CovNet: Covariance networks for functional data on multidimensional domains

Soham Sarkar1 | Victor M. Panaretos2

1Theoretical Statistics and Mathematics Unit, Indian Statistical Institute, Delhi Centre, New Delhi, India
2Institut de Mathématiques, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

Correspondence
Victor M. Panaretos, Institut de Mathématiques, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.
Email: victor.panaretos@epfl.ch

Funding information
Schweizerischer Nationalfonds zur Förderung der Wissenschaftlichen Forschung

Abstract
Covariance estimation is ubiquitous in functional data analysis. Yet, the case of functional observations over multidimensional domains introduces computational and statistical challenges, rendering the standard methods effectively inapplicable. To address this problem, we introduce Covariance Networks (CovNet) as a modelling and estimation tool. The CovNet model is universal—it can be used to approximate any covariance up to desired precision. Moreover, the model can be fitted efficiently to the data and its neural network architecture allows us to employ modern computational tools in the implementation. The CovNet model also admits a closed-form eigendecomposition, which can be computed efficiently, without constructing the covariance itself. This facilitates easy storage and subsequent manipulation of a covariance in the context of the CovNet. We establish consistency of the proposed estimator and derive its rate of convergence. The usefulness of the proposed method is demonstrated via an extensive simulation study and an application to resting state functional magnetic resonance imaging data.

KEYWORDS
deep learning, FDA, neural network, non-parametric model, random field, universal approximation

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2022 The Authors. Journal of the Royal Statistical Society: Series B (Statistical Methodology) published by John Wiley & Sons Ltd on behalf of Royal Statistical Society.
1 | INTRODUCTION

We consider the problem of covariance estimation from a collection of functional observations defined over a multidimensional domain. To be precise, let $\mathcal{X} = \{X(u) : u \in Q\}$ be a compactly supported random field, that is, a real-valued second-order stochastic process on a compact set $Q \subset \mathbb{R}^d$, with covariance kernel $c(u, v) = \text{Cov}(X(u), X(v))$. We want to estimate $c$ based on an independent and identically distributed (i.i.d.) sample $X_1, \ldots, X_N \sim \mathcal{X}$. In particular, we work in the framework of functional data analysis (FDA, Hsing & Eubank, 2015; Ramsay & Silverman, 2002), where we assume that $\mathcal{X}$ takes values in $L_2(Q)$, the space of all real-valued square-integrable functions on $Q$.

Covariance estimation, along with mean estimation, is a fundamental problem in functional data analysis and has multifaceted applications, for example, in regression, prediction, classification. This problem has been studied extensively for observations over one-dimensional domains (i.e., $d = 1$) or curve data (see Wang et al., 2016 for a detailed review). The same is, however, not true for observations over multidimensional domains. Although, in principal, these two regimes are similar, and most methods for curve data can be ‘readily used’ for data over multidimensional domains, in practice, the dimensionality of the problem draws a clear distinction between the two paradigms. To appreciate this, suppose that we observe the random fields on a grid of size $K \times \cdots \times K$ in $Q \subset \mathbb{R}^d$. In this case, the estimation of the empirical covariance requires $\mathcal{O}(K^{2d})$ computations. The storage cost for this estimator is also of the order $\mathcal{O}(K^{2d})$ which, for $d = 2$, can be prohibitive even for $K \approx 100$. The problem becomes even more severe when $d$ is larger ($d \geq 3$), which is increasingly common, for example, for observations over spatial volumes or of a spatio-temporal nature. Moreover, subsequent manipulation, for example, inversion, needed in applications, requires computation in the order of $\mathcal{O}(K^{3d})$, leading to a prohibitive computational burden.

To put things into perspective, consider the 1000 Functional Connectomes Project (https://www.nitrc.org/projects/fcon_1000/) which contains functional magnetic resonance imaging (fMRI) of brains for more than 1200 individuals. For each individual, the data consist of three-dimensional (3D) brain scans on a grid of size $64 \times 64 \times 33$ taken at 2-s intervals over 225 time points. Covariance estimation is of utmost importance in fMRI studies as it captures the connectivity patterns in the brain (Aston & Kirch, 2012; Stoehr et al., 2021). It is also instrumental in identifying active regions in the brain and to statistically check their significance (Bowring et al., 2021; Sommerfeld et al., 2018; Telschow & Schwartzman, 2022; Vandekar et al., 2019). At the same time, it is extremely difficult to do so non-parametrically because of the dimensionality of the problem. For instance, the empirical covariance estimator for the 3D fMRI data would be an object of size $64 \times 64 \times 33 \times 64 \times 64 \times 33$ which requires 68 GB memory to compute and store (at 32-byte precision). This is impossible with a regular computer which usually has 16 or 32 GB of memory. Also, apart from looking at the connectivity pattern of the brain as a 3D object, it is also of importance to check how these patterns evolve over time, that is, to consider the full four-dimensional data on a grid of size $64 \times 64 \times 33 \times 225$. The problem becomes even more severe in this case, where the empirical covariance would require approximately $3.4 \times 10^6$ GB of memory during computation and for storage (see also Aston & Kirch, 2012; Stoehr et al., 2021).

To alleviate this curse of dimensionality, further modelling assumptions are often made on the underlying covariance, the most popular being that of separability. A separable model assumes that the true covariance over the multidimensional domain can be factored into several
covariances over one-dimensional domains, that is, \( c(\mathbf{u}, \mathbf{v}) = c_1(u_1, v_1) \times \cdots \times c_d(u_d, v_d) \) for \( \mathbf{u}, \mathbf{v} \in Q \). This greatly simplifies the problem and entails enormous computational savings. For instance, in the case of observations on a grid, a separable model can be estimated with \( O(dK^2) \) computations and has the same order of storage requirements. The gain during application of the model is even better—the inversion of the model requires \( O(dK^3) \) computations compared to the \( O(K^{3d}) \) for the empirical covariance. Despite all these advantages, separability is merely a modelling assumption, which is highly restrictive and often violated in practice (Aston et al., 2017; Bagchi & Dette, 2020; Constantinou et al., 2017; Rougier, 2017). Still, it is often the preferred choice in practice, not because it is believed to hold, but rather for the savings that it entails (Gneiting et al., 2006; Pigoli et al., 2018). Perhaps it is safe to say that the popularity of the separable model stems from the non-availability of a better alternative. It is worth clarifying here that when the data are sparse (each random field is observed at a few randomly scattered locations) there do exist methods applicable to multidimensional domains without assuming separability (e.g., Wang et al. (2022) used a penalised method, leading to the use of tensor-products of splines, yielding a low-rank approximation to the covariance). However, such approaches are infeasible in the dense regime, where each random field is measured on the same dense grid (e.g., the fMRI data), and which is our main interest in this paper. It is the denseness of the measurements that gives rise to the severe computational challenges mentioned above. In a dense regime, the approach of Wang et al. (2022) is infeasible since it requires computation of the ‘raw covariances’—equivalent to the computation of the empirical covariance. For instance, for random fields on \( \mathbb{R}^2 \), the method of Wang et al. (2022) breaks down for grid sizes as small as \( 10 \times 10 \) (for \( N = 500 \)) on a regular laptop with Windows 11 (64-bit) operating system, 16 GB RAM and Intel Core i7-8650U (1.90 GHz) CPU. The situation degrades quickly with dimension: in \( \mathbb{R}^3 \), the method breaks at a rather modest grid size of \( 5 \times 5 \times 5 \).

In an effort to deliver a more general yet tractable approach, we propose a new model for covariance estimation using neural networks. Neural networks have long been successfully used in non-parametric function estimation, and recently, they have been shown to successfully overcome the curse of dimensionality in non-parametric regression (Bauer & Kohler, 2019; Schmidt-Hieber, 2020). Also, they have been used for mean estimation of functional data over multidimensional domains (Wang et al., 2021). Motivated by the success of neural networks, we propose Covariance Networks (CovNet) as a framework for the estimation of the covariance of multidimensional random fields. A CovNet is a positive semi-definite function on \( Q \times Q \) described by a neural network architecture. In particular, we define and study three variants: the shallow CovNet model and the deep CovNet model which differ with respect to the depth of the network; and the deep shared CovNet model which is a restricted (regularised) version of the deep CovNet model. Our framework features several advantages, namely:

(a) It is genuinely non-parametric—any covariance can be approximated up to arbitrary precision via a CovNet structure. We establish this so-called universal approximation property of the CovNet models in Theorems 1 and 2. Moreover, the proposed model has an explicit functional form. This functional form has its own advantage in applications such as kriging, where we do not need to interpolate or smooth the estimated covariance before use.

(b) Fitting a CovNet to the data is computationally tractable. The models we introduce can be fitted at the level of the data, without the need to compute or store any high-order objects. Moreover, the neural network structure allows us to exploit modern machine learning tools during the estimation. These are discussed in Section 3.
The special structure of the CovNet models ensures that the eigendecomposition of the associated operator can be obtained without the need to explicitly form the operator itself. Thus, we can access the eigensystem of the fitted CovNet very easily without ever forming any higher order objects. This allows us to store and subsequently manipulate the fitted model very efficiently (see Section 4).

The CovNet estimators come with theoretical guarantees. In particular, we establish their consistency and derive their rates of convergence (Section 6). Our analyses are fully non-parametric—we make no structural assumption on the underlying covariance \( \mathcal{C} \) for our derivations.

The rest of the article is organised as follows. We lay out our methodology in the next section. In particular, we begin by describing the shallow CovNet model in Section 2.1 and establish its universality. In Section 2.2, we extend the shallow CovNet model by using deep architectures, leading to the deep and the deep shared CovNet models. In Section 3, we demonstrate how the CovNet models can be efficiently estimated in practice. The eigendecomposition of the CovNet operator is discussed in Section 4. In Section 5, we demonstrate the usefulness of the proposed CovNet models by means of a detailed simulation study and an application to the fMRI data. We establish the theoretical properties of these models in Section 6. Some concluding remarks are made in Section 7. The proofs of our asymptotic results are provided in the Online Supplement. The Online Supplement also covers some related mathematical ideas, as well as some further numerical results.

## 2 COVARIANCE NETWORKS

We start with some background concepts, more details can be found in Hsing and Eubank (2015) and Section A of the Online Supplement. Let \( Q \) be a compact subset of \( \mathbb{R}^d \) and let \( \mathcal{X} = \{ X(u) : u \in Q \} \) be a random element of \( L_2(Q) \). For \( d = 1 \), \( \mathcal{X} \) is usually referred to as a random curve, whereas for \( d > 1 \), it is referred to as a random field. We assume that \( \mathcal{X} \) has finite second moment, that is, \( \mathbb{E}(||\mathcal{X}||^2) < \infty \), which ensures the existence of its mean \( m = \mathbb{E}(\mathcal{X}) \) and covariance \( \mathcal{C} = \mathbb{E}\{(\mathcal{X} - m) \otimes (\mathcal{X} - m)\} \) (both the expectations are in the Bochner sense, see Hsing & Eubank, 2015). Here, \( ||\cdot|| \) is the \( L_2 \)-norm associated with the inner-product \( \langle f, g \rangle = \int_Q f(u)g(u)\,du \) for \( f, g \in L_2(Q) \), and the tensor product \( h \otimes g \) denotes the rank-1 operator \( f \mapsto \langle f, g \rangle h \). The covariance \( \mathcal{C} \) is a linear operator from \( L_2(Q) \) onto itself, given by

\[
\mathcal{C}f(u) = \int_Q c(u, v) f(v)\,dv, \quad f \in L_2(Q).
\]

Here, \( c \in L_2(Q \times Q) \) defined as \( c(u, v) = \text{Cov}(X(u), X(v)) \) is the covariance kernel associated with \( \mathcal{X} \). We also say that \( \mathcal{C} \) is the integral operator associated with the kernel \( c \). The operator \( \mathcal{C} \) is positive semi-definite and the kernel \( c \) is non-negative definite. The Hilbert–Schmidt norm \( ||\cdot||_2 \) of \( \mathcal{C} \) is finite, and \( ||\mathcal{C}||_2 = ||c||_{L_2(Q \times Q)} \). Thus, the covariance operator \( \mathcal{C} \) and the covariance kernel \( c \) are linked by an isometric isomorphism. Consequently, we can use \( \mathcal{C} \) and \( c \) interchangeably, and the estimation of the covariance \( \mathcal{C} \) is equivalent to the estimation of the kernel \( c \). Since the object of interest is the covariance rather than the mean, we work under the assumption that \( m = \mathbb{E}(\mathcal{X}) = 0 \), unless specifically mentioned.
2.1 Shallow architecture

We propose to estimate the covariance kernel $c$ using the following neural network structure:

$$c_{\text{sh}}(u, v) = \sum_{r=1}^{R} \sum_{s=1}^{R} \lambda_{r,s} \sigma(w_r^T u + b_r) \sigma(w_s^T v + b_s), \quad u, v \in Q,$$

where $R \in \mathbb{N}$ is the width, $\sigma: \mathbb{R} \to \mathbb{R}$ is an activation function, and $\Lambda := ((\lambda_{r,s}))$ is a positive semi-definite matrix. The parameters $w_r \in \mathbb{R}^d$ and $b_r \in \mathbb{R}$ for $r = 1, \ldots, R$ are the weights and the biases of the model (1). Positive semi-definiteness of $\Lambda$ readily implies that $c_{\text{sh}}$ is a non-negative definite kernel. For a given activation function $\sigma$ and width $R \in \mathbb{N}$, we define

$$F_{R,\sigma}^{\text{sh}} = \{ c_{\text{sh}} \text{ of the form (1)} : \Lambda = ((\lambda_{r,s})) \succeq 0, w_1, \ldots, w_R \in \mathbb{R}^d, b_1, \ldots, b_R \in \mathbb{R} \},$$

to be the class of shallow Covariance Network kernels or shallow CovNet kernels. We also define

$$F_{R,\sigma}^{\text{sh}} = \{ G : G \text{ is an integral operator with kernel } g \in F_{R,\sigma}^{\text{sh}} \},$$

to be the class of shallow covariance network operators or shallow CovNet operators.

We call the structure (1) shallow because each of the constituents $\sigma(w_r^T \cdot + b_r)$ for $r = 1, \ldots, R$ of the kernel (1) is a shallow neural network, that is, a neural network with a single hidden layer. The special structure of the kernel (1) allows us to visualise it as a neural network with two hidden layers, as depicted in Figure 1. In the first layer, starting from the inputs $u$ and $v$, single-layer perceptrons $\sigma(w_r^T u + b_r)$ and $\sigma(w_s^T v + b_s)$, $r = 1, \ldots, R$ are computed. In the next layer, these outputs are cross-multiplied with the weights $\lambda_{r,s}$ to produce the final result $c_{\text{sh}}(u, v)$. As one can see, this is a feed-forward neural network (see Anthony & Bartlett, 1999, chapter 6), which is completely determined (for fixed $\sigma$ and $R$) by the parameters $w_1, \ldots, w_R, b_1, \ldots, b_R$ and $\Lambda = ((\lambda_{r,s}))$ (with the added restriction on $\Lambda$).

As mentioned in the introduction, the shallow CovNet structure (1) is a general model, in the sense that any covariance kernel can be approximated with arbitrary precision using a shallow CovNet kernel of the form (1). Thus, we do not need to make any assumption on the underlying covariance $c$, resulting in a completely non-parametric procedure. However, we do need a particular condition on the activation function $\sigma$ of the network.

**Definition 1.** (Sigmoidal activation) An activation function $\sigma: \mathbb{R} \to [0, 1]$ is said to be sigmoidal if it is non-decreasing with $\lim_{x \to -\infty} \sigma(x) = 1$ and $\lim_{x \to +\infty} \sigma(x) = 0$.

In probabilistic terms, a sigmoidal function is a cumulative distribution function. In the machine learning literature, such a function is sometimes referred to as a squashing function or a squasher. Sigmoidal activations are very common in the literature of neural networks. One of the most popular activation functions, the logistic function $\sigma(t) = 1/(1 + \exp(-t))$ is a sigmoidal function often also referred to as the sigmoid. Many other popularly used activation functions are also sigmoidal (see Györfi et al., 2002, chapter 16, for a plethora of examples). It is worthwhile to note that the definition of sigmoidal functions is not universal. In this article, whenever we refer to a sigmoidal function, we mean it in the sense of Definition 1.

If we use a sigmoidal activation function, then any covariance kernel can be approximated up to arbitrary precision using a shallow CovNet kernel of the form (1). Such a property is often referred to as the universal approximation property in the machine learning literature.
Theorem 1. Let $c : Q \times Q \to \mathbb{R}$ be the kernel of the covariance operator $C$. Also, assume that the activation function $\sigma$ is sigmoidal. Then, for every $\epsilon > 0$, there exists $R \in \mathbb{N}$ and $c_{sh} \in P_{R,\sigma}^{sh}$ such that

$$\int_{Q \times Q} |c(u, v) - c_{sh}(u, v)|^2 \, du \, dv \leq \epsilon.$$ 

If in addition $c$ is continuous, then the same conclusion holds uniformly. That is, for every $\epsilon > 0$, we can find $R \in \mathbb{N}$ and $c_{sh} \in P_{R,\sigma}^{sh}$ such that

$$\sup_{u,v \in Q} |c(u, v) - c_{sh}(u, v)| \leq \epsilon.$$ 

Remark 1. The proof of the theorem rests on the universal approximation property of single hidden layer neural networks on the class of square integrable functions on $Q$. The sigmoidal condition on the activation function ensures this property, but is not necessary (see, e.g., Pinkus, 1999).

The construction of the shallow CovNet in Figure 1 shows an immediate way to make the structure deep by augmenting more layers. It is well-known that shallow networks may require a rather large width to approximate a function, whereas the same precision can be achieved by using a deep and less wide network (Eldan & Shamir, 2016; Liang & Srikant, 2017; Poggio et al., 2017). Deep networks can capture more complex structures than the shallow networks with much fewer parameters. Moreover, during training, shallow networks are more prone to get stuck at bad local minima, which are usually avoided by deep networks (Choromanska et al., 2015). In the next section, we extend the CovNet structure by incorporating deep networks instead of perceptrons in the construction.
2.2 Deep architectures

We start with a brief description of deep neural networks. For an integer \( L > 1 \), an integer-tuple \( p = (p_1, \ldots, p_L) \), matrices \( W_1 \in \mathbb{R}^{p_1 \times d} \), \( W_2 \in \mathbb{R}^{p_2 \times p_1} \), \ldots \, W_L \in \mathbb{R}^{p_L \times p_{L-1}} \), vectors \( b_1 \in \mathbb{R}^{p_1} \), \ldots \, b_L \in \mathbb{R}^{p_L} \), \( w_{L+1} \in \mathbb{R}^{p_L} \), and \( b_{L+1} \in \mathbb{R} \), define the function \( g : \mathbb{R}^d \rightarrow \mathbb{R} \) which maps \( u \mapsto g(u) \) recursively as follows:

\[
\begin{align*}
  u_1 &= \sigma(W_1 u + b_1) \\
  u_{l+1} &= \sigma(W_{l+1} u_l + b_{l+1}) \quad \text{for } l = 1, \ldots, L - 1, \\
  g(u) &= \sigma(W_{L+1}^T u_L + b_{L+1}).
\end{align*}
\]  

(4)

Here, for a vector \( z \in \mathbb{R}^p \), \( \sigma(z) \) represents the component-wise application of the function \( \sigma \). The function \( g \) is a deep neural network, where \( L \) is the number of hidden layers or depth, \( p_1, \ldots, p_L \) are the widths of the hidden layers (\( p_{\text{max}} = \max\{p_1, \ldots, p_L\} \) is sometimes referred to as the width of the network) and \( W_1, \ldots, W_L, w_{L+1}, b_1, \ldots, b_L, b_{L+1} \) are the network parameters. A schematic representation of the deep neural network is shown in Figure 2a. Starting from the input \( u \), we go to the first hidden layer by multiplying it with the weight matrix \( W_1 \), adding the bias \( b_1 \), and applying the activation function \( \sigma \) component-wise on the resultant. The same structure is repeated for all the subsequent layers. We define the class

\[
D_{L,p} = \{ g : \mathbb{R}^d \rightarrow \mathbb{R} \text{ of the form (4)} \} \text{ with } W_1 \in \mathbb{R}^{p_1 \times d}, b_1 \in \mathbb{R}^{p_1}, W_2 \in \mathbb{R}^{p_2 \times p_1}, b_2 \in \mathbb{R}^{p_2}, \ldots, W_L \in \mathbb{R}^{p_L \times p_{L-1}}, b_L \in \mathbb{R}^{p_L}, w_{L+1} \in \mathbb{R}^{p_L}, b_{L+1} \in \mathbb{R} \},
\]

(5)

to be the class of all deep neural networks with depth \( L \) and widths \( p_1, \ldots, p_L \). A network from the class \( D_{L,p} \) has \( \sum_{l=0}^{L} (p_l + 1)p_{l+1} \) parameters, where \( p_0 = d \) and \( p_{L+1} = 1 \).

We define the deep CovNet kernel as

\[
c_{d}(u, v) = \sum_{r=1}^{R} \sum_{s=1}^{R} \lambda_{r,s} g_r(u) g_s(v), \quad u, v \in Q,
\]

(6)

where \( g_1, \ldots, g_R \in D_{L,p} \) and \( \Lambda := ((\lambda_{r,s})) \) is positive semi-definite. This is similar to the shallow CovNet kernel (1), except the constituents \( g_r(u) \) are deep networks of the form (4) instead of the perceptrons \( \sigma(w_r^T u + b_r) \). A schematic representation of the deep CovNet kernel is shown in Figure 2b. We define the class of deep CovNet kernels and the corresponding class of operators as

\[
\begin{align*}
F_{R,L,p,\sigma}^d &= \{ c_{d} \text{ of the form (6)} : \Lambda = ((\lambda_{r,s})) \geq 0, g_1, \ldots, g_R \in D_{L,p} \} \\
\tilde{F}_{R,L,p,\sigma}^d &= \{ G : G \text{ is the integral operator associated with kernel } g \in F_{R,L,p,\sigma}^d \}.
\end{align*}
\]

(7)

Remark 2. It is possible to allow the individual networks \( g_1, \ldots, g_R \) in (6) to have different depths and widths, allowing for more flexible models. However, this complicates the analysis, so we do not pursue this model in this paper.

The deep CovNet model is quite rich. Moreover, it retains the universal approximation property of the shallow CovNet model (Theorem 2). But the number of parameters of the deep CovNet model can be quite large, making it prone to overfitting. Thus, some sort of regularisation is needed for the deep CovNet structure to make it more stable. We do this by enforcing weight
sharing among the constituents as follows. For an integer $L > 1$, integer-tuple $\mathbf{p} = (p_1, \ldots, p_L)$, matrices $W_1 \in \mathbb{R}^{p_1 \times d}$, $W_2 \in \mathbb{R}^{p_2 \times p_1}$, $\ldots$, $W_L \in \mathbb{R}^{p_L \times p_{L-1}}$, and vectors $\mathbf{b}_1 \in \mathbb{R}^{p_1}$, $\ldots$, $\mathbf{b}_L \in \mathbb{R}^{p_L}$, we define the networks $g_1, \ldots, g_R$ jointly as

$$
\begin{align*}
\mathbf{u}_1 &= \sigma(W_1 \mathbf{u} + \mathbf{b}_1), \\
\mathbf{u}_{l+1} &= \sigma(W_{l+1} \mathbf{u}_l + \mathbf{b}_{l+1}) \quad \text{for } l = 1, \ldots, L - 1, \\
g_r(\mathbf{u}) &= \sigma(\omega_r^\top \mathbf{u}_L + \beta_r), \quad r = 1, \ldots, R,
\end{align*}
$$

(8)
where \( \omega_r \in \mathbb{R}^{p_L} \) and \( \beta_r \in \mathbb{R} \) for \( r = 1, \ldots, R \). Individually, each of the networks \( g_1, \ldots, g_R \) is an element of the class \( D_{L,p} \). But collectively, they share certain patterns among themselves, specifically they share all their parameters except for the ones in the final layer (see Figure 3). We formally define the deepshared CovNet kernel as

\[
c_{ds}(\mathbf{u}, \mathbf{v}) = \sum_{r=1}^{R} \sum_{s=1}^{R} \lambda_{rs} g_r(\mathbf{u}) g_s(\mathbf{v}), \quad \mathbf{u}, \mathbf{v} \in Q,
\]

where \( g_1, \ldots, g_R \) are networks with shared structures as defined in (8). Note that the form in (9) is essentially the same as (6) except for how the functions \( g_1, \ldots, g_R \) are defined. A schematic representation of the structure (9) is shown in Figure 3. We also define the class of deepshared CovNet kernels and the corresponding operators as

\[
F_{R,L,p,\sigma}^{ds} = \{ c_{ds} \text{ of the form } (9) : \Lambda = ((\lambda_{rs})) \geq 0, g_1, \ldots, g_R \text{ of the form } (8) \}
\]

\[
\hat{F}_{R,L,p,\sigma}^{ds} = \left\{ G : G \text{ is the integral operator associated with kernel } g \in F_{R,L,p,\sigma}^{ds} \right\}.
\]

The shared structure drastically reduces the number of parameters of the model. A deepshared CovNet kernel from the class \( F_{R,L,p,\sigma}^{ds} \) requires \( \sum_{i=0}^{L-1} (p_i + 1)p_{i+1} + R(p_L + 1) + R(R+1)/2 \) parameters, compared to \( R \left( \sum_{i=0}^{L} (p_i + 1)p_{i+1} \right) + R(R+1)/2 \) for a deep CovNet kernel from the class \( F_{R,L,p,\sigma}^{d} \). To appreciate this, suppose that each of the hidden layers have the same width \( R \), that is, \( p_1 = \cdots = p_L = R \). Then, the deepshared kernel contains \( O(R^2) \) parameters, compared to \( O(R^3) \) parameters for the deep kernel.

Similar to the shallow CovNet model, both the deep and the deepshared models are universal approximators, that is, they can approximate any covariance kernel up to any desired accuracy.

**Theorem 2.** Let \( C \) be a covariance operator on \( L_2(Q) \) with kernel \( c \). Also, assume that the activation function \( \sigma \) is sigmoidal. Then, for every \( \epsilon > 0 \), we can find a deep CovNet kernel \( c_d \) and a deepshared CovNet kernel \( c_{ds} \) such that

\[
\int_{Q \times Q} |c(\mathbf{u}, \mathbf{v}) - c_d(\mathbf{u}, \mathbf{v})|^2 \, d\mathbf{u} \, d\mathbf{v} \leq \epsilon \quad \text{and} \quad \int_{Q \times Q} |c(\mathbf{u}, \mathbf{v}) - c_{ds}(\mathbf{u}, \mathbf{v})|^2 \, d\mathbf{u} \, d\mathbf{v} \leq \epsilon.
\]

If in addition \( c \) is continuous, then the same conclusion holds uniformly. That is, for every \( \epsilon > 0 \), we can find \( c_d \) and \( c_{ds} \) such that

\[
\sup_{\mathbf{u}, \mathbf{v} \in Q} |c(\mathbf{u}, \mathbf{v}) - c_d(\mathbf{u}, \mathbf{v})| \leq \epsilon \quad \text{and} \quad \sup_{\mathbf{u}, \mathbf{v} \in Q} |c(\mathbf{u}, \mathbf{v}) - c_{ds}(\mathbf{u}, \mathbf{v})| \leq \epsilon.
\]

Although both the models are universal approximators, the depth, width and number of components \( R \) of the approximator for the two models can be different. Moreover, although this result is similar to the one for the shallow model, we expect the deep and the deepshared approximators to have a much smaller number of parameters compared to the shallow approximator (Poggio et al., 2017).

**Remark 3.** The proof of the theorem again depends on the universal approximation property of deep neural networks, and the sigmoidal condition on the activation function is not
FIGURE 3  A schematic representation of the deepshared CovNet structure

necessary. From the proof of the theorem, one can see that the universal approximation
property of the deep (and the deepshared) CovNet structure is guaranteed whenever the
associated class of deep neural networks has the universal approximation property. In par-
ticular, deep and deepshared CovNet models with the highly popular ReLU activation:
\[ \sigma(t) = \max\{t, 0\} \]
are also universal approximators.

Theorems 1 and 2 justify the use of covariance networks (shallow, deep or deepshared) for
modelling the covariance kernel \( c \). Given a fixed width \( R \) and activation function \( \sigma \), approximation
by a shallow CovNet amounts to determining a \( G \) in the class \( \mathcal{F}_{R,\sigma}^{sh} \) that is closest to \( C \) in terms of
the Hilbert-Schmidt norm, that is,

\[ \hat{C}_{R,\sigma}^{sh} \in \arg \min_{G \in \mathcal{F}_{R,\sigma}^{sh}} \| C - G \|_{2}^{2}. \] (11)

Given a sample of random fields \( \mathcal{X}_1, \ldots, \mathcal{X}_N \sim \mathcal{X} \) in \( L_2(Q) \), with covariance \( C \), we can replace \( C \)
in (11) by the empirical covariance operator \( \hat{C}_N = N^{-1} \sum_{n=1}^{N} \mathcal{X}_n \otimes \mathcal{X}_n \) to obtain an estimator:

\[ \hat{C}_{R,N}^{sh} \in \arg \min_{G \in \mathcal{F}_{R,\sigma}^{sh}} \| \hat{C}_N - G \|^{2}. \] (12)
We call this the shallow CovNet estimator. Similarly, for given width \( R \), depth \( L \) and activation \( \sigma \), we can define the estimators based on the deep and deepshared models as

\[
\hat{C}^{d}_{R,L,N} \in \arg \min_{G \in \mathcal{F}^{d}_{R,L,p}} \| \hat{C} - G \|_{2}^{2} \quad \text{and} \quad \hat{C}^{ds}_{R,L,N} \in \arg \min_{G \in \mathcal{F}^{ds}_{R,L,p}} \| \hat{C} - G \|_{2}^{2},
\]

(13)

which we call the deep CovNet and the deepshared CovNet estimators, respectively. The estimators \( \hat{C}^{sh}_{R,N} \), \( \hat{C}^{d}_{R,L,N} \) and \( \hat{C}^{ds}_{R,L,N} \) can be seen as regularised versions of the empirical covariance, by projection into the corresponding CovNet classes. Nevertheless, it is crucial to note here that, although the definition of the estimators involve \( \hat{C} \), we never actually need to form the empirical covariance in order to construct them. The estimators \( \hat{C}^{sh}_{R,N} \), \( \hat{C}^{d}_{R,L,N} \) and \( \hat{C}^{ds}_{R,L,N} \) can be computed directly at the level of the data, without the need to ever store or access the 2d-dimensional object \( \hat{C} \). We discuss the implementation details in the next section.

Note that although these estimators depend on the widths \( p_1, \ldots, p_L \), we have suppressed it in the notation for ease of exposition. Also, in our numerical experiments, we have used \( p_1 = \cdots = p_L = R \), which justifies this notation. This choice is motivated by the empirical evidence that suggests using the same width for all the hidden layers (see Bengio, 2012, Section 19.3.2).

**Remark 4.** Observe the notation in (11)–(13). In either of the equations, we cannot guarantee that the minimiser is unique. Here, and throughout the article, by the notation \( \hat{G} \in \arg \min_{G \in \mathcal{F}} \| G - C \|_{2}^{2} \), we mean that \( \hat{G} \) is an element (out of possibly many) of the class \( \mathcal{F} \) satisfying \( \| \hat{G} - C \|_{2}^{2} \leq \| G - C \|_{2}^{2} \) for all \( G \in \mathcal{F} \). Note that this non-uniqueness does not affect the subsequent developments, in particular the asymptotic theory for the estimator.

### 3 Practical Implementation

Note that for all three CovNet models (1), (6) and (9), the covariance kernel is of the form

\[
\sum_{r=1}^{R} \sum_{z=1}^{R} \lambda_{r,z} g_{r}(u) g_{z}(v), \quad u, v \in \mathcal{Q},
\]

(14)

where \( \Lambda := (\lambda_{r,z}) \) is positive semi-definite, and \( g_{1}, \ldots, g_{R} \) are allowed to vary keeping up to the model under consideration. In particular, \( g_{r}(u) = \sigma(w_{r}^{T} u + b_{r}) \) for shallow CovNet, \( g_{r}'s \) are the individual deep neural networks from the class \( D_{R,L} \) for deep CovNet, and \( g_{r}'s \) are jointly defined as in (8) for deepshared CovNet. We denote the generic class of all such kernels (with the additional structures on the functions \( g_{1}, \ldots, g_{R} \)) by \( \mathcal{F}_{R} \) and the corresponding class of operators by \( \tilde{\mathcal{F}}_{R} \). Thus, \( \mathcal{F}_{R} \) (resp., \( \tilde{\mathcal{F}}_{R} \)) can be \( \mathcal{F}^{sh}_{R,\sigma}, \mathcal{F}^{d}_{R,L,p,\sigma} \) or \( \mathcal{F}^{ds}_{R,L,p,\sigma} \) (resp., \( \tilde{\mathcal{F}}^{sh}_{R,\sigma}, \tilde{\mathcal{F}}^{d}_{R,L,p,\sigma} \) or \( \tilde{\mathcal{F}}^{ds}_{R,L,p,\sigma} \)) depending on the situation. Here, and throughout, we suppress the dependence on \( L, p_1, \ldots, p_L \) and \( \sigma \) for convenience, unless specifically mentioned.

For a given \( R \in \mathbb{N} \), the CovNet structure (14) is completely determined by the parameters of \( g_{1}, \ldots, g_{R} \), and the coefficients \( \Lambda = (\lambda_{r,z}) \). In particular, apart from \( \Lambda \), these parameters are \( w_{1}, \ldots, w_{R}, b_{1}, \ldots, b_{R} \) for the shallow CovNet (1), the weights and the biases of the individual deep neural networks \( g_{1}, \ldots, g_{R} \) for the deep CovNet (6), and \( W_{1}, \ldots, W_{L}, b_{1}, \ldots, b_{L} \) and \( \omega_{1}, \ldots, \omega_{R}, \beta_{1}, \ldots, \beta_{R} \) for the deepshared CovNet (9). Thus, obtaining the estimators in (12) or (13) is equivalent to finding these parameters minimising the corresponding criterion

\[
\ell' := \ell'(\Theta) = \| \hat{C} - G \|_{2}^{2}.
\]
Here, we use $\Theta$ to denote all the estimable parameters (i.e., the parameters of $g_1, \ldots, g_R$, and $\Lambda$), taking into account the positive-definiteness of $\Lambda$ (which reduces the number of free parameters).

As already mentioned, we do not need to form the tensor $\hat{\Sigma}$ (or candidate tensor $\mathcal{G}$) to minimise $\mathcal{E}$. The trick is to not fit the covariance directly, but to instead fit the observed fields $\mathcal{X}_1, \ldots, \mathcal{X}_N$ themselves by neural networks with shared structures. To be precise, consider the fields

$$\mathcal{X}^\text{NN}_n(u) = \sum_{r=1}^R \xi_{n,r} g_r(u), \quad n = 1, \ldots, N,$$

where $\xi_{n,r} \in \mathbb{R}$ for $n = 1, \ldots, N$, $r = 1, \ldots, R$, and $g_1, \ldots, g_R$ are the constituents of the CovNet model (14) under consideration. The fields $\mathcal{X}^\text{NN}_1, \ldots, \mathcal{X}^\text{NN}_N$ are themselves neural networks with shared components $g_1, \ldots, g_R$ (and hence shared parameters), but potentially different coefficients $\xi_{n,r}$. Define the operator

$$\mathcal{G}^\text{NN}_{R,N} = \frac{1}{N} \sum_{n=1}^N (\mathcal{X}^\text{NN}_n - \mathcal{X}^\text{NN}) \otimes (\mathcal{X}^\text{NN}_n - \mathcal{X}^\text{NN}),$$

which is the empirical covariance based on the neural networks $\mathcal{X}^\text{NN}_1, \ldots, \mathcal{X}^\text{NN}_N$, and let $\mathcal{F}^\text{NN}_{R,N}$ be the class of all such covariance operators:

$$\mathcal{F}^\text{NN}_{R,N} = \{\text{all empirical covariance operators of the form (16)}\}.$$

Because of the shared structure of the networks $\mathcal{X}^\text{NN}_n$, the kernel of the operator $\mathcal{G}^\text{NN}_{R,N}$ has the CovNet form (14):

$$\mathcal{G}^\text{NN}_{R,N}(u, v) = \sum_{r=1}^R \sum_{s=1}^R \lambda_{r,s} g_r(u) g_s(v),$$

where $\lambda_{r,s} = N^{-1} \sum_{n=1}^N (\xi_{r,n} - \bar{\xi}_r)(\xi_{s,n} - \bar{\xi}_s)$. At the same time, if $N > R$, any CovNet operator from the class $\mathcal{F}_R$ can be expressed as an empirical covariance operator of the form (16). Specifically, for every $\mathcal{G} \in \mathcal{F}_R$, we can find $N$ networks of the form (15) such that $\mathcal{G}$ is the empirical covariance operator of those $N$ networks. This should be intuitively clear, but we nevertheless state this formally below, and a detailed construction is shown in Section C of the Online Supplement.

**Proposition 1.** If $N > R$, then $\mathcal{F}^\text{NN}_{R,N} = \mathcal{F}_R$.

This simple correspondence between the CovNet operator class $\mathcal{F}_R$ and the class of empirical covariances of neural networks with shared structure (17) is of great consequence in estimating the CovNet model based on the observed data. Note that the criterion $\mathcal{E}$ is non-convex in the parameters, so we cannot find the explicit minimiser. Instead, we need to rely on some iterative minimisation procedure, for example, gradient descent or its variants (Buduma & Locascio, 2017, chapters 2 and 4). The application of gradient descent requires us to calculate the gradient of the minimisation criterion. But, with modern optimisation routines, this can be done numerically on a computer, without the need to compute the derivatives analytically. In particular, the special neural network structure of our method allows us to employ automatic differentiation techniques to efficiently compute the derivative at machine precision (Baydin et al., 2018). In essence, the minimiser can be efficiently obtained if we can compute the criterion efficiently. This is where the empirical covariance formulation (15) and (16) come in handy. With this
formulation, we compute the criterion $\ell'$ as a function of the parameters of $g_1, \ldots, g_R$, and coefficients $\xi_{n,r}$ instead of $\lambda_{r,s}$. At each step of gradient descent, we obtain the fields $\chi_n^{\text{NN}}$ as feed-forward neural networks. The minimisation criterion $\ell'$ can be computed by simply computing inner products between the observed fields $\chi_n$ and the fitted networks $\chi_n^{\text{NN}}$, as shown below. For simplicity, we assume that the observed fields $\chi_1, \ldots, \chi_N$ are centred, so that the empirical covariance is $\hat{C}_N = N^{-1} \sum_{n=1}^N \chi_n \otimes \chi_n$. We also assume that the fitted networks $\chi_1^{\text{NN}}, \ldots, \chi_N^{\text{NN}}$ are centred, so that their empirical covariance is $\hat{S}_{R,N} = N^{-1} \sum_{n=1}^N \chi_n^{\text{NN}} \otimes \chi_n^{\text{NN}}$. With these, we get the following formula for the minimisation criterion:

$$\ell' := \left\| \hat{C}_N - \hat{S}_{R,N} \right\|_2^2 = \frac{1}{N^2} \sum_{n=1}^N \sum_{m=1}^N \langle \chi_n, \chi_m \rangle^2 + \frac{1}{N^2} \sum_{n=1}^N \sum_{m=1}^N \langle \chi_n^{\text{NN}}, \chi_m^{\text{NN}} \rangle^2 - \frac{2}{N^2} \sum_{n=1}^N \sum_{m=1}^N \langle \chi_n, \chi_m^{\text{NN}} \rangle^2.$$

The detailed derivations are shown in Section C.1 of the Online Supplement.

**Remark 5.** The alternative formulation also helps us in imposing positive semi-definiteness on $\Lambda$. After estimating the parameters from the reformulated problem—in particular, the $\xi_{n,r}$’s—we obtain $\lambda_{r,s}$ as $N^{-1} \sum_{n=1}^N (\xi_{n,r} - \bar{\xi}_r)(\xi_{n,s} - \bar{\xi}_s)$. By virtue of this construction, the resulting matrix $\Lambda = (\lambda_{r,s})$ is automatically positive semi-definite. This is quite useful, as it circumvents the need to work with a constrained optimisation problem on a cone.

In practice, we observe the data on a grid of size $D = K_1 \times \cdots \times K_d$, say $(u_1, \ldots, u_D)$. Let us denote the $i$th measurement corresponding to the $n$th field by $X_{ni}$ for $n = 1, \ldots, N$, $i = 1, \ldots, D$. So, we can store the observed fields as an $N \times D$ matrix $X = ((X_{ni}))$. Similarly, the fitted networks $\chi_1^{\text{NN}}, \ldots, \chi_N^{\text{NN}}$ can be evaluated at the $D$ grid points and all of these can also be stored as an $N \times D$ matrix $X^{\text{NN}} = ((X_{ni}^{\text{NN}}))$, where $X_{ni}^{\text{NN}} = \chi_n^{\text{NN}}(u_i)$. We can approximate $\langle \chi_n, \chi_m \rangle$ by the average over the grid points, that is, $\langle \chi_n, \chi_m \rangle \approx D^{-1} \sum_{i=1}^D X_{ni} X_{mi}$. It is easy to see that this is the $(n,m)$th element of the $N \times N$ matrix $D^{-1}XX^T$. Similarly, we approximate $\langle \chi_n^{\text{NN}}, \chi_m^{\text{NN}} \rangle$ and $\langle \chi_n, \chi_n^{\text{NN}} \rangle$ by the corresponding averages $D^{-1} \sum_{i=1}^D X_{ni}^{\text{NN}} X_{mi}^{\text{NN}}$ and $D^{-1} \sum_{i=1}^D X_{ni} X_{ni}^{\text{NN}}$, which are the $(n,m)$th elements of $D^{-1}XX^T$, $X^{\text{NN}}X^{\text{NN}}^T$, and $D^{-1}XX^{\text{NN}}X^{\text{NN}}^T$, respectively. Thus, apart from the computation of $X^{\text{NN}}$, the computational cost of $\ell'$ is $O(N^2D)$. Moreover, to store the model, we only need to store the parameters of $g_1, \ldots, g_R$, and the coefficient matrix $\Lambda$, which is completely free of the grid size $D$. In particular, this amounts to a storage cost of $O(R^2 + Rd)$ for the shallow CovNet model, $O(R^2 + Rp_L + R \sum_{l=0}^{L-1} (p_l + 1)p_{l+1})$ for the deep CovNet model, and $O(R^2 + Rp_L + \sum_{l=0}^{L-1} (p_l + 1)p_{l+1})$ for the deepshared CovNet model ($p_0 = d$ for the latter two). It is easy to see the savings relative to the empirical covariance, which requires $O(ND^2)$ computations and $O(D^2)$ storage. Here, it is worth noting that for $N < D$, instead of storing the whole $D \times D$ empirical covariance, one can only store the raw data matrix, reducing the storage cost to $O(ND)$. However, the covariance function needs to be computed everywhere it needs to be evaluated, thus increasing the computational complexity.

In the above discussion, we have not addressed the computational requirements for $X^{\text{NN}}$. It is not difficult to show that for a fixed set of parameters, computation of $X^{\text{NN}}$ needs $O(DR(N + d))$ operations for the shallow CovNet, $O(DR(N + \sum_{l=0}^{L-1} p_l p_{l+1} + p_l))$ operations for the deep CovNet, and $O(D(NR + \sum_{l=0}^{L-1} p_l p_{l+1} + Rp_L))$ operations for the deepshared CovNet (see Section C.3 in the Online Supplement). Thus, for a fixed set of parameters, the computational cost for the evaluation of $\ell'$ remains linear in the grid size $D$ for all three CovNet models. Of course, we need to re-evaluate the criterion for each step of the gradient descent algorithm. But that is also the case for other modern machine learning methods. Moreover, the computation can be sped up by considering
other techniques from machine learning, such as stochastic or mini-batch version of gradient descent and parallel computing (Bengio, 2012; Buduma & Locascio, 2017).

**Remark 6.** We have not tried to find an analytic expression for the derivative of $\ell$ as a function of the parameters. Instead, we focused more on evaluating the criterion efficiently, and relied on automatic differentiation to compute the gradient. There are three reasons for doing this. Firstly, because of the complex neural network structure, finding analytic expressions for the gradient is cumbersome. This becomes more relevant for the deep and the deepshared CovNet structures. Secondly, we have at our disposal modern optimisation routines, which are very efficient in automatic differentiation, especially with neural network structures such as ours. In our implementation, we have used the autograd feature of pytorch (https://pytorch.org/). Finally, even if we compute the derivatives analytically, when implementing the method on a computer the accumulation of errors for analytic derivatives may sometimes be quite large, especially for complex structures such as neural networks. Automatic differentiation, on the other hand, produces results which are exact up to machine precision, and thus are preferred to analytic derivatives (Baydin et al., 2018).

**Remark 7.** In our derivations, we have assumed that the fields $\mathcal{X}_1, \ldots, \mathcal{X}_N$ as well as the fitted fields $\mathcal{X}^{NN}_1, \ldots, \mathcal{X}^{NN}_N$ are centred. In practice, we can centre the observed fields by subtracting the mean (empirical or estimated by some other method), with negligible computational overhead. For the fitted fields $\mathcal{X}^{NN}_1, \ldots, \mathcal{X}^{NN}_N$, because of their shared structure, the mean turns out to be $\overline{\mathcal{X}}^{NN}(u) = \sum_{r=1}^R \xi_r g_r(u)$. So, centering the fitted fields boils down to centering the coefficients $\xi_{n,r}$. We can use another approach as well, where we do not centre the fields (observed or fitted) beforehand and minimise a slightly different criterion. In this case, we also get an estimate of the mean as a by-product (see Section C.2 in the Online Supplement for details).

### 4 Eigendecomposition of the Estimated Covariance Operator

Once we estimate the covariance, it is important to be able to manipulate it, for example, for regression, prediction or even for visualisation purposes. For such tasks, typical manipulations involve inverting the covariance operator or obtaining its eigendecomposition, either of which may be quite demanding in practice. For instance, for data observed on a grid of size $D = K_1 \times \cdots \times K_d$, the empirical covariance is stored as a $D \times D$ matrix. The inversion in this case requires $\mathcal{O}(D^3)$ operations, which is highly demanding and sometimes even prohibitive. Even if the inverse is constructed, it is available only at the $D \times D$ pre-specified locations. To evaluate the inverse (or the covariance itself, for that matter) at any other location, as required e.g., in kriging, one needs to apply some sort of interpolation or smoothing on a high-dimensional (in our case, $\mathbb{R}^d \times \mathbb{R}^d$) object, which can be even more demanding than the inversion itself. Finally, the cost of storing the inverse and/or the eigenfunctions adds another layer of burden.

By contrast, the proposed CovNet estimators enjoy considerable advantage in this respect. The special form of the CovNet operators allow us to easily compute their eigendecomposition. Note that our estimated CovNet kernels are of the form

$$\hat{c}(u, v) = \sum_{r=1}^R \sum_{s=1}^R \hat{\lambda}_{r,s} \hat{g}_r(u) \hat{g}_s(v), \quad u, v \in \mathcal{Q}.$$
Thus, for the estimated CovNet operator \( \hat{C} \) and for any \( f \in L^2(Q) \),

\[
\hat{C}f(u) = \int_Q \hat{c}(u, v) f(v) \, dv = \sum_{r=1}^{R} \sum_{s=1}^{R} \hat{g}_{r,s}(u) \int_Q \hat{g}_s(v) f(v) \, dv = \sum_{r=1}^{R} a_r \hat{g}_r(u),
\]

where \( a_r = \sum_{s=1}^{R} \hat{g}_{r,s}(v) f(v) \, dv \). This shows that the eigenfunctions of \( \hat{C} \) are of the form \( \psi(u) = \sum_{r=1}^{R} a_r \hat{g}_r(u) \) for some \( a_1, \ldots, a_R \in \mathbb{R} \). Now, for such a function \( \psi \),

\[
\|\psi\|^2 = \sum_{r=1}^{R} \sum_{s=1}^{R} a_r a_s \int_Q \hat{g}_r(u) \hat{g}_s(u) \, du = \sum_{r=1}^{R} \sum_{s=1}^{R} a_r a_s \hat{g}(r, s) = a^T \tilde{G} a,
\]

where \( \hat{g}(r, s) = \int_Q \hat{g}_r(u) \hat{g}_s(u) \, du \), \( a = (a_1, \ldots, a_R)^T \) and \( \tilde{G} = ((\hat{g}(r, s)))_{1 \leq r, s \leq R} \). Also,

\[
\langle \hat{C} \psi, \psi \rangle = \int \int_{Q \times Q} \hat{c}(u, v) \psi(u) \psi(v) \, du \, dv
\]

\[
= \sum_{r=1}^{R} \sum_{s=1}^{R} \hat{g}_{r,s} \int \int_{Q \times Q} \hat{g}_r(u) \hat{g}_s(v) \psi(u) \psi(v) \, du \, dv
\]

\[
= \sum_{r=1}^{R} \sum_{s=1}^{R} \hat{g}_{r,s} \sum_{i=1}^{R} \sum_{j=1}^{R} a_i a_j \int_{Q} \hat{g}_r(u) \hat{g}_i(u) \, du \int_{Q} \hat{g}_s(v) \hat{g}_j(v) \, dv
\]

\[
= \sum_{i=1}^{R} \sum_{j=1}^{R} a_i a_j \left( \sum_{r=1}^{R} \sum_{s=1}^{R} \hat{g}(r, i) \hat{g}(s, j) \right) = \sum_{i=1}^{R} \sum_{j=1}^{R} a_i a_j (\tilde{G} \Lambda \tilde{G})_{ij} = a^T \tilde{G} \Lambda \tilde{G} a.
\]

Thus, finding the leading eigenvalue and eigenfunction of \( \hat{C} \) reduces to maximising \( a^T \tilde{G} \Lambda \tilde{G} a \) subject to \( a^T \tilde{G} a = 1 \). This amounts to solving

\[
(\tilde{G} \Lambda \tilde{G} - \eta \tilde{G}) a = 0.
\]

Again, if \( \psi_i(u) = \sum_{r=1}^{R} a_{i,r} \hat{g}_r(u) \), then we can similarly show that

\[
\langle \psi_i, \psi_j \rangle = a_i^T \tilde{G} a_j \quad \text{and} \quad \langle \hat{C} \psi_i, \psi_j \rangle = a_i^T \tilde{G} \Lambda \tilde{G} a_j,
\]

where \( a_i = (a_{i,1}, \ldots, a_{i,R})^T \) is the vector of coefficients of \( \psi_i \). Thus, finding the subsequent eigenvalues and eigenfunctions also amounts to solving \( (\tilde{G} \Lambda \tilde{G} - \eta \tilde{G}) a = 0 \), with added orthogonality constraints. In summary, finding the eigensystem of the CovNet operator \( \hat{C} \) boils down to finding the solution of a generalised eigenvalue problem (Golub & Van Loan, 2013, chapter 7) involving the non-negative definite matrices \( \tilde{G} \Lambda \tilde{G} \) and \( \tilde{G} \). Several optimisation routines are available to obtain the solution. Also, this can be done very efficiently since the matrices \( \Lambda \) and \( \tilde{G} \) involved in the computations are of the order \( R \times R \), and typical values of \( R \) will be much smaller than \( D = K_1 \times \ldots \times K_d \). The matrix \( \Lambda \) is obtained during the estimation procedure. The only bottleneck is the computation of the matrix \( \tilde{G} \), which involves the integrals

\[
\tilde{g}(r, s) = \int_Q \hat{g}_r(u) \hat{g}_s(u) \, du.
\]
These are integrals on a compact subset of $\mathbb{R}^d$, and any suitable quadrature formula can be used to numerically approximate them. For instance, when $d$ is moderate, we can approximate the integrals using Monte Carlo methods, while for large $d$, we can resort to using quasi-Monte Carlo methods (Dick et al., 2013). Other methods, for example, evaluation on a dense grid, can also be used. In typical FDA applications, $d$ is 2, 3 or 4 (corresponding to spatial/spatio-temporal data on $\mathbb{R}^2$ and $\mathbb{R}^3$), and it suffices to use Monte Carlo integration. For this, we generate independent observations $\bm{u}_1, \ldots, \bm{u}_M$ distributed uniformly on $Q$, and approximate the integral as

$$
\tilde{g}(r, s) \approx \frac{1}{M} \sum_{j=1}^{M} \widehat{g}_r(\bm{u}_j) \widehat{g}_s(\bm{u}_j).
$$

Also, when the functions $\widehat{g}_1, \ldots, \widehat{g}_R$ are bounded (e.g., when the activation $\sigma$ is sigmoidal), we can control the approximation error up to any desired accuracy by selecting $M$ large enough. After generating the observations $\bm{u}_1, \ldots, \bm{u}_M$, $\widehat{G}$ can be obtained by passing them through $\widehat{g}_1, \ldots, \widehat{g}_R$ to create an $R \times M$ matrix $G$, and then computing the outer-product $GG^\top$. It can be verified that the overall computational cost remains linear in $M$ (see Section C.3 in the Online Supplement). So, even with a large value of $M$, the computational time is quite small. Moreover, after obtaining the eigendecomposition, the complete eigenstructure can be stored using an $R \times R$ matrix of coefficients $\widehat{A} = (\widehat{a}_1^\top, \ldots, \widehat{a}_R^\top)^\top$ and a vector of eigenvalues $\widehat{\eta} = (\widehat{\eta}_1, \ldots, \widehat{\eta}_R)$, in addition to the already estimated parameters. The usefulness of the eigendecomposition is shown in Section 5.2.

## 5 | EMPIRICAL STUDY

We now demonstrate the usefulness of the proposed methods by means of a variety of numerical examples. We start with some simulated examples, where we generate the data from a Gaussian process (Adler & Taylor, 2007, chapter 1) on $[0, 1]^d$ with mean 0 and variance $C$. We consider the following five choices for the kernel $c$.

1. **Brownian sheet**: $c(\bm{u}, \bm{v}) = c_{\text{bms}}(u_1, v_1) \times \cdots \times c_{\text{bms}}(u_d, v_d)$ for $\bm{u}, \bm{v} \in [0, 1]^d$, where $c_{\text{bms}}(u, v) = \min\{u, v\}$ is the covariance of the standard Brownian motion (Adler & Taylor, 2007, section 1.4.3).

2. **Rotated Brownian sheet**: $c(\bm{u}, \bm{v}) = \tilde{c}(O\bm{u}, O\bm{v})$, where $O$ is a rotation matrix and $\tilde{c}$ is the covariance kernel of the Brownian sheet from Ex 1.

3. **Integrated Brownian sheet**: $c(\bm{u}, \bm{v}) = c_{\text{ibms}}(u_1, v_1) \times \cdots \times c_{\text{ibms}}(u_d, v_d)$ for $\bm{u}, \bm{v} \in [0, 1]^d$, where $c_{\text{ibms}}(u, v) = (u^2/2)(v - u/3)1\{u \leq v\} + (v^2/2)(u - v/3)1\{u > v\}$ is the covariance of the integrated Brownian motion.

4. **Rotated integrated Brownian sheet**: $c(\bm{u}, \bm{v}) = \tilde{c}(O\bm{u}, O\bm{v})$, where $O$ is a rotation matrix and $\tilde{c}$ is the covariance kernel of the integrated Brownian sheet from Ex 3.

5. **Matérn covariance**: $c_\nu(\bm{u}, \bm{v}) = 2^{1-\nu}/\Gamma(\nu) \left(\sqrt{2\nu} \| \bm{u} - \bm{v} \|_d\right)^\nu K_\nu \left(\sqrt{2\nu} \| \bm{u} - \bm{v} \|_d\right)$, where $\Gamma$ is the gamma function, $K_\nu$ is the modified Bessel function of the second kind and $\| \cdot \|_d$ is the Euclidean distance on $\mathbb{R}^d$ (Rasmussen & Williams, 2006, chapter 4). The Matérn covariance is indexed by the parameter $\nu > 0$, which regulates its smoothness.

Note that the covariance kernels in Ex 1 and 3 are separable. We eliminate the separability in Ex 2 and 4 by introducing a rotation of the domain. The Matérn covariance in Ex 5 is stationary.
and isotropic, but not separable for any finite $v$. On the other hand, none of the other covariances are stationary. Ex 1 and 2 yield continuous but nowhere differentiable random fields, whereas Ex 3 and 4 yield continuously differentiable random fields. For Ex 5, the random fields are $|v| - 1$ times differentiable in the mean-square sense.

We carried out our experiments with $d = 2$ and $d = 3$, which we refer to as two-dimensional (2D) and three-dimensional (3D), respectively. For each experiment, we generated $N$ independent fields at $K \times \cdots \times K$ regular grid points on $[0, 1]^d$. Henceforth, we refer to $K$ as the resolution. We used the three CovNet models (shallow, deep and deep shared) on the generated data to estimate $C$. To facilitate comparison, we also considered the empirical covariance estimator and the best separable covariance estimator (e.g., Dette et al., 2022). For each of these estimators, we computed the relative estimation error $||| \hat{C} - C |||_2 / |||C|||_2$. Note that

$$||| \hat{C} - C |||_2^2 = \int \int _{[0, 1]^d \times [0, 1]^d} (\hat{c}(u, v) - c(u, v))^2 \, du \, dv \quad \text{and} \quad |||C|||_2^2 = \int \int _{[0, 1]^d \times [0, 1]^d} c^2(u, v) \, du \, dv,$$

cannot always be computed analytically. So, we used a Monte Carlo approximation. We generated $M$ points $(u_1, v_1), \ldots, (u_M, v_M)$ from the uniform distribution on $[0, 1]^d \times [0, 1]^d$ and approximated

$$||| \hat{C} - C |||_2^2 \approx \frac{1}{M} \sum_{i=1}^M (\hat{c}(u_i, v_i) - c(u_i, v_i))^2 \quad \text{and} \quad |||C|||_2^2 \approx \frac{1}{M} \sum_{i=1}^M c^2(u_i, v_i).$$

These were then used to approximate the relative errors of the estimators. The advantage of using Monte Carlo is that we can control the approximation error up to any desired accuracy by selecting $M$ large enough. Also, by evaluating the estimators on different sets of locations than where the data were generated, we avoid committing an *inverse crime* (Kaipio & Somersalo, 2005). In particular, we used $M = 50,000$ in 2D and $M = 100,000$ in 3D.

We also considered two different setups based on the sample size and the resolution: (a) fixed resolution $K$ and varying sample size $N$ and (b) fixed sample size $N$ and varying resolution $K$. Also, for the Matérn example, we considered different values of $v$ with fixed sample size and resolution. For setup (a), the results are unremarkable—the errors of all the estimators decrease as $N$ increases. These are reported in Section F of the Online Supplement. The results for setup (b) are rather interesting and exhibit the superiority of the CovNet estimators. We show these results for 2D in Figures 4 and 5, where the reported numbers are the averages of the relative errors based on 25 simulation runs. In the figures, we also show the variability of the estimators by drawing error bars corresponding to three times the SE over the replications. The results in 3D are qualitatively similar, and we report them in Section F of the Online Supplement.

For the CovNet estimators, the results depend on the choice of hyperparameters $R$ and $L$. We used $R = 5, 10, 20, 40, 80$ for the shallow CovNet model, and $L = 2, 3, 4, R = 5, 10, 20, 40$ for the deep and the deep shared CovNet models in our experiments. In Figures 4 and 5, we report the best result (i.e., minimum average estimation error) over the choice of hyperparameters obtained by each CovNet model. In Section 5.1, we discuss a practical method to select the hyperparameters and exhibit the corresponding results. It is seen there that the selection method yields values close to the ‘best choice’. For all the CovNet models, we used the standard sigmoid activation function $\sigma(t) = 1/(1 + \exp(-t))$. For the optimisation involved in fitting these models, we used the Adam optimiser (Kingma & Ba, 2014) available in *pytorch* with a learning rate of 0.01. Further details are given in Section F of the Online Supplement.
FIGURE 4 Relative errors of different methods for different two-dimensional examples. Results are reported for a fixed sample size of 500 and varying resolution. The numbers are averages based on 25 simulation runs. The error bars correspond to 3× SE over the 25 replications. (a) Brownian sheet; (b) Rotated Brownian sheet; (c) Integrated Brownian sheet; (d) Rotated integrated Brownian sheet [Colour figure can be viewed at wileyonlinelibrary.com]

In Figure 4, we show the results for the first four examples (Ex1–4) in 2D with fixed sample size $N = 500$ and varying resolutions $K = 5, 10, 20, 40, 80$. The Brownian sheet and the integrated Brownian sheet examples (Ex 1 and 3) are separable. But even for these examples, the proposed CovNet estimators perform better than the best separable estimator (Figure 4a,c), especially in low resolutions. This shows the ability of the CovNet model to learn the underlying pattern, even when we observe the fields at a rather small number of locations. For the rotated examples (Ex 2 and 4), we chose $O$ to be the $45^\circ$-rotation matrix along the $x$-axis:

$$
O = \begin{pmatrix}
1/\sqrt{2} & -1/\sqrt{2} \\
1/\sqrt{2} & 1/\sqrt{2}
\end{pmatrix}.
$$

The absence of separability of $C$ has dire consequence on the performance of the best separable estimator. The other estimators are seemingly unaffected by this, and the CovNet estimators outperform the empirical estimator. Among the CovNet estimators, the deepshared variant
FIGURE 5 Relative errors of different methods for the Matérn covariance model with $d = 2$. In (a), results are for $N = 250$ and resolution $25 \times 25$ with varying smoothness parameter $\nu$. In (b), results are for $N = 500$ and $\nu = 0.01$ with varying resolution. The numbers are averages based on 25 simulation runs. The error bars correspond to $3 \times$ SE over the 25 replications. [Colour figure can be viewed at wileyonlinelibrary.com]

performed much better than the others. Recall that the integrated Brownian sheet (both the usual and the rotated) is one order smoother than the (corresponding version of) Brownian sheet. While this added smoothness enhances the performance of the CovNet estimators, we see an opposite effect on the other estimators, especially with small resolutions. The difference between the performance of the different CovNet models is also lesser in the smoother examples.

In Figure 5, we show the results for the Matérn covariance model (Ex 5) in 2D. We consider two different setups. In panel (a), we show the results with $N = 250$ and $K = 25$ with varying $\nu$. Here, the empirical covariance performs very poorly, especially when the surfaces are rougher (i.e., for smaller values of $\nu$). The CovNet estimators perform better than the best separable estimator when $\nu$ is small. When $\nu$ is large, that is, the surfaces are smoother, the errors of the best separable estimator is almost indistinguishable from those of the CovNet estimators. However, same relative error does not mean that the estimators share the same characteristics. In fact, in this example, the CovNet estimators have an advantage over the other estimators, which is evident from the eigendecomposition of the estimators (see Figure 8). Detailed discussion on this is given in Section 5.2. In panel (b), we report the results for $N = 500$ and $\nu = 0.01$ with varying resolution $K$. Here, we again see the superiority of the CovNet estimators, especially when the resolution is low.

A few words are in order about the cost of storage and manipulation of the estimators. The storage of the empirical covariance estimator becomes prohibitive rather quickly. Although we can compute the estimation error of the empirical covariance relatively easily, it is very costly to manipulate it, for example, by inverting, for further applications like kriging. The scenario is much better for the best separable estimator. But, both the empirical and the best separable estimators produce a discretised object. Thus, even to evaluate the estimated covariance at a location outside of the observation grid, one needs to interpolate or smooth the estimated covariance. Depending on the smoother used, this can dramatically increase the cost associated with the estimator. The functional form of the CovNet estimators, on the other hand, do not suffer from such
problems. After estimation, the storage of the model is quite cheap—one only needs to store the matrices and vectors associated with the neural network model, which can be done very efficiently. Moreover, using the eigendecomposition methods discussed in Section 4, we can easily manipulate the fitted model.

### 5.1 Choice of hyperparameters

The performance of the proposed method depends on the choice of hyperparameters, namely the number of components $R$ and the depth of the network $L$ (for deep and deepshared models). Thus, it is important to select these hyperparameters from the data, which is quite challenging for neural networks (Bengio, 2012). We can use $V$-fold cross-validation for this purpose, where we split the data into $V$ parts. One of these $V$ parts is used as the validation set and the rest are used as the training set. For a particular choice of hyperparameters, the training set is used to fit the model, and its performance is evaluated on the validation set. This procedure is repeated for all the $V$ parts to get the average cross-validation score for a particular set of hyperparameters. Finally, we select the set of hyperparameters that admits the smallest average cross-validation score.

The alternative formulation of the loss function (cf. Section 3) is again useful for the cross-validation. Suppose that our data is split as $\mathbf{X}_1, \ldots, \mathbf{X}_N$, constituting the training and the validation sets, respectively. The model is fitted on the training set to produce the estimate $\hat{G}^t$. We evaluate the performance of the model on the validation set by computing the loss $\ell_{CV} = \|\hat{G}^t - \hat{G}^v\|_2^2$, where $\hat{G}^v$ is the empirical covariance based on the validation set. Recall that by construction, both $\hat{G}^t$ and $\hat{G}^v$ are of the form

$$\hat{G}^t = \frac{1}{N_1} \sum_{n=1}^{N_1} \mathbf{X}_{n}^{NN} \otimes \mathbf{X}_{n}^{NN}, \quad \hat{G}^v = \frac{1}{N_2} \sum_{n=1}^{N_2} \mathbf{X}_{n}^{NN} \otimes \mathbf{X}_{n}^{NN},$$

where $\mathbf{X}_{1}^{NN}, \ldots, \mathbf{X}_{N_1}^{NN}$ are the neural networks fitted to the training sample (see Section 3). Here, we have assumed w.l.o.g. that the observations are centred. Now, it is easy to see that the loss has the explicit form

$$\ell_{CV} = \frac{1}{N_1^2} \sum_{n=1}^{N_1} \sum_{m=1}^{N_1} \langle \mathbf{X}_n^{NN}, \mathbf{X}_m^{NN} \rangle^2 + \frac{1}{N_2^2} \sum_{n=1}^{N_2} \sum_{m=1}^{N_2} \langle \mathbf{X}_n^{NN}, \mathbf{X}_m^{NN} \rangle^2 - \frac{2}{N_1 N_2} \sum_{n=1}^{N_1} \sum_{m=1}^{N_2} \langle \mathbf{X}_n^{NN}, \mathbf{X}_m^{NN} \rangle^2,$$

which depends only on the inner-products. Thus, we can compute the loss efficiently, without forming the high-order covariances $\hat{G}^v$ or $\hat{G}^t$.

The results for the proposed cross-validation strategy are shown in Table 1 for the examples in 2D. For each of the examples, we report the average relative errors for the three CovNet models (shallow, deep and deepshared) selected via cross-validation based on 20 simulation runs with 500 observations at a resolution of 25 × 25. For each model, we also show the average difference from the least observed relative error over the range of hyperparameters (in parentheses). The average relative errors for the empirical and the best separable estimators are also reported to facilitate comparison. Also, for all the estimators, we report the SEs over the 20 simulations runs (in the next line in italics). In all the examples, the average difference from the best result was less than 0.75% for the shallow CovNet model, 2.11% for the deep CovNet model and 0.76% for

---

**Table 1:**

| Example | Shallow CovNet | Deep CovNet | Deepshared CovNet |
|---------|----------------|-------------|-------------------|
| Relative Error | 0.75% | 2.11% | 0.76% |

| 

---
Table 1 Relative errors (in %) of the CovNet models with hyperparameters chosen using fivefold cross-validation

| Example                          | Emp  | BestSep | Shallow | Deep            | Deepshared |
|----------------------------------|------|---------|---------|-----------------|------------|
| Brownian sheet                   | 9.17 | 8.85    | 9.95 (0.17) | 9.44 (1.39)  | 8.20 (0.21) |
|                                 | 0.61 | 0.62    | 0.57    | 0.78            | 0.66       |
| Rotated Brownian sheet           | 11.70| 65.83   | 12.53 (0.23) | 11.25 (0.46)  | 9.95 (0.36) |
|                                 | 0.32 | 0.12    | 0.34    | 0.29            | 0.38       |
| Integrated Brownian sheet        | 8.28 | 8.28    | 7.37 (0.28) | 7.59 (2.11)   | 6.73 (0.25) |
|                                 | 0.68 | 0.68    | 0.72    | 0.85            | 0.80       |
| Rotated integrated Brownian sheet| 11.92| 71.39   | 10.75 (0.75) | 10.14 (1.66)  | 8.28 (0.68) |
|                                 | 0.47 | 0.23    | 0.49    | 0.58            | 0.63       |
| Matérn ($\nu = 0.001$)          | 51.88| 17.57   | 11.42 (0.36) | 12.18 (0.94)  | 11.30 (0.60) |
|                                 | 0.21 | 0.38    | 0.51    | 0.61            | 0.53       |
| Matérn ($\nu = 0.01$)           | 51.55| 18.27   | 12.63 (0.35) | 12.84 (0.69)  | 12.84 (0.76) |
|                                 | 0.20 | 0.36    | 0.49    | 0.47            | 0.47       |
| Matérn ($\nu = 0.1$)            | 16.52| 11.28   | 11.19 (0.19) | 11.47 (0.45)  | 10.69 (0.20) |
|                                 | 0.34 | 0.50    | 0.49    | 0.49            | 0.49       |
| Matérn ($\nu = 1$)              | 7.56 | 7.53    | 7.62 (0.09) | 7.87 (0.43)   | 7.53 (0.18) |
|                                 | 0.71 | 0.72    | 0.70    | 0.70            | 0.71       |

Notes: The reported numbers are averages based on 20 simulation runs with $N = 500$ and $K = 25$. Average difference from the least observed error over the range of hyperparameters is shown in parentheses. Relative errors (in %) for the empirical and the best separable estimators are also reported. SEs (in %) for all the methods are reported in the next line in italics.

the deepshared CovNet model. These results make a convincing case that the cross-validation method can identify a good set of hyperparameters in practice.

5.2 Estimated eigenstructure

Here, we demonstrate the usefulness of the eigendecomposition of the CovNet estimators. For this purpose, we consider three examples in 2D, the rotated Brownian sheet (Ex 2), the rotated integrated Brownian sheet (Ex 4) and the Matérn covariance (Ex 5) with $\nu = 0.01$. For each example, we plot the eigensurfaces of the CovNet estimators obtained using the method proposed in Section 4. The reported results are based on a single simulation with $N = 500$. For the first two examples, we used a resolution of $10 \times 10$. The Matérn example with $\nu = 0.01$ is much more rough, and a resolution of $10 \times 10$ was too low for all the methods (see Figure 5b). So, for this example, we used a resolution of $25 \times 25$. For comparison, we have also plotted the leading eigensurfaces of the true covariance $C$, the empirical estimator and the best separable estimator. For the CovNet estimators, we selected the hyperparameters via the cross-validation strategy described in the previous section.

In Figure 6, we plot the eigensurfaces for the rotated Brownian sheet. Note that in this case, the eigensurfaces of order seven and beyond explain less than 1% of the total variation of the true covariance. Hence, we plot the first six eigensurfaces for each covariance (the truth and the
FIGURE 6  Heatmaps of the first six eigensurfaces of the true covariance and different covariance estimators in the rotated Brownian sheet example (Ex 2) with $N = 500$ and resolution $10 \times 10$ [Colour figure can be viewed at wileyonlinelibrary.com]

The usefulness of the CovNet estimators is quite evident from these plots. Here, for each surface, we make observations at 100 ($10 \times 10$) locations. Clearly, this is not enough for the empirical or the best separable estimators. In contrast, the shallow and the deepshared CovNet estimators are able to extract the features of the true covariance.

The results for the rotated integrated Brownian sheet are shown in Figure 7. We plot the top four eigensurfaces as the others explain less than 1% of the total variation. In this case, the true covariance is quite smooth, hence the empirical covariance does a better job. Yet, the functional form of the CovNet estimators gives them an edge, which is reflected in the estimation errors.
FIGURE 7 Heatmaps of the first four eigensurfaces of the true covariance and different covariance estimators in the rotated integrated Brownian sheet example (Ex 4) with $N = 500$ and resolution $10 \times 10$. [Colour figure can be viewed at wileyonlinelibrary.com]

In Figure 8, we show the results for the Matérn covariance with $\nu = 0.01$. In this case, the underlying process has rough sample paths. This roughness of the observations has a damaging effect on the performance of the empirical covariance. And, although the estimation error of the best separable estimator is relatively low, the estimated eigensurfaces have little resemblance with the true eigensurfaces. In fact, they are not able to capture the underlying features, and the roughness of the observations can be clearly seen to affect the performance. The shallow and the deepshared CovNet estimators do an excellent job in identifying the salient features, even from the rough observations.

A few comments are in order for the deep CovNet estimator. In all three examples, the deep CovNet estimator is seemingly unable to capture the true eigenstructure. However, the estimation error for this model is rather low. This is perhaps due to the high complexity of the model, which allows it to approximate the covariance well enough. But, without any restriction, the model apparently does not really learn interesting patterns of variation from the data. The added restriction of the deepshared model (in terms of weight sharing) resolves this problem. The deepshared model is quite rich, but at the same time it is able to extract interesting traits from the data.

We would also like to point out the favourable computational aspect of the CovNet estimators in this context. As already mentioned, the eigendecomposition for the CovNet estimators can be performed without forming the covariance operators. This is sharply in contrast with the empirical covariance, for which we need to apply eigendecomposition on a $K^d \times K^d$-dimensional
FIGURE 8 Heatmaps of the first six eigensurfaces of the true covariance and different covariance estimators in the Matérn example (Ex 5) with $\nu = 0.01$, $N = 500$ and resolution $25 \times 25$ [Colour figure can be viewed at wileyonlinelibrary.com]

object, which can be prohibitive depending on $K$ and $d$. The best separable estimator is seemingly immune to this problem. But, even for this estimator, the computation for the eigendecomposition increases with $K$ in the order of $O(dK^3)$ (eigendecomposition of $d$ matrices, each of the order $K \times K$). The eigendecomposition for the CovNet estimators, on the other hand, is completely free of $K$ (after estimation of the model, of course). There is, however, a Monte-Carlo step involved in the process. But, in all the examples, it took us only a few seconds to obtain the eigendecomposition, which was much faster than the other two estimators. Moreover, the functional form of the CovNet models have additional benefits, as can be seen from the plots.
5.3 | Application to fMRI data

To explore the usefulness of our methodology in a real setting, we consider the fMRI data sets from the 1000 Functional Connectomes Project. These data sets are available at https://www.nitrc.org/projects/fcon_1000/, and consist of resting state fMRI scans for more than 1200 subjects collected at different locations all over the world. For each subject, the data consist of 3D scans of the brain taken at a resolution of $64 \times 64 \times 33$ over 225 time points. These data sets were previously analysed by Aston & Kirch (2012); Stoehr et al., (2021) in the FDA setting, where they checked for the stationarity of the 3D MRI scans over time for each of 197 individuals from Beijing, China. For our demonstration, we considered sub69518 from Beijing, which was identified to exhibit stationarity by Aston and Kirch (2012). This gives us 225 3D scans on a grid of size $64 \times 64 \times 33$, which we treat as i.i.d. observations. We preprocessed the data set by removing a polynomial trend of order 3 from each voxel as suggested by Aston and Kirch (2012). Further, we scaled the scans to have voxel-wise unit variance before applying the methodology.

As already mentioned in the introduction, covariance estimation is one of the most important problems for resting state fMRI data as it enables to understand the connectivity patterns of the brain. At the same time, the high-dimensionality of the problem makes it extremely difficult to achieve. In contrast, the CovNet models can be fitted efficiently to this data. In particular, we applied the deepshared CovNet model to this data owing to its superiority over the other CovNet models in the simulations. We selected the hyperparameters ($depth = 6, R = 30$) via the cross-validation strategy described in Section 5.1. In Figure 9a, we show the leading eigenfunction of the fitted CovNet model. For comparison, we also fitted a separable covariance model. But, as pointed out by Aston and Kirch (2012), finding the best separable approximation is also difficult in 3D. So, we used the separable estimator via marginalisation as proposed by Aston and Kirch (2012). The leading eigenfunction of the separable estimator is shown in Figure 9b.

The plots clearly show that the deepshared CovNet is able to capture a much richer structure than the separable model. While the leading eigenfunction of the CovNet model accounts for $42.74\%$ of the variability explained, the same for the separable model is only $0.0004\%$. In fact, to explain $42.74\%$ variability, we would need 45,538 components in the separable model. Moreover, the fitted CovNet is able to extract the natural symmetry in the brain during resting state (Raemaekers et al., 2018), while no such constraints were imposed a priori.

In Figure 10, we show the second and the third leading eigenfunctions of the fitted CovNet which, together with the leading eigenfunction, account for $88.64\%$ of the variability. In contrast, for the separable model, we would require 84578 components to explain $88.64\%$ variability. The plots clearly exhibit the ability of the deepshared CovNet to identify complex structures from the data.

6 | ASYMPTOTIC THEORY

We conclude the article by developing asymptotic theory for our estimators. In particular, we prove that the CovNet estimators are consistent, and derive their rates of convergence. We will consider two different setups—(a) when the fields are fully observed and (b) when the fields are observed on a grid with possible noise contamination.
6.1 The case of fully observed fields

Here, our data consist of i.i.d. random fields $\mathcal{X}_1, \ldots, \mathcal{X}_N$ distributed as $\mathcal{X}$. For convenience, we start by assuming that the random field $\mathcal{X}$ is bounded, that is, there exists $\beta_N > 0$ such that $||\mathcal{X}||^2 \leq \beta_N$ almost surely. This type of boundedness assumption is quite common in the theoretical analysis of neural networks (e.g., Györfi et al., 2002; Schmidt-Hieber, 2020). Note that the
FIGURE 10  (a) The second and (b) the third leading eigenfunctions of the deepshared CovNet fitted to the three-dimensional functional magnetic resonance imaging data. The plots are heatmaps of two-dimensional slices taken over the z-axis of the three-dimensional eigenfunctions. [Colour figure can be viewed at wileyonlinelibrary.com]

bound is allowed to grow with $N$, so this can also be seen as a growing truncation level. We will eventually remove this boundedness condition.

We also need to impose some condition on the approximating class, for which we restrict the eigenstructure of the matrix $\Lambda = (\lambda r s)$ in (14). Specifically, for a constant $\lambda N > 0$, we enforce that $\Lambda \preceq \lambda_N I_R$. Note that for any non-negative definite matrix $\Lambda$, we always have $\Lambda \preceq \lambda_{\text{max}} I_R$, where $\lambda_{\text{max}}$ is the largest eigenvalue of $\Lambda$. Thus, our assumption imposes a restriction on the largest eigenvalue of the matrix $\Lambda$ in (14). We write $F_{R, \lambda_N}$ to denote the restricted class of kernels corresponding to the generic class $F_R$. In particular,
\[ F_{R,\lambda_N}^{\text{sh}} = \{ c_{\text{sh}} \text{ of the form (1)} : 0 \leq \Lambda := (\lambda, r, s) \leq \lambda_N I_R, w_r \in \mathbb{R}^d, b_r \in \mathbb{R} \}, \]
\[ F_{R,L}^{\text{d}} = \{ c_{\text{d}} \text{ of the form (6)} : 0 \leq \Lambda := (\lambda, r, s) \leq \lambda_N I_R, g_1, \ldots, g_R \in D_{L,R} \} \] and
\[ F_{R,L}^{\text{ds}} = \{ c_{\text{ds}} \text{ of the form (9)} : 0 \leq \Lambda := (\lambda, r, s) \leq \lambda_N I_R, \quad g_1, \ldots, g_R \text{ of the form (8) with } p_1 = \cdots = p_L = R \}. \] (18)

denote the restricted shallow, deep, and deepshared CovNet classes. Here, we write \( D_{L,R} \) to denote the deep neural network class \( D_{L,R} \) when \( p_1 = \cdots = p_L = R \). We also denote the class of operators corresponding to the class of kernels \( F_{R,\lambda_N} \) by \( \tilde{F}_{R,\lambda_N} \). Recall that our estimators are
\[ \hat{C}_{R,N}^{\text{sh}} \in \arg \min_{C \in F_{R,\lambda_N}^{\text{sh}}} \| \hat{C}_{N} - C \|_2^2, \quad \hat{C}_{R,L,N}^{\text{d}} \in \arg \min_{C \in F_{R,L}^{\text{d}}} \| \hat{C}_{N} - C \|_2^2 \text{ and } \hat{C}_{R,L,N}^{\text{ds}} \in \arg \min_{C \in F_{R,L}^{\text{ds}}} \| \hat{C}_{N} - C \|_2^2. \] (19)

In the following, we start by proving two different kinds of results. First, we prove consistency of the estimators under appropriate conditions, and then derive their rates of convergence.

**Theorem 3.** Let \( \mathcal{X}_1, \ldots, \mathcal{X}_N \overset{i.i.d.}{\sim} \mathcal{X} \), where \( \mathcal{X} \) takes values in \( L_2(Q) \), and \( Q \) is a compact subset of \( \mathbb{R}^d \). Also assume that \( \| \mathcal{X} \|_2 \leq \beta_N \) almost surely, \( \mathbb{E}(\mathcal{X}) = 0 \) and \( \text{Cov}(\mathcal{X}) = C \). Let \( \hat{C}_{R,N}^{\text{sh}}, \hat{C}_{R,L,N}^{\text{d}} \) and \( \hat{C}_{R,L,N}^{\text{ds}} \) be the shallow, the deep, and the deepshared CovNet estimators given by (19). Suppose that \( R \to \infty, \lambda_N \to \infty \) as \( N \to \infty \), and define \( \Delta_N = \max\{\beta_N, |Q|R \lambda_N\} \).

(a) If \( dR^2 \Delta_N^4 \log(\Delta_N)/N \to 0 \) as \( N \to \infty \), then the shallow CovNet estimator is weakly consistent for \( C \), that is, \( \| \hat{C}_{R,N}^{\text{sh}} - C \|_2^2 \overset{P}{\to} 0 \). Additionally, if \( \Delta_N^4/N^{1-\delta} \to 0 \) for some \( \delta \in (0,1) \), then the estimator is strongly consistent for \( C \), that is, \( \| \hat{C}_{R,N}^{\text{sh}} - C \|_2 \overset{a.s.}{\to} 0 \) as \( N \to \infty \).

(b) Let \( R > d \). If \( L^4 R^8 \Delta_N^4 \log^2(\Delta_N)/N \to 0 \) as \( N \to \infty \), then the deep CovNet estimator is weakly consistent for \( C \), that is, \( \| \hat{C}_{R,L,N}^{\text{d}} - C \|_2^2 \overset{P}{\to} 0 \). Additionally, if \( L^4 R^8 \Delta_N^4 \log^2(\Delta_N)/N^{1-\delta} \to 0 \) for some \( \delta \in (0,1) \), then the estimator is strongly consistent for \( C \), that is, \( \| \hat{C}_{R,L,N}^{\text{d}} - C \|_2 \overset{a.s.}{\to} 0 \) as \( N \to \infty \).

(c) Let \( R > d \). If \( L^4 R^8 \Delta_N^4 \log^2(\Delta_N)/N \to 0 \) as \( N \to \infty \), then the deepshared CovNet estimator is weakly consistent for \( C \), that is, \( \| \hat{C}_{R,L,N}^{\text{ds}} - C \|_2^2 \overset{P}{\to} 0 \). Additionally, if \( L^4 R^8 \Delta_N^4 \log^2(\Delta_N)/N^{1-\delta} \to 0 \) for some \( \delta \in (0,1) \), then the estimator is strongly consistent for \( C \), that is, \( \| \hat{C}_{R,L,N}^{\text{ds}} - C \|_2 \overset{a.s.}{\to} 0 \) as \( N \to \infty \).

The proof of the theorem is based on bias-variance-type decompositions for the estimation error \( \| \hat{C} - C \|_2^2 \). To control the bias, we need the universal approximation property (Theorems 1 and 2), but now with the additional restriction on the classes (18). This is ensured by assuming that \( R, L \) and \( \lambda_N \) go to infinity as \( N \) diverges (see Remark 11 in the Online Supplement). On the other hand, \( dR^2 \Delta_N^4 \log(\Delta_N)/N, L^4 R^8 \Delta_N^4 \log^2(\Delta_N)/N, \) and \( L^4 R^8 \Delta_N^4 \log^2(\Delta_N)/N \) are linked to the variance of the estimators. The conditions of the theorem ensure that the variance also converges to 0 with the sample size for the different estimators.

**Remark 8.** There is a small technical ambiguity in the statement of Theorem 3. The theorem is stated for i.i.d. observations distributed as \( \mathcal{X} \) when \( N \) goes to infinity, whereas the
bound on $\mathcal{X}$ is also allowed to evolve with $N$. Thus, the result is to be understood for a triangular sequence of arrays where, for each $N$, the observations are bounded by a constant, which in turn is allowed to diverge keeping up with the assumption of the theorem. However, we avoid stating the theorem in this generality for ease of exposition. The special case of i.i.d. observations (i.e., when $\beta_N$ is a constant) follows easily from the theorem.

Next, we derive the rate of convergence of the estimators.

**Theorem 4.** Let $\mathcal{X}_1, \ldots, \mathcal{X}_N \overset{i.i.d.}{\sim} \mathcal{X}$, where $\mathcal{X}$ takes values in $\mathcal{L}_2(Q)$, and $Q$ is a compact subset of $\mathbb{R}^d$. Suppose that $\|\mathcal{X}\|^2 \leq \beta_N$ almost surely, $\mathbb{E}(\mathcal{X}) = 0$ and Cov$(\mathcal{X}) = C$. Let $\hat{\mathcal{C}}_{R,N}$, $\hat{\mathcal{C}}_{R,L,N}$ and $\hat{\mathcal{C}}_{R, L,N}$ be the shallow, the deep, and the deep shared CovNet estimators in (19). Then,

$$
\mathbb{E}\left(\|\hat{\mathcal{C}}_{R,N} - C\|^2\right) \leq 2 \inf_{\mathcal{G} \in \mathcal{F}_{R,N}} \|\mathcal{G} - C\|^2 + \mathcal{O}\left(\frac{dR^2\Delta_N^4 \log(N)}{N}\right),
$$

$$
\mathbb{E}\left(\|\hat{\mathcal{C}}_{R,L,N} - C\|^2\right) \leq 2 \inf_{\mathcal{G} \in \mathcal{F}_{R,L,N}} \|\mathcal{G} - C\|^2 + \mathcal{O}\left(\frac{L^4R^8\Delta_N^4 \log^2(N)}{N}\right),
$$

$$
\mathbb{E}\left(\|\hat{\mathcal{C}}_{R, L,N} - C\|^2\right) \leq 2 \inf_{\mathcal{G} \in \mathcal{F}_{R}} \|\mathcal{G} - C\|^2 + \mathcal{O}\left(\frac{L^4R^6\Delta_N^4 \log^2(N)}{N}\right).
$$

Here, $\Delta_N = \max\{\beta_N, |Q|R\lambda_N\}$ is as defined in Theorem 3. In the above, for the deep and the deep shared CovNet estimators, we have assumed that $R > d$.

The theorem clearly shows the bias-variance-type decomposition for the proposed estimators. To get the exact rates of convergence, we need to quantify the bias terms, which is an approximation theoretic problem. If, for example, the bias term is zero for some finite $R$, $L$ and $\lambda_N$, then the derived rate of convergence is the same as that of the empirical estimator, except for the logarithmic term. Thus, in this case, our estimator enjoys a nearly minimax rate of convergence. In general, to get the rate of convergence of the bias, we need to make further assumptions. There are two ways of doing this, either by making assumptions on the eigenstructure of the true covariance or by making assumptions on the smoothness of the underlying field $\mathcal{X}$ (see Section B.2 in the Online Supplement for details). For this line of derivations, the rates depend crucially on the approximation error of the constituents of the CovNet model under consideration. For instance, for the shallow CovNet, if we assume that the underlying field $\mathcal{X}$ takes values in $S^d(Q)$, the Sobolev space of functions of order $a$ on $Q$ (see Mhaskar, 1996), with almost surely bounded Sobolev norm (i.e., $\|\mathcal{X}\|^2_{S^d(Q)} \leq \beta$ a.s.), then by selecting $\lambda_N \asymp R$, we can bound the bias term as $\inf_{\mathcal{G} \in \mathcal{F}_{R,N}} \|\mathcal{G} - C\|^2 \leq R^{-a/d}$ (see (11) in the Online Supplement). So, for consistency we need $R^{10} = \mathcal{O}(N/(d \log(N)))$, while the optimal rate is achieved for $R \asymp (N/d \log(N))^{d/(10d+a)}$, leading to the rate of convergence $\mathcal{O}\left((d \log(N)/N)^{a/(10d+a)}\right)$. Similarly, one can use results from (Langer, 2021; Ohn & Kim, 2019) to bound the bias of deep and deep shared CovNet models.

**Remark 9.** Both Theorems 3 and 4 are derived here for a global minimiser of the loss function. But, in practice, we are not guaranteed to find a global minima. Following the proof of Theorem 3, it can be shown that the consistency results hold as long as the estimator is within $\sigma_F(1)$ of the minimiser, that is, for estimators $\hat{\mathcal{C}}$ satisfying $\|\hat{\mathcal{C}}_N - \hat{\mathcal{C}}\|^2 \leq \inf_{\mathcal{G} \in \mathcal{F}_{R,N}} \|\hat{\mathcal{C}}_N - \mathcal{G}\|^2 + \sigma_F(1)$. Similarly, from the proof of Theorem 4, one can check that
for an approximate minimiser, the rate of convergence gets inflated by the ‘expected minimization gap’ $\mathbb{E}\left(\|\tilde{C}_N - \hat{C}\|^2 - \inf_{\mathcal{G} \in \mathcal{F}_{R,\lambda_N}} \|\tilde{C}_N - \mathcal{G}\|^2 \middle| \mathcal{X}_N\right)$, where $\mathcal{X}_N = \{X_1, \ldots, X_N\}$ denotes the observed data, and $\tilde{C}_N$ is distributed identically to $\hat{C}_N$ but independently of $\mathcal{X}_N$.

Finally, we prove consistency without the boundedness condition on $X$. In this case, we need to modify our estimators slightly. To this extent, let $\mathcal{G}$ be a CovNet operator from the unrestricted class $\mathcal{F}_R$ with kernel $g(u, v) = \sum_{j=1}^{R} \sum_{i=1}^{\lambda} \eta_i e_i^T g_R(u) g_R(v)$. For $\lambda > 0$, define $\mathcal{P}_\lambda$, to be the CovNet operator obtained by thresholding the eigenvalues of $\Lambda = (\lambda_{r,s})$ to $\lambda$. To be the shallow, the deep and the deepshared CovNet estimators, respectively, without any restriction on the underlying structure. We assumed $\lambda_{r,s}$ is the eigendecomposition of $\Lambda$, then we define $\Lambda \lambda_{r,s} = \sum_{i=1}^{\lambda} \min\{\eta_i, \lambda\} e_i e_i^T$ to be the $\lambda$-thresholded version of $\Lambda$. We define $\mathcal{P}_\lambda$, to be the operator with kernel $g_{\lambda}(u, v) = \sum_{j=1}^{R} \sum_{i=1}^{\lambda} \eta_i e_i^T g_R(u) g_R(v)$, where $\eta_i e_i$ is the $(r,s)$th element of the matrix $\Lambda_{r,s}$. By construction, $0 \leq \lambda_{r,s} \leq \lambda$ and consequently, $\mathcal{P}_\lambda$ is an element of the restricted class $\mathcal{F}_{R,\lambda}$. We are now ready to re-define the estimator. Define

$$\tilde{C}_{R,N}^{sh} = \inf_{\mathcal{G} \in \mathcal{F}_{R,\lambda}} \|\hat{C}_N - \mathcal{G}\|^2, \quad \tilde{C}_{R,N}^{d} = \inf_{\mathcal{G} \in \mathcal{F}_{R,\lambda}} \|\hat{C}_N - \mathcal{G}\|^2 \quad \text{and} \quad \tilde{C}_{R,N}^{ds} = \inf_{\mathcal{G} \in \mathcal{F}_{R,\lambda}} \|\hat{C}_N - \mathcal{G}\|^2,$$

This modified estimators are consistent, as shown in the following theorem.

**Theorem 5.** Let $X_1, \ldots, X_N \overset{i.i.d.}{\sim} X$, where $X$ takes values in $L_2(Q)$ with $\mathbb{E}(\|X\|^4) < \infty$. Also assume that $\mathbb{E}(X) = 0$ and $\text{Cov}(X) = C$. Let $\tilde{C}_{R,N}^{sh}, \tilde{C}_{R,N}^{d}$ and $\tilde{C}_{R,N}^{ds}$ be the modified shallow, deep, and deepshared CovNet estimators given by (20). Let $R > d$, and $R \to \infty$, $\lambda \to \infty$ as $N \to \infty$.

(a) If $d^R \lambda_{N}^4 \log(R \lambda_N) / N \to 0$ as $N \to \infty$, then the modified shallow CovNet estimator is weakly consistent for $C$, that is, $\|\tilde{C}_{R,N}^{sh} - C\|_2 \overset{P}{\to} 0$. Additionally, if $d^R \lambda_{N}^4 \log(R \lambda_N) / N^{1-\delta} \to 0$ for some $\delta \in (0, 1)$, then it is strongly consistent for $C$, that is, $\|\tilde{C}_{R,N}^{sh} - C\|_2 \overset{a.s.}{\to} 0$ as $N \to \infty$.

(b) If $L^R \lambda_{N}^4 \log^2(L R \lambda_N) / N \to 0$ as $N \to \infty$, then the modified deep CovNet estimator is weakly consistent for $C$, that is, $\|\tilde{C}_{R,N}^{d} - C\|_2 \overset{P}{\to} 0$. Additionally, if $L^R \lambda_{N}^4 \log^2(L R \lambda_N) / N^{1-\delta} \to 0$ for some $\delta \in (0, 1)$, then it is strongly consistent for $C$, that is, $\|\tilde{C}_{R,N}^{d} - C\|_2 \overset{a.s.}{\to} 0$ as $N \to \infty$.

(c) If $L^R \lambda_{N}^4 \log^2(L R \lambda_N) / N \to 0$ as $N \to \infty$, then the modified deepshared CovNet estimator is weakly consistent for $C$, that is, $\|\tilde{C}_{R,N}^{ds} - C\|_2 \overset{P}{\to} 0$. Additionally, if $L^R \lambda_{N}^4 \log^2(L R \lambda_N) / N^{1-\delta} \to 0$ for some $\delta \in (0, 1)$, then it is strongly consistent for $C$, that is, $\|\tilde{C}_{R,N}^{ds} - C\|_2 \overset{a.s.}{\to} 0$ as $N \to \infty$.

**Remark 10.** Our derived rates are truly non-parametric, with minimal assumptions on the underlying structure. We assumed $\mathbb{E}(\|X\|^4) < \infty$, which is standard for covariance estimation. Moreover, we made no assumption on the underlying covariance operator $C$. As a
consequence, our derived rates are slow in terms of the number of parameters of the models. These can be improved by making further assumptions on the random field $\mathcal{X}$ or the eigenfunctions of $C$ (e.g., the ones used by Bauer & Kohler, 2019 or Schmidt-Hieber, 2020 in the context of non-parametric regression; see Remark 15 in the Online Supplement for details). However, such specialised treatments are beyond the scope of the present article. If the rank of $C$ is small, which is very often the case in FDA, then a small $R$ is enough to control the bias term (see Section B in the Online Supplement). On the other hand, such a small $R$ gives us a considerable gain in terms of the variance, thus reducing the overall estimation error. Nevertheless, one should also note that the derived rates are only upper bounds, and we do not claim tightness of the bounds.

6.2 The case of discretely observed fields

The results derived so far are for fully observed random fields. But in practice, we observe the fields on a grid, with possible noise contamination. Here, we develop asymptotic properties of our estimators in this scenario. W.l.o.g., we assume that $Q = [0, 1]^d$, and we observe the data on a $K_1 \times \cdots \times K_d$ regular grid on $[0, 1]^d$. To this extent, let $\{T_{1,1}^{K_1}, \ldots, T_{1,K_1}^{K_1}\}, \ldots, \{T_{d,1}^{K_d}, \ldots, T_{d,K_d}^{K_d}\}$ be regular partitions of $[0, 1]$ of sizes $K_1, \ldots, K_d$, respectively. Define $V^K_{i_1,\ldots,i_d} = T_{i_1}^{K_1} \times \cdots \times T_{i_d}^{K_d}$ to be the $(i_1, \ldots, i_d)$th voxel for $1 \leq i_1 \leq K_1, \ldots, 1 \leq i_d \leq K_d$. The voxels are non-overlapping (i.e., $V^K_{i_1,\ldots,i_d} \cap V^K_{j_1,\ldots,j_d} = \emptyset$ for $(i_1, \ldots, i_d) \neq (j_1, \ldots, j_d)$), and they form a regular partition of $[0, 1]^d$. In particular, $|V^K_{i_1,\ldots,i_d}| = 1/(\prod_{i=1}^d K_i)$. For each random field $\mathcal{X}_n$, we make a single measurement at each of the voxels. These measurements are assumed to be of the form

$$X^K_n[i_1, \ldots, i_d] = X^K_n[i_1, \ldots, i_d] + E^K_n[i_1, \ldots, i_d], \quad 1 \leq i_1 \leq K_1, \ldots, 1 \leq i_d \leq K_d, \quad n = 1, \ldots, N, \quad (21)$$

where $X^K_n[i_1, \ldots, i_d]$ is a discretisation of $\mathcal{X}_n$ over the $(i_1, \ldots, i_d)$-th voxel and $E^K_n[i_1, \ldots, i_d]$ is the corresponding measurement error or noise. We consider two different measurement schemes which relate the discrete object $X^K_n = (X^K_n[i_1, \ldots, i_d])$ to the respective field $\mathcal{X}_n = (X_n(u) : u \in [0, 1]^d)$.

(M1) Point-wise measurement:

$$X^K_n[i_1, \ldots, i_d] = X_n(u_{i_1}, \ldots, u_{i_d}), \quad 1 \leq i_1 \leq K_1, \ldots, 1 \leq i_d \leq K_d,$$

where $(u_{i_1}, \ldots, u_{i_d}) \in V^K_{i_1,\ldots,i_d}$ is a location within the $(i_1, \ldots, i_d)$th voxel. For the measurements to be meaningful, we need to assume that $\mathcal{X}$ has continuous sample paths (e.g., Hsing & Eubank, 2015).

(M2) Voxel-wise average:

$$X^K_n[i_1, \ldots, i_d] = \frac{1}{|V^K_{i_1,\ldots,i_d}|} \int_{V^K_{i_1,\ldots,i_d}} X_n(u) \, du, \quad 1 \leq i_1 \leq K_1, \ldots, 1 \leq i_d \leq K_d.$$

For the measurement errors $E^K_n[i_1, \ldots, i_d]$, we assume that they are i.i.d. with mean 0 and variance $\sigma^2_K$, and are uncorrelated with the $X^K_n[i_1, \ldots, i_d]$’s. In line with our previous assumptions, we also assume that $|E^K_n[i_1, \ldots, i_d]|^2 \leq \rho^2_{N,K}$ almost surely.
We denote the measurements corresponding to the $n$-th field $x_n$ by $\tilde{X}_n^K = (\tilde{X}_n^K[i_1, \ldots, i_d])$. Define $\tilde{C}_N = N^{-1} \sum_{n=1}^N \overline{X}_n^K \otimes \overline{X}_n^K$ to be the empirical covariance based on the discretely observed data. For a generic class of CovNet operators $\tilde{F}_R$, our estimator is given by

$$\hat{C}_{R,N}^K \in \text{argmin}_{\mathcal{C} \in \tilde{F}_R} \left\| \tilde{C}_N^K - \mathcal{C}^K \right\|_F^2,$$

where $\| \cdot \|_F$ is the Frobenius norm and $\mathcal{C}_N^K$ is the discretisation of the operator $\mathcal{C}$ over the voxels, defined as

$$G^K[i_1, \ldots, i_d; j_1, \ldots, j_d] = g(v_{i_1}, \ldots, v_{i_d}; v_{j_1}, \ldots, v_{j_d}), \quad 1 \leq i_1, j_1 \leq K_1, \ldots, 1 \leq i_d, j_d \leq K_d,$$

where $g$ is the kernel corresponding to $\mathcal{C}$ and $(v_{i_1}, \ldots, v_{i_d})$ is a location in the $(i_1, \ldots, i_d)$th voxel $V^K_{i_1, \ldots, i_d}$. If we define $\tilde{C}_N^K$ to be the voxel-wise continuation of $\tilde{C}_N^K$, with kernel

$$\tilde{C}_N^K(u, v) = \sum_{i_1=1}^{K_1} \cdots \sum_{i_d=1}^{K_d} \tilde{C}_N^K[i_1, \ldots, i_d; j_1, \ldots, j_d] 1\{u \in V^K_{i_1, \ldots, i_d}, v \in V^K_{j_1, \ldots, j_d}\}, u, v \in [0, 1]^d,$$

then it is easy to see that

$$\text{arg min}_{\mathcal{C} \in \tilde{F}_R} \left\| \tilde{C}_N^K - \mathcal{C}^K \right\|_F^2 \approx \text{arg min}_{\mathcal{C} \in \tilde{F}_R} \| \tilde{C}_N^K - \mathcal{C} \|_2^2,$$

where the approximation holds when the resolution $K_1 \times \cdots \times K_d$ is large (see also (47) in the Online Supplement). We will derive theoretical properties for this approximation. In particular, we define

$$(22) \quad \hat{C}_{R,N}^{\text{sh,K}} \in \text{arg min}_{\mathcal{C} \in \tilde{F}_{R,N}^{\text{sh}}} \| \tilde{C}_N^K - \mathcal{C} \|_2^2, \quad \hat{C}_{R,L,N}^{\text{d,K}} \in \text{arg min}_{\mathcal{C} \in \tilde{F}_{R,L,N}^{\text{d}}} \| \tilde{C}_N^K - \mathcal{C} \|_2^2 \quad \text{and} \quad \hat{C}_{R,L,N}^{\text{ds,K}} \in \text{arg min}_{\mathcal{C} \in \tilde{F}_{R,L,N}^{\text{ds}}} \| \tilde{C}_N^K - \mathcal{C} \|_2^2,$$

to be the shallow, deep and deepshared CovNet estimators based on the discrete measurements. The asymptotic behaviour of these estimators is established in the following theorem.

**Theorem 6.** Let $\mathcal{X}_1, \ldots, \mathcal{X}_N \overset{i.i.d.}{\sim} \mathcal{X}$, where $\mathcal{X}$ takes values in $\mathcal{L}_2([0, 1]^d)$ with $\mathbb{E}(\mathcal{X}) = 0$ and $\text{Cov}(\mathcal{X}) = C$. Suppose that the kernel $c$ of $C$ is Lipschitz on $[0, 1]^d$ with Lipschitz constant $\rho$. Consider the measurement model (21), where the measurement errors $E^K_n[i_1, \ldots, i_d]$ are i.i.d. and uncorrelated with $X^K_1, \ldots, X^K_N$, and satisfy $|E^K_n[i_1, \ldots, i_d]|^2 \leq \beta^2_N$ almost surely, $\mathbb{E}(E^K_n[i_1, \ldots, i_d]) = 0$, $\text{Var}(E^K_n[i_1, \ldots, i_d]) = \sigma^2_K$. Suppose that one of the following two hold.

(a) $\mathcal{X}$ has continuous sample paths, $\|\mathcal{X}\|_\infty^2 \leq \beta_N$ almost surely, and the measurements $X^K_1, \ldots, X^K_N$ are obtained from $\mathcal{X}_1, \ldots, \mathcal{X}_N$ via (M1).

(b) $\|\mathcal{X}\|_2^2 \leq \beta_N$ almost surely, and the measurements $X^K_1, \ldots, X^K_N$ are obtained from $\mathcal{X}_1, \ldots, \mathcal{X}_N$ via (M2).

Let $\hat{C}_{R,N}^{\text{sh,K}}, \hat{C}_{R,L,N}^{\text{d,K}}$ and $\hat{C}_{R,L,N}^{\text{ds,K}}$ be the shallow, deep and deepshared CovNet estimators given by (22). Then,
\[ ||\hat{\mathbb{C}}_{R,N}^{\text{sh},K} - C||_2^2 \leq 18 \inf_{\mathcal{G} \in \mathbb{P}_{\text{sh}}^{R,N}} ||\mathcal{G} - C||_2^2 + \mathcal{O}\left( \frac{dR^2\Delta^4_{N,K} \log(N)}{N} \right) + a_K, \]
\[ ||\hat{\mathbb{C}}_{R,L,N}^{d,K} - C||_2^2 \leq 18 \inf_{\mathcal{G} \in \mathbb{P}_{\text{d}}^{R,L,N}} ||\mathcal{G} - C||_2^2 + \mathcal{O}\left( \frac{L^4R^6\Delta^4_{N,K}\log^2(N)}{N} \right) + a_K, \text{ and} \]
\[ ||\hat{\mathbb{C}}_{R,L,N}^{\text{sh},K} - C||_2^2 \leq 18 \inf_{\mathcal{G} \in \mathbb{P}_{\text{sh}}^{R,L,N}} ||\mathcal{G} - C||_2^2 + \mathcal{O}\left( \frac{L^4R^6\Delta^4_{N,K}\log^2(N)}{N} \right) + a_K, \]

where \( \Delta_{N,K} = \max\{2(\beta_N + \beta_{N,K}^c), RL_N\} \) and \( a_K = 21\rho^2\left(K_1^{-2} + \cdots + K_d^{-2}\right) + 21\sigma^4_K/(K_1 \cdots K_d). \)

The theorem clearly shows the effect of grid size and noise contamination on the estimators. The rates are qualitatively the same as in Theorem 4, except \( \Delta_N \) is replaced by \( \Delta_{N,K} \) and a couple of terms depending on \((K_1, \ldots, K_d)\) are added. The term \( \Delta_{N,K} \) can be viewed as a noise contaminated version of \( \Delta_N \). If we assume that \( \beta_{N,K}^c = o(\beta_N) \) (which can be seen as assuming that the signal-to-noise ratio diverges), then \( \Delta_{N,K} \) is asymptotically equivalent to \( \Delta_N \). In this case, we can see a clear separation in the estimation error, one due to the sample size and the other due to the resolution. Among the remaining two terms, \( \rho^2(K_1^{-2} + \cdots + K_d^{-2}) \) arises due to the discretisation of the fields, while \( \sigma^4_K/(K_1 \cdots K_d) \) is due to noise contamination. Thus, for consistency of our estimators, we require \( (K_1^{-2} + \cdots + K_d^{-2}) \to 0 \), which is ensured if \( \min\{K_1, \ldots, K_d\} \to \infty \). Moreover, the noise level \( \sigma_K \) is allowed to diverge, but at a slower rate than \( (K_1 \cdots K_d)^{1/4} \). The Lipschitz assumption on \( c \) in the theorem is convenient, but is by no means necessary. It can be verified that consistency of the estimators holds as long as \( c \) is continuous.

7 | CONCLUDING REMARKS

We have proposed three new classes of neural network models for covariance estimation of functional data observed over multidimensional domains. The advantages of the proposed models include efficient estimation, storage, manipulation and performance guarantees. Our approach is motivated by the demonstrated ability of neural networks in solving complex problems. And indeed, our empirical studies show the superiority of the proposed methods, especially the deepshared CovNet model, which we advocate to use in practice. At the same time, our methods will also be amenable to the shortcomings of neural networks, for example, lack of theoretical optimisation guarantees for convergence to global minima. But, as is the case with neural networks, despite these limitations, our experimental results appear compelling. Any progress in the study of neural networks will, in principle, translate to a commensurate progress in the understanding of covariance networks.

Throughout the article, we have used the sigmoidal activation function. But, most of the results, especially the ones for the deep CovNet models, can be easily extended to include other activation functions, for example, the ReLU. In some preliminary numerical studies, we observed similar performance by the sigmoid and the ReLU. We prefer the sigmoid because of the smoothness that it provides, which is often beneficial for functional covariance estimation.

At the level of generality they are derived, our convergence rates are arguably slow. But, these do not reveal the complete picture and are rather a reflection of our completely non-parametric treatment of the problem. These rates can be improved by considering more structured problems, which is now a topic of interest in theoretical studies of neural networks (e.g., Bauer &
Kohler, 2019; Schmidt-Hieber, 2020). Such additional structural assumptions may also allow us to derive approximation errors for the models, which we have not fully addressed here.

In all our theoretical derivations, we have assumed that the observations $\mathcal{X}_1, \ldots, \mathcal{X}_N$ are i.i.d. A careful look at the proofs reveal that the results hold even when the observations are not identically distributed. However, the same is not true for the independence assumption, which has been used to derive concentration inequalities for averages and $V$-statistics. Extending these results to the dependent setup requires some rigour, which is beyond the scope of the current article.

Although our primary focus here was on the densely observed functional data case, the presented methodology can be adapted to sparse and irregular functional data as well. In particular, we can take the classical approach of first smoothing the observed fields and then working with these smoothed fields. A similar approach was taken by Masak et al. (2022), following whom it is also possible to derive theoretical guarantees for the modified estimator.

ACKNOWLEDGEMENTS

This research was supported by a Swiss National Science Foundation grant. Majority of the work was done while the first author was with the Institute of Mathematics, EPFL. We are grateful to Prof. Sir John A. D. Aston for providing access to the fMRI data and enlightening us on some of their key aspects. We also thank the Editor, an Associate Editor and two anonymous reviewers for their suggestions on an earlier version of the manuscript. Open access funding provided by École Polytechnique Fédérale de Lausanne.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available via the Functional Connectomes Project at https://www.nitrc.org/projects/fcon_1000/, as stated in the first paragraph of Section 5.3.

ORCID

Soham Sarkar https://orcid.org/0000-0003-0412-6826
Victor M. Panaretos https://orcid.org/0000-0002-2442-9907

REFERENCES

Adler, R.J. & Taylor, J.E. (2007) Random fields and geometry. New York: Springer.
Anthony, M. & Bartlett, P.L. (1999) Neural network learning: theoretical foundations. Cambridge: Cambridge University Press.
Aston, J.A. & Kirch, C. (2012) Evaluating stationarity via change-point alternatives with applications to fMRI data. The Annals of Applied Statistics, 6, 1906–1948.
Aston, J.A.D., Pigoli, D. & Tavakoli, S. (2017) Tests for separability in nonparametric covariance operators of random surfaces. The Annals of Statistics, 45, 1431–1461.
Bagchi, P. & Dette, H. (2020) A test for separability in covariance operators of random surfaces. The Annals of Statistics, 48, 2303–2322.
Bauer, B. & Kohler, M. (2019) On deep learning as a remedy for the curse of dimensionality in nonparametric regression. The Annals of Statistics, 47, 2261–2285.
Baydin, A.G., Pearlmutter, B.A., Radul, A.A. & Siskind, J.M. (2018) Automatic differentiation in machine learning: a survey. Journal of Machine Learning Research, 18, 1–43.
Bengio, Y. (2012) Practical recommendations for gradient-based training of deep architectures. In: Neural Networks: Tricks of the Trade, 2nd edition. Berlin, Heidelberg: Springer, pp. 437–478.
Bowring, A., Telschow, F.J., Schwartzman, A. & Nichols, T.E. (2021) Confidence Sets for Cohen’s $d$ effect size images. *NeuroImage*, 226, 117477.

Buduma, N. & Locascio, N. (2017) *Fundamentals of deep learning: Designing next-generation machine intelligence algorithms*. Sebastopol, CA: O’Reilly Media, Inc.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B. & LeCun, Y. (2015) The loss surfaces of multilayer networks. In: *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 38, 192–204.

Constantinou, P., Kokoszka, P. & Reimherr, M. (2017) Testing separability of space-time functional processes. *Biometrika*, 104, 425–437.

Dette, H., Dierickx, G. & Kutta, T. (2022) Quantifying deviations from separability in space-time functional processes. *Bernoulli*, 28, 2909–2940.

Dick, J., Kuo, F.Y. & Sloan, I.H. (2013) High-dimensional integration: the quasi-Monte Carlo way. *Acta Numerica*, 22, 133–288.

Eldan, R. & Shamir, O. (2016) The power of depth for feedforward neural networks. In: *29th Annual Conference on Learning Theory*, PMLR, 49, 907–940.

Gneiting, T., Genton, M.G. & Guttorp, P. (2006) Geostatistical space-time models, stationarity, separability, and full symmetry. In: *Statistical methods for spatio-temporal systems*. Boca Raton: Chapman & Hall/CRC Press, pp. 151–175.

Golub, G.H. & Van Loan, C.F. (2013) *Matrix computations*, 4th edition. Baltimore, MD: Johns Hopkins University Press.

Györfi, L., Kohler, M., Krzyżak, A. & Walk, H. (2002) *A distribution-free theory of nonparametric regression*. New York: Springer-Verlag.

Hsing, T. & Eubank, R. (2015) *Theoretical foundations of functional data analysis, with an introduction to linear operators*. Hoboken: John Wiley & Sons.

Kaipio, J. & Somersalo, E. (2005) *Statistical and computational inverse problems*. New York: Springer-Verlag.

Kingma, D.P. & Ba, J. (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Langer, S. (2021) Approximating smooth functions by deep neural networks with sigmoid activation function. *Journal of Multivariate Analysis*, 182, 104696.

Liang, S. & Srikant, R. (2017) Why deep neural networks for function approximation? In: *Proceedings of the 5th International Conference on Learning Representations*.

Masak, T., Sarkar, S. & Panaretos, V.M. (2022) Separable expansions for covariance estimation via the partial inner product. *Biometrika*, asac035. [https://doi.org/10.1093/biomet/asac035](https://doi.org/10.1093/biomet/asac035)

Mhaskar, H.N. (1996) Neural networks for optimal approximation of smooth and analytic functions. *Neural Computation*, 8, 164–177.

Ohn, I. & Kim, Y. (2019) Smooth function approximation by deep neural networks with general activation functions. *Entropy*, 21, 627.

Pigoli, D., Hadjipantelis, P.Z., Coleman, J.S. & Aston, J.A. (2018) The statistical analysis of acoustic phonetic data: exploring differences between spoken romance languages. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67, 1103–1145.

Pinkus, A. (1999) Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8, 143–195.

Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B. & Liao, Q. (2017) Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *International Journal of Automation and Computing*, 14, 503–519.

Raemaekers, M., Schellekens, W., Petridou, N. & Ramsey, N.F. (2018) Knowing left from right: a symmetric functional connectivity during resting state. *Brain Structure & Function*, 223, 1909–1922.

Ramsay, J.O. & Silverman, B.W. (2002) *Applied functional data analysis: methods and case studies*. New York: Springer-Verlag.

Rasmussen, C.E. & Williams, C.K.I. (2006) *Gaussian processes for machine learning*. *Adaptive Computation and Machine Learning*. Cambridge, MA: MIT Press.

Rougier, J. (2017) A representation theorem for stochastic processes with separable covariance functions, and its implications for emulation. *arXiv preprint arXiv:1702.05599*.

Schmidt-Hieber, J. (2020) Nonparametric regression using deep neural networks with RELU activation function. *The Annals of Statistics*, 48, 1875–1897.
Sommerfeld, M., Sain, S. & Schwartzman, A. (2018) Confidence regions for spatial excursion sets from repeated random field observations, with an application to climate. *Journal of the American Statistical Association*, 113, 1327–1340.

Stoehr, C., Aston, J.A.D. & Kirch, C. (2021) Detecting changes in the covariance structure of functional time series with application to fMRI data. *Econometrics and Statistics*, 18, 44–62.

Telschow, F.J. & Schwartzman, A. (2022) Simultaneous confidence bands for functional data using the Gaussian Kinematic formula. *Journal of Statistical Planning and Inference*, 216, 70–94.

Vandekar, S.N., Satterthwaite, T.D., Xia, C.H., Adebimpe, A., Ruparel, K., Gur, R.C. et al. (2019) Robust spatial extent inference with a semiparametric bootstrap joint inference procedure. *Biometrics*, 75, 1145–1155.

Wang, J., Wong, R.K.W. & Zhang, X. (2022) Low-rank covariance function estimation for multidimensional functional data. *Journal of the American Statistical Association*, 117, 809–822.

Wang, J.-L., Chiou, J.-M. & Müller, H.-G. (2016) Functional data analysis. *Annual Review of Statistics and Its Application*, 3, 257–295.

Wang, S., Cao, G. & Shang, Z. (2021) Estimation of the mean function of functional data via deep neural networks. *Stat*, 10, e393 For the Alzheimer’s Disease Neuroimaging Initiative.

**SUPPORTING INFORMATION**

Additional supporting information may be found in the online version of the article at the publisher’s website.

**How to cite this article:** Sarkar, S. & Panaretos, V.M. (2022) CovNet: Covariance networks for functional data on multidimensional domains. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(5), 1785–1820. Available from: [https://doi.org/10.1111/rssb.12551](https://doi.org/10.1111/rssb.12551)