Deep neural network classifier for multidimensional functional data

Shuoyang Wang1 | Guanqun Cao2 | Zuofeng Shang3 | for the Alzheimer’s Disease Neuroimaging Initiative

1Department of Biostatistics, Yale University, New Haven, Connecticut, USA
2Department of Mathematics and Statistics, Auburn University, Auburn, Alabama, USA
3Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, New Jersey, USA

Correspondence
Zuofeng Shang, Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, NJ 07102, USA.
Email: zshang@njit.edu

Funding information
NSF, Grant/Award Numbers: DMS 1736470, DMS 1764280, 1821157; Simons Foundation, Grant/Award Number: 849413

Abstract
We propose a new approach, called as functional deep neural network (FDNN), for classifying multidimensional functional data. Specifically, a deep neural network is trained based on the principal components of the training data which shall be used to predict the class label of a future data function. Unlike the popular functional discriminant analysis approaches which only work for one-dimensional functional data, the proposed FDNN approach applies to general non-Gaussian multidimensional functional data. Moreover, when the log density ratio possesses a locally connected functional modular structure, we show that FDNN achieves minimax optimality. The superiority of our approach is demonstrated through both simulated and real-world datasets.

KEYWORDS
functional classification, functional data analysis, functional neural networks, Minimax excess misclassification risk, multidimensional functional data

1 | INTRODUCTION

Due to modern advanced technology, complex functional data are ubiquitous. A fundamental problem in functional data analysis is to classify a data function based on training samples. A typical one-dimensional (1D) example is the speech recognition data extracted from the TIMIT...
database, in which the training samples are digitized speech curves of American English speakers from different phoneme groups, and the task is to predict the phoneme of a new speech curve. Typical two- (2D) and three-dimensional (3D) examples include the brain imaging data extracted from Early Mild Cognitive Impairment (EMCI) or Alzheimer’s Disease (AD), in which the training samples are digitized brain images, and the task is to predict the stage of a new patient. Besides above examples, functional data classification has wide applications in various fields such as machine learning, genetics, agriculture, chemometrics and artificial intelligence (Chamroukhi & Glotin, 2012; Leng & Müller, 2006; Rossi et al., 2005; Song et al., 2016). Recent monographs (Hsing & Eubank, 2015; Kokoszka & Reimherr, 2017) provide comprehensive and general discussions on this field.

In this paper, we propose a new approach, called as functional deep neural network (FDNN), for multidimensional functional data classification. A mainstream technique in functional data classification is based on functional principal component analysis (FPCA) such as functional discriminant analysis (Adrover et al., 2004; Berrendero et al., 2018; Dai et al., 2017; Delaigle et al., 2012; Delaigle & Hall, 2012, 2013; Galeano et al., 2015; Park et al., 2020; Shin, 2008; Wang, Shang, et al., 2021). We start from FPCA to extract the functional principal components of the data functions, and then train a DNN-based classifier on these FPCs as well as their corresponding class labels. As demonstrated through numerical studies, our FDNN approach performs well in classifying complex curve or imaging data. Moreover, our FDNN has desirable theoretical properties. Intuitively, when the network architectures are suitably selected, DNN shall have large expressive power (see Petersen & Voigtlaender, 2018; Yarotsky, 2021) so that functional Bayes classifier can be accurately recovered, even though data distributions are complex. Specifically, we show that, when the log-ratio of the population densities demonstrates a locally connected functional modular structure, our FDNN is proven minimax optimal. The proposed functional modular structure is useful to overcome the infinite dimensionality of functional data, and is meaningful as demonstrated in various examples (see Section 5). Relevant modular structures have been recently adopted by researchers in nonparametric regression and classification to characterize the local behavior of the multivariate input variables, based on which DNN approaches are proven to overcome the “curse of dimensionality.” See Schmidt-Hieber (2020), Bauer and Kohler (2019), Liu et al. (2021, 2022), Wang, Cao, and Shang (2021), Li et al. (2021), Hu et al. (2020), Kim et al. (2021), and Bos and Schmidt-Hieber (2021).

The contributions of the present paper are threefold. First, to the best of our knowledge, this is the first work on proposing a classifier for multidimensional functional data. Most existing literature only aims for one-dimensional functional data classification setting. Second, the proposed FDNN classifier enjoys the minimax optimality. However, the optimality and misclassification risk of the most existing functional classifiers has never been exploited yet. The last but not least, we do not require the data function follows Gaussian process, which has been required in some literature (Berrendero et al., 2018; Galeano et al., 2015), but is often violated in practice. When data distributions are general non-Gaussian, the resulting decision boundary is often complicated which cannot be accurately recovered by most existing approaches.

The rest of this article is organized as follows. In Section 2 we review functional Bayes classifier in general setting. In Section 3, we propose FDNN classifier. In Section 4, we establish theoretical properties of FDNN under suitable technical assumptions. Section 5 provides three progressive examples to demonstrate the validity of these technical assumptions. In Section 6, performances of FDNN and its competitors are demonstrated through simulation studies. In
Section 7, we apply FDNN to speech recognition data and Alzheimer’s Disease data. Section 8 summarizes the conclusions. Technical proofs are provided in Appendix and Data S1. R programs for implementing our method are provided on GitHub; see Section 6.

2 FUNCTIONAL BAYES CLASSIFIER UNDER NON-GAUSSIANITY

In this section, we review functional Bayes classifier for binary classification. Let \( X(s), s \in S := [0, 1]^d \) be a random process with \( \int_S \mathbb{E}X(s)^2 ds < \infty \), where \( d \in \mathbb{N}^+ \), and \( Y \in \{-1, 1\} \) be a uniform random class label such that, under \( Y = k \), \( X(s) \) has unknown mean function \( \mu_k(s) \) and unknown covariance function \( \Omega_k(s, s') \), for \( s, s' \in S \). Suppose that \( \Omega_k \) satisfies a functional principal component decomposition:

\[
\Omega_k(s, s') = \sum_{j=1}^{\infty} \lambda_{kj} \psi_{kj}(s) \psi_{kj}(s'), s, s' \in S,
\]

where \( \psi_{kj}, j \geq 1 \) is an orthonormal basis of \( L^2(S) \) with respect to the usual \( L^2 \) inner product, and \( \lambda_{k1} \geq \lambda_{k2} \geq \cdots > 0 \) are nonincreasing positive eigenvalues. Notably, (1) requires the covariance functions being decomposed in terms of the same eigenfunctions, which is a common assumption in functional classification literature; see Delaigle and Hall (2012) and Dai et al. (2017). Further relaxation of this assumption is discussed in Section 4.

Under \( Y = k \), write \( X(s) = \sum_{j=1}^{\infty} \xi_j \psi_{kj}(s) \), where \( \xi_j \)'s are pairwise uncorrelated random coefficients. Let \( \xi = (\xi_1, \xi_2, \ldots) \) and \( h_k(\cdot) \) be the unknown conditional density of \( \xi \) under \( Y = k \). Define \( Q^*(\cdot) \) as the log density ratio functional between the two classes:

\[
Q^*(\xi) = \log \left( \frac{h_1(\xi)}{h_{-1}(\xi)} \right).
\]

The functional Bayes rule for classifying a data function \( X \in L^2(S) \) thus has an expression

\[
G^*(X) = \begin{cases} 
1, & Q^*(\xi) \geq 0, \\
-1, & Q^*(\xi) < 0.
\end{cases}
\]

Direct estimation of \( Q^* \) is infeasible due to the infinite dimensionality of the input. A common practice is to estimate its finite-dimensional truncation. For \( J \geq 1 \), let \( \xi_j = (\xi_1, \ldots, \xi_J)^T \) be the leading \( J \) components of \( \xi \) and \( h_k^{(J)}(\cdot) \) be the marginal density of \( \xi_j \) under \( Y = k \), for \( k = \pm 1 \). Define the truncated log density ratio

\[
Q^*_J(\xi) = \log \left( \frac{h_1^{(J)}(\xi)}{h_{-1}^{(J)}(\xi)} \right),
\]

which is the log density ratio of \( h_1^{(J)} \) to \( h_{-1}^{(J)} \). The intuition is that, when \( J \) is large, \( h_k^{(J)} \) approaches \( h_k \) so that \( Q^*_J \) is an accurate approximation of \( Q^* \). Our aim is to design an efficient method to estimate \( Q^*_J \), which will in turn estimate \( Q^* \).
3 | FDNN CLASSIFIER

Suppose we observe \( n \) i.i.d. training samples \( \{(X_i(s), Y_i) : 1 \leq i \leq n, s \in S\} \), which are independent of \( X(s), s \in S \) to be classified. For \( k = \pm 1 \), define sample covariance function

\[
\hat{\Omega}_k(s, s') = \frac{1}{n_k} \sum_{i \in I_k} (X_i(s) - \bar{X}_k(s))(X_i(s') - \bar{X}_k(s')), \quad s, s' \in S,
\]

where \( I_k \) is the collection of \( i \) such that \( Y_i = k \), \( n_k := |I_k| \) and \( \bar{X}_k(s) = \frac{1}{n_k} \sum_{i \in I_k} X_i(s) \) is the sample mean function of class \( k \). Perform Karhunen–Loéve decomposition for \( \hat{\Omega}_k \):

\[
\hat{\Omega}_k(s, s') = \sum_{j=1}^{\infty} \hat{\lambda}_j \hat{\psi}_j(s) \hat{\psi}_j(s'), \quad s, s' \in S,
\]

and write the sample data function \( X_i \), under \( Y_i = k \), as

\[
X_i(s) = \sum_{j=1}^{\infty} \hat{\xi}_{ij} \hat{\psi}_j(s), \quad i = 1, \ldots, n.
\]

Intuitively, \( \hat{\xi}^{(i)} := (\hat{\xi}_{i1}, \hat{\xi}_{i2}, \ldots) \) is an estimator of \( \xi^{(i)} := (\xi_{i1}, \xi_{i2}, \ldots) \), in which \( \xi_{ij} \) are unobservable random coefficients of \( X_i \) with respect to the population basis \( \psi_{kj} \). Hence, it is natural to design classifiers based on \( \xi^{(i)} \)’s.

Let \( \hat{\xi}_J = (\hat{\xi}_{i1}, \ldots, \hat{\xi}_{ij}, \ldots) \) be the \( J \)-dimensional truncation of \( \hat{\xi}^{(i)} \) for \( i = 1, \ldots, n \). When \( X_i \)’s are Gaussian processes, various classifiers have been proposed such as centroid classifier (Delaigle & Hall, 2012), QDA (Delaigle & Hall, 2013) and nonparametric Bayes (NB) classifier (Dai et al., 2017). When \( X_i \)’s are non-Gaussian, one major challenge is the underlying complicated form of the conditional densities \( h_1 \) and \( h_{-1} \) so that estimation of \( Q_j^* \) is typically difficult. Inspired by the rich approximation power of DNN, in this section, we propose a new classifier called FDNN (functional+DNN) that can accurately estimate Bayes classifiers even when \( h_1 \) and \( h_{-1} \) are non-Gaussian complicated.

We will train a DNN to estimate \( Q_j^* \) based on \( \hat{\xi}_J \)’s. In what follows, we will describe our method in details. Let \( \sigma \) denote the rectifier linear unit activation function, that is, \( \sigma(x) = (x)_+ \) for \( x \in \mathbb{R} \). For any real vectors \( V = (v_1, \ldots, v_L)^T \) and \( y = (y_1, \ldots, y_w)^T \), define the shift activation function \( \sigma_V(y) = (\sigma(y_1 - v_1), \ldots, \sigma(y_w - v_w))^T \). For \( L \geq 1 \), \( p = (p_1, \ldots, p_L) \in \mathbb{N}^L \), let \( F(L, J, p) \) denote the class of fully connected feedforward DNN with \( J \) inputs, \( L \) hidden layers and, for \( l = 1, \ldots, L, p_l \) nodes on the \( l \)-th hidden layer. Equivalently, any \( f \in F(L, J, p) \) has an expression

\[
f(x) = W_L \sigma_{V_L} W_{L-1} \sigma_{V_{L-1}} \cdots W_1 \sigma_{V_1} W_0 x, \quad x \in \mathbb{R}^J,
\]

where \( W_l \in \mathbb{R}^{p_{l+1} \times p_l} \), for \( l = 0, \ldots, L \), are weight matrices, \( V_l \in \mathbb{R}^{p_l} \), for \( l = 1, \ldots, L \), are shift vectors. Here we adopt the convention that \( p_0 = J \) and \( p_{L+1} = 1 \).

We consider the following class of DNN:

\[
F(L, J, p, B) = \left\{ f \in F(L, J, p) : \max_{0 \leq l \leq L} \|W_l\|_\infty \leq B, \max_{1 \leq l \leq L} \|V_l\|_\infty \leq B \right\},
\]
where \( \| \cdot \|_\infty \) denotes the maximum-entry norm of a matrix/vector or supnorm of a function, and \( B > 0 \) controls the largest weights and shifts.

Given the training data \((\xi_j^{(1)}, Y_1), \ldots, (\xi_j^{(n)}, Y_n)\), let

\[
\hat{\phi}^*(\cdot) = \arg \min_{\phi \in \mathcal{F}(L,J,p,B)} \frac{1}{n} \sum_{i=1}^{n} \phi(f(\xi_j^{(i)})Y_i),
\]

(4)

where \( \phi(x) = \max(1-x, 0) \) denotes the hinge loss. We then propose the following FDNN classifier: for \( X \in L^2(S) \),

\[
\hat{G}^{FDNN}(X) = \begin{cases} 
1, & \hat{\phi}(\xi_j) \geq 0, \\
-1, & \hat{\phi}(\xi_j) < 0.
\end{cases}
\]

(5)

In practice, we suggest the following data-splitting method for selecting \((L,J,p,B)\):

- Step 1. Randomly divide the whole sample \((\xi_j^{(i)}, Y_i)\)'s into two subsets indexed by \( I_1 \) and \( I_2 \), respectively, with about \(|I_1| = 0.8n\) and \(|I_2| = 0.2n\).
- Step 2. For each \((L,J,p,B)\), we train a DNN \( \hat{f}_{L,J,p,B} \) using (4) based on subset \( I_1 \), and then calculate the testing error based on subset \( I_2 \) as

\[
\text{err}(L,J,p,B) = \frac{1}{|I_2|} \sum_{i \in I_2} I(\hat{f}_{L,J,p,B}(\xi_j^{(i)})Y_i < 0).
\]

(6)

- Step 3. Choose \((L,J,p,B)\), possibly from a preselected set, to minimize \( \text{err}(L,J,p,B) \).

4 \quad \textbf{MINIMAX OPTIMALITY OF FDNN}

For a generic functional classifier \( \hat{G} \), its excess misclassification risk is defined as \( \mathcal{E}_h(\hat{G}) := E[R_h(\hat{G}) - R_h(G^*)] \), where \( R_h(\hat{G}) := E[h[\hat{G}(X) \neq Y]] \) is the misclassification risk of \( \hat{G} \) taken with respect to \((X,Y)\) under \( h := \langle h_1, h_{-1} \rangle \), with \( Y \) the true class label of \( X \), and \( R_h(G^*) \) is defined accordingly for the functional Bayes classifier \( G^* \). A central task is to design \( \hat{G} \) that achieves minimax excess misclassification risk (MEMR), that is,

\[
\max_{h \in \mathcal{H}} \mathcal{E}_h(\hat{G}) \approx \inf_{\hat{G}} \max_{h \in \mathcal{H}} \mathcal{E}_h(\hat{G}),
\]

(7)

where \( \mathcal{H} \) is a proper class of \( h \) to be described later and the infimum is taken over all classifiers based on training samples. Classifiers satisfying (7) are called as minimax optimal.

There is a rich literature on construction of minimax optimal classifiers when data dimension is fixed or diverging. For instance, classic nonparametric approaches, such as ones directly estimating Bayes classifier nonparametrically, are proven minimax optimal in fixed-dimension regime (Farnia & Tse, 2016; Galeano et al., 2015; Hu et al., 2020; Lecué, 2008; Mammen & Tsybakov, 1999; Mazuelas et al., 2020; Tsybakov, 2004, 2009). When data are high-dimensional Gaussian, discriminant analysis approaches are proven minimax optimal (Cai & Zhang, 2019a, 2019b). On the other hand, under functional Gaussian data, researchers have proposed various functional classifiers, including functional quadratic discriminant analysis (FQDA) (Berrendero et al., 2018; Cai & Zhang, 2019a, 2019b; Dai et al., 2017; Delaigle et al., 2012; Delaigle & Hall, 2012,
Gaëtan's Gaussianity leads to a linear or quadratic polynomial $Q^*$ which can be effectively estimated by FQDA, based on which Wang, Shang, et al. (2021) showed that FQDA is minimax optimal. It is still unclear how to design optimal functional classifiers when data are non-Gaussian, a gap that the present article attempts to close.

In this section, we will establish minimax optimality of FDNN classifier under non-Gaussian functional data. For technical convenience, assume that the two populations have common known basis, that is, $\psi_{+1}(\cdot) = \psi_{-1}(\cdot)$. Therefore, we can train FDNN classifier based on $\xi^{(i)}_{\ell} := (\xi_{i1}, ..., \xi_{in})^T$, for $i = 1, ..., n$. We will first derive an upper bound for the excess misclassification risk of our FDNN classifier, and then derive a lower bound for the MEMR which matches the above upper bound. Therefore, our FDNN is able to achieve sharp rate of MEMR. The assumption for same basis functions can be directly extended to the same eigenfunction system without restricting the exact order. Extensions to general basis are possible with more tedious technical arguments.

Before proceeding further, we introduce some technical assumptions. At high levels, our assumptions are different from those proposed under Gaussian case. For instance, in either high- or infinite-dimensional Gaussian data, it is well known that density ratio under which minimax optimality shall be established. Such arguments.

In traditional non-Gaussian multivariate data classification, a common strategy is to assume smooth density ratio and controllable noise, under which minimax optimal classifiers were proposed (see Audibert & Tsybakov, 2007; Kim et al., 2021; Mammen & Tsybakov, 1999; Tsybakov, 2004). In the functional data framework, the input variable of $Q^*$ is infinite-dimensional, hence, the above strategy no longer works. We instead propose a set of functional conditions on $Q^*$ under which minimax optimality shall be established. Such conditions are viewed as infinite-dimensional extensions of Audibert and Tsybakov (2007) and Schmidt-Hieber (2020).

For $t \geq 1$, a measurable subset $D \subset \mathbb{R}^J$ and constants $\beta, K > 0$, define

$$C^\beta(D, K) = \left\{ f : D \mapsto \mathbb{R} \mid \sum_{|\alpha'| < |\alpha|} \| \partial^\alpha f \|_\infty + \sum_{|\alpha'| = |\beta|} \sup_{x, x' \in D, x \neq x'} \frac{\| \partial^\alpha f(x) - \partial^\alpha f(x') \|_\infty}{\| x - x' \|^{\beta - |\beta|}} \leq K \right\},$$

where $\partial^\alpha = \partial^{\alpha_1} ... \partial^{\alpha_t}$ denotes the partial differential operator with multi-index $\alpha = (\alpha_1, ..., \alpha_t) \in \mathbb{N}_1^t$, $|\alpha| = \alpha_1 + \cdots + \alpha_t$. Equivalently, $C^\beta(D, K)$ is the ball of $\beta$-Hölder smooth functions on $D$ with radius $K$. A function $f : \mathbb{R}^J \to \mathbb{R}$ is said to be locally $\beta$-Hölder smooth if for any $a, b \in \mathbb{R}$, there exists a constant $K$ (possibly depending on $a, b$) such that $f \in C^\beta([a, b]^J, K)$.

For $q \geq 0, J \geq 1$, let $d_0 = J$ and $d_{q+1} = 1$. For $d = (d_1, ..., d_q) \in \mathbb{N}_+^q$, $t = (t_0, ..., t_q) \in \mathbb{N}_+^{q+1}$ with $t_u \leq d_u$ for $u = 0, ..., q$, $\beta := (\beta_0, ..., \beta_q) \in \mathbb{R}^{q+1}_+$, let $\mathcal{G}(q, J, d, t, \beta)$ be the class of functions $g$ satisfying a modular expression

$$g(x) = g_q \circ \cdots \circ g_0(x), \quad \forall x \in \mathbb{R}^{d_0},$$

(8)
where $g_u = (g_{u1}, \ldots, g_{ud_{a+1}}) : \mathbb{R}^{d_u} \mapsto \mathbb{R}^{d_{a+1}}$ and $g_{av} : \mathbb{R}^{d_u} \mapsto \mathbb{R}$ are locally $\beta_u$-Hölder smooth. The $d_u$ arguments of $g_u$ are locally connected in the sense that each component $g_{av}$ only relies on $t_u(\leq d_u)$ arguments. Similar structures have been considered by Schmidt-Hieber (2020), Bauer and Kohler (2019), Liu et al. (2021, 2022), Wang, Cao, and Shang (2021), Li et al. (2021), Hu et al. (2020), Kim et al. (2021) in multivariate regression or classification to overcome high-dimensionality. Generalized additive model (Hastie & Tibshirani, 1990) and tensor product space ANOVA model (Lin, 2000) are special cases; see Liu et al. (2021).

Let $H^* \equiv H^*(q, d, t, \beta)$ be the class of population densities $h = \{h_1, h_{-1}\}$ of $\xi$ such that, for any $J \geq 1$, $Q^*_j \in \mathcal{G}(q, J, d, t, \beta)$. Equivalently, for any $h \in H^*$ and $J \geq 1$, the corresponding truncated log density ratio $Q^*_j$ has a modular structure (8) with certain smoothness. Although $Q^*_j$ has $J$ arguments, it involves at most $t_0d_1$ effective arguments, implying that the two population densities differ by a small number of variables. Relevant conditions are necessary for high-dimensional classification. For instance, in high-dimensional Gaussian data classification (Cai & Zhang, 2019a, 2019b) shows that to consistently estimate Bayes classifier, it is necessary that the mean vectors differ at a small number of components. The modular structure holds for arbitrary $J$, which may be viewed as a functional extension of Schmidt-Hieber (2020). Note that the density class $H^*$ covers many popular models studied in literature, either Gaussian or non-Gaussian; see Section 5. Moreover, we introduce the following regularity conditions on $Q^*$.

**Assumption 1.** (Functional Tsybakov noise condition) There exist constants $C > 0$ and $a \geq 0$ such that

$$P\left(\frac{1 - \exp\{-Q^*(\xi)\}}{1 + \exp\{-Q^*(\xi)\}} \leq x\right) \leq Cx^a, \forall x > 0. \quad (9)$$

**Assumption 2.** (Approximation error of $Q^*_j$) There exist a constant $J_0 \geq 1$ and decreasing functions $e(\cdot) : [1, \infty) \mapsto \mathbb{R}_+$ and $\Gamma(\cdot) : [0, \infty) \mapsto \mathbb{R}_+$, with $\sup_{J \geq 1} J^\rho e(J) < \infty$ for some $\rho > 0$ and $\int_0^\infty \Gamma(x)dx < \infty$, such that for any $J \geq J_0$ and $x > 0$,

$$P\left(|Q^*(\xi) - Q^*_j(\xi_j)| \geq x\right) \leq e(J)\Gamma(x). \quad (10)$$

Assumption 1 characterizes the discrepancy between $Q^*$ and random guess. Specifically, it requires that the probability of $Q^*$ close to 0 by $x$ is upper bounded by an order $x^a$. Assumption 1 is a functional extension of the classic Tsybakov noise condition, which is necessary in establishing minimax classification in multivariate case (Mammen & Tsybakov, 1999; Tsybakov, 2004). Assumption 2 provides an upper bound on the probability of $Q^*$ differing from $Q^*_j$ by at least $x$, which approaches zero if either $J$ or $x$ tends to infinity, implying that $Q^*_j$ is an accurate approximation of $Q^*$. Both assumptions can be verified in several concrete examples included in Section 5.

Our MEMR results will be based on the following class of population densities of $\xi$:

$$H \equiv H(q, d, t, \beta, \alpha, C, e(\cdot), \Gamma(\cdot)) = \{h \in H^* : Q^* \text{ satisfies both Assumptions 1 and 2}\}.$$  

Finally, we introduce an assumption on the orders of $(L, J, p, B)$, under which the exact rate of MEMR shall be established. Let

$$S_0 = \min_{0 \leq u \leq q} \frac{\beta_u(\alpha + 1)}{\beta_u(\alpha + 2) + t_u}, S_1 = \max_{0 \leq u \leq q} \frac{t_u}{\beta_u(\alpha + 2) + t_u}, S_2 = \min_{0 \leq u \leq q} \frac{1}{\beta_u(\alpha + 2) + t_u},$$

where $\beta_u^* := \beta_u \prod_{k=u+1}^q (\beta_k \wedge 1)$. 


Assumption 3. The DNN class \( F(L, J, p, B) \) satisfies

(a) \( L \geq \log n; \)
(b) \( (n \log^{-3} n)^{S_0/p} \leq J \leq (n \log^{-3} n)^{S_1}; \)
(c) \( \max_{1 \leq \ell \leq L} p_\ell \approx (n \log^{-3} n)^{S_1}; \)
(d) \( B \approx (n \log^{-3} n)^{S_2}. \)

Assumption 3(a), 3(c), and 3(d) provide exact orders on \( L, p, B, \) respectively. Assumption 3(b) provides a range on \( J. \) Notably, this condition implies \( \rho \geq S_0/S_1, \) that is, the function \( \epsilon(J) \) rapidly converges to zero when \( J \to \infty. \)

Theorem 1. There exist positive constants \( C_1, C_2, \) depending on \( q, d, t, \beta, \alpha, C, \epsilon(\cdot), \) and \( \Gamma(\cdot), \) such that the following results hold:

(i) \( \inf_{\hat{G}} \sup_{h \in H} E_h(\hat{G}) \geq C_1 n^{-S_0}, \) where the infimum is taken over all classifiers \( \hat{G} \) based on training samples;
(ii) under Assumption 3, it holds that

\[
\sup_{h \in H} E_h(\hat{G}^{FDNN}) \leq C_2 \left( \frac{\log^3 n}{n} \right)^{S_0}.
\]

Theorem 1 establishes a nonasymptotic rate for the MEMR which is of order \( n^{-S_0}. \) Moreover, the proposed FDNN classifier is able to achieve this rate up to a logarithmic factor, and hence, is minimax optimal. Since \( S_0 \) involves the intrinsic dimensions \( t_0 \)'s rather than the original dimensions \( d_u \)'s, the rate of MEMR is typically fast, demonstrating the theoretical advantage of our FDNN classifier.

5 | EXAMPLES

The minimax results in Section 4 are based on parameter space \( H. \) In this section, we provide some concrete examples to demonstrate the validity of such space.

5.1 | Gaussian functional data with independent coefficients

Suppose that, under \( Y = k, \) the random coefficients \( \xi_j \) are independent Gaussian with mean \( \mu_{kj} \) and variance \( \lambda_{kj}. \) Define \( M = \{ j : \mu_{kj} \neq \mu_{-kj} \} \) and \( N = \{ j : \lambda_{kj} \neq \lambda_{-kj} \}. \) Assume that \( M, N \) are mutually disjoint with common cardinality \( \omega. \) It can be shown that, for any \( J \geq J_0 := \max M \cup N, \) \( Q^*_j(\xi_j) = g_1(0, \xi_j), \) where \( g_0 \) has components \( g_{0j}(\xi_j) = a_j \xi_j^2 + b_j \xi_j + c_j \) for some constants \( a_j, b_j, c_j \) depending on \( \mu_{1j}, \mu_{-1j}, \lambda_{1j}, \lambda_{-1j}, \) and \( g_1(0, \xi_j)) = \sum_{j \in M \cup N} g_{0j}(\xi_j). \) Clearly, \( d_0 = J \) and \( t_0 = 1, \) and \( M \cup N \) has cardinality \( 2\omega, d_1 = t_1 = 2\omega. \) So \( Q^*_j \in \mathcal{G}(1, J, 2\omega, 1, 2\omega), \beta \) for any \( \beta > 0. \) Meanwhile, since \( Q^*_j = 0 \) for all \( J \geq J_0, \) and for any function \( \epsilon(h) \) with exponential tails and any density \( \Gamma(h), \) Assumption 1 holds for \( \alpha = 1 \) and Assumption 2 holds for \( J_0. \)

5.2 | Student’s \( t \) functional data with independent coefficients

Suppose that, under \( Y = k, \) \( \xi_j \) are independent Student’s \( t \) variables \( v_{kj}, \) where \( v_{kj} \geq 1 \) are degrees of freedom of the \( t \) variables. Define \( M = \{ j : \mu_{kj} \neq \mu_{-kj} \} \) whose cardinality is \( \omega. \) It can be shown
that, for any \( J \geq J_0 := \max M \cup N, Q^*_f(\xi_j) = g_1(g_0(\xi_j)) \), where \( g_0 \) has components
\[
g_{0j}(\xi_j) = \log e_j - \frac{v_{ij} + 1}{2} \log \left( 1 + \frac{e_{ij}^2}{v_{ij}} \right) + \frac{v_{-ij} + 1}{2} \log \left( 1 + \frac{e_{ij}^2}{v_{-ij}} \right),
\]
for some constant \( e_j \) depending on \( v_{kj} \), and \( g_1(g_0(\xi_j)) = \sum_{j \in M \cup N} g_{0j}(\xi_j) \). Similar to Section 5.1, we have \( Q^*_f \in \mathcal{G}(1,J,2\omega,(1,2\omega),\beta) \) for any \( \beta > 0 \). Assumptions 1 and 2 can be similarly verified as well.

### 5.3 Student’s t functional data with dependent coefficients

We consider an extension of Section 5.2 which involves dependent coefficients. Let \( p \geq 1 \) and \( v \geq 2 \) be integers. Suppose that, under \( Y = k, \zeta_j := (\xi_j, \xi_{j+1}, \ldots, \xi_{j+p-1})^T, j = 1, p + 1, 2p + 1, \ldots \) are independent multivariate Student’s t vectors following \( t_v(\mu_{kj}, \Sigma_{kj}) \), where \( \mu_{kj} \in \mathbb{R}^p \) and positive definite \( \Sigma_{kj} \) is \( p \times p \) positive definite. Define \( M = \{ j : \mu_{kj} \neq \mu_{-kj} \} \) and \( N = \{ j : \Sigma_{kj} \neq \Sigma_{-kj} \} \). Assume sets \( M \) and \( N \) are mutually disjoint with common cardinality \( \omega \). For any \( J \geq J_0 := J_0 := \max M \cup N + p - 1 \), then it can be shown that
\[
Q^*_f(\xi_j) = \sum_{j \in M \cup N} \left\{ \frac{1}{2} \log \left( \frac{|\Sigma_{-kj}|}{|\Sigma_{kj}|} \right)^{1/2} + \frac{v + p}{2} \log \left( \frac{1 + v^{-1}(\xi_j - \mu_{-kj})^T \Sigma_{-kj}^{-1}(\xi_j - \mu_{-kj})}{1 + v^{-1}(\xi_j - \mu_{kj})^T \Sigma_{kj}^{-1}(\xi_j - \mu_{kj})} \right) \right\}.
\]
Note that there are \( 2\omega \) terms in the above sum. Similar to Sections 5.1, we have \( Q^*_f \in \mathcal{G}(1,J,2\omega,(1,2\omega),\beta) \) for any \( \beta > 0 \). Assumptions 1 and 2 can be similarly verified as well.

### 6 SIMULATION STUDY

In this section, we examine the performances of FDNN and two competitors, quadratic discriminant method (QD) proposed in Delaigle and Hall (2012, 2013), and the NB classifier proposed in Dai et al. (2017), through simulation studies. Our studies involve both \( d = 1 \) and \( d = 2 \), corresponding to 1D and 2D functional data, respectively. All experiments are conducted in \( \textit{R} \). We summarize R codes and examples for the proposed FDNN algorithms on \texttt{GitHub} (https://github.com/FDASTATAUBURN/fdnn-classification).

For 1D functional data, we considered two data generation processes (DGP).

- **DGP1**: Generate \( X(s) = \sum_{j=1}^{s^2} \xi_j \psi_j(s), s \in [0,1] \), where \( \psi_1(s) = \log(s + 2), \psi_2(s) = s \) and \( \psi_3(s) = s^3 \). Under class \( k \), generate independently \( (\xi_1, \xi_2, \xi_3) \sim N(\mu_k, \Sigma_k) \), where \( \mu_1 = (-1,2,-3)^T, \Sigma_1^{1/2} = \text{diag}\left( \frac{3}{5}, \frac{2}{5}, \frac{1}{5} \right) \), \( \mu_{-1} = \left( -\frac{1}{2}, \frac{5}{2}, -\frac{5}{2} \right)^T \), \( \Sigma_{-1}^{1/2} = \text{diag}\left( \frac{9}{10}, \frac{1}{2}, \frac{3}{10} \right) \).

- **DGP2**: Generate \( X(s) = \sum_{j=1}^{s^3} \xi_j \psi_j(s), s \in [0,1] \), where \( \psi_j(s)’s \) are the same as in DGP1. Under class 1, generate independently \( (\xi_1, \xi_2, \xi_3) \sim N(\mu_1, \Sigma_1) \), where \( \mu_1 = (-1,2,-3)^T, \Sigma_1^{1/2} = \text{diag}(3,2,1) \); under class -1, generate independently \( \xi_j \sim t_{\gamma-2j}, j = 1, 2, 3 \).
TABLE 1 Misclassification rates (%) with standard errors in brackets for DGP1 and DGP2.

| n  | FDNN   | QD     | NB     | FDNN   | QD     | NB     |
|----|--------|--------|--------|--------|--------|--------|
| 40 | 31.76(0.10) | 38.58(0.02) | 38.33(0.02) | 16.69(0.04) | 39.99(0.01) | 39.26(0.03) |
| 100| 18.82(0.10)  | 37.91(0.02)  | 41.03(0.02)  | 13.20(0.01)  | 38.42(0.09)  | 40.27(0.03)  |
| 200| 13.19(0.10)  | 37.35(0.02)  | 39.92(0.02)  | 12.29(0.01)  | 42.63(0.02)  | 39.84(0.04)  |
| 400| 9.62(0.04)   | 36.75(0.02)  | 38.54(0.02)  | 12.40(0.01)  | 43.98(0.09)  | 38.51(0.04)  |

Abbreviations: FDNN, functional deep neural network; NB, nonparametric Bayes; QD, quadratic discriminant method.

TABLE 2 Misclassification rates (%) with standard errors in brackets for DGP3 and DGP4.

| n  | FDNN       | KNN       | FDNN       | KNN       |
|----|------------|-----------|------------|-----------|
| 40 | 17.01(0.07) | 31.80(0.04) | 13.95(0.06) | 17.30(0.03) |
| 100| 14.80(0.06) | 31.49(0.02) | 12.73(0.04) | 17.09(0.02) |
| 200| 13.92(0.05) | 31.04(0.02) | 12.71(0.01) | 17.42(0.02) |
| 400| 12.74(0.04) | 30.68(0.02) | 12.30(0.01) | 17.24(0.01) |

Abbreviations: FDNN, functional deep neural network; KNN, k-nearest neighborhood.

For 2D functional data, we considered two DGPs:

- **DGP3**: Generate $X(s_1, s_2) = \sum_{j=1}^{4} \xi_j \psi_j(s_1, s_2), 0 \leq s_1, s_2 \leq 1$, where $\psi_1(s_1, s_2) = s_1 s_2, \psi_2(s_1, s_2) = s_1^2 s_2^2, \psi_3(s_1, s_2) = s_1 s_2^3, \psi_4(s_1, s_2) = s_1^2 s_2^2$. Under class $k$, generate independently $(\xi_1, \xi_2, \xi_3, \xi_4)^T \sim N(\mu_k, \Sigma_k)$, where $\mu_1 = (8, -6, 4, -2)^T, \Sigma_1^{1/2} = \text{diag}(8, 6, 4, 2), \mu_{-1} = \left(-\frac{7}{2}, -\frac{5}{2}, \frac{3}{2}, -\frac{1}{2}\right)^T, \Sigma_{-1}^{1/2} = \text{diag}\left(\frac{9}{2}, \frac{7}{2}, \frac{5}{2}, \frac{3}{2}\right)$.

- **DGP4**: Generate $X(s_1, s_2) = \sum_{j=1}^{4} \xi_j \psi_j(s_1, s_2), 0 \leq s_1, s_2 \leq 1$, where $\psi_j(s_1, s_2)$’s are the same as in DGP3. For $j = 1, 2, 3, 4$, under class 1, generate independently $\xi_j \sim t_2(0)$; under class $-1$, generate independently $\xi_j \sim t_{2j+1}(\mu_j)$, with noncentral parameter $\mu_1 = 2, \mu_2 = \frac{3}{2}, \mu_3 = 1, \mu_4 = \frac{1}{2}$.

In each DGP, we generated $n$ training data functions and 500 testing data functions for each class, with $n = 20, 50, 100, 200$. Each 1D data function was sampled over 50 grid points, and 2D data function was sampled over $30 \times 30$ grid points in the respective domain. Misclassification rates were evaluated based on 100 replicated datasets. We apply Fourier basis (1D) and product of Fourier basis (2D) to extract the projection scores owing to its orthogonality and convenience. Network parameters were selected based on training data using Steps 1–3 in Section 3. Tables 1 and 2 summarize the misclassification rates with SDs for 1D and 2D functional data, respectively. Specifically, the proposed FDNN outperforms QDA and NB in all settings. Though FDNN has larger SD than QD and NB, the values decrease when $n$ becomes large. We observe that the misclassification risk decreases as the sample size increase for both Gaussian and non-Gaussian functional data. As suggested by one referee, we compare our method with a modified $k$-nearest neighborhoods (KNN) classifier introduced in Cai and Wei (2021). We first take the projection scores of 2D
TABLE 3 Misclassification rates (%) with SEs in brackets for DGP3 and DGP4 with the functional deep neural network classifier.

|     | n   |       |       |       |
|-----|-----|-------|-------|-------|
|     | 40  | 100   | 200   | 400   |
| DGP3| 0.170 (0.066) | 0.148 (0.055) | 0.139 (0.054) | 0.127 (0.040) |
| DGP4| 0.139 (0.055) | 0.127 (0.014) | 0.127 (0.040) | 0.123 (0.011) |

simulated data to capture the main features, which is the same first step as our FDNN classifier. Second, we apply the KNN approach (“knn” in R Package class) to the projection scores to do the classification. Table 3 below demonstrates the results of the two scenarios compared with our FDNN method. It can be found that the FDNN classifier outperforms in each case, especially for DGP3, where our misclassification risks are only one third of KNN’s when sample size is 400.

7 | **REAL DATA ILLUSTRATIONS**

7.1 | **TIMIT database**

This benchmark data example was extracted from the TIMIT database (https://catalog.ldc.upenn.edu/LDC93s1), which is a widely used resource for research in speech recognition and functional data classification. The dataset we used was constructed by selecting four phonemes for classification based on digitized speech from this database. From each speech frame, a log-periodogram transformation is applied so as to cast the speech data in a form suitable for speech recognition. The five phonemes in this dataset are transcribed as follows: “sh” as in “she,” “dcl” as in “dark,” “iy” as the vowel in “she,” “aa” as the vowel in “dark,” and “ao” as the first vowel in “water.” For illustration purpose, we focus on the “aa,” “ao,” “iy,” and “dcl” phoneme classes. Each speech frame is represented by $n = 400$ samples at a 16-kHz sampling rate; the first $M = 150$ frequencies from each subject are retained. Figure 1 displays 10 log-periodograms for each class phoneme.

We randomly select training sample size $n_1 = n_2 = 100$ to train the three classifiers and the rest of 300 samples remained as the test samples. Fourier basis is applied to extract the projection scores, and network parameters were selected based on training data using Steps 1–3 in Section 3. Table 4 reports the mean percentage (averaged over the 100 repetitions) of misclassified test curves. Overall, FDNN outperformed QD and NB in all the four classification tasks. As depicted in the Figure 1, “aa” and “ao” phoneme class trajectories looks extremely similar to each other, the misclassification rate is fairly larger than other classification results. The proposed classifiers FDNN still provides smallest risks and smaller standard errors compared with QD and NB classifiers. For “ao” versus “iy,” the misclassification rate of FDNN is less than one third of that of QD; For “ao” versus “dcl,” the misclassification rate of FDNN is nearly half of that of NB.

7.2 | **ADNI database**

The dataset used in the preparation of this article were obtained from the ADNI database (http://adni.loni.usc.edu). The ADNI is a longitudinal multicenter study designed to develop clinical, imaging, genetic, and biochemical biomarkers for the early detection and tracking of AD. From this database, we collect PET data from 79 patients in AD group, and 45 patients in EMCI group.
This PET dataset has been spatially normalized and post-processed. These AD patients have three to six times doctor visits and we select the PET scans obtained in the third visits. People in EMCI group only have the second visit, and we select the PET scans obtained in the second visits. For AD group, patients’ age ranges from 59 to 88 and average age is 76.49, and there are 33 females and 46 males among these 79 subjects. For EMCI group, patients’ age ranges from 57 to 89 and average age is 72.33, and there are 26 females and 19 males among these 45 subjects. All scans were reoriented into $79 \times 95 \times 68$ voxels, which means each patient has 68 sliced 2D images with $79 \times 95$ pixels. For 2D case, it means each subject has $N = 79 \times 95 = 7,505$ observed pixels for each selected image slice. For 3D case, the observed number of voxels for each patient’s brain sample is $N = 79 \times 95 \times 68$.  

**TABLE 4** Misclassification rates (%) with SEs in brackets for Speech Recognition data. 

| Classes          | FDNN       | QD          | NB          |
|------------------|------------|-------------|-------------|
| “aa” versus “ao”| 20.744 (0.016) | 25.402 (0.026) | 25.378 (0.021) |
| “aa” versus “iy”| 0.193 (0.002)   | 0.288 (0.005)   | 0.273 (0.006)   |
| “ao” versus “iy”| 0.183 (0.004)   | 0.578 (0.005)   | 0.232 (0.005)   |
| “ao” versus “dcl”| 0.229 (0.002)   | 0.391 (0.005)   | 0.472 (0.006)   |

Abbreviations: FDNN, functional deep neural network; NB, nonparametric Bayes; QD, quadratic discriminant method.
FIGURE 2  Averaged images of the 5th, the 10th, the 15th, the 20th, and the 25th slices of Early Mild Cognitive Impairment (left column) group and Alzheimer’s Disease group (right column).
1680

FIGURE 3 Grouped boxplot of misclassification rates for the 5th, the 10th, the 15th, the 20th, the 25th slices and three-dimensional data of the first 25 slices between Early Mild Cognitive Impairment and Alzheimer’s Disease groups.

It is well known that AD destroys neurons and their connections in hippocampus, the entorhinal cortex, and the cerebral cortex. These parts are corresponding to the first 25 slices. Therefore, for our 2D case study, we specifically select the 5th, 10th, 15th, 20th, and 25th slices from 68 slices for each patient. We aim to conduct classification based on the information of those slices respectively; see Mu and Gage (2011). Figure 2 shows the averaged 2D images for two groups at each slice. For 3D case, we focus on the total 25 slices, so the 3D data is observed on $79 \times 95 \times 25$ points. Figure 3 demonstrates the misclassification rates for both 2D and 3D brain imaging data. There are several interesting finds. First, given a single slice 2D imaging data, the misclassification rates tend to be larger than using total 25 slices data (3D data). It indicates that 3D data contains more helpful information to decrease the misclassification risk. Second, the 20th slice provides the lowest one among all 2D data. It is a promising finding for neurologists, as this smallest risk indicates this particular slice presents useful information to distinguish the EMCI and AD groups. Further medical checkups are meaningful for this special location in the brain.

8 | CONCLUSION

We propose a new FDNN classifier for classifying non-Gaussian complex function data. Our contributions are twofold. First, we establish sharp convergence rates for MEMR when data are of functional type, and the result can be applied to a large scope of functional data with complex density functions. The proposed FDNN is able to attain the sharp rate. Second, our FDNN classifier is able to handle various 1D or multi-dimensional complex functional data. As demonstrated through extensive simulated and real-data examples, the proposed FDNN classifier has outstanding performances in both Gaussian and non-Gaussian settings.

ACKNOWLEDGMENTS

We thank an Associate Editor and two referees for their constructive comments which have helped to improve the presentation of the paper. Data used in preparation of this article were
obtained from the Alzheimers Disease Neuroimaging Initiative (ADNI) database (https://adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

**FUNDING INFORMATION**

Shuoyang Wang and Guanqun Cao’s research were partially supported by NSF award DMS 1736470. Guanqun Cao’s research was also partially supported by Simons Foundation under Grant number 849413. Zuofeng Shang’s research was supported in part by NSF DMS 1764280 and 1821157.

**ORCID**

Zuofeng Shang  
https://orcid.org/0000-0003-1125-2302

**REFERENCES**

Adrover, J., Salibian-Barrera, M., & Zamar, R. (2004). Globally robust inference for the location and simple linear regression models. *Journal of Statistical Planning and Inference, 119*, 353–375.

Audibert, J.-Y., & Tsybakov, A. B. (2007). Fast learning rates for plug-in classifiers. *The Annals of Statistics, 35*, 608–633.

Bauer, B., & Kohler, M. (2019). On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *The Annals of Statistics, 47*, 2261–2285.

Berrendero, J. R., Cuevas, A., & Torrecilla, J. L. (2018). On the use of reproducing kernel hilbert spaces in functional classification. *Journal of the American Statistical Association, 113*, 1210–1218.

Bos, T., & Schmidt-Hieber, J. (2021). Convergence rates of deep relu networks for multiclass classification. *arXiv:2108.00969*.

Cai, T. T., & Wei, H. (2021). Transfer learning for nonparametric classification: Minimax rate and adaptive classifier. *Annals of Statistics, 49*, 100–128.

Cai, T. T., & Zhang, L. (2019a). A convex optimization approach to high-dimensional sparse quadratic discriminant analysis. *arXiv:1912.02872*.

Cai, T. T., & Zhang, L. (2019b). High dimensional linear discriminant analysis: Optimality, adaptive algorithm and missing data. *Journal of the Royal Statistical Society. Series B. Statistical Methodology, 81*, 675–705.

Chamroukhi, F., & Glotin, H. (2012). *Mixture model-based functional discriminant analysis for curve classification* [Conference presentation]. Proceedings of the International Joint Conference on Neural Networks (IJCNN), 1–8.

Dai, X., Müller, H.-G., & Yao, F. (2017). Optimal Bayes classifiers for functional data and density ratios. *Biometrika, 104*, 545–560.

Delaigle, A., & Hall, P. (2012). Achieving near-perfect classification for functional data. *Journal of the Royal Statistical Society. Series B, 74*, 267–286.

Delaigle, A., & Hall, P. (2013). Classification using censored functional data. *Journal of the American Statistical Association, 108*, 1269–1283.

Delaigle, A., Hall, P., & Bathia, N. (2012). Componentwise classification and clustering of functional data. *Biometrika, 99*, 299–313.

Farnia, F., & Tse, D. (2016). *A minimax approach to supervised learning*. In *Proceedings of the 30th international conference on neural information processing systems, NIPS’16* (pp. 4240–4248). Red Hook.

Galeano, P., Joseph, E., & Lillo, R. E. (2015). The Mahalanobis distance for functional data with applications to classification. *Technometrics, 57*, 281–291.

Hastie, T. J., & Tibshirani, R. J. (1990). *Generalized additive models*. Chapman & Hall/CRC.

Hsing, T., & Eubank, R. (2015). *Theoretical foundations of functional data analysis, with an introduction to linear operators Wiley series in probability and statistics*. John Wiley & Sons, Ltd.
Hu, T., Shang, Z., & Cheng, G. (2020). Sharp rate of convergence for deep neural network classifiers under the teacher-student setting. *arXiv:2001.06892*.

Kim, Y., Ohn, I., & Kim, D. (2021). Fast convergence rates of deep neural networks for classification. *Neural Networks, 138*, 179–197.

Kokoszka, P., & Reimherr, M. (2017). *Introduction to functional data analysis Texts in statistical science series*. CRC Press.

Lecué, G. (2008). Classification with minimax fast rates for classes of Bayes rules with sparse representation. *Electronic Journal of Statistics, 2*, 741–773.

Leng, X., & Müller, H. (2006). Classification using functional data analysis for temporal gene expression data. *Bioinformatics, 22*, 68–76.

Li, K., Wang, F., Liu, R., Yang, F., & Shang, Z. (2021). Calibrating multi-dimensional complex ode from noisy data via deep neural networks. *arXiv:2106.03591*.

Lin, Y. (2000). Tensor product space anova models. *The Annals of Statistics, 28*, 734–755.

Liu, R., Boukai, B., & Shang, Z. (2022). Optimal nonparametric inference via deep neural network. *Journal of Mathematical Analysis and Applications, 505*, 125561.

Liu, R., Shang, Z., & Cheng, G. (2021). On deep instrumental variables estimate. *arXiv:2004.14954*.

Mammen, E., & Tsybakov, A. B. (1999). Smooth discrimination analysis. *The Annals of Statistics, 27*, 1808–1829.

Mazuelas, S., Zanoni, A., & Perez, A. (2020). Minimax classification with 0-1 loss and performance guarantees. *arXiv:2010.07964*.

Mu, Y., & Gage, F. H. (2011). Adult hippocampal neurogenesis and its role in Alzheimer’s disease. *Molecular Neurodegeneration, 10*, 1–10.

Park, J., Ahn, J., & Jeon, Y. (2020). Sparse functional linear discriminant analysis. *arXiv:2012.06488*.

Petersen, P., & Voigtlaender, F. (2018). Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural Networks, 108*, 296–330.

Rossi, F., Nicolas, D., Conan-Guez, B., & Verleysen, M. (2005). Representation of functional data in neural networks. *Neurocomputing, 64*, 183–210.

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

Shin, H. (2008). An extension of fisher’s discriminant analysis for stochastic processes. *Journal of Multivariate Analysis, 99*, 1191–1216.

Song, J., Deng, W., Lee, H., & Kwon, D. (2016). Optimal classification for time-course gene expression data using functional data analysis. *Biometrika, 103*, 147–159.

Tsybakov, A. B. (2004). Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics, 32*, 135–166.

Tsybakov, A. B. (2009). *Introduction to nonparametric estimation Springer series in statistics*. Springer.

Wang, S., Cao, G., & Shang, Z. (2021). Estimation of the mean function of functional data via deep neural networks. *Stat, 10*, e393.

Wang, S., Shang, Z., Cao, G., & Liu, J. S. (2021). Optimal classification for functional data. *arXiv:2103.00569*.

Yarotsky, D. (2021). Error bounds for approximations with deep ReLu networks. *Neural Networks, 94*, 103–114.

**SUPPORTING INFORMATION**

Additional supporting information can be found online in the Supporting Information section at the end of this article.

**How to cite this article:** Wang, S., Cao, G., Shang, Z., & for the Alzheimer’s Disease Neuroimaging Initiative (2023). Deep neural network classifier for multidimensional functional data. *Scandinavian Journal of Statistics, 50*(4), 1667–1686. [https://doi.org/10.1111/sjos.12660](https://doi.org/10.1111/sjos.12660)
APPENDIX

This section provides the technical proof of Theorem 1. We introduce some preliminary results that will be used in the proof of Theorem 1.

For any $M > 0$, we define $\varepsilon = \max_u M^{-\beta_u}$ for simplicity. The following lemma provides an error bound for excess risk.

**Lemma 1.** There exists an $\tilde{f} \in \mathcal{F}(L, J, \mathbf{p}, B)$ and $\tilde{G} = \text{sign}(\tilde{f})$ satisfying

$$
\sup_{h \in H} E \left[ R_h(\tilde{G}) - R_h(G^*_J) \right] \lesssim \varepsilon^{a+1} + \varepsilon(J),
$$

such that $L \lesssim \log_2 M$, $\|\mathbf{p}\|_{\infty} \lesssim \max_{u=0, \ldots, q} d_{(u+1)\ell}u(M+1)^{\ell}$, $B \lesssim M$, where $G^*_J$ is the Bayes classifier for the first $J$ scores and $J_0 \leq J \lesssim \max_{u=0, \ldots, q} M^t_u$.

The following two propositions are implied by Assumption 2.

**Proposition 1.** Assumption 2 implies that

$$
|E(Q^* - Q_J)| \lesssim \varepsilon(J),
$$

for all $J \geq J_0$.

**Proposition 2.** Assumption 2 implies that

$$
P(|Q^*(\xi)| < \infty) = 1.
$$

The proofs of Lemma 1 and two propositions are provided in the Data S1.

A.1 Proof of Theorem 1(i)

Let $u^* = \arg \min_{u=0, \ldots, q} \frac{\beta_u(A+1)}{\beta_u(\alpha+2)+t_u}$, $\beta^* = \beta_{u^*}$, $t^* = t_{u^*}$, $\beta^{**} = \beta_{u^*}$ and $\tilde{\beta} = \prod_{k=u^*+1}^q (\beta_k \wedge 1)$. Without loss of generality, we assume $u^*$ is unique.

For an integer $w \geq 1$, define the regular grid on $\mathbb{R}^{t^*}$ as

$$
G_w = \left\{ \left( \frac{2k_1+1}{2w}, \ldots, \frac{2k_{t^*}+1}{2w} \right) : k_\ell \in \{0, \ldots, w-1\}, \ell = 1, \ldots, t^* \right\}.
$$

Let $n_w(x) \in G_w$ be the closest point to $x \in \mathbb{R}^{t^*}$ among points in $G_w$. Let $\mathcal{X}_{\ell}, \ell = 0, \ldots, m$ be the partition of $\mathbb{R}^{t^*}$ defined in the proof of theorem 4.1 in Audibert and Tsybakov (2007), where $m \leq w^{t^*}$.

Let $p : \mathbb{R}^+ \to \mathbb{R}^+$ be a nonincreasing infinitely differentiable function such that $p = 1$ on $[0, 1/4]$ and $p = 0$ on $[1/2, \infty)$. For instance, $p$ can be constructed as in Audibert and Tsybakov (2007): $p(x) = \left( \int_{1/4}^{1/2} p_1(s) ds \right)^{-1} \int_x^\infty p_1(t) dt$, where

$$
p_1(x) = \begin{cases} 
\exp \left\{ \frac{1}{(x-1/4)(x-1/2)} \right\}, & x \in (1/4, 1/2), \\
0, & \text{otherwise}.
\end{cases}
$$
Let $h_{u^*}: \mathbb{R}^r \to \mathbb{R}^+$ be a function defined as $h_{u^*}(x) = w^{-\rho^*} C_{\rho} p(|w(x - n_{u^*}(x))|)$, where $D^j h_{u^*}(x) = q^{[s - \rho^*]} C_{\rho} D^j p(|w(x - n_{u^*}(x))|)$ for any $s \in \mathbb{N}^r$ such that $|s| \leq [\rho^*]$, and $C_{\rho}$ is a constant small enough to ensure $h_{u^*} \in C^{\rho^*}(\mathbb{R}^r, K^*)$ for a constant $K^* > 0$. Here, we require $C_{\rho}$ being small so that $h_{u^*}$ has Lipschitz constant $K^*$.

In the following, we construct a special composition function based on $h_{u^*}$. For $\overline{\sigma} = (\sigma_1, \ldots, \sigma_m) \in \{-1, 1\}^m$, let $h(x) = \sum_{j=1}^m \sigma_j h_j(x)$ such that $h_j(x) = h_{u^*}(x) \mathbb{I} (x \in \chi_j)$). It is easy to verify that $h \in C^{\rho^*}(\mathbb{R}^r, 2K^*)$. Define the following functions

$$
\begin{align*}
g_u(x_1, \ldots, x_d_u) &= (x_1, \ldots, x_d_u), \quad u < u^*, \\
g_u(x_1, \ldots, x_d_u) &= (h(x_1, \ldots, x_r), 0, \ldots, 0), \quad u = u^*, \\
g_u(x_1, \ldots, x_d_u) &= (x_1^m, 0, \ldots, 0), \quad u > u^*.
\end{align*}
$$

Let $J^0$ be a (relatively) large integer up to $O(n^c)$ for some universal positive constant $c$. For $z \in \mathbb{R}^{J^0}$ and $x \in \mathbb{R}^r$, define $z_\sigma(z)$ as the first element of $g_{u^*} \circ g_{u^* - 1} \circ \cdots \circ g_{u^*}(z)$, and $h(x) = z_\sigma(z)$. Let

$$
\eta_\sigma(z) = \frac{1}{2} + \frac{1}{2} \sum_{j=0}^m \sigma_j h_{u^*}(x) \mathbb{I} (x \in \chi_j).
$$

For all $J \geq J^0$, let $\eta(z') = \eta_\sigma(z)$, where $z' \in \mathbb{R}^J$ and $z \in \mathbb{R}^{J^0}$ is the first $J^0$ elements of $z'$. It is easy to see that $\eta_\sigma \in \mathcal{G}(q, J^0, \mathbf{d}, \mathbf{t}, \beta)$, and Assumption 2 is satisfied for all $J \geq J^0$. According to Proposition 2, $Q^*$ is finite in probability, and it is equivalent to the mild density assumption in Audibert and Tsybakov (2007). Assumption 1 is equivalent to the margin assumption in Audibert and Tsybakov (2007), which can be justified accordingly. The rest of proof can simply follow the proof of theorem 4.1 in Audibert and Tsybakov (2007), where for any generic classifier $\hat{G}$, there exists a universal constant $C_1$, such that

$$
\sup_{h \in \mathcal{H}} E \left[ R_h(\hat{G}) - R_h(G^*) \right] \geq \sup_{h \in \mathcal{H}} E \left[ R_h(\hat{G}) - R_h(G_{J^0}^\beta) \right] \geq C_1 \left( \frac{1}{n} \right)^{S_0}.
$$

In the following, we consider the local property of the aforementioned function class $\mathcal{G}$. With a little abuse of notation, let the local version function class $\mathcal{G}(q, J, \mathbf{d}, \mathbf{t}, \beta, \mathbf{K})$ be the class of functions of the form

$$
g(x) = g_0 \circ \cdots \circ g_0(x), \quad \forall x \in [a_0, b_0]^{d_0}, \tag{A1}
$$

where for any constants $a_0, b_0 \in \mathbb{R}$, there exists $a_1, \ldots, a_{q+1}, b_1, \ldots, b_{q+1}, K_0, \ldots, K_q$, such that $g_u = (g_{u1}, \ldots, g_{ud_u}) : [a_u, b_u]^{d_u} \to [a_{u+1}, b_{u+1}]^{d_{u+1}}$, with $g_{uv} \in C^{\beta_u}_{t_u}([a_u, b_u]^{d_u}, K_u)$ being $\beta_u$-Hölder functions of radius $K_u$ involving only $t_u(\leq d_u)$ variables. Define the set of the effective inputs: $A = \{j : \xi_j \text{ is effective for } g_{0v} \text{ for all } v = 1, \ldots, d_1\}$, such that $|A| \leq t_0 d_1 < \infty$. Note that $|A|$ is finite.

**Proof of Theorem 1(ii)**

Let $B_{M_0} = \{|\xi_j| \leq M_0 \text{ for all } j \in A\}$. We first find the minimax upper bound on a bounded set when $\xi_j \in [-M_0, M_0]$ for all $j \in A$ and $M_0 > 0$. Let the $\delta$ in inequality (S.7) be $e^{a+1}$, we have
\[
(e^a+1)^{-\max_u \frac{t_u}{u(u+1)} - \frac{(e+2)}{e+1}} \log^3(e^{-1}) \leq n,
\]

which leads to
\[
e^{a+1} \geq \left( \frac{\log^3 n}{n} \right) \min_u \frac{t_u}{u(u+1)}.
\]

Together with Lemma S.11 in Data S1, the excess risk of the first J scores via DNN satisfies
\[
\sup_{h \in H} E \left[ R_h(\tilde{f}) - R_h(G^*_J) \right] \leq (n^{-1} \log^3 n)^{S_0}.
\]

The asymptotic order of \( L \), \( \max_{0 \leq \ell \leq L} p_{\ell} \) and \( B \) can be simply derived by letting \( e^{a+1} \asymp \left( \frac{\log^3 n}{n} \right) \min_u \frac{t_u}{u(u+1)} \) and applying the result in Lemma 1. Note that the input \( J \) satisfies \( J \leq \max_{1 \leq \ell \leq L} p_{\ell} \leq (n \log^{-3} n)^{S_1} \).

Next, we approximate the first J scores and the whole process. Since
\[
R_h(G^*_J) - R_h(G^*) = P(Q^* > 0) - P(Q^*_J > 0),
\]

and
\[
P(Q^* > 0) - P(Q^*_J > 0) = P(Q^* > 0) - P(Q^* > Q^* - Q^*_J) = P(Q^* > 0) - E \left[ P \left( Q_\infty > Q^* - Q^*_J \right) | Q^* - Q^*_J \right]
\]
\[
= \left( 1 - \int_{-\infty}^{0} f_{Q^*}(t) dt \right) - E \left( 1 - \int_{-\infty}^{Q^* - Q^*_J} f_{Q^*}(t) dt \right) = E \left( \int_{Q^* - Q^*_J}^{0} f_{Q^*}(t) dt \right).
\]

By considering \( h \in H \), we have \( 0 < f_{Q^*}(t) < \infty \), and
\[
E \left( \int_{0}^{Q^* - Q^*_J} f_{Q^*}(t) dt \right) \asymp E \left( Q^* - Q^*_J \right).
\]

Therefore, by Proposition 1, \( R_h(G^*_J) - R_h(G^*) = O(e(J)) \).

By Assumption 2 and Assumption 3(b), we have \( e(J) = O(J^{-\rho}) \leq (n^{-1} \log^3 n)^{S_0} \). When \( \rho > S_0/S_1 \), there always exists a constant \( C' \) and \( C'' \), such that \( C'(n \log^{-3} n)^{S_0/\rho} < C''(n \log^{-3} n)^{S_1} \), and the optimal \( J \) exists.

Therefore, for those optimal \( J \)'s, we have
\[
\sup_{h \in H} E \left[ R_h(\hat{G}^{FDNN}) - R_h(G^*) \right] \asymp \left( \frac{\log^3 n}{n} \right)^{S_0} \text{, under event } B_{M_0}.
\]

Finally, we control the minimax upper bound for \( \xi_j \in \mathbb{R} \) for all \( j \in A \). For any \( h \) and \( M_0 > 0 \), we have
\[
\mathcal{E}_h(\hat{G}) = \mathcal{E}_h(\hat{G}^{\mathbb{P}_B(B_{M_0})}) + \mathcal{E}_h(\hat{G}^{\mathbb{C}_B(B_{M_0})}) \leq \left( \frac{\log^3 n}{n} \right)^{S_0} + P(B_{M_0}^c).
It is trivial to see that, for any $e > 0$, there exists an $M > 0$, such that

$$P(B_M) = P(|\xi_j| \leq M \text{ for all } j \in \mathcal{A}) \geq 1 - e,$$

(A2)

for all $j \in \mathcal{A}$. Therefore, choose $P(B_{M_0}^c) \leq \left(\frac{\log n}{n}\right)^{S_0}$, there exists a corresponding $M_0$, to make the aforementioned asymptotic inequality hold. Choose some constant $C_2$ which depends on $q, d, t, \beta, \alpha, C, \epsilon(\cdot), \Gamma(\cdot)$, the proof is complete.