Streamflow Prediction with Uncertainty Quantification for Water Management: A Constrained Reasoning and Learning Approach

Mohammed Amine Gharsallaoui , **Bhupinderjeet Singh** , **Supriya Savalkar** , **Aryan Deshwal** , **Ananth Kalyanaraman** , **Kirti Rajagopalan** and **Janardhan Rao Doppa**

Washington State University, Pullman, WA, USA

{m.gharsallaoui, bhupinderjeet.singh, supriya.savalkar, aryan.deshwal, ananth, kirtir, jana.doppa}@wsu.edu

Abstract

Predicting the spatiotemporal variation in streamflow along with uncertainty quantification enables decision-making for sustainable management of scarce water resources. Process-based hydrological models (aka physics-based models) are based on physical laws, but using simplifying assumptions which can lead to poor accuracy. Data-driven approaches offer a powerful alternative, but they require large amount of training data and tend to produce predictions that are inconsistent with physical laws. This paper studies a *constrained reasoning and learning (CRL) approach* where physical laws represented as logical constraints are integrated as a layer in the deep neural network. To address small data setting, we develop a theoretically-grounded training approach to improve the generalization accuracy of deep models. For uncertainty quantification, we combine the synergistic strengths of Gaussian processes (GPs) and deep temporal models (i.e., deep models for time-series forecasting) by passing the learned latent representation as input to a standard distance-based kernel. Experiments on multiple real-world datasets demonstrate the effectiveness of both CRL and GP with deep kernel approaches over strong baseline methods.

1 Introduction

Streamflow is fundamental to meeting societal needs including food and energy security, and environmental health. In "fully appropriated" Western US watersheds—where every drop of water is associated with a beneficial use under a legal water rights system—an accurate representation of the spatiotemporal variation in streamflow is essential for sustainable management of scarce water resources. A typical decision context in the snow-dominant Western US is reservoir operations leading into the spring and summer months when most of the streamflow is generated (*high streamflow months* from March through July which is the focus of our study).

The reservoirs need to be operated to manage competing objectives based on expectations of upcoming streamflow. Typical objectives include meeting flood control, irrigation demands for the season, and hydro power generation needs

(see Fig. 1). One concrete example of the conflicting objectives is as follows. The reservoirs need to be evacuated enough to capture upcoming streamflow and avoid flood risk. However, this cannot be at the expense of not being able to fill the reservoir before the irrigation season starts. Having a good forecast of expected flows in high flow months, and reliable uncertainty bounds around it can help optimize storage and release decisions to meet all objectives. Moreover, different decisions will involve utilizing uncertainty bounds in different ways. When flood risk needs to be minimized, the reservoir manager will likely utilize flows at the upper bound of the uncertainty interval in water release decisions. In contrast, while focusing on meeting irrigators' needs, the reservoir manager will likely make storage decisions based on the lower bound of expected streamflow. Under the auspices of a federal grant, this project team has been interacting with United States Bureau of Reclamation (USBR)—the federal agency tasked with managing reservoir operations for all reservoirs in the Western United States where water for irrigation is a key operational priority along with a multitude of other uses (see Fig. 1). Therefore, this work has a real potential for impact at the U.S. national scale.

Process-based hydrological models (aka physics-based models) that take weather, soil, and other inputs to translate precipitation to streamflow using biophysical representations of the hydrological process are commonly used to estimate streamflow. These models have been known to yield best results when calibrated to individual watersheds but they continue to struggle in regional-scale modeling across multiple watersheds because hydrological response is heterogeneous and watershed-specific [Kratzert *et al.*, 2019].

Data-driven (aka deep learning) models can capture this heterogeneity and push the frontiers in learning universal, regional, and local hydrological behavior [Kratzert *et al.*, 2019; Peters-Lidard *et al.*, 2017; Karpatne *et al.*, 2016] while increasing streamflow prediction accuracy. However, this learning methodology has to maintain the basic laws governing hydrological processes (e.g., conservation of mass for water balance) and capture the essence of important intermediary processes (e.g., evapotranspiration) that translate precipitation to streamflow. Since regional-scale water management decisions are made on "seasonal" time scales, we seek to make streamflow prediction for high-flow months to allow a targeted impact. As a result, we lack large amounts of train-
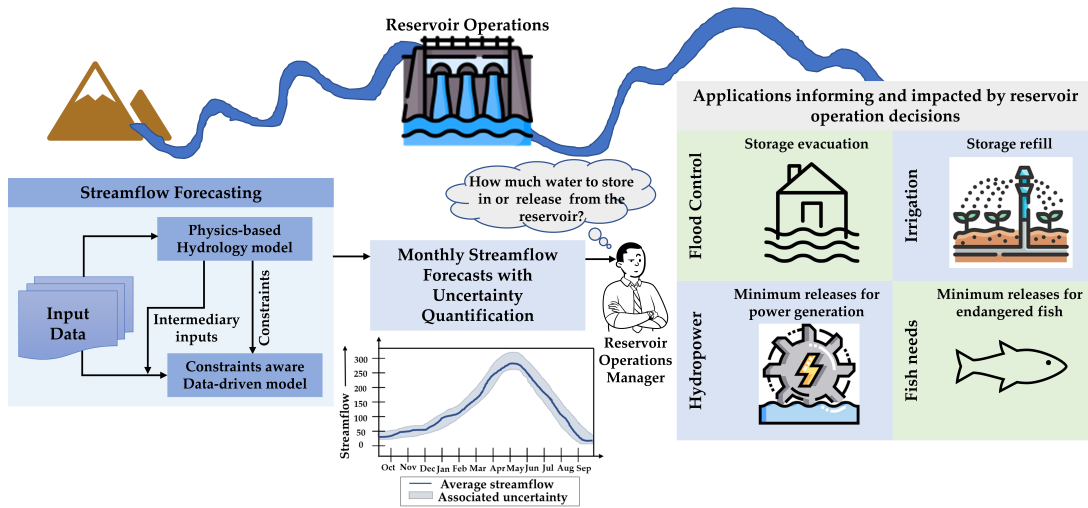
Figure 1: Overview of our overall approach for streamflow forecasting to enable water management decisions.

ing data required by the deep learning methods—e.g., even with 25 years of monthly data, there are only 125 samples.

Furthermore, in watersheds with competing water uses, when streamflow predictions go wrong, their societal impacts are significant—including losses for agriculture and the rural economy given less-than-ideal drought management; and negative impacts on endangered fish species due to higher/lower flows than expected, e.g., the 2015 drought year resulted in ∼$0.7 billion agricultural revenue loss in the U.S. Washington State [McLain *et al.*, 2015]. Therefore, quantifying uncertainty under which water managers make these decisions is critical for public trust, conflict resolution, and informed decision making.

Challenges. There are three key technical challenges: **1)** Predicting outputs which are consistent with the physical laws; **2)** Training predictive models from small data which generalize well to unseen data; and **3)** Quantifying the uncertainty in terms of a prediction interval such that the ground-truth output lies in this interval in most cases.

Contributions. This paper's key contribution is a principled approach for streamflow prediction and uncertainty quantification to address the above-mentioned three challenges.

Prior work has attempted to address the first challenge using physics-guided training or physics-informed neural networks [Karpatne *et al.*, 2017a; Cai *et al.*, 2021] where the key idea is to penalize the deep model when the predictions are not consistent with the physical laws. This approach was shown to improve over the pure data-driven approach on predicting depth-specific lake water temperatures [Jia *et al.*, 2021; Jia *et al.*, 2019; Karpatne *et al.*, 2017b] and other use-cases [Muralidhar *et al.*, 2020]. However, this approach and similar methods [Kamruzzaman *et al.*, 2021; Hoedt *et al.*, 2021; Sit *et al.*, 2021; Xiang and Demir, 2022] do not guarantee that *every* test-time prediction is consistent with the domain knowledge. To overcome this drawback, as our first contribution, *we study a new constrained reasoning and learning (CRL) approach using a semantic probabilistic*

layer (SPL) [Ahmed *et al.*, 2022b]. SPL compiles the domain knowledge specified in the form of logical constraints into a tractable circuit representation [Vergari *et al.*, 2020] and adds it as a replacement for predictive layers in deep networks for efficient end-to-end training. We empirically show that CRL approach with SPL instantiation uses the domain knowledge to improve the predictive accuracy over baseline methods.

To address the small data challenge, *we propose a novel importance-weighted (IW) training approach to improve the generalization accuracy of predictive models learned over the latent representation from the deep temporal model.* The key principle is to assign importance weights directly proportional to the errors on data samples during the training process, and to compute the weighted gradients using the data samples for gradient-based training. We provide theoretical analysis to justify the IW training approach. Our experiments show that IW training further improves the predictive accuracy over the standard training. Our IW algorithm and its theoretical analysis is specific to the small data challenge unlike prior work [Xu *et al.*, 2021].

For uncertainty quantification (UQ), *we study a principled approach by combining the synergistic strengths of Gaussian processes (GPs) and deep temporal models.* Deep neural networks are known to extrapolate over-confidently [Valentin Jospin *et al.*, 2020]. On the other hand, GP models are gold standard [Williams and Rasmussen, 2006] for UQ providing principled Bayesian uncertainty estimates, unlike existing ad-hoc techniques [Liu *et al.*, 2023], but have difficulty scaling to high-dimensional structured inputs (e.g., time-series). Deep kernel learning (DKL) [Wilson *et al.*, 2016] is a recent promising technique towards handling this drawback while maintaining the good UQ properties of a GP. Our experiments demonstrate that GP based approach with deep kernel produces prediction intervals with high coverage (i.e., ground truth output lies within the interval). We make our code and datasets available to the AI community at https://github.com/aminegha/StreamPred to catalyze research in this important problem space.

2 Background and Problem Setup

In streamflow prediction, utilizing graph representations is an efficient method for illustrating watersheds. Within a defined geographical expanse, a watershed represents the entirety of land that contributes water to a shared outlet, which might be a river, lake, or ocean. Within the watershed, a river network showcases the flow paths and connections of water in that region. A watershed is subdivided into grids, representing discrete small sections that facilitate the analysis and modeling of factors such as water movement. In the graph representation, the *nodes* represent the grids within the watershed, and the *edges* represent the flows between them. Each node has a set of *environmental* features (including variables related to precipitation and temperature). Given this rich graph representation, our objective is to predict the streamflow at the watershed outlet at fixed time intervals. Since water management decisions are made seasonally, *we aim to make streamflow predictions for high-flow months (March to July).*

Domain Knowledge. The water balance equation (WBE) is a fundamental concept in hydrology: the amount of water in a system should be balanced between the inputs (e.g., precipitation), outputs (e.g., evapotranspiration, streamflow), and changes in storage (e.g., soil moisture or snowpack). It is given by $P \geq ET + S$, where P is precipitation, ET is evapotranspiration, and S is streamflow. WBE is not an equality because all components of the water balance are not present here and holds only at an annual aggregated level.

Problem Setup. For a given watershed, we consider a sequence of graphs over discrete time steps denoted as $G_1, G_2, \cdots, G_T$, where each graph G_t represents a directed graph with a set of nodes V and a set of edges $E \subseteq V \times V$. At each time step, the corresponding features matrix $X_t \in \mathbb{R}^{n \times m}$, where $n = |V|$ and m is the number of features. The graph G_t is also characterized by an adjacency matrix $A \in \mathbb{R}^{n \times n}$, with the property that A_t remains invariant since grid connections are static. At each (monthly) time step t, we also have the target variable y_t which is the streamflow at the watershed *outlet* node. Given a training data of time-series $\mathcal{D} = \{(X_t, y_t)\}_{t=1}^T$ and domain knowledge $\mathcal{K}$ in the form of water balance equation, we have two goals:

a) To learn a predictor F that can make accurate predictions consistent with the knowledge $\mathcal{K}$ for unseen inputs $X_{t'}$ where $t' > T$ (*high-flow months* only).

b) To get uncertainty estimates in the form of a prediction interval $[lb, ub]$ for $X_{t'}$, where the true output $y_{t'} \in [lb, ub]$ with a high probability.

3 Technical Approach

In this section, we outline our overall methodology shown in Algorithm 1 to address the three technical challenges, namely, predictions consistent with domain knowledge, small training data, and uncertainty quantification.

3.1 Constrained Reasoning and Learning

Recurrent Convolutional Graph Neural Network. We employ a recurrent convolutional graph convolutional neural network (RCGNN) [Seo *et al.*, 2016; Sit *et al.*, 2021;

Xiang and Demir, 2022] to solve the streamflow prediction task. RCGNN is a generalization of classical graph convolutional network that incorporates the recurrent nature of RNN to capture temporal dependencies in graph-structured data. Notably, in graph embedding learning, for a given node, we integrate information from both preceding time steps and adjacent river segments (i.e., edges). The RCGNN model establishes transition connections for the derived latent representation using a recurrent cell configuration. In the scope of this study, we employ the Gated Recurrent Unit (GRU) cell, known for its prowess in capturing sequential patterns and dependencies in graph-structured data. For time step t, given a graph represented by its adjacency matrix A and its input feature matrix X, a graph convolutional GRU layer is parameterized with multiple weight matrices W's, U's and bias terms b's, and starts by computing the reset gate r_t and updates gate z_t from time step $t - 1$:

$$r_t = \sigma(W_r \cdot X + U_r \cdot (A \cdot H^{(t-1)}) + b_r)$$
$$z_t = \sigma(W_z \cdot X + U_z \cdot (A \cdot H^{(t-1)}) + b_z),$$

where $H^{(t-1)}$ is the hidden state of the previous time step t, while σ denotes the sigmoid activation function. Next, it computes candidate hidden state (h_t'):

$$h_t' = \tanh(W_h \cdot X + U_h \cdot (r_t \odot (A \cdot H^{(t-1)})) + b_h).$$

Here, $\odot$ represents element-wise multiplication. Using z_t and h_t', we update the current hidden state $H^{(t)}$ as follows:

$$H^{(t)} = z_t \odot H^{(t-1)} + (1 - z_t) \odot h_t'.$$

Therefore, the learnable weight matrices of the model are specifically denoted above by W_r, U_r, W_z, U_z, W_h and U_h, and trainable bias terms are b_r, b_z and b_h. For the subsequent output layers, two distinct activation functions are employed alongside two linear layers. For the activation function, we use LeakyRelu to introduce non-linearity. We build our RCGNN model to learn the graph embedding according to the above architecture, denoted by $f(X)$.

Semantic Probabilistic Layer. To address the challenge of maintaining domain knowledge consistency, the constrained reasoning and learning (CRL) approach instantiates the concept of semantic probabilistic layer (SPL) [Ahmed *et al.*, 2022b]. SPL can be replaced with the predictive layer of any deep neural network. It allows to model domain knowledge as logical constraints over multiple inter-related output variables and guarantees consistency of predicted outputs.

Given a structured input X, the computation of the probability of a candidate structured output Y is decomposed as $p(Y|f(X)) = q_\Theta(Y|f(X)) \cdot c_{\mathcal{K}}(X, Y)/\mathcal{Z}(X)$, where $f(X)$ represents the feature embedding or latent representation of input X; $q_\Theta(Y|f(X))$ parameterized by Θ allows us to perform probabilistic reasoning over candidate structured outputs Y; $c_{\mathcal{K}}(X, Y)$ is a constraint reasoner to ensure that predicted structured outputs Y are consistent with the domain knowledge $\mathcal{K}$ and produces evaluation 1 (i.e., $c_{\mathcal{K}}(X, Y)$=1) only when Y satisfies the declarative knowledge $\mathcal{K}$; and $\mathcal{Z}(x)$ is a normalization constant. SPL leverages the beneficial properties of probabilistic circuits [Vergari *et al.*, 2020;

Algorithm 1 Overall Training Algorithm

Input: Training data of graph time-series $\mathcal{D} = \{(X_t, y_t)\}_{t=1}^T$; domain knowledge $\mathcal{K}$; Recurrent GNN model $f_{\theta_{f_e}}$; Predictive model F_{θ_p}; GP kernel κ
Output: Trained feature extractor $f_{\theta_{f_e}}$, predictive model F_{θ_p}, and GP model with hyper-parameters of κ

1: Train the recurrent GNN model $f_{\theta_{f_e}}$ on $\mathcal{D}$ with SPL layer for domain knowledge $\mathcal{K}$ to create latent representations $z_i = f(X_i)$ // Section 3.1
2: *Train the predictive regression model F_{θ_p} using $\mathcal{D}_{tr} = \{z_i, y_i\}_{i=1}^T$ with T regression examples using the importance weighted training approach* // Section 3.2
3: Train the GP model on $\mathcal{D}_{tr} = \{z_i, y_i\}_{i=1}^T$ to optimize the hyper-parameters of the kernel κ *// Section 3.3*
4: ***return*** *feature extractor, predictor, and GP model*

Choi *et al.*, 2020] for tractable learning and reasoning. It encodes $q_\Theta(Y|f(X))$ and $c_\mathcal{K}(X,Y)$ as circuits, and considers constrained computational graphs to achieve tractability. Therefore, SPL allows us to produce consistent and correct structured outputs for every inference computation task, i.e., satisfies the domain knowledge in the form of water balance equation. We refer the reader to SPL paper [Ahmed *et al.*, 2022b] for additional details.

3.2 Importance Weighted Training

Given the learned latent representations $\{z_i = f(X_i)\}_{i=1}^T$ from the recurrent GNN model, we describe a novel importance weighted training approach to improve the generalization accuracy of the predictive model F.

Notations. Our goal is to train regression models over the training data $\mathcal{D}_{tr} = \{z_i, y_i\}_{i=1}^T$ with T examples. Suppose each data sample (z_i, y_i) is drawn from a target distribution $\mathcal{P}$ over the space $\mathcal{Z} \times \mathcal{Y}$ such that z_i is an input from the input space $\mathcal{Z}$ and $y_i \in \mathcal{Y}$ is the corresponding ground-truth output. Let $F : \mathcal{Z} \to \mathcal{Y}$ denote a predictive model trained on the training set $\mathcal{D}_{tr}$. A loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is used to measure the accuracy of predictions made by F. In this paper, we consider the residual $\ell(F(z), y) = |F(z) - y|$ as loss and assume that there exists an upper bound of loss as $L = \max_{(z,y) \in \mathcal{Z} \times \mathcal{Y}} \ell(F(x), y) < \infty$. Typically, F is expected to reach a small *true risk*, defined as $R(F) = \mathbb{E}_{(z,y) \sim \mathcal{P}}[\ell(F(z), y)]$ in the sense of population. Unfortunately, we cannot measure $R(F)$ due to the availability of only finite training samples drawn from $\mathcal{P}$.

Hence, the *empirical risk* $\widehat{R}(F) = \frac{1}{T} \sum_{i=1}^T \ell(F(z_i), y_i)$ on $\mathcal{D}_{tr}$ is employed to estimate it, leaving a standard estimation error bound in $O(1/\sqrt{T})$ (see Lemma 1). The rate of convergence of generalization error bound is particularly important when the number of training samples T is small. For example, to ensure the same $O(10^{-2})$ generalization error bound, $O(1/\sqrt{T})$ requires $T = O(10^4)$, while the faster rate $O(1/T)$ only needs $T = 10^2$, significantly reducing the requirement of large training sets from modern ML methods.

High-level Algorithm. Importance-weighted (IW) training is an iterative gradient-based approach. It assigns importance weight $\omega(z, y)$ to each sample $(z, y) \in \mathcal{D}_{tr}$ such that $\omega(z, y)$ is directly proportional to the loss $\ell(F(z), y)$ based on the current model F and employs the weighted gradient to update the model parameters of F in each iteration. Analogous to the empirical risk $\widehat{R}(F)$, we define the *IW empirical risk* with weights $\omega(z, y)$ for F as

$$\widehat{R}_\omega(F) = \frac{1}{T} \sum_{i=1}^T \omega(z_i, y_i) \cdot \ell(F(z_i), y_i). \tag{1}$$

$\widehat{R}_\omega(F)$ is general enough to include the empirical risk $R(F)$ as a special case: setting $\omega(z_i, y_i) = 1$ for all data samples reduces $\widehat{R}_\omega(F)$ to $\widehat{R}(F)$. In what follows, we show that under some configurations of importance weighting function ω, $\widehat{R}_\omega(F)$ gives a tighter estimation of the true risk $R(F)$.

Theoretical Analysis. The following lemma shows the generalization error bound for IW empirical risk $R_\omega(F)$.

Lemma 1. *(Error bound of IW risk, Theorem 1 in [Cortes et al., 2010]) Let $M = \sup_{z,y} \omega(z, y)$ denote the infinity norm of ω on the domain. For given F and $\delta > 0$, with probability at least $1 - \delta$, the following bound holds:*

$$R(F) - \widehat{R}_\omega(F) \leq \frac{2M \log(1/\delta)}{3T} + \sqrt{\frac{2d_2(\mathcal{P}||\frac{\mathcal{P}}{\omega}) \log(1/\delta)}{T}}, \tag{2}$$

where $d_2(\mathcal{P} \| \mathcal{Q}) = \int_x \mathcal{P}(x) \cdot \frac{\mathcal{P}(x)}{\mathcal{Q}(x)} dx$ is the base-2 exponential for Rényi divergence of order 2 between two distributions $\mathcal{P}$ and $\mathcal{Q}$ and T is the number of training samples.

In the above prior result, the last term on the right side ($O(\sqrt{d_2/T})$) typically dominates the first term ($O(M/T)$), i.e., $O(M/T) \ll O(\sqrt{d_2/T})$. Consequently, the choice of ω mainly changes the quantity of Rényi divergence d_2 and thus the dominating term $O(\sqrt{d_2/T})$. In what follows, we theoretically show that a simple principle to choose group-wise weights ω can easily shrink d_2 in the dominating term compared with the conventional empirical risk. Specifically, the grouping principle is based on the prediction loss $\ell(F(z), y)$ of F overall data samples. Given a total number of groups K, denote the group k by $\mathcal{D}_k$ such that any data from $\mathcal{D}_k$ suffers smaller loss than that from the next group $\mathcal{D}_{k+1}$, i.e., $\ell(F(z), y) \leq \ell(F(z'), y')$ for all $(z, y) \in \mathcal{D}_k$ and $(z', y') \in \mathcal{D}_{k+1}$ with $k \in \{1, ..., K - 1\}$ (see discussion in "**Practical Algorithm**"). Accordingly, the probability of drawing data from $\mathcal{D}_k$ is denoted by $\mathcal{P}_k = \mathbb{P}\{(z, y) \in \mathcal{D}_k\}$. Specifically, for the fixed F and the same analysis framework under Lemma 1, let $\widehat{B}$ and $\widehat{B}_\omega$ be the generalization error bounds of empirical risk and IW empirical risk, respectively, i.e., $R(F) - \widehat{R}(F) \leq \widehat{B}$ and $R(F) - \widehat{R}_\omega(F) \leq \widehat{B}_\omega$. Below we prove that $\widehat{B}_\omega < \widehat{B}$ under a certain group-wise setting for ω. Before proceeding to the main theorem, we define group-wise weights as $\{\omega_k\}_{k=1}^K$, where ω_k is assigned to data from group $\mathcal{D}_k$ as the weight shown by $\widehat{R}_\omega(F)$ in (1).

Theorem 1. *(IW Improves generalization bound) Assume $\frac{2M \log(1/\delta)}{3T} \leq \sqrt{\frac{2d_2(\mathcal{P}||\frac{\mathcal{P}}{\omega}) \log(1/\delta)}{T}}$ in (2) of Lemma 1. Set*

$\omega(z, y) = \omega_k$ for $(z, y) \in \mathcal{D}_k$ such that $\mathcal{P}_k \cdot \omega_k = \frac{k^a}{K^{b+1}}$ for any $k \in \{1, ..., K\}$. Under the conditions $0 < a \leq \min\{K - 1, b - 1/\ln(K)\}$, and $K \geq 2$, the IW empirical risk $\widehat{R}_\omega(F)$ gives a tighter generalization bound than empirical risk $\widehat{R}(F)$ with $\widehat{B}_\omega = \sqrt{\frac{2 \log(1/\delta)}{(a+1)T}}$ and $\widehat{B} = \sqrt{\frac{2 \log(1/\delta)}{T}}$.

The complete proof of Theorem 1 is in the Appendix.

Remark. Theorem 1 shows some mild conditions to achieve a tighter generalization bound by strategically setting the group-wise weights ω_k in IW method than empirical risk, i.e., $\widehat{B}_\omega < \widehat{B}$. Furthermore, it demonstrates how fast the hyper-parameter a reduces $\widehat{B}_\omega$, and thus builds the connection between a and the tightness of the bound. For example, setting $a = 10^2$ makes $\widehat{B}_\omega$ 10 times smaller than $\widehat{B}$.

Practical Algorithm. To devise a practical IW algorithm, we need to appropriately set the values of K (no. of groups) and ω_k (weights for each of the k groups). The key idea of setting the group-wise weight is to make $\omega_k \propto k/\mathcal{P}_k$ when we group data into $\{\mathcal{D}_k\}_{k=1}^K$ such that all data in $\mathcal{D}_k$ suffers smaller loss than that the data in $\mathcal{D}_{k+1}$. We propose range-wise weighting, i.e., grouping data into several bins (loss intervals) according to their loss values. For example, for $K = 2$, a threshold τ partitions all data into two bins, i.e., the group suffering smaller loss $\mathcal{D}_1 = \{(z, y) : \ell(z, y) \leq \tau\}$ and the other one suffering larger loss $\mathcal{D}_2 = \{(z, y) : \ell(z, y) > \tau\}$, with $\mathcal{P}_1$ and $\mathcal{P}_2$ denoting the percentage of data in the two groups. This strategy can be easily generalized to $K > 2$ by using multiple thresholds, such as uniformly grouping data into K ranges with interval $\frac{L}{K}$ for maximal loss L: $\mathcal{D}_k = \{(z, y) : \frac{L(k-1)}{K} \leq \ell(F(z), y) \leq \frac{Lk}{K}\}$ for $k \in \{1, \cdots, K\}$. In practice, typically more data samples suffer from smaller training loss, i.e., concentrating around the true output, which implies $\mathcal{D}_k$ with smaller group ID k leads to larger $\mathcal{P}_k$. Hence, setting $\omega_k \propto k/\mathcal{P}_k$ indicates that group $\mathcal{D}_k$ with smaller group ID k gets smaller weight ω_k, In contrast, larger weights ω_k are encouraged for $\mathcal{D}_k$ with larger group ID k. Training model with this IW configuration improves the generalization performance of F according to Theorem 1. In our implementation, we employed this strategy for assigning IWs with $K=10$.

3.3 Gaussian Process with Deep Kernels for UQ

Gaussian processes (GPs) [Williams and Rasmussen, 2006] are considered as gold standard for uncertainty quantification (UQ). GPs are non-parametric models and allows the practitioner to incorporate domain knowledge in the form of kernels $\kappa(x, x')$ to measure similarity between pairs of inputs x and x' (e.g., RBF kernel). GPs are Bayesian models whose posterior mean and standard deviation provide prediction and uncertainty for each input. A major drawback of GPs is that they have difficulty scaling to high-dimensional structured inputs (e.g., time-series of graphs) as in our problem setting. On the other hand, deep kernel learning (DKL) [Wilson et al., 2016] is a recent technique that has shown to be a promising solution towards handling this drawback. To synergistically combine the strengths of GPs and DKL, we parameterize the kernel function of the GP model with a neural network feature extractor that is passed to a canonical distance-aware kernel

such as RBF kernel. Specifically, we add a GP layer on top of the latent representations learned by the recurrent GNN model. Given the learned latent representations $z_i = f(X_i)$ for all $i=1$ to T, we create a GP using $\mathcal{D}_{\text{tr}} = \{z_i, y_i\}_{i=1}^T$ with T training examples. We employ the latent representations $z_i = f(X_i)$ to define a RBF kernel as shown below.

$$k(z, z') = \sigma_f e^{-||z-z'||_2^2/(2\gamma^2)}. \tag{3}$$

Training GPs refers to estimating the hyper-parameters of the GP kernel. For example, the RBF kernel in equation (3) has the length-scale γ and the signal variance σ_f hyperparameters. To learn the hyperparameters of the kernel , we maximize the marginal likelihood of the observed data using the training data $\mathcal{D}_{\text{tr}}$. Another advantage of GPs for UQ is that we can update the model online based on new training examples to improve the uncertainty estimates for future predictions.

The predictive uncertainty estimate for an input with latent representation z can be computed in closed form (for the noiseless observations setting) as given below:

$$\sigma^2(z) = k(z, z) - k_*^T \mathbf{K}^{-1} k_* \tag{4}$$

where $k_* = [k(z, z_1), k(z, z_2) \cdots k(z, z_T)]$ and $\mathbf{K}$ is the kernel matrix computed over the latent representation of training inputs. We can compute a GP driven confidence interval for input z as $[\mu(z) - 2\sigma(z), \mu(z) + 2\sigma(z)]$, where $\mu(z)$ is the posterior mean prediction of the GP.

4 Experiments and Results

In this section, we describe our empirical results and analysis.

4.1 Experimental Setup

Watershed	BR	C	CCR	F	SFC	Y
#grids	57	437	103	100	97	449

Table 1: Description of the watershed datasets. All watersheds were observed at a monthly rate from Oct. 1979 through Sep. 2014 (i.e., 420 months). Temporal features for each grid (graph node) includes incoming precipitation, max and min temperature, average temperature, wind speed, specific humidity, solar radiation, max and min relative humidity, and evapotranspiration. The environmental variables are obtained from a 4-km gridded dataset, regridded to the VIC-CropSyst model's resolution of 6-km via linear interpolation.

Watershed Datasets. We collected US Geological Survey's observed monthly streamflow at the outlet of six watersheds in the Columbia River basin of the Pacific Northwest US : Boise River (BR), Clearwater (C), Clearwater Canyon Ranger (CCR), Flathead (F), South Fork Clearwater (SFC), and Yakima (Y). Our intent was to primarily focus on "natural" watersheds with minimal human influence, and this criterion along with a constraint on the watershed size (minimum of 1800 sq. km) resulted in the first five watersheds. Additionally, we included the Yakima River basin as an example of human-influenced watershed knowing that the model performance will likely be poor. Table 1 provides a high-level summary. Each watershed has different number of grids (no.

of nodes in graph) and river network corresponds to the adjacency matrix of the graph. Each graph node has multiple time-varying (monthly) environmental features.

The watersheds encompass a span of water years, commonly used in hydrology to capture the hydrological cycles, starting in October and ending in September next year. We employed the following training/validation/testing splits. The first 20 years of data was used for training and the next 6 years for hyper-parameter tuning. We employ the last 9 years of data for testing the generalization of various methods.

Configuration of Predictive Algorithms. We compare our proposed approach with several baseline methods.

1) Physics-based model (VIC-CropSyst): We use VIC-CropSyst [Malek *et al.*, 2017; Rajagopalan *et al.*, 2018] which comprises two models—the VIC model, which is a semi-distributed macroscale hydrological model [Liang *et al.*, 1994], and the CropSyst model [Stöckle *et al.*, 2003].

2) Recurrent Convolutional GNN (RCGNN): We train the RCGNN model (Section 3.1) using PyTorch to optimize the Mean Squared Error (MSE) loss. It is a pure data-driven model without the knowledge of water balance equation.

3) Physics-guided RCGNN (PG-RCGNN): This method incorporates water balance equation (WBE) into the training of RCGNN following approach by [Karpatne *et al.*, 2017a]. It employs the combined loss $loss_{mse} + \lambda_{wbe} \cdot loss_{wbe}$, where the $loss_{wbe} = max(0, \hat{y} + ET - P)$ term penalizes predictions deviating from the water balance equation.

4) RCGNN with SPL layer (RCGNN-CRL): This approach employs constrained reasoning and learning (CRL) by adding a semantic probabilistic layer to incorporate the water balance equation in yearly aggregates. We use the Pylon software library [Ahmed *et al.*, 2022a] for this implementation.

5) RCGNN-CRL with Importance Weighted Training (RCGNN-CRL-IW): This is a variant of RCGNN-CRL with importance weighted training. We employ K=10 groups.

Configuration of UQ Algorithms. We compare our proposed Gaussian process approach with a dropout baseline.

1) Gaussian process w/ deep kernel learning (GP w/ DKL). We define a GP model by defining a RBF kernel over the latent representations $\{z_i = f(X_i)\}_{i=1}^{T}$ learned by the recurrent GNN model. We find the hyper-parameters of the RBF kernel by optimizing the negative log-likelihood of data.

2) Dropout method: We introduce two dropout layers after obtaining the latent representation to inject stochasticity into inference [Hinton *et al.*, 2012]. To obtain output variance σ, we generate distinct sets of dropped neurons within each iteration, to capture the intrinsic uncertainty of deep model. The prediction interval is $[\hat{y} - 2\sigma, \hat{y} + 2\sigma]$ where σ is calculated over 30 iterations with a dropout rate of 0.2.

Evaluation Methodology. For comparing the predictive accuracy of different methods, we employ two different metrics: Normalized Nash–Sutcliffe Efficiency (NNSE) [Nossent and Bauwens, 2012] and Mean Absolute Error (MAE). NSE is a widely recognized metric to assess the precision of hydrolog-

ical models [Nash and Sutcliffe, 1970] and is calculated as:

$$NSE = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (5)$$

Here, n is the number of distinct year-months during testing for the streamflow prediction, y_i is the observed value at month i, $\hat{y}_i$ is the predicted value at month i, and $\bar{y}$ is the mean of the observed values. Since the theoretical lower limit of the NSE $(-\infty)$ can lead to issues in implementations, we normalize the NSE within the range of [0, 1]. The Normalized NSE is given by $NNSE = \frac{1}{2-NSE}$. NNSE takes into account both accuracy of predictions as well as the spread of predicted values around ground truth. Hence, a higher NNSE value indicates better predictive performance.

For evaluating UQ, we employ two metrics: prediction interval (PI) size and marginal coverage (whether ground truth lies in the PI or not). Ideally, we prefer to achieve some minimum target coverage (e.g., 90%) and smaller prediction intervals. We report all results with a single training run.

4.2 Results and Discussion

Performance of Streamflow Predictive Models. Table 2 shows the accuracy results comparing different streamflow prediction models across six watersheds in the high-flow months. We make the following observations. **1)** All data-driven models perform better than physics-based model VIC-CropSyst across all watersheds. **2)** All methods that combine data and knowledge perform better than the pure data-driven model RCGNN for all cases. This demonstrates the value of domain knowledge in the form of the water balance equation **3)** Importance-weighted training improves the predictive accuracy of RCGNN-CRL-IW over the standard training. This result demonstrates the practical value of our theoretically-justified IW training approach. **4)** Accuracy enhancement, relative to the physics-based model, varies by watershed, and time of the year (Fig 2) and Figures in Appendix).

	VIC-CropSyst	RCGNN	PG-RCGNN	RCGNN-CRL	RCGNN-CRL-IW
BR	0.823	0.915	0.919	0.93	**0.934**
C	0.837	0.902	0.925	0.948	**0.952**
CCR	0.817	0.883	0.896	0.931	**0.935**
F	0.684	0.885	0.902	0.917	**0.936**
SFC	0.741	0.834	0.866	0.91	**0.92**
Y	0.212	0.405	0.509	0.635	**0.654**

Table 2: Comparison of predictive models across watersheds during high flow months. Values are reported in terms of the NNSE metric (higher the better). NNSE values less than 0.5 indicate that the mean is a better predictor than the model's output.

Table 3 illustrates the frequency and magnitude of water balance equation violations by PG-RCGNN and RCGNN-CRL-IW across various datasets during test years. Reported magnitudes for PG-RCGNN vary between datasets, reflecting the disparity in violation severity. Only years with constraint violations are included in the magnitude calculations, for a fair comparison. Conversely, RCGNN-CRL-IW consistently maintains the water balance equation across all test
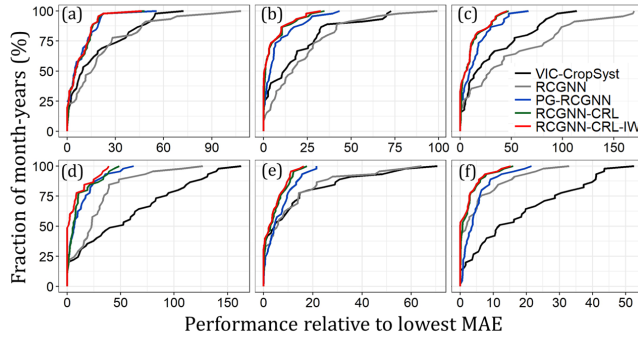
Figure 2: Relative performance chart comparing all models for high flow months (a) Boise, (b) Clearwater, (c) Clearwater Canyon Ranger, (d) Flathead, (e) Southfork Clearwater, and (f) Yakima. For each test month and year, each model's MAE was compared against the best performing model's MAE for that month and year (difference shown in X-axis). The fraction of test month-years for which that performance is achieved is shown along the Y-axis. Closer a curve is aligned with the Y-axis for larger fraction, the better.

years and datasets. This result demonstrates the effectiveness of RCGNN-CRL-IW in always obeying the water balance equation during inference.

	BR	C	CCR	F	SF	Y
PG-RCGNN	0.44 (51.41)	0.44 (25.93)	0.44 (106.1)	0.67 (52.26)	0.11 (8.09)	0.0 (0.0)
RCGNN-CRL-IW	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)

Table 3: Comparison of fraction (magnitude) of constraint violation for PG-RCGNN and RCGNN-CRL-IW. Fraction is the proportion of test years violating the constraint, while magnitude is the average yearly constraint violation measured in millimeters.

Fig 2 shows the relative performance of each model compared to other models. RCGNN-CRL-IW (red line) yields the best performance most consistently for the high flow months. Even, when it is not the best model, its degradation in MAE relative to the best model is low (i.e., the red line sticks closer to the Y-axis). On the other hand, the VIC-CropSyst and RCGNN models are best performing in very few cases, and when they are not the best models, the degradation in performance is relatively high (the lines are further away from the Y-axis). This is primarily due to the failure to capture the timing and magnitude of peak streamflow (see Appendix).

Performance of UQ Methods. Table 4 shows the UQ performance of GP w/ deep kernel learning (DKL) and dropout. The prediction intervals are in streamflow units. We make the following observations. **1)** Dropout produces very small prediction intervals but the coverage is low (less than 80 percent for all but one watershed) which is not desirable. **2)** GP w/ DKL achieves significantly higher coverage (more than 91 percent for all watersheds) at the expense of a slight increase in the prediction interval size over the dropout method.

	Dropout		GP w/ DKL	
	coverage	PI width	Coverage	PI width
BR	74.07	4.77	91.66	5.99
C	76.85	4.7	93.52	6.3
CCR	79.62	5.47	92.59	6.27
F	75.92	4.92	94.44	6.28
SFC	78.7	4.59	95.73	6.08
Y	81.48	4.37	94.44	5.84

Table 4: Comparison of marginal coverage and prediction interval size for dropout and GP w/ DKL approaches.

Modeling streamflow in human-influenced watersheds. The VIC-CropSyst model simulates "natural" flows without human interventions such as reservoir operations or irrigation. Hence, its performance tends to be poor in human-influenced watersheds such as the Yakima River basin, where much river water is diverted for agriculture. The RCGNN-CRL-IW model shows better performance in such contexts, capturing some human influence nuances. Nonetheless, it exhibits relatively lower NNSE values, highlighting the challenge of modeling streamflow in human-influenced watersheds. This underscores the necessity to explicitly incorporate elements of human influence into the learning approach.

5 Roadmap to Deployment

The intended application base for our streamflow prediction methodology are federal and state/regional agencies that manage reservoirs or engage in other forms of water resources management and planning. However, to enable such partnerships in the near term, several improvements and extensions have been planned. First, the problem formulation needs to be extended to a forecast setting where expectations of meteorological and other input variables inform the streamflow predictions. Datasets for expectations of weather are available in the same format as our current inputs and can be directly integrated. Our current prediction accuracies provide an upper bound which will diminish when errors associated with expectations of input variables are added in. To this end, the provision of uncertainty bounds supported by our framework will be crucial for the end-user. User interfaces that display the uncertainty metrics in a way that is most useful for decision-making are needed for deployment. We have been exploring the current decision-making interfaces used by the agency to seamlessly integrate. Finally, our stakeholder agency identified explainability of the model as a key feature of interest in transitioning to deployment and we can adapt existing work on explainable AI for this purpose.

6 Summary

This paper studied a constrained reasoning and learning approach for streamflow prediction by combining domain knowledge with deep temporal models, and Gaussian process with deep kernel learning approach for uncertainty quantification. Experimental results demonstrated the effectiveness of our methodology on real-world watershed datasets and in guaranteeing that constraints are satisfied for all predictions.

Contribution Statement

Yan Yan contributed to this paper. Due to a human error during the submission process, we missed adding him as author. We were not able to fix this error due to IJCAI policy.

Acknowledgements

References

[Ahmed *et al.*, 2022a] Kareem Ahmed, Tao Li, Thy Ton, Quan Guo, Kai-Wei Chang, Parisa Kordjamshidi, Vivek Srikumar, Guy Van den Broeck, and Sameer Singh. Pylon: A pytorch framework for learning with constraints. In *NeurIPS 2021 Competitions and Demonstrations Track*, pages 319–324. PMLR, 2022.

[Ahmed *et al.*, 2022b] Kareem Ahmed, Stefano Teso, Kai-Wei Chang, Guy Van den Broeck, and Antonio Vergari. Semantic probabilistic layers for neuro-symbolic learning. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, dec 2022.

[Cai *et al.*, 2021] Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (PINNs) for fluid mechanics: A review, 2021.

[Choi *et al.*, 2020] Y Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. *UCLA. URL: http://starai. cs. ucla. edu/papers/ProbCirc20. pdf*, 2020.

[Cortes *et al.*, 2010] Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In *Advances in neural information processing systems*, volume 23, 2010.

[Hinton *et al.*, 2012] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[Hoedt *et al.*, 2021] Pieter-Jan Hoedt, Frederik Kratzert, Daniel Klotz, Christina Halmich, Markus Holzleitner, Grey S Nearing, Sepp Hochreiter, and Günter Klambauer. Mc-lstm: Mass-conserving lstm. In *International conference on machine learning*, pages 4275–4286. PMLR, 2021.

[Jia *et al.*, 2019] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan Read, Jacob Zwart, Michael Steinbach, and Vipin Kumar. Physics guided rnns for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM international conference on data mining*, pages 558–566. SIAM, 2019.

[Jia *et al.*, 2021] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan S Read, Jacob A Zwart, Michael Steinbach, and Vipin Kumar. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. *ACM/IMS Transactions on Data Science*, 2(3):1–26, 2021.

[Kamruzzaman *et al.*, 2021] Sarker Md Kamruzzaman, Zhou Lu, Eberhart Aaron, and Hitzler Pascal. Neurosymbolic artificial intelligence: Current trends. *arXiv preprint arXiv: 2105.05330*, 2021.

[Karpatne *et al.*, 2016] Anuj Karpatne, Ankush Khandelwal, Xi Chen, Varun Mithal, James Faghmous, and Vipin Kumar. Global monitoring of inland water dynamics: State-of-the-art, challenges, and opportunities. *Computational sustainability*, pages 121–147, 2016.

[Karpatne *et al.*, 2017a] Anuj Karpatne, Gowtham Atluri, James H Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on knowledge and data engineering*, 29(10):2318–2331, 2017.

[Karpatne *et al.*, 2017b] Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, 2, 2017.

[Kratzert *et al.*, 2019] Frederik Kratzert, Daniel Klotz, Mathew Herrnegger, Alden K Sampson, Sepp Hochreiter, and Grey S Nearing. Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resources Research*, 55(12):11344–11354, 2019.

[Liang *et al.*, 1994] Xu Liang, Dennis P Lettenmaier, Eric F Wood, and Stephen J Burges. A simple hydrologically based model of land surface water and energy fluxes for general circulation models. *Journal of Geophysical Research: Atmospheres*, 99(D7):14415–14428, 1994.

[Liu *et al.*, 2023] Siyan Liu, Dan Lu, Scott L Painter, Natalie A Griffiths, and Eric M Pierce. Uncertainty quantification of machine learning models to improve streamflow prediction under changing climate and environmental conditions. *Frontiers in Water*, 5:1150126, 2023.

[Malek *et al.*, 2017] Keyvan Malek, Claudio Stöckle, Kiran Chinnayakanahalli, Roger Nelson, Mingliang Liu, Kirti Rajagopalan, Muhammad Barik, and Jennifer C Adam. Vic–cropsyst-v2: A regional-scale modeling platform to simulate the nexus of climate, hydrology, cropping systems, and human decisions. *Geoscientific Model Development*, 10(8):3059–3084, 2017.

[McLain *et al.*, 2015] Kelly McLain, Jaclyn Hancock, and Margaret Drennan. 2015 drought and agriculture, 2015. Accessed: June 26, 2024.

[Muralidhar *et al.*, 2020] Nikhil Muralidhar, Jie Bu, Ze Cao, Long He, Naren Ramakrishnan, Danesh Tafti, and Anuj Karpatne. Phynet: Physics guided neural networks for particle drag force prediction in assembly. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 559–567. SIAM, 2020.

[Nash and Sutcliffe, 1970] J Eamonn Nash and Jonh V Sutcliffe. River flow forecasting through conceptual models part i—a discussion of principles. *Journal of hydrology*, 10(3):282–290, 1970.

[Nossent and Bauwens, 2012] Jiri Nossent and Willy Bauwens. Application of a normalized nash-sutcliffe efficiency to improve the accuracy of the sobol'sensitivity analysis of a hydrological model. In *EGU general assembly conference abstracts*, page 237, 2012.

[Peters-Lidard *et al.*, 2017] Christa D Peters-Lidard, Martyn Clark, Luis Samaniego, Niko EC Verhoest, Tim Van Emmerik, Remko Uijlenhoet, Kevin Achieng, Trenton E Franz, and Ross Woods. Scaling, similarity, and the fourth paradigm for hydrology. *Hydrology and earth system sciences*, 21(7):3701–3713, 2017.

[Rajagopalan *et al.*, 2018] K Rajagopalan, KJ Chinnayakanahalli, CO Stockle, RL Nelson, CE Kruger, MP Brady, K Malek, ST Dinesh, ME Barber, AF Hamlet, et al. Impacts of near-term climate change on irrigation demands and crop yields in the columbia river basin. *Water Resources Research*, 54(3):2152–2182, 2018.

[Seo *et al.*, 2016] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks, 2016.

[Sit *et al.*, 2021] Muhammed Sit, Bekir Demiray, and Ibrahim Demir. Short-term hourly streamflow prediction with graph convolutional gru networks. *arXiv preprint arXiv:2107.07039*, 2021.

[Stöckle *et al.*, 2003] Claudio O Stöckle, Marcello Donatelli, and Roger Nelson. Cropsyst, a cropping systems simulation model. *European journal of agronomy*, 18(3-4):289–307, 2003.

[Valentin Jospin *et al.*, 2020] Laurent Valentin Jospin, Wray Buntine, Farid Boussaid, Hamid Laga, and Mohammed Bennamoun. Hands-on bayesian neural networks–a tutorial for deep learning users. *arXiv e-prints*, pages arXiv–2007, 2020.

[Vergari *et al.*, 2020] Antonio Vergari, Y Choi, Robert Peharz, and Guy Van den Broeck. Probabilistic circuits: Representations, inference, learning and applications. *AAAI Tutorial*, 2020.

[Williams and Rasmussen, 2006] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[Wilson *et al.*, 2016] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016.

[Xiang and Demir, 2022] Zhongrun Xiang and Ibrahim Demir. Fully distributed rainfall-runoff modeling using spatial-temporal graph neural network. 2022.

[Xu *et al.*, 2021] Da Xu, Yuting Ye, and Chuanwei Ruan. Understanding the role of importance weighting for deep learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.