Statistica Sinica Preprint No: SS-2025-0231	
Title	Functional Imaging Data Classification with Crowdsourced Noisy Labels
Manuscript ID	SS-2025-0231
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202025.0231
Complete List of Authors	Shuoyang Wang, Grace Y Yi and Guanqun Cao
Corresponding Authors	Guanqun Cao
E-mails	caoguanq@msu.edu
Notice: Accepted author version.	

Functional Imaging Data Classification with Crowdsourced Noisy Labels

Shuoyang WANG[1], Grace Y. YI[2], and Guanqun CAO[3]

[1] *University of Louisville, Louisville, KY 40208, USA*

[2] *University of Western Ontario, London, ON N6A 3K7, Canada*

[3] *Michigan State University, East Lansing, MI 48824, USA*

Abstract: Supervised learning typically assumes access to accurate ground truth labels, but in many real-world applications, label acquisition is costly, time-consuming, and often prone to expert disagreement and annotation noise. We address this challenge in the context of functional imaging data by proposing the annotated functional deep neural network (`afDNN`), a novel classification framework for noisy annotated imaging data. By modeling images as random functions over a bounded spatial domain, we extract projection scores and use them as inputs to a sparse deep neural network to jointly estimate the ground truth label distribution and annotator-specific confusion matrices. Our method incorporates a regularized cross-entropy loss to ensure identifiability of the annotator noise structure and requires no anchor labels or prior knowledge of annotator reliability. We establish theoretical convergence guarantees for the proposed estimator. Extensive simulations and application to chest X-ray and brain imaging datasets demonstrate the effectiveness and robustness of the proposed method.

Key words and phrases: Noisy annotations, Classification, Imaging data, Functional data analysis, Label integration.

1. Introduction

The effectiveness of supervised learning methods relies on the availability of accurate ground truth labels for model training. However, obtaining such labels is typically challenging. Annotations are often provided by multiple human raters with varying levels of expertise and experiences. Blindly treating these noisy and heterogeneous labels as ground truth can greatly degrade the performance of learning algorithms, especially when annotator disagreement and systematic errors are pronounced. This issue is particularly critical in high-stakes applications such as medical imaging classification, where both annotation costs

and inter-observer variability are high.

For instance, in biomedical imaging, labeling often requires multiple experienced radiologists to annotate each patient's scans, making the process both labor-intensive and costly. One motivating example is a dataset of chest radiographs obtained from the National Institute for Occupational Safety and Health image repository, which includes scans from U.S. coal workers collected since 2000. Expert classification of such radiographs is vital for monitoring and protecting workers exposed to coal dust. However, the shortage of qualified radiologists raises concerns about the consistency and diagnostic reliability (Nishie et al., 2015). In practice, it is not uncommon for patient scans to be reviewed by radiologists with varying levels of expertise. While this variability can introduce inconsistencies, it also helps streamline the diagnostic process. With the increasing reliance on machine learning and deep learning models for imaging data classification, the sensitivity of these models to noisy labels presents a major challenge. Robust statistical approaches are essential to effectively integrate multi-source datasets and extract more precise, reproducible insights from complex imaging data. Consequently, developing methodologies that can both accurately classify imaging data and assess annotator reliability is crucial.

In this paper, we introduce a novel, theoretically grounded methodology for simultaneously addressing label noise induced from multiple annotations and estimating the ground truth label distribution in the framework of Functional Data Analysis (FDA). FDA offers effective tools for analyzing imaging data by treating observed images as random functions defined over a two-dimensional (2D) domain (Zhu et al., 2014; Wang et al., 2020; Huang and Zhu, 2022; Shieh and Ogden, 2023). This view enables a richer characterization of spatial structures inherent in image data.

To handle the infinite-dimensional nature of functional data, we extract projection scores from the observed functions (image data) and use them to learn annotators' confusion matrices (CMs), which capture individual annotator skills and the ground truth label distribution

through a deep neural network (DNN) framework. Our method effectively separates annotation noise from true labels by minimizing the cross-entropy (CE) loss. With a regularization term introduced, our approach encourages not only accurate label prediction but also characterization of annotator behavior. The formulation allows for robust recovery of both the classifier and label noise structure, even when ground labels are entirely unavailable.

1.1. Related work

The simplest yet widely used approach for integrating multiple annotators' labels is majority voting (Raykar et al., 2009; Whitehill et al., 2009; Han et al., 2019), which assumes uniform annotator reliability, a rarely valid assumption in applications (Arpit et al., 2017). To address this issue, Li and Yu (2014) proposed weighted majority voting, which assigns different weights to annotators based on estimated reliability, but this method still lacks flexibility in facilitating annotator-ground truth-specific structures in label noise.

Recent research has introduced more sophisticated techniques that estimate annotator-specific noise transition matrices, also known as CMs. These approaches typically follow a two-step pipeline: first estimating the CM and then correcting noisy labels before training classifiers (Liu et al., 2012; Zhang et al., 2016; Ibrahim and Fu, 2021; Guo et al., 2023). More recently, end-to-end methods that simultaneously learn annotators' CMs and the classifier have demonstrated significantly improved performance in practice (Chen et al., 2021; Wei et al., 2022; Ibrahim et al., 2023; Tanno et al., 2019).

Existing methods typically fall into two settings for modeling annotator CMs: instance-independent and instance-dependent. With the instance-independent assumption, for each annotator, given the true label, the corruption process is independent of the input instance. In contrast, instance-dependent annotation noise does not require this conditional independence, and thus is more realistic for real-world datasets; factors such as instance characteristics (e.g., difficulty, ambiguity, or features) and/or annotator expertise can influence how labels are assigned. While the instance-independent setting can be restrictive, it has attracted most

attention in the literature. Recently, Ibrahim et al. (2023) provided some theoretical insights into the estimation of annotators' CMs. However, their approach assumes the Near-Class Specialist condition, which requires the presence of reliable annotators for each class. Other works (Zhang et al., 2020; Guo et al., 2023) have addressed data-dependent CMs, but their method requires the existence of anchor points (the points whose input values uniquely determine their ground truth labels. Another relevant line of research involves computer vision methods for medical data analysis, where convolutional neural networks (CNNs) are commonly used for image classification. For instance, Warfield et al. (2004) proposed a label fusion method that models the reliability of individual experts and uses this information to weight their opinions during the label aggregation step. Asman and Landman (2011) introduced voxel-wise consensus to address the underestimation of annotator reliability. Asman and Landman (2012) further refined the framework by modeling annotator reliability across different pixels in images. However, these label fusion methods do not integrate information across multiple training images. Importantly, these methods lack theoretical guarantees regarding their asymptotic misclassification rates, raising concerns about their interpretability and reliability.

1.2. Contributions

Our work contributes to the literature on functional data classification, which has focused largely on one-dimensional curve data, where the observations are modeled as continuous functions over a one-dimensional domain (Delaigle and Hall, 2012; Dai et al., 2017; Berrendero et al., 2018; Park et al., 2021). Recent developments have extended these ideas to 2D functional imaging data using DNN-based classifiers (Wang et al., 2024, 2023) under the assumptions of clean labels. To the best of our knowledge, robust classification under noisy labels for functional imaging data remains an open problem.

We propose the annotated functional deep neural network (`afDNN`) method to address the problem of noisy annotations in imaging data classification. Our contributions are threefold.

First, we propose the first FDA framework for robust classification of imaging data with noisy labels, combined with projection-based functional representation with deep learning networks. Second, we rigorously establish the first convergence rate results for noisy label classification in imaging data, bridging a significant gap in the literature that largely focuses on empirical performance without theoretical guarantees. Our development provides solid theoretical foundations while demonstrating computational flexibility for both low- and high-resolution imaging data. Finally, our method does not require anchor points or prior knowledge of ground truth labels. It achieves high estimation efficiency of annotator CMs and provides interpretable insight into annotator reliability.

The rest of this article is organized as follows. In Section 2, we introduce the imaging data classification model with annotated noisy labels. We then propose the DNN-based classifier and estimation framework for CMs in Section 3. In Section 4, we establish theoretical properties of the proposed classifier with accompanying regularity conditions identified. In Section 5, we evaluate the performance of our proposed method and its competitors through simulation studies, and then further extend our framework to settings of imbalanced annotators. In Section 6, we apply our method to a chest X-ray data and the Alzheimer's Disease Neuroimaging Initiative (ADNI) database. Section 7 concludes the article with discussion. Technical proofs, additional numerical results, and the Python code for implementing our method are all provided in Supplementary Material.

2. Notation and framework

Let $X(\boldsymbol{s})$ denote the input that represents a function of $\boldsymbol{s}$, with $\boldsymbol{s} \in [0,1]^d$; let Y denote the associated label, taking a value from $[K]$, where $K \geq 2$, and $[K]$ represents $\{1, \ldots, K\}$. Suppose we observe n i.i.d. training samples of functional data $\{Y_i, X_i(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\}_{i=1}^n$, which are independent copies of $\left\{Y, X(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\right\}$ to be classified and $X(\boldsymbol{s})$ be a random process with $\int_{[0,1]^d} \mathbb{E}[\{X(\boldsymbol{s})\}^2] d\boldsymbol{s} < \infty$. Here, n, d and K are positive integers. Throughout the paper, the number of classes K is taken as a universal constant not depending on sample

size n. We are interested in using the training data to learn a classifier to label a future input.

In typical label collection processes, the true label Y_i is often not observed for each input $X_i(\boldsymbol{s})$. Instead, multiple annotators provide subjective, noisy estimates of the "truth", influenced by their varying skill levels and potential biases. Hence, there are a set of noisy crowdsourced labels $\widetilde{\boldsymbol{Y}}_i \triangleq \left\{\widetilde{Y}_i^{(1)}, \ldots, \widetilde{Y}_i^{(R)}\right\}$ from R distinct annotators, where $\widetilde{Y}_i^{(r)}$ represents the label given by the r-th annotator for $r \in [R]$. Thus, a noisy dataset $\{\widetilde{\boldsymbol{Y}}_i, X_i(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\}_{i=1}^n$ is available, where for each instance $X_i(s)$, ground truth label Y_i is unavailable. Under this setting, we aim to simultaneously estimate the annotator CMs and the ground truth label distribution $\mathbb{P}(Y = k|X(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d)$ by utilizing the noisy crowdsourced dataset, where $k \in [K]$.

Assume that R annotations are independently collected. The conditional probability of the R noisy labels, given the instance $X_i(\boldsymbol{s})$, can then be formulated as

$$\begin{aligned}
&\mathbb{P}\left(\widetilde{Y}^{(1)} = \widetilde{y}^{(1)}, \ldots, \widetilde{Y}^{(R)} = \widetilde{y}^{(R)}|X(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\right) \\
=&\prod_{r=1}^{R} \sum_{k\in[K]} \left\{\mathbb{P}\left(\widetilde{Y}^{(r)} = \widetilde{y}^{(r)}|Y = k, X(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\right) \mathbb{P}\left(Y = k|X(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\right)\right\},
\end{aligned}$$

where the conditional probability $\mathbb{P}(Y = k|X(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d)$ is called the base model, and $\mathbb{P}(\widetilde{Y}^{(r)}|Y = k, X(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d)$ is referred to as the transition model for the r-th annotator. The distribution of noisy annotation may vary with different annotators and the value of the underlying true label Y. We consider instance-independent label noise that is commonly considered in the literature of label noise (Tanno et al., 2019; Ibrahim and Fu, 2021). That is, for $r \in [R]$ and $k \in [K]$,

$$\mathbb{P}\left(\widetilde{Y}^{(r)} = j|Y = k, X(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\right) = \mathbb{P}\left(\widetilde{Y}^{(r)} = j|Y = k\right),$$

and let $a_{jk}^{(r)}$ denote this probability. Define $\boldsymbol{A}^{*(r)} = \left[a_{jk}^{(r)}\right]_{1\le j,k\le K}$ to be the $K \times K$ transition matrix, also called the CM of the r-th annotator. The values of $\boldsymbol{A}^{*(r)}$ indicate the skill level for the r-th annotator.

Define a K-dimensional vector $\boldsymbol{p}^{(r)} = \left(\mathbb{P}\left(\widetilde{Y}^{(r)} = k|X(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\right)\right)_{k\in[K]}^{\mathsf{T}}$ and the base model classifier $\boldsymbol{f}^{\circledast}\left(X(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\right) = \left(\mathbb{P}\left(Y = k|X(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\right)\right)_{k\in[K]}^{\mathsf{T}}$, where classifier $\boldsymbol{f}^{\circledast} = (f_1^{\circledast}, \ldots, f_K^{\circledast})^{\mathsf{T}}$ and $f_k^{\circledast} : L^2([0,1]^d) \to [0,1]$, $k \in [K]$. The vector $\boldsymbol{p}^{(r)}$ can be rewritten as

$$\boldsymbol{p}^{(r)} = \boldsymbol{A}^{*(r)}\boldsymbol{f}^{\circledast}\left(X(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\right), \quad r \in [R].$$

Hence, given the input information, the observed annotated value for i-th subject from r-th annotator can be viewed as a realization of a categorical random variable, with the distribution: $\widetilde{Y}_i^{(r)} \sim \text{categorical}(\boldsymbol{p}_i^{(r)})$. In the next section, we introduce our optimization algorithm for jointly estimating the true label distribution $\mathbb{P}(Y|X(\boldsymbol{s}))$ and the true CM, $\boldsymbol{A}^{*(r)}$.

3. Simultaneous estimation of labels and confusion matrices

3.1. Confusion matrices for annotators

For any $K \times 1$ vector $\boldsymbol{b}$ and $k \in [K]$, let $[\boldsymbol{b}]_k$ denote the kth element of $\boldsymbol{b}$. For $r \in [R]$, let $\widetilde{\boldsymbol{p}}^{(r)}$ denote a generic probability vector. The CE is commonly used to evaluate its discrepancy from $\boldsymbol{p}^{(r)}$: $CE(\widetilde{\boldsymbol{p}}^{(r)}, \boldsymbol{p}^{(r)}) = -\sum_{k=1}^{K}\left[\widetilde{\boldsymbol{p}}^{(r)}\right]_k \log\left[\boldsymbol{p}^{(r)}\right]_k$. Therefore, for any plug-in estimator $\widetilde{\boldsymbol{p}}^{(r)} = \widetilde{\boldsymbol{A}}^{*(r)}\widetilde{\boldsymbol{f}}^{\circledast}\left(X(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\right)$, minimizing the CE loss function seeks an estimator that matches the empirical point mass function $\widehat{\boldsymbol{p}}^{(r)}$ obtained from the training data.

Considering all annotator responses, we minimize the following the CE criterion:

$$\left(\widehat{\boldsymbol{f}}^{\circledast}, \widehat{\boldsymbol{A}}^{*}\right) = \underset{\boldsymbol{f}\in\mathcal{F},\, \boldsymbol{A}^{(r)}\in\mathcal{A}}{\arg\min}\left\{-\frac{1}{nR}\sum_{r=1}^{R}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{I}\left(\widetilde{Y}_i^{(r)} = k\right)\log\left[\boldsymbol{A}^{(r)}\,\boldsymbol{f}\left(X_i(\boldsymbol{s}),\, \boldsymbol{s} \in [0,1]^d\right)\right]_k\right\}, \tag{3.1}$$

where $\widehat{\boldsymbol{A}}^{*} = \{\widehat{\boldsymbol{A}}^{*(r)} : r \in [R]\}$ is the collection of estimated CM, $\mathcal{F}$ is the class of candidate classifiers $\boldsymbol{f} = (f_k)_{k\in[K]}^{\top}$ with each f_k being a functional mapping the data $X(\boldsymbol{s})$: $[0,1]^d \to \mathbb{R}$ to a real value in $[0,1]$, and $\mathcal{A}$ is the constrained set of possible CMs $\{\boldsymbol{A} \in \mathbb{R}^{K\times K} : [\boldsymbol{A}]_{kk'} \geq 0 \text{ for all } k, k' \in [K],\ \mathbf{1}^{\top}\boldsymbol{A} = \mathbf{1}^{\top}\}$. Here, $\mathbf{1}$ is the K-dimensional

unit vector.

Practically, it is imperative to employ a dimensional reduction strategy for functional observations to address their infinite dimensionality. Specifically, the sample data function $X_i(\boldsymbol{s})$ can be estimated as $X_i(\boldsymbol{s}) \approx \sum_{j=1}^{\infty} \widehat{\xi}_{ij}\widehat{\psi}_j(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d$. Intuitively, $\widehat{\boldsymbol{\xi}}^{(i)} \triangleq (\widehat{\xi}_{i1}, \widehat{\xi}_{i2}, \ldots)$ is an estimator of $\boldsymbol{\xi}^{(i)} \triangleq (\xi_{i1}, \xi_{i2}, \ldots)$, in which ξ_{ij}'s are unobservable random coefficients of $X_i(\boldsymbol{s})$ with respect to the population basis ψ_j. Hence, it is natural to design classifiers based on $\widehat{\boldsymbol{\xi}}^{(i)}$'s. In this representation, we suppose $\widehat{\psi}_j(\boldsymbol{s})$ is a set of pre-selected basis functions, such as Fourier, B-spline, or wavelet bases, which are computationally efficient to be used for any dimensional functional observations. It is important to note that while $\widehat{\psi}_j(\boldsymbol{s})$ may differ from $\psi_j(\boldsymbol{s})$, this discrepancy does not affect the numerical performance of the proposed classifier. This is because our chosen function class is sufficiently flexible, encompassing a wide range of functional forms beyond the constraints of parametric models. For any pre-specified basis, the decomposition ensures consistent data processing across all observations—projecting the functional data onto a common space, where any differences manifest solely through the extracted functional projection scores. See Section 5 for numerical results details. Consequently, the proposed DNN classifier effectively distinguishes between different labels by leveraging the entirety of the captured information, with its theoretical guarantees formally established in Section 4.

Let $\widehat{\boldsymbol{\xi}}_J^{(i)} = \left(\widehat{\xi}_{i1}, \ldots, \widehat{\xi}_{iJ}\right)^{\mathsf{T}}$ be the J-dimensional truncation of $\widehat{\boldsymbol{\xi}}^{(i)}$ for $i \in [n]$, which is a finite realization of $\widehat{\boldsymbol{\xi}}^{(i)}$. Note that J is a tuning parameter, which is to be selected later. By using the reduced dimensional vector $\widehat{\boldsymbol{\xi}}_J$, the version of $\widehat{\boldsymbol{\xi}}_J^{(i)}$ with the subject index i dropped, we define the classifier $f_k^* : \mathbb{R}^J \to [0,1]$ and $\boldsymbol{f}^* = (f_k^*)_{k\in[K]}^{\top}$, such that $\boldsymbol{f}^*\left(\widehat{\boldsymbol{\xi}}_J\right) \approx \boldsymbol{f}^{\circledast}\left(X(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\right)$, and the associated function class is denoted by $\mathcal{F}_J$. By projecting the function space for $X(\boldsymbol{s})$ to the vector space for $\widehat{\boldsymbol{\xi}}_J$, $f_k^*(\boldsymbol{x})$ represents the conditional probability $f_k^*(\boldsymbol{x}) = \mathbb{P}\left(Y = k | \boldsymbol{\xi}_J = \boldsymbol{x}\right)$, yielding $\sum_{k=1}^{K} f_k^*(\boldsymbol{x}) = 1$ for any $\boldsymbol{x} \in \mathbb{R}^J$. Consequently, the minimization problem in Equation (3.1) involving infinite

dimensional functional data is modified to be a finite dimensional minimization problem:

$$(\widehat{\boldsymbol{f}}^*, \widehat{\boldsymbol{A}}^*) = \underset{\boldsymbol{f}\in\mathcal{F}_J, \boldsymbol{A}^{(r)}\in\mathcal{A}}{\arg\min} \left\{ -\frac{1}{nR} \sum_{r=1}^{R}\sum_{i=1}^{n}\sum_{k=1}^{K} \mathbb{I}(\widetilde{Y}_i^{(r)} = k) \log\left[\boldsymbol{A}^{(r)}\boldsymbol{f}\left(\widehat{\boldsymbol{\xi}}_J^{(i)}\right)\right]_k \right\}, \quad (3.2)$$

which determines an estimated true label $\widehat{Y} = \arg\max_{k\in[K]} \widehat{f}_k^*$, where $\widehat{f}_k^*$ is the kth element of $\widehat{\boldsymbol{f}}^*$. It is noteworthy that f_k^* is J-associated. Assuming a one-to-one correspondence between f_k^* and J, we omit J in f_k^* for notational simplicity in the following development.

3.2. Annotation deep neural networks

Let $\sigma(x) = x_+$ be the ReLU activation function, and $\boldsymbol{\sigma}^*(\boldsymbol{y}) = \left(\frac{\exp(y_1)}{\sum_{k=1}^K \exp(y_k)}, \ldots, \frac{\exp(y_K)}{\sum_{k=1}^K \exp(y_k)}\right)^\intercal$ be the K-dimensional softmax activation function for $\boldsymbol{y} = (y_1, \ldots, y_K)^\intercal \in \mathbb{R}^K$. For any real vectors $\boldsymbol{V} = (v_1, \ldots, v_w)^\intercal$ and $\boldsymbol{z} = (z_1, \ldots, z_w)^\intercal$, define the shift activation function $\sigma_{\boldsymbol{V}}(\boldsymbol{z}) = (\sigma(z_1 - v_1), \ldots, \sigma(z_w - v_w))^\intercal$. For $L \geq 1$ and $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_L)^\intercal \in \mathbb{N}^L$, let $\mathcal{D}(L, J, \boldsymbol{\gamma})$ denote the class of fully connected feedforward DNN with J inputs, L hidden layers, and, for $l = 1, \ldots, L$, γ_l nodes on the l-th hidden layer. Equivalently, any $\boldsymbol{f} \in \mathcal{D}(L, J, \boldsymbol{\gamma})$ has an expression

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{\sigma}^*(\mathbf{W}_L\sigma_{\boldsymbol{V}_L}(\mathbf{W}_{L-1}\sigma_{\boldsymbol{V}_{L-1}}\ldots(\mathbf{W}_1\sigma_{\boldsymbol{V}_1}(\mathbf{W}_0\boldsymbol{x})))\ldots), \quad \boldsymbol{x}\in\mathbb{R}^J,$$

where $\mathbf{W}_l \in \mathbb{R}^{p_{l+1}\times p_l}$, for $l = 0, \ldots, L$, are weight matrices, and $\boldsymbol{V}_l \in \mathbb{R}^{p_l}$, are shift vectors for $l = 1, \ldots, L$. Here, we adopt the convention that $\gamma_0 = J$ and $\gamma_{L+1} = K$.

Due to the complexity of the fully connected DNN class, to avoid overparameterization, we consider the following sparse DNN class:

$$\mathcal{D}(L, J, \boldsymbol{\gamma}, s) = \left\{ \boldsymbol{f} \in \mathcal{D}(L, J, \boldsymbol{\gamma}) : \sum_{l=0}^{L} \|\mathbf{W}_l\|_0 + \|\mathbf{V}_l\|_0 \leq s, \max_{0\leq l\leq L} (\|\mathbf{W}_l\|_{\max} \vee \|\mathbf{V}_l\|_{\max}) \leq 1 \right\},$$

where $\|\cdot\|_{\max}$ denotes the maximum-entry norm of a matrix or a vector, $\|\cdot\|_0$ represents the number of non-zero elements of a matrix or a vector, and s controls the total number of active neurons.

Let $\mathcal{D} \triangleq \mathcal{D}(L, J, \boldsymbol{\gamma}, s)$, with the dependence on L, J, and $\boldsymbol{\gamma}$ suppressed in the notation.

Given the training data $\left\{\widetilde{\boldsymbol{Y}}_i, X_i(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\right\}_{i=1}^n$, the CE minimizer in Equation (3.2) can be modified as

$$(\widehat{\boldsymbol{f}}^*, \widehat{\boldsymbol{A}}^*) = \underset{\boldsymbol{f}\in\mathcal{D}, \boldsymbol{A}^{(r)}\in\mathcal{A}}{\arg\min} \left\{ -\frac{1}{nR} \sum_{r=1}^{R}\sum_{i=1}^{n}\sum_{k=1}^{K} \mathbb{I}(\widetilde{Y}_i^{(r)} = k) \log\left[\boldsymbol{A}^{(r)} \boldsymbol{f}(\boldsymbol{\xi}_J^{(i)})\right]_k \right\}. \tag{3.3}$$

The optimization problem in (3.3) two components: the base classifier which estimates the ground truth class probability vector, with its kth element approximating $\mathbb{P}(Y = k|\boldsymbol{\xi}_J)$, and the set of the CM estimators $\widehat{\boldsymbol{A}}^*$. Each $\widehat{\boldsymbol{A}}^{*(r)}\widehat{\boldsymbol{f}}^*(\boldsymbol{\xi}_J)$ represents the estimated class probability vector of $\left(\mathbb{P}\left(\widetilde{Y}^{(r)} = k|\boldsymbol{\xi}_J\right) : k \in [K]\right)^\intercal$ for the rth annotator.

The minimization (3.3) encourages each annotator-specific prediction $\widehat{\boldsymbol{A}}^{*(r)}\widehat{\boldsymbol{f}}^*(\boldsymbol{\xi}_J)$ to be as close as possible to the noisy label distribution $\boldsymbol{p}^{(r)}$ of the corresponding annotator. However, this loss function alone is not capable of separating the annotation noise from the true label distribution; there are infinite combinations of $\widehat{\boldsymbol{A}}^{*(r)}$ and classification model $\widehat{\boldsymbol{f}}^*(\boldsymbol{\xi}_J)$ such that $\widehat{\boldsymbol{A}}^{*(r)}\widehat{\boldsymbol{f}}^*(\boldsymbol{\xi}_J)$ perfectly matches the annotator's label distribution $\boldsymbol{p}^{(r)}$ for any input $\boldsymbol{\xi}_J$. This indicates that regularization is necessary to guide the optimization toward meaningful solutions, i.e., $\widehat{\boldsymbol{A}}^{*(r)}$ getting close to $\boldsymbol{A}^{*(r)}$ after several iterations in the optimization procedure. The estimated true label is as accurate as possible, i.e., $\left[\mathbb{P}\left(\widehat{Y} = k|Y = j\right)\right]_{(k,j)\in[K]}$ converges to $\boldsymbol{I}_K$ the $K \times K$ identity matrix, after a number of iterations in the optimization procedure. To this end, we add the trace of the estimated CM to the loss function in (3.3) and then minimize the trace-regularized loss function

$$(\widehat{\boldsymbol{f}}^{\texttt{afDNN}}, \widehat{\boldsymbol{A}}_T^*) = \underset{\boldsymbol{f}\in\mathcal{D}, \boldsymbol{A}^{(r)}\in\mathcal{A}}{\arg\min} \left\{ -\frac{1}{nR} \sum_{r=1}^{R}\sum_{i=1}^{n}\sum_{k=1}^{K} \mathbb{I}(\widetilde{Y}_i^{(r)} = k) \log\left[\boldsymbol{A}^{(r)} \boldsymbol{f}(\boldsymbol{\xi}_J^{(i)})\right]_k + \lambda \mathrm{tr}\left(\sum_{r=1}^{R} \boldsymbol{A}^{(r)}\right) \right\}, \tag{3.4}$$

where $\mathrm{tr}(\boldsymbol{A})$ denotes the trace of matrix $\boldsymbol{A}$ and $\lambda > 0$ is a tuning parameter. The trace $\mathrm{tr}\left(\sum_{r=1}^{R} \boldsymbol{A}^{(r)}\right)$ is the total correct-labeling probability aggregated across annotators, which may yield the overall average probability that an annotator provides an accurate label. Essentially, minimizing the trace encourages the estimated CMs toward lower overall reliability,

while minimizing the cross entropy ensures fidelity with observed noisy annotators.

Let $\left\{\{X_i(\boldsymbol{s}_m)\}_{m=1}^{N_i}, \widetilde{\boldsymbol{Y}}_i\right\}_{i=1}^{n}$ be the collection of samples, where N_i is the number of observations for the ith subjects, and $\{\boldsymbol{s}_m\}_{m\in[N_i]}$ are the corresponding grid points. The implementation steps for the proposed `afDNN` classifier are summarized in Algorithm 1, the related theories presented in Section 4. The tuning parameters include J, L, $\boldsymbol{\gamma}$, s, and λ, whose values can be determined using a simple training/validation data split: we randomly partition the n subjects into training and validation sets with ratio $9:1$. We fit `afDNN` on the training data over a grid of candidate values, and the configuration that yields the smallest loss on the validation set is chosen (see Algorithm 1). Note that the sparsity is not directly applicable by assigning the number of active neurons. Instead, regularization techniques are typically employed during training to control model complexity and mitigate overfitting. In contrast to regularization methods that operate exclusively through the loss function, dropout introduces stochastic modifications to the network architecture during training and has been shown to be particularly effective in preventing overfitting (Srivastava et al., 2014). Consequently, in practical implementations we recommend the use of dropout as a computationally efficient surrogate for the sparsity assumptions employed in our theory.

4. Theoretical properties

In this section, we first validate the use of the trace regularizer in (3.4). Next, we examine the performance of the proposed classifier with respect to the KL divergence, and establish the fast convergence rates for both densely and discretely observed functional data.

4.1. Justification of the trace regularization

Let $\widehat{Y}$ denote a predicted label obtained from a classifier, and let $\boldsymbol{P}$ denote the matrix whose (k,j) element is $\mathbb{P}\left(\widehat{Y}=k|Y=j\right)$. We first impose regularity conditions on the $\boldsymbol{A}^{*(r)}$ and $\boldsymbol{P}$.

Assumption 1. *Assume that*

Algorithm 1: Training algorithm for `afDNN`

Input: Collection of samples $\left\{\{X_i(\boldsymbol{s}_m)\}_{m=1}^{N_i}, \widetilde{\boldsymbol{Y}}_i\right\}_{i=1}^{n}$, training index set $\mathcal{I}_1$, validation index set $\mathcal{I}_2$, $|\mathcal{I}_1| = \lfloor 0.9n \rfloor$, $|\mathcal{I}_2| = \lfloor 0.1n \rfloor$, candidates for number of projection scores $\{J_0, \ldots, J_{\tau_1}\}$, candidates for depth $\{L_0, \ldots, L_{\tau_2}\}$, candidates for width $\{\gamma_0, \ldots, \gamma_{\tau_3}\}$, candidates for dropout rates $\{s_0, \ldots, s_{\tau_4}\}$, candidates for the trace tuning $\{\lambda_0, \ldots, \lambda_{\tau_5}\}$, number of epochs, batch size, learning rate

1 Obtain $\widehat{\boldsymbol{\xi}}_J^{(i)} = (\widehat{\xi}_1^{(i)}, \ldots, \widehat{\xi}_J^{(i)})$ by $\widehat{\xi}_j^{(i)} = \sum_{m=1}^{N_i} X_i(\boldsymbol{s}_m)\, \widehat{\psi}_j(\boldsymbol{s}_m)\, \omega_m, j \in [J], i \in [n]$, where $\{\omega_m\}_{m=1}^{N_i}$ are prescribed quadrature weights (e.g., $\omega_m = 1/N_i$ for equally spaced observations), $\widehat{\psi}_j$ is the jth basis function discussed in Section 3.1, and $J = \max\{J_0, \ldots, J_{N_1}\}$.

2 For all candidates,

(i) Train $\widehat{\boldsymbol{f}}_{J_{\ell_1}, L_{\ell_2}, \gamma_{\ell_3}, s_{\ell_4}, \lambda_{\ell_5}}$ by Equation (3.4) with $\left(\{X_i(\boldsymbol{s}_m)\}_{m=1}^{N_i}, \widetilde{\boldsymbol{Y}}_i\right)_{i \in \mathcal{I}_1}$.

(ii) Compute $\text{err}(\ell_1, \ell_2, \ell_3, \ell_4, \ell_5) \triangleq \frac{1}{|\mathcal{I}_2| R} \sum_{i \in \mathcal{I}_2, r \in [R], k \in [K]} \mathbb{I}(\widetilde{Y}_i^{(r)} = k)$ $\log\left[\boldsymbol{A}^{(r)_{J_{\ell_1}, L_{\ell_2}, \gamma_{\ell_3}, s_{\ell_4}, \lambda_{\ell_5}}} \boldsymbol{f}_{J_{\ell_1}, L_{\ell_2}, \gamma_{\ell_3}, s_{\ell_4}, \lambda_{\ell_5}}\left(\boldsymbol{\xi}_J^{(i)}\right)\right]_k + \lambda_{\ell_5} \text{tr}\left(\boldsymbol{A}^{(r)}_{J_{\ell_1}, L_{\ell_2}, \gamma_{\ell_3}, s_{\ell_4}, \lambda_{\ell_5}}\right)$

3 Obtain $\left(J_{\ell_1^*}, L_{\ell_2^*}, \gamma_{\ell_3^*}, s_{\ell_4^*}, \lambda_{\ell_5^*}\right) = \arg\min_{\ell_1, \ell_2, \ell_3, \ell_4, \ell_5} \text{err}(\ell_1, \ell_2, \ell_3, \ell_4, \ell_5)$

4 Train $\widehat{\boldsymbol{f}}_{J_{\ell_1^*}, L_{\ell_2^*}, \gamma_{\ell_3^*}, s_{\ell_4^*}, \lambda_{\ell_5^*}}$ and $\boldsymbol{A}^{(r)}_{J_{\ell_1}, L_{\ell_2}, \gamma_{\ell_3}, s_{\ell_4}, \lambda_{\ell_5}}$ by Equation (3.4) with $\left\{\{X_i(\boldsymbol{s}_m)\}_{m=1}^{N_i}, \widetilde{\boldsymbol{Y}}_i\right\}_{i=1}^{n}$

Output: $\widehat{\boldsymbol{f}}^{\texttt{afDNN}}$ and $\widehat{\boldsymbol{A}}_T^*$

(a) $\widehat{\boldsymbol{A}}^{*(r)} \boldsymbol{P} = \boldsymbol{A}^{*(r)}$ *for any* $r \in [R]$*;*

(b) The average $R^{-1} \sum_{r=1}^{R} \boldsymbol{A}^{*(r)}$ *and its estimate* $R^{-1} \sum_{r=1}^{R} \widehat{\boldsymbol{A}}^{*(r)}$ *are diagonally dominant, i.e.,* $a_{jj} > \max(a_{j\ell}, a_{\ell j})$ *and* $\widehat{a}_{jj} > \max(\widehat{a}_{j\ell}, \widehat{a}_{\ell j})$*, where* $\widehat{a}_{jk}$ *represents the* (j,k) *element of* $\widehat{\boldsymbol{A}}^{*(r)}$ *for* $j, k \in [K]$*.*

Remark 1. *The assumptions imposed on the CM are mild. For the* (k,j)*-th entry of* $\boldsymbol{A}^{*(r)}$*, it follows that* $\mathbb{P}\left(\widetilde{Y}^{(r)} = k | Y = j\right) = \sum_{l=1}^{K} \mathbb{P}\left(\widetilde{Y}^{(r)} = k | \widehat{Y} = l\right) \mathbb{P}\left(\widehat{Y} = l | Y = j\right)$*, where* $\widehat{Y}$ *can be an estimated label from any generic classifier. Hence, Assumption 1(a) is universally valid. Assumption 1(b) requires that annotators have reasonable skills to ensure that for any* $j \in [K]$*,* $P(\widetilde{Y}^{(r)} = j | Y = j) = \max_{k \in [K]} P(\widetilde{Y}^{(r)} = k | Y = j)$*, where* $r \in [R]$*. In other words, an annotator is more likely to annotate the true label than any other incorrect label even though the true label cannot be surely annotated. To encourage the diagonally dominated estimators* $\widehat{\boldsymbol{A}}^*$*, we start from the initial matrices as the identity matrix.*

Proposition 1. *Under Assumption 1, we have that for arbitrary $J \in \mathbb{N}^+$ and $\lambda > 0$,*

$$\begin{aligned}&\left(\boldsymbol{A}^{*(1)}, \ldots, \boldsymbol{A}^{*(R)}\right)\\ =&\underset{\left\{\widehat{\boldsymbol{A}}^{*(r)} \in \mathcal{A}\right\}_{r=1}^{R}}{\arg\min} -\frac{1}{nR}\sum_{r=1}^{R}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{I}(\widetilde{Y}_i^{(r)}=k)\log[\widehat{\boldsymbol{A}}^{*(r)}\boldsymbol{f}^*(\boldsymbol{\xi}_J^{(i)})]_k + \lambda \operatorname{tr}\left(\sum_{r=1}^{R}\widehat{\boldsymbol{A}}^{*(r)}\right),\end{aligned}$$

and the solution $\left(\boldsymbol{A}^{*(1)}, \ldots, \boldsymbol{A}^{*(R)}\right)$ is unique.

Remark 2. *Proposition 1 shows that, when the classifier is an oracle, the true CM is the unique minimizer of the traced-adjusted CE loss in (3.4). Meanwhile, minimizing the CE loss with an appropriately chosen λ drives $\widehat{\boldsymbol{f}}^*$ close to the ground truth $\boldsymbol{f}^*$. Intuitively, this joint optimization disentangles the underlying true distribution from the annotation noise by identifying the maximal extent of confusion that can still explain the noisy observations accurately. It is noted that the classifier f itself is not identifiable at the level of neural-network parameters. In our implementation, f is represented by a ReLU feed forward network, and different parameter vectors can yield exactly the same classification function (Grigsby et al., 2023). Our identification result in Proposition 1 is thus formulated at the level of the CM for a given classifier f^*, rather than at the level of the neural-network parameters. This is sufficient for our theoretical developments, where the DNN serves as a flexible function approximator and the primary object of interest is the identifiable collection of confusion matrices. The proof of Proposition 1 is provided in the Supplementary file.*

4.2. Function class for the true conditional probability

Define the true conditional probability $\boldsymbol{f}^{\circledast} = (f_1^{\circledast}, \ldots, f_K^{\circledast})^{\intercal}$, and $f_k^{\circledast}(\boldsymbol{\xi}) = \mathbb{P}(Y = k|\boldsymbol{\xi})$ is defined on $\mathbb{R}^{\infty}$. For any $\widehat{\boldsymbol{f}}^*$ trained from $\left\{\left(\widetilde{\boldsymbol{Y}}_i, X_i(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\right)\right\}_{i=1}^{n}$, we evaluate its performance by the log-likelihood ratio

$$\mathbb{E}\left[\sum_{k=1}^{K} f_k^{\circledast}(X(\boldsymbol{s}))\log\left(\frac{f_k^{\circledast}(X(\boldsymbol{s}))}{\widehat{f}_k^*(\boldsymbol{\xi}_J)}\right)\right] = \mathbb{E}\left[KL\left(\boldsymbol{f}^{\circledast}(X(\boldsymbol{s})), \widehat{\boldsymbol{f}}^*(\boldsymbol{\xi}_J)\right)\right],$$

where the expectations are evaluated with respect to the joint distribution of $\left\{\widetilde{\boldsymbol{Y}}_i, X_i(\boldsymbol{s}), \boldsymbol{s} \in [0,1]^d\right\}_{i=1}^{n}$. The corresponding discrete version Kullback-Leibler (KL) diver-

gence is defined as

$$\widehat{KL}\left(\boldsymbol{f}^{\circledast}, \widehat{\boldsymbol{f}}^{*}\right) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} f_k^{\circledast}(X_i(\boldsymbol{s}))\log\left(\frac{f_k^{\circledast}(X_i(\boldsymbol{s}))}{\widehat{f}_k^{*}(\boldsymbol{\xi}_{iJ})}\right).$$

Given an absolute constant $C_0 \geq 2$, for any classifier $\widehat{\boldsymbol{f}}$, define the truncated KL risk for the density $\boldsymbol{f}^{\circledast}$ as

$$R_{\boldsymbol{f}^{\circledast},C_0}\left(\widehat{\boldsymbol{f}}^{*}\right) = \mathrm{E}_{\boldsymbol{f}^{\circledast}}\left\{\sum_{k=1}^{K} f_k^{\circledast}(X(\boldsymbol{s}))\left[C_0 \wedge \log\left(\frac{f_k^{\circledast}(X(\boldsymbol{s}))}{\widehat{f}_k^{*}(\boldsymbol{\xi})}\right)\right]\right\}, \tag{4.5}$$

such that C_0 prevents the exploding components of $\widehat{\boldsymbol{f}}_k^{*}$. Here $a \wedge b$ represent $\min(a, b)$ for any two values a and b. In the following, we derive the fast convergence rate with respect to the truncated KL risk.

We first show the boundary conditions required for the entire functional covariate $X(\boldsymbol{s})$. Equivalently, we consider the infinite dimensional vector $\boldsymbol{\xi}$ as defined in Section 3.

Assumption 2. *(Boundary condition)*

(a) There exists a relatively small $\epsilon > 0$, such that $\mathbb{P}\left(f_k^{\circledast}(\boldsymbol{\xi}) > \epsilon\right) = 1$, for any $k \in [K]$;

(b) There exist an absolute constant $C > 0$ and a positive vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)^{\intercal}$, such that $\mathbb{P}\left(f_k^{\circledast}(\boldsymbol{\xi}) \leq x\right) \leq Cx^{\alpha_k}$, for any $x \in (0, 1]$ and $k \in [K]$.

Remark 3. *Assumption 2(a) provides a uniform lower bound on $f_k^{\circledast}$, ensuring that each $f_k^{\circledast}$ is bounded away from zero almost surely. In other words, if $f_k^{\circledast} = 0$ for some k, then only $K - 1$ classes are present. Assumption 2(b) characterizes the behavior of $f_k^{\circledast}$ near zero; in the trivial case where $\alpha_k = 0$ for all $k = 1, \ldots, K$, the condition holds universally provided $C \geq 1$. More precisely, this assumption controls the decay rate of the probability measure $\mathbb{P}\{0 \leq f_k^{\circledast}(\boldsymbol{\xi}) \leq x\}$. Similar conditions are found in multi-class classification for multivariate data (Bos and Schmidt-Hieber, 2022; Wang and Cao, 2024) and in binary classification (Mammen and Tsybakov, 1999; Tsybakov, 2004).*

Next, to establish a connection between the true conditional probability $\boldsymbol{f}^{\circledast}$ and its finite

counterpart $\boldsymbol{f}^*$, on which the classifier is constructed, we introduce the following assumption to characterize the discrepancy between $\boldsymbol{f}^{\circledast}$ and $\boldsymbol{f}^*$.

Assumption 3. *(Approximation error of $\boldsymbol{f}^*$) There exist a universal constant $J_0 \geq 1$ and decreasing functions $\zeta(\cdot) : [1, \infty) \to \mathbb{R}_+$ and $\Gamma(\cdot) : [0, \infty) \to \mathbb{R}_+$, with $\sup_{J \geq 1} J^{\rho}\zeta(J) < \infty$ for some $\rho > 0$ and $\int_0^{\infty} \Gamma(x)dx < \infty$, such that for any $J \geq J_0$, $k \in [K]$ and $x > 0$,*

$$\mathbb{P}\left(|f_k^{\circledast}(\boldsymbol{\xi}) - f_k^*(\boldsymbol{\xi}_J)| \geq x\right) \leq \zeta(J)\Gamma(x).$$

Remark 4. *Since it is impractical to employ an infinite number of projections, Assumption 3 sets a uniform upper bound on the probability that $f_k^{\circledast}$ deviates from f_k^* by at least x. This bound vanishes as either J or x tends to infinity, implying that for a sufficiently large J, f_k^* approximates $f_k^{\circledast}$ with high probability. Moreover, the parameter ρ regulates the decay of $\zeta(J)$, with larger ρ leading to a faster decay. Similar conditions are imposed for both binary classification (Wang et al., 2023) and multi-class classification (Wang and Cao, 2024).*

In the sequel, we introduce a function class that contains the true function $\boldsymbol{f}^*$ defined on $\mathbb{R}^J$, where J is an arbitrary finite positive integer. Suppose $f_k^*(\cdot)$ belongs to the class of Hölder-smooth functions defined as below.

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

$$\mathcal{C}^{\beta}(D, C) = \left\{ f : D \mapsto \mathbb{R} \middle| \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|<\beta} \|\partial^{\boldsymbol{\alpha}} f\|_{\infty} f + \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|=\lfloor\beta\rfloor} \sup_{\boldsymbol{z},\boldsymbol{z}' \in D, \boldsymbol{z} \neq \boldsymbol{z}'} \frac{|\partial^{\boldsymbol{\alpha}} f(\boldsymbol{z}) - \partial^{\boldsymbol{\alpha}} f(\boldsymbol{z}')|}{\|\boldsymbol{z} - \boldsymbol{z}'\|_{\infty}^{\beta - \lfloor\beta\rfloor}} \leq C \right\},$$

where $\partial^{\boldsymbol{\alpha}} = \partial^{\alpha_1} \ldots \partial^{\alpha_t}$ denotes the partial differential operator with multi-index $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_t) \in \mathbb{N}^t$, $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_t$.

Equivalently, $\mathcal{C}^{\beta}(D, C)$ is the ball of β-Hölder smooth functions on D with radius R. For a given $t \geq 1$, function $f : \mathbb{R}^t \to \mathbb{R}$ is said to be locally β-Hölder smooth if for any $a, b \in \mathbb{R}$, there exists a constant C (possibly depending on a, b) such that $f \in \mathcal{C}^{\beta}([a, b]^t, C)$.

For $q \geq 0$ and $J \geq 1$, let $d_0 = J$ and $d_{q+1} = 1$. For $\boldsymbol{d} = (d_1, \ldots, d_q) \in \mathbb{N}_+^q$, $\boldsymbol{t} = (t_0, \ldots, t_q) \in \mathbb{N}_+^{q+1}$ with $t_u \leq d_u$ for $u = 0, \ldots, q$, and $\boldsymbol{\beta} \triangleq (\beta_0, \ldots, \beta_q) \in \mathbb{R}_+^{q+1}$,

let $\mathcal{G}(q, J, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta})$ be the class of functions g satisfying a modular expression $g(\boldsymbol{z}) = g_q \circ \cdots \circ g_0(\boldsymbol{z}) \in (0,1)$, for any $\boldsymbol{z} \in \mathbb{R}^{d_0}$, where $g_u = (g_{u1}, \ldots, g_{ud_{u+1}}) : \mathbb{R}^{d_u} \mapsto \mathbb{R}^{d_{u+1}}$ and $g_{uv} : \mathbb{R}^{t_u} \mapsto \mathbb{R}$ are locally β_u-Hölder smooth, with $u = 0, \ldots, q$ and $v = 1, \ldots, d_{u+1}$. The d_u arguments of g_u are locally connected in the sense that each component g_{uv} 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); Wang et al. (2021); Bos and Schmidt-Hieber (2022); Kim et al. (2021); Hu et al. (2020) in multivariate regression or classification to overcome high-dimensionality. Generalized additive models (Hastie and Tibshirani, 1990) and tensor product space ANOVA models (Lin, 2000) are special cases (Liu et al., 2021).

We consider the class of the true conditional probability $\boldsymbol{f}^{\circledast}$ as

$$\mathcal{H} \triangleq \mathcal{H}\left(K, \{q^{(k)}\}_{k=1}^K, \{\boldsymbol{d}^{(k)}\}_{k=1}^K, \{\boldsymbol{t}^{(k)}\}_{k=1}^K, \{\boldsymbol{\beta}^{(k)}\}_{k=1}^K, \{\alpha_k\}_{k=1}^K, \zeta(\cdot), \Gamma(\cdot), C, \rho, \epsilon\right),$$

such that for any $J \geq 1$, the corresponding finite realization $f_k^* \in \mathcal{G}\left(q^{(k)}, J, \boldsymbol{d}^{(k)}, \boldsymbol{t}^{(k)}, \boldsymbol{\beta}^{(k)}\right)$, where $\boldsymbol{d}^{(k)} = (d_1^{(k)}, \ldots, d_{q^{(k)}}^{(k)}) \in \mathbb{N}_+^{q^{(k)}}$, $\boldsymbol{t}^{(k)} = (t_0^{(k)}, \ldots, t_{q^{(k)}}^{(k)}) \in \mathbb{N}_+^{q^{(k)}+1}$ with $t_u^{(k)} \leq d_u^{(k)}$ for $u = 0, \ldots, q^{(k)}$, and $\boldsymbol{\beta}^{(k)} \triangleq (\beta_0, \ldots, \beta_{q^{(k)}}^{(k)}) \in \mathbb{R}_+^{q^{(k)}+1}$, and Assumptions 2 and 3 are satisfied. Note that this class $\mathcal{H}$ includes many popular models studied in literature, both Gaussian and non-Gaussian.

Throughout the paper, we explore f_k^* in some complicated $\mathcal{G}$ with group-specific parameters $q^{(k)}, \boldsymbol{d}^{(k)}, \boldsymbol{t}^{(k)}$ and $\boldsymbol{\beta}^{(k)}$ for arbitrary J. The selection range of the optimal truncation parameter J is based on the asymptotic order provided in Assumptions S1 and S2 in Supplementary Material. Although f_k^* has J arguments, it involves at most $t_0^{(k)} d_1^{(k)}$ 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 and Zhang (2019b,a) show 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 an extension of Schmidt-Hieber (2020) in the FDA setting.

4.3. Convergence rate for `afDNN` classifier

In this section, we establish the fast convergence rate of our proposed classifier relative to the Bayes classifier.

Theorem 1. *Under Assumptions 1–3 and Assumption S1 in Supplementary Material, it follows that*

$$\sup_{\boldsymbol{f}^{\circledast}\in\mathcal{H}} R_{\boldsymbol{f}^{\circledast},C_0}\left(\widehat{\boldsymbol{f}}^{\texttt{afDNN}}\right) \lesssim n^{-\theta}\log^3 n + R^{-1}\sum_{r=1}^{R}\operatorname{tr}\left(\boldsymbol{A}^{*(r)}-\boldsymbol{I}_K\right),$$

where the risk function $R_{\boldsymbol{f}^{\circledast},C_0}$ is defined in Equation (4.5) and network classifier $\widehat{\boldsymbol{f}}^{\texttt{afDNN}} \in \mathcal{D}$ defined in Assumption S1.

Remark 5. *When the functional trajectories are densely observed, Theorem 1 provides an upper bound for the KL misclassification risk, which consists of two components. The first component $n^{-\theta}\log^3 n$ corresponds to the classical nonparametric rate for estimating decision boundaries. This rate aligns with the convergence rates established in high-dimensional settings (Bos and Schmidt-Hieber, 2022) and multi-dimensional functional settings (Wang and Cao, 2024). The decay order θ is governed by the class that presents the most significant classification challenge. The second component $\frac{1}{R}\sum_{r=1}^{R}\operatorname{tr}\left(\boldsymbol{A}^{*(r)}-\boldsymbol{I}_K\right)$ is determined by the average trace of the CM across all annotators. Since the trace quantifies how accurately an annotator predicts the correct label given the ground truth, a trace far from the identity matrix indicates that the classifier is trained on highly unreliable information. In such cases, the classifier fails to converge to the Bayes classifier, even when the sample size is large enough. As a result, the KL misclassification risk is slowed down by annotators with poor knowledge but accelerated by those with excellent knowledge.*

In practice, several ways can be used to mitigate the impact of poorly performing annotators. A key advantage of our framework is that `afDNN` estimates annotator-specific confusion matrices. Under mild conditions, these matrices can still be consistently estimated even when some annotators are of lower quality. This allows us to diagnose annotator

reliability: annotators with very small estimated diagonal entries (or strongly off-diagonal patterns) can be identified as very unreliable annotators and subsequently excluded from the aggregation step. After screening out such annotators, one can retrain the classifier using only the labels from the remaining, more reliable annotators, which typically leads to improved performance.

Next, we consider the discretely observed functional data. Compared to densely observed functional data, these discrete measurements necessarily contain less information.

Theorem 2. *Under Assumptions 1–3 and Assumption S2 in Supplementary Material, there exists a universal constant c, such that the corresponding classifier follows that*

$$\sup_{\boldsymbol{f}^{\circledast}\in\mathcal{H}} R_{\boldsymbol{f}^{\circledast},C_0}\left(\widehat{\boldsymbol{f}}^{afDNN}\right) \lesssim n^{-\theta}\log^3 n\mathbb{I}(\underline{N}\geq N^*)+\underline{N}^{-\theta'}\mathbb{I}(\underline{N}<N^*)+R^{-1}\sum_{r=1}^{R}\mathrm{tr}\left(\boldsymbol{A}^{*(r)}-\boldsymbol{I}_K\right),$$

where $\underline{N}=\min_{i=1,\ldots,n} N_i$, $N^*=c\left(n^{\theta}/\log^3 n\right)^{1/\theta'}$ and $\widehat{\boldsymbol{f}}^{afDNN}\in\mathcal{D}$ defined in Assumption S2.

Remark 6. *Theorem 2 establishes the convergence rate of the truncated KL risk for discretely observed functional data, which is particularly relevant in practical applications. It reveals that N^* serves as a critical sampling frequency for any fixed sample size n within the true parameter space $\mathcal{H}$. Specifically, when $\underline{N}>N^*$, the truncated KL risk is bounded by $n^{-\theta}\log^3 n+R^{-1}\sum_{r=1}^{R}\mathrm{tr}\left(\boldsymbol{A}^{*(r)}-\boldsymbol{I}_K\right)$, which aligns with the rate established in Theorem 1. This result indicates that when the sampling frequency is sufficiently high, the convergence rate based on discretely observed data approaches that of densely observed functional trajectories. Conversely, when $\underline{N}<N^*$, the sampling resolution is insufficient to adequately capture the functional trajectories, and the truncated KL risk is instead dominated by $\underline{N}^{-\theta'}+R^{-1}\sum_{r=1}^{R}\mathrm{tr}\left(\boldsymbol{A}^{*(r)}-\boldsymbol{I}_K\right)$, where both the sampling frequency $\underline{N}$ and the average trace of the CM jointly determine the rate of convergence. Notably, N^* increases with n, implying that achieving faster convergence necessitates a higher sampling frequency as the sample size grows.*

5. Numerical studies

5.1. Implementation

Our proposed neural networks are implemented using `Python` within the `PyTorch` framework. Specifically, the feedforward neural network architecture is defined by the number of hidden layers L and the number of neurons γ_l in the Lth hidden layer for $l \in [L]$. For our numerical studies, we consider $L \in \{2, 3\}$ and $\gamma_l = \{32, 64, 128, 256\}$ for all $l \in [L]$. Throughout this work, we choose the number of basis functions $J \in \{3, 5, 10, 20, 30\}$. Throughout all experiments, the number of training epochs and the batch size are fixed prior to training. For practical purposes, we select a moderately large epoch number to ensure stability of results. The batch size is set to $2^{\lfloor \log n \rfloor}$, where $\lfloor \cdot \rfloor$ denotes the floor function. To select the optimal neural network architecture and truncation vectors, we employ Algorithm 1. For all numerical studies, 90% of the data are randomly assigned to the training set and 10% to the validation set. To optimize the training process with respect to objective (3.4), we use the `Adam` optimizer with a learning rate of $\alpha = 0.001$ to train the initial model. We adopt a Fourier basis to extract functional projection scores. The Python implementation code for the proposed `afDNN` algorithms, together with examples, is available on `GitHub` (https://github.com/ShuoyangWang/afDNN). Additional simulation results are provided in Supplementary Material.

5.2. Simulation studies

In this section, we present numerical results to demonstrate the superior performance of the proposed `afDNN` method. We consider two types of annotators as considered by Tanno et al. (2019): (i) *hammer-spammer annotators* provide the correct label with a certain probability and otherwise assign a label uniformly at random; and (ii) *pairwise-flipper annotators*, who are correct with a certain probability, and otherwise flip the label of each class to another class, where the target class is selected uniformly at random for each original label. For each

annotator type and skill level, we create a group of $R = 5$ annotators by generating CM from the corresponding distribution.

For Models 1 – 4, we consider 2D functional imaging data with fewer generating basis functions. Specifically, for $k = 1, 2, 3$, we generate image data $X_i^{(k)}(s, s') = \sum_{j=1}^{5} \xi_{ij}^{(k)} \psi_j(s, s')$, for $s, s' \in [0, 1]$. Let $\psi_1(s, s') = s$, $\psi_2(s, s') = s'$, $\psi_3(s, s') = ss'$, $\psi_4(s, s') = s^2 s'$, and $\psi_5(s, s') = ss'^2$. Define $\mathbf{1}_k$ to be a $k \times 1$ vector with all the elements one. We specify the distribution of $\xi_{ij}^{(k)}$'s as following.

Model 1 (Gaussian): $\left(\xi_{i1}^{(k)}, \ldots, \xi_{i5}^{(k)}\right)^\intercal \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ for $k = 1, 2, 3$, and they are independent of each other, where $\boldsymbol{\mu}_1 = (4, 4, 3, 3, 3)^\intercal$, $\boldsymbol{\Sigma}_1^{1/2} = \text{diag}\,(8, 7, 6, 5, 4)$, $\boldsymbol{\mu}_2 = -\mathbf{1}_5$, $\boldsymbol{\Sigma}_2^{1/2} = \text{diag}\,(5, 4, 3, 2, 1)$, $\boldsymbol{\mu}_3 = \mathbf{0}_5$, $\boldsymbol{\Sigma}_3^{1/2} = \text{diag}\,(2.5, 2, 1.5, 1, 0.5)$.

Model 2 (Mixed 1): $\left(\xi_{i1}^{(k)}, \ldots, \xi_{i5}^{(k)}\right)^\intercal \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ for $k = 1, 2$, and $\xi_{ij}^{(3)} \sim t_{2j+1}(\nu_j)$, where $\boldsymbol{\mu}_1 = -\mathbf{1}_5$, $\boldsymbol{\Sigma}_1^{1/2} = \text{diag}\,(5, 4, 3, 2, 1)$, $\boldsymbol{\mu}_2 = \mathbf{0}_5$, $\boldsymbol{\Sigma}_2^{1/2} = \text{diag}\left(\frac{5}{2}, 2, \frac{3}{2}, 1, \frac{1}{2}\right)$, $(\nu_1, \ldots, \nu_5)^\intercal = 3 \cdot \mathbf{1}_5$.

Model 3 (Mixed 2): $\left(\xi_{i1}^{(1)}, \ldots, \xi_{i5}^{(1)}\right)^\intercal \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, $\xi_{ij}^{(2)} \sim t_{j+1}(\nu_{2j})$, $\xi_{ij}^{(2)} \sim t_{2j+1}(\nu_{3j})$, where $\boldsymbol{\mu}_1 = \mathbf{0}_5$, $\boldsymbol{\Sigma}_1^{1/2} = \text{diag}\left(\frac{5}{2}, 2, \frac{3}{2}, 1, \frac{1}{2}\right)$, $(\nu_{21}, \ldots, \nu_{25})^\intercal = \mathbf{1}_5$, $(\nu_{31}, \ldots, \nu_{35})^\intercal = 3 \cdot \mathbf{1}_5$.

Model 4 (Mixed 3): $\xi_{ij}^{(1)} \sim Exp(r_{1j})$, $\xi_{ij}^{(2)} \sim t_{2j+1}(\nu_{2j})$, and $\left(\xi_{i1}^{(3)}, \ldots, \xi_{i5}^{(3)}\right)^\intercal \sim N(\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$, where $(r_{11}, \ldots, r_{15})^\intercal = (0.1, 0.3, 0.5, 0.7, 0.9)^\intercal$, $(\nu_{21}, \ldots, \nu_{25})^\intercal = 3 \times \mathbf{1}_5$, $\boldsymbol{\mu}_3 = \mathbf{0}_5$, $\boldsymbol{\Sigma}_3^{1/2} = \text{diag}\,(2.5, 2, 1.5, 1, 0.5)$.

To further demonstrate the performance of our proposed method in 2D functional data, we additionally consider Model 5 with 20 basis functions.

Model 5 (Complex basis functions): $\xi_{ij}^{(1)} \sim Exp(r_{1,j})$, $\xi_{ij}^{(2)} \sim t_{2j+1}$, and $\left(\xi_{i1}^{(3)}, \ldots, \xi_{i20}^{(3)}\right)^\intercal \sim N(\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$, where $(r_{1,1}, \ldots, r_{1,20})^\intercal = (j/10)_{j\in[20]}^\intercal$, $\boldsymbol{\mu}_3 = \mathbf{0}_{20}$, $\boldsymbol{\Sigma}_3^{1/2} = \text{diag}\,(5, 4.75, 4.5, 4.25, 4, 1, \ldots, 1)$. For $k = 1, 2, 3$, we generate image data $X_i^{(k)}(s, s') = \sum_{j=1}^{2} 0\xi_{ij}^{(k)} \psi_j(s, s')$, for $s, s' \in [0, 1]$. Let $\psi_1(s, s') = s$, $\psi_2(s, s') = s'$, $\psi_3(s, s') = ss'$, $\psi_4(s, s') = s^2 s'$, $\psi_5(s, s') = ss'^2$, $\psi_6(s, s') = s^2$, $\psi_7(s, s') = s'^2$, $\psi_8(s, s') = s^3$, $\psi_9(s, s') = s'^3$, $\psi_{10}(s, s') = s^4$, $\psi_{11}(s, s') = s^3 s'$, $\psi_{12}(s, s') = s^2 s'^2$, $\psi_{13}(s, s') = ss'^3$, $\psi_{14}(s, s') = s'^4$, $\psi_{15}(s, s') = s^5$,

$\psi_{16}(s,s') = s^4s'$, $\psi_{17}(s,s') = s^3s'^2$, $\psi_{18}(s,s') = s^2s'^3$, $\psi_{19}(s,s') = ss'^4$, and $\psi_{20}(s,s') = s'^5$.

Additionally, we examine our method in a 3D functional imaging data setting with 18 basis functions.

Model 6 (Complex basis functions for 3D data): $\xi_{ij}^{(1)} \sim Exp(r_{1j})$, $\xi_{ij}^{(2)} \sim t_{2j+1}$, and $\left(\xi_{i1}^{(3)}, \ldots, \xi_{i18}^{(3)}\right)^\intercal \sim N(\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$, where $(r_{1,1}, \ldots, r_{1,18})^\intercal = (j/10)_{j\in[20]}^\intercal$, $\boldsymbol{\mu}_3 = \mathbf{0}_{20}$, and $\boldsymbol{\Sigma}_3^{1/2} = \text{diag}\,(5, 4.75, 4.5, 4.25, 4, 1, \ldots, 1)$. For $k = 1, 2, 3$, we generate image data $X_i^{(k)}(s,s',s'') = \sum_{j=1}^{2} 0\xi_{ij}^{(k)}\psi_j(s,s',s'')$, for $s, s' \in [0,1]$. Let $\psi_1(s,s',s'') = s$, $\psi_2(s,s',s'') = s'$, $\psi_3(s,s',s'') = s''$, $\psi_4(s,s',s'') = ss'$, $\psi_5(s,s',s'') = ss''$, $\psi_6(s,s',s'') = s's''$, $\psi_7(s,s',s'') = s^2$, $\psi_8(s,s',s'') = s'^2$, $\psi_9(s,s',s'') = s''^2$, $\psi_{10}(s,s',s'') = s^2s'$, $\psi_{11}(s,s',s'') = s^2s''$, $\psi_{12}(s,s',s'') = s'^2s''$, $\psi_{13}(s,s',s'') = ss'^2$, $\psi_{14}(s,s',s'') = ss''^2$, $\psi_{15}(s,s',s'') = s's''^2$, $\psi_{16}(s,s',s'') = s^3$, $\psi_{17}(s,s',s'') = s'^3$, and $\psi_{18}(s,s',s'') = s''^3$.

For each model, we observe image data on 3×3, 5×5, 10×10, 20×20, and 50×50 grids over the domains $[0,1]^2$ for 2D data and $[0,1]^3$ for 3D data, corresponding to sampling frequencies $\underline{N} = 9, 25, 100, 400, 2500$ (2D) and $\underline{N} = 27, 125, 1000, 8000, 125000$ (3D), respectively. These setups cover a wide range of functional data types and grid resolutions, thereby providing strong support for the applicability of our method in the real data analysis. These settings reflect a progression from sparse to dense functional observations. The sample size is set to 100 for each class, and an additional 30 subjects per class are generated to form the test set for evaluating classifier performance. The Fourier basis is used to extract projection scores. We set aside 10% of the training examples as a validation set.

5.2.1. Hammer-spammer noisy labels

We corrupt the training and validation sets according to true $K \times K$ CM $\boldsymbol{A}_\epsilon^*$, defined by, $(\boldsymbol{A}_\epsilon^*)_{jj} = 1-\epsilon$ and $(\boldsymbol{A}_\epsilon^*)_{jk} = \frac{\epsilon}{K-1}$ for $j \neq k$, where K is the number of the class. For instance, $\boldsymbol{A}_{0.5}^*$ represents heavy label noise, with approximately half of the instances mislabeled, while $\boldsymbol{A}_{0.1}^*$ reflects mild label noise, affecting roughly 10% of instances. Given ground truth labels,

noisy annotations are generated for each annotator based on their associated CM. These noisy labels are used during training.

To assess the performance of our proposed method, we compare it against several baseline approaches. The first benchmark is the DNN classifier (Oracle) proposed by Wang and Cao (2024), where each training instance is labeled with its true class. This classifier is trained on noise-free data and thus provides an upper bound on achievable classification accuracy. Given that the experimental settings (Models 1–4) align with those in Wang and Cao (2024), the oracle classifier has been demonstrated to significantly outperform several widely used classification methods, which include the ℓ_1-penalized Fisher's discriminant analysis approach from Mai et al. (2019), the penalized linear discriminant analysis classifier in Witten and Tibshirani (2011), a vanilla CNN, and random forest classifiers. The second baseline (All sample) is a vanilla DNN classifier trained on duplicated samples, where individuals with conflicting labels from multiple annotators are treated as separate instances. The third baseline (Majority vote) is a vanilla DNN classifier assuming the true label is the most frequently assigned label for each instance. To the best of our knowledge, standard CNN or Vision Transformer architectures in the literature do not explicitly model annotator-specific noise and thus are not directly aligned with our goal of jointly learning a classifier and annotator CMs. Hence, we compare our method to CNN-EM, the noisy label learning method introduced by Tanno et al. (2019), which integrates a CNN with the expectation-maximization (EM) algorithm to account for label noise. Tanno et al. (2019) demonstrated that their method consistently achieves superior or comparable performance in both classification accuracy and CM estimation relative to six state-of-the-art approaches, including MBEM (Khetan et al., 2017), generalized MBEM (Raykar et al., 2009), Single CM (Sukhbaatar et al., 2014), Weighted Doctor Net (Guan et al., 2018), Soft-bootstrap (Reed et al., 2014), and a vanilla CNN (Reed et al., 2014). Given its strong empirical results, we restrict our comparisons to the CNN-EM method from Tanno et al. (2019).

We assess CM estimation by computing the average Frobenius norm (F-norm) between each true CM and its estimate across annotators. This metric is normalized to be in the range [0, 1] and computed as $R^{-1}K^{-1}\sum_{r\in[R]}\sum_{i,j\in[K]}\|a_{ij}^{(r)}-\widehat{a}_{ij}^{(r)}\|^2$.

To ensure fair comparison, all methods use the same DNN architecture. For the CNN-EM method, the same DNN structure is employed after flattening the input. Specifically, we first construct a convolutional layer consisting of 32 filters of size 3×3, followed by The ReLU activation that is applied to introduce non-linearity. Next, a max pooling layer of size of 2×2 is added to reduce the spatial dimensions of the data while retaining crucial features. The model then proceeds with a second convolutional layer with 64 filters of size 3×3, another ReLU activation, and a second max pooling layer of size 2×2. It is important to note that CNNs are primarily designed for processing high-resolution images to effectively extract spatial features. When applied to low-resolution inputs such as 3×3 images (i.e., $\underline{N}=9$), techniques such as striding and max pooling become impractical due to the lack of meaningful spatial structures. As a result, CNN approach is not applicable in the $m=9$ setting. Figures 1 and 2 presents boxplots of classification accuracy, demonstrating that our proposed method consistently outperforms all baseline methods for Models 1 to 4. Notably, across all contamination levels ($\epsilon=0.1,0.3,0.5$), the proposed `afDNN` achieves performance comparable to the oracle classifier, indicating its ability to accurately estimate CM and correctly classify new subjects. This finding is further supported by Figure 3, which reports significantly lower F-norm errors in the estimated CM compared to CNN-EM, especially under severe label noise ($\epsilon=0.5$) for Models 1 to 4. Due to space limitations, the comparison of classification accuracy and F-norm errors for Models 5 and 6 are provided in Section S2.2 in the Supplementary file.

5.2.2. Pairwise flipper noisy labels

In this section, we manually corrupt the training and validation sets using $\boldsymbol{A}_\epsilon^*$, defined by $(\boldsymbol{A}_\epsilon^*)_{jj}=1-\epsilon$. For the off-diagonal elements, we set $(\boldsymbol{A}_\epsilon^*)_{12}=(\boldsymbol{A}_\epsilon^*)_{23}=(\boldsymbol{A}_\epsilon^*)_{31}=\epsilon$, with all other entries set to zero. Models 7–12 are defined analogously to Models 1–6, respectively.

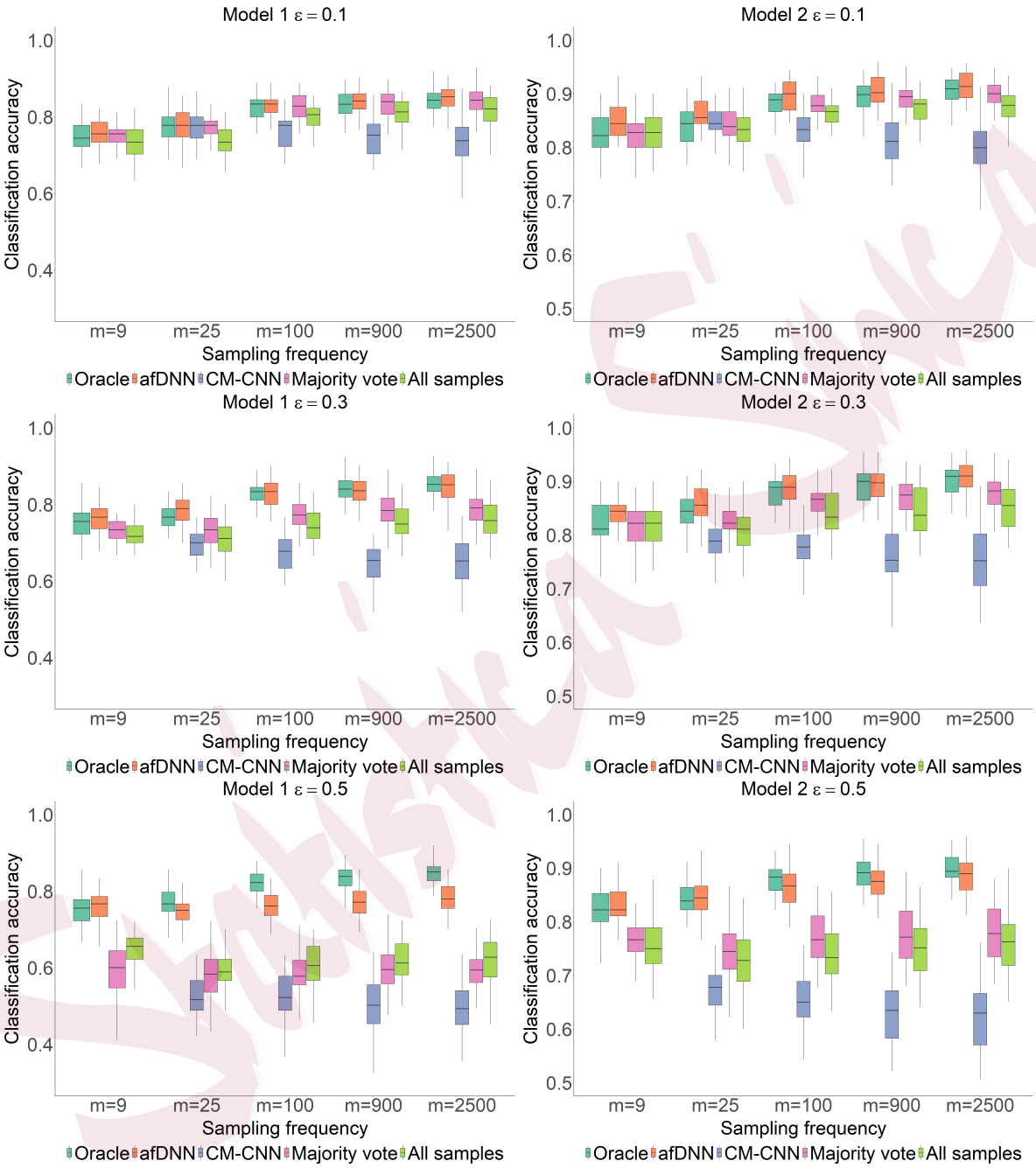

Figure 1: Classification accuracy for all methods across different sampling frequencies for Models 1 – 2. From left to right, the rows represent the results for Model 1 – 2, respectively. From top to bottom, the columns correspond to $\epsilon = 0.1, 0.3$, and 0.5.

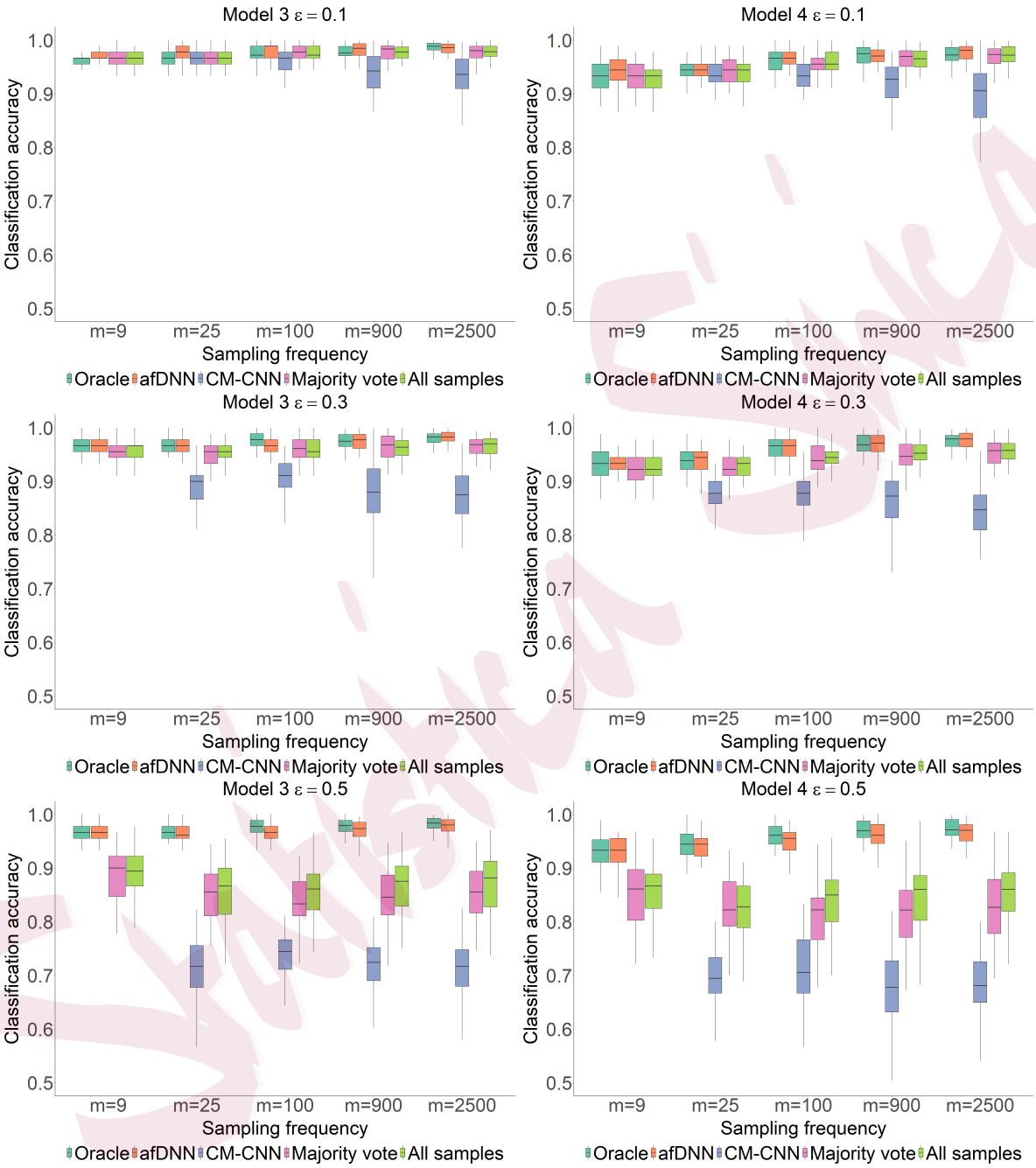

Figure 2: Classification accuracy for all methods across different sampling frequencies for Models 3 – 4. From left to right, the rows represent the results for Model 3 – 4, respectively. From top to bottom, the columns correspond to $\epsilon = 0.1, 0.3$, and 0.5.

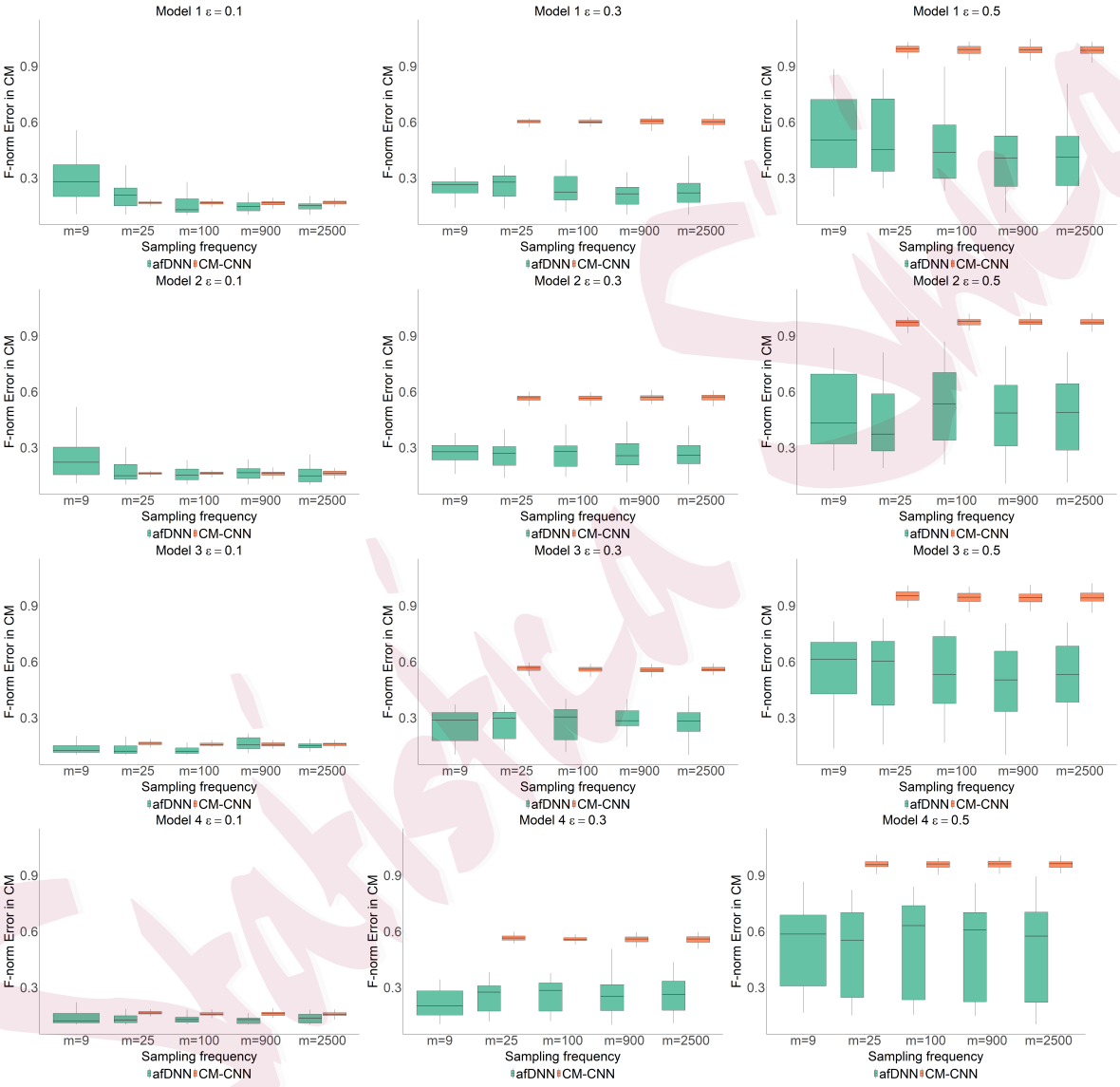

Figure 3: F-norm error in CM for all methods across different sampling frequencies for Models 1 – 4. From top to bottom, the rows represent the results for Model 1 – 4, respectively. From left to right, the columns correspond to $\epsilon = 0.1, 0.3$, and 0.5.

Figure S9 to S11 in Supplementary Material presents boxplots of classification accuracy for Models 7–12. Notably, for low noisy levels ($\epsilon = 0.1, 0.3$), the proposed `afDNN` achieves performance comparable to the oracle model and outperforms all other alternatives. When the noise level increases to $\epsilon = 0.5$, our proposed method continues to significantly outperform the alternatives but shows a performance drop relative to the oracle model, particularly in Model 8. This result further supports the theoretical conclusion in Theorem 1, which states that heavily biased CM can ultimately degrade classification performance. In addition, Figures S10 and S11 in Supplementary Material show that our method maintains a significantly lower F-norm bias in the estimated CM compared to CM-CNN, further highlighting its robustness in handling label noise.

6. Real data analysis

6.1. Chest radiographs dataset with human annotations

Pneumoconioses are occupational interstitial lung diseases caused by the inhalation of mineral dust particles. Expert classification of chest radiographs is a critical tool for protecting workers exposed to coal dust. However, both inter-reader and intra-reader variability remain major concerns, affecting the consistency and reliability of diagnostic assessments. Although certain training programs aim to improve the proficiency of certified physicians, significant challenges persist. These include the limited number of certified B-readers—currently fewer than 300 in the U.S.—as wells as continued concerns about interpretive variability, and potential financial conflicts of interest.

The radiographs used in this study were obtained from the image repository maintained by the National Institute for Occupational Safety and Health (NIOSH). The dataset includes images from both male and female coal workers in U.S. coal mining populations, with data collected since 2000. A custom dataset of $1,042$ chest radiographs was curated, comprising both pneumoconiosis and non-pneumoconiosis cases. All radiographs are posterior-anterior

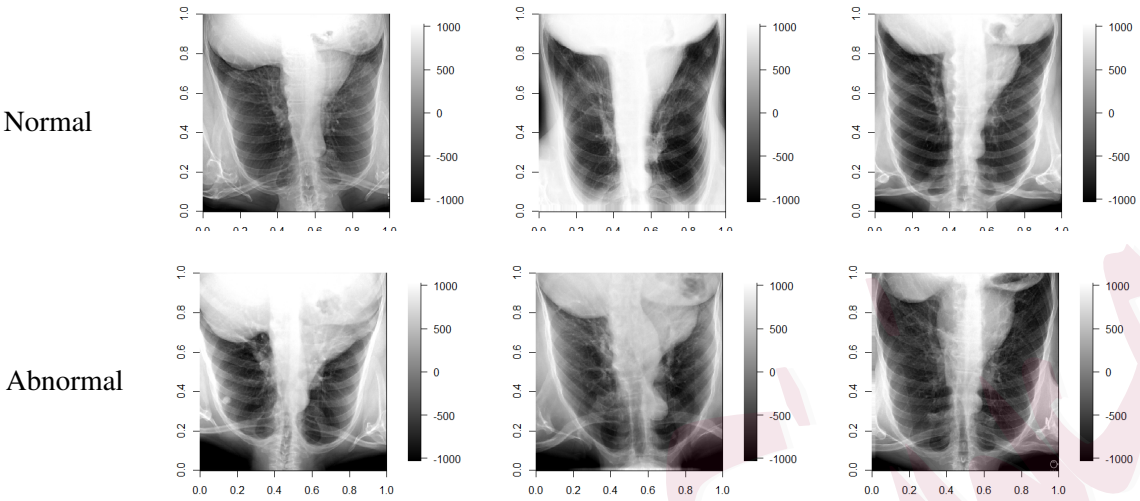

Figure 4: Chest image samples for normal individuals (top row) and abnormal individuals (bottom row).

images and were preprocessed by the MIDI lab at Michigan State University to extract hand-crafted radiomic features. Preprocessing included resizing the DICOM (Digital Imaging and Communications in Medicine) images to a uniform resolution of 512×512 pixels. Figure 4 provides examples from both normal and abnormal groups. Each radiograph was reviewed by between 1 and 21 readers, including repeated assessments from the same reader, to evaluate the presence of small opacities. The severity of abnormalities was rated on a scale from 0 to 3, where 0 represents a normal radiograph and 3 indicates the most severe abnormality. On average, each subject was annotated by 3.69 readers, with a median of 2. Since definitive ground-truth labels are nearly impossible to obtain, we assume that consensus readings were established through panel discussions among certified B-readers. Based on these consensus labels, regarded as “ground-truth" labels, we observed that 21% of individuals in the normal group were mislabeled at least once, while 42% of individuals in the abnormal group experienced mislabeling.

Using the expert ratings and accounting for potential misclassification mechanisms, we conducted a classification analysis to evaluate the performance of the proposed `afDNN` method in comparison with alternative approaches. A test dataset which was also provided by the

Table 1: Classification accuracy for chest radiographs dataset.

	Oracle	`afDNN`	ALL sample	Majority vote	CM-CNN
Accuracy	**0.73**	**0.73**	0.69	0.64	0.59

MIDI lab, comprises 131 chest radiographs, with the median expert ratings serving as the ground-truth labels. Table 1 reports the classification accuracy of each method. Both the oracle and `afDNN` models achieved the highest accuracy rates, highlighting the effectiveness of the proposed approach in mitigating the impact of label noise.

6.2. ADNI database with machine annotations

To implement the analysis of the chest radiographs data, we further apply the proposed method to analyze a dataset that has clean labels, where we artificially generate synthetic data with varying degrees of label noise.

6.2.1. Data description

Data for this study were acquired from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (`http://adni.loni.usc.edu`), a longitudinal, multicenter effort aimed at developing clinical, imaging, genetic, and biochemical biomarkers for the early detection and progression monitoring of Alzheimer's disease (AD). Each scan was reformatted into a voxel grid of $79 \times 95 \times 68$, yielding 68 two-dimensional slices of 79×95 pixels per subject. From the ADNI repository, we extracted ^{18}F-fluorodeoxyglucose positron emission tomography (FDG–PET) scans for three diagnostic cohorts: 79 patients diagnosed with AD, 45 individuals with Early Mild Cognitive Impairment (EMCI), and 101 cognitively normal (CN) subjects. To ensure standardized inputs, we utilized pre-processed and spatially normalized PET images from the ADNI-1 and ADNI-GO cohorts. AD and control participants were selected from ADNI-1, while EMCI participants were drawn from ADNI-GO. To ensure a uniform comparison and minimize confounding variables related to disease progression, analysis was restricted to baseline scans for all groups. Figure S12 in Supplementary Material

illustrates the averaged images of the 20th, 40th and 60th slices for the AD, EMCI, and CN groups. To ensure robust model evaluation, the dataset was randomly partitioned into training, validation, and hold-out sets using a $6:1:3$ ratio, repeated across 100 independent trials to ensure balanced sampling and stability of results. Additional background details and numerical results of the ADNI dataset are provided in Supplementary Material.

6.2.2. Noise generation

Since the ground truth labels in the ADNI image dataset are available but noisy annotations are not, we manually corrupted the labels in the training and validation sets using the hammer-spammer mechanism described in Section 5.2.1. Specifically, noisy labels are generated according to a true 3×3 CM $\boldsymbol{A}_\epsilon^*$, where $(\boldsymbol{A}_\epsilon^*)_{ii} = 1-\epsilon$ and $(\boldsymbol{A}_\epsilon^*)_{ij} = 0.5\epsilon$ for $\epsilon = 0.1, 0.3$, or 0.5. For each annotator, a group of $R = 5$ annotators is created by sampling CM from the specified distribution.

6.2.3. Classification accuracy

We implemented `afDNN` and several competing methods as in simulation studies using six different 2D slices from the 10th to the 60th. As reported in Table 2 and Table S1 in Supplementary Materials, DNN-based methods generally achieved slightly higher classification accuracy compared to CM-CNN. Remarkably, our proposed `afDNN` maintained superior performance even under severe label contamination ($\epsilon = 0.5$). Furthermore, classification accuracy was observed to decline with slice index, which is consistent with established neuropathological findings that Alzheimer's disease primarily affects regions such as the hippocampus, entorhinal cortex, and cerebral cortex—regions roughly corresponding to the first 25 slices. These results are particularly informative for neurologists, as they suggest that focusing on these early slices may enhance diagnostic differentiation among CN, EMCI, and AD subjects, thus improving early-stage diagnosis and monitoring strategies.

We also extend our ADNI analysis to a fully 3D setting, where we treat each PET scan as

Table 2: Averaged classification accuracy with standard errors in brackets for ADNI 2D brain images with different values of ϵ for the 20th, 40th, and 60th slices.

Methods	$\epsilon = 0.1$			$\epsilon = 0.3$			$\epsilon = 0.5$		
	20-th	40-th	60-th	20-th	40-th	60-th	20-th	40-th	60-th
Oracle	**0.58** (0.05)	0.52 (0.04)	0.50 (0.03)	**0.58** (0.05)	**0.52** (0.04)	0.50 (0.03)	**0.58** (0.05)	**0.52** (0.04)	**0.50** (0.03)
afDNN	0.57 (0.03)	**0.55** (0.01)	**0.54** (0.01)	0.49 (0.02)	**0.52** (0.02)	**0.51** (0.02)	0.48 (0.02)	0.48 (0.02)	0.44 (0.02)
All sample	0.53 (0.05)	0.49 (0.01)	0.47 (0.03)	0.48 (0.02)	0.51 (0.02)	0.44 (0.01)	0.43 (0.04)	0.48 (0.04)	0.41 (0.02)
Majority vote	0.46 (0.05)	0.46 (0.02)	0.51 (0.01)	0.47 (0.04)	0.45 (0.02)	0.45 (0.01)	0.36 (0.01)	0.36 (0.00)	0.40 (0.03)
CM-CNN	0.48 (0.05)	0.46 (0.04)	0.40 (0.05)	0.46 (0.06)	0.45 (0.06)	0.38 (0.05)	0.39 (0.07)	0.40 (0.06)	0.36 (0.07)

a volumetric functional object and extract scores using 3D basis functions defined over the entire brain volume. Table S2 in the Supplementary file presents the classification accuracy comparison results. We find that all methods achieve slightly better classification accuracy in the 3D setting than the 2D slice-based analysis. Importantly, our proposed method continues to outperform and remains competitive with the oracle model that is trained using the true labels.

Since the PET images in the ADNI dataset are stored within a rectangular grid, background voxels are naturally included. While it is possible to apply established methods to extract the brain's precise geometry to exclude background voxels, we have not pursued this approach in the current study to avoid potentially introducing new issues, such as inadvertently removing useful information. Instead, we used the full volume as input at this stage. We acknowledge that refining the input to exclude background noise is an interesting direction for future work.

7. Discussion

In this paper, we introduce a new framework that integrates the annotator-specific noise modeling within a functional DNN architecture. Our method offers a practical and flexible solution for addressing classification tasks with noisy annotations, an increasingly common challenge in real-world applications. By aggregating noisy annotations through a carefully designed label correction algorithm, we leverage the powerful DNN in combination with a regularized cross-entropy loss to jointly learn annotator-specific behaviors individual annotator model and the underlying true label distribution. Notably, this is accomplished using only noisy observations without access to clean data with ground-truth labels. We provide theoretical

guarantees for the classification accuracy of our method, which accommodates both densely and sparsely observed imaging data. Experiments on image classification tasks with both simulated and real labels show that our method consistently outperforms or performs on par with the state-of-the-art methods.

To estimate the annotator-specific noise transition matrix, we assume that, conditional on the true label, the corruption process is independent of the input features, i.e., an instance-independent assumption. While this assumption simplifies the modeling process and allows us to obtain attractive theoretical results, it may not always hold in real applications. Instance-dependent annotation noise can be more realistic in featuring real-world scenarios, such as medical imaging, where the quality of images or the varying expertise levels of human annotators can greatly influence diagnostic outcomes. An interesting direction for future work is to extend our current development to accommodate to instance-dependent annotation noise. Modeling this general case is expected to enhance the applicability of our introduced framework, though more refined techniques are needed for the establishment of theoretical guarantees. While there are established methods for extracting the brain's precise geometry to exclude background voxels, we have treated the full volume as the input for this stage of our study. We acknowledge that refining the input to exclude background noise is a valuable direction, and we intend to explore this in future work. Furthermore, it is interesting to extend our development to accommodate covariate measurement error (Sun and Yi, 2026b,a), a topic that has attracted extensive interest in the literature of measurement error models (Carroll et al., 2006; Yi, 2017; Yi et al., 2021).

Acknowledgments

We thank the Editor, the Associate Editor, and the two anonymous reviewers for their constructive comments on the initial submission.

Grace Y. Yi is Canada Research Chair in Data Science (Tier 1). Her research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC)

and the Canada Research Chair Program. Guanqun Cao's research is partially supported by the National Science Foundation under Grants DMS-2413301, CNS-2319342 and CNS-2319343.

ADNI data used our analysis of this article were obtained from the Alzheimers Disease Neuroimaging Initiative (ADNI) database (`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`.

References

Arpit, D., S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien (2017). A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pp. 233–242. PMLR.

Asman, A. J. and B. A. Landman (2011). Robust statistical label fusion through consensus level, labeler accuracy, and truth estimation (collate). *IEEE Transactions on Medical Imaging 30*(10), 1779–1794.

Asman, A. J. and B. A. Landman (2012). Formulating spatially varying performance in the statistical fusion framework. *IEEE Transactions on Medical Imaging 31*(6), 1326–1336.

Bauer, B. and M. Kohler (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., A. Cuevas, and J. L. Torrecilla (2018). On the use of reproducing kernel hilbert spaces in functional classification. *Journal of the American Statistical Association 113*(523), 1210–1218.

Bos, T. and J. Schmidt-Hieber (2022). Convergence rates of deep relu networks for multiclass classification. *Electronic Journal of Statistics 16*, 2724–2773.

Cai, T. T. and L. Zhang (2019a). A convex optimization approach to high-dimensional sparse

quadratic discriminant analysis. *arXiv:1912.02872*.

Cai, T. T. and L. Zhang (2019b). High dimensional linear discriminant analysis: optimality, adaptive algorithm and missing data. *Journal of the Royal Statistical Society (Series B) 81*(4), 675–705.

Carroll, R. J., D. Ruppert, L. A. Stefanski, and C. M. Crainiceanu (2006). *Measurement Error in Nonlinear Models: A Modern Perspective*. Chapman and Hall/CRC.

Chen, Z., H. Wang, H. Sun, P. Chen, T. Han, X. Liu, and J. Yang (2021). Structured probabilistic end-to-end learning from crowds. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 1512–1518.

Dai, X., H.-G. Müller, and F. Yao (2017). Optimal Bayes classifiers for functional data and density ratios. *Biometrika 104*(3), 545–560.

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

Grigsby, J. E., K. Lindsey, and D. Rolnick (2023). Hidden symmetries of relu networks. In A. C. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett (Eds.), *Proceedings of the 40th International Conference on Machine Learning*, Volume 202 of *Proceedings of Machine Learning Research*, pp. 11734–11760. PMLR.

Guan, M., V. Gulshan, A. Dai, and G. Hinton (2018). Who said what: Modeling individual labelers improves classification. In *Proceedings of the AAAI conference on artificial intelligence*, Volume 32.

Guo, H., B. Wang, and Y. G. Yi (2023). Label correction of crowdsourced noisy annotations with an instance-dependent noise transition model. *Advances in Neural Information Processing Systems 36*, 347–386.

Han, B., I. W. Tsang, L. Chen, J. T. Zhou, and P. Y. Celina (2019). Beyond majority voting: A coarse-to-fine label filtration for heavily noisy labels. *IEEE Transactions on Neural Networks and Learning Systems 30*(12), 3774–3787.

Hastie, T. J. and R. J. Tibshirani (1990). *Generalized Additive Models*. Chapman & Hall/CRC.

Hu, T., Z. Shang, and G. Cheng (2020). Sharp rate of convergence for deep neural network classifiers under the teacher-student setting. *arXiv:2001.06892*.

Huang, C. and H. Zhu (2022). Functional hybrid factor regression model for handling heterogeneity in imaging studies. *Biometrika 109*(4), 1133–1148.

Ibrahim, S. and X. Fu (2021). Crowdsourcing via annotator co-occurrence imputation and provable symmetric nonnegative matrix factorization. In *International Conference on Machine Learning*, pp. 4544–4554. PMLR.

Ibrahim, S., T. Nguyen, and X. Fu (2023). Deep learning from crowdsourced labels: Coupled cross-entropy minimization, identifiability, and regularization. In *international Conference on Representation Learning*.

Khetan, A., Z. C. Lipton, and A. Anandkumar (2017). Learning from noisy singly-labeled data. *arXiv preprint arXiv:1712.04577*.

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

Li, H. and B. Yu (2014). Error rate bounds and iterative weighted majority voting for crowdsourcing. *arXiv preprint arXiv:1411.4086*.

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

Liu, Q., J. Peng, and A. T. Ihler (2012). Variational inference for crowdsourcing. *Advances in Neural Information Processing Systems 25*.

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

Mai, Q., Y. Yang, and H. Zou (2019). Multiclass sparse discriminant analysis. *Statistica Sinica 29*, 97–111.

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

Nishie, A., D. Kakihara, T. Nojo, K. Nakamura, S. Kuribayashi, M. Kadoya, K. Ohtomo, K. Sugimura, and H. Honda (2015). Current radiologist workload and the shortages in japan: how many full-time radiologists are required? *Japanese Journal of Radiology 33*, 266–272.

Park, J., J. Ahn, and Y. Jeon (2021). Sparse functional linear discriminant analysis. *Biometrika 109*(1), 209–226.

Raykar, V. C., S. Yu, L. H. Zhao, A. Jerebko, C. Florin, G. H. Valadez, L. Bogoni, and L. Moy (2009). Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 889–896.

Reed, S., H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich (2014). Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*.

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

Shieh, D. and R. T. Ogden (2023). Permutation-based inference for function-on-scalar regression with an application in PET brain imaging. *Journal of Nonparametric Statistics 35*(4), 820–838.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15*, 1929–1958.

Sukhbaatar, S., J. Bruna, M. Paluri, L. Bourdev, and R. Fergus (2014). Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*.

Sun, Y. and G. Y. Yi (2026a). Debiased estimation and variable selection under function-on-scalar linear regression models with ultrahigh-dimensional covariates subject to measurement error. *Bernoulli*, To appear.

Sun, Y. and G. Y. Yi (2026b). Estimation and variable selection under the function-on-scalar

linear model with covrirate measurement error. *Statistica Sinica 36(2)*.

Tanno, R., A. Saeedi, S. Sankaranarayanan, C. D. Alexander, and N. Silberman (2019). Learning from noisy labels by regularized estimation of annotator confusion. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11244–11253.

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

Wang, S. and G. Cao (2024). Multiclass classification for multidimensional functional data through deep neural networks. *Electronic Journal of Statistics 18*, 1248–1292.

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

Wang, S., G. Cao, and Z. Shang (2023). Deep neural network classifier for multi-dimensional functional data. *Scandinavian Journal of Statistics 50*, 1667–1686.

Wang, S., Z. Shang, G. Cao, and S. J. Liu (2024). Optimal classification for functional data. *Statistica Sinica 34*, 1545–1564.

Wang, Y., G. Wang, L. Wang, and R. T. Ogden (2020). Simultaneous confidence corridors for mean functions in functional data analysis of imaging data. *Biometrics 76*(2), 427–437.

Warfield, S. K., K. H. Zou, and W. M. Wells (2004). Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation. *IEEE Transactions on Medical Imaging 23*(7), 903–921.

Wei, H., R. Xie, L. Feng, B. Han, and B. An (2022). Deep learning from multiple noisy annotators as a union. *IEEE Transactions on Neural Networks and Learning Systems 34*(12), 10552–10562.

Whitehill, J., T.-f. Wu, J. Bergsma, J. Movellan, and P. Ruvolo (2009). Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems 22*.

Witten, D. M. and R. Tibshirani (2011). Penalized classification using Fisher's linear dis-

criminant. *Journal of the Royal Statistical Society. Series B. Statistical Methodology 73*(5), 753–772.

Yi, G. Y. (2017). *Statistical Analysis with Measurement Error or Misclassification: Strategy, Method and Application*. Springer, New York.

Yi, G. Y., A. Delaigle, and P. Gustafson (2021). *Handbook of measurement error models*. Chapman & Hall/CRC, Boca Raton, FL.

Zhang, L., R. Tanno, M.-C. Xu, C. Jin, J. Jacob, O. Ciccarelli, F. Barkhof, and C. D. Alexander (2020). Disentangling human error from the ground truth in segmentation of medical images. *Advances in Neural Information Processing Systems 1321*, 15750–15762.

Zhang, Y., X. Chen, D. Zhou, and M. I. Jordan (2016). Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *Journal of Machine Learning Research 17*(102), 1–44.

Zhu, H., J. Fan, and L. Kong (2014). Spatially varying coefficient model for neuroimaging data with jump discontinuities. *Journal of the American Statistical Association 109*(507), 1084–1098.