Deep Learning for Multi-Dimensional Functional Data

Carter Hall* Connor McNeill* Kris Wilson*

December 10, 2025

1 Introduction

Functional data analysis (FDA) is an important area of statistics research. It is related to providing insights into curves, surfaces, images, or really anything within a continuous domain. It can deal with one-dimensional data, but recently there has been a developing need and interest for higher-dimensional functional data analysis, especially with the rise of image analysis. Functional data is characterized through inherent dependence and smoothness observed within each data curve.

Per Wang et al. (2024), there are three main areas in FDA: regression, classification, and representation. Functional data regression focuses on uncovering and understanding the underlying patterns in data that evolve over a continuum. Classification focuses on categorizing functional observations into distinct groups or classes. Lastly, representation focuses on efficiently identifying the underlying structure of functional data. We will explore all three areas through our project.

As mentioned, one of the most common forms of two-dimensional functional data is images. As such, it's important to understand how image data is structured. We can model the image as a function

$$f(x, y) = \text{intensity at spatial location}(x, y)$$

where the spatial location is a pixel in the image. The intensity of the pixel is measured on a scale of 0 to 255. The image below (Figure 1) shows the Fashion-MNIST data, which is used to test classifiers.

Figure 1: This image is selected from the Fashion-MNIST dataset. (Wang et al., 2024)

*North Carolina State University

Previous methods have focused on using single-layer neural networks. However, there are several advantages to utilizing deep neural networks (DNNs) which are neural networks with 2 or more hidden layers. First, DNNs thrive in high-dimensional spaces. This means that the so-called "curve of dimensionality" is not an issue when they are utilized. Second, deep learning models also offer a lot of flexibility and cater to a variety of different functional data structures. Lastly, it's important to note that DNNs have better predictive accuracy and are scalable.

In recent years, deep neural networks (DNNs) have enjoyed success in medical imaging analysis. This is partially attributed to the unstructured, heterogeneous data found in medical imaging analysis. Additionally, for both single-layer and multi-layer neural networks, DNNs were demonstrated to be able to approximate arbitrarily nonparametric models, a property known as universal approximation. While this work was done in the 1990s, recent work was done to establish convergence rates for nonparametric regression, in nearly every case imaginable: smooth regression, non-smooth regression, semiparametric regression, quantile regression, variable selection, and low-dimensional data structures. As apart of this work, we utilize deep neural networks to estimate functions in a regression model through multi-layer feed-forward neural networks. The theoretical results include establishing bounds on both globally smooth functions and also piecewise smooth functions.

In this report, first we will explore the three methods that we will then use in the simulation study and real data analysis. Since these methods all have different uses, they will not be compared to one another but instead will be compared to competing methods.

2 Methods

2.1 Functional Deep Neural Networks

As stated in the introduction, one of the main areas of FDA is functional data classification. We often want to classify a data function based on training samples. One example that shows this importance is brain imaging: we want to have methodology that allows input of an image and the output of the model might be something like the stage of the patient's illness. However, imaging is 2-dimensional (or 3-dimensional) data, which previously was not explored for methodology. Current methods focus only on one-dimensional functional data classification. Additionally, these methods often require that the data function follows a Gaussian process - which is also violated in practice. We need approaches that can work for non-Gaussian data as a lot of real-world data is non-Gaussian. Additionally, in current methods when data distributions are non-Gaussian, the decision boundary can often not be accurately be recovered. Due to these reasons, we need new methodology that can work on multi-dimensional non-Gaussian data.

Wang et al. (2023) proposed the approach of Functional Deep Neural Networks (FDNNs) in order to help solve these problems with previous methods. FDNN, like other classification methods, starts by doing functional principle components analysis (FPCA) which allows us to perform dimension reduction by getting an eigendecomposition of the data. Once this is done, we then train a Deep Neural Networks-based classifier on the principle components

as well as their corresponding class labels. This part is considered to be supervised learning since we train the model on the known class labels. We'll explore how to utilize the FDNN method in the real data analysis, but first we'll go through an overview on how the method works.

Suppose we observe n independent and identically distribution training samples. Let $X(s)$ be a random process and let $Y \in \{-1, 1\}$ be our class labels. $X(s)$ has unknown mean function $\mu_k(x)$ and unknown covariance function $\Omega_k(s, s')$ for $s, s' \in \mathcal{S}$. Define the sample covariance function

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

We then perform the Karhunen-Loeve decomposition for the sample covariance function and write the sample data function X_i, under $Y_i = k$, as

$$X_i(s) = \sum_{j=1}^{\infty} \hat{\xi}_{ij} \hat{\psi}_{kj}(s), i = 1, \ldots, n.$$

Intutively, $\hat{\boldsymbol{\xi}}^{(i)} := (\hat{\xi}_{i1}, \hat{\xi}_{i2}, \ldots)$ is an estimator of $\boldsymbol{\xi}^{(i)} = (\xi_{i1}, \xi_{i2}, \ldots)$ which are unobservable random coefficients of X_i with respect to the population basis ψ_{kj}. We will truncate $\boldsymbol{\xi}$ at J (which is chosen in order to best represent the data) and use these as classifiers.

We then train a DNN to estimate the log density ratio functional Q^* based on the $\hat{\boldsymbol{\xi}}_J^{(i)}$'s. Let σ denote the RELU function, and define the shift activation function for any real vectors $\boldsymbol{V}$ and $\boldsymbol{y}$ of dimension w as

$$\sigma_{\boldsymbol{V}}(\boldsymbol{y}) = (\sigma(y_1 - v_1), \ldots, \sigma(y_w - v_w)).$$

With J inputs, L hidden layers, and p_ℓ nodes on the lth hidden layer, we fit our DNN.

Given the training data $\{(\boldsymbol{\xi}_J^{(1)}, Y_1), \ldots, (\boldsymbol{\xi}_J^{(n)}, Y_n)\}$ let

$$\hat{f}_\phi(\bullet) = \arg \min_{f \in \mathcal{F}(L,J,p,B)} \frac{1}{n} \sum_{i=1}^{n} \phi(f(\boldsymbol{\xi}_J^{(i)}, Y_i)),$$

where $\phi(x)$ is the hinge loss. The following then is our FDNN classifer:

$$\hat{G}^{FDNN}(X) = \begin{cases} 1, & \hat{f}_\phi(\boldsymbol{\xi}_J) \geq 0 \\ -1, & \hat{f}_\phi(\boldsymbol{\xi}_J) < 0. \end{cases}$$

Wang et al. (2023) also discuss some theoretical properties as well. To briefly note this, the primary thing is that the minimax excess misclassification risk (MEMR) meets the condition that means that the classifier is minimax optimal. For more details and proofs, the paper goes into detail in section 4 and the appendix.

2.2 Covariance Estimation with Neural Networks (CovNet)

Functional data are, naturally, objects that represent some quantity that can lie in a one-dimensional or multidimensional space. Common examples include a temperature curve (one dimension), spatial surfaces, such as images (two dimensions), and brain activation over space and time (three dimensions). Rather than be scalar or vector quantities, in truth these are 'infinite-dimensional' variables that exist through some continuum; as such, modeling these types of data through discretization into independent units is unrealistic. Modeling covariance *operators*, as opposed to a covariance matrix, has the following advantages:

1. **Grid/resolution-invariant**: A matrix would exponentially increase in number of entries with the dimension.
2. **Defines expansions in terms of eigenfunctions**: The Karhunen-Loeve (KL) expansion allows us to decompose functional observations through an eigendecomposition that is defined by a population-level covariance operator.
3. **Estimability in high-dimensional scenarios**: If a function is discretized at K points and one has N samples, reliable estimation of a covariance matrix requires $N \gg K$. Regardless of dimensionality, a fine resolution requires in traditional methods the computation and storage of massive objects (e.g., matrices) on which operations (e.g., inversion) can become computationally prohibitive.
4. **Allows for consideration of scenario-specific properties**: Certain covariance operators exhibit stationarity (covariances only dependent on difference between locations) or separability (expressed as the product of coordinate-wise covariances). *However*, the assumption of separability is often violated in practice or is implicitly assumed, where tests have been developed toward evaluating its validity (Aston et al. (2017)).

To generalize previous approaches to estimation of covariance operators while remaining computationally tractable, Sarkar and Panaretos (2022) introduce *Covariance Networks* (CovNet). The innovations of their approach include

1. **Universal Approximation**: Up to an arbitrary precision, *any* covariance operator can be approximated (see Theorems 1 and 2 of Sarkar and Panaretos (2022)).
2. **Explicit Functional Form**: No interpolation (e.g., between grid points in a particular dimension) is required when using the proposed model.
3. **No Computation or Storage of High-Order Objects**: Fitting a CovNet is computationally tractable because of an explicit eigendecomposition that avoids the need to store and invert high-dimensional matrices or tensors.
4. **Theoretical Guarantees**: Consistency and rates of convergence are presented by the authors, where it is important to note that no structural assumption is made regarding any underlying operator.

Background. Before introducing the specific CovNet architectures presented in Sarkar and Panaretos (2022), we begin with some background to motivate the novelty of their contri-

butions. As the authors refer to Hsing and Eubank (2015) for additional explanation, we do the same. Let $\mathcal{Q} \subset \mathbb{R}^d$ be compact and the space of square-integrable, d-dimensional elements be $L_2(\mathcal{Q})$. Define $\mathcal{X} := \{X(\mathbf{u}) : \mathbf{u} \in \mathcal{Q}\} \in L_2(\mathcal{Q})$ to be a random field (a 'curve' if $d = 1$) and assume it has finite second moment $\mathbb{E}[\|\mathcal{X}\|^2] < \infty$, which gives the existence of mean and covariance operators.

The purpose of this extensive mathematical formalism is to equate the problem of estimating a covariance *operator* – an infinite-dimensional object – with an associated **kernel** which can be expressed with a finite eigendecomposition. Let $\mathcal{C}$ be this covariance operator, defined as a linear operator from $L_2(\mathcal{Q})$ onto itself given by

$$\mathcal{C}f(\mathbf{u}) = \int_{\mathcal{Q}} c(\mathbf{u}, \mathbf{v}) f(\mathbf{v}) d\mathbf{v}, \quad f \in L_2(\mathcal{Q}) \tag{1}$$

where

$$c(\mathbf{u}, \mathbf{v}) := \text{Cov}(X(\mathbf{u}), X(\mathbf{v})) \in L_2(\mathcal{Q} \times \mathcal{Q}) \tag{2}$$

is the associated covariance *kernel*. With the assumptions on $\mathcal{X}, \mathcal{Q}$, we have the Hilbert-Schmidt norm $|||\mathcal{C}|||_2 < \infty$ and the isometric (inner-product preserving) isomorphism (invertible map) $|||\mathcal{C}|||_2 \equiv \|c\|_{\mathcal{L}_2(\mathcal{Q} \times \mathcal{Q})}$ which allows us to treat the kernel as a coordinate representation of the infinite-dimensional covariance operator. As such, the mathematics that follows is framed in terms of the covariance kernel $c(\cdot, \cdot)$.

Proposed Models. CovNet comes in three different flavors: Shallow, Deep, and DeepShared. Simulations will focus on the latter of these models, though we present all types to motivate the discussion. As the name suggests, neural network architectures underscore the approximations of covariances. Sarkar and Panaretos (2022) use perceptrons – single-layer neural networks – governed by a sigmoidal activation function $\sigma(\cdot)$. Recall that the canonical sigmoidal function is the logistic function $\sigma(t) = (1 + \exp\{-t\})^{-1}$.

Let $\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}$ refer to weight vectors and biases of perceptrons of the form $\sigma(\mathbf{w}^T\mathbf{u} + b)$ for some $\mathbf{u} \in \mathcal{Q}$. We can express the various CovNet architectures as follows:

1. **Shallow**: For a positive semidefinite matrix $\Lambda = ((\lambda_{r,s}))$, $r = 1, ..., R$ and $s = 1, ..., S$, define

$$c_{sh}(\mathbf{u}, \mathbf{v}) := \sum_{r=1}^{R} \sum_{s=1}^{S} \lambda_{r,s} \sigma(\mathbf{w}_r^T \mathbf{u} + b_r) \sigma(\mathbf{w}_s^T \mathbf{v} + b_s), \quad \mathbf{u}, \mathbf{v} \in \mathcal{Q} \tag{3}$$

   where this can be visualized as a two-hidden-layer neural network (see Figure 1 of Sarkar and Panaretos (2022)).

2. **Deep**: Let $\mathbf{W}_1 \in \mathbb{R}^{p_1 \times d}, ..., \mathbf{W}_L \in \mathbb{R}^{p_L \times p_{L-1}}$ be weight matrices and consider bias vectors $\mathbf{b}_l \in \mathbb{R}^{p_l}, l = 1, ..., L$. We define the function $g(\cdot)$, a deep neural network, recursively through

$$\begin{aligned} \mathbf{u}_1 &= \sigma(\mathbf{W}_1 \mathbf{u} + \mathbf{b}_1) \\ \mathbf{u}_{l+1} &= \sigma(\mathbf{W}_{l+1} \mathbf{u}_l + \mathbf{b}_{l+1}) \qquad l = 1, ..., L-1 \\ g(\mathbf{u}) &:= \sigma(\mathbf{w}_{L+1}^T \mathbf{u}_L + \mathbf{b}_{L+1}) \end{aligned}$$

where at the final 'step' we return to weight/bias vectors rather than matrices. The deep CovNet is then

$$c_d(\mathbf{u},\mathbf{v}) = \sum_{r=1}^{R}\sum_{s=1}^{S} \lambda_{r,s} g_r(\mathbf{u}) g_s(\mathbf{v}), \quad \mathbf{u},\mathbf{v} \in \mathcal{Q} \tag{4}$$

where we have $R(\sum_{l=0}^{L}(p_l+1)p_{l+1}) + R(R+1)/2$ parameters for R networks – that is, R functions $g_r(\cdot)$. This can be prone to overfitting, so a weight-sharing analog is proposed in the DeepShared model below to mitigate this.

3. **DeepShared**: In the Deep CovNet, if one defines R networks, one repeats the construction R times. The 'shared' analog, naturally, shares weights in the form

$$\begin{aligned} \mathbf{u}_1 &= \sigma(\mathbf{W}_1\mathbf{u} + \mathbf{b}_1) \\ \mathbf{u}_{l+1} &= \sigma(\mathbf{W}_{l+1}\mathbf{u}_l + \mathbf{b}_{l+1}) \qquad l = 1,...,L-1 \\ g_r(\mathbf{u}) &:= \sigma(\boldsymbol{\omega}_r^T \mathbf{u}_L + \beta_r) \qquad r = 1,...,R \end{aligned}$$

which jointly defines networks $g_1,...,g_R$. Under this formulation, the DeepShared CovNet kernel c_{ds} takes the same form as in Equation 4. See Figure 2 for the authors' visualization of the neural network architecture.

Explicit Functional Form. An estimated DeepShared covariance $\widehat{c}_{ds}$ is of the form

$$\widehat{c}_{ds}(\mathbf{u},\mathbf{v}) := \sum_{r=1}^{R}\sum_{s=1}^{S} \widehat{\lambda}_{r,s} \widehat{g}_r(\mathbf{u}) \widehat{g}_r(\mathbf{v}), \quad \mathbf{u},\mathbf{v} \in \mathcal{Q}$$

from a sample of N random fields and is obtained by minimizing the Hilbert-Schmidt norm between an empirical covariance operator $\widehat{C}_N := \frac{1}{N}\sum_{n=1}^{N} \mathcal{X}_n \otimes \mathcal{X}_n$ and covariance networks $\mathcal{G}$ in the *class* of DeepShared CovNets whose form is given above.

Rather than explicitly forming this empirical covariance, one fits the observed fields themselves through neural networks with shared structures, much in the spirit of the DeepShared CovNet itself. Nonetheless, the manipulation of estimated CovNet operators $\widehat{C}$ is done by considering the eigenvalues $\lambda_{r,s}$ and eigenfunctions $\psi(\mathbf{u}) := \sum_{r=1}^{R} a_r \widehat{g}_r(\mathbf{u})$ for some $a_1,...,a_R \in \mathbb{R}$, obtained by solving an eigenvalue problem (see Section 4 of Sarkar and Panaretos (2022) for a complete treatment, as it involves a lot of mathematics). It should be noted that this eigenproblem requires Monte Carlo integration to compute integrals

$$\tilde{g}(r,s) := \int_{\mathcal{Q}} \widehat{g}_r(\mathbf{u}) \widehat{g}_s(\mathbf{u}) d\mathbf{u} \approx \frac{1}{M}\sum_{j=1}^{M} \widehat{g}_r(\mathbf{u}_j) \widehat{g}_s(\mathbf{u}_j) \tag{5}$$

Theoretical Results. CovNet estimators are consistent (Theorems 3 and 4, Remark 8, Sarkar and Panaretos (2022)) under mild conditions (e.g., bounded second moment or finite fourth moment in a modified case). Rates of convergence are presented for the Shallow CovNet – which is not the principal focus of our survey report – and can be similarly derived for the Deep and Deepshared analogs. The important takeaway is not only the consistency, but

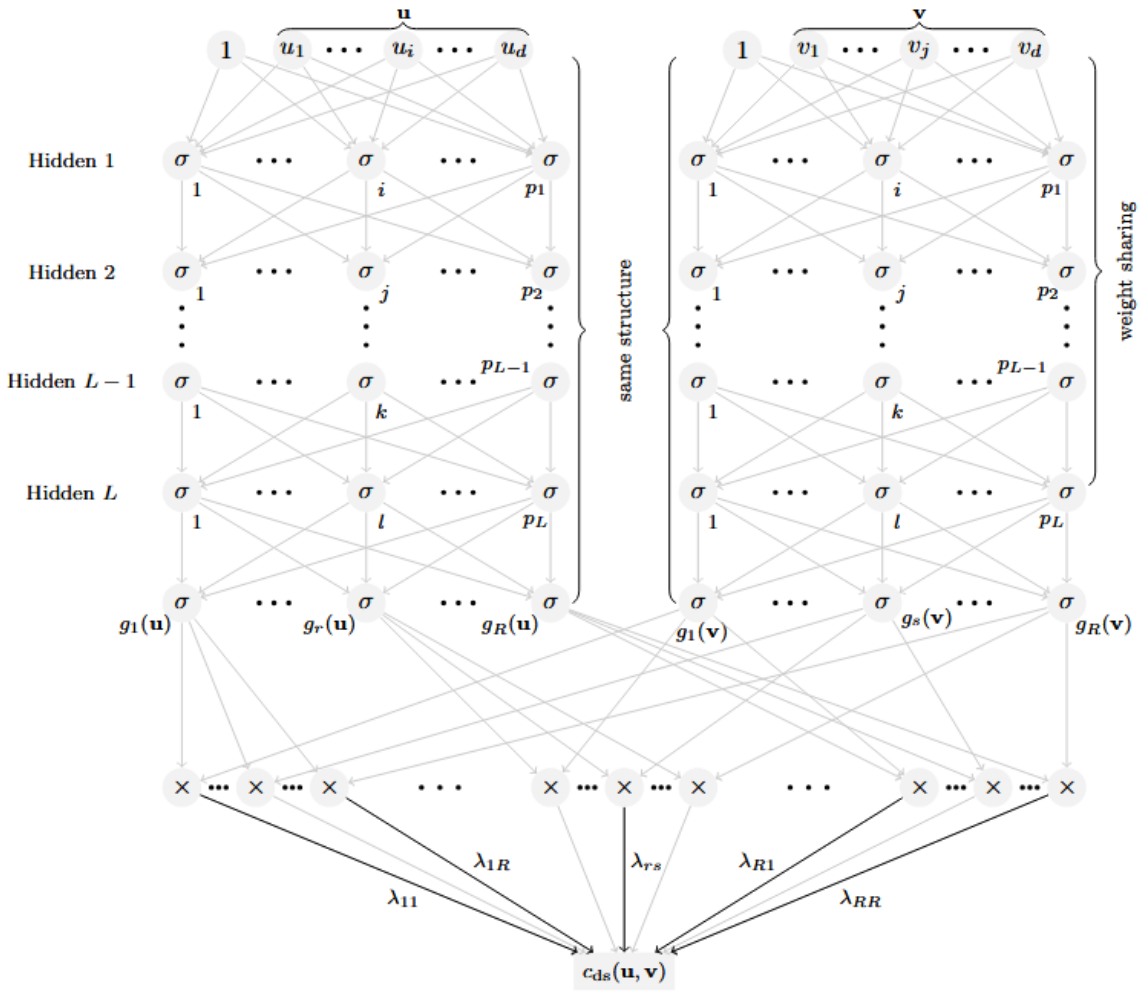

Figure 2: The DeepShared CovNet architecture as taken from Sarkar and Panaretos (2022).

that Sarkar and Panaretos (2022) establish results for when the random fields are observed with noise (Theorem 6, Sarkar and Panaretos (2022)) under certain conditions such as Lipschitz continuity of the covariance kernel. The mathematics behind the theoretical results is rather technical and lengthy; the authors give an example of the convergence rate under $\mathcal{X}$ taking values in an α-order Sobolev space wherein the Shallow CovNet converges at a rate of $\mathcal{O}((d\ln(N)/N)^{\alpha/(10d+\alpha)})$.

2.3 Image Response Regression Via Deep Neural Networks

As stated in Zhang et al. (2024), a central task in medical imaging studies is to delineate the associations between images and a set of covariates. These covariates often include but are not limited to demographic information and clinical features. We can form this as a regression problem, where the image scans (2D or 3D) are treated as the response variable and the aforementioned covariates are predictors. We refer to this as *image response regression*.

Image response regression is a difficult problem: First, different brain regions have distinct anatomical structures and biological properties, leading to complex imaging and spatial patterns. Second, the imaging dimension can be extremely high. A typical MRI image may contain hundreds of thousands of voxels (3D pixels). If we have multiple subjects, this can quickly get into the millions or multi-millions of voxels. Third, imaging data is often heterogeneous across subjects, which introduces more complex variability. Without accounting for these difficulties, image response regression may suffer from large bias and low reproducibility (Zhang et al. (2024)).

While several analyses have been proposed to address these issues, they all fall short in one way or another. The most common solution, *mass univariate analysis* (MUA), fits a generalized linear model to the associate the image measure at each voxel with the covariates (Friston (2003)). A major limitation of MUA is that the framework cannot account for spatial correlations. This leads to a low detection power. While some subsequent methods have been developed to overcome this, the challenge remains.

Another line of research to address these issues is to treat the image as a tensor, which is a multidimensional array. From there, fit a tensor response regression under numerous low-rank and sparsity structures. While these solutions *implicitly* account for the complex spatial smoothness, they do not do it explicitly. More importantly, they do not account for subject heterogeneity.

Finally, the third strategy is to employ a spatially varying coefficient model (SVCM) in a functional data analysis framework, where the response images and the regression coefficients are modeled as the realizations of spatially varying functions evaluated on discrete grid points, (i.e. voxels), in a spatial domain. Crucially, this framework incorporates spatial smoothness and other complexities (e.g., jump discontinuities) and subject heterogeneity, and has been shown to be quite effective (Zhu et al. (2014), Friston (2003)). The proposal in Zhang et al. (2024) is novel in several ways. Compared to the above approaches, Zhang et al. (2024) explicitly addresses both spatial smoothness and subject-level heterogeneity by utilizing piecewise smooth functions and incorporating individual deviation functions for the subjects. Moreover, when compared to DNN-based medical imaging analysis , the pro-

posed model in Zhang et al. (2024) is more straightforward to interpret and more robust to neural network hyperparameters. Using DNNs to approximate the main effects, individual deviations, and error variance functions in the spatially varying coefficient model framework allows the interpretations to remain the same as a classical varying coefficient model. Secondly, to improve interpretability, sparsity is imposed on the *output value* of the DNN, but *not* on the weight and bias parameters within the DNN. This bypasses the need for interpretation and selection of individual model parameters. Thirdly, when fitting the neural networks to the spatially varying functions, each spatial coordinate, not subject, is treated as a sample observation. This improves the effective sample size from the double digits of people to millions of voxels, leading to more accurate estimation via the DNNs.

Below we present the image response regression through SVCM. After, we propose to estimate the key functions in the model using DNNs and devise the multi-step estimation procedure. After introducing the model and estimation, we highlight some theoretical properties and their implications.

The model. Suppose the imaging data are collected from a $\mathcal{D}-$dimensional compact space $\mathcal{S} \subset \mathbb{R}^{\mathcal{D}}$ observed at V spatial locations, along with J covariates, from N individual subjects. For each subject $i = 1,\ldots,N$, let $y_i(\boldsymbol{s}_v) \in \mathbb{R}$ denote the image measurement at the spatial location $\boldsymbol{s}_v \in S, v = 1,\ldots,V$ and let $S_V = \{\boldsymbol{s}_v\}_{v=1}^V$ collect all those locations. Let $\boldsymbol{x}_i = (x_{i1},\ldots,x_{iJ})^\top \in \mathbb{R}^J$ denote the $J-$dimensional covariate vector. We consider the following SVCM, for $i = 1,\ldots,N,\ v = 1,\ldots,V$,

$$y_i(\boldsymbol{s}_v) = \sum_{j=1}^{J} x_{ij}\beta_j(\boldsymbol{s}_v) + \alpha_i(\boldsymbol{s}_v) + \varepsilon_i(\boldsymbol{s}_v), \qquad \text{Var}\{\varepsilon_i(\boldsymbol{s}_v)\} = \sigma^2(\boldsymbol{s}_v), \tag{6}$$

where $\beta_j(\boldsymbol{s_v}) \in \mathbb{R}$ characterizes the main effect of the jth covariate x_{ij} common for all subjects i, $\alpha_i(\boldsymbol{s_v}) \in \mathbb{R}$ characterizes the individual deviation from the common main effect that is specific to subject i, and $\varepsilon_i(\boldsymbol{s}_v)$ is measurement error with zero mean and spatially varying variance.

In model (6), we interpret $\beta_j(\boldsymbol{s}_v)$, $\alpha_i(\boldsymbol{s}_v)$, and $\varepsilon_i(\boldsymbol{s}_v)$, as the realization of the functions $\beta_j(\boldsymbol{s}_v)$, $\alpha_i(\boldsymbol{s}_v)$, and $\varepsilon_i(\boldsymbol{s}_v)$ at the set of spatial locations $\{s_v\}_{v=1}^V$. Additionally, the individual deviations $\alpha_i(s)$ play a similar role to the random effect term in Zhu et al. (2014) and Li et al. (2020). However, a key difference is that $\alpha_i(s)$ are treated as stochastic (random) whose variance-covariance structure is to be estimated, akin to a random effect in a (generalized) linear mixed model framework. Whereas, in Zhang et al. (2024), each $\alpha_i(s)$ is treated as deterministic and thus estimated directly. While we could do this if $\alpha_i(s)$ were realizations of a stochastic process (by, say, extracting random effects), estimating this covariance in high-dimensions would prove to be challenging. Moreover, since the error variance is a function of s_v, we can flexible capture the spatial heterogeneity with respect to the degree of variation in the measurement error.

Furthermore, we require the main effect functions, the individual deviations, and the error variance to be all piecewise smooth with a finite number of continuous components. Additionally, we impose sparsity on the main effect functions such that there are image regions where the main effect $\beta_j(s) = 0$, and that the absolute value of the nonzero main effect

has a positive lower bound. These assumptions are common in neuroimaging literature. We also assume there are some image regions where the individual deviations $\alpha_i(s) = 0$, and these can be different across individuals, in addition to being different from those of the main effects. Before moving to the approximation via DNNs, we note that, conditioning on the design matrix $\boldsymbol{X}$, the parameters are identifiable, as long as $\boldsymbol{X}$ has full rank and $\boldsymbol{X}^\top \boldsymbol{\alpha}(s) = 0$. This latter condition is akin to $\boldsymbol{X}^\top \boldsymbol{\epsilon} = 0$ in classical linear models (that is, the covariates and noise are uncorrelated).

Approximation via DNNs. We consider an L-layer feed-forward neural network with either the sigmoid function or rectified linear unit (ReLU) activation function. We note that the results are relatively robust to the choice of activation function. Importantly, to impose sparsity, we add a hard thresholding layer, by employing an elementwise hard thresholding function $\mathcal{T}_\eta^{(B)}$ with threshold η and truncation level B:

$$\mathcal{T}_\eta^{(B)}(u)_j = \text{sign}(u_j) \cdot \mathbb{1}(|u_j| \geq \eta_j) \cdot \min(|u_j|,\ B), \tag{7}$$

where $B \in \mathbb{R}^+$ bounds the magnitude of the truncated output. This is one of the regularity conditions in the theoretical investigation.

We then approximate $\boldsymbol{\beta}(\boldsymbol{s})$, $\boldsymbol{\alpha}(\boldsymbol{s})$, and $\log\{\sigma^2(s)\}$ as

$$\boldsymbol{\beta}(\boldsymbol{s}) = \mathcal{T}_{\eta_\beta}^{(B)} \circ N(\boldsymbol{s}|\boldsymbol{\theta}_\beta), \quad \boldsymbol{\alpha}(\boldsymbol{s}) = \mathcal{T}_{\eta_\alpha}^{(B)} \circ N(\boldsymbol{s}|\boldsymbol{\theta}_\alpha), \quad \log\{\sigma^2(\boldsymbol{s})\} = \mathcal{T}_0^{(B)} \circ N(\boldsymbol{s}|\boldsymbol{\theta}_\sigma), \tag{8}$$

where $\circ$ denotes the elementwise function composition operation. The term $N(\boldsymbol{s}|\boldsymbol{\theta})$ is the neural network output before thresholding (specifically, a concatenation of N neural networks whose input is the $D-$dimensional spatial coordinate). In short, the model in (8) is akin to classical regression with fixed design predictors. A graphical illustration of the procedure is in Figure 3.

Some main takeaways of using (8) to estimate the unknown functions in (6) are as follows. Notably, the interpretation of the resulting model is straightforward. We do not need to interpret the individual neural network parameters, but the entire fitted neural networks, where the interpretation remains the same as in the classical varying coefficient model. Moreover, we do not impose the sparsity on the individual parameters either, but rather apply hard thresholding to the output of the entire neural networks to achieve sparsity and to further facilitate model interpretation. Lastly, the input of each neural network in our model is the spatial coordinate of a single voxel, and the output of the neural network is a model coefficient, both of which are low-dimensional. Therefore, a feed-forward neural network suffices for our setting, and it does not require more sophisticated DNN architectures. Numerical experiments suggest that the model fitting is relatively robust to the neural network architecture, as well as to the network hyperparameters, such as the number of hidden layers and hidden nodes. This approach reflects a major difference between the usage of DNNs in our varying coefficient model and the classical usage of DNNs in imaging analysis.

The estimation algorithm for (6) is a three-step process. First, the neural network equation is fit and $\hat{\boldsymbol{\theta}}_\beta$ is obtained via minibatch stochastic gradient descent (SGD). Next, the individual

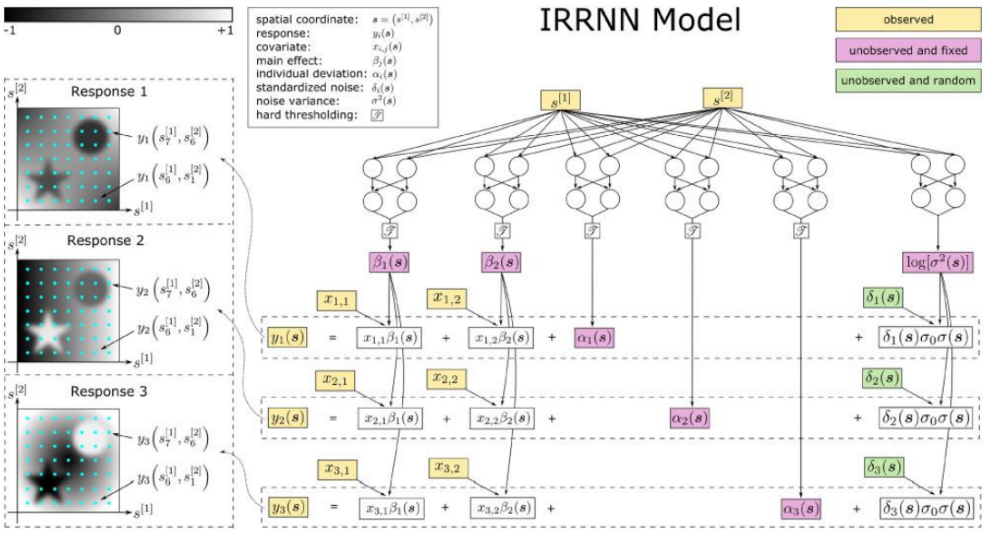

Figure 3: An illustration of the proposed model with $J = 2$ covariates and $N = 3$ images of dimension $D = 2$ observed on a grid of $V = 7 \times 7$ voxels.

deviation function is estimated with the neural network and using the resulting estimator $\hat{\boldsymbol{\beta}}(\boldsymbol{s}) = \mathcal{T}_{\eta_\beta}^{(B)} \circ N(\boldsymbol{s}_v | \hat{\boldsymbol{\theta}}_\beta)$ from step 1, $\hat{\boldsymbol{\theta}}_\alpha$ is obtained. Minibatch SGD is employed and the resulting estimator is obtained similarly. For the error variance function, the mean squared residual estimator based upon $\hat{\boldsymbol{\beta}}(\boldsymbol{s})$ and $\hat{\boldsymbol{\alpha}}(\boldsymbol{s})$ is first obtained, then the neural network is fit and the log-scale error variance is obtained. Again, optimization is carried out via minibatch SGD and the estimator is obtained. The steps are not done interatively, only once, as numerical experiments have shown the estimate does not improve much after the first iteration.

Theoretical Results. The two main theoretical results are summarized as follows. First, the L_2 error bound of the sparse main effect estimator is established, proving it is a consistent estimator. Second, the error bounds for $\hat{\boldsymbol{\alpha}}(\boldsymbol{s})$ and $\hat{\boldsymbol{\sigma}}(\boldsymbol{s})$ are established. We now present simulation results.

3 Simulation Studies

Code for the simulation studies and real data analysis can be found at this link: here.

3.1 CovNet

Setup. Because CovNet (Sarkar and Panaretos (2022)) discusses multiple desirable properties or contributions to the field of deep learning, we target smaller simulation studies to emulate and illustrate the authors' conclusions. They are listed below:

1. **Universal Approximation (Scenario A)**: Fix the dimension $d = 2$, grid size per dimension $K = 25$, and sample size $N = 500$, as well as the number of Monte Carlo replicates used in Equation 5. For different covariance structures (Brownian sheet, rotated Brow-

nian sheet, Integration Brownian sheet, rotated Integrated Brownian sheet, Matern – see Section 5 of Sarkar and Panaretos (2022) for formulae), measure approximation error via relative L^2-error of the estimated covariance.

In Figure 4, we see this property illustrated in that no covariance structure is approximated poorly, relative to others. Of course, there are other covariance kernels beyond the five originally presented in Sarkar and Panaretos (2022) that we use here, but the breadth of kernels here across the presence/absence of stationarity and/or other often desired properties showcases the power of the method.

2. **Convergence (Scenario B)**: Fix d, K and consider a moderately-sized DeepShared network. Vary N and compute the same L^2-error metric as in Scenario A.

   In Figure 5, we see the convergence on display, where the increasing sample size *tightens* the dispersion of relative L^2-errors across replications of the same simulation setup.

3. **Dimensional scaling (Scenario C)**: This involves varying d, K. Due to the limitations of computational resources on a personal computer, we only consider $d = 2, 3$ and $K = 5, 10$.

   In Figure 6, we see the capability of dimensional scaling of CovNet as the DeepShared CovNet again does not exhibit worsening performance. Some kernels (e.g., integrated Brownian motion) see slightly increased error metrics as the grid size increases for a fixed dimension, while others such as Matern are relatively robust in error with respect to dimension.

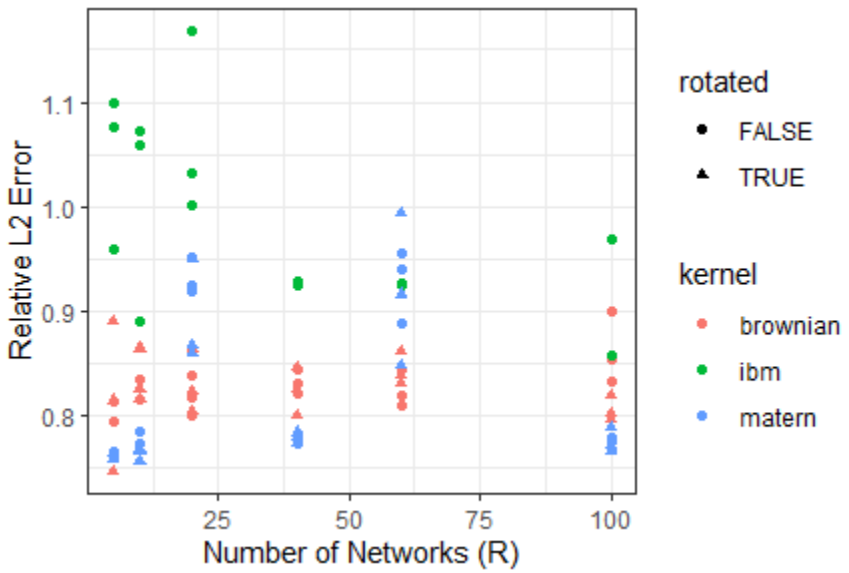

Figure 4: Visualization of Scenario A simulation study showing relative L^2-error to approximation of true covariance kernels as a function of the number of networks.

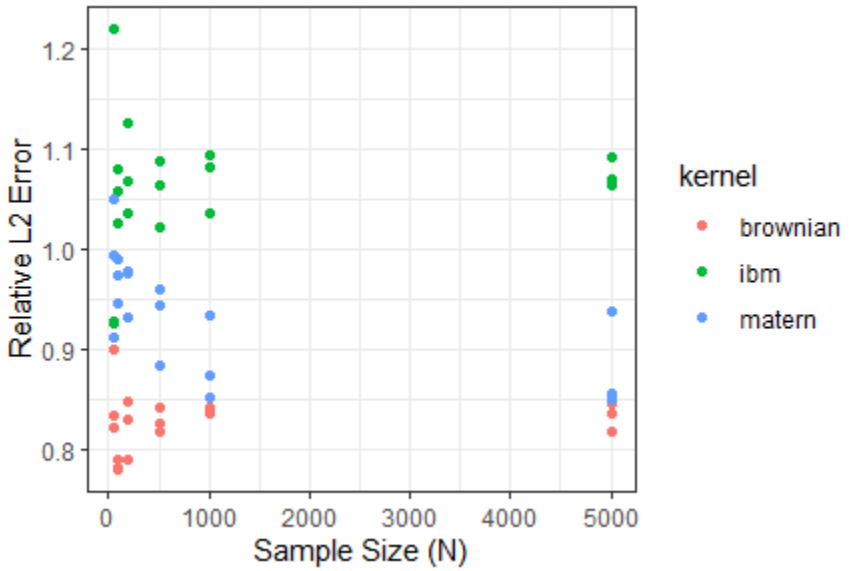

Figure 5: Visualization of Scenario B simulation study showing convergence of covariance estimators as a function of sample size.

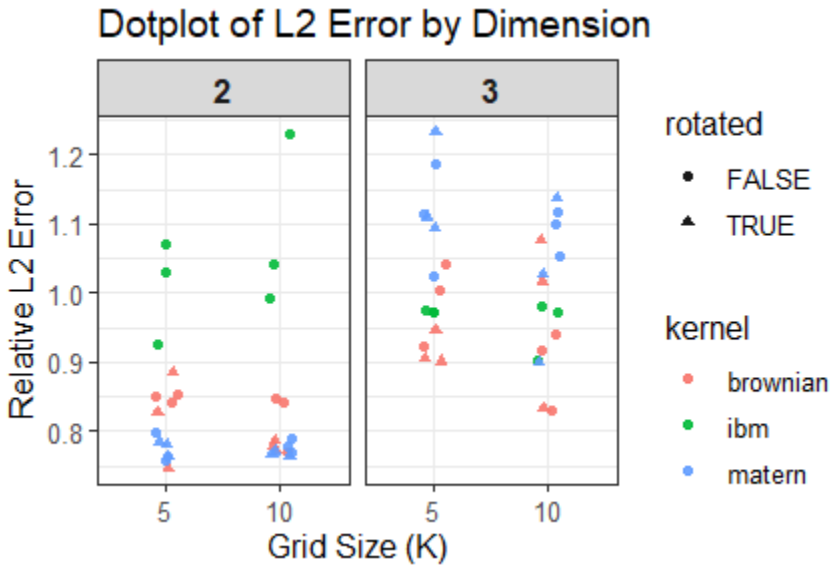

Figure 6: Visualization of Scenario C simulation showing relative L^2-error when varying dimension and grid size.

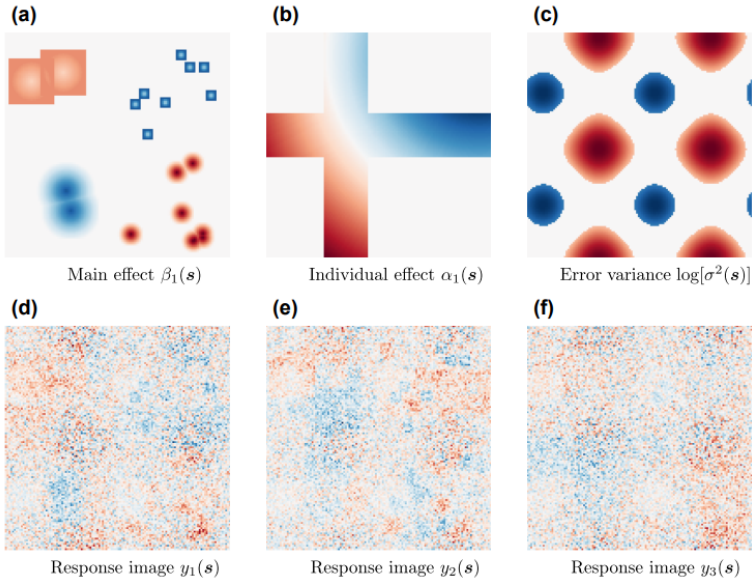

Figure 7: Examples of main effect function, individual deviation function, log-scale error variance function, and response images.

3.2 Image Response Regression

Setup. We generate data from (6) with $J = 3$ covariates drawn independently from $N(0,1)$ and number of subjects $N = 20$ and 50. We consider two image dimensions $64 \times 64 \times 64$ and $128 \times 128 \times 128$ resulting in $V = 262,144$ and $V = 2,097,152$, which are sufficiently large to train a neural network. The model parameters are generated from a 3D cube and are spatially heterogeneous with many zero-valued regions and follow patterns illustrated in Figure 7a-c.

For the estimation accuracy, IRRNN achieves the smallest MSE almost uniformly (not pictured). Such an advantage of IRRNN is more prominent in more challenging settings, when the number of images is small, or when the signal-to-noise ratio is low. For the selection accuracy, IRRNN achieves the false positive rate (FPR) below 0.03 in all settings while other methods achieve the FPR no more than 0.04. Among all methods with the FPR below 0.05, IRRNN achieves the highest TPR uniformly (not pictured). Such an advantage in power is more prominent in the settings when the sample size is limited, the noise is high, and the imaging dimension is high. Overall, the simulations show that the proposed method outperforms the existing solutions in both estimation accuracy and selection accuracy.

To visualize the estimation results, Figure 8a–f shows the estimates of the main effect function in Figure 7a by different methods for one data replication with $STN = 0.10$, $N = 20$, $V = 128 \times 128 \times 128$ and normal noise. Clearly, the proposed IRRNN successfully detects all the activation regions and clearly identifies the correct shapes and boundaries.

Figure 8: Estimate of the main effect function by various methods

4 Real Data Analysis

For the real data analysis, we will focus on comparing the FDNN method to two other classification methods: using a convolutional neural network (CNN) for classification and using multinomial GAM modeling to classify. Our data comes from the Fashion-MNIST dataset (Figure 1), specifically, we will be utilizing a subset of the dataset being the data that is the first three types: 0 = Tshirt/top, 1 = Trouser, and 2 = Pullover. Each method outputs a probability matrix, and we get the classifier by taking the probability that is the maximum for that row. In terms of results, we are looking at the misclassification rate for each method, i.e., the proportion of images in the test dataset that are not classified as the true label. In our dataset, there are 18000 images in the training set and 3000 images in the test set. Each image is a 28x28 grid of pixels.

Method	Misclassification Rate
Multinomial Functional Regression	25.1%
CNN-based classifier	2.2%
Functional DNN	2.0%

Table 1: This table shows the results (the misclassification rate) of each of the three methods used in the real data analysis.

The table above shows the results of the real data analysis. It shows that the neural networks do significantly better than the regression-based classifier!

5 Conclusion

Deep learning methods for functional data exist within many different statistical settings. Our report catalogues their recent presence within the areas of **classification** through FDNNs, **representation** through CovNet, and **regression** through Image Response Regression and the SVCM framework. FDNNs allow for classification beyond traditional assumptions of Gaussian functional data and are computationally tractable in high dimensions. CovNet sidesteps popular assumptions of separability to present a more general, thorough, and tractable method utilizing neural network architectures to efficiently explain spatial/spatiotemporal co-variations in multidimensional data (e.g., fMRI data). The SVCM framework when applied to image responses allows for regression models to efficiently estimate responses while retaining interpretability, a common feature lacking in canonical deep learning or neural-network-based methods. The report performs simulation studies to emulate the authors' findings and extends this to a real data analysis under the fashion-MNIST dataset.

References

Friston, K. J. (2003). Statistical parametric mapping. *Neuroscience databases: a practical guide*.

Zhu, H., Fan, J., & Kong, L. (2014). Spatially varying coefficient model for neuroimaging data with jump discontinuities. *Journal of the American Statistical Association, 109*(507), 1084–1098. https://doi.org/10.1080/01621459.2014.881742

Hsing, T., & Eubank, R. (2015). *Theoretical foundations of functional data analysis, with an introduction to linear operators*. Wiley. https://doi.org/10.1002/9781118762547

Aston, J. A. D., Pigoli, D., & Tavakoli, S. (2017). Tests for separability in nonparametric covariance operators of random surfaces. *The Annals of Statistics, 45*(4), 1431–1461. https://doi.org/10.1214/16-AOS1502

Li, X., Wang, L., & Wang, H. J. (2020). Sparse learning and structure identification for ultrahigh-dimensional image-on-scalar regression. *Journal of the American Statistical Association, 116*(536), 1994–2008. https://doi.org/10.1080/01621459.2020.1753523

Sarkar, S., & Panaretos, V. M. (2022). CovNet: Covariance Networks for Functional Data on Multidimensional Domains. *Journal of the Royal Statistical Society Series B: Statistical Methodology, 84*(5), 1785–1820. https://doi.org/10.1111/rssb.12551

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

Wang, S., Zhang, W., Cao, G., & Huang, Y. (2024). Functional data analysis using deep neural networks. *WIREs Computational Statistics, 16*(4). https://doi.org/10.1002/wics.70001

Zhang, D., Li, L., Sripada, C., & Kang, J. (2024). Image response regression via deep neural networks. *Journal of the Royal Statistical Society Series B: Statistical Methodology, 85*(5), 1589–1614. https://doi.org/10.1093/jrsssb/qkad073