National Institute for Applied Statistics Research Australia

University of Wollongong, Australia

Working Paper

07-23

Statistical Deep Learning for Spatial and Spatio-Temporal Data

Christopher K. Wikle and Andrew Zammit-Mangion

National Institute for Applied Statistics Research Australia, University of Wollongong, Wollongong NSW 2522, Australia T: +61 2 42215076. E: karink@uow.edu.au

Statistical Deep Learning for Spatial and Spatio-Temporal Data

Christopher K. Wikle[1], and Andrew Zammit-Mangion[2]

[1] Department of Statistics, University of Missouri, MO
[2] School of Mathematics and Applied Statistics, University of Wollongong, Australia

Abstract

Deep neural network models have become ubiquitous in recent years, and have been applied to nearly all areas of science, engineering, and industry. These models are particularly useful for data that have strong dependencies in space (e.g., images) and time (e.g., sequences). Indeed, deep models have also been extensively used by the statistical community to model spatial and spatio-temporal data through, for example, the use of multi-level Bayesian hierarchical models and deep Gaussian processes. In this review, we first present an overview of traditional statistical and machine learning perspectives for modeling spatial and spatio-temporal data, and then focus on a variety of hybrid models that have recently been developed for latent process, data, and parameter specifications. These hybrid models integrate statistical modeling ideas with deep neural network models in order to take advantage of the strengths of each modeling paradigm. We conclude by giving an overview of computational technologies that have proven useful for these hybrid models, and with a brief discussion on future research directions.

Keywords— Bayesian hierarchical models; convolutional neural networks; deep Gaussian processes; recurrent neural networks; reinforcement learning; warping

1 Introduction

Deep learning has revolutionized prediction and classification for many types of dependent data. These dependencies are often temporal and spatial in nature, and state-of-the-art modifications of recurrent and convolutional neural networks have been especially adept at accounting for this structure, at least when there are copious amounts of training data. Deep learning models have therefore been extensively used, with great success, for natural language processing and image classification (see the sidebar titled The Deep Learning Revolution). For the most part, these models have been developed independently of models in statistics to account for such dependencies (e.g., Gaussian process (GP) models, and dynamic models; see Wikle et al. (2019a) for a recent overview of spatio-temporal statistical models). At first glance, the neural network models used in machine learning and the statistical approaches for spatio-temporal data may seem quite distinct. However, in complex data applications, both approaches tend to rely on multi-level (hierarchical) representations, with the primary differences being in the types of applications for which they are used, and in the fact that the statistical approaches for multi-level models tend to accommodate uncertainty quantification more naturally within a probabilistic framework.

There has been an increasing number of works in recent years, especially in the statistics literature, that take a hybrid approach to modeling complex spatial or spatio-temporal data. These hybrid models are predominantly classical hierarchical statistical models that borrow some of the effective ideas from deep neural models in order to

facilitate modeling the data, process, and/or parameter components that form the hierarchy. These hybrid models have led to many notable advances in statistical spatial/spatio-temporal modeling and inference. For example:

- Sidén & Lindsten (2020) use deep learning to construct a flexible family of Gaussian Markov random fields, subsequently used to model and predict land surface temperatures,[1] while Chen et al. (2021) integrate deep learning with the basis function approach to modeling spatial processes to predict $PM_{2.5}$ concentrations across the United States. Both methods are shown to be superior to more classical approaches to spatial prediction.

- Lenzi et al. (2021) use deep learning to estimate parameters with statistical spatial models of extremes, whose likelihood is intractable or difficult to evaluate. They apply their methods for efficiently estimating parameters in a Brown–Resnick process fitted to surface temperature data.[2] Zammit-Mangion & Wikle (2020) show how deep learning methods can be used to estimate spatially and temporally varying dynamics of a statistical spatio-temporal model, and provide uncertainty quantification on these dynamics, at a fraction of the computational cost that would typically be required. They apply their methods to efficiently forecast sea-surface temperature, using data available from the Copernicus Marine Environment Monitoring Service (CMEMS),[3] and for nowcasting rainfall, using radar reflectivity data available from Wikle et al. (2019b).

- McDermott & Wikle (2017) show how spatio-temporal echo state networks (ESNs) can be used to produce long-lead forecasts and uncertainty quantification of tropical Pacific Ocean sea surface temperature (STT) anomalies that suggest developing El Niño or La Niña conditions. The SST data are available from Wikle et al. (2019b). McDermott & Wikle (2019b) also show how deep versions of ESNs can be used for long-lead spatio-temporal prediction of soil moisture over the "corn belt" region of the U.S. given SST data in previous months. They use SST data from the extended reconstruction SST (ERSST) data set[4] and soil moisture data from the Climate Prediction Center's high resolution monthly global soil moisture data set.[5] ESNs can also be used effectively for short-term forecasts of complex processes. Huang et al. (2021a) use a spatio-temporal ESN to forecast short-term wind speeds for power production over Saudi Arabia with high-resolution wind fields.[6]

The above are just some of the highlights that will be expanded on in this review in Sections 3 and 4, following an overview of machine learning and statistical approaches for spatial and spatio-temporal data in Section 2. The review discusses technologies that enable deep learning for spatial and spatio-temporal data in Section 5, and concludes with a brief discussion of future research directions in Section 6.

2 Conventional statistical and machine learning approaches for spatial and spatio-temporal data

We begin the review with a brief overview of some of the key statistical and machine learning approaches for analyzing spatial and spatio-temporal data.

[1] https://github.com/finnlindgren/heatoncomparison
[2] https://ldas.gsfc.nasa.gov/nldas/v2/models
[3] https://marine.copernicus.eu/
[4] http://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCDC/.ERSST/
[5] https://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCEP/.CPC/.GMSM/.w/
[6] https://github.com/hhuang90/KSA-wind-forecast

The deep learning revolution

A deep learning model is a statistical or machine learning model that encompasses multiple connected components to classify or predict with complex data sets. There have been remarkable success stories reported in popular media outlets about how deep learning algorithms have mastered complex games such as Go, Chess, or Shogi. More generally, these methods have proven exceptionally effective at classifying image and sequence (e.g., natural language) data because they are particularly suited for learning patterns in complex data. Indeed, deep learning models are today at the heart of many devices and applications that are in everyday use, such as smart phone applications, web searching, and advertising, to name just a few. Although these models have proven useful, they are not without their flaws. For example, they typically require a great deal of training data and computational resources to train, and they have been known to fail spectacularly on occasion. Perhaps more troubling is that most deep learning algorithms are "black boxes" and it is difficult to know why they are producing the prediction or classification that they do, and there is usually no measure of uncertainty associated with those outputs. The first issue is problematic as it can lead to models that are learning unknown, and possibly unpredictable, biases. The second issue is problematic when one is attempting to use the output from these models to make decisions – it is challenging to make decisions based on model output without knowing how reliable the output is. For these reasons, recent years have seen an increased and sustained effort by machine learners and statisticians to understand these models, produce uncertainties for model outputs, and combine these deep neural methods with more traditional multi-layer statistical models.

2.1 Statistical approaches for spatial data

Statistical methods for spatial data are well summarized in a variety of monographs; see, for example, Cressie (1993) and Banerjee et al. (2014). These methods have historically been grouped by the support of the data they are used to model. *Geostatistical* spatial methods are generally used with responses that are available at a set of point-referenced locations in a domain of interest G, where we term G the "geographic domain". *Areal* or *lattice* spatial methods are often used with responses that are defined over a finite number of (generally non-overlapping) partitions of G. *Spatial point process* methods are used when the data are a finite set of random locations in G (indicative of presence/absence). Other types of spatial data include random sets and trajectories.

Assume we can write our random spatial process as

$$Y(\mathbf{s}) = f(\mathbf{s};\boldsymbol{\beta}) + \eta(\mathbf{s}), \quad \mathbf{s} \in G, \tag{1}$$

where $f(\cdot\,;\boldsymbol{\beta})$ is the conditional mean of $Y(\cdot)$ that contains covariate relationships and associated fixed effects $\boldsymbol{\beta}$ (e.g., $f(\cdot\,;\boldsymbol{\beta}) = \mathbf{x}'(\cdot)\boldsymbol{\beta}$, where $\mathbf{x}(\cdot)$ is a set of known covariates or "features"), and $\eta(\cdot)$ is the spatially-dependent random process. In the context of geostatistical models, $\eta(\cdot)$ is often modeled as a Gaussian process (GP). A GP is a dependent process where all of its finite-dimensional distributions are Gaussian and defined through a mean function $\mu(\cdot)$ and a covariance function $C(\mathbf{s},\tilde{\mathbf{s}}) = \text{cov}(Y(\mathbf{s}), Y(\tilde{\mathbf{s}}))$ where $\mathbf{s},\tilde{\mathbf{s}} \in G$. A GP $\eta(\cdot)$ with mean $\mu(\cdot)$ and covariance function $C(\cdot,\cdot)$ is denoted as $\eta(\cdot) \sim \text{GP}(\mu(\cdot), C(\cdot,\cdot))$. In geostatistical applications, it is typically the case that $\mu(\cdot) = 0$ since the conditional mean is accounted for by $f(\cdot;\boldsymbol{\beta})$ (note, this implies that $Y(\cdot) \sim \text{GP}(f(\cdot;\boldsymbol{\beta}), C(\cdot,\cdot))$).

Perhaps the biggest challenge in implementing GP-based spatial prediction has to do with the covariance function, $C(\cdot,\cdot)$. In practice, this is not known, and one must

typically assume stationarity (intrinsic or second-order) and often isotropy (directional invariance) to proceed. In addition, the functional form of the stationary covariance matrix is often specified (e.g., Gaussian, Matérn, power) up to some unknown parameters, $\boldsymbol{\theta}_y$ say. Even in these cases, the form of the optimal predictor requires that one evaluates the inverse of the covariance matrix associated with all observation locations, and this can be problematic in high-data-volume problems. Much of the research in spatial statistics over the last decade has been concerned with developing methods for such high-data-volume prediction problems; these approaches have tended to fall into two categories – neighbor-based methods and basis function approaches (see Heaton et al., 2019, for a recent overview).

Covariance stationarity: a covariance function $C(\mathbf{s}, \cdot), \mathbf{s} \in G$, is stationary if it only depends on the displacement from $\mathbf{s}$, that is, if one can write $C(\mathbf{s}, \mathbf{u}) = C^o(\mathbf{h})$ for any $\mathbf{s}, \mathbf{u} \in G$, where $\mathbf{h} \equiv \mathbf{s} - \mathbf{u}$

Covariance isotropy: a covariance function $C(\mathbf{s}, \cdot), \mathbf{s} \in G$, is isotropic if it only depends on the distance from $\mathbf{s}$, that is, if one can write $C(\mathbf{s}, \mathbf{u}) = C^d(\mathbf{s}, d)$ for any $\mathbf{s}, \mathbf{u} \in G$, where $d \equiv \|\mathbf{s} - \mathbf{u}\|$

Gaussian Markov random field: a collection of variables that are jointly Gaussian, and where one or more variables is conditionally independent of others when conditioned on variables in a neighboring set

Hierarchical Model: a model constructed via a series of conditional distributions and marginal distributions (e.g., $[A, B, D] = [D \mid A, B][A \mid B][B]$)

Bayesian Hierarchical Model (BHM): a hierarchical model where Bayes' rule is used to make inference on all unknown quantities given data. For example, if D represents data, $[A, B|D] \propto [D|A, B][A|B][B]$

When dealing with areal or lattice data, the random component in Equation 1 is typically modeled as a Gaussian Markov random field (MRF). In this case, one often is interested in making inference on the conditional mean parameters, or in smoothing noisy areal observations (e.g., as in disease mapping). Gaussian MRFs lead to a highly structured and yet parsimonious sparse precision matrix, which can facilitate computation in marginal formulations or conditional (Bayesian) implementations.

Regardless of whether one treats $Y(\cdot)$ from a GP or an MRF perspective, it is best to treat it as a latent process, which is only partially observed via a finite set of m, say, spatial observations, $\mathbf{Z} \equiv (Z_i : i = 1, \ldots, m)'$, where each Z_i is an observation made at $\mathbf{r}_i \in G$, or averaged over $\mathbf{r}_i \subset G, i = 1, \ldots, m$. Then, a model can be specified for the observations conditional on this latent process, $[\mathbf{Z} \mid Y(\cdot), \boldsymbol{\theta}_z]$ say, where the brackets $[\,\cdot\,]$ denote a probability distribution, and $\boldsymbol{\theta}_z$ here denotes parameters that parameterize the conditional distribution of the data. As in generalized linear mixed models, this model easily accommodates non-Gaussian observations as well as measurement error. Although the parameters associated with the data model and latent process model, $\{\boldsymbol{\theta}_z, \boldsymbol{\theta}_y\}$, can be estimated through likelihood methods, it is often the case that one specifies prior distributions for these parameters and considers a Bayesian implementation (e.g., see Cressie & Wikle, 2011, Banerjee et al., 2014). This leads to a multi-level Bayesian hierarchical model (BHM):

$$\begin{aligned} \text{Data Model:} \quad & [\mathbf{Z} \mid Y(\cdot), \boldsymbol{\theta}_z], \\ \text{Process Model:} \quad & [Y(\cdot) \mid f(\cdot\,; \boldsymbol{\beta}), \boldsymbol{\theta}_y], \\ \text{Parameter Models:} \quad & [\boldsymbol{\theta}_z, \boldsymbol{\theta}_y, \boldsymbol{\beta}]. \end{aligned}$$

This BHM formulation is important as each model component is easily expanded to account for more complexity such as multiple data sources, multivariate processes, and parameters that are themselves processes (e.g., spatially-varying fixed effects). Such multi-level models are very much "deep" models as we see in the remainder of the review.

2.2 Statistical approaches for spatio-temporal data

Statistical methods for spatio-temporal data are extensively described in the monographs of Cressie & Wikle (2011) and Wikle et al. (2019a). Such methods can also be classified according to the spatial support of the data they are used to analyze (i.e., geostatistical, lattice, point process, etc.) and primarily differ from their spatial counterparts through the inclusion of a time index, which is assumed to come from a discrete set or a continuum. We represent the spatio-temporal process as $\{Y(\mathbf{s}; t) : \mathbf{s} \in G, t \in \mathcal{T}\}$ where t indexes the temporal domain $\mathcal{T} \subset \mathbb{R}^1$. As with the purely spatial process, we can consider either a GP or an MRF representation for the process $Y(\mathbf{s}; t) = f(\mathbf{s}, t; \boldsymbol{\beta}) + \eta(\mathbf{s}; t)$. Now the dependence is given by a spatio-temporal covariance function, say, $C(\mathbf{s}, t; \tilde{\mathbf{s}}, \tilde{t}) \equiv \text{cov}(Y(\mathbf{s}; t), Y(\tilde{\mathbf{s}}, \tilde{t}))$. In the GP case, the same covariance specification challenges occur as with the purely spatial case in terms of stationarity and high dimensionality (both of which are more challenging in

the spatio-temporal case). The spatio-temporal case is further complicated by the difficulty in specifying realistic joint covariances. For this reason, and to facilitate computation, spatio-temporal covariance functions are often assumed to be separable, that is, $C(\mathbf{s}, t; \tilde{\mathbf{s}}, \tilde{t}) = C_s(\mathbf{s}, \tilde{\mathbf{s}}) \cdot C_t(t, \tilde{t})$. The extra complexity of spatio-temporal data make the multi-level BHM representation even more appealing in practice.

The GP approach to spatio-temporal modeling is often called a "descriptive" approach since it does not explicitly account for the mechanisms that generated the data. The alternative paradigm is a "dynamical" approach, which specifies models describing the evolution of the spatial process with time, thereby attempting to follow the etiology of the data generating mechanism. Dynamic process models typically make a Markovian assumption in time; for example, for a discrete-time spatio-temporal model with unit time intervals, a first-order Markov assumption states that the process at time $t+1$ is independent of the process at times $t-1, t-2, \ldots$ given the process at time t. Perhaps the most used dynamic spatio-temporal model (DSTM) is based on the *integro-difference equation* (e.g., Wikle et al., 2019a, Chapter 5):

$$Y_{t+1}(\mathbf{s}) = \int_G k(\mathbf{s}, \mathbf{r}; \boldsymbol{\theta}_{k,t}) Y_t(\mathbf{r}) \mathrm{d}\mathbf{r} + \eta_t(\mathbf{s}), \quad \mathbf{s} \in G, \tag{2}$$

where $t = 1, 2, \ldots,$ denotes discrete time (note, we typically use subscripts to index time with discrete-time processes), G is the spatial domain over which the process is evolving, $k(\cdot, \cdot; \boldsymbol{\theta}_{k,t})$ is the mixing or transition kernel, $\{\boldsymbol{\theta}_{k,t}\}$ are temporally-evolving parameters appearing in the mixing kernel, and $\eta_t(\cdot)$ is an additive, Gaussian, spatial disturbance (typically with mean zero and temporally uncorrelated). Such models can easily be parameterized to incorporate mechanistic spatio-temporal behavior (e.g., diffusion and advection), and when discretized in space become vector autoregressive processes. More complex spatio-temporal dynamics can be accommodated by considering nonlinear dynamic models (such as quadratic nonlinear dynamic models; see Wikle & Hooten, 2010). As with the GP and MRF approaches, the DSTM can be used with non-Gaussian observations, in which case the dynamical process is treated as latent. It is natural to use the multi-level BHM framework to specify these models. Wikle (2019) presents a prototypical "deep" DSTM that includes layers for the data model, the conditional mean, the process model (composed of a dynamic and non-dynamic component), the dynamic process model, the non-dynamic process model, the prior distributions (that act as regularizers), and the hyperprior distributions (for a total of seven levels). This depth is not unusual in complex spatio-temporal data applications.

2.3 Machine learning/ai methods for spatial and spatio-temporal data

Many major success stories in deep learning involve spatially dependent data (e.g., image classification) and sequence data (e.g., natural language processing and time series forecasting). While classical multilayer perceptrons have been used for spatial prediction (e.g., Wang et al., 2019), recent successes owe much to the use of structured deep networks, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), which have architectures particularly suited for the problem at hand. Fan et al. (2021) present a comprehensive tutorial overview of these and other classical deep neural models from a statistician's perspective. Not surprisingly, soon after the development of these effective CNN and RNN models, they were used to model spatio-temporal data with slight modifications (e.g., Wang et al., 2016). More elaborate variants of the vanilla neural networks, such as autoencoders, generative adversarial networks, tensor networks, sequence-to-sequence networks, and graph neural networks, to name a few, have also been used to model spatio-temporal data (e.g., Oh

et al., 2015, Shi et al., 2015, Yu et al., 2017, Bai et al., 2019, Guo et al., 2019, Song et al., 2020).

As powerful as these artificial intelligence (AI) spatio-temporal methods are, they can present limitations in the context of statistical modeling for spatial and spatio-temporal data. For example, there is often substantial uncertainty in these types of data, including data gaps (as with satellite data), spatial or temporal supports that are at odds with the desired prediction supports, and substantial sampling and measurement uncertainty. Traditional deep model implementations do not directly provide model-based estimates of prediction and/or classification error that can account for these sources of error, nor can they easily incorporate or enforce known mechanistic relationships that are often present in spatio-temporal data. Further, given that these methods are essentially complex black-boxes, they are unable to perform inference or even provide guidance as to which inputs are important in explaining/predicting the response. Reichstein et al. (2019) gives an insightful in-depth discussion of many of these issues.

The AI community is, however, rising to these challenges. Increased interest in uncertainty quantification and explainability is the impetus behind the so-called "explainable AI (XAI)" (e.g., Gunning et al., 2019) and "interpretable AI" (Rudin et al., 2022) movements. In the context of uncertainty quantification (UQ), some success has been reported through four key approaches: variational Bayesian inference, Monte Carlo dropout, mixture density networks, and ensemble techniques (see Abdar et al., 2021, for an extensive overview and references). In the context of explainability, the main aim is to ensure that models are transparent so that deep learning models do not include unanticipated biases. Approaches to increase explainability include the use of interpretable local surrogates (such as the Local Interpretable Model-Agnostic Explanations (LIME) approach), occlusion analysis (e.g., Shapley values, SHapley Additive exPlanations (SHAP), Kernel SHAP, meaningful perturbation), integrated gradients (e.g., SmoothGrad), and layerwise relevance propagation. There are also self-explainable models (e.g., that utilize attention mechanisms) and specialized models that allow interpretability (e.g., graph neural networks). Molnar (2022) and Samek et al. (2021) provide definitions and comprehensive overviews of these methods. Many of the favored explainability approaches are model agnostic, meaning that they can be applied to essentially any predictive/classification model regardless of architecture (e.g., interpretable local surrogates and Shapley values) and could likely be used more broadly in statistical modeling.

The ubiquitous CNN-based methods used for image-type data are often not appropriate for continuous spatial processes, which are of most concern in "geostatistical" applications. These applications typically require optimal interpolation, and thus require the ability to predict at any location in space and provide a model-based assessment of uncertainty. The long-standing optimal linear prediction approach to this problem is kriging (and its variants), which is based on Gaussian process theory (e.g., see Cressie, 1993). As discussed below in Section 3, many of the hybrid statistical deep-learning approaches that afford UQ and continuous prediction are based on deep GPs. A recent alternative that lies somewhere in-between the hybrid GP approaches and the CNN image approach is given by Kirkwood et al. (2020). Here, they consider non-linear functions of gridded covariates and point-level information (location/elevation data) in a CNN, and Monte Carlo dropout to do spatial prediction at any location in the domain of interest. The modeled output is the mean and variance of a normal distribution (as in the mixture density network approach to output uncertainty), but they also use Monte Carlo dropout as a Bayesian approximation (Gal & Ghahramani, 2016) to account for model uncertainty. Alternatively, Amato et al. (2020) decompose spatio-temporal processes in terms of temporal basis functions and stochastic spatial coefficients. This is a common approach in spatial and spatio-temporal statistics (see Wikle et al., 2019a), but the difference here is that the model for the spatial coeffi-

Convolutional Neural Network (CNN): a deep neural network that takes image inputs and that learns shared filters that are convolved across the image to feed into the next layer; primarily used in image classification and computer vision for learning important multi-resolution features at each hidden layer

Recurrent Neural Network (RNN): a deep neural network that takes sequential or time series data as input, and that learns patterns in the data while accounting for "memory"; primarily used for natural language processing (e.g., translation, speech recognition) and time-series forecasting

Autoencoder: a deep learning architecture that consists of an encoder segment that finds a low dimensional representation of the data, and a decoder segment that reconstructs the data from its low dimensional representation.

Local Interpretable Model-Agnostic Explanations (LIME): an approach to model explainability that uses local (usually linear) surrogate models to explain the features that most influence individual predictions

Shapley Values: developed in game theory to assign credit to players, depending on their contribution to the total payout from a game; now also used in AI to quantify which features are most important for a predicted response

cients is specified in terms of a set of regressions based on spatial covariates, which are then trained via a deep feedforward neural network.

In the context of spatio-temporal modeling, as mentioned in Section 2.1, one can take a GP approach as well, but it is often more realistic to consider a dynamic model. The aforementioned hybrid CNN/RNN models can do this, albeit typically with no UQ, little interpretability, and no restrictions to enforce known mechanistic (physical/biological) relationships. Regarding the latter, there has recently been a flurry of interest in enforcing such restrictions in deep models. Generally, one can encourage known dynamical constraints by adding an appropriate penalization term to the objective function and then proceeding as usual using stochastic gradient descent (e.g., Raissi et al., 2020, Wu et al., 2020, Momenifar et al., 2022). Such approaches amount to a "soft constraint" and the solutions are not guaranteed to be physically consistent. This can present a problem in applications that require certain balance relations (e.g., continuity, conservation of mass, etc.). Recently, there has been progress in enforcing "hard" mechanistic constraints in deep models. For example, Mohan et al. (2020) consider a two-stage model in which the first stage uses an unconstrained CNN-type model to obtain a potential surface. This surface is then fed into an untrained physical model that performs appropriate transformations of the potential surface to obtain quantities of interest (e.g., velocities) in a manner that is physically consistent. Other approaches, which take multiple model types and connect them such that physical constraints are enforced in one "physics" model component, appear in Reichstein et al. (2019), Chattopadhyay et al. (2022), and Huang et al. (2021c). This work is quite promising, but has yet to be integrated into a framework that provides model-based UQ and explainability.

3 Hybrid deep learning spatial and spatio-temporal hierarchical models

The models in Section 2 are well-suited for some applications involving spatial/spatio-temporal data where prediction is the end goal and where model interpretability and uncertainty quantification have a lower priority. In applications were the latter two properties of the model and the predictions are important, the fusion of deep learning methods with classical statistical models offers an attractive way forward. Such approaches generally involve first constructing classical spatial and spatio-temporal probability models as outlined in Sections 2.1 and 2.2, and then integrating deep learning for characterizing some, or all, of the conditional distributions. The resulting models inherit the best of both worlds: the interpretability and uncertainty quantification properties commonly associated with statistical models, and the complexity of input/output relationships commonly associated with deep learning models. In this section we look at several ways in which deep learning has been used to some success in this way.

3.1 Deep learning for characterizing complex process models

Although the archetypal spatial/spatio-temporal hierarchical model consists of three layers, it is often the process model that is the hardest to characterize: the process model generally embodies physical, chemical, ecological or biological principles that are often simplified for analytical or computational tractability. On the other hand, measurement processes are often well understood, while parameter models generally embody expert judgment on low-dimensional quantities that are relatively straightforward to construct once correctly elicited. It is therefore not a surprise that most effort in this area has focussed on integrating deep learning in the process model.

Generally, deep learning models are treated as "black-box" models that map the

inputs (in this context, the spatial or spatio-temporal coordinates) to the outputs (observed quantities). Such models (e.g., Calandra et al., 2016) are rarely seen in the statistics literature, largely because the low-dimensional setting typically encountered in spatial applications creates some challenges that effectively preclude their use; see Duvenaud et al. (2014) and Dunlop et al. (2018) for insights. As a result, deep learning models in spatial and spatio-temporal statistics tend to contain quite a lot of structure, as we show in this sub-section.

3.1.1 Warping space to model nonstationary covariances

We first consider the most direct way to incorporate deep structures in spatial statistical models: that of applying a (structured) deep learning model on the spatial coordinates themselves.

Consider, for ease of exposition, a mean zero spatial process $Y(\mathbf{s}), \mathbf{s} \in G$, where the "geographic domain" $G \subset \mathbb{R}^2$. As discussed in Section 2.1, typically $Y(\cdot)$, is assumed to be Gaussian and stationary. The property of covariance stationarity can be relaxed by adopting the "deformation" approach of Sampson & Guttorp (1992), whereby one first formulates a "warping function" $\mathbf{f} : G \to D, D \subset \mathbb{R}^2$, and then assumes the relationship,

$$\text{cov}(Y(\mathbf{s}), Y(\mathbf{u})) \equiv C_G(\mathbf{s}, \mathbf{u}) = C_D^o(\|\mathbf{f}(\mathbf{s}) - \mathbf{f}(\mathbf{u})\|),$$

for $\mathbf{s}, \mathbf{u} \in G$, where $C_G(\cdot, \cdot)$ is a nonstationary covariance function on G, and $C_D^o(\cdot)$ is a stationary covariance function on the "warped domain", D.

Various approaches have been used for modeling $\mathbf{f}(\cdot)$: Sampson & Guttorp (1992) represented $\mathbf{f}(\cdot)$ using smoothing splines, Smith (1996) used basis functions derived from thin-plate splines, Snoek et al. (2014) used beta cumulative density functions, and Schmidt & O'Hagan (2003) used a bivariate Gaussian process. In a deep learning setting, one expresses the warping function as the composition $\mathbf{f}(\cdot) = \mathbf{f}_{n-1} \circ \cdots \circ \mathbf{f}_1(\cdot)$, where $\mathbf{f}_l(\cdot), l = 1, \ldots, n-1$, are in themselves simple, elementary functions. To our knowledge, the first to adopt this approach in a spatial statistics setting were Perrin & Monestiez (1999). Their approach was subsequently coined the "deep compositional spatial model" and extended to include other elementary warping functions, technologies often used in deep learning (see Section 5), and spatio-temporal and multivariate dependencies, by Zammit-Mangion et al. (2021), Vu et al. (2022b), and Vu et al. (2022c). These approaches based on function composition tend to ensure that the function $\mathbf{f}(\cdot)$ is injective by constraining each of the elementary functions to be itself injective. Injectivity guarantees that space does not fold on itself after warping, which is often seen as undesirable or unphysical in many spatial and spatio-temporal applications. A downside of this constraint is that the elementary warping functions need to be highly structured for injectivity to be ensured, and can limit the type of warpings that can be constructed. There are "black-box" deep learning architectures known as "normalizing flows" that are injective maps by design (e.g., Rezende & Mohamed, 2015); whilst they are largely suited for density estimation, they have also started to see some use in spatial statistics (e.g., see Sections 3.1.2 and 4.1.1).

Space folding: occurs when there exists $\mathbf{s}, \mathbf{u} \in G$ such that $\mathbf{f}(\mathbf{s}) = \mathbf{f}(\mathbf{u})$, where $\mathbf{f}(\cdot)$ is the warping function

In Figure 1 we show the utility of such models. The top-left panel depicts an underlying process, which is constructed as a variation of the Rosenbrock function, which we define on $G = [-1, 2] \times [-1, 2]$ as

$$Y(\mathbf{s}) = ((1 - s_1)^2 + 100(s_2 - s_1^2)^2)^{\frac{1}{4}}, \quad \mathbf{s} \in G. \tag{3}$$

The top-right panel depicts observations of this underlying process, which are point-referenced, incomplete, and noisy. The bottom-left panel shows the prediction from a deep compositional spatial model with 18 layers, where fitting was done using maximum likelihood techniques. Each layer in the model corresponds to a radial basis function (Perrin & Monestiez, 1999) that injectively warps (expands or contracts) a

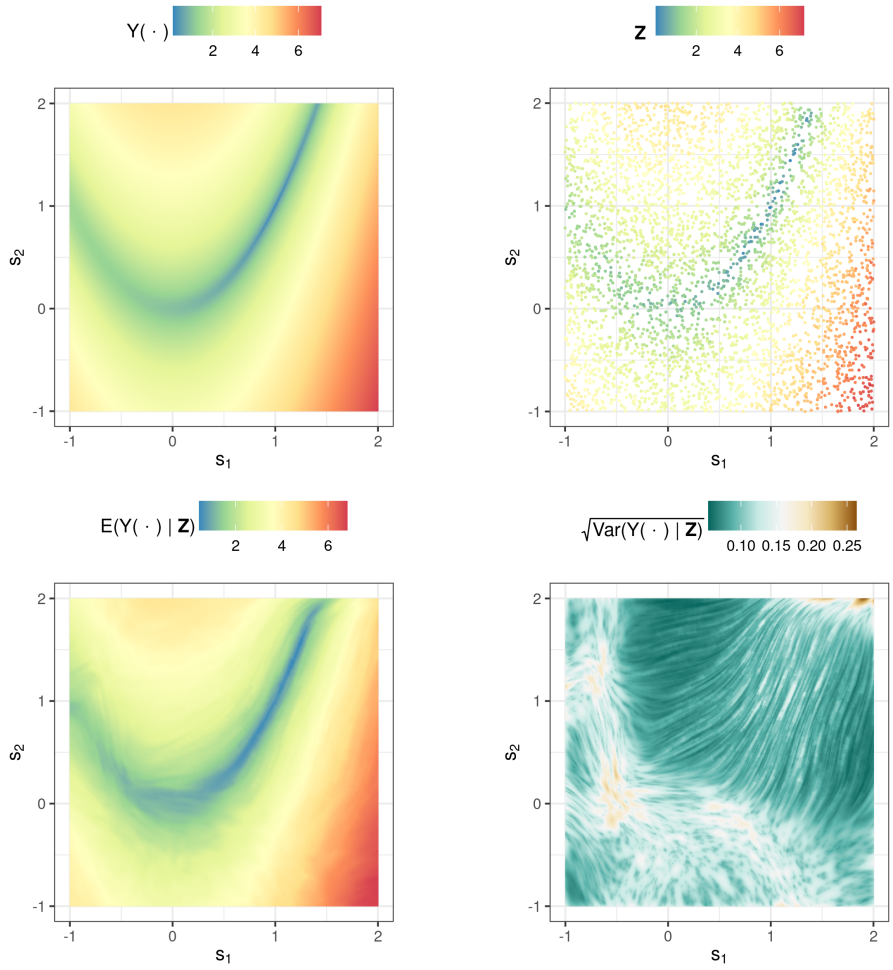

Figure 1: Illustration of spatial prediction using deep compositional spatial models. The top-left panel depicts the true underlying process, generated on $G = [-1, 2] \times [-1, 2]$ via Equation 3. The top-right panel depicts the observations of the true process, that are used to train the deep spatial model. The bottom-left and bottom-right panels depict the prediction (conditional expectation) and standard error (square root of the conditional variance) on G after fitting the deep model using maximum likelihood.

part of the domain; the bottom-right panel shows the associated prediction standard errors. Note how the warping captures the spatially-varying anisotropy and variability of the process, with the valley and plateau areas of the modified Rosenbrock function readily apparent in the standard error surface.

Rosenbrock function: The classic Rosenbrock function is given by $Y(\mathbf{s}) = (c_1 - s_1)^2 + c_2(s_2 - s_1^2)^2$, where $c_1, c_2 \in \mathbb{R}$

Several of the above techniques have been considered in a Bayesian setting. One may simply equip the weights in a deep neural network with prior distributions (e.g., Neal, 1996, Chapter 1), and this was the approach adopted by Zammit-Mangion et al. (2021). Alternatively, one can express the outputs at intermediate layers as Gaussian processes over the outputs of the previous layers. This construction leads to what was coined the *deep Gaussian process* in the machine literature by Damianou & Lawrence (2013), but which made an appearance in the spatial statistics literature a decade earlier (Schmidt & O'Hagan, 2003). The main contribution of Damianou & Lawrence (2013) was the use of sparse Gaussian processes (Quiñonero-Candela & Rasmussen, 2005) and the development of a variational inference scheme for the deep Gaussian

process, that allows it to be fitted with large data sets; their architecture has been used to success in a variety of applications (e.g., Salimbeni & Deisenroth, 2017).

Deep Gaussian processes play a central role in the construction of general deep spatial models, as we discuss next.

Deep Gaussian process: a process constructed as $Y(\cdot) = Y_n(Y_{n-1}(\cdots(Y_1(\cdot))))$, where $Y_1(\cdot),\ldots,Y_n(\cdot)$ are Gaussian processes

3.1.2 Nested spatial processes

Models that successively warp space can be viewed as special cases of "nested spatial processes" where one constructs models by defining a chain of conditional probability models (see Dunlop et al., 2018, for a general framework).

Bolin & Lindgren (2011) consider the case where the probability models are constructed through stochastic partial differential equations via the nesting

$$\begin{aligned}
\mathcal{L}_n Y_n(\cdot) &= Y_{n-1}(\cdot),\\
\mathcal{L}_{n-1} Y_{n-1}(\cdot) &= Y_{n-2}(\cdot),\\
&\vdots \qquad\quad \vdots\\
\mathcal{L}_2 Y_2(\cdot) &= Y_1(\cdot),\\
\mathcal{L}_1 Y_1(\cdot) &= \mathcal{L}_W W(\cdot),
\end{aligned}$$

where $\mathcal{L}_1,\ldots,\mathcal{L}_n$ and $\mathcal{L}_W$ are linear operators, $W(\cdot)$ is spatial white noise, $Y_1(\cdot),\ldots,Y_{n-1}(\cdot)$ are intermediate process layers, and $Y(\cdot) \equiv Y_n(\cdot)$ is the (often latent) process of interest. A straightforward substitution yields the model $\mathcal{L}_1 \cdots \mathcal{L}_n Y_n(\cdot) = \mathcal{L}_W W(\cdot)$ which, under certain choices for the operators, can be easily discretized to obtain (possibly nonstationary) Gaussian MRFs (GMRFs) with highly flexible covariance matrices. The models of Sidén & Lindsten (2020), coined deep GMRFs, are of similar form, but specifically designed to take advantage of CNNs that are readily available in deep learning software libraries. Their approach only considers stationary probability models and yet is seen to outperform many state-of-the-art spatial prediction methods. Both Bolin & Lindgren (2011) and Sidén & Lindsten (2020) focus on models that are nested linearly (nonlinearity is briefly considered by Sidén & Lindsten, 2020), resulting in linear, Gaussian models that can subsequently be fitted using likelihood, variational, or Markov chain Monte Carlo (MCMC) techniques.

Variational inference: an approximate Bayesian inference technique where the objective is to maximize a lower-bound on the marginal likelihood

Maroñas et al. (2021) consider the related model,

$$\begin{aligned}
Y_n(\cdot) &= f_{\boldsymbol{\vartheta}}(Y_1(\cdot)),\\
Y_1(\cdot) &\sim \mathrm{GP}(\mu(\cdot), C(\cdot,\cdot)),
\end{aligned}$$

where $f_{\boldsymbol{\vartheta}}(\cdot)$ is constructed using compositions of Sinh-Arcsinh transforms as in Rios & Tobar (2019) (see also Section 3.2), but where the parameters of the transformations are themselves outputs of neural networks:

$$\begin{aligned}
f_{\boldsymbol{\vartheta}(\cdot)}(Y_1(\cdot)) &= \tilde{f}_{\boldsymbol{\vartheta}_{n-1}(\cdot)} \circ \cdots \circ \tilde{f}_{\boldsymbol{\vartheta}_1(\cdot)}(Y_1(\cdot)),\\
\boldsymbol{\vartheta}_l(\cdot) &= \mathrm{NN}(\cdot\,, \mathbf{W}_l), \quad l = 1,\ldots,n-1,
\end{aligned}$$

where $\mathrm{NN}(\cdot\,,\mathbf{W})$ denotes an arbitrary neural network with weights $\mathbf{W}$ that takes as inputs the spatial coordinates as well as possibly other covariates of interest at the corresponding locations; $\tilde{f}_{\boldsymbol{\vartheta}_l(\cdot)}(\cdot)$ is the Sinh-Arcsinh transform with parameters $\boldsymbol{\vartheta}_l(\cdot)$; $\mathbf{W}_l, l = 1,\ldots,n-1$, denote neural network weights; and where $\boldsymbol{\vartheta}(\cdot) \equiv (\boldsymbol{\vartheta}_1(\cdot)',\ldots,\boldsymbol{\vartheta}_{n-1}(\cdot)')'$ are now input (i.e., spatially) dependent transformation parameters. Maroñas et al. (2021) develop computationally efficient techniques for estimation and prediction with this model, and illustrate their approach on air quality (temporal) and precipitation (spatial) data.

In another class of nested processes, conditional dependence is modeled via the covariance function. Consider, for example, the following nesting,

$$\begin{aligned} Y_n(\cdot) &= \text{GP}(\mathbf{x}(\cdot)'\boldsymbol{\beta}, C_n(\cdot,\cdot\,; \mathbf{Y}_{n-1}(\cdot))), \\ \mathbf{Y}_{n-1}(\cdot) &= \text{GP}(\mathbf{0}, \mathbf{C}_{n-1}(\cdot,\cdot\,; \mathbf{Y}_{n-2}(\cdot))), \\ \vdots \quad &= \quad \vdots \\ \mathbf{Y}_1(\cdot) &= \text{GP}(\mathbf{0}, \mathbf{C}_1(\cdot,\cdot)), \end{aligned}$$

where $Y(\cdot) \equiv Y_n(\cdot)$ is the underlying (often latent) process of interest and $\mathbf{Y}_1(\cdot), \ldots, \mathbf{Y}_{n-1}(\cdot)$, are nested processes parameterizing the covariance functions of subsequent layers. Monterrubio-Gómez et al. (2020) consider the $n = 2$ case, where $Y_1(\cdot)$ is the (univariate) process describing the log of the length scale for the covariance function of $Y_2(\cdot)$, that is, $C_2(\cdot,\cdot\,; Y_1(\cdot))$, which is given by the non-stationary Matérn representation of Paciorek & Schervish (2006). Zhao et al. (2021) call this model the "batch deep Gaussian process regression model". They also employ the Paciorek & Schervish (2006) representation, but let $\mathbf{Y}_1(\cdot)$ be a bivariate Gaussian process, with the first variate the square root of the length scale, and the second variate the square root of the variance parameter. Zhao et al. (2021) also present a dynamic state-space representation for the deep Matérn regression model for the *temporal* case, based on known equivalencies between the two representations (e.g., Särkkä & Solin, 2019, Chapter 12). This representation is attractive as it allows sequential estimation methods (e.g., Kalman filtering based methods) to be used with deep hierarchies. This approach has yet to be applied in a spatio-temporal setting.

State-space model: A two-layer temporally-indexed model, where the first layer models the evolution of a latent state in time, and the second layer models the observations of the latent states

A related nested spatial process model is given by Chen et al. (2021) in what they call "DeepKriging". Here, the bottom layer (referred to as an "embedding layer") of the model is given by the conventional multivariate spatial random effects model (e.g., Nguyen et al., 2017),

$$\mathbf{Y}_1(\cdot) = \mathbf{W}_{1,x}\mathbf{x}(\cdot) + \mathbf{W}_{1,\phi}\boldsymbol{\phi}(\cdot) + \mathbf{b}_1, \tag{4}$$

where $\mathbf{W}_{1,x}$ and $\mathbf{W}_{1,\phi}$ are weight matrices that need to be estimated, $\mathbf{x}(\cdot)$ are covariates, $\boldsymbol{\phi}(\cdot)$ are spatial basis functions, and $\mathbf{b}_1$ are bias parameters. The multivariate spatial process is then treated, up to a monotonic nonlinear transformation, as a set of basis functions for the subsequent layer. This approach yields the following nesting for $l = 2, \ldots, n-1$,

$$\mathbf{Y}_l(\cdot) = \mathbf{W}_l\psi_{l-1}(\mathbf{Y}_{l-1}(\cdot)) + \mathbf{b}_l,$$

where $\psi_l(\cdot)$ is the monotonic nonlinear transformation for the lth layer (applied element-wise) and where the other quantities are defined similarly as in Equation 4. The final layer (i.e., the process of interest) is then modeled as $Y_n(\cdot) = \psi_n(\mathbf{W}_n\psi_{n-1}(\mathbf{Y}_{n-1}(\cdot)) + b_n)$. Chen et al. (2021) show that the DeepKriging predictor is highly adaptable to non-stationary data, and that it can be quick to implement using GPU acceleration (see Section 5). The DeepKriging architecture can be viewed as a special case of the deep generalized linear (mixed) model proposed by Tran et al. (2020).

Kalman filter: a sequential estimation method for linear, Gaussian, state-space models

Embedding layer: a layer in a neural network which projects the input into a lower-dimensional space

3.1.3 Deep hybrid spatio-temporal process models

In this section we describe a few hybrid approaches for modeling the process component of dynamic spatio-temporal statistical models. First, as noted in Section 2.3, RNNs have provided an effective way to model complex temporal dependency. They have been used in the statistics context for time series applications (e.g., see Nguyen et al., 2019, for a hybrid RNN/stochastic volatility model). RNNs have also been combined in various ways with other neural architectures to accommodate spatial input (e.g., Dixon et al., 2019). As with multi-level BHM implementations of spatio-temporal

processes, these implementations have a very large number of parameters and thus require a high volume of data and computational overhead to implement. An alternative implementation of an RNN with a significantly more parsimonious representation is the echo state network (ESN, Jaeger, 2001, 2007b). The ESN is a type of "reservoir computing" in which the hidden states and inputs evolve in a dynamical reservoir where the parameters (weights) that describe their evolution are drawn at random, with most assumed to be zero. Only parameters (weights) that are estimated at the output stage, that is, those that connect the hidden states to the output response, are estimated.

McDermott & Wikle (2017) use this idea in a hybrid statistical/ESN model for spatio-temporal prediction with an additional quadratic output state. Their model is as follows: For time $t = 1, \ldots, T$,

$$\text{Response:} \quad \mathbf{Z}_t = \mathbf{V}_1\mathbf{h}_t + \mathbf{V}_2\mathbf{h}_t^2 + \boldsymbol{\epsilon}_t, \quad \text{for} \quad \boldsymbol{\epsilon}_t \sim \text{Gau}(\mathbf{0}, \sigma_\epsilon^2\mathbf{I}), \tag{5}$$

$$\text{Hidden states:} \quad \mathbf{h}_t = g_h\left(\frac{\nu}{|\lambda_w|}\mathbf{W}\mathbf{h}_{t-1} + \mathbf{U}\mathbf{x}_t\right), \tag{6}$$

$$\begin{aligned}\text{Parameters:} \quad & \mathbf{W} = [w_{i,\ell}]_{i,\ell} : w_{i,\ell} = \gamma^w_{i,\ell} \cdot \text{Unif}(-a_w, a_w) + (1 - \gamma^w_{i,\ell})\ \delta_0, \\ & \mathbf{U} = [u_{i,j}]_{i,j} : u_{i,j} = \gamma^u_{i,j} \cdot \text{Unif}(-a_u, a_u) + (1 - \gamma^u_{i,j})\delta_0, \\ & \gamma^w_{i,\ell} \sim \text{Bern}(\pi_w), \ \ \gamma^u_{i,j} \sim \text{Bern}(\pi_u),\end{aligned}$$

where $\mathbf{Z}_t$ is the response vector at time t; $\mathbf{h}_t$ is the hidden state vector; $\mathbf{x}_t$ is a vector of input covariates; $\mathbf{W}$ is the hidden-process-evolution weight matrix; $g_h(\cdot)$ is an activation function; $\mathbf{U}$ is the input weight matrix; and $\mathbf{V}_1$, $\mathbf{V}_2$ are weight matrices associated with the linear and quadratic output, respectively. In addition, δ_0 is the Kronecker delta function at zero, λ_w corresponds to the largest eigenvalue of $\mathbf{W}$, and ν is an ESN control parameter. The only parameters estimated in this formulation are $\mathbf{V}_1$, $\mathbf{V}_2$, and σ_ϵ^2 from Equation 5, obtained using a regularized regression (e.g., ridge or lasso). Importantly, the elements of the matrices $\mathbf{W}$ and $\mathbf{U}$ are drawn randomly as either a zero or uniformly in the range $(-a_w, a_w)$ and $(-a_u, a_u)$, respectively.

Reservoir Computing: a type of machine learning where all hidden layer weights are chosen randomly and only the weights (parameters) associated with the output layer are learned

McDermott & Wikle (2017) used an ensemble approach to do uncertainty quantification with this model, by generating many copies of the hidden units $\mathbf{h}_t$ through different simulated weight matrices, and using the ensemble output to generate a predictive distribution. Bonas & Castruccio (2021) showed that one can ensure proper coverage in the ensemble approach through calibration. Although McDermott & Wikle (2019a) demonstrated that one could use Bayesian inference to estimate the reservoir weight matrices in Equation 6 with some computational effort, it only performed slightly better than the ensemble approach, and is therefore not worth the extra computational effort.

It is straightforward to implement a spatio-temporal ESN in a deep context – one simply allows the hidden states at one level to be the inputs to the next level. Indeed, it has been demonstrated that such models are beneficial in that they can more readily utilize multi-scale temporal and spatial dependence in their predictions (Jaeger, 2007a, McDermott & Wikle, 2019b). Typically, as with CNNs, one performs some type of dimension reduction between layers to reduce the dimensionality of the hidden states and, ultimately, the number of variables used for prediction in the final layer. For example, McDermott & Wikle (2019b) give the following approach for time t: starting with the nth hidden layer that takes $\mathbf{x}_t$ as input, the model iterates from $l = n-1, \ldots, 1$ (note, the label ordering is opposite to that presented in Section 3.1.2,

where here n corresponds to the input layer):

$$\text{Input Stage:} \qquad \mathbf{h}_{t,n} = g_h\left(\frac{\nu_n}{|\lambda_{W_n}|}\mathbf{W}_n\mathbf{h}_{t-1,n} + \mathbf{U}_n\mathbf{x}_t\right), \tag{7}$$

$$\text{Reduction Stage } l+1: \qquad \widetilde{\mathbf{h}}_{t,l+1} \equiv \mathcal{Q}(\mathbf{h}_{t,l+1}), \;\; l = n-1,\ldots,1, \tag{8}$$

$$\text{Hidden Stage } l: \qquad \mathbf{h}_{t,l} = g_h\left(\frac{\nu_l}{|\lambda_{W_l}|}\mathbf{W}_l\mathbf{h}_{t-1,l} + \mathbf{U}_l\widetilde{\mathbf{h}}_{t,l+1}\right), \;\; l = n-1,\ldots,1 \tag{9}$$

where the weight matrices are generated randomly as above, and $\lambda_{W_l}, l = n,\ldots,1$ are the largest eigenvalues of their respective weight matrices, ν_l are ESN control parameters (that are pre-specified), and $\mathcal{Q}(\cdot)$ is a dimension reduction function that reduces the dimension of $\mathbf{h}_{t,l}$ to $\widetilde{\mathbf{h}}_{t,l}$ for each level $l = n,\ldots,2$ (level 1 is typically not reduced, but could be). McDermott and Wikle (2019) use each of the dimension-reduced hidden states and the level 1 hidden states as possible predictors in the response level of the model (analogous to Equation 5). Note that the dimension reduction in Equation 8 can be unsupervised (e.g., principal component reduction or random projection) or supervised (e.g., with the use of autoencoders). In the case of unsupervised dimension reduction, the hidden-state construction represented in Equations 8 and 9 is simply a multi-resolution stochastic transformation of the input.

As with the shallow spatio-temporal ESN presented above, uncertainty quantification can be accounted for by bootstrap ensemble approaches or through Bayesian inference. For example, the Bayesian approach given in McDermott & Wikle (2019b) extends the basic output function and data stage of the ensemble deep spatio-temporal ESN. Specifically, the ensemble member hidden states sampled from different random reservoirs are used in a regularized linear regression to model a transformation of the mean response from the data stage, similar to generalized additive models. Specifically:

$$\text{Data Stage:} \qquad \mathbf{Z}_t|\boldsymbol{\alpha}_t \;\sim\; \text{Dist}(\tilde{g}(\boldsymbol{\alpha}_t), \boldsymbol{\Theta}), \tag{10}$$

$$\text{Output Stage:} \qquad \boldsymbol{\alpha}_t = \frac{1}{n_{\text{res}}}\sum_{j=1}^{n_{\text{res}}}\left[\boldsymbol{\beta}_1^{(j)}\mathbf{h}_{t,1}^{(j)} + \sum_{l=2}^{n}\boldsymbol{\beta}_l^{(j)}\widetilde{\mathbf{h}}_{t,l}^{(j)}\right] + \boldsymbol{\eta}_t, \tag{11}$$

where $\boldsymbol{\eta}_t \sim \text{Gau}(\mathbf{0}, \sigma_\eta^2\mathbf{I})$, "Dist" denotes an unspecified distribution (e.g., exponential family), $\tilde{g}(\cdot)$ is some specified transformation (e.g., inverse link function), $\boldsymbol{\beta}_l^{(j)}$ are regression matrices for $l = 1,\ldots,n$ and the jth reservoir replicate, where $j = 1,\ldots,n_{\text{res}}$. In the Bayesian implementation, stochastic search variable selection (SSVS) or other Bayesian variable selection methods can be used to regularize the regression matrices.

The hybrid ESN approach, and variations thereof, have been used to successfully predict sea surface temperatures (McDermott & Wikle, 2017), soil moisture (McDermott & Wikle, 2019b), wind power (Huang et al., 2021b), industrial processes (Dixon, 2021), electricity prices (Klein et al., 2020), asset volatility (Parker et al., 2021), and air pollution (Bonas & Castruccio, 2021).

3.2 Deep learning for characterizing complex data models

It is common in various branches of statistics to develop probability models for a transformation of the data, rather than for the data themselves. Such transformations include the log transform, the exponential and power transforms (e.g., Cressie, 1978), the Box–Cox transform (Box & Cox, 1964), and the Tukey g-and-h transform (Tukey, 1977). Many of these have been used in spatial statistics (e.g., De Oliveira et al., 1997, Xu & Genton, 2017). These transformations generally have a parsimonious parameterization, and are relatively simple in form. More flexibility can be achieved by expressing the transformation using a (suitably structured) deep learning architecture.

Exponential transform: the one-parameter transform given by $g(Y(\cdot)) = (\exp(\lambda Y(\cdot)) - 1)/\lambda$ for $\lambda \neq 0$ and $g(Y(\cdot)) = Y(\cdot)$ for $\lambda = 0$

The "warped GP" of Snelson et al. (2004) is given by

$$g_{\boldsymbol{\vartheta}}(Z_i) = Y(\mathbf{s}_i) + \epsilon_i, \quad i = 1, \ldots, m, \tag{12}$$

$$Y(\cdot) \sim \mathrm{GP}(\mu(\cdot), C(\cdot, \cdot)), \tag{13}$$

Box–Cox transform: the one-parameter transform given by $g(Y(\cdot)) = (Y(\cdot)^\lambda - 1)/\lambda$ for $\lambda \neq 0$ and $g(Y(\cdot)) = \log Y(\cdot)$ for $\lambda = 0$

where $\{Z_i\}$ are the observations, $\{\epsilon_i\}$ are the measurement errors, $g_{\boldsymbol{\vartheta}}(\cdot)$ is a monotonic function parameterized via $\boldsymbol{\vartheta}$ and, in the context of spatial statistics, $Y(\cdot)$ is the spatial process of interest with mean function $\mu(\cdot)$ and covariance function $C(\cdot, \cdot)$. Snelson et al. (2004) propose using a one-layer net of $\tanh(\cdot)$ functions to model $g_{\boldsymbol{\theta}}(\cdot)$. Rios & Tobar (2019) extend the warped GP to the "compositionally warped GP" by expressing $g_{\boldsymbol{\vartheta}}(\cdot)$ as a (deep) composition of elementary functions which have explicit derivatives and inverses; these include the Box–Cox transform and the Sinh-Arcsinh transform. Murakami et al. (2021) use the compositionally warped GP in a spatial mixed-model setting, while Maroñas et al. (2021) propose a computationally-efficient variational algorithm for fitting the model. Note that, unlike in conventional generalized linear models (GLMs), no distribution is pre-specified for the $\{Z_i\}$, which depends on $\boldsymbol{\vartheta}$ that needs to be estimated. This approach is thus distinct from the deep GLMM setting of Tran et al. (2020) where the $\{Z_i\}$ are assumed to come from a known exponential family and the link function is fixed, and where a deep net is used to model what is conventionally the linear component. The latter, however, bears connections to the spatial GLMM of Diggle et al. (1998) and thus is a strong candidate for use in spatial applications. Recently, Bradley (2022) considered a multi-level BHM for unknown transformations of multiple response-type data. Their approach takes into account the uncertainty associated with the unknown transformation and has been applied to both spatial and spatio-temporal data.

Sinh-Arcsinh transform: the two-parameter transform given by $g(Y(\cdot)) = \sinh(\lambda \sinh^{-1} Y(\cdot) - \gamma)$

3.3 Deep learning for parameter estimation in spatial and spatio-temporal empirical hierarchical models

As discussed in Sections 2.1 and 2.2, classical hierarchical spatial and spatio-temporal statistical models generally involve parameters that characterize the first-order, second-order, and sometimes third and higher order, properties of the process. Estimating these parameters is often a computational bottleneck for a variety of reasons. Sometimes the likelihood is difficult to evaluate, especially as the size of the data set increases. In other cases the likelihood may be computationally tractable, but difficult to explore using conventional optimization techniques. This problem has led to a recent drive in the spatial statistics literature to employ neural networks to construct mappings between the observation space and the parameter space. Once trained, such a network is, in principle, able to provide practically-usable parameter estimates from any observed data set in a fraction of the time needed by conventional (e.g., likelihood) techniques, irrespective of the model complexity.

Recall the IDE dynamic spatio-temporal model discussed in Section 3.1.3. One of the greatest challenges with IDE models is estimating the parameters associated with the mixing kernel, especially when they are parameterized in a highly flexible manner (e.g., to allow for spatio-temporally-varying mixing). In particular, joint inference over the latent spatio-temporal process and a spatio-temporally-varying mixing kernel is notoriously difficult and time consuming. Zammit-Mangion & Wikle (2020) proposed to alleviate the computational burden by finding a (highly complex) time-invariant map between the mixing kernel parameters and lagged values of the process. This complex map was described using a CNN as in De Bézenac et al. (2019), and was fitted offline using a copious amount of reanalysis geophysical data. Zammit-Mangion & Wikle (2020) then converted Equation 2 into a state-dependent (and hence nonlinear) IDE, for which standard Kalman-based filtering techniques could be used. Their results showed a 100-fold decrease in the time required to make inference on the process

and the parameters over standard moving-window-based maximum-likelihood methods. They also demonstrated that their approach has robust transfer learning potential by generating successful forecasts of a process (precipitation) that is considerably different from that on which the CNN was trained (sea surface temperature).

Gerber & Nychka (2021) used a similar CNN architecture to that of Zammit-Mangion & Wikle (2020) to estimate the length scale and the effective degrees of freedom (in their case a variance parameter) of a Gaussian process with a Matérn covariance function observed in noise. The CNN was trained using thousands of simulated fields, corresponding to different parameters, as input data. The parameters the fields were simulated at were then used as output data. Gerber & Nychka (2021) showed that their CNN estimator was comparable to the maximum likelihood estimator in terms of bias and variance and, like Zammit-Mangion & Wikle (2020), reported a hundred-fold speed up in estimation. Lenzi et al. (2021) considered a similar CNN framework for estimating parameters in models of spatial extremes. This approach to parameter estimation is still in its infancy in spatial statistics, but has seen wide use in a variety of related areas that require parameter estimation. For example, Rudi et al. (2021) use a similar approach for estimating the parameters of a system of ordinary differential equations.

Deep networks have also been used to facilitate inference of spatial models with intractable likelihoods. Vu et al. (2022a), for example, used deep compositional spatial models to emulate the sufficient statistics required to construct synthetic likelihood functions. These synthetic likelihood functions were then used to speed up parameter inference with the spatial Potts model and the spatial autologistic model.

Potts and autologistic models: Spatial models on a lattice where each vertex may be in one of several (finite) states

4 Other uses of deep learning in spatial and spatio-temporal statistics

Section 3 showcased instances of classical spatial and spatio-temporal models that incorporate ideas or model formalisms that are often seen in the deep learning literature. In this section, we focus on specific types or applications of spatial and spatio-temporal models (specifically, point-process models, emulation, and reinforcement learning) that at their core incorporate deep learning architectures.

4.1 Deep Poisson point process models

4.1.1 Measure transport for modeling non-homogeneous Poisson point processes

Tabak & Vanden Eijnden (2010) introduced the concept of a *flow* for constructing complicated probability density functions, which has seen substantial use and development in the machine learning literature (e.g., Rezende & Mohamed, 2015). Consider some continuous and differentiable density function $f_0(\mathbf{x}), \mathbf{x} \in \mathcal{X}$, which one wishes to model, and let $T_{\boldsymbol{\theta}}(\cdot)$ be some bijective and differentiable map parameterized through the parameter vector $\boldsymbol{\theta}$. Let $f_1(\cdot)$ be a *reference* density that is easy to evaluate. In this setting, one expresses $f_0(\cdot)$ via the popular change of variables formula

$$f_0(\mathbf{x}) = f_1(T_{\boldsymbol{\theta}}(\mathbf{x}))|\det(\nabla(T_{\boldsymbol{\theta}}(\mathbf{x})))|, \quad \mathbf{x} \in \mathcal{X}. \tag{14}$$

The attraction of this construction is that one need only estimate $\boldsymbol{\theta}$ in order to construct $f_0(\cdot)$, which can be done relatively efficiently through likelihood methods. Specifically, assume that one has a sample $\mathbf{x}_i, i = 1, \ldots, N$; then one could obtain an estimate $\hat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}$ via the operation,

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left\{ \sum_{i=1}^{N} \log f_1(T_{\boldsymbol{\theta}}(\mathbf{x}_i)) + \log|\det(\nabla(T_{\boldsymbol{\theta}}(\mathbf{x}_i)))| \right\}.$$

Since bijection and differentiability are preserved through composition, it is common to string together several transformations through composition, K say, to obtain a more complex mapping (this stringing together is what gives rise to the term "flow"), so that $T_{\boldsymbol{\theta}}(\cdot) \equiv T_{K,\boldsymbol{\theta}} \circ \cdots \circ T_{1,\boldsymbol{\theta}}(\cdot)$, where each of $T_{1,\boldsymbol{\theta}}(\cdot), \ldots T_{K,\boldsymbol{\theta}}(\cdot)$ is bijective and differentiable. Various models for $T_{k,\boldsymbol{\theta}}(\cdot), k = 1, \ldots, K$, have been proposed that, while being bijective and differentiable, also lead to a Jacobian determinant in Equation 14 that is computationally tractable. Pertinent to this review are those constructed via triangular maps, where the ith output of $T_{k,\boldsymbol{\theta}}(\mathbf{x})$, that is, $T^{(i)}_{k,\boldsymbol{\theta}}(\mathbf{x})$, is a monotonic, nonlinear, function of the ith dimension of $\mathbf{x}$ (i.e., $\mathbf{x}^{(i)}$), with parameters that often depend in a highly nonlinear manner on $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(i-1)}$ via a deep learning framework (e.g., Kingma et al., 2016, Papamakarios et al., 2017, Huang et al., 2018).

Normalizing flows, as they are often called since $f_1(\cdot)$ is typically the normal probability density function, are of interest in spatio-temporal statistics for various reasons. First, they could be used to improve variational inference when dealing with large spatial data (e.g., Hensman et al., 2013, Rezende & Mohamed, 2015). More directly though, one can use normalizing flows to model the intensity function of a non-homogeneous (temporal, spatial, or spatio-temporal) Poisson point process. The connection arises from the fact that if $\lambda(\cdot)$ is the intensity function of a Poisson point process on $\mathcal{X}$, and if $\mu_\lambda(\mathcal{X}) \equiv \int_{\mathcal{X}} \lambda(\mathbf{x})\mathrm{d}\mathbf{x}$ is the integrated intensity, then $\lambda(\cdot)/\mu_\lambda(\mathcal{X})$ is a density, referred to as the process density by Taddy & Kottas (2010). Ng & Zammit-Mangion (2022a) propose using the neural autoregressive flow of Huang et al. (2018) for modeling the process density, and estimate the integrated intensity as the number of observed points, N. They show that, under mild regularity assumptions on the intensity function, the neural autoregressive flow is a universal approximator (i.e., that it can model arbitrarily complex intensity functions). Ng & Zammit-Mangion (2022b) apply a similar approach to point processes on the sphere using exponential map radial flows (Sei, 2013, Rezende et al., 2020).

Non-homogeneous Poisson point process with intensity function $\lambda(\cdot)$: a random finite subset of a domain G, where the number of points in any $A \subset G$ is Poisson distributed with parameter $\int_A \lambda(\mathbf{s})\mathrm{d}\mathbf{s}$, and where the number of points in disjoint regions are independent

4.1.2 Conditional intensity function modeling

A considerable literature has recently emerged on the modeling of conditional intensity functions using ordinary differential equations that are built using neural networks. These intensity functions, which are conditional on the event history, are often more applicable to real world processes, such as financial or crime data. Jia & Benson (2019) apply the neural ordinary differential equations of Chen et al. (2018) to model conditional intensity functions, while Chen et al. (2020) extend the concept to the spatio-temporal case.

Zhu et al. (2020) take a different approach and directly model the influence of a past point on the intensity function using a Gaussian mixture, where the parameters in each mixture component are the outputs of a simple, one-layer, neural net that take the spatial coordinates as input. The model yields a flexible conditional-intensity function that is highly spatially heterogeneous and applicable in various contexts such as the analysis of earthquake data and crime data.

4.2 Deep emulation

An emulator is a model that acts as a surrogate for a more complex, generally numerical and physics-based, model. Emulators are most often forward maps between the (input) parameter space and the (output) response space of the numerical model, although emulators for inverse maps have also been developed (see below). They tend to be of greatest use when the numerical model is computationally intensive to run, and when the output varies reasonably smoothly with small changes in the input. Emulators find most use in prediction (i.e., for predicting numerical model output for as yet unprobed parameters), experimental design, and calibration (i.e., for tuning

numerical models to observed data).

The most common emulator is the Gaussian process emulator (Kennedy & O'Hagan, 2001), where the map between the input parameters and the numerical model outputs is modeled via a Gaussian process. Several developments on the vanilla Gaussian process have been proposed for the purpose of emulation, such as the treed Gaussian process (Gramacy & Lee, 2008) and the deep Gaussian process of Damianou & Lawrence (2013) or variants thereof (Monterrubio-Gómez et al., 2020, Ming et al., 2021, Marmin & Filippone, 2022, Sauer et al., 2022).

Treed Gaussian process: a nonstationary Gaussian process model constructed by modeling several stationary Gaussian process models on a random, marginalized, partitioning of the spatial domain

The output of a numerical, physics-based, model is often temporal, spatial, or spatio-temporal, and several approaches have been developed to deal with this extra level of complexity. Leeds et al. (2013) used random forests to model the three-dimensional output of a biogeochemical model, while Leeds et al. (2014) considered a deep quadratic nonlinear model to emulate a multivariate spatio-temporal process. Zhang et al. (2015) used a nonseparable Gaussian process on the parameter-spatial-temporal input space to model the output of complex computational fluid dynamic models; see also Castruccio et al. (2014) and Chang et al. (2016) for related examples.

Random forest: A collection of decision or regression trees, trained on random subsets of the data and input variables

Research into the use of deep learning models for emulating the spatial or spatio-temporal output of numerical models is in its infancy, but has a lot of potential benefit that will likely make it an active area of research in the coming years. This benefit is seen in the work of Bhatnagar et al. (2022) who use deep learning models known as long-short term memory (LSTM) models to find a complex map between the numerical model output and the input parameters used to generate the output (i.e., an inverse map); this model can then be directly used for calibration. Another example is the work of Cartwright et al. (2022), who emulate the spatial output of a Lagrangian particle dispersion model (LPDM, which simulates particle trajectories in the atmosphere from a source location) using a deep learning model known as a convolutional variational autoencoder (CVAE). The authors show that the CVAE can be used to effectively predict the output of the LPDM over a wide spatial domain given only a few simulations, and that the CVAE considerably outperforms the conventional emulator based on singular vectors (e.g., Hooten et al., 2011). Recently, Gopalan & Wikle (2022) extended the singular vector approach to higher-order tensor decompositions to emulate complex multi-dimensional spatio-temporal data, including the movement trajectories of agents in an agent-based model. Their approach is flexible in that different machine learning methods (e.g., random forests and neural networks) or GP regression models can be used in the various tensor dimensions.

Convolutional variational autoencoder: An autoencoder where the encodings are defined as random variables, where training is done using variational Bayes, and where the encoder and decoder layers are convolutional layers

4.3 Reinforcement learning

Reinforcement learning (RL) is goal-oriented learning from the interaction between agents and their environment, where the agents learn to take actions that maximize a specified reward function. The RL framework expands the characterization of agents from traditional agent-based models to include notions of perception and memory, where perception is related to the agent's state, and memory is incorporated by allowing control parameters to be learned based on the agents' experience with their environment (see Sutton & Barto, 1998, for a classic overview). RL has seen a resurgence due to the success of embedding deep models in various components of the learning process (see the overview in Henderson et al., 2018). Given that many agent-based systems, such as autonomous vehicle control systems and collective animal movement, are formulated in space and time, it is natural to consider RL for such problems (e.g., Ma et al., 2021, Tampuu et al., 2017). However, for many such systems it is challenging to define the local costs or rewards that control agent behavior *a priori*, which has led to interest in inverse reinforcement learning (IRL), whereby one uses observed system behavior to learn the underlying costs or rewards (e.g., Ng & Russell, 2000). In the spatio-temporal statistical context, Schafer et al. (2020) used Bayesian IRL to

recover the costs that guppies in a tank consider with respect to the tradeoffs that exist between collective movement and movement to a safe zone.

Agent-Based Modeling: a bottom-up modeling approach that specifies simple rules for how agents interact with each other and their environment; these rules lead to complex behavior

5 Technologies used for deep learning in spatial statistics

Most of the methods and techniques discussed in Sections 2–4 require the estimation of parameters appearing in a deep hierarchy, generally via the optimization of a (regularized) likelihood function. The past decade has seen a dramatic increase in the availability of tools and hardware specifically designed to solve this problem. The two most important ones, which are indispensable for anyone implementing deep learning models in spatial statistics, are deep learning software libraries, and graphics processing units (GPUs).

At the time of writing, the two most popular deep learning libraries are `PyTorch` and `TensorFlow/Keras`. Both are open source Python libraries that greatly facilitate model construction and fitting through various features. First, they allow one to construct large deep learning models with ease; for example, creating a convolutional layer in a CNN would only require calling a single function. Second, they both offer functionality for automatic differentiation (e.g., Paszke et al., 2017), so that derivatives during optimization can be obtained quickly and effortlessly. Third, they offer a wide-range of (stochastic) gradient-descent strategies (also referred to as (stochastic) optimizers) that have been proven useful for these models, such as Adagrad (Duchi et al., 2011) and Adam (Kingma & Ba, 2014). Finally, they both offer seamless GPU integration, which are a practical requirement when training large, deep learning, models. Both `PyTorch` and `TensorFlow/Keras` functionality have been made available to `R` users (R Core Team, 2022); see, for example, `https://tensorflow.rstudio.com/`.

Autodifferentiation: a tool to automatically compute the derivatives of a computation by application of the chain rule on the elementary operations from which it is constructed

Each likelihood evaluation when training a model incorporating a deep hierarchy generally requires a substantial amount of high-dimensional, but relatively simple, matrix calculations that are naïvely parallelizable (such as addition and multiplication). GPUs contain an exorbitant number of processing cores: In the year 2022 a high-end GPU contained several thousands of cores while a typical high-end central processing unit (CPU) contained a few dozen compute cores. GPUs are thus poised to take advantage of the parallelizable matrix operations, and offer a drastic computational benefit over conventional CPUs. They are also typically available with a very large memory bandwidth (on the order of terabytes per second, as opposed to gigabytes per second) so that the large portions of data in memory that are required for computation can be quickly accessed. GPUs also offer considerable improvement in compute speed, even when shallow models are used; see `https://hpc.niasra.uow.edu.au/azm/Spatial_GPUs_TFv2.html` for a spatial modeling experiment where a GPU is used to fit a (shallow) spatial model nearly 50 times faster than a CPU, although both are using the same optimizer and carrying out the same calculations.

Section 3 reviewed several statistical deep learning/hierarchical models that have been adopted for analyzing spatial data. Software for implementing these methods is still in its infancy, and largely in the form of 'reproducible software' accompanying journal articles. Yet, such software is also an excellent starting point for exploring the features and implementation of these deep learning models, and are a valuable resource. At the time of writing, software for DeepKriging was available at `https://github.com/aleksada/DeepKriging`, software for fitting deep GMRFs was available at `https://bitbucket.org/psiden/deepgmrf/src/master/`, and software for fitting deep compositional spatial models was available at `https://github.com/andrewzm/deepspat`.

6 Conclusion

In this overview, we presented a statistician's perspective and contemporary snapshot of deep learning for spatial and spatio-temporal data. We gave a brief overview of traditional statistical and deep learning models for such data, noting that "deep" models have been integral to modeling spatial and spatio-temporal data in statistics since the 1990s, when computational approaches for fitting multi-level (deep) Bayesian hierarchical models became available. Our focus in this review was on hybrid machine-learning/statistical models that utilize deep learning and that can still accommodate uncertainty quantification and some measure of explainability or interpretability. In the context of latent processes, we discussed hybrid methods such as deep Gaussian processes and deep echo state networks. We also discussed how deep models can be used to characterize complex data models in more traditional multi-level statistical models. We then presented some recent work that has used deep learning to estimate the parameters of various statistical models, from those appearing in covariance functions of spatial models, to those characterizing the transition operator in a spatio-temporal dynamical model. We proceeded to present some other examples where deep models were used in the context of spatial and spatio-temporal data, namely in point process modeling, computer model emulation, and reinforcement learning. Finally, we concluded with a brief overview of technologies that have enabled deep learning at large, and that will be indispensable to the practitioner aspiring to do deep learning with spatial and spatio-temporal data.

Although the fusion of deep machine learning and statistical approaches for spatial and spatio-temporal data is in its infancy, there is substantial research interest in these methods. In addition to extensions and implementations of the methods described here, there are several areas that will likely see greater exploration in the near future. These include the increased use of novel stochastic optimization algorithms, the development of methods for covariance free spatial and spatio-temporal prediction, the development of new explainability and interpretability methods, the incorporation of multi-type, multi-support data, and the development of efficient ways to specify optimal deep architectures. Deep learning itself is a growing area in machine learning/computer science, and as new methods are developed, they will almost certainly be used as inspiration for enhancing traditional statistical methods for analyzing spatial and spatio-temporal data.

Acknowledgements

Christopher K. Wikle's research was supported by the U.S. National Science Foundation (NSF) grant SES-1853096. Andrew Zammit-Mangion's research was supported by an ARC Discovery Early Career Research Award, DE180100203. The authors would like to thank Yi Cao, Wanfang Chen, and Per Sidén, for help with implementing and running software for DeepKriging and deep GMRFs.

References

Abdar M, Pourpanah F, Hussain S, Rezazadegan D, Liu L, et al. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion* 76:243–297

Amato F, Guignard F, Robert S, Kanevski M. 2020. A novel framework for spatio-temporal prediction of environmental data using deep learning. *Scientific Reports* 10(1):1–11

Bai L, Yao L, Kanhere S, Wang X, Sheng Q, et al. 2019. Stg2seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting. *arXiv preprint arXiv:1905.10069*

Banerjee S, Carlin BP, Gelfand AE. 2014. Hierarchical modeling and analysis for spatial data. Boca Raton, FL: CRC Press

Bhatnagar S, Chang W, Kim S, Wang J. 2022. Computer model calibration with time series data using deep learning and quantile regression. *SIAM/ASA Journal on Uncertainty Quantification* 10(1):1–26

Bolin D, Lindgren F. 2011. Spatial models generated by nested stochastic partial differential equations, with an application to global ozone mapping. *The Annals of Applied Statistics* 5:523–550

Bonas M, Castruccio S. 2021. Calibration of spatial forecasts from citizen science urban air pollution data with sparse recurrent neural networks. *arXiv preprint arXiv:2105.02971*

Box GE, Cox DR. 1964. An analysis of transformations. *Journal of the Royal Statistical Society: Series B* 26(2):211–243

Bradley JR. 2022. Joint Bayesian analysis of multiple response-types using the hierarchical generalized transformation model. *Bayesian Analysis* 17(1):127–164

Calandra R, Peters J, Rasmussen CE, Deisenroth MP. 2016. Manifold Gaussian processes for regression, In *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 3338–3345, Vancouver, BC, Canada: IEEE

Cartwright L, Zammit-Mangion A, Deutscher N. 2022. Emulation of greenhouse-gas sensitivities using variational autoencoders. *arXiv preprint arXiv:2112.12524*

Castruccio S, McInerney DJ, Stein ML, Liu Crouch F, Jacob RL, Moyer EJ. 2014. Statistical emulation of climate model projections based on precomputed gcm runs. *Journal of Climate* 27(5):1829–1844

Chang W, Haran M, Applegate P, Pollard D. 2016. Calibrating an ice sheet model using high-dimensional binary spatial data. *Journal of the American Statistical Association* 111(513):57–72

Chattopadhyay A, Mustafa M, Hassanzadeh P, Bach E, Kashinath K. 2022. Towards physics-inspired data-driven weather forecasting: integrating data assimilation with a deep spatial-transformer-based U-NET in a case study with ERA5. *Geoscientific Model Development* 15:2221–2237

Chen RT, Amos B, Nickel M. 2020. Neural spatio-temporal point processes. *arXiv preprint arXiv:2011.04583*

Chen RT, Rubanova Y, Bettencourt J, Duvenaud D. 2018. Neural ordinary differential equations. *arXiv preprint arXiv:1806.07366*

Chen W, Li Y, Reich BJ, Sun Y. 2021. DeepKriging: Spatially dependent deep neural networks for spatial prediction. *arXiv preprint arXiv:2007.11972*

Cressie N. 1978. The exponential and power data transformations. *Journal of the Royal Statistical Society: Series D* 27(1):57–60

Cressie N. 1993. Statistics for spatial data. Hoboken, NJ: John Wiley & Sons

Cressie N, Wikle CK. 2011. Statistics for spatio-temporal data. Hoboken, NJ: John Wiley & Sons

Damianou A, Lawrence N. 2013. Deep Gaussian processes, In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, eds. CM Carvalho, P Ravikumar, vol. 31 of *Proceedings of Machine Learning Research*, pp. 207–215, PMLR, Scottsdale, AZ

De Bézenac E, Pajot A, Gallinari P. 2019. Deep learning for physical processes: incorporating prior scientific knowledge. *Journal of Statistical Mechanics: Theory and Experiment* 2019(12):124009

De Oliveira V, Kedem B, Short DA. 1997. Bayesian prediction of transformed Gaussian random fields. *Journal of the American Statistical Association* 92(440):1422–1433

Diggle PJ, Tawn JA, Moyeed RA. 1998. Model-based geostatistics. *Journal of the Royal Statistical Society: Series C* 47(3):299–350

Dixon MF. 2021. Industrial forecasting with exponentially smoothed recurrent neural networks. *Technometrics* 64(1):114–124

Dixon MF, Polson NG, Sokolov VO. 2019. Deep learning for spatio-temporal modeling: dynamic traffic flows and high frequency trading. *Applied Stochastic Models in Business and Industry* 35(3):788–807

Duchi J, Hazan E, Singer Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12(7):2121–2159

Dunlop MM, Girolami MA, Stuart AM, Teckentrup AL. 2018. How deep are deep Gaussian processes? *Journal of Machine Learning Research* 19(54):1–46

Duvenaud D, Rippel O, Adams R, Ghahramani Z. 2014. Avoiding pathologies in very deep networks, In *Artificial Intelligence and Statistics*, pp. 202–210, PMLR

Fan J, Ma C, Zhong Y. 2021. A selective overview of deep learning. *Statistical Science* 36(2):264–290

Gal Y, Ghahramani Z. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, In *International Conference on Machine Learning*, pp. 1050–1059, PMLR, New York, NY

Gerber F, Nychka D. 2021. Fast covariance parameter estimation of spatial Gaussian process models using neural networks. *Stat* 10(1):e382

Gopalan G, Wikle CK. 2022. A higher-order singular value decomposition tensor emulator for spatiotemporal simulators. *Journal of Agricultural, Biological and Environmental Statistics* 27(1):22–45

Gramacy RB, Lee HKH. 2008. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association* 103(483):1119–1130

Gunning D, Stefik M, Choi J, Miller T, Stumpf S, Yang GZ. 2019. XAI—Explainable artificial intelligence. *Science Robotics* 4(37):aay7120

Guo S, Lin Y, Li S, Chen Z, Wan H. 2019. Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting. *IEEE Transactions on Intelligent Transportation Systems* 20(10):3913–3926

Heaton MJ, Datta A, Finley AO, Furrer R, Guinness J, et al. 2019. A case study competition among methods for analyzing large spatial data. *Journal of Agricultural, Biological and Environmental Statistics* 24(3):398–425

Henderson P, Islam R, Bachman P, Pineau J, Precup D, Meger D. 2018. Deep reinforcement learning that matters, In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32 of *AAAI'18/IAAI'18/EAAI'18*. New Orleans, Louisiana, LA: AAAI Press

Hensman J, Fusi N, Lawrence ND. 2013. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*

Hooten MB, Leeds WB, Fiechter J, Wikle CK. 2011. Assessing first-order emulator inference for physical parameters in nonlinear mechanistic models. *Journal of Agricultural, Biological, and Environmental Statistics* 16(4):475–494

Huang CW, Krueger D, Lacoste A, Courville A. 2018. Neural autoregressive flows, In *Proceedings of the 35th International Conference on Machine Learning*, eds. J Dy, A Krause, vol. 80 of *Proceedings of Machine Learning Research*, pp. 2078–2087, PMLR, Stockholm, Sweden

Huang H, Castruccio S, Genton MG. 2021a. Forecasting high-frequency spatio-temporal wind power with dimensionally reduced echo state networks. *arXiv preprint arXiv:2102.01141*

Huang H, Castruccio S, Genton MG. 2021b. Forecasting high-frequency spatio-temporal wind power with dimensionally reduced echo state networks. *arXiv preprint arXiv:2102.01141*

Huang Y, Li J, Shi M, Zhuang H, Zhu X, et al. 2021c. ST-PCNN: Spatio-temporal physics-coupled neural networks for dynamics forecasting. *arXiv preprint arXiv:2108.05940*

Jaeger H. 2001. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *GMD Report 148, German National Research Center for Information Technology*

Jaeger H. 2007a. Discovering multiscale dynamical features with hierarchical echo state networks. *Technical Report No. 10, School of Engineering and Science, Jacobs University*

Jaeger H. 2007b. Echo state network. *Scholarpedia* 2(9):2330

Jia J, Benson AR. 2019. Neural jump stochastic differential equations. *arXiv preprint arXiv:1905.10403*

Kennedy MC, O'Hagan A. 2001. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B* 63(3):425–464

Kingma DP, Ba J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*

Kingma DP, Salimans T, Jozefowicz R, Chen X, Sutskever I, Welling M. 2016. Improved variational inference with inverse autoregressive flow, In *Advances in Neural Information Processing Systems*, eds. D Lee, M Sugiyama, U Luxburg, I Guyon, R Garnett, vol. 29. Curran Associates, Inc.

Kirkwood C, Economou T, Pugeault N. 2020. Bayesian deep learning for mapping via auxiliary information: a new era for geostatistics? *arXiv preprint arXiv:2008.07320*

Klein N, Smith MS, Nott DJ. 2020. Deep distributional time series models and the probabilistic forecasting of intraday electricity prices. *arXiv preprint arXiv:2010.01844*

Leeds W, Wikle C, Fiechter J, Brown J, Milliff R. 2013. Modeling 3-D spatio-temporal biogeochemical processes with a forest of 1-D statistical emulators. *Environmetrics* 24(1):1–12

Leeds WB, Wikle CK, Fiechter J. 2014. Emulator-assisted reduced-rank ecological data assimilation for nonlinear multivariate dynamical spatio-temporal processes. *Statistical Methodology* 17:126–138

Lenzi A, Bessac J, Rudi J, Stein ML. 2021. Neural networks for parameter estimation in intractable models. *arXiv preprint arXiv:2107.14346*

Ma X, Li J, Kochenderfer MJ, Isele D, Fujimura K. 2021. Reinforcement learning for autonomous driving with latent state inference and spatial-temporal relationships, In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6064–6071, IEEE

Marmin S, Filippone M. 2022. Deep Gaussian processes for calibration of computer models. *Bayesian Analysis* 1(1):1–30

Maroñas J, Hamelijnck O, Knoblauch J, Damoulas T. 2021. Transforming Gaussian processes with normalizing flows, In *International Conference on Artificial Intelligence and Statistics*, pp. 1081–1089, PMLR. Online:`http://proceedings.mlr.press/v130/maronas21a/maronas21a.pdf`

McDermott PL, Wikle CK. 2017. An ensemble quadratic echo state network for nonlinear spatio-temporal forecasting. *Stat* 6(1):315–330

McDermott PL, Wikle CK. 2019a. Bayesian recurrent neural network models for forecasting and quantifying uncertainty in spatial-temporal data. *Entropy* 21(2):184

McDermott PL, Wikle CK. 2019b. Deep echo state networks with uncertainty quantification for spatio-temporal forecasting. *Environmetrics* 30(3):e2553

Ming D, Williamson D, Guillas S. 2021. Deep Gaussian process emulation using stochastic imputation. *arXiv preprint arXiv:2107.01590*

Mohan AT, Lubbers N, Livescu D, Chertkov M. 2020. Embedding hard physical constraints in convolutional neural networks for 3D turbulence, In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*. Online: `https://openreview.net/pdf?id=IaXBtMNFaa`

Molnar C. 2022. Interpretable Machine Learning. Independently published. Available at `https://christophm.github.io/interpretable-ml-book/`

Momenifar M, Diao E, Tarokh V, Bragg AD. 2022. A physics-informed vector quantized autoencoder for data compression of turbulent flow. *arXiv preprint arXiv:2201.03617*

Monterrubio-Gómez K, Roininen L, Wade S, Damoulas T, Girolami M. 2020. Posterior inference for sparse hierarchical non-stationary models. *Computational Statistics & Data Analysis* 148:106954

Murakami D, Kajita M, Kajita S, Matsui T. 2021. Compositionally-warped additive mixed modeling for a wide variety of non-Gaussian spatial data. *Spatial Statistics* 43:100520

Neal RM. 1996. Bayesian Learning for Neural Networks. New York, NY: Springer

Ng AY, Russell S. 2000. Algorithms for inverse reinforcement learning, In *Proceedings of the 17th International Conf. on Machine Learning*, pp. 663–670, Morgan Kaufmann

Ng TLJ, Zammit-Mangion A. 2022a. Non-homogeneous Poisson process intensity modeling and estimation using measure transport. *Bernoulli, in press*

Ng TLJ, Zammit-Mangion A. 2022b. Spherical poisson point process intensity function modeling and estimation with measure transport. *Spatial Statistics, in press*

Nguyen H, Cressie N, Braverman A. 2017. Multivariate spatial data fusion for very large remote sensing datasets. *Remote Sensing* 9(2):142

Nguyen N, Tran MN, Gunawan D, Kohn R. 2019. A long short-term memory stochastic volatility model. *arXiv preprint arXiv:1906.02884*

Oh J, Guo X, Lee H, Lewis R, Singh S. 2015. Action-conditional video prediction using deep networks in atari games. *arXiv preprint arXiv:1507.08750*

Paciorek CJ, Schervish MJ. 2006. Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics* 17(5):483–506

Papamakarios G, Pavlakou T, Murray I. 2017. Masked autoregressive flow for density estimation, In *Advances in Neural Information Processing Systems*, eds. I Guyon, UV Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, R Garnett, vol. 30. Curran Associates, Inc.

Parker PA, Holan SH, Wills SA. 2021. A general Bayesian model for heteroskedastic data with fully conjugate full-conditional distributions. *Journal of Statistical Computation and Simulation* 91(15):3207–3227

Paszke A, Gross S, Chintala S, Chanan G, Yang E, et al. 2017. Automatic differentiation in PyTorch. `https://openreview.net/forum?id=BJJsrmfCZ`

Perrin O, Monestiez P. 1999. Modelling of non-stationary spatial structure using parametric radial basis deformations. In *GeoENV II–Geostatistics for Environmental Applications*, eds. J Gómez-Hernández, A Soares, R Froidevaux. Springer, New York, NY, 175–186

Quiñonero-Candela J, Rasmussen CE. 2005. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research* 6:1939–1959

R Core Team. 2022. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria

Raissi M, Yazdani A, Karniadakis GE. 2020. Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. *Science* 367(6481):1026–1030

Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, et al. 2019. Deep learning and process understanding for data-driven earth system science. *Nature* 566(7743):195–204

Rezende DJ, Mohamed S. 2015. Variational inference with normalizing flows, In *International Conference on Machine Learning*, pp. 1530–1538, PMLR. Online: `https://dl.acm.org/doi/10.5555/3045118.3045281`

Rezende DJ, Papamakarios G, Racaniere S, Albergo M, Kanwar G, et al. 2020. Normalizing flows on tori and spheres, In *Proceedings of the 37th International Conference on Machine Learning*, eds. H Daumé III, A Singh, vol. 119 of *Proceedings of Machine Learning Research*, pp. 8083–8092, PMLR

Rios G, Tobar F. 2019. Compositionally-warped Gaussian processes. *Neural Networks* 118:235–246

Rudi J, Bessac J, Lenzi A. 2021. Parameter estimation with dense and convolutional neural networks applied to the FitzHugh-Nagumo ODE, In *2nd Annual Conference on Mathematical and Scientific Machine Learning*, vol. 145 of *Proceedings of Machine Learning Research*, pp. 781–808, PMLR. Online: `https://msml21.github.io/papers/id54.pdf`

Rudin C, Chen C, Chen Z, Huang H, Semenova L, Zhong C. 2022. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistical Surveys* 16:1–85

Salimbeni H, Deisenroth M. 2017. Doubly stochastic variational inference for deep Gaussian processes, In *Advances in Neural Information Processing Systems*, eds. I Guyon, UV Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, R Garnett, vol. 30, pp. 4588–4599, California, CA: Curran Associates, Inc.

Samek W, Montavon G, Lapuschkin S, Anders CJ, Müller KR. 2021. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE* 109(3):247–278

Sampson PD, Guttorp P. 1992. Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Statistical Association* 87(417):108–119

Särkkä S, Solin A. 2019. Applied stochastic differential equations, vol. 10. Cambridge, UK: Cambridge University Press

Sauer A, Cooper A, Gramacy RB. 2022. Vecchia-approximated deep Gaussian processes for computer experiments. *arXiv preprint arXiv:2204.02904*

Schafer TL, Wikle CK, Hooten MB. 2020. Bayesian inverse reinforcement learning for collective animal movement. *arXiv preprint arXiv:2009.04003*

Schmidt AM, O'Hagan A. 2003. Bayesian inference for non-stationary spatial covariance structure via spatial deformations. *Journal of the Royal Statistical Society: Series B* 65(3):743–758

Sei T. 2013. A Jacobian inequality for gradient maps on the sphere and its application to directional statistics. *Communications in Statistics – Theory and Methods* 42(14):2525–2542

Shi X, Chen Z, Wang H, Yeung DY, Wong Wk, Woo Wc. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting, In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, vol. 1 of *NIPS'15*, p. 802–810, Cambridge, MA: MIT Press

Sidén P, Lindsten F. 2020. Deep Gaussian Markov random fields, In *Proceedings of the 37th International Conference on Machine Learning*, eds. H Daumé III, A Singh, vol. 119 of *Proceedings of Machine Learning Research*, pp. 8916–8926, PMLR. Online: `https://proceedings.mlr.press/v119/siden20a.html`

Smith RL. 1996. Estimating nonstationary spatial correlations. Online: Available from `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.40.5988&rep=rep1&type=pdf`

Snelson E, Rasmussen CE, Ghahramani Z. 2004. Warped Gaussian processes. *Advances in Neural Information Processing Systems* 16:337–344

Snoek J, Swersky K, Zemel R, Adams R. 2014. Input warping for Bayesian optimization of non-stationary functions, In *Proceedings of the 31st International Conference on Machine Learning*, eds. EP Xing, T Jebara, vol. 32 of *Proceedings of Machine Learning Research*, pp. 1674–1682, PMLR, Bejing, China

Song C, Lin Y, Guo S, Wan H. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting, In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 914–921, New York, NY

Sutton RS, Barto AG. 1998. Reinforcement learning: An introduction. Massachusetts, MA: MIT Press

Tabak E, Vanden Eijnden E. 2010. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences* 8(1):217–233

Taddy M, Kottas A. 2010. Mixture modeling for marked Poisson processes. *Bayesian Analysis* 7(2):335–362

Tampuu A, Matiisen T, Kodelja D, Kuzovkin I, Korjus K, et al. 2017. Multiagent cooperation and competition with deep reinforcement learning. *PloS One* 12(4):e0172395

Tran MN, Nguyen N, Nott D, Kohn R. 2020. Bayesian deep net GLM and GLMM. *Journal of Computational and Graphical Statistics* 29(1):97–113

Tukey JW. 1977. Modern techniques in data analysis, In *Proceedings of the NSF-Sponsored Regional Research Conference*, vol. 7. Southern Massachusetts University, Massachusetts, MA

Vu Q, Moores MT, Zammit-Mangion A. 2022a. Warped gradient-enhanced Gaussian process surrogate models for inference with intractable likelihoods. *arXiv preprint arXiv:2105.04374*

Vu Q, Zammit-Mangion A, Chuter SJ. 2022b. Constructing large nonstationary spatio-temporal covariance models via compositional warpings. *arXiv preprint arXiv:2202.03560*

Vu Q, Zammit-Mangion A, Cressie N. 2022c. Modeling nonstationary and asymmetric multivariate spatial covarianves via deformations. *Statistica Sinica, in press*

Wang H, Guan Y, Reich B. 2019. Nearest-neighbor neural networks for geostatistics, In *2019 International Conference on Data Mining Workshops (ICDMW)*, pp. 196–205, IEEE, Beijing, China

Wang J, Yang Y, Mao J, Huang Z, Huang C, Xu W. 2016. CNN-RNN: A unified framework for multi-label image classification, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2285–2294, Las Vegas, NV

Wikle CK. 2019. Comparison of deep neural networks and deep hierarchical models for spatio-temporal data. *Journal of Agricultural, Biological and Environmental Statistics* 24(2):175–203

Wikle CK, Hooten MB. 2010. A general science-based framework for dynamical spatio-temporal models. *Test* 19(3):417–451

Wikle CK, Zammit-Mangion A, Cressie N. 2019a. Spatio-temporal statistics with R. Boca Raton, FL: Chapman and Hall/CRC Press

Wikle CK, Zammit-Mangion A, Cressie N. 2019b. Spatio-temporal statistics with R (supplementary R package). Online: `https://github.com/andrewzm/STRbook`

Wu JL, Kashinath K, Albert A, Chirila D, Xiao H, et al. 2020. Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems. *Journal of Computational Physics* 406:109209

Xu G, Genton MG. 2017. Tukey g-and-h random fields. *Journal of the American Statistical Association* 112(519):1236–1249

Yu B, Yin H, Zhu Z. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*

Zammit-Mangion A, Ng TLJ, Vu Q, Filippone M. 2021. Deep compositional spatial models. *Journal of the American Statistical Association* doi:10.1080/01621459.2021.1887741

Zammit-Mangion A, Wikle CK. 2020. Deep integro-difference equation models for spatio-temporal forecasting. *Spatial Statistics* 37:100408

Zhang B, Konomi BA, Sang H, Karagiannis G, Lin G. 2015. Full scale multi-output Gaussian process emulator with nonseparable auto-covariance functions. *Journal of Computational Physics* 300:623–642

Zhao Z, Emzir M, Särkkä S. 2021. Deep state-space Gaussian processes. *Statistics and Computing* 31(75):1–26

Zhu S, Li S, Peng Z, Xie Y. 2020. Interpretable deep generative spatio-temporal point processes, In *Proceedings of the NeurIPS Workshop AI for Earth Sciences*. Online: https://ai4earthscience.github.io/neurips-2020-workshop/papers/ai4earth_neurips_2020_09.pdf