_{l,i}^2$ are bounded. By the calculation in Gandhi et al. (2018),

$$\mathbb{E}(\tanh(\tilde{Y}_l^{(j,i)}))^2 \approx 1 - \frac{2}{\sqrt{4 + 2\pi\varsigma_{l,i}^2}} \exp\left(-\frac{\mu_{l,i}^2}{4/\pi + 2\varsigma_{l,i}^2}\right),$$

which implies

$$\mathbb{E}[\nabla_{\tilde{Y}_l^{(j,i)}} \Psi(\tilde{Y}_l^{(j,i)})] = 1 - \mathbb{E}(\tanh(\tilde{Y}_l^{(j,i)}))^2 \approx \frac{2}{\sqrt{4 + 2\pi\varsigma_{l,i}^2}} \exp\left(-\frac{\mu_{l,i}^2}{4/\pi + 2\varsigma_{l,i}^2}\right).$$

and, furthermore, by Cauchy–Schwarz inequality (or Jensen's inequality for the convex function $x \mapsto x^2$),

$$\mathbb{E}[\nabla_{\tilde{Y}_l^{(j,i)}} \Psi(\tilde{Y}_l^{(j,i)})]^2 \geq \frac{2}{2 + \pi\varsigma_{l,i}^2} \exp\left(-\frac{\mu_{l,i}^2}{2/\pi + \varsigma_{l,i}^2}\right).$$

By boundedness of $\mu_{l,i}$ and $\varsigma_{l,i}^2$ (as implied by (A14)), there exists a constant $c_1 > 0$ such that

$$\mathbb{E}[\nabla_{\tilde{Y}_l^{(j,i)}} \Psi(\tilde{Y}_l^{(j,i)})]^2 \geq c_1. \tag{A15}$$

Additionally, by the Lipschitz continuity of $\tanh(\cdot)$, see Assumption 1-(iii), there exists a constant c_1' such that

$$\mathbb{E}[\nabla_{\tilde{Y}_l^{(j,i)}} \Psi(\tilde{Y}_l^{(j,i)})]^2 \leq c_1'. \tag{A16}$$

By Weyl's inequality, combining (A15) with (A12), we have

$$\lambda_{\min}(\mathbf{\Sigma}_l) \geq c_1 \sigma_l^2.$$

Furthermore, combining (A16) with Lemma A1, (A11), (A12), and (A13), we have

$$\lambda_{\max}(\mathbf{\Sigma}_l) \leq \kappa_{\max}\tau_{\max} + o(\sigma_l) + c_1'\sigma_l^2.$$

(i-b) *sigmod activation function.* By calculation in Daunizeau (2017),

$$\begin{aligned}\mathbb{E}[\nabla_{\tilde{Y}_l^{(j,i)}} \Psi(\tilde{Y}_l^{(j,i)})] &= \mathbb{E}[\Psi(\tilde{Y}_l^{(j,i)})(1 - \Psi(\tilde{Y}_l^{(j,i)}))] \\ &= \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{\varsigma_{l,i}^2 + \pi^2/3}} \exp\left(-\frac{1}{2}\frac{\mu_{l,i}^2}{\varsigma_{l,i}^2 + \pi^2/3}\right),\end{aligned}$$

which implies, by Cauchy-Schwarz inequality,

$$\mathbb{E}[\nabla_{\tilde{Y}_l^{(j,i)}} \Psi(\tilde{Y}_l^{(j,i)})]^2 \geq \frac{1}{2\pi} \frac{1}{\varsigma_{l,i}^2 + \pi^2/3} \exp\left(-\frac{\mu_{l,i}^2}{\varsigma_{l,i}^2 + \pi^2/3}\right).$$

By boundedness of $\mu_{l,i}$ and $\varsigma_{l,i}^2$ (as implied by (A14)), and by the Lipschitz continuity

of the sigmoid activation function, there exist constants $c_2 > 0$ and $c_2' > 0$ such that

$$c_2 \le \mathbb{E}[\nabla_{\tilde{Y}_l^{(j,i)}} \Psi(\tilde{Y}_l^{(j,i)})]^2 \le c_2'.$$

Combining it with (A11), (A12), (A13), and Lemma A1, we have

$$\lambda_{\min}(\mathbf{\Sigma}_l) \ge c_2 \sigma_l^2, \quad \text{and} \quad \lambda_{\max}(\mathbf{\Sigma}_l) \le \kappa_{\max}\tau_{\max} + o(\sigma_l) + c_2' \sigma_l^2,$$

by Weyl's inequality.

(ii) *ReLU activation function.*

Let $Y = \tilde{Y} + e$, where $e \sim N(0, \sigma^2)$ is independent of $\tilde{Y}$, and $\Psi(y) = y_+ := \max\{y, 0\}$. For simplicity, we suppress indices and work component-wisely. The following exact truncated–normal identities hold for $u := \tilde{Y}/\sigma$:

$$\mathbb{E}\Big[\Psi(Y) \mid \tilde{Y}\Big] = \tilde{Y}\,\Phi(u) + \sigma\,\phi(u), \tag{A17}$$

$$\mathbb{E}\Big[\Psi(Y)^2 \mid \tilde{Y}\Big] = (\tilde{Y}^2 + \sigma^2)\,\Phi(u) + \tilde{Y}\sigma\,\phi(u), \tag{A18}$$

where Φ and ϕ denote, respectively, the CDF and PDF of the standard normal distribution.

Conditional mean correction. Define

$$r(\tilde{Y}) \;:=\; \mathbb{E}\Big[\Psi(Y) \mid \tilde{Y}\Big] - \Psi(\tilde{Y}) = \sigma\,\phi(u) + \tilde{Y}\big(\Phi(u) - \mathbf{1}\{\tilde{Y} > 0\}\big). \tag{A19}$$

In what follows, we show $r(\tilde{Y})$ is nonnegative and symmetric about 0. In particular,

$$r(-\tilde{Y}) = \sigma\phi(-u) - \tilde{Y}\,\Phi(-u) = \sigma\phi(u) + \tilde{Y}\big(\Phi(u) - 1\big) = r(\tilde{Y}),$$

so r is symmetric about 0. Furthermore, since $x \mapsto x_+$ is convex, we have

$$\mathbb{E}[(\tilde{Y} + \sigma Z)_+ \mid \tilde{Y}] \ge (\mathbb{E}[\tilde{Y} + \sigma Z \mid \tilde{Y}])_+ = \tilde{Y}_+,$$

by Jensen's inequality. Therefore, $r(\tilde{Y}) \ge 0$.

To find the maximum of $r(\tilde{Y})$, we write $r(\tilde{Y}) = \sigma f(u)$ with

$$f(u) = \begin{cases} \phi(u) + u(\Phi(u) - 1), & u > 0, \\ \phi(u) + u\,\Phi(u), & u < 0. \end{cases}$$

Then, using $\phi'(u) = -u\phi(u)$ and $\Phi'(u) = \phi(u)$, we obtain

$$\frac{dr(\tilde{Y})}{d\tilde{Y}} = f'(u) = \begin{cases} \Phi(u) - 1 < 0, & \tilde{Y} > 0, \\ \Phi(u) > 0, & \tilde{Y} < 0, \end{cases}$$

so r is strictly decreasing on $(0, \infty)$ and strictly increasing on $(-\infty, 0)$. Therefore, r attains

its global maximum at $\tilde{Y} = 0$, where

$$r(0) = \sigma\,\phi(0) = \frac{\sigma}{\sqrt{2\pi}}.$$

Hence, for all $\tilde{Y} \in \mathbb{R}$, we have

$$0 \le r(\tilde{Y}) \le r(0) = \frac{\sigma}{\sqrt{2\pi}}. \tag{A20}$$

Conditional variance: scalar bounds. Let's first derive some scalar bounds for the conditional variance $\mathrm{Var}(\Psi(Y)|\tilde{Y})$. Let $\mu = \tilde{Y}$, $u = \mu/\sigma$, and write

$$m_1 := \mathbb{E}[\Psi(Y) \mid \tilde{Y}] = \mu\,\Phi(u) + \sigma\,\phi(u), \quad m_2 := \mathbb{E}[\Psi(Y)^2 \mid \tilde{Y}] = (\mu^2 + \sigma^2)\,\Phi(u) + \mu\sigma\,\phi(u).$$

Then

$$\mathrm{Var}\big(\Psi(Y) \mid \tilde{Y}\big) = m_2 - m_1^2 = \big[(\mu^2 + \sigma^2)\,\Phi(u) + \mu\sigma\,\phi(u)\big] - \big[\mu\,\Phi(u) + \sigma\,\phi(u)\big]^2.$$

Expand the square and collect terms:

$$\begin{aligned}\mathrm{Var}\big(\Psi(Y) \mid \tilde{Y}\big) &= \mu^2\Phi(u) + \sigma^2\Phi(u) + \mu\sigma\phi(u) - \mu^2\Phi(u)^2 - 2\mu\sigma\Phi(u)\phi(u) - \sigma^2\phi(u)^2 \\ &= \mu^2\big[\Phi(u) - \Phi(u)^2\big] + \sigma^2\big[\Phi(u) - \phi(u)^2\big] + \mu\sigma\big[\phi(u) - 2\Phi(u)\phi(u)\big].\end{aligned}$$

Now factor out σ^2 using $\mu = \sigma u$:

$$\begin{aligned}\mathrm{Var}\big(\Psi(Y) \mid \tilde{Y}\big) &= \sigma^2\Big\{u^2\Phi(u)\big[1 - \Phi(u)\big] + \big[\Phi(u) - \phi(u)^2\big] + u\,\phi(u)\big[1 - 2\Phi(u)\big]\Big\} \\ &:= \sigma^2 g(u),\end{aligned}$$

where

$$g(u) = \Phi(u) - \phi(u)^2 + u\,\phi(u)(1 - 2\Phi(u)) + u^2\Phi(u)(1 - \Phi(u)).$$

Taking derivative for $g(u)$ (using $\phi'(u) = -u\phi(u)$ and $\Phi'(u) = \phi(u)$) leads to

$$\frac{dg(u)}{du} = 2(1 - \Phi(u))\,(\phi(u) + u\,\Phi(u)).$$

Note that

$$\phi(u) + u\,\Phi(u) = \mathbb{E}\big[(u + Z)\,\mathbf{1}\{Z > -u\}\big] = \int_{-u}^{\infty}(u + z)\,\phi(z)\,dz \;\ge\; 0,$$

since the integrand is nonnegative on $[-u, \infty)$. Therefore, $dg(u)/du \ge 0$ for all u. That is, g is increasing on $\mathbb{R}$. It is easy to verify that $\lim_{u\to-\infty} g(u) = 0$ and $\lim_{u\to\infty} g(u) = 1$, so we conclude that $0 < g(u) < 1$ for any finite $u \in \mathbb{R}$.

Thus, for any finite $\tilde{Y}$,

$$\sigma^2 > \mathrm{Var}(\Psi(Y) \mid \tilde{Y}) = \sigma^2 g(\tilde{Y}/\sigma) > 0.$$

For the unconditional bound, write $U = \tilde{Y}/\sigma$. By monotonicity, for any threshold $t \in \mathbb{R}$,

$$\sigma^2 \geq \mathbb{E}[\mathrm{Var}(\Psi(Y) \mid \tilde{Y})] = \sigma^2\, \mathbb{E}[g(U)] \;\geq\; \sigma^2\, g(t)\, \mathbb{P}(U \geq t). \tag{A21}$$

Therefore, a strictly positive lower bound follows if there exists a finite bound $t \in (-\infty, \infty)$ with $\mathbb{P}(\tilde{Y} \geq t\sigma) > 0$. Such a finite bound t exits following from the conditions: Θ is compact by assumption 1, $\boldsymbol{X} \in [0,1]^p$ by Assumption 3, and σ_l^2 are bounded for $l = 1, 2, \ldots, h$ by Assumption 1-(v). Hence, there exists a constant $c_+ > 0$ such that

$$\mathbb{E}[\mathrm{Var}(\Psi(Y) \mid \tilde{Y})] \geq c_+ \sigma^2. \tag{A22}$$

By the law of total variance and the conditional mean correction formula (A19),

$$\begin{aligned} \mathrm{Var}(\Psi(Y)) &= \mathrm{Var}(\mathbb{E}[\Psi(Y) \mid \tilde{Y}]) + \mathbb{E}[\mathrm{Var}(\Psi(Y) \mid \tilde{Y})] \\ &= \mathrm{Var}(\Psi(\tilde{Y}) + r(\tilde{Y})) + \mathbb{E}[\mathrm{Var}(\Psi(Y) \mid \tilde{Y})]. \end{aligned} \tag{A23}$$

By (A22), we have the lower bound:

$$\mathrm{Var}(\Psi(Y)) \;\geq\; \mathbb{E}[\mathrm{Var}(\Psi(Y) \mid \tilde{Y})] \;\geq\; c_+\, \sigma^2. \tag{A24}$$

By (A23) and (A21), we have the upper bound:

$$\mathrm{Var}(\Psi(Y)) \;\leq\; \mathrm{Var}(\Psi(\tilde{Y})) + \mathrm{Var}(r(\tilde{Y})) + 2\sqrt{\mathrm{Var}(\Psi(\tilde{Y}))\,\mathrm{Var}(r(\tilde{Y}))} + \sigma^2. \tag{A25}$$

Eigenvalues of the covariance matrix at layer l. Let $\boldsymbol{\Sigma}_l = \mathrm{Cov}(\Psi(\boldsymbol{Y}_l))$. Since different components of $\boldsymbol{e}_l$ are mutually independent with variance σ_l^2, (A24) implies

$$\lambda_{\min}(\boldsymbol{\Sigma}_l) \;\geq\; c_+\, \sigma_l^2.$$

For the upper bound, (A25), together with Lemma A1 and the compactness of Θ, implies

$$\lambda_{\max}(\boldsymbol{\Sigma}_l) \;\leq\; \kappa_{\max}\tau_{\max} + v_r\, d_l + 2\sqrt{v_r d_l\, \kappa_{\max}\tau_{\max}} + \sigma_l^2, \tag{A26}$$

where $v_r := \max_i \mathrm{Var}(r(\tilde{Y}_l^{(i)})) \leq \sigma_{l-1}^2/(2\pi) = O(\sigma_{l-1}^2)$ as implied by (A20). Under Assumption 1, $\sigma_{l-1}^2 d_l \prec 1/h$, we have $v_r d_l = o(1)$, so (A26) simplifies to

$$\lambda_{\max}(\boldsymbol{\Sigma}_l) \;\leq\; \kappa_{\max}\tau_{\max} + o(1) + \sigma_l^2,$$

which concludes the proof.

□

A.3 Proof of Lemma 4

Proof. Consider the multinomial logistic regression. Let $\boldsymbol{x}^{(i)} \in \mathbb{R}^p$ denote the covariate vector of sample i, where $p \leq n$ and is allowed to increase with n. For simplicity of notation, its

dependence on n is depressed. Let $\boldsymbol{\pi}^{(i)} = (\pi_0^{(i)}, \pi_1^{(i)}, \pi_2^{(i)}, \ldots, \pi_m^{(i)})^T$ denote respective probabilities of m classes in the model, where

$$\pi_j^{(i)} = \frac{e^{\boldsymbol{\beta}_j^T \boldsymbol{x}^{(i)}}}{1 + \sum_{k=1}^m e^{\boldsymbol{\beta}_k^T \boldsymbol{x}^{(i)}}}, \quad j = 0, 1, \ldots, m,$$

by setting $\boldsymbol{\beta}_0 = \mathbf{0}$. It follows from Böhning (1992) and Nwaigwe and Rychlik (2020) that the Hessian matrix of the multinomial logistic regression is given by

$$H^{(i)} := -\nabla^2_{\vec{\boldsymbol{B}}} L^{(i)} = K^T(\Lambda_{\pi^{(i)}} - \boldsymbol{\pi}^{(i)}(\boldsymbol{\pi}^{(i)})^T)K \otimes (\boldsymbol{x}^{(i)}(\boldsymbol{x}^{(i)})^T) := A^{(i)} \otimes (\boldsymbol{x}^{(i)}(\boldsymbol{x}^{(i)})^T),$$

where $L^{(i)}$ denotes the log-likelihood function for sample i, $\otimes$ denotes the Kronecker product,

$$\Lambda_{\pi^{(i)}} = \text{diag}\{\pi_0^{(i)}, \pi_1^{(i)}, \pi_2^{(i)}, \ldots, \pi_m^{(i)}\},$$

denote a diagonal matrix, $A^{(i)} = K^T(\Lambda_{\pi^{(i)}} - \boldsymbol{\pi}^{(i)}(\boldsymbol{\pi}^{(i)})^T)K$, and $K \in \mathbb{R}^{(m+1)\times m}$ is chosen such that $K^T K = I_m$. Here we set

$$K = \begin{pmatrix} \mathbf{0}_m^T \\ I_m \end{pmatrix},$$

where $\mathbf{0}_m$ denotes an m-vector of zeros.

The Hessian matrix for a data set of sample size n is given by

$$H = \sum_{i=1}^n H^{(i)}.$$

Since the eigenvalues of $\mathbb{X}^T\mathbb{X}$ are all positive and bounded, we have

$$n\kappa_{\min} \le \sum_{i=1}^n x_{ij}^2 \le n\kappa_{\max}, \quad j = 1, 2, \ldots, p, \tag{A27}$$

which implies that x_{ij}, the (i,j)th element of $\mathbb{X}$, is bounded in the second moment as n becomes large.

Assume that $X_{ij} \sim N(\mu_j, \varsigma_n^2)$, where $\varsigma_n^2 = n^{-\alpha}$ for some $\alpha > 0$. For CDF $\Phi(u)$ of the standard Gaussian distribution, we employ the approximation formula (Hart, 1957; Eidous and Abu-Shareefa, 2020):

$$\Phi(u) = 1 - \frac{\phi(u)}{u + \sqrt{2/\pi}e^{-0.4u}}, \quad u \ge 0. \tag{A28}$$

Therefore, for some sufficiently large constant $E > 0$, by (A28),

$$P(|X_{ij} - \mu_j| > E) = \frac{2\phi(E/\varsigma_n)}{E/\varsigma_n + \sqrt{2/\pi}e^{-0.4E/\varsigma_n}},$$

and for any $\alpha > 0$,

$$\begin{aligned} &P(X_{ij} \in (\mu_j - E, \mu_j + E) : 1 \le i \le n,\ 1 \le j \le p) = (1 - P(|X_{ij} - \mu_j| > E))^{np} \\ &\approx 1 - \sqrt{\frac{2}{\pi}}\frac{n^{1+\alpha/2}p}{E}e^{-\frac{1}{2}E^2 n^\alpha} \to 1, \quad \text{as } n \to \infty. \end{aligned} \tag{A29}$$

Set

$$\nu_0 = \min_{x_j \in [\mu_j - E, \mu_j + E], 0 \le j \le p} \frac{e^{\beta_j^T \boldsymbol{x}}}{1 + \sum_{k=1}^m e^{\beta_k^T \boldsymbol{x}}}, \tag{A30}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)^T$.

By the eigenvalue property of Kronecker products, see e.g. Lemma 4 in Nwaigwe and Rychlik (2020),

$$\lambda_{\min}(H^{(i)}) \ge (\min_{0 \le j \le m} \pi_j^{(i)}) \|\boldsymbol{x}^{(i)}\|^2,$$

where $\min_{0 \le j \le m} \pi_j^{(i)}$ is a lower bound of $\lambda_{\min}(A^{(i)})$. This, combing with (A27), implies

$$\lambda_{\min}(H) \ge (\min_{1 \le i \le n, 0 \le j \le m} \pi_j^{(i)}) \sum_{i=1}^n \|\boldsymbol{x}^{(i)}\|^2 \ge n(\min_{1 \le i \le n, 0 \le j \le m} \pi_j^{(i)}) \kappa_{\min}.$$

Furthermore, by (A29) and (A30), the following inequality holds with probability converging to 1,

$$\lambda_{\min}(H) \ge n \nu_0 \kappa_{\min}.$$

Then the lemma can be concluded based on the asymptotic normality of the MLE. □

A.4 Proof of Lemma 5

Proof. For convenience, Lemma 3 and Lemma 4 are stated in terms of sample covariance matrices. However, as implied by Lemma 2, the strong law of large numbers holds for the sample covariance matrices of the regressions formed in the StoNet. Consequently, each such regression satisfies the conditions of Lemma 3 and Lemma 4 almost surely. Aggregating the l_2-errors of the coefficient estimates across all $\sum_{l=1}^{h+1} d_l$ linear/logistic regressions then completes the proof. □

A.5 Proof of Lemma 6

Proof. This theorem directly follows from Theorem 4 of Liang et al. (2018b), and it implies that the estimator $\hat{\boldsymbol{\theta}}_n^{(t)}$ is consistent when both n and t are sufficiently large. □

A.6 Proof of Theorem 1

Proof. Part (i): Lemma 6 proves that the IRO algorithm gives a constructive proof for the parameter estimation consistency for sublinear DNNs. Therefore, any other estimator

$$\widehat{\boldsymbol{\theta}}_n^* = \arg\max_{\boldsymbol{\theta}} \left\{ \frac{1}{n} \log \pi(\mathbb{Y}, \mathbb{Y}_{\text{mis}} | \mathbb{X}, \boldsymbol{\theta}) \right\}$$

is also consistent due to the uniqueness of the optimal solution (up to some loss-invariant transformations).

Part (ii): It follows from a specific setting for σ_l's: $\sigma_1 = \sigma_2 = \cdots = \sigma_h = \sigma_{\min}$, where

$$\sigma_{\min} = 1 / \sqrt{d_1 (\prod_{i=2}^h d_i^2) d_{h+1} h^{1+\delta}},$$

for some $\delta > 0$. □

A.7 Proof of Theorem 2

Proof. Poggio et al. (2017) analyze the approximation power of deep neural networks for hierarchically compositional functions whose constituent maps have bounded arity (at most s variables; e.g., $s = 2$ for a binary tree). For this class of functions, they show:

Lemma A4 (Theorem 4 of Poggio et al. (2017)). *Let $f : [0,1]^{d_0} \to \mathbb{R}$ be L-Lipschitz and admit a hierarchical compositional representation in which each constituent depends on at most s variables. Then a ReLU DNN that mirrors this compositional architecture can achieve approximation error at most ε (in ℓ^p-norm) with*

$$m = \mathcal{O}\big((d_0 - 1)\,(L/\varepsilon)^{\,s}\big)$$

hidden neurons.

Let $d_0 = O(n^\alpha)$ for some $0 < \alpha < 1$ and set $\varepsilon = n^{-(1-\alpha-\delta)/s}$ with $0 < \delta < 1 - \alpha$. Then, by Lemma A4,

$$m = \mathcal{O}\big(d_0\,\varepsilon^{-s}\big) = \mathcal{O}\big(n^\alpha\, n^{1-\alpha-\delta}\big) = \mathcal{O}\big(n^{1-\delta}\big),$$

which satisfies the structural constraint in Theorem 1(ii).

In practice, this approximation result is hard to deploy directly because the target's compositional structure is typically unknown. However, the parameter-estimation consistency in Theorem 1 implies that sublinear-width DNNs can recover this structure. To further enhance structure learning, one may impose sparsity penalties, such as Lasso (Tibshirani, 1996a), SCAD (Fan and Li, 2001), or MCP (Zhang, 2010)), in training the deep networks. See Sun and Liang (2025) for theory on sparse structure recovery for DNNs under the StoNet framework.

Consequently, for sublinear-width ReLU networks, Theorem 2-(i) follows from Lemma A4 together with Theorem 1.

For sublinear-width DNNs with smooth activations, including sigmoid and tanh, analogous guarantees hold for continuously differentiable, hierarchically compositional functions by combining Theorem 2 of Poggio et al. (2017) (omit the details) with Theorem 1. □

We refer to Liang et al. (2018a) for empirical examples where a sparsity-regularized neural network recovers the structure of a hierarchical compositional function; in particular, their Figure 4 recovers the structure of the nonlinear regression function $y = 10x_2/(1+x_1^2)+5\sin(x_3x_4)+2x_5+\epsilon$, where x_i's and ϵ are Gaussian random variables.

A.8 Proof of Theorem 3

Proof. In the proof, Montanelli and Du (2019) studied the functions in the Korobov space $\mathcal{K}^{2,p}$ (with p indicating the ℓ^p-norm) and proved the following result:

For any $0 < \varepsilon < 1$ and any function $f \in \mathcal{K}^{2,p}([0,1]^{d_0})$ that satisfies $\big\|\partial_{x_1}^2 \cdots \partial_{x_{d_0}}^2 f\big\|_\infty \le 1$, there exists a deep ReLU network on inputs $(x_1, x_2, \ldots, x_{d_0})^T \in [0,1]^{d_0}$ that approximates f to accuracy ε, with depth $\mathcal{O}(|\log_2 \varepsilon|\, \log_2 d_0)$ and the number of hidden neurons

$$m = \mathcal{O}\Big(\varepsilon^{-2}\,|\log_2 \varepsilon|^{\frac{3}{2}(d_0-1)+1}\,(d_0 - 1)\Big). \tag{A31}$$

If we set $\varepsilon = n^{-(1/2-\delta)}$ for some $0 < \delta < 1/2$ as the target approximation accuracy, then a deep ReLU network can achieve this accuracy for the target function f, provided that the network has depth $O\big((\frac{1}{2}-\delta)\log_2 n \log_2 d_0\big)$ and the number of hidden neurons

$$m \;=\; O\Big(n^{\,1-2\delta}\,(\log_2 n)^{\frac{3}{2}(d_0-1)+1}\,\big(\tfrac{1}{2}-\delta\big)^{\frac{3}{2}(d_0-1)+1}\,(d_0-1)\Big) \;=\; o\big(n^{\,1-\delta}\big), \tag{A32}$$

by noting $\big(\frac{1}{2}-\delta\big)^{\frac{3}{2}(d_0-1)+1}(d_0-1) \prec 1$ when d_0 is reasonably large. □