

Advancing Gaussian Process Modelling for Structural Health Monitoring



A Thesis submitted to the University of Sheffield
for the degree of Doctor of Philosophy in the Faculty of Engineering

by

J. H. Mclean

Department of Mechanical Engineering

University of Sheffield

July 2026

ACKNOWLEDGEMENTS

This thesis could not have been completed without the friendship, wisdom and patience of many. Although I cannot thank everyone here, I would like to recognize a few whose support made this thesis possible.

I would first like to thank my supervisors Dr Tim Rogers and Dr Nikolaos Dervilis. Tim, thank you for your continued belief in me since that early summer research project. Your mentorship, from pair coding JAX and discussing EP maths to countless hours of chess, has taught me more than I can express. Nikos, your constant positivity in our meetings made the difficult moments much easier to navigate.

Completing this PhD in an environment as supportive, knowledgable, and fun as the DRG has been an incredible privilege. My collaborators Matty and Brandon, along with all the researchers I engaged with in technical discussions, deserve heartfelt thanks. I am especially grateful to Dan, Adam, and Ewan, my fellow PhD comrades and friends. Apologies for the plentiful distractions of chess, diplomacy and idle chatter, being in the PhD trenches was far more bearable alongside you. The impromptu DRG chess club also deserves a mention; I am unreasonably proud of our communal 2K ELO achievement on Dan's account.

I must thank and acknowledge my family, without whose support and love I would not be where or who I am today.

Finally, to Shae. Just like with everything else, you are my rock. You make it all worth it, and I couldn't have done this without you.

Highlights:

- Bounded GPs are introduced and investigated as a way of enforcing physically meaningful predictions in SHM contexts
- A power curve from an active offshore wind turbine is used as a motivating example, representing significant novelty as one of the first works to bound predictive models for the problem
- A heteroscedastic GP with a Beta likelihood is investigated and found to have strong prediction power and higher uncertainty quantification than a normal GP would
- A warping method is used but found to be inadequate as a bounding method due to inducing discontinuity in the warped space
- GP Classification is explored as a underutilised tool in SHM
- Binary and Multi-class GPCs are benchmarked on SHM problems
- Power Expectation Propagation is introduced as the approximation tool of choice for GPCs
- A generalised formulation of PEP is introduced to show how to extend PEP to chained GPs (including multi-classification)
- Active learning is introduced as a way to deal with the problem of expensive labels
- The Expected Prediction Information Gain (EPIG) algorithm is introduced to SHM, and is shown to be an empirical improvement over current methods when using physically informed targets
- A novel method of updating GPCs is introduced to avoid expensive retraining at each active learning step, enabling broader online deployment of GPCs

ABSTRACT

Structural Health Monitoring (SHM) is the intelligent monitoring of any type of structure to enable proactive maintenance and extend service life. An active area of research, the field faces significant challenges including noisy and uncertain processes and limited model interpretability. Gaussian Processes (GPs) offer a compelling framework to address these challenges, yet their practical application in SHM lags behind theoretical capabilities. This thesis bridges this gap by advancing GP methodology in SHM, combining cutting-edge adaptations from the literature with novel developments tailored to SHM challenges.

While GPs provide principled uncertainty quantification and flexibility, they do not inherently enforce physical constraints, leading to physically impossible predictions. This work investigates Bounded GPs as a weakly physics-informed solution. Using in-service SHM data, the work demonstrates that bounding via a beta-likelihood significantly improves uncertainty quantification over baseline models.

Classification provides a robust way to handle the discrete labelling of structural states required by SHM. This thesis investigates GP Classification (GPC) for SHM, developing binary models via the Power Expectation Propagation (PEP) approximation and introducing a novel chained PEP-GP. It is shown that they outperform similar methods while providing uncertainty quantification across a range of SHM datasets.

To mitigate the reliance of GPCs on labeled datasets, the thesis explores active learning. Expected Predicted Information Gain (EPIG) is introduced as an acquisition function with physically informed target distributions. This method is shown to

leverage physical knowledge to converge faster than traditional functions, reducing labeling costs.

Finally, to accommodate online monitoring environments where standard GPs are computationally prohibitive, the thesis derives a generalized extension to the online PEP framework. This allows for the use of all likelihoods in online settings, improving upon previous variational methods.

Collectively, this thesis creates a modernized toolbox of GP methods better suited to the requirements of SHM.

Contents

Acknowledgements	i
1 Introduction	1
1.1 Structural Health Monitoring	1
1.1.1 Incorporating physics	4
1.1.2 Uncertainty quantification	6
1.1.3 The way forward	7
1.2 Bayesian Inference	8
1.3 Gaussian Processes	13
1.3.1 Sparse Gaussian Processes	17
1.3.2 Further Limitations of GPs	18
1.3.3 Gaussian Processes in SHM	19
1.4 Contributions of this thesis	21
2 Bounded Gaussian Processes in SHM	24
2.1 Bounding approaches and ancillary theory	28
2.1.1 Heteroscedastic noise	28
2.1.2 Warping functions	29
2.1.3 Beta likelihood	30
2.1.4 Approximations and Quadrature solutions	33
2.2 Case study	34
2.2.1 Simple GP Regression	36
2.2.2 Heteroscedastic warped Gaussian process	38
2.2.3 Beta Likelihood	40
2.3 Discussion	42
3 Power Expectation Propagation and Gaussian Process Classifiers in SHM	44

3.1	Binary Gaussian Process Classification	48
3.2	Expectation Propagation	50
3.2.1	EP for Bayesian posteriors	56
3.2.2	Power EP for Sparse Gaussian Processes	58
3.3	Application of Binary GPC in SHM	61
3.4	PEP for multi-latent GPs	66
3.4.1	Application of multi-class GPC in SHM	71
3.5	Discussion	74
4	An exploration of Active learning: the power of prior belief and fast model updating	77
4.1	Do BALD men dream of Electric PIGs (EPIG)?	81
4.1.1	Information theoretic acquisition	82
4.1.2	EPIG SHM case study	84
4.2	Fast updating for Active Learning via generalised online learning . . .	87
4.2.1	Sparse online PEP updating	90
4.2.2	Sparse Streaming case study	95
4.3	Discussion	98
5	Conclusions and future work	100
5.1	Bounded Gaussian Processes for SHM	101
5.2	Gaussian Process Classification for SHM	102
5.3	Targetted Active Learning for SHM	103
5.4	Generalised Online PEP Updates for SHM	104
5.5	Future Work	105
	Appendix	107
A	PEP derivation and implementation	107
A.1	Full rank factors	108
A.2	Rank 1 factors	109
A.3	Full implementation	111
A.4	Marginal likelihood optimisation	112
B	Chained PEP implementation	114
C	Streaming PEP derivation	116

C.1	$q_1(b)$	116
C.2	Full rank streaming PEP	118
Bibliography		119

INTRODUCTION

“Matrix is not positive semi-definite”

- jax.numpy

It has been abundantly clear for some time now that engineering practice is undergoing a profound transformation. Ever-increasing computing power is driving this shift, enabling new and more efficient approaches to engineering. This is made evident in tools such as computational aided design and finite element methods; once cutting edge computational technologies that are now industry standard. Machine Learning (ML) is one of the latest phases of this transformation, enabling practitioners to learn complex behaviours and relationships directly from measured data, rather than relying solely on classical physical or numerical models that may not fully capture reality. This thesis forms a small part of the ongoing scientific body of work underpinning this shift.

1.1 Structural Health Monitoring

Structural Health Monitoring (SHM) is the automated monitoring of any type of structure, with the aim to enable proactive maintenance and extend operational service life. Significant benefits can be realised if structural damage can be detected and monitored in its early stages while it is cheaper and simpler to repair. This

reduces downtime from critical failures or emergency repairs, reduces operational cost by allowing for proactive rather than scheduled and emergency maintenance and improves the safety and resilience of the structure. Advancing reliable SHM methods is thus a practical and high-impact research goal.

SHM is typically described as a hierarchy of tasks, often referred to as Rytter's hierarchy [1], later modernised in Worden et al. [2]. Each task builds on the previous one:

1. **Detection:** Is there damage present?
2. **Localisation:** Where might the damage be?
3. **Classification:** What type of damage is there?
4. **Quantification:** What is the extent of the damage?
5. **Prognosis:** What is the remaining operational life of the structure?

At their core, each task requires some model of the underlying structure in order to achieve their goal. Specifically, that model must be over measurable physical phenomena that are sensitive to the target outcome. For example, the detection stage aims to detect deviations from normal structural behaviour, and thus requires a model over measurable phenomena that are sensitive to damage, such as natural frequencies or power output [3, 4]. The localisation stage, in turn, requires models that can locate the position of damage, and thus will typically use phenomena sensitive to damage location, such as guided or ultrasonic waves [5, 6].

Traditional approaches revolve around analytical/first principle models or numerical models. These can be powerful, but the real world is uncertain and complex, and often these approaches cannot fully describe and capture the phenomena of interest. To address these limitations and to take advantage of the streams of sensor data modern structures can generate, more recent approaches make use of data-driven ML models. By learning patterns from measured structural behaviour, machine learning methods enable data-driven solutions to all stages of the SHM hierarchy that can be entirely divorced from any physical underpinning. A useful example here is the detection stage, where a wide body of literature use various ML algorithms

in novelty detection or outlier analysis in order to detect damage (e.g. [7–11]). In these cases, ML is used to learn normal operational behaviour for a damage sensitive feature. Deviation from this operational norm can thus be quickly identified and fed up the hierarchy of tasks.

While ML offers tremendous potential for the various tasks within SHM, direct adoption is not straightforward. Many state-of-the-art ML models function as opaque black boxes, providing accurate point predictions without quantifying their confidence in their predictions or revealing the reasoning behind them. Deployed, these algorithms will be subject to significant environmental and measurement noise and operational conditions far from their training datasets. This potential for unreliability is problematic: the goal of SHM is to automatically mobilise the resources necessary to prolong the useful life of the structure based on the prognoses of these algorithms. This mobilisation is an expenditure of capital, time and operational disruption purely on the authority of algorithms operating in uncertain and complex environments whose level of certainty and functionality is often opaque. When financial, operational and safety consequences hinge on algorithmic outcomes, confidence in their output is not optional.

It thus becomes clear that machine learning based SHM is an exercise in trust. It is an exercise in trust of predictive systems which are expected to predict phenomena subject to high degrees of measurement and environmental uncertainty and often poorly understood even in classical mechanics. Embedding such outputs into operational workflows, long-term asset management, or financial planning therefore requires clear justification that the system’s predictions are reliable. From this perspective, several desiderata emerge for ML models in SHM: they must be (somewhat) explainable, generalise well to unseen conditions, be robust to noise, quantify the uncertainty underlying their predictions, and inspire confidence that they will behave consistently under real-world conditions.

To satisfy these desiderata for ML models in SHM, this thesis explores two strategies: physics-informed modelling and robust uncertainty quantification.

1.1.1 Incorporating physics

The simplest approach is often the most effective: if a classical physical or numerical model can adequately describe the phenomena of interest, it should be used. However, in practice, real-world systems are rarely captured fully by such models, making it attractive to learn these “un-modelable” phenomena from measured data using data-driven ML. Yet, as established in the previous section, while data-driven ML can capture complex phenomena, their black-box nature raises serious concerns about reliability and trust. Further, data-driven models typically require large amounts of data in order to reliably capture the phenomena of interest. In an engineering context, such large data corpuses are unlikely to be available. This raises a natural question: could a combination of physical models and data-driven learning offer a more reliable solution?

It is useful here to introduce the black-white spectrum of models [12, 13], which characterises the mixture of data-driven learning and explicit physical modelling. At one extreme, a purely black-box ML model relies entirely on data, with no incorporation of physical knowledge. At the other extreme, a white-box model is grounded entirely in physical principles and cannot deviate from them. A grey-box model is thus one that incorporates some level of physical structure or governing laws into a machine learning model, alternatively known as Physically Informed Machine Learning (PIML). The hope of this union is that the data-driven component can learn behaviours that are difficult or impossible to model using traditional white-box methods, while the incorporation of physical principles ensures convergence of the learning to physically plausible solutions.

The adoption of grey-box models can address all of the desiderata not related to uncertainty. The incorporation of physics or physical insight allows grey-box models to fall back on encoded information, allowing for better extrapolation outside of the training distribution and better predictions when confronted with low data regimes [14, 15]. Further, grey-box models provide a rationale grounded in known governing laws for their predictions, thereby reducing (though not eliminating) opacity and improving confidence in their consistency and predictions. It is important to note that no grey-box method completely eliminates model opacity, as all current approaches still rely, at least in part, on a black-box data-driven component.

It is useful here to briefly examine the wider PIML literature. To do this, the taxonomy laid out in Karniadakis et al. [16, 17] is used. This is a wide and active field of research, and the reader is referred to [16, 18] for comprehensive review papers. The literature surrounding Gaussian Process PIML is left for a more in-depth review in subsection 1.3.3.

Inductive bias involves embedding physical principles directly into the model architecture or representation, so that the learned function inherently respects the structure of the underlying physics. Many physical systems exhibit inherent symmetries, such as invariance under rotations, translations, or parity transformations. Rigid body dynamics are symmetric under rotations, described by the group $SO(3)$, while fluid flows are typically symmetric under both translations and rotations. Equivariant neural architectures explicitly encode these symmetries, preventing the model from relearning identical physics in multiple forms. Examples include equivariant attention mechanisms for 3D rotations and translations [19] and networks embedding Lorentz-group equivariance for relativistic systems [20]. Similarly, graph neural networks (GNNs) leverage the connectivity and locality structure inherent in mesh-based physical systems, providing an architecture that respects spatial relationships and locality constraints [21, 22]. Operator learning architectures, such as the Galerkin Transformer [23], can also be interpreted as embedding inductive biases by representing inputs in a functional basis.

Learning bias is the direct incorporation of weak physics into the training objective or loss function. Physics-Informed Neural Networks (PINNs) are a canonical example, incorporating the residual of the governing PDE into the loss function [15]. This can be understood as a minimisation toward the optima that best fits both the data and the PDE, rather than a PDE solver. Extensions of this idea include Lagrangian and Hamiltonian PINNs, which embed the Euler-Lagrange or Hamiltonian residuals into the loss function [24, 25]. Bayesian PINNs additionally enable principled uncertainty quantification in the inferred solutions and parameters [26]. Equation discovery methods, such as Sparse Identification of Nonlinear Dynamics (SINDy) [27] and its Bayesian extension BINDy [28], and those based on genetic algorithms [29] also fall into this category, where the learning objective searches over a dictionary of candidate terms to identify parsimonious governing equations.

1.1.2 Uncertainty quantification

As mentioned previously, many conventional machine-learning models are deterministic; they return only a point prediction and offer no indication of how confident that prediction is. In SHM applications, where data are often subject to measurement and environmental noise and operating conditions often diverge from training data, this neglect of uncertainty can be problematic. A model may produce seemingly confident predictions while relying on noisy training data or operating far outside its training distribution. Without an explicit and robust measure of confidence, engineers cannot reliably base decisions on the model's outputs. Consequently, the ability to quantify predictive uncertainty is essential for the safe and effective deployment of machine learning in SHM. This is known as Uncertainty Quantification (UQ). Before discussing the benefits of UQ, it is necessary to clarify what uncertainty means. In the literature, it is commonly decomposed into two types:

- Aleatoric uncertainty: irreducible uncertainty arising from intrinsic randomness or measurement noise in the system. More data does not eliminate it; it can only be characterised (e.g. via a noise model).
- Epistemic uncertainty: reducible uncertainty arising from lack of knowledge, poor coverage of the input space, or limited model capacity. It can, in principle, be reduced with better modelling or more informative data.

A useful UQ ML model should handle both: it should express what is fundamentally unknown in the world (aleatoric) and also when the model does not know because it has insufficient information (epistemic).

The literature surrounding UQ is extensive, and the reader is referred to [30, 31] for comprehensive reviews on the field. Broadly, UQ can be divided into Bayesian and non-Bayesian approaches. Bayesian methods leverage Bayes' theorem to provide a principled framework for quantifying both epistemic and aleatoric uncertainties [32, 33]. While theoretically robust, scaling Bayesian methods to large models is a challenge that recent techniques increasingly overcome. Non-Bayesian approaches, in contrast, offer practical alternatives that can be easier to implement and scale, which are often model agnostic. One class of non-Bayesian UQ is known as conformal

prediction [34, 35], which provides distribution-free prediction intervals with statistically valid coverage. Monte Carlo (MC) dropout [36, 37] approximates Bayesian inference by performing multiple stochastic forward passes through a network with dropout enabled at test time, capturing epistemic uncertainty without explicit posterior estimation. Ensemble methods [38, 39], including deep ensembles, quantify uncertainty through the variance of predictions from independently trained models. Finally, quantile regression [40, 41] estimates conditional quantiles of the target, providing intervals that reflect aleatoric uncertainty.

Beyond improving confidence in a models' predictions, uncertainty-aware approaches are also better suited to noisy data. By explicitly accounting for variability and measurement noise, UQ-based methods tend to remain more stable and accurate under uncertain sensing conditions than purely frequentist point estimators [42, 43]. Moreover, the quantified uncertainty itself is a valuable output; it is mathematically useful in a variety of tasks, as will be shown in chapter 4, and it can be propagated downstream to enable and enrich other tasks such as risk-aware decision making. For example, Hughes et al. [44] integrate UQ into a probabilistic, risk-based SHM framework, while Kamariotis et al. [45] introduce an uncertainty-aware decision analysis framework.

1.1.3 The way forward

The need for a grey-box machine learning framework capable of uncertainty quantification is therefore evident. Gaussian Processes (GPs) are a promising class of models, innately capable of both requirements. Their ability to handle nonlinear regression tasks makes them especially useful in cases where physical models are difficult or impossible to define. As a result, GPs have seen significant application in SHM.

However, as shall be shown shortly, application of GPs to SHM lags the latest theoretical advances from the statistical and machine learning community. This thesis aims to bridge this gap by introducing recent developments and exploring avenues of application not yet covered in SHM literature. Further, novel GP advancements are derived to address specific engineering requirements, with implications that extend to the wider ML community. While the methods in this thesis are motivated as tools for the early stages of Rytter's hierarchy, they are broadly applicable across

engineering applications of machine learning.

1.2 Bayesian Inference

Although there are several ways to interpret or formulate Gaussian Processes, the Bayesian view is the most common. Thus before introducing Gaussian Processes, it is necessary to briefly cover Bayesian statistics and inference. Bayes theorem (Equation 1.1) arises naturally out of the laws of probability, and is used to update beliefs in light of new evidence. Here, $p(A)$ is the prior information on A , it describes the current information or belief about A . $p(B|A)$ is the likelihood, which describes how well A currently describes B . $p(A|B)$ is known as the posterior, which can intuitively be thought of as the updated belief on A having now observed B .

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \quad (1.1)$$

Bayes' theorem can be used to infer probability distributions over unknown model parameters or latent functions, a process known as Bayesian inference. Bayesian inference also offers multiple natural ways to incorporate physical knowledge or constraints into the learning procedure.

To introduce Bayesian inference practically and illustrate how the Bayesian framework lends itself to incorporating physics, consider a simple SHM flavoured example: a cantilever beam subject to varying and cyclic static point loads F at its tip (Figure 1.1). Over time, damage accumulates, reducing the beam's elastic stiffness k_s . In order to detect this, k_s will be monitored via measurements of input force F and beam displacement δ , as in the ideal case the static displacement for a cantilever is $\delta = \frac{F}{k_s}$. However, the displacement measurements δ are noisy, making direct calculation unreliable. The problem can thus be formulated as $\delta = \frac{F}{k_s} + \epsilon$, where ϵ represents the measurement noise.

A reasonable first stab at this problem would be to apply Ordinary Least Squares linear regression (OLS). A slight reformulation $\delta = \beta F + \epsilon$, where $\beta = \frac{1}{k_s}$ shows that the relationship between δ and F is linear, suggesting linear regression is a natural choice.

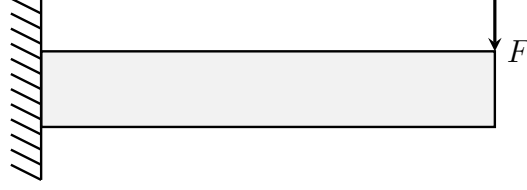


Figure 1.1: Cantilever beam with the downward point load F at the free end.

Linear regression aims to choose β to minimise the sum of squared residuals $\frac{1}{n} \sum_i (\delta_i - \beta F_i)^2$, where n is the number of data points, in order to find the best-fit slope β for the data. This has a closed form solution:

$$\beta = \frac{\sum_{i=1}^n F_i \delta_i}{\sum_{i=1}^n F_i^2}. \quad (1.2)$$

Once the best fit β is found, the model can be used to make predictions for any δ given F . The results of the linear regression applied to this problem are shown in Figure 1.2. This is a purely data driven approach, although it is not a black-box model since the solutions for OLS are closed form and well known.

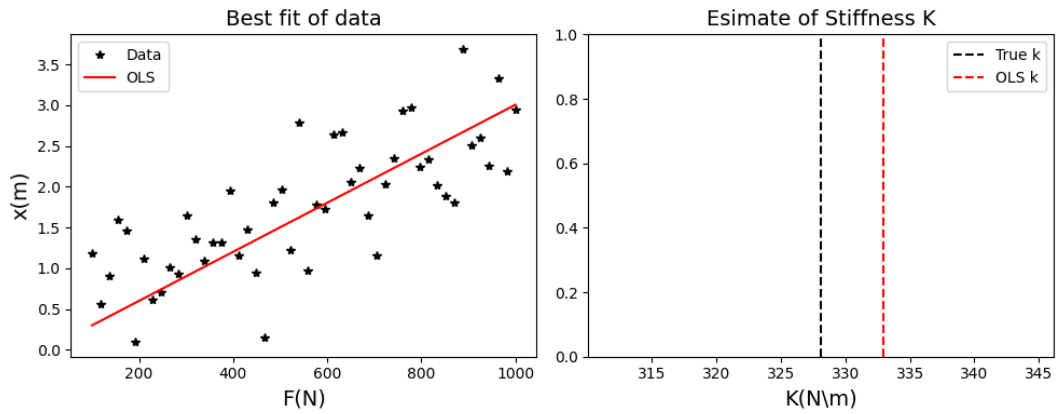


Figure 1.2: Linear regression estimate of k_s

Extending linear regression into the Bayesian realm is straightforward. Instead of estimating a single point value for the slope β , the goal is to obtain a posterior distribution over β , as shown in Equation 1.3. Here a more general notation is introduced as it will be used heavily throughout the thesis. Inputs are denoted $\mathbf{X}$ (which may be multidimensional) and outputs by $\mathbf{y}$, while unseen or test data will

be represented as $\mathbf{X}_*$ and $\mathbf{y}_*$, respectively. Notation in bold implies a vector or matrix. For this example, the input features $\mathbf{X}$ are the measurements of $\mathbf{F}$, while the output features $\mathbf{y}$ are the noisy measurements δ . This posterior enables UQ over $\boldsymbol{\beta}$, naturally incorporates prior knowledge about the parameter, and, with standard Gaussian prior and likelihood, converges to LR.

$$p(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta})p(\boldsymbol{\beta}) \quad (1.3)$$

Since the observation noise ϵ is assumed Gaussian, the likelihood is given as $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}) = \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 I)$, where σ^2 is the noise variance. The likelihood term describes how well a particular choice of weights describes the observed (training) data. The prior $p(\boldsymbol{\beta})$ is used to encode prior belief about $\boldsymbol{\beta}$. It is useful to ensure that the prior is conjugate to the likelihood, as it ensures the posterior will have the same functional form and be simpler to calculate. If there are no beliefs to encode, a wide or general prior is typically chosen. Here, a wide Gaussian prior is chosen $p(\boldsymbol{\beta}) = \mathcal{N}(0, \tau^2 I)$, where τ is the prior precision. Note that τ acts as a regularisation term; shrinking overly large weights and penalising overly complex solutions. This is mathematically the same as L2 regularisation, which is typically applied in an ad-hoc manner in deterministic ML. Elegant and innate regularisation is one of the key advantages of Bayesian inference. When prior and likelihood are Gaussian, so too is the posterior, meaning Equation 1.3 has a closed form solution as the product of two Gaussians. Since the likelihood and prior terms are Gaussian, the posterior is also Gaussian and can be calculated in closed form as the product of two Gaussians:

$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}} = \left(\frac{1}{\sigma^2} \mathbf{X}^\top \mathbf{X} + \frac{1}{\tau^2} \mathbf{I} \right)^{-1} \quad (1.4)$$

$$\boldsymbol{\mu}_{\boldsymbol{\beta}} = \frac{1}{\sigma^2} \boldsymbol{\Sigma}_{\boldsymbol{\beta}} \mathbf{X}^\top \mathbf{y} \quad (1.5)$$

$$p(\boldsymbol{\beta} | \mathbf{X}, \mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}) \quad (1.6)$$

It is important to reinforce the point that this posterior only exists in closed form because the prior is conjugate to the posterior. Bayesian posteriors are often not available in closed form. To deal with this, sampling methods such as MCMC [46] or

approximations such as variational approximations [47, 48] are necessary. This will be expanded upon further in section 1.3 and throughout the thesis where applicable.

The goal of this example is the posterior over β , so in this case it is sufficient to stop at the equations for the posterior above. However, often practitioners will want to make predictions of unseen data $\mathbf{y}_*$ using the posterior over the parameters. To do this, β is marginalised out of the joint distribution $p(\mathbf{y}_*, \mathbf{y}, \beta)$. This averaging over all possible weights is a robust way of dealing with uncertainty in β while returning a distribution over $\mathbf{y}_*$.

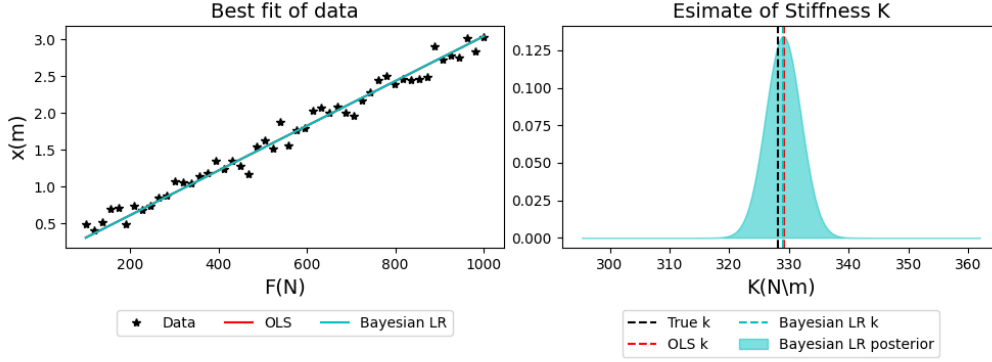
$$p(\mathbf{y}_* | \mathbf{X}, \mathbf{y}, \mathbf{x}_*) = \int p(\mathbf{y}_* | \mathbf{X}_*, \beta) p(\beta | \mathbf{X}, \mathbf{y}) d\beta \quad (1.7)$$

The results for the Bayesian linear regression are shown in Figure 1.3a, where it is clear that the BLR has converged on the LR, but now provides quantified uncertainty about how sure it is of its estimate. The utility of the UQ is highlighted in Figure 1.3b, where the observation noise ϵ has been significantly increased. It is clear that given such noisy observations, the BLR is very unsure about its estimate of β . This UQ is rich, useful information that can be used to inform any engineering decision made on this result, and would be hidden if only the OLS model was used.

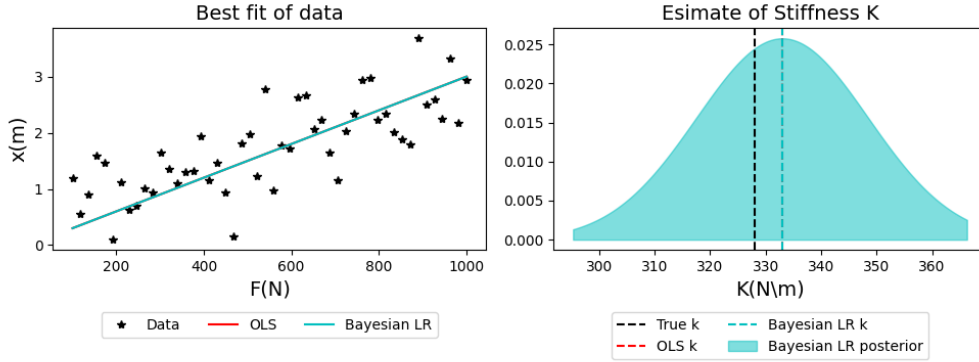
But $p(\beta) = \mathcal{N}(0, \tau^2 I)$ is an uninformative prior. An engineer can make an educated guess that k_s is probably going to be close to $k_s = \frac{3EI}{L^3}$, where E is the (undamaged) Young's modulus of the beam material, I is the second moment of the area and L is the length of the beam. This can be easily incorporated into the prior as:

$$p(\beta) = \mathcal{N}\left(\frac{3EI}{L^3}, \tau^2 I\right) \quad (1.8)$$

While Equation 1.4 assumes a zero-mean prior, introducing a non-zero prior mean (μ_0) updates the posterior mean to $\mu_\beta = \Sigma_\beta \left(\frac{1}{\sigma^2} \mathbf{X}^\top \mathbf{y} + \frac{1}{\tau^2} \mu_0 \right)$, leaving the covariance unchanged. This can be thought of as a very simple example of physics informed machine learning. Here, the model is still entirely learnt from the data, but a sensible and physically informed initial guess is provided. The results of this incorporation are shown in Figure 1.4, where it is clear that the physics informed guess returns a stronger prediction, and that the model is more certain of this prediction. It is



(a) Standard prior



(b) Noisier data

Figure 1.3: Bayesian linear regression estimate of k_s under varying observation noise. Recall that the (here Gaussian) Bayesian LR posterior is given here as $\mathcal{N}(\mu_\beta, \Sigma_\beta)$, as shown in Equation 1.4.

important to note here that the prior can work in reverse too, a bad prior can require much more data to overcome, and significantly bias the posterior away from the truth.

This has been a quick introduction to Bayesian ML via a simple linear regression example. For a rigorous introduction to Bayesian analysis, the reader is referred to [49, 50]. It has been show how Bayesian ML provides innate and powerful UQ, modelling both aleatoric and epistemic uncertainty. Epistemic via the returned uncertainty distribution over the object of interest, and aleatoric via the inclusion of a noise term in the likelihood, which captures everything that cannot be captured from the data. Further, Bayesian ML provides a strong and flexible framework through which physics can be incorporated. The following section introduces Gaussian Processes, a high flexible Bayesian model and the focus of this thesis.

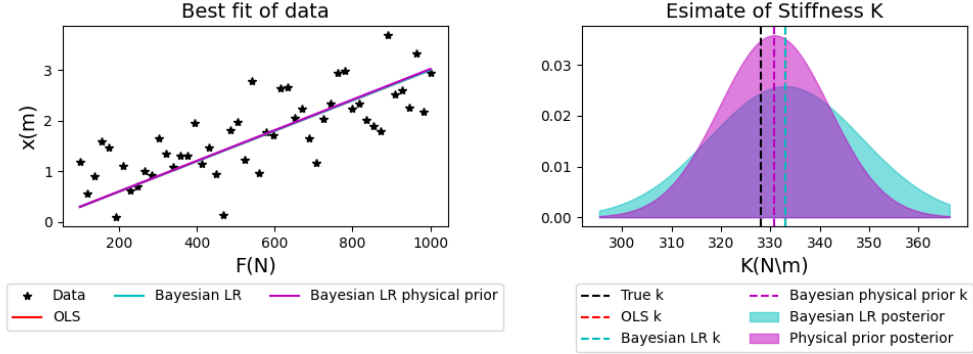


Figure 1.4: Physical prior BLR estimation of k_s . The posterior is still a closed form Gaussian: $\mathcal{N}(\mu_\beta, \Sigma_\beta)$.

1.3 Gaussian Processes

As mentioned previously, there are many ways in which to define and interpret a Gaussian Process [50]. A natural view is to consider the GP as a probability distribution over the possible functions that could fit a given set of points. This can be intuitively understood as a type of Bayesian inference, where the target of interest is f , a continuous latent function modelling the underlying process being observed. However, to perform inference on a given dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, this continuous function is evaluated at specific training inputs $\mathbf{X}$. This yields $\mathbf{f}$, a finite vector of scalar latent function values. Applying Bayes' theorem to this finite vector results in the formulation shown in Equation 1.9. This section will introduce naive GP regression, where the observations $\mathbf{y}$ are assumed to be corrupted by independent Gaussian noise, such that $y_i = f(\mathbf{x}_i) + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$.

$$p(\mathbf{f}|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f}, \mathbf{X})p(\mathbf{f}, \mathbf{X})}{p(\mathbf{y}|\mathbf{X})} \quad (1.9)$$

Where $p(\mathbf{f}|\mathbf{X}, \mathbf{y})$ is the posterior over the latent function of interest given the data. $p(\mathbf{y}|\mathbf{f}, \mathbf{X})$ is the likelihood, which here describe how well the latent functions currently describe the data. In the regression case, this likelihood is: $p(y_i|f, x_i, \theta) = \mathcal{N}(y_i; f(x_i), \sigma^2)$ (i.e. $y_i = f(x_i) + \epsilon_i$), where θ are the hyperparameters of the GP, and shall be fully examined shortly. Finally, $p(\mathbf{f}, \mathbf{X}) = GP(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}})$ is the prior, which induces a distribution over functions and encodes prior belief about the form

of these functions.

The prior takes the form $p(\mathbf{f}, \mathbf{X}) = \mathcal{N}(0, \mathbf{K}_{\mathbf{xx}})$, where $\mathbf{K}_{\mathbf{xx}}$ is the covariance matrix, otherwise known as the kernel. The kernel specifies the family of functions that may have generated the data, and is one of the strengths of the GP. Kernels allow users to encode belief about the family of underlying functions such as periodicity, smoothness or linearity, and can be combined in order to better fit the problem. As will be shown shortly, kernels can also be used to encode physical information into the GP. A common kernel function is the squared exponential, given in Equation 1.10, where σ and l are the variance and length scale respectively. These are known as hyperparameters (commonly denoted θ), and their choice strongly defines the kernel. Figure 1.5a shows function draws from a GP with an SE kernel. An important note here is that the mean of the prior is often left as 0 for mathematical convenience, but this need not be true, and can also be used to encode prior belief or information about a problem as illustrated in the previous section.

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{(\mathbf{x} - \mathbf{x}')^2}{2l^2}\right) \quad (1.10)$$

Calculating the posterior over $\mathbf{f}$ will proceed as normal in Bayesian tasks; the prior is updated with training data via the likelihood. Of course, draws from the posterior at locations in the training data is rarely the goal. Instead, the task is to evaluate the posterior over $\mathbf{f}$ at test locations $\mathbf{X}_*$ (which have related observations $y_* = f(\mathbf{x}_*) + \varepsilon$).

The goal then is $p(\mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_*)$. The GP prior defines a joint model over all latent function values, giving rise to Equation 1.11, where $\mathbf{K}_{\mathbf{xx}_*}$ is the covariance between $\mathbf{x}$ and $\mathbf{x}_*$ and $\mathbf{K}_{\mathbf{x}_*\mathbf{x}_*}$ is the covariance of the test data.

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_{\mathbf{xx}} + \sigma_n^2 I & K_{\mathbf{xx}_*} \\ K_{\mathbf{x}_*\mathbf{x}} & K_{\mathbf{x}_*\mathbf{x}_*} + \sigma_n^2 I \end{bmatrix}\right) \quad (1.11)$$

The conditioning of $\mathbf{f}_*$ on the training data is then simply the conditioning of a joint multivariate Gaussian distribution, giving the following:

$$p(\mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_*) = \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \quad (1.12)$$

$$\boldsymbol{\mu}_* = \mathbf{K}_{\mathbf{x}_* \mathbf{x}} (\mathbf{K}_{\mathbf{x} \mathbf{x}} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} \quad (1.13)$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{\mathbf{x}_* \mathbf{x}_*} - \mathbf{K}_{\mathbf{x}_* \mathbf{x}} (\mathbf{K}_{\mathbf{x} \mathbf{x}} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{x} \mathbf{x}_*} \quad (1.14)$$

This is equivalent to marginalising out $\mathbf{f}$ from the joint Gaussian $p(\mathbf{f}_*, \mathbf{f} | \mathbf{X}, \mathbf{y}, \mathbf{X}_*)$, shown in Equation 1.15. The parallels to Bayesian inference for linear regression in Equation 1.7 are clear. Indeed, many sources introduce GPs as an nonlinear, infinite dimensional extension of Bayesian linear regression.

$$p(\mathbf{f}_* | X, \mathbf{y}, x_*) = \int p(\mathbf{f}_* | \mathbf{f}, X, \mathbf{y}, x_*) p(\mathbf{f} | X, \mathbf{y}) d\mathbf{f} \quad (1.15)$$

Finally, prediction of the noisy test data $\mathbf{y}_*$ is found via marginalisation of $\mathbf{f}_*$ from the joint $p(\mathbf{y}_*, \mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_*)$:

$$p(\mathbf{y}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_*) = \int p(\mathbf{y}_*, \mathbf{f}_* | X, \mathbf{y}) d\mathbf{f}_* = \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_* + \sigma_n^2 \mathbf{I}) \quad (1.16)$$

To tie this all together, a brief visualisation of the GP in action is shown in Figure 1.5. Here, posterior samples (functions) are drawn from a posterior that progressively sees more data. It is clear that the drawn functions (and distribution over these functions) concentrate around the observed data.

It was mentioned previously that the choice of hyperparameters θ define the kernel. How then, to choose the best hyperparameters? A *true* Bayesian would of course have already placed a prior over θ , and would be furiously marginalising out the hyperparameters in the new predictive posterior. Unfortunately, the true Bayesian will quickly find themselves faced with intractable integrals and computationally expensive MCMC methods. Instead, practitioners often prefer to maximise the marginal likelihood w.r.t. θ , known as type-II maximum likelihood problem. The marginal likelihood is the denominator in Equation 1.9, and in the tractable naive GPR case is given by:

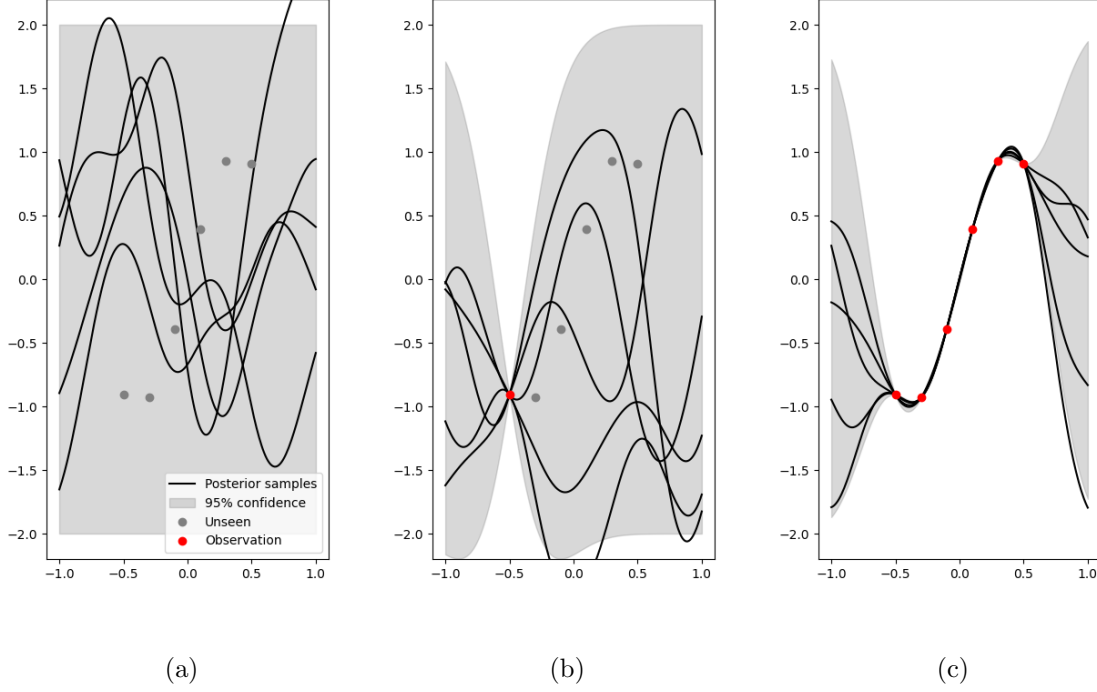


Figure 1.5: Draws from the prior in (a) and the posterior in (b) and (c) after seeing more and more data.

$$p(\mathbf{y}|\mathbf{X}, \theta) = \int p(\mathbf{y}|\mathbf{f}, \mathbf{X}, \theta)p(\mathbf{f}|\mathbf{X}, \theta)d\mathbf{f} \quad (1.17)$$

Thus the marginal likelihood integrates out $\mathbf{f}$, allowing for an evaluation of the total evidence for the hyperparameters given all possible functions consistent with the prior. It is also tractable in the regression case, and contains self regularising properties that make it an attractive optimisation target. Maximisation of the marginal likelihood is typically done via gradient descent, but any optimisation algorithm is suitable here. A gradient descent based optimisation of a GP looks like this: initialise the hyperparameters, take the partial derivative of the marginal likelihood w.r.t. all θ and take a step in that direction. Continue until convergence in θ or a set number of iterations.

Unfortunately, this requires an inversion of the covariance $\mathbf{K}_{\mathbf{xx}}$ at each step. Since inversion is an $\mathcal{O}(n^3)$ operation, this creates a significant computational bottleneck for scaling GPs to large datasets.

1.3.1 Sparse Gaussian Processes

The $\mathcal{O}(n^3)$ computational bottleneck is one of the most important practical limitations of GPs. Considerable research has therefore focused on various methods to mitigate this bottleneck, almost all of these induce some kind of sparsity into the GP in order to reduce the size of the required inversion without losing model flexibility.

The most widely used class of sparse GP models are known as sparse inducing point methods, and aim to summarise the latent function via a small set of $m \ll n$ inducing variables instead of the full data of size n . Inducing point frameworks vary in the details, but have this over-arching goal in common. FITC and PITC [51, 52] impose independence assumptions in the prior to achieve sparsity, while variational approaches induce sparsity through an approximate posterior [53]. Hensman et al. [54] show how variational approaches can be scaled to very large datasets via mini-batching. These inducing point approaches are unified under the Power Expectation Propagation (PEP) framework [55], which will be expanded upon in chapter 3.

Beyond inducing point methods, a variety of approaches have been proposed, each balancing flexibility, approximation accuracy, and computational efficiency in different ways. One class of methods relies on spectral approximations, such as the Sparse Spectrum GP introduced by Lázaro-Gredilla et al. [56], which approximate stationary kernels using a finite set of basis functions derived from the kernel’s spectral density. Similarly, reduced-rank representations in Hilbert spaces provide computational gains by projecting the GP onto a lower-dimensional basis [57]. Another strategy involves inter-domain approaches, where inducing variables are defined as functionals of the process rather than direct observations [58]. Local GPs offer a different angle by dividing the input space into regions and chaining together many smaller GPs, which can reduce computational complexity (via the cube law) while capturing local variations effectively [59, 60]. More recently, sparsifying kernel methods have emerged, which learn sparse structures in the kernel itself directly from data [61].

It is clear that the literature on sparse and computationally efficient GPs is extensive and well established. Many of these methods have been shown to work on very large datasets ($>1\text{mn}$ data points [54, 61]), and thus a practitioner is free to choose the method that best suits their use case. This thesis uses inducing point methods throughout, as they also enable approximation of non-Gaussian likelihoods. The

rationale for each choice is provided at the point of application throughout the work.

The inducing point methods covered in this thesis all require the computation of 1-dimensional non-tractable integrals of the form:

$$\int \mathcal{N}(\cdot)g(x)dx \quad (1.18)$$

where $\mathcal{N}$ is a Gaussian over x and $g(x)$ is some non-Gaussian function or distribution. This integral is therefore intractable, and approximation is required. Gauss-Hermite quadrature approximates integrals of the kind:

$$\int e^{-x^2}g(x)dx \approx \sum_{i=1}^N w_i, g(x_i), \quad (1.19)$$

where x_i are the roots of the N^{th} Hermite polynomial $H_N(x)$ and w_i are the associated quadrature weights. It is clear that via a simple change of variables, Gaussian-Hermite can approximate the 1-dimensional integral shown in Equation 1.18.

In practice, a small number of quadrature points (typically between 10 and 20) is often sufficient to approximate the integral to a high degree of accuracy, making Gauss-Hermite quadrature a practical and efficient method for approximating the intractable expectations that arise in sparse Gaussian process inference.

1.3.2 Further Limitations of GPs

Beyond the computational bottleneck of matrix inversion, GPs are faced with many practical limitations. First amongst these is the fact that the closed form solution to the posterior only exists when the likelihood is Gaussian. When it is not, an approximation is required. Many of the inducing point sparsification schemes discussed in the previous section are specifically designed to enable approximation of non-Gaussian likelihoods [54, 55, 62], making these methods a fundamental component of practical GP implementations. If accuracy is mission critical, the non-tractable posterior can be sampled via MCMC [46] or related sampling methods, although this is typically computationally expensive.

GPs may also struggle on non-tabular or structured domains (e.g., graphs or manifolds), unless the covariance is carefully engineered to reflect the geometry of the inputs. Graph GPs and message-passing extensions have emerged to address these cases [63], as have convolutional GPs to deal with high dimensional inputs such as images [64]. Similarly, non-stationary behaviour cannot be captured by traditional stationary kernels unless the kernel is adapted to allow spatially or temporally varying lengthscales; this has motivated non-stationary kernel families [65, 66].

Another important practical constraint is that standard GPs assume globally smooth latent functions. Highly non-smooth or discontinuous phenomena (e.g., change points [67, 68], regime shifts [69], strong heteroscedasticity [70]) require bespoke kernel design or hierarchical constructions. Further, GPs assume infinite support on the distributions of f , and as such cannot respect bounds in the prediction space. Bounded GPs have been developed to these constraints [71, 72], ensuring that outputs respect known limits.

As mentioned in the previous section, kernel choice is critical; an inappropriate kernel can severely hinder performance. Deep GPs [73], which learn flexible, hierarchical kernel structures, and Automatic kernel search [74] have been proposed to address this.

GP extrapolation is often weak unless strong prior structure is encoded into the kernel. GPs revert to the prior outside the support of the data. This is beneficial when the prior encodes the correct physics, but otherwise leads to conservative behaviour. Multi-output GPs can achieve better extrapolation by leveraging correlations between related outputs, as demonstrated by Jones et al. [75].

1.3.3 Gaussian Processes in SHM

As mentioned previously, GPs have seen wide adoption in SHM literature, likely due to the innate UQ combined with a powerful and flexible predictive framework. Naive black-box GPs are common, applied to various problems in SHM [76, 77]. More recently, it has become common to model heteroscedastic noise using an additional latent function [70, 78, 79]. These approaches remain data-driven and black-box in nature, capitalising on the robust UQ that GPs inherently provide. Many of these

applications will include variational or similar sparsification techniques in order to scale the GPs to the medium-large sized datasets found in SHM.

A substantial number of recent advances in GP PIML have been developed for and applied to SHM literature. A majority of these models are inductive bias approaches, leveraging the inherent flexibility of the Bayesian framework to incorporate physical knowledge directly into the learning process. Several methods achieve this by encoding physics through the kernel function, such as the single-degree-of-freedom (SDOF) kernel introduced by Cross et al. [13] and the switching kernel proposed by Pitchforth et al. [68]. Other approaches aim to constrain the GP so that its sample paths are guaranteed to satisfy a given set of physical principles or structural properties, such as the bounding kernels introduced in [80]. Broader theoretical work on GP bounding, such as the likelihood based bounds proposed by Jensen et al. [71], remains to be applied within the SHM domain. Additional inductive bias approaches combine GPs with classical system identification tools; for instance, GPs have been integrated into NARX models [81], and convolutional GPs have been employed to mimic the convolutional structure of linear dynamical systems [82].

Another class of inductive bias approaches involves state-space formulations of GPs. State-space GPs [83, 84], embed physical knowledge directly within the state-space representation. The Kalman filter can be interpreted as performing exact GP inference in a particular state-space representation, motivating direct adaptations of GP regression to filtering contexts [85, 86]. This connection extends naturally to latent force models [87, 88], where unknown forcing terms are also modelled as GPs.

The detection and localisation tasks in Rytter’s hierarchy introduced in section 1.1 are explicitly discrete classifications of structural states. The literature often tackles this with novelty or outlier methods based on regression techniques [7–11], but direct classification offers numerous advantages, including the ability to handle higher-dimensional, non-linear problems and increased robustness to noise. Consequently, Gaussian Process Classification (GPC) presents a promising tool for SHM. Despite this, the literature on GPC application in SHM is very limited, and existing works only examine binary GPCs [89].

Finally, GPs are a natural choice for prognostics because they provide trajectory predictions with calibrated uncertainty that can be propagated to Remaining Useful

Life estimates, as used in Vuckovic et al. [90]. Examples of such applications include GP mixtures to handle multimodal degradation in Li et al. [91] and deep nonstationary GPs in Xu et al. [92].

1.4 Contributions of this thesis

This thesis aims to develop an improved and modernised GP toolbox for the detection and localisation stages of SHM. The outcome of this is a robust, state-of-the-art UQ framework that can be readily extended to PIML when appropriate. This chapter has identified two key gaps in the application of theoretical GPs to SHM: the limited use of GP classification and the incomplete exploration of bounded GPs. This work addresses these gaps and examines the associated methodological and practical challenges. While only the bounded GP component is directly PIML, the enhanced GP methods developed here provide a strong probabilistic foundation for future GP PIML applications in SHM.

Chapter two explores bounded GPs in SHM. Since GPs are necessarily Gaussian distributions with infinite support, they cannot be bounded. This is a problem when using them to predict physically bounded things, such as power output or density. Physically implausible predictions or probability mass imperil downstream use and makes the immediate prediction less useful. In order to rectify this, this chapter applies two bounding methods from the theoretical GP community, both are novel to SHM. These are; beta likelihoods and heteroscedastic warping. It is shown that the Beta likelihood approach is highly effective and better models the uncertainty of the data. The work present in this chapter is published in the SHM journal under the name “Physically meaningful uncertainty quantification in probabilistic wind turbine power curve models as a damage-sensitive feature”. The author was the primary contributor of this publication.

Chapter three examines GP classification. As mentioned previously, detection and localisation are classification tasks of discrete structural states. Classification models are therefore interesting model architectures to explore for these tasks, however GPCs have seen little application to the field. This chapter introduces GPCs with the state-of-the-art approximation theorem, Power Expectation Propagation (PEP),

and benchmarks binary GPCs on a range of SHM tasks. It also introduces and similarly benchmarks multi-class GPCs, a novelty for SHM. In order to introduce the multi-class derivation for GPCs, it is shown how to derive PEP approximations for any chained GP, an important contribution to the PEP literature. The experiments find that in both the binary and multi class setting, GPCs offer similar or improved predictive performance compared to similar algorithms while also providing robust UQ.

Finally, chapter four tackles the problem of lack of data labelling in SHM. This is done via an investigation in active learning. Expected Predicted Information Gain (EPIG) is introduced as an algorithm of interest, since it explicitly targets improvements over specified target distributions. It is argued that since SHM has the benefit of often knowing where regions will be the most informative, this restriction on exploration can be worthwhile. EPIG is benchmarked against a range of acquisition functions, and is shown to outperform other algorithms in cases where target distributions are defined. Active learning requires constant updating of the predictive model, but this is very expensive for GPs. This chapter derives a novel extension to an online PEP framework for general GP likelihoods, enabling fast updates and online deployments. This generalised online PEP framework has implications not only for active learning, but wider deployment of online GPCs in SHM and the wider GP community. The generalised online framework is shown to not only have equivalent predictive performance to a full re-train, but also to be more performant than similar online algorithms for GP classification.

To conclude the introduction, it is appropriate to again reiterate the aim of this thesis and make explicit its specific objectives.

Aim

This thesis aims to develop an improved and modernised toolbox of Gaussian process methods for Structural Health Monitoring (SHM).

Objectives

To achieve this aim, the following objectives are pursued:

1. Advance the study of bounded GPs in SHM.
2. Introduce state-of-the-art GP classification techniques into SHM.

-
3. Address the challenge of data labelling costs for classification methods in SHM.
 4. Enable the online deployment of GP classifiers.

BOUNDED GAUSSIAN PROCESSES IN SHM

Chapter Highlights:

- Bounded GPs are introduced and investigated as a way of enforcing physically meaningful predictions.
- A power curve from an active offshore wind turbine is used as a motivating example, representing significant novelty as one of the first works to bound predictive models for the problem
- A heteroscedastic GP with a Beta likelihood is investigated and found to have strong prediction power and higher uncertainty quantification than a normal GP would
- A warping method is used but found to be inadequate as a bounding method due to inducing discontinuity in the warped space

A major challenge in applying GPs to SHM is that many SHM problems involve strict physical constraints on the outputs (e.g. negative density or power output are physically impossible). Standard GPs assume outputs follow a Gaussian distribution with infinite support, which means their predictions inevitably assign probability mass to regions that violate these constraints. This occurs even when the bounds

are well known and clearly represented in the data, leading to predictions that are physically meaningless. Physically implausible predictions reduce immediate utility and imperil downstream decision making involving the model such as risk based decision frameworks. This chapter explores how constraining GPs can lead to GP frameworks that retain their elegance and utility while becoming physically credible.

The motivating example used are wind turbine power curves, illustrated in Figure 2.1. Power curves are widely used in SHM literature as damage-sensitive features for wind turbines [93, 94], and their natural physical bounds make them especially suitable for this study. Further, as global demand for wind turbines continues to grow [95], so too does the need for effective methods of modelling and understanding these turbines, making it an important SHM contribution.

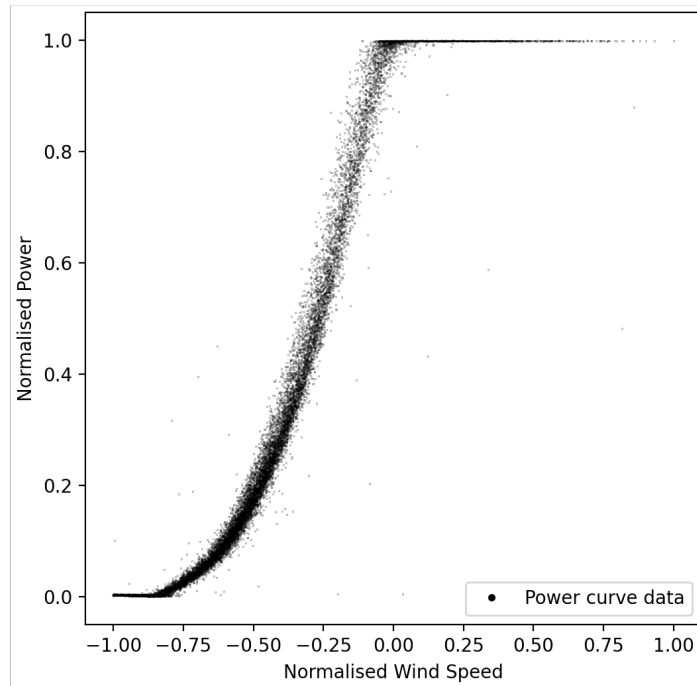


Figure 2.1: Example of a power curve

The power curve is a fundamental characteristic of a wind turbine, describing how much power a turbine can be expected to extract from a given wind speed over the turbine. The power curve exhibits a theoretically cubic relationship between power and wind speed [96], and is bounded on two sides in the power dimension. The lower bound of the power curve occurs at zero power, where the wind does not contain

enough energy to move the turbine. The upper bound is the maximum power that the turbine is designed to safely extract from the wind, also known as the rated power. If wind speed increases while the turbine is operating at rated power, it will regulate itself to maintain rated power before shutting down should speeds get too high. An accurate power curve model means accurate power generation predictions. This has numerous financial and engineering benefits; knowing the expected power output of a turbine can be used to inform strike price, help with grid management and monitor the structural integrity of the turbine [93].

Using a GP to model the power curve is a sensible choice; the data shown in Figure 2.1 is stochastic, and can better be modelled with a probabilistic approach. Further, the returned uncertainty over the power prediction can be used to understand and model uncertainty propagation on financial returns and structural health of the turbine, allowing for robust long term risk planning. The prevalence of Gaussian Processes in the power curve literature reflects their utility, this is especially notable in anomaly and outlier detection tasks where the uncertainty quantification of GPs provides additional value [93, 97, 98]. More complex GPs are also employed, such as a heteroscedastic GP to capture the input varying noise in Rogers et al. [99]. It is clear then that GPs provide significant utility to power curve modelling, however, GP predictions have no guarantee of being physically meaningful (and in practise for this problem, are often not). While one could argue that the predictive density could simply be truncated at the bounds, this is not a statistically robust or satisfying approach. A robust method of modelling the uncertainty such that it remains physically meaningful is therefore desirable.

Before establishing how this bounding of the uncertainty will be done, it is useful to first examine existing approaches to power curve modelling. A broad range of models have been proposed, and a comprehensive review is given by Lydia et al. [100]. Relatively few studies, however, attempt physically meaningful probabilistic power curve modelling. Those that do often focus on fitting bounded parametric distributions, such as the Weibull, through conventional estimation techniques [101, 102]. From the literature, it is clear that there remains a clear gap for application of ML tools that obey the physical bounds of the power curve in the prediction space. Similarly, exploration of bounded Gaussian processes (GPs) in the wider SHM literature is limited, though Jones et al. [75, 80] have begun investigating approaches to bounding GPs via the kernel function.

This chapter seeks to address this gap in the literature by investigating two bounded Gaussian Process (GP) models. The first is a warping method, which transforms the data and the GP outputs to satisfy boundary conditions. The second is a beta likelihood approach, which naturally bounds the outputs to the unit interval. Both approaches are modelled with sparse heteroscedastic¹ GPs in order to deal with the large, heteroscedastic dataset. The literature surrounding bounded GPs is extensive², and both modelling strategies build on established ideas. Warping functions have long been recognized as an effective way to impose bounds on GPs [103], and the heteroscedastic beta likelihood was briefly proposed in Saul et al. [104]. While numerous bounding methods exist, these two were selected for their ability to enforce two-sided bounds on the power curve within a single model while preserving the GP's capacity for robust uncertainty quantification.

The contribution of this chapter lies not in proposing new models, but in their novel application and interpretation. Although Saul et al. [104] introduce the heteroscedastic beta model, the role of the beta distribution as a bounding mechanism receives little attention. Similarly, while warping [103] has long been established, this chapter advances the approach by incorporating heteroscedasticity and employing a modern sparse variational framework. Most importantly, this chapter presents robust, probabilistic bounded models for power curves for the first time, contributing both practically and theoretically, and extends the study of bounded GPs within the wider SHM literature.

The chapter proceeds as follows. First, the necessary theory for heteroscedastic GPs and sparsity is introduced, followed by a presentation of two bounding methods. These bounded models are then evaluated against one another, as well as against a standard GP baseline, using anonymised power curve data from an operational offshore wind turbine. The results demonstrate that bounded GPs offer both practical and conceptual advantages within SHM frameworks. In particular, the beta GP is shown to capture uncertainty more effectively, providing improved uncertainty quantification while also remaining physically meaningful.

¹A heteroscedastic GP is a GP designed to model heteroscedastic noise, otherwise known as input dependant noise. This will be expanded upon in subsection 2.1.1

²For a comprehensive literature review of GP bounding methods see Swiller et al. [72]

2.1 Bounding approaches and ancillary theory

This section describes the two bounding methods used, as well as the necessary theory in heteroscedastic GPs (which both methods use) and the necessary sparse variational approximation scheme.

2.1.1 Heteroscedastic noise

Heteroscedastic noise is noise that varies as a function of a function's input. The power curve data in Figure 2.1 is an excellent example of this; it is clear that the data is much noisier in the centre of the power curve than towards its limits. Therefore, the input (the wind speed) is correlated to the amount of noise, making this heteroscedastic noise. The standard (homoscedastic) GP assumes that noise and hence uncertainty is constant across the full input range, and as such cannot model the way the noise changes with wind speed. In order to model the heteroscedastic noise and return a more useful probabilistic power curve, a heteroscedastic GP model can be used, as was done in Rogers et al. [99].

The particular heteroscedastic model adopted in this work (distinct from [99]) uses two latent GPs to learn the location (Equation 2.1a) and scale (Equation 2.1b) of the distributions required, first shown in [105]. This is done by connecting a Multi-output kernel to a heteroscedastic likelihood, mapping the latent GPs onto a single function. The heteroscedastic GP model can be described as:

$$f_1(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k_1(\mathbf{x}, \mathbf{x}')) \quad (2.1a)$$

$$f_2(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k_2(\mathbf{x}, \mathbf{x}')) \quad (2.1b)$$

$$m(\mathbf{x}) = f_1(\mathbf{x}) \quad (2.1c)$$

$$s(\mathbf{x}) = g(f_2(\mathbf{x})) \quad (2.1d)$$

$$y(\mathbf{x}) \sim \mathcal{N}(m(\mathbf{x}), s(\mathbf{x})) \quad (2.1e)$$

The transform $g(\cdot)$ in Equation 2.1d in this case is an exponential transform, in

order to keep the function positive, since it models the variance of the noise which must be greater than zero. In the model defined in Equation 2.5, the two latent GPs have been assumed independent. However, this does not necessarily have to be true - the latent GPs can be modelled as dependent should the need arise by linking Equation 2.1a and Equation 2.1b through a multi-output GP, e.g. [106].

2.1.2 Warping functions

The first bounded model investigated in this chapter is the warping function. A visual illustration of how the warping function works is shown in Figure 2.2. Figure 2.2a is an illustrative example of how a simple normal distribution modelling uncertainty at rated power has a large portion of probability mass outside of what is physically possible.

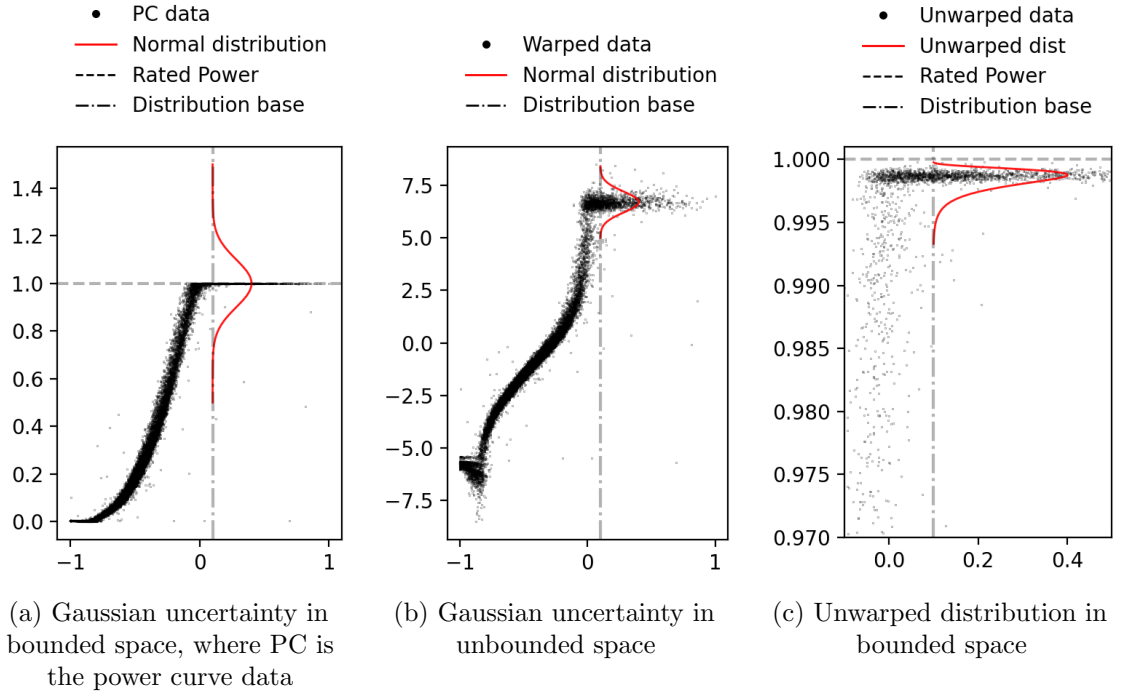


Figure 2.2: Illustration of each step of the warping process - the normal distributions are for illustrative purposes only

The warped bounding method starts with transforming the power data using the logit function [72], shown in Equation 2.2. Conceptually, the warped GP moves the data from the bounded space to an unbounded one by means of a nonlinear transformation. The warped power curve is shown in Figure 2.2b. The logit function

moves the power curve from a bounded space ($p \in (0, 1)$) in the power dimension to an unbounded space ($p \in (-\infty, \infty)$) in the power dimension. The GP is then applied to the warped power curve, as in this unbounded space the bounds on the power curve no longer apply, so the assumption of Gaussianity holds.

$$\text{logit}(p) = \ln \left(\frac{p}{1-p} \right) \text{ for } p \in (0, 1) \quad (2.2)$$

After fitting, the data and the returned uncertainty distribution are unwarped using the inverse of the logit function, resulting in the bounded data and distribution shown in Figure 2.2c. An important note on this method is that it is difficult to evaluate the unwarped uncertainty as no closed form solution exists for it. It is clearly possible to sample it via some Monte Carlo approach, but this is beyond the scope of the current work. The density shown in Figure 2.2c is presented for visualisation but is not exact. It is therefore simpler to work with and evaluate the model directly in the warped space. To illustrate how the uncertainty looks in the unwarped space for the entire power curve, a confidence bound can be created in the warped space at $\pm 2\sigma$ and unwarped with the predictive mean.

2.1.3 Beta likelihood

A typical GP model is formulated using a Gaussian likelihood. This assumes that the data is drawn from a normal distribution, i.e. $f(x)$ has noise ε added and $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$. This results in a normally distributed posterior function, as shown in Figure 2.2a. Without any modifications, this is not a suitable model for probabilistic power curve modelling. Instead of warping the output space, the likelihood can be changed to a non Gaussian likelihood - one that better represents the distribution of the observed data for the problem at hand.

The likelihood chosen here is the Beta likelihood, imposing the assumption that the data is drawn from a Beta distribution. The Beta distribution is a continuous probability distribution in the interval $[0, 1]$. It is defined in Equation 2.3 where α and β are the shape parameters of the distribution. The probability density function is given in Equation 2.4, where $B(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha + \beta)$ and Γ is the Gamma function. The use of the Beta likelihood is a bounding method because the GP with

a Beta likelihood would imply that data were generated from Beta distributions, which are inherently bound between $[0,1]$.

$$y \sim \text{Beta}(\alpha, \beta) \quad (2.3)$$

$$p(y) = y^{\alpha-1} \frac{(1-y)^{\beta-1}}{B(\alpha, \beta)} \quad (2.4)$$

Using a non Gaussian likelihood in the GP like the Beta likelihood results in an analytically intractable posterior [71]. The posterior must instead be approximated, numerous approximations exist for this problem. A common method (used in the popular GP packages GPflow [107] and GPML [108]) to approximate the posterior is MCMC [109] sampling. However, this can require a large amount of samples to approximate the posterior well, and can be very slow.

Instead of using MCMC, this work approximates the Beta likelihood using variational inference and quadrature of a heteroscedastic GP. The model proceeds by introducing two latent GPs to calculate α and β as functions of the input, instead of the mean and variance as used in the normal heteroscedastic GP in subsection 2.1.3. This is expressed mathematically below, where $t(\cdot)$ represents an exponential transform to keep the functions over α and β positive.

$$f_1(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k_1(\mathbf{x}, \mathbf{x}')) \quad (2.5a)$$

$$f_2(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k_2(\mathbf{x}, \mathbf{x}')) \quad (2.5b)$$

$$\mathbf{g}_1 = t(\mathbf{f}_1) \quad (2.5c)$$

$$\mathbf{g}_2 = t(\mathbf{f}_2) \quad (2.5d)$$

$$\mathbf{y} \sim \text{Beta}(\mathbf{g}_1, \mathbf{g}_2) \quad (2.5e)$$

A flow diagram of this process is shown in Figure 2.3. Figure 2.3 illustrates how f_1 and f_2 are calculated from the power curve data using the heteroscedastic GP, both f_1 and f_2 are distributed normally for each input. These are then transformed

exponentially to create g_1 and g_2 which are approximations of α and β . Using total variance and mean, the distributions of g_1 and g_2 are used to find a final α and β value for each wind speed input. These are then used to create the final Beta distributions at each input shown in the final panel.

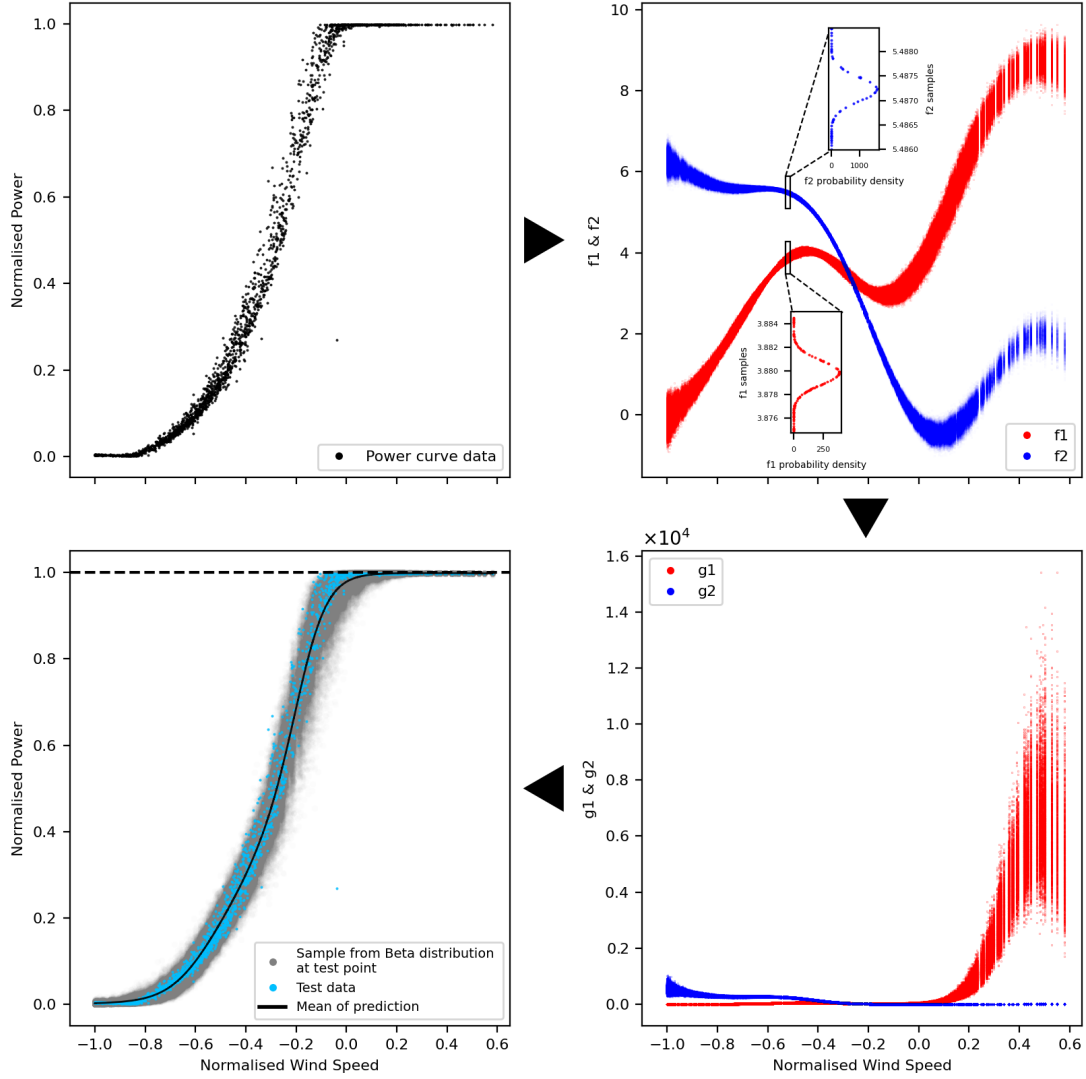


Figure 2.3: Flow chart of the Heteroscedastic Beta Process

This approach is useful because it allows for fast posterior approximation with established methods, in this case the Stochastic Variational Inference approximation (SVGP) [54], explained further in subsection 2.1.4.

A further benefit to this approach is that the heteroscedasticity of the model is not lost, as α and β vary with inputs and as such capture the varying uncertainty with inputs as one would expect, alongside capturing the progression of the mean function.

2.1.4 Approximations and Quadrature solutions

When considering the large heteroscedastic models used here, two problems arise. The first is that solutions for heteroscedastic and non-Gaussian models do not, in general, exist in closed form, and must be approximated to reach a solution. The second is that GPs are computationally inefficient when used on large datasets. This is due to their reliance on a matrix inversion when calculating the posterior and the marginal likelihood, which has time complexity $\mathcal{O}(N^3)$ and therefore scales poorly with increasing numbers of training data points. In this case, calculating the entire posterior using the dataset of $>17,000$ is impractical and would require an excessive amount of computational resource and time.

As mentioned in subsection 1.3.1, inducing point approximations offer a solution to both of these problems. This chapter uses a stochastic variational inference approach [54] to approximate and sparsify the posteriors of interest. Similarly to other inducing point methods, the method applied in this work uses a set of inducing points $\mathbf{u}$ to summarise the model. These “summarising” inducing points are then marginalised out of the joint posterior $q(\mathbf{f}, \mathbf{u})$ in order to obtain the approximate marginal posterior over the latent function f , as shown in Equation 2.6. Given a Gaussian prior and Gaussian approximating forms, the joint $p(\mathbf{f}|\mathbf{u})$ has a closed form solution, as illustrated in [104]

$$q(\mathbf{f}) = \int q(\mathbf{f}, \mathbf{u}) d\mathbf{u} = \int p(\mathbf{f}|\mathbf{u}) q(\mathbf{u}) d\mathbf{u} \quad (2.6)$$

The approximate posterior is found by maximising the bound on the marginal likelihood. This bound is known as the Evidence Lower Bound (ELBO) and is used instead of the marginal likelihood to reduce computational cost. The global stochastic variational approach from [54] differs from other variational inference bounds in that it allows for the inducing points to be global points representing the entire GP despite being calculated locally - thus allowing faster computation.

A formulation of the evidence lower bound (ELBO) of a heteroscedastic GP is shown in Equation 2.7 [104], where $\int q(f_i)q(g_i) \log p(y_i|f_i, g_i) df_i dg_i$ is the likelihood term. The terms $q(f_i)$ and $q(g_i)$ are the marginal variational distributions (as shown in Equation 2.6) of the mean and variance latent GPs respectively, and $\log p(y_i|f_i, g_i)$ is

the conditional probability term. The model can then be optimised from this bound using stochastic gradient descent.

$$\begin{aligned} \log p(\mathbf{y}) \geq & \sum_{i=1}^n \int q(f_i)q(g_i) \log p(y_i|f_i, g_i) df_i dg_i \\ & - \text{KL}(q(\mathbf{u}_f)||p(\mathbf{u}_f)) - \text{KL}(q(\mathbf{u}_g)||p(\mathbf{u}_g)) \end{aligned} \quad (2.7)$$

The likelihood term is clearly an n term integration over the marginal variational distributions and the conditional probability. As long as these terms are all Gaussian (such as in the heteroscedastic GP model), this integration has an analytical solution. However, the conditional term $\log p(y_i|f_i, g_i)$ is not Gaussian in the HBP model, and as such the likelihood term does not have an analytical solution. It is therefore approximated with Gaussian-Hermite quadrature, as discussed in subsection 1.3.1 and shown in Saul et al. [104].

The use of sparse variational inference allows a GP to be computed with time complexity $\mathcal{O}(NM^2)$ instead of $\mathcal{O}(N^3)$, where M is the number of inducing points. In a chained GP like the heteroscedastic one explored here, this complexity scales linearly with the number of latent GPs being explored. Note that the inducing points M do not have to be the same for each latent GP. This saving solves the computational complexity problem, as it now requires significantly less time to train the heteroscedastic models. As mentioned earlier, the approximated posterior is found by maximising the ELBO. This means that the heteroscedastic methods can be used, as they can be approximated efficiently.

2.2 Case study

The data used in this chapter comes from the supervisory control and data acquisition (SCADA) system of an operating wind turbine. The SCADA data is the 10 minute average of power output and wind speed (as taken on the nacelle) is comprised of 50,000 readings. The measured values of wind and power have been obscured by scaling for confidentiality reasons.

The dataset was split into three equal parts: a training dataset, a testing dataset and a validation set. The GP models used in this chapter are trained using the training data. The predictive performance of the trained models are then evaluated on the test data, which they have not seen before. The validation set is put aside should further validation of predictive performance be required.

After the initial scaling to anonymity, the power output data was again normalised between 0 and 1 in order to allow for the Beta likelihood approach to work. Neither the original scaling or the subsequent normalisation impacted the relationship between power output and wind speed as both were linear transforms [110]. The training data was then cleaned of severe outliers, for example known curtailments.

Two performance metrics will be used in order to compare the success of the various models in modelling the power curve. The first is Normalised Mean Squared Error (NMSE), shown in Equation 2.8, where $\mathbf{y}$ are the true values of the test vector, σ^2 is the variance of the test vector and $\hat{\mathbf{y}}$ is the prediction. The NMSE assesses how close to the mean of the predictive distribution the test value was, and can be thought of (and functions similarly to) percentage error.

$$\text{NMSE} = \frac{100}{N\sigma^2} \sqrt{(\mathbf{y} - \hat{\mathbf{y}})^T (\mathbf{y} - \hat{\mathbf{y}})} \quad (2.8)$$

The second metric being used is the joint predictive log likelihood (PLL) which is computed as the sum of the predictive log likelihood at every test value under the posterior, Given in Equation 2.9.

$$\text{PLL} = \sum_{i=1}^{N_*} \log p(\mathbf{y}_{*,i} | \mathbf{x}_{*,i}, \mathcal{D}) \quad (2.9)$$

This is an especially important metric for this chapter as it is a measurement of how good the uncertainty of a specific model was at capturing the test data and not solely the performance of the mean estimate. The larger the joint log likelihood, the better the uncertainty of the model has captured the test data and the more successful the model. While interpreting the absolute value of the PLL can be challenging since it scales with dataset size, this poses no issue for this comparative analysis since the test set is identical across all evaluated models. Readers interested in a normalised

or standardized metric may instead consider the Negative Log Predictive Density or the Standardized Mean Log Loss respectively.

To facilitate the discussion of results, the NMSE and joint log likelihood of each model is shown in Table 2.1. The results of each model will be further discussed in their respective following subsections. Each model was run with 250 inducing points (for the heteroscedastic models, this is 250 for each latent function) and a minibatch size of 800. Both hyperparameters and inducing point locations were optimised via the ADAM gradient descent optimiser. Inducing points were initialised via uniform grid spacing over the input space. Alternative methods such as clustering or prior sampling having been found to have no or little impact on model performance. Since this is a relatively simple 1D regression problem, optimisation was robust across a range of reasonable hyperparameter initialisations.

Model	NMSE	Joint log likelihood
Standard GP	0.24	32312
Warped heteroscedastic GP	5.4	-12469
HBP	0.53	68866

Table 2.1: Case study results

It will be argued by the authors in the remainder of this section that the HBP best captures the physical behaviour and uncertainty associated with the power curve of an operating wind turbine. All the models presented in this chapter were written in Python and run on GPflow [107].

2.2.1 Simple GP Regression

A standard (homoscedastic sparse) GP model was trained over the power curve to act as a benchmark to the two bounded models being tested. The standard GP used an additive composite kernel compromised of a Matérn 3/2 kernel and a Linear kernel. This combination was used to capture the upward linear trend of the data.

In Figure 2.4a the results of the standard GP model can be seen. The model seemingly performs well in its mean predictions with an NMSE of 0.24, it also shows a log likelihood of 32312, less than half of that seen for the HBP, indicating worse

performance in the uncertainty quantification. Figure 2.4b shows a magnified portion of the same model at rated power in order to examine how successful the model was at quantifying uncertainty at rated power. The standard GP model captured the shape of the power curve very well on average across the range of wind speeds, and the NMSE reflects this. The uncertainty prediction is comparatively poor; the uncertainty distribution is underestimated away from the bounds and overestimated close to them. This can be seen in that some data is not covered by the distribution in the middle of the power curve, and that the distributions at rated and zero power overestimate the variance.

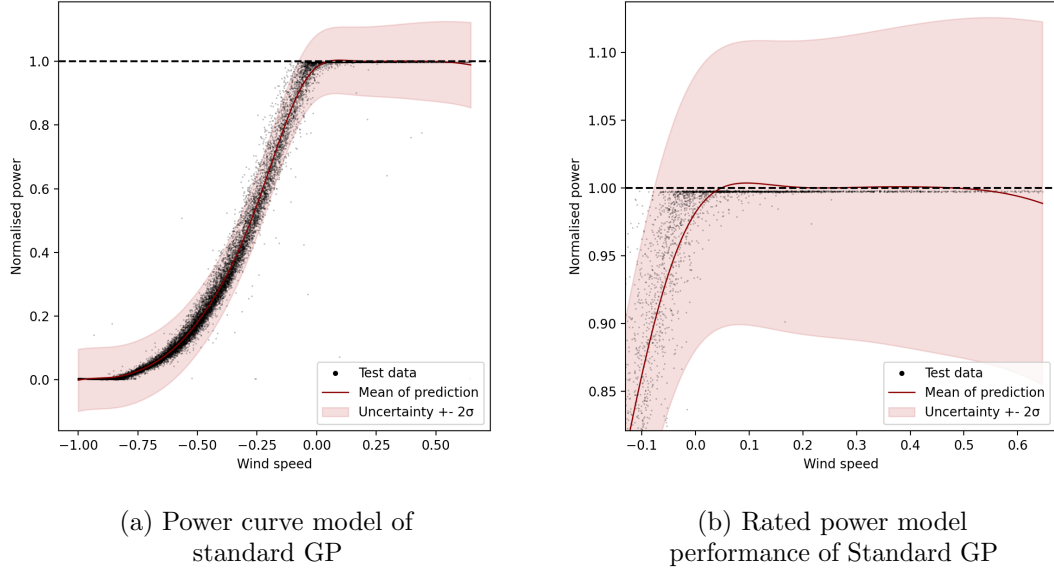


Figure 2.4: Initial power curve prediction with GP

While the GP has an excellent NMSE and good joint log likelihood, it is clear from visual examination of Figure 2.4b that the predicted mean and uncertainty do not obey physical limits and overestimate possible values of the rated power. Perhaps the point of most concern here, is that despite the excellent NMSE score, there are still predictions above the rated power output. This output is then not a particularly useful model if carried forward into further analysis, as the errors in the uncertainty and mean estimation could mask damage in the turbine or suggest that the financial performance of the wind could exceed what is physically possible.

Finally, it is clear that the GP struggles to correctly predict the transition to rated power. This could be because continuous function models such as the GP can struggle

with abrupt transitions such as this one. The transition to rated power was addressed using Bayesian committees in [99].

2.2.2 Heteroscedastic warped Gaussian process

The prediction of the power curve with the warped GP is shown in Figure 2.5a, and a magnified image of the model at rated power is shown in Figure 2.5b. The warped heteroscedastic GP used a Matérn 3/2 kernel for both the mean and variance GPs. Note that the inclusion of an additive linear kernel for the mean (as used in the other two models) created numerical stability issues. The performance metrics for this model were comparatively very poor, with an NMSE of 5.4 and a log likelihood of -12469. The warping approach has failed to correctly capture either the mean behaviour or the uncertainty in the power curve.

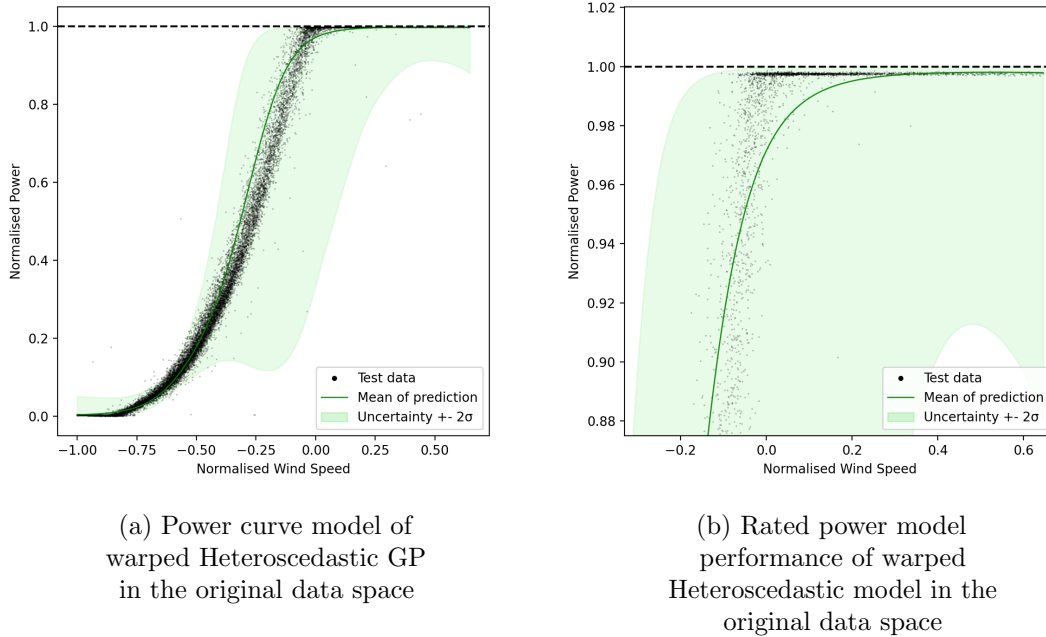


Figure 2.5: Initial power curve prediction with GP

The performance metrics clearly show that the warped GP has performed poorly. In order to understand why, the fitted GP must be examined in the warped space (shown in Figure 2.6), as this is where the GP interacts with the data being modelled. It is clear from Figure 2.6 that the GP has failed to map the warped function correctly.

This is likely because warping the power curve has resulted in a data shape that is too noisy and discontinuous for the GP with a Gaussian likelihood to model well.

The inherent noise of the power curve in the power axis was exacerbated during the warping process, resulting in large vertical noise distributions that the GP cannot capture except by increasing the variance. This would confuse the training of the GP and is why such large uncertainty distributions are returned in Figure 2.5a and Figure 2.6. The discontinuity of the warped shape is also a problem because the GP is a continuous function and struggles to model abrupt changes as discussed in subsection 2.2.1 and [111]. This results in the GP smoothing over the abrupt changes in the model as seen in Figure 2.6. It appears that this challenge in modelling the discontinuous transition is amplified when projecting the data through the warping function.

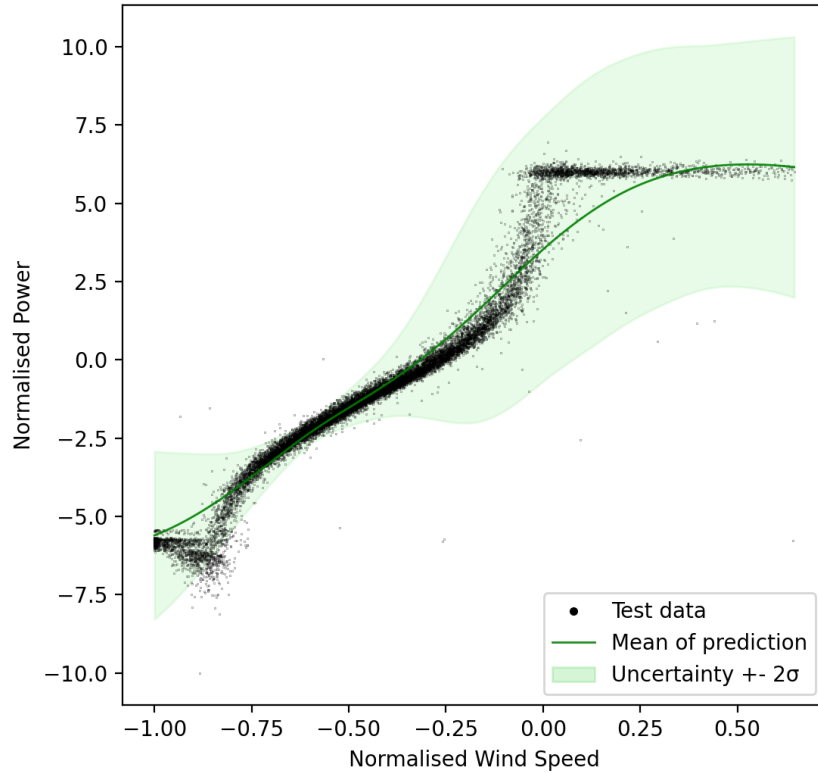


Figure 2.6: Heteroscedastic GP in the warped space

Despite being an intuitive approach, the heteroscedastic warped GP is clearly not that useful in assessing either the predictive mean or the uncertainty. The issues

with this model lie in the warped space, where the data shape becomes too noisy and discontinuous. These problems could be addressed in several ways such as with further cleaning in the warped space, with a Bayesian committee such as in [99] or an alternative warping function. However, the author proposes that altering the likelihood function in the model is a superior approach to application of warping.

2.2.3 Beta Likelihood

The results from the HBP model can be seen in Figure 2.7a. The HBP used an additive composite kernel comprised of a Matérn 3/2 kernel and a Linear kernel for the transform of α , and a squared exponential kernel for the transform of β . In order to demonstrate the effectiveness of the bounded UQ provided by the HBP, as well as illustrate that the uncertainty is recoverable in closed form, a heatmap of the predictive uncertainty over the test data is shown in Figure 2.8. The heat map is capped at 30 due to the likelihood being so high towards the bounds.

The NMSE for this model is 0.53, while the log likelihood is 68866. These are excellent results. The log likelihood is the highest of all the models tested, and shows that the bounded distribution best captures the power curve data. The NMSE is higher than for the standard GP, however the difference (0.53 vs 0.24) is small enough that these results can be considered comparable. It is worth noting again the previous discussion regarding the failure of the standard GP model to obey the physical bounds of the system in both the mean and the variance despite its lower NMSE.

Choosing between the standard GP and HBP will depend on the desired use case. When uncertainty matters, the HBP is clearly a superior model. If only the mean prediction matters then the standard GP could be considered the better choice, however it is worth considering that the lower NMSE of the standard GP is misleading; the predictive mean exceeds rated power (as seen in Figure 2.4b) and as such is physically implausible. If this is not accounted for, using the unbounded model could have serious consequences should its predictions be taken forward - such as over-predicting financial returns or masking turbine damage.

As seen in Figure 2.7a the model has captured the shape of the power curve well.

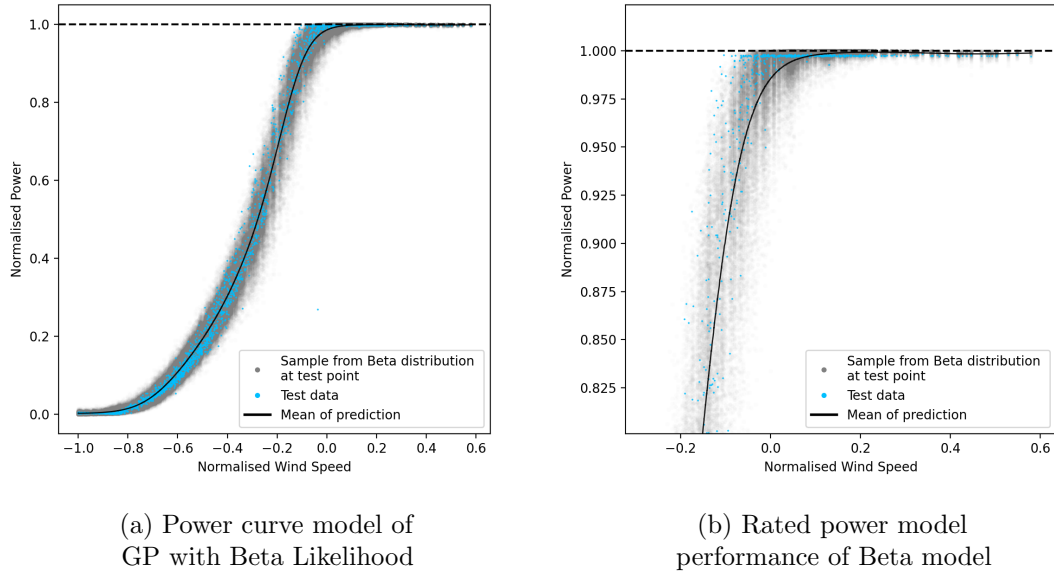


Figure 2.7: Initial power curve prediction with GP

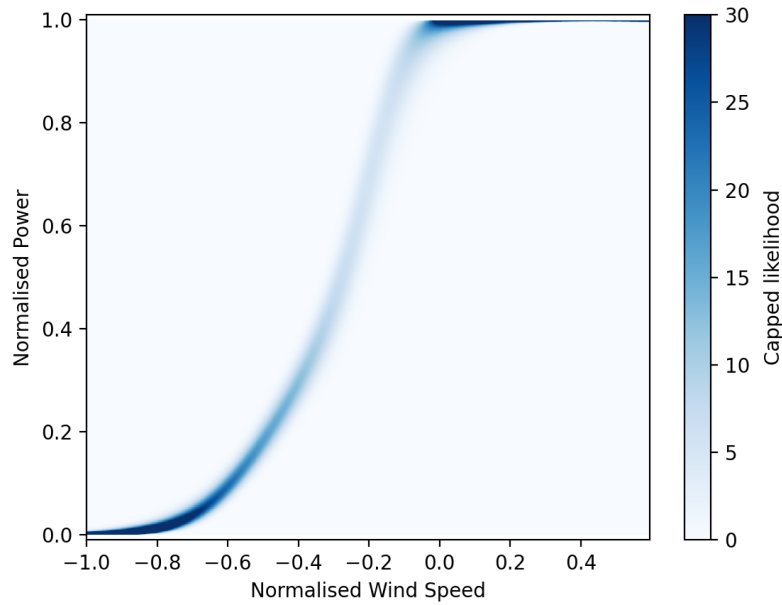


Figure 2.8: Capped heat map of the predictive uncertainty of the HBP

Similarly to the other models, this model also struggles to accurately fit the transition to rated power. This is marginally less pronounced as the use of Beta distributions allow the mean to be “squashed” into the transition better than a normal distribution would allow.

It is clear from the performance metrics that the HBP model is successful. It presents significant utility in that a full bounded predictive uncertainty distribution is returned. This uncertainty distribution also captures the data better than the standard GP and warped model. As mentioned in previous sections, this has significant financial and engineering benefits. The NMSE is comparable to the standard GP but offers the extra security that the mean prediction is bounded, removing the potential for error that an unbounded model carries and providing the basis for increased operator confidence in the model.

2.3 Discussion

The novelty of this chapter was to investigate the use of two bounded GP methods in relation to power curve modelling. As discussed previously, this is a comparatively underdeveloped area of power curve modelling, and while both methods have been theorized about in wider GP literature, have never been applied to SHM problems or interpreted/implemented in this way.

Of the two methods investigated, it was argued that HBP model was the most successful, and performs favourably compared to a baseline GP, with a significantly higher joint likelihood and comparable NMSE. The higher joint likelihood score indicates that the bounded uncertainty region better captures the test data than its unbounded counterpart, a key result and especially significant when the use of predictive uncertainty in further calculations is considered. The difference in NMSE between the standard GP and HBP is small enough to be functionally comparable, and ignores the fact that some of the predictions returned by the standard GP are not physically plausible. Even in a perfect scenario, where predictions from the standard GP do not appear exceed the bounds with its mean predictions, the model cannot provide confidence that its mean predictions are 100% physically plausible for all possible test cases going forward. The HBP is therefore superior in establishing operator trust and reliance on the power curve model, even when compared to a theoretically perfect standard GP.

While the proposed methods for bounding the GPs met with varying levels of success, both models open up avenues for further work. It was clear that the heteroscedastic

GP did not perform adequately in the warped space. Several methods to remedy this exists, and although outside the scope of this thesis, may be considered in the future. These include using Bayesian committees to tackle the discontinuous nature of the warped data shape or experimenting with alternative warping functions. The GP with Beta likelihood was successful, and while the fast approximation of the likelihood with the heteroscedastic method worked well, the random nature of the sampling and unification of the various distributions into an average distribution at each test point is not the most statistically robust method. Uniting the beta distributions of each sample mathematically rather than using the mean and total variance would be more robust and would be worth pursuing. Finally, it may be worth considering again the use of committee machines to better handle the transition to rated power seen in the power curve.

To conclude, the work presented in this chapter has significant novelty in the field of SHM. The HBP model is an important step in better quantifying the uncertainty of power curve models, and both models investigated have opened up new avenues of research in the field. The key result of this chapter was to show that bounding the GP to respect physical limitations significantly improves uncertainty quantification, a result with interesting ramifications for industrial use moving forward. The remainder of this thesis will now move on to examining the use of Gaussian Process classifier tools in SHM, and present methods in improving their use and application.

POWER EXPECTATION PROPAGATION AND GAUSSIAN PROCESS CLASSIFIERS IN SHM

Chapter Highlights:

- GP Classification is explored as a underutilised tool in SHM
- Binary and Multi-class GPCs are benchmarked on SHM problems
- Power Expectation Propagation is introduced as the approximation tool of choice for GPCs
- It is intuitively shown how to extend PEP to multi-latent likelihoods (including multi-classification)

As previously discussed, this thesis aims to enhance data driven methods for the Detection and Localisation stages of Rytter’s hierarchy [1]. Since these stages are discrete classifications of structural states, a classification approach may be more natural than a regression one.

That classification models are better at classification is obvious, but it is important to understand *why* this is so and *when* they should be used in SHM. Classification

models explicitly learn a decision boundary between states, either directly, as in support vector machines, or implicitly, as in Gaussian Process Classifiers. Regression, by contrast, is designed for prediction of continuous outputs, so using it for discrete state prediction typically requires arbitrary thresholds or distance metrics in the prediction space (e.g. [112–114] are all novelty or outlier detectors using regression models for damage detection).

This distinction matters. A learnt decision boundary can typically accommodate noisy data, nonlinear separations, and multiple classes. Conditions under which a regression approach will struggle. The major drawback of classification however, is its reliance on expensive labelled data from all damage scenarios of interest. Since damage cases and locations are functionally infinite, and exponentially rarer the more catastrophic they are, this flaw presents a critical challenge for the field. Thus, when appropriate data are available, classification methods offer a more principled and robust framework for detection and localisation tasks within Rytter’s hierarchy.

A vast body of literature exists on the use of classification models in SHM when all damage classes are known (known as supervised classification). The studies in [115–117] all provide benchmarks for common off-the-shelf classifiers (e.g. Support Vector Machines (SVMs), random forest, feedforward neural networks and naive Bayes classifiers) applied to various supervised SHM problems. There are three notable approaches in the literature for addressing the data labelling challenge. These are; semi-supervised learning, transfer learning, and active learning. Semi-supervised learning involves learning on both labelled and unlabelled data, and is a common approach in SHM. Models range from Gaussian Mixture models [118], graph neural networks [119] to Dirichlet Process Mixture models [120]. Transfer learning is the transfer of data between similar structures [121], and is an exciting avenue of research for SHM where structures often do bear significant similarity. Active learning can be thought of as learning which points are the most informative to label in order to deal with the expensive labelling issue, and will be explored further in the next chapter.

Following the narrative trend of this thesis the following question can be formulated: “Given that classification is a task of interest in SHM, is Gaussian Process Classification (GPC) an avenue worth exploring?”. The rest of this thesis contends that the answer is *yes*; presenting philosophical justification, demonstrating the benefits of the UQ innately provided by GPCs and highlighting the limited existing literature on their

application in the field. While it will be shown that GPCs offer significant benefits to SHM, it must be noted that they are a supervised method and thus require fully labelled data. This chapter examines their performance under the assumption that these data are available, while the next explores active learning as a way to mitigate this limitation.

Philosophically, the arguments for probabilistic classification in SHM remain similar to the regression setting: Uncertainty quantification provides richer and more meaningful results for immediate stakeholders, and a probabilistic classifier is inherently more suited to modelling noisy data. Further, the UQ results can easily be propagated downstream to other tasks such as risk-based decision making [44, 122]. Finally (and unique to classification), a Bayesian model can be easily formulated to deal with unbalanced classes via priors, usefully redressing the comparative rarity of damage classes in real datasets.

Before continuing it is worth considering if other probabilistic classification methods have been applied to SHM that might already provide adequate uncertainty quantification. Certainly, a wide the range of Bayesian classification methods have been applied to the field; From Bayesian neural networks [123] and Bayesian Linear Discriminant analysis [124] to relevance vector machines [125], the list is not short. Interestingly, there is a noticeable Gaussian shaped hole in this probabilistic wall of classifiers, with only a few prior works such as Alvarez et al. [89] examining the use of GPCs in the field. This is despite GPCs offering several benefits that make them uniquely suited to SHM labelling problems. As discussed previously, GPs offer a high degree of modelling flexibility, and crucially in the SHM case are comparatively interpretable and scale relatively well when sparsified¹. Gaussian Processes also allow prior knowledge and physical insights to be incorporated via the kernel function, enabling the model to reflect known structural behaviour such as smoothness, periodicity, or spatial correlation. Finally, the GPC posterior is an interpretable, already calculated form of uncertainty quantification, simplifying its use in downstream tasks (as shown in the next chapter).

Prior work such as Alvarez et al. [89] have made an excellent first step into the use of binary GPCs in SHM, however the application was with the FITC approximation

¹GPs operate extremely well in small data environments, and when sparsified (which they are throughout this thesis) can be extended to much larger data environments

scheme [126] and on a limited problem set. This chapter will expand upon this work by applying it to a more common set of SHM problems and crucially by introducing the state-of-the-art (SOTA) approximation scheme for GPCs; Power Expectation Propagation (PEP) [55]. Further, it is illustrated why PEP is the approximation scheme of choice in section 3.3. Results closely mirror that of [127].

The extension of binary GPCs to multi-class is clearly desirable - doing so would unlock a wider range of problems with a much greater level of predictive power. To the authors knowledge, no literature exists on the use of multi-class GPCs in SHM. Since it is SOTA for classification ([55, 127, 128]) is desirable to maintain the use of PEP as an approximating framework in the extension to multi-class. Building on the multi-class PEP GP derivation of Villacampa et al. [129], the multi-class formulation can be viewed as one specific member of the wider class of multi-latent (or “chained”) GPs [104]. This observation motivates a generalised PEP approximation covering the full spectrum of chained GPs. This generalisation is presented in this chapter, with the multi-class GPC and a heteroscedastic GP used as motivating examples. The resulting framework represents a meaningful contribution to the PEP literature, enabling more flexible and advanced applications of PEP approximations. Finally, the multi-class GPC is, for the first time, benchmarked on two challenging SHM datasets.

This chapter therefore focuses on the use of GPCs in SHM. Much of the literature surrounding Expectation Propagation (EP) is mathematically dense, presenting a significant challenge for practitioners looking to apply this approximation framework. This chapter aims to help bridge this gap by offering intuitive tutorials on Expectation Propagation and its extension to multi-latent GPs. Both binary and multi-class GPCs will be benchmarked on a variety of SHM problems, significantly furthering the highly limited exploration of these methods in the SHM literature. The GPCs will be benchmarked against SVMs with an RBF kernel, as SVMs are common deterministic models of a similar class as GPCs (used in small-medium data environments, performant but relatively light and interpretable). It will be shown that GPCs perform similarly or better than SVMs but offer the additional benefit of a probabilistic framework underpinning its predictions.

3.1 Binary Gaussian Process Classification

In the binary case, a Gaussian Process Classifier (GPC) can intuitively be thought of as a discriminator over the latent function representing the class labels [62]. This is illustrated below in Figure 3.1. In the binary case, the two classes are represented as values above or below zero. The latent function is attempting to model if a point is positive or negative rather than assign it a real value. In prediction, if the latent function at that test point is confidently above zero, the test point is labelled as class one.

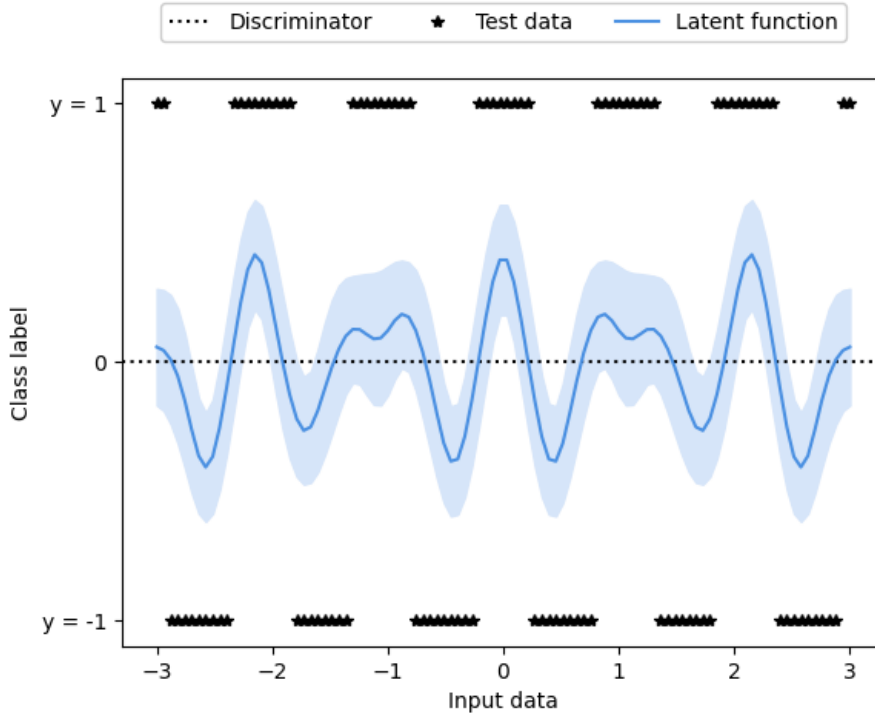


Figure 3.1: Basic example of a GPC classifying some problem generated by a sine wave. Data greater than 0 is in class 1, and data less than 0 is in class 2.

Following from the notation introduced in section 1.3, it will be shown that the structure of a GP is almost entirely unchanged when applied to classification. The key change is the probabilistic mapping (likelihood) $p(y_i|f, x_i, \theta)$ between inputs and outputs. Where in regression it is $p(y_i|f, x_i, \theta) = \mathcal{N}(y_i; f(x_i), \sigma^2)$ (i.e. $y_i = f(x_i) + \epsilon_i$), in binary classification it becomes:

$$p(y_i|f, x_i, \theta) = \mathcal{B}(y_i; \sigma(f(x_i))) \quad (3.1)$$

Where $\mathcal{B}$ is the bernoulli distribution and $\sigma(f(x_i))$ is a sigmoidal link function, which maps the latent function onto the unit interval. It follows very naturally then that $p(\mathbf{y} = +1|\mathbf{x}) = \sigma(f(\mathbf{x}))$, hopefully making it clear that the GPC is very simply a discriminator over $f(\mathbf{x})$ mapped to the unit interval, as shown in Figure 3.1.

Since the labels are modelled as independent Bernoulli variables in Equation 3.1, and the probit sigmoid is chosen here for analytical convenience ($p(\mathbf{y} = +1|x) = \Phi(f(\mathbf{x}))$), the joint likelihood factorises into Equation 3.2:

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^n p(y_i|f_i) = \prod_{i=1}^n \Phi(y_i f_i) \quad (3.2)$$

Given a prior on $\mathbf{f} \sim \mathcal{N}(0, \mathbf{K})$, this results in the posterior on the latent function for given hyperparameters θ as given by Equation 3.3, where $Z = \int p(\mathbf{f}) \prod_{i=1}^n \Phi(y_i f_i) d\mathbf{f}$ is the normalisation constant.

$$p(\mathbf{f}|\mathcal{D}, \theta) = \frac{1}{Z} p(\mathbf{f}) \prod_{i=1}^n \Phi(y_i f_i) \quad (3.3)$$

Following the structure for regression, the predictive posterior over f_* given x_* can be computed by marginalisation, and is given by:

$$p(f_*|\mathcal{D}, \theta, x_*) = \int p(f_*|\mathbf{f}, \mathbf{X}, \theta, x_*) p(\mathbf{f}|\mathcal{D}, \theta) d\mathbf{f} \quad (3.4)$$

Again following the structure for regression, the predictive distribution for y^* is given by marginalisation of the joint $p(y_*, f_*)$ over f_* , shown in Equation 3.5. When the likelihood is approximated Gaussian, the predictive distribution that y belongs to class 1 has the analytical form $p(y_* = +1|\mathcal{D}, \theta, x_*) = \Phi(\mu/\sqrt{1 + \sigma})$ where Φ is the Gaussian CDF.

$$p(y_* = +1 | \mathcal{D}, \theta, x_*) = \int p(y_*, f_* | \mathcal{D}, \theta) d\mathbf{f}^* = \int p(y_* | f_*) p(\mathbf{f}_* | \mathbf{f}, \mathcal{D}, \theta, x_*) d\mathbf{f}_* \quad (3.5)$$

Unfortunately, none of the above objects of interest are analytically tractable. It is therefore necessary to approximate these objects. In the literature, Expectation Propagation (EP) is widely regarded as the state-of-the-art approximation for GPCs compared to alternatives such as Variational or Laplace approximations [127, 128]. Note that these are empirical findings as EP has no formal proof of convergence, unlike variational approaches. Importantly EP also lends itself well to sparse approximations, a necessary feature in the real world usage of GPs. The rest of this chapter will build gradually from a primer in the fundamental theory of expectation propagation to its application to sparse GPs through the Power Expectation Propagation framework that the rest of this thesis leverages. This chapter will also demonstrate and discuss why PEP is the approximation theory of choice in section 3.3.

3.2 Expectation Propagation

This section aims to provide a more approachable and intuitive introduction to Expectation Propagation (EP) than is commonly found in the literature. An interested reader is referred to [130] for a comprehensive and rigorous overview of EP.

Expectation Propagation (EP) [131] is a general-purpose framework for approximating distributions p which are intractable or otherwise too difficult to compute. The distribution p must be factorisable as:

$$p(x) = \prod_i t_i(x) \quad (3.6)$$

Where $t_i(x)$ is known as a *site*. The goal is to approximate p with $q \in \mathcal{F}$, where $\mathcal{F}$ is the set of exponential families. EP works by visiting each site in turn, and approximating it with $\tilde{t}_i(x)$, which intuitively gives q the form:

$$q(x) = \prod_i \tilde{t}_i(x) \quad (3.7)$$

As is common in machine learning literature, the local approximation is achieved by minimising a KL divergence (Equation 3.9) between the target and approximating distribution. In EP this step is known as projection. The following derivation is written in terms of p and q in order to improve legibility and maintain generalisability for later reference, but will apply exactly to $t(x)$ and $\tilde{t}_i(x)$. Interestingly, the KL divergence in EP is the reverse of that used in variational inference. EP's KL is mass-covering, encouraging the approximation to capture all regions of significant posterior probability, while the variational KL is mode seeking, focusing on the highest-probability regions and often ignoring smaller modes, resulting in under-estimated variance. As will be shown in section 3.3, this has significant ramifications for the prediction. Mode seeking behaviours typically lead to overconfident predictions in classification. When data is noisy, such overconfidence is highly problematic and can undermine the usefulness of the returned probability, as illustrated in section 3.3.

$$\text{proj}(p) = \arg \min_{q \in \mathcal{F}} \text{KL}(p||q) \quad (3.8)$$

$$\text{KL}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx \quad (3.9)$$

Since the only free parameters are that of the site approximation $q(x)$, the only way to minimise the objective is to take derivatives of it w.r.t. the parameters of q and then set this equal to 0. This task is greatly simplified by the standard properties of the exponential family to which q belongs. While the exponential family encompasses most common continuous and discrete distributions, this thesis explicitly chooses the Gaussian distribution for q so it is used here as the motivating example.

A probability distribution belongs to the exponential family if its density can be expressed in the canonical form:

$$q = \exp(\theta^T \phi(x) - \mathcal{Z}(\theta)) \quad (3.10)$$

The term $\mathcal{Z}(\theta)$ is the log-partition function, which serves as the normalization constant ensuring the distribution integrates to 1, and is given as $\mathcal{Z}(\theta) = \log \int \exp(\theta^T \phi(x)) dx$. The term θ is the vector of natural parameters. Natural parameters offer an alternative characterization to standard moment parameters (which for a Gaussian are the mean (μ) and variance (σ^2)). One of the key advantages of representing a distribution in its natural form is that multiplying two distributions simplifies to an addition of their natural parameters. This will be key later.

The term $\phi(x)$ denotes the vector of sufficient statistics, which captures all the information the data contains about the underlying parameters. For a 1D Gaussian distribution, the sufficient statistics are $\phi(x) = [x, x^2]^T$. A key property of the exponential family is that the expectation of these sufficient statistics with respect to $q(x)$ yields the moment parameters η , as does the gradient of the log partition function with respect to the natural parameters:

$$\eta = \mathbb{E}_q[\phi(x)] = \frac{d}{d\theta_q} \mathcal{Z}(\theta) \quad (3.11)$$

To illustrate this for the Gaussian case, evaluating the expectation of the first sufficient statistic recovers the standard mean:

$$\eta_1 = \mathbb{E}_q[x] = \int x q(x) dx = \mu \quad (3.12)$$

Armed with this information and returning now to the task at hand (minimising the KL divergence), expand out the KL divergence using the exponential form of q , given in Equation 3.10:

$$\mathbf{KL}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx \quad (3.13)$$

$$= \int p(x) \log(p(x)) dx - \int p(x) \log(\exp(\theta^T \phi(x) - \mathcal{Z}(\theta))) dx \quad (3.14)$$

$$= \mathbb{E}_{p(x)}[\log(p(x))] - \theta^T \mathbb{E}_{p(x)}[\phi(x)] + \mathcal{Z}(\theta) \quad (3.15)$$

Now take the derivative w.r.t the parameters of q_i :

$$0 = \frac{dKL}{d\theta_q} = -\mathbb{E}_{p(x)}[\phi(x)_q] + \frac{d}{d\theta_q} \mathcal{Z}(\theta) = -\mathbb{E}_{p(x)}[\phi(x)_q] + \mathbb{E}_{q(x)}[\phi(x)_q] \quad (3.16)$$

Therefore, the objective is minimised when $\mathbb{E}_{p(x)}[\phi(x)_q] = \mathbb{E}_{q(x)}[\phi(x)_q]$. Since one of the central features of the exponential family is that $\eta = \mathbb{E}_{q(x)}[\phi(x)_q]$, it is therefore intuitive to say that the new parameters of q will be $\eta_q^{\text{New}} = \mathbb{E}_{p(x)}[\phi(x)_q]$. In even simpler terms; if q is chosen to be a Gaussian, calculate the mean and variance of $t_i(x)$ and set the mean and variance of $\tilde{t}_i(x)$ equal to the calculated values. This is known as moment matching.

Calculating moments is not always simple if $t_i(x)$ is complicated, however this difficulty can be addressed in two ways. The first is that the integrals will be one dimensional, and that usually some quadrature is available for these integrals to a reasonable degree of accuracy. The second is that some tractable moment matching equations fall out when the target of approximation is a Bayesian posterior, as shall be shown shortly.

Stopping here seems sensible. The KL divergence has been minimised at each site, what else is there left to do? What has been laid out so far is known as *Assumed Density Filtering* (ADF). There are however, significant shortcomings to ADF. An examination of the update maths $\eta_{q_i}^{\text{New}} = \mathbb{E}_{p(x)}[\phi(x)_{q_i}]$ makes it clear that the ADF update is entirely local to the site being approximated. This local approximation is problematic because objects of interest such as posteriors are often highly coupled between sites, thus leading to poor approximations of the true global distribution.

EP is an extension of ADF, as shown in [131]. The central extension lies in the

idea of a cavity distribution, which can be thought of as a representation of the rest of the distribution *without* the site being examined, i.e. $q(x)$ with $q_i(x)$ removed. When $q(x)$ and $\tilde{t}_i(x)$ are Gaussian, this division is simple in the natural parameters: $\theta^{\setminus i} = \theta - \tilde{\theta}$.

$$q^{\setminus i}(x) = q(x)/\tilde{t}_i(x) \quad (3.17)$$

Then instead of minimising the KL divergence between $t_i(x)$ and $\tilde{t}_i(x)$, EP minimises between the two while holding $q^{\setminus i}(x)$ fixed, as shown in Equation 3.18. This minimisation is known as the projection step, and is denoted with the proj as shown in Equation 3.18, and will be referred to as such throughout the thesis. Here the product $t_i(x)q^{\setminus i}(x)$ defines the tilted distribution, denoted $\hat{P}(x)$. Note that the idea behind the moment matching introduced earlier still holds true, and an interested reader can expand out the KL divergence below to find that the new moments of projection will be equal to the moments of the tilted distribution. EP requires a further “division” by the cavity distribution to recover the new site, but again in natural parameter space this is very simple.

$$\tilde{t}_i(x)^{new} = \arg \min \mathbf{KL}(t_i(x)q^{\setminus i}(x) || \tilde{t}_i(x)q^{\setminus i}(x)) \quad (3.18)$$

$$= \text{proj}[t_i(x)q^{\setminus i}(x)]/q^{\setminus i}(x) \quad (3.19)$$

It should hopefully be clear that while the projection step still achieves the same thing after the division of the cavity, the shortcomings of ADF have been overcome. It is now possible to iterate over the current global best guess, since it is contained in the minimisation. The projection also contains global information, making EP a global approximation, rather than a local one like ADF.

Since q is a member of the exponential family ($q \in \mathcal{F}$), the tilted distribution $\hat{P}(x)$ remains in a closely related form called a *tilted exponential distribution*. This preserves many of the properties of $q^{\setminus i}(x)$ (such as its parametric structure) while shifting it to incorporate information from the true site $t_i(x)$. This structure is useful because the moments of the tilted distribution can be expressed very simply, as shown in Equation 3.20, where $Z_i = \int q^{\setminus i}(x)t_i(x)dx$ is the normaliser of the tilted

distribution $\hat{P}(x)$. Note here that $\log Z_i$ will typically not be available in closed form, but since it is a one dimensional integral, quadrature and other simple numerical approximations abound. When q is chosen Gaussian, this guarantees that $\log Z_i$ will be approximatable with a Gaussian-Hermite quadrature.

$$\eta_{new} = \mathbb{E}_{\hat{P}(x)}[\phi(x)_q] = \frac{d}{d\theta^{\setminus i}} \log Z_i + \eta^{\setminus i} \quad (3.20)$$

To tie everything together, Algorithm 1 shows the EP algorithm, and illustrates how it is iterated over.

Algorithm 1 EP algorithm for approximating $\prod t_i(x)$

For each i point, calculate:

- 1: **Cavity:** $q^{\setminus i}(x) = q(x)/\tilde{t}_i(x)$ In natural parameters: $\theta^{\setminus i} = \theta - \tilde{\theta}$
- 2: **Projection:** $\eta_{new} = \mathbb{E}_{\hat{P}(x)}[\phi_q(x)] = \frac{d}{d\theta^{\setminus i}} \log Z_i + \eta^{\setminus i}$, and pick $Q^{new} \in \mathcal{F}$
- 3: **Update:** Replace $\tilde{t}_i(x)$ with: $\tilde{t}_i(x)^{new} = Q^{new}/Q^{\setminus i}$ in natural parameters: $\tilde{\theta}^{new} = \theta^{new} - \theta^{\setminus i}$

Repeat until convergence.

Algorithmically, the extension of EP to Power Expectation Propagation (PEP) is trivial. The cavity is now calculated by dividing the global approximation by the site approximation raised to the power of α :

$$q^{\setminus i}(x) = q(x)/\tilde{t}_i(x)^\alpha \quad (3.21)$$

This results in the new factor being calculated as shown in Equation 3.22. Note that with the factor raised to a power, the KL divergence minimisation is equivalent to an alpha divergence minimisation instead [132].

$$\tilde{t}_i(x)^{new} = (\text{proj}[t_i(x)q^{\setminus i}(x)]/q^{\setminus i}(x))^{1/\alpha} \quad (3.22)$$

Power EP was originally motivated in terms of simplifying projection step, as certain power terms can significantly simplify the resulting projection [132]. It will also be discussed in the following section how the introduction of the α power creates some stability in the update step. By itself, PEP is a useful tool for approximating complicated distributions, but only really comes into play later, when PEP is applied

to sparse GPs.

3.2.1 EP for Bayesian posteriors

In order to introduce EP as intuitively as possible, the introduction was done for a general class of factorisable distributions. This section briefly explains how to apply EP to a Gaussian Process posterior, as well as introducing the marginal likelihood approximation that arises naturally. Note that this can be adapted to any Bayesian posterior. Assuming an intractability in the likelihood, the target of approximation of choice here is the joint distribution (unnormalised posterior) because it has a simple structure and contains the marginal likelihood. The unnormalised GP posterior is:

$$p(f|y, x) \propto p(f)p(y|f) \propto p(f) \prod p(y_i|f_i) \quad (3.23)$$

Thus the approximation q is given as:

$$q(f) \approx p(f) \prod_{i=1}^n \tilde{t}_i(x) \quad (3.24)$$

The cavity would be calculated as:

$$q^{\setminus i}(f) = q(f)/\tilde{t}_i(x) \quad (3.25)$$

And thus projection also happens as normal:

$$\tilde{t}_i(x)^{new} = \text{proj}[t_i(x)q^{\setminus i}(f)]/q^{\setminus i}(f) \quad (3.26)$$

Recall equation Equation 3.20 for the shortcut to moment matching. Recall that in this case, since $q^{\setminus i}(f)$ is Gaussian, the 1D integral can be approximated with Gauss-Hermite quadrature [133].

The marginal likelihood as expected is: $p(y|X) = \int \prod_{i=1}^n t_i(x)p(f|x)dx$. Since the approximation for $t_i(x)$ has already been calculated ($\tilde{t}_i(x)$), the approximation for

$p(y|x)$ arises very naturally. Before simply plugging in the approximated sites, it is important to match the normalisation constants in order to make the correct normalisation of $\tilde{t}_i(x)$ explicit. Consider the two normalisation constant of interest, shown in Equation 3.27, where Φ denotes the log normaliser of the denoted distribution.

$$Z_i = \mathbb{E}_{\theta^{\setminus i}}[t_i(x)] \quad (3.27)$$

$$\tilde{Z}_i = \mathbb{E}_{\theta^{\setminus i}}[\tilde{t}_i(x)] = \exp(\Phi(\theta) - \Phi(\theta)^{\setminus i}) \quad (3.28)$$

Since the goal is simply to match the two normalisation constants (and there is nothing left to control, a new parameter C_i is added such that $Z_i = C_i \tilde{Z}_i$). Thus resulting in:

$$\log C_i = \log Z_i - \Phi(\theta) + \Phi(\theta)^{\setminus i} \quad (3.29)$$

Thus the log marginal likelihood arises very simply as:

$$L = \log \int \exp\left(\sum_{i=1}^n \log C_i + \theta^T \phi(x) - \Phi(\theta^{(0)})\right) \quad (3.30)$$

$$= \sum_{i=1}^n \log C_i + \Phi(\theta) - \Phi(\theta^{(0)}) \quad (3.31)$$

Note that the log marginal likelihood depends on both the approximate sites and the parameters. It is thus necessary to assume that the parameters are smoothly differentiable in an area around the current sites, and alternate taking steps in the log likelihood and in the posterior approximation. This is simple to implement, but certainly lacks the elegance of the variational ELBO.

For a GP, the derivatives w.r.to the parameters of the log likelihood exist in closed form, however this work has found it sufficient (and far more flexible) to use automatic differentiation to approximate these instead. To recap, the EP algorithm for approximating a GP is shown in Algorithm 2.

Algorithm 2 EP algorithm for approximating $p(y) \prod p(x_i|y_i)$

For each **i** point, calculate:

- 1: **Cavity:** $q^{\setminus i}(y) = q(x)/\tilde{t}_i(x)$
- 2: **Projection:** $\eta_{new} = \mathbb{E}_{\hat{P}(x)}[\phi_q(x)] = \frac{d}{d\theta^{\setminus i}} \log Z_i + \eta^{\setminus i}$, And pick $Q^{new} \in \mathcal{F}$
- 3: **Update:** Replace $\tilde{t}_i(x)$ with: $\tilde{t}_i(x)^{new} = Q^{new}/Q^{\setminus i}$

Once each site has been visited, take a step in the hyperparameter space via $\frac{d}{d\theta} \sum_{i=1}^n \log C_i + \Phi(\theta) - \Phi(\theta^{(0)})$.

Repeat until convergence.

As is always the case, there is no free lunch. Other algorithms such as variational ones perform better in regression and have seen much wider adoption and implementation. Crucially EP has no formal proof of convergence, and can be numerically unstable. Practitioners have found that numerical instability can be efficiently dealt with via parallel updates instead of sequential ones, as well as “damping” the updates by only taking a fraction of the new site instead of all of it. It has also been found that PEP has inherent stability benefits, mostly by keeping the variance in the cavity small [134]. Some theoretical work does underpin EP. Numerical stability is guaranteed when the factors are log concave (as in GP regression for instance) [134].

3.2.2 Power EP for Sparse Gaussian Processes

As has previously been discussed, a key problem with the use of GPs is their computational scaling cost of $(\mathcal{O}(n)^3)$. This is a universal problem across GP usage, as small datasets do not typically contain enough information to capture the underlying system to a sufficient degree for meaningful and accurate predictions. To tackle this problem this work previously examined the use of Sparse Variational approximations, however since the aim here is to use EP to better classify data, a sparse EP framework is needed. This section introduces the sparse PEP framework from [55] and [135]. While the original derivation is mostly followed, small numerical stability updates are introduced for parallel updating.

There is an important clarification to be made here in that the rest of this thesis focuses on classification, which requires expensive labelled data. It could be convincingly argued that the models are unlikely to bump into real computational challenges given these expense constraints on dataset size, but this is not correct. A practitioner will have an unpleasant time with GPs over a thousand data points, well

within the size of labelled data that is feasible in an engineering context. Further, the goal of SHM machine learning is often fast or online predictions, adding another reason to sparsify the GP. Sparse approximations are valid even in the medium data density environments that this thesis operates in, especially under the lens of SHM where online computations might be crucial.

As previously in this chapter, the target of approximation is the joint distribution, written as:

$$p(\mathbf{f}|\theta, \mathbf{y}) = p(\mathbf{f}, \theta) \prod_{i=1}^n p(y_i|f_i, \theta) \quad (3.32)$$

As mentioned in subsection 1.3.1, all inducing point sparse GP methods aim to use inducing points to sparsify the posterior over the latent function by expressing it through a smaller subset of values (the inducing points u). Note that while inducing points can correspond to observed data, they do not have to be. They are often treated as virtual pseudo inputs that lie on the latent function. The structure of approximation, and the way in which sparse methods project the real data onto the latent function through these inducing points varies, but the idea remains the same. Variational Free Energy methods [136] will do this projection via a KL divergence between the true posterior and the variational distribution over the inducing variables. Sparse methods that induce sparsity in the prior instead such as FITC and PITC methods [52], [137] do this projection by assuming that all dependencies between datapoints are mediated entirely by the inducing variables, approximating the full prior as a product of conditionals $p(f_n|u)$, which sparsifies the latent function by treating the training outputs as conditionally independent given the inducing points.

The PEP framework introduced by Bui et al. [55] is no different in its aims, although it is perhaps uniquely intuitive. Importantly, it unifies all inducing point sparsifying frameworks through the EP framework.

The core EP approximation behind the PEP framework is as covered in the previous section (with the addition of the power factor, as shown in 3.2). The approximating form can be seen below, where u are the inducing points:

$$p(\mathbf{f}|\mathbf{y}, \theta) = p(\mathbf{f}, \theta) \prod_{i=1}^n p(\mathbf{y}|\mathbf{f}, \theta) \approx p(\mathbf{f}, \theta) \prod_{i=1}^n \tilde{t}_i(u) = q(f|\theta) \quad (3.33)$$

Here $q(f)$ is chosen to have the variational structure $q(f) = p(f_{\neq u}|u)q(u)$. The crucial step for sparsification comes in the moment matching step: when $\mathbf{KL}(\hat{P}(f)||q(f))$ is minimised. The expansion of this equation follows exactly as before, however now the result is:

$$\mathbb{E}_{\hat{P}(x)}[\phi(u)] = \mathbb{E}_{q(f)}[\phi(u)] \quad (3.34)$$

That is despite matching the moments over f , the moments only need to match at u , meaning that the PEP parameters stored and updated need only be for u for each f_i , resulting in significant computational savings.

The resulting high level algorithm is shown in Algorithm 3. Note that the log likelihood for PEP remains unchanged from EP other than a division by α . In practise, it is convenient to express the approximating site in terms of natural parameters (which Bui et al. show can be expressed in rank 1 form). For a detailed derivation, the reader is referred to [55].

Algorithm 3 PEP algorithm for approximating $p(f) \prod p(y|f)$

For each $\mathbf{i}$ point, calculate:

- 1: **Cavity:** $q^{\setminus i}(f) = q(f)/\tilde{t}_i(u)^\alpha$
- 2: **Projection:** $\eta_{new} = \mathbb{E}_{\hat{P}(f)}[\phi_q(u)] = \frac{d}{d\theta^{\setminus i}} \log Z_i + \eta^{\setminus i}$, And pick $Q^{new} \in \mathcal{F}$
- 3: **Update:** Replace $\tilde{t}_i(u)$ with: $\tilde{t}_i(u)^{new} = (Q^{new}/Q^{\setminus i})^{1/\alpha}$

Once each site has been visited, take a step in the hyperparameter space via $\frac{d}{d\theta} \frac{1}{\alpha} \sum_{i=1}^n \log C_i + \Phi(\theta) - \Phi(\theta^{(0)})$.

Repeat until convergence.

It was mentioned earlier that (P)EP can be unstable, and that one way of dealing with this is by updating every site approximation in parallel. This is generally true, however this requires considerable swapping between the natural and moment parameters, a famously unstable process. This is sadly unavoidable (it is easier to divide in natural parameter space, but it is also easier to calculate new moments in moment space). Here, a small change to the original derivation involving a QR decomposition to avoid explicit calculation of an inverse of the variance has been

implemented and is strongly recommended. This is highlighted in Appendix A.

The interested reader might be wondering why Power EP is used at all, given that the Power term has made no impact on the derivation. And, apart from the stability benefits mentioned earlier, this is true. The key finding of Bui et al's. seminal work is that when set to 0, the EP approximation collapses to VFE, and when set to 1, a more traditional EP/FITC approximation emerges. This is a genuinely beautiful result that does a great deal in unifying the field of inducing point approximations for Gaussian Processes. For details of how this is true, the reader is referred to the work [55].

3.3 Application of Binary GPC in SHM

Having established the PEP approximation framework, the binary GPC is now applied to two SHM problems classifying damaged and non-damaged states. Further, it is illustrated why the PEP is the approximation theory of choice for the classification setting. The two SHM problems used here have been chosen as standard damage sensitive features practitioners would look to first when examining a structure. All are derived from the Los Alamos bookshelf dataset [138]. The GPC is benchmarked against an SVM with an RBF kernel, a popular and robust choice for classification in the machine learning and SHM community. The three story experimental structure consists of aluminium columns and plates. Each floor is a four degree-of-freedom system, with four columns connecting the top and bottom plates via bolted joints. Damage is simulated via a centre column suspended from the top floor, which induces non-linear behaviour when it contacts an adjustable bumper on the floor below. It is constrained to move in the x direction via linear rails, and environmental changes are simulated with an 87.5% stiffness reduction in one or more columns. The dataset is comprised of roughly 50-50 damaged un-damaged cases, with nine undamaged but environmentally changing classes, and eight damaged classes in increasing severity by decreasing the bumper gap. The division of damage/undamaged classes follows the original paper, and states 1-9 are labelled as undamaged, while 10-17 are labelled damaged.

The bookshelf dataset here is feature engineered to represent two distinct classification

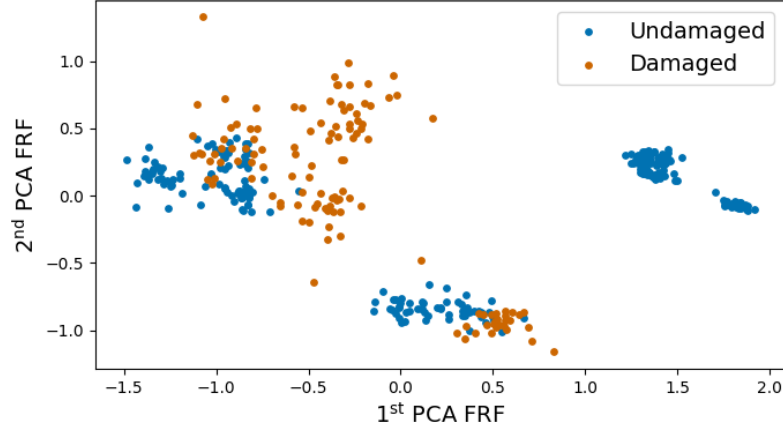
problems. Case study 1 is a damage classification task given the first two dimensions of the PCA of the FRFs of the bookshelf dataset. This presents a noisy, but clustered classification problem, as shown in Figure 3.2a. Case study 2 is again a damage classification task but over the coherences of the FRFs instead [139], shown in Figure 3.2b. This is a separable classification problem, but remains interesting as a problem for active learning (which will be introduced next chapter), and as an easier counterpart to the first problem. The separability found with the coherence method highlights the importance of feature engineering in SHM, turning a complex task into a simple one without reaching for the machine learning. For both problems, the GPC was trained with a Matérn 3/2 kernel with 30 inducing points for case study 1 and 20 for case study 2. Both were initialised via K means clustering. Results were robust against a range of sensible hyperparameter initialisations. Several different hyperparameters for the SVM were tested, with the best results shown.

In order to benchmark the GPC against the SVM, an F_1 score is used as a performance metric, shown in Equation 3.35, where TP are the true positives, FP are the false positives and FN are the false negatives. The F_1 score represents the mean of the precision and recall, and is standard in the literature for binary classification tasks.

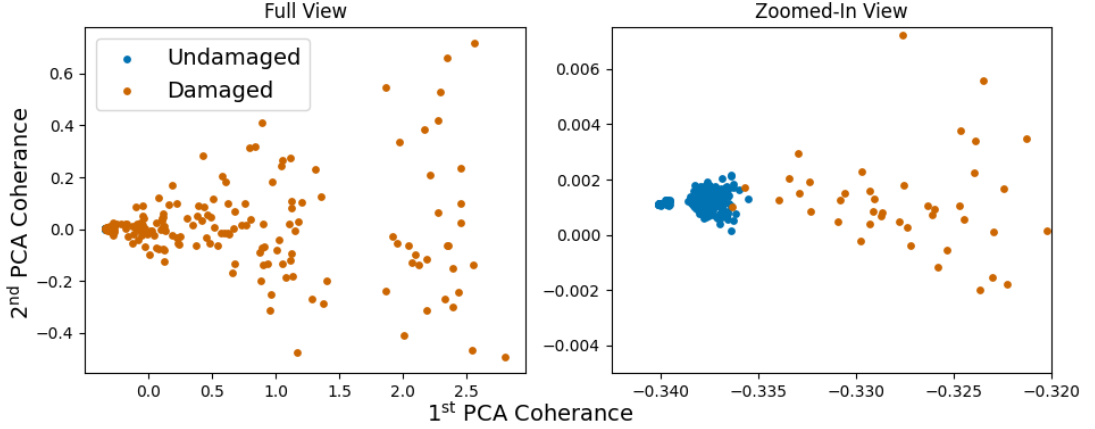
$$F_1 = \frac{2TP}{2TP + FP + FN} \quad (3.35)$$

The results of the GPC and SVM are given in Table 3.1. As expected, the GPC and SVM perform very similarly, with the GPC returning better predictions across the board. Such a result is not surprising, as the GPC is a more descriptive model. Further, the GP is returning much more information in its posterior, which quantifies how certain the model is about its predictions, as illustrated in Figure 3.3. Unlike the SVM, which establishes rigid geometric boundaries, the GPC yields a continuous probability distribution across the feature space. As has been highlighted throughout this thesis, such a probability on the prediction is invaluable to downstream engineering decisions and tasks.

On top of benchmarking the GPC on SHM problems, it is worth examining *why* the PEP is a suitable approximation choice, especially given the abundance of fast, well implemented variational methods such as the Stochastic Variational GP (SVGP). To evaluate this, the Laplace, SVGP and PEP approximations (with PEP $\alpha = 0.9$ to



(a) Case study 1: First two PCAs of bookshelf FRFs



(b) Case study 2: First two PCAs of the FRF Coherences

Figure 3.2: Test points for both case studies

represent EP and remain distinct from variational solutions) are benchmarked on case study 1 (Figure 3.2a).

The approximation schemes are compared against a ground truth MCMC baseline. Specifically, the parameters of the latent predictive posterior (the mean μ and variance σ^2 as well as the final marginal predictive probability p) are compared against their MCMC equivalent. Recall that $p = \Phi(\mu/\sqrt{1+\sigma})$ where Φ is the Gaussian CDF. To ensure a strict comparison of the actual approximation mechanisms, the inducing points are kept static for all three and are found via k means clustering while each approximation optimises its hyperparameters. The results of this are shown in Figure 3.4. A good result here would be the dots lining up with the grey line, implying that for each data point, the parameter is very similar to its ground truth

	GPC	SVM
Case Study 1	0.84	0.77
Case Study 2	0.99	0.99

Table 3.1: Case study results. Case study 1 is the FRF clustering problem illustrated in Figure 3.2a, and case study 2 is the coherence problem illustrated in Figure 3.2b.

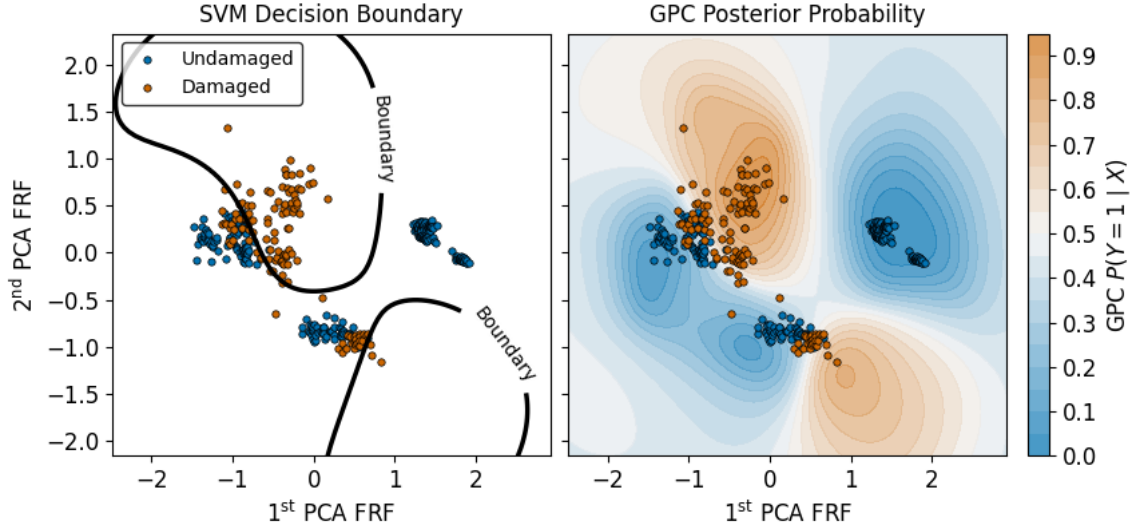


Figure 3.3: Decision boundary of the SVM with RBF kernel (left) compared to the posterior of the GPC (right).

equivalent. This comparison methodology mirrors the benchmarking tests performed on a range of approximations by [127]. To understand how these marginals track to predictive performance, the F1 score and predictive log likelihood (introduced in Equation 2.9 as a measure of how well the uncertainty of a model captures the test data) of each approximation are compared against each other and the MCMC ground truth in Figure 3.5. Here, the higher the score the better the model has performed for both metrics.

Figure 3.4 reveals several interesting findings. The first is that the Laplace approximation performs badly, confirming results shown in [127]. This poor result for Laplace is corroborated by the performance metrics shown in Figure 3.5, where it performs the worst of any approximation. Perhaps more surprisingly, the SVGP performs quite similarly to the Laplace approximation. This behaviour stems from SVGP's objective function, which minimizes the mode seeking KL divergence ($KL(q \parallel p)$) (opposite to the EP KL divergence) which typically seeks the mode and underestimates the variance. As expected, this results in a slightly better mean and predictive

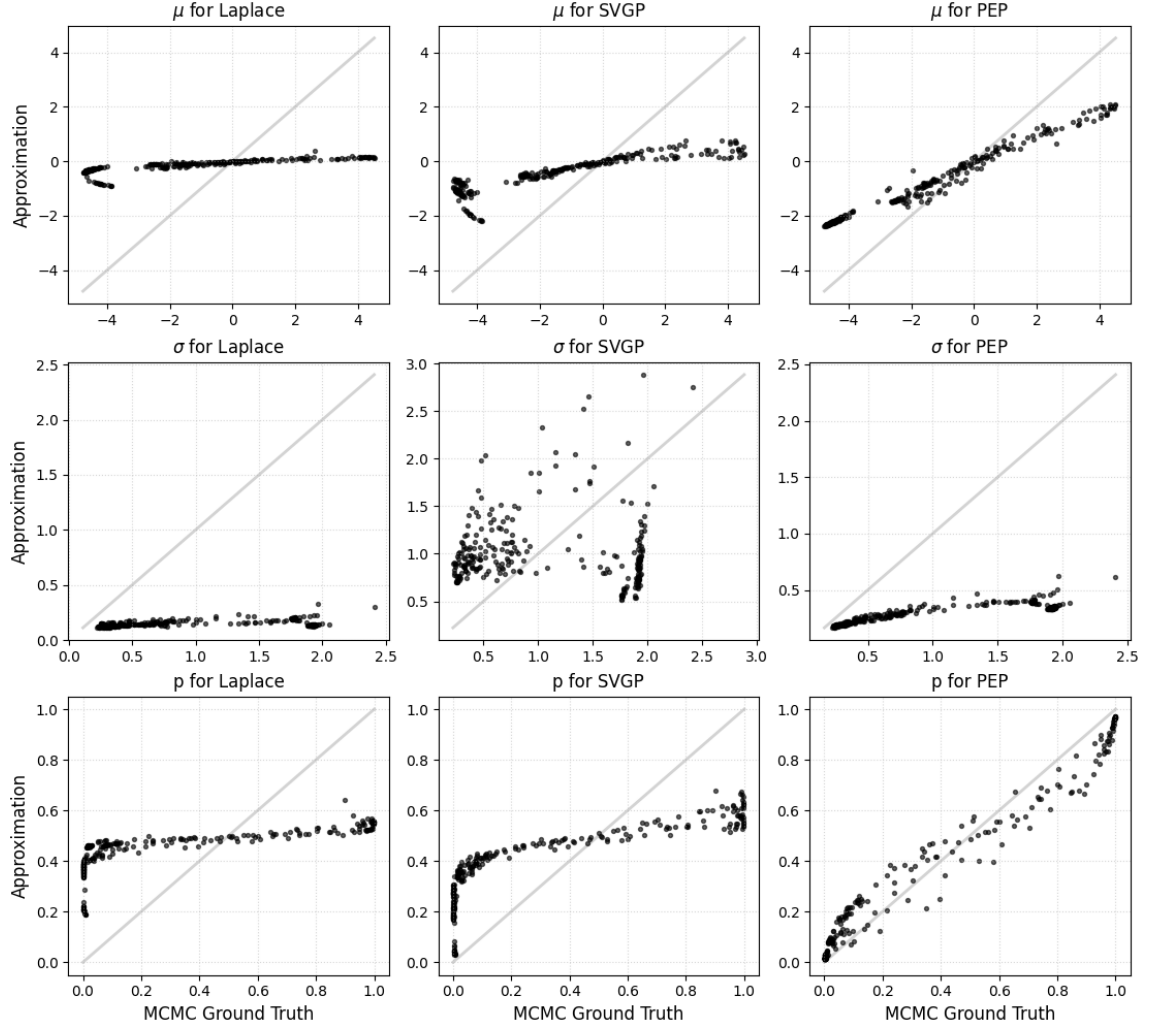


Figure 3.4: Marginals of different GPC approximation schemes plotted against an MCMC GPC. PEP had $\alpha=0.9$ such to be closest to EP and furthest from SVGP. μ is the predictive mean, σ is the predictive variance and $p = \Phi(\mu/\sqrt{1+\sigma})$ is the predictive probability that any label belongs to class 1.

probability fit than Laplace with significantly underestimated variance (note the large clump of σ points under the grey line). Finally, the PEP seems to behave well. The mean and variance margins are linearly out of scale with the MCMC ground truth implying a different optimum for the hyperparameters found, but the predictive probability loosely resembles the MCMC ground truth. This result is reflected in Figure 3.5, where PEP performs the best in both F1 score and PLL of all the approximations. This result again corroborates [127] which with the same experiment (although on a different dataset) and many more show that PEP tends to a good choice for classification tasks.

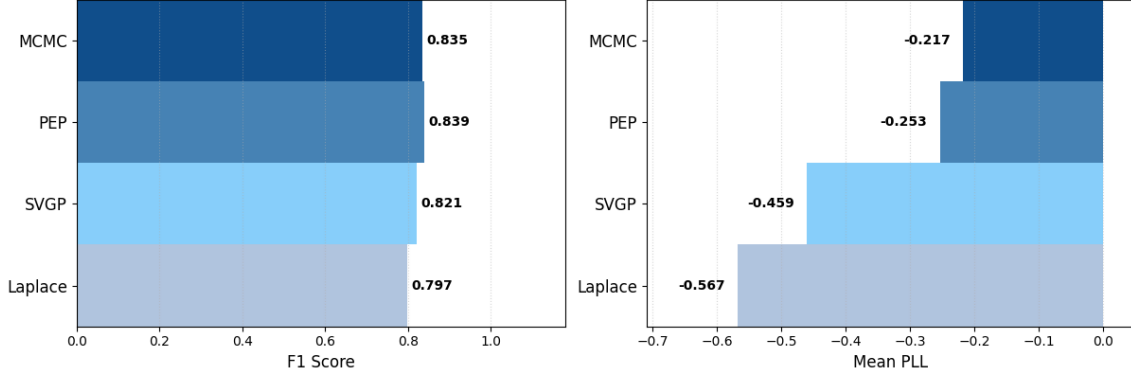


Figure 3.5: Bar chart comparing the F1 score (left) and the Predictive Log Likelihood (right) of all approximation schemes and the MCMC ground truth. For both metrics, a higher score is better.

While the results presented here would indicate that PEP would always be the best approximation choice for GPC, this is not the case. There is no free lunch, and there will always be cases where it is outperformed by other approximations. Crucially though the PEP framework allows for tuning between most inducing point approximations, allowing the user to manually choose which would best fit the current problem. While some users may be concerned that unlike variational approximations, the PEP has no guarantee of convergence, it is important to remember that such a guarantee is only to local optima, and is not a guarantee of a good result.

3.4 PEP for multi-latent GPs

The GPs presented so far have been defined with a single latent function, designed to model a single input-output relationship. However, many real world problems are more complex and require richer representations. A GP can instead model multiple latent functions (each representing a distinct process of interest), and define their interaction through a joint likelihood over these latent functions. These models are known as chained GPs [104]. This section presents a general and intuitive derivation showing how EP can be extended to approximate these models. While prior work has applied PEP to multi-class GPs [129], it is limited to classification and does not follow standard notation. In contrast, the derivation here is fully general, covering all chained GPs (of which multi-class classification is only a subset), and provides a consistent, field-standard formulation, enabling future work with PEP

approximations for chained GPs. For brevity, EP/PEP is simply referred to as PEP, though everything here applies equally to a pure EP implementation.

To illustrate the PEP extension to multi-latent likelihoods, this section walks through two thematic GP examples. The first being multi-class classification, and the second being a heteroscedastic GP. Note that the inclusion of the Heteroscedastic GP gives this chapter the dubious honour of novelty for the derivation of a PEP approximation to the SHGP.

The multi-latent likelihoods of interest here are of the form, $p(y_i|f_{n_1}, \dots, f_{n_m})$. Each latent function has their own independent prior, such that their form and parameters are not necessarily uniform across latent functions: $f_{n_1} \sim \mathcal{N}(0|K_{n_1}), \dots, f_{n_m} \sim \mathcal{N}(0|K_{n_m})$.

The likelihood of a multi-class classification GP is given in Equation 3.36, where $\mathcal{H}$ is the heaviside step function and $\mathbf{f} = (\mathbf{f}_1, \dots, \mathbf{f}_k)$ is the vector of latent functions representing every label. A more robust extension with the introduction of ϵ to deal with noise is introduced in Equation 3.37, where ϵ represents the probability of choosing y_i from the labels and c is the number of classes. Quite intuitively, it is simply the probability that the latent function associated with the label y_i is larger than all the other latent functions. To keep notation light, this section uses the non-robust version as the motivating example, however the following maths and framework apply equally to the robust likelihood.

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^n \prod_{k \neq y_i} \mathcal{H}(f_{y_i} - f_k) \quad (3.36)$$

$$p(\mathbf{y}|\mathbf{f}) = (1 - \epsilon) \prod_{i=1}^n \prod_{k \neq y_i} \mathcal{H}(f_{y_i} - f_k) + \frac{\epsilon}{c} \quad (3.37)$$

The likelihood of the SHGP is given in Equation 3.38, where $\mathbf{f} = (\mathbf{f}_1, \mathbf{f}_2)$. It states that the likelihood of the GP is normal ($\mathbf{f}_1$) but that its scale parameter ($\mathbf{f}_2$) is modelled by another GP, which is bounded to be positive via an exponent.

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^n \mathcal{N}(y_i|f_{1,i}, \exp(f_{2,i})) \quad (3.38)$$

At the core of this extension to PEP is the approximation of independence in the likelihood. The approximation to the posterior is denoted $q(f)$, and is shown in Equation 3.39. For the multi-class, this is intuitive as it inherently carries this approximation. This is not so for the heteroscedastic, and instead the approximating form will mimic the relationship of the true likelihood during PEP training.

$$q(f) = \prod_{k=1}^c \mathcal{N}(0|\mathbf{K}_k) \prod_{k=1}^c \prod_{i=1}^n p(y_i|f_{k_i}) \quad (3.39)$$

PEP aims to approximate each $p(y_i|f_{k_i})$ individually with its respective site $\tilde{t}(u)_{k,i}$, looping over i (as usual), as well as k . This is actually very typical for EP problems with multiple targets of approximation, and as discussed previously was one of the original motivators of the power term in PEP. The algorithm is shown in Algorithm 4. It is clear that at a high level the functionality is similar to PEP for single GPs. Three differences characterise this extension and require further exploration, discussed below.

Algorithm 4 PEP algorithm for approximating $\prod p(f) \prod \prod p(y|f)$

For each **k**th latent function, calculate:

For each **i** point, calculate:

- 1: **Cavity:** $q^{\setminus i}(f_c) = q(f)/\tilde{t}_{c,i}(u)^\alpha$
- 2: **Projection:** $\eta_{new,c} = \mathbb{E}_{\hat{P}(f)}[\phi_q(u)] = \frac{d}{d\theta^{\setminus i}} \log \tilde{Z}_i + \eta^{\setminus i}$, And pick $Q^{new} \in \mathcal{F}$
- 3: **Update:** Replace $\tilde{t}_{c,i}(u)$ with: $\tilde{t}_{c,i}(u)^{new} = (Q^{new}/Q^{\setminus i})^{1/\alpha}$

Once each site has been visited, take a step in the hyperparameter space via $\frac{d}{d\theta} \frac{1}{\alpha} \sum_{k=1}^c \sum_{i=1}^n \log C_{k,i} + \Phi(\theta) - \Phi(\theta^{(0)})$.

Repeat until convergence.

The first difference is the cavity step. For any latent function f_c at any point i , the cavity will take the form shown in Equation 3.40. That is, the cavity only changes over f_c . It follows that the only parameters that will be updated are also only over f_c , so PEP parameters only need be created and updated for the f_c being visited, rather than for all f . The rest of the cavity ($q(f_{k \neq c})$) can be treated as a constant throughout the process, and is anyway a product of Gaussian's that plays nicely with the Gauss-Hermite quadrature introduced for the projection step. Note this is a general form for the cavity of all multi-latent likelihoods.

$$\begin{aligned}
q^{\setminus i}(f_c) &= \frac{q(f)}{\tilde{t}_{c,i}(u)^\alpha} \\
&= \prod_{k \neq c}^m p(f_k | \theta) \prod_{k \neq c}^m p(\mathbf{y}_i | \mathbf{f}_{\mathbf{k}_i}) \frac{p(f_c | \theta) p(\mathbf{y}_i | \mathbf{f}_{\mathbf{c}_i})}{\tilde{t}(u)_{c,i}^\alpha} \\
&= q^{\setminus f_{c,i}}(f) q(f_{k \neq c})
\end{aligned} \tag{3.40}$$

The second difference is the projection step, which requires calculating the normaliser of the tilted distribution $\tilde{Z}_i = \int q^{\setminus i}(f_c) p(\mathbf{y}_i | \mathbf{f}_{\mathbf{k}_i}) d\mathbf{f}_i$. As a general rule, the amount of dimensions this integral requires depends on the factorisation of the likelihood. First examine the heteroscedastic example (Equation 3.41). Since it contains a joint likelihood over both latent functions, so too must $\tilde{Z}_i$ require a joint integral (and resulting quadrature) over both latent functions.

$$\begin{aligned}
Z_i &= \int q^{\setminus i}(f_c) p(\mathbf{y}_i | \mathbf{f}_{\mathbf{k}_i}) d\mathbf{f}_i \\
&= \int \int q^{\setminus f_{c,i}}(f) q(f_{k \neq c}) \mathcal{N}(\mathbf{y}_i | \mathbf{f}_1, \exp(\mathbf{f}_2)) df_1 df_2
\end{aligned} \tag{3.41}$$

In contrast, the likelihood of the multi-class factorises over $\prod_{k \neq y_i}(\cdot)$. This means that the Z_i integral factorises into $k - 1$ double integrals (Equation 3.42). To explain this factorisation intuitively, the cavity notation $q^{\setminus f_{c,i}}$ has been dropped. Simply drop the relevant calculated cavity into whatever latent function is currently being visited. Nested quadrature approximations are still computationally feasible in low dimensions, meaning that the necessary double quadrature for Equation 3.42 and Equation 3.41 are feasible and accurate to a reasonable degree.

$$\begin{aligned}
Z_i &= \int q(f_{y_i}) \prod_{k \neq y_i} q(f_k) \prod_{k \neq y_i} \mathcal{H}(\mathbf{f}_{\mathbf{y}_i} - \mathbf{f}_{\mathbf{k}}) d\mathbf{f} \\
&= \int \int q(f_{y_i}) \prod_{k \neq y_i} \int q(f_k) \mathcal{H}(\mathbf{f}_{\mathbf{y}_i} - \mathbf{f}_{\mathbf{k}}) d\mathbf{f}_k df_{y_i}
\end{aligned} \tag{3.42}$$

The third difference is the log marginal likelihood approximation, given in Equ-

tion 3.43 in a general form. Here $\Phi(\theta)$ is the log normaliser of the global approximate posterior, $\Phi(\theta^{(0)})$ is the log normaliser of the global prior and $C_{k,i} = \log Z_{\text{tilted},i} + \Phi(\theta^{\setminus k,i}) - \Phi(\theta_{k,i})$ is the unique normalising constant for each latent function. Since the approximating form $q(f)$ is a product of Gaussians (shown in Equation 3.39), it has a closed form solution, and therefore so does its log normaliser $\Phi(\theta)$. This makes the calculation of the marginal likelihood approximation no more difficult than for a standard GP-PEP. To simplify things further, the entire marginal likelihood factorises over k , since both the global log normalisers of the prior and posterior factorise also over k , as made explicit in Equation 3.39. Thus the hyperparameters of each k^{th} latent function can be optimised over the terms of the marginal likelihood that are over that k^{th} latent function, rather than over the whole marginal likelihood

$$L = \frac{1}{\alpha} \sum_{k=1}^c \sum_{i=1}^n \log C_{k,i} + \theta^T \phi(x) - \Phi(\theta^{(0)}) \quad (3.43)$$

Importantly, the equivalence to variational bounds when $\alpha = 0$ still hold, and after referring to the proof in [55], it is self evident that the above marginal likelihood approximation will recover the variational bound for chained GPs introduced in Saul et al. [104].

When calculating multi-class PEP, some practitioners (Hernandez et al. [129]) only visit the f_{y_i} at each i , leaving all $q(f_{k \neq y})$ to sit at their initialisation of zero. This results in some computational saving in return for some strange behaviour at class boundaries, where a deeply unlikely latent function has the same likelihood as all others. The author strongly recommends that all latent functions be visited in parallel. With modern programming architectures, this additional compute can easily be parallelised.

Both examples above require the calculation of at least one double integral. This is not necessarily true of all multi-latent likelihoods. Some may not factorise at all and require k nested integrals to be computed, while others may even factorise further or benefit from clever marginalisations to reduce down to a single integral or even an analytical form. In the cases presented here, all integrals contain at least one Gaussian in them (and this is true as a general case due to the inclusion of the cavity, which must necessarily be Gaussian), this means that the calculation

can be approximated with a sufficient degree of accuracy via nested Gauss-Hermite quadrature. Computational complexity generally scales linearly with dataset size: $\mathcal{O}(knm^2 + ns^d)$, where k is the number of latent functions, n is the number of data, m is the number of inducing points, s is the number of Gauss-Hermite degrees and d is the depth of nested quadrature required. For the examples covered here, $D = 2$, and thus computational complexity scales linearly with the number of latent function as the quadrature term is negligible. If the likelihood does not factorise nicely, the complexity will come to be dominated by the nested quadrature.

As previously, the theory covered here is relatively high level. It should hopefully be clear that since each latent function is approximated separately, a PEP approximation implementation will look identical to that of a standard GP except the likelihood and marginal likelihood are now chained. This is made explicit in Appendix B.

Before moving on to a more serious examination of the application of multi-class GPC in SHM, this section concludes by validating the heteroscedastic PEP approximation on a synthetic toy dataset. The dataset consists of a sine wave characterized by input-dependent variance, where the noise envelope dynamically modulates with x . As illustrated in Figure 3.6, the predictive variance of the heteroscedastic GP successfully tracks these localized noise fluctuations, validating the PEP approximation for a chained GP.

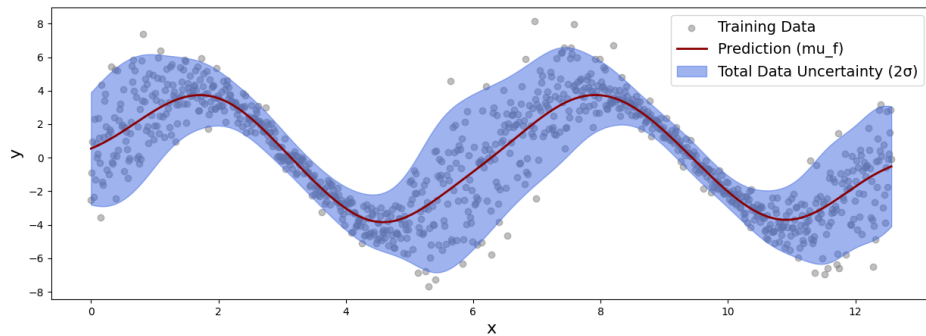


Figure 3.6: Heteroscedastic GP predicting a noisy sine wave

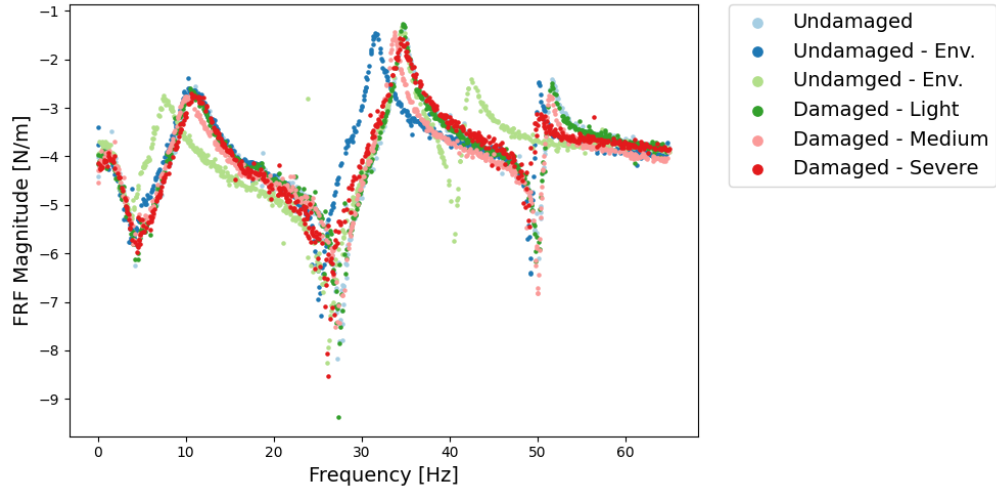
3.4.1 Application of multi-class GPC in SHM

The multi-class GPC is benchmarked against two distinct SHM classification problems. Case study 3 is the FRFs of the Los Alamos bookshelf dataset introduced in section 3.3.

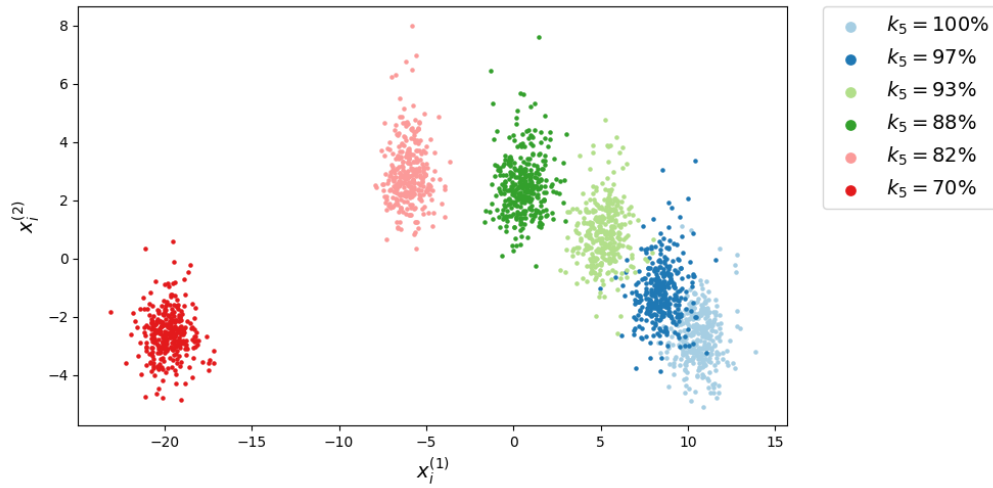
FRFs are standard damage sensitive features in SHM and are often the first point of analysis for practitioners. There are four transducers in this dataset, and the FRFs measured by the first one are shown in Figure 3.7a as an illustration of the problem, “Env.” here indicates undamaged environmental conditions, as detailed in the original work [138]. Although only one is shown, all four FRFs are used as inputs, resulting in a five-dimensional input space; frequency and the four FRF magnitudes at each frequency. The associated structural states at each data point are the classification targets. Note that only six classes of the original 17 are used for this problem, both to enable reasonable computation of the multi-class GPC (which scales linearly with the number of class labels) and to simplify the dataset down to a noisy if separable problem. It can be seen that the problem is separable, especially around the FRF peaks of damage cases, which should come as no surprise with a large part of the environmentally noisy classes removed from the dataset. Case study 4, shown in Figure 3.7b is a synthetic clustering problem simulating an 8-DOF system under white noise excitation, where linear damage is induced by decreasing stiffness of the fifth spring (as indicated in the legend, lower k_5 stiffness corresponds to a higher state of damage). Introduced in Bull et al. [140] the axes are the first two PCA components of the transmissibility between masses one and eight. It provides an interesting clustering problem, with overlapping and noisy class boundaries as well as interesting outliers that some classifiers might struggle to deal with.

The results are shown in Table 3.2. Here a macro F1 score is used, which is the averaged F1 score for each class. The Macro F1 score was chosen here since it weighs all classes equally, and the goal is to benchmark GPC labelling performance across the input space, not just in areas of interest such as damage classes. For case study 3 all latent functions used a Matérn 3/2 kernel, with 150 total inducing points shared between classes. A very similar set up was used for case study 4. Each latent function again had a Matérn 3/2 kernel and this time only 50 inducing points were used since this is a simpler classification problem. For both experiments hyperparameter optimisation was stable across multiple initialisations, and inducing points were initialised via clustering. Again the SVM was initialised with a range of hyperparameters, with best results shown.

The clustered problem sees strong performance from both classifiers. Both classifiers are able to deal with the limited class overlap relatively well, and extrapolate class boundaries well to deal with outliers. Interestingly, there is a large gap in



(a) Case study 3: FRFs of the first transducer of the bookshelf dataset



(b) Case study 4: Noisy SHM cluster dataset

Figure 3.7: Test points for both multi-class case studies

model performance for the FRF problem, likely because the probabilistic GPC can better handle the highly noisy class overlaps. These results are significant as they demonstrate that multi-class GPCs are a powerful and expressive classifier fully capable of handling SHM data.

	GPC	SVM
FRFs	0.79	0.68
PCA transmissibility	0.96	0.95

Table 3.2: Case study results. Case study 3 is the raw FRF classification problem in Figure 3.7a, and case study 4 is the synthetic clustering problem in Figure 3.7b.

3.5 Discussion

This chapter has introduced both binary and multi-class Gaussian Process Classification as a classification algorithm of interest for SHM data, implemented via the SOTA approximation scheme PEP. Binary GPC was shown to perform well on multiple SHM datasets with comparable or better predictive performance to an SVM. The chapter then showed how to extend PEP to multi-latent GP problems. A novel heteroscedastic PEP approximation was derived, and the multi-class GPC was applied to two SHM problems for the first time.

Establishing the binary GPC as a practical tool in SHM is valuable for several reasons. First, GPCs are grounded in a strong Bayesian framework, allowing for a principled treatment of uncertainty in model predictions. This innate uncertainty quantification which can be readily propagated downstream, and, as will be shown in the next chapter, proves mathematically useful and convenient in active learning.

The introduction of a PEP-based multi-class GPC to SHM is a notable contribution, allowing significant probabilistic progression up Rytter’s hierarchy. By demonstrating strong classification performance across two distinct and common SHM challenges, the model is established as a flexible and effective tool for tackling complex supervised classification tasks in this domain. Further, its inherent ability to quantify uncertainty is again particularly valuable in SHM applications.

This chapter also spent significant time on the PEP approximation, walking intuitively through a simple approximation to complex, multi-latent approximations. This presentation is an important contribution to the literature as many works are theoretically dense, and can thus be difficult to approach for practitioners. Further, this chapter presented a generalised approximation framework for chained GPs, an important step for PEP literature. Second, the generalized approximation framework

for chained GPs presented in this chapter, representing an important contribution to PEP literature that simplifies future work with chained GPs.

The GPC models presented and applied to SHM in this chapter face two critical challenges before they can become fully practical tools in the SHM toolbox. The first challenge is data availability. In this chapter, labelled data was assumed to exist for all structural states of interest. In reality, this assumption rarely holds, and the following chapter addresses this issue through the application of active learning strategies. The second challenge is scalability. Even with sparsification, GPCs do not scale efficiently to large datasets compared to other models. Further, training GPCs is computationally intensive, and some form of adaptation is required to deploy them in the streaming settings typical of SHM applications. This challenge is also explored in the following chapter, where an online PEP framework is extended to generalized likelihoods.

The work presented in this chapter opens several promising avenues for future research in SHM. An immediate avenue is understanding the impact of prior (kernel) selection on GPC performance. The introduction of non-stationary kernels to GPCs could be an especially promising area of research, as discrete class data is often non-stationary across the input space. One of the most important areas of future research is the incorporation of PIML into GPCs. While challenging due to the abstract latent space of GPCs, this could be achieved via physics-informed feature augmentation or multi-output GPs, where a PIML regression branch influences the GPC latent discriminator. That PIML GPCs are a realistic future research direction highlight that the introduction of GPCs to SHM in this chapter create an important foundation, establishing GPCs as powerful probabilistic model class fully capable of tackling SHM problems.

As stated, the next chapter will focus on tackling the two critical challenges facing the use of GPCs in SHM: data availability and online data streams. Active learning will be explored as a solution to the data availability problem, with a focus on how prior knowledge can be incorporated into and guide active learning. It will be shown that good use of expert knowledge can drastically improve convergence rates, thus reducing the amount of new labels necessary. At the same time, the problem of scaling GPCs to online environments will be investigated and a novel solution proposed, ensuring that GPCs can move closer to becoming practical, deployable

tools for SHM applications.

AN EXPLORATION OF ACTIVE LEARNING: THE POWER OF PRIOR BELIEF AND FAST MODEL UPDATING

Chapter Highlights:

- Active learning is introduced as a way to deal with the problem of expensive labels
- The EPIG algorithm is introduced to SHM, and is shown to be an empirical improvement over current methods.
- A novel method of updating GPCs is introduced to avoid expensive retraining at each active learning step

One of the primary challenges facing classification algorithms is the lack and expense of labels. This is particularly pronounced in SHM; where labelling will often require extensive and skilled labour hours (e.g. might necessitate a site visit or an engineer reading a report). This capital and time cost must therefore be minimised in order to maximise the utility of classification methods within SHM. The previous chapter touched on a few methods in which the labelling problem is approached in the literature; namely semi-supervised learning and transfer learning. Both are active

and exciting fields of research with much promise, however this chapter will instead focus on active learning as a simpler, more immediate alternative.

Active learning addresses the challenge of expensive data labeling by identifying which unlabelled instances would be most informative to the current model if labelled. The process is broken down in Figure 4.1. After training an initial small model, the most informative data to label is identified by taking the maximum of the *acquisition function*. The data is added, the model is updated or retrained, and this loop continues until a satisfactory threshold in performance is reached or a labelling budget is used up. The acquisition function is the secret sauce of active learning, and is what differentiates different active learning strategies. When the acquisition function is not specific to a model architecture, the active learning process is model agnostic. This is one of the great strengths of the approach, as it creates a drag and drop framework that allows users to leverage their chosen models strengths without needing to design a separate classifier (unlike in say, semi-supervised learning).

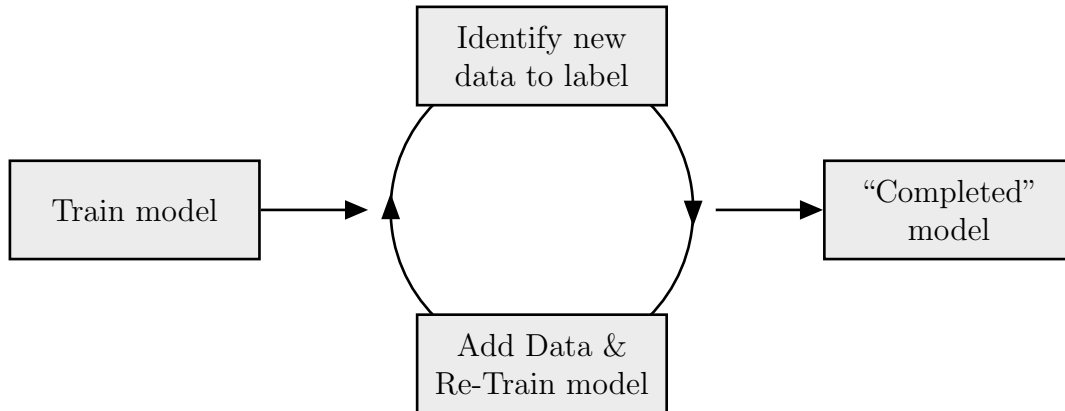


Figure 4.1: The active learning loop

This chapter explores active learning with GPCs as the model of choice, however it is important to highlight that the acquisition functions explored are model agnostic. Instead, the focus on GPCs is for narrative reasons. Firstly, it is necessary to drive home the point made in the previous chapter that the UQ provided by GPs is highly useful and worth the extra effort. Second, it ties in nicely with the second half of this chapter which proposes a novel extension to a fast updating framework for GPs.

There are two types of scenarios an Active Learner can be deployed in: *online active learning* [141], where data arrives in a stream and labeling decisions must be made in real time, and *pool-based active learning* [142], where a fixed set of unlabelled data

is available from the outset. Many acquisition functions apply to both scenarios, and as such this chapter focuses on pool based scenarios for the sake of convenience.

A wide variety of acquisition functions have been proposed in the literature, making strict classification of them challenging. However, broad families can still be identified. The family of interest in this chapter is information theoretic acquisition functions. These rely on fundamental principles from information theory, such as entropy and mutual information to quantify uncertainty to guide the selection of the next evaluation point. Perhaps the most widely used information theoretic active learner is the Bayesian Active Learning by Disagreement (BALD) acquisition function [143], which targets reductions in parameter uncertainty.

The simplest acquisition functions are heuristics (e.g. select the prediction with the greatest entropy/uncertainty [144], [145]). More advanced acquisition functions range from clustering-based approaches (e.g., H. Nguyen et al. [146]) to methods based on adversarial training (e.g., S. Sinha et al. [147]). While active learning has been applied to many classifiers, a surprisingly large number of works focus on GPCs. This is likely because GP posteriors are easy to query and naturally provide uncertainty estimates useful for active learning. Some particularly notable examples of active learning with GPs include Buisson et al. [148], who apply it to dynamic systems; Zhao et al. [149], who propose a gradient-based acquisition function that aims to minimize classification error; and Schreiter et al. [150], who focus on ensuring physical safety during the active learning process.

Active learning in SHM has seen considerably less attention. Feng et al. [151] apply it to defect detection in civil infrastructure, whilst Bull et al. [152] use it in a machining setting. Bull et al. [153] also investigate its use for managing inspections, applying the guided semi supervised clustering process to real bridge data. Finally, Hughes et al. [154] introduce an alternative way of looking at risk in order to incorporate the acquisition function into larger risk calculations.

Active learning remains a relatively recent addition to the field of Structural Health Monitoring (SHM), and there is considerable value in expanding the range of acquisition functions applied in this context. This chapter introduces *Expected Predicted Information Gain* (EPIG) (proposed by Smith et al. [155]) to SHM. EPIG is an information theoretic acquisition function that builds upon earlier approaches such

as BALD. EPIG aims to reduce prediction error over specified “target” distributions of interest, effectively incorporating a prior over current belief of where the most informative data points may lie. This is especially useful in SHM, where damage sensitive features are often highly localised in the input domain (e.g. FRF peaks). EPIG allows the user to leverage the mechanics underlying SHM problems to place a target only where there are the strongest damage signatures.

Comparatively little attention in the literature has been given to the updating step of the active learning loop, despite it being a clear computational bottleneck. This bottleneck is especially pronounced in acquisition functions which require “test” updates to find the most informative point (such as BALD and EPIG). These update challenges are model-specific, and in the case of Gaussian Process Classifiers (GPCs), the options for efficient updates are limited.

One approach involves rank-one Cholesky updates [156], which are efficient but only applicable to GPs with exact posteriors and are restricted to single-point updates, making them unsuitable for GPCs. A second approach leverages variational methods [157], enabling more widely applicable updates through a variational framework. However, it would be a shame to abandon the PEP framework and its associated performance gains in classification. Although Bui et al. [157] proposed a PEP based update method, their derivation applies only to GPs with Gaussian likelihoods. This chapter presents a novel extension of that approach, generalizing it to GPs with arbitrary likelihoods, including GPCs.

This chapter therefore contributes meaningfully to both parts of the AL loop as well as wider GP approximation theory. The chapter will start with introduction to information theoretic active learning before introducing the EPIG algorithm. It will be shown that prior knowledge of SHM problems can be significantly leveraged with the EPIG framework via benchmarks on the previously introduced Los Alamos classification problems. The chapter will then introduce the Sparse Streaming GP (SSGP) framework and the proposed extension for generalised likelihoods. This will finally be benchmarked against other streaming approximations and real updates across a range of problems to illustrate accuracy.

4.1 Do BALD men dream of Electric PIGs (EPIG)?

In many SHM problems, the information desired and the physics of damage sensitivity are highly localized in the input domain. Further, these localisations are understood by well known mechanical principles underlying the systems being modelled. Examples include: FRF peaks, guided wave time windows, anti node regions of specific nodes and bands around the first few bending modes in operational modal analysis. It therefore follows that it is intuitive to restrict prediction to only these areas. However, simply cutting the input domain down to the area of interest is an arbitrary process with significant risks. For instance there will always be data outside the arbitrary cut that influences the decision boundary in the target. Clearly, it would be highly useful to introduce some notion of this target in the input space into acquisition functions used in SHM.

This section will introduce the EPIG [155], a relatively simple acquisition function that enables a user to define a target distribution in the input space and make explicit that the goal is improved prediction over the defined target, rather than the whole input space. This neatly sidesteps the arbitrary cutting issue, allowing all areas of the input space to be checked for informativeness for prediction in the target. EPIG will be comprehensively benchmarked on the SHM problems introduced in the previous chapter using a binary GPC. It will be shown that that using engineering knowledge to direct a target distribution can greatly improve convergence in active learning problems. A parallel can be drawn to feature engineering, where smart feature choice (e.g. the coherences illustrated in Figure 3.2b) can greatly simplify the predictive problem. In practice, thoughtful use of prior knowledge and feature design often outperforms more sophisticated mathematical techniques.

It is worth discussing here that while the notion of a target distribution in SHM is appealing. There are drawbacks; fundamentally the methodology proposed by EPIG is a restriction on exploration of the data space. This restriction can be justified with strong prior knowledge of the problem, and comes with significant benefits such as improved convergence and decreased computational overhead, but also carries the risk of ignoring potentially highly useful areas if prior knowledge turns out incorrect. As such, practitioner discretion is advised.

The derivation for EPIG presented here follows original formulation, and presents an additional discussion on how it might be more efficiently calculated for GPCs. The reader is referred to Figure 4.2 to situate themselves as to where the following section sits within the AL framework, as well as to the original text for a more in depth introduction to EPIG.

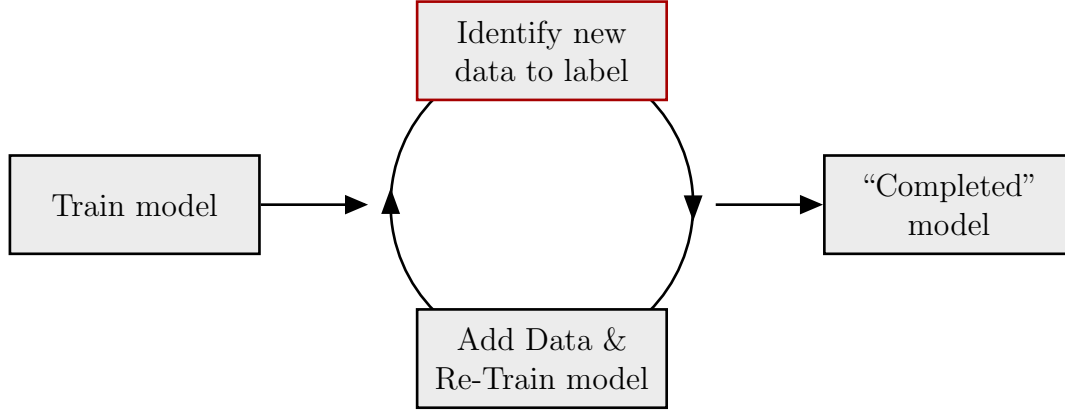


Figure 4.2: Situation of (this) section 4.1 within the AL loop

4.1.1 Information theoretic acquisition

One of the core ideas of information theory is that of Shannon’s entropy [158], which is a measure of the average uncertainty or information content in a random variable’s possible outcomes. For a random variable X with a continuous probability distribution $p(x)$, Shannon’s entropy is defined as $H(X) = - \int p(x) \log p(x) dx$.

Given a posterior $p(\psi|D)$ over some quantity of interest ψ , Information Gain (IG) extends entropy to measure what new data (x_*, y_*) most reduces the uncertainty on the quantity of interest, shown in Equation 4.1. An important note on notation here: (x_*, y_*) denotes both the predictive data and the active learning pool (from which new data can be drawn). This is because the AL pool is often the target of prediction, and the predictive posterior of the GP is explicitly needed to calculate the uncertainty (over ψ) before new data is observed. As usual D is suppressed to lighten notation.

$$\text{IG}_\psi = H[p(\psi)] - H[p(\psi|x^*, y^*)] \quad (4.1)$$

Since y^* is a random variable, it is necessary to take expectations over possible realisations of it given x^* and the current predictive model $p(y^*|x^*)$, as shown in Equation 4.2. This is known as Expected Information Gain (EIG).

$$\text{EIG}_\psi = \mathbb{E}_{p(y^*|x^*)}[H[p(\psi)] - H[p(\psi|x^*, y^*)]] \quad (4.2)$$

Note that BALD (one of the most widely used information theoretic acquisition functions) sets $\psi = \theta$, which targets a gain in information on the parameters. However, since the posterior over the parameters can often be intractable, Houlsby et al. [143] note that the information gain is equivalent to the mutual information, which is symmetric, hence allowing a much more tractable formulation as shown in Equation 4.3.

$$\text{BALD}(x) = \mathbb{E}_{p(y^*|x^*)}[H[p(\theta)] - H[p(\theta|x^*, y^*)]] \quad (4.3)$$

$$= \mathbb{E}_{p(\theta)}[H[p(y^*|x^*)] - H[p(y^*|x^*, \theta)]] \quad (4.4)$$

One of the critical shortfalls of BALD is that information gain on the parameters does not necessarily lead to information gain on prediction [155]. This is especially obvious in high dimensional problems, where many inputs might actually be irrelevant to actual prediction. To deal with this, Smith et al. [155] introduce the idea of a target distribution $p(y_s|x_s)$, over the which the acquisition function seeks to gain information. Here, y_s denotes the labels associated with the target distribution $p(x_s)$. The IG for this is shown in Equation 4.5.

$$\text{IG}_{y_s} = H[p(y_s|x_s)] - H[p(y_s|x_s, x_*, y_*)] \quad (4.5)$$

An expectation over a number of realisations of x_s must be taken to calculate the average reduction in predictive uncertainty over the target distribution of interest. Obviously, the expectation over y_* must also be taken as it is still a random variable.

$$\text{EPIG}_{y_s} = \mathbb{E}_{p(x_s)p(y_*|x_*)}[\text{IG}_{y_s}] \quad (4.6)$$

Since it is equivalent to mutual information, EPIG can be written as a KL divergence:

$$\text{EPIG}_{y_s} = \mathbb{E}_{p(x_s)}[\text{KL}[p(y_s, y_* | x_s, x_*) || p(y_s | x_s)p(y_* | x_*)]] \quad (4.7)$$

As mentioned in the previous chapter and throughout this one, the GP has some innate properties that make it highly attractive for active learning. In the context of EPIG, one of these properties is the fact that the entropy for a GP posterior with probit likelihood (after a posterior approximation) has an analytical solution, as given in [143]. The analytical expression is shown in Equation 4.8, where Φ is the Gaussian CDF and μ_{x_*} and $\sigma_{x_*}^2$ are the predictive mean and variance of the GP respectively. Note that the binary entropy function h is given as $h(p) = -p \log p - (1-p) \log(1-p)$. For acquisition functions not centred around entropy, the returned Gaussian posterior is still massively useful. Many of these acquisition functions will require some form of uncertainty calculation which is typically innately provided by the GP.

$$H[y_* | x_*, D] \approx h\left(\int \Phi(f_{x_*}) \mathcal{N}(f_{x_*} | \mu_{x_*}, \sigma_{x_*}^2) df_{x_*}\right) \quad (4.8)$$

$$= h\left(\Phi\left(\frac{\mu_{x_*}}{\sqrt{\sigma_{x_*}^2 + 1}}\right)\right) \quad (4.9)$$

If one still wishes to calculate the joint distribution $p(y_s = 1, y_* = 1 | x_s, x_*)$ in Equation 4.7, there exists a shortcut as shown in [149] and [62] to reduce it to a linear time singular integral.

4.1.2 EPIG SHM case study

The binary problems from Los Alamos Bookshelf dataset introduced the previous chapter provide an excellent first SHM test for EPIG. A binary GPC is used, and EPIG is benchmarked against BALD, random sampling and uncertainty sampling.

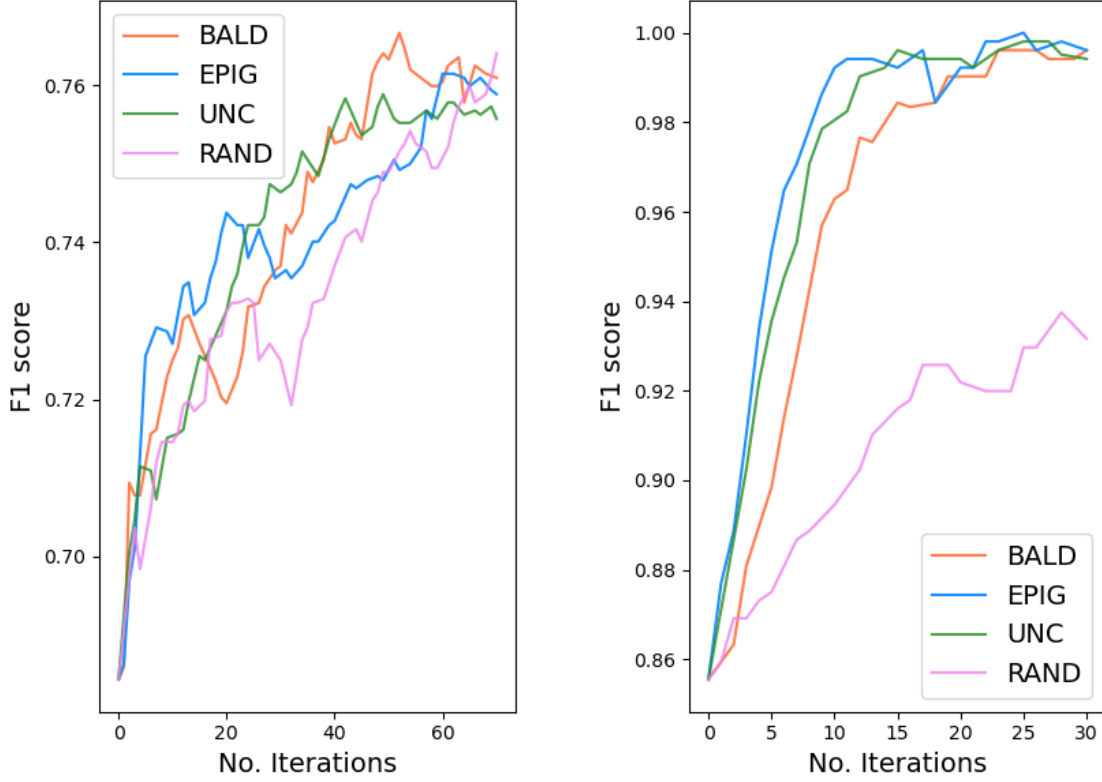
To evaluate an acquisition function, it is typical to examine convergence of the metric of choice (in this case F1 score) as new data is added. A faster improvement in F1 score implies a better acquisition function, and thus a reduced labelling cost. The

results of the benchmark on the PCA FRF problem are shown in Figure 4.3a, while the results for the PCA Coherence are shown in Figure 4.3b. In both cases, only one observation is added per iteration. Each plot is an average of 15 runs. Note that EPIG is evaluated here with the target distribution equal to the prediction data (no target distribution). It is clear that EPIG converges slightly faster than BALD in the Coherence problem, and more slowly in the FRF problem. This difference is not truly significant, and instead corroborates the results of the original paper in that EPIG is a suitable drop in replacement for BALD even when no target distribution is used. In the FRF results it is notable that all acquisition functions do not significantly outperform random sampling. Since all acquisition still see a significant improvement, this implies that most of the available data was informative to the classifier. If there had been no improvement in F1 score and random sampling provided equal convergence, this could instead imply that the underlying GP is not quite trustworthy enough in the low data environment the experiment was initialised in to be truly useful. This is a useful lesson to highlight for practitioners: active learning is only useful if your underlying model has enough useful data to make *a* prediction.

To illustrate why a target distribution is useful in SHM two case studies are investigated, only BALD and EPIG are benchmarked here, since uncertainty and random based sampling converged much slower in the previous task. The first case study is a simple example, taking the coherence dataset and placing a tight Gaussian target distribution $p(y_s|x_s)$ over the undamaged cluster. A reasonable prior here could be to prioritise information gain around what is known to be a tightly clustered group. The mean convergence rates are shown in Figure 4.4, averaged over 15 runs with variance shaded in. Both active learners are given a restricted prediction set centred around the distribution. This ensures that it is a fair comparison, since BALD is now cycling through target samples drawn from the prediction set.

Although it is likely rare that a practitioner would desire better prediction over undamaged cases, this serves a useful example of a simple and interpretable target distribution. It is clear from Figure 4.4 that EPIG is able to accurately target the distribution, whilst BALD cannot. Note that since the example is over a different prediction set, convergence rates are different to Figure 4.3b.

The second example targets a more complex damage sensitive cluster. The 8-DOF



(a) First two PCAs of the FRFs (Problem shown in Figure 3.2a)

(b) First two PCAs of the Coherences (problem shown in Figure 3.2b)

Figure 4.3: Convergence rates for binary SHM case studies. Only one new observation is added per iteration. EPIG is expected predicted information gain, BALD is Bayesian Active Learning by Disagreement, UNC is choosing the next observation as the one with the highest predictive uncertainty, and RAND is choosing the next observation randomly.

simulated data from Bull et al. [140] is modified to represent a more interesting binary classification challenge by changing each odd class label to damage, and each even to undamaged (recall in Figure 3.7 six classes of increasing damage are simulated). The modified classification problem is shown in Figure 4.5a. A target $p(y_s|x_s)$ is placed over the first two classes, shown as the colour points in Figure 4.5b. This target can be thought of as specifically aiming for more information over the selected decision boundary of interest (the two clusters in Figure 4.5b). Only the target is included in the test set.

In the original problem, the two target classes are the undamaged class and the slightly damaged class. This decision boundary is perhaps one of the most important

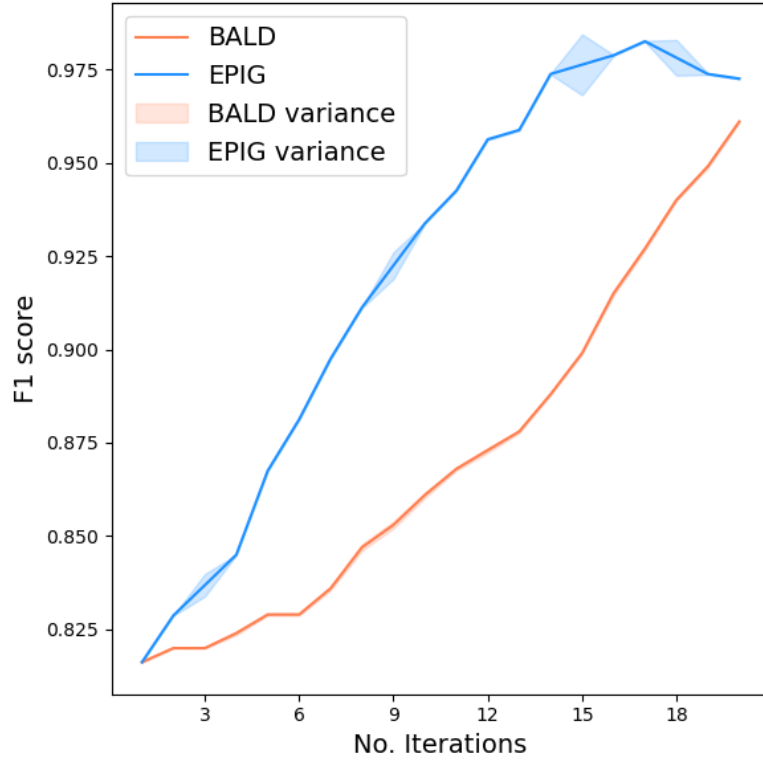


Figure 4.4: Coherence convergence study with a target distribution over undamaged cluster

to understand in the entire problem space; it is noisy and overlapping, and further damage is distinct and can readily be handled by classification algorithms. A Gaussian target distribution is centred on the decision boundary of these two classes. The convergence results can be seen in Figure 4.6. It is visually clear that EPIG improves in F1 score faster than BALD on this targetted dataset, and settles on a higher converged F1 score.

4.2 Fast updating for Active Learning via generalised online learning

One of the central parts of active learning is the updating step. This is often the most computationally expensive step, but is often left undiscussed in the literature, largely because this step is model specific. This section introduces a novel extension to online PEP updates for Gaussian Processes, and is one of the first works to apply

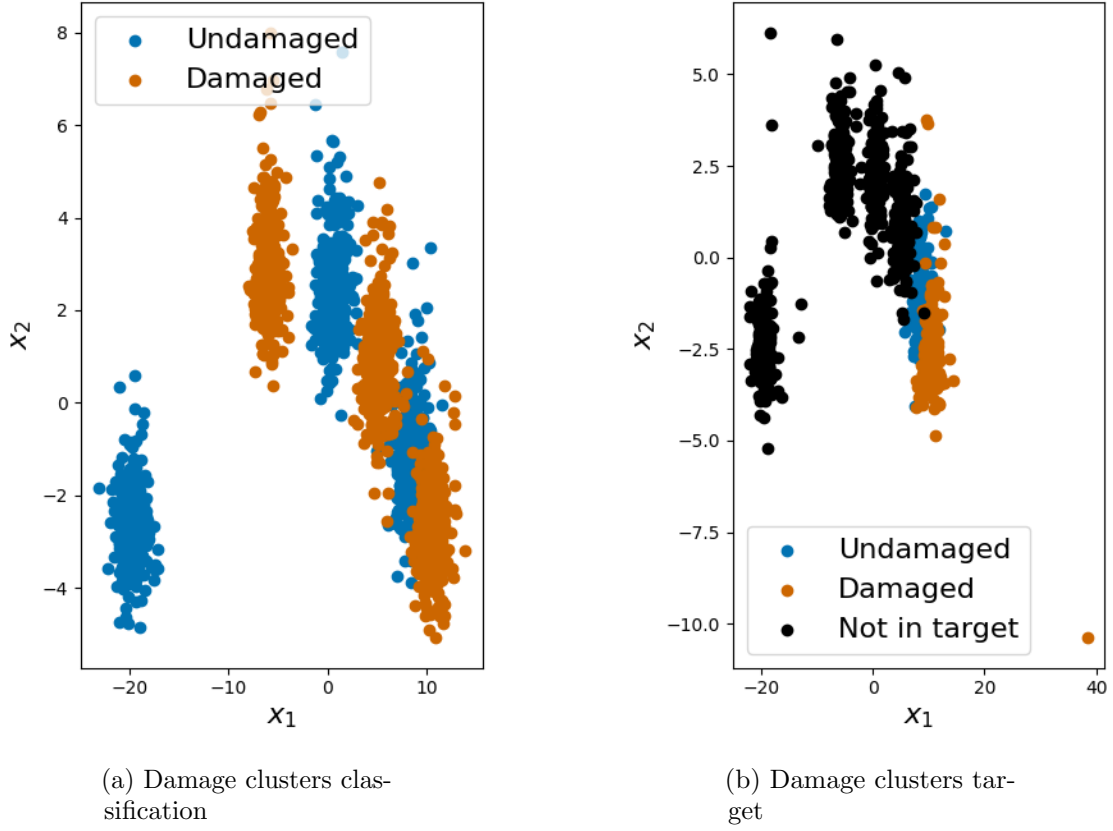


Figure 4.5: Damage clusters classification problem

this method to active learning. As a reminder, the graphic in Figure 4.7 illustrates where in the AL framework the contribution introduced in this chapter sits.

Before introducing the online PEP update, it is worth again taking some time to discuss the current approaches to online updates for GPs. As discussed previously, Rank one Cholesky updates are not a suitable choice for the problem at hand; the GP is approximate and sparse, and many data may be labelled and added at once, all things a Cholesky update is not able to handle. Online variational frameworks do exist and are quite prevalent, and could easily be adapted to an active learning framework. However, if the performance of the PEP approximation for classification is to be kept, this is not an appropriate choice for an updating scheme. One final option to consider is that within the PEP framework, a reasonable update would be to fix the number of inducing points between updates, and take several EP steps using the previous EP parameters. In practise, this approach can suffer from instability especially if the change in the input data is large, but is typically suitable for small updates within the acquisition function to take the expectation over y_* (as shown in

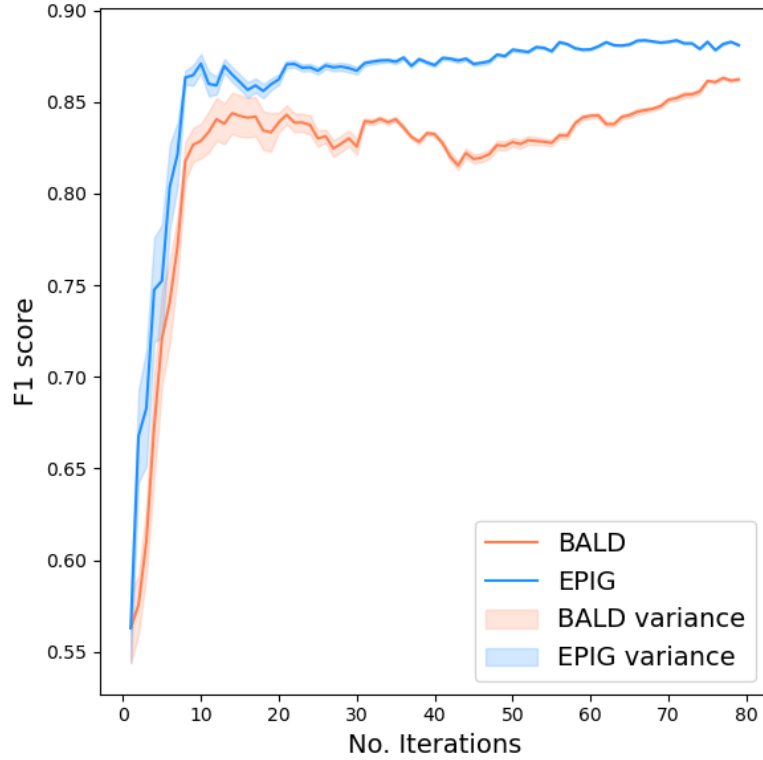


Figure 4.6: Damage cluster convergence study with target distribution over clusters of interest

Equation 4.2).

Bui et al. [157] introduced a variational streaming framework for GPs, and in the work also illustrated how this extends to PEP ¹. However the online PEP framework introduced in [157] was only derived for Gaussian likelihoods. Following the methodology for joint PEP problems introduced in section 3.4, an extension to generalise the online PEP framework to arbitrary non Gaussian likelihoods is introduced here.

This section will begin by introducing the online PEP framework, largely following the original derivation. The extension will then be highlighted, and then benchmarked on several posteriors of interest.

¹Recall in chapter 3 that Bui et al. [55] proved that sparse variational approximation schemes can be considered equivalent to PEP when alpha equals zero.

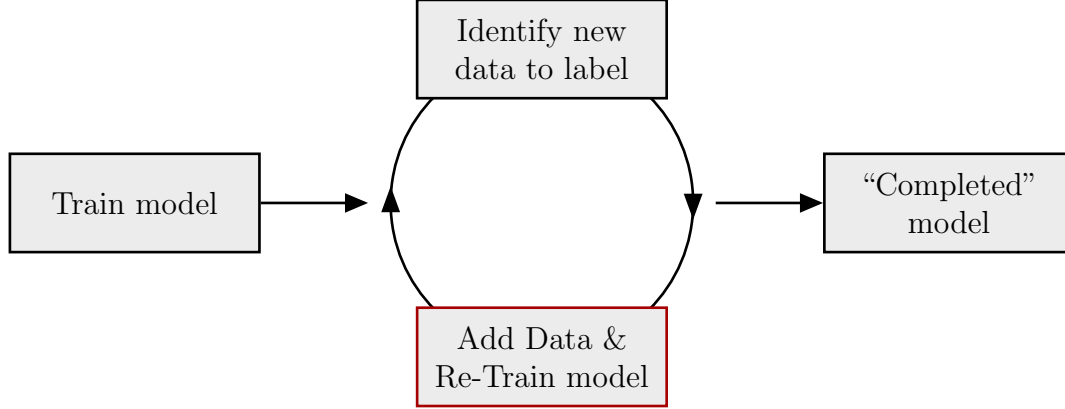


Figure 4.7: Situation of (this) section 4.2 within the AL loop

4.2.1 Sparse online PEP updating

The exact sparse posterior upon observing new data is given in Equation 4.10. Note that $\mathbf{a} = f(u_{\text{old}})$ and $\mathbf{b} = f(u_{\text{new}})$. Assuming that the hyperparameters do not change significantly after each update, the exact posterior can be approximated via Equation 4.12

$$p(f|\mathbf{y}, \mathbf{y}_{\text{old}}) = \frac{1}{\mathcal{Z}} p(f_{\neq a}|\mathbf{a}, \theta_{\text{old}}) q(\mathbf{a}) p(\mathbf{y}|f) \quad (4.10)$$

$$= \frac{1}{\mathcal{Z}} p(f|\theta_{\text{old}}) \frac{q(\mathbf{a})}{p(\mathbf{a}|\theta_{\text{old}})} p(\mathbf{y}|f) \quad (4.11)$$

$$p(f|\mathbf{y}, \mathbf{y}_{\text{new}}) = \frac{1}{\mathcal{Z}} p(f|\theta_{\text{new}}) \frac{q(\mathbf{a})}{p(\mathbf{a}|\theta_{\text{old}})} p(\mathbf{y}|f) \quad (4.12)$$

Bui et al. [157] propose the following approximate posterior, shown in Equation 4.13, where $q_1(b)$ and $q_2(b)$ are the PEP approximations to $\frac{q(\mathbf{a})}{p(\mathbf{a}|\theta_{\text{old}})}$ and $p(\mathbf{y}|f)$ respectively. In plain terms, $q_1(b)$ uses the new inducing points b to summarise the information of the old posterior contained by the old inducing points a , and $q_2(b)$ summarises the likelihood at the new inducing points b . Recall that this is a very similar structure to the variational structure introduced in Equation 3.34. Much of the same benefits and patterns will follow from this choice.

$$q(f) \propto p(f|\theta_{\text{new}})q_1(b)q_2(b) \quad (4.13)$$

Following the traditional PEP framework, each $q_i(b)$ is approximated separately. The PEP approximation for these factors contains a neat shortcut when the factors are Gaussian since they are only over the inducing points, rather than all of f . To intuitively illustrate what this means, the example of a simple PEP approximation for a GP is revisited with some notation over u made explicit. Recall that $q(f, u) = p(f_{\neq u}|u)q(u)$. It follows that the cavity is as given in Equation 4.14. One of the key benefits of this trick is that it is no longer necessary to factorise over f_i (and thus all notation over i is no longer necessary), as will be shown shortly.

$$q(f, u)^\setminus = \frac{q(f, u)}{t(u)} \quad (4.14)$$

$$= q^\setminus(u)p(f|u) \quad (4.15)$$

It then follows that the tilted distribution $\tilde{q}(u)$ (note that this is the tilted over u not f) is the joint tilted distribution $q(f, u)$ marginalised over f :

$$\tilde{q}(u) = \int q(f, u)^\setminus p(y|f)^\alpha df \quad (4.16)$$

$$= q^\setminus(u) \int p(f|u)p(y|f)^\alpha df \quad (4.17)$$

If everything inside the integral $\int p(f|u)p(y|f)^\alpha df$ is Gaussian the result will also be a Gaussian which can be calculated in closed form. And since the objective is updates of the sites at u (not f), the update step requires a division of the cavity of u , neatly cancelling the cavity of u outside the integral. Thus the new distribution $q(u)^{\text{new}}$ can be set equal to the closed form Gaussian that pops out of the integral (which in the Gaussian case is closed form), as shown in Equation 4.18.

$$q(u)_{new} = \left[\frac{q^\backslash(u) \int p(f|u) p(y|f)^\alpha df}{q^\backslash(u)} \right]^{\frac{1}{\alpha}} \quad (4.18)$$

$$= \left[\int p(f|u) p(y|f)^\alpha df \right]^{\frac{1}{\alpha}} \quad (4.19)$$

This is a shortcut because there is no longer a need to sweep over each i^{th} point and calculate the cavity and projection for each. The closed form for the *entire* update is now available. This is only possible when the factor being approximated is Gaussian, as otherwise the integral over f is intractable and not a suitable target for approximation.

Moving back now to the problem at hand, this trick applies almost directly to $q_1(b)$. It is slightly different here in that since the approximation only need approximate the latent function of the old posterior at a , the marginalisation of the tilted is instead over a (rather than f). This is briefly shown below, with the cavity defined in Equation 4.20 and the tilted in Equation 4.22.

$$q_1^\backslash(f) = p(f) q_1^{1-\alpha}(\mathbf{b}) q_2(\mathbf{b}) \quad (4.20)$$

$$= p(f_{\neq \mathbf{a}, \mathbf{b}} | \mathbf{b}) p(\mathbf{b}) p(\mathbf{a} | \mathbf{b}) q_1^{1-\alpha}(\mathbf{b}) q_2(\mathbf{b}) \quad (4.21)$$

$$\tilde{q}_1(f) = p(f_{\neq \mathbf{a}, \mathbf{b}} | \mathbf{b}) p(\mathbf{b}) p(\mathbf{a} | \mathbf{b}) q_1^{1-\alpha}(\mathbf{b}) q_2(\mathbf{b}) \frac{q(\mathbf{a})}{p(\mathbf{a} | \theta_{old})}^\alpha \quad (4.22)$$

Remembering now that a must be marginalised out, and that the resulting cavity over b is cancelled when the new factor is calculated, the new factor is calculated as shown in Equation 4.23. Since everything is Gaussian in this integral (and always will be Gaussian since the old posterior must necessarily be Gaussian), a closed form solution exists for the moments of the new factor, this is omitted here for brevity but shown in Appendix C.

$$q_{1,new}(b) \propto \left[\int p(\mathbf{a} | \mathbf{b}) \frac{q(\mathbf{a})}{p(\mathbf{a} | \theta_{old})}^\alpha \right]^{\frac{1}{\alpha}} \quad (4.23)$$

The same applies to $q_2(\mathbf{b})$ when the likelihood is Gaussian, and the derivation for this is found in the original text [157]. When the likelihood is not Gaussian, this trick is no longer possible.

If an online and generalised PEP approximation is the desired goal, then it is necessary to revert to a normal PEP procedure, visiting each f_i in turn. This is only necessary for $q_2(b)$, since the previous term $\frac{q(\mathbf{a})}{p(\mathbf{a}|\theta_{\text{old}})}$ must necessarily always be Gaussian, and the integral over a will thus always be analytic. In order to make PEP over $q_2(b)$ sparse, it is easier to change the approximating structure proposed by Bui et al. [157], shown in Equation 4.13 to a conditional $p(f \neq b|b)$. This results in the approximating posterior:

$$q(f) \propto p(f \neq b|b)q_1(b)q_2(b) \quad (4.24)$$

This is a necessary change since a conditional prior over the inducing points makes the PEP maths significantly neater. Note that this change does not impact the factorisation trick required for $q_1(b)$, since $p(a|b)$ still drops out of the new prior via the laws of conditional probability $p(a, f_{\neq a,b}|b) = p(f_{\neq a,b}|b, a)p(a|b)$. Here the dependence on a is dropped, just as it is in the original work.

When approximating $q_2(b)$, $q_1(b)$ is held fixed. It is acceptable to ignore $q_1(b)$ during the approximation and multiply the resulting posterior by $q_1(b)$ after, but this is undesirable, since inclusion will bias the cavity in interesting ways that should be captured. Luckily, once $q_1(b)$ is centred on b via a completing of the square (Appendix C), it is simple to combine $q_1(b)$ with the approximate posterior via natural parameters, and continue with a standard PEP approximation.

Following Appendix A, the approximate factors for $q_2(b)$ in natural parameter form are $t_n(\mathbf{b}) = \mathcal{N}(\mathbf{b}; \mathbf{T}_{1,n}, \mathbf{T}_{2,n})$. This naturally leads to the approximate posterior with natural parameters $T_{1,b} = \sum_i T_{1,n} + T_1^{q_1(b)}$ and $T_{2,b} = K_{bb}^{-1} + \sum_i T_{2,n} + T_2^{q_1(b)}$. If no minimisation to rank 1 natural parameters is desired, then the PEP approximation can continue as laid in Appendix A.

The marginal likelihood proceeds just as the marginal likelihood approximation for chained GPs in section 3.4. Where the approximation is given in Equation 4.25. Since $q_1(b)$ is now given in terms of b , this makes the log normaliser of the full

posterior ϕ_{post} very simple to compute, as it is now a product of Gaussians over b .

$$\mathcal{F} = \phi_{\text{post}} - \phi_{\text{prior}} + \frac{1}{\alpha}(\log \tilde{Z}_1 + \phi_{\text{cav},1} - \phi_{\text{post}}) + \frac{1}{\alpha}(\log \tilde{Z}_2 + \phi_{\text{cav},2} - \phi_{\text{post}}) \quad (4.25)$$

The high level idea of the generalised streaming PEP is shown in Algorithm 5, while Appendix C lays out the full derivation for the full rank case.

Algorithm 5 PEP algorithm for generalised streaming PEP

Calculate $q_1(b)$:

1: **Cavity:**

$$q_1^\backslash(f, a, b) = p(f, a, b)q_1^{1-\alpha}(\mathbf{b})q_2(\mathbf{b}) = p(f_{\neq \mathbf{a}, \mathbf{b}}|\mathbf{b})p(\mathbf{b})p(\mathbf{a}|\mathbf{b})q_1^{1-\alpha}(\mathbf{b})q_2(\mathbf{b})$$

2: **Combined Projection & Update:**

$$q_{1,\text{new}}(b) \propto \left[\frac{q_1^\backslash(f, b) \int p(\mathbf{a}|\mathbf{b}) \frac{q(\mathbf{a})}{p(\mathbf{a}|\theta_{\text{old}})}^\alpha}{q_1^\backslash(f, b)} \right]^{\frac{1}{\alpha}} = \left[\int p(\mathbf{a}|\mathbf{b}) \frac{q(\mathbf{a})}{p(\mathbf{a}|\theta_{\text{old}})}^\alpha \right]^{\frac{1}{\alpha}}$$

Since $q_{1,\text{new}}(b)$ is Gaussian, it is simple to combine it into the approximate posterior. Now treat the approximation of $q_2(b)$ as a normal PEP approximation but with a prior that now includes $q_{1,\text{new}}(b)$.

Calculate $q_2(b)$; for each $\mathbf{n}$ point, calculate:

1: **Cavity:** $q^\backslash^n(f) = q(f)/\tilde{t}_n(u)^\alpha$

2: **Projection:** $\eta_{\text{new}} = \mathbb{E}_{\hat{P}(f)}[\phi_q(u)] = \frac{d}{d\theta^\backslash_i} \log Z_n + \eta^\backslash^n$, And pick $Q^{\text{new}} \in \mathcal{F}$

3: **Update:** Replace $\tilde{t}_i(u)$ with: $\tilde{t}_n(u)^{\text{new}} = (Q^{\text{new}}/Q^\backslash^n)^{1/\alpha}$

Once each site has been visited, take a step in the hyperparameter space via $\frac{d}{d\theta} \frac{1}{\alpha} \sum_{k=1} \sum_{n=1} \log C_{k,n} + \Phi(\theta) - \Phi(\theta^{(0)})$.
Repeat until convergence.

The astute reader will have noticed that when the likelihood is not Gaussian, something that looks suspiciously like a standard PEP approximation emerges. The attractive computational shortcut of the online PEP framework depends explicitly on the integral over all f in Equation 4.13 being analytical. When this is not the case, the neat shortcut which collapses most of the PEP calculations into a closed form one step sweep does not exist. Put simply, there is no longer a highly efficient way to compute $q_2(b)$. Instead, a traditional PEP approximation is used (which is still sparse and efficient). This is not strictly bad news though: the PEP framework is designed for online cases where previous data is no longer seen. This means

that *actual* training data sizes are only as large as incoming batches of data - all previous data being summarised by $q_1(b)$. The computational cost of retraining is therefore dominated by the PEP approximation of $q_2(b)$, with $\mathcal{O}(nm^2)$ computational cost. Since it is unlikely incoming batches will be very large, n will be small and thus retraining will be quick. What has been derived then is still a robust PEP approximation ready for deployment in online environments within SHM or in active learning cases.

If the full computational savings are truly desirable, then several options remain. The first is to simply accept defeat and switch to the variational approach outlined in the same paper. The second is that in very very small updates (3 data points), the integral will be approximatable via nested quadrature. The last option is to directly approximate the non Gaussian likelihood with a Gaussian via one-shot variational inference in transformed bases [159]. While also abandoning PEP, this approach has been shown to perform very well in the streaming case.

4.2.2 Sparse Streaming case study

To benchmark how well the generalised streaming framework performs in prediction tasks and how well $q_1(b)$ captures previously seen data, two problems are investigated. Both problems are set up the same way: a classification problem is split up into three batches. At each batch, the streaming PEP is tasked with predicting *all* previously seen test cases but can only see the current batch of training data (the coloured data). Previously seen data will be shown in grey, and a full baseline PEP GP is trained at each batch on *all* previously seen data. For both problems, each data batch is partitioned to be significantly out of distribution of previously seen data in order to mimic a worst-case streaming scenario. Computational benchmarking for this framework is not informative, since computational cost is tied to the sparse approximation of $q_2(b)$ and is dominated by training dataset size.

The first is the synthetic data from Bull et al. introduced in Figure 3.7 [140]. Here, as earlier in Figure 4.5a each cluster is alternatively labelled as damaged undamaged, rather than as a spectrum of progressively more damaged classes in the multi class GP case study and in the original work. The results of this in the streaming case are shown in Figure 4.8a, with the baseline GP shown in Figure 4.8b. The second

streaming case study is the FRF case study, previously shown in Figure 3.2a. The results of which are shown in Figure 4.9a, with the baseline GP shown in Figure 4.9b.

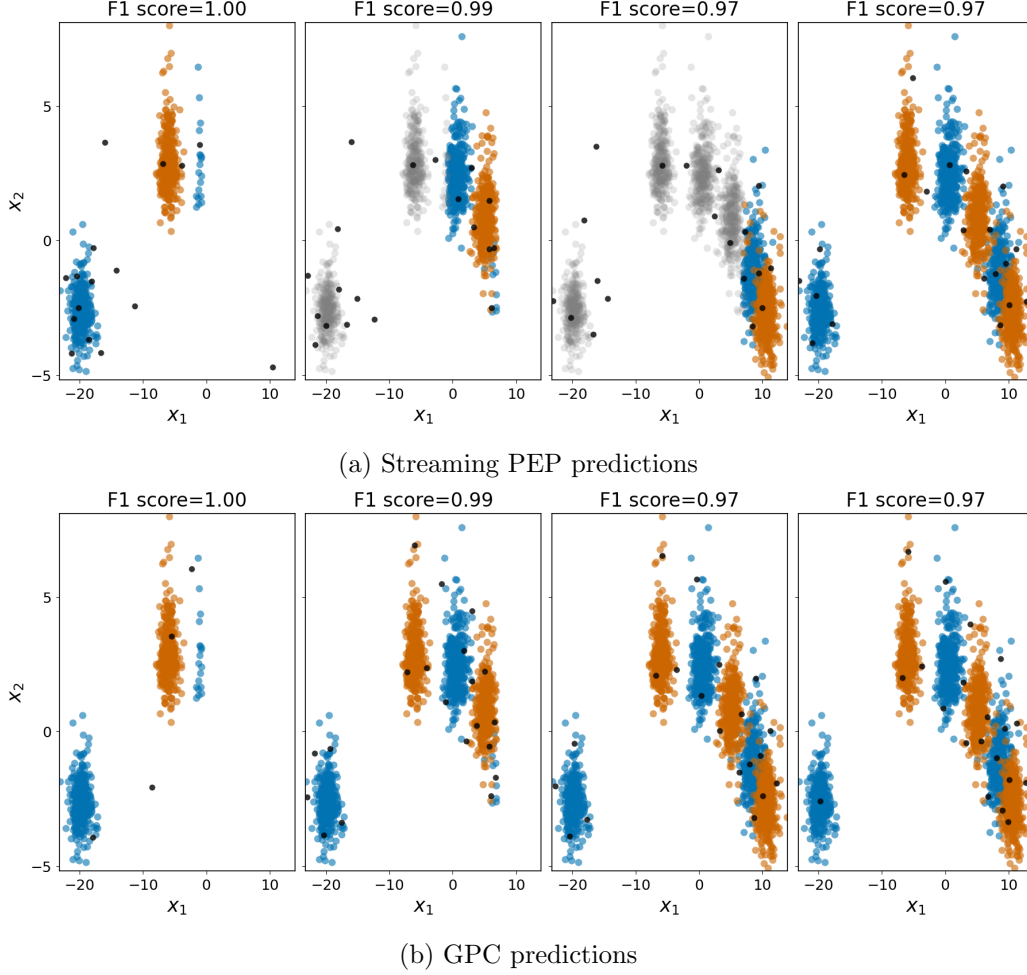


Figure 4.8: Benchmark of streaming PEP over the FRF dataset from Figure 4.5. Each column represents a new batch of data coming in. At each step, the classifiers are asked to predict all previously seen data, but the streaming PEP can only see the coloured data.

The results show that at each step, the streaming PEP correctly predicts the entire historical dataset despite only seeing a small subset of it. This is best exemplified in the third update step (third column), when a majority of the dataset is unavailable to the streaming PEP (unavailable data being shown as greyed out) and yet its F1 score remains comparable to the baseline GPC that sees the *entire* dataset at each batch. It is thus clear that the streaming PEP is able to completely capture the entire dataset via $q_1(b)$ despite not currently seeing it. Since previously seen data can be robustly captured by $q_1(b)$ a practitioner can be confident that the *only* training data necessary at each step is the incoming batch, all previous information

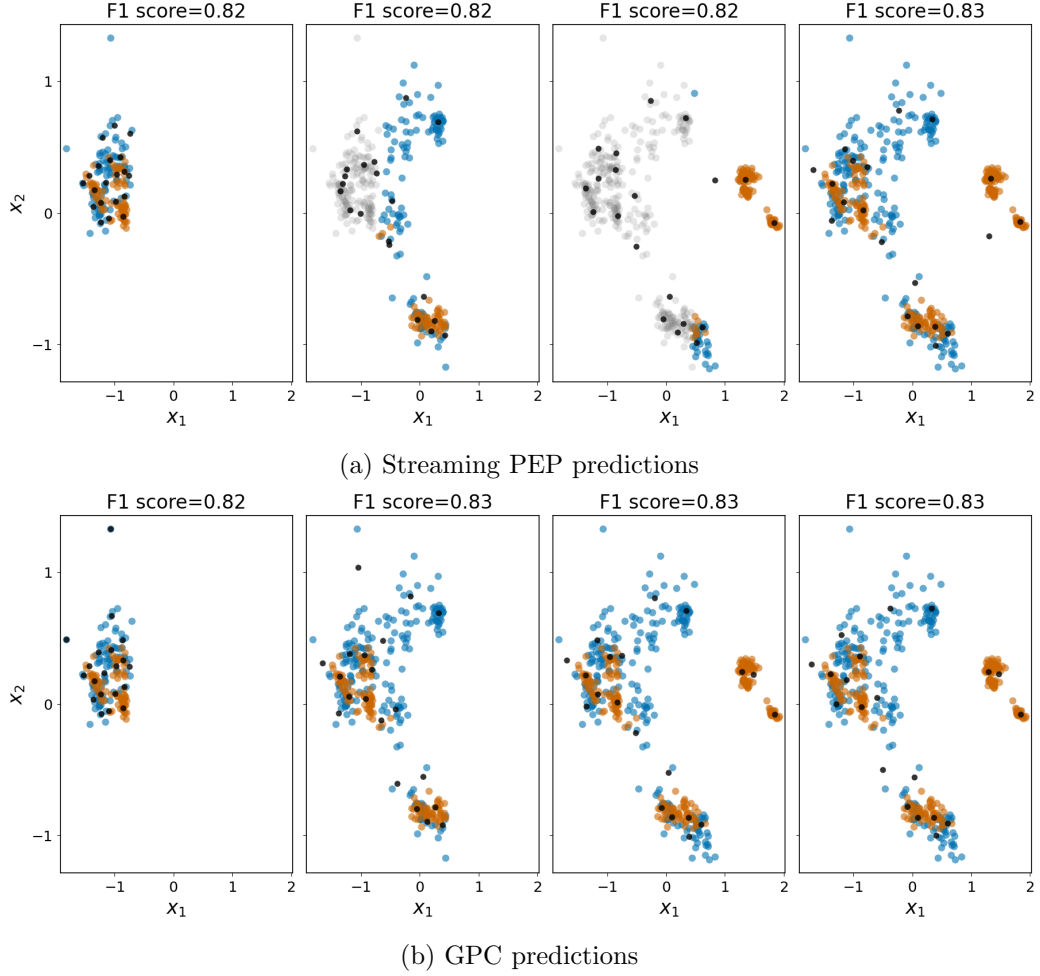


Figure 4.9: Benchmark of streaming PEP over the FRF dataset from Figure 3.2a. Each column represents a new batch of data coming in. At each step, the classifiers are asked to predict all previously seen data, but the streaming PEP can only see the coloured data.

being stored in $q_1(b)$. This is a significant result, as it enables significant memory and computational savings when PEP GPs are deployed for online monitoring, active learning or other streaming setting.

Finally, the PEP extension is benchmarked against its VFE equivalent. The results of this are shown in Figure 4.10. The results are unsurprising; a very slight edge to the PEP approximation, with VFE outperforming PEP on the last test case. For the relatively simple classification problems presented here, it is expected both approximations will converge well to the true GPC. For further comparison between the two approximation methods, the reader is again referred to [127, 128].

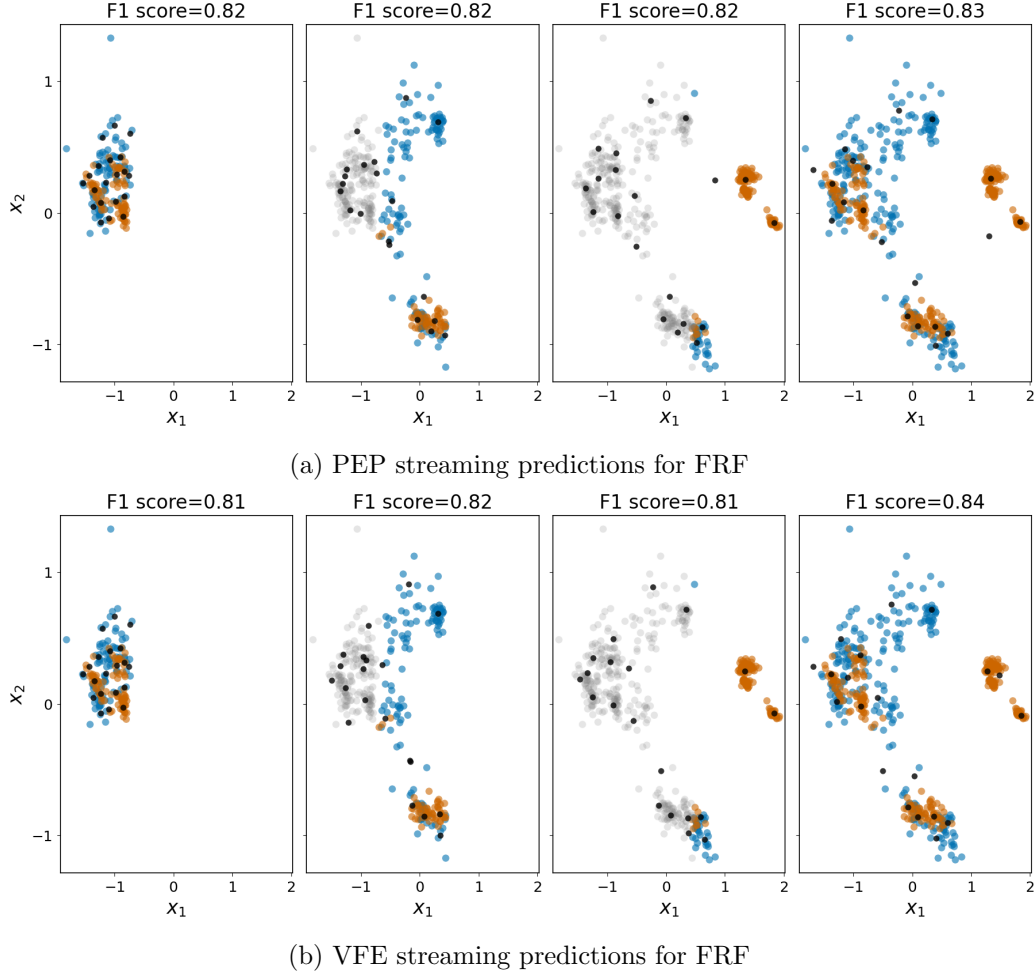


Figure 4.10: Streaming PEP vs streaming VFE over the FRF dataset from Figure 3.2a. Each column represents a new batch of data coming in. At each step, the classifiers are asked to predict all previously seen data, but the streaming PEP can only see the coloured data.

4.3 Discussion

This chapter has presented contributions in all sections of the active learning cycle. It first introduced EPIG as an attractive acquisition function for SHM before introducing a new tool for fast updates of non-Gaussian GPs in streaming settings (including active learning).

The introduction of EPIG was motivated by the fact that many SHM problems have known underlying mechanics that allow a practitioner to focus on a particularly important part of the input space. With the introduction of a target distribution,

EPIG creates the framework for these parts of the space to be targetted without the need for arbitrary cut-offs in the input space. It is shown that EPIG operates on parity with other acquisition functions when there is no target distribution, and significantly better when there is one. The motivating examples used were two damage clustering problems, where it was shown that EPIG significantly outperformed BALD. While it is a constrain on exploration, EPIG can be a powerful tool when the input space can be targetted with known engineering principles or sensible engineering targets (such as the reasonable desire to prioritise the damage/undamaged decision boundary in the second case study).

BALD, EPIG, and many other acquisition functions explicitly depend on samples drawn from the candidate pool of labels in order to identify which candidate is most informative. When sample sizes are constrained by computation or data availability, these methods become sensitive to sample quality and may generate uninformative or misleading suggestions. EPIG is especially vulnerable, as it depends on draws from both the candidate and target pools. A potential future direction of research could be to investigate ways to proof acquisition functions against sampling bias.

The introduction of the online PEP update framework to active learning is novel in and of itself, however the extension to the non-Gaussian case is a significant step towards making the PEP framework more widely useable. It is also a significant contribution to the streaming GP literature, introducing another streaming approximation framework with its own benefits and drawbacks.

To conclude, by visiting all areas of the active learning cycle, this chapter has made significant contributions to the literature in SHM and GPC based active learning, as well as the wider streaming GP literature.

Further research should be investigations into the limits of the online framework. Questions such as “When do the inducing points presenting the previous data cease to do so?” and “What are the robust limits of this sparse representation of the dataset?” are important ones that must be resolved by deployment into real world online settings.

CONCLUSIONS AND FUTURE WORK

This thesis set out to develop an improved and modernized Gaussian Process (GP) toolbox for SHM. In order to achieve this, four specific objectives were laid out. These were: (1) Advance the study of bounded GPs in SHM. (2) Introduce state-of-the-art GP classification techniques into SHM. (3) Address the challenge of data labelling costs for classification methods in SHM. (4) Enable the online deployment of GP classifiers. The detection, localisation and classification stages of Rytter’s hierarchy were motivated as use cases for the work presented throughout.

To meet these objectives, several theoretical advancements from the machine learning community were adapted and applied to SHM. In addition, a novel generalized framework for online PEP updates was derived and demonstrated, with relevance to broader theoretical GP research. The main contributions of this thesis were as follows:

1. Bounded GPs were investigated, and shown to be more effective in modelling the uncertainty in a real world bounded problem
2. Gaussian Process Classification (binary and multi-class) were shown to be effective modelling approaches for the detection and localisation stages of SHM. It was shown how to derive a PEP approximation for any chained GP.
3. Targetted active learning via EPIG was shown to be more effective than traditional techniques when engineering prior knowledge can guide the targets

effectively.

4. A novel generalised online PEP framework was derived.

The objectives set out at the beginning of this thesis have thus been successfully achieved, resulting in a set of tools that together form a state-of-the-art GP framework for SHM. These tools have wide-ranging applications within SHM: bounded GPs can better model bounded physical problems; GPC is a powerful tool for predicting discrete structural states, delivering both accurate predictions and reliable uncertainty estimates; targeted active learning via EPIG can efficiently reduce labeling costs when targetting areas of interest; and the online PEP framework enables the deployment of GPCs (and other GPs that benefit from PEP approximations) in real-time SHM applications. Although only the bounded GP component is directly PIML, the enhanced GP methods developed in this thesis offer a solid probabilistic foundation that easily lends itself to future GP-based PIML applications in SHM.

At the time of writing, advancements in AI and ML have been dominated by very large black-box models. Faced with these results, it can sometimes feel difficult to justify research in smaller methodologies and frameworks. This work has shown that Gaussian Processes are powerful and flexible tools, both philosophically and practically justified within the SHM domain. They offer robust uncertainty quantification and multiple ways to incorporate physics, making them an essential component of the SHM toolbox.

5.1 Bounded Gaussian Processes for SHM

When an SHM problem involves strict physical bounds in the output space, standard GPs become suboptimal, as their mean predictions and probability mass do not necessarily respect these bounds. Physically implausible predictions reduce immediate utility and imperil downstream decision making involving the model such as risk based decision frameworks. To address this issue, Chapter 2 introduced and evaluated two bounding methods. The motivating example used are anonymised power curves from operational offshore wind turbines.

Of the two methods investigated, the Heteroscedastic Beta Process proved most

effective, outperforming the baseline GP with a significantly higher joint likelihood and comparable NMSE. The higher joint likelihood indicates that the bounded uncertainty region more accurately captures the test data. Since one of the main attractions of GPs are their innate UQ, improving this ability in bounded problems is a key result. Although the NMSE difference between the standard GP and HBP is minimal, it overlooks that some standard GP predictions are physically implausible. Even in ideal cases where mean predictions remain within bounds, the standard GP cannot guarantee physical plausibility for all future test cases. The HBP model therefore offers greater reliability and operator trust in the power curve model, even when compared to a theoretically perfect standard GP. While this was not implemented, the bounded GPs presented in this chapter are a tool for damage detection, with the provided physically meaningful uncertainty enriching the outlier or novelty detection process, as well as other downstream tasks.

5.2 Gaussian Process Classification for SHM

Gaussian Process Classifiers were explored in chapter 3 as potential models for the Detection and Localisation stages of Rytter’s hierarchy, which require discrete classification of structural states. All GPCs were implemented via the state-of-the-art approximation Power Expectation Propagation. Both binary and multi-class GPCs are introduced and benchmarked against an SVM on a variety of laboratory and synthetic datasets.

The binary GPC demonstrated strong performance across multiple SHM datasets, achieving predictive accuracy comparable to or better than that of an SVM. Given the highly limited attention binary GPCs have received in SHM literature, their further exploration using a modernised approximation framework and broader problem set constitutes a meaningful contribution.

Introducing a PEP-based multi-class GPC represents a significant novelty in SHM literature, enabling probabilistic progression up Rytter’s hierarchy with state-of-the-art predictive performance. By demonstrating strong classification performance across two SHM challenges, the model is demonstrated to be both flexible and effective for complex supervised classification tasks. Its inherent ability to quantify

uncertainty over class labels further enhances its utility in SHM applications.

Chapter 3 devoted some time to intuitively introducing the PEP approximation, walking through a simple, general approximation to normal GP approximations to complex, chained GP approximations. Consequently, this presentation serves as a practical resource for implementation, contrasting with the often highly theoretical presentations prevalent in current literature.

While effective predictive classification models, the GPCs presented face two critical challenges before they can become fully practical tools in the SHM toolbox. The first challenge is data availability. In this chapter, labelled data was assumed to exist for all structural states of interest. In reality, this assumption rarely holds. Even with the active learning approach investigated in chapter 4, this challenge does not completely disappear. Most importantly there is no mechanism to deal with new classes appearing during deployment. This is an issue that must be solved before GPCs can be considered for deployment in SHM uses. The second challenge is scalability. Even with sparsification, GPCs do not scale efficiently to large datasets when compared to other models. Further, training GPCs is computationally intensive, and some form of adaptation is required to deploy them in the streaming settings typical of SHM applications.

5.3 Targetted Active Learning for SHM

Active learning was investigated as a method of dealing with the need for expensive, labelled data. Chapter 4 introduced Expected Predicted Information Gain as an attractive acquisition function for SHM. The introduction of EPIG was motivated by the fact that many SHM problems have known underlying mechanics, allowing practitioners to focus on critical regions of the input space. By explicitly targetting improvements in prediction over a user defined target distribution, EPIG enables these regions to be prioritized without relying on arbitrary cut-offs.

It was shown that EPIG outperforms other acquisition functions when guided by a target distribution, lowering labeling costs and enhancing model performance, as demonstrated on two synthetic damage clustering problems. It was also shown that EPIG performs on parity with other acquisition functions when there is no

target distribution, making it a flexible substitute for BALD. Although it constrains exploration, EPIG is a powerful tool when input space targetting can be informed by engineering principles or objectives, such as identifying the damage/undamaged decision boundary in the second case study. Of course, the biggest problem with EPIG is that wrongly defined target distributions will suggest sampling in irrelevant or suboptimal regions, potentially biasing the model and reducing overall performance. A key problem with the active learning strategies explored in chapter 4 is their dependence on sampling. EPIG, BALD and many other acquisition functions depend on samples from the candidate pool to identify the best improvement. When sample draw sizes are limited by computation or data size, these methods become sensitive to their quality and may produce uninformative (or even damaging!) suggestions if the draws themselves lack information. EPIG is particularly susceptible to this, since it also depends on target pool draws as well as candidate draws.

5.4 Generalised Online PEP Updates for SHM

Chapter 4 highlighted that updating GPs in active learning presents a computational bottleneck, an issue that extends to broader SHM applications. Since SHM ultimately requires online algorithm deployment, standard GPs are currently ill-suited to this task.

Given that PEP is the preferred approximation for GPCs, developing an online framework for PEP capable of handling general non-Gaussian likelihoods is therefore desirable. The introduction of an online PEP update framework for active learning was novel in itself, and its extension to non-Gaussian likelihoods represents a significant step toward making the PEP framework more widely applicable. This contribution also advances the streaming GP literature, providing a new approximation framework with its own benefits and trade-offs. Further, the non-trivial rank-one derivation demonstrates how the general PEP framework can be leveraged for more complex applications.

5.5 Future Work

Chapter 2 investigated several bounding techniques, opening multiple accessible avenues for future work. The heteroscedastic GP underperformed in the warped space, suggesting potential remedies such as Bayesian committees to address the discontinuities in the warped data or alternative warping functions, though these were beyond the scope of this thesis. While the HBP model was highly successful, the random nature of the sampling and unification of the various distributions into an average distribution at each test point is not the most statistically robust way of dealing with the quantified uncertainty. Uniting the beta distributions of each sample mathematically rather than using the mean and total variance would be more robust and would be worth pursuing. Finally, it may also be worth considering the use of committees in the HBP model in order to better handle the transition to rated power seen in the power curve.

The GPC investigation in chapter 3 similarly highlights several avenues for future research. An immediate and more accessible research direction is exploration of the impact of prior (kernel) selection on GPC performance. Introducing non-stationary kernels to GPCs may be an especially promising approach here, as discrete class data often exhibits non-stationarity across the input space. Another significant research direction is the incorporation of PIML into GPCs. PIML in the context of GPCs is challenging, as the latent space is a highly abstract discriminator rather than a latent-model of a physical process. The most straightforward way to achieve this would be via physics informed feature augmentation - deriving additional sensitive features to the task that might improve predictive performance. While feature augmentation represents a valuable, low-hanging fruit, a more architecturally significant way to incorporate PIML into GPCs could be via a multi-output GP, with one branch as a classifier and another as a GP regression model. Such combined GPC-GPR models already exist as heterogeneous multi-output GPs [160], where the regression branch has traditionally been used to fill in missing data. Instead, the GP regression model could be made PIML, influencing the latent discriminator of the GPC. This approach also connects naturally to semi-supervised GPs [161], which exploit correlations between labelled and unlabelled data to improve predictions. This structure could be similarly leveraged to incorporate PIML into GPCs.

These proposed extensions to GPCs to PIML make it clear that the introduction of GPCs to SHM in this thesis was an important stepping stone, establishing GPCs as powerful probabilistic model class fully capable of tackling SHM problems worthy of further investigations. Beyond these methodological developments, an important operational research direction is understanding how a GPC would handle deployment in practice, particularly in the presence of new, unseen classes.

Finally, chapter 4 also opens new research directions. It is clear that active learners not reliant on sampling would be beneficial. Some studies have explored approximating acquisition functions with smooth, differentiable functions and guiding acquisition using these gradients [149], but these approaches are not universally applicable and depend on the availability of a suitable smooth approximation. A true sample-less active learner would be a significant step forward in the literature. Further incremental research within the online PEP framework is also warranted. Most important is further investigation into how well the inducing points capture the previously seen posterior as that posterior increases in size. It also remains an open question whether a “true” generalised online PEP framework can be developed, but continued investigation is clearly justified.

PEP DERIVATION AND IMPLEMENTATION

This section takes the algorithm PEP algorithm (Algorithm 6) introduced in subsection 3.2.2. Note that this is an abbreviated version of the original work in [55] for reference, an interested reader is referred to the original work for more detail. This section also highlights the QR decomposition in REF introduced in subsection 3.2.2 as a small change to increase computational stability.

To reiterate, the sparsity is induced via the result $\mathbb{E}_{\hat{P}(x)}[\phi(u)] = \mathbb{E}_{q(f)}[\phi(u)]$. That is, during the projection step the moments need only be matched at u , so for each n , only u moments need be matched (instead of n moments).

Algorithm 6 PEP algorithm for approximating $p(f) \prod p(y|f)$

For each $\mathbf{n}$ point, calculate:

- 1: **Cavity:** $q^{\setminus n}(f) = q(f)/\tilde{t}_n(u)^\alpha$
 - 2: **Projection:** $\eta_{new} = \mathbb{E}_{\hat{P}(f)}[\phi_q(u)] = \frac{d}{d\theta^{\setminus i}} \log Z_n + \eta^{\setminus n}$, And pick $Q^{new} \in \mathcal{F}$
 - 3: **Update:** Replace $\tilde{t}_i(u)$ with: $\tilde{t}_n(u)^{new} = (Q^{new}/Q^{\setminus n})^{1/\alpha}$
-

Set-up: Given the approximating posterior $q(f) = p(f_{\neq u}|u)q(u)$ and approximate factors in natural parameter form $t(u)_n = \tilde{\mathcal{N}}(\mathbf{u}; \mathbf{T}_{1,n}, \mathbf{T}_{2,n})$. The approximate posterior in natural parameters is $\mathbf{T}_{1,\mathbf{u}} = \sum_n \mathbf{T}_{1,n}$, $\mathbf{T}_{2,f} = \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} + \sum_n \mathbf{T}_{2,n}$. In mean and variance:

$$m_f = \mathbf{K}_{fu} \mathbf{K}_{uu}^{-1} \mathbf{T}_{2,f}^{-1} \mathbf{T}_{1,n} = \mathbf{K}_{fu} \gamma \quad (\text{A.1})$$

$$V_{ff'} = \mathbf{K}_{ff'} - \mathbf{Q}_{ff'} + \mathbf{K}_{fu} \mathbf{K}_{uu}^{-1} \mathbf{T}_{2,f}^{-1} \mathbf{K}_{uu}^{-1} \mathbf{K}_{uf'} = \mathbf{K}_{ff'} - \mathbf{K}_{fu} \beta \mathbf{K}_{uf'} \quad (\text{A.2})$$

Where γ and β are algebraically easy terms to store and update the approximate factors.

A.1 Full rank factors

Cavity: For each n , the cavity is $q^{\setminus n}(f) = q(f)/\tilde{t}_n(u)^\alpha$. In natural parameters $\mathbf{T}_{1,f}^{\setminus n} = \mathbf{T}_{1,f} - \alpha \mathbf{T}_{1,n}$, $\mathbf{T}_{2,f}^{\setminus n} = \mathbf{T}_{2,f} - \alpha \mathbf{T}_{2,n}$. In mean and variance:

$$m_f^{\setminus n} = \mathbf{K}_{f_n u} \mathbf{K}_{uu}^{-1} \mathbf{T}_{2,f}^{\setminus n, -1} \mathbf{T}_{1,f}^{\setminus n} = \mathbf{K}_{f_n u} \gamma^{\setminus n} \quad (\text{A.3})$$

$$V_{ff}^{\setminus n} = \mathbf{K}_{f_n f_n} - \mathbf{Q}_{f_n f_n} + \mathbf{K}_{f_n u} \mathbf{K}_{uu}^{-1} \mathbf{T}_{2,f}^{\setminus n, -1} \mathbf{K}_{uu}^{-1} \mathbf{K}_{uf_n} = \mathbf{K}_{f_n f_n} - \mathbf{K}_{f_n u} \beta^{\setminus n} \mathbf{K}_{uf_n} \quad (\text{A.4})$$

Projection: Recalling that moments at u only need be matched and using the tilted property $\eta_{new} = \mathbb{E}_{\hat{P}(f)}[\phi_q(u)] = \frac{d}{d\theta^{\setminus i}} \log Z_n + \eta^{\setminus n}$ introduced in Equation 3.20 gives approximating factor a mean and variance:

$$\mathbf{m}_u = \mathbf{m}_u^{\setminus n} + \mathbf{V}_{uf_n}^{\setminus n} \frac{d \log \tilde{Z}_n}{dm_f^{\setminus n}} \quad (\text{A.5})$$

$$\mathbf{V}_u = \mathbf{V}_u^{\setminus n} + \mathbf{V}_{uf_n}^{\setminus n} \frac{d^2 \log \tilde{Z}_n}{d(m_f^{\setminus n})^2} \mathbf{V}_{f_n u}^{\setminus n} \quad (\text{A.6})$$

Update: $\tilde{t}_n(u)^{\text{new}} = (Q^{\text{new}}/Q^{\setminus n})^{1/\alpha}$. Resulting in updated natural parameters:

$$\mathbf{T}_{1,n,\text{new}} = \mathbf{V}_u^{-1} \mathbf{m}_u - \mathbf{V}_u^{\setminus n, -1} \mathbf{m}_u^{\setminus n} \quad (\text{A.7})$$

$$\mathbf{T}_{2,n,\text{new}} = \mathbf{V}_u^{-1} - \mathbf{V}_u^{\setminus n, -1} \quad (\text{A.8})$$

Cycle over n and repeat cycling till convergence.

A.2 Rank 1 factors

However, $\mathbf{T}_{2,n}$ can be represented as rank 1, bringing further computational savings. For the derivation of how, the reader is referred to [55]. This is given as:

$$\mathbf{T}_{2,n} = w_n v_n^{-1} w_n^T \quad (\text{A.9})$$

Where $w_n = \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{K}_{\mathbf{uf}_n}$ and $v = \alpha(-d_2^{-1} - \mathbf{V}_{\mathbf{uf}_n}^{\setminus n} \mathbf{V}_{\mathbf{u}}^{\setminus n, -1} \mathbf{V}_{f_n \mathbf{u}}^{\setminus n})$ and $d_2 = \frac{d^2 \log \tilde{Z}_n}{d(m_f^{\setminus n})^2}$. This results in the first natural parameter as:

$$\mathbf{T}_{1,n} = w_n v_n^{-1} g_n \quad (\text{A.10})$$

Where $g_n = -\frac{d_1}{d_2} + \mathbf{K}_{f_n \mathbf{u}} \gamma^{\setminus n}$ and $d_1 = \frac{d \log \tilde{Z}_n}{d m_f^{\setminus n}}$. Resulting in an approximate factor with the form $t_n(u) = \mathcal{N}(\mathbf{K}_{f_n \mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{u}; g_n, v_n)$. Thus only g_n , v_n , γ and β need be updated and stored, resulting in significant computational savings. This requires some slight rejigging of the cavity and projection steps.

Cavity: Bui et al. Use an alternative deletion method here, likely to avoid explicitly parametrising on m_f and $V_f f$. Instead of subtracting the natural parameters, they note that the deletion step can also be thought of as a KL minimisation, and solved with the same moment matching procedure. Since it will be necessary, the log normaliser of the cavity is obviously $\log \tilde{Z}_n = \int \frac{q(f)}{t_n^\alpha(u)} df$. The cavity for γ and β are thus:

$$\gamma^{\setminus n} = \gamma + \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{uf}_n}^{\setminus n} \tilde{d}_1 \quad (\text{A.11})$$

$$\beta^{\setminus n} = \beta + \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{uf}_n}^{\setminus n} \tilde{d}_2 \mathbf{V}_{f_n \mathbf{u}}^{\setminus n} \mathbf{K}_{\mathbf{uu}}^{-1} \quad (\text{A.12})$$

Where:

$$\tilde{d}_2 = \frac{d^2 \log \tilde{Z}_n}{d m_{f_n}^2} = \left[\mathbf{K}_{f_n \mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{K}_{\mathbf{uf}_n} - \mathbf{K}_{f_n \mathbf{u}} \beta \mathbf{K}_{\mathbf{uf}_n} - \frac{v_n}{\alpha} \right] \quad (\text{A.13})$$

$$\tilde{d}_1 = \frac{d \log \tilde{Z}_n}{dm_{f_n}} = (\mathbf{K}_{f_n \mathbf{u}} \gamma - g_n) \tilde{d}_2 \quad (\text{A.14})$$

The full derivation of which the reader is again referred to the original text.

Projection: The projection equations from earlier need to be rewritten as:

$$\gamma = \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{m}_{\mathbf{u}} \quad (\text{A.15})$$

$$= \mathbf{K}_{\mathbf{uu}}^{-1} (\mathbf{m}_{\mathbf{u}}^{\setminus n} + \mathbf{V}_{\mathbf{uf}_n}^{\setminus n} d_1) \quad (\text{A.16})$$

$$= \gamma^{\setminus n} + \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{uf}_n}^{\setminus n} d_1 \quad (\text{A.17})$$

$$\beta = \mathbf{K}_{\mathbf{uu}}^{-1} (\mathbf{K}_{\mathbf{uu}} - \mathbf{V}_{\mathbf{u}}) \mathbf{K}_{\mathbf{uu}}^{-1} \quad (\text{A.18})$$

$$= \mathbf{K}_{\mathbf{uu}}^{-1} (\mathbf{K}_{\mathbf{uu}} - \mathbf{V}_{\mathbf{u}}^{\setminus n} - \mathbf{V}_{\mathbf{uf}_n}^{\setminus n} d_2 \mathbf{V}_{f_n \mathbf{u}}^{\setminus n}) \quad (\text{A.19})$$

$$= \beta^{\setminus n} - \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{uf}_n}^{\setminus n} d_2 \mathbf{V}_{f_n \mathbf{u}}^{\setminus n} \mathbf{K}_{\mathbf{uu}}^{-1} \quad (\text{A.20})$$

A.3 Full implementation

Algorithm 7 Full PEP implementation

Initialise parameters: $\{g_n = 0\}_{n=1}^N$, $\{v_n = \inf\}_{n=1}^N$, $\gamma = \mathbf{0}_{M \times 1}$ and $\beta = \mathbf{0}_{M \times M}$

For each $\mathbf{n}$ point, calculate:

1: **Cavity:**

$$\gamma^{\setminus n} = \gamma + \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{uf}_n} \tilde{d}_1 \quad (\text{A.21})$$

$$\beta^{\setminus n} = \beta + \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{uf}_n} \tilde{d}_2 \mathbf{V}_{f_n \mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \quad (\text{A.22})$$

2: **Projection 1:**

$$\gamma = \gamma^{\setminus n} + \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{uf}_n}^{\setminus n} d_1 \quad (\text{A.23})$$

$$\beta = \beta^{\setminus n} - \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{uf}_n}^{\setminus n} d_2 \mathbf{V}_{f_n \mathbf{u}}^{\setminus n} \mathbf{K}_{\mathbf{uu}}^{-1} \quad (\text{A.24})$$

3: **Projection 2:**

$$g_{n,\text{new}} = -\frac{d_1}{d_2} + \mathbf{K}_{f_n \mathbf{u}} \gamma^{\setminus n} \quad (\text{A.25})$$

$$v_{n,\text{new}} = -d_2^{-1} - \mathbf{V}_{f_n \mathbf{u}}^{\setminus n} \mathbf{V}_{\mathbf{uu}}^{\setminus n, -1} \mathbf{V}_{\mathbf{uf}_n}^{\setminus n} \quad (\text{A.26})$$

4: **Update:**

$$v_n = (v_{n,\text{new}}^{-1} + (1 - \alpha)v_n^{-1})^{-1} \quad (\text{A.27})$$

$$g_n = v_n(g_{n,\text{new}}v_{n,\text{new}}^{-1} + (1 - \alpha)g_nv_n^{-1}) \quad (\text{A.28})$$

Once each site has been visited, take a step in the hyperparameter space via $\frac{d}{d\theta} \sum_n^{i=1} \log C_i + \Phi(\theta) - \Phi(\theta^{(0)})$.

Repeat until convergence.

Where:

$$\mathbf{V}_{\mathbf{uf}_n} = \mathbf{K}_{\mathbf{uf}_n} - \mathbf{K}_{\mathbf{uu}} \beta \mathbf{K}_{\mathbf{uf}_n} \quad (\text{A.29})$$

$$\mathbf{V}_{\mathbf{uf}_n}^{\setminus n} = \mathbf{K}_{\mathbf{uf}_n} - \mathbf{K}_{\mathbf{uu}} \beta^{\setminus n} \mathbf{K}_{\mathbf{uf}_n} \quad (\text{A.30})$$

Moving this from a sequential to parallel implementation (where each n is PEP'd at once) has significant computational and stability savings, and is thus desirable. In practise, this is done by creating a tensor along n and leveraging linear algebra tools such as einsum in order to effectively calculate all PEPs at once.

The problem when implemented in parallel is that the projections for γ and β in Equation A.23 and Equation A.24 calculate a γ and β for each n . This is fine when looping over n , as γ and β are updated sequentially. However now they must be updated all at once, incorporating the information from all n .

This is on the surface simple, via a simple rearrange: $\gamma = \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m}_{\mathbf{u}}$ and $\beta = K_{\mathbf{uu}}^{-1}(K_{\mathbf{uu}} - \mathbf{V}_{\mathbf{u}})\mathbf{K}_{\mathbf{uu}}^{-1}$. Unfortunately, the problem lies in the fact that the approximate factors are stored as rank 1 natural parameters (v_n and g_n), and switching between natural and moment parameters is highly unstable. The crux of this instability lies in the inversion of $\mathbf{V}_{\mathbf{u}}^{-1} = \mathbf{K}_{\mathbf{uu}}^{-1} + \mathbf{T}_{\mathbf{2},\mathbf{f}}$.

To circumnavigate it, an inversion via QR decomposition is introduced. Here $\mathbf{L}_{\mathbf{2},\mathbf{f}}$ is the lower triangular cholesky of $\mathbf{T}_{\mathbf{2},\mathbf{f}}$ and $\mathbf{L}_{\mathbf{K}_{\mathbf{uu}}}$ is the lower triangular cholesky of $\mathbf{K}_{\mathbf{uu}}^{-1}$:

$$\begin{bmatrix} L_{Tu} \\ L_{2u}^\top \end{bmatrix} = QR \quad (\text{A.31})$$

Ensure R is positive in the diagonal (since R is not unique in the QR). Then:

$$L_v = R^T \quad (\text{A.32})$$

$$\mathbf{V}_{\mathbf{u}} = L_v^{-T} L_v^{-1} = (L_v L_v^T)^{-1} \quad (\text{A.33})$$

Where L_v^{-1} is calculated via stable triangular solve. This results in a parallel implementation as follows:

Where, $\mathbf{m}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}\mathbf{T}_{\mathbf{1},\mathbf{f}}$ and $\mathbf{V}_{\mathbf{u}} = (L_v L_v^T)^{-1}$.

A.4 Marginal likelihood optimisation

Recall that the marginal likelihood is given as:

$$L = \log \int \exp\left(\sum_{i=1}^{i=1} \log C_i + \theta^T \phi(x) - \Phi(\theta^{(0)})\right) \quad (\text{A.42})$$

$$= \sum_{i=1}^n \log C_i + \Phi(\theta) - \Phi(\theta^{(0)}) \quad (\text{A.43})$$

Algorithm 8 Parallel PEP implementation

Initialise parameters: $\{g_n = 0\}_{n=1}^N$, $\{v_n = \inf\}_{n=1}^N$, $\gamma = \mathbf{0}_{M \times 1}$ and $\beta = \mathbf{0}_{M \times M}$
Broadcast over n :

1: **Cavity:**

$$\gamma^{\setminus n} = \gamma + \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{u}f_n} \tilde{d}_1 \quad (\text{A.34})$$

$$\beta^{\setminus n} = \beta + \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{u}f_n} \tilde{d}_2 \mathbf{V}_{f_n \mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \quad (\text{A.35})$$

2: **Projection:**

$$g_{n,\text{new}} = -\frac{d1}{d2} + \mathbf{K}_{f_n \mathbf{u}} \gamma^{\setminus n} \quad (\text{A.36})$$

$$v_{n,\text{new}} = -d2^{-1} - \mathbf{V}_{f_n \mathbf{u}}^{\setminus n} \mathbf{V}_{\mathbf{uu}}^{\setminus n, -1} \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} \quad (\text{A.37})$$

3: **Update:**

$$v_n = (v_{n,\text{new}}^{-1} + (1 - \alpha)v_n^{-1})^{-1} \quad (\text{A.38})$$

$$g_n = v_n(g_{n,\text{new}}v_{n,\text{new}}^{-1} + (1 - \alpha)g_nv_n^{-1}) \quad (\text{A.39})$$

4: **Update γ and β :**

$$\gamma = \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{m}_{\mathbf{u}} \quad (\text{A.40})$$

$$\beta = K_{\mathbf{uu}}^{-1} (K_{\mathbf{uu}} - \mathbf{V}_{\mathbf{u}}) \mathbf{K}_{\mathbf{uu}}^{-1} \quad (\text{A.41})$$

Once each site has been visited, take a step in the hyperparameter space via $\frac{d}{d\theta} \sum_n^{i=1} \log C_i + \Phi(\theta) - \Phi(\theta^{(0)})$.
Repeat until convergence.

Where Φ denotes the log normaliser of the denoted distribution, and:

$$\log C_i = \log Z_i - \Phi(\theta) - \Phi(\theta)^{\setminus i} \quad (\text{A.44})$$

[55] provide a rank 1 derivation for the marginal likelihood, however through experience it is simpler and just as efficient to store all of the required PEP parameters as the PEP update happens, and then wrap the update inside an optimiser loop.

CHAINED PEP IMPLEMENTATION

Algorithm 9 Parallel PEP implementation

Initialise parameters: $\{g_n = 0\}_{n=1}^N$, $\{v_n = \inf\}_{n=1}^N$, $\gamma = \mathbf{0}_{M \times 1}$ and $\beta = \mathbf{0}_{M \times M}$

Broadcast over n :

1: **Cavity:**

$$\gamma^{\setminus n} = \gamma + \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{V}_{\mathbf{u}f_n} \tilde{d}_1 \quad (\text{B.1})$$

$$\beta^{\setminus n} = \beta + \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{V}_{\mathbf{u}f_n} \tilde{d}_2 \mathbf{V}_{f_n \mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \quad (\text{B.2})$$

2: **Projection:**

$$g_{n,\text{new}} = -\frac{d1}{d2} + \mathbf{K}_{f_n \mathbf{u}} \gamma^{\setminus n} \quad (\text{B.3})$$

$$v_{n,\text{new}} = -d2^{-1} - \mathbf{V}_{f_n \mathbf{u}}^{\setminus n} \mathbf{V}_{\mathbf{u}\mathbf{u}}^{\setminus n,-1} \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} \quad (\text{B.4})$$

3: **Update:**

$$v_n = (v_{n,\text{new}}^{-1} + (1 - \alpha)v_n^{-1})^{-1} \quad (\text{B.5})$$

$$g_n = v_n(g_{n,\text{new}}v_{n,\text{new}}^{-1} + (1 - \alpha)g_nv_n^{-1}) \quad (\text{B.6})$$

4: **Update γ and β :**

$$\gamma = \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{m}_{\mathbf{u}} \quad (\text{B.7})$$

$$\beta = K_{\mathbf{u}\mathbf{u}}^{-1} (K_{\mathbf{u}\mathbf{u}} - \mathbf{V}_{\mathbf{u}}) \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \quad (\text{B.8})$$

Once each site has been visited, take a step in the hyperparameter space via $\frac{d}{d\theta} \sum_n^{i=1} \log C_i + \Phi(\theta) - \Phi(\theta^{(0)})$.

Repeat until convergence.

Where, $\mathbf{m}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}} \mathbf{T}_{1,\mathbf{f}}$ and $\mathbf{V}_{\mathbf{u}} = (L_v L_v^T)^{-1}$.

In chained GPs, the key difference will lie in d_1 and d_2 **everything else** should

remain the same!. Recall that the log normaliser in the tilted classification case is:

$$\begin{aligned}
Z_i &= \int q(f_{y_i}) \prod_{k \neq y_i} q(f_k) \prod_{k \neq y_i} \mathcal{H}(\mathbf{f}_{y_i} - \mathbf{f}_k) d\mathbf{f} \\
&= \int \int q(f_{y_i}) \prod_{k \neq y_i} \int q(f_k) \mathcal{H}(\mathbf{f}_{y_i} - \mathbf{f}_k) d\mathbf{f}_k df_{y_i}
\end{aligned} \tag{B.9}$$

This results in the Gauss-Hermite quadrature implementation:

$$\left[\frac{1}{\sqrt{\pi}} \sum w_i \left[\prod_{k \neq y_i} \frac{1}{\sqrt{\pi}} \sum w_i \Theta[t_y - t_k] \right] \right]^\alpha \tag{B.10}$$

Where w_i are the Hermite weights, Θ is the heaviside step function and $t_k = \sqrt{2}\sigma_k x_i + \mu_k$, where x_i are the Hermite roots.

Again, note here that the cavity is undefined. Both t_y and t_k can be valid cavities, it just depends on which latent function is being visited (e.g. for the k^{th} latent function, some data points the cavity will be t_y and some it will be t_k). This is simple information to store and index into the correct cavity as necessary. Note that this only applies if the user is implementing a multi-class GP where all latent functions see all data. If, like in [129], the latent function only sees the training data associated with that class, then t_y will always be the cavity.

STREAMING PEP DERIVATION

C.1 $q_1(b)$

In order to illustrate the streaming PEP extension, it is necessary to follow the derivation of Bui et al. [157] for $q_1(b)$.

$$q_1^\backslash(f) = p(f)q_1^{1-\alpha}(\mathbf{b})q_2(\mathbf{b}) \quad (\text{C.1})$$

$$= p(f_{\neq \mathbf{a}, \mathbf{b}}|\mathbf{b})p(\mathbf{b})p(\mathbf{a}|\mathbf{b})q_1^{1-\alpha}(\mathbf{b})q_2(\mathbf{b}) \quad (\text{C.2})$$

$$\tilde{q}_1(f) = p(f_{\neq \mathbf{a}, \mathbf{b}}|\mathbf{b})p(\mathbf{b})p(\mathbf{a}|\mathbf{b})q_1^{1-\alpha}(\mathbf{b})q_2(\mathbf{b}) \frac{q(\mathbf{a})}{p(\mathbf{a}|\theta_{\text{old}})}^\alpha \quad (\text{C.3})$$

Where:

$$p(\mathbf{a}|\mathbf{b}) = \mathcal{N}(\mathbf{a}; \mathbf{K}_{\text{ab}}\mathbf{K}_{\text{bb}}^{-1}\mathbf{b}, \mathbf{K}_{\text{aa}} - \mathbf{K}_{\text{ab}}\mathbf{K}_{\text{bb}}^{-1}\mathbf{K}_{\text{ba}}) = \mathcal{N}(\mathbf{a}; \mathbf{W}_{\text{a}}\mathbf{b}, \mathbf{Q}_{\text{a}}) \quad (\text{C.4})$$

And noting that $q(\mathbf{a}) = \mathcal{N}(\mathbf{a}; \mathbf{m}_{\text{a}}, \mathbf{S}_{\text{a}})$ and $p(\mathbf{a}|\theta_{\text{old}}) = \mathcal{N}(\mathbf{a}; 0, \mathbf{K}_{\text{aa}})$, this leads to:

$$\frac{q(\mathbf{a})}{p(\mathbf{a}|\theta_{\text{old}})}^\alpha = C_1 \mathcal{N}(\mathbf{a}; \hat{\mathbf{m}}_{\mathbf{a}}, \hat{\mathbf{S}}_{\mathbf{a}}) \quad (\text{C.5})$$

Where:

$$\hat{\mathbf{m}}_{\mathbf{a}} = \mathbf{D}_{\mathbf{a}} \mathbf{S}_{\mathbf{a}}^{-1} \mathbf{m}_{\mathbf{a}} \quad (\text{C.6})$$

$$\hat{\mathbf{S}}_{\mathbf{a}} = \frac{1}{\alpha} \mathbf{D}_{\mathbf{a}} \quad (\text{C.7})$$

$$\mathbf{D}_{\mathbf{a}} = (\mathbf{S}_{\mathbf{a}}^{-1} - \mathbf{K}_{\mathbf{a}\mathbf{a}}^{-1})^{-1} \quad (\text{C.8})$$

Since the projection step is all gaussian, the new factor is simply equal to the marginalised tiled:

$$q_{1,\text{new}}(b) \propto \left[\int p(\mathbf{a}|\mathbf{b}) \frac{q(\mathbf{a})}{p(\mathbf{a}|\theta_{\text{old}})}^\alpha \right]^\frac{1}{\alpha} \quad (\text{C.9})$$

Letting $\Sigma_{\mathbf{a}} = \mathbf{D}_{\mathbf{a}} + \alpha \mathbf{Q}_{\mathbf{a}}$, and recalling that the new factor must be set to the power of $\frac{1}{\alpha}$ as well as divided by the cavity in the update step, this results in:

$$q_1(\mathbf{b}) = \mathcal{N}(\hat{\mathbf{m}}_{\mathbf{a}}; \mathbf{W}_{\mathbf{a}} \mathbf{b}, \Sigma_{\mathbf{a}}) \quad (\text{C.10})$$

In order to make integrating $q_1(b)$ into the PEP process easier, it is necessary to change the variable of $q_1(b)$ to b . Via a completion of the square in the exponent this results in (up to a normalising constant):

$$q_1(\mathbf{b}) = \mathcal{N}(\mathbf{b}; \mu_{\mathbf{b}}, \Sigma_{\mathbf{b}}) \quad (\text{C.11})$$

Where $\mu_{\mathbf{b}} = \Sigma_{\mathbf{b}} \mathbf{W}_{\mathbf{a}}^T \Sigma_{\mathbf{a}} \hat{\mathbf{m}}_{\mathbf{a}}$ and $\Sigma_{\mathbf{b}} = \mathbf{W}_{\mathbf{a}}^T \Sigma_{\mathbf{a}} \mathbf{W}_{\mathbf{a}}$

C.2 Full rank streaming PEP

Following Appendix A, the approximate factors for $q_2(b)$ in natural parameter form are $t_n(\mathbf{b}) = \mathcal{N}(\mathbf{b}; \mathbf{T}_{1,n}, \mathbf{T}_{2,n})$. This naturally leads to the approximate posterior over b with natural parameters $T_{1,b} = \sum_i T_{1,n} + T_1^{q_1(b)}$ and $T_{2,b} = K_{bb}^{-1} + \sum_i T_{2,n} + T_1^{q_1(2)}$. The moment parameters of $q_1(b)$ are given in Equation C.11. In mean and variance:

In mean and variance:

$$m_f = \mathbf{K}_{fu} \mathbf{K}_{uu}^{-1} \mathbf{T}_{2,f}^{-1} \mathbf{T}_{1,n} \quad (\text{C.12})$$

$$V_{ff'} = \mathbf{K}_{ff'} - \mathbf{Q}_{ff'} + \mathbf{K}_{fu} \mathbf{K}_{uu}^{-1} \mathbf{T}_{2,f}^{-1} \mathbf{K}_{uu}^{-1} \mathbf{K}_{uf'} \quad (\text{C.13})$$

In full rank, the PEP distribution then proceeds exactly as before except now $T_{1,b}$ and $T_{2,b}$ contain information on $q_1(b)$.

Cavity: For each n , the cavity is $q^{\setminus n}(f) = q(f)/\tilde{t}_n(u)^\alpha$. In natural parameters $\mathbf{T}_{1,f}^{\setminus n} = \mathbf{T}_{1,f} - \alpha \mathbf{T}_{1,n}$, $\mathbf{T}_{2,f}^{\setminus n} = \mathbf{T}_{2,f} - \alpha \mathbf{T}_{2,n}$. In mean and variance:

$$m_f^{\setminus n} = \mathbf{K}_{f_n u} \mathbf{K}_{uu}^{-1} \mathbf{T}_{2,f}^{\setminus n, -1} \mathbf{T}_{1,f}^{\setminus n} = \mathbf{K}_{f_n u} \gamma^{\setminus n} \quad (\text{C.14})$$

$$V_{ff}^{\setminus n} = \mathbf{K}_{f_n f_n} - \mathbf{Q}_{f_n f_n} + \mathbf{K}_{f_n u} \mathbf{K}_{uu}^{-1} \mathbf{T}_{2,f}^{\setminus n, -1} \mathbf{K}_{uu}^{-1} \mathbf{K}_{u f_n} = \mathbf{K}_{f_n f_n} - \mathbf{K}_{f_n u} \beta^{\setminus n} \mathbf{K}_{u f_n} \quad (\text{C.15})$$

Projection: Recalling that moments at u only need be matched and using the tilted property $\eta_{new} = \mathbb{E}_{\hat{P}(f)}[\phi_q(u)] = \frac{d}{d\theta^{\setminus i}} \log Z_n + \eta^{\setminus n}$ introduced in Equation 3.20 gives approximating factor a mean and variance:

$$\mathbf{m}_u = \mathbf{m}_u^{\setminus n} + \mathbf{V}_{u f_n}^{\setminus n} \frac{d \log \tilde{Z}_n}{d m_f^{\setminus n}} \quad (\text{C.16})$$

$$\mathbf{V}_u = \mathbf{V}_u^{\setminus n} + \mathbf{V}_{u f_n}^{\setminus n} \frac{d^2 \log \tilde{Z}_n}{d(m_f^{\setminus n})^2} \mathbf{V}_{f_n u}^{\setminus n} \quad (\text{C.17})$$

Update: $\tilde{t}_n(u)^{\text{new}} = (Q^{\text{new}}/Q^{\setminus n})^{1/\alpha}$. Resulting in updated natural parameters:

$$\mathbf{T}_{1,n,\text{new}} = \mathbf{V}_{\mathbf{u}}^{-1} \mathbf{m}_{\mathbf{u}} - \mathbf{V}_{\mathbf{u}}^{\setminus n,-1} \mathbf{m}_{\mathbf{u}}^{\setminus n} \quad (\text{C.18})$$

$$\mathbf{T}_{2,n,\text{new}} = \mathbf{V}_{\mathbf{u}}^{-1} - \mathbf{V}_{\mathbf{u}}^{\setminus n,-1} \quad (\text{C.19})$$

Cycle over n and repeat cycling till convergence.

BIBLIOGRAPHY

- [1] A. Rytter. Vibrational based inspection of civil engineering structures. 1993.
- [2] K. Worden, C. R. Farrar, G. Manson, and G. Park. The fundamental axioms of structural health monitoring. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2082):1639–1664, 2007.
- [3] S. W. Doebling, C. R. Farrar, M. B. Prime, *et al.* A summary review of vibration-based damage identification methods. *Shock and Vibration Digest*, 30(2):91–105, 1998.
- [4] O. S. Salawu. Detection of structural damage through changes in frequency: a review. *Engineering structures*, 19(9):718–723, 1997.
- [5] Z. Yang, H. Yang, T. Tian, D. Deng, M. Hu, J. Ma, D. Gao, J. Zhang, S. Ma, L. Yang, *et al.* A review on guided-ultrasonic-wave-based structural health monitoring: From fundamental theory to machine learning techniques. *Ultrasonics*, 133:107014, 2023.
- [6] Y. Song, S. Shan, Y. Zhang, and L. Cheng. Physics-guided neural network for structural health monitoring with lamb waves through boundary reflection elimination. *Structural Health Monitoring*, 2025.
- [7] K. Worden, G. Manson, and N. R. Fieller. Damage detection using outlier analysis. *Journal of Sound and vibration*, 229(3):647–667, 2000.
- [8] G. Park, A. C. Rutherford, H. Sohn, and C. R. Farrar. An outlier analysis framework for impedance-based structural health monitoring. *Journal of Sound and Vibration*, 286(1-2):229–250, 2005.

- [9] J. Toivola, M. A. Prada, and J. Hollmén. Novelty detection in projected spaces for structural health monitoring. In *International Symposium on Intelligent Data Analysis*, pages 208–219, 2010.
- [10] N. Dervilis, I. Antoniadou, R. J. Barthorpe, E. J. Cross, and K. Worden. Robust methods for outlier detection and regression for shm applications. *International Journal of Sustainable Materials and Structural Systems*, 2(1-2):3–26, 2015.
- [11] J. L. Tadjula, S. Kanakambaran, S. Kalyani, P. Rajagopal, and B. Srinivasan. Outlier analysis for defect detection using sparse sampling in guided wave structural health monitoring. *Structural Control and Health Monitoring*, 28(3):e2690, 2021.
- [12] B. Sohlberg and E. W. Jacobsen. Grey box modelling—branches and experiences. *IFAC Proceedings Volumes*, 41(2):11415–11420, 2008.
- [13] E. J. Cross, S. J. Gibson, M. R. Jones, D. J. Pitchforth, S. Zhang, and T. J. Rogers. Physics-informed machine learning for structural health monitoring. In *Structural health monitoring based on data science techniques*, pages 347–367. Springer, 2021.
- [14] K. Wang, M. Zeng, J. Wang, W. Shang, Y. Zhang, T. Luo, and A. W. Dowling. When physics-informed data analytics outperforms black-box machine learning: A case study in thickness control for additive manufacturing. *Digital Chemical Engineering*, 6:100076, 2023.
- [15] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [16] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.
- [17] J. Pateras, P. Rana, and P. Ghosh. A taxonomic survey of physics-informed machine learning. *Applied Sciences*, 13(12):6892, 2023.

- [18] C. Meng, S. Griesemer, D. Cao, S. Seo, and Y. Liu. When physics meets machine learning: A survey of physics-informed machine learning. *Machine Learning for Computational Science and Engineering*, 1(1):20, 2025.
- [19] F. Fuchs, D. Worrall, V. Fischer, and M. Welling. Se (3)-transformers: 3d rotation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33:1970–1981, 2020.
- [20] A. Bogatskiy, B. Anderson, J. Offermann, M. Roussi, D. Miller, and R. Kondor. Lorentz group equivariant neural network for particle physics. In *International Conference on Machine Learning*, pages 992–1002, 2020.
- [21] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2020.
- [22] P. Reiser, M. Neubert, A. Eberhard, L. Torresi, C. Zhou, C. Shao, H. Metni, C. van Hoesel, H. Schopmans, T. Sommer, *et al.* Graph neural networks for materials science and chemistry. *Communications Materials*, 3(1):93, 2022.
- [23] S. Cao. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34:24924–24940, 2021.
- [24] Q. Xiao and Z. Zeng. Lagrange stability for t-s fuzzy memristive neural networks with time-varying delays on time scales. *IEEE Transactions on Fuzzy Systems*, 26(3):1091–1103, 2017.
- [25] S. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.
- [26] L. Yang, X. Meng, and G. E. Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.
- [27] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

- [28] M. D. Champneys and T. J. Rogers. Bindy: Bayesian identification of nonlinear dynamics with reversible-jump markov-chain monte carlo. *Proceedings of the Royal Society A*, 481(2319):20240620, 2025.
- [29] Y. Chen, Y. Luo, Q. Liu, H. Xu, and D. Zhang. Symbolic genetic algorithm for discovering open-form partial differential equations (sga-pde). *Physical Review Research*, 4(2):023174, 2022.
- [30] Y. Shi, P. Wei, K. Feng, D.-C. Feng, and M. Beer. A survey on machine learning approaches for uncertainty quantification of engineering systems. *Machine Learning for Computational Science and Engineering*, 1(1):11, 2025.
- [31] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, *et al.* A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021.
- [32] Y. Gal, P. Koumoutsakos, F. Lanusse, G. Louppe, and C. Papadimitriou. Bayesian uncertainty quantification for machine-learned models in physics. *Nature Reviews Physics*, 4(9):573–577, 2022.
- [33] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun. Hands-on bayesian neural networks—a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17(2):29–48, 2022.
- [34] G. Shafer and V. Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.
- [35] V. Balasubramanian, S.-S. Ho, and V. Vovk. *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes, 2014.
- [36] L. L. Folgoc, V. Baltatzis, S. Desai, A. Devaraj, S. Ellis, O. E. M. Manzanera, A. Nair, H. Qiu, J. Schnabel, and B. Glocker. Is mc dropout bayesian? *arXiv preprint arXiv:2110.04286*, 2021.
- [37] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016.

- [38] J. Liu, J. Paisley, M.-A. Kioumourtzoglou, and B. Coull. Accurate uncertainty estimation and decomposition in ensemble learning. *Advances in neural information processing systems*, 32, 2019.
- [39] R. Rahaman *et al.* Uncertainty quantification and deep ensembles. *Advances in neural information processing systems*, 34:20063–20075, 2021.
- [40] Q. Regression. *Handbook of quantile regression*. CRC Press: Boca Raton, FL, USA, 2017.
- [41] N. Meinshausen and G. Ridgeway. Quantile regression forests. *Journal of Machine Learning Research*, 7(6), 2006.
- [42] A. F. Psaros, X. Meng, Z. Zou, L. Guo, and G. E. Karniadakis. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *Journal of Computational Physics*, 477:111902, 2023.
- [43] Z. Zou, X. Meng, and G. E. Karniadakis. Uncertainty quantification for noisy inputs–outputs in physics-informed neural networks and neural operators. *Computer Methods in Applied Mechanics and Engineering*, 433:117479, 2025.
- [44] A. J. Hughes, R. J. Barthorpe, N. Dervilis, C. R. Farrar, and K. Worden. A probabilistic risk-based decision framework for structural health monitoring. *Mechanical Systems and Signal Processing*, 150:107339, 2021.
- [45] A. Kamariotis, E. Chatzi, and D. Straub. A framework for quantifying the value of vibration-based structural health monitoring. *Mechanical Systems and Signal Processing*, 184:109708, 2023.
- [46] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.
- [47] J. T. Ormerod and M. P. Wand. Explaining variational approximations. *The American Statistician*, 64(2):140–153, 2010.
- [48] J. Grimmer. An introduction to bayesian inference via variational approximations. *Political Analysis*, 19(1):32–47, 2011.
- [49] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

- [50] J. O. Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.
- [51] E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. *Advances in neural information processing systems*, 18, 2005.
- [52] J. Quinonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6: 1939–1959, 2005.
- [53] M. Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pages 567–574, 2009.
- [54] J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.
- [55] T. D. Bui, J. Yan, and R. E. Turner. A unifying framework for gaussian process pseudo-point approximations using power expectation propagation. *Journal of Machine Learning Research*, 18(104):1–72, 2017.
- [56] M. Lázaro-Gredilla, J. Quinonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal. Sparse spectrum gaussian process regression. *The Journal of Machine Learning Research*, 11:1865–1881, 2010.
- [57] A. Solin and S. Särkkä. Hilbert space methods for reduced-rank gaussian process regression. *Statistics and Computing*, 30(2):419–446, 2020.
- [58] M. Lázaro-Gredilla and A. Figueiras-Vidal. Inter-domain gaussian processes for sparse inference using inducing features. *Advances in Neural Information Processing Systems*, 22, 2009.
- [59] C. Park and D. Apley. Patchwork kriging for large-scale gaussian process regression. *Journal of Machine Learning Research*, 19(7):1–43, 2018.
- [60] C. Rasmussen and Z. Ghahramani. Infinite mixtures of gaussian process experts. *Advances in neural information processing systems*, 14, 2001.
- [61] M. M. Noack, H. Krishnan, M. D. Risser, and K. G. Reyes. Exact gaussian processes for massive datasets via non-stationary sparsity-discovering kernels. *Scientific reports*, 13(1):3155, 2023.

- [62] C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.
- [63] A. Venkitaraman, S. Chatterjee, and P. Handel. Gaussian processes over graphs. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5640–5644, 2020.
- [64] M. Van der Wilk, C. E. Rasmussen, and J. Hensman. Convolutional gaussian processes. *Advances in neural information processing systems*, 30, 2017.
- [65] S. Remes, M. Heinonen, and S. Kaski. Non-stationary spectral kernels. *Advances in neural information processing systems*, 30, 2017.
- [66] M. M. Noack and J. A. Sethian. Advanced stationary and nonstationary kernel designs for domain-aware gaussian processes. *Communications in applied mathematics and computational science*, 17(1):131–156, 2022.
- [67] Y. Saatçi, R. D. Turner, and C. E. Rasmussen. Gaussian process change point models. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 927–934, 2010.
- [68] D. J. Pitchforth, M. R. Jones, and E. J. Cross. Physically-informed change-point kernels for variable levels of physical knowledge inclusion in gaussian processes. In *Proceedings of the 10th European Workshop on Structural Health Monitoring (EWSHM 2024)*, volume 29, 2024.
- [69] V. Tresp. Mixtures of gaussian processes. *Advances in neural information processing systems*, 13, 2000.
- [70] N. Dervilis, H. Shi, K. Worden, and E. Cross. Exploring environmental and operational variations in shm data using heteroscedastic gaussian processes. In *Dynamics of Civil Structures, Volume 2: Proceedings of the 34th IMAC, A Conference and Exposition on Structural Dynamics 2016*, pages 145–153, 2016.
- [71] B. S. Jensen, J. B. Nielsen, and J. Larsen. Bounded Gaussian process regression. In *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2013.
- [72] L. P. Swiler, M. Gulian, A. L. Frankel, C. Safta, and J. D. Jakeman. A survey of constrained Gaussian process regression: Approaches and implementation

challenges. *Journal of Machine Learning for Modeling and Computing*, 1(2), 2020.

- [73] A. Damianou and N. D. Lawrence. Deep gaussian processes. In *Artificial intelligence and statistics*, pages 207–215, 2013.
- [74] D. Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, 2014.
- [75] M. Jones, D. Pitchforth, and E. Cross. Partially structured gaussian processes for grey-box learning in shm. *STRUCTURAL HEALTH MONITORING 2023*, 2023.
- [76] S. Park and K. Jung. Gaussian process regression-based structural response model and its application to regional damage assessment. *ISPRS International Journal of Geo-Information*, 10(9):574, 2021.
- [77] L. Bull, T. J. Rogers, N. Dervilis, E. J. Cross, and K. Worden. A gaussian process form for population-based structural health monitoring. In *Proceedings of the 13th International Conference on Damage Assessment of Structures*, 2019.
- [78] Q.-A. Wang and Y.-Q. Ni. Measurement and forecasting of high-speed rail track slab deformation under uncertain shm data using variational heteroscedastic gaussian process. *Sensors*, 19(15):3311, 2019.
- [79] Q.-A. Wang, C. Zhang, Z.-G. Ma, and Y.-Q. Ni. Modelling and forecasting of shm strain measurement for a large-scale suspension bridge during typhoon events using variational heteroscedastic gaussian process. *Engineering Structures*, 251:113554, 2022.
- [80] M. R. Jones, T. J. Rogers, and E. J. Cross. Constraining gaussian processes for physics-informed acoustic emission mapping. *Mechanical Systems and Signal Processing*, 188:109984, 2023.
- [81] K. Worden, W. Becker, T. Rogers, and E. Cross. On the confidence bounds of gaussian process narx models and their higher-order frequency response functions. *Mechanical Systems and Signal Processing*, 104:188–223, 2018.

- [82] F. Tobar, T. D. Bui, and R. E. Turner. Learning stationary time series using gaussian processes with nonparametric kernels. *Advances in neural information processing systems*, 28, 2015.
- [83] O. Hamelijnck, A. Solin, and T. Damoulas. Physics-informed variational state-space gaussian processes. *Advances in Neural Information Processing Systems*, 37:98505–98536, 2024.
- [84] R. Frigola, Y. Chen, and C. E. Rasmussen. Variational gaussian process state-space models. *Advances in Neural Information Processing Systems*, 27, 2014.
- [85] J. Ko, D. J. Klein, D. Fox, and D. Haehnel. Gp-ukf: Unscented kalman filters with gaussian process prediction and observation models. In *2007 IEEE/RSJ international conference on intelligent robots and systems*, pages 1901–1907, 2007.
- [86] J. Hartikainen and S. Särkkä. Kalman filtering and smoothing solutions to temporal gaussian process regression models. In *2010 IEEE international workshop on machine learning for signal processing*, pages 379–384, 2010.
- [87] M. A. Alvarez, D. Luengo, and N. D. Lawrence. Linear latent force models using gaussian processes. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2693–2705, 2013.
- [88] T. J. Rogers and T. Friis. A latent restoring force approach to nonlinear system identification. *Mechanical Systems and Signal Processing*, 180:109426, 2022.
- [89] J. Alvarez-Montoya, M. Torres-Arredondo, and J. Sierra-Pérez. Gaussian process modeling for damage detection in composite aerospace structures by using discrete strain measurements. In *Proceedings of the 7th Asia-Pacific Workshop on Structural Health Monitoring, APWSHM*, pages 710–718, 2018.
- [90] K. Vuckovic and S. Prakash. Remaining useful life prediction using gaussian process regression model. In *Annual conference of the phm society*, volume 14, 2022.
- [91] L. Li, P. Wang, K.-H. Chao, Y. Zhou, and Y. Xie. Remaining useful life prediction for lithium-ion batteries based on gaussian processes mixture. *PloS one*, 11(9):e0163004, 2016.

- [92] Z. Xu, Y. Guo, and J. H. Saleh. Accurate remaining useful life prediction with uncertainty quantification: a deep learning and nonstationary gaussian process approach. *IEEE Transactions on Reliability*, 71(1):443–456, 2021.
- [93] E. Papatheou, N. Dervilis, A. E. Maguire, I. Antoniadou, and K. Worden. A performance monitoring approach for the novel Lillgrund offshore wind farm. *IEEE Transactions on Industrial Electronics*, 62(10):6636–6644, 2015.
- [94] C. Carrillo, A. O. Montaña, J. Cidrás, and E. Díaz-Dorado. Review of power curve modelling for wind turbines. *Renewable and Sustainable Energy Reviews*, 21:572–581, 2013.
- [95] D. Gielen, F. Boshell, D. Saygin, M. D. Bazilian, N. Wagner, and R. Gorini. The role of renewable energy in the global energy transformation. *Energy Strategy Reviews*, 24:38–50, 2019.
- [96] J. F. Manwell, J. G. McGowan, and A. L. Rogers. *Wind energy explained: theory, design and application*. John Wiley & Sons, 2010.
- [97] R. K. Pandit and D. Infield. Scada-based wind turbine anomaly detection using Gaussian process models for wind turbine condition monitoring purposes. *IET Renewable Power Generation*, 12(11):1249–1255, 2018.
- [98] R. K. Pandit and D. Infield. Comparative analysis of Gaussian process power curve models based on different stationary covariance functions for the purpose of improving model accuracy. *Renewable Energy*, 140:190–202, 2019.
- [99] T. J. Rogers, P. Gardner, N. Dervilis, K. Worden, A. E. Maguire, E. Papatheou, and E. J. Cross. Probabilistic modelling of wind turbine power curves with application of heteroscedastic Gaussian process regression. *Renewable Energy*, 148:1124–1136, 2020.
- [100] M. Lydia, S. S. Kumar, A. I. Selvakumar, and G. E. P. Kumar. A comprehensive review on wind turbine power curve modeling techniques. *Renewable and Sustainable Energy Reviews*, 30:452–460, 2014.
- [101] S. Ge, M. J. Zuo, and Z. Tian. Wind turbine power output estimation with probabilistic power curves. In *2020 Asia-Pacific International Symposium on Advanced Reliability and Maintenance Modeling (APARM)*, pages 1–6, 2020.

- [102] E. Yun and J. Hur. Probabilistic estimation model of power curve to enhance power output forecasting of wind generating resources. *Energy*, 223:120000, 2021.
- [103] E. Snelson, Z. Ghahramani, and C. Rasmussen. Warped gaussian processes. *Advances in neural information processing systems*, 16, 2003.
- [104] A. D. Saul, J. Hensman, A. Vehtari, and N. D. Lawrence. Chained Gaussian processes. In *Artificial Intelligence and Statistics*, pages 1431–1440, 2016.
- [105] M. Lázaro-Gredilla and M. K. Titsias. Variational heteroscedastic Gaussian process regression. In *ICML*, 2011.
- [106] M. A. Alvarez and N. D. Lawrence. Computationally efficient convolved multiple output Gaussian processes. *The Journal of Machine Learning Research*, 12: 1459–1500, 2011.
- [107] A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrà, Z. Ghahramani, and J. Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18 (40):1–6, 2017.
- [108] C. E. Rasmussen and H. Nickisch. Gaussian processes for machine learning (gpml) toolbox. *J. Mach. Learn. Res.*, 11:3011–3015, 2010. ISSN 1532-4435.
- [109] D. Gamerman and H. F. Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. CRC Press, 2006.
- [110] 3 - data preprocessing. In J. Han, M. Kamber, and J. Pei, editors, *Data Mining (Third Edition)*, The Morgan Kaufmann Series in Data Management Systems, pages 83–124. Morgan Kaufmann, Boston, third edition edition, 2012. ISBN 978-0-12-381479-1.
- [111] D. Cornford, I. Nabney, and C. Williams. Adding constrained discontinuities to Gaussian process models of wind fields. *Advances in Neural Information Processing Systems*, 11, 1998.
- [112] W. S. L. Wah, Y.-T. Chen, and J. S. Owen. A regression-based damage detection method for structures subjected to changing environmental and operational conditions. *Engineering Structures*, 228:111462, 2021.

- [113] I. Laory, T. N. Trinh, D. Posenato, and I. F. Smith. Combined model-free data-interpretation methodologies for damage detection during continuous monitoring of structures. *Journal of Computing in Civil Engineering*, 27(6): 657–666, 2013.
- [114] S. Sharma and S. Sen. Real-time structural damage assessment using lstm networks: regression and classification approaches. *Neural Computing and Applications*, 35(1):557–572, 2023.
- [115] D. Tibaduiza, M. Á. Torres-Arredondo, J. Vitola, M. Anaya, and F. Pozo. A damage classification approach for structural health monitoring using machine learning. *Complexity*, 2018(1):5081283, 2018.
- [116] W. Nick, J. Shelton, K. Asamene, and A. C. Esterline. A study of supervised machine learning techniques for structural health monitoring. *MAICS*, 1353: 36, 2015.
- [117] A. Entezami, H. Shariatmadar, and H. Sarmadi. Condition assessment of civil structures for structural health monitoring using supervised learning classification methods. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 44(Suppl 1):51–66, 2020.
- [118] L. A. Bull, K. Worden, and N. Dervilis. Towards semi-supervised and probabilistic classification in structural health monitoring. *Mechanical Systems and Signal Processing*, 140:106653, 2020.
- [119] V.-H. Dang, K. Le-Nguyen, and T.-T. Nguyen. Semi-supervised vibration-based structural health monitoring via deep graph learning and contrastive learning. In *Structures*, volume 51, pages 158–170, 2023.
- [120] T. Rogers, K. Worden, R. Fuentes, N. Dervilis, U. Tygesen, and E. Cross. A bayesian non-parametric clustering approach for semi-supervised structural health monitoring. *Mechanical Systems and Signal Processing*, 119:100–119, 2018.
- [121] R. S. Battu, K. Agathos, J. M. L. Monsalve, K. Worden, and E. Papatheou. Combining transfer learning and numerical modelling to deal with the lack of training data in data-based shm. *Journal of Sound and Vibration*, 595:118710, 2025.

- [122] J. S. Nielsen, D. Tcherniak, and M. D. Ulriksen. A case study on risk-based maintenance of wind turbine blades with structural health monitoring. *Structure and Infrastructure Engineering*, 17(3):302–318, 2021.
- [123] H. Yu, A. H. Seno, Z. Sharif Khodaei, and M. F. Aliabadi. Structural health monitoring impact classification method based on bayesian neural network. *Polymers*, 14(19):3947, 2022.
- [124] M. Chai, Z. Gao, Y. Li, Z. Zhang, Q. Duan, and R. Chen. An approach for identifying corrosion damage from acoustic emission signals using ensemble empirical mode decomposition and linear discriminant analysis. *Measurement Science and Technology*, 33(6):065018, 2022.
- [125] Y. Wang, P. Wang, K. Ding, H. Li, J. Zhang, X. Liu, Q. Bai, D. Wang, and B. Jin. Pattern recognition using relevant vector machine in optical fiber vibration sensing system. *IEEE Access*, 7:5886–5895, 2019.
- [126] A. Naish-Guzman and S. Holden. The generalized fitc approximation. *Advances in neural information processing systems*, 20, 2007.
- [127] H. Nickisch, C. E. Rasmussen, *et al.* Approximations for binary gaussian process classification. *Journal of Machine Learning Research*, 9(10):2035–2078, 2008.
- [128] M. Kuss, C. E. Rasmussen, and R. Herbrich. Assessing approximate inference for binary gaussian process classification. *Journal of Machine Learning Research*, 6(10), 2005.
- [129] C. Villacampa-Calvo and D. Hernandez-Lobato. Alpha divergence minimization in multi-class gaussian process classification. *Neurocomputing*, 378:210–227, 2020.
- [130] M. Seeger. Expectation propagation for exponential families. 2005.
- [131] T. P. Minka. The ep energy function and minimization schemes. Technical report, 2001.
- [132] T. Minka. Power ep. Technical report, 2004.

- [133] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. US Government Printing Office, 1948.
- [134] M. Seeger, F. Steinke, and K. Tsuda. Bayesian inference and optimal design in the sparse linear model. In *Artificial Intelligence and Statistics*, pages 444–451, 2007.
- [135] Y. Qi, A. H. Abdel-Gawad, and T. P. Minka. Sparse-posterior gaussian processes for general likelihoods. In *Proceedings of the 26th conference on Uncertainty in Artificial Intelligence*, pages 450–457, 2010.
- [136] A. G. d. G. Matthews, J. Hensman, R. Turner, and Z. Ghahramani. On sparse variational methods and the kullback-leibler divergence between stochastic processes. In *Artificial Intelligence and Statistics*, pages 231–239, 2016.
- [137] A. Schwaighofer and V. Tresp. Transductive and inductive methods for approximate gaussian process regression. *Advances in Neural Information Processing Systems*, 15, 2002.
- [138] E. Figueiredo, G. Park, J. Figueiras, C. Farrar, and K. Worden. Structural health monitoring algorithm comparisons using standard data sets. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2009.
- [139] T. J. Rogers. *Towards Bayesian system identification: with application to SHM of offshore structures*. PhD thesis, University of Sheffield, 2019.
- [140] L. A. Bull. *Towards probabilistic and partially-supervised structural health monitoring*. PhD thesis, University of Sheffield, 2019.
- [141] X. Zhu, P. Zhang, X. Lin, and Y. Shi. Active learning from data streams. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 757–762, 2007.
- [142] M. Wang, F. Min, Z.-H. Zhang, and Y.-X. Wu. Active learning through density clustering. *Expert systems with applications*, 85:305–317, 2017.

- [143] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.
- [144] K. Kowalska and L. Peel. Maritime anomaly detection using gaussian process active learning. In *2012 15th International Conference on Information Fusion*, pages 1164–1171, 2012.
- [145] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *2007 IEEE 11th international conference on computer vision*, pages 1–8, 2007.
- [146] H. T. Nguyen and A. Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine Learning*, page 79, 2004.
- [147] S. Sinha, S. Ebrahimi, and T. Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5972–5981, 2019.
- [148] M. Buisson-Fenet, F. Solowjow, and S. Trimpe. Actively learning gaussian process dynamics. In *Learning for Dynamics and Control*, pages 5–15, 2020.
- [149] G. Zhao, E. Dougherty, B.-J. Yoon, F. Alexander, and X. Qian. Efficient active learning for gaussian process classification by error reduction. *Advances in Neural Information Processing Systems*, 34:9734–9746, 2021.
- [150] J. Schreiter, D. Nguyen-Tuong, M. Eberts, B. Bischoff, H. Markert, and M. Toussaint. Safe exploration for active learning with gaussian processes. In *Machine Learning and Knowledge Discovery in Databases: European Conference, Part III 15*, pages 133–149, 2015.
- [151] C. Feng, M.-Y. Liu, C.-C. Kao, and T.-Y. Lee. Deep active learning for civil infrastructure defect detection and classification. In *Computing in civil engineering 2017*, pages 298–306. 2017.
- [152] L. Bull, K. Worden, T. Rogers, C. Wickramarachchi, E. Cross, T. McLeay, W. Leahy, and N. Dervilis. A probabilistic framework for online structural health monitoring: active learning from machining data streams. In *Journal of Physics: Conference Series*, volume 1264, page 012028, 2019.

- [153] L. A. Bull, N. Dervilis, K. Worden, E. J. Cross, and T. J. Rogers. A sampling-based approach for information-theoretic inspection management. *Proceedings of the Royal Society A*, 478(2262):20210790, 2022.
- [154] A. J. Hughes, L. A. Bull, P. Gardner, R. J. Barthorpe, N. Dervilis, and K. Worden. On risk-based active learning for structural health monitoring. *Mechanical Systems and Signal Processing*, 167:108569, 2022.
- [155] F. B. Smith, A. Kirsch, S. Farquhar, Y. Gal, A. Foster, and T. Rainforth. Prediction-oriented bayesian active learning. In *International conference on artificial intelligence and statistics*, pages 7331–7348, 2023.
- [156] A. Ranganathan, M.-H. Yang, and J. Ho. Online sparse gaussian process regression and its applications. *IEEE Transactions on Image Processing*, 20(2):391–404, 2010.
- [157] T. D. Bui, C. Nguyen, and R. E. Turner. Streaming sparse gaussian process approximations. *Advances in Neural Information Processing Systems*, 30, 2017.
- [158] C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.
- [159] T. D. Bui. Likelihood approximations via gaussian approximate inference. *arXiv preprint arXiv:2410.20754*, 2024.
- [160] P. Moreno-Muñoz, A. Artés, and M. Alvarez. Heterogeneous multi-output gaussian process prediction. *Advances in neural information processing systems*, 31, 2018.
- [161] Y. C. Ng, N. Colombo, and R. Silva. Bayesian semi-supervised learning with graph gaussian processes. *Advances in Neural Information Processing Systems*, 31, 2018.