# Self-Explaining Variational Gaussian Processes for Transparency and Modelling of Prior Knowledge

Sarem Seitz

*University of Bamberg, An der Weberei 5, 96049 Bamberg, Germany*

**Abstract**

Bayesian methods have become a popular way to incorporate prior knowledge and a notion of uncertainty into machine learning models. At the same time, the complexity of modern machine learning makes it challenging to comprehend a model's reasoning process, let alone express specific prior assumptions in a rigorous manner. While primarily interested in the former issue, recent developments in transparent machine learning could also broaden the range of prior information that we can provide to complex Bayesian models. Inspired by the idea of self-explaining models, this paper introduces a corresponding concept for variational Gaussian Processes. While the proposed method is inherently transparent, the bayesian nature of the underlying Gaussian Process allows to incorporate prior knowledge about the underlying problem. In one sentence, the goal is to let the human expert explain how to solve a supervised learning problem in a language that both the model and the user understand. For now, we evaluate these capabilities on simple problems.

**Keywords**

Explainable Machine Learning, Bayesian Machine Learning, Gaussian Processe

## 1. Introduction

As the field of explainable machine learning is getting more and more traction, methods that were once incomprehensible to human users are beginning to become more transparent. While a general solution to the challenge of humanly tangible, yet sufficiently complex models still seems to be far off in the future, recent developments have yielded promising results. The primary advantages of interpretable models are, as noted in [1], (scientific) understanding on the one hand and on the other hand safety, especially operational and ethical safety.

Typically, interpretable methods aim to encode the implicit decision process of a black-box[1] in a representation that humans can understand and evaluate. This begs, in particular, the following research question: *Can we use an interpretable representation to induce existing knowledge about a complex modelling problem into a target model?*. While there is no normed definition of what a 'complex' modelling problem actually is, we can loosely define it as a task that, at least for now, typically needs to be handled by a black-box. Under this definition, standard linear regression

*Proceedings of the Second International Workshop on Explainable and Interpretable Machine Learning (XI-ML 2022) co-located with the 45rd German Conference on Artificial Intelligence (KI 2022), September 20, 2022, Trier (Virtual), Germany*

sarem.seitz@uni-bamberg.de (S. Seitz)

[1] in the context of machine learning, the term 'black-box' is commonly used for models whose decision process cannot be understood even by experts in the modeled domain

models lack the complexity characteristic as regression coefficients can be used by humans to understand the underlying decision process. On the other hand, we can encode existing knowledge about the modelling task in such models. This is usually done either via Bayesian priors over the coefficients or constrained optimization.

**Problem.** For complex machine learning models, it is usually not as straightforward to encode existing knowledge as in the linear example. Our goal for this paper is therefore twofold. First, we want to derive an approach that can model complex problems in a transparent manner. Subsequently, we want to be able to exploit the transparent representation to encode existing prior knowledge and use it in the model's training procedure.

Let us split these goals further into three concrete requirements: *Transparency* - The solution needs to provide insights into its decision process that can be understood by a sufficiently trained domain expert. *Flexibility* - In order to be useful for complex problems, the proposed approach needs to be flexible, i.e. be able to handle a broad range of functional relations between input and target variables. *Teachability* - Finally, we need to be able to use an interpretable representation of existing knowledge and align the model's decision process with that knowledge.

Apart from that, a practically relevant solution should also be able to handle with real-world problem. This implies, in particular, that scalability to reasonably large datasets has to be possible.

**Contribution.** To achieve the above desiderata, this paper proposes *self-explaining variational GPs* (SEVGPs). The *self-explanatory* component aims to solve the transparency requirement. By using the right kernel functions, GPs can handle complex functional relations as demanded under the *flexibility* specification. Since GPs are part of the family of Bayesian models, they are naturally able to incorporate prior knowledge, i.e. they also fall under the idea *teachability*. The primary limitation in this regard is the representations in which we are able to express our prior knowledge.

While GP models in their original form are unable to deal with large datasets, there exist many scalable solutions nowadays. Our approach will apply the concept of sparse variational GPs (SVGPs) in order to achieve scalability to big data problems as well.

**Related work.** The results of [2, 3, 4] directly inspired this approach from an explainability and transparency point of view. In fact, the approach [3] relates to this work in a similar way as GPs relate to SVGPs. However, as will be seen, this paper does not merely provide a scalable variant of the former work via SVGPs.

In addition to the transparency component, our aim is to also create a tool that can be used to provide human expert knowledge via transparent representations. [5, 6, 7, 8] all discuss the potentially beneficial role of expert and domain knowledge in machine learning, yet either mention Bayesian methods only briefly or not at all. Nevertheless, Bayesian non-parametrics have already been applied successfully in countless classical statistical modeling problems with an emphasis on incorporating prior knowledge - see [9] for a variety of examples.

Recent work on functional variational inference as discussed particularly in [10, 11] could be a fruitful step towards a synthesis of meaningful prior models and modern Machine Learning architectures.

**Outline.** In the next section we conduct a brief recap on transparent machine learning with focus on varying coefficient and self-explaining methods. Thereafter, we proceed similarly for GPs and SVGPs. The fourth section marks the main contribution of this paper where the primary formulas of our approach are exposed and discussed. Experimental validation of the approach is conducted in section five. Finally, we discuss limitations and potential extensions of our methodology in the last section. Proofs and derivations, as well as additional details can be found in the appendix.

## 2. Transparent Machine Learning

In regards to transparency in machine learning, terms like *interpretable machine learning* or *explainable artificial intelligence* (XAI) have become quite widespread and popular. However, up to this date, there is still no uniquely accepted definition for many terms in this field. In our context, where we consider supervised learning problems, we will use the following definitions of interpretation and explanation from [12]:

**Definition 1.** *An* ***interpretation*** *is the mapping of an abstract concept into a domain that the human can make sense of. An* ***explanation*** *is the collection of features of the interpretable domain, that have contributed for a given example to produce a decision.*

The corresponding authors particularly name images and text as *interpretable* domains. *Explanations*, on the other hand, could be visualizations that highlight image regions or certain words that contributed in favour of or against a given decision.

As we will see, it makes sense to allow for explanations to also quantify the **strength** of contribution per interpretable feature. For example, consider a fixed grey-scale image and denote the corresponding vector of the $D$ image's pixels, encoded in the range $[0, 1]$, as $x \in [0, 1]^D$. By introducing a coefficient vector $\beta \in \mathbb{R}^D$ with the same dimensionality as $x$, we can derive the usual linear model for a single example

$$y = x^T \beta \tag{1}$$

The outcome scalar $y \in \mathbb{R}$ could then be mapped to a valid probability via some monotone, increasing function $\sigma : \mathbb{R} \mapsto (0, 1)$. This obviously results in a binary classification problem. Notice that we can equally write (1) as the sum of pixel-coefficient products, i.e.

$$y = \sum_{d=1}^{D} x_{(d)} \beta_{(d)} \tag{2}$$

With respect to the mentioned classification problem, (2) now implies the following logic for quantifiable explanations:

Image pixels where $x_{(d)}\beta_{(d)} > 0$ contribute towards a positive classification whereas pixels where $x_{(d)}\beta_{(d)} < 0$ contribute towards a negative classification[2]. Also pixels where $|x_{(d)}\beta_{(d)}|$ close to zero provide almost no contribution to the outcome and pixel where $|x_{(d)}\beta_{(d)}|$ is large provide large contribution. From now on, let us explicitly name the product $x_{(d)}\beta_{(d)}$ as the **contribution** of the $d$-th feature.

[2] Notice that we might have to add a constant term to this representation in order to account for cases where $x_{(d)} = 0$. Otherwise, the contribution of those features will always be zero. For simplicity though, we will only consider the model as in (2).

Obviously, the contribution of each pixel must be able to differ for different images. Even under a mere translation of some baseline image, the corresponding contributions must also shift accordingly. As a result, the static coefficients as implied in (1) are unrealistic when considering multiple, different images. Rather, the $\beta$ should vary with the given input image, i.e.

$$y = x^T \beta(x) \tag{3}$$

Equation (3) now implies that the coefficient vector is a function of the input vector; in the context of the above grey-scale input: $\beta : [0,1]^D \mapsto \mathbb{R}^D$. At this point, we should reiterate that this formulation is not restricted to image classification but can easily be extended to other domains that permit a similar representation of its input features. In fact, models like (3) were proposed as early as in [13] for classical statistical regression problems with tabular data.

More recent work around these 'varying coefficient' models has been done in [2], who considered them, under the umbrella term *self-explaining models*, for modern machine learning problems like image or text classification. The most important novelty is the replacement of regression splines to model $\beta(\cdot)$ with a feedforward neural network with $D$ output neurons.

## 3. Gaussian Processes

The building blocks of GPs, see [14], are a prior distribution over functions, $p(f)$, and a likelihood $p(y|f)$. Using Bayes' law, we are interested in a posterior distribution $p(f|y)$ obtained as

$$p(f|y) = \frac{p(y|f)p(f)}{p(y)}. \tag{4}$$

The prior distribution is a Gaussian Process, fully specified by $m(\cdot) : \mathcal{X} \mapsto \mathbb{R}$, typically $m(x) = 0$, and covariance kernel function $k(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}_0^+$:

$$p(f) = \mathcal{GP}(f|m(\cdot), k(\cdot,\cdot)) \tag{5}$$

We assume the input domain for $f$ to be a bounded subset of the real numbers, $\mathcal{X} \subset \mathbb{R}^D$. Technically, this invalidates (5) as $f$ then becomes an infinite-dimensional object for which a probability density does not exist. Since we are dealing with finite-dimensional datasets only, this techincal inaccuracy does not pose a problem in our further treatment. To exemplify our focus on finite dimensional marginals, we will make heavy use of subscripts to match inter-related objects.

Most importantly, we denote the $N \times D$ matrix of input data-points as $X_N$ and the corresponding marginal GP output as $f_N = f(X_N)$. This allows us to discuss GPs either at their multivariate Gaussian marginal output or as actual random functions. We will switch between both concepts depending on the situation.

A common choice for $k(\cdot,\cdot)$ is the ARD[3]-kernel

$$k_{ARD}(x, x') = \theta \cdot exp(-0.5(x - x')\Sigma(x - x'))) \tag{6}$$

[3] **A**utomatic **R**elevance **D**etermination

where $\Sigma = diag(l_1^2, ..., l_K^2)$ is a diagonal matrix with entries in $\mathbb{R}_0^+$ and $\theta > 0$. For $K = 1$, (6) is equivalent to an SE[4]-kernel. We denote by $K$ the positive semi-definite *Gram-Matrix*, obtained as $K_{(ij)} = k(x_i, x_j)$, $x_i$ the $i$-th row of training input matrix $X_N$. As before, we denote the kernel gram-matrix belonging to $X_N$ as $K_{NN}$ and a potential mean vector as $m_N = m(X_N)$.

Provided that $p(y_N|f_N) = \prod_{i=1}^{N} \mathcal{N}(y_i|f_i, \sigma^2)$, i.e. training observations $y_N$ are i.i.d. univariate Gaussian conditioned on $f$, it is possible to directly calculate a corresponding posterior distribution for new inputs $X_*$ as

$$p(f_*|y_N) = \mathcal{N}(f_*|\Lambda_{*N} y_N, K_{**} - \Lambda_{*N}(K_{NN} + I\sigma^2)\Lambda_{*N}^T) \tag{7}$$

where $\Lambda_{*N} = K_{*N}(K_{NN} + I\sigma^2)^{-1}$, $K_{*N,(ij)} = k(x_i^*, x_j)$, $K_{**,(ij)} = k(x_i^*, x_j^*)$; $I$ is the identity matrix with according dimension.

In order to make GPs feasible for large datasets, the work of [15, 16, 17] developed and refined Sparse Variational Gaussian Processes (SVGPs). SVGPs, introduce a set of $M$ so called inducing locations $Z_M \subset \mathcal{X}$ and corresponding inducing variables $f_M$. The resulting posterior distribution, $p(f, f_M|y)$, is then approximated through a variational distribution $q(f, f_M) = p(f|f_M)q(f_M)$ - often $q(f_M) = \mathcal{N}(f_M|a, S), S = LL^T$ - by maximizing the *evidence lower bound* (ELBO):

$$ELBO = \sum_{i=1}^{N} \mathbb{E}_{p(f|f_M)q(f_M)}\left[\log p(y_i|f_i)\right] - KL(\mathcal{N}(a, S)||\mathcal{N}(m_M, K_{MM})) \tag{8}$$

where $KL(\mathcal{N}(\cdot,\cdot)||\mathcal{N}(\cdot,\cdot))$ denotes the KL-divergence between two (multivariate) Normal distributions. Finally, let us recall the following distributional properties of the marginal variational posterior process $q(\tilde{f}_*) = \int p(f_*|f_M)q(f_M)df_M$:

$$q(f_*) = \mathcal{N}(f_*|\tilde{\Lambda}_{*M} a, K_{**} - \tilde{\Lambda}_{*M}(K_{MM} - S)\tilde{\Lambda}_{*M}) \tag{9}$$

where $\tilde{\Lambda}_{*M} = K_{*M}K_{MM}^{-1}$. Also, we will write $\tilde{m}_* := \tilde{\Lambda}_{*M} a$ and $\tilde{K}_{**} := K_{**} - \tilde{\Lambda}_{*M}(K_{MM} - S)\tilde{\Lambda}_{*M}$. If two input matrices, $x_i$ and $x_j$ each consist of a single datapoint, $\tilde{m}_i, \tilde{m}_j$ and $\tilde{K}_{ij}$ can be viewed as the mean and kernel functions of the variational GP, evaluated at $x_i$ and $x_j$. We then denote the implicit GP mean and kernel functions as $\tilde{m}(\cdot) = \tilde{\Lambda}_{\cdot M} a$ and $\tilde{k}(\cdot,\cdot) = K_{\cdot\cdot} - \tilde{\Lambda}_{\cdot M}(K_{MM} - S)\tilde{\Lambda}_{\cdot M}^T$.

This allows us to hide the underlying dependencies on $a$ and $S$ in our notation and treat the variational GP as a separate entity from the original GP whose posterior distribution we are trying to approximate.

## 4. Self-explaining variational posterior distributions

The preceding two sections easily motivate the replacement of the feedforward neural network in self-explaining models by a GP model. For a given matrix of training data $X_N$ and target vector $y_N$, we obtain the following likelihood model:

[4] **S**quared **E**xponential

$$p(y_N|f^1,...,f^D;X_N) = p(y_N|X_N \cdot f^{1,D}(X_N)) \tag{10}$$

where "$\cdot$" means matrix multiplication for clarity (we will omit the "$\cdot$" from now),

$$f^{1,D}(X_N) = \begin{bmatrix} f^1(X_N)^T \\ \vdots \\ f^D(X_N)^T \end{bmatrix}$$

and we explicitly included the input matrix $X_N$ to exemplify the relation to self-explaining models. Also, let us require independence between the individual GPs. Now, we are dealing with a linear combination of $D$ independent GPs instead of a single one. Combining (10) and the concept of SVGPs, we can introduce $D$ variational processes and approximate the respective varying-coefficient GPs:

This directly implies the following ELBO:

$$ELBO = \mathbb{E}_{q(f^{1,D})}\left[\log p\left(y_N|X_N^T f_N^{1,D}\right)\right] - \sum_{d=1}^{D} KL(\mathcal{N}(a^d, S^d)||\mathcal{N}(m_M^d, K_{MM}^d)) \tag{11}$$

where $X_i$ denotes the $i$-th row of $X_N$. The derivation of (11) can be found in **Appendix A**. Notice that we now have $D$ sets of inducing variables, $I_{M^d}$. Obtaining a posterior predictive distribution for a Gaussian likelihood is also straightforward under this model:

$$\begin{aligned} p(y_*|X_*) &= \int p(y_*|X_* f_*^{1,D}) q(f_*^{1,D}) df_*^{1,D} \\ &= \mathcal{N}\left(y_* \middle| \sum_{d=1}^{D} X_*^d \odot \tilde{m}_*^d, \sum_{d=1}^{D} diag\left(X_*^d \left(X_*^d\right)^T \odot \tilde{K}_{**}^d\right) \odot I + \sigma^2 \cdot I\right) \end{aligned} \tag{12}$$

where $\odot$ denotes element-wise multiplication, $I$ is a unit-diagonal matrix of according dimension and $\sigma^2$ is the variance hyperparameter of the Gaussian likelihood. Finally, we can calculate a posterior distribution of the **contribution** of the $d$-th feature for a given input vector $X_i$:

$$X_i^d f_i^d \sim \mathcal{N}\left(X_i^d \cdot \tilde{m}^d(X_i), (X_i^d)^2 \cdot \tilde{k}^d(X_i, X_i)\right) \tag{13}$$

Now, let us introduce $D$ GPs - $\tilde{f}^1,...,\tilde{f}^D$ - with the following finite dimensional marginal distributions:

$$\tilde{f}_*^d \sim \mathcal{N}\left(X_*^d \odot \tilde{m}_*^d, X_*^d \left(X_*^d\right)^T \odot \tilde{K}_{**}^d\right) \tag{14}$$

with $\tilde{m}_*^d, \tilde{K}_{**}^d$ the mean vector and kernel Gram-matrices per GP as defined in (9). For a given set of inputs and the underlying mean and kernel functions $m^d(\cdot), k^d(\cdot,\cdot)$ fixed, the behavior of the $\tilde{f}^1,..,\tilde{f}^D$ can be manipulated by adjusting $a^d, L^d$, the hyper-parameters of the underlying inducing variables. Clearly, (14) can be interpreted as the attribution corresponding to the respective marginal SVGP.

By summing up the $\tilde{f}^d$, we obtain yet another GP, $\tilde{g}$, with trivial marginal distribution:

$$\tilde{g}_* \sim \mathcal{N}\left(\sum_{d=1}^{D} X_*^d \odot \tilde{m}_*^d, \sum_{d=1}^{D} X_*^d \left(X_*^d\right)^T \odot \tilde{K}_{**}^d\right) \tag{15}$$

Notice that $\tilde{g}$ yields a self-explaining GP whose $d$-th attribution can easily be queried via the corresponding summand GP, $\tilde{f}_*^d$.

Our goal now is to use $\tilde{g}_*$ as a variational posterior distribution for an arbitrary GP $f$ by finding a set of parameters, namely $a^d, L^d$ (and potential hyperparameters for $m^d(\cdot), k^d(\cdot,\cdot)$), for $\tilde{g}_*$ that minimize

$$KL(q_{\tilde{g}}(f)||p(f|y)) \tag{16}$$

with $q_{\tilde{g}}(\cdot)$ the GP distribution as defined in (14). Unfortunately, the usual route for SVGP inference is not possible since $q_{\tilde{g}}(f) = \int p_{\tilde{g}}(f|f_M)q_{\tilde{g}}(f_M)df_M,\ p(f) = \int p(f|f_M)q(f_M)df_M$, hence $p_{\tilde{g}}(f|f_M) \neq p(f|f_M)$ and therefore the conditional distributions do not cancel in the derivation of the ELBO. To solve the resulting infinite dimensional variational problem between the two respective GPs, we apply functional variational inference as proposed by [10]. The authors show, that there exists a functional evidence lower bound (fELBO) which can be maximized in order to solve the optimization problem in (16):

$$fELBO = \mathbb{E}_{q(f)}[p(y_N|f_N)] - \mathbb{E}_{p(A)}\left[KL(q(f_{(N,A)})||p(f_{(N,A)}))\right] \tag{17}$$

where $X_A$ is a so-called measurement set, obtained by sampling uniformly from the space of all possible inputs, $\mathcal{X}$. $X_{(N,A)}$ then denotes the union of $X_N$ and $X_A$ via row-wise stacking, i.e. $X_{(N,A)} = \begin{bmatrix} X_N \\ X_A \end{bmatrix}$. By applying the fELBO to our prior and variational processe, we obtain

$$\begin{gathered} fELBO_1 = \\ \mathbb{E}_{q_{\tilde{g}}(f_N)}[\log p(y_N|f_N)] \\ -\mathbb{E}_{p(A)}\left[KL(\mathcal{N}(m_{\tilde{g},(N,A)}, K_{\tilde{g},(N,A)(N,A)})||\mathcal{N}(m_{(N,A)}, K_{(N,A)(N,A)}))\right] \end{gathered} \tag{18}$$

where $m_{\tilde{g},(N,A)}, K_{\tilde{g},(N,A)(N,A)}$ denote the evaluation of the mean vector and Kernel-gram matrix from (15), evaluated at $X_{(N,A)}$. $m_{(N,A)}, K_{(N,A)(N,A)}$ denote the mean vector and Kernel-gram matrix of the prior GP, evaluated accordingly.

In essence, this approach allows us to encode functional prior knowledge via the prior GP as usual. By decomposing the variational posterior GP after optimizing (18) into its summand attribution GPs, we obtain a transparent approximation of the true posterior distribution in the tradition of varying coefficient or self-explaining models.

Another promising use-case arises, when we place prior distributions on the attribution GPs themselves, e.g. for arbitrary input $X$:

$$f_X^d := X^d f^d \sim \mathcal{GP}(m_X^d(\cdot), k_X^d(\cdot,\cdot)) \tag{19}$$ [5]

[5] $X^d$ can be seen a linear operator on $f^d$ that transforms all finite-dimensional marginals of $f^d$ via $X_N^d \odot f_N^d$.

If the respective mean and kernel functions can be decomposed as $x_i \cdot m^d(x_i)$ and $x_i x_j \cdot k^d(x_i, x_j)$, (19) is a GPX problem as discussed before. If this not the case, however, and if we want to retain transparency of the respective posterior distribution, we can approximate the attribution GPs by $\tilde{f}^1, ..., \tilde{f}^d$. As in (16) we want to minimize

$$KL\left(q_{\tilde{f}^1,...,\tilde{f}^D}\left(f_X^1, ..., f_X^D\right) \middle|\middle| p\left(f_X^1, ..., f_X^D \middle| y\right)\right) \tag{20}$$

By invoking (17) again and by the fact that the KL-divergence of the joint distribution between prior and variational GPs decomposes as the sum of the KL divergences for mutually independent GPs, we get:

$$\begin{gathered} fELBO_2 = \\ \mathbb{E}_{q_{\tilde{f}^1,...,\tilde{f}^D}(f_X^1,...,f_X^D)}\left[\log p\left(y_N \middle| f_X^1, ..., f_X^D\right)\right] \\ -\mathbb{E}_{p(A)}\left[\sum_{d=1}^{D} KL\left(\mathcal{N}\left(m_{\tilde{f}^d,(N,A)}, K_{\tilde{f}^d,(N,A)}\right) \middle|\middle| \mathcal{N}\left(m_{f_X^d,(N,A)}, K_{f_X^d,(N,A)(N,A))}\right)\right)\right] \end{gathered} \tag{21}$$

As a brief example, we could choose $m_X^d(\cdot) << 0$ to exemplify the prior belief that the attribution of the $d$-th feature is negative with high probability. Obviously, potential priors could be much more complex. In fact, it might be fruitful to consider implicit processes as introduced in [18] as a prior and use our self-explaining posterior as an approximation.

## 5. Experiments

In this section, we evaluate the proposed method on several experimental tasks. In particular, we are interested in the explanations generated by our method, its ability to incorporate prior assumption and its predictive performance. All experiments were conducted on regression problems, where the likelihood could be assumed to be Gaussian.

Extended implementation details can be found in **Appendix B**.

### 5.1. Evaluation of explanations

In addition to point values for the varying coefficients, the SVGP components allow to also evaluate the variance of varying coefficients. In accordance with the typical interpretation of posterior variance in Bayesian models, this can be interpreted as a measure of coefficient uncertainty or explanation uncertainty.

To evaluate these measures, the coefficient means and variances of a trained SEVGP model (via (11) ) were calculated for two datapoints from the boston housing dataset. Figures 1. and 2. show the results. While the coefficient means are relatively stable for both examples, the variances differ visibly. Interestingly, the coefficients of the left example show high uncertainty for the most influential coefficient (feature **CHAS**). The respective outputs can be used to check for hidden biases or erroneous reasoning in the respective model.

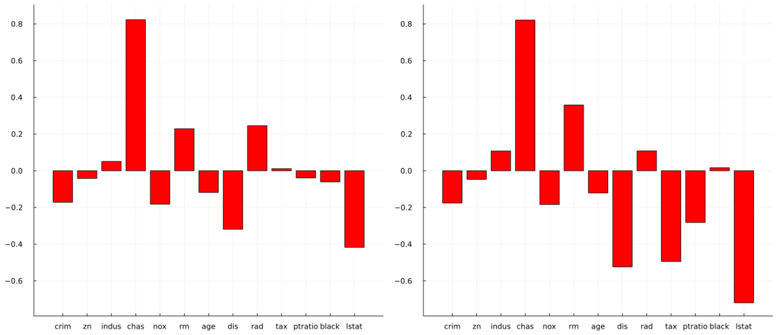

**Figure 1:** *Coefficient means for two input datapoints from the boston housing dataset.*

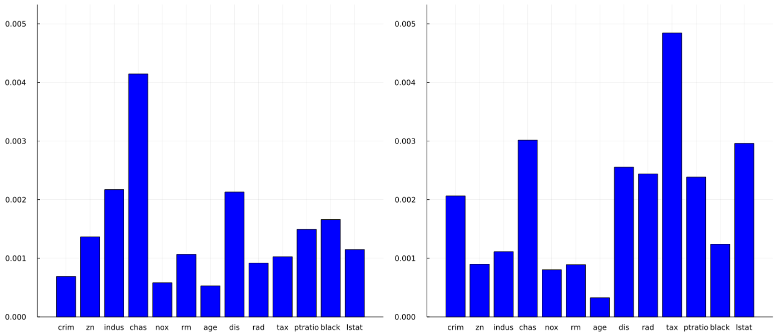

**Figure 2:** *Coefficient var for two input datapoints from the boston housing dataset (corresponding to coefficient means.)*

## 5.2. Evaluation of inclusion of prior knowledge

To verify the model's capability to incorporate existing prior knowledge, a random sample from a quadratic function with gaussian noise was created in the interval $[-2, 2]$. A model that is able to handle knowledge about the underlying quadratic function should be able to extrapolate accordingly beyond the range of the observed data (often termed *out-of-distribution* problem).

In order to validate this claim for our approach, the three models implied in (11), (18) and (21) were compared. For (18) (= prior knowledge about $g$) a GP prior with second-order polynomial kernel was used. For (21) (= prior knowledge about the feature-wise effects) a GP prior with linear kernel was placed on $f^d$, which is technically equivalent to placing a polynomial kernel on $X^d f^d$.

The results in Figure 3 indicate that the model is able to correctly handle the functional prior knowledge about the underlying quadratic function. It can be see, that both models that were trained with additional prior knowledge (middle and right) were able to correctly extrapolate the quadratic function. Without such prior knowledge (left model), the resulting posterior predictive distribution only fits the in-sample data but is unable to extrapolate out of distribution.

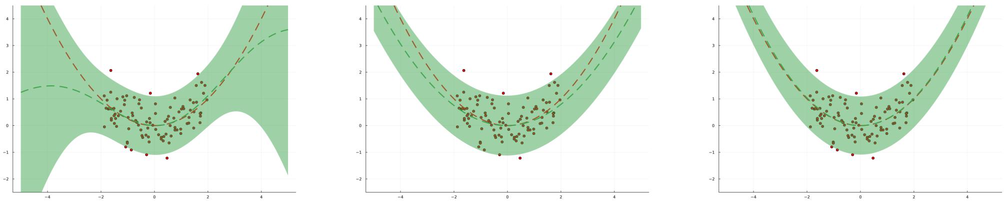

**Figure 3:** *Variational posterior predictive distributions for the approaches in* (11) *(left),* (18) *(middle),* (21) *(right)*

| | **SVGP** | **SEVGP** (this paper) |
|---|---|---|
| Boston | $0.1658 \pm 0.1052$ | $0.1531 \pm 0.0736$ |
| Concrete | $0.0099 \pm 0.0027$ | $0.0106 \pm 0.0048$ |
| Wine red | $0.6212 \pm 0.0410$ | $0.6564 \pm 0.0563$ |
| Wine white | $0.6512 \pm 0.0256$ | $0.7224 \pm 0.0595$ |

**Table 1**
*MSE for SVGP and SEVGP posterior mean; average and standard deviation over 5-fold cross validation*

### 5.3. Evaluation of predictive performance

To validate the predictive performance of the proposed method, it was evaluated over four regression datasets (*boston housing*, *concrete*, *wine red* and *wine white*) via five-fold cross validation. For comparison, standard SVGP was also trained and evaluated on the same folds. Table 1 shows average MSE and MSE standard deviation over the folds. All GP models used an ARD covariance kernel and zero-mean prior functions.

Since SEVGP uses one SVGP per coefficient, the amount of inducing points in the SVGP was increased accordingly to account for the increased model capacity of SEVGP. See **Appendix B** for more details.

It can be seen that our proposed method achieves comparable performance to SVGP. This implies that problems where the latter perform well, allow for the SVGP to be replaced by SEVGP in case the discussed benefits are deemed advantageous.

## 6. Limitations and discussion

This paper presented a method that combines GPs and recent developments in varying-coefficient/self-explaining methods for machine learning. By taking advantage of the Bayesian properties of GPs it is also possible to inject prior knowledge into respective models. One area where both the transparency and the teachability aspects can be helpful is the field of fair and unbiased machine learning. On the one hand, transparency allows to detect biased or discriminating results on a per instance basis. On the other hand, teachability could help prevent or eliminate potential biases by carefully encoding non-biasing prior knowledge into the model. While this would certainly not be a silver bullet, there might nevertheless be considerable, general potential at the intersection of explainable and human-in-the-loop machine learning.

A clear limitation is the fact that the idea of explainability that we considered in this paper is a statistical one, with focus on local, per-pixel explanations. In complex problems like image classification, this might not suffice if a class is inferred from multiple symbolic relations of different objects that are present in a given image instance. Nevertheless, statistical approaches have recently been shown to be quite successful on such complex problems despite possessing no inherent capabilities for logic deduction.

Future work on the proposed method should try to find a way to make the proposed method scalable to other, potentially high dimensional, supervised learning problems. Particularly problems with image inputs, like image classification or reinforcement learning might greatly benefit from external prior knowledge when training data is only sparsely available.

## References

[1] F. Doshi-Velez, B. Kim, Towards a rigorous science of interpretable machine learning, arXiv preprint arXiv:1702.08608 (2017). URL: https://arxiv.org/abs/1702.08608.

[2] D. Alvarez-Melis, T. S. Jaakkola, Towards robust interpretability with self-explaining neural networks, in: S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, 2018, pp. 7786–7795. URL: https://proceedings.neurips.cc/paper/2018/hash/3e9f0fc9b2f89e043bc6233994dfcf76-Abstract.html.

[3] Y. Yoshikawa, T. Iwata, Gaussian process regression with local explanation, CoRR abs/2007.01669 (2020). URL: https://arxiv.org/abs/2007.01669. arXiv:2007.01669.

[4] R. Guhaniyogi, C. Li, T. D. Savitsky, S. Srivastava, Distributed bayesian varying coefficient modeling using a gaussian process prior, arXiv preprint arXiv:2006.00783 (2020). URL: https://arxiv.org/abs/2006.00783.

[5] P. Niyogi, F. Girosi, T. Poggio, Incorporating prior information in machine learning by creating virtual examples, Proceedings of the IEEE 86 (1998) 2196–2209. URL: https://ieeexplore.ieee.org/document/726787#:~:text=DOI%3A-,10.1109/5.726787,-Publisher%3A%20IEEE.

[6] D. Ferranti, D. Krane, D. Craft, The value of prior knowledge in machine learning of complex network systems, Bioinform. 33 (2017) 3610–3618. URL: https://doi.org/10.1093/bioinformatics/btx438. doi:10.1093/bioinformatics/btx438.

[7] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, et al., Informed machine learning–a taxonomy and survey of integrating knowledge into learning systems, arXiv preprint arXiv:1903.12394 (2019). URL: https://arxiv.org/abs/1903.12394.

[8] J. Yang, S. Ren, A quantitative perspective on values of domain knowledge for machine learning, arXiv preprint arXiv:2011.08450 (2020). URL: https://arxiv.org/abs/2011.08450.

[9] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, D. B. Rubin, Bayesian data analysis, CRC press, 2013. URL: https://doi.org/10.1201/b16018.

[10] S. Sun, G. Zhang, J. Shi, R. B. Grosse, Functional variational bayesian neural networks, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, 2019. URL: https://openreview.net/forum?id=rkxacs0qY7.

[11] D. R. Burt, S. W. Ober, A. Garriga-Alonso, M. van der Wilk, Understanding variational inference in function-space, CoRR abs/2011.09421 (2020). URL: https://arxiv.org/abs/2011.09421. arXiv:2011.09421.

[12] G. Montavon, W. Samek, K. Müller, Methods for interpreting and understanding deep neural networks, Digit. Signal Process. 73 (2018) 1–15. URL: https://doi.org/10.1016/j.dsp.2017.10.011. doi:10.1016/j.dsp.2017.10.011.

[13] T. Hastie, R. Tibshirani, Varying-coefficient models, Journal of the Royal Statistical Society: Series B (Methodological) 55 (1993) 757–779. URL: https://www.jstor.org/stable/2345993.

[14] C. E. Rasmussen, Gaussian processes in machine learning, in: O. Bousquet, U. von Luxburg, G. Rätsch (Eds.), Advanced Lectures on Machine Learning, ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures, volume 3176 of *Lecture Notes in Computer Science*, Springer, 2003, pp. 63–71. URL: https://doi.org/10.1007/978-3-540-28650-9_4. doi:10.1007/978-3-540-28650-9_4.

[15] M. K. Titsias, Variational learning of inducing variables in sparse gaussian processes, in: D. A. V. Dyk, M. Welling (Eds.), Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida, USA, April 16-18, 2009, volume 5 of *JMLR Proceedings*, JMLR.org, 2009, pp. 567–574. URL: http://proceedings.mlr.press/v5/titsias09a.html.

[16] J. Hensman, N. Fusi, N. D. Lawrence, Gaussian processes for big data, in: A. Nicholson, P. Smyth (Eds.), Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013, Bellevue, WA, USA, August 11-15, 2013, AUAI Press, 2013. URL: https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=2389&proceeding_id=29.

[17] J. Hensman, A. G. de G. Matthews, Z. Ghahramani, Scalable variational gaussian process classification, in: G. Lebanon, S. V. N. Vishwanathan (Eds.), Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015, volume 38 of *JMLR Workshop and Conference Proceedings*, JMLR.org, 2015. URL: http://proceedings.mlr.press/v38/hensman15.html.

[18] C. Ma, Y. Li, J. M. Hernández-Lobato, Variational implicit processes, in: International Conference on Machine Learning, PMLR, 2019, pp. 4222–4233. URL: https://proceedings.mlr.press/v97/ma19b.html.

## A. Derivation of $ELBO$ (11)

We write $p(f^{1,D}) = p(f^1, ..., f^D)$ and $p(f_M^{1,D}) = p(f_M^1, ..., f_M^D)$. Notice that $p(f^{1,D})$ does technically not exist as it involves the infinite dimensional stochastic processes where densities don't exist. As these objects will cancel out anyway and since such notation is commonly seen in the GP literature, we will keep it here for simplicity. Otherwise, to be notationally exact, we would have to work with KL divergences over probability measures which would make the results much less convenient to derive.

$$KL\left(q\left(f^{1,D}, f_M^{1,D}\right) \middle|\middle| p\left(f^{1,D}, f_M^{1,D} | y_N\right)\right)$$

$$= \int \log \frac{p\left(f_N^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right)}{p\left(f^{1,D}, f_M^{1,D} | y_N\right)} p\left(f^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right) df^{1,D} df_M^{1,D}$$

$$= \int \log \frac{p\left(f_N^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right) p\left(y_N\right)}{p\left(y_N | f^{1,D}, f_M^{1,D}\right) p\left(f_N^{1,D} | f_M^{1,D}\right) p\left(f_M^{1,D}\right)} p\left(f^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right) df^{1,D} df_M^{1,D}$$

$$= \int \log \frac{q\left(f_M^{1,D}\right) p\left(y_N\right)}{p\left(y_N | f^{1,D}, f_M^{1,D}\right) p\left(f_M^{1,D}\right)} p\left(f^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right) df^{1,D} df_M^{1,D}$$

$$\begin{gathered} = \int \log \frac{q\left(f_M^{1,D}\right)}{p\left(f_M^{1,D}\right)} p\left(f^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right) df^{1,D} df_M^{1,D} \\ - \int \log p\left(y_N | f^{1,D}, f_M^{1,D}\right) p\left(f^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right) df^{1,D} df_M^{1,D} \\ + \int \log p(y) p\left(f^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right) df^{1,D} df_M^{1,D} \end{gathered}$$

$$\begin{gathered} = KL\left(q\left(f_M^{1,D}\right) \middle|\middle| p\left(f_M^{1,D}\right)\right) \\ -\mathbb{E}_{p\left(f^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right)}\left[\log p\left(y_N | f^{1,D}, f_M^{1,D}\right)\right] \\ +p(y) \end{gathered}$$

$$\begin{gathered} = KL\left(q\left(f_M^{1,D}\right) \middle|\middle| p\left(f_M^{1,D}\right)\right) \\ -\mathbb{E}_{p\left(f^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right)}\left[\log p\left(y_N | f_N^{1,D}\right)\right] \\ +p(y) \end{gathered}$$

since $y_N$ depends on $f_M^{1,D}$ only via $f^{1,D}$ and only on the marginals given by $X_N$.

$$\begin{gathered} = KL\left(q\left(f_M^{1,D}\right) \middle|\middle| p\left(f_M^{1,D}\right)\right) \\ -\mathbb{E}_{q\left(\tilde{f}^{1,D}\right)}\left[\log p\left(y_N | \tilde{f}_N^{1,D}\right)\right] \\ +p(y) \end{gathered}$$

by marginalizing out $f_M^{1,D}$ and writing $\tilde{f}^{1,D}$ for clarity as explained before.

$$
\begin{aligned}
&= \sum_{d=1}^{D} KL(\mathcal{N}(a^d, S^d)||\mathcal{N}(m_M^d, K_{MM}^d)) \\
&\quad -\mathbb{E}_{q(\tilde{f}^{1,D})}\left[\log p\left(y_i|\tilde{f}_i^{1,D}\right)\right] \\
&\quad +p(y)
\end{aligned}
$$

by independence of prior and variational GPs and by standard i.i.d. assumption about observed datapoints

$$
\Rightarrow p(y) \geq \mathbb{E}_{q(\tilde{f}^{1,D})}\left[\log p\left(y_i|\tilde{f}_i^{1,D}\right)\right] - \sum_{d=1}^{D} KL(\mathcal{N}(a^d, S^d)||\mathcal{N}(m_M^d, K_{MM}^d))
$$

$$
ELBO = \mathbb{E}_{q(\tilde{f}^{1,D})}\left[\log p\left(y_i|\tilde{f}_i^{1,D}\right)\right] - \sum_{d=1}^{D} KL(\mathcal{N}(a^d, S^d)||\mathcal{N}(m_M^d, K_{MM}^d))
$$

## B. Implementation details

We write $p(f^{1,D}) = p(f^1, ..., f^D)$ and $p(f_M^{1,D}) = p(f_M^1, ..., f_M^D)$. Notice that $p(f^{1,D})$ does technically not exist as it involves the infinite dimensional stochastic processes where densities don't exist. As these objects will cancel out anyway and since such notation is commonly seen in the GP literature, we will keep it here for simplicity. Otherwise, to be notationally exact, we would have to work with KL divergences over probability measures which would make the results much less convenient to derive.

$$
KL\left(q\left(f^{1,D}, f_M^{1,D}\right)\Big|\Big|p\left(f^{1,D}, f_M^{1,D}|y_N\right)\right)
$$

$$
= \int \log \frac{p\left(f_N^{1,D}|f_M^{1,D}\right)q\left(f_M^{1,D}\right)}{p\left(f^{1,D}, f_M^{1,D}|y_N\right)} p\left(f^{1,D}|f_M^{1,D}\right)q\left(f_M^{1,D}\right)df^{1,D}df_M^{1,D}
$$

$$
= \int \log \frac{p\left(f_N^{1,D}|f_M^{1,D}\right)q\left(f_M^{1,D}\right)p\left(y_N\right)}{p\left(y_N|f^{1,D}, f_M^{1,D}\right)p\left(f_N^{1,D}|f_M^{1,D}\right)p\left(f_M^{1,D}\right)} p\left(f^{1,D}|f_M^{1,D}\right)q\left(f_M^{1,D}\right)df^{1,D}df_M^{1,D}
$$

$$
= \int \log \frac{q\left(f_M^{1,D}\right)p\left(y_N\right)}{p\left(y_N|f^{1,D}, f_M^{1,D}\right)p\left(f_M^{1,D}\right)} p\left(f^{1,D}|f_M^{1,D}\right)q\left(f_M^{1,D}\right)df^{1,D}df_M^{1,D}
$$

$$
\begin{aligned}
&= \int \log \frac{q\left(f_M^{1,D}\right)}{p\left(f_M^{1,D}\right)} p\left(f^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right) df^{1,D} df_M^{1,D} \\
&- \int \log p\left(y_N | f^{1,D}, f_M^{1,D}\right) p\left(f^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right) df^{1,D} df_M^{1,D} \\
&+ \int \log p(y) p\left(f^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right) df^{1,D} df_M^{1,D}
\end{aligned}
$$

$$
\begin{aligned}
&= KL\left(q\left(f_M^{1,D}\right) \big|\big| p\left(f_M^{1,D}\right)\right) \\
&-\mathbb{E}_{p\left(f^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right)}\left[\log p\left(y_N | f^{1,D}, f_M^{1,D}\right)\right] \\
&+p(y)
\end{aligned}
$$

$$
\begin{aligned}
&= KL\left(q\left(f_M^{1,D}\right) \big|\big| p\left(f_M^{1,D}\right)\right) \\
&-\mathbb{E}_{p\left(f^{1,D} | f_M^{1,D}\right) q\left(f_M^{1,D}\right)}\left[\log p\left(y_N | f_N^{1,D}\right)\right] \\
&+p(y)
\end{aligned}
$$

since $y_N$ depends on $f_M^{1,D}$ only via $f^{1,D}$ and only on the marginals given by $X_N$.

$$
\begin{aligned}
&= KL\left(q\left(f_M^{1,D}\right) \big|\big| p\left(f_M^{1,D}\right)\right) \\
&-\mathbb{E}_{q\left(\tilde{f}^{1,D}\right)}\left[\log p\left(y_N | \tilde{f}_N^{1,D}\right)\right] \\
&+p(y)
\end{aligned}
$$

by marginalizing out $f_M^{1,D}$ and writing $\tilde{f}^{1,D}$ for clarity as explained before.

$$
\begin{aligned}
&= \sum_{d=1}^{D} KL(\mathcal{N}(a^d, S^d) || \mathcal{N}(m_M^d, K_{MM}^d)) \\
&-\mathbb{E}_{q(\tilde{f}^{1,D})}\left[\log p\left(y_i | \tilde{f}_i^{1,D}\right)\right] \\
&+p(y)
\end{aligned}
$$

by independence of prior and variational GPs and by standard i.i.d. assumption about observed datapoints

$$\Rightarrow p(y) \geq \mathbb{E}_{q(\tilde{f}^{1,D})}\left[\log p\left(y_i \middle| \tilde{f}_i^{1,D}\right)\right] - \sum_{d=1}^{D} KL(\mathcal{N}(a^d, S^d)||\mathcal{N}(m_M^d, K_{MM}^d))$$

$$ELBO = \mathbb{E}_{q(\tilde{f}^{1,D})}\left[\log p\left(y_i \middle| \tilde{f}_i^{1,D}\right)\right] - \sum_{d=1}^{D} KL(\mathcal{N}(a^d, S^d)||\mathcal{N}(m_M^d, K_{MM}^d))$$