

Scalable spatiotemporal prediction with Bayesian neural fields

Received: 19 February 2024

Accepted: 7 August 2024

Published online: 11 September 2024

 Check for updates


Feras Saad^{1,2}  , Jacob Burnim², Colin Carroll², Brian Patton², Urs Köster², Rif A. Saurous² & Matthew Hoffman²

Spatiotemporal datasets, which consist of spatially-referenced time series, are ubiquitous in diverse applications, such as air pollution monitoring, disease tracking, and cloud-demand forecasting. As the scale of modern datasets increases, there is a growing need for statistical methods that are flexible enough to capture complex spatiotemporal dynamics and scalable enough to handle many observations. This article introduces the Bayesian Neural Field (BAYESNF), a domain-general statistical model that infers rich spatiotemporal probability distributions for data-analysis tasks including forecasting, interpolation, and variography. BAYESNF integrates a deep neural network architecture for high-capacity function estimation with hierarchical Bayesian inference for robust predictive uncertainty quantification. Evaluations against prominent baselines show that BAYESNF delivers improvements on prediction problems from climate and public health data containing tens to hundreds of thousands of measurements. Accompanying the paper is an open-source software package (<https://github.com/google/bayesnf>) that runs on GPU and TPU accelerators through the JAX machine learning platform.

Spatiotemporal data, which consists of measurements gathered at different times and locations, is ubiquitous across diverse disciplines. Government bodies such as the European Environment Agency¹ and United States Environmental Protection Agency², for example, routinely monitor a variety of air quality indicators (PM₁₀, NO₂, O₃, etc.) in order to understand their ecological and public health impacts^{3,4}. As it is physically impossible to place sensors at all locations in a large geographic area, environmental data scientists routinely develop statistical models to predict these indicators at new locations or times where no data is available^{5,6}. Spatiotemporal data analysis also plays an important role in cloud computing, where consumer demand for resources such as CPU, RAM, and storage is driven by time-evolving macroeconomic factors and varies across data center location. Cloud service providers build sophisticated demand-forecasting models to determine prices⁷, perform load balancing⁸, save energy⁹, and achieve service level agreements¹⁰. Additional applications of spatiotemporal data analysis include meteorology (forecasting rain volume¹¹ or wind speeds¹²), epidemiology (“nowcasting” active flu cases¹³), and urban planning (predicting rider congestion patterns at metro stations¹⁴).

Unlike traditional regression or classification methods in machine learning that operate on independent and identically distributed (i.i.d.) data, accurate models of spatiotemporal data must capture complex and highly nonstationary dynamics in both the time and space domains. For example, two locations twenty miles apart in California’s central valley may exhibit nearly identical temperature patterns, whereas two locations only one mile apart in nearby San Francisco might have very different microclimates; and these effects may differ depending on the time of year. Handling such variability across different scales is a key challenge in designing accurate statistical models. Another challenge is that spatiotemporal observations are typically driven by unknown and noisily observed data-generating processes, which require models that report probabilistic predictions to account for the aleatoric and epistemic uncertainty in the data.

The dominant approach to spatiotemporal data modeling in statistics rests on *Gaussian processes*, a rich class of Bayesian nonparametric priors on random functions^{15–17}. Consider a spatiotemporal field $Y(\mathbf{s}, t)$ indexed by spatial locations $\mathbf{s} \in \mathbb{R}^d$ and time points $t \in \mathbb{R}$. A typical Gaussian-process based “prior probability” distribution (used in

¹Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA. ²Google Research, Mountain View, CA, USA.  e-mail: fsaad@cmu.edu

popular geostatistical software packages such as R-INLA¹⁸ and sdm-TMB¹⁹) over the random field Y is given by:

$$\eta \sim \text{GP}(0, k_\theta); \quad F(\mathbf{s}, t) = h(x(\mathbf{s}, t); \beta) + \eta(\mathbf{s}, t); \quad Y(\mathbf{s}, t) \sim \text{Dist}(g(F(\mathbf{s}, t)), \gamma). \quad (1)$$

In Eq. (1), η is a random function whose covariance over space and time is determined by a *kernel function* $k_\theta((\mathbf{s}, t), (\mathbf{s}', t'))$ parameterized by θ ; $x(\mathbf{s}, t)$ is a covariate vector associated with index (\mathbf{s}, t) ; h is a mean function with parameters β (e.g., for a linear function, $h(x; \beta) = \beta'x$) of the latent field F ; and Dist is a noise model (e.g., Normal, Poisson) for the observations $Y(\mathbf{s}, t)$, with index-specific parameter $g(F(\mathbf{s}, t))$ (where g is a link function, e.g., exp) and global parameters γ .

Given an observed dataset $\mathcal{D} = \{Y(\mathbf{s}_1, t_1) = y_1, \dots, Y(\mathbf{s}_N, t_N) = y_N\}$, the inference problem is to determine the unknown parameters (θ, β, γ) , which in turn define a *posterior distribution* over the processes (η, F, Y) given \mathcal{D} . Advantages of the model (1) are (i) its flexibility, as η is capable of representing highly complex covariance structure; and (ii) its ability to quantify uncertainty, as the posterior spreads its probability mass over a range of functions and model parameters that are consistent with the data. Moreover, the model easily handles arbitrary patterns of missing data by treating them as latent variables. A number of recent articles have developed specialized Gaussian process techniques for modeling rich spatiotemporal fields e.g., refs. 19–23.

Despite their flexibility, spatiotemporal models based on Gaussian processes (such as Eq. (1)) come with significant challenges. The first is computational. The simplest and most accurate posterior inference algorithms for these models have a computational cost of $O(N^3)$, where N is the number of observations, which is unacceptably high in datasets with tens or hundreds of thousands of observations. Reducing this cost requires compromises, either on the modeling side (e.g., imposing a discrete Markovian structure on the model^{18,19}) or on the posterior-inference side (e.g., approximating the true posterior with a simpler Gaussian process^{20,21,23}). Either way, the resulting models have less expressive power and cannot explain the data as accurately. These approximations also involve delicate linear-algebraic derivations or stochastic differential equations, which are challenging to implement and apply to new settings.

The second challenge is expertise, where the accuracy of model (1) on a given dataset is dictated by key choices such as the covariance kernel k_θ and mean function h . Even for seasoned data scientists, designing these quantities is difficult because it requires detailed knowledge about the application domain. Further, even small modifications to the model can impose large changes to the learning algorithm, and so most software packages only support a small set of predetermined covariance structures k_θ (e.g., separable Matérn kernels, radial basis kernel, polynomial kernel) that are optimized enough to work effectively on large datasets.

To alleviate these fundamental tensions, this article introduces the *Bayesian Neural Field* (BAYESNF)—a method that combines the scalability of deep neural networks with many of the attractive properties of Gaussian processes. BAYESNF is built on a Bayesian neural network model²⁴ that maps from multivariate space-time coordinates to a real-valued field. The parameters of the network are assigned a prior distribution, and as in Gaussian processes, conditioning on observed data induces a posterior over those parameters (and in turn over the entire field). Because inference is performed in “weight space” rather than “function space”, the cost of analyzing a dataset grows linearly with the number of observations, as opposed to cubically for a Gaussian process. Because BAYESNF is a hierarchical model (Fig. 1), it naturally handles missing data as latent variables and quantifies uncertainty over parameters and predictions. And because BAYESNF defines a field over continuous space–time, it can model non-uniformly sampled data, interpolate in space, and extrapolate in time to make predictions at novel coordinates.

Our description of BAYESNF as a neural “field” is inspired by the recent literature on neural radiance fields (NeRFs^{25,26}) in computer vision. A key discovery that enabled the success of NeRFs is that neural networks are biased towards learning functions whose Fourier spectra are dominated by low frequencies, and that this bias can be corrected by concatenating sinusoidal positional encodings to the raw spatial inputs²⁷. To ensure that our BAYESNF model assigns high prior probability to data that includes both low- and high-frequency variation, we append Fourier features to the raw time and position data that are fed to the network. In Methods, we show that these Fourier features, coupled with learned scale factors and convex combinations of activation functions, improve BAYESNF models’ ability to learn flexible and well-calibrated distributions of spatiotemporal data. Incorporating sinusoidal seasonality features lets BAYESNF models make predictions based on (multiple) seasonal effects as well. Taken together, these characteristics enable state-of-the-art performance in terms of point predictions and 95% prediction intervals on diverse large-scale spatiotemporal datasets, without the need to heavily customize the BAYESNF model structures on a per-dataset basis.

BAYESNF belongs to a family of emerging techniques that leverage deep neural networks with hierarchical Bayesian models for spatiotemporal data analysis—a thorough survey of these advances is given in Wikle and Zammit-Mangion²⁸. Our method is inspired by limitations of existing deep neural network approaches for probabilistic prediction in spatiotemporal data. For example, the Bayesian spatiotemporal recurrent neural networks introduced in McDermott and Wikle²⁹ require the data to be observed at a fixed spatial grid and regular discrete-time intervals. In contrast, BAYESNF is defined over continuous space–time coordinates, enabling prediction at novel locations and in datasets with irregularly sampled time points. The deep “Empirical Orthogonal Function” model³⁰ is a powerful exploratory analysis tool but is less useful for prediction: it cannot handle missing data, make predictions at new time points, or deliver uncertainty estimates. Additional methods in this category include Bayesian neural networks that are highly task oriented—e.g., for analyzing power flow³¹, wind speed³², or floater intrusion risk³³. These methods leverage domain-specific architectures designed specifically for the analysis problem at hand, and do not aim to provide software libraries that are easy for practitioners to apply in new spatiotemporal datasets beyond the application domain. In contrast, a central goal of BAYESNF is to provide a domain-general modeling tool that is easily applicable to the same type of datasets as the Gaussian process model (1), without the need to redesign substantial parts of the probabilistic model or network architecture for each new task.

Neural processes³⁴ also integrate deep neural networks with probabilistic modeling, but are based on a graphical model structure that is fundamentally difficult to apply to spatiotemporal datasets. In particular, because neural processes aim to “meta-learn” a prior distribution over random functions, the authors note it is essential to have access to a large number of independent and identically distributed (i.i.d.) datasets during training. However, most spatiotemporal data analyses are based on only a single real-world dataset (e.g., those in Table 1) where there is no notion of sharing statistical strength across multiple i.i.d. observations of the entire field.

Graph neural networks (GNNs), surveyed in Jin et al.³⁵, are another popular deep-learning approach for spatiotemporal prediction which have been particularly useful in settings such as analyzing traffic or population-migration patterns. These models require as input a graph describing the connectivity structure of the spatial locations, which makes them less appropriate for spatial data that lack such discrete connectivity structure. Moreover, the requirement that the graph be fixed makes it harder for GNNs to interpolate or extrapolate to locations that are not included in the graph at training time. The BAYESNF model, on the other hand, operates over continuous space, and is therefore more appropriate for spatial data

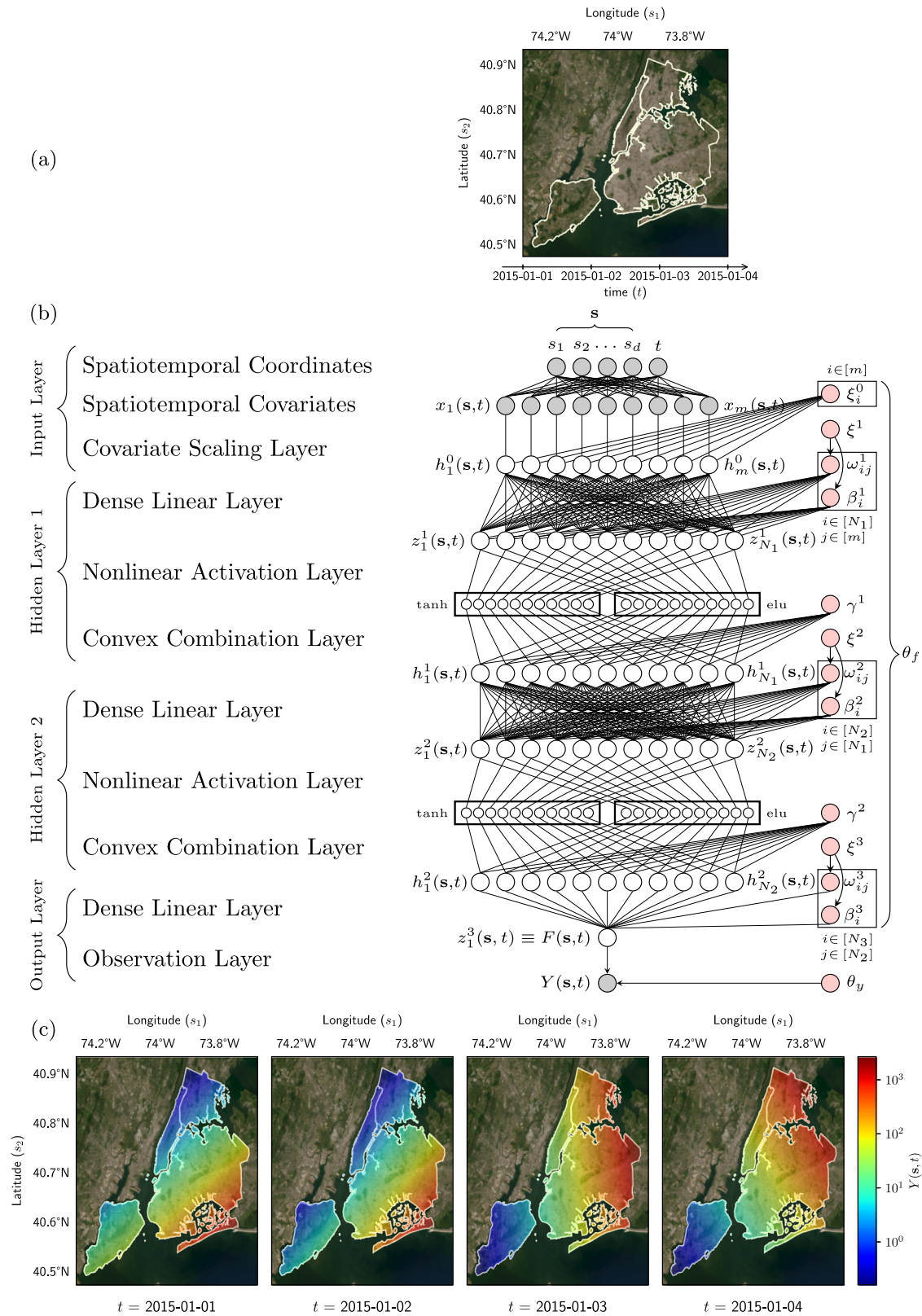


Fig. 1 | Probabilistic graphical model representation of the Bayesian Neural Field. **a** An example spatiotemporal domain comprised of two spatial coordinates (latitude, longitude) and a daily time coordinate. **b** In the probabilistic graphical model, each node denotes a model variable and each edge denotes a direct relationship between a pair of variables. Gray nodes are observed variables and white nodes are local latent variables, which are both associated with an

observation $Y(\mathbf{s}, t)$ at a spatiotemporal coordinate (\mathbf{s}, t) . Pink nodes are global latent variables (parameters), which are shared across all spatiotemporal coordinates. **c** Realizations of the spatiotemporal field generated from the BAYESNF at four example time points. Satellite basemap source: Esri, DigitalGlobe, GeoEye, i-cubed, USDA FSA, USGS, AEX, Getmapping, Aerogrid, IGN, IGP, swisstopo, and the GIS User Community¹⁸.

Table 1 | Spatiotemporal datasets analyzed in the empirical evaluation

| Dataset | Region | Frequency | Locations | Time Points | Observations | Missing | Start | End |
|---------------------------------------|---------------|-----------|-----------|-------------|--------------|---------|------------|------------|
| Wind Speed ³⁸ | Ireland | Daily | 12 | 6574 | 78,888 | 0% | 1961-01-01 | 1978-12-31 |
| Air Quality 1 ³⁹ | Germany | Daily | 70 | 4383 | 149,151 | 52% | 1998-01-01 | 2009-12-31 |
| Air Quality 2 ²⁰ | London | Hourly | 72 | 2159 | 144,570 | 7% | 2018-12-31 | 2019-03-31 |
| Chickenpox Cases ⁴⁰ | Hungary | Weekly | 20 | 522 | 10,440 | 0% | 2005-01-03 | 2014-12-29 |
| Precipitation ⁴¹ | Colorado | Monthly | 358 | 576 | 134,800 | 35% | 1950-01-01 | 1997-12-01 |
| Sea Surface Temperature ¹⁷ | Pacific Ocean | Monthly | 2261 | 399 | 902,139 | 0% | 1970-01-01 | 2003-03-01 |

without known discrete connectivity structure. In addition, as noted in Jin et al.³⁵, GNNs have not yet been demonstrated on probabilistic prediction tasks, and we are unaware of the existence of open-source software libraries based on GNNs that can easily handle the sparse datasets in Table 1.

Results

Model description

Consider a dataset $\mathcal{D} = \{y(\mathbf{s}_i, t_i) | i = 1, \dots, N\}$ of N spatiotemporal observations, where $\mathbf{s}_i \in \mathcal{S} \subset \mathbb{R}^d$ denotes a d -dimensional spatial coordinate and $t_i \in \mathcal{T} \subset \mathbb{R}$ denotes a time index. For example, if the field is observed at longitude-latitude coordinates in discrete time, then $\mathcal{S} = (-180, 180] \times [-90, 90] \subset \mathbb{R}^2$ and $\mathcal{T} = \{1, 2, \dots\}$. If the field also incorporates an altitude dimension, then $\mathcal{S} \subset \mathbb{R}^3$. We model this dataset as a realization $\{Y(\mathbf{s}_i, t_i) = y(\mathbf{s}_i, t_i), 1 \leq i \leq N\}$ of a random field $Y : \mathcal{S} \times \mathcal{T} \rightarrow \mathbb{R}$ over the entire spatiotemporal domain. Following the notation in Wikle and Zammit-Mangion²⁸, we describe the field using a hierarchical Bayesian model:

$$\text{Observation Model: } [Y(\cdot) | F(\cdot), \Theta_y], \tag{2}$$

$$\text{Process Model: } [F(\cdot) | x(\cdot), \Theta_f], \tag{3}$$

$$\text{Parameter Models: } [\Theta_y, \Theta_f]. \tag{4}$$

In this notation, upper case letters denote random quantities, Greek letters denote model parameters, lower case letters denoted non-random (fixed) quantities, and square brackets $[\cdot]$ denote (yet-to-specified) probability distributions. The distribution of the observable random variables $Y(\mathbf{s}, t)$ is parameterized by global parameters Θ_y and an unobservable (latent) spatiotemporal field $F(\mathbf{s}, t)$. In turn, $F(\mathbf{s}, t)$ is parameterized by a set of random global parameters Θ_f and a

collection $x(\mathbf{s}, t) = [x_1(\mathbf{s}, t), \dots, x_m(\mathbf{s}, t)]$ of m fixed covariates associated with index (\mathbf{s}, t) .

Box 1 completes the definition of BAYESNF by showing specific probability distributions for the model (2)–(4). Figure 1 shows a probabilistic-graphical-model representation of a BAYESNF model with $H = 3$ layers, which takes a spatiotemporal index (\mathbf{s}, t) at the input layer and generates a realization $Y(\mathbf{s}, t)$ of the observable field at the output layer. At a high level, the input layer transforms the spatiotemporal coordinates (\mathbf{s}, t) into a fixed set of spatiotemporal covariates, which include linear terms, interaction terms, and Fourier features in time and space. The second layer performs a linear scaling of these covariates using a learnable scale factor—this layer aims to avoid the need for the practitioner to manually specify how to appropriately scale the data, which is known to heavily influence the learning dynamics³⁶. Next, the hidden layers of the network contain the usual dense connections, except that the activations are specified as a learnable convex combination of “primitive” activations, such as rectified linear units (relu), exponential linear unit (elu), or hyperbolic tangent (tanh). The goal of these convex combinations is to automate the discovery of the covariance structure in the field, given that activation functions correspond directly to covariance of random functions defined by Bayesian neural networks³⁷. At the final layer, the output of the feedforward network is used to parameterize a probability distribution over the observable field values, which serves to capture the fundamental aleortic uncertainty in the noisy data. Epistemic uncertainty in BAYESNF is expressed by assigning prior probability distributions to all learnable parameters, such as covariate scale factors; connection weights, biases, and their variances; and additional parameters of the observation distribution.

We next describe the components of this process in sequence from inputs to outputs in more detail. This description defines a *prior* distribution over Bayesian Neural Fields—in Methods we discuss ways of inferring the *posterior* over the random variables defined in Box 1.

BOX 1

Generative process for the Bayesian Neural Field in Fig. 1. Global parameters are shared by all locations in the field. Local latent variables are associated with a given spatiotemporal index (\mathbf{s}, t)

$$\text{Configuration} \begin{cases} L & : \text{number of internal layers} \\ x_i & : \text{covariate functions } (1 \leq i \leq m) \\ N^\ell & : \text{number of hidden units } (1 \leq \ell \leq L) \\ A^\ell & : \text{number of activation functions } (1 \leq \ell \leq L) \\ u_j^\ell & : \text{activation functions } (1 \leq \ell \leq L; 1 \leq j \leq A^\ell) \\ \text{Dist} & : \text{observation distribution} \end{cases}$$

Covariate Scaling Layer

$$(\xi_1^0, \dots, \xi_m^0) \sim \text{iid Normal}(0, 1) \quad h^0(\mathbf{s}, t) := (e^{\xi_1^0} x_1(\mathbf{s}, t), \dots, e^{\xi_m^0} x_m(\mathbf{s}, t))$$

Hidden Layers $(\ell = 1, \dots, L+1; i = 1, \dots, N^\ell)$

$$(\xi^{\ell}, \gamma_1^{\ell}, \dots, \gamma_{A^{\ell}}^{\ell}) \sim \text{iid Normal}(0, 1), \quad z_i^{\ell}(\mathbf{s}, t) := \sum_{j=1}^{N^{\ell-1}} \frac{\omega_{ij}^{\ell}}{\sqrt{N^{\ell-1}}} h_j^{\ell-1}(\mathbf{s}, t) + \beta_i^{\ell}$$

$$(\omega_{i1}^{\ell}, \dots, \omega_{iN^{\ell-1}}^{\ell}, \beta_i^{\ell}) \sim \text{iid Normal}(0, \sigma^{\ell}) \quad h_i^{\ell}(\mathbf{s}, t) := \sum_{j=1}^{A^{\ell}} \frac{e^{u_j^{\ell}}}{\sum_{k=1}^{A^{\ell}} e^{u_k^{\ell}}} u_j^{\ell}(z_i^{\ell}(\mathbf{s}, t))$$

$$\text{where } \sigma^{\ell} := \ln(1 + e^{\xi^{\ell}}) \quad (\text{only if } \ell < L + 1)$$

Observation Layer

$$\Theta_y \sim \pi_y \quad F(\mathbf{s}, t) := z_1^{L+1}(\mathbf{s}, t) \\ Y(\mathbf{s}, t) \sim \text{Dist}(F(\mathbf{s}, t), \Theta_y)$$

Spatiotemporal covariates. Letting $(\mathbf{s}, t) = ((s_1, \dots, s_d), t)$ denote a generic index in the field, the covariates $[x_1(\mathbf{s}, t), \dots, x_m(\mathbf{s}, t)]$ may include the following functions:

$$\{t, s_1, \dots, s_d\} \text{ Linear Terms} \tag{5}$$

$$\{ts_1, \dots, ts_d\} \text{ Temporal – Spatial Interactions} \tag{6}$$

$$\{s_i s_j; 1 \leq i < j \leq d\} \text{ Spatial – Spatial Interactions} \tag{7}$$

$$\{(\cos(2\pi h t / p_t), \sin(2\pi h t / p_t)); p \in \mathcal{P}, h \in \mathcal{H}_p^t\} \text{ Temporal Seasonal Features} \tag{8}$$

$$\{(\cos(2\pi 2^h s_i), \sin(2\pi 2^h s_i)); 1 \leq i \leq d, h \in \mathcal{H}_i^s\} \text{ Spatial Fourier Features} \tag{9}$$

The linear and interaction covariates (5)–(7) are the usual first and second-order effects used in spatiotemporal trend-surface analysis models (Section 3.2 of ref. 17). In Eq. (8), the temporal seasonal features are defined by a set $\mathcal{P} = \{p_1, \dots, p_\ell\}$ of ℓ seasonal periods, where each p_i has harmonics $\mathcal{H}_{p_i}^t \subset \{1, 2, \dots, \lfloor p_i/2 \rfloor\}$ for $i = 1, \dots, \ell$. For example, if the time unit is hourly data and there are $m = 2$ seasonal effects (daily and monthly), the corresponding periods are $p_1 = 24$ and $p_2 = 730.5$, respectively. Non-integer periodicities handle seasonal effects that have varying duration in the time measurement unit (e.g., days per month or weeks per year). The Methods section discusses how to construct appropriate seasonal features for a variety of time units and seasonal effect combinations. In Eq. (9), the spatial Fourier features for coordinate s_i are determined by a set $\mathcal{H}_i^s \subset \mathbb{N}$ of additional frequencies that capture periodic structure in the i th dimension ($i = 1, \dots, d$). These covariates correct for the tendency of neural networks to learn low-frequency signals²⁷; the empirical evaluation in the next section confirms that their presence greatly improves the quality of learned models. Covariates may also include static (e.g., “continent”) or dynamic (e.g., “temperature”) exogenous features, provided they are known at all locations and time points in the training and testing datasets.

Covariate scaling layer. Scaling inputs improves neural network learning e.g., ref. 36, but determining the appropriate strategy (e.g., z-score, min/max, tanh, batch-norm, layer-norm, etc.) is challenging. BAYESNF uses a prior distribution over scale factors to learn these quantities as part of Bayesian inference within the overall probabilistic model. In particular, the next stage in the network is a width- m hidden layer $h_i^0(\mathbf{s}, t) = e^{c_i} x_i(\mathbf{s}, t)$ obtained by randomly scaling each of the m covariates $x(\mathbf{s}, t)$, where e^{c_i} is a log-normally distributed scale factor (for $i = 1, \dots, m$).

Hidden layers. The model contains $L + 1 \geq 1$ hidden layers, where layer l has N^ℓ units $h^\ell = (h_1^\ell, \dots, h_{N^\ell}^\ell)'$ (for $l = 1, \dots, L$). These hidden units are derived from N^ℓ pre-activation units $z^\ell = N_{\ell-1}^{-1/2} \Omega^\ell h^{\ell-1} + \beta^\ell$ where $\Omega^\ell = [\omega_{ij}^\ell; 1 \leq i \leq N^\ell, 1 \leq j \leq N_{\ell-1}]$ is a random $N^\ell \times N_{\ell-1}$ weight matrix and $\beta^\ell = (\beta_1^\ell, \dots, \beta_{N^\ell}^\ell)'$ a random bias term. The network parameters ω_{ij}^ℓ and β_j^ℓ are drawn i.i.d. $N(0, \sigma^\ell)$, where the variance $\sigma^\ell = \ln(1 + e^{c^\ell})$ a learnable parameter whose prior is obtained by applying a softplus transformation to $\xi^\ell \sim N(0, 1)$. The $N_{\ell-1}^{-1/2}$ prefactor ensures the network has a well-defined Gaussian process limit as the number of hidden units $N^\ell \rightarrow \infty$ ²⁴.

In addition to the covariate scaling layer, BAYESNF departs from a traditional Bayesian neural network by using $A^\ell \geq 1$ activation functions ($u_1^\ell, \dots, u_{A^\ell}^\ell$) at hidden layer l , instead of the usual $A^\ell = 1$. For example,

the architecture shown in Fig. 1 uses $A^\ell = 2$ where u_1^ℓ is the hyperbolic tangent (tanh) and u_2^ℓ is the exponential linear unit (elu) activation (where $l = 1, 2$). Each post-activation unit h_i^ℓ (for $i = 1, \dots, N^\ell$) is then a random convex combination of the activations $u_1^\ell(z_i^\ell), \dots, u_{A^\ell}^\ell(z_i^\ell)$, where the coefficient of u_j^ℓ is the output of a softmax function $e^{y_j^\ell} / \sum_{k=1}^{N_d} e^{y_k^\ell}$ whose j -th input is $y_j^\ell \sim N(0, 1)$ (for $j = 1, \dots, A^\ell$). The activation function governs the overall covariance properties of the random function defined by a Bayesian neural network^{24,37}. By specifying the overall activation at each layer as a learnable convex combination of A^ℓ “basic” activation functions (e.g., tanh, relu, elu), BAYESNF aims to automate the process of selecting an appropriate activation and in turn the covariance structure within the random field.

Finally, the latent stochastic process $F(\mathbf{s}, t)$ is defined as the pre-activation unit z_1^{L+1} of layer $L + 1$, which has exactly $N^{L+1} = 1$ unit. We let Θ_f denote all n_f random network parameters in Box 1 and denote the prior as π_f . Further, the notation $F_{\theta_f}(\mathbf{s}, t)$ denotes the (deterministic) value of the process F at index (\mathbf{s}, t) when $\Theta_f = \theta_f$.

Observation layer. The final layer connects the stochastic process $F(\mathbf{s}, t)$ with the observable spatiotemporal field $Y(\mathbf{s}, t) \sim \text{Dist}(F(\mathbf{s}, t); \Theta_y)$ through a noise model that captures aleatoric uncertainty in the data. The parameter vector $\Theta_y = (\Theta_{y,1}, \dots, \Theta_{y,n_y})$ is n_y -dimensional and has a prior π_y . There are many choices for this distribution, depending on the field $Y(\mathbf{s}, t)$; for example,

$$Y(\mathbf{s}, t) \sim \text{Normal}(F(\mathbf{s}, t), \Theta_{y,1}), \tag{10}$$

$$Y(\mathbf{s}, t) \sim \text{StudentT}_{\Theta_{y,2}}(F(\mathbf{s}, t), \Theta_{y,1}), \tag{11}$$

$$Y(\mathbf{s}, t) \sim \text{Poisson}(e^{F(\mathbf{s}, t)}), \tag{12}$$

which correspond to a Gaussian noise model with mean $F(\mathbf{s}, t)$ and variance $\Theta_{y,1}$ ($n_y = 1$), a StudentT model with location $F(\mathbf{s}, t)$, scale $\Theta_{y,1}$ and $\Theta_{y,2}$ degrees of freedom ($n_y = 2$); and a Poisson counts model with rate $\exp F(\mathbf{s}, t)$ ($n_y = 0$), respectively. A key design choice in these observation distributions is that certain parameters such as $\Theta_{y,1}$ in Eq. (10) or $\Theta_{y,1}, \Theta_{y,2}$ in Eq. (11) are not index-specific but rather shared across all inputs, which serves to mitigate the model’s sensitivity to over-fitting noise fluctuations from high-frequency Fourier features.

Posterior inference and querying. Let $P(\Theta_f, \Theta_y, Y)$ be the joint probability distribution over the parameters and observable field in Box 1. The posterior distribution given \mathcal{D} is

$$P(\theta_f, \theta_y | \{Y(\mathbf{s}_i, t_i) = y(\mathbf{s}_i, t_i)\}_{i=1}^N) \propto \left(\prod_{i=1}^{n_f} \pi_f(\theta_{f,i}) \right) \left(\prod_{i=1}^{n_y} \pi_y(\theta_{y,i}) \right) \prod_{i=1}^n \text{Dist}(y(\mathbf{s}_i, t_i); F_{\theta_f}(\mathbf{s}_i, t_i), \theta_y) \tag{13}$$

While the right-hand side of Eq. (13) is tractable to compute, the left-hand side cannot be normalized or sampled from exactly. In the Posterior Inference section of Methods, we discuss two approximate posterior inference algorithms for BAYESNF: maximum a-posteriori ensembles and variational inference ensembles. They each produce a collection of parameters $\{(\theta_f^i, \theta_y^i)\}_{i=1}^M \approx P(\Theta_f, \Theta_y | \mathcal{D})$ drawn from an approximation to the posterior (13). The Prediction Queries subsection of Methods discusses how these posterior samples be used to compute point predictions $\hat{y}(\mathbf{s}, t)$ of the spatiotemporal field at a novel index (\mathbf{s}, t) and the associated prediction intervals $[\hat{y}_{\text{low}}(\mathbf{s}, t), \hat{y}_{\text{hi}}(\mathbf{s}, t)]$ for a given level $\alpha \in (0, 1)$ (e.g., $\alpha = 95\%$).

Prediction accuracy on scientific datasets

Datasets. To quantitatively assess the effectiveness of BAYESNF on challenging prediction problems, we curated a benchmark set

comprised of six publicly available, large-scale spatiotemporal datasets that together cover a range of complex empirical processes:

1. Daily wind speed (km/h) from the Irish Meteorological Service³⁸. 1961-01-01 to 1978-12-31; 12 locations; 78,888 observations, 0% missing.
2. Daily particulate matter 10 (PM10, $\mu\text{g}/\text{m}^3$) air quality in Germany from the European Environment Information and Observation Network³⁹. 1998-01-01 to 2009-12-31; 70 locations; 149,151 observations, 52% missing.
3. Hourly particulate matter 10 (PM10, $\mu\text{g}/\text{m}^3$) from the London Air Quality Network²⁰. 2018-12-31 to 2019-03-31; 72 locations; 144,570 observations, 7% missing.
4. Weekly chickenpox counts (thousands) from the Hungarian National Epidemiology Center⁴⁰ 2005-01-03 to 2014-12-29; 20 locations; 10,440 observation, 0% missing.
5. Monthly accumulated precipitation (mm) in Colorado and surrounding areas from the University Corporation for Atmospheric Research⁴¹. 1950-01-01 to 1997-12-01; 358 locations; 134,800 observations, 35% missing.
6. Monthly sea surface temperature ($^{\circ}\text{C}$) anomalies in the Pacific Ocean from the National Oceanic and Atmospheric Administration Climate Prediction Center¹⁷ 1970-01-01 to 2003-03-01; 2261 locations; 902,139 observations, 0% missing.

Table 1 summarizes key statistics of these datasets. Figure 2 shows snapshots of the observed data at a fixed point in time (Fig. 2a) and in space (Fig. 2b), highlighting the complex statistical patterns (e.g., nonstationarity and periodicity) in the underlying fields along these two dimensions. Five train/test splits were created for each benchmark. Each test set contains $(\#locations)/(\#splits)$ locations, holding out the 10% most recent observations.

Baselines. The prediction accuracy on the benchmark datasets in Table 1 using BAYESNF is compared to several state-of-the-art baselines. This evaluation focuses specifically on baseline methods that (i) have high-quality and widely used open-source implementations; (ii) can generate both point and interval predictions; and (iii) are directly applicable to new spatiotemporal datasets (e.g., those in Table 1) without the need to redevelop substantial parts of the model. The methods are:

1. StSVGP: Spatiotemporal Sparse Variational Gaussian Process²⁰. This method handles large datasets (i.e., linear time scaling in the number of time points) by leveraging a state-space representation based on stochastic partial differential equations and Bayesian parallel filtering and smoothing on GPUs. Parameter estimation is performed using natural gradient variational inference.
2. StGBoost: Spatiotemporal Gradient Boosting Trees⁴². Prediction intervals are estimated by minimizing the quantile loss using an ensemble of 1000 tree estimators. As this baseline is not a typical time series model, the same covariates $[x_1(\mathbf{s}, t), \dots, x_m(\mathbf{s}, t)]$ (5)–(9) provided to BAYESNF are also provided as regression inputs.
3. StGLMM: Spatiotemporal Generalized Linear Mixed Effects Models¹⁹. These methods handle large datasets by integrating latent Gaussian-Markov random fields with stochastic partial differential equations. Parameter estimation is performed using maximum marginal likelihood inference. Three observation noise processes are considered:
IID: Independent and identically distributed Gaussian errors.
ARI: Order 1 auto-regressive Gaussian errors.
RW: Gaussian random walk errors.
4. NBEATS: Neural Basis Expansion Analysis⁴³. This baseline employs a “window-based” deep learning auto-regressive model where future data is predicted over a fixed-size horizon conditioned on a

window of previous observations and exogenous features. The model is configured with indicators for all applicable seasonal components—e.g., hour of day, day of week, day of month, week of year, month—as well as trend and seasonal Fourier features. The method contains a large number of numeric hyperparameters which are automatically tuned using the NeuralForecast⁴⁴ package. Prediction intervals are estimated by minimizing quantile loss.

5. TSREG: Trend Surface Regression with Ordinary Least Squares (OLS) (Section 3.2 of ref. 17). The observation noise model is Gaussian with maximum likelihood estimation of the variance. As with StGBoost, the regression covariates are identical to those provided to BAYESNF.
6. BAYESNF: Bayesian Neural Field; using variational and maximum a-posteriori inference.

We also attempted to use the fixed-rank kriging (FRK) method²², but were unable to perform inference over noise parameters for spatiotemporal data. Taken together, the baselines provide broad coverage over recent statistical, machine learning, and deep learning methods for large-scale prediction. All methods were run on a TPU v3-8 accelerator, which consists of 8 cores each with 16 GiB of memory. Additional evaluation details are described in Methods.

Quantitative results. Table 2 shows accuracy and runtime results for all baselines and benchmarks. Point predictions are evaluated using root-mean square error (RMSE (25)) and mean absolute error (MAE (26)) and 95% prediction intervals are evaluated using the mean interval score (MIS (27)), averaged over all train/test splits. The final column shows the wall-clock runtime in seconds that each method was run. While runtime cannot be perfectly aligned due to variety of learning algorithms used and their iterative nature, the wall-clock numbers show that all baselines were run for sufficiently long to ensure a fair comparison. Figure 3 compares predictions on held-out data at one representative spatial location in each of the six benchmarks. We discuss several takeaways from these results.

BAYESNF using VI is the strongest baseline in 12/18 cases followed by BAYESNF using MAP: it is tied with VI in 3/18 cases (Precipitation) and superior in 3/18 cases (Sea Surface Temperature). In 2/18 cases (Chickenpox; MAE and RMSE) errors from the BAYESNF methods are slightly higher than the StGLMM (ARI) baseline, although the running time of the latter is $\sim 4x$ higher. The most apparent improvements of BAYESNF occur in the Wind Speed, Precipitation, and Sea Surface Temperature datasets, shown qualitatively in rows 1, 5, 6 of Fig. 3. Results using additional ablations are discussed in the Ablations subsection of Methods. Combined with Table 2, these results highlight the expressive modeling capacity of BAYESNF models, their ability to accurately quantify predictive uncertainty, and the benefit of using spatial embeddings to capture high-frequency signals in the data.

While predictions from StSVGP generally follow the overall “shape” of the held-out data, the mean and interval predictions are not well calibrated (Fig. 3, second column). StSVGP requires several modeling trade-offs to ensure linear-time scaling in the number of time points, including the use of Matérn kernels (which cannot express effects such as seasonality) and kernels that are separable in time and space. Additional difficulties include manually selecting the number of spatial inducing points and complex algorithms needed to optimize their locations. StSVGP runs out of memory on the Sea Surface Temperature benchmark (1 million observations).

The StGLMM methods (ARI, IID, RW) fail to complete on 4/6 benchmarks. The scaling characteristics are also unpredictable: for example, StGLMM runs on Air Quality 2 (144,570 observations) but fails on Wind Speed (78,888 observations). On the two datasets they can handle (rows 3 and 4 of Fig. 3), the StGLMM methods are highly

competitive on Chickenpox and not competitive on Air Quality 2, with the ARI error model delivering the lowest errors.

StGBoost delivers reasonable prediction intervals but its point predictions underfit (Fig. 3, third column). It has a high computational

cost because (i) a large number of estimators is needed to obtain accurate predictions (using 1000 estimators provided statistically significant improvements over 500 estimators in 17/18 benchmarks); (ii) three models must be separately trained from scratch: one model

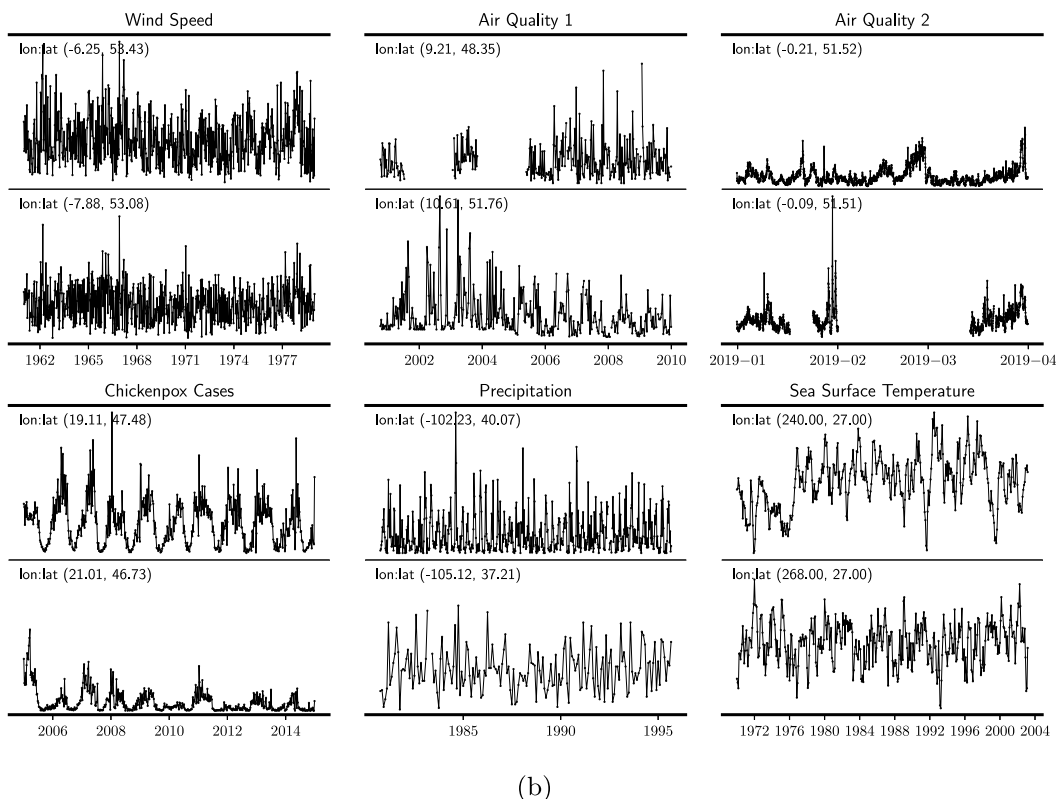
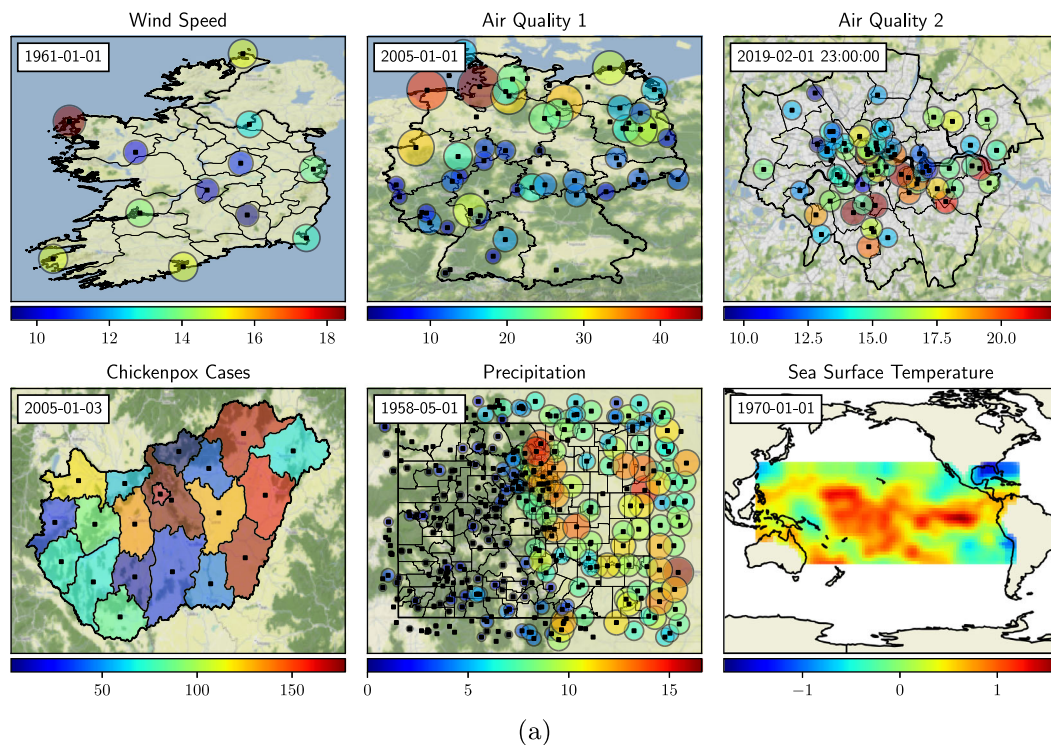


Fig. 2 | Spatial and temporal observations for evaluation datasets from Table 1. **a** Snapshots of spatial observations at fixed points in time. **b** Snapshots of temporal observations at fixed locations in space. Satellite basemap source: © Stadia Maps, ©

OpenMapTiles, © OpenStreetMap, © Stamen Design, © CNES, Distribution Airbus DS, © Airbus DS, © PlanetObserver (Contains Copernicus Data)⁴⁹.

to predict the mean and two models to predict upper and lower quantiles. Whereas BAYESNF uses a single learned distribution for all queries, StGBOST trains different models for different queries, which does not guarantee probabilistically coherent answers.

NBEATS is only competitive on the Sea Surface Temperature benchmark, where it is the next-best baseline after BAYESNF. Its runtime on this benchmark is 3x–4x faster than BAYESNF due to automatic early stopping. The method fails to deliver predictions on the Precipitation

Table 2 | Point prediction errors in terms of root-mean square error (RMSE) and mean average error (MAE); interval prediction error in terms of mean interval score (MIS); and wall-clock runtime in seconds on spatiotemporal benchmark datasets using multiple baselines methods

| Dataset | Method | Prediction Error | | | |
|--------------------------|---|------------------|--------------|---------------|---------|
| | | RMSE | MAE | MIS | Runtime |
| Wind Speed | Bayesian Neural Field (VI) | 2.44 | 1.81 | 11.88 | 1167 |
| | Bayesian Neural Field (MAP) | 2.61 | 1.93 | 12.65 | 927 |
| | Sparse Spatiotemporal Variational Gaussian Process | 5.04 | 4.18 | 24.72 | 1112 |
| | Spatiotemporal Gradient Boosting Trees | 3.74 | 2.79 | 18.43 | 2907 |
| | Neural Basis Expansion Analysis | 5.20 | 4.07 | 22.92 | 9237 |
| | Spatiotemporal Generalized Linear Mixed Model (All) | x | x | x | x |
| | Trend Surface Regression | 4.94 | 3.88 | 24.83 | ≤1 |
| Air Quality 1 | Bayesian Neural Field (VI) | 5.02 | 2.94 | 22.52 | 1169 |
| | Bayesian Neural Field (MAP) | 5.33 | 3.15 | 24.84 | 1284 |
| | Sparse Spatiotemporal Variational Gaussian Process | 6.24 | 3.91 | 35.59 | 1348 |
| | Spatiotemporal Gradient Boosting Trees | 7.42 | 4.40 | 31.56 | 5665 |
| | Neural Basis Expansion Analysis | 9.23 | 5.95 | 45.11 | 1461 |
| | Spatiotemporal Generalized Linear Mixed Model (All) | x | x | x | x |
| | Trend Surface Regression | 9.35 | 6.62 | 55.98 | ≤1 |
| Air Quality 2 | Bayesian Neural Field (VI) | 8.39 | 5.19 | 40.08 | 618 |
| | Bayesian Neural Field (MAP) | 8.82 | 5.42 | 43.24 | 678 |
| | Sparse Spatiotemporal Variational Gaussian Process | 9.92 | 6.78 | 56.12 | 628 |
| | Spatiotemporal Gradient Boosting Trees | 8.77 | 5.57 | 43.71 | 2671 |
| | Neural Basis Expansion Analysis | 12.63 | 8.24 | 63.84 | 778 |
| | Spatiotemporal Generalized Linear Mixed Model (AR1) | 11.92 | 7.81 | 73.00 | 17100 |
| | Spatiotemporal Generalized Linear Mixed Model (RW) | 14.62 | 9.48 | 157.10 | 9447 |
| | Spatiotemporal Generalized Linear Mixed Model (IID) | 12.87 | 8.78 | 127.48 | 3545 |
| | Trend Surface Regression | 18.44 | 12.32 | 117.90 | ≤1 |
| Chickenpox Cases | Bayesian Neural Field (VI) | 25.96 | 16.09 | 137.74 | 141 |
| | Bayesian Neural Field (MAP) | 26.54 | 17.63 | 114.44 | 70 |
| | Sparse Spatiotemporal Variational Gaussian Process | 32.00 | 21.22 | 212.87 | 63 |
| | Spatiotemporal Gradient Boosting Trees | 26.83 | 15.84 | 122.39 | 189 |
| | Neural Basis Expansion Analysis | 29.51 | 17.56 | 167.27 | 250 |
| | Spatiotemporal Generalized Linear Mixed Model (AR1) | 25.30 | 15.26 | 179.29 | 887 |
| | Spatiotemporal Generalized Linear Mixed Model (RW) | 26.92 | 16.79 | 179.63 | 386 |
| | Spatiotemporal Generalized Linear Mixed Model (IID) | 28.23 | 16.85 | 327.72 | 264 |
| | Trend Surface Regression | 29.75 | 21.30 | 172.43 | ≤1 |
| Precipitation | Bayesian Neural Field (VI) | 1.80 | 1.23 | 8.33 | 778 |
| | Bayesian Neural Field (MAP) | 1.83 | 1.21 | 8.28 | 1069 |
| | Sparse Spatiotemporal Variational Gaussian Process | 3.14 | 2.27 | 31.00 | 1203 |
| | Spatiotemporal Gradient Boosting Trees | 2.63 | 1.67 | 11.13 | 2064 |
| | Neural Basis Expansion Analysis | x | x | x | x |
| | Spatiotemporal Generalized Linear Mixed Model (All) | x | x | x | x |
| | Trend Surface Regression | 3.61 | 2.69 | 20.81 | ≤1 |
| Sea Surface Temperature | Bayesian Neural Field (VI) | 0.14 | 0.09 | 0.77 | 3335 |
| | Bayesian Neural Field (MAP) | 0.10 | 0.06 | 0.63 | 4624 |
| | Sparse Spatiotemporal Variational Gaussian Process | x | x | x | x |
| | Spatiotemporal Gradient Boosting Trees | 0.45 | 0.33 | 1.94 | 12379 |
| | Neural Basis Expansion Analysis | 0.20 | 0.15 | 0.97 | 1120 |
| | Spatiotemporal Generalized Linear Mixed Model (All) | x | x | x | x |
| Trend Surface Regression | 0.55 | 0.42 | 2.89 | 3 | |

Each error measurement, shown to two significant figures, is an average over five independent test/train splits. The symbol x denotes an experiment that failed to complete successfully (timeout, out-of-memory, too sparse, etc.). Bold values indicate statistically significant lowest errors (Mann–Whitney U-Test at the 5% level with Bonferroni correction).

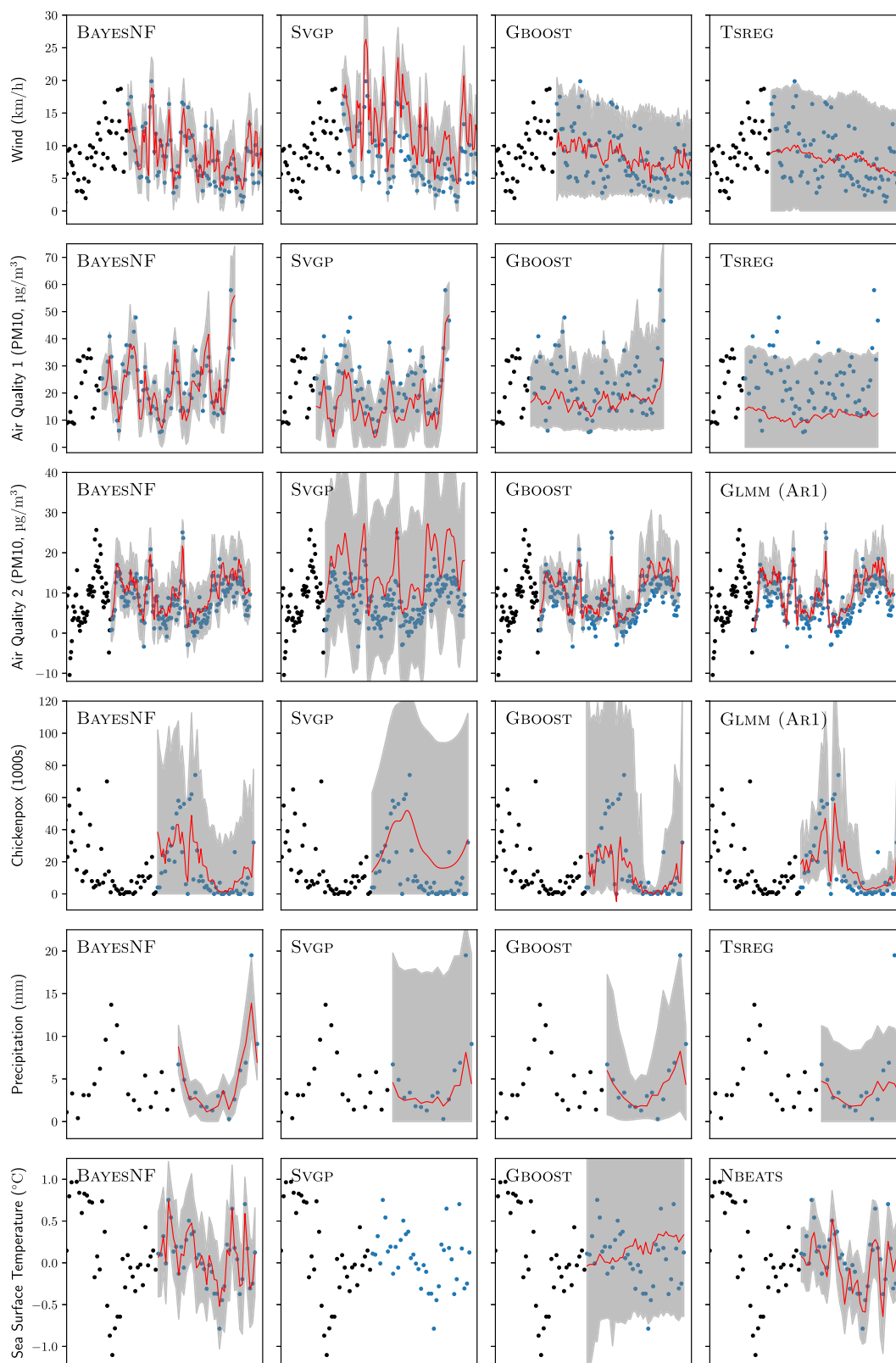


Fig. 3 | Comparison of predictions using BAYESNF and various baselines. Each row shows results for a given spatiotemporal benchmark dataset at one spatial location. Black dots are observed data, blue dots are test data, red lines are median forecasts, and gray regions are 95% prediction intervals. (BAYESNF: Bayesian Neural

Field. SVGP: Spatiotemporal Sparse Variational Gaussian Process. GBOOST: Spatiotemporal Gradient Boosting Trees. SrGLMM: Spatiotemporal Generalized Linear Mixed Effect Models. NBEATS: Neural Basis Expansion Analysis. TSREG: Trend-Surface Regression).

benchmark because the training and test datasets contain time series that are too sparse to handle; e.g., the number of observed timepoints is smaller than the auto-regressive window size or prediction horizon. The prediction errors on the remaining three benchmarks are high even though all the seasonal effects were added to the model, suggesting that either (i) the model is not able to effectively leverage spatial correlations for cross time-series learning; or (ii) the hyperparameter tuning algorithm does not converge to sensible values within the allotted time.

TSREG requires less than 1 second to train, but does not capture any meaningful structure and produces poor predictions. Using LASSO or ridge regression instead of OLS did not improve the results. TSREG uses identical covariates to BAYESNF but performs much worse, highlighting the need to capture nonlinear dependencies in the data for generating accurate forecasts.

Analyzing German air quality data

Atmospheric particulate matter (PM10) is a key indicator of air quality used by governments worldwide, as these particles can induce adverse health effects when inhaled into the lungs. Accurate predictions of PM10 values at novel points in space and time within a geographic region can help decision makers characterize pollution patterns and inform public health decisions.

We explore predictions from BAYESNF on the German Air Quality dataset³⁹, which contains daily PM10 measurements from 70 stations between 1998-01-01 and 2009-12-31. We infer a BAYESNF model for this dataset with depth $H=2$; weekly, monthly, and yearly seasonal effects (8); and harmonics $\mathcal{H}_1^S = \mathcal{H}_2^S = \{1, \dots, 4\}$ for the spatial Fourier features (9). The distribution of Y given the stochastic process F is a StudentT (11) truncated to $\mathbb{R}_{\geq 0}$

Spatial and temporal interpolation. Figure 4a shows the PM10 observations at 2003-02-01, 2005-01-01, 2005-04-01, and 2007-01-01, where roughly 50% of the stations do not have an observed measurement at a given point in time. Figure 4b shows the median PM10 predictions $y_{0.5}(\mathbf{s}, t)$ (24) interpolated at a grid of 10,000 novel spatial indexes (\mathbf{s}, t) within Germany. Figure 4c shows the width $\hat{y}_{hi}(\mathbf{s}, t) - \hat{y}_{low}(\mathbf{s}, t)$ of the inferred 95% prediction interval. These plots reflect the spatiotemporal structure captured by BAYESNF and identify coordinates within the field with low and high predictive uncertainty about air pollution. The axis-aligned artifacts in Fig. 4b, where predictions are consistent along certain thin regions, are a result of the spatial Fourier features (9). How well these artifacts reflect the true behavior can be empirically investigated by obtaining PM10 measurements at the novel locations along these regions. Figure 4d shows the observed and median predicted PM10 values across all time points at four stations with the highest missing data rates: DEBWO31, southwest Germany, 51% missing; DEBB056, north-east Germany, 84% missing; DEBU034, northwest Germany, 99% missing; DESLO08, west Germany, 89% missing. PM10 trajectories predicted by BAYESNF at time points where data is missing reproduce the temporal patterns at time points with observed data, which include high frequency periodic variation and irregular, spatially correlated jumps.

Variography. The accuracy of PM10 predictions in Fig. 4d cannot be quantitatively assessed because the ground-truth values are not known at the predicted time points. However, we can gain more insight into how well the learned spatiotemporal field matches the observed field by comparing the empirical and inferred semi-variograms. The semi-variogram γ of a process Y characterizes the joint spatiotemporal dependence structure; it is defined as

$$2\gamma(\mathbf{h}, \tau) = \text{Var}[Y(\mathbf{s} + \mathbf{h}, t + \tau) - Y(\mathbf{s}, t)] \quad (\mathbf{h} \in \mathcal{S}, \tau \in \mathcal{T}), \quad (14)$$

where the choice of $\mathbf{s} \in \mathcal{S}, \tau \in \mathcal{T}$ is arbitrary (e.g., $(\mathbf{s}, \tau) = (\mathbf{0}, 0)$), under the assumption that only the displacements in time and space affect the dependence (Section 2.4.2 of ref. 17).

The surface plots in Fig. 5 compare the empirical semi-variogram (left) computed at the 70 observed stations with the inferred semi-variogram (right) computed at 70 uniformly chosen random locations within Germany, for distances $\mathbf{h} \in [0, 1000]$ kilometers and time lags $\tau \in \{0, \dots, 10\}$ days. The agreement between these two plots suggests that BAYESNF accurately generalizes the spatiotemporal dependence structure from the observed locations to novel locations in the field. The lower two panels in Fig. 5 show the empirical (solid line) and inferred (dashed line) semi-variograms, separately for each of the 10 time lags τ . The difference between the semi-variograms is highest for $\tau \in \{0, 1, 2\}$ days, suggesting that the learned model is expressing relatively smooth phenomena and assuming that the high-frequency day-to-day variance is due to unpredictable independent noise. The differences between the semi-variograms become small for $\tau > 2$ days, which suggests that BAYESNF effectively captures these longer-term temporal dependencies.

Discussion

This article proposes a probabilistic approach to scalable spatiotemporal prediction called the Bayesian Neural Field. The model combines a deep neural network architecture for high-capacity function approximation with hierarchical Bayesian modeling for accurate uncertainty estimation over complex spatiotemporal fields. Posterior inference is conducted using stochastic ensembles of maximum a-posteriori estimation or variationally trained surrogates, which are easy to apply and deliver well-calibrated 95% prediction intervals over test data. The results in Fig. 6 confirm that quantifying uncertainty using MAP or VI ensembles is superior to performing maximum-likelihood estimation (MLE), which ignores the parameter priors. While these inference methods are approximate in nature and are not guaranteed to match the true posterior, the BAYESNF model is a deep neural network where interpreting parameters such as weights and biases is not of inherent interest to a practitioner in a given data analysis task. Rather, we expect BAYESNF to be most useful in cases where the predictive calibration is more relevant. Additional advantages of BAYESNF are its relative simplicity, ability to handle missing data, and ability to learn a full probability distribution over arbitrary space-time indexes within the spatiotemporal field.

Evaluations against prominent statistical and machine learning baselines on large-scale datasets show that BAYESNF delivers significant improvements in both point and interval forecasts. The results also show that combining periodic effects in the temporal domain with Fourier features in the spatial domain enables BAYESNF to capture spatiotemporal patterns with multiple (non-integer) periodicity and high-frequency components. As a domain-general method, BAYESNF can produce strong results on multiple datasets without the need to hand-design the model from scratch each time or apply dataset-specific inference approximations. For a representative air quality dataset, the semi-variograms inferred by BAYESNF evaluated at novel spatial locations agree with the empirical semi-variogram computed at observed locations, which highlights the model's ability to generalize well in space and time.

Practitioners across a spectrum of disciplines—from meteorology to urban studies and environmental informatics—are in need of more scalable and easy-to-use statistical methods for spatiotemporal prediction. A freely available implementation of BAYESNF built on the JAX machine learning platform, along with user documentation and tutorials, is available at <https://github.com/google/bayesnf>. We hope these materials help practitioners obtain strong BAYESNF

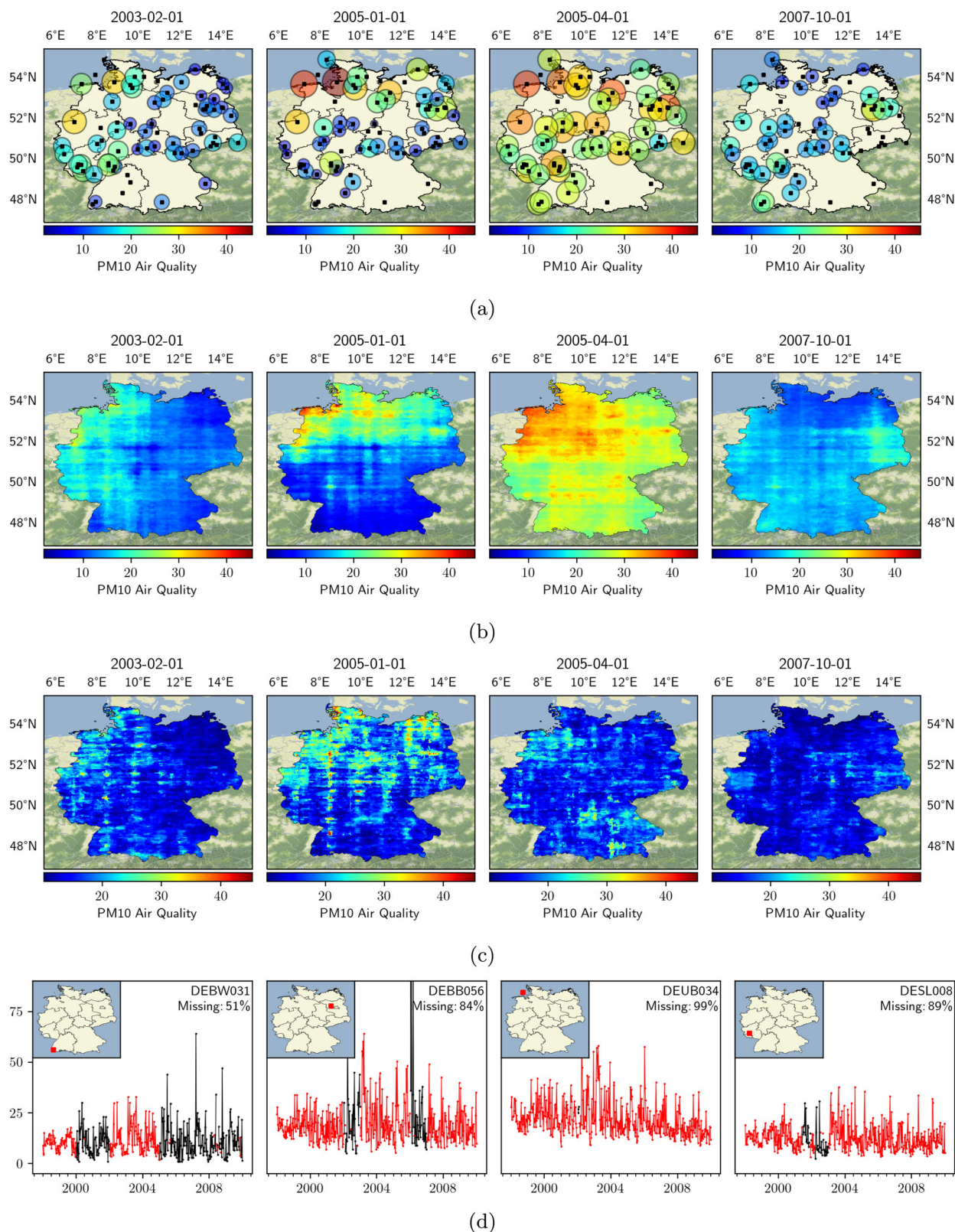


Fig. 4 | Spatiotemporal prediction of atmospheric particulate matter (PM10) in German air dataset. a shows the observed data at four time points: each shaded circle represents a measurements of PM10 at a given station. Higher values of PM10 correspond to lower air quality. The data is sparse: at any given time point, only 47% of stations (on average) are associated with a PM10 observation. **b** Median predictions of PM10 air quality at four time points across the whole spatial field. **c** Width of 95% predictions of PM10 air quality at four time points across the whole spatial field. **d** Observed PM10 data (black) and median prediction (red) at four sparsely observed locations across time. Satellite basemap source: © Stadia Maps, © OpenMapTiles, © OpenStreetMap, © Stamen Design, © CNES, Distribution Airbus DS, © Airbus DS, © PlanetObserver (Contains Copernicus Data)⁴⁹.

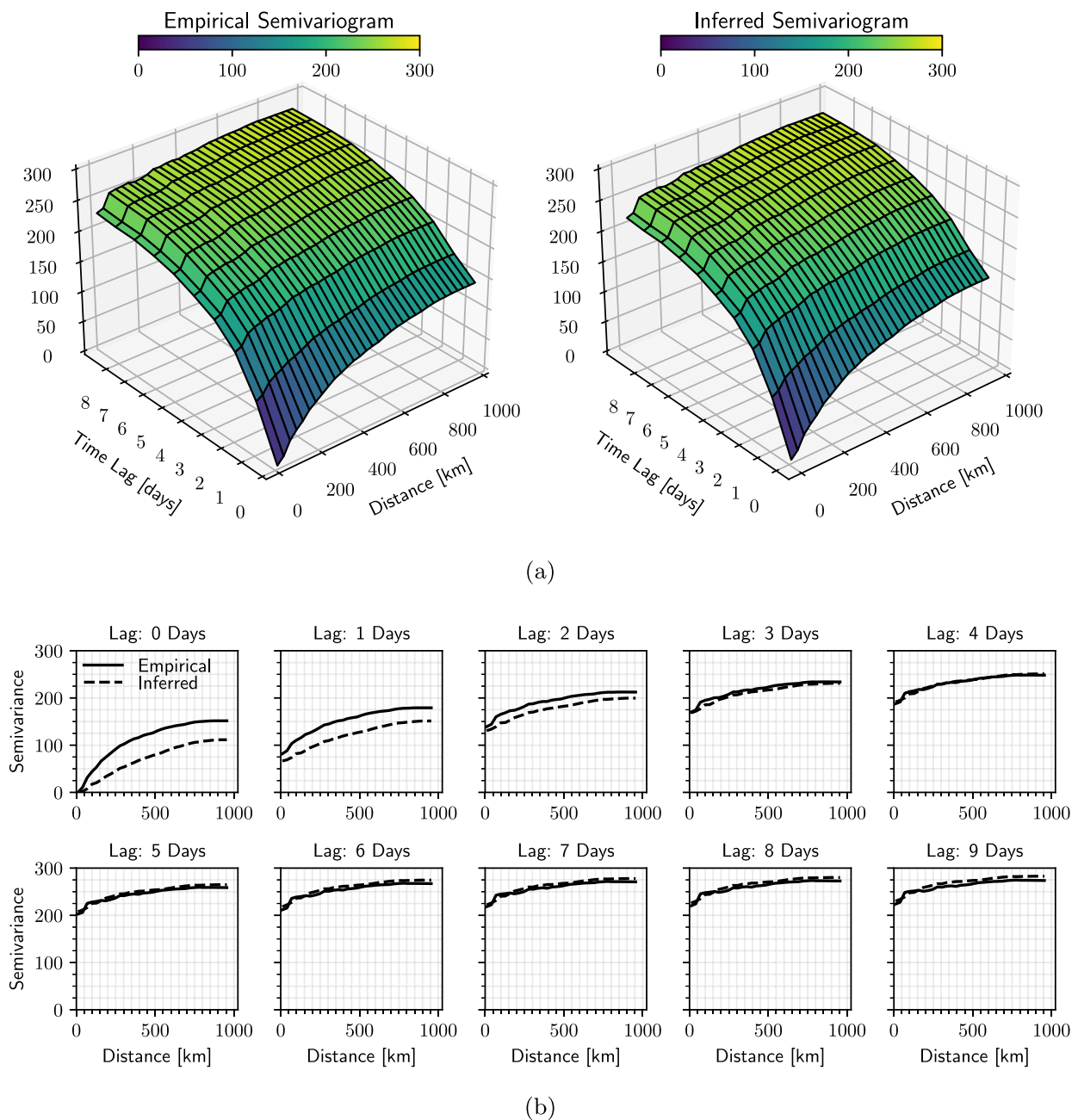


Fig. 5 | Comparison of the empirical and inferred spatiotemporal semivariograms, which measure the variance of the difference between field values at a pair of locations, for German PM10 air quality dataset. The empirical semivariogram is computed using the locations of the 70 stations in the observed dataset. The inferred semivariogram is computed on 70 novel spatial locations, sampled uniformly at random within the boundary of the field. **a** The agreement between the

semivariogram surfaces indicates that BAYESNF extrapolates the joint spatiotemporal dependence structure between locations in the observed data to novel locations. **b** For short time lags less than three days, the empirical variogram is higher than the inferred variogram at all distances, showing that BAYESNF models high-frequency day-to-day variance as unpredictable observation noise.

models for many spatiotemporal problems that existing software cannot easily handle.

The approach discussed in this paper opens several avenues to future work. While Bayesian Neural Fields are designed to minimize the user's involvement in constructing a predictive model, further improvements can be achieved by enabling domain experts to incorporate specific statistical covariance structure that they know to be present. It is also worthwhile to explore applications of BAYESNF for modeling the residuals of causal or mechanistic laws in physical systems where there exist strong domain theories of the average data-generating process, but poor models of the empirical noise

process. Another promising extension is using BAYESNF models to handle not only “geostatistical” datasets, in which the measurements are point-referenced in space, but also “areal” or “lattice” datasets, where the measurements represent aggregated quantities over a geographical region. While areal datasets are often converted to geostatistical datasets by using the centroid of the region as the representative point, a more principled approach would be to compute the integral of a Bayesian Neural Field over the region. Finally, BAYESNF can be generalized to handle multivariate spatiotemporal data, where each spatial location is associated with multiple time series that contain within-location and

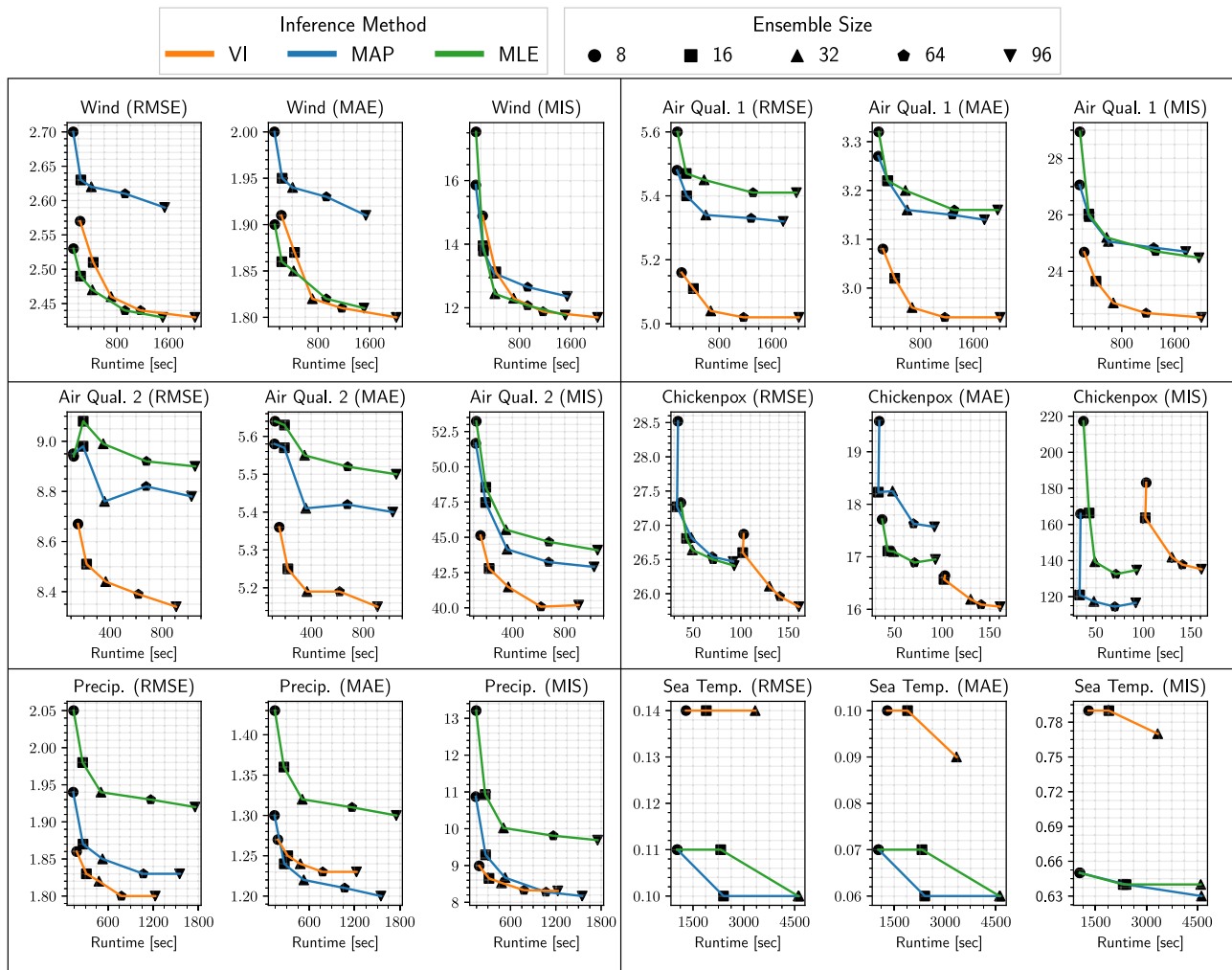


Fig. 6 | Runtime versus prediction error profiles using variational inference (VI; orange), maximum a-posteriori (MAP; blue), and maximum likelihood estimation (MLE; green) for BAYESNF on the spatiotemporal benchmarks from

Table 1. Markers indicate ensemble size (8, 16, 32, 64, 96). The predictions errors are given in terms of root-mean square error (RMSE) and mean average error (MAE) for point forecasts and in terms of mean interval score (MIS) for 95% interval forecasts.

across-location covariance structure. Effectively handling such datasets will even further broaden the scope of problems that BAYESNF can solve.

Methods

Posterior inference

Let $P(\theta_f, \theta_y, Y)$ denote the joint probability distribution over the parameters and observable field in Box 1. The posterior distribution is given by Eq. (13) in the main text. We describe two approximate posterior inference algorithms for BAYESNF. In these sections, we define $\theta = (\theta_f, \theta_y)$, $\theta = (\theta_f, \theta_y)$ and $\mathbf{r} = (\mathbf{s}, t)$.

Stochastic MAP ensembles. A simple approach to uncertainty quantification is based on the “maximum a-posteriori” estimate:

$$\theta^* = \arg \max_{\theta} \left\{ \log P(\theta_f, \theta_y, \{Y(\mathbf{r}_i) = y(\mathbf{r}_i)\}_{i=1}^N) \right\}. \quad (15)$$

We find an approximate solution to the optimization problem (15) using stochastic gradient ascent on the joint log probability, according to the following procedure, where $B \leq N$ is a mini-batch size and $(\epsilon_1, \epsilon_2, \dots)$ is a sequence of learning rates:

$$\text{Initialize } \theta_0 \sim \pi_f \pi_y; t \leftarrow 0 \quad (16)$$

Repeat until convergence

$$\{I_1, \dots, I_B\} \sim \text{Uniform}(\{K \subset [N] \mid \text{card}(K) = B\}) \quad (17)$$

$$\hat{g}_t = \nabla_{\theta} \left[\log \pi_f(\theta_f) + \log \pi_y(\theta_y) + \frac{N}{B} \sum_{j=1}^B \log \left(\text{Dist}(y(\mathbf{r}_j); F_{\theta_f}(\mathbf{r}_j), \theta_y) \right) \right]_{\theta_{t-1}} \quad (18)$$

$$\theta_t = \theta_{t-1} + \epsilon_t \hat{g}_t; t \leftarrow t+1. \quad (19)$$

We construct an overall “deep ensemble” $\{(\theta_f^i, \theta_y^i)\}_{i=1}^M$ containing $M \geq 1$ MAP estimates by repeating the above procedure M times, each with a different initialization of θ_0 and random seed.

Stochastic variational inference. A more uncertainty-aware alternative to MAP ensembles is mean-field variational inference, which uses a surrogate posterior $q_{\phi}(\theta) = \prod_{i=1}^{n_f} \nu(\theta_{f,i}; \phi_{f,i}) \prod_{i=1}^{n_y} \nu(\theta_{y,i}; \phi_{y,i})$ over Θ to approximate the true posterior $P(\theta_f, \theta_y | \mathcal{D})$ (13) given the data \mathcal{D} . Optimal values for the variational parameters $\phi = (\phi_{f,1}, \dots, \phi_{f,n_f}, \phi_{y,1}, \dots, \phi_{y,n_y})$

are obtained by maximizing the “evidence lower bound”:

$$\text{ELBO}(\phi) = \log P(\mathcal{D}) - \text{KL}(q_\phi(\theta) \| P(\theta | \mathcal{D})) = \mathbb{E}_\phi \left[\log \frac{P(\mathcal{D}, \theta)}{q_\phi(\theta)} \right] \quad (20)$$

$$= \mathbb{E}_\phi [\log P(\mathcal{D} | \theta)] - \text{KL}(q_\phi(\theta) \| \pi(\theta)). \quad (21)$$

$$= \sum_{i=1}^N \mathbb{E}_\phi \left[\log \left(\text{Dist}(y(\mathbf{r}_i); F_{\theta_f}(\mathbf{r}_i, \theta_y)) \right) \right] - \sum_{i=1}^{n_f} \mathbb{E}_{\phi_{f,i}} \left[\log \left(\frac{\nu(\theta_{f,i}; \Phi_{f,i})}{\pi_f(\theta_{f,i})} \right) \right] - \sum_{i=1}^{n_y} \mathbb{E}_{\phi_{y,i}} \left[\log \left(\frac{\nu(\theta_{y,i}; \Phi_{y,i})}{\pi_y(\theta_{y,i})} \right) \right]. \quad (22)$$

where Eq. (22) follows from the independence of the priors. Finding the maximum of Eq. (22) is a challenging optimization problem. Our implementation leverages a Gaussian variational posterior q_ϕ with KL reweighting, as described in Blundell et al. (Sections 3.2 and 3.4 of ref. 45).

Mean-field variational inference is known to underestimate posterior variance and can also get stuck in local optima of Eq. (21). To alleviate these problems, we use a *variational ensemble* that is analogous to the MAP ensemble described above. More specifically, we first perform $M \geq 1$ runs of stochastic variational inference with different initializations and random seeds, which gives us an ensemble $\{\phi^i, i = 1, \dots, M\}$ of variational parameters. We then approximate the posterior $P(\theta | \mathcal{D})$ with an equal-weighted mixture of the resulting variational distributions $\{q_{\phi^i}\}_{i=1}^M$.

Prediction queries

We can approximate the posterior (13) using a set of samples $\{(\theta_{f,i}^j, \theta_{y,i}^j)\}_{i=1}^M$, which may be obtained from either MAP ensemble estimation or stochastic variational inference (by sampling from the ensemble of M variational distributions). We can then approximate the posterior-predictive distribution $P(Y(\mathbf{r}_*) | \mathcal{D})$ (which marginalizes out the parameters θ) of $Y(\mathbf{r}_*)$ at a novel field index $\mathbf{r}_* = (\mathbf{s}_*, t_*)$ by a mixture model with M equally weighted components:

$$\hat{P}(Y(\mathbf{r}_*) | \mathcal{D}) = \frac{1}{M} \sum_{i=1}^M \text{Dist}(Y(\mathbf{r}_*); F_{\theta_f^i}(\mathbf{r}_*, \theta_{y,i}^j)). \quad (23)$$

Equipped with Eq. (23), we can directly compute predictive probabilities of events $\{Y(\mathbf{r}_*) \leq y\}$, predictive probability densities $\{Y(\mathbf{r}_*) = y\}$, or conditional expectations $\mathbb{E}[\varphi(Y(\mathbf{r}_*) | \mathcal{D})]$ for a probe function $\varphi: \mathbb{R} \rightarrow \mathbb{R}$. Prediction intervals around $Y(\mathbf{r}_*)$ are estimated by computing the α -quantile $y_\alpha(\mathbf{r}_*)$, which satisfies

$$\hat{P}(Y(\mathbf{r}_*) \leq y_\alpha(\mathbf{r}_*) | \mathcal{D}) = \alpha \quad \alpha \in [0, 1]. \quad (24)$$

For example, the median estimate is $y_{0.50}(\mathbf{s}_*, t_*)$ and 95% prediction interval is $[y_{0.025}(\mathbf{s}_*, t_*), y_{0.975}(\mathbf{s}_*, t_*)]$. The quantiles (24) are estimated numerically using Chandrupatla’s root finding algorithm⁴⁶ on the cumulative distribution function of the mixture (23).

Temporal seasonal features

Including seasonal features (c.f. Eq. (8)), where possible, is often essential for accurate prediction. Example periodic multiples p for datasets with a variety of time units and seasonal components are listed below (Y=Yearly; Q=Quarterly; Mo=Monthly; W=Weekly; D=Daily; H=Hourly; Mi=Minutely; S=Secondly):

Q: Y=4

Mo: Q=3, Y=12

W: Mo=4.35, Q=13.045, Y=52.18

D: W=7, Mo=30.44, Q=91.32, Y=365.25

H: D=24, W=168, Mo=730.5, Q=2191.5, Y=8766

Mi: H=60, D=1440, W=10080, Mo=43830, Q=131490, Y=525960

S: Mi=60, H=3600, D=86400, W=604800, Mo=2629800, Q=7889400, Y=31557600

Ablations

To better understand how the prediction accuracy of BAYESNF varies with the choices of inference algorithm and network architecture, results from two classes of ablation studies for the benchmarks in Table 2 are reported.

Inference methods: comparison of VI, MAP, and MLE. Figure 6 shows a comparison of runtime vs. accuracy profiles on the six benchmarks from Table 1 using three parameter inference methods for BAYESNF—VI, MAP, and MLE. MLE is the maximum likelihood estimation baseline described in Lakshminarayanan et al.⁴⁷, which is identical to Box 1 expect that the terms π_f and π_y in Eq. (18) are ignored. MLE performs no better than MAP or VI in all 18/18 profiles (and is typically worse), illustrating the benefits of parameter priors and posterior uncertainty which do not impose runtime overhead. Between MAP and VI, the latter performs better in 13/18 profiles: that is, on all metrics for Wind, Air Quality 1, and Air Quality 2; on RMSE and MAE for Chickenpox; and on RMSE and MIS for Precipitation.

Model architectures. Figure 7 shows the percentage change in RMSE, MAE, MIS, and runtime using BAYESNF (MAP inference; 64 particles; fixed number of training epochs) while applying a single change to the reference model for each benchmark. The goal of these ablations is to study how changes to the network structure affect the predictive performance.

Figure 7a, b shows results for decreasing or increasing the network depth by one layer. The Sea Surface Temperature benchmark is the most sensitive to the network depth, where decreasing the depth causes the forecast errors to increase by around 50%, whereas increasing the depth delivers 5–10% decreases. The MIS error is particularly sensitive to reducing the network depth where the results become significantly worse in 5/6 benchmarks, although the runtime also decreases by up to 50%.

Figure 7c, d shows results for halving or doubling the width of the hidden layers. The Sea Surface Temperature benchmark is highly sensitive to halving the network width, with errors increasing above 25%. The remaining benchmarks demonstrate slight improvements in the errors which are not statistically significant, suggesting that the runtime gains could justify halving the width in these benchmarks. Doubling the width causes substantial increases in the runtime with no systematic pattern in the RMSE, MAE, or MIS values across the benchmarks.

Figure 7e, f shows results using only tanh or elu activations instead of the convex combination layer. Discarding the convex combination layer delivers runtime speedups, which are larger using tanh as compared to elu. However, there is no clear winner in terms of error when using only tanh or only elu; and no error metric is consistently negative by selecting one of the two activations. The changes in error which are consistently positive (as compared to the convex combination layer) are (i) tanh only: Air Quality 2 (MIS 16%); (ii) elu only: Sea Surface Temperature (RMSE 59%, MAE 76%, MIS 49%) and Precipitation (MAE 7.8%, MIS 16%).

Figure 7g shows results for disabling the covariate scaling layer. The runtime is only slightly changed in all benchmarks. However, several errors increase consistently on average, namely in the Precipitation (RMSE 24%, MAE 27%, MIS 33%), Chickenpox (MIS 32%), and Air Quality 1 (MAE 13%) benchmarks. The remaining changes are neither consistently above nor below zero.

Figure 7h shows results for omitting the spatial Fourier features (Eq. (9)). While omitting these features delivers small runtime improvements, it also causes substantial increases in RMSE, MAE, and MIS values across all benchmarks except for Wind. These results support the hypothesis that spatial Fourier features are essential for accurate generalization across space and time.

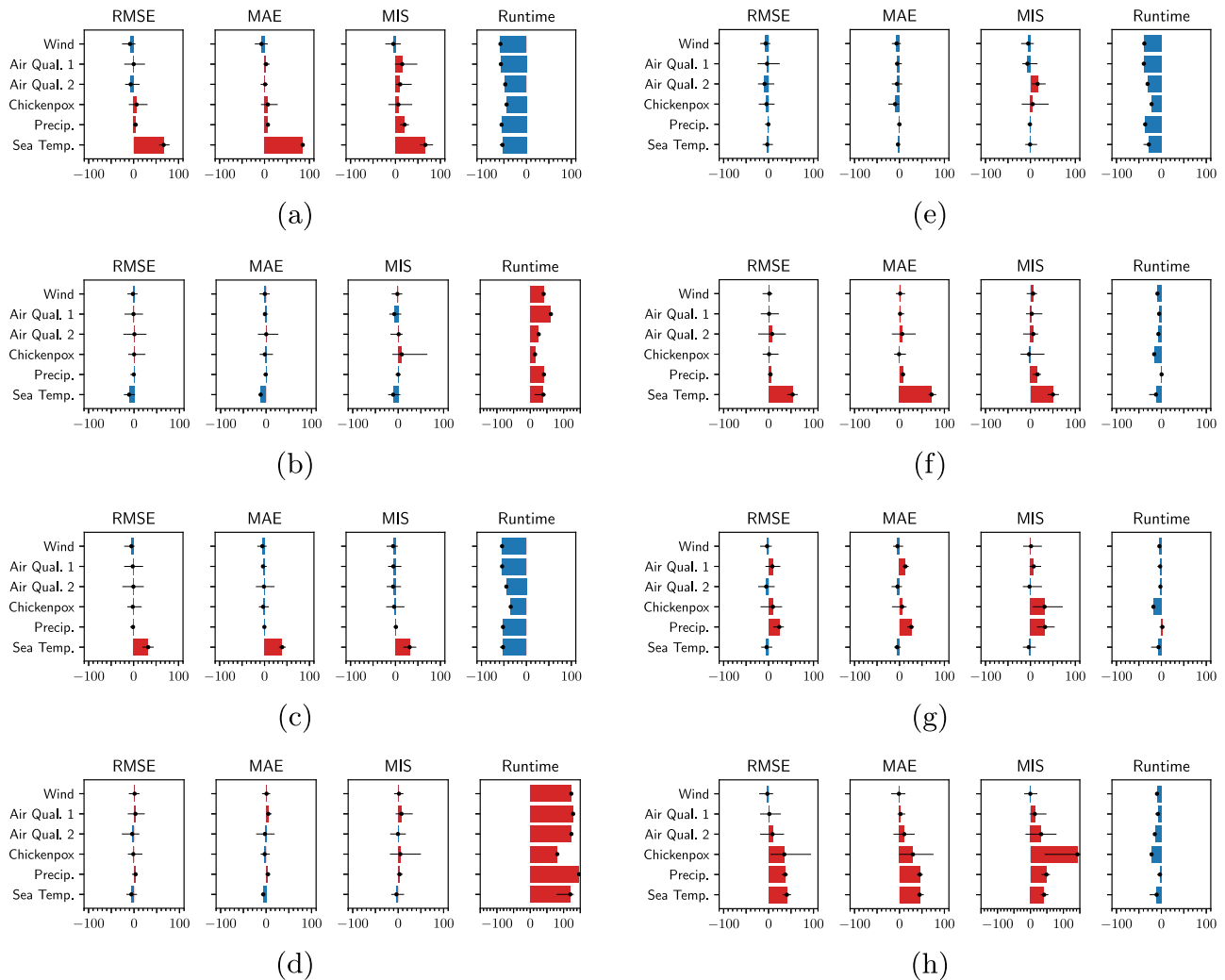


Fig. 7 | Percentage change in error (RMSE, MAE, MIS) and runtime on benchmark datasets using BAYesNF with various modeling ablations. Horizontal bars in red (resp. blue) show an increase (resp. decrease) in the error and runtime measurements. Errors bars show the minimum and maximum across all train/test

splits. **a** Decrease network depth by 1. **b** Increase network depth by 1. **c** Half network width. **d** Double network width. **e** No convex combination (tanh activation). **f** No convex combination (elu activation). **g** No covariate scaling layer. **h** No spatial Fourier features.

In summary, the results (specifically Fig. 7e–h), demonstrate that architectural choices in BAYesNF such as the spatial Fourier features, convex combination layer, and covariate scaling are effective in reducing the prediction error across several benchmarks and metrics at the cost of a manageable runtime overhead.

Evaluation metrics

The quality of point forecasts are evaluated using RMSE and MAE scores. Interval forecasts are evaluated using the MIS score at level $\alpha = 0.05$. The definitions are as follows:

$$\text{Root Mean Squared Error (RMSE)} \quad \sqrt{\sum_{i=1}^n (y_i - \hat{y}_i)^2 / n} \quad (25)$$

$$\text{Mean Absolute Error (MAE)} \quad \sum_{i=1}^n |y_i - \hat{y}_i| / n \quad (26)$$

$$\text{Mean Interval Score (MIS)} \quad \sum_{i=1}^n \left[(u_i - \ell_i) + \frac{2}{\alpha} (\ell_i - y_i) \mathbf{1}_{\{y_i < \ell_i\}} + \frac{2}{\alpha} (y_i - u_i) \mathbf{1}_{\{u_i < y_i\}} \right] / n, \quad (27)$$

where y_i is the true value, \hat{y}_i is the point forecast, and (ℓ_i, u_i) are end-points of the interval forecast.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All datasets from Table 1 are publicly available under open-source licenses. • Wind Speed. GNU GPL v2. <https://r-spatial.github.io/gstat/reference/wind.html>. • Air Quality 1. GNU GPL v3. <https://rdrr.io/cran/spacetime/man/air.html>. • Air Quality 2. CC Attribution 1.0 Generic. <https://doi.org/10.5281/zenodo.4531304>. • Chickenpox Cases. CC Attribution 4.0 International. <https://doi.org/10.24432/C5103B>. • Precipitation. Public Domain. <https://www.image.ucar.edu/Data/US.monthly.met/>. • Sea Surface Temperature. GNU GPL v2. <https://github.com/andrewzm/STRbook/>. The full datasets, test/train splits, model predictions, and ablation results are available at <https://doi.org/10.5281/zenodo.12735404>. Refer to the README in these files for additional information.

Code availability

An open-source Python implementation of BAYESNF is available at <https://github.com/google/bayesnf> under an Apache-2.0 License. The full evaluation pipeline containing all model configurations for the baselines is also provided in the repository. The source code of bayesnf v0.1.3 is uploaded in the Supplementary Code.

References

- European Environment Agency. European air quality index. <https://airindex.eea.europa.eu/AQI/index.html> (2024).
- U.S. Environmental Protection Agency. U.S. air quality index. <https://www.airnow.gov/> (2024).
- Wang, S., Yuan, W. & Shang, K. The impacts of different kinds of dust events on PM10 pollution in northern China. *Atmos. Environ.* **40**, 7975–7982 (2006).
- Medina, S., Plasencia, A., Ballester, F., Mücke, H. G. & Schwartz, J. Apehis: public health impact of PM10 in 19 European cities. *J. Epidemiol. Community Health* **58**, 831–836 (2004).
- Huang, W. et al. An overview of air quality analysis by big data techniques: Monitoring, forecasting, and traceability. *Inf. Fusion* **75**, 28–40 (2021).
- Karagulian, F. et al. Review of the performance of low-cost sensors for air quality monitoring. *Atmosphere* **10**, 506 (2019).
- Niu, D., Feng, C. & Li, B. Pricing cloud bandwidth reservations under demand uncertainty. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, 151–162 (Association for Computing Machinery, 2012).
- Mishra, S. K., Sahoo, B. & Paramita Parida, P. Load balancing in cloud computing: a big picture. *J. King Saud. Univ. Comput. Inf. Sci.* **32**, 149–158 (2020).
- Cao, J., Wu, Y. & Li, M. Energy efficient allocation of virtual machines in cloud computing environments based on demand forecast. In *Advances in Grid and Pervasive Computing, of Lecture Notes in Computer Science* **7296**, 137–151 (Springer, 2012).
- Faniyi, F. & Bahsoon, R. A systematic review of service level management in the cloud. *ACM Comput. Surveys* **48**, 1–27 (2015).
- Sigrist, F., Künsch, H. R. & Stahel, W. A. A dynamic nonstationary spatio-temporal model for short term prediction of precipitation. *Ann. Appl. Stat.* **6**, 1452–1477 (2012).
- Jung, J. & Broadwater, R. P. Current status and future advances for wind speed and power forecasting. *Renew. Sustain. Energy Rev.* **31**, 762–777 (2014).
- Lu, F. S., Hattab, M. W., Clemente, C. L., Biggerstaff, M. & Santillana, M. Improved state-level influenza nowcasting in the United States leveraging internet-based data and network approaches. *Nat. Commun.* **10**, 147 (2019).
- Gan, Z., Yang, M., Feng, T. & Timmermans, H. Understanding urban mobility patterns from a spatiotemporal perspective: Daily ridership profiles of metro stations. *Transportation* **47**, 315–336 (2020).
- Rasmussen, C. E. & Williams, C. K. I. *Gaussian Processes for Machine Learning* (The MIT Press, 2006).
- Cressie, N. & Wikle, C. K. *Statistics for Spatio-Temporal Data*. Wiley Series in Probability and Statistics (John Wiley & Sons, 2011).
- Wikle, C. K., Zammit-Mangion, A. & Cressie, N. *Spatio-Temporal Statistics with R* (Chapman and Hall/CRC, 2019).
- Rue, H. et al. Bayesian computing with INLA: A review. *Annu. Rev. Stat. Appl.* **4**, 395–421 (2017).
- Anderson, S. C., Ward, E. J., English, P. A. & K., B. L. A. sdmTMB: An R package for fast, flexible, and user-friendly generalized linear mixed effects models with spatial and spatiotemporal random fields. *bioRxiv* (2022).
- Hamelijnck, O., Wilkinson, W., Loppi, N., Solin, A. & Damoulas, T. Spatio-temporal variational Gaussian processes. In *Proc. 35th International Conference on Neural Information Processing Systems*, vol. 34 of *Advances in Neural Information Processing Systems*, 23621–23633 (Curran Associates, Inc., 2021).
- Zhang, J., Ju, Y., Mu, B., Zhong, R. & Chen, T. An efficient implementation for spatial-temporal Gaussian process regression and its applications. *Automatica* **147**, 110679 (2023).
- Zammit-Mangion, A. & Cressie, N. FRK: an R package for spatial and spatio-temporal prediction with large datasets. *J. Stat. Softw.* **98**, 1–48 (2021).
- Banerjee, S. Modeling massive spatial datasets using a conjugate Bayesian linear modeling framework. *Spat. Stat.* **37**, 100417 (2020).
- Neal, R. M. *Bayesian Learning for Neural Networks*. Ph.D. thesis (University of Toronto, 1996).
- Mildenhall, B. et al. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM* **65**, 99–106 (2021).
- Hoffman, M. D. et al. ProbNeRF: Uncertainty-aware inference of 3D shapes from 2D images. In *International Conference on Artificial Intelligence and Statistics*, 10425–10444 (PMLR, Norfolk, 2023).
- Tancik, M. et al. Fourier features let networks learn high frequency functions in low dimensional domains. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, vol. 33 of *Advances in Neural Information Processing Systems*, 7537–7547 (Curran Associates, Inc., 2020).
- Wikle, C. K. & Zammit-Mangion, A. Statistical deep learning for spatial and spatiotemporal data. *Annu. Rev. Stat. Appl.* **10**, 247–270 (2023).
- McDermott, P. L. & Wikle, C. K. Bayesian recurrent neural network models for forecasting and quantifying uncertainty in spatial-temporal data. *Entropy* **21**, 184 (2019).
- Amato, F., Guignard, F., Sylvain, R. & Kanevski, M. A novel framework for spatio-temporal prediction of environmental data using deep learning. *Nat. Sci. Rep.* **10**, 22243 (2020).
- Gao, F., Xu, Z. & Yin, L. Bayesian deep neural networks for spatio-temporal probabilistic optimal power flow with multi-source renewable energy. *Appl. Energy* **353**, 122106 (2024).
- Liu, Y. et al. Probabilistic spatiotemporal wind speed forecasting based on a variational Bayesian deep learning model. *Appl. Energy* **260**, 114259 (2020).
- Wang, J. et al. Predicting wind-caused floater intrusion risk for overhead contact lines based on Bayesian neural network with spatiotemporal correlation analysis. *Reliab. Eng. Syst. Saf.* **225**, 108603 (2022).
- Garnelo, M. et al. Neural processes (2018). [arXiv:1807.01622](https://arxiv.org/abs/1807.01622).
- Jin, M. et al. A survey on graph neural networks for time series: forecasting, classification, imputation, and anomaly detection (2023). [arXiv:2307.03759](https://arxiv.org/abs/2307.03759).
- LeCun, Y. A., Bottou, L., Orr, G. B. & Müller, K.-R. Efficient backprop. In Montavon, G., Orr, G. B. & Müller, K.-R. (eds.) *Neural Networks: Tricks of the Trade*, 9–48, 2nd edn (Springer, 2012).
- Pearce, T., Tsuchida, R., Zaki, M., Brintrup, A. & Neely, A. Expressive priors in Bayesian neural networks: Kernel combinations and periodic functions. In *Proceedings of the 35th Uncertainty in Artificial Intelligence Conference*, vol. 115 of *Proceedings of Machine Learning Research*, 134–144 (PMLR, Norfolk, 2020).
- Haslett, J. & Raftery, A. E. Space-time modelling with long-memory dependence: assessing Ireland’s wind power resource. *J. R. Stat. Soc. C (Appl. Stat.)* **38**, 1–50 (1989).
- Pebesma, E. spactime: spatio-temporal data in R. *J. Stat. Softw.* **51**, 1–30 (2012).
- UCI Machine Learning Repository. Hungarian chickenpox cases. <https://doi.org/10.24432/C5103B> (2021).
- University Corporation for Atmospheric Research. US precipitation and temperature data 1895–1997. <https://www.image.ucar.edu/Data/US.monthly.met/> (2010).
- Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

43. Oreshkin, B. N., Carpvov, D., Chapados, N. & Yoshua, B. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations* (2020).
44. Nixtla Labs. NeuralForecast: Scalable and user friendly neural forecasting algorithms. <https://github.com/Nixtla/neuralforecast> (2024).
45. Blundell, C., Cornebise, J., Kavukcuoglu, K. & Wierstra, D. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, **37** of *Proceedings of Machine Learning Research*, 1613–1622 (PMLR, Norfolk, 2015).
46. Chandrupatla, T. R. A new hybrid quadratic/bisection algorithm for finding the zero of a nonlinear function without using derivatives. *Adv. Eng. Softw.* **28**, 145–149 (1997).
47. Lakshminarayanan, B., Pritzel, A. & Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, of *Advances in Neural Information Processing Systems* **30**, 6405–6416 (Curran Associates, Inc., 2017).
48. Esri, DigitalGlobe, GeoEye, i-cubed, USDA FSA, USGS, AEX, Getmapping, Aerogrid, IGN, IGP, swisstopo, and the GIS User Community. World imagery [basemap] - captured Mar 15, 2022. <https://www.arcgis.com/home/item.html?id=10df2279f9684e4a9f6a7f08fcbac2a9>.
49. CIRCLEDC Stadia Maps, CIRCLEDC OpenMapTiles, CIRCLEDC OpenStreetMap, CIRCLEDC Stamen Design, CIRCLEDC CNES, Distribution Airbus DS, CIRCLEDC Airbus DS, CIRCLEDC PlanetObserver (Contains Copernicus Data). Stadia maps. <https://stadiamaps.com>, <https://stamen.com>, <https://openstreetmap.org/copyright>.

Author contributions

The statistical model was designed and implemented by F.S., M.H., J.B., and U.K. Evaluations were designed by F.S. and implemented by F.S., J.B., C.C., and B.P.; R.S. and B.P. provided guidance and oversight. F.S. drafted the manuscript, all authors contributed to its revision and completion.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41467-024-51477-5>.

Correspondence and requests for materials should be addressed to Feras Saad.

Peer review information *Nature Communications* thanks Tom Rainforth and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2024