Towards High Resolution Probabilistic Coastal Inundation Forecasting from Sparse Observations

Kazi Ashik Islam[1*], Zakaria Mehrab[1*], Mahantesh M Halappanavar[2], Henning Mortveit[1], Katragadda Sridhar[3], Jon Derek Loftis[4], Stefan Hoops[1], Madhav Marathe[1]

[1]Biocomplexity Institute, University of Virginia
[2]Pacific Northwest National Laboratory
[3] Department of Communications and Information Technology, City of Virginia Beach
[4] Virginia Institute of Marine Science, College of William and Mary
{ki5hd, zm8bh}@virginia.edu, hala@pnnl.gov, Henning.Mortveit@virginia.edu, SKatraga@vbgov.com, jdloftis@vims.edu, {shoops, marathe}@virginia.edu

Abstract

Coastal flooding poses increasing threats to communities worldwide, necessitating accurate and hyper-local inundation forecasting for effective emergency response. However, real-world deployment of forecasting systems is often constrained by sparse sensor networks, where only a limited subset of locations may have sensors due to budget constraints. To approach this challenge, we present DIFF-SPARSE, a masked conditional diffusion model designed for probabilistic coastal inundation forecasting from sparse sensor observations. DIFF-SPARSE primarily utilizes the inundation history of a location and its neighboring locations from a context time window as spatiotemporal context. The fundamental challenge of spatiotemporal prediction based on sparse observations in the context window is addressed by introducing a novel masking strategy during training. Digital elevation data and temporal co-variates are utilized as additional spatial and temporal contexts, respectively. A convolutional neural network and a conditional UNet architecture with cross-attention mechanism are employed to capture the spatiotemporal dynamics in the data. We trained and tested DIFF-SPARSE on coastal inundation data from the Eastern Shore of Virginia and systematically assessed the performance of DIFF-SPARSE across different sparsity levels ($0\%, 50\%, 95\%$ missing observations). Our experiment results show that DIFF-SPARSE achieves upto 62% improvement in terms of two forecasting performance metrics compared to existing methods, at 95% sparsity level. Moreover, our ablation studies reveal that digital elevation data becomes more useful at high sparsity levels compared to temporal co-variates.

Code — https://github.com/KAI10/Diff-Sparse
Extended version — https://arxiv.org/abs/2505.05381

Introduction

The projection in sea level rise have highlighted the increasing vulnerability of many coastal regions to inundation (Wuebbles et al. 2017). Among these regions, the East Coast in US is a region of particular policy concern due to its geography and low-lying topography (Ezer and Atkinson 2014). In recent periods, this region has been experiencing frequent inundation events (Ezer 2020), causing disruptions in daily life (e.g., traffic, economic) (Jacobs et al. 2018). In future, this situation will only continue to decline due to the combined effect of climate change, sea level rise, and urbanization (Swain et al. 2020). Therefore, forecasting coastal inundation is of utmost importance from a policy standpoint for undertaking risk prevention measurement and safe evacuation of residents during emergencies (Yang et al. 2019).

The problem of forecasting coastal inundation from sparse observation was brought to our attention through conversations with staff at the City of Virginia Beach. The region faces challenges with coastal flooding which causes slow-down and/or road closures; disrupting the day-to-day activities of the inhabitants. A fast and high-resolution flood forecasting model would enable city officials to identify localized flood risks, resulting in more effective preparedness and response. In order to effectively identify inundated locations, policymakers deploy sensors at various locations (Tao et al. 2024). These sensors are responsible for providing periodical inundation reports and other variables of interest in the surrounding area. However, there are several challenges associated with these sensor placements. *First,* due to budget and resource constraints, the number of sensors that can be deployed is limited (Saad et al. 2024; Karagulian et al. 2019). *Second,* the readings from the sensor are often associated with noise due to mechanical, environmental or calibration issues (Barrenetxea et al. 2008; Mydlarz et al. 2024). *Third,* due to the difficulty associated with finding an optimal location assignment of spatial sensors (Krause, Singh, and Guestrin 2008) or changes carried out by the stakeholders, the locations of these sensors may change from time to time (Garg, Singh, and Ramos 2012). As a result, a forecasting method that relies on sensor data but does not account for the sparsity and uncertain nature of sensor placements will not be reliable (Tao et al. 2024).

The traditional method for predicting inundation involves hydro-dynamical, physics-based models (PBM). Commer-

*These authors contributed equally.

cially available (Syme 2001; Deltares 2014) (e.g., SOBEK, TUFLOW) or open-sourced (Zhang et al. 2016) (e.g, SCHISM) hydro-dynamical models can simulate inundation accurately at a high resolution. However, the computation time associated with these PBMs makes them challenging for real-time forecasting (Guo et al. 2021). Researchers have investigated the use of Deep-Learning (DL) based prediction methods as surrogates to these PBMs (Roy et al. 2023). However, these models often struggle to: (i) scale in large-scale high-resolution forecasting tasks under sparse availability of data, and (ii) effectively use additional spatiotemporal contexts. Therefore, a surrogate model to PBMs that can make accurate real-time predictions based on sparse sensor observations, will have high relevance to policymakers.

Motivated by the above observations, we explore Diffusion Models (Sohl-Dickstein et al. 2015) for high-resolution coastal inundation forecasting task. The reason is two-fold: their potential to model physical properties (Ahmad, Venkataswamy, and Fox 2024; Amram and Pedro 2023), and their ability to utilize conditioning contexts of various dimensions (Rombach et al. 2022; Zhang et al. 2023) that can improve prediction quality in the presence of sparse observations. Although diffusion models were originally developed for image generation, they have been adapted for time-series and spatiotemporal forecasting (Rasul et al. 2021; Wen et al. 2023). However, the existing methods struggle in terms of computational scalability and performance, under sparse availability of data. Using the coastal inundation of the Virginia Eastern Shore as case study, we address these challenges by making the following contributions:

- We formulate the problem of probabilistic inundation forecasting from sparse observations as a conditional distribution function learning problem and present DIFF-SPARSE, a scalable model to learn this function. DIFF-SPARSE is trained using data from physics-based hydrodynamic model; however, during inference it makes probabilistic forecasts based on sparse sensor observations and additional spatiotemporal contexts (e.g., elevation, temporal co-variates).
- We introduce a novel masking strategy to train DIFF-SPARSE which enables it to make forecasts from sparse observations. Our robust training strategy enables DIFF-SPARSE to make accurate spatiotemporal forecasts from different spatial placement configurations of the sensors, without requiring training the model from scratch.
- Extensive evaluation of our method against several baseline methods under sparse observation underscores the superiority of our method at large-scale high resolution inundation forecasting tasks. Specifically, we observe upto 62% improvement in prediction performance over existing methods in terms of two predictive performance metrics.
- We perform extensive ablation studies to highlight the utility of different context data (e.g., sensor data, elevation, temporal co-variates) at varying sparsity levels.

Related Work

We present the most relevant literature in this section (see (Islam et al. 2025) for a more extensive literature review).

Physcis-based Models: Traditionally, various Physics-Based hydrodynamic Models (PBMs) have been employed to simulate inundation. SOBEK (Deltares 2014) is a two-dimensional model that solves the Saint-Venant flow equations and shallow water equations to determine water levels across rectangular grids. Similarly, TELEMAC-2D (Vu et al. 2015) employs comparable method specifically tailored for coastal applications. Polymorphic models such as SCHISM (Zhang et al. 2016) captures water flow at very high resolutions. However, the substantial computational time associated with solving these complex equations at high spatial resolution renders these PBMs impractical for real-time forecasting applications.

Deep Learning Researchers have explored Deep Learning (DL) methods for developing surrogates to PBMs. Regression-based methods (Zahura and Goodall 2022), recurrent neural networks (Roy et al. 2023), and hybrid machine learning-based methods (Zanchetta and Coulibaly 2022) have emerged as popular choices, among others. However, these models inherently struggle to incorporate spatiotemporal contexts effectively. Originally developed for traffic forecasting, the Diffusion Convolutional Recurrent Neural Network (DCRNN) method proposed by (Li et al. 2017) captures some aspect of spatiotemporal contexts by utilizing bidirectional random walks and Sequence-to-Sequence architecture. More recently, BayesNF (Saad et al. 2024) has emerged as the state-of-the-art DL spatiotemporal forecasting model, constructed using a Bayesian Neural Network framework. By assigning prior distributions to model parameters and adjusting the posterior based on observations, BayesNF effectively learns to map multivariate spatiotemporal points to continuous real-valued fields.

Diffusion Model The field of generative modeling has been dominated by the diffusion model for quite some time. Although it had primarily been studied for image synthesis (Ho, Jain, and Abbeel 2020; Dhariwal and Nichol 2021; Austin et al. 2021), its fascinating ability to use flexible conditioning to guide the generation process has been utilized in other domains like video generation (Yang, Srivastava, and Mandt 2023), prompt-based image generation (Ramesh et al. 2022), text-to-speech generation (Yang et al. 2023). TimeGrad (Rasul et al. 2021) was the pioneering method that approached the time-series forecasting problem by utilizing a diffusion model. Other models like CSDI (Tashiro et al. 2021), DPSD-CSPD (Biloš et al. 2023), TDSTF (Chang et al. 2024) were later introduced. These models try to predict the noise added during the forward process of the diffusion model (ϵ-parameterization). An alternative parameterization is to predict the ground-truth data (x-parameterization). Different from models mentioned above, D^3VAE (Li et al. 2022) uses coupled diffusion process and bidirectional variational autoencoder to better forecast in longer horizon and improve interpretability. The State-of-the-Art Diffusion-based model DiffSTG (Wen et al. 2023) devised a modified architecture for the reverse process of diffusion model combining UNet and GNN to capture temporal and spatial dependencies, respectively. However, both methods scale poorly with the growth of variables.

Problem Formulation

Formally, we define the inundation forecasting from sparse observations problem as follows.

Problem 1 ***Multi-patch Probabilistic Inundation Forecasting.*** *Let $\mathcal{P} = \{P_1, P_2, ..., P_K\}$ denote a set of two-dimensional rectangular grids each of size $D \times D$. We refer to P_k as the k^{th} 'patch'. Let $\mathcal{M} = \{\mathbf{M}_1, \mathbf{M}_2, ..., \mathbf{M}_K\}$ denote a set of binary matrices, henceforth referred to as 'sensor masks'. If $\mathbf{M}_k(i,j) = 1$ then the cell (i,j) of patch P_k has a water-level sensor; $\mathbf{M}_k(i,j) = 0$ means cell (i,j) of patch P_k does not have a water-level sensor. Let ${}_k\mathbf{y}_t^0 \in \mathbb{R}^{D\times D}$ denote the inundation matrix of patch P_k at timestep t. If $\mathbf{M}_k(i,j) = 1$ then ${}_k\mathbf{y}_{ij,t}^0 \in \mathbb{R}$ contains the inundation level on cell (i,j) of patch P_k at time step t; $i,j \in \{1,...,D\}, t \in \{1,...,T\}$. However, for cells (i,j) where $\mathbf{M}_k(i,j) = 0$, ${}_k\mathbf{y}_{ij,t}^0, t \in \{1,...,T\}$ has missing value. Let ${}_k\mathbf{x}_t^0 \in \mathbb{R}^{D\times D}$ denote the complete inundation matrix of patch P_k at timestep t, where there are no missing values. Let $\mathbf{s}_k \in \mathbb{R}^{D\times D}$ and ${}_k\mathbf{z}_t \in \mathbb{R}^l, t \in \{1,\ldots,T\}$ denote the elevation matrix and time series of co-variates associated with patch P_k, respectively. Let $t_0 : t_{c+l}$ denote the increasing sequence of timesteps $t_0, t_1, ..., t_{c+l}$, where $c, l > 0$. We aim to learn the conditional distribution function $q({}_k\mathbf{x}_{t_c:t_{c+l}}^0 | {}_k\mathbf{y}_{t_0:t_{c-1}}^0, {}_k\mathbf{z}_t, \mathbf{s}_k, \mathbf{M}_k)$.*

Here, we aim to forecast the inundation on all cells in a patch (i.e., $\mathbf{x}$) even though in the context inundation history (i.e., $\mathbf{y}$) we only know the inundation levels on cells where sensors are placed. Problem 1 is a probabilistic forecasting problem, as we want to learn the distribution of inundation values instead of making a single prediction.

The following section presents the basics of a denoising diffusion model used by our method DIFF-SPARSE for learning our target distributions. Subsequently, we present details on how we have designed DIFF-SPARSE to capture spatial and temporal dynamics of inundation values.

Denoising Diffusion Model

In this section, we provide a brief overview of denoising diffusion models. A more comprehensive view is provided in the extended version of the paper (Islam et al. 2025).

Let $\mathbf{x}^0 \sim q_\chi(\mathbf{x}^0)$ denote the multivariate training vector from some input space $\chi = \mathbb{R}^D$, where $q_\chi(\mathbf{x}^0)$ denotes the true distribution. In diffusion probabilistic models, we aim to approximate $q_\chi(\mathbf{x}^0)$ by a probability density function $p_\theta(\mathbf{x}^0)$ where θ denotes the set of trainable parameters. Diffusion probabilistic models (Sohl-Dickstein et al. 2015; Luo 2022) are a special class of Hierarchical Variational Autoencoders (Kingma et al. 2016; Sønderby et al. 2016) with the form $p_\theta(\mathbf{x}^0) = \int p_\theta(\mathbf{x}^{0:N}) d\mathbf{x}^{1:N}$; here $\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^N$ are latent variables. Three key properties of diffusion models are: (i) all of the latent variables are assumed to have the dimension D as $\mathbf{x}^0$, (ii) the approximate posterior ,

$$q(\mathbf{x}^{1:N}|\mathbf{x}^0) = \prod_{n=1}^N q(\mathbf{x}^n|\mathbf{x}^{n-1})$$

is fixed to a Markov chain (often referred to as the *forward* process), and (iii) the structure of the latent encoder at each hierarchical step is pre-defined as a linear Gaussian model:

$$q(\mathbf{x}^n|\mathbf{x}^{n-1}) = \mathcal{N}(\mathbf{x}^n; \sqrt{1-\beta_n}\mathbf{x}^{n-1}, \beta_n\mathbf{I}).$$

Additionally, the Gaussian parameters $(\beta_1, \beta_2, ..., \beta_n)$ are chosen in such a way that we have: $q(\mathbf{x}^N) = \mathcal{N}(\mathbf{x}^N; \mathbf{0}, \mathbf{I})$.

The joint distribution $p_\theta(\mathbf{x}^{0:N})$ is called the reverse process. It is defined as a Markov chain with learned Gaussian transitions beginning at $p(\mathbf{x}^N)$:

$$p_\theta(\mathbf{x}^{0:N}) = p(\mathbf{x}^N) \prod_{n=1}^N p_\theta(\mathbf{x}^{n-1}|\mathbf{x}^n)$$

Each subsequent transition is parameterized as follows:

$$p_\theta(\mathbf{x}^{n-1}|\mathbf{x}^n) = \mathcal{N}(\mathbf{x}^{n-1}; \mu_\theta(\mathbf{x}^n, n), \Sigma_\theta(\mathbf{x}^n, n)\mathbf{I}) \quad (1)$$

Both $\mu_\theta : \mathbb{R}^D \times \mathbb{N} \to \mathbb{R}^D$ and $\Sigma_\theta : \mathbb{R}^D \times \mathbb{N} \to \mathbb{R}^+$ take two inputs: $\mathbf{x}^n \in \mathbb{R}^D$ and the noise step $n \in \mathbb{N}$. The parameters θ are learned by fitting the model to the data distribution $q_\chi(\mathbf{x}^0)$. This is done by minimizing the negative log-likelihood via a variational bound:

$$-\log p_\theta(\mathbf{x}^0) \leq \mathbb{E}_{q(\mathbf{x}^{1:N}|\mathbf{x}^0)} \log \frac{q(\mathbf{x}^{1:N}|\mathbf{x}^0)}{p_\theta(\mathbf{x}^{0:N})} \quad (2)$$

We indirectly minimize the negative log-likelihood of the data by minimizing the right hand side of (2).

(Ho, Jain, and Abbeel 2020) showed that the forward process has the following property: we can sample $\mathbf{x}^n$ using $\mathbf{x}^0$ at any noise step n in closed form . Let, $\alpha_n = 1 - \beta_n$, and $\bar{\alpha}_n = \prod_{i=1}^n \alpha_i$. Then, we have:

$$q(\mathbf{x}^n|\mathbf{x}^0) = \mathcal{N}\left(\mathbf{x}^n; \sqrt{\bar{\alpha}_n}\mathbf{x}^0, (1-\bar{\alpha}_n)\mathbf{I}\right). \quad (3)$$

Based on this property, (Ho, Jain, and Abbeel 2020) showed that one possible parameterization of (1) is as follows:

$$\mu_\theta(\mathbf{x}^n, n) = \frac{\sqrt{\alpha_n}(1-\bar{\alpha}_{n-1})}{1-\bar{\alpha_n}}\mathbf{x}^n + \frac{\sqrt{\bar{\alpha}_{n-1}}\,\beta_n}{1-\bar{\alpha_n}}\hat{\mathbf{x}}_\theta(\mathbf{x}^n, n) \quad (4)$$

$$\Sigma_\theta(\mathbf{x}^n, n) = \frac{1-\bar{\alpha}_{n-1}}{1-\bar{\alpha_n}}\beta_n \quad (5)$$

Here, $\hat{\mathbf{x}}_\theta(\mathbf{x}^n, n)$ is a neural network that predicts $\mathbf{x}^0$ from noisy vector $\mathbf{x}^n$ and noise step n. (Ho, Jain, and Abbeel 2020) then showed that, the right hand side of (2) can be minimized by minimizing:

$$\mathbb{E}_{n,q(\mathbf{x}^n|\mathbf{x}^0)} \frac{1}{2\tilde{\beta}_n} \frac{\bar{\alpha}_{n-1}\,\beta_n^2}{(1-\bar{\alpha}_n)^2} ||\mathbf{x}^0 - \hat{\mathbf{x}}_\theta(\mathbf{x}^n, n)||_2^2 \quad (6)$$

Therefore, optimizing a diffusion model can be done by training a neural network to predict the original ground truth vector from an arbitrarily noisy version of it. Once trained, to sample from the reverse process $\mathbf{x}^{n-1} \sim p_\theta(\mathbf{x}^{n-1}|\mathbf{x}^n)$, we first sample $\mathbf{x}^N$ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Then, we compute:

$$\mathbf{x}^{n-1} = \mu_\theta(\mathbf{x}^n, n) + \sqrt{\Sigma_\theta}\,\mathbf{u} \quad (7)$$

Here, $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for $n \in [2, N]$, and $\mathbf{u} = \mathbf{0}$ when $n = 1$. $\mu_\theta(\mathbf{x}^n, n)$, Σ_θ are computed using (4) and (5), respectively.

DIFF-SPARSE Method

DIFF-SPARSE is designed for probabilistic inundation forecasting from sparse observations. Following our notations in Problem 1, let ${}_k\mathbf{y}_t^0 \in \mathbb{R}^{D\times D}$ denote the 'sparse' inundation matrix of the patch P_k at time step t, with sensor mask $\mathbf{M}_k$. Let ${}_k\mathbf{x}_t^0 \in \mathbb{R}^{D\times D}$ denote the complete inundation matrix of patch P_k at timestep t. Towards our goal of learning the conditional distribution $q_\chi({}_k\mathbf{x}^0_{t_c:t_{c+l}}|{}_k\mathbf{y}^0_{t_0:t_{c-1}}, {}_k\mathbf{z}_{t_0:t_{c-1}}, \mathbf{s}_k, \mathbf{M}_k)$, where $\mathbf{s}_k \in \mathbb{R}^{D\times D}$ is the elevation matrix of patch P_k, and ${}_k\mathbf{z}_t$ denotes a time-series of co-variates associated with P_k, we assume:

$$q_\chi({}_k\mathbf{x}^0_{t_c:t_{c+l}}|{}_k\mathbf{y}^0_{t_0:t_{c-1}}, {}_k\mathbf{z}_{t_0:t_{c-1}}, \mathbf{s}_k, \mathbf{M}_k) = \prod_{t=t_c}^{t_{c+l}} q_\chi({}_k\mathbf{x}^0_t|{}_k\mathbf{y}^0_{t_0:t-1}, {}_k\mathbf{z}_{t_0:t-1}, \mathbf{s}_k, \mathbf{M}_k) \quad (8)$$

In DIFF-SPARSE, we use a denoising diffusion model to learn the conditional distributions on the right hand side of (8). We refer to ${}_k\mathbf{y}^0_{t_0:t-1}$, ${}_k\mathbf{z}_{t_0:t-1}$, $\mathbf{s}_k$, and $\mathbf{M}_k$ together as *context* data. To capture spatial correlation, we augment the sparse inundation data in the context (i.e., ${}_k\mathbf{y}^0_{t_0:t-1}$) by concatenating the elevation matrix $\mathbf{s}_k$ and the sensor mask $\mathbf{M}_k$ to it as additional channels (9). The goal is to capture the correlation of elevation and inundation values. We then use convolution blocks to extract spatial features ${}_k\mathbf{f}_{t_0:t-1}$ (10).

$${}_k\mathbf{v}_{t_0:t-1} = {}_k\mathbf{y}^0_{t_0:t-1} \oplus \mathbf{s}_k \oplus \mathbf{M}_k \quad (9)$$

$${}_k\mathbf{f}_{t_0:t-1} = \text{CONV}_{\theta_1}({}_k\mathbf{v}_{t_0:t-1}) \quad (10)$$

Since coastal inundation is expected to be correlated with tide cycles, we use hour of day, and day of month as temporal features $\mathbf{z}_t$. We use sinusoidal encoding (Vaswani et al. 2017) to capture the cyclic nature of these temporal features.

$${}_k\mathbf{g}_{t_0:t-1} = \text{SINUSOID}({}_k\mathbf{z}_{t_0:t-1}) \quad (11)$$

We concatenate the spatial and temporal features, and then pass it through a linear fully connected layer to get an embedding of the context.

$${}_k\mathbf{h}_{t_0:t-1} = \text{LINEAR}_{\theta_2}({}_k\mathbf{f}_{t_0:t-1} \oplus {}_k\mathbf{g}_{t_0:t-1}) \quad (12)$$

An overview of DIFF-SPARSE architecture is provided in Figure 1. We approximate (8) by the model in (13). Here, θ comprises of the trainable parameters of the convolution blocks (θ_1), fully connected layer (θ_2), and the denoising diffusion model (θ_3).

$$\prod_{t=t_c}^{t_{c+l}} p_\theta({}_k\mathbf{x}^0_t|{}_k\mathbf{h}_{t_0:t-1}) \quad (13)$$

As described in the previous section, within the denoising diffusion model, we need to train a neural network $\hat{\mathbf{x}}_{\theta_3}({}_k\mathbf{x}^n_t, n, {}_k\mathbf{h}_{t_0:t-1})$ that predicts ${}_k\mathbf{x}^0_t$ from noisy vector ${}_k\mathbf{x}^n_t$, noise step n, and context embedding ${}_k\mathbf{h}_{t_0:t-1}$. We use a conditional UNet for this purpose. Within the UNet architecture, we employ a cross-attention conditioning mechanism (Rombach et al. 2022) to map the context embedding to the intermediate layers of the UNet. It allows us to predict ${}_k\mathbf{x}^0_t$ while paying attention to the context embedding.

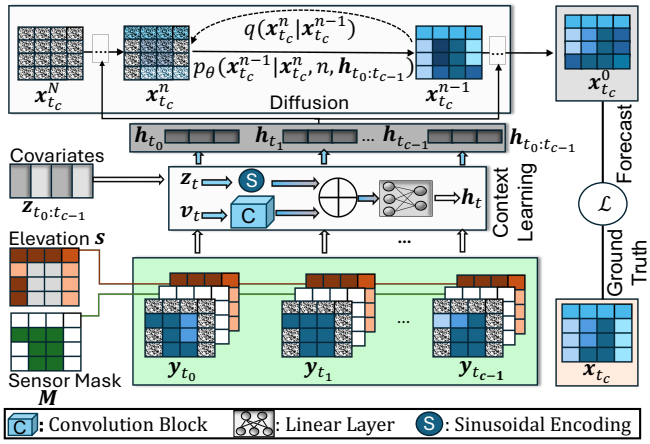

Figure 1: DIFF-SPARSE architecture. It has two main components. The first component involves context learning, where the sparse context inundation data $(\mathbf{y}_{t_0}, \mathbf{y}_{t_1}, ..., \mathbf{y}_{t_{c-1}})$, sensor mask $(\mathbf{M})$, elevation data $(\mathbf{s})$ and temporal covariates $(\mathbf{z}_{t_0:t_{c-1}})$ are used to generate context embedding $\mathbf{h}_{t_0:t_{c-1}}$. The second component involves a diffusion model where the corresponding forecast is sampled by conditioning on the context embedding. $q(\mathbf{x}^n_{t_c}|\mathbf{x}^{n-1}_{t_c})$ denotes the forward diffusion process; $p_\theta(\mathbf{x}^{n-1}_{t_c}|\mathbf{x}^n_{t_c}, n, \mathbf{h}_{t_0:t_{c-1}})$ denotes the reverse diffusion process, parameterized by θ.

Training

To train DIFF-SPARSE, we sample the following from the training time-series data of each patch: (i) inundation history in the context window $({}_k\mathbf{x}^0_{t_0:t-1})$ and temporal covariates $({}_k\mathbf{z}_{t_0,t-1})$, (ii) inundation values in the immediate next timestep $({}_k\mathbf{x}^0_t)$. DIFF-SPARSE is trained to forecast the latter. We apply Algorithm 1 on each sample. First, we generate a random binary sensor mask $\mathbf{M}_k$. This mask is applied on the context inundation history $({}_k\mathbf{x}^0_{t_0:t-1})$ to generate a sparse inundation history ${}_k\mathbf{y}^0_{t_0:t-1}$ (line 3-5, $\odot$ denotes

Algorithm 1: Training step on a data point of P_k.

Input: Data: ${}_k\mathbf{x}^0_t \sim q({}_k\mathbf{x}^0_t)$, Context: ${}_k\mathbf{x}^0_{t_0:t-1}$, ${}_k\mathbf{z}_{t_0:t-1}$, $\mathbf{s}_k$.

1 **while** *not converged* **do**
// Masking
2 Randomly generate a binary sensor mask $\mathbf{M}_k$.
3 **for** $\tau \in [t_0, t-1]$ **do**
4 $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5 ${}_k\mathbf{y}^0_\tau = {}_k\mathbf{x}^0_\tau \odot \mathbf{M}_k + (\mathbf{1} - \mathbf{M}_k) \odot \mathbf{u}$
// Compute context embedding
6 ${}_k\mathbf{f}_{t_0:t-1} \leftarrow \text{CONV}_{\theta_1}({}_k\mathbf{y}^0_{t_0:t-1} \oplus \mathbf{s}_k \oplus \mathbf{M}_k)$
7 ${}_k\mathbf{g}_{t_0:t-1} \leftarrow \text{SINUSOID}({}_k\mathbf{z}_{t_0:t-1})$
8 ${}_k\mathbf{h}_{t_0:t-1} \leftarrow \text{LINEAR}_{\theta_2}({}_k\mathbf{f}_{t_0:t-1} \oplus {}_k\mathbf{g}_{t_0:t-1})$
// Forward diffusion
9 Initialize $n \sim \text{Uniform}(1, N)$.
10 ${}_k\mathbf{x}^n_t \leftarrow \mathcal{N}\left(\sqrt{\bar{\alpha}_n}\ {}_k\mathbf{x}^0_t, (1-\bar{\alpha}_n)\mathbf{I}\right)$.
// prediction
11 ${}_k\hat{\mathbf{x}}^0_t \leftarrow \hat{\mathbf{x}}_{\theta_3}({}_k\mathbf{x}^n_t, n, {}_k\mathbf{h}_{t_0:t-1})$
12 Take gradient step on $\nabla_\theta||{}_k\mathbf{x}^0_t - {}_k\hat{\mathbf{x}}^0_t||^2_2$

Algorithm 2: Sampling ${}_k\mathbf{x}_t^0$

Input: Noise: ${}_k\mathbf{x}_t^N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, Context: ${}_k\mathbf{y}_{t_0:t-1}^0$, ${}_k\mathbf{z}_{t_0:t-1}$, $\mathbf{s}_k$, and $\mathbf{M}_k$.

1 **for** $n \in N, ..., 1$ **do**
2 $\quad \mathbf{u} \leftarrow \mathbf{0}$
3 $\quad$ **if** $n > 1$ **then**
4 $\quad\quad \mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5 $\quad$ Compute context embedding ${}_k\mathbf{h}_{t_0:t-1}$.
6 $\quad {}_k\hat{\mathbf{x}}_t^0 \leftarrow \hat{\mathbf{x}}_\theta({}_k\mathbf{x}_t^n, n, {}_k\mathbf{h}_{t_0:t-1})$
7 $\quad {}_k\mathbf{x}_t^{n-1} \leftarrow \frac{\sqrt{\alpha_n}(1-\bar{\alpha}_{n-1})}{1-\bar{\alpha}_n} {}_k\mathbf{x}_t^n + \frac{\sqrt{\bar{\alpha}_{n-1}}\beta_n}{1-\bar{\alpha}_n} {}_k\hat{\mathbf{x}}_t^0$
8 $\quad\quad + \sqrt{\Sigma_\theta}\mathbf{u}$
9 **return** ${}_k\mathbf{x}_t^0$

element-wise multiplication). Here, we retain the inundation values in cells where sensors are placed (according to the sensor mask $\mathbf{M}_k$). For cells that do not have sensors, their inundation values are replaced with standard gaussian noise. This serves as a signal to the model to ignore these values. As we train DIFF-SPARSE with different random sensor masks, it is able to make forecasts based on different sensor placement configurations during inference. We use the sparse inundation history, temporal co-variates and the sensor mask to calculate the context embedding ${}_k\mathbf{h}_{t_0:t-1}$ (lines 6-8). We then optimize the model parameters $(\theta_1, \theta_2, \theta_3)$ by minimizing the negative log-likelihood of the data conditioned on the context embedding $-\log p_\theta({}_k\mathbf{x}_t^0 | {}_k\mathbf{h}_{t_0:t-1})$. Through a similar derivation as shown in the previous section, the objective to minimize becomes:

$$\mathbb{E}_{{}_k\mathbf{x}_t^0, n} || {}_k\mathbf{x}_t^0 - \hat{\mathbf{x}}_{\theta_3}({}_k\mathbf{x}_t^n, n, {}_k\mathbf{h}_{t_0:t-1}||_2^2 \tag{14}$$

Here, the neural network $\hat{x}_{\theta_3}$ is now also conditioned on the context embedding ${}_k\mathbf{h}_{t_0:t-1}$.

Inference

During inference, given a context $({}_k\mathbf{y}_{t_0:t-1}^0, {}_k\mathbf{z}_{t_0:t-1}, \mathbf{s}_k, \mathbf{M}_k)$, we first compute the context embedding ${}_k\mathbf{h}_{t_0:t-1}$. We then apply Algorithm 2 to obtain a sample ${}_k\mathbf{x}_t^0$ of the next time step. Based on this sample, we compute the new context embedding ${}_k\mathbf{h}_{t_0:t}$ and repeat until the desired number of prediction time steps have been reached. We repeat the process multiple times to get multiple predictions.

Experiment

In our experiments, we first evaluate DIFF-SPARSE on the TideWatch dataset against six competitive baseline models; taking forecasting inundation within the *Virginia Eastern Shore* region as our case study. Then, we conduct an ablation study to demonstrate the importance of the different context information utilized in DIFF-SPARSE.

Dataset

To train DIFF-SPARSE, we downloaded inundation data of the Virginia Eastern Shore from Tidewatch Maps (Loftis et al. 2019; Loftis 2022); a street-level inundation mapping tool developed by the Virginia Institute of Marine Science (VIMS). It utilizes the open-source hydro-dynamic model, SCHISM (Zhang et al. 2016) and provides 36 hour inundation forecast maps with a 12 hour update frequency. We used this dataset due to its high spatial resolution, coverage of areas affected by coastal inundation, and lack of other publicly available inundation datasets. Moreover, using it as training data allows DIFF-SPARSE to learn to make forecasts from a physics-based hydro-dynamic model.

We downloaded 89 days of hourly inundation data of our study area. The data comes in a raster geodatabase format with a pixel resolution of $10m \times 10m$ and provides inundation level at each pixel in feet. We resampled the data to a pixel resolution of $30m \times 30m$. We then consider different square-shaped patches. Each patch P_k has a time series of inundation levels denoted by ${}_k\mathbf{x}_t^0 \in \mathbb{R}^{D \times D}$, where $D \in \{16, 64, 80, 96\}$ is the side length of the patch. We also used digital elevation data (OpenTopography 2021) for the same area. We resampled the elevation data to the same resolution and then aligned it with our inundation data. It serves as spatial context within DIFF-SPARSE.

We have used 81 days of data for training, 1 day of data for validation, and 7 days of data for testing. Training data points are sampled by following the process described in Section 'DIFF-SPARSE Method (Training)'. We use a context window of 12 hours. During testing, we predict the inundation levels in the subsequent 12 hours. Inundation values are standardized based on the mean and standard deviation of the inundation values in the training data. Elevation values are standardized separately based on the mean and standard deviation of the elevation values in the study area.

Evaluation Metric

We use two metrics to evaluate forecast quality. The first metric is Normalized Root Mean Squared Error (NRMSE). Let $\mathbf{x}$ be the set of all observations over all cells for all timesteps. Let $x_{ij,t} \in \mathbf{x}$ denote the observed inundation value of cell (i, j) at time t and let, $\tilde{x}_{ij,t}$ denote the corresponding average forecast value. Then:

$$\text{NRMSE} = \frac{1}{\max_{ij,t}(\mathbf{x}) - \min_{ij,t}(\mathbf{x})} \sqrt{\frac{\sum_{ij,t} (x_{ij,t} - \tilde{x}_{ij,t})^2}{\sum_{ij,t} 1}} \tag{15}$$

While NRMSE captures the mean error between the forecasts and the ground truth, it cannot take into account the uncertainty of the prediction. To capture this, we use an extended version of the Continuous Ranked Probability Score (CRPS) (Winkler et al. 1996). The CRPS between a single observation $x_{ij,t}$ and the empirical CDF $\hat{F}_{ij,t}^M$ of M corresponding forecasts, denoted as $\text{CRPS}(\hat{F}_{ij,t}^M, x_{ij,t})$, is extended for the multivariate case as Normalized Average CRPS (NACRPS).

$$\text{NACRPS}(\hat{F}^M, \mathbf{x}) = \frac{\sum_{ij,t} \text{CRPS}(\hat{F}_{ij,t}^M, x_{ij,t})}{\sum_{ij,t} |x_{ij,t}|} \tag{16}$$

Here, $|x_{ij,t}|$ denotes the absolute value of $x_{ij,t}$.

Baseline Comparison

We now describe our baseline methods. All our experiments were conducted on an HPC cluster using the SLURM scheduler. Each job used up to 256GB CPU RAM, 8 CPU cores and one GPU with maximum 40 GB GPU RAM.

Patch $D^2 \times K$	Metric	DiffSTG	DCRNN	DeepVAR	LSTNet	TimeGrad	BayesNF	DIFF-SPARSE
$16^2 \times 2$	NACRPS	1.2375± 0.0473	N/A	2.9912± 0.09	N/A	1.1233± 0.1139	**0.714**±0.062 (↑27.28%)	<u>0.9819</u>± 0.1248
	NRMSE	<u>0.1295</u>± 0.0018	0.1643± 0.0003	0.2564± 0.0061	0.1454± 0	0.1504± 0.0082	**0.1023**±0.006 (↑22.38%)	0.1318± 0.0036
$16^2 \times 5$	NACRPS	1.2959± 0.0398	N/A	2.4144± 0.0462	N/A	0.9952± 0.0347	**0.707**± 0.0661 (↑24.44%)	<u>0.9357</u>± 0.2
	NRMSE	0.1359± 0.0026	0.2026± 0.0113	0.2329± 0.0042	0.1515± 0	0.1477± 0.0019	**0.1088**± 0.0073 (↑13.44%)	<u>0.1257</u>± 0.0157
$64^2 \times 10$	NACRPS	Failed	N/A	0.967± 0.0124	N/A	0.5741± 0.0249	<u>0.2668</u>± 0.0136	**0.2028**± 0.0446 (↑23.99%)
	NRMSE		0.2735± 0.0004	0.2907± 0.0021	0.2696± 0.0009	0.1845± 0.0039	<u>0.097</u>± 0.0047	**0.069**± 0.016 (↑28.87%)
$64^2 \times 20$	NACRPS	Failed	Failed	0.9165± 0.0125	N/A	0.6221± 0.0199	<u>0.2826</u>± 0.0167	**0.2622**± 0.0593 (↑7.22%)
	NRMSE			0.2927± 0.0021	0.2735± 0.0009	0.2064± 0.0043	<u>0.113</u>± 0.0057	**0.0924**± 0.0192 (↑18.23%)
$80^2 \times 20$	NACRPS	Failed	Failed	0.9028± 0.0123	N/A	<u>0.6195</u>± 0.0271	Failed	**0.2377**± 0.0419 (↑61.63%)
	NRMSE			0.1994± 0.0013	0.1866± 0.0011	<u>0.1387</u>± 0.0035		**0.0559**± 0.0104 (↑59.7%)
$96^2 \times 20$	NACRPS	Failed	Failed	0.9487± 0.0182	N/A	<u>0.6641</u>± 0.0353	Failed	**0.2645**± 0.0574 (↑60.17%)
	NRMSE			0.1091± 0.0008	0.103± 0.0002	<u>0.0781</u>± 0.0019		**0.0312**± 0.0048 (↑60.05%)

Table 1: Performance Comparison of DIFF-SPARSE with baseline models at 95% sparsity level (i.e., only 5% of cells in a patch have sensors). All metrics are reported up to one standard deviation after conducting ten experiments with different random seeds. For each configuration, the best performance is bold-highlighted and the second-best performance is underlined. For DIFF-SPARSE, percentage improvement in performance with its closest baseline competitor is shown. State-of-the-art spatiotemporal forecasting methods (e.g., DiffSTG, DCRNN) could not handle training for large patch configurations; these are marked as 'Failed'. Since, DCRNN and LSTNet are point forecasting methods, their NACRPS metrics are marked as 'N/A'.

1. DiffSTG (Wen et al. 2023): The State-of-the-art (SOTA) diffusion-based model for spatiotemporal forecasting.
2. TimeGrad (Rasul et al. 2021) The SOTA diffusion-based model designed for multivariate time-series forecasting.
3. DeepVAR (Salinas et al. 2019): Combines an RNN-based architecture with a Gaussian copula process output model with a low-rank covariance structure.
4. LSTNet (Lai et al. 2018): Captures local dependency pattern among variables and long-term temporal correlation among time-series trends using CNN and RNN.
5. DCRNN (Li et al. 2017): A spatiotemporal forecasting framework that combines diffusion convolution and sequence-to-sequence architecture to capture temporal and spatial dependencies.
6. BayesNF (Saad et al. 2024): SOTA DL method based on a Bayesian Neural Network that maps a multivariate space-time coordinate to a real-valued field.

We used six different patch settings in our experiments as shown in Table 1. The experiments are performed at a sparsity level of 95%; meaning when forecasting for a patch, only 5% of the cells of that patch will have sensors and inundation values in the context window. During training, we generate random sensor masks ($\mathbf{M}_k$) that have $\sim 5\%$ of the cell values to be 1. For testing, we generate 10 different sensor masks at random with same (95%) sparsity level for each patch setting. These masks are applied on test data points (from Tidewatch) in a round-robin manner; i.e., mask 1 applied on data point 1, mask 2 applied on data point 2, ..., mask 10 applied on datapoint 10, mask 1 applied on datapoint 11, and so on. This ensures that all forecasting methods are being tested with same datapoints. By masking the test datapoints and then making forecasts based on them, we simulate the real-world scenario of forecasting based on sparse sensor observations.

It should to be noted that the number of variables increases quadratically with patch size and linearly with the number of patches. We found that some models do not scale well with increments in the number of variables. Across all experiments, we set the context and prediction length to 12 (experiment results with varying prediction length are provided in the extended version of the paper). Eight scenarios were sampled to understand the quality of probabilistic forecasting. Table 1 summarizes the performance of DIFF-SPARSE with respect to the baseline methods. Some key observations from the results are as follows:

First, we observe that DiffSTG, DCRNN and BayesNF scale poorly to large patch configurations. DiffSTG was unable to process patch configurations where $D > 16$. For DCRNN, $(D = 64, K = 10)$ was the highest patch configuration that was processed successfully, whereas BayesNF fails for $D > 64$. In larger settings, we ran out of GPU memory for these methods. It demonstrates the relatively worse computational scalability of these three methods.

Second, in small patch configurations, specifically $D = 16$ and $K \in \{2, 5\}$, DIFF-SPARSE performs second best. Here, BayesNF is the superior model. DIFF-SPARSE outperforms the other baselines in these configurations.

Third, in moderate and large patch configurations ($D \in \{64, 80, 96\}$, $K \in \{10, 20\}$), DIFF-SPARSE outperforms all six baseline methods in terms of both metrics. Across moderate configuration where $D = 64$, which is the largest configuration that BayesNF scales to, DIFF-SPARSE outperforms BayesNF by a margin of $\sim$ 7-24% and $\sim$ 18-29% in terms of NACRPS and NRMSE, respectively. In larger configurations ($D \in \{80, 96\}, K = 20$), DIFF-SPARSE achieves 59-61% improvement over the second-best method (TimeGrad), in terms of the two performance metrics. This illustrates the superiority of DIFF-SPARSE over baseline methods in performance scalability.

These experiment results highlight that existing spatiotemporal forecasting methods have either poor computational scalability or performance scalability, making it challenging to apply them to large-scale high-resolution spatiotemporal forecasting tasks. DIFF-SPARSE offers both

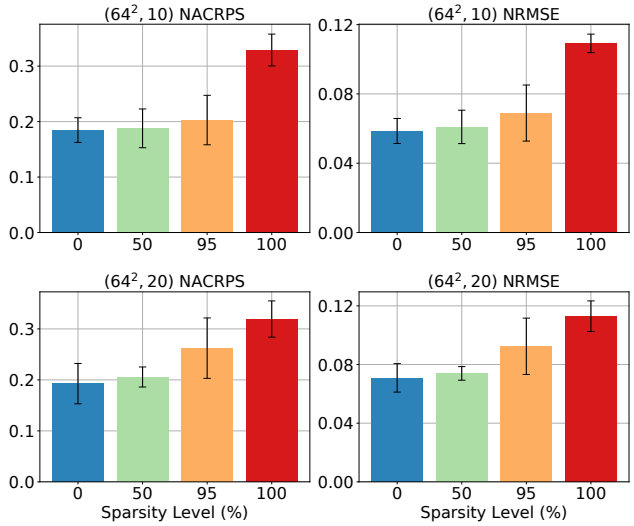

Figure 2: Bar-plots showing DIFF-SPARSE performance at varying sparsity levels in two patch configurations. Ten experiment runs were performed in each setting; mean and standard deviation of the performance metrics are shown.

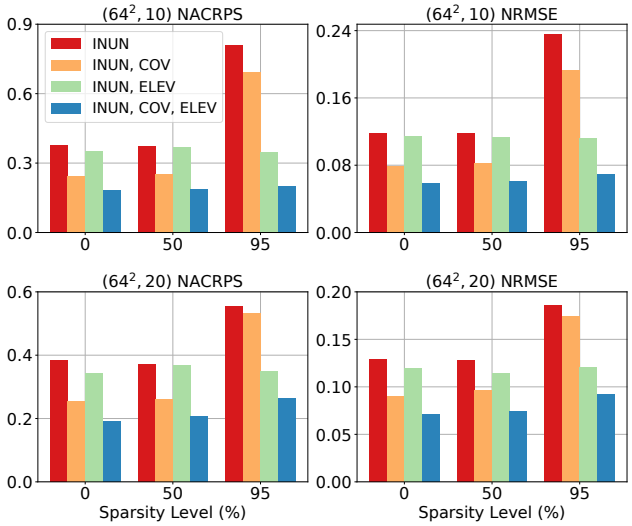

Figure 3: Ablation study at varying sparsity levels for two patch configurations. Ten experiment runs were performed in each setting; mean value of the performance metrics are shown using bar-plots.

computation and performance scalability, making it suitable for large-scale high-resolution spatiotemporal forecasting tasks under sparse observations.

Ablation Study

In this section, we perform ablation studies on DIFF-SPARSE to understand the importance of different context data. First, we investigate the predictive performance of DIFF-SPARSE at various sparsity levels. We trained and tested DIFF-SPARSE at four different sparsity levels (0%, 50%,95%, and 100%) for the patch configurations $(64^2, 10)$ and $(64^2, 20)$. Sparsity level of 100% means no sensor data is available. Figure 2 shows the performance of DIFF-SPARSE in these settings, in terms of our two performance metrics. We observe that in both patch configurations, DIFF-SPARSE has the best performance at 0% sparsity level in terms of both metrics. As the sparsity level increases, performance of DIFF-SPARSE degrades; the worst performance observed at 100% sparsity level. It implies that, having (more) sensors, i.e., observed inundation values in the context window, helps DIFF-SPARSE to make better predictions.

Next, we investigate the utility of elevation data and temporal co-variates as additional context. Given sparse inundation history (INUN: ${}_k\mathbf{y}^0_{t_0:t-1}$), elevation (ELEV: $\mathbf{s}_k$), and temporal co-variates (COV: $\mathbf{z}_{t_0:t-1}$), we examine four configurations of DIFF-SPARSE based on how the context embedding is computed: (1) INUN: only sparse inundation values used, (2) INUN, ELEV: sparse inundation values and elevation data used, (3) INUN, COV: sparse inundation values and temporal co-variates used, (4) INUN, ELEV, COV: all three components used. Figure 3 shows the performance of these four settings at three different sparsity levels (0%, 50%, and 95%) for the patch configurations $(64^2, 10)$ and $(64^2, 20)$. We see that in all sparsity levels, both (INUN, COV) and (INUN, ELEV) performs better than INUN; meaning both elevation data and temporal covariates are individually helpful in making better forecasts. At low sparsity levels (0%, 50%), the setting (INUN, COV) performs better than (INUN, ELEV), indicating the higher utility of temporal co-variates than elevation at low sparsity levels. However, at high sparsity level (95%), we see the opposite; (INUN, ELEV) performs better than (INUN, COV). It means at high sparsity level, elevation data becomes more useful compared to temporal co-variates. In all sparsity levels, using all three components (i.e., INUN, COV, ELEV) yields the best result.

Conclusion

In this paper, we have addressed the problem of high resolution probabilistic coastal inundation forecasting from sparse sensor observations. To solve it, we first formulated it as a problem of learning a conditional distribution function. We then presented DIFF-SPARSE, a scalable spatiotemporal forecasting method that utilizes sparse sensor observations, elevation, and temporal covariates to make probabilistic forecasts. DIFF-SPARSE is trained on data generated by a physics-based hydro-dynamical model. A novel masking strategy is employed to tackle the challenge of making forecasts based on sparse observations during inference. Moreover, due to our robust training strategy, DIFF-SPARSE can make forecasts based on different sensor placement configurations without requiring retraining from scratch. Through our experiments, we demonstrated that DIFF-SPARSE outperforms existing forecasting methods in terms of computational and performance scalability. Our extensive ablation study reveals that all three spatial and temporal context data: elevation data, temporal covariates, and sensor observations (even if sparse), help DIFF-SPARSE make better forecasts.

Acknowledgments

This work was partially supported by the following grants: (i) NSF Grants CCF-1918656, OAC-1916805, RISE-2053013; (ii) AI Institute: Agricultural AI for Transforming Workforce and Decision Support (AgAID) award No. 2021-67021-35344; (iii) U.S. Department of Energy, through the Office of Advanced Scientific Computing Research's "Data-Driven Decision Control for Complex Systems (DnC2S)" project. Pacific Northwest National Laboratory is operated by Battelle Memorial Institute for the U.S. Department of Energy under Contract No. DE-AC05-76RL01830.

References

Ahmad, F. Y.; Venkataswamy, V.; and Fox, G. 2024. CaloBench: A Benchmark Study of Generative Models for Calorimeter Showers. In *International Symposium on Benchmarking, Measuring and Optimization*, 70–95. Springer.

Amram, O.; and Pedro, K. 2023. Denoising diffusion models with geometry adaptation for high fidelity calorimeter simulation. *Physical Review D*, 108(7): 072014.

Austin, J.; Johnson, D. D.; Ho, J.; Tarlow, D.; and Van Den Berg, R. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34: 17981–17993.

Barrenetxea, G.; Ingelrest, F.; Lu, Y. M.; and Vetterli, M. 2008. Assessing the challenges of environmental signal processing through the SensorScope project. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 5149–5152. IEEE.

Biloš, M.; Rasul, K.; Schneider, A.; Nevmyvaka, Y.; and Günnemann, S. 2023. Modeling temporal data as continuous functions with stochastic process diffusion. In *International Conference on Machine Learning*, 2452–2470. PMLR.

Chang, P.; Li, H.; Quan, S. F.; Lu, S.; Wung, S.-F.; Roveda, J.; and Li, A. 2024. A transformer-based diffusion probabilistic model for heart rate and blood pressure forecasting in Intensive Care Unit. *Computer Methods and Programs in Biomedicine*, 246: 108060.

Deltares. 2014. SOBEK—Hydrodynamics, Rainfall Runoff and Real Time Control.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34: 8780–8794.

Ezer, T. 2020. Analysis of the changing patterns of seasonal flooding along the US East Coast. *Ocean Dynamics*, 70(2): 241–255.

Ezer, T.; and Atkinson, L. P. 2014. Accelerated flooding along the US East Coast: On the impact of sea-level rise, tides, storms, the Gulf Stream, and the North Atlantic Oscillations. *Earth's Future*, 2(8): 362–382.

Garg, S.; Singh, A.; and Ramos, F. 2012. Learning non-stationary space-time models for environmental monitoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, 288–294.

Guo, K.; et al. 2021. Urban surface water flood modelling–a comprehensive review of current models and future challenges. *Hydrology and Earth System Sciences*, 25(5): 2843–2860.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Islam, K. A.; Mehrab, Z.; Halappanavar, M.; Mortveit, H.; Katragadda, S.; Loftis, J. D.; Hoops, S.; and Marathe, M. 2025. Towards High Resolution Probabilistic Coastal Inundation Forecasting from Sparse Observations.

Jacobs, J. M.; Cattaneo, L. R.; Sweet, W.; and Mansfield, T. 2018. Recent and future outlooks for nuisance flooding impacts on roadways on the US East Coast. *Transportation research record*, 2672(2): 1–10.

Karagulian, F.; Barbiere, M.; Kotsev, A.; Spinelle, L.; Gerboles, M.; Lagler, F.; Redon, N.; Crunaire, S.; and Borowiak, A. 2019. Review of the performance of low-cost sensors for air quality monitoring. *Atmosphere*, 10(9): 506.

Kingma, D. P.; Salimans, T.; Jozefowicz, R.; Chen, X.; Sutskever, I.; and Welling, M. 2016. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29.

Krause, A.; Singh, A.; and Guestrin, C. 2008. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(2).

Lai, G.; Chang, W.-C.; Yang, Y.; and Liu, H. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, 95–104.

Li, Y.; Lu, X.; Wang, Y.; and Dou, D. 2022. Generative time series forecasting with diffusion, denoise, and disentanglement. *Advances in Neural Information Processing Systems*, 35: 23009–23022.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

Loftis, J. D. 2022. Exploring Latent Verification Methods for Inundation Forecasting Models through Remote Sensing Networks and Community Science. In *OCEANS 2022, Hampton Roads*. IEEE.

Loftis, J. D.; Mitchell, M.; Schatt, D.; Forrest, D. R.; Wang, H. V.; Mayfield, D.; and Stiles, W. A. 2019. Validating an Operational Flood Forecast Model Using Citizen Science in Hampton Roads, VA, USA. *Journal of Marine Science and Engineering*, 7(8): 242.

Luo, C. 2022. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*.

Mydlarz, C.; Sai Venkat Challagonda, P.; Steers, B.; Rucker, J.; Brain, T.; Branco, B.; Burnett, H. E.; Kaur, A.; Fischman, R.; Graziano, K.; et al. 2024. FloodNet: Low-Cost Ultrasonic Sensors for Real-Time Measurement of Hyperlocal, Street-Level Floods in New York City. *Water Resources Research*, 60(5): e2023WR036806.

OpenTopography. 2021. USGS 1/3 arc-second Digital Elevation Models.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2): 3.

Rasul, K.; Seward, C.; Schuster, I.; and Vollgraf, R. 2021. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, 8857–8868. PMLR.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10684–10695.

Roy, B.; Goodall, J. L.; McSpadden, D.; Goldenberg, S.; and Schram, M. 2023. Application of LSTM and seq2seq LSTM surrogate models for forecasting multi-step-ahead nuisance flooding of flood-vulnerable streets in Norfolk, Virginia. In *AGU Fall Meeting Abstracts*, volume 2023, H41V–02.

Saad, F.; Burnim, J.; Carroll, C.; Patton, B.; Köster, U.; A. Saurous, R.; and Hoffman, M. 2024. Scalable spatiotemporal prediction with Bayesian neural fields. *Nature Communications*, 15(1): 7942.

Salinas, D.; Bohlke-Schneider, M.; Callot, L.; Medico, R.; and Gasthaus, J. 2019. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *Advances in neural information processing systems*, 32.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, 2256–2265. PMLR.

Sønderby, C. K.; Raiko, T.; Maaløe, L.; Sønderby, S. K.; and Winther, O. 2016. Ladder variational autoencoders. *Advances in neural information processing systems*, 29.

Swain, D.; Wing, O. E.; Bates, P. D.; Done, J.; Johnson, K.; and Cameron, D. 2020. Increased flood exposure due to climate change and population growth in the United States. *Earth's Future*, 8(11): e2020EF001778.

Syme, W. 2001. TUFLOW-Two & Onedimensional unsteady flow Software for rivers, estuaries and coastal waters. In *IEAust Water Panel Seminar and Workshop on 2d Flood Modelling, Sydney*.

Tao, Y.; Tian, B.; Adhikari, B. R.; Zuo, Q.; Luo, X.; and Di, B. 2024. A Review of Cutting-Edge Sensor Technologies for Improved Flood Monitoring and Damage Assessment. *Sensors*, 24(21): 7090.

Tashiro, Y.; Song, J.; Song, Y.; and Ermon, S. 2021. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34: 24804–24816.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Vu, T. T.; Nguyen, P. K.; Chua, L. H.; and Law, A. W. 2015. Two-dimensional hydrodynamic modelling of flood inundation for a part of the Mekong River with TELEMAC-2D. *British Journal of Environment and Climate Change*, 5(2): 162–175.

Wen, H.; Lin, Y.; Xia, Y.; Wan, H.; Wen, Q.; Zimmermann, R.; and Liang, Y. 2023. Diffstg: Probabilistic spatiotemporal graph forecasting with denoising diffusion models. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, 1–12.

Winkler, R. L.; Munoz, J.; Cervera, J. L.; Bernardo, J. M.; Blattenberger, G.; Kadane, J. B.; Lindley, D. V.; Murphy, A. H.; Oliver, R. M.; and Ríos-Insua, D. 1996. Scoring rules and the evaluation of probabilities. *Test*, 5: 1–60.

Wuebbles, D.; Fahey, D.; Takle, E.; Hibbard, K.; Arnold, J.; DeAngelo, B.; Doherty, S.; Easterling, D.; Edmonds, J.; Edmonds, T.; et al. 2017. Climate science special report: Fourth national climate assessment (NCA4), Volume I.

Yang, D.; Yu, J.; Wang, H.; Wang, W.; Weng, C.; Zou, Y.; and Yu, D. 2023. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31: 1720–1733.

Yang, K.; Davidson, R. A.; Vergara, H.; Kolar, R. L.; Dresback, K. M.; Colle, B. A.; Blanton, B.; Wachtendorf, T.; Trivedi, J.; and Nozick, L. K. 2019. Incorporating inland flooding into hurricane evacuation decision support modeling. *Natural Hazards*, 96: 857–878.

Yang, R.; Srivastava, P.; and Mandt, S. 2023. Diffusion probabilistic modeling for video generation. *Entropy*, 25(10): 1469.

Zahura, F. T.; and Goodall, J. L. 2022. Predicting combined tidal and pluvial flood inundation using a machine learning surrogate model. *Journal of Hydrology: Regional Studies*, 41: 101087.

Zanchetta, A.; and Coulibaly, P. 2022. Hybrid Surrogate Model for Timely Prediction of Flash Flood Inundation Maps Caused by Rapid River Overflow, Forecasting, 4, 126–148.

Zhang, L.; et al. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3836–3847.

Zhang, Y. J.; Ye, F.; Stanev, E. V.; and Grashorn, S. 2016. Seamless cross-scale modeling with SCHISM. *Ocean Modelling*, 102: 64–81.