

# 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 20, 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  $n - T + 1$ :

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

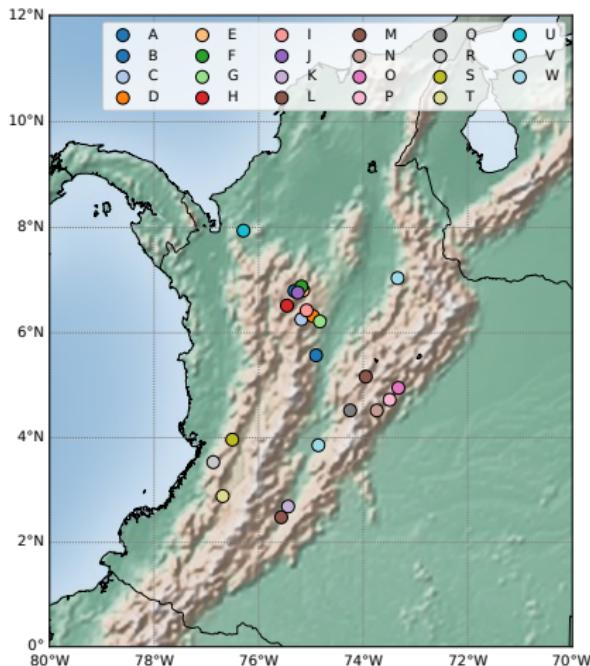
Here,  $T$  is the model order and  $H$  is the prediction horizon and  $\mathcal{X} \subset \mathbb{R}^{DT}$  represents the input space.

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

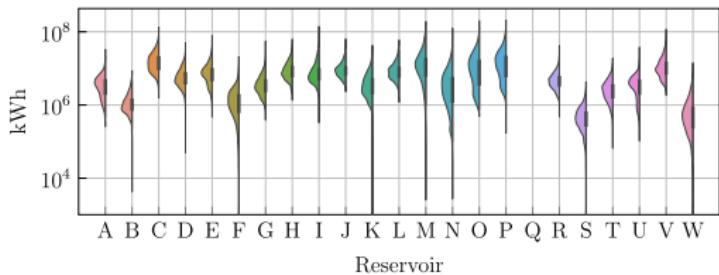
$$\mathbf{y}_n = \mathbf{v}_{n+H} \in \mathbb{R}^D$$

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  $f(\cdot)$  maps inputs  $x_n$  to outputs  $y_n$ . Adding i.i.d. Gaussian noise  $\epsilon$ , the model becomes:

$$y_n = f(x_n) + \epsilon$$

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  $\Sigma_\epsilon$  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  $\theta$  and the observation noise matrix  $\Sigma_\epsilon$ . 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:

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

However, the main challenge lies in the computational complexity of  $\mathcal{O}(N^3 D^3)$  and the storage demand of  $\mathcal{O}(N^2 D^2)$  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):

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

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  $\Sigma_\epsilon$  to  $q(\mathbf{f}_*) = \int p(\mathbf{f}_* | \mathbf{u})q(\mathbf{u})d\mathbf{u}$ .

# Model Setup

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

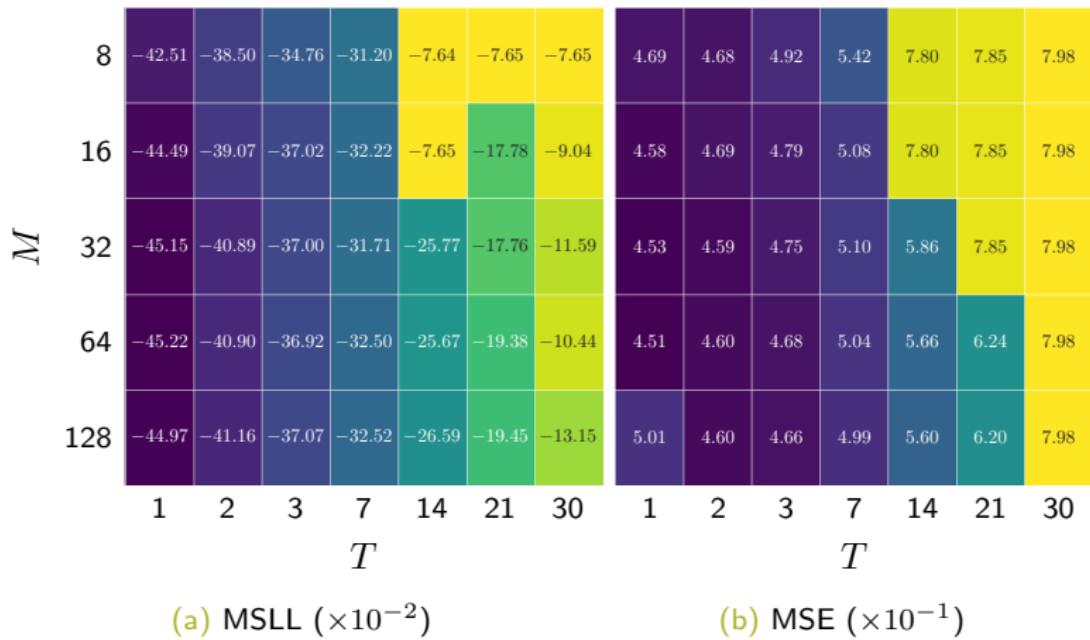
$$k((\mathbf{x}, d), (\mathbf{x}', d')) = k_{\mathcal{X}}(\mathbf{x}, \mathbf{x}' | \Theta_d) k_D(d, d' | \sigma_d),$$

with:

$$k_{\mathcal{X}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^\top \Theta_d^{-2}(\mathbf{x} - \mathbf{x}')\right),$$
$$k_D(d, d') = \sigma_d^2 \delta_{d,d'},$$

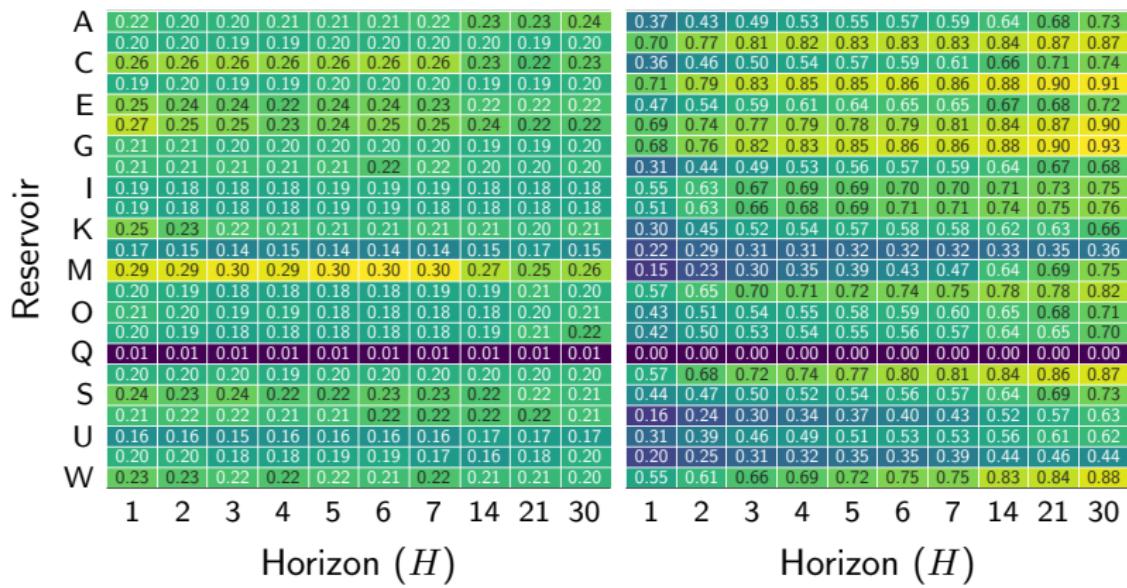
where  $\delta_{d,d'}$  is the Kronecker delta,  $\Theta_d$  is the lengthscale matrix, and  $\sigma_d^2$  is the output scale. This reduces complexity to  $\mathcal{O}(NM^2D)$  by avoiding explicit task correlations.

# Tuning $M$ and $T$



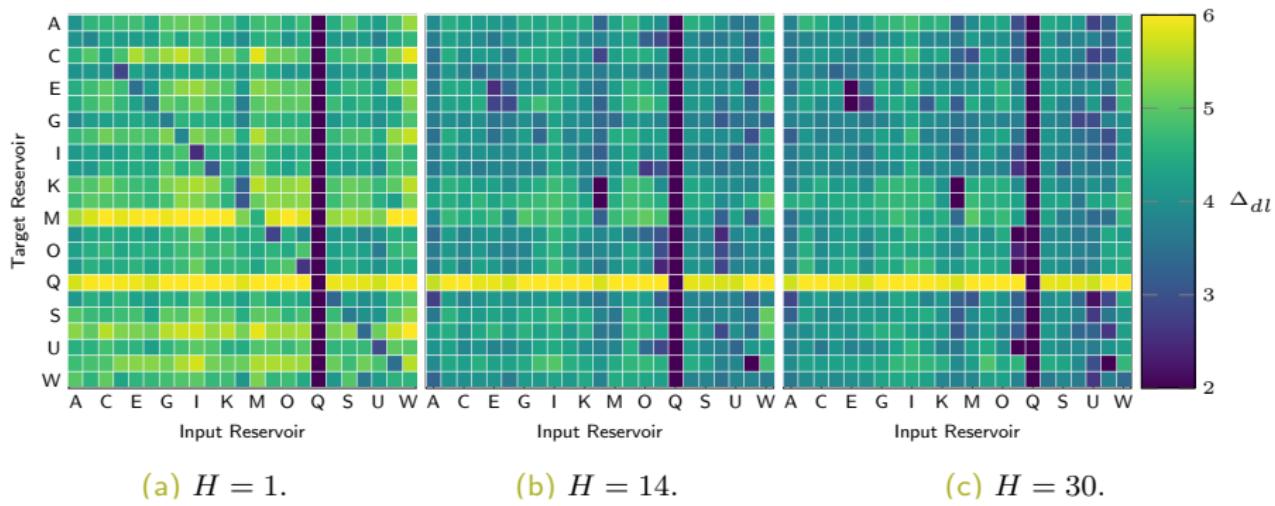
Grid search average values for tuning the model order  $T$  and the number of inducing points  $M$ . The optimal settings are  $M = 64$  and  $T = 1$

# Reservoir-Wise Output Scales and Noise Variance



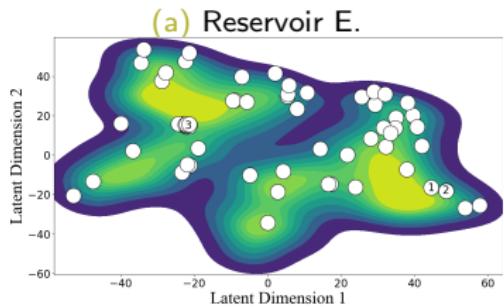
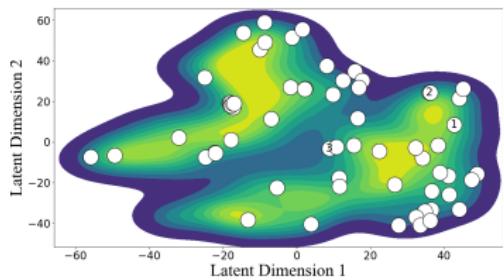
Output scales  $\sigma_d^2$  and noise variance  $\Sigma_\epsilon$  tuned for each horizon and reservoir.  
Longer horizons generally show smaller output scales and higher noise variance.

# Lengthscale Analysis

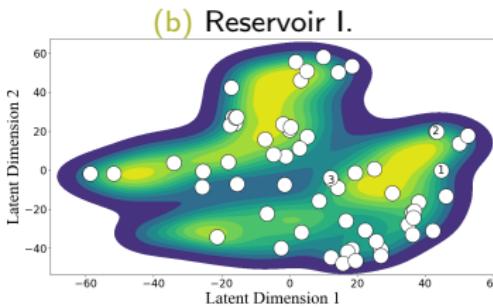
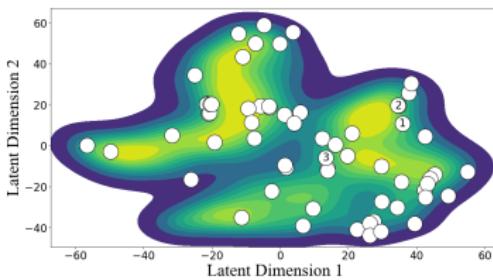


Trained lengthscales from input features (columns) to output tasks (rows) for three prediction horizons. As the horizon increases, main diagonal lengthscales lose relevance, while off-diagonal ones gain importance.

# t-distributed Stochastic Neighbor Embedding (t-SNE)



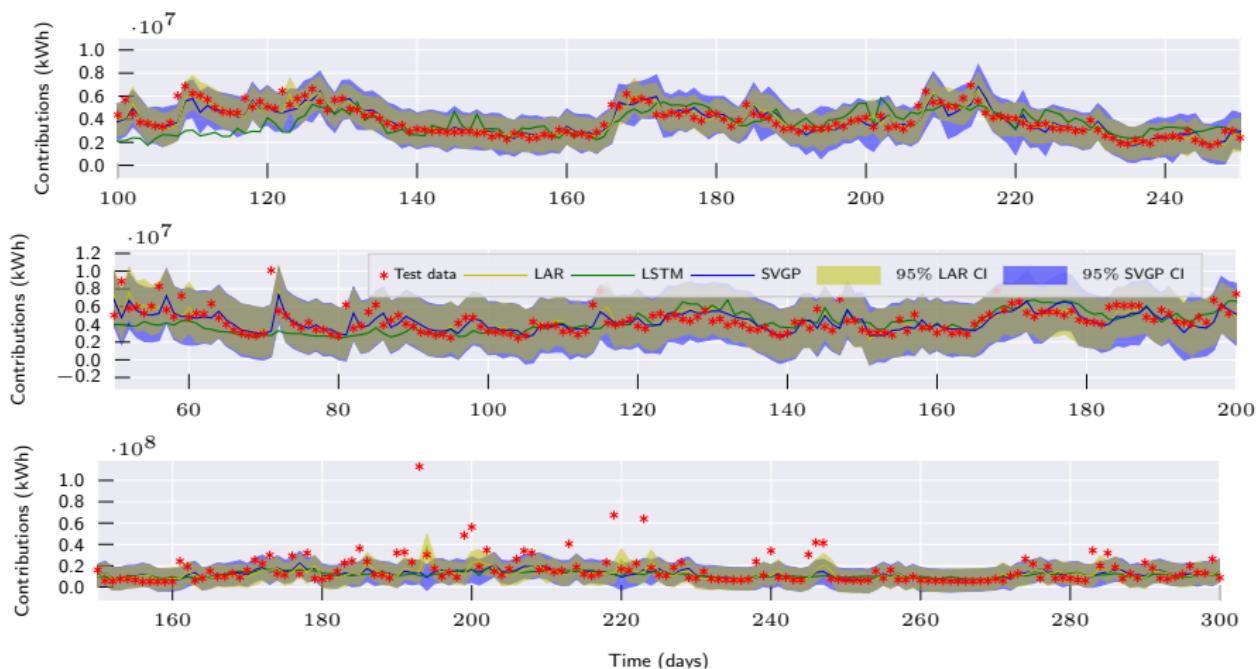
(c) Reservoir L.



(d) Reservoir U.

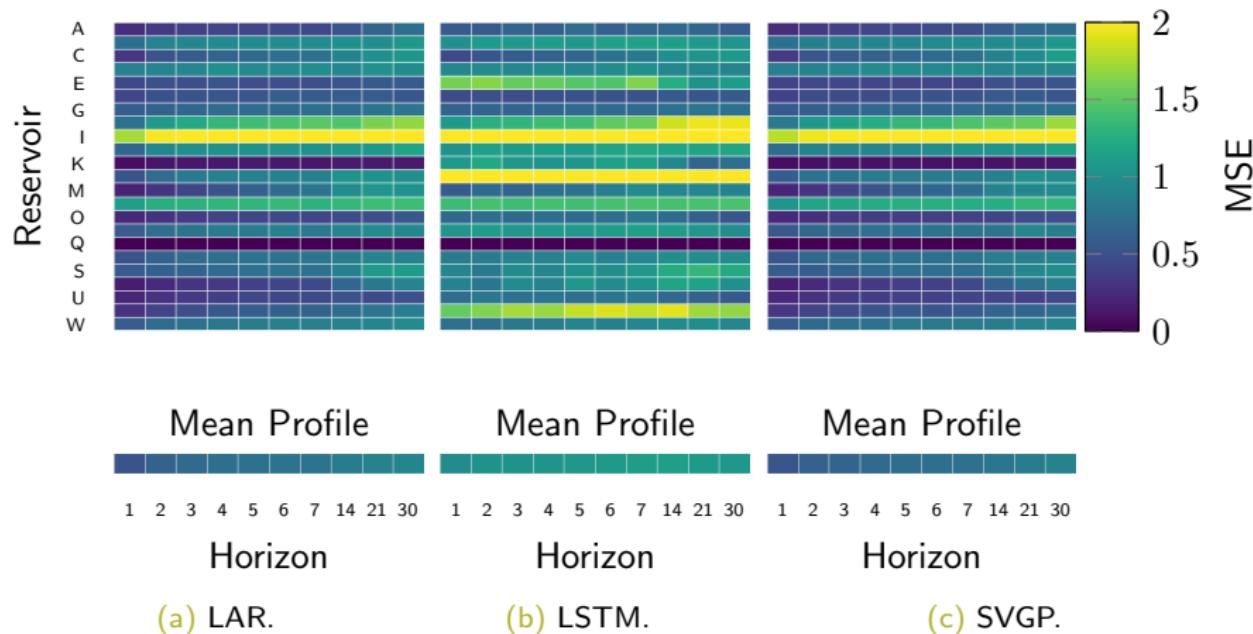
t-SNE-based 2D mapping of the SVGP latent space and inducing points' locations for four target reservoirs. The shared inducing points allow for the capturing of task-wise, and global information about the streamflow dynamics.

# 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.

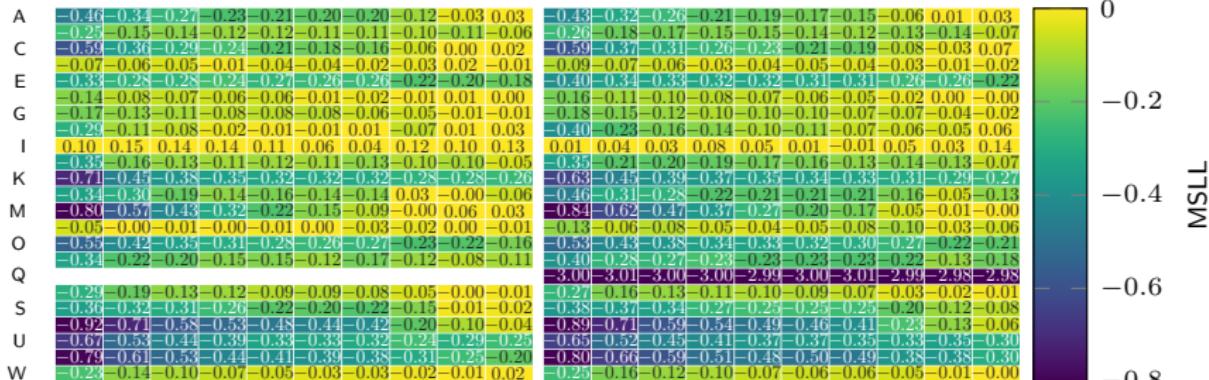
# MSE Scores



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

Reservoir



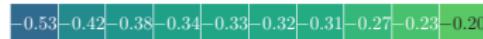
Mean Profile



1 2 3 4 5 6 7 14 21 30 1 2 3 4 5 6 7 14 21 30

Horizon

Mean Profile



1 2 3 4 5 6 7 14 21 30 1 2 3 4 5 6 7 14 21 30

Horizon

(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.

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

## 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

# Independent Gaussian Process (IGP)

## Independent Process

$$u_d(\mathbf{x}) \sim \mathcal{GP}(0, k_d(\mathbf{x}, \mathbf{x}'))$$



## Latent Process

$$f_d(\mathbf{x}) = u_d(\mathbf{x})$$



## Multi-Output Model

$$\mathbf{f}_* \sim \mathcal{N} \left( \mathbf{0}, \sum_{d=1}^D \mathbf{B}_d \otimes K_{d**} \right)$$



- $\mathbf{B}_d = (\delta_{d,d'}) \in \mathbb{R}^{D \times D}$ .
- $K_{d**} \in \mathbb{R}^{N_* \times N_*}$  is the  $d$ -th covariance matrix at  $X_*$  test inputs.
- $\otimes$  denotes the Kronecker product.



# Linear Model of Coregionalization GP (LMCGP)

## Independent Process

$$u_q(\mathbf{x}) \sim \mathcal{GP}(0, k_q(\mathbf{x}, \mathbf{x}'))$$

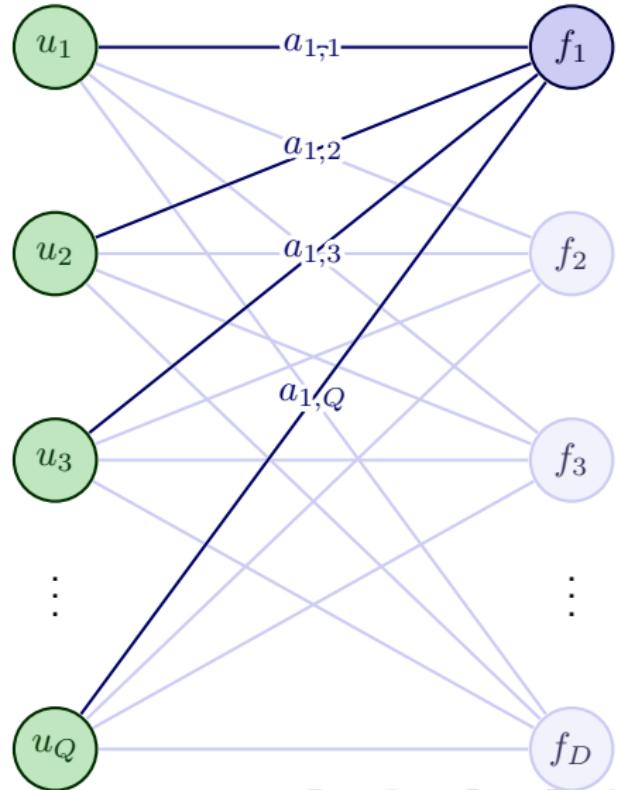
## Latent Process

$$f_d(\mathbf{x}) = \sum_{q=1}^Q a_{d,q} u_q(\mathbf{x})$$

## Multi-Output Model

$$\mathbf{f}_* \sim \mathcal{N} \left( \mathbf{0}, \sum_{q=1}^Q \mathbf{B}_q \otimes K_{q**} \right)$$

- $\mathbf{B}_q = (a_{d,q} a_{d',q}) \in \mathbb{R}^{D \times D}$  is the  $q$ -th coregionalization matrix.
- $K_{q**} \in \mathbb{R}^{N_* \times N_*}$ .



# Variational Inference, ELBO, and Predictive Distribution

We extend variational inference to include the independent set, utilizing the inducing variables  $u_q$  derived from independent processes. The ELBO is given by:

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

The posterior over test points  $X_*$ ,  $p(\mathbf{f}_* | \mathbf{y})$ , is given by:

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

Gaussian noise  $\sigma_{Nd}^2$  is added to obtain the predictive distribution.

# Model Setup

## Covariance Function (LMCGP)

The LMCGP model uses a squared exponential kernel:

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

Here,  $\Theta_q$  is the lengthscale matrix, and  $\mathbf{B}_q$  works as outputscale.

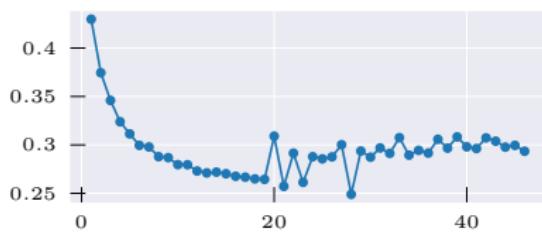
## Optimization and Model Variants

Strong dependencies between parameters may cause poor local minima [12]. We address this by combining Natural Gradient (NG) to optimize variational parameters, and Adam for the rest [13].

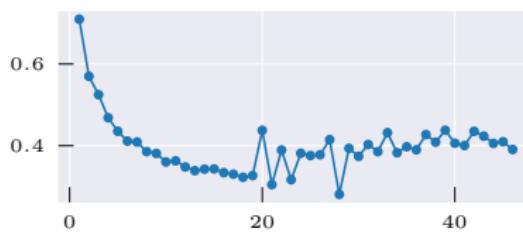
### Variants:

- IGP: Independent GP (Adam).
- IGP+: Independent GP (Adam+NG).
- LMCGP: Correlated GP (Adam+NG)

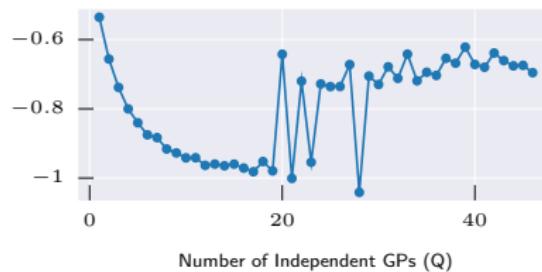
# Tuning Q



(a) CRPS



(b) MSE

Number of Independent GPs ( $Q$ )

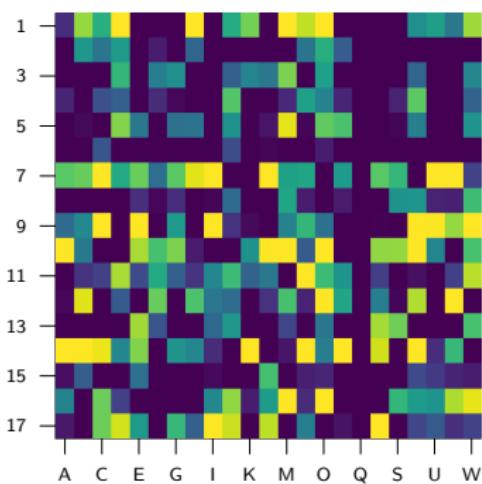
(c) MSLL

Number of Independent GPs ( $Q$ )

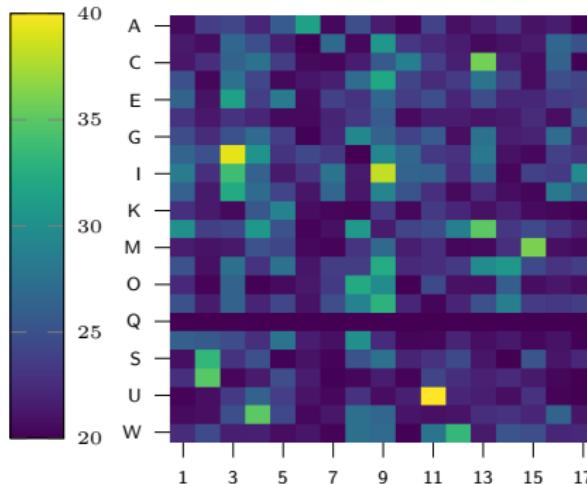
(d) NLPD

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

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



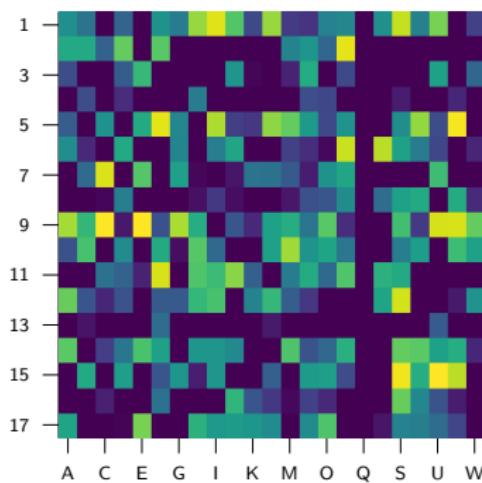
(a)



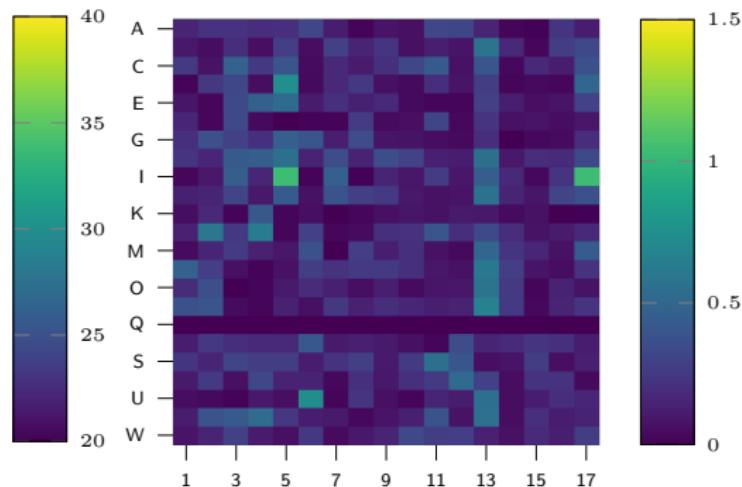
(b)

Lengthscale values (left) and coefficients  $a_{d,q}$  (right) for horizon  $H = 1$  reveal two feature usage patterns: focused extraction from few features and broad dynamics capture from many, with smaller individual contributions.

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



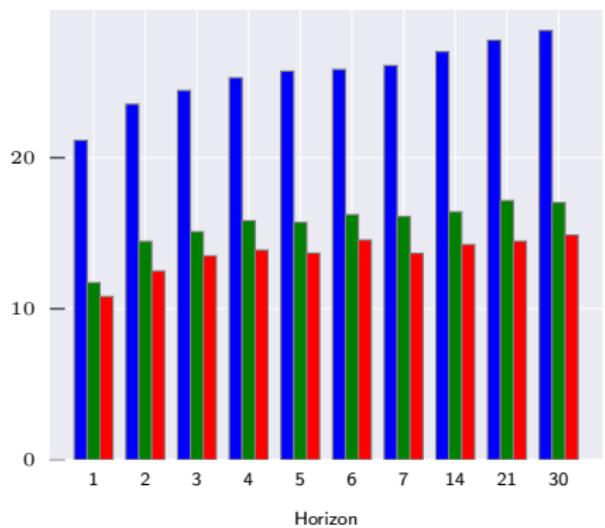
(a)



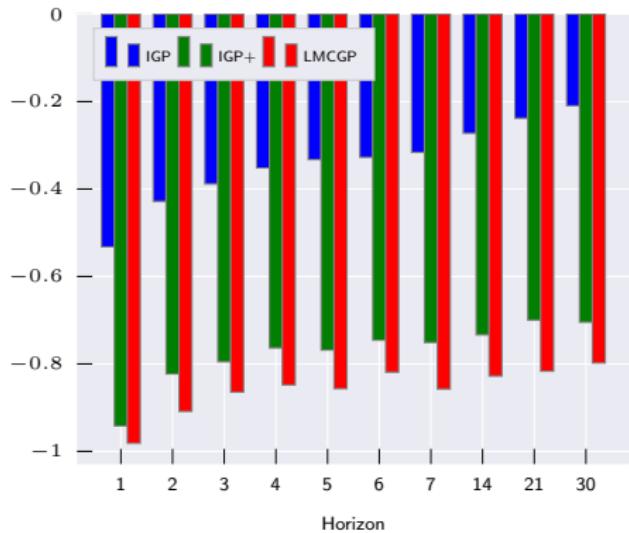
(b)

For horizon  $H = 30$ , lengthscale values (left) and coefficients  $a_{d,q}$  (right) show less selective input feature usage. All independent GPs incorporate more features due to the extended time gap. The  $a_{d,q}$  coefficients are smaller, indicating weaker individual feature contributions to each output.

# LMCGP vs IGP+ vs IGP



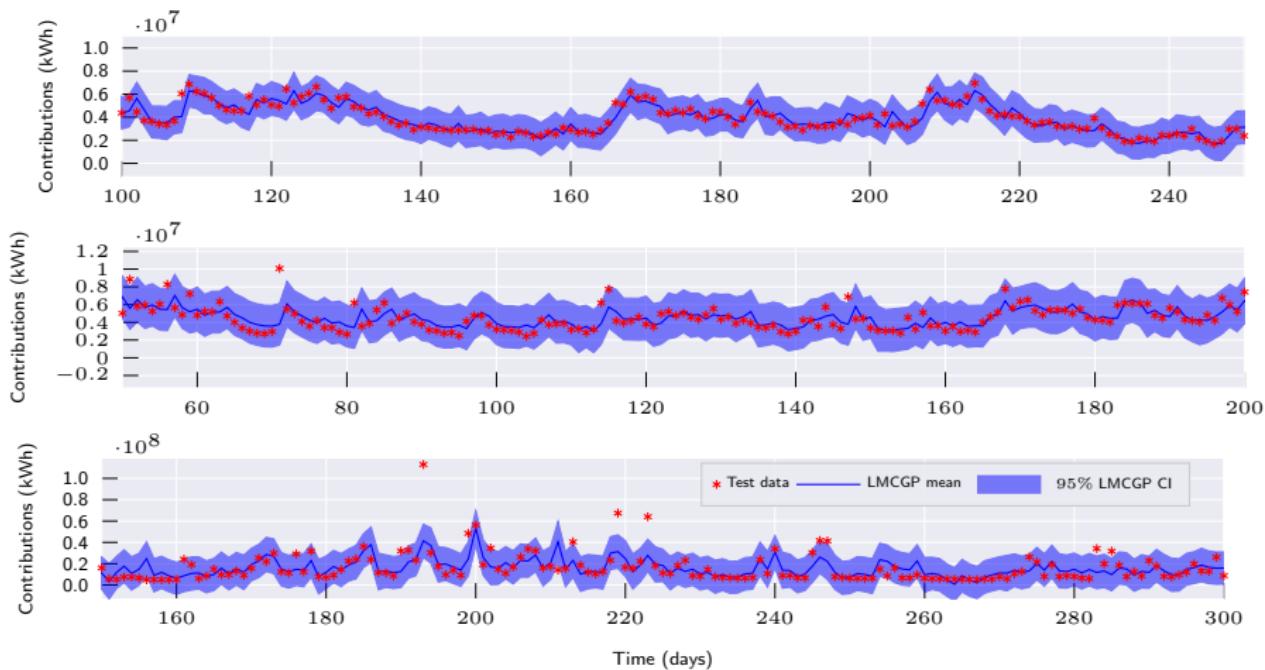
(a) NLPD



(b) MSLL

Bar plots comparing LMCGP, IGP+, and IGP model performance across different horizons  $H$ . The Adam+NG optimizer significantly boosts performance, and with LMCGP showing the most improvement, especially for larger horizons.

# Model Forecasting



Test data for reservoirs T, A, and I (top to bottom) in one day ahead LMCGP model prediction. The model more accurately follows the peaks due its complex behavior.

## 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

# Chained Correlated GP (ChdGP)

## Independent Process

$$u_q(\mathbf{x}) \sim \mathcal{GP}(0, k_q(\mathbf{x}, \mathbf{x}'))$$

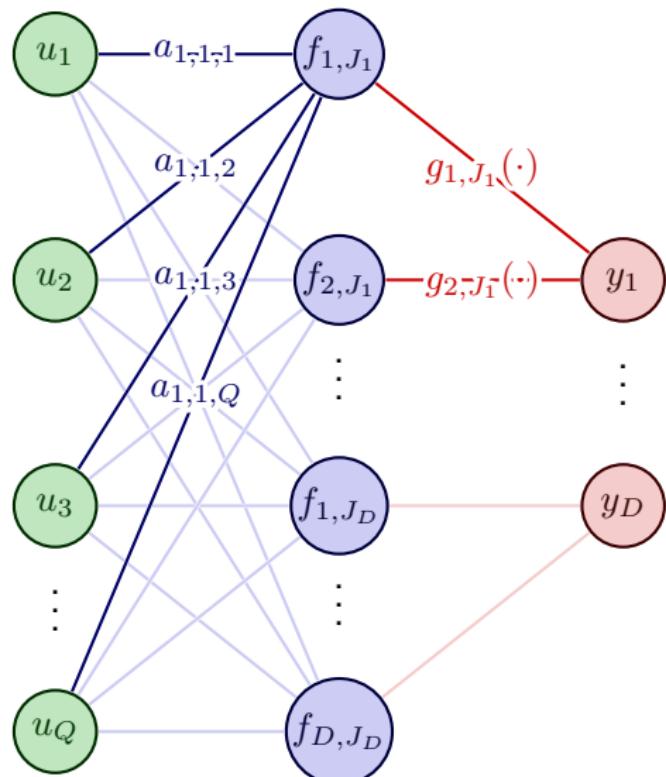
## Latent Process

$$f_{d,j}(\mathbf{x}) = \sum_{q=1}^Q a_{d,j,q} u_q(\mathbf{x})$$

## Likelihood

$$p(\mathbf{y} \mid \mathbf{f}) = \prod_{d=1}^D p(y_d \mid \theta_{d,1}, \dots, \theta_{d,J_d})$$

- $\theta_{d,j} = g_{d,j}(f_{d,j})$  is a likelihood parameter.
- $g_{d,j}(\cdot)$  is a deterministic function.



# Variational Inference, ELBO and Predictive Distribution

We can extend our variational inference, providing the following ELBO:

$$\begin{aligned}\mathcal{L} = & \sum_{d=1}^D \sum_{n=1}^N \mathbb{E}_{q(f_{d,1,n}), \dots, q(f_{d,J_d,n})} \{ \log p(y_{d,n} | f_{d,1,n}, \dots, f_{d,J_d,n}) \} \\ & - \sum_{q=1}^Q \text{KL} \{ q(\mathbf{u}_q) \| p(\mathbf{u}_q) \}\end{aligned}$$

The approximated posterior over test points is given by:

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

And the predictive distribution for a new output  $\mathbf{y}_*$ :

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

The expectation values can be approximated via Monte Carlo methods.



# 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

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

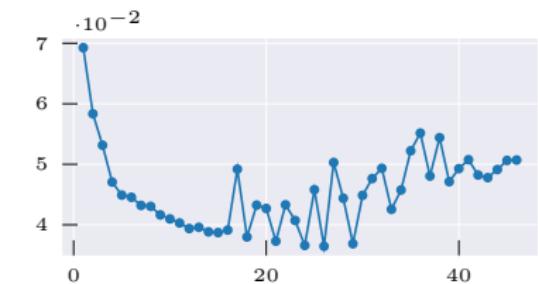
In this formulation,  $g_{d,1}(\cdot) = \cdot$ , while  $g_{d,2}(\cdot) = \ln(\exp(\cdot) + 1)$ . We call this model Chd Normal.

## Gamma Likelihood

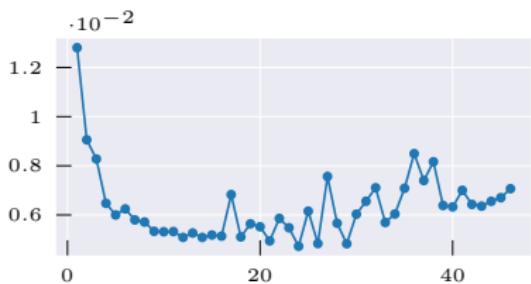
$$p(\mathbf{y} \mid \mathbf{f}) = \prod_{d=1}^D \text{Gamma}(y_d \mid g_{d,1}(f_{d,1}), g_{d,2}(f_{d,2}))$$

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

Number of Independent GPs ( $Q$ )

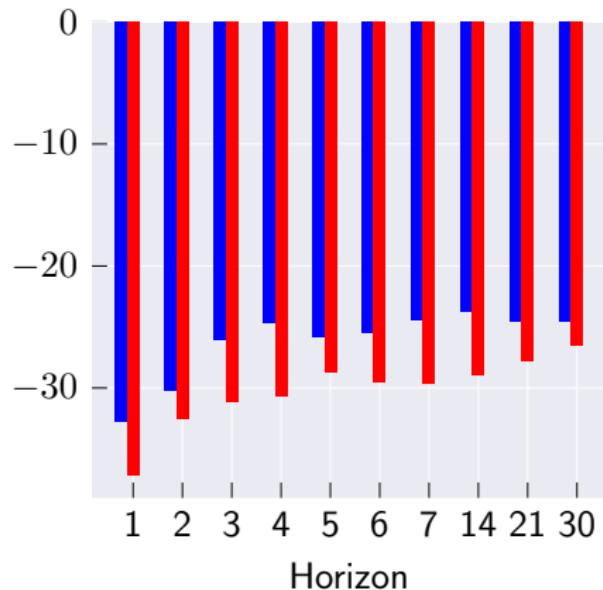
(c) MSLL

Number of Independent GPs ( $Q$ )

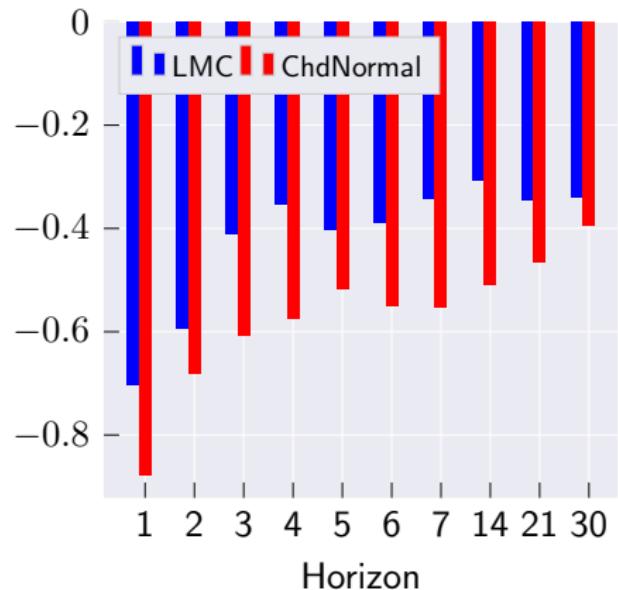
(d) NLPD

**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



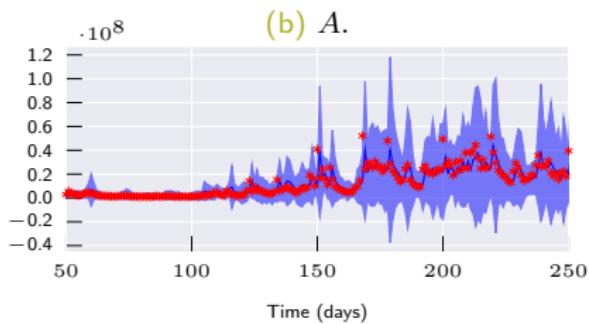
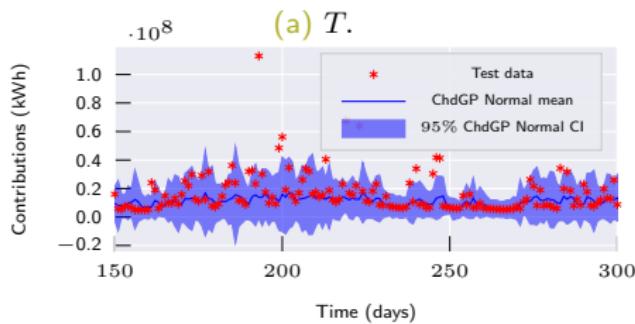
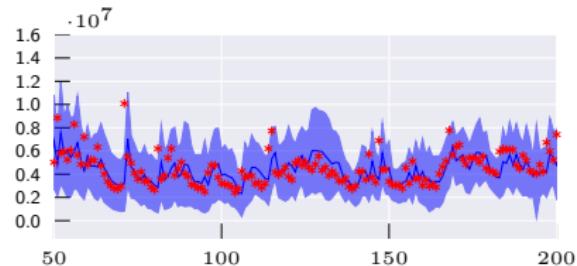
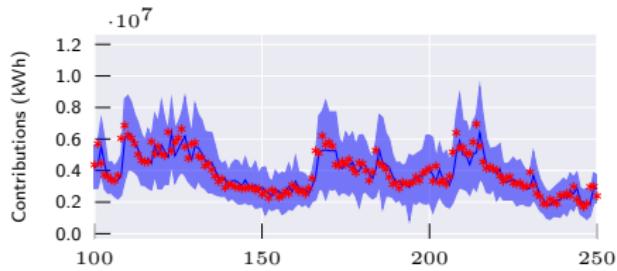
(a) NLPD



(b) MSLL

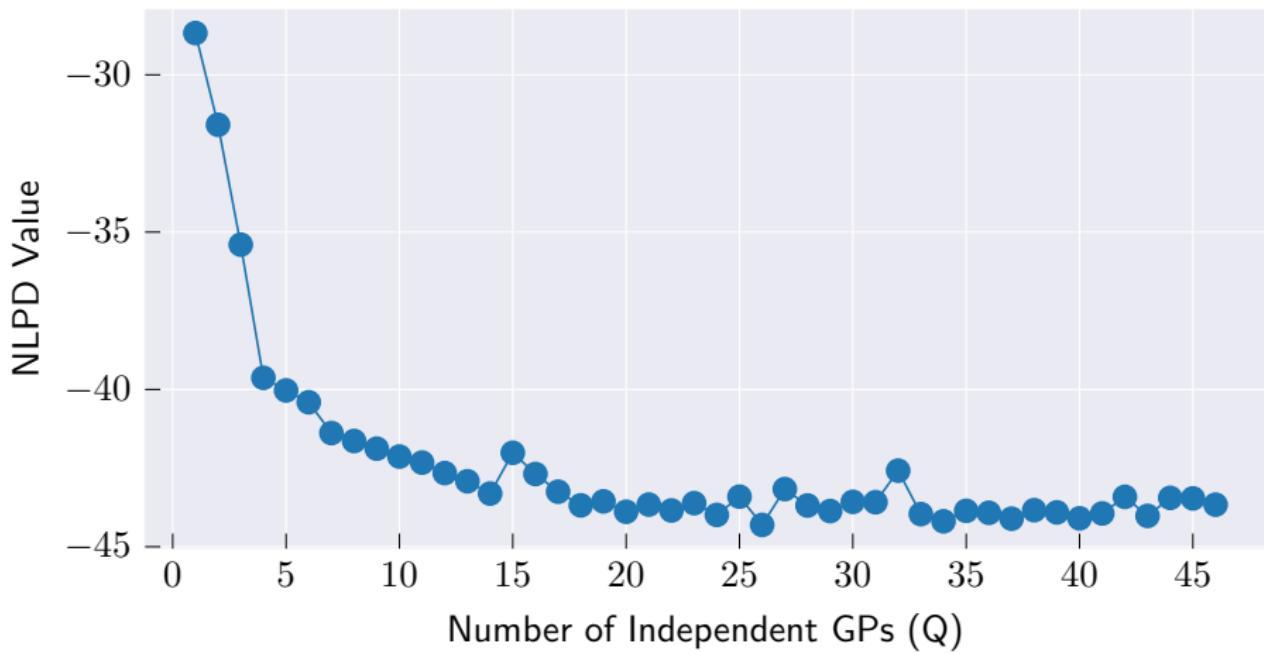
**Figure:** Bar plots comparing the performance of LMCGP, and ChdGP Normal models for different  $H$  values.

# ChdGP Normal Forecasting



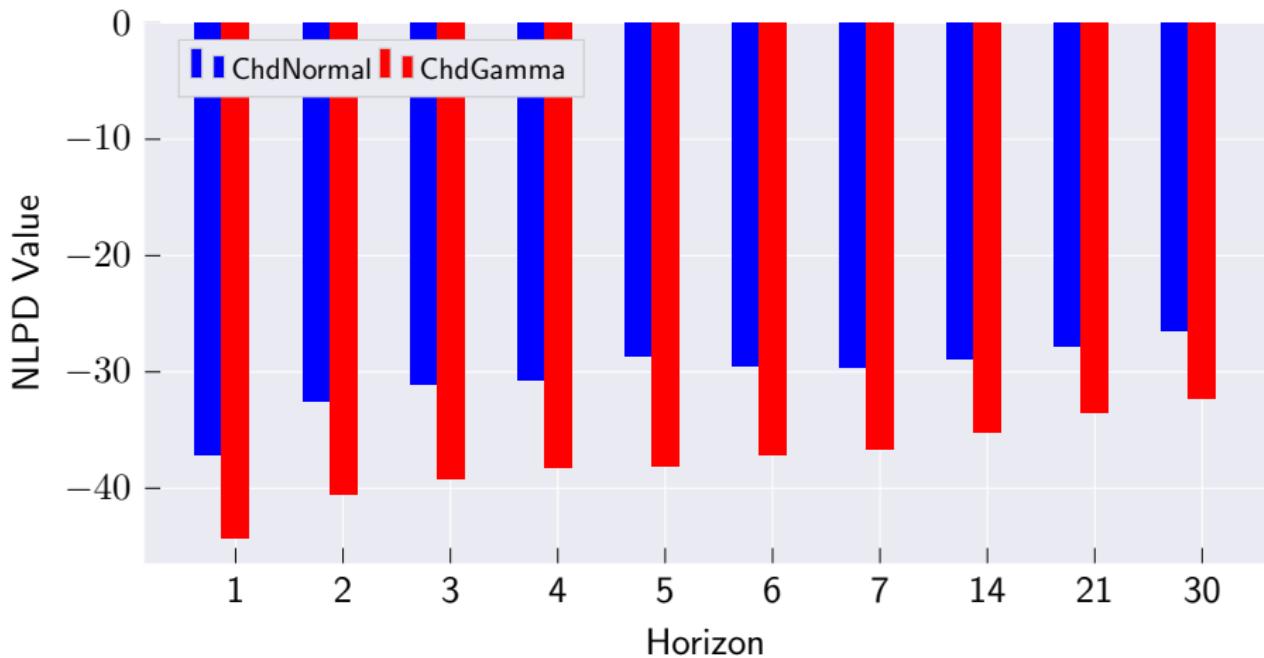
**Figure:** Test data for four reservoirs in one day ahead ChdGP Normal model prediction ( $H = 1$ ). Blue shaded areas represent the 95% centered confidence interval for the model's prediction.

# Tuning Q for Chd Gamma



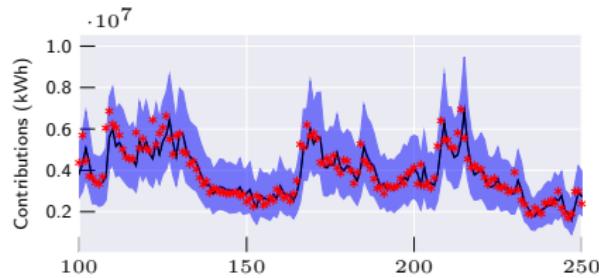
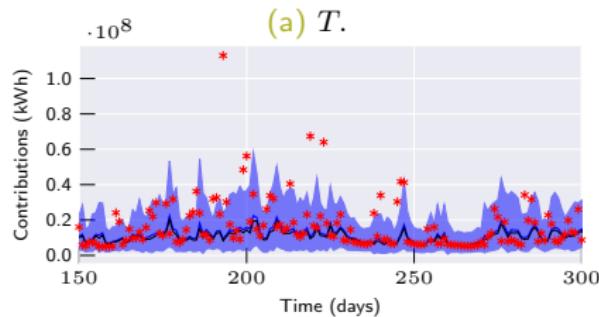
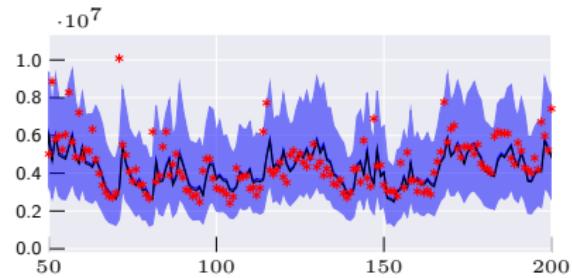
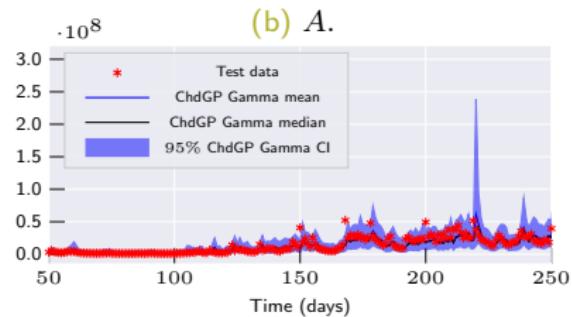
**Figure:** NLPD metric for ChdGP Gamma models as a function of the number of independent GPs. We select  $Q = 26$  as the optimal value.

# 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.*(c) *I.*(b) *A.*(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.

# References

- [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 Dinپashoh, 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] Juan-José Giraldo and Mauricio A Alvarez. A fully natural gradient scheme for improving inference of the heterogeneous multioutput gaussian process model. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11):6429–6442, 2021.
- [13] Hugh Salimbeni, Stefanos Eleftheriadis, and James Hensman. Natural gradients in practice: Non-conjugate variational inference in gaussian process models. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 689–697. PMLR, 09–11 Apr 2018.