Healing Products of Gaussian Process Experts

Samuel Cohen [*1] **Rendani Mbuvha** [*2 3] **Tshilidzi Marwala** [2] **Marc Peter Deisenroth** [1 2]

Abstract

Gaussian processes (GPs) are nonparametric Bayesian models that have been applied to regression and classification problems. One of the approaches to alleviate their cubic training cost is the use of local GP experts trained on subsets of the data. In particular, product-of-expert models combine the predictive distributions of local experts through a tractable product operation. While these expert models allow for massively distributed computation, their predictions typically suffer from erratic behaviour of the mean or uncalibrated uncertainty quantification. By calibrating predictions via a tempered softmax weighting, we provide a solution to these problems for multiple product-of-expert models, including the generalised product of experts and the robust Bayesian committee machine. Furthermore, we leverage the optimal transport literature and propose a new product-of-expert model that combines predictions of local experts by computing their Wasserstein barycenter, which can be applied to both regression and classification.

1. Introduction

Gaussian processes (GPs) (Rasmussen & Williams, 2006) are nonparametric stochastic processes that have been applied extensively to regression and classification problems. However, their cubic training and quadratic prediction cost hinders their application in large-scale problems. Different approaches alleviate this issue, including sparse approximations (Snelson & Ghahramani, 2006; Csato & Opper, 2002; Quiñonero Candela & Rasmussen, 2005; Titsias, 2009), the exploitation of structural assumptions (Wilson & Nickisch, 2015) and local-expert models (Tresp, 2000a; Rasmussen & Ghahramani, 2001; Cao & Fleet, 2014; Deisenroth & Ng, 2015; Rullière et al., 2018; Trapp et al., 2019).

Sparse approximations effectively reduce the rank of the covariance matrix through inducing inputs, reducing the training cost from $O(n^3)$ to $O(nm^2)$, where m is the number of inducing points and n is the size of the training dataset. Optimisation consists of jointly learning kernel hyperparameters and inducing locations. In particular, Titsias (2009) treats inducing locations as variational parameters and optimises them and the kernel hyperparameters by maximising a lower bound on the marginal likelihood. Hensman et al. (2013) scale this approach by introducing mini-batching, reducing the complexity to $O(m^3)$, while Gal et al. (2014) reparametrise the problem to allow for distributed inference.

An alternative to sparse GP approximations is to use local experts. Here, the training dataset is partitioned into J subsets of size m where $m \ll n$. Then, J local GP experts are trained on each of these subsets, thereby reducing the training complexity to $O(Jm^3)$. Importantly, this approach scales to large datasets because training and prediction with each expert can be distributed across computing units (Deisenroth & Ng, 2015). For instance, Rasmussen & Ghahramani (2001); Tresp (2000a); Trapp et al. (2019) consider mixture-of-expert models (MoEs). In particular, Trapp et al. (2019) propose a sum-product network with local-expert GP leaves allowing for tractable and exact posterior inference. Other approaches leverage product-of-experts models (PoEs) (Tresp, 2000b; Cao & Fleet, 2014), whereby a global prediction can be obtained by means of averaging the predictions of local experts. Generalisations of these models can control the relevance of different experts when making predictions (Cao & Fleet, 2015; Deisenroth & Ng, 2015; Liu et al., 2018).

In this work, we focus on PoEs because closed-form inference and training are tractable, which is not the case with typical MoEs. However, previous PoE approaches to combining predictions at test time suffer from unrealistic over- or under-estimation of the variance and erratic mean behaviours. This holds especially when the number of points m assigned to each expert is low, in which case a significant number of experts are weak (Deisenroth & Ng, 2015). These approaches are thus not overly robust to variations

[*]Equal contribution [1]Department of Computer Science, University College London, UK [2]Institute of Intelligent Systems, University of Johannesburg, South Africa [3]School of Statistics and Actuarial Science, University of the Witwatersrand, South Africa. Correspondence to: Samuel Cohen <samuel.cohen.19@ucl.ac.uk>, Rendani Mbuvha <rendani.mbuvha@wits.ac.za>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

in m, which is a significant shortcoming. Unfortunately, scalability requires the number of points per expert to be reasonably small due to the $O(m^3)$ scaling of individual experts. We propose a solution to these problems by controlling the sparsity of expert weights through a tempered softmax at test time, leveraging tools from the the extensive uncertainty calibration literature (Platt, 1999; Bishop & Svensén, 2003; Guo et al., 2017). We also propose a novel principled PoE approach arising from the optimal transport literature, which we name the barycenter of GPs, and demonstrate that its performance is competitive to the best PoE models on small and large-scale datasets. We demonstrate empirically that calibrating expert weights lead to substantial performance gains in both mean prediction and uncertainty quantification. We also discuss common failures of PoE models extensively and propose guidelines to remediating these.

Contributions: 1) We introduce a new method for averaging GP experts based on optimal transport theory that performs competitively with the best-performing PoE models. 2) We propose a solution to the shortcomings of previously proposed PoEs, based on controlling the weight sparsity. 3) We analyse and disentangle contradictory results arising from recent GP experts papers on commonly used PoEs.

2. Gaussian Processes

Gaussian processes are powerful nonparametric Bayesian models, often used for regression. A GP is defined as a collection of random variables, every finite subset of which is jointly Gaussian distributed (Rasmussen & Williams, 2006). GPs are fully defined by a mean $m(\cdot)$ and a kernel $k(\cdot,\cdot)$.

Consider a regression problem with a training dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ of n noisy observations $y_i = f(\boldsymbol{x}_i) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_y^2)$. With a GP prior on f, it follows that $f(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{m}_x, \boldsymbol{K}_x + \sigma_y^2 \boldsymbol{I})$ where $(\boldsymbol{m}_x)_i = m(\boldsymbol{x}_i)$ and $(K_x)_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The mean and variance of the Gaussian posterior predictive distribution of the function value $f(\boldsymbol{x}_*)$ at a test point $\boldsymbol{x}_*$, are given by

$$\mathbb{E}[f_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}] = \boldsymbol{m}_{\boldsymbol{x}_*} + \boldsymbol{k}_*^T (\boldsymbol{K}_x + \sigma_y^2 \boldsymbol{I})^{-1} (\boldsymbol{y} - \boldsymbol{m}_{\boldsymbol{x}}),$$
$$\text{var}[f_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}] = \boldsymbol{k}_{**} - \boldsymbol{k}_*^T (\boldsymbol{K}_x + \sigma_y^2 \boldsymbol{I})^{-1} \boldsymbol{k}_*,$$

respectively, where $\boldsymbol{k}_{**} = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$ and $\boldsymbol{k}_* = k(\boldsymbol{X}, \boldsymbol{x}_*)$. Here $\boldsymbol{X}, \boldsymbol{y}$ contain the training inputs and targets, respectively. Kernel hyperparameters and the noise parameter σ_y are learned by maximising the log-marginal likelihood

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \log \mathcal{N}(\boldsymbol{y}|\boldsymbol{m}_x, \boldsymbol{K}_{xx} + \sigma_y^2 \boldsymbol{I}). \quad (1)$$

Computing (1) requires the inversion of the matrix $\boldsymbol{K}_{xx} + \sigma_y^2 \boldsymbol{I} \in \mathbb{R}^{n \times n}$, so that GP training scales in $O(n^3)$, where n is the size of the training dataset. Optimizing the log-marginal likelihood in (1) and the computation of the posterior predictive distribution at a test input $\boldsymbol{x}_*$ become computationally intractable for large training sets (Rasmussen & Williams, 2006).

Several approaches have been explored to avoid the cubic training cost of GPs. These are mostly based on either sparse approximations and structure-exploiting assumptions to the covariance matrix (Quiñonero Candela & Rasmussen, 2005; Titsias, 2009; Hensman et al., 2013; Wilson & Nickisch, 2015) or training distributed (weak) experts on subsets of the full dataset (Tresp, 2000b; Cao & Fleet, 2014; Deisenroth & Ng, 2015; Trapp et al., 2019; Liu et al., 2018). An alternative path towards scaling GPs is to use large-scale computing infrastructure and incomplete Cholesky decomposition of large kernel matrices (Wang et al., 2019).

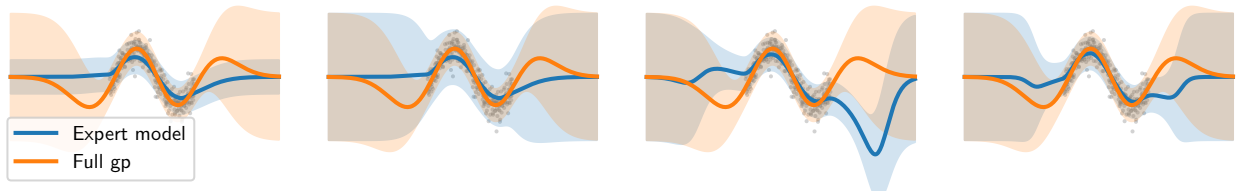
2.1. Sparse Gaussian Processes

Sparse GPs (Quiñonero Candela & Rasmussen, 2005; Snelson & Ghahramani, 2006) leverage inducing inputs to reduce the rank of the matrix to be inverted. Sparse variational GPs extend this by introducing a variational approximation to the posterior (Titsias, 2009), treating inducing inputs as variational parameters, and mini-batching (Hensman et al., 2013) to scale. Wilson & Nickisch (2015) exploit structural assumptions and combine inducing-point approaches with Kronecker and Toeplitz methods to perform kernel approximations leading to increased scalability. The approximation quality of sparse GPs relies on the number of inducing points, and a large number of these can be required to represent the local structures of fast varying functions.

2.2. Gaussian Process Experts

Another approach to scaling GPs to large datasets is to use expert models. Here, multiple GPs are trained on subsets of the data, and predictions are recombined using either a product-of-expert (log-opinion pool) approach (Hinton, 1999; Tresp, 2000b; Cao & Fleet, 2014; Deisenroth & Ng, 2015; Rullière et al., 2018; Bertone et al., 2019), or a mixture-of-expert (linear-opinion pool) approach (Tresp, 2000a; Rasmussen & Ghahramani, 2001; Trapp et al., 2019). MoEs are useful in heteroskedastic and nonstationary settings, but do not typically allow tractable posterior inference, by contrast with PoEs.

In this paper, we thus focus on product-of-expert models with M experts, which all share hyperparameters. We first describe the training of such models. Assuming a full GP is the model we seek to approximate, sharing kernel hyperparameters automatically regularises the population of experts: individual experts can not overfit to the local subset of the data they are fed with due to this shared set of hyperparameters. Assuming independence across experts (given the



(a) PoE (b) gPoE (c) BCM (d) rBCM

Figure 1. Different expert models trained on synthetic data with three points per GP expert on a dataset of 300 observations. (a) PoE; (b) gPoE; (c) BCM; (d) rBCM. All models display some shortcomings in their vanilla forms. For instance (a): over-confidence, (b) under-confidence within data region, and (c)-(d) erratic mean in the transitioning region.

training data), the log-marginal likelihood is

$$\log p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) = \sum_{j=1}^{J} \log p_j(\boldsymbol{y}^{(j)}|\boldsymbol{X}^{(j)},\boldsymbol{\theta}), \qquad (2)$$

where $\{\boldsymbol{X}^{(j)}, \boldsymbol{y}^{(j)}\}$ is the data assigned to the j^{th} expert. To train the model, we maximise the log-marginal likelihood (2) with respect to the (shared) kernel hyperparameters (Deisenroth & Ng, 2015). Training can be massively distributed across diverse compute clusters, enabling scaling with total time complexity $O(Jm^3)$ where J is the number of experts, and $m \ll n$ is the size of the training set of each local expert. With J compute nodes, the computational complexity per node reduces to $O(m^3)$. This is in stark contrast to the $O(n^3)$ scaling of full GPs.

In the following, we describe the process of predicting with product-of-GP-experts models. In particular, we introduce several approaches to recombining predictions from trained experts. We note that an important particularity of these models is that all predictive distributions $p(f_*|\boldsymbol{x}_*)$ of function values are Gaussians, which is not the case with MoEs. Also, throughout the paper, aggregation is performed in function space, and the likelihood is subsequently applied.

(Generalised) product of experts – (g)PoE The (g)PoE aggregates predictions of M experts at test point $\boldsymbol{x}_*$ via

$$p(f_*|\boldsymbol{x}_*) \propto \prod_{j=1}^{J} p_j^{\beta_j(\boldsymbol{x}_*)}(f_*|\boldsymbol{x}_*, \mathcal{D}^{(j)}), \qquad (3)$$

where the predictive mean and precision are

$$m_{(g)poe}(\boldsymbol{x}_*) = \sigma^2_{(g)poe}(\boldsymbol{x}_*) \sum_{j=1}^{J} \beta_j(\boldsymbol{x}_*)\sigma_j^{-2}(\boldsymbol{x}_*)m_j(\boldsymbol{x}_*),$$

$$\sigma^{-2}_{(g)poe}(\boldsymbol{x}_*) = \sum_{j=1}^{J} \beta_j(\boldsymbol{x}_*)\sigma_j^{-2}(\boldsymbol{x}_*).$$

respectively. Here, $\mathcal{D}^{(j)} = \{\boldsymbol{X}^{(j)}, \boldsymbol{y}^{(j)}\}$ is the data assigned to expert j, $\beta_j(\boldsymbol{x}_*)$ controls the contribution of expert j at $\boldsymbol{x}_*$ (typically a measure of its confidence at $\boldsymbol{x}_*$), and the PoE model is recovered when setting $\beta_j(\boldsymbol{x}_*) = 1$ for all j. As the number of experts J increases, the PoE's aggregated variance vanishes, which leads to overconfident predictions (Deisenroth & Ng, 2015; Liu et al., 2018). An illustration of such behaviour is shown in Figure 1(a).

The gPoE with uniform weights $\sum_j \beta_j(\boldsymbol{x}_*) = 1$ falls back to the prior far from training points, which is a desirable property. However, a drawback is that it over-estimates the variance close to training points (Deisenroth & Ng, 2015) when setting the weights uniformly ($\beta_j(\boldsymbol{x}_*) = \frac{1}{J}$). We also observe such behaviour in Figure 1(b).

(Robust) Bayesian committee machine – (r)BCM The (robust) Bayesian committee machine (r)BCM (Tresp, 2000b; Deisenroth & Ng, 2015) assumes conditional independence $\mathcal{D}_i \perp\!\!\!\perp \mathcal{D}_j | f_*$. By repeated application of Bayes' theorem, we obtain the predictive distribution

$$p(f_*|\boldsymbol{x}_*) = \frac{\prod_{j=1}^{J} p_j^{\beta_j(\boldsymbol{x}_*)}(f_*|\boldsymbol{x}_*, \mathcal{D}^{(j)})}{p^{-1+\sum_j \beta_j(\boldsymbol{x}_*)}(f_*|\boldsymbol{x}_*)} \qquad (4)$$

at test point $\boldsymbol{x}_*$. Then the predictive mean and precision are

$$m_{(r)bcm}(\boldsymbol{x}_*) = \sigma^2_{(r)bcm}(\boldsymbol{x}_*) \sum_{j=1}^{J} \beta_j(\boldsymbol{x}_*)\sigma_j^{-2}(\boldsymbol{x}_*)m_j(\boldsymbol{x}_*),$$

$$\sigma^{-2}_{(r)bcm}(\boldsymbol{x}_*) = \sum_{j=1}^{J} \beta_j(\boldsymbol{x}_*)(\sigma_j^{-2}(\boldsymbol{x}_*) - \sigma_*^{-2}) + \sigma_*^{-2},$$

respectively. BCM is recovered when $\beta_j(\boldsymbol{x}_*) = 1$ for all j. This predictive distribution guarantees that the model falls back to the prior far from training data. However, the BCM exhibits uncharacteristic behaviour in regions transitioning from high to low-density data (Deisenroth & Ng, 2015); see Figure 1(c). The rBCM mitigates some of the issues of the BCM and allows for flexible weighting of GP experts, via $\beta_j(\boldsymbol{x}_*)$, but it still exhibits problematic behaviour in regions with density transitioning; see Figure 1(d).

Generalised rBCM – grBCM Liu et al. (2018) proposed the grBCM in which a master expert communicates

with the children experts leading to a consistent predictive distribution of the form

$$p(y_*|\boldsymbol{x}_*) = \frac{\prod_{j=2}^{J} p_{+j}^{\beta_j(\boldsymbol{x}_*)}(y_*|\boldsymbol{x}_*, \mathcal{D}^{(+j)})}{p^{-1+\sum_{j=2}^{p}\beta_j(\boldsymbol{x}_*)}(\mathcal{D}^c|y_*, \boldsymbol{x}_*)}. \quad (5)$$

$\mathcal{D}^c$ is the global data assigned to the master expert, and $\mathcal{D}^{(+j)} = \{\mathcal{D}^c, \mathcal{D}^{(j)}\}$ is the data of the j^{th} expert aggregated with the data of the master expert. The predictive mean and variance of (5) are

$$m_{grb}(\boldsymbol{x}_*) = \sigma^2_{grb}(\boldsymbol{x}_*)\Big[\sum_{j=2}^{J}\beta_j(\boldsymbol{x}_*)\sigma^{-2}_{+j}(\boldsymbol{x}_*)m_{+j}(\boldsymbol{x}_*) - (\sum_{j=2}\beta_j - 1)\sigma_c^{-2}(\boldsymbol{x}_*)\mu_c(\boldsymbol{x}_*)\Big],$$

$$\sigma^{-2}_{grb}(\boldsymbol{x}_*) = \sum_{j=2}^{J}\beta_j(\boldsymbol{x}_*)(\sigma^{-2}_{+j}(\boldsymbol{x}_*) - \sigma_c^{-2}(\boldsymbol{x}_*)) + \sigma_c^{-2}(\boldsymbol{x}_*).$$

where $m_{+j}(\boldsymbol{x}_*)$ and $\sigma^2_{+j}(\boldsymbol{x}_*)$ are the predictive mean and variance of the j^{th} expert at $\boldsymbol{x}_*$, conditioned on the aggregated dataset $\mathcal{D}^{(+j)}$. Liu et al. (2018) perform aggregation in y-space, in contrast to Deisenroth & Ng (2015), who perform it in f-space. The former is not directly applicable in non-conjugate cases (e.g., classification), and can lead to erratic mean and variance behaviours (especially when used for rBCM/BCM/PoE as observed in (Liu et al., 2018; Zhang & Williamson, 2019)). We therefore consider in this paper aggregation in f-space and discuss differences between both approaches further in later sections.

Likelihoods As expert averaging is performed in function space throughout the paper, we will need to map the aggregated predictive GP distribution $p(f_*)$ through a likelihood function to predict labels y_*. In the conjugate regression case with a Gaussian likelihood, this can be done in closed form (Rasmussen & Williams, 2006). For classification, we consider non-conjugate likelihoods, such as the Bernoulli or Poisson likelihoods. Since the aggregated predictive distribution $p(f_*)$ in PoEs is Gaussian, we obtain the expected predicted label by averaging under the posterior predictive latent distribution

$$\mathbb{E}[y_*|\boldsymbol{x}_*] = \int \phi(f(\boldsymbol{x}_*))\mathcal{N}(f_*|m(\boldsymbol{x}_*), \sigma^2(\boldsymbol{x}_*))df_*, \quad (6)$$

where ϕ is a classification likelihood (e.g., Bernoulli, Probit). The integral in (6) is intractable, but we can resort to standard approximate inference techniques for GP classification, such as MAP estimation, Laplace approximation, expectation propagation, variational inference, or numerical integration (Rasmussen & Williams, 2006; Hensman et al., 2015). Similarly, the marginal likelihood, which we use for training the experts, becomes intractable. Therefore, we use stochastic variational inference to train models in that setting (Hensman et al., 2015), and apply the same strategies for training and prediction with other GP expert models.

3. Barycenters of Predictive Distributions

Now, we propose a new way of combining experts' predictions leveraging optimal transport theory. We begin by introducing two important tools, namely the Wasserstein distance and barycenter between 1D Gaussians, noting that both can be computed using simple closed-form formulas.

Given two Gaussians $\mu = \mathcal{N}(\boldsymbol{m}_1, \boldsymbol{K}_1)$ and $\nu = \mathcal{N}(\boldsymbol{m}_2, \boldsymbol{K}_2)$, we define the 2-Wasserstein distance between them as (Villani, 2008)

$$\mathcal{W}_2^2(\mu, \nu) = \|\boldsymbol{m}_1 - \boldsymbol{m}_2\|_2^2 + Tr\Big(\boldsymbol{K}_1 + \boldsymbol{K}_2 - 2(\boldsymbol{K}_1^{\frac{1}{2}}\boldsymbol{K}_2\boldsymbol{K}_1^{\frac{1}{2}})^{\frac{1}{2}}\Big). \quad (7)$$

Equation (7) can be interpreted as the minimal expected cost of transporting mass from the first Gaussian μ to the second Gaussian ν.

Given that distance, the barycenter between Gaussian-distributed $\mu_1, ..., \mu_J$ with weights $\boldsymbol{\beta}$ is

$$\bar{\mu} = \arg\min_{\mu} \sum_{j=1}^{J} \beta_j \mathcal{W}_2^2(\mu_j, \mu), \quad (8)$$

where $\sum_j \beta_j = 1$, $0 \leq \beta_j \leq 1$. Álvarez Esteban et al. (2016) show that if $\mu_j = \mathcal{N}(\boldsymbol{m}_j, \boldsymbol{K}_j)$ for all j, the Wasserstein barycenter with weights $\boldsymbol{\beta}$ is itself a Gaussian measure $\bar{\mu} = \mathcal{N}(\bar{\boldsymbol{m}}, \bar{\boldsymbol{K}})$, where

$$\bar{\boldsymbol{m}} = \sum_{j=1}^{J}\beta_j\boldsymbol{m}_j, \quad \bar{\boldsymbol{K}} = \sum_{j=1}^{J}\beta_j(\bar{\boldsymbol{K}}^{\frac{1}{2}}\boldsymbol{K}_j\bar{\boldsymbol{K}}^{\frac{1}{2}})^{\frac{1}{2}}. \quad (9)$$

The authors also propose a fixed-point iteration algorithm to efficiently compute $\bar{\boldsymbol{K}}$ in (9).

In the following, we discuss our approach to aggregating GP experts' predictions for regression and classification. In all product-of-experts models we discussed, each expert computes a predictive distribution of the form $p_j(f(\boldsymbol{x}_*)|\mathcal{D}^{(j)}) = \mathcal{N}(m_j(\boldsymbol{x}_*), \sigma_j^2(\boldsymbol{x}_*))$, where m_j and σ_j^2 are the posterior predictive mean and variance of the jth GP expert at test point $\boldsymbol{x}_*$. Since these distributions (in latent space of f) are all Gaussian (by definition of the GP), we propose combining these into their weighted 2-Wasserstein barycenter using (9), which can be computed in closed form in the one-dimensional case (Bonneel & Pfister, 2013). We

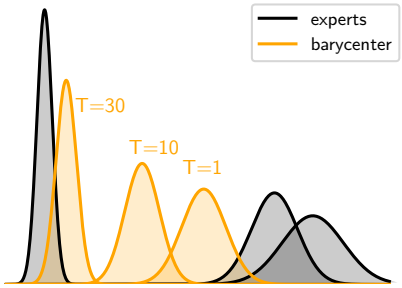


Figure 2. Illustration of the barycenter of GPs with tempered softmax weighting. At x_*, one expert (left) is highly confident about its prediction, and two are highly unconfident (right). As temperature increases, only confident experts get weight (sparsity increases), thus the barycenter is pulled towards the confident expert.

obtain the closed-form Gaussian predictive distribution

$$p(f_*|\boldsymbol{x}_*) = \mathcal{N}(m_{bar}(\boldsymbol{x}_*), \sigma^2_{bar}(\boldsymbol{x}_*)) \tag{10}$$

$$m_{bar}(\boldsymbol{x}_*) = \sum_{j=1}^{J} \beta_j(\boldsymbol{x}_*) m_j(\boldsymbol{x}_*), \tag{11}$$

$$\sigma^2_{bar}(\boldsymbol{x}_*) = \sum_{j=1}^{J} \beta_j(\boldsymbol{x}_*) \sigma^2_j(\boldsymbol{x}_*). \tag{12}$$

The barycenter of GPs is a product-of-experts variant, and the mean and variance of the predictive distribution consist of the weighted average of the predictive means and variances of the experts. Importantly, such weights can be a function of test points, analogously to the gPoE and the rBCM.

We train the barycenter of GPs following the training procedure of other PoEs discussed in Section 2.2, namely by optimising the marginal likelihood (2), and we share expert hyperparameters for regularising the expert pool.

The barycenter of GP's predictive distribution is deeply connected to that of previously proposed PoEs. In particular, the aggregated mean is a weighted mean of the experts' predictive means, which is also the case for other expert models. The aggregated variance is a weighted mean of the experts' variances, which has a similar interpretation to the predictive precision of other PoEs, itself a weighted mean of the experts' precisions. Moreover, the barycenter of GPs falls back to the prior outside of the data regime which is a highly desirable property, and is also the case for gPoE with uniform weights, and rBCM. Further connections are discussed in Section 4.

4. Calibrating Product-of-Experts

In the previous sections, we introduced several approaches to combining predictions of local GP experts, including our proposal, the barycenter of GPs. We also discussed shortcomings of previous PoE approaches in low-data regimes, including under- (Figure 1(a)) and over-estimation of the variance (Figure 1(b)), but also erratic and uncharacteristic behaviours of the mean and variance predictions (Figures 1(c)–1(d)). These behaviours are exacerbated when the number of points assigned per expert is low, which leads to a significant number of weak experts[1].

Whilst exact Gaussian processes are well-known for well-calibrated uncertainty estimates, approximate Bayesian methods fall prey to inferior calibration. These issues in the context of sparse GP approximations are discussed in depth by Bauer et al. (2016). Our aim in this section is to remediate such calibration issues for PoE models. There has been a significant recent emphasis on uncertainty calibration in the deep learning community (Guo et al., 2017), and we will extend tools from this literature to the problem of training product-of-experts-based GP approximations.

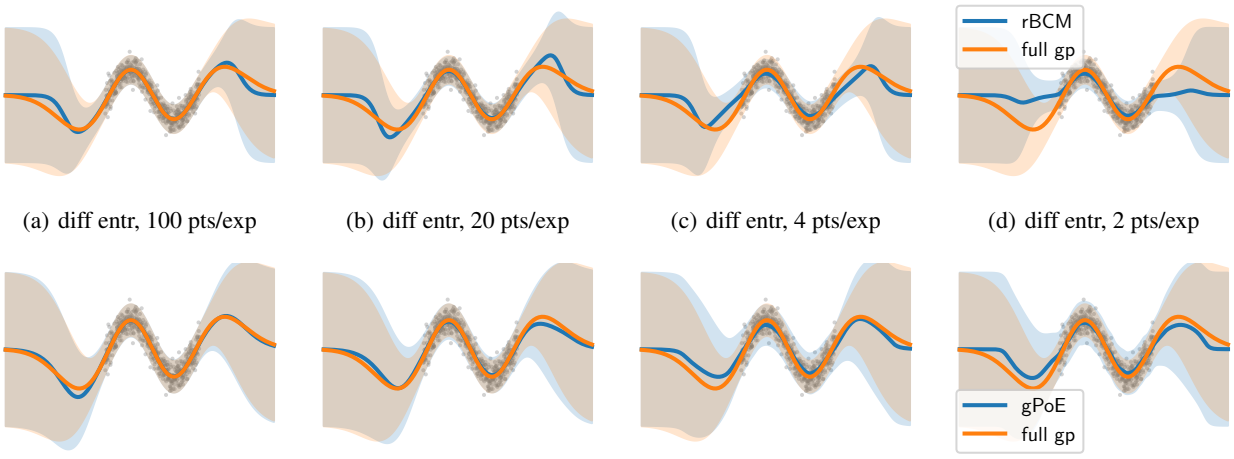
The prevalence of weak experts is significantly affected by the data assignment strategy. For example, when using stationary kernels, clustering-based partition approaches tend to create localised experts which leads to greater weak expert prevalence. The latter approach is intuitively sensible if we choose stationary kernels, as expert approaches can be interpreted as divide-and-conquer strategies. However, this strategy can have disastrous consequences if expert weights are not properly regulated. Indeed, the lower the number of points per expert, the weaker the experts are overall if the training data associated with these experts is not dense in the vicinity of test inputs. This can be observed in Figure 3 (Top), where pathologies arise as the number of points per expert decreases significantly. This is mainly caused by the poorly regulated expert weighting.

In that setting, weight sparsity has to increase to alleviate the weakness of most experts by relying only on locally-calibrated predictions. In the following, we propose a solution to such shortcomings that can be applied to gPoE, rBCM and the barycenter of GPs.

The softmax function provides a natural mechanism for controlling the sparsity of experts' importance weights. In particular, an (inverse) temperature parameters T can directly control the degree of smoothness and sparsity in the resulting weights. Using a temperature-endowed softmax to combat miscalibrated predictions has seen widespread use, ranging from hierarchical mixtures of experts (Bishop & Svensén, 2003) to support vector machines (Platt, 1999) and deep learning (Guo et al., 2017).

We adapt these ideas to weighted ensembles of GP experts, such as the gPoE, the rBCM and the barycenter. We there-

[1]We refer to weak experts as experts that provide calibrated predictions only on local subsets of the data manifold.



(a) diff entr, 100 pts/exp (b) diff entr, 20 pts/exp (c) diff entr, 4 pts/exp (d) diff entr, 2 pts/exp

(e) soft-var, 100pts/exp (f) soft-var, 20 pts/exp (g) soft-var, 4 pts/exp (h) soft-var, 2 pts/exp

Figure 3. Full GP baseline (orange) and expert models (blue) trained on synthetic data with a decreasing number of points per experts (Left to Right), and for different weighting methods: rBCM with differential entropy in Figures (a)–(d) and the gPoE with proposed softmax-variance in Figures (e)–(h). Our method is significantly more robust to variations in the number of points per experts.

fore propose a general expression for expert weights as

$$\beta_j(\boldsymbol{x}_*) \propto \exp(-T\,\psi_j(\boldsymbol{x}_*)), \quad \sum_{j=1}^{M} \beta_j(\boldsymbol{x}_*) = 1, \qquad (13)$$

where T is an (inverse) temperature parameter that controls the sparsity between experts by multiplicatively compounding the weights of stronger experts. The functional $\psi_j(\boldsymbol{x}_*)$ describes the level of confidence of the jth expert at test point $\boldsymbol{x}_*$. We provide an illustration of such framework in Figure 2. In particular, we plot the barycenter of GP experts' predictive distribution at $\boldsymbol{x}_*$ under several temperature values, highlighting that as temperature increases, the barycenter gets pulled towards the most confident expert, i.e., uncertain experts are not given weight in the prediction.

We now discuss the choice of confidence functional ψ. We propose setting $\psi_j(\boldsymbol{x}_*)$ to the posterior predictive variance at $\boldsymbol{x}_*$, i.e.,

$$\psi_j(\boldsymbol{x}_*) = \sigma_j^2(\boldsymbol{x}_*). \qquad (14)$$

Intuitively, this will give high weight to experts with low posterior predictive variance (high confidence) in their prediction. Such experts have training data close to test points (all experts share the same hyperparameters), and should thus have a high contribution in the final prediction. Our proposal can also be combined with the previously proposed differential entropy weighting (Cao & Fleet, 2014)

$$\psi_j(\boldsymbol{x}_*) = \tfrac{1}{2}(\log \sigma_*^2 - \log \sigma_j^2(\boldsymbol{x}_*)) \qquad (15)$$

or with the Wasserstein distance (7), leveraging its closed-form computation in the 1D case (Álvarez Esteban et al., 2016), which has the same complexity as differential entropy. In the infinite temperature limit, weight sparsity is maximised. We show that in this regime, the gPoE, the rBCM and the barycenter of GPs are equivalent, and provide proofs in Appendix A

Proposition 1. *In the infinite-temperature limit $T \to \infty$, and if $\psi_j = \sigma_j^2(\boldsymbol{x}_*)$, the gPoE, the rBCM and the barycenter of GPs have equivalent predictive distributions.*

Intuitively, in such a regime, only the most confident experts have (equal) weight, and as a result the inverse of the weighted sum of precisions of the two former equals the weighted sum of the variances, and thus predictive distributions are equal. Under weaker assumptions, the rBCM and the gPoE are equivalent:

Proposition 2. *If $\sum_j \beta_j(\boldsymbol{x}_*) = 1$ for all $\boldsymbol{x}_*$, then $m_{rbcm}(\boldsymbol{x}_*) = m_{gpoe}(\boldsymbol{x}_*)$ and $\sigma^2_{rbcm}(\boldsymbol{x}_*) = \sigma^2_{gpoe}(\boldsymbol{x}_*)$.*

Proposition 2 highlights that under normalised weights, gPoE and rBCM are equivalent. Therefore, under our weighting proposal, which consists of using normalised tempered softmax functionals, gPoE and rBCM's predictive distributions are equal.

5. Experiments

Throughout this section, we evaluate the performance of our approaches to calibrating GP experts when applied to regression and classification, while comparing with sparse variational methods and previous approaches to local-expert weighting and averaging. We consider performance metrics including the negative log-predictive density (NLPD), and

dataset	N	D	rBCM/gPoE_unif	rBCM/gPoE_var	rBCM_entr	BAR_var	grBCM_f	SVGP$_{500}$	linear	full GP
Concrete	1030	8	0.506 (0.370)	**0.288 (0.342)**	0.292 (**0.343**)	**0.288 (0.342)**	**0.285 (0.339)**	**0.289 (0.338)**	0.953 (0.626)	0.261 (0.330)
Airfoil	1503	5	0.699 (0.474)	**0.411 (0.350)**	**0.409 (0.360)**	**0.411 (0.351)**	**0.413 (0.350)**	**0.409 (0.353)**	1.096 (0.721)	0.358 (0.331)
Parkinsons	5875	20	1.057 (0.713)	**0.101 (0.338)**	0.157 (**0.339**)	**0.100 (0.337)**	0.145 (0.345)	0.554 (0.412)	1.282 (0.871)	0.079 (0.320)
Power	9568	4	0.303 (0.318)	**-0.084 (0.222)**	**-0.079 (0.223)**	**-0.076 (0.224)**	**-0.074 (0.224)**	-0.044 (0.231)	0.098 (0.267)	-0.079 (0.223)
Kin40K	40000	8	1.078 (0.693)	-0.329(0.186)	0.359 (0.191)	-0.339 (0.183)	**-0.432 (0.150)**	0.124 (0.263).	1.419 (1.000)	N/A
Protein	45730	9	1.379 (0.961)	**0.775 (0.582)**	0.799 (0.608)	**0.775 (0.583)**	0.797 (0.607)	1.083 (0.715)	1.257 (0.850)	N/A
Airline	500000	7	1.446 (0.992)	**1.316 (0.911)**	**1.311 (0.906)**	**1.315 (0.911)**	**1.311 (0.905)**	1.329(0.918)	1.381 (0.962)	N/A
average			0.924 (0.645)	**0.354 (0.418)**	0.464 (0.423)	**0.353 (0.419)**	**0.350 (0.417)**	0.535 (0.461)	1.069 (0.757)	N/A

Table 1. Average NLPD (RMSE) for small (1K+ points) and large-scale (40K+ points) benchmarks under clustering partitioning.

the root mean squared error (RMSE). [2] [3]

Baselines: We consider the gPoE, rBCM and barGP with random and K-means partitioning to assess the effect of the data assignment strategy. For the rBCM, gPoE and barGP, we evaluate the proposed softmax weighting strategy (BAR_var, rBCM_var, gPoE_var) with different temperature choices as proposed in Section 4. We also evaluate differential entropy (_entr) weighting (Cao & Fleet, 2014) and uniform weighting (_unif). According to Remark 2, when using normalised weights, the gPoE and rBCM are equivalent. We thus combine their results into 'rBCM/gPoE_...'. We further compare to the grBCM (Liu et al., 2018) and showcase results of this model obtained by averaging in y-space as proposed in (Liu et al., 2018), and in f-space. For all other expert models, averaging is done in f-space following (Deisenroth & Ng, 2015). Finally, we also consider a full GP baseline and linear regression.

5.1. Regression

We evaluate the performance of our approach to setting local experts' weights and compare it to previous weighting methods. In particular, we evaluate the robustness of the rBCM using differential entropic weighting as motivated by Deisenroth & Ng (2015), and the gPoE and barycenter with softmax-variance weighting (proposed in this paper), when reducing the number of points per experts. As motivated in Section 4, the softmax weighting should encourage expert sparsity, and as such be effective when the number of points per experts decreases (causing the number of strong experts to decrease). In this case, we set the temperature T to 15 (for $T \geq 15$, sparsity is well-controlled; see Figure 4).

Figure 3 shows that the gPoE with softmax-variance weighting provides sensible and calibrated predictions even with only two points per experts, while the rBCM with differential entropic weights leads to erratic mean and variance behaviours in the transitioning region even with 20 points per experts. Thus, encouraging sparsity in the expert weights through the variance-softmax weighting enables expert models to be robust to the reduction in the number of points per experts, thereby addressing a shortcoming of local-expert models. Also, the erratic behaviour in the transitioning region appears remediated. With very weak experts, it is unrealistic to expect uncertainties that are identical to the full GP's uncertainty. Importantly, the predictions are (moderately) on the conservative side for the softmax-variance weighting, which is preferable to overconfidence. We report similar behaviours for the barycenter combination (Section 3) in the Appendix (Figure 7).

We now perform a large-scale evaluation of the different expert models with different choices of weighting, including our approach (softmax-variance) and previous approaches (uniform for gPoE and differential entropy for rBCM) on 7 datasets of size ranging between 1000 and 500000. We also use SVGP$_{500}$, grBCM, and linear regression baselines. For softmax weightings, we use a temperature of 100, which performs well across small and large-scale benchmarks (i.e., it induces enough weight sparsity). We provide extensive additional results with different temperatures and random partitioning in the Appendix.

Table 1 shows that the gPoE and the barGP with softmax-variance weighting perform on par with grBCM and outperform all other models on all datasets. They significantly outperform the SVGP with 500 inducing points, but also the rBCM with differential entropy weighting, and linear regression across benchmarks. Moreover, the gPoE with softmax-variance weighting outperforms the gPoE with uniform weighting by a large margin across small and large-scale datasets. Also, whilst the grBCM outperforms rBCM with differential entropic weights across datasets, our weighting proposal (softmax-variance) outperforms or performs on par with it on all datasets, besides on Kin40k, while having a significantly lower prediction cost.

This demonstrates that controlling the sparsity of expert weights heals issues of the product-of-expert models and leads to more calibrated uncertainty quantification and mean estimation, while having the same running cost (and a significantly lower prediction cost than grBCM). The complexity of predictions of the grBCM is $8\times$ higher than under other expert models, because children datasets are aggregated with the master's. Therefore, these performance gains are

[2] Code available at https://github.com/samcohen16/Healing-POEs-ICML

[3] Datasets are from https://github.com/hughsalimbeni/bayesian_benchmarks.

accompanied by computational gains.

Finally, Liu et al. (2018) and Zhang & Williamson (2019) found that the rBCM and the gPoE under-perform when averaging in y-space, which is the reason we average in f-space in this paper. We analyse the performance of gr-BCM in both regimes, and observe that the latter leads to substantial performance gains, thus motivating averaging in f-space for all product-of-expert models (see Tables 1 and 8). We provide a more thorough discussion in Section 6.

5.2. Sensitivity and Robustness Analysis

We now consider the sensitivity of the gPoE, rBCM and barGP with softmax-variance weighting to the temperature hyperparameter T. For the gPoE and barGP, we use the normalised version of the softmax (in which case the gPoE is equivalent to the rBCM with such weights). We also evaluate the rBCM's robustness, when using unnormalised softmax-variance weights. To that end, we consider the Parkinsons, Kin40k, Airfoil and Concrete datasets, and plot the NLPD as a function of the temperature (Figure 4). We observe that the NLPD decreases monotonically until stabilising for both the gPoE and the barGP, demonstrating the robustness of these models with respect to the choice of the temperature parameter. Hence, the NLPD is stable across temperatures (for $T > 15$) when using normalised weights. We also produce such an analysis for unnormalised softmax-variance weights (in which case the rBCM is not equivalent to the gPoE). In this case, the model is more sensitive to the change in temperature, and it is difficult to find a single softmax scaling that performs well across small- and large-scale benchmarks. Hence, using normalised softmax weights is important to obtaining models that are robust to the choice of temperature.

5.3. Classification Benchmarks

We now assess the classification performance of expert models in a non-conjugate multi-class classification setting (MNIST dataset). The dataset comprises 10 classes with a training/test split of $60,000/10,000$ images. We reduce the dimensionality of images with PCA (20 principal components). Note that the overall accuracy resulting from PCA features will not outperform the state of the art. However, PCA features provide a deterministically reproducible basis for relative comparison of various methods. We assign 500 training points to each SVGP expert, and provide them with 100 trainable inducing inputs each. We use a multiclass likelihood with a robust-max link function. Note that in the setting of Liu et al. (2018), classification is not directly applicable because averaging is happening in y-space, which is more challenging in non-conjugate settings.

Table 2 shows classification results. We report top-n accuracy and NLPD. Consistently, we observe that the BAR_var

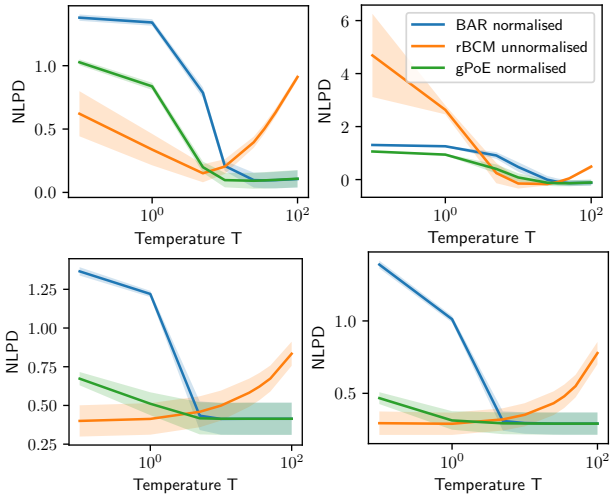


Figure 4. NLPD against temperature for different expert models with softmax-variance weighting on Parkinson (top left), Kin40K (top right), Airfoil (bottom left) and Concrete (bottom right).

and gPoE_var outperform all products of experts and SVGP baseline models. The difference in performance between the rBCM_entr and gPoE_var shows that introducing weight sparsity via a tempered softmax improves the performance as it only allows confident experts to contribute to the aggregated predictions. We observe similar performance gaps between gPoE_unif and our proposals which suggests that using tempered softmax-variance weighting results in more informed posterior predictive means and variances.

The improvement of the SVGP_{100} expert models over a single (full) SVGP_{500} is not surprising since every single SVGP expert has the modelling capacity of the global SVGP, so that the distributed models effectively work with M times as many inducing inputs as the SVGP. This suggests that the combination of sparse GPs and expert models can be useful in settings, where a large number of inducing inputs for a full SVGP is required for good modelling.

6. Discussion

In light of empirical results, we aim to explain our findings and compare them to those of similar papers, in particular (Liu et al., 2018), (Trapp et al., 2019) and (Zhang & Williamson, 2019).

All three papers reported poor NLPDs and RMSEs for the PoE, the BCM, but also the gPoE and the rBCM across small and large-scale benchmarks. At first sight, this seems to contradict the empirical observations drawn in this paper. We thus aim to disentangle such conflicting conclusions. The main reason for the discrepancies is that those papers depart from the framework of Deisenroth & Ng (2015) and aggregate GP predictions in y-space instead of f-space.

	BAR_var	gPoE_unif	gPoE_var	rBCM_entr	rBCM_var_u	SVGP_{500}
Top-1-accur.	**0.900**	0.877	**0.901**	0.878	0.880	0.873
Top-2-accur.	**0.958**	0.947	**0.958**	0.948	0.949	0.942
Top-3-accur.	**0.977**	0.971	**0.977**	0.972	0.9726	0.967
NLPD	**0.345**	0.412	**0.344**	0.947	0.821	0.451

Table 2. Top-n accuracy and NLPDs on the MNIST dataset (PCA features). Here, rBCM var has unnormalised weights.

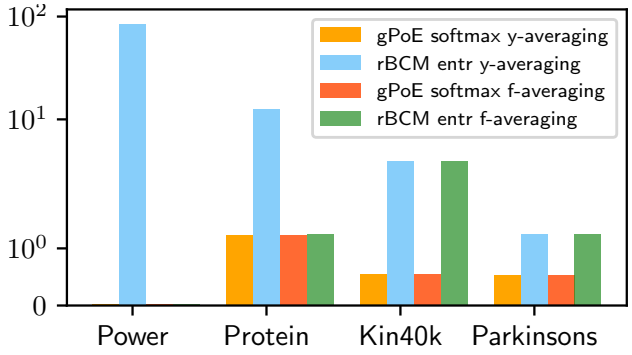


Figure 5. NLPD of the gPoE with our softmax proposal, and the rBCM with differential entropy under f- and y- averaging regimes (random partitioning). The latter model leads to weak performance under y-averaging whilst our proposal is robust in both regimes.

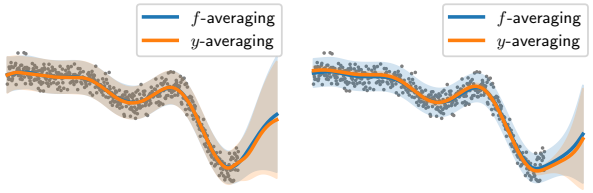


Figure 6. Predictions with BarGP softmax-var (Left) and rBCM diff-entropy (Right) in the f- and y- averaging regimes. The former works in both whilst the latter's variance shrinks under y-averaging

However, this is problematic for such models, especially with unnormalised weights and random partitioning, because the variance shrinks for all such models, explaining the bad NLPDs reported in all three papers.

In Figure 5, we show that with random partitioning, the rBCM with differential entropy has weak performance using y-averaging (in particular on power and protein), which is similar to results reported in all three papers, whilst its performance is good when using f-averaging. This explains the discrepancy between the results in (Deisenroth & Ng, 2015) and in our paper (f-averaging) and the results reported by Liu et al. (2018); Trapp et al. (2019); Zhang & Williamson (2019) (y-averaging). The problem with y-averaging has to do with weight calibration. We see in Figure 5 that our calibrating approach leads to strong performance in both y- and f- averaging, thus healing PoEs in both settings. We also provide an illustration of such results in Figure 6, using data from (Liu et al., 2018). We observe that when using differential entropic weighting, we recover the poor results reported bu Liu et al. (2018); Zhang & Williamson (2019), in which the variance shrinks significantly under y-averaging (whilst the model is sensible under f-averaging). By contrast, when using softmax-variance weighting, which calibrates the model, sensible results are obtained for both f- and y- averaging, further corroborating the quantitative results of Figure 5. We argue that using normalised weights for the rBCM is essential. Otherwise, even in the case, where the number of experts is $M = 1$, the rBCM does not have a predictive distribution equivalent to the one of a full GP; see Proposition 3 (Appendix), which gives an intuition for the erratic behaviours typically observed in the transitioning region for rBCM with unnormalised weights (e.g., Figure 1(d)). We therefore advise practitioners to average in f-space, use normalised weighting approaches and appropriate weight calibration.

7. Conclusion

We identified significant shortcomings of previous approaches, notably the PoE, BCM, gPoE and rBCM, to scaling GP regression and classification via local-expert averaging. These models struggle in settings, where the number of strong experts is small, but the experts' weights are not sparse enough. Weight sparsity should thus be set to account for the overall strength of experts. To address these shortcomings, we control weight sparsity via the use of (normalised) softmax weights, along with a temperature to enforce this trade-off. Note that our approach can be combined with SVGPs (Hensman et al., 2015) (as was done in Section 5.3) but also with other methods, such as KISS-GP (Wilson & Nickisch, 2015). We provide strong empirical evidence that shortcomings of previous expert models can be addressed through this approach, which leads to substantial performance gains across benchmarks. We further propose a novel scalable and distributable approach to averaging GP experts' predictions by means of Wasserstein barycenters, which can be used for regression and classification problems. When combined with our weighting proposal, it obtains state-of-the art performance across most datasets.

Acknowledgements

We are grateful to James T. Wilson for constructive feedback on the paper. We would also like to thank Martin Trapp and Michael Zhang for providing code and helping in investigating characteristics of expert models. SC is supported by the Engineering and Physical Sciences Research Council [grant number EP/S021566/1]. RM is supported by a Google PhD Fellowship in Machine Learning. TM is supported by the National Research Foundation of South Africa.

References

Bauer, M., van der Wilk, M., and Rasmussen, C. E. Understanding probabilistic sparse Gaussian process approximations. In *NIPS*, 2016.

Bertone, G., Deisenroth, M. P., Kim, J. S., Liem, S., Ruiz de Austri, R., and Welling, M. Accelerating the BSM interpretation of LHC data with machine learning. *Physics of the Dark Universe*, 2019.

Bishop, C. M. and Svensén, M. Bayesian hierarchical mixtures of experts. In *UAI*, 2003.

Bonneel, N. and Pfister, H. Sliced Wasserstein barycenter of multiple densities. *Harvard Technical Report*, 2013.

Cao, Y. and Fleet, D. J. Generalized product of experts for automatic and principled fusion of Gaussian process predictions. *arXiv:1410.7827*, 2014.

Cao, Y. and Fleet, D. J. Transductive log opinion pool of Gaussian process experts. *arXiv:1511.07551*, 2015.

Csato, L. and Opper, M. Sparse online Gaussian processes. *Neural Computation*, 2002.

Deisenroth, M. P. and Ng, J. W. Distributed Gaussian processes. In *ICML*, 2015.

Gal, Y., van der Wilk, M., and Rasmussen, C. E. Distributed variational inference in sparse Gaussian process regression and latent variable models. In *NeurIPS*, 2014.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *ICML*, 2017.

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian processes for big data. In *UAI*, 2013.

Hensman, J., de G. Matthews, A. G., and Ghahramani, Z. Scalable variational Gaussian process classification. In *AISTATS*, 2015.

Hinton, G. E. Products of experts. In *ICANN*, 1999.

Liu, H., Cai, J., Wang, Y., and Ong, Y.-S. Generalized robust Bayesian committee machine for large-scale Gaussian process regression. *arXiv:1806.00720*, 2018.

Platt, J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 1999.

Quiñonero Candela, J. and Rasmussen, C. E. A unifying view of sparse approximate Gaussian process regression. *JMLR*, 2005.

Rasmussen, C. E. and Ghahramani, Z. Infinite mixtures of Gaussian process experts. In *NeurIPS*, 2001.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

Rullière, D., Durrande, N., Bachoc, F., and Chevalier, C. Nested Kriging predictions for datasets with large number of observations. *Statistics and Computing*, 2018.

Snelson, E. and Ghahramani, Z. Sparse Gaussian processes using pseudo-inputs. In *NeurIPS*, 2006.

Titsias, M. K. Variational learning of inducing variables in sparse Gaussian processes. In *AISTATS*, 2009.

Trapp, M., Peharz, R., Pernkopf, F., and Rasmussen, C. E. Deep structured mixtures of Gaussian processes. In *AISTATS*, 2019.

Tresp, V. Mixtures of Gaussian processes. In *NeurIPS*, 2000a.

Tresp, V. A Bayesian committee machine. *Neural Computation*, 2000b.

Villani, C. *Optimal Transport: Old and New*, volume 338. Springer Science & Business Media, 2008.

Wang, K. A., Pleiss, G., Gardner, J. R., Weinberger, K. Q., and Wilson, A. G. Exact Gaussian processes on a million data points. In *NeurIPS*, 2019.

Wilson, A. G. and Nickisch, H. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *ICML*, 2015.

Zhang, M. M. and Williamson, S. A. Embarrassingly parallel inference for Gaussian processes. *Journal of Machine Learning Research*, 2019.

Álvarez Esteban, P. C., del Barrio, E., Cuesta-Albertos, J. A., and Matrán, C. A fixed-point approach to barycenters in Wasserstein space. *Journal of Mathematical Analysis and Applications*, 2016.