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 11, 2024

Universidad Tecnológica de Pereira

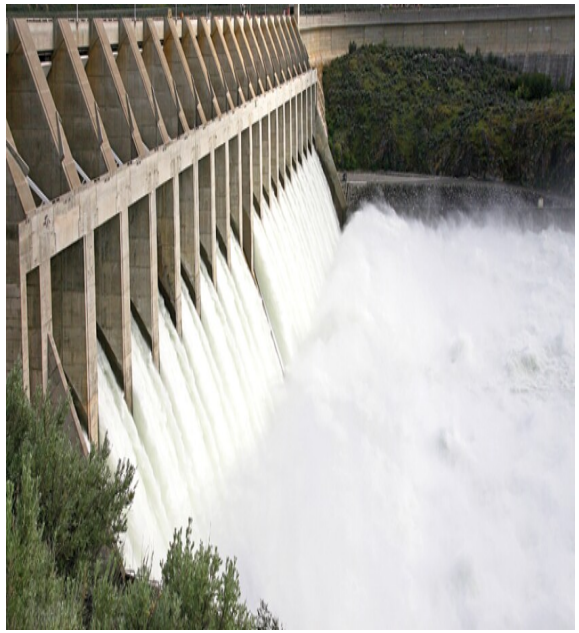
Introduction

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

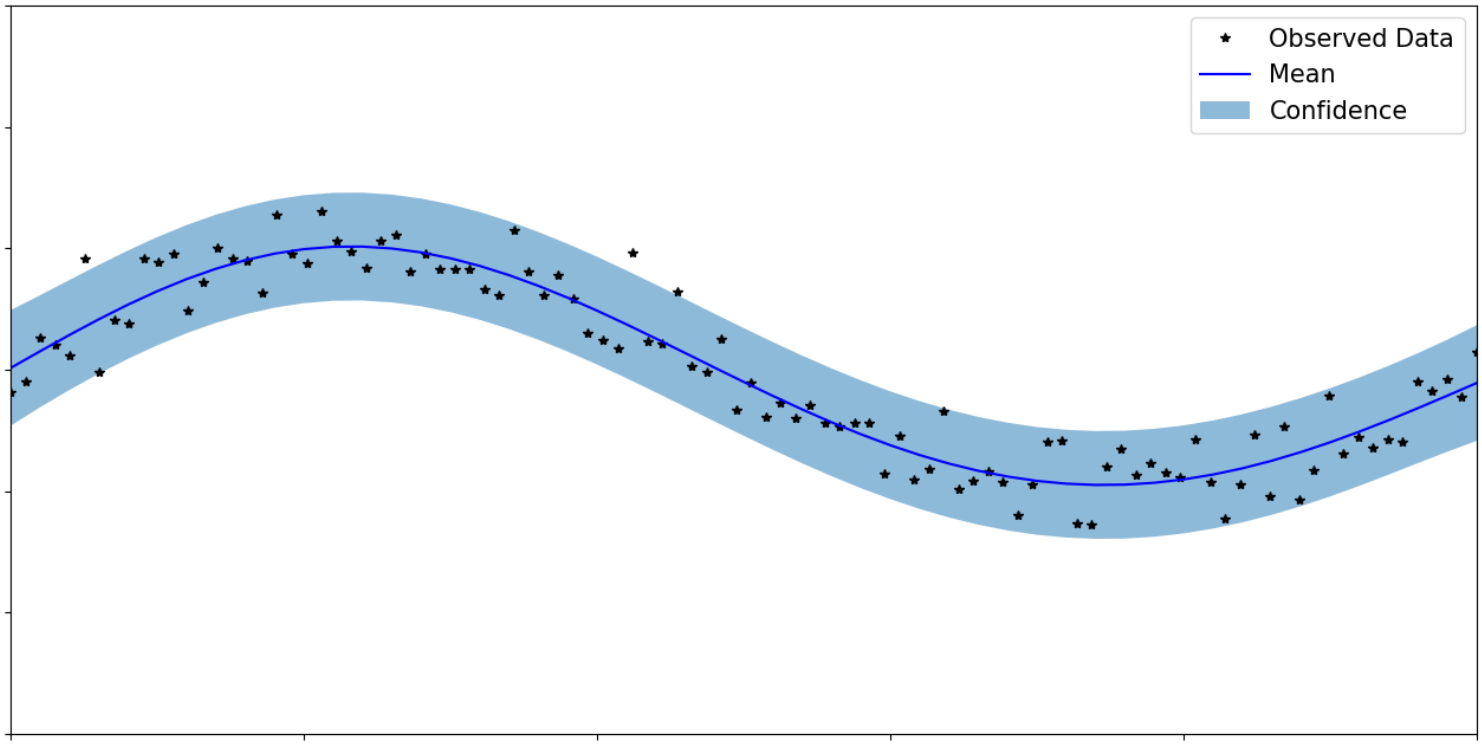
Understanding hydrological processes has become increasingly critical in the field of natural resource management, anticipation capacity of extreme hydrological events such as droughts and heavy rainfall.

(a) Drought Condition

(b) Full Dam

The Proposed Model

A Gaussian Process (GP) is a Bayesian non-parametric model that provides not just point predictions but a full probability distribution, capturing uncertainty and enabling confidence intervals in time series forecasting.



Objectives

General Objective

Develop a stochastic forecasting model for making multiple simultaneous predictions of hydrological time series, taking advantage of cross-correlations among the tasks to improve the performance keeping the scalability in its implementation for the short-term horizon.

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 - Part 1

Consider a time-series vector of hydrological resources observed across all D outputs at the n-th time instant, denoted as $\mathbf{v}_n \in \mathbb{R}^D$. Our model employs the entire sequence of resource vectors from time n back to $n-T+1$ as input to predict the resource vector at the future time step $n+H$. Here, T represents the model order, and H denotes the prediction horizon.
Consequently, we define the input vector $\mathbf{x}_n$ as follows:

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

Problem Setting - Part 2

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

$$\mathbf{y}_n = \mathbf{v}_{n+H}.$$

This formulation enables the model to leverage historical hydrological data for accurate future predictions. Accordingly, we build a dataset $\mathcal{D} = \{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^{N} = \{\mathbf{X}, \mathbf{y}\}$ comprising N input–output pairs, where $\boldsymbol{x}_n \in \mathcal{X} \subset \mathbb{R}^{DT}$ represents the dimensional input space. The target function is a vector-valued function where $\boldsymbol{y}_n \in \mathbb{R}^{D}$ comprises the observations of all outputs (tasks) at the same input $\boldsymbol{x}_n$.

The hydrological forecasting task for validating the proposed model regressors considers time-series streamflow contributions data from 23 Colombian reservoirs.

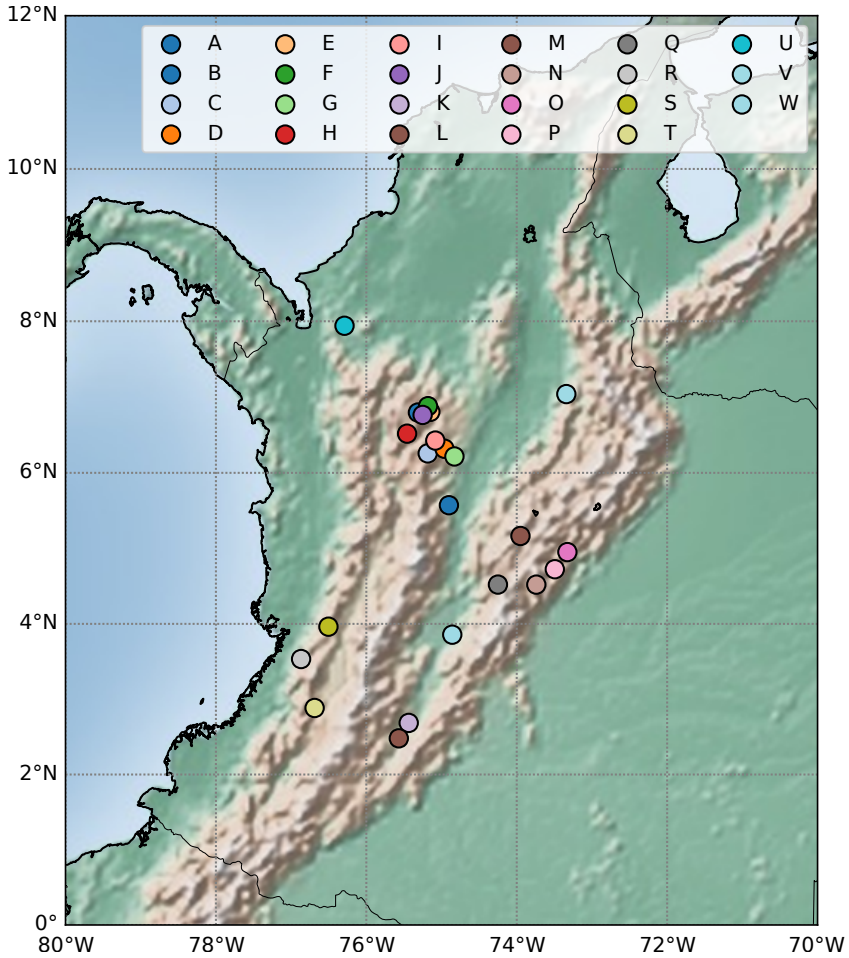


Figure: Reservoir locations in Colombia.

The dataset was selected due to the strong dependence of time series on weather patterns, which are closely tied to hydropower dispatch operations.

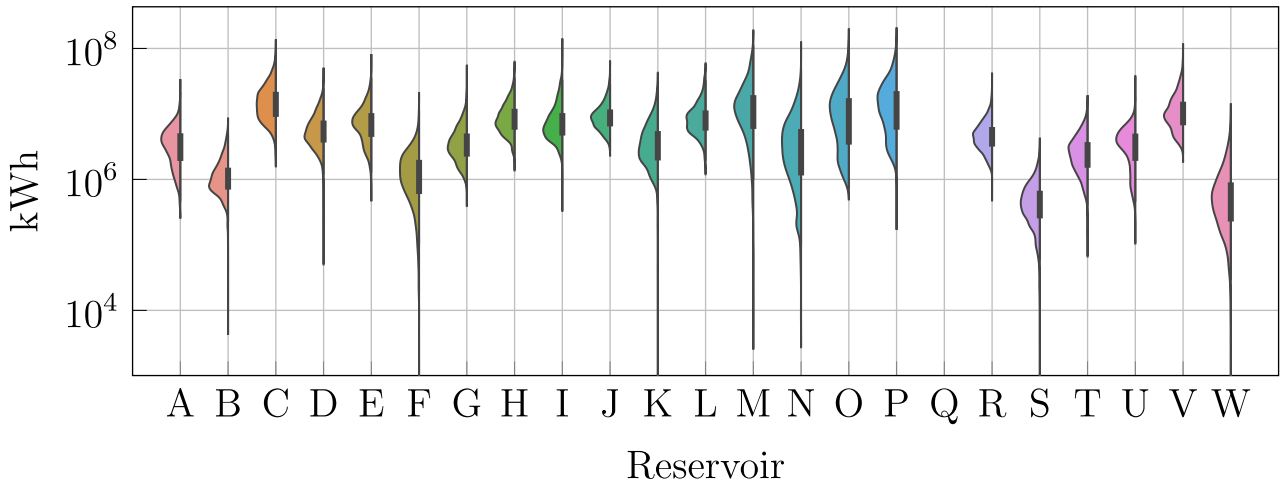


Figure: Time-series Violin plot depicting the streamflow contribution for each reservoir within the dataset.

These streamflow contributions were recorded daily from 1 January 2010 to 28 February 2022. While these contributions represent volumetric values, they are reported in kilowatt-hours (kWh) by the hydroelectric power plants.

Gaussian Process Regression: Bayesian Non-Parametric Model

Gaussian Process (GP) Framework

In the GP framework, the dataset $\mathcal{D}$ is used to learn a random mapping function $\boldsymbol{f}(\cdot)$, which captures the relationship between the input $\boldsymbol{x}_n$ and the output $\boldsymbol{y}_n$. We define the GP distribution over the function $\boldsymbol{f}(\cdot)$ as follows:

$$\boldsymbol{f}(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{0}, \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x}' \mid \boldsymbol{\theta})\right)$$

Where:

- $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^D$ maps the input space $\boldsymbol{x}$ to the output space.
- $\boldsymbol{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{D \times D}$ is the cross-covariance matrix function, parameterized by $\boldsymbol{\theta}$.

By adding i.i.d Gaussian noise $\boldsymbol{\epsilon}$ with diagonal covariance matrix $\Sigma_\epsilon = \text{diag}\{\sigma_{Nd}^2\}_{d=1}^D$, the model prediction becomes:

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

Joint Distribution

Let $\mathbf{X}_* = \{\boldsymbol{x}_{n*}\}_{n=1}^{N_*}$ represent a set of test points $\boldsymbol{x}_{n*} \in \mathcal{X}$, with the corresponding test output vector $\mathbf{f}_*$ defined as:

$$\mathbf{f}_* = [\boldsymbol{f}(\boldsymbol{x}_1)^\top, \boldsymbol{f}(\boldsymbol{x}_2)^\top, \cdots, \boldsymbol{f}(\boldsymbol{x}_{N_*})^\top]^\top \in \mathbb{R}^{N_* D}.$$

The joint Gaussian distribution over the training outputs $\mathbf{y}$ and the test outputs $\mathbf{f}_*$ can be expressed as:

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

- $\mathbf{K}_y = \mathbf{K} + \Sigma_\epsilon$ is the covariance matrix for the training set, where $\mathbf{K} \in \mathbb{R}^{ND \times ND}$ is the covariance matrix formed by evaluating the covariance function on all pairs of input vectors in $\mathbf{X}$.
- $\mathbf{K}_{**} \in \mathbb{R}^{N_* D \times N_* D}$ 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.

Posterior Distribution

The joint distribution of training and test points enables us to derive the conditional distribution, known as the posterior, as follows:

$$\mathbf{f}_*|\mathbf{X}_*, \mathcal{D} \sim \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*))$$

where the mean and covariance of the posterior distribution are given by:

$$\bar{\mathbf{f}}_* = \mathbf{K}_*^\top \mathbf{K}_y^{-1} \mathbf{y}$$

$$\text{cov}(\mathbf{f}_*) = \mathbf{K}_{**} - \mathbf{K}_*^\top \mathbf{K}_y^{-1} \mathbf{K}_*$$

Here, $\bar{\mathbf{f}}_*$ represents the predicted mean of the test points, and $\text{cov}(\mathbf{f}_*)$ is the predicted covariance of the 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

A common method to reduce the computational complexity of GPs is the introduction of a reduced set of $M \ll N$ inducing point locations, denoted as Z. These inducing points contain vectors $\mathbf{z} \in \mathcal{X}$, and an inducing variable vector $\mathbf{u} = [\boldsymbol{f}(\boldsymbol{z}_1)^\top, \boldsymbol{f}(\boldsymbol{z}_2)^\top, \cdots, \boldsymbol{f}(\boldsymbol{z}_M)^\top]^\top \in \mathbb{R}^{MD}$. This approach leads to the following extended joint distribution:

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

In this formulation:

- $\mathbf{K}_{uu} \in \mathbb{R}^{MD \times MD}$ represents the covariance matrix of the inducing points.
- $\mathbf{K}_{uf} \in \mathbb{R}^{MD \times ND}$ represents the covariance matrix between the inducing points and the training inputs.

This model is referred to as the Sparse Variational Gaussian Process (SVGP).

Variational Distribution

The conditional distribution given inducing variables is:

$$p(\mathbf{f} \mid \mathbf{u}) = \mathcal{N}\left(\mathbf{f} \mid \mathbf{K}_{uf}^{\top}\mathbf{K}_{uu}^{-1}\mathbf{u}, \mathbf{K} - \mathbf{K}_{uf}^{\top}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right),$$

with prior:

$$p(\mathbf{u}) = \mathcal{N}(\mathbf{u} \mid \mathbf{0}, \mathbf{K}_{uu}).$$

The joint posterior distribution is:

$$p(\mathbf{f}, \mathbf{u} \mid \mathbf{y}) = p(\mathbf{f} \mid \mathbf{u})p(\mathbf{u} \mid \mathbf{y}),$$

which is approximated using variational inference:

$$q(\mathbf{u}) = \mathcal{N}\left(\mathbf{u} \mid \boldsymbol{\mu}, \boldsymbol{S}\right),$$

resulting in:

$$p(\mathbf{f}, \mathbf{u} \mid \mathbf{y}) \approx q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \mid \mathbf{u})q(\mathbf{u}).$$

ELBO - Evidence Lower Bound

The variational parameters $\boldsymbol{\mu}$ and $\boldsymbol{S}$ are estimated by maximizing the Evidence Lower Bound (ELBO) on the log marginal likelihood:

$$\ln p(\mathbf{y}) \geq \int \int q(\mathbf{f}, \mathbf{u}) \ln \frac{p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathbf{u}) p(\mathbf{u})}{q(\mathbf{f}, \mathbf{u})} d\mathbf{f} d\mathbf{u} = \mathcal{L}$$

This inequality results from Jensen's inequality. The ELBO can be rewritten as:

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{f})}\{\ln p(\mathbf{y} \mid \mathbf{f})\} - \mathsf{KL}\{q(\mathbf{u}) \parallel p(\mathbf{u})\}$$

where the Kullback–Leibler (KL) divergence measures the difference between the variational distribution $q(\mathbf{u})$ and the prior $p(\mathbf{u})$.

ELBO - Evidence Lower Bound

Under the i.i.d. assumption, the likelihood function $p(\mathbf{y} \mid \mathbf{f})$ can be factorized across observations and outputs:

$$p(\mathbf{y} \mid \mathbf{f}) = \prod_{d=1}^{D} \prod_{n=1}^{N} p(y_{dn} \mid f_d(\mathbf{x}_n)),$$

where $f_d(\mathbf{x}_n)$ represents the latent function value for the d-th output at input $\mathbf{x}_n$, and y_{dn} is the corresponding observed value. The ELBO can then be written as:

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

This formulation enables efficient training through mini-batch updates, making it scalable for large datasets.

Posterior and Predictive Distribution

The variational distribution for $\mathbf{f}$ is defined as:

$$q(\mathbf{f}) = \int p(\mathbf{f} \mid \mathbf{u}) q(\mathbf{u}) d\mathbf{u}$$
$$= \mathcal{N}\left(\mathbf{f} \mid \mathbf{K}_{uf}^{\top}\mathbf{K}_{uu}^{-1}\boldsymbol{\mu}, \mathbf{K} - \mathbf{K}_{uf}^{\top}\mathbf{K}_{uu}^{-1}(\mathbf{K}_{uu} - \mathbf{S})\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right).$$

This reduces computational complexity to $\mathcal{O}(NM^2D^3)$, as $\mathbf{K}_{uu}$ is smaller than $\mathbf{K}_y$. For predictions at new points $\mathbf{x}_*$, the posterior distribution is approximated by:

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

with Gaussian noise Σ_ϵ added to the latent posterior $\mathbf{f}_* \mid \mathbf{y}$ for predictions.

Model Setup

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

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

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

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

where $\delta_{d,d'}$ is the Kronecker delta, σ_d^2 is the output scale, and $\boldsymbol{\Theta}_d$ is the task-specific lengthscale matrix. The input kernel $k_{\mathcal{X}}$ uses a squared-exponential function for smooth mapping, while k_D reduces complexity to $\mathcal{O}(NM^2D)$ by avoiding explicit task correlations. Task dependencies are still captured through autoregressive inputs and shared inducing points.

Performance Metrics

We use three metrics to evaluate model performance:

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

Mean Squared Error (MSE)

MSE measures the average squared difference between the observed outcome y_{dn} and the expected value of the predicted outcome $\mathbb{E}\{y_{dn*}\}$ for N_* test observations:

$$\text{MSE} = \frac{1}{DN_*} \sum_{d=1}^{D} \sum_{n=1}^{N_*} (y_{dn} - \mathbb{E}\{y_{dn*}\})^2$$

MSE does not account for prediction uncertainty.

Mean Standardized Log Loss (MSLL)

MSLL evaluates probabilistic prediction quality by computing the log probability of observed values under the predicted distribution:

$$\text{MSLL} = \frac{1}{2DN_*} \sum_{d=1}^{D} \sum_{n=1}^{N_*} \left(\frac{(y_{dn} - \mathbb{E}\{y_{dn*}\})^2}{\text{var}\{y_{dn*}\}} - \frac{(y_{dn} - \mu_d)^2}{\sigma_d^2} + \ln \left(\frac{\text{var}\{y_{dn*}\}}{\sigma_d^2} \right) \right)$$

where $\text{var}\{y_{dn*}\}$, μ_d, and σ_d^2 are the predicted variance, mean, and variance of the training data for the d-th output. Positive MSLL indicates worse performance compared to a Gaussian baseline, while negative values suggest improvement.

Continuous Ranked Probability Score (CRPS)

CRPS measures prediction quality by comparing the predicted CDF $\Phi(y_{dn})$ to the empirical CDF of the observed values. For Gaussian predictive distributions, CRPS is given by:

$$\text{CRPS} = \frac{1}{DN_*} \sum_{d=1}^{D} \sum_{n=1}^{N_*} \sqrt{\text{var}\{y_{dn*}\}} \left(\beta_{dn} \left(2\Phi \left(\beta_{dn} \right) - 1 \right) + 2\phi \left(\beta_{dn} \right) - \frac{1}{\sqrt{\pi}} \right)$$

with $\beta_{dn} = \frac{y_{dn} - \mathbb{E}\{y_{dn*}\}}{\sqrt{\text{var}\{y_{dn*}\}}}$ and the Gaussian distribution $\phi(\beta_{dn}) = \mathcal{N}(\beta_{dn} \mid 0, 1)$.

Hyperparameter Tuning

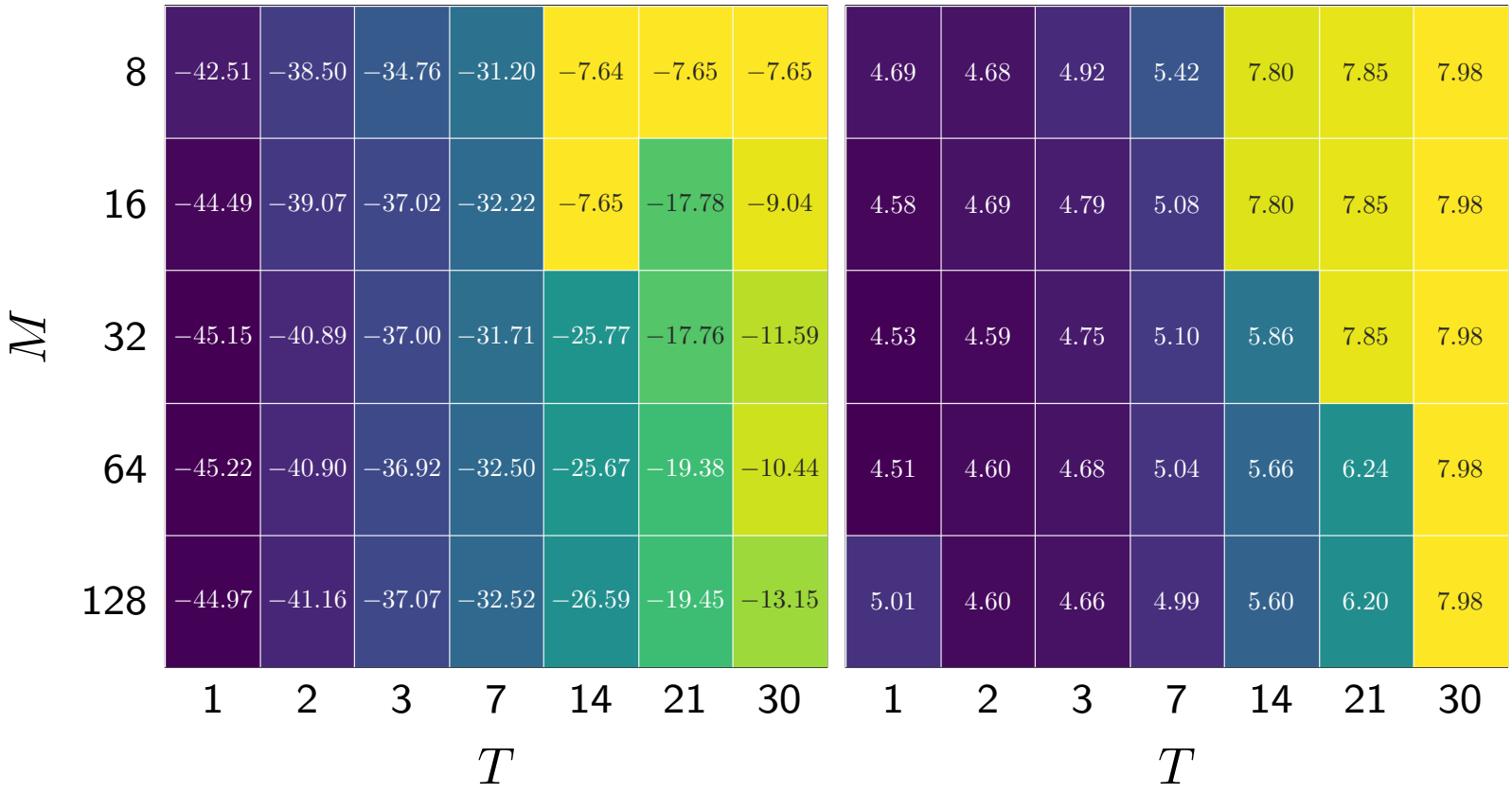
The forecasting task predicts the $D = 23$ reservoir contributions at day $n+H$ using the $L = 23$ contributions at day n. The forecasting horizon H consists of:

- **Short-term predictions:** up to a week ahead, $H \in \{1, 2, 3, 4, 5, 6, 7\}$.
- **Medium-term predictions:** up to one month ahead, $H \in \{9, 14, 21, 30\}$.

The SVGP model requires tuning two important hyperparameters:

- The number of inducing points M
- The model order size T

Tuning M and T

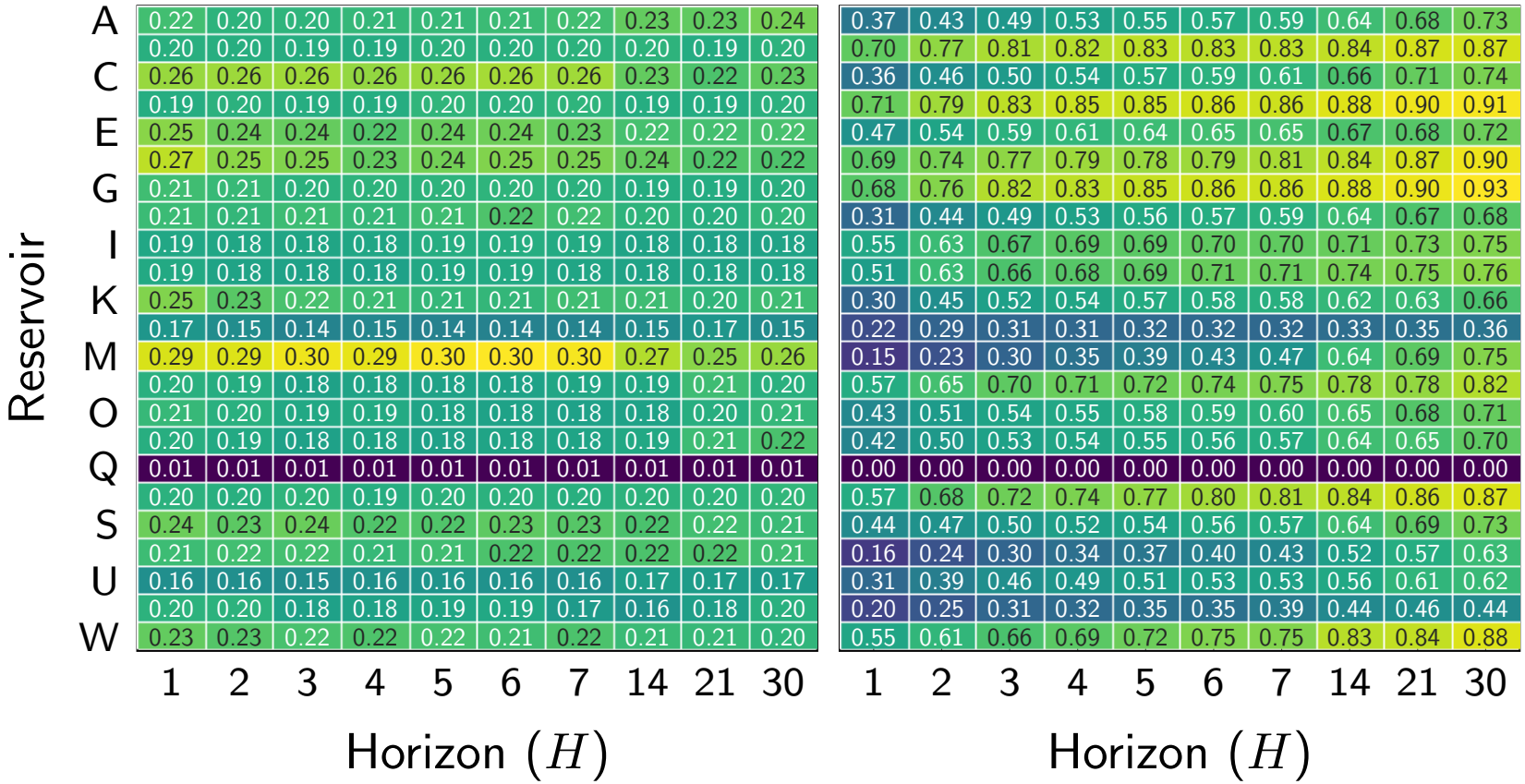


(a) MSLL ($\times 10^{-2}$)

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

Figure: 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

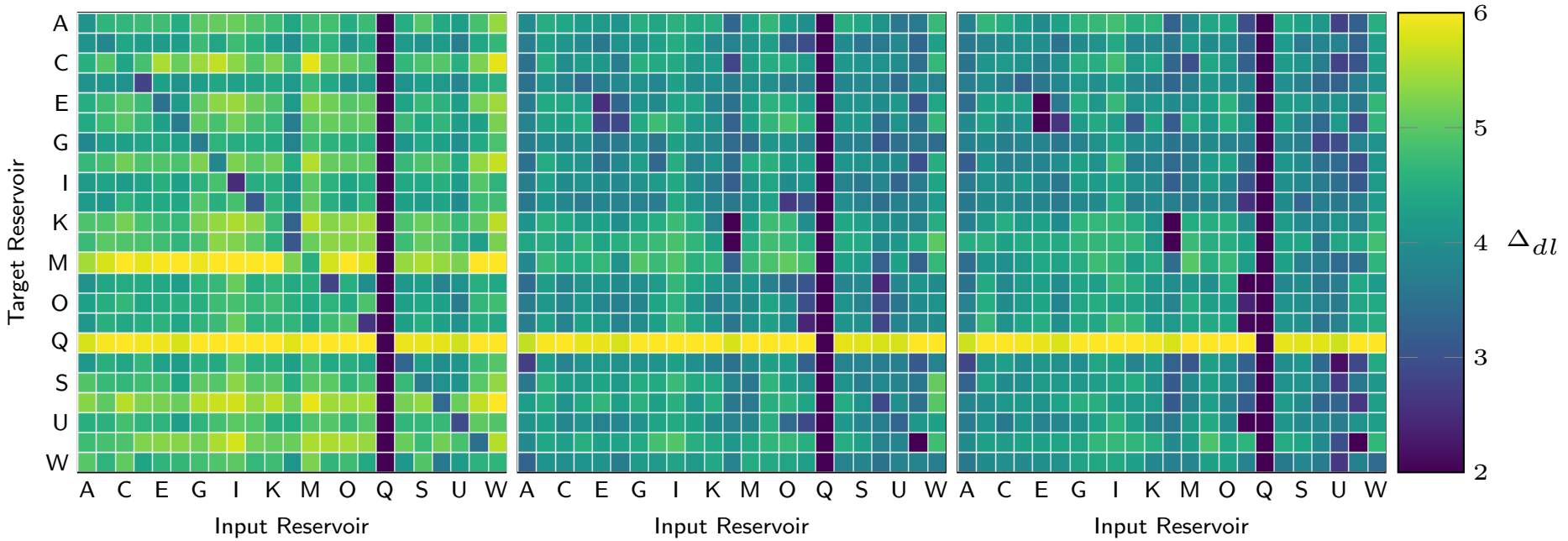


(a) Output scales

(b) Noise variances

Figure: Trained model Output scales σ_d^2 and noise variance Σ_ϵ parameters for each forecasting horizon and input reservoir.

Lengthscale Analysis



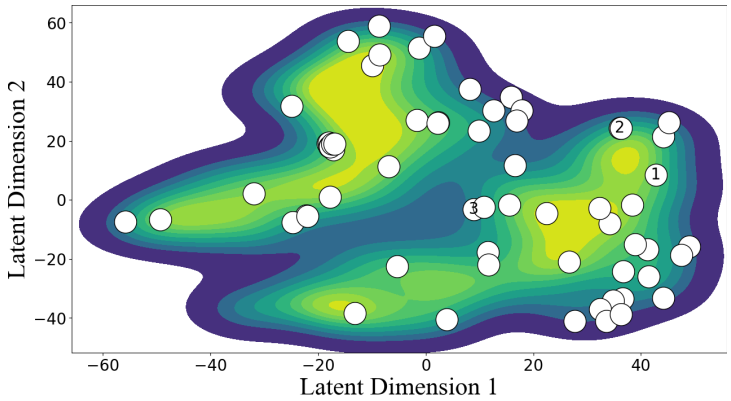
(a) $H = 1$. (b) $H = 14$. (c) $H = 30$.

Figure: Trained lengthscales from input feature (columns) to each output task (rows) for three prediction horizons.

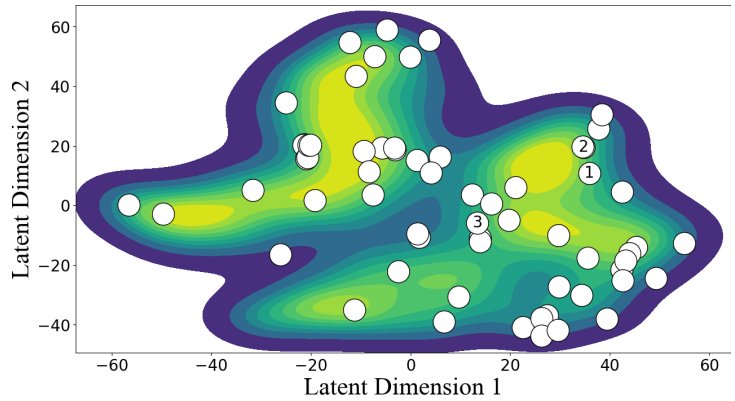
The shortest horizon distributes the smallest lengthscale values over the main diagonal contrary to the high off-diagonal values. This result proves the relevance of short-term memory components within each streamflow.
At a longer horizon of fourteen days, the lengthscales over the main diagonal become higher, losing relevance, and several off-diagonal become lower, gaining relevance, than at a one-day horizon. Such a change implies that a reservoir provides less information to predict itself at longer horizons, leading the model to look for knowledge in other reservoirs.
The longest horizon illustrates even less relevant lengthscales, indicating that the SVGP restrains the number of relevant features. Thus, the lengthscale variations across different prediction horizons reveal the dynamic interplay between reservoirs and highlight the Gaussian Process's ability to select relevant features adaptively.

t-distributed Stochastic Neighbor Embedding (t-SNE)

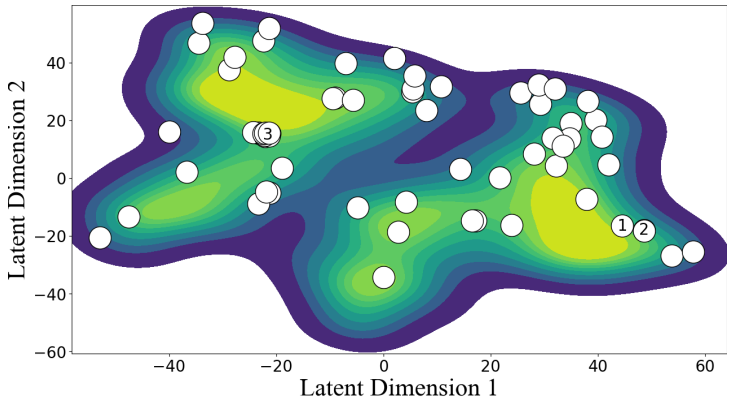
For visual interpretation of SVGP inner workings, the following figure maps in 2D the reservoir data along with the optimized inducing points using the t-distributed Stochastic Neighbor Embedding (t-SNE) technique for each target reservoir. The filled contours outline the density distribution of training data, whereas the color scatter locates the optimized inducing points. We strategically labeled three inducing points for analyzing the SVGP behavior. Since the forecasting tasks share the inducing points, not the kernel parameters, each target reservoir holds a different t-SNE mapping.



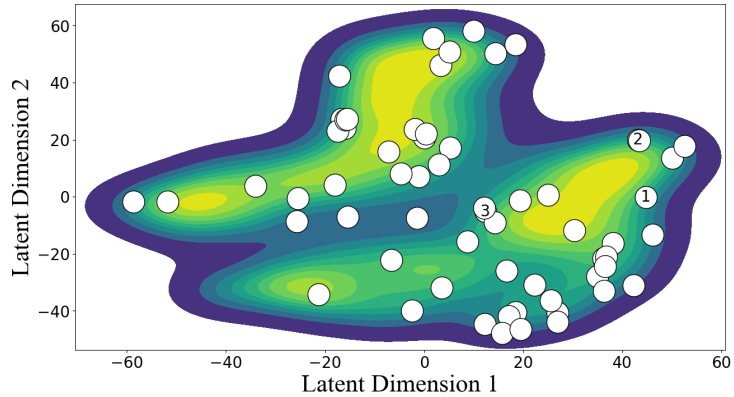
(a) Reservoir E.



(b) Reservoir I.



(c) Reservoir L.



(d) Reservoir U.

Figure: t-SNE-based two-dimensional mapping of the SVGP latent space and inducing points' locations for four target reservoirs. Three inducing points are numbered to analyze the model behavior.

The 2D plots exhibit multi-modal distributions for all tasks, indicating clustered information and varying dynamics within a reservoir. Hence, the SVGP trades off the location of the inducing points between the within-task centroids and the among-task relationships. Therefore, the shared inducing points allow for the capturing of task-wise, group-wise, and global information about the streamflow dynamics.

Model Comparison

For performance analysis, the proposed SVGP-based forecasting is contrasted against the straightforward Linear AutoRegressive model (LAR) and the nonlinear Long-Short-Term Memory network (LSTM).
Particularly for LAR, we trained a single output model for each task. Further, the frequentist perspective treats the model weights as random estimators following a Gaussian distribution. Such a treatment turns the LAR into a probabilistic model, allowing it to generate confidence intervals for its predictions.
In the case of LSTM, we set up a single multi-output model to forecast all reservoirs simultaneously. To tune the number of hidden units of the LSTM, we run a grid search, yielding an optimum of one unit. As a deterministic model, LSTM lacks a predictive distribution and cannot be assessed with the probabilistic performance metrics MSLL and CRPS.

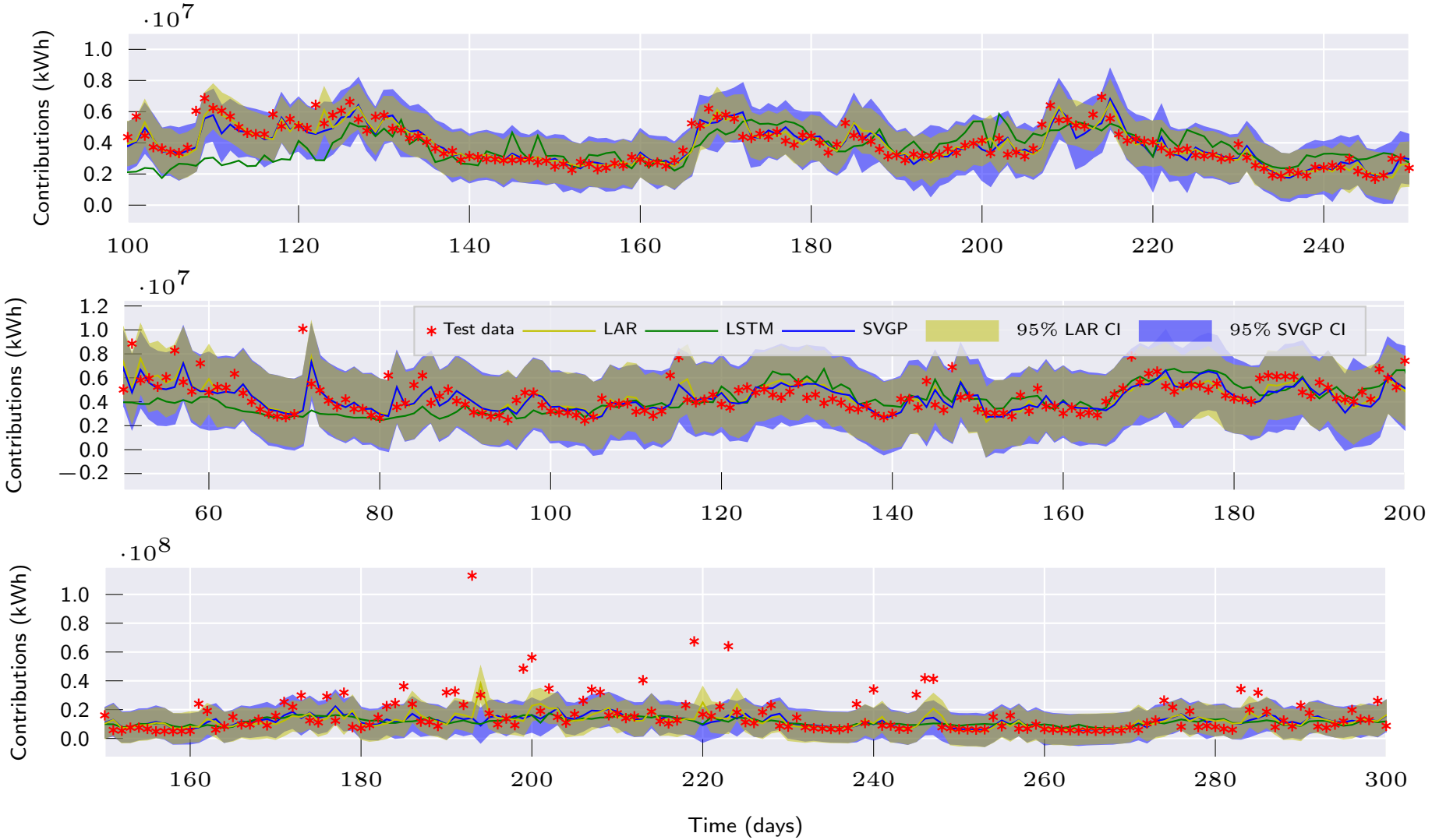
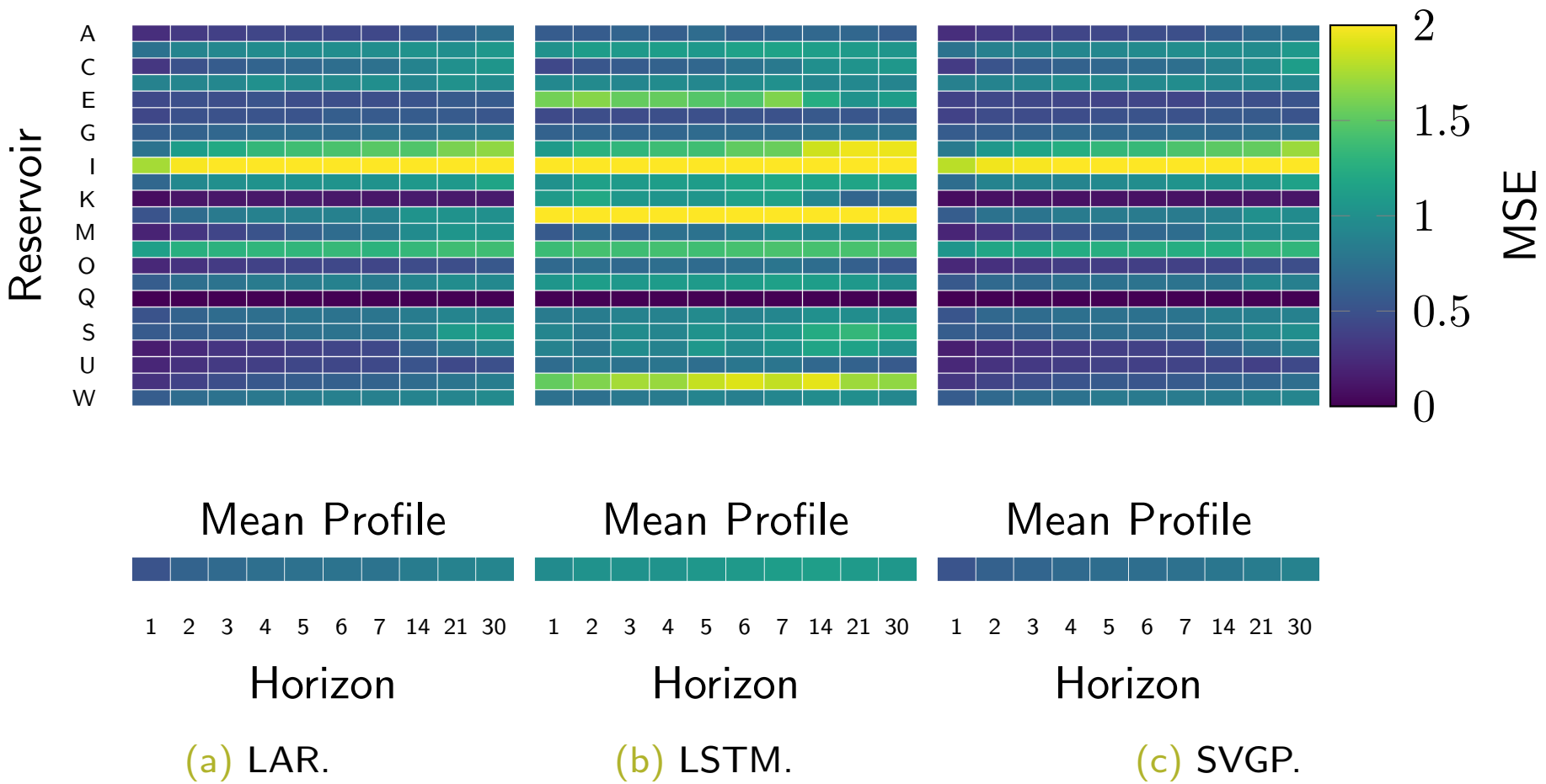


Figure: Test data for reservoirs T (low complexity), A (medium complexity), and I (high complexity), top to bottom in a one-day ahead model's prediction. Yellow and blue shaded areas represent the 95% centered confidence interval for the LAR and SVGP predictions, respectively.

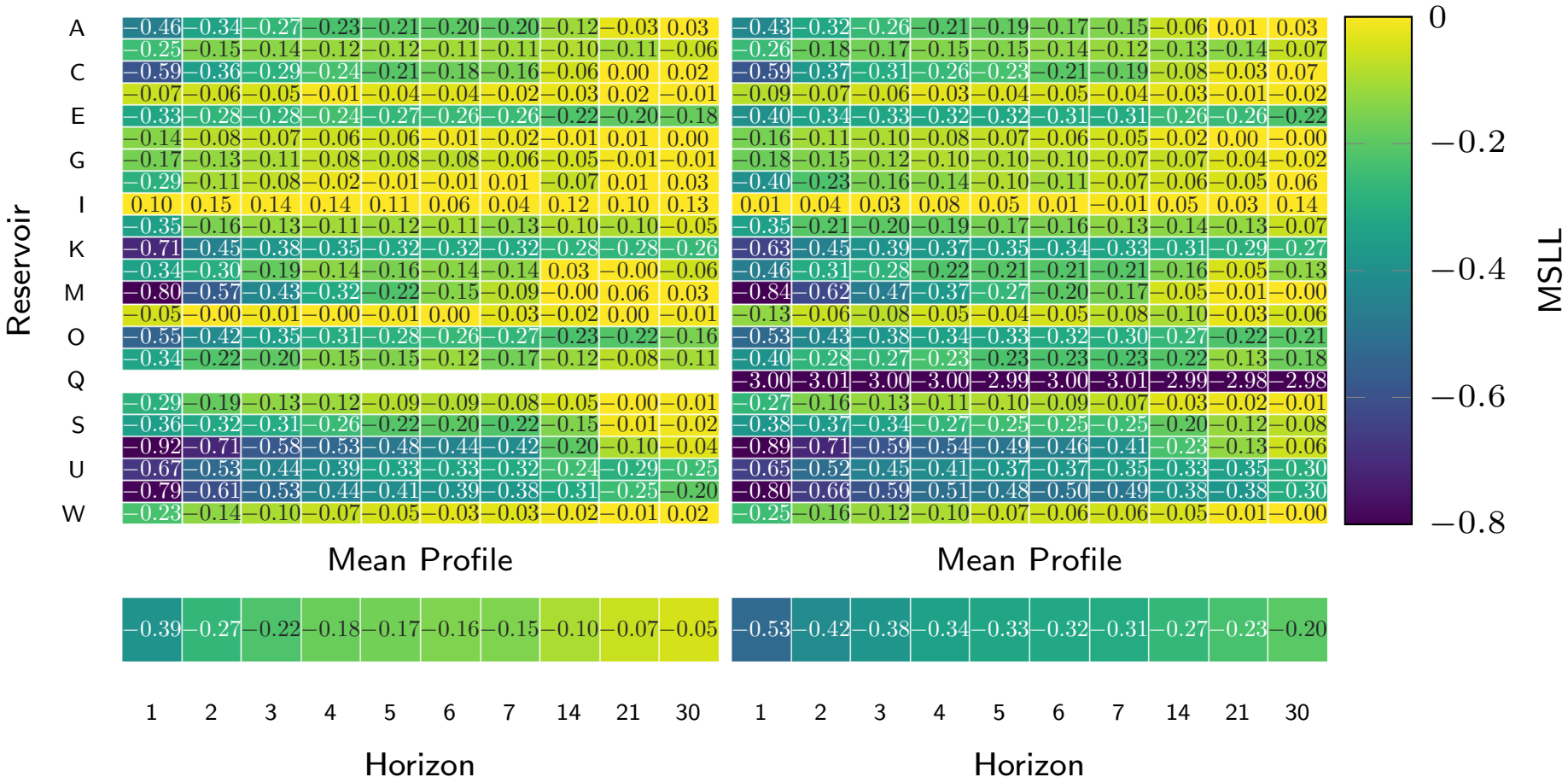
MSE Scores



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

Figure: Achieved MSE for LAR, LSTM, and SVGP forecasting models at each horizon and reservoir.

MSLL Score



(a) LAR. (b) SVGP.

Figure: Achieved MSLL for LAR and SVGP forecasting models at each horizon and reservoir.

Performance t-test

Table: Performance metrics for LAR, LSTM, and SVGP on the considered horizons H. Bold and asterisk denote a p-value $p < 1\%$ in a one-tailed paired t-test for LAR vs. SVGP and LSTM vs. SVGP.

	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 diminishes the computational complexity from cubic to linear regarding the number of samples, becoming more scalable to forecasting tasks on large datasets.
- The optimal number of inducing points performs as an inherent regularization by encoding the most information from the data while avoiding overfitting.
- The t-SNE-based distribution plots reveal that the proposed model strategically places the shared inducing points to capture task-wise, group-specific, and global dynamics, trading off between capturing the reservoir-wise unique characteristics and the between-reservoir dependencies, improving the overall performance in streamflow forecasting.

To Conclude

- The trained lengthscales prove that the GP model successfully adjusts its focus to prediction horizon changes between short-term memory components and long-term relationships. This adaptability makes the Gaussian Process model robust and versatile for multiple-output forecasting tasks with varying time features.
- The performance analysis evidences the advantage of the proposed SVGP model over the baseline LAR and LSTM models for streamflow prediction in three main aspects. Firstly, adaptability to changing dynamics within and between reservoirs. Secondly, the Bayesian scheme returns a posterior distribution and provides informative confidence intervals. Thirdly, the SVGP better copes with the enhanced forecasting complexity for long horizons than baseline approaches, as proved by the slower error growth.

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

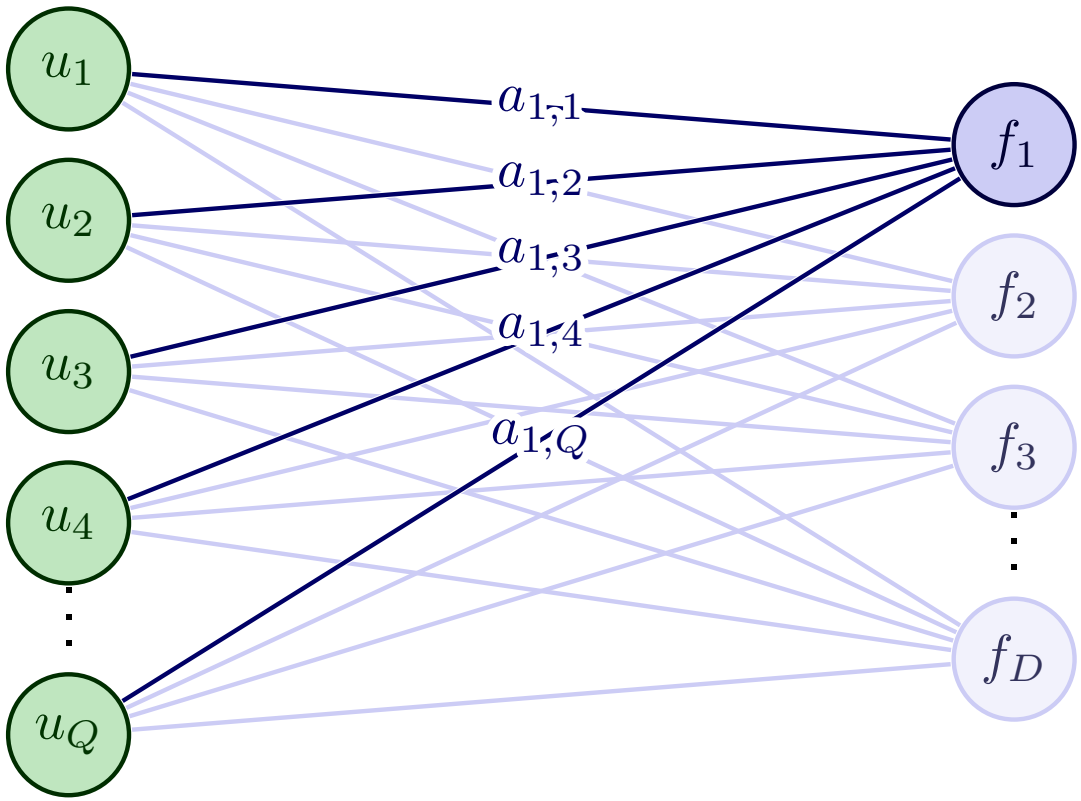


Figure: Graphical representation of the Linear Model of Coregionalization GP

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.

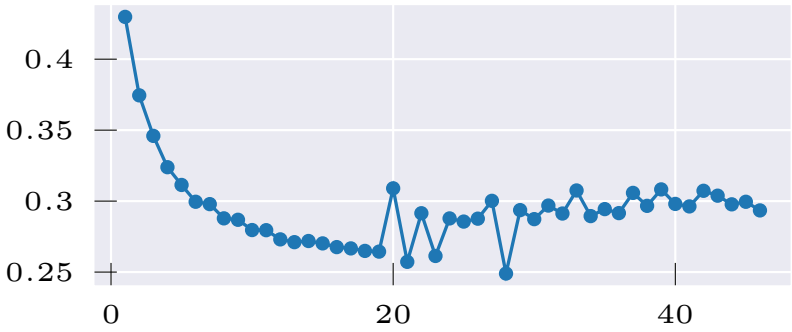
The trainable covariance parameters are $\{\mathbf{B}_q, \Theta_q\}_{q=1}^{Q}$.

Negative Log Predictive Density

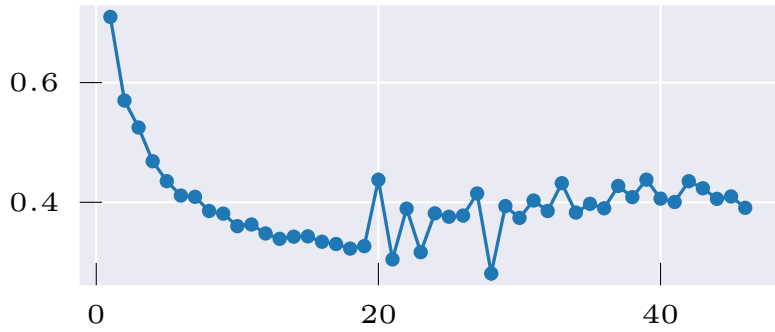
To evaluate the models' performance, we use the implemented metrics MSE, MSLL, and CRPS. Additionally, we propose a new metric called Negative Log Predictive Density (NLPD), which applies to any type of predictive distribution and is defined as:

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

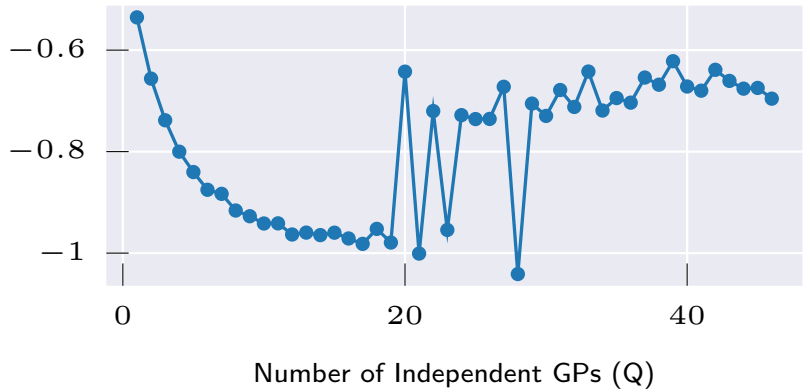
Tuning Q



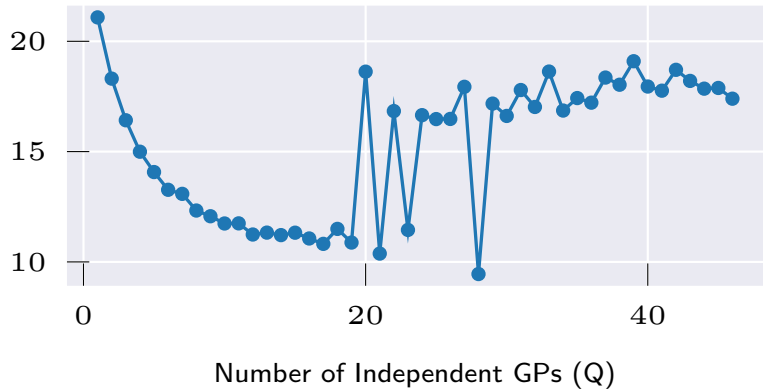
(a) CRPS



(b) MSE



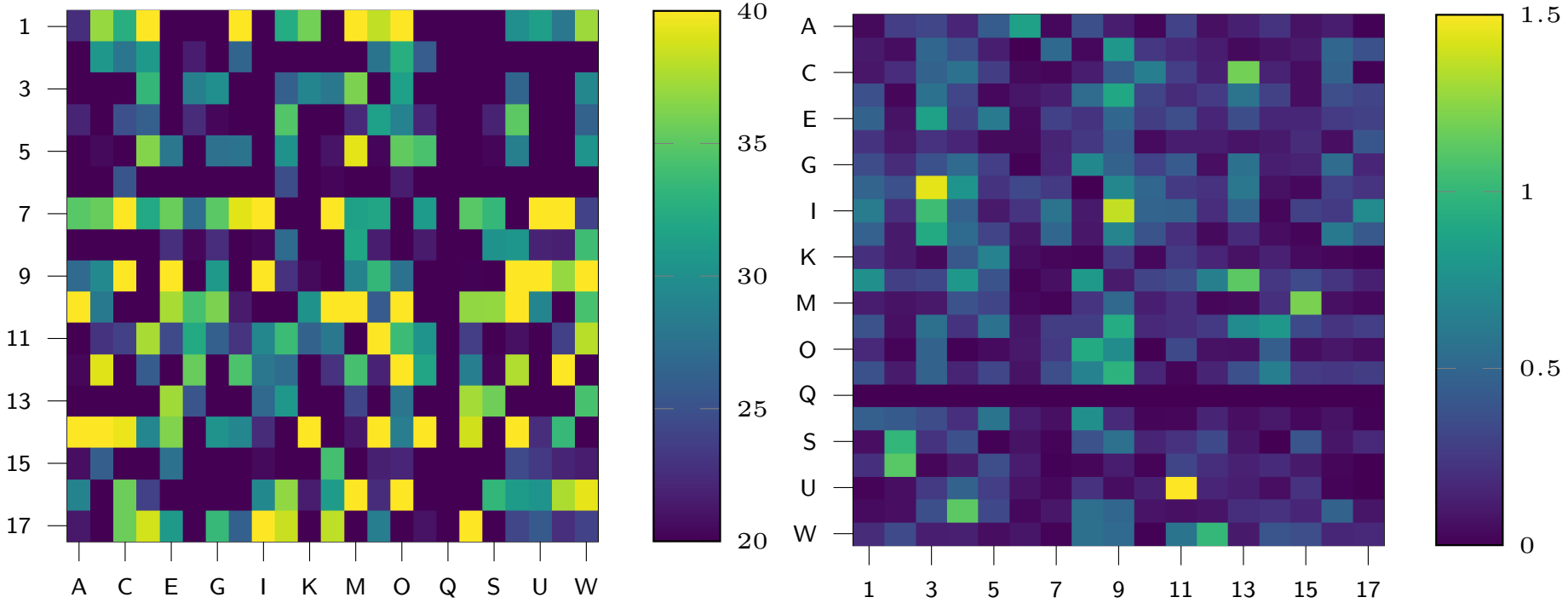
(c) MSLL



(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

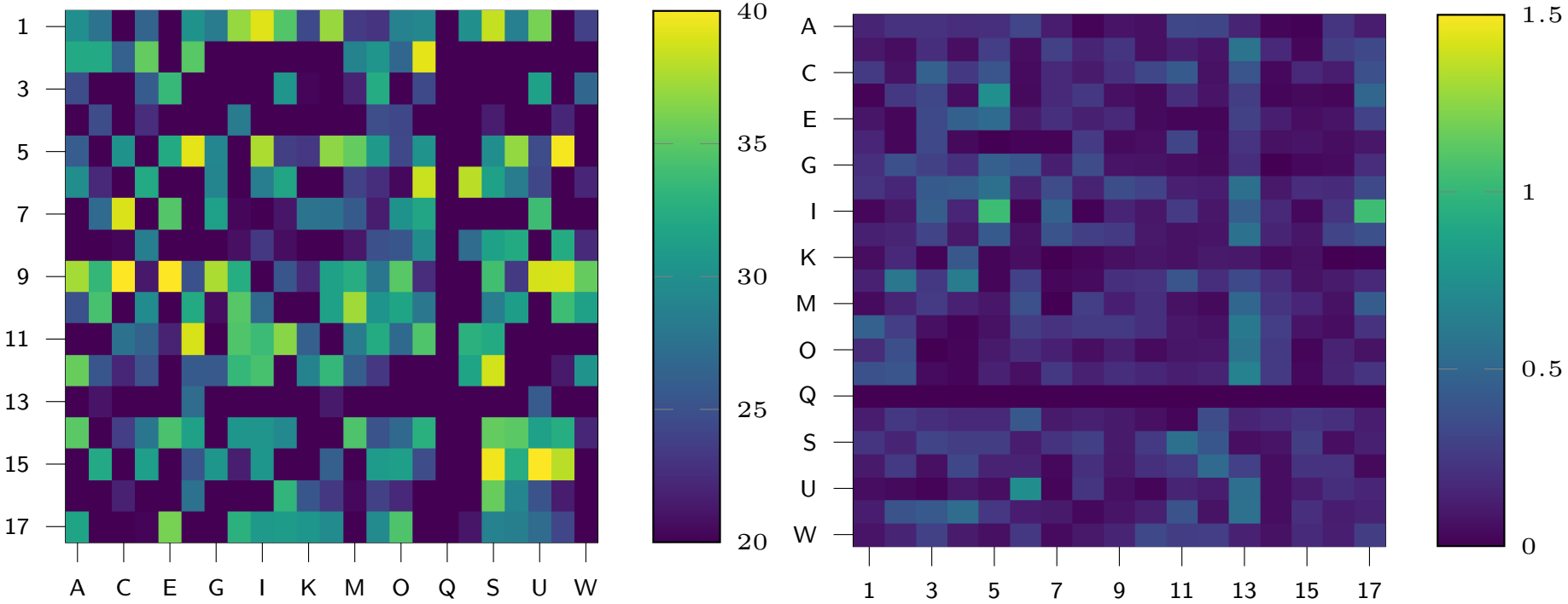
Lengthscale and $a_{d,q}$ Values (H=1)



(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)

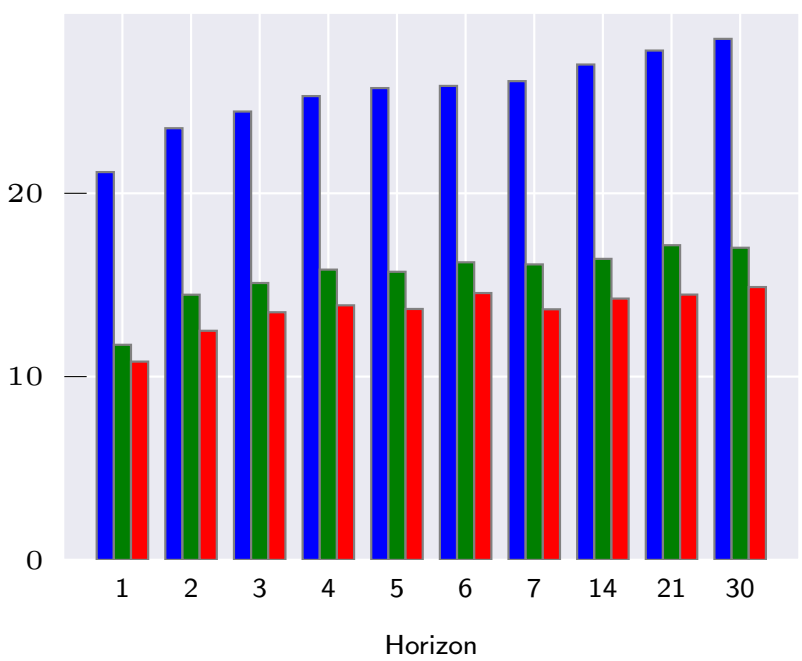


(a) (b)

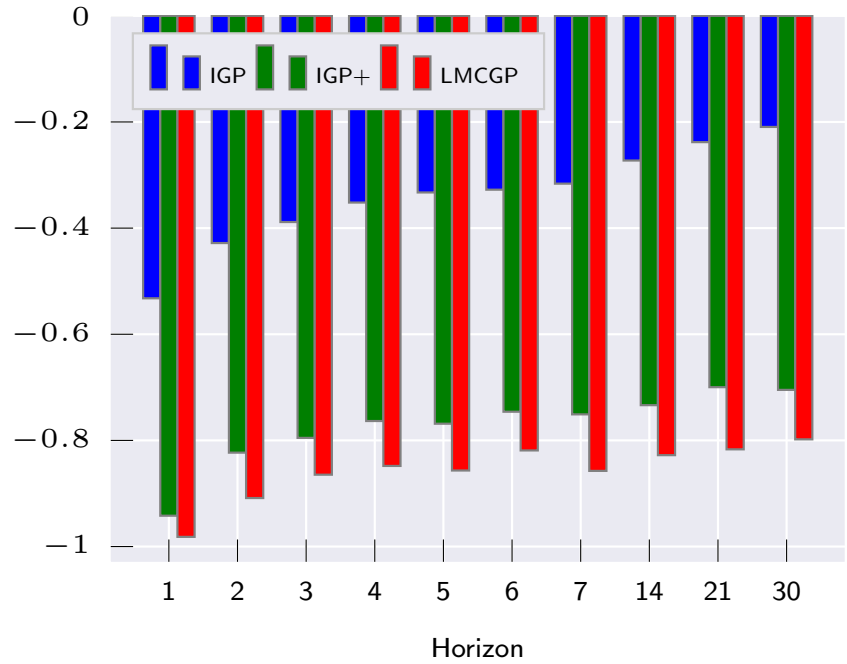
Figure: Lengthscale values of Input Features for Each Independent GP Model (left) and coefficients $a_{d,q}$ (right) for horizon $H = 30$.

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

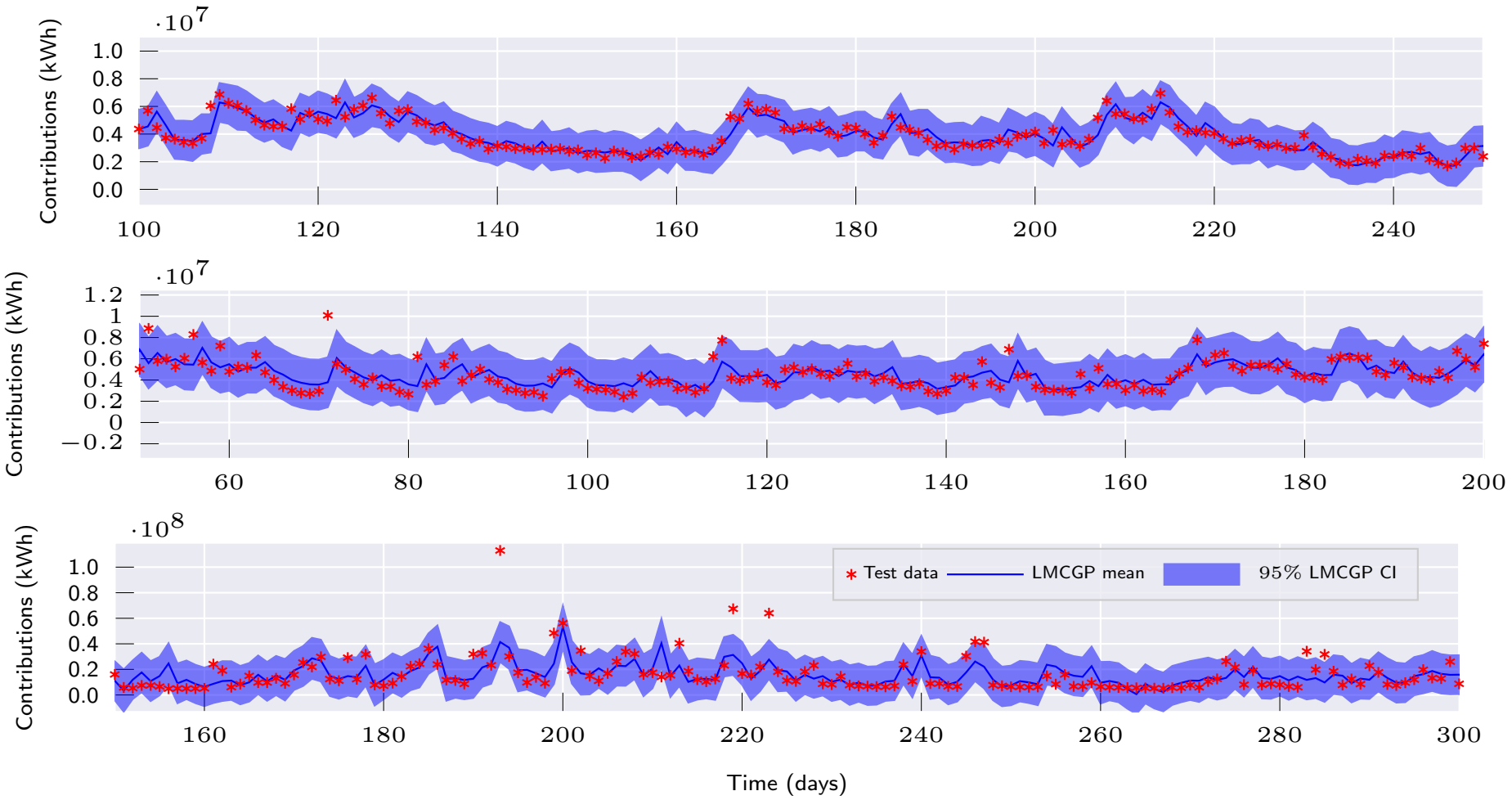


Figure: Test data for reservoirs T, A, and I (top to bottom) in one day ahead LMCGP model prediction ($H = 1$). Blue shade areas represent the 95% centered confidence interval for LMCGP prediction.

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**

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

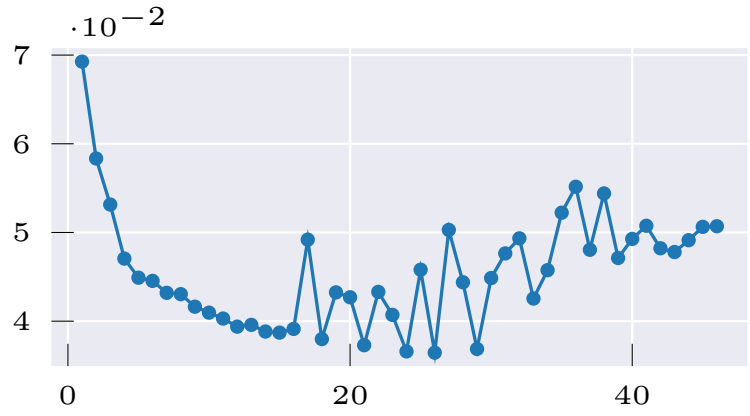
  In this formulation, $g_{d,1}(\cdot) = \cdot$, while $g_{d,2}(\cdot) = \ln(\exp(\cdot) + 1)$. We call this model Chd Normal.

- **Gamma Likelihood**

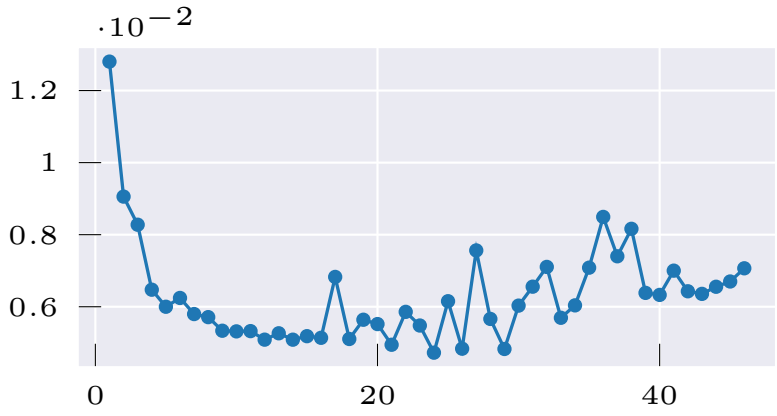
$$p(\mathbf{y} \mid \mathbf{f}) = \prod_{d=1}^{D} \prod_{n=1}^{N} \mathcal{G}amma\left(y_{d,n} \mid g_{d,1}(f_{d,1,n}), g_{d,2}(f_{d,2,n})\right)$$

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

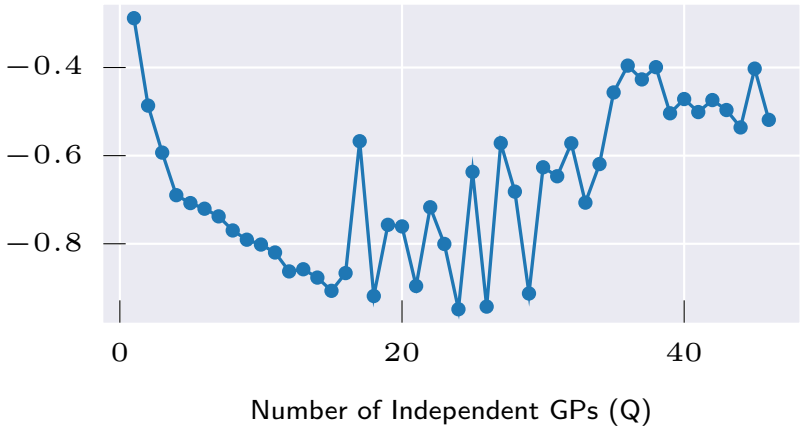
Tuning Q for Chd Normal



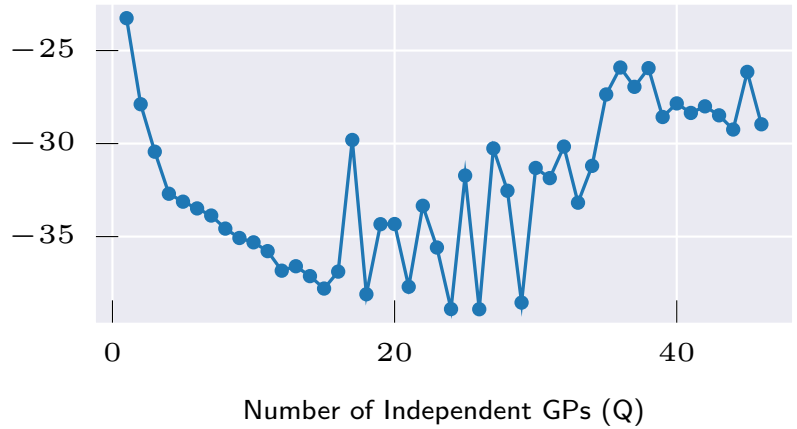
(a) CRPS



(b) MSE



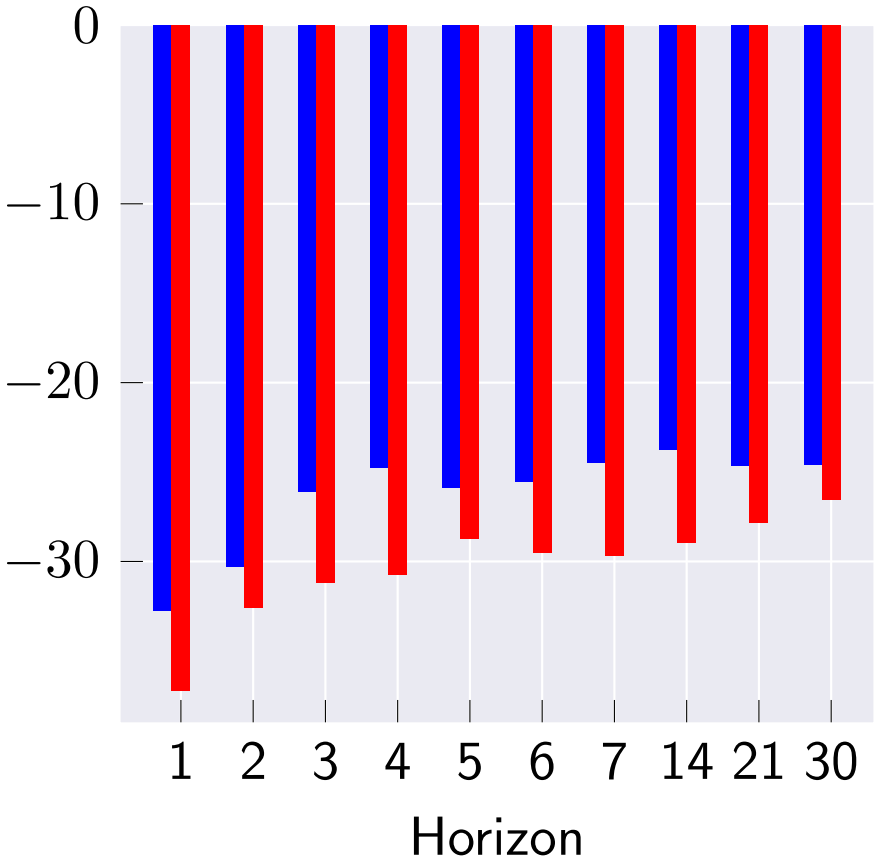
(c) MSLL



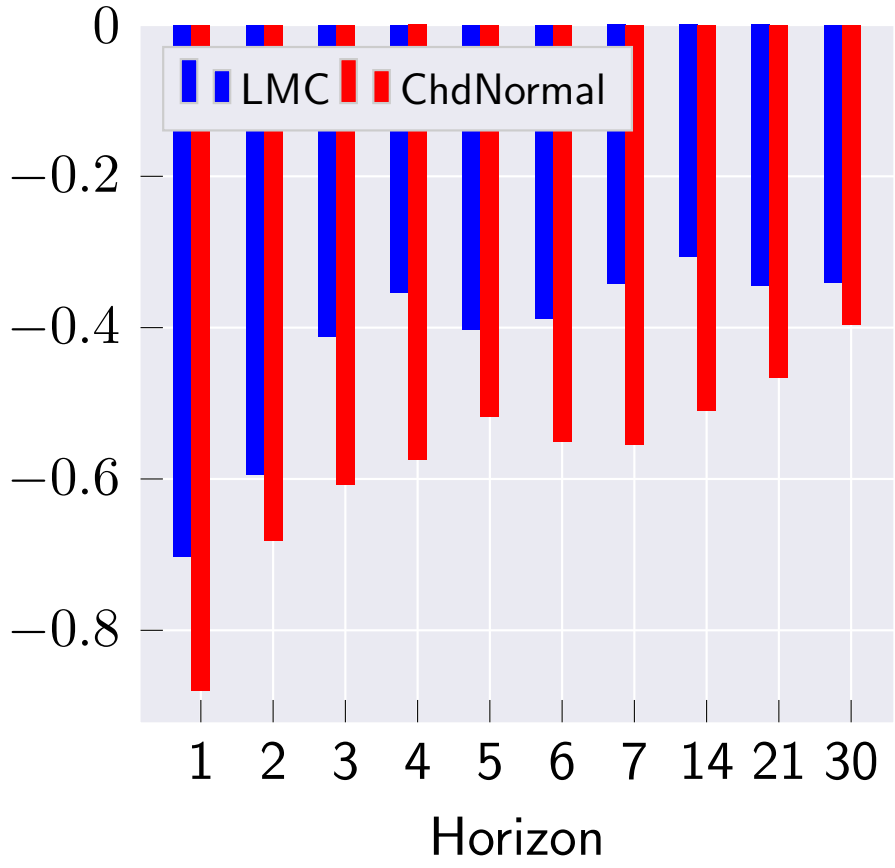
(d) NLPD

Figure: Performance metrics for ChdGP Normal model as a function of the number of independent GPs Q.

ChdGP Normal vs LMCGP



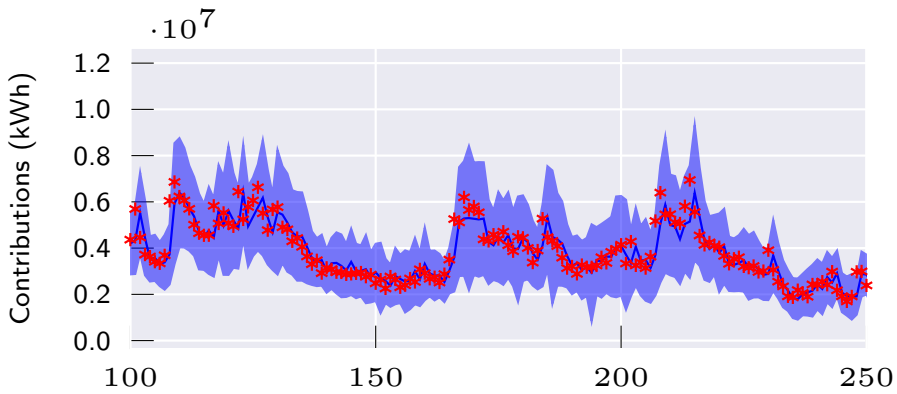
(a) NLPD



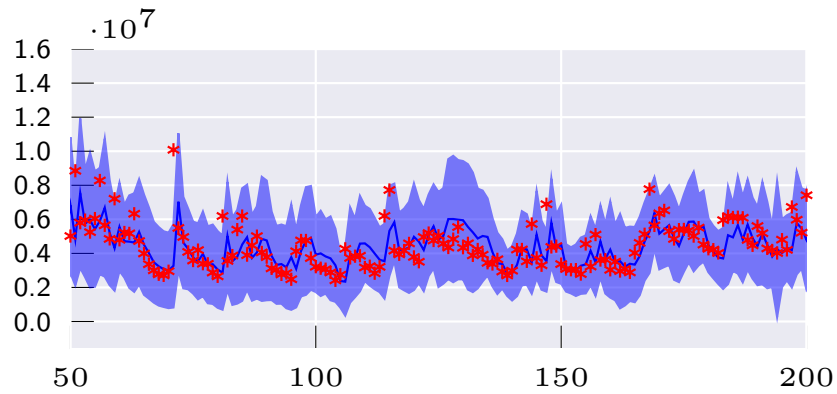
(b) MSLL

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

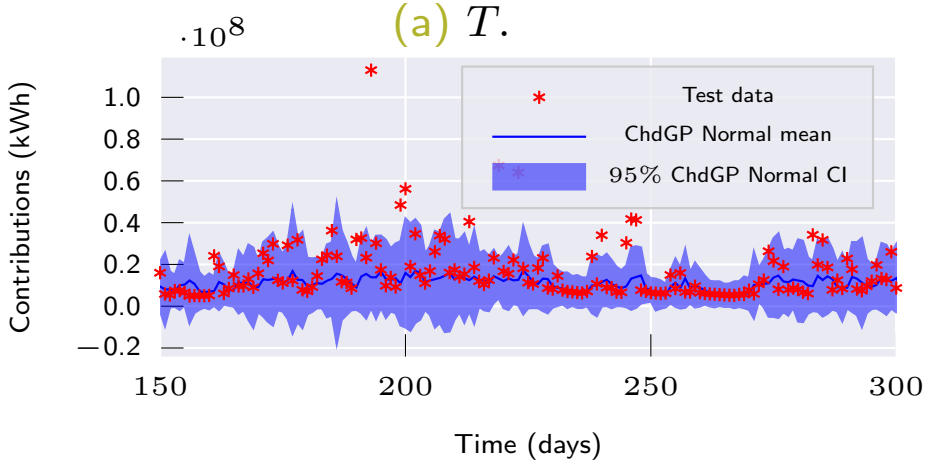
ChdGP Normal Forecasting



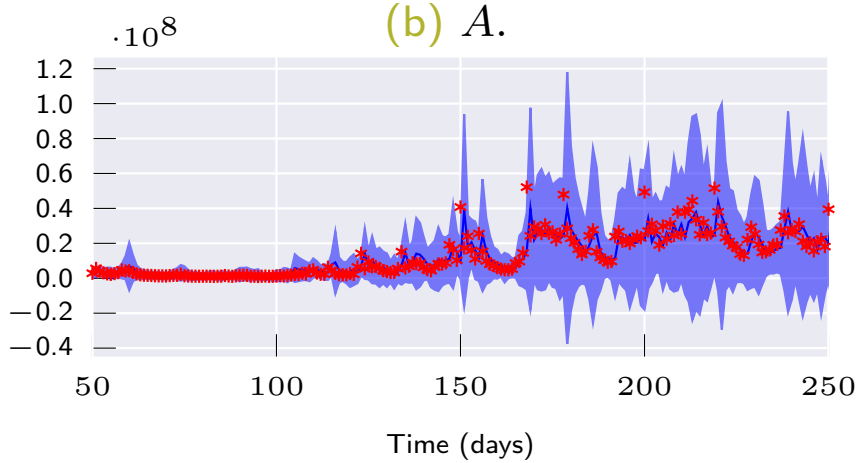
(a) T.



(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

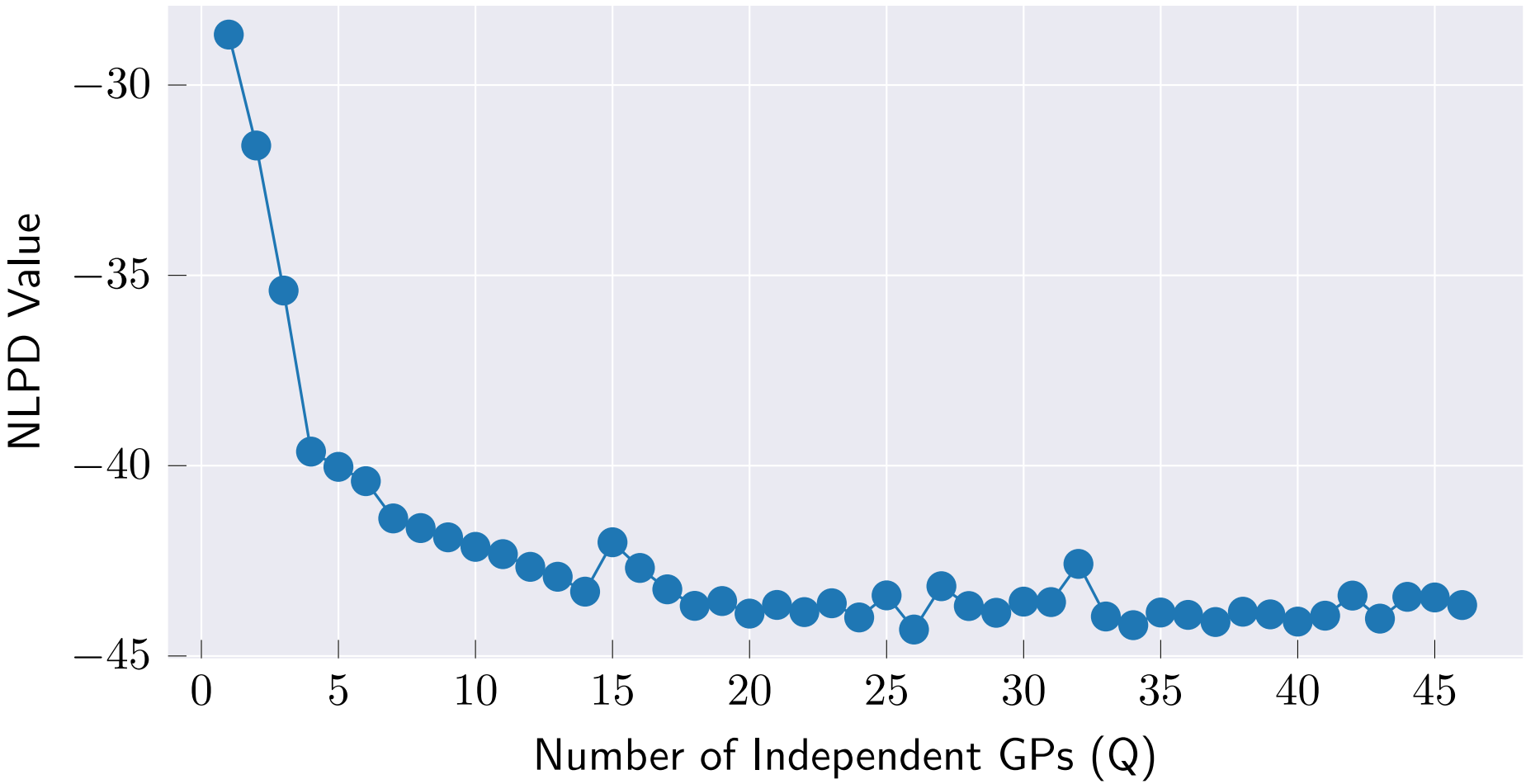


Figure: NLPD metric for ChdGP Gamma models as a function of the number of independent GPs.

ChdnGP Gamma vs ChdGP Normal

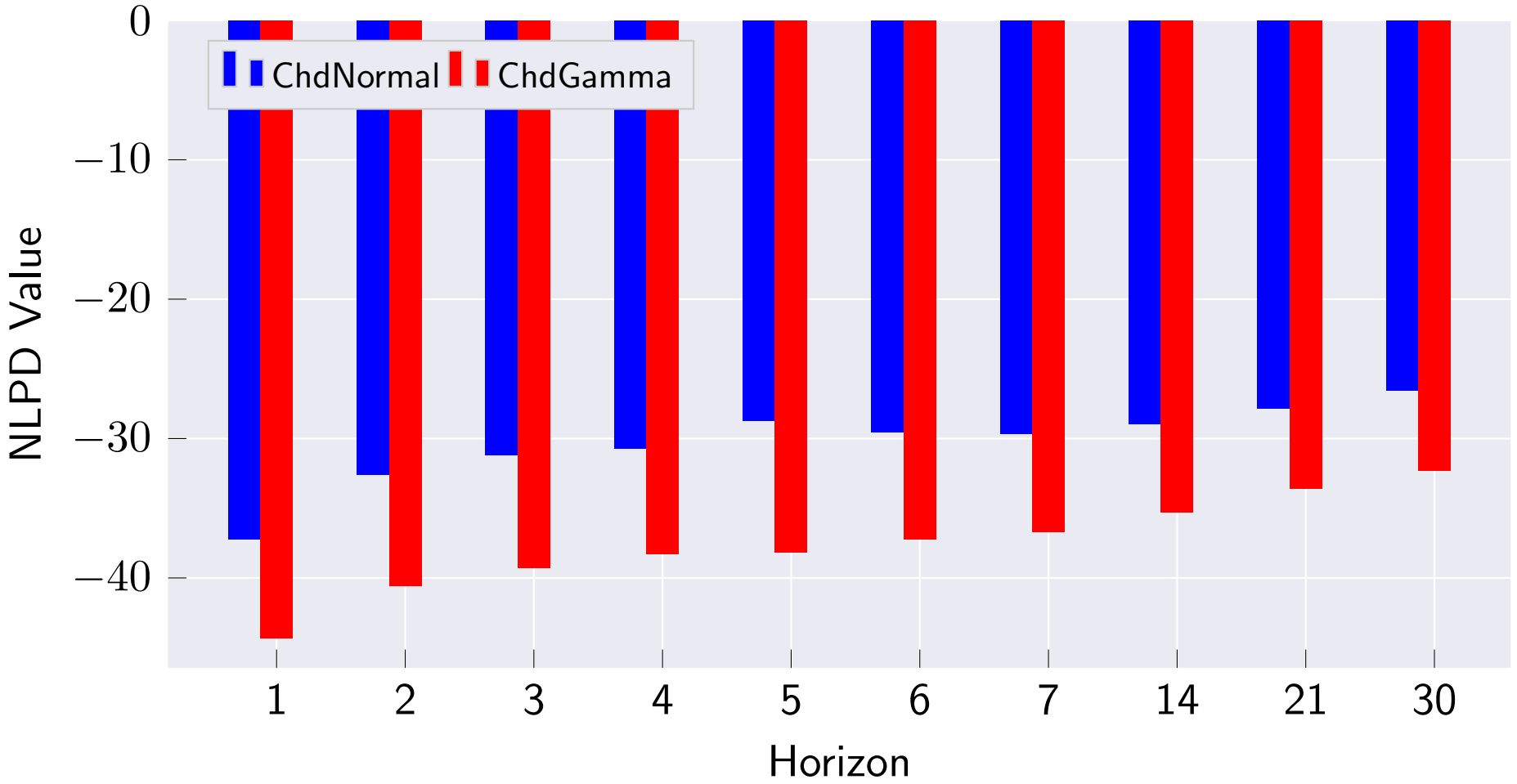
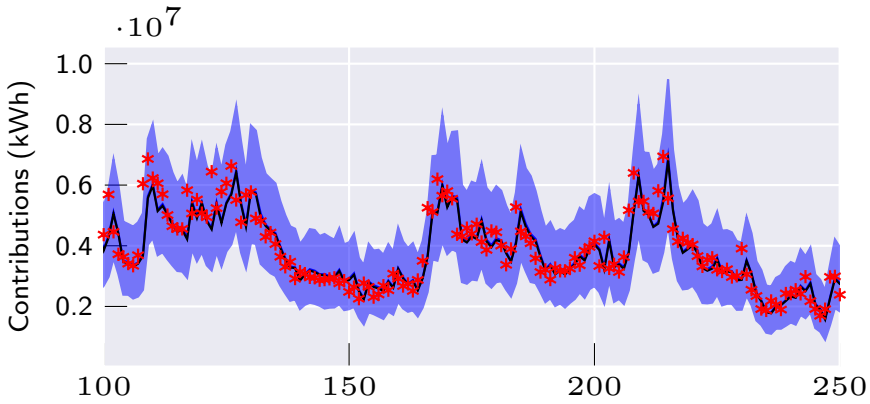
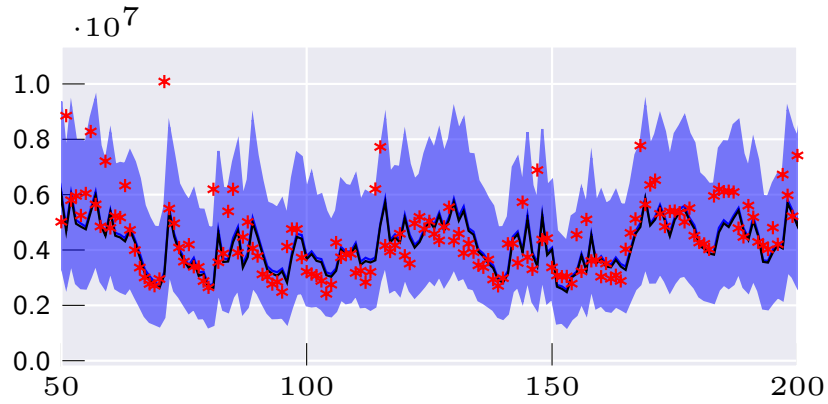


Figure: Comparison of NLPD metric across different prediction horizons for the ChdGP Normal, and ChdGP Gamma models.

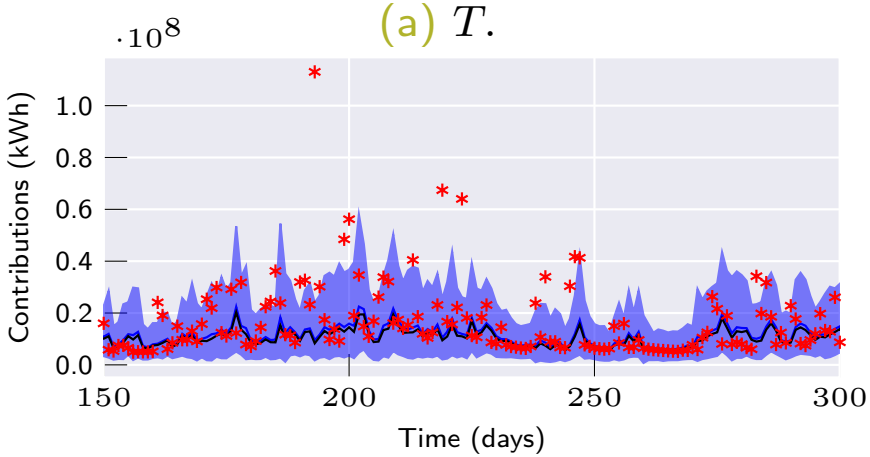
ChdGP Gamma Forecasting



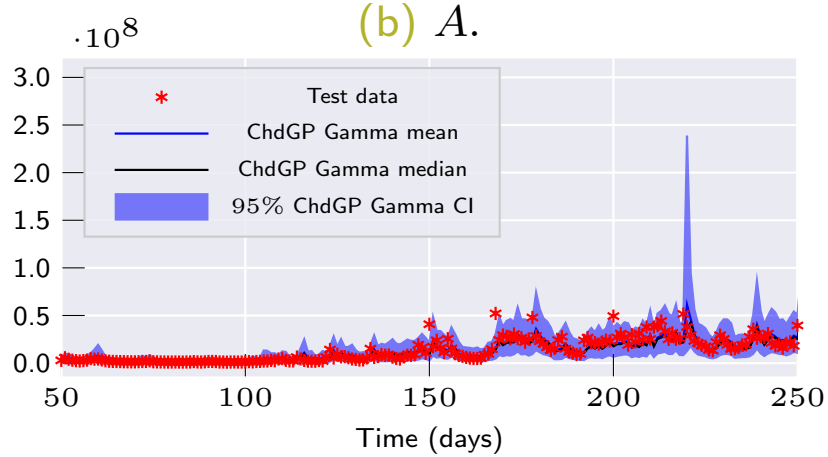
(a) T.



(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.