Journal of Machine Learning Research 26 (2025) 1-49 Submitted 5/23; Revised 10/25; Published 12/25

Nonlinear functional regression by functional deep neural network with kernel embedding

Zhongjie Shi ZSHI332@GATECH.EDU
School of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332, USA

Jun Fan JUNFAN@HKBU.EDU.HK
Department of Mathematics
Hong Kong Baptist University
Kowloon, Hong Kong

Linhao Song SONGINCSU@CSU.EDU.CN
School of Mathematics and Statistics
Central South University
Hunan, 410083, China

Ding-Xuan Zhou DINGXUAN.ZHOU@SYDNEY.EDU.AU
School of Mathematics and Statistics
University of Sydney
Sydney, NSW 2006, Australia

Johan A.K. Suykens JOHAN.SUYKENS@ESAT.KULEUVEN.BE
Department of Electrical Engineering
ESAT-STADIUS, KU Leuven
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

Editor: John Shawe-Taylor

Abstract

Recently, deep learning has been widely applied in functional data analysis (FDA) with notable empirical success. However, the infinite dimensionality of functional data necessitates an effective dimension reduction approach for functional learning tasks, particularly in nonlinear functional regression. In this paper, we introduce a functional deep neural network with an adaptive and discretization-invariant dimension reduction method. Our functional network architecture consists of three parts: first, a kernel embedding step that features an integral transformation with an adaptive smooth kernel; next, a projection step that uses eigenfunction bases based on a projection Mercer kernel for the dimension reduction; and finally, a deep ReLU neural network is employed for the prediction. Explicit rates of approximating nonlinear smooth functionals across various input function spaces by our proposed functional network are derived. Additionally, we conduct a generalization analysis for the empirical risk minimization (ERM) algorithm applied to our functional net, by employing a novel two-stage oracle inequality and the established functional approximation results. Ultimately, we conduct numerical experiments on both simulated and real datasets to demonstrate the effectiveness and benefits of our functional net.

Keywords: Deep learning theory, functional deep neural network, kernel smoothing, nonlinear functional, approximation rates, learning rates

1. Introduction

Functional data analysis (FDA) is a rising subject in recent scientific studies and daily life, which analyzes data with information about curves, surfaces, or anything else over a continuum, such as time, spatial location, and wavelength which are commonly considered in physical background. Readers seeking a thorough introduction to functional data analysis may consult Ramsay and Silverman (2005); Ferraty and Vieu (2006); Hsing and Eubank (2015), which cover key methodologies and theoretical foundations.

One significant task in FDA is functional regression, which aims to learn the relationship between a functional covariate and a scalar (or functional) response. Among the various regression models, the functional linear model and its variants (Yao et al., 2005a; Müller and Stadtmüller, 2005), such as the generalized linear model, single-index model, and multiple-index model, are widely used and studied. The influential study of Hall and Horowitz (2007) derived the optimal convergence rate for functional linear regression when the slope function was estimated using plug-in eigenpairs. In subsequent developments, Dou et al. (2012) established the methodology and theory for the functional generalized linear model by employing a truncated likelihood approach to handle the difficulties posed by the nonlinear link function. Chen et al. (2011) developed methods for the single-index and multiple-index models and derived polynomial convergence rates for prediction. In parallel, functional linear regression was also studied under the reproducing kernel Hilbert space (RKHS) framework (Yuan and Cai, 2010; Cai and Yuan, 2012). Notably, all the aforementioned studies assumed fully observed functional data, which represented an idealized scenario. However, in practice, observations are typically available only at discrete time points. We model such discretely observed data through a two-stage process. The first-stage dataset $\{f_i(\cdot), y_i\}_{i=1}^m$ are i.i.d. sampled from some true unknown probability distribution, where f_i are random functions and y_i are corresponding responses. We cannot observe f_i over the entire continuum. Instead, we observe f_i at discrete grid points $\{t_{i,j}\}_{j=1}^{n_i}$. Thus, the second-stage data $\{\{f_i(t_{i,j})\}_{j=1}^{n_i}, y_i\}_{i=1}^m$ is what we typically work with in practice. Theoretical results for functional linear regression with discretely observed covariates were analyzed in Zhou et al. (2022), which established optimal convergence rates for both slope estimation and prediction. Functional principal component analysis for discretely observed data was studied in Amini and Wainwright (2012), under the assumption that the covariates lay in an RKHS, and in Zhou et al. (2025), which conducted perturbation analyses of the covariance operator with a diverging number of eigencomponents.

While functional linear model and its variants are simple and interpretable, they can struggle in scenarios where the true relationship is nonlinear. This has motivated the exploration of nonparametric regression methods. Kadri et al. (2016) extended kernel methods to infinite-dimensional settings and studied the least-squares regularization using functional-valued RKHSs. Subsequently, Wang and Xu (2019) investigated the regularized learning schemes for non-point-evaluation functional data. Beyond kernel-based approaches, deep learning methods have also been explored for functional nonlinear regression, motivated by their empirical success in automatic feature extraction and flexible architecture design across diverse data formats. This is particularly advantageous for high-dimensional data with an underlying low-dimensional structure, as DNNs can efficiently capture the intrinsic features. In contrast, kernel methods rely on manual selection of a kernel function to exploit such

structures, and the optimal choice is often unknown. Consequently, in practical settings, DNNs might perform better than kernel methods when handling high-dimensional or even infinite-dimensional data. Since functional data are intrinsically infinite-dimensional, one key idea in applying deep learning is to first summarize the information contained in each function f_i into a finite-dimensional vector based on the discrete observations $\{f_i(t_{i,j})\}_{j=1}^{n_i}$, and then feed the resulting multivariate data into a deep neural network.

Recently, much effort has been dedicated to this idea in the literature. Chen and Chen (1995) proposed a functional network structure that directly uses the discrete observations $\{f_i(t_{i,j})\}_{j=1}^{n}$ at grid points $t_{i,j} = t_j$, treating these observations as multivariate input data for a shallow network. This straightforward network architecture was demonstrated to be universal in Chen and Chen (1995) and was subsequently extended to a deeper version by Lu et al. (2021). However, this structure is mesh-dependent, meaning that it requires a fixed sampling grid $\{t_j\}_{j=1}^{n}$. Consequently, if this grid is altered, it would be computationally costly to retrain the model. A straightforward approach to designing mesh-independent (or discretization invariant) models is to use basis representations. For any input function f, the truncated basis expansion $\sum_{k=1}^{d_1}\langle f, \phi_k\rangle\phi_k$ serves as an approximation of f by projecting it onto the subspace spanned by an orthonormal basis $\{\phi_k\}_{k=1}^{d_1}$. The coefficient vector $[\langle f, \phi_1\rangle, \cdots, \langle f, \phi_{d_1}\rangle)]^T$ can then be fed into various deep neural networks, such as Multi-Layer Perceptron (MLP) and Radial-Basis Function Networks (RBFN) (Rossi et al., 2005). The universality and consistency of these models have been established in Rossi and Conan-Guez (2005). The choice of basis for dimension reduction can vary. One approach is to use preselected bases that do not require learning, such as Legendre polynomials (Mhaskar and Hahm, 1997), B-splines (Rice and Wu, 2001; Cardot et al., 2003), and Fourier basis (Kovachki et al., 2021). However, these preselected bases do not fully leverage the information contained in the data. As a result, we often need a high-dimensional vector to reconstruct the original function, which can lead to the curse of dimensionality in the subsequent deep neural network. Therefore, it might be better to consider a data-dependent basis, such as the principal component basis (Besse and Ramsay, 1986; Silverman, 1996; Yao et al., 2005b) or neural network basis (Rossi et al., 2002; Yao et al., 2021). However, these approaches come with their own drawbacks. While the principal component method effectively captures information from the input data, it overlooks the information derived from the output data. In the case of neural network basis, there are numerous free parameters within the basis layer that must be trained alongside the parameters of the subsequent network layers. This substantially increases the model's capacity, requiring a larger dataset to prevent the risk of overfitting.

Therefore, this paper aims to design a discretization-invariant dimension reduction method that adapts to both input and output data without increasing the complexity of the network structure. The key technique in our dimension reduction method is called kernel embedding, which is inspired by the kernel mean embedding approach in Smola et al. (2007), where a distribution is mapped to an element in an RKHS through an embedding map. Our dimension reduction process can be summarized in two main steps. First, we specify an embedding kernel $K : \Omega \times \Omega \to \mathbb{R}$ and apply the integral transformation $L_K f_i(x) = \int_\Omega K(x,t)f_i(t)dt$ induced by K as the embedding map. The input functional data f_i will then be transformed to $L_K f_i$. Since only the second-stage data $\{f_i(t_{i,j})\}_{j=1}^{n_i}$ are accessible, this embedding can be approximated using an empirical alternative $\widehat{L_K}$, which

serves as an estimate of L_K. Formal definition of $\widehat{L_K}$ will be provided in the subsequent section. After the embedding step, we then map $\widehat{L_K} f$ to a d_1 dimensional coefficient vector, through a projection onto the subspace spanned by the first d_1 eigenfunctions of the integral operator induced by a projection Mercer kernel K_0. This coefficient vector is then fed into a deep neural network to predict the output. Since the embedding kernel K is equipped with hyperparameters that are optimized through cross-validation, this approach offers adaptivity in the dimension reduction process without increasing the model's capacity.

In this paper, we propose a functional network structure that integrates the previously discussed dimension reduction method. Theoretical analysis and numerical experiments are conducted to evaluate this structure. In summary, our contributions are as follows:

1. We evaluate the expressivity of our proposed functional network by deriving explicit rates of approximating nonlinear smooth functionals defined on various input function spaces. For input functions within Besov spaces or mixed smooth Sobolev spaces, we establish logarithmic rates of approximation. For input functions within Gaussian RKHSs, we enhance the approximation rates to $\exp(-\alpha(\log M)^\beta)$, where $\alpha > 0$ and $0 < \beta < 1$. This improvement in the rates indicates that our functional network can exploit the regularity within the input functions.

2. We propose a learning algorithm through empirical risk minimization (ERM) applied to our functional network based on the second-stage data. Generalization analysis is carried out on this learning algorithm within the classical learning theory framework. Specifically, we establish a new two-stage oracle inequality that considers both the first-stage sample size m and the second-stage sample size n. By applying this new oracle inequality along with a theoretically optimal quadrature scheme, we derive convergence rates for learning target functionals defined on various input function spaces. Our theoretical findings reveal that the second-stage sample size required for achieving unimpaired generalization error can be significantly smaller than that of the first-stage sample size, indicating the potential of our functional network in handling sparsely observed functional data.

3. Numerical experiments conducted on both simulated and real datasets indicate that our functional network utilizing kernel embedding surpasses the performance of functional networks that rely on other baseline dimension reduction techniques, including discrete observations, B-splines, FPCA, and neural network bases. The numerical results also provide valuable insights into the key factors affecting the generalization performance of our functional network. Furthermore, we validate the advantages of our approach by showcasing its discretization-invariant properties, its adaptability to a range of datasets, and its robustness to noisy observations.

The rest of the paper is arranged as follows. In Section 2, we introduce the architecture of our functional deep neural network with kernel embedding and elaborate on its practical implementation. In Section 3, we conduct the theoretical analysis of our functional network by presenting its approximation rates for various input spaces when the target functional is smooth. In Section 4, we carry out a generalization analysis by first providing a two-stage oracle inequality, and then utilizing it to derive learning rates for the input function spaces considered in Section 3. Section 5 discusses related work and summarizes the theoretical

findings of this paper. In Section 6, numerical experiments are performed to verify the performance and benefits of our proposed functional network. The proofs of the main results in this paper are given in the Appendix.

2. Architecture of Functional Deep Neural Network with Kernel Embedding

In this section, we give a detailed introduction to the architecture of our functional deep neural network with the usage of kernel embedding. Suppose that the input function space $\mathcal{F}$ is a compact subset of $L_\infty(\Omega)$ and of $L_2(\Omega)$, where Ω is a measurable subset of $\mathbb{R}^d$. We denote $L_p(\Omega)$ as the Lebsegue space of order p with respect to the Lebsegue measure on Ω, and denote its norm as $\|\cdot\|_{L_p(\Omega)}$. Furthermore, we denote $L_p(\mu)$ as the Lebesgue space of order p with respect to the measure μ on Ω, and denote its norm as $\|\cdot\|_{L_p(\mu)}$.

Let μ be a positive Borel measure on Ω and $L_2(\mu)$ be the Hilbert space of the square integrable functions on Ω w.r.t. μ. For a *kernel* $K : \Omega \times \Omega \to \mathbb{R}$, we denote L_K^μ as

$$\left(L_K^\mu f\right)(x) = \int_\Omega K(x,t) f(t) d\mu(t). \tag{2.1}$$

Moreover, if Ω is a compact subset of $\mathbb{R}^d$ and the kernel K qualifies as a *Mercer kernel* (or *reproducing kernel*), meaning that it is continuous, symmetric, and positive definite. In this case, the linear *integral operator* $L_K^\mu : L_2(\mu) \to C(\Omega)$ is a compact, self-adjoint, positive operator. Consequently, the Spectral Theorem is applicable, indicating that there exists an orthonormal basis $\{\phi_1, \phi_2, \cdots\}$ of $L_2(\mu)$ consisting of the eigenfunctions of L_K^μ. If λ_k denotes the eigenvalue associated with ϕ_k, the set $\{\lambda_k\}$ is either finite or approaches zero when $k \to \infty$. Moreover, Mercer's Theorem states that uniformly,

$$K(x,t) = \sum_{k=1}^{\infty} \lambda_k \phi_k(x) \phi_k(t). \tag{2.2}$$

Furthermore, if we denote

$$C_K = \sup_{x,t \in \Omega} |K(x,t)|, \tag{2.3}$$

then the operator norm is bounded as $\|L_K^\mu\| \le \sqrt{\mu(\Omega)} C_K$. We note that in this paper, unless specified otherwise, the term "kernel" refers specifically to a Mercer kernel.

Definition 1 (Functional net with kernel embedding) *Let $f \in \mathcal{F}$ represent an input function. We start by performing a kernel embedding step on f through an integral transformation, given by:*

$$L_K f = \int_\Omega K(\cdot, t) f(t) dt, \tag{2.4}$$

where the embedding kernel $K : \Omega \times \Omega \to \mathbb{R}$ is a continuous kernel.

Next, we define the functional deep neural network with a depth of J and width $\{d_j\}_{j=1}^J$. This network is structured iteratively as follows:

$$h^{(j)}(f) = \begin{cases} T(L_K f), & j = 1, \\ \sigma\left(F^{(j)} h^{(j-1)}(f) + b^{(j)}\right), & j = 2, 3, \ldots, J. \end{cases} \tag{2.5}$$

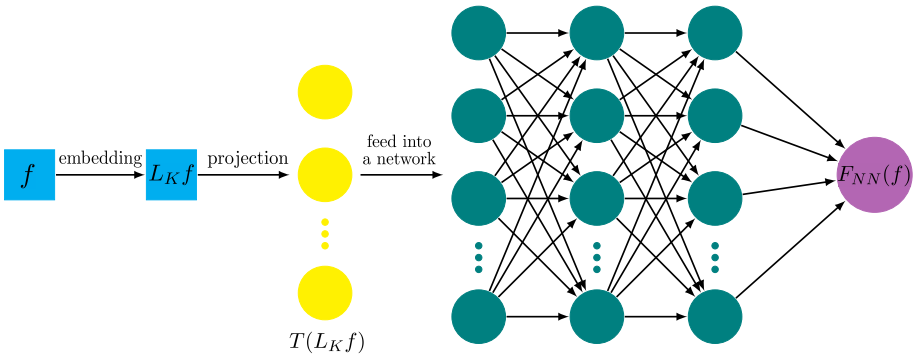


Figure 1: Architecture of functional net with kernel embedding.

In this formulation, the first layer is identified as a projection step using the basis $\{\phi_i\}_{i=1}^{d_1}$*:*

$$T(L_K f) = \left[\int_\Omega L_K f(t)\phi_1(t)d\mu(t), \int_\Omega L_K f(t)\phi_2(t)d\mu(t), \cdots, \int_\Omega L_K f(t)\phi_{d_1}(t)d\mu(t)\right]^T, \tag{2.6}$$

where $\{\phi_i, \lambda_i\}$ *represents the eigensystem of the integral operator* $L_{K_0}^\mu$ *induced by a projection Mercer kernel* K_0 *and a positive Borel measure* μ*. The subsequent layers follow the standard deep neural network, where* $\{F^{(j)} \in \mathbb{R}^{d_j \times d_{j-1}}\}_{j=1}^J$ *are the weight matrices,* $\{b^{(j)} \in \mathbb{R}^{d_j}\}_{j=1}^J$ *are the bias vectors, and* $\sigma(u) = \max\{u, 0\}$ *is the ReLU activation function.*

The final output of the functional net is derived as a linear combination of the last layer

$$F_{NN}(f) = c \cdot h^{(J)}(f), \tag{2.7}$$

where $c \in \mathbb{R}^{d_J}$ *is the coefficient vector. Figure 1 specifically depicts the architecture of the functional net with kernel embedding.*

Below, we provide two examples regarding the selection of embedding kernels and projection kernels in our functional network. In the first example, we choose the embedding kernel K to be the product of a Gaussian kernel with a density function. In the second example, the embedding kernel K is selected as a linear combination of Gaussian kernels. Let $\mathcal{H}_\gamma$ denote the RKHS corresponding to the Gaussian kernel k_γ:

$$k_\gamma(x,t) = \exp\left(-\frac{\|x-t\|_2^2}{\gamma^2}\right), \qquad x, t \in \mathbb{R}^d, \tag{2.8}$$

where $\gamma > 0$ represents the bandwidth, and we denote its norm as $\|\cdot\|_{H_\gamma}$.

Example 1 *If we choose the embedding kernel* K *as* $K(x,t) = k_\gamma(x,t)u(t)$ *where* u *is the density of a positive Borel measure* μ*, notice that* $L_K f \in \mathcal{H}_\gamma(\Omega)$ *for any* $f \in L_2(\Omega)$*, we can set the projection kernel* K_0 *to be* k_γ*.*

Example 2 *If we choose the embedding kernel* K *to be a linear combination of a collection of Gaussian kernels with varying bandwidths* γ_i*:*

$$K(x,t) = \sum_{i=1}^P \beta_i k_{\gamma_i}(x,t). \tag{2.9}$$

Denote $\gamma = \min_i \gamma_i$. *Notice that* $L_K f \in \mathcal{H}_\gamma(\Omega)$ *for any* $f \in L_2(\Omega)$, *we can set the projection kernel* K_0 *to be* k_γ.

The architecture described in Definition 1 serves as a theoretical framework. Next, we will outline how this framework can be practically implemented. The two critical steps for empirical implementation are the kernel embedding step specified in (2.4) and the projection step described in (2.6).

For the kernel embedding step defined in (2.4), we can assume, without loss of generality, that the domain of the input functions is $\Omega = [0,1]^d$. This allows us to express the embedding step as follows:

$$L_K f_i = \int_\Omega K(\cdot, t) f_i(t) d\mathcal{U}(t), \tag{2.10}$$

where $d\mathcal{U}$ represents the uniform distribution on Ω. The quadrature problem focuses on approximating this integral using linear combinations:

$$\widehat{L_K} f_i = \sum_{j=1}^{n_i} \theta_{i,j} K(\cdot, t_{i,j}), \tag{2.11}$$

where the points $t_{i,j} \in \Omega$ and the weights $\theta_{i,j}$ are chosen appropriately to minimize the approximation error as much as possible.

Remark 2 *The standard Monte-Carlo method is to consider observation points* $\{t_{i,j}\}_{j=1}^{n_i}$ *i.i.d. sampled from* $d\mathcal{U}$ *and the weights* $\theta_{i,j} = f_i(t_{i,j})/n_i$, *which results in a decrease of the error in the rate of* $1/\sqrt{n_i}$. *Alternative sampling methods, such as Quasi Monte-Carlo methods and Trapezoidal rules, can lead to improved convergence rates.*

For the projection step defined in (2.6), we apply numerical integration over the discrete grid points $\{s_k\}_{k=1}^N$ within Ω, using weights $\{w_k\}_{k=1}^N$. This results in the following equation:

$$\widehat{T}(\widehat{L_K} f_i)_\ell = \sum_{k=1}^{N} w_k \sum_{j=1}^{n_i} \theta_{i,j} K(s_k, t_{i,j}) \phi_\ell(s_k), \quad \ell = 1, \ldots, d_1. \tag{2.12}$$

When the grid points are chosen to be sufficiently dense with a large value of N, this numerical integration can yield an accurate approximation of the integral.

Finally, we provide Algorithm 1 as a comprehensive description of how to train our functional network in practice. This algorithm is precisely what we used for the training in our numerical simulations. In this algorithm, we select the embedding kernel K to be the Gaussian kernel k_γ with bandwidth $\gamma > 0$. The basis functions $\{\phi_\ell\}_{\ell=1}^{d_1}$ used in the projection step are chosen as the eigenfunctions of the integral operator L_K^μ, which are explicitly represented in terms of Hermite polynomials, as discussed in (Fasshauer, 2011, pp. 28). Additionally, the measure $d\mu$ is selected to be the Gaussian distribution, characterized by the density function $u(x) = \frac{\beta}{\sqrt{\pi}} \exp^{-\beta^2 x^2}$, where β serves as a global scaling parameter.

Algorithm 1 Training functional network

Input: second-stage data $\widehat{D} = \{\{t_{i,j}, f_i(t_{i,j})\}_{j=1}^{n_i}, y_i\}_{i=1}^m$, a 3D grid of hyperparameters: Gaussian kernel bandwidth $\gamma > 0$, scaling parameter $\beta > 0$ of the Gaussian distribution μ, reduced dimension d_1.

Output: the learned functional net model $F_{\widehat{D}}$, and the corresponding test-set MSE.

Step 1: For the particular hyperparameters γ, β, d_1, scale the discretization points $\{t_{i,j}\}_{j=1}^{n_i}$ to the domain $\Omega = [0,1]^d$, convert the input data $\{t_{i,j}, f_i(t_{i,j})\}_{j=1}^{n_i}$ from $\widehat{D}$ into a d_1 dimensional vector $x_i \in \mathbb{R}^{d_1}$ for $i = 1, \ldots, m$, using the following calculation:

$$(x_i)_\ell = \widehat{T}(\widehat{L_K} f_i)_\ell = \sum_{k=1}^{N} w_k \sum_{j=1}^{n_i} \theta_{i,j} K(s_k, t_{i,j}) \phi_\ell(s_k), \quad \ell = 1, \ldots, d_1.$$

Step 2: Split the pre-processed data $\{x_i, y_i\}_{i=1}^m$ to the training, validation, and test sets.

Step 3: Use a training set to train a functional net model denoted as F_{γ,β,d_1}.

Step 4: Employ cross-validation to determine the optimal model $F_{\widehat{D}}$ across the 3D hyperparameter grid based on the validation set, and compute the mean squared error (MSE) on the test set using $F_{\widehat{D}}$.

3. Approximation Results

In this section, we state the main approximation results of our proposed functional net with kernel embedding. We begin by deriving rates of approximating nonlinear smooth functionals defined on Besov spaces. The utilization of data-dependent kernels allows us to attain improved approximation rates, particularly when the input function space is smaller, such as in the case of Gaussian RKHSs and mixed-smooth Sobolev spaces. These findings highlight the flexibility of our functional network in exploiting the regularity properties of the input functions.

3.1 Rates of approximating nonlinear smooth functionals on Besov spaces

Our primary finding focuses on the rates of approximating nonlinear smooth functionals in Besov spaces $B^\alpha_{2,\infty}(\Omega)$ with $\alpha > 0$ using our functional net. Besov spaces provide a more nuanced measure of smoothness compared to Sobolev spaces $W_2^\alpha(\Omega)$, which consists of functions whose weak derivatives up to order α exist and belong to $L_2(\Omega)$. For more information on Sobolev and Besov spaces, refer to (DeVore and Lorentz, 1993, Chapter 2).

Specifically, for $r \in \mathbb{N}$, the r-th difference $\Delta_h^r(f, \cdot) : \Omega \to \mathbb{R}$ for a function $f \in L_2(\Omega)$ and $h = (h_1, \ldots, h_d) \in [0, \infty)^d$ is defined as

$$\Delta_h^r(f, x) = \begin{cases} \sum_{j=0}^r \binom{r}{j} (-1)^{r-j} f(x + jh), & \text{if } x \in X_{r,h}, \\ 0 & \text{if } x \notin X_{r,h}, \end{cases} \tag{3.1}$$

where $X_{r,h} := \{x \in \Omega : x + sh \in \Omega, \forall s \in [0, r]\}$. To measure the smoothness of functions, the r-th modulus of smoothness for $f \in L_2(\Omega)$ is defined as

$$\omega_{r,L_2(\Omega)}(f, t) = \sup_{\|h\|_2 \le t} \|\Delta_h^r(f, \cdot)\|_{L_2(\Omega)}, \qquad t \ge 0. \tag{3.2}$$

Let $r = \lfloor \alpha \rfloor + 1$, where $\lfloor \alpha \rfloor$ is the greatest integer that is smaller than or equal to α. Then the Besov space $B^{\alpha}_{2,\infty}(\Omega)$ is defined as

$$B^{\alpha}_{2,\infty}(\Omega) = \left\{ f \in L_2(\Omega) : |f|_{B^{\alpha}_{2,\infty}(\Omega)} < \infty \right\}, \tag{3.3}$$

where $|f|_{B^{\alpha}_{2,\infty}(\Omega)} := \sup_{t>0}(t^{-\alpha}\omega_{r,L_2(\Omega)}(f,t))$ is the semi-norm of $B^{\alpha}_{2,\infty}(\Omega)$, and the norm of $B^{\alpha}_{2,\infty}(\Omega)$ is defined as $\|f\|_{B^{\alpha}_{2,\infty}(\Omega)} = \|f\|_{L_2(\Omega)} + |f|_{B^{\alpha}_{2,\infty}(\Omega)}$.

Suppose that the target functional $F : \mathcal{F} \to \mathbb{R}$ is continuous with the modulus of continuity defined as

$$\omega_F(r) = \sup\{|F(f_1) - F(f_2)| : f_1, f_2 \in \mathcal{F}, \|f_1 - f_2\|_{L_2(\mu)} \le r\}, \tag{3.4}$$

satisfying the condition that

$$\omega_F(r) \le C_F r^{\lambda}, \qquad \text{for some } \lambda \in (0, 1], \tag{3.5}$$

where the measure μ is a positive Borel measure, such as the Lebesgue measure or Gaussian measures restricted on Ω, and C_F is a positive constant.

Let us consider the input function domain as $\Omega = \mathbb{R}^d$. Suppose that the input function space $\mathcal{F}$ is a compact subset of $L_\infty(\mathbb{R}^d) \cap B^{\alpha}_{2,\infty}(\mathbb{R}^d)$, where for any function $f \in \mathcal{F}$, its norm satisfies $\|f\|_{B^{\alpha}_{2,\infty}(\mathbb{R}^d)} \le 1$. In the following theorem, we demonstrate that our functional net can effectively approximate nonlinear smooth target functionals defined on Besov spaces. This result is obtained by selecting the embedding kernel K as a linear combination of Gaussian kernels, as demonstrated in Example 2. In particular, we have:

$$K(x,t) = \sum_{j=1}^{r} \binom{r}{j} (-1)^{1-j} \frac{1}{j^d} \left(\frac{1}{\gamma^2 \pi} \right)^{\frac{d}{2}} k_{j\gamma}(x,t), \tag{3.6}$$

where $r = \lfloor \alpha \rfloor + 1$ is determined by the smoothness of the input function. As shown in Example 2, $L_K f$ belongs to $H_\gamma(\mathbb{R}^d)$. Therefore, we can choose the projection kernel K_0 as k_γ in the projection step, and select the projection basis as the first d_1 eigenfunctions of the integral operator $L^{\mu}_{K_0}$. We note that in this theorem, although the domain $\Omega = \mathbb{R}^d$ is not compact, the selection of the Gaussian measure μ in L^{μ}_K guarantees that it remains a compact operator. Consequently, this allows the application of the Spectral Theorem, as noted in (Fasshauer, 2011, pp. 28).

Theorem 3 *Let $\alpha > 0$, $d, M \in \mathbb{N}$. Assume that the input function space $\mathcal{F}$ is a compact subset of $L_\infty(\mathbb{R}^d) \cap B^{\alpha}_{2,\infty}(\mathbb{R}^d)$, with the condition that $\|f\|_{B^{\alpha}_{2,\infty}(\mathbb{R}^d)} \le 1$ for any function $f \in \mathcal{F}$. Additionally, suppose that the modulus of continuity of the target functional $F : \mathcal{F} \to \mathbb{R}$ adheres to condition (3.5) with $\lambda \in (0, 1]$. By selecting the embedding kernel K as defined in* (3.6)*, the projection kernel as $K_0 = k_\gamma$, and*

$$d_1 = \tilde{c}_1 \frac{\log M}{\log \log M}, \quad \gamma = \tilde{c}_2 \left(\frac{\log \log M}{\log M} \right)^{\frac{1}{d}} \log \log M, \tag{3.7}$$

there exists a functional network F_{NN} that follows the architecture specified in Definition 1, with M nonzero parameters and the depth

$$J \leq \tilde{c}_3 \left(\frac{\log M}{\log \log M} \right)^2, \tag{3.8}$$

such that

$$\sup_{f \in \mathcal{F}} |F(f) - F_{NN}(f)| \leq \tilde{c}_4 \left(\log M\right)^{-\frac{\alpha\lambda}{d}} \left(\log \log M\right)^{\left(\frac{1}{d}+1\right)\alpha\lambda}, \tag{3.9}$$

where $\tilde{c}_1, \tilde{c}_2, \tilde{c}_3, \tilde{c}_4$ are positive constants.

Since $W_2^\alpha(\mathbb{R}^d) \subset B_{2,\infty}^\alpha(\mathbb{R}^d)$, our findings are applicable to approximate nonlinear smooth functionals defined on Sobolev spaces as well. The leading term in the approximation rates aligns with previous studies (Mhaskar and Hahm, 1997; Song et al., 2023a,b), although the $\log \log M$ term exhibits slight variations. The polynomial rates in terms of $\log M$, as opposed to M in traditional function approximation, arise from the curse of dimensionality, given that the input function space is infinite-dimensional. To address this curse of dimensionality, it is necessary to impose stronger smoothness conditions on the target functional.

It is important to note that Theorem 3 is specifically applicable to input function spaces defined as Besov spaces on $\mathbb{R}^d$, rather than on any subset of it. In the following, we will explore how to extend the results in Theorem 3 to Sobolev spaces defined on a domain $\Omega \subset \mathbb{R}^d$. Let us denote $H_0^\alpha(\Omega)$ as the closure of infinitely differentiable compactly supported functions $C_c^\infty(\Omega)$ within the space $W_2^\alpha(\Omega)$. For any function $f \in H_0^\alpha(\Omega)$, we can consider its extension to $\mathbb{R}^d$ by defining $\tilde{f} \in L_2(\mathbb{R}^d)$ as follows:

$$\tilde{f}(x) = \begin{cases} f(x) & x \in \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{3.10}$$

With this construction, it follows that $\tilde{f} \in W_2^\alpha(\mathbb{R}^d)$, and we have the equality for the norms: $\|\tilde{f}\|_{W_2^\alpha(\mathbb{R}^d)} = \|f\|_{W_2^\alpha(\Omega)}$ (Adams and Fournier, 2003, Lemma 3.22). This enables us to obtain the subsequent result.

Theorem 4 *Let $\alpha, d, M \in \mathbb{N}$, $\Omega = [0,1]^d$. Assume that the input function space $\mathcal{F}$ is a compact subset of $H_0^\alpha(\Omega)$, with the condition that $\|f\|_{W_2^\alpha(\Omega)} \leq 1$ for any function $f \in \mathcal{F}$. Additionally, suppose that the modulus of continuity of the target functional $F : \mathcal{F} \to \mathbb{R}$ adheres to condition (3.5) with $\lambda \in (0,1]$. By selecting the embedding kernel K as defined in* (3.6)*, the projection kernel as $K_0 = k_\gamma$, and*

$$d_1 = \tilde{c}_1 \frac{\log M}{\log \log M}, \quad \gamma = \tilde{c}_2 \left(\frac{\log \log M}{\log M} \right)^{\frac{1}{d}} \log \log M, \tag{3.11}$$

there exists a functional network F_{NN} that follows the architecture specified in Definition 1, with M nonzero parameters and the depth

$$J \leq \tilde{c}_3 \left(\frac{\log M}{\log \log M} \right)^2, \tag{3.12}$$

such that

$$\sup_{f\in\mathcal{F}}|F(f)-F_{NN}(f)|\leq \tilde{c}_4\left(\log M\right)^{-\frac{\alpha\lambda}{d}}\left(\log\log M\right)^{\left(\frac{1}{d}+1\right)\alpha\lambda}, \tag{3.13}$$

where $\tilde{c}_1,\tilde{c}_2,\tilde{c}_3,\tilde{c}_4$ *are positive constants.*

It would be intriguing to explore additional, more general scenarios in which the approximation results presented in Theorem 3 can be extended to Besov spaces defined on domains $\Omega\subset\mathbb{R}^d$. Such investigations could enrich our understanding of functional approximation in a broader context and potentially lead to new insights and results applicable to a wider range of applications.

3.2 Rates of approximating nonlinear smooth functionals on Gaussian RKHSs

We then investigate scenarios where the input function spaces are more limited. For example, when the input function spaces are defined as RKHSs induced by specific Mercer kernels, our functional net can still attain satisfactory approximation rates. This is made possible by the flexible choice of the embedding kernel K.

Let us consider $\Omega=[0,1]^d$ and assume that the input function space $\mathcal{F}$ is a compact subset of the unit ball of the Gaussian RKHS $H_\gamma(\Omega)$, which is indeed a subset of the Besov space $B^\alpha_{2,\infty}(\Omega)$ (Steinwart and Christmann, 2008). Furthermore, we assume that the target functional $F:\mathcal{F}\to\mathbb{R}$ meets the same modulus of continuity condition as in (3.5). Our second main result demonstrates rates of approximating nonlinear smooth functionals on Gaussian RKHSs using our functional net. In this case, the projection kernel K_0 is chosen to be the Gaussian kernel k_γ, and the bases used in the projection step consist of the first d_1 eigenfunctions of the integral operator $L^\mu_{K_0}$. The embedding kernel K is chosen as

$$K(x,t)=k_\gamma(x,t)u(t), \tag{3.14}$$

where u is the density of the measure μ.

Theorem 5 *Let* $\gamma>0$, $d,M\in\mathbb{N}$, $\Omega=[0,1]^d$. *Assume that the input function space* $\mathcal{F}$ *is a compact subset of the unit ball of Gaussian RKHS* $H_\gamma(\Omega)$, *and the modulus of continuity of the target functional* $F:\mathcal{F}\to\mathbb{R}$ *satisfies the condition (3.5) with* $\lambda\in(0,1]$. *By selecting the embedding kernel* K *as defined in* (3.14), *the projection kernel* K_0 *as* k_γ, *and*

$$d_1=c_6\left(\log M\right)^{\frac{d}{d+1}}, \tag{3.15}$$

there exists a functional network F_{NN} *that follows the architecture specified in Definition 1, with* M *nonzero parameters and the depth*

$$J\leq \tilde{c}_5\left(\log M\right)^{\frac{2d}{d+1}}, \tag{3.16}$$

such that

$$\sup_{f\in\mathcal{F}}|F(f)-F_{NN}(f)|\leq \tilde{c}_6 e^{-c_7\lambda(\log M)^{\frac{1}{d+1}}}\left(\log M\right)^{\frac{d}{d+1}}, \tag{3.17}$$

where $\tilde{c}_5,\tilde{c}_6,c_6,c_7$ *are positive constants, with* $c_7>\left(\frac{d}{3e}\right)^{\frac{d}{d+1}}$.

It is essential to highlight that although the proof technique employed for this result may seem more straightforward than that used in Theorem 3, it cannot be directly applied to prove Theorem 3. This restriction occurs because this proof technique is applicable only when Sobolev spaces $W_2^\alpha(\Omega)$ are RKHSs, which is true if and only if $\alpha > \frac{d}{2}$ (Berlinet and Thomas-Agnan, 2011, Theorem 121).

In this context, the dominant term of the approximation rate is given by $e^{-c_7 \lambda (\log M)^{\frac{1}{d+1}}}$, where $c_7 > \left(\frac{d}{3e}\right)^{\frac{d}{d+1}}$, along with an additional $\log M$ factor. This rate surpasses the $(\log M)^{-a}$ bound for any polynomial rate of $\log M$ with $a > 0$, which characterizes the situation in Theorem 3. However, it is inferior to the M^{-a} rate for any polynomial rate of M with $a > 0$ in the asymptotic sense when $M \to \infty$. This suggests that even when the input function spaces are infinitely differentiable, we still cannot achieve polynomial approximation rates due to the curse of dimensionality.

3.3 Rates of approximating nonlinear smooth functionals on mixed smooth Sobolev spaces

It is important to note that although the approximation rates for target functionals defined on Gaussian RKHSs in Section 3.2 show significant improvement compared to those on Besov spaces in Section 3.1, the dominant terms of these rates still depend on the dimension d of the input function spaces. Our third main result demonstrates that our functional net can achieve approximation rates that are independent of d when the input function spaces possess certain special properties, specifically in the case of mixed smooth Sobolev spaces.

Let $\partial^{(k)}$ denote the k-th derivative for a multi-index $k \in \mathbb{N}_0^d$. For an integer $\alpha > 0$, the mixed smooth Sobolev spaces $H_{mix}^\alpha(\Omega)$, which consist of functions with square-integrable partial derivatives with all individual orders less than α, are defined as:

$$H_{mix}^\alpha(\Omega) = \left\{ f \in L_2(\Omega) : \partial^{(k)} f \in L_2(\Omega), \forall \|k\|_\infty \leq \alpha \right\}, \tag{3.18}$$

where $\|k\|_\infty = \max_{1 \leq i \leq d} k_i$. The norm on this space is given by:

$$\|f\|_{H_{mix}^\alpha(\Omega)} = \left(\sum_{\|k\|_\infty \leq \alpha} \|\partial^{(k)} f\|_{L_2(\Omega)}^2 \right)^{\frac{1}{2}}. \tag{3.19}$$

The mixed smooth Sobolev space can be understood as a tensor product of univariate Sobolev spaces. Specifically, if we denote

$$k_\nu(x, y; \ell) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \ell |x - y| \right)^\nu K_\nu \left(\sqrt{2\nu} \ell |x - y| \right) \tag{3.20}$$

as the Matérn kernel that reproduces the univariate Sobolev space $H^\alpha([0,1])$ with $\alpha = \nu + \frac{1}{2}$ (where Γ denotes the gamma function, K_ν is the modified Bessel function of the second kind, and ℓ is a positive constant), the eigenvalues of the integral operator associated with this kernel decay polynomially as $\lambda_k = O(k^{-2\alpha})$ (Wendland, 2004). Consequently, the mixed smooth Sobolev space $H_{mix}^\alpha([0,1]^d)$ can be characterized as an RKHS induced by K_α, which

is the pointwise product of the individual Matérn kernels, specifically:

$$K_\alpha(x,y)=\prod_{j=1}^{d}k_\nu(x_j,y_j),\quad \text{for } x,y\in[0,1]^d.$$

Assume that the input function space $\mathcal{F}$ is a compact subset of the unit ball of $H^\alpha_{mix}(\Omega)$, and that the target functional $F:\mathcal{F}\to\mathbb{R}$ adheres to the same modulus of continuity condition as in (3.5). Our third main result demonstrates the rates of approximating this target functional by our functional net, where we select the projection kernel K_0 as K_α, and the projection bases are chosen as the first d_1 eigenfunctions of the integral operator $L^\mu_{K_0}$. The embedding kernel K is chosen as

$$K(x,t)=K_\alpha(x,t)u(t), \tag{3.21}$$

where u is the density of the measure μ.

Theorem 6 *Let $\alpha,d,M\in\mathbb{N}$, $\Omega=[0,1]^d$. Assume that the input function space $\mathcal{F}$ is a compact subset of the unit ball of the mixed smooth Sobolev space $H^\alpha_{mix}(\Omega)$, and the modulus of continuity of the target functional $F:\mathcal{F}\to\mathbb{R}$ adheres to the condition (3.5) with $\lambda\in(0,1]$. By selecting the embedding kernel K as defined in* (3.21)*, the projection kernel K_0 as K_α, and*

$$d_1=C_4\frac{\log M}{\log\log M}, \tag{3.22}$$

there exists a functional network F_{NN} that follows the architecture specified in Definition 1, with M nonzero parameters and the depth

$$J\le C_5\left(\frac{\log M}{\log\log M}\right)^2, \tag{3.23}$$

such that

$$\sup_{f\in\mathcal{F}}|F(f)-F_{NN}(f)|\le C_6(\log M)^{-\alpha\lambda}(\log\log M)^{(d-2)\alpha\lambda}, \tag{3.24}$$

where C_4,C_5,C_6 are positive constants.

It is noteworthy that the dominant term $(\log M)^{-\alpha\lambda}$ in the approximation rates is independent of d, and it only manifests in the negligible $\log\log M$ term. As a result, the overall rate experiences only a slight degradation as d increases. This finding illustrates that our functional network can exploit the regularity of the input functions when they possess mixed smooth properties, particularly by choosing the embedding kernel in a data-dependent manner. It would be intriguing to explore additional scenarios where our functional network can exploit other specific characteristics of the input functions.

4. Generalization Analysis

In this section, we conduct the theoretical analysis of the generalization error for the ERM algorithm applied to learn nonlinear functionals, utilizing our functional net as outlined in Definition 1. Moreover, the notation $\widetilde{O}$ is used to indicate that we conceal the additional logarithmic factors within the conventional O notation.

4.1 Problem settings and notations

We begin by defining the functional regression problem following the classical learning theory framework (Cucker and Smale, 2002; Cucker and Zhou, 2007). We assume that the first-stage data $D = \{f_i, y_i\}_{i=1}^m$ are i.i.d. samples drawn from the true unknown Borel probability distribution ρ on $\mathcal{Z} = \mathcal{F} \times \mathcal{Y}$. Here, $\mathcal{F}$ represents the input function space, which is a compact subset of $L_\infty(\Omega) \cap L_2(\Omega)$ with $\Omega = [0,1]^d$, and satisfies the condition that $\|f\|_{L_2(\Omega)} \leq 1$ for any $f \in \mathcal{F}$, while $\mathcal{Y} = [-L, L]$ denotes the output space bounded by some constant $L > 0$. However, in the case of functional data, the input functions cannot be observed directly; instead, we only have access to observations at discrete points. Thus, what we actually possess in practice are the second-stage data $\widehat{D} = \{\{t_{i,j}, f_i(t_{i,j})\}_{j=1}^{n_i}, y_i\}_{i=1}^m$, where $\{n_i\}_{i=1}^m$ indicates the sample size for the second stage.

Following the classical learning theory framework, our objective is to learn the regression functional

$$F_\rho(f) = \int_{\mathcal{Y}} y d\rho(y|f), \tag{4.1}$$

where $\rho(y|f)$ is the conditional distribution at f induced by ρ. This regression functional is the one that minimizes the generalization error using least squares loss

$$\mathcal{E}(F) = \int_{\mathcal{Z}} (F(f) - y)^2 \, d\rho. \tag{4.2}$$

We denote $\rho_{\mathcal{F}}$ as the marginal distribution of ρ on $\mathcal{F}$, and $\left(L^2_{\rho_{\mathcal{F}}}, \|\cdot\|_\rho\right)$ as the space of square integrable functionals w.r.t. $\rho_{\mathcal{F}}$.

The hypothesis space $\mathcal{H}_{d_1,M}$ we use for the ERM algorithm is defined as

$$\begin{aligned}\mathcal{H}_{d_1,M} = \{&H_{NN} \circ T : H_{NN} \text{ is a structured deep ReLU neural network with input} \\ &\text{dimension } d_1, \text{ depth } J = d_1^2 + d_1 + 1, M \text{ non-zero parameters, and} \\ &\text{whose output has the following formulation (4.4)}\}.\end{aligned} \tag{4.3}$$

It takes the embedded function $L_K f$ as input, and T serves as a projection step with the bases chosen as $\{\phi_i\}_{i=1}^{d_1}$, which consist of the eigenfunctions of the integral operator induced by the projection kernel K_0:

$$T(g) = \left[\int_\Omega g(t)\phi_1(t)d\mu(t), \int_\Omega g(t)\phi_2(t)d\mu(t), \cdots, \int_\Omega g(t)\phi_{d_1}(t)d\mu(t)\right]^T.$$

The architecture of the deep ReLU neural network H_{NN} is specifically designed to approximate Hölder continuous functions on $[-R, R]^{d_1}$, and it has the explicit formulation given by

$$H_{NN}(y) = \sum_{i=1}^{(N+1)^{d_1}} c_i \psi\left(\frac{N}{2R}(y - b_i)\right), \quad y \in [-R, R]^{d_1}, \tag{4.4}$$

where $c_i \in \mathbb{R}, b_i \in \mathbb{R}^{d_1}$ satisfying $|c_i| \leq L$, $\|b_i\|_\infty \leq R$, and $\psi : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\psi(y) = \sigma\Big(\min\Big\{\min_{k \neq j}(1 + y_k - y_j), \min_k(1 + y_k), \min_k(1 - y_k)\Big\}\Big). \tag{4.5}$$

Moreover, by Lemma 14, N has the following relationship with M:

$$\bar{C}_1 d_1^4 (N+1)^{d_1} \leq M \leq \bar{C}_2 d_1^4 (N+1)^{d_1}, \tag{4.6}$$

where $\bar{C}_1$ and $\bar{C}_2$ are positive constants. Notice that

$$\begin{aligned}\|T(L_K f)\|_\infty &\leq \|L_K f\|_{L_2(\mu)} \leq \sqrt{\mu(\Omega)}\|L_K f\|_{L_\infty(\Omega)} \\ &= \sqrt{\mu(\Omega)} \sup_{x\in\Omega} \int_\Omega K(x,t) f(t) dt \leq \sqrt{\mu(\Omega)} C_K \|f\|_{L_2(\Omega)} \leq \sqrt{\mu(\Omega)} C_K.\end{aligned}$$

Let $C_\mu = \mu(\Omega)$, then we can set $R = \sqrt{C_\mu} C_K$. Denote $\mathcal{H}_{NN}$ as the function space of H_{NN}. The following proposition demonstrates that the difference in the outputs of H_{NN} for different inputs can be bounded by the difference in those inputs.

Proposition 7 *For any $h \in \mathcal{H}_{NN}$ and $x, y \in \mathbb{R}^{d_1}$, we have*

$$|h(x) - h(y)| \leq \frac{L}{\sqrt{C_\mu} C_K} N (N+1)^{d_1} (d_1^2 + d_1) ||x - y||_1. \tag{4.7}$$

Furthermore, the empirical generalization error using the first-stage data $D = \{f_i, y_i\}_{i=1}^m$ is defined as

$$\mathcal{E}_D(F) = \frac{1}{m} \sum_{i=1}^m \left(F(f_i) - y_i\right)^2. \tag{4.8}$$

If we consider employing the hypothesis space defined in (4.3) along with the kernel embedding step within the first-stage ERM algorithm, the first-stage empirical generalization error can be formulated as

$$\mathcal{E}_D(H \circ L_K) = \frac{1}{m} \sum_{i=1}^m \left(H \circ L_K f_i - y_i\right)^2. \tag{4.9}$$

Moreover, if we denote

$$H_D = \arg \min_{H \in \mathcal{H}_{d_1, M}} \mathcal{E}_D(H \circ L_K), \tag{4.10}$$

then the first-stage empirical target functional has the form

$$F_D = H_D \circ L_K. \tag{4.11}$$

However, as previously noted, in practice, we can only acquire the second-stage data $\widehat{D} = \{\{t_{i,j}, f(t_{i,j})\}_{j=1}^{n_i}, y_i\}_{i=1}^m$. Therefore, rather than using the kernel embedding $L_K f_i$, we must rely on a quadrature scheme as outlined in (2.11):

$$\widehat{L_K} f_i = \sum_{j=1}^{n_i} \theta_{i,j} K(\cdot, t_{i,j}).$$

Furthermore, if we consider utilizing the hypothesis space defined in (4.3) within the second-stage ERM algorithm, the second-stage empirical generalization error can be expressed as

$$\mathcal{E}_{\widehat{D}}(H \circ \widehat{L_K}) = \frac{1}{m} \sum_{i=1}^m \left(H \circ \widehat{L_K} f_i - y_i\right)^2. \tag{4.12}$$

Similarly, if we denote

$$H_{\widehat{D}} = \arg \min_{H \in \mathcal{H}_{d_1,M}} \mathcal{E}_{\widehat{D}}(H \circ \widehat{L_K}), \tag{4.13}$$

then the second-stage empirical target functional has the form

$$F_{\widehat{D}} = H_{\widehat{D}} \circ L_K. \tag{4.14}$$

Finally, we define the projection operator π_L on the functional space $F : \mathcal{F} \to \mathbb{R}$ as

$$\pi_L(F)(f) = \begin{cases} L, & \text{if } F(f) > L, \\ -L, & \text{if } F(f) < -L, \\ F(f), & \text{if } -L \leq F(f) \leq L. \end{cases}$$

Since the regression functional F_ρ is bounded by L, we will use the truncated empirical target functional

$$\pi_L F_{\widehat{D}}$$

as the final estimator.

4.2 A two-stage oracle inequality

Given that the ERM algorithm operates on the second-stage data for the FDA, in contrast to traditional regression problems that only consider first-stage data, we must develop a new oracle inequality for the generalization analysis of the ERM algorithm applied to our functional network. The primary approach involves employing a two-stage error decomposition method, where the first-stage empirical error serves as an intermediary term in this decomposition of errors.

In the following, for the convenience, we denote $\mathcal{H} = \mathcal{H}_{d_1,M}$, and $u_i = L_K f_i$, $\hat{u}_i = \widehat{L_K} f_i$, for $i = 1, 2, \ldots, m$.

Proposition 8 *Let $F_{\widehat{D}}$ be the empirical target functional defined in* (4.14)*, and $\pi_L F_{\widehat{D}}$ be the truncated empirical target functional, then*

$$\mathcal{E}\left(\pi_L F_{\widehat{D}}\right) - \mathcal{E}\left(F_\rho\right) \leq I_1 + I_2 + I_3. \tag{4.15}$$

where

$$\begin{aligned} I_1 &= \left\{\mathcal{E}\left(\pi_L F_{\widehat{D}}\right) - \mathcal{E}\left(F_\rho\right)\right\} - 2\left\{\mathcal{E}_D\left(\pi_L F_{\widehat{D}}\right) - \mathcal{E}_D\left(F_\rho\right)\right\}, \\ I_2 &= 2\left\{\mathcal{E}_D\left(\pi_L F_D\right) - \mathcal{E}_D\left(F_\rho\right)\right\}, \\ I_3 &= 16L \sup_{H \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \left|H(u_i) - H(\hat{u}_i)\right|. \end{aligned} \tag{4.16}$$

Our next objective is to establish a two-stage oracle inequality for analyzing the generalization ability of the ERM algorithm applied to our functional network. This will be influenced by the capacity of the hypothesis space we employ, as well as the quadrature scheme $\widehat{L_K} f$ in (2.11), which is used for approximating $L_K f$ as described in (2.10).

In this theoretical analysis, we consider the utilization of a theoretically optimal kernel quadrature scheme described in (Bach, 2017, pp. 21–22). To achieve an approximation accuracy ϵ, the quadrature formula is given by:

$$\widehat{L_K} f_i = \sum_{j=1}^{n_i} \theta_{i,j} K(\cdot, t_{i,j}), \tag{4.17}$$

where the observation points $\{t_{i,j}\}_{j=1}^{n_i}$ are i.i.d. sampled from an optimal distribution on Ω with density τ w.r.t. $d\mathcal{U}$ satisfying

$$\tau(x) \propto \sum_{k\geq 1} \frac{\lambda_k}{\lambda_k + \epsilon} \phi_k(x)^2, \tag{4.18}$$

and the weights $\{\theta_{i,j}\}_{j=1}^{n_i}$ are computed by minimizing

$$\sum_{j=1}^{n_i}\sum_{k=1}^{n_i} \theta_{i,j}\theta_{i,k} K(t_{i,j}, t_{i,k}) - 2\sum_{j=1}^{n_i} \theta_{i,j} \int_\Omega K(t, t_{i,j}) f_i(t) d\mathcal{U}(t),$$

subject to $\sum_{j=1}^{n_i} \theta_{i,j}^2 \leq 4/n$ ((Bach, 2017, pp. 21–22), (Kanagawa et al., 2016, pp. 3–4)).

We are now prepared to present the two-stage oracle inequality for the ERM algorithm when the hypothesis space is defined by our functional network. In this context, the embedding kernel K and the projection kernel K_0 are selected as specified in Section 3.

Theorem 9 *Let $m, n \in \mathbb{N}$, $\Omega = [0,1]^d$. Assume that the second-stage sample sizes are equal, specifically $n_1 = n_2 = \cdots = n_m = n$, the bound of the output space $L \geq 1$, the input function $f \in L_\infty(\Omega) \cap L_2(\Omega)$ and satisfies the norm constraint $\|f\|_{L_2(\Omega)} \leq 1$. Let $F_{\widehat{D}}$ denote the empirical target functional defined in* (4.14)*, F_ρ represent the regression functional defined in* (4.1)*, then we have the following inequality*

$$E\,||\pi_L F_{\widehat{D}} - F_\rho||_\rho^2 \leq \frac{c_1' JM \log M \log m}{m} + \frac{c_3'}{C_K} M^{1+\frac{1}{d_1}} \sqrt{\lambda_q} + 2 \inf_{F \in \{H \circ L_K : H \in \mathcal{H}\}} \|F - F_\rho\|_\rho^2, \tag{4.19}$$

where the expectation is taken with respect to the first-stage data D and the second-stage data $\widehat{D}$, $\{\lambda_i\}$ are the eigenvalues of the integral operator $L_{K_0}^\mu$ associated with the projection kernel K_0. Moreover, $q = \frac{c_5' n}{\log n}$, $J = d_1^2 + d_1 + 1$, $C_K = \sup_{x,t} |K(x,t)|$ depending on the embedding kernel K, and c_1', c_3', c_5' are positive constants.

4.3 Generalization error bounds

Building on the two-stage oracle inequality outlined in Theorem 9, we are now prepared to establish the generalization error bounds for the ERM algorithm applied to our functional network. We first examine the generalization analysis scenario where the input function space is Sobolev space, by leveraging the approximation results presented in Section 3.1. For the dimension reduction step in this context, the embedding kernel K is selected as defined in (3.6), the projection kernel is selected as $K_0 = k_\gamma$, and the hyperparameters d_1, γ are selected according to (3.11) within Theorem 4.

Theorem 10 *Let $\alpha, m, n \in \mathbb{N}$, $\Omega = [0,1]^d$. Assume that the input function space $\mathcal{F}$ is a compact subset of $H_0^\alpha(\Omega)$, with the condition that $\|f\|_{W_2^\alpha(\Omega)} \leq 1$ for any function $f \in \mathcal{F}$, and the modulus of continuity of the regression functional $F_\rho : \mathcal{F} \to \mathbb{R}$ satisfies the condition (3.5) with $\lambda \in (0,1]$. Additionally, suppose that the second-stage sample sizes are equal, specifically $n_1 = n_2 = \cdots = n_m = n$. By choosing the number of nonzero parameters M in the functional network and the second-stage sample size n as*

$$M = \left\lfloor \frac{m}{(\log m)^{\frac{2\alpha\lambda}{d}+4}} \right\rfloor, \qquad n \geq \hat{C}_2 (\log m)^d \log\log m, \tag{4.20}$$

we obtain the following generalization error bound

$$E\,||\pi_L F_{\widehat{D}} - F_\rho||_\rho^2 \leq \tilde{C}_2\,(\log m)^{-\frac{2\alpha\lambda}{d}}\,(\log\log m)^{2\left(\frac{1}{d}+1\right)\alpha\lambda}, \tag{4.21}$$

where $\tilde{C}_2, \hat{C}_2$ are positive constants.

Since $\log\log m$ is negligible compared to $\log m$, the generalization error bounds we obtain for the Sobolev input function spaces converge at rates of $\widetilde{O}((\log m)^{-\frac{2\alpha\lambda}{d}})$. Similarly, we can derive the learning rates for input function spaces that are Gaussian RKHSs and mixed smooth Sobolev spaces by applying the approximation results from Section 3.2 and Section 3.3, respectively. The proofs of the following two theorems are omitted, as they follow the same approach as in the proof of Theorem 10. In Theorem 11, we select the embedding kernel K as defined in (3.14) and the projection kernel K_0 as the Gaussian kernel k_γ, leading to an eigenvalue decay of $\lambda_q = O(e^{-2c_2 q^{\frac{1}{d}}})$, and the hyperparameter d_1 is selected according to (3.15) within Theorem 5. In Theorem 12, we select the embedding kernel K as defined in (3.21) and the projection kernel K_0 as the Matérn kernel K_α, resulting in an eigenvalue decay of $\lambda_q = O(q^{-2\alpha})$, and the hyperparameter d_1 is selected according to (3.22) within Theorem 6.

Theorem 11 *Let $\gamma > 0, m, n \in \mathbb{N}$, $\Omega = [0,1]^d$. Assume that the input function space $\mathcal{F}$ is a compact subset of the unit ball of the Gaussian RKHS $H_\gamma(\Omega)$, and the modulus of continuity of the regression functional $F_\rho : \mathcal{F} \to \mathbb{R}$ satisfies the condition (3.5) with $\lambda \in (0,1]$. Additionally, suppose that the second-stage sample sizes are equal, specifically $n_1 = n_2 = \cdots = n_m = n$. By choosing the number of nonzero parameters M and second-stage sample size n as*

$$M = \left\lfloor \frac{m e^{-2c_7\lambda(\log m)^{\frac{1}{d+1}}}}{(\log m)^4} \right\rfloor, \qquad n \geq \hat{C}_3 (\log m)^d \log\log m, \tag{4.22}$$

we obtain the following generalization error bound

$$E\,||\pi_L F_{\widehat{D}} - F_\rho||_\rho^2 \leq \tilde{C}_3 e^{-2c_7\lambda(\log m)^{\frac{1}{d+1}}}\,(\log m)^{\frac{2d}{d+1}}, \tag{4.23}$$

where $\tilde{C}_3$, $\hat{C}_3$, c_7 are positive constants with $c_7 > \left(\frac{d}{3e}\right)^{\frac{d}{d+1}}$.

Since $(\log m)^{\frac{2d}{d+1}}$ is negligible compared to $e^{-2c_7\lambda(\log m)^{\frac{1}{d+1}}}$, the generalization error bounds obtained for the Gaussian RKHS input function space exhibit convergence rates of $\widetilde{O}(e^{-2c_7\lambda(\log m)^{\frac{1}{d+1}}})$, with $c_7 > \left(\frac{d}{3e}\right)^{\frac{d}{d+1}}$. This rate is superior to $(\log m)^{-a}$ for any $a > 0$, yet inferior to m^{-a} for any $a > 0$ in the asymptotic context as $m \to \infty$. This suggests that we are still unable to achieve polynomial learning rates for infinitely differentiable functions.

Theorem 12 *Let $\alpha, m, n \in \mathbb{N}$, $\Omega = [0,1]^d$. Assume that the input function space $\mathcal{F}$ is a compact subset of the unit ball of the mixed smooth Sobolev space $H^{\alpha}_{mix}(\Omega)$, and the modulus of continuity of the regression functional $F_\rho : \mathcal{F} \to \mathbb{R}$ satisfies the condition (3.5) with $\lambda \in (0,1]$. Additionally, suppose that the second-stage sample sizes are equal, specifically $n_1 = n_2 = \cdots = n_m = n$. By choosing the number of nonzero parameters M and second-stage sample size n as*

$$M = \left\lfloor \frac{m}{(\log m)^{2\alpha\lambda+4}} \right\rfloor, \qquad n \geq \hat{C}_4 m^{\frac{1}{\alpha}} (\log m)^{1+\frac{1}{C_4\alpha}+2\lambda-2d\lambda-\frac{4}{\alpha}}, \tag{4.24}$$

we obtain the following generalization error bound

$$E\,||\pi_L F_{\widehat{D}} - F_\rho||_\rho^2 \leq \tilde{C}_4\,(\log m)^{-2\alpha\lambda}\,(\log\log m)^{2(d-2)\alpha\lambda}\,, \tag{4.25}$$

where $\hat{C}_4$, C_4, $\tilde{C}_4$ are positive constants.

Since $\log\log m$ is negligible compared to $\log m$, the generalization error bound we obtain for the mixed smooth Sobolev input function spaces shows convergence rates of $\widetilde{O}((\log m)^{-2\alpha\lambda})$. This rate is independent of the dimension d of the input function spaces, similar to that in the approximation rates.

5. Related Work and Discussion

Classical statistical approaches to functional regression have been extensively developed within the kernel methods framework. For example, Yuan and Cai (2010) tackled the functional linear model, employing an RKHS framework to derive the minimax optimal convergence rates for estimating the slope function. Building on this, Cai and Yuan (2012) extended the minimax analysis within the same linear model to the problem of prediction, and further developed an adaptive estimator that achieved the optimal rate without prior knowledge of the covariance operator's eigenvalue decay. Moving beyond the linear paradigm, Meister (2016) addressed the fully nonparametric setting for both regression and classification by considering the Nadaraya-Watson estimator with usage of a kernel function. By leveraging a metric entropy approach without imposing an algebraic structure for the input space, this work revealed a fundamental shift: the minimax-optimal rates become logarithmic, i.e., $O((\log n)^{\frac{-2\lambda}{\gamma}})$, where γ denotes the order of the kernel's metric entropy, and λ represents the Hölder exponent of the target functional's modulus of continuity. This contrasts sharply with the polynomial rates achievable in linear models, highlighting the intrinsic difficulty of nonparametric inference in infinite-dimensional spaces. By substituting the metric entropy of the input spaces considered in our paper, we recover the same convergence rates as in their work, up to additional logarithmic terms.

Compared with Yuan and Cai (2010); Cai and Yuan (2012), which focus exclusively on the functional linear regression model, we study nonlinear functional regression. Although Meister (2016) also investigates nonlinear functional regression, their analysis is based on the classical Nadaraya–Watson (NW) estimator, which differs from our KEFNN methodology. The NW estimator is a local weighted average of the responses in the training data, therefore it may suffer from several inherent limitations, e.g., it might be sensitive to the curse of dimensionality (Hastie et al., 2009; Wasserman, 2006), it lacks built-in mechanisms for feature extraction, as it is a memory-based method that operates directly on the raw input space without learning a parametric transformation (Hastie et al., 2009), and its performance depend on the choice of a global bandwidth parameter, making it potentially non-adaptive to heterogeneous smoothness in the target functional (Wand and Jones, 1994). In contrast, our KEFNN model offers a more flexible framework: The initial kernel embedding step provides a discretization-invariant and smooth representation of the input functions, which enhances robustness to noisy and irregularly sampled data, and the subsequent deep neural network acts as an adaptive feature extractor, capable of learning complex features from the embedded data that are adaptive to the regression task, thereby potentially mitigating the curse of dimensionality.

The analysis of the NW estimator typically relies on its explicit form as a local average. Our theoretical contribution, however, rests on two novel pillars that are absent in the analysis of Meister (2016). First, we provide a rigorous approximation analysis of the KEFNN architecture, demonstrating its ability to approximate smooth nonlinear functionals over various input spaces. This analysis highlights the expressive power of KEFNN, whereas the NW estimator, as a relatively limited kernel smoother, lacks such flexibility. Second, acknowledging the practical reality of functional data, we explicitly model the two-stage data generation process and derive a novel two-stage oracle inequality for the estimator obtained from two-stage data. It is by integrating these two key components—the approximation power of KEFNN and the statistical analysis of the two-stage sampling—that we derive our generalization error bounds. Our theoretical contributions provide a theoretical foundation for studying nonlinear functional regression with deep learning.

Numerous studies have explored the applications of deep learning in the FDA. For instance, in the context of solving parametric partial differential equations (PDEs), works such as Khoo et al. (2021) and Lu et al. (2021) focus on using operator neural networks to learn the relationship between the parametric function space and the solution space. Regarding inverse scattering problems, Khoo and Ying (2019) and Wei and Chen (2019) employ deep learning techniques to learn an operator that maps the observed data function space to the parametric function space that characterizes the underlying PDE for high-frequency phenomena. In the field of signal processing, research works like Andreotti et al. (2016) and Grais and Plumbley (2017) have investigated using deep learning to learn a nonlinear operator that maps a signal function to one or multiple other signal functions. Additionally, there have been various efforts in image processing tasks utilizing deep neural networks, including phase retrieval (Deng et al., 2020), image super-resolution (Qiao et al., 2021), image inpainting (Qin et al., 2021), and image denoising (Tian et al., 2020).

The theoretical exploration of artificial neural networks has been ongoing for over thirty years. Recent studies have quantitatively shown that deep ReLU neural networks possess approximation capabilities on par with traditional deep neural networks that use infinitely

differentiable activation functions, such as sigmoid and tanh functions (Telgarsky, 2016; Yarotsky, 2017; Suzuki, 2019; Zhou, 2020; Mao et al., 2022). Regarding approximation results in FDA, the foundational result is the universal approximation theorem for nonlinear operators established in Chen and Chen (1995). Subsequent researches by Mhaskar and Hahm (1997), Bhattacharya et al. (2021), Kovachki et al. (2021), and Lanthaler et al. (2022) have further quantitatively examined the expressivity of deep neural networks in approximating operators. More recently, Mhaskar (2023) investigated the local approximation of operators through deep neural networks. Building on these approximation results, generalization analyses for the ERM algorithm over deep ReLU neural networks have been thoroughly developed in (Chui et al., 2019; Schmidt-Hieber, 2020; Mao et al., 2021). Recent advancements in FDA have taken a further step by analyzing the generalization error in operator learning using deep neural networks (Lanthaler et al., 2022; de Hoop et al., 2023). In addition, Liu et al. (2024) has focused on the nonparametric estimation of Lipschitz operators with deep neural networks, providing non-asymptotic bounds for the generalization error of the ERM algorithm based on a suitably selected class of networks.

Discretization-invariant learning is also an important area of research within the FDA. It focuses on learning within infinite-dimensional function spaces while effectively managing various discrete representations of functions either as inputs or outputs in a learning model. To develop mesh-independent models, a traditional approach involves pre-processing the input data and post-processing the output data to ensure that the processed data aligns with the requirements of deep neural networks. Common techniques for processing include interpolation, padding, resizing (Keys, 1981), and cropping (Ravuri et al., 2021). Recent studies have introduced additional methods for discretization invariant learning that use kernel integral operators. In this framework, the input function f is transformed to the next layer via a linear integral transformation represented as $v(x) = \int_\Omega K(x, t; \theta) f(t) dt$, followed by a nonlinear activation. Here, $K(x, t; \theta)$ denotes the integral kernel parameterized by θ. This integral transformation can be applied independently of the discretization of the input function f. The choice of the integral kernel can include convolutional kernels, leading to pure convolutional neural networks (Ronneberger et al., 2015; Guo et al., 2016; Khoo et al., 2021), as well as parameterized neural networks (Anandkumar et al., 2020; Li et al., 2020, 2021; Ong et al., 2022) that are used in tasks such as image classification, solving parametric PDEs, and addressing initial value problems.

We note that from another perspective, employing kernel integral transformation with a smooth kernel can be regarded as a method of kernel smoothing (Wand and Jones, 1994). This technique is commonly used in various domains, including image processing (Chung, 2013) and functional linear regression based on FPCA (Yao et al., 2005b; Zhou et al., 2022). For instance, in the context of image processing, selecting K as a translation-invariant kernel effectively results in a convolution. In functional linear regression, an empirical integral transformation defined as $\widehat{L_K} f_i = \frac{1}{n_i} \sum_{j=1}^{n_i} f_i(t_{i,j}) K(\cdot, t_{i,j})$ serves as a pre-smoothing technique for the input function data that is observed discretely, leveraging a smoothing density kernel.

As a summary, in this paper, we improve the theoretical comprehension of the FDA by examining the expressiveness and generalization capabilities of the functional deep neural network with kernel embedding that we propose. As detailed in Table 1, we establish explicit rates for approximating nonlinear smooth functionals across various input function spaces,

Function classes	Approximation error	Estimation error
Sobolev spaces $H_0^\alpha(\Omega)$	$\widetilde{O}\left((\log M)^{-\frac{\alpha\lambda}{d}}\right)$	$\widetilde{O}\left((\log m)^{-\frac{2\alpha\lambda}{d}}\right)$
Gaussian RKHSs $H_\gamma(\Omega)$	$\widetilde{O}\left(e^{-c_7\lambda(\log M)^{\frac{1}{d+1}}}\right)$	$\widetilde{O}\left(e^{-2c_7\lambda(\log m)^{\frac{1}{d+1}}}\right)$
mixed smooth Sobolev spaces $H_{mix}^\alpha(\Omega)$	$\widetilde{O}\left((\log M)^{-\alpha\lambda}\right)$	$\widetilde{O}\left((\log m)^{-2\alpha\lambda}\right)$

Table 1: Summary of approximation and learning rates achieved by our functional net. M denotes the number of non-zero parameters in the functional net, and m represents the first-stage sample size.

such as Sobolev spaces, Gaussian RKHSs, and mixed-smooth Sobolev spaces. Based on these approximation results, we further derive explicit learning rates for the ERM algorithm applied to our functional network, when the second-stage sample size n is sufficiently large based on the employed quadrature scheme. This is achieved through a novel two-stage oracle inequality that takes into account both the first-stage sample size m and the second-stage sample size.

6. Numerical Simulations

In this section, we conduct standard numerical simulations to evaluate the performance and advantages of our proposed functional network with kernel embedding (KEFNN), which is trained as outlined in Algorithm 1 in Section 2. We start by examining the numerical effectiveness of our model through simulated examples, comparing it with functional networks that use different dimension reduction methods, as well as with a standard deep neural network. Following this, we carry out numerical simulations on synthetic data to gain further insights into our model's behavior. This includes assessing the impact of the first-stage sample size, second-stage sample size, network depth, and noise levels on performance, validating the discretization invariant property, and exploring the effects of various quadrature schemes on performance. Finally, we evaluate the effectiveness of KEFNN on several small real functional datasets.

6.1 Comparison with baseline approaches

To evaluate the performance of our model against baseline approaches, we adopt the same data-generating process used in Yao et al. (2021), which is commonly employed in functional data simulations. The input random function is generated through the process $f(t) = \sum_{k=1}^{50} c_k\phi_k(t)$ with $t \in [0,1]$, where $\phi_1(t) = 1$, $\phi_k(t) = \sqrt{2}cos((k-1)\pi t)$ for $k \in \{2,3,\ldots,50\}$. The coefficients are defined as $c_k = z_k r_k$, with r_k being i.i.d. uniform random variables on $[-\sqrt{3},\sqrt{3}]$. The first-stage data consists of 4000 samples, while the second-stage data of $f_i(t)$ are observed at discrete points $\{t_{i,j}\}_{j=1}^{51}$ that are equally spaced on $[0,1]$. The training, validation, and test split follows a ratio of $64:16:20$.

We assess the performance of our method in comparison to some baseline approaches that use different dimension reduction techniques, including the direct use of discrete observation points ("Raw data") (Rossi et al., 2002), projection by basis approaches employing

"B-spline" basis (Rossi et al., 2005), "FPCA" basis (Rossi et al., 2005), and the neural network basis used in "AdaFNN" (Yao et al., 2021). It is important to note that the number of bases in "B-spline" and "FPCA" should not be too small (as this would fail to capture the main information in the input function) or too large (which could lead to overfitting due to noisy observations). In contrast, our method leverages kernel embedding as a pre-smoothing step, allowing for a larger number of eigenfunctions during the projection step, thus retaining as much useful information as possible without being affected by noise. The number of bases in AdaFNN is limited to a maximum of four, as the target functionals depend on at most two bases in the cases considered in the simulation. We provide the mean squared error (MSE) on the test set for different methods in Table 2, including the best test-set MSE reported in Yao et al. (2021) for the other approaches.

For *case 1*, we set $z_1 = z_3 = 5$, $z_5 = z_{10} = 3$, and $z_k = 1$ for other k. The response $Y = (\langle f, \phi_5 \rangle)^2 = c_5^2$ has a nonlinear relationship with the input function and only depends on one Fourier basis. We don't consider noise in this case. For *case 2*, the other settings are the same as case 1, except that the observations of $f_i(t)$ at each point include a Gaussian noise $N(0, \sigma_1^2)$ with $\sigma_1^2 = 11.4$, and the observations of the response Y also have a Gaussian noise $N(0, \sigma_2^2)$ with $\sigma_2^2 = 0.3$. For *case 3*, $z_k = 1$ for all k, and $Y = \langle f, \beta_2 \rangle + (\langle f, \beta_1 \rangle)^2$, with $\beta_1(t) = (4 - 16t) \cdot 1\{0 \le t \le 1/4\}$ and $\beta_2(t) = (4 - 16|1/2 - t|) \cdot 1\{1/4 \le t \le 3/4\}$. The noises are also included with $\sigma_1^2 = 5$, $\sigma_2^2 = 0.1$. For *case 4*, the settings are the same as case 3, except that the noises in the observation of Y are enlarged, i.e., the variance is doubled with $\sigma_2^2 = 0.2$.

The deep neural network architectures after the dimension reduction approach are consistent across all methods, each comprising three hidden layers with 128 neurons in each layer, implemented using PyTorch. The functional input and response data are standardized entry-wise for all learning tasks. All functional networks are trained for 500 epochs, with the best model selected based on validation loss. The Adam optimizer is employed for optimization, with a learning rate set at 3×10^{-4}. In our model, we use a Gaussian kernel with bandwidth γ as the embedding kernel and use the first d_1 eigenfunctions of its integral operator, explicitly formulated using Hermite polynomials (Fasshauer, 2011, pp. 28), as the bases for the projection step. The measure μ in the integral operator is characterized by a Gaussian distribution, whose density function is $u(x) = \frac{\beta}{\sqrt{\pi}} \exp^{-\beta^2 x^2}$, where β acts as a global scaling parameter. weights $\{\theta_{i,j}\}_{j=1}^n$ are determined using the composite trapezoidal rule. Numerical integration is employed to compute the coefficient vector for the projection step. The hyperparameters γ, β, d_1 are selected through cross-validation over a 3D grid of parameters. For the first three cases, the optimal hyperparameters are $\gamma = 0.033$, $\beta = 0.008$, $d_1 = 150$. In case 4, the other hyperparameters remain the same, with the exception that $\gamma = 0.02$.

From Table 2, it is evident that KEFNN consistently surpasses other dimension reduction techniques, primarily due to the careful tuning of hyperparameters utilizing the information from the entire dataset. In these simulations, Gaussian kernels are exclusively employed as the embedding kernel; however, exploring alternative embedding kernels could potentially enhance performance further. Additionally, as shown by the numerical computation (2.12) of the coefficient vector in our dimension reduction method, it relies solely on hyperparameters without introducing any training parameters that might escalate computational demands—such as those incurred by eigenfunction estimation in FPCA or the

Table 2: Comparison of the test-set MSE for functional deep neural networks utilizing different dimension reduction methods.

	Case 1	Case 2	Case 3	Case 4
Raw data+NN	0.038	0.275	0.334	0.339
B-spline+NN	0.019	0.206	0.251	0.257
FPCA+NN	0.023	0.134	0.667	0.693
AdaFNN	0.003	0.127	0.193	0.207
KEFNN	**0.001**	**0.100**	**0.142**	**0.191**

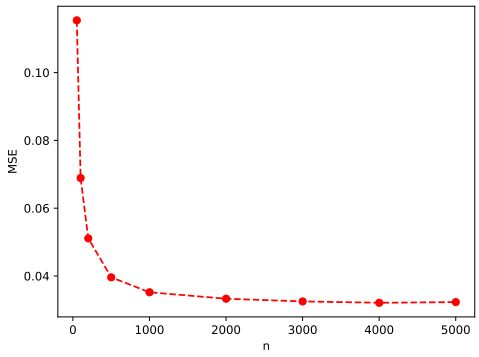
optimization of free parameters in the neural network basis of AdaFNN. Moreover, dimension reduction methods based on B-spline and FPCA depend purely on input function data while ignoring response data, which could result in the selection of bases misaligned with the true target functional. Conversely, in our dimension reduction strategy, hyperparameters are determined using both input and response data. These aspects collectively account for the superior performance of KEFNN compared to the baseline approaches.

6.2 Additional insights on KEFNN

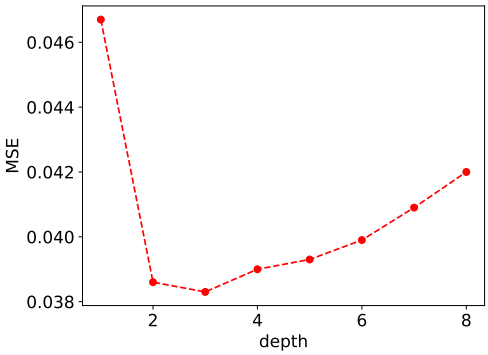
In this subsection, we conduct numerical simulations to further explore the behavior of our model. Specifically, we analyze the impact of varying the first-stage sample size m and the second-stage sample size n, the depth of KEFNN, the level of observational noise in input functions and responses, as well as the effects of discretization and quadrature schemes on generalization performance. The target functional remains consistent with case 3 from the preceding subsection, expressed as $Y = \langle f, \beta_1 \rangle + (\langle f, \beta_2 \rangle)^2$. Unless otherwise indicated, the discrete points are uniformly spaced across the interval $[0, 1]$. Furthermore, the hyperparameter settings in this subsection are maintained as $\gamma = 0.02$, $\beta = 0.008$, and $d_1 = 150$.

We begin by examining the effect of the second-stage sample size n on generalization performance, with the first-stage sample size fixed at $m = 4000$ and the noise variances set to $\sigma_1^2 = 3$ and $\sigma_2^2 = 0.05$. As illustrated in Figure 2(a), a phase transition phenomenon is observed in our model around $n = 4000$, indicating that further increasing n beyond this point does not lead to improvements in generalization performance. This requirement for the second-stage sample size is approximately equivalent to the first-stage sample size when trapezoidal rules are applied for the quadrature problem.

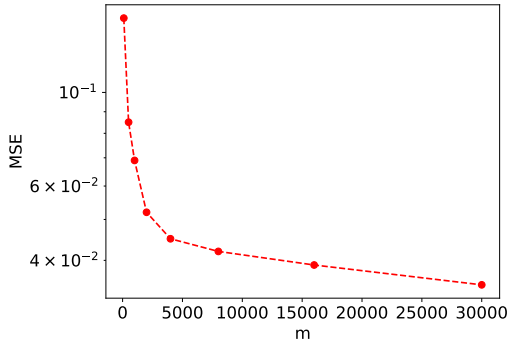
Next, we investigate how the depth of KEFNN influences generalization performance within the under-parameterized regime. For this study, we set the first-stage sample size to $m = 5000$, the second-stage sample size to $n = 500$, and the noise variances to $\sigma_1^2 = 3$ and $\sigma_2^2 = 0.05$. The width of the deep neural network in KEFNN is fixed at 16. Figure 2(b) reveals that as the depth increases, the generalization error initially decreases but then begins to rise. This behavior aligns with our theoretical analysis in Section 4, which suggests that the generalization error is minimized when the number of nonzero parameters in KEFNN follows a specific rate relative to the first-stage sample size. This finding provides partial guidance on how to conduct model selection for KEFNN.



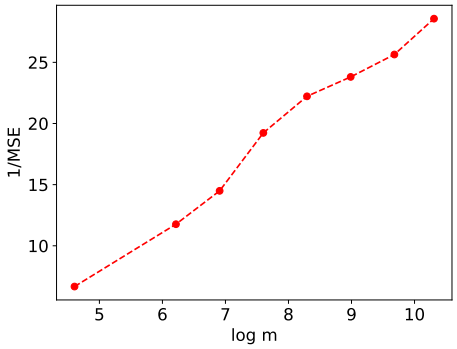
(a) Test-set MSE w.r.t. second-stage sample size



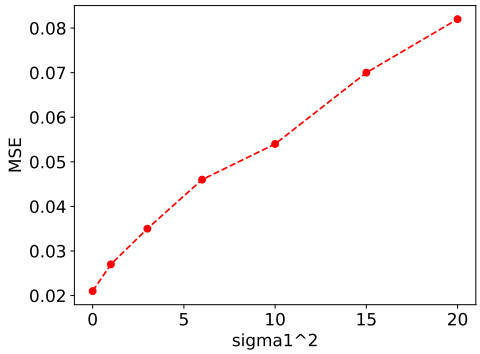
(b) Test-set MSE w.r.t. depth of KEFNN



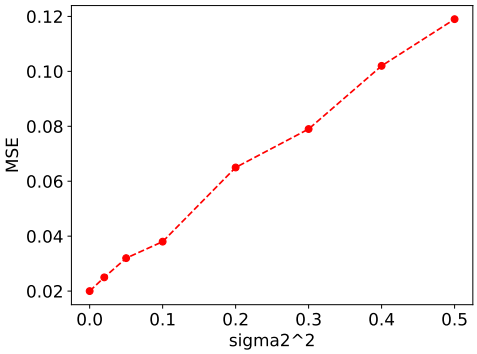
(c) Test-set MSE w.r.t. first-stage sample size



(d) Inverse of test-set MSE w.r.t. log of first-stage sample size



(e) Test-set MSE w.r.t. the variance of Gaussian noises in observations of input functions



(f) Test-set MSE w.r.t. the variance of Gaussian noises in observations of responses

Figure 2: The influence of second-stage sample size n, depth of KEFNN, first-stage sample size m, variances of Gaussian noises in observations of input functions and responses on the test-set MSE.

Table 3: The test-set MSE for different discretization schemes with varying second-stage sample sizes (n) across three cases of first-stage sample sizes (m).

	n=3965	n=3987	n=4002	n=4034	n=4042
m=1000	0.0669	0.0640	0.0665	0.0650	0.0659
m=2000	0.0519	0.0496	0.0499	0.0513	0.0496
m=3000	0.0470	0.0476	0.0475	0.0462	0.0467

In Figure 2(c), we illustrate the effect of the first-stage sample size m on generalization performance, where the second-stage sample size is fixed at $n = 500$, and the noise variances are set to $\sigma_1^2 = 3$ and $\sigma_2^2 = 0.05$. To further highlight the decay rate of the generalization error with respect to increasing m, we plot the inverse of the test-set MSE against the log of the first-stage sample size m in Figure 2(d). From this, we observe that the generalization error decreases at a rate of approximately $O((\log m)^{-1})$, which corresponds to a polynomial rate in $\log m$. This observation aligns with the learning rates established in Section 4. Such polynomial rates in $\log m$ arise from the curse of dimensionality while approximating nonlinear smooth functionals. Therefore, it is crucial to investigate the conditions under which this curse of dimensionality can be mitigated for functional data.

Next, we analyze the impact of noise in the observations of input functions and responses on generalization performance. The variation in test-set MSE loss with respect to the variances of Gaussian noise in the observations of input functions and responses is depicted in Figure 2(e) and Figure 2(f), respectively. For this simulation, we set the first-stage sample size to $m = 4000$ and the second-stage sample size to $n = 500$. In Figure 2(e), the variances of noise in responses are fixed at $\sigma_2^2 = 0$, while in Figure 2(f), the variances of noise in input functions are fixed at $\sigma_1^2 = 0$. Interestingly, in both scenarios, we observe that the generalization error increases almost linearly with the variances of Gaussian noise in the observations. This phenomenon prompts further theoretical exploration to offer a rigorous explanation. Moreover, it underscores that employing kernel embedding as a pre-smoothing technique imparts a certain level of robustness to KEFNN against observational noise in both input functions and responses.

We further investigate the impact of various discretizations on the generalization performance of KEFNN. The noise variances are fixed at $\sigma_1^2 = 3$ and $\sigma_2^2 = 0.05$, and we assess the performance under different discretizations with varying second-stage sample sizes: $n = 3965$, $n = 3987$, $n = 4002$, $n = 4034$, and $n = 4042$, across three cases of first-stage sample sizes: $m = 1000$, $m = 2000$, and $m = 3000$. The discretization points are i.i.d. sampled from a uniform distribution on $[0, 1]$. The averaged test-set MSE over five trials is presented in Table 3. The results suggest that KEFNN demonstrates a property of discretization invariance, since functional networks trained with different levels of discretization yield similar generalization performance.

Finally, we analyze how different kernel quadrature methods affect the generalization performance of KEFNN. We keep the first-stage sample size at $m = 2000$ and the noise variances fixed at $\sigma_1^2 = 3$ and $\sigma_2^2 = 0.05$. We evaluate various quadrature schemes, including Monte-Carlo methods, Quasi Monte-Carlo methods, and trapezoidal rules, across different second-stage sample sizes. In the case of Monte-Carlo methods, the discretization points

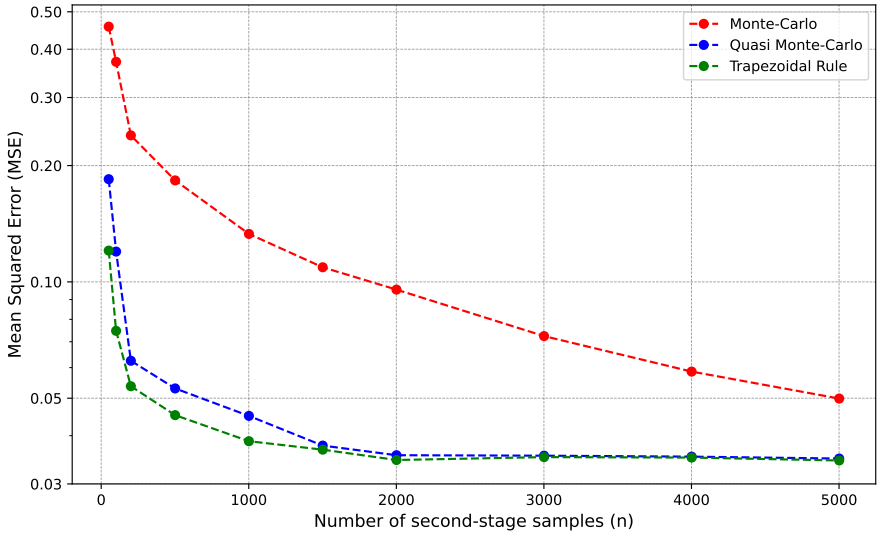


Figure 3: The test-set MSE vs. number of second-stage samples (n) in logarithmic scale.

$\{t_{i,j}\}_{j=1}^{n}$ are i.i.d. sampled from a uniform distribution on $[0,1]$. For the Quasi Monte-Carlo methods, we use low-discrepancy Sobol sequences for the discretization. The trapezoidal rule uses discretizations consisting of evenly spaced points on $[0,1]$. The weights $\{\theta_{i,j}\}_{j=1}^{n}$ of all three quadrature rules are determined by the composite trapezoidal rule. We plot the test-set MSE, averaged over five trials, against the number of second-stage samples n for these three quadrature methods in Figure 3. The results indicate that Monte-Carlo method exhibits quite slow convergence, while the Quasi Monte-Carlo method converges more rapidly, although it is still slightly less effective than the evenly spaced discretizations.

6.3 Examples on real functional datasets

In this subsection, we apply the KEFNN to several classical small real functional datasets and evaluate its performance by comparing it with baseline dimension reduction techniques, specifically "Raw data", "B-spline", and "FPCA" methods. Each of these techniques is followed by a consistent deep neural network that consists of three hidden layers, each containing 32 neurons. We have opted not to include a comparison with AdaFNN, as the limited number of training samples in these datasets can lead to overfitting issues for AdaFNN in certain cases. To assess the performance of each method, we use RMSE (root mean square error) as our evaluation metric. The dimension reduction scores for "B-spline" and "FPCA" methods are computed using functions from the Python package "scikit-fda" (Ramos-Carreño et al., 2024). Furthermore, the selection of the number of B-spline bases in "B-spline" approach, the number of eigenfunction bases in "FPCA" approach, and the hyperparameters in our "KEFNN" approach is conducted through cross-validation to ensure optimal model performance. The performance of these approaches across four learning tasks

Table 4: The test-set RMSE of the functional deep neural network employing various dimension reduction methods across four learning tasks with real functional datasets.

	Task 1	Task 2	Task 3	Task 4
Raw data+NN	0.2394 ± 0.010	0.1505 ± 0.007	0.2076 ± 0.028	0.1268 ± 0.013
B-spline+NN	0.2421 ± 0.003	0.1288 ± 0.010	0.2059 ± 0.010	0.1238 ± 0.033
FPCA+NN	0.2550 ± 0.085	0.1121 ± 0.015	0.1255 ± 0.015	0.1763 ± 0.023
KEFNN	**0.2254 ± 0.002**	**0.0614 ± 0.003**	**0.1003 ± 0.011**	**0.1163 ± 0.013**

with real functional datasets is summarized in Table 4, presenting the mean and standard deviation of the test-set RMSE based on five training runs.

In *Task 1*, we use the Medflies dataset from the Python package "scikit-fda". This dataset comprises 534 samples of Mediterranean fruit flies (Medfly), documenting the daily number of eggs laid from day 5 to day 34 for each fly. Our objective is to leverage the early trajectory of daily egg-laying (the first 20 days) to predict the overall reproduction over a 30-day period. For *Task 2* and *Task 3*, we employ the Tecator dataset, also from the "scikit-fda" package. This dataset consists of 240 meat samples, each represented by a 100-channel spectrum of absorbances along with the respective contents of moisture (water), fat, and protein. In Task 2, we aim to predict the fat content of a meat sample based on its absorbance spectrum, while in Task 3, we focus on predicting the moisture content. In *Task 4*, we work with the Moisture dataset from the R package "fds". This dataset features near-infrared reflectance spectra of 100 wheat samples, measured at 2 nm intervals from 1100 to 2500 nm, alongside their associated moisture content. Our task is to predict the moisture content of a wheat sample based on its near-infrared reflectance spectra. For all tasks, we follow a training, validation, and test split ratio of 64:16:20. In Task 1, the hyperparameters for KEFNN are set to $\gamma = 0.033$, $\beta = 0.01$, $d_1 = 100$. In the other three tasks, the hyperparameters are adjusted to $\gamma = 0.033$, $\beta = 0.1$, $d_1 = 150$.

From Table 4, it is evident that KEFNN consistently outperforms the other baseline dimension reduction methods across various real functional datasets. This result underscores the advantages and flexibility of our approach in learning various target functionals. In contrast, no single baseline method consistently outperforms the others across all learning tasks. Additionally, we observe that the standard deviation of the test-set RMSE for KEFNN is notably smaller than that of the other methods. This observation suggests that KEFNN might be less influenced by the randomness of data, leading to a more stable performance. Consequently, the generalization performance of KEFNN is likely to fall within a narrower range with higher probability, indicating enhanced reliability and consistency in its predictions.

Acknowledgments

The research leading to these results received funding from the European Research Council under the European Union's Horizon 2020 research and innovation program/ERC Advanced Grant E-DUALITY (787960). This article reflects only the authors' views, and the EU is not liable for any use that may be made of the contained information; Flemish govern-

ment (AI Research Program); Leuven.AI Institute. The research of Jun Fan is supported partially by the Research Grants Council of Hong Kong [Project No. HKBU 12302819] and [Project No. HKBU 12301619]. The work of D. X. Zhou described in this paper was fully/substantially/partially supported by the InnoHK initiative, The Government of the HKSAR, and the Laboratory for AI-Powered Financial Technologies. We also thank Dr. Zhen Zhang for his helpful discussions on this work.

Appendix A. Proof of Main Results

In this appendix, we prove the main results in Section 3 and Section 4.

A.1 Proof of main results in Section 3

A.1.1 Proof of Theorem 3

Proof. Denote Ξ_{d_1} as the space spanned by the basis $\{\phi_1, \phi_2, \ldots, \phi_{d_1}\}$, and the isometric isomorphism $U_{d_1} : (\mathbb{R}^{d_1}, \|\cdot\|_2) \to (\Xi_{d_1}, \|\cdot\|_{L_2(\mu)})$ as

$$U_{d_1}(v) = \sum_{i=1}^{d_1} v_i \phi_i, \qquad v \in \mathbb{R}^{d_1}, \tag{A.1}$$

where the measure μ is specified later in the proof.

To prove the main results of approximation rates for the target functional F, the key is the following error decomposition with three error terms to bound.

$$\begin{aligned} &\sup_{f\in\mathcal{F}} |F(f) - F_{NN}(f)| \\ \leq &\sup_{f\in\mathcal{F}} |F(f) - F(L_K f)| \\ + &\sup_{f\in\mathcal{F}} |F(L_K f) - F(U_{d_1} \circ T(L_K f))| \\ + &\sup_{f\in\mathcal{F}} |F \circ U_{d_1}(T(L_K f)) - F_{NN}(f)|, \end{aligned} \tag{A.2}$$

The first term focuses on the error of approximating the input function f by its embedded function $L_K f$, the second term focuses on the error of approximating the embedded function $L_K f$ by its projection on the subspace spanned by the eigenfunction basis, and the final term focuses on the error of approximating a λ-Hölder continuous function by a deep ReLU neural network. We then bound these three error terms individually in the following.

For the first term, with the choice of the embedding kernel K being (3.6), according to Lemma 13, $\forall f \in \mathcal{F}$, we have

$$\|f - L_K f\|_{L_2(\mathbb{R}^d)} \leq C_{d,r} \omega_{r, L_2(\mathbb{R}^d)} \left(f, \frac{\gamma}{\sqrt{2}}\right) \leq C_{d,r,\alpha} \gamma^{\alpha},$$

where $C_{d,r,\alpha} = C_{d,r}(\sqrt{2})^{-\alpha}$ is a constant only depending on d, r, α. Thus,

$$\sup_{f\in\mathcal{F}} |F(f) - F(L_K f)| \leq \sup_{f\in\mathcal{F}} C_F \|f - L_K f\|^{\lambda}_{L_2(\mathbb{R}^d)} \leq C_F C^{\lambda}_{d,r,\alpha} \gamma^{\alpha\lambda}. \tag{A.3}$$

For the second term, since $\forall f \in \mathcal{F}$, $L_K f \in \mathcal{H}_\gamma$, we can write $L_K f = \sum_{i=1}^{\infty} a_i \sqrt{\lambda_i} \phi_i$, with $\|L_K f\|_{H_\gamma} = \sqrt{\sum_{i=1}^{\infty} a_i^2} \leq (2^r - 1)\pi^{-\frac{d}{4}} \gamma^{-\frac{d}{2}}$ by Lemma 13. Then we have

$$T(L_K f) = \left[a_1\sqrt{\lambda_1}, a_2\sqrt{\lambda_2}, \ldots, a_{d_1}\sqrt{\lambda_{d_1}}\right],$$

and hence

$$U_{d_1} \circ T(L_K f) = \sum_{i=1}^{d_1} a_i \sqrt{\lambda_i} \phi_i.$$

It follows that

$$\begin{aligned} \|L_K f - U_{d_1} \circ T(L_K f)\|_{L_2(\mu)}^2 &= \left\| \sum_{i=d_1+1}^{\infty} a_i \sqrt{\lambda_i} \phi_i \right\|_{L_2(\mu)}^2 \\ = \sum_{i=d_1+1}^{\infty} a_i^2 \lambda_i &\leq \lambda_{d_1+1} \sum_{i=d_1+1}^{\infty} a_i^2 \leq \lambda_{d_1+1} \|L_K f\|_{H_\gamma}^2. \end{aligned}$$

The eigenvalues of the integral operator associated with a one-dimensional Gaussian kernel can be expressed with an explicit formulation that exhibits exponential decay (Fasshauer, 2011, pp. 28), specifically

$$\lambda_i = \frac{\beta}{\gamma^{2i} \left(\frac{\beta^2}{2} \left(1 + \sqrt{1 + \left(\frac{2}{\beta\gamma} \right)^2} \right) + \frac{1}{\gamma^2} \right)^{i+\frac{1}{2}}} \leq \frac{\beta\gamma}{\left(1 + \frac{\beta\gamma}{2} \right)^{2i+1}},$$

where the measure μ is selected as a Gaussian distribution, characterized by the density function $u(x) = \frac{\beta}{\sqrt{\pi}} \exp^{-\beta^2 x^2}$, and β serves as a global scaling parameter. We use this explicit form because the bandwidth γ is a hyperparameter that requires careful tuning later and should not be treated as a constant.

Notice that the $\binom{j+d}{d}$-th eigenvalue of the integral operator associated with a multi-dimensional Gaussian kernel k_γ can be bounded by $\frac{(\beta\gamma)^d}{\left(1+\frac{\beta\gamma}{2}\right)^{2j+d}}$ (Rasmussen and Williams, 2006, pp. 98). If we denote $\binom{j+d}{d} \leq \frac{(j+\frac{d}{2})^d}{d!} =: d_1$, then the d_1-th eigenvalue of the integral operator corresponding to k_γ can be bounded by

$$\lambda_{d_1} \leq \frac{(\beta\gamma)^d}{\left(1 + \frac{\beta\gamma}{2} \right)^{2c_2 d_1^{\frac{1}{d}}}}, \tag{A.4}$$

where $c_2 = (d!)^{\frac{1}{d}} > \frac{d}{e}$ by Stirling's theorem. By choosing $\beta = 2$, we get

$$\begin{aligned} \sup_{f \in \mathcal{F}} |F(L_K f) - F(U_{d_1} \circ T(L_K f))| &\leq \sup_{f \in \mathcal{F}} C_F \|L_K f - U_{d_1} \circ T(L_K f)\|_{L_2(\mu)}^{\lambda} \\ &\leq C_F C_{d,r,\lambda} \left(1+\gamma\right)^{-c_2 \lambda d_1^{\frac{1}{d}}}, \end{aligned} \tag{A.5}$$

where $C_{d,r,\lambda}$ is a constant depending only on d, r, λ.

For the third term, since $F \circ U_{d_1}$ follows the smoothness of F, $\forall y_1, y_2 \in \mathbb{R}^{d_1}$, we have

$$\begin{aligned} |F \circ U_{d_1}(y_1) - F \circ U_{d_1}(y_2)| &\leq C_F \|U_{d_1}(y_1) - U_{d_1}(y_2)\|_{L_2(\mu)}^{\lambda} \\ &= C_F \|y_1 - y_2\|_2^{\lambda}. \end{aligned} \tag{A.6}$$

Therefore, $F \circ U_{d_1}$ is essentially a λ-Hölder continuous function defined on $[-R, R]^{d_1}$, where $R = \|T(L_K f)\|_\infty$. Then by Lemma 14, there exists a deep ReLU neural network H_{NN} with depth $d_1^2 + d_1 + 1$, and M nonzero parameters such that

$$\sup_{y \in [-R,R]^{d_1}} |F \circ U_{d_1}(y) - H_{NN}(y)| \leq 2C_F d_1 \left(\frac{c_0 d_1^{\frac{4}{d_1}} R}{M^{\frac{1}{d_1}}} \right)^\lambda,$$

where c_0 is a constant independent of d_1, M.

Recalling the architecture of our functional network as defined in Definition 1, it can be alternatively represented as $F_{NN}(f) = H_{NN} \circ T(L_K f)$, where H_{NN} denotes a standard deep ReLU neural network with an input dimension d_1. Moreover, according to Lemma 13 and the fact that $C_K = \tilde{c}_0 \gamma^{-d}$ with $\tilde{c}_0 = \pi^{-\frac{d}{2}} \sum_{j=1}^r \binom{r}{j} \frac{1}{j^d}$ being a positive constant, the radius of the input cube can be bounded by

$$\begin{aligned} R = \|T(L_K f)\|_\infty &\leq \|L_K f\|_{L_2(\mu)} \leq \sqrt{\mu(\mathbb{R}^d)} \|L_K f\|_\infty \\ &\leq \sqrt{\mu(\mathbb{R}^d) C_K} \|L_K f\|_{H_\gamma} \leq \sqrt{\tilde{c}_0 \mu(\mathbb{R}^d)} (2^r - 1) \pi^{-\frac{d}{4}} \gamma^{-d}, \end{aligned}$$

Hence, we conclude that there exists a functional net F_{NN} following the architecture in Definition 1 with depth $J = d_1^2 + d_1 + 2$ and M nonzero parameters such that

$$\sup_{f \in \mathcal{F}} |F \circ U_{d_1}(T(L_K f)) - F_{NN}(f)| \leq 2C_F d_1 \left(\frac{c_1 d_1^{\frac{4}{d_1}} \gamma^{-d}}{M^{\frac{1}{d_1}}} \right)^\lambda, \tag{A.7}$$

where $c_1 = c_0 \sqrt{\tilde{c}_0 \mu(\mathbb{R}^d)} (2^r - 1) \pi^{-\frac{d}{4}}$ is a constant independent of d_1, M.

Finally, by combining (A.3), (A.5), and (A.7), we have

$$\sup_{f \in \mathcal{F}} |F(f) - F_{NN}(f)| \leq C_F C_{d,r,\alpha}^\lambda \gamma^{\alpha\lambda} + C_F C_{d,r,\lambda} \left(1 + \gamma\right)^{-c_2 \lambda d_1^{\frac{1}{d}}} + 2C_F d_1 \left(\frac{c_1 d_1^{\frac{4}{d_1}} \gamma^{-d}}{M^{\frac{1}{d_1}}} \right)^\lambda. \tag{A.8}$$

To balance these three error terms, we can choose

$$d_1 = \tilde{c}_1 \frac{\log M}{\log \log M}, \quad \gamma = \tilde{c}_2 \left(\frac{\log \log M}{\log M} \right)^{\frac{1}{d}} \log \log M, \tag{A.9}$$

where $\tilde{c}_1, \tilde{c}_2$ are positive constants that will be determined later. Thus, considering the first error term, we have

$$\gamma^{\alpha\lambda} = \tilde{c}_2^{\alpha\lambda} \left(\log M\right)^{-\frac{\alpha\lambda}{d}} \left(\log \log M\right)^{\left(\frac{1}{d}+1\right)\alpha\lambda}, \tag{A.10}$$

since $(1 + \frac{1}{x})^x \geq c_3$ for some constant $c_3 \in (1, e)$ when x is sufficiently large, we can set $x = \frac{1}{\gamma}$. When M is large enough, considering the second error term, we have

$$\left(1 + \gamma\right)^{-c_2 \lambda d_1^{\frac{1}{d}}} = \left[(1 + \frac{1}{x})^x \right]^{-c_2 \tilde{c}_1^{\frac{1}{d}} \tilde{c}_2 \lambda \log \log M} \leq \left(\log M\right)^{-c_2 \tilde{c}_1^{\frac{1}{d}} \tilde{c}_2 \lambda \log c_3}. \tag{A.11}$$

Finally, since $d_1^{\frac{4}{d_1}} \leq c_4$ for some constant c_4, and given that $M^{\frac{1}{d_1}} = (\log M)^{\frac{1}{\tilde{c}_1}}$. Considering the third error term, we have

$$d_1 \left(\frac{c_1 d_1^{\frac{4}{d_1}} \gamma^{-d}}{M^{\frac{1}{d_1}}} \right)^{\lambda} \leq c_1^{\lambda} c_4^{\lambda} \tilde{c}_1 \tilde{c}_2^{-d} (\log M)^{\frac{3\lambda}{2} + 1 - \frac{\lambda}{\tilde{c}_1}} (\log \log M)^{-1-(d+1)\lambda}. \tag{A.12}$$

Therefore, to balance the error terms (A.10), (A.11), and (A.12) w.r.t. the exponential rates on $\log M$, we can choose the constants to satisfy

$$\tilde{c}_1 < \frac{1}{1 + \frac{\alpha}{d} + \frac{1}{\lambda}}, \qquad \tilde{c}_2 > \frac{\alpha}{d c_2 \tilde{c}_1^{\frac{1}{d}} \log c_3}. \tag{A.13}$$

Thus we obtain the desired bounds on the approximation error and the depth of the functional network, with the constants specified as $\tilde{c}_3 = 2\tilde{c}_1^2 + 2$, and $\tilde{c}_4 = C_F C_{d,r,\alpha}^{\lambda} \tilde{c}_2^{\alpha\lambda} + C_F C_{d,r,\lambda} + 2 C_F c_1^{\lambda} c_4^{\lambda} \tilde{c}_1 \tilde{c}_2^{-d\lambda}$. ■

A.1.2 Proof of Theorem 4

Proof. The only difference between the functional network architecture used in the proof here and that presented in Theorem 3 lies in the kernel embedding step, where the domain of the input function is changed from $\mathbb{R}^d$ in Theorem 3 to Ω considered here. By denoting

$$L_K f(x) = \int_{\Omega} K(x,t) f(t) dt, \quad x \in \Omega,$$

and $\tilde{f}$ as the extension by zero of f to $\mathbb{R}^d$. We have

$$L_K \tilde{f}(x) = \int_{\mathbb{R}^d} K(x,t) \tilde{f}(t) dt, \quad x \in \mathbb{R}^d.$$

Notice that $L_K f = (L_K \tilde{f})|_{\Omega}$, it follows that

$$\|f - L_K f\|_{L_2(\mu)} \leq \left\| \tilde{f} - L_K \tilde{f} \right\|_{L_2(\tilde{\mu})},$$

where we set $\mu = \tilde{\mu}|_{\Omega}$, and $\tilde{\mu}$ is chosen as the same Gaussian distribution in the proof of Theorem 3. The rest of the proof follows directly from the result in Theorem 3 applied to approximate nonlinear functionals defined on $\tilde{f}$. ■

A.1.3 Proof of Theorem 5

Proof. In this case, the domain of the target functional is already a Gaussian RKHS. As a result, we can use the error decomposition with only the last two error terms considered in Theorem 3, with a focus on balancing the choice of d_1. Specifically, by denote the mapping $V_{d_1} : (\mathbb{R}^{d_1}, |\cdot|) \to (\Xi_{d_1}, \|\cdot\|_{L_2(\mu)})$ as

$$V_{d_1}(v) = \sum_{i=1}^{d_1} \frac{v_i}{\lambda_i} \phi_i, \qquad v \in \mathbb{R}^{d_1}. \tag{A.14}$$

Now the error decomposition contains only two error terms,

$$\begin{aligned}&\sup_{f\in\mathcal{F}}|F(f)-F_{NN}(f)|\\ \leq&\sup_{f\in\mathcal{F}}|F(f)-F(V_{d_1}\circ T(L_Kf))|\\ +&\sup_{f\in\mathcal{F}}|F\circ V_{d_1}(T(L_Kf))-F_{NN}(f)|.\end{aligned}\tag{A.15}$$

Let us consider the first error term. For any $f\in\mathcal{H}_\gamma$, denote $f=\sum_{i=1}^{\infty}a_i\sqrt{\lambda_i}\phi_i$, with $\|f\|_{H_\gamma}=\sqrt{\sum_{i=1}^{\infty}a_i^2}\leq 1$. Since we choose the embedding kernel K as defined in (3.14), the projection kernel K_0 as k_γ, and the projection bases as the first d_1 eigenfunctions of the integral operator $L_{K_0}^{\mu}$, we have

$$T(L_Kf)=\Big[a_1\lambda_1^{\frac{3}{2}},a_2\lambda_2^{\frac{3}{2}},\ldots,a_{d_1}\lambda_{d_1}^{\frac{3}{2}}\Big],$$

$$V_{d_1}\circ T(L_Kf)=\sum_{i=1}^{d_1}a_i\sqrt{\lambda_i}\phi_i.$$

It follows that

$$\begin{aligned}&\|f-V_{d_1}\circ T(L_Kf)\|_{L_2(\mu)}^2=\Big\|\sum_{i=d_1+1}^{\infty}a_i\sqrt{\lambda_i}\phi_i\Big\|_{L_2(\mu)}^2\\ =&\sum_{i=d_1+1}^{\infty}a_i^2\lambda_i\leq\lambda_{d_1+1}\sum_{i=d_1+1}^{\infty}a_i^2\leq\lambda_{d_1+1}\|f\|_{H_\gamma}^2.\end{aligned}$$

Therefore, by utilizing the eigenvalue upper bound of the Gaussian kernel as specified in (A.4), the first error term is bounded by

$$\begin{aligned}\sup_{f\in\mathcal{F}}|F(f)-F(V_{d_1}\circ T(L_Kf))|&\leq\sup_{f\in\mathcal{F}}C_F\|f-V_{d_1}\circ T(L_Kf)\|_{L_2(\mu)}^{\lambda}\\&\leq C_FC_{d,\gamma,\lambda}e^{-c_2\lambda d_1^{\frac{1}{d}}},\end{aligned}\tag{A.16}$$

where $C_{d,\gamma,\lambda}$ is a constant depending only on d,γ,λ, and c_2 is a constant with $c_2>\frac{d}{e}$.

As for the second error term, notice that $F\circ V_{d_1}$ follows the smoothness of F with a scaling on the constant term, that is

$$\begin{aligned}|F\circ V_{d_1}(y_1)-F\circ V_{d_1}(y_2)|&\leq C_F\|V_{d_1}(y_1)-V_{d_1}(y_2)\|_{L_2(\mu)}^{\lambda}\\&\leq\frac{C_F}{\lambda_{d_1}^{\lambda}}\|y_1-y_2\|_2^{\lambda}.\end{aligned}\tag{A.17}$$

Furthermore, the radius of the input for the deep ReLU neural network can be bounded by

$$\begin{aligned}R=\|T(L_Kf)\|_\infty&\leq\|L_Kf\|_{L_2(\mu)}\leq\sqrt{\mu(\Omega)}\|L_Kf\|_{L_\infty(\Omega)}\\&=\sqrt{\mu(\Omega)}\sup_{x\in\Omega}\int_\Omega K(x,t)f(t)dt\leq\sqrt{\mu(\Omega)}C_K\|f\|_{L_2(\Omega)}\leq\sqrt{\mu(\Omega)}C_K,\end{aligned}$$

Therefore, similar to the proof of Theorem 3, we can apply Lemma 14 to establish the existence of a functional network $F_{NN} = H_{NN} \circ T(L_K f)$ in the format defined by Definition 1, with a depth of $J = d_1^2 + d_1 + 2$ and M nonzero parameters, such that

$$\begin{aligned} \sup_{f\in\mathcal{F}} |F \circ V_{d_1}(T(L_K f)) - F_{NN}(f)| &\leq 2C_F \frac{d_1}{\lambda_{d_1}^{\lambda}} \left(\frac{c_0 R d_1^{\frac{4}{d_1}}}{M^{\frac{1}{d_1}}} \right)^{\lambda} \\ &\leq c_5 d_1 e^{2c_2 \lambda d_1^{\frac{1}{d}}} \left(\frac{d_1^{\frac{4}{d_1}}}{M^{\frac{1}{d_1}}} \right)^{\lambda}, \end{aligned} \tag{A.18}$$

where for the second inequality we use the eigenvalue lower bound of the Gaussian kernel, and $c_5 = 2C_F c_0^{\lambda} \mu(\Omega)^{\frac{\lambda}{2}} (\frac{\pi\gamma^2}{2})^{\frac{d\lambda}{4}}$ is a positive constant.

Finally, by combining (A.16) and (A.18), we have

$$\sup_{f\in\mathcal{F}} |F(f) - F_{NN}(f)| \leq C_F C_{d,\gamma,\lambda} e^{-c_2 \lambda d_1^{\frac{1}{d}}} + c_5 d_1 e^{2c_2 \lambda d_1^{\frac{1}{d}}} \left(\frac{d_1^{\frac{4}{d_1}}}{M^{\frac{1}{d_1}}} \right)^{\lambda}. \tag{A.19}$$

Furthermore, by choosing

$$d_1 = c_6 \left(\log M\right)^{\frac{d}{d+1}}, \tag{A.20}$$

where c_6 is a constant to be determined later. Considering the first error term, we have

$$e^{-c_2 \lambda d_1^{\frac{1}{d}}} = e^{-c_2 c_6^{\frac{1}{d}} \lambda (\log M)^{\frac{1}{d+1}}}. \tag{A.21}$$

Considering the second error term, we have

$$d_1 e^{2c_2 \lambda d_1^{\frac{1}{d}}} \left(\frac{d_1^{\frac{4}{d_1}}}{M^{\frac{1}{d_1}}} \right)^{\lambda} \leq c_6 c_4^{\lambda} e^{\left(2c_2 c_6^{\frac{1}{d}} - \frac{1}{c_6}\right) \lambda (\log M)^{\frac{1}{d+1}}} \left(\log M\right)^{\frac{d}{d+1}}. \tag{A.22}$$

Therefore, we can choose $c_6 = (3c_2)^{-\frac{d}{d+1}}$ to balance the dominated terms specified in (A.21) and (A.22). It follows that

$$\sup_{f\in\mathcal{F}} |F(f) - F_{NN}(f)| \leq \tilde{c}_6 e^{-c_7 \lambda (\log M)^{\frac{1}{d+1}}} \left(\log M\right)^{\frac{d}{d+1}}, \tag{A.23}$$

where $\tilde{c}_6 = C_F C_{d,\gamma,\lambda} + c_5 c_6 c_4^{\lambda}$, and

$$c_7 = 3^{-\frac{1}{d+1}} c_2^{\frac{d}{d+1}} > 3^{-\frac{1}{d+1}} \left(\frac{d}{e}\right)^{\frac{d}{d+1}} > \left(\frac{d}{3e}\right)^{\frac{d}{d+1}}. \tag{A.24}$$

Thus we complete the proof, with the constant $\tilde{c}_5 = 2c_6^2 + 2$. ■

A.1.4 Proof of Theorem 6

Proof. The proof essentially follows the framework of the proof for Theorem 5, with the only difference being the rates of eigenvalue decay for the integral operator $L^{\mu}_{K_\alpha}$. The upper bounds for the eigenvalues were established in (Bach, 2017, pp. 30), and we can obtain the lower bounds using the same approach outlined there. Consequently, we have

$$C_1(\log k)^{2\alpha}k^{-2\alpha} \leq \lambda_k \leq C_2(\log k)^{2\alpha(d-1)}k^{-2\alpha}, \tag{A.25}$$

where C_1, C_2 are positive constants. According to the same error decomposition in the proof of Theorem 5, there exists a functional net F_{NN} in the form of Definition 1 with depth $J = d_1^2 + d_1 + 2$ and number of nonzero parameters M such that

$$\sup_{f\in\mathcal{F}} |F(f) - F_{NN}(f)| \leq C_F C_2^{\frac{\lambda}{2}} (\log d_1)^{\alpha\lambda(d-1)} {d_1}^{-\alpha\lambda} + C_3 d_1^{2\alpha\lambda+1} (\log d_1)^{-2\alpha\lambda} \left(\frac{d_1^{\frac{4}{d_1}}}{M^{\frac{1}{d_1}}} \right)^{\lambda}, \tag{A.26}$$

where C_3 is a positive constant independent of d_1, M. Finally, by choosing

$$d_1 = C_4 \frac{\log M}{\log\log M}, \tag{A.27}$$

where C_4 is a constant to be determined later. Considering the first error term, we have

$$(\log d_1)^{\alpha\lambda(d-1)} {d_1}^{-\alpha\lambda} \leq C_4^{-\alpha\lambda} (\log M)^{-\alpha\lambda} (\log\log M)^{\alpha\lambda(d-2)}. \tag{A.28}$$

Considering the second error term, we have

$$d_1^{2\alpha\lambda+1} \left(\frac{d_1^{\frac{4}{d_1}}}{M^{\frac{1}{d_1}}} \right)^{\lambda} \leq c_4^{\lambda} C_4^{2\alpha\lambda+1} (\log M)^{2\alpha\lambda+1-\frac{\lambda}{C_4}} (\log\log M)^{-2\alpha\lambda-1}. \tag{A.29}$$

Therefore, we can choose $C_4 \leq \frac{\lambda}{3\alpha\lambda+1}$ to balance the dominated $\log M$ terms in (A.28) and (A.29). Then we can obtain the desired bounds on the depth of functional network and the approximation error, with the constants specified as $C_5 = 2C_4^2 + 2$, and $C_6 = C_F C_2^{\frac{\lambda}{2}} C_4^{-\alpha\lambda} + C_3 c_4^{\lambda} C_4^{2\alpha\lambda+1}$. ∎

A.2 Proof of main results in Section 4

A.2.1 Proof of Proposition 7

Proof.

$$\begin{aligned} |h(x) - h(y)| &\leq \left| \sum_{i=1}^{(N+1)^{d_1}} c_i \psi\left(\frac{N}{2R}(x - b_i)\right) - \sum_{i=1}^{(N+1)^{d_1}} c_i \psi\left(\frac{N}{2R}(y - b_i)\right) \right| \\ &\leq \sum_{i=1}^{(N+1)^{d_1}} L \left| \psi\left(\frac{N}{2R}(x - b_i)\right) - \psi\left(\frac{N}{2R}(y - b_i)\right) \right|. \end{aligned}$$

According to Lemma 15, we have

$$\begin{aligned}|\psi(x)-\psi(y)| &\le |\min\big\{\min_{k\neq j}(1+x_k-x_j),\min_k(1+x_k),\min_k(1-x_k)\big\}\\ &\quad-\min\big\{\min_{k\neq j}(1+y_k-y_j),\min_k(1+y_k),\min_k(1-y_k)\big\}|\\ &\le 2(d_1^2+d_1)||x-y||_1.\end{aligned}$$

Therefore we have

$$\begin{aligned}|h(x)-h(y)| &\le \sum_{i=1}^{(N+1)^{d_1}} 2(d_1^2+d_1)L\left|\left|\frac{N}{2R}(x-y)\right|\right|_1\\ &\le \frac{L}{R}N(N+1)^{d_1}(d_1^2+d_1)||x-y||_1,\end{aligned}$$

which completes the proof by replacing R with $\sqrt{C_\mu}C_K$. ∎

A.2.2 Proof of Proposition 8

Proof. Note that

$$\begin{aligned}\mathcal{E}\left(\pi_L F_{\widehat{D}}\right)-\mathcal{E}\left(F_\rho\right) &= I_1+I_2+2\left\{\mathcal{E}_D\left(\pi_L F_{\widehat{D}}\right)-\mathcal{E}_D\left(\pi_L F_D\right)\right\}\\ &\le I_1+I_2+2\left\{\mathcal{E}_D\left(\pi_L F_{\widehat{D}}\right)-\mathcal{E}_D\left(\pi_L F_D\right)\right\}\\ &\quad+2\left\{\mathcal{E}_{\widehat{D}}\left(\pi_L H_D\circ\widehat{L_K}\right)-\mathcal{E}_{\widehat{D}}\left(\pi_L H_{\widehat{D}}\circ\widehat{L_K}\right)\right\}.\end{aligned}$$

The inequality from above holds since

$$\mathcal{E}_{\widehat{D}}\left(\pi_L H_D\circ\widehat{L_K}\right)\ge\mathcal{E}_{\widehat{D}}\left(H_{\widehat{D}}\circ\widehat{L_K}\right)\ge\mathcal{E}_{\widehat{D}}\left(\pi_L H_{\widehat{D}}\circ\widehat{L_K}\right).$$

The desired result is achieved by noting

$$\begin{aligned}&2\left\{\mathcal{E}_D\left(\pi_L F_{\widehat{D}}\right)-\mathcal{E}_D\left(\pi_L F_D\right)\right\}+2\left\{\mathcal{E}_{\widehat{D}}\left(\pi_L H_D\circ\widehat{L_K}\right)-\mathcal{E}_{\widehat{D}}\left(\pi_L H_{\widehat{D}}\circ\widehat{L_K}\right)\right\}\\ &\le 2\left|\frac{1}{m}\sum_{i=1}^m\left(\pi_L H_{\widehat{D}}(u_i)-y_i\right)^2-\frac{1}{m}\sum_{i=1}^m\left(\pi_L H_{\widehat{D}}(\hat{u}_i)-y_i\right)^2\right|\\ &+2\left|\frac{1}{m}\sum_{i=1}^m\left(\pi_L H_D(u_i)-y_i\right)^2-\frac{1}{m}\sum_{i=1}^m\left(\pi_L H_D(\hat{u}_i)-y_i\right)^2\right|\\ &\le 16L\sup_{H\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^m|\pi_L H(u_i)-\pi_L H(\hat{u}_i)|\\ &\le 16L\sup_{H\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^m|H(u_i)-H(\hat{u}_i)|=I_3,\end{aligned}$$

where the last inequality above is from $|\pi_L H(u_i)-\pi_L H(\hat{u}_i)|\le|H(u_i)-H(\hat{u}_i)|$. ∎

A.2.3 Proof of Theorem 9

In the proof, we will use the covering number as the tool of the complexity measure of a hypothesis space $\mathcal{H}$. Let $f_1^m = (f_1, \ldots, f_m)$ be m fixed functions in the input function space $\mathcal{F}$. Let ν_m be the corresponding empirical measure, i.e.,

$$\nu_m(A) = \frac{1}{m}\sum_{i=1}^{m} I_A(f_i), \qquad A \subseteq \mathcal{F}. \tag{A.30}$$

Then

$$\|F\|_{L_p(\nu_m)} = \left\{\frac{1}{m}\sum_{i=1}^{m}|F(f_i)|^p\right\}^{\frac{1}{p}}, \tag{A.31}$$

and any ϵ-cover of $\mathcal{H}$ w.r.t. $\|\cdot\|_{L_p(\nu_m)}$ is called a L_p ϵ-cover of $\mathcal{H}$ on f_1^m, the ϵ-covering number of $\mathcal{H}$ w.r.t. $\|\cdot\|_{L_p(\nu_m)}$ is denoted by

$$\mathcal{N}_p(\epsilon, \mathcal{H}, f_1^m),$$

which is the minimal integer N such that there exist functionals $F_1, \ldots, F_N : \mathcal{F} \to \mathbb{R}$ with the property that for every $F \in \mathcal{F}$, there is a $j = j(F) \in \{1, \ldots, N\}$ such that

$$\left\{\frac{1}{m}\sum_{i=1}^{m}|F(f_i) - F_j(f_i)|^p\right\}^{\frac{1}{p}} < \epsilon.$$

Moreover, we denote $\mathcal{M}_p(\epsilon, \mathcal{H}, f_1^m)$ as the ϵ-packing number of $\mathcal{H}$ w.r.t. $\|\cdot\|_{L_p(\nu_m)}$, which is the largest integer N such that there exist functionals $F_1, \ldots, F_N : \mathcal{F} \to \mathbb{R}$ satisfying $\|F_j - F_k\|_{L_p(\nu_m)} \geq \epsilon$ for all $1 \leq j < k \leq N$.

Proof. From Proposition 8, we know it suffices to bound the expectation of I_1, I_2 and I_3 separately.

- First, we derive a bound for I_1 in a probability form. Denote $\pi_L\mathcal{H} \circ L_K = \{\pi_L H \circ L_K : H \in \mathcal{H}\}$. For any $\epsilon > 0$,

$$\begin{aligned}
&P\left\{I_1 > \epsilon\right\} \\
=&P\left\{||\pi_L F_{\widehat{D}} - F_\rho||_\rho^2 - \left(\mathcal{E}_D(\pi_L F_{\widehat{D}}) - \mathcal{E}_D(F_\rho)\right) > \frac{1}{2}\left(\epsilon + ||\pi_L F_{\widehat{D}} - F_\rho||_\rho^2\right)\right\} \\
\leq&P\Bigg\{\exists F \in \pi_L\mathcal{H} \circ L_K : ||F - F_\rho||_\rho^2 - \left(\mathcal{E}_D(F) - \mathcal{E}_D(F_\rho)\right) \\
&\quad > \frac{1}{2}\left(\frac{\epsilon}{2} + \frac{\epsilon}{2} + ||F - F_\rho||_\rho^2\right)\Bigg\} \\
\leq&14\sup_{f_1^m}\mathcal{N}_1\left(\frac{\epsilon}{80L}, \pi_L\mathcal{H} \circ L_K, f_1^m\right)\exp\left(-\frac{m\epsilon}{5136L^4}\right),
\end{aligned}$$

where we have used Lemma 16 in deriving the last inequality with $\alpha = \beta = \frac{\epsilon}{2}$, and $\delta = \frac{1}{2}$. Therefore, for any $a \geq \frac{1}{m}$,

$$\begin{aligned} EI_1 &\leq \int_0^\infty P\{I_1 > u\}du \leq a + \int_a^\infty P\{I_1 > u\}du \\ &\leq a + \int_a^\infty 14 \sup_{f_1^m} \mathcal{N}_1\left(\frac{1}{80Lm}, \pi_L\mathcal{H} \circ L_K, f_1^m\right) \exp\left(-\frac{mu}{5136L^4}\right) du \\ &\leq a + 14 \sup_{f_1^m} \mathcal{N}_1\left(\frac{1}{80Lm}, \pi_L\mathcal{H} \circ L_K, f_1^m\right) \frac{5136L^4}{m} \exp\left(-\frac{ma}{5136L^4}\right), \end{aligned}$$

which is minimized if we choose

$$a = \frac{5136L^4}{m} \log\left(14 \sup_{f_1^m} \mathcal{N}_1\left(\frac{1}{80Lm}, \pi_L\mathcal{H} \circ L_K, f_1^m\right)\right),$$

therefore, we have

$$\begin{aligned} EI_1 &\leq \frac{5136L^4}{m}\left\{\log\left(14 \sup_{f_1^m} \mathcal{N}_1\left(\frac{1}{80Lm}, \pi_L\mathcal{H} \circ L_K, f_1^m\right)\right) + 1\right\} \\ &\leq \frac{5136L^4}{m}\left\{\log(28) + 2c_2' JM\log(M)\log(320eL^2m) + 1\right\} \\ &\leq \frac{c_1' JM\log M\log m}{m}, \end{aligned} \tag{A.32}$$

where the second inequality above is from Lemma 17, and c_1' is a constant.

- Second, we estimate EI_2. Since $\mathcal{E}_D(\pi_L F_D) \leq \mathcal{E}_D(F_D)$, we have

$$\begin{aligned} EI_2 &\leq 2E\left\{\mathcal{E}_D(F_D) - \mathcal{E}_D(F_\rho)\right\} \\ &= 2E\left\{\inf_{F \in \{H \circ L_K : H \in \mathcal{H}\}} \mathcal{E}_D(F) - \mathcal{E}_D(F_\rho)\right\} \\ &\leq 2 \inf_{F \in \{H \circ L_K : H \in \mathcal{H}\}} E\left\{\mathcal{E}_D(F) - \mathcal{E}_D(F_\rho)\right\} \\ &= 2 \inf_{F \in \{H \circ L_K : H \in \mathcal{H}\}} \|F - F_\rho\|_\rho^2. \end{aligned} \tag{A.33}$$

- Third, we estimate EI_3. Note that

$$\begin{aligned} EI_3 &= E\left\{16L \sup_{H \in \mathcal{H}} \frac{1}{m}\sum_{i=1}^m |H(u_i) - H(\hat{u}_i)|\right\} \\ &\leq E\left\{16L \frac{1}{m}\sum_{i=1}^m \sup_{H \in \mathcal{H}} |H(u_i) - H(\hat{u}_i)|\right\} \\ &\leq 16L \cdot E\left\{\sup_{H \in \mathcal{H}} |H(u_1) - H(\hat{u}_1)|\right\}. \end{aligned}$$

According to Proposition 7, we have

$$\sup_{H\in\mathcal{H}}|H(u_1)-H(\hat{u}_1)|\leq\frac{L}{\sqrt{C_\mu}C_K}N(N+1)^{d_1}(d_1^2+d_1)||Tu_1-T\hat{u}_1||_1,$$

Note that

$$\begin{aligned}||Tu_1-T\hat{u}_1||_1&=\sum_{i=1}^{d_1}\left|\int u_1(t)\phi_i(t)d\mu(t)-\int\hat{u}_1(t)\phi_i(t)d\mu(t)\right|\\&\leq d_1||u_1-\hat{u}_1||_{L_2(\mu)}.\end{aligned}$$

Therefore, we have

$$\begin{aligned}EI_3&\leq\frac{16C_\mu L^2}{C_K}N(N+1)^{d_1}(d_1^3+d_1^2)E\left(||u_1-\hat{u}_1||_{L_2(\mu)}\right)\\&\leq\frac{c_5'}{C_K}M^{1+\frac{1}{d_1}}E\left(||u_1-\hat{u}_1||_{L_2(\mu)}\right),\end{aligned}$$

where the last inequality is from (4.6), and $c_5'=\frac{32L^2}{\sqrt{C_\mu}\bar{C}_1}$ is a positive constant. Since K and K_0 are chosen as the cases stated in Section 3, where we have that $K(\cdot,t)\in\mathcal{H}_{K_0}$ for any $t\in\Omega$. Therefore, by Lemma 18 we get

$$E\left(||u_1-\hat{u}_1||_{L_2(\mu)}\right)\leq\sqrt{C_\mu C_{K_0}}E\left(||u_1-\hat{u}_1||_{K_0}\right)\leq c_6'\sqrt{\lambda_q},\tag{A.34}$$

where $q=\frac{c_5'n}{\log n}$, and c_5', $c_6'=c_4'\sqrt{C_\mu C_{K_0}}$ are positive constants. Therefore, it follows that

$$EI_3\leq\frac{c_3'}{C_K}M^{1+\frac{1}{d_1}}\sqrt{\lambda_q},\tag{A.35}$$

where $c_3'=c_5'c_6'$.

Thus the proof is completed by combining (A.32), (A.33), and (A.35). ■

A.2.4 Proof of Theorem 10

Proof. By Theorem 4, there exists a functional network F_{NN} with M nonzero parameters, first hidden layer width $d_1=\tilde{c}_1\frac{\log M}{\log\log M}$, depth $J\leq\tilde{c}_3\left(\frac{\log M}{\log\log M}\right)^2$, and embedding kernel being (3.6) with $\gamma=\tilde{c}_2\left(\frac{\log\log M}{\log M}\right)^{\frac{1}{d}}\log\log M\leq 1$, such that

$$\sup_{f\in\mathcal{F}}|F_{NN}(f)-F_\rho(f)|\leq\tilde{c}_4\left(\log M\right)^{-\frac{\alpha\lambda}{d}}\left(\log\log M\right)^{\left(\frac{1}{d}+1\right)\alpha\lambda},$$

then we have

$$\begin{aligned}\inf_{F\in\{H\circ L_K:H\in\mathcal{H}\}}\|F-F_\rho\|_\rho^2\leq||F_{NN}-F_\rho||_\rho^2&\leq\sup_{f\in\mathcal{F}}|F_{NN}(f)-F_\rho(f)|^2\\&\leq\tilde{c}_4^2\left(\log M\right)^{-\frac{2\alpha\lambda}{d}}\left(\log\log M\right)^{2\left(\frac{1}{d}+1\right)\alpha\lambda}.\end{aligned}\tag{A.36}$$

Plugging it into Theorem 9, since $M^{\frac{1}{d_1}} = (\log M)^{\frac{1}{\tilde{c}_1}}$, $\lambda_q \leq \hat{C}_1 e^{-2c_2 q^{\frac{1}{d}}}$, and $C_K = \tilde{c}_0 \gamma^{-d}$ with $\tilde{c}_0$ being a positive constant, we get

$$\begin{aligned} E\,||\pi_L F_{\widehat{D}} - F_\rho||_\rho^2 \leq & \frac{c_1' \tilde{c}_3 M \log M \log m}{m} \left(\frac{\log M}{\log\log M}\right)^2 \\ & + \frac{c_3' c_4'}{\tilde{c}_0} \sqrt{\hat{C}_1} M (\log M)^{\frac{1}{\tilde{c}_1}} e^{-c_2 (\frac{c_5' n}{\log n})^{\frac{1}{d}}} \\ & + 2\tilde{c}_4^2 \,(\log M)^{-\frac{2\alpha\lambda}{d}} \,(\log\log M)^{2\left(\frac{1}{d}+1\right)\alpha\lambda}. \end{aligned} \tag{A.37}$$

To balance the first and the third error terms, we can choose the number of nonzero parameters in the functional net as

$$M = \left\lfloor \frac{m}{(\log m)^{\frac{2\alpha\lambda}{d}+4}} \right\rfloor. \tag{A.38}$$

Then to make the second error term have the same rate as the other two terms, we can choose the second-stage sample size

$$n \geq \hat{C}_2 (\log m)^d \log\log m, \tag{A.39}$$

where $\hat{C}_2$ is a sufficiently large positive constant. When m is sufficiently large, we have that $(\frac{2\alpha\lambda}{d}+4)\log\log m \leq \tilde{C}_1 \log m$ for some constant $\tilde{C}_1 \in (0,1)$. Therefore, we get the desired convergence rate with the constant $\tilde{C}_2 = c_1' \tilde{c}_3 + \frac{c_3' c_4'}{\tilde{c}_0}\sqrt{\hat{C}_1} + 2\tilde{c}_4^2 (1-\tilde{C}_1)^{-\frac{2\alpha\lambda}{d}}$. ■

Appendix B. Useful Lemmas

In this appendix, we provide some additional useful lemmas for the proof. The following lemma, which demonstrates the rates of approximating a function f by its embedded function $L_K f$ is used in our approximation analysis, and can be found in (Eberts and Steinwart, 2013, Theorem 2.2, Theorem 2.3), where we choose P_X as the Lebesgue measure on $\mathbb{R}^d$, $p = \infty$, and $q = 2$ as specified in their results.

Lemma 13 *Let $f \in L_2(\mathbb{R}^d) \cap L_\infty(\mathbb{R}^d)$, and suppose that the embedding kernel K is chosen as* (3.6)*. Then we have*

$$\|f - L_K f\|_{L_2(\mathbb{R}^d)} \leq C_{d,r}\, \omega_{r,L_2(\mathbb{R}^d)}\left(f, \frac{\gamma}{\sqrt{2}}\right), \tag{B.1}$$

where $C_{d,r}$ is a constant only depending on d and r. Moreover, we have that $L_K f \in \mathcal{H}_\gamma(\mathbb{R}^d)$, with the norm

$$\|L_K f\|_{H_\gamma} \leq (2^r - 1)\pi^{-\frac{d}{4}} \|f\|_{L_2(\mathbb{R}^d)} \gamma^{-\frac{d}{2}}. \tag{B.2}$$

Next, we present a result from our previous work (Song et al., 2023a, Proposition 2) that establishes the rates of approximating a continuous function by a deep ReLU neural network, which essentially leverages the idea of realizing the multivariate piecewise linear interpolation through a deep ReLU neural network, as discussed in Yarotsky (2018).

Lemma 14 *Let $M, N \in \mathbb{N}$, ω_g be the modulus of continuity of a function $g : [-R, R]^{d_1} \to [-L, L]$ with $L > 0$, then there exists a deep ReLU neural network H_{NN} with depth $J = d_1^2 + d_1 + 1$ and M nonzero parameters, such that*

$$\sup_{y\in[-R,R]^{d_1}} |g(y) - H_{NN}(y)| \leq 2d_1\omega_g\left(\frac{c_0 d_1^{\frac{4}{d_1}} R}{M^{\frac{1}{d_1}}}\right), \tag{B.3}$$

where c_0 is a constant that is independent of R, d_1, and M. Moreover, H_{NN} is constructed to output

$$H_{NN}(y) = \sum_{i=1}^{(N+1)^{d_1}} c_i \psi\left(\frac{N}{2R}(y - b_i)\right), \tag{B.4}$$

where $c_i \in \mathbb{R}, b_i \in \mathbb{R}^{d_1}$ are free parameters that depend on g, and satisfy the conditions $|c_i| \leq L$ and $\|b_i\|_\infty \leq R$. The function $\psi : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\psi(y) = \sigma\Big(\min\big\{\min_{k\neq j}(1 + y_k - y_j), \min_k(1 + y_k), \min_k(1 - y_k)\big\}\Big). \tag{B.5}$$

Furthermore, N is the number of grid points in each direction, and has the following relationship with M:

$$\bar{C}_1 d_1^4 (N + 1)^{d_1} \leq M \leq \bar{C}_2 d_1^4 (N + 1)^{d_1}, \tag{B.6}$$

where $\bar{C}_1$ and $\bar{C}_2$ are positive constants.

The following lemma gives a bound of the difference between two min functions.

Lemma 15 *For any $k \in \mathbb{N}$, and any $x_i, y_i \in \mathbb{R}$, $i = 1, ..., k$. Then there holds*

$$|\min\{x_1, ..., x_k\} - \min\{y_1, ..., y_k\}| \leq 2\sum_{i=1}^{k} |x_i - y_i|. \tag{B.7}$$

Proof. Note that for any $l \in \mathbb{N}$, we have $\min\{x_1, ..., x_l\} = x_l - \sigma(x_l - \min\{x_1, ..., x_{l-1}\})$, hence,

$$\begin{aligned}
&|\min\{x_1, ..., x_k\} - \min\{y_1, ..., y_k\}| \\
&\leq |x_k - \sigma(x_k - \min\{x_1, ..., x_{k-1}\}) - y_k + \sigma(y_k - \min\{y_1, ..., y_{k-1}\})| \\
&\leq 2|x_k - y_k| + |\min\{x_1, ..., x_{k-1}\} - \min\{y_1, ..., y_{k-1}\}| \\
&\leq \cdots \\
&\leq 2|x_k - y_k| + ... + 2|x_2 + y_2| + |x_1 - y_1| \\
&\leq 2\sum_{i=1}^{k} |x_i - y_i|,
\end{aligned}$$

which completes the proof. ■

The following concentration inequality, which is employed in our generalization analysis, can be found in (Györfi et al., 2002, Theorem 11.4). While this result specifically considers elements of $\mathcal{F}$ defined on $\mathbb{R}^d$, it is also applicable to situations where the elements of $\mathcal{F}$ are defined on an arbitrary set.

Lemma 16 *Let $m \in \mathbb{N}$, and assume that $\rho(\{(f,y) \in \mathcal{Z} : |y| \leq L\}) = 1$ for some $L \geq 1$. Let $\mathcal{G}$ be a set of functions mapping from $\mathcal{F}$ to $[-L, L]$. Then for any $0 < \delta \leq 1/2$ and $\alpha, \beta > 0$, we have*

$$\begin{aligned} &P\left\{\exists G \in \mathcal{G} : ||G - F_\rho||_\rho^2 - (\mathcal{E}_D(G) - \mathcal{E}_D(F_\rho)) \geq \delta\left(\alpha + \beta + ||G - F_\rho||_\rho^2\right)\right\} \\ &\leq 14 \sup_{f_1^m} \mathcal{N}_1\left(\frac{\beta\delta}{20L}, \mathcal{G}, f_1^m\right) \exp\left(-\frac{\delta^2(1-\delta)\alpha m}{214(1+\delta)L^4}\right). \end{aligned} \tag{B.8}$$

The following lemma establishes the covering number bound for our functional network and is used in the generalization analysis.

Lemma 17 *Let $\epsilon > 0$, then for any $m \in \mathbb{N}$ and $f_1, ..., f_m \in \mathcal{F}$, we have*

$$\mathcal{N}_1\left(\epsilon, \pi_L \mathcal{H}_{d_1,M} \circ L_K, f_1^m\right) \leq 2\left(\frac{4eL}{\epsilon}\right)^{c_2' JM \log M}, \tag{B.9}$$

where $J = d_1^2 + d_1 + 1$, and M is the number of nonzero parameters in the hypothesis space.

Proof. For any functional class $\mathcal{H}$, (Györfi et al., 2002, Lemma 9.2) shows a relationship between the ϵ-covering number and the ϵ-packing number,

$$\mathcal{N}_1\left(\epsilon, \mathcal{H}, x_1^m\right) \leq \mathcal{M}_1\left(\epsilon, \mathcal{H}, x_1^m\right). \tag{B.10}$$

Furthermore, we denote $Pdim(\mathcal{H})$ as the pseudo-dimension of $\mathcal{H}$, which is defined as the largest integer N for which there exists a set of points $(x_1, \ldots, x_N, y_1, \ldots, y_N) \in \mathcal{X}^N \times \mathbb{R}^N$ such that for any binary vector $(a_1, \ldots, a_N) \in \{0,1\}^N$, there exists some function $h \in \mathcal{H}$ that satisfies the condition

$$h(x_i) > y_i \Longleftrightarrow a_i = 1, \qquad \forall i.$$

A relationship between the ϵ-packing number and the pseudo-dimension was stated in (Haussler, 1992, Theorem 6), showing that

$$\mathcal{M}_1\left(\epsilon, \pi_L \mathcal{H}, x_1^m\right) \leq \mathcal{M}_1\left(\epsilon, \mathcal{H}, x_1^m\right) \leq 2\left(\frac{2eL}{\epsilon}\log\frac{2eL}{\epsilon}\right)^{Pdim(\mathcal{H})}. \tag{B.11}$$

Then, by utilizing the pseudo-dimension bound for deep neural networks with any piecewise polynomial activation function, including ReLU as a specific case, as established in (Bartlett et al., 2019, Theorem 7), we obtain a complexity bound for $\mathcal{H}_{NN}$, the class of deep ReLU networks H_{NN} within the hypothesis space defined in (4.3):

$$Pdim(\mathcal{H}_{NN}) \leq c_2' JM \log M, \tag{B.12}$$

where $J = d_1^2 + d_1 + 1$, and c_2' is an absolute constant. Finally, it is important to note that the nonzero parameters in $\pi_L \mathcal{H}_{d_1,M} \circ L_K$ are exclusively contained within $\mathcal{H}_{NN}$. By combining (B.10), (B.11), and (B.12), we can derive the following result

$$\mathcal{N}_1\left(\epsilon, \pi_L \mathcal{H}_{d_1,M} \circ L_K, f_1^m\right) = \mathcal{N}_1\left(\epsilon, \pi_L \mathcal{H}_{NN}, x_1^m\right) \leq 2\left(\frac{4eL}{\epsilon}\right)^{c_2' JM \log M},$$

where $x_i = T(L_K f_i)$, for $i = 1, \ldots, m$. We thus complete the proof. ■

The following lemma states the convergence rates of kernel quadrature when utilizing the theoretically optimal quadrature scheme outlined in Section 2, as detailed in (Bach, 2017, pp. 17–19). In general, these convergence rates are influenced by the smoothness of the RKHS reproduced by the kernel and can be quantified by the eigenvalue decay of its associated integral operator.

Lemma 18 *Let $\Omega = [0,1]^d$, and let $K_0 : \Omega \times \Omega \to \mathbb{R}$ be a Mercer kernel with the eigensystem of its associated integral operator represented by $\{\lambda_i, \phi_i\}$. Additionally, let $K : \Omega \times \Omega \to \mathbb{R}$ be a continuous kernel that satisfies $K(\cdot, t) \in \mathcal{H}_{K_0}$ for all $t \in \Omega$. For the kernel embedding step expressed in* (2.10) *and its theoretically optimal quadrature scheme detailed in* (4.17)*, we can establish the following error bound*

$$E\left(\left\|L_K f_i - \widehat{L_K} f_i\right\|_{K_0}\right) \leq c_4' \sqrt{\lambda_q}, \tag{B.13}$$

where $q = \frac{c_5' n}{\log n}$, and c_4', c_5' are positive constants.

References

Robert A Adams and John JF Fournier. *Sobolev Spaces*. Elsevier, 2003.

Arash A Amini and Martin J Wainwright. Sampled forms of functional PCA in reproducing kernel Hilbert spaces. *The Annals of Statistics*, 40(5):2483–2510, 2012.

Anima Anandkumar, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Nikola Kovachki, Zongyi Li, Burigede Liu, and Andrew Stuart. Neural operator: Graph kernel network for partial differential equations. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.

Fernando Andreotti, Joachim Behar, Sebastian Zaunseder, Julien Oster, and Gari D Clifford. An open-source framework for stress-testing non-invasive foetal ECG extraction algorithms. *Physiological Measurement*, 37(5):627, 2016.

Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. *The Journal of Machine Learning Research*, 18(1):714–751, 2017.

Peter L Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *The Journal of Machine Learning Research*, 20(1):2285–2301, 2019.

Alain Berlinet and Christine Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer Science & Business Media, 2011.

Philippe Besse and James O Ramsay. Principal components analysis of sampled functions. *Psychometrika*, 51(2):285–311, 1986.

Kaushik Bhattacharya, Bamdad Hosseini, Nikola B Kovachki, and Andrew M Stuart. Model reduction and neural networks for parametric PDEs. *The SMAI Journal of Computational Mathematics*, 7:121–157, 2021.

Tony Cai and Ming Yuan. Minimax and adaptive prediction for functional linear regression. *Journal of the American Statistical Association*, 107(499):1201–1216, 2012.

Hervé Cardot, Frédéric Ferraty, and Pascal Sarda. Spline estimators for the functional linear model. *Statistica Sinica*, pages 571–591, 2003.

Dong Chen, Peter Hall, and Hans-Georg Müller. Single and multiple index functional regression models with nonparametric link. *The Annals of Statistics*, 39(3):1720–1747, 2011.

Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995.

Charles K Chui, Shao-Bo Lin, and Ding-Xuan Zhou. Deep neural networks for rotation-invariance approximation and learning. *Analysis and Applications*, 17(05):737–772, 2019.

Moo K Chung. *Statistical and Computational Methods in Brain Image Analysis*. CRC press, 2013.

Felipe Cucker and Steve Smale. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39(1):1–49, 2002.

Felipe Cucker and Ding-Xuan Zhou. *Learning Theory: An Approximation Theory Viewpoint*, volume 24. Cambridge University Press, 2007.

Maarten V de Hoop, Nikola B Kovachki, Nicholas H Nelsen, and Andrew M Stuart. Convergence rates for learning linear operators from noisy data. *SIAM/ASA Journal on Uncertainty Quantification*, 11(2):480–513, 2023.

Mo Deng, Shuai Li, Alexandre Goy, Iksung Kang, and George Barbastathis. Learning to synthesize: robust phase retrieval at low photon counts. *Light: Science & Applications*, 9(1):36, 2020.

Ronald A DeVore and George G Lorentz. *Constructive Approximation*, volume 303. Springer Science & Business Media, 1993.

Winston W Dou, David Pollard, and Harrison H Zhou. Estimation in functional regression for general exponential families. *The Annals of Statistics*, 40(5):2421–2451, 2012.

Mona Eberts and Ingo Steinwart. Optimal regression rates for SVMs using Gaussian kernels. *Electronic Journal of Statistics*, 7:1–42, 2013.

Gregory E Fasshauer. Positive definite kernels: past, present and future. *Dolomites Research Notes on Approximation*, 4:21–63, 2011.

Frédéric Ferraty and Philippe Vieu. *Nonparametric Functional Data Analysis: Theory and Practice*, volume 76. New York: Springer, 2006.

Emad M Grais and Mark D Plumbley. Single channel audio source separation using convolutional denoising autoencoders. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1265–1269. IEEE, 2017.

Xiaoxiao Guo, Wei Li, and Francesco Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 481–490, 2016.

László Györfi, Michael Köhler, Adam Krzyżak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*, volume 1. Springer, 2002.

Peter Hall and Joel J Horowitz. Methodology and convergence rates for functional linear regression. *The Annals of Statistics*, 35(1):70–91, 2007.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data mining, Inference, and Prediction*. Springer, 2009.

David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

Tailen Hsing and Randall Eubank. *Theoretical Foundations of Functional Data Analysis, with an Introduction to Linear Operators*. John Wiley & Sons, 2015.

Hachem Kadri, Emmanuel Duflos, Philippe Preux, Stéphane Canu, Alain Rakotomamonjy, and Julien Audiffren. Operator-valued kernels for learning from functional response data. *The Journal of Machine Learning Research*, 17(20):1–54, 2016.

Motonobu Kanagawa, Bharath K Sriperumbudur, and Kenji Fukumizu. Convergence guarantees for kernel-based quadrature rules in misspecified settings. *Advances in Neural Information Processing Systems*, 29, 2016.

Robert Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981.

Yuehaw Khoo and Lexing Ying. SwitchNet: a neural network model for forward and inverse scattering problems. *SIAM Journal on Scientific Computing*, 41(5):A3182–A3201, 2019.

Yuehaw Khoo, Jianfeng Lu, and Lexing Ying. Solving parametric PDE problems with artificial neural networks. *European Journal of Applied Mathematics*, 32(3):421–435, 2021.

Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra. On universal approximation and error bounds for fourier neural operators. *The Journal of Machine Learning Research*, 22(1):13237–13312, 2021.

Samuel Lanthaler, Siddhartha Mishra, and George E Karniadakis. Error estimates for DeepONets: A deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1):tnac001, 2022.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. *Advances in Neural Information Processing Systems*, 33:6755–6766, 2020.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Hao Liu, Haizhao Yang, Minshuo Chen, Tuo Zhao, and Wenjing Liao. Deep nonparametric estimation of operators between infinite dimensional spaces. *The Journal of Machine Learning Research*, 25(24):1–67, 2024.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

Tong Mao, Zhongjie Shi, and Ding-Xuan Zhou. Theory of deep convolutional neural networks III: Approximating radial functions. *Neural Networks*, 144:778–790, 2021.

Tong Mao, Zhongjie Shi, and Ding-Xuan Zhou. Approximating functions with multi-features by deep convolutional neural networks. *Analysis and Applications*, pages 1–33, 2022.

Alexander Meister. Optimal classification and nonparametric regression for functional data. *Bernoulli*, 22:1729–1744, 2016.

Hrushikesh N Mhaskar. Local approximation of operators. *Applied and Computational Harmonic Analysis*, 64:194–228, 2023.

Hrushikesh Narhar Mhaskar and Nahmwoo Hahm. Neural networks for functional approximation and system identification. *Neural Computation*, 9(1):143–159, 1997.

Hans-Georg Müller and Ulrich Stadtmüller. Generalized functional linear models. *The Annals of Statistics*, 33(2):774 – 805, 2005.

Yong Zheng Ong, Zuowei Shen, and Haizhao Yang. Integral autoencoder network for discretization-invariant learning. *The Journal of Machine Learning Research*, 23(286): 1–45, 2022.

Chang Qiao, Di Li, Yuting Guo, Chong Liu, Tao Jiang, Qionghai Dai, and Dong Li. Evaluation and development of deep neural networks for image super-resolution in optical microscopy. *Nature Methods*, 18(2):194–202, 2021.

Zhen Qin, Qingliang Zeng, Yixin Zong, and Fan Xu. Image inpainting based on deep learning: A review. *Displays*, 69:102028, 2021.

Carlos Ramos-Carreño, José Luis Torrecilla, Miguel Carbajo-Berrocal, Pablo Marcos, and Alberto Suárez. scikit-fda: a python package for functional data analysis. *Journal of Statistical Software*, 109:1–37, 2024.

J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. New York: Springer, 2005.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. ISBN 026218253X.

Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021.

John A Rice and Colin O Wu. Nonparametric mixed effects models for unequally sampled noisy curves. *Biometrics*, 57(1):253–259, 2001.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

Fabrice Rossi and Brieuc Conan-Guez. Functional multi-layer perceptron: a non-linear tool for functional data analysis. *Neural Networks*, 18(1):45–60, 2005.

Fabrice Rossi, Brieuc Conan-Guez, and François Fleuret. Functional data analysis with multi layer perceptrons. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, volume 3, pages 2843–2848. IEEE, 2002.

Fabrice Rossi, Nicolas Delannay, Brieuc Conan-Guez, and Michel Verleysen. Representation of functional data in neural networks. *Neurocomputing*, 64:183–210, 2005.

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

Bernard W Silverman. Smoothed functional principal components analysis by choice of norm. *The Annals of Statistics*, 24(1):1–24, 1996.

Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A Hilbert space embedding for distributions. In *Algorithmic Learning Theory*, pages 13–31. Springer, 2007.

Linhao Song, Jun Fan, Di-Rong Chen, and Ding-Xuan Zhou. Approximation of nonlinear functionals using deep ReLU networks. *Journal of Fourier Analysis and Applications*, 29 (4):50, 2023a.

Linhao Song, Ying Liu, Jun Fan, and Ding-Xuan Zhou. Approximation of smooth functionals using deep ReLU networks. *Neural Networks*, 166:424–436, 2023b.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Science & Business Media, 2008.

Taiji Suzuki. Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality. In *International Conference on Learning Representations*, 2019.

Matus Telgarsky. Benefits of depth in neural networks. In *Conference on Learning Theory*, pages 1517–1539. PMLR, 2016.

Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. Deep learning on image denoising: An overview. *Neural Networks*, 131:251–275, 2020.

Matt P Wand and M Chris Jones. *Kernel Smoothing*. CRC press, 1994.

Rui Wang and Yuesheng Xu. Functional reproducing kernel hilbert spaces for non-point-evaluation functional data. *Applied and Computational Harmonic Analysis*, 46(3):569–623, 2019.

Larry Wasserman. *All of Nonparametric Statistics*. Springer, 2006.

Zhun Wei and Xudong Chen. Physics-inspired convolutional neural network for solving full-wave inverse scattering problems. *IEEE Transactions on Antennas and Propagation*, 67(9):6138–6148, 2019.

Holger Wendland. *Scattered Data Approximation*, volume 17. Cambridge University Press, 2004.

Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional linear regression analysis for longitudinal data. *The Annals of Statistics*, 33(6):2873–2903, 2005a.

Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association*, 100(470):577–590, 2005b.

Junwen Yao, Jonas Mueller, and Jane-Ling Wang. Deep learning for functional data analysis with adaptive basis layers. In *International Conference on Machine Learning*, pages 11898–11908. PMLR, 2021.

Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.

Dmitry Yarotsky. Optimal approximation of continuous functions by very deep ReLU networks. In *Conference on Learning Theory*, pages 639–649. PMLR, 2018.

Ming Yuan and Tony Cai. A reproducing kernel hilbert space approach to functional linear regression. *The Annals of Statistics*, 38(6):3412–3444, 2010.

Ding-Xuan Zhou. Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 48(2):787–794, 2020.

Hang Zhou, Fang Yao, and Huiming Zhang. Functional linear regression for discretely observed data: from ideal to reality. *Biometrika*, 2022.

Hang Zhou, Dongyi Wei, and Fang Yao. Theory of functional principal component analysis for discretely observed data. *The Annals of Statistics*, 53(5):2103–2127, 2025.