

Data Association with Gaussian Processes

Markus Kaiser^{1,2*} ✉, Clemens Otte¹,
Thomas A. Runkler^{1,2}, and Carl Henrik Ek³

¹ Siemens AG, markus.kaiser@siemens.com

² Technical University of Munich

³ University of Bristol

Abstract. The data association problem is concerned with separating data coming from different generating processes, for example when data comes from different data sources, contain significant noise, or exhibit multimodality. We present a fully Bayesian approach to this problem. Our model is capable of simultaneously solving the data association problem and the induced supervised learning problem. Underpinning our approach is the use of Gaussian process priors to encode the structure of both the data and the data associations. We present an efficient learning scheme based on doubly stochastic variational inference and discuss how it can be applied to deep Gaussian process priors.

1 Introduction

Real-world data often include multiple operational regimes of the considered system, for example a wind turbine or gas turbine [12]. As an example, consider a model describing the lift resulting from airflow around the wing profile of an airplane as a function of the attack angle. At low values the lift increases linearly with the attack angle until the wing stalls and the characteristic of the airflow changes fundamentally. Building a truthful model of such data requires learning two separate models and correctly associating the observed data to each of the dynamical regimes. A similar example would be if our sensors that measure the lift are faulty in a manner such that we either get an accurate reading or a noisy one. Estimating a model in this scenario is often referred to as a *data association problem* [2, 8], where we consider the data to have been generated by a mixture of processes and we are interested in factorising the data into these components.

Figure 1 shows an example of faulty sensor data, where sensor readings are disturbed by uncorrelated and asymmetric noise. Applying standard machine learning approaches to such data can lead to model pollution, where the expressive power of the model is used to explain noise instead of the underlying signal. Solving the data association problem by factorizing the data into signal and noise gives rise to a principled approach to avoid this behavior.

Early approaches to explaining data using multiple generative processes are based on separating the input space and training local expert models explaining

* The project this report is based on was supported with funds from the German Federal Ministry of Education and Research under project number 01 IS 18049 A.

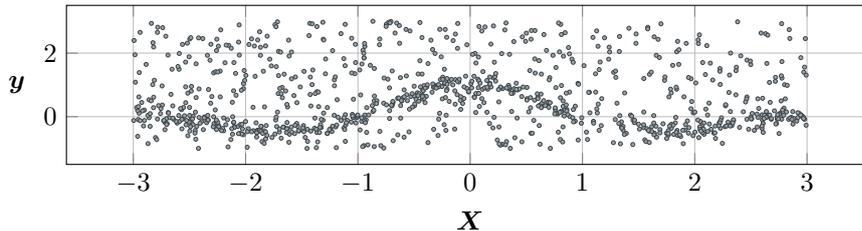


Fig. 1. A data association problem consisting of two generating processes, one of which is a signal we wish to recover and one is an uncorrelated noise process.

easier subtasks [15, 21, 25]. The assignment of data points to local experts is handled by a gating network, which learns a function from the inputs to assignment probabilities. However, it is still a central assumption of these models that at every position in the input space exactly one expert should explain the data. Another approach is presented in [4], where the multimodal regression tasks are interpreted as a density estimation problem. A high number of candidate distributions is reweighed to match the observed data without modeling the underlying generative process.

In contrast, we are interested in a generative process, where data at the same location in the input space could have been generated by a number of global independent processes. Inherently, the data association problem is ill-posed and requires assumptions on both the underlying functions and the association of the observations. In [18] the authors place Gaussian process (GP) priors on the different generative processes which are assumed to be relevant globally. The associations are modelled via a latent association matrix and inference is carried out using an expectation maximization algorithm. This approach takes both the inputs and the outputs of the training data into account to solve the association problem. A drawback is that the model cannot give a posterior estimate about the relevance of the different generating processes at different locations in the input space. This means that the model can be used for data exploration but additional information is needed in order to perform predictive tasks. Another approach in [5] expands this model by allowing interdependencies between the different generative processes and formulating the association problem as an inference problem on a latent space and a corresponding covariance function. However, in this approach the number of components is a free parameter and is prone to overfitting, as the model has no means of turning off components.

In this paper, we formulate a Bayesian model for the data association problem. Underpinning our approach is the use of GP priors which encode structure both on the functions and the associations themselves, allowing us to incorporate the available prior knowledge about the proper factorization into the learning problem. The use of GP priors allows us to achieve principled regularization without reducing the solution space leading to a well-regularized learning problem. Importantly, we simultaneously solve the association problem for the training

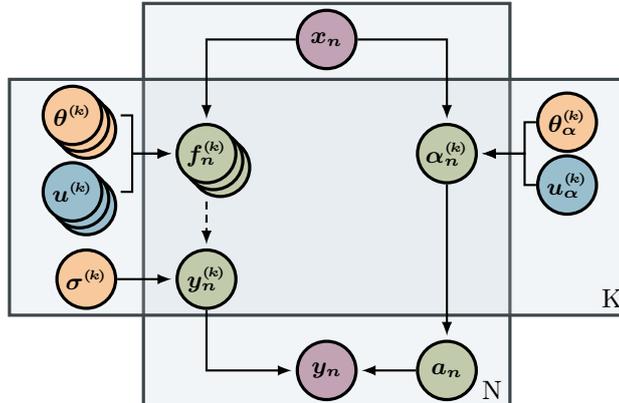


Fig. 2. The graphical model of DAGP. The violet observations $(\mathbf{x}_n, \mathbf{y}_n)$ are generated by the latent process (green). Exactly one of the K latent functions $f_n^{(k)}$ and likelihood $y_n^{(k)}$ are evaluated to generate \mathbf{y}_n . We can place shallow or deep GP priors on these latent function values $f_n^{(k)}$. The assignment \mathbf{a}_n to a latent function is driven by input-dependent weights $\alpha_n^{(k)}$ which encode the relevance of the different functions at \mathbf{x}_n . The different parts of the model are determined by the hyperparameters θ, σ (yellow) and variational parameters \mathbf{u} (blue).

data taking both inputs and outputs into account while also obtaining posterior belief about the relevance of the different generating processes in the input space. Our model can describe non-stationary processes in the sense that a different number of processes can be activated in different locations in the input space. We describe this non-stationary structure using additional GP priors which allows us to make full use of problem specific knowledge. This leads to a flexible yet interpretable model with a principled treatment of uncertainty.

The paper has the following contributions: In Section 2, we propose the data association with Gaussian processes model (DAGP). In Section 3, we present an efficient learning scheme via a variational approximation which allows us to simultaneously train all parts of our model via stochastic optimization and show how the same learning scheme can be applied to deep GP priors. We demonstrate our model on a noise separation problem, an artificial multimodal data set, and a multi-regime regression problem based on the cart-pole benchmark in Section 4.

2 Data Association with Gaussian Processes

The data association with Gaussian processes (DAGP) model assumes that there exist K independent functions $\{f^{(k)}\}_{k=1}^K$, which generate pairs of observations $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$. Each data point is generated by evaluating one of the K latent functions and adding Gaussian noise from a corresponding likelihood. The assignment of the n^{th} data point to one of the functions is specified by the indicator vector $\mathbf{a}_n \in \{0, 1\}^K$, which has exactly one non-zero entry. Our goal

is to formulate simultaneous Bayesian inference on the functions $f^{(k)}$ and the assignments \mathbf{a}_n .

For notational conciseness, we follow the GP related notation in [14] and collect all N inputs as $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ and all outputs as $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_N)$. We further denote the k^{th} latent function value associated with the n^{th} data point as $\mathbf{f}_n^{(k)} = f^{(k)}(\mathbf{x}_n)$ and collect them as $\mathbf{F}^{(k)} = (\mathbf{f}_1^{(k)}, \dots, \mathbf{f}_N^{(k)})$ and $\mathbf{F} = (\mathbf{F}^{(1)}, \dots, \mathbf{F}^{(K)})$. We refer to the k^{th} entry in \mathbf{a}_n as $a_n^{(k)}$ and denote $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_N)$.

Given this notation, the marginal likelihood of DAGP can be separated into the likelihood, the latent function processes, and the assignment process and is given by,

$$\begin{aligned} \text{p}(\mathbf{Y} | \mathbf{X}) &= \int \text{p}(\mathbf{Y} | \mathbf{F}, \mathbf{A}) \text{p}(\mathbf{F} | \mathbf{X}) \text{p}(\mathbf{A} | \mathbf{X}) \text{d}\mathbf{A} \text{d}\mathbf{F} \\ \text{p}(\mathbf{Y} | \mathbf{F}, \mathbf{A}) &= \prod_{n=1}^N \prod_{k=1}^K \mathcal{N}(\mathbf{y}_n | \mathbf{f}_n^{(k)}, (\sigma^{(k)})^2)^{\mathbb{I}(a_n^{(k)}=1)} \end{aligned} \quad (1)$$

where $\sigma^{(k)}$ is the noise of the k^{th} Gaussian likelihood and \mathbb{I} the indicator function.

Since we assume the K processes to be independent given the data and assignments, we place independent GP priors on the latent functions $\text{p}(\mathbf{F} | \mathbf{X}) = \prod_{k=1}^K \mathcal{N}(\mathbf{F}^{(k)} | \mu^{(k)}(\mathbf{X}), \mathcal{K}^{(k)}(\mathbf{X}, \mathbf{X}))$ with mean function $\mu^{(k)}$ and kernel $\mathcal{K}^{(k)}$. Our prior on the assignment process is composite. First, we assume that the \mathbf{a}_n are drawn independently from multinomial distributions with logit parameters $\boldsymbol{\alpha}_n = (\alpha_n^{(1)}, \dots, \alpha_n^{(K)})$. One approach to specify $\boldsymbol{\alpha}_n$ is to assume them to be known a priori and to be equal for all data points [18]. Instead, we want to infer them from the data. Specifically, we assume that there is a relationship between the location in the input space \mathbf{x} and the associations. By placing independent GP priors on $\boldsymbol{\alpha}^{(k)}$, we can encode our prior knowledge of the associations by the choice of covariance function $\text{p}(\boldsymbol{\alpha} | \mathbf{X}) = \prod_{k=1}^K \mathcal{N}(\boldsymbol{\alpha}^{(k)} | \mathbf{0}, \mathcal{K}_{\boldsymbol{\alpha}^{(k)}}(\mathbf{X}, \mathbf{X}))$. The prior on the assignments \mathbf{A} is given by marginalizing the $\boldsymbol{\alpha}^{(k)}$, which, when normalized, parametrize a batch of multinomial distributions,

$$\text{p}(\mathbf{A} | \mathbf{X}) = \int \mathcal{M}(\mathbf{A} | \text{softmax}(\boldsymbol{\alpha})) \text{p}(\boldsymbol{\alpha} | \mathbf{X}) \text{d}\boldsymbol{\alpha}. \quad (2)$$

Modelling the relationship between the input and the associations allows us to efficiently model data, which, for example, is unimodal in some parts of the input space and bimodal in others. A simple smoothness prior will encode a belief for how quickly the components switch across the input domain.

Since the GPs of the $\boldsymbol{\alpha}^{(k)}$ use a zero mean function, our prior assumption is a uniform distribution of the different generative processes everywhere in the input space. If inference on the \mathbf{a}_n reveals that, say, all data points at similar positions in the input space can be explained by the same k^{th} process, the belief about $\boldsymbol{\alpha}$ can be adjusted to make a non-uniform distribution favorable at this position, thereby increasing the likelihood via $\text{p}(\mathbf{A} | \mathbf{X})$. This mechanism introduces an incentive for the model to use as few functions as possible to explain the data and importantly allows us to predict a relative importance of these functions when calculating the posterior of the new observations \mathbf{x}_* .

Figure 2 shows the resulting graphical model, which divides the generative process for every data point in the application of the latent functions on the left side and the assignment process on the right side. The interdependencies between the data points are introduced through the GP priors on $\mathbf{f}_n^{(k)}$ and $\alpha_n^{(k)}$ and depend on the hyperparameters $\theta = \{\theta^{(k)}, \theta_\alpha^{(k)}, \sigma^{(k)}\}_{k=1}^K$.

The priors for the $f^{(k)}$ can be chosen independently to encode different prior assumptions about the underlying processes. In Section 4.1, we use different kernels to separate a non-linear signal from a noise process. Going further, we can also use deep GP as priors for the $f^{(k)}$ [9, 23]. Since many real word systems are inherently hierarchical, prior knowledge can often be formulated more easily using composite functions [16].

3 Variational Approximation

Exact inference is intractable in this model. Instead, we formulate a variational approximation following ideas from [13, 23]. Because of the rich structure in our model, finding a variational lower bound which is both faithful and can be evaluated analytically is hard. To proceed, we formulate an approximation which factorizes along both the K processes and N data points. This bound can be sampled efficiently and allows us to optimize both the models for the different processes $\{f^{(k)}\}_{k=1}^K$ and our belief about the data assignments $\{\mathbf{a}_n\}_{n=1}^N$ simultaneously using stochastic optimization.

3.1 Variational Lower Bound

As first introduced by Titsias [24], we augment all GP in our model using sets of M inducing points $\mathbf{Z}^{(k)} = (\mathbf{z}_1^{(k)}, \dots, \mathbf{z}_M^{(k)})$ and their corresponding function values $\mathbf{u}^{(k)} = f^{(k)}(\mathbf{Z}^{(k)})$, the inducing variables. We collect them as $\mathbf{Z} = \{\mathbf{Z}^{(k)}, \mathbf{Z}_\alpha^{(k)}\}_{k=1}^K$ and $\mathbf{U} = \{\mathbf{u}^{(k)}, \mathbf{u}_\alpha^{(k)}\}_{k=1}^K$. Taking the function $f^{(k)}$ and its corresponding GP as an example, the inducing variables $\mathbf{u}^{(k)}$ are jointly Gaussian with the latent function values $\mathbf{F}^{(k)}$ of the observed data by the definition of GPs. We follow [13] and choose the variational approximation $q(\mathbf{F}^{(k)}, \mathbf{u}^{(k)}) = p(\mathbf{F}^{(k)} | \mathbf{u}^{(k)}, \mathbf{X}, \mathbf{Z}^{(k)}) q(\mathbf{u}^{(k)})$ with $q(\mathbf{u}^{(k)}) = \mathcal{N}(\mathbf{u}^{(k)} | \mathbf{m}^{(k)}, \mathbf{S}^{(k)})$. This formulation introduces the set $\{\mathbf{Z}^{(k)}, \mathbf{m}^{(k)}, \mathbf{S}^{(k)}\}$ of variational parameters indicated in Figure 2. To simplify notation we drop the dependency on \mathbf{Z} in the following.

A central assumption of this approximation is that given enough well-placed inducing variables $\mathbf{u}^{(k)}$, they are a sufficient statistic for the latent function values $\mathbf{F}^{(k)}$. This implies conditional independence of the $\mathbf{f}_n^{(k)}$ given $\mathbf{u}^{(k)}$ and \mathbf{X} . The variational posterior of a single GP can then be written as,

$$\begin{aligned} q(\mathbf{F}^{(k)} | \mathbf{X}) &= \int q(\mathbf{u}^{(k)}) p(\mathbf{F}^{(k)} | \mathbf{u}^{(k)}, \mathbf{X}) d\mathbf{u}^{(k)} \\ &= \int q(\mathbf{u}^{(k)}) \prod_{n=1}^N p(\mathbf{f}_n^{(k)} | \mathbf{u}^{(k)}, \mathbf{x}_n) d\mathbf{u}^{(k)}, \end{aligned} \tag{3}$$

which can be evaluated analytically, since it is a convolution of Gaussians. This formulation simplifies inference within single GPs. Next, we discuss how to handle the correlations between the different functions and the assignment processes.

Given a set of assignments \mathbf{A} , this factorization along the data points is preserved in our model due to the assumed independence of the different functions in (1). The independence is lost if the assignments are unknown. In this case, both the (a priori independent) assignment processes and the functions influence each other through data with unclear assignments. Following the ideas of doubly stochastic variational inference (DSVI) presented by Salimbeni and Deisenroth [23] in the context of deep GPs, we maintain these correlations between different parts of the model while assuming factorization of the variational distribution. That is, our variational posterior takes the factorized form,

$$\begin{aligned} q(\mathbf{F}, \boldsymbol{\alpha}, \mathbf{U}) &= q\left(\boldsymbol{\alpha}, \{\mathbf{F}^{(k)}, \mathbf{u}^{(k)}, \mathbf{u}_{\boldsymbol{\alpha}}^{(k)}\}_{k=1}^K\right) \\ &= \prod_{k=1}^K \prod_{n=1}^N p(\boldsymbol{\alpha}_n^{(k)} | \mathbf{u}_{\boldsymbol{\alpha}}^{(k)}, \mathbf{x}_n) q(\mathbf{u}_{\boldsymbol{\alpha}}^{(k)}) \prod_{k=1}^K \prod_{n=1}^N p(\mathbf{f}_n^{(k)} | \mathbf{u}^{(k)}, \mathbf{x}_n) q(\mathbf{u}^{(k)}). \end{aligned} \quad (4)$$

Our goal is to recover a posterior for both the generating functions and the assignment of data. To achieve this, instead of marginalizing \mathbf{A} , we consider the variational joint of \mathbf{Y} and \mathbf{A} ,

$$q(\mathbf{Y}, \mathbf{A}) = \int p(\mathbf{Y} | \mathbf{F}, \mathbf{A}) p(\mathbf{A} | \boldsymbol{\alpha}) q(\mathbf{F}, \boldsymbol{\alpha}) d\mathbf{F} d\boldsymbol{\alpha}, \quad (5)$$

which retains both the Gaussian likelihood of \mathbf{Y} and the multinomial likelihood of \mathbf{A} in (2). A lower bound $\mathcal{L}_{\text{DAGP}}$ for the log-joint $\log p(\mathbf{Y}, \mathbf{A} | \mathbf{X})$ of DAGP is given by,

$$\begin{aligned} \mathcal{L}_{\text{DAGP}} &= \mathbb{E}_{q(\mathbf{F}, \boldsymbol{\alpha}, \mathbf{U})} \left[\log \frac{p(\mathbf{Y}, \mathbf{A}, \mathbf{F}, \boldsymbol{\alpha}, \mathbf{U} | \mathbf{X})}{q(\mathbf{F}, \boldsymbol{\alpha}, \mathbf{U})} \right] \\ &= \sum_{n=1}^N \mathbb{E}_{q(\mathbf{f}_n)} [\log p(\mathbf{y}_n | \mathbf{f}_n, \mathbf{a}_n)] + \sum_{n=1}^N \mathbb{E}_{q(\boldsymbol{\alpha}_n)} [\log p(\mathbf{a}_n | \boldsymbol{\alpha}_n)] \\ &\quad - \sum_{k=1}^K \text{KL}(q(\mathbf{u}^{(k)}) \| p(\mathbf{u}^{(k)} | \mathbf{Z}^{(k)})) - \sum_{k=1}^K \text{KL}(q(\mathbf{u}_{\boldsymbol{\alpha}}^{(k)}) \| p(\mathbf{u}_{\boldsymbol{\alpha}}^{(k)} | \mathbf{Z}_{\boldsymbol{\alpha}}^{(k)})). \end{aligned} \quad (6)$$

Due to the structure of (4), the bound factorizes along the data enabling stochastic optimization. This bound has complexity $\mathcal{O}(NM^2K)$ to evaluate.

3.2 Optimization of the Lower Bound

An important property of the variational bound for DSVI [23] is that taking samples for single data points is straightforward and can be implemented efficiently. Specifically, for some k and n , samples $\hat{\mathbf{f}}_n^{(k)}$ from $q(\mathbf{f}_n^{(k)})$ are independent of all other parts of the model and can be drawn using samples from univariate unit Gaussians using reparametrizations [17, 22].

Note that it would not be necessary to sample from the different processes, since $q(\mathbf{F}^{(k)})$ can be computed analytically [13]. However, we apply the sampling scheme to the optimization of both the assignment processes $\boldsymbol{\alpha}$ and the assignments \mathbf{A} as for $\boldsymbol{\alpha}$, the analytical propagation of uncertainties through the softmax renormalization and multinomial likelihoods is intractable but can easily be evaluated using sampling.

We optimize $\mathcal{L}_{\text{DAGP}}$ to simultaneously recover maximum likelihood estimates of the hyperparameters $\boldsymbol{\theta}$, the variational parameters $\{\mathbf{Z}, \mathbf{m}, \mathbf{S}\}$, and assignments \mathbf{A} . For every n , we represent the belief about \mathbf{a}_n as a K -dimensional discrete distribution $q(\mathbf{a}_n)$. This distribution models the result of drawing a sample from $\mathcal{M}(\mathbf{a}_n | \text{softmax}(\boldsymbol{\alpha}_n))$ during the generation of the data point $(\mathbf{x}_n, \mathbf{y}_n)$.

Since we want to optimize $\mathcal{L}_{\text{DAGP}}$ using (stochastic) gradient descent, we need to employ a continuous relaxation to gain informative gradients of the bound with respect to the binary (and discrete) vectors \mathbf{a}_n . One straightforward way to relax the problem is to use the current belief about $q(\mathbf{a}_n)$ as parameters for a convex combination of the $\mathbf{f}_n^{(k)}$, that is, to approximate $\mathbf{f}_n \approx \sum_{k=1}^K q(\mathbf{a}_n^{(k)}) \hat{\mathbf{f}}_n^{(k)}$. Using this relaxation is problematic in practice. Explaining data points as mixtures of the different generating processes violates the modelling assumption that every data point was generated using exactly one function but can substantially simplify the learning problem. Because of this, special care must be taken during optimization to enforce the sparsity of $q(\mathbf{a}_n)$.

To avoid this problem, we propose using a different relaxation based on additional stochasticity. Instead of directly using $q(\mathbf{a}_n)$ to combine the $\mathbf{f}_n^{(k)}$, we first draw a sample $\hat{\mathbf{a}}_n$ from a concrete random variable as suggested by Maddison et al. [19], parameterized by $q(\mathbf{a}_n)$. Based on a temperature parameter λ , a concrete random variable enforces sparsity but is also continuous and yields informative gradients using automatic differentiation. Samples from a concrete random variable are unit vectors and for $\lambda \rightarrow 0$ their distribution approaches a discrete distribution.

Our approximate evaluation of the bound in (6) during optimization has multiple sources of stochasticity, all of which are unbiased. First, we approximate the expectations using Monte Carlo samples $\hat{\mathbf{f}}_n^{(k)}$, $\hat{\boldsymbol{\alpha}}_n^{(k)}$, and $\hat{\mathbf{a}}_n$. And second, the factorization of the bound along the data allows us to use mini-batches for optimization [13, 23].

3.3 Approximate Predictions

Predictions for a test location \mathbf{x}_* are mixtures of K independent Gaussians, given by,

$$q(\mathbf{f}_* | \mathbf{x}_*) = \int \sum_{k=1}^K q(a_*^{(k)} | \mathbf{x}_*) q(\mathbf{f}_*^{(k)} | \mathbf{x}_*) d\mathbf{a}_*^{(k)} \approx \sum_{k=1}^K \hat{a}_*^{(k)} \hat{\mathbf{f}}_*^{(k)}. \quad (7)$$

The predictive posteriors of the K functions $q(\mathbf{f}_*^{(k)} | \mathbf{x}_*)$ are given by K independent shallow GPs and can be calculated analytically [13]. Samples from

the predictive density over $q(\mathbf{a}_* | \mathbf{x}_*)$ can be obtained by sampling from the GP posteriors $q(\boldsymbol{\alpha}_*^{(k)} | \mathbf{x}_*)$ and renormalizing the resulting vector $\boldsymbol{\alpha}_*$ using the softmax-function. The distribution $q(\mathbf{a}_* | \mathbf{x}_*)$ reflects the model’s belief about how many and which of the K generative processes are relevant at the test location \mathbf{x}_* and their relative probability.

3.4 Deep Gaussian Processes

For clarity, we have described the variational bound in terms of a shallow GP. However, as long as their variational bound can be efficiently sampled, any model can be used in place of shallow GPs for the $f^{(k)}$. Since our approximation is based on DSVI, an extension to deep GPs is straightforward. Analogously to [23], our new prior assumption about the k^{th} latent function values $p(\mathbf{F}'^{(k)} | \mathbf{X})$ is given by,

$$p(\mathbf{F}'^{(k)} | \mathbf{X}) = \prod_{l=1}^L p\left(\mathbf{F}'_l^{(k)} \mid \mathbf{u}'_l^{(k)} \mathbf{F}'_{l-1}^{(k)}, \mathbf{Z}'_l^{(k)}\right), \quad (8)$$

for an L -layer deep GP and with $\mathbf{F}'_0^{(k)} := \mathbf{X}$. Similar to the single-layer case, we introduce sets of inducing points $\mathbf{Z}'_l^{(k)}$ and a variational distribution over their corresponding function values $q(\mathbf{u}'_l^{(k)}) = \mathcal{N}(\mathbf{u}'_l^{(k)} | \mathbf{m}'_l^{(k)}, \mathbf{S}'_l^{(k)})$. We collect the latent multi-layer function values as $\mathbf{F}' = \{\mathbf{F}'_l^{(k)}\}_{k=1, l=1}^{K, L}$ and corresponding \mathbf{U}' and assume an extended variational distribution,

$$\begin{aligned} q(\mathbf{F}', \boldsymbol{\alpha}, \mathbf{U}') &= q\left(\boldsymbol{\alpha}, \{\mathbf{u}'_{\boldsymbol{\alpha}}\}_{k=1}^K, \{\mathbf{F}'_l^{(k)}, \mathbf{u}'_l^{(k)}\}_{k=1, l=1}^{K, L}\right) \\ &= \prod_{k=1}^K \prod_{n=1}^N p(\boldsymbol{\alpha}_n^{(k)} | \mathbf{u}'_{\boldsymbol{\alpha}}^{(k)}, \mathbf{x}_n) q(\mathbf{u}'_{\boldsymbol{\alpha}}^{(k)}) \prod_{k=1}^K \prod_{l=1}^L \prod_{n=1}^N p\left(\mathbf{f}'_{n, l}^{(k)} \mid \mathbf{u}'_l^{(k)}, \mathbf{x}_n\right) q(\mathbf{u}'_l^{(k)}), \end{aligned} \quad (9)$$

where we identify $\mathbf{f}'_n^{(k)} = \mathbf{f}'_{n, L}^{(k)}$. As the n^{th} marginal of the L^{th} layer depends only on the n^{th} marginal of all layers above sampling from them remains straightforward [23]. The marginal is given by,

$$q(\mathbf{f}'_{n, L}^{(k)}) = \int q(\mathbf{f}'_{n, L}^{(k)} | \mathbf{f}'_{n, L-1}^{(k)}) \prod_{l=1}^{L-1} q(\mathbf{f}'_{n, l}^{(k)} | \mathbf{f}'_{n, l-1}^{(k)}) d\mathbf{f}'_{n, l}^{(k)}. \quad (10)$$

The complete bound is structurally similar to (6) and given by,

$$\begin{aligned} \mathcal{L}'_{\text{DAGP}} &= \sum_{n=1}^N \mathbb{E}_{q(\mathbf{f}'_n)} [\log p(\mathbf{y}_n | \mathbf{f}'_n, \mathbf{a}_n)] + \sum_{n=1}^N \mathbb{E}_{q(\boldsymbol{\alpha}_n)} [\log p(\mathbf{a}_n | \boldsymbol{\alpha}_n)] \\ &\quad - \sum_{k=1}^K \sum_{l=1}^L \text{KL}(q(\mathbf{u}'_l^{(k)}) \parallel p(\mathbf{u}'_l^{(k)} | \mathbf{Z}'_l^{(k)})) - \sum_{k=1}^K \text{KL}(q(\mathbf{u}'_{\boldsymbol{\alpha}}^{(k)}) \parallel p(\mathbf{u}'_{\boldsymbol{\alpha}}^{(k)} | \mathbf{Z}'_{\boldsymbol{\alpha}}^{(k)})). \end{aligned} \quad (11)$$

To calculate the first term, samples have to be propagated through the deep GP structures. This extended bound thus has complexity $\mathcal{O}(NM^2LK)$ to evaluate in the general case and complexity $\mathcal{O}(NM^2 \cdot \max(L, K))$ if the assignments \mathbf{a}_n take binary values.

Table 1. Comparison of qualitative model capabilities. A model has a capability if it contains components which enable it to solve the respective task in principle.

Experiment	Predictive Posterior	Multimodal Data	Scalable Inference	Interpretable Priors	Data Association	Predictive Associations	Separate Models
					Table 2	Table 3	Figure 4
DAGP (Ours)	✓	✓	✓	✓	✓	✓	✓
OMGP [18]	✓	✓	–	✓	✓	–	✓
RGPR [21]	✓	✓	–	✓	–	–	–
GPR	✓	–	✓	✓	–	–	–
BNN+LV [10]	✓	✓	✓	–	–	–	–
MDN [4]	✓	✓	✓	–	–	–	–
MLP	✓	–	✓	–	–	–	–

4 Experiments

In this section, we investigate the behavior of the DAGP model. We use an implementation of DAGP in TensorFlow [1] based on GPflow [20] and the implementation of DSVI [23]. Table 1 compares qualitative properties of DAGP and related work. All models can solve standard regression problems and yield unimodal predictive distributions or, in case of multi-layer perceptrons (MLP), a single point estimate. Both standard Gaussian process regression (GPR) and MLP do not impose structure which enables the models to handle multi-modal data. Mixture density networks (MDN) [4] and the infinite mixtures of Gaussian processes (RGPR) [21] model yield multi-modal posteriors through mixtures with many components but do not solve an association problem. Similarly, Bayesian neural networks with added latent variables (BNN+LV) [10] represent such a mixture through a continuous latent variable. Both the overlapping mixtures of Gaussian processes (OMGP) [18] model and DAGP explicitly model the data association problem and yield independent models for the different generating processes. However, OMGP assumes global relevance of the different modes. In contrast, DAGP infers a spacial posterior of this relevance. We evaluate our model on three problems to highlight the following advantages of the explicit structure of DAGP:

Interpretable priors give structure to ill-posed data association problems. In Section 4.1, we consider a noise separation problem, where a signal of interest is disturbed with uniform noise. To solve this problem, assumptions about what constitutes a signal are needed. The hierarchical structure of DAGP allows us to formulate independent and interpretable priors on the noise and signal processes.

Predictive associations represent knowledge about the relevance of generative processes. In Section 4.2, we investigate the implicit incentive of DAGP to explain data using as few processes as possible. Additional to a joint posterior explaining the data, DAGP also gives insight into the relative importance of the different

Table 2. Results on the ChoiceNet data set. The gray part of the table shows RMSE results for baseline models from [7]. For our experiments using the same setup, we report RMSE comparable to the previous results together with MLL. Both are calculated based on a test set of 1000 equally spaced samples of the noiseless underlying function.

Outliers	DAGP	OMGP	DAGP	OMGP	CN	MDN	MLP	GPR	RGPR
	MLL	MLL	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE
0%	2.86	2.09	0.008	0.005	0.034	0.028	0.039	0.008	0.017
20%	2.71	1.83	0.008	0.005	0.022	0.087	0.413	0.280	0.013
40%	2.12	1.60	0.005	0.007	0.018	0.565	0.452	0.447	1.322
60%	0.874	1.23	0.031	0.006	0.023	0.645	0.636	0.602	0.738
80%	0.126	-1.35	0.128	0.896	0.084	0.778	0.829	0.779	1.523

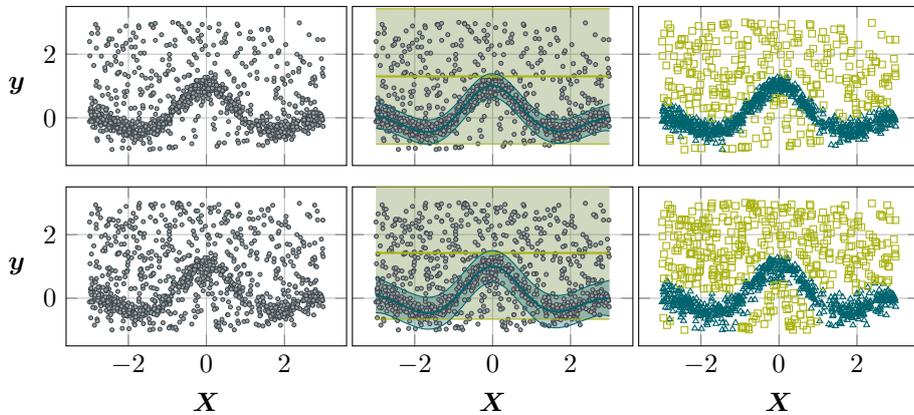


Fig. 3. DAGP on the ChoiceNet data set with 40% outliers (upper row) and 60% outliers (lower row). We show the raw data (left), joint posterior (center) and assignments (right). The bimodal DAGP identifies the signal perfectly up to 40% outliers. For 60% outliers, some of the noise is interpreted as signal, but the latent function is still recovered.

processes in different parts of the input space. DAGP is able to explicitly recover the changing number of modes in a data set.

Separate models for independent generating processes avoid model pollution. In Section 4.3, we simulate a system with multiple operational regimes via mixed observations of two different cart-pole systems. DAGP successfully learns an informative joint posterior by solving the underlying association problem. We show that the DAGP posterior contains two separate models for the two original operational regimes.

4.1 Noise Separation

We consider an experiment based on a noise separation problem. We apply DAGP to a one-dimensional regression problem with uniformly distributed

asymmetric outliers in the training data. We use a task proposed by Choi et al. [7] where we sample $x \in [-3, 3]$ uniformly and apply the function $f(x) = (1 - \delta)(\cos(\pi/2 \cdot x) \exp(-(x/2)^2) + \gamma) + \delta \cdot \epsilon$, where $\delta \sim \mathcal{B}(\lambda)$, $\epsilon \sim \mathcal{U}(-1, 3)$ and $\gamma \sim \mathcal{N}(0, 0.15^2)$. That is, a fraction λ of the training data, the outliers, are replaced by asymmetric uniform noise. We sample a total of 1000 data points and use 25 inducing points for every GP in our model.

Every generating process in our model can use a different kernel and therefore encode different prior assumptions. For this setting, we use two processes, one with a squared exponential kernel and one with a white noise kernel. This encodes the problem statement that every data point is either part of the signal we wish to recover or uncorrelated noise. To avoid pathological solutions for high outlier ratios, we add a prior to the likelihood variance of the first process, which encodes our assumption that there actually is a signal in the training data.

The model proposed in [7], called ChoiceNet (CN), is a specific neural network structure and inference algorithm to deal with corrupted data. In their work, they compare their approach to the MLP, MDN, GPR, and RGPR models. We add experiments for both DAGP and OMGP. Table 2 shows results for outlier rates varied from 0% to 80%. Besides the root mean squared error (RMSE) reported in [7], we also report the mean test log likelihood (MLL).

Since we can encode the same prior knowledge about the signal and noise processes in both OMGP and DAGP, the results of the two models are comparable: For low outlier rates, they correctly identify the outliers and ignore them, resulting in a predictive posterior of the signal equivalent to standard GP regression without outliers. In the special case of 0% outliers, the models correctly identify that the process modelling the noise is not necessary, thereby simplifying to standard GP regression. For high outlier rates, stronger prior knowledge about the signal is required to still identify it perfectly. Figure 3 shows the DAGP posterior for an outlier rate of 60%. While the function has still been identified well, some of the noise is also explained using this process, thereby introducing slight errors in the predictions.

4.2 Multimodal Data

Our second experiment applies DAGP to a multimodal data set. The data, together with recovered posterior attributions, can be seen in Figure 4. We uniformly sample 350 data points in the interval $x \in [-2\pi, 2\pi]$ and obtain $y_1 = \sin(x) + \epsilon$, $y_2 = \sin(x) - 2 \exp(-1/2 \cdot (x - 2)^2) + \epsilon$ and $y_3 = -1 - 3/8\pi \cdot x + 3/10 \cdot \sin(2x) + \epsilon$ with additive independent noise $\epsilon \sim \mathcal{N}(0, 0.005^2)$. The resulting data set $\mathcal{D} = \{(x, y_1), (x, y_2), (x, y_3)\}$ is trimodal in the interval $[0, 5]$ and is otherwise bimodal with one mode containing double the amount of data than the other.

We use squared exponential kernels as priors for both the $f^{(k)}$ and $\alpha^{(k)}$ and 25 inducing points in every GP. Figure 4 shows the posterior of a DAGP with $K = 4$ modes applied to the data, which correctly identified the underlying functions. The figure shows the posterior belief about the assignments \mathbf{A} and illustrates that DAGP recovered that it needs only three of the four available modes to

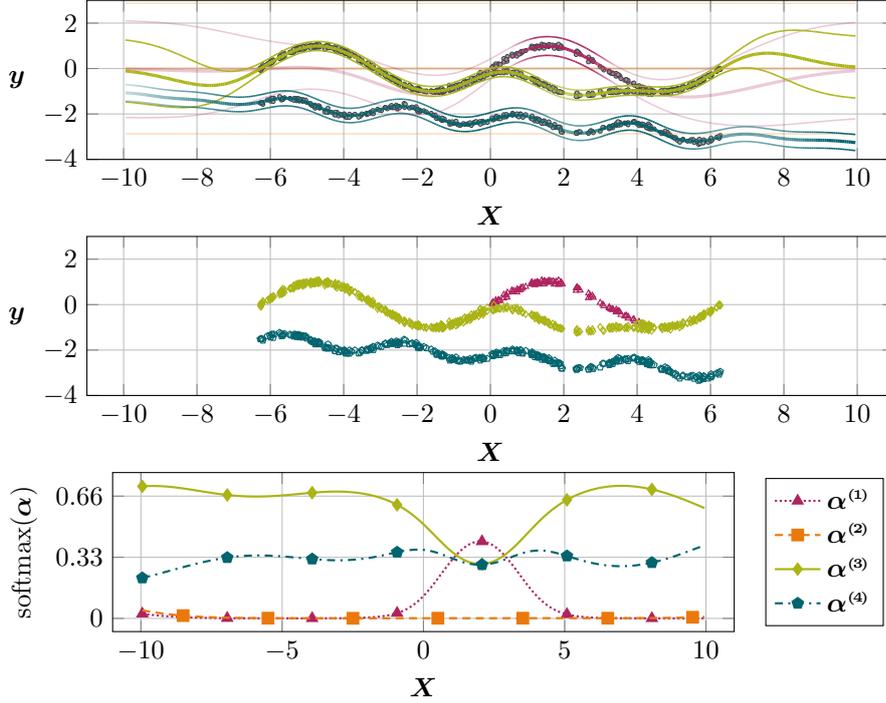


Fig. 4. The DAGP posterior on an artificial data set with bimodal and trimodal parts. The joint predictions (top) are mixtures of four Gaussians weighed by the assignment probabilities α (bottom). The weights are represented via the opacity of the modes. The model has learned that the mode $k = 2$ is irrelevant, that the mode $k = 1$ is only relevant around the interval $[0, 5]$. Outside this interval, the mode $k = 3$ is twice as likely as the mode $k = 4$. The concrete assignments \mathbf{a} (middle) of the training data show that the mode $k = 1$ is only used to explain observations where the training data is trimodal. The mode $k = 2$ is never used.

explain the data. One of the modes is only assigned points in the interval $[0, 5]$ where the data is actually trimodal.

This separation is explicitly represented in the model via the assignment processes α (bottom panel in Figure 4). Importantly, DAGP does not only cluster the data with respect to the generating processes but also infers a factorization of the input space with respect to the relative importance of the different processes. The model has disabled the mode $k = 2$ in the complete input space and has learned that the mode $k = 1$ is only relevant in the interval $[0, 5]$ where the three enabled modes each explain about a third of the data. Outside this interval, the model has learned that one of the modes has about twice the assignment probability than the other one, thus correctly reconstructing the true generative process. The DAGP is implicitly incentivized to explain the data using as few modes as possible through the likelihood term of the inferred \mathbf{a}_n in (6). At

Table 3. Results on the cart-pole data set. We report mean log likelihoods with their standard error for ten runs. The upper results are obtained by training the model on the mixed data set and evaluating it jointly (left) on multi-modal predictions. We evaluate the two inferred sub-models for the default system (center) and short-pole system (right). We provide gray baseline comparisons with BNN+LV and GPR models which cannot solve the data assignment problem. BNN+LV yields joint predictions which cannot be separated into sub-models. Specialized GPR models trained the individual training sets give a measure of the possible performance if the data assignment problem would be solved perfectly.

	Mixed		Default only	Short-pole only
	Train	Test	Test	Test
DAGP	0.575 ± 0.013	0.521 ± 0.009	0.844 ± 0.002	0.602 ± 0.005
DAGP 2	0.548 ± 0.012	0.519 ± 0.008	0.859 ± 0.001	0.599 ± 0.011
DAGP 3	0.527 ± 0.004	0.491 ± 0.003	0.852 ± 0.002	0.545 ± 0.012
OMGP	-1.04 ± 0.02	-1.11 ± 0.03	0.66 ± 0.02	-0.81 ± 0.12
BNN+LV	0.519 ± 0.005	0.524 ± 0.005	—	—
GPR Mixed	0.452 ± 0.003	0.421 ± 0.003	—	—
GPR Default	—	—	0.867 ± 0.001	-7.54 ± 0.14
GPR Short	—	—	-5.14 ± 0.04	0.792 ± 0.003

$x = -10$ the inferred modes and assignment processes start reverting to their respective priors away from the data.

4.3 Mixed Cart-pole Systems

Our third experiment is based on the cart-pole benchmark for reinforcement learning as described by Barto et al. [3] and implemented in OpenAI Gym [6]. In this benchmark, the objective is to apply forces to a cart moving on a frictionless track to keep a pole, which is attached to the cart via a joint, in an upright position. We consider the regression problem of predicting the change of the pole’s angle given the current state of the cart and the action applied. The current state of the cart consists of the cart’s position and velocity and the pole’s angular position and velocity. To simulate a dynamical system with changing system characteristics our experimental setup is to sample trajectories from two different cart-pole systems and merging the resulting data into one training set. The task is not only to learn a model which explains this data well, but to solve the association problem introduced by the different system configurations. This task is important in reinforcement learning settings where we study systems with multiple operational regimes.

We sample trajectories from the system by initializing the pole in an almost upright position and then applying 10 uniform random actions. We add Gaussian noise $\epsilon \sim \mathcal{N}(0, 0.01^2)$ to the observed angle changes. To increase the non-linearity of the dynamics, we apply the action for five consecutive time steps and allow the pole to swing freely instead of ending the trajectory after reaching a specific

angle. The data set consists of 500 points sampled from the *default* cart-pole system and another 500 points sampled from a *short-pole* cart-pole system in which we halve the mass of the pole to 0.05 and shorten the pole to 0.1, a tenth of its default length. This short-pole system is more unstable and the pole reaches higher speeds. Predictions in this system therefore have to take the multimodality into account, as mean predictions between the more stable and the more unstable system can never be observed. We consider three test sets, one sampled from the default system, one sampled from the short-pole system, and a mixture of the two. They are generated by sampling trajectories with an aggregated size of 5000 points from each system for the first two sets and their concatenation for the mixed set.

For this data set, we use squared exponential kernels for both the $f^{(k)}$ and $\alpha^{(k)}$ and 100 inducing points in every GP. We evaluate the performance of deep GPs with up to three layers and squared exponential kernels as models for the different functions. As described in [16, 23], we use identity mean functions for all but the last layers and initialize the variational distributions with low covariances. We compare our models with OMGP and three-layer relu-activated Bayesian neural networks with added latent variables (BNN+LV). The latent variables can be used to effectively model multimodalities and stochasticity in dynamical systems for model-based reinforcement learning [11]. We also compare DAGP to three kinds of sparse GPs (GPR) [14]. They are trained on the mixed data set, the default system and the short-pole system respectively and serve as a baseline comparison as these models cannot handle multi-modal data.

Table 3 shows results for ten runs of these models. The GPR model predicts a unimodal posterior for the mixed data set which covers both systems. Its mean prediction is approximately the mean of the two regimes and is physically implausible. The DAGP and BNN+LV models yield informative multi-modal predictions with comparable performance. In our setup, OMGP could not successfully solve the data association problem and thus does not produce a useful joint posterior. The OMGP’s inference scheme is tailored to ordered one-dimensional problems. It does not trivially translate to the 4D cart-pole problem.

As BNN+LV does not explicitly solve the data association problem, the model does not yield sub-models for the two different systems. Similar results would be obtained with the MDN and RGPR models, which also cannot be separated into sub-models. OMGP and DAGP yield such sub-models which can independently be used for predictions in the default or short-pole systems. Samples drawn from these models can be used to generate physically plausible trajectories in the respective system. OMGP fails to model the short-pole system but does yield a viable model for the default system which evolves more slowly due to higher torque and is therefore easier to learn. In contrast, the two sub-models inferred by DAGP perform well on their respective systems, showing that DAGP reliably solves the data association problem and successfully avoids model pollution by separating the two systems well. Given this separation, shallow and deep models for the two modes show comparable performance. The more expressive deep GPs

model the default system slightly better while sacrificing performance on the more difficult short-pole system.

5 Conclusion

We have presented a fully Bayesian model for the data association problem. Our model factorises the observed data into a set of independent processes and provides a model over both the processes and their association to the observed data. The data association problem is inherently ill-constrained and requires significant assumptions to recover a solution. In this paper, we make use of interpretable GP priors allowing global a priori information to be included into the model. Importantly, our model is able to exploit information both about the underlying functions and the association structure. We have derived a principled approximation to the marginal likelihood which allows us to perform inference for flexible hierarchical processes. In future work, we would like to incorporate the proposed model in a reinforcement learning scenario where we study a dynamical system with different operational regimes.

Bibliography

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Software available from tensorflow.org.
- [2] Y. Bar-Shalom. *Tracking and Data Association*. Academic Press Professional, Inc., San Diego, CA, USA, 1987. ISBN 0-120-79760-7.
- [3] A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):834–846, September 1983. ISSN 0018-9472. <https://doi.org/10.1109/TSMC.1983.6313077>.
- [4] Christopher M. Bishop. Mixture density networks. Technical report, 1994.
- [5] Erik Bodin, Neill D. F. Campbell, and Carl Henrik Ek. Latent Gaussian Process Regression. *arXiv:1707.05534 [cs, stat]*, July 2017.
- [6] Greg Brockman, Vicki Cheung, Ludwig Pettersson, et al. OpenAI Gym. *arXiv:1606.01540 [cs]*, June 2016.
- [7] Sungjoon Choi, Sanghoon Hong, and Sungbin Lim. ChoiceNet: Robust Learning by Revealing Output Correlations. *arXiv:1805.06431 [cs, stat]*, May 2018.
- [8] Ingemar J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10:53–66, 1993.
- [9] Andreas Damianou and Neil Lawrence. Deep Gaussian Processes. In *Artificial Intelligence and Statistics*, pages 207–215, April 2013.
- [10] Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Learning and Policy Search in Stochastic Dynamical Systems with Bayesian Neural Networks. *arXiv:1605.07127 [cs, stat]*, May 2016.

- [11] Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of Uncertainty in Bayesian Deep Learning for Efficient and Risk-sensitive Learning. In *International Conference on Machine Learning*, pages 1192–1201, 2018.
- [12] Daniel Hein, Stefan Depeweg, Michel Tokic, et al. A benchmark environment motivated by industrial control problems. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, Honolulu, HI, November 2017. IEEE. ISBN 978-1-5386-2726-6. <https://doi.org/10.1109/SSCI.2017.8280935>.
- [13] James Hensman, Nicolo Fusi, and Neil D. Lawrence. Gaussian Processes for Big Data. In *Uncertainty in Artificial Intelligence*, page 282. Citeseer, 2013.
- [14] James Hensman, Alexander G. de G. Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. *Journal of Machine Learning Research*, 38:351–360, 2015.
- [15] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.
- [16] Markus Kaiser, Clemens Otte, Thomas Runkler, and Carl Henrik Ek. Bayesian Alignments of Warped Multi-Output Gaussian Processes. In S. Bengio, H. Wallach, H. Larochelle, et al., editors, *Advances in Neural Information Processing Systems 31*, pages 6995–7004. Curran Associates, Inc., 2018.
- [17] Diederik P Kingma, Tim Salimans, and Max Welling. Variational Dropout and the Local Reparameterization Trick. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2575–2583. Curran Associates, Inc., 2015.
- [18] Miguel Lázaro-Gredilla, Steven Van Vaerenbergh, and Neil D. Lawrence. Overlapping mixtures of Gaussian processes for the data association problem. *Pattern Recognition*, 45(4):1386–1395, 2012.
- [19] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. *arXiv:1611.00712 [cs, stat]*, November 2016.
- [20] Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, et al. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, 2017.
- [21] Carl E. Rasmussen and Zoubin Ghahramani. Infinite Mixtures of Gaussian Process Experts. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 881–888. MIT Press, 2002.
- [22] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. January 2014.
- [23] Hugh Salimbeni and Marc Deisenroth. Doubly Stochastic Variational Inference for Deep Gaussian Processes. In I. Guyon, U. V. Luxburg, S. Bengio, et al., editors, *Advances in Neural Information Processing Systems 30*, pages 4588–4599. Curran Associates, Inc., 2017.

- [24] Michalis K. Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *AISTATS*, volume 5, pages 567–574, 2009.
- [25] Volker Tresp. Mixtures of Gaussian Processes. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 654–660. MIT Press, 2001.