LONDON'S GLOBAL UNIVERSITY
Department of Computer Science

Probabilistic Solar PV Nowcasting

Eirik Aalstad Baekkelund

Supervisors: So Takao, Marc Peter Deisenroth

Faculty of Engineering

Department of Computer Science

University College London

Master Thesis

M.Sc. Data Science & Machine Learning

September 2023

Abstract

This thesis aims to establish a probabilistic framework for short-term forecasts of photovoltaic (PV) power production, with the overarching objective of contributing to optimized energy grid operations and diminishing reliance on fossil fuels. Concentrating on local-scale PV predictions within the UK, we employ Linear Coregionalization Model Multitask Gaussian Processes (LCM MT-GPs) to produce probabilistic forecasts for 2 and 6-hour horizons, capitalizing on the inter-dependencies among neighboring PV stations.

The project explores the effectiveness of LCM MT-GPs in two distinct scenarios: one employing shared temporal input across tasks and the other integrating task-dependent weather data. These scenarios utilize Kroenecker-structured and Hadamard Product covariance functions, respectively. To test the robustness of our approach, we conduct a comprehensive evaluation of model performance across different seasons, providing insights into its adaptability across diverse weather patterns.

Furthermore, we conduct an ablation study centered on Approximate Latent Force Models (ALFMs) to assess their potential applicability in PV output modeling. The study show that while ALFMs exhibit proficiency in synthetic settings, they encounter substantial challenges when confronted with the inherent uncertainty of Numerical Weather Predictions (NWPs) and the need for judicious kernel selection. This highlights the intricate nature of employing ALFMs for PV nowcasting, emphasizing the need for further research in this domain.

I dedicate this to my parents, Christine and Tor. Thank you for your unwavering belief in me, your endless support, and for instilling in me the values of proactively shaping one's own future and placing a strong emphasis on education. It is these principles that have shaped me into who I am today, and I owe most of my successes to you.

Acknowledgements

First, I want to express my gratitude to So Takao for his continuous guidance and support throughout my project, despite the challenging visa circumstances that compelled him to supervise from Tokyo. His dedication, even with an 8-hour time difference, late-night meetings, and the complexities that came with it, was greatly appreciated. I wish you the best of luck in your continued journey at Caltech.

Second, I am also deeply thankful to Professor Marc Deisenroth for granting me the opportunity to be part of the Statistical Machine Learning Group at UCL. His guidance and support, especially in the context of this project, were invaluable, and I greatly appreciate the time and effort he dedicated to assisting me while managing numerous other commitments.

A special thanks also go to Daniel Agusto for his technical expertise and support, stepping in admirably during So's absence.

Table of Contents

List of Figures

List of Tables

List of Algorithms

1 | Introduction

1.1 The Energy Crisis

As of July 2023, an alarming 6.6 million households in the UK are facing fuel poverty. This dire situation has been caused by a significant spike in energy costs, with the average household energy bill surging from £1,250 to £2,500 between October 2021 and October 2022 [7]. This sharp rise in energy expenses did not only fuel inflation but also intensified the prevailing cost-of-living crisis.

It is important to recognize that such crises rarely form in isolation but emerge from a complex interplay between various factors. Firstly, the aftermath of the Brexit transition in January 2021 saw the UK Emissions replace the European Trading Scheme leading to increased price volatility and market uncertainty in the energy sector [8]. Secondly, reduced wind speeds in 2021 had a significant impact on the proportion of wind energy production in the UK, where wind energy accounts for 26.8% of total energy generation, which is much higher than in mainland Europe [9]. Thirdly, the outbreak of conflict between Ukraine and Russia in February 2022 triggered a global scarcity of available gas, which happens to be the UK's primary energy source [10]. Collectively, these events played a major role in the energy price surge witnessed between 2021 and 2022.

These incidents underscore the degree to which the energy market depends on gas and wind resources, emphasizing the vulnerability of the UK's energy prices to fluctuations in these sources. The escalating energy crisis in the UK highlights the pressing need to explore alternative and stable energy resources. In this context, the relevance of PV forecasting become evident, as it can contribute to a more stable and sustainable energy grid in the face of these challenges.

1.2 The National Grid

The National Grid, managed by the National Grid Electricity System Operator (NG ESO), ensures a balance between electricity supply and demand in the UK [11]. This is achieved through the Balancing Mechanism (BM), an open market where electricity market participants collaborate. The main objective of the NG ESO is to analyze bids in the market to assess electricity demand and then manage supply accordingly.

The objective of the NS EGO is to keep its system frequency at $50Hz$, which is achieved by estimating the real-time equilibrium between supply and demand through the BM. To prevent abrupt frequency changes caused by supply and demand imbalances, the NG ESO maintains spinning reserves relative to the uncertainty in the BM. A rise in bid and offer expenses in the BM has been highlighted as a prominent concern [12], especially in scenarios where:

- Supply exceeds demand, leading to a frequency drop. Positive reserves in the BM act as a safety buffer.
- Demand exceeds supply, causing a frequency spike. Negative reserves in the BM can be disconnected from the grid for stability.

The NG ESO manages capacity regulations to address these grid imbalances, with costs increasing as energy sources become less predictable [13]. To handle this, the BM allows for securing short-term regulating reserves, requiring short-term reliable resources.

In this project, we will explore how short-term probabilistic forecasts for photovoltaic (PV) energy generation can benefit the national grid. This aims to reduce reliance on volatile fossil fuel markets and mitigate future energy price fluctuations through reliable forecasts of PV power generation, aligning with the need for short-term reliable resources to address grid imbalances.

1.3 Solar Power Nowcasting

Solar power nowcasting refers to predicting PV energy the "next few hours" at temporal resolutions of 5-15 minutes. It is estimated that accurate uncertainty quantification of PV nowcasts can reduce CO2 emissions by 100,000 tonnes in the UK annually, thereby reducing the magnitude of spinning reserves relying on natural gas [14].

The PV forecasting literature is typically categorized into three groups - physical, statistical, and hybrid forecasts - depending on the forecasting method used [15].

Physical methods converts solar radiance to PV output in association with weather variables. They are simpler when using solar irradiance, but suffer from increased complexity when adding additional weather parameters [16]. A concern regarding physical PV nowcasts are the uncertainty related to the inherent volatility of local weather conditions (e.g. cloud coverage) [17]. Precisely forecasting these weather parameters can be challenging, which pose a weakness for physical approaches.

Statistical methods rely on persistence or random time series. They construct models by connecting input variables to PV output power. These methods use two approaches: direct and indirect forecasting. Direct forecasting models leverage connections between input variables and PV output using historical data [18]. In contrast, indirect forecasting methods involve two steps: modeling weather factors influencing PV output power and converting this modeling into predicted PV output power [18, 19, 20, 21].

1.4 Data Providers

1.4.1 Open Climate Fix

Open Climate Fix (OCF) is a company dedicated to bridging the gap between the energy industry and cutting-edge technology to reduce CO2 emissions through AI-driven solutions in the energy sector. One of their key objectives is improving PV power forecasting.

In March 2022, OCF achieved a significant milestone by publishing forecasts that outperformed NS EGO's predictions by a factor of 2.8 [22]. This achievement resulted from the fusion of massive datasets, including high-resolution satellite imagery captured at 5-minute intervals, PV data collected from numerous PV systems, and numerical weather predictions (NWPs).

The comprehensive dataset encompassed solar PV data from around 1000 UK stations [23]. NWPs, critical for weather insights, were obtained from the UK Met Office's "UKV" model, offering a resolution of 128km x 128km [24]. Additionally, EUMETSAT provided satellite imagery depicting cloud coverage over the UK at a similar resolution as NWPs [25].

OCF identified OptiFlow as their most successful model, predicts satellite images for the next two hours using optical flow at a 30-minute temporal resolution. These forecasts are processed by a Perceiver model with attention mechanisms [26]. The model also integrates past four-hour NWP data and recent half-hour PV output from each station (up to 128 stations), combining spatial and temporal components. The Perceiver's output feeds into a multi-layer perceptron, generating predicted distributions for the target PV station at each time step.

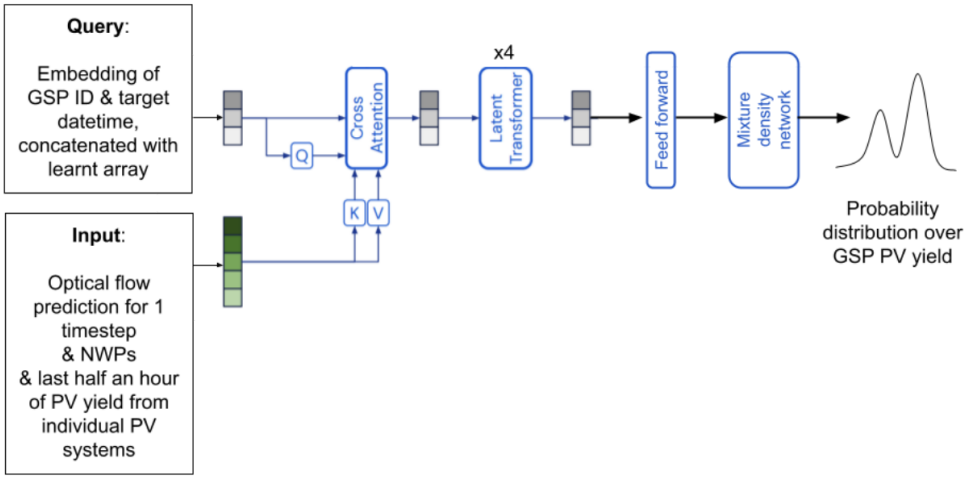


Figure 1.1: The OptiFlow architecture designed by OCF (adapted from Jaegle et. al [1]). The plot is taken from Open Climate Fix's publication [2].

1.4.2 Meteomatics

We often face the challenge of computational complexity for predicting weather variables at high spatial resolution [27] as they inherently require higher resolution to capture local dynamics. Meteomatics is a weather service provider specializing in weather forecasting at high spatial- and temporal resolutions. Their approach leverages substantial computational power, boasting a network of 40,000 CPUs and advertise a low model error within the first 6-12 hours [28], particularly relevant for PV nowcasting.

Meteomatic's EURO1K model [28] integrates data from the European Centre for Medium-Range Weather Forecasts Integrated Forecasting System (ECMWF-IFS) at a 10km x 10km resolution. Further, it combines the data with NASA's terrain model and use down-scaling to predict weather variables in spatial resolutions up to 90m x 90m at one hour intervals. As PV generation has an immediate dependency on weather conditions forecasting local meteorological variations is pivotal. Our goal is to leverage their EURO1k model's NWPs for variables such as cloud coverage and solar radiation to improve the reliability of our PV nowcasts at a localized scale.

1.5 Project Aims and Contributions

We will aim to complement OCF's work instead of improving their current predictive accuracy of PV nowcasts. We focus on Gaussian Process (GP) methodologies that offer a more interpretable probabilistic framework compared to neural networks. The thesis, we will focus on:

1. Probabilistic PV nowcasts for individual stations by capitalizing on the inter-correlation between neighboring PV stations.
2. Evaluating the effect weather data has on predictive accuracy.

To achieve this, we will leverage Linear Coregionalization Model Multitask Gaussian Process (LCM MT-GP) models. Inherently, they assume a shared trend for its tasks. Therefore, we aim to capitalize on shared trends among neighboring PV systems, exploring how they can effectively predict PV output by taking advantage of these shared trends. We will focusing on two distinct settings. One setting leverages shared temporal input, while the other incorporates weather parameters specific to each PV station. These settings involve two different approaches to constructing our covariance function. First, we incorporate a Kronecker structured covariance function when we consider the shared temporal domain of our PV stations. Second, we utilize Hadamard product kernels when considering weather parameters specific to each PV stations.

In PV nowcasting, it is important to emphasize the need of fast model training and re-calibration to account for recent observations of PV output and observed weather conditions. LCM MT-GPs exhibits a computational complexity of $\mathcal{O}(N^3M^3)$ and memory requirements of $\mathcal{O}(N^2M^2)$, where N denotes the number of observations, and M is the number of tasks. To mitigate these challenges, we utilize

- the inversion property of a Kroenecker matrix. Given two matrices $A \in \mathbb{R}^{N\times N}$ and $B \in \mathbb{R}^{\mathbb{M}\times\mathbb{M}}$, we have $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$. This reduce the computational

complexity to $\mathcal{O}(N^3 + M^3)$.

- inducing points [29] and data sub-sampling for the Hadamard covariance function.

Given that PV output is directly influenced by its recent weather conditions [17], we will focus on training on recent observations (1 week of data). PV generation exhibit short-term and seasonal differences, such as changes in the sun's position, day lengths, this impacts the corresponding PV output (see Figure 3.11). These seasonal dynamics influence kernel parameters, such as lengthscales and periodicity. Similarly, week-to-week patterns can undergo quick changes, which consequently affect predictions.

In cases where we rely solely on temporal input, we limit the nowcast horizon to 2 hours to better align with the shorter-term temporal dependencies of PV generation. When including weather parameters it enables us to account for longer-term influences these observations have on PV generation. As such, we extend the nowcasting horizon to 6 hours in this setting.

The contributions of our project highlights two key achievements. Firstly, in the temporal setting the Kroenecker LCM MT-GP slightly surpasses our statistical benchmarks in terms of Negative Log Probability Density (NLPD) and exhibits competitive performance in terms of Mean Absolute Error (MAE). Secondly, the incorporation of weather data in the Hadamard LCM MT-GP leads to results that on average marginally surpass both machine learning and statistical benchmarks for 2-hour and 6-hour nowcasts for both MAE and NLPD.

2 | Background

This chapter aims to provide the reader with the theoretical background of this project. First, we will cover time series analysis with a brief introduction to statistical -and machine learning methods. Then, we will give an introduction to the relevant aspects of GPs considered for this project.

2.1 Time Series Forecasting

Time series data, denoted as $\{y_t\}_{t=1}^{T}$, represents a sequence of observations that track the evolution of a process over time. Forecasting in time series involves predicting how this process will continue into the future, based on historical observations. There are two primary approaches to time series forecasting:

1. **Statistical models:** models who rely solely on the observed time series data to make predictions.
2. **Machine Learning models:** models who incorporate exogenous regressors, assuming their influence can enhance forecasting accuracy.

In the realm of statistical models, common techniques like Vector Autoregression (VAR) and Exponential Smoothing (ES) are notable for their simplicity and interpretability, making them well-established tools. In contrast, machine learning models, such as Multilayer Perceptrons (MLPs), tend to have more complex architectures and nonlinear computational complexity, which makes them less interpretable and computationally more demanding.

The debate over the superiority of machine learning (ML) versus statistical models in time series forecasting has attracted significant attention. A comprehensive review by Makridakis et al. [3] confronted this debate by addressing issues often found in machine

learning publications. Some of the issues they related to the publications included limited sample sizes in experiments, short forecasting horizons, and only benchmarking against other machine learning frameworks.

Their own experiments, based on 1045 monthly time series from the M3 competition [30], revealed that statistical methods exhibited superior generalization capabilities compared to machine learning models. Thereby, they address a general bias to popular belief assuming the latter has superior predictive power. As such, we will report our findings in this paper by testing on a large collection of time series, and benchmark them against both statistical- and ML methods.

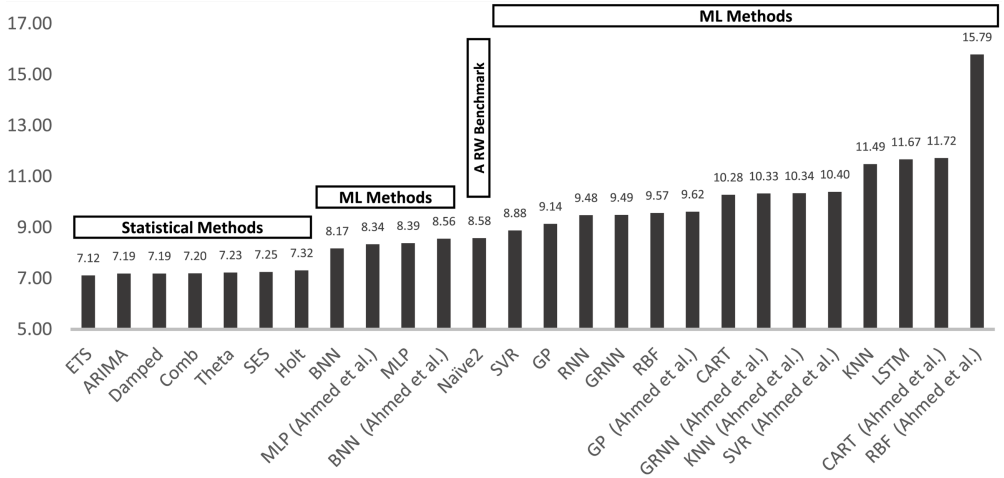


Figure 2.1: A comparative result of time series forecasting models from [3]. We observe the statistical models beating the ML counterparts. Performance was evaluated using the symmetric Mean Absolute Percentage Error (sMAPE). It is defined as sMAPE $= \frac{2}{T}\sum_{t=1}^{T}\frac{|y_t-\hat{y_t}|}{|y_t|+|\hat{y_t}|}\cdot 100\%$ with T denoting the forecast horizon and y_t and $\hat{y}_t$ denotes the true- and predicted value at time step t.

2.2 Gaussian Processes

In this section we will cover the most important aspects of Gaussian Processes and briefly introduce concepts relevant to Multitask Gaussian Processes.

2.2.1 Definition

A Gaussian process (GP) represents a stochastic process, where each finite set of random variables follows a joint Gaussian distribution [31]. Stochastic processes enable the assignment of probability distributions to functions instead of point estimates. As the Gaussian distribution, GPs are fully characterized by a mean function and a covariance function. Typically, they are denoted by

$$\mu(\mathbf{x}) = \mathbb{E}[(f(\mathbf{x})] \tag{2.1}$$

$$k(\mathbf{x}, \mathbf{x}^*) = \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}^*) - \mu(\mathbf{x}^*))]. \tag{2.2}$$

The mean function $\mu(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}$ captures the expected value of the process at each point. Equivalently, the covariance function $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ models the similarity between data points in the input space. Any stochastic process, f, following a GP with mean function and covariance function (as given above) is denoted

$$f(\mathbf{x}) \sim GP(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}^*)). \tag{2.3}$$

2.2.2 Inference with Gaussian Processes

When modeling functions with GPs, we hold a Bayesian view to how past observations can be used to infer the probability of future observations. To illustrate this, consider a dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$ where $\mathbf{y} = [y_1, \ldots, y_N]^T \in \mathbb{R}^N$ represents the observed target variables, with N being the number of observations, and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times D}$ representing the corresponding feature variables. If we assume conditional independence among the latent features generated by a Gaussian process, which we denote as $f(\mathbf{X})$, it

allows us to factorize the likelihood function as

$$p(\mathbf{y} \mid \mathbf{X}) = \prod_{i=1}^{N} p(y_i \mid f(\mathbf{x}_i)). \tag{2.4}$$

This likelihood captures the probability distribution of the observed target variables $\mathbf{y}$ given the associated feature variables $\mathbf{X}$. Each data point's likelihood is modeled as a conditional probability $p(y_i \mid f(\mathbf{x}_i))$, with an underlying assumption of independence between each data point.

In GPs, the latent function, $f(\cdot)$ (Equation 2.3), enables us to incorporate prior knowledge regarding the data through its covariance function k, which will be elaborated upon in Section 2.3. Under this framework, we need to compute the marginal likelihood (see Equation 2.6) for making predictions. This involves finding the posterior of our GP prior, which comprises three components: the prior, the likelihood, and the model evidence. Utilizing Bayes' rule, the posterior distribution can be defined as follows:

$$p\left(\mathbf{f} \mid \mathcal{D}, \theta\right) = \frac{N\left(\mathbf{f} \mid \mathbf{0}, \mathbf{K}\right)}{p\left(\mathbf{y} \mid \mathbf{X}, \theta\right)} \prod_{i=1}^{N} p\left(y_i \mid f(\mathbf{x}_i)\right). \tag{2.5}$$

On the left-hand side, the posterior function is represented, where $\mathcal{D}$ denotes the data as mentioned earlier, and θ accounts for all the model parameters. On the right-hand side, the leftmost term in the numerator corresponds to our GP prior, $\mathbf{f} = [f(x_1), \ldots, f(x_N)]$, following a joint Gaussian distribution. The covariance of our prior, k, consists of entries $\mathbf{K}_{i,j} = k(x_i, x_j)\ \forall i, j = 1, \ldots, N$. The denominator represents our marginal likelihood, also known as the model evidence, defined by:

$$p(\mathbf{y} \mid \mathbf{X}, \theta) = \int p(\mathbf{y} \mid \mathbf{X}, \mathbf{f}, \theta) p(\mathbf{f} \mid \mathbf{X}, \theta) d\mathbf{f}. \tag{2.6}$$

When the likelihood of our GP is non-conjugate it leads to an intractable integral. Con-

sequently, we cannot obtain the marginal likelihood nor the posterior distribution in closed-form. Section 3.2.5 will delve into a discussion of how variational inference can provide approximations for the intractable distributions.

2.2.3 Gaussian Process Regression

In Gaussian Process Regression (GPR), we aim to predict a latent function $f(\mathbf{x}^*)$ given a test point $\mathbf{x}^*$. We call $f(\mathbf{x})$ a realization of the GP prior such that if we assume a Gaussian likelihood we have

$$\mathbf{y} = f(\mathbf{x}^*) + \varepsilon,\ \varepsilon \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right) \tag{2.7}$$

where ε is added Gaussian noise. Given a Gaussian likelihood, we can obtain a closed-form posterior (see Equation 2.5):

$$p(\mathbf{f}_* \mid \mathcal{D}, \mathbf{X}_*) \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \tag{2.8}$$

$$\boldsymbol{\Sigma} = k_{**} - k_*^T(k + \sigma^2\mathbf{I})^{-1}k_* \tag{2.9}$$

$$\boldsymbol{\mu_*} = k_*^T(k + \sigma^2\mathbf{I})^{-1}\mathbf{y} \tag{2.10}$$

where k is the covariance between training data, k_* is the covariance between train- and test data, and k_{**} is the covariance between the test data. Here, the inversion of $(k + \sigma^2\mathbf{I}) \in \mathbb{R}^{\mathbb{N}\times\mathbb{N}}$ scales to $\mathcal{O}(N^3)$, creating a bottleneck of GP applications to large datasets. However, in [29] they have shown how variational approximations can reduce the computational burden. We will elaborate on this idea in Section 3.2.5.

2.3 Kernels

The kernel, which is the covariance function of our GP, is used to expressing the expected behavior of our GP prior $\mathbf{f}$. We say the kernel is stationary if it only depends on the distance $\mathbf{x} - \mathbf{x}'$. We can then write our kernel as

$$k(\mathbf{x}, \mathbf{x}') \triangleq k(\tau), \ \tau = \mathbf{x} - \mathbf{x}' \tag{2.11}$$

Bochner's theorem provides an understanding for characterizing valid stationary covariance functions in Gaussian Processes [31]. It establishes a direct link between positive-definite kernels and continuous positive-definite functions using the Fourier-Transform:

$$k(\tau) = \int_{\mathbb{R}^D} e^{2\pi i \mathbf{s}\cdot\tau} d\mu(\mathbf{s}) \tag{2.12}$$

2.3.1 Kernel Functions

For this project, we will be considering the Matérn, Periodic, and Squared-Exponential (SE) covariance functions. First, we will introduce their definitions and briefly discuss their relevance to modeling PV generation using GPs.

2.3.1.1 The Squared Exponential Kernel

The SE covariance function is defined by

$$k_{\text{SE}}(\tau) = \sigma_f^2 \exp\left(\frac{-\tau^2}{2\ell^2}\right) \tag{2.13}$$

where ℓ is the characteristic length-scale parameter, which controls the correlation between features by re-scaling the inputs $\mathbf{x}$ by a factor of $\frac{1}{\ell}$ before computing the function. The lengthscale effectively controls the correlation between inputs. Larger lengthscales

result in closely correlated outputs with functions that become increasingly linear over the input range. Short length-scales decorrelates the function values, resulting in a varied and non-linear function over the input range.

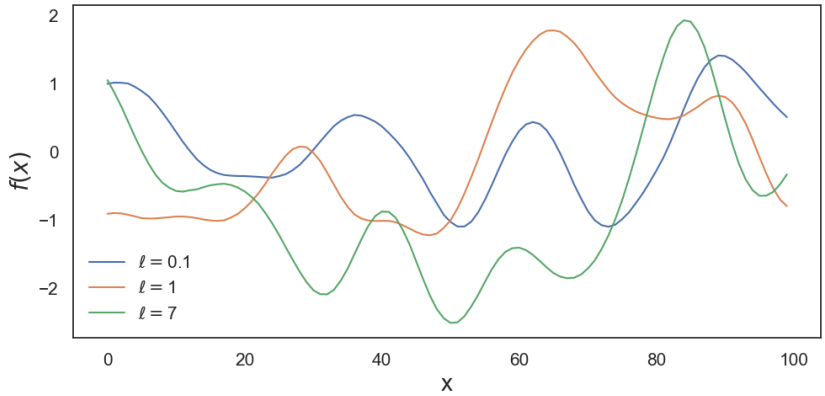


Figure 2.2: A demonstration of how the lengthscale magnitude effects the resulting function variance across the input range. The larger the lengthscale, the more linear is the function values from the GP.

The term σ_f^2 is the signal variance, which determines the amplitude of the function we model. This can be demonstrated by looking at the covariance between one input points against itself:

$$\mathbb{V}(f(\mathbf{x})) = k(\mathbf{x}, \mathbf{x}) = \sigma_f^2 k(\mathbf{x} - \mathbf{x}) = \sigma_f^2 k(0) = \sigma_f^2. \tag{2.14}$$

Thereby, the function amplitude at $\mathbf{x}$ corresponds to the square root of the signal variance with the majority of values likely falling within a $2\sigma_f$ range centered at 0 [32]. The SE kernel holds relevance to PV modeling when we consider weather parameters as input. The SE kernel assumes that the function being modeled is infinitely differentiable and exhibits a smooth behavior. For both PV output and weather data, we can experience significant fluctuations. By modeling the underlying smooth trends in weather conditions it enables the GP model to distinguish genuine signal variations from noisy fluctuations. Thereby, it can make our GP models robust to instances where PV values undergo rapid fluctuations (e.g. passing of cirrus clouds - thin clouds observed at high altitudes).

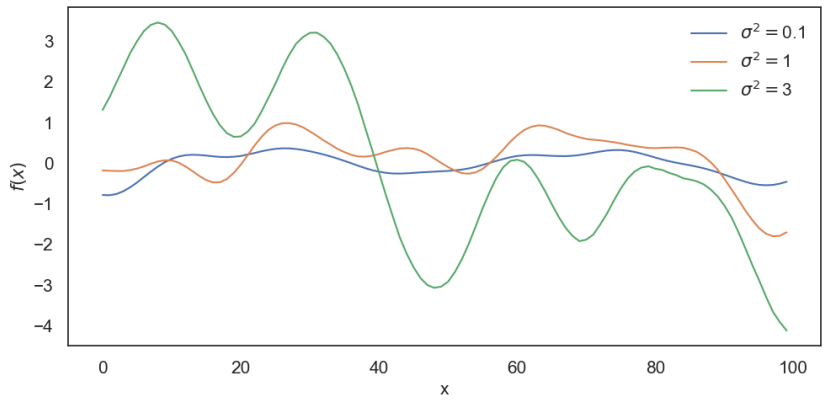


Figure 2.3: An illustration of how the signal variance changes the samples from a GP prior with mean function $\mu(x) = 0$ and a fixed lengthscale $\ell = 1$ using an SE covariance function.

2.3.1.2 The Matérn Kernel

The Matérn kernel is known for its ability to model functions with varying degrees of smoothness. It is defined by

$$k_{M_\nu}(\tau) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\tau}{\ell} \right) K_\nu \left(\frac{\sqrt{2\nu}\tau}{\ell} \right) \tag{2.15}$$

where ℓ and σ_f^2 is the same as above. ν is the order of the kernel, often considered at half-integer values $\nu = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}$. It controls the smoothness of the function. Intuitively, $\nu = \frac{3}{2}$ produce once-differentiable functions, $\nu = \frac{5}{2}$ produces twice-differentiable functions, etc. Additionally, the function K_ν is a modified Bessel function of the second kind (see [33]).

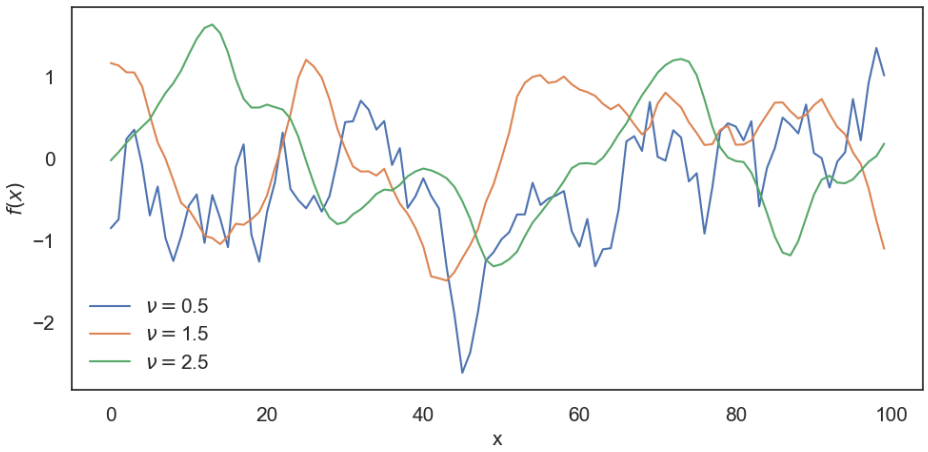


Figure 2.4: Illustrates of how ν affects the smoothness of a GP prior with a Matérn kernel having a fixed length-scale $\ell = 1$ and mean function $\mu(x) = 0$.

The Matérn kernel, showcased in Figure 2.4, is less smooth than the SE kernel. It offers the flexibility to capture intricate short-term fluctuations and long-term trends typically associated with PV power generation.

2.3.1.3 The Periodic Kernel

The periodic kernel aim to capture cyclic and repeating patterns in data, which holds particular significance in modeling PV nowcasting due to its inherent daily fluctuations caused by the sun's periodicity. The periodic kernel is defined by

$$k_{\text{Periodic}}(\tau) = \sigma_f^2 \exp\left(\frac{2\sin^2\left(\pi\frac{\tau}{p}\right)}{\ell^2}\right). \tag{2.16}$$

Here, the signal variance σ_f^2 controls the amplitude of the periodic function, while ℓ is the lengthscale as above. The additional parameter p models the periodicity of the sinusoidal curve.

2.3.2 Kernel Compositions

Kernel compositions are combinations of several kernel functions. They leverage the distinct attributes of individual kernels and combine them into more complex kernel functions. In this project, we will consider first order additive kernels [34], which generalize GPs and Generalized Additive Models. They are defined by

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \sum_{i=1}^{D} k_i(x_i, x_i'), \tag{2.17}$$

where D denotes the dimension of our input space and $i = \{1, \ldots, D\}$ specifies the assignment of a one-dimensional base kernel to each dimension. The resulting design choice of our kernel then consists of which kernel is assigned to each dimension. Indeed, one can also assign multiple base kernels to one input. Specifically, we can construct a kernel defined by

$$k(\mathbf{x}, \mathbf{x}') = \sum_{n=1}^{N} k_n(\mathbf{x}, \mathbf{x}'). \tag{2.18}$$

where k_n for $n = 1, \ldots, N$ denotes the number of base kernels we assign to the input. Additive kernels can combine individual kernels with separate attributes aiming to capture a distinct characteristic within the data.

Product kernels are used to model interactions of different attributes. They are formed by multiplying kernels together, which results in a composite kernel defined by

$$k(\mathbf{x}, \mathbf{x}') = \prod_{n=1}^{N} k_n(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') \cdot k_2(\mathbf{x}, \mathbf{x}') \cdots k_N(\mathbf{x}, \mathbf{x}'). \tag{2.19}$$

Here, k_n and N denotes the individual kernels and number of kernels, respectively. The operation $\cdot$ denotes the element-wise multiplication of the kernel values. Product kernels enable GP models to capture more complex interactions between data features or individ-

ual kernel characteristics. For instance, a kernel product between a linear- and periodic kernel enables the resulting kernel to capture a combination of linear and periodic trends.

2.3.3 The Quasi-Periodic Kernel

As PV data often exhibit a combination of short-term fluctuations and periodic variations, known as quasi-periodicity, it makes them challenging to model accurately. Building upon the concepts introduced in the previous section, we define a quasi-periodic kernel tailored for PV nowcasting. In doing so, we leverage the periodic kernel for the PV periodicity and the Matérn kernel for the inherit fluctuations associated with PV output. Specifically, we define our quasi-periodic kernel as followed:

$$k_{\text{QP}}(\mathbf{x}, \mathbf{x}') = \sigma_1^2 k_P(\mathbf{x}, \mathbf{x}') \cdot \sigma_2^2 k_{M_\nu}^{(1)}(\mathbf{x}, \mathbf{x}') + \sigma_3^2 k_{M_\nu}^{(2)}(\mathbf{x}, \mathbf{x}') \tag{2.20}$$

where k_P is a periodic kernel and $k_{M_\nu}^{(i)}$ is a Matérn kernel for $i = 1, 2$ with $\nu = \frac{3}{2}$. This composite kernel comprises three components: First, the periodic kernel, k_P, accounts for the daily trend observed in PV output corresponding to the sun's periodicity. By composing a Matérn product to the periodic kernel, enable the quasi-periodic kernel to capture smaller, irregular changes. Lastly, the added Matérn, $k_{M_\nu}^{(2)}$, accommodates larger short-term fluctuations, ensuring that the model can account for abrupt deviations from the underlying periodic trends. Figure 2.5 illustrates the behavior of the composite quasi-periodic kernel.

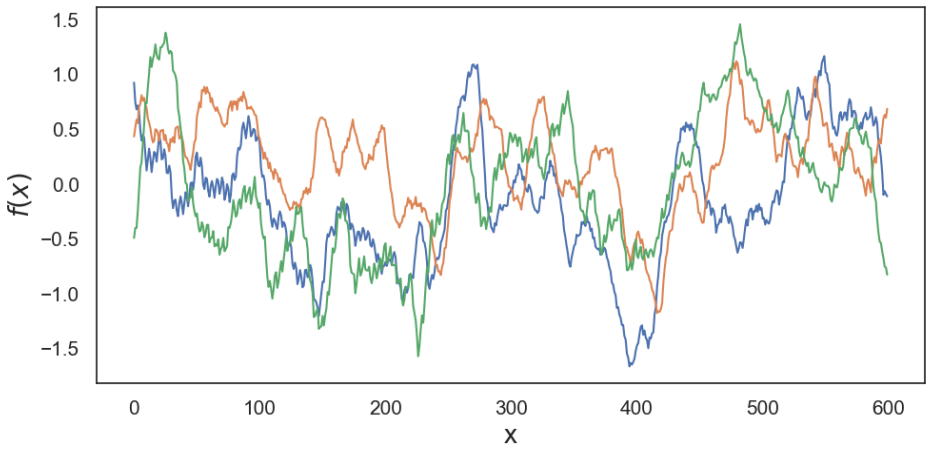


Figure 2.5: Three samples drawn from the GP prior with a quasi-periodic kernel with a mean function $\mu(x) = 0$. We can observe how the draws mimic the behavior often seen in PV values.

2.3.4 The Kronecker Product

In this project we are concerned with modeling the correlation between neighboring PV stations. A key component to this is the Kronecker product, denoted $\otimes$. It is a mathematical operation combining two matrices. If we consider matrices, $A \in \mathbb{R}^{N\times M}$ and $B \in \mathbb{R}^{P\times Q}$, their Kronecker product is given by:

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1N}B \\ a_{21}B & a_{22}B & \cdots & a_{2N}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1}B & a_{M2}B & \cdots & a_{MN}B \end{bmatrix} \tag{2.21}$$

The resulting matrix $Z = A \otimes B$ will have dimensions $R^{(NP)\times(MQ)}$. In the context of Multitask Gaussian Processes (MT-GPs), the Kronecker product provides a way of handling multiple tasks concurrently by introducing output covariances [35]. This enables the modeling of shared trends and relationships among different tasks. We will elaborate this further in Section 3.2.

2.3.5 The Hadamard Product

Another operation we consider for the correlation across PV stations is the Hadamard product, represented as $\odot$. It is another mathematical operation involving element-wise multiplication. If we let A, B be two matrices in $\in \mathbb{R}^{N \times M}$, then their Hadamard product is given by

$$A \odot B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1N}b_{1N} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2N}b_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1}b_{M1} & a_{M2}b_{M2} & \cdots & a_{MN}b_{MN} \end{bmatrix} \tag{2.22}$$

We can utilize the Hadamard product through element-wise interactions. This holds relevance for modeling dependencies among tasks in MT-GPs to capture task-interactions. We will come back to this in Section 3.2.3.

3 | Methodology

3.1 Datasets and Pre-processing

The PV data was provided by OCF [23]. It contained data from 2018 to November 2021 at a 5 minute temporal resolution across 1311 stations in the UK. For each system, we have access to their latitude and longitude, total capacity (in kilowatts), PV values, orientation, and tilt. Meteomatics provided weather data (Table 3.1) at the location of 300 selected systems sampled at a 1 hour resolution from 2018 to 2019.

To get a comparative measure for all stations, they were scaled to the relative percentage of their total capacity such that all values are contained within the interval [0,1]. Any systems reporting production during night hours were removed to avoid corrupted signals. Further, any system with >5% missing observations were removed. The remaining missing values were then linearly interpolated. Finally, we sliced the time series to only include observations between 08:00-16:00, which corresponded to daily trend of PV production giving 96 daily observations. Figure 3.1 demonstrates how there is a yearly-and daily seasonality in PV output.

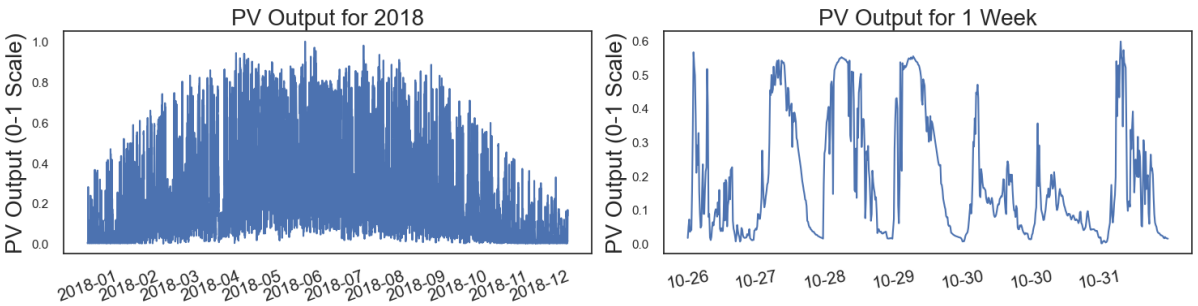


Figure 3.1: The left shows PV production for one system from 2018-2019. The right shows the production over 1 week in October. We can see how irregularities in daily productions make PV outputs noisy and consequently hard to predict.

3.1.1 Exploratory Data Analysis

In [6], the authors demonstrate the autocorrelation and seasonality of the PV time series for the same dataset, which show a clear daily and seasonal trend in the PV output. Additionally, they studied the effect of system variables by analyzing the correlation between PV output with respect to tilt, orientation, and the system capacity. They concluded that no apparent relationship could be established between the variables. As such, we decided to exclude them from this project.

Next, we will investigate the weather parameters provided from Meteomatics (given in Table 3.1):

Table 3.1: Weather Parameters

Variable	**Explanation**
Global Radiation	The total solar radiation received on a horizontal surface, including both direct and diffuse radiation. Measured in watts per square meter (W/m^2).
Diffuse Radiation	Solar radiation that is scattered in the atmosphere and reaches the Earth's surface indirectly. It contributes to the total radiation but does not come directly from the sun. Measured in W/m^2.
Temperature	The measure of the average kinetic energy of the air molecules in the atmosphere. Measured in Celsius (C°).
Humidity	The amount of water vapor present in the air as a percentage of the maximum amount the air could hold at a given temperature. Measured in Pascal (Pa).
Wind Speed	The rate at which air is moving horizontally past a certain point. Measured in meters per second (m/s).
Effective Cloud Coverage	A weighted sum of low, medium, and high-level cloud covers where clouds at low and medium heights have a higher weight as they block more radiation. Measured in Octas (0-8 scale).

The weather data from Meteomatics were given at a 1 hour temporal resolution. To account for the required temporal resolution of nowcasting (5-15 minutes), we linearly interpolated each parameter to obtain weather data at a 5 minute temporal resolution. Given two known values x_1 and x_2 where $t_1 < t_2$. Then, to estimate x for some $t \in (t, t')$ we interpolate as followed

$$x = x_1 \left(\frac{t_2 - t}{t_2 - t_1} \right) + x_2 \left(\frac{t - t_1}{t_2 - t_1} \right). \tag{3.1}$$

Subsequently, we applied the Savitzky-Golay (SG) filter [36] to enhance data precision while preserving the signal's underlying trends. This filter employs linear least squares to smooth data by convoluting neighboring data points with a low-degree polynomial. Specifically, the smoothed value $\tilde{x}_k$ at data point x_k is given by:

$$\tilde{x}_k = \sum_{i=-n}^{n} c_i \cdot x_{k+i} \tag{3.2}$$

where x_{k+i} are the neighboring data points around x_k, n is the half-width of the filter window, and c_i are the filter coefficients. We considered a filter window of $n = 12$ corresponding to our known values, and a polynomial degree $p = 3$. Despite the interpolation, weather data often exhibit intermediate fluctuations, making the temporal component the most precise measurement at the 5-minute interval.

To understand the effect of the weather variables, we investigated the correlations they had with PV output and their respective seasonal changes post interpolation. The resulting analysis revealed both clear seasonal changes in PV output and weather variables (see Figure 3.2 and 3.3), as well as significant correlations between PV output and all given weather parameters as observed by the correlation matrix in Figure 3.2.

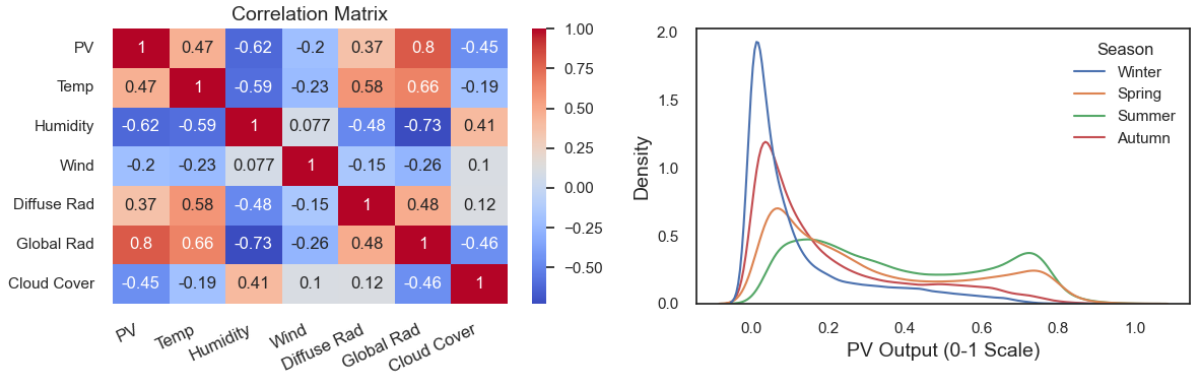


Figure 3.2: The left plot shows the correlations between weather variables. We can see that all the weather parameters share a significant correlation with PV values with a correlation coefficient $|\ \rho\ | \geq 0.2$. The right plot demonstrates how the seasons affect the distribution of PV values. From the plot we can infer that PV output significantly increases during spring and summer relative to the winter and autumn.

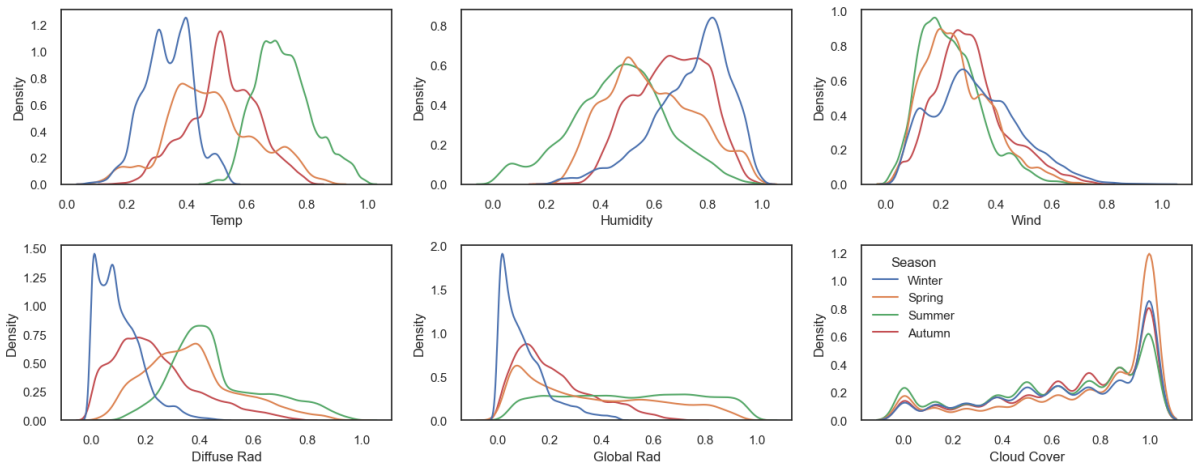


Figure 3.3: The figure demonstrates the 0-1 scaled seasonal distributions of our weather parameters. These seasonal variations highlight the dynamic of weather conditions throughout the year, which can significantly influence the effectiveness of these variables in predicting PV output.

To highlight the seasonal patterns of how the correlations between PV and weather changes, we looked at the kernel density estimation (KDE) of the weather variables with respect to PV output. We can see how the variability in weather patterns shift

with seasons, as displayed by Figure 3.4. Notably, the summer and spring have a larger spread in weather conditions and corresponding PV values, as evident from the wider distribution in the figure. The increased variability can pose challenges in accurately predicting PV output solely based on weather parameters.

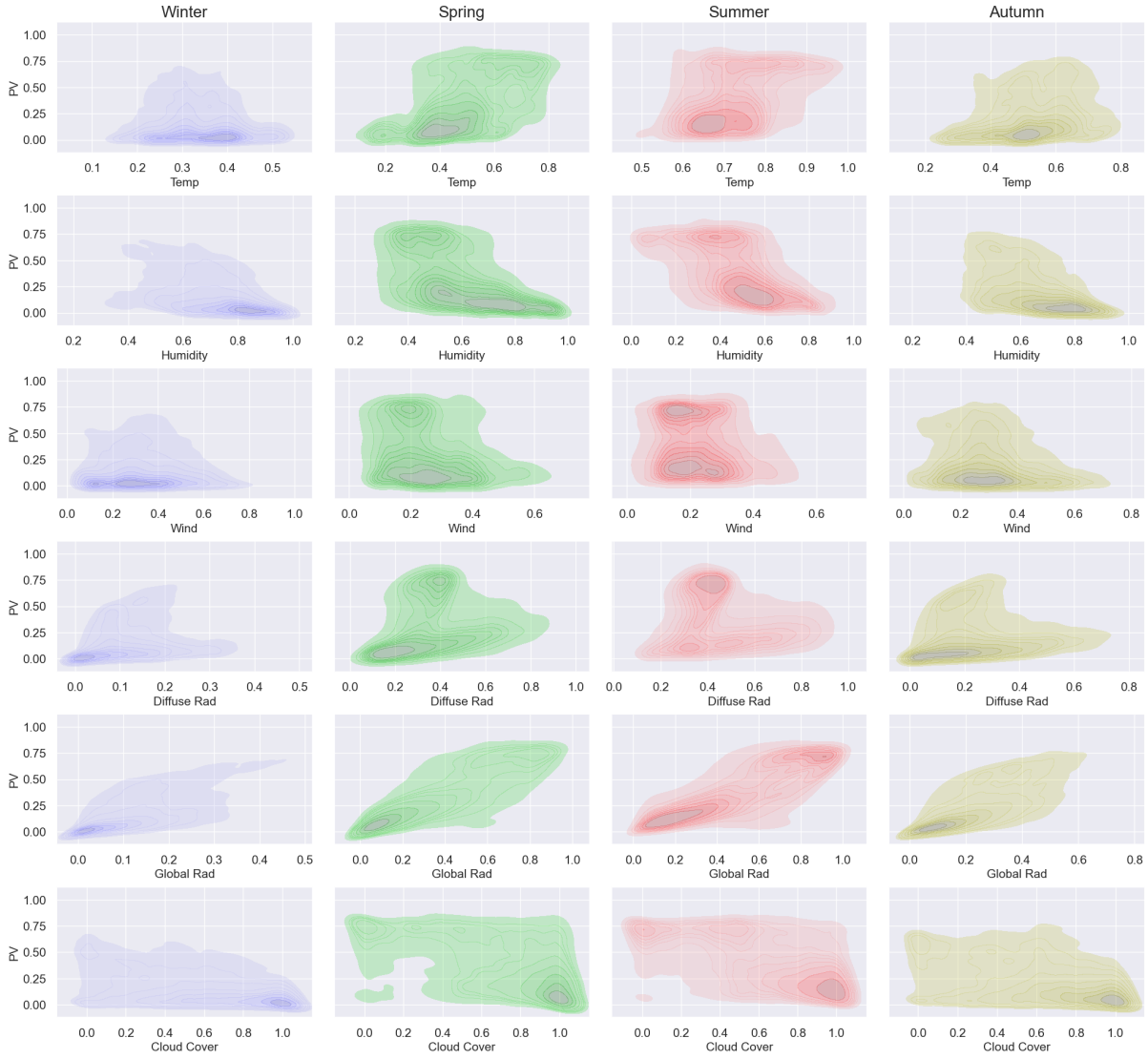


Figure 3.4: The distributions of weather variables with respect to PV over different seasons. We see how the summer and spring has a significantly bigger spread in weather conditions and corresponding PV values, which can result in making it harder to infer the corresponding PV value from the weather parameters.

3.2 Multitask Gaussian Processes

Multitask Gaussian Processes (MT-GPs) represent a multi-task learning paradigm within the domain of Gaussian Processes [35]. They extend the GP methodology by leveraging the natural correlations and shared information across diverse tasks. This approach holds particular relevance to model PV output where systems are close in the euclidean sense. The spatial proximity of such stations often implies shared underlying weather conditions that influence PV power generation. However, it is important to note that in instances when tasks are unrelated it can be detrimental to learn them together. This has been highlighted in [37] and empirically supported in [38]. Therefore, we investigate the MT-GP model in a small region with a high density of PV stations with the aim of satisfying relatedness between the PV between the stations considered.

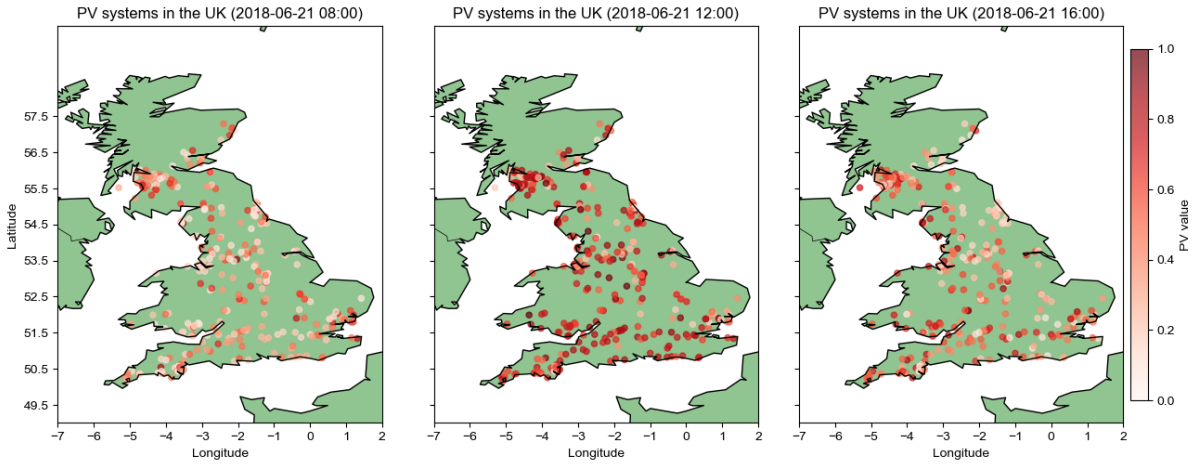


Figure 3.5: The plot displays scattered PV systems across the UK, captured at four-hour intervals on June 21, 2018. Each circle represents a PV system, with the intensity of red color indicating the level of PV power generated by that system. Darker shades of red represents higher PV output. The observed spatial pattern reveals a correlation in PV output with localized variations. These variations can likely be attributed to small scale factors, such as passing clouds.

In [4], the authors demonstrate the correlation across predictions for PV systems (Figure 3.6). We can see from Figure 3.6 that the correlation deprecates quickly as systems are separated from each other. This highlights the importance of a close proximity between PV systems, as the inter-task correlation will be the foundation of our framework.

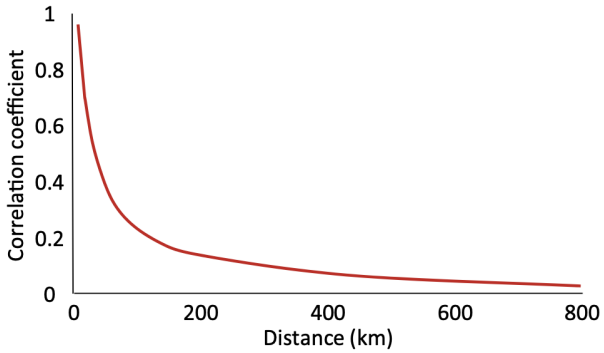


Figure 3.6: The correlation coefficient of forecast errors of PV systems as measured by the distance between them. Plot from [4].

3.2.1 Linear Coregionalization Model

The Linear Coregionalization Model (LCM) is a foundational framework within MT-GPs [35, 39]. The LCM MT-GP jointly model multiple related tasks by capturing an underlying latent structure. The GP prior is extended to utilize shared latent GPs, which are linearly mixed to express the M related tasks.

First, we define $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_1(\mathbf{x}), \ldots, f_Q(\mathbf{x})]^T$ to be a vector of Q latent, independent GPs. Next, we define

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_M \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1Q} \\ a_{21} & a_{22} & \cdots & a_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{T2} & \cdots & a_{MQ} \end{bmatrix}$$

to be a matrix with learnable entries where each row $\mathbf{a}_m \in \mathbb{R}^Q$ corresponds to a set of

coefficients associated with a task specific prior g_m for $m = 1, \ldots, M$ (see Equation 3.4). The latent function of our LCM MT-GP is expressed by

$$\mathbf{g}_i = \mathbf{A}\mathbf{f}(\mathbf{x}_i) = [g_{1,i}, g_{2,i}, \ldots, g_{M,i}]^T, \tag{3.3}$$

where m$^{\text{th}}$ entry of $\mathbf{g}_i$ is a linear mixture of m$^{\text{th}}$ row in $\mathbf{A}$ and our Q latent functions in $\mathbf{f}$ evaluated at $\mathbf{x}_i$, which is given by:

$$g_{m,i} = \sum_{q=1}^{Q} a_{mq} f_q(\mathbf{x_i}) + \epsilon_i, \tag{3.4}$$

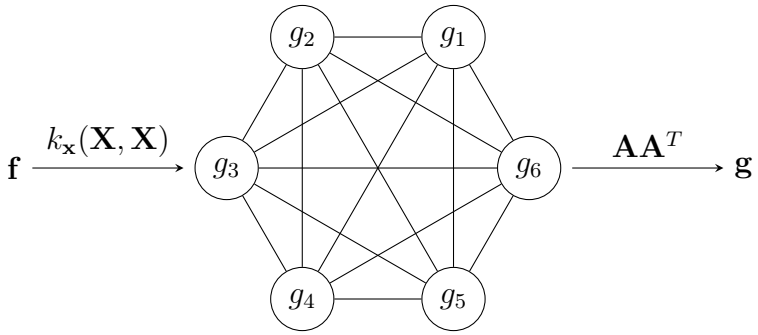
where $\epsilon_i \sim \mathcal{N}(\epsilon_i \mid 0, \sigma_\epsilon^2)$. Given that $\mathbf{g} = \mathbf{A}\mathbf{f}(\cdot)$ is an affine transformation, we know it preserves Gaussianity in $\mathbf{g}$. We assume independence between the GPs in $\mathbf{f}$ where each function f_q has an equivalent kernel function $k_\mathbf{x}$ on any input points $\mathbf{x}$ and $\mathbf{x}'$. That leads to $\text{Cov}(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x}')) = k_\mathbf{x}(\mathbf{x}, \mathbf{x}')I$ where $I \in \mathbb{R}^{Q \times Q}$ is the identity matrix. Then, assuming zero mean, we have

$$\mathbf{f} \sim \mathcal{GP}(\mathbf{0}, k_\mathbf{x} I). \tag{3.5}$$

Consequently, that leads to the following covariance for the latent function between given two input points $\mathbf{x}$ and $\mathbf{x}'$:

$$\begin{aligned} \text{Cov}(\mathbf{g}(\mathbf{x}), \mathbf{g}(\mathbf{x}')) &= \mathbb{E}\left[\mathbf{A}\mathbf{f}(\mathbf{x})(\mathbf{A}\mathbf{x})^T)\right] + \sigma_\epsilon^2 I \\ &= \mathbf{A}\mathbb{E}\left[\mathbf{f}(\mathbf{x})\mathbf{f}(\mathbf{x}')^T\right]\mathbf{A}^T + \sigma_\epsilon^2 I \\ &= k_\mathbf{x}(\mathbf{x}, \mathbf{x}')\mathbf{A}\mathbf{A}^T + \sigma_\epsilon^2 I, \end{aligned}$$

Intuitively, we can visualize the inter-task dependencies through a graph structure between the latent GP priors and the tasks. Assuming 6 tasks our graph can be represented as:



First, the latent GPs supply the input related covariance matrix evaluated at $\mathbf{X}$. Then, the edge between g_i and g_j specifies the inter-task correlation given by the $\mathbf{A}\mathbf{A}^T$ matrix. Specifically, assume we have shared input across M tasks given by $\mathbf{X} \in \mathbb{R}^{N \times D}$. If $k_{\mathbf{x}}(\mathbf{X}, \mathbf{X})$ is the kernel on the inputs and $A(i, j)$ is an output covariance function from the inter-task correlations. Then the covariance for a Kroenecker structured GP is given by $k_{xx} = k_{\mathbf{x}}(\mathbf{X}, \mathbf{X}) \otimes A(\cdot, \cdot) \in \mathbb{R}^{NM \times NM}$, where $\otimes$ denotes the Kroenecker product from Section 2.3.4. k_{xx} can thereby be seen as a block matrix whose blocks are $M \times M$ matrices of task correlations. Then, assuming a zero mean, the prior distribution of $\mathbf{g}$ is given by:

$$p\left(\mathbf{g}(\cdot)\right) = \mathcal{GP}\left(\mathbf{0}, k_{\mathbf{x}}(\cdot, \cdot) \otimes \mathbf{A}\mathbf{A}^T + \sigma_\epsilon^2 I\right) \tag{3.6}$$

As highlighted in [35] the joint Gaussian distribution over $\mathbf{g}$ is not block-diagonal with respect to tasks. So, the observations of one task can affect the predictions on another. We observe that if the latent functions capture shared patterns across tasks, the predictions of one task can indirectly affect another task's predictions through this latent information coupling, illustrating the interconnection of tasks in MT-GPs.

A note is that the order which we stack the vector $\mathbf{g}$ matters. From above, we have $\text{Cov}(\mathbf{g}) = k_{\mathbf{x}}(\mathbf{X}, \mathbf{X}) \otimes \mathbf{A}\mathbf{A}^T + \sigma_\epsilon^2 I$. In contrast to, $\text{Cov}(\mathbf{g}^T) = \mathbf{A}\mathbf{A}^T \otimes k_{\mathbf{x}}(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 I$. Ultimately, they are two different ways of representing the prior covariance function, and $\mathbf{A}\mathbf{A}^T$ and $k_{\mathbf{x}}(\mathbf{X}, \mathbf{X})$ still operates on the output of and input space, respectively.

We will consider two cases of LCM MT-GPs to support the aims of this project:

1. A temporal MT-GP that leverages shared input exclusively from the time domain, featuring a Kronecker-structured output covariance.

2. Extending the temporal MT-GP to incorporate exogenous regressors linked to each PV station, utilizing a Hadamard-structured output covariance.

3.2.2 Kroenecker Linear Coregionalization Model

When only considering the temporal information when modeling the PV output, we let $\mathbf{T} = \{t_i\}_{i=1}^{N}$ represent the distinct points in time and a set of corresponding response variables $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y_2}, \ldots, \mathbf{y}_M] \in \mathbb{R}^{N \times M}$ where $\mathbf{y}_m \in \mathbb{R}^N$ is the response of the m$^{\text{th}}$ PV station. Given a set of past values $\mathbf{y}_{\text{observed}}$ we wish to make short-term predictions on future values $\mathbf{y}^*$ by capitalizing on the shared information across PV stations. We adapt the LCM model from Section 3.2.1 and the quasi-periodic kernel from Section 2.3.3. The resulting covariance function then becomes:

$$\mathbb{E}[\mathbf{g}(\mathbf{T})\mathbf{g}(\mathbf{T})^T] = k_{\text{QP}}(\mathbf{T}, \mathbf{T}) \otimes \mathbf{A}\mathbf{A}^T + \sigma_\epsilon^2 I \tag{3.7}$$

where $\otimes$ denotes the Kronecker product , k_{QP} is the quasi-periodic covariance function over our GP priors, and $\mathbf{g} = [g_1, \ldots, g_M]^T$ is the vector of latent functions for each respective PV station. Figure 3.7 demonstrates how the covariance function is constructed when trained on 5 PV stations using 4 latent GPs $\mathbf{f} = \{f_q\}_{q=1}^{Q}$.

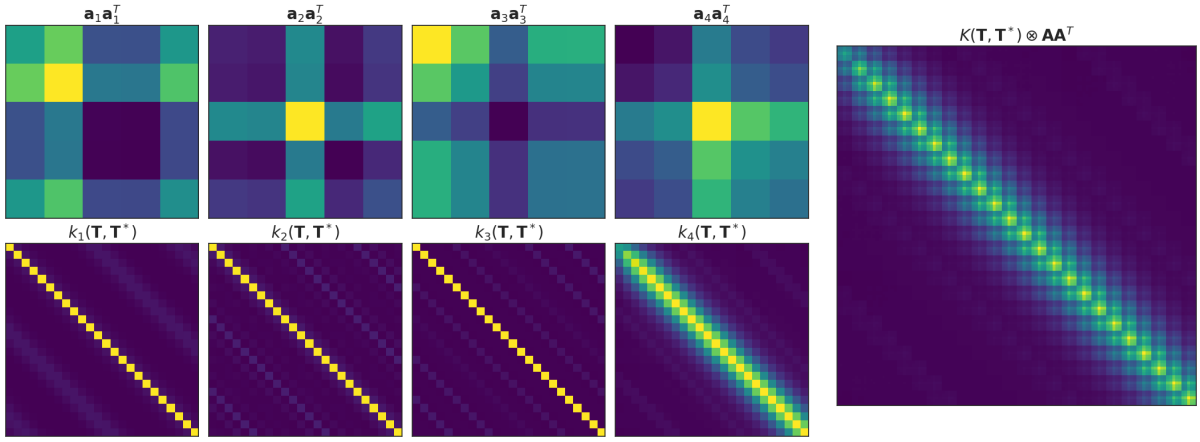


Figure 3.7: The covariances and inter-task factors for a Kroenecker structured LCM MT-GP with 5 PV stations and 4 latent GPs. The plots on the top-left row is the outer product of each column vector $\mathbf{a}_q$ for $q = 1, \ldots, 4$ such that $\sum_{q=1}^{4} \mathbf{a}_q\mathbf{a}_q^T = \mathbf{A}\mathbf{A}^T$. The bottom-left row shows the input-related covariance matrix for each latent GP f_q. The right plot then shows the full covariance of the latent function given by $\text{Cov}(\mathbf{g}) = k_{\text{QP}}(\mathbf{T},\mathbf{T}^*) \otimes \mathbf{A}\mathbf{A}^T + \sigma_\epsilon^2 I$ where $\mathbf{T}$ is our training input and $\mathbf{T}^*$ is the test input.

3.2.3 Hadamard Linear Coregionalization Model

While MT-GPs are known for learning correlated tasks efficiently [35], they are limited to be in the same input domain. Therefore, they are unable to tackle heterogeneous, meaning inputs are no longer shared across tasks. When modeling several PV outputs in a local area we aim to effectively capture the diverse influences affecting these outputs by accommodating for local variations in weather conditions. To accommodate this, we will use a LCM MT-GP model with a Hadamard product for the output covariance function. This will differ from the temporal MT GP in one key way.

Suppose we have M PV stations with $M \geq 2$. The PV values of the m^{th} station, $\mathbf{y}_m$, has corresponding weather variables given by $\mathbf{X}_m$. We construct a joint input matrix of all the observed weather parameters for our stations denoted $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_M]^T \in \mathbb{R}^{NM \times D}$ where N and D is the number of observations and weather variables, respectively. The input is associated with our M stations PV values given by $\mathbf{y} = [\mathbf{y}_1, \ldots, \mathbf{y}_M] \in \mathbb{R}^{NM}$.

In this setting, we have the PV output $\mathbf{y}_m$, with corresponding weather parameters $\mathbf{X}_m$ at the each station m. Then for any station $m \neq m'$ it follows that $\{\mathbf{X}_m, \mathbf{y}_m\} \neq \{\mathbf{X}_{m'}, \mathbf{y}_{m'}\}$. Specifically, the weather parameters at station m, $\mathbf{X}_m$, is only associated to the PV output at the same station $\mathbf{y}_m$. Contrary to the Kroenecker LCM MT-GP where inputs are shared across the PV stations, we now have input specific to each PV station.

We highlighted in Section 3.1 that the temporal component is the only precise input for the 5-minute intervals. Therefore, we add the time domain as inputs in addition to our weather parameters. Therefore, we include the temporal input $\mathbf{T} = \{t_i\}_{i=1}^N$. Then for each task $m = 1, \ldots, M$, we now have input given by $\mathbf{Z}_m = (\mathbf{X}_m, \mathbf{T}) \in \mathbb{R}^{N \times D+1}$, which combines the weather parameters and the temporal input.

To highlight the difference between the Kroenecker LCM MT-GP, we can look at the change in function mapping. Assuming $\mathcal{X}$ and $\mathcal{Y}$ denotes the input and output domain, respectively. Then for the Kroenecker LCM MT-GP, we learn a mapping

$$f : \mathcal{X} \to (\mathcal{Y}_1 \times \mathcal{Y}_2 \times \cdots \times \mathcal{Y}_M)\,.$$

In contrast, the Hadamard setting extends the Kroenecker LCM MT-GP methodology to learn a mapping

$$f : (\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_M) \to (\mathcal{Y}_1 \times \mathcal{Y}_2 \times \cdots \times \mathcal{Y}_M)\,,$$

More precisely, there is only one task $\mathbf{y}_m$ associated to each input $\mathbf{Z}_m$. Then, for a single point the mapping is given by $f : \mathbb{R}^{D+1} \to \mathbb{R}$.

The covariance between two tasks i and j provided inputs $\mathbf{z}_t = [\mathbf{x}_t, t]$ and $\mathbf{z}_{t'} = [\mathbf{x}_{t'}, t']$

are now given by:

$$K_{\mathbf{z}}\left((\mathbf{z}, i), (\mathbf{z}', j)\right) = \mathbb{E}\left[\mathbf{f}(\mathbf{z})\mathbf{f}(\mathbf{z}')^T\right] \odot A^f_{i,j} + \sigma^2_\epsilon I \tag{3.8}$$

$$= k_{\mathbf{z}}(\mathbf{z}, \mathbf{z}') \odot A^f_{i,j} + \sigma^2_\epsilon \tag{3.9}$$

where $(\mathbf{z}, \mathbf{z}') \in \mathbb{R}$ denotes the covariance function at the inputs $\mathbf{z}_t$ and $\mathbf{z}_{t'}$. A^f is a kernel containing the inter-task correlations from the GP priors $\{f_q\}_{q=1}^Q$ in the LCM from Equation 3.4 such that $A^f_{i,j} \in \mathbb{R}$ contains the inter-task covariance between task i and task j. Lastly, $\odot$ denotes the Hadamard product from Section 2.3.5

Consequently, the resulting covariance function $\text{Cov}(\mathbf{g}) = k_{\mathbf{z}}(\mathbf{z}, \mathbf{z}') \odot A^f \in \mathbb{R}^{NT \times D+1}$ base itself on the some foundation as the Kroenecker LCM MT-GP with $k_{\mathbf{z}}(\mathbf{z}, \mathbf{z}')$ denoting our input covariance and A^f specifying our output covariance given by the inter-task correlations.

As in the Kroenecker LCM-MT GP, we adopt a quasi-periodic kernel for the temporal domain. For modeling weather parameters, we use a Squared Exponential (SE) kernel to capture similarities in observed weather conditions. By taking the product of these two kernels, our goal is to harness the strengths of both: the quasi-periodic kernel captures periodicity and deviations in PV output, while the SE kernel captures similarities in weather conditions. The resulting composite kernel is defined as

$$k_{\mathbf{z}}(\mathbf{z}, \mathbf{z}') = k_{\text{QP}}(\mathbf{t}, \mathbf{t}') \cdot k_{\text{SE}}(\mathbf{x}, \mathbf{x}'). \tag{3.10}$$

Figure 3.8 displays the resulting Hadamard covariance with exogenous regressors with 6 tasks and 4 latent functions.

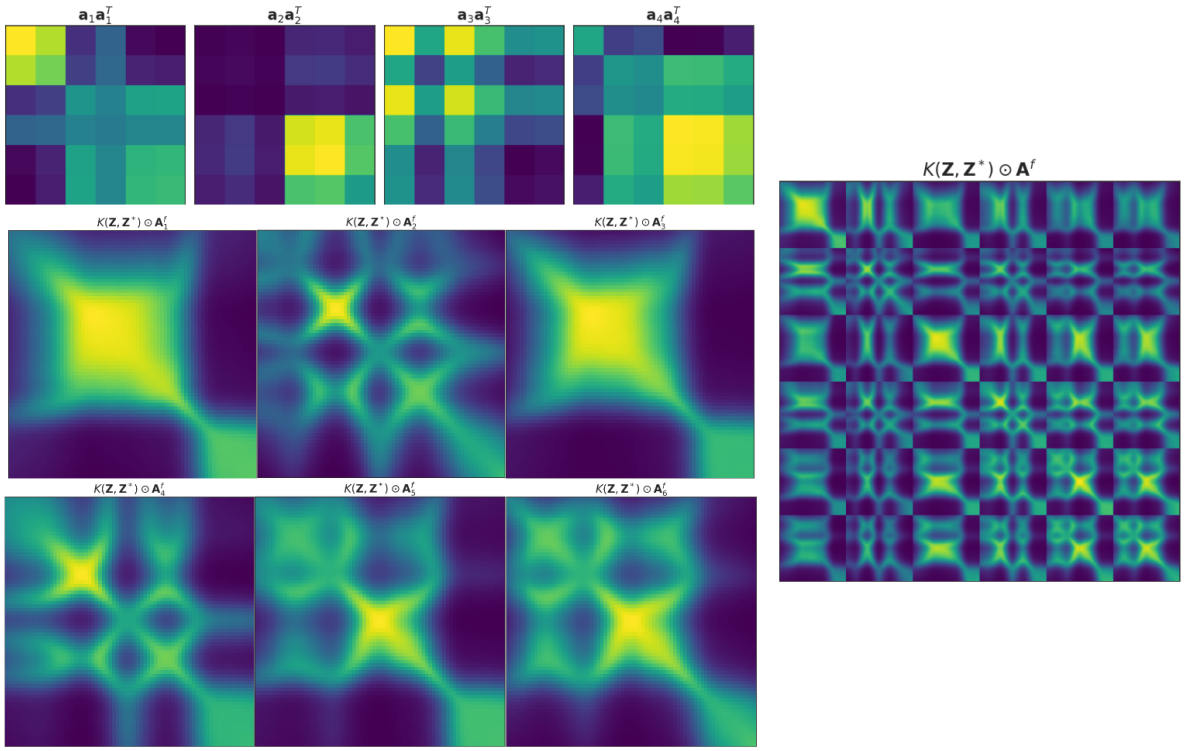


Figure 3.8: The covariances and inter-task factors for a Hadamard structured LCM MT-GP with 6 tasks and 4 latent GPs. The top-left row shows the outer product of the column vectors $\mathbf{a}_q\mathbf{a}_q^T$ for $q = 1, \ldots, 4$. The two bottom-left row shows the covariance for each latent task $\mathbf{g}_t$ for $t = 1, \ldots, 6$. The right plot shows the covariance over all tasks given by $k_{\mathbf{z}}(\mathbf{Z}, \mathbf{Z}^*) \odot A^f(T, T) + \sigma_\epsilon^2 I$ where T denotes the set of all tasks, $\mathbf{Z}$ is our training data, and $\mathbf{Z}^*$ denotes the test inputs.

3.2.4 Beta Likelihood for Bounded Predictions

As mentioned in Section 3.1, the response variable $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_M] \in \mathbb{R}^{N \times M}$ is constrained within the interval $[0, 1]$. The Beta distribution is well-suited for modeling our outputs due to its continuous probability distribution over the same interval. To parameterize the Beta distribution, we define a strictly positive vector $\boldsymbol{\nu} = [\nu_1, \cdots, \nu_M] \in \mathbb{R}^M$, which acts as an (inverse) dispersion parameter for each task and a mean parameter $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M] \in \mathbb{R}^{N \times M}$. For a task $\mathbf{y}_m$ with latent function $\mathbf{g}_m$ its corresponding mean, $\boldsymbol{\mu}_m$, is approximated using the link function $\Phi(\mathbf{g}_m)$, which is an inverse cdf of a

standard Gaussian function, where

$$\boldsymbol{\mu}_m = \Phi^{-1}(\mathbf{g}_m) = \int_{-\infty}^{\mathbf{g}_m} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx. \tag{3.11}$$

Under the independence assumption from Equation 2.4, the factorized likelihood over all tasks is defined by:

$$\boldsymbol{\alpha} = \boldsymbol{\mu}\boldsymbol{\nu}, \ \boldsymbol{\beta} = \boldsymbol{\nu} - \boldsymbol{\alpha} \tag{3.12}$$

$$p(\mathbf{Y} \mid \boldsymbol{g}, \boldsymbol{\nu}) = \text{Beta}(\mathbf{Y} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{i=1}^{N} \text{Beta}(\mathbf{y_i} \mid \boldsymbol{\alpha_i}, \boldsymbol{\beta_i}). \tag{3.13}$$

Here, $\boldsymbol{\mu}\boldsymbol{\nu}$ denotes the row-wise product product such that $\alpha_{i,m} = \nu_m \cdot \Phi^{-1}(g_{i,m})$. Similarly, $\boldsymbol{\beta}$ is the row-wise subtraction of $\boldsymbol{\alpha}$ from $\boldsymbol{\nu} \in \mathbb{R}^{1\times M}$. For example, for a single task we have $\beta_{i,m} = \nu_m - \alpha_{i,m}$ for $i = 1, \ldots, N$.

Together, the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ define the Beta distribution used to model the responses for each task. For instance, a single task $\mathbf{y}_m$, will have its likelihood defined by:

$$p(\mathbf{y}_m) = \prod_{i=1}^{N} \text{Beta}(y_{i,m} \mid \alpha_{i,m}, \beta_{i,m}) \tag{3.14}$$

The dispersion parameter ν governs the distribution's level of uncertainty. Larger ν enhances fitting flexibility but reduces variance, while smaller values decrease flexibility but raise variance. Consequently, ν shapes the expected variability in the output. When forecasting PV, it is therefore important that the dispersion adapts to the frequency of PV fluctuations. High short-term fluctuations should have a low ν for heightened uncertainty In contrast, less fluctuating PV should anticipate a high ν to reflect the low variance. Refer to Figure 3.9 for a visual representation.

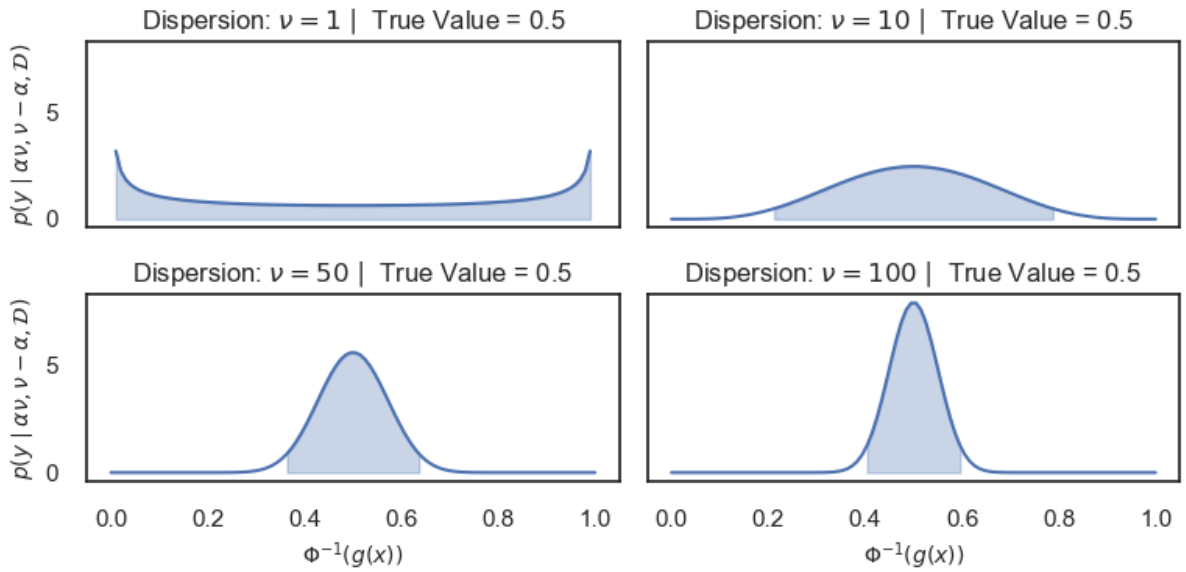


Figure 3.9: The 95% confidence interval for varying degrees of dispersion for a true value $y = 0.5$. We can see that the higher ν becomes, thus more peaked is the resulting beta likelihood. It is important to note that this is not completely representative of the resulting variance of our marginal likelihood, as we will integrate out the uncertainty in the latent function $g(x)$, which can affect the uncertainty quantification.

3.2.5 Variational Inference

We now formulate how we estimate the parameters of the LCM MT-GP models. When using a beta likelihood, both the marginal likelihood from Equation 2.6 and the posterior from Equation 2.5 becomes intractable. To address this challenge, we employ variational inference by utilizing the Evidence Lower Bound (ELBO).

3.2.5.1 Temporal Inference

In the temporal setting, we define the temporal input $\mathbf{T} \in \mathbb{R}^N$ for N observations corresponding to our target PV values $\mathbf{Y} \in \mathbb{R}^{N \times M}$ having M tasks. As seen above, the Kroenecker structured LCM MT-GP's covariance is given by $k_{\mathbf{t}}(\mathbf{T}, \mathbf{T}) \otimes \mathbf{A}\mathbf{A}^T + \sigma_\epsilon^2 I \in \mathbb{R}^{NM \times NM}$. Consequently, its inversion for the posterior covariance scales to $\mathcal{O}(M^3 N^3)$.

Luckily, we can take advantage of the Kroenecker inversion property:

$$\left(k_{\mathbf{t}}(\mathbf{T},\mathbf{T}) \otimes \mathbf{A}\mathbf{A}^T\right)^{-1} = k_{\mathbf{t}}(\mathbf{T},\mathbf{T})^{-1} \otimes \left(\mathbf{A}\mathbf{A}^T\right)^{-1} \tag{3.15}$$

which then scales to $\mathcal{O}(N^3 + M^3)$ ($\approx \mathcal{O}(N^3)$ as the number of PV stations is significantly smaller than the number of observations). As we will consider weekly folds during training (see Section 4.1), this allows us to use all our observations.

To approximate the intractable posterior we define a distribution $q_\lambda(\mathbf{g}) = N(\mathbf{g} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \approx p(\mathbf{g} \mid \mathcal{D}, \theta)$ where $\lambda = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ denotes the set of the posterior mean and covariance to be optimized. We apply the Cholesky decomposition by introducing a lower triangular matrix $\mathbf{L}$ such that $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^T$. This simplifies the optimization process by enabling efficient gradient-based updates while ensuring the covariance matrix is PSD.

As highlighted in Section 2.2.2, we can factorize the likelihood under the assumption of independence. Under this assumption, the objective becomes (see Appendix C.1 for complete derivation)

$$\mathcal{L}_{\text{ELBO}} = \sum_{\mathbf{T},\mathbf{y}} \mathbb{E}_{q(\mathbf{g}(t))}[\log p(y_i \mid g_i)] - D_{\text{KL}}\left(q_\lambda(\mathbf{g}) \,||\, p(\mathbf{g})\right). \tag{3.16}$$

Here we sum over all our training inputs $\{t_i, y_i\}_{i=1}^N$, and $\mathbf{g}(t) = \mathbf{A}\mathbf{f}(t)$ is the vector of latent functions obtained from our GP prior $\mathbf{f}$ and the weight matrix $\mathbf{A}$. The expected log-likelihood term encourages the model to have latent functions that can explain the observed data well. The Kullback-Leibler (KL) divergence, denoted D_{KL}, is a measure of relative entropy from q to p. For any two distributions $p(x)$ and $q(x)$, the KL divergence between $p(x)$ and $q(x)$ is defined by:

$$D_{\text{KL}}(p \parallel q) = \int p(x) \log \frac{p(x)}{q(x)}\, dx \tag{3.17}$$

In our ELBO, it penalizes the divergence between the approximate posterior, $q_\lambda(\mathbf{g})$ and the prior $p(\mathbf{g})$. It encourages the approximate posterior to stay close to the prior distribution. This results in the approximation being regularized by the prior distribution while fitting the data.

3.2.5.2 Exogenous Inference

In this framework, scalability issues arise due to how we represent the data in the software implementation. We set our PV values to be a vector $\mathbf{y} \in \mathbb{R}^{NM}$ with corresponding covariates $\mathbf{Z} = (\mathbf{X}, \mathbf{T}) \in \mathbb{R}^{NM \times D+1}$. Then use a masked-based approach to map the task-related input to the corresponding output task. As a result, the inversion of our covariance matrix $k_{\mathbf{z}}(\mathbf{Z}, \mathbf{Z}) \odot A^f \in \mathbb{R}^{NT \times NT}$ scales to $\mathcal{O}(N^3 M^3)$.

To address this challenge, we employ a two-fold approach. First, we adopt a sub-sampling strategy, systematically selecting equispaced intervals of every sixth data point. This results in a modified dataset where every other data point is linearly interpolated, while the rest remain the exact NWPs. Subsequently, we leverage inducing points as introduced by Titsias et al. [29].

As before, we seek to optimize the parameters to make predictions using Equation 2.6. Because of the non-conjugate beta likelihood, we cannot derive an exact posterior of the latent function $\mathbf{g}$. So, we resort to the ELBO for the marginal likelihood *with inducing points*:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\lambda(\mathbf{g}(u))} \left[\log p \left(\mathbf{y} \mid \mathbf{g}, \boldsymbol{\theta} \right) \right] - D_{\text{KL}} \left(q_\lambda(\mathbf{u}) \mid\mid p(\mathbf{u}) \right) \tag{3.18}$$

where θ denotes the variational parameters of our posterior approximation and $\boldsymbol{\theta}$ denotes the additional parameters (in the likelihood, the kernel, etc). We use a set of $P \ll NM$ inducing variables $\mathbf{U} = \{u(\hat{z_m})\}_{m=1}^M \in \mathbb{R}^{M \times D+1}$ contained within $\hat{\mathbf{Z}} = [\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_M] \in \mathbb{R}^{M \times D+1}$. To keep the paper self-contained, we refer to Appendix C.2 for full details.

3.2.5.3 Training Scheme

When optimizing the Evidence Lower Bound (ELBO), our goal is to maximize a lower bound on the logarithm of the marginal probability of our observations, as defined in Equation 2.6. Initially, we considered Variational Inference (VI) using natural gradients as the assumptions of an Euclidean loss geometry can be a bad assumption [40]. However, we observed two challenges: slower training and instability in the optimization process. This can root from a highly non-convex optimization landscape attributed by the intricate parameterization of the LCM MT-GP combined with the non-conjugacy of the beta likelihood. Therefore, we transitioned to Stochastic Variational Inference (SVI) using ADAM [41] (Table 1). This proved to be more stable whilst upholding a <5-minute training mark[1] necessary for fast model re-calibration, a critical aspect in effectively integrating recent observations in PV nowcasting.

Algorithm 1 Stochastic Variational Inference (SVI) with Adam

Input: Dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, The set of model- and variational parameters $\phi = \{\boldsymbol{\theta}, \lambda\}$, Learning rate η, Adam hyperparameters α, β, Number of iterations N
Initialize $\mathbf{m}_\phi = \mathbf{0}$, $\mathbf{v}_\phi = 0$
for $n = 1$ to N **do**
    Sample $\mathbf{g}$ from the variational distribution $\mathbf{g} \sim q_\lambda(\mathbf{g}(\mathbf{X}))$
    Compute the gradient of the ELBO
    $\nabla_\phi \mathcal{L}_{\text{ELBO}} = \nabla_\phi \left[\mathbb{E}_{q_\lambda(\mathbf{g})}[\log p(\mathbf{y} \mid \mathbf{g}, \boldsymbol{\theta})] - D_{\text{KL}}(q_\lambda(\mathbf{g}) \mid\mid p(\mathbf{g}))\right]$
    Update moving averages for gradients:
    $\mathbf{m}_\phi = \beta^n \mathbf{m}_\phi + (1 - \beta^n) \nabla_\phi \mathcal{L}_{\text{ELBO}}$
    $\mathbf{v}_\phi = \beta^n \mathbf{v}_\phi + (1 - \beta^n)(\nabla_\phi \mathcal{L}_{\text{ELBO}})^2$
    Correct bias in moving averages:
    $\hat{\mathbf{m}}_\phi = \frac{\mathbf{m}_\phi}{1-\beta^n}$
    $\hat{\mathbf{v}}_\phi = \frac{\mathbf{v}_\phi}{1-\beta^n}$
    Update parameters:
    $\phi = \phi + \eta \frac{\hat{\mathbf{m}}_\phi}{\sqrt{\hat{\mathbf{v}}_\phi} + \epsilon}$
end for
Output: Optimized parameters $\phi = \{\lambda, \boldsymbol{\theta}\}$

[1] Using a 2020 MacBook Pro with M1Chip and 8GB memory. Training was done using CPU due to no current support for Cholesky Decomposition on MPS in PyTorch.

3.3 Training Set-Up

Throughout our model evaluation and hyperparameter tuning phases, we employed weekly folds of PV data. This inherently assumes that the trend from the past week provides sufficient information for making predictions. As discussed in Section 1.5, PV output is predominantly influenced by immediate weather conditions. Thereby, we assume that the observations over the past week is adequate for nowcasting. We will discuss the limitations of this approach in Chapter 5.

Before splitting our data into training and testing sets, we normalized the weather parameters within each fold. Subsequently, we randomly selected test data to initiate between 08:00 and 14:00, as depicted in Figure 3.10. This random selection introduces diversity into the test data, enabling us to assess the model's generalization capabilities.

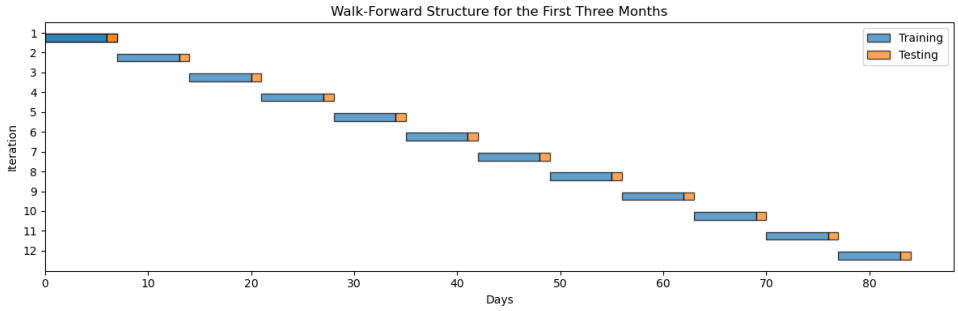


Figure 3.10: Experimental set up. The plot shows the walk-forward procedure for three months.

3.4 Seasonal Considerations

Given that solar energy generation is subject to the sun's seasonal variations, we investigate how the seasonal shifts impact local variations of PV systems. Because diverse weather conditions, such as rapid weather pattern changes experienced by PV stations, it can challenge the validity of the LCM MT-GP's assumption of a consistent shared trend.

To comprehensively assess the model's practical applicability, we conduct performance evaluations across seasons to gauge how seasonal changes can influence the relative performance of the LCM MT-GP compared to our benchmarks. As depicted in Figure 3.11, we see how the PV energy generation evolves with changing seasons and that the test data fluctuates more rapidly for the summer and spring months relative to the winter and autumn.

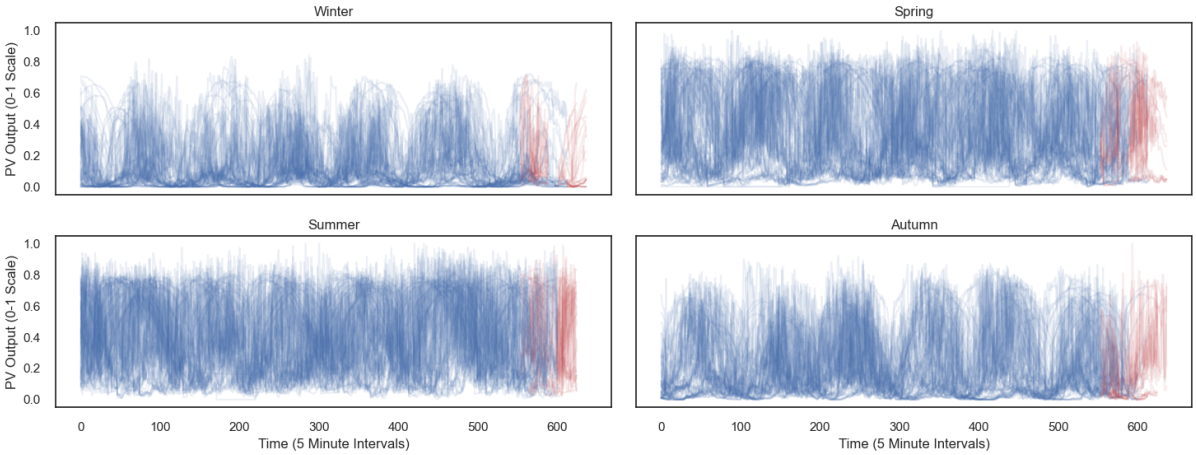


Figure 3.11: The PV train-and test data for a sample of folds for different seasons. The blue lines indicate training data and the red lines indicate the test data. We can see that the periodic trends are more established for the winter and autumn, but fluctuates more during the spring and summer.

3.5 Warm Starting

To initialize the model parameters to reasonable values, we used information from the previous week's fold. Specifically, we initialize the approximate posterior as follows:

$$q_\lambda^{(i+1)}(\mathbf{g}) = q_\lambda^{(i)}(\mathbf{g}) = \mathcal{N}(\mathbf{g} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{3.19}$$

where q_λ is the approximate posterior from Section 3.2.5. A note here is that this only applies when the previous week fold was split at the same hour as the subsequent fold.

Otherwise, there is a mismatch in the number of training points for the variational distribution. Further, we set the model parameters $\boldsymbol{\theta}^{(i+1)}$ (which encompass kernel parameters, mixture coefficients from the LCM MT-GP, etc.) to be the same as $\boldsymbol{\theta}^{(i)}$ at the start of each fold for $i = 2, \ldots, 51$, where i indicates the week. Consequently, we leverage the past observed fold in case the preceding fold exhibit a similar periodic nature. The warm-start approach combined with early-stopping at convergence contributed to reducing the training time by approximately twofold without affecting performance.

3.6 Model Set-up

In this section, we describe the process of initializing the hyperparameters for the LCM MT-GP models. To perform the hyperparameter selection, we train and validate on a subset of 6 stations in the Manchester region centered at (N53.28°, W3.05°) with a radius of 20km. Our approach involves two key steps.

First, we explored the imposition of priors and the initialization of quasi-periodic kernel parameters. Surprisingly, we observed negligible impacts on kernel parameter convergence from these initializations.

Next, we selected the number of latent functions, derived from $\mathbf{g}$ (as defined in Section 3.2), and the initial dispersion parameter $\boldsymbol{\nu}$ (as introduced in Section 3.2.4), through Bayesian Optimization (BO) using Optuna [42]. This involved determining the initial dispersion of the beta likelihood and deciding the number of latent functions for the LCM MT-GP. For computational efficiency, we constrained the number of tasks to six, aligning with a 5-minute training window for the Hadamard kernel structure. A similar constraint was adapted for the Kroenecker structure to compare the effect of exogenous regressors.

During the BO, we faced a trade-off between a low Mean Absolute Error (MAE) and minimizing the Negative Log Predictive Density (NLPD) (see Section 4.1.1 for defini-

tions). Higher dispersion led to better MAE but reduced uncertainty in the predictive density. To address this trade-off, we assigned $\frac{1}{N}p(y_i \mid \mathcal{D}, \theta) = 0.1$ at points where the predictive likelihood assigned zero probability to test values, which through empirical observations corresponded to a poor NLPD. This, effectively combined with MAE, allowed us to formulate a unified objective function given by

$$\mathcal{L}(f) = \frac{1}{T}\sum_{t=1}^{T}[\mid y_t^* - \hat{y}_t^* \mid -\log p(y_t^* \mid \mathcal{D}, \theta)] \tag{3.20}$$

where we have T test points, observed data $\mathcal{D}$, and model parameters θ. Further, y_t^* and $\hat{y}_t^*$ denotes the true and predicted PV value at time t. This balanced both objectives. Without the combination, the BO would prefer a small dispersion, resulting in overly wide uncertainty regions, as depicted in Figure 3.9. Such outcomes offer little practical insight. By incorporating both MAE and NLPD, we aimed to guide the optimization process towards a more balanced model that effectively balances prediction accuracy and uncertainty quantification.

The results of the hyperparameter sweep when only considering NLPD, illustrated in Figure 3.12, indicated that the LCM MT-GP performs optimally with an initial dispersion value of $\nu = 1$ in the beta likelihood (Section 3.2.4). After training, this led to large uncertainty boundaries at convergence (typically with $\nu \in \{3, 10\}$). When considering the trade-off, an optimal initial value for ν was found to be $\nu = 15$. Notably, the hyperparameter importance remained equivalent, placing a 94% importance on the dispersion $\boldsymbol{\nu}$ and 6% on the number of latent functions.

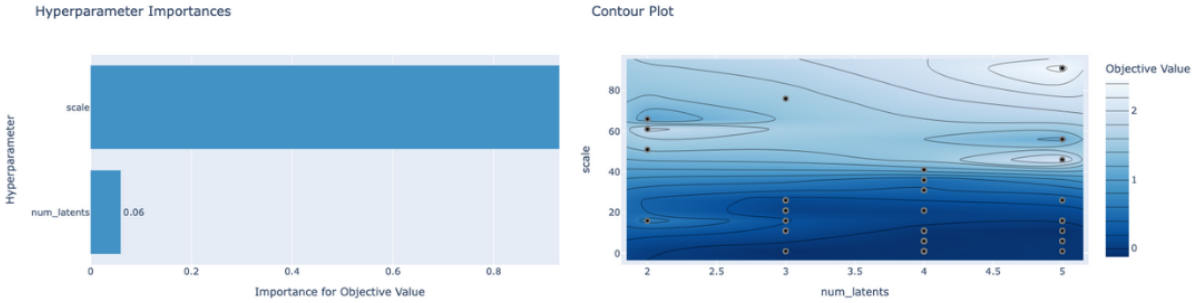


Figure 3.12: A plot of the hyperparameter sweep when only using NLPD as the objective. On the left we see the importance of the hyperparameters with the scale (referring to the dispersion parameter $\boldsymbol{\nu}$ in the beta likelihood from Section 3.2.4 and the number of latent functions in the LCM MT-GP. In this scenario it will prefer a selection of $\boldsymbol{\nu} = \mathbf{1}$, but we can observe a low objective value in the region of $\boldsymbol{\nu} \in \{\mathbf{1}, \mathbf{20}\}$. Note that $\boldsymbol{\nu}$ here refers to the vector containing the dispersion scale for each task.

Figure 3.13 illustrates how the resulting fit of a low initial dispersion scale affected the model's resulting fit of training data and predictions.

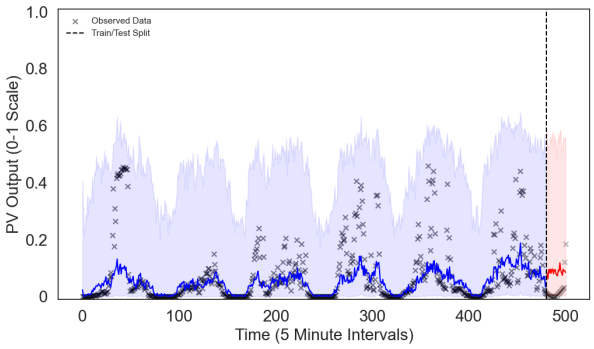


Figure 3.13: The resulted fit of a Kroenecker LCM MT-GP with initial dispersion scale $\boldsymbol{\nu} = \mathbf{1}$ in the beta likelihood. We can see that it will discard the noisy fluctuations with this initial scale.

3.7 Making Predictions - Monte Carlo Method

The resulting marginal likelihood, used for making predictions, is expressed by

$$p\left(\mathbf{Y} \mid \theta\right) = \int \text{Beta}\left(\mathbf{Y} \mid \boldsymbol{\alpha}(\mathbf{g}), \boldsymbol{\beta}(\mathbf{g})\right) q_{\lambda}(\mathbf{g}) d\mathbf{g} \tag{3.21}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$,are defined as in Section 3.2.4, $\mathbf{g}$ is the GP prior of the LCM-MT GP, and θ denotes the LCM MT-GP parameters. Due to non-conjugacy, this necessitates an approximation of the intractable integral. We apply Monte Carlo integration (MCI) to estimate the marginal likelihood. MCI relies on random sampling to numerically estimate integrals. By drawing samples $g_i \sim q_{\lambda}(g(x_i))$, we can evaluate the likelihood $p(\mathbf{y} \mid \mathbf{g}, \theta)$. By accumulating N samples $\{g_i\}_{i=1}^{N}$, the MCI approximation is defined by

$$\hat{\mathcal{I}} = \frac{1}{N} \sum_{i=1}^{N} p(\mathbf{y} \mid g_i, \theta). \tag{3.22}$$

With an increasing number of samples N, this approximation gradually converges towards the true marginal likelihood [43]. Thus a large sample of size N is likely to give a good indication of our true predictive distribution. Figure 3.14 demonstrates the MCI of the marginal likelihood for a Hadamard GP where we can see that the 95% confidence interval covers the distribution of our data points well.

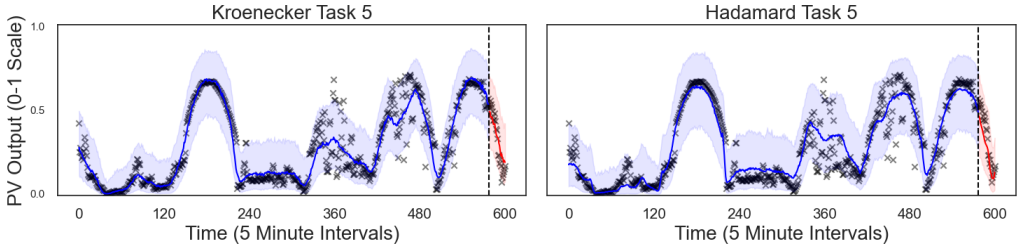


Figure 3.14: 2 hour predictions obtained from MCI on a sample task. The left plot show predictions obtained from the Kroenecker LCM MT-GP with temporal input. The right plot show predictions from the Hadamard LCM MT-GP that includes exogenous regressors. More examples can be found in Appendix B.

3.8 Benchmarks

To assess the performance of our model, we will consider three naive models and a wide range of both statistical and ML based models due to the concerns addressed by Makridakis et. al [3]. We will consider univariate statistical models whose implementations are conceptually simple with computational cost significantly lower than the GP model.

3.8.1 Univariate Benchmarks

3.8.1.1 Naive Persistence

The Naive Persistence model assumes that conditions remain the same for our forecast horizon. Specifically, it assumes the last observed will continue into the the future meaning no underlying trend, seasonality, or other patterns are present in the data. It is therefore recommended when the modeled time series is stationary, which is associated with a consistent weather pattern (e.g. clear sky PV production) [4]. The model is defined as

$$\hat{y}_{t+i} = y_t \tag{3.23}$$

where y_t is the last observed value and i denotes the forecasting step for $i = 1, \ldots, H$.

3.8.1.2 Yesterday

Yesterday is another naive benchmark assuming a strong daily trend. It assumes that yesterday's observed value will continue into the following day. It is given by

$$\hat{y}_{t+i} = y_{t+i-24h} \tag{3.24}$$

where $24h$ denotes a 24-hour lookback window.

3.8.1.3 Hourly Average

The hourly average is a simple forecast averaging the past hour's observed value. Thereby, it assumes future values will follow the averaged pattern of the last hour, and gives equal weight importance to all the observed values:

$$\hat{y}_{t+i} = \frac{1}{12} \sum_{j=1}^{12} y_{t+i-5j} \tag{3.25}$$

Since our data is sampled at 5-minute intervals it corresponds to 12 observed values per hour, and a step size of length $5i$. The model predicts uses the 12 points, which is recursively rolled forward followed when computing the mean. The resulting prediction at $t + i$ is then an averaged of the past 12 observed values.

3.8.1.4 Exponential Smoothing

Exponential Smoothing models are a class of statistical time series forecasting techniques that place higher importance to recent observations while gradually reducing the impact of less recent observations (in contrast to the hourly model). Consequently, they are particularly useful for generating short-term forecasts as they can capture trend and tackle noise coming from older observations.

We will adapt a Simple Exponential Smoothing (ES), which proved to be the best performing model in [6] for PV nowcasting. It is also known to be a reliable method when forecasting on a wide range of time series, especially when no clear trend/seasonality is present in the data [5]. The model is given by:

$$\hat{y}_{t+i} = \sum_{j=0}^{i-1} \alpha(1-\alpha) y_{i-j} + (1-\alpha)^T \ell_0, \tag{3.26}$$

where ℓ_0 is the first forecasted value and α is a smoothing parameter obtained from

minimizing the sum of squared errors (SSE):

$$\text{SSE} = \sum_{t=1}^{T} (y_t - \hat{y}_t)^2 \tag{3.27}$$

3.8.1.5 Seasonal Exponential Smoothing

The Seasonal Exponential Smoothing (SES) model is a seasonal extension of the ES [5]. It has two variations, an additive method and a multiplicative method. The additive method is preferred when the seasonal variations remain nearly constant, whereas the multiplicative method is preferred for seasonal variations. As we will consider weekly folds of our data (see Section 4.1), we assume roughly constant variations such that we adapt the additive method. The model consists of three smoothing components in addition to the forecast equation:

$$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \tag{3.28}$$

$$b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \tag{3.29}$$

$$s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m} \tag{3.30}$$

where ℓ_t is a level parameter, b_t is a trend parameter, and s_t is a seasonal parameter. m denotes a seasonal hyperparameter, which we set to be 96 corresponding to the number of daily points when we have 5 minute intervals between 08:00 and 16:00 for each day. The level is a weighted mean of the seasonally adjusted observation $(y_t - s_{t-m})$ and the non-seasonal forecast $(\ell_{t-1} + b_{t-1})$ at time t. The trend equation follows Holt's linear method [5]. The seasonality presents a weighted average of the current seasonal index, $(y_t - \ell_{t-1} - b_{t-1})$, and the seasonal index at the same season for the previous day denoted by the subscript $t - m$. The forecasts are then given by the following equation:

$$\hat{y}_{t+i} = \ell_t + i \cdot b_t + s_{t+i-m(k+1)} \tag{3.31}$$

where i is the forecasting step and $k = \lfloor \frac{i-1}{m} \rfloor$ ensures the estimates of the seasonal indices when forecasting come from the last day of the sample. The measurement equations described above form the state-space model, which we can update to generate additive errors that can be used to derive confidence intervals for the predictions [5]. We generate these errors by simulating the HW model and sampling the noise distributions. Consequently, we can estimate the uncertainty using the derived distribution generated by the noise from the simulations.

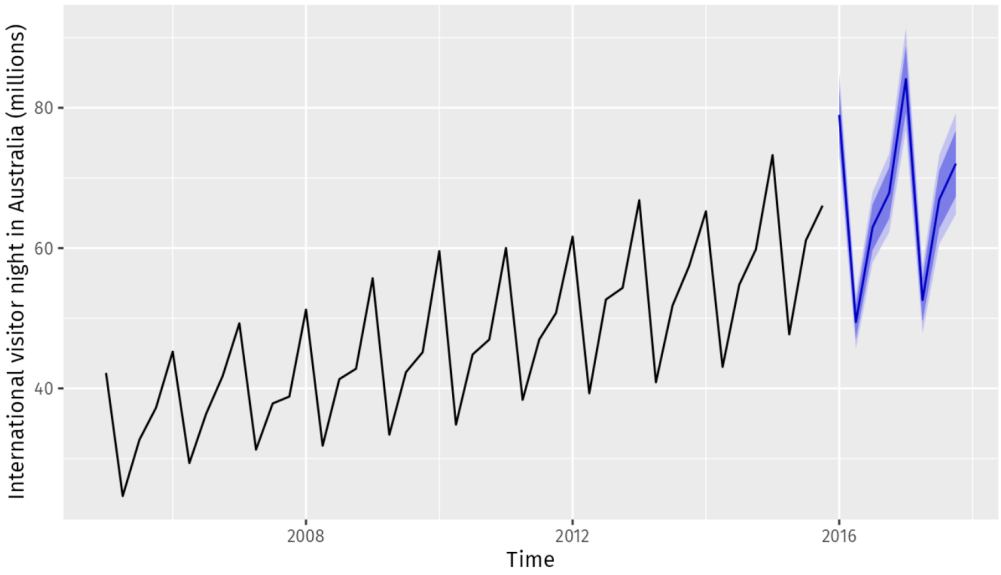


Figure 3.15: The HW model predicting international visitors in Australia with confidence intervals generated by the additive errors. The plot is taken from [5].

3.8.1.6 Vector Autoregression

The Vector Autoregression (VAR) model is the only benchmark model that considers inter-task correlations. In the VAR framework, all tasks are treated symmetrically meaning they influence each other uniformly. It generalizes the univariate autoregressive (AR) model by composing one equation per task using lags of all variables within the system. For notational simplicity, consider two tasks with one lag. The two-dimensional VAR is

then given by:

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} \phi_{1,1}^1 & \phi_{1,2}^1 \\ \phi_{2,1}^1 & \phi_{2,2}^1 \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \end{bmatrix} \tag{3.32}$$

where the coefficient $\phi_{i,i}^{\ell}$ captures the effect the ℓ-th lag from task y_i has on itself. Similarly, $\phi_{i,j}^{\ell}$ captures the effect of the ℓ-th lag from task y_j on task y_i. The parameters b_i and $\epsilon_i \sim \mathcal{N}(\epsilon_i \mid 0, \sigma_i^2)$ $(i = 1, 2)$ are the bias terms and the white noise, respectively. We can observe that the VAR replicates the idea of the LCM MT-GP of learning inter-task correlations but in a more direct manner.

3.8.2 Multivariate Benchmarks

3.8.2.1 XGBoost

XGBoost (XGB) for time series forecasting is an ensemble learning method that constructs a collection of decision trees during training and outputs a prediction by aggregating the trees [44]. It adds trees iteratively by correcting the errors of previous trees (boosting).

Decision trees splits the feature space by a subspace that minimize the variance of the target variable. The split with the largest reduction in variance is selected as the best split. If we let S be the set of samples at a node, and let y_i be the target variable for the i^{th} sample. The variance of y at node S is defined by

$$\mathbb{V}(y_S) = \frac{1}{|S|} \sum_{i \in S} (y_i - \bar{y}_S)^2 \tag{3.33}$$

where $\bar{y}_S$ is the mean of y at node S. When finding the best split, we consider all possible splits on the features and choose the one that yields the largest reduction in variance. We consider all possible split points s on feature j, and create the left and right subsets

of S as followed:

$$S_{\text{left}} = \{i \in S : x_{i,j} \leq s\}, \tag{3.34}$$

$$S_{\text{right}} = \{i \in S : x_{i,j} > s\} \tag{3.35}$$

where $x_{i,j}$ is the value for sample i at feature j. The resulting variance reduction is defined by

$$\Delta\mathbb{V}(y_S) = \mathbb{V}(y_S) - \frac{|S_{\text{left}}|}{|S|}\mathbb{V}(y_{S_{\text{left}}}) - \frac{|S_{\text{right}}|}{|S|}\mathbb{V}(y_{S_{\text{right}}}). \tag{3.36}$$

The split point s that maximizes $\Delta\mathbb{V}(y_S)$ is chosen as the best split for that node. XGB builds a series of trees $f_1, f_2, \ldots, f_T$, where each subsequent tree is trained to correct the errors of the previous tree. This is achieved by adding a new tree to the ensemble that adjusts the weights of the samples that with the larger error based on previous trees. To prevent overfitting, XGB uses a regularization technique that penalize complex models. This is achieved by adding a regularization term in the following objective function:

$$\mathcal{L}_{\text{XGB}}(\theta) = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{3.37}$$

where θ represents the set of parameters that XGBoost optimizes, l is the loss function, y_i is the true label of the ith sample, $\hat{y}_i^{(t-1)}$ is the predicted label of the i-th sample by the ensemble of trees up to round $t-1$, f_t is the t-th tree, x_i is the feature vector of the ith sample, and $\Omega(f_t)$ is the regularization term.

3.8.2.2 Long Short-Term Memory Network

A Long Short-Term Memory (LSTM) Network is a Recurrent Neural Network (RNN) designed to capture long-range dependencies and to tackle vanishing gradients. It consists of memory cells that can store information over long sequences, and a set of gates that

control the flow of information into and out of these cells.

The building block of an LSTM is a cell, which maintains a cell state having three gates: an input gate, a forget gate, and an output gate. The cell state is updated based on the three gates and the input at each time step. The input gate controls the information to be stored in the cell state, the forget gate determines what information should be discarded from the cell state, and the output gate is the output of the cell. The LSTM is defined by the following operations:

$$f_t = \sigma_g(W_f x_t + V_f h_{t-1} + b_f) \tag{3.38}$$

$$i_t = \sigma_g(W_i x_t + V_i h_{t-1} + b_i) \tag{3.39}$$

$$o_t = \sigma_g(W_o x_t + V_o h_{t-1} + b_o) \tag{3.40}$$

$$\tilde{c}_t = \sigma_c(W_c x_t + V_c h_{t-1} + b_c) \tag{3.41}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{3.42}$$

$$h_t = o_t \odot \sigma(C_t) \tag{3.43}$$

where x_t is the input at time step t, h_t is the hidden state at time step t, c_t is the cell state at time step t, and $\odot$ represent the Hadamard product. $W_k \in \mathbb{R}^{h \times D}$, $V_k \in \mathbb{R}^{h \times h}$, and b_k are weight matrices to be learned during training with the subscripts $k \in \{f, i, o\}$ representing the cell's input, input-to-state, and output-to-state transformations. h is the number of hidden units in the LSTM cell, D is the dimensionality of the input, and σ_g and σ_c are the activation functions for the gates and cell updates, respectively.

3.8.2.3 Bayesian Ridge Regression

Bayesian Ridge Regression (BRR) is a probabilistic approach to Linear Regression (LR). In contrast to LR, which provides point estimates for the model coefficients, BRR treats these coefficients as random variables and estimates their posterior distributions. The

linear relationship between the predictor variables and the target variable is given by:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{3.44}$$

where $\mathbf{y}$ is the vector of observed target values, $\mathbf{X}$ is the design matrix, $\boldsymbol{\beta}$ is the vector of coefficients, and $\boldsymbol{\epsilon} \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\epsilon} \mid \mathbf{0}, \sigma_\epsilon^2 I)$ are noise terms. We specify a Gaussian prior distributions over the regression coefficients $\boldsymbol{\beta}$:

$$\boldsymbol{\beta} \sim \mathcal{N}\left(\boldsymbol{\beta} \mid \mathbf{0}, \alpha^{-1} I\right) \tag{3.45}$$

where α is a regularization parameter, and I is the identity matrix. This prior assumes that each coefficient follows a Gaussian distribution with zero mean and precision α. Assuming a Gaussian likelihood due to the noise assumption and the prior $p(\beta)$, we have:

$$p(\mathbf{y} \mid \mathbf{X}, \beta) = \mathcal{N}\left(y \mid \mathbf{X}\beta, \sigma_\epsilon^2 I\right) \tag{3.46}$$

Using Gaussian identities and Bayes' theorem, we can derive a closed-form expression of the posterior using Equation 2.5:

$$p(\beta \mid \mathcal{D}) = \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \tag{3.47}$$

$$\boldsymbol{\Sigma} = \left(\alpha I + \frac{1}{\sigma_\epsilon^2}\mathbf{X}^T\mathbf{X}\right)^{-1} \tag{3.48}$$

$$\boldsymbol{\mu} = \frac{1}{\sigma_\epsilon^2}\boldsymbol{\Sigma}\mathbf{X}^T\mathbf{y} \tag{3.49}$$

The predictions can be made by considering the uncertainty in the coefficients. The predictive distribution (marginal likelihood) for a new data point $\mathbf{x}^*$ is given by:

$$p(\mathbf{y}^* \mid \mathbf{x}^*, \mathcal{D}) = \int \mathcal{N}\left(\mathbf{y}^* \mid \boldsymbol{\beta}^T\mathbf{x}^*, \sigma_\epsilon^2 I\right) \cdot \mathcal{N}\left(\beta \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) d\beta \tag{3.50}$$

$$= \mathcal{N}\left(\mathbf{y}^* \mid (\mathbf{x}^*)^T\boldsymbol{\mu}, (\mathbf{x}^*)^T\boldsymbol{\Sigma}\mathbf{x}^* + \sigma_\epsilon^2\right) \tag{3.51}$$

3.8.3 Simple Gaussian Process

To investigate the benefit of using inter-task correlations, we include a simple GP containing the same base kernel and likelihood as the MT-GPs. Thus, it provides a direct measure of whether there is any benefit of modeling the systems jointly. We adapt this to both the temporal and exogenous benchmarks.

The simple GP will share the same properties as the LCM MT-GP. That is, we adapt a quasi-periodic kernel and a beta likelihood, and train the model using SVI. Specifically, we impose a GP prior $f \sim \mathcal{GP}\left(\mathbf{0}, k_{\text{QP}}(\cdot,\cdot)\right)$ and the beta likelihood is parameterized as in 3.2.4. We have data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$ where $\mathbf{X} \in \mathbb{R}^{N\times D}$ and $\mathbf{y} \in \mathbb{R}^N$ Now, the mapping $f(\mathbf{x}_i) : \mathbb{R}^D \to \mathbb{R}$ gives us

$$\mu_i = \Phi^{-1}(f(\mathbf{x}_i)) \tag{3.52}$$

$$\alpha_i = \nu\mu_i \tag{3.53}$$

$$\beta_i = \nu - \alpha_i \tag{3.54}$$

where ν denotes the dispersion parameter, Φ is the inverse Gaussian link function, and α_i and β_i is the parameterization of the beta likelihood for the i$^{\text{th}}$ data point. As before, the likelihood over our entire dataset then reads

$$p\left(\mathbf{y} \mid \nu, \mathbf{f}\right) = \prod_{i=1}^{N} \text{Beta}\left(y_i \mid \alpha_i, \beta_i\right). \tag{3.55}$$

As before, train using SVI (outlined in section 3.2.5.3). Then, we apply MCI for predictions to approximate.

$$p\left(\mathbf{y} \mid \theta\right) = \int p\left(\mathbf{y} \mid \nu, \mathbf{f}\right) q_\lambda(\mathbf{f}) d\mathbf{f} \tag{3.56}$$

We will leverage this in both the temporal setting and when including weather parameters.

4 | Results and Analysis

4.1 Experiments

For computational efficiency, our experiments were constrained to a subset of PV stations. The aim is to assess the nowcasting performance of our MT-GPs while accounting for spatial correlations between PV stations, as detailed in Section 3.2. Therefore, we strategically chose PV stations located in North-East England, centered around the Newcastle region at coordinates (N55°, W1.5°), within a 25km radius. This region was selected due to its dense concentration of PV stations in the area. Figure 4.1 displays our station selection and the shared trend of the PV systems within the region.

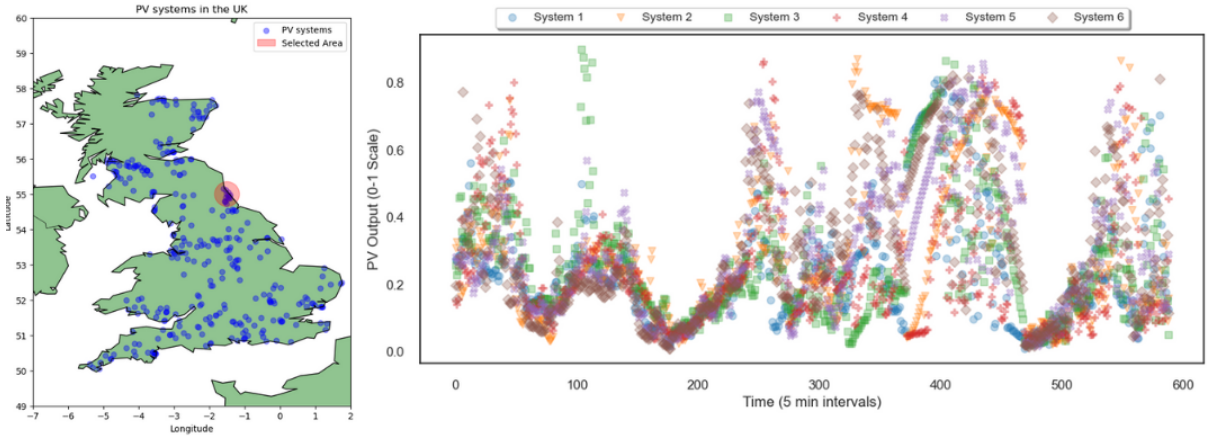


Figure 4.1: The left plot displays scattered PV systems across the UK, with the red circle marking our selected sample. The right plot showcases PV values for one week's train data from the six distinct systems. Noticeably, these PV values exhibit a shared underlying trend with localized variations unique to each PV system.

4.1.1 Evaluation Metrics

We conduct nowcasting for 2-hour horizons in the temporal setting and 2-hour and 6-hour horizons for the exogenous setting. To assess model performance, we employ two metrics: Mean Absolute Error (MAE) for point estimates and Negative Log Predictive Density (NLPD) for predictive uncertainty evaluation.

MAE offers a straightforward measure of prediction accuracy in the context of PV energy production. It accounts for variations in energy generation and is robust to outliers, making it well-suited for handling the inherent stochasticity of PV generation, such as the impact of passing clouds. MAE is defined by:

$$\text{MAE} = \frac{1}{T}\sum_{t=1}^{T} \mid y_t - \hat{y}_t \mid \tag{4.1}$$

where T represents the forecast horizon, y_t denotes the true value observed at time t, and $\hat{y}_t$ represents the predicted value.

NLPD was used to assess the predictive uncertainty of our probabilistic frameworks. It provides insights into how well the model captures uncertainty. In this context, NLPD allows us to compare the predictive distributions of GP models against Exponential Smoothing and BRR benchmarks, supporting the evaluation of our project's scope. It is given by:

$$\text{NLPD} = -\frac{1}{T}\sum_{t=1}^{T} \log p(y_t^* \mid \mathcal{D}, \boldsymbol{\theta}) \tag{4.2}$$

where T is the number of test points, $\mathcal{D}$ denotes the observed data, and $\boldsymbol{\theta}$ is the set of all model parameters.

4.2 Results

We will comprehensively assess the performance of Gaussian Processes (GPs) for PV nowcasting, utilizing the range of statistical and machine learning models detailed in Section 3.8. To ensure consistency, all machine learning models underwent fine-tuning via Bayesian Optimization in Optuna, focusing on the same subset of PV stations employed in tuning the LCM MT-GP model (see Section 3.6). This subset of data consists of weekly observations during the winter and spring seasons, where we encounter both noisy and periodic time series, and spans from 2018/01/02 to 2018/31/06.

Our evaluation of model performance considers weekly splits of PV data, as described in Section 3.3, resulting in 52 distinct folds that cover the entire year, from 2018/01/01 to 2018/31/12. To initialize the kernel parameters for the LCM MT-GP model, we ran the Simple GP benchmark on the same subset used in our hyperparameter tuning process, allowing us to observe the convergence of the kernel parameters, given that the LCM MT-GP models share the same base kernel.

We used the mean convergence of the GP kernel parameters observed to initialize our latent GP priors in the LCM MT-GP model. Additionally, we employed the findings from our hyperparameter tuning efforts in Section 3.6 to guide the application of four latent GP priors and an initial dispersion scale of $\nu = 15$ for each task. In all experimental settings we will include the Simple GP model from our benchmark. By doing so, we can assess the extent to which incorporating inter-task correlations enhances the GP framework. A note is that for most instances, this dispersion rate would decrease, which means the GP models will prioritize smooth trends with larger degrees of variance when the PV exhibits high fluctuations.

4.2.1 Temporal Results

In the temporal setting, we evaluate the performance of our Kroenecker LCM MT-GP model. The temporal benchmarks include Yesterday (Y.day), Persistence (Pers.), Hourly (Hr.), Vector Autoregression (VAR), Simple Exponential Smoothing (ES), Seasonal Exponential Smoothing (SES), and the Simple GP (GP). We calculate the MAE for all the statistical benchmarks, and the NLPD was adopted for the Exponential Smoothing benchmarks and the Simple GP.

Given our focus on two-hour forecasts at a 5-minute temporal resolution, our forecast horizon consists of a total of 24 steps. In Table 4.1, we provide the outcomes of our temporal experiments, showcasing how Kroenecker LCM MT-GPs compare to our benchmarks across different seasons in terms of mean MAE and NLPD.

Season	Metrics	Kr.GP	GP	SES	ES	Y	Pers.	Hr.	VAR
Spring	**MAE**	0.140	0.162	**0.121**	0.133	0.200	0.132	0.145	0.122
	Std Dev	0.126	0.141	**0.123**	0.140	0.181	0.181	0.143	0.140
	NLPD	-0.315	-0.285	**-0.417**	-0.370	N/A	N/A	N/A	N/A
	Std Dev	0.981	**0.821**	1.286	1.340	N/A	N/A	N/A	N/A
Summer	**MAE**	0.142	0.182	0.128	**0.119**	0.229	0.121	0.139	0.120
	Std Dev	0.111	0.132	0.120	0.127	0.212	0.128	0.126	**0.099**
	NLPD	**-0.335**	-0.235	-0.279	-0.308	N/A	N/A	N/A	N/A
	Std Dev	0.866	**0.509**	1.542	0.794	N/A	N/A	N/A	N/A
Autumn	**MAE**	0.105	0.132	0.097	0.098	0.150	0.150	0.099	**0.089**
	Std Dev	0.108	0.129	0.119	0.128	0.163	0.130	0.113	**0.092**
	NLPD	-0.621	**-0.728**	-0.485	-0.497	N/A	N/A	N/A	N/A
	Std Dev	0.784	1.146	1.131	**0.753**	N/A	N/A	N/A	N/A
Winter	**MAE**	0.075	0.085	0.080	0.081	0.124	0.081	0.089	**0.069**
	Std Dev	0.089	0.106	0.088	0.102	0.138	0.101	0.112	**0.084**
	NLPD	-0.867	**-0.988**	-0.716	-0.708	N/A	N/A	N/A	N/A
	Std Dev	1.265	1.835	**0.600**	0.695	N/A	N/A	N/A	N/A

Table 4.1: Results of the temporal experiments of 2 hour PV nowcasting. We abbreviate the Kroenecker LCM MT-GP as Kr.GP. The remaining abbreviations are provided above. We report the mean results and report their variance in the standard deviation (denoted Std. Dev.)

Consistent with the research on time series forecasting in [3] (discussed in Section 2.1 and

illustrated in 2.1), we observe that the statistical models outperform our GP models in terms of MAE. During the spring-and summer seasons, the SES and ES models achieve the lowest errors. For the autumn and spring seasons, our VAR benchmark displays the strongest performance. Notably, it is the only benchmark leveraging inter-task correlations similar to the LCM MT-GP. It also exhibits the lowest variability in terms of MAE for summer, autumn, and winter.

However, we see that the Kroenecker LCM MT-GP performs consistently well in terms of MAE, only beaten by the VAR model during winter, and with competitive performance to most benchmarks. For the other seasons, the Kroenecker LCM MT-GP maintains its competitiveness, and displays a considerably low standard deviation compared to our benchmarks.

For the NLPD metric, we observe the ES and SES model having a lower NLPD for the spring season but is beaten by the both the Kroenecker LCM MT-GP and the Simple GP for the remaining seasons. This validates the GP models' ability to better quantify its uncertainty in its prediction compared to the Exponential Smoothing models.

Figure 4.2 clearly demonstrates the superior performance of the Kroenecker LCM-MT GP in comparison to the Simple GP, underlining the model's adeptness in leveraging task-related information. However, it is noteworthy that either the Exponential Smoothing models or the VAR model outperform the Kroenecker LCM MT-GP for all four seasons in terms of MAE.

During our hyperparameter tuning, we made the following observation: a high dispersion scale from (ν) our beta likelihood tended to yield a lower MAE at the expense of a reduced variance in the predictive distribution. A high dispersion scale allowed the GP models to fit larger fluctuations to a higher extend at the expense of low variance. This phenomenon is further illustrated in Figure 3.9. It is plausible that this behavior explains why the GP models exhibit a higher MAE as it can tend to generate smoother approximations

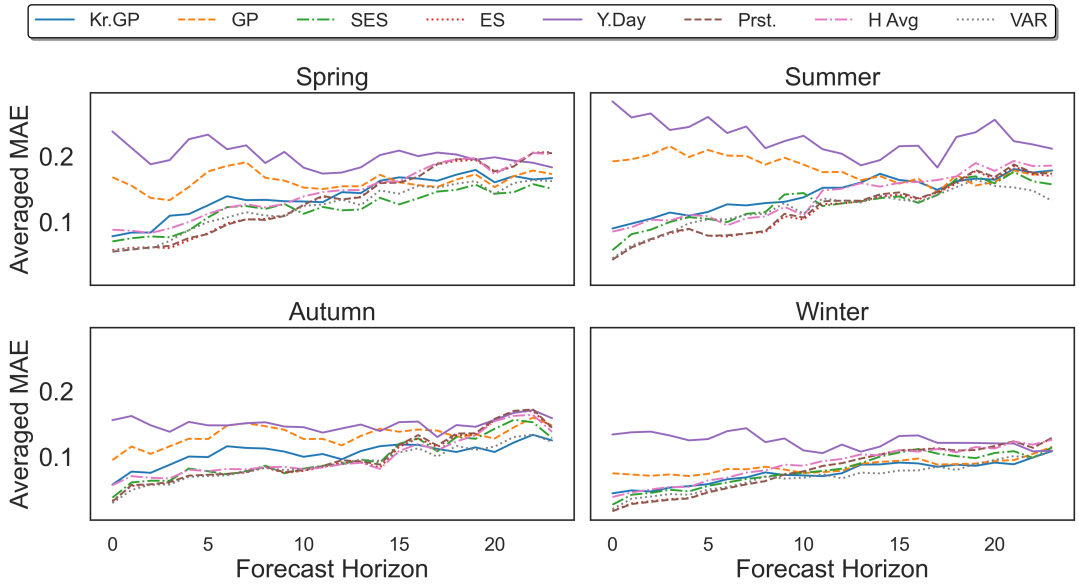


Figure 4.2: The plot depicts the average Mean Absolute Error (MAE) across 24 time steps for our temporal benchmarks and the Kroenecker LCM MT-GP (abbreviated as Kr.GP). Notably, the Simple GP (denoted GP) struggles to extrapolate accurately within the temporal context. Conversely, the SES, ES, and VAR benchmarks consistently demonstrate strong performance, aligning with the findings from [6].

of the historical trend, potentially at the cost of capturing fluctuations in PV output, which we observed for lower dispersion scales (See Figure 3.13). Even though the SES, ES, and VAR models also serve as moving average models, their capabilities of fitting shorter-term fluctuations and adapting to transient changes in the data can potentially give them an advantage in capturing PV output dynamics more effectively, especially for predictions close to the observed past.

Furthermore, the higher MAEs may be attributed to the Matérn kernel gaining dominance over the quasi-periodic component of our base kernel during the optimization process. Such a shift in dominance can lead to decreased extrapolation capabilities for the GP models. Specifically, the heightened fluctuations observed during the spring and summer seasons can cause the periodic component to adapt to short-term fluctuations instead of daily trends, while the Matérn component becomes more prominent in mod-

eling the overall trend. Consequently, this shift may diminish the GP models' ability to accurately extrapolate the inherent periodicity in the PV data. This could explain the increased MAEs for our Simple GP during the spring- and summer seasons.

It would be worthwhile to consider longer training periods than our weekly folds and to closely monitor the convergence of periodicity and signal variance, which specifies the amplitude of the periodic trend. Unfortunately, due to time constraints, we were unable to conduct an in-depth investigation in this regard.

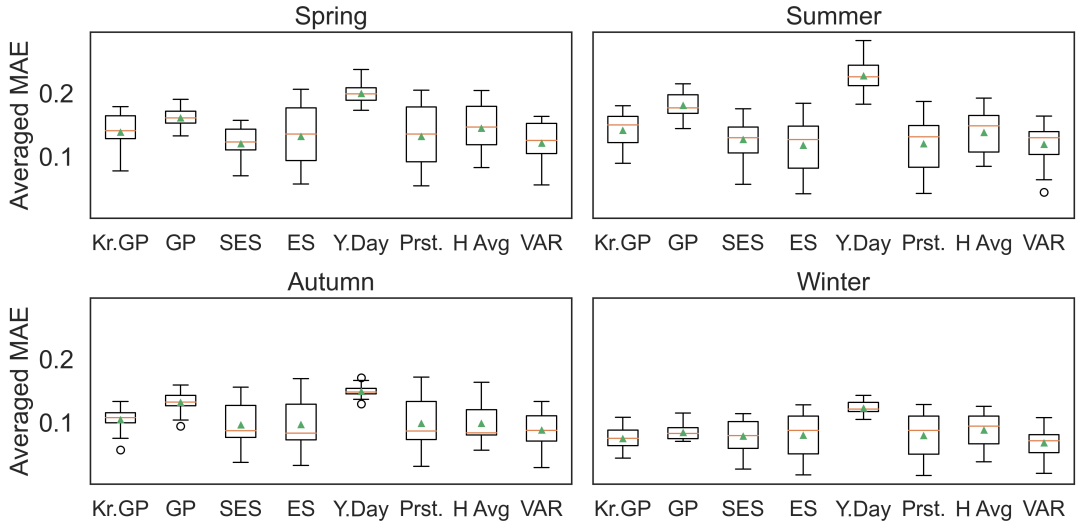


Figure 4.3: The boxplot displays the interquartile range (IQR), spanning from the 25th percentile to the 75th percentile of the Mean Absolute Error (MAE) for each model. Median values for each model are indicated by orange lines within the boxes, and the mean values are represented by green triangles. Notably, our GP models exhibit a narrower spread of MAE errors compared to our statistical benchmarks. Of particular interest is the VAR model, which demonstrates robust performance across seasons. Similar to the Kroenecker LCM MT-GP, it is the only benchmark leveraging inter-task correlations.

In Figure 4.3, we present the distribution of Mean Absolute Error (MAE) values using a boxplot, illustrating their statistical quantiles. Within each boxplot, the interquartile range (IQR) spans from the 25th percentile (Q1) to the 75th percentile (Q3) of the MAE for each model. The line inside the box represents the median value, while the triangle indicates the mean value.

Notably, the statistical benchmarks, with the exception of Y.Day, display a wider spread of MAE quantiles compared to our GP models. Particularly striking is the exceptionally low values observed at the 25th percentile of the statistical benchmarks. This is likely attributed to their moving averages at early time steps, which can closely match the most recent observed values. This can explain why these models exhibit greater deviations from these values as the forecasting horizon extends.

We further illustrate this effect by examining the IQRs of MAE errors over our forecast horizons in Figure 4.4, where we compare our GP models against the VAR model, which is the top-performing benchmark. While the VAR model boasts relatively low median and mean MAE values, these values experience a significant increase as the forecasting horizon extends. The same trend was observed for the other statistical benchmarks. This trend is also apparent in our Kroenecker LCM MT-GP model, albeit its early MAE errors has a higher level than those of the VAR model. Once again, this behavior may be attributed to the aforementioned argument of generating smoother approximations.

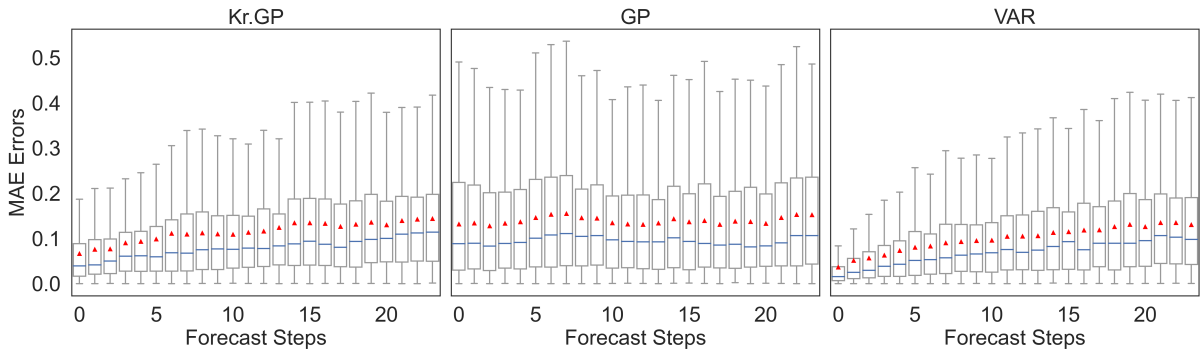


Figure 4.4: The plot of IQR ranges given by the forecast horizons averaged across 52 weeks of data for six PV stations. We can see the VAR has exceptionally low values for early forecasts that rapidly increase with respect to the forecast horizon. The blue horizontal lines denote the median whereas the red whisker denotes the mean value.

As evident in Table 4.1, our GP models demonstrate slightly lower mean NLPDs compared to our ES and SES benchmarks. Figure 4.5 shows how the NLPD metrics of our

temporal models evolve over the forecast horizon. The GP models' 95% Confidence Intervals (CIs) indicate superior uncertainty estimation in the short-term, with a tendency toward lower NLPDs. In contrast, ES models tend to exhibit higher NLPDs in the long term. Over the long-term, the difference in NLPD between GP and Exponential Smoothing benchmarks remains negligible. Additionally, we can observe the upper 95% CI of the SES model has a spike in the long-term whereas the ES model has a similar spike for short-term NLPD.

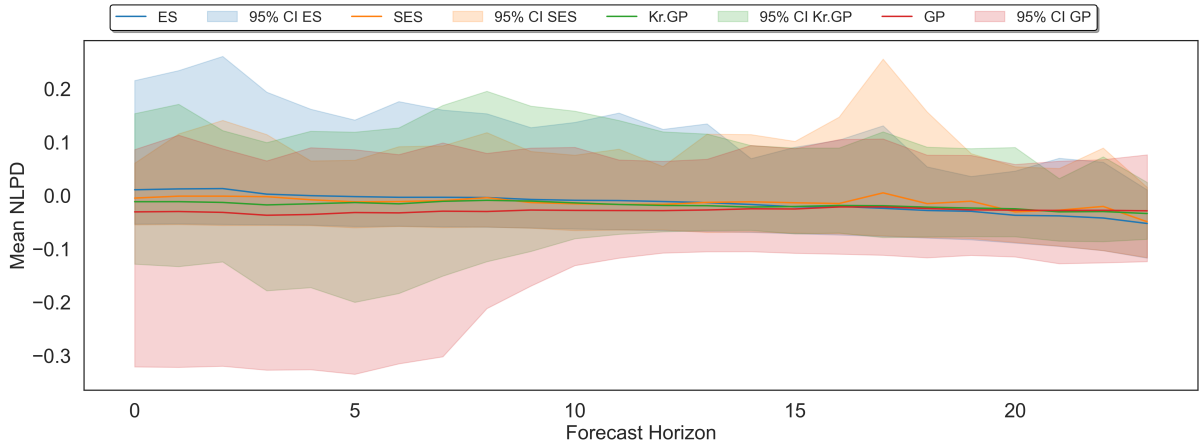


Figure 4.5: The mean NLPDs of the probabilistic temporal models over the 2 hour forecast horizon.

In the study carried out in [6], which consider the same benchmarks, we can observe significantly lower NLPD values for their state-space GP model. We posit two potential reasons for this discrepancy.

Firstly, the authors do not provide a clear definition of how the NLPD metric is computed. It appears that their NLPD results for the ES and SES models are significantly lower. This suggests a potential normalization occurring within the logarithmic probability computation. More specifically, an NLPD defined by

$$\text{NLPD} = \sum_{t=1}^{T} \log\left(\frac{1}{T} p(y_t \mid \mathcal{D}, \theta)\right).$$

This effectively shifts the values inside the logarithm towards lower magnitudes, resulting in a lower NLPD score. It is worth noting that this method deviates from the standard approach in NLPD calculations.

Second, when examining the plots of their resulted fitting and predictions we observe that the training data fits the noisy fluctuations in PV values during training. This can imply a high dispersion scale in the beta likelihood, resulting in sharply peaked distributions. Despite this, the predictions maintain a high variance. We suspect that the elevated predictive variance may arise from a very low lengthscale. As a consequence, many values may fall outside the uncertainty intervals drawn from the MC samples of these distributions. This, in turn, could yield infinite values discarded in the NLPD metric computation, potentially providing a distorted view of the reported NLPD metric by discarding a large proportion of the test data in the computation.

When comparing the GP models, both demonstrate a strong ability to effectively quantify predictive uncertainty, as illustrated in Figure 4.6. The vast majority of PV values fall within the 95% CI for both GP models. However, the Simple GP tends to have a higher count of values within the 95% CI, consistent with the results presented in our NLPDs in Table 4.1. This observation suggests the possibility that inter-task learning may be overly confident in certain settings.

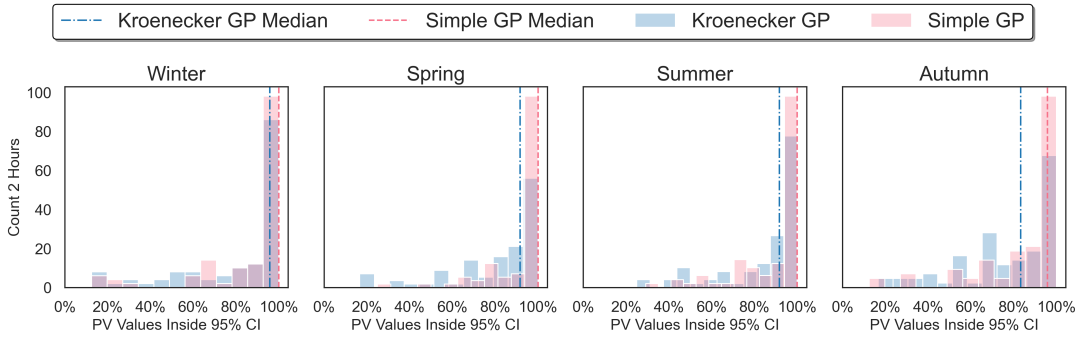


Figure 4.6: The count of PV values falling inside the 95% CI for the GP models. We can see that the Simple GP consistently holds a higher count of values falling inside its 95% CI compared to the Kroenecker LCM MT-GP.

4.2.2 Exogenous Results

In this section, we explore the impact of weather data on PV nowcasting models. In the temporal setting, we assume the models capture short-term dependencies of our PV values. As we include exogenous regressors, we suspect this horizon can be extended with contextual information given by the weather parameters. Therefore, we will investigate both 2- and 6 hour nowcasts in this regime.

Our primary model of interest is the Hadamard LCM MT-GP, while the Simple GP serves as a benchmark for evaluating the influence of modeling inter-task correlations as before. Both the Hadamard GP and Simple GP incorporate exogenous regressors using a Squared Exponential (SE) kernel. For both models we subsequently construct the input kernel to be defined by a product between the SE kernel taking weather data as input and a quasi-periodic kernel for temporal data (detailed in Section 3.2.3).

In addition to the GP models, our benchmarks include Bayesian Ridge Regression (BRR), Long-Short-Term Memory (LSTM) neural networks, and XGBoost (XGB) models. These benchmarks only utilize the weather data. A note on the BRR model is it can be interpreted as a GP with a linear kernel. Inherently, this implies a linear relationship between PV values and weather variables.

4.2.2.1 Forecasting 2 hours

Season	Metrics	Hadamard GP	Simple GP	BRR	LSTM	XGB
Spring	**MAE**	**0.114**	0.119	0.129	0.206	0.127
	Std Dev	**0.110**	0.114	0.116	0.183	0.127
	NLPD	**-0.642**	-0.640	-0.247	N/A	N/A
	Std Dev	**0.821**	0.894	1.191	N/A	N/A
Summer	**MAE**	**0.107**	0.108	0.116	0.210	0.126
	Std Dev	0.104	0.095	**0.094**	0.173	0.122
	NLPD	**-0.747**	-0.705	-0.296	N/A	N/A
	Std Dev	0.834	**0.506**	0.543	N/A	N/A
Autumn	**MAE**	0.096	**0.091**	0.099	0.184	0.111
	Std Dev	0.090	0.099	**0.088**	0.176	0.111
	NLPD	-1.048	**-1.053**	-0.365	N/A	N/A
	Std Dev	1.100	1.112	**0.624**	N/A	N/A
Winter	**MAE**	0.066	**0.064**	0.065	0.127	0.086
	Std Dev	0.085	0.086	**0.075**	0.146	0.110
	NLPD	**-1.527**	-1.083	-0.029	N/A	N/A
	Std Dev	2.421	2.450	**0.599**	N/A	N/A

Table 4.2: Results for 2 hour nowcasts with weather parameters as input. Results are reported with respect to mean values across the experiments along with the standard deviation (abbrv. Std. Dev.). We can see that the Hadamard LCM MT-GP marginally beats the benchmarks in terms of MAE and NLPD for all seasons except from the winter where the BRR has an MAE that is 0.001 lower.

Table 4.2 summarizes the outcomes of 2-hour PV nowcasts incorporating weather parameters, illustrating model performance across various seasons. Notably, the Hadamard LCM MT-GP demonstrates a slight advantage over benchmarks in Mean Absolute Error (MAE) and Negative Log Predictive Density (NLPD) for spring and summer. However, the competitive performance of the Simple GP, closely tracking the Hadamard LCM MT-GP, raises questions about the benefit of modeling inter-task correlations. During autumn and winter, the Simple GP marginally outperforms the Hadamard LCM MT-GP in MAE, but in winter, the Hadamard LCM MT-GP substantially improves NLPD. This improvement may stem from the Hadamard LCM MT-GP effectively capturing inter-dependencies between PV stations, as the winter can have more consistent trends and less fluctuations across stations. This could enhance the resulting uncertainty quantifi-

cation of the Hadamard LCM MT-GP compared to the Simple GP.

Comparing our best-performing exogenous- and temporal models demonstrates a consistent improvement in predictive accuracy across seasons. In spring, the SES model yielded an MAE of 0.121 ± 0.123 and an NLPD of -0.417 ± 1.286, while the Hadamard LCM MT-GP performed an MAE of 0.114 ± 0.110 and an NLPD of -0.642 ± 0.821. For the summer, the ES model achieved an MAE of 0.119 ± 0.127 whereas the Hadamard LCM MT-GP had an MAE of 0.107 ± 0.104 and an NLPD of -0.747 ± 0.834. Only for the autumn does the statistical benchmark beat the model with exogenous regressors. The VAR model recorded an MAE of 0.089 ± 0.092, which was lower than the best performing exogenous model. Here, the Simple GP with weather variables recorded an MAE of 0.091 ± 0.099 but its NLPD outperformed all temporal models. Lastly, during winter, the VAR model achieved an MAE of 0.069 ± 0.084, and the simple GP outperformed it with an MAE of 0.066 ± 0.085. In terms of NLPD for the winter, the Hadamard LCM MT-GP shows an NLPD of -1.527 ± 2.421 compared to the NLPD from the temporal Simple GP, which was -0.988 ± 1.835. This consistent improvement underscores the enhanced predictive accuracy achieved by incorporating weather variables across all seasons.

Interestingly, the BRR model demonstrates a notably stronger competitive performance than the LSTM and XGB models in terms of MAE. This unexpected outcome could be attributed to the assumption that relationships between PV values and weather parameters may be linear. This is indeed observed for the global radiation from Figure 3.4, which showed the strongest correlation with PV output. Also, the penalization term in BRR effectively regularizes its feature weighting, which can prevent overfitting. Conversely, LSTM and XGB may over-parameterize and overfit the data, especially as we considered training on weekly folds of data. This introduces a limitation in our benchmarking approach, as these models can inherently handle larger amounts of training data, which we did not fully explore due to time constraints. Consequently, we did not assess their performance in scenarios where they are exposed to larger training sets, which could

impact their predictive performance and reduce the potential overfitting.

In Figure 4.7, it is evident that the LSTM model, which is the most parameterized model, exhibits quite poor performance relative to the other models. This poor performance can be caused by the model's complexity, which can result in the LSTM learning training data idiosyncrasies instead of generalized patterns. In contrast, the Hadamard LCM MT-GP, Simple GP, and BRR models demonstrate consistently low MAE across the different seasons with negligible differences between them in mean MAE over the 2 hour forecast horizon.

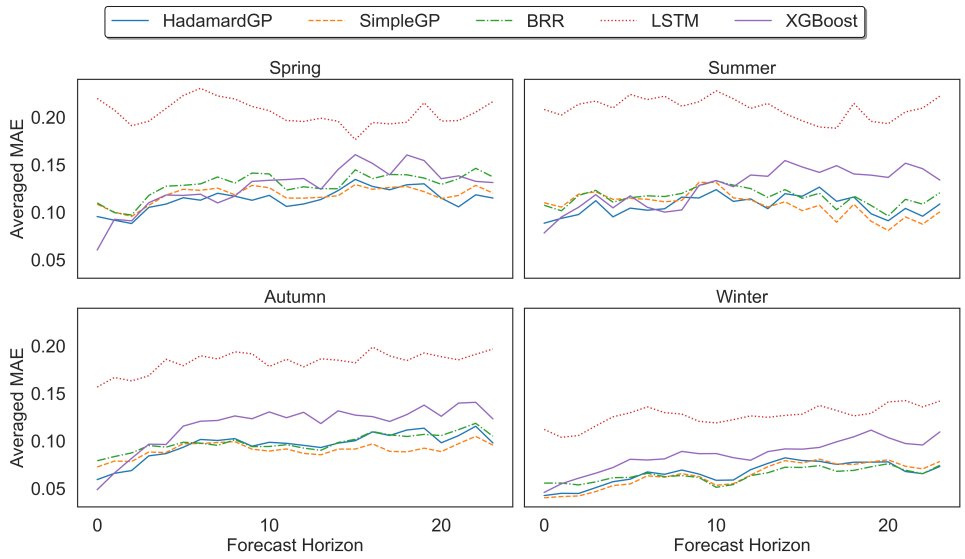


Figure 4.7: The IQR Ranges of models incorporating weather variables for the 2-Hour forecast horizon. The compact IQR ranges signify consistent predictive performance among the models. The horizontal blue line represents the median and the red whisker shows the mean.

Comparing our models in the exogenous setting to the temporal setting, as evident in Figure 4.8, reveals a substantial reduction within the IQR of MAE errors. The boxplot further illustrates the consistent and low IQR ranges of MAE for the Hadamard LCM MT-GP, Simple GP, and BRR models compared to their temporal counterparts. Unsurprisingly, this means our model benefits from the use of weather parameters. Compared

to Figure 4.3, we do not see equally low 25th percentiles in this setting. Specifically, when examining Figure 4.4, we saw the VAR model exhibits exceptionally low MAE errors in the early phase, which can be attributed to its close fitting to the most recent observed data, meaning it would capture noise. In contrast, this is not prevalent for our GP models, which indicate that in both the temporal and exogenous settings they likely prioritize fitting the overall trajectory of PV output, resulting in predictions with less sensitivity to noise (demonstrated in Appendix B).

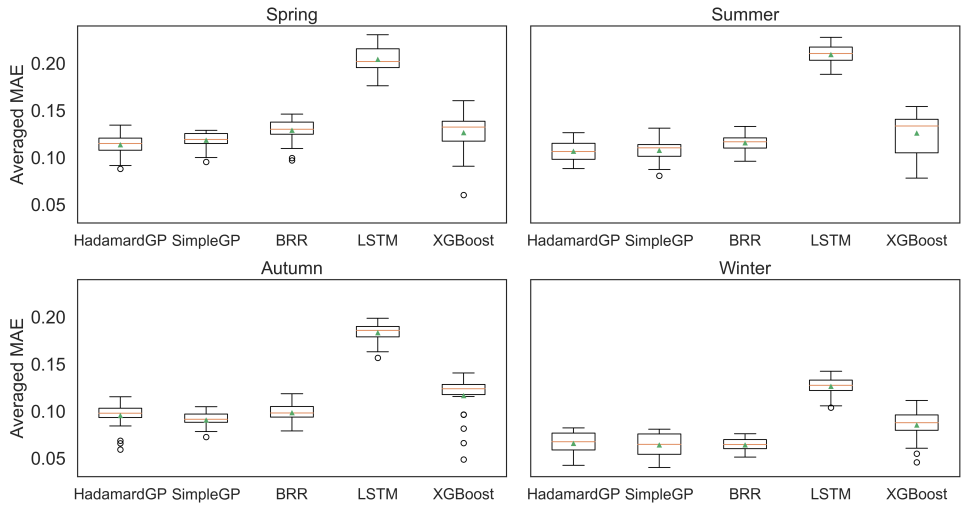


Figure 4.8: The averaged MAE of the 2-hour nowcasts incorporating weather parameters for the different seasons. We see that the IQR is significantly reduced compared to our statistical benchmarks.

In Figure 4.9, we observe that incorporating weather variables leads to a reduction in the lower 75th percentile of forecast errors, especially in the long term, compared to Figure 4.4. While the median and mean indicate that the models generally yield low errors on average, it is noteworthy that the 75th percentile extends to errors as high as 0.3. Therefore, an assessment of the percentage of PV values falling within the 95% CI is warranted to assess how effectively the predictive distribution accounts for prediction uncertainty. We will return to this in Section 4.2.2.2.

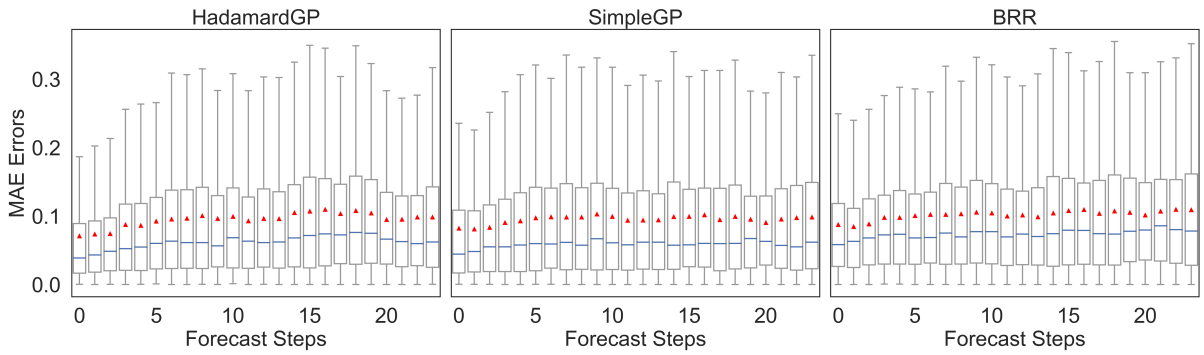


Figure 4.9: The IQR averaged over 2 hour forecasts across 52 weeks of PV data from the six stations for the top-performing models. we can observe that the overall IQR remains relatively consistent as the forecast horizon increases.

From the mean NLPD values over the forecast horizon, we observe a strong overlap between the Hadamard LCM MT-GP and the Simple GP model, as depicted in Figure 4.10. The same is true for their upper and lower quantiles. We can, however, see that the GP models on average performs better than the BRR benchmark. Additionally, we observe the same trends as for the temporal setting, with the lower 95% quantile being lower at early time steps. This could be attributed to the GPs smoothing out short-term fluctuations in the PV output which can lead to less volatile predictions and extended uncertainty for the short-term test points.

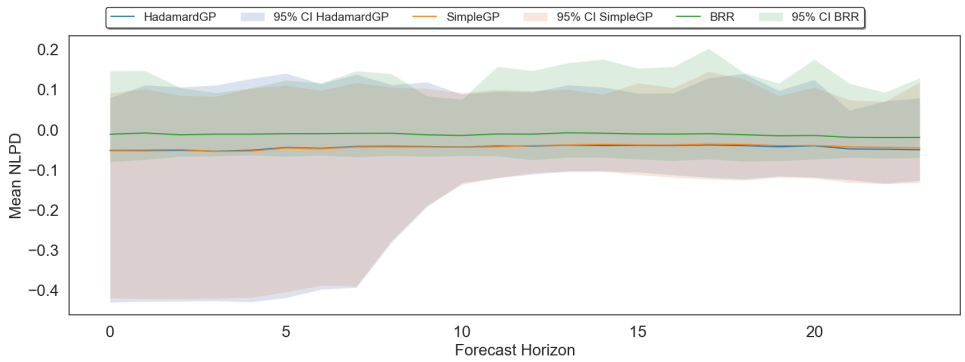


Figure 4.10: Average NLPD across 2-hour forecasts over 52 weeks from six PV stations for our probabilistic models utilizing weather data. The GP models exhibit substantial overlap, yielding lower NLPDs than the BRR model.

4.2.2.2 Forecasting 6 hours

Season	Metrics	Hadamard GP	Simple GP	BRR	LSTM	XGB
Spring	**MAE**	**0.119**	0.152	0.126	0.158	0.124
	Std Dev	0.109	0.134	0.114	0.143	0.122
	NLPD	**-0.606**	-0.477	-0.185	N/A	N/A
	Std Dev	0.710	1.102	**0.612**	N/A	N/A
Summer	**MAE**	**0.119**	0.129	0.132	0.147	0.150
	Std Dev	0.108	0.115	0.107	0.133	0.139
	NLPD	**-0.608**	-0.475	-0.211	N/A	N/A
	Std Dev	0.673	0.627	**0.529**	N/A	N/A
Autumn	**MAE**	0.098	**0.087**	0.092	0.117	0.117
	Std Dev	0.097	0.090	0.083	0.119	0.115
	NLPD	-0.948	-0.957	-0.410	N/A	N/A
	Std Dev	0.831	1.001	**0.466**	N/A	N/A
Winter	**MAE**	**0.059**	0.069	0.063	0.085	0.086
	Std Dev	0.064	0.080	0.067	0.091	0.085
	NLPD	**-1.926**	-1.666	-0.568	N/A	N/A
	Std Dev	0.603	0.760	**0.455**	N/A	N/A

Table 4.3: Results for 6 hour nowcasts using weather parameters. Results are reported with respect to the mean value across the experiments with standard deviation, denoted Std.Dev.

Table 4.3 reveals the Hadamard LCM MT-GP's consistent performance, particularly in mean MAE and NLPD during seasons with higher PV output fluctuations, as seen for summer and spring. This suggests that inter-task modeling may offer advantages when dealing with higher noise levels when including weather parameters. Additionally, we see the Simple GP and BRR models demonstrate competitive performance whereas the LSTM and XGB model suffer from the same issues addressed in the preceding section.

The variation in performance metrics aligns with the seasonal fluctuation in PV output, as evidenced by the higher average MAE and NLPDs observed during these seasons in comparison to autumn and winter. Notably, during autumn the Simple GP marginally outperforms the Hadamard LCM MT-GP in MAE and NLPD. In contrast, for the winter the Hadamard LCM-MT GP achieves marginally better MAE, while demonstrating notably lower NLPD values compared to the Simple GP and the BRR models.

Compared to Table 4.2, we observe minimal changes in performance. This suggests that as we extend the forecast horizon from two to six hours, the models maintain their predictive accuracy. Consequently, this indicate that the incorporation of weather variables provides stability to the resulting predictions.

Whilst we observed an increase in forecast errors over the 2 hour nowcasting scheme, Figure 4.11 suggest that the average MAE fluctuates with a relatively flat trend over the 6 hour horizon. The lowest MAE errors are generally observed at early time steps, but we can see that the upwards trend of MAE errors as the forecast horizon extends is less evident, especially compared to our temporal setting. This suggests, as also evident from Table 4.3, that models with weather parameters maintain relatively stable predictions over extended time frames. Thus, integrating weather data unsurprisingly provides reliability of PV output forecasts for longer horizons. We can also notice from Figure 4.11 that the Hadamard LCM MT-GP on average hold lower MAEs than our benchmarks for the summer- and spring seasons.

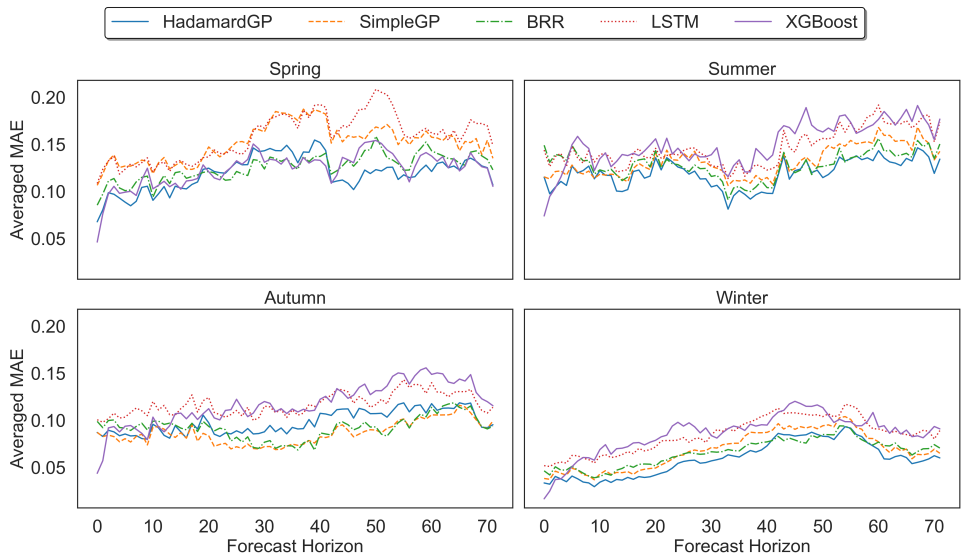


Figure 4.11: The average MAE over a 6-hour forecast horizon for six stations over 52 weeks of PV data, which corresponds to 72 time steps given our 5-minute temporal resolution.

To provide a view of the forecast error distribution, Figure 4.12 highlights that the mean (green whisker) and median (orange line across the boxes) errors align closely with those of the 2-hour nowcasts. However, a marginal increase in variability is noticeable in the 25th and 75th percentiles, as anticipated with the inclusion of additional data points.

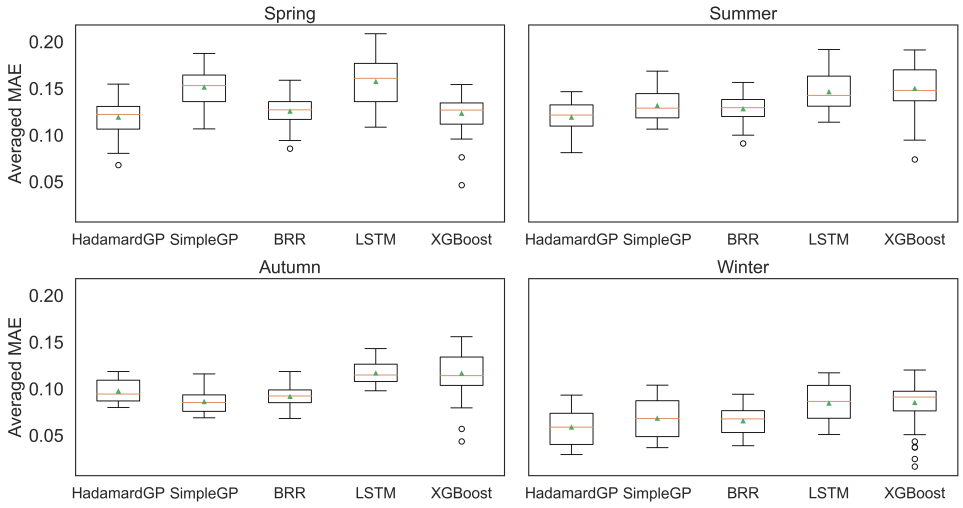


Figure 4.12: The IQR of our models, which include weather parameters for a six-hour forecast horizon, derived from predictions made across 52 weeks of data for six PV stations, with predictions generated for each week.

Figure 4.13 presents the same IQR as previously seen, representing the mean MAE across all seasons. Notably, the Hadamard LCM MT-GP demonstrates slightly lower 75th percentiles compared to the Simple GP. It is worth noting that while the mean and median values are low for all models, the 75th percentiles approach 0.4, indicating the predictions with relatively high errors.

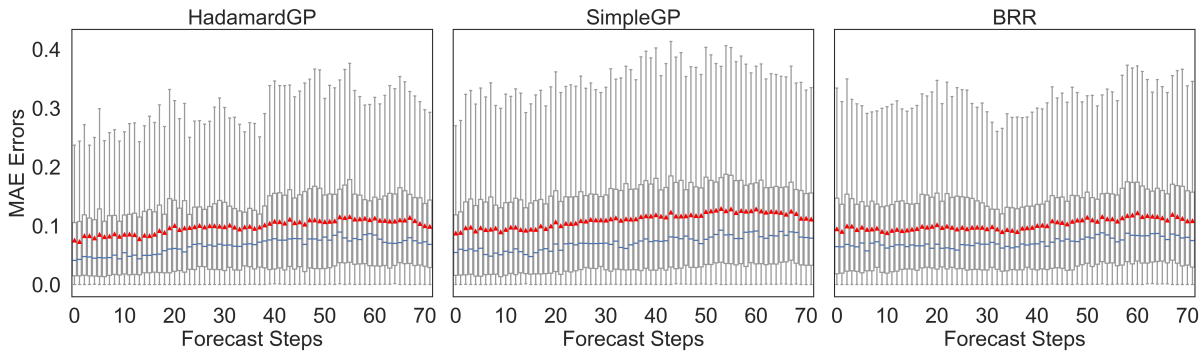


Figure 4.13: The IQR of MAE errors over time averaged across 52 weeks for the six PV stations for our best performing models. The discrepancies between median and mean values indicate a positively skewed error distribution with occasional large errors, as observed in previous experiments. The positive skewness could stem from the aggregation of seasons as PV fluctuations are higher for spring and summer.

Similar to the 2 hour nowcasts, Figure 4.14 demonstrates a large overlap between the Hadamard LCM MT-GP and the simple GP. For longer time horizons, we can see that the upper 95% CI of the Simple GP is marginally higher than the Hadamard LCM MT-GP.

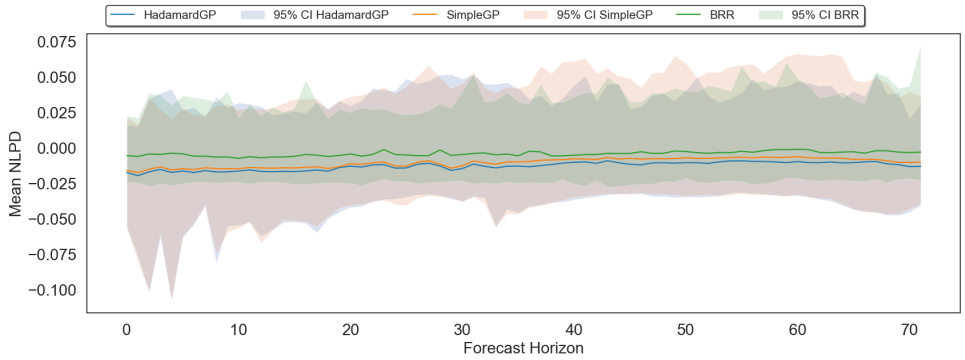


Figure 4.14: The mean and 95% CI of NLPD for the probabilistic models over time, averaged across 52 weeks for six PV stations.

In both the two-hour and six-hour nowcasting scenarios, we observed error distributions characterized by consistently low mean and median values across time steps. However, a larger skew towards higher prediction errors is apparent in both the Hadamard LCM MT-GP and the Simple GP models. Consequently, it is important that these models

accurately quantify their predictive uncertainties within the predictive posterior distribution. We demonstrate this in Figure 4.15 by visualizing the count of PV values falling within the 95% CI.

In the two-hour setting, the vast majority of PV values, for both models, fall within this interval, confirming that the GP models effectively quantify their predictive uncertainties within this time frame. For this setting, the discrepancy between the Hadamard LCM MT-GP and the Simple GP is negligible. In contrast, when we extend to six hours we observe that while most PV values reside within the 95% CI, the distribution of counts is more skewed. This variation can be attributed to increased exposure to fluctuations as the forecast horizon extends. As depicted in Appendix B, our GP models operate as adept smooth approximators. Consequently, they tend to be sensitive to highly fluctuating PV values, leading to a greater proportion falling outside the 95% CI when exposed to more noisy behavior. Interestingly, the Simple GP tends to have higher counts of values inside the 95% CI even though its NLPD is higher compared compared to the Hadamard LCM MT-GP. This is indicative of it asserting less confidence in its predictive distribution.

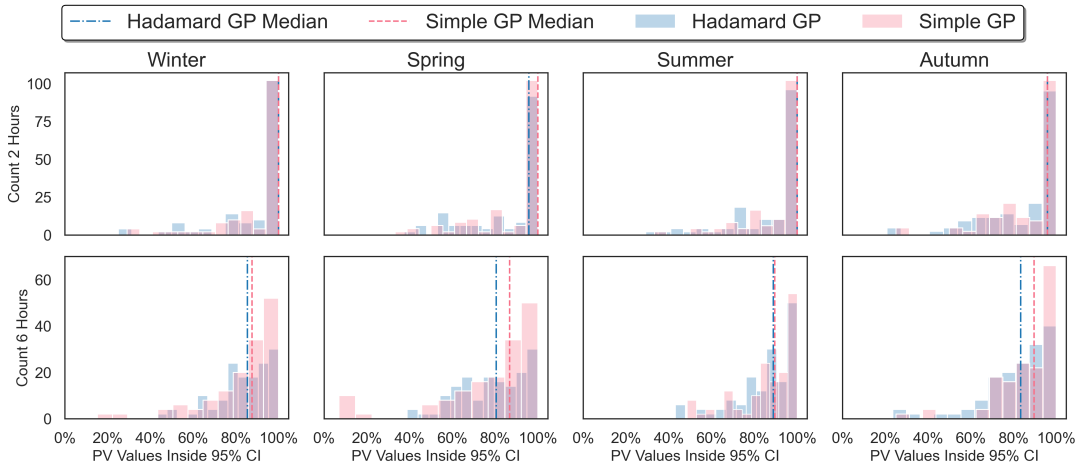


Figure 4.15: Count of PV values falling within the 95% CI for the Hadamard LCM MT-GP (Hadamard GP) and the Simple GP across seasons. The top row displays counts for the two-hour nowcasts, while the bottom row illustrates counts for the six-hour nowcasts.

4.2.3 Independent Gaussian Processes vs. Multitask Gaussian Processes

In this section, we compare the LCM MT-GP framework with independent GPs. Our aim is to evaluate the advantage of multitask learning and assess how the inclusion of weather parameters enhances predictive accuracy. To obtain a holistic view of their performance, we aggregate the results for the entire year (2018/01/01-2018/31/12). Furthermore, we extend our analysis by incorporating Automatic Relevance Determination (ARD) [45] for both models, a technique that assigns separate lengthscales to each dimension in the feature space. We consistently employ the SE kernel for the weather parameters, ensuring a transparent basis for comparison.

Model	Weather Parameters		Temporal	
	MAE	NLPD	MAE	NLPD
LCM MT-GP 2hr	0.096 ± 0.099	-0.951 ± 1.561	0.116 ± 0.113	-0.491 ± 1.064
ARD LCM MT-GP 2hr	**0.094 ± 0.098**	**-1.002 ± 1.582**	N/A	N/A
Simple GP 2hr	0.095 ± 0.101	-0.855 ± 1.546	0.145 ± 0.138	-0.485 ± 1.375
ARD Simple GP 2hr	**0.094 ± 0.102**	-0.904 ± 1.669	N/A	N/A
LCM MT-GP 6hr	0.099 ± 0.100	**-0.903 ± 0.869**	N/A	N/A
ARD LCM MT-GP 6hr	0.104 ± 0.107	-0.809 ± 1.274	N/A	N/A
Simple GP 6hr	0.111 ± 0.113	-0.722 ± 1.035	N/A	N/A
ARD Simple GP 6hr	**0.096 ± 0.103**	-0.822 ± 1.265	N/A	N/A

Table 4.4: Results for the LCM MT-GP and Simple GP models. In the weather parameters column, LCM MT-GP specifically refers to the Hadamard LCM MT-GP, while in the temporal column, it refers to the Kroenecker LCM MT-GP. ARD indicates the inclusion of a lengthscale for each feature dimension when using the SE kernel. Models without the ARD extension assume an SE kernel with a shared lengthscale.

In Table 4.4 we observe, as before, a negligible difference in MAE between the Hadamard LCM MT-GP and the Simple GP when we include weather parameters whereas in the temporal setting the Kroenecker LCM MT-GP outperforms the Simple GP. However, the NLPD of the Hadamard LCM MT-GP is noticeably better but as we saw from Figures 4.6 and 4.15 it contains less values inside its 95% CI. This discrepancy suggests that the LCM MT-GP may occasionally exhibit excessive confidence in its uncertainty quantification, possibly resulting in predictions that are too narrow.

Assigning a shared lengthscale to all input features assumes all input features have the same relevance on predictions. In theory, applying ARD would be sensible as we cannot assume all input features carry equal importance. ARD can effectively perform dimensionality reduction by assigning a long lengthscale to features that are considered less relevant. This would reduce the influence of noisy/irrelevant dimensions, potentially leading to more accurate models. From Table 4.4, however, the improvements associated with ARD are limited, with a notable exception for the Simple GP in the 6-hour setting. This observation may imply that all the weather parameters may hold comparable relevance for PV nowcasting.

4.3 An Ablation: Approximate Latent Force Models

When using Gaussian Processes (GPs) for modeling, memory and computation issues arise as pointed out in Section 2.2.3. This limits Vanilla GP to smaller data subsets, constraining exposure to the full data distribution. Thereby, the model can suffer from epistemic uncertainty.

In physical sciences, many phenomena are represented by differential equations. A large proportion of these problems have been approached by the use of numerical methods, such as Runge-Kutta models. Leveraging these models, machine learning approaches, such as neural ordinary differential equations (ODEs) [46] and non-parametric ODEs [47], combine numerical methods with black-box ML models. However, these loose interpretability by a lack of explicit equations and the ability to infer latent forces.

Whilst maintaining interpretability and the limitations of purely mechanical or data-driven models, Latent Force Models (LFMs) introduce a hybrid system by merging simplified mechanical models and GPs [48]. However, an intractable integration over the kernel function can limit the usability of LFMs. An approximate LFM [49] address this drawback by using variational inference.

4.3.1 Model Definition

LFMs are differential equation models leveraging the dynamics of a system to infer latent forcing terms. The ODE equation $\Delta\mathbf{y}$, parameterized by $\mathbf{\Phi}$, provide a structural relationship between the output $\mathbf{y} \in \mathbb{R}^N$ and D latent forces $\mathbf{f}(\mathbf{t}) \in \mathbb{R}^{N \times D}$ over time. Assuming we have discrete points $\mathbf{t} \in \mathbb{R}^N$, we assign GP priors to the latent forces $f_i \sim \mathcal{GP}(\mathbf{0}, k_i(t, t')$ for $i = 1, \ldots, D$. These can be mixed by a non-linear response $G(\mathbf{f})$, giving us the following ODE:

$$\overbrace{\Delta\mathbf{y}(t)}^{\text{differential}} = \overbrace{g(\mathbf{y}, \mathbf{t}, G(\mathbf{f}), \mathbf{\Phi})}^{\text{differential equation}} \tag{4.3}$$

where Δ is any n^{th} order derivative of the ODE and $\mathbf{\Phi}$ is the ODE parameters. An analytical expression of the covariance functions can be obtained when G is linear, which means the marginal likelihood gives the ODE parameters and can be used for inference under standard GP posterior identities assuming a conjugate likelihood. Otherwise, one can use variational inference as in Section 3.2.5 to construct Approximate LFMs (ALFMs). This greatly expands the class of models that can benefit from interpretable LFMs [49]. By leveraging variational inference, we can impose any differentiable kernel and freely select the likelihood on our output. Let $\mathbf{f} = \{f_k\}_{k=1}^K$ be a set of K latent forces where $f_k \sim \mathcal{GP}(\mathbf{0}, k_\theta(\mathbf{x}, \mathbf{x}'))$ with kernel parameters θ. We seek to optimize the parameters by marginalizing over the latent forces:

$$p\left(\mathbf{y} \mid \mathbf{\Phi}, \theta\right) = \int p\left(\mathbf{y} \mid \mathbf{f}, \mathbf{\Phi}\right) \prod_{k=1}^{K} p(f_k \mid \mathbf{X}, \theta) \tag{4.4}$$

If the likelihood is non-conjugate or G is non-linear. We cannot derive the posterior over the latent forces. Therefore, we use a variational approximation by leveraging the ELBO

for the marginal likelihood:

$$\log p\left(\mathbf{y} \mid \theta, \mathbf{\Phi}\right) = \mathbb{E}_{q(\mathbf{f})}\left[\log p\left(\mathbf{y} \mid G(\mathbf{f}), \mathbf{\Phi}\right)\right] - D_{\mathrm{KL}}\left(q_\lambda(\mathbf{u}) \,||\, p(\mathbf{u})\right) \tag{4.5}$$

where we use $M \ll N$ inducing points $\mathbf{U} = \{\mathbf{u}_i^T\}_{m=1}^M$ being the inducing points for latent force f_k and $q_\lambda(\mathbf{f}) = \int p(\mathbf{f} \mid \mathbf{U}) q_\lambda(\mathbf{U}) d\mathbf{U}$. The variational posterior is given by $p(f_k^* \,||\, \mathbf{u}_k) = \int p(f_k^* \mid \mathbf{u}_k) q_\lambda(\mathbf{u}_k) d\mathbf{u}_k = \mathcal{N}(f_k^* \mid \mathbf{m}_k^*, \mathbf{S}_k^*)$ where

$$\mathbf{m}_k^* = K_{*M} K_{MM}^{-1} m_k \tag{4.6}$$

$$\mathbf{S}_K^* = K_{**} + K_{*M} K_{MM}^{-1} (\mathbf{S}_K - K_{MM}) K_{MM}^{-1} K_M \tag{4.7}$$

The full details can be found in Appendix C.2.

While striving to find an ODE system for modeling PV output with NWP data, we encountered time-related challenges. Consequently, we turned our attention to showcasing the potential of LFMs for PV output modeling.

Consider the following non-linear set first-order ODEs where cloud cover (c) negatively affects PV output (p):

$$\frac{dp(t)}{dt} = \delta \cdot c(t) + \eta \sin\left(\omega \pi t\right) \cdot p(t) \tag{4.8}$$

$$\frac{dc(t)}{dt} = \zeta \cdot p(t) - \gamma \sin\left(\omega \pi t\right) \cdot c(t) \tag{4.9}$$

where $p(t)$ and $c(t)$ are the state variables for PV power and cloud coverage, respectively. δ and ζ affect the linear terms in rate of change of the equations, η and γ controls the rate of periodic change of each variable, and ω is the periodic frequency where we assume cloud coverage and PV outputs follow the same periodic frequency. We will come back to the drawbacks of the system in Section 4.3.3.

We assume the PV output is unknown and in [0,1] such that we assign a GP prior $p(t) \sim \mathcal{GP}(\mathbf{0}, k_\theta(t,t'))$ with a beta likelihood making the solution intractable. The aim is to recover the latent force, PV, and the equation parameters for cloud coverage $\mathbf{\Phi} = \{\zeta, \gamma, \omega\}$. The system has a clear periodic nature, as such it is sensible to assign a periodic kernel (See Section 2.3) for extrapolation beyond the range of our training data.

4.3.2 Results

We ran 100 simulations sampled from the above system where we sample $\zeta \sim \mathcal{U}(0.001, 0.1)$, $\gamma \sim \mathcal{U}(0.1, 0.5)$, and $\omega \sim \mathcal{U}(0.1, 0.2)$. Here, $\mathcal{U}(a,b)$ denotes the uniform distribution for two real numbers with $a < b$. The table below presents a comparison between the true parameters used in these simulations and the parameters predicted by our model.

Value	Mean	Std Deviation
$\lvert\zeta - \hat{\zeta}\rvert$	0.003	0.0035
$\lvert\gamma - \hat{\gamma}\rvert$	0.01	0.0092
$\lvert\omega - \hat{\omega}\rvert$	0.003	0.0012

Table 4.5: Mean Difference and Standard Deviation of Parameter Differences. $\hat{\zeta}$, $\hat{\gamma}$, and $\hat{\omega}$ denotes the predicted values for ζ, γ, and ω, respectively.

We can see that the small difference between the true values and the predicted values suggest the model is performing well in estimating the true parameters of the underlying ODE system. The low standard deviations of these differences further suggest that the model's predictions on average are very close to the true values. That means, we are able to overcome the complex integral over the kernels, which are both intractable and require a complex integration [48]. In context of PV nowcasting, predicting PV output can be significantly more challenging than other weather variables. As PV output is strongly correlated with these variables (see Figure 3.2), we can leverage the PV dynamic without directly knowing the PV output. In the figure below, we can see that the PV output is well recovered from the simplified system given by the ODE system above.

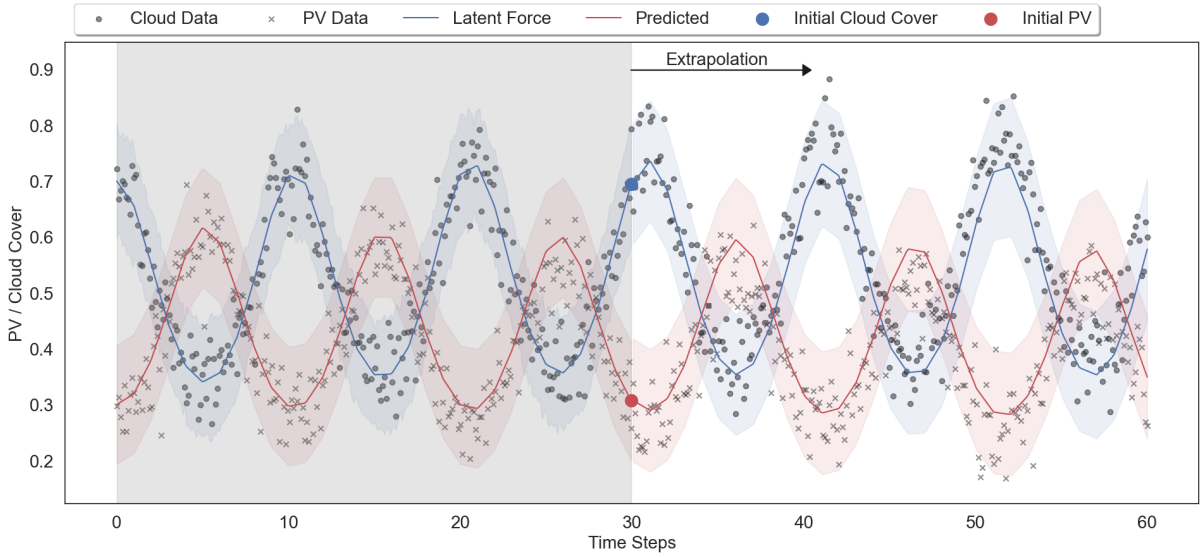


Figure 4.16: Result from the synthetic ALFM model with a periodic kernel. The shaded region represents our training region where the unshaded region represents our extrapolation space. The blue line indicates the predicted cloud cover. The grey circles for cloud data is the training data of cloud cover, which is the latent force. The PV data, marked by crosses, represent the unknown PV output not seen during training. In this simplified setting we see that the ALFM captures the PV dynamic without observing it directly.

4.3.3 Limitations

When constructing ODEs from PVs, we acknowledge the difficulty in accurate constructing an ODE system for PV output. We initially explored adapting the Stochastic Differential Equation framework from [50] but abandoned it due to time constraints and unsatisfactory results. There are two main obstacles that can make ALFMs for PV nowcasting challenging.

Firstly, weather conditions are inherently uncertain, making it difficult to accurately provide ODE forecasting as the uncertainty of the NWPs carries into the forecasting of PV in the ALFM model. Consequently, this limits the reliability of using ALFM to model PV output. However, the integration of GPs can help quantifying this uncertainty, which is the main attribute of the LFM framework in contrast to mechanical systems.

Secondly, to accurately extrapolate the PV forecasts when using ALFMs one relies heavily on having a kernel that can efficiently extrapolate from the latent forces. More often than not, a periodic kernel would not consistently align with the true PV dynamics, which raise a concern for ALFMs to efficiently learn PV dynamics. This necessitates a careful consideration of the kernel choice. The quasi-periodic kernel from Section 2.3.3 has been adapted for predicting both PV output and other weather phenomena [6, 51]. However, its capability to learn and extrapolate in the context of ALFMs was not investigated in this project.

In conclusion, this ablation study shed light on the potential of ALFMs as a promising avenue for PV output modeling. Regardless of encouraging results in an overly simplified (and inaccurate) system, it is important to acknowledge the inherent challenges. Accurately constructing ODEs for PV output from Numerical Weather Prediction (NWP) data remains a complex task, compounded by the inherent uncertainties in weather conditions and the limitations of kernel choices for extrapolation. However, ALFMs offer valuable tools for quantifying and addressing these uncertainties, making them a potential avenue for PV nowcasting. Future successful applications of ALFMs in the context of PV nowcasting will require tailored approaches that account for these challenges, leveraging the strengths of ALFMs while mitigating their limitations.

5 | Conclusion

This thesis explored the application of Linear Coregionalization Model Multitask Gaussian Processes (LCM MT-GPs) for PV nowcasting, with a primary focus on multitask learning for neighboring PV stations and the integration of weather parameters. Our central research question sought to understand the effectiveness of leveraging correlations between PV stations to enhance PV nowcasts and to what extent weather parameters contribute to improved predictive performance.

Our investigation revealed that in the temporal setting the Kroenecker LCM MT-GP exhibited superior performance when incorporating neighboring PV stations compared to independent GPs for each station. This underscored the potential of inter-task learning in this context. Furthermore, both the Kroenecker LCM MT-GP and independent GPs consistently generated reliable 95% Confidence Intervals (CIs). This observation highlighted the effectiveness of probabilistic GP frameworks in accurately quantifying uncertainties, as a substantial proportion of PV values fell within these intervals.

However, it is essential to acknowledge the competitiveness of the statistical benchmarks, especially in short-term predictions. Notably, the VAR model, which leverages inter-task correlations, outperformed other benchmarks, including the LCM MT-GP. Its exceptional performance in immediate forecasting could be attributed to its ability to closely fit localized patterns. Nevertheless, as the forecast horizon extended, the VAR model's effectiveness deprecated. In contrast, GP models functioned as smoother approximators, prioritizing the expansion of uncertainty and mitigating abrupt fluctuations. Consequently, GPs demonstrated greater robustness in quantifying predictive uncertainties, especially in settings with extended forecast horizons, addressing the challenge of fluctuating PV values effectively.

When temporal input was coupled with weather parameters, GP models predictably

exhibited enhanced performance, evident in reduced Mean Absolute Error (MAE) and Negative Log Predictive Density (NLPD) scores. Furthermore, GP models outperformed their machine learning (ML) benchmarks when trained using the same weekly scheme. It is crucial to recognize that these ML benchmarks have the capability to leverage larger training datasets, potentially enhancing their comparative performance. Nonetheless, the Bayesian Ridge Regression (BRR) model, a linear model, demonstrated remarkable competitiveness, hinting at a potential linear relationship between weather parameters and PV output.

Interestingly, the degree of improvement in the Hadamard LCM MT-GP, compared to independent GPs, diminished when weather parameters were integrated. Despite consistently observing lower NLPD values for the Hadamard LCM MT-GP in both 2- and 6-hour nowcasts, a smaller proportion of PV values fell within the 95% CI for the 6-hour nowcasts compared to independent GPs. This observation raises questions regarding the significance of inter-task learning when weather parameters are introduced, suggesting that the inclusion of weather conditions from neighboring stations may not substantially enhance predictive performance in this context. Further research is needed to identify scenarios where inter-task learning remains valuable.

To conclude, this research contributes valuable insights into the potential benefits of LCM MT-GPs for PV nowcasting, the integration of weather parameters, and the intricate balance between inter-task learning and temporal modeling. While statistical benchmarks excel in short-term predictions, Gaussian Process models offer a robust framework for quantifying predictive uncertainties for the dynamic and fluctuating PV generation. Notably, the inclusion of weather parameters within the Gaussian Process framework significantly enhances predictive performance, particularly when contrasted with relying solely on temporal input. Weather parameters offer crucial insights into local conditions that directly influence PV generation, solidifying their contribution to improved predictive accuracy for PV nowcasting.

5.1 Future Work

To advance the field of PV nowcasting, we advise future research to address the technical and practical challenges concerning this project.

First, we want to iterate that a machine learning model is only as good as its data. Our reliance on Numerical Weather Predictions (NWPs) with one-hour intervals, combined with a five-minute temporal resolution for PV values, results in substantial temporal gaps. These gaps introduce uncertainty, as interpolation becomes necessary to align the two data sources. Future studies should address this uncertainty by investigating the accuracy of interpolation techniques. We would suggest an assessment of the uncertainty associated with NWPs and incorporating it into the modeling process. One approach is to assign probability distributions to input data and sample from these distributions during training to reflect the inherent uncertainty in the dataset. Additionally, the evolution of uncertainty in NWPs as forecast horizons extend merits investigation, as this aspect was not thoroughly explored in this project.

Efficient computation is critical for practical PV nowcasting. The reliance on weekly folds for training, driven by computational limitations, can lead to reduced confidence in the model's generalization ability to unseen data. Thus, it is essential to explore methods for enhancing computational efficiency and optimizing the training process. Leveraging high-performance computing resources, such as GPUs, and developing strategies for managing training data across multiple folds without data loss is worth pursuing. Developing techniques for rapid model re-calibration and prediction is crucial to harness the full potential of GP models in PV nowcasting. An expansion of the state-space model addressed in [6] with the inclusion of weather parameters is an avenue of exploration to how the computational burden can be reduced. This would offer faster retraining and adaptation capabilities, ensuring that the model stay synchronized with evolving weather conditions. In the ablation, we also highlighted the potential of Approximate Latent Force Models

(ALFMs), which if implemented successfully, could cope with limited training data.

While the results of this study demonstrate the potential of inter-task learning with LCM MT-GPs, its practical value remains questionable, especially when incorporating weather variables. The relatively small improvements observed, coupled with increased model complexity and training times, raise questions about the cost-effectiveness of inter-task learning in this context.

Lastly, future research should address the role of temporal components in PV nowcasting, particularly in the presence of weather parameters. Investigating how the temporal component influence model performance and whether they complement or degrade predictive accuracy can help deciding their importance.

In summary, these future directions aim to make contributions to the advancement of PV nowcasting, particularly within the GP framework. The hope being that GPs probabilistic framework can play a pivotal role in streamlining integration of PV energy to the National Grid and optimizing grid management practices by providing decision-making support. We noted in Section 1.3 that this could help reduce CO2 emissions by a staggering 100,000 tonnes in the UK alone. Future testing of our approaches would need to be investigated on how much improvement can be gained on this front.

References

[1] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General perception with iterative attention," *CoRR*, vol. abs/2103.03206, 2021.

[2] OCF, "Solar pv nowcasting using deep learning." https://drive.google.com/file/d/1sDKZ8WEJlTNa5oyonbNl2xGyZ7GLXKtQ/view, December 31, 2021. Accessed on 2023-08-18.

[3] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PLoS ONE*, vol. 13, 2018.

[4] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. M. de Pison, and F. Antonanzas-Torres, "Review of photovoltaic power forecasting," *Solar Energy*, vol. 136, pp. 78–111, 2016.

[5] R. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. Australia: OTexts, 2nd ed., 2018.

[6] S. Nassimiha, P. Dudfield, J. Kelly, M. P. Deisenroth, and S. Takao, "Short-term prediction and filtering of solar power using state-space gaussian processes," 2023.

[7] NEA, "Timeline of the energy crisis." https://www.nea.org.uk/energy-crisis/energy-crisis-timeline/, Aug 2023. Accessed on 2023-08-17.

[8] J. Ward and J. Burke, "The uk emission trading system." https://lifedicetproject.eui.eu/2022/05/30/the-uks-emission-trading-system-delinking-challenges-after-brexit/, May 2022. Accessed on 2023-08-17.

[9] SEFE.Energy, "A brief history of the uk energy crisis." https://www.sefe-energy.co.uk/blog/

a-brief-history-of-the-uk-energy-crisis-and-what-to-expect-next/, Feb 2023. Accessed on 2023-08-17.

[10] IEA, "Natural gas markets." https://www.iea.org/news/ Accessed on 2023-08-17.

[11] NGESO, "Eso homepage." https://www.nationalgrideso.com, 2023. Accessed on 2023-08-17.

[12] NGESO, "Balancing reserve." https://www.nationalgrideso.com/industry-information/balancing-services/reserve-services/balancing-reserve, 2023. Accessed on 2023-08-17.

[13] NGESO, "What does electricity national control center do." https://www.nationalgrideso.com/what-we-do/electricity-national-control-centre/what-does-electricity-national-control-centre-do, 2023. Accessed on 2023-08-17.

[14] D. Travers and R. Tipton, "Nowcasting: How ocf will reduce carbon emissions with solar forecasts." https://www.openclimatefix.org/post/nowcasting-how-ocf-will-reduce-carbon-emissions-with-solar-forecasts, Nov, 2022. Accessed on 2023-08-18.

[15] K. J. Iheanetu, "Solar photovoltaic power forecasting: A review," *Sustainability*, vol. 14, no. 24, 2022.

[16] A. Dolara, F. Grimaccia, S. Leva, M. Mussetta, and E. Ogliari, "A physical hybrid artificial neural network for short term forecasting of pv plant power output," *Energies*, vol. 8, no. 2, pp. 1138–1153, 2015.

[17] H. Sheng, J. Xiao, Y. Cheng, Q. Ni, and S. Wang, "Short-term solar power forecasting based on weighted gaussian process regression," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 1, pp. 300–308, 2018.

[18] M. Kudo, A. Takeuchi, Y. Nozaki, H. Endo, and J. Sumita, "Forecasting electric power generation in a photovoltaic power system for an energy network," *Electrical Engineering in Japan*, vol. 167, no. 4, pp. 16–23, 2009.

[19] Z. Zhen, F. Wang, Y. Sun, Z. Mi, C. Liu, B. Wang, and J. Lu, "Svm based cloud classification model using total sky images for pv power forecasting," in *2015 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, 2015.

[20] M. Abdel-Nasser and K. Mahmoud, "Accurate photovoltaic power forecasting models using deep lstm-rnn," *Neural Computing and Applications*, vol. 31, no. 7, pp. 2727–2740, 2019.

[21] A. D. Plasencia Lotufo, A. El hendouzi, and A. Bourouhou, "Solar photovoltaic power forecasting," *Journal of Electrical and Computer Engineering*, vol. 2020, p. 8819925, 2020.

[22] K. Krasucka, "Six months into the nowcasting project: Our results are highly promising." https://openclimatefix.org/post/six-months-into-the-nowcasting-project-our-results-are-highly-promising, March, 2022. Accessed on 2023-08-18.

[23] OCF, "Uk photovoltaic data." https://huggingface.co/datasets/openclimatefix/uk_pv, 2023. Accessed on 2023-08-18.

[24] UKV, "Ukv weather charts and data." https://www.netweather.tv/charts-and-data/ukv, 2023. Accessed on 2023-08-18.

[25] EUMETSAT, "Eumetsat rapid scanning service." https://www.eumetsat.int/rapid-scanning-service, 2023. Accessed on 2023-08-18.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[27] Y. A. Lawati, J. Kelly, and D. Stowell, "Short-term prediction of photovoltaic power generation using gaussian process regression," *CoRR*, vol. abs/2010.02275, 2020.

[28] Meteomatics, "What is a numerical weather model?." https://www.meteomatics.com/en/weather-model-europe/. Accessed: 2023-18-08.

[29] M. Titsias, "Variational learning of inducing variables in sparse gaussian processes," in *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics* (D. van Dyk and M. Welling, eds.), vol. 5 of *Proceedings of Machine Learning Research*, (Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA), pp. 567–574, PMLR, 16–18 Apr 2009.

[30] S. Makridakis and M. Hibon, "The m3-competition: results, conclusions and implications," *International Journal of Forecasting*, vol. 16, no. 4, pp. 451–476, 2000. The M3- Competition.

[31] C. Rasmussen, O. Bousquet, U. Luxburg, and G. Ratsch, "Gaussian processes in machine learning," *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tubingen, Germany, August 4 - 16, 2003, Revised Lectures, 63-71 (2004)*, vol. 3176, 09 2004.

[32] M. P. Deisenroth, Y. Luo, and M. Wilk. https://infallible-thompson-49de36.netlify.app, Dec. 3, 2020. Accessed: 2023-20-08.

[33] M. Abramowitz, *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables,*. USA: Dover Publications, Inc., 1974.

[34] D. Duvenaud, H. Nickisch, and C. E. Rasmussen, "Additive gaussian processes," 2011.

[35] E. V. Bonilla, K. Chai, and C. Williams, "Multi-task gaussian process prediction," vol. 20, 2007.

[36] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chemistry*, vol. 36, pp. 1627–1639, jan 1964.

[37] J. Baxter, "A model of inductive bias learning," *CoRR*, vol. abs/1106.0245, 2011.

[38] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, pp. 41–75, 1997.

[39] H. Liu, K. Wu, Y.-S. Ong, C. Bian, X. Jiang, and X. Wang, "Learning multitask gaussian process over heterogeneous input domains," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–13, 2023.

[40] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, pp. 859–877, apr 2017.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[42] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," 2019.

[43] A. A. Ahmadi, "Monte carlo integration." https://www.cs.princeton.edu/courses/archive/fall11/cos323/notes/cos323_f11_lecture15_monte.pdf, 2011. Accessed: 2023-23-08.

[44] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *CoRR*, vol. abs/1603.02754, 2016.

[45] D. Wipf and S. Nagarajan, "A new view of automatic relevance determination," vol. 20, 2007.

[46] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," *CoRR*, vol. abs/1806.07366, 2018.

[47] M. Heinonen, C. Yildiz, H. Mannerstrom, J. Intosalmi, and H. Lahdesmaki, "Learning unknown ode models with gaussian processes," 2018.

[48] M. Alvarez, D. Luengo, and N. D. Lawrence, "Latent force models," in *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics* (D. van Dyk and M. Welling, eds.), vol. 5 of *Proceedings of Machine Learning Research*, (Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA), pp. 9–16, PMLR, 16–18 Apr 2009.

[49] J. Moss, F. L. Opolka, B. Dumitrascu, and P. Lió, "Approximate latent force model inference," *CoRR*, vol. abs/2109.11851, 2021.

[50] E. B. Iversen, J. M. Morales, J. K. Moeller, and H. Madsen, "Probabilistic forecasts of solar irradiance by stochastic differential equations," 2013.

[51] I. La, S. S. Yum, I. Gultepe, J. M. Yeom, J. I. Song, and J. W. Cha, "Influence of quasi-periodic oscillation of atmospheric variables on radiation fog over a mountainous region of korea," *Atmosphere*, vol. 11, no. 3, 2020.

A | Source Code

Source code for all of the methods implemented for the project can be found in the GitHub repository:
https://github.com/eirikbaekkelund/Dissertation.

B | Predictions - LCM MT-GP Models

In the following figures, we present predictions generated by both the Kroenecker- and Hadamard LCM MT-GPs for 2-hour nowcasts. The blue line represents the median of the predictive distribution for the training data, accompanied by the corresponding 95% Confidence Interval (CI) denoted by the shaded blue region. To distinguish between the training and testing phases, a vertical dotted line marks the train/test split. In the testing phase, the red line signifies the median predictions for the test data, also accompanied by a 95% CI represented by the red shaded region.

It is evident from the training data that as PV values exhibit higher fluctuations, the models tend to produce smoother approximations. This behavior carries over to the models' predictions, which is particularly noticeable in Figure B.2. Furthermore, we observe that the Kroenecker LCM MT-GP may encounter challenges when historical periodicity does not align with the test data, as illustrated in Figure B.3. In such cases, the Hadamard LCM MT-GP benefits from the contextual information provided by the weather data.

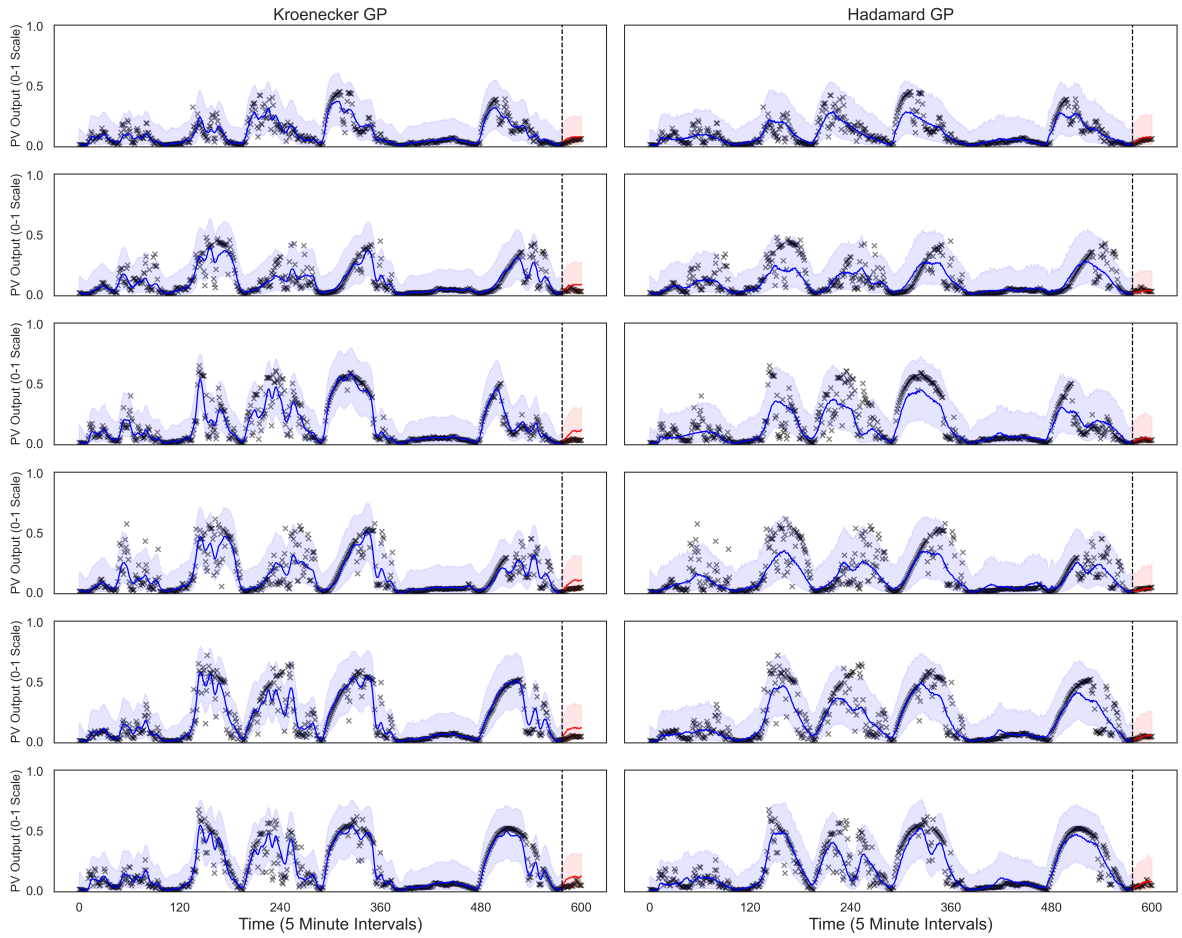


Figure B.1: Predictions from a sample week during winter.

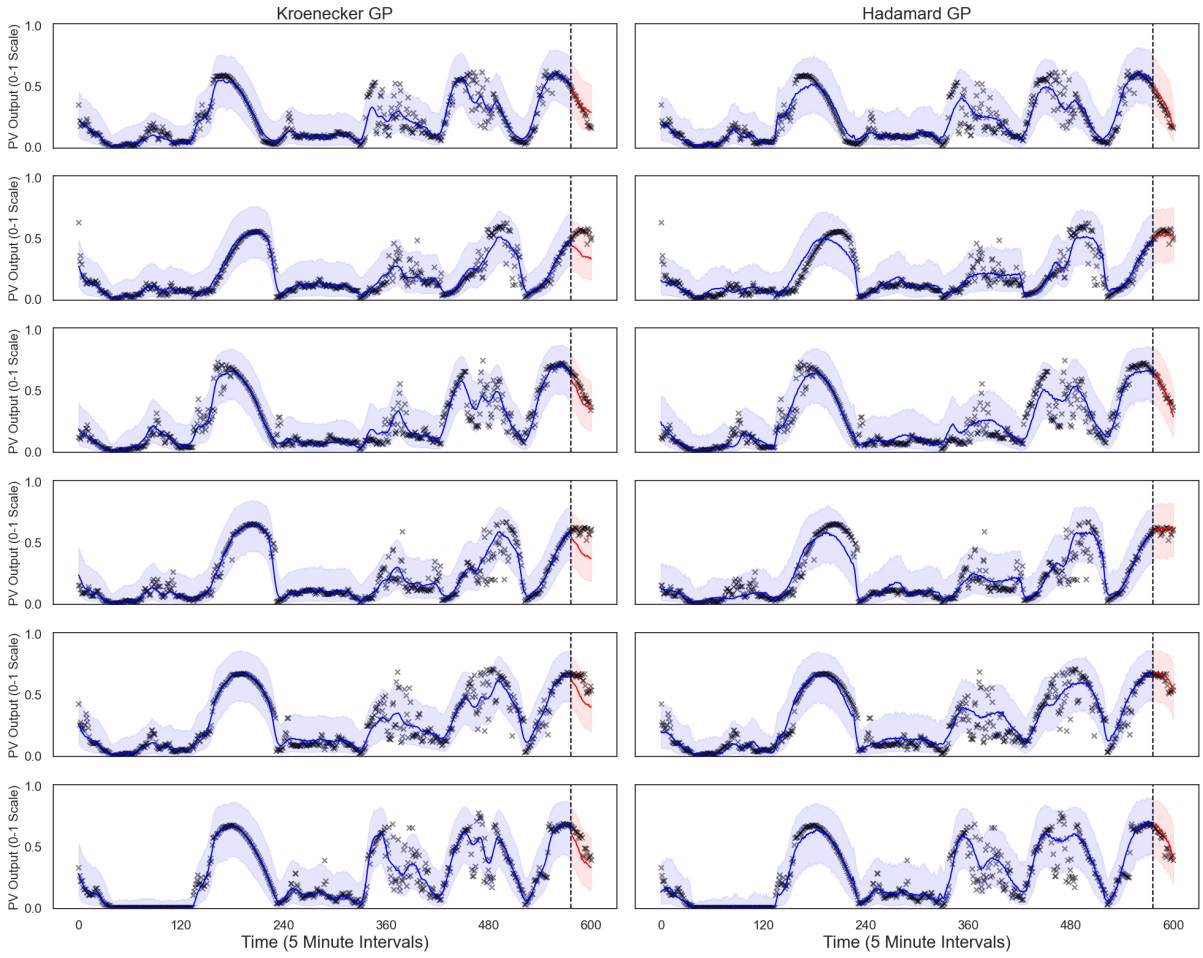


Figure B.2: Predictions from a sample week during spring.

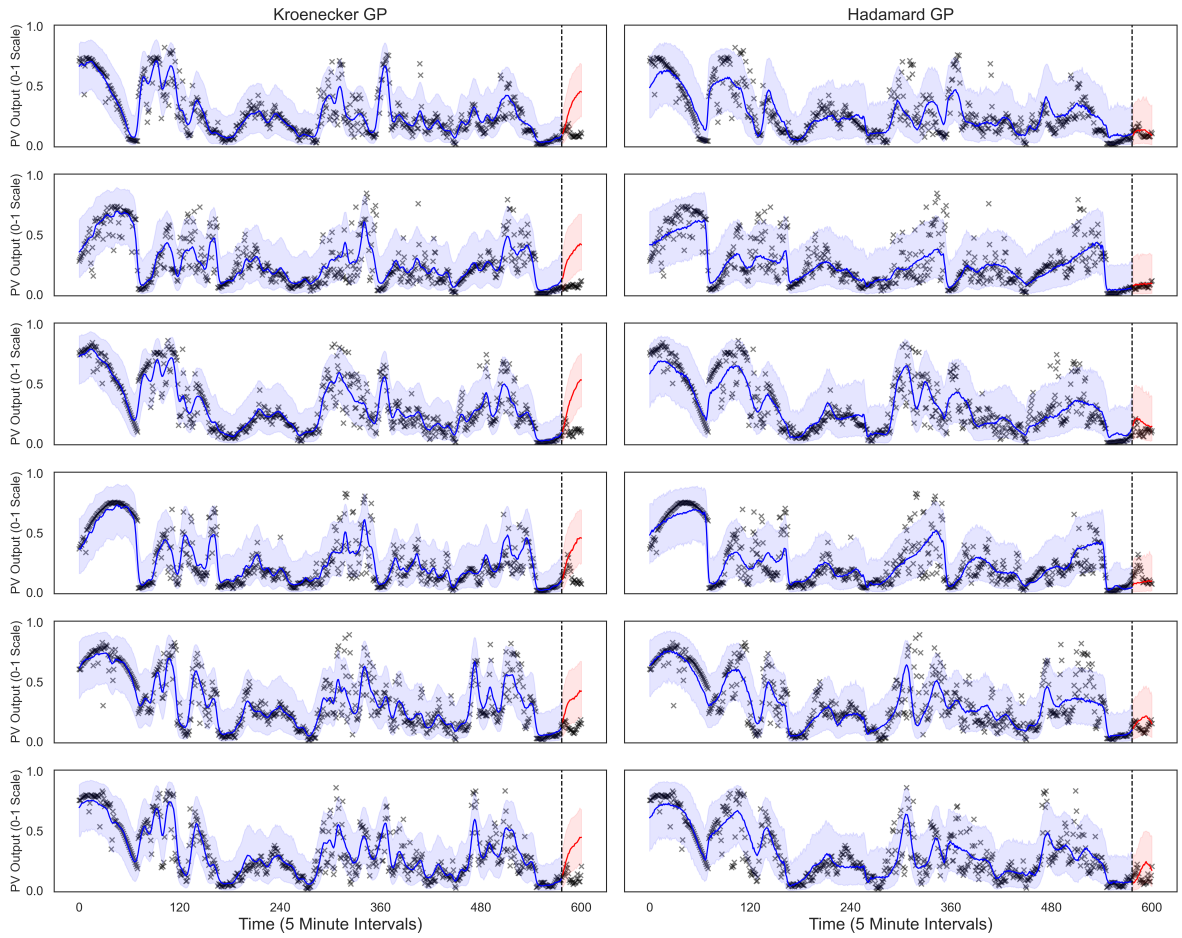


Figure B.3: Predictions from a sample week during summer. Here, the Kroenecker LCM MT-GP suffers from not having the included weather parameters when the test data differs from the historic PV data.

Figure B.4 shows us a sample forecast for the Hadamard LCM MT-GP when forecasting 6 hours ahead. We can see that the SE kernel smooths the oscillating PV values with satisfactory confidence intervals. The labeling remains the same as in the figures above.

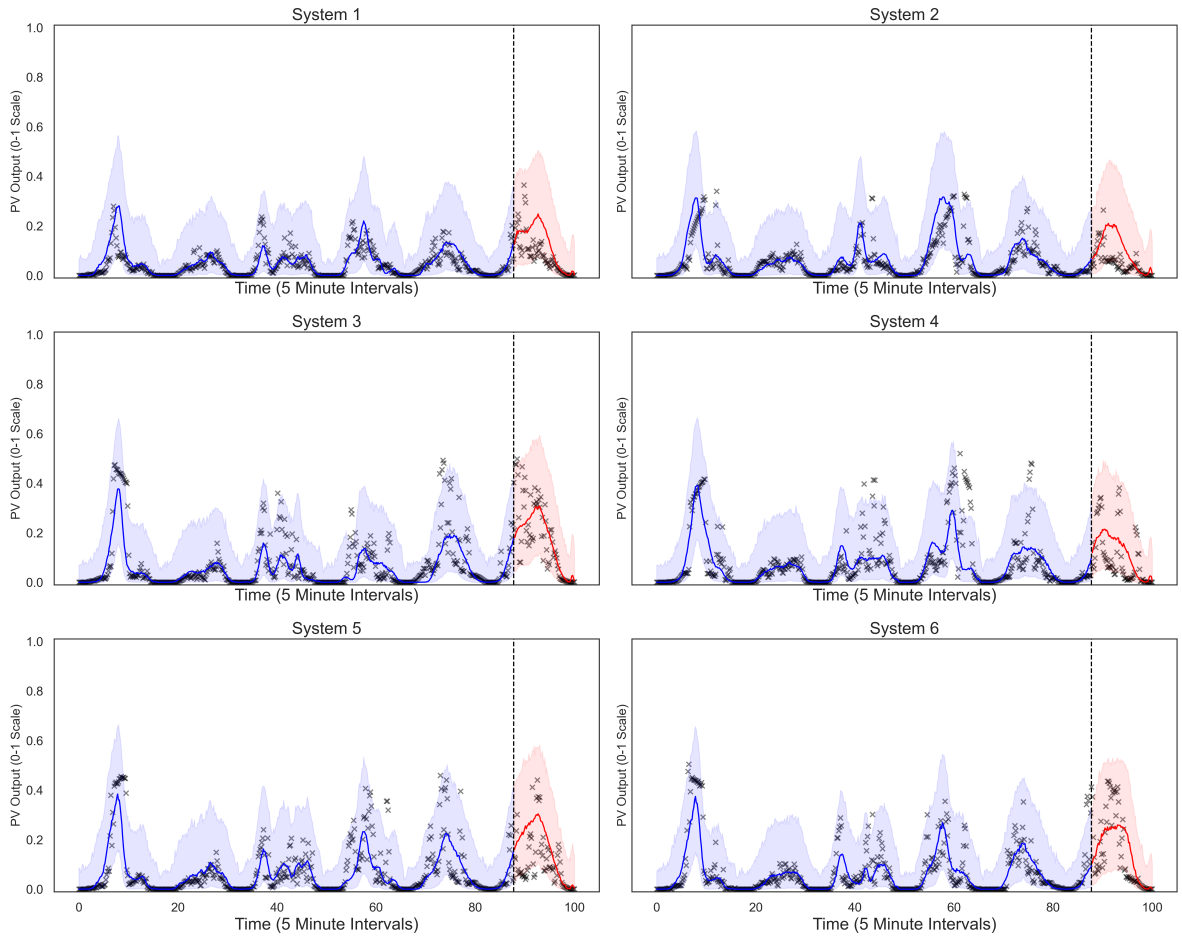


Figure B.4: A sample week for 6 hour nowcasts using the Hadamard LCM MT-GP.

C | Evidence Lower Bound

Following the definition of the KL divergence given in Section 3.2.5, we derive the expression for the ELBO:

C.1 Without Inducing Points

$$
\begin{aligned}
& D_{\mathrm{KL}}[q_\lambda(\mathbf{f})||p(\mathbf{f}|\mathbf{y})] \\
&= \int q_\lambda(\mathbf{f}) \log \frac{q_\lambda(\mathbf{f})}{p(\mathbf{f}|\mathbf{y})}\, d\mathbf{f} \\
&= \int q_\lambda(\mathbf{f}) \left[\log q_\lambda(\mathbf{f}) - \log p(\mathbf{f}|\mathbf{y})\right]\, d\mathbf{f} \\
&= \int q_\lambda(\mathbf{f}) \left[\log q_\lambda(\mathbf{f}) - \log p(\mathbf{f}) - \log p(\mathbf{y}|\mathbf{f}) + \log p(\mathbf{y})\right]\, d\mathbf{f} \\
&= \int q_\lambda(\mathbf{f}) \frac{q_\lambda(\mathbf{f})}{p(\mathbf{f})}\, d\mathbf{f} - \int q_\lambda(\mathbf{f}) \log p(\mathbf{y}|\mathbf{f})\, d\mathbf{f} + \log p(\mathbf{y}) \\
&= D_{\mathrm{KL}}[q_\lambda(\mathbf{f})||p(\mathbf{f})] - \int q_\lambda(\mathbf{f}) \log p(\mathbf{y}|\mathbf{f})\, d\mathbf{f} + \log p(\mathbf{y})
\end{aligned}
$$

where $p(\mathbf{f}|\mathbf{y})$ denotes the posterior distribution of latent function values $\mathbf{f}$ given observed data $\mathbf{y}$. Similarly, $p(\mathbf{y}|\mathbf{f})$ represents the likelihood under the observation of our latent values $\mathbf{f}$. Lastly, $p(\mathbf{y})$ denotes the model evidence, which is the overall likelihood of the data across the latent function values. For the sake of streamlining notations, we opted to omit explicit reference to the input $\mathbf{X}$ and additional model parameters θ. The above derivation establishes a fundamental lower bound to the model evidence:

$$
\begin{aligned}
\log p(\mathbf{y}) &= \int q_\lambda(\mathbf{f}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} - D_{\mathrm{KL}}[q_\lambda(\mathbf{f})||p(\mathbf{f})] + D_{\mathrm{KL}}[q_\lambda(\mathbf{f})||p(\mathbf{f}|\mathbf{y})] \\
\log p(\mathbf{y}) &\geq \int q_\lambda(\mathbf{f}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} - D_{\mathrm{KL}}[q_\lambda(\mathbf{f})||p(\mathbf{f})]
\end{aligned}
$$

Assuming independence, we can factorize the likelihood, such that:

$$\mathcal{L}_{\text{ELBO}} = \sum_{i=1}^{N} \int q_\lambda(f_i)[\log p(y_i|f_i)]df_i - D_{\text{KL}}[q_\lambda(\mathbf{f})||p(\mathbf{f})] \tag{C.1}$$

$$\approx \sum_{\mathbf{X},\mathbf{y}} \mathbb{E}_{q(f(\mathbf{x}))}[\log p(y_i|f_i)] - D_{\text{KL}}[q_\lambda(\mathbf{f})||p(\mathbf{f})]. \tag{C.2}$$

C.2 With Inducing Points

When dealing with large datasets in Gaussian Processes we are faced with computational complexity of $\mathcal{O}(N^3)$ and memory requirements of $\mathcal{O}(N^2)$. To address this challenge, we follow the work of Titsias [29] by using inducing points.

C.2.1 Defining Variational Distributions

First, we augment the joint distribution $p(\mathbf{y}, \mathbf{f})$ with auxiliary variables u such that our joint becomes

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}, \mathbf{u}) \tag{C.3}$$

where $\mathbf{u} = \{u(z_i)\}_{i=1}^{M}$ is a set of inducing variables, which are latent function values giving the inducing point locations contained within the matrix $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_M] \in \mathbb{R}^{M \times D}$. Now, the joint prior distribution over our latent GPs and the inducing points $\mathbf{u}$ is:

$$p(\mathbf{f}, \mathbf{u}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{f} \\ \mathbf{u}\end{bmatrix} \middle| \begin{bmatrix}\mathbf{0} \\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{K_{ff}} & \mathbf{K_{uf}^T} \\ \mathbf{K_{uf}} & \mathbf{K_{uu}}\end{bmatrix}\right) \tag{C.4}$$

Assuming the joint prior factorizes as $p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u})p(\mathbf{u})$ we can derive the marginal prior $p(\mathbf{u}) \sim \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K_{uu}})$. Equivalently, we can express it using GP notation over the inducing variable $u(\mathbf{z})$ as $p\left(u(\mathbf{z})\right) \sim \mathcal{GP}\left(\mathbf{0}, k_\theta(\mathbf{z}, \mathbf{z}')\right)$.

Now, we define the function $\boldsymbol{\psi} : \mathbb{R}^D \to \mathbb{R}^M$ by

$$\boldsymbol{\psi}(\mathbf{x}) \triangleq \mathbf{K}_{\mathbf{uu}}^{-1} k_\theta(\mathbf{Z}, \mathbf{x}) \tag{C.5}$$

where $k_\theta(\mathbf{Z}, \mathbf{x})$ denotes the covariance function between the inducing points $\mathbf{Z}$ and the input $\mathbf{x}$. Next, we define $\boldsymbol{\Psi} \in \mathbb{R}^{M \times N}$ to be the matrix of $\boldsymbol{\psi}$ applied row-wise on the input data $\mathbf{X} \in \mathbb{R}^{N \times D}$:

$$\boldsymbol{\Psi} \triangleq [\boldsymbol{\psi}(\mathbf{x_1}), \ldots \boldsymbol{\psi}(\mathbf{x_N})] = \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{K}_{\mathbf{uf}} \tag{C.6}$$

If we condition the derived prior distribution on the inducing variables, we can obtain the conditional distribution

$$p(\mathbf{f}|\mathbf{u}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{S}) \tag{C.7}$$

where the mean $\mathbf{m} = \boldsymbol{\Psi}^T \mathbf{u}$ and the covariance $\mathbf{S} = \mathbf{K}_{\mathbf{ff}} - \boldsymbol{\Psi}^T \mathbf{K}_{\mathbf{uu}} \boldsymbol{\Psi}$. We can note the second term in $\mathbf{S}$, which is known as the Nystrom approximation of $\mathbf{K}_{\mathbf{ff}}$ [35]. Again, this can be expressed in GP notation as:

$$p(\mathbf{f}(\mathbf{x})|\mathbf{u}) = \mathcal{GP}\left(\mathbf{m}(\mathbf{x}), \mathbf{S}(\mathbf{x}, \mathbf{x}')\right) \tag{C.8}$$

with mean function $\mathbf{m}(\mathbf{x}) = \psi(\mathbf{x})^T \mathbf{u}$ and covariance function $\mathbf{S}(\mathbf{x}, \mathbf{x}') = k_\theta(\mathbf{x}, \mathbf{x}') - \psi(\mathbf{x})^T \mathbf{K}_{\mathbf{uu}} \psi(\mathbf{x}')$. Similar to the preceding Section (C.1), we specify a variational distribution $q_\lambda(\mathbf{f}, \mathbf{u}) \triangleq p(\mathbf{f}|\mathbf{u}) q_\lambda(\mathbf{u})$, which we assume to be Gaussian such that $q_\lambda(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\mathbf{b}, \mathbf{W}\mathbf{W}^T\right)$ having variational parameters $\lambda = \{\mathbf{b}, \mathbf{W}\}$. If we marginalize out $\mathbf{u}$ from the joint variational distribution, we obtain the marginal variational distribution with respect to $\mathbf{f}$:

$$q_\lambda(\mathbf{f}) = \int q_\lambda(\mathbf{f}, \mathbf{u}) d\mathbf{u} = \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{C.9}$$

where using Gaussian identities we have the resulting mean function $\boldsymbol{\mu} = \boldsymbol{\Psi}^T \mathbf{b}$ and covariance function $\boldsymbol{\Sigma} = \mathbf{K}_{\mathbf{ff}} - \boldsymbol{\Psi}^T \left(\mathbf{K}_{\mathbf{uu}} - \mathbf{W}\mathbf{W}^T\right) \boldsymbol{\Psi}$.

C.2.2 Inference

As we have defined the variational distribution for the inducing points and our latent functions, we can aim to approximate the posterior distribution $p(\mathbf{f}, \mathbf{u}|\mathbf{y})$ through the variational distribution $q_\lambda(\mathbf{f}, \mathbf{u})$. As above (Section C.1), we resort to the minimization of the KL divergence between $q_\lambda(\mathbf{f}, \mathbf{u})$ and $p(\mathbf{f}, \mathbf{u}|\mathbf{y})$.

$$D_{\mathrm{KL}}[q_\lambda(\mathbf{f}, \mathbf{u})||p(\mathbf{f}, \mathbf{u}|\mathbf{y})] \tag{C.10}$$

$$= \iint q_\lambda(\mathbf{f}, \mathbf{u}) \log \frac{q_\lambda(\mathbf{f}, \mathbf{u})}{p(\mathbf{f}, \mathbf{u}|\mathbf{y})} \, d\mathbf{u} d\mathbf{f} \tag{C.11}$$

$$= \iint q_\lambda(\mathbf{f}, \mathbf{u}) \log \frac{q_\lambda(\mathbf{f}, \mathbf{u}) p(\mathbf{y})}{p(\mathbf{f}, \mathbf{u}, \mathbf{y})} \, d\mathbf{u} d\mathbf{f} \tag{C.12}$$

$$= \iint q_\lambda(\mathbf{f}, \mathbf{u}) \log p(\mathbf{y}) d\mathbf{u} d\mathbf{f} + \iint q_\lambda(\mathbf{f}, \mathbf{u}) \log \frac{q_\lambda(\mathbf{f}, \mathbf{u}) p(\mathbf{y})}{p(\mathbf{f}, \mathbf{u}, \mathbf{y})} \, d\mathbf{u} d\mathbf{f} \tag{C.13}$$

$$= \log p(\mathbf{y}) + \iint q_\lambda(\mathbf{f}, \mathbf{u}) \log \frac{p(\mathbf{f}|\mathbf{u}) q_\lambda(\mathbf{u})}{p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\mathbf{u}) p(\mathbf{u})} \, d\mathbf{u} d\mathbf{f} \tag{C.14}$$

$$= \log p(\mathbf{y}) + \int q_\lambda(\mathbf{u}) \frac{q_\lambda(\mathbf{u})}{\phi(\mathbf{f}, \mathbf{u}) p(\mathbf{u})} \, d\mathbf{u} \tag{C.15}$$

$$= \log p(\mathbf{y}) + D_{\mathrm{KL}}(q_\lambda(\mathbf{u})||\phi(\mathbf{f}, \mathbf{u}) p(\mathbf{u})) \tag{C.16}$$

where in Equation B.14 we make use of the definition $q_\lambda(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u}) q_\lambda(\mathbf{u})$ and observe that the terms $p(\mathbf{f}|\mathbf{u})$ cancel. Additionally, in B.15 we define

$$\phi(\mathbf{f}, \mathbf{u}) \triangleq \exp\left(\int \log p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\mathbf{u}) p(\mathbf{u}) d\mathbf{f} \right) \tag{C.17}$$

We take the ELBO to be defined by $-D_{\mathrm{KL}}(q_\lambda(\mathbf{u})||\phi(\mathbf{f},\mathbf{u})p(\mathbf{u}))$. Then, assuming a non-Gaussian likelihood we have:

$$\mathcal{L}_{\mathrm{ELBO}} = \int \log \frac{\phi(\mathbf{y},\mathbf{u})p(\mathbf{u})}{q_\lambda(\mathbf{u})} q_\lambda(\mathbf{u})\, d\mathbf{u} \tag{C.18}$$

$$= \int \left(\log \phi(\mathbf{y},\mathbf{u}) + \log \frac{p(\mathbf{u})}{q_\lambda(\mathbf{u})} \right) q_\lambda(\mathbf{u})\, d\mathbf{u} \tag{C.19}$$

$$= \mathbb{E}_{q_\lambda(\mathbf{u})} \left[\log \phi(\mathbf{y},\mathbf{u}) \right] - D_{\mathrm{KL}}(q_\lambda(\mathbf{u})||p(\mathbf{u})) \tag{C.20}$$

where the left term at the bottom is our expected log-likelihood and the right term is the KL divergence between our prior latent function and our variational distribution. To see how the expected likelihood term is derived, we can expand our function ϕ:

$$\begin{aligned}
&\mathbb{E}_{q_\lambda(\mathbf{u})} \left[\log \phi(\mathbf{y},\mathbf{u}) \right] \\
&= \int \log \phi(\mathbf{y},\mathbf{u}) q_\lambda(\mathbf{u}) d\mathbf{u} \\
&= \int \left(\int \log p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\mathbf{u}) d\mathbf{f} \right) q_\lambda(\mathbf{u}) d\mathbf{u} \\
&= \int \log p(\mathbf{y}|\mathbf{f}) \left(\int p(\mathbf{f}|\mathbf{u}) q_\lambda(\mathbf{u}) d\mathbf{u} \right) d\mathbf{f} \\
&= \int \log p(\mathbf{y}|\mathbf{f}) q_\lambda(\mathbf{f}) d\mathbf{f} \\
&= \mathbb{E}_{q(\mathbf{f})} \left[\log p(\mathbf{y}|\mathbf{f}) \right]
\end{aligned}$$

Thereby, we get an objective function similar to Appendix C.1, now including inducing points:

$$\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_{q_\lambda(\mathbf{f})} \left[\log p(\mathbf{y}|\mathbf{f}) \right] - D_{\mathrm{KL}}(q_\lambda(\mathbf{u})||p(\mathbf{u})) \tag{C.21}$$