Stochastic Modeling of Multiple Streamflow Time Series in Colombian Based on Gaussian Processes

Author: Julián David Pastrana-Cortés

Director: Álvaro Angel Orozco-Gutiérrez

Co-director: David Augusto Cardenas-Peña

Automatic Research Group

September 17, 2024

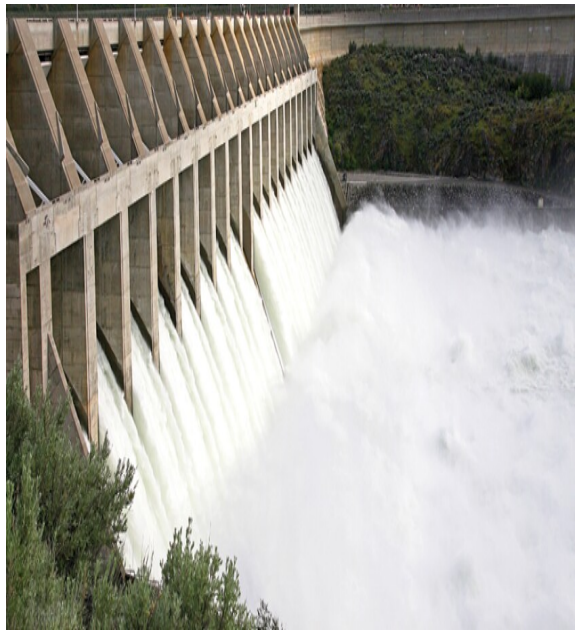
Introduction

Motivation

Understanding the implications of time series associated with hydrological variables, such as flow rates or reservoir levels, is essential for hydroelectric generation and the planning of other generation systems in Colombia

(a) Irrigation

(b) Flood control



(c) Hydropower generation

Challenges: non-linearities, high stochasticity, and complex water resource patterns.

The Importance of Hydrological Forecasting

Understanding hydrological processes has become increasingly critical in the field of natural resource management, anticipation capacity of extreme hydrological events such as droughts and heavy rainfall.

(a) Drought Condition

(b) Full Dam

Problem Statement

- Physically driven models for water resource forecasting are complex and require extensive parameter knowledge, limiting their practicality [1, 2].
- Data-driven models, such as autoregressive (AR) models, struggle to capture nonlinearities in water resources time series [3].
- Neural networks (ANNs, RNNs) improve on nonlinearity, but face overfitting, gradient vanishing and exploding, limiting ability to capture long-term dependencies [4, 5, 6].
- LSTM networks overcome these issues, but lack uncertainty quantification, crucial for decision-making in hydrological forecasting [7, 8].
- Gaussian Processes (GPs) provide uncertainty and handle nonlinearities, but scaling to multi-task forecasting remains a challenge [9, 10, 11].

Objectives

Objectives

General Objective:
Develop a stochastic forecasting model for making multiple simultaneous predictions of hydrological time series. This model will take advantage of cross-correlations among the tasks to improve performance, while maintaining scalability for short-term horizons.

Specific Objectives:

- Develop a model that allows the forecasting of hydrological time series, properly quantifying the uncertainty associated with each value within the prediction horizons.
- Design a multi-task forecasting methodology that captures and models cross-correlations between hydrological time series, to improve forecast accuracy within forecast horizons.
- Develop a multi-task prediction methodology that handles data constraints across reservoirs while maintaining high forecasting performance as measured by probabilistic metrics.

The Dataset

Problem Setting

We model hydrological time series using observed resource vectors. At each time step n, the vector $\mathbf{v}_n \in \mathbb{R}^D$ represents resources across D outputs.
The input vector $\mathbf{x}_n$ for the model is constructed from the resource vectors from time n back to $n - T + 1$:

$$\mathbf{x}_n = \begin{bmatrix} \mathbf{v}_n^\top \\ \mathbf{v}_{n-1}^\top \\ \vdots \\ \mathbf{v}_{n-T+1}^\top \end{bmatrix} \in \mathcal{X}$$

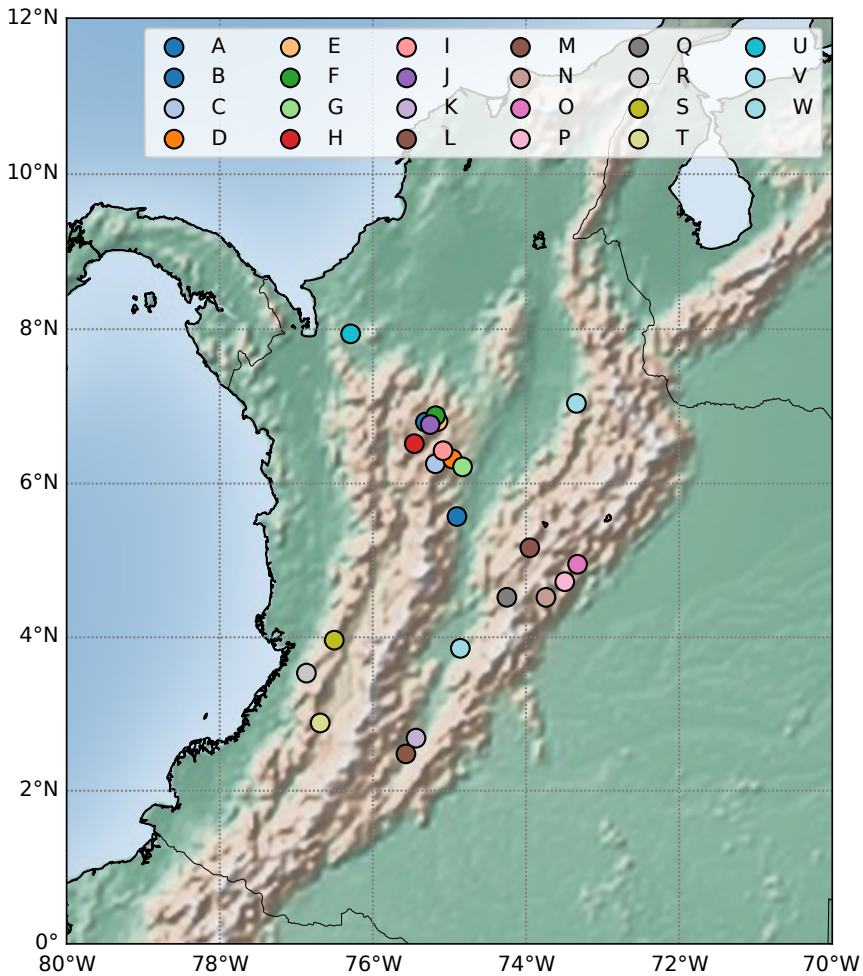
Here, T is the model order and H is the prediction horizon and $\mathcal{X} \subset \mathbb{R}^{DT}$ represents the input space.

The target output vector $\mathbf{y}_n$ is:

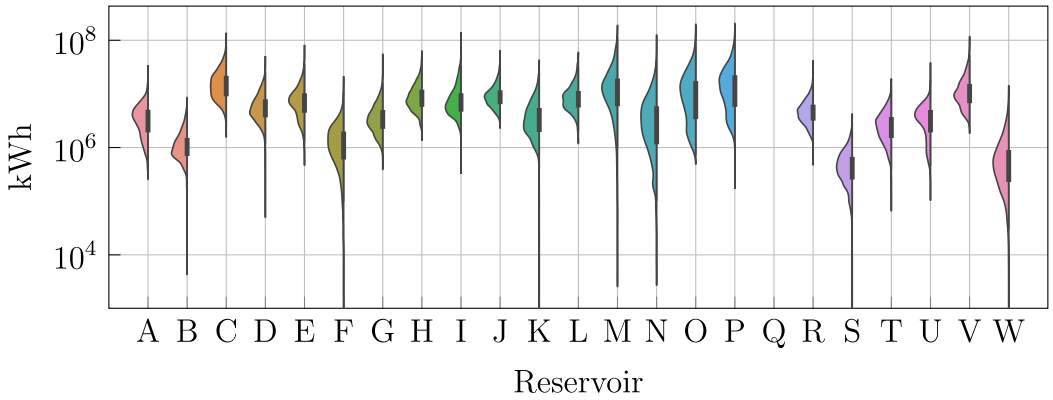
$$\mathbf{y}_n = \mathbf{v}_{n+H} \in \mathbb{R}^D$$

We build a dataset $\mathcal{D} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N = \{\mathbf{X}, \mathbf{y}\}$, comprising N input-output pairs.

Reservoir Locations and Dataset Overview



The hydrological forecasting task utilizes daily streamflow data from 23 Colombian reservoirs from January 1, 2010, to February 28, 2022.



Although volumetric measurements are recorded, they are reported in kilowatt-hours (kWh) by the hydroelectric power plants.

Methodology

Performance Metrics:

- Mean Squared Error (MSE)
- Mean Standardized Log Loss (MSLL)
- Continuous Ranked Probability Score (CRPS)
- Negative Log Predictive Density (NLPD)

Gaussian Process Models:

- Start with a single-output GP for stochastic regression.
- Extend to multi-output GPs, capturing dependencies across multiple reservoirs.
- Introduce Chained Correlated Gaussian Processes to handle non-Gaussian likelihoods.

Gaussian Process Regression: Bayesian Non-Parametric Model

Gaussian Process (GP) Framework

In a GP framework, the function $\boldsymbol{f}(\cdot)$ maps inputs $\boldsymbol{x}_n$ to outputs $\boldsymbol{y}_n$. Adding i.i.d. Gaussian noise $\boldsymbol{\epsilon}$, the model becomes:

$$\boldsymbol{y}_n = \boldsymbol{f}(\boldsymbol{x}_n) + \boldsymbol{\epsilon}$$

For test inputs $\mathbf{X}_*$, the joint distribution of training outputs $\mathbf{y}$ and test outputs $\mathbf{f}_*$ is:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_y & \mathbf{K}_* \\ \mathbf{K}_*^\top & \mathbf{K}_{**} \end{bmatrix}\right)$$

The posterior distribution for test points is:

$$\mathbf{f}_* | \mathbf{X}_*, \mathcal{D} \sim \mathcal{N}(\mathbf{K}_*^\top \mathbf{K}_y^{-1} \mathbf{y}, \mathbf{K}_{**} - \mathbf{K}_*^\top \mathbf{K}_y^{-1} \mathbf{K}_*)$$

- $\mathbf{K}_y = \mathbf{K} + \Sigma_\epsilon$, where $\mathbf{K} \in \mathbb{R}^{ND \times ND}$ is the covariance matrix for the train set and Σ_ϵ contains task-wise noise.
- $\mathbf{K}_{**} \in \mathbb{R}^{N_* D \times N_* D}$ is the covariance matrix for the test set.
- $\mathbf{K}_* \in \mathbb{R}^{ND \times N_* D}$ represents the cross-covariance matrix between the training and test points.

The Marginal Log-likelihood

The prediction performance achieved by the conditional distribution is influenced by the selected parameter set $\boldsymbol{\theta}$ and the observation noise matrix Σ_ϵ. These parameters are determined by maximizing the marginal log-likelihood, where the marginal likelihood $p(\mathbf{y})$ follows a Gaussian distribution:

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y} \mid \mathbf{0}, \mathbf{K}_y)$$

The optimization problem is defined as:

$$\begin{aligned}\{\boldsymbol{\theta}_{\text{opt}}, \Sigma_{\epsilon\text{opt}}\} &= \arg\max_{\boldsymbol{\theta}, \Sigma_\epsilon} \quad \ln p(\mathbf{y}) \\ &= \arg\min_{\boldsymbol{\theta}, \Sigma_\epsilon} \quad \frac{1}{2}\mathbf{y}^\top \mathbf{K}_y^{-1}\mathbf{y} + \frac{1}{2}\ln|\mathbf{K}_y| + \frac{ND}{2}\ln 2\pi\end{aligned}$$

However, the main challenge lies in the computational complexity of $\mathcal{O}(N^3D^3)$ and the storage demand of $\mathcal{O}(N^2D^2)$ due to the need to invert the matrix $\mathbf{K}_y$.

Variational Inference and Sparse Variational GPs (SVGPs)

We introduce $M \ll N$ inducing points Z, with inducing variables $\mathbf{u} \in \mathbb{R}^{MD}$ to reduce computational complexity. The joint distribution becomes:

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{f} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0} \, , \begin{bmatrix} \mathbf{K}_{uu} & \mathbf{K}_{uf} \\ \mathbf{K}_{uf}^{\top} & \mathbf{K} \end{bmatrix} \right)$$

Where $\mathbf{K}_{uu} \in \mathbb{R}^{MD \times MD}$, and $\mathbf{K}_{uf} \in \mathbb{R}^{MD \times ND}$. The posterior distribution uses the variational approximation $q(\mathbf{u}) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{S})$. Now we maximizing the Evidence Lower Bound (ELBO):

$$\mathcal{L} = \sum_{d=1}^{D} \sum_{n=1}^{N} \mathbb{E}_{q(f_d(\mathbf{x}_n))} \{ \ln p(y_{dn} \mid f_d(\mathbf{x}_n)) \} - \sum_{d=1}^{D} \mathrm{KL}\{ q(\mathbf{u}_d) \parallel p(\mathbf{u}_d) \} \leq \ln p(\mathbf{y})$$

where $f_d(\mathbf{x}_n)$ represents the d-th latent function value at input $\mathbf{x}_n$, and y_{dn} is the corresponding observed value. This reduces the complexity to $\mathcal{O}(NM^2D^3)$. For predictions at new points $\mathbf{x}_*$, we add noise in Σ_ϵ to $q(\mathbf{f}_*) = \int p(\mathbf{f}_* \mid \mathbf{u}) q(\mathbf{u}) d\mathbf{u}$.

Model Setup

The GP covariance is factorized into two kernels: $k_{\mathcal{X}}$ for input correlations and k_D for task correlations:

$$k\left((\boldsymbol{x}, d), (\boldsymbol{x}', d')\right) = k_{\mathcal{X}}\left(\boldsymbol{x}, \boldsymbol{x}' \mid \boldsymbol{\Theta}_d\right) k_D\left(d, d' \mid \sigma_d\right),$$

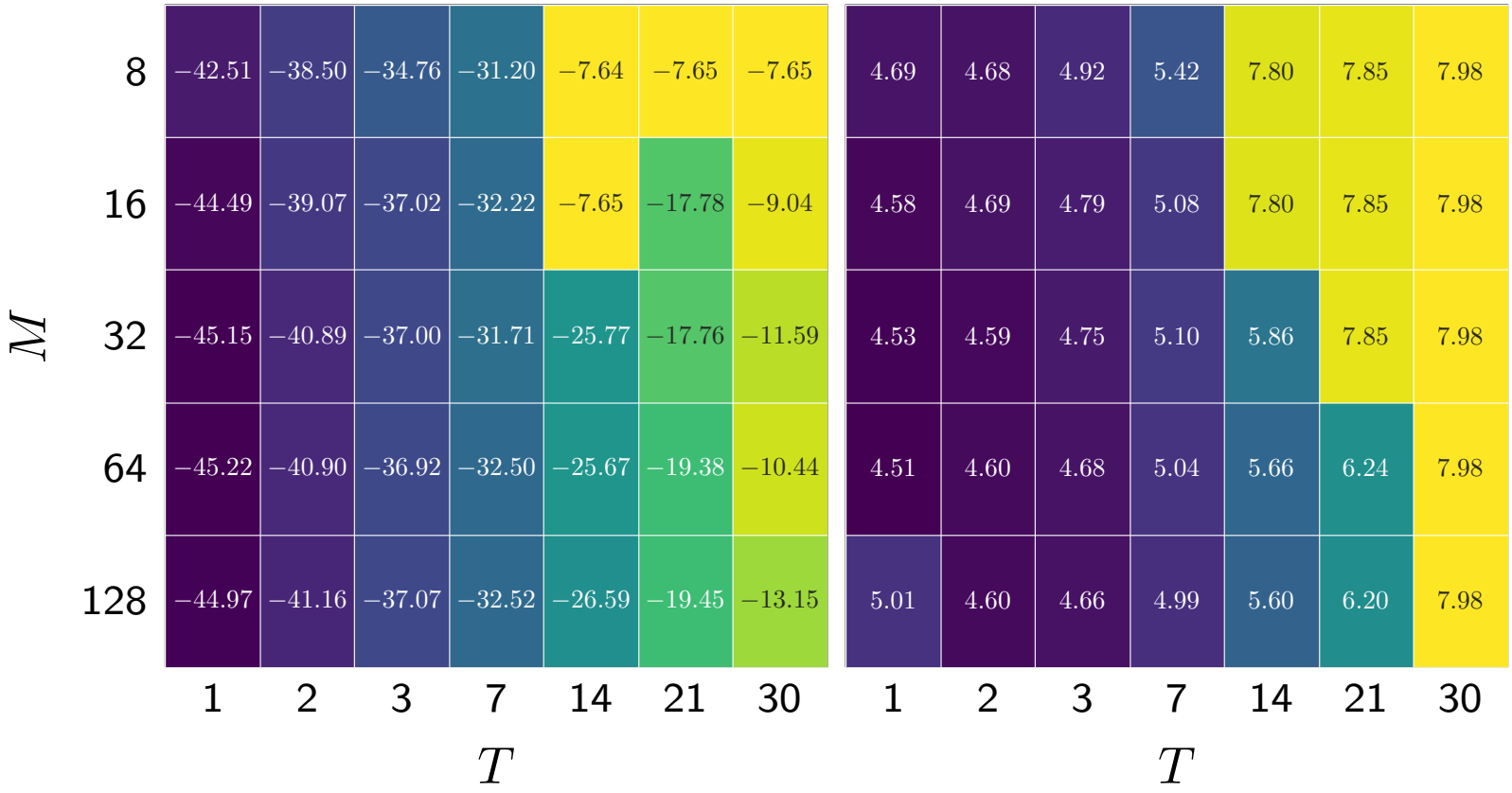
with:

$$k_{\mathcal{X}}\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')^\top \boldsymbol{\Theta}_d^{-2}(\boldsymbol{x} - \boldsymbol{x}')\right),$$

$$k_D(d, d') = \sigma_d^2 \delta_{d,d'},$$

where $\delta_{d,d'}$ is the Kronecker delta, $\boldsymbol{\Theta}_d$ is the lengthscale matrix, and σ_d^2 is the output scale. This reduces complexity to $\mathcal{O}(NM^2D)$ by avoiding explicit task correlations.

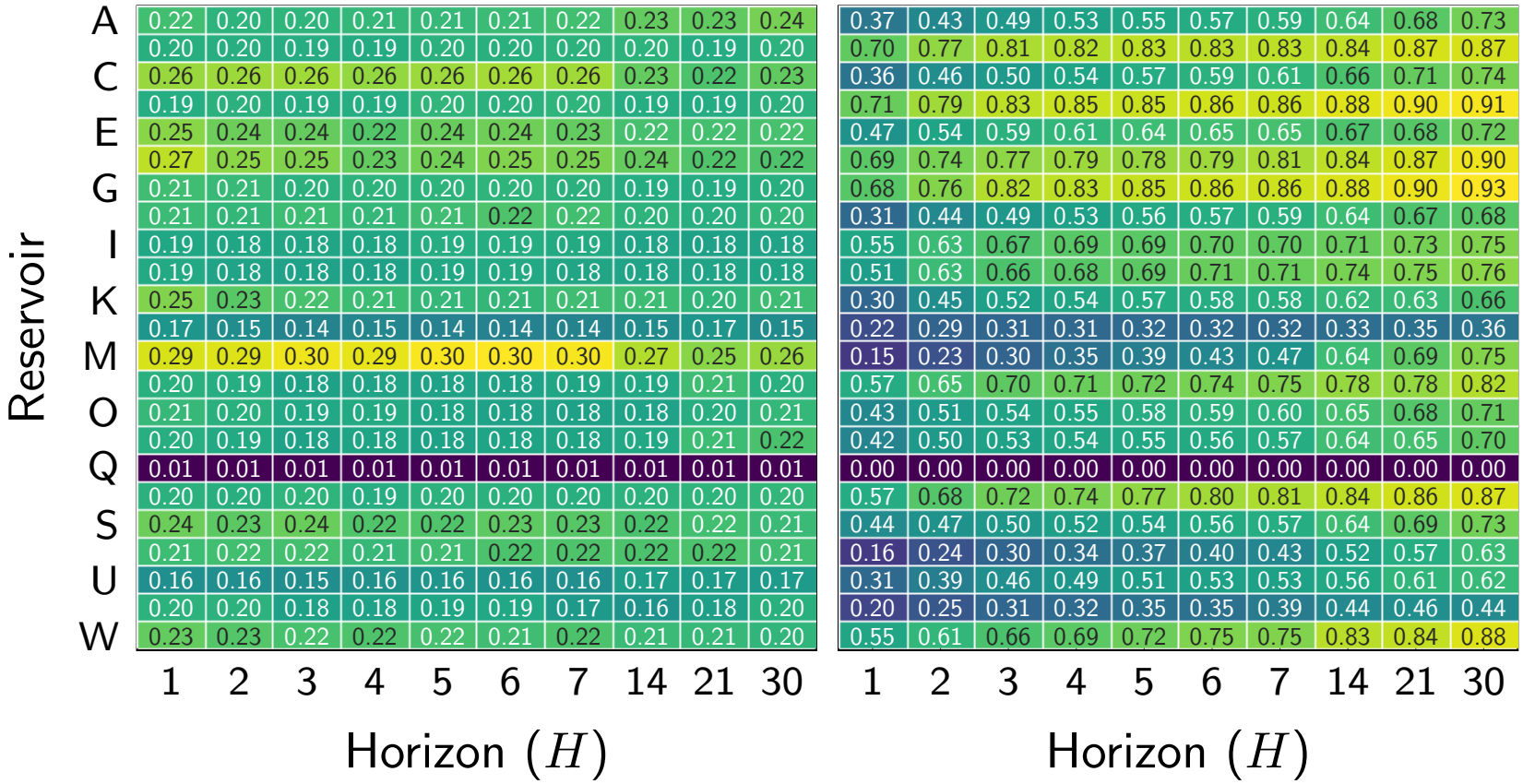
Tuning M and T



(a) MSLL ($\times 10^{-2}$) (b) MSE ($\times 10^{-1}$)

Grid search average values for tuning the model order T and the number of inducing points M. The optimal settings are $M = 64$ and $T = 1$

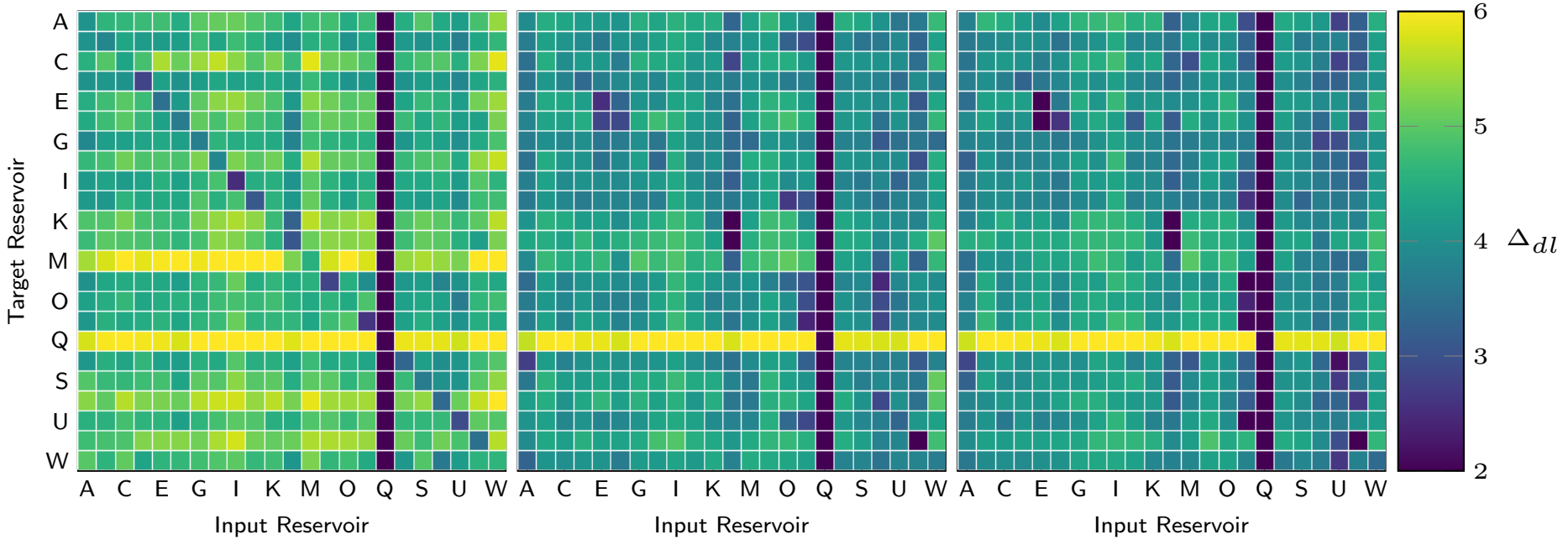
Reservoir-Wise Output Scales and Noise Variance



(a) Output scales (b) Noise variances

Output scales σ_d^2 and noise variance Σ_ϵ tuned for each horizon and reservoir. Longer horizons generally show smaller output scales and higher noise variance.

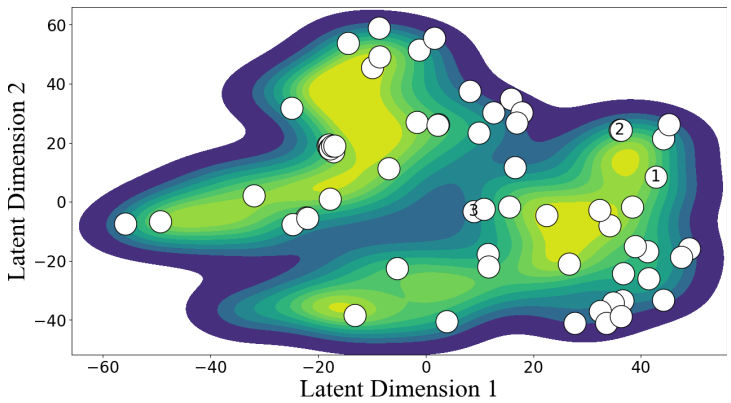
Lengthscale Analysis



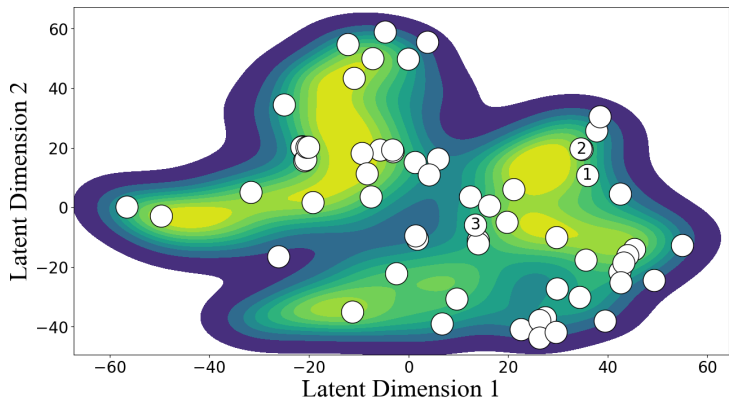
(a) $H = 1$. (b) $H = 14$. (c) $H = 30$.

Trained lengthscales from input features (columns) to output tasks (rows) for three prediction horizons. As the horizon increases, main diagonal lengthscales lose relevance, while off-diagonal ones gain importance.

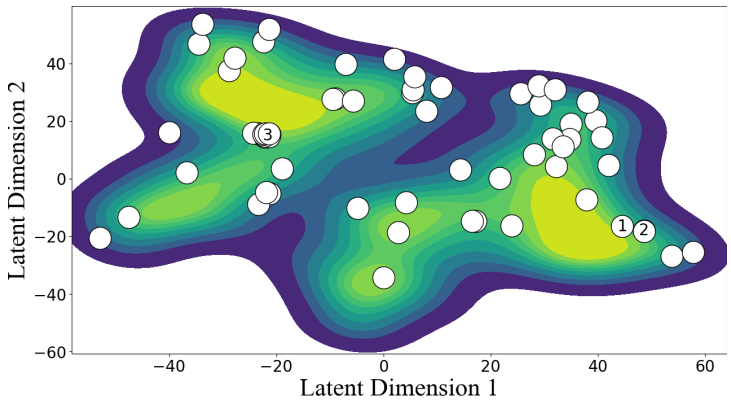
t-distributed Stochastic Neighbor Embedding (t-SNE)



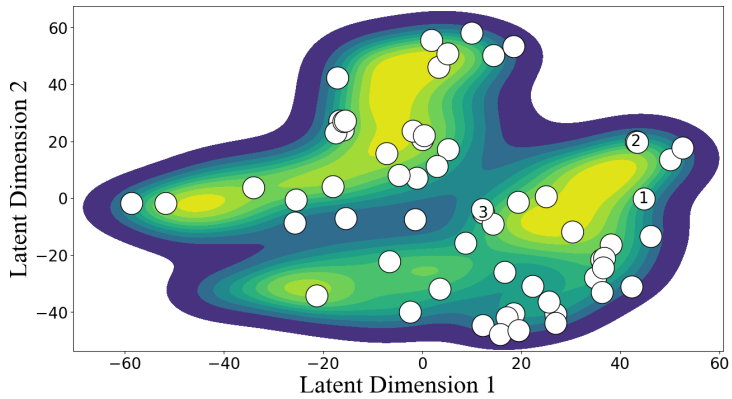
(a) Reservoir E.



(b) Reservoir I.



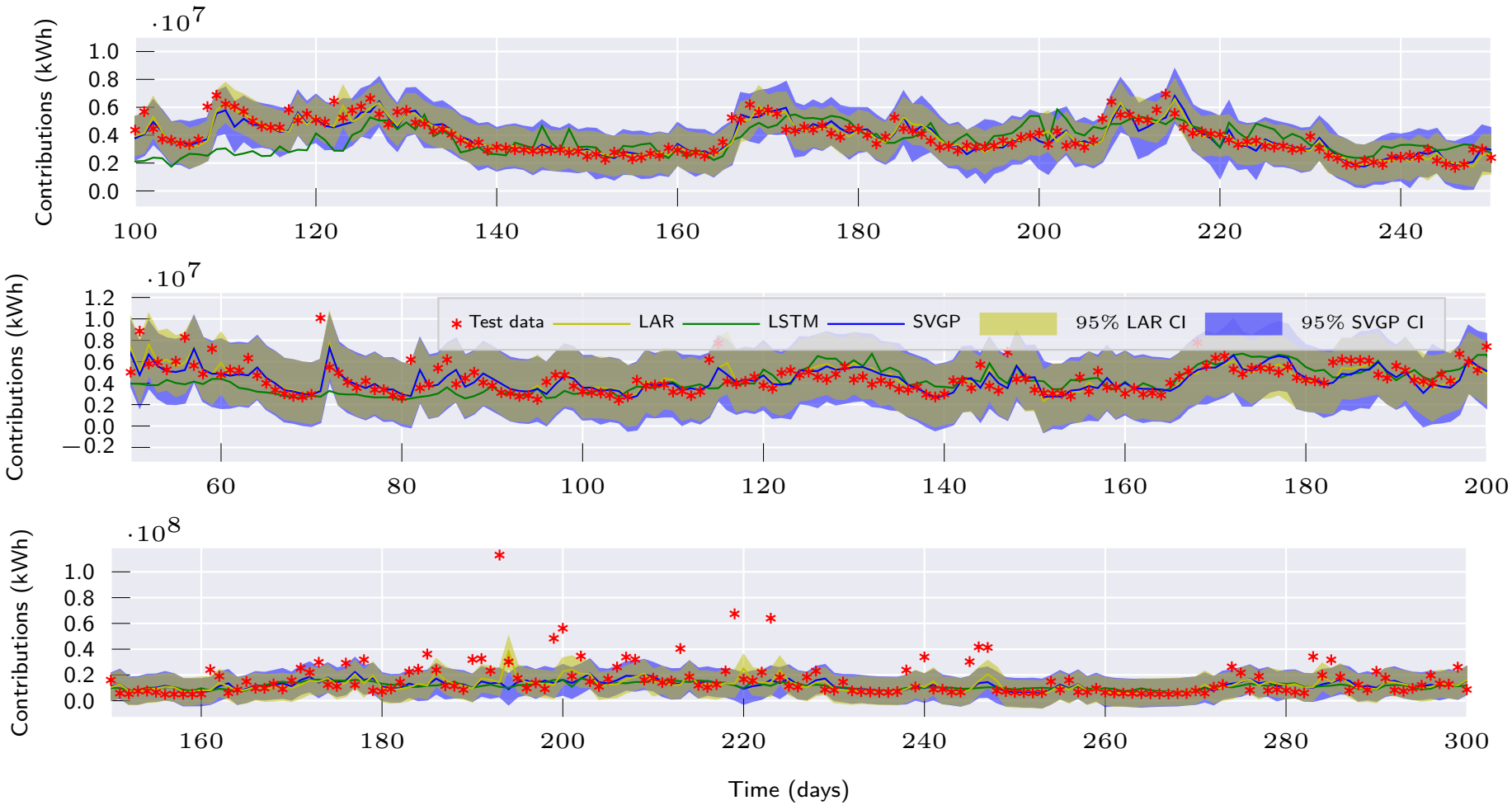
(c) Reservoir L.



(d) Reservoir U.

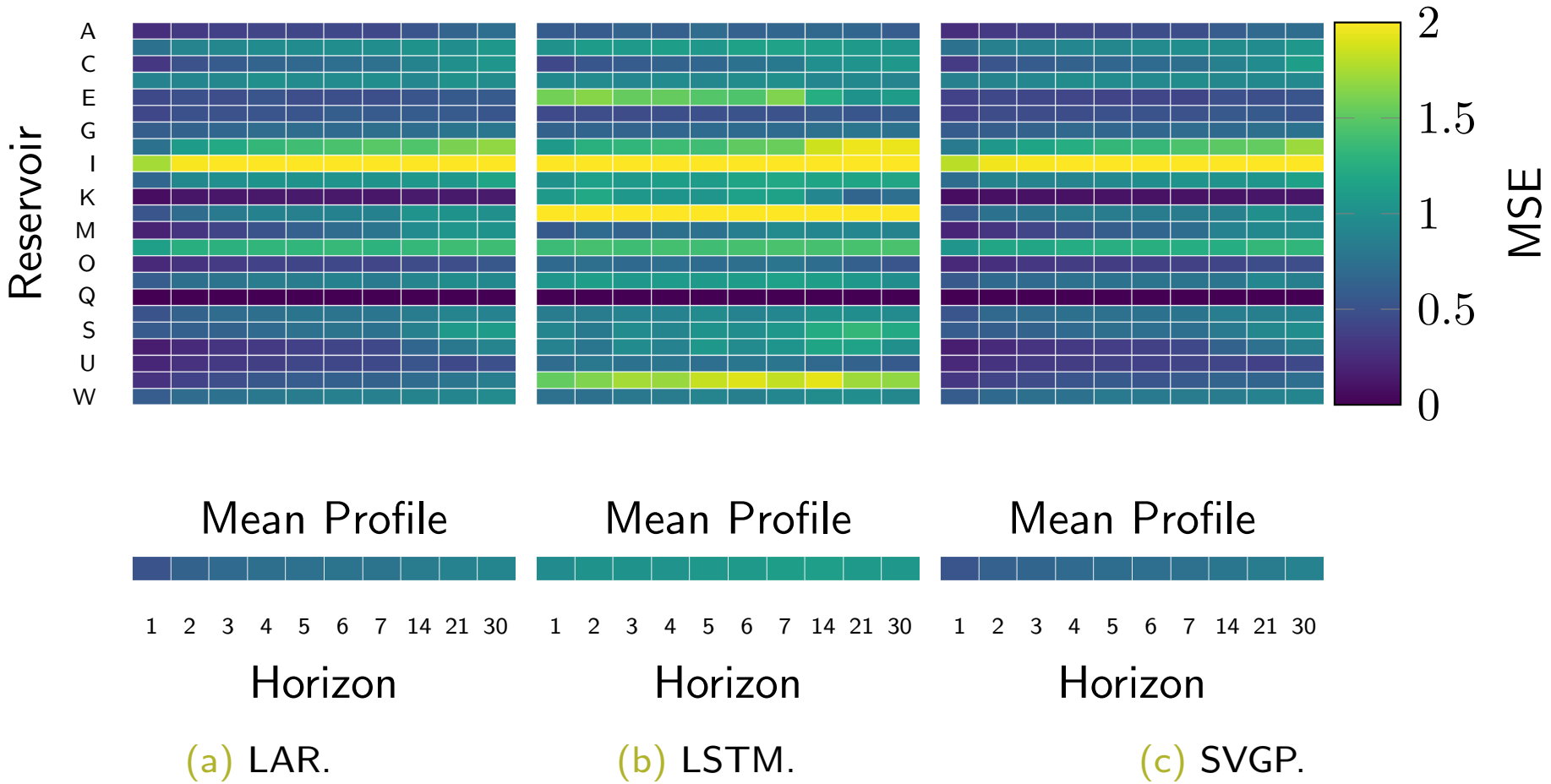
t-SNE-based 2D mapping of the SVGP latent space and inducing points' locations for four target reservoirs. The shared inducing points allow for the capturing of task-wise, and global information about the streamflow dynamics.

Models Forecasting



One-day-ahead model predictions for reservoirs T, A, and I (top to bottom). The SVGP adapts better to time-series data and captures its stochastic nature through the predictive distribution.

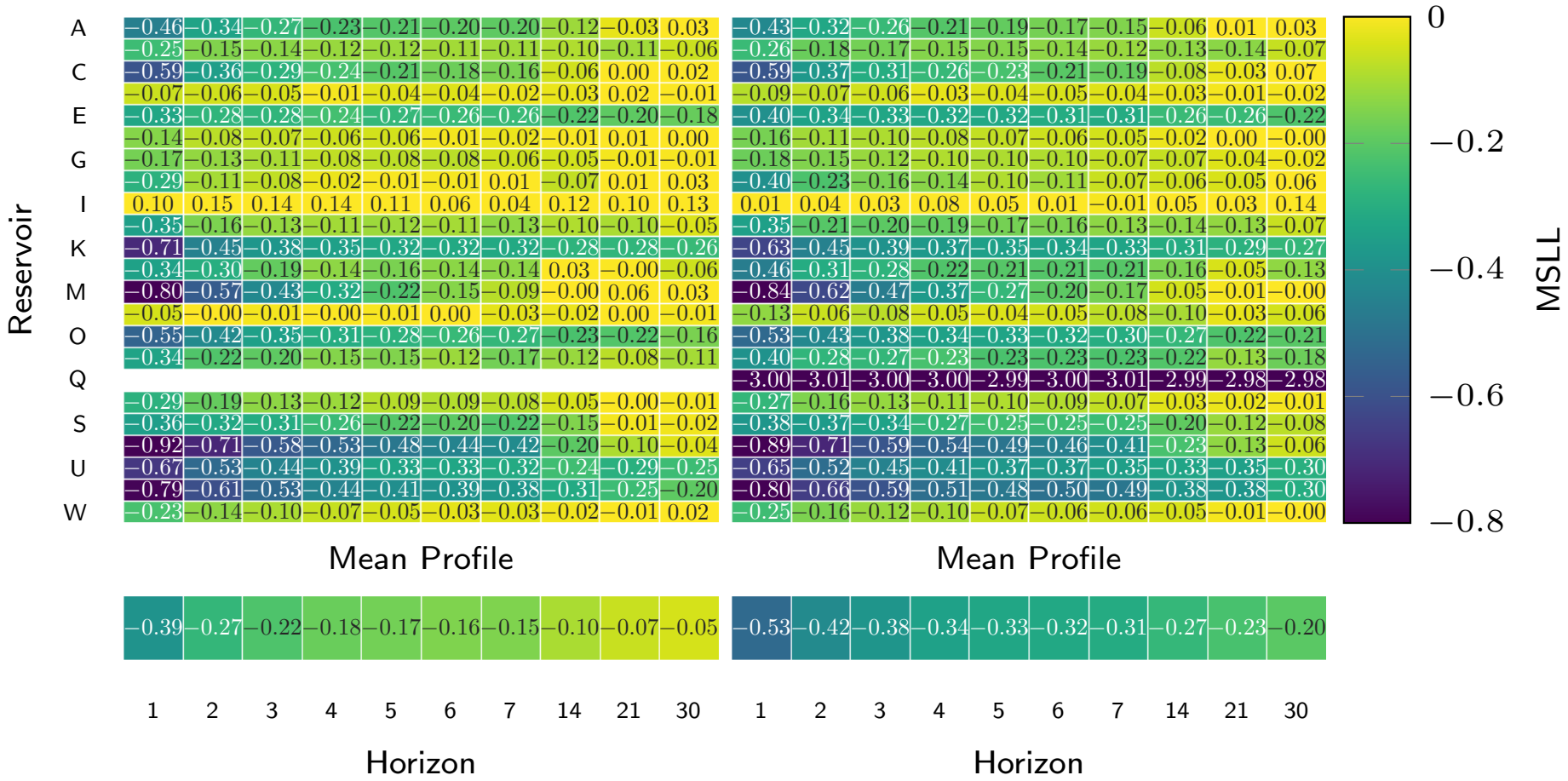
MSE Scores



(a) LAR. (b) LSTM. (c) SVGP.

MSE achieved by LAR, LSTM, and SVGP forecasting models for each horizon and reservoir. The LAR and SVGP models significantly outperform the LSTM models across all scenarios.

MSLL Score



(a) LAR. (b) SVGP.

Achieved MSLL for LAR and SVGP models across horizons and reservoirs. Longer horizons yield larger errors. The SVGP models exhibits lower error and a slower error increase with the horizon compared to the LAR models.

Performance t-test

Performance metrics for LAR, LSTM, and SVGP across horizons H. Bold and asterisk indicate a p-value $p < 1\%$ (LAR vs. SVGP, LSTM vs. SVGP). SVGP outperforms all models, except LAR at $H = 1$, where linear dependence is stronger. As horizon increases, SVGP captures complex input-output relations, significantly outperforming the other models.

	MSE			MSLL		CRPS	
H	**LAR**	**LSTM**	**SVGP**	**LAR**	**SVGP**	**LAR**	**SVGP**
1	0.51	0.96	0.52 *	−0.39	−0.53	0.34	0.32
2	0.63	1.01	**0.61** *	−0.27	**−0.42**	0.39	**0.36**
3	0.68	1.02	**0.65** *	−0.22	**−0.38**	0.41	**0.38**
4	0.72	1.03	**0.69** *	−0.18	**−0.34**	0.42	**0.39**
5	0.74	1.06	**0.71** *	−0.17	**−0.33**	0.43	**0.40**
6	0.76	1.07	**0.72** *	−0.16	**−0.32**	0.44	**0.40**
7	0.76	1.11	**0.74** *	−0.15	**−0.31**	0.44	**0.41**
14	0.83	1.12	**0.79** *	−0.10	**−0.27**	0.46	**0.43**
21	0.88	1.08	**0.83** *	−0.07	**−0.23**	0.48	**0.45**
30	0.91	1.06	**0.88** *	−0.05	**−0.20**	0.49	**0.46**
Grand Average	0.74	1.05	**0.71** *	−0.18	**−0.33**	0.43	**0.40**

To Conclude

- The proposed methodology reduces computational complexity from cubic to linear, improving scalability for large datasets.
- The optimal number of inducing points provides regularization, avoiding overfitting while capturing key data features.
- The model strategically places shared inducing points, balancing task-specific and global dynamics to enhance streamflow forecasting.
- Adaptive lengthscales allow the model to adjust to varying prediction horizons, improving robustness for multi-output tasks.
- The SVGP model outperforms LAR and LSTM by better handling dynamics, providing uncertainty estimates, and showing slower error growth over long horizons.

Multi-Output Gaussian Processes: Modeling Inter-Output Dependencies

Multi Output Gaussian Processes

We start to extending the notation developed so far to learning multiple outputs with a single model, we consider

$$\boldsymbol{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_d(\mathbf{x}), \cdots, f_D(\mathbf{x})]^\top$$

as a multi-output Gaussian process, comprising D single-output Gaussian processes. Now our target is a vector-valued function.

Independent Gaussian Process (IGP)

Combining all independent GP-based models evaluated in the N_* test point set X_*, assuming that all outputs are fully observed for each input, the vector function space inference corresponds to the following generative model:

$$\begin{bmatrix} \mathbf{f}_{1*} \\ \mathbf{f}_{2*} \\ \vdots \\ \mathbf{f}_{D*} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0} \ , \begin{bmatrix} K_{1**} & 0 & \cdots & 0 \\ 0 & K_{2**} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K_{D**} \end{bmatrix} \right)$$

Here, $K_{d**} \in \mathbb{R}^{N_* \times N_*}$ correspond to the covariance matrix of the d-th output. Due to the independence assumption, the grand covariance matrix of the multi-output model $\mathbf{K}_{**} \in \mathbb{R}^{DN_* \times DN_*}$ is diagonal by block.

We call this model Independent Gaussian process (IGP).

Modeling Output Interactions

We introduce a framework based on a set of independent Gaussian processes $\{u_q(\mathbf{x})\}_{q=1}^{Q}$, with their own covariance function $k_q(\mathbf{x}, \mathbf{x}')$ and each latent process $f_d(\mathbf{x})$ is represented as an instantaneous, time-invariant linear combination of the independent processes:

$$f_d(\mathbf{x}) = \sum_{q=1}^{Q} a_{d,q} u_q(\mathbf{x}) \qquad u_q(\mathbf{x}) \sim \mathcal{GP}(0, k_q(\mathbf{x}, \mathbf{x}'))$$

Graphical Representation

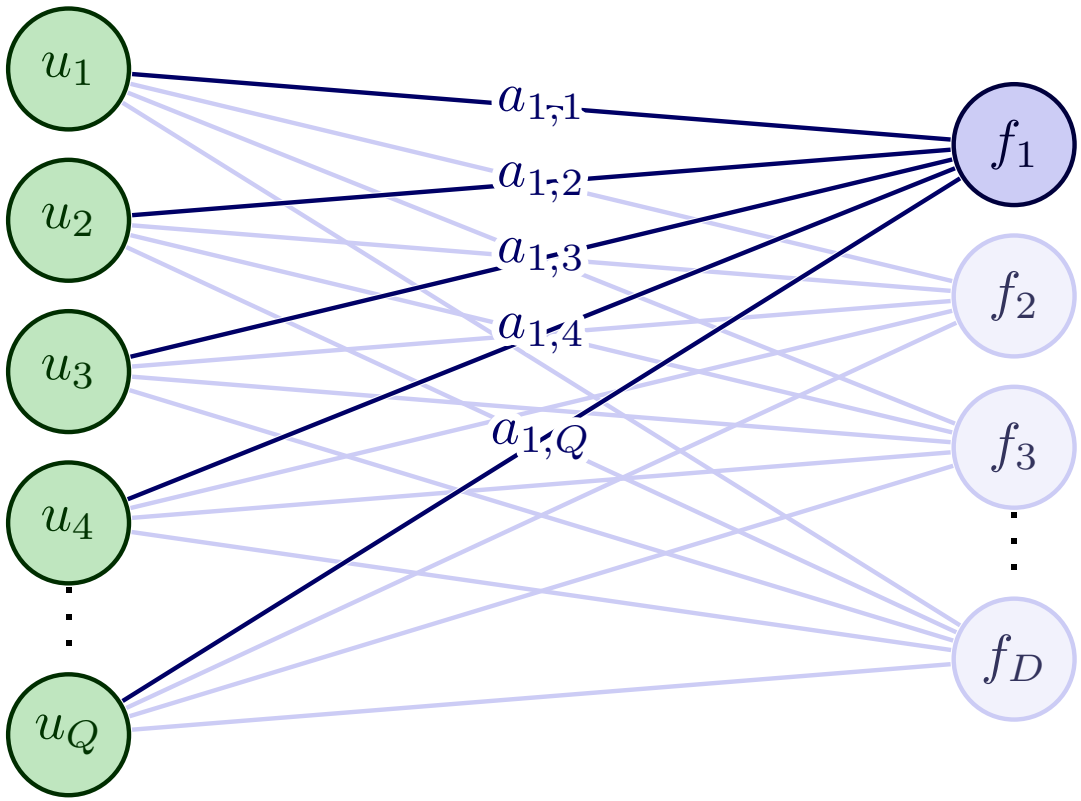


Figure: Graphical representation of the Linear Model of Coregionalization GP

Cross-covariance function

The cross-covariance function of the latent Gaussian process is expressed as:

$$
\begin{aligned}
k_{d,d'}(\mathbf{x}, \mathbf{x}') &= \text{cov}\{f_d(\mathbf{x}), f_{d'}(\mathbf{x}')\} \\
&= \sum_{q=1}^{Q} \sum_{q'=1}^{Q} a_{d,q} a_{d',q'} \text{cov}\{u_q(\mathbf{x}), u_{q'}(\mathbf{x}')\} \\
&= \sum_{q=1}^{Q} a_{d,q} a_{d',q} k_q(\mathbf{x}, \mathbf{x}') \\
&= \sum_{q=1}^{Q} b^q_{d,d'} k_q(\mathbf{x}, \mathbf{x}')
\end{aligned}
$$

Here, $b^q_{d,d'} = a_{d,q} a_{d',q}$ captures the interactions among the outputs induced by the q-th independent process, while $k_q(\mathbf{x}, \mathbf{x}')$ characterizes the interaction among input spaces viewed from the perspective of the q-th independent process.

The Matrix Kernel Function

Instead of a scalar kernel function, we now have a matrix kernel function $\mathbf{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{D \times D}$ with elements $k_{d,d'}(\mathbf{x}, \mathbf{x}')$:

$$\mathbf{k}(\mathbf{x}, \mathbf{x}') = \sum_{q=1}^{Q} \mathbf{B}_q k_q(\mathbf{x}, \mathbf{x}')$$

Here, each $\mathbf{B}_q \in \mathbb{R}^{D \times D}$ is referred to as the coregionalization matrix, with elements $(\mathbf{B}_q)_{d,d'} = b^q_{d,d'}$. Evaluating this at all test points X_* allows us to recover the covariance matrix as follows:

$$\mathbf{K}_{**} = \sum_{q=1}^{Q} \mathbf{B}_q \otimes K_{q**}$$

where $\otimes$ denotes the Kronecker product between matrices.

Linear Model of Coregionalization GP (LMCGP)

This model can effectively capture the cross-covariances of the output given by $k_{d,d'}(\mathbf{x}, \mathbf{x}')$, allowing to fill zeros in covariance matrix of IGP as follows:

$$\begin{bmatrix} \mathbf{f}_{1*} \\ \mathbf{f}_{2*} \\ \vdots \\ \mathbf{f}_{D*} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0} \ , \sum_{q=1}^{Q} \begin{bmatrix} b_{1,1}^q & b_{1,2}^q & \cdots & b_{1,D}^q \\ b_{2,1}^q & b_{2,2}^q & \cdots & b_{2,D}^q \\ \vdots & \vdots & \ddots & \vdots \\ b_{D,1}^q & b_{D,2}^q & \cdots & b_{D,D}^q \end{bmatrix} \otimes K_{q**} \right)$$

We call this model Linear Model of Coregionalization Gaussian Process (LMCGP).

Variational Inference and ELBO

We can extent our variational inference to include the independent set. Furthermore, instead of utilizing the latent processes f_d, we utilize the inducing variables derived from the independent processes u_q providing the following ELBO:

$$\mathcal{L} = \sum_{d=1}^{D} \sum_{n=1}^{N} \mathbb{E}_{q(f_{dn})}\{\log p(y_{dn} \mid f_{dn})\} - \sum_{q=1}^{Q} \mathsf{KL}\{q(\mathbf{u}_q) \parallel p(\mathbf{u}_q)\}$$

Latent posterior and Predictive distribution

The posterior over test points X_*, denoted as $p(\mathbf{f}_* \mid \mathbf{y})$, is approximated as follows:

$$p(\mathbf{f}_* \mid \mathbf{y}) \approx q(\mathbf{f}_*) = \int p(\mathbf{f}_* \mid \mathbf{u}) q(\mathbf{u}) d\mathbf{u}$$

We add Gaussian noise σ^2_{Nd} to the corresponding task into the above random vector to obtain the predictive distribution.

Adam + Natural Gradient Optimization

Optimization Parameters for LMCGP:

- **Kernel:** lengthscales, output scales
- **Likelihood:** data noise $\{\sigma^2_{Nd}\}_{d=1}^{D}$
- **Inducing Points:** Z
- **Variational Parameters:** $\{\mu_{\mathbf{q}}, S_q\}_{q=1}^{Q}$

Challenges: Strong dependency between variational parameters and others makes the model sensitive. ELBO loss function is non-convex, leading to poor local minima with traditional optimizers.

Solution: Combine Natural Gradient (NG) with Adam. NG optimizes variational parameters, while Adam optimizes other parameters.

Benefits: This hybrid method, Adam + NG, improves optimization performance, allowing better convergence.

Model Setup

The proposed methodology constructs the LMCGP covariance function using the widely applied squared exponential kernel:

$$k_q\left(\mathbf{x}, \mathbf{x}' \mid \Theta_q\right) = \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{x}')^\top \Theta_q^{-2}(\mathbf{x}-\mathbf{x}')\right)$$

Upon examining, one might question the absence of an output scale parameter. However, a detailed look at shows that the elements in the matrix $\mathbf{B}_q$ perform the function of rescaling the exponential term in the k_q kernel.

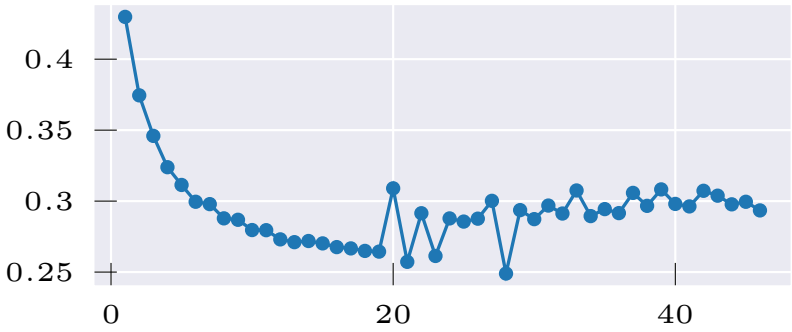
The trainable covariance parameters are $\{\mathbf{B}_q, \Theta_q\}_{q=1}^{Q}$.

Negative Log Predictive Density

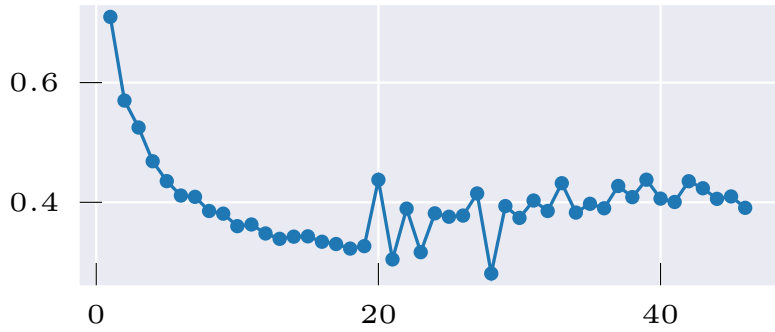
To evaluate the models' performance, we use the implemented metrics MSE, MSLL, and CRPS. Additionally, we propose a new metric called Negative Log Predictive Density (NLPD), which applies to any type of predictive distribution and is defined as:

$$\begin{aligned}\text{NLPD} &= -\frac{1}{N_*}\log p(\mathbf{y}_* \mid \mathbf{y}) \\ &= -\frac{1}{N_*}\sum_{n=1}^{N_*}\log p(\mathbf{y}_{n*} \mid \mathbf{y})\end{aligned}$$

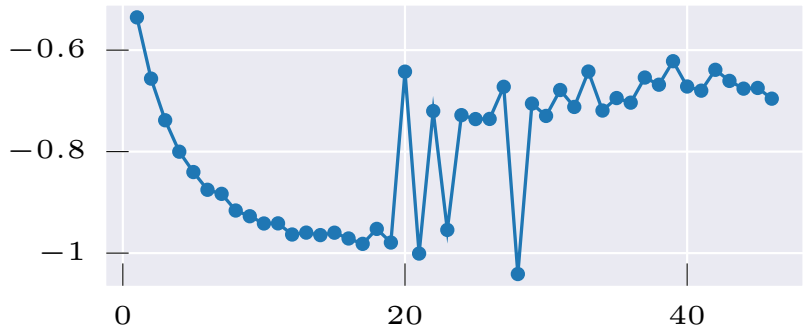
Tuning Q



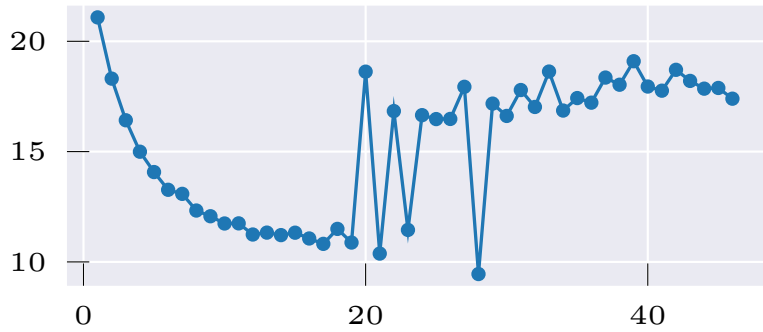
(a) CRPS



(b) MSE



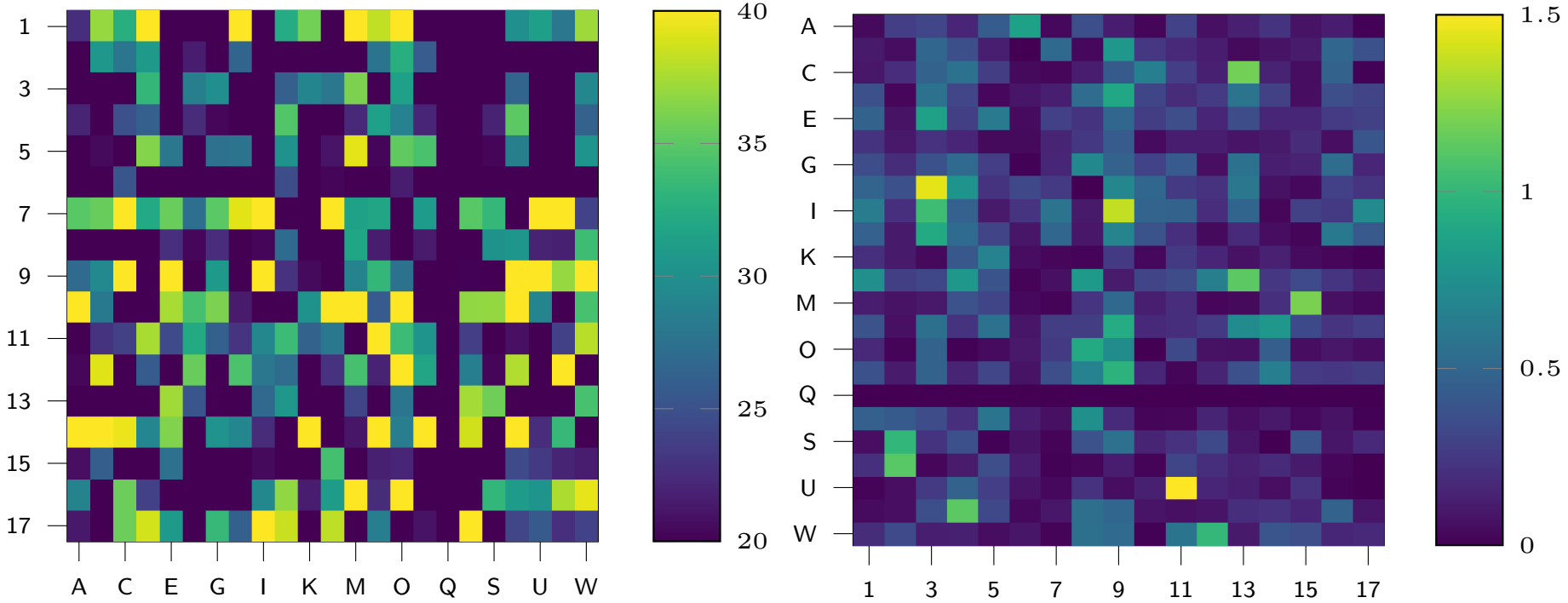
(c) MSLL



(d) NLPD

Figure: Performance metrics for LMCGP models as a function of the number of independent GPs. We finally select $Q = 17$ as the proper parameter

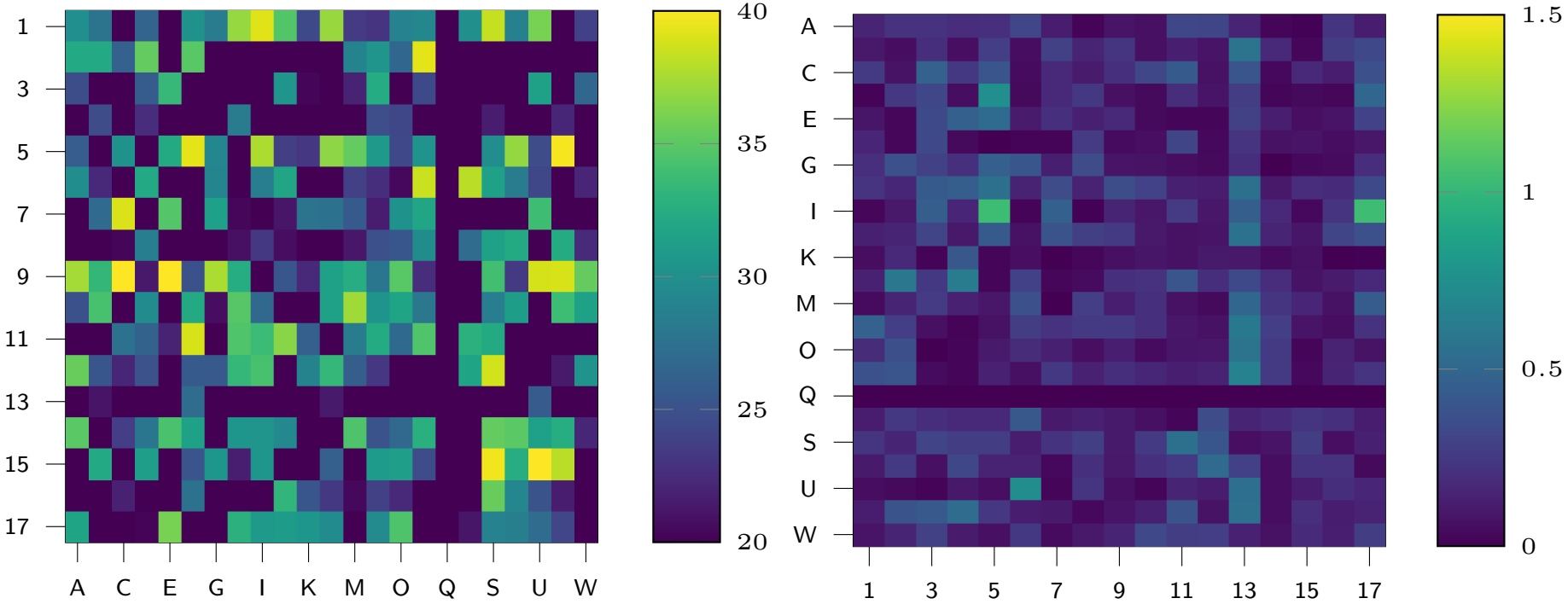
Lengthscale and $a_{d,q}$ Values (H=1)



(a) (b)

Figure: Lengthscale values of Input Features for Each Independent GP Model (left) and coefficients $a_{d,q}$ (right) for horizon $H = 1$.

Lengthscale and $a_{d,q}$ Values (H=30)

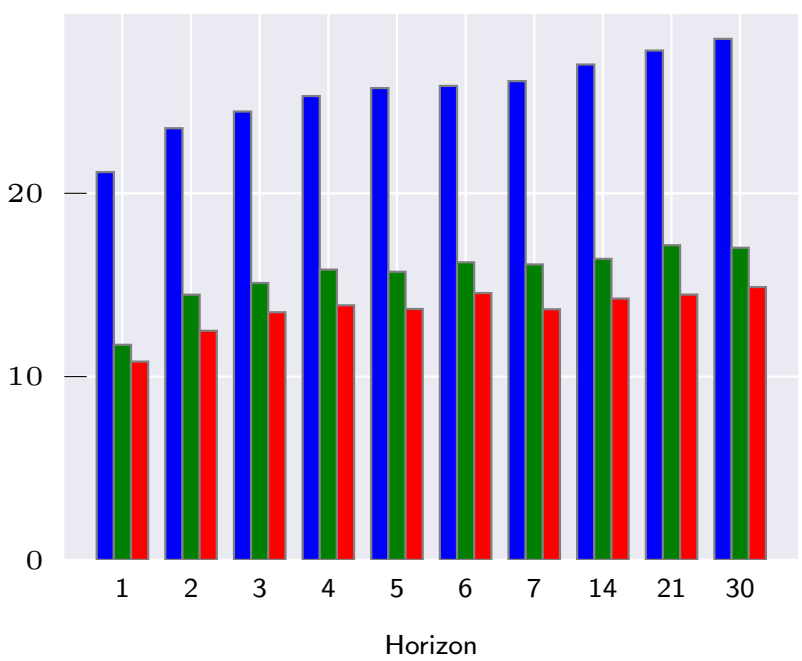


(a) (b)

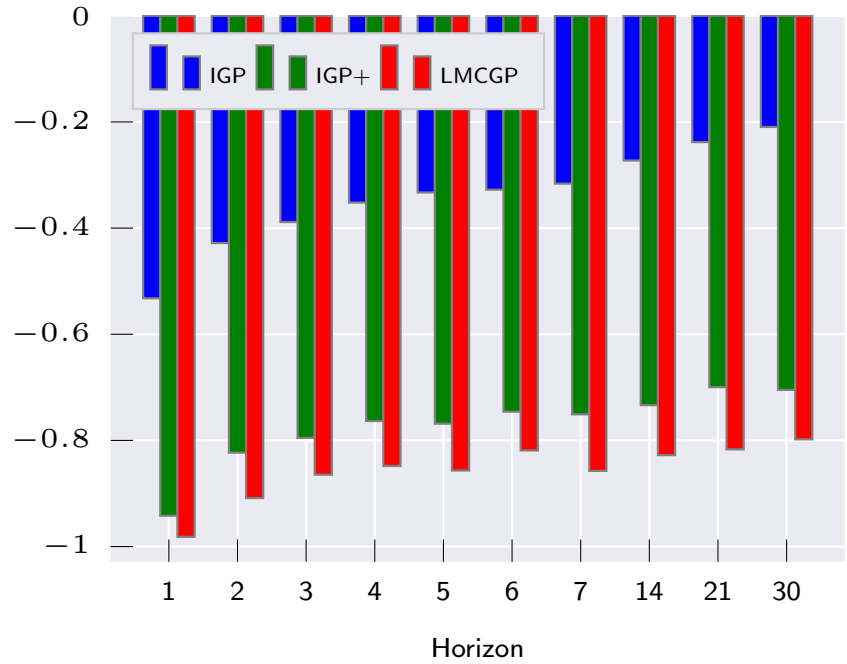
Figure: Lengthscale values of Input Features for Each Independent GP Model (left) and coefficients $a_{d,q}$ (right) for horizon $H = 30$.

LMCGP vs IGP+ vs IGP

The performance analysis compares LMCGP against two multi-output GP models: the IGP implemented, trained using only Adam optimizer, and another IGP trained into Adam + NG framework, called here IGP+.



(a) NLPD



(b) MSLL

Figure: Bar plots comparing the performance of LMCGP, IGP+, and IGP models for different H values.

Model Forecasting

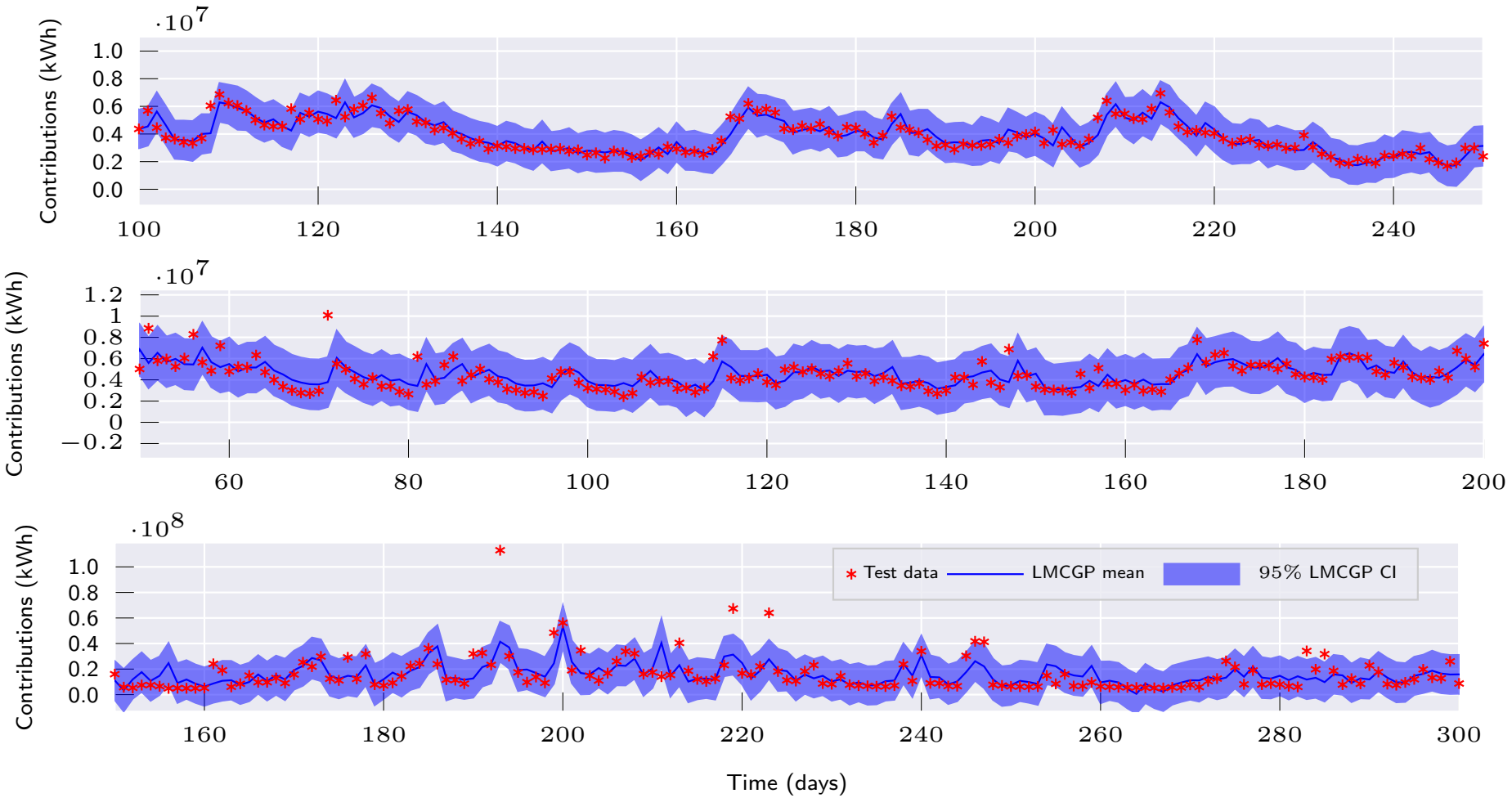


Figure: Test data for reservoirs T, A, and I (top to bottom) in one day ahead LMCGP model prediction ($H = 1$). Blue shade areas represent the 95% centered confidence interval for LMCGP prediction.

To Conclude

- The LMCGP effectively captures shared features and dynamics for multi-output tasks. However, increasing the number of independent GPs beyond a threshold leads to instability.
- The lengthscale matrix and task dependency coefficients, $a_{d,q}$, provide critical insights into feature selection, with some GPs specializing in specific tasks and others covering a broader range of outputs.
- To improve optimization performance, using Adam + NG optimizer proved superior to traditional methods, leading to more robust results.
- The LMCGP outperformed the IGP in terms of NLPD and MSLL across all horizons, emphasizing the benefits of task dependency modeling.
- The LMCGP's forecasting ability showed stronger learning of complex patterns by leveraging data from multiple tasks.

Chained Correlated Gaussian Processes

Likelihood Model

We assume the outputs are conditionally independent given a parameter vector $\boldsymbol{\theta}_d \in \mathbb{R}^{J_d}$, where:

- J_d is the number of parameters that define the likelihood for the d-th output.
- $\boldsymbol{\theta}_d = [\theta_{d,1}, \theta_{d,2}, \ldots, \theta_{d,J_d}]^\top$.

The complete parameter vector across all outputs is denoted as:

$$\boldsymbol{\theta} = [\boldsymbol{\theta}_1^\top, \boldsymbol{\theta}_2^\top, \ldots, \boldsymbol{\theta}_D^\top]^\top \in \mathbb{R}^J$$

where $J = \sum_{d=1}^{D} J_d$.
The likelihood of observing all output realizations, $\mathbf{y}$, given the parameter vector $\boldsymbol{\theta}$, is expressed as the product of individual likelihoods:

$$p(\mathbf{y} \mid \boldsymbol{\theta}) = \prod_{d=1}^{D} p\left(\mathbf{y}_d \mid \boldsymbol{\theta}_d\right)$$

Chained Gaussian Process

Each j-th parameter of the d-th likelihood distribution $\theta_{d,j}$ is a deterministic transformation of a latent Gaussian Process prior realization f_{dj}, given by $\theta_{d,j} = g_{d,j}(f_{d,j})$.

- $\mathbf{f}_{d,j} = [f_{d,j,1}, f_{d,j,2}, \cdots, f_{d,j,N}]^\top \in \mathbb{R}^N$, where $f_{d,j,n} = f_{d,j}(\mathbf{x}_n)$.
- $\mathbf{f}_d = [\mathbf{f}_{d,1}^\top, \mathbf{f}_{d,2}^\top, \cdots, \mathbf{f}_{d,J_d}^\top]^\top \in \mathbb{R}^{J_d N}$
- $\mathbf{f} = [\mathbf{f}_1^\top, \mathbf{f}_2^\top, \cdots, \mathbf{f}_D^\top]^\top \in \mathbb{R}^{JN}$

The conditionally independent likelihood is then formulated as follows:

$$p(\mathbf{y} \mid \boldsymbol{\theta}) = p(\mathbf{y} \mid \mathbf{f}) = \prod_{d=1}^{D} p(\mathbf{y}_d \mid \mathbf{f}_d) = \prod_{d=1}^{D} \prod_{n=1}^{N} p(y_{d,n} \mid f_{d,1,n}, \cdots, f_{d,J_d,n})$$

This formulation introduces J latent parameter functions $f_{d,j}(\mathbf{x})$, each governed by a GP prior.

LMCGP for Chained GPs

The correlation between $f_{d,j}(\mathbf{x})$ and $f_{d',j'}(\mathbf{x}')$ can be modeled by using the LMCGP framework:

$$f_{d,j}(\mathbf{x}) = \sum_{q=1}^{Q} a_{d,j,q} u_q(\mathbf{x}) \qquad u_q(\mathbf{x}) \sim \mathcal{GP}(0, k_q(\mathbf{x}, \mathbf{x}'))$$

Where k_q is the kernel function of the independent process q government by the parameters set Φ_q. The cross-covariance function of the latent parameter GP $\mathbf{f}$ is as follows:

$$k_{(d,j),(d',j')}(\mathbf{x}, \mathbf{x}') = \sum_{q=1}^{Q} b^q_{(d,j),(d',j')} k_q(\mathbf{x}, \mathbf{x}')$$

where $b^q_{(d,j),(d',j')} = a_{d,j,q} a_{d',j',q}$ encodes the interactions among the outputs models, meanwhile, $k_q(\mathbf{x}, \mathbf{x}')$ mange the interaction among input space.

Variational Inference and ELBO

We can extent our variational inference to include conditional independent likelihood function, providing the following ELBO:

$$\mathcal{L} = \sum_{d=1}^{D} \sum_{n=1}^{N} \mathbb{E}_{q(f_{d,1,n}),\cdots,q(f_{d,J_d,n})} \left\{ \log p\left(y_{d,n} \mid f_{d,1,n}, \cdots, f_{d,J_d,n}\right) \right\} - \sum_{q=1}^{Q} \mathsf{KL}\left\{ q(\mathbf{u}_q) \parallel p(\mathbf{u}_q) \right\}$$

The expectation values can be approximated via Monte Carlo methods.

Latent posterior and Predictive distribution

Consider a test points set $X_* \in \mathcal{X}$. Assuming a good approximation of the variational posterior $p(\mathbf{u} \mid \mathbf{y}) \approx q(\mathbf{u})$, the posterior of latent parameter function vector at test points $\mathbf{f}_*$ is

$$q(\mathbf{f}_*) = \int p(\mathbf{f}_* \mid \mathbf{u}) q(\mathbf{u}) d\mathbf{u}$$

The predictive distribution for a new observation $\mathbf{y}_*$, given the observed data $\mathbf{y}$, can thus be approximated as:

$$p(\mathbf{y}_* \mid \mathbf{y}) \approx \int p(\mathbf{y}_* \mid \mathbf{f}_*) q(\mathbf{f}_*) \, d\mathbf{f}_*,$$

which integrates over the latent function vector $\mathbf{f}_*$ to account for its uncertainty in the prediction of $\mathbf{y}_*$.

Model Setup

We again make use of squared exponential kernel to construct the covariance function and Adam + NG framework to train the models.

- **Gaussian Likelihood**

$$p(\mathbf{y} \mid \mathbf{f}) = \prod_{d=1}^{D} \prod_{n=1}^{N} \mathcal{N}\left(y_{d,n} \mid g_{d,1}(f_{d,1,n}), g_{d,2}(f_{d,2,n})\right)$$

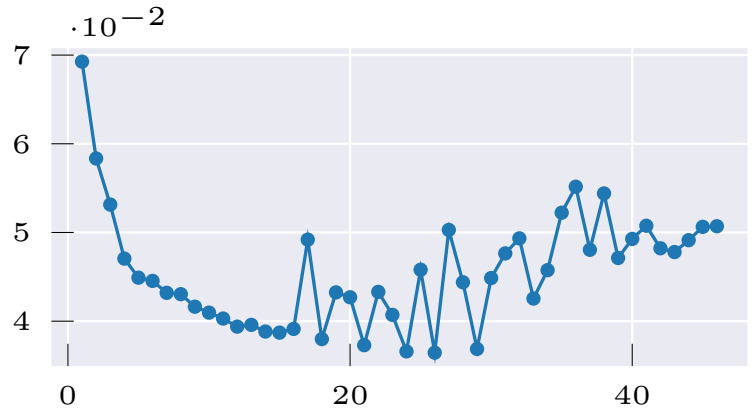
  In this formulation, $g_{d,1}(\cdot) = \cdot$, while $g_{d,2}(\cdot) = \ln(\exp(\cdot) + 1)$. We call this model Chd Normal.

- **Gamma Likelihood**

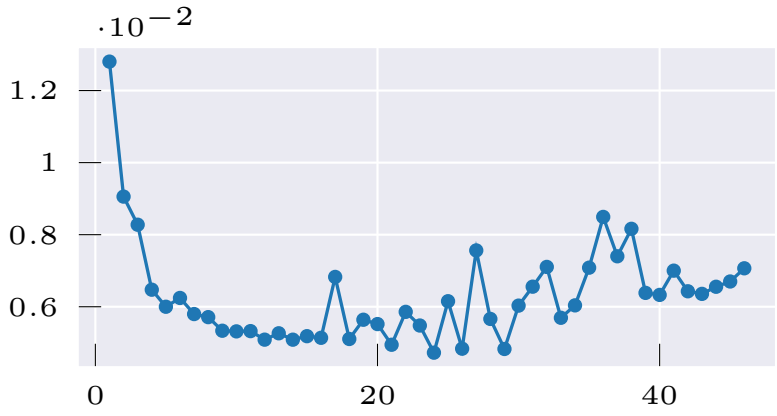
$$p(\mathbf{y} \mid \mathbf{f}) = \prod_{d=1}^{D} \prod_{n=1}^{N} \mathcal{G}amma\left(y_{d,n} \mid g_{d,1}(f_{d,1,n}), g_{d,2}(f_{d,2,n})\right)$$

  In this formulation $g_{d,1}(\cdot) = g_{d,2}(\cdot) = \ln(\exp(\cdot) + 1)$. We call this model Chd Gamma.

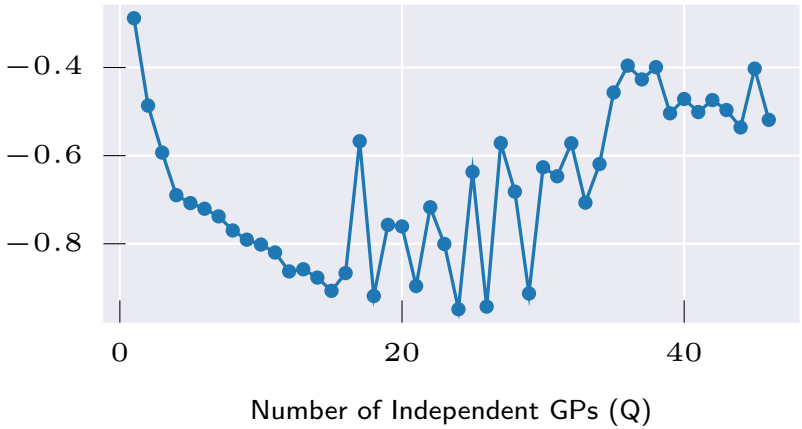
Tuning Q for Chd Normal



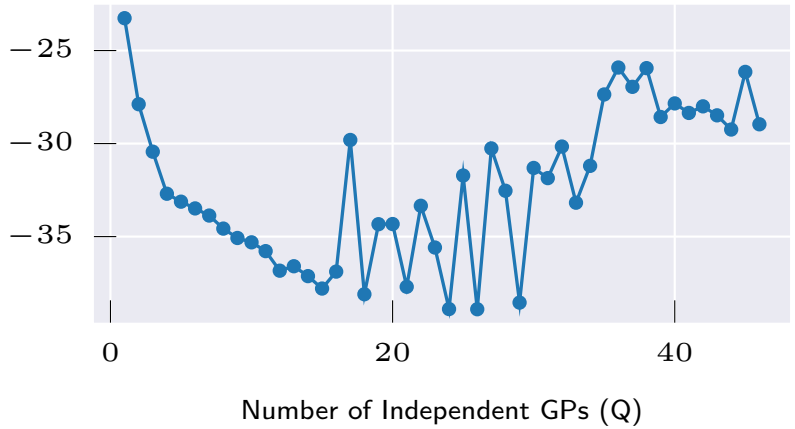
(a) CRPS



(b) MSE



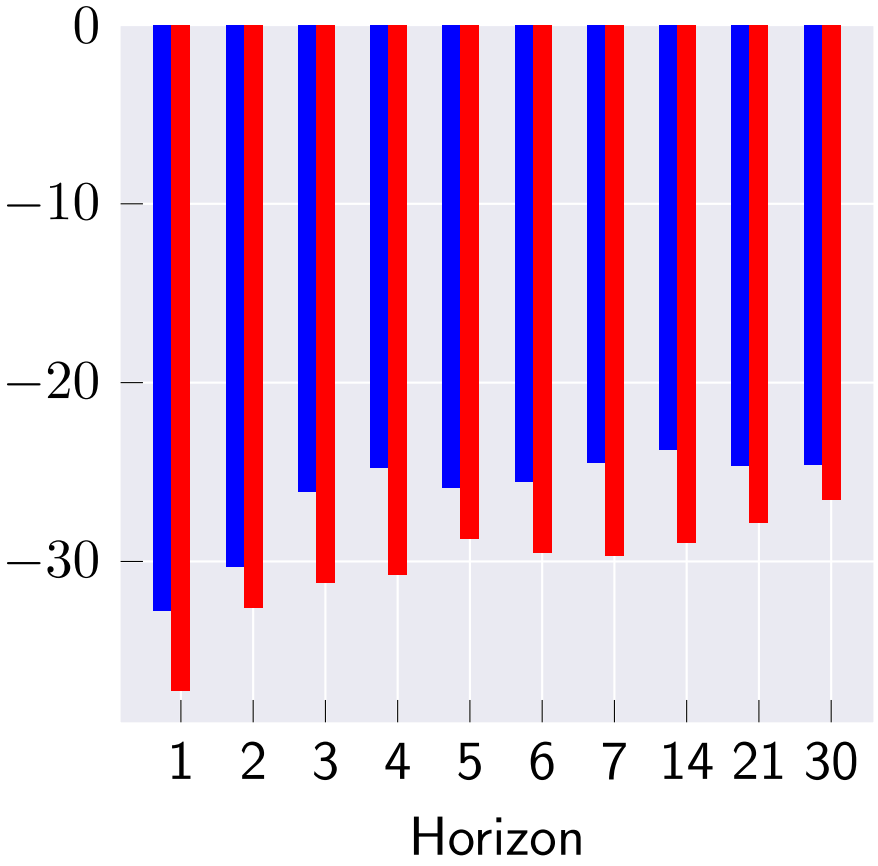
(c) MSLL



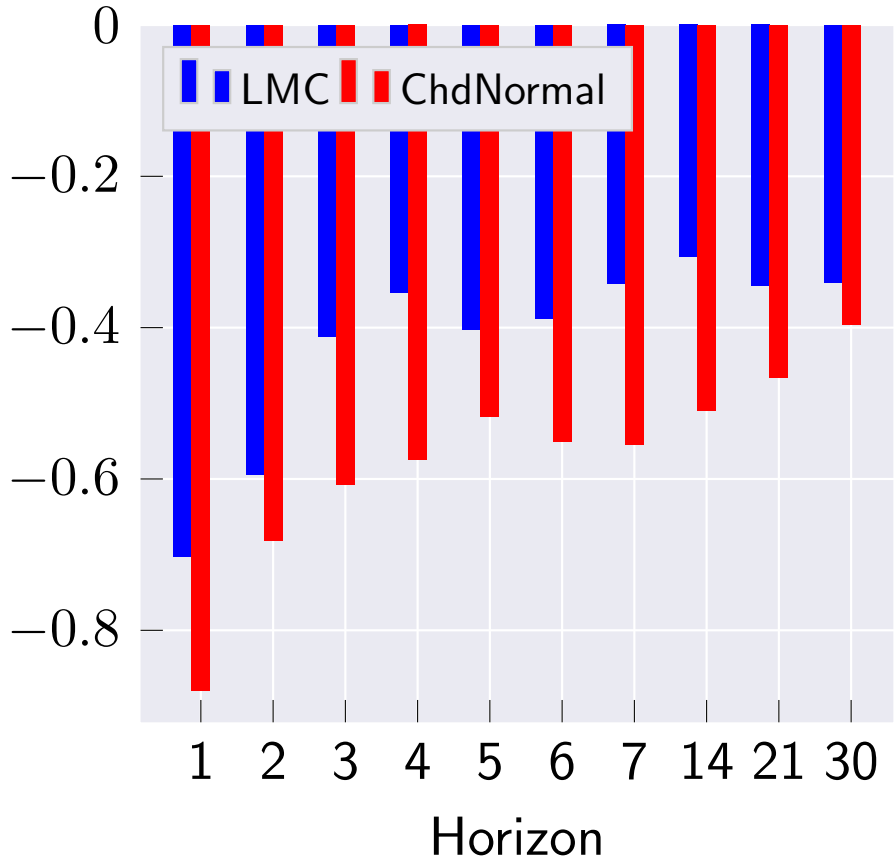
(d) NLPD

Figure: Performance metrics for ChdGP Normal model as a function of the number of independent GPs Q. We select $Q = 15$ as the optimal value.

ChdGP Normal vs LMCGP



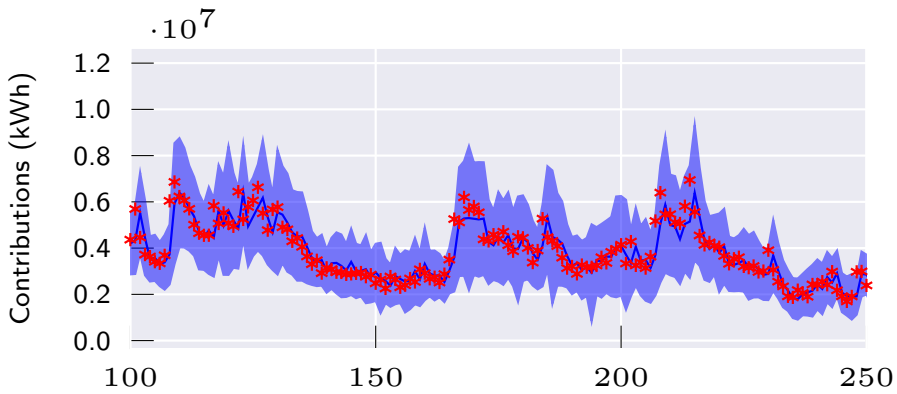
(a) NLPD



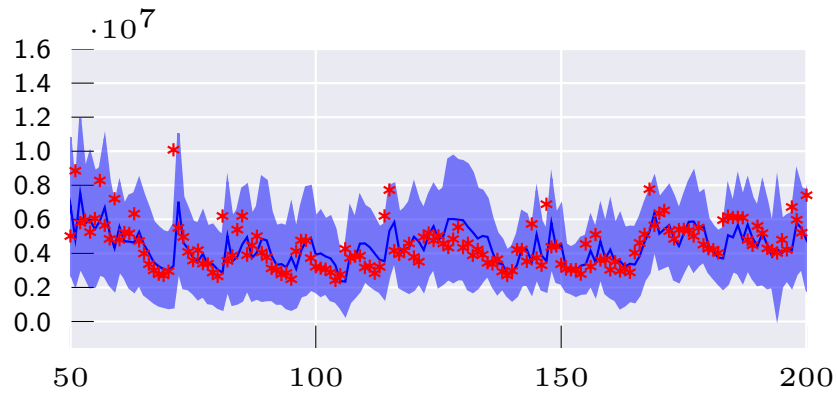
(b) MSLL

Figure: Bar plots comparing the performance of LMCGP, and ChdGP Normal models for different H values.

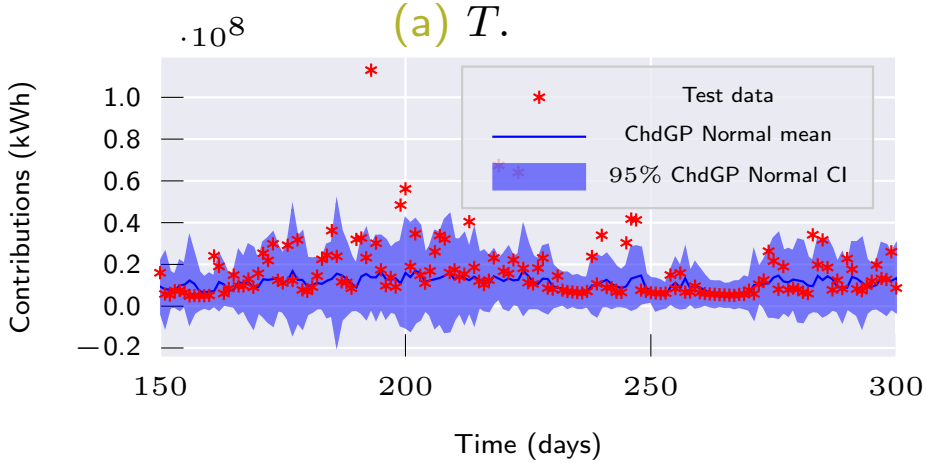
ChdGP Normal Forecasting



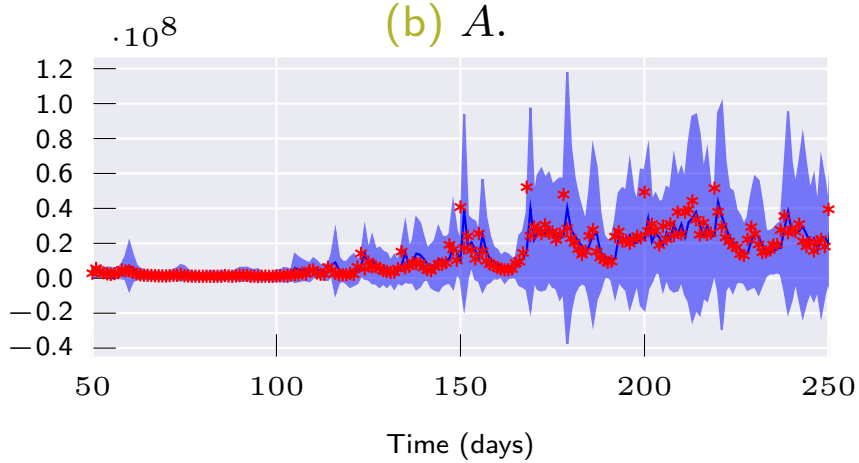
(a) T.



(b) A.



(c) I.



(d) O.

Figure: Test data for four reservoirs in one day ahead ChdGP Normal model prediction ($H = 1$). Blue shaded areas represent the 95% centered confidence interval for the model's prediction.

Tuning Q for Chd Gamma

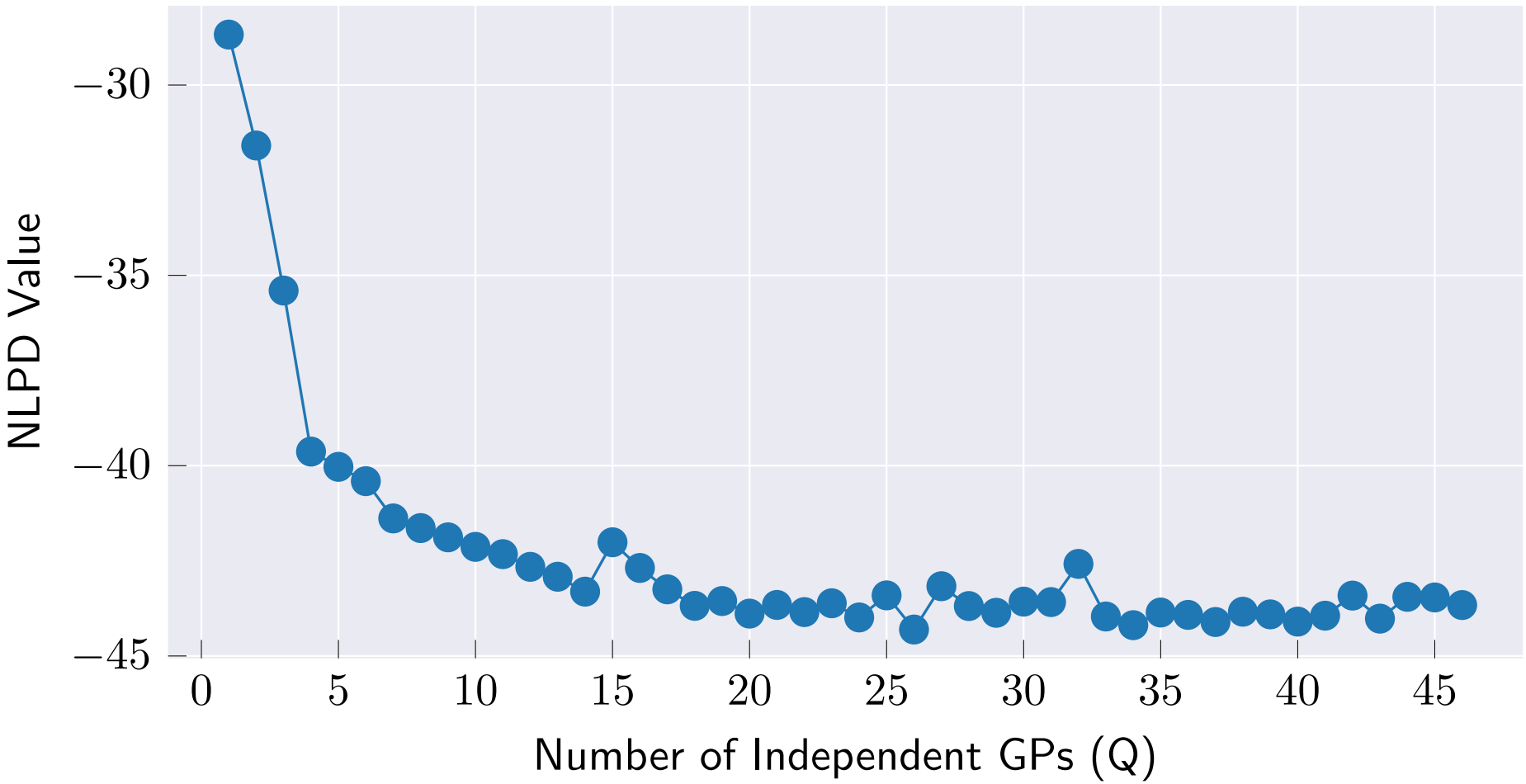


Figure: NLPD metric for ChdGP Gamma models as a function of the number of independent GPs. We select $Q = 26$ as the optimal value.

ChdnGP Gamma vs ChdGP Normal

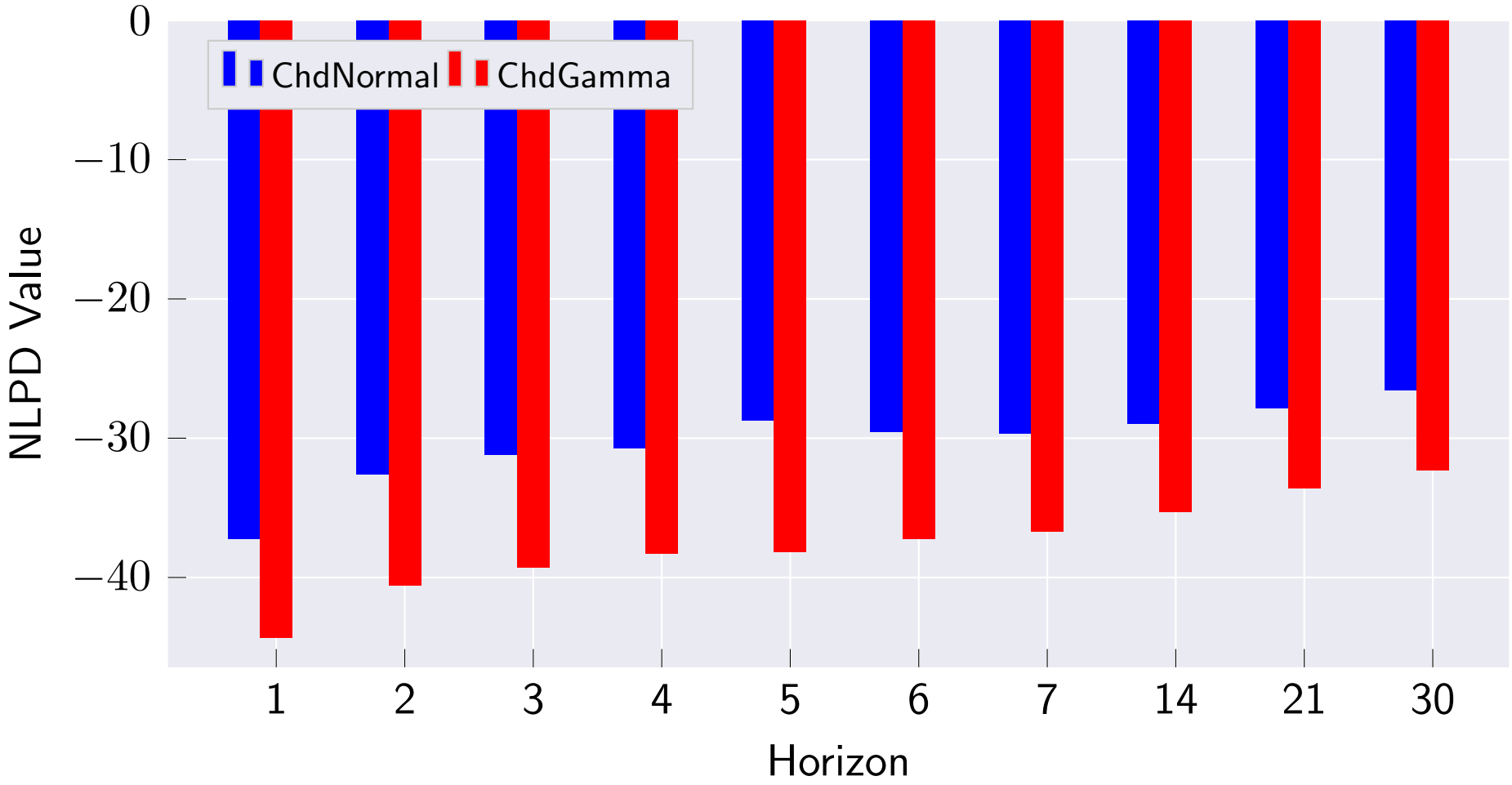
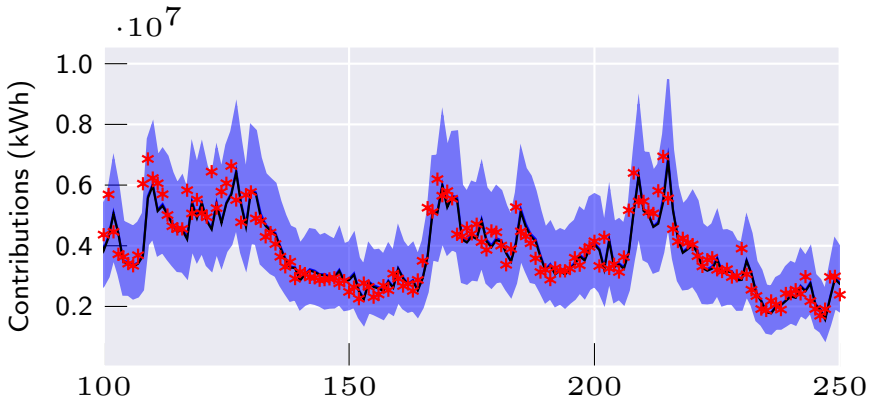
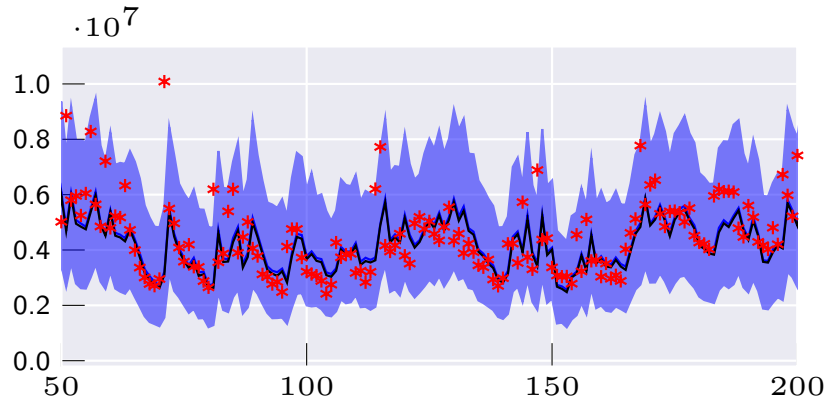


Figure: Comparison of NLPD metric across different prediction horizons for the ChdGP Normal, and ChdGP Gamma models.

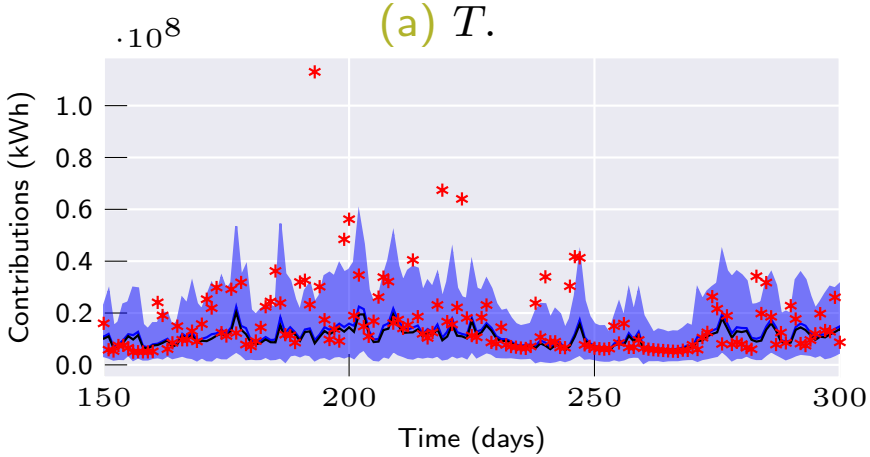
ChdGP Gamma Forecasting



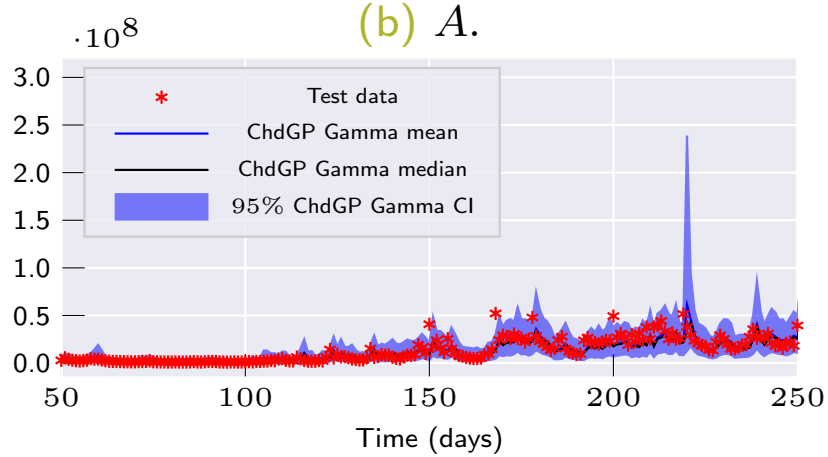
(a) T.



(b) A.



(c) I.



(d) O.

Figure: Test data for four reservoirs in one day ahead ChdGP Normal model prediction. Blue shaded areas represent the 95% centered confidence interval for the model's prediction.

To Conclude

- The ChdGP model generalizes all previously developed GP-based models, enhancing expressiveness by modeling likelihood parameters and enabling the handling of natural output restrictions.
- The ChdGP Normal model outperformed the LMCGP model across all forecasting horizons, primarily due to its ability to adaptively vary data noise over the input space, providing a more refined capture of the underlying data structure.
- The ChdGP with Gamma likelihood ensured non-negative predictions. The tuning process revealed a significant improvement in model stability as the number of independent GPs (Q) increased, suggesting superior data modeling capabilities.
- The Gamma likelihood configuration outperformed the Gaussian likelihood across all evaluated horizons by avoiding the allocation of predictive distribution mass to negative values and utilizing an asymmetric distribution to more effectively handle peak outliers.

[1] Zaher Mundher Yaseen, Mohammed Falah Allawi, Ali A. Yousif, Othman Jaafar, Firdaus Mohamad Hamzah, and Ahmed El-Shafie. Non-tuned machine learning approach for hydrological time series forecasting. *Neural Computing and Applications*, 30(5):1479–1491, 2018.

[2] Mahsa H. Kashani, Mohammad Ali Ghorbani, Yagob Dinpashoh, and Sedaghat Shahmorad. Integration of volterra model with artificial neural networks for rainfall-runoff simulation in forested catchment of northern iran. *Journal of Hydrology*, 540:340–354, 2016.

[3] Muhammed Sit, Bekir Z. Demiray, Zhongrun Xiang, Gregory J. Ewing, Yusuf Sermet, and Ibrahim Demir. A comprehensive review of deep learning applications in hydrology and water resources. *Water Science and Technology*, 82(12):2635–2670, 08 2020.

[4] Sajjad Abdollahi, Jalil Raeisi, Mohammadreza Khalilianpour, Farshad Ahmadi, and Ozgur Kisi. Daily mean streamflow prediction in perennial and non-perennial rivers using four data driven techniques. *Water Resources Management*, 31(15):4855–4874, 2017.

[5] Jenq-Tzong Shiau and Hui-Ting Hsu. Suitability of ann-based daily streamflow extension models: a case study of gaoping river basin, taiwan. *Water Resources Management*, 30(4):1499–1513, 2016.

[6] Md. Munir Hayet Khan, Nur Shazwani Muhammad, and Ahmed El-Shafie. Wavelet based hybrid ann-arima models for meteorological drought forecasting. *Journal of Hydrology*, 590:125380, 2020.

[7] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016.

[9] John Quilty and Jan Adamowski. A stochastic wavelet-based data-driven framework for forecasting uncertain multiscale hydrological and water resources processes. *Environmental Modelling & Software*, 130:104718, 2020.

[10] Wen jing Niu and Zhong kai Feng. Evaluating the performances of several artificial intelligence methods in forecasting daily streamflow time series for sustainable water resources management. *Sustainable Cities and Society*, 64:102562, 2021.

[11] Wessel P. Bruinsma, Eric Perim, Will Tebbutt, J. Scott Hosking, Arno Solin, and Richard E. Turner. Scalable exact inference in multi-output gaussian processes, 2020.

[12] Marc Verriere, Nicolas Schunck, Irene Kim, Petar Marević, Kevin Quinlan, Michelle N. NGo, David Regnier, and Raphael David Lasseri. Building surro-

gate models of nuclear density functional theory with gaussian processesand autoencoders, 2022.